From Millions to Billions: Scaling Explicit Red Blood Cell Modeling on Exascale Systems

1st Runxin Wu
Biomedical Engineering
Duke University
Durham, USA
wendy.wu@duke.edu

2nd Wentao Ma
Biomedical Engineering
Duke University
Durham, USA
wentao.ma@duke.edu

3rd Jorik Stoop
Biomedical Engineering
Duke University
Durham, USA
jorik.stoop@duke.edu

4th Amanda Randles
Biomedical Engineering
Duke University
Durham, USA
amanda.randles@duke.edu

***Abstract*—Explicitly resolving deformable red blood cells (RBCs) at physiological scale remains a fundamental challenge in computational hemodynamics due to high-levels of memory, communication, and initialization demands. We present the first explicit billion-scale RBC simulation using HARVEY, a massively-parallel fluid–structure interaction solver characterizing high-fidelity deformable cell simulations. By combining scalable cell packing, distributed data management, and heterogeneous CPU–GPU execution, we initialize and simulate 4.95 billion RBCs within a 1 mL blood volume using 9,400 Frontier nodes (601,600 CPU cores and 75,200 AMD GPUs). The simulation occupies 3.76 PB of CPU memory and 1.44 PB of GPU memory while maintaining approximately 75% weak-scaling efficiency near full-system scale. These results extend explicit RBC modeling from millions to billions of cells and establish the computational foundation for physiologically-scaled cell-resolved blood flow simulations.**

***Index Terms*—Computational fluid dynamics, fluid structure interaction, GPU computing, high performance computing**

I. Introduction

Explicit simulation of deformable red blood cells (RBCs) is a fundamental challenge in computational hemodynamics because it couples fluid dynamics, membrane mechanics, and cell–cell interactions across billions of moving entities. Physiologically relevant blood volumes contain trillions of RBCs, making cell-resolved simulation prohibitively expensive due to extreme memory requirements, irregular communication patterns, and costly initialization. Nevertheless, resolving individual RBCs is essential for understanding how cellular mechanics influence blood rheology and disease processes such as sickle cell vaso-occlusion, thrombosis, and cancer metastasis [1]. Recent advances in exascale computing and heterogeneous CPU–GPU architectures provide an opportunity to revisit these computational limits. In this work, we extend explicit deformable RBC simulation from millions to billions of cells by efficiently packing and managing 4.95 billion RBCs within a 1 mL blood volume using 9,400 Frontier nodes, establishing a new scale for physiologically relevant cell-resolved blood simulation.

II. Related work

Large-scale RBC simulation has steadily progressed as computational resources and numerical methods have advanced. Early simulations represented RBCs as rigid coarse-grained particles, enabling approximately 300 million cells over 1.95 mL on the CPU-only Intrepid Blue Gene/P system [2]. Subsequent work introduced deformable membrane models, increasing each RBC representation from a handful of rigid-body variables to hundreds of membrane degrees of freedom. The Gordon Bell-winning SC10 study simulated 262.1 million deformable RBCs on 196,608 Jaguar CPU cores [3]. Later work incorporated higher-fidelity membrane mechanics, collision detection, vascular geometries, and heterogeneous CPU–GPU execution, reaching 1.05 million cells on Stampede2 and 17 million cells on Summit [4], [5].

As RBC models become more physically faithful, memory footprint, data movement, and communication rapidly emerge as the dominant scalability bottlenecks. Although modern exascale systems provide unprecedented aggregate memory capacity, bandwidth, and accelerator performance, efficient initialization, distributed data management, workload balancing, and communication-efficient fluid–structure coupling remain essential for exploiting these resources. Consequently, explicit deformable RBC simulations have remained orders of magnitude below physiologically relevant scales despite continued advances in hardware and numerical methods. Our work addresses these computational bottlenecks through scalable cell packing, memory-efficient distributed data structures, balanced heterogeneous execution, and communication-efficient coupling, extending explicit deformable RBC simulation from millions to 4.95 billion cells.

III. Method

RBC-resolved blood flow was simulated using HARVEY, a massively parallel fluid–structure interaction solver [5], [6]. Fluid dynamics were solved using the lattice Boltzmann method, while RBC membranes were modeled using the finite element method. Each biconcave cell membrane was discretized into 642 vertices and 1,280 triangular elements, with three-component position, velocity, and force variables defined at each vertex. The hyperelastic Skalak constitutive law described membrane resistance to deformation and interactions with neighboring cells and vascular walls [7]. The two-way coupling between cell mechanics and the surrounding fluid was achieved using the immersed boundary method [8].

