PII: S1877-7503(15)00041-1
DOI: http://dx.doi.org/doi:10.1016/j.jocs.2015.04.003
Reference: JOCS 346

To appear in:

Massively Parallel Simulations of Hemodynamics in the Primary Large Arteries of the Human Vasculature

Amanda Randles[1], Erik W. Draeger[1], and Peter E. Bailey[2]

[1] Lawrence Livermore National Laboratory, Livermore, CA, U.S.A.
randles2@llnl.gov, draeger1@llnl.gov
[2] University of Arizona, Tucson, Arizona, U.S.A.
pbailey@cs.arizona.edu

Abstract
We present a computational model of three-dimensional and unsteady hemodynamics within the primary large arteries in the human on 1,572,864 cores of the IBM Blue Gene/Q. Models of large regions of the circulatory system are needed to study the impact of local factors on global hemodynamics and to inform next generation drug delivery mechanisms. The HARVEY code successfully addresses key challenges that can hinder effective solution of image-based hemodynamics on contemporary supercomputers, such as limited memory capacity and bandwidth, flexible load balancing, and scalability. This work is the first demonstration of large fluid dynamics simulations of the aortofemoral region of the circulatory system at resolutions as small as 10 μm.

Keywords: lattice Boltzmann, computational fluid dynamics, high performance computing, patient-specific hemodynamics, strong scaling

1 Introduction

A longstanding goal within the field of computational biomechanics has been to understand the principles that govern vascular disease localization and progression[20, 5, 28]. Such image-based simulations can yield insight into the impact of local factors on global hemodynamics, direct the design of next-generation drug delivery mechanisms, and inform surgical planning. Although important progress towards this goal has been made using various algorithmic methods [11, 29, 12, 21, 13, 32], the computational demands of these simulations have historically restricted the resolution and size of the circulatory system that can be modeled.

In recent years, there has been a great deal of work in the area of computational hemodynamics. These studies are typically limited to small regions of the body or use a one-dimensional setting to describe the human arterial network [26, 1, 30]. Xiao *et al.* presented the first model of full unsteady and three-dimensional hemodynamics in the primary large arteries from head to foot. While this was a significant advance in computational fluid dynamics, the goal was to demonstrate the feasibility of the 3D framework. However, the resolution presented was insufficient to reach grid independence [32]. In that work, the finite element mesh consisted of 14,438,720 linear tetrahedra and 2,674,545 nodes. High resolution studies based on 3-D reconstructions of patient-specific data are typically focused on either the cerebral vasculature [9, 13, 7] or the cardiovascular region [3, 10, 19]. The current state of the art in the numerical investigation of hemodynamics in patient-specific geometries are those by Bernaschi *et al.* which studies the coronary arteries in a bounding box of 300 billion grid points containing one billion fluid nodes [3, 4]. Work presented here goes beyond these scales, simulating a vertical section of the aortofemoral section of the circulatory system spanning 614 cm at 10 μm resolution, consisting of more than 128 billion fluid nodes. To the best of our knowledge, our work is the first large-scale study of blood flow in a region of this size and level of detail.

Building realistic models of transport phenomena in the human circulatory system presents a formidable computational challenge due to the geometric complexity of the system, memory requirements associated with high-resolution grids, and load balancing issues associated with the processor core counts required. Our proposed solution extends the design and parallel efficiency of HARVEY [22], a computational fluid dynamics code based on the lattice Boltzmann method (LBM). One fundamental hurdle for high-resolution fluid simulations is the size of the underlying data grid and associated memory requirements. In order to study key macroscopic risk factors in patient-specific data, a resolution of at least 20 microns is required [18]. For full body simulations, this resolution corresponds to 18.4 billion fluid nodes in a bounding box of 8.8 trillion total grid points. These data sizes create an additional challenge to load balance, as work must be assigned to over one million tasks without computing or storing global data. We present a multi-step iterative load balance algorithm that allows for the distribution of large, complex arterial geometries on a 3D process grid. Efficiently modeling the hemodynamics in the large primary arteries also required data-reordering techniques to increase spatial locality, both optimized data structures and access patterns to reduce the overall memory footprint and efficient communication layout to overcome bandwidth limitations.

