A MULTI-DOMAIN DIRECT BOUNDARY ELEMENT FORMULATION FOR PARTICULATE FLOW IN MICROCHANNELS

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

Alper Topuz

August 2021

A multi-domain direct boundary element formulation for particulate flow in microchannels
By Alper Topuz
August 2021

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)

Mehmet Şahin

Ali Karakuş

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

A MULTI-DOMAIN DIRECT BOUNDARY ELEMENT FORMULATION FOR PARTICULATE FLOW IN MICROCHANNELS

Alper Topuz
M.S. in Mechanical Engineering
Advisor: Barbaros Çetin
August 2021

In the present study, a multi-domain boundary element formulation is developed for high surface-area-to-volume ratio problems (i.e. particulate flow in high aspect ratio microfluidic channels, in a porous medium or in microfluidic devices with repetitive structures). The solution domain is decomposed into subdomains and the variable condensation technique is implemented. The solution matrices are built for each subdomain, and the matrices are updated at each time step only for the subdomains in which the particles move at each time step. Ghost domains, which are fictitious domains encapsulating the interfaces between the subdomains, are also introduced in the formulation to treat the particles crossing the interfaces between the subdomains. The formulation reveals that the computation of the subdomain matrices is further simplified for solution domains composed of periodic structures. The results of our study revealed that speed-up values as high as 50 for 2D structures and 80 for 3D structures is achievable with the current formulation.

Keywords: BEM, multi-domian, particulate flow, particle tracking.

ÖZET

MİKROKANALLARDAKI PARÇACIK AKIŞI İÇİN ÇOK ALANLI DİREKT SINIR ELEMAN FORMÜLASYONU

Alper Topuz
Makine Mühendisliği, Yüksek Lisans
Tez Danışmanı: Barbaros Çetin
Ağustos 2021

Bu çalışmada, yüksek yüzey alanı-hacim oranına sahip mikrokanallardaki parçacık hareketinin simüle edilebilmesi için çok alanlı bir sınır elemanı formülasyonu geliştirilmiştir (yüksek en-boy oranlı mikroakışkan kanallarda, gözenekli bir ortamda veya tekrarlayan yapılara sahip mikroakışkan cihazlarda partikül akışı). Çözüm alanı alt alanlara ayrıştırılmıştır ve değişken yoğunlaştırma tekniği uygulanarak çözüm matrisleri elde edilmiştir. Alt alanlar arasındaki arayüzleri içine alan hayali alanlar, arayüzlerden parçacıkların geçmesi için formülasyona dahil edilmiştir.Formülasyon, alt alan matrislerinin hesaplanmasının, periyodik yapılardan oluşan çözüm alanları için daha da basitleştirildiğini ortaya koymaktadır. Mevcut formülasyonla 2B yapılar için 50 ve 3B yapılar için 80 gibi yüksek hızlanma değerlerine ulaşılabileceği gösterilmiştir.

Anahtar sözcükler: SEY,çoklu alan,parçacık akışı, parçacık takibi.

To my family

Acknowledgement

I would like to offer my special thanks to Dr. Barbaros Çetin for his useful discussions. By taking me to conferences and academic meetings, he enabled me to meet with people from industry and academy. I would also like to thank Dr. Besim Baranoğlu for his advises on integral equations. I would like to express the deepest appreciation to all faculty members of mechanical engineering department because of their supports and suggestions. I would like to thank our department administrative staff Ms. Ela Baycan for behaving friendly manner and helping me in administrative all aspects.

I would like to thank to past and current members of Microfluidics and Lab-on-a-Chip Research Group (MLRG), specifically to Oğuz Altunkaş because of his irreplaceable friendship and supports, Atakan Atay for being an amazing colleague and Büşra Sarıarslan for her friendship. From past members I especially would like to thank Dilara Uslu for her assistance and support.

I would like to thank my colleagues form the graduate office; Arda Seçme and Tamer Taşkıran for making my life more enjoyable and boosting my morale. A very special thanks and gratitude belongs to Burak Sarı because of being such a perfect friend, he supported me from the beginning of my undergraduate studies. Furthermore, I would like to thank Tufan Erdoğan, Osman Berkay Şahinoğlu, Berke Demiralp, Batuhan Emre Kaynak, Emre Eraslan, Uzay Tefek, Berk Küçükoğlu, Mehmet Kelleci for their friendships.

Finally, I owe my deepest gratitude to my family for supporting me whenever I need, being with me all the time and their everlasting contributions and sacrifices.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent progress in micro and nano manufacturing technologies has been enabled researchers to design complex Lab-on-a-Chip (LOC) devices that fulfil complicated physical and chemical processes at micron and/or sub-micron length scales. When aforementioned operations are combined, a LOC device can be used for many biological and chemical applications such as single cell assays, drug and particle synthesis, molecule diagnosis, sorting, filtering, etc. The miniaturization of these complex operations brings many advantages such as high throughput and sensitivity levels in return to low costs. When compared with its room-sized counterparts, LOC devices are easily integrable to other LOCs or external systems, wearable, compact and require small amount of samples with increased reaction times. In order to perform the assigned task a LOC device may host sensors and actuators as well as microchannels with complex geometries. The geometric dimensions of the microchannel network, layout of the sensors and actuators, their working parameters are subject of design process and simulation methods offer an inexpensive way to ensure that they are functioning properly and carrying out their duties in an efficient manner [2].

Fluid inside a typical LOC device is laminar though counter-examples in which flow is turbulent can be found in the literature [3] yet they are not common. The physical nature of flow is characterized by the Reynolds number (*i.e.* ratio of inertial over viscous forces) which is typically very small $Re << 1$ for microflows and such flows are in the scope of this thesis. At that level of Reynolds number fluid flow is governed by Stokes' equations and flow is termed as creeping. The fluid itself can involve particles of various kinds (*e.g.* active, soft) and shapes at different length scales (*e.g.* μm to sub-μm) and these particles can be subjected to different kind of external agencies (*e.g.* electric, thermal, chemical). The particulate flow nature of the fluid further makes the problem multiphase. Such a setting becomes relatively complex and difficult to simulate. However, under different assumptions various modelling methodologies over years have been developed for particulate flow problems at low Reynolds numbers and they can be divided into two broad categories.

- Point Particle Approach
- Finite Sized Particle Approach

The former method assumes particles are point like and hence do not disturb the external field, only the effect of external field is felt by the particle. The field variables are obtained neglecting the existence of particles [4]. Hence for particle-particle (p-p) and particle-wall (p-w) interactions a correction factor has to be incorporated. In addition, particles with different geometries have no difference and their rotational motion is disregarded due to point particle assumption [5,6]. One major advantage of point particle approach is that it does not require high computational cost. Once the field variables are obtained without the presence of particles, particle trajectories can be obtained at the post-processing step using the correlations that return the forces acting on the particles as a function of field variables. This method is valid if the particle concentration is low (*i.e.* dilute solution). Therefore, Lagrangian Tracking Method (LTM) can simulate many particles by ignoring any particle-particle interactions, but allowing a statistical analysis [7–11]. Despite its limitations, the method has been implemented for the simulation of particle motion in the literature for electrokinetic [10, 12–14], acoustophoretic [7–9] as well as inertial microfluidic applications [11]. In the stress tensor approach, the size effect of particles are incorporated and the field variables are solved with the presence of finite-sized particles. The resultant force on the particles can be obtained by integrating the appropriate stress tensor on the particles' surface.

On the finite sized particle side analytical, semi-analytical and numerical methods are available. Each of them has advantages and disadvantages over each other. The analytical methods are only valid for simple set of geometries that can be expressed exactly on a specific type of coordinate system. Typically these methods are based upon bi-polar representation of particle-particle and particle-wall geometries and a very early example of such treatment can be found in the seminal paper of Stokes on the drag force of a stagnant sphere inside a infinite flow domain [15]. Combining bi-polar representation with the method of reflections, hydrodynamic particle-wall [16–18] and particle-particle interactions can be obtained [1, 19, 20]. This analysis can be further extended to include potential fields [21–23] as well. One another major drawback of these methods arises when more than two particles are present since the problem at hand becomes a many body problem. These problems are very challenging to solve with conventional analytical methods.

In order to capture the hydrodynamics of many body problems Stokesian dynamics is a viable tool. This method relates the forces and velocities by resistance and mobility tensors. These tensors are constructed by expanding the corresponding integral form of the governing equation around particle centres. This method successfully captures short range lubrication forces as well as long range hydrodynamic forces [24, 25]. Though its is an inexpensive tool for the simulation of many particles, shapes of which are limited to simple geometries. Furthermore simulations in case of confined geometries are also limited to simple configurations [26].

The hydrodynamics of particles as well as their interactions with other fields can be captured via molecular dynamics (MD) simulations yet for a typical particulate flow simulation the computational cost is huge. At this point mesoscale methods comes into picture as a simulation tool which can capture both microscale and macroscale effects. When compared with MD they can capture relatively longer length and time scales. Two common examples are Lattice Boltzmann Method (LBM) and Dissipative Particle Dynamics (DPD). LBM solves macroscopic momentum and continuity equation over a lattice and implement the macroscopic boundary conditions via numerical algorithms. However the aforementioned algorithms are many and become advantageous over each other depending upon the characteristics of flow. Unlike Stokesian dynamics velocity-force coupling over particle surface may require iterations and complicated algorithms [27]. Similar to LBM, DPD is another mesoscale method but unlike LBM it is not connected to macroscale equations rather it is a coarse grained version of MD. Particles are modelled as a collection of molecules and they interact with each other in terms of random, conservative and dissipative forces yet the parameters that adjust these forces are not easy to obtain [28, 29]. The mesoscale methods overall can capture both microscale and macroscale at long run relatively to MD but still they cannot overwhelm the length and time scales of continuum methods. With a few exceptions, continuum modelling can capture vast majority of physics that occurs in a typical LOC device.

On the top of mesoscale approaches, continuum modelling lies and it can be divided into two broad categories; mesh methods and mesh-free methods. Mesh methods require a network of connected nodes inside the solution domain and on the boundary surrounding it. Either differential or integral form of the governing equations of the fluid flow (*i.e.* momentum and continuity equation) are discretized using the network and resulting system of linear equations is solved to obtain the corresponding field variables at the nodal points. However, the flows with solid and/or deformable objects which fall under the moving boundary problems constitutes a challenging problem for mesh methods. In such a setting, the methods that require meshing of solution domain suffer from major drawbacks. The utmost one is the remeshing problem. As boundaries move the domain mesh has to be deformed/remeshed according to the new configuration which is quite burdensome even for simple particulate flow problems. In a typical fluid-solid interaction problem two type of mesh networks comes into the picture; Eulerian and Lagrangian. An Eulerian mesh is static and no remeshing is required. Objects inside the fluid moves through the static mesh and this kind of mesh convenient for fluids. In Lagrangian approach, mesh deforms with the moving particles and remeshing is required, this kind of mesh is appropriate for solids rather than fluids. If the selection between mesh types arbitrary then the method is called as Arbitrary Lagrangian Eulerian (ALE) Method. Typically, particulate flow problems are handled via ALE methods instead of using pure Lagrangian or Eulerian meshes [30, 31]. In a conventional ALE setting, first the particle boundaries is meshed then the mesh network on the surface of particles is extended into the flow domain. With structured and unstructured grids conventional mesh methods such as Finite Element Method (FEM) or Finite Volume Method (FVM) can be used to solve governing equations. Upon obtaining field variables and combining them with rigid body motion conditions next configuration of particles is obtained. For large numbers of objects this method suffers from continuous remeshing and/or interpolation of data from previous configuration to the next one [4,32,33]. Moreover, in case of arbitrarily large deformations meshing becomes problematic and arbitrary movements of particles can create complex structures hence require complicated mesh networks.

Immersed Boundary Method (IBM) circumvents this drawback by defining a simple Cartesian mesh around the particle that do not exactly conforms the local particle geometry. Typically, a Cartesian mesh around the particle is sufficient and such a setting makes easier to resolve the hydrodynamics of the fluid. This meshing strategy is now relatively easier to implement and can accommodate arbitrarily large deformations unlike ALE. However, since the mesh is not conforming to the particle geometry, governing equations must be modified in order compensate the change in boundary conditions since they are not satisfied exactly. This can be achieved by forcing functions and depend upon the type of boundary condition itself hence overall require complex algorithms to fully satisfy them. [33,34]. One another option is Ficticious Domain Method (FDM) that transforms actual domain into geometrically simpler fictitious one. This domain is now easy to mesh and depending upon the case, becomes static even the actual domain is dynamic. The boundary conditions are again enforced by means of constraints. Consequently, the methods that require meshing of solution domain are powerful in the solution of linear and non-linear PDEs. However, when a moving boundary is considered meshing becomes challenging. Additional strategies have to be implemented in order to incorporate the moving boundaries and apply the boundary conditions which does not circumvent the moving boundary problem completely.

In case of methods that only require a mesh on the boundary, moving body problems becomes easy to handle. There is no need to construct a mesh that conforms the motion and geometry of particle inside the solution domain. One such method is the Boundary Element Method (BEM) that can be considered as a general numerical tool to solve the boundary integral equations. The main idea of the method is to convert the partial differential equation that governs the physics of the problem to its boundary integral representation in which singularity solutions or in other words free space Green's function of corresponding partial differential equation is used as test functions. The domain integrals are satisfied exactly hence they do not appear in the method only the boundary integrals are numerically solved hence gives better approximations over methods require a domain mesh. The boundary mesh then can be translated and rotated according to the instantaneous motion of the particles. Moreover, since the geometric representation of the particles are achieved accurately by boundary elements, there is no restriction on the shape of the particles. These features of BEM circumvent the mesh problem hence no special treatment is required. Furthermore, both the primary and secondary variables appear in the same integral equation and thus they are approximated to the same accuracy level. This representation is then discretized in a proper way in order to obtain a system of linear equations which will be solved for unknown boundary values. The primary and secondary variables inside the solution domain is determined analytically therefore makes the method semi-analytic [35, 36]. Alternative to the numerical models based on domain discretization, boundary only discretization nature of the BEM provides a significant reduction in the number of unknowns for the problems with low surface-area-to-volume ratio or for the problems with unbounded domain. It is also convenient to handle external problems with BEM where solution domain tends to an infinite extent. The corresponding decaying conditions peculiar to external problems are automatically satisfied hence there is no need to define artificial boundary conditions or additional constraints. However, BEM is only limited to problems that are governed by linear partial differential equations. A non-linearity in the governing equation or in boundary conditions requires special treatment hence the method becomes unfeasible to use that is in order to avoid the non-linearity a domain mesh might be required and thus method loses its advantages. Furthermore, the system of linear equations after the discretization is fully populated with high-condition numbers unlike methods that require a domain mesh [37–39]. Another drawback of the method arises in case of particulate flow problems. On the top of BEM discretization the implementation of special dynamic constraints over the motion of particles is required and in BEM framework several special methods exists. They offer advantages and disadvantages over each other and these will be discussed in the subsequent chapter.

Historically, the boundary element representation of creeping flow was first developed by Youngren and Acrivos [40] for the analysis of flow over passing 3D and axisymmetric rigid particles. Following this study, Youngren and Acrivos extended their work for similar flow over deformable bubbles [41], and later a similar study involving viscous drops was introduced by Rallison and Acrivos [42]. BEM is a viable option for the simulation of many particles immersed in fluids in the Stoke's flow regime which was vital to predict the macroscopic and rheological properties of suspensions (*i.e.* predicting the sedimentation rate and relative viscosities of suspensions) [43–45].

1.1 Motivation and Purpose of the Thesis

Overall, BEM handles the mesh problem but brings two major disadvantages; first being its limitation to linear problems and second being the fully populated system of equations. Fortunately, the particulate flows that take place in LOC devices is often governed by linear differential equations due to slowness of fluid flow. Moreover, the additional physical phenomenon that is used to manipulate the particle motion or any other process pertaining the particles is often governed by again linear differential equations [46]. The second problem has two consequences firstly the fully populated matrix indicates the problem at hand is an N body problem which is computationally expensive when compared with its counterparts such as FEM or FVM, second this N body problem returns a matrix with high condition number.

