Cell mechanics in flow: algorithms and applications

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Kirill Lykov

under the supervision of
Igor V. Pivkin

September 2017

Dissertation Committee

Rolf Krause	Università della Svizzera italiana, Switzerland
Olaf Schenk	Università della Svizzera italiana, Switzerland
Petia Vlahovska	Northwestern University, USA
Chopard Bastien	University of Geneva, Switzerland
George Lykotrafitis	University of Connecticut, USA

Dissertation accepted on 7 September 2017

Research Advisor	PhD Program Director
Igor V. Pivkin	**Walter Binder**

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Kirill Lykov
Lugano, 7 September 2017

Abstract

The computer simulations are pervasively used to improve the knowledge about biophysical phenomena and to quantify effects which are difficult to study experimentally. Generally, the numerical methods and models are desired to be as accurate as possible on the chosen length and time scales, but, at the same time, affordable in terms of computations. Until recently, the cell mechanics and blood flow phenomena on the sub-micron resolution could not be rigorously studied using computer simulations. However, within the last decade, advances in methods and hardware catalyzed the development of models for cells mechanics and blood flow modeling which, previously, were considered to be not feasible.

In this context, a model should accurately describe a phenomenon, be computationally affordable, and be flexible to be applied to study different biophysical changes. This thesis focuses on the development of the new methods, models, and high-performance software implementation that expand the class of problems which can be studied numerically using particle-based methods.

Microvascular networks have complex geometry, often without any symmetry, and to study them we need to tackle computational domains with several inlets and outlets. However, an absence of appropriate boundary conditions for particle-based methods hampers study of the blood flow in these domains. Another obstacle to model complex blood flow problems is the absence the high-performance software. This problem restricts the applicability of the of particle-based cell flow models to relatively small systems. Although there are several validated red blood cell models, to date, there are no models for suspended eukaryotic cells .

The present thesis addresses these issues. We introduce new open boundary conditions for particle-based systems and apply them to study blood flow in a part of a microvascular network. We develop a software demonstrating outstanding performance on the largest supercomputers and used it to study blood flow in microfluidic devices. Finally, we present a new eukaryotic cell model which helps in quantifying the effect of sub-cellular components on the cell mechanics during deformations in microfluidic devices.

Acknowledgements

First of all, my utmost gratitude goes to my thesis adviser, Professor Igor V. Pivkin, for his guidance, opportunity to learn and develop myself. I am thankful to him for connecting me with the researchers all over the world, providing access to supercomputer resources and for excellent opportunities to visit scientific conferences. Professor Pivkin taught me how to work on scientific problems and deliver the results in journal papers and conferences.

I would like to express my gratitude to all faculty members of the Faculty of Informatics for their advice and the great courses. I also thank the Faculty staff for their tremendous help in any paperwork and activities. My gratitude is extended to the Swiss National Supercomputing Centre staff for providing computational resources and support. I also thank our collaborators, especially, Xuejin Li, Diego Rossinelli, Yu-Hang Tang, Dmitry Alexeev, Yasaman Nematbakhsh, Professor Petros Koumoutsakos, Professor George Em Karniadakis, and Professor Chwee Teck Lim for their knowledge and contribution.

I am grateful to my dissertation committee, Professor Rolf Krause, Professor Olaf Schenk, Professor Petia Vlahovska, Professor Chopard Bastien, and Professor George Lykotrafitis who have been kind to critique my work as readers of this Thesis.

Last but certainly not least, I would like to express my deepest gratitude to my family. My parents, Elena and Alexander, and grandfather, Anatoly, have fostered my developments since childhood by encouraging my interest in Mathematics and Computer Science. I thank my wife Maria for her love and support during the years at Ph.D. program.

My research has been supported by generous grants from Swiss National Science Foundation (200021_138231, 205321_173020) and Platform for Advanced Scientific Computing (PASC), as well as Swiss National Supercomputer Center (CSCS) grants for supercomputing time (u4, s311, s340, s422, s653, s747).

Contents

Figures

Tables

Chapter 1

Introduction

Blood is a suspension of different cells and a liquid component. The most numerous blood cells are red blood cells (RBCs) which occupy up to 45% of the blood volume. By traversing blood circulating system, RBCs deliver oxygen to all the parts of the body. Due to its composition, blood is a multiphase non-Newtonian fluid which leads to many hydrodynamical effects [188, 186]. For example, RBCs tend to migrate towards the middle of the capillary, leaving a cell-free layer next to the vessel walls, which leads to the higher RBCs flow rate in comparison to the solvent. Cell-free layer, mechanics of individual cells and the vessels geometry determine the uneven flux distribution in the bifurcations. Capillary networks are composed of bifurcations, and their transport properties are determined by the blood rheology at bifurcations.

Besides of RBCs, there are many other cells which can be found in the blood: white blood cells, platelets, circulating tumor cells, other epithelial and mesenchymal cells of different origin. Not all of them positively contribute to the organism health. Probably the most dangerous cells are circulating tumor cells (CTCs) which are the agents of the cancer metastasis. Cancer metastasis is an important problem attributed to nine out of ten cases of cancer deaths. During hematogenous metastasis, CTCs intravasate into the leaky vasculature around the tumor and eventually enter the bloodstream. After circulating for an unknown amount of time, the CTCs extravasate from the vasculature and grow secondary tumors.

Studies of the blood flow effects and cell migration have a great fundamental importance and find many medical applications. The blood flow phenomena on the capillary level impact the oxygen delivery and are exploited in the cell-plasma separation devices [249]. The detection of CTCs in the blood is one of the most potent methods for the early diagnosis of cancer [3] and a key target of liquid biopsies [169]. Improvements in the blood rheology and cell mechanics knowledge led to a wealth of activities in developing microfluidic devices. These devices are used for the suspended cells classification according to the deformability and size, blood-plasma separation, and blood filtering [212, 24].

Although some blood phenomena are quantified, we do not have the full understanding of the blood flow effects, cell migration, and suspended cells mechanics. The complexity of these problems might be attributed to the interplay between fluid hydrodynamics and cell mechanics. Due to the wide range of cell types with, sometimes, different composition, size, and deformability, cell mechanics is an especially challenging field. Cells have composite structure and mechanics of the cell as a whole is determined by the interplay of sub-cellular components. Currently, we have a lack of knowledge of the role each component plays during different deformations. This information would help to design the experimental setups for the cell mechanics investigation, which, now, is often long trial and error process.

Due to the complexity of these problems, it is often difficult to quantify involved phenomena using experimental studies. Therefore, numerical simulation is a valuable tool accompanying experimental work.

Today, the majority of simulations for microfluidic systems employs continuum models to solve the linear viscous Stokes equations of homogeneous fluids in microfluidic. These modeling approaches can handle flow dynamics in complex geometries of microfluidic channels, but they cannot resolve the essential biophysics of the flow on the submicron length scale [59]. Another limitation is that continuum models cannot be readily extended to capture cell topological changes, membrane transport, and electrochemical interactions. Such processes are essential for important microfluidic applications ranging from drug delivery and immunoassays [24] to micro-robots for non-invasive surgery [161]. These limitations led, firstly, to the development of the numerous RBC models and, later, the creation of more complicated eukaryotic cell models. The RBC models allowed getting valuable insights into blood rheology in straight pipes and simple microfluidic devices for healthy and malaria-infected RBCs [127, 19, 181]. At the same time, these simulations were, until a very recent time, limited to the simple domain geometries with a number of cells less than 1000. In particular, there were no simulations in the capillary bifurcations with RBCs models able to describe cell deformations accurately. This fact can be partially explained by the lack of the appropriate boundary conditions which would allow performing simulations in the domains which are asymmetric regarding the flow direction. A less visible aspect of the pursuit of RBC flow simulations is that they created formidable high-performance computing (HPC) challenges that limited the number of cells involved in the simulation. Finally, to our knowledge, there are no validated eukaryotic cell models suitable for modeling of the suspended cells undergoing significant deformations during passage in capillaries or microfluidic devices.

The purpose of this thesis is to extend the applicability of the particle-based method for the cell flow modeling by providing new models, algorithms, and techniques for their efficient HPC implementation. Open Boundary Conditions would allow performing simulations in the complex geometries, such as capillary networks, and could be applied for quantification of the related flux effects. Development of the computationally efficient software would open the route for the predictive studies in the entire microfluidic devices and could potentially reduce time to solution by two orders of magnitude, thus, making particle-based methods to be more affordable for the blood rheology research. A new eukaryotic cell model would improve understanding of the role different cell components play during cell deformation as well as allow performing predictive studies in the capillary networks and microfluidic devices.

1.1 Summary of contributions

This dissertation builds on Dissipative Particle Dynamics [84] method and Multiscale Visco-Elastic RBC model [180]. Our contributions are as follows:

- Open Boundary Conditions (OBCs) method for particle-based modeling [140]. To our knowledge, our method is the first boundary conditions suitable for the particle-based flow simulations with deformable bodies of arbitrary complexity. This development allowed us to perform, for the first time, simulations in capillary bifurcations and a part of microvasculature network employing an high-resolution RBC model.

- Large-scale simulation of the blood flow [197]. We present algorithms, techniques, and software for simulations of blood and cancer cell separation in complex microfluidic channels with sub-cellular resolution. This software was used to perform the largest simulation of deformable cells in complex flow domains to date, involving 1.4B cells and utilizing up to 65.5% of the available 39.4 PetaInstructions/s in the $18,688$ GPU nodes of the Titan supercomputer (Oak Ridge National Laboratory).

- Eukaryotic cell model [141]. We propose a new eukaryotic cell model that explicitly describes sub-cellular components and can describe cell deformations in microfluidic devices. To the best of our knowledge, this is the first mesoscale particle-based eukaryotic cell model suitable to study significant deformations which cells undergo in the flow.

- Polarized protein model [173]. We developed a systematic coarse-graining procedure for proteins. We used DPD extended with electrostatic interactions. This work is not aligned with the other part of the thesis and, hence, is briefly reviewed in Appendix A.

1.2 Outline

Chapter 2 provides background information about relevant cell biology and describes prior modeling efforts. We start with a discussion of the composition of different cells and relevant alternations in the mechanics caused by diseases. After that, we overview RBC models along with their applications and simulations of blood flow phenomena. Next, we present a review of the recent eukaryotic cell models, emphasizing those which explicitly model sub-cellular components. We, then, discuss the limitations of the existing methods due to the lack of the open boundary conditions. Finally, we review large-scale blood cell simulations, focusing on the computational aspects.

Chapter 3 is devoted to the numerical method and RBC model. We provide details of the Dissipative Particle Dynamics method along with fluid-structure interactions implementation and solid walls description. Viscoelastic RBC membrane model is discussed in details.

In *Chapter* 4, we present new OBCs which are robust approach for flow simulations using particle-based methods in complex simulation domains. Specifically, we apply OBCs to study the blood flow. We validate them by finding agreement between simulations of RBCs flow in the straight pipe using OBCs and periodic boundary conditions. As a model problem we consider flow simulations in micro-vessels bifurcations. We investigated the effect of the flow rate ratio between branches along with angle between them on the cells distribution in two daughter vessels. We, then, apply this method to simulate blood flow in part of a capillary network with several inlets and numerous outlets. This work was done in collaboration with the group of George Em Karniadakis (Brown University).

In *Chapter* 5, we address numerical challenges in large-scale blood flow simulations in complex geometries. We present software targeting two the most powerful supercomputers in Europe and world correspondingly available at the time. We describe software design and techniques which allowed us to achieve the efficient memory usage. We provide details of the communication strategies we employed to achieve the almost perfect weak and strong scalings. We discuss the optimizations and algorithms which allowed utilizing modern GPUs efficiently. The developed software was used to perform large-scale flow simulations in microfluidic devices and the obtained results were in general agreement with the experimental data. These simulations involved billions of high-resolution deformable RBCs. This work was one of the finalists of 2015 Gordon Bell Prize and was done in collaboration with groups of Petros Koumoutsakos (ETHZ) and George Em Karniadakis (Brown University).

Chapter 6 focuses on the modeling of a single eukaryotic cell. This is a combined experimental and simulation study. We describe the experimental setups for individual and collective cells analysis. We introduce a new model for a eukaryotic cell which explicitly describes sub-cellular components. We perform parameterization and validation of the proposed model. Finally, we present results and discuss the effect of different model components on the mechanics of the whole cell. We did this work in collaboration with the group of Chwee Teck Lim (NUS).

We conclude in Chapter 7 by summarizing the accomplishments in this work and discussing possible future research directions.

Chapter 2

Background and related work

2.1 Introduction

Cells are building blocks of live beings and there is a wide variety of cell types each tailored for a specific purpose in the body. Probably, structurally simplest cells are RBCs. RBCs are designed for oxygen transport and operate as containers of hemoglobin, a long protein which binds oxygen. This function imposes a simplicity of the cell composition: RBCs are made of the membrane and the supporting 2D proteins network. Other cells typically have a more complicated structure incorporating, beside of cell membrane, various organelles, nucleus, and 3D cytoskeletal network. Eukaryotic cells, by definition, are cells that contain a membrane-bound nucleus.

Numerous diseases lead to alternations of the mechanics of cells or introduce some atypical cells into the blood stream [162]. Detection and classification of such cells are of the great interest for diseases diagnostics.

Besides of mechanics of individual cells, the rheology of the cells suspension is also of the great importance for many applications. On the length-scale of capillaries and small vessels, the multiphase nature of blood along with mechanics of individual cells creates many interesting phenomena.

Techniques, used to study cell mechanics, can be split into two groups. The first group includes methods which operate with individual cells. They are typically used to estimate the viscoelastic properties of a single cell. The second group of experimental techniques, microfluidic devices, have a higher throughput and do not require skilled manual operations. There is a great variety of microfluidic devices and principles they employ to operate. We are primarily interested in the devices which exploit hydrodynamics of the cells suspension in the precisely designed flow geometry. The microfluidic devices are also actively used for the blood rheology analysis.

Despite progress in the experimental techniques in cell biophysics, the computational modeling is still an indispensable tool. The simulations can help in the acquisition of information of the fluid micro-rheology, including, often unexpected, fluid flow properties in complex device geometry, and can quantify such complex phenomena as cells adhesion, damage, or signaling [159, 254, 176]. By performing numerical studies, we can improve our understanding of the subcellular components mechanics which is valuable for many applications. In particular, quantifying the effect of different cell components would influence the drug discovery by offering a rich drug target space by providing possible corrections of cells behavior. Finally, numerical simulations allow prediction of cell behavior in a complex environment and accelerate the design cycle of the microfluidic devices [197].

In this chapter, we give an overview of the state-of-the-art cell models and discuss the corresponding numerical frameworks. The chapter is structured as follows. We start with the discussion of RBCs and eukaryotic cells structure and mechanics, Section 2.2. We, then, provide a retrospective on the RBC modeling, see Section 2.3. We overview well-established RBC models along with their most prominent applications. Next, we describe the state-of-the-art eukaryotic cell models, Section 2.4. Models for cells in adherent and suspended states are discussed. After that, we overview works dedicated to the cell suspension modeling in different microfluidic devices and microvasculature, Section 2.5. The choice of boundaries conditions (BCs) is important for modeling flow in complex geometries and BCs are also discussed in this section. In the Section 2.6, we discuss state-of-the-art implementations of cell flow models targeting supercomputers.

2.2 Cells composition and mechanics

The RBC membrane comprises a lipid bilayer and an attached cytoskeleton, which consists primarily of spectrin proteins arranged in a network and linked by short actin filaments at junction-complexes. Although this membrane has a mechanism for the fluid and proteins exchange with the surrounding, it is mostly impermeable to water which leads to volume conservation of a cell. Low compressibility of the lipid bilayer results in area conservation. In contrast to eukaryotic cells, RBC does not have the internal structure.

Eukaryotic cells are composed of a plasma membrane, various organelles, one or several nuclei, and internal cytoskeleton. Mechanics-wise, plasma membrane is similar to RBC membrane: it plays the same role and has similar mechanical properties. Under the membrane, there is a 2D lattice of proteins, called cell cortex, which plays a central role in shape control [203]. Cell cortex is attached to the membrane by membrane-anchoring proteins.

Cytoskeleton of eukaryotic cells is a complex 3D network made of different filaments, and it occupies the space between plasma membrane and nucleus. This component plays a significant role in many cell processes: migration, adhesion, division, stress resistance, and mechanotransduction. There are three types of cytoskeleton filaments: actin filaments, microtubules, and intermediate filaments. Actin filaments are semi-flexible biopolymers which exhibit the highest resistance to deformations until some critical stress value. If the stress is greater than this value, they are fluidized. Intermediate filaments can resist moderate deformations, and they do not fluidize under high values of shear stress, hence, providing structural integrity to the cell. Microtubules do not contribute significantly to the cell stiffness. Being interconnected with filaments of other types, they stabilize the cytoskeleton. They also might contribute to the compression resistance under high pressure [21]. The stiffness of the cells is primarily determined by the composition of the internal cytoskeleton [162]. The cytoskeleton is attached to the cell cortex and membrane by numerous linking proteins which also might be specific for a particular cell line [95].

Nucleus is the largest eukaryotic organelle, and it is typically stiffer than the cell itself [75]. Mechanics-wise, the most important components of the nucleus are nucleus envelope and chromatin network. The nucleus envelope is composed of two phospholipid bilayers with an attached lamins meshwork. These bilayers, or membrane, act as a barrier between cytoplasm and nucleus internal structure [121]. In comparison to the cell membrane, it exposes weaker area and volume constraints allowing fluid to get in and out [198]. The nuclear lamina, which is considered to be the main contributor to the nucleus stiffness [63], is 2D meshwork attached to the inner bilayer. According to a very recent study, lamina levels control nuclear strain stiffening at large extensions [214]. Contrary, chromatin network, which occupies the nucleus volume, governs response to small extensions. Chromatin is also believed to play an important role in the cell migration [63].

Eukaryotic cells of different types might have different mechanical properties. As an example, a cell, which belongs to tumorigenic line MDA-MB-231, is several times softer than non-tumorigenic MCF-10A cell [138]. Besides of cell type, the state of the cell significantly influences the cell mechanics.

Progression of diseases might also lead to alternations in cell components mechanics [162]. Some disorders cause changes in the plasma membrane elasticity. For instance, membrane can be softened by HIV [2] or stiffed by specific antigen produced by some parasites harboring cells [154]. Progression of other diseases alters properties of the cytoskeleton. The most prominent example is the cytoskeleton structure degradation and density decrease of epithelial tumor cells [217]. The higher is the metastatic potential of a cell, the lower is the stiffness of the cytoskeleton, and, hence, the cell as a whole [28]. A review of other diseases affecting cytoskeleton might be found elsewhere [191]. Some tissue-specific disorders influence the mechanical properties of the nucleus due to the mutations in genes encoding lamins and associated nuclear envelope proteins: Emery-Dreifuss muscular dystrophy, dilated cardiomyopathy, familial partial lipodystrophy, as well as over 20 other diseases [198, 199].

2.3 RBC models

The computational models of RBCs help to solve many practical problems. They found applications in testing hypothesis of the cell mechanics, used to predict the performance of microfluidic devices, and improve the knowledge of the cell mechanics changes caused by diseases. RBC modeling is a very active research area, and there are many different modeling approaches, each having its advantages. In this work, we are primarily interested in models suitable for the RBC flow modeling. Thus, we do not discuss models which are developed for protein-level resolution modeling [224, 129]. We, instead, focus on the models which, being mechanically accurate on the given scale, are computationally affordable to simulate blood rheology. Here, we briefly discuss corresponding approaches while complete reviews might be found elsewhere [251, 134].

To model suspended cells deformations and rheology in the microvasculature and microfluidic devices, a numerical framework should incorporate three essential components: the mechanical model of an individual cell, fluid and fluid-structure interactions models, and solid walls to model microfluidic devices or microvasculature geometry. Every framework utilizes one or several computational methods to discretize the problem. Although the detailed comparison of computational methods is out of the scope of this work, we give a short overview of the most popular methods. The relevant computational methods can be split into three groups: mesh-based, particle-based, and combined. The most popular mesh-based method is Finite Elements Method (FEM). Among particle-based methods, Dissipative Particle Dynamics (DPD) and Smoothed Dissipative Particle Dynamics (SDPD) are used more often than others. Both methods allow parameterizing the fluid and fluid-structure interactions, while the cell itself is modeled with the help of various many-bodies potentials. Among combined methods, a well-developed and dominating approach is to model the blood plasma with the Lattice-Boltzmann method (LB), RBC membrane forces with FEM, and cell-fluid interactions using immersed boundary method (IB). Further, we will call this numerical framework LB/IB.

Most of these methods can correctly describe hydrodynamics and fluid-structure interactions. A possible disadvantage of DPD, in comparison to other methods, is that DPD parameters do not have a straightforward physical interpretation and have to be tuned manually to achieve necessary properties of the modeled matter, such as viscosity. At the same time, DPD supports thermal fluctuations which, typically, are not supported by other methods. One of the pitfalls of particle-based methods is that they often use the same particles for fluid and membrane which might lead to an unrealistically high mass of the cell membrane. LB/IB method does not have this disadvantage because fluid and cell are modeled by different methods coupled through IB. Yet LB/IB might be more computationally intensive [251].

Simulations, involving mechanical models for RBCs, were established by the pioneering work of Boal *et al.* [15, 14] that predicted the mechanical properties of the RBC spectrin network. The first 3D spectrin-based RBC membrane model, developed by Discher *et al.* [32], was used to investigate how the RBC cytoskeleton deforms during the micropipette aspiration. This work was extended by Li *et al.* [131] and the resulting spectrin-level RBC model became a starting point for later developments. Another extension of Discher's work was published the same year by Noguchi and Gommper [165]. Contrary to the model by Li *et al.*, Noguchi and Gommper proposed a coarse-grained RBC model and coupled it with Stochastic Rotation Dynamic method, which allowed them to study the effect of fluid flow on RBC membrane. Later, Dupin *et al.* introduced an alternative low-resolution RBC model using LB/IB formalisms [36]. This model is developing the ideas behind Li's model yet it is less computationally demanding, and was used to perform one of the first simulation of RBC flow in pipe. A year later, a systematic coarse-graining procedure for Li's model was introduced [180]. This development provided a systematic procedure for the reduction of the number of vertices by more than one order of magnitude. The DPD method extension, proposed by Pan *et al.* [168], allowed adding a viscous term to the pure elastic RBC model. The resulting model, which will be referred to as the Multiscale Viscoelastic RBC model (MS-VE-RBC), was integrated into a DPD fluid and intensively used to study the mechanical properties of individual RBCs [189, 124, 170] as well as blood flows under various conditions [127, 49, 45]. Quinn *et al.* performed combined simulation and experimental study for the RBC passing a microfluidic device [189]. Lei *et al.* studied blood rheology for healthy and sickle cell anemia RBCs [124]. In a later study, Lei *et al.* investigated the blood cell flow in pipes with various diameter and hematocrit to study the distribution of cells in the flow, cell-free layer and viscosity of the proposed RBCs suspension.

However, to the moment the most popular approach is to use LB/IB framework. Here, we will mention only a few representative works [36, 116, 89, 25]. One of the first large-scale simulations of the deformable RBCs flow was performed using LB/IB by Clausen *et al.* [25]. Using the same method, Krüger investigated tumbling to the tank-treading transition of RBC in the shear flow and simulated deformation of RBC in microfluidic device [116]. In the next study, Krüger *et al.* investigated the transit of a detailed RBC (1000 vertices/cell) past an array of cylindrical obstacles [117], and studied the RBC deformations close to the obstacle along with the displacement of the RBC due to the characteristic features of the device geometry design. In a later study, this model was used to investigate the displacement of the cells in the deterministic lateral displacement microfluidic devices [233]. Hyakutake *et al.* studied RBCs distribution in small bifurcations [89].

Besides of DPD and LB/IB, there are many other methods used in the context of RBC modeling. Here we will give a few examples. Finite Volume method for fluid modeling combined with FEM for the cell modeling was applied to study RBCs aggregation [248]. Moving particle semi-implicit method was used to study RBCs flow effects in capillaries, an effect of malaria infection on the RBC mechanics [90, 4]. Smoothed Particle Hydrodynamics is another method suitable for modeling of the RBC mechanics and flow rheology. Wu and Feng conducted 3D simulations of a healthy and malaria-infected RBC flowing through a sudden constriction, to find out whether the RBC can go through this constriction [243]. There are also various combinations of the before mentioned methods. For instance, in the very recent work Ye *et al.* used SDPD. methods [41], coupled with IB method to describe RBC flow in microvessels [252].

Summarizing, a wide range of different RBC models has been successfully employed for the blood flow rheology studies. MS-VE-RBC model integrated into DPD environment arguably has several advantages over other models. Firstly, MS-VE-RBC model was obtained from a spectrin-level model using a systematic coarse-graining procedure and, hence, can be applied to study problems at an arbitrary level of discretization. Secondly, this model takes into account viscous properties of the membrane, which is hard to achieve using alternative methods. Thirdly, MS-VE-RBC model was extensively validated using optical tweezers and various flow experiments [180, 127, 49, 189]. Finally, the choice of DPD method

Table 2.1. Computational models and their applications.

Source	Applications	Cell type
[257]	Squeezing through channels of different geometry	CTC
[167]	Flow in pipe	$HL60$,$CF34^+$
[220, 221]	Flow in pipe, cell-wall adhesion	WBC
[246, 247]	Flow in pipe, passage though a narrow slit	CTC, WBC
[46]	Flow in pipe	WBC
[197]	Flow in microfluidic devices	CTC, WBC
[44]	AFM	$SH-SY5Y$
[104]	Artificial continuous deformations	Osteocyte
[107, 108, 109]	Cell migration in pipes and extracellular matrix; filopodia formation	CHO, HUVEC
[136]	Needle microinjection	Zebrafish embryo cell
[78]	Flow around adherent cell in pipe, Micropipette aspiration	hPDC
[255]	Plasma and nucleus membranes induced by micropipette pulling	Bovine capillary endothelial cell
[230, 229]	Stretching, compression, and shear deformations	Fibroblast
[34, 195]	Probe indentation	Articular cartilage chondrocytes

as a general framework provides a unified way to describe fluid, fluid-structure, and structure-structure interactions.

2.4 Eukaryotic cell models

In comparison to RBCs, eukaryotic cells have a more complex structure, and the modeling efforts in this direction were started later. The advances in computational methods, RBC modeling and significant progress in hardware made this area to be very active in recent years. In this section, we will focus on the models for adherent and suspended eukaryotic cells. We do not consider cell models for the tissue modeling, a review of these models can be found elsewhere [57].

Since RBC and eukaryotic cell membranes are structurally and mechanically

Table 2.2. Computational models classified by methods, fluid flow modeling, and cell model components.

Source	Simulation method	Fluid flow	Internal structure
[257]	Volume of Fluid	Yes	No
[167]	Cont.	Yes	No
[220, 221]	LB/IB	Yes	No
[246, 247]	DPD	Yes	No
[46]	DPD	Yes	No
[197]	DPD	Yes	No
[44]	Particle	No	Cyt. and Nucl.
[104]	FEM	No	Cyt. and Nucl.
[107, 108, 109]	Langevin dynamics	No	Cyt. and Nucl.
[136]	DPD	No	Cyt.
[78]	LB/IB	Yes	Nucl.
[255]	Particle	No	Cyt. and Nucl.
[230, 229]	Particle-based, energy minimization	No	Cyt. and nucl.
[34, 195]	FEM	No	Cyt. and nucl.

similar, the modeling efforts for eukaryotic cells started with the application of RBC models with altered parameters. Such models usually represent a cell as an empty deformable shell. Despite oversimplified structure, these models capture the size and the shape of a cell and, being coupled with the fluid model, are often used to predict microfluidic effects such as deterministic lateral displacement or cell margination.

The interest in studying processes which involve significant deformation of the cell and, thus, requiring mechanical response of the internal cell components, led to the development of approaches which explicitly model nucleus and cytoskeleton. Typically, these models use the same hollow sphere model for the cell membrane and the nucleus. These models were applied to the numerical investigation of such processes as needle microinjection, Atomic Force Microscopy (AFM), cell migration, and deformation in microfluidic devices, see Table 2.1. The classification of the models according to the presence of the sub-cellular components is shown in Table 2.2. We additionally classify models by the support of hydrodynamics and observe that models, which describe fluid, usually do not consider internal cell structure. Computational methods used for the eukaryotic cells modeling are the same as those utilized for the RBC modeling, see Table 2.2.

The variety in cells modeling approaches can be explained by the complex nature of the problem. The first source of this complexity is the diversity in the cell lines along with processes in which they are involved. The second one is the lack of understanding of the role each cell component plays during mechanical deformations. These factors also explain the fact that the current state-of-the-art models are phenomenological and each one is tailored to a very particular cell line and related processes.

For the next two sections, we split the existing models by the application area. The first group contains models for the suspended cells which are usually employed to study fluid flow problems. The second group is dedicated to the modeling of the cells in the adherent state. Some models target problems which are at the intersection of these two groups.

2.4.1 Suspended cell models

The majority of suspended cells are blood cells such as RBCs, WBCs, and platelets. Among these cells, only WBCs are eukaryotic. CTCs, contrary to other cells, are atypical and their presence in the blood indicates cancer metastasis. These cells are used as biomarkers and have attracted a lot of attention in the last decade. Both WBCs and CTCs are several times bigger and stiffer than RBCs. This difference in size between RBCs and suspended eukaryotic cells leads to a rheological phenomenon called margination, which is a process whereby stiff cells are displaced to the vessel wall [65]. Margination and related effects are often studied with numerical models. Another popular application of the cell models is to investigate the cell passage in the microfluidic devices. In the rest of the section, we will briefly discuss some recent modeling efforts.

In a series of papers, Takeishi *et al.* modeled margination of WBCs and CTCs and, in the later work, adhesion of CTCs to vessel walls [219, 220]. This development allowed them to numerically investigate the effect of the cell size on the flow mode and the cell velocity and identify similarities and differences between leukocytes and CTCs. Further, it was found that the bullet motion enables firm adhesion of a cell to the capillary wall [221]. These results suggest that even under the interaction between proteins responsible for WBC rolling, a cell can show firm adhesion in a small capillary. Modeling-wise, Takeishi *et al.* employed LB/IB method and empty shell cell representation. For the model validation, simulation results for the deformation of a spherical cell in shear flow are compared with previously published works.

Fedosov *et al.* also studied WBCs margination but they, additionally, took into account effect of RBCs [46]. It was found that WBC margination occurs mainly within a region of intermediate hematocrits and for relatively low flow rates. Moreover, simulations showed that RBC aggregation slightly enhances WBC margination, particularly at the high hematocrit values. DPD method was used to describe fluid and cell particles.

The work by Xiao *et al.* is dedicated to the mutual effect of rolling CTC in the capillary on the blood flow dynamics [247]. By the help of computational modeling of suspended CTC and RBCs, it was demonstrated that, in the microvessel of $15\mu m$ diameter, the CTC has an increased probability of adhesion due to a growing wall-directed force. However, with the increase in microvessel size, an enhanced lift force at higher hematocrit detaches the adherent CTC quickly. An increased blood flow resistance in the presence of CTC was also found. Moreover, the significant deformation induced by high flow rate and the presence of aggregation promote the adhesion of CTC. In another work, the same group of authors applied a very similar cell model to investigate an individual cell passing through a narrow slit [246]. Specifically, they studied the effect of cell size, nucleus and cell membrane shape on the transmigration through a slit. This model represents cell and nucleus as an empty shell.

Although considered models use different discretization methods, all of them employ a simple empty shell cell representation. This approach can be applied to study phenomena in tubes and microfluidic devices which are caused primarily by hydrodynamical forces. However, such models often lack details to describe processes which involve significant cell deformation such as squeezing in small clefts. In addition, they do not allow quantification of the impact of different sub-cellular components on the studied phenomena.

2.4.2 Adherent cell models

Most of cell lines considered for modeling in this section have a structural function in the body and, hence, they are relatively stiff. Among the numerical model applications, popular experiments are mechanical tests (stretching, compression), migration on the extracellular matrix, and microinjection.

Kardas *et al.* proposed a computational approach to model the structure of bone cells [104]. It was shown that the load acting on the nucleus is rising with increasing deformation applied to the integrins. The numerical simulations demonstrated that the nucleus is more affected by stress if the distribution of intermediate filaments and microtubules are random than if they are regular. The model takes into account integrins, nucleus, centrosome, and cytoskeletal proteins. Computational-wise, FEM was employed to discretize governing equations.

Ujihara *et al.* presented a cell model to study tensile and compression tests [229, 230].During the tensile test, it was observed that the total elastic energy of the model is dominated by actin fibers. The compression test revealed that the alignment of bundles of actin filaments significantly affects the cell stiffness. In addition, the passive reorientation of actin filaments bundles perpendicular to the compression induced an increase in the resistance to the vertical elongation of a cell and, thereby, increased the cell stiffness. The model incorporates a cell membrane, a nuclear envelope, and actin filaments. This is a particle-based model which utilizes minimum energy concept. The model was validated by comparing load-deformation curve obtained from simulation with experimental data.

