Editors: Muhammad Sahimi, moe@iran.usc.edu
Barry I. Schneider, nsfphyman@gmail.com
and Gabriel Wainer, Gabriel.Wainer@sce.carleton.ca

TIGHTLY COUPLED ATOMISTIC-CONTINUUM SIMULATIONS OF BRAIN BLOOD FLOW ON PETAFLOP SUPERCOMPUTERS

By Leopold Grinberg, Joseph A. Insley, Dmitry A. Fedosov, Vitali Morozov, Michael E. Papka, and George Em Karniadakis

Several advances on the mathematical, computational, and visualization fronts have led to the first truly multiscale simulation and visualization of a realistic biological system.

Understanding physical phenomena at diverse scales has always been a great challenge. Experimental setups require sophisticated equipment to collect and process data at scales ranging from nanometers to kilometers in space, and from nanoseconds to days or even months in time. In the computational domain, designing mathematical models that can accurately predict physics in a multidimensional and parametrically uncertain environment isn't a trivial task. Furthermore, the extremely large number of degrees of freedom in the analyzed system requires the development of robust computational algorithms that are capable of both extracting the main features and of quantifying the possible errors in the predicted results. These algorithms also must scale to hundreds of thousands of computer processors, and might even promote the development of specialized hardware. Similar to experimental data, computational simulation results must be collected and processed with adequate accuracy. Parallel multiscale visualization tools are required to interactively explore scale interactions.

Molecular dynamics (MD) simulations allow the study of physical phenomena at extremely small temporal and spatial scales (below microseconds and micrometers) and are computationally expensive. Coarse-grained MD models, however, such as the dissipative particle dynamics (DPD) or peridynamics methods, alleviate this problem and permit the study of systems at much larger scales. Employing a continuum-based method further extends the range of problems that we can simulate.

Multiscale simulations based on the coupling of the aforementioned models can provide better insight into scale interactions. For example, multiscale modeling can help to understand crack formation and propagation, tornado formation and progression, and the process of platelet aggregation leading to blood clot formation. Interfacing an atomistic-based model with a continuum-based model has become necessary to simulate many of today's multiscale physical and biological system problems. Multiscale modeling requires the use of multiple mathematical models, each describing different scale regimes and corresponding codes. Properly coupling such heterogeneous descriptions and their implementations is currently one of the most difficult problems in computational physics and scientific computing. New fundamental advances in algorithms are needed to provide the proper mathematical interface conditions between micro- and macroscale systems. These algorithms must be capable of accurately capturing the problem's physics and of providing computational accuracy, stability, and efficiency. There's also a need for mechanisms to quickly extract and exchange data between solvers operating concurrently at various resolutions and on different computer processors. Finally, a need exists for quantitative data analysis tools that are suitable for operating on multiscale data and clearly presenting the scale interactions. Such tools are in the early stages of development.

Frameworks for coupled multiscale simulations are still quite immature, largely due to the mathematical and computational complexity required. Another limiting factor is the availability of resources with sufficient compute capabilities. Even with a 10-petaflop computer, capable of executing up to 10 quadrillion floating-point operations per second, microscale simulations are still limited to time intervals of 1 millisecond.

In short, advances are needed on multiple fronts to enable breakthroughs in multiscale modeling. These advances include the following:

- accurate mathematical models for each scale considered;

 Copublished by the IEEE CS and the AIP 1521-9615/12/$31.00 © 2012 IEEE

- robust parallel solvers for each model;
- stable interface conditions and parallel paradigms for coupling heterogeneous solvers;
- flexibility in using hardware and middleware for coupled multiscale simulations—for example, configurations suitable for use of accelerators by some of the coupled solvers, or even use of specialized computers for some components of coupled solvers; and
- developing the field of interactive multiscale visualization.

The latter is important for browsing through the data output representing billions of unknowns and for drawing physical conclusions based on scale interactions.

Here, we present a methodology for a tightly coupled, multiscale simulation of blood clot formation. Although the current study's scope is limited to a specific physiological phenomenon, the mathematical and computational methods we present are applicable to many other areas. The main objective in this study is to develop new computational and visualization methodologies that can be used in a broad range of applications.

Why Simulate Blood Clotting?

Blood clot (thrombus) formation is a multiscale process involving the interaction between platelets, blood cells about 2μm in size, and the proteins present in a damaged arterial wall endothelium. Blood clotting is a protective mechanism that helps seal an injured arterial wall; however, under certain pathological conditions blood clots can lead to vessel occlusion and block blood supply to vital organs.

Under other conditions, a clot could detach and travel with the flow, eventually blocking smaller arteries and causing a stroke. Simulations can help researchers understand the flow conditions that accelerate clot formation or lead to clot dissolution, and suggest ways to alter the local blood flow dynamics to prevent or aid such events.