Considering the first consequence, when solved with BEM a typical Stokes flow problem can be considered as a N (*i.e.* the number of degrees of freedom) body problem which refers to evaluation of N^2 layer potentials and requires storage of a fully populated $\lambda N \times \lambda N$ matrix where λ is 2 for two dimensional problems and 3 for three dimensional problems and due to high condition number it has to be solved by a direct method (standard iterative methods fail to converge to the true solution) which has a arithmetic complexity of $(\lambda N)^3$. Note that there are other methods that returns a more stable system of linear equations hence they can be solved iteratively. They are discussed in the subsequent chapters. In this thesis a direct BEM is used thus the matrices are ill-conditioned. In particular to particulate flow problems this matrix construction and solution process is repeated many times to obtain the particle velocities and hence the trajectories. This computational burden restricts the simulation of large suspensions. As an example, in order to capture the rigid body velocity distribution of a typical suspension of 10^3 particles on Intel Paragon parallel cluster using 1844 processors can take a computation time of around 20 minutes in two dimensions [47, 48]. On the top of trajectories of particles the local fields around the particles might be of interest which again constitutes another N body problem. In order to circumvent the computational problem researchers resorted to multipole methods which is based upon power series expansion of Green's function. Notice that corresponding Green's functions of the differential equation is subjected to decay condition at infinity. Hence in the framework of collocation BEM if the distance between the source and evaluation point is large then the boundary integrals return values close to zero. The far-field contribution from distinct poles then can be neglected by only considering the near-field effects and clustering the far-field contribution to a single pole. This famous algorithm first introduced in [49] which reduces the computational requirement that is both for memory and layer computations from N^2 to N which is a considerable reduction but in the cost of truncation error that appear in the power series expansions. This expansion process furthermore is complicated for example in case of singularity solutions of Stoke's equation expansion in terms of harmonics as well as expansion in terms of complex variables are available [50, 51]. The clustering of far-field poles into one cluster further requires search algorithms (*e.g.* tree search) and different selection of poles results in different scale of errors [52]. The multipole methods on the top of standard BEM formulation are very fast but they bring additional complexities and approximation errors. One other option is the multi-domain methods which relies on partitioning the solution domain into subdomains. The method first introduced to handle piecewise homogenous material property distribution over the solution domain and then extended to large scale domains in order to decrease the memory and computation costs. When compared with multipole methods it is slower but easier to implement with higher accuracy. However, in the literature there is no application to multi-domain boundary element method for particulate flow problems which is the main theme of this thesis.

1.2 Thesis Outline

This thesis involves following chapters:

Chapter 1 is the introductory chapter to microfludics and Lab-on-a-Chip technologies. It briefly introduces the simulation techniques to particulate flow problems that are common in applications in microfluidics. There is a special emphasis on boundary integral methods and its variants that is used to simulate particle laden flows. The motivation of this work is also presented in this chapter.

Chapter 2 introduces the standard boundary element formulation and its mathematical basis. Special attention given on the BEM applications on particulate flows. In the parts that follow mathematical background of the proposed methodology is discussed in detail.

Chapter 3 gives the results of numerical tests that is made to test the performance of the method. Two and three dimensional problems with different geometries are used. Two main parameters are varied to asses the corresponding performance of the proposed method first being the number of particles and second being the number of elements. The performance is measured according to speed-up with respect to single domain computation time. A discussion is separately made on the ghost domain topology since it severely effects the performance of the method. This chapter is concluded with a brief discussion on condition numbers.

Chapter 4 is the summary of the findings of this thesis and comments on future directions of research.

Chapter 2

Boundary Element Formulation

It is essential to resort a number of simplifying assumptions in order to construct amenable mathematical models of particle-laden flow systems. In this chapter, some basic definitions pertaining to BEM, proposed formulation together with aforementioned assumptions will be addressed. In this thesis the fluid is considered to be Newtonian and incompressible. The Reynolds number is small hence the flow is creeping (i.e. $Re << 1$) and therefore the governing equations of the fluid flow are Stoke's equations.

$$-\nabla P(\mathbf{x}) + \mu\nabla^2\mathbf{u}(\mathbf{x}) + \rho\mathbf{g} = 0 \tag{2.0.1}$$

$$\nabla \cdot \mathbf{u}(\mathbf{x}) = 0 \tag{2.0.2}$$

$$\tag{2.0.3}$$

where P is the pressure, **u** is the velocity vector, **g** is the gravitational acceleration, μ and ρ are viscosity and density respectively. σ is the hydrodynamic stress tensor given by $\sigma = -\mathbf{I}p + 2\mu\mathbf{E}$ where **E** is strain rate tensor. Hydrodynamic traction is defined by Cauchy's lemma $\mathbf{t} = \sigma(P, \mathbf{u}) \cdot \mathbf{n}$ where **n** is the unit outward normal vector. Equation 2.0.1 together with 2.0.2 form a set of linear partial differential equation of second order and analytical solutions are available only for small set of simple geometries and boundary conditions.

2.1 Formulation for Stoke's Flow

Free space Green's function to corresponding set of equations plays an important role in the derivation of Boundary Integral Equation (BIE) of the Stoke's flow. The singularity solutions to above set of equations are given by point forcing term (*i.e.* Dirac delta function) as source and corresponding decay conditions at infinity

$$-\nabla P(\mathbf{x}) + \mu \nabla^2 \mathbf{u}(\mathbf{x}) = -\mathbf{g}\delta(\mathbf{x} - \mathbf{x}_0) \tag{2.1.1}$$

$$\nabla \cdot \mathbf{u}(\mathbf{x}) = 0 \tag{2.1.2}$$

$$\lim_{x\to\infty} \mathbf{u} = \mathbf{0} \quad \lim_{x\to\infty} P = 0 \tag{2.1.3}$$

An excellent review about solution techniques related to singularity solutions of equation set above and important properties of Green's functions are given in [53] hence details are omitted here. The corresponding three and two dimensional Green's functions of Stoke's equations are given below.

$$\mathbf{G}(\mathbf{x}_0, \mathbf{x}) = \frac{1}{8\pi\mu r}\left(\mathbf{I} + \frac{\mathbf{r}\otimes\mathbf{r}}{r^2}\right) \quad \text{(3D)} \tag{2.1.4}$$

$$\mathbf{G}(\mathbf{x}_0, \mathbf{x}) = \frac{1}{2\pi\mu}\left(-\mathbf{I}\log r + \frac{\mathbf{r}\otimes\mathbf{r}}{r^2}\right) \quad \text{(2D)} \tag{2.1.5}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{x}_0$, $r = \sqrt{\mathbf{r}\cdot\mathbf{r}}$ and $\mathbf{I} = \delta_{ij}\left(\mathbf{e}_i \otimes \mathbf{e}_j\right)$. Here r is the Euclidian distance between source point $\mathbf{x}_0$ and field point $\mathbf{x}$, μ is the viscosity of corresponding fluid and $\mathbf{I}$ is the identity tensor.

Given two distinct Newtonian fluids with velocity fields $\mathbf{u}$ and $\mathbf{u}^{'}$, pressure fields P and $P^{'}$ that form corresponding hydrodynamic stress tensors σ and $\sigma^{'}$. Using the Green's second identity one gets.

$$\mathbf{u}^{'}\nabla.\sigma - \mathbf{u}\nabla.\sigma^{'} = \nabla.\left(\mathbf{u}^{'}\sigma - \mathbf{u}\sigma^{'}\right) \tag{2.1.6}$$

Note that the stress tensors satisfy the Stoke's equations (*i.e.* $\nabla.\sigma = 0$), then the

Lorentz reciprocal relation between two distinct fluids is given by

$$\nabla.\left(\mathbf{u}^{'}\sigma - \mathbf{u}\sigma^{'}\right) = 0 \tag{2.1.7}$$

Now if one lets equation 2.1.7 is satisfied for an arbitrary Stoke's flow and a Stoke's flow driven by a point source and uses the divergence theorem with integration by parts can obtain BIE for Stoke's flow. The details of which can be found in [35, 37, 38]. The limiting process that gives the right hand side of 2.1.8 is also omitted the details can be found in aforementioned references.

$$\fint_{\Gamma} \mathbf{G}(\mathbf{x}_0, \mathbf{x})\mathbf{t}(\mathbf{x})d\Gamma - \mathop{=\!\!\!\!\!\!\!\int}_{\Gamma} \left(\mathbf{n}\cdot\nabla\mathbf{G}(\mathbf{x}_0, \mathbf{x})\right)\mathbf{u}(\mathbf{x})d\Gamma = \begin{cases} \mathbf{u}(\mathbf{x}_0) & \mathbf{x}_0 \in \Omega \\ \frac{1}{2}\mathbf{u}(\mathbf{x}_0) & \mathbf{x}_0 \in \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{2.1.8}$$

Note that a derivation based upon direct method was given here which refers deriving BIE constructed upon physical quantities that are related to governing equations. There is also an indirect method which is much more general and can be extended to potential, acoustic and linear elasticity problems as well. The method relies upon constructing the Clarendon projector of a general elliptic boundary value problem such projector involves double and single layer potentials of several kinds. Equation 2.1.8 can be obtained via the aforementioned method as well [36, 38, 54]

The first integral on the left hand side of equation 2.1.8 is weakly singular whereas the second integral is strongly singular due to the nature of free space Green's function. Weak singularity is order of $\ln(r)$ for two dimensional problems and $1/r$ for three dimensional problems. Strong singularity is order of $1/r$ and $1/r^2$ for two and three dimensional problems respectively. The singularities are indicated by single and double dash symbols on the integral signs in equation 2.1.8. Special treatment is required in order to evaluate those integrals and these treatments are rigorous hence they are omitted here but can be found in standard textbooks [35].

The underlying idea of BEM is to obtain the complete Cauchy data (*i.e.* the velocity and traction) on the boundary using equation 2.1.8. Once this data is obtained, the velocity field inside the solution domain can be found again using equation 2.1.8 itself. If pressure is of interest there is a corresponding pressure integral equation given with its own pressure singularity solution. The derivation of which again can be found in [53].

In case of a arbitrary Stoke's flow problem in which there are no particles either velocity or traction is known on the boundary hence equation 2.1.8 can be used to obtain unknown velocity and traction distributions. If only the traction distribution over the boundary is given the problem at hand becomes a pure Neumann problem and further constraints has to implemented in order to obtain corresponding velocity field. In case of a Robin problem a special treatment on the boundary conditions is required but it is straightforward [38, 54].

In case of a Stoke's flow with particles one seeks a solution of either resistance or mobility problem. In the former one, velocity distribution over the particle boundary or the rigid body velocities of the particle are given and the traction distribution and/or rigid body forces and torques are solved for. In the later one the traction distribution or the rigid body forces and torques are given and corresponding velocity distribution and/or rigid body velocities are solved for. If either of traction or velocity distribution are given then 2.1.8 can be used to obtain the unknown field. Furthermore, taking the derivative of equation 2.1.8 which is hypersingular one can obtain a BIE for traction field and In case of constraints over tractions and velocities both equations can be used together. Velocity and traction equations can be combined to satisfy for example power constraints that are typical for active particles [55, 56] .

However, given rigid body forces or velocities equation 2.1.8 does not involve required information to enforce following force and/or velocity constraints. They have to be implemented separately.

$$\mathbf{F^B} = \int_\Gamma \mathbf{t}(\mathbf{x}) d\Gamma \tag{2.1.9a}$$

$$\mathbf{T^B} = \int_\Gamma (\mathbf{x} - \mathbf{x}_c) \times \mathbf{t}(\mathbf{x}) d\Gamma \tag{2.1.9b}$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{u^B} + \omega^{\mathbf{B}} \times (\mathbf{x} - \mathbf{x}_c) \tag{2.1.9c}$$

Where $\mathbf{F^B}$ is the rigid body force, $\mathbf{T^B}$ is the rigid body torque around point $\mathbf{x}_c$, $\mathbf{u^B}$ and $\omega^{\mathbf{B}}$ is the translational and angular velocity of particle.

The implementation of rigid body motion conditions can be cumbersome. There are rigorous methods in the literature exist based upon direct [43–45,57–60] or indirect [61–67] BIE formulations. In this thesis a direct boundary element formulation, which included condensation process for the particle nodes imposing rigid body motion to obtain only the particles' trajectories and avoid the calculation of the element interactions on non-moving boundaries at each time-step, has developed to simulate the particle motion inside microfluidic channels [48]. The main disadvantage of this method in case of mobility problems equation 2.1.8 reduces to a Fredholm integral equation of first kind which is ill-conditioned , hence direct solution methods such as Gauss elimination needs to be implemented. The situation may worsen for the case of particles with large aspect ratio and/or particles are in close proximity of each other or with the confinement. However, poor conditioning can be circumvented by the application of higher-order elements and accurate calculation of the singular integrals [68]. Furthermore when compared with indirect methods this method is computationally expensive [36, 69]. Alternative to direct methods Completed Double Layer Boundary Integral Method (CDLBIE) can be used to implement the above constraints in which the resulting equations are Fredholm integral equations of second kind. The resulting system of equations are well-posed, hence the solution is stable and iterative solvers with pre-conditioners may be implemented. A major difficulty with the indirect methods is the problem specific formulation and problem specific iterative solution of the problem [67]. Moreover, the indirect formulation may require problem specific additional mathematical derivations for the determination of an external force (*e.g.* electric, magnetic and/or acoustic) acting on the particle where as the implementation of the direct formulation is straightforward for any other physics.