Dowling *et al.* presented a study of the role of the active remodeling and contractility of the actin cytoskeleton in the response of chondrocytes (cartilage cells) to shear [34]. The key feature of the model is that it incorporates both passive viscoelastic component and active, describing cytoskeleton remodeling. By the help of numerical simulations, the authors showed that a purely passive cell model is incapable of predicting the response of normal chondrocytes to the stress while adding the cytoskeleton remodeling to the model gives results close to the in vitro study. Interestingly, the passive model can predict drug-treated cells with the disrupted cytoskeleton, which might be considered as evidence that actin cytoskeletal network is the main contributor to the stress resistance during discussed processes. In another paper, where a similar model was used, authors predicted the increased compressive resistance of spread cells compared with round cells [195]. The nucleus and membrane are represented as a passive hyperelastic material, while ODE describes the cytoskeleton remodeling. The equations are discretized by the help of FEM.

Fang and Lai studied the changes in the cell mechanics as a cell shifts from suspended to the adherent state [44]. In order to estimate the cell model elastic modulus, simulation of AFM experiment was employed.The force-indentation relationship was used to determine the mechanical changes in cells during state shift. The explicit modeling of the sub-cellular components allowed investigating the impact of different cell components on the resistance to the external stress as well as examine the effect of nucleus presence on the AFM results. The model consists of a cell membrane, nucleus envelope and three internal networks representing microtubules, F-actin, and intermediate filaments. In addition, it takes into account movement of adhesion molecules which allows simulating cell spreading. The model was validated by comparing the AFM simulation results with the experimental data.

Kim *et al.* developed a model to predict cell migration behavior on 2D and 3D curved surfaces [107]. The simulations revealed that the cell migration speed depends on the cross sectional area of the lumen. The relationship between migration speed and the lumen width agrees with the microfluidic experimental data. This model was also applied to study cell migration on 2D micropatterned geometries [108]. The model is assumed to be used for predictive studies to assist in the design of upcoming microfluidic cell migration assays. It takes into account focal adhesion dynamics, actin motor activity as well as cytoskeleton remodeling and nucleus. The further development of the model allowed prediction of cell invasion into 3D extracellular matrix (ECM) in response to different extracellular biochemical cues [109]. The new model takes into account filopodia penetration dynamics. The average filopodia speed was predicted, and that of the cell membrane advance agreed with experiments of 3D HUVEC migration for diverse ECMs with different pore sizes and stiffness.

By the help of numerical modeling, Liu *et al.* predicted the cell damage induced by the needle during the microinjection procedure [136]. In particular, authors quantified the effect of the size, shape of the microinjector tip, and the injection velocity on the cell damage. The proposed model is based on DPD method. It explicitly describes cell membrane as well as the cytoskeleton and motors activity. To validate the model, authors measured the mechanical properties of the model using the particle tracking microrheology. The cell model exposes power law behavior in terms of mean square displacement and lag time. The mechanical moduli obtained from the simulations are in agreement with the experimental data.

Zeng *et al.* applied cell model to study the role of the actin cytoskeleton network in mechanotransduction and nucleus deformation [255]. During the experiment, modeled in this work, a micropipette is pushed into the cytoplasm of an endothelial cell and then pulled away at a constant rate. By the help of numerical simulations, it was shown that the stress propagation through the random cytoskeletal network could be a mechanism to effect nucleus deformation, without invoking any biochemical signaling activity. It was reported that nucleus strain varies in a sigmoidal manner with actin filament concentration, while there exists an optimal concentration of actin-binding proteins that maximize nucleus displacement. In addition, a theoretical analysis for these nonlinearities in terms of the connectivity of the random cytoskeletal network was provided. This is a particle-based model which explicitly describes cell membrane, nucleus envelope, and cytoskeletal actin network. The simulation results were validated using the experimental data.

The most important common feature of the models discussed in this section is that they take into account, besides of nucleus envelope and cell membrane, mechanics of the cytoskeleton. Hence, these models can describe changes in the cell during different mechanical tests as well as cytoskeleton-related processes. In addition, they give an insight into the impact of sub-cellular components during particular experiments. What is beyond the scope of these modeling efforts is the deformation due to the flow and solid-cell interactions which might be useful for investigation of cell behavior in the flow.

2.5 Cell suspension simulations and boundary conditions

With the development of the computational methods and hardware, it became affordable to use high-resolution deformable RBC models to simulate phenomena involving numerous RBCs. This, indeed, allowed to study different blood flow effects and perform predictive simulations for microfluidic devices. There are many numerical simulations of RBC flow [165, 149, 150, 189, 52, 135, 49]. By modeling flow in relatively simple domain geometries such as a straight pipe or a periodical cube, these works created a solid background to move towards simulations in more complicated domains and, in the past 5 years, numerical modeling of blood flow in complex domains has attracted increasing attention [59, 134, 9, 117]. The limiting factor of particle-based methods is that there is no boundary conditions suitable for flow simulation in complex computational domains with multiple inlets and outlets. Periodic Boundary Conditions (PBCs) are the traditional choice, yet it might be difficult or inefficient to apply PBCs to study systems which do not have a symmetry with respect to the flow direction. Due to this methodological gap, all the mentioned flow simulations were performed in systems with PBCs along the flow direction, including some studies of simulating non-periodic blood flow phenomena [18, 96]. To study such effects as plasma skimming in capillary bifurcations, where the blood flow properties such as velocity and pressure fields differ drastically at the inlet and outlet regions, there is a need in developing alternative boundary conditions. This is a non-trivial problem, especially for particle-based Lagrangian methods.

For an open boundary system, the velocity profile at the inlet is generally specified, whereas the outflow profiles are rarely known. For a single-phase system, the inflow condition could be simply obtained by extending the inflow length so that the flow becomes fully developed at the inlet. However, for multiphase systems, the inflow conditions even for a fully developed flow are unknown - a situation similar to turbulent inflow in the single phase. For example, for a blood flow, the flow and viscous properties, as well as the cell-free layer distribution in arterioles, differ considerably with the change in hematocrit level and shear rate. Thus, the inflow length should be long enough to generate inflow condition for blood flow. As a consequence, it is inefficient to perform blood flow simulations using this relatively straightforward method because of prohibitively expensive computations.

To the date, there are a few studies that focused on the development of Open Boundary Conditions. An attempt to develop such boundary conditions has been presented by Flekkøy *et al.* [56], in which the simulation domain includes an auxiliary buffer domain for particle generation. However, the complexity of the flux control makes it difficult to perform flow simulations. Recently, a new method for such open systems has been developed by Lei *et al.* [125], where particles at the inlet are generated according to the local flux, and adaptive forces are introduced to control the flow rate at the outlet. This method has been successfully applied to single phase flow in straight channels and bifurcations [125]. However, in multiphase systems, e.g., flows with colloids, polymer chains or RBCs, it is hard to insert them at the inlet and remove them from the outlet. Thus, these two methods cannot be readily extended to the cases of complex multiphase flows.

2.6 Large scale simulations

Most of considered in this chapter cell models are computationally expensive, and simulations of the cell suspension behavior require usage of supercomputers. A prominent example is modeling of the microfluidic devices designed to catch CTCs in the blood flow. CTCs are extremely rare in the blood: there is one such cell per 10^9 or RBCs. To simulate such amount of blood, we need to use over 10^{12} particles: RBC model requires at least 500 elements to represent the dynamics of a RBC [180] and a huge amount of particles are needed to represent the fluid.

The efficiency of the model implementation is an important factor which can limit the applicability of the model. Software, which efficiently uses supercomputers, can, from one hand, reduce the time to solution and, hence, allows solving problems faster. From the other hand, such software allows modeling large scale effects which are difficult to study on a smaller scale. Another important aspect is that we need to utilize the limited and expensive computational resources efficiently. This section is dedicated to the high-performance implementations of the cell flow models. The efficient implementation of the underlying computational method is also considered here. Most of the work is dedicated to the RBCs modeling since eukaryotic cell models were not yet used in this context.

To the best of our knowledge, the first large-scale simulation of RBCs was performed by Rahimian *et al.* [190]. They used a boundary integral solution of the continuum linear Stokes equation to simulate 262 Million RBCs, each discretized with 84 vertices, totaling 88 billion unknowns. In terms of throughput, they measured about 290 Million unknowns/s, leading to a theoretical throughput of about 2.6 Billion RBC unknowns/s on Jaguar, Cray XT5. Lattice Boltzmann simulations of the blood flow into a coronary arteries network employed 450 Million non-deformable RBCs observed a throughput of 540 Million unknowns/s and 1 PFLOP/s in double precision on Tsubame [9]. Xu *et al.* performed blood flow simulations with 50,000 RBCs at the exceptional resolution of 3,300 vertices per RBC using TSUBAME2 supercomputer [248]. Large-scale rheological simulations, albeit with non-deforming RBCs, have reached 4 Millions RBCs [97] on BG/P (maximum 65,636 PowerPC nodes). Another remarkable work in this context was accomplished by LB/IB method [25], performing simulations with 220,000 RBCs and achieving a throughput of 7 Million unknowns/s.

The DPD simulations of RBCs transiting in small microfluidics channels started with modeling the flow in simple computational domains such as straight pipes [189, 48]. These simulations involved maximum 200 RBCs and provided insight on the magnitude of the adhesive forces acting between cells. In the last few years, the usage of DPD has burgeoned for the study of cellular systems and in particular for the study blood flow in microfluidic devices. These results include the first ever combination of experiments and DPD simulations in the study of large deformations of RBCs in microfluidic devices [189] and the identification of the governing mechanisms for increased blood viscosity in sickle cell anemia [126].

DPD is a very computationally intensive method and it is essential to implement it efficiently. Thus, we review here two state-of-the-art implementations of DPD. Currently, software for DPD deployed on supercomputers is based on extensions of code originally developed for MD, such as LAMMPS [183] and HOOMD-Blue [5]. LAMMPS is one of the six projects selected for the Center for Accelerated Application Readiness by the Oak Ridge Leadership Computing Facility (OLCF) and has been extended to perform DPD simulations [164]. The largest LAMMPS/DPD simulation reported 256K particles per node. They simulated 256 Million particles on 1024 Titan nodes with a throughput of 5 - 7.6 Million Particles per second [MP/s][164]. The largest and fastest DPD simulations so far had been performed with USER-MESO, LAMMPS fork by Tang and Karniadakis [223]. These simulations employed 1 Billion particles on 1024 Titan nodes with reported throughput of $10-30$ [MP/s] per node, for the simulation of spontaneous vesicle formation. Another package supporting DPD is HOOMD-Blue [5] which aims at taking advantage of the compute capabilities offered by Single Instruction Multiple Thread (SIMT) architectures [64] and has also been extended to support DPD simulations [174]. To the best of our knowledge, the fastest HOOMD-Blue/DPD simulation reported throughput 1.85 [MP/s] on a GeForce GTX 480 [174], which, projected on a Titan node, would correspond to 6.7 [MP/s].

Summarizing, although there are some large scale attempts in RBC modeling, all of them share two weaknesses - oversimplified RBC mechanics and considering too simple computational domains. Accurate, high-resolution RBC models were not used for simulations involving millions of RBCs so far, which can be partially explained by the fact that the general-purpose packages for particle systems simulations are not ready for such challenges.

2.7 Summary

In this chapter, we discussed cells composition, gave a review of cell models, reviewed boundary conditions, and large-scale simulations. Due to the simplicity of the RBC structure, the mechanics of this cell is relatively well-understood. Yet, regarding blood flow effects, there are some knowledge gaps which can be filled in by the help of numerical simulations. The applicability of the particle-based methods to study the flow problem in the complex simulation domains is currently limited to relatively simple computational domains due to the absence of the appropriate boundary conditions. Another limiting factor is the absence of efficient software that is capable to model complex flow phenomena in reasonable time. Although there is a lot of data regarding eukaryotic cells in various experimental setups, we do not have a complete understanding of their mechanics. The more in-depth knowledge of the impact of the sub-cellular components on the whole cell behavior might contribute, in particular, to the development of more efficient drugs targeting specific sub-cellular components. A better understanding of cell mechanics could also contribute to the design of the microfluidic devices tailored to catch cells with very particular alternations in composition for diseased cell isolation.

Chapter 3

Numerical methods

3.1 Introduction

Dissipative Particles Dynamics is a particle-based method that bridges the gap between Molecular Dynamics (MD) and continuum-based methods [84, 40]. There are two different interpretations of DPD. The first one is that DPD is a coarse-grained MD method. In this case, each DPD particle is assumed to represent a small cluster of atoms (up to 10) and DPD is rigorously derived from MD through the Mori-Zwanzig coarse-graining procedure [81]. In this interpretation, it has been used extensively to model complex fluids such as colloidal suspensions, emulsions, and polymers, as well as proteins folding and other systems which are typically modeled by Coarse-Grained MD [83, 173].

The second interpretation of DPD comes from the fact that all DPD forces conserve linear and angular momentum. Combined with the conservation of mass (the total number of particles is constant in simulations), DPD provides a correct description of hydrodynamic interactions in the system [40, 47]. In this case, each DPD particle can represent an arbitrary large cluster of atoms and there is no direct link between DPD force coefficients and the atomistic forces. The complex interaction between fluids, cells, and walls, are modeled with appropriate DPD parameters. Even though the values of DPD parameters do not have a direct physical interpretation any longer, equilibrium and transport properties in DPD (velocity, viscosity, self-diffusion) do have such an interpretation. There are no intrinsic units in the second interpretation of DPD, making DPD essentially scale-free method. Except for the work discussed in Appendix A, we rely on the second interpretation and treat DPD as a particle-based method, which gives us a correct description of hydrodynamic interactions. This interpretation allowed DPD to become one of the key methods for the study of the blood microrheology

[189, 258]. DPD resolves cells at a submicron scale to simulate microfluidic [213] and drug delivery systems [156], reaching time scales that have been previously accessible only to Navier-Stokes solvers [59]. In this section, we provide a brief description of DPD method and discuss MS-VE-RBC model.

3.2 Dissipative Particle Dynamics method

Fluids in DPD are modeled by particles with the choice of the number density governed primarily by the computational efficiency [71]. The time evolution of positions and velocities of DPD particles in the system are described by Newton's equations of motion,

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i \tag{3.1}$$

and

$$\frac{d\mathbf{v}_i}{dt} = \mathbf{f}_i \ , \tag{3.2}$$

where $\mathbf{r}_i$ and $\mathbf{v}_i$ are position and velocity of particle i, respectively.

To advance the system in time, we use modified velocity-Verlet scheme with $\lambda = 0.5$ [71]:

$$\mathbf{r}_i(t+dt) = \mathbf{r}_i(t) + \mathbf{v}_i(t)dt + 0.5\mathbf{f}_i(t)dt^2, \tag{3.3}$$

$$\widetilde{\mathbf{v}}(t+dt) = \mathbf{v}_i(t) + \lambda\mathbf{f}_i(t)dt, \tag{3.4}$$

$$\mathbf{f}_i(t+dt) = \mathbf{f}_i(\mathbf{r}_i(t+dt), \widetilde{\mathbf{v}}(t+dt)), \tag{3.5}$$

$$\mathbf{v}(t+dt) = \mathbf{v}_i(t) + 0.5(\mathbf{f}_i(t) + \mathbf{f}_i(t+dt))dt \tag{3.6}$$

The force $\mathbf{f}_i$ which acts on particle i is expressed by three additive parts,

$$\mathbf{f}_i = \sum_{j\neq i}(\mathbf{f}^C_{ij} + \mathbf{f}^D_{ij} + \mathbf{f}^R_{ij}) \ , \tag{3.7}$$

which are non-zero within a cutoff radius R_c. The conservative force $\mathbf{f}^C_{ij}$ is a soft repulsion force, acting along the vector between particles i and j, with a parameter a defining the maximum repulsion between the two particles,

$$\mathbf{f}^C_{ij} = \begin{cases} a(1 - r_{ij}/R_c)\widehat{\mathbf{r}}_{\mathbf{ij}}, & r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases} , \tag{3.8}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $r_{ij} = |\mathbf{r}_{ij}|$ and $\widehat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$.

The dissipative force, $\mathbf{f}^D_{ij}$, and the random force, $\mathbf{f}^R_{ij}$ are expressed as

$$\mathbf{f}^D_{ij} = -\gamma w^D(r_{ij}) < \widehat{\mathbf{r}}_{ij}, \mathbf{v}_{ij} > \widehat{\mathbf{r}}_{ij} \ , \tag{3.9}$$

and

$$\mathbf{f}^R_{ij} = \sigma w^R(r_{ij})\theta_{ij}\widehat{\mathbf{r}}_{ij} \,, \tag{3.10}$$

where $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, and $\theta_{ij} = \theta_{ji}$ stands for a random variable with zero mean and unit variance, $<,>$ is a dot product. Parameters a, γ, and σ control the strength of conservative, dissipative and random forces, respectively. The last two forces form the DPD thermostat and are related by the fluctuation dissipation theorem as $w^D(r_{ij}) = [w^R(r_{ij})]^2$ and $\sigma^2 = 2\gamma k_B T$ [40]. We use a generalized weighting function $w^R(r_{ij}) = (1 - r_{ij}/R_c)^{0.25}$ [43]. The choice of DPD parameters affects such fluid properties as viscosity and compressibility.

One of the advantages of DPD is that the fluid-structure interactions can be specified by defining corresponding DPD parameters. These parameters depend on the considered system.

In the simulations involving solid walls, the boundaries are described by the geometry of the wall coupled with static particles inside the wall. Contrary to other works, where the walls geometry described by triangulation or density gradient, we represent geometry implicitly by a Signed Distance Function, which provides a flexible way to describe domains of arbitrary complexity. One of the advantages of this geometry representation is that it simplifies the collision detection for free particles with the wall. To create wall particles, as a preliminary step, we equilibrate a homogeneous DPD fluid in the whole computational domain and, then, “freeze” particles which are inside the wall geometry. Boundary conditions are enforced through DPD interactions between wall and solvent particles and a bounce-back mechanism after the update of the particle positions [178, 235, 179, 178].

3.3 Multiscale viscoelastic RBC model

In the current study, we use MS-VE-RBC model for red blood cell flow simulations as well as a starting point for the eukaryotic cell model development. For completeness, this model is briefly reviewed below, whereas the detailed description of the model is available elsewhere [180]. The cell membrane is modeled by a triangular mesh with N_v vertices, where each vertex is represented by a DPD particle. The RBC membrane model takes into account the elastic energy, bending energy, and constraints of fixed surface area and enclosed volume, which is defined as

$$V = V_s + V_b + V_a + V_v \tag{3.11}$$

where V_s is the elastic energy that mimics the elastic spectrin network, given by

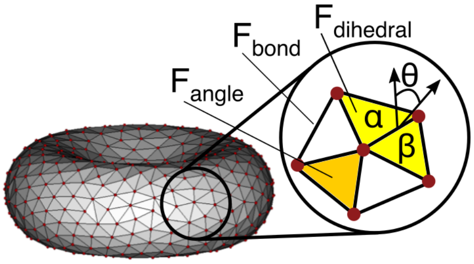

Figure 3.1. RBC membrane model made of 500 vertices. Every edge (black) is an elastic spring, every triangle (orange) contributes to the elastic energy as well as to volume/area conservation term, every couple of adjacent triangles α and β (yellow) contribute to the bending rigidity.

$$V_s = \sum_{i \in \text{springs}} \left[\frac{k_B T l_m}{4p} \frac{3x_i^2 - 2x_i^3}{1 - x_i} \right] + \sum_{\alpha \in \text{triangles}} \frac{1}{A_\alpha} \left[\frac{3\sqrt{3} k_B T l_m^3 x_0^4}{64p} \frac{4x_0^2 - 9x_0 + 6}{(1 - x_0^2)} \right], \quad (3.12)$$

where $k_B T$ is the energy unit, A_α is the area of triangle α formed by three vertices. $x_i = l_i / l_m$, $x_0 = l_0 / l_m$, where l_i is the length of spring i, l_0 and l_m are the equilibrium spring length and maximum spring extension, p is the persistence length.

The cell membrane viscosity is imposed by introducing a viscous force on each spring, which has the form,

$$\mathbf{F}_{ij}^D = -\gamma^T \mathbf{v}_{ij} - \gamma^C (\mathbf{v}_{ij} \cdot \mathbf{e}_{ij}) \mathbf{e}_{ij}, \quad (3.13)$$

$$\mathbf{F}_{ij}^R dt = \sqrt{2k_\text{B} T} \left(\sqrt{2\gamma^T} d\overline{\mathbf{W}_{ij}^S} + \sqrt{3\gamma^C - \gamma^T} \frac{tr[d\mathbf{W}_{ij}]}{3} \mathbf{1} \right) \cdot \mathbf{e}_{ij}, \quad (3.14)$$

where γ^T and γ^C are dissipative parameters, $\mathbf{v}_{ij}$ is the relative velocity of spring ends, and $d\overline{\mathbf{W}_{ij}^S} = d\mathbf{W}_{ij}^S - tr[d\mathbf{W}_{ij}^S]\mathbf{1}/3$ is the traceless symmetric part of a the Wiener increments matrix, the elements of which are from $N_{0,1}$ distribution.

The bending resistance of the RBC membrane is modeled by

$$V_b = \sum_{\alpha,\beta \text{ pair}} k_b \left[1 - \cos(\theta_{\alpha\beta} - \theta_0) \right], \quad (3.15)$$

where k_b is the bending modulus constant, $\theta_{\alpha\beta}$ is instantaneous angle between two adjacent triangles α and β having common edge (see Figure 3.1), and θ_0 is the equilibrium angle. The RBC model includes the area and volume conservation constraints, which mimic the area-incompressibility of the lipid bilayer and the incompressibility of the interior fluid, respectively. The corresponding energy terms are given by

$$V_a = \frac{k_a k_B T (A - A_0)^2}{2 l_0^2 A_0}, \quad V_v = \frac{k_v k_B T (V - V_0)^2}{2 l_0^3 V_0} \tag{3.16}$$

where k_a and k_v are the area and volume constraint coefficients. The terms A_0 and V_0 are the equilibrium area and volume of a cell, respectively.

Implementation-wise, the forces acting on the membrane particles are split into three groups according to the involved number of particles, see Figure 3.1. Bond forces are used to describe interactions between pairs of particles, such as worm-like chain (WLC) and viscous forces. The membrane bending resistance, which requires four particles interactions, is described by dihedrals. All other interactions, which involve three particles, are modeled by angle forces.

The MS-VE-RBC model is multiscale, as the RBC can be represented on the spectrin level, where each spring in the network directly corresponds to a single spectrin tetramer in the RBC membrane with the equilibrium distance between two neighboring actin connections of $\sim 75nm$. On the other hand, for more efficient computations, the RBC network can also be highly coarse-grained with the equilibrium spring lengths of up to $600nm$, as shown in Figure 3.1. The parameters for the coarse-grained model can be found elsewhere [180].

3.4 Summary

In this section, we introduced DPD method and MS-VE-RBC model which will be employed in the next chapters. DPD forms a flexible framework suitable for modeling wide range of problems. It will be used to model complex fluid phenomena in vascular channels and microfluidic devices. Since we employed mesoscopic interpretation of DPD, interaction parameters do not have a straightforward physical interpretation and are tuned manually to obtain required fluid or fluid-structure interactions properties. The parameterization of MS-VE-RBC model is systematic and particular values of parameters can be found in the literature [180, 50, 51]. This model with altered parameters, as we will see in Chapter 6, is also employed to model eukaryotic cell membranes.

Chapter 4

Open boundary conditions: blood flow modeling in bifurcations and networks

4.1 Introduction

RBCs constitute approximately 40% of the total blood volume, plasma around 55%, while the rest is taken up by white blood cells (WBCs) and platelets. Lateral migration of RBCs occurs in blood vessels, leading to the formation of two phases, *i.e.*, a core consisting mainly of RBCs and a cell-free plasma layer adjacent to the vessel wall [61, 13]. The tendency of RBCs to concentrate at the vessel center also leads to plasma skimming - an asymmetric distribution of RBCs and plasma between two daughter branches when blood flows through a microvascular bifurcation: the RBCs prefer a daughter branch with higher flow rate leaving very few RBCs (even reaching zero) flowing into lower flow rate daughter branch [38]. The blood-plasma separation in bifurcations was extensively investigated in the past few decades, and it is believed that three factors mainly determine the degree of plasma skimming that will occur [53]: feed hematocrit [188, 37, 209], size of the parent channel [53, 187] and flow rate ratio of daughter branches [61]. Studies of blood flow through bifurcations have revealed significant variability for a complete RBCs separation from the whole blood (*all-or-nothing* phenomenon). The theoretical critical flow rate ratio between the daughter branches for predicting such phenomenon is approximately 2.5:1 [188]. However, more recent experimental measurements showed that for this flow rate ratio only 88.7% of RBCs enter into the higher flow rate daughter branches [249]. These data raise the question as to what ratio value is more meaningful in determining the blood-plasma separation. These phenomena are, in particular, used to design microfluidic devices for blood-plasma separation.

In the context of simulation of the aforementioned phenomena, the choice of Boundary Conditions plays a crucial role. For modeling of the flow in bifurcations, BCs should describe the flow at the one inlet and two outlets. For modeling of the flow in part of the capillary network, proper BCs should be able to describe inflow/outflow at several inlets/outlets. PBCs, which are usually employed for particle-based simulations, cannot be used for such systems.

In this chapter, we present a general framework for open boundary systems including the inflow and outflow boundaries for particle-based approaches targeting simulations of multiphase flows. We implemented this framework in the context of parallel computations. We demonstrate that particles flowing in a complex computational domain can be treated as a system in contact with a simpler subsystem with a fully developed flow for the inflow combined with an osmotic membrane to control the outflow. This is a non-trivial enabling technology that could be used in different open boundary systems and particle-based simulations. We apply this method to perform 3D simulations of blood flow in arteriolar bifurcations and elucidated the biophysical mechanism of blood-plasma separation as well as quantify the effects of branch size and bifurcation angle on cell separation efficiency, which have not been addressed before. We also demonstrate the applicability of the methodology in arterial trees with multiple inlets and outlets. To the best of our knowledge, the existing methods cannot be used to the cases of complex flows such as blood flow.

4.2 Inflow/outflow boundary conditions

In this section, we will describe the implementation of the OBCs. For simple computational domains where PBCs can be applied, our framework is shown to exhibit exactly the same flow characteristics. Of course, what is more important is that it can be applied to domains where PBCs can not be employed. Unlike previous works, our approach has a great advantage of being applicable to both simple fluid flows and suspensions. In the proposed scheme, the computational domain is divided into three regions as illustrated in Figure 4.1. Here, the simulation of the studied phenomenon is performed in region B, while regions A and C are auxiliary. To generate an inflow in the main computational domain, we use the generating region A as a source of new particles. At the same time, when a particle leaves the main simulation domain, it enters into the region C and is removed from the system.

To illustrate the approach in more details, let us consider a microtube flow of RBCs and plasma suspension as shown in Figure 4.2. In order to have a fully

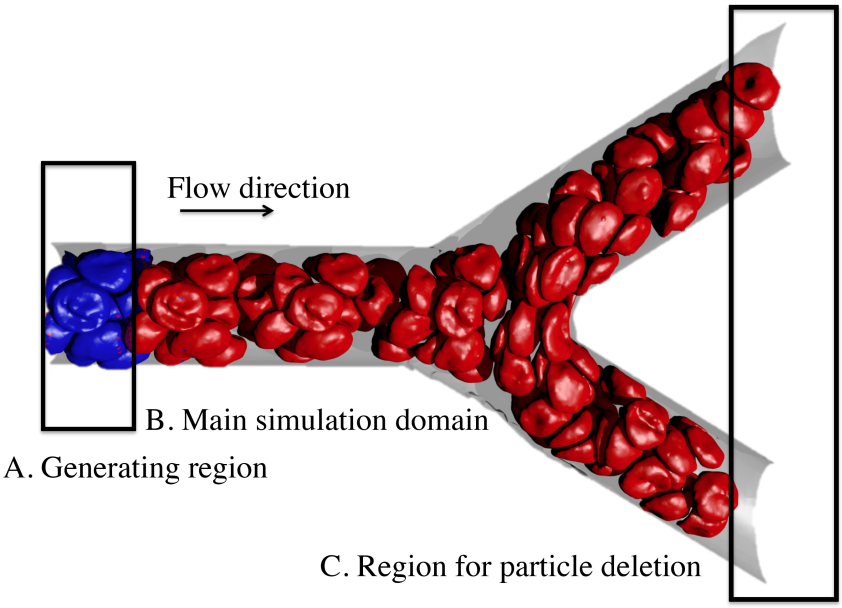

Figure 4.1. Simulation of blood flow in a microvascular bifurcation using the proposed open boundary conditions. The simulation domain is subdivided into three regions: generating region with fully developed flow, main simulation domain, and outlet region where particles are deleted. Fluid particles and frozen wall particles are not shown for clarity.

developed inflow in the main computational domain, the generating region is expected to mimic the flow in an infinite tube and be independent of the simulation in the main simulation domain. To reach these requirements, the PBCs in the generating region were implemented by ghost particles along with the particles shifting from one side of the domain to another side when they leave the periodic box. For all particles from zone A2 we create ghost particles and place them in zone A4, and a similar procedure is used for zones A3 and A1. The width of zones A1-A4 is the maximum of the cutoff radii of the force interactions used in simulations. Ghost particles are created after integration in the velocity-verlet algorithm, but before computer processors exchange forces. The independence of the generating region from the main simulation domain is achieved by turning off forces acting from particles in the main simulation domain on the particles in the generating region. At the same time, the interactions in the opposite direction are preserved because it is desirable to prevent penetration of created particles and their topological structures into the generating region. To connect the aforementioned regions, we design a procedure to duplicate particles from the generating region to the main simulation domain. That is, when a particle in the generating region crosses the copy border, a duplication of the particle is created in the main simulation domain and, by construction, the inflow is fully developed.

The extension of the proposed inflow boundary conditions for blood flow requires extra care for the cell topology. Specifically, for the ghost interaction implementation, a ghost particle, corresponding to a cell vertex, needs to keep bonded potential terms (bonds, angles, and dihedrals). In addition, the size effect of the periodic domain for computing the corresponding interactions needs to be considered. The procedure of RBC generation at the inlet is also different from the one for a single particle. First, RBC cannot be duplicated particle by particle. Instead, the whole RBC from generating region is duplicated once its center-of-mass crosses the copy border, see Figure 4.2. Second, when a new RBC has been generated in the main simulation domain, some of its vertices may still be in the generating domain. We have to pay more attention to the RBC particles and their duplications in the generating domain because they may be in close contact and cause artificial strong repulsive interactions. To avoid this, we adopt a mapping strategy and enforce the duplicated RBC to move exactly like the original RBC in the generating domain until the entire RBC is fully inside the main simulation domain. Without extra effort, the proposed procedure allows us to add RBCs and fluid naturally to the main simulation domain. It is worth mentioning that the proposed method can achieve “seamless” connection between the generating region and main simulation domain, so the RBCs flowing

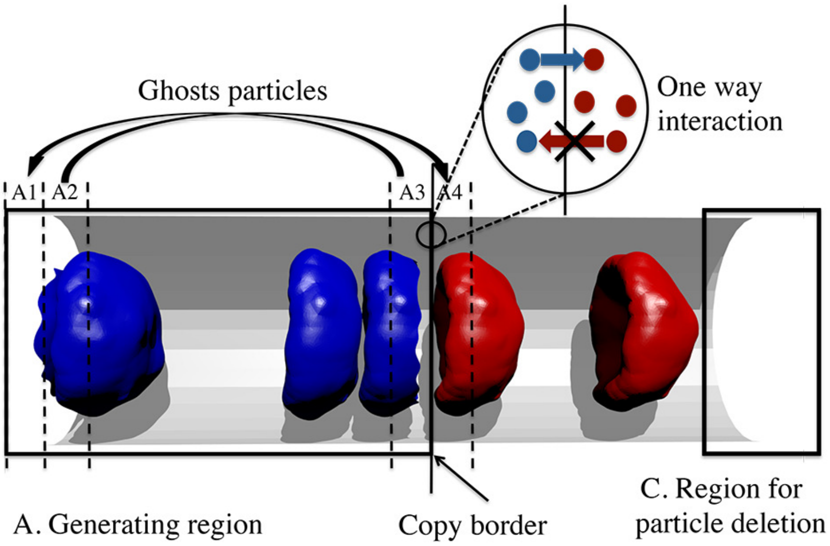

Figure 4.2. Schematic illustration of the computational domain of an open system. The generating region is divided into zones: zones A2 and A3 are sources of ghost particles while zones A1 and A4 are used for placing ghost particles. As soon as a particle crosses the copy border, its copy is created. There is one way interaction between particles in the generating region and particles in the main simulation domain.

in a complex computational domain can be treated as a system in contact with a simpler subsystem with a fully developed flow for the inflow. An increase of the length of the generating region leads to better time-averaged flow properties at the inlet, but its effect on the simulation results is insignificant since the flow has already been fully developed at the inlet. Also, it takes extra computation time.

