ISOGEOMETRIC BOUNDARY ELEMENT FORMULATION FOR DEFORMABLE PARTICLES IN MICROCHANNEL CONFINEMENT

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING

By

Özgür Can Gümüş

August 2023

Isogeometric boundary element formulation for deformable particles in microchannel confinement
By Özgür Can Gümüş
August 2023

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Barbaros Çetin(Advisor)

Gökberk Kabacaoğlu

Ali Karakuş

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

ISOGEOMETRIC BOUNDARY ELEMENT FORMULATION FOR DEFORMABLE PARTICLES IN MICROCHANNEL CONFINEMENT

Özgür Can Gümüş
M.S. in Mechanical Engineering
Advisor: Barbaros Çetin
August 2023

Numerical simulations of deformable particles are essential to understand quantities that are not possible with experimental techniques. The boundary element method is an advantageous technique to analyze deformable particles in viscous flow conditions since it reduces the dimensionality of the problem by one for linear partial differential equations. Isogeometric boundary element formulation is proposed to model the motion of deformable particles which provides unique advantages in terms of the higher-order continuity of elements and exact geometry representation. Moreover, it enables the calculation of surface normal and curvature analytically. Deformable particles, more specifically droplets, may undergo high deformation which deteriorates the mesh. Moreover, numerical inaccuracies result in a nonphysical change in the volume of the particle. Hence, volume correction and mesh relaxation algorithms are implemented in the isogeometric boundary element method to alleviate the aforementioned numerical artifacts. Without loss of generality, droplets in free and bounded flow cases are formulated and several benchmark problems are solved to assess the accuracy of the proposed formulation. Isogeometric boundary element method supported by stabilization methods explained in the study allows for obtaining stable and accurate results with low-resolution simulations.

Keywords: Boundary element method, isogeometric analysis, deformable particles, droplet motion.

ÖZET

MİKROKANAL İÇİNDEKİ ESNEK PARÇACIKLAR İÇİN İZOGEOMETRİK SINIR ELEMAN YÖNTEMİ FORMÜLASYONU

Özgür Can Gümüş
Makine Mühendisliği, Yüksek Lisans
Tez Danışmanı: Barbaros Çetin
Ağustos 2023

Esnek parçacıkların sayısal simülasyonları, deneysel tekniklerle elde edilemeyen nicelikleri anlamak için temel öneme sahiptir. Sınır eleman yöntemi, viskoz akış koşullarında deformasyona uğrayabilen parçacıkların analizinde avantajlı bir tekniktir, çünkü lineer kısmi diferansiyel denklemler için problem boyutunu bir azaltır. İzogemetrik sınır elemanı formülasyonu, elemanların yüksek dereceden sürekliliği ve geometri modellemedeki hataları azaltmasından dolayı esnek parçacıkların hareketini modellemek için kullanılmıştır. Ayrıca, yüzey normalinin ve eğriliğinin analitik olarak hesaplanmasına olanak tanır. Daha spesifik olarak damlacıklar gibi deformasyona uğrayabilen parçacıklar, yüksek deformasyona uğrayabilir ve bu, ağ yapısının bozulmasına neden olur. Ayrıca, sayısal hatalar parçacığın hacminde fiziksel olmayan bir değişime yol açar. Bu nedenle, izogemetrik sınır eleman yönteminde, söz konusu sayısal hataları azaltmak için hacim düzeltme ve ağ gevşetme algoritmaları uygulanmıştır. Genelliği kaybetmeden, serbest ve sınırlı akış durumlarındaki damlacık hareketi formüle edilmiş ve önerilen formülasyonun doğruluğunu değerlendirmek için çeşitli örnek problemler çözülmüştür. Çalışmada açıklanan stabilizasyon yöntemleriyle desteklenen izogemetrik sınır elemanı yöntemi, düşük çözünürlüklü simülasyonlarla stabil ve doğruluk payı yüksek sonuçlar elde etmeye olanak tanır.

Anahtar sözcükler: Sınır eleman yöntemi, izogeometrik analiz, deforme olan parçacıklar, damlacık hareketi.

Acknowledgement

I would like to express my gratitude to my advisor Assoc. Prof. Dr. Barbaros Çetin who made this work possible. Working with him has been a rewarding experience, as he not only imparted his expertise as an educator but also treated me as a respected colleague. His constructive feedback, encouragement, and deep knowledge in the field have significantly contributed to the development of this research. I would like to thank Dr. Gökberk Kabacaoğlu for enriching the outcomes of this work. Learning from his extensive experience in boundary integral equations and deformable particles was valuable. I would also like to thank Dr. Ali Karakuş for his valuable suggestions.

I would like to express my appreciation to Microfluidics & Lab-on-a-Chip Research Group (MLRG) members Kaan Atak and Zahra Babaie for their continuous support. I would like to thank my dear friends from the graduate office; Aykut Bakan and Emirhan İnanç for making these two years amazing, Batuhan Emre Kaynak and Enise Kartal for the pleasant memories we had, and Abdullah Habboush for being a loyal office companion. I would also like to thank Ahmet Faruk Keleş, Burak Sarı, Hande Nur Açıkgöz, Sayedus Salehin, Uzay Tefek, Fahim Shariar, Doğa Dağ, Yusuf Uzun, and Berkan Dedeoğlu for making the graduate office a better place.

Finally, I would like to express my heartfelt gratitude to my family for being the source of my passion.

Contents

1 Introduction **1**

1.1 Motivation and Objectives 6

1.2 Thesis Outline 8

2 Isogeometric Formulation **9**

2.1 Two-dimensional Isogeometric Analysis 10

2.1.1 Benchmark I: Two-dimensional Heat Transfer 16

2.2 Three-dimensional Isogeometric Analysis 21

2.2.1 Benchmark II: Three-dimensional Heat Transfer 27

2.2.2 Benchmark III: Three-dimensional Stokes Flow 30

2.2.3 Benchmark IV: Moving Sphere in Stokes Flow 35

3 Two-Dimensional Droplet Motion **37**

3.1 Boundary Element Formulation 37

3.2 Results and Discussion 42

4 Three-Dimensional Droplet Motion **47**

4.1 Unbounded Flow 48

4.1.1 Volume correction 49

4.1.2 Mesh relaxation 49

4.2 Bounded Flow 57

5 Concluding Remarks **64**

List of Figures

2.1 B-spline (A) basis functions and (B) curve with control points representation a square 11

2.2 B-spline (A) basis functions and (B) curve compared with the exact circle 12

2.3 (A) Basis functions and (B) control points after the knot insertion 13

2.4 (A) Basis functions and (B) control points after degree elevation . 14

2.5 NURBS (A) basis functions and (B) control points compared with an exact circle 15

2.6 (A) The schematic representation of the problem. (B) The positions of control points, nodes (collocation points), and element boundaries 17

2.7 (A) Post-process and (B) temperature distribution of the benchmark problem 20

2.8 MAPE of the temperature values on the surfaces where symmetry boundary condition is defined 21

2.9 Mapping B-spline surface from parameter space to physical space 22

2.10 (A) The schematic of the problem, and (B) the subdivision of the isogeometric element 23

2.11 The schematic of the problem with subdivision of the degenerated NURBS element 25

2.12 (A) The schematic representation of the problem (B) Subdivision of the element into triangles in parameter space 27

2.13 (A) Schematic of the problem, and (B) the surfaces of the computational domain generated by quadratic NURBS 29

2.14 (A) Temperature distribution, and (B) the convergence of the proposed formulation 29

2.15 (A) Schematic of the problem domain. (B) NURBS elements and collocation points 30

2.16 (A) Schematic of the problem domain, and (B) the elements and collocation points 31

2.17 The convergence of the solver with reference values taken from [1] 32

2.18 MAPE values of the pressure gradient with respect to DoF 33

2.19 The convergence of the solver compared with results available in the literature 34

2.20 The magnitude of the velocity of the flow field around the sphere obtained by using 482 collocation points 35

3.1 Time evolution of the particle major and minor axes 41

3.2 The effect of volume correction to Taylor deformation parameter and volume 42

3.3 The effect of mesh relaxation in high-deformation regime 43

3.4 (A) Schematic representation of the benchmark problem and (B) Taylor deformation parameter for different channel widths 44

3.5 Time evolution of two particles between parallel plates with mesh relaxation and volume correction algorithms 45

4.1 Taylor deformation parameter and non-dimensional volume history of the droplet with and without employing volume correction for $\lambda = 1$. 51

4.2 Volume changes for droplets with different viscosity ratios 52

4.3 Droplet shapes for $Ca = 0.35$, $\lambda = 1.0$ with and without mesh relaxation (number of collocation points = 202) 53

4.4 Time evolution of droplet in a shear flow ($Ca = 0.5$, $\lambda = 0.1$): (A) $\tau = 0$, (B) $\tau = 4$, (C) $\tau = 8$ and (D) $\tau = 12$ 55

4.5 Taylor deformation parameter, D, at different Ca values 57

4.6 (A) The schematic representation of the droplet between parallel plates (B) Elements on the plate 59

4.7 Deformation parameters at different confinement ratios. Blue and yellow lines correspond to the analytical [2] and BEM [3] results, respectively. Black markers represent IGABEM solutions. 60

4.8 Evolution of the deformation parameter in time for $Ca = 0.2$ and $R/W = 0.8$ for the droplet between parallel plates 61

4.9 (A) Isometric and (B) upper view of the Y-junction microchannel 62

4.10 Trajectories of the droplet and rigid particle 62

4.11 The final shapes of the droplet in (A) $x - y$ plane (B) $y - z$ plane 63

List of Tables

2.1 The integration results of Eq. (2.42) obtained by different techniques 24

2.2 The integration results of Eq. (2.44) obtained by different techniques 26

2.3 The integration results of Eq. (2.46) obtained by polar-coordinate transformation 28

4.1 Effect of mesh relaxation. Note that the analytic solution is valid only for small deformation. Hence the high-resolution solutions should be considered ground truth for these high-deformation cases. 54

Chapter 1

Introduction

Deformable particles appear in many industrial and biological applications and play a vital role in the end-properties of immiscible polymer blends [4]. For example, micro-explosions of emulsion droplets are known to be critical for fuel efficiency [5]. Apart from industrial applications, deformable particles may also be employed as a delivery vehicle for biomedical applications [6]. Due to the small velocity and length scales of the particulate flows, the viscous forces dominate, and the flows are governed by the Stokes equation in the limit of vanishing Reynolds number. Deformable particles can be categorized based on their extensibility. The surface area of extensible particles (e.g., droplets) changes to minimize the free energy of the interface, which makes it an extensible body with constant volume [7]. Extensible particles might undergo severe elongation which results in topological changes like break-up which is not the case for inextensible particles (e.g., vesicles). From the theoretical modeling perspective, an analytical description of the interfacial dynamics is a challenging task, especially for high deformation regimes owing to the presence of nonlinearities. Therefore, numerical methods play a vital role in modeling interfacial dynamics and estimating quantities like local stresses which may not be possible through experimentation. Numerical techniques developed to simulate moving boundary problems are categorized mainly into two groups:

- Interface capturing method
- Interface tracking method

Interface-capturing methods rely on a fixed-spatial domain where the interface is described by a function. The resolution of the interface depends on the grid employed on the fixed-spatial domain [8]. Several examples of interface-capturing methods can be listed as Lattice-Boltzmann method, level-set method, and volume of fluid method. These methods are especially useful for resolving the breakup and coalescence of droplets since they do not require additional meshing operations when topological changes occur as in interface tracking methods [9]. In Lattice-Boltzmann method (LBM), simplified kinetic equations are utilized to model the macroscopic behavior of the system without solving the full Boltzmann equations [10]. The availability of highly parallel algorithms makes LBM a viable tool to study moving boundary problems. A comparison between LBM and the front-tracking finite difference method was made in [11] to simulate the motion of bubbles. Although qualitatively similar results were obtained from both methods, limitations of LBM related to compressibility were mentioned in the paper. In [12], multiple-relaxation time LBM was employed to study the motion of a water droplet in a microchannel. A parametric study was conducted to understand the penetration dynamics of the droplet through micropillars inside the channel where breakup phenomenons were captured by the proposed formulation.

Level-set method is another interface-capturing method that computes the motion of an interface by defining a level-set function [13]. The level-set function is zero on the interface, positive inside the domain, and negative outside the domain bounded by the interface. It was applied to investigate droplet dynamics in the presence of an electric field [14]. Navier-Stokes equations were solved for the fluid flow and the Poisson equation, for the electric potential, was solved by using the ghost-fluid technique to resolve electric surface forces. Later, Choi *et al.* [15] employed a level-set method to understand the effects of droplet size, velocity, and porosity on droplet mobility and impact in a porous medium. Maintaining the conservation of mass is the main drawback of the level-set method which requires higher-order discretization schemes [16].

An additional function is defined in the volume of fluid method (VOF) to represent the fractional volume occupied in a cell by the fluid which is capable of capturing topological changes on the interface without requiring any rescaling of the mass [17]. The coalescence of a liquid column and a droplet was studied in [18] which analyzed the effect of coalescence on the propulsion mechanism inside a microchannel. The coalescence was observed to advance the liquid due to increased curvature which also increases the capillary forces. The droplet in a simple shear flow at different conditions was investigated in [19]. The results beyond steady configurations were validated by comparing the details of daughter droplets with experimental results. Note that, classical FEM techniques generally perform poorly with the interface capturing method since discontinuities appear inside the elements. The extended finite element method is proposed to overcome this problem by locally enriching FE space such that jumps or kinks can be exactly approximated [20].

Interface tracking methods are Lagrangian in which the interface is explicitly represented by the computational mesh. They are accurate and robust for simulating the interface until topological changes occur. The finite element method (FEM), immersed boundary method, and boundary element method can be given as examples of interface tracking methods. The arbitrary Lagrangian-Eulerian (ALE) technique was used to solve fluid-solid particle problems in [21] where a moving mesh strategy was employed by updating the mesh explicitly. Later, Quan and Schmidt [22] implemented the moving mesh technique to analyze two-phase flows including two-layer Couette flow and droplet in shear flow. They have also applied a mesh separation technique to resolve topological changes. Immersed boundary method proposed by Peskin [23] to solve fluid-structure interactions employs an Eulerian grid for the fluid and a Lagrangian grid for the structure. The forces exerted by the structure are used as a source term in the momentum equation of the fluid. This method was developed initially to model the flow around heart valves and has found many applications in fluid-structure interaction problems.

The boundary element method (BEM) is a numerical tool to solve partial differential equations by converting them into integral equations defined on the

surface of the computational domain if the governing equation is linear. It yields a system of linear equations after discretizing the surface. BEM has a semi-analytical nature since the boundary integral equations are obtained by an analytical derivation by virtue of the fundamental solutions. Numerical techniques are required only in the calculation of layer potentials and the solution of the linear system. The unknown field variables, the main variable and its derivative, are obtained after the solution of the system of equations. The boundary-only discretization feature of BEM reduces the dimension of the problem by one which reduces the computational cost. This feature of BEM makes it a suitable tool to analyze moving-boundary problems since no re-meshing is required inside the computational domain. It also allows analyzing unbounded domains without any bounding box due to the nature of fundamental solutions. The fluid flow problems in the creeping flow limit (*i.e.* vanishing Reynolds number) can be modeled effectively using BEM due to the linearity of the Stokes equation. Finite Reynolds problems can also be analyzed by BEM with an additional cost of domain integrals due to the presence of non-linearities in the governing equation by employing internal cells [24] or the radial integration method [25]. Moreover, the simulation of particulate flows both in unbounded [26] and bounded flows [27] can be also performed using BEM, especially when a rigorous modeling of particle-wall and particle-particle interactions are necessary [28–31].

In the pioneering work by Rallison and Acrivos [32] boundary integral equations have been used to study the motion of liquid droplets. The deformation and breakup conditions of a liquid droplet were introduced, but the study was limited to a certain range of viscosity ratios in an axisymmetric coordinate system. Planar droplet deformations in confined flow for both Newtonian and non-Newtonian fluids were investigated by Khayat *et al.* [33, 34] where two integral equations for the interface and solid boundaries are solved simultaneously. Pozrikidis [35] detailed the mathematical description of BEM for Stokes flow in the presence of interfaces with various properties like isotropic or elastic tension which basically changes the stress discontinuity term defined on the interface. The stress discontinuity term for a red blood cell is given in [36].

Even when the droplet flows are simulated with high spatiotemporal resolutions, numerical errors accumulate, and regardless of the numerical method, complex interface dynamics lead to mesh distortion. Hence, the simulations may lead to inaccurate results (such as non-physical changes in the droplet volume) and unstable solutions. To circumvent this issue, several mesh stabilization and relaxation methods have been proposed to avoid these numerical artifacts. Coulliette and Pozrikidis [37] used periodic Green's functions to analyze the motion of droplets in a circular tube positioned off the centreline where they have modified the tangential velocity such that the elements are relocated consistently with the flow profile. Zinchenko *et al.* [38] proposed to define a global tangential field to stabilize the mesh such that the internode distances are controlled. This method is extended later to take triangle quality and curvature into account [39] and applied to concentrated emulsion flows [40, 41]. Loewenberg and Hinch [42] proposed to employ a tangential flow field to maintain a uniform grid which is not capable of completely removing the flow-field dependency of the mesh. Siqueria *et al.* [43] extended that work such that the tangential flow field defined by control parameters is applied in a repetitive manner between each physical time step so that the Lagrangian characteristics of the mesh are removed. Passive and active mesh stabilization methods are applied in [44] to investigate magnetic fluid droplet dynamics and they have pointed out that passive stabilization techniques cannot stop mesh degradation completely.

The isogeometric analysis proposed by Hughes *et al.* [45] closes the gap between CAD and engineering analysis models by implementing the same parametric functions (generally in the form of B-splines) used to create geometry to approximate the unknown field variables. Boundary element method with isogeometric analysis for surface discretization (IGABEM) has been successfully implemented for many different problems such as acoustics [46], heat transfer [47], and linear elastostatics [48]. IGABEM was also utilized for the modeling of deformable particles in isoviscous conditions (i.e. viscosity ratio of unity) [49]. In IGABEM, B-splines or NURBS, which are the industry standard for storing geometry information, are employed for the exact geometry representation, and hence meshing operations are compressed since the same parametric functions are also used to represent the

field variables. Moreover, IGABEM offers higher continuity and accuracy (which are quite essential for modeling deformable particles) compared to Lagrangian basis functions. In the context of deformable particles, IGABEM has a unique opportunity related to normal vector and curvature calculations. For triangular meshes, an iterative best-paraboloid technique has been used to calculate the mean curvature and normal vector [38]. However, B-spline representation allows obtaining normal vector and mean curvature directly from the parametric equations by using the first and second fundamental forms of the surface [36]. Hence, the code implementation becomes straightforward.

1.1 Motivation and Objectives

IGABEM offers a unified way to model the motion of deformable particles in microchannel confinements with high accuracy. It allows us to analyze complex microchannel geometries in a straightforward way due to the integration of CAD and engineering analysis steps. Although IGABEM was used to simulate the motion of droplets and inextensible membranes [49], the study was limited to free flows in isoviscous conditions. In this study, IGABEM framework for modeling deformable particles under different flow conditions is presented. In general, deformable particles are differentiated by their stress discontinuity term which compensates the stress jump between different phases. The stress discontinuity term takes the following form for a droplet [36]:

$$\Delta\mathbf{f}(\mathbf{x}) = 2\sigma\kappa(\mathbf{x})\mathbf{n}(\mathbf{x}) \tag{1.1}$$

where σ is the interface tension. $\kappa(\mathbf{x})$ and $\mathbf{n}(\mathbf{x})$ represent the mean curvature and normal vector. For an inextensible membrane, it is defined as follows [36]:

$$\Delta\mathbf{f}(\mathbf{x}) = 2\tau\kappa(\mathbf{x})\mathbf{n}(\mathbf{x}) - \nabla_s\tau \tag{1.2}$$

where τ is the surface tension and ∇_s is the surface gradient operator. For vesicles or red blood cells, it can be expressed as:

$$\Delta\mathbf{f}(\mathbf{x}) = h\left[\frac{\kappa(\mathbf{x})^3}{2} - 2K\kappa(\mathbf{x}) + \Delta_s\kappa(\mathbf{x})\right]\mathbf{n}(\mathbf{x}) \tag{1.3}$$

where h is the membrane rigidity and K is the Gaussian curvature. Δ_s denotes the surface Laplacian. Hence, the proposed formulation constitutes the basis for analyzing different types of particles. Without loss of generality, droplets are analyzed but the stress jump term can be modified to simulate different types of particles.

Similar to other numerical methods, IGABEM also suffers from mesh distortion; therefore, volume correction and mesh relaxation are also required for efficient and stable simulations of deformable particles in Stokes flow with high accuracy. In this study, we propose an isogeometric boundary element formulation that stably and accurately models the motion of a droplet with arbitrary viscosity in free and confined flows, even at low resolutions. Volume correction is applied to ensure that the droplet volume is preserved. Moreover, the active mesh relaxation method, which is originally proposed for triangular elements by Siqueira *et al.* [43], is implemented to overcome the mesh degradation problem where non-physical tangential velocity is introduced. The proposed formulation allows accurate long-run simulations of a droplet for a wide range of capillary number values and viscosity ratios of fluids that are internal and external to the interface.

1.2 Thesis Outline

The rest of this thesis includes

Chapter 1 is the introductory chapter of moving boundary problems. Different numerical techniques to model interfacial dynamics are explained briefly. The boundary element method is introduced with applications in deformable particle simulations. The advantages of isogeometric discretization especially for deformable particle simulations are given. Needs for stabilization techniques are emphasized to obtain accurate and stable results with low-resolution simulations.

Chapter 2 introduces the isogeometric boundary element formulation. Algorithms for B-spline and NURBS curves/surfaces are explained. Isogeometric discretization is shown on boundary integral equations. Implementation details of quadratures are explained. Several potential and viscous flow problems are solved to assess the accuracy of the proposed formulation.

Chapter 3 introduces IGABEM formulation for the two-dimensional droplet motion. Boundary integral equations for both free and confined flows are shown. Volume correction and mesh relaxation algorithms are proposed to obtain accurate and stable results in 2D. Several benchmark simulations are performed to validate the formulation. The proposed formulation is tested for two-particle interaction between parallel plates problem.

Chapter 4 introduces IGABEM formulation for the three-dimensional droplet motion both in free and confined flows. Algorithmic details of the isogeometric discretization with stabilization methods are given. Benchmark problems for both flow types are used to assess the accuracy of the method where analytical solutions are available. Finally, the trajectories of rigid and deformable particles are compared for a flow in a microchannel.