In order to use BIE for an arbitrary Stoke's flow problem the discretization of equation 2.1.8 is required. The discretized integral equation then transformed into a system of linear equations solution to which gives the Cauchy data of the problem under consideration. The discretization process takes place both for geometric and field variables (*i.e.* velocity and traction field) and various interpolation schemes of different order can be utilized. The order and type of approximation is problem specific. In order to avoid multiple definition of field variables discontinuous type of approximation is preferred over continuous approximation for the field variables, whereas geometric variables are always interpolated continuously. The generic approximation scheme both for 2D and three 3D problems are given below. Note that the approximation functions do not live in global coordinate system where physical system lies, instead in order to handle integrals numerically in a straightforward manner they are defined in local coordinate system.

$$\mathbf{x} \approx \sum_{i=1}^{N_d^g} N_i(\xi)\tilde{\mathbf{x}} \tag{2.1.10}$$

$$\mathbf{u} \approx \sum_{j=1}^{N_d^f} N_j(\xi)\tilde{\mathbf{u}} \tag{2.1.11}$$

$$\mathbf{t} \approx \sum_{k=1}^{N_d^f} N_k(\xi)\tilde{\mathbf{t}} \tag{2.1.12}$$

where N_d^g number of degrees of freedom to interpolate geometric variables and N_d^f is the degrees of freedom used to interpolate field variables. $\tilde{\mathbf{u}}$, $\tilde{\mathbf{t}}$ are the values of velocity and traction at colocation points and $\tilde{\mathbf{x}}$ is the coordinates obtained from the mesh network that is used to approximate the solid body. Plugging in the above set of equations to BIE one gets:

$$\begin{aligned}&\sum_{i=1}^{N_e}\int_{\Gamma_i}\mathbf{G}(\mathbf{x}_0,\mathbf{x})\sum_{k=1}^{N_d}(\tilde{\mathbf{t}}_k N_k(\xi))J(\xi)d\Gamma\\&-\sum_{i=1}^{N_e}\int_{\Gamma_i}(\mathbf{n}\cdot\nabla\mathbf{G}(\mathbf{x}_0,\mathbf{x}))\sum_{j=1}^{N_d}(\tilde{\mathbf{u}}_j N_j(\xi))J(\xi)d\Gamma=\frac{1}{2}\tilde{\mathbf{u}}(\mathbf{x}_0)\end{aligned}\tag{2.1.13}$$

$$\begin{aligned}&\sum_{i=1}^{N_e}\sum_{j=1}^{N_d}\tilde{\mathbf{t}}_k\int_{\Gamma_i}\mathbf{G}(\mathbf{x}_0,\mathbf{x})N_k(\xi)J(\xi)d\Gamma\\&-\sum_{i=1}^{N_e}\sum_{j=1}^{N_d}\tilde{\mathbf{u}}_j\int_{\Gamma_i}(\mathbf{n}\cdot\nabla\mathbf{G}(\mathbf{x}_0,\mathbf{x}))\,N_j(\xi)J(\xi)d\Gamma=\frac{1}{2}\tilde{\mathbf{u}}(\mathbf{x}_0)\end{aligned}\tag{2.1.14}$$

where $J(\xi)$ is the Jacobian of transformation which transforms an arbitrary surface in three dimension to standard triangle and an arbitrary curve in two dimension to standard line. The normal vector $\mathbf{n}$ can be obtained in terms of approximated geometric variables as

$$\mathbf{n}^{3D}=\left\{\begin{array}{c}\partial x/\partial\xi\\\partial y/\partial\xi\\\partial z/\partial\xi\end{array}\right\}\times\left\{\begin{array}{c}\partial x/\partial\eta\\\partial y/\partial\eta\\\partial z/\partial\eta\end{array}\right\}=\left\{\begin{array}{c}\partial y/\partial\xi\;\partial z/\partial\eta-\partial y/\partial\eta\;\partial z/\partial\xi\\\partial z/\partial\xi\;\partial x/\partial\eta-\partial z/\partial\eta\;\partial x/\partial\xi\\\partial x/\partial\xi\;\partial y/\partial\eta-\partial x/\partial\eta\;\partial y/\partial\xi\end{array}\right\}\tag{2.1.15}$$

$$\mathbf{n}^{2D}=\left\{\begin{array}{c}dx/d\xi\\dy/d\xi\\dz/d\xi\end{array}\right\}\times\left\{\begin{array}{c}0\\0\\1\end{array}\right\}=\left\{\begin{array}{l}dy/d\xi\\-dx/d\xi\\0\end{array}\right\}\tag{2.1.16}$$

The Jacobian of the transformation to local coordinate system in which the standard triangle and standard line lies given by the norm of the normal vectors.

$$J^{2D,3D}=\sqrt{n_x^2+n_y^2+n_z^2}\tag{2.1.17}$$

where the partial derivatives of space coordinates are defined as

$$\frac{\partial\mathbf{x}}{\partial\xi}=\sum_{n=1}^{N_d^g}\frac{\partial N_n}{\partial\xi}\tilde{\mathbf{x}}\tag{2.1.18}$$

$$\frac{\partial \mathbf{x}}{\partial \eta} = \sum_{n=1}^{N_d^g} \frac{\partial N_n}{\partial \eta} \tilde{\mathbf{x}} \tag{2.1.19}$$

Equation 2.1.14 is the discretized form of 2.1.8. Since all functions under integral sign are defined on local coordinate system an appropriate integration scheme must be used to obtain the system of linear equations yet it is not still that straightforward. Due to nature of the integral operators and kernel functions three types of integral arise these are regular, singular and nearly singular integrals. Singular integrals are further categorized as weakly and strongly singular integrals.

2.2 Subdomain Boundary Element Method

Standard single domain BEM can be considered as a N body problem which implies all points that are used to discretize the field variables have to interact with each other. Mathematically speaking, in BEM the image of kernel functions do not have compact support. This is for instance on the contrary of FEM in which the approximation function that lies under the integral sign have compact support which results in a banded system of linear equations. The advantages offered by BEM for particulate flows are hindered as the surface-area-to-volume ratio of the problem increases. Some examples of these kind of problems are the particle flow in high aspect ratio microchannels and/or particle flow in a porous medium, more specifically flow in direct lateral displacement (DLD) microfluidic devices. DLD devices utilizes the hydrodynamic interactions of particles with a series of periodic or non-periodic posts within a microchannel to achieve size based separation [46, 70, 71]. For these large scaled problems, direct boundary element formulation becomes unfeasible since the resulting system of equations are fully populated and have high condition numbers which makes the solution process computationally costly. In addition, a huge main memory allocation is required for the computation on large matrices [72–75]. One option to overcome these drawbacks is to renounce BEM and employ Stokesian dynamics—a semi-analytical method— which relies on constructing a resistance matrix for particle-obstacle systems to capture the hydrodynamic interactions. Stokesian dynamics become a powerful tool when the particles and obstacles posses spherical geometry and distance between the particles and obstacles are either large or small. Non-spherical geometries are not well suited for the method and hydrodynamic interactions of bodies can be resolved with an interpolation when the distance is mediocre and thus such a technique is not well-suited for DLD chip simulations [76].

In order to circumvent the disadvantages of single domain BEM, multi-domain boundary element method (MDBEM) has been proposed in the literature. Unlike single domain BEM, MDBEM decomposes the solution domain into number of subdomains, discretizes the boundary of each subdomain and finally either solves each local system of linear equations iteratively or assembles a global matrix by employing compatibility conditions at the interfaces. Since the resulting matrices are smaller than that of the single domain case, memory requirement, and hence the CPU time can be drastically reduced [77, 78]. Furthermore, the final assembly matrix is diagonally dominant, therefore the condition numbers of the resulting matrices significantly improve [74, 79, 80]. MDBEM was successfully employed for heat transfer [72,73,75,77,78,81–86], elasticity [74,87–89] and Stoke's flow [79, 80, 90–93] problems, and the studies reported significant improvements both in memory requirements and CPU time. In MDBEM, two common approaches were implemented in the solution of boundary element equations. Some studies employed iterative schemes [72, 73, 77–79, 82, 84–86, 90–93], and the others employed variable condensation technique [74, 75, 81, 83, 87–89] in which some variables are eliminated and the size of the final system of equations are reduced. The selection of iterative schemes or variable condensation technique is problem specific. For large scale problems, iterative techniques may suffer from convergence issues, and the fine tuning of convergence parameters as well as initial guesses may be required [73, 78]. Condensation techniques, on the other hand, are not easy to implement on computer when the solution domain is partitioned with unstructured subdomains. Microfluidic networks are slender structures with large aspect ratio (*i.e.* large length over width ratio). This slender geometry can be easily decomposed into consecutive structured subdomains. However, the convergence of the iterative methods may be problematic for the case of large numbers of consecutive structured subdomains since in most of the typical particulate flow problems boundaries are subjected to homogenous Dirichlet and Neumann boundary conditions (e.g. no slip and pressure outlet), and so the convergence of the iterative schemes may be severely affected [73]. Therefore, the topology with consecutive structured subdomains is more suitable for the condensation techniques rather than the iterative methods. Despite the aforementioned benefits of MDBEM, the implementation is not straightforward for particulate flow problems. In multi-domain formulation, subdomain boundaries are present within the solution domain. Since particles are modeled as finite sized objects, they cannot cross the interfaces between the subdomains. Furthermore as a particle approaches a subdomain interface near-singularities may occur. In the case of poor treatment of these integrals, the physics of the problem might be disrupted.

The schematic representation of how the formulation works is demonstrated in Fig. 2.1. The goal is to obtain the particle trajectories within a microfluidic channel as shown in Fig. 2.1-(E) with the red curve. As a general case, a 2D microchannel with number of random inclusions is presented. For the multi-domain formulation, the solution domain is divided into smaller subdomains (in the schematic figure the domain is divided into three pieces denoted by DI, DII and DIII). Once the particle comes close to the interface between the different subdomains, a ghost domain (denoted by GDI and GDII on the figure) is introduced. When the particle passes the interface and is sufficiently away from the interface, the ghost domain is removed from the solution domain. This process continues as the particle reaches the exit of the solution domain.

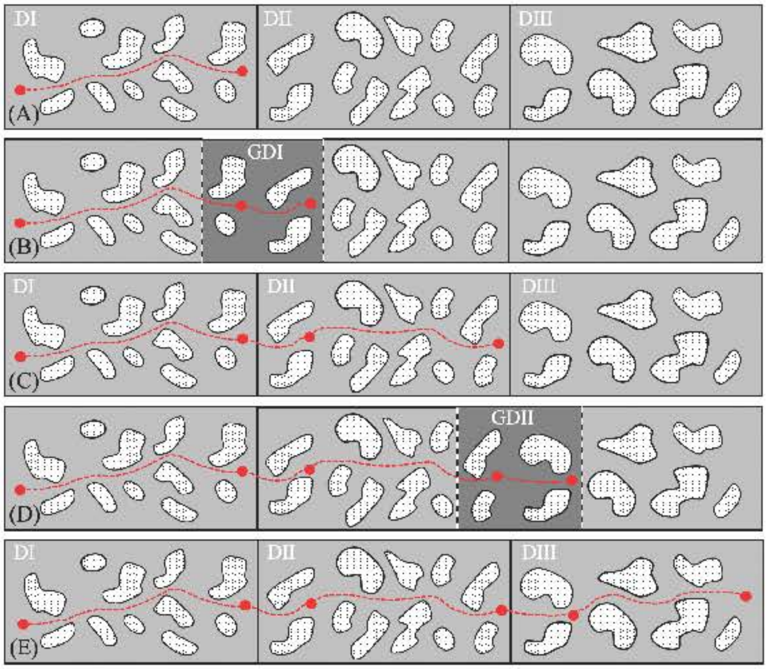

Figure 2.1: Schematic representation of the particle flow in the solution domain

2.2.1 Subdomain Level Equations

Depending upon the problem geometry, various partitioning schemes can be utilized to construct subdomains. For unstructured subdomain networks, when used with variable condensation techniques, a proper enumeration of the subdomains and subdomain locations significantly reduces the condition numbers by increasing the sparsity of the matrices [74, 79]. In addition, overlapping subdomains can be used to avoid singular and near-singular integrations [83]. In the present work, we consider a 2D microchannel with periodic and non-periodic inclusions. Schematic representation of our computational domain is shown in Fig. 2.2 (for the clarity of the figures the inclusions are presented on the figure). Since the channels considered are rectangular, (n) subdomains are placed in a consecutive and structured manner, which results in $(n-1)$ interfaces. Each subdomain $(\Omega^{(\alpha)})$ has two interfaces $(I_{\alpha-1}$ and $I_{\alpha})$ except the first and last subdomains as shown in the figure. The subdomains are numbered from left to right. Depending on the position of the particles, some subdomains may include many particles or a single particle, and some subdomains may be not include any particle at all. The

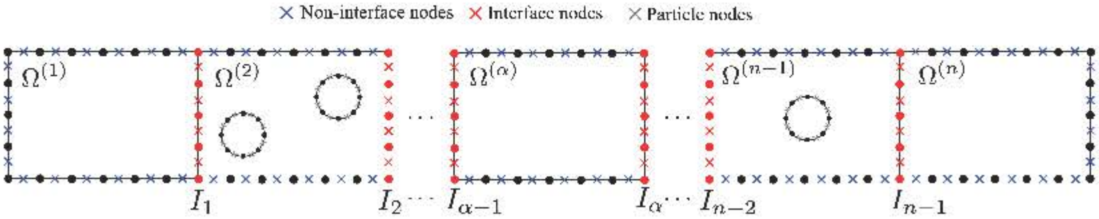

Figure 2.2: Schematics of the solution domain

boundary of each subdomain is discretized with constant elements. In order to implement multi-domain formulation, the nodes on the boundaries are classified as non-interface, interface and particle nodes following the notation in [35, 75]. Interface nodes are shared by two subdomains and non-interface nodes belong to only one subdomain. Particle nodes on which boundary conditions are not specified belong to the particles. Note that problem under consideration is a Stoke's problem, and no external source exists, therefore no need for any volume integral. In case of presence of volume integrals, there would be nodes inside the solution domain. Then, these nodes would be classified as internal nodes [75].

The condensation procedure takes place in two steps: (i) we first condense the non-interface nodes keeping the particle and interface nodes, and then (ii) we condensate the particle nodes imposing rigid body motion and compatibility conditions. We first construct the conventional BEM matrices for each subdomain. For subdomain-α, the subdomain level BEM matrices can be written as:

$$\begin{bmatrix} \mathbf{G}_{AA}^{(\alpha)} & \mathbf{G}_{AK}^{(\alpha)} \\ \mathbf{G}_{KA}^{(\alpha)} & \mathbf{G}_{KK}^{(\alpha)} \end{bmatrix} \begin{Bmatrix} \mathbf{t}_A^{(\alpha)} \\ \mathbf{t}_K^{(\alpha)} \end{Bmatrix} = \begin{bmatrix} \mathbf{H}_{AA}^{(\alpha)} & \mathbf{H}_{AK}^{(\alpha)} \\ \mathbf{H}_{KA}^{(\alpha)} & \mathbf{H}_{KK}^{(\alpha)} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_A^{(\alpha)} \\ \mathbf{u}_K^{(\alpha)} \end{Bmatrix} \tag{2.2.1}$$

where subscript A stands for non-interface nodes, and K for interface and particle nodes. Note that either the velocity or traction components of the non-interface nodes are known whereas neither the velocity nor traction components on the interface and particle nodes are known. Assuming all velocities are specified on the non-interface nodes, and all traction values are unknown without loss of generality, Eqn. (2.2.1) can be recast into:

$$\begin{aligned}\mathbf{G}_{AA}^{(\alpha)}\cdot\mathbf{t}_{A}^{(\alpha)}+\mathbf{G}_{AK}^{(\alpha)}\cdot\mathbf{t}_{K}^{(\alpha)} &= \mathbf{H}_{AA}^{(\alpha)}\cdot\mathbf{u}_{A}^{(\alpha)}+\mathbf{H}_{AK}^{(\alpha)}\cdot\mathbf{u}_{K}^{(\alpha)} \\ \mathbf{G}_{KA}^{(\alpha)}\cdot\mathbf{t}_{A}^{(\alpha)}+\mathbf{G}_{KK}^{(\alpha)}\cdot\mathbf{t}_{K}^{(\alpha)} &= \mathbf{H}_{KA}^{(\alpha)}\cdot\mathbf{u}_{A}^{(\alpha)}+\mathbf{H}_{KK}^{(\alpha)}\cdot\mathbf{u}_{K}^{(\alpha)}\end{aligned} \tag{2.2.2}$$

To proceed with the first step of the condensation process, $\mathbf{t}_{A}^{(\alpha)}$ can be solved using Eqn. (2.2.2) to obtain the interface equation (i.e. $\mathbf{t}_{K}^{(\alpha)}$) as:

$$\mathbf{t}_{K}^{(\alpha)}=\left(\boldsymbol{\Lambda}^{(\alpha)}\right)^{-1}\left(\boldsymbol{\Gamma}^{(\alpha)}\cdot\mathbf{u}_{K}^{(\alpha)}+\boldsymbol{\phi}^{(\alpha)}\right) \tag{2.2.3}$$

where

$$\begin{aligned}\boldsymbol{\Lambda}^{(\alpha)} &= \mathbf{H}_{KK}^{(\alpha)}-\mathbf{G}_{KA}^{(\alpha)}\cdot\left(\mathbf{G}_{AA}^{(\alpha)}\right)^{-1}\cdot\mathbf{G}_{AK}^{(\alpha)} \\ \boldsymbol{\Gamma}^{(\alpha)} &= \mathbf{G}_{KK}^{(\alpha)}-\mathbf{G}_{KA}^{(\alpha)}\cdot\left(\mathbf{G}_{AA}^{(\alpha)}\right)^{-1}\cdot\mathbf{H}_{AK}^{(\alpha)} \\ \boldsymbol{\phi}^{(\alpha)} &= \left[\mathbf{H}_{AK}^{(\alpha)}-\mathbf{G}_{KA}^{(\alpha)}\cdot\left(\mathbf{G}_{AA}^{(\alpha)}\right)^{-1}\cdot\mathbf{H}_{AA}^{(\alpha)}\right]\cdot\mathbf{u}_{A}^{(\alpha)}\end{aligned} \tag{2.2.4}$$

Eqn. (2.2.3) can be re-written as:

$$\mathbf{t}_{K}^{(\alpha)}=\mathbf{A}^{(\alpha)}\cdot\mathbf{u}_{K}^{(\alpha)}+\mathbf{b}^{(\alpha)} \tag{2.2.5}$$

where $\mathbf{A}^{(\alpha)}=\left(\boldsymbol{\Lambda}^{(\alpha)}\right)^{-1}\cdot\boldsymbol{\Gamma}^{(\alpha)}$ and $\mathbf{b}^{(\alpha)}=\left(\boldsymbol{\Lambda}^{(\alpha)}\right)^{-1}\cdot\boldsymbol{\phi}^{(\alpha)}$.