In order to impose the outflow BCs for the simple fluid flow, we employ a method similar to Lei *et al.* [125]. Specifically, those particles leaving the simulation domain and entering into particle deletion region are reflected back to the main simulation domain with a probability $(1-P)$ depending on the particle number density ρ we want to preserve. They are computed at each iteration using the following algorithm:

1. Calculate density ρ in the main simulation domain.
2. Compute probability increment $dP = h * \frac{|\rho_{current} - \rho_{target}|}{\rho_{target}}$, where h is a weighting factor and it is set at 0.05 in this study.
3. If $\rho \leq \rho_{target}$, P is updated to $(P + dP)$; otherwise it is $P = (P$ - $dP)$.
4. For a particle crossing the outflow plane, reflect the particle back with probability (1 - P).

We note that identical outflow BCs might be implemented by applying adaptive forces in the vicinity of the outflow plane [125]; however, the proposed formulation is much simpler to implement. A similar reflecting membrane has been used to generate a fluid flow in previous molecular dynamics simulations [130]. The outflow BCs for RBCs are implemented in a different way. When the whole RBC is inside the region for cell deletion, we destroy the cell topology but leave particles in place and change their type from cell to fluid. Fluid particles are removed downstream as described above. We found that this method outperforms an alternative implementation, where the whole cell is deleted because the removal may create density artifacts in the region of cell deletion.

4.3 Validation: flow in a pipe

To validate the proposed OBCs, first, a single phase flow (without RBCs) in straight microtubes is simulated and compared with the analytical solution. Numerical simulations are carried out in a 3D geometry representing the microtube used in DPD simulations. Here, for simple fluid and, later, for blood suspension, the following DPD parameters are employed [125]: $a = 4.0$, $\gamma = 30$, $R_c = 1.5$,

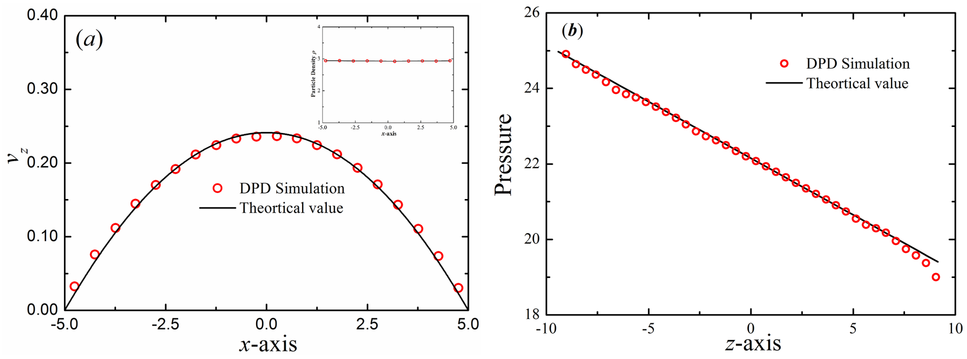

Figure 4.3. Verification of the accuracy of the proposed method. (a) Velocity profiles in the cross-flow (x-axis) direction of the Poiseuille flow corresponding to open boundary conditions. The incompressible Navier-Stokes solution is shown with lines. Inset shows the particle density profile in the cross-flow (x-axis) direction of the Poiseuille flow. (b) Pressure profile for the Poiseuille flow along the z direction. The symbols represent the DPD simulation results and the solid line represents the analytical solution.

$k_B T = 0.0945$, and fluid number density is 2.96. An external body force with magnitude of $g = 0.1$ in DPD units is exerted on each fluid particle to generate a Poiseuille flow in the microtube. The microtube diameter is $d = 10.0\ \mu m$. Figure 4.3(a) shows the average velocity and density profiles obtained from the simulations with the OBCs. The average density profile is uniform in the simulation domain, thus, the inflow and outflow boundary conditions are equilibrated and the total mass is preserved during the simulation. In addition, there is no density artifact in the vicinity of the outflow as shown in Figure 4.3(a). Moreover, the velocity profile is exactly parabolic, which agrees well with the analytical prediction, which proves the correctness of the scheme. For more quantitative analysis, we compute the pressure profile along the flow (z-axis) direction (see Figure 4.3(b)). The DPD simulation results are in good agreement with the analytical prediction given by $dP/dz = 8v_{max}\mu/D^2 = ng$, where μ is the viscosity of the DPD fluid.

Having verified a single phase flow, we simulate the motion of RBC suspension through a straight tube. Figure 4.4 shows the average velocity profiles for blood flow at two different hematocrit levels (H_t), *i.e.*, $H_t = 15.0\%$ and 30.0%. In this figure, quasi-parabolic (or flat pug-like) shapes of the typical velocity profiles of blood flow are shown. These results indicate that the blood flow in microtubes can be accurately implemented by the described scheme.

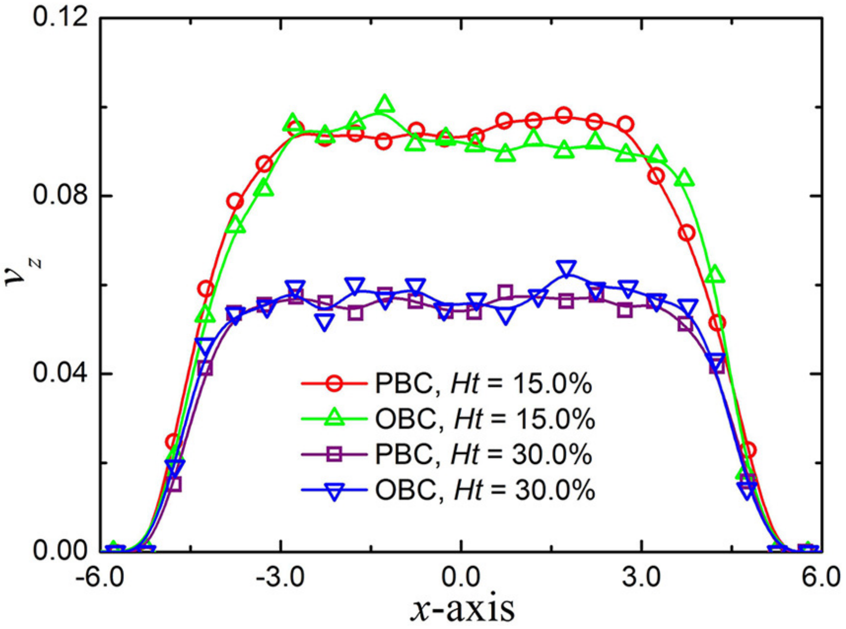

Figure 4.4. Validation of the open boundary conditions. Typical velocity profiles of blood flow in microtubes at $H_t = 15.0\%$ and 30.0%. The simulation results are compared to those in periodic systems at same hematocrit levels. x and z represent the radial and axial distances for the cylinder geometry; v_z is the velocity along the flow direction.

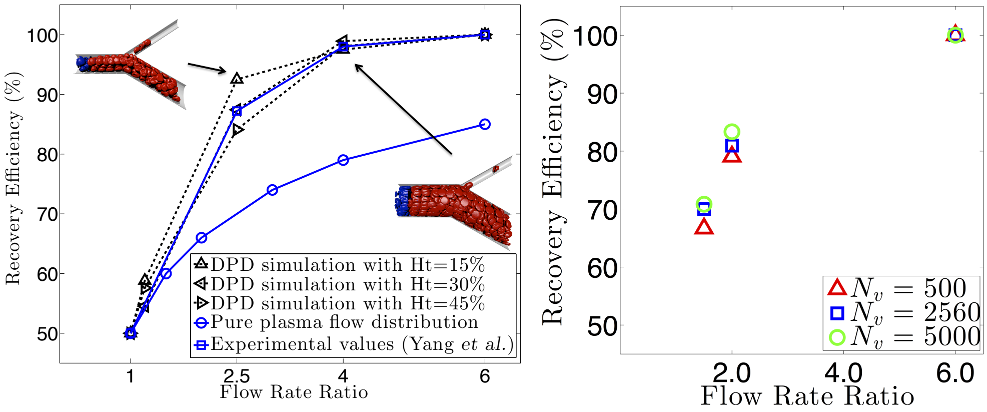

Figure 4.5. Particle recovery efficiency with respect to flow rate ratio. (a) Particle recovery efficiency at different hematocrit levels. Two snapshots of the RBCs at microvascular bifurcations with flow rate ratios of 2.5 and 4.0 at H_t = 45.0% are shown. Simulation data (black squares) from [133] are shown. In the figures, red and blue particles belong to RBCs. Fluid and solid wall particles are not shown for clarity. (b) Particle recovery efficiency at different levels of coarse-graining of the MS-VE-RBC model at H_t = 15.0%. The simulations are conducted using the MS-VE-RBC model with N_v = 500, 2560 and 5000.

4.4 Application: flow in the bifurcations

Next, we apply the proposed OBCs to model the blood flow in microvascular bifurcations. We used flow rate ratios, ϕ_d between branches $1.0, 1.2, 2.5, 4.0$ and 6.0 to study the particle recovery efficiency (the proportion of parent RBC flux entering each daughter branch) with respect to the flow rate ratios between two daughter branches. We perform simulations with three hematocrit levels (H_t = 15.0%, 30.0% and 45.0%) and find that the higher flow rate ratio yields the higher particle recovery efficiency. A 100% recovery efficiency is achieved for the flow rate ratios starting at 6:1. We also demonstrate that the higher the hematocrit is, the higher is the probability for RBC to follow a lower flow rate branch (see Figure 4.5(a)). We also simulate the blood flow and study the particle recovery efficiency at different levels of coarse graining with N_v = 500, 2560 and 5000, using the MS-VE-RBC model, see Figure 4.5(b). We find that 100% recovery efficiency can be achieved for all three cases at ϕ_d = 6:1. We find that a slight decrease in particle recovery efficiency for suspensions of RBCs with finer DPD resolution at lower ϕ_d. For example: for $\phi_d = 1.5$, 66.8% particle recovery efficiency is obtained for the MS-VE-RBC model with $N_v = 500$, which

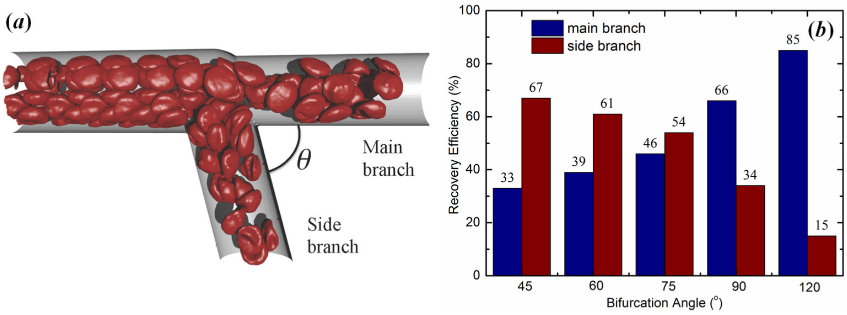

Figure 4.6. Effect of bifurcation angle on particle recovery efficiency. (a) A sketch of the microvascular bifurcation model by changing the bifurcation angle θ. In this model, the diameter of the parent branch is 20.0 μm, and the diameters of two daughter branches are both 16.5 μm. The average velocity of blood flow in parent branch is about 0.12 mm/s. (b) Relationship between the particle recovery efficiency and bifurcation angle.

decreases to 65.1% with $N_v = 2560$ and to 62.5% with $N_v = 5000$; it reduces from 79.2% for $\phi_d = 2.0$ with $N_v = 500$ down to 76.3% with $N_v = 2560$ and to 74.9% with $N_v = 5000$.

The blood flow and stress characteristics in human arteriolar bifurcations are affected by the branch location and bifurcation angle variation [171, 209]. The proposed approach offers a good method in investigating these effects on the behavior of RBCs flowing through a microvascular bifurcation. Here, we control the flow rate ratio by changing the bifurcation angle, θ, between the two daughter branches (see Figure 4.6(a)) at a fixed flow rate in the parent branch. We find that the particle recovery efficiency is clearly different in bifurcation with small and large angles, see Figure 4.6(b). More RBCs move into the side branch at a smaller angle bifurcation, while more RBCs move into the main branch at a larger θ. A critical angle value can be estimated at $\theta \approx 78^o$ when nearly half of the RBCs move into each branch. Therefore, our 3D DPD simulations demonstrated that the bifurcation angle influences RBC flux to the daughter branches so that it's effect on the RBC flux distributions in microvascular bifurcation cannot be neglected.

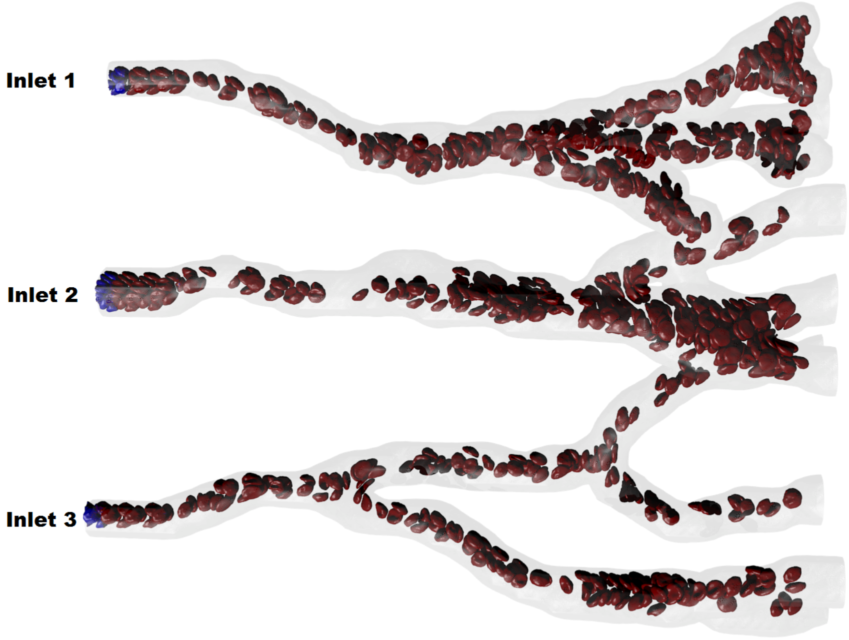

Figure 4.7. Snapshot for simulation of the blood flow in part of arterial network with three inlets and multiple outlets. The complex microvascular network was constructed using the angiogenesis model.

4.5 Application: flow in capillary networks

Finally, we demonstrate that the proposed approach can be employed to target blood flow simulations for multiple inlets and outlets. Here, for illustration purposes, we present a simulation of blood flow in an arteriole network, which was constructed using the angiogenesis model [153]. As shown in Figure 4.7, the network has three inlets and multiple outlets. Thus, the developed OBCs can be used to simulate blood flow in vascular network.

4.6 Summary

In this chapter, we have described a general framework for open boundary conditions including the inflow and outflow boundaries for particle-based methods [140]. This approach offers a straightforward way for open system simulations such as blood flow in microvascular bifurcations, which opens the possibility to simulate particulate flows for various systems with open boundaries. It was implemented as an extension to LAMMPS [183] and extensively tested in simulations at different domains in the High-Performance Computing environment. To the best of our knowledge, this is the first simulation of blood flow in an arterial network with the MS-VE-RBC model. There are some disadvantages to consider. First, the inflow at the inlet is duplicated from a fully developed flow generated in a pilot simulation with PBCs, the proposed technique is therefore inappropriate when modeling time-dependent non-periodic flow system, such as the transient phenomena in human arteries. Second, for a particle fluid system with non-periodic open boundaries, the flow rate at the outflow boundaries need to be controlled, the proposed technique may be inefficient for blood flow in arterial trees with numerous outlets.

Chapter 5

Large scale blood flow simulations in microfluidic devices

5.1 Introduction

Although some cells flow phenomena can be studied numerically considering only a few cells, other require modeling of numerous interacting cells. Even to model a small part of microvasculature, like one shown on the Figure 4.7, we employed a thousand of RBCs. As has been discussed in Chapter 2, existing large-scale simulations of RBCs flow employ naive, usually rigid, RBC models and describe flow in simple computational domains. This motivated us to work on large-scale blood flow simulations using validated MS-VE-RBC model to study flow in microfluidic devices for Circulating Tumor Cells detection. Due to the fact that the existing general-purpose software packages for particle-based modeling are not computationally efficient enough, we developed a new software package, called uDeviceX. This chapter is based on the paper published in 2015 and, hence, the performance data was obtained on supercomputers Titan and Piz Daint before later hardware upgrades, see the relevant configuration of these machines in Section 5.2. For comparison, we used LAMMPS and HOOMD-Blue packages of the version available at that time.

State of the art DPD solvers are typically based on extensions of MD software packages such as LAMMPS [183] and HOOMD-Blue [5] targeting both CPU-only and GPU-accelerated supercomputers. However, while MD and DPD share similarities as particle-based solvers, the physics represented by their interaction potentials have significant differences. These differences translate into a number of challenges for the effective parallelization of DPD. As an example, in DPD the overall exposed Instruction-Level Parallelism (ILP) is higher than in MD and it can be extracted by the current CPU and GPU microarchitectures.

GPU-accelerated supercomputers exhibit significant differences from those based solely on CPUs in terms of throughput and latencies of the different instructions. On the CPU, integer instructions are usually the fastest, while double precision (DP) instructions are among the most expensive and single precision (SP) instructions have twice the throughput of the DP instructions. In contrast, on the GPUs of current supercomputers, SP instructions are the fastest. These SP instructions take just one cycle and the throughput imbalance between SP and DP instructions is 3:1, instead of the more common 2:1. Furthermore, on GPUs, many integer instructions are as expensive as the DP ones.

These imbalances suggest that current supercomputer GPUs have been designed to process SP instructions more efficiently than any other instructions. This leads to counter-intuitive observations regarding the "traditional" performance metrics in supercomputing. For example, a compute kernel featuring exclusively DP instructions that is capable of achieving 100% of the nominal DP peak performance, is actually performing at 1/3 of the potential compute power offered by the GPU. This challenges metrics based on DP, since the performance achieved by such a DP kernel is arguably 33% of the nominal throughput. Hence, in this work we use the vector instruction per second [I/s] and instruction per cycle [I/c] (or IPC) as performance metrics.

We developed a high-throughput software for DPD simulations based on thoroughly optimized kernels for GPUs. Our code simulates for the first time blood flow and circulating tumor cells in realistic geometries of complex microfluidic channels at submicron resolution and times that are relevant to clinical applications. The software maps effectively the DPD method to the 18,688 GPUs of the Titan supercomputer by overcoming the following challenges:

- The irregular computational patterns inherent to DPD are not readily data-parallel. These irregularities prevent the simulations from reaching high fractions of the peak performance.

- DPD involves particle interactions that require hundreds of instructions. This is an order of magnitude more instructions than Lennard-Jones interactions in MD simulations.

- DPD accesses time scales that are several orders of magnitude larger than those accessible by MD. At the same time, in DPD simulations, the particle neighborhood changes practically at each time step. Hence during the evaluation of the interactions around 80% of the instructions are spent in identifying possible neighbors.

- The DPD thermostat (dissipative and random forces) is the most expensive part in the computation of DPD interactions. It consists mostly of integer instructions which do not map efficiently on current GPUs.

- DPD implies challenging inter-node communication patterns. It requires non-trivial CPU-GPU cooperations to pack/unpack messages, and hundreds of MPI point-to-point communications per rank and time step. Message sizes are not known a-priori, preventing a straightforward Computation/-Transfer (C/T) overlap.

The developed DPD solver effectively exploits the capabilities of the Titan supercomputer and allows simulations at spatiotemporal scales that are impossible for MD, while maintaining molecular level details that are inaccessible to continuum solvers. The solver was applied to accelerate the design and testing of microfluidics devices by order of magnitude while resolving phenomena at submicron scales. We focused on RBCs and Circulating Tumor Cells simulations in microfluidic devices employing rheology of cell suspension.

The Chapter is structured as follows: in Section 5.2 we describe the computing platforms used for our simulations. In Section 5.3 we report key design features of the present software. Next, we overview the considered microfluidic devices, Section 5.4. In Section 5.5 we present the results of our simulations. Finally, we demonstrate our performance results in Section 5.6 and provide conclusions in Section 5.7.

5.2 Target platforms

The present simulation software was developed targeting two supercomputers: Titan, the fastest supercomputer available in 2015, from Oak Ridge National Laboratory; and Piz Daint, the fastest supercomputer in Europe, from the Swiss National Supercomputing Centre (CSCS). These supercomputers have a nominal aggregate CPU-GPU peak rate of 39 and 12 normalized[1] PetaInstructions/s respectively (see Table 5.1).

The per-node performance for Titan and Piz Daint is primarily attributed to the NVIDIA K20X GPU, a SIMT architecture which dispatches the computation to its 14 Streaming Multiprocessors (SMX), each containing 192 CUDA cores. Each CUDA core is capable of performing 1 [SP FLOP/c]. The K20X GPU is based on

[1] #nodes × (#CPU cores × CPU-freq × SIMD-width × 2 [I/c] + GPU-freq × 14 SMX × 192 CUDA cores × 1 [I/c]).

Table 5.1. Target platforms.

	Titan	**Piz Daint**
Model	Cray XK7	Cray XC30
Host/CPU	Opteron 6274	Xeon E5-2670
Device/GPU	K20X	K20X
Nodes	18,688	5,272
Inter-node bw. [GB/s]	2.5	8.1
Aggregate GPU [I/s]	3.7×10^{16}	1.0×10^{16}
Aggregate CPU-GPU [I/s]	3.9×10^{16}	1.2×10^{16}

Type	**Model**	**[GB]**	**[I/s]**	**[I/c]**
GPU	K20X	6	2.0×10^{12}	2.7×10^{3}
CPU	Opteron 6274	32	0.14×10^{12}	64
CPU	Xeon E5-2670	32	0.33×10^{12}	128

the NVIDIA Kepler microarchitecture, CUDA compute capability 3.5, featuring 6 GB of device memory.

Each SMX can schedule the work of 16 CUDA blocks concurrently, fragmented into warps of 32 threads. At every cycle, the four per-SMX schedulers pick four eligible warps out of a pool of 64, and fetch up to two instructions each, totaling a maximum of six warp-level instructions per SMX. Because of the dual-issue nature of the schedulers, the compute kernel must expose enough ILP to reach the peak throughput. To maximize the number of eligible warps, and thus maximize the available work per cycle, each thread should require no more than 32 registers. The measured peak device bandwidth is about 180 [GB/s] and the measured peak throughput $1.8 \cdot 10^{12}$ [I/s]. The best measured per-warp throughput was 5.5 [I/c] as opposed to the nominal one of 6 [I/c].

The K20X exhibits a 3-to-1 throughput imbalance of DP versus SP instructions. We remark that in terms of throughput, integer instructions are 1.2X - 6X slower than SP instructions. These observations suggest that the K20X was designed to aggressively process SP instructions, over any other instructions.

The XK7 and XC30 nodes can effectively exploit the “zero-copy” capabilities of the K20X, avoiding explicit data transfer between the host (CPU) and the device (GPU). The zero-copy feature is beneficial as it allows kernels to pack/unpack MPI-messages directly from the device and hence can hide the cost of the memory transfer by overlapping it with computation. In addition, both XK7 and XC30 nodes benefit from the Hyper-Q technology [20] and a multi-process service [166] enabling multiple MPI tasks to access the same GPU concurrently.

The irregular computational patterns of DPD exhibit a massively parallel, fine-grained heterogeneous workload. This heterogeneity leads to a number of control flow exceptions that is excessively large to be effectively handled explicitly on SIMD architectures. Even if masking instructions alleviate this problem, the burden of dealing with masks inadvertently impairs both coding productivity and kernel performance.

Furthermore, compared to SIMD architectures, SIMT provides a number of unmatched advantages, starting with the ability to discover vectorization at runtime. The detection of exceptional cases in the computation (such as data boundaries, misalignment, etc.) is delegated to the hardware. This dramatically increases the readability of the code and the productivity of the programmer. A second advantage is the ability of scattering/gathering data to/from arbitrary locations in the shared memory. These operations enable effectively to "homogenize" the irregular DPD workload into data-parallel-ready computation. Finally, with control flow instructions at warp-level, the cooperation among the threads within a warp can be dynamically adjusted to the available workload so as to minimize the overhead or the intra-warp imbalance.

5.3 Software design

The complexity of flows, involving suspended RBCs and CTCs in microscale geometries, impacts the performance of the simulations adversely. In turn, this poses significant challenges in mapping effectively the computational patterns onto GPU accelerated supercomputers.

A number of design choices have been made in order to significantly improve the performance:

- Computations are performed in single precision
- No global particle ids are used
- Neighbor lists are stored in on-chip memory
- Fast Random Number Generator (RNG) is employed
- Each kernel operates at most on two particle types
- Cell lists involve particle reordering
- Load balancing is achieved by rank reordering

Time steps in DPD simulations are usually between 10^{-3}-10^{-2} in DPD time unit, while the magnitude of the forces rarely exceeds 10^{2}. It means that it takes hundreds of thousands of time steps to resolve one cycle of the tumbling of a single RBC. This suggests that the single precision mantissa is large enough (10^{6}) to capture the difference in the orders of magnitudes across the unknowns of the simulation. This observation renders the vast majority of the DPD computational patterns amenable for SP computation, which in turn accelerates up to 3X the time-to-solution.

A global particle id may simplify the management and diagnostics of the simulation. However, it represents an unnecessary overhead, when involved in the computation of the forces or the dynamic creation of new solvent particles as well as new RBCs or CTCs. Similarly, the overhead of keeping a per-particle neighbor list is hardly amortized if we consider that high flow strain rates impose a complete reconstruction of the neighbor lists at every time step. Furthermore, the cost of the DPD interactions can be straightforwardly hidden by the irregular work required for producing the neighbor list.

The random force of the DPD interactions requires a computationally efficient random number generator (RNG). Among the algorithmically most efficient RNGs are those based on Tiny Encryption Algorithms (TEA) [239], such as the SARU algorithm [1]. For the K20X, a performance issue of the TEA-based RNGs is that they mostly feature integer instructions. As described below, we have developed a novel RNG based on the Logistic map which features exclusively FMA instructions.

In order to achieve high performance in the compute kernels we restrain them to operate concurrently on just two particle types. This enables us to apply type-specific optimizations when computing the interactions. These compute kernels rely heavily on cell-lists. The construction of cell-lists involves a data-reordering stage to enforce optimal spatial and temporal locality.

Most of the workload imbalance is expected to come from the irregularity of the microscale geometry. Since the geometry is known a priori, we introduce a preprocessing step to provide an optimized rank-to-node mapping to the application-level placement scheduler. The reordering assigns an heterogeneous number of MPI tasks to each node so as to equalize the per-node workload.

The software implements Lustre-aware parallel I/O for data dumps, whose formats are compliant to ParaView [6]. The mesh data generated by the present simulations was rendered with the Mitsuba Renderer [92].

5.3.1 Memory layout

The domain of the present simulations is decomposed in a Cartesian fashion in parts of equal size and shape. The memory layout of the present simulations has the following properties:

- Array-of-Structure (AoS) format for the particles
- Particles are represented contiguously in memory

- There is one stream for each particle type
- The representation of the wall is entirely local

We employ an Array-of-Structure (AoS) format that provides us with the best trade-off between spatial locality and effective memory access for DPD simulations. This layout is also reasonably efficient when packing/unpacking MPI messages containing particle information.

The representation of the wall geometry is entirely local to its respective subdomain. Once created the wall data are immutable objects. Hence, after the wall creation the associated information can be exchanged only once across neighboring MPI tasks, in a fashion similar to a halo exchange. This renders the handling of collisions between walls and DPD fluid as well as cells entirely local to each MPI task.

5.3.2 Cluster-level optimizations

At this level, the design choices aim to minimize the amount of communication and achieve maximum C/T-overlap. For the computation of the forces within the solvent, C/T-overlap is achieved by exchanging messages of a predefined size. After packing the messages on the GPU, each MPI task performs non-blocking point-to-point communication with its 26 surrounding tasks while computing the local DPD forces. Since the force evaluation takes $1-10$ ms and the communication takes $0.01-0.1$ ms, we expect the messages to arrive before the local computation has been completed. We then compute the inter-node DPD interactions by fetching directly the MPI messages from the GPU using zero-copy memory.

In certain cases, the number of solvent particles to be communicated exceeds what can be held by a single message. These cases are safely detected and handled by sending a second message in a blocking fashion, whereas the C/T-overlap may be decreased for that timestep.

Simpler C/T-overlap techniques are used to compute the interactions between solvent and cells. In this case, we do not employ predefined message sizes. The simulation patterns associated to RBCs and CTCs are split into communication and computational stages. Hiding the communication stages with the computational ones is relatively straightforward by pairing the stages.

5.3.3 Node-level optimizations

At node-level, the simulation code employs multiple CUDA streams to coordinate the execution of the different compute kernels. We rely both on the operating

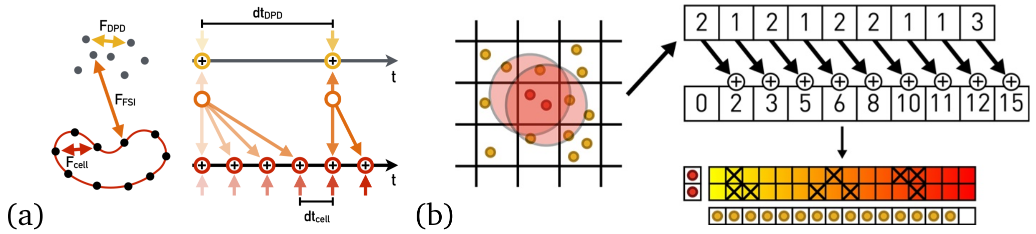

Figure 5.1. (a) The separation of the temporal scales between solvent and cells allows us to consider multi-timesteps algorithms [206]. (b) The workload of a cell is mapped to a warp so as to decrease warp divergence and work imbalance. Firstly, the particle count of the surrounding cells is fetched, then the prefix sum is computed.

system and MPS to effectively run multiple MPI tasks with heterogeneous workloads on the node.

We exploit the separation of the temporal scales between the solvent and the RBC membrane since the viscosity of the latter is $10-100$ times larger. Our time scales separation approach is similar to the one discussed by Schlick [206]. When considering diluted suspensions of cells, time-to-solution can be shortened by taking advantage of the separation of the temporal scales within the simulation. The DPD dynamics is significantly slower than the viscoelastic response of the deforming cells. Since the computation of the membrane forces is relatively inexpensive ($1\%-5\%$ of the total workload), we are able to increase the DPD time step by a factor of 10 and updating just the cells at substeps of $1/10$ of a time step. As illustrated in Figure 5.1(a), for a coarse time step, we first calculate the total external forces (from the solvent and wall particles) on the cells, and then evolve the cells by adding internal forces of the cells to the external ones and integrating at finer time steps. These finer steps are entirely local to the node and do not require any MPI communication. The bounce-back ensuring the no-through condition is also performed at every sub-time step.

5.3.4 GPU-level optimizations

The most time-consuming kernels are those which compute the interaction forces: DPD, fluid-structure interactions, and walls. The DPD kernel computes the DPD interactions within each subdomain. The FSI kernel computes the interactions between the solvent particles and the particles representing the suspended cells. The Wall kernel is responsible for computing the interactions between the DPD particles and the wall particles. These kernels take advantage of Newton's 3rd law and work cooperatively on the cell-lists, minimizing warp divergence and maximizing temporal and spatial locality.

Figure 5.1(b) depicts how the computation of a cell list entry is mapped to a warp. A reindexing of the workload is performed through the shared memory and consists of a prefix sum of the cell count of the 27 surrounding cells. The result is then shared across the CUDA threads, which are organized into an interaction matrix (16x2 in the case of Figure 5.1(b)). For every thread, the corresponding source particle is found by performing a trinary search of the prefix sum.

All compute-bound kernels were subject to a "floatization" process, where integer instructions are replaced by FP instructions. This is implemented by reinterpreting integer variables as denormalized SP numbers and perform SP operations at full precision. The correctness of such operations is guaranteed by using inline PTX assemblies with explicit rounding and flushing modifiers.