Chapter 5 summarizes the outcomes of the study with concluding remarks. Future direction of the research is discussed.

Chapter 2

Isogeometric Formulation

Analytical solutions of integral equations are generally challenging and require numerical discretization. Hence, a mathematical description of the surface is necessary to calculate boundary integrals present in a typical BEM equation as shown below:

$$C(\mathbf{x_o})\phi(\mathbf{x_o}) + \int_\Gamma \mathcal{H}(\mathbf{x}, \mathbf{x_o})\phi(\mathbf{x})d\Gamma = \int_\Gamma \mathcal{G}(\mathbf{x}, \mathbf{x_o})q(\mathbf{x})d\Gamma \tag{2.1}$$

where ϕ and q represent field variables in which q is correlated with the first derivative of ϕ such as temperature and heat flux or velocity field and traction. $\mathcal{H}$ and $\mathcal{G}$ are the fundamental solutions of PDE and C is the free-term coefficient which takes different values depending on the position of the source point $\mathbf{x_o}$.

$$C(\mathbf{x_o}) = \begin{cases} 1, & \text{if } \mathbf{x_o} \in \Omega \\ \psi/2\pi, & \text{if } \mathbf{x_o} \in \Gamma \\ 0, & \text{if } \mathbf{x_o} \notin (\Omega \cup \Gamma) \end{cases} \tag{2.2}$$

where ψ is the interior angle of the boundary. The collocation strategy will be employed in all simulations presented in this study. Eq. (2.1) will be expressed for each source point (collocation points) which results in a system of equations. Spatial discretization of Eq. (2.1) will be performed by using B-splines or NURBS depending on the problem. First, a B-spline definition will be given which will lead to a more general surface representation, NURBS. It should be noted that

the NURBS toolbox developed by Spink [50] is utilized for algorithms required to create B-spline and NURBS curves/surfaces.

2.1 Two-dimensional Isogeometric Analysis

B-spline basis is a type of basis in which a curve is generated from a control polygon (or net in 3D) by allowing local changes on the curve [51]. This non-global property of B-splines is due to the fact that each vertex in the control polygon is associated with a different basis function. Two different entities are required to generate a B-spline curve: Control points and a knot vector. Control points are a set of points, $\mathbf{P} \in \mathbb{R}^2$, in the physical space which forms the control polygon. The knot vector, $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$, is a set of non-decreasing coordinates, $\xi \in \mathbb{R}$, chosen in parameter space where p is the order of the curve and n denotes the number of basis functions to define the curve. The coordinates in the parameter space will be mapped to the physical space by employing B-spline basis functions to generate the curve. Once the knot vector is given, one can define B-spline basis functions by utilizing Cox-de Boor recursion formula [51] for $p = 0$ as shown below:

$$N_{i,0}(\xi) = \begin{cases} 1, & \xi \leq \xi < \xi_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

For higher-order functions:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2.4}$$

After creating B-spline basis functions from the given knot vector, the curve in the physical space can be defined as follows:

$$\mathbf{x}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) \mathbf{P}_i \tag{2.5}$$

Two different examples are shown below for a square and a curve that approximates a circle. Control points for both shapes are the same and they can be listed as:

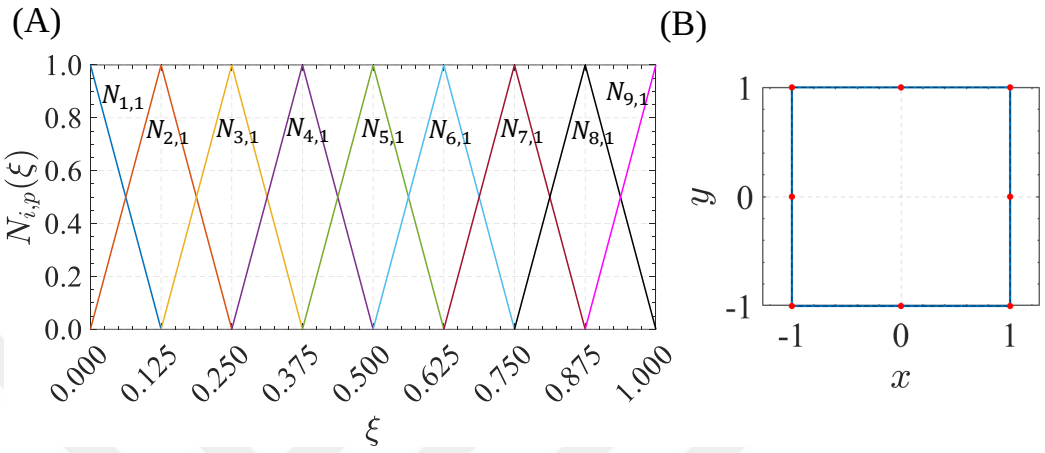

Figure 2.1: B-spline (A) basis functions and (B) curve with control points representation a square

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \end{bmatrix} \tag{2.6}$$

The first row of the P matrix denotes the x-coordinates of control points and the second row contains the y-coordinates of control points. Control points start and end at the same coordinate to create a closed spline. The knot vector to create a square is:

$$\Xi = \{0, 0, 0.125, 0.250, 0.375, 0.500, 0.625, 0.750, 0.875, 1, 1\} \tag{2.7}$$

The resulting basis functions are shown in Fig. 2.1A where 9 different linear unique basis functions are obtained. The red markers in the figure represent control points. The interval between each unique knot value is denoted as a knot span and they constitute the elements in the analysis. B-spline parameter space which is defined as $\xi \in \{0, 1\}$ is called a patch where multiple patches might be required for the analyses of complex shapes. Basis functions' values are 1 at unique knot values which correspond to control points in physical space. Hence, the curve passes through each control point as shown in Fig. 2.1B. On the contrary, the following knot vector results in cases where the curve does not pass through each control point:

$$\Xi = \{0, 0, 0, 0.25, 0.25, 0.50, 0.50, 0.75, 0.75, 1, 1, 1\} \tag{2.8}$$

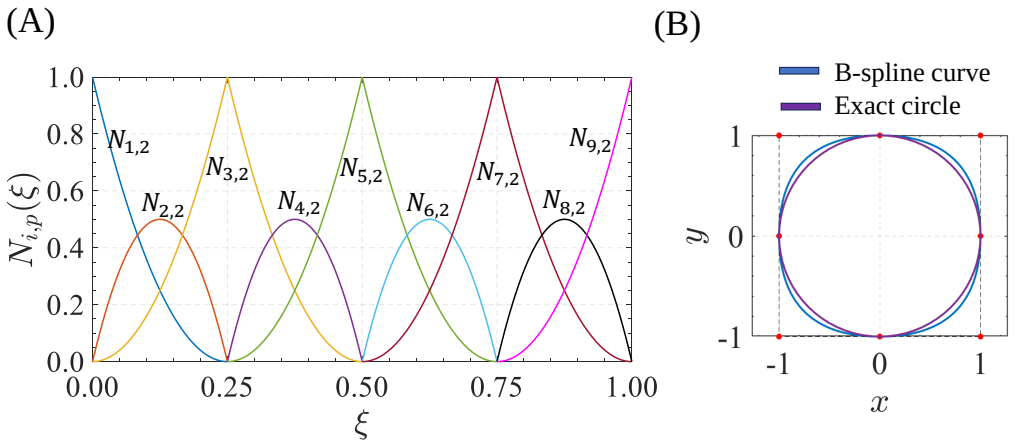

Figure 2.2: B-spline (A) basis functions and (B) curve compared with the exact circle

The resulting basis functions are shown in Fig. 2.2A where 9 different quadratic unique basis functions are obtained by using the knot vector shown above. The maximum value of $2^{nd}, 4^{th}, 6^{th}$ and 8^{th} basis functions is 0.5 hence the curve does not pass through control points corresponding to those basis functions as shown in Fig. 2.2B. The resulting shape can not represent a circle exactly. Rational equations, NURBS, are required to do that. The derivative of physical coordinates, x & y, with respect to parametric coordinate, ξ, might also be required when calculating the Jacobian or normal vector on a curve (or surface). The derivative of a B-spline can be expressed as shown below:

$$\frac{dN_{i,p}(\xi)}{d\xi} = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2.9}$$

Knot insertion is the counterpart of $h-$refinement in FEM literature. One can increase the number of basis functions and the corresponding control point without modifying the geometry. Suppose a new knot value, $\bar{\xi} \in \{\xi_k, \xi_{k+1}\}$, will be inserted into a knot vector $\Xi = \{\xi_1, \xi_2, ..., \xi_k, \xi_{k+1}, ..., \xi_{n+p+1}\}$. One has to generate $n+1$ new basis functions by employing Eq. (2.3) and Eq. (2.4). Moreover, the new $n+1$ control points are determined from the original set of control points, $\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_n$, as shown below [52]:

$$\bar{\mathbf{P}}_i = \alpha_i \mathbf{P}_i + (1 - \alpha_i)\mathbf{P}_{i-1} \tag{2.10}$$

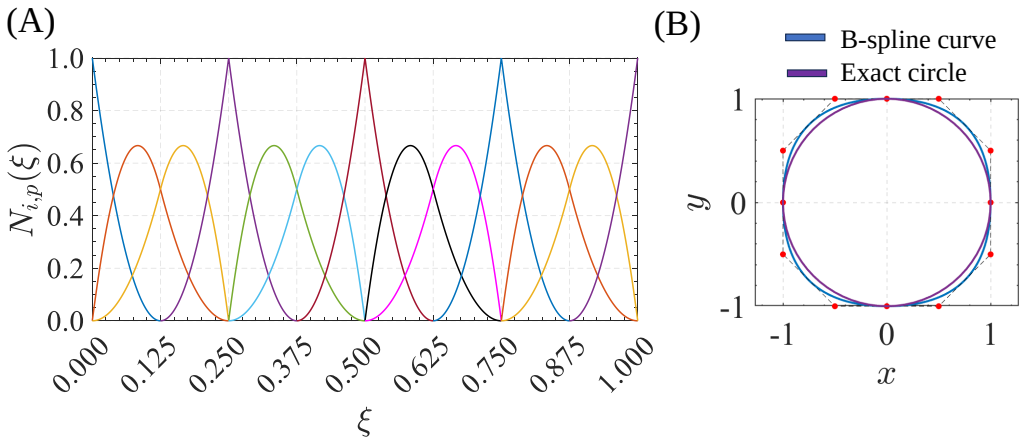

Figure 2.3: (A) Basis functions and (B) control points after the knot insertion

where

$$\alpha_i = \begin{cases} 1, & \text{if } 1 \leq i \leq k-p \\ \dfrac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & \text{if } k-p+1 \leq i \leq k \\ 0, & \text{if } k+1 \leq i \leq n+p+2 \end{cases} \tag{2.11}$$

An example of knot insertion is shown in Fig. 2.3 for the B-spline circle. The inserted knots are $\bar{\Xi} = \{0.125, 0.375, 0.625, 0.875\}$ which subdivides each original knot spans into two.

As can be seen from Fig. 2.3 that the number of basis functions and unique knot values in the patch has been increased due to the knot insertion. The modified control points can also be seen in Fig. 2.3B where the curve itself remains the same. Apart from $h-$refinement, there is another technique in FEM, called $p-$refinement, where the polynomial order of the basis functions is increased. The degree elevation for B-spline curves is performed by subdividing the curve into Bezier segments and increasing their degrees [51]. A one-degree elevated version of the B-spline circle is shown in Fig. 2.4 with modified basis functions and control points. It can be seen from Fig. 2.4A that although the number of knot spans or elements did not change the number of basis functions inside each element has been increased. One can combine both refinement techniques together, called $k-$refinement, which has no counterpart in standard techniques. The advantage of this technique is that one would have $q-1$ continuous derivatives at the inserted

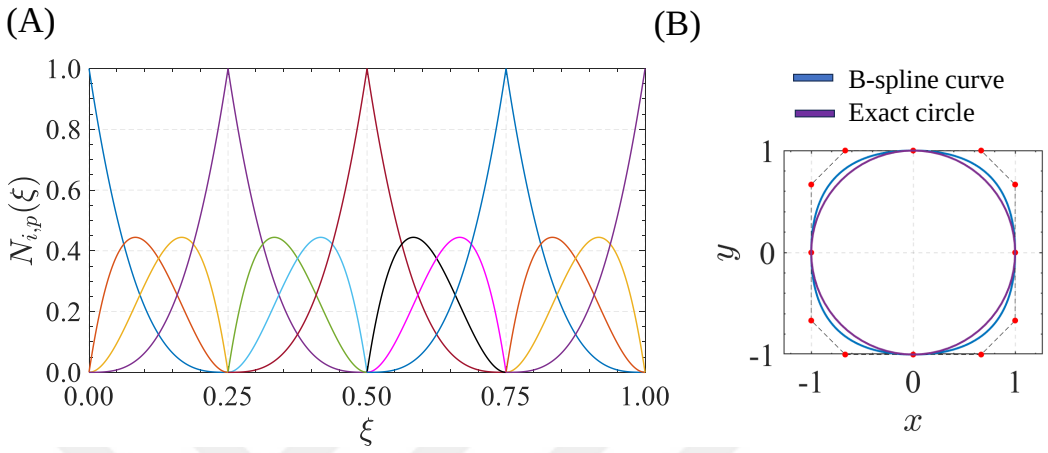

Figure 2.4: (A) Basis functions and (B) control points after degree elevation

knot $\bar{\xi}$ if the order is elevated to q initially and then the knot is inserted.

It is emphasized that B-splines are not capable of representing circular and conical shapes exactly. NURBS were introduced [53] to overcome this problem by using rational equations. To do that, B-spline basis functions are projected with a set of weights to NURBS basis functions where weights introduce biases to control points. If all weights are 1 then NURBS basis functions become simply B-spline basis functions. Hence, B-splines are a subset of NURBS. NURBS basis functions can be obtained as shown below where ω denotes a weight:

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi)\omega_i}{\sum_{j=1}^{n} N_{j,p}(\xi)\omega_j} \tag{2.12}$$

Once the NURBS basis functions are obtained from Eq. (2.12) the NURBS curve can be expressed similarly to a B-spline curve:

$$\mathbf{x}(\xi) = \sum_{i=1}^{n} R_{i,p}(\xi)\mathbf{P}_i \tag{2.13}$$

The exact circle can be obtained by modifying the basis functions of the B-spline circle shown in Fig. 2.2A with $\omega = \{1, \sqrt{2}/2, 1, \sqrt{2}/2, 1, \sqrt{2}/2, 1, \sqrt{2}/2, 1\}$. The resulting curve is shown below with markers selected on an exact circle.

All aforementioned refinement techniques are also applicable to NURBS curves. The derivatives of NURBS basis functions can also be computed where

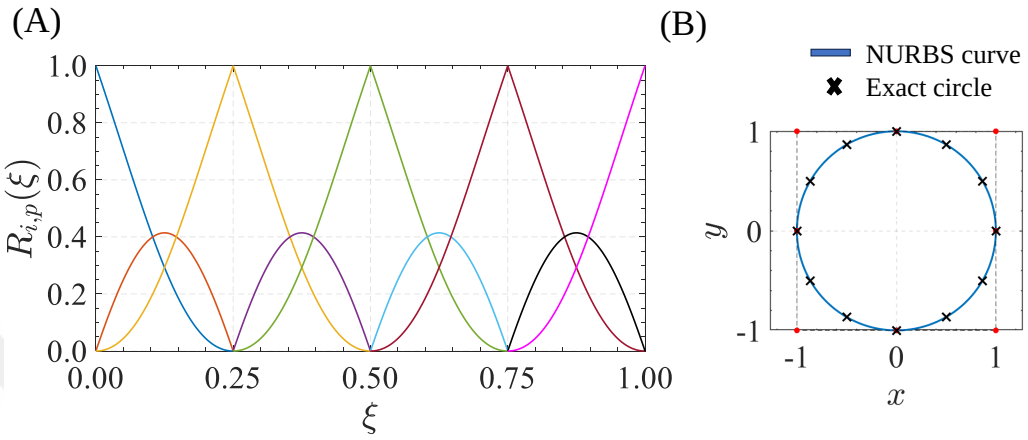

Figure 2.5: NURBS (A) basis functions and (B) control points compared with an exact circle

the details were shown in [51].

The basis functions used to represent the geometry will also be used to represent field variables such as temperature or flux. The representation of field variables by NURBS basis functions can be shown as follows [45]:

$$\phi(\xi) = \sum_{m=1}^{n} R_{m,p}(\xi)\Phi_m \tag{2.14a}$$

$$q(\xi) = \sum_{m=1}^{n} R_{m,p}(\xi)Q_m \tag{2.14b}$$

where Φ and Q are the field variable values at control points. The positions of collocation points in the parameter space are chosen by using the Greville abscissae definition [54]:

$$\xi_{o,i} = \frac{\xi_{i+1} + ... + \xi_{i+p}}{p} \tag{2.15}$$

which is basically the mean of p consecutive knots. The coordinates of source points can be calculated by mapping $\xi_{o,i}$ to the physical space. The spatial discretization of the BIE shown in Eq. (2.1) then becomes:

$$C(\mathbf{x_o}) \sum_{m=1}^{n} R_{m,p}(\xi_o)\Phi_m + \sum_{e=1}^{N^e} \mathrm{H}^{\mathrm{e}}(\mathbf{x_o})\Phi_m = \sum_{e=1}^{N^e} \mathrm{G}^{\mathrm{e}}(\mathbf{x_o})Q_m \tag{2.16}$$

where the discretized double and single-layer potentials are, respectively:

$$\mathrm{H}^e(\mathbf{x_o}) = \int_{\hat{\Gamma}^e} \sum_{m=1}^{M} R_{m,p}(\xi)\mathcal{H}(\mathbf{x_o}, \mathbf{x})J(\xi)\, d\hat{\Gamma}^e \tag{2.17a}$$

$$\mathrm{G}^e(\mathbf{x_o}) = \int_{\hat{\Gamma}^e} \sum_{m=1}^{M} R_{m,p}(\xi)\mathcal{G}(\mathbf{x_o}, \mathbf{x})J(\xi)\, d\hat{\Gamma}^e \tag{2.17b}$$

where $J(\xi)$ denotes the Jacobian from parameter to physical space and it is expressed as:

$$J = \frac{d\Gamma}{d\xi} = \sqrt{\left(\frac{d\mathrm{x}}{d\xi}\right)^2 + \left(\frac{d\mathrm{y}}{d\xi}\right)^2} \tag{2.18}$$

Writing down Eq. (2.16) for each collocation point yields a system of equations. The linear boundary conditions can be imposed on the system by applying column shifting which results in the following form:

$$\mathbf{H} \cdot \mathbf{x} = \mathbf{G} \cdot \mathbf{b} \tag{2.19}$$

where $\mathbf{x}$ and $\mathbf{b}$ are the unknown and linear boundary condition vectors. Note that, one has to map $\mathbf{x}$ to the physical space after solving the system to find the physical values of field variables at collocation points.

2.1.1 Benchmark I: Two-dimensional Heat Transfer

The governing equation for steady heat conduction in an isotropic solid can be expressed by Laplace's equation:

$$k\nabla^2\phi(\mathbf{x}) = 0 \tag{2.20}$$

where k is the thermal conductivity and ϕ denoting the temperature with $\mathbf{x}$ representing the spatial coordinates. Note that, an arbitrary source can also be included in the problem which can be solved by applying Radial Integration Method (RIM) which converts domain integrals to surface integrals by using radial integrals. The application of RIM to IGABEM with non-linear boundary conditions was shown in [47]. The problem domain is shown in Fig. 2.6A where symmetry boundary conditions are defined at boundaries corresponding to $\theta = \{0, \pi/2\}$.