In this work we make the following contributions: increasing the number of fluid nodes that can be simulated by an order of magnitude (thereby increasing potential system size and/or resolution), incorporating preprocessing to reduce storage and I/O burdens, and enabling an unprecedented scale of hemodynamic simulation demonstrated by the 10 micron resolution simulation of the aortofemoral region of the circulatory system.

2 Methodology

This work relies on the lattice Boltzmann method (LBM), an alternative to the conventional Navier-Stokes equation, introduced by both teams of McNamera and Zanetti [17] and Higuera and Jimenez [15]. LBM comes from kinetic theory and is a minimal form of the Boltzmann equation based on the collective dynamics of fictitious particles that represent a local ensemble of molecules moving between the points of a regular Cartesian lattice. The time advancement is explicit and the computational stencil is formed by local neighbors of each computational node, making it particularly well-suited for massively parallel simulations (c.f. [6, 31, 23]).

The governing equation describes the evolution of the distribution function denoted by $f_i(\vec{x}, t)$, describing the probability of finding a particle at grid point $\vec{x}$, at time t, with discrete velocity $\vec{c}_i$. In this work, we use the 19-speed cubic stencil, with the Bhatnager-Gross-Krook (BGK) collision formulation with a single relaxation time. The grid spacing is defined by Δx, where discrete velocities connect grid points to first and second neighbors on the 19-point stencil. The fluid populations are advanced in a timestep Δt through:

$$f_i(\vec{x}+\vec{c}_i\Delta t, t+\Delta t) = f_i(\vec{x},t) - \omega\Delta t[f_i(\vec{x},t) - f_i^{eq}(\vec{x},t)] \tag{1}$$

The local equilibrium, $f_i^{eq}(\vec{x},t)$, is the result of a second-order expansion in the fluid velocity of a local Maxwellian with speed $\vec{u}$ and is defined by:

$$f_i^{eq} = w_i\rho\left[1+\frac{\vec{c}_i\cdot\vec{u}}{c_s^2}+\frac{1}{2}\left(\frac{(\vec{c}_i\cdot\vec{u})^2}{(c_s^2)^2}-\frac{u^2}{c_s^2}\right)\right] \tag{2}$$

where ρ denotes the density, $\vec{u}$ the average fluid speed, $c_s = 1/\sqrt{3}$ the speed of sound in the lattice, and w_i the weights attributed to each discretized velocity as determined by the lattice structure. Due to the use of explicit time-stepping, LBM requires small time-steps that scale with Δx^2. In the case of the 10 μm simulations discussed in this work, approximately 3 million time-steps would be required to simulate one cardiac cycle.

We implement the Zou-He boundary conditions [33], in which a pulsating velocity is imposed at the inlet through a *plug profile* at the entrance to the vessel and a constant pressure is imposed at the outlets. While the inlet condition does not assert the known parabolic profile that drops to zero close to the wall, it allows a total flow to be imposed at a set value. In a short distance past the inlet, the parabolic profile is recovered. This method uses information streamed from the bulk fluid nodes alongside a completion scheme for the unknown particle populations whose neighbors are outside the fluid domain. This method can be executed with second-order accuracy [16]. In this paper, the modification introduced by Hecht and Harting [14] is used in which the velocity conditions are specified on-site, thus removing the constraint that all nodes of a given inlet or outlet must be aligned on a plane that is perpendicular to one of the three main axes. Furthermore, this addition allows the boundary conditions to be applied locally. A no-slip boundary condition is imposed at the walls via the full bounce-back method. For more details regarding the lattice Boltzmann method, see Ref. [27].

3 HARVEY implementation details

All simulations presented here were carried out using the HARVEY code. Despite the excellent scalability reported previously[24], significant restructuring of the code had to be done to enable the resolution and scale of the systems studied in this work. Details of the original implementation can be found in Ref. [22]. All simulations were run on the Sequoia machine at Lawrence Livermore National Laboratory, a 98,304 node IBM Blue Gene/Q machine (1,572,864 cores).

