

# **Stony Brook University**



OFFICIAL COPY

**The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.**

**© All Rights Reserved by Author.**

# **Design and Analysis of Parallel Algorithms for Multiscale Modeling of Platelets**

A Dissertation Presented

by

**Na Zhang**

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

**Doctor of Philosophy**

in

**Applied Mathematics and Statistics**

**(Computational Applied Mathematics)**

Stony Brook University

**December 2015**

**Stony Brook University**

The Graduate School

**Na Zhang**

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

**Yuefan Deng – Dissertation Advisor**

**Professor, Department of Applied Mathematics and Statistics, Stony Brook University**

**James Glimm – Chairman of Defense**

**Distinguished Professor, Department of Applied Mathematics and Statistics, Stony Brook University**

**Robert Harrison – Committee Member**

**Professor, Department of Applied Mathematics and Statistics, Stony Brook University**

**Danny Bluestein – Outside Committee Member**

**Professor, Department of Biomedical Engineering, Stony Brook University**

This dissertation is accepted by the Graduate School

Charles Taber

Dean of the Graduate School

Abstract of the Dissertation

**Design and Analysis of Parallel Algorithms for Multiscale Modeling of Platelets**

by

**Na Zhang**

**Doctor of Philosophy**

in

**Applied Mathematics and Statistics**

**(Computational Applied Mathematics)**

Stony Brook University

**2015**

This work presents multiscale models and efficient numerical algorithms for analyzing the activation mechanisms of platelets under blood flow conditions at disparate spatiotemporal scales on supercomputers, with applications in initial thrombogenicity study and medical device optimization. Modeling the multiscale structures of platelets and the dynamics of their motion in viscous blood plasma require multiple time stepping (MTS) algorithm to optimally utilize the computing resources. This MTS algorithm improves the computational efficiency while maintaining stability and prescribed precisions. Our study of the dynamic properties of flipping platelets adapts the hybridized dissipative particle dynamics and coarse-grained molecular dynamics methods, which resolve the appropriate spatial scales of the platelet and the blood flow, respectively. In addition to the algorithmic strategies, general-purpose graphics processing units are also introduced to speed up the computationally intensive force field evaluations. Examinations of the implementation of the double-punch speedup strategy, i.e., algorithmic MTS and hardware acceleration, reveal significant speedups over single time stepping algorithms and CPU-only solutions. Detailed performance analysis on three representative supercomputers affords the possibility of simulating the millisecond-scale hematology at resolutions of nanoscale platelet and mesoscale bio-flow using millions of particles, the state-of-the-art for the field at the present time.

This dissertation is dedicated to my parents.

# Table of Contents

<b>List of Figures</b> .....	<b>vii</b>
<b>List of Tables</b> .....	<b>x</b>
<b>Acknowledgments</b> .....	<b>xi</b>
<b>List of Publications and Major Presentations</b> .....	<b>xii</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
1.1 Motivation .....	1
1.2 Contributions .....	3
1.3 Organization .....	5
<b>Chapter 2 Multiscale Modeling and Simulation Methods</b> .....	<b>7</b>
2.1 Introduction .....	7
2.2 Platelets Flipping in Viscous Flows .....	10
2.2.1 Viscous Flow Model.....	10
2.2.2 Deformable Platelet Model.....	11
2.2.3 Spatial Interfacing.....	12
2.3 Multiple Time Stepping Algorithm.....	14
2.3.1 MTS for DPD .....	15
2.3.2 MTS for CGMD .....	17
2.3.3 Temporal Interfacing .....	19
2.4 Accuracy versus Speed.....	22
2.4.1 Measures of Accuracy .....	23
2.4.2 Measures of Speed.....	24
2.5 Results and Analysis .....	25
2.5.1 Analysis of Accuracy.....	26
2.5.2 Analysis of Speed .....	33
2.6 Summary .....	38
<b>Chapter 3 Discovery of Force Field</b> .....	<b>39</b>
3.1 Introduction .....	39
3.2 Simulation .....	41

3.3	Parameterization.....	43
3.4	Results and Analysis .....	47
3.4.1	Green Kubo Autocorrelation .....	48
3.4.2	Radial Distribution Function .....	49
3.4.3	Counter-Poiseuille and Couette flows .....	49
3.4.4	The Impact of System Sizes .....	52
3.5	Application of Modified Morse Potential .....	55
3.6	Summary .....	56
<b>Chapter 4 Techniques for Accelerating Simulations.....</b>		<b>58</b>
4.1	Demand of Tasks.....	58
4.2	Supply of Supercomputer Resources .....	60
4.3	Mapping and GPGPU Acceleration .....	62
4.4	Performance Examinations .....	64
4.4.1	The Impact of MTS and GPGPU Acceleration .....	66
4.4.2	The Impact of System Architectures .....	73
4.5	Summary .....	85
<b>Chapter 5 Conclusions and Future Work .....</b>		<b>87</b>
<b>Bibliography .....</b>		<b>89</b>

## List of Figures

Figure 2-1: The wall-driven Couette flow and the initial position of the platelet model .....	12
Figure 2-2: Multiple methods: DPD and CGMD in the multiscale approach. ....	14
Figure 2-3: Multiple spatiotemporal scales in one model.....	15
Figure 2-4: Multiple time step sizes in the MTS algorithm.....	20
Figure 2-5: Normalized deviations for temperature $T$ of the system $\varepsilon(T, t)$ over time $t$ .....	27
Figure 2-6: Normalized deviations for pressure $P$ of the system $\varepsilon(P, t)$ over time $t$ .....	27
Figure 2-7: Normalized deviations for total energy of the system $\varepsilon(E_{tot}, t)$ over time $t$ .....	28
Figure 2-8: Normalized deviations for kinetic energy of single platelet $\varepsilon((k_B T)_{platelet}, t)$ over time $t$ .....	28
Figure 2-9: Normalized deviations for stress distribution on the membrane $\varepsilon(\tau, t)$ over time $t$ ...	29
Figure 2-10: Evolution of RMSD for stress distribution on the membrane over time $t$ .....	29
Figure 2-11: The dynamic stress distributions on the surface membrane using STS and MTS ...	30
Figure 2-12: Change of rotational angles of the platelet using MTS-L, comparing with the Jeffery's orbit (analytical).....	30
Figure 2-13: Parallel speeds $\mathbb{S}(P)$ vs. various processor cores $p$ for the non-MTS and MTS solvers (A base 10 logarithmic scale for the vertical axis) .....	33
Figure 2-14: Speedups of MTS algorithms over the STS algorithm (A base 10 logarithmic scale for the vertical axis) .....	34
Figure 2-15: The parallel efficiencies $\mathbb{E}(10, p_2)$ versus various processor cores $p_2$ for the non-MTS and MTS solvers .....	34
Figure 2-16: Percentiles of computation (Compt.) and communication (Comm.) over the total running time vs. the number of cores for the STS and MTS-L integrators .....	35
Figure 2-17: Percentile error and wallclock time (in days) for 1- <i>ms</i> simulation vs. different scales of step sizes for DPD in which the CGMD is integrated at $10^{-7}$ .....	37
Figure 2-18: Percentile error and wallclock time (in days) for 1- <i>ms</i> simulation vs. different scales of step sizes for CGMD in which the DPD is integrated at $10^{-3}$ .....	37
Figure 3-1: The relationship of the average distances over mass scales: $M_{CG}$ ( <i>amu</i> ) vs. $\mu$ ( $\text{\AA}$ ) where $x$ -axis is in logarithm scale of base 10. ....	41



Figure 3-2: Impact of $\varepsilon$ in classical Morse potential on normalized compressibility $\kappa_T$ and viscosity $\eta$ where $\alpha=10$ and $M_{CG} = 72.06 \text{ amu}$ .....	45
Figure 3-3: Impact of $\varepsilon$ in classical Morse potential on normalized compressibility $\kappa_T$ and viscosity $\eta$ where $\alpha=10$ and $M_{CG} = 720.62 \text{ amu}$ .....	45
Figure 3-4: The form factor $R(\mu)$ in modified Morse potential for $M_{CG} = 720.62 \text{ amu}$ .....	47
Figure 3-5: Pressure $P$ (in <i>bar</i> ) with the increase of $\mu$ (in $\text{\AA}$ ) in modified Morse potential for $M_{CG} = 720.62 \text{ amu}$ .....	47
Figure 3-6: The form factor $R(\mu)$ in modified Morse potential for $M_{CG} = 7206.20 \text{ amu}$ .....	47
Figure 3-7: Pressure $P$ (in <i>bar</i> ) with the increase of $\mu$ (in $\text{\AA}$ ) in modified Morse potential for $M_{CG} = 7206.20 \text{ amu}$ .....	47
Figure 3-8: Auto stress correlation functions $C_{\alpha\beta}(t)$ vs. time $t$ (in <i>ps</i> ) .....	48
Figure 3-9: Radial distribution functions of the Morse fluids $g(r)$ vs. $r$ .....	49
Figure 3-10: Schematic representation of the periodic Poiseuille flow .....	51
Figure 3-11: Schematic representation of the Couette flow .....	51
Figure 3-12: Counter Poiseuille flows velocity profiles .....	51
Figure 3-13: Couette flows velocity profiles .....	52
Figure 3-14: Multiscale model of human platelets: physical structures and constituents .....	56
Figure 4-1: Multiscale modeling framework for simulating platelets in blood flows .....	60
Figure 4-2: Parallel speeds of STS simulations .....	66
Figure 4-3: Parallel speeds of MTS cases for Exp-M.....	68
Figure 4-4: Parallel speeds of MTS cases for Exp-L.....	68
Figure 4-5: Speedups of MTS cases over STS case for Exp-M .....	69
Figure 4-6: Speedups of MTS cases over STS cases for Exp-L.....	69
Figure 4-7: The ratios of communication time over computation time for all MTS cases and STS for Exp-M.....	70
Figure 4-8. The ratios of communication time over computation time for all MTS cases and STS for Exp-L.....	70
Figure 4-9: Parallel speeds of GPU-enabled MTS cases for Exp-M .....	71
Figure 4-10: Speedups GPU over CPU-only for different MTS cases for Exp-M.....	71
Figure 4-11: Parallel speeds of no_mts/mts algorithms on Tianhe-2 .....	74
Figure 4-12: Performance improvement of mts over no_mts on Tianhe-2 .....	75

Figure 4-13: Ratio of communication over computation on Tianhe2.....	75
Figure 4-14: Parallel speeds of no_mts/mts algorithms on Stampede .....	76
Figure 4-15: Performance improvement of mts over no_mts on Stampede .....	76
Figure 4-16: Ratio of communication over computation on Stampede .....	77
Figure 4-17: Parallel speeds of no_mts/mts algorithms on 8-K40m CS-Storm .....	77
Figure 4-18: Performance improvement of mts over no_mts on 8-K40m CS-Storm .....	78
Figure 4-19: Ratio of communication over computation on 8-K40m CS-Storm .....	78
Figure 4-20: Parallel speeds of no_mts/mts algorithms on 16-K80 CS-Storm .....	79
Figure 4-21: Performance improvement of mts over no_mts on 16-K80 CS-Storm .....	79
Figure 4-22: Ratio of communication over computation on 16-K80 CS-Storm .....	80
Figure 4-23: Parallel speeds of no_mts and mts algorithms for Exp-L on Tianhe-2 and Stampede .....	80
Figure 4-24: Ratio of communication over computation for no_mts and mts algorithms for Exp-L on Tianhe-2 and Stampede.....	81

## List of Tables

Table 2-1: The multiple time-stepping algorithm within a 2-level integrator in DPD .....	17
Table 2-2: The multiple time-stepping algorithm within a 2-level integrator in CGMD .....	19
Table 2-3: Overview of the multiple time-stepping algorithms for the multiscale model.....	21
Table 2-4: The time steps and configurations for different test cases .....	22
Table 2-5: Accuracy and speedup comparisons for different MTS parameters .....	32
Table 3-1: Mechanical properties of blood plasma fluid .....	42
Table 3-2: Impact of model parameters on target properties .....	43
Table 3-3: Parameters of the modified Morse potential for various mass scales .....	47
Table 3-4: Characteristic time of different CG levels: $M_{CG}$ , $\mu$ (average distance, $\text{\AA}$ ), $dt$ (timestep, $fs$ ) and $\tau$ (characteristic time, $ps$ ) .....	48
Table 3-5: Results and analyses of the same simulation box using three coarse graining levels .	53
Table 3-6: Comparisons of physical properties using different simulation boxes for $M_{CG} = 72.06 amu$ .....	54
Table 3-7: Comparisons of physical properties using different simulation boxes for $M_{CG} = 720.62 amu$ .....	54
Table 4-1: Problem sizes and dimensions of multiscale benchmarks.....	59
Table 4-2: Nodal configurations and peak performance ( $GFlops$ ) for Tianhe-2, Stampede and CS-Storm.....	62
Table 4-3: Mappings of each system size .....	62
Table 4-4: The timesteps and configurations for each MTS test case .....	65
Table 4-5: Best performances for STS and MTS algorithms on three supercomputers for Exp-S .....	81
Table 4-6: Best performances for STS and MTS algorithms on three supercomputers for Exp-M .....	82
Table 4-7: Best performances for STS and MTS algorithms on three supercomputers for Exp-L .....	82
Table 4-8: Simulation speeds ( $\mu s/day$ ) for multiscale model and MTS algorithm on Tianhe-2, Stampede and CS-Storm .....	83

## Acknowledgments

First of all, I would like to express my deepest gratitude to my advisor, Professor Yuefan Deng, for his four years' guidance, support, encouragement and provision that benefited me much in the completion of this PhD study at Stony Brook University. Thank him so much for guiding me to this great high performance computing research field. I am also very thankful to him for providing me access to large amount of supercomputing resources and for excellent opportunities to visit scientific conferences.

I would like to thank Professor Danny Bluestein for his guidance in developing multiscale application in biomedical engineering field and all the constructive comments to finish manuscripts and thank him for being on my dissertation committee. I am very grateful to Professor James Glimm and Professor Robert Harrison for being on my dissertation committee, especially Professor James Glimm, for being the chairman of my preliminary defense committee and dissertation defense committee. I thank them for reading through my research, and providing feedback and suggestions.

I would like to thank all present and past members in multiscale modeling group, Dr. Peng Zhang, Dr. Seetha Pothapragada, Dr. Jawaad Sheriff, Chao Gao and Li Zhang for all scientific discussions and private helps. I would like to give my special thanks to Dr. Peng Zhang for sharing his high performance computing expertise, having numerous discussions, writing collaborative publications with me, without whom my research would never so smooth.

I would like to express my love and thanks to my parents for their endless love and supports.

The text of this dissertation in part is a reprint of materials as it appears in following publication page. The co-authors listed in the publication page directed and supervised the research that forms the basis for this dissertation.

## List of Publications and Major Presentations

### Peer-Reviewed Articles:

- (1) N. Zhang, P. Zhang, W. Kang, D. Bluestein, and Y. Deng, "Parameterizing the Morse potential for coarse-grained modeling of blood plasma," *Journal of Computational Physics*, vol. 257, pp. 726-736, 01/2014.
- (2) P. Zhang, C. Gao, N. Zhang, M. J. Slepian, Y. Deng, and D. Bluestein, "Multiscale particle-based modeling of flowing platelets in blood plasma using dissipative particle dynamics and coarse grained molecular dynamics," *Cellular and Molecular Bioengineering*, vol. 7 pp. 552-574, 12/2014.
- (3) P. Zhang, N. Zhang, Y. Deng, and D. Bluestein, "A multiple time stepping algorithm for efficient multiscale modeling of platelets flowing in blood plasma," *Journal of Computational Physics*, vol. 284, pp. 668-686, 01/2015.
- (4) N. Zhang, P. Zhang, L. Zhang, X. Zhu, L. Huang, and Y. Deng, "Performance examinations of multiple time-stepping algorithms on Stampede supercomputer," *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, St. Louis, MO, 07/2015.

### Manuscript under Review:

- (1) P. Zhang, N. Zhang, Y. Deng, and D. Bluestein, "Scalability test of multiscale fluid-platelet model for three top supercomputers," *Computer Physics Communications*, 09/2015.

### Conference Presentations:

- (1) The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC15), "Efficient multiscale platelets modeling using supercomputers," Austin, TX, 11/2015.

- (2) 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure (XSEDE15), "Performance examinations of multiple time-stepping algorithms on Stampede supercomputer," St. Louis, MO, 07/2015.
- (3) The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC14), "A multiple time stepping algorithm for efficient multiscale modeling of platelets flowing in blood plasma," New Orleans, LA, 11/2014.
- (4) The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC13), "Design and analysis parallel multiscale algorithms for modeling platelets," Denver, CO, 11/2013.

# Chapter 1 Introduction

## 1.1 Motivation

Large-scale computational modeling is widely accepted as a method of choice for solving complex problems due to its great values in reducing time and monetary costs for experimental design, implementation and analysis. Examples range from human brain project [1], bimolecular simulations [2], weather forecasting [3], to astrophysical N-body simulations [4]. Although Moore's law together with new architectural and chip technological innovations boost the power of supercomputers, there is no foreseeable end to the need for ever larger and more powerful systems [5]. Demand for supercomputing resources greatly exceeds supply, and meanwhile, the gap between achieved performance from those resources and the raw performance of the fast systems is increasing. Essentially, how to efficiently utilize the resources for optimal performance becomes a serious challenge for algorithm and software design of large-scale simulations.

The motivation of this work is the integration of multiscale models, efficient algorithms and heterogeneous supercomputing for efficiently simulating platelets and their functions in thrombosis formation. Advances in medical imaging techniques and computational methods have enabled accurate simulations of subject-specific blood flows at the level of blood cell and in complex arterial networks [6-8]. Those advances in studying blood flow dynamics have in turn improved cardiovascular devices design, a process relying heavily on numerical simulations in recent years [9, 10]. While the advent of these devices has provided life-saving solutions to millions of patients globally, thrombogenic risk potential of these devices remains an impediment [11-14]. Thrombosis formation on the surface of these implantable devices is initialized by chain reactions of platelets disorders. Aggregating platelets release factors that promote accumulation of circulating protein called fibrin. A blood clot is a meshwork of platelets and red blood cells woven together by fibrin. Thrombosis is an abnormal localized activation of platelets. Among all pathways completing such process, abnormal shear stress at various exposure times or a sudden increase will affect platelet activation [12, 15-19]. However, quantitative analysis of the platelet dynamics including the shape change is, at best, confined in the laboratory experiments [20]. Successfully computer-modeling such intricate phenomena and

analyzing stress field upon platelets activation will help design drugs and implantable blood recirculating devices for fighting vascular diseases [9]. Nevertheless, this task is full of physiological, physical, mathematical and computational challenges.

First of all, physical insights are essential for developing accurate and predictive platelet-fluid models with parameters matching laboratory mechanical and thermodynamic blood properties. Secondly, efficient simulation methods require modeling not just the flow dynamics but process pertinent to flow-induced platelet-mediated blood clotting. As such, multiple scales must be considered to characterize system behavior and the associated modeling tools must be developed for high-resolution low-cost studies of complex biological problems.

Classical simulation methods are often limited in their applications to certain range of space and time scales. Four length and time scales of classical simulation methods are: (1) Scale of continuum models, such as computational fluid dynamics (CFD) in which conservations of mass, momentum and energy govern the fluid motion; (2) Scale of mesoscale models, such as dissipative particle dynamics (DPD) in which grouped particles are introduced to reduce computational complexity by integrating out the grouped particles' internal degrees of freedom; or lattice Boltzmann method (LBM) in which fluid consists of fictive particles, and such particles perform consecutive propagation and collision processes over a discrete lattice mesh; (3) Scale of molecular dynamics models, the trajectories of actual atoms governed by Newton's equations of motion are studied; (4) Scale of quantum mechanics models, quantum effects that produce averaged parameters are considered. For modeling platelet-related phenomena, one single scale will exhibit a serious dilemma: continuum models fail to capture small-scale molecular mechanisms of structure changes and interactions of key players in blood coagulation [21] while molecular dynamics that can capture the full dynamics at molecular scales, are too computationally intensive to be practical [22, 23]. Mesoscopic approaches provide a "coarse-grained" description of a simulated system compared with a "fine-grained" atomic description and keep some of molecular details [24]. They significantly expand the space and time scale with respect to molecular dynamics but their spatiotemporal representations are limited by coarse-graining levels [25]. A multiscale platelet modeling approach to uncouple macroscopic blood flow and finer features of platelet may offer a compromise for balancing computational accuracy and feasibility.



In addition to multiple scale algorithms, implementing them on a particular computing platform for high efficiency is another challenge. The large gap in the space and time scales and algorithm complexities are two major challenges in limiting computing speed. With the unprecedented advances in computer speeds, it's still not practical to simulate one-millisecond hematology phenomena by straightforward porting to supercomputers without massive tuning. Porting simulations to, or developing a new programming model for, novel computer architectures for optimal performance is another aspect of the dissertation work.

In summary, advances are needed to enable breakthroughs in multiscale modeling of platelets. These advances include:

- (1) Accurate mathematical models for each scale considered;
- (2) Stable spatial and temporal interfacing methods;
- (3) Robust algorithms capable of both extracting main features and of quantifying possible errors; and
- (4) Parallel solvers for each model and be scalable to tens of thousands of computer processors, and be portable to heterogeneous architectures.

## **1.2 Contributions**

The main focus of this dissertation is development of computational methodology that will efficiently simulate the millisecond-scale hematology with heterogeneous supercomputing. The main contributions and major results of this work are as follows:

- (1) Modified Morse potential for coarse-grained modeling of blood plasma

A coarse-grained (CG) particle model is developed for simulating blood flow by modifying the Morse potential, traditionally used for modeling vibrating structures. The modified Morse potential is parametrized with effective mass scales for reproducing blood viscous flow properties, including density, pressure viscosity, compressibility and characteristic flow dynamics. The parameterization follows a standard inverse-problem approach in which the optimal micro parameters are systematically searched, by gradually decoupling loosely

correlated parameter spaces, to match the macro physical quantities of viscous blood flow. The predictions of this particle-based multiscale model compare favorably to classic viscous flow solutions such as Counter-Poiseuille and Couette flows. It demonstrates that such coarse-grained particle model can be applied to replicate the dynamics of viscous blood flow, with the advantage of bridging the gap between macroscopic flow scales and the cellular scales characterizing blood flow that continuum based models fail to handle adequately. One of its applications is it can be used as a simulation method for modeling of platelet cytoplasm, with corroborations with in-vitro results for cytoplasmic biorheology. The highlight of this work is to decouple the parameter space and map the parameters to physical properties.

- (2) Multiple time stepping algorithm for efficient multiscale modeling of dynamics of platelet flipping in blood flow

A flexible multiple time stepping (MTS) algorithm with four-level integrators is developed. The four-level integrator can be adjusted via three loop factors to optimize precision and computing speed, resulting in huge performance improvement while maintaining stability and accuracy. The study of the dynamics of platelet flipping in blood flow employs multiscale modeling. In this multiscale model, three simulation domains are identified: platelet domain, blood flow domain and interfacing domain. For platelet domain, the microscopic method using coarse-grained molecular dynamics (CGMD) is employed to emulate components of platelet such as viscous cytoplasm, extensible cytoskeleton and bilayer elastic membrane. For blood flow domain, the mesoscopic method using dissipative particle dynamics (DPD) is employed to address the bulk transfer of viscous blood flows in arterials with adjustable flow Reynolds numbers. For interfacing domain, a hybrid force field containing the stochastic and random terms from DPD and Lenard-Jones potential from MD is exploited to mimic friction between platelet surface membrane and surrounding blood flow. The highlight of this work is the multi-level integration scheme, allowing an optimization procedure for seeking a faster performer within certain error boundaries.

- (3) Parallel implementation on GPGPU/multi-GPGPU system and detained performance investigations on different system architectures

Efforts have been made to port multiscale simulations to systems with accelerators. Localizing computations and porting intensive computation workloads to accelerators such as GPGPU help reduce the inter-process communication costs on CPU-only systems. Detailed performance examinations have been carried out on three supercomputers-Tianhe-2, Stampede and CS-Storm (a high-density multi-accelerator system) in order to investigate (a) impacts of double-punch acceleration strategy (algorithmic MTS and hardware GPGPU acceleration) to see how much improvement can be achieved by the combining acceleration techniques; (b) impacts of system architectures to see how to map different problem characteristics to architectures. The optimized performance results manifest the possibility of simulating the millisecond-scale hematology at resolutions of nanoscale platelets and mesoscale bio-flows using millions of particles.

### **1.3 Organization**

Chapter 2 describes multiscale modeling methodology and simulation methods for platelet and blood flow, focusing on:

- (1) The viscous flow model, deformable platelet model and the spatial interfacing method to study dynamics of platelet flipping in blood flow;
- (2) Multiple time stepping algorithm for speeding up above simulations;
- (3) Accuracy and computing speed analysis.

Chapter 3 presents the novel modified Morse potential for modeling blood plasma at multiple coarse-grained levels, focusing on:

- (1) Simulation and analytics of particle-based fluid system;
- (2) Parameterization procedure for modified Morse potential for reproducing blood viscous flow properties;
- (3) Verification of characteristic flow dynamics including Green Kubo (GK) autocorrelation, radial distribution function (RDF), velocity profiles of counter-Poiseuille and Couette flows;
- (4) Investigations of the impact of system sizes;

- (5) Application of modified Morse potential for modeling platelet cytoplasm and its corroboration with cytoplasmic biorheology.

Chapter 4 presents the detailed performance evaluations of running multiscale benchmark with CPU-only and GPGPU-accelerated solutions on three supercomputers, focusing on:

- (1) Description of supply of computing resources and demand of tasks;
- (2) Mapping and implementation of GPGPU-accelerated solutions;
- (3) Performance results regarding the double-punch speedup strategy, i.e., algorithmic MTS and GPGPU acceleration;
- (4) Performance results regarding different system architectures, i.e., Tianhe-2, Stampede, and Cray CS-Storm and guideline for mapping problem characteristics onto platform characteristics.

Chapter 5 provides the conclusion and delineates the projection for future work.

## Chapter 2 Multiscale Modeling and Simulation Methods

### 2.1 Introduction

The coagulation cascade of blood may be initiated by flow-induced platelet activation, which prompts clot formation in prosthetic cardiovascular devices and arterial disease processes [16, 26, 27]. While platelet activation may be induced by biochemical agonists, shear stresses arising from pathological flow patterns enhance the propensity of platelets to activate and initiate coagulation pathway, leading to thrombosis [17-19]. Quantitatively determining the illusive dynamics and mechanics of platelets in pathological cases can facilitate developing effective treatments [28] and elucidating the platelet initiate processes such as thrombus formation; platelets undergo complex biochemical and morphological transitions during activation, resulting in aggregation and adhesion to blood vessel to form thrombi. In addition to traditional laboratory experiments, numerical simulations augment investigation into understanding of the behavior of platelets at molecular scales [23]. Clearly, a useful model must encapsulate sufficient spatial and temporal details. Fedosov *et al.* [29] describe an elastic model for the membrane with an accurate representation of mechanic properties of red blood cells. Martinez et al. [30] conclude that total rigidity of cortex stiffening significantly influenced detachment forces of adherent platelets and cell-membrane internal stresses. These indicate that the membranous viscoelasticity may be a factor for modeling resulting in, unavoidably, more computational complexity.

While a complete all-atom molecular dynamics simulation of a biological system can capture the dynamics of platelets at molecular scales [23, 31], this is not practical due to the prohibitive computational resources required [23, 31, 32] as evident by two recent ACM Gordon Bell supercomputing performance records. The first conducted a simulation using 13 trillion grid points in 2013 [33] and the second performed gravitational N-body simulations with one trillion particles in 2012 [34]. These are infinitesimal compared to the atomic modeling of platelets flowing in the viscous fluid flows. A typical platelet could consist of more than 0.7 trillion atoms, roughly estimated by an average mean platelet volume of  $\sim 7.1 \times 10^{-15}$  L and the possible number of atoms contained in it as compared with a C12 atom with volume of  $1.0 \times 10^{-26}$  L [35]. Additionally, some atoms interact at diverse scales both in space and time. Thus, we must

develop algorithms to model the dynamics of these atoms at corresponding spatiotemporal scales [36-39].

A multiscale particle-based model is proposed to combine the DPD-CGMD methods for modeling the dynamics of platelets flowing in response to extracellular flow-induced stresses [38]. The model adapts the DPD method for describing macroscopic transport of blood plasma in vessels and the CGMD method for handling individual platelets. A hybrid force field is formulated for establishing a functional interface between the platelet membrane and the surrounding fluid, in which the microstructural changes of platelets may respond to extracellular flow-induced stresses transferred to them. The DPD and CGMD force fields are technically different particle-based coarse-graining methods: the coarse-graining stochastic dynamics and the coarse-graining molecular dynamics [40]. As DPD is the only CGSD method in the work, we use the DPD term for better clarity. First, CGMD forces are conservative [41] but DPD forces depend on relative velocities between particles [42, 43]. Second, the time scale of CGMD is of picoseconds to nanoseconds while that of high-frequency oscillations of covalent bonds is of femtoseconds. However, the time scale of DPD is of microseconds, much slower than those of the constituents in CGMD [44, 45]. Typically, standard time-stepping (STS) algorithms use a single step size which must be the smallest of all time scales involved [41, 46] to gain the needed accuracy at the expenses of significant loss of efficiency.

Admittedly, the STS algorithm can solve the model accurately; however, ignoring the temporal scale differential in a variety of integrators would result in a massive number of redundant computations leading to inefficient use of computational resources. To remedy such deficiency, a variety of multiple time-stepping (MTS) algorithms have been developed for CGMD and for DPD independently. For example, the reversible reference system propagator algorithm (r-RESPA) [47] is widely adopted as the CGMD integrator and it greatly accelerates simulations of systems with multiple times scales and long-ranged forces [47-49]. Second, DPD integrators use the modified velocity Verlet integrator derived from stochastic Trotter formula [43, 50, 51], departing from usual CGMD integrators. Following the modified velocity Verlet integrator, Symeonidis et al. [45] integrated the MTS scheme to hybrid DPD models for simulating dilute polymer solutions. It introduced relaxation parameters for optimization with which it gained a 10 folds of efficiency while keeping the accuracy. Jakobsen et al. [52] studied

several MTS update schemes for a coarse-grained model of a lipid bilayer in water and it reduced considerably the simulation time by optimizing two time scales separately for lipid and solvent within error brackets. MTS algorithms are essential for improving efficiencies of CGMD and DPD [42, 53].

Although multiscale models of cell and tissue dynamics are prevailing [29, 36, 54] and MTS algorithms are established at various scales [45, 47], the temporal coupling is by no means a trivial task. For instance, in the triple decker scheme [54], the time progression in each subdomain is independent and communication of boundary conditions such as velocity averaging is performed every certain time steps. The same exchange of boundary information is widely applied to multiscale simulations. Even with rapid advances in raw computing speeds and algorithms, these simulations are still computational expansive [55] and it is necessary to construct models that leverage on accurate understanding of the underlying biomechanics, for maximum utilization of computing resources [42, 44, 53].

Extending our previous efforts on a multiscale model of platelets flowing in viscous blood plasma flows, we develop an integrated MTS algorithm for performing multiscale simulations. We present the force fields and describe the formulation and properties of MTS algorithms for coupling the DPD and CGMD. In this integrated MTS algorithm, various parameters are introduced to guide the selection of step sizes for achieving performance optimization. To estimate the quality of the MTS algorithms, we introduce the accuracy vs. speed relationships as functions of step sizes. Extensive numerical experiments were conducted using performance metrics, to measure the relative performances of standard single-scaled time-stepping (STS) algorithm vs. ours.

This study is an integral part to a full multiscale particle-based model that studies flow-induced platelet-mediated thrombosis composed of: (i) the spatial interface of the top-scale and bottom-scale methods: DPD and CGMD, whereby the microscale model of platelets allows to continuously undergo microstructural changes in response to extracellular flow-induced stresses [38, 56] and (ii) the temporal interface of the top-scale and bottom-scale integrators in which the step sizes are appropriately specified as multiple scales in order to enhance the computational efficiency while keeping the solutions within predefined error boundaries.

## 2.2 Platelets Flipping in Viscous Flows

In our multiscale model, we consider two scales: (i) the macroscopic/top-scale using DPD for describing viscous fluid flows; and (ii) the microscopic/bottom-scale using CGMD (classical molecular dynamics with reduced degrees of freedom) for describing deformable platelets with internal constituents including membrane, cytoskeleton and cytoplasm. A functional interface hybridizes DPD and CGMD for continued shape changes of flowing platelets in response to extracellular flow-induced stresses.

### 2.2.1 Viscous Flow Model

DPD is employed to simulating human blood flows, in which each DPD particle embodies a cluster of atoms or molecules and their collective motion is governed by [24, 57]:

$$d\mathbf{v}_i = \frac{1}{m_i} \sum_{j \in i}^N (\mathbf{F}^C dt + \mathbf{F}^D dt + \mathbf{F}^R \sqrt{dt} + \mathbf{F}^E dt) \quad \text{Equation 2-1}$$

where  $\mathbf{F}^C = \alpha \omega^C(r_{ij}) \mathbf{e}_{ij}$      $\mathbf{F}^D = -\gamma \omega^D(r_{ij}) (\mathbf{e}_{ij} \cdot \mathbf{v}_{ij}) \mathbf{e}_{ij}$      $\mathbf{F}^R = \sigma \omega^R(r_{ij}) \zeta_{ij} \mathbf{e}_{ij}$

$\mathbf{F}^C$ ,  $\mathbf{F}^D$  and  $\mathbf{F}^R$  are the conservative, dissipative and random forces acting on the particle and  $\mathbf{F}^E$  is the external force exerted to each particle to lead the fluid flow.  $r_{ij}$  is the inter-particle distance,  $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$  is the relative velocity and  $\mathbf{e}_{ij}$  is a unit vector in the direction  $r_i - r_j$ .  $\zeta_{ij}$  is a Gaussian random variable with zero mean and unit variance. The appearance of the factor  $\sqrt{dt}$  in Equation 2-1 was discussed in [24].  $\alpha$  is the maximum inter-particle repulsion given by  $\alpha = 75k_B T / (\rho_f r_c)$  where  $\rho_f$  is the number of fluid particles. The weight function  $\omega^C$  is set to zero beyond the cutoff length  $r_c$  and is given by

$$\omega^C(r_{ij}) = \begin{cases} (1 - r_{ij}/r_c), & r_{ij} < r_c \\ 0, & r_{ij} \geq r_c \end{cases} \quad \text{Equation 2-2}$$

Español *et al.* [57] established a relation between  $\gamma$  and  $\sigma$  and weight functions given by  $\sigma^2 = 2\gamma k_B T$  and  $\omega^D = [\omega^R]^2$ . We construct the wall-driven Couette flow (Figure 2-1) in a  $16 \times 16 \times 8$  box (length  $\times$  height  $\times$  width in  $\mu\text{m}$ ) = 2,048  $\mu\text{m}^3$  in volume. The upper and lower vessel walls are moving in opposite directions at a rate of 30 cm/s. Periodic boundary condition [41] is imposed along the  $x$ -/ $z$ -dimension. The no-slip boundary condition between the flow and the vessel is imposed following the work [58]. This general no-slip condition consists of the



inclusion of fictitious particles beyond the vessel with reversed velocity to develop an equilibrated shear layer, naturally enforcing zero velocities at the wall plane [58, 59]. Following [43], we work with a number density of the fluid system  $\rho_f = 3$ , i.e., 4,236 particles/ $\mu\text{m}^3$ . For the flow domain, we have 8.73 million particles.