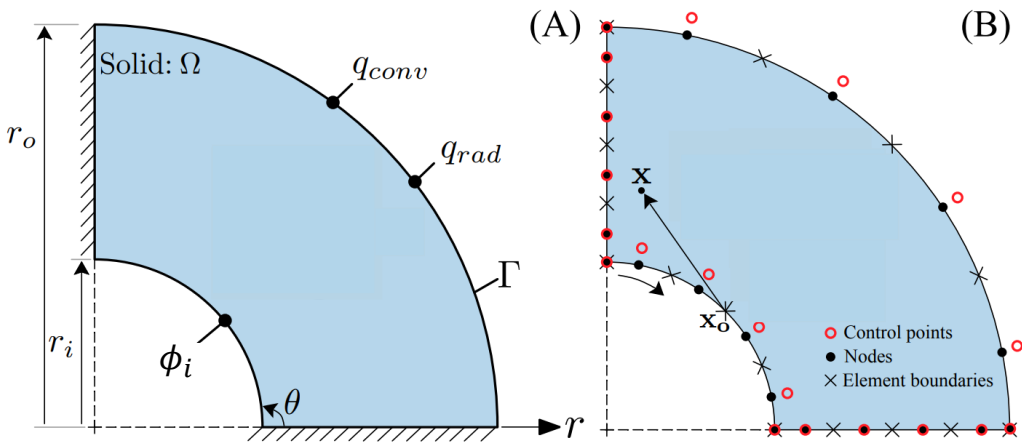

Figure 2.6: (A) The schematic representation of the problem. (B) The positions of control points, nodes (collocation points), and element boundaries

The boundaries of the computational domain are generated by using quadratic NURBS. Uniform temperature ϕ_i is defined at the inner surface. Convective and radiative boundary conditions are defined at the outer surface. The problem is 1D (in radial coordinates) due to the shape of the computational domain and the type of boundary conditions defined at the surfaces. The boundary integral equation for the problem described in Eq. (2.20) can be expressed as:

$$\begin{aligned} C(\mathbf{x_o})\phi(\mathbf{x_o}) = &\int_\Gamma \left[\mathbf{n}(\mathbf{x}) \cdot \nabla\phi(\mathbf{x})\right] \mathcal{G}(\mathbf{x_o}, \mathbf{x})\, d\Gamma \\ &- \int_\Gamma \phi(\mathbf{x}) \left[\mathbf{n}(\mathbf{x}) \cdot \nabla\mathcal{G}(\mathbf{x_o}, \mathbf{x})\right]\, d\Gamma \end{aligned} \tag{2.21}$$

The free space Green's functions for the Laplace problem in 2D can be expressed as:

$$\mathcal{G}(\mathbf{x_o}, \mathbf{x}) = \frac{1}{2\pi} \ln |\mathbf{r}| \tag{2.22}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{x_o}$ and $|\mathbf{r}|$ denotes the Eulerian distance between the field and source points with n_i is the i^{th} component of a unit outward vector. Single and double-layer potentials exhibit singularities due to the nature of free space Green's function. Singularities in the single-layer potential, the first term on the right-hand side of Eq. (2.21), are alleviated by applying Telles transformation [55] which applies a coordinate transformation such that the Jacobian cancels out the

singularity. It has been shown in [56] that the integrand in the double-layer potential tends to a constant as $r \to 0$ so no additional treatment is required for the calculation of double-layer potential. Regular integrals are calculated by applying Gauss-Legendre quadrature. Uniform temperature is defined at the inner surface:

$$\phi(\mathbf{x}) = \phi_D(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_D \tag{2.23}$$

The heat flux on a boundary can be expressed as:

$$q(\mathbf{x}) = k\frac{\partial\phi(\mathbf{x})}{\partial n} = -q_N(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_N \tag{2.24}$$

which is 0 on the surfaces where the symmetry boundary condition is defined. Note that, $\Gamma_D \cup \Gamma_N \in \Gamma_L$ where Γ_L represents the whole surfaces where linear boundary conditions are defined. Apart from linear boundary conditions, the convective and radiative boundary condition is defined at the outer surface which is expressed as:

$$q(\mathbf{x}) = k\frac{\partial\phi(\mathbf{x})}{\partial n} = -f(\phi), \qquad \mathbf{x} \in \Gamma_R \tag{2.25}$$

where the nonlinear function $f(\phi)$ can be expressed as:

$$f(\phi) = q_{conv} + q_{rad} = h(\phi - \phi_\infty) + \varepsilon(\phi^4 - \phi_\infty^4) \tag{2.26}$$

The discretized version of Eq. (2.21) can be expressed as Eq. (2.16). The system of equations can not be solved directly due to the presence of non-linearities in boundary conditions. Although the whole system can be solved by applying any non-linear solver, the variable condensation technique is employed to reduce the size of the system which is used in the non-linear solver. The idea behind variable condensation is manipulating the system of equations such that only the field variables residing on the surfaces where the non-linear boundary condition is defined are calculated inside the solver. The remaining values, collocation points on the surfaces where linear boundary condition is defined, are calculated by performing a post-process-like operation. The system of equations , Eq. (2.19) can be recast into the following form:

$$\begin{bmatrix} \widetilde{\mathbf{H}}_{LL} & \mathbf{H}_{LR} \\ \widetilde{\mathbf{H}}_{RL} & \mathbf{H}_{RR} \end{bmatrix} \begin{Bmatrix} \mathbf{X}_L \\ \mathbf{\Phi}_R \end{Bmatrix} = \begin{bmatrix} \widetilde{\mathbf{G}}_{LL} & \mathbf{G}_{LR} \\ \widetilde{\mathbf{G}}_{RL} & \mathbf{G}_{RR} \end{bmatrix} \begin{Bmatrix} \mathbf{b}_L \\ \mathbf{Q}_R \end{Bmatrix} \tag{2.27}$$

where L and R represent collocation points residing on surfaces where linear and non-linear boundary conditions are defined, respectively. $\mathbf{X}_L$ and $\mathbf{b}_L$ are the column vectors for unknowns and boundary conditions for collocation points at Γ_L. $\boldsymbol{\Phi_R}$ and $\mathbf{Q}_R$ are the potential and flux values of collocation points on Γ_R. The modification of entities inside the system matrices due to the application of linear boundary conditions is denoted by "˜ " symbol. The unknowns on Γ_L can be expressed as:

$$\mathbf{X}_L = \widetilde{\mathbf{H}}_{LL}^{-1} \cdot \left(\widetilde{\mathbf{G}}_{LL} \cdot \mathbf{b}_L + \mathbf{G}_{LR} \cdot \mathbf{Q}_R - \mathbf{H}_{RR} \cdot \boldsymbol{\Phi}_R \right) \tag{2.28}$$

The following expression can be obtained from the second row of Eq. (2.27) by utilizing Eq. (2.28):

$$\mathbf{y}\left(\boldsymbol{\Phi}_R\right) = \mathbf{A}_{RR} \cdot \boldsymbol{\Phi}_R - \mathbf{B}_{RR} \cdot \mathbf{f}\left(\boldsymbol{\phi}_R\right) - \mathbf{z}_R = 0 \tag{2.29}$$

where

$$\mathbf{A}_{RR} = \mathbf{H}_{RR} - \widetilde{\mathbf{H}}_{RL} \cdot \widetilde{\mathbf{H}}_{LL}^{-1} \cdot \mathbf{H}_{LR} \tag{2.30a}$$

$$\mathbf{B}_{RR} = \widetilde{\mathbf{G}}_{RR} - \widetilde{\mathbf{H}}_{RL} \cdot \widetilde{\mathbf{H}}_{LL}^{-1} \cdot \mathbf{G}_{LR} \tag{2.30b}$$

$$\mathbf{z}_R = \left(\widetilde{\mathbf{G}}_{RL} - \widetilde{\mathbf{H}}_{RL} \cdot \widetilde{\mathbf{H}}_{LL}^{-1} \cdot \widetilde{\mathbf{G}}_{LL} \right) \cdot \mathbf{b}_L \tag{2.30c}$$

$\mathbf{f}(\boldsymbol{\Phi})$ can be described by the boundary condition defined at the outer surface Eq. (2.26). $\mathbf{A}_{RR}, \mathbf{B}_{RR}$ and $\mathbf{z}_R$ are the coefficient matrices obtained after applying variable condensation and their expressions can be found from Eq. (2.30a) to Eq. (2.30c). Eq. (2.29) is a system of non-linear equations that are solved by employing the Newton-Raphson technique. Newton-Raphson iterations are performed until an error tolerance of $\left\|\bar{\mathbf{e}}\right\|_2 = 10^{-10}$ is achieved where:

$$\mathbf{e} = \boldsymbol{\phi}_R^{\mathbf{k+1}} - \boldsymbol{\phi}_R^{\mathbf{k}} \tag{2.31}$$

$$\left\|\mathbf{e}\right\|_2 = \sqrt{\left(\sum_j \left|\mathrm{e}_j\right|^2 \right)} \tag{2.32}$$

The non-dimensionalization for the radial coordinate and temperature can be expressed as $\bar{r} = r/r_i$ and $\bar{\phi} = \phi/\phi_\infty$ where ϕ_∞ denotes the temperature at infinity. Moreover, the following non-dimensional parameters can be defined for

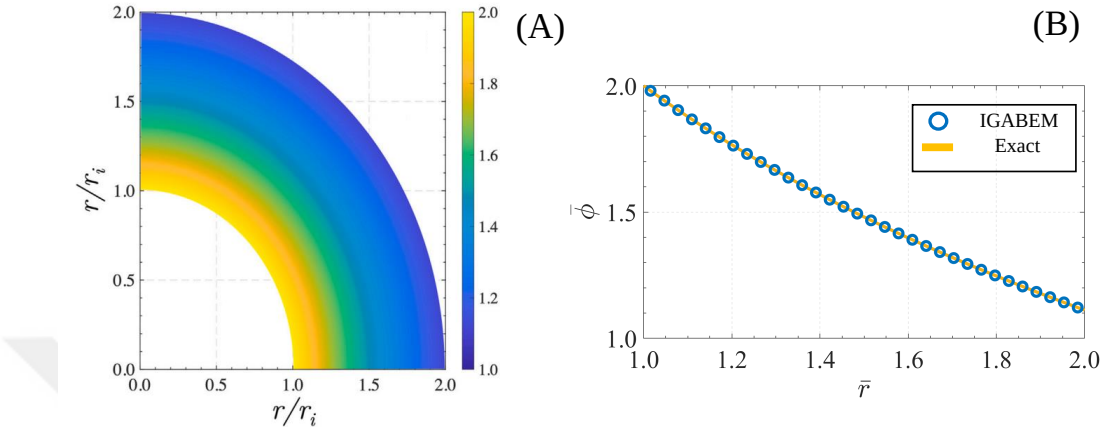

Figure 2.7: (A) Post-process and (B) temperature distribution of the benchmark problem

the boundary condition: $Bi = hr_i/k$, $Bi_r = (\varepsilon\sigma\phi_\infty^3)r_i/k$. The general solution to the 1D heat equation in cylindrical coordinates can be expressed as:

$$\bar{\phi}(\bar{r}) = C_1 \ln \bar{r} + C_2 \tag{2.33}$$

C_2 can be determined from the Dirichlet boundary condition defined at the inner surface since $\bar{r}$ becomes 1. The solution of a quartic equation is required to determine C_1 due to the radiative boundary condition defined at the outer surface. The following non-dimensional parameters are used for the benchmark problem: $\bar{\phi}_i = 2, \bar{r}_o = 2, Bi = 1$, and $Bi_r = 1$.

Post-process result of the benchmark problem is shown in Fig. 2.7A which is obtained by assigning unity to the free term coefficient in Eq. (2.21) after determining the values of field variables defined on the surfaces. Fig. 2.7B shows the temperature distribution obtained by the proposed formulation with a total of 132 collocation points and an exact solution. The temperature values demonstrated in Fig. 2.7B correspond to the collocation points on the surface where $\theta = 0$. The mean absolute percentage error is used to quantify the accuracy of the proposed formulation which can be expressed as:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\phi_i - \phi_{exact,i}|}{|\phi_{exact,i}|} \times 100 \tag{2.34}$$

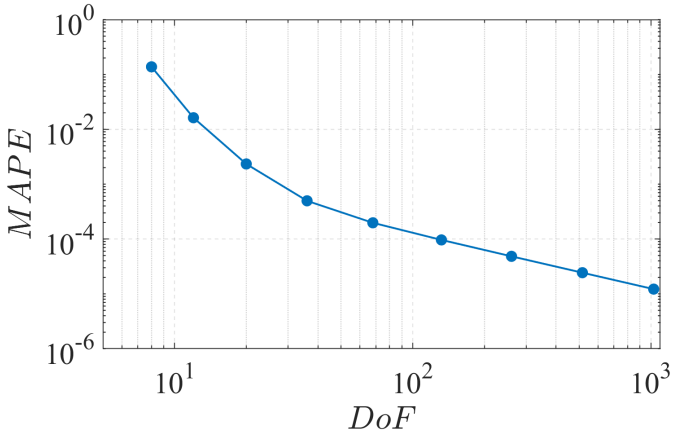

Figure 2.8: MAPE of the temperature values on the surfaces where symmetry boundary condition is defined

The convergence of the formulation is shown in Fig. 2.8. MAPE values of the temperature values on the surfaces where the symmetry boundary condition is defined are plotted with respect to the different numbers of collocation points (*i.e.*, degree of freedom, DoF). It can be seen from Fig. 2.8 that even using 1 element on each surface results in an error value close to 0.1% which decreases to 10^{-5}% as DoF increases. One can also employ Lagrangian basis functions instead of NURBS basis functions to represent field variables. Utilizing Lagrangian basis functions ease the code implementation due to their Kronecker delta property. It has been observed in [47] that higher accuracy is achieved by employing NURBS instead of Lagrangian basis functions with the same polynomial order.

2.2 Three-dimensional Isogeometric Analysis

Surfaces in 3D are generated by the tensor product of B-spline curves. Suppose there are two knot vectors $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$ and $H = \{\eta_1, \eta_2, ..., \eta_{m+q+1}\}$. The tensor product B-spline surface can be defined for the control net $\mathbf{P}_{i,j}$ as shown below:

$$\mathbf{x}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{P}_{i,j} \tag{2.35}$$

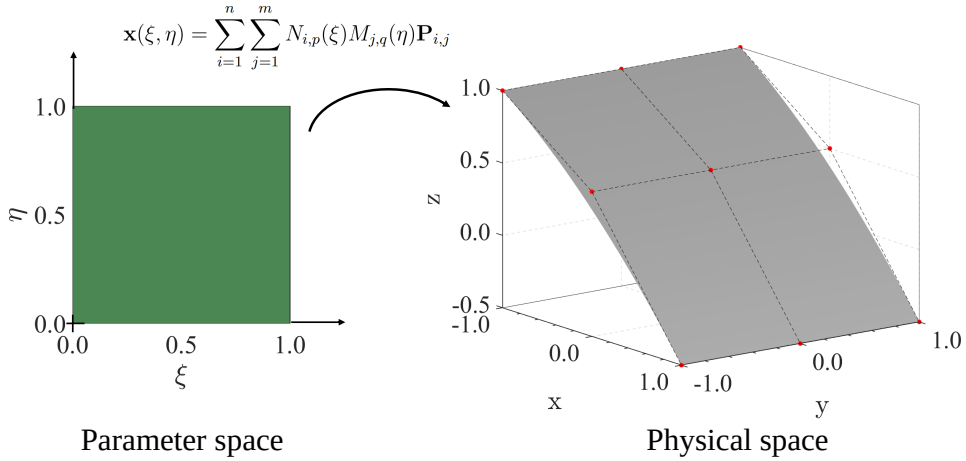

Figure 2.9: Mapping B-spline surface from parameter space to physical space

which maps parametric coordinates, $\boldsymbol{\xi} = [\xi, \eta] \in \mathbb{R}^2$, to physical coordinate system $\mathbf{x} \in \mathbb{R}^3$ as shown in Fig. 2.9. Bivariate NURBS basis functions can be obtained by modifying B-spline basis functions with weights, $\omega_{i,j}$, as shown below:

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_i(\xi)M_j(\eta)\omega_{i,j}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{n} N_{\hat{i}}(\xi)M_{\hat{j}}(\eta)\omega_{\hat{i},\hat{j}}} \tag{2.36}$$

Then, NURBS surfaces can be defined as:

$$\mathbf{x}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\mathbf{P}_{i,j} \tag{2.37}$$

Algorithms explained for B-spline and NURBS curves like knot insertion, and degree elevation can be directly applied to their surface counterparts. Greville's abscissae definition is employed for both knot vectors, Ξ and H, to determine the position collocation points in parameter space. Bivariate basis functions can be expressed in a compact form by using a single index for the bivariate basis functions $l = i + (j-1)m$ where $l \in [1, ..., nm]$:

$$R_l(\xi,\eta) = R_{i,p}(\xi)R_{j,q}(\eta) \tag{2.38}$$

The field variables can be represented as shown in Eq. (2.14a) and Eq. (2.14b). Then the discretization of the BIE, Eq. (2.1), will be in the form of Eq. (2.16).

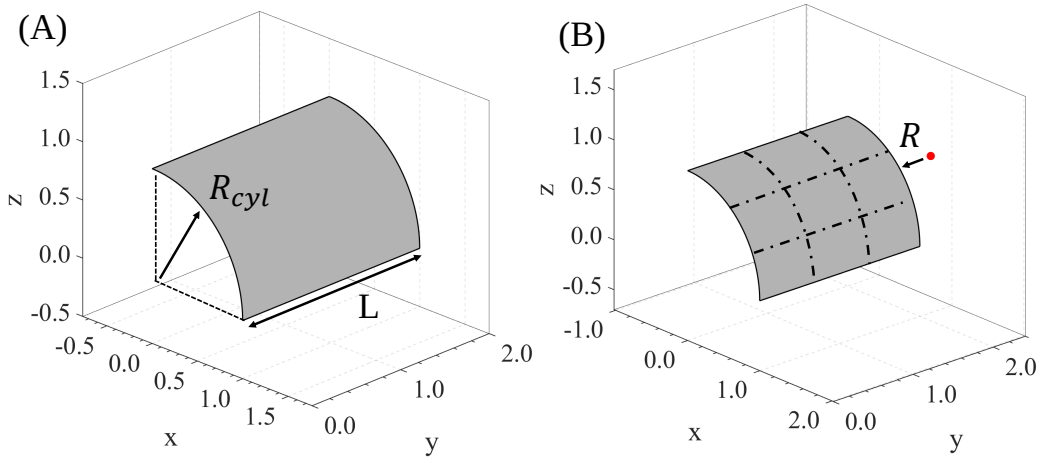

Figure 2.10: (A) The schematic of the problem, and (B) the subdivision of the isogeometric element

The double and single-layer potentials can be expressed as:

$$\mathrm{H}^e(\mathbf{x_o}) = \int_{\hat{\Gamma}^e} \sum_{m=1}^{M} R_{m,p}(\xi,\eta)\mathcal{H}(\mathbf{x_o},\mathbf{x})J(\xi,\eta)\, d\hat{\Gamma}^e \tag{2.39a}$$

$$\mathrm{G}^e(\mathbf{x_o}) = \int_{\hat{\Gamma}^e} \sum_{m=1}^{M} R_{m,p}(\xi,\eta)\mathcal{G}(\mathbf{x_o},\mathbf{x})J(\xi,\eta)\, d\hat{\Gamma}^e \tag{2.39b}$$

where M is the number of non-zero basis functions at the given parametric coordinate, (ξ,η) [57]. The Jacobian of transformation is:

$$J(\xi,\eta) = \left|\frac{\partial \mathbf{x}}{\partial \xi} \times \frac{\partial \mathbf{x}}{\partial \eta}\right| \tag{2.40}$$

The Gauss-Legendre quadrature is applied to calculate regular integrals. Singularities in single and double-layer potentials require special attention in 3D. Although there are singularity subtraction techniques for interfacial dynamics problems [58], special quadratures should be employed for potential or viscous flow problems. Hence, the techniques used to alleviate singularities are explained later in the benchmark problems. The calculations might get affected by near-singularities when the source point lies close to but not inside the surface element. The adaptive integral method proposed by Gong *et al.* [59] is employed to identify and calculate near-singular integrals. The smallest distance between the surface

Table 2.1: The integration results of Eq. (2.42) obtained by different techniques

Parameters			Standard	Adaptive Subdivision	Exact Results
ρ	θ	δ			
1.0	$\pi/4$	3.000	0.13245704	0.13245704	0.13245704
1.0	$\pi/4$	2.800	0.14915423	0.14915423	0.14915423
1.0	$\pi/4$	2.600	0.17094271	0.17094271	0.17094272
1.0	$\pi/4$	2.400	0.20073733	0.20073733	0.20073774
1.0	$\pi/4$	2.200	0.24470265	0.24470265	0.24474870
1.0	$\pi/4$	2.100	0.27639831	0.27705677	0.27705794
1.0	$\pi/4$	2.010	0.31295413	0.32201384	0.32201384
1.0	$\pi/4$	2.001	0.31687109	0.32979370	0.32979372

element and the source point and the element lengths in both parametric directions are calculated to determine the Gauss order required to obtain a prescribed accuracy. A subdivision is performed on the element if the distance results in a Gauss order higher than the prescribed level. The required Gauss order, m_i, to calculate the integral with a prescribed relative error, e_i (upper bound), can be determined by [59]:

$$m_i = -0.1\sqrt{\frac{2}{3}p + \frac{2}{5}}\ \ln\left(\frac{e_i}{2}\right)\left[\left(\frac{8L_i}{3R}\right)^{3/4} + 1\right] \tag{2.41}$$

where p is the order of the singularity. R and L_i represent the smallest distance between the source point and element and the element length in the i^{th} parametric direction. Note that, the same relative error is used in each parametric direction, and the maximum value of m_i is considered for the calculations. Two benchmark integrals [59] on isogeometric surfaces are calculated here to test the proposed quadrature scheme. The first integral is given as:

$$I_\Gamma = \int_\Gamma \frac{1}{4\pi r} d\Gamma \tag{2.42}$$

where r denotes the Eulerian distance between the source and field points. The integral has a singularity of order 1 and it is calculated on a 90° cylindrical surface as shown in Fig. 2.10A. The radius, R_{cyl}, and the length of the cylinder, L, are 1 and 2, respectively. The position of the source point, shown by the red marker

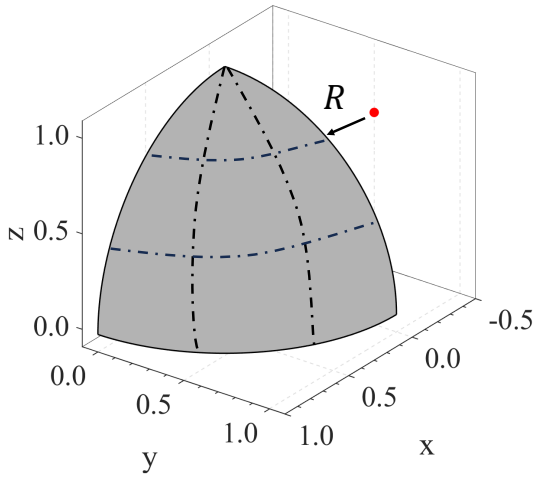

Figure 2.11: The schematic of the problem with subdivision of the degenerated NURBS element

in Fig. 2.10B, is determined by 3 parameters: ρ, θ, and δ. The x, y, z coordinates of the source point can be defined as:

$$x = \rho \cos\theta \tag{2.43a}$$

$$y = \delta \tag{2.43b}$$

$$z = \rho \sin\theta \tag{2.43c}$$

If the predetermined Gauss order is not enough to produce results with the desired accuracy, because of the large L_i/R ratio, the subdivision is applied on the isogeometric element as shown in Fig. 2.10B. The subdivision level is determined by `ceil`(L_i/L_{req}) where L_{req} is the element length required to obtain results with a relative error of e_i. The results obtained by employing standard Gauss quadrature and adaptive integration are tabulated in Tab. 2.1. The relative error employed in the study is $e_i = 10^{-4}$ with ten Gauss points (order) in both directions. It can be seen from Tab. 2.1 that the adaptive subdivision algorithm is activated after the $\delta = 2.1$ value which results in higher errors than the prescribed tolerance if the adaptive quadrature is not employed.

The algorithm works also for degenerated elements where one boundary of the element shrinks down to a point which is the case for spherical surfaces represented by NURBS. An example geometry is shown in Fig. 2.11 with a subdivision

Table 2.2: The integration results of Eq. (2.44) obtained by different techniques

Parameters			Standard	Adaptive Subdivision	Exact Results
ρ	θ	ϕ			
1.0	$24\pi/32$	$\pi/4$	0.11647824	0.11647824	0.11647795
1.0	$23\pi/32$	$\pi/4$	0.12947313	0.12947313	0.12947206
1.0	$22\pi/32$	$\pi/4$	0.14547727	0.14547727	0.14547517
1.0	$21\pi/32$	$\pi/4$	0.16567295	0.16567295	0.16567895
1.0	$20\pi/32$	$\pi/4$	0.19197182	0.19207505	0.19207504
1.0	$19\pi/32$	$\pi/4$	0.22753990	0.22839105	0.22839107
1.0	$18\pi/32$	$\pi/4$	0.27688449	0.28299932	0.28300340
1.0	$17\pi/32$	$\pi/4$	0.33875882	0.38272197	0.38274752

level of 3. The following integral is calculated on one octant of a sphere where control points and weights of the sphere are available in [59]:

$$I_\Gamma = \int_\Gamma \frac{r_{,x}}{4\pi r^2} d\Gamma \tag{2.44}$$

The location of the source point is defined in spherical coordinates:

$$x = \rho \cos\theta \sin\phi \tag{2.45a}$$

$$y = r \cos\theta \sin\phi \tag{2.45b}$$

$$z = \rho \cos\phi \tag{2.45c}$$

The results obtained by $m_i = 10$ and e_i are tabulated in Tab. 2.2 where the algorithm is activated after $\theta = 20\pi/32$.