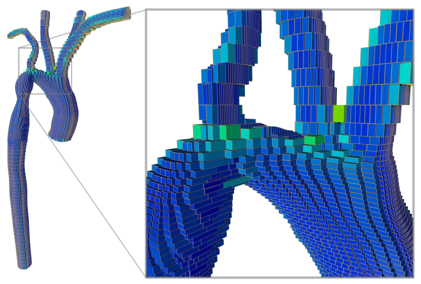

Figure 1: Bounding boxes of aorta geometry computed by multi-step load balance algorithm.

In order to simulate the hemodynamics in the aortofemoral geometry at a high-resolution, we had to overcome the following challenges:

- Memory footprint. Large numbers of grid points are required to reach convergence of macroscopic quantities of interest. These requirements impose a high demand in terms of on-node memory requirements.
- I/O bandwidth. Setting up the large, high-resolution grids through the existing preprocessing and initialization stages involves I/O operations on petabytes of data, causing the simulation setup to actually overwhelm the overall simulation time even for the large number of time steps, $\sim 10^6$, needed to model a full cardiac cycle.
- Scalability. As each Blue Gene/Q core has only 1 GB of available total memory, the scope of this problem requires use of the entire LLNL Sequoia system of 1,572,864 cores. Effective utilization of such a large core count means that traditional parallelization tools like global communication tables are not feasible options.
- Load imbalance. The geometry of the human vasculature is incredibly complex. The bounding box holds grid points representing fluid, inlets, outlets, walls, and exterior points. Distributing the workload across hundreds of thousands to millions of cores requires careful attention to load balance.

We address these challenges by extending the capability of HARVEY through (i) embedding of preprocessing and use of buffered meshing to avoid global bottlenecks and reduce I/O stress, (ii) the introduction of indirect addressing to reduce the memory footprint, (iii) development of a multi-step load balancing scheme that prioritizes locality and memory reduction, (iv) removal of global communication tables to improve scalability.

3.1 Parallel Preprocessing

The original implementation of HARVEY used multiple preprocessing steps to construct the 3D spatial grid from the surface mesh and set up neighbor lists and communication tables. This strategy becomes infeasible at the target scales of this work, as the full 3D mesh must be read from disk and distributed across tasks, creating both an I/O and memory bottleneck. Instead, we have integrated these routines into HARVEY so that only the surface mesh is used in the initial load balance and the volume grid can be generated in place on local MPI tasks.

Communication tables can be generated during setup using only local information due to the use of a structured process grid, i.e. tasks need only talk to their process grid neighbors to discover who owns fluid nodes in their stencil.

3.2 Minimizing Memory Footprint

One fundamental challenge to high-resolution lattice Boltzmann simulations of large arterial geometries is the size of the underlying data grid and the associated storage requirements, particularly during the setup and load balance phases of the calculation. For smaller scale calculations, it is possible to store global information to simplify the construction of the grid from the surface mesh, build communication tables, and load-balance the workload across tasks. At larger length scales, however, such storage is no longer feasible. For example, the bounding simulation box of the aortofemoral geometry shown in Fig. 2a has physical dimensions 168 x 115 x 614 mm, which corresponds to a data grid of 33,530 x 22,992 x 122,808 grid points at 5 μm resolution. An integer array of the node types for a single xy-plane would consume 3.1 GB, over three times the total memory available to a single Blue Gene/Q core.

In the full simulation box, however, only a small subset of grid points actually represent fluid points. As shown in Table 4, the cerebral vasculature exhibits a 0.412% fluid density, the aorta 2.36%, and the aortofemoral 1.08%. In such cases where the volume of the flow domain makes up a small percentage of the bounding box, maintaining the entire domain in memory leads to a large waste of storage and can lead to a high degree of load imbalance. To avoid such issues we use an efficient data structure based on indirect addressing which allows nearly arbitrary geometries to be handled at a minimal additional cost. Similar to the topologically unstructured grid introduced by Schulz *et al.* [25], we only store the locations of grid points that represent boundaries or fluid. During the preprocessing routine, we calculate and store the location and node type of each neighbor by relying on the underlying stencil organization. This procedure results in an 18-degree stencil that contains the necessary information about each neighbor being stored for each non-exterior point. This can be used to establish the communication tables discussed in the following sections. The use of indirect addressing means that the distribution function size is minimized and equal to the product of the number of local fluid nodes (N_{loc}) and number of stencil points ($N_{stencil}$).