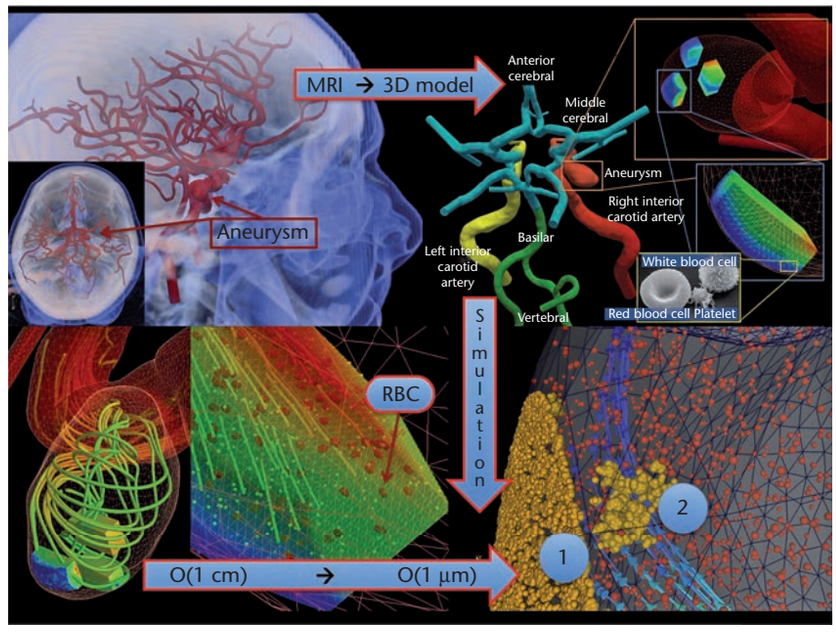

Figure 1. Multiscale simulation of the brain's blood flow. Vessel geometry is reconstructed from 3D magnetic resonance imaging (MRI), and small subdomains for solution at an atomistic level are inserted in the aneurysm. Plots at the bottom show the computed solution at the continuum and atomistic levels: streamlines indicate flow direction, yellow particles labeled "1" show clot formation at the wall, and particles labeled "2" represent a small cluster of platelets detached from the clot.

Challenges in Multiscale Blood Clot Simulation

Clot formation is a multistep process, starting from the slowing of platelet motion close to the arterial wall (platelet tethering). The platelet tethering follows platelet activation, leading to firm binding between the platelets and the wall, and binding between platelets. In later stages, red and white blood cells and fibrin fibers become part of the clot. The rate at which a clot forms also depends on local flow features, such as wall shear stress and its spatiotemporal variation, hence the need to simultaneously resolve the blood flow and the blood rheology.

Because of the large number of degrees of freedom required, it currently isn't feasible to have a multiscale simulation of clot formation and growth that accurately resolves blood flow and blood rheology using only a continuum or an atomistic approach. However, we found that coupling atomistic descriptions to resolve local microscale dynamics (such as protein interactions) and continuum descriptions to resolve large-scale flow dynamics helps substantially reduce the problem size. With this in mind, we created a parallel-computing paradigm that we developed and implemented in a coupled atomistic and continuum-based solver that now makes such simulations possible.

Multiscale Modeling: Mathematical Formulation

We performed a multiscale simulation of blood flow and blood rheology in a patient-specific model of arteries reconstructed from magnetic resonance imaging (MRI) data. The model considered in this study includes the major arteries of the neck and brain, and an aneurysm, as Figure 1 shows. The computational domain is

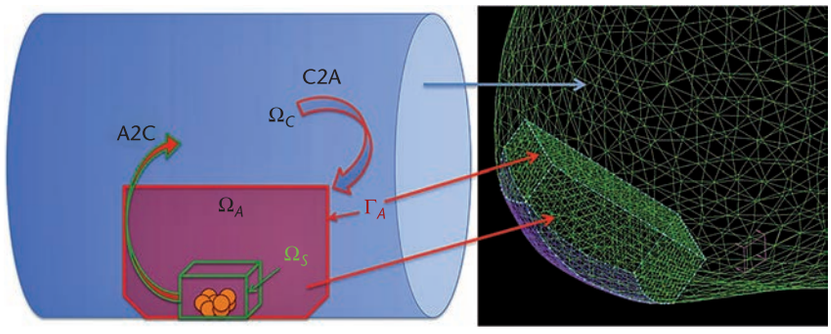

Figure 2. Coupling of atomistic and continuum descriptions: illustration of computational domains and interdomain data exchange. (a) An atomistic domain Ω_A is inserted into the continuum domain Ω_C in the region where platelet deposition (the yellow dots) is simulated. Interface velocity conditions are imposed at the boundaries Γ_A of Ω_A. Virtual boundary conditions for velocity are imposed inside Ω_C using an immersed boundary method. The reference velocity and immersed boundary geometry are sampled within the subdomain Ω_S placed inside Ω_A. These data are computed by the atomistic solver and projected onto the continuum field. (b) An atomistic domain of about 4 mm^3 is placed inside an aneurysm (Ω_C). (A2C stands for atomistic to continuum and C2A stands for continuum to atomistic.)

decomposed into a number of overlapping regions, which can employ different descriptions such as MD, DPD, or continuum. The solution in each region is integrated independently, while the continuity is established by proper interface conditions.

Figure 2 illustrates a setup for a coupled atomistic-continuum simulation of platelet aggregation. First, the continuum domain Ω_C is created. Second, the atomistic domain Ω_A is placed in the area of interest such that it completely overlaps with Ω_C. Third, an additional subdomain Ω_S for the sampling of atomistic data is inserted into Ω_A. The boundaries of Ω_A are discretized using 2D elements (triangles), while the volume of Ω_S is discretized using 3D elements (bins). The continuum and atomistic solvers run concurrently on non-overlapping groups of processors, and exchange data required for coupling the solutions in Ω_C and Ω_A via message passing. Next, we'll provide more details on the continuum and atomistic formulations integrated in our multiscale solver and also review the coupling of the two solvers.

Continuum-Based Modeling: Spectral Element Method

To model large-scale flow dynamics, we assume blood to be an incompressible Newtonian fluid with constant density and viscosity. The large-scale flow dynamics are modeled by Navier-Stokes equations: The flow problem is defined in a rigid domain Ω_C of the patient-specific arterial network. The domain Ω_C is bounded by arterial walls and is truncated at multiple inlets and outlets where flow and pressure boundary conditions are imposed.