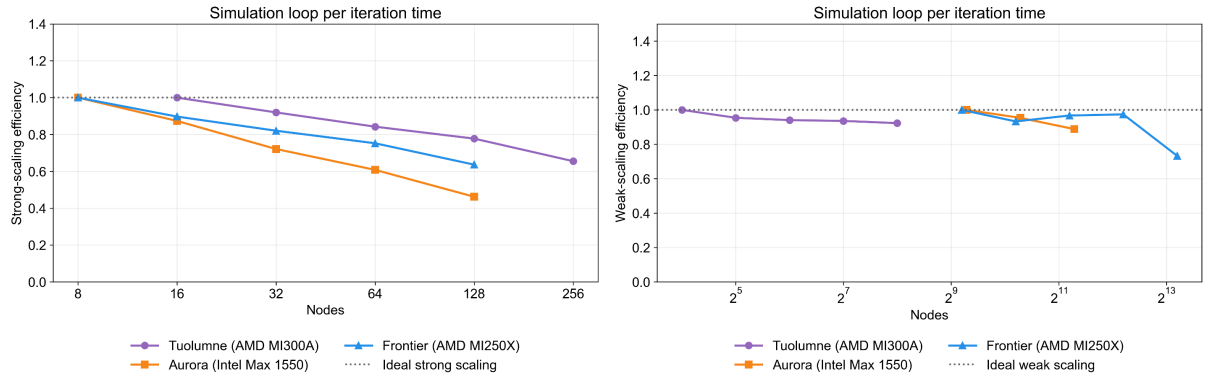


Fig. 1. Performance evaluation of RBC packing across Frontier, Aurora, and Tuolumne. Left: strong scaling efficiency on a 1100um side length cube across 8 to 256 nodes. Right: weak scaling efficiency evaluated across cubic domains with volumes scaled up as node count increased by a factor 2. The weak scaling tests were designed at maximum cell packing capacity per node for each system. Scaling efficiency was calculated with respect to starting node count of each system.

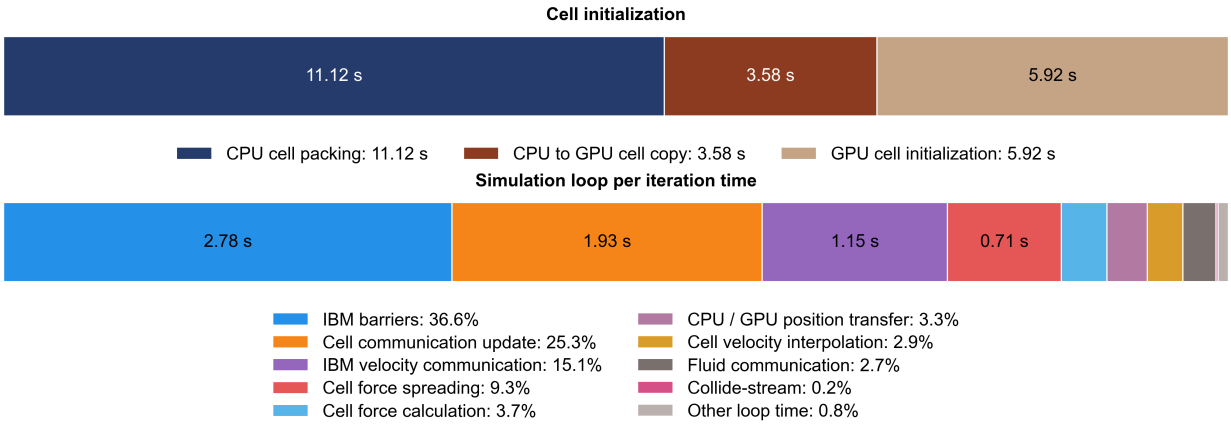


Fig. 2. Performance profiling for full system run on Frontier with 4.95 billion cells. Top: cell initialization routines. Bottom: subroutines in every iteration of simulation loop. IBM Barrier time and MPI communication time are major performance bottlenecks. Long IBM Barrier time is due to synchronization time waiting on boundary ranks handling cell inlet and outlet routines.

To efficiently initialize large RBC populations, pre-packed tiles containing randomly oriented cells were used to populate the domain on the CPUs [9]. Cell overlaps were then communicated and resolved between tile boundaries and computational-task boundaries. To accommodate billions of packed cells, 64 bit integers were used for cell ID to prevent numerical overflow. After initialization, cell properties and associated data were transferred to GPUs for accelerated computation. During simulation, GPUs performed lattice Boltzmann collision and streaming, fluid-velocity interpolation, cell deformation calculations, and cell-force spreading to the fluid [5]. Meanwhile, CPUs managed distributed cell ownership and data communication.