3.3 Improving Load Balance

In HARVEY, we apply a three-dimensional Cartesian grid across the simulation box. Grid points are then classified as fluid, inlet, outlet, wall, or dead, i.e. those falling outside the mesh. This grid of fluid and wall nodes is constructed at runtime from a triangular surface mesh supplied as input. Overlapping slices of the triangular mesh are distributed across z planes of a 3D process grid. Within these slices, the grid points just inside the mesh are determined from the angle-weighted pseudo-normals[2] of the closest triangle. This defines a shell of grid points that border each side of the mesh. The rest of the interior nodes are filled in by sweeping across a single grid dimension and assigning all unidentified nodes between known interior points as fluid nodes. Any interior point that borders a point outside of the mesh is labeled as a wall node. Because only a single full grid dimension is needed to identify which nodes are inside the mesh, we can buffer this calculation and distribute the results to avoid memory bottlenecks.

Load balance is handled iteratively and work is computed as a function of each Cartesian direction. The algorithm prioritizes locality and limited memory usage by avoiding global data as much as possible, using a multi-step approach:

1. Estimate work of each xy-plane using mesh boundaries.
2. Assign ownership of xy-planes to process planes.
3. Read triangular mesh and compute local grid points.
4. Compute total work of each xy-plane.
5. Reassign ownership of xy-planes, recompute local grid points.
6. For each local xy-plane, compute work as a function of y.
7. Assign y-strips of local grid points to y-strips of tasks.
8. Distribute local strips across tasks in x-direction.

The load balancer currently uses a simple work model that is proportional to the number of fluid nodes on a task, but any performance model could be used, provided the estimated work can be computed from local data. The final two steps (distributions in the y- and x- directions) assign tasks in groups when multiple spatially-disconnected sections of work are detected. This prevents the creation of large local bounding boxes that span different arterial branches and which could cause a memory bottleneck.

3.4 Reduced Memory Stencil Operations

Once load balance is complete, each task "owns" the fluid nodes within its rectangular bounding box and computes the particle distribution function for these nodes at every time step. Determining the neighbors of a given fluid node could be done once and stored, but would require storage equal to the distribution function itself. To save memory, rather than storing an explicit neighbor list for all stencil points of each fluid node, we store the array index of each fluid point on a regular grid throughout the bounding box, with non-fluid nodes indicated by negative values. Stencil operations can then be applied to this bounding box array to determine the indices of the corresponding fluid nodes and which ones border wall, inlet or outlet nodes. This reduces the local memory requirement of neighbor calculations by a factor of $f_{bb}N_{stencil}$, where f_{bb} is the fraction of the bounding box occupied by fluid nodes.

Communication tables are computed during initialization and stencil points owned by neighboring tasks updated every iteration using non-blocking MPI Isend and Irecv calls. To maximize cache performance and data reuse, local data is sorted to group sent points and received points to contiguous array locations.

4 Simulations

To demonstrate the scalability of the code in a range of regimes, we chose three different simulation geometries: aortofemoral (Fig. 2a), aorta (Fig. 2b), and cerebral (Fig. 2c). Patient-specific volumetric image data was obtained via CT imaging. The aortofemoral is the vasculature of a 21 year-old female. The cerebral vasculature was that of a 31 year-old female and the aorta was from an 8 year-old female. Data was obtained from the Open Source Medical Software Corporation. The geometry was created by identifying the centerline paths through the vessels and connecting series of 2D segmentations along these lines. Each vessel was constructed individually and a Boolean addition used to combine the vessels into one model.