Algorithm 1 Logistic based parallel RNG for DPD.

```
// $k = k(t)$ is a time-dependent global random number sequence
// $G$ is the golden ratio $\frac{\sqrt{5}-1}{2} \approx 0.618\cdots$
// $S$ is another irrational number $\sqrt{2}-1 \approx 0.414\cdots$
function Mean0Var1(i,j,k)
  // Low-discrepancy number
  $u \leftarrow$ Min(i,j)
  $v \leftarrow$ Max(i,j)
  $p_{ij} \leftarrow$ Mod($u \times G + v \times S$,1)
  $y \leftarrow k - p_{ij}$

  // Pass through Logistic map
  for N rounds do
    $y \leftarrow 4y(1-y)$
  end for

  // Normalize
  $z \leftarrow$ Normalize($y$)
  return $z$
end function
```

We developed a novel stateless RNG that consumes mostly FMA instructions by using the Logistic map [66, 236], a non-linear chaotic recurrence relation defined for real numbers. The series $X_{n+1} = 4X_n(1-X_n)$, $X_n \in [0,1]$ maps the interval $[0,1]$ onto itself with a probability density function $p(x) = (\pi x\sqrt{1-x})^{-1}, x \in [0,1]$. The recurrence relation is chaotic and hence generates unpredictable sequences after a sufficiently large number of iterations. The distribution is symmetric on $[0,1]$ with a mean of $\frac{1}{2}$ and a variance of $\frac{1}{8}$. A random number sequence of zero mean and unit variance can thus be obtained by simply shifting and scaling X_n. We introduce Algorithm 1 for generating a unique sequence of random numbers for *each* pair of interacting particles. The algorithm combines a time-dependent global seed with particle indices into pairwise-unique seed values, which then get crunched through the Logistic pipeline for further randomization. A serial high-quality RNG, in our case KISS [145], generates the global seed since only one value is used for each time step.

The generator passes 158 out of 160 tests in BigCrush [122] with 18 Logistic iterations and achieves zero mutual information across sequences for different particle pairs.

5.4 Microfluidic devices

The quest towards miniaturization of devices capable of handling fluids and biological matter at micron resolution is revolutionizing medicine and pharmaceutical research. Microfluidic systems enable precise analysis and large-scale automation for important biochemical processes [240, 215]. By benefiting from an economy of scales, microfluidics is advancing technologies ranging from clinical diagnostics and cancer detection to regenerative medicine and proteomics. Microfluidic devices exploit the functional properties of fluids confined in microscopic channels to perform tasks such as blood separation, detection of circulating tumor cells, and molecular recognition with high sensitivity and specificity [212, 24]. However, the acquisition of information of the fluid micro-rheology, including phenomena such as cell merging, lysis [159] and biochemical signaling [254] remains a formidable task [175]. Without such information, the unique, and often unexpected, properties of fluids at the microscale can be a hindrance for new designs while complex prototypes may be impossible to manufacture at an economy of scales. Fast turnaround times for design are essential for the evolving technologies (3D printing, organ-on-a-chip) of microfluidic chips [39]. Predictive simulations are becoming an invaluable tool in accelerating the design cycle for microfluidics [105]. For the numerical investigations in this work, we will consider two devices, relying primarily on the cell suspension rheology.

The first device, CTC-iChip, relies on arrays of pillars forming crescent-shaped isolation wells [222], as shown on Figure 5.2 (a). Each of the wells has gaps of $5\mu m$ allowing small and deformable RBCs to sieve through while CTCs are isolated. The microscopic picture demonstrates the device state at the end of the filtration so all the wells are occupied by CTCs. It exploits geometries where target cells follow streamlines different than those for the other blood cells [103]. Although the actual device consists of two connected components, called CTC-iChip1 and CTC-iChip2, we consider only the first part. CTC-iChip1 separates a

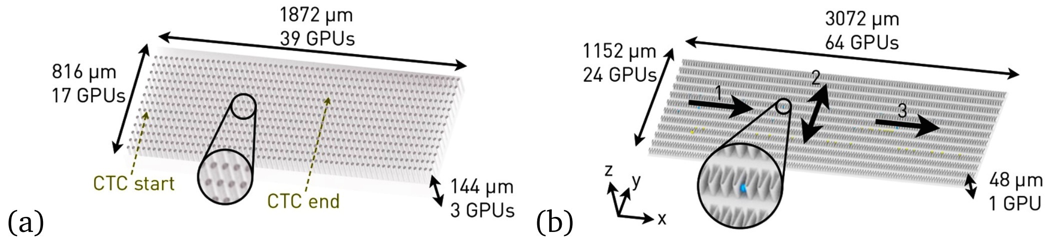

Figure 5.2. Device I: (a) Stage 1 of the CTC-iChip indicating the initial and final position of the 1 CTC spiked in 200,000 RBCs. Device II: (b) Funnel ratchets geometry.

few CTCs and WBCs from hundreds of thousands of RBCs by taking advantage of the deterministic lateral displacement effect [86]. The separation is achieved through the precise placement of multiple arrays of obstacles in a laminar flow so bigger cells, such as CTCs, are displaced to a different outlet than other cells. Pillared egg-shaped obstacles form a microarray of 13 rows and 118 posts along the module, as shown on the Figure 5.2 (a).

Another device, funnel ratchet, exploits array of obstacles with a parabolic shape and narrower channels [147] to filter various CTCs, WBC, and benign cells. A microscopic picture of a part of this device with stacked cells is shown on Figure 5.2 (b). The geometry consists of a 2D array of funneled constrictions with 128 funnels arranged in 12 rows. The funnel pore size is constant in each row, and in each successive column the pore size decreases by 1 μm from 12 μm to 2 μm. First, a dilute solution of a mixture of cells is infused into the sorting area before the first row. The pressure difference is applied to the system in forwards direction followed by short period of backward pressure. At the end of the forward stage, cells which have distinct mechanical properties are stacked in different rows of the device, as shown by red circles on the Figure 5.2 (b). After the following backward stage, these cells are located in between rows where they stacked and can be collected by applying a horizontal flow.

5.5 Large-scale simulations

5.5.1 Device I: CTC-iChip

In this part, as a proof of the capabilities of our software, we simulate flows of RBCs and CTCs in the CTC-iChip [103]. The one-to-one geometric correspondence between the CTC-iChip and the geometries studied herein, allow us to

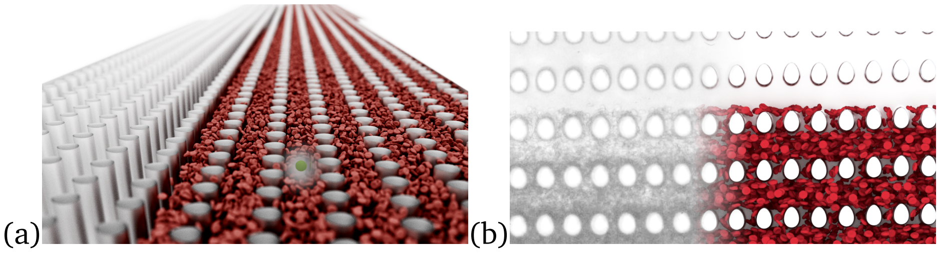

Figure 5.3. (a) Simulation of the first module of the CTC-iChip with 200,000 RBCs spiked with 1 CTC. (b) Experiment [103] of RBC flows in the CTC-iChip geometry (left part) and simulation (right part).

confirm these features.

The size of the part of CTC-iChip1 device, considered for simulations, is 150 μm tall, $\sim$ 750 μm wide and at least 3.8 mm long. Pillared obstacles form a microarray of 13 rows and at least 118 posts along the module, as illustrated in Figure 5.2 (a). The post cross sections are “egg”-shaped of size 17 μm along the X-direction and 24 μm in the Y-direction. This microarray is characterized by a gap of 32 μm between the rows, 15 μm between the columns and a 1.7° slope with respect to X-axis. This geometry affects the deflection of particles larger than 4 μm [87].

In simulations, the domain size is $1872\mu m \times 816\mu m \times 144\mu m$ with an array of 767 (59×13) obstacles, see Figure 5.2 (a). The computational box is bounded in the Y and Z directions by flat walls, with periodic boundary conditions applied in X direction. The flow is driven by a uniform pressure gradient along the X-direction.

We initialize 200, 000 RBCs at uniformly random positions in the leftmost part of the domain with respect to Y direction (Figure 5.3(a)). We insert among the RBCs a single CTC with a diameter of 14 μm. The simulation entails for 3×10^6 timesteps during which the CTC traversed the whole domain in the whole device $\sim$ 7.5 times. We observe that RBCs were primarily located around the obstacles at the bottom in accordance with the experimental snapshot, as depicted in Figure 5.3(b). In Figure 5.5 (a) we report the displacement of the CTC and RBCs in the Y-direction over time in order to assess the effectiveness of the device in separating the CTC from the RBC cells. We note that the CTC was drastically displaced in the Y-direction (Figure 5.5 (a)) allowing its separation from the RBCs in agreement with the experiment. Eventually, few RBCs diffuse across the entire device; however, most of the RBCs remain in the bottom rows as shown in Figure 5.5(b).

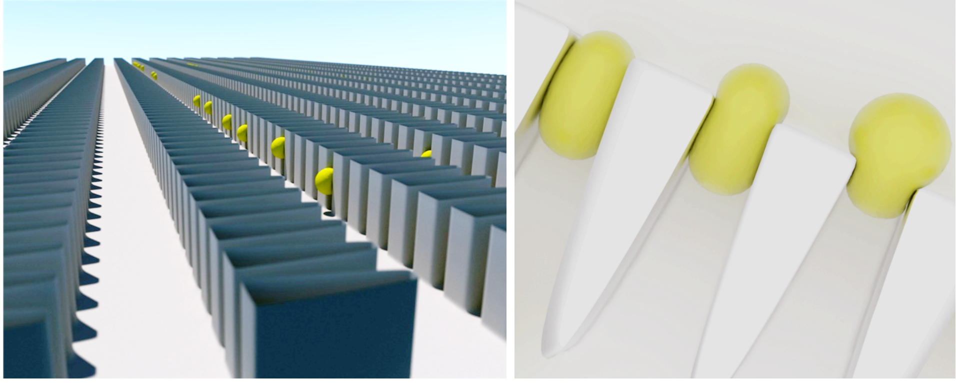

Figure 5.4. Device II, simulation of the funnels ratchet (left), CTCs are squeezing through funnel constrictions (right).

5.5.2 Device II: Funnel ratchets

We perform a numerical investigation of CTC filtering in funnel ratchet device by discretizing the domain into $1152 \times 3072 \times 48$ length units. Shape and separation of the tapers are exactly as they appear in the experiment. The box is bounded by flat walls in the Z direction, while PBCs are applied in X and Y directions. This simulation does not involve RBCs, it is used to further isolate CTCs from WBCs [147].

In this simulation, we initialize ~ 70 cells of two different types uniformly along the Y direction in front of the first row of obstacles. Cells of type 1, mimicking WBCs, have a diameter of $14\,\mu m$, while cells of type 2, corresponding to CTCs, have a diameter of $20\,\mu m$. Type 2 cells are about 20% stiffer than type 1 cells. The period of the oscillations is 1500 DPD time units and the positive force is maintained for 1125 time units, while the remaining 325 time units the force is reversed. We run $2.9M$ timesteps which correspond to 9 full oscillatory cycles.

We clearly observe cell separation even after the first cycle as illustrated in Figure 5.5 (c), even though irreversibility is achieved only after the fifth cycle. Due to the flow properties and random initial conditions, type 2 cells end up getting trapped between the funnels as shown in Figure 5.4.

5.6 Performance results

The present software and simulations have enabled the following achievements:

- *Peak performance:* our software reaches 65.5% of the nominal peak for the most computationally-intensive kernel and 34% overall nominal peak

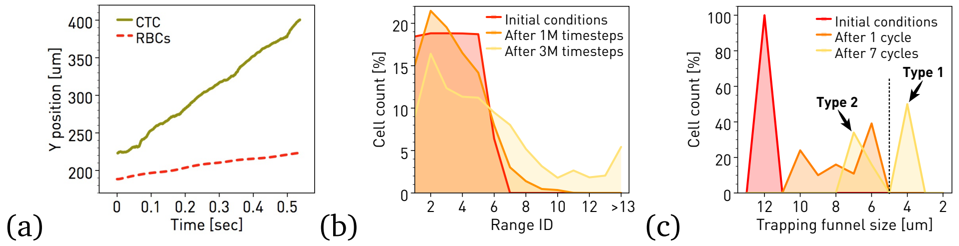

Figure 5.5. (a) Device I, displacement in the Y-direction of the CTC versus time. (b) RBCs count between columns of obstacles with Range ID. Initially, all cells placed below column with ID 7, with time some cells distribution becomes smoother. (c) Device II, the evolution of the cell distribution.

performance considering the aggregate CPU and GPU.

- *Time-to-solution:* our software outperforms the current, leading, state-of-the-art software LAMMPS[2] by a factor of up to 45X.
- *Throughput:* we simulate 1.8 Billion cells/s and 1.55×10^{13} unknowns/s, a three orders of magnitude improvement over state-of-the-art blood flow simulations [190].
- *Complexity:* Simulations considered microfluidic devices with a volume of 132 mm^3 with at submicron resolution, handling up to 1.43 Billion deforming RBCs (0.15 ml of blood), each discretized with 500 elements. To the best of our knowledge, there have never been such large-scale simulations of blood flow in realistic geometries of microfluidic channel, at sub-micron resolution, accessing continuum time scales.

These results were collected on Titan with simulations at full scale and include I/O time. The performance was measured throughout the simulation with high-resolution timers and hardware counters with *nvprof*. More precisely, the overall aggregate IPC was measured by counting the total number of executed vector instructions over hundreds of timesteps, divided by the wall-clock time. The profiling was performed in the middle of the simulation.

Table 5.2 shows the comparison of our code with LAMMPS for a plane Poiseuille flow. The test case consisted of running 10^4 time steps of DPD solvent in a periodic domain with the number density of 4. We have varied the number of nodes, ranks per node and particles per rank as well as the frequency N_R of rebuilding neighbor lists in LAMMPS. The parameter N_R depends on the problem, e.g. flow problems exhibiting high shear strain rates and large time steps require $N_R = 1$. We observed a consistent improvement of time to solution with respect

[2] LAMMPS GPU package, compiled for SP arithmetic.

Table 5.2. Outperforming factor over LAMMPS. N_R is neighbor list rebuilding frequency.

Nodes	Ranks/node	Particles/rank	Outperforming factor	
			$N_R = 4$	$N_R = 1$
1	1	3,538,944	26.7	44.1
48	1	3,538,944	26.7	**44.6**
48	8	442,368	7.40	17.2
18600	1	3,538,944	22.2	**37.9**
18600	8	442,368	4.9	11.4

Table 5.3. Achieved throughput in terms of instructions and unknowns.

Metric	Measured value
Maximum [I/s]	2.58×10^{16}
Overall [I/s]	1.34×10^{16}
Unknowns/s	1.55×10^{13}

to LAMMPS of up to 45X.

We benchmarked HOOMD-Blue[3] against the same flow problem. The HOOMD-Blue version, which we used for this study, were not capable of large-scale simulations using more than 124 Million particles. Therefore the benchmark is limited to a maximum of 280 nodes for a constant workload of at least 442,368 particles per rank. Under this limitation, we report a constant gain from 3X to 4X over HOOMD-Blue.

[3] HOOMD-Blue was compiled for SP arithmetic.

Table 5.4. Weak scaling efficiency in percent.

Ranks/node	Nodes					
	1	300	1200	4800	10800	18600
1	-	98.8	92.4	89.9	82.1	80.5
2	-	99.6	99.1	97.7	94.0	87.9
4	-	99.9	99.7	99.7	98.6	98.2
8	-	99.9	99.9	99.6	99.7	99.7
16	-	99.9	99.9	99.9	99.9	99.9

Table 5.5. Strong scaling efficiency in percent (Piz Daint).

Nodes	Speedup	Efficiency
625	-	-
1250	1.98	98.9%
2500	3.82	95.4%
5000	7.50	**93.8%**

Table 5.6. Summary of the large-scale simulations on Titan.

	Funnels ratchet	CTC-iChip1	Suspended RBCs
Simulation volume [mm^3]	33	39	94
Total cells	—	3.4×10^7	1.4×10^9
Total unknowns	5.9×10^{11}	8.6×10^{11}	6.0×10^{12}
RBC/s	—	1.7×10^8	1.8×10^9
Unknowns/s	4.1×10^{12}	4.3×10^{12}	7.5×10^{12}

Table 5.3 shows the peak performance of our code in terms of instructions and unknowns throughput. Our software reaches 65% of the nominal IPC peak performance of Titan. This leads to a rate of 13.4 PetaInstructions/s, effectively solving $15.5 \cdot 10^{12}$ [Unknowns/s].

A close-to-perfect weak scaling of our software is depicted in Table 5.4. As the number of ranks per node increases, we observe an excellent weak efficiency of more than 98% on the 18,688 nodes of Titan. Table 5.5 reports a 94% strong scaling efficiency on Piz Daint from 625 nodes to 5000 nodes.

Table 5.6 shows a summary of a series of large-scale simulations performed on Titan. The four cases considered are simulations involving solvent, RBCs and CTCs in complex geometry with volumes of up to 150 mm^3 and 1.43 Billion deforming cells, where we observe a computational throughput of 1.8 Billion cells/s. Considering the solvent as well as the cells, we observe an update rate of 4.4 Tera-unknowns per second.

5.6.1 GPU kernels

We carried out a detailed analysis of the four most demanding CUDA kernels which take approximately 85% of the execution time on GPU (Figure 5.6). The analysis is based on the event counters and metrics supported by nvprof.

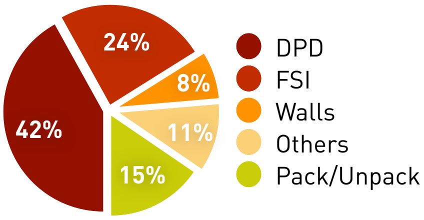

Figure 5.6. Distribution of the GPU execution time.

The DPD kernel computes the interactions and accounts for 35% of the total GPU time. For each potential interaction between pairs of particles in neghboring cells, the kernel loads eight half-precision floats via texture and performs 8 floating point instructions. For each non-zero interaction the kernel further loads sixteen single-precision floats through texture memory, issues up to 6 SP atomic additions and executes 92 floating point instructions including 3 reciprocal square roots. Given an average of 8 non-zero interactions out of 50 potential ones, this roughly adds up to 0.6 flop per byte read. The kernel reaches an executed IPC of 2.7, corresponding to 45% of the nominal GPU peak performance. The kernel performance limiters are instruction level parallelism and texture fetch latency. The execution dependency and texture fetch account for 33% and 25% of the latencies, respectively. This kernel achieves a backend utilization of 65%, among which 60% are floating point operations. The kernel exhibits an L2 cache throughput of 452 GB/s, where 40% of it consists of atomic operations. Texture cache hit rate is 75% while texture cache throughput is 250 GB/s. As a consequence, available memory bandwidth may be considered the main performance limiter.

The FSI kernel computes the interactions between cells and solvent particles and accounts for 17% of the total GPU time. The IPC for this kernel is 2.5, corresponding to 43% of the nominal GPU peak performance, whereas the achieved occupancy is close to 60%. The performance limiter here is memory and texture bandwidth, the kernel exhibits 300 [GB/s] and 200 [GB/s] of L2 cache and texture bandwidth, respectively.

The computation of the interactions between the DPD particles and the wall achieves an IPC of 3.2, corresponding to 53% of the nominal peak. The performance of this computational stage is latency-bound. Latencies are significant due to the irregularities of the wall geometry, even if the occupancy is above 84%. About 40% of stall reasons are attributed to memory dependencies due to the indirect access to the wall particles.

The kernels for packing and unpacking MPI messages are responsible for about 20% of the execution times. These kernels are divided into two groups: particle redistribution across subdomains and halo particle exchange. Kernels of both groups are capable of extracting about 110-165 GB/s of device memory bandwidth, corresponding 60-95% of the maximum measurable peak.

5.6.2 Time scales separation

Taking advantage of the separation of the temporal scales within the simulation leads to a 5.5X gain in time-to-solution. The secondary benefit of time scales separation is an increase in the GPU utilization: the IPC is observed to increase up to 40% of the nominal peak performance overall.

5.7 Summary

In this chapter, we presented a software for large-scale simulations of flows in complex geometries of microfluidic devices involving RBCs, CTCs and WBCs. This software employs the DPD method to simulate microscale flow phenomena at sub-cellular resolution and accesses time scales that so far have been only accessible by continuum solvers. In terms of time-to-solution, we outperform the current state-of-the-art DPD solvers by 38X-45X. The software demonstrates the potential to accelerate by order of magnitude the design cycle for microfluidic systems for medical diagnosis and drug design.

Despite the irregular nature of the computational patterns involved, the SIMT architecture was shown to deliver a significant performance gain for DPD simulations. We have described a set of strategies and techniques to effectively map DPD simulations on GPU-accelerated supercomputers. The unique features of the present software lead to full-scale simulations reaching peaks of up to 65%, and 34% in average, of aggregated nominal CPU-GPU peak performance. The present software redefines the frontier capabilities of simulations using multi-scale particle-based methods.

With the present software, we were able to consider flow simulations involving up to 1.43 Billion deformable RBCs (0.15ml of blood) at sub-micron resolution. Domain sizes up to tens of cubic millimeters were modeled for two microscale geometries, covering the entire functional compartments of microfluidics devices. We numerically assessed the operational efficiency of two microfluidics devices: the CTC-iChip, which isolates CTCs from hundreds of thousands of RBCs, and the funnels ratchet, which sorts non-RBC cells according to their size and deformability.

Chapter 6

Eukaryotic cell model

6.1 Introduction

Cell mechanics has proved to be a widely used label-free biomarker to discern phenotypes, detect pathologies and more importantly, monitor existence or progression of a disease [162, 120, 123]. The most prominent example is the changes in cell biology and morphology when it evolves from a healthy to a cancerous state [162, 123]. These changes take place at the molecular level affecting properties of individual components of cell internal structure, but eventually leading to alterations in mechanical properties of the whole cell.

One of the novel approaches for studying mechanical properties of cells involves development of custom-designed microfluidic devices where deformability of cells is estimated; this is usually done by measuring the time taken for a cell to pass through a tight straight channel, or its average velocity as it transits through a series of small openings, or by monitoring a cell as it squeezes under hydrodynamic forces [207, 19, 77, 68, 115]. These devices can provide higher-throughput systems than conventional technologies such as atomic force microscopy and micropipette aspiration [216] and can be used as a comparative tool between different subpopulations of cells. They, however, often lack in-depth mechanical analysis (ex. elasticity, viscosity) and have little or no regard to the differences in intrinsic properties of these cells.

To obtain a more detailed analysis of the cell mechanics with all its major underlying components, researchers have utilized modeling. Computational approaches to model cell deformation through microfluidic devices as complementary of experimental investigations are prominent for multiple reasons. Firstly, such modeling approaches give an insight into how cell components function under stress. Secondly, they can improve our understanding of the changes that

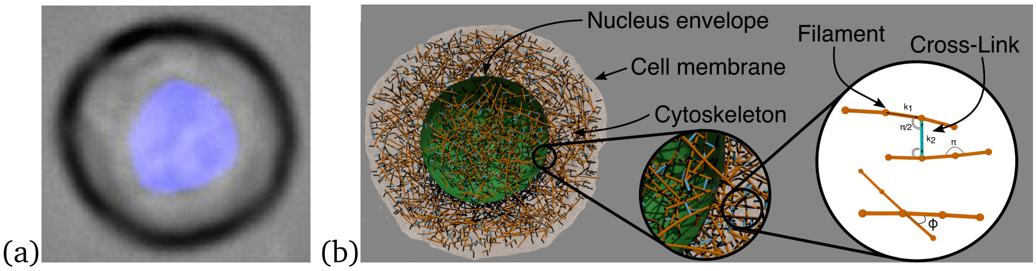

Figure 6.1. (a) Microscopy of a cell with nucleus shown in blue. (b) Three component cell model: cell membrane is shown in gray, nucleus is in green, the cytoskeleton is in orange, the connections between cytoskeleton and membranes are in black. Cytoskeleton network model is composed of long and stiff filaments (orange) connected by short cross-links (blue).

occur during disease progression which, in turn, might uncover reasons for corresponding alterations occurring in cell mechanics [114, 199]. Finally, computational models can be used as predictive tools for the experimental design.

In this chapter, we present a new model integrated into DPD framework that is suitable for modeling of eukaryotic cells with wide range of viscoelastic properties and, at the same time, computationally efficient to be employed to study large and complex flow domains. To the best of our knowledge, this is the first study that models significant flow-induced deformations of cells using mesoscale particle-based model that explicitly takes into account cell membrane, nuclues, and internal cytoskeleton. To validate our model experimentally, we have chosen normal breast epithelial cells (MCF-10A). Experiments were carried out by our collaborators, group of Chwee Teck Lim at NUS, Singapore. Using micropipette aspiration experiment, we first probed cells' elastic properties and used these data to set up parameters of the computational model. We, then, validated the model using data from microfluidic experiments, where MCF-10A cells were flown through three different microfluidic devices with a series of triangular shaped constrictions and transit velocity of these cells was measured. Using our computational model, we related transit velocity to cell elastic and viscous properties and examined the effect of cell model components such as cytoskeleton and nucleus on whole cell mechanics. We envision that the developed model will bring us closer to understanding the role of cell biomechanics in the broad spectrum of phenomena, such as the mechanical consequences of the structural changes that occur as cell evolves from healthy to diseased state.

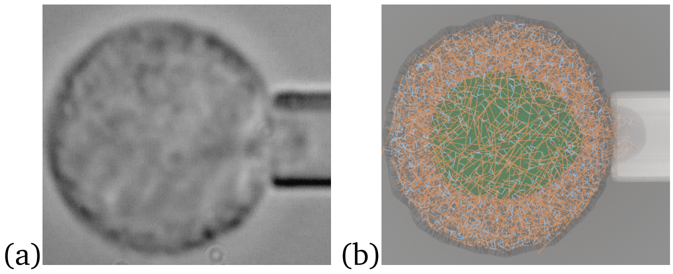

Figure 6.2. (a) Micropipette aspiration experiment. (b) Simulation snapshot of a cell during the micropipette aspiration.

6.2 Materials and methods

6.2.1 Experimental setup and material preparation

Cell line

Human mammary epithelial cell line of MCF-10A was used for the experiment (Figure 6.1 (a)). Cells were cultured with MEGM (mammary epithelial growth medium) (Lonza, Basel, Switzerland) supplemented with $0.1\mu g/ml$ cholera toxin (Sigma Aldrich, USA) and were maintained at $37°C$ in a 95% air and 5% CO_2 incubator (SANYO, Japan). The cell media was replaced with basal media without Fetal Bovine Serum (Thermo Fisher Scientific, USA), and incubated overnight prior to every experiment to synchronize all cell cycles at G_0. Cell size is varied between $7.5\mu m$ and $22\mu m$ in diameter. To describe the nucleus size, we used nuclear-cytoplasmic ratio (NC). This was estimated from microscopy as a ratio between visible nucleus area divided by cell area. We found that for MCF-10A cells, the NC ratio is 0.29 ± 0.11. Prior to each experiment, cells were harvested using standard cell passaging protocol ($330g$ for $5min$) and resuspended in 0.1% pluronic solution; they were then kept on ice and immediately used for experiment.

Micropipette aspiration experiments

We employed micropipette aspiration technique to measure the elastic properties of cells [204, 132]. Prior to each experiment cells were suspended in 10% Bovine Serum Albumin (BSA, Sigma Aldrich) to minimize cell to pipette adhesion. During each pipette aspiration measurement, a cell was placed next to a micropipette of radius R_p and increasing suction pressure was applied, driving portion of the cell into the micropippete (see Figure 6.2 (a)). The suction pres-

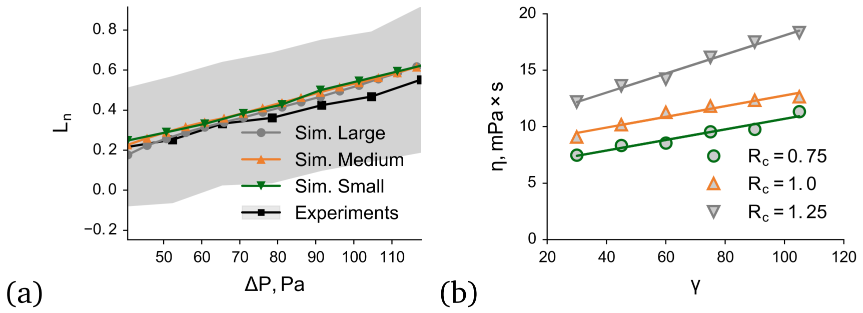

Figure 6.3. (a) Comparison between experimental data and simulation for micropipette aspiration, where L_n is normalized indentation length. (b) Cell viscosity as a function of dissipative force parameter γ and cutoff length R_c obtained from micropipette aspiration simulations.

sure was increased from 0 to $117.72Pa$ with a constant rate of $3.27Pa/s$. Cell deformation was recorded at 2 frames per second. The cell response during this experiment can be described by a normalized aspiration length $L_n = (L_p - L_p^0)/R_p$, where L_p is an aspiration length which depends on the applied pressure and elastic properties of the cell, while constant L_p^0 is aspirated length at tiny pressure. Measurements of normalized aspiration length were used to define parameters of the simulation model of the cell as will be described later in the text. These measurements can further be used to estimate elastic modulus of each cell under the assumption of the cell being a homogeneous elastic solid using Theret *et al.* model [226]. The estimated elastic modulus of MCF-10A cell was found to be 237.47 ± 66 Pa.

Microfluidic device experiments

In our experiments, we used three different microfluidic devices with varying constriction gap sizes. For each device, microfluidic master wafer was fabricated using standard soft-lithography techniques. Microfluidic chips were fabricated in polydimethylsiloxane (PDMS) (Sylgard 184 Silicone Elastomer Kit, Dow Croning, USA) by double casting process. The working part of all three devices consisted of ten rows of triangular shaped constrictions as shown in Figure 6.4. The depth of the devices was $25.8\mu m$, while the spacing between the rows of obstacles was chosen to be $60\mu m$ (more than three times the average cell diameter). To compensate for variability in cell size, we have divided cell population into three categories of small ($D = 10-14\mu m$), medium ($D = 14-18\mu m$), and large ($D = 18-22\mu m$) cells, where D stands for the cell diameter. In order to study

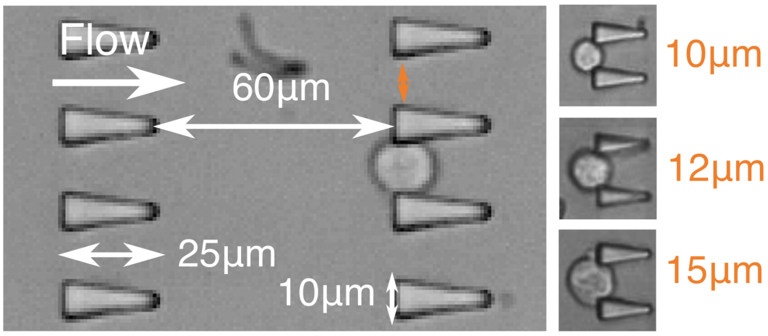

Figure 6.4. Microscopy image of section of the device.

each category, we used microfluidic devices which differ in gap size between the obstacles: device I with 10μm gap size for small size cell population, device II with 12μm gap size for medium cell population, and device III with 15μm gap size for large cell population. Prior to each experiment, microfluidic chips were washed for 10 minutes with 1% BSA to reduce cell-surface friction. Primed chips were placed under a 20x objective of an Olympus IX71 microscope for brightfield imaging. The cells were counted with BIO-RAD TC20 Automated Cell Counter and were diluted with a 0.1% pluronic solution to obtain desired cell concentration of 200$kcells/ml$ to ensure that devices are not overwhelmed with too many cells. A pressure pump (Eleveflow, France) was used to drive the cells into the constrictions and a high-speed camera (phantom V9.1) was used to record cell passage as avi with 1400 frames per second. Pressure drop was optimized to be 0.67$Pa/\mu m$ to provide ample time for cell recovery after each deformation in all devices. As cells were flown into the devices, they had to deform their way through the constrictions and differences in average velocity was determined by these cells' mechanical properties. The recorded videos were processed using a custom-written ImageJ [205] macro to eliminate the constrictions and to threshold the cells. The thresholded videos were further processed using IMARIS 8.1 tracking algorithm (Bitplane, Switzerland) to track each cell by surface rendering. Transit velocity of each cell was then calculated by dividing its traveled distance over time.

