A Lattice Boltzmann Simulation of Hemodynamics in a Patient-Specific Aortic Coarctation Model

Amanda Peters Randles*, Moritz Bächer, Hanspeter Pfister, and Efthimios Kaxiras

School of Engineering and Applied Sciences,
Harvard University
Cambridge, MA, USA 02138
`apeters@fas.harvard.edu`
`hemo.seas.harvard.edu`

Abstract. In this paper, we propose a system to determine the pressure gradient at rest in the aorta. We developed a technique to efficiently initialize a regular simulation grid from a patient-specific aortic triangulated model. On this grid we employ the lattice Boltzmann method to resolve the characteristic fluid flow through the vessel. The inflow rates, as measured physiologically, are imposed providing accurate pulsatile flow. The simulation required a resolution of at least 20 microns to ensure a convergence of the pressure calculation. HARVEY, a large-scale parallel code, was run on the IBM Blue Gene/Q supercomputer to model the flow at this high resolution. We analyze and evaluate the strengths and weaknesses of our system.

Keywords: computational fluid dynamics, coarctation of the aorta, lattice boltzmann, parallel computing.

1 Introduction

Coarctation of the aorta (CoA) can pose a significant problem as this narrowing of the aorta can inhibit blood flow through the artery. CoA accounts for 8%-11% of congenital heart defects, which makes it affect tens of thousands of patients annually in the Western world [1]. Hence, there is a need to efficiently diagnose the degree of arterial narrowing so that preventative action such as balloon angioplasty or stent implantation can be taken [2]. These methods serve to alleviate the pressure gradient through the coarctation in order to reduce the burden on the heart.

An obstruction is characterized as significant when the peak to peak systolic pressure gradient across the coarcted vessel is measured at greater than 20 mmHg. This pressure gradient is not only determined by the size of the narrowing but also factors such as the flowrate of the fluid and the size, number, and placement of collateral vessels [2]. The physiological state of the patient can also

* Corresponding author.

© Springer-Verlag Berlin Heidelberg 2013

contribute to an increase in the pressure gradient if, e.g., the patient is in an exercised state due to the associated elevation in heart rate. When the patient is at rest, clinicians can easily measure the pressure gradient; however, this measurement is difficult to obtain under exercise conditions. This difficulty causes simulation to play a key role in determining the pressure gradient non-invasively.

In this paper, we discuss a method for using data from medical imaging alongside a lattice Boltzmann fluid model to simulate blood flow and pressure in a model built from patient data. We will present the methods to impose a regular grid and model the fluid flow using parameters provided from patient data. We use the provided inflow waveform to produce realistic pulsatile flow that upholds the measured flow distribution via velocity imposed boundary conditions at the inlets and outlets.

2 Data

The data used in this paper was provided by the STACOM CFD Challenge for the simulation of hemodynamics in a patient-specific aorta coarctation model. The geometry of the vessels were obtained through gadolinium-enhances MR angiography (MRA) with a 1.5-T GE Signa scanner. A segmented STL file was provided defining the ascending aorta, arch, descending aorta, and upper branch vessels. Flow rates were measured by PC-MRI sequence encoding and provided for the course of a cardiac cycle as well as the percent of flow seen in each branch of the model [1].

3 Computational Fluid Dynamics (CFD) Framework

In this work, we use the lattice Boltzmann method (LBM) as the basis of our simulation [3]. The LBM has proven to be a strong alternative to simulations derived from the Navier-Stokes equation of continuum mechanics. In recent years, its ease of handling complex geometries and parallelization has made it increasingly popular. Unlike conventional CFD methods based on the discretization of macroscopic continuum equations, the LBM constructs a simplified kinetic model incorporating the essential physics and preserving macroscopically averaged quantities like mass and momentum [4].

In this model, the volume of a 3-dimensional mesh is filled with a regular array of lattice points on which a minimal form of the classical Boltzmann is simultaneously solved for a set of fictitious particles [3]. These particles represent the collective motion of a group of physical particles and the dynamics are such that they ensure hydrodynamic behavior in the continuum limit.

3.1 Algorithm

The fundamental quantity is the probability density function defining the likelihood of finding particles at a specific location, at a specific time, traveling along a specific velocity path. In this work, we use the 19-speed velocity model, D3Q19, in which the discrete velocities, c_i, connect lattice points to the first and second topological neighbors [4]. The distribution function is advanced through (1) [3].