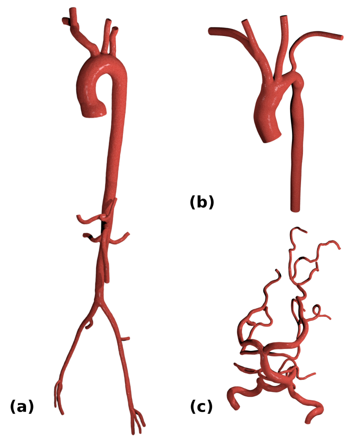

Figure 2: Mesh geometries used as HARVEY input: (a) Aortofemoral, (b) Aorta, and (c) Cerebral.

The three different geometries represent systems of varying size, surface-fluid-ratio, and complexity (in terms of the number of inlets and outlets). Table 4 demonstrates the data size associated with each geometry. This emphasizes the overall data demands of such high-resolution simulations. For the aortofemoral system, the overall bounding box has the dimensions of 16765

	Cerebral	Aorta	Aortofemoral
20 micron resolution			
fluid nodes	607,924,802	3,175,878,044	16,033,887,284
data grid	4288 x 4951 x 6955	5375 x 2719 x 9212	8383 x 5748 x 30702
fluid fraction	0.412%	2.36%	1.08%
10 micron resolution			
fluid nodes	4,922,115,786	25,502,717,509	128,666,443,295
data grid	8575 x 9902 x 13909	10750 x 5437 x 18424	16765 x 11496 x 61404
fluid fraction	0.412%	2.36%	1.08%

Table 1: Computational details of the cerebral, aorta and aortofemoral geometries.

$\times$ 11496 $\times$ 61404 at 10 micron resolution. For a lattice Boltzmann model requiring two 8-byte doubles for each stencil direction, this would require 3.6 Petabytes of storage capacity. It is also important to note that increasing the grid resolution from 20 micron to 10 micron results in an 8-fold increase in the number of grid points required.

Experiments have shown that the shear rate observed in vessels of the sizes studied here is in the range in which the elastic behavior of blood becomes insignificant. As such, in this work, blood is considered to be a Newtonian, isotropic, and homogenous fluid [8]. The viscosity is assumed to have a value of 0.04 g/(cm s) and the density of blood is taken as 1.06 g/cm^3. A rigid-wall approximation is used.

5 Performance and Scalability

In this section, we present strong scaling measurements on the full 1.5 million core LLNL Sequoia Blue Gene/Q machine. All simulations were run with 16 MPI tasks per 16-core Blue Gene/Q compute node and one thread per task. Results for all three geometries are shown in Fig. 3. The average iteration time was computed from the maximum time spent by any task in the main lattice Boltzmann iteration loop. For consistency, all I/O beyond simple standard output was disabled. In cases where multiple MPI process grids were used with the same number of nodes, only the result with the fastest overall time-to-solution is shown.

All systems showed excellent strong scalability, with parallel efficiencies ranging from 34% (519-fold speedup over a 1536x increase in task count) for the 20 μm cerebral geometry to 96% (11.5-fold speedup over a 12x increase in task count) for the 10 μm aortofemoral geometry. Communication time was a relatively small fraction of the total iteration time, ranging from 0.4% to 19% depending on the system size and resolution (see Fig. 4). The significant increase in relative communication time cannot simply be explained by the inevitable increase in the surface-to-volume ratio of processor domains as the total number of tasks increases, which to first order will only scale as the cube root of the total number of tasks. This model would predict a maximum communication fraction of 5% for the 20 μm cerebral geometry rather than the 19% we observe, indicating that other factors such as load imbalance are responsible for the significant increase in communication time at scale.