Polar coordinate transformation is a quadrature technique used to calculate weakly-singular integrals where the integration is transformed into a vertex singularity problem [60]. Employing polar coordinates introduces a Jacobian which cancels out the weakly-singular part of the integral. An example integral [61] is shown below where a weakly-singular integral on a rectangular element is calculated:

$$I_\Gamma = \int_0^4 \int_{-1}^1 \frac{r_{,x}}{r} dxdy \tag{2.46}$$

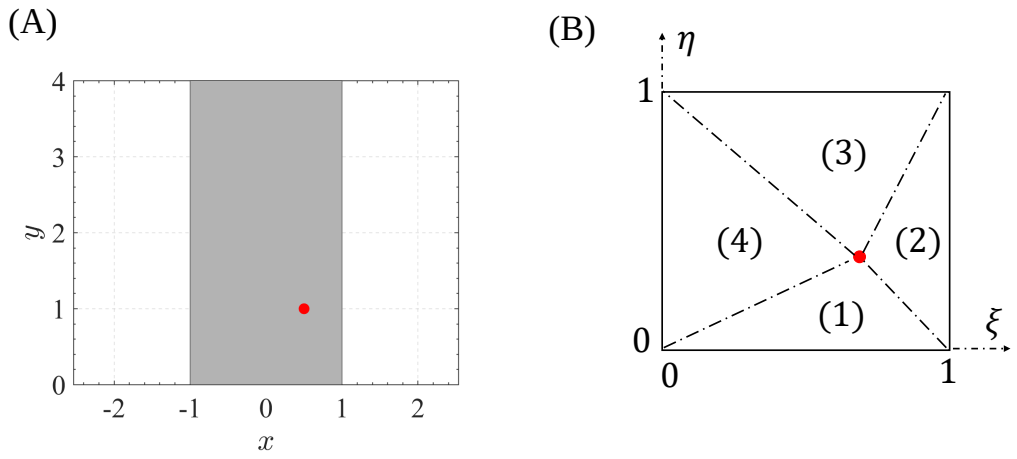

Figure 2.12: (A) The schematic representation of the problem (B) Subdivision of the element into triangles in parameter space

The computational domain and an example source point are shown in Fig. 2.12A. First, triangulation is performed in the parameter space which converts it into a vertex-singularity problem. Then polar-coordinate transformation is employed to introduce the desired Jacobian. The results of the integral shown in Eq. (2.46) with 3 different source points (with their parametric coordinates) are tabulated below where exact results are taken from [61]. The relative error is defined as $|I_{exact} - I_{polar}|/|I_{exact}|$ and the number of Gauss points employed is 20.

2.2.1 Benchmark II: Three-dimensional Heat Transfer

A 3D heat conduction problem in a hollow sphere is analyzed where radiative and convective boundary condition, Eq. (2.26), is defined on the outer surface as shown in Fig. 2.13A. Uniform temperature, ϕ_i, is defined at the inner surface. The boundary integral equation for the corresponding problem can be described as in Eq. (2.21). The free space Green's function in 3D is given by [58]:

$$\mathcal{G}(\mathbf{x_o}, \mathbf{x}) = \frac{1}{4\pi|\mathbf{r}|} \tag{2.47}$$

Table 2.3: The integration results of Eq. (2.46) obtained by polar-coordinate transformation

Parameters		**Coordinate Trans.**	**Exact Results**	**Relative Error**
ξ	η			
0.750	0.25	-2.057678	-2.057701	1.12×10^{-5}
0.875	0.25	-3.129290	-3.129422	4.22×10^{-5}
0.995	0.25	-4.203397	-4.203354	1.02×10^{-5}

which exhibits singularity as the field point, $\mathbf{x}$, approaches the source point, $\mathbf{x}_o$. It has been shown in [62] that both layer potentials for 3D Stokes flow are weakly singular which is also the case for Laplace equation. Hence, polar-coordinate transformation [60] is applied to calculate both singular integrals. The surfaces of the computational domain are generated by using single-patch quadratic NURBS which exactly represent a sphere. NURBS elements on a sphere are demonstrated in Fig. 2.13B. Collocation points are demonstrated with red markers where their positions on an element might change depending on their parametric locations. Degenerated elements are present around the poles of the sphere. Quadrature techniques employed in this study do not require any additional treatment for the degenerated elements except for the augmentation of Gauss points with increasing refinement levels. This augmentation is performed to achieve optimal convergence since the polar-coordinate transformation technique is sensitive to element shapes. For large refinement levels, the aspect ratio of the degenerated elements increases which causes inaccuracies in the calculation of layer potentials. Duffy-distance transformation is proposed to overcome this problem [60] which is not applied in this study. The variable condensation technique is applied to perform Newton-Raphson iterations only on the outer surface. The same error tolerance, $\|\bar{\mathbf{e}}\|_2 = 10^{-10}$, is employed to terminate iterations. The non-dimensional parameters introduced in the 2D case are also employed here. The general solution to the 1D heat equation in spherical coordinates is given by:

$$\bar{\phi}(\bar{r}) = \frac{C_1}{\bar{r}} + C_2 \tag{2.48}$$

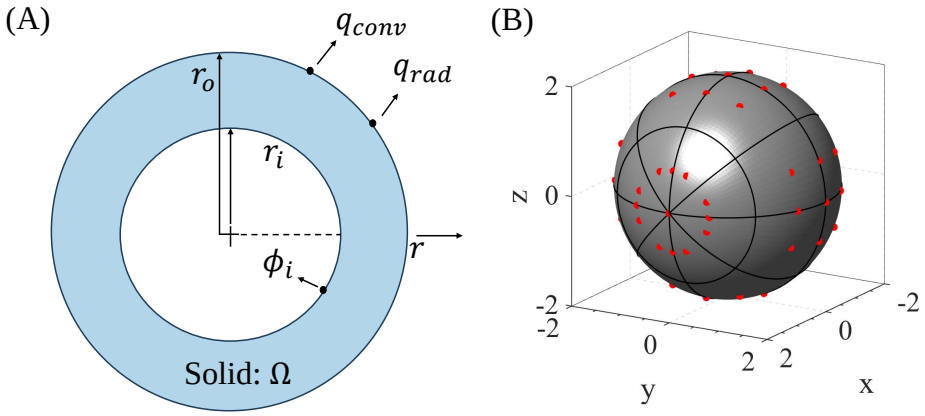

Figure 2.13: (A) Schematic of the problem, and (B) the surfaces of the computational domain generated by quadratic NURBS

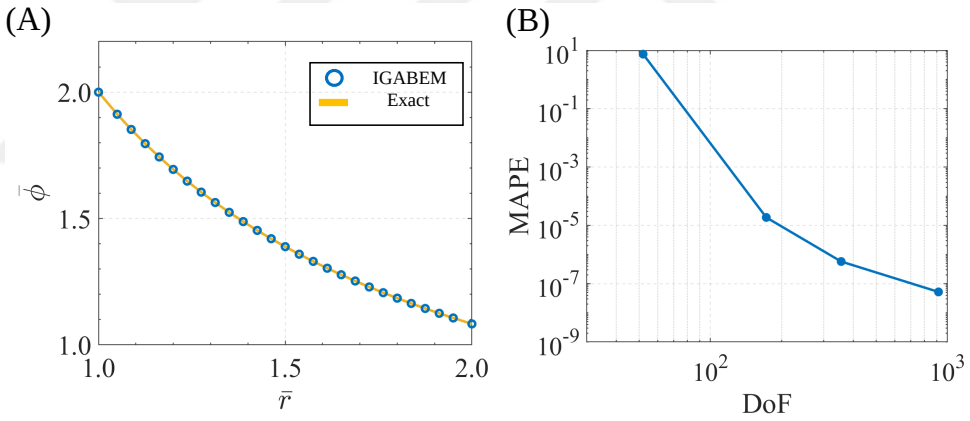

Figure 2.14: (A) Temperature distribution, and (B) the convergence of the proposed formulation

where C_2 can be expressed as a function of C_1 and C_1 can be determined from the heat flux expression on the outer surface. The quartic equation should be solved to determine C_1 which can be performed analytically. The parameters used in this problem are: $\bar{\phi}_i = 2, \bar{r}_o = 2, Bi = 1$ and $Bi_r = 1$. The temperature distribution of 25 equally spaced post-process points is demonstrated in Fig. 2.14A where the total number of collocation points to calculate demonstrated temperatures is 172. The coordinates of post-process points are $(x, y, z) \in [0, 0, 1.05 - 1.95]$. The convergence study for the temperature values at the post-process points is demonstrated in Fig. 2.14B.

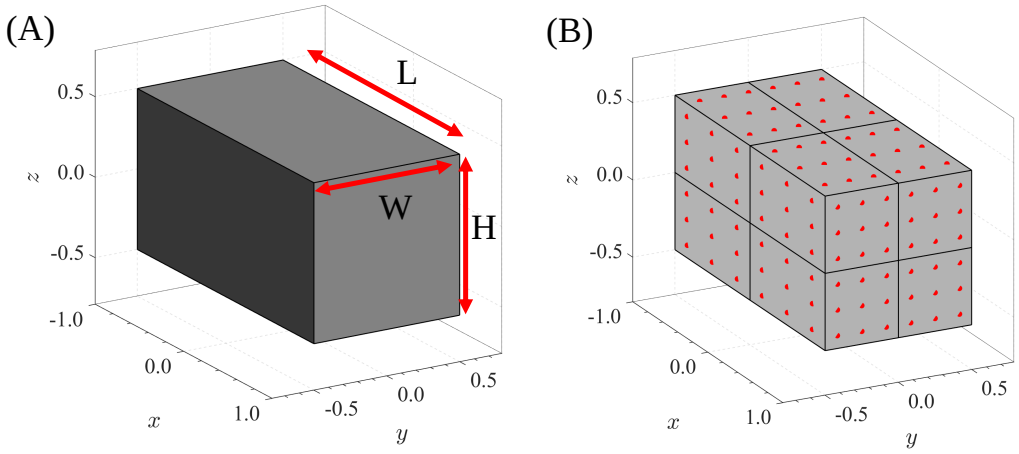

Figure 2.15: (A) Schematic of the problem domain. (B) NURBS elements and collocation points

2.2.2 Benchmark III: Three-dimensional Stokes Flow

Stokes flow in a microchannel constitutes the basis for many microfluidic applications. Hence, it is important to capture the flow properties like stresses or velocities with high accuracy. In this problem, Stokes flow in a square microchannel, Fig. 2.15A, is analyzed where an analytical solution was derived by using Finite Fourier Transform [63]. The governing equations for the momentum and mass balance can be expressed as:

$$-\nabla P + \mu \nabla^2 \mathbf{u} = 0 \tag{2.49a}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.49b}$$

where $\mathbf{u}$ and μ represent the velocity field and fluid viscosity with P denoting the modified pressure which is used to convert the equation into a homogeneous form instead of fluid pressure p. The modified pressure expression is:

$$P = p - \rho \mathbf{g} \cdot \mathbf{x} \tag{2.50}$$

where ρ and $\mathbf{g}$ are the fluid density and acceleration of gravity. The boundary integral equation for the 3D Stokes flow can be expressed as:

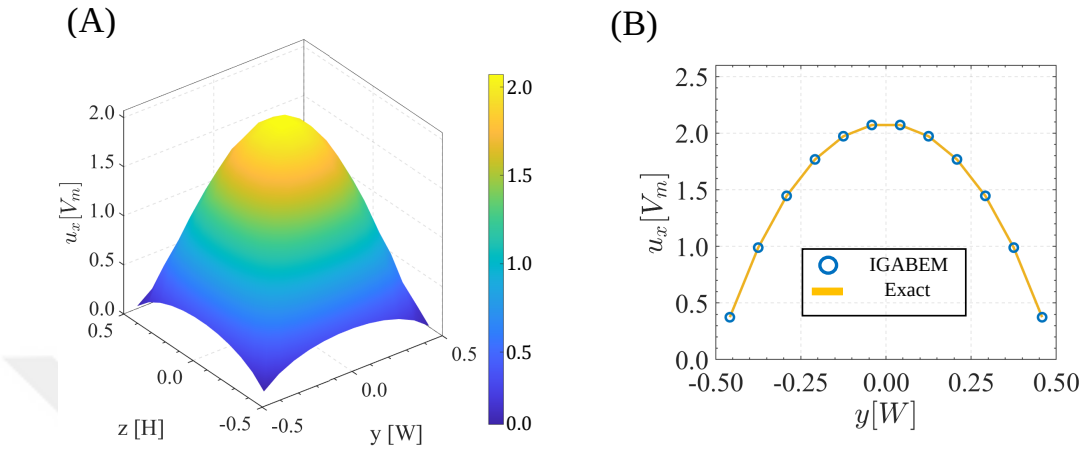

Figure 2.16: (A) Schematic of the problem domain, and (B) the elements and collocation points

$$\mathbf{u}(\mathbf{x_o}) = -\frac{1}{4\pi\mu}\int_\Gamma \boldsymbol{\mathcal{G}}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{f}(\mathbf{x})d\Gamma + \frac{1}{4\pi}\int_\Gamma \mathbf{u}(\mathbf{x})\cdot\boldsymbol{\mathcal{T}}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{n}(\mathbf{x})d\Gamma \tag{2.51}$$

where $\mathcal{G}$ and $\mathcal{T}$ are Stokeslet and stresslet, respectively. $\mathbf{f}(\mathbf{x})$ denotes the traction and the normal vector, $\mathbf{n}(\mathbf{x})$, points towards the fluid. Stokeslet and stresslet can be expressed as [58]:

$$\mathcal{G}_{ij}(\mathbf{x},\mathbf{x_o}) = \frac{\delta_{ij}}{|\mathbf{r}|} + \frac{r_i r_j}{|\mathbf{r}|^3} \tag{2.52a}$$

$$\mathcal{T}_{ijk}(\mathbf{x},\mathbf{x_o}) = -6\frac{r_i r_j r_k}{|\mathbf{r}|^5} \tag{2.52b}$$

where r_i is the i^{th} component of the position vector between the field and source points with $i = 1, 2, 3$. The surfaces of the computational domain are generated in SolidWorks® which is then saved as an IGES file. Information about different patches can be read directly from the IGES file by searching for B-spline surface entities. The scale and sizes of the patches can be modified by changing their control points which enables analyzing different microchannel geometries in a unified way. The model is analysis-ready after determining element ranges in parameter space and collocation points with their connectivities, once the geometry information is imported into MATLAB. This feature of employing NURBS allows us to work with more complex channel geometries. Using Greville's abscissae to determine the coordinates of collocation points in parameter space results in points

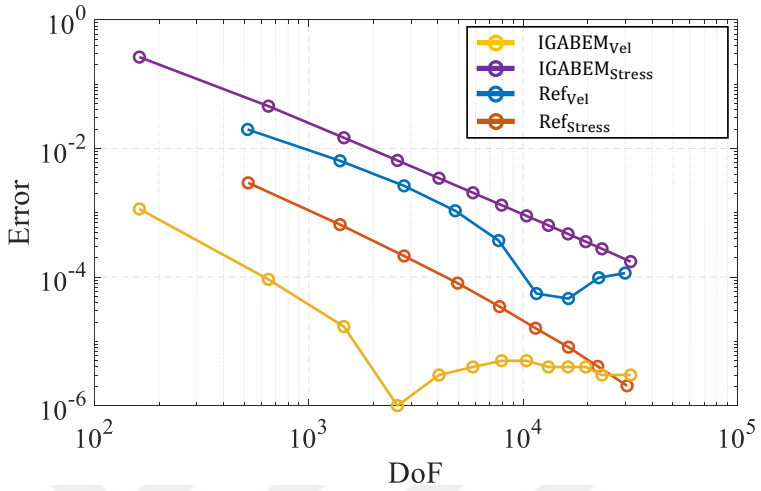

Figure 2.17: The convergence of the solver with reference values taken from [1]

lying on the edges of the patches which is undesired since special treatments are required at corners [64]. The discontinuous collocation scheme proposed by Marussig *et al.* [65] is employed here to apply an offset to collocation points such that they lie inside elements so that no corner treatment is required. The polar-coordinate transformation is employed to calculate singular integrals appearing in single and double-layer potentials with ten Gauss points in each direction. Regular and near-singular integrals are also calculated with ten Gauss points in each direction. Relative error tolerance used to calculate near-singular integrals is $e_i = 10^{-4}$ with a subdivision limit of 20. The subdivision limit is employed to prevent cases where subdivision takes too much time for that collocation point and element pair.

The problem domain, shown in Fig. 2.15A, is a channel with height H, width W, and length L. The limits of the computational domain are $x \in \left[\frac{-L}{2}, \frac{L}{2}\right]$, $y \in \left[-\frac{W}{2}, \frac{W}{2}\right]$ and $z \in \left[\frac{-H}{2}, \frac{H}{2}\right]$. W is used to non-dimensionalize the spatial coordinates. No slip boundary condition is defined on the side walls and pressure changes only in $x-$direction. The downstream velocity normalized with the mean velocity can be expressed as [63]:

$$u_x(y,z) = \frac{\pi^2}{4} \sum_{n=1,3,5,\ldots}^{\infty} \sum_{m=1,3,5,\ldots}^{\infty} \sin\left(\frac{n\pi y}{W} + \frac{n\pi}{2}\right) \sin\left(\frac{m\pi z}{W} + \frac{m\pi}{2}\right) \tag{2.53}$$

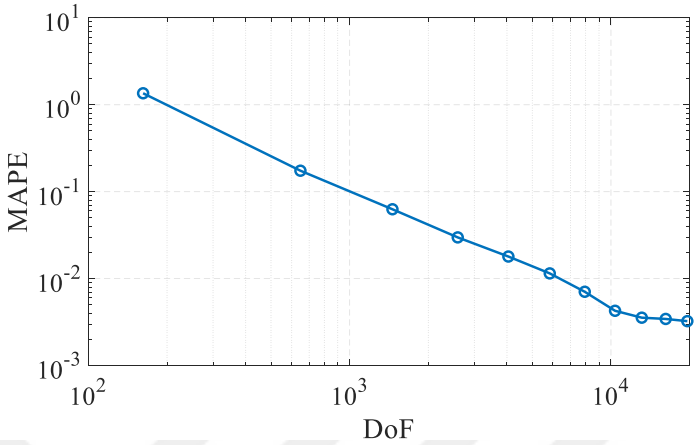

Figure 2.18: MAPE values of the pressure gradient with respect to DoF

For the benchmark problem, W and H are taken as 1 and the length of the channel is $L = 2$. $\beta = W/H$ denotes the aspect ratio of the channel. The mean velocity defined at the inlet is $V_m = 1.0$ with pressure defined as zero at the outlet. Inlet velocity is expressed by using Eq. (2.53) with hundred terms for both n and m.

The velocity profile at the outlet obtained by employing 846 collocation points is shown in Fig. 2.16A. Down-stream velocities calculated at the outlet $(x, z) = [1, -0.047]$ for varying y values are shown in Fig. 2.16B with the exact solution. The following integrals are used to quantify the accuracy of the proposed formulation:

$$\int_{\Gamma} \mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma = 0 \tag{2.54a}$$

$$\int_{\Gamma} \mathbf{f}(\mathbf{x}) d\Gamma = 0 \tag{2.54b}$$

The values of the integrals expressed in Eq. (2.54a) and Eq. (2.54b) with respect to DoF of the problem (which is three times the number of collocation points) are shown in Fig. 2.17. The solutions available in [1], denoted by Ref, are also shown in the figure. The accuracy in terms of the velocity calculation, Eq. (2.54a), is superior to the results available in the literature. However, the calculation of Eq. (2.54b) results in errors larger than the reference values. This behavior is caused by the discontinuous collocation technique employed in this study. A finite value is obtained when comparing the traction values of collocation points

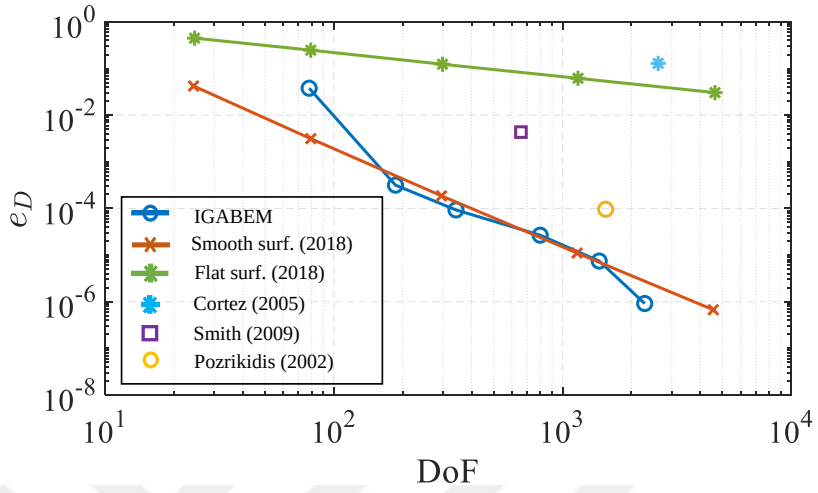

Figure 2.19: The convergence of the solver compared with results available in the literature

located on the inlet and outlet patches with those on the side patches. This is not the case for elements employed in [1] where there are nodes on the edges.

The saturation of the errors for $\text{IGABEM}_{\text{Vel}}$ for DoF values larger than 3000 is because of the relative error tolerance used in the nearly singular integral calculation and subdivision limit to accelerate calculations. Moreover, it is observed that the number of terms used to calculate Eq. (2.53) has an effect on the point where sudden decrease and increase occurs for the calculation of Eq. (2.54a) which doesn't change the results after using 100 terms for n and m in Eq. (2.53). MAPE values of the pressure gradient are shown in Fig. 2.18 where IGABEM values are calculated from the difference between inlet and outlet pressure values. The analytical description of the pressure gradient is given by [63]:

$$\frac{\partial p}{\partial x} = -\frac{\pi^6}{64\beta^2} \cdot \frac{\mu U}{WH} \cdot \left[\sum_{n=1,3,5,\ldots}^{\infty} \sum_{m=1,3,5,\ldots}^{\infty} \frac{1}{n^2 m^2 (\beta^2 n^2 + m^2)} \right]^{-1} \tag{2.55}$$

where U denotes the velocity scale and it is expressed as shown below:

$$U = \frac{W^2}{\mu} \cdot \left(-\frac{\partial p}{\partial x} \right) \tag{2.56}$$

The saturation behavior is observed at large DoF values due to the limitations defined in nearly-singular integral calculations.