In each time step, the particles advect along the straight trajectories defined by the discretized velocities. Fluid-fluid collisions are then handled through a relaxation towards a local equilibrium, shown on the right side of (1). In this work, we leverage the Bhatnager-Gross-Krook (BGK) operator, a collision operator which relaxes to equilibrium on a single time scale [5]. The equilibrium distribution is defined through a second-order Hermite expansion of a local Maxwellian with density ρ and speed u as shown in (2) [6]. The relaxation frequency ω is related to the kinematic viscosity of the fluid through (3) [7].

$$f(x + c_i\Delta t, t + \Delta t) = f(x, t) - \omega\Delta t(f(x, t) - f^{eq}(x, t)). \tag{1}$$

$$f_i^{eq} = w_i\rho\left\{1 + \frac{\xi_i \cdot u}{c_s^2} + \frac{uu : (\xi\xi - c_s^2 I)}{2c_s^2}\right\}. \tag{2}$$

$$\nu = c_s^2\Delta t(\frac{1}{\omega} - \frac{1}{2}) \tag{3}$$

In (2), w denotes the quadrature weight normalized to unity and the speed of sound is a lattice constant: $c_s^2 = \frac{1}{3}$. I is the unit tensor in Cartesian space.

A key advantage of the LBM is that macroscopic quantities such as density are moments of the distribution function. This means that they can be calculated based on its summation and therefore are available entirely locally. In the study of CoA, the fluid pressure is particularly important.

Pressure can be easily recovered through the ideal gas relation: $P = c_s^2\rho$. This means that the value is available locally which is particularly advantageous as this means they do not require solving an expensive Poisson problem as in other CFD methods [8].

3.2 Boundary Conditions

As prescribed by the challenge definition, we employ rigid walls in this simulation. At the wall, we impose a no-slip boundary condition through the use of a full bounce-back method. To this end, the velocity of any particle which is set to advect to a lattice point designated as a wall node is reversed. In this case, the directions of post-collisional particles are reversed if the prescribed velocity points to a lattice point designated as a wall node as shown in Fig. 1. The curved vessels are shaped on the regular (axis-aligned) grid via a staircase representation as opposed to the body-fitted grids found in direct Navier-Stokes solvers. This does come at the expense of numerical accuracy, which has been shown to degrade to first order [3]. This representation is improved systematically by increasing the resolution of the mesh via increased density of lattice points.

In this model, there is one inlet for the aorta and multiple outlets for each of the collateral vessels. The imposition of the boundary conditions is based on the knowledge of local flow profiles as provided by the measured patient-specific data

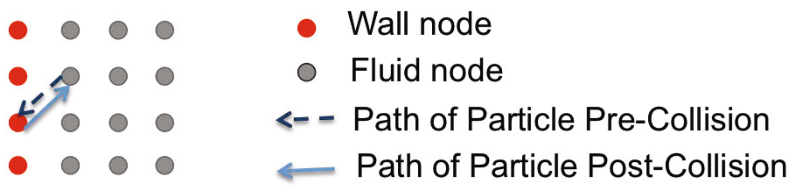
Wall node
Fluid node
Path of Particle Pre-Collision
Path of Particle Post-Collision

Fig. 1. Bounceback boundary condition. The no-slip boundary is enforced by reversing the direction of each particle just inside the wall boundary.

of this challenge. We employ a simple method of imposing a plug flow profile based on the flow rates. The inlet condition comes from the aortic flow measured by a phase-constract (PC) MRI sequence providing the inlet flow over the course of one cardiac cycle [1]. Any node defined as an inlet node has it's velocity set based on the time point in the simulation ensuring proper pulsatile flow that matches the measured data.

For the outlet condition, flow rate is determined based on the percent of flow through the various branches in the model as measured with the PC MRI as well. This allows us to calculate the flow rate at each branch and subsequent outlet. For most of the vessels, we are thus able to use clinically measured values for the outflows. In order to determine flow going through the right subclavian and right common carotid outlet faces, we use the empirical procedure based on the following assertion from previous studies: "in the coronary system we assume the physiological condition that the pressure drop in each vessel is driven by the oxygen request from the tissues nourished by the vessel" [8]. That means that we use flow splitting conditions of $\phi_1/\phi_2 = S_1/S_2$ in which ϕ_1 and ϕ_2 denote the outgoing flow rates and S_1 and S_2 the corresponding sectional areas. Coupling the incoming flow rate with the known flow splitting at each bifurcation allows us to impose consistent outflow conditions.