6.2.2 Cell Model

We developed a new computational model mimicking structure of a cell. The model has three components: cell membrane, nuclear envelope, and internal cytoskeleton, as shown in Figure 6.1 (b). The cell membrane model describes lipid bilayer together with underlying cortical actin layer. The nuclear membrane with lamins meshwork are defined by the nucleus envelope model. Both membrane models are based on MS-VE-RBC model, see Section 3.3. The cytoskeleton is represented by a network of cross-linked filaments mimicking the topology of F-actin network. For the sake of simplicity, the chromatin network inside the nucleus is described by the same model. The random cytoskeleton network formation and its integration with the membranes are explained in Section 6.2.2. To model fluid and fluid-structure interactions, we used the DPD, parameterization can be found in Section 6.2.2.

Although we explicitly model sub-cellular components, we follow the phenomenological approach, rather than reductionist method, to explain the properties of a system from the properties of its constituents [114]. This is due to our limited knowledge of the cell mechanics on the considered length scale. The aim of our cell model is to describe the correct mechanics of the whole cell, rather than to accurately reproduce the mechanical response of each individual molecular constituent at the microscale. Thus, although we, where it is possible, incorporate the knowledge of the microscale mechanics, we do not state that the model describing every particular constituent is correct. The important implication of this is that many model parameters can not be directly related to experimentally measured properties of individual cell constituents.

DPD parameterization

Values of conservative and dissipative force parameters used in simulations are summarized in Table 6.1. For filament-filament and filament-cell interactions, we employed different R_c for conservative force (0.5) and for thermostat (1.0). The short cut-off radius for repulsion together with high value of corresponding conservative force parameter allows preventing filaments from crossing each other, while it is desirable to have the freedom in varying R_c and γ for the thermostat to model cytoskeletons of different viscosities. Cell membranes interactions are strongly repulsive following previous works [140].

Cytoskeleton model

In order to describe the viscoelastic properties of the internal cytoskeleton, we developed a new mesoscopic model.

The cytoskeleton of a typical cell is composed of multitude of linked molecules with different elastic properties [139]. It is impractical to explicitly model them

Table 6.1. DPD parameters listed in the format of a/γ. Values shaded with yellow describe interactions with $R_c = 0.5$ for repulsive interaction while $R_c = 1$ for thermostat. For dark gray, $R_c = 0.5$.

	1. Fluid	2. Filaments	3. Membrane	4. Nucleus	5. Wall
1	10/30	10/45	4/45	4/45	10/30
2		100/65	100/65	4/65	4/65
3			100/45	100/45	10/45
4				100/45	10/45
5					excl.

for the problems on the chosen length scale. Instead, we propose a simplified model which involves filaments and cross-links of only one type, where every filament in our model represents a bunch of filaments rather than individual ones. We emphasize that our model was not built to capture the microscopic details of a real cytoskeleton, but rather it was developed to be relatively easy to parameterize and to have a modest computational complexity.

We took advantage of the recent progress in the cytoskeleton network modeling and employed ideas developed in a series of papers dedicated to F-actin network modeling by Kamm *et al.* [111, 17, 112]. In these works, semi-flexible actin filaments are represented as a series of cylindrical segments, connected by elastic hinges [12]. Cross-linking proteins (CLs) are modeled similarly. The harmonic potentials are employed to describe the extension and bending of both actin filaments and CLs. Although in this model filaments and CLs are expressed by a linear elastic elements, it was shown that they semi-qualitatively capture experimentally observed behavior characteristic for semi-flexible networks [111]. Other authors, who use similar approaches, also report that the network as a whole appears to have non-linear deformation properties [231, 136]. For the cytoskeleton model used in this work shown in Figure 6.1 (b), we adopted general structure and potentials from Kamm *et al.* model. Yet our model is different in several aspects. First, we use $1\mu m$ as unit of length while for Kamm *et al.* model, it is $70nm$, which leads to a different parameterization. Selected length scale also potentially limits the range of deformations the model can describe accurately. It may be feasible to overcome this limitation in the future by developing a multiscale model where the level of coarse-graining can be chosen arbitrary, similarly to previously developed multiscale RBC model [180]. Second, we employ much smaller unit of time because the aim of the presented model is to study relatively short-term deformation which do not exceed $10s$. Third, the aim of our cytoskeleton network is to model not only the viscoelastic effect of F-actin proteins, but also the resistance to compression which is believed to be due to different types of cytoskeletal proteins. Finally, the model was implemented within the DPD framework, which allowed us to seamlessly couple entire cell mechanics with surrounding fluid.

Forces acting on particles forming cytoskeleton are described by the harmonic law:

$$E_{bond} = \kappa_{fil}(r - r_0)^2,\ E_{angle} = \kappa_{bend}(\theta - \theta_0)^2 \tag{6.1}$$

where κ_{fil} is spring constant, r_0 is equilibrium length, κ_{bend} is bending stiffness, and equilibrium angle $\theta_0 = \pi$. The same potentials are used for CLs with corresponding parameters κ_{CL}, $\kappa_{CL/fil}$, and $\theta_0 = \pi/2$. The model parameter values are summarized in Table 6.2. We also use four body interactions to model torsion between two filaments connected by CL, which is described by bending potential (Equation 3.15).

Table 6.2. List of major parameters with their values.

	Parameter	Phys. units value	Sim. units
Cytoskeleton	Filament length	$4.0\mu m$	4.0
	Filament number density, N_{fil}	-	3.5
	Filament spring constant, κ_{fil}	$0.092 N/m$	8×10^4
	Filament bending stiffness, κ_{bend}	$4.025 \times 10^{-16} J$	350
	Cross-links number density, N_{CL}	-	0.525
	CL spring constant, κ_{CL}	$0.0092 N/m$	8×10^3
	CL-filament bending stiffness, $\kappa_{CL/Fil}$	$6.325 \times 10^{-16} J$	550
	CL-filament torsion stiffness, k_{tor}	$4.7 \times 10^{-16} J$	470
Cell membrane	Persistence length, p	$0.00141\mu m$	0.00141
	Viscosity parameter, γ^C	$1.15 \times 10^{-9} Ns/m$	30
	Area cons. constant, k_A	-	10000
	Volume cons. constant, k_V	-	15000
	Spring max length, l_{max}^{cell}	$3.0\mu m$	3
	Bending stiffness	$6.14 \times 10^{-18} J$	65
Nucl. membr.	Persistence length, p	$0.00141\mu m$	0.00141
	Viscosity parameter, γ^C	$1.15 \times 10^{-9} Ns/m$	30
	Area cons. constant, k_A	-	5000
	Volume cons. constant, k_V	-	15000
	Spring max length, l_{max}^{nucl}	$1.2\mu m$	1.2
	Bending stiffness	$2.6 \times 10^{-17} J$	250

Cell model formation

The random filament network is generated by the procedure mimicking formation of F-actin networks [111]. First, a periodic domain is filled in randomly with filaments of length $4.0\mu m$ [255]. The density of filament particles is in the range of around 3.5 particles per cubic micron. This value was chosen to provide us with relatively uniform and dense cytoskeleton structure. It also allows us to model cells with wide range of elastic moduli. Second, we simulate spontaneous polymerization and depolymerization of the filaments. During this process, particles from one end of a filament can unbind, while a new particle can be added to another end of a filament. A particle binds if the distance to the closest filament's end is less than $0.5\mu m$. A particle unbinds with the probability proportional to the difference between number of added and removed particles obtained on the previous iteration. Thus, the total length of the filaments remains approximately constant during the simulation. We run this simulation for 10^4 time steps.

At the next stage, auxiliary cross-link particles with density of 0.525 particles per cubic micron are added to the system. We employ auxiliary CL particles instead of connecting filaments directly because it significantly simplifies the control over the number of connections between filaments. The chosen concentration of cross-links to filaments ratio is within the range used in other works [110]. Formation of cross-links is simulated as follows. If the distance r between a CL particle and a filament is less than $r_0 = 0.25\mu m$, they bind. One CL particle can bind only to two different filaments. One filament particle can have only one CL and this CL cannot bind filament to itself. The CL is able to unbind in a force-dependent manner following Bell's equation [8]:

$$\kappa(F) = \begin{cases} \kappa^0 \exp(\frac{\lambda F}{k_B T}) & \text{if } r \geq r_0 \\ \kappa^0 & \text{if } r < r_0 \end{cases} \tag{6.2}$$

where κ^0 is a zero-force unbinding rate constant, λ is the mechanical sensitivity for unbinding, and F is the magnitude of the force acting on this bond. We set $\kappa^0 = 78s^{-1}$, $\lambda = 3.5 \times 10^{-5}\mu m$ in this work. We utilize this equation to simulate the stochastic nature of bond rupture [112]. This simulation is performed until the number of free CL particles does not converge to the minimum. Typically, this process takes 6×10^5 time steps.

Once the network has been generated, we post-process it to make it computationally more efficient. We first delete CL particles which are attached to only one filament, since they do not contribute to the cytoskeleton mechanics. We then delete other CL particles, merging two bonds between each CL particle and filaments into one CL bond connecting pair of filaments.

The obtained random filament network is incorporated with the cell membrane and nucleus envelope. First, we cut part of the network which corresponds to the cell volume and simulate spontaneous formation of connections between the membrane vertices and filaments. The new bond is created if the distance between filament and membrane particles is less than $0.5\mu m$; for the nucleus, the chosen distance is $0.4\mu m$. One filament particle can be connected to only one membrane particle. The unbinding of the newly created bonds happens on force-dependent manner following Equation 6.2 with parameters $\kappa^0_{cell} = 30s^{-1}$, $\lambda_{cell} = 10^{-4}\mu m$, $\kappa^0_{nucl} = 78s^{-1}$, $\lambda_{nucl} = 2 \times 10^{-4}\mu m$ for cell and nucleus correspondingly. This simulation runs until number of connected membrane vertices converges. At the next step, we add four body potential to model resistance of two connected filaments to the torsion with respect to connecting CL. The equilibrium angle for a dihedral is set to the initial value of torsion angle between connected filaments after the cell generation. The resulting cytoskeleton network topology can be characterized by the following properties. The number of CLs per filament is 2.1 ± 1.1, the number of cell membrane particles binded with cytoskeleton relative to the total number of membrane particles is 0.21, similar value for nucleus membrane is 0.26.

We do not consider cytoskeleton reorganization after the cell model has been generated, since the timescale of these processes is too large in comparison with the timescale of deformations in microfluidic devices. Namely, polymerization and depolymerization take up to $10s$ and $100s$ correspondingly [119] and CL unbinding requires at least $2.5s$ [54].

6.3 Results and Discussion

In this section, we provide the rational behind the choice of model parameters. We consider the model of medium size MCF-10A cell, which has diameter of $D = 16\mu m$ and with nuclear-cytoplasmic ratio of $NC = 0.29$. We first focus on parameters defining elastic properties of the cell, while parameters affecting cell viscous properties are considered later. We explain the choice of parameters grouped by sub-cellular components.

Cell membrane model describes both lipid bilayer and cortical meshwork. Although some parameters can be directly extracted from experimental data, such as cell and nucleus size, the experimental values of others are unknown. It is widely accepted though that the main contributor to the cell stiffness is the internal cytoskeleton and the impact of the membrane is not that significant. Thus, we model membrane as relatively soft material and we base its parameters on the

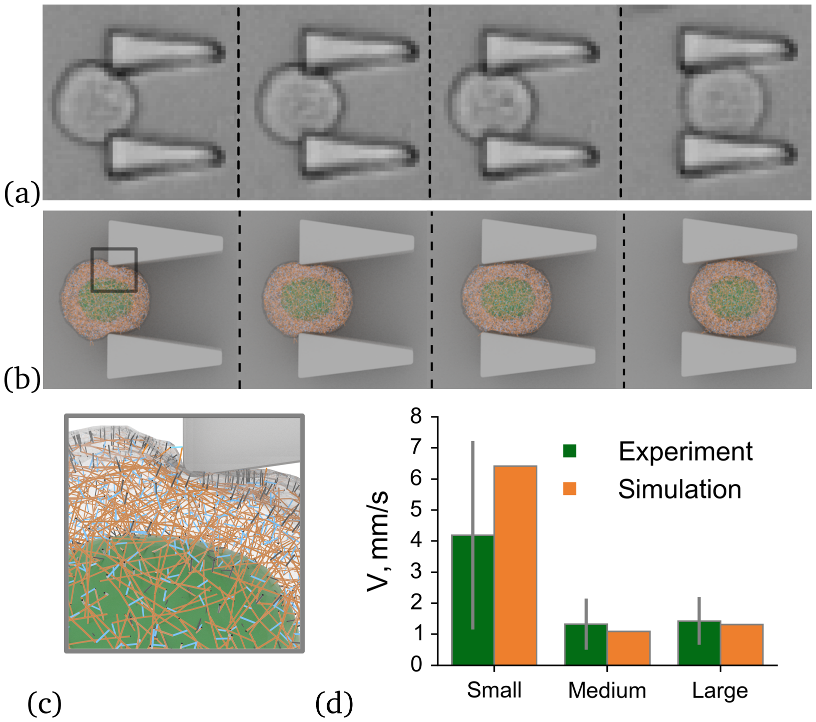

Figure 6.5. (a) Microscopy of a MCF-10A cell squeezing between obstacles. (b-c) Simulation snapshots for a MCF-10A cell model squeezing between two diverging constrictions. Fluid particles are not shown. (d) Comparison between experiments and simulations for cell velocities.

values previously used for RBC membrane modeling (persistence length, viscosity, bending stiffness). We represent cell surface with triangular mesh with 3500 vertices, so that the average bond equilibrium length is $0.518\mu m$. The stiffness of the membrane model can be controlled by the value of the maximum elongation of the WLC link l_{max}. Considering experimentally observed values of elastic modulus for different cell lines, we found that $l_{max} = 3.0\mu m$ is a reasonable choice. This rather high value allows us to model even very soft cells, increasing the range of model applications.

The cytoskeleton model has more parameters with unknown values than other components. The procedure used to generate the network was explained previously in Section 6.2.2. Parameters of the cytoskeleton model which regulate stiffness of filaments and cross-links, namely spring and angle constants (see Equation 6.1), have to be specified. Since every filament in the model represents a bunch of protein filaments of different origin, we cannot relate the model parameters describing them with the molecular-level data. Instead, we have chosen such values which result in the correct estimated elastic modulus for the whole cell model. We used relatively stiff filaments with high bending rigidity, while CLs are one order of magnitude softer which is usually the case for actin networks [17]. Variation of filament-CL torsion stiffness can be used as an additional way to control the overall stiffness of the entire cytoskeleton network.

Parameters for the nucleus envelope, which in our model describes both lipid membrane and underlying lamins network, must be defined as well. To minimize the total number of parameters in the cell model, we chose nucleus envelope membrane parameters to be equal to corresponding parameters in the cell membrane model. The only parameter which we vary to control the nucleus stiffness is l_{max}, which was set to 1.2.

To define the values of model parameters mentioned above, we performed a series of simulations of micropipette aspiration experiments (see Figure 6.2 (b)), which allowed us to estimate elastic properties of the whole cell model. Specifically, we tuned model parameters until the desired properties of the cell were obtained. We note here, that the resulting set of parameters is not uniquely defined. It is possible that some of the parameters can be chosen in a more rigorous way. However, since our goal is to match the properties of the entire cell, we are not concerned here with particular values of parameters describing individual constituents of the model. The resulting set of parameters (see Table 6.2) is verified by performing micropipette simulations with 16 independently generated cells with each cell rotated by 4 different angles. For every case, we examined the dependence between applied aspiration pressure δP and normalized aspirated length L_n. We compared mean L_n among all the simulations for each δP with the corresponding mean values obtained from the experiments and found a good agreement, as shown in Figure 6.3 (a).

The estimation of cell viscous properties usually requires a more complicated analysis than needed for elastic modulus estimation. For the current simulation study, cell viscous properties were determined using the time dependence of micropipette aspiration length at constant applied pressure. The longer it takes for a cell to reach plateau value of aspirated length, the higher is its viscosity. We perform this analysis using an extension of the Theret model proposed by Guevorkian *et al.* [73] which provides a procedure for more accurate viscosity (η) estimation than the original model. In order to control η, we use cutoff radius R_c and dissipative force coefficient γ parameters for interaction between filaments and other particles. We observe that the viscosity positively correlates with both parameters and the highest value of viscosity can be achieved by setting both R_c and γ to the highest possible values, see Figure 6.3 (b). The proposed model allows us to vary viscosity up to 3 times which might be useful for modeling cells of different types. For MCF-10A, experimentally observed values for η are in the range of $6.75 - 13.75 mPa \times s$ [27]. Thus, we have chosen $R_c = 1.0$ and $\gamma = 65$ to have $\eta = 10 mPa \times s$.

6.3.1 Validation of cell model, microfluidic device II

With the model parameters defined by measuring the whole cell response in micropipette aspiration simulations, we perform validation of the model using experimental data for medium size MCF-10A cells traversal through microfluidic device II described in Section 6.2.1.

We set up the pressure difference, driving the flow in the working part of microfluidic device in simulations, by matching the average velocity ($22.57 mm/s$) of small $3.85 \mu m$ beads which we added to the flow in experiments. We considered the same 16 cell models used in the micropipette aspiration simulations and modeled their passage through microfluidic device. In Figure 6.5 (b) we show snapshots of a typical cell squeezing between the obstacles in an experiment. Corresponding snapshots obtained in simulations are shown in Figure 6.5 (c,d). Comparison of mean cell velocity in simulations and experiments is shown in Figure 6.5 (e). The results are in a good agreement.

6.3.2 Validation of cell model, microfluidic devices I and III

In the previous section, we validated the cell model using data from microfluidic experiments with medium size cells in device II. In this section, we will consider two other devices, I and III, which were used with small and large size cell populations, respectively. The microfluidic experiments in devices I and III were performed with the same pressure difference driving the flow as in device II.

The difference between microfluidic devices I, II and III is the gap size between the obstacles which was chosen based on the average cell size (see Figure 6.5 (a)). Specifically, the ratio between the gap size and the average cell diameter in device III is the same as in device II and is equal to 0.75. Device I, on the contrary, has the ratio between the gap size and the average cell diameter equal to 0.83. The experimental results (Figure 6.5(e)) showed that the average velocity is approximately the same in devices II for medium size cells and device III for large cells. The average velocity of small size cells in device I was found to be much higher than for the medium size cells in device II.

We created models for small and large size cells using the same procedure and parameters as used for the medium size cell model. The only difference was the diameters of the cell, which for small and large cell models were $12 \mu m$

and $20\mu m$, respectively. We employed 16 independently generated cell models for each cell size. The micropipette aspiration setup was used to estimate elastic properties of all cells. Applying the same procedure as for medium cells, we found dependence of normalized aspiration length L_n on pressure δP for small and large cell groups, see Figure 6.3 (a). We do not observe impact of the cell model size on L_n and, thus, on the elastic properties of cells. Viscosity is also the same for all cell models. In Figure 6.5 (e) we show the results from microfluidics simulations performed with small and large size cell models. Similar to experiment, we observe that the average velocity of small size cells in device I is much larger than the velocity of medium and large size cells in devices II and III. In general, simulation results and experimental data are in a good agreement.

Although the interactions of cells with the obstacles in microfluidic devices are complex, some simple considerations may help explain the observed results for three devices. One of the important parameters is the effective size of the opening between the adjacent obstacles in the devices. More specifically, we can define the effective size as a radius of the circle with the same area as the area of the opening. The values which we obtain for three devices then are 9.1, 9.93 and 11.1. Another important parameter, is the ratio between the effective size of the openings and the average cell diameter. The values we obtain for three devices are 0.76, 0.62 and 0.55.

The effective size of the openings in device III is roughly 1.1 times larger than in device II. Therefore, we expect the average fluid velocity to be higher in device III comparing to device II. At the same time, the ratio between the opening size and the cell diameter is smaller in device III, and therefore the cells in device III have to squeeze through the opening which has smaller relative size, making it more difficult to pass. These two effects partially cancel each other, and the resulting average cell velocities are approximately the same in the two devices.

If we consider devices I and II, the effective size of the opening is smaller in device I comparing to device II, so we expect the average fluid velocity to be smaller. At the same time, the ratio between the opening size and cell diameter is larger in device I, making it easier for cells to pass through device I. The last effect strongly dominates the reduction in the average fluid velocity and, hence, the velocity of the small cells in device I is several times higher.

Interactions of cell with the obstacles in all three microfluidic devices make accurate modeling of mechanical properties of cell essential to obtain correct prediction of the average cell velocity. With the help of simulations, we now should be able to analyze and explain such interactions in more details.

6.3.3 Effect of the cytoskeleton

The cytoskeleton is believed to be one of the main contributors to cell stiffness. During progression of several diseases, changes in cytoskeleton structural properties may lead to significant softening of the cell. Such alternations include reduction of the filaments density as well as decrease in number of cross-links. For instance, it has been shown that to facilitate metastasis, cell undergoes a process called epithelial to mesenchymal transition where its cytoskeleton transforms from well-organized network into fragmented arrangement of filaments [79]. By altering cytoskeletal properties, the present model can accommodate for such processes. In this section we perform simulations to quantify how differences in cell internal structures, such as cytoskeleton and cross-links densities, affect cell mechanical properties.

To simulate cytoskeleton density variations, we use filaments number density parameter, N_{fil} (model parameters are listed in Table 6.2). By varying its value between 1.25 and 4, we obtain elastic modulus between $75Pa$ and $260Pa$ in micropippete aspiration simulations, demonstrating strong dependence of cell stiffness on the cytoskeletal density. This dependence appears to be similar for all cell sizes as shown in Figure 6.6 (a-c). Cytoskeletal density also affects the velocity of cells in microfluidic simulations. As expected, the average cell velocity is lower with higher cytoskeleton density for all cell sizes. However, devices II and III appear to be more sensitive, demonstrating faster decrease of cell velocity, in comparison with device I, see Figure 6.6 (a-c). Smaller ratio of effective opening size to average cell diameter in these devices results in larger relative cell deformations. This suggests that devices II and III are more suitable for studying the effect of cytoskeleton structure variation.

Cross-links density, N_{CL}, is another parameter which significantly affects the cytoskeleton properties. In our model, we can vary this parameter directly by changing the number of CLs particles during cytoskeleton network generation. We examined the impact of N_{CL} on elastic modulus for medium size cell model and found that this parameter is as significant as cytoskeleton density. N_{CL} can alter elastic modulus from $110Pa$ to almost $300Pa$, see Figure 6.6 (d). Dependence of average cell velocity on cross-link density for medium size cells in microfluidic device II obtained in simulations is also shown in the same Figure.

The simulation results allow us to predict dependence of cell velocity in microfluidic device on its elastic modulus as shown in Figure 6.6 (e) for medium size cells in device II. Two sets of results are plotted corresponding to two alternative approaches we used to vary elastic modulus of cells, i.e. by changing the cytoskeleton density or by changing the cross-links density. The agreement be-

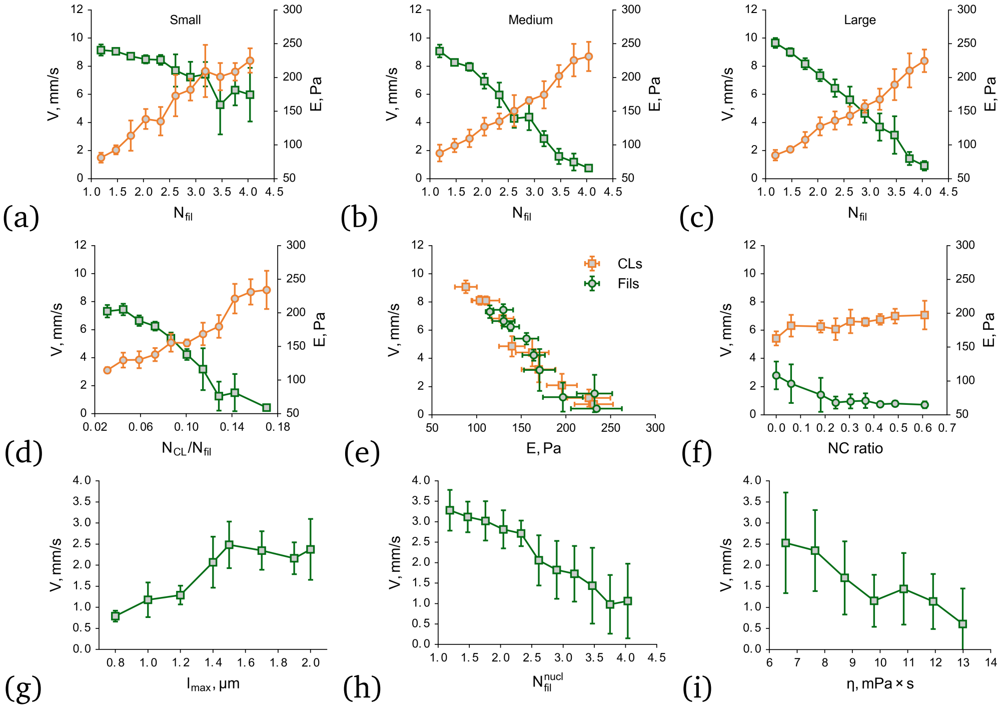

Figure 6.6. (a-c) The simulation results for the effect of the cytoskeleton filaments number density N_{fil} on the elastic modulus (orange) and velocity (green). (d) Influence of the cross-links N_{CL} to filaments N_{fil} density ratio on the elastic modulus (orange) and on velocity (green). (e) Influence of the elastic modulus on the velocity for the case when the stiffness is changed by varying filaments density N_{fil} (green) and for the case when we varied cross-links density N_{CL} (orange). (f) The relationship between nuclear-cytoplasmic (NC) ratio and cell elastic modulus, E and velocity. (g) Effect of filaments number density inside the nucleus on cell velocity. (h) The impact of nuclear laminar properties varied using parameter l_{max} in the nucleus membrane model on the cell velocity. (i) Effect of viscosity on cell velocity in the microfluidic device.

tween two sets of results provides additional support to one of the assumptions we use in our modeling approach, that not all of the structural constituents at the microscale should be resolved explicitly for the purpose of our studies, as long as whole cell properties are captured accurately.

6.3.4 Effect of nucleus

The nucleus deformability may be a critical factor in the cells' ability to pass through small openings. There are two main determinants of nuclear stiffness - nuclear lamina meshwork and the chromatin network inside the nucleus. During the mesenchymal transition, nucleus often becomes bigger and softer. It is known that its softening is primarily due to the chromatin pattern alteration which is the hallmark of malignant nuclei [102]. Altered expression of lamins in a variety of human tumors is also often associated with malignant phenotypes, whether lamins level is upregulated or downregulated depends on the cancer type [121]. Despite current advances in live cell imaging and other biophysical techniques, it is still challenging to study the effect of each component on cells mechanics. In this section, we perform a computational study of the effect of morphological and structural changes of the nucleus. We focus on the medium size cells in microfluidic device II.

First, we vary the size of the nucleus to evaluate its influence on cell elastic modulus as well as its velocity in microfluidic device. We varied NC ratio between 0.0 (no nucleus) and 0.7 for medium cells with elastic modulus of around $180Pa$, see Figure 6.6 (f). We have chosen a cell with relatively low elastic modulus because cells with very large nucleus tend to get stuck in microfluidic device. We observed that the nucleus size has a minor impact on results of micropipette aspiration simulations, indicating maximum increase of elastic modulus, E, only by 17% comparing to the cell model without nucleus. Results from microfluidics simulations, on the contrary, suggest that nucleus plays an important role in cell passage as shown in Figure 6.6 (f). In particular, in the absence of nucleus ($NC = 0.0$), we observed approximately 2.5 times increase in average velocity in comparison to cells with nucleus of normal size ($NC = 0.29$). From the modeling prospective, it means that it is essential to explicitly model nucleus for the considered type of cells.

To study the effect of chromatin concentration, we vary the filaments number density inside the nucleus (N_{fil}^{nucl}) while keeping the density of the cytoskeleton filaments constant. Our results suggest that the chromatin network has a significant impact on the cell stiffness, see Figure 6.6 (g). For example, if the network is very sparse ($N_{fil}^{nucl} = 1.25$), the velocity increases significantly, exceeding the velocity of the cell model without nucleus. We note here, that for $NC = 0.0$, the cell interior is completely filled by cytoskeleton with filament density N_{fil}.

Next, we study the effect of the nuclear lamina on cell traversal in microfluidic device. The density of the nuclear lamina meshwork is modeled in the present study by parameter l_{max}^{nucl} in the nucleus membrane model. By varying l_{max}^{nucl}, we observe that by reducing stiffness of the envelope, we can again increase the average cell velocity significantly, see Figure 6.6 (h).

Our simulation results indicate that the impact of the nucleus on cell traversal through microfluidic device cannot be explained primarily by nucleus size, nuclear lamina or chromatin networks contributions, but rather all components may significantly alter cell dynamics.

6.3.5 Effect of the cell viscosity

Cell viscosity is yet another property that may affect cell passage through microfluidic device. In general, dependence of cell velocity in microfluidic device on its viscosity can be non-trivial. In previous studies with healthy and malaria infected RBCs in microfluidic device of similar design [19], the device was found to be sensitive mostly to elastic properties of cells. Due to the specific interplay between the time needed for a cell to travel from one row of obstacles to the next and RBC characteristic relaxation time, the average cell velocity was almost independent of its viscosity. The distance between rows of obstacles and the driving pressure gradient were set in experiments so that RBCs did no have enough time to completely recover their shape during passage from one row of obstacles to the next. Cells with higher viscosity required longer time to deform and squeeze between pair of obstacles. However, these cells also required longer time to recover their shape, and therefore approached the next pair of obstacles with the shape making passage through the opening easier.

In the devices used in the present study, cells recover their shape almost completely. Therefore, we do not expect similar effects to take place. Indeed, the results of simulations with medium size cells in microfluidic device II show roughly linear decrease of average cell velocity with increasing viscosity, as one would expect (Figure 6.6 (i)). The cell viscosity was varied in simulations by changing cutoff radius R_c and dissipative force coefficient γ parameters for interaction between filaments and other particles. The obtained dependence shows that viscous properties of the cell can have comparable effect on its traversal to cell elastic properties.

6.4 Summary

We developed a new eukaryotic cell model which takes into account cell membrane, cytoskeleton and nucleus. The non-tumorigenic breast epithelial cells (MCF-10A) were used in our studies. To estimate the viscoelastic properties of cells and to calibrate our computational model, we performed micropipette aspiration experiments. The model was then validated using data from three microfluidic experiments with devices designed to take into account size variation in MCF-10A cell population.

Using the validated model, we probed contributions of sub-cellular components to whole cell mechanics in micropipette aspiration and microfluidics experiments. We obtained that the main contributor to cell stiffness is its cytoskeleton. This finding is in agreement with previous experimental studies [62, 232, 72]. Our model showed that both filament and cross-links concentrations play equally important role in defining whole cell mechanics, dominating over the effects due to variation of cell nucleus properties. Simulation results indicate that it is important to model nucleus explicitly in microfluidics simulations. Each of considered nucleus properties, namely nucleus size, stiffness of nuclear lamina and chromatin network, can significantly affect deformability of the cell. The viscous properties of the cell can have comparable effect to cell elastic properties on its traversal through microfluidic device.

We believe that the new model will allows studying in silico numerous problems in the context of cell biomechanics in flows in complex domains, such as capillary networks and microfluidic devices. Our ongoing work indicates that the proposed cell model parameterization has the flexibility to be used in simulations of various cell types, including cancer cells with different mechanical properties. With further development, the present model with explicit description of subcellular components may be used to study alterations in cell mechanics caused by diseases or functional changes.

Chapter 7

Conclusion

The work presented in this thesis introduces new methods, models, and algorithms for modeling of the individual cell mechanics and rheology of cell suspension in microflows. In Chapter 2, we began with the biological introduction into cells mechanics and blood flow rheology. We, then, gave a review of modern cell models along with the summary of their applications. In the same chapter, we overviewed the boundary conditions and state-of-the-art high-performance simulations of the suspended cells flow and underlined the existing gaps in methods and models which we will fill in the following chapters. In Chapter 3, we briefly described DPD method and RBC model which are used further in the text. In Chapter 4 we presented and validated new open boundary conditions. It was shown that our method can be applied to study the flow of detailed RBC membranes in complex geometries. We provided new algorithms and techniques along with the corresponding software which demonstrated unprecedented performance on the biggest supercomputers available at 2015 in Chapter 5. We showed that despite the computational complexity of the DPD and accurate RBC model, they can be efficiently applied to study phenomena involving billions of cells. This development opens the route for DPD method to be used to study phenomena which before could only be investigated with continuum-based methods. It was demonstrated how the presented software can be used to investigate the suspended cells flow in complex microfluidic devices. Finally, in Chapter 6 we proposed a new eukaryotic cell model which explicitly describes sub-cellular components. It was parameterized and validated using quantitative experimental data from micropipette aspiration and three microfluidic devices. This model was applied to study the effect of sub-cellular elements on the cell stiffness and viscosity. The proposed model can be employed in simulations of cells with wide range of viscoelastic properties.