2.2.2 Compatibility Conditions at the Interface

For a subdomain, $\Omega^{(\alpha)}$, without any particle, $\mathbf{t}_{K}^{(\alpha)}$ consists of the traction values on the left interface ($I_{\alpha-1}$) and right interfaces (I_{α}) of the $\Omega^{(\alpha)}$, respectively. In this case, Eqn. (2.2.5) can be written as:

$$\left\{\begin{array}{c}-\mathbf{t}_{I_{\alpha-1}}^{(\alpha)} \\ \mathbf{t}_{I_{\alpha}}^{(\alpha)}\end{array}\right\}=\left[\begin{array}{cc}\mathbf{A}_{(\alpha-1)(\alpha-1)}^{(\alpha)} & \mathbf{A}_{(\alpha-1)\alpha}^{(\alpha)} \\ \mathbf{A}_{\alpha(\alpha-1)}^{(\alpha)} & \mathbf{A}_{\alpha\alpha}^{(\alpha)}\end{array}\right]\left\{\begin{array}{c}\mathbf{u}_{I_{\alpha-1}}^{(\alpha)} \\ \mathbf{u}_{I_{\alpha}}^{(\alpha)}\end{array}\right\}+\left\{\begin{array}{c}\mathbf{b}_{I_{\alpha-1}}^{(\alpha)} \\ \mathbf{b}_{I_{\alpha}}^{(\alpha)}\end{array}\right\} \tag{2.2.6}$$

For the first ($\alpha = 1$) and last domain ($\alpha = n$), since there is only one interface, Eqn. (2.2.5) can be written as:

$$\mathbf{t}_{I_1}^{(1)} = \mathbf{A}_{11}^{(1)} \cdot \mathbf{u}_{I_1}^{(1)} + \mathbf{b}_{I_1}^{(1)} \tag{2.2.7}$$

$$\mathbf{t}_{I_{n-1}}^{(n)} = \mathbf{A}_{(n-1)(n-1)}^{(n)} \cdot \mathbf{u}_{I_{n-1}}^{(n)} + \mathbf{b}_{I_{n-1}}^{(n)} \tag{2.2.8}$$

For a subdomain $\Omega^{(\alpha)}$ with particle(s), $\mathbf{t}_K^{(\alpha)}$ consists of traction values on the left interface, on the particle and on the right interface, and Eqn. (2.2.5) reads as

$$\left\{ \begin{array}{c} -\mathbf{t}_{I_{\alpha-1}}^{(\alpha)} \\ \mathbf{t}_P^{(\alpha)} \\ \mathbf{t}_{I_\alpha}^{(\alpha)} \end{array} \right\} = \begin{bmatrix} \mathbf{A}_{(\alpha-1)(\alpha-1)}^{(\alpha)} & \mathbf{A}_{(\alpha-1)P}^{(\alpha)} & \mathbf{A}_{(\alpha-1)\alpha}^{(\alpha)} \\ \mathbf{A}_{P(\alpha-1)}^{(\alpha)} & \mathbf{A}_{PP}^{(\alpha)} & \mathbf{A}_{P\alpha}^{(\alpha)} \\ \mathbf{A}_{(\alpha(\alpha-1)}^{(\alpha)} & \mathbf{A}_{\alpha P}^{(\alpha)} & \mathbf{A}_{\alpha\alpha}^{(\alpha)} \end{bmatrix} \left\{ \begin{array}{c} \mathbf{u}_{I_{\alpha-1}}^{(\alpha)} \\ \mathbf{u}_P^{(\alpha)} \\ \mathbf{u}_{I_\alpha}^{(\alpha)} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{b}_{I_{\alpha-1}}^{(\alpha)} \\ \mathbf{b}_P^{(\alpha)} \\ \mathbf{b}_{I_\alpha}^{(\alpha)} \end{array} \right\} \tag{2.2.9}$$

where subscript P stands for particle. In here, the size of the $\mathbf{u}_{I_\alpha}^{(\alpha)}$ is $3N_{I_\alpha}^{(\alpha)} \times 1$ where $N_{I_\alpha}^{(\alpha)}$ is the number of nodes on I_α, the size of the $\mathbf{u}_P^{(\alpha)}$ is $3N_P^{(\alpha)} \times 1$ where $N_P^{(\alpha)}$ is the total number of nodes on the particles' surface in $\Omega^{(\alpha)}$. Once the subdomain level equations are written for each subdomain, the compatibility condition at each interface $\mathbf{u}_{I_\alpha}^{(\alpha)} = \mathbf{u}_{I_\alpha}^{(\alpha+1)}$ and $\mathbf{t}_{I_\alpha}^{(\alpha)} = -\mathbf{t}_{I_\alpha}^{(\alpha+1)}$ can be imposed by the appropriate matrix operation to obtain the global assembly matrix as:

$$\left\{ \begin{array}{c} 0 \\ \mathbf{t}_P^{(2)} \\ 0 \\ \vdots \\ 0 \\ \vdots \\ \mathbf{t}_P^{(n-1)} \\ 0 \end{array} \right\} = \tilde{\mathbf{A}} \cdot \left\{ \begin{array}{c} \mathbf{u}_{I_1}^{(1)} \\ \mathbf{u}_P^{(2)} \\ \mathbf{u}_{I_2}^{(2)} \\ \vdots \\ \mathbf{u}_{I_\alpha}^{(\alpha)} \\ \vdots \\ \mathbf{u}_P^{(n-1)} \\ \mathbf{u}_{I_{n-1}}^{(n-1)} \end{array} \right\} + \tilde{\mathbf{b}} \tag{2.2.10}$$

where $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$ are the modified $\mathbf{A}$ and $\mathbf{b}$ matrices after the necessary matrix operations, respectively. For each interface (I_α), there is a row whose left-hand side value is equal to zero, and for particle(s) in each subdomian, there is a row whose left-hand side is equal to $\mathbf{t}_P^{(\alpha)}$ which is the traction vector on the particles' surfaces in $\Omega^{(\alpha)}$. At this point, the first step of the condensation process is completed. For the second step of the condensation process, rigid body motion constraint will be introduced.

2.2.3 Rigid Body Motion

In the case of a particulate flow, the problem at hand is a moving boundary problem. There exist contours inside the solution domain which represents the inclusions and/or the particles inside the solution domain. Although no-slip boundary condition is imposed on inclusions, neither traction nor velocity values are known on the particle boundary. However, a standard BEM formulation requires imposition of a boundary condition on the boundary of a solution domain. As a closure for the problem, we impose the constraint associated with the rigid body motion of the particles (assuming rigid particles), which means each boundary point has a velocity given by

$$\mathbf{u}_P = \mathbf{u}_B + \boldsymbol{\omega} \times \mathbf{r}_P \tag{2.2.11}$$

where $\mathbf{u}_P$ is the velocity at a node on the boundary of the particle, $\mathbf{u}_B$ is the velocity of the selected center of the particle, $\boldsymbol{\omega}$ is the angular velocity vector and $\mathbf{r}_P$ is the relative position vector of the particle point to the selected center of the particle. Imposing Eqn. (2.2.11) for all the boundary points on the particle can be related to the linear and angular velocities at the centroid of the particles through a kinematic matrix $\mathbf{M}$ as:

$$
\begin{aligned}
\mathbf{u}_P &= \mathbf{M}\cdot\mathbf{u}_B \qquad (2.2.12)\\
\mathbf{u}_P &= \left\{ \left\{ \begin{matrix} u_1 & u_2 & u_3 \end{matrix} \right\}^1 \left\{ \begin{matrix} u_1 & u_2 & u_3 \end{matrix} \right\}^2 \cdots \left\{ \begin{matrix} u_1 & u_2 & u_3 \end{matrix} \right\}^{N_P^{(\alpha)}} \right\}^{\mathbf{T}}\\
\mathbf{u}_B &= \left\{ \begin{matrix} u_1^B & u_2^B & u_3^B & \omega_1^B & \omega_2^B & \omega_3^B \end{matrix} \right\}^{\mathbf{T}}
\end{aligned}
$$

where $\mathbf{u}_q$ is the velocity vector of each particle node, $\mathbf{u}_B$ contains the linear and angular velocity at the center of gravity of the particle. Similarly, the force and moment on each particle can be obtained in a matrix form as [48]:

$$
\begin{aligned}
\mathbf{f}_B &= \mathbf{F}\cdot\mathbf{t}_P \qquad (2.2.13)\\
\mathbf{f}_B &= \left\{ \begin{matrix} f_1^B & f_2^B & f_3^B & m_1 & m_2 & m_3 \end{matrix} \right\}^{\mathbf{T}}\\
\mathbf{t}_P &= \left\{ \left\{ \begin{matrix} t_1 & t_2 & t_3 \end{matrix} \right\}^1 \left\{ \begin{matrix} t_1 & t_2 & t_3 \end{matrix} \right\}^2 \cdots \left\{ \begin{matrix} t_1 & t_2 & t_3 \end{matrix} \right\}^{N_P^{(\alpha)}} \right\}^{\mathbf{T}}
\end{aligned}
$$

where $\mathbf{f}_B$ contains the net force and moments with respect to center of gravity, $\mathbf{t}_P$ is the corresponding traction vector on the particles' surface and $N_P^{(\alpha)}$ is the number of particles in each subdomain. Neglecting the particle acceleration which can be justified with the low Re number nature of the flow field, and as a result, imposing the force-free and moment-free condition for the particle (i.e. the summation of all the forces and the moments on the particle are equal to zero), $\mathbf{f}_B = \mathbf{f}_{\text{ext}}$ for each particle (note that if there is no external force other than drag, $\mathbf{f}_{\text{ext}} = 0$) [48]. To condensate the particle nodes, in Eqn. (2.2.12), $\mathbf{u}_P = \mathbf{M}\cdot\mathbf{u}_B$

can be substituted, and equating $\mathbf{f}_B^{(\alpha)} = \mathbf{F}^{(\alpha)} \cdot \mathbf{t}_P^{(\alpha)} = \mathbf{f}_{\text{ext}}^{(\alpha)}$ for each particle as:

$$\left\{ \begin{array}{c} 0 \\ \mathbf{f}_{\text{ext}}^{(2)} \\ 0 \\ \vdots \\ 0 \\ \vdots \\ \mathbf{f}_{\text{ext}}^{(n-1)} \\ 0 \end{array} \right\} = \tilde{\mathbf{F}} \cdot \tilde{\mathbf{A}} \cdot \tilde{\mathbf{M}} \cdot \left\{ \begin{array}{c} \mathbf{u}_{I_1}^{(1)} \\ \mathbf{u}_B^{(2)} \\ \mathbf{u}_{I_2}^{(2)} \\ \vdots \\ \mathbf{u}_{I_\alpha}^{(\alpha)} \\ \vdots \\ \mathbf{u}_B^{(n-1)} \\ \mathbf{u}_{I_{n-1}}^{(n-1)} \end{array} \right\} + \tilde{\mathbf{F}} \cdot \tilde{\mathbf{b}} \tag{2.2.14}$$

where $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{M}}$ are

$$\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & & & \cdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{(2)} & \mathbf{0} & & & & \\ & \ddots & \ddots & \ddots & & & \vdots \\ & & \mathbf{0} & \mathbf{I} & \mathbf{0} & & \\ \vdots & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{0} & \mathbf{M}^{(n-1)} & \mathbf{0} \\ \mathbf{0} & & \cdots & & & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.2.15}$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & & & \cdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^{(2)} & \mathbf{0} & & & & \\ & \ddots & \ddots & \ddots & & & \vdots \\ & & \mathbf{0} & \mathbf{I} & \mathbf{0} & & \\ \vdots & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{0} & \mathbf{F}^{(n-1)} & \mathbf{0} \\ \mathbf{0} & & \cdots & & & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.2.16}$$

The global system above can be solved for the interface and particle velocities at each time step. Note that as a particle moves inside the solution domain corresponding interface equation has to be updated (*i.e.* equation 2.2.9) since geometry of the corresponding subdomain evolves. This is reflected to global system by updating the matrix $\tilde{\mathbf{A}}$ and vector $\tilde{\mathbf{b}}$ by only updating the corresponding entries. No computation is carried out for other empty subdomains in the next step until the particle moves to subsequent subdomain but matrix $\tilde{\mathbf{A}}$ is inverted at each and every time step. This process continues until the particle completes its journey inside the microchannel. For example if there is only one particle in the second subdomain then only the second interface equation is updated and no computation required for all other subdomains (*i.e.* 1 and 3 to N). Furthermore for subdomain number two only the corresponding entries of interface equation 2.2.9 is updated. These are all the matrices with subscript P.

In short, such a formulation offers two advantages; first being the reduction in the integral computation time since particles do not have to interact with long range elements and this is done only by using standard multi-domain BEM hence ”no additional approximations” added into the computations. Second because of the condensation process the final size of the global matrix reduces significantly. The performance of the proposed method is tested in the subsequent chapter.

Chapter 3

Results and Discussion

Results for 2D and 3D problems are presented here. The proposed method is tested against varying number of elements and number of particles. There is no unique way to construct ghost domains around the subdomain interfaces hence as a separate parameter they effect the performance of simulations which is also discussed. The discussion on condition numbers is carried out only for three dimensional problems since they are quite similar to two dimensional problems. For all simulations, Euler integration scheme with time step of 10^{-3} seconds is used to obtain the particle trajectories. All simulations are done on DELL R720 workstation with 32 Core CPU with 384 GB of RAM code development is performed in MATLAB programming environment. It is assumed that particles are buoyant in the buffer solution, and hence the only force acting on the particles is the hydrodynamic force which implies $\mathbf{f}_b = 0$.

3.1 2D Problems

For 2D problems the geometry is discretized with linear elements and field variables are approximated by constant elements. In such a setting, singular integrals can be evaluated via analytical methods and details of which can be found in ordinary textbooks [35]. 20-point Gauss quadrature is employed for the regular integrals, the number of integration points were increased for near singular integrals according to the degree of near singularity. The degree of near singularity is computed according to [94] but the number of integration points are used as at least three times larger than of the aforementioned reference this is because of the near singularity effects that arise when the close contact between particles and wall occur.

As mentioned in the previous chapter, the global system of equations has to be updated in each and every time step. After the update, matrix $\tilde{\mathbf{A}}$ has to be inverted in order to get the current velocity distribution on the interfaces and particle boundaries. An assumption is retained for 2D problems; the p-w and p-p interactions come into play when the separation distance is less than $10\times$diameter of the particle (d), therefore the effect of one particle in one subdomain would not affect the solution for other subdomains unless the particle is close to the interfaces of the subdomain. In this sense, if a particle is within the $\pm 5d$ of the interface, $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$ matrices are not updated during the computations. In the implementation, the ghost domains are generated before the particle reaches the proximity of $5d$ to an interface, and the size of the ghost domains is also chosen so that the distance between the interfaces of the ghost domain is as at least $\pm 10d$ away from the interfaces of the subdomains to expedite the computations. The validity of this assumption is verified in the appendix by implementing a sensitivity analysis.