4 Simulation

The simulation of blood flow in the patient specific data involves the following five steps:

1. Acquisition of medical imaging data
2. Image segmentation to identify vessel geometry
3. Grid initialization
4. Flow simulation
5. Data analysis and simulation

In this work, the first two steps were provided by the competition framework. In this section, we illustrate how we handle (3) as a pre-processing routing, step (4) with our HARVEY code, and (5) with a separate visualization routine coupled with the use of Paraview [9].

4.1 Initializing the Regular Simulation Grid

To guarantee a proper initialization of our simulation grid, we require the patient's triangulated vessel geometry to be a closed, 2-manifold with no overlaps of interior volume. We start by computing the axis-aligned bounding box (AABB) of the input geometry offset by ε (we use $\varepsilon = 1.8c_i\Delta t$) on each side, then discretize the box's volume into a regular grid of targeted resolution. Note that we choose ε to be slightly bigger than the length of the diagonal of a regular grid cube ($\varepsilon > \sqrt{3}c_i\Delta t$). With this choice, we guarantee that every interior grid point has a neighbor not only in any 6-neighborhood but in any diagonal grid direction also.

Next, we classify each grid point to either be inside or outside of the given vessel geometry. Note that it would be prohibitively slow to run an inside-outside test for each individual grid point. We therefore correctly initialize the grid points falling inside the union of all sphere-swept triangles (the volume of a sphere-swept triangle is given by the union of all spheres of radius ε with centers on the triangle) and then "fill in" the inside-outside classification for all remaining grid points. More specifically, we iterate over all vessel triangles: for each triangle, we compute the grid points that overlap with its AABB offset by ε and then check them against its sphere-swept bounding volume. For the remaining grid points, we then compute the closest point on the triangle and – if the point hasn't been initialized yet or the current point is closer to the vessel geometry than the previously initialized one – classify it as either inside or outside using the angle weighted pseudonormal approach by Bærentzen and Aanæs [10]. Note that [10] guarantees a correct inside-outside classification for points with respect to non-convex geometries provided that we know their closest points on the mesh. After a run through all triangles, we can guarantee to find the correct closest points for all grid points falling in the union of the sphere-swept triangles, hence, to correctly classify them using [10]. By using a radius of ε to sweep the triangles, we can further guarantee to correctly initialize at least two inside grid points in any 6-neighborhood and diagonal grid direction within a distance ε of the vessel's boundary. Finally, we "fill in" the remaining grid points by looping over the three grid indices (that are monotonously decreasing or increasing in their respective dimension) and classifying all grid points as inside if the loop index of the most inner loop has crossed the grid boundary an odd number of times.

This classification is then refined into wall, inlet, and outlet points by using the grid point's 6-neighborhood (if at least one of its six neighbors are classified as outside, we have a wall point) together with proximity information to inlet or outlet triangles. The remaining grid points are either "fluid" nodes (inside the wall), or "dead" (outside and not considered).

4.2 Flow Simulation

The HARVEY simulation package is designed to handle complex geometries and to run large-scale simulations on high performance hardware resources. It has been developed from the ground up with parallel efficiency in mind to enable high resolution runs. The mesh is Cartesian which enables straightforward data handling. It is written in C and uses MPI as the communication library. This code takes advantage of optimizations such as a) hand loop unrolling b) Single Instruction, Multiple Data (SIMD) intrinsics c) removal of redundant operations d) non-blocking communication [11]. The domain is split such that each processor handles a set division of the Cartesian mesh. In HARVEY a double buffer approach is used in which a starting distribution of fluid particles is initialized for each lattice point. The advection step propagates the particles to adjacent lattice points and stores these values in a temporary distribution function. This is the data exchanged with the neighboring processors, and subsequently used for the collision step. The result of the collision step is then used to update the local portion of the original array containing the distribution function for each lattice point. In this manner, HARVEY acts as a typical stencil code that draws information from its neighbors, updates its local value, and pushes this data to the neighbors, however, the data accessed in the temporary data structure is from another phase space as well as from another lattice location.