Combining the results, we developed a general framework which can be effectively applied to study suspended cells hydrodynamics and mechanics in complex flow domains. We believe that the presented developments will expand the boundaries of relevant methods and models applicability.

7.1 Future directions

We suggest some areas of future research following the developments presented in this thesis:

- The proposed methods could contribute in quantification of the impact of the microvascular architecture on the oxygen delivery. It is well-known that the cancer-induced microvasculature is very different from the normal one, which leads to the tissue necrosis [22]. It would be interesting to find patterns in vessels geometry and, then, perform simulations to study the contribution of these characteristic features of microvascular network architecture on the oxygen delivery.

- RBC membrane model can be extended to study electrophoresis and dielectrophoresis. Due to the sialic acid groups embedded into the membrane, RBC is polarizable [128]. This extension could find applications in the modeling of the corresponding phenomena in microfluidic devices [35] and to study the effect of RBC charge on the aggregation [94].

- Our framework can be applied to perform predictive simulations to evaluate microfluidic devices performance. Currently, there are numerous devices which are composed of obstacles of different shapes placed in a regular manner. It might be useful to quantify the impact of the obstacles shape, spacing and pressure drop on the performance of these devices in capturing cells from different lines.

- The eukaryotic cell model alone can be applied to study a wide range of mechanical problems, including the difference in the estimated elasticity and viscosity provided by different experimental setups.

- The development of a more complicated, high resolution, eukaryotic cell model might be an interesting study direction. Currently, all the eukaryotic cell models are phenomenological, which means that the parameterization and the model composition have little connections with the properties of the cytoskeleton network components on the atomistic level. In this regards, we are very interested in the latest bottom-up developments in F-actin networks modeling [12], which connected the mechanical properties of the actin filaments on the atomistic level with the properties of the Coarse-Grained protein networks. Yet, currently, there is a lack of similar reductionist models for other cytoskeleton components, namely microtubules, intermediate filaments, chromatin, and lamins, which might be itself an interesting research direction. As soon as these models are developed, we could build a eukaryotic cell model on the nanometers resolution. Such a model would have a great fundamental impact and, additionally, push forward underlying computational methods.

Appendix A

Polarizable coarse-grained protein model for DPD

A.1 Introduction

Computer simulations on an atomistic scale are suitable for studying biomolecular systems and can be used complementary to experiments for the exploration of free energy landscapes [260, 208] and binding free energies [31]. However, at present all-atom approaches are limited in their timescale and can be computationally expensive. Coarse-graining (CG) is a method capable to reach long time-scales by solving the time-scale problem through a reduction of the degrees of freedom in the system [163, 194, 185, 184]. The CG-method has been used extensively in the popular MARTINI approach [144]. Systematic approaches for coarse-graining have been developed, including the iterative Boltzmann inversion approach [237], inverse Monte Carlo [158] and the force-matching method [98, 82]. These methods rely on a construction of coarse-grained potentials from all-atomistic trajectories.

Coarse-graining of proteins ranges over different length and time-scales and several methods have been used for the description of these biomolecular systems, including Gō-models in protein folding [218], where the native structure is biased using restraint potentials, and other models [227, 242, 155]. Very recently, polarizability effects have been incorporated into coarse-grained force-fields, especially for the coarse-grained representation of water [253, 143, 244, 192, 146]. In addition, charge-charge interactions have been added for coarse-grained protein simulations to optimize the stability of secondary and tertiary structure formation [225, 211]. Atomistic and coarse-grained force fields have also been connected in multi-scale simulations [200, 210, 163, 152, 193].

Dissipative Particle Dynamics (DPD) is a method which employs soft repulsive potentials and which enables sampling of particle based systems with large timesteps, which cannot be used in conventional molecular dynamics [84, 71, 55]. In combination with suitable parameters, this method enhances sampling many orders of magnitude faster than other simulation techniques. In this paper, we present a novel polarizable coarse grained forcefield for the description of proteins on the basis of DPD method. We mention that protein conformations have been previously described using Dissipative Particle Dynamics in references [234, 74], but not at the level of detail as presented here. Our model uses a polarizable description for water [172], as well as a polarizable backbone of the protein. The polarizability in the system is described by Drude oscillators, represented by 2 opposite charges connected by an harmonic potential, while the electrostatic interactions are calculated using a modified particle-particle particle mesh ewald (P3ME) formalism for DPD [70, 201]. We define the parameters of the new forcefield using the experimental structures of TrpZip2 and TrpCage [26, 160], and validate it on 5 different proteins, where the timescales of folding exceed the present capabilities of the all-atom simulation method [118]: GB1 [241], the WW-domain [142], the B-domain of Protein A [69], the Peripheral binding subunit [101] and Villin headpiece [148]. The new model further extends the scope of applications of Dissipative Particle Dynamics to simulations of biomolecular systems. We emphasize that our model is free from elastic network restraints needed for instance in the MARTINI coarse-grained model to keep the secondary structure stable, which we consider as a strong advantage of the presented here forcefield.

A.2 Methods

A.2.1 Dissipative Particle Dynamics

For the simulations presented here, we use the Dissipative Particle Dynamics method which was described in Section 3.2.

Additionally, we consider two harmonic bonded forces: the force between bonded particles, $\mathbf{F}_{ij}^{B}$, and the harmonic angular force, $\mathbf{F}_{ij}^{A}$, given by

$$\mathbf{F}_{ij}^{B} = dU^{B}/dr_{ij} \text{ ,with } U^{B} = \frac{1}{2}k_{D}(r_{ij} - d_{0})^{2} \text{ ,} \tag{A.1}$$

and

$$\mathbf{F}_{ij}^{A} = dU^{A}/d\theta_{ijk} \text{ ,with } U^{A} = \frac{1}{2}k_{\theta}(\theta_{ijk} - \theta_{0})^{2} \text{ ,} \tag{A.2}$$

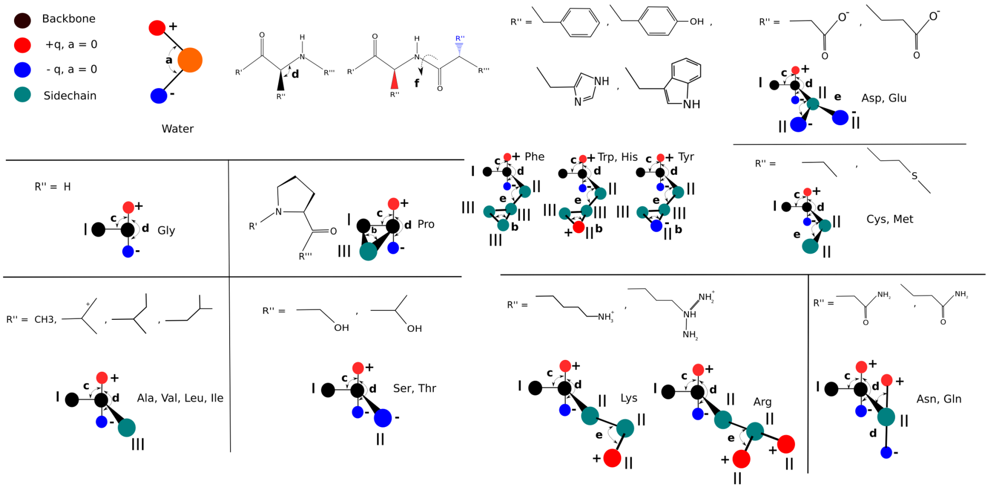

Figure A.1. Coarse-grained model of polarizable amino-acids and water. The peptide backbone is represented by 2 hydrophilic I type beads (black) and a dipole (negative charges - blue, positive charges - red), while the sidechain is attached perpendicular to the 2 backbone atoms. The polarity of sidechains is modeled by point charges, while the aliphatic groups are represented by hydrophobic beads (cyan). We use a polarizable water model consisting of one central bead and 2 Drude particles carrying opposite charges. Each of the different aminoacids has been distributed into different classes of aminoacid models with different representations of the sidechain - one bead, aromatic sidechains and hydrophilic sidechains.

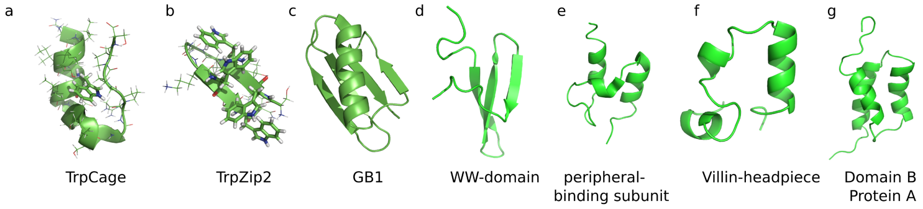

Figure A.2. Experimental structures of TrpCage (a), TrpZip2 (b), GB1 (c) , the WW-domain (d), the Peripheral binding subunit (e), Villin headpiece (f) and the B-domain of Protein A (g) simulated in the present work. TrpCage and TrpZip2 were used for the calibration, so that the coarse-grained model could reliably describe formation of α-helical and β-stranded peptides. GB1, the WW-domain, the peripheral binding subunit, Villin headpiece and the B-domain of Protein A were used for the validation of our new model. TrpCage (Sequence: NLYIQWLKDGGPSSGRPPPS [160]) PDB : 1L2Y. TrpZip2 (Sequence: SWTWENGKWTWKX [26]) PDB : 1LE1. GB1 (Sequence: MTYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGVDGEWTYDAATKTFTVTE) PDB : 2J52, [241]. (d) WW-domain (sequence : GATAVSEWTEYKTADGKTYYYNNRTLESTWEKPQELK), PDB : 1E0L [142]. (e) Peripheral binding subunit (sequence : VIAMPSVRKYAREKGVDIRLVQGTGKNGRVLKEDIDAFLAGGA), PDB : 2PDD [101]. (f) Villin headpiece (sequence : MLSDEDFKAVFGMTRSAFANLPLWKQQNLKKEKGLF), PDB : 1VII[148]. (g) B domain, Protein A (sequence : TADNKFNKEQQNAFYEILHLPNLNEEQRNGFIQSLKDDPSQSANLLAEAKKLNDAQAPKA), PDB : 1BDC[69]. For the NMR structures, we chose Model # 1 as reference structure.

where d_0 is the equilibrium bond length, θ_{ijk} stands for the instantaneous angle between 3 particles, while θ_0 is the equilibrium angle. The same potential is applied for dihedrals in the system, while each dihedral angle θ_{ijkl} is determined between 4 beads in the system. The description of electrostatics is implemented in the form of a non-bonded Coulombic force

$$\mathbf{F}_{ij}^{Coul} = dU^{Coul}/dr_{ij} \,. \tag{A.3}$$

A.2.2 Particle-Mesh Ewald electrostatics

The electrostatic interactions are described by the Particle Particle-Particle Mesh Ewald (P3ME) method in combination with a Slater type smearing out of charges [201, 67, 70]. The electrostatic potential between N point charges is described by

$$U(\mathbf{r}^N)^{Coul} = \frac{1}{4\pi\epsilon_0\epsilon_r}\sum_i\sum_{j>i}\frac{\rho_i\rho_j}{r_{ij}} , \tag{A.4}$$

where ρ_i and ρ_j are the charge density values of a pair of DPD particles, r_{ij} is the distance between the charges, ϵ_0 and ϵ_r are the dielectric constants of vacuum and water ($\epsilon_r = 78$) at room temperature. In this P3ME treatment, the long-range electrostatic energy given by equation A.4 is decomposed into a real space and a reciprocal space [42, 58]. This expression is then written as

$$U(\mathbf{r}^N)^{Coul} = \frac{1}{4\pi\epsilon_0\epsilon_r}\Bigg(\sum_i\sum_{j>i}\rho_i\rho_j\frac{\mathrm{erfc}(\alpha r_{ij})}{r_{ij}}+ \\ \frac{2\pi}{V}\sum_{\mathbf{k}\neq 0}^{\infty}Q(k)S(\mathbf{k})S(-\mathbf{k})-\frac{\alpha}{\sqrt{\pi}}\sum_i^N\rho_i^2\Bigg) , \tag{A.5}$$

with

$$Q(k) = \frac{e^{-k^2/4\alpha^2}}{k^2} , \tag{A.6}$$

$$S(\mathbf{k}) = \sum_{i=1}^{N} q_i e^{i\mathbf{k}\mathbf{r}_i} , \tag{A.7}$$

$$\mathbf{k} = \frac{2\pi}{L}(m_x, m_y, m_z) , \tag{A.8}$$

$$k = |\mathbf{k}| \tag{A.9}$$

where α is the parameter that controls the contribution in real space, k is the magnitude of the reciprocal vector $\mathbf{k}$, while m_x, m_y, m_z are integer numbers. Since the soft conservative force would allow a full overlap of particles at $r_{ij} = 0$ and infinite ion-pair formation, we use the approach of Melchor *et al.*, and apply the charge distribution $\rho(|\mathbf{r}|)$ over a 3 dimensional mesh [201, 67, 202]

$$\rho(|\mathbf{r}|) = \frac{q}{\pi\lambda^3}e^{-2|\mathbf{r}|/\lambda} , \tag{A.10}$$

with $\lambda = 0.7R_c$ and $|\mathbf{r}|$ standing for the distance between the particle center and each grid point. Thus, the reduced interaction potential u_{ij} between two charge distributions in DPD, separated by a distance r_{ij} from center to center is given by [67]

$$\frac{4\pi u_{ij}(r_{ij})}{\Gamma} = \frac{\rho_i\rho_j}{r_{ij}}[1-(1+\frac{R_c}{\lambda}r_{ij}e^{-2R_c/\lambda r_{ij}}] , \tag{A.11}$$

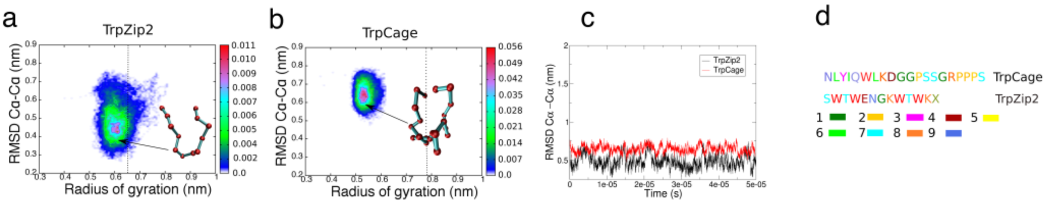

Figure A.3. (a) Probability plot of coarse-grained simulation of TrpZip2 as function of $RMSD_{C\alpha-C\alpha}$ to the native structure (given by forward mapped PDB: 1LE1) and the radius of gyration, Rg. (b) Probability plot of coarse grained simulation of TrpCage as function of $RMSD_{C\alpha-C\alpha}$ to the native structure (given by forward mapped PDB: 1L2Y) and the radius of gyration, Rg. The sidechains have been omitted in this representation and only the backbone atoms are shown for clarity. We do note that the RMSD to the backbone of the native structure is affected by an error of $\pm 0.4\ nm$ due to coarse-graining approach as described in the text. The experimental radii of gyration are indicated by a dashed line. (c) $RMSD_{C\alpha-C\alpha}$ to the native structure as function of simulation time. (d) Color-assigned sequences of TrpCage and TrpZip2. The colors indicate the assignment of each aminoacid to the different groups in the coarse-graining approach (See Figure A.1).

with

$$\Gamma = \frac{e^2}{k_B T \epsilon_0 \epsilon_r R_c} \,. \tag{A.12}$$

Table A.1. Parameters of the Drude oscillator DPD water model, i.e. angular constant k_θ, equilibrium angle θ_0, bond force constant k_D, charge of Drude particles $|q|$ and equilibrium bond length d_0. Detailed description of the water model can be found in reference [172]

k_θ	θ_0	k_D	$\|q\|$	d_0
$7.5\ k_B T/rad^2$	0^o	$1 * 10^5\ k_B T$	0.75 e	$0.2\ R_c$

A.2.3 Polarizable water for DPD simulations

In our simulations we employ the recently developed DPD model of polarizable water. For detailed description we refer to reference [172] (see also Figure A.1). In this model, a central uncharged bead is connected to 2 oppositely charged

Drude particles. A harmonic angle potential between the 2 Drude particles is applied, so that there is a match with the dielectric constant of water. Only the central particle interacts with the rest of the system through a conservative force, while the Drude particles have a conservative force parameter of $a_{ij} = 0$. Through this modeling approach, the effective dielectric behavior of water is considered and polarization effects are included, which play a role in the dynamics of proteins. We use the numerical density of central water particles $N_p = 3$, a dissipative force coefficient $\gamma = 4$, $\sigma^2 = 2\gamma k_B T$ as random force coefficient, and a DPD temperature $k_B T = 1$. For the water model [172], we modified the point charges and increased the harmonic angular constant using a $\epsilon_r = 78$ as electrostatic screening parameter (see Table A.1). Each DPD particle representing water in our simulations corresponds to $N_m = 3$ water molecules, defining the unit of length as $R_c = 0.646\ nm$ [71, 106]. We chose a conservative force coefficient of $a_{ij} = 35$ for interactions between central water particles to match the dimensionless compressibility of water. The time scale is defined by matching the self diffusion constant of water [71] and equal to $\tau = 25 \times 10^{-9}\ s$ [70, 71]. Thus, one timestep of $dt = 0.001$ in the DPD framework used in our simulations, corresponds to $25 \times 10^{-12}\ s$.

We use these parameters for water model in all simulations and analyses presented in this work.

Table A.2. Conservative force parameter a_{ij} used in the simulations. See Figure A.1 for the model chosen in the coarse-grained approach.

type	Drude	type I	type II	type III	water
Drude	0.0	0.0	0.0	0.0	0.0
type I	0.0	5.0	5.0	5.0	15.0
type II	0.0	5.0	10.0	10.0	20.0
type III	0.0	5.0	10.0	5.0	30.0
water	0.0	15.0	20.0	30.0	35.0

A.2.4 Polarizable protein model

Protein backbone. In the coarse-graining approach of the polypeptide, we take into account that peptide dynamics, secondary and tertiary structure formation are governed by the motion of the protein backbone, i.e. the formation and breaking of contacts along the chain [182]. The different accessibility of the peptide backbone to water and its mobility mainly determine the tendency of the protein to fold into defined tertiary structures [196, 16]. In the parametrization

Table A.3. Equilibrium angles θ_0 for the internal coordinates of the sidechains used in the coarse-grained model. Dihedral angle potentials (type f) with an angle of 180^o are applied between the backbone dipole and the sidechain of the nearest neighbor along the polypeptide chain. See Figure A.1 for the model chosen in the coarse-grained approach.

type	a	b	c	d	e	f
angle θ_0	0^o	60^o	71^o	90^o	120^o	180^o
$k_\theta(k_BT/(rad^2))$	1.0	1.0	1.0	1.0	1.0	1.0

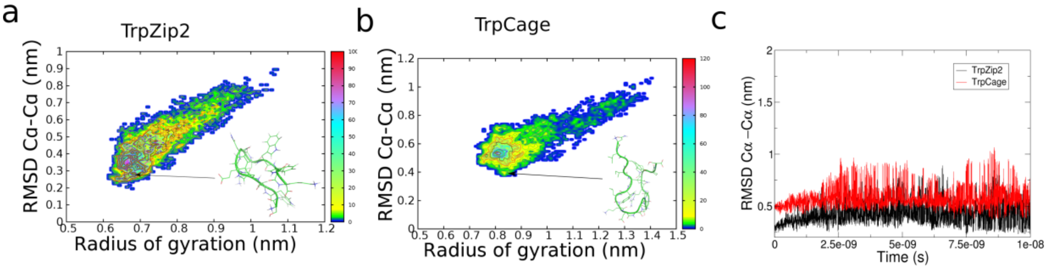

Figure A.4. Results from 10 ns all-atom replica exchange MD simulation with backmapped structures obtained from DPD coarse-grained simulations as starting structures. (a) Probability plot of all-atom simulation of TrpZip2 as function of $RMSD_{C\alpha-C\alpha}$ to the native structure (given by PDB: 1LE1) and the radius of gyration, Rg. (b) Probability plot of all-simulation of TrpCage as function of $RMSD_{C\alpha-C\alpha}$ to the native structure (given by PDB: 1L2Y) and the radius of gyration, Rg. (c) $RMSD_{C\alpha-C\alpha}$ to the native structure as function of simulation time.

procedure of the protein backbone it is our goal to display with coarse-grained model the formation and the unfolding of secondary structure elements, i.e. β-strands and α-helical elements. In contrast to different systematic approaches of coarse-graining [91, 237, 82], we aimed on a correct representation of probability distributions, i.e. free energy surfaces as close as possible to the native state (while the unfolded ensemble is partially neglected). In order to display the equilibrium between folding and unfolding of secondary structure elements, we model the attractive part of the peptide backbone as an electrostatic dipole, which is oriented perpendicular to the peptide sidechain, mimicking the dynamics of hydrogen bonding along the backbone and the solvation dynamics by polarizable water [211, 225]. As in the case of polarizable water model, we use a Drude oscillator for the dipolar contribution of the protein backbone. Each of the Drude particles carries a charge of $\pm 0.25e$. It is bonded to the central particle by a harmonic bond with a force constant of $1 * 10^5$ $k_B T$ and equilibrium length of 0.1 R_c. We applied an harmonic angle constant of 1.0 $k_B T/(rad^2)$ and equilibrium angle of 90 degrees for the angle between the 2 Drude particles and the central backbone particle. In addition, we applied harmonic angles with constant 1.0 $k_B T/(rad^2)$ for the internal coordinates of each sidechain as shown in Table A.3. All beads representing the protein sidechains and the backbone are connected by a harmonic bond potential with an equilibrium length of 0.3 R_c and force constant of $1 * 10^5$ $k_B T$. Finally, the N-terminus is charged with $+1.0e$, while the C-terminus carries the opposite charge.

We used a knowledge-based approach for the parametrization of the protein model [93, 60, 7, 157, 227]. Specifically, we started by considering a general topology for each aminoacid similar to other approaches [82, 60, 93], while we additionally included a polarizable backbone and charges for specific aminoacids as shown in Figure A.1. We then added internal coordinates in the form of angular and dihedral potentials as implemented in common all-atom forcefields [113]. Finally, we adjusted the non-bonded conservative force parameters for the interactions within the protein and the interactions with water, as well as the polarizability of the protein backbone through scaling the 2 opposite charges, until we reached a converged behavior in folding of an α-helical and a β-stranded peptide. Specifically, we used TrpZip2 [26] and TrpCage [160] as model systems, to test the ability of our approach to fold these proteins into their native structure.

For the sidechains of each aminoacid, we mainly distinguished between hydrophilic and hydrophobic aminoacids. We then further made a differentiation between aromatic aminoacids and aliphatic aminoacids. In the following, we describe the approaches for modeling the different types of aminoacids.

One bead sidechain. As aminoacids with one single bead representing the sidechain, we selected hydrophobic aminoacids alanine, valine, leucine and isoleucine, and modeled each of them with one single hydrophobic bead of type III (see Figure A.1 and Table A.2). For hydrophilic aminoacids with one single bead in the sidechain, we selected serin and threonin, and represented each of them with a bead of type II and one negative charge in the center of this bead. Glycine is the only aminoacid with no sidechain, while proline has an additional bond between the sidechain bead with a particle type III and the backbone bead, so that the cyclic ring of this aminoacid is resembled. The central Cα bead to which the sidechain bead is connected carries both Drude particles and is connected to a second bead in the backbone which represents the amide nitrogen.

Two bead sidechain. For the aminoacids cysteine and methionine, we modelled the sidechain with 2 beads of hydrophobic type II, which should resemble the sulfur with a larger vdW radius than in the case of serine and threonine with oxygen in its sidechain.

Aromatic aminoacids. For the aminoacids with an aromatic sidechain, we described the aromatic ring as a ring consisting of 3 beads and one additional bead which represents $C\beta$ connected to $C\alpha$. In the case of phenylalanine, all particles in the ring are of bead type III, while we included an additional bead of type II and a positive charge of $q = +0.4e$ for tryptophan and histidine with one additional nitrogen in the ring structure (see Figure A.1 and Table A.2). In contrast to the latter 2 aminoacids, we modeled the hydroxy-group of tryrosine as an additional bead of type II carrying a negative charge of $q = -0.4e$.

Hydrophilic aminoacids. For the hydrophilic aminoacids, we chose a more detailed differentiation between the different types of aminoacids, since polarization effects play an essential role in the system due to the electrostatics and the polarizable water model. For lysine, we modeled a sidechain with 3 beads, while the last bead carries a full positive charge of $1e$. For arginine, we modeled the guanidine sidechain by a branched sidechain consisting of 3 beads, with 2 beads representing the amines with a positive charge of $q = +0.4e$. For aspartic and glutamic acid, the sidechain is shorter in its length by 1 bead and the beads in both endings carry a negative charge of $q = -0.4e$. Finally, for the aminoacids with an amide group in their sidechain, i.e. asparagine and glutamine, we modelled the sidechain by 3 beads, with 2 beads of opposite charge of $\pm 0.25e$. These 2 charged beads have DPD conservative force parameter equal to 0.

Levels of hydrophobicity and hydrophilicity. In our knowledge-based coarse-grained model, we made an estimate of the relative level of hydrophobicity and hydrophilicity based on the average effective vdW-radius of each amino-acid sidechain. Through this procedure, we made empirical estimates on each conservative force parameter in our model. A conservative force parameter value of 5 corresponds to a high level of attraction between beads of the same type, while 10 represents moderate repulsion. The overall reference value is defined by the interaction with water, where parameter of $a_{ij} = 15$ corresponds to hydrophilic interactions with water, while $a_{ij} = 30$ represents hydrophobic amino-acid properties. In this way, aminoacids with aromatic sidechains (hydrophobic) have a larger effective radius than hydrophilic residues (hydrophilic). For the aliphatic $C\beta$ atoms, we selected a non-bonded neutral type, to resemble partial hydrophobic properties of this part of the sidechain. A similar type of classification of particle types has been made in the prominent MARTINI approach [144].

A.2.5 Implementation, Program and System Preparation

All simulations have been done using the modified version of the LAMMPS simulation package [183]. For the analysis of the trajectories and the replica exchange MD simulations, we used modified parts of the GROMACS-4.5 simulation suite and in-house programs [80]. The interactions in the REMD simulations on an atomistic scale were described with the GROMOS 53a5 forcefield. The starting configurations and the protein topology were produced using in-house code. We employed the forward mapping procedure, implemented and distributed by Tielemann and coworkers [238], to map the experimental structures of proteins to our coarse-grained representation, which we used for the comparison of our simulations with the experiment (root mean square deviation from the native structure, $RMSD_{C\alpha-C\alpha}$). In this mapping procedure, the center of mass of the backbone and the sidechain of each individual amino-acid is mapped on the center of mass of the coarse-grained structure by geometrical assignments. The same process was applied for the reverse mapping. We mapped each experimental structure into the coarse-grained representation and then calculated the RMSD (in the coarse-grained trajectory) to the central atoms of the backbone of the forward mapped structure, which represent the $C\alpha$ carbon atoms. Then we mapped this quantity forward to real parameters. In the same way, we transformed the radius of gyration. For the calculation of the free energies in our protein folding simulations, we used the relation $F = -k_B T \ln P$.

A.2.6 Estimation of absolute error

The error in the comparison of our simulation data from the coarse-grained trajectory with the forward mapped structure (on the coarse-grained scale) from the experiment is associated with following factors:

- The centers of mass of each amino-acid in our coarse-grained model deviates slightly from the center of mass of the corresponding amino-acid on the all-atom level. This error is minimal for residues with a small sidechain, such as Glycine, but comparably large for Lysine or for amino-acids with larger sidechains, such as Trptophan. We tried to minimize this error through our parameterization and model development. For each aminoacid the error is on the order of $\approx 0.2\ nm$ for the backbone atoms (Cα), which were used for the analysis of our trajectories. For the sidechains in the system the deviations are larger on the order of $\approx 0.4\ nm$.

- In our modeling approach, the bond distances between the backbone beads

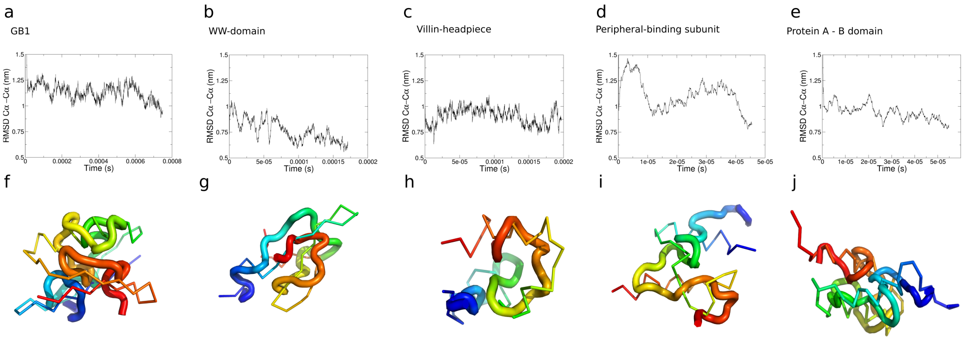

Figure A.5. Results from the folding simulations on GB1, the WW-domain, Villin-headpiece, the peripheral binding subunit and the B-domain of ProteinA. (a-e) $RMSD_{C\alpha-C\alpha}$ as function of simulation time. (f-j) Comparison of least square fitted backward-mapped structures from the simulations (cartoon) to the experimental structures (ribbon) of all 5 proteins.

and along the sidechains are equal to 0.3 R_c, while this is not the case for the inter-particle distance of $C\alpha$ for the experimental structure which we used as reference structure, where the distances are of varying length. We estimate the error connected to the variance of the reference native structure in the case of the RMSD calculation with an associated value of $\approx 0.2\ nm$.

Finally we estimate a total error in the calculation of the $RMSD_{C\alpha-C\alpha}$ and Rg to be about 0.4 nm, where we only take the backbone of each protein into account. Thus, each quantity X_c is mapped backwards to real units depending on the length scale of the system according to

$$X(r) = X_c * R_c \ \pm 0.4\ [nm]\ . \tag{A.13}$$

After the model parameters were defined, we additionally simulated the experimental structures of TrpCage and TrpZip2 mapped forward from folded state to our coarse grained description. We found that these 2 proteins remained at $RMSD_{C\alpha-C\alpha}$ 0.5−0.7 nm from the experimental structure, in agreement with the error analysis presented above.

A.2.7 Simulation details

For the simulation of TrpZip2, we centered the extended polypeptide chain (81 beads) into a box with dimensions 10 × 10 × 10 DPD units filled with 3064 polarizable coarse-grained waters. We simulated the protein for a total of 2000000 steps, with a timestep of $dt = 0.001$, which corresponds to a total simulation time of 50×10^{-6} s. For the simulation of TrpCage, we centered the extended coarse grained polypeptide chain consisting of 119 beads into a box with dimensions 12.5 × 12.5 × 12.5 DPD units and filled the box with 5863 polarizable CG waters. We simulated the protein for a total of 2000000 steps, with a timestep of $dt = 0.001$ which corresponds to a simulation time of 50×10^{-6} s. For GB1, we modeled the peptide as an extended chain with 309 beads in a box with dimensions 20 × 20 × 20 DPD units filled with 23279 coarse-grained waters. After an equilibration for 20000 steps, we centered the peptide into a smaller box with dimensions 14×14×14 DPD units and filled the box with 8433 CG waters. We then simulated the system for 5340000 steps corresponding to 0.1335×10^{-3} s. For the simulation of the WW-domain, Villin-headpiece and the Peripheral binding subunit, we centered each extended coarse grained peptide (WW-domain - 236 beads, Villin-headpiece - 185 beads, Peripheral binding subunit - 215 beads) into a box with dimensions 15 × 15 × 15 DPD units and added 10612 coarse grained waters. We simulated each system for 4800000 steps. For the simulations of the B-domain of Protein A, we centered the extended polypeptide chain (368 beads) in a box with dimensions 15 × 15 × 15 and filled it with 10606 coarse-grained waters. In all of our simulations, we used an electrostatic cutoff of 1.5 R_c and a cutoff for the other DPD forces of 1 R_c. We integrated the equations of motion with a timestep of 0.001 τ. For the all atom simulations of the backmapped structures of TrpCage and TrpZip2, we used the GROMOS 53a5 forcefield. AA-TrpZip2 was centered in a box with dimensions $3.34 \times 3.34 \times 3.34$ nm^3 and the box was filled with 1158 SPC/E waters. AA-TrpCage was centered in a box with dimensions $3.39 \times 3.39 \times 3.39$ nm^3 and the box was filled with 1184 SPC/E waters. We applied PME electrostatics to the system, while we used a shift function for the description of the Lennard-Jones interactions. Both potentials were truncated at a cutoff of 1.0 nm. We used no constraints and applied a timestep $dt = 1.0 \times 10^{-15}$ s to integrate the equations of motion. We simulated the system for a total of 10×10^{-9} s using the replica exchange MD (REMD) methodology [88]. We used 12 replicas in the temperature range from 300 to 322 K. The exchange statistics of this comparably short run was sufficient.