The 3D Navier-Stokes equations are solved using the open-source parallel code *Nektar*, developed at Brown University. Nektar employs the spectral/*hp* element spatial discretization (SEM/*hp*; here *h* represents the mesh and *p* represents polynomial expansion),[1] which provides high spatial resolution and is well suited for solving unsteady flow problems in geometrically complex domains. The computational domain is decomposed into polymorphic elements. Within each element the solution is approximated by hierarchical, mixed-order, semi-orthogonal Jacobi polynomial expansions.[1] Figure 3 illustrates the domain decomposition and the polynomial basis used in Nektar. The SEM/*hp* discretization allows control over the accuracy of the solution by performing mesh refinement (*h*-refinement) and by varying the order of polynomial expansion (*p*-refinement). The *h*-refinement is particularly useful in regions of high geometric complexity, such as arterial junctions, and the region of a forming clot. The *p*-refinement helps to significantly reduce the number of degrees of freedom (grid points) in regions where the solution and the geometry are relatively smooth. For time integration, Nektar employs a high-order semi-implicit time-stepping scheme.[2]

To simulate moving objects or time-evolving structures (such as blood clots) within a fixed computational domain, we use the smooth profile method (SPM).[3,4] SPM belongs to the family of immersed boundary methods, and hence has no requirements on the mesh to conform to the boundaries of moving objects or structures, substantially simplifying computational procedures.

The patient-specific arterial networks considered in our study are very large; moreover, to accurately represent the wall shear stresses (an important characteristic of biological flows), a high spatial resolution is required. This leads to an extremely large computational problem on the order of one billion degrees of freedom. To efficiently solve such a large problem, we employed a multipatch domain decomposition method.[5] This method decomposes the full, tightly coupled problem into a number of smaller, tightly coupled problems (tasks) defined in subdomains (patches), where the global continuity (coupling) is enforced by providing proper interface conditions. The reconstructed domain

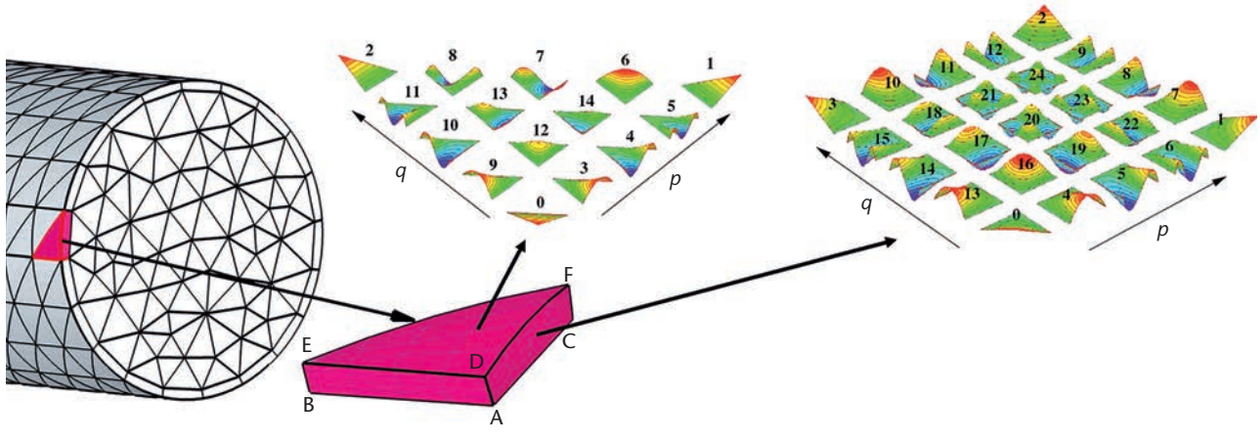

Figure 3. The leftmost graphical element is a schematic of the unstructured surface grid and the polynomial basis employed in Nektar. The computational domain is decomposed into non-overlapping elements (shown as a red prism). Within each element the solution is approximated by mixed-order, semi-orthogonal Jacobi polynomial expansions. The shape functions associated with the vertex, edge, and face modes for a fourth-order polynomial expansion are defined on triangular and quadrilateral elements (shown top center and top right).

of brain arteries presented in Figure 1 (top, right) has been subdivided into four overlapping patches shown in different colors. The multipatch method reduces the size of the problem solved with the semi-implicit method, while improving solver scalability and speed. Furthermore, this approach is well suited to the task-parallel framework that we employ in multiscale simulations.

Atomistic-Based Modeling: Dissipative Particle Dynamics Method

To model blood flow dynamics at the atomistic/mesoscopic scale, we employ a coarse-grained molecular dynamics approach[6] using the DPD method,[7,8] implemented in the code DPD-Large-scale Atomic/Molecular Massively Parallel Simulator (DPD-LAMMPS). DPD is a mesoscopic particle method, with each particle representing a molecular cluster rather than an individual molecule. The method can be seamlessly applied to simulate bonded structures (such as polymers or blood cells) and nonbonded particles (blood plasma). The DPD method is used to resolve physiological phenomena at scales coarser than the MD method, but it's sufficiently accurate for a highly detailed modeling of blood cell interactions. Further advantages of applying the DPD method include substantially larger time steps and considerably fewer degrees of freedom, hence the ability to analyze systems over significantly larger time intervals, such as minutes. The DPD system consists of N point particles interacting through pairwise conservative, dissipative, and random forces. The motion of DPD particles is governed by Newton's second law and a modified Verlet scheme is employed to advance the solution in time.

The atomistic problem is defined in a fixed nonperiodic domain Ω_A. The domain boundaries Γ_A are discretized into triangular elements T, where velocity and flux boundary conditions are imposed. The velocity vector computed by the continuum solver is interpolated on the center of each T. The particles are inserted through T with the rate and direction corresponding to the continuum data. Particles crossing T are deleted if the direction of the flow computed by the continuum solver points outside the atomistic domain.