## 2.2.2 Deformable Platelet Model

CGMD is employed to model 3D deformable platelets [38] and structural details are shown in Figure 2-1. The platelet model comprises of microscale internal constituents including an elastic membrane, a supporting cytoskeleton structure and padding cytoplasm. The model contains 73,036 particles including 19,675 membrane particles (27%), 36,296 cytoskeleton particles (50%) and the remaining 17,065 cytoplasm particles (23%). The initial shape of platelet is an oblate spheroid with semi-major and semi-minor axes of 2.0 and 0.5  $\mu\text{m}$  leading to a total volume of 8.37  $\mu\text{m}^3$ . Thus, the number density of the model is 8,722 particles/ $\mu\text{m}^3$  and it is roughly twice that of the fluid system. A reduced molecular-scale force field is developed and it includes the bonded interactions (bonds and angles) and the nonbonded interactions (Lennard-Jones (L-J) potential) and given by:

$$V(r) = \sum_{\text{bonds}} k_b(r - r_0)^2 + \sum_{\text{angles}} k_\theta(\theta - \theta_0)^2 + \sum_{\substack{\text{nonbonded} \\ \text{pairs}}} U_{LJ}(r) \quad \text{Equation 2-3}$$

where

$$U_{LJ} = 4\varepsilon \left[ \left(\frac{\sigma}{r}\right)^6 - 2\left(\frac{\sigma}{r}\right)^{12} \right]$$

$V$  is the total energy. The first two terms on the right-hand side are the bond and angle components where  $k_b$  and  $k_\theta$  are the force constants while  $r_0$  and  $\theta_0$  are the equilibrium distance and angle. The last term is the nonbonded L-J potential where  $\varepsilon$  is the depth of potential well and  $\sigma$  is the characteristic distance at which the inter-particle potential vanishes. The bonded terms exist in membrane and cytoskeleton, and the nonbonded terms are present between any particle pairs within a radius of interaction. Young's modulus of the membrane model is  $(1.5 \pm 0.6) \times 10^3 \text{ dyn/cm}^2$  [38]. Young's modulus of human platelet membrane is reported as  $(1.7 \pm 0.6) \times 10^3 \text{ dyn/cm}^2$  in [60] (values are mean  $\pm$  standard deviation). For other functional structures, the force constants for cytoskeleton structure are selected strong enough to maintain rigidity and thus to provide support to the ellipsoidal shape of resting platelets. The L-J potential for cytoplasm is

selected to preserve the volume of the platelet. Advancing from previous rigid spheroid models [61-66], this model characterizes the proper deforming capability of the membrane and allows us to observe the responsive deformation of platelets and to investigate dynamic stress mapping on the surface membrane resulting from the fluid-platelet interaction.

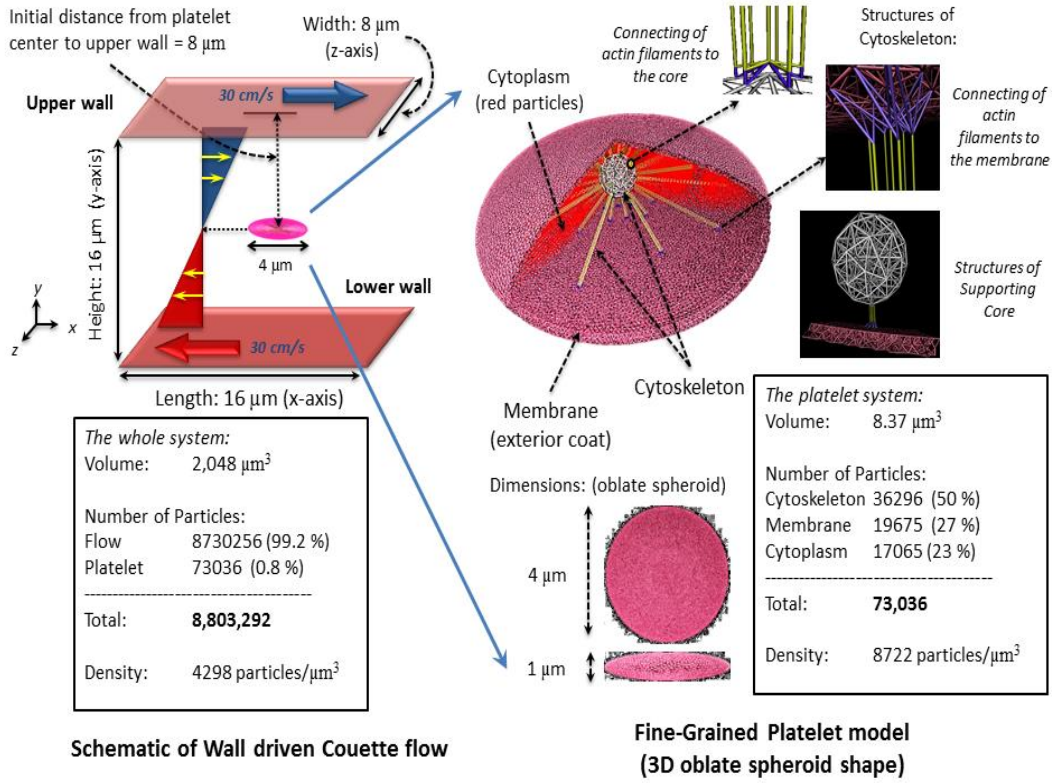


Figure 2-1: The wall-driven Couette flow and the initial position of the platelet model

### 2.2.3 Spatial Interfacing

The DPD-CGMD methods are interfaced on the surface membrane where a hybrid force field is established [38]. Figure 2-2 shows the schematics of the spatial-interfacing approach. The hybrid force field is defined as:

$$d\mathbf{v}_i = \frac{1}{m_i} \sum_{j \neq i}^N (-\nabla U_{LJ}(r_{ij}) dt + \mathbf{F}^D dt + \mathbf{F}^R \sqrt{dt}) \quad \text{Equation 2-4}$$

$$U(r_{ij}) = 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^6 - 2 \left( \frac{\sigma}{r_{ij}} \right)^{12} \right] \quad \mathbf{F}^D = -\gamma_p \omega^D(r_{ij}) (\mathbf{e}_{ij} \cdot \mathbf{v}_{ij}) \mathbf{e}_{ij} \quad \mathbf{F}_{ij}^R = \sigma_p \omega^R(r_{ij}) \zeta_{ij} \mathbf{e}_{ij}$$

$\varepsilon$  and  $\sigma$  are the characteristic energy and distance parameters in CGMD. Other parameters including  $\gamma_p$  and  $\sigma_p$  retain the same definitions as in DPD. All forces are truncated beyond a cutoff radius which defines the length scale in the fluid-platelet contact region [54]. The L-J force  $\nabla U(r_{ij})$  helps the cytoskeleton-confined shapes and the incompressibility of platelets against the applied stress of circumfluent plasma flow. The dissipative and random force terms maintain the flow local thermodynamic and mechanical properties and exchange momentum to express interactions between the platelet and the surrounding flows. A no-slip boundary condition was applied at the fluid-membrane surface interface. A dissipative or drag force was added to enforce the no-slip boundary condition at the fluid and membrane interface, so the fluid particles are dragged by dissipative forces of membrane particles as they are getting closer to the membrane, mimicking boundary layer mechanism where one layer drags its adjacent layers. The hard-core L-J force simultaneously provides a bounce-back reflection of fluid particles on the membrane (to prevent fluid particles from penetrating through the platelet membrane) with the no-slip achieved by slowing down the fluid particles (by the same repulsive term) as the fluid particles are getting closer to the membrane surface. The magnitude of the L-J force increases to infinity as the distance decreases, guaranteeing that the L-J force be strong enough to slow down and bounce back fluid particles. The parameters for these forces were appropriately selected to preserve the dynamic properties of flowing platelets in shear viscous flows. This complex repulsive-drag force used to achieve the no-slip condition at the surface, was used to compute the values of the stresses on the surface of the membrane, following the force virial contribution using the algorithm in [41, 67].

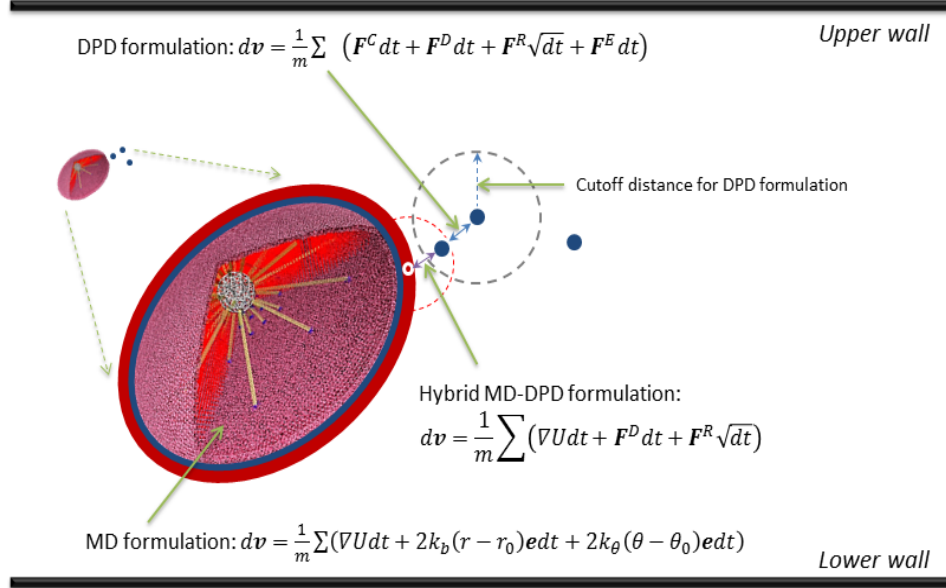


Figure 2-2: Multiple methods: DPD and CGMD in the multiscale approach.

### 2.3 Multiple Time Stepping Algorithm

The disparity in the spatial and temporal scales between the CGMD-based platelet and the DPD-based fluid is depicted in Figure 2-3, where bonded forces  $F^B$  and nonbonded force  $F^N$  cover molecular-scale interactions within the platelet.  $F^B$  consists of bond and angle forces between cytoskeleton particles.  $F^N$  is the L-J force between cytoplasm and other intra-platelet components including membrane and cytoskeleton. The macroscopic-scale interaction means the DPD force  $F^P$  for describing the bulk transport of flowing plasma. The mesoscopic hybrid force field  $F^M$  integrates the conservative force  $\nabla U_{LJ}$  from the molecular level and the thermostatic forces  $F^D + F^R$  from the macroscopic level and this requires a median time integrator. The bottom-scale CGMD force field is conservative while the top-scale DPD, as a novel thermostating method [43], considers the additive dissipative and random interactions.  $F^M$  follows such thermostating method as a variant of the DPD force field in which the soft conservative potential is replaced with a hard one.

The temporal scales for CGMD and DPD require nanoseconds and microseconds respectively [43, 68]. Standard time stepping (STS) algorithm requires a timestep small enough to resolve fastest motions [69]. Thus, the nanoscale integrator is applied to both top-scale and

bottom-scale methods for capturing the dynamics of platelets flowing in viscous blood plasma. However, this significantly increases the unnecessary computation for DPD with fine-grained integrators [43]. Moreover, even for the CGMD, a long-range and low-frequency force can be calculated with a larger time step than a short-range and high-frequency one [49]. To improve the computing efficiency within error boundaries, we must design MTS algorithms for handling the disparity in temporal scales between DPD and CGMD.

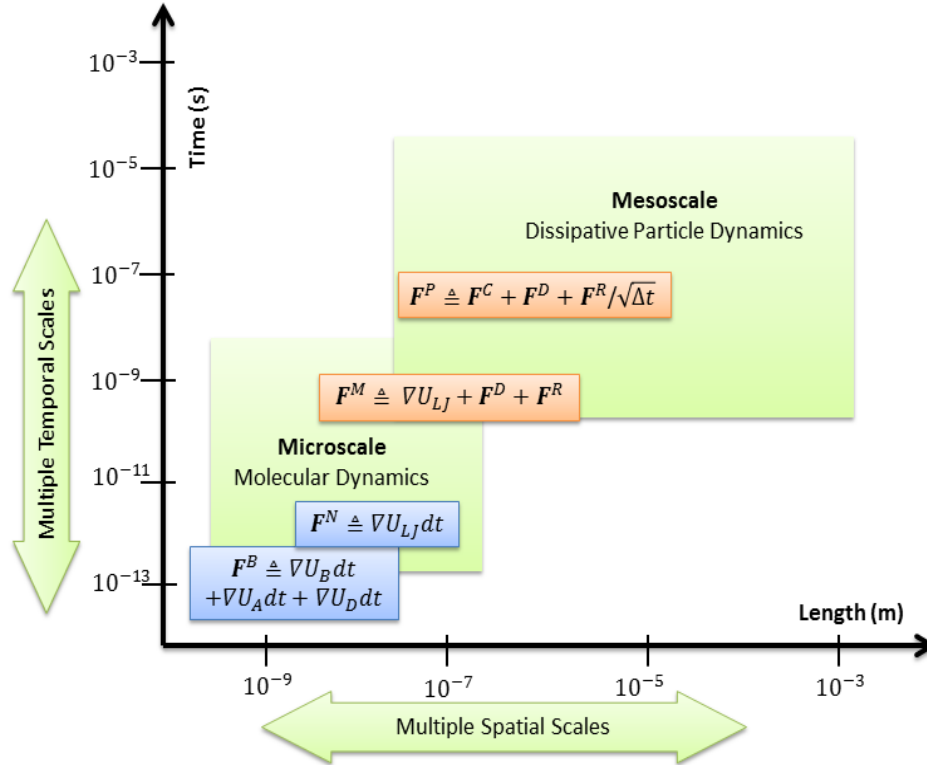


Figure 2-3: Multiple spatiotemporal scales in one model

### 2.3.1 MTS for DPD

In DPD, the forces are stochastic and nonlinear as the dissipative force depends on velocity [43, 45, 57] and, particularly, the conservative force  $F^C(r)$  in Equation 2-1 is similar to that in CGMD. The stochastic of this process disables the previous Euler-type algorithm used in the standard velocity Verlet integrator [70] that requires a velocity-independent force. Thus, a modified velocity Verlet integrator is derived from stochastic Trotter formula [43, 50, 51]:

$$\mathbf{r}(\Delta t) = \mathbf{r}(0) + \Delta t \cdot \mathbf{v}(0) + \frac{\Delta t^2}{2m} \mathbf{F}[\mathbf{r}(0), \mathbf{v}(0)]$$

$$\tilde{\mathbf{v}}(\Delta t) = \mathbf{v}(0) + \lambda \cdot \frac{\Delta t}{m} \cdot \mathbf{F}[\mathbf{r}(0), \mathbf{v}(0)]$$

compute  $\mathbf{F}[\mathbf{r}(\Delta t), \tilde{\mathbf{v}}(\Delta t)]$

$$\mathbf{v}(\Delta t) = \mathbf{v}(0) + \frac{\Delta t}{2} (\mathbf{F}[\mathbf{r}(0), \mathbf{v}(0)] + \mathbf{F}[\mathbf{r}(\Delta t), \tilde{\mathbf{v}}(\Delta t)])$$

Equation 2-5

Starting from initial conditions  $\{\mathbf{r}(0), \mathbf{v}(0)\}$ , one computes the position at full-step then a prediction for the new velocity, which is denoted by  $\tilde{\mathbf{v}}$ , and then computes the force and corrects the velocity at the last step. This is a basic predictor-corrector approach in which the velocities, for each time step, are predicted while estimating the force and are corrected at the end. The provisional values of velocities are crucial as the dissipative force depends on relative velocities of particles. The empirical factor  $\lambda$  accounts for the additive effects of stochastic interactions [43]. If the force is conservative without either dissipative or random terms,  $\lambda = 0.5$  restores the velocity Verlet integrator with  $O(\Delta t^2)$ .

Symeonidis et al [45] extend a MTS scheme to DPD to simulate a complex fluid with hard/soft potentials using two integrator scales:  $\delta t$  and  $\Delta t = n\delta t$  where  $n$  is a positive integer. Standard integrator in DPD is subdivided into a multi-rated dynamics of two matters with the hard and soft potentials described by the L-J potential and the soft-repulsive potential, respectively. The soft (or hard) matter employs the soft (or hard) potential so the dynamics is advanced with a larger (or smaller) time-step  $\Delta t$  (or  $\delta t$ ). With extension of velocity Verlet integrator, MTS for DPD is summarized in Table 2-1. In the table, the subscripts  $h$  and  $s$  correspond to variables for hard and soft potentials.  $F_h$  and  $F_s$  represent the forces derived for corresponding potentials and  $\lambda_h$  and  $\lambda_s$  are relaxation parameters.

Table 2-1: The multiple time-stepping algorithm within a 2-level integrator in DPD

▶	$\tilde{v}_s(\Delta t) \leftarrow v_s(0) + \lambda_s \cdot \frac{\Delta t}{m} \cdot F_s[r_s(0), v_s(0)]$	Soft potential
▶	For $l = 0 \dots n - 1$ ( $\delta t = \Delta t/n$ )	Hard potential
▶	set $t_0 = l \cdot \delta t, t_1 = (l + 1) \cdot \delta t$	Hard potential
▶	$\tilde{v}_h(t_1) \leftarrow v_h(t_0) + \lambda_h \cdot \frac{\delta t}{m} \cdot F_h[r_h(t_0), v_h(t_0)]$	Hard potential
▶	$r(t_1) \leftarrow r(t_0) + \delta t \cdot \tilde{v}(t_1)$	Soft/Hard potentials
▶	compute $F_h[r_h(t_1), \tilde{v}_h(t_1)]$	Hard potential
▶	$v_h(t_1) \leftarrow v_h(t_0) + \frac{\delta t}{2m} \cdot \{F_h[r_h(t_0), v_h(t_0)] + F_h[r_h(t_1), \tilde{v}_h(t_1)]\}$	Hard potential
▶	compute $F_s[r_s(\Delta t), \tilde{v}_s(\Delta t)]$	Soft potential
▶	$v_s(\Delta t) \leftarrow v_s(0) + \frac{\Delta t}{2m} \cdot \{F_s[r_s(0), v_s(0)] + F_s[r_s(\Delta t), \tilde{v}_s(\Delta t)]\}$	Soft potential

### 2.3.2 MTS for CGMD

In CGMD, the reversible reference system propagator algorithm (rRESPA) [47] is widely adopted. Standard integrators in molecular dynamics require evaluating forces at every time step regardless of interaction range, resulting in massive and, sometimes, unnecessary computation. A faster solution is to subdivide the pair forces  $F(x)$  into short- and long-ranged components  $F_s$  and  $F_l$ . The short-ranged force determines the time step  $\delta t$  and the long-ranged uses step size  $\Delta t = n \cdot \delta t$  where  $n$  is usually chosen as a positive integer.  $F_s$  includes the bonded forces while  $F_l$  the non-bonded forces such as Coulombic potential. rRESPA is based on Trotter expansion of classical Liouville operator [47-49] and the Liouville operator  $L$  for a system of  $N$  degrees of freedom in Cartesian coordinates is  $iL = \{\dots, H\} = \sum_{j=1}^f \left( \dot{x}_j \frac{\partial}{\partial x_j} + F_j \frac{\partial}{\partial p_j} \right)$  where  $\Gamma = \{x_j, p_j\}$  are the position and conjugate momenta of the system,  $F_j$  is the force on the  $j^{th}$  degree of freedom, and  $\{\dots, \dots\}$  is Poisson bracket of the system.  $L$  is a linear Hermitian operator on the space of square integrable functions of  $\Gamma$  [47] so it can be decomposed as  $iL = iL_1 + iL_2$  where  $iL_1 = F(x) \frac{\partial}{\partial p}$  and  $iL_2 = \dot{x} \frac{\partial}{\partial x}$ . Classical time propagator is  $U(t) = \exp(iLt)$  and the state of the system

at time  $t$  is  $\Gamma(t) = U(t)\Gamma(0) = e^{iLt}\Gamma(0)$  where  $\Gamma(0)$  is the initial state. Trotter expansion yields the velocity Verlet integrator [47, 48]:

$$\begin{aligned} \mathbf{v}\left(\frac{\Delta t}{2}\right) &= \mathbf{v}(0) + \frac{\Delta t}{2m} \cdot \mathbf{F}[r(0)] \\ \mathbf{r}(\Delta t) &= \mathbf{r}(0) + \Delta t \cdot \mathbf{v}\left(\frac{\Delta t}{2}\right) \\ \mathbf{v}(\Delta t) &= \mathbf{v}\left(\frac{\Delta t}{2}\right) + \frac{\Delta t}{2m} \cdot \mathbf{F}[r(\Delta t)] \end{aligned} \quad \text{Equation 2-6}$$

Starting from initial conditions  $\{\mathbf{r}(0), \mathbf{v}(0)\}$ , one computes the velocity at the half-step then the position at the full step, and completes calculation of the velocity at the second half-step. This velocity Verlet integrator performs better at small time steps than the position Verlet integrator [47] and is used in LAMMPS [71]. With rRESPA, the MTS algorithm for CGMD [71] is summarized Table 2-2. In our platelet model, the bonded force is used in cytoskeleton structure (filamentous bundles) and the nonbonded L-J force is used between cytoplasm particles. These forces act at nanometer scales. The equilibrium bond length for filamentous bundles is 21.3 nm and the equilibrium distance at which the L-J potential for cytoplasm is zero is 71.1 nm [38]. In this case, the bonded and nonbonded forces  $F^B$  and  $F^N$  are considered as the short and long range forces  $F_s$  and  $F_l$  (Table 2-2), respectively. This yields a separation of short-range bonded and long-range nonbonded forces in the CGMD-based platelet system. In the notation, we follow the traditional terminology for  $F_l$  and  $F_s$  in expressing MTS algorithms for molecular dynamics (Table 2-2) [47, 48]. When describing specific MTS schemes, we change to more specific terms  $F^N$  (CGMD-NB) and  $F^B$  (CGMD-BD). Recently, novel equations of motions are suggested such as in [72] that showed promising performance for molecular dynamics with large time step without resonant problems [73] and might be useful within CGMD when the framework needs remodeling for the future. This work followed standard Trotter expansion formulas as in [43, 47, 48, 50, 51].



Table 2-2: The multiple time-stepping algorithm within a 2-level integrator in CGMD

▶	$\mathbf{v}\left(\frac{\Delta t}{2}\right) \leftarrow \mathbf{v}(0) + \frac{\Delta t}{2m} \mathbf{F}_l[\mathbf{r}(0)]$	Long range force
▶	For $l = 0 \dots n - 1$ ( $\delta t = \Delta t/n$ )	Short range force
▶	set $t_0 = \frac{\Delta t}{2} + l \cdot \delta t$	Short range force
▶	$\mathbf{v}\left(t_0 + \frac{\delta t}{2}\right) \leftarrow \mathbf{v}(t_0) + \frac{\delta t}{2m} \mathbf{F}_s[\mathbf{r}(t_0)]$	Short range force
▶	$\mathbf{r}(t_0 + \delta t) \leftarrow \mathbf{r}(t_0) + \delta t \cdot \mathbf{v}\left(t_0 + \frac{\delta t}{2}\right)$	Long/Short range force
▶	compute $\mathbf{F}_s[\mathbf{r}(t_0 + \delta t)]$	Short range force
▶	$\mathbf{v}(t_0 + \delta t) \leftarrow \mathbf{v}\left(t_0 + \frac{\delta t}{2}\right) + \frac{\delta t}{2m} \mathbf{F}_s[\mathbf{r}(t_0 + \delta t)]$	Short range force
▶	compute $\mathbf{F}_l[\mathbf{r}(\Delta t)]$	Long range force
▶	$\mathbf{v}(\Delta t) \leftarrow \mathbf{v}\left(\frac{\Delta t}{2}\right) + \frac{\Delta t}{2m} \cdot \mathbf{F}_l[\mathbf{r}(\Delta t)]$	Long range force

### 2.3.3 Temporal Interfacing

MTS algorithms have been widely used in each of CGMD [47-49, 74] and DPD [43, 45, 71] independently and have been extended with the velocity Verlet integrator. Therefore, in our proposed integration of MTS algorithms for the DPD-CGMD model, we decompose the whole integrator process into four levels (Figure 2-4). The topmost two levels use the scheme in Table 2-1, referred as DPD-MTS, because both of them employ the DPD thermostatting method. The bottommost two levels use the scheme in Table 2-2, referred as CGMD-MTS, because both of them employ the conservative potentials. In each of DPD-MTS and CGMD-MTS, the integrator is subdivided into two time scales, one for the soft potential with a larger step size and the other for the hard potential with a smaller step size. Table 2-3 describes the 4-level integration procedure where communication and external forces are considered. In our multiscale model, “DPD” in the table implies the soft-repulsive, dissipative and random forces between the plasma particles, as well as the external forces applied to the plasma. It is advanced by the DPD-MTS integrator with  $\Delta t$  as defined in Table 2-4. “DPD-CGMD” implies the deformable platelet membrane as the contact region between the platelet and the surrounding fluid and it combines

the hard-repulsive L-J potential with thermostating variables, so that it is advanced by DPD-MTS but with a smaller step  $\delta t_1 = \Delta t / K_1$ . Wrapped by the membrane are the CGMD-governing intra-platelet constituents and thus CGMD-MTS is used for cytoskeleton and cytoplasm. The cytoskeleton component is for describing the very molecular-scale events such as filopodia formation and the cytoplasm component is for emulating transport of intra-platelet biofluids. The latter structure is grainier than the former resulting in the appropriate integrator choice. The cytoplasmic flow is applied with  $\delta t_2 = \delta t_1 / K_2$  and the cytoskeleton with the smallest  $\delta t_3 = \delta t_2 / K_3$ .  $K_1, K_2$  and  $K_3$  are all chosen to be integers. Following [43, 45], we use  $\lambda_p = \lambda_m = 0.5$ . The dimensionless and physical units rescaling adopts a dimensionless unit of time equal to  $1.2 \mu s$  in physical units. Thus, the step size for the DPD integrator is performed at the scale of  $10^{-3}$  in dimensionless units (i.e.,  $1.2 ns$ ), and the step size for the CGMD integrator at the scale of  $10^{-7}$  in dimensionless units (i.e.,  $120 fs$ ).

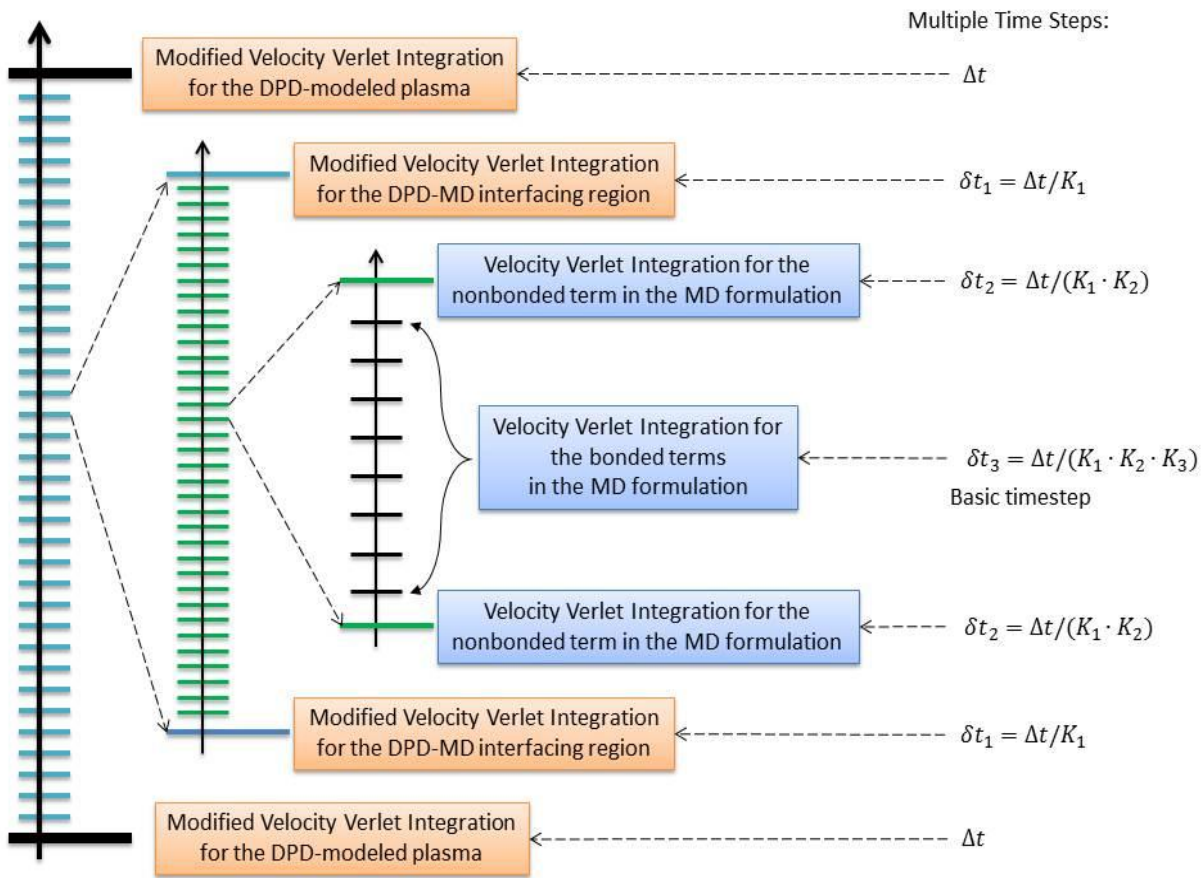


Figure 2-4: Multiple time step sizes in the MTS algorithm

Table 2-3: Overview of the multiple time-stepping algorithms for the multiscale model

▶	$\mathbf{v}_p \leftarrow \mathbf{v}_p + \lambda_p \cdot (\Delta t/m) \cdot \mathbf{F}^P$	DPD
▶	For $l_1 = 0 \dots K_1 - 1$	DPD-CGMD
▶	set $\delta t_1 \equiv \Delta t/K_1$	DPD-CGMD
▶	$\mathbf{v}_m \leftarrow \mathbf{v}_m + \lambda_m \cdot (\delta t_1/m) \cdot \mathbf{F}^M$	DPD-CGMD
▶	For $l_2 = 0 \dots K_2 - 1$	CGMD-NB
▶	set $\delta t_2 \equiv \delta t_1/K_2 = \Delta t/(K_1 \cdot K_2)$	CGMD-NB
▶	$\mathbf{v}_n \leftarrow \mathbf{v}_n + (\delta t_2/2m) \cdot \mathbf{F}^N$	CGMD-NB
▶	For $l_3 = 0 \dots K_3 - 1$	CGMD-BD
▶	set $\delta t \equiv \delta t_3 = \delta t_2/K_3 = \Delta t/(K_1 \cdot K_2 \cdot K_3)$	CGMD-BD
▶	$\mathbf{v}_b \leftarrow \mathbf{v}_b + (\delta t/2m) \cdot \mathbf{F}^B$	CGMD-BD
▶	$\mathbf{r} \leftarrow \mathbf{r} + \delta t \cdot \mathbf{v}$	All Particles
▶	Communication of positions and velocities	
▶	Compute $\mathbf{F}^B(\mathbf{r})$	CGMD-BD
▶	Communication of forces	
▶	$\mathbf{v}_b \leftarrow \mathbf{v}_b + (\delta t/2m) \cdot \mathbf{F}^B$	CGMD-BD
▶	compute $\mathbf{F}^N(\mathbf{r})$	CGMD-NB
▶	Communication of forces.	
▶	$\mathbf{v}_n \leftarrow \mathbf{v}_n + (\delta t_2/2m) \cdot \mathbf{F}^N$	CGMD-NB
▶	compute $\tilde{\mathbf{F}}^M(\mathbf{r}, \mathbf{v})$	DPD-CGMD
▶	Communication of forces.	
▶	$\mathbf{v}_m \leftarrow \mathbf{v}_m + (\delta t_1/2m) \cdot (\mathbf{F}^M + \tilde{\mathbf{F}}^M)$	DPD-CGMD
▶	$\mathbf{F}^M \leftarrow \tilde{\mathbf{F}}^M$	DPD-CGMD
▶	compute $\tilde{\mathbf{F}}^P(\mathbf{r}, \mathbf{v})$	DPD
▶	Communication of forces.	
▶	Add external forces to the viscous flow if any.	Add Forces to flow
▶	$\mathbf{v}_p \leftarrow \mathbf{v}_p + (\Delta t/2m) \cdot (\mathbf{F}^P + \tilde{\mathbf{F}}^P)$	DPD
▶	$\mathbf{F}^P \leftarrow \tilde{\mathbf{F}}^P$	DPD

## 2.4 Accuracy versus Speed

MTS algorithm is usually a matter of trade-offs between speed and accuracy for efficient multiscale simulations. Energy conservation and maintenance of adequate precisions of other measures must be verified while accelerating the computations. We aim to investigate the microscopic shape changes of platelets in response to macroscopic flow-induced stresses. Thus, in measuring numerical solutions we focus on the accuracy of characterizing the hybrid system and the flowing platelets, as well as the accuracy of calculating the dynamic flow-induced stresses on the surface membrane.

For comparative purposes we benchmark all algorithms against the standard time-stepping (STS) algorithm at smallest  $\Delta t = 10^{-7}$  in its integrator. The largest step size for the DPD flow regime is 0.001 which is within critical step size [43, 75]. In theoretical accuracy study, “critical step size” is the step size beyond which the numerical method starts to show pronounced artifacts in [75]. By varying parameters  $K_1$ ,  $K_2$  and  $K_3$ , we produce four representative MTS configurations as shown in Table 2-4. MTS-L denotes the most liberal integrator in which all timescales are increased up to the top scale while MTS-S represents the most conservative in which the top scale is further fine grained. The other two cases, MTS-M1 and MTS-M2 are the middle levels in which only step sizes for the hybrid force field and nonbonded intra-platelet potentials are adjusted. Through numerical experimentation, the impacts of MTS parameters are examined against STS.