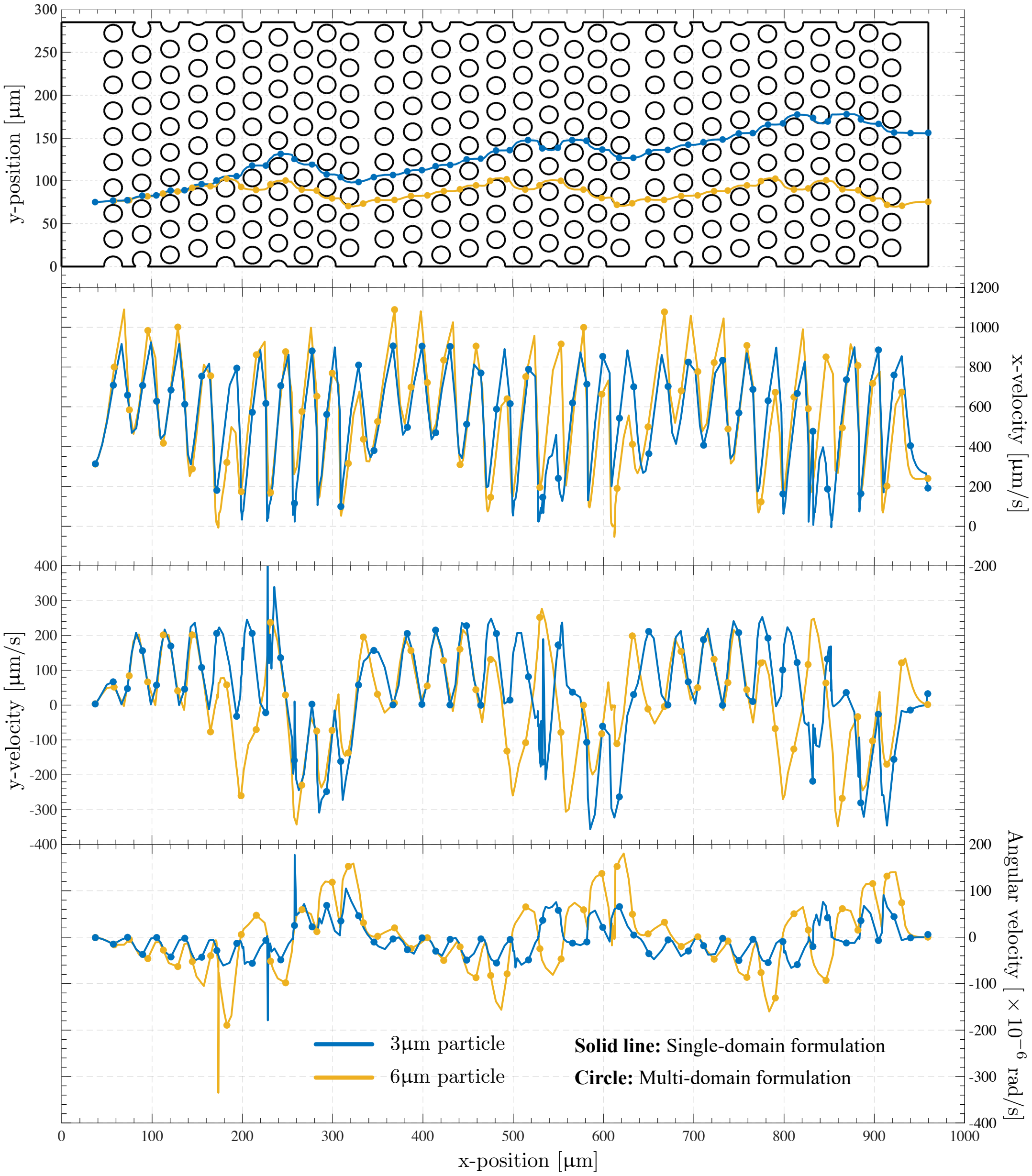

Figure 3.1: Comparison of MDBEM formulation with single domain formulation

Two geometries both of which are 2D are considered to investigate the speed-up factors as a function of the number of subdomains. The first one is a microfluidic channel for DLD separation with inclusions but without any repeating structure which can be seen in Figure 3.1 (top row). Strictly speaking, this geometry has also a repeating structure, however it is called as non-periodic due to the fact that when the domain is divided into a number of subdomains, the subdomains will not be the same. The second one is a microchannel with staggered and periodic inclusions. The major difference between two geometries is the layout of the non-moving inclusions. Inclusions inside DLD have non-regular topology, and hence not easy to partition to symmetric subdomains; whereas the staggered one has inclusions with regular topology, and therefore, easy to partition to various symmetric subdomain configurations. A uniform inlet velocity of 250 μm/s and zero pressure boundary conditions are assigned at the inlet and outlet of the microchannel, respectively. No-slip boundary conditions are assigned for the channel walls and non-moving inclusions (*i.e.* circular posts). 3 μm and 6 μm particles (in diameter) are released from the same initial position close to the lower channel wall. As seen in the figure, with the appropriate geometric configuration, two particles have clear separation distance between each other since the hydrodynamic interaction of the particles with the inclusions differs depending on the size of the particle. The computational domain has about one millimetre length which is smaller than the length of typical DLD chips [70, 71]. One can expect that the distance between the two particles would definitely amplify with the increasing length of the microchannel. The trajectories of the both particles are presented in the figure. The trajectories are quite close to the inclusions where strong hydrodynamic interactions occur. The computational model can capture these interactions quiet effectively and as a result separation by size is realized as observed in the literature [46, 70, 71]. The x– and y–velocities as well as the angular velocities of the both particles are illustrated in Figure 3.1. The comparison of the single-domain and multi-domain formulations in terms of the particle trajectory and particle velocities are also given in Figure 3.1, and quite a well agreement is observed between the two formulations which verifies the multi-domain formulation.

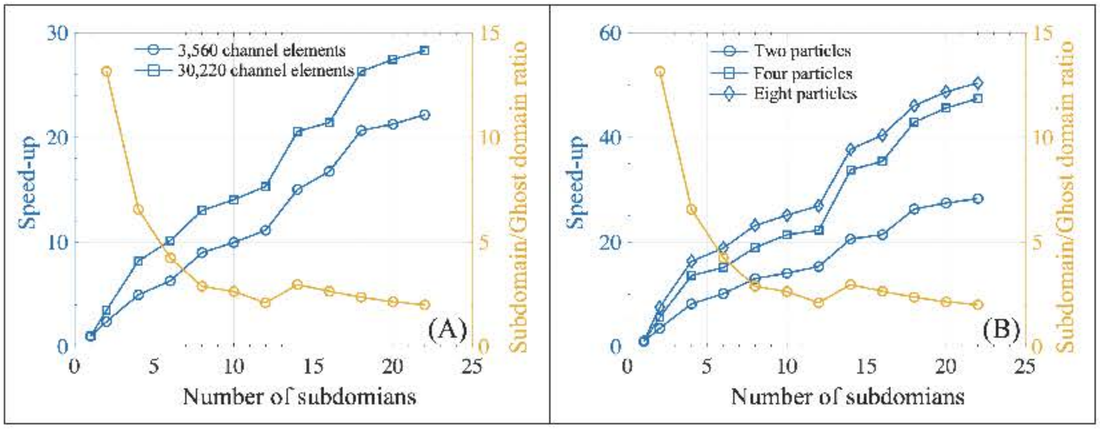

Figure 3.2: Speed-up as a function of number of subdomains for a non-periodic geometry

In Figure 3.2A, the speed-up factors for different number of subdomains are presented for the DLD chip with non-periodic inclusions. Two test runs were conducted with two particles in order to compare the effect of number of elements that were used to discretize the solution domain (3560 and 30220 elements for the same geometry). For both test runs the problem is divided into several subdomains ranging between two and twenty-two. The size of the ghost domains are at least $20d$ long. Depending on the size of the ghost domain, the number of elements on the ghost domain varies but in the simulations it is between 25 and 40 for the case of 3560 channel elements, and between 250 and 300 for the case of 30220 channel elements. As depicted in the figure, higher the number of elements, higher the speed-up. Speed-up factors as high as 30 are achieved with the proposed multi-domain formulation. In order to generate BEM matrices, one has to solve an N-body problem (one load point interacts with all elements). Multi-domain formulation has the same main feature of the N-body problem, so the computational time increases with increasing N. Multi-domain formulation becomes more efficient as the number of elements to discretize the single domain increases. This has mainly two reasons: First, the construction of the BEM matrices of slender geometries includes unnecessary calculations for the far-field interactions. As the distance between the load point and element grows, the interaction level decreases with $\mathcal{O}(1/r)$, therefore the effect of far-field elements diminishes. Actually this is the underlying idea of fast-multipole methods for BEM formulations [95]. As the number of elements becomes larger, the number of unnecessary calculations tremendously increases due to N-body problem nature of BEM. In addition, for a particulate flow problems (*i.e.* moving boundary problems), this N-body problem needs to be solved again and again through the course of motion which leads to heavy computational load. Such a drawback cannot be avoided with a single domain formulation for slender body moving boundary problems. Second, keeping in mind the BEM matrices are fully populated, as the size of BEM matrices grows condition numbers increases which may lead to ill-conditioned system that requires special care for the solution of the system.

In Figure 3.2A, one can observe two subsequent first-order system like convergence (from single domain to 13 subdomains, and 13 subdomains to 22) which is inherent to multi-domain formulations. The speed-up factors are saturated at a certain number of subdomains since each and every subdomain introduces an additional interface to the solution domain which results in extra elements. In 2D, a single domain with N-elements leads to $2N \times 2N$ BEM matrices (this would be $3N \times 3N$ in 3D). On the other hand, a domain with (n) subdomains results in $N/n + (n-1)\varepsilon$ elements with $[2N/n + 2(n-1)\varepsilon] \times [2N/n + 2(n-1)\varepsilon]$ BEM matrices where ε is the number of elements on an interface which is considerable smaller than N for slender bodies. As (n) grows so does $[(n-1)\varepsilon]$, and the size of the matrices becomes comparable with (N/n) and the improvement stops at a certain point. As an extension to conventional multi-domain formulations, a ghost domain that encapsulates the subdomain boundaries inside the solution domain is introduced to ensure seamless transition of particles from one subdomain to another. The presence of the ghost domain also necessitates extra computation time (note that the velocity on ghost domain boundaries are computed during the post-process and this calculation is out of discussion at this point). Therefore, the key factor here is the ratio of the ghost domain and subdomain ratio which severely determines the behaviour of the convergence curves (note the subdomain/ghost domain ratio is also given on the second axis). Inherently, the size of a ghost domain needs to be smaller than the size of a subdomain. We keep the ghost domain size about the same as we further subdivide the solution domain until 13 subdomains. Then, the size of ghost domains is halved, and the size of the subdomains further reduced. At this point, the ratio of the size of the subdomains to ghost domains experience an increase, however the convergence behaviour re-initiates after this point on. During the motion of particle inside the microchannel there exist a optimum ghostdomain-subdomain topology that minimizes the number of integral evaluations and hence the size of system of linear equations. The change in size of ghost domains after 13 subdomains takes the ghostdomain-subdomain topology closer to the optimum one and thus the convergence reinitiates. Note that such behaviour is not observed in the periodic case in which the size of ghostdomains is not changed.

In Figure 3.2B, the effect of the number of particles is represented. Speed-factor is the indication of the performance of the multi-domain formulation against the single domain formulation. To be able to compare the computational performance of different cases, CPU time for each case is also summarized in Table 3.2. As the number of particles increases, CPU time also rises due to the increase in the size of the system matrices. However, as seen in the table, when the number particles increases from two to four, the CPU time also almost doubles which is actually quite unexpected since the majority of the elements on the non-moving boundaries, and the elements of the moving boundaries (*i.e.* particles) do not contribute to the overall size of the system matrices. The reason for this behaviour is the formulation implemented for the tracking of particles–the details of which is given in [48]. In this formulation, the partition of the system matrices which corresponds to non-moving boundaries are computed once and then stored for further calculations. In this way, the number of particles is the determining parameter rather than the number of channel elements for the CPU time. The cases with four and eight particles have similar behaviour to that of the case with two particles, but the speed-up factors are better, and reaches about 50 which is a significant improvement achieved by just modifying formulation without any manipulation on the computation side.

In microfluidic applications, repeating sections exists in many cases, therefore the performance of the proposed multi-domain formulation is also evaluated for

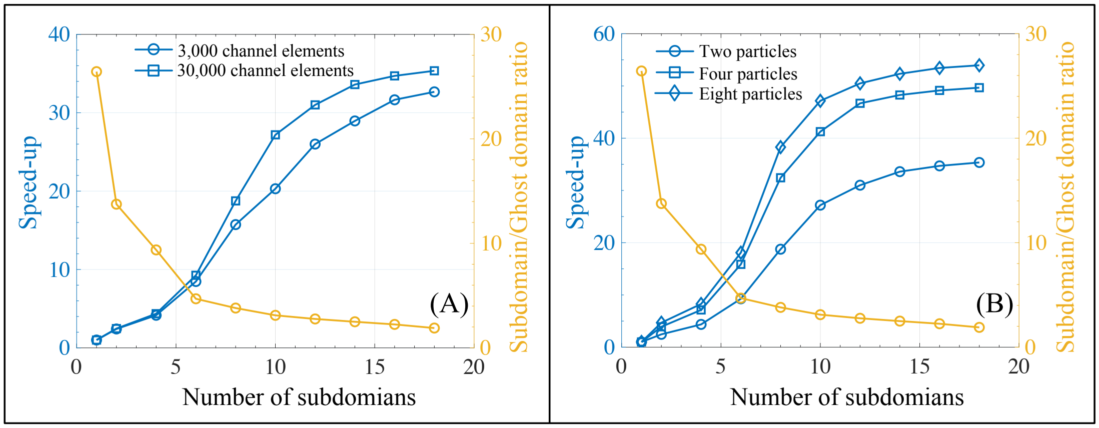

Figure 3.3: Speed-up as a function of number of subdomains for a periodic geometry

a periodic geometry. Figure 3.3 illustrates the performance of the multi-domain formulation with a periodic geometry, and the corresponding CPU time for each case is also tabulated in Table 3.1. Since the computational domain has repeated subdomains, the BEM matrices are computed only once for the subdomains without any particles. As a result, CPU time significantly decreases compared to the the case with non-periodic geometry. This trend also holds as the number of particles increases. As the number of particles increases from two to four, this time CPU time is not doubled this is due to fact that the overhead of the calculation of the partition of the system matrices corresponding to the non-moving boundaries.

Another important observations is that the decrease in the ratio of the size of subdomains to ghost domains is gradual, and so does the increase in the speed-up. This is due to the fact that the size of the subdomains is less restricted and an interface can be located between any two repeating structures regarding the periodic structure of the solution domain. However, owing to the fact that the limitations on the selection of the size of the subdomains and the ghost domains, the ratio reaches a limit where the speed-up values also saturate. Like the case with non-periodic geometry, the speed-up factors of 50 is achieved with a significant reduction in CPU time.

