A new computational paradigm in multiscale simulations with applications to brain blood flow

Leopold Grinberg
Division of Applied Math.
Brown University
Providence, RI 02912
lgrinb@dam.brown.edu

Vitali Morozov
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439
morozov@anl.gov

Dmitry Fedosov
Institute of Complex Systems
FZ Juelich
Juelich, 52425, Germany
d.fedosov@fz-juelich.de

Joseph Insley
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439
insley@anl.gov

Kalyan Kumaran
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439
kumaran@anl.gov

Michael Papka
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439
papka@anl.gov

George Em Karniadakis
Division of Applied Math.
Brown University
Providence, RI 02912
gk@dam.brown.edu

ABSTRACT
Interfacing atomistic-based with continuum-based simulation solvers is now required in many multiscale physical and biological systems. We present the computational advances that have enabled the first multiscale simulation on 131,072 processors by coupling a high-order (spectral element, SEM) Navier-Stokes solver with a stochastic (coarse-grained) Molecular Dynamics solver based on Dissipative Particle Dynamics (DPD). We study blood flow in a patient-specific cerebrovasculature with a brain aneurysm, and analyze the interaction of blood cells with the arterial walls that lead to thrombus formation and eventual aneurysm rupture. The macro-scale dynamics (more than 3 billion unknowns) are resolved by NEKTAR - a multi-level parallel SEM solver – while the micro-scale flow and cell dynamics within the aneurysm is resolved by an in-house version of DPD-LAMMPS (more than a billion molecules). The key contributions are: proper interface conditions for overlapped domains, topology-aware communication, SIMDization of all basic operations, and multiscale visualization.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous;
D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 2011 ACM X-XXXXX-XX-XX/XX ...$10.00.

Keywords
Coupled solvers, Continuum-atomistic simulations, Multiscale modeling, Cerebrovascular circulation

1. INTRODUCTION

Figure 1: (in-color) Human arterial tree: telescoping multi-scale modeling approach. Major arteries of the brain, reconstruction from MRI images. A flow in a small sub-region inside the aneurysm is computed using a coarse-grained molecular dynamics approach. Modeling interaction of the flow and the red and white blood cells and platelets ...

needed: on importance of multiscale simulations -> transition to coupled simulations -> transition
to blood flow

The cardiovascular system of the human body is the envy of every engineer. In just one minute, the average heart beats about 70 times pumping the entire blood supply of 5 liters through 62,000 miles of vessels, that is one-fourth the distance between the Moon and the Earth! The human brain, in particular, although less than 2% of the body weight, receives about 20% of the resting cardiac output of blood and 25% of body’s oxygen supply. Interactions of blood flow in the human brain occur between different scales, determined by flow features in the large arteries (diameter of 0.5 mm or larger), the smaller arteries and arterioles (500 µm to 10 µm), and the capillaries (mean diameter of 5 µm) all being coupled to cellular and sub-cellular biological processes. While many biological aspects have been studied systematically, surprisingly less effort has been put into studying blood flow patterns and oxygen transport within the brain, i.e., the fundamental biomechanical processes of the integrated intracranial vascular network. However, recent pioneering 3D imaging of the human brain by Cassot et al. in [3] and of the mouse brain by Mayerich et al. [4] provides statistical information for constructing realistic topological models on which future brain simulations will be based.

Following these results, we have been focusing on the development of an integrated model of the vascular network in the human brain (cerebrovasculature) characterized by three distinct spatial length scales: (1) The macrovascular network (MaN) consisting of large arteries, down to diameter of 0.5 mm, which are patient-specific and can be reconstructed from CT/MR imaging. Typically, about 1001 such arteries start from the circle of Willis, which is formed downstream of the four main arterial inlets at the neck (two carotids and two vertebral arteries); (2) The mesovascular network (MeN) consisting of small arteries and arterioles, from 500 µm down to 10 µm, which follow a tree-like structure governed by specific fractal laws. The human brain contains about (10 millions)² small arteries and arterioles. (3) The microvascular network (MiN) consisting of the capillary bed, which follows a net-like structure; its topological statistics have been recently quantified for the human brain in [3]. The typical number of capillary segments in the brain is more than 1 billion.

Previous works have shown that the resolution of each of them is extremely important and must be led with unprecedented accuracy to eliminate computational artifacts and reduced confidence in the simulation results [REFERENCE ?]. In this study, we report the results of the first of its kind simulation of the patient-specific brain blood flow. We have performed an image-based 3D Navier-Stokes simulations for fully resolving MaN, coupled to subpixel stochastic simulations of MeN and MiN. To this end, for MaN/MiN was simulated by NexTarg – a metasolver developed by our team for that purpose; and report on the performance of the solver. Results of multiscale patient-specific simulation performed on Argonne’s Blue Gene/P supercomputer are presented.

In this paper we describe a new computational paradigm in multiscale simulations enabling the aforementioned multiscale flow simulations. We also focus on the structure of NexTarg - a metasolver developed by our team for that purpose; and report on the performance of the solver. Results of multiscale patient specific simulation performed on Argonne’s Blue Gene/P supercomputer are presented.