Table 2-4: The time steps and configurations for different test cases

Case Name	Time steps for each scale				Configurations*			
	DPD	DPD-CGMD	CGMD-NB	CGMD-BD	$\Delta t$	$K_1$	$K_2$	$K_3$
MTS-L	$10^{-3}$	$2 \times 10^{-4}$	$2 \times 10^{-4}$	$2 \times 10^{-4}$	$10^{-3}$	5	1	1
MTS-M1	$10^{-3}$	$10^{-5}$	$10^{-6}$	$10^{-7}$	$10^{-3}$	100	10	10
MTS-M2	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-7}$	$10^{-3}$	10	10	100
MTS-S	$10^{-4}$	$10^{-4}$	$10^{-6}$	$10^{-7}$	$10^{-4}$	1	100	10
STS	$10^{-7}$	$10^{-7}$	$10^{-7}$	$10^{-7}$	$10^{-7}$	1	1	1

\*Note: The configurations used the same notations as in Figure 2-4 and Table 2-3.

## 2.4.1 Measures of Accuracy

Most measurable variables for our simulations depend on time and they fluctuate around a “mean” value and the long-time average is often considered as a steady state normal value [41]. Time-dependent function  $\varepsilon(v; t)$  measures the normalized deviation for variable  $v(t)$  from equilibrium over time (i.e., the time-averaged value  $v_0$ ) is

$$\varepsilon(v; t) = \frac{\|v(t) - v_0\|}{\|v_0\|} \quad \text{Equation 2-7}$$

where  $\|\cdot\|$  is a norm operator. In our model, we directly apply to temperature  $T$ , pressure  $P$  and total energy  $E_{tot}$  of the hybrid system, as well as kinetic energy  $(k_B T)_{\text{platelet}}$  of the platelet where  $k_B$  is Boltzmann’s constant. This indicator is for assessing the impacts of MTS parameters on the statistical stability of these measures. The dynamic stress distribution on the surface membrane also measures the accuracy of assessing the platelet activation factors. In this, we introduce a more rigorous per-particle comparison between MTS and STS, and design a 3-step procedure as follows: in Step I, the per-particle stress tensor  $\tau_{ij}(p, t) = [\tau_{\alpha\beta}]_{3 \times 3}$  where  $\alpha$  and  $\beta \in x, y, z$  to generate the 6 time-varying components of symmetric tensor [46, 67]:

$$\tau_{\alpha\beta} = - \left\{ m v_\alpha v_\beta + \frac{1}{2} \sum_{n=1}^{N_p} (r_{1\alpha} F_{1\beta} + r_{2\alpha} F_{2\beta}) + \frac{1}{2} \sum_{m=1}^{N_b} (r_{1\alpha} F_{1\beta} + r_{2\alpha} F_{2\beta}) \right\} \quad \text{Equation 2-8}$$

where  $r_1, r_2$  and  $F_1$  and  $F_2$  are the positions and resulting pairwise forces of two particles. The first term comes from kinetic energy and the rest computes a scalar virial produced by a group of interacting particles as defined in [67, 76]. Specifically, the second and third terms are from nonbonded and bonded pairwise energy, respectively, where  $n$  loops over  $N_p$  neighboring particles and  $m$  loops over  $N_b$  bonded particles. The virial theorem routinely computes the volume averaged tensors for a collection of particles and thus the per-particle stress needs to be divided by a per-particle volume to have the correct measure of stress [67, 76]. Equation 2-9 is an alternate formula of per-particle stress in CGMD [41, 76]. In Step II, the instantaneous stress tensor is rendered into a scalar  $\hat{t}(p, t)$  as defined in [18, 77]:

$$\hat{\tau}(p, t) = \frac{1}{\sqrt{3}} \left( \tau_{xx}^2 + \tau_{yy}^2 + \tau_{zz}^2 - \tau_{xx}\tau_{yy} - \tau_{xx}\tau_{zz} - \tau_{yy}\tau_{zz} + 3(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2) \right)^{\frac{1}{2}} \quad \text{Equation 2-9}$$

In Step III, the per-particle stress scalars are produced on the surface membrane and the root-mean-square-deviation (RMSD) is calculated to measure deviations of the per-particle stress scalars between the two systems:

$$\text{RMSD}(\hat{\tau}, t) = \sqrt{\frac{1}{N_m} \sum_{p=1}^{N_m} \|\hat{\tau}_m(p, t) - \hat{\tau}_0(p, t)\|^2} \quad \text{Equation 2-10}$$

where  $N_m$  is the total number of the membrane particles.  $\hat{\tau}_m(p, t)$  is the stress scalar of particle  $p$  at time  $t$  through using MTS. Similar to Equation 2-7,  $\hat{\tau}_0(p, t)$  is the averaged stress scalar of the same particle over a period of dynamic equilibrium through using STS.  $\text{RMSD}(\hat{\tau}, t)$  evolves with time and it is an indicator for stability of accurately assessing the dynamic stress distributions on the membrane and such an indicator for a MTS algorithm must be bounded when comparing with STS. Finally, integrating  $\text{RMSD}(\hat{\tau}, t)$  with  $\varepsilon(v; t)$  yields:

$$\varepsilon(\hat{\tau}; t) = \frac{\sqrt{\frac{1}{N_m} \sum_{p=1}^{N_m} \|\hat{\tau}_m(p, t) - \hat{\tau}_0(p, t)\|^2}}{\sqrt{\frac{1}{N_m} \sum_{p=1}^{N_m} \|\hat{\tau}_0(p, t)\|^2}} \quad \text{Equation 2-11}$$

The numerator is  $\text{RMSD}(\hat{\tau}, t)$  and the denominator is the spatial-averaged magnitude of the per-particle stresses on the surface membrane. Collectively,  $\varepsilon(\hat{\tau}; t)$  indicated the relative deviations of averaged stress  $\hat{\tau}(t)$  on the membrane.

## 2.4.2 Measures of Speed

The wallclock time of a simulation,  $\mathbb{T}(ts, np)$  in which  $ts$  is the simulated time (for the physical problem) and  $np$  is the number of processor cores allows us to define a normalized speed  $\mathbb{S}(np)$  as:

$$\mathbb{S}(np) = \frac{ts}{np \cdot \mathbb{T}(ts, np)} \quad \text{Equation 2-12}$$

The unit of  $\mathbb{T}(ts, np)$  is hours and the unit of  $ts$  is the dimensionless simulation time unit. Thus,  $\mathbb{S}(p)$  measures the length of simulated time per core-hour. A larger  $\mathbb{S}(p)$  means a faster simulation. Speedup of a MTS algorithm is defined as the ratio of the speed of a MTS algorithm over that of a STS algorithm so it refers to how much MTS is faster than STS using the same resources. Parallel efficiency of a parallel program is traditionally defined as:

$$\text{Efficiency}(ts, np) = \frac{\mathbb{T}(ts, 1)}{np \cdot \mathbb{T}(ts, np)} \quad \text{Equation 2-13}$$

where  $\mathbb{T}(ts, 1)$  is the wallclock time of a sequential program and  $ts$  defines the problem size. However, this efficiency measure is not conveniently applicable to our program of size far beyond the capability of any single node. To address this issue, we define the efficiency  $\mathbb{E}(np_1, np_2)$  as:

$$\mathbb{E}(np_1, np_2) \triangleq \frac{np_1 \cdot \mathbb{T}(ts_1, np_1)/ts_1}{np_2 \cdot \mathbb{T}(ts_2, np_2)/ts_2} = \frac{\mathbb{S}(np_2)}{\mathbb{S}(np_1)} \quad \text{for } np_1 < np_2 \quad \text{Equation 2-14}$$

where  $np_1$  and  $np_2$  are the numbers of cores and  $\mathbb{E}(np_1, np_2)$  uses the speed  $\mathbb{S}(np_1)$  as a baseline to measure the speed ratio due to core count difference  $\Delta p = np_2 - np_1$ . 100% means the perfect efficiency for adding these  $\Delta p$  cores. When  $np_1$  is small enough and the program has perfect parallelization on fairly small numbers of cores, then  $\mathbb{E}(np_1, np_2)$  asymptotically approaches  $\mathbb{E}(1, np_2)$ , i.e., conventional measures such as parallel efficiency in Equation 2-13 and thus this efficiency measure measures scalability more accurately.

## 2.5 Results and Analysis

To measure the accuracy vs. speed described earlier, we have performed numerous experiments of platelets flowing in benchmark Couette flows. In our simulations, the base step size for STS is  $\Delta t = 10^{-7}$  (dimensionless unit). The various MTS parameters are listed in Table 2-4. The simulations using MTS are compared with those using STS in terms of all selected measures described earlier. The spatial cutoff is set as  $r_c = 1.8$  for DPD,  $r_c = 1.6$  for DPD-CGMD and  $r_c = 1.0$  for CGMD. For our experiments, the system must go through a lengthy equilibration process before the plasma laminar flow is stably reproduced and results are collected.

The empirical selection of the larger step size in MTS is up to a multiplier of 10 of the well-established step size in STS and limited by the simulation stability. Symeonidis et al. [45]

use a multiplier of 10 in the time-staggered DPD model and the multiplier is smaller in CGMD involving the short- and long-ranged forces [48, 69]. However, in our model, the disparity in temporal scales between CGMD and DPD intrinsically poses a great gap, clearly mandating to perform such multiscale simulation more efficiently in order to overcome the computational challenge. For a further analysis of computational feasibility within certain error boundaries, two groups of measures, chaotic (sensitive to MTS parameters) and non-chaotic (insensitive) measures [53] are examined.

### 2.5.1 Analysis of Accuracy

Figure 2-5 to Figure 2-7 show the evolution of normalized deviations for temperature  $T$ , the pressure  $P$  and total energy  $E_{tot}$  of the hybrid system, respectively. Figure 2-8 and Figure 2-9 show the evolution of normalized deviations for kinetic energy of the platelet and dynamic stress distributions on the surface membrane, respectively. In all figures,  $\varepsilon(v; t)$  (as defined in Equation 2-7) is plotted as a function of the dimensionless time  $t$ , where  $v$  takes one of these measures. Mean values and standard deviations of these measures are present in Table 2-5 for comparisons. Furthermore, Figure 2-10 shows evolution of RMSD values for flow-induced stresses on the surface membrane by comparing with STS and Figure 2-11 illustrates 3D stress distributions in a color contour-magnitude diagram. Figure 2-12 illustrates the trajectories of platelet flipping using MTS-L and compare with the Jeffery's orbit that is often viewed as a reference point [58, 61, 78].



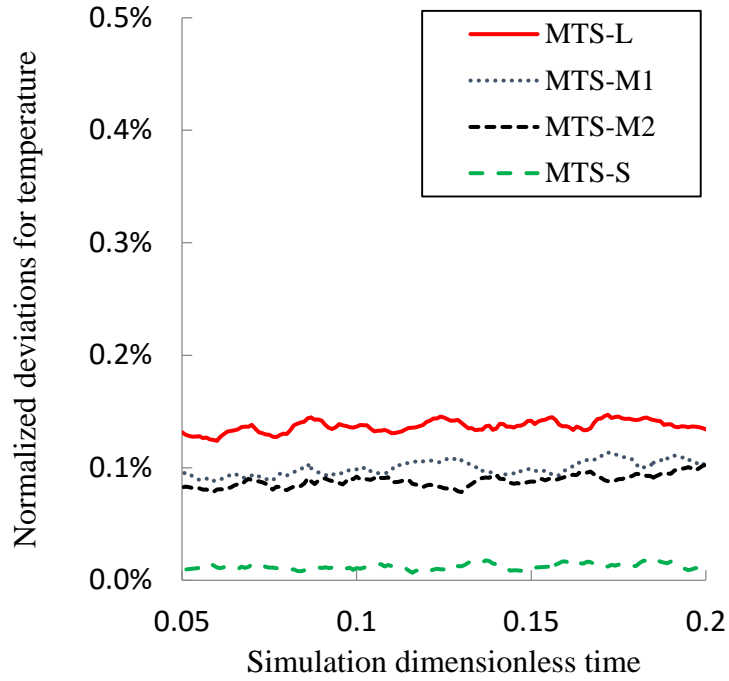


Figure 2-5: Normalized deviations for temperature  $T$  of the system  $\varepsilon(T, t)$  over time  $t$

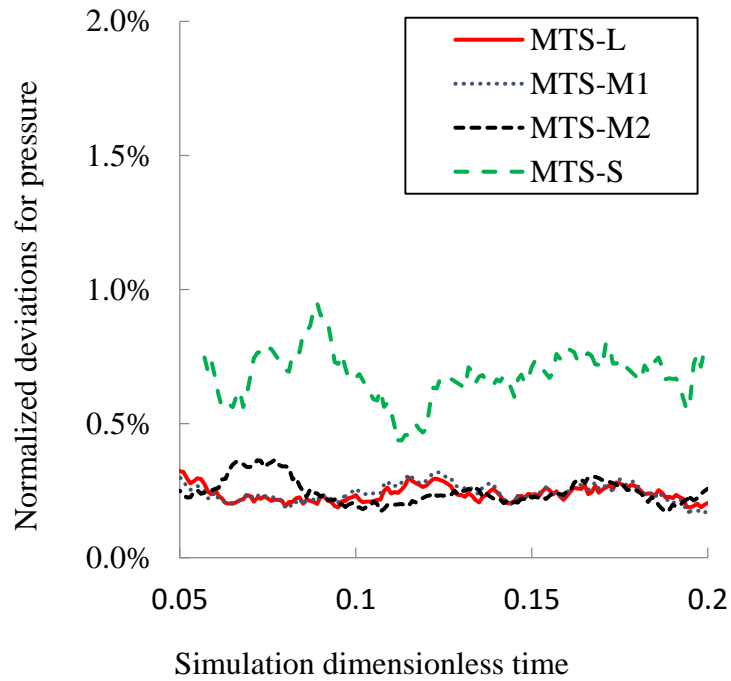


Figure 2-6: Normalized deviations for pressure  $P$  of the system  $\varepsilon(P, t)$  over time  $t$

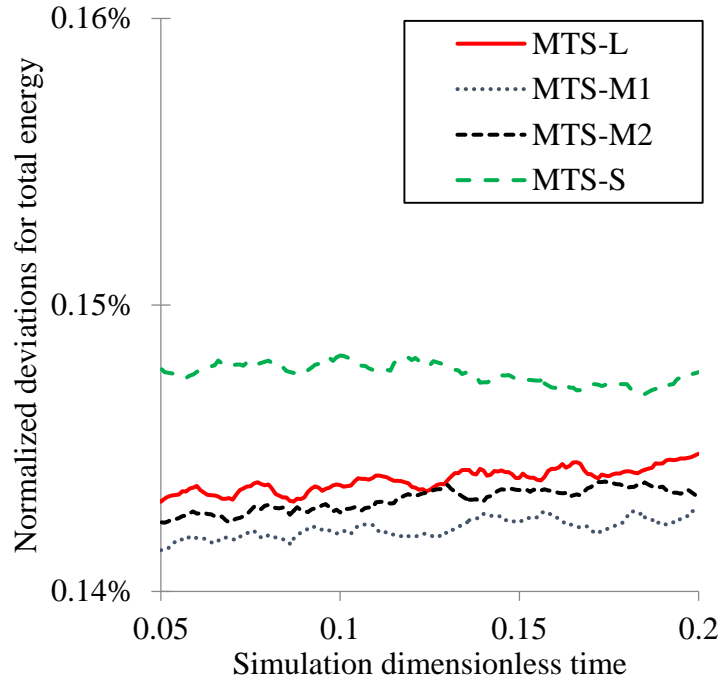


Figure 2-7: Normalized deviations for total energy of the system  $\varepsilon(E_{tot}, t)$  over time  $t$

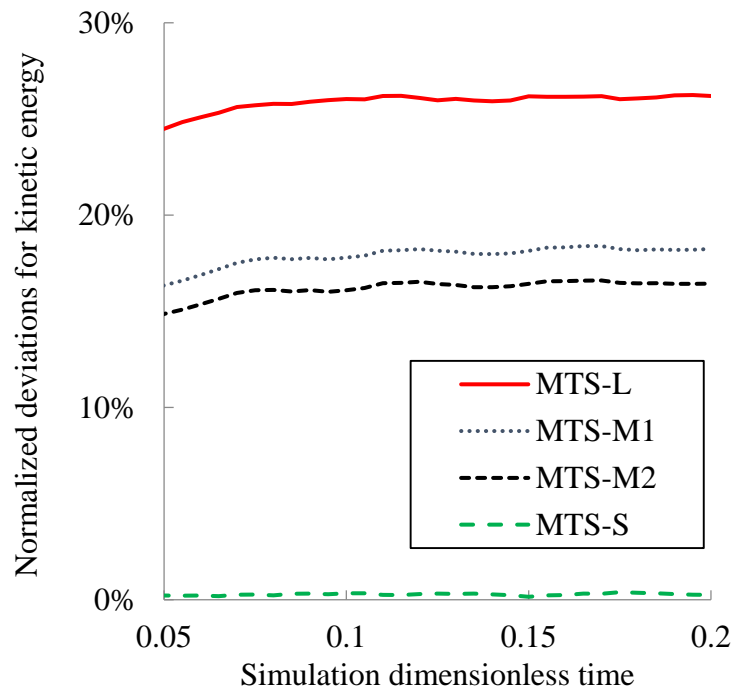


Figure 2-8: Normalized deviations for kinetic energy of single platelet  $\varepsilon((k_B T)_{platelet}, t)$  over time  $t$

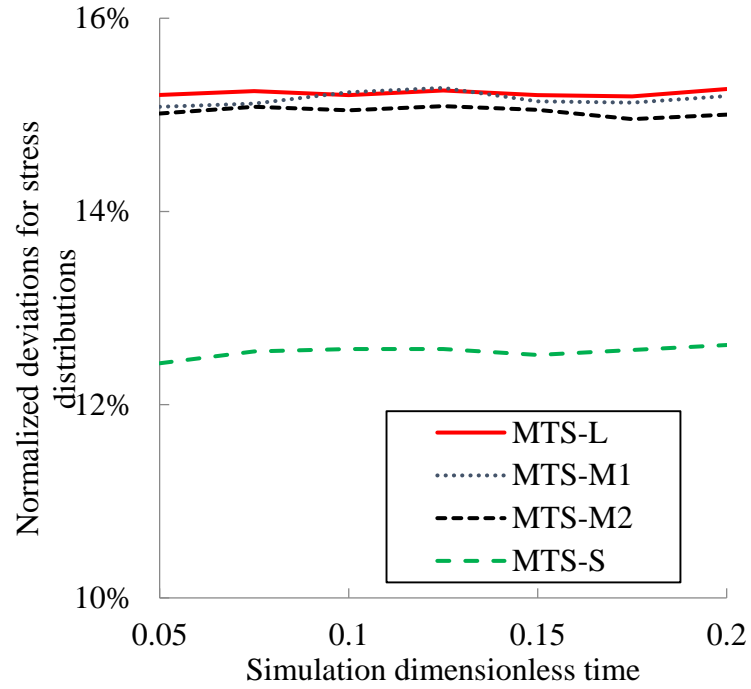


Figure 2-9: Normalized deviations for stress distribution on the membrane  $\varepsilon(\hat{t}, t)$  over time  $t$

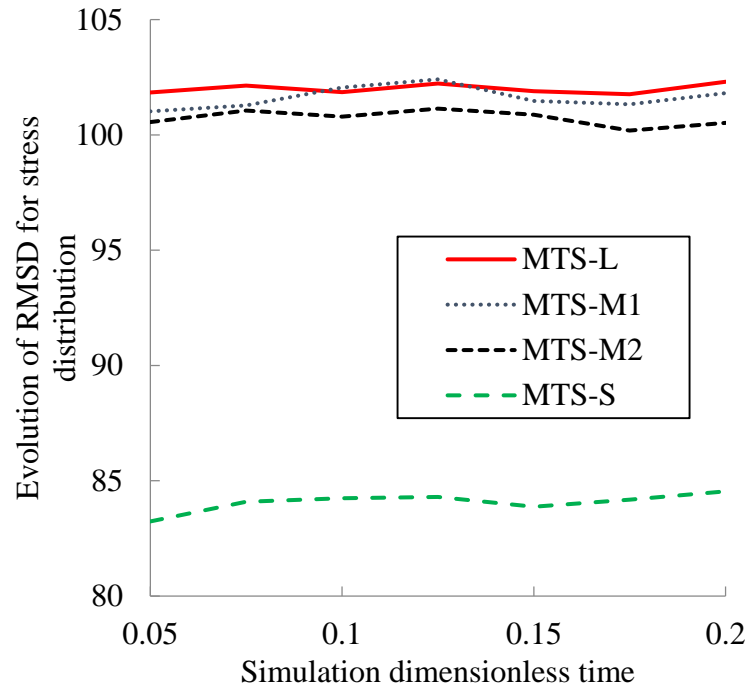


Figure 2-10: Evolution of RMSD for stress distribution on the membrane over time  $t$

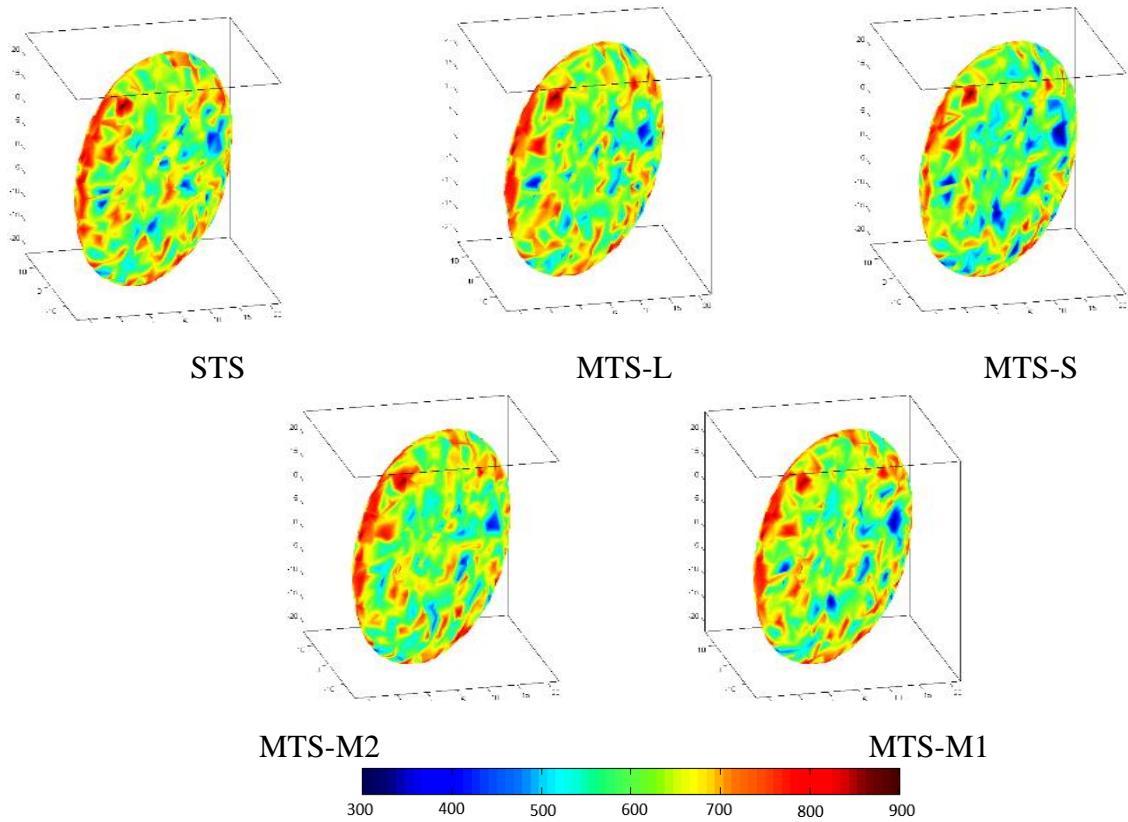


Figure 2-11: The dynamic stress distributions on the surface membrane using STS and MTS

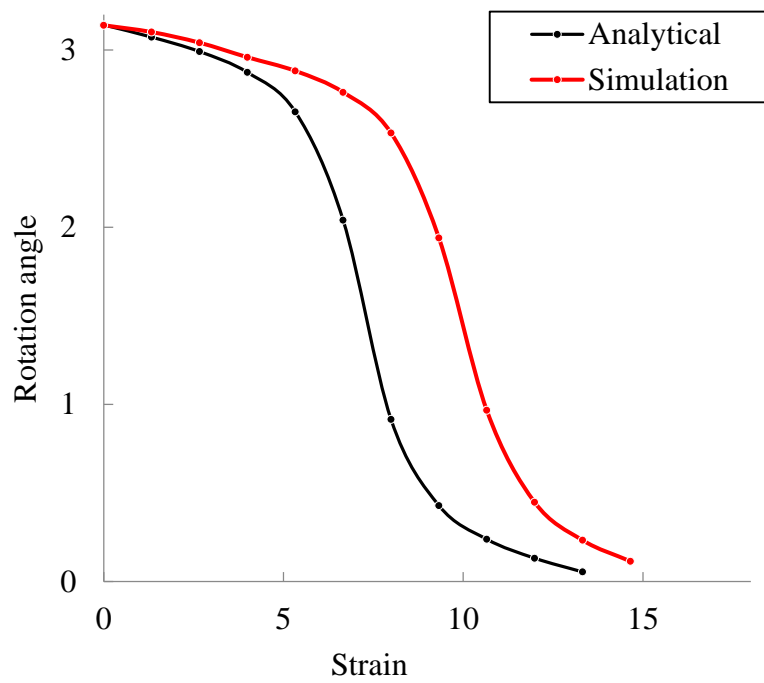


Figure 2-12: Change of rotational angles of the platelet using MTS-L, comparing with the Jeffery's orbit (analytical)

Among these measures, temperature and total energy of the hybrid system behave the most stable as their normalized deviations are consistently  $< 0.2\%$ . This verifies that selected MTS parameters maintained energy conservation from the system point of view. Pressure is another non-chaotic measure since it has achieved  $> 95\%$  in accuracy, though its normalized deviations are larger than those of  $T$  and  $E_{tot}$ . Obviously, there is no a clear-cut distinction between chaotic and non-chaotic measures and the assignment depends on MTS parameters. Thus, within selected MTS parameters and simulation period, macroscopic measures including temperature, pressure and total energy could be viewed as non-chaotic.

Comparatively, kinetic energy of the platelet is fairly sensitive to MTS parameters. The conservative scheme, MTS-S allowed a negligible deviation from STS but the aggressive scheme, MTS-L caused  $>25\%$  loss of accuracy. Two middle schemes, MTS-M1 and MTS-M2 posed a similar loss of accuracy between 15% and 20%. An advantage of four MTS schemes is that they all appear a bounded error range, thus avoiding propagation of resultant errors. This advantage offers an alternative way towards a computationally feasible simulation for large-scale time-consuming simulations, within certain error boundaries. Using MTS-L the trajectory of flipping platelets was consistent with Jeffery's orbit (Figure 2-12).

Finally, the dynamic stress distributions on the surface membrane are sensitive to MTS parameters but they are not as chaotic as the kinetic energy of the platelet. The loss of accuracy remains at  $\sim 15\%$  and it depends more on the choice of the DPD step size than the choice of other parameters. That is because: (a) MTS-L, MTS-M1 and MTS-M2 used the same step size  $\Delta t = 10^{-3}$  for DPD and they obtained a similar loss of accuracy. (b) Only MTS-S used  $\Delta t = 10^{-4}$  and increased the accuracy by  $>2\%$  demonstrates that the propagation of resultant errors was prevented within 16%. Figure 2-10 clearly reaffirms the absolute error boundaries of RMSD values. In an intuitive perspective, the visual representation of the resultant stresses acting on the platelet surface membrane is depicted in Figure 2-11 for comparing the effect of the different integrators on the resultant stress distribution.

We conclude that in MTS, the macroscopic measures such as  $T$ ,  $P$  and  $E_{tot}$  are non-chaotic while the kinetic energy of the platelet and the dynamic stress distributions on the surface membranes are chaotic. These experiments demonstrated that (i) the microscopic measures for

single platelets are more sensitive to MTS parameters than the macroscopic measures for the hybrid system. Specifically, the bulk transport of viscous fluid flows can safely advance with a larger step size and the study of microstructural changes of flowing platelets requires a smaller step size. (ii) Within the simulated period and selected step sizes, the errors do not propagate.

In addition to the simple scheme for integrating the DPD flow regime [43], there proposed many integrator variants [42, 75, 79] that allow larger time step sizes for accurate stochastic dynamics. Though the impact of CGMD MTS scheme dominates in this work with the goal of describing platelets dynamics in flowing blood plasma, one may extend other DPD MTS techniques in molecular modeling such as for complex fluids and polymers [72, 75].

Table 2-5: Accuracy and speedup comparisons for different MTS parameters

Metrics for Accuracy and Speed	MTS Test Cases			
	MTS-L	MTS-M1	MTS-M2	MTS-S
Normalized deviation for temperature: $\varepsilon(T, t)$	(0.136 $\pm$ 0.012) %	(0.099 $\pm$ 0.013)%	(0.088 $\pm$ 0.012)%	(0.012 $\pm$ 0.009)%
Normalized deviation for pressure: $\varepsilon(P, t)$	(0.245 $\pm$ 0.183)%	(0.247 $\pm$ 0.187)%	(0.247 $\pm$ 0.175)%	(0.706 $\pm$ 0.552)%
Normalized deviation for total energy: $\varepsilon(E_{tot}, t)$	(0.144 $\pm$ 0.001)%	(0.142 $\pm$ 0.001)	(0.143 $\pm$ 0.001)%	(0.148 $\pm$ 0.001)%
Normalized deviation for kinetic energy of platelet: $\varepsilon(k_B T, t)$	(25.15 $\pm$ 2.12)%	(16.96 $\pm$ 2.60)%	(15.36 $\pm$ 2.34)%	(0.303 $\pm$ 0.177)%
Normalized deviation for the stress distribution: $\varepsilon(\hat{\tau}, t)$	(15.22 $\pm$ 0.11)%	(15.16 $\pm$ 0.10)%	(15.03 $\pm$ 0.09)%	(12.53 $\pm$ 0.11)%
Parallel speed (simulated time per core-hour)	$5.58 \times 10^{-5}$	$1.53 \times 10^{-7}$	$2.00 \times 10^{-7}$	$1.56 \times 10^{-7}$
Speedup (ratio of MTS speed over STS speed)	2682.1	7.3	9.6	7.5
Parallel efficiency	51%	47%	52%	45%

Note: (i) The values for normalized deviations are represented as (mean $\pm$ standard deviation); (ii) Parallel performance metrics are compared based on experiments using 600 cores.

## 2.5.2 Analysis of Speed

There is a significant improvement in computing efficiency, though some measures showed somewhat chaotic behavior. Figure 2-13 presents absolute speeds of simulations using MTS and STS. Figure 2-14 shows speedups of MTS algorithms by comparing to STS. Figure 2-15 demonstrates parallel efficiencies of MTS and STS. Table 2-5 presents the performance of these MTS algorithms.

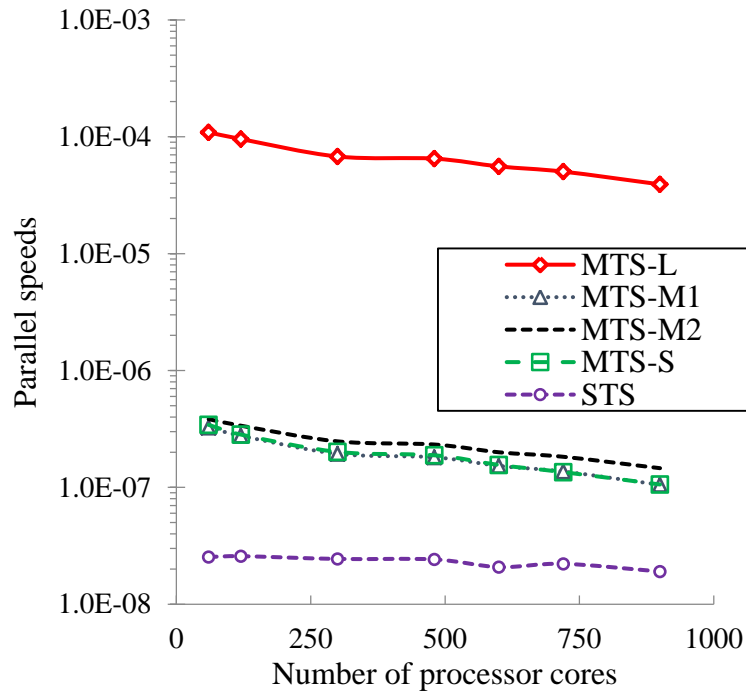


Figure 2-13: Parallel speeds  $S(P)$  vs. various processor cores  $p$  for the non-MTS and MTS solvers (A base 10 logarithmic scale for the vertical axis)

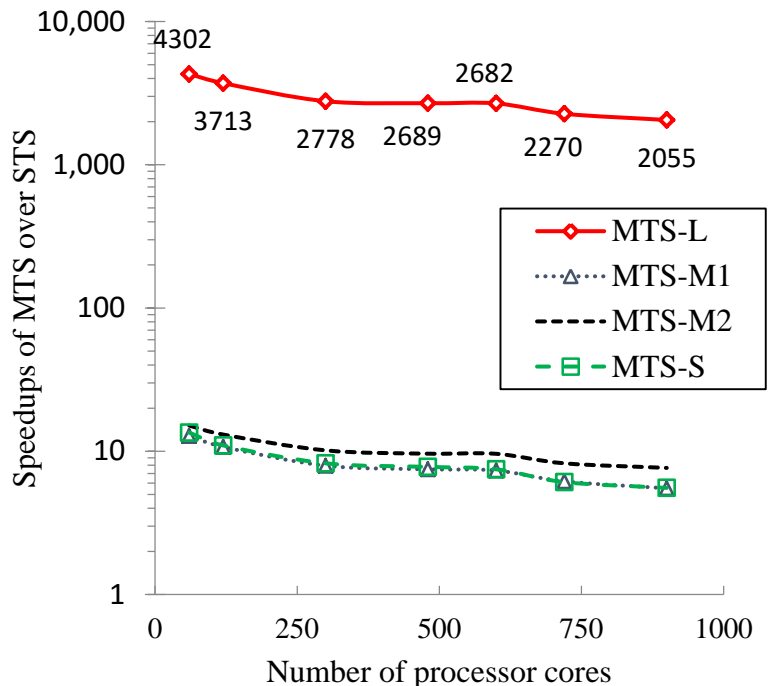


Figure 2-14: Speedups of MTS algorithms over the STS algorithm (A base 10 logarithmic scale for the vertical axis)

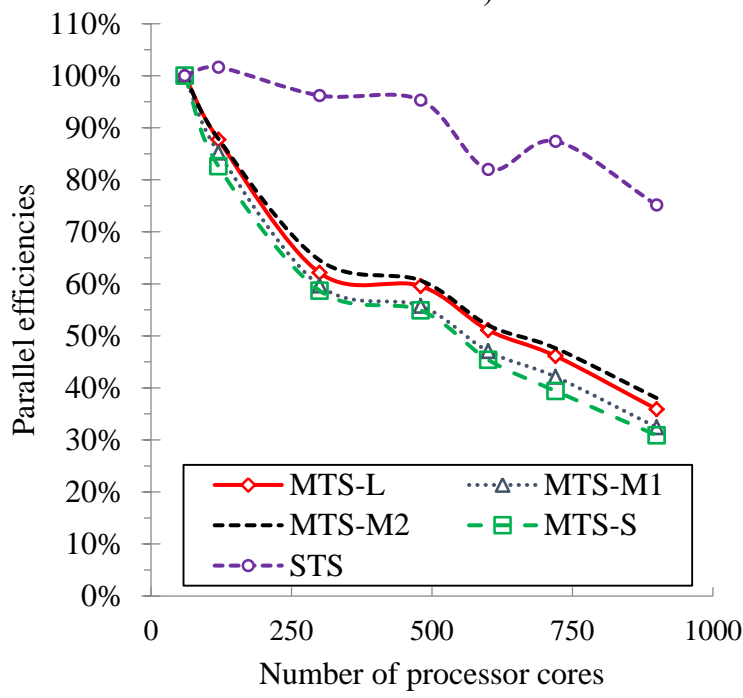


Figure 2-15: The parallel efficiencies  $\mathbb{E}(10, p_2)$  versus various processor cores  $p_2$  for the non-MTS and MTS solvers

In terms of absolute speeds, MTS-M1, MTS-M2 and MTS-S are roughly 10-fold faster than STS and MTS-L, achieving about 2000~3000 reduction in computation time. The timing



records presented in Figure 2-13, indicate that for the computational challenge of simulating one millisecond multiscale phenomena of platelets flowing in viscous fluid flow, STS requires 26 years using 600 cores, where using the same resources MTS requires only ~3.5 days. MTS algorithms offered an opportunity of simulating the molecular-scale mechanisms of deformable platelets flowing in the viscous fluid flows within affordable computing demands.