Coupling Atomistic and Continuum Solvers

A framework for coupling atomistic and continuum formulations for the simulation of steady flows has been described in previous work.[9] The MD, DPD, and continuum formulations were coupled by imposing interface conditions (Dirichlet velocity conditions) at the boundaries of overlapping domains. The solvers were employed sequentially and the interface conditions were based on the velocity fields statistically averaged over long time intervals. The method was later reformulated for multiscale simulations of unsteady flow in complex geometries.[10]

Blood flow in a cranial arterial tree and particularly within an aneurysm is highly unsteady, which imposes obvious restrictions on the length of time over which the atomistic data can be averaged. One way of improving the statistical average of nonstationary fields is to replicate the computational domain where identically defined flow problems are solved. The variation of solutions obtained in each domain replica is due to the generation of random variables in the DPD equation with distinct seed values. In simulations using hundreds of thousands of processors, such statistical averaging techniques become prohibitively expensive. To overcome this difficulty, we employ the window proper orthogonal decomposition (WPOD)-based method.[10] This method computes the deterministic component of the nonstationary field by decomposing it into

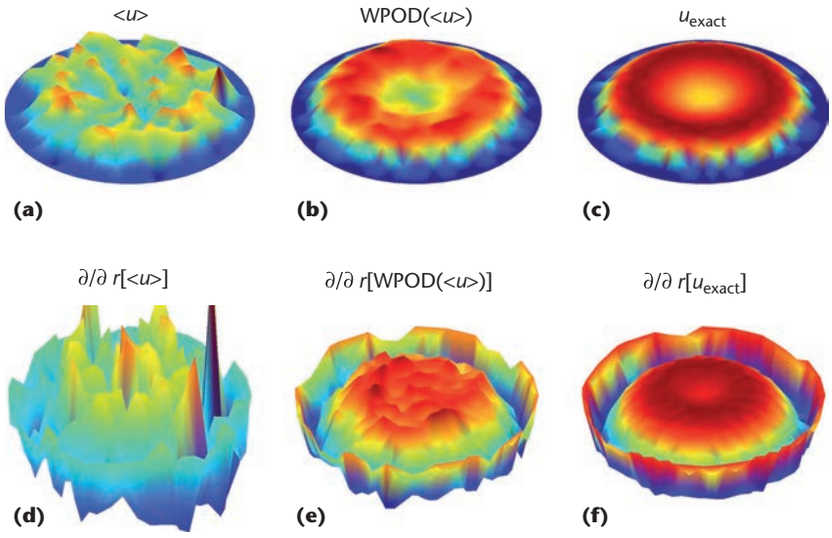

Figure 4. Processing data from a dissipative particle dynamics (DPD) simulation of unsteady flow in a pipe: plot (a) shows the averaged solution (streamwise velocity component) computed with a standard averaging over 50 time steps; plot (b) shows the averaged solution processed with the window proper orthogonal decomposition (WPOD) method, based on the correlation of a number of fields (snapshots), each averaged more than 50 time steps. Here, the data is reconstructed using the first two POD modes; plot (c) shows the exact solution; plots (d, e, and f) show the gradients of the velocity field's streamwise component.

orthogonal modes and reconstructing the velocity from modes characterized by a high correlation length. The modes corresponding to the short correlation length represent thermal fluctuations and are filtered out. Figure 4 demonstrates the effectiveness of using the WPOD method to process atomistic data in a simulation of a pulsatile pipe flow. The plots clearly show that WPOD substantially improves the solution's accuracy when reconstructing the deterministic components of an unsteady flow field, as well as its derivatives.

To set up a multiscale problem with heterogeneous descriptions, we must define length and time scales. In principle, the choice of spatiotemporal scales might be flexible, but it's limited by various factors such as method applicability (such as stability and flow regime) and problem constraints (for example, temporal resolution and microscale phenomena). To properly couple these different domains, we consistently nondimensionalize the time and length scales, and match nondimensional numbers characterizing the flow.

Platelet Aggregation Model

Each platelet is modeled by a single DPD particle with a larger effective radius[11] than that of the plasma particle, and is coupled to the plasma through the DPD dissipative interactions. The model of platelet aggregation is adopted from previous work,[12] where platelets can be in three different states: passive, triggered, and activated. In the passive state, platelets are non-adhesive and interact with each other through the repulsive DPD forces that provide their excluded volume interactions. Passive platelets might be triggered if they're in close vicinity to an activated platelet or to an injured wall. When a platelet is triggered, it still remains non-adhesive during the so-called activation delay time, which is chosen randomly from a specified time range. After the selected activation delay time, a triggered platelet becomes activated and adhesive. Activated platelets interact with other activated particles and adhesive sites, which are placed at the wall representing an injured wall section, through the Morse potential.

The Morse potential interactions are implemented between every activated platelet or adhesive site if they're within a defined potential cutoff radius r_d. The Morse interactions consist of a short-range repulsive force when $r < r_0$, and of a long-range attractive force for $r > r_0$ such that r_0 corresponds approximately to a platelet's effective radius. Finally, activated platelets can become passive again if they didn't interact with any activated platelet or adhesive site during a finite recovery time.

Coupled Multiscale Solver: Parallelization Strategies

From the viewpoint of parallel computing, coupled multiscale/multiphysics solvers can be characterized as a collection of interacting heterogeneous tasks. The heterogeneity is due to different mathematical formulations, solvers, programming models, and workloads associated with each task. The parallel implementation of our coupled solver is based on hierarchical task (functional) and data parallelism. The key idea of the multiscale solver design is to place heterogeneous tasks, such as solving an atomistic or a continuum problem, on non-overlapping groups of processors, while minimizing the data exchange rate between these tasks. Data parallel decomposition is performed then within each task independently. The data within each task is handled to support either distributed or shared memory or a hybrid parallel model. Additional fine-grain parallelism is achieved by employing hardware optimizations, such as use of Streaming SIMD Extension (SSE) or Quad Processing

Extension (QPX) instructions for CPUs. Use of accelerators such as GPUs and Intel's Many Integrated Cores (MIC) architecture is also an option.

To better load balance the coupled solver, the number of processors executing each task is set according to the computational load put on each task. Specifically, the solver's performance, such as time-to-solution and strong scaling within each task, dictate the required compute power. Although the solvers are tightly coupled—that is, they exchange a significant amount of data required for the interface conditions—this data exchange occurs once every 10 to 100 time steps and presents a negligible computational overhead.