Table 3.1: CPU time and speed-up values for a periodic geometry

	Two particles		Two particles		Four particles		Eight particles	
# of channel elements	**3,000**		**30,000**		**30,000**		**30,000**	
# of subdomain	CPU time [s]	Speed-up	CPU time [s]	Speed-up	CPU time [s]	Speed-up	CPU time [s]	Speed-up
1	5096	1.00	19064	1.00	32452	1.0	42541	1.00
2	2116	2.40	7750	2.46	8308	3.90	9075	4.68
4	1227	4.15	4372	4.36	4553	7.12	5143	8.27
6	602.2	8.46	2059	9.26	2047	15.9	2351	18.1
8	324.4	15.7	1017	18.8	1001	32.4	1111	38.3
10	251.4	20.3	701.3	27.2	786.7	41.3	902.2	47.1
12	196.2	26.0	614.9	31.0	695.2	46.7	842.5	50.4
14	176.2	28.9	567.4	33.6	672.3	48.3	813.0	52.3
16	161.6	31.7	549.3	34.7	660.4	49.1	796.0	53.4
18	156.1	32.7	539.2	35.4	653.4	49.7	788.4	53.4

Table 3.2: CPU time and speed-up values for non-periodic geometry

	Two particles		Two particles		Four particles		Eight particles	
# of channel elements	**3,560**		**30,220**		**30,220**		**30,220**	
# of subdomain	CPU time [s]	Speed-up	CPU time [s]	Speed-up	CPU time [s]	Speed-up	CPU time [s]	Speed-up
1	7828	1.00	88474	1.00	156652	1.00	221155	1.00
2	3275	2.39	25309	3.49	28138	5.56	29466	7.50
4	1578	4.96	10799	8.19	11530	13.6	13542	16.3
6	1245	6.28	8732	10.1	10371	15.1	11736	18.8
8	871.1	8.98	6796	13.0	8264	19.0	9548	23.2
10	785.9	9.95	6301	14.0	7328	21.4	8799	25.1
12	703.6	11.1	5776	15.3	7035	22.3	8211	26.9
14	521.0	15.0	4299	20.6	4637	33.8	5859	37.7
16	467.3	16.8	4131	21.4	4422	35.4	5471	40.4
18	378.9	20.7	3363	26.3	3650	42.9	4800	46.1
20	368.1	21.3	3222	27.5	3434	45.6	4540	48.7
22	353.1	22.2	3126	28.3	3302	47.4	4388	50.4

3.2 3D Problems

For 3D problems the geometry and field variables are discretized with discontinious triangular quadratic elements. In such a setting, singular integrals have to be integrated by means of numerical methods. In case of the weakly singular integrals, polar coordinate transformation strongly singular integrals polar coordinate transformation together with subtraction of singularity methods are employed. Details of which can be found in standard textbooks [35]. As an alternative Tanh-Sinh quadrature can be implemented to evaluate the singular integrals which further requires transformation of triangular elements to quadrilateral elements [96]. 88-point Gauss quadrature is used to handle regular integrals. For near singular integrals an adaptive integration scheme is used and details can be found in [68]. The assumption that is mentioned in the previous chapter is alleviated in order to fully resolve the speed-up characteristic of the method. This has two implications the global system $\tilde{\mathbf{A}}$ is inverted in each and every time step and the velocity distribution on the ghost domain boundaries is not post processed rather they are obtained by re-constructing the global system for the new partition configuration.

One test problem considered to investigate the speed-up factors as a function of the number of subdomains. The geometry is a an empty rectangular microchannel with geometric dimensions of $5040\mu m \times 360\mu m \times 360\mu m$ (This dimensions are selected in order to obtain periodic subdomains). Note that GMSH package program is utilized to mesh the boundary with transfinite triangular mesh. A uniform inlet velocity of $100\mu m/s$ and zero pressure boundary conditions are assigned at the inlet and outlet of the microchannel, respectively. No-slip boundary condition is assigned on the channel walls. $5\mu m$ particles (in diameter) are released from the close vicinity of inlet. Similar to 2D problem the number of particles and number of elements are varied. For both tests runs the problem is divide into several subdomains ranging between two and sixteen. Note that there is no constraint on the size of the ghost domains and hence size of which is kept constant for different number of subdomains. Overall, this problem is quite similar to one that is introduced in the previous section as periodic microchannel

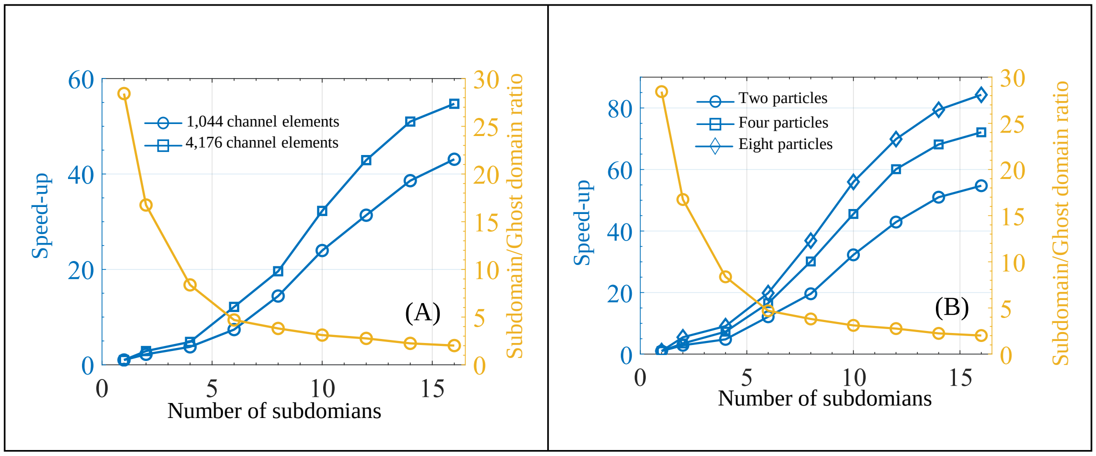

Figure 3.4: Speed-up as a function of number of subdomains for a periodic 3D geometry

and therefore BEM matrices are computed only once for the subdomains without any particles.

Similar to 2D problems one can observe two main characteristic of the speed-up curves. Firstly, high number of elements indicates higher speed-up. Secondly, increasing the number of particles enhances the performance of the proposed method. This behaviour can be seen in Figure 3.4 and corresponding data in Table 3.3 Unlike 2D problems however, speed-up factors are significantly higher and saturation of the corresponding curves is delayed. This is due to the fact that the number of elements that is introduced with the presence of an interface is even smaller relatively to the total number of elements for a given subdomain (for a rectangular domain in 3D, there are six surfaces two of which become an interface; however there are four edges two of which become an interface in 2D). Therefore, with introduction of each interface, the increase in the size of the system matrices is relatively low compared to 2D. This significant increase in the speed-up factors also results from the usage of quadratic elements. Each element is now introduces a element matrix of 18×18. The corresponding element level matrix size for 2D problems is 3×3. Therefore, each level of subdivision prevents redundant calculation of far-field element contributions significantly.

As indicated earlier, the global matrix $\tilde{\mathbf{A}}$ is inverted in each and every time step. Due to the nature of multi-domain BE formulation the global matrix of equation has a block diagonal formation and hence Thomas algorithm is used to obtain the interface and particle velocities. Hereby, only the diagonal elements of matrix $\tilde{\mathbf{A}}$ is inverted which further enhances the performance of formulation. Furthermore, even though the matrix $\tilde{\mathbf{A}}$ is block diagonal its condition number is at the order of 10^{20} which is huge when compared with single domain formulation. However, its constituents have significantly smaller condition number (*i.e.* 10^{5}) which makes them much easier to invert individually. The condition number of either matrix $\tilde{\mathbf{A}}$ or its constituents is affected by two parameters. First being the amount of condensation which refers to the amount of size reduction in subdomain level matrices due to condensation. As it increases the condition number also increases. For example amount of condensation for a particle is much larger when compared with an interface therefore, a condensed matrix for a particle has much higher condition number (*i.e.* 10^{10}) when compared with an interface matrix (*i.e.* 10^{5}). Second factor is the effect of near-singular integrals in case of miscalculation (*e.g.* insufficient number of Gauss points) increases the condition number. This can be observed for instance in case of very narrow ghost domains.

Table 3.3: CPU time and speed-up values for a periodic 3D geometry

	Two particles		Two particles		Four particles		Eight particles	
# of channel elements	**1,044**		**4,176**		**4,176**		**4,176**	
# of subdomain	CPU time [s]	Speed-up	CPU time [s]	Speed-up	CPU time [s]	Speed-up	CPU time [s]	Speed-up
1	25804	1.00	96768	1.00	152064	1.00	194961	1.00
2	11752	2.19	33629	2.87	43855	3.46	35619	5.47
4	6862	3.76	20176	4.79	20850	7.29	21425	9.09
6	3497	7.3	7972	12.1	9186	16.5	9819	19.8
8	1790	14.4	4934	19.6	5050	30.1	5287	36.8
10	1077	23.9	2994	32.2	3338	45.5	3483	55.9
12	822.93	31.3	2255	42.9	2530	60.0	2789	69.8
14	668.50	38.6	1897	51.0	2231	68.1	2455	79.4
16	598.83	43.0	1768	54.7	2109	72.0	2313	84.2

Chapter 4

Conclusions

The method introduced in this thesis is extension to the conventional MDBEM. Ghost domains encapsulating the interfaces between the subdomains are introduced in the formulation in order to ensure a seamless transition of particles form one subdomain to another. The method posses all the advantages of BEM for simulation of LOC devices further it significantly decreases the required computation time and memory requirements yet the accuracy of the method still holds. No new approximation or assumption is introduced. Although the formulation presented in this work only includes hydrodynamic interactions, it may easily be extended for multi-physics problems for wider range of microfluidic applications.

The formulation is demonstrated in 2D and 3D and the speed-up values are reported. The speed-up values are calculated with respect to performance of single domain BEM formulation. Two main parameters that are number of elements and number of particles, are varied in order to test the performance of the method. Whether it is 2D or 3D, the main finding is that the performance of the method enhances with increasing number of elements and number of particles. Note also that the topology of the ghost-domains further affects the performance of the method. The increase in speed-up saturates with increasing number of divisions. This is because of newly introduced interfaces that separates the subdomains with each other. Each new interface brings a computational burden and thus the required computation becomes comparable with previous subdivision.

In 2D, two test problems are considered; one for a periodic domain and one for non-periodic domain. For periodic domains, the proposed method further simplifies and hence required computation time further reduces. For periodic or non-periodic domain with 8 particles speed-up factor up to 50 is achieved. With decreasing number of particles and number of elements speed-up factors reduces to 20 yet this is still a significant speed-up. For non-periodic problem the saturation of the speed-up factors can be delayed by further changing the topology of the ghost-domains.

In 3D, only one test problem is considered. This is an empty microfludic channel, and hence is treated as a periodic geometry. When compared with its 2D counterpart, in 3D the speed-ups are much higher this is because of extra elements introduced for new dimension by means of which speed-up up to 80 is achieved. Similar to 2D problems with decreasing number of elements and number of particles the performance of the method decreases to 40 yet this is still a dramatic enhancement.

As a future direction, the utmost important point is to devise an adaptive subdivision algorithm depending upon the configuration of particles. In this thesis, the subdomains and ghostdomains are constructed beforehand of simulations. With such an algorithm optimum topology of subdomain and ghostdomain distribution can be constructed which will further augment the performance of the method. Secondly, though it is easy to implement the method presented here can be applied to handle multi-physics problems which is fairly common in LOC applications. Keeping in mind the calculation of elements of the BEM matrices is inherently parallel process, the computational performance of the formulation can be further enhanced with parallel programing [48] which will be one of the future research directions. Finally, regular integrals are evaluated using 88-point Gauss quadrature in 3D which is quite conservative for integration. A more computationally efficient integration algorithm that determines the number of quadrature points based upon distance between the evaluation element and load point will be implemented.

Bibliography

[1] A. Goldman, R. Cox, and H. Brenner, “The slow motion of two identical arbitrarily oriented spheres through a viscous fluid,” *Chemical Engineering Science*, vol. 21, no. 12, pp. 1151–1170, 1966.

[2] J. Castillo-León and W. E. Svendsen, *Lab-on-a-Chip devices and micro-total analysis systems: a practical guide*. Springer, 2014.

[3] G. Wang, F. Yang, and W. Zhao, “There can be turbulence in microfluidics at low reynolds number,” *Lab on a Chip*, vol. 14, no. 8, pp. 1452–1458, 2014.

[4] B. Cetin, S. D. Oner, and B. Baranoglu, “Modeling of dielectrophoretic particle motion: Point particle vs finite-sized particle,” *Electrophoresis*, vol. 38, pp. 1407–1418, 2017.

[5] G. Akiki, T. Jackson, and S. Balachandar, “Pairwise interaction extended point-particle model for a random array of monodisperse spheres,” *Journal of Fluid Mechanics*, vol. 813, p. 882, 2017.

[6] Y. Zhang, Y. Wang, and A. Prosperetti, “Laminar flow past an infinite planar array of fixed particles: point-particle approximation, oseen equations and resolved simulations,” *Journal of Engineering Mathematics*, vol. 122, pp. 139–157, 2020.

[7] S. Buyukkocak, M. B. Ozer, and B. Cetin, “Numerical modeling of ultrasonic particle manipulation for microfluidic applications,” *Microfluid Nanofluid*, vol. 17, no. 6, pp. 1025–1037, 2014.

[8] B. Cetin, M. B. Ozer, E. Cagatay, and S. Buyukkocak, "An integrated acoustic and dielec- trophoretic particle manipulation in a microfluidic device for particle wash and separation fabricated by mechanical machining," *Biomicrofluidics*, vol. 10, no. 014112, 2016.

[9] M. A. Şahin, B. Çetin, and M. B. Özer, "Investigation of effect of design and operating parameters on acoustophoretic particle separation via 3d device-level simulations," *Microfluid. Nanofluid.*, vol. 24, no. 1, p. 8, 2019.

[10] S. Zeinali, B. Cetin, S. Oliaei, and Y. Karpat, "Fabrication of continuous flow microfluidic device with 3D electrode structures for high throughput DEP applications using mechanical machining," *Electrophoresis*, vol. 36, pp. 1432–1442, 2015.

[11] R. Rasooli and B. Çetin, "Assessment of lagrangian modeling of particle motion in a spiral microchannel for inertial microfluidics," *Micromachines*, vol. 9, no. 9, p. 433, 2018.

[12] K. H. Kang, X. Xuan, Y. Kang, and D. Li, "Effects of DC-dielectrophoretic force on particle trajectories in microchannels," *J. Applied Physics*, vol. 99, no. 064702, pp. 1–8, 2006.

[13] B. Cetin and D. Li, "Continuous particle separation based on electrical properties using AC-dielectrophoresis," *Electrophoresis*, vol. 30, pp. 3124–3133, 2009.

[14] B. Cetin and D. Li, "Lab-on-a-chip device for continuous particle and cell separation based on electrical properties via AC-dielectrophoresis," *Electrophoresis*, vol. 31, pp. 3035–3043, 2010.

[15] G. G. Stokes *et al.*, "On the effect of the internal friction of fluids on the motion of pendulums," 1851.

[16] H. A. Lorentz, "Ein allgemeiner satz, die bewegung einer reibenden flüssigkeit betreffend, nebst einigen anwendungen desselben," *Abh. Theor. Phys*, vol. 1, p. 23, 1907.

[17] A. J. Goldman, R. G. Cox, and H. Brenner, "Slow viscous motion of a sphere parallel to a plane wall. part i motion through a quiescent fluid," *Chemical engineering science*, vol. 22, no. 4, pp. 637–651, 1967.

[18] A. Goldman, R. Cox, and H. Brenner, "Slow viscous motion of a sphere parallel to a plane wall. part ii couette flow," *Chemical engineering science*, vol. 22, no. 4, pp. 653–660, 1967.

[19] H. Faxén, *Einwirkung der Gefässwände auf den Widerstand gegen die Bewegung einer kleinen Kugel in einer zähen Flüssigkeit.* Verlag nicht ermittelbar, 1921.

[20] M. Stimson and G. B. Jeffery, "The motion of two spheres in a viscous fluid," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 111, no. 757, pp. 110–116, 1926.

[21] H.-J. Keh and J. Anderson, "Boundary effects on electrophoretic motion of colloidal spheres," *Journal of Fluid Mechanics*, vol. 153, pp. 417–439, 1985.