Figure 2-14 reiterates the constant effectiveness of MTS algorithms at a variety of computing system sizes. As the number of cores increases, communication is becoming a more dominant factor of limiting the scalability than computation. Thus, the speedups of MTS-L over STS decreased from >4000 to ~2000 while the parallel efficiencies decreased dramatically, as illustrated in Figure 2-15. Figure 2-16 compares the percentiles of communication and computation for MTS-L and STS. The results obviously demonstrate that MTS-L is effective at reducing the computation but it poses higher demands on communication than STS does. The performance analysis clearly indicated that a powerful communication system is quite important for improving the performance of multiscale modeling using MTS [32, 80].

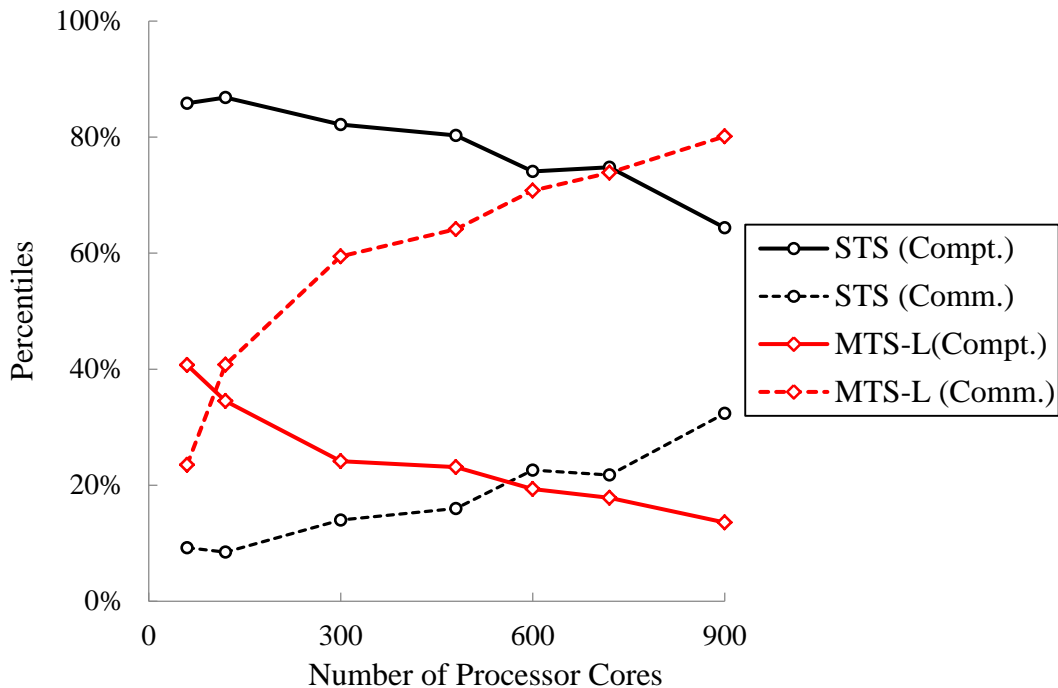


Figure 2-16: Percentiles of computation (Compt.) and communication (Comm.) over the total running time vs. the number of cores for the STS and MTS-L integrators

Accuracy analysis indicated that, as a measure, the kinetic energy of the platelet system is most chaotic. Analyzing the speed, we realized that the step sizes for CGMD and DPD significantly impacted the computational performance. These observations inspired us to seek a balance between accuracy and speed. In our experiments, we use normalized deviations of kinetic energy of the single platelet as the criterion of accuracy, terming it as the percentile error. A smaller percentile error usually results from a more fine-grained integrator. Additionally, we use the wallclock time (in days) to complete a 1-*ms* simulated time as the criterion of speed. Our computer has 300 cores. We start with step size for CGMD at  $10^{-7}$  and vary the step sizes for DPD as shown in Figure 2-17. We observed that: (a) when the step sizes for DPD increased from  $10^{-7}$  to  $10^{-5}$ , the speed considerably improved; (b) when the step size for DPD is between  $10^{-5}$  and  $10^{-3}$ , both the accuracy and speed curves reached a plateau. The percentile error was within 20%, though DPD advanced with a nanoscale step size ( $\Delta t = 10^{-3}$ ). Following that, we continued to use the step size  $10^{-3}$  for DPD and then varied the step sizes for CGMD as shown in Figure 2-18. We observed that: (c) increasing the step sizes for CGMD caused an obvious performance gain. However, (d) there was a critical point for the accuracy: when the step size for CGMD was close to  $10^{-3}$ , the accuracy quickly deteriorated. These experiments suggest that a nanoscale integrator for coarse-grained stochastic dynamics ( $\Delta t \sim 1.2 \text{ ns}$ ) and a sub-nanoscale integrator for coarse-grained molecular dynamics ( $\Delta t \sim 0.12 \text{ ns}$ ) constitute an optimal combination for accelerating the simulation speed while keeping the percentile error within 20%. These experiments demonstrated that appropriate choice of MTS parameters considerably improve the computational efficiency without a significant loss of accuracy, thus establishing a computationally feasible approach for solving a particle-based system at multiple scales for performing efficient multiscale simulations. The MTS optimization scheme developed and presented herein, tunes a multiple time stepping algorithm for combined DPD-CGMD simulations, which we have applied to the modeling of platelets dynamics in flowing blood plasma. This approach can be employed in any application using such a combined scheme. Generally, the top-scale method of such scheme could advance with much larger step sizes than the bottom-scale method and the spatial-interface between two scales should then adopt a corresponding middle step size. An optimal MTS scheme could be obtained for keeping accuracy of simulating bottom-scale within certain error boundaries.

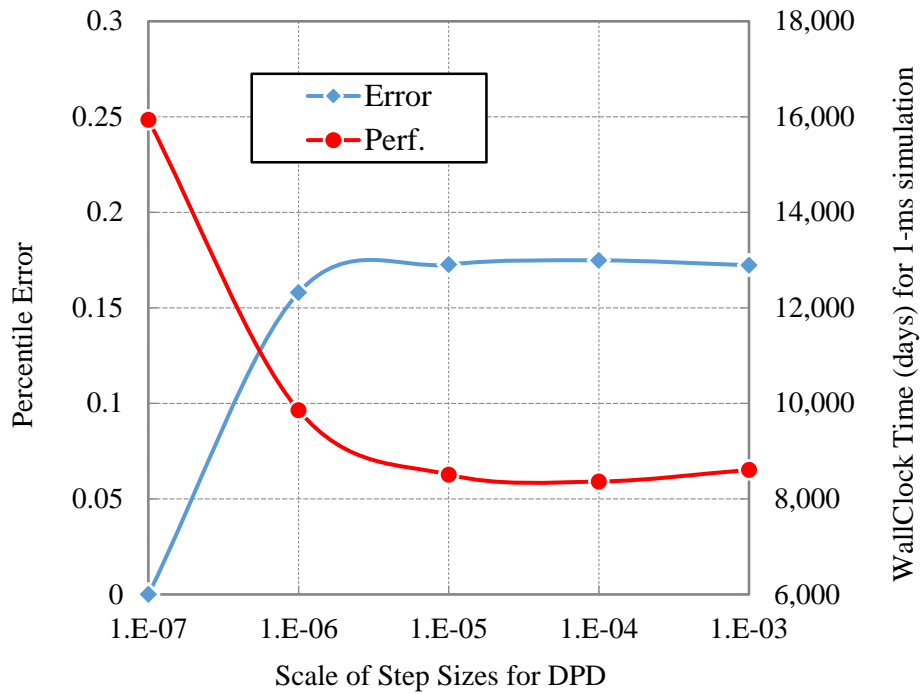


Figure 2-17: Percentile error and wallclock time (in days) for 1-*ms* simulation vs. different scales of step sizes for DPD in which the CGMD is integrated at  $10^{-7}$

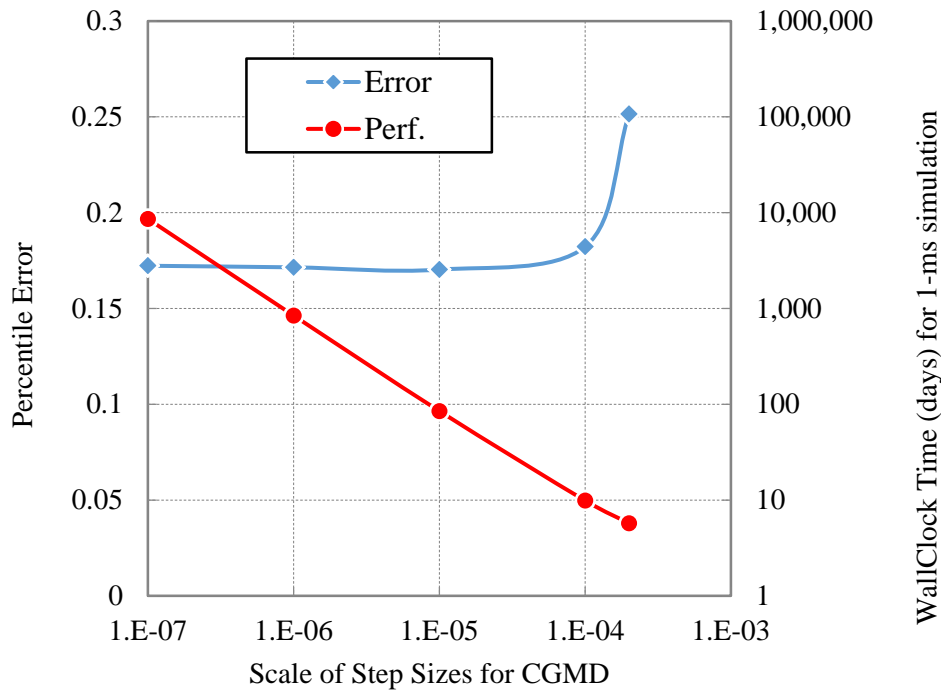


Figure 2-18: Percentile error and wallclock time (in days) for 1-*ms* simulation vs. different scales of step sizes for CGMD in which the DPD is integrated at  $10^{-3}$

## 2.6 Summary

This chapter presents an integrated multiple time-stepping (MTS) algorithm to solve a multiscale model of the dynamics of platelets flowing in viscous blood flow. The MTS algorithm proposes a multi-level integration scheme, allowing an optimization procedure for seeking a faster performer within certain error boundaries. The results demonstrated that the microscopic measures for single platelets are more sensitive to the MTS parameters than the macroscopic measures for the hybrid system; and the error propagation prevented in all cases studied. The results reaffirms that a separation of temporal scales in MTS considerably improves the efficiency of utilizing parallel computing resources, as compared to conventional single-scale methods in which considerable time is wasted conducting massive unnecessary computations (for example, completing 1-ms multiscale simulation of a 10-million particle system is reduced from 2.6 years to 3.5 days only). This work establishes a computationally feasible approach for solving large-scale particle-based systems at multiple scales for performing efficient multiscale simulations. Integrated MTS algorithms such as the one presented in the current work are essential for achieving full multiscale particle-based modeling of complex physiological flows, e.g., flow-induced platelet-mediated thrombosis, etc. Using MTS, we have achieved a dramatic reduction of the modeling time for system sizes and simulation times that are relevant to multiscale phenomena such as thrombosis formation. By developing computationally feasible and efficient approaches, such challenging simulations of molecular biomechanics at multiple scales are brought within the reach of current high-performance computing resources.

## Chapter 3 Discovery of Force Field

### 3.1 Introduction

Viscous blood flow dynamics play a major role in cardiovascular devices (CVS) design process which in recent years relies more heavily on numerical simulation [9, 10]. While the advent of these devices has provided life-saving solutions to millions of patients in the United States [81, 82], thromboembolism remains an impediment in which shear induced platelet activation stimulates blood clotting [12-14]. To reduce the thrombogenic risk potential of the devices, efficient numerical simulations of blood flow need to be able to model not just the flow dynamics but processes pertinent to flow induced blood clotting.

Computational fluid dynamics (CFD) is a well-established and universal continuum approach to study complex fluid flows. However, CFD simulations, while able to capture the overall flow mechanisms, are too coarse to model the finer features of blood particulate flow and fully describe the interactions of key players in blood coagulation such as platelets and other cells those may involve [24, 83]. To address the limits of continuum approaches [84, 85], dissipative particle dynamics (DPD) approach is introduced [86] to model heterogeneous fluids and biophysical details that are difficult to achieve using continuum approaches because the molecular effects, e.g., adhesion and aggregation bonds of blood clotting occur at the nano to micro scales. The coupling of the disparate spatial and time scales inspires multiscale simulation studies using approaches which depart from the continuum approaches [21, 87, 88]. A potential approach for such studies presented here is by coarse graining the atomistic based molecular dynamics (MD) to tradeoff between physical details and modeling feasibility.

In the past two to three decades, considerable efforts have been devoted to developing coarse-grained (CG) molecular models for studying polymers, bio membranes, surfactants and hemodynamics [89-97]. These CG models enable to focus on the particular scales of interactions by averaging the less essential degrees of freedom, resulting in a reduction of redundant computational loads. The selection of the degrees of freedom for coarse graining depends on the phenomena the simulation is trying to achieve. Therefore, some CG models encapsulate whole molecules while others treat several molecules as one effective CG particle. For example, one CG particle represents one water molecule in [89, 91, 93, 98], or three water molecules in [96,

97], or four in [99, 100], or even five in [95], according to the simulation scales that best represent the phenomenon the simulation is trying to achieve. These CG models, while losing some resolution of the physical properties, enable us to consider larger systems within reasonable computational costs. Typically, the target properties include the thermodynamic properties, e.g., enthalpy of vaporization, free energy of solvation and interfacial tension, and statistical properties, e.g., radial distribution function and mean square displacement.

Developing a reliable CG model is challenging, in particular the effective force field with properly fitting parameters. Characteristically, a CG model simplifies the molecular description by smearing the interacting complexes, as save the computational cost of resolving the intricate details of physical properties. Deriving the effective potential analytically from statistical mechanics or other lower order principles may become limited to few simple cases [97]. A viable alternative for defining force field potential functions is to employ some simple and empirical models with a finite precision. Lennard-Jones (LJ) potential is the most popular pairwise non-bonded interaction model for simple applications, with limited accuracy. However, for, e.g., the case of complex viscous fluids, it offers only limited predictive capabilities [21]. Morse potential may fill the void in the “model space” although in its original form it is also limited. In 1929, physicist P. M. Morse developed the potential to describe chemical bond formation and dissociation [101]. In 2003, a Morse-like potential was adapted and proven capable of reproducing phase transitions and liquid-vapor coexistence curves of real fluids [90]. In 2010, it was parameterized for CG modeling of water and the *n*-alkanes and achieved a good agreement of various properties, including the enthalpy of vaporization, bulk densities, interfacial tensions, free energies of transfer, diffusion coefficients and isothermal compressibility [99].

We extended the Morse potential approach presented in [99] to a CG model which includes a modification of the Morse potential, in order to simulate the flow of a blood plasma fluid. The key contributions of our work include parameterizing the model under multiple scales for fitting commonly used hemodynamic properties: density, pressure, isothermal compressibility and viscosity. Our model also reproduces the Counter-Poiseuille and Couette flows in agreement with these analytical benchmark solutions.

### 3.2 Simulation

We model the blood plasma fluid by using CG particles, each of which lumps the aggregate effect of an ensemble of molecules. The total mass  $M_{CG}$  of each CG particle is the sum of the masses of these molecules, measured in atomic mass unit ( $amu$ ), and the position of a CG particle is the center of mass of this ensemble. The average distance  $\mu$  is measured as the mean-free-path of the CG particles:

$$\mu = \rho_p^{-1/3} = (M_{CG}/\rho_m)^{1/3} \quad \text{Equation 3-1}$$

where  $\rho_m$  and  $\rho_p$  are the mass and particle density respectively. Obviously, for a system with fixed number of molecules, a larger  $\mu$  implies a larger coarse-graining level, i.e., each CG particle contains more molecules. The growth of  $\mu$  with the exponential increase of  $M_{CG}$  is shown in Figure 3-1, showing the relationship between  $M_{CG}$  and the graining level.

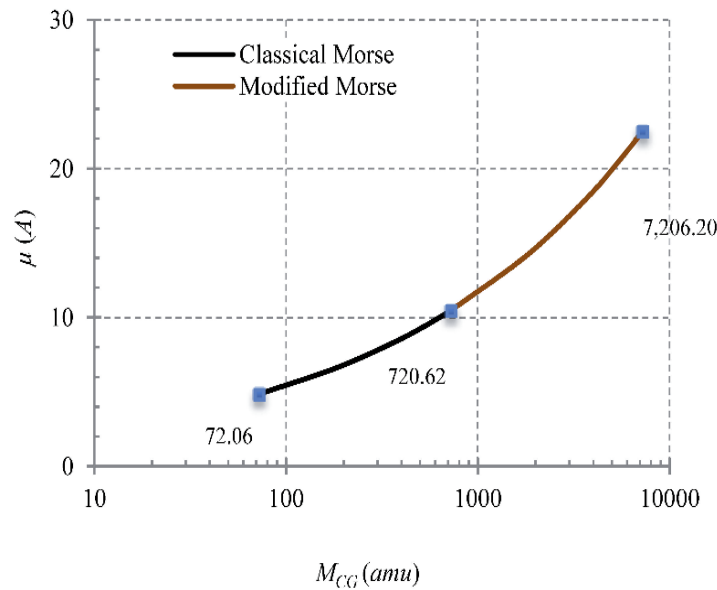


Figure 3-1: The relationship of the average distances over mass scales:  $M_{CG} (amu)$  vs.  $\mu (\text{\AA})$  where  $x$ -axis is in logarithm scale of base 10.

We use a single water molecule of 18.0154  $amu$  as the mass unit of a basic molecule since blood plasma, which constitutes 55% of blood fluid, consists mostly of water (92% by volume). The W4 model, developed recently [99] for coarse-graining water, lumps four water molecules of 72.062  $amu$  into one effective CG particle. In adapting the model to blood plasma we had to coarsen further the W4 model to include approximately 40 to 400 water molecules.

Specifically, we extend the W4 model by further increasing the CG levels:  $M_{CG}=72$ , 720 and 7206 *amu* for blood plasma. However, the original form of the Morse potential previously applied for the W4 model fails to express the interactions of the coarser model we are considering. After carefully studying several alternatives, we introduce the following modified Morse potential:

$$U_{ij}(r) = \varepsilon[\exp(\alpha(1 - r_{ij}/R(\mu)) - 2\exp(\alpha(1 - r_{ij}/R(\mu))/2)] \quad \text{Equation 3-2}$$

where  $R(\mu) = a/(\mu - b) + c$ ,  $U_{ij}(r)$  is a pairwise non-bonded potential energy and  $r_{ij}$  is the relative distance of a particle pair,  $R(\mu)$  is the distance of minimum energy  $\varepsilon$  and  $\alpha$  is a parameter that measures the curvature of the potential around  $R(\mu)$ . It contains three positive parameters  $a$ ,  $b$  and  $c$  in units of  $\text{\AA}^2$ ,  $\text{\AA}$  and  $\text{\AA}$  respectively. Parameter  $a$  is related to the surface tension of the CG particle. It specifies the deviation of our modified Morse potential from the original Morse potential and it becomes 0 when our model reduces to the original. The parameters:  $\varepsilon$ ,  $\alpha$  and  $R(\mu)$  in Equation 3-2 are obtained through the conventional inverse problem approach, i.e., the parameters are adjusted so that the simulation output best matches published blood plasma properties (summarized in Table 3-1).

Table 3-1: Mechanical properties of blood plasma fluid

Properties	Symbol	Values	Units
Mass density	$\rho_{m,0}$	1.05	$g/cm^3$
Fluid pressures	$P_0$	1.12 / 1.17	<i>bar</i>
Isothermal compressibility [102]	$\kappa_{T,0}$	$4.6 \times 10^{-5}$	$bar^{-1}$
Shear viscosity [103]	$\eta_0$	1.10-1.30	$mPa \cdot s$

All simulations were performed at 310 *Kelvin* with NVT ensemble [104], using the LAMMPS (Large-scale of Atomic/Molecular Massively Parallel Simulator) code [46] (21-Dec-2011 version). A cubic box with 27,000 CG particles and mass scales of 72, 720 and 7200 *amu* were tested. Specific side lengths with reference to the density were used together with periodic boundary conditions. The CG particles are treated as mathematical dots for which all internal rotational and vibrational degrees of freedom within each CG particle are smeared out. The Berendsen thermostat method [105], which is realized by coupling to external bath, is implemented for temperature control. The isothermal compressibility  $\kappa_T$  is calculated using the finite difference expression [106] and is expressed as:



$$\kappa_T = -\frac{1}{V} \left( \frac{\partial V}{\partial p} \right)_T = \left( \frac{\partial \ln(\rho)}{\partial p} \right)_T \approx \left( \frac{\ln(\rho_2/\rho_1)}{p_2 - p_1} \right)_T \quad \text{Equation 3-3}$$

The shear viscosity  $\eta$  is calculated using the Green Kubo (GK) method [107, 108]. In this method,  $\eta$  is given by integral of an accurate time-correlation of the equilibrium fluctuations of the corresponding flux and is expressed as:

$$\eta = \frac{V}{3k_B T} \int_0^\infty \sum \langle P_{\alpha\beta}(0)P_{\alpha\beta}(t) \rangle dt \quad \text{Equation 3-4}$$

where  $\alpha\beta \in \{xy, yz, xz\}$ .  $V$  is the volume of the system,  $k_B$  is the Boltzmann constant and  $T$  is the temperature.  $P_{\alpha\beta}$  refers to off-diagonal component of the pressure tensor. The angle brackets around the summation refer to an average of a “sufficiently large” sample. Equation 3-4 can be re-written in the form:

$$\eta = \lambda \int_0^\infty C_{\alpha\beta}(t) dt \quad \text{Equation 3-5}$$

where  $C_{\alpha\beta}(t)$  is the stress tensor autocorrelation function and  $\lambda$  is a constant. Parameter  $\tau_v$  is the characteristic time and is used for determining the number of samples to control the error  $E_v$  at below 5%.

### 3.3 Parameterization

Determining the interacting potential of CG particles involves at least two steps: (1) constructing the mathematical structures of the modified Morse potential and (2) deciding the parameters in the formula. This second step is referred to as parameterization and is accomplished through numerical experiments, as described below.

A series of numerical experiments on classical Morse potential allowed us to understand the dependencies of different target properties on various parameters, summarized in Table 3-2.

Table 3-2: Impact of model parameters on target properties

Impacts		Target Properties		
		$P$	$\kappa_T$	$\eta$
Change of parameters	Increase $\alpha$	<->	↓	<->
	Increase $R$	↑	<->	↑
	Increase $\varepsilon$	↓	↓	↑
	Increase $\rho$	↑	↓	↑

We notice that a series of models with  $\alpha = 7\sim 10$  are capable of reproducing desired properties, in particular,  $\alpha = 10$  allows a longer timestep for dynamic simulations.  $R$  greatly influences the pressure; both  $\kappa_T$  and  $\eta$  are closely related with  $\varepsilon$ ; and, if  $\varepsilon$  is larger than some threshold, GK autocorrelation will diverge, resulting in a non-fluid behavior. With these observations, we exhaustively search the parameter space for the optimal set of parameters. Particularly, for a given  $M_{CG}$ , we decouple the parameters in individual stages:

Stage 1: Given  $\alpha$  and  $\varepsilon$ , search  $R$  to approximate  $P_0$ , resulting in a series of isothermal-isobaric curves under equilibrium. This yields  $R = R(\alpha, \varepsilon)$ .

Stage 2: Given  $\alpha$ , compute  $\kappa_T$  and  $\eta$  under a series of  $\varepsilon$  and  $R(\alpha, \varepsilon)$ , to find the optimal combination.

To measure the accuracy of the approximating parameters, we normalize  $\kappa_T$  and  $\eta$  by dividing them by their target values. Obviously, the normalized  $\kappa_T$  or  $\eta$  is expected to be “1” as its ideal value.

Using the 2-stage approach, Figure 3-2 demonstrates that the classical Morse potential is parameterized well to express interactions of all CG particles at  $M_{CG} = 72.06 \text{ amu}$  (*W4*). However, the 2-stage exhaustive search approach fails to parameterize the classical Morse potential for the CG level of  $M_{CG} = 720.62 \text{ amu}$ . This is depicted in Figure 3-3: when desired  $\eta$  is obtained,  $\kappa_T$  is still far from its target value. On the other hand, when we continue increasing  $\varepsilon$  for improving  $\kappa_T$ , GK autocorrelation diverges, driving the system out of its liquid phase range. This demonstrates that the classical Morse potential is inadequate for expressing the interactions of the larger CG particles needed for simulating blood plasma fluid.

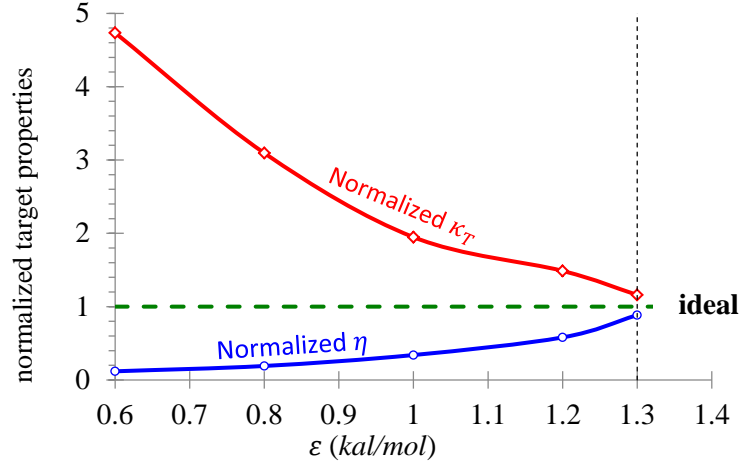


Figure 3-2: Impact of  $\epsilon$  in classical Morse potential on normalized compressibility  $\kappa_T$  and viscosity  $\eta$  where  $\alpha=10$  and  $M_{CG} = 72.06 \text{ amu}$

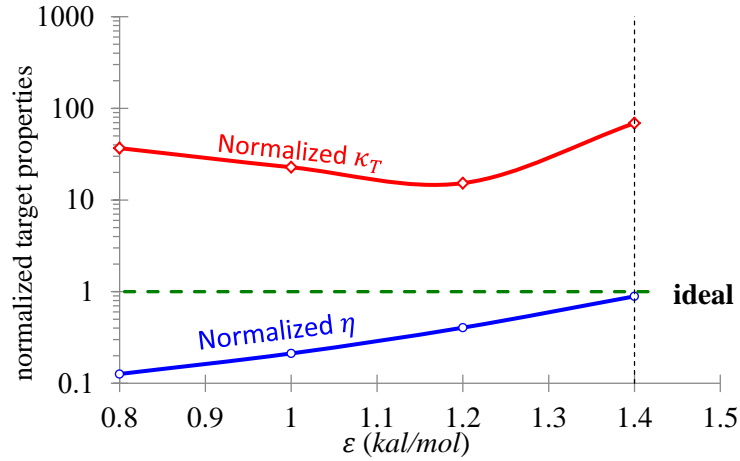


Figure 3-3: Impact of  $\epsilon$  in classical Morse potential on normalized compressibility  $\kappa_T$  and viscosity  $\eta$  where  $\alpha=10$  and  $M_{CG} = 720.62 \text{ amu}$

To address the limitation of the classical Morse potential we introduce a form factor  $R$  as a function of average distance  $\mu$ . Accordingly, we modify the exhaustive search approach as follows:

Stage 1: Given  $\alpha$  and  $\epsilon$ , search  $R$  to approximate  $P_0$ , resulting in a series of isothermal-isobaric curves under equilibrium. This yields  $R = R(\alpha, \epsilon)$ .

Stage 2: Given  $\alpha$ , compute  $\eta$  under a series of  $\epsilon$  and  $R(\alpha, \epsilon)$ , to find the optimal combination  $(\alpha_c, \epsilon_c, R_c)$  for approximating  $\eta_0$ .

Stage 3: Given  $(\alpha_c, \varepsilon_c, R_c)$ , the system is compressed in a certain range to get a series of  $\{\mu_i\}_{i=0}^n$ . The compression ratio is no more than 5%. The distance  $\mu_0$  measures the average distance of the uncompressed system and  $R_c$  is associated with  $\mu_0$ . For  $\{\mu_i\}_{i=1}^n$ ,  $\{R_i\}_{i=1}^n$  is adjusted to obtain the desired pressure increment as calculated in Equation 3-3. Lastly, the form  $R(\mu) = a/(\mu - b) + c$  is adopted by applying a non-linear least square fit of the data points  $(\mu_i, R_i)$ .

Using this 3-stage approach, the modified Morse potential can be conveniently parameterized to express interactions of CG particles at the range of  $M_{CG}=720.62$  through  $7206.20 \text{ amu}$  effective mass scales. Figure 3-5 and Figure 3-7 depict the pressure variation when the system is compressed, where the ideal values are computed through Eq. (3) and the experimental values are obtained through simulations. The corresponding functions  $R(\mu)$  are present in Figure 3-4 and Figure 3-6 respectively. The parameters for all tested CG levels are present in Table 3-3. In the table, we can see that a more coarsening level would lose some resolution of physical properties than a less coarsening level. For example, experimenting with various  $M_{CG}$  values indicates that the viscosity cannot reach the desired value for  $M_{CG} = 7206.20 \text{ amu}$  at normal pressure but the characteristic viscosity could be maintained at a higher pressure. Therefore, we selected the parameters in Table 3-3 to enable our fit to most of the physical characteristics including the viscosity and the pressure. Although the high level of coarsening does increase the inaccuracies of the physical properties, these inaccuracies are under control and we are able to consider larger systems without adding too much computational cost.