The schematic representation of functional decomposition of the coupled solver is presented in Figure 5. At the top level, two tasks are identified—the solution of continuum and atomistic problems. The continuum solver consists of 1D and 3D solvers. The 1D solver provides a closure for pressure and flow-rate relations at the outlets of the truncated arterial domains and typically runs on 1 CPU. The 3D solver employs overlapping multipatch domain decomposition[5]—that is, solutions in each patch are computed concurrently on different nonoverlapping groups of processors. Solution continuity is achieved by interpatch conditions, requiring bidirectional data exchange between the overlapping patches. Dealing with interpatch conditions and data can also be seen as an additional subtask and handled by a small group of processors. Such additional task parallelism allows a simultaneous exchange of data between different interfaces, and it also lets us perform computations and even block communications within each subtask concurrently.

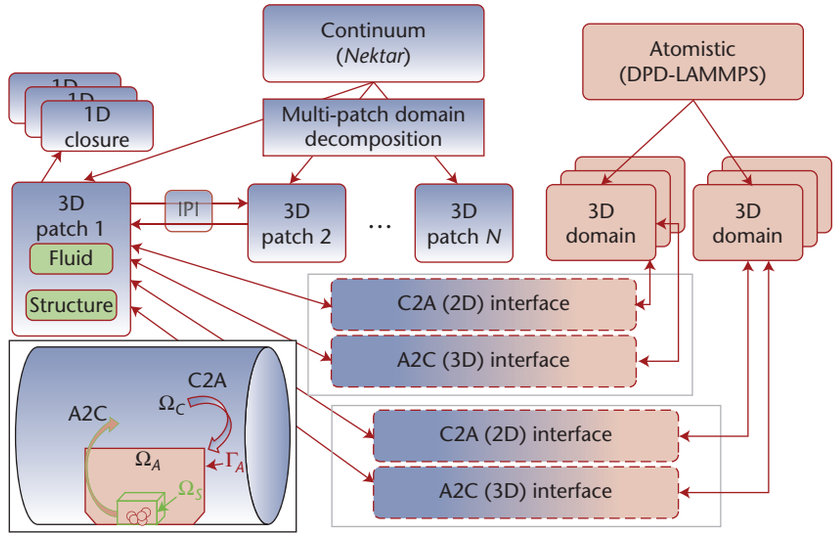

Figure 5. A schematic representation of task parallelism in the coupled solver. The continuum domain is partitioned into N overlapping 3D patches Ω_{C_i}, which exchange data via interpatch communicators. Each Ω_{C_i} has multiple inlets and outlets, each of which can be connected to a 0D or a 1D arterial network model to model boundary conditions (BCs) for inlets and outlets. Two atomistic domains Ω_{A_i} are placed inside Ω_{C_1}. Each Ω_{A_i} can be replicated several times to reduce statistical error. Each Ω_{A_i} is linked to a Ω_{C_1} via two interfaces: a 2D interface that considers the boundaries of Ω_{A_i} and uses data computed in Ω_{C_1} as BCs, and a 3D interface that's tailored to the immersed boundary method. The Ω_{C_1} uses an external force field computed in Ω_{A_i}. (IPI stands for inter-patch interface.)

The atomistic solver can be applied in one or more domains Ω_{A_i}, $i = 1, 2, \ldots$. Because there's only indirect coupling between the solution in each Ω_{A_i} through the solution in the continuum domain, two or more atomistic solvers perform embarrassingly parallel simulations. To obtain better statistics, each Ω_{A_i} can be further replicated, and each domain replica can be mapped to different groups of processors. Communication between the processors assigned to each replica is required to handle interface boundary conditions and compute statistics. Figure 5 shows how the atomistic solver creates two tasks to solve the problems defined in Ω_{A_1} and Ω_{A_2}; each Ω_{A_i} is replicated three times so that a total of six atomistic solvers are running in parallel. The solver limits neither the number of Ω_{A_i} nor the number of replicas. In a coupled atomistic-continuum simulation of platelet aggregation in an aneurysm (performed on about 300,000 processors on the IBM Blue Gene/P computer), three atomistic domains were placed inside the aneurysm (see Figure 1), and each Ω_{A_i} was replicated four times. The total number of DPD particles in the 12 replicas was about 10 billion. At the same time, the continuum domain was divided into four overlapping patches, and only one of the patches was interacting with the atomistic domains.

Synchronization between the atomistic solver and the continuum solver is based on imposing interface conditions. Data between related domains or groups of processors is exchanged directly—that is, no explicit data manager or master CPU is required. Direct data exchange removes unnecessary communication overhead associated with a data manager (which can be quite substantial on petaflop computers).

Table 1 presents the scaling of the coupled solver in simulations of platelet aggregation in patient-specific brain

Table 1. Scaling of the coupled solver in simulations of platelet aggregation in patient-specific brain vasculature with an aneurysm.

Blue Gene/P computer (4 cores/node)		
N_{core}	CPU time (in seconds)	Efficiency
32,768	3580.34	1.00
131,072	861.11	1.04
262,144	403.92	1.07
294,912	389.85	0.92
Cray XT5 computer (12 cores/node)		
N_{core}	CPU time (in seconds)	Efficiency
21,396	2194	1.00
30,036	1177	1.24
38,676	806	1.10
97,428	280	1.07
190,740	206	0.68

vasculature with an aneurysm. An excellent strong scaling is observed. The coupled Nektar-DPD-LAMMPS solver shows strong scaling in coupled blood flow simulation in the domain of Figure 1 (but with one atomistic domain). N_{core} is the number of cores and CPU time is the time required for 4,000 DPD-LAMMPS time steps. The total number of DPD particles is 823,079,981. Efficiency is computed as a gain in CPU-time divided by the expected gain due to an increase in N_{core} with respect to a simulation with lower core-count. Simulations were performed on the IBM Blue Gene/P computers at Argonne National Laboratory and the Jülich Supercomputing Centre in Forschungszentrum Jülich (FZJ), and on the Cray XT5 computers at Oak Ridge National Laboratory and the US National Institute for Computational Sciences.