[22] H. J. Keh and S. B. Chen, "Electrophoresis of a colloidal sphere parallel to a dielectric plane," *Journal of Fluid Mechanics*, vol. 194, pp. 377–390, 1988.

[23] H. J. Keh and J. Y. Chiou, "Electrophoresis of a colloidal sphere in a circular cylindrical pore," *AIChE journal*, vol. 42, no. 5, pp. 1397–1406, 1996.

[24] L. Durlofsky, J. F. Brady, and G. Bossis, "Dynamic simulation of hydrodynamically interacting particles," *Journal of fluid mechanics*, vol. 180, pp. 21–49, 1987.

[25] J. F. Brady and G. Bossis, "Stokesian dynamics," *Annual review of fluid mechanics*, vol. 20, no. 1, pp. 111–157, 1988.

[26] J. Frechette and G. Drazer, "Directional locking and deterministic separation in periodic arrays," *Journal of fluid mechanics*, vol. 627, p. 379, 2009.

[27] J. Zhang, "Lattice boltzmann method for microfluidics: models and applications," *Microfluidics and Nanofluidics*, vol. 10, no. 1, pp. 1–28, 2011.

[28] D. Fedosov, B. Caswell, G. Karniadakis, and C. Pozrikidis, “Dissipative particle dynamics modeling of red blood cells,” *Computational hydrodynamics of capsules and biological cells*, pp. 183–218, 2010.

[29] T. Steiner, C. Cupelli, R. Zengerle, and M. Santer, “Simulation of advanced microfluidic systems with dissipative particle dynamics,” *Microfluidics and nanofluidics*, vol. 7, no. 3, pp. 307–323, 2009.

[30] J. Feng, H. H. Hu, and D. D. Joseph, “Direct simulation of initial value problems for the motion of solid bodies in a newtonian fluid part 1. sedimentation,” 1994.

[31] B. Maury, “Direct simulations of 2d fluid-particle flows in biperiodic domains,” *Journal of computational physics*, vol. 156, no. 2, pp. 325–351, 1999.

[32] J. Sarrate, A. Huerta, and J. Donea, “Arbitrary lagrangian–eulerian formulation for fluid–rigid body interaction,” *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 24-25, pp. 3171–3188, 2001.

[33] J. Tu, G. H. Yeoh, and C. Liu, *Computational fluid dynamics: a practical approach.* Butterworth-Heinemann, 2018.

[34] F. Sotiropoulos and X. Yang, “Immersed boundary methods for simulating fluid–structure interaction,” *Progress in Aerospace Sciences*, vol. 65, pp. 1–21, 2014.

[35] L. Gaul, M. Kögl, and M. Wagner, *Boundary Element Methods for Engineers and Scientists: An Introductory Course with Advanced Topics.* Springer Science & Business Media, 2013.

[36] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary Element Techniques: Theory and Applications in Engineering.* Springer Science & Business Media, 2012.

[37] G. Beer, *Programming the boundary element method.* John Wiley & Sons, Inc., 2000.

[38] C. Pozrikidis, *A practical guide to boundary element methods with the software library BEMLIB.* CRC Press, 2002.

[39] J. T. Katsikadelis, *Boundary elements: theory and applications.* Elsevier, 2002.

[40] G. K. Youngreen and A. Acrivos, "Stokes flow past a particle of arbitrary shape: a numerical method of solutions," *J. Fluid Mech.*, vol. 69, pp. 377–403, 1975.

[41] G. K. Youngreen and A. Acrivos, "On the shape of a gas bubble in a viscous extensional flow," *J. Fluid Mech.*, vol. 76, pp. 433–442, 1976.

[42] 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.

[43] T. Tran-Cong, N. Phan-Thien, and A. L. Graham, "Stokes problems of multiparticle systems: Periodic arrays," *Physics of Fluids A: Fluid Dynamics*, vol. 2, no. 5, pp. 666–673, 1990.

[44] N. Phan-Thien, T. Tran-Cong, and A. L. Graham, "Shear flow of periodic arrays of particle clusters: a boundary-element method," *J. Fluid Mech.*, vol. 228, pp. 275–293, 1991.

[45] L. A. Mondy, M. S. Ingber, and S. E. Dingman, "Boundary element method simulations of a ball falling through quiescent suspensions," *J. Rheology*, vol. 35, no. 5, pp. 825–848, 1991.

[46] B. Cetin, M. B. Ozer, and M. E. Solmaz, "Microfluidic bio-particle manipulation for biotechnology," *Biochem. Eng. J.*, vol. 92, pp. 63–82, 2014.

[47] A. Mammoli and M. Ingber, "Parallel multipole bem simulation of two-dimensional suspension flows," *Engineering Analysis with Boundary Elements*, vol. 24, no. 1, pp. 65–73, 2000.

[48] Z. Karakaya, B. Baranoglu, B. Cetin, and A. Yazici, "A parallel boundary element formulation for tracking multiple particle trajectories in Stoke's flow for microfluidic applications," *CMES-Computer Modeling in Engineering and Science*, vol. 104, pp. 227–249, 2015.

[49] L. Greengard and V. Rokhlin, “A fast algorithm for particle simulations,” *Journal of computational physics*, vol. 73, no. 2, pp. 325–348, 1987.

[50] N. A. Gumerov and R. Duraiswami, “Fast multipole method for the biharmonic equation in three dimensions,” *Journal of Computational Physics*, vol. 215, no. 1, pp. 363–383, 2006.

[51] H. Wang, T. Lei, J. Li, J. Huang, and Z. Yao, “A parallel fast multipole accelerated integral equation scheme for 3d stokes equations,” *International journal for numerical methods in engineering*, vol. 70, no. 7, pp. 812–839, 2007.

[52] Y. Liu, “A new fast multipole boundary element method for solving 2-d stokes flow problems based on a dual bie formulation,” *Engineering Analysis with Boundary Elements*, vol. 32, no. 2, pp. 139–151, 2008.

[53] M. Lisicki, “Four approaches to hydrodynamic green’s functions–the oseen tensors,” *arXiv preprint arXiv:1312.6231*, 2013.

[54] O. Steinbach, *Numerical approximation methods for elliptic boundary value problems: finite and boundary elements*. Springer Science & Business Media, 2007.

[55] K. Ishimoto and E. A. Gaffney, “Squirmer dynamics near a boundary,” *Physical Review E*, vol. 88, no. 6, p. 062702, 2013.

[56] K. Ishimoto and E. A. Gaffney, “Boundary element methods for particles and microswimmers in a linear viscoelastic fluid,” *Journal of Fluid Mechanics*, vol. 831, 2017.

[57] T. Tran-Cong and N. Phan-Thien, “Stokes problems of multiparticle systems: A numerical method for arbitrary flows,” *Physics of Fluids A: Fluid Dynamics*, vol. 1, no. 3, pp. 453–4–61, 1989.

[58] M. S. Ingber, “Numerical simulation of the hydrodynamic interaction between a sedimenting particle and a neutrally buoyant particle,” *Int. J. Numer. Meth. Fl.*, vol. 9, no. 3, pp. 263–273, 1989.

[59] M. S. Ingber, “Dynamic simulation of the hydrodynamic interaction among immersed particles in stokes flow,” *International Journal for Numerical Methods in Fluids*, vol. 10, no. 7, pp. 791–809, 1990.

[60] S. E. Dingman, M. S. Ingber, L. A. Mondy, J. R. Abbott, and H. Brenner, “Particle tracking in three-dimensional stokes flow,” *J. Rheology*, vol. 36, no. 3, pp. 413–440, 1992.

[61] H. Power and G. Miranda *SIAM Journal on Applied Mathematics*, vol. 47, no. 4, pp. 689–698, 1987.

[62] S. J. Karrila and S. Kim, “Integral equations of the second kind for stokes flow: Direct solution for physical variables and removal of inherent accuracy limitations,” *Chemical Engineering Communications*, vol. 82, no. 1, pp. 123–161, 1989.

[63] F. Xijun and Y. L. Yeow, “A boundary integral equation method for the stokes problem of multiparticle systems,” *Physics of Fluids A: Fluid Dynamics*, vol. 4, no. 5, pp. 1074–1076, 1992.

[64] H. Power and B. F. de Power, “Second-kind integral equation formulation for the slow motion of a particle of arbitrary shape near a plane wall in a viscous fluid,” *SIAM Journal on Applied Mathematics*, vol. 53, no. 1, pp. 60–70, 1993.

[65] H. Power, “A second kind integral equation formulation for the low reynolds number interaction between a solid particle and a viscous drop,” *Journal of Engineering Mathematics*, vol. 30, no. 1, pp. 225–237, 1996.

[66] A. A. Mammoli and M. S. Ingber, “Parallel multipole bem simulation of two-dimensional suspension flows,” *Engineering Analysis with Boundary Elements*, vol. 24, no. 1, pp. 65–73, 2000.

[67] H. Power and J. E. Gomez, “The completed second kind integral equation formulation for stokes flow with mixed boundary conditions,” *Communications in Numerical Methods in Engineering*, vol. 17, no. 4, pp. 215–227, 2001.

[68] C. Y. Chan, A. N. Beris, and S. G. Advani, "Second-order boundary element method calculations of hydrodynamic interactions between particles in close proximity," *International Journal for Numerical Methods in Fluids*, vol. 14, no. 9, pp. 1063–1086, 1992.

[69] K. J. Marshall and J. F. Brady, "The hydrodynamics of an active squirming particle inside of a porous container," *Journal of Fluid Mechanics*, vol. 919, 2021.

[70] L. R. Huang, E. C. Cox, R. H. Austin, and J. C. Sturm, "Continuous particle separation through deterministic lateral displacement," *Science*, vol. 304, no. 5673, pp. 987–990, 2004.

[71] N. Li, D. T. Kamei, and C.-M. Ho, "On-chip continuous blood cell subtype separation by deterministic lateral displacement," in *2007 2nd IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, pp. 932–936, IEEE, 2007.

[72] E. Divo, A. J. Kassab, and F. Rodriguez, "Parallel domain decomposition approach for large-scale three-dimensional boundary-element models in linear and nonlinear heat conduction," *Numer. Heat. Tt B-Fund.*, vol. 44, no. 5, pp. 417–437, 2003.

[73] N. Kamiya, H. Iwase, and E. Kita, "Parallel implementation of boundary element method with domain decomposition," *Eng. Anal. Bound. Elem.*, vol. 18, no. 3, pp. 209–216, 1996.

[74] J. Kane, B. K. Kumar, and S. Saigal, "An arbitrary condensing, noncondensing solution strategy for large scale, multi-zone boundary element analysis," *Comput. Methods Appl. Mech. Engrg.*, vol. 79, no. 2, pp. 219–244, 1990.

[75] X.-W. Gao, L. Guo, and C. Zhang, "Three-step multi-domain BEM solver for nonhomogeneous material problems," *Eng. Anal. Bound. Elem.*, vol. 31, no. 12, pp. 965–973, 2007.

[76] J. Frechette and G. Drazer, "Directional locking and deterministic separation in periodic arrays," *J. Fluid Mech.*, vol. 627, p. 379, 2009.

[77] M. Ramšak and L. Škerget, “A highly efficient multidomain BEM for multimillion subdomains,” *Eng. Anal. Bound. Elem.*, vol. 43, pp. 76–85, 2014.

[78] B. Wang, Y. Feng, S. Pieraccini, S. Scialò, and C. Fidelibus, “Iterative coupling algorithms for large multidomain problems with the boundary element method,” *Int. J. Numer. Meth. Eng.*, vol. 117, no. 1, pp. 1–14, 2019.

[79] G. Liu and K. E. Thompson, “A domain decomposition method for modelling stokes flow in porous materials,” *Int. J. Numer. Methods Fluids*, vol. 38, no. 11, pp. 1009–1025, 2002.

[80] M. Bush, “Stratified newtonian flow calculations by the boundary element method,” *Comput. Mech.*, vol. 7, no. 3, pp. 195–204, 1991.

[81] X.-W. Gao and M.-C. He, “A new inverse analysis approach for multi-region heat conduction BEM using complex-variable-differentiation method,” *Eng. Anal. Bound. Elem.*, vol. 29, no. 8, pp. 788–795, 2005.

[82] M. Ramšak and L. Škerget, “3D multidomain BEM for solving the laplace equation,” *Eng. Anal. Bound. Elem.*, vol. 31, no. 6, pp. 528–538, 2007.

[83] T. T. Bui and V. Popov, “Domain decomposition boundary element method with overlapping sub-domains,” *Eng. Anal. Bound. Elem.*, vol. 33, no. 4, pp. 456–466, 2009.

[84] M. S. Ingber and P. Tanski, J. A and. Alsing, “A domain decomposition tool for boundary element methods,” *Eng. Anal. Bound. Elem.*, vol. 31, no. 11, pp. 890–896, 2007.

[85] K. Erhart, E. Divo, and A. J. Kassab, “A parallel domain decomposition boundary element method approach for the solution of large-scale transient heat conduction problems,” *Eng. Anal. Bound. Elem.*, vol. 30, no. 7, pp. 553–563, 2006.

[86] J. Iljaž, L. Wrobel, M. Hriberšek, and J. Marn, “Subdomain BEM formulations for the solution of bio-heat problems in biological tissue with melanoma lesions,” *Eng. Anal. Bound. Elem.*, vol. 83, pp. 25–42, 2017.

[87] S. Ahmad and P. K. Banerjee, "Multi-domain BEM for two-dimensional problems of elastodynamics," *Int. J. Numer. Meth. Eng.*, vol. 26, no. 4, pp. 891–911, 1988.

[88] X. Lu and W.-l. Wu, "A new subregion boundary element technique based on the domain decomposition method," *Eng. Anal. Bound. Elem.*, vol. 29, no. 10, pp. 944–952, 2005.

[89] J. Layton, S. Ganguly, C. Balakrishna, and J. Kane, "A symmetric galerkin multi-zone boundary element formulation," *Int. J. Numer. Meth. Eng.*, vol. 40, no. 16, pp. 2913–2931, 1997.

[90] R. Larson and J. J. Higdon, "Microscopic flow near the surface of two-dimensional porous media. part 1. axial flow," *J. Fluid Mech.*, vol. 166, pp. 449–472, 1986.

[91] R. Larson and J. J. Higdon, "Microscopic flow near the surface of two-dimensional porous media. part 2. transverse flow," *J. Fluid Mech.*, vol. 178, pp. 119–136, 1987.

[92] J. Ravnik, L. Škerget, and Z. Žunič, "Combined single domain and subdomain BEM for 3D laminar viscous flow," *Eng. Anal. Bound. Elem.*, vol. 33, no. 3, pp. 420–424, 2009.

[93] J. Ravnik, L. Škerget, and Z. Žunič, "Velocity–vorticity formulation for 3D natural convection in an inclined enclosure by BEM," *Int. J. Heat Mass Transf.*, vol. 51, no. 17-18, pp. 4517–4527, 2008.

[94] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary element techniques: theory and applications in engineering.* Springer Science & Business Media, 2012.

[95] Y. Liu, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering.* Cambridge University Press, 2009.

[96] K. Ata and M. Sahin, "An integral equation approach for the solution of the stokes flow with hermite surfaces," *Eng. Anal. Bound. Elem.*, vol. 96, pp. 14–22, 2018.

[97] A. B. Richou, A. Ambaria, M. Lebeyc, and J. Nacirid, “Drag force on a circular cylinder midway between two parallel plates at very low Reynolds numbers–Part 1: Poiseuille flow (numerical),” *Chemical Engineering Science*, vol. 59, no. 15, pp. 3215–3222, 2004.

[98] H. Brenner, “The slow motion of a sphere through a viscous fluid towards a plane surface,” *Chemical engineering science*, vol. 16, no. 3-4, pp. 242–251, 1961.