This double buffer approach further increases the already large memory demand of the simulation. In the case of this data set when simulated at a 200 micron resolution, there are 64,435 fluid voxels in a bounding box of 11,254,320 voxels. For each lattice point, there are two buffers that make up the bulk of the memory requirements. These buffers store the density data for each discrete velocity at each lattice point as a floating point number. For a 200 micron resolution simulation, this requires at least three gigabytes of memory. While some commodity desktops may now be able to meet the memory needs for 200 microns, this becomes increasingly difficult at finer resolutions. For a 20 micron simulation, 3 terabytes of data are necessary. This is feel beyond the capabilities of traditional computers and requires the use of large-scale platforms such as the IBM Blue Gene/Q described in a following section, especially when simulating full heartbeats.

The second issue is the runtime for the simulation to complete. At high resolutions, the LBM requires rather small time steps on the order of 10^{-6} seconds resulting in the need for 700,000 time steps to complete one heartbeat. The computation of the solution of the LBM equations for each lattice point in a serial manner can take from hours to days at these resolutions. In order to drastically decrease our time to solution, we leverage a parallel implementation that allows us to simulate the full cardiac cycle in minutes.

For the work in this challenge, we relied on the IBM Blue Gene/Q architecture. Similar to previous Blue Gene systems, it is built on a system-on-a-chip backbone and has expanded options for threading and memory access. The Blue Gene/Q system has a 64-bit PowerPC processor operating at 1.6 GHz frequency. Each node consists of 16 cores with 4 potential threads per core. There are capabilities for a 4-wide double precision FPU SIMD resulting in a 204.8 GFlop/s peak performance per node [12]. Memory per node is expanded to 16 gigabytes. In this work, we used 256 processors on 16 nodes of Blue Gene/Q for our simulations. The resulting velocity distribution is shown in Fig. 2. This is at 0.14 seconds in a 100 micron resolution simulation.

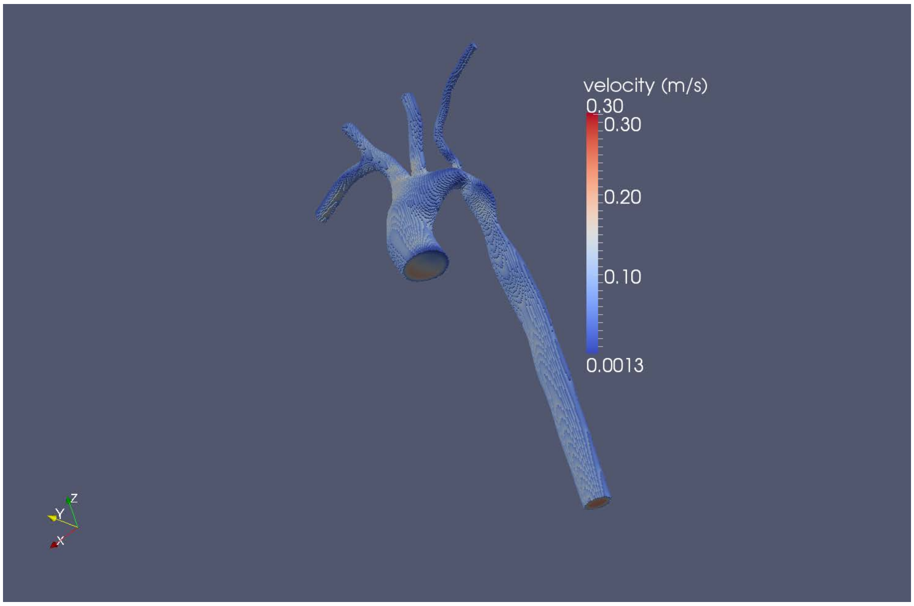

Fig. 2. Mapping showing the velocity distribution at .14 seconds in a 100 micron resolution simulation

4.3 Data Analysis and Visualization

The distribution function, f, at each lattice point is saved at a set time interval during the simulation allowing for post processing of the data to determine relevant macroscopic properties. It is during the post processing stage that the data is shifted from lattice units to SI units to allow for analysis of factor like fluid velocity, density, and pressure gradients. Only a subsample of time points are recorded and used for visualization and analysis. In this case, checkpoints were invoked every 20000 time steps. Paraview from Kitware is used to view the results [9].

5 Results

We assume rigid walls as well as the Newtonian behavior of the blood. The physical density is set at .001 g/mm^3 and the dynamic viscosity is .004 gr/mm/s.