A.3 Results

A.3.1 Calibration simulations used to define model parameters

In the following, we describe the results of simulations, where we applied our coarse-grained model on folding of TrpZip2 and TrpCage. Both proteins have been extensively studied in detail, both experimentally and computationally, and serve as model systems for α-helical and β-strand folding. We chose these systems for the calibration, since both proteins cover the different aminoacid groups in our coarse-grained model, except group 5 containing cysteine and methionine (See Figures A.1 and A.3d). We start with the simulations of TrpZip2 using the model described above. In Figure A.3a, we show the probability distribution of TrpZip2 as function of $RMSD_{C\alpha-C\alpha}$ and the radius of gyration (Rg). It is important to note that interpretation of the results should be different from typical molecular dynamics analysis. Due to the backmapping procedure (see section A.2.6), the reported quantities have an associated error of ±0.4 nm, which we also see in the comparison with the experimental radii of gyration. This error is connected to the overall coarse-graining approach which we use. We observe that TrpZip2 resides with a higher probability between $RMSD_{C\alpha-C\alpha}$ of 0.32 nm and 0.7 nm and Rg ranging from 0.55 to 0.75 nm (experimental value 0.66 nm). The maxima in the probability is located at $RMSD_{C\alpha-C\alpha}$ ranging from 0.34 to 0.55 nm and Rg between 0.44 and 0.68 nm, where we find near-native configurations in which the loop is already formed, as well as collapsed configurations in which contacts exist between Trp-residues 2 and 11. In the native state, we find that the hairpin is aligned in a β-stranded form, while the Trp-residues 4 and 9 form a hydrophobic contact, which is in agreement with experimental data [26]. As a folding mechanism, we observe a zipper-like closure of the hairpin, which agrees well with a study by Swope and coworkers [177]. We observe that TrpZip2 folds along a 2-state like mechanism, with no favored intermediate near-native state. The lowest free energy is at $RMSD_{C\alpha-C\alpha}$ of 0.4 nm with a value of approximately $-3.9\ k_BT$. We note that the free energy value obtained with our coarse-grained model is in good agreement with results by Yang *et al.* ($-4k_BT$ for the native minimum) [250]. Finally, we emphasize that our coarse-grained model is not meant to be used as an alternative for all atomistic methods for *detailed* investigation the protein folding pathway, nevertheless it decays into the native state of this protein.

We continue with simulations of an α-helical peptide TrpCage [160]. In Figure A.3b, we show the probability distribution as function of $RMSD_{C\alpha-C\alpha}$ and the radius of gyration. We observe that the main conformer is located within

$RMSD_{C\alpha-C\alpha}$ of 0.55 nm and 0.75 nm and Rg between 0.5 and 0.65 nm (experimental value 0.79 nm), which corresponds to the folded state of TrpCage. In this state, which is very close to the native state of TrpCage, the α-helical element between residues Asn1 and Asp9 is formed and Trp6 points into the hydrophobic cleft of the protein. We find that the potential of mean force is approximately $-4.5\ k_BT$ for the folded state of the protein. This value is in good agreement with results from all-atom simulations [259, 30]. Finally, we mention that we observe a diffusion collision folding mechanism of TrpCage with formation of the α-helical element prior to the final collapse into the native structure. Similar pathways have been observed in experiments and simulations as major folding pathways of TrpCage [99, 100, 151]. We note though that the timescale and dynamics are faster in our coarse-grained approach than in all-atom simulations. Nevertheless, the model describes self-assembly of proteins in solution in good agrement with the experiment, without any restraints on the system. In the $RMSD_{C\alpha-C\alpha}$ as function of simulation time of both proteins, we see that the configurations are stable and the initial collapse occurs within the first steps of the simulation (approximately 50000 steps) (see Figure A.3c). This indicates that the unfolded region of both proteins is not sampled accurately, but rather the native state of both peptides.

A.3.2 Simulation of backmapped structures in all-atom MD simulations

We mapped the lowest RMSD structures of TrpCage and TrpZip2 back to the atomistic scale in order to verify our coarse-graining approach and to compare our measures (absolute errors) of the $RMSD_{C\alpha-C\alpha}$ and Rg with the explicit scale. For the subsequent simulations, we only chose a small simulation time of $10 \times 10^{-9}\ s$ and a low number of 12 replicas, so that we could ensure that we sample mainly the structure of the probability maximum in the coarse-grained simulation. For the simulations of TrpZip2 and TrpCage, we extracted the lowest RMSD structure from the CG-trajectory and mapped this structure to the all-atom (AA) description as implemented by Wassenaar *et al.* [238].

We start with the description of TrpZip2, where we find that the starting structure (backmapped from the coarse grained trajectory) already has a RMSD to the backbone of the native structure of 0.29 nm. In the REMD simulation, we find that the structure mainly resides at $RMSD_{C\alpha-C\alpha}$ of 0.3 nm, while also slightly unfolded conformations in $RMSD_{C\alpha-C\alpha}$ exist in the range between 0.5 nm and 0.8 nm (see Figure A.4a). The lowest RMSD structures correspond to $RMSD_{C\alpha-C\alpha}$

of 0.22 *nm* and Rg of 0.65 *nm*. In these lowest RMSD conformations, we find that the hydrophobic core itself is not as compact as in the native structure of TrpZip2 [26], and Trp2 is exposed to the solvent.

We continue with the description of TrpCage, where we see that the starting structure is only near native at $RMSD_{C\alpha-C\alpha}$ of 0.5 *nm*. We also observe that the TrpCage remains mainly at this RMSD value, while also lower RMSD values occur throughout this short REMD run (see Figure A.4b). We notice that in these lowest RMSD structures the 3-10 helix and the polyproline helix are well described, while the α-helical part is loosely packed. We mention that Trp6 is correctly aligned and prone for the formation of the hydrophobic core of this protein. The same counts for residues Asp9 and Arg16, which are on the same perimeter and are able to form the salt-bridge.

When we compare the measures in the all-atom and the coarse-grained simulations, we directly see that our estimate of the errors (see section A.2.6) are suitable for our model. We observe that the deviation in Rg is on the order of $\approx$ 0.4 *nm* while the deviation in the RMSD is $\approx$ $0.2-0.3$ *nm*. Additionally, we find that the RMSD curves as function of simulation time do not have a drift towards unfolded conformers, indicating that the coarse-grained structures represent a stable free energy minimum (see Figure A.4c).

We conclude that our calibration simulations using the new DPD based parameter set is suitable for the description of both proteins, TrpZip2 and TrpCage.

A.3.3 Validation on 5 different proteins

In the following, we present the results of the validation of our new methodology on folding of 5 different proteins : GB1, the WW-domain, domain B from Protein A, the peripherial binding subunit and Villin Headpiece. We start with describing our simulations on folding of GB1, a mixed α-β peptide. In this simulation, we observe that the protein collapses to a $RMSD_{C\alpha-C\alpha}$ of 0.9 *nm*, into a state which has the same tertiary structure as the native state of GB1, while the protein is in a more compact state than the native structure (see Figure A.5a,f). Here we observe that 3 of the 4 β-strands of the native structure are present with the α-helix between residues Asp22 and Gly37. The outer β-strand between residues Glu42 and Asp48 is not preserved as well as the other motifs of GB1.

In the simulation of the WW-domain, we observe 3 collapses into a near-native structure followed by a slow decrease in the $RMSD_{C\alpha-C\alpha}$ at $1.7*10^{-4}$ *s* to $RMSD_{C\alpha-C\alpha}$ of 0.65 *nm*. In the tertiary structure of this protein connected to the lowest $RMSD_{C\alpha-C\alpha}$, we observe 3 turns in the structure and 3 longitudinal extensions which represent the 3 β-stranded parts of this domain well. As in the case of GB1, we observe that the near native structure has the same tertiary structure as the native state, while the protein is not as compact as the native state (see Figure A.5b,g). In the case of Villin-headpiece, we observe that the protein constantly resides between $RMSD_{C\alpha-C\alpha}$ of 0.65 nm and 1 nm. In the analysis of the near native state of this protein obtained from our simulation, we observe that the tertiary structure is identical with the structure of the native state within the associated error (See Figure A.5c,h). As in the case of the WW-domain, the 3 turns between the alpha-helical regions of this protein are well described in the structure. Slight deviations from the (forwarded) native structure occur in the α-helical regions themselves, leading to a more compact state. Again, we emphasize that during the forward mapping procedure and due to our coarse-grained model a certain error is associated with the structural comparison between both structures (see section A.2.6). For the peripheral binding subunit, we observe a final collapse to a near native structure with $RMSD_{C\alpha-C\alpha}$ of 0.77 nm at a simulation time of $4.5*10^{-5}$ s. For this lowest RMSD structure, we observe a N-terminal part with 2 turns and 2 α-helical regions are well described, while the C-terminal region (residues 62 to residue 76) is extended in our structure and only partially forms an α-helix in the range between residues leucine 69 and Phe 76. In that case the structure is less compact than the native state, but describes the tertiary structure and secondary structure well (see Figures A.5d,i).

In the simulation of the B-domain of Protein A, we observe that the structure collapses in the native state with $RMSD_{C\alpha-C\alpha} = 0.77$ nm at a simulation time of $5.5 * 10^{-5}$ s. In this state, the protein has all 3 turns formed with the 3 helices in the regions (Gln11-His19, Asn24-Asp37 and Gln41-Gln56). In that case the 3 helices are packed more strongly, while the 3 helices in the native structure are more loosely packed (See Figures A.5e,j).

In summary, we have validated our model on 5 different proteins. In the cases of the WW-domain and the peripheral binding subunit, we obtained good agreement within the error associated with our forward mapping procedure. For the B-domain of protein A, we observed a higher packing compared to the native structure, while all 3 helices and the turns are preserved. In the case of GB1, we observe correct arrangement of secondary structure elements and close alignment in the tertiary structure, while one outer β-sheet is not conserved.

A.4 Summary

In this section, we described a novel coarse-grained polarizable protein model for Dissipative Particle Dynamics simulation technique. This model is based on polarizable backbone dipoles which enable polypeptides to form their native structure. We define the model parameters based on folding of the 2 peptides TrpCage and TrpZip2, with the condition that the main conformers of the coarse-grained model agree with the experimental structures [26, 160]. We validate the model by simulating folding of 5 additional single domain proteins, i.e. GB1, the WW-domain, the peripheral binding subunit, Villin headpiece and the B-domain of Protein A. We believe that the presented model is suitable for the coarse-grained description of proteins and has the potential to improve sampling of native states in coarse-grained protein simulations. Currently, we are extending the model to simulations of peptide aggregation, where self-assembly processes play an important role.

Bibliography

[1] Y. Afshar, F. Schmid, A. Pishevar, and S. Worley. Exploiting seeding of random number generators for efficient domain decomposition parallelization of dissipative particle dynamics. *Comput. Phys. Commun.*, 184(4): 1119–1128, 2013.

[2] H. Agrawal, M. Zelisko, L. Liu, and P. Sharma. Rigid proteins and softening of biological membranes - with application to HIV-induced cell membrane softening. *Scientific Reports*, 6(April):25412, 2016.

[3] C. Alix-Panabieres and K. Pantel. Challenges in circulating tumour cell research. *Nat. Rev. Cancer*, 14(9):623–631, 2014.

[4] D. Alizadehrad, Y. Imai, K. Nakaaki, T. Ishikawa, and T. Yamaguchi. Parallel simulation of cellular flow in microvessels using a particle method. *Journal of Biomechanical Science and Engineering*, 7(1):57–71, 2012.

[5] J. Anderson, C. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Comput. Phys.*, 227(10):5342–5359, 2008.

[6] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, 2015.

[7] M. Baaden and S. J. Marrink. Open boundary molecular dynamics. *Curr. Opin. Struct. Biol.*, 23:878–886, 2013.

[8] G. Bell. Models for the specific adhesion of cells to cells. *Science*, 200 (4342):618–627, 1978.

[9] M. Bernaschi, M. Bisson, T. Endo, S. Matsuoka, M. Fatica, and S. Melchionna. Petaflop biofluidics simulations on a two million-core system. In *Proc. of 2011 Intl. Conf. for High Perf. Computing, Networking, Storage and Analysis*, SC ’11, pages 1–12, New York, NY, USA, 2011. ACM.

[10] M. Bernaschi, M. Bisson, M. Fatica, and S. Melchionna. 20 petaflops simulation of proteins suspensions in crowding conditions. In *Proc. of 2013 Intl. Conf. for High Perf. Computing, Networking, Storage and Analysis*, SC '13, pages 2:1–2:11, New York, NY, USA, 2013. ACM.

[11] M. Bernaschi, M. Bisson, M. Fatica, S. Melchionna, and S. Succi. Petaflop hydrokinetic simulations of complex flows on massive gpu clusters. *Computer Physics Communications*, 184(2):329 – 341, 2013.

[12] T. C. Bidone, T. Kim, M. A. Deriu, U. Morbiducci, and R. D. Kamm. Multiscale impact of nucleotides and cations on the conformational equilibrium, elasticity and rheology of actin filaments and crosslinked networks. *Biomechanics and Modeling in Mechanobiology*, 14(5):1143–1155, 2015.

[13] J. J. Bishop, A. S. Popel, M. Intaglietta, and P. C. Johnson. Effect of erythrocyte aggregation on velocity profiles in venules. *Am. J. Physiol. Heart Circ. Physiol.*, 281:H939–H950, 2001.

[14] D. Boal. Computer simulation of a model network for the erythrocyte cytoskeleton. *Biophys. J.*, 67(2):521 – 529, 1994.

[15] D. Boal, U. Seifert, and J. Shillcock. Negative poisson ratio in two-dimensional networks under tension. *Phys. Rev. E*, 48:4274–4283, Dec 1993.

[16] D. W. Bolen and G. D. Rose. Structure and energetics of the hydrogen-bonded backbone in protein folding. *Annu. Rev. Biochem.*, 77:339–362, 2008.

[17] C. Borau, T. Kim, T. Bidone, J. M. Garcia-Aznar, and R. D. Kamm. Dynamic mechanisms of cell rigidity sensing: Insights from a computational model of actomyosin networks. *PLoS ONE*, 7(11):e49174, 11 2012.

[18] K. Boryczko, W. Dzwinel, and D. Yuen. Modeling heterogeneous mesoscopic fluids in irregular geometries using shared memory systems. *Mol Simul.*, 31:45–56, 2005.

[19] H. Bow, I. V. Pivkin, M. Diez-silva, S. J. Goldfless, M. Dao, and J. C. Niles. A microfabricated deformability-based flow cytometer with application to malaria. *Lab on a chip*, 11(6):1065–1073, 2011.

[20] T. Bradley. Hyper-Q example, 2012.

[21] C. Brangwynne, F. MacKintosh, S. Kumar, N. Geisse, J. Talbot, L. Mahadevan, K. Parker, D. Ingber, and D. Weitz. Microtubules can bear enhanced compressive loads in living cells because of lateral reinforcement. *J. Cell Biol.*, (173):733–741, 2006.

[22] P. Carmeliet and R. Jain. Angiogenesis in cancer and other diseases. *Nature*, 407:249–257, 2000.

[23] R. Chiotaki, H. Polioudaki, and P. A. Theodoropoulos. Differential nuclear shape dynamics of invasive and non-invasive breast cancer cells are associated with actin cytoskeleton organization and stability. *Biochemistry and Cell Biology*, 92:287–295, 2014.

[24] K. Choi, A. Ng, R. Fobel, and A. R. Wheeler. Digital Microfluidics. *Annual Review of Analytical Chemistry*, 5(1):413–440, 2012.

[25] J. R. Clausen, D. Reasor, and C. K. Aidun. Parallel performance of a lattice-Boltzmann/finite element cellular blood flow solver on the IBM Blue Gene/P architecture. *Comput. Phys. Commun.*, 181(6):1013–1020, 2010.

[26] A. G. Cochran, N. J. Skelton, and M. A. Starovasnik. Tryptophan zippers: Stable, monomeric beta-hairpins. *Proceedings of the National Academy of Sciences*, 98(10):5578–5583, 2001.

[27] E. Corbin, F. Kong, C. T. Lim, W. P. King, and R. Bashir. Biophysical properties of human breast cancer cells measured using silicon MEMS resonators and atomic force microscopy. *Lab on a chip*, 15(3):839–47, 2015.

[28] S. E. Cross, J. Yu-Sheng, R. Jianyu, and J. K. Gimzewski. Nanomechanical analysis of cells from cancer patients. *Nature Nanotechnology*, 2(12):780–783, 2007.

[29] C. Cupelli, T. Borchardt, T. Steiner, N. Paust, R. Zengerle, and M. Santer. Leukocyte enrichment based on a modified pinched flow fractionation approach. *Microfluid Nanofluidics*, 14(3-4):551–563, 2013.

[30] R. Day, D. Paschek, and A. E. Garcia. Microsecond simulations of the folding/unfolding thermodynamics of the trp-cage miniprotein. *Proteins*, 78:1889–1899, 2010.

[31] Y. Deng and B. Roux. Computations of standard binding free energies with molecular dynamics simulations. *The Journal of Physical Chemistry B*, 113 (8):2234–2246, 2009.

[32] D. E. Discher, D. H. Boal, and S. K. Boey. Simulations of the erythrocyte cytoskeleton at large deformation. II. Micropipette aspiration. *Biophys. J.*, 75:1584–1597, 1998.

[33] Y. Dong, A. Skelley, K. Merdek, K. Sprott, C. Jiang, W. E. Pierceall, J. Lin, M. Stocum, W. Carney, and D. A. Smirnov. Microfluidics and circulating tumor cells. *The Journal of Molecular Diagnostics*, 15(2):149 – 157, 2013.

[34] E. P. Dowling, W. Ronan, G. Ofek, V. S. Deshpande, R. M. McMeeking, K. Athanasiou, and J. P. McGarry. The effect of remodelling and contractility of the actin cytoskeleton on the shear resistance of single cells: a computational and experimental investigation. *Journal of The Royal Society Interface*, 9(July):3469–3479, 2012.

[35] E. Du, M. Dao, and S. Suresh. Quantitative biomechanics of healthy and diseased human red blood cells using dielectrophoresis in a microfluidic system. *Extreme Mechanics Letters*, 1:35 – 41, 2014.

[36] M. M. Dupin, I. Halliday, C. M. Care, L. Alboul, and L. L. Munn. Modeling the flow of dense suspensions of deformable particles in three dimensions. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 75(6):1–17, 2007.

[37] G. Enden and A. S. Popel. A numerical study of the shape of the surface separating flow into branches in microvascular bifurcations. *J. Biomech. Eng.*, 114:398–405, 1991.

[38] G. Enden and A. S. Popel. A numerical study of plasma skimming in small vascular bifurcations. *J. Biomech. Eng.*, 116:79–88, 1994.

[39] E. Esch, A. Bahinski, and D. Huh. Organs-on-chips at the frontiers of drug discovery. *Nat. Rev. Drug Discov.*, 14(4):248–260, 2015.

[40] P. Español. Hydrodynamics from dissipative particle dynamics. *Phys. Rev. E*, 52:1734–1742, Aug 1995.

[41] P. Español and M. Revenga. Smoothed dissipative particle dynamics. *Phys. Rev. E*, 67:026705, Feb 2003.

[42] P. P. Ewald. Die berechnung optischer und elektrostatischer gitterpotentiale. *Ann. Phys.*, 64:253, 1921.

[43] X. Fan, N. Phan-Thien, S. Chen, X. Wu, and T. Yong Ng. Simulating flow of DNA suspension using dissipative particle dynamics. *Physics of Fluids*, 18(6):063102–063102, June 2006.

[44] Y. Fang and K. W. C. Lai. Modeling the mechanics of cells in the cell-spreading process driven by traction forces. *Phys. Rev. E*, 93:042404, 2016.

[45] D. Fedosov and G. Gompper. White blood cell margination in microcirculation. *Soft Matter*, 10:2961–70, 2014.

[46] D. Fedosov and G. Gompper. White blood cell margination in microcirculation. *Soft matter*, 10:2961–70, 2014.

[47] D. Fedosov, I. V. Pivkin, and G. E. Karniadakis. Velocity limit in DPD simulations of wall-bounded flows. *J. Comput. Phys.*, 227(4):2540–2559, 2008.

[48] D. Fedosov, W. Pan, B. Caswell, G. Gompper, and G. E. Karniadakis. Predicting human blood viscosity in silico. *Proceedings of the National Academy of Sciences*, 108(29):11772–11777, 2011.

[49] D. Fedosov, W. Pan, B. Caswell, G. Gompper, and G. E. Karniadakis. Predicting human blood viscosity in silico. *Proceedings of the National Academy of Sciences*, 108(29):11772–11777, 2011.

[50] D. A. Fedosov. *Multiscale Modeling of Blood Flow and Soft Matter*. Phd thesis, 2009.

[51] D. A. Fedosov, H. Lei, B. Caswell, S. Suresh, and G. E. Karniadakis. Multiscale modeling of red blood cell mechanics and blood flow in malaria. *PLOS Computational Biology*, 7(12):1–13, 12 2011.

[52] D. A. Fedosov, J. Fornleitner, and G. Gompper. Margination of white blood cells in microcapillary flow. *Phys. Rev. Lett.*, 108:028104, Jan 2012.

[53] B. M. Fenton, R. T. Carr, and G. R. Cokelet. Nonuniform red cell distribution in 20 to 100 μm bifurcations. *Microvas. Res.*, 29:103–126, 1985.

[54] J. M. Ferrer, H. Lee, J. Chen, B. Pelz, F. Nakamura, R. D. Kamm, and M. J. Lang. Measuring molecular rupture forces between single actin filaments and actin-binding proteins. *Proceedings of the National Academy of Sciences*, 105(27):9221–9226, 2008.

[55] E. G. Flekkøy and P. V. Coveney. From molecular dynamics to dissipative particle dynamics. *Phys. Rev. Lett.*, 83(9):1775–1778, 1999.

[56] E. G. Flekkøy, R. Delgado-Buscalioni, and P. V. Coveney. Flux boundary conditions in particle simulations. *Phys. Rev. E*, 72:026703, Aug 2005.

[57] A. G. Fletcher, F. Cooper, R. E. Baker, and A. G. Fletcher. Mechanocellular models of epithelial morphogenesis. *Phil. Trans. R. Soc. B*, (372), 2017.

[58] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic, New York, 1996.

[59] J. Freund. Numerical simulation of flowing blood cells. *Annual Review of Fluid Mechanics*, 46(1):67–95, 2014.

[60] M. Friedel and J.-E. Shea. Self-assembly of peptides into a beta-barrel motif. *J. Chem. Phys.*, 120:5809, 2004.

[61] Y. C. Fung. Stochastic flow in capillary blood vessels. *Microvasc. Res.*, 5: 34–48, 1973.

[62] M. L. Gardel, J. H. Shin, F. C. MacKintosh, L. Mahadevan, P. Matsudaira, and D. a. Weitz. Elastic behavior of cross-linked and bundled actin networks. *Science (New York, N.Y.)*, 304(5675):1301–1305, 2004.

[63] G. Gerlitz and M. Bustin. The role of chromatin structure in cell migration. *Trends Cell Biol.*, 21(1):6–11, 2012.

[64] J. Glaser, T. Nguyen, J. Anderson, P. Lui, F. Spiga, J. Millan, D. Morse, and S. Glotzer. Strong scaling of general-purpose molecular dynamics simulations on gpus. *Comput. Phys. Commun.*, 192:97–107, 2015.

[65] H. L. Goldsmith and S. Spain. Margination of leukocytes in blood flow through small tubes. *Microvascular Research*, 27(2):204–222, 1984.

[66] J. Gonzalez and R. Pino. A random number generator based on unpredictable chaotic functions. *Comput. Phys. Commun.*, 120(2–3):109–114, 1999.

[67] M. Gonzalez-Melchor, E. Mayoral, M. E. Velazquez, and J. Alejandre. Electrostatic interactions in dissipative particle dynamics using the ewald sums. *J. Chem. Phys.*, 125:224107, 2006.

[68] D. R. Gossett, H. T. K. Tse, S. A. Lee, Y. Ying, A. G. Lindgren, O. O. Yang, J. Rao, A. T. Clark, and D. Di Carlo. Hydrodynamic stretching of single cells for large population mechanical phenotyping. *Proceedings of the National Academy of Sciences*, 109(20):7630–7635, 2012.

[69] H. Gouda, M. Shiraishi, H. Takahashi, K. Kato, H. Torigoe, Y. Arata, , and I. Shimada. Nmr study of the interaction between the b domain of staphylococcal protein a and the fc portion of immunoglobulin g. *Biochemistry*, 37(1):129–136, 1998.

[70] R. D. Groot. Electrostatic interactions in dissipative particle dynamics, simulation of polyelectrolytes and anionic surfactants. *The Journal of Chemical Physics*, 118(24):11265–11277, 2003.

[71] R. D. Groot and P. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *J. Chem. Phys.*, 107(11): 4423–4435, 1997.

[72] J. Guck, S. Schinkinger, B. Lincoln, F. Wottawah, S. Ebert, M. Romeyke, D. Lenz, H. M. Erickson, R. Ananthakrishnan, D. Mitchell, J. Käs, S. Ulvick, and C. Bilby. Optical deformability as an inherent cell marker for testing malignant transformation and metastatic competence. *Biophysical journal*, 88(May):3689–3698, 2005.

[73] K. Guevorkian, M.-J. Colbert, M. Durth, S. Dufour, and F. m. c. Brochard-Wyart. Aspiration of biological viscoelastic drops. *Phys. Rev. Lett.*, 104: 218101, 2010.

[74] G. Guigas, D. Morozova, and M. Weiss. Exploring membrane and protein dynamics with dissipative particle dynamics. *Adv. Protein Chem. Struct. Biol.*, 85:143–182, 2011.

[75] F. Guilak, J. R. Tedrow, and R. Burgkart. Viscoelastic properties of the cell nucleus. *Biochemical and biophysical research communications*, 269: 781–786, 2000.

[76] Q. Guo, S. Park, and H. Ma. Microfluidic micropipette aspiration for measuring the deformability of single cells. *Lab Chip*, 12:2687–2695, 2012.

[77] Q. Guo, S. P. Duffy, K. Matthews, A. T. Santoso, M. D. Scott, and H. Ma. Microfluidic analysis of red blood cell deformability. *Journal of Biomechanics*, 47(8):1767 – 1776, 2014.

[78] Y. Guyot, B. Smeets, T. Odenthal, R. Subramani, F. P. Luyten, H. Ramon, I. Papantoniou, and L. Geris. Immersed Boundary Models for Quantifying Flow-Induced Mechanical Stimuli on Stem Cells Seeded on 3D Scaffolds in Perfusion Bioreactors. *PLoS Computational Biology*, 12(9):1–21, 2016.

[79] A. Hall. The cytoskeleton and cancer. *Cancer and Metastasis Reviews*, 28 (1):5–14, 2009.

[80] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. Gromacs 4:âĂŁ algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.*, 4:435–447, 2008.

[81] C. Hijon, P. Espanol, E. Vanden-Eijnden, and R. Delgado-Buscalioni. Mori-zwanzig formalism as a practical computational tool. *Faraday Discuss.*, 144:301–322, 2010.

[82] R. D. Hills, L. Lu, and G. A. Voth. Multiscale coarse-graining of the protein energy landscape. *PLOS Comput. Biol.*, 6:e1000827, 2010.

[83] D. F. Hinz, A. Panchenko, T.-Y. Kim, and E. Fried. Motility versus fluctuations in mixtures of self-motile and passive agents. *Soft Matter*, 10: 9082–9089, 2014.

[84] P. Hoogerbrugge and J. Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *EPL (Europhys. Lett.)*, 19 (3):155, 1992.

[85] H. Hou, Q. Li, G. Lee, P. Kumar, C. Ong, and C. Lim. Deformability study of breast cancer cells using microfluidics. *Biomedical Microdevices*, 11: 557–564, 2009.

[86] L. Huang, E. Cox, R. Austin, and J. Sturm. Continuous particle separation through deterministic lateral displacement. *Science*, 304(5673):987–990, 2004.

[87] L. R. Huang, E. C. Cox, R. H. Austin, and J. C. Sturm. Continuous particle separation through deterministic lateral displacement. *Science*, 304 (5673):987–990, 2004.

[88] K. Hukushima and K. Nemoto. Exchange monte carlo method and application to spin glass simulations. *J. Phys. Soc. Jpn.*, 65:1604–1608, 1996.

[89] T. Hyakutake and S. Nagai. Numerical simulation of red blood cell distributions in three-dimensional microvascular bifurcations. *Microvascular Research*, 97:115 – 123, 2015.

[90] Y. Imai, H. Kondo, T. Ishikawa, C. T. Lim, and T. Yamaguchi. Modeling of hemodynamics arising from malaria infection. *Journal of Biomechanics*, 43(7):1386 – 1393, 2010.

[91] S. Izvekov and G. A. Voth. Systematic coarse-graining of nanoparticle interactions in molecular dynamics simulation. *J. Phys. Chem. B*, 109: 2469–2473, 2005.

[92] W. Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[93] M. Jamroz, M. Orozco, A. Kolinski, and S. Kmiecik. Cabs-flex: server for fast simulation of protein structure fluctuations. *J. Chem. Theory Comput.*, 9:119–125, 2013.

[94] K. M. Jan and S. Chien. Influence of the ionic composition of fluid medium on red cell aggregation. *The Journal of general physiology*, 61:655–668, 1973.

[95] P. Janmey. *Cell Membranes and the Cytoskeleton*, volume 1. 1995.

[96] F. Janoschek, F. Toschi, and J. Harting. Simplified particulate model for coarse-grained hemodynamics simulations. *Phys Rev E.*, 82:056710, 2010.

[97] F. Janoschek, F. Toschi, and J. Harting. Simulations of blood flow in plain cylindrical and constricted vessels with single cell resolution. *Macromol. Theory Simul.*, 20(7):562–570, 2011.

[98] W. Jiang, T. Yan, Y. Wang, , and G. Voth. Molecular dynamics simulation of the energetic room-temperature ionic liquid, 1-hydroxyethyl-4-amino-1,2,4-triazolium nitrate (heatn). *The Journal of Physical Chemistry B*, 112 (10):3121–3131, 2008.

[99] J. Juraszek and P. G. Bolhuis. Sampling the multiple folding mechanisms of trp-cage in explicit solvent. *Proceedings of the National Academy of Sciences*, 103:15859–15864, 2006.

[100] J. Juraszek and P. G. Bolhuis. Rate constant and reaction coordinate of trp-cage folding in explicit water. *Biophys. J.*, 95:4246–4257, 2008.

[101] Y. N. Kalia, S. M. Brocklehurst, S. S. Hipps, E. Appella, K. Sakaguchi, and R. N. Perham. The high-resolution structure of the peripheral subunit-binding domain of dihydrolipoamide acetyltransferase from the pyruvate dehydrogenase multienzyme complex of bacillus stearothermophilus. *Journal of Molecular Biology*, 230(1):323 – 341, 1993.

[102] U. Kapur and E. M. Wojcik. Follicular neoplasm of the thyroid–vanishing cytologic diagnosis? *Diagnostic cytopathology*, 35(8):525–528, 2007.

[103] N. Karabacak, P. Spuhler, F. Fachin, E. Lim, V. Pai, E. Ozkumur, J. Martel, N. Kojic, K. Smith, P. Chen, J. Yang, H. Hwang, B. Morgan, J. Trautwein, T. Barber, S. Stott, S. Maheswaran, R. Kapur, D. Haber, and M. Toner. Microfluidic, marker-free isolation of circulating tumor cells from blood samples. *Nature protocols*, 9(3):694–710, 2014.

[104] D. Kardas, U. Nackenhorst, and D. Balzani. Computational model for the cell-mechanical response of the osteocyte cytoskeleton based on self-stabilizing tensegrity structures. *Biomechanics and Modeling in Mechanobiology*, 12(1):167–183, 2013.

[105] G. Karniadakis, A. Beskok, and N. Aluru. *Microflows and nanoflows: fundamentals and simulation*, volume 29. Springer Science & Business Media, 2005.

[106] E. E. Keaveny, I. V. Pivkin, M. Maxey, and G. E. Karniadakis. A comparative study between dissipative particle dynamics and molecular dynamics for simple-and complex-geometry flows. *J Chem Phys*, 123(10), 2005.

[107] M.-C. Kim, C. Kim, L. Wood, D. Neal, R. D. Kamm, and H. H. Asada. Integrating focal adhesion dynamics, cytoskeleton remodeling, and actin motor activity for predicting cell migration on 3D curved surfaces of the extracellular matrix. *Integrative Biology*, 4(11):1386, 2012.