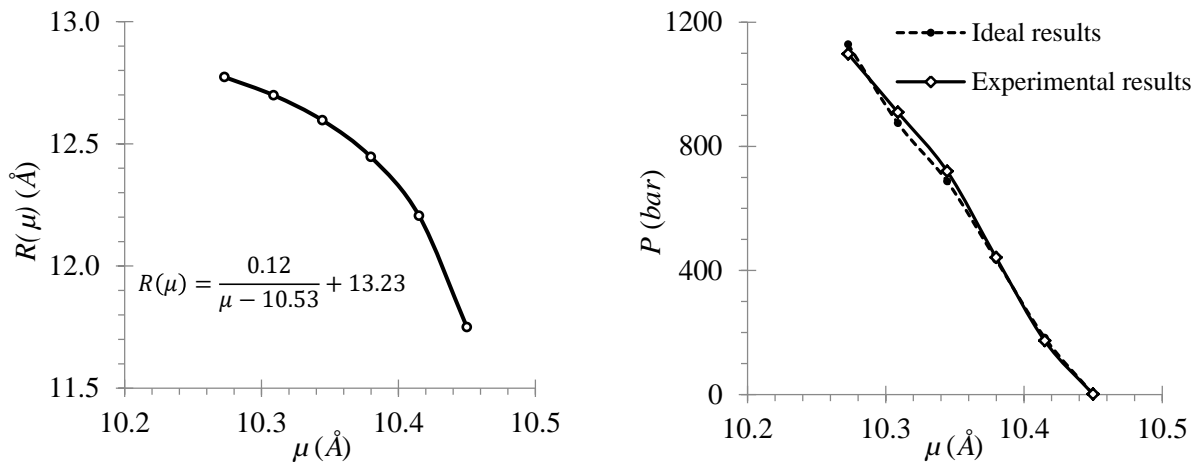


Figure 3-4: The form factor  $R(\mu)$  in modified Morse potential for  $M_{CG} = 720.62 \text{ amu}$

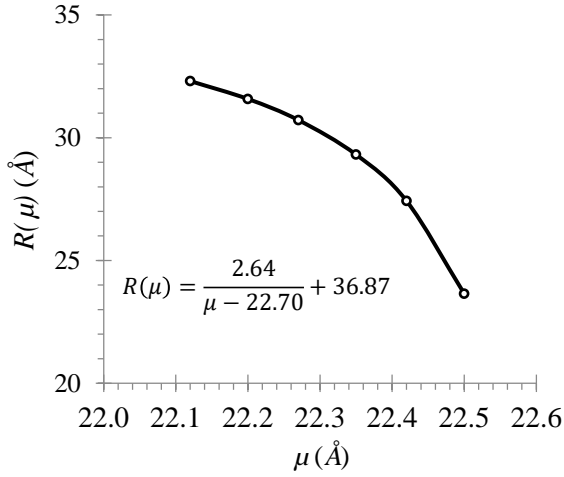


Figure 3-6: The form factor  $R(\mu)$  in modified Morse potential for  $M_{CG} = 7206.20 \text{ amu}$

Figure 3-5: Pressure  $P$  (in  $\text{bar}$ ) with the increase of  $\mu$  (in  $\text{Å}$ ) in modified Morse potential for  $M_{CG} = 720.62 \text{ amu}$

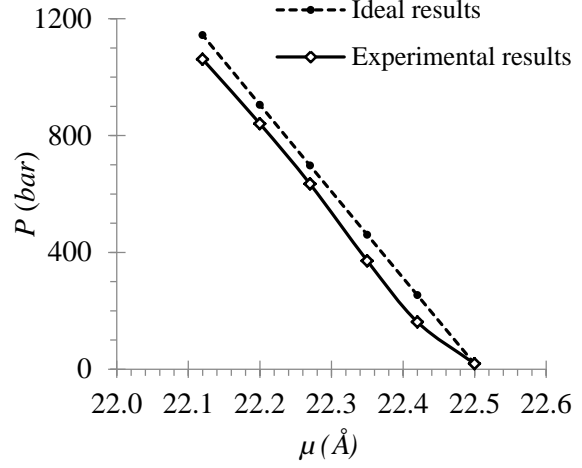


Figure 3-7: Pressure  $P$  (in  $\text{bar}$ ) with the increase of  $\mu$  (in  $\text{Å}$ ) in modified Morse potential for  $M_{CG} = 7206.20 \text{ amu}$

Table 3-3: Parameters of the modified Morse potential for various mass scales

$M_{CG}$	$\mu$	$\rho_m$	$\rho_p$	$\alpha$	$\varepsilon$	$R = a/(\mu - b) + c$			$r_{cut}$	$P$	$\kappa_T$	$\eta$
						$a$	$b$	$c$				
72.06	4.85		8.776		1.3	0.00	-	5.49	12	1.5	5.44	0.97
720.62	10.44	1.05	0.8778	10	1.4	0.12	10.53	13.23	27	1.2	4.66	0.96
7206.20	22.50		0.0878		1.6	2.64	22.70	36.87	27	18.4	4.90	0.57

Note:  $M_{CG}$  (mass of an effective CG particle,  $\text{amu}$ ),  $\mu$  (average distance,  $\text{Å}$ ),  $\rho_m$  (mass density,  $\text{g}/\text{cm}^3$ ),  $\rho_p$  (particle density,  $10^{-3} \cdot \text{particles}/\text{Å}^3$ ),  $\alpha$  (curvature control parameter, a number),  $\varepsilon$  (minimum energy,  $\text{kcal}/\text{mol}$ ),  $R = a/(\mu - b) + c$  (equilibrium distance,  $\text{Å}$ ),  $a$  ( $\text{Å}^2$ ),  $b$  ( $\text{Å}$ ),  $c$  ( $\text{Å}$ ),  $r_{cut}$  (cut-off distance of potential,  $\text{Å}$ ),  $P$  (pressure,  $\text{bar}$ ),  $\kappa_T$  (isothermal compressibility,  $10^5 \cdot \text{bar}^{-1}$ ),  $\eta$  (shear viscosity,  $\text{mPa} \cdot \text{s}$ ).

### 3.4 Results and Analysis

We tested and analyzed the model and parameters under three effective mass scales 72, 720 and 7200  $\text{amu}$ , for determining the transport coefficients through Green Kubo autocorrelation. We show structural properties by using the radial distribution functions, and validate two classic viscous flows scenarios- representing the behavior of our CG model for

blood plasma flow under various shear stress conditions: Counter-Poiseuille flow and Couette flow.

### 3.4.1 Green Kubo Autocorrelation

We calculate the GK autocorrelation in a system of 27,000 CG particles. In the simulation, the sample interval is  $\Delta s = 5$  and the time step  $dt = 15, 50$  and  $100 fs$  for  $M_{CG} = 72, 720, 7206 amu$  respectively. We test and select a proper correlation length long enough to capture the decaying behavior but not too long to add noise. Parameter  $\eta$  is the integral of correlation function as in Equation 3-4 and is presented

Figure 3-8 shows a normalized autocorrelation function. Table 3-4 summarizes the characteristic time  $\tau_v$  of the model under various CG levels.

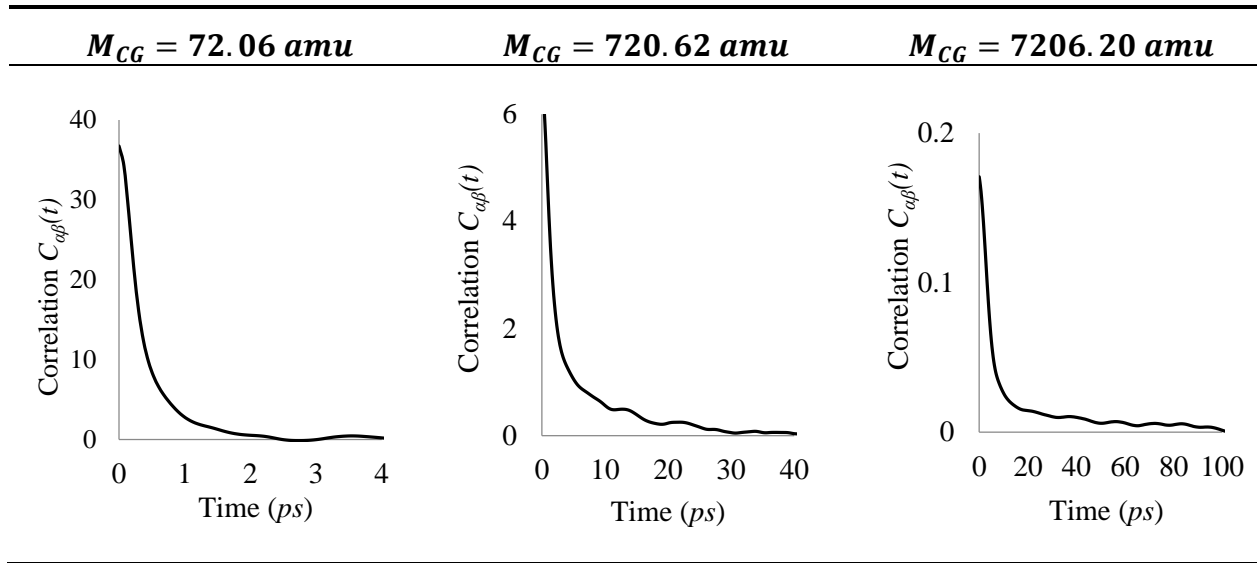


Figure 3-8: Auto stress correlation functions  $C_{\alpha\beta}(t)$  vs. time  $t$  (in  $ps$ )

Table 3-4: Characteristic time of different CG levels:  $M_{CG}$ ,  $\mu$  (average distance,  $\text{\AA}$ ),  $dt$  (timestep,  $fs$ ) and  $\tau$  (characteristic time,  $ps$ )

$M_{CG}$	$\mu$	$dt$	$\tau_v$
72.06	4.85	15	0.35
720.62	10.44	50	2.25
7206.20	22.50	100	5.00

### 3.4.2 Radial Distribution Function

We calculate the radial distribution functions (RDF) for a system of 27,000 CG particles to validate the structural properties of our model at various CG levels as shown in Figure 3-9.

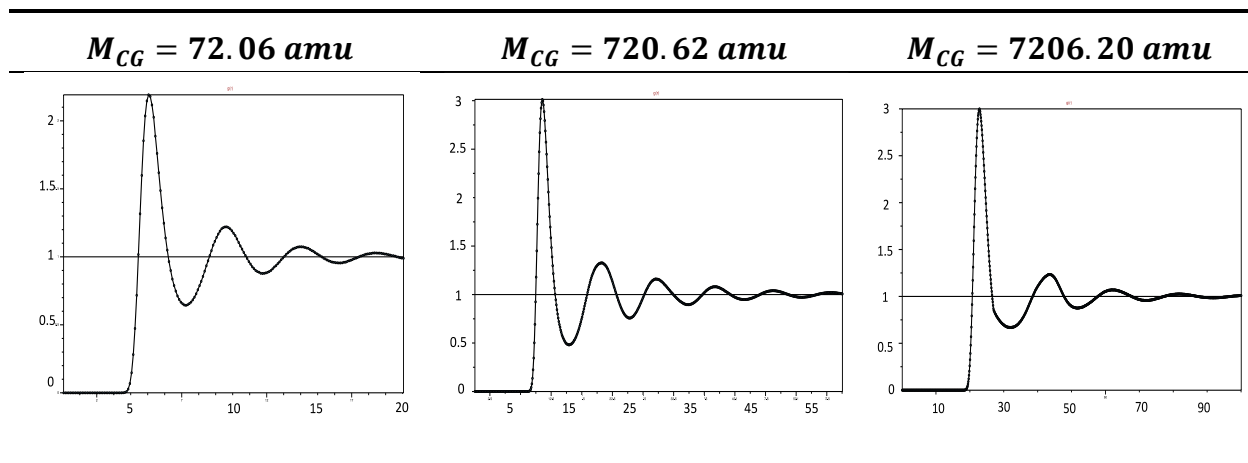


Figure 3-9: Radial distribution functions of the Morse fluids  $g(r)$  vs.  $r$

### 3.4.3 Counter-Poiseuille and Couette flows

We test our model and parameters of different CG levels to reproduce two typical viscous flows: the Counter-Poiseuille and Couette flows, and compare the simulation results with the analytical solutions.

The Counter-Poiseuille flow is tested with three effective mass scales:  $M_{CG} = 72, 720, 7206 \text{ amu}$  respectively. Figure 3-10 illustrates the simulation domain for the Counter-Poiseuille flow. This domain is divided into a left region and a right region in which two opposing forces are applied on all fluid particles with magnitudes  $0.05 \text{ kal/mol/\AA}$  in  $y$  direction. A system size of  $40 \times 20 \times 20$  CG particles with periodic boundary conditions is utilized. We have run the simulation for 200,000 simulation steps in total. After 50,000 simulation steps, we observe that the Counter-Poiseuille flow is fully developed as the velocity profile no longer changes with time. The velocity profiles of fully developed Counter-Poiseuille flows for three mass scales are shown in Figure 3-12.

Additionally, we test our model to reproduce the Couette flow with the same effective mass scales, system size and the total number of simulation steps as above.

Figure 3-11 illustrates the simulation domain for the Couette flow. This domain is divided into three regions: lower wall region, flow region and upper wall region. We apply two opposing forces with velocity magnitude of  $-3.5$ ,  $-1.1$ , and  $0.35 \text{ \AA}/ps$  respectively on all the upper and lower wall CG particles. We impose the periodic boundary conditions along the  $x$ - and  $z$ -directions and use virtual wall particles reflection method at the inner layer of  $y$ -direction walls to achieve a characteristic no-slip boundary condition at the interface of the fluid and the wall [104, 109]. Briefly, wall reflection is applied such that if a particle moves outside the wall on a certain time step by a distance  $\delta$ , the particle is dragged back by the same  $\delta$ , and the sign of the corresponding component of its velocity is flipped [46, 104]. After 10,000 simulation steps, we observe that the Couette flow is fully developed as the velocity profile no longer changes with time. The velocity profiles of fully developed Couette flows for three mass scales are shown in Figure 3-13.

The velocity profiles of fully developed Counter-Poiseuille and Couette flows are compared with the analytical solutions for all cases in Figs. 12 and 13. The analytical solution of Counter-Poiseuille is described by [110]:

$$v_y(x) = \frac{\rho g_y}{2\eta} (xD - x^2) \quad \text{Equation 3-6}$$

Here,  $v_y$  is the velocity distribution,  $\eta$  is the dynamic viscosity,  $\rho$  is the mass density, and  $g_y$  is the force.

These comparisons in Figure 3-12 and Figure 3-13 show that the velocity profiles of fully developed Counter-Poiseuille and Couette flows almost overlap with expected analytical solutions. Additionally, we observe no density fluctuations across the flow domains. This demonstrates the validity of our methodology for imposing no-slip boundary conditions and obtaining the characteristic velocity distribution in both Counter-Poiseuille flow and Couette flow benchmark solutions for a broad range of spatiotemporal scales.



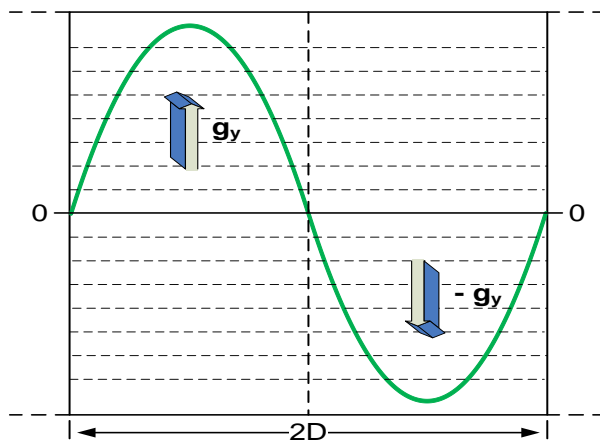


Figure 3-10: Schematic representation of the periodic Poiseuille flow

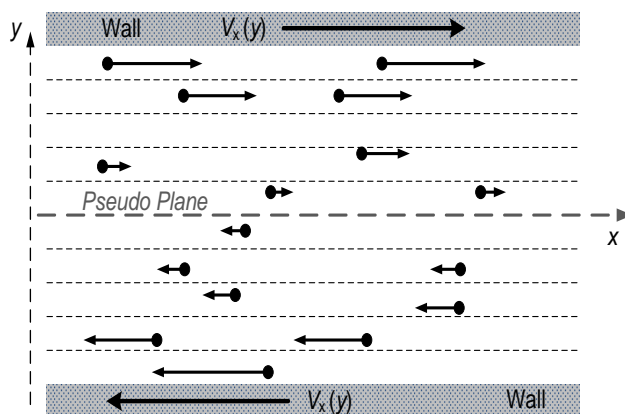


Figure 3-11: Schematic representation of the Couette flow

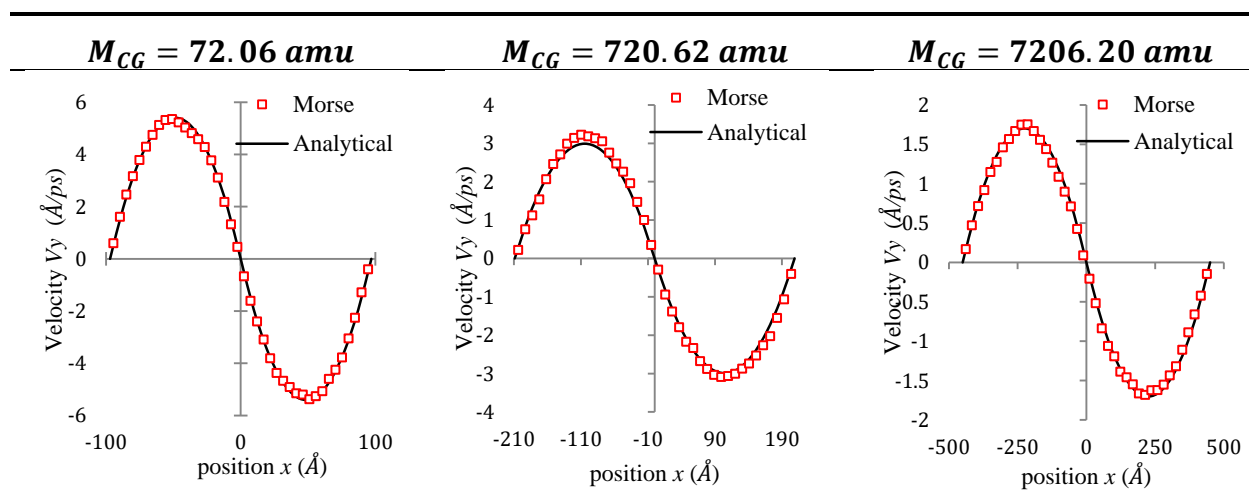


Figure 3-12: Counter Poiseuille flows velocity profiles

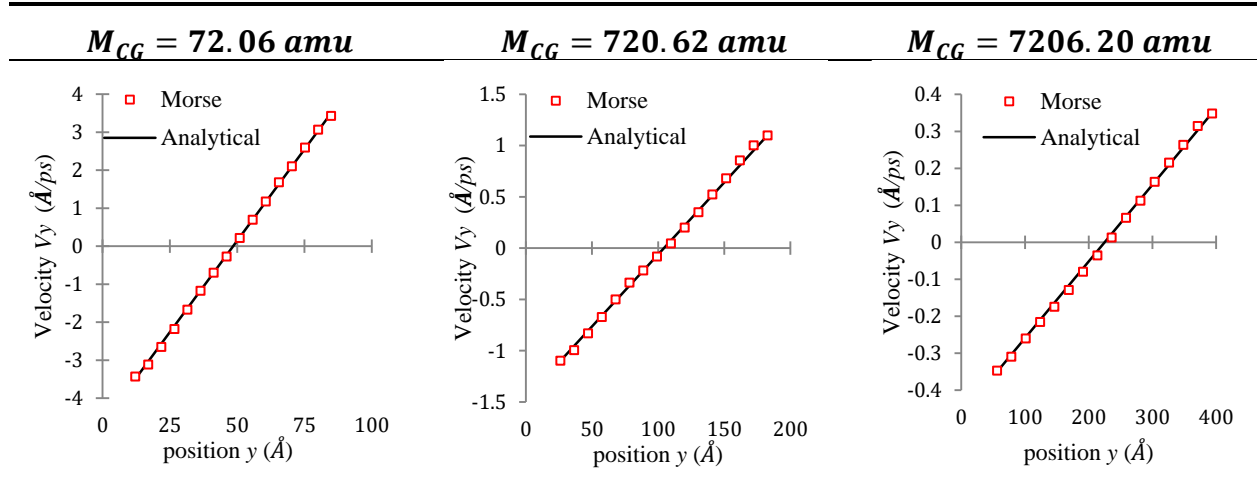


Figure 3-13: Couette flows velocity profiles

### 3.4.4 The Impact of System Sizes

We build the identical system of size  $675\text{\AA} \times 675\text{\AA} \times 675\text{\AA}$  for three coarse-graining approaches. We perform the analysis by using a radial distribution functions (RDF) and then compute the flow properties for both Couette and Counter-Poiseuille flows (as illustrated in Figure 3-9, Figure 3-12 and Figure 3-13).

Table 3-5 presents the results and analyses of the same simulation box using three different coarsening levels. These results demonstrate that the physical properties vary slightly at different coarsening levels but the characteristics of the viscous flows stay unchanged. Table 3-6 and Table 3-7 compare the results of different simulation boxes using the same coarsening level. These results demonstrate that the impact of the simulation box changes on the results including the pressure, viscosity, compressibility and radial distribution function profile is negligible.

Table 3-5: Results and analyses of the same simulation box using three coarse graining levels

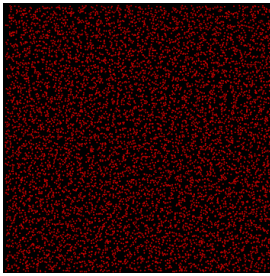
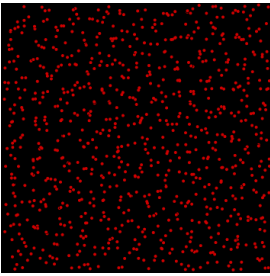
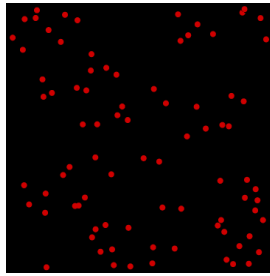
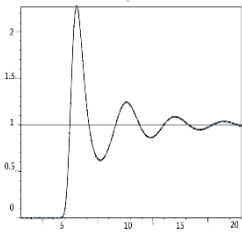
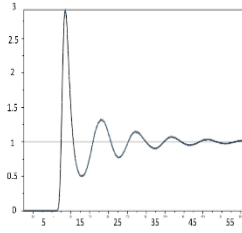
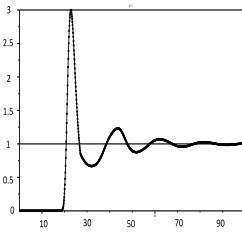
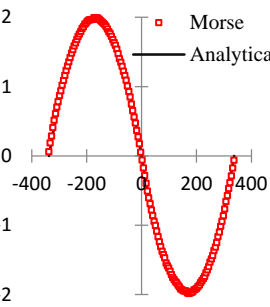
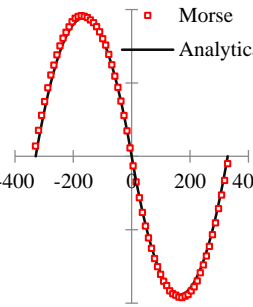
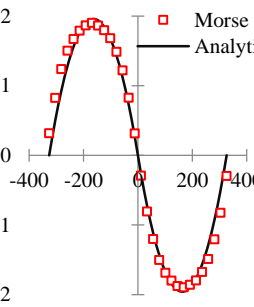
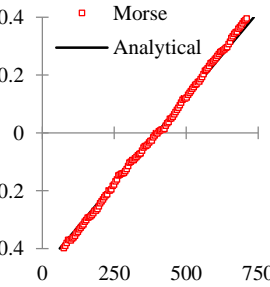
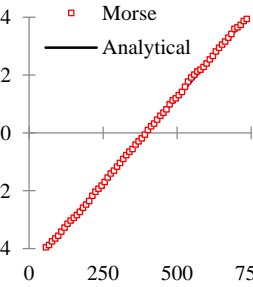
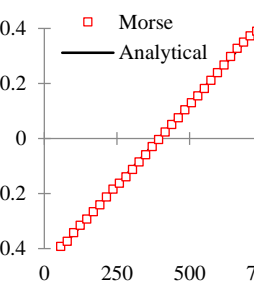
	$M_{CG} = 72.06 \text{ amu}$	$M_{CG} = 720.62 \text{ amu}$	$M_{CG} = 7206.20 \text{ amu}$
Illustration of coarse graining levels			
Number of particles	2,744,000	262,144	27,000
Compressibility ( $10^{-5} \cdot \text{bar}^{-1}$ )	5.52	4.38	4.90
Viscosity ( $\text{mPa} \cdot \text{s}$ )	0.93	0.98	0.57
Radial Distribution Function			
Poiseuille flows velocity profiles (x-axis: velocity in $\text{Å}/\text{ps}$ ; y-axis: position in $\text{Å}$ )			
Couette flows velocity profiles (x-axis: velocity $v_y$ in $\text{Å}/\text{ps}$ ; y-axis: position in $\text{Å}$ )			

Table 3-6: Comparisons of physical properties using different simulation boxes for  $M_{CG} = 72.06 \text{ amu}$

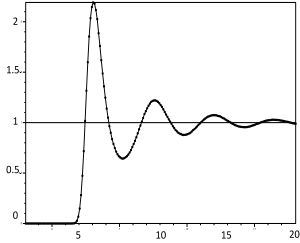
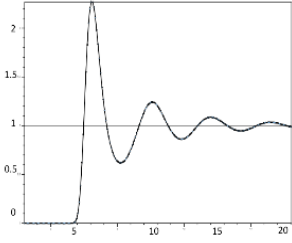
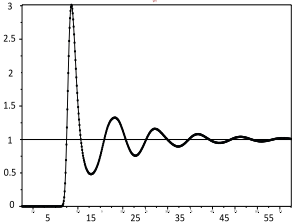
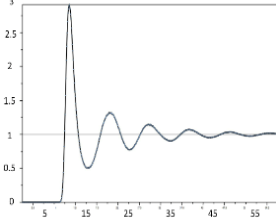
Box dimensions ( $\text{\AA}^3$ )	$145 \times 145 \times 145$	$675 \times 675 \times 675$	Target values*
Number of particles	27,000	2,744,000	
Compressibility ( $10^{-5} \cdot \text{bar}^{-1}$ )	5.44	5.52	4.6
Viscosity ( $\text{mPa} \cdot \text{s}$ )	0.97	0.93	1.1 ~ 1.3
Pressure ( $\text{bar}$ )	1.5	1.1	1.12 / 1.17
Radial Distribution Function			-

Table 3-7: Comparisons of physical properties using different simulation boxes for  $M_{CG} = 720.62 \text{ amu}$

Box dimensions ( $\text{\AA}^3$ )	$314 \times 314 \times 314$	$675 \times 675 \times 675$	Target values*
Number of particles	27,000	262,144	
Compressibility ( $10^{-5} \cdot \text{bar}^{-1}$ )	4.66	4.38	4.6
Viscosity ( $\text{mPa} \cdot \text{s}$ )	0.96	0.98	1.1 ~ 1.3
Pressure ( $\text{bar}$ )	1.2	1.2	1.12 / 1.17
Radial Distribution Function			-

\*Note: target physical properties are present in Table 3-1.

### 3.5 Application of Modified Morse Potential

We proposed an advanced multiscale platelet model for describing molecular-level mechanical properties of platelet intracellular constituents, including viscous cytoplasm, extensible cytoskeleton and bilayer elastic membrane [111]. The model is coarse grained at nanometer scales and is tested by correlating with in-vitro experiments, including biorheology of cytoplasm, the stiffness of actin filaments and the deformability of membrane. Figure 3-14 demonstrates structural constituents of the platelet model.

One of applications for modified Morse potential is used as a simulation method for modeling of platelet cytoplasm, with corroborations with *in-vitro* results for cytoplasmic biorheology. Cytoplasm is the main liquid-formed component of the platelet interior and it has crucial impact on platelet kinetics and mechanisms under flow stresses [34]. The viscosity of cytoplasm falls in the range of 4.1~23.9  $mPa \cdot s$  [4, 6]. Modified Morse with aforementioned properties can cover this range neatly and thus can be adopted for modeling the cytoplasm [39, 90]. These rheological properties of viscous cytoplasm results as shown in [111] demonstrate that Morse potential fluid favorably predicts the rheological properties of cytoplasm, offering an intracellular viscous flow condition. Moreover, Morse potential has a computational superiority over the traditional inter-atomic method by leveraging less essential degrees of freedom, resulting in an efficient method of modeling viscous biofluids at the coarser grained levels.

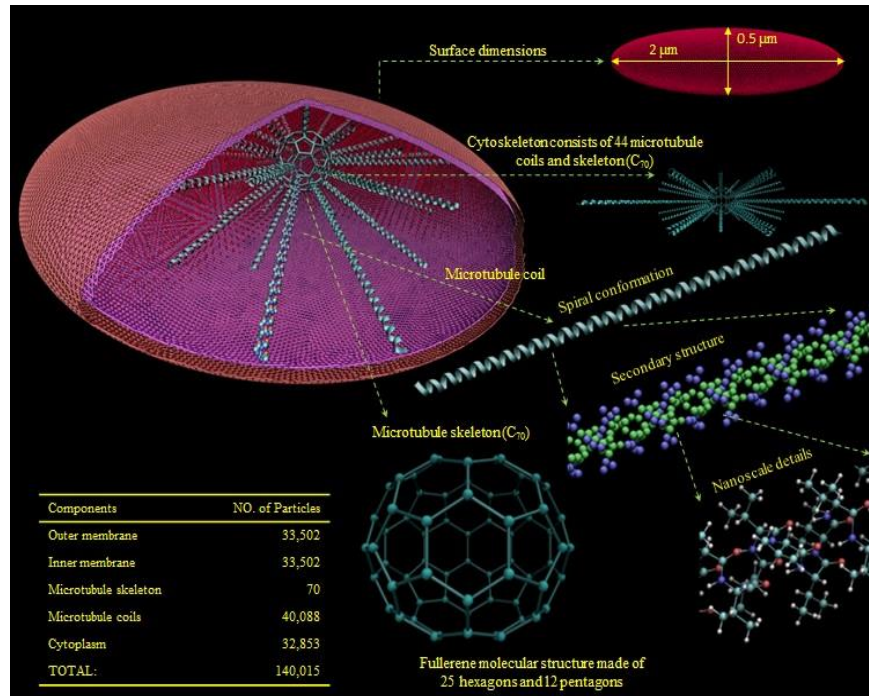


Figure 3-14: Multiscale model of human platelets: physical structures and constituents  
(This figure was adopted with permission from ref [111])

### 3.6 Summary

We modified the classical Morse potential to express the interactions of coarse-grained all-particle hemodynamics. The key contributions include the introduction of the form factor  $R(\mu)$  to the classical Morse potential, in order to enable it to cover spatiotemporal scales ranging from atomistic scales to nanoscales and to adapt it to capture the hallmarks of viscous flows dynamics. This facilitates studying complex flow mechanics such as of human blood plasma whose force field is approximated for the first time by this modified form of the Morse potential. Through extensive numerical experimentation, we obtained the parameters for three CG levels by multiple-staged methods as to parameterize the classical and modified Morse potentials. We have further studied the accuracy of our model by analyzing its transport coefficients and structural properties and validated it by reproducing two benchmark viscous flows solutions: counter-Poiseuille and Couette flows.

Future work includes further verification of our model and parameters with more complex biological fluids such as blood constituents suspended in plasma, flowing in three-

dimensional vasculature in patient specific geometries. Analyzing such complex fluids pushes the limit of continuum based numerical approaches as it requires an efficient multiscale methodology and fast parameterization scheme. Our model and methodology provides the means to realize such simulations at an appropriate CG level.

## Chapter 4 Techniques for Accelerating Simulations

### 4.1 Demand of Tasks

Supercomputer powers are widely used to model simulations, carry out brute force calculations or perform large data processing [112]. For example, molecular dynamics simulations by tracing atom motions of biological molecules enable studies of molecular mechanisms that are unobservable or immeasurable in practical laboratory settings [2, 113-116]. Bio and life sciences, such as genomics, proteomics, pharmacology, etc., are the third-largest market segment for HPC usage [117]. However, such simulations exhibit major challenges, e.g., inefficiency of parallelization, dramatic rise in volume and availability of data, diverse spatiotemporal scales of mechanisms happen and so on. Yet multiscale simulations are emerging as powerful tools for high-resolution low-cost modeling of clinical problems. Examples include the multiscale models for systems with elementary particles and biomolecules [118, 119] and various studies of vascular diseases [120-122].

As for our demands, we seek to study flow-induced platelet activation which may initiate coagulation cascade of blood and prompt blood clotting formation in prosthetic cardiovascular devices and in arterial diseases processes. Our simulation needs include millions of particles, milliseconds or longer multiscale phenomena, complex particle-particle or cell-cell interactions as well as thermodynamic control and analysis.

In order to understand a priori scalability and time-to-solution, we extend our previous efforts for modeling platelets under viscous shear flow conditions at multiple spatial-temporal scales [56, 111, 123-126] and develop this multiscale model as a benchmark test for predicting the problem-specific capability of supercomputers for solving multiscale phenomena of multi-component biological systems.

Figure 4-1 shows the scheme of this multiscale benchmark. The diversified force fields at different scales escalate computational complexities. The nanoscale platelet model is described in section 3.5 and mesoscopic flow is described in section 2.2.1. The wall-driven Couette benchmark flow is yielded to flip the deformable platelets. The velocity field of Couette flow is reproduced by applying two opposing momenta with velocity magnitude of 1.5 (dimensionless)



to the upper and lower walls in  $y$  dimension, which causes a uniform shear stress with magnitude of  $8.57 Pa$  in real units. This is designed to emulate the shear stress range of stenotic vessels where thrombogenicity is most likely to happen [127]. In this model, the microstructure of flowing platelets changes in response to the extracellular hemodynamic stresses and in the meantime the complex intracellular components including cytoplasm and cytoskeleton are coarsened to describe the intra-platelet microenvironment at the nanoscale. Through correlating with *in-vitro* observations, the phenomena of spontaneous filopodia formation could be mimicked [128]. This numerical model could be extended to simulate the processes involving in platelet activation leading to thrombosis, offering a practical numerical solution.

Three problem sizes of our multiscale simulations are selected as benchmarks (summarized in Table 4-1):

- (1) The smallest system contains 680,718 particles, forming a single platelet that flows in a viscous flow. In the system, the platelet and flow sub-models contain 140,015 (21%) and 540,703 (79%) particles, respectively. This system is used for parameterization and observation of micro morphological changes for platelet;
- (2) The middle system contains 2,722,872 particles forming 4 platelets. This system is suitable for validating the inter-platelet and platelet-vessel interactions;
- (3) The largest system contains 10,891,488 particles forming 16 platelets. This system helps study the initial stage of blood clotting processes involving multiple platelets.

Table 4-1: Problem sizes and dimensions of multiscale benchmarks

<b>Experiments</b>	<b># of Platelets</b>	<b># of Particles</b>	<b>Dimensions</b>
Exp-S	1	680,718	45×90×45
Exp-M	4	2,722,872	90×90×90
Exp-L	16	10,891,488	180×90×180

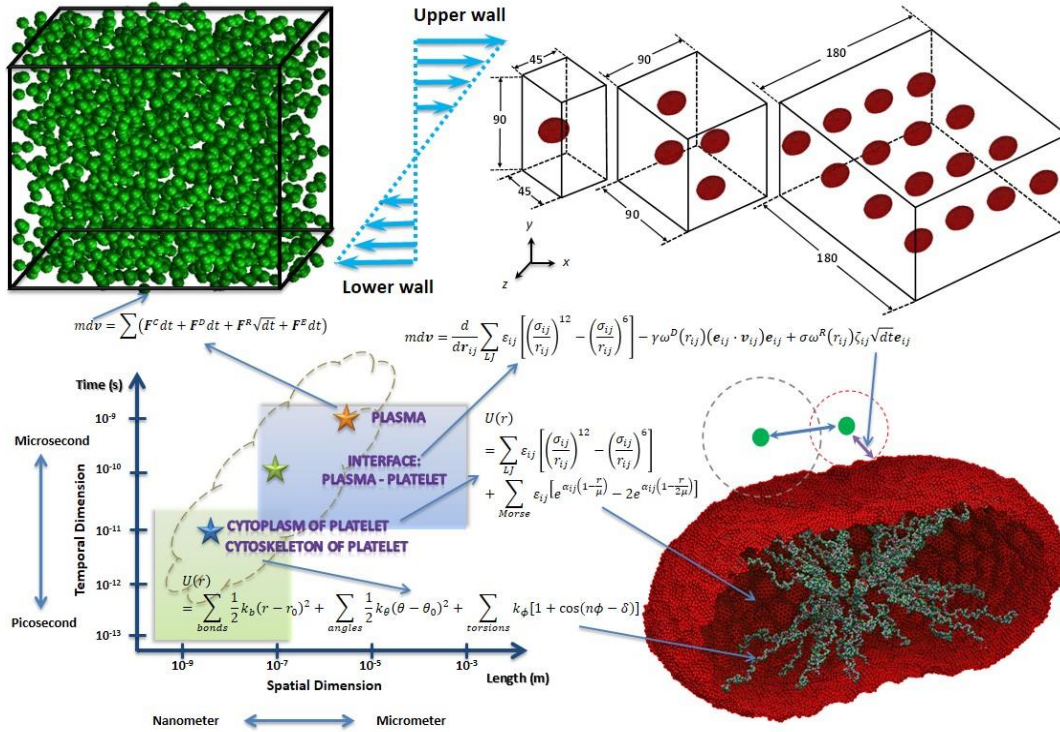


Figure 4-1: Multiscale modeling framework for simulating platelets in blood flows

## 4.2 Supply of Supercomputer Resources

Nowadays, the top four systems, i.e., Tianhe-2 (MilkyWay-2), Titan, Sequoia, and K computer on the Top500 release list (Nov 2015), demonstrate higher than 10 Petaflops sustained performance. As the era of increasing computer performance simply by increasing CPU clock speeds or packing more dense processors into small space as much as possible ends, the HPC community is exploring development efforts for breaking the Exaflop barrier [112, 129, 130]. Interconnection networks are developed to support much larger number of processors at the expense of increased engineering complexities and programming difficulties [32, 80, 131, 132]. Accelerators are used for boosting floating-point performance, though it further escalates the difficulty of programming [133-135]. Another dimension of performance supply is driven by cloud platforms [136]. HPC cloud services can component existing solutions, with advantages of higher utilization of resources, economy of scales, and operational efficiency. The availability of HPC cloud services allow researchers to have access to computing resources immediately when they need without waiting in queue, which can greatly improve the time-to-result efficiency.