For systems of a few hundred thousand MPI tasks and below, the load imbalance was below 20%, but became substantial at full scale, e.g. as large as 96% in the 20 μm cerebral geometry. We define load imbalance as the difference between the average time and the maximum time

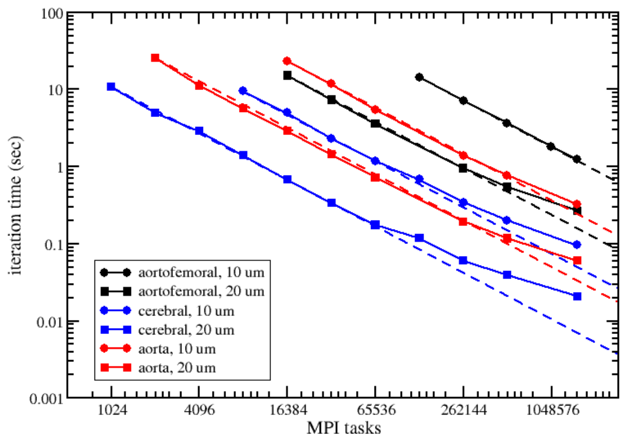

Figure 3: Strong scaling of cerebral, aorta and aorta-femoral geometries at 10 and 20 μm resolutions. Dashed lines indicate perfect scaling from the smallest task count.

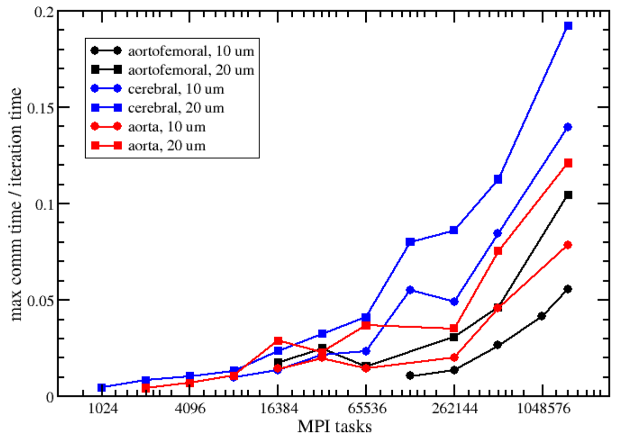

Figure 4: Maximum communication time as a fraction of total iteration time of cerebral, aorta and aorta-femoral geometries at 10 and 20 μm resolutions.

spent in the iteration loop, normalized by the average iteration time:

$$\lambda = \left(\frac{t_{max}}{t_{avg}} - 1 \right) \tag{3}$$

Therefore, although one might initially conclude from Figures 3 and 4 that the code is becoming communication-bound at full scale, Figure 5 shows that load imbalance accounts for the majority of the deviation from ideal strong scaling. Moreover, the similar shapes of Figures 4 and 5 indicate that the tasks with the highest overall workload correspond to those with the highest communication volume. This is likely exacerbated by the fact that the load balance algorithm currently does not take into account the differences in total communication

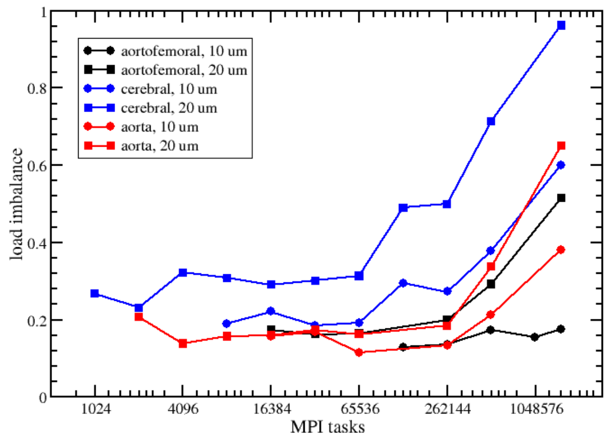

Figure 5: Load imbalance of cerebral, aorta and aorta-femoral geometries at 10 and 20 μm resolutions.

volume between tasks with many neighbors (e.g. in the middle of arteries) and tasks with relatively few (e.g. at edges or on smaller branches). Although the overall performance is quite good for a lightweight load balancer on these geometries at this scale, we anticipate that further performance gains can be realized by integrating a more robust performance model into the load balancer.