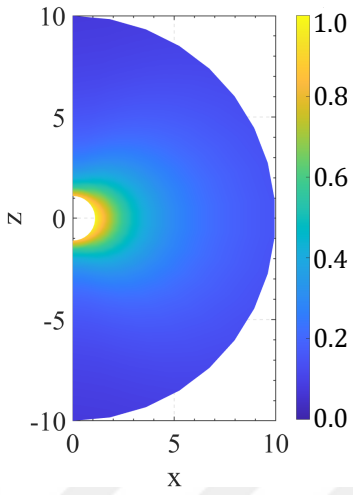

Figure 2.20: The magnitude of the velocity of the flow field around the sphere obtained by using 482 collocation points

2.2.3 Benchmark IV: Moving Sphere in Stokes Flow

The final validation case is the moving sphere in an otherwise quiescent fluid. The analytical expression for the drag force on the sphere is used to quantify the accuracy. For a sphere with radius R, the drag force in a fluid with viscosity μ can be expressed as [66]:

$$\mathbf{F} = \int_{\Gamma} \mathbf{f}(\mathbf{x}) d\Gamma = -6\pi\mu R\mathbf{U} \tag{2.57}$$

The spherical surface with a unit radius is represented by quadratic NURBS elements as shown in Fig. 2.13B. A constant velocity, $\mathbf{u}_{sp} = \mathbf{U} = [1, 0, 0]$, is defined on the sphere. The flow field tends to approach 0 at infinity which is satisfied by the nature of free-space Green's functions. The boundary integral equation used to simulate the problem is described in Eq. (2.51). The polar-coordinate transformation is utilized to calculate singular integrals with 3 Gauss points in each direction. The same number of Gauss points is also used to calculate regular and near-singular integrals with a relative error tolerance of $e_i = 10^{-8}$ with no subdivision limit. The relative error of the drag force, $e_D = |\mathbf{F}_{exact} - \mathbf{F}_{BEM}|/|\mathbf{F}_{exact}|$, with respect to the number of collocation points is shown in Fig. 2.19 with results available in the literature [1].

Flat and smooth surface representations [1] where bilinear and higher-order

interpolation schemes are included for comparison. The accuracy of IGABEM matches with higher-order interpolation schemes which require additional steps to implement. The magnitude of the velocity of the flow field around the moving sphere is shown in Fig. 2.20. 482 collocation points are employed in the simulation. The velocity of 360 different points has been calculated for the visualization. The flow-field speed approaches 1 as it gets closer to the sphere and converges to 0 as $\mathbf{x} \to \infty$, as expected.

Chapter 3

Two-Dimensional Droplet Motion

In this section, 2D droplet motions in both bounded and unbounded flows are investigated. The isogeometric discretization of the boundary integral equation for 2D droplet motion is explained. The volume correction algorithm is introduced here to keep the volume (inner area of the droplet) constant which deviates from the initial value due to the numerical inaccuracies. The mesh relaxation algorithm is demonstrated to maintain the mesh quality. The algorithms are tested in unbounded flow problems which is then applied to a two-particle interaction between parallel plate problem.

3.1 Boundary Element Formulation

For a point residing on the boundaries of the deformable object, Γ_0, the boundary integral equation (BIE) of an unbounded Stokes flow with an interface in 2D can be expressed as [1]:

$$2\pi(1+\lambda)\mathbf{u}(\mathbf{x_o}) = -\frac{1}{\mu_1}\int_{\Gamma_0} \mathcal{G}(\mathbf{x},\mathbf{x_o})\cdot\boldsymbol{\Delta}\mathbf{f}(\mathbf{x})d\Gamma + (1-\lambda)\int_{\Gamma_0}\mathbf{u}(\mathbf{x})\cdot\mathcal{T}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{n}(\mathbf{x})d\Gamma \tag{3.1}$$

where $\mathbf{x}$ and $\mathbf{x_o}$ represent the field and source points, respectively. $\mathbf{u}(\mathbf{x})$ is the velocity field with $\mathbf{n}(\mathbf{x})$ being the unit normal vector pointing towards the external fluid. The surface tension causes a pressure drop across the interface if the curvature is present, and this stress discontinuity is denoted by $\Delta\mathbf{f}(\mathbf{x})$ in the equation. $\lambda = \mu_0 \backslash \mu_1$ is the ratio between the viscosities of internal and external fluids with μ_1 taken as unity. $\mathcal{G}(\mathbf{x}, \mathbf{x_o})$ and $\mathcal{T}(\mathbf{x}, \mathbf{x_o})$ denote *Stokeslet* and *stresslet*, respectively. The two-dimensional form of these fundamental solutions are [58]:

$$\mathcal{G}_{ij}(\mathbf{x}, \mathbf{x_o}) = -\delta_{ij} \ln |\mathbf{r}| + \frac{r_i r_j}{|\mathbf{r}|^2} \tag{3.2a}$$

$$\mathcal{T}_{ijk}(\mathbf{x}, \mathbf{x_o}) = -4\frac{r_i r_j r_k}{|\mathbf{r}|^4} \tag{3.2b}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{x}_o$ and $|\mathbf{r}|$ denotes the Eulerian distance between the field and source points. The second term on the right-hand side of Eq. (3.1) (double-layer potential) is non-singular in 2D and it is calculated by employing Gauss-Legendre quadrature which is also the case for all regular integrals. On the other hand, the first term (single-layer potential) exhibits logarithmic singularity and it requires special attention. The stress discontinuity takes the following form for a droplet [1]:

$$\Delta\mathbf{f}(\mathbf{x}) = \frac{2}{Ca}\kappa(\mathbf{x})\mathbf{n}(\mathbf{x}) \tag{3.3}$$

where Ca denotes the Capillary number and $\kappa(\mathbf{x})$ represents the mean curvature. The singularity subtraction technique is employed here to alleviate weak singularities present in the single-layer potential [36]. An external flow field can be defined as shown below in the absence of boundaries other than the droplet [67]:

$$\begin{aligned} 2\pi(1+\lambda)\mathbf{u}(\mathbf{x_o}) = 4\pi\mathbf{u_0^\infty}(\mathbf{x}) - \frac{1}{\mu_1}\int_{\Gamma_0} \mathcal{G}(\mathbf{x}, \mathbf{x_o}) \cdot \mathbf{\Delta f}(\mathbf{x}) d\Gamma + \\ (1-\lambda)\int_{\Gamma_0} \mathbf{u}(\mathbf{x}) \cdot \mathcal{T}(\mathbf{x}, \mathbf{x_o}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \end{aligned} \tag{3.4}$$

The following BIE can be utilized in the presence of solid boundaries, Γ_1:

$$\begin{aligned} \alpha(\mathbf{x_o})\mathbf{u}(\mathbf{x_o}) = &-\frac{1}{\mu_1}\int_{\Gamma_1} \mathcal{G}(\mathbf{x}, \mathbf{x_o}) \cdot \mathbf{f}(\mathbf{x}) d\Gamma + \int_{\Gamma_1} \mathbf{u}(\mathbf{x}) \cdot \mathcal{T}(\mathbf{x}, \mathbf{x_o}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \\ &-\frac{1}{\mu_1}\int_{\Gamma_0} \mathcal{G}(\mathbf{x}, \mathbf{x_o}) \cdot \Delta\mathbf{f}(\mathbf{x}) d\Gamma + (1-\lambda)\int_{\Gamma_0} \mathbf{u}(\mathbf{x}) \cdot \mathcal{T}(\mathbf{x}, \mathbf{x_o}) \cdot \mathbf{n}(\mathbf{x}) d\Gamma \end{aligned} \tag{3.5}$$

where

$$\alpha(\mathbf{x_o}) = \begin{cases} 2\pi, & \mathbf{x_o} \in \Gamma_1 \\ 2\pi(1+\lambda), & \mathbf{x_o} \in \Gamma_0 \end{cases} \tag{3.6}$$

The single-layer potential on Γ_1 exhibits logarithmic singularity and it is alleviated by Telles transformation. The geometry and field variables (velocity and traction field) are discretized by employing NURBS functions [45]:

$$\mathbf{x}(\xi) = \sum_{m=1}^{n} R_{m,p}(\xi)\mathbf{P}_m \tag{3.7a}$$

$$\mathbf{u}(\xi) = \sum_{m=1}^{n} R_{m,p}(\xi)\mathbf{U}_m \tag{3.7b}$$

$$\mathbf{f}(\xi) = \sum_{m=1}^{n} R_{m,p}(\xi)\mathbf{F}_m \tag{3.7c}$$

where $\mathbf{P}$ denotes the control points in the physical space and ξ is the coordinate in parameter space. R is the p^{th} order NURBS basis function with n representing the number of non-zero basis functions at ξ. Quadratic, $p = 2$, NURBS elements are used in this study so p will not be shown explicitly in the following formulations. Knot vector and control point selection for approximating a circle is detailed in [49]. $\mathbf{U}$ and $\mathbf{F}$ are the control point velocity and tractions which when combined with NURBS basis functions are mapped to the physical counterparts $\mathbf{u}$ and $\mathbf{f}$, respectively. The discretized version of Eq. (3.5) is shown below:

$$\begin{aligned} \alpha(\mathbf{x_o}) \sum_{m=1}^{n} R_m \mathbf{U}_m(\xi_o) = &- \frac{1}{\mu_1} \sum_{e=1}^{N_1^e} \mathbf{B}_e(\mathbf{x}, \mathbf{x_o}) + \sum_{e=1}^{N_1^e} \mathbf{C}_e(\mathbf{x}, \mathbf{x_o}) \\ &- \frac{1}{\mu_1} \sum_{e=1}^{N_0^e} \mathbf{D}_e(\mathbf{x}, \mathbf{x_o}) + (1-\lambda) \sum_{e=1}^{N_0^e} \mathbf{E}_e(\mathbf{x}, \mathbf{x_o}) \end{aligned} \tag{3.8}$$

where

$$\mathbf{B}_e(\mathbf{x}, \mathbf{x_o}) = \int_{\hat{\Gamma}_1^e} \mathcal{G}(\mathbf{x}, \mathbf{x_o}) \cdot \sum_{m=1}^{n} R_m \mathbf{F}_m(\xi) d\hat{\Gamma} \tag{3.9a}$$

$$\mathbf{C}_e(\mathbf{x}, \mathbf{x_o}) = \int_{\hat{\Gamma}_1^e} \sum_{m=1}^{n} R_m \mathbf{U}_m(\xi) \cdot \mathcal{T}(\mathbf{x}, \mathbf{x_o}) \cdot \mathbf{n}(\mathbf{x}) d\hat{\Gamma} \tag{3.9b}$$

$$\mathbf{D}_e(\mathbf{x}, \mathbf{x_o}) = \int_{\hat{\Gamma}_0^e} \mathcal{G}(\mathbf{x}, \mathbf{x_o}) \cdot \Delta\mathbf{f}(\mathbf{x}) d\hat{\Gamma} \tag{3.9c}$$

$$\mathbf{E}_e(\mathbf{x}, \mathbf{x_o}) = \int_{\hat{\Gamma}_0^e} \sum_{m=1}^{n} R_m \mathbf{U}_m(\xi) \cdot \mathcal{T}(\mathbf{x}, \mathbf{x_o}) \cdot \mathbf{n}(\mathbf{x}) d\hat{\Gamma} \tag{3.9d}$$

N_0^e and N_1^e denote the number of elements on the particle and channel boundaries, respectively. Collocating Eq. (3.8) for each source point yields the following system of equations:

$$\begin{bmatrix} \mathbf{A}_{00} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{11} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{01} \\ \mathbf{B}_{11} \end{bmatrix} \cdot \mathbf{F}_1 + \begin{bmatrix} \mathbf{C}_{01} \\ \mathbf{C}_{11} \end{bmatrix} \cdot \mathbf{U}_1 + \begin{bmatrix} \mathbf{D}_{00} \\ \mathbf{D}_{10} \end{bmatrix} + \begin{bmatrix} \mathbf{E}_{00} \\ \mathbf{E}_{10} \end{bmatrix} \cdot \mathbf{U}_0 \tag{3.10}$$

Subscripts, 0 and 1 stand for the particle and channel boundaries, respectively. The matrices shown above are then put in such a form that the resultant set of equations yields the velocities of collocation points on the deformable object in addition to the velocity or traction field on the channel boundaries depending on the type of boundary conditions. Once the control point velocities are determined on the particle, the interface geometry is updated:

$$\frac{d\mathbf{P}_i(t)}{dt} = \mathbf{U}_i(t) \tag{3.11}$$

An explicit Euler scheme is utilized for the temporal discretization. Note that the rate of deformation of the interface is determined by the normal component of the velocities defined on the interface [32]. Hence, control point velocities are mapped to the collocation points on the interface where their normal components are evaluated as shown below:

$$\mathbf{u}_n(\mathbf{x}) = [\mathbf{u}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})] \, \mathbf{n}(\mathbf{x}) \tag{3.12}$$

Then, these calculated velocities are mapped back to the control points to modify their positions. Numerical inaccuracies in the evaluation of layer potentials lead to changes in the volume (the area inside the boundary in 2D) of the particle which is undesirable, especially for long-time horizon simulations. One solution to alleviate this problem is to apply a volume correction algorithm [31]. This algorithm modifies the position of control points such that the volume is kept within a prescribed tolerance. Positions of control points are modified, $\mathbf{X}_i$ by solving a non-linear optimization problem:

$$\arg\min_{V=V_0} \|\mathbf{X}_i - \mathbf{P}_i\| \tag{3.13}$$

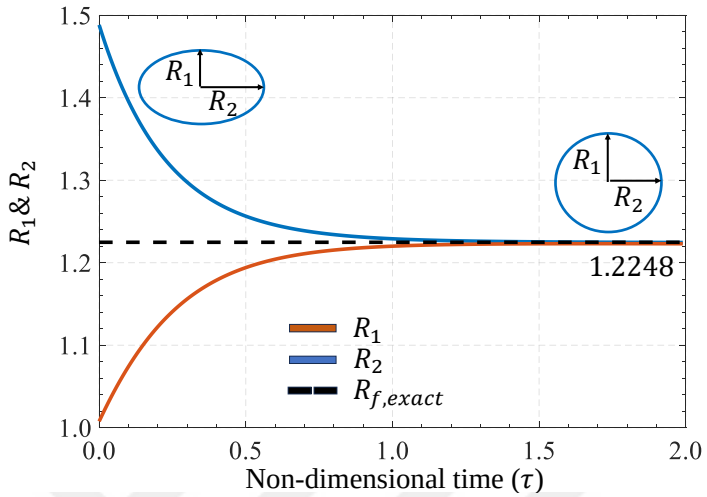

Figure 3.1: Time evolution of the particle major and minor axes

$i = 1, ..., N$ where N is the total number of collocation points and V_0 is the initial volume. Sequential quadratic programming is employed for the optimization problem. The mesh relaxation algorithm proposed by [43] for triangular elements is employed here for 2D NURBS elements to preserve the mesh quality. The collocation points are modified with the non-physical tangential velocity field shown below:

$$\mathbf{w}(\mathbf{x}_i) = \frac{(N_0^e)^{3/2}}{1+\lambda}[\mathbf{I} - \mathbf{n}(\mathbf{x}_i)\mathbf{n}(\mathbf{x}_i)] \cdot \sum_{j=1}(C_{r1} + C_{r2}|\kappa_j(\mathbf{x})|^{3/2})S_j^e(\mathbf{x}_j - \mathbf{x}_i) \quad (3.14)$$

where C_{r1} and C_{r2} are two control parameters for the correction where C_{r1} enforces element lengths to remain as the same C_{r2} is important for clustering collocation points to zones where the curvature is high. S_j^e is the element length between collocation points $\mathbf{x}_i$ and $\mathbf{x}_j$. Two important integral expressions are frequently calculated to determine the length and inner area of the droplet. To calculate the length of the boundary of the droplet, the following integral is used:

$$L_d = \int_\Gamma d\Gamma = \int_{\hat{\Gamma}} \sqrt{x_\xi^2 + y_\xi^2}\, d\hat{\Gamma} \quad (3.15)$$

The inner area can be calculated by employing Green's theorem:

$$A_d = \iint_\Omega 1\, dx\, dy = \frac{1}{2}\int_{\hat{\Gamma}} (xy_\xi - yx_\xi)\, d\hat{\Gamma} \quad (3.16)$$

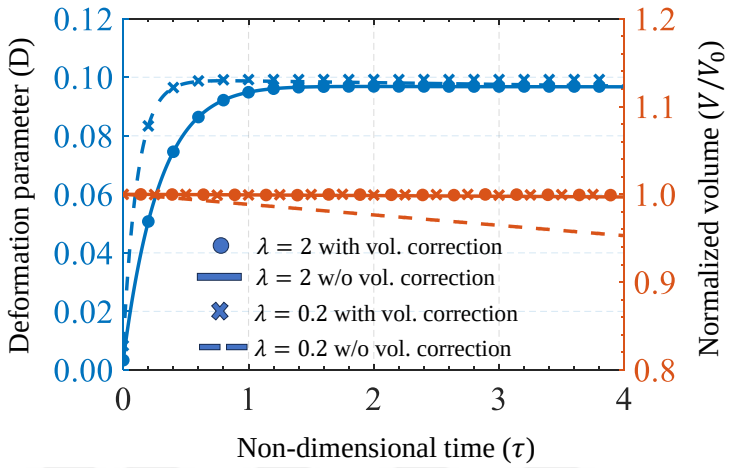

Figure 3.2: The effect of volume correction to Taylor deformation parameter and volume

3.2 Results and Discussion

The relaxation of a particle is analyzed as a benchmark problem where an initially elliptical particle is relaxed to a circle in a quiescent fluid. The initial inner area of the particle is 4.7126 so the length of both major and minor axes will converge to 1.2248. The simulation parameters are $Ca = 0.2, \lambda = 1, \Delta\tau = 0.01$ and $N = 32$. Fig. 3.1 shows the time evolution of the major and minor axes of the particle where the convergence towards the exact value is observed. The particle is assumed to reach the steady-state shape after $\tau = 2$. The errors for R_1 and R_2 are 0.1461% and 0.0203%, respectively.

The effect of volume correction is demonstrated in Fig. 3.2 where an initially circular particle is placed in a shear flow with the shear rate $\dot{\gamma} = 1$ by defining an appropriate external flow field. The simulation parameters for each case are $Ca = 0.2, \Delta\tau = 0.01$, and $N = 24$ with a unit radius for the initial geometry. Note that, non-dimensionalization for the time is achieved by $\tau = \dot{\gamma}t$. Two different viscosity ratio values are analyzed here: $\lambda = \{0.2, 2\}$. Fig. 3.2 shows the time evolution of the area enclosed by the particle surface, V, and Taylor deformation parameter, D. Taylor deformation parameter is used to quantify the amount of

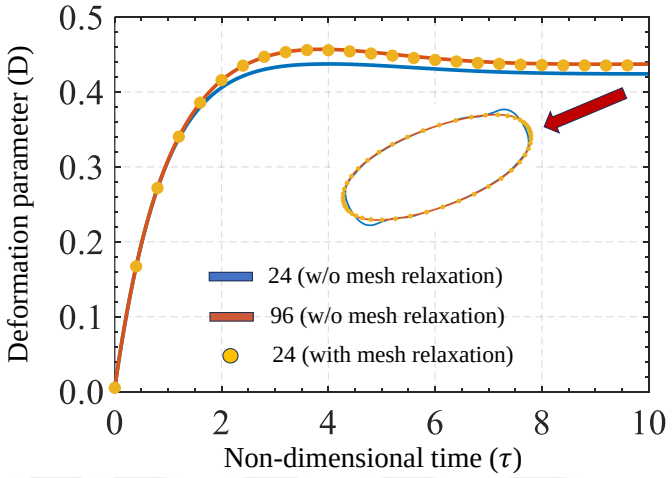

Figure 3.3: The effect of mesh relaxation in high-deformation regime

deformation which is calculated by the following expression [49]:

$$D = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \tag{3.17}$$

where L_{max} and L_{min} denote the maximum and minimum distances from the center of gravity of the particle, respectively. As can be seen from Fig. 3.2 the particle deforms up to a certain value and it achieves a steady shape around $D \approx 0.1$ for both cases. Particles with lower viscosity deform faster compared to high-viscosity cases. Change in the volume of the particle is severe for $\lambda = 0.2$. On the other hand, it is negligible for $\lambda = 2$. The results for applying the volume correction with 0.1% tolerance are also demonstrated in Fig. 3.2. The conservation of the volume is forced within the given tolerance by solving the optimization problem. The change in the particle volume also affects the Taylor parameter in the long run which is observable for $\lambda = 0.2$ case where there is a nonphysical change in D. Hence, the application of the volume correction algorithm preserves the incompressibility constraint which is important for low-viscosity ratio simulations without increasing the degree of freedom of the problem. Moreover, it also provides stable results in terms of maintaining accurate Taylor deformation parameter values.