We implemented our multiscale benchmarks on three platforms: Tianhe-2 (No. 1 in TOP500 from Jun 2013 until now), Stampede (No. 7 in TOP500 from Nov 2012 through Nov 2014) and Cray CS-Storm, a high-density multi-accelerator system. The CS-Storm system is used in the No. 10 computer of the Nov 2014 release of TOP500 with NVIDIA K40. In this work, we tested the CS-Storm system with K40 and K80, which is the latest NVIDIA GPU generation so far. Table 4-2 summarizes the nodal configurations and peak performance (*GFlops*) of the three systems.

- (1) Tianhe-2 supercomputer, first released in 2013, is powered by 3.12 million cores from 16,000 compute nodes. Each node has two Intel Ivy Bridge processors (Xeon E5-2692 v2 12C 2.2GHz) with 64GB of memory, providing 422.4 *GFlops* per dual socket node. Each node also has three Intel Phi coprocessors, providing 3 *TFlops* per node. The processing nodes are connected by the TH Express-2 customized interconnect networks [137].
- (2) Stampede supercomputer, first released in 2012, is powered by 0.46 million cores from 6,400 compute nodes. Each node has two Intel Sandy Bridge processors (Xeon E5-2680 8C 2.7GHz) with 32GB of memory, providing 345.6 *GFlops* per dual socket node. Each node also has one Intel Phi coprocessor, providing 1 *TFlops* per node. These nodes use the FDR InfiniBand network in a 2-level fat-tree topology. The Tianhe-2 single-node peak-performance is 2.54 times that of Stampede.
- (3) CS-Storm is a high-density multi-accelerator system and it can integrate up to eight GPU cards, e.g., incorporate 8 K40m or 16 K80 where K80 is a dual GPU accelerator. A single CS-Storm 8-K40m/16-K80 server provides the double-precision (single-precision) peak performance of 11.4 / 15 (34.3 / 44.8) *TFlops* respectively. The memory is 12 GB per K40/K80 GPU. Quite obviously, the CS-Storm node is much more powerful than those of Tianhe-2 and Stampede. We tested two configurations of the CS-Storm node: one with 8 NVIDIA K40m GPUs and 2 Intel Xeon E5-2670 v2 2.5GHz CPUs (264GB memory) and the other with 16 K80 GPUs and 2 Intel Xeon E5-2680 v3 2.5 GHz CPUs (132GB memory).

Table 4-2: Nodal configurations and peak performance (*GFlops*) for Tianhe-2, Stampede and CS-Storm

<b>Systems</b>	<b>Nodal Configuration*</b>	<b>Nodal Perf. (<i>GFlops</i>)</b>
Tianhe-2	24 cores	422.4
Stampede	16 cores	345.6
Stampede (K20)	16 cores + K20	1,515.6
CS-Storm (K40)	20 cores + 8 K40	11,840.0
CS-Storm (K80)	24 cores + 16 K80	15,440.0

\*Note: We only consider CPUs and GPUs here.

### 4.3 Mapping and GPGPU Acceleration

The parallelism in our simulations follows spatial decomposition, i.e., each processor is assigned a portion of physical domain [46]. Each processor computes the forces on atoms in its sub-domain. Each process exchanges atoms with its neighboring processes as atoms on the boundary are likely to move from one sub-domain to another. The spatial decomposition mappings for each system size are presented in Table 4-3. Boundary conditions in  $(x, y, z)$  directions are  $(p\ s\ p)$ . For the non-periodic  $y$  direction, we referenced non-slip boundary condition for the DPD fluid [25].

Table 4-3: Mappings of each system size

<b>Exp-S</b>		<b>Exp-M</b>		<b>Exp-L</b>	
<b>Number of Processes</b>	<b>Parallel Partition</b>	<b>Number of Processes</b>	<b>Parallel Partition</b>	<b>Number of Processes</b>	<b>Parallel Partition</b>
32	2×4×4	32	4×2×4	32	N/A
64	4×4×4	64	4×4×4	64	4×4×4
128	4×8×4	128	4×4×8	128	8×2×8
256	4×8×8	256	4×8×8	256	4×8×8
512	8×8×8	512	8×8×8	512	8×4×16

The performance of simulations in general can be fueled up by two thrusts. The first is to develop new algorithms. The second is to utilize accelerators with the rapidly growing performance. The MTS algorithm reduces computational costs but without lowering the communication costs proportionally. The basic idea of a multiple time stepping scheme is to

compute the more costly non-bonded and long-range forces only once per long time step. In each step, there are two kinds of inter-processor communication: (1) the atom coordinates exchange after they are updated and (2) the forces exchange after every force evaluation. The latter is reduced at the same rate at which the force evaluation is reduced by the MTS algorithm and this communication can be eliminated if Newton's 3rd law is ignored among neighbor bin. Unfortunately, the former cannot be reduced because the updated atom coordinates must be exchanged in the innermost loop in order to update velocities for the next step. When computation time is reduced, the unchanged communication costs, resulting in a higher communication-to-computation ratio, start to ruin scalability. A way out of this dilemma is the increased computational workloads for local computation and utilization of accelerators to alleviate local computation.

In order to successfully simulate the minimum requirement of 1- *ms* multiscale phenomena, we exploited the double-punch speedup strategy, i.e., combination of algorithmic MTS and hardware GPU acceleration. MTS (section 2.3) that help improve the efficiency of integrations while maintaining stability and accuracy (section 2.4). GPGPU acceleration can help calculate the computation intensive kernels thus over-take enormous computation loads from CPU.

Numerically, all our simulations were performed with LAMMPS [46] and LAMMPS GPU package[138-140]. In LAMMPS GPU package implementation, for each time step, it moves particle data back-and-forth between GPU and CPU. It enables neighbor lists building and force calculations to be accelerated on GPU. It allows force computations to be performed on GPU in three precisions: single or double or mixed, where pairwise forces are computed in single precision, but accumulated at double-precision. Performance benefits depend on the hardware configurations and simulations. Additionally, we implemented tailored modifications (based on LAMMPS version Oct 2014), including:

- (1) No-slip boundary condition between the flows and blood vessels (for CPUs) [58];
- (2) Multiple time stepping algorithm (for CPUs and GPUs) [124];
- (3) DPD-CGMD hybrid force field (for CPUs and GPUs) [123].

It should be noted that though LAMMPS starts to support some functions on Intel Xeon Phi coprocessors recently, the code is still unavailable to support all of necessary functions on the co-processors for our multiscale simulation, such as DPD pair style.

## 4.4 Performance Examinations

We performed our detailed performance examinations by two stages:

Stage 1: Test the impact of double-punch acceleration strategy, i.e., the algorithmic MTS and hardware GPGPU acceleration to see how much improvement can be achieved by combining the acceleration techniques.

Stage 2: Test the impact of different supercomputer architectures to investigate the mapping strategies.

For stage 1, we examined different MTS configurations as shown in Table 4-4. Similar to section 2.3, there are four-level integrators:  $\Delta t_1$  for the molecular bonded level,  $\Delta t_2$  for the molecular non-bonded level,  $\Delta t_3$  for the spatial interface level, and  $\Delta t_4$  for the DPD fluid level. The only difference here is that MTS is applied to advanced platelet model (section 3.5) based on modified Morse potential. The four-level integrators can be adjusted via three loop factors ( $K_1$   $K_2$   $K_3$ ). Adjusting the step sizes will impact computational performance but also simulation precision. The balance of precision and speed must be considered in terms of specific interested properties. As discussed earlier in section 2.4, for multiscale platelets modeling, there are two kinds of properties considered: non-chaotic and chaotic properties. Non-chaotic properties, such as temperature, pressure and total energy of whole system, are not sensitive to choices of step sizes. While chaotic properties, such as the kinetic energy of platelet and stress distribution on the platelet membrane mapped from fluid can be susceptible to aggressive step sizes. Here, the percentile errors in terms of most chaotic property (kinetic energy of platelets) are well prevented and converged in all MTS cases in Table 4-4. Besides the algorithmic MTS, we also examined the impacts of incorporating GPGPU accelerations. The stage-1 study is only performed on Stampede supercomputer for two problem sizes: Exp-M and Exp-L.

Table 4-4: The timesteps and configurations for each MTS test case

Case	Time Steps For Each Scale*				Configurations		
	CGMD-BD $\Delta t_1 \times 10^{-6}$	CGMD-NB $\Delta t_2 \times 10^{-6}$	DPD-CGMD $\Delta t_3 \times 10^{-6}$	DPD $\Delta t_4 \times 10^{-6}$	$K_1$	$K_2$	$K_3$
Case A	2.5	2.5	25.0	500.0	1	10	20
Case B	5.0	5.0	50.0	1000.0	1	10	20
Case C	5.0	5.0	50.0	500.0	1	10	10
Case D	10.0	10.0	100.0	500.0	1	10	5
Case E	10.0	10.0	100.0	1000.0	1	10	10
STS	1.0	1.0	1.0	1.0	1	1	1

\*Note: please note the time step values are in dimensionless units. The coherent physical units for scaling dimensionless quantities refer to [111]. By unit conversion, 1 dimensionless time unit represents  $2.08 \mu\text{s}$  physical time.

Stage 2 tests were performed on all three aforementioned architectures. We reduced one dimension by fixing MTS configuration with Case C, which the step sizes for the flow regime (DPD) and platelet-flow interface (DPD-CGMD) in MTS are  $5 \times 10^{-4}$  and  $5 \times 10^{-5}$ , and the smallest step size for intra-platelet force fields is  $5 \times 10^{-6}$ . Here case C and STS are labeled as `mts` and `no_mts` for clarity.

The timing output from the LAMMPS code is used as the final timing results. Using these timing results, we compute the simulation speed in units of: (1)  $\text{days}/\mu\text{s}$ , i.e., the number of wallclock days it takes to complete  $1\text{-}\mu\text{s}$  multiscale phenomena, for showing the days-per-microsecond performance rate; (2)  $\mu\text{s}/\text{day}$ , i.e., the number of microseconds one physical day's simulation can achieve, for demonstrating the microseconds-per-day rate. Then we compute the performance improvement percentage of `mts` over `no_mts` using the formula:

$$P(t_{mts}, t_{no\_mts}) = (t_{no\_mts} - t_{mts})/t_{no\_mts} \quad \text{Equation 4-1}$$

where  $t_{mts}$  and  $t_{no\_mts}$  are the timing results in  $\text{days}/\mu\text{s}$  for `mts` and `no_mts` algorithms. For detailed analysis, we present the communication-to-computation ratio. The communication uses the `comm` timing and the computation is the sum of the `pair` and `bond` timing. In all tests, we notice that the output timing is almost zero since most outputs are disabled and the `neigh` timing is relatively small, less than 5%.

## 4.4.1 The Impact of MTS and GPGPU Acceleration

### For Single Time Stepping Algorithm

The simulation speeds for STS ranging from 32 to 512 MPI processes are shown in Figure 4-2. For Exp-M, due to memory limitation of processor on Stampede, minimum number of processes to start the simulation is 64. As shown in Figure 4-2, the speeds are  $0.85 \text{ day}/\mu\text{s}$  and  $1.29 \text{ day}/\mu\text{s}$  with optimal 256 MPI processes for Exp-M and Exp-L respectively. The multiscale phenomena we aim to simulate such as platelet-related reactions including platelet activation and platelet aggregation usually happen at millisecond scale. In order to complete 1-*ms* simulation, Exp-M needs 848 days and Exp-L needs 1290 days with 256 MPI processes, which are obviously impractical using merely the STS algorithm.

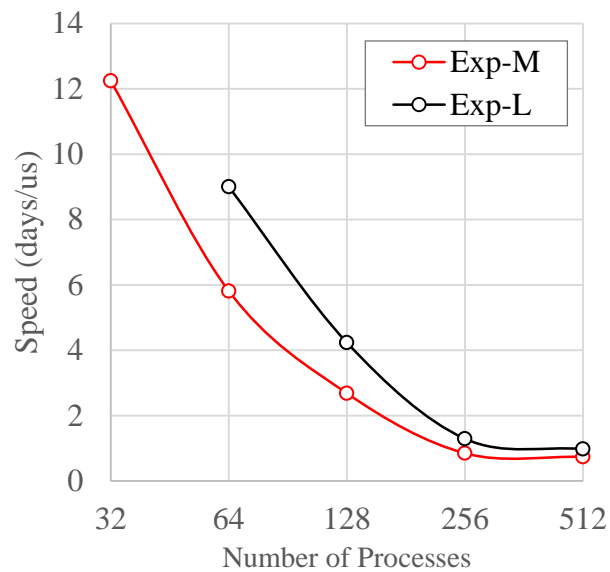


Figure 4-2: Parallel speeds of STS simulations

### For Multiple Time Stepping Algorithm

The absolute parallel speeds using MTS algorithm with different jump configurations are shown in Figure 4-3 (Exp-M) and Figure 4-4 (Exp-L). Speedups of different MTS cases over STS are shown in Figure 4-5 and Figure 4-6. These results reveal significant performance gains for all MTS cases. For 64 MPI processes, the most aggressive case (Case E) has 73 and 60 folds of computing time reduction for Exp-M and Exp-L and the most conservative case (Case A) has



18 and 15 folds of computing time reduction for Exp-M and Exp-L. Case E in general is four times faster than Case A because all time steps in Case E doubles those of Case A. In addition to the time steps chosen, the improvements of MTS algorithm are also sensitive to scalability of this problem. As the number of processes increases, the relative speedups of MTS over STS decrease. For 256 MPI processes, Case E only has 11 and 15 times reduction over STS for Exp-M and Exp-L and Case A only has 3 and 4 times reduction for Exp-M and Exp-L. The timing ratios of communication over computation are shown in Figure 4-7 (Exp-M) and Figure 4-8 (Exp-L). All MTS cases and STS case show a general trend: the communication time dominates as the number of processes increases. But compared with STS, the ratios of communication are much higher for MTS. For MTS, compared with Exp-L, the ratios of communication are much higher for Exp-M. The computation costs including pair, bond, angle, and dihedral force field evaluation were greatly reduced by MTS. As such, the ratios of communication to computation elevate.

### **For GPGPU Acceleration**

The GPU-enabled performance tests were conducted only for Exp-M with 16 and 32 GPU nodes, due to the memory limitation of K20 on Stampede supercomputer. NVIDIA K40 and K80 GPU cards with larger memory can remedy this problem. Figure 4-9 and Figure 4-10 show the speeds of GPU and their comparisons with the same number of nodes using CPU-only. Using 16 GPU nodes (each node has two 8-core Xeon E5 processors and 1 GPU K20 card), we obtain a total speedup of 23 over STS. The MTS algorithm is around 11.5 times faster than the STS algorithm with 16 nodes (256 MPI processes); the GPU-enabled system help gain the remaining speedup of 2 over the CPU-only system. In total, simulation of 1-*ms* multiscale phenomena of flowing platelets in blood vessels can be completed within approximate 37 days instead of 848 days, allowing practical modeling of millisecond-scale biological phenomena with spatial resolutions at the nanoscales.

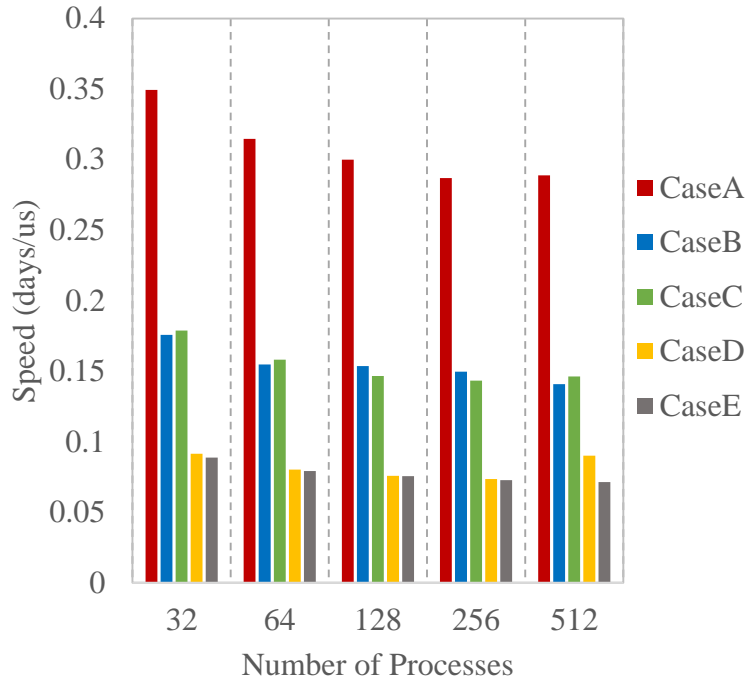


Figure 4-3: Parallel speeds of MTS cases for Exp-M

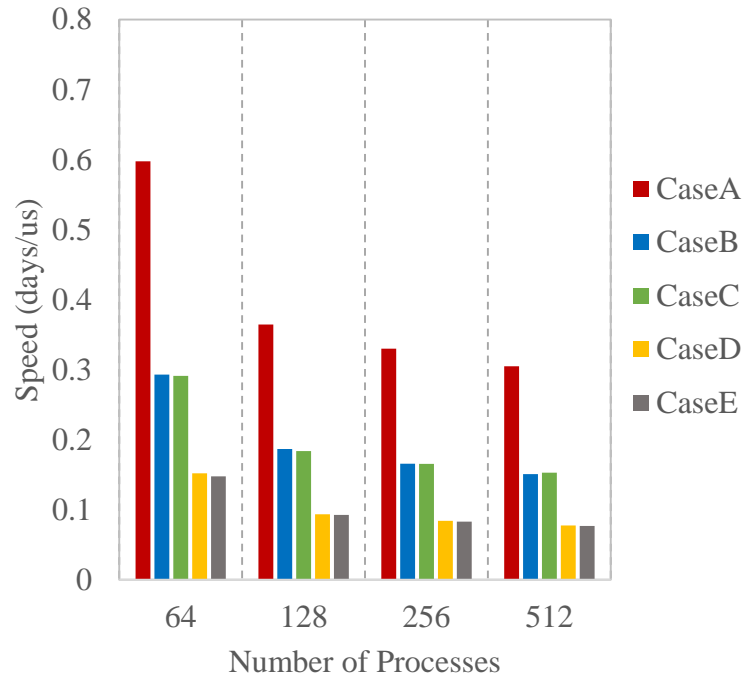


Figure 4-4: Parallel speeds of MTS cases for Exp-L

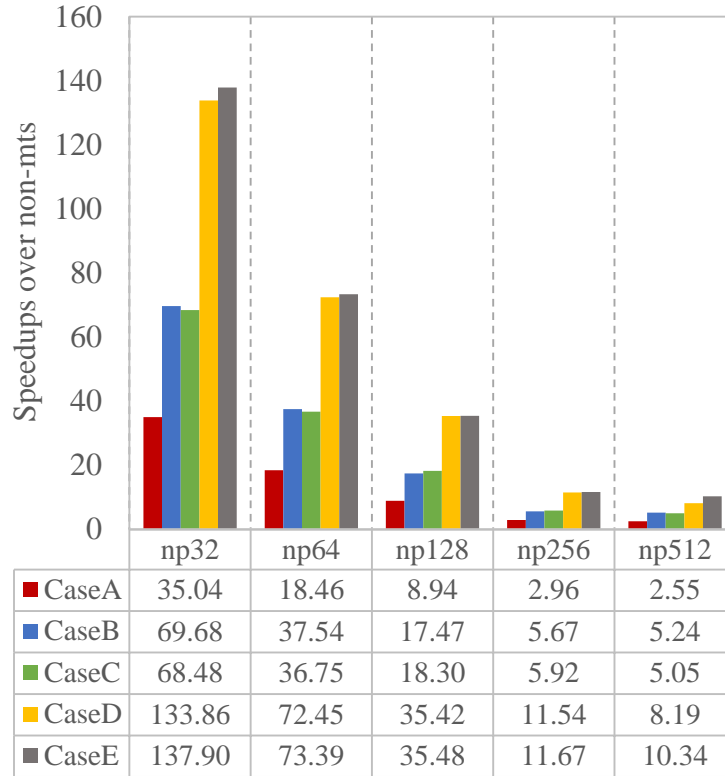


Figure 4-5: Speedups of MTS cases over STS case for Exp-M

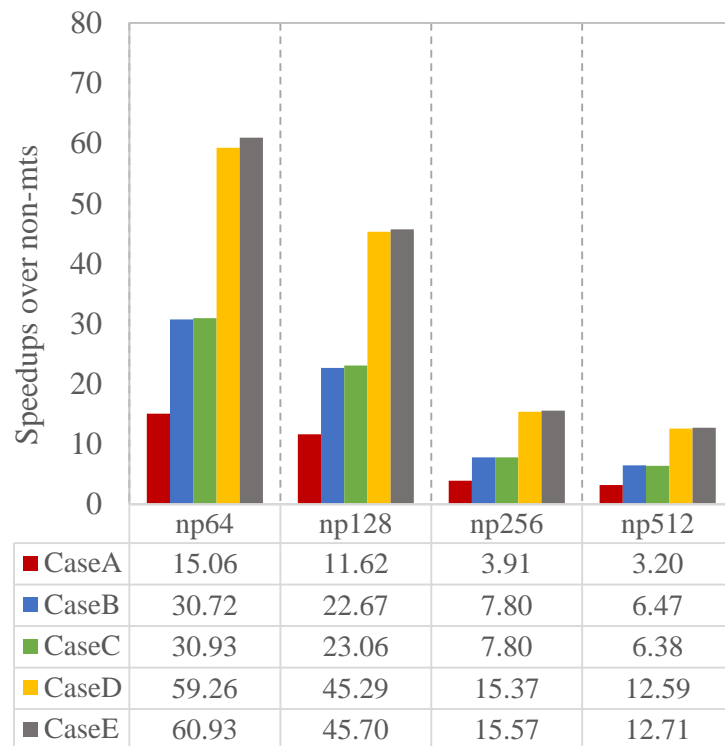


Figure 4-6: Speedups of MTS cases over STS cases for Exp-L

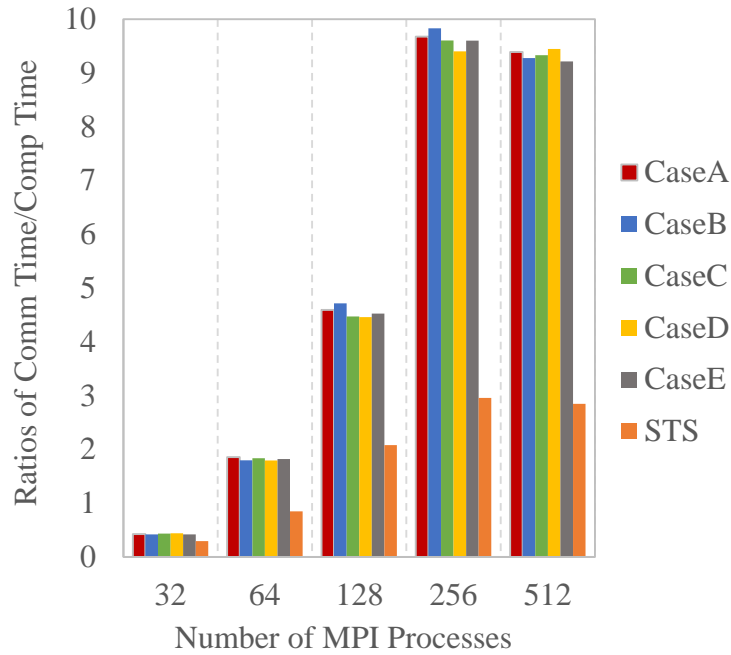


Figure 4-7: The ratios of communication time over computation time for all MTS cases and STS for Exp-M

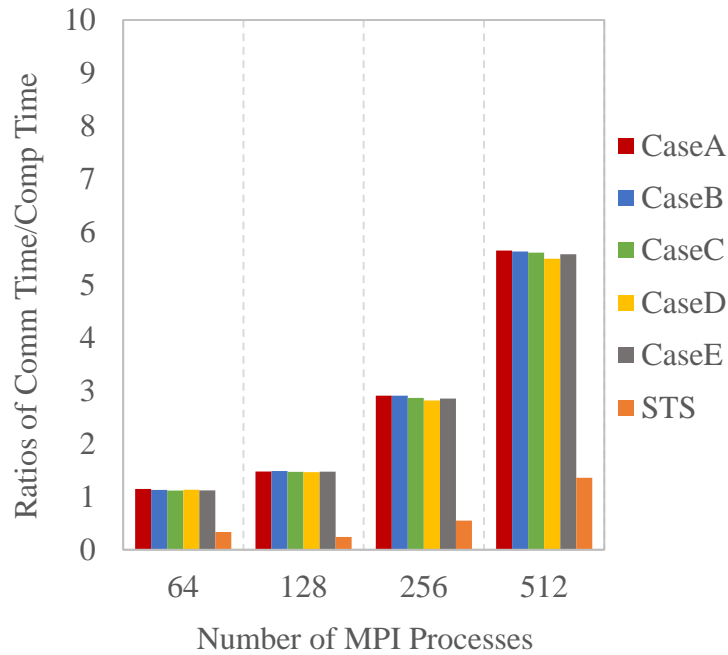


Figure 4-8. The ratios of communication time over computation time for all MTS cases and STS for Exp-L

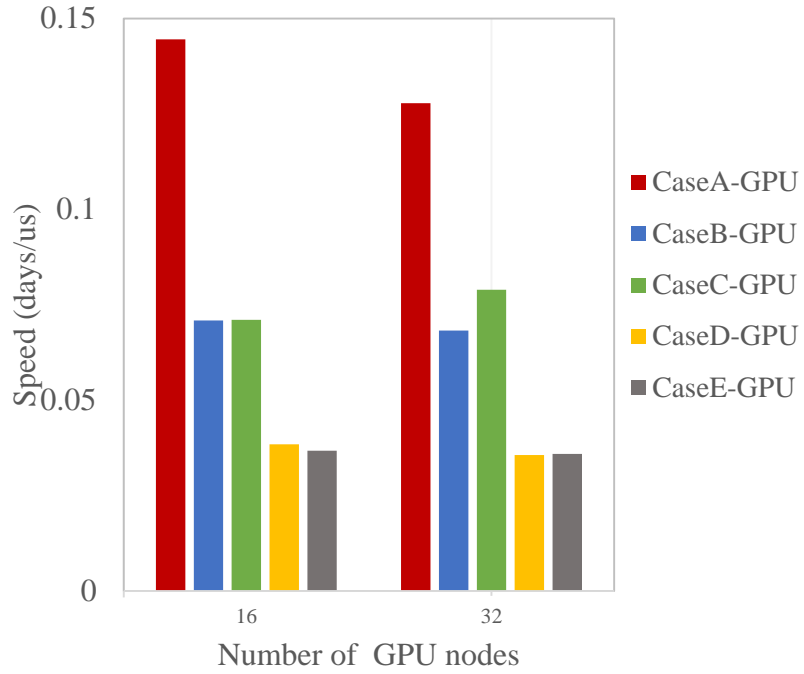


Figure 4-9: Parallel speeds of GPU-enabled MTS cases for Exp-M

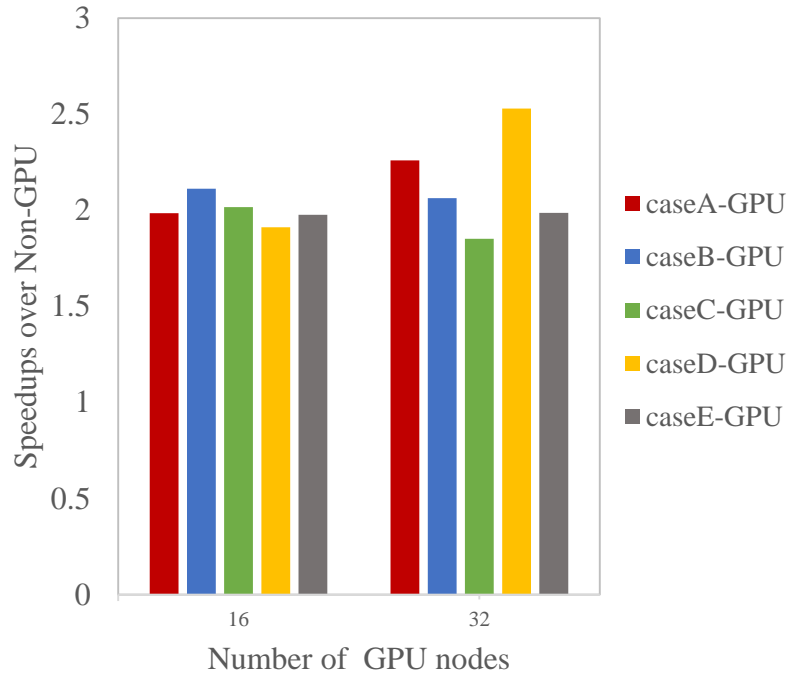


Figure 4-10: Speedups GPU over CPU-only for different MTS cases for Exp-M

## Analysis

The double-punch acceleration strategy, i.e., the algorithmic MTS and hardware GPU acceleration, can significantly improve the overall performance of multiscale simulations. Simulating 1- $\mu$ s multiscale phenomena can be reduced to 0.037 day by MTS from 0.848 day using STS, as shown by Case E configuration with 16 GPU nodes. Thus simulating 1-*ms* multiscale phenomena can be achieved within 37 days from 848 days.

MTS algorithms, naturally, reduce computational costs without lowering the communication costs proportionally. For ten-million-particle level system (Exp-L), even higher speedups of MTS cases over STS due to lower communication-to-computation ratios can be achieved. As the number of particles per process reduces, a critical point will be reached where it is more efficient to run on few cores due to the escalating communication costs. Compared absolute speeds, MTS moves faster to that critical point than STS. For example, the critical point for STS is 256 MPI processes while for MTS CaseE shows poor scalability at early 32 for Exp-M and 128 for Exp-L. The basic idea of a multiple time-stepping scheme is to compute the more costly non-bond and long-range forces only once per long time step. In each MD step, there are two kinds of inter-processor communication [46]. The first is the atom coordinates exchanges after they are updated and the second is the force exchanges after every force evaluation. The second communication is reduced at the same rate at which the force computation is reduced by the MTS algorithms. Unfortunately, the first communication cannot be reduced because the updated atom coordinates must be exchanged in the innermost loop in order to update velocities for the next step. When computation time has been saved, the unchanged communication costs start to ruin scalability. The second communication can be eliminated if Newton's 3<sup>rd</sup> law is ignored among neighbor bin. In that case, force evaluation will require more computing that the MTS algorithms can afford to bear. In this multiscale platelets modeling simulation, Newton's 3<sup>rd</sup> law has to be applied because the DPD pair style needs the Newton pair for momentum conservation thus the second communication cannot be further reduced.

The mechanism of GPU computing for molecular dynamics by the implementation of LAMMPS GPU is to port force calculations to accelerators and build neighbor lists optionally at CPU-host or GPU-device. Thus, it also suffers the communication bottleneck since the first communication issue has not been addressed. However, GPU can over-take enormous

computation loads from CPU and the emerging high-density multi-GPU supercomputers will definitely lead the way for modeling larger-particle systems [141].

## 4.4.2 The Impact of System Architectures

### For System Architectures

Performance results are presented in Figure 4-11 through Figure 4-22. Every three successive figures present the performance metrics of one supercomputer, which are the simulation speed in *days/μs*, the performance improvement for MTS over STS and the communication-to-computation ratio. The supercomputers are presented at the order: Tianhe-2, Stampede, CS-Storm 8-K40m and 16-K80. For Tianhe-2 and Stampede, the horizontal axis is the number of processor cores (referred to as Number of CPUs) in Figure 4-11 through Figure 4-16. For CS-Storm systems, the horizontal axis is the number of accelerators (referred to as Number of GPUs) in Figure 4-17 through Figure 4-22. These results for speeds vs. number of CPUs/GPUs show the strong scaling of multiscale simulations, in which the problem size is fixed while the number of processing elements (CPU/GPU) is increased. The weak scaling is not applicable since the problem size has to be determined by the biological nature (i.e., number of platelet and platelet density) so it is hardly tailored to offer the fixed workload per processing element.

Constrained by limited memory per compute node, the multiscale simulations failed to run on a small group of nodes. This is common for million-scale complex simulations. Thus, we begin our tests with 16 cores for Exp-S/Exp-M and 64 cores for Exp-L on Tianhe-2 and Stampede. Since the improvement of simulation speeds diminish with the rapid increase of nodes (Figure 4-2), we stop the tests until 512 cores are tested.

For the CS-Storm tests, we use 16 CPU cores and vary the number of GPUs. Similarly, constrained by the limited memory of accelerators, the Exp-M failed to run on single GPU and the minimum numbers for K40m/K80 for the Exp-L with no\_mts are eight. However, the minimum requirements for the Exp-L with mts are 8 K40m and 16 K80 GPUs since the mts needs more memory to store multi-level force vectors than the no\_mts.

Upon these results, the simulation speeds and the communication-to-computation ratios for Exp-L are compared between Tianhe-2 and Stampede in Figure 4-23 and Figure 4-24, respectively. The best performances in *days/μs* and *μs/day* are presented in Table 4-5 Table 4-6 and Table 4-7 for the Exp-S, Exp-M and Exp-L. Moreover, we extend the tests for Exp-S on Stampede K20-enabled nodes. Together with Exp-M results from section 4.4.1, we present the results in Table 4-5 and Table 4-6 respectively. Exp-L failed to run on K20-enabled nodes due to the memory deficiency. Table 4-8 is a summary of simulation speed of on three architectures for three problem sizes.