With these results, it's apparent that our parallel paradigm makes it feasible to have full-scale simulations for arterial blood flow in the human body at the continuum level with local microscale resolution refinement in a few regions of interest for petascale supercomputers.

Multiscale Visualization

Scientific data visualizations are a highly effective exploratory and communication tool and can be offered as both evidence and proof in research. The visual representation of multiscale atomistic-continuum simulation data computed with billions of degrees of freedom is often the only manageable way to gain insight into a complex physical phenomena. The major challenges posed by multiscale atomistic-continuum visualizations are the following:

- how to represent the solution associated with point particles as a continuum;
- how to represent scale interactions by simultaneously presenting data from atomistic and continuum solvers; and
- how to rapidly process terabyte datasets stored in different formats.

To address these challenges, we developed several mathematical tools and computational methods. For example, we employed the WPOD to project the atomistic data associated with each particle to the continuum representation on a finite-element mesh. To work with the data and maintain spectral accuracy throughout the post-processing phase, we developed a custom parallel visualization tool based on Nektar and ParaView, an open source, multiplatform data analysis and visualization application.

Because we work with data structures and file formats that are optimized for simulation performance, the data needs to be converted to formats that are appropriate for visualization. This process can be costly in terms of I/O overhead and disk usage. To efficiently visualize the continuum data, we developed a parallel, coupled ParaView-Nektar code. This code uses ParaView's parallel visualization algorithms and image-rendering capabilities while leveraging Nektar's parallel data-processing capabilities. Key advantages of using this coupled code is that it reads the Nektar data in its native spectral element format, it performs all necessary data conversions on the fly, and it passes the transformed data on to the ParaView pipeline, thereby bypassing expensive I/O. Furthermore, we can use Nektar's utilities to process the data with high-order spectral accuracy, which means that interpolation, integration, and differentiation are performed on a Gaussian quadrature consistent with the simulation resolution. Thus, derived quantities such as vorticity are computed using high-order operators, resulting in a more accurate representation of the results. Computed data are interpolated on a new grid, with resolution controlled by the user. Nektar's capabilities can also be used to extract boundary geometry, which represents the arterial wall, and to calculate additional quantities, such as wall shear stress, taking into account the vessel curvature (see Figure 6).

There are two types of data computed by our atomistic solver: quantities associated with each particle, such as particle type, coordinate and velocity vectors, and activation level for particles representing

platelets; and an ensemble average velocity and density computed using the WPOD method.[10,13,14]

Typical DPD simulations of blood flow include particles representing the plasma, red blood cells (RBCs), platelets, glycocalyx, and so on, which might have different properties. Moreover, some blood cells (RBCs and glycocalyx) must be represented by a collection of particles with fixed connectivity. The number of particles in blood flow simulations can be extremely large, affecting not only the I/O complexity and size of the data files, but also the visualization. Fortunately, the majority of DPD particles in the simulations considered here represent the blood plasma, and their visualization can be substituted by presenting the flow field as the continuum, rather than as discrete particles (see Figure 7).

For example, the large-scale flow features can be presented using cutting planes (Figure 7b) and streamlines (Figure 7c), which show the direction the fluid will travel at a particular point in time, calculated from continuum data. Individual particles, or a subset of them, can be represented using glyphs. Glyphs are geometric primitives that are used to represent point data, where the object's location is dictated by the particle's coordinates, while other graphical attributes such as size, color, and orientation are dictated by other aspects of the data. For example, Figure 7d shows platelet particles colored by velocity. A substantially smaller number of particles can be used to illustrate the small-scale features, such as RBC membrane folding (Figure 7e), which can be presented by showing a surface representing each RBC membrane, constructed from a collection of particles.

The methods and software described here have been used to visualize multiscale data from coupled atomistic-continuum simulations of platelet deposition in an aneurysm. Having the capability to visualize data of dramatically different scales from within a single visualization environment is extremely important for understanding how events happening at one end of the spectrum are impacted by those occurring at the other. Here, the continuum data computed by Nektar reveal the complex large-scale flow features such as flow direction, recirculation regions, and swirling flow. These are represented as streamlines in Figure 8a. Visualization of the atomistic data computed with the DPD method—particularly that of activated platelets—illustrate the platelet deposition, along with the detachment of small platelet clusters due to high shear flow, which could not be detected without visualization. This is depicted by the yellow spheres in Figure 8a, which represent active platelet particles.

In addition to enabling scientists to explore and gain scientific insight from multiscale simulation data, visualization is also a useful tool for verifying the computed results. For example, ensemble average velocity fields computed with the WPOD method help to verify the correctness of the coupling between Nektar and the DPD-LAMMPS simulations. In Figure 8b, the data computed by Nektar and DPD-LAMMPS are compared by plotting the velocity vector fields extracted along the slice intersecting the continuum and atomistic domains. The solution inside the overlapping region depicted by the rectangle

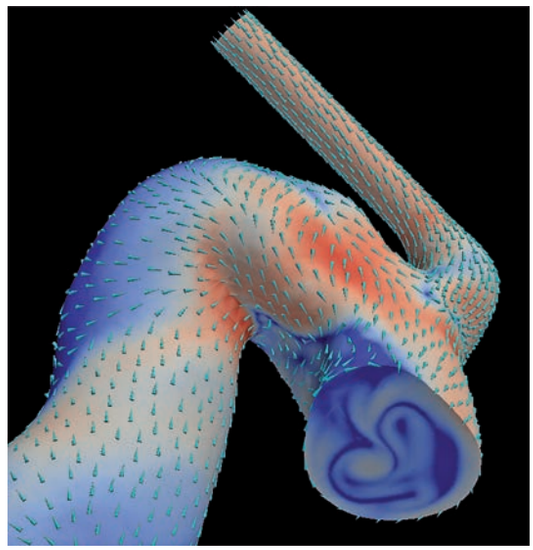

Figure 6. Visualization generated using the Nektar-ParaView tool. The small arrows on the surface of the artery indicate the direction of the wall shear stress, while the color of the artery indicates the pressure on the arterial wall.