We measure the mean pressure gradient between the upper and lower body by taking the difference of the average pressure of the fluid in the plane at the proximal and distal locations. The results of simulations at three different resolutions are provided in Table 1. The pressure proximal to the coarctation is measured at 113.1 mmHg (systolic) and 62.3 mmHg (diastolic) which corresponds well to the measured values of 115 mmHg and 65 mmHg respectively.

Table 1. Mean pressure gradient at different mesh resolutions

Resolution	Pressure Gradient at Diastole	Pressure Gradient at Systole
200μm	10.1 mmHg	12.2 mmHg
100μm	8.7 mmHg	10.9 mmHg
50μm	8.1 mmHg	10.4mmHg
20μm	8.2 mmHg	10.3 mmHg

Table 2. Required simulation results

Peak pressure difference between Plane 1 and Plane 2	10.6 mmHg
Mean pressure difference between Plane 1 and Plane 2	9.2 mmHg
Flow splits in supra-aortic and DAo	40% and 60%
Pressure in AAo(Systolic/Diastolic)	10.3mmHg/8.2mmHg

6 Conclusion

We have presented a system to simulate blood flow in a patient specific geometry in order to measure the pressure gradient in the aorta. The system imposes a regular grid on the vessel geometry derived from the segmentation of MRA data and uses HARVEY, a lattice Boltzmann application, to model the blood flow through the arteries and to derive the fluid pressure gradients.

We have tested our system on the provided datasets from the STACOM'12 CFD Challenge and analyzed the results. This has been a useful exercise to assist in validating our application. Our preliminary results demonstrate a 8.2 mmHg pressure differential at diastole and 10.3 mmHg at systole.

Acknowledgments. The authors would like to thank Dr. Hudong Chen and Dr. Raoyang Zhang of EXA Corporation for their helpful discussions. This work was supported by the Department of Energy's Computational Science Graduate Fellowship, grant number DOE CSGF DE-FG02-97ER25308. We would like to thank the LLNL Support Team for access to their Blue Gene/Q system.

References

1. Figueroa, A., Mansi, T., Sharma, P., Wilson, D.N.: CFD challenge: Simulation of hemodynamics in a patient-specific aortic coarctation model (2012), `http://www.vascularmodel.org/miccai2012/`
2. Rao, P.: Coarctation of the aorta. Current Cardiology Reports 7, 425–434 (2005), doi:10.1007/s11886-005-0060-0
3. Succi, S.: The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press (2001)
4. Chen, S., Doolean, G.D.: Lattice Boltzmann method for fluid flow. Annual Review Fluid Mechanics 30, 329–364 (1998)

5. Bhatnagar, P., Gross, E., Krook, M.: A model for collision processes in gases. Physics Review Letters 94, 511 (1954)
6. Zhang, R., Shan, X., Chen, H.: Efficient kinetic method for fluid simulation beyond the Navier-Stokes equation. Physics Review E 74, 1–7 (2006)
7. Peters, A., Melchionna, S., Kaxiras, E., Lätt, J., Sircar, J., Bernaschi, M., Bison, M., Succi, S.: 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 2010, pp. 1–10. IEEE Computer Society (2010)
8. Melchionna, S., Bernaschi, M., Succi, S., Kaxiras, E., Rybicki, F.J., Mitsouras, D., Coskun, A.U., Feldman, C.L.: Hydrokinetic approach to large-scale cardiovascular blood flow. Computer Physics Communications 181(3), 462–472 (2010)
9. Henderson, A.: Paraview guide, a parallel visualization application (2007)
10. Bærentzen, J.A., Anaæs, H.: Signed distance computation using the angle weighted pseudonormal. IEEE Transactions on Visualization and Computer Graphics 11(3), 243–253 (2005)
11. Peters Randles, A., Kale, V., Gropp, W., Kaxiras, E.: Performance analysis of the Lattice Boltzmann model beyond Navier-Stokes. Manuscript submitted for publication (2012)
12. Haring, R.A., Ohmacht, M., Fox, T.W., Gschwind, M.K., Satterfield, D.L., Sugavanam, K., Coteus, P.W., Heidelberger, P., Blumrich, M.A., Wisniewski, R.W., Gara, A., Chiu, G.L.T., Boyle, P.A., Chist, N.H., Kim, C.: The IBM Blue Gene/Q compute chip, vol. 32, pp. 48–60. IEEE Computer Society, Los Alamitos (2012)