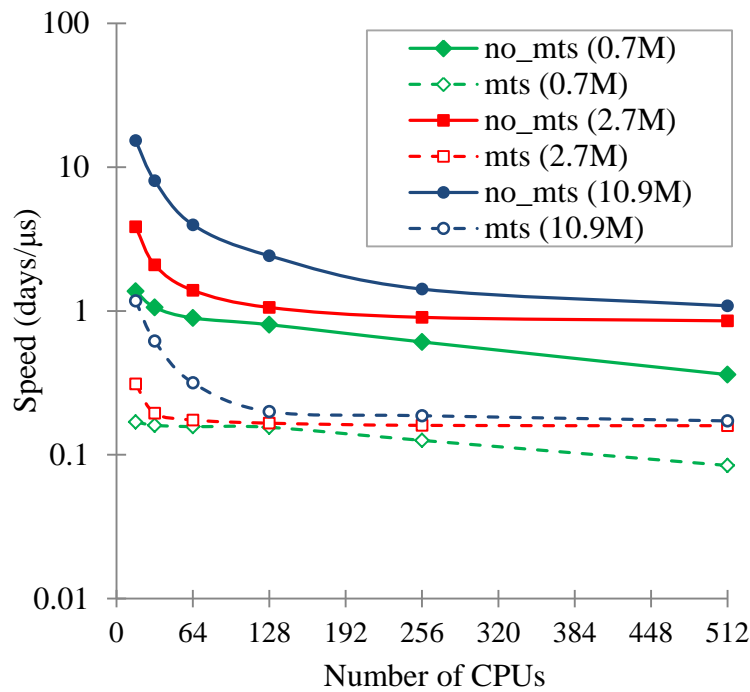


Figure 4-11: Parallel speeds of no\_mts/mts algorithms on Tianhe-2



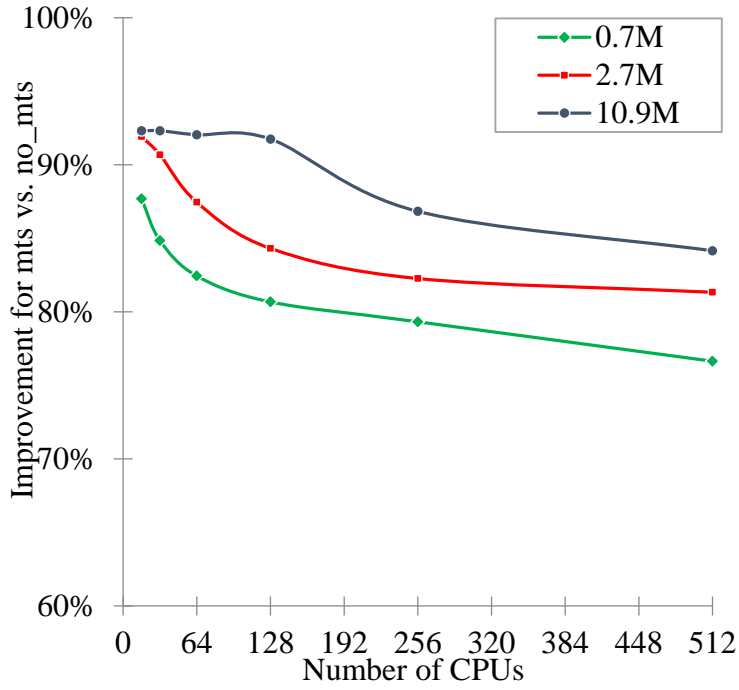


Figure 4-12: Performance improvement of mts over no\_mts on Tianhe-2

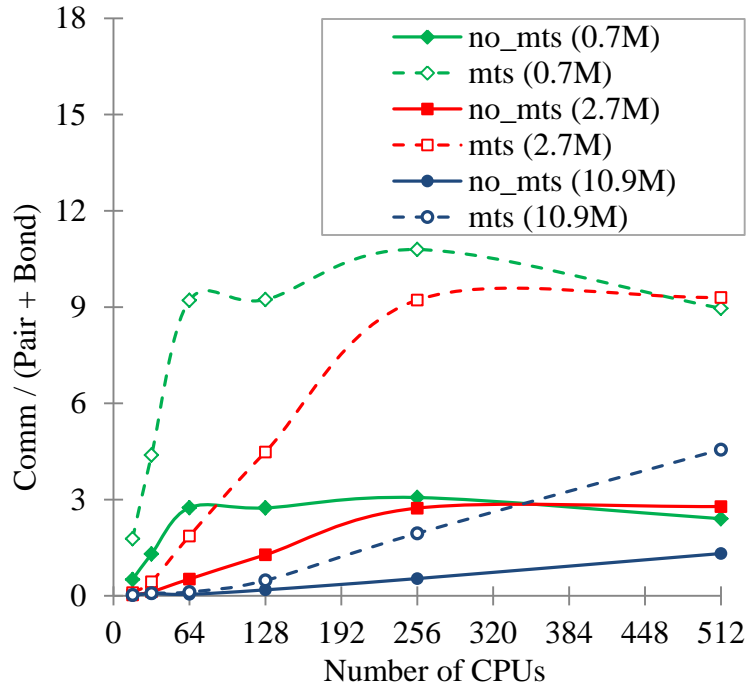


Figure 4-13: Ratio of communication over computation on Tianhe2

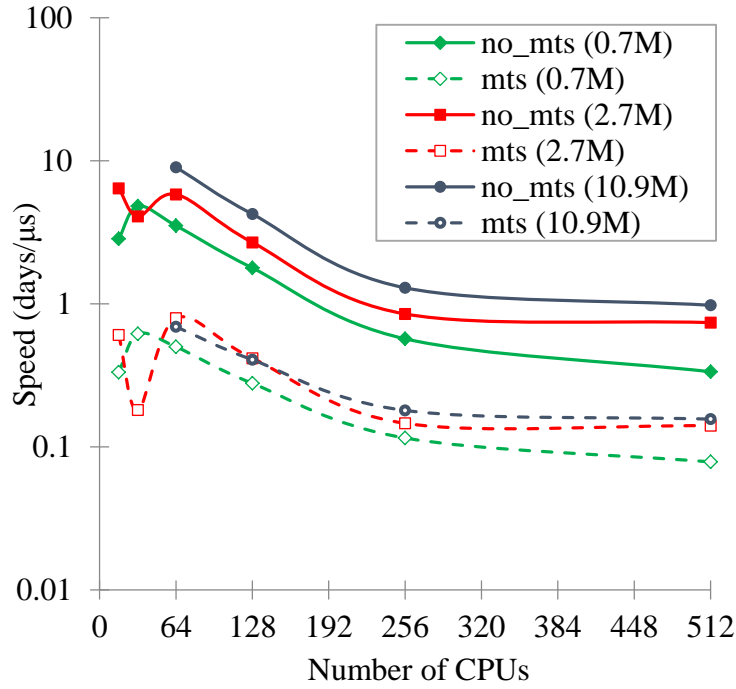


Figure 4-14: Parallel speeds of no\_mts/mts algorithms on Stampede

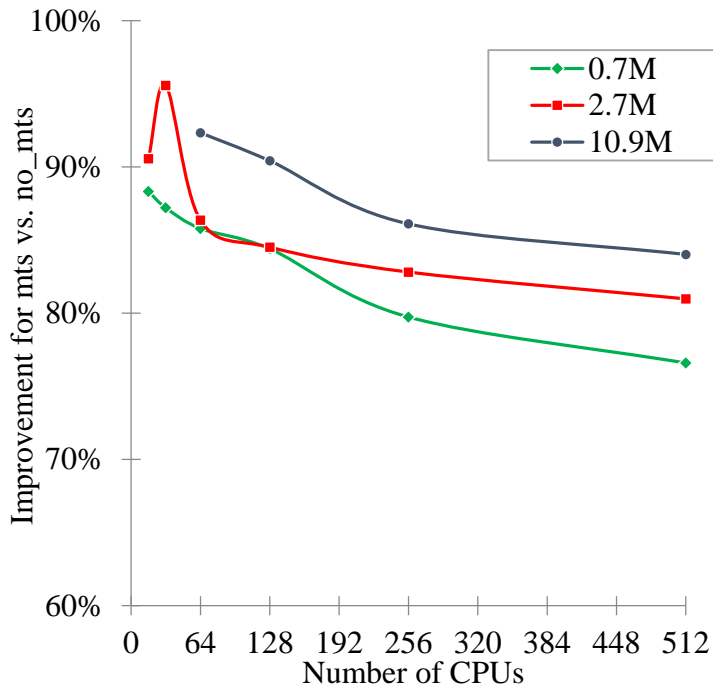


Figure 4-15: Performance improvement of mts over no\_mts on Stampede

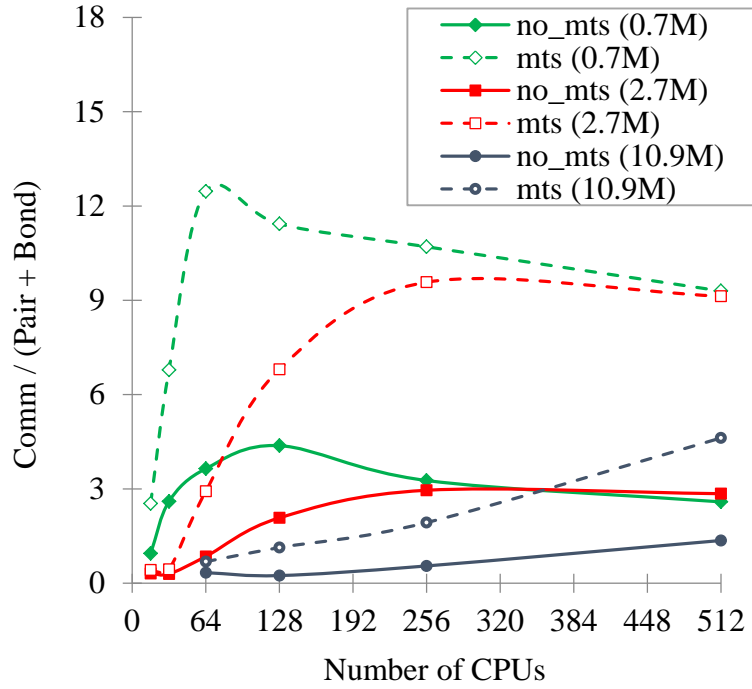


Figure 4-16: Ratio of communication over computation on Stampede

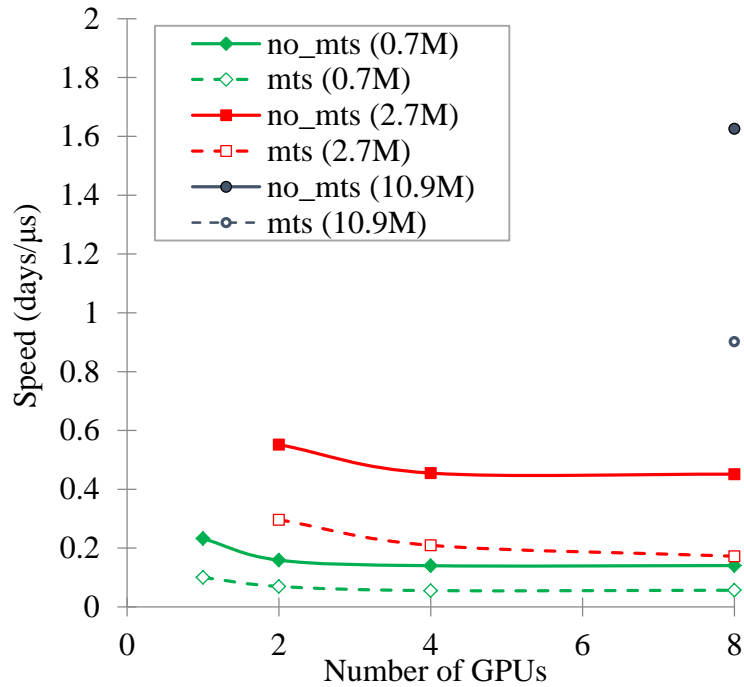


Figure 4-17: Parallel speeds of no\_mts/mts algorithms on 8-K40m CS-Storm

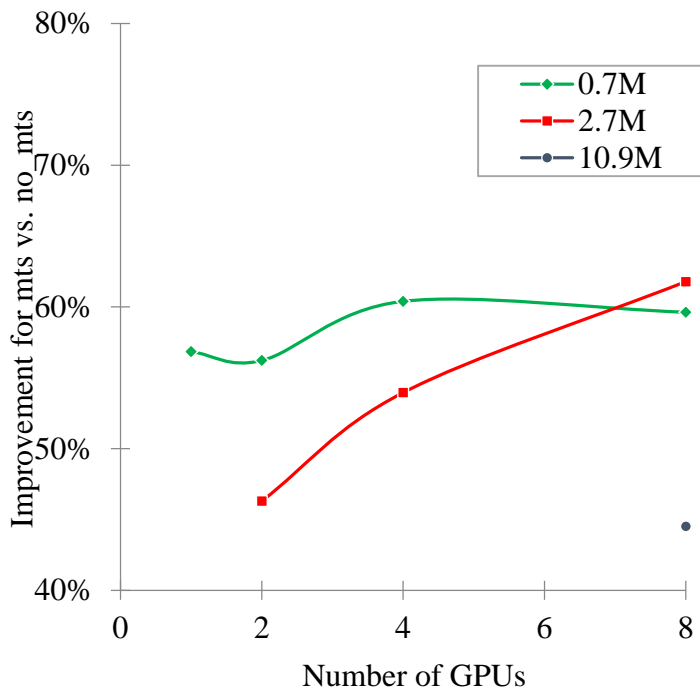


Figure 4-18: Performance improvement of mts over no\_mts on 8-K40m CS-Storm

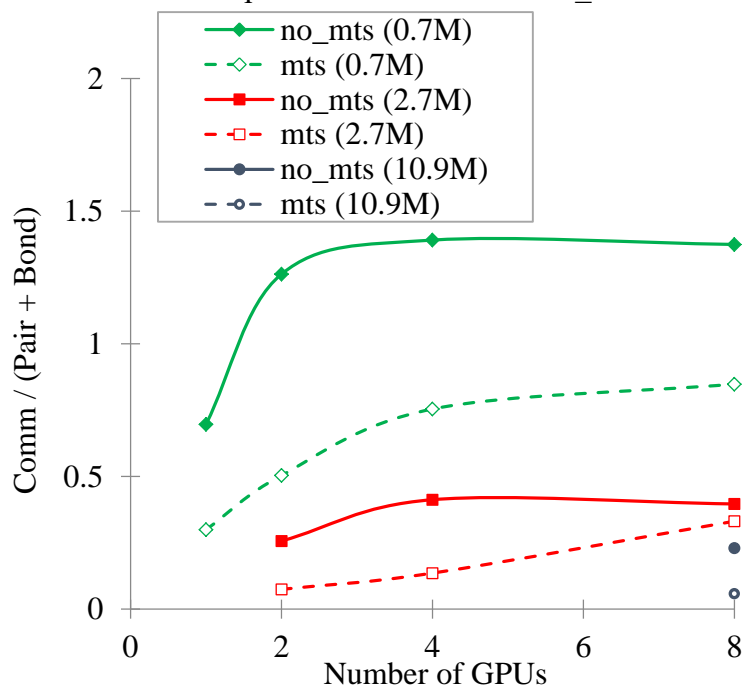


Figure 4-19: Ratio of communication over computation on 8-K40m CS-Storm

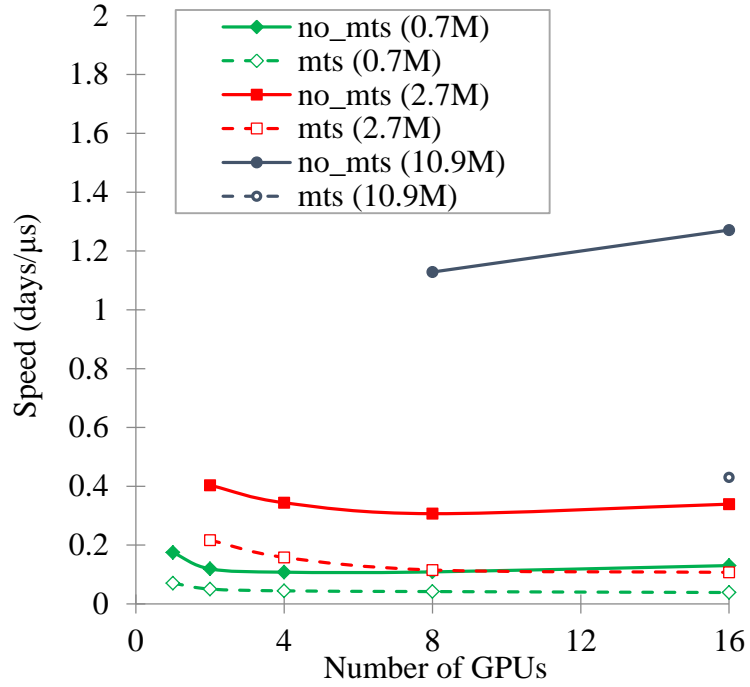


Figure 4-20: Parallel speeds of no\_mts/mts algorithms on 16-K80 CS-Storm

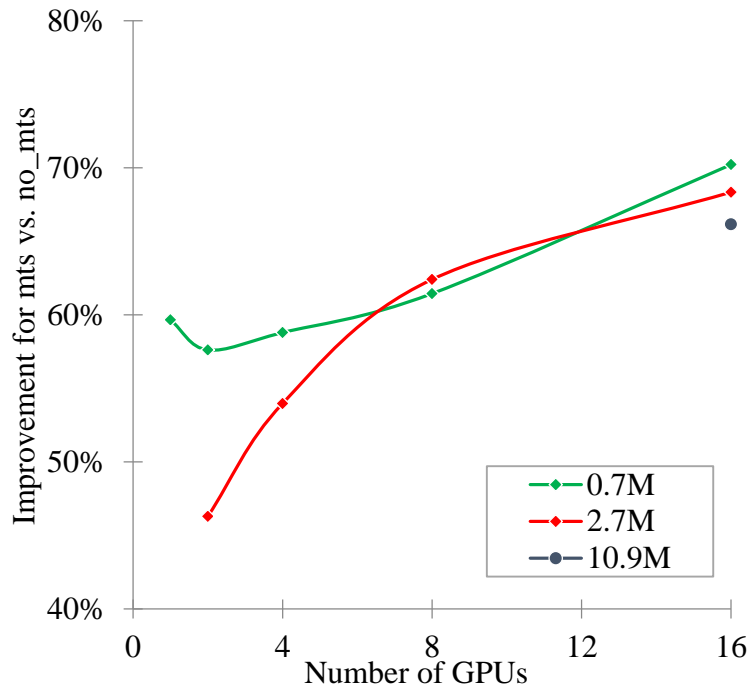


Figure 4-21: Performance improvement of mts over no\_mts on 16-K80 CS-Storm

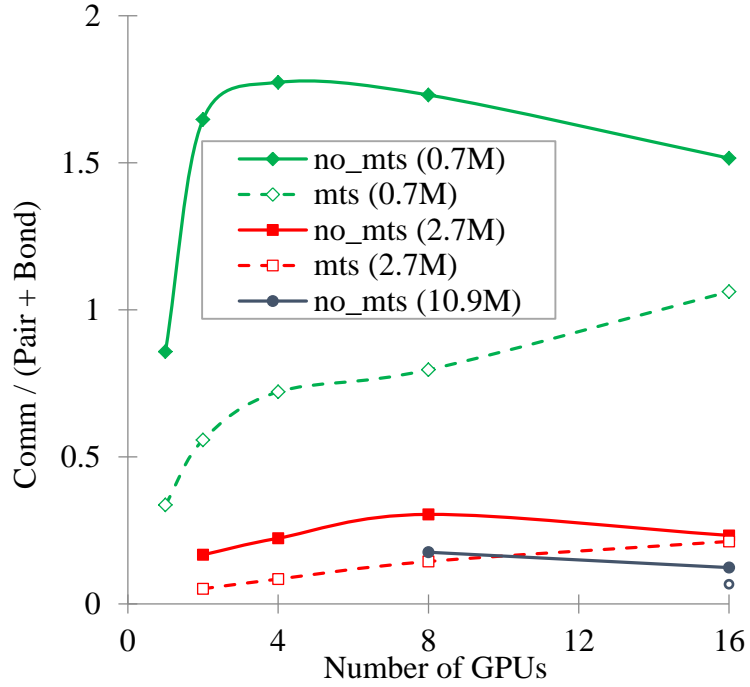


Figure 4-22: Ratio of communication over computation on 16-K80 CS-Storm

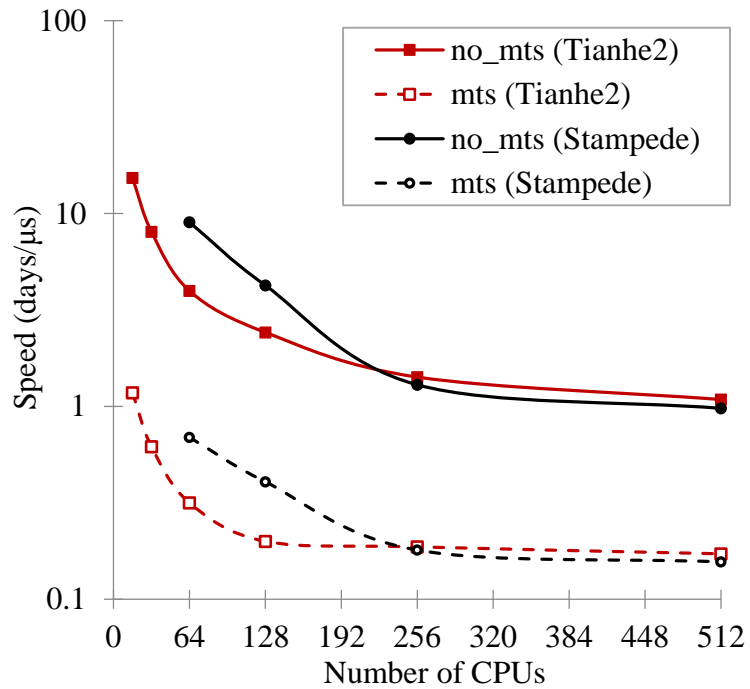


Figure 4-23: Parallel speeds of no\_mts and mts algorithms for Exp-L on Tianhe-2 and Stampede

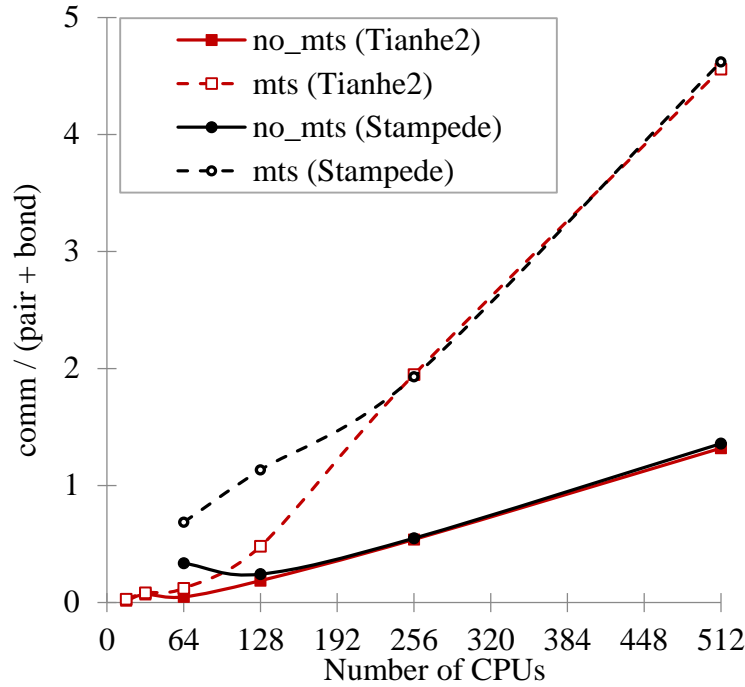


Figure 4-24: Ratio of communication over computation for no\_mts and mts algorithms for Exp-L on Tianhe-2 and Stampede

Table 4-5: Best performances for STS and MTS algorithms on three supercomputers for Exp-S

Systems	Resources		Speed (days/ $\mu$ s)		Speed ( $\mu$ s/day)		Speedup for MTS vs. STS
	# of GPUs	# of Cores	STS	MTS	STS	MTS	
K40m	8	16	0.14	0.06	7.14	16.67	2.33
K80	16	16	0.13	0.04	7.69	25.00	3.25
Tianhe-2		256	0.61	0.13	1.64	7.69	4.69
Tianhe-2		512	0.36	0.08	2.78	12.50	4.50
Stampede		256	0.57	0.12	1.75	8.33	4.75
Stampede		512	0.34	0.08	2.94	12.50	4.25
Stampede/K20	16	256	0.14	0.04	7.14	25.00	3.50
Stampede/K20	32	512	0.62	0.03	1.61	35.50	20.67

Table 4-6: Best performances for STS and MTS algorithms on three supercomputers for Exp-M

Systems	Resources		Speed (days/ $\mu$ s)		Speed ( $\mu$ s/day)		Speedup for MTS vs. STS
	# of GPUs	# of Cores	STS	MTS	STS	MTS	
K40m	8	16	0.45	0.17	2.22	5.88	2.65
K80	16	16	0.34	0.11	2.94	9.09	3.09
Tianhe-2		256	0.90	0.16	1.11	6.25	5.63
Tianhe-2		512	0.85	0.16	1.18	6.25	5.31
Stampede		256	0.85	0.15	1.18	6.67	5.67
Stampede		512	0.74	0.14	1.35	7.14	5.29
Stampede/K20	16	256	0.15	0.07	6.67	14.29	2.14
Stampede/K20	32	512	0.14	0.08	7.14	12.50	1.75

Table 4-7: Best performances for STS and MTS algorithms on three supercomputers for Exp-L

Systems	Resources		Speed (days/ $\mu$ s)		Speed ( $\mu$ s/day)		Speedup for MTS vs. STS
	# of GPUs	# of Cores	STS	MTS	STS	MTS	
K40m	8	16	1.63	0.90	0.61	1.11	1.81
K80	16	16	1.27	0.43	0.79	2.33	2.95
Tianhe-2		256	1.42	0.19	0.70	5.26	7.47
Tianhe-2		512	1.09	0.17	0.92	5.88	6.41
Stampede		256	1.29	0.18	0.78	5.56	7.17
Stampede		512	0.98	0.16	1.02	6.25	6.13
Stampede/K20	32	512	0.29	-	3.45	-	-



Table 4-8: Simulation speeds ( $\mu\text{s}/\text{day}$ ) for multiscale model and MTS algorithm on Tianhe-2, Stampede and CS-Storm

Systems		Tianhe-2 (512 cores)	Stampede (512 cores)	Stampede (512 cores + 32 K20)	CS-Storm (20 cores + 8 K40)	CS-Storm (24 cores + 16 K80)
Problems	single platelet 680,718 particles	12.50	12.50	35.50	16.67	25.00
	4 platelets 2,722,872 particles	6.25	7.14	12.50	5.88	9.09
	16 platelets 10,891,488 particles	5.88	6.25	-	1.11	2.33

### Analysis

Based on performance examinations for three multiscale benchmarks on three supercomputers, MTS proves to be an efficient algorithm that achieves the desired performance of multiscale simulations, developing the problem-specific MTS algorithms may enable the solutions of a class of multiple spatial-temporal problems. The results in Figure 4-12, Figure 4-15, Figure 4-18 and Figure 4-21 show that MTS is consistently superior over STS in all tests on all supercomputers in terms of simulation speeds. The improvement for MTS over STS is larger on Tianhe-2/Stampede than that on CS-Storm. For example, for the Exp-L, the improvement is 84% on Tianhe-2/Stampede and it is 66% and 45% on CS-Storm 16-K80/8-K40m.

MTS excels at reducing the computation while without altering the communication pattern so the communication-to-computation ratio varies across diversified supercomputers. In general, MTS has the higher ratios on Tianhe-2/Stampede (Figure 4-13 and Figure 4-16) and the lower ratios on CS-Storm 8-K40m/16-K80 (Figure 4-19 and Figure 4-22), in comparison with STS. These phenomena could be the result of widely different communication schemes: in the code, the inter-process communication employs the intra-node wires in CS-Storm and the inter-node cables in Tianhe-2 and Stampede. The intra-node wires are usually faster and more reliable connections than the inter-node cables.

Comparing speeds in Figure 4-23, we can see that for Exp-L, Tianhe-2 is faster than Stampede for up to 128 cores and then Stampede is faster for 256 and 512 cores than Tianhe-2. This could be the result of nodal configurations. Tianhe-2 and Stampede have 24 and 16 cores per node, respectively. To provide the run with 128 cores, Stampede and Tianhe-2 needs 8 and 5 nodes respectively and thus Tianhe-2 would need less inter-node data transfers than Stampede. In addition, Tianhe-2 has more host memory (64GB per node) than Stampede (32GB per node), further boosting the nodal performance. Thus, when using a small group of cores, Tianhe-2 is better than Stampede. With the increase of cores, communication becomes more stressful than computation (Figure 4-24) then the performance gap between Stampede and Tianhe-2 appears to diminish: 0.16~0.17 *days/μs* (MTS) and 0.98~1.09 *days/μs* (STS) for Stampede/Tianhe-2 using 512 cores.

Multiscale simulation can provide nanoscale details of biomedical problems with affordable computing resources. The results demonstrate that the simulation rate of multiple microseconds of physical time per day for systems with multiple platelets and 10-million particles is achieved on supercomputers. Our multiscale approach offers nanoscale details for intracellular details while modeling the bulk transport of blood flows at the microscale [123]. The multiscale nature helps improve the scale of stepping sizes from femtosecond to nanosecond [124] for the deforming platelets. In addition, the MTS allows the viscous flow particles at the scale of 100-nanosecond stepping sizes. This improved the simulation speeds. For example within one day, the CS-Storm 16-K80 server can simulate 25 microseconds for a single-platelet system (Table 4-5). The 512-cores in Tianhe-2 and Stampede can simulate 5.88~6.25 microseconds for a 16-platelet system (Table 4-7) within one day. Accordingly, the 1-millisecond nano/micro-composite simulation for a 10-million-particle system can be completed in 160 wallclock days.

The accelerator could further boost the performance of large scale computer systems. For example, the K20-enabled nodes improve the performance of CPU-only nodes by a factor of 2~3 for the Exp-M on Stampede. In other words, the time it takes for 1-millisecond simulation of a 2.7-million-particle system can be reduced from 140 days (32 CPU-only nodes) to 70 days (16 K20-enabled nodes) on Stampede. These results demonstrated that the millisecond-scale nano/micro-composite simulations for million-particle systems are computationally affordable on

supercomputers. To compare, Anton 2 (the recipient of the 2014 ACM Gordon Bell Prize for best performance of a high performance application), a special-purpose molecular dynamics supercomputer demonstrated the achievement of 2.2-million-atom ribosome system at a rate of  $3.6 \mu s/day$  [142]. Our results demonstrated the achievement of multiscale 10-million-particle 16-platelet system at a rate of 6.25 and  $5.88 \mu s/day$  on Stampede and Tianhe-2 (Table 4-7); and the achievement of multiscale 2.7-million-particle 4-platelet system at a rate of 14.29 and  $6.25 \mu s/day$  on Stampede/K20 and Tianhe-2.

As the supercomputer architectures are becoming more diversified, there is a need for a match of the architectural supply and application demand for optimal performance. The match can be established through discovering the characteristics of supercomputers and models. For example, the simulation for parameterizing the multiscale models could be handled by a high-density multi-accelerator computer (Table 4-5). The simulation for validating the inter-platelet interactions could be handled by using a small group of computer nodes (Table 4-6). The simulation for studying large-scale multi-platelet and multi-component interactions in arteriole need use the large scale supercomputers with high performance interconnects (Table 4-7). Though novel supercomputers are emerging as dominant tools for computer simulations, it has to wisely match the state-of-the-art supercomputer supply and state-of-the-practical application demand for efficient use of invaluable computing resources.

## 4.5 Summary

We examined the performance for multiscale simulations with flexible MTS algorithms on Stampede supercomputer with multiple CPUs and GPUs solutions. The MTS algorithm consistently outperforms STS algorithm and the GPU system is 2~3 times faster than the CPU-only system. Due to communication overheads, 2-million particle systems accelerate less than the 10-million particle systems. The results show: (a) for medium size system (2.72 million particles, 4 platelets), simulating 1-*ms* phenomena can be reduced from 848 days (STS + 256 cores) to 37 days (MTS + 16 GPU nodes); (b) for large size system (10.89 million particles, 16 platelets), simulating 1-*ms* phenomena can be reduced from 976 days (STS + 512 cores) to 77 days (MTS + 512 cores).

We used the multiscale models to benchmark three top supercomputers. The results demonstrate that (1) the MTS algorithm is superior over classical STS algorithm at improving the computation on all three supercomputers. The improvement percentages are 84% for Tianhe-2/Stampede and 66% and 45% for CS-Storm 16-K80 and 8-K40m for the 10-million-particle system. (2) The high-density multi-accelerator computer, CS-Storm is capable for support of a single-platelet multiscale simulation. In this, completion of 1-*ms* multiscale simulation for the single-platelet system needs 40 days with nanoscale stepping sizes. (3) Tianhe-2 and Stampede are capable for very support of multiscale multi-platelet simulations. In this, completion of 1-*ms* simulation for 16-platelet system (10.9 million particles) needs 160 days. If without MTS or supercomputers, completion of 1-*ms* multiscale simulation of the 16-platelet system needs 15,293 days (i.e., 42 years) using 16 cores. Thus our multiple spatial-temporal model could enable the millisecond-scale simulations of platelet-mediated processes at the junction of mathematics, bioengineering and computer science.

These performance examinations support the assertion that the triple alliance of models, algorithms and powerful computers can be formed to drive the forefront of high performance computing for domain-specific applications. The computational approaches using multiscale models and efficient algorithms on supercomputers will enable these predictive simulations for thrombogenicity and address the challenges in clinical problems.

## Chapter 5 Conclusions and Future Work

Multiscale models, numerical algorithms, and advanced hardware provide an orchestrated effort for solving new classes of problems in science, engineering, and medicine. This research is at the heart of such development.