The mesh relaxation algorithm is tested in the high-deformation regime $Ca = 1$ where using $N = 24$ resulted in nonphysical geometrical structures such as shown

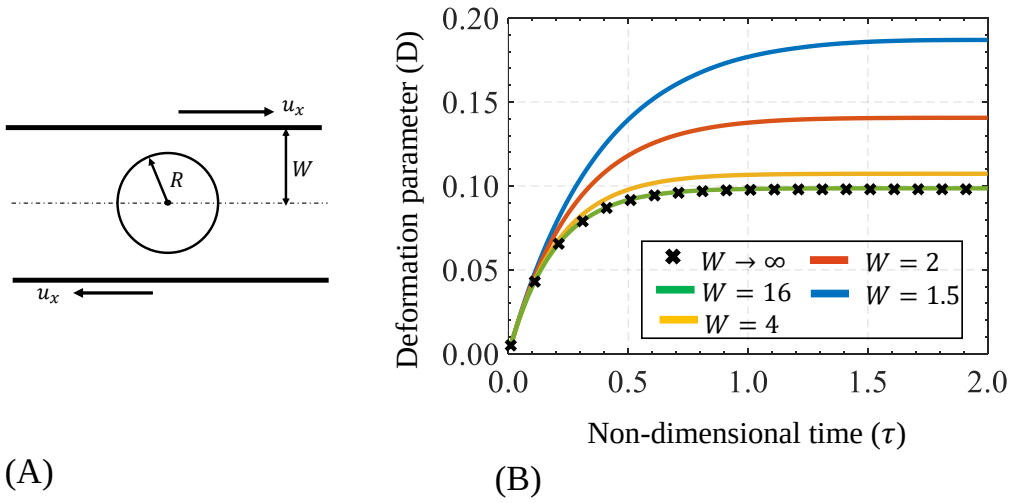

Figure 3.4: (A) Schematic representation of the benchmark problem and (B) Taylor deformation parameter for different channel widths

with the blue line in Fig. 3.3. The simulation parameters are $\lambda = 1$, $\Delta\tau = 0.01$. C_{r1} and C_{r2} are chosen as 0.2 after experimenting with different values. 10 virtual steps are applied between each physical time step. Note that the volume correction is employed for each case. $N = 96$ case is utilized as the ground truth solution. In addition to the geometrical errors, distortion of elements results in inaccurate estimation of the Taylor deformation parameter. The application of the mesh relaxation algorithm not only preserves the mesh quality but also increases the accuracy of estimating the deformation as shown in Fig. 3.3. In this case, it is important to cluster the collocation points around high curvature zones which is accomplished with the mesh relaxation algorithm. Depending on the need, equal element size can also be forced by choosing appropriate C_{r1} and C_{r2} values. Note that the results presented so far focuses on droplets in an unbounded flow where single and double-layer potentials for the channel boundaries are not present in the formulation.

The interaction between two droplets in parallel plate configuration is also analyzed here to understand the performance of the proposed formulation. The schematic of the benchmark problem for the bounded flow is shown in Fig. 3.4A

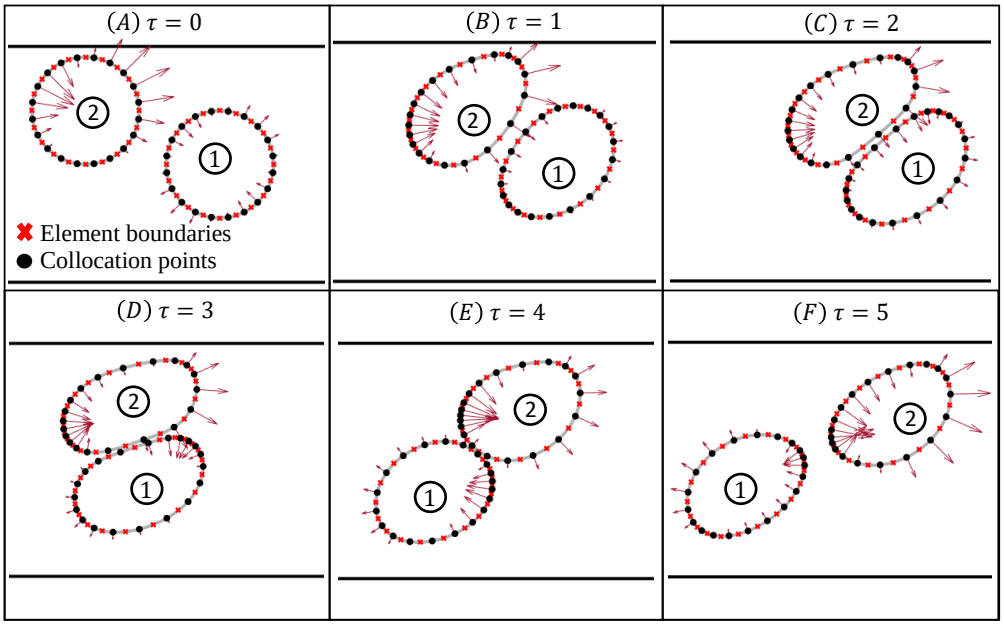

Figure 3.5: Time evolution of two particles between parallel plates with mesh relaxation and volume correction algorithms

where a droplet in the parallel plate configuration is analyzed. Although the channel boundaries can be generated by linear elements, the degree elevation algorithm is applied to increase their polynomial order to two. The particle radius is chosen as unity, $R = 1$, and different width, W, values are used. u_x is defined on the plates such that it results in a unit shear rate. The evolution of D with respect to time is shown in Fig. 3.4B where results for $W = 16$ match with the unbounded flow case. The simulation parameters are $Ca = 0.2, \lambda = 1$, and $\Delta\tau = 0.01$. The number of collocation points on the particle and channels are 20 and 166, respectively. The channel length is 10 times the W for all simulations. It can also be observed from Fig. 3.4B that decreasing channel width results in an increased amount of deformation.

The application of mesh relaxation and volume correction to two-particle interaction in parallel plate configuration is demonstrated in Fig. 3.5 with arrows denoting the collocation point velocities normal to the surface. The simulation parameters are $Ca = 0.2, \lambda = 1$, and $\Delta\tau = 0.01$ with 20 and 83 collocation points on each particle and channel boundary. The initial coordinates of the center of

gravity of the particles are $CG_1 = (0, 0)$ and $CG_2 = (-2.5, 1)$ with $W = 2.2$. The elements on the second particle distort severely and result in an intersection with the wall which blows up the simulation. Although this problem can be alleviated by choosing a high number of collocation points with smaller time steps, it is not a feasible solution since it increases the computational cost dramatically. It can be seen from Fig. 3.5 that the mesh relaxation algorithm overcomes these problems by preserving the mesh quality throughout the simulation which ensures the stability of calculations. Maintaining similar element length is important in this problem due to the close interaction of particles which can be seen from Fig. 3.5C to Fig. 3.5E. Although singularity subtraction and Telles transformation are used to calculate singular integrals, near-singularities might appear due to the close distances between two surfaces. Hence, the mesh relaxation algorithm prevents such problems by maintaining the element lengths similar which decreases the severity of near-singular integrals.

Chapter 4

Three-Dimensional Droplet Motion

In this section, IGABEM formulations are proposed for 3D droplet motion in both unbounded and bounded flow cases. First, the droplet motion in an unbounded flow with an external flow field is discussed where singularity-subtraction and stabilization techniques are introduced with implementation details. High-deformation cases are investigated where stabilization techniques are used to reduce the computational cost. High-viscosity ratio cases are investigated where the singularity-subtraction technique is not sufficient to produce accurate results. Then, the isoviscous droplet deformation between parallel plates is analyzed as a benchmark problem for droplet motion in bounded flow. Finally, the droplet motion in a Y-junction is demonstrated.

4.1 Unbounded Flow

The boundary integral equation for the velocity of a point $\mathbf{x}_o$ positioned on the interface Γ can be expressed as follows:

$$\begin{aligned}(1+\lambda)u_j(\mathbf{x}_o) = 2u_j^{\infty}(\mathbf{x}_o) - \frac{1}{4\pi}\int_{\Gamma}\mathcal{G}_{ij}(\mathbf{x},\mathbf{x}_o)\Delta f_i(\mathbf{x})d\Gamma \\ + \frac{1-\lambda}{4\pi}\int_{\Gamma}u_i(\mathbf{x})\mathcal{T}_{ijk}(\mathbf{x},\mathbf{x}_o)n_k(\mathbf{x})d\Gamma\end{aligned} \tag{4.1}$$

for $j = 1, 2, 3$. u_j represents the velocity component in the j^{th} direction. $\lambda = \mu_0/\mu_1$ denotes the viscosity ratio of the inner fluid to the external fluid. u_j^{∞} represents the j^{th} component of the background flow. $\mathcal{G}$ and $\mathcal{T}$ are Stokeslet and Stresslet, respectively, and they are expressed in Eq. (2.52a) and Eq. (2.52b). The stress discontinuity at the interface can be expressed for a droplet (*i.e.* extensible body) as follows [49]:

$$\Delta\mathbf{f}(\mathbf{x}) = \frac{2}{Ca}\kappa(\mathbf{x})\mathbf{n}(\mathbf{x}) \tag{4.2}$$

The following identities can be used to subtract the singularities that appear in surface integrals [58]:

$$\int_{\Gamma}\mathcal{G}(\mathbf{x},\mathbf{x}_o)\mathbf{n}(\mathbf{x})d\Gamma = 0 \tag{4.3a}$$

$$\frac{1}{8\pi}\int_{\Gamma}\mathcal{T}(\mathbf{x},\mathbf{x}_o)\mathbf{n}(\mathbf{x})d\Gamma = -\frac{1}{2}\mathbf{I} \tag{4.3b}$$

when $\mathbf{x}_o$ is on Γ. Single- and double-layer potentials take the following form after singularity subtraction is performed [38, 42, 68]:

$$\int_{\Gamma}\mathcal{G}(\mathbf{x},\mathbf{x}_o)\Delta\mathbf{f}(\mathbf{x})d\Gamma = \frac{2}{Ca}\int_{\Gamma}\mathcal{G}(\mathbf{x},\mathbf{x}_o)[\kappa(\mathbf{x})-\kappa(\mathbf{x}_o)]\mathbf{n}(\mathbf{x})d\Gamma \tag{4.4a}$$

$$\int_{\Gamma}\mathbf{u}(\mathbf{x})\mathcal{T}(\mathbf{x},\mathbf{x}_o)\mathbf{n}(\mathbf{x})d\Gamma = \int_{\Gamma}\left[\mathbf{u}(\mathbf{x})-\mathbf{u}(\mathbf{x_o})\right]\mathcal{T}(\mathbf{x},\mathbf{x_o})\mathbf{n}(\mathbf{x})d\Gamma - 4\pi\mathbf{u}(\mathbf{x_o}) \tag{4.4b}$$

Both geometry and unknown field variables are represented by B-splines as described in Eq. (3.7). Positions of source points in the parameter domain, $\boldsymbol{\xi}_i^{'}$, are determined by using Greville abscissae which can then be mapped to the physical domain to obtain locations of $\mathbf{x}_o$ (collocation points). The initial configuration of the object is selected such that the surface is C^2 continuous at collocation points [36]. Once the initial configuration of the object is determined,

the evolution of the object's interface can be described as shown in Eq. (3.11) by considering the normal components of velocities. In the simulation of deformable bodies, there are two important issues that require special attention: **(i)** volume correction and **(ii)** mesh relaxation.

4.1.1 Volume correction

The incompressibility condition dictates no volume change for the fluid enclosed in a droplet; however, discretization errors lead to drifts of the volume in long-time horizon simulations (see Fig.4.1). Therefore, at some point during the time marching, volume correction needs to be applied when the volume change exceeds a certain tolerance similar to the 2D case. To achieve this, a non-linear optimization problem can be solved to find the modified positions of the control points, $\mathbf{X}_i$ as: [69, 70]

$$\arg \min_{V=V_0} \|\mathbf{X}_i - \mathbf{P}_i\| \tag{4.5}$$

for $i = 1, ..., N$ where N is the total number of collocation points. Sequential quadratic programming is applied for the local correction where the constraint requires the droplet volume to be fixed. V_0 is the initial volume that has to be conserved throughout the simulation. Note that the implementation is the same as in 2D.

4.1.2 Mesh relaxation

Although the initial distances between adjacent nodes can be chosen appropriately, these distances may change and cause instabilities, especially when the collocation points shrink to a point. Moreover, in some cases, it may be desirable to concentrate nodes towards the regions with high curvature values. To accomplish that, a non-physical tangential velocity is implemented to the i^{th} collocation

point to prevent mesh distortion as shown below: [43]

$$\mathbf{w}(\mathbf{x}_i) = \frac{N_\Delta^{3/2}}{1+\lambda}[\mathbf{I} - \mathbf{n}(\mathbf{x}_i)\mathbf{n}(\mathbf{x}_i)] \cdot \sum_{j=1}(C_{r1} + C_{r2}|\kappa_j(\mathbf{x})|^{3/2}) S_j^e(\mathbf{x}_j - \mathbf{x}_i) \tag{4.6}$$

where N_Δ is the total number of elements and j stands for the neighboring points to the i^{th} collocation point. S_j^e is the surface area of the element where j^{th} node is located. Note that, Greville abscissae results in face-centered collocation points for quadratic B-splines meaning that each node has its corresponding element. This is not the case for collocation points located at the poles where S_j^e is the mean area of the degenerated elements around the poles. C_{r1} and C_{r2} are control parameters (details about these parameters can be found elsewhere [43]). Depending on the type of flow or magnitude of deformation, these parameters help to maintain similar element lengths and/or cluster nodes around high curvature zones. C_{r1} controls the element lengths which is quite essential for the cases where the elongation of droplet is drastic. With the increasing C_{r1}, the length of the inter-node distances tends to remain similar to each other. On the other hand, C_{r2} controls the clustering of collocation points to the regions with high deformation. With the increasing C_{r2}, the level of clustering improves around the high deformation zones. The pseudo-code of the proposed method for taking one time step is shown below:

Algorithm 1 Taking one time step with the proposed method

1: Calculate single and double-layer potentials
2: Calculate the velocity at each collocation point
3: Evaluate the normal components of collocation point velocities and map those velocities to the control points
4: Update the position of control points using their modified velocities
5: **if** the volume change exceeds the tolerance **then**
6: Apply volume correction
7: **end if**
8: Apply mesh relaxation

As a test case, a droplet in a shear flow ($u^\infty = [\dot{\gamma}z, 0, 0]$) is considered. The code development is performed in MATLAB® environment where elementwise parallelization is performed for the calculation of layer-potentials by using `parfor`

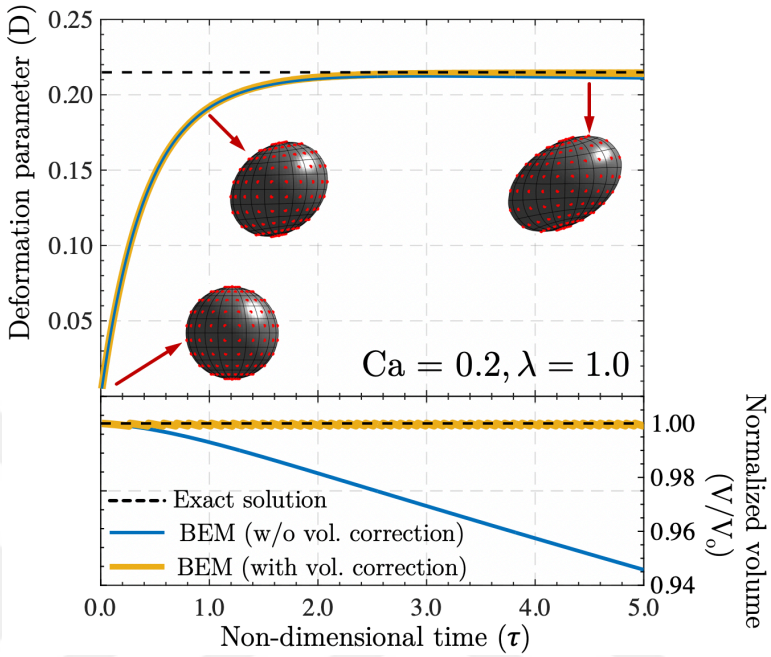

Figure 4.1: Taylor deformation parameter and non-dimensional volume history of the droplet with and without employing volume correction for $\lambda = 1$.

command. The mesh relaxation algorithm requires the calculation of the normal vector, curvature, and surface area of the element for each collocation point which is inherently parallel so a similar parallelization is performed for the determination of the tangential velocities. MATLAB built-in function `fmincon` is used to solve Eq. 4.5 iteratively which can also be parallelized. HP-Z6-G4 workstation which utilizes 24 Cores of CPU and 128 GB of RAM is used for the simulations. 0.1% relative volume change limit is set as the tolerance for applying the volume correction algorithm. In this specific problem, shear flow results in two symmetric high-curvature zones. By condensing the nodes towards those regions, stable and accurate results can be obtained without requiring high-fidelity simulations. To accomplish this, following some preliminary tests, we used $C_{r1} = 0.0001$ and $C_{r2} = 0.0002$ in this study. The droplet deformation is quantified by the Taylor deformation parameter which is defined as $D = (L_{max} - L_{min})/(L_{max} + L_{min})$. L_{max} and L_{min} denotes the maximum and minimum distances from the center of gravity of the droplet. An analytical solution for Taylor deformation under shear flow is also available in the literature at the small deformation (i.e. $Ca < 0.3$)

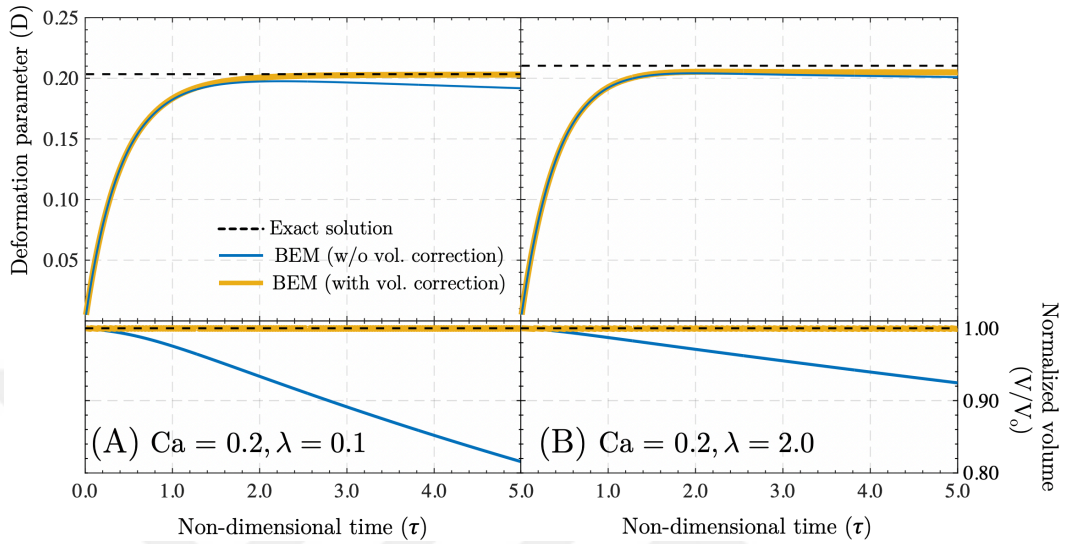

Figure 4.2: Volume changes for droplets with different viscosity ratios

limit as: [71]

$$D = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} = \frac{5(19\lambda + 16)}{4(\lambda + 1)\sqrt{\left(\frac{20}{Ca}\right)^2 + (19\lambda)^2}} \tag{4.7}$$

The results are plotted as a function of time, and time is non-dimensionalized as $\tau = t\dot{\gamma}$ where the shear rate is taken as unity for all cases considered. Fig. 4.1, shows the evolution of the deformation parameter and normalized volume for $Ca = 0.2$ and $\lambda = 1.0$. Both cases with and without the volume correction algorithm as well as the steady analytical results coming from the small deformation theory are also included in the figure. The number of collocation points used in these simulations is 202 for both cases. When the volume correction is not applied, the error in the Taylor deformation parameter is 1.9%, and the change in the droplet's volume is 5.4%. The change in the volume can be decreased by employing a higher number of collocation points with additional computational costs. However, the presence of volume correction enables simulations at low spatial resolutions while preserving the volume. Moreover, the error for the Taylor deformation parameter decreases from 1.9% to 0.2% when the volume correction is applied. Therefore the results reveal that volume correction also improves the accuracy of the solution. Fig. 4.2 exhibits the change in volume for droplets with

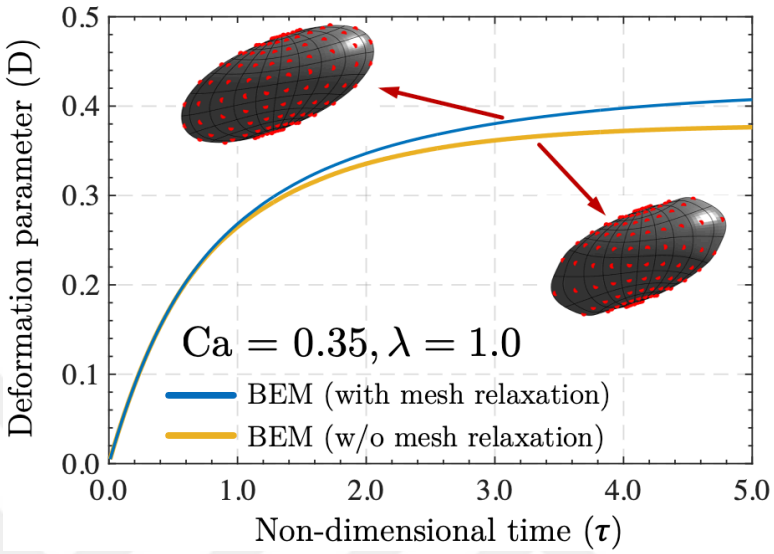

Figure 4.3: Droplet shapes for $Ca = 0.35$, $\lambda = 1.0$ with and without mesh relaxation (number of collocation points = 202)

different viscosity ratios ($\lambda = 0.1$ and $\lambda = 2.0$) for $Ca = 0.2$. Here, the droplets are discretized with 130 collocation points. Although the change in volume is acceptable for $\lambda = 2.0$ case, it reaches 18.5% for $\lambda = 0.1$ case. These results show that the volume correction algorithm is even more essential at low viscosity ratio values.

Mesh distortion occurs especially for high Ca cases where the element size located in high curvature zones grows which leads to non-physical droplet shape. One extreme example is shown in Fig. 4.3 for $Ca = 0.35$ and $\lambda = 1.0$ where volume correction is applied. Increasing the number of collocation points yields smoother results with additional computational costs. To obtain smoother results without increasing the number of collocation points, mesh relaxation is employed. The mesh relaxation step (virtual step) is implemented once between each physical time step. The analytical result for the deformation parameter at this Ca is 0.3633, but the analytical solution is valid only for low deformation regimes and in this example, the droplet deforms significantly. Hence, the result obtained by IGABEM with 514 collocation points, $D = 0.4036$ [49] is considered as the ground truth for the comparison. Fig. 4.3 shows the evolution of the Taylor deformation parameter when both volume correction and mesh relaxation are applied. The

Table 4.1: Effect of mesh relaxation. Note that the analytic solution is valid only for small deformation. Hence the high-resolution solutions should be considered ground truth for these high-deformation cases.