In addition to improving load balance, we plan to further decrease the overall time-to-solution by implementing newly developed data access patterns to reduce the cost of data movement in bandwidth-bound regions of the code. We have also begun developing optimized kernels to exploit specialty hardware such as SIMD vector units and will explore whether similar kernels can be used to make efficient use of heterogeneous architectures, e.g. GPU machines.

6 Conclusion

We presented computational advancements to the HARVEY code designed to enable scalable simulations of large, high-resolution arterial geometries. Our results show strong scalability for three different systems on 1.5 million Blue Gene/Q cores, where parallel efficiencies of 35% to 96% were observed. As this work is the first direct simulation of a significant fraction of the full circulatory system carried out at resolutions as high as 10 μm, we believe that it will set the stage for the next generation of high-fidelity hemodynamics simulations.

7 Acknowledgments

The authors would like to thank Liam Krauss at LLNL for visualization and analysis work, and the staff at Livermore Computing for system support. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] Jordi Alastruey, Ashraf W Khir, Koen S Matthys, Patrick Segers, Spencer J Sherwin, Pascal R Verdonck, Kim H Parker, and Joaquim Peiró. Pulse wave propagation in a model human arterial network: Assessment of 1-d visco-elastic simulations against¡ i¿ in vitro¡/i¿ measurements. *Journal of biomechanics*, 44(12):2250–2258, 2011.

[2] J.A. Baerentzen and H. Aanaes. Signed distance computation using the angle weighted pseudonormal. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):243–253, 2005.

[3] M. Bernaschi, M. Bisson, T. Endo, S. Matsuoka, M. Fatica, and S. Melchionna. Petaflop biofluidics simulations on a two million-core system. In *Proceedings of the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 4. ACM, 2011.

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

[5] CG Caro, TJ Pedley, RC Schroter, WA Seed, and KH Parker. *The mechanics of the circulation*, volume 192633236. Oxford University Press Oxford, 1978.

[6] J. Carter, M. Soe, L. Oliker, Y. Tsuda, G. Vahala, L. Vahala, and A. Macnab. Magnetohydrodynamic turbulence simulations on the earth simulator using the lattice Boltzmann method. In *Proceedings of the 2005 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '05. IEEE Computer Society, 2005.

[7] J.R. Cebral, M.A. Castro, J.E. Burgess, R.S. Pergolizzi, M.J. Sheridan, and C.M. Putman. Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics models. *American Journal of Neuroradiology*, 26(10):2550–2559, 2005.

[8] S. Chien, S. Usami, H.M. Taylor, J.L. Lundberg, and M.I. Gregersen. Effects of hematocrit and plasma proteins on human blood rheology at low shear rates. *J Appl Physiol*, 21(1):81–87, 1966.

[9] Paolo Di Achille and Jay D Humphrey. Toward large-scale computational fluid-solid-growth models of intracranial aneurysms. *The Yale journal of biology and medicine*, 85(2):217, 2012.

[10] David JW Evans, Patricia V Lawford, Julian Gunn, D Walker, DR Hose, RH Smallwood, B Chopard, M Krafczyk, J Bernsdorf, and A Hoekstra. The application of multiscale modelling to the process of development and prevention of stenosis in a stented coronary artery. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1879):3343–3360, 2008.

[11] D.J.W. Evans, P.V. Lawford, J. Gunn, D. Walker, D.R. Hose, RH Smallwood, B Chopard, M Krafczyk, J Bernsdorf, and A Hoekstra. The application of multiscale modeling to the process of development and prevention of stenosis in a stented coronary artery. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1879):3343–3360, 2008.

[12] C Alberto Figueroa, Irene E Vignon-Clementel, Kenneth E Jansen, Thomas JR Hughes, and Charles A Taylor. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. *Computer methods in applied mechanics and engineering*, 195(41):5685–5706, 2006.

[13] L. Grinberg, V. Morozov, D. Fedosov, J.A. Insley, M.E. Papka, K. Kumaran, and G.E. Karniadakis. A new computational paradigm in multiscale simulations: Application to brain blood flow. In *Proceedings of the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12. IEEE, 2011.

[14] M. Hecht and J. Harting. Implementation of on-site velocity boundary conditions for D3Q19 lattice Boltzmann simulations. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(01):P01018, 2010.