The design and implementation of multiscale model for solving the multi-component biological problem, specifically, dynamics of platelets flipping in viscous blood flow, as presented in Chapter 2. In this model, CGMD models the molecular effects of platelets at the  $nm$  to  $\mu m$  scales while the DPD simulates the macroscale pathological blood flows at the upper  $\mu m$  scales. To integrate them, a hybrid CGMD-DPD force field is proposed at overlaying region for facilitating rheological information sharing over the spatial scales. In order to accommodate the multiple spatial scales and accelerate simulations, an adaptive MTS algorithm with four-level integrators is developed. The four-level integrator can be adjusted via three loop factors to optimize accuracy and computing speed, resulting in huge speedup of computation while maintaining stability and accuracy.

The parameterization and design of a modified Morse potential for coarse-grained modeling of blood plasma is presented in Chapter 3. Through supercomputing and numerical variational separation technologies, we parameterize modified Morse potential and systematically analyze the impacts of model parameters on the accuracy for representing the physical observables, biomedical phenomena and computational loads. One of its suitable applications is in modeling the biorheology of platelet cytoplasm and this helps design an advanced molecular-level platelet model with accurate mechanics.

The MTS algorithm helps reduce computational costs but without lowering the communication costs proportionally. In order to reduce communication, GPGPU acceleration solutions are implemented. Chapter 4 presents detailed performance analysis of the double-punch speedup strategy, i.e., the algorithmic MTS algorithm and GPGPU acceleration, on three representative supercomputers: Tianhe-2, Stampede and CS-Storm. The performance results manifest the possibility of simulating the millisecond-scale hematology at resolutions of nanoscale platelets and mesoscale bio-flows using millions of particles.

All of those multiscale schemes and speedup strategies are integral steps to enable efficient predictive simulations for initial thrombogenicity study and may provide a useful guide for exploring mechanisms of other complex biomedical problems at disparate spatiotemporal scales. With the advent of novel algorithms and the advance of computer engineering, the era of solving biological system with nanoscale details to a time scale of seconds is starting. It is not difficult to project that more sophisticated biological systems can be simulated using millions of computer processors, in the near future.

Despite all simulation results in this dissertation, further improvements and developments are required in order to study larger systems for longer time, resulting in understanding of physical importance and of practical usage in medicine and biology. Those improvement aspects include all of the following:

The study of multiscale platelet-flow modeling is spatial-temporally limited by top-scale DPD simulation method. An advanced multiscale model couples atomistic-mesoscopic-continuum approaches is required in order to expand the upper limit. This hybrid multiscale modeling may compose of three levels: atomistic level, where atomic interactions of platelets are computed; mesoscopic level, where finer features of blood particulate flow are simulated; and continuum level, where the Navier-Stokes equations are solved for coarsening blood flows. Sophisticated boundary conditions are required to couple all domains.

Although the performance results manifest the possibility of simulating the millisecond-scale hematology phenomena, numerous biological details, such as red blood cell white blood cell, vessel walls were not considered. A larger platelet aggregation simulation system, among others, is required for predicting blood clotting mechanics, rheology, and dynamics, to better understand thrombosis pathologies.

Cost-optimized computing algorithms are in need of deployment to ensure these demands. The great reduction of computing time by our MTS scheme is achieved and GPGPU-enabled acceleration offers a balanced solution for the communication bottleneck imposed by MTS. However, the MPI communications on conventional cluster bulk large amount of time. A new asynchronous communication strategy may help reduce communication cost and achieve greater scalability.

## Bibliography

- [1] K. Amunts, C. Lepage, L. Borgeat, H. Mohlberg, T. Dickscheid, M. E. Rousseau, *et al.*, "BigBrain: an ultrahigh-resolution 3D human brain model," *Science*, vol. 340, pp. 1472-5, Jun 21 2013.
- [2] D. E. Shaw, J. P. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, *et al.*, "Anton2: Raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer," presented at the Proceedings of SC14: International Conference for High Performance Computing, Networking, Storage and Analysis New Orleans, Louisiana, 2014.
- [3] *National Center for Atmospheric Research 2014-2019 Strategic Plan*: National Center for Atmospheric Research, 2014.
- [4] T. Ishiyama, K. Nitadori, and J. Makino, "4.45 Pflops astrophysical N-body simulation on K computer: the gravitational trillion-body problem," presented at the Proceedings of SC14: the International Conference on High Performance Computing, Networking, Storage and Analysis, Salt Lake City, Utah, 2012.
- [5] S. L. Graham, M. Snir, and C. A. Patterson, *Getting up to speed: the future of supercomputing*: Committee on the Future of Supercomputing, Computer Science and Telecommunications Board Division on Engineering and Physical Sciences, National Research Council, 2004.
- [6] T. Anor, L. Grinberg, H. Baek, J. R. Madsen, M. V. Jayaraman, and G. E. Karniadakis, "Modeling of blood flow in arterial trees," *Wiley Interdiscip Rev Syst Biol Med*, vol. 2, pp. 612-23, Sep-Oct 2010.
- [7] A. C. Benim, A. Nahavandi, A. Assmann, D. Schubert, P. Feindt, and S. H. Suh, "Simulation of blood flow in human aorta with emphasis on outlet boundary conditions," *Applied Mathematical Modelling*, vol. 35, pp. 3175-3188, 2011.
- [8] P. Crosetto, P. Reymond, S. Deparis, D. Kontaxakis, N. Stergiopoulos, and A. Quarteroni, "Fluid-structure interaction simulation of aortic blood flow," *Computers & Fluids*, vol. 43, pp. 46-57, 2011.
- [9] D. Bluestein, "Towards optimization of the thrombogenic potential of blood recirculating cardiovascular devices using modeling approaches," *Expert Review of Medical Devices*, vol. 3, pp. 267-270, May 2006.
- [10] T. Yamaguchi, T. Ishikawa, Y. Imai, N. Matsuki, M. Xenos, Y. F. Deng, *et al.*, "Particle-based methods for multiscale modeling of blood flow in the circulation and in devices: challenges and future directions," *Annals of Biomedical Engineering*, vol. 38, pp. 1225-1235, Mar 2010.
- [11] R. D. Allen, L. R. Zacharski, S. T. Widirstky, R. Rosenstein, L. M. Zaitlin, and D. R. Burgess, "Transformation and motility of human-platelets - details of the shape change and release reaction observed by optical and electron-microscopy," *J. Cell Biol.*, vol. 83, pp. 126-142, 1979.

- [12] J. R. O'Brien, "Shear-induced platelet-aggregation," *Lancet*, vol. 335, pp. 711-713, Mar 24 1990.
- [13] L. A. Harker and S. J. Slichter, "Studies of platelet and fibrinogen kinetics in patients with prosthetic heart valves," *New England Journal of Medicine*, vol. 283, pp. 1302-&, 1970.
- [14] J. R. O'Brien and G. P. Salmon, "An independent hemostatic mechanism - shear induced platelet-aggregation," *Thrombosis and Haemostasis*, vol. 62, pp. 258-258, Aug 19 1989.
- [15] Y. Alemu and D. Bluestein, "Flow-induced platelet activation and damage accumulation in a mechanical heart valve: numerical studies," *Artificial Organs*, vol. 31, pp. 677-88, 2007.
- [16] W. Yin, Y. Alemu, K. Affeld, J. Jesty, and D. Bluestein, "Flow-induced platelet activation in bileaflet and monoleaflet mechanical heart valves," *Ann Biomed Eng*, vol. 32, pp. 1058-66, Aug 2004.
- [17] J. R. O'Brien and G. P. Salmon, "An independent haemostatic mechanism: shear induced platelet aggregation," *Adv Exp Med Biol*, vol. 281, pp. 287-96, 1990.
- [18] Y. Alemu and D. Bluestein, "Flow-induced platelet activation and damage accumulation in a mechanical heart valve: numerical studies," *Artif Organs*, vol. 31, pp. 677-88, Sep 2007.
- [19] M. H. Kroll, J. D. Hellums, L. V. McIntire, A. I. Schafer, and J. L. Moake, "Platelets and shear stress," *Blood*, vol. 88, pp. 1525-41, Sep 1 1996.
- [20] J. Sheriff, J. S. Soares, M. Xenos, J. Jesty, M. J. Slepian, and D. Bluestein, "Evaluation of shear-induced platelet activation models under constant and dynamic shear stress loading conditions relevant to devices," *Ann Biomed Eng*, vol. 41, pp. 1279-96, Jun 2013.
- [21] R. Feng, M. Xenos, G. Girdhar, W. Kang, J. W. Davenport, Y. F. Deng, *et al.*, "Viscous flow simulation in a stenosis model using discrete particle dynamics: a comparison between DPD and CFD," *Biomechanics and Modeling in Mechanobiology*, vol. 11, pp. 119-129, Jan 2012.
- [22] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, *et al.*, "Scalable molecular dynamics with NAMD," *Journal of Computational Chemistry*, vol. 26, pp. 1781-1802, Dec 2005.
- [23] R. O. Dror, R. M. Dirks, J. P. Grossman, H. Xu, and D. E. Shaw, "Biomolecular simulation: a computational microscope for molecular biology," *Annu Rev Biophys*, vol. 41, pp. 429-52, 2012.
- [24] R. D. Groot and P. B. Warren, "Dissipative particle dynamics:bridging the gap between atomistic and mesoscopic simulation," *Journal of Chemical Physics*, vol. 107, pp. 4423-4435, Sep 15 1997.



- [25] J. S. Soares, C. Gao, Y. Alemu, M. Slepian, and D. Bluestein, "Simulation of platelets suspension flowing through a stenosis model using a dissipative particle dynamics approach," *Ann Biomed Eng*, vol. 41, pp. 2318–2333, 2013.
- [26] D. Bluestein, W. Yin, K. Affeld, and J. Jesty, "Flow-induced platelet activation in mechanical heart valves," *J Heart Valve Dis*, vol. 13, pp. 501-8, May 2004.
- [27] A. A. Alsheikh-Ali, G. D. Kitsios, E. M. Balk, J. Lau, and S. Ip, "The Vulnerable Atherosclerotic Plaque: Scope of the Literature," *Ann Intern Med*, vol. 153, pp. 387-W149, Sep 21 2010.
- [28] G. Girdhar, M. Xenos, Y. Alemu, W. C. Chiu, B. E. Lynch, J. Jesty, *et al.*, "Device thrombogenicity emulation: a novel method for optimizing mechanical circulatory support device thromboresistance," *PloS One*, vol. 7, p. e32463, 2012.
- [29] D. A. Fedosov, B. Caswell, and G. E. Karniadakis, "A Multiscale Red Blood Cell Model with Accurate Mechanics, Rheology, and Dynamics," *Biophysical journal*, vol. 98, pp. 2215-2225, 2010.
- [30] E. J. Martinez, Y. Lanir, and S. Einav, "Effects of contact-induced membrane stiffening on platelet adhesion," *Biomech Model Mechanobiol*, vol. 2, pp. 157-67, Mar 2004.
- [31] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, *et al.*, "Scalable molecular dynamics with NAMD," *J Comput Chem*, vol. 26, pp. 1781-1802, 2005.
- [32] Y. Deng, P. Zhang, C. Marques, R. Powell, and L. Zhang, "Analysis of Linpack and power efficiencies of the world's TOP500 supercomputers," *Parallel Computing*, vol. 39, pp. 271-279, 2013.
- [33] D. Rossinelli, B. Hejziahosseini, P. Hadjidoukas, C. Bekas, A. Curioni, A. Bertsch, *et al.*, "11 PFLOP/s simulations of cloud cavitation collapse," presented at the Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, Colorado, 2013.
- [34] T. Ishiyama, K. Nitadori, and J. Makino, "4.45 Pflops astrophysical N-body simulation on K computer: the gravitational trillion-body problem," presented at the Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Salt Lake City, Utah, 2012.
- [35] J. K. W. Chesnutt and H.-C. Han, "Effect of red blood cells on platelet activation and thrombus formation in tortuous arterioles," *Frontiers in Bioengineering and Biotechnology*, vol. 1, 2013-December-3 2013.
- [36] T. Yamaguchi, T. Ishikawa, Y. Imai, N. Matsuki, M. Xenos, Y. Deng, *et al.*, "Particle-based methods for multiscale modeling of blood flow in the circulation and in devices: challenges and future directions. Sixth International Bio-Fluid Mechanics Symposium and Workshop March 28-30, 2008 Pasadena, California," *Ann Biomed Eng*, vol. 38, pp. 1225-35, Mar 2010.

- [37] J. McCarty, I. Y. Lyubimov, and M. G. Guenza, "Multiscale Modeling of Coarse-Grained Macromolecular Liquids," *The Journal of Physical Chemistry B*, vol. 113, pp. 11876-11886, 2009/09/03 2009.
- [38] P. Zhang, C. Gao, N. Zhang, M. Slepian, Y. Deng, and D. Bluestein, "Multiscale Particle-Based Modeling of Flowing Platelets in Blood Plasma Using Dissipative Particle Dynamics and Coarse Grained Molecular Dynamics," *Cellular and Molecular Bioengineering*, pp. 1-23, 2014/09/04 2014.
- [39] N. Zhang, P. Zhang, W. Kang, D. Bluestein, and Y. Deng, "Parameterizing the Morse potential for coarse-grained modeling of blood plasma," *Journal of Computational Physics*, vol. 257, Part A, pp. 726-736, 2014.
- [40] J. T. Padding and W. J. Briels, "Systematic coarse-graining of the dynamics of entangled polymer melts: the road from chemistry to rheology," *J Phys Condens Matter*, vol. 23, p. 233101, Jun 15 2011.
- [41] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids*: Clarendon Press, 1989.
- [42] G. Besold, I. I. Vattulainen, M. Karttunen, and J. M. Polson, "Towards better integrators for dissipative particle dynamics simulations," *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, vol. 62, pp. R7611-4, Dec 2000.
- [43] R. Groot and P. Warren, "Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation," *The Journal of Chemical Physics*, vol. 107, pp. 4423-4435, 1997.
- [44] I. Vattulainen, M. Karttunen, G. Besold, and J. M. Polson, "Integration schemes for dissipative particle dynamics simulations: From softly interacting systems towards hybrid models," *The Journal of Chemical Physics*, vol. 116, p. 3967, 2002.
- [45] V. Symeonidis and G. E. Karniadakis, "A family of time-staggered schemes for integrating hybrid DPD models for polymers: Algorithms and applications," *Journal of Computational Physics*, vol. 218, pp. 82-101, 2006.
- [46] S. Plimpton, "Fast parallel algorithms for short-range molecular-dynamics," *Journal of Computational Physics*, vol. 117, pp. 1-19, Mar 1 1995.
- [47] M. Tuckerman, B. J. Berne, and G. J. Martyna, "Reversible multiple time scale molecular dynamics," *The Journal of Chemical Physics*, vol. 97, p. 1990, 1992.
- [48] M. E. Tuckerman, B. J. Berne, and G. J. Martyna, "Molecular-dynamics algorithm for multiple time scales - systems with long-range forces," *Journal of Chemical Physics*, vol. 94, pp. 6811-6815, May 15 1991.
- [49] M. E. Tuckerman, G. J. Martyna, and B. J. Berne, "Molecular-dynamics algorithm for condensed systems with multiple time scales," *Journal of Chemical Physics*, vol. 93, pp. 1287-1291, Jul 15 1990.
- [50] F. Thalmann and J. Farago, "Trotter derivation of algorithms for Brownian and dissipative particle dynamics," *Journal of Chemical Physics*, vol. 127, Sep 28 2007.

- [51] M. Serrano, G. De Fabritiis, P. Espanol, and P. V. Coveney, "A stochastic Trotter integration scheme for dissipative particle dynamics," *Mathematics and Computers in Simulation*, vol. 72, pp. 190-194, Sep 9 2006.
- [52] A. F. Jakobsen, G. Besold, and O. G. Mouritsen, "Multiple time step update schemes for dissipative particle dynamics," *J Chem Phys*, vol. 124, p. 94104, Mar 7 2006.
- [53] H. Grubmüller and P. Tavan, "Multiple time step algorithms for molecular dynamics simulations of proteins: How good are they?," *J Comput Chem*, vol. 19, pp. 1534-1552, 1998.
- [54] D. A. Fedosov and G. E. Karniadakis, "Triple-decker: Interfacing atomistic-mesoscopic-continuum flow regimes," *Journal of Computational Physics*, vol. 228, pp. 1157-1171, 2009.
- [55] W. Wang and M. R. King, "Multiscale modeling of platelet adhesion and thrombus growth," *Annals of Biomedical Engineering*, vol. 40, pp. 2345-54, Nov 2012.
- [56] P. Zhang, J. Sheriff, J. S. Soares, C. Gao, S. Pothapragada, N. Zhang, *et al.*, "Multiscale Modeling of Flow Induced Thrombogenicity Using Dissipative Particle Dynamics and Coarse Grained Molecular Dynamics," in *ASME 2013 Summer Bioengineering Conference*, 2013, pp. V01BT36A002-V01BT36A002.
- [57] P. Espanol and P. Warren, "Statistical mechanics of dissipative particle dynamics," *EPL (Europhysics Letters)*, vol. 30, p. 191, 1995.
- [58] J. S. Soares, C. Gao, Y. Alemu, M. Slepian, and D. Bluestein, "Simulation of platelets suspension flowing through a stenosis model using a dissipative particle dynamics approach," *Ann Biomed Eng*, vol. 41, pp. 2318-33, Nov 2013.
- [59] S. M. Willemsen, H. C. J. Hoefsloot, and P. D. Iedema, "No-slip boundary condition in dissipative particle dynamics," *International Journal of Modern Physics C*, vol. 11, pp. 881-890, Jul 2000.
- [60] J. H. Haga, A. J. Beaudoin, J. G. White, and J. Strony, "Quantification of the passive mechanical properties of the resting platelet," *Annals of Biomedical Engineering*, vol. 26, pp. 268-277, 1998.
- [61] C. R. Sweet, S. Chatterjee, Z. Xu, K. Bisordi, E. D. Rosen, and M. Alber, "Modelling platelet–blood flow interaction using the subcellular element Langevin method," *Journal of The Royal Society Interface*, vol. 8, pp. 1760-1771, 2011.
- [62] N. A. Mody and M. R. King, "Three-dimensional simulations of a platelet-shaped spheroid near a wall in shear flow," *Physics of Fluids*, vol. 17, p. 113302, 2005.
- [63] N. A. Mody, O. Lomakin, T. A. Doggett, T. G. Diacovo, and M. R. King, "Mechanics of transient platelet adhesion to von Willebrand factor under flow," *Biophysical Journal*, vol. 88, pp. 1432-1443, 2005.
- [64] C. Pozrikidis, "Flipping of an adherent blood platelet over a substrate," *Journal of Fluid Mechanics*, vol. 568, pp. 161-172, Dec 10 2006.

- [65] N. A. Mody and M. R. King, "Platelet adhesive dynamics. Part I: characterization of platelet hydrodynamic collisions and wall effects," *Biophysical Journal*, vol. 95, pp. 2539-2555, Sep 1 2008.
- [66] N. A. Mody and M. R. King, "Platelet adhesive dynamics. Part II: high shear-induced transient aggregation via GPIIb/IIIa-vWF-GPIIb/IIIa bridging," *Biophysical Journal*, vol. 95, pp. 2556-74, Sep 2008.
- [67] A. P. Thompson, S. J. Plimpton, and W. Mattson, "General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions," *Journal of Chemical Physics*, vol. 131, Oct 21 2009.
- [68] W. Dzwiniel, D. A. Yuen, and K. Boryczko, "Mesoscopic dynamics of colloids simulated with dissipative particle dynamics and fluid particle model," *Journal of Molecular Modeling*, vol. 8, pp. 33-43, 2002.
- [69] G. Han, Y. Deng, J. Glimm, and G. Martyna, "Error and timing analysis of multiple time-step integration methods for molecular dynamics," *Computer Physics Communications*, vol. 176, pp. 271-291, 2007.
- [70] L. Verlet, "Computer Experiments on Classical Fluids .I. Thermodynamical Properties of Lennard-Jones Molecules," *Physical Review*, vol. 159, pp. 98-&, 1967.
- [71] S. Plimpton, R. Pollock, and M. Stevens, "Particle-mesh ewald and rRESPA for parallel molecular dynamics simulations," in *PPSC*, 1997.
- [72] B. Leimkuhler, D. T. Margul, and M. E. Tuckerman, "Stochastic, resonance-free multiple time-step algorithm for molecular dynamics with very large time steps," *Molecular Physics*, vol. 111, pp. 3579-3594, Dec 1 2013.
- [73] T. Schlick, M. Mandziuk, R. D. Skeel, and K. Srinivas, "Nonlinear resonance artifacts in molecular dynamics simulations," *Journal of Computational Physics*, vol. 140, pp. 1-29, Feb 10 1998.
- [74] M. E. Tuckerman, B. J. Berne, and A. Rossi, "Molecular-dynamics algorithm for multiple time scales - systems with disparate masses," *Journal of Chemical Physics*, vol. 94, pp. 1465-1469, Jan 15 1991.
- [75] B. Leimkuhler and X. Shang, "On the numerical treatment of dissipative particle dynamics and related systems," *arXiv preprint arXiv:1405.4839*, 2014.
- [76] H. Heinz, W. Paul, and K. Binder, "Calculation of local pressure tensors in systems with many-body interactions," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 72, p. 066704, Dec 2005.
- [77] J. Apel, R. Paul, S. Klaus, T. Siess, and H. Reul, "Assessment of hemolysis related quantities in a microaxial blood pump by computational fluid dynamics," *Artif Organs*, vol. 25, pp. 341-7, May 2001.

- [78] G. B. Jeffery, "The motion of ellipsoidal particles in a viscous fluid," *Proceedings of the Royal Society of London Series a-Containing Papers of a Mathematical and Physical Character*, vol. 102, pp. 161-179, Nov 1922.
- [79] B. Leimkuhler and C. Matthews, "Rational construction of stochastic numerical methods for molecular sampling," *Applied Mathematics Research eXpress*, vol. 2013, pp. 34-56, 2013.
- [80] P. Zhang, R. Powell, and Y. Deng, "Interlacing Bypass Rings to Torus Networks for More Efficient Networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, pp. 287-295, 2011.
- [81] R. M. Guezuraga and D. Y. Steinbring, "View from industry," *European Journal of Cardio-Thoracic Surgery*, vol. 26, pp. S19-S23, Dec 2004.
- [82] C. De Lazzari, M. Darowski, G. Ferrari, and F. Clemente, "The influence of left ventricle assist device and ventilatory support on energy-related cardiovascular variables.," *Medical Engineering & Physics*, vol. 20, pp. 83-91, Mar 1998.
- [83] Y. D. Dwizel W, Boryczko K, "Mesoscopic dynamics of colloids simulated with dissipative particle dynamics and fluid particle model.," *J Mol Model*, vol. 8(1):33-43, 2002.
- [84] E. S. Boek, P. V. Coveney, and H. N. W. Lekkerkerker, "Computer simulation of rheological phenomena in dense colloidal suspensions with dissipative particle dynamics," *Journal of Physics-Condensed Matter*, vol. 8, pp. 9509-9512, Nov 18 1996.
- [85] P. Espanol, "Fluid particle model," *Physical Review E*, vol. 57, pp. 2930-2948, Mar 1998.
- [86] P. J. Hoogerbrugge and J. M. V. A. Koelman, "Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics," *Europhysics Letters*, vol. 19, pp. 155-160, Jun 1 1992.
- [87] T. Y. Hou and X. H. Wu, "A multiscale finite element method for elliptic problems in composite materials and porous media," *Journal of Computational Physics*, vol. 134, pp. 169-189, Jun 1997.
- [88] K. Boryczko, W. Dzwinel, and D. A. Yuen, "Modeling heterogeneous mesoscopic fluids in irregular geometries using shared memory systems," *Molecular Simulation*, vol. 31, pp. 45-56, Jan 15 2005.
- [89] H. Wang, C. Junghans, and K. Kremer, "Comparative atomistic and coarse-grained study of water: what do we lose by coarse-graining?," *European Physical Journal E*, vol. 28, pp. 221-229, Feb 2009.
- [90] C. C. Liew and M. Mikami, "A coarse-grained model for particle dynamics simulations of complex fluids," *Chemical Physics Letters*, vol. 368, pp. 346-351, Jan 17 2003.
- [91] Y. Liu and T. Ichiye, "Soft sticky dipole potential for liquid water: A new model," *Journal of Physical Chemistry*, vol. 100, pp. 2723-2730, Feb 15 1996.

- [92] V. Molinero and W. A. Goddard, "M3B: a coarse grain force field for molecular simulations of malto-oligosaccharides and their water mixtures," *Journal of Physical Chemistry B*, vol. 108, pp. 1414-1427, Jan 29 2004.
- [93] M. Praprotnik, L. Delle Site, and K. Kremer, "Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly," *Journal of Chemical Physics*, vol. 123, Dec 8 2005.
- [94] D. Reith, M. Putz, and F. Muller-Plathe, "Deriving effective mesoscale potentials from atomistic simulations," *Journal of Computational Chemistry*, vol. 24, pp. 1624-1636, Oct 2003.
- [95] S. Riniker and W. F. Gunsteren, "A simple, efficient polarizable coarse-grained water model for molecular dynamics simulations," *Journal of Chemical Physics*, vol. 134, Feb 28 2011.
- [96] J. C. Shelley, M. Y. Shelley, R. C. Reeder, S. Bandyopadhyay, P. B. Moore, and M. L. Klein, "Simulations of phospholipids using a coarse grain model," *Journal of Physical Chemistry B*, vol. 105, pp. 9785-9792, Oct 11 2001.
- [97] W. Shinoda, R. Devane, and M. L. Klein, "Multi-property fitting and parameterization of a coarse grained model for aqueous surfactants," *Molecular Simulation*, vol. 33, pp. 27-36, Jan-Feb 2007.
- [98] S. Garde and H. S. Ashbaugh, "Temperature dependence of hydrophobic hydration and entropy convergence in an isotropic model of water," *Journal of Chemical Physics*, vol. 115, pp. 977-982, Jul 8 2001.
- [99] S. W. Chiu, H. L. Scott, and E. Jakobsson, "A coarse-grained model based on Morse potential for water and n-Alkanes," *Journal of Chemical Theory and Computation*, vol. 6, pp. 851-863, Mar 2010.
- [100] S. J. Marrink, A. H. de Vries, and A. E. Mark, "Coarse grained model for semiquantitative lipid simulations," *Journal of Physical Chemistry B*, vol. 108, pp. 750-760, Jan 15 2004.
- [101] P. M. Morse, "Diatomic molecules according to the wave mechanics. II. Vibrational levels," *Physical Review*, vol. 34, pp. 57-64, Jul 1929.
- [102] R. E. Bolz, *Handbook of tables for applied engineering sciences* vol. 72, 1973.
- [103] G. Kesmarky, P. Kenyeres, M. Rabai, and K. Toth, "Plasma viscosity: a forgotten variable," *Clinical Hemorheology and Microcirculation*, vol. 39, pp. 243-246, 2008.
- [104] S. Plimpton, A. Thompson, and P. Crozier. (2012). *LAMMPS Molecular Dynamics Simulator*. Available: <http://lammps.sandia.gov>
- [105] H. J. C. Berendsen, "Molecular dynamics with coupling to an external bath," *J. Chem. Phys.*, vol. 81, 1984.

- [106] K. A. Motakabbir and M. Berkowitz, "Isothermal compressibility of Spc/E water," *Journal of Physical Chemistry*, vol. 94, pp. 8359-8362, Oct 18 1990.
- [107] R. Zwanzig, "Time-correlation functions and transport coefficients in statistical mechanics," *Annual Review of Physical Chemistry*, vol. 16, pp. 67-&, 1965.
- [108] D. Nevins and F. J. Spera, "Accurate computation of shear viscosity from equilibrium molecular dynamics simulations," *Molecular Simulation*, vol. 33, pp. 1261-1266, 2007.
- [109] H. A. Lei, D. A. Fedosov, and G. E. Karniadakis, "Time-dependent and outflow boundary conditions for Dissipative Particle Dynamics," *Journal of Computational Physics*, vol. 230, pp. 3765-3779, May 10 2011.
- [110] J. A. Backer, C. P. Lowe, H. C. J. Hoefsloot, and P. D. Iedema, "Poiseuille flow to measure the viscosity of particle model fluids," *Journal of Chemical Physics*, vol. 122, Apr 15 2005.
- [111] L. Zhang, P. Zhang, N. Zhang, and Y. Deng, "A Multiscale platelet model and the corroborations between computational and in-vitro experiments," *Multiscale Modeling & Simulation (under review)*, 2015.
- [112] J. Dongara, J. Hittinger, J. Bell, L. Chacon, R. Falgout, M. Heroux, *et al.*, "Applied Mathematics Research for Exascale Computing," U.S. Department of Energy&Office of Science&Advanced Scientific Computing Research Program March 2014.
- [113] J. L. Klepeis, K. Lindorff-Larsen, R. O. Dror, and D. E. Shaw, "Long-timescale molecular dynamics simulations of protein structure and function," *Curr Opin Struct Biol*, vol. 19, pp. 120-7, Apr 2009.
- [114] F. Khalili-Araghi, B. Ziervogel, J. C. Gumbart, and B. Roux, "Molecular dynamics simulations of membrane proteins under asymmetric ionic concentrations," *J Gen Physiol*, vol. 142, pp. 465-75, Oct 2013.
- [115] D. E. Shaw, "Architectures and algorithms for millisecond-scale molecular dynamics simulations of proteins," *Abstracts of Papers of the American Chemical Society*, vol. 237, Mar 22 2009.
- [116] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, *et al.*, "Anton, a special-purpose machine for molecular dynamics simulation," *Communications of the Acm*, vol. 51, pp. 91-97, Jul 2008.
- [117] A. Snell, "HPC forecast: vertical markets and economic sectors," Intersect360 Research HPC market advisory service 2010.
- [118] M. Karplus, "Development of multiscale models for complex chemical systems: from H+H(2) to biomolecules (Nobel Lecture)," *Angew Chem Int Ed Engl*, vol. 53, pp. 9992-10005, Sep 15 2014.
- [119] H. Hodak, "The Nobel Prize in chemistry 2013 for the development of multiscale models of complex chemical systems: a tribute to Martin Karplus, Michael Levitt and Arieh Warshel," *J Mol Biol*, vol. 426, pp. 1-3, Jan 9 2014.

- [120] 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*, vol. 7, Dec 2011.
- [121] W. Wang, N. A. Mody, and M. R. King, "Multiscale model of platelet translocation and collision," *J Comput Phys*, vol. 244, pp. 223-235, Jul 1 2013.
- [122] Z. Xu, N. Chen, M. M. Kamocka, E. D. Rosen, and M. Alber, "A multiscale model of thrombus development," *J R Soc Interface*, vol. 5, pp. 705-22, Jul 6 2008.
- [123] P. Zhang, C. Gao, N. Zhang, M. J. Slepian, Y. Deng, and D. Bluestein, "Multiscale Particle-Based Modeling of Flowing Platelets in Blood Plasma Using Dissipative Particle Dynamics and Coarse Grained Molecular Dynamics," *Cellular and Molecular Bioengineering*, vol. 7, pp. 552-574, 2014/12 2014.
- [124] P. Zhang, N. Zhang, Y. Deng, and D. Bluestein, "A multiple time stepping algorithm for efficient multiscale modeling of platelets flowing in blood plasma," *Journal of Computational Physics*, vol. 284, pp. 668-686, 2015.
- [125] S. Pothapragada, P. Zhang, J. Sheriff, M. Livelli, M. J. Slepian, Y. Deng, *et al.*, "A phenomenological particle-based platelet model for simulating filopodia formation during early activation," *Int j numer method biomed eng*, vol. 31, pp. 1-16, 2015.
- [126] N. Zhang, P. Zhang, W. Kang, D. Bluestein, and Y. F. Deng, "Parameterizing the Morse potential for coarse-grained modeling of blood plasma," *Journal of Computational Physics*, vol. 257, pp. 726-736, Jan 15 2014.
- [127] M. H. Kroll, J. D. Hellums, L. V. McIntire, A. Schafer, and J. L. Moake, "Platelets and Shear Stress," *Blood*, vol. 88, 1996.
- [128] S. Pothapragada, P. Zhang, J. Sheriff, M. Livelli, M. J. Slepian, Y. F. Deng, *et al.*, "A phenomenological particle-based platelet model for simulating filopodia formation during early activation," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 31, pp. 1-16, 2015.
- [129] *Exascale Initiative*. Available: <http://www.exascaleinitiative.org/>
- [130] E. Abraham, C. Bekas, I. Brandic, S. Genaim4, E. B. Johnsen, I. Kondov, *et al.*, "Preparing HPC Applications for Exascale: Challenges and Recommendations," *arXiv*, 24 Mar 2015.
- [131] R. Feng, P. Zhang, and Y. Deng, "Network Design Considerations for Exascale Supercomputers," in *Proceedings of the 24th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2012)*, 2012, pp. 86-93.
- [132] P. Zhang and Y. Deng, "Design and Analysis of Pipelined Broadcast Algorithms for the All-Port Interlaced Bypass Torus Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 2245-2253, 2012.



- [133] P. Zhang and Y. Gao, "Matrix Multiplication on High-Density Multi-GPU Architectures: Theoretical and Experimental Investigations," in *ISC High Performance 2015*, Frankfurt, Germany, 2015.
- [134] Y.-H. Tang and G. E. Karniadakis, "Accelerating dissipative particle dynamics simulations on GPUs: Algorithms, numerics and applications," *Computer Physics Communications*, vol. 185, pp. 2809-2822, 2014.
- [135] M. Bernaschi, M. Bisson, and F. Salvatore, "Multi-Kepler GPU vs. multi-Intel MIC for spin systems simulations," *Computer Physics Communications*, vol. 185, pp. 2495-2503, Oct 2014.
- [136] "Cloud Computing for Science," U.S. Department of Energy Office of Advanced Scientific Computing Research (ASCR), The Magellan Report2011.
- [137] Z. Pang, M. Xie, J. Zhang, Y. Zheng, G. Wang, D. Dong, *et al.*, "The TH Express high performance interconnect networks," *Frontiers of Computer Science*, vol. 8, pp. 357-366, 2014/06/01 2014.
- [138] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington, "Implementing molecular dynamics on hybrid high performance computers - particle-particle particle-mesh," *Computer Physics Communications*, vol. 183, pp. 449-459, Mar 2012.
- [139] W. M. Brown, P. Wang, S. J. Plimpton, and A. N. Tharrington, "Implementing molecular dynamics on hybrid high performance computers - short range forces," *Computer Physics Communications*, vol. 182, pp. 898-911, Apr 2011.
- [140] W. M. Brown and M. Yamada, "Implementing molecular dynamics on hybrid high performance computers-three-body potentials," *Computer Physics Communications*, vol. 184, pp. 2785-2793, Dec 2013.
- [141] *Cray CS-Storm Specifications* Available: <http://www.cray.com/products/computing/cs-series/cs-storm>
- [142] D. E. Shaw, J. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, *et al.*, "Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 41-53.