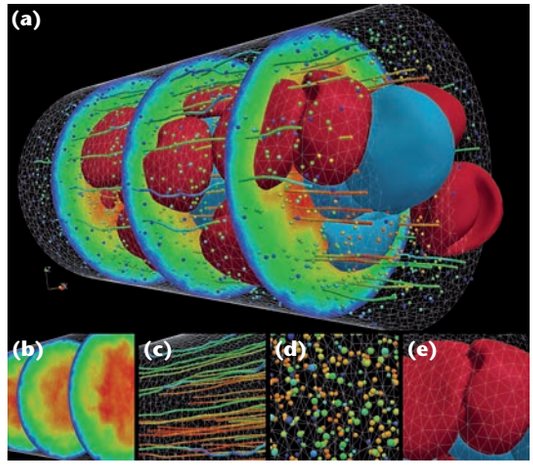

Figure 7. (a) Visualization of atomistic data, with representations from (b and c) continuum data, and from (d and e) individual or small collections of particles.

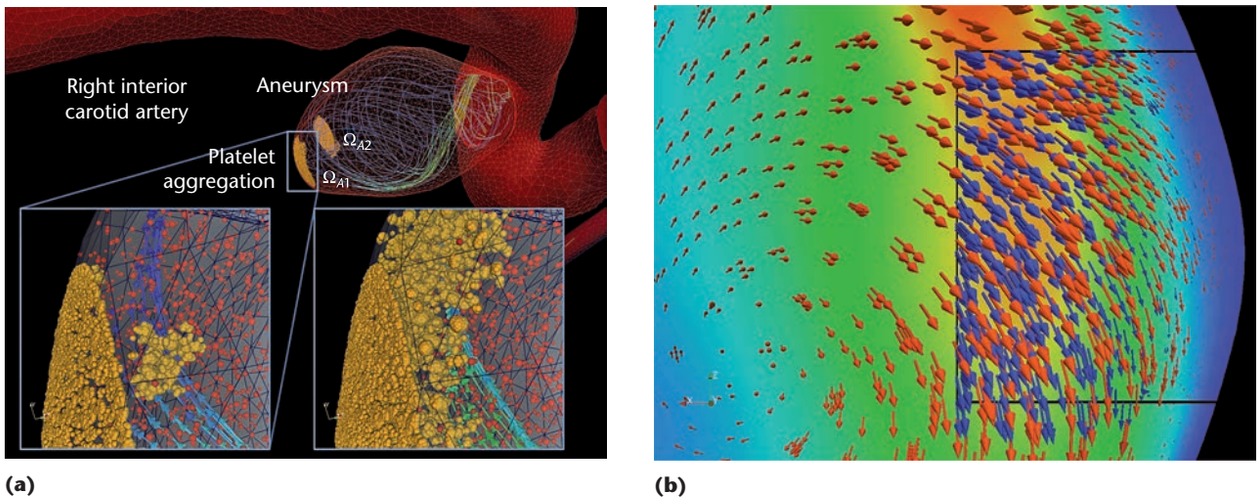

Figure 8. Visualizing multiscale blood flow simulation data. (a) A multiscale visualization of large-scale flow patterns, shown as streamlines, and microscale features such as platelet aggregation, represented by glyphs. Platelet aggregation is shown at initial (lower left) and more advanced (lower right) time steps. (b) A comparison of the flow fields calculated by the two different scale codes (Nektar for macroscale, DPD-LAMMPS for microscale). Arrows represent the velocity field computed by the continuum solver (red) and the atomistic solver (blue).

(outlined in black) was calculated by DPD-LAMMPS and by the continuum solver Nektar, while the outside region was calculated by Nektar only. This visualization shows that good correlation exists between the velocity fields computed by the two coupled solvers.

The use of multiscale and multiphysics modeling to examine the complex processes of materials, fluids, plasmas, and biological systems could lead to many discoveries in a broad range of research domains, including new alternative energy, therapeutic interventions in medicine, and climate science. At the same time, such modeling presents substantial challenges to computational scientists, mathematicians, and researchers alike.

Multiscale modeling requires the use of multiple codes and corresponding mathematical models that can describe different scale regimes. Properly coupling such heterogeneous descriptions and their implementations, however, is one of the most difficult problems in computational mathematics and scientific computing today. Multiscale visualization—an indispensable companion of multiscale solvers—similarly requires the development of new methodologies and computational techniques.

Here, we presented several advances on the mathematical, computational, and visualization fronts that enabled us to perform what we believe is the first truly multiscale simulation and visualization of a realistic biological system. Our approach is general and can be used in many other fields, thereby shifting the computational paradigm in large-scale simulation from one based on a monolithic single code to a more flexible approach where multiple heterogeneous codes are integrated seamlessly. This further opens up the possibilities for exploring multiscale phenomena and investigating long-range interactions in an effective way.

The availability of computational resources will continue to play an essential role in computational modeling. However, the potential of modern petaflop and future exaflop machines can be effectively utilized only through the use of fast and scalable algorithms.

Considering the ever-increasing number of cores in multicore architecture designs, the purely message passing interface (MPI)-based application is likely to suffer from lower memory per core availability. Specifically, the high memory demands in simulations with billions of degrees of freedom might not fit the distributed memory paradigm. Additionally, the scalability of codes that are based on the message passing model is also uncertain.

Hence, there's a pressing need to develop hybrid MPI-OpenMP applications. We've observed up to a threefold speedup in simulations running a hybrid MPI-OpenMP code as compared to pure MPI code running on the same number of compute nodes.

New computational paradigms for concurrent computing on multicore CPUs and acceleration devices such as GPUs and the coming Intel MIC also should be explored and integrated for multiscale simulations.