		$\lambda = \mathbf{1.0}$		$\lambda = \mathbf{0.1}$	
	Ca	$\boldsymbol{D}$	**Avg. Time [s]**	$\boldsymbol{D}$	**Avg. Time [s]**
Low resolution	0.35	0.3763	17.06	0.3559	16.79
w\o mesh relax.	0.40	0.4267	16.89	0.3997	17.76
Low resolution	0.35	0.4072	17.90	0.3920	18.06
with mesh relax.	0.40	0.4688	18.44	0.4503	18.26
High resolution	0.35	0.4028	176.54	0.3890	177.91
w\o mesh relax.	0.40	0.4583	176.91	0.4429	172.97

Taylor deformation parameter values obtained with and without mesh relaxation at the time horizon are 0.4072 and 0.3763, respectively. Hence, mesh relaxation decreases the error from 6.8% to 0.9%. In addition, once mesh relaxation is employed, the frequency of the execution of volume correction also reduces. The number of execution of the volume correction algorithm is found to be 34 in the case of mesh relaxation instead of 89 without mesh relaxation. Hence, preserving the mesh quality also reduces the need for volume correction which also leads to lower computational cost (which will be further quantified in the next example).

To analyze the computational efficiency of the proposed method, more droplet simulations at relatively high deformation are performed for the viscosity ratio values ($\lambda = 0.1$ and $\lambda = 1.0$) and the capillary number values ($Ca = 0.35$ and $Ca = 0.40$). These simulations are performed at a low resolution (202 collocation points) with and without mesh relaxation and at a high resolution. Without any mesh relaxation, high spatial and temporal resolutions are required to achieve comparable accuracy to that of mesh relaxation. In these tests, 802 collocation points are needed. The results are presented in Tab. 4.1. The Taylor deformation parameter values are reported at the non-dimensional time of $\tau = 5.0$. The results show that applying the mesh relaxation at low resolution delivers results close to that of fine resolution. It should be noted that the time step used for

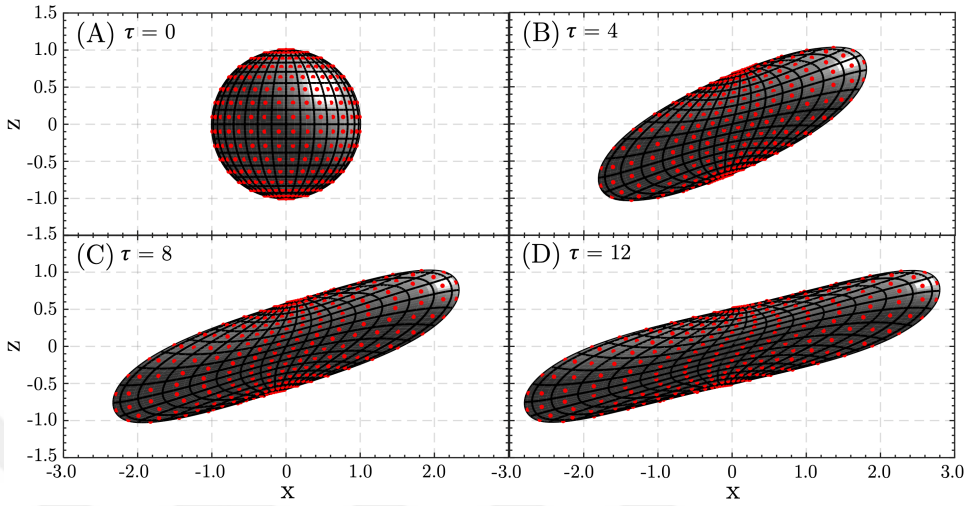

Figure 4.4: Time evolution of droplet in a shear flow ($Ca = 0.5$, $\lambda = 0.1$): (A) $\tau = 0$, (B) $\tau = 4$, (C) $\tau = 8$ and (D) $\tau = 12$

the low-resolution cases is 0.01 whereas it is halved for the high-resolution cases since when the element size decreases, the time step size needs to be decreased to satisfy the CFL (Courant-Friedrichs-Lewy) condition. Average elapsed time values for one-time steps are also tabulated to compare the computational cost of mesh relaxation. The results reveal that the mesh relaxation algorithm slightly increases the computational cost since the algorithm is inherently parallel and it also reduces the need to apply the volume correction algorithm (For $Ca = 0.4$, the number of execution of the volume correction algorithm is found to be 38 in the case of mesh relaxation compared to 111 without mesh relaxation). Low-fidelity simulations are more than 10 times faster compared to high-fidelity simulations. Hence, mesh stabilization algorithms (i.e. volume correction and mesh relaxation) are crucial for long-run simulations of droplets at low computational cost which would be extremely important, especially for the cases where the additional degree of freedom for the system is introduced due to some confinement as in the case of microfluidic applications. It has been observed in additional simulations that the application of mesh relaxation ensures the stability of high-fidelity simulations with $\Delta\tau = 0.01$ since it prevents nodes to shrink down to the poles.

The performance of the proposed method is also tested at an extreme deformation case where $Ca = 0.5$ and $\lambda = 0.1$. The shape evolution can be seen in

Fig. 4.4. The change in the surface area is quite significant —- 48.3% change in the surface area at $\tau = 12$. Although the elongation is severe, the mesh relaxation algorithm preserves the mesh quality, especially around high curvature zones as seen in the figure. The stable and accurate simulation of this case with the mesh relaxation algorithm was achieved only with 514 collocation points.

The singularity-subtraction method presented at the beginning of the section provides a unified way to calculate layer potentials without the need of identifying singular and regular integrals. It makes the implementation easier since layer potentials are calculated by using Eq. (4.4) only. Although the singularity-subtraction method performs well for low viscosity ratio or isoviscous cases, relatively larger errors have been observed for high-viscosity case analyses which are also reported in [43]. The singularities in integrals are weakened but they are still available in the integral domain. Non-singular boundary integral equations are proposed in [72] to overcome this problem by converting surface integrals to line integrals where source points do not lie on the integral domain. In this study, polar-coordinate transformation is applied for singular integrals on the droplet. Regular integrals are calculated by using the Gauss-Legendre quadrature. The droplet in the extension flow is analyzed where analytical results are shown by Barthes-Biesel and Acrivos [73]. The droplet with a radius R is placed in an extensional flow field, with a velocity scale V, described as:

$$\mathbf{u}(\mathbf{x}) = \frac{V}{2R} \begin{bmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{x} \tag{4.8}$$

where $\mathbf{x}$ is a point on the droplet surface. The droplet is initially spherical and it will elongate up to a certain deformation parameter, D. The capillary number is defined as $Ca = \mu_1 V/\gamma$ where γ represents the surface tension. Note that the stress discontinuity defined on the interface is taken as $\Delta\mathbf{f}(\mathbf{x}) = 2\gamma\kappa(\mathbf{x})\mathbf{n}(\mathbf{x})$. R, V and μ are taken as unity with $\lambda = 10$ which is the problematic case when singularity-subtraction is applied. Results obtained by IGABEM using polar-coordinate transformation and singularity-subtraction with the analytical results

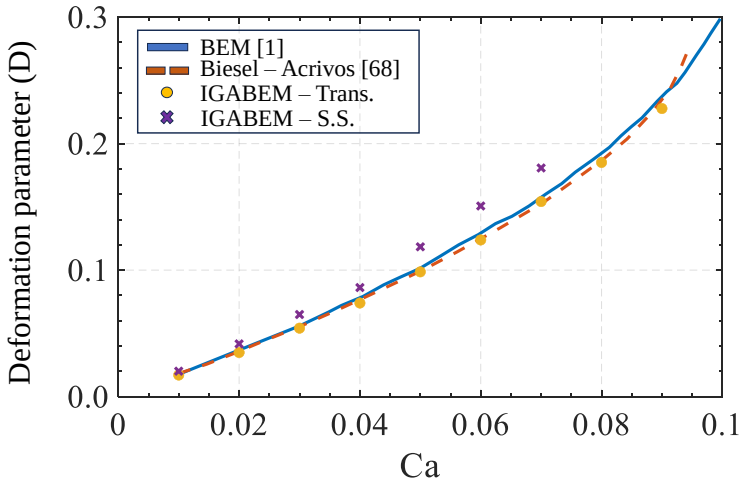

Figure 4.5: Taylor deformation parameter, D, at different Ca values

are shown in Fig. 4.5. Note that, BEM results from [1] are also shown for comparison. It can be seen from the figure that the singularity-subtraction method deviates from the expected results and it gives unstable results for $Ca > 0.07$. On the other hand, applying the polar-coordinate transformation to alleviate singular integrals on the droplet performs well even with the cost of additional coding effort and computational cost.

4.2 Bounded Flow

The boundary integral equation for a droplet, Γ_0, in wall-bounded, Γ_1, flow can be expressed as follows for $\mathbf{x}_o \in (\Gamma_0 \cup \Gamma_1)$ [1]:

$$\begin{aligned}\alpha(\mathbf{x_o})\mathbf{u}(\mathbf{x_o}) = &-\frac{1}{\mu_1}\int_{\Gamma_1}\mathcal{G}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{f}(\mathbf{x})d\Gamma + \int_{\Gamma_1}\mathbf{u}(\mathbf{x})\cdot\mathcal{T}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{n}(\mathbf{x})d\Gamma \\ &-\frac{1}{\mu_1}\int_{\Gamma_0}\mathcal{G}(\mathbf{x},\mathbf{x_o})\cdot\Delta\mathbf{f}(\mathbf{x})d\Gamma + (1-\lambda)\int_{\Gamma_0}\mathbf{u}(\mathbf{x})\cdot\mathcal{T}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{n}(\mathbf{x})d\Gamma\end{aligned} \tag{4.9}$$

where

$$\alpha(\mathbf{x_o}) = \begin{cases} 4\pi, & \mathbf{x_o} \in \Gamma_1 \\ 4\pi(1+\lambda), & \mathbf{x_o} \in \Gamma_0 \end{cases} \tag{4.10}$$

The identities presented in Eq.(4.3) are employed to subtract singularities for layer potentials defined on the droplet. The isogeometric discretization of Eq.(4.9) is given by the following expression:

$$\alpha(\mathbf{x_o})\sum_{m=1}^{M} R_m(\xi_o,\eta_o)\mathbf{U}_m = -\sum_{e=1}^{N_1^e}\frac{\mathbf{B}_e(\mathbf{x},\mathbf{x_o})}{\mu_1} + \sum_{e=1}^{N_1^e}\mathbf{C}_e(\mathbf{x},\mathbf{x_o}) - \sum_{e=1}^{N_0^e}\frac{\mathbf{D}_e(\mathbf{x},\mathbf{x_o})}{\mu_1} + (1-\lambda)\left(\sum_{e=1}^{N_0^e}\mathbf{E}_e(\mathbf{x},\mathbf{x_o}) - 4\pi\sum_{m=1}^{M} R_m(\xi_o,\eta_o)\mathbf{U}_m\right) \tag{4.11}$$

where

$$\mathbf{B}_e(\mathbf{x},\mathbf{x_o}) = \int_{\hat{\Gamma}_1^e} \mathcal{G}(\mathbf{x},\mathbf{x_o})\cdot\left(\sum_{m=1}^{M} R_m(\xi,\eta)\mathbf{F}_m\right) J(\xi,\eta)d\hat{\Gamma} \tag{4.12a}$$

$$\mathbf{C}_e(\mathbf{x},\mathbf{x_o}) = \int_{\hat{\Gamma}_1^e}\left(\sum_{m=1}^{M} R_m(\xi,\eta)\mathbf{U}_m\right)\cdot\mathcal{T}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{n}(\mathbf{x})J(\xi,\eta)d\hat{\Gamma} \tag{4.12b}$$

$$\mathbf{D}_e(\mathbf{x},\mathbf{x_o}) = \frac{2}{Ca}\int_{\hat{\Gamma}_0^e}\left[\kappa(\mathbf{x})-\kappa(\mathbf{x_o})\right]\mathcal{G}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{n}(\mathbf{x})J(\xi,\eta)d\hat{\Gamma} \tag{4.12c}$$

$$\mathbf{E}_e(\mathbf{x},\mathbf{x_o}) = \int_{\hat{\Gamma}_0^e}\left(\sum_{m=1}^{M} R_m(\xi,\eta)\mathbf{U}_m - \sum_{m=1}^{M} R_m(\xi_o,\eta_o)\mathbf{U}_m\right)\cdot\mathcal{T}(\mathbf{x},\mathbf{x_o})\cdot\mathbf{n}(\mathbf{x})J(\xi,\eta)d\hat{\Gamma} \tag{4.12d}$$

Collocating Eq. (4.11) at each collocation point yields the following system of equations:

$$\begin{bmatrix}\mathbf{A}_{00} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_{11}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{U}_0\\ \mathbf{U}_1\end{bmatrix} = \begin{bmatrix}\mathbf{B}_{01}\\ \mathbf{B}_{11}\end{bmatrix}\cdot\mathbf{F}_1 + \begin{bmatrix}\mathbf{C}_{01}\\ \mathbf{C}_{11}\end{bmatrix}\cdot\mathbf{U}_1 + \begin{bmatrix}\mathbf{D}_{00}\\ \mathbf{D}_{10}\end{bmatrix} + \begin{bmatrix}\mathbf{H}_{00}\\ \mathbf{H}_{10}\end{bmatrix}\cdot\mathbf{U}_0 \tag{4.13}$$

where subscripts 0 and 1 denote integral calculations on Γ_0 and Γ_1, respectively. $\mathbf{H}_{00}$ and $\mathbf{H}_{10}$ are obtained by collocating the last term in the parenthesis on the right-hand side of Eq. (4.11). Note that, $\mathbf{B}_{11}$ and $\mathbf{C}_{11}$ represent single- and double-layer potentials calculated on the channel. These two matrices are calculated prior to the iterations performed to simulate the evolution of the droplet interface so that computational cost is decreased. The discontinuous collocation scheme is utilized for the elements residing on the channel. Polar-coordinate transformation is employed to calculate singular integrals on Γ_1 with 10 Gauss points in each

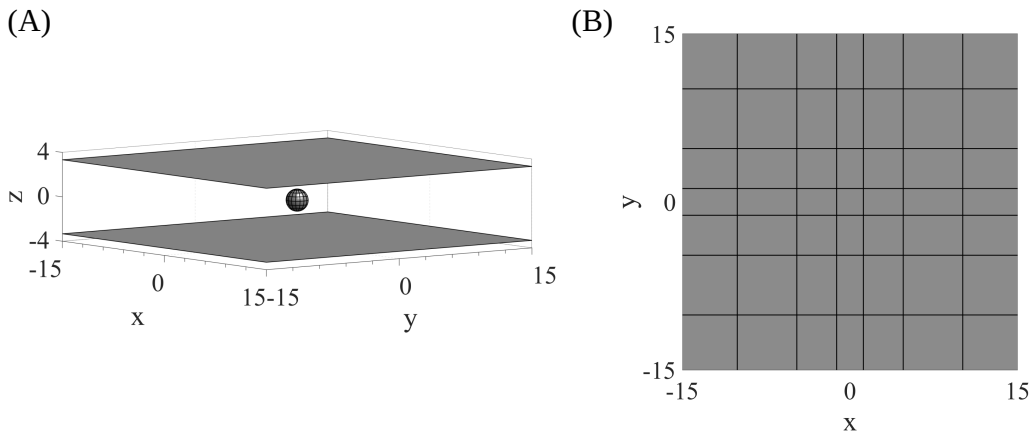

Figure 4.6: (A) The schematic representation of the droplet between parallel plates (B) Elements on the plate

direction. The adaptive subdivision scheme is used to calculate nearly-singular integrals with a relative error tolerance and a subdivision limit of $e_i = 10^{-2}$ and 10, respectively.

As a benchmark problem, isoviscous droplet deformation between parallel plates is analyzed. Janssen and Anderson [3] worked on the same problem by using the modified Green's functions and compared their results with the small deformation theory [2]. The plates are generated by defining a square surface initially. Then, the degree elevation algorithm is implemented to increase the degree of the surfaces to second order. Knot insertion is applied such that the elements are refined around the center of the plates where the discontinuous collocation technique is implemented afterwards. The schematic representation of the problem and meshes on the plates are shown in Fig. 4.6. The parameters of the simulations are the same as shown in Fig. 3.4A where the Taylor deformation parameter obtained at different confinement ratios is used for comparison. Dirichlet boundary condition is defined on the plates such that flow with a unit shear rate is generated in the $x-$direction. The length of the plates is chosen 24 times larger than the width of the confinement after analyzing the particle near a wall problem [74]. 202 and 729 collocation points are used on the droplet and

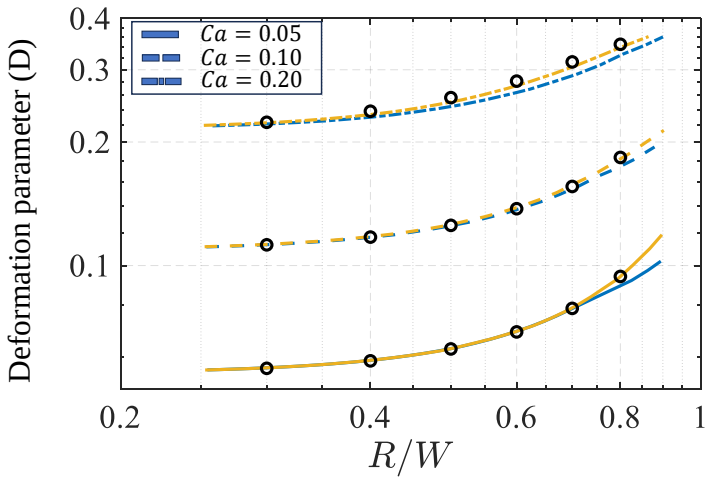

Figure 4.7: Deformation parameters at different confinement ratios. Blue and yellow lines correspond to the analytical [2] and BEM [3] results, respectively. Black markers represent IGABEM solutions.

on a single plate. Deformation parameters when the steady state is reached are shown in Fig. 4.7.

Although the droplet elongates up to a specific configuration where the steady state is reached, it is more complex for $Ca = 0.2$ at $R/W = 0.8$. The droplet deforms up to a maximum value and then slowly retracts until a steady state shape is obtained [3]. The evolution of the deformation parameter in time is shown in Fig. 4.8. The droplet stretches out up to $D = 0.3548$ and then retracts to $D = 0.3461$ where steady shape is attained.

In the final example, the deformation of a droplet and rigid particle in a microchannel is examined. The schematic representation of the Y-junction microchannel is shown in Fig. 4.9. The whole geometry is non-dimensionalized with respect to the particle radius which is equal to one. The microchannel height is $L = 4$ and the width of each branch is $W = 4$. The velocity is defined at both inlets with a mean velocity equal to 3 where outlet pressure is taken as 0. The surface of the channel consists of 18 different quadratic patches that are obtained after the degree elevation algorithm is applied to linear patches. The impedance formulation proposed by Karakaya *et al.* [27] is employed to simulate the motion

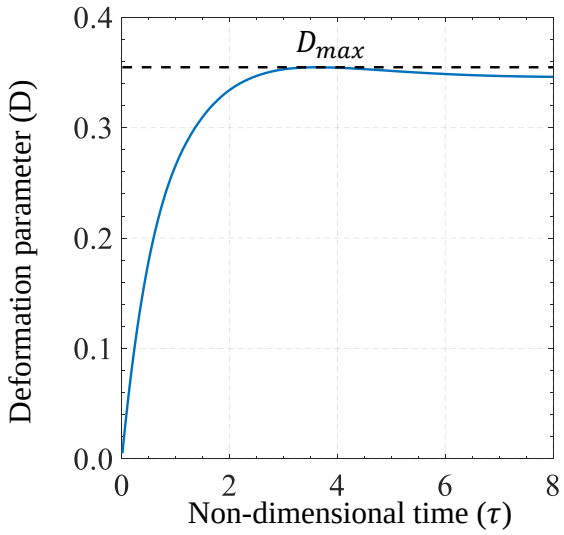

Figure 4.8: Evolution of the deformation parameter in time for $Ca = 0.2$ and $R/W = 0.8$ for the droplet between parallel plates

of the rigid particle. The velocity of each point, $\mathbf{u}$, can be expressed as follows for a rigid particle:

$$\mathbf{u} = \mathbf{u}_B + \boldsymbol{\omega} \times \mathbf{r} \tag{4.14}$$

where $\mathbf{u}_B$ is the velocity of the center of the particle and $\boldsymbol{\omega}$ is the angular velocity of the particle with $\mathbf{r}$ being the position vector of points on the particle surface with respect to the center of the particle. $\mathbf{u}_B$ and $\boldsymbol{\omega}$ are determined by using the fact that the particle is force and torque-free. The expressions for the force and torque on the particle are:

$$\mathbf{F}_B = \int_\Gamma \mathbf{f}(\mathbf{x}) d\Gamma \tag{4.15a}$$

$$\mathbf{T}_B = \int_\Gamma (\mathbf{x} - \mathbf{x}_B) \times \mathbf{f}(\mathbf{x}) d\Gamma \tag{4.15b}$$

where $\mathbf{x}_B$ denotes the position of the center of gravity. $\mathbf{F}_B$ and $\mathbf{T}_B$ are the net force and torque on the particle, respectively.

The rigid particle is modeled by employing quadratic NURBS as explained in Section 2.2.1 since the curvature information is not needed for a rigid particle. 202 and 182 collocation points are employed for the deformable and rigid particles,

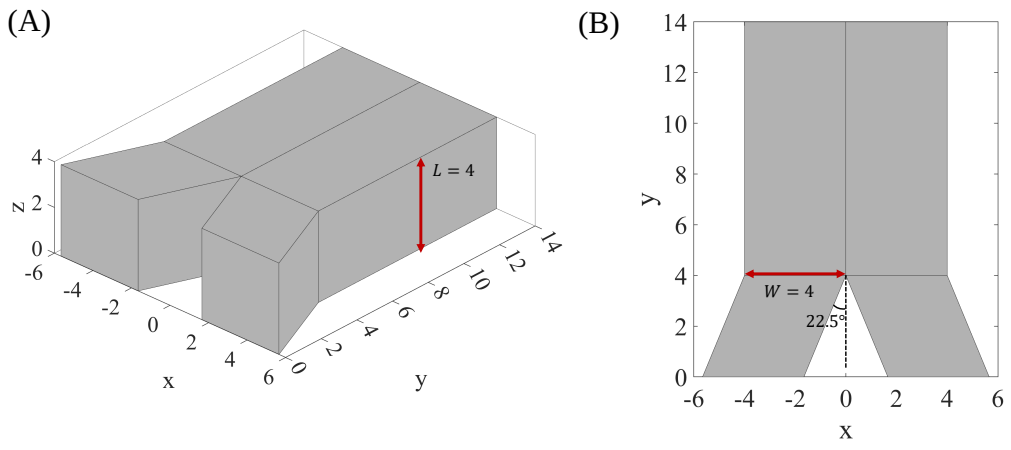

Figure 4.9: (A) Isometric and (B) upper view of the Y-junction microchannel

respectively. 1458 collocation points are employed on the channel by applying the discontinuous collocation technique. The initial position of particles is $\mathbf{x}_B(0) = [2.8284, 2, 2]$ and $Ca = 0.05$ for the deformable particle with $\lambda = 1$. The time step used in both simulations is $\Delta\tau = 0.01$ and simulations are terminated when $\tau = 2$. The velocity calculated at collocation points is directly applied to modify the geometry without taking the normal components into account only since the mesh distortion is observed to diminish in this way.