IV. Towards Billion-scale RBC Modeling

We demonstrate billion-scale, explicitly resolved blood simulation on Frontier by packing and simulating 4.95 billion deformable RBCs within a 1mL blood volume, extending the tractable scale of explicit RBC modeling from millions to billions (Table I). This near-full-system calculation utilized 9,400 Frontier nodes, corresponding to 601,600 CPU cores and 75,200 AMD GPUs, with an aggregate memory footprint of 3.761 PB of CPU memory and 1.444 PB of GPU memory.

TABLE I
Extending the maximum RBC modeling scale from millions to billions.

Metric	Value
Resources used	601,600 CPU cores; 75,200 AMD GPUs
Blood volume	1 mL (1cm side length cube)
Resolution	1 μm
Fluid lattice points	999.4 billion
RBC population	4.95 billion
CPU memory usage	3.761 PB, 80% usage
GPU memory usage	1.444 PB, 30% usage

Strong-scaling experiments on a 1.1 mm cube demonstrated substantial reductions in runtime through 256 nodes, with particularly strong scaling of the coupled fluid–structure simulation loop (Fig. 1.). Weak-scaling studies spanning 588

to 9,408 nodes maintained a constant blood volume per rank by proportionally increasing the computational domain (Fig. 1.). Cell-packing initialization achieved approximately 90% parallel efficiency on both CPU and CPU-to-GPU data transfer, while the coupled simulation retained approximately 75% efficiency at full system scale (Fig. 1.). The major performance bottleneck lies in rank synchronization and extreme communication demands of billion-cell calculations (Fig. 2.). In addition, HARVEY presented excellent strong scaling and weak scaling performance across leadership supercomputers Frontier(AMD MI250X GPUs), Aurora(Intel Max 1550 GPUs), and Tuolumne(AMD MI300A GPUs). These results demonstrate that explicit deformable RBC simulation remains computationally efficient across systems with different GPU models.

This work establishes a new computational scale for explicit RBC modeling and demonstrates that physiologically relevant, cell-resolved blood simulation is now feasible on exascale systems. Beyond advancing computational hemodynamics, these capabilities provide a foundation for future studies of blood rheology, cellular transport, and multiscale cardiovascular disease at physiologically relevant scales.

V. Acknowledgment

This work used Frontier from Oak Ridge Leadership Computing Facility and Aurora from Argonne National Laboratory under DOE INCITE award, and Tuolumne from Lawrence Livermore National Laboratory under the LLNL Grand Challenge program. We sincerely thank the support from Dr.Antigoni Georgiadou and the Oak Ridge National lab. The research was funded by NIH R01EB024989, DP1AG082343, and NSF DGE 2139754. The content does not necessarily represent the official views of NIH and NSF.

References

[1] R. Wu, G. Ferreira, N. S. Khan, S. T. Mahmud, J. Stoop, L. L. Sohn, J. A. Leopold, and A. Randles, "Digital twins and digital models of the human circulatory system," *Nature Reviews Bioengineering*, pp. 1–20, 2026.

[2] A. Peters, S. Melchionna, E. Kaxiras, J. Latt, J. Sircar, M. Bernaschi, M. Bisson, and S. Succi, "Multiscale simulation of cardiovascular flows on the IBM Blue Gene/P: Full heart-circulation system at red-blood-cell resolution," in *Proc. 2010 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2010, pp. 1–10.

[3] A. Rahimian, I. Lashuk, S. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, *et al.*, "Petascale direct numerical simulation of blood flow on 200K cores and heterogeneous architectures," in *Proc. 2010 ACM/IEEE Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2010, pp. 1–11.

[4] L. Lu, M. J. Morse, A. Rahimian, G. Stadler, and D. Zorin, "Scalable simulation of realistic volume-fraction red blood cell flows through vascular networks," in *Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2019, pp. 1–30.

[5] J. Ames, D. F. Fuleri, P. Balogh, J. Gounley, E. W. Draeger, and A. Randles, "Multi-GPU immersed boundary method hemodynamics simulations," *J. Comput. Sci.*, vol. 44, Art. no. 101153, 2020.

[6] A. P. Randles, V. Kale, J. Hammond, W. Gropp, and E. Kaxiras, "Performance analysis of the lattice Boltzmann model beyond Navier–Stokes," in *Proc. IEEE 27th Int. Parallel and Distributed Processing Symp. (IPDPS)*, 2013, pp. 1063–1074.

[7] R. Skalak, A. Tozeren, R. Zarda, and S. Chien, "Strain energy function of red blood cell membranes," *Biophys. J.*, vol. 13, no. 3, pp. 245–264, 1973.

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

[9] S. Roychowdhury, E. W. Draeger, and A. Randles, "Establishing metrics to quantify spatial similarity in spherical and red blood cell distributions," *J. Comput. Sci.*, vol. 71, Art. no. 102060, 2023.