[108] M. C. Kim, D. M. Neal, R. D. Kamm, and H. H. Asada. Dynamic Modeling of Cell Migration and Spreading Behaviors on Fibronectin Coated Planar Substrates and Micropatterned Geometries. *PLoS Computational Biology*, 9(2), 2013.

[109] M. C. Kim, J. Whisler, Y. R. Silberberg, R. D. Kamm, and H. H. Asada. Cell Invasion Dynamics into a Three Dimensional Extracellular Matrix Fibre Network. *PLoS Computational Biology*, 11(10):1–29, 2015.

[110] T. Kim, W. Hwang, and R. D. Kamm. Computational analysis of a cross-linked actin-like network. *Experimental Mechanics*, 49(1):91–104, 2009.

[111] T. Kim, W. Hwang, H. Lee, and R. D. Kamm. Computational analysis of viscoelastic properties of crosslinked actin networks. *PLoS Computational Biology*, 5(7), 2009.

[112] T. Kim, M. L. Gardel, and E. D. Munro. Determinants of fluidlike behavior and effective viscosity in cross-linked actin networks. *Biophysical Journal*, 106(3):526–534, 2014.

[113] P. A. Kollman. Advances and continuing challenges in achieving realistic and predictive simulations of the properties of organic and biological molecules. *Acc. Chem. Res.*, 29:461–469, 1996.

[114] P. Kollmannsberger and B. Fabry. Linear and Nonlinear Rheology of Living Cells. *Annual Review of Materials Research*, 41:75–97, 2011.

[115] P. Koumoutsakos, I. V. Pivkin, and F. Milde. The fluid mechanics of cancer and its therapy. *Annu. Rev. Fluid Mech.*, 45(1):325–355, 2013.

[116] T. Krüger, M. Gross, D. Raabe, and F. Varnik. Crossover from tumbling to tank-treading-like motion in dense simulated suspensions of red blood cells. *Soft Matter*, 9:9008–9015, 2013.

[117] T. Krüger, D. Holmes, and P. Coveney. Deformability-based red blood cell separation in deterministic lateral displacement devices—a simulation study. *Biomicrofluidics*, 8(5):054114, 2014.

[118] J. Kubelka, J. Hofrichter, and W. A. Eaton. The protein folding speed limit. *Current Opinion in Structural Biology*, 14(1):76–88, 2004.

[119] J. R. Kuhn and T. D. Pollard. Real-time measurements of actin filament polymerization by total internal reflection fluorescence microscopy. *Biophysical Journal*, 88(2):1387 – 1402, 2005.

[120] S. Kumar and V. M. Weaver. Mechanics, malignancy, and metastasis: The force journey of a tumor cell. *Cancer and Metastasis Reviews*, 28(1):113–127, 2009.

[121] C. D. Lammerding and Jan. Nuclear Mechanics in Cancer. *Adv Exp Med Biol.*, 773:435–470, 2014.

[122] P. L'Ecuyer and R. Simard. Testu01: Ac library for empirical testing of random number generators. *ACM Trans. Math. Software (TOMS)*, 33(4): 22, 2007.

[123] G. Y. Lee and C. T. Lim. Biomechanics approaches to studying human diseases. *Trends in Biotechnology*, 25(3):111 – 118, 2007.

[124] H. Lei and G. E. Karniadakis. Quantifying the rheological and hemodynamic characteristics of sickle cell anemia. *Biophys. J.*, 102(2):185–194, 2012.

[125] H. Lei, D. A. Fedosov, and G. E. Karniadakis. Time-dependent and outflow boundary conditions for dissipative particle dynamics. *J. Comput. Phys.*, 230:3765–3779, 2011.

[126] H. Lei, D. Fedosov, B. Caswell, and G. E. Karniadakis. Blood flow in small tubes: quantifying the transition to the non-continuum regime. *J. Fluid Mech.*, 722:214–239, 2013.

[127] H. Lei, D. Fedosov, B. Caswell, and G. E. Karniadakis. Blood flow in small tubes: quantifying the transition to the non-continuum regime. *J. Fluid Mech.*, 722:214–239, 2013.

[128] S. Levine, M. Levine, K. a. Sharp, and D. E. Brooks. Theory of the electrokinetic behavior of human erythrocytes. *Biophysical journal*, 42(2): 127–135, 1983.

[129] H. Li and G. Lykotrafitis. Erythrocyte membrane model with explicit description of the lipid bilayer and the spectrin network. *Biophysical Journal*, 107(3):642–653, 2014.

[130] J. Li, D. Liao, and S. Yip. Coupling continuum to molecular-dynamics simulation: Reflecting particle method and the field estimator. *Phys. Rev. E*, 57:7259–7267, 1998.

[131] J. Li, M. Dao, C. Lim, and S. Suresh. Spectrin-level modeling of the cytoskeleton and optical tweezers stretching of the erythrocyte. *Biophys. J.*, 88(5):3707–3719, 2005.

[132] Q. Li. *Understanding structure-mechanical property relationship of breast cancer cells*. PhD thesis, National University of Singapore, 2009.

[133] X. Li, A. S. Popel, and G. E. Karniadakis. Blood-plasma separation in Y-shaped bifurcating microfluidic channels: a dissipative particle dynamics simulation study. *Physical Biology*, 9:026010, 2012.

[134] X. Li, P. Vlahovska, and G. Karniadakis. Continuum- and particle-based modeling of shapes and dynamics of red blood cells in health and disease. *Soft Matter*, 9(1):28–37, 2013.

[135] X. Li, Z. Peng, H. Lei, M. Dao, and G. E. Karniadakis. Probing red blood cell mechanics, rheology and dynamics with a two-component multi-scale model. *Phil. Trans. R. Soc. A*, 372(2021), 2014.

[136] F. Liu, D. Wu, X. Wu, and K. Chen. Analyses of the cell mechanical damage during microinjection. *Soft Matter*, 11(7):1434–1442, 2015.

[137] H. Liu, J. Wen, Y. Xiao, J. Liu, S. Hopyan, M. Radisic, C. A. Simmons, and Y. Sun. In situ mechanical characterization of the cell nucleus by atomic force microscopy. *ACS Nano*, 8(4):3821–3828, 2014.

[138] Z. Liu, Y. Lee, J. H. Jang, Y. Li, X. Han, K. Yokoi, M. Ferrari, L. Zhou, and L. Qin. Microfluidic cytometric analysis of cancer cell transportability and invasiveness. *Scientific reports*, 5:14272, 2015.

[139] H. Lodish, A. Berk, D. Baltimore, P. Matsudaira, S. Zipursky, and J. Darnell. *Molecular Cell Biology*. W. H. Freeman, 1995.

[140] K. Lykov, X. Li, H. Lei, I. V. Pivkin, and G. E. Karniadakis. Inflow/outflow boundary conditions for particle-based blood flow simulations: Application to arterial bifurcations and trees. *PLoS Comput Biol*, 11(8):e1004410, 08 2015.

[141] K. Lykov, Y. Nematbakhsh, M. Shang, C. T. Lim, and I. V. Pivkin. Probing eukaryotic cell mechanics via mesoscopic simulations. *PLoS Computational Biology*, in press.

[142] M. J. Macias, V. Gervais, C. Civera, and H. Oschkinat. Structural analysis of ww domains and design of a ww prototype. *Nat. Struct. Biol.*, 7:375, 2000.

[143] S. J. Marrink and D. P. Tieleman. Perspective on the martini model. *Chem. Soc. Rev.*, 42:6801–6822, 2013.

[144] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. de Vries. The martini force field: a coarse grained model for biomolecular simulations. *The Journal of Physical Chemistry B*, 111(27):7812–7824, 2007.

[145] G. Marsaglia and A. Zaman. The KISS generator. Technical report, Dept. of Statistics, University of Florida, 1993.

[146] M. Masella, D. Borgis, and P. Cuniasse. Combining a polarizable force-field and a coarse-grained polarizable solvent model. ii. accounting for hydrophobic effects. *J. Comput. Chem.*, 29:1707–1724, 2008.

[147] S. McFaul, B. Lin, and H. Ma. Cell separation based on size and deformability using microfluidic funnel ratchets. *Lab Chip*, 12:2369–2376, 2012.

[148] C. J. McKnight, P. T. Masudaira, and P. S. Kim. Nmr structure of the 35-residue villin headpiece subdomain. *Nat. Struct. Biol.*, 4:180–184, 1997.

[149] J. L. McWhirter, H. Noguchi, and G. Gompper. Flow-induced clustering and alignment of vesicles and red blood cells in microcapillaries. *Proceedings of the National Academy of Sciences*, 106(15):6039–6043, 2009.

[150] J. L. McWhirter, H. Noguchi, and G. Gompper. Deformation and clustering of red blood cells in microcapillary flows. *Soft Matter*, 7:10967–10977, 2011.

[151] H. Meuzelaar, K. A. Marino, A. Huerta-Viga, M. R. Panman, L. E. J. Smeenk, A. J. Kettelarij, P. T. J. H. van Maarseveen, P. G. Bolhuis, and S. Woutersen. Folding dynamics of the trp-cage miniprotein: Evidence for a native-like intermediate from combined time-resolved vibrational spectroscopy and molecular dynamics simulations. *J. Phys. Chem. B*, 117: 11490–11501, 2013.

[152] J. Michel, M. Orsi, and J. W. Essex. Prediction of partition coefficients by multiscale hybrid atomic-level/coarse-grain simulations. *J. Phys. Chem. B*, 112:657–660, 2008.

[153] F. Milde, M. Bergdorf, and P. Koumoutsakos. A hybrid model for three-dimensional simulations of sprouting angiogenesis. *Biophys. J.*, 95:3146–3160, 2008.

[154] J. P. Mills, M. Diez-Silva, D. J. Quinn, M. Dao, M. J. Lang, K. S. W. Tan, C. T. Lim, G. Milon, P. H. David, O. Mercereau-Puijalon, S. Bonnefoy, and

S. Suresh. Effect of plasmodial resa protein on deformability of human red blood cells harboring plasmodium falciparum. *Proceedings of the National Academy of Sciences*, 104(22):9213–9217, 2007.

[155] A. Morriss-Andrews and J.-E. Shea. Simulations of protein aggregation: Insights from atomistic and coarse-grained models. *J. Phys. Chem. Lett.*, 5:1899–1908, 2014.

[156] K. Muller, D. Fedosov, and G. Gompper. Margination of micro- and nanoparticles in blood flow and its effect on drug delivery. *Sci. Rep.*, 4, 2014.

[157] J. W. Mullinax and W. G. Noid. Recovering physical potentials from a model protein databank. *Proceedings of the National Academy of Sciences*, 107:19867–19872, 2010.

[158] T. Murtola, T. Róg, E. Falck, M. Karttunen, and I. Vattulainen. Multiscale simulations of domain formation in single-component phospholipid bilayers. *Chemistry and Physics of Lipids*, 149:S15, 2007.

[159] L. Nan, Z. Jiang, and X. Wei. Emerging microfluidic devices for cell lysis: a review. *Lab Chip*, 14:1060–1073, 2014.

[160] J. W. Neidigh and R. M. Fesinmeyer. Designing a 20-residue protein. *Nat.Struct.Biol.*, 9:425–430, 2002.

[161] B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott. Microrobots for minimally invasive medicine. *Annu. Rev. Biomed. Eng.*, 12:55–85, 2010.

[162] Y. Nematbakhsh and C. T. Lim. Cell biomechanics and its applications in human disease diagnosis. *Acta Mechanica Sinica*, 31(2):268–273, 2015.

[163] M. Neri, C. Anselmi, M. Cascella, A. Maritan, and P. Carloni. Coarse-grained model of proteins incorporating atomistic detail of the active site. *Phys. Rev. Lett.*, 95:218102, Nov 2005.

[164] T. Nguyen and S. Plimpton. Accelerating dissipative particle dynamics simulations for soft matter systems. *Comput. Mater. Sci.*, 100:173–180, 2015.

[165] H. Noguchi and G. Gompper. Shape transitions of fluid vesicles and red blood cells in capillary flows. *Proceedings of the National Academy of Sciences*, 102(40):14159–14164, 2005.

[166] Nvidia Corporation. Sharing a GPU between MPI processes: Multi-Process Service (MPS) Overview, 2013.

[167] O. Otto, P. Rosendahl, A. Mietke, S. Golfier, C. Herold, D. Klaue, S. Girardo, S. Pagliara, A. Ekpenyong, A. Jacobi, M. Wobus, N. Töpfner, U. F. Keyser, J. Mansfeld, E. Fischer-Friedrich, and J. Guck. Real-time deformability cytometry: on-the-fly cell mechanical phenotyping. *Nature Methods*, 12 (3), 2015.

[168] W. Pan, I. V. Pivkin, and G. E. Karniadakis. Single-particle hydrodynamics in dpd: A new formulation. *Europhys. Lett.*, 84(1):10012, 2008.

[169] K. Pantel and M. Speicher. The biology of circulating tumor cells. *Oncogene*, 35(10):1216–1224, 2015.

[170] Z. Peng, X. Li, I. V. Pivkin, M. Dao, G. E. Karniadakis, and S. Suresh. Lipid bilayer and cytoskeletal interactions in a red blood cell. *Proceedings of the National Academy of Sciences*, 110:13356–13361, 2013.

[171] K. Perktold, R. O. Peter, M. Resch, and G. Langs. Pulsatile non-newtonian blood flow in three-dimensional carotid bifurcation models: a numerical study of flow phenomena under different bifurcation angles. *J. Biomed. Eng.*, 13:507–515, 1991.

[172] E. K. Peter and I. V. Pivkin. A polarizable coarse-grained water model for dissipative particle dynamics. *The Journal of Chemical Physics*, 141(16): 164506, 2014.

[173] E. K. Peter, K. Lykov, and I. V. Pivkin. A polarizable coarse-grained protein model for dissipative particle dynamics. *Phys. Chem. Chem. Phys.*, 17: 24452–24461, 2015.

[174] C. Phillips, J. Anderson, and S. Glotzer. Pseudo-random number generation for Brownian Dynamics and Dissipative Particle Dynamics simulations on GPU devices. *J. Comput. Phys.*, 230(19):7191–7201, 2011.

[175] D. Pinho, T. Yaginuma, and R. Lima. A microfluidic device for partial cell separation and deformability assessment. *Biochip J.*, 7:367–374, 2013.

[176] D. Pinho, T. Yaginuma, and R. Lima. A microfluidic device for partial cell separation and deformability assessment. *Biochip J.*, 7:367–374, 2013.

[177] J. W. Pitera, I. Haque, and W. C. Swope. Absence of reptation in the high-temperature folding of the trpzip2 β-hairpin peptide. *J. Chem. Phys.*, 124: 141102, 2006.

[178] I. V. Pivkin and G. E. Karniadakis. A new method to impose no-slip boundary conditions in dissipative particle dynamics. *J. Comput. Phys.*, 207(1): 114–128, 2005.

[179] I. V. Pivkin and G. E. Karniadakis. Controlling density fluctuations in wall-bounded dissipative particle dynamics systems. *Phys. Rev. Lett.*, 96: 206001, May 2006.

[180] I. V. Pivkin and G. E. Karniadakis. Accurate coarse-grained modeling of red blood cells. *Phys. Rev. Lett.*, 101(11):1–4, 2008.

[181] I. V. Pivkin, Z. Peng, G. E. Karniadakis, P. A. Buffet, M. Dao, and S. Suresh. Biomechanics of red blood cells in human spleen and consequences for physiology and disease. *Proceedings of the National Academy of Sciences*, 113(28):7804–7809, 2016.

[182] K. W. Plaxco, K. T. Simons, and D. Baker. Contact order, transition state placement and the refolding rates of single domain proteins. *J. Mol. Biol.*, 277:985–994, 1998.

[183] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1–19, 1995.

[184] R. Potestio, P. Espanol, R. Delago-Buscalioni, R. Everaers, K. Kremer, and D. Donadio. Hamiltonian adaptive resolution simulation for molecular liquids. *Phys. Rev. Lett.*, 111:060601, 2013.

[185] R. Potestio, S. Fritsch, P. Español, R. Delgado-Buscalioni, K. Kremer, R. Everaers, and D. Donadio. Hamiltonian adaptive resolution simulation for molecular liquids. *Phys. Rev. Lett.*, 110:108301, Mar 2013.

[186] A. Pries, T. Secomb, and P. Gaehtgens. Biophysical aspects of blood flow in the microvasculature. *Cardiovascular Research*, 32:654–667, 1996.

[187] A. R. Pries and T. W. Secomb. Microvascular blood viscosity in vivo and the endothelial surface layer. *Am. J. Physiol. Heart Circ. Physiol.*, 289: 2657–2664, 2005.

[188] A. R. Pries, K. Ley, M. Claassen, and P. Gaehtgens. Red cell distribution at microvascular bifurcations. *Microvasc. Res.*, 38:81–101, 1989.

[189] D. J. Quinn, I. Pivkin, S. Y. Wong, K.-H. Chiam, M. Dao, G. E. Karniadakis, and S. Suresh. Combined simulation and experimental study of large deformation of red blood cells in microfluidic systems. *Annals of Biomedical Engineering*, 39(3):1041–1050, 2011.

[190] A. Rahimian, I. Lashuk, S. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, D. Zorin, and G. Biros. Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures. In *Proc. of 2010 Intl. Conf. for High Perf. Computing, Networking, Storage and Analysis*, SC '10, pages 1–11, Washington, DC, USA, 2010. IEEE Computer Society.

[191] F. C. Ramaekers and F. T. Bosman. The cytoskeleton and disease. *Journal of Pathology*, 204(4):351–354, 2004.

[192] S. Riniker and W. F. van Gunsteren. Definition and testing of the gromos force-field versions 54a7 and 54b7. *J. Chem. Phys.*, 134:084110, 2011.

[193] S. Riniker, A. P. Eichenberger, and W. F. van Gunsteren. Solvating atomic level fine-grained proteins in supra-molecular level coarse-grained water for molecular dynamics simulations. *Eur. Biophys. J.*, 41:647–661, 2012.

[194] H. J. Risselada, S. J. Marrink, and M. Mueller. Curvature-dependent elastic properties of liquid-ordered domains result in inverted domain sorting on uniaxially compressed vesicles. *Phys. Rev. Lett.*, 106:148102, 2011.

[195] W. Ronan, V. S. Deshpande, R. M. McMeeking, and J. P. McGarry. Numerical investigation of the active role of the actin cytoskeleton in the compression resistance of cells. *Journal of the Mechanical Behavior of Biomedical Materials*, 14:143–157, 2012.

[196] G. D. Rose, P. J. Fleming, J. R. Banavar, and A. Maritan. A backbone-based theory of protein folding. *Proceedings of the National Academy of Sciences*, 103:16623–16633, 2006.

[197] D. Rossinelli, Y.-H. Tang, K. Lykov, D. Alexeev, M. Bernaschi, P. Hadjidoukas, M. Bisson, W. Joubert, C. Conti, G. Karniadakis, M. Fatica, I. Pivkin, and P. Koumoutsakos. The in-silico lab-on-a-chip: Petascale and high-throughput simulations of microfluidics at cell resolution. In *Proc. of*

2015 Intl. Conf. for High Perf. Computing, Networking, Storage and Analysis, SC '15, pages 2:1–2:12, New York, NY, USA, 2015. ACM.

[198] A. Rowat, J. Lammerding, and J. Ipsen. Mechanical properties of the cell nucleus and the effect of emerin deficiency. *Biophysical Journal*, 91(12): 4649 – 4664, 2006.

[199] A. L. Rynearson and C. R. Sussman. Nuclear structure, organization, and oncogenesis. *Journal of Gastrointestinal Cancer*, 42(2):112–117, 2011.

[200] A. J. Rzepiela, M. Louhivuori, C. Peter, and S. J. Marrink. Hybrid simulations: combining atomistic and coarse-grained force fields using virtual sites. *Phys. Chem. Chem. Phys.*, 13:10437–10448, 2011.

[201] C. Sagui and T. Darden. Multigrid methods for classical molecular dynamics simulations of biomolecules. *The Journal of Chemical Physics*, 114(15): 6578–6591, 2001.

[202] H. Saint-Martin, J. Hernandez-Cobos, M. I. Bernal-Uruchurtu, I. Ortega-Blake, and H. J. C. Berendsen. Study of the stabilization energies of halide-water clusters: An application of first-principles interaction potentials based on a polarizable and flexible model. *J. Chem. Phys.*, 113:10899, 2000.

[203] G. Salbreux, G. Charras, and E. Paluch. Actin cortex mechanics and cellular morphogenesis. *Trends in Cell Biology*, 22(10):536–545, 2012.

[204] M. Sato, D. Theret, L. Wheeler, N. Ohshima, and R. Nerem. Application of the micropipette technique to the measurement of cultured porcine aortic endothelial cell viscoelastic properties. *ASME. J Biomech Eng.*, 112(3): 263–268, 1990.

[205] J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri. The imagej ecosystem: An open platform for biomedical image analysis. *Molecular Reproduction and Development*, 82(7-8):518–529, 2015.

[206] T. Schlick. Time-trimming tricks for dynamic simulations: splitting force updates to reduce computational work. *Structure*, 9(4), 2001.

[207] J. Shaw Bagnall, S. Byun, S. Begum, D. T. Miyamoto, V. C. Hecht, S. Maheswaran, S. L. Stott, M. Toner, R. O. Hynes, and S. R. Manalis. Deformability of Tumor Cells versus Blood Cells. *Scientific reports*, 5(November): 18542, 2015.

[208] J.-E. Shea and C. L. Brooks III. From folding theories to folding proteins: a review and assessment of simulation studies of protein folding and unfolding. *Annu. Rev. Phys. Chem.*, 52:499–535, 2001.

[209] J. M. Sherwood, E. Kaliviotis, J. Dusting, and S. Balabani. Hematocrit, viscosity and velocity distributions of aggregating and non-aggregating blood in a bifurcating microchannel. *Biomech. Model. Mechanobiol.*, 13: 259–273, 2014.

[210] Q. Shi, S. Izvekov, and G. A. Voth. Mixed atomistic and coarse-grained molecular dynamics: simulation of a membrane-bound ion channel. *J. Phys. Chem. B*, 110:15045–15048, 2006.

[211] E. Spiga, D. Alemani, M. Degiacomi, M. Cascella, and M. D. Peraro. Electrostatic-consistent coarse-grained potentials for molecular simulations of proteins. *Journal of chemical theory and computation*, 9(8):3515–3526, 2013.

[212] T. Squires and S. Quake. Microfluidics: Fluid physics at the nanoliter scale. *Rev. Mod. Phys.*, 77:977–1026, Oct 2005.

[213] T. Steiner, C. Cupelli, R. Zengerle, and M. Santer. Simulation of advanced microfluidic systems with dissipative particle dynamics. *Microfluid Nanofluidics*, 7(3):307–323, 2009.

[214] A. D. Stephens, E. J. Banigan, S. A. Adam, R. D. Goldman, , and J. F. Marko. Chromatin and lamin A determine two different mechanical response regimes of the cell nucleus. *Mol. Biol. Cell*, 28(2):1–29, 2017.

[215] A. Streets and Y. Huang. Chip in a lab: Microfluidics for next generation life science research. *Biomicrofluidics*, 7(1):011302, 2013.

[216] Y. Sun, D.-H. Kim, and C. A. Simmons. *Integrative Mechanobiology: Micro- and Nano-Techniques in Cell Mechanobiology*. Cambridge University Press, 2015.

[217] S. Suresh. Biomechanics and biophysics of cancer cells. *Acta Biomaterialia*, 3(4):413–438, 2007.

[218] S. Takada. Go-ing for the prediction of protein folding mechanisms. *Proceedings of the National Academy of Sciences*, 96:11698–11700, 1999.

[219] N. Takeishi, Y. Imai, K. Nakaaki, T. Yamaguchi, and T. Ishikawa. Leukocyte margination at arteriole shear rate. *Physiological reports*, 2(6):1–8, 2014.

[220] N. Takeishi, Y. Imai, T. Yamaguchi, and T. Ishikawa. Flow of a circulating tumor cell and red blood cells in microvessels. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 92(6):1–6, 2015.

[221] N. Takeishi, Y. Imai, S. Ishida, T. Omori, R. D. Kamm, and T. Ishikawa. Cell adhesion during bullet motion in capillaries. *American Journal of Physiology - Heart and Circulatory Physiology*, 311(2):H395–H403, 2016.

[222] S. Tan, L. Yobas, G. Lee, C. Ong, and C. Lim. Microdevice for the isolation and enumeration of cancer cells from blood. *Biomedical Microdevices*, 11 (4):883–892, 2009.

[223] Y. Tang and G. E. Karniadakis. Accelerating dissipative particle dynamics simulations on gpus: Algorithms, numerics and applications. *Comput. Phys. Commun.*, 185(11):2809–2822, 2014.

[224] Y.-H. Tang, L. Lu, H. Li, C. Evangelinos, L. Grinberg, V. Sachdeva, and G. E. Karniadakis. Openrbc: A fast simulator of red blood cells at protein resolution. *Biophysical Journal*, 112(10):2030 – 2037, 2017.

[225] T. Terakawa and S. Takada. Respac: method to determine partial charges in coarse-grained protein model and its application to dna-binding proteins. *J. Chem. Theory Comput.*, 10:711–721, 2014.

[226] D. Theret, M. Levesque, M. Sato, R. Nerem, and L. Wheeler. The application of a homogeneous half-space model in the analysis of endothelial cell micropipette measurements. *ASME. J Biomech Eng.*, 110(3):190–199, 1988.

[227] V. Tozzini. Coarse-grained models for proteins. *Curr. Opin. Struct. Biol.*, 15:144–150, 2005.

[228] H. Turlier, D. A. Fedosov, B. Audoly, T. Auth, N. S. Gov, C. Sykes, J.-F. Joanny, G. Gompper, and T. Betz. Equilibrium physics breakdown reveals the active nature of red blood cell flickering. *Nature Physics*, 12(5):513–519, 2016.

[229] Y. Ujihara, M. Nakamura, H. Miyazaki, and S. Wada. Proposed spring network cell model based on a minimum energy concept. *Annals of Biomedical Engineering*, 38(4):1530–1538, 2010.

[230] Y. Ujihara, M. Nakamura, H. Miyazaki, and S. Wada. Contribution of actin filaments to the global compressive properties of fibroblasts. *Journal of the Mechanical Behavior of Biomedical Materials*, 14:192–198, 2012.

[231] Y. Ujihara, M. Nakamura, M. Soga, K. Koshiyama, H. Miyazaki, and S. Wada. Computational studies on strain transmission from a collagen gel construct to a cell and its internal cytoskeletal filaments. *Computers in Biology and Medicine*, 56:20–29, 2015.

[232] M. Verderame, D. Alcorta, M. Egnor, K. Smith, and R. Pollack. Cytoskeletal F-actin patterns quantitated with fluorescein isothiocyanate-phalloidin in normal and transformed cells. *Proceedings of the National Academy of Sciences*, 77(11):6624–6628, 1980.

[233] R. Vernekar and T. Krüger. Breakdown of deterministic lateral displacement efficiency for non-dilute suspensions: A numerical study. *Medical Engineering & Physics*, 37(9):845–854, 2015.

[234] A. Vishnyakov, D. S. Talaga, and A. V. Neimark. Dpd simulation of protein conformations: From α-helices to β-structures. *J. Phys. Chem. Lett.*, 3: 3081–3087, 2012.

[235] D. Visser, H. Hoefsloot, and P. Iedema. Comprehensive boundary method for solid walls in dissipative particle dynamics. *J. Comput. Phys.*, 205(2): 626–639, 2005.

[236] N. Wagner. The logistic lattice in random number generation. In *Proc. of the Annual Conf. on Communication Control and Computing*, volume 30, pages 922–922. University of Illinois, 1992.

[237] H. Wang, C. Junghans, and K. Kremer. Comparative atomistic and coarse-grained study of water: What do we lose by coarse-graining? *The European Physical Journal E: Soft Matter and Biological Physics*, 28(2):221–229, 2009.

[238] T. A. Wassenaar, K. Pluhackova, R. A. Boeckmann, S. J. Marrink, and D. P. Tielemann. Going backward: a flexible geometric approach to reverse transformation from coarse grained to atomistic models. *J. Chem. Theory Comput.*, 10:676–690, 2014.

[239] D. J. Wheeler and R. M. Needham. Tea, a tiny encryption algorithm. In *Fast Software Encryption*, pages 363–366. Springer, 1995.

[240] G. Whitesides. The origins and the future of microfluidics. *Nature*, 442 (7101):368–372, 2006.

[241] D. J. Wilton, R. B. Tunnicliffe, Y. O. Kamatari, K. Akasaka, and M. P. Williamson. Pressure-induced changes in the solution structure of the gb1 domain of protein g. *Proteins: Structure, Function, and Bioinformatics*, 71 (3):1432–1440, 2008.

[242] C. Wu and J.-E. Shea. Coarse-grained models for protein aggregation. *Curr. Opin. Struct. Biol.*, 21:209–220, 2011.

[243] T. Wu and J. J. Feng. Simulation of malaria-infected red blood cells in microfluidic channels: Passage and blockage. *Biomicrofluidics*, 7(4), 2013.

[244] Z. Wu, Q. Cui, and A. Yethiraj. A new coarse-grained model for water: the importance of electrostatic interactions. *J. Phys. Chem. B*, 114:10524–10529, 2010.

[245] Z. Wu, Z. Xu, O. Kim, and M. Alber. Three-dimensional multi-scale model of deformable platelets adhesion to vessel wall in blood flow. *Phil. Trans. R. Soc. A*, 372(2021), 2014.

[246] L. L. Xiao, Y. Liu, S. Chen, and B. M. Fu. Numerical simulation of a single cell passing through a narrow slit. *Biomechanics and Modeling in Mechanobiology*, 15(6):1655–1667, Dec 2016.

[247] L. L. Xiao, Y. Liu, S. Chen, and B. M. Fu. Effects of flowing rbcs on adhesion of a circulating tumor cell in microvessels. *Biomechanics and Modeling in Mechanobiology*, 16(2):597–610, Apr 2017.

[248] D. Xu, E. Kaliviotis, A. Munjiza, E. Avital, C. Ji, and J. Williams. Large scale simulation of red blood cell aggregation in shear flows. *Journal of Biomechanics*, 46(11):1810 – 1817, 2013.

[249] S. Yang, A. Undar, and J. Zahn. A microfluidic device for continuous, real time blood plasma separation. *Lab Chip*, 6:871–880, 2006.

[250] W. Y. Yang, J. W. Pitera, W. C. Swope, and M. Gruebele. Heterogeneous folding of the trpzip hairpin: Full atom simulation and experiment. *J. Mol. Biol.*, 336:241–251, 2004.

[251] T. Ye, N. Phan-Thien, and C. T. Lim. Particle-based simulations of red blood cells - A review. *Journal of Biomechanics*, 49(11):2255–2266, 2016.

[252] T. Ye, N. Phan-Thien, C. T. Lim, L. Peng, and H. Shi. Hybrid smoothed dissipative particle dynamics and immersed boundary method for simulation of red blood cells in flows. *Physical Review E*, 95(6):063314, 2017.

[253] S. Yesylevskyy, L. Schäfer, D. Sengupta, and S. Marrink. Polarizable water model for the coarse-grained martini force field. *PLoS Computational Biology*, 6(6):1–17, 2010.

[254] C. Yung, J. Fiering, A. Mueller, and D. Ingber. Micromagnetic-microfluidic blood cleansing device. *Lab Chip*, 9:1171–1177, 2009.

[255] Y. Zeng, A. Kia, and Y. S.-k. Teo. A three-dimensional random network model of the cytoskeleton and its role in mechanotransduction and nucleus deformation. *Biomech ModelMechanobiol*, (11):49–59, 2012.

[256] P. Zhang, N. Zhang, Y. Deng, and D. Bluestein. A multiple time stepping algorithm for efficient multiscale modeling of platelets flowing in blood plasma. *J. Comput. Phys.*, 284:668–686, 2015.

[257] Z. Zhang, J. Xu, B. Hong, and X. Chen. The effects of 3D channel geometry on CTC passing pressure - towards deformability-based cancer cell separation. *Lab on a chip*, 14:2576–84, 2014.

[258] Z. Zhang, X. Chen, and J. Xu. Entry effects of droplet in a micro confinement: Implications for deformation-based circulating tumor cell microfiltration. *Biomicrofluidics*, 9(2):024108, 2015.

[259] R. Zhou. Trp-cage: Folding free energy landscape in explicit water. *Proceedings of the National Academy of Sciences*, 100:13280–13285, 2003.

[260] R. Zhou, B. J. Berne, and R. Germain. The free energy landscape for beta hairpin folding in explicit water. *Proceedings of the National Academy of Sciences*, 98:14931–14936, 2001.