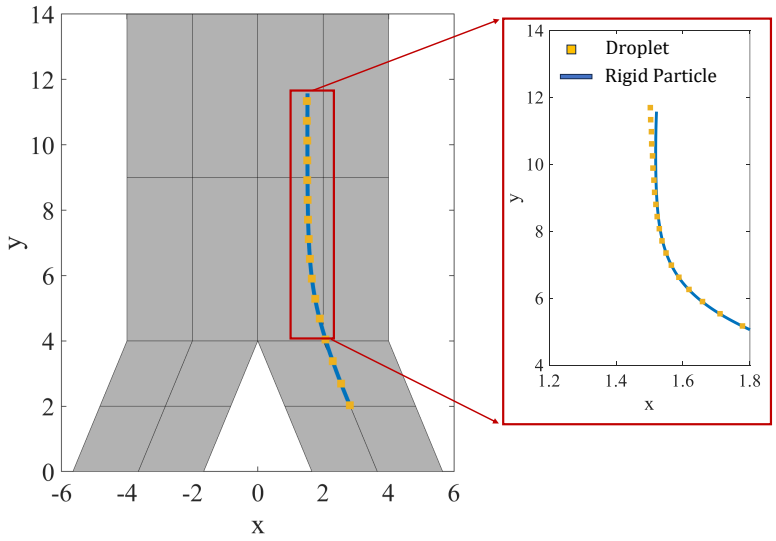

Figure 4.10: Trajectories of the droplet and rigid particle

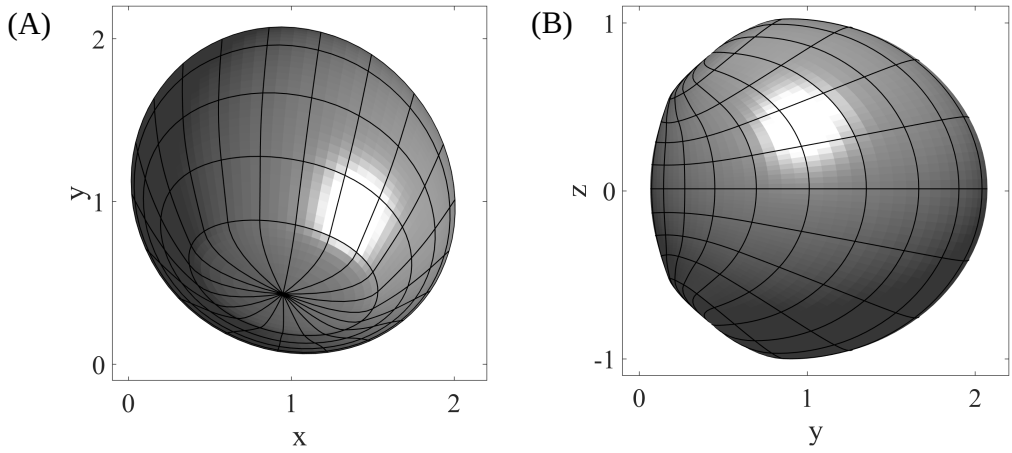

Figure 4.11: The final shapes of the droplet in (A) $x-y$ plane (B) $y-z$ plane

The trajectories for both particles are shown in Fig. 4.10. It can be seen from the trajectory that drift is observed for a droplet which is not the case for a rigid particle due to symmetry. The cross-stream drift of drops is observed when the symmetry is lost due to the shear gradient in the microchannel [75–77]. The final shape of the droplet is shown in Fig. 4.11. The shear gradient in the $x-$direction results in a cross-stream drift towards the centerline. z value of the center of gravity of the particle does not change since the symmetry is not lost in the $z-$direction and the particle deforms according to the parabolic flow profile.

Chapter 5

Concluding Remarks

IGABEM formulation for deformable particles with arbitrary viscosity ratios both in free and confined flows is proposed in this study. Deformable particles are differentiated by their stress discontinuity terms and expressions for droplet, inextensible membrane, and red blood cells are shown in Eq. (1.1)-(1.3). Without loss of generality, simulations are performed for droplets. Stabilization methods, namely volume correction and mesh relaxation, are applied to obtain accurate and stable results with low-resolution simulations. The volume correction algorithm avoids the drift in the enclosed volume inside a droplet by solving a non-linear optimization problem that modifies the control points of the B-spline curve or surface. Apart from preventing unphysical volume changes, it also increases the accuracy of the simulation since volume change causes a change in the deformation parameter in long-run simulations. The mesh relaxation algorithm modifies the positions of control points based on the element size and local curvature values. It prevents mesh distortion which is essential for high Ca flows. The proposed stabilization methods are general in the sense that they can be applied to both 2D and 3D problems. The stabilization methods provide ten times faster results obtained by low-fidelity simulations compared to high-fidelity simulations with similar accuracy. It has been shown that the performance of the singularity-subtraction method is poor when the particle is highly viscous. Polar-coordinate transformation method is used to overcome this problem which

is shown for a droplet in an extensional flow example. IGABEM formulation for a droplet with microchannel confinement is also shown. Different and complex channel geometries can be analyzed in an efficient way due to the CAD integration feature of IGABEM which is shown by the channel with Y-junction. Although the proposed formulation is applied to droplet simulations, it can be extended to different deformable particles like inextensible membranes or red blood cells by modifying the stress discontinuity term. The stress discontinuity terms for different types of particles are shown in [36].

Apart from droplet simulations, the implementation details of IGABEM for potential and viscous flow problems are explained in this study by solving several benchmark problems. A special emphasis is put on the numerical quadrature schemes for nearly-singular and singular integrals. An adaptive subdivision technique is employed to calculate nearly-singular integrals. First, the required number of subdivisions is determined based on the desired relative error. Then, the Gauss-Legendre quadrature is applied to each part of the element. For the singular integrals, Polar-coordinate transformation is utilized successfully to calculate both single- and double-layer potentials. It should be noted that the employed quadrature techniques are applied in both potential and viscous flow problems. Moreover, they give highly accurate results for different element types like degenerated elements observed in the NURBS sphere. Non-linear heat transfer problems are solved with high accuracy both in 2D and 3D in which variable condensation technique is applied to enhance the efficiency of the proposed formulation. Viscous flow benchmark problems constitute the basis for many microfluidic applications where IGABEM provides similar accuracy obtained by implementing higher-order discretization techniques.

The proposed formulation can be extended to analyze multiple particles in microchannel confinement where accurate simulations with low resolution would be critical. Acoustic or electric fields can also be employed in the simulations so that different microfluidic applications (e.g., sorting and separation) would be modeled by IGABEM. Isogeometric analysis allows for analyzing complex microchannel geometries in an efficient way. Hence, microchannel optimization can also be performed by the proposed formulation.

Bibliography

[1] S. S. Simon, *3D Boundary Element Simulation of Droplet Dynamics in Microchannels: How Droplets Squeeze Through Constrictions and Move in Electric Fields*. PhD dissertation, École Polytechnique Fédérale De Lausanne, 2018.

[2] M. Shapira and S. Haber, "Low reynolds number motion of a droplet in shear flow including wall effects," *Int. J. Multiph. Flow*, vol. 16, no. 2, pp. 305–321, 1990.

[3] P. Janssen and P. Anderson, "Boundary-integral method for drop deformation between parallel plates," *Phys. Fluids*, vol. 19, no. 4, 2007.

[4] I. Fortelný and J. Jůza, "Description of the droplet size evolution in flowing immiscible polymer blends," *Polymers*, vol. 11, p. 761, 2019.

[5] S. Shen, H. Liu, Y. Liu, X. Liu, H. Hu, Z. Hu, and T. Wang, "Dynamic details inside water-in-oil (w/o) emulsion droplet and its impact on droplet evaporation and micro-explosion," *Fuel*, vol. 338, p. 127254, 2023.

[6] J. Liu, E. Sprujit, A. Miserez, and R. Langer, "Peptide-based liquid droplets as emerging delivery vehicles," *Nat. Rev. Mater.*, 2023.

[7] C. N. Baroud, F. Gallaire, and R. Dangla, "Dynamics of microfluidic droplets," *Lab Chip*, vol. 10, pp. 2032–2045, 2010.

[8] T. Tezduyar, "Interface-tracking and interface-capturing techniques for computation of moving boundaries and interfaces," in *Proceedings of the*

Fifth World Congress on Computational Mechanics, On-line publication: http://wccm. tuwien. ac. at/, Paper-ID, vol. 81513, 2002.

[9] V. Cristini and Y.-C. Tan, "Theory and numerical simulation of droplet dynamics in complex flows—a review," *Lab Chip*, vol. 4, no. 4, pp. 257–264, 2004.

[10] S. Chen and G. D. Doolen, "Lattice boltzmann method for fluid flows," *Annu. Rev. Fluid Mech.*, vol. 30, no. 1, pp. 329–364, 1998.

[11] K. Sankaranarayanan, I. Kevrekidis, S. Sundaresan, J. Lu, and G. Tryggvason, "A comparative study of lattice boltzmann and front-tracking finite-difference methods for bubble simulations," *Int. J. Multiph. Flow*, vol. 29, no. 1, pp. 109–116, 2003.

[12] G. Wang, L. Fei, and K. H. Luo, "Lattice boltzmann simulation of a water droplet penetrating a micropillar array in a microchannel," *Phys. Fluids*, vol. 33, no. 4, 2021.

[13] S. Osher and R. P. Fedkiw, "Level set methods: an overview and some recent results," *J. Comput. Phys.*, vol. 169, no. 2, pp. 463–502, 2001.

[14] E. Bjørklund, "The level-set method applied to droplet dynamics in the presence of an electric field," *Comput. Fluids*, vol. 38, no. 2, pp. 358–369, 2009.

[15] M. Choi, G. Son, and W. Shim, "A level-set method for droplet impact and penetration into a porous medium," *Comput. Fluids*, vol. 145, pp. 153–166, 2017.

[16] S. Van der Pijl, A. Segal, C. Vuik, and P. Wesseling, "A mass-conserving level-set method for modelling of multi-phase flows," *Int. J. Numer. Methods Fluids*, vol. 47, no. 4, pp. 339–361, 2005.

[17] R. Scardovelli and S. Zaleski, "Direct numerical simulation of free-surface and interfacial flow," *Annu. Rev. Fluid Mech.*, vol. 31, no. 1, pp. 567–603, 1999.

[18] S.-Z. Li, R. Chen, H. Wang, Q. Liao, X. Zhu, and Z.-B. Wang, "Simulation on the coalescence of the moving liquid column and droplet in a hydrophilic microchannel by volume of fluid method," *Appl. Therm. Eng.*, vol. 64, no. 1-2, pp. 129–138, 2014.

[19] J. Li, Y. Y. Renardy, and M. Renardy, "Numerical simulation of breakup of a viscous drop in simple shear flow through a volume-of-fluid method," *Phys. Fluids*, vol. 12, no. 2, pp. 269–282, 2000.

[20] H. Sauerland and T.-P. Fries, "The extended finite element method for two-phase and free-surface flows: A systematic study," *J. Comput. Phys.*, vol. 230, no. 9, pp. 3369–3390, 2011.

[21] H. H. Hu, N. A. Patankar, and M. Zhu, "Direct numerical simulations of fluid–solid systems using the arbitrary lagrangian–eulerian technique," *J. Comput. Phys.*, vol. 169, no. 2, pp. 427–462, 2001.

[22] S. Quan and D. P. Schmidt, "A moving mesh interface tracking method for 3d incompressible two-phase flows," *J. Comput. Phys.*, vol. 221, no. 2, pp. 761–780, 2007.

[23] C. S. Peskin, "The immersed boundary method," *Acta Numerica*, vol. 11, pp. 479–517, 2002.

[24] X. Gao, "A boundary-domain integral equation method in viscous fluid flow," *Int. J. Numer. Methods Fluids*, vol. 45, pp. 463–484, 2004.

[25] X.-W. Gao, "The radial integration method for evaluation of domain integrals with boundary-only discretization," *Eng. Anal. Bound. Elem.*, vol. 26, no. 10, pp. 905–916, 2002.

[26] G. Kabacaoğlu and E. Lushi, "Cross-stream migration of a vesicle in vortical flows," *Phys. Rev. E .*, vol. 107, no. 4, p. 044608, 2023.

[27] Z. Karakaya, B. Baranoğlu, B. Çetin, and A. Yazici, "A parallel boundary element formulation for tracking multiple particle trajectories in stoke's flow for microfluidic applications," *CMES - Comput. Model. Eng.*, vol. 104, no. 3, pp. 227–249, 2015.

[28] A. Topuz, B. Baranoğlu, and B. Çetin, "A multi-domain direct boundary element formulation for particulate flow in microchannels," *Eng. Anal. Bound. Elem.*, vol. 132, pp. 221–230, 2021.

[29] A. Atay, A. Beşkök, and B. Çetin, "Dc-electrokinetic motion of colloidal cylinder(s) in the vicinity of a conducting wall," *Electrophoresis*, vol. 43, no. 12, pp. 1263–1274, 2022.

[30] G. Kabacaoğlu and G. Biros, "Sorting same-size red blood cells in deep deterministic lateral displacement devices," *J. Fluid Mech.*, vol. 859, pp. 433–475, 2019.

[31] G. Kabacaoglu, B. Quaife, and G. Biros, "Low-resolution simulations of vesicle suspensions in 2D," *J. Comput. Phys.*, vol. 357, pp. 43–77, 2018.

[32] 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.*, vol. 89, pp. 191–200, 1978.

[33] R. Khayat, A. Luciani, and L. Utracki, "Boundary-element analysis of planar drop deformation in confined flow. Part 1. Newtonian fluids," *Eng. Anal. Bound. Elem.*, vol. 19, pp. 279–289, 1997.

[34] R. Khayat, "Boundary-element analysis of planar drop deformation in confined flow. Part 2. Viscoelastic fluids," *Eng. Anal. Bound. Elem.*, vol. 22, pp. 291–306, 1998.

[35] C. Pozrikidis, "Interfacial dynamics for Stokes flow," *J. Comput. Phys.*, vol. 169, pp. 250–301, 2001.

[36] A. A. Joneidi, *Three-dimensional boundary integral modeling of viscous drops and capsules*. PhD dissertation, Eindhoven University of Technology, 2014.

[37] C. Coulliette and C. Pozrikidis, "Motion of an array of drops through a cylindrical tube," *J. Fluid Mech.*, vol. 358, pp. 1–28, 1998.

[38] A. Z. Zinchenko, M. A. Rother, and R. H. Davis, "A novel boundary-integral algorithm for viscous interaction of deformable drops," *Phys. Fluids*, vol. 9, no. 6, pp. 1493–1511, 1997.

[39] 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.*, vol. 391, pp. 249–292, 1999.

[40] A. Z. Zinchenko and R. H. Davis, "Shear flow of highly concentrated emulsions of deformable drops by numerical simulations," *J. Fluid Mech.*, vol. 455, pp. 21–61, 2002.

[41] A. Z. Zinchenko and D. Robert H, "Large–scale simulations of concentrated emulsion flows," *Phil. Trans. R. Soc. Lond. A*, vol. 361, no. 1806, pp. 813–845, 2003.

[42] M. Loewenberg and E. Hinch, "Numerical simulation of a concentrated emulsion in shear flow," *J. Fluid Mech.*, vol. 321, pp. 395–419, 1996.

[43] I. R. Siqueira, R. B. Rebouças, T. F. Oliveira, and F. R. Cunha, "A new mesh relaxation approach and automatic time-step control method for boundary integral simulations of a viscous drop," *Int. J. Numer. Meth. Fluids*, vol. 84, pp. 221–238, 2017.

[44] A. Langins, A. P. Stikuts, and A. Cebĕrs, "A three-dimensional boundary element method algorithm for simulations of magnetic fluid droplet dynamics," *Phys. Fluids*, vol. 34, p. 062105, 2022.

[45] J. A. Cottrel, T. J. R. Hughes, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Comput. Methods. Appl. Mech. Eng.*, vol. 194, pp. 4135–4195, 2005.

[46] C. Lu, L. Chen, J. Luo, and H. Chen, "Acoustic shape optimization based on isogeometric boundary element method with subdivision surfaces," *Eng. Anal. Bound. Elem.*, vol. 146, pp. 951–965, 2023.

[47] Ö. C. Gümüş, B. Baranoğlu, and B. Çetin, "Isogeometric and NURBS-enhanced boundary element formulations for steady-state heat conduction with volumetric heat source and nonlinear boundary conditions," *Eng. Anal. Bound. Elem.*, vol. 145, pp. 299–309, 2022.

[48] Y. Wang and D. Benson, "Multi-patch nonsingular isogeometric boundary element analysis in 3 D," *Comput. Methods Appl. Mech. Eng.*, vol. 293, pp. 71–91, 2015.

[49] A. A. Joneidi, C. V. Verhoosel, and P. D. Anderson, "Isogeometric boundary integral analysis of drops and inextensible membranes in isoviscous flow," *Comput. Fluids*, vol. 109, pp. 49–66, 2015.

[50] D. Spink, "NURBS Toolbox." https://www.mathworks.com/matlabcentral/fileexchange/26390-nurbs-toolbox-by-d-m-spink, 2021.

[51] D. F. Rogers, *Introduction to NURBS With Historical Perspective*. Morgan Kauffmann, 2001.

[52] L. Piegl and W. Tiller, *The NURBS Book*. Springer-Verlag, 1997.

[53] K. J. Versprille, *Computer-aided design applications of the rational b-spline approximation form*. PhD dissertation, Syracuse University, 1975.

[54] T. N. E. Greville, "Numerical procedures for interpolation by spline functions," *SIAM J. Numer. Anal.*, vol. 1, pp. 53–68, 1964.

[55] J. C. F. Telles, "A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals," *Int. J. Numer. Methods Eng.*, vol. 24, pp. 959–973, 1987.

[56] L. C. Wrobel, *The Boundary Element Method, Volume 1: Applications in Thermo-Fluids and Acoustics*. WILEY, 2002.

[57] G. Beer, B. Marussig, and C. Duenser, *The Isogeometric Boundary Element Method*. Springer, 2020.

[58] Pozrikidis, *A Practical Guide to Boundary Elements Methods with the Software Library BEMLIB*. CRC Press, 2002.

[59] Y. P. Gong and C. Y. Dong, "An isogeometric boundary element method using adaptive integral method for 3D potential problems," *J. Comput. Appl. Math.*, vol. 319, pp. 141–158, 2017.

[60] F. Tan, J. Lv, Y. Jiao, J. Liang, and S. Zhou, "Efficient evaluation of weakly singular integrals with Duffy-distance transformation in 3D BEM," *Eng. Anal. Bound. Elem.*, vol. 104, pp. 63–70, 2019.

[61] Y. Gong, C. Dong, and X. Qin, "An isogeometric boundary element method for three dimensional potential problems," *J. Comput. Appl. Math*, vol. 313, pp. 454–468, 2017.

[62] M. Harmel and R. Sauer, "New hybrid quadrature schemes for weakly singular kernels applied to isogeometric boundary elements for 3D Stokes flow," *Eng. Anal. Bound. Elem.*, vol. 153, pp. 172–200, 2023.

[63] M. Spiga and G. L. Morino, "A symmetric solution for velocity profile in laminar flow through rectangular ducts," *Int. Commun. Heat Mass Transf*, vol. 21, pp. 469–475, 1994.

[64] L. Gaul, M. Kögl, and M. Wagner, *Boundary Element Methods for Engineers and Scientists - An Introductory Course with Advanced Topics*. Springer, 2003.

[65] B. Marussig, J. Zechner, G. Beer, and T. Fries, "Fast isogeometric boundary element method based on independent field approximation," *Comput. Methods Appl. Mech. Eng.*, vol. 284, pp. 458–488, 2015.

[66] C. Pozrikidis, *Introduction to Theoretical and Computational Fluid Dynamics*. Oxford University Press, 2011.

[67] X. Sun and X. Li, "A spectrally accurate boundary integral method for interfacial velocities in two-dimensional Stokes flow," *Commun. Comput. Phys.*, vol. 8, pp. 993–946, 2010.

[68] M. Loewenberg and E. Hinch, "Collision of two deformable drops in shear flow," *J. Fluid Mech.*, vol. 338, pp. 299–315, 1997.

[69] A. Rahimian, S. K. Veerapaneni, D. Zorin, and G. Biros, "Boundary integral method for the flow of vesicles with viscosity contrast in three dimensions," *J. Comput. Phys.*, vol. 298, pp. 766–786, 2015.

[70] S. K. Veerapaneni, A. Rahimian, G. Biros, and D. Zorin, "A fast algorithm for simulating vesicle flows in three dimensions," *J. Comput. Phys.*, vol. 230, pp. 5610–5634, 2011.

[71] R. Cox, "The deformation of a drop in a general time-dependent fluid flow," *J. Fluid Mech.*, vol. 37, pp. 601–623, 1969.

[72] I. B. Bazhlekov, P. D. Anderson, and H. E. Meijer, "Nonsingular boundary integral method for deformable drops in viscous flows," *Phys. Fluids*, vol. 16, no. 4, pp. 1064–1081, 2004.

[73] D. Barthes-Biesel and A. Acrivos, "Deformation and burst of a liquid droplet freely suspended in a linear shear field," *J. Fluid Mech.*, vol. 61, no. 1, pp. 1–22, 1973.

[74] A. J. Goldman, R. G. Cox, and H. Brenner, "Slow viscous motion of a sphere parallel to a plane wall—i motion through a quiescent fluid," *Chem. Eng. Sci.*, vol. 22, no. 4, pp. 637–651, 1967.

[75] S. Haber and G. Hetsroni, "The dynamics of a deformable drop suspended in an unbounded stokes flow," *J. Fluid Mech.*, vol. 49, no. 2, pp. 257–277, 1971.

[76] P. R. Wohl and S. Rubinow, "The transverse force on a drop in an unbounded parabolic flow," *J. Fluid Mech.*, vol. 62, no. 1, pp. 185–207, 1974.

[77] J. A. Hanna and P. M. Vlahovska, "Surfactant-induced migration of a spherical drop in stokes flow," *Phys. Fluids*, vol. 22, no. 1, 2010.