[15] FJ Higuera, S. Succi, and R. Benzi. Lattice gas dynamics with enhanced collisions. *EPL (Europhysics Letters)*, 9(4):345–349, 1989.

[16] O. Malaspinas, B. Chopard, and J. Latt. General regularized boundary condition for multi-speed lattice Boltzmann models. *Computers & Fluids*, 49(1):29–35, 2011.

[17] G.R. McNamara and G. Zanetti. Use of the Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61(20):2332–2335, 1988.

[18] S. Melchionna, J. Lätt, E. Kaxiras, A. Peters, M. Bernaschi, and S. Succi. Endothelial shear stress from large-scale blood flow simulations. In *Proceedings of Fifth European Conference on Computational Fluid Dynamics*, ECCOMAS CFD'10, 2010.

[19] K. Pekkan, B. Whited, K. Kanter, S. Sharma, D. De Zelicourt, K. Sundareswaran, D. Frakes, J. Rossignac, and A.P. Yoganathan. Patient-specific surgical planning and hemodynamic computational fluid dynamics optimization through free-form haptic anatomy editing tool (surgem). *Medical & Biological Engineering & Computing*, 46(11):1139–1152, 2008.

[20] Charles S Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

[21] A. Peters, S. Melchionna, E. Kaxiras, J. Lätt, J. Sircar, M. Bernaschi, M. Bison, and S. Succi. Multiscale simulation of cardiovascular flows on the IBM Blue Gene/P: Full heart-circulation system at red-blood cell resolution. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '10, 2010.

[22] A. Peters Randles, V. Kale, J.R. Hammond, W. Gropp, and E. Kaxiras. Performance analysis of the lattice Boltzmann model beyond Navier-Stokes. In *Proceedings of the 27th IEEE International Parallel and Distributed Processing Symposium*, IPDPS '13, 2013.

[23] T. Pohl, F. Deserno, N. Thurey, U. Rude, P. Lammers, G. Wellein, and T. Zeiser. Performance evaluation of parallel large-scale lattice Boltzmann applications on three supercomputing architectures. In *Proceedings of the 2004 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '04. IEEE Computer Society, 2004.

[24] A. Randles and E. Kaxiras. A spatio-temporal coupling method to reduce the time-to-solution of cardiovascular simulations. In *Proceedings of the 28th IEEE International Parallel and Distributed Processing Symposium*, IPDPS '14, 2014.

[25] M Schulz, M Krafczyk, J Tölke, and E Rank. Parallelization strategies and efficiency of cfd computations in complex geometries using lattice boltzmann methods on high-performance computers. In *High performance scientific and engineering computing*, pages 115–122. Springer, 2002.

[26] N Stergiopulos, DF Young, and TR Rogge. Computer simulation of arterial flow with applications to arterial and aortic stenoses. *Journal of biomechanics*, 25(12):1477–1488, 1992.

[27] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Oxford University Press, 2001.

[28] C.A. Taylor, T. Hughes, and C.K. Zarins. Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering*, 158(1):155–196, 1998.

[29] D.A. Vorp, D.A. Steinman, and C.R. Ethier. Computational modeling of arterial biomechanics. *Computing in Science & Engineering*, 3(5):51–64, 2001.

[30] Nicolaas Westerhof, Frederik Bosman, Cornelis J De Vries, and Abraham Noordergraaf. Analog studies of the human systemic arterial tree. *Journal of biomechanics*, 2(2):121–143, 1969.

[31] S. Williams, L. Oliker, J. Carter, and J. Shalf. Extracting ultra-scale lattice Boltzmann performance via hierarchical and distributed auto-tuning. In *Proceedings of the 2011 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 1–10. IEEE Computer Society, 2011.

[32] N. Xiao, J.D. Humphrey, and C.A. Figueroa. Multi-scale computational model of three-dimensional hemodynamics within a deformable full-body arterial network. *Journal of Computational Physics*, 244:22–40, 2013.

[33] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids*, 9:1591, 1997.