Acknowledgments

Computations were performed on the IBM Blue Gene/P computer of Argonne National Laboratory and the Jülich Supercomputing Centre, along with the Cray XT5 computer of the US National Institute for Computational Sciences and Oak Ridge National Laboratory. The visualization was performed on Eureka at the Argonne Leadership Computing Facility. Support was provided by the US

Department of Energy's Innovative and Novel Computational Impact on Theory and Experiment (INCITE) grant DE-AC02-06CH11357, as well as National Science Foundation grants OCI-0636336 and OCI-0904190.

References

1. G.E. Karniadakis and S.J. Sherwin, *Spectral/hp Element Methods for CFD*, 2nd ed., Oxford Univ, Press, 2005.
2. G.E. Karniadakis, M. Israeli, and S. Orszag, "High-Order Splitting Methods for the Incompressible Navier-Stokes Equations." *J. Computational Physics*, vol. 97, no. 1, 1991, pp. 414–443.
3. Y. Nakayama and R. Yamamoto, "Simulation Method to Resolve Hydrodynamic Interactions in Colloidal Dispersions," *Physical Rev. E*, vol. 71, no. 3, 2005, article ID 036707.
4. X. Luo, M.R. Maxey, and G.E. Karniadakis, "Smoothed Profile Method for Particulate Flows: Error Analysis and Simulations," *J. Computational Physics*, vol. 228, no. 5, pp. 1750–1769.
5. L. Grinberg and G. Karniadakis, "A New Domain Decomposition Method with Overlapping Patches for Ultrascale Simulations: Application to Biological Flows," *J. Computational Physics*, vol. 229, no. 15, 2010, pp. 5541–5563.
6. H. Lei, B. Caswell, and G.E. Karniadakis, "Direct Construction of Mesoscopic Models from Microscopic Simulations," *Physical Rev. E*, vol. 81, no. 2, 2010, article ID 026704.
7. P.J. Hoogerburgge and J.M.V.A. Koelman, "Simulating Microscopic Hydrodynamic Phenomena with Dissipative Particle Dynamics," *Europhysics Letters*, vol. 19, no. 3, 1992, pp. 155–160.
8. P. Espanol and P. Warren, "Statistical-Mechanics of Dissipative Particle Dynamics," *Europhysics Letters*, vol. 30, no. 4, 1995, pp. 191–196.
9. D.A. Fedosov and G.E. Karniadakis, "Triple-Decker: Interfacing Atomistic Mesoscopic-Continuum Flow Regimes," *J. Computational Physics*, vol. 228, no. 4, 2009, pp. 1157–1171.
10. L. Grinberg, "Proper Orthogonal Decomposition of Atomistic Flow Simulations," *J. Computational Physics*, vol. 231, no. 16, 2012, pp. 5542–5556.
11. W. Pan et al., "Hydrodynamic Interactions for Single Dissipative-Particle-Dynamics Particles and Their Clusters and Filaments, *Physical Rev. E*, vol. 78, no. 4, 2008, article ID 046706.
12. I.V. Pivkin, P.D. Richardson, and G.E. Karniadakis, "Blood Flow Velocity Effects and Role of Activation Delay Time on Growth and Form of Platelet Thrombi," *Proc. National Academy of Sciences USA*, vol. 103, no. 46, 2006, pp. 17164–17169.
13. J.A. Insley, L. Grinberg, and M.E. Papka, "Visualizing Multiscale Simulation Data: Brain Blood Flow," *Proc. IEEE Symp. Large-Scale Data Analysis and Visualization*, IEEE, 2011, pp. 3–7.
14. J.A. Insley, L. Grinberg, and M.E. Papka, "Visualization of Multiscale Simulation Data: Brain Blood Flow," *Proc. 2011 TeraGrid Conf.*, ACM, 2011, article no. 18.

Leopold Grinberg is a senior research associate in the Division of Applied Mathematics at Brown University. His research interests include spectral methods, parallel numerical algorithms, high-performance computing, and methods and software for multiscale simulations and visualization. Grinberg has a PhD in applied mathematics from Brown University. Contact him at leopold_grinberg@brown.edu.

Joseph A. Insley is a principal software development specialist at Argonne National Laboratory and at the University of Chicago. His research interests include the development of parallel and scalable methods for large-scale data analysis and visualization on current and next-generation systems. Insley has an MS in computer science from the University of Illinois at Chicago. Contact him at insley@anl.gov.

Dmitry A. Fedosov is a group leader in the Institute of Complex Systems at the Forschungszentrum Jülich, Germany. His research interests include multiscale modeling of soft matter and biological systems. Fedosov has a PhD in applied mathematics from Brown University. Contact him at d.fedosov@fz-juelich.de.

Vitali Morozov is a principal application performance engineer at the Argonne Leadership Computing Facility. His research interests include the simulation of plasma generation, material interactions, and thermal and optical properties; and applications related to laser- and discharge-produced plasmas. Morozov has a PhD in computer science from the A.P. Ershov Institute of Information Systems, Novosibirsk, Russia. Contact him at morozov@anl.gov.

Michael E. Papka is the director of the Argonne Leadership Computing Facility and the deputy associate laboratory director for computing, environment, and life sciences at the Argonne National Laboratory, and an associate professor of computer science at Northern Illinois University. His research interests include data visualization and analysis, and the development and deployment of high-performance computing infrastructures in support of scientific research. Papka has a PhD in computer science from the University of Chicago. Contact him at papka@anl.gov.

George Em Karniadakis is a professor of applied mathematics at Brown University, a research scientist at MIT, and the director of a new Department of Energy (DOE) Center on Mathematics for Mesoscopic Modeling of Materials (CM4). His research interests include spectral elements, parallel scientific computing, and microflows and nanoflows. Karniadakis has a PhD in mechanical engineering from MIT. Contact him at gk@dam.brown.edu.