Appendix A

Benchmark Problems

A.1 2D Problems

A benchmark problem from the literature [97] is employed to validate the BE formulation in two dimensions. The drag force generated on a stationary particle placed in a Poiseuille flow as shown in Fig. A.1 is considered. The non-dimensional drag force is calculated for varying k–parameter which is the ratio of the particle radius to channel height. The results given in [97] are based on finite element (FE) formulation. For comparison, the results are also compared against the Faxén formula, which is an analytical solution for the same problem and valid for $k \leq 0.5$. The results are summarized in Table A.1. The relative error of the current formulation is less than 1% against the Faxén formula. Note that the relative error of the results in [97] is higher than that of the present study. The relative error is less than 5% between the BE and FE formulations.

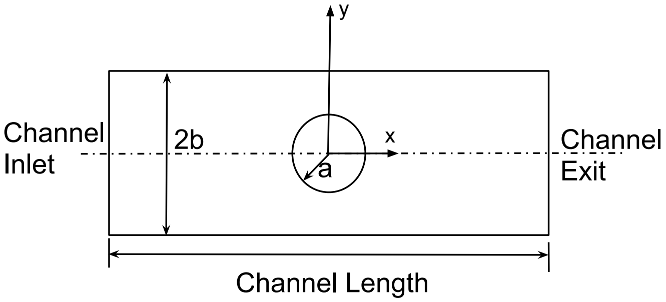

Figure A.1: Schematic drawing of the benchmark problem

Table A.1: Comparison between present study and results from literature for a stationary cylinder in a Poiseuille flow ($Re = 2 \times 10^{-4}$)

k(=a/b)	Present Study (PS)	Results from [97]	Faxen results from [97] (FR)	Rel. % error between PS and FR	Rel. % error between [97] and FR
0.0100	3.4012	3.5393	3.4057	0.13	3.77
0.0250	4.5218	4.7145	4.5282	0.14	3.96
0.0500	6.0123	5.9999	6.0214	0.15	0.36
0.1000	8.8907	9.1630	8.9060	0.17	2.80
0.2000	16.1724	16.3585	16.2072	0.21	0.92
0.4000	48.4175	48.8511	48.6229	0.42	0.47
0.8000	1231.8493	1256.1650	–	–	–
0.9000	7681.9006	7361.3287	–	–	–
0.9500	44,488.9807	44,093.1632	–	–	–

A.2 Sensitivity Analysis

As mentioned in Chapter 3 an assumption is retained in order to prevent inversion of global assembly matrix. The size of the ghostdomains is crucial for the sake of this assumption and hence a sensitivity analysis is done. To clarify the issue, a long channel with 22 subdomains is considered and a particle is placed in the mid subdomain to asses the effect of particle position to the velocity distribution on the interface. As seen, clearly, if the particle is closer than 5 diameters of particle, the velocity profile deviates from that of without particle.

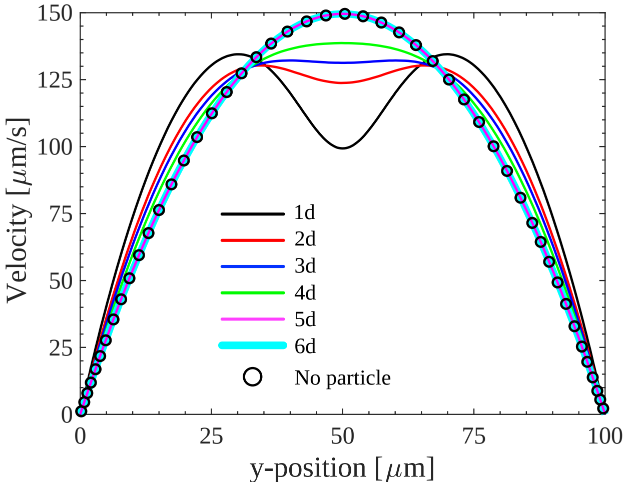

Figure A.2: Velocity distribution over an interface in case of an approaching particle

A.3 3D Problems

Three benchmark problems from the literature are considered. Unlike two dimensional problems, two problems arise in three dimensions. First, since the integrals are surface integrals number of Gauss points are limited to a certain number. For line integrals arising in two dimensions the number of Gauss points are unbounded and can be obtained analytically hence in case of near singularities one can increase the number of integration points accordingly. This strategy is employed in this thesis as well. Unfortunately, for surface integrals one has to pick other means to circumvent the near singular integrals. In this thesis an adaptive integration scheme is used details of which can be found in [68].Second, the BE formulation that is employed here originates from an integral equation of first kind hence there is no guarantee that the inverse of linear system of equations exist as the number of elements increase. However, in case of near-contact conditions in order to capture the hydrodynamics one has to increase the number of degrees of freedom accordingly which in return increases the condition number of linear system of equations. With these benchmark problems both near singularity issue as well as near-contact problem are adressed.

First problem being the drag force generated on a stagnant sphere placed inside a uniform flow. The absolute percentage error is calculated against increasing number of elements. Isoparametric discontinuous quadratic elements are used for approximation. The results are presented on a log-log graph which are depicted in Fig. A.3 in order to obtain the convergence which is two. The analytical solution is given by seminal work of Stokes [15] based upon bi-polar representation.

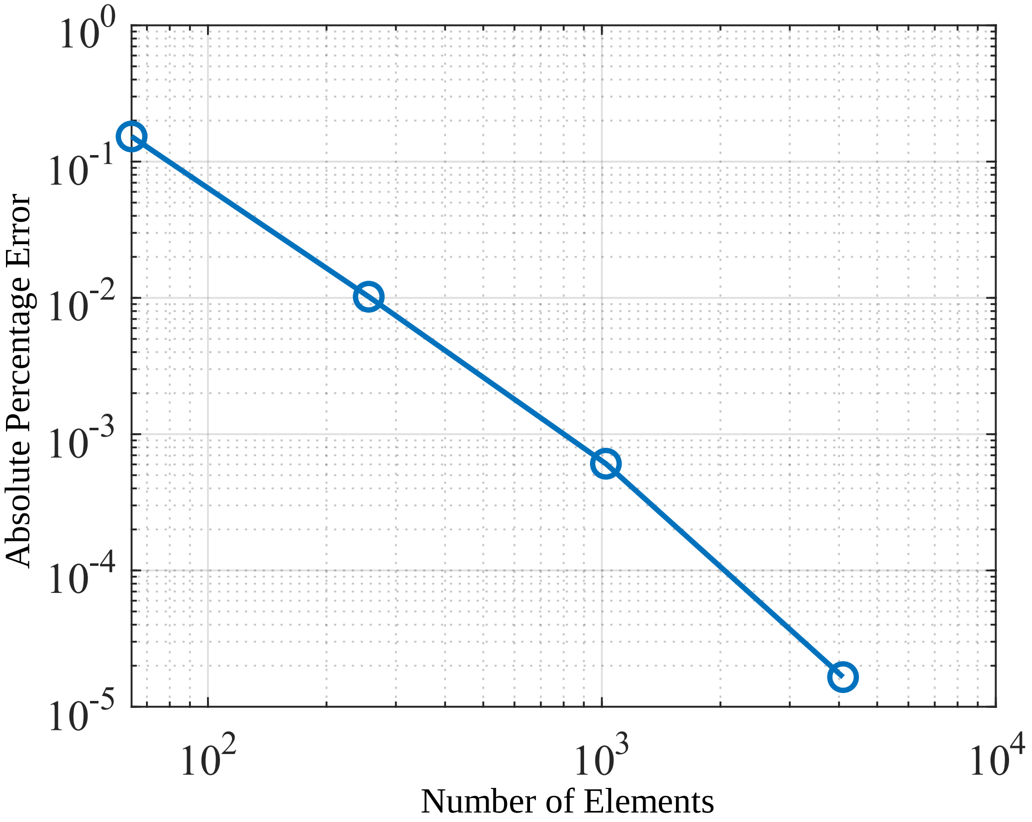

Figure A.3: Quadratic convergence of benchmark problem I

Second problem being the drag force generated on a sphere slowly approaching to a stationary wall with constant rigid body velocity. The schematic of the problem is depicted in Fig. A.4. The correction factor to Stokes equation is calculated with varying α which is adjusted according to the ratio of center to wall distance over the radii of the particle. The results given in [98] are based on semi-analytical solution which is based upon bi-polar representation of problem. The results are summarized in A.2. The relative error of the current formulation is less than 1% against the semi-analytical solution even in the worst case (*i.e.* particle-wall distance less than 100 nm).

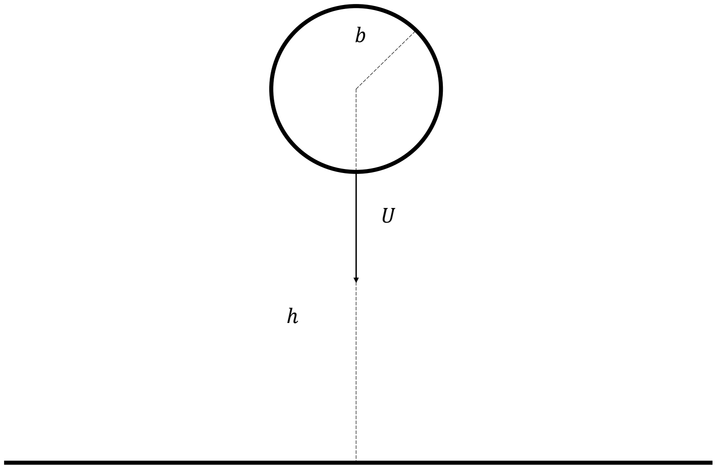

Figure A.4: Schematic drawing of the benchmark problem II

Table A.2: Comparison between present study and results from literature for the slow motion of sphere through a plane wall.

$\alpha\left(=\cosh^{-\mathbf{1}}\left(\frac{\mathbf{h}}{\mathbf{b}}\right)\right)$	Present Study (PS)	Results from [98]	Rel. % error between (PS) and [98]
3.00	1.1241	1.1242	0.0104
2.75	1.1653	1.1660	0.0560
2.50	1.2216	1.2220	0.0329
2.25	1.3001	1.3004	0.0228
2.00	1.4126	1.4129	0.0187
1.75	1.5792	1.5795	0.0171
1.50	1.8372	1.8375	0.0164
1.25	2.2628	2.2631	0.0157
1.00	3.0356	3.0361	0.0143
0.75	4.6731	4.6736	0.0104
0.50	9.2524	9.2515	0.0070
0.25	33.5936	33.5076	0.2565

Third problem being the settling motion of two spheres with arbitrary angle of attack from [1]. The schematic of the problem is depicted in Fig. A.5. The ratio of vertical settling velocity as a function of normalized relative spacing ($\frac{a}{h}$) and orientation angle (θ) is reported. Together with normalized angular velocity that occurs due to relative particle motion. Here only the percentage errors and results from present study due to large number of test cases. Interested reader can found the corresponding semi-analytical solutions in the aforementioned paper.

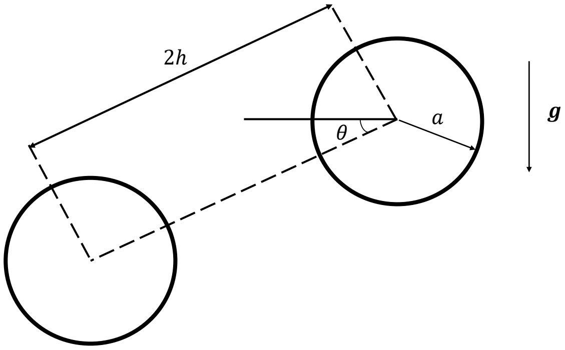

Figure A.5: Schematic drawing of the benchmark problem III

Table A.3: Absolute percentage error with resect to semi-analytical solutions given in [1] in vertical settling velocity ratio as a function of normalized relative spacing ($\frac{a}{h}$) and orientation angle θ.

$\frac{a}{h}$	$\theta = 0$	$\theta = \frac{\pi}{6}$	$\theta = \frac{\pi}{4}$	$\theta = \frac{\pi}{3}$	$\theta = \frac{\pi}{2}$
0.0135	0.0112	0.0115	0.0118	0.0121	0.0114
0.0366	0.0113	0.0113	0.0113	0.0113	0.0114
0.0993	0.0124	0.0123	0.0113	0.0112	0.0112
0.1000	0.0117	0.0118	0.0110	0.0111	0.0112
0.1111	0.0112	0.0104	0.0106	0.0107	0.0099
0.1250	0.0111	0.0112	0.0114	0.0107	0.0108
0.1429	0.0117	0.0118	0.0110	0.0111	0.0111
0.1631	0.0114	0.0107	0.0109	0.0111	0.0104
0.1667	0.0112	0.0114	0.0107	0.0109	0.0111
0.2000	0.0119	0.0120	0.0113	0.0105	0.0107
0.2500	0.0120	0.0113	0.0107	0.0109	0.0103
0.2658	0.0111	0.0107	0.0105	0.0102	0.0099
0.3333	0.0114	0.0106	0.0107	0.0100	0.0093
0.3374	0.0109	0.0103	0.0098	0.0102	0.0097
0.4251	0.0110	0.0106	0.0103	0.0100	0.0096
0.5000	0.0099	0.0099	0.0101	0.0094	0.0095
0.5295	0.0101	0.0095	0.0097	0.0098	0.0092
0.6481	0.0160	0.0141	0.0123	0.0106	0.0097
0.7477	0.0090	0.0087	0.0091	0.0094	0.0096
0.8868	0.0081	0.0086	0.0097	0.0097	0.0096
0.9566	0.0086	0.0037	0.0075	0.0042	0.0081
0.9950	0.1019	0.1771	0.0285	0.3577	0.0091
0.9976	0.0395	0.3953	0.6132	0.6390	0.0090
1.0000	0.9707	0.9128	0.9176	0.6990	0.0111

Table A.4: Vertical settling velocity ratio as a function of normalized relative spacing ($\frac{a}{h}$) and orientation angle θ.

$\frac{a}{h}$	$\theta = 0$	$\theta = \frac{\pi}{6}$	$\theta = \frac{\pi}{4}$	$\theta = \frac{\pi}{3}$	$\theta = \frac{\pi}{2}$
0.0135	1.0052	1.0064	1.0077	1.0090	1.0102
0.0366	1.0139	1.0173	1.0207	1.0241	1.0276
0.0993	1.0374	1.0467	1.0559	1.0652	1.0745
0.1000	1.0377	1.0470	1.0563	1.0656	1.0750
0.1111	1.0419	1.0522	1.0626	1.0729	1.0832
0.1250	1.0471	1.0587	1.0703	1.0820	1.0936
0.1429	1.0539	1.0671	1.0803	1.0936	1.1068
0.1631	1.0615	1.0766	1.0916	1.1067	1.1217
0.1667	1.0629	1.0783	1.0936	1.1090	1.1244
0.2000	1.0756	1.0939	1.1122	1.1305	1.1488
0.2500	1.0948	1.1173	1.1398	1.1623	1.1848
0.2658	1.1010	1.1247	1.1485	1.1722	1.1960
0.3333	1.1274	1.1563	1.1851	1.2139	1.2428
0.3374	1.1290	1.1581	1.1873	1.2164	1.2455
0.4251	1.1642	1.1988	1.2334	1.2680	1.3026
0.5000	1.1951	1.2332	1.2713	1.3093	1.3474
0.5295	1.2076	1.2467	1.2857	1.3248	1.3639
0.6481	1.2587	1.3000	1.3412	1.3825	1.4237
0.7477	1.3030	1.3438	1.3847	1.4255	1.4663
0.8868	1.3649	1.4027	1.4405	1.4783	1.5161
0.9566	1.3934	1.4294	1.4656	1.5016	1.5377
0.9950	1.4020	1.4372	1.4756	1.4583	1.5488
0.9976	1.4021	1.4450	1.4670	1.5224	1.5495
1.0000	1.4034	1.4385	1.4515	1.5180	1.5502