2. BLOOD PHYSIOLOGY

Blood is a physiological fluid that consists of red blood cells (RBCs), white blood cells (WBCs), platelets, and plasma with various molecules. Blood transports oxygen and nutrients to cells of the body, removes waste products, and circulates a number of molecules and cells which mediate many vital processes in the organism such as immune response, tissue repair, etc. The volume fraction of RBCs is approximately 45%, of WBCs around 0.7%, and the rest is taken up by blood plasma and its substances. Due to a high volume fraction of RBCs, the rheological properties of blood are mainly determined by the RBC properties.

In vitro experiments [5, 15, 14] of blood flow in glass tubes with diameters ranging from 3 µm to 1000 µm have shown a dependence of the apparent blood viscosity on the tube diameter, RBC volume fraction, cell aggregability, and flow rate. Thus, in tubes with diameters larger than 400–500 µm blood can be assumed to be a nearly Newtonian fluid with a constant effective viscosity, while in smaller tubes it shows complex rheological behavior. This supports the application of continuum type methods within the MaN where a char-
actoristic vessel size is larger than 500 μm. However, accurate blood flow representation in MeN and in MiN requires explicit modeling of blood cells [12, 6] using mesoscale techniques (e.g., DPD). Moreover, continuum modeling of blood flow is not able to capture active processes in blood (e.g., RBC aggregation, WBC adhesion, blood clotting) which can be modeled using mesoscale methods.

Blood flow in aneurysm can be significantly reduced resulting in thrombus formation. Blood clots are formed mostly by platelets and fibrin, and may appear at sites of endothelial lining damages. Thrombus formation is a very complex process which involves a number of biochemical constituents, cells, and flow conditions. Recent modeling of platelet thrombli formation and growth in tube flow [13] assumed platelets to be spherical particles and introduced effective adhesive interactions between them. The model was able to capture essential dynamics of thrombus formation in flow and is adapted in this work to model blood clotting in the aneurysm coupled to the blood flow in the large portion of MaN network.

3. COMPUTATIONAL ALGORITHMS

Our multiscale solver \( \text{NekTarG} \) is based on coupling of several massively parallel codes. Each code uses different mathematical model and is effective for flow simulations at certain spatial and temporal scales. The original codes have been modified such that they can effectively exchange data required to impose boundary condition. The solvers share the default World communicator, while compute local solution in parallel over derived subcommunicators. The continuity in the overall solution is achieved by imposing proper interface conditions. The treatment of interface conditions is optimized to minimize the inter-solver communication, which leads to effective implementation of the coupled code on computers with heterogenous interconnect. The reuse of an existing software allows almost a non-intrusive approach in scaling-up the performance and the solver’s capabilities, as the overall scalability of the coupled code depends almost exclusively on the performance of its components.

The schematic representation of \( \text{NekTarG} \)'s structure is given in Figure 2. The three major components of \( \text{NekTarG} \) are:

1) \( \text{NekTar-1D} \) – a high-order spectral/hp element (SEM/hp) solver for unsteady three-dimensional (3D) flow problems [7].

Figure 2: Overall structure of the metasolver \( \text{NekTarG} \).

2) \( \text{NekTar-3D} \) – a high-order spectral/hp element solver for unsteady 1D flow problems [7].

3) \( \text{NekTar-LAMMPS} \) – a modified version of LAMMPS with major enhancements in DPD simulations for unsteady flows and complex geometries. DPD [9, 8] is a mesoscopic particle-based simulation technique, where each particle represents a cluster of atoms or molecules rather than an individual atom. DPD particles interact through pairwise soft forces and move according to the Newton’s second law of motion. The main challenge here is in imposing non-periodic boundary conditions (BCs) for unsteady flows in complex geometries. The boundary of a DPD domain is discretized (e.g., triangulated) into small enough elements where local BC velocities are set. In general, we impose effective boundary forces \( F_{eff} \) on the particles near boundaries that represent solid walls and inflow/outflow BCs. Such forces impose no-slip at solid walls and control flow velocities at inflow/outflow. The \( F_{eff} \) can be calculated during pre-processing, see [11] for details. In addition, at inflow/outflow we insert/delete particles according to local particle flux [11]. The algorithm allows to handle multiple particle species to enable simulations of various processes at mesoscale such as platelet aggregation. \( \text{NekTar-1D} \) and \( \text{NekTar-3D} \) are employed for solution of flow problems at the continuum but with different level of resolution; in the current study we focus on \( \text{NekTar-3D} \).

DPD-LAMMPS is used for meso- and micro-scale problems, such as simulations of RBCs and platelets dynamics. Each solver can be called by \( \text{NekTarG} \) multiple times; for example it is possible to couple through lightweight interfaces two or more 3D domains (patches), i.e., \( \text{NekTar-3D to NekTar-3D coupling} \) to \( \text{NekTar-3D coupling} \) at three artificial interfaces. Flow dynamics at the meso-vascular scale is computed by atomistic solver DPD-LAMMPS. Subdomain \( \Omega_j \) is embedded into one of the patches of \( \Omega_C \) as illustrated in Figure 1, this requires \( \text{NekTar-3D to NekTar-3D coupling} \) at three artificial interfaces. Flow dynamics at the meso-vascular scale is computed by atomistic solver DPD-LAMMPS.

In the following we describe the general approach for coupling parallel solvers. First, we describe the Multilevel Communicating Interface (MCI) designed for efficient coupling two or more parallel solvers. Second, we review the \( \text{NekTar-3D to DPD-LAMMPS coupling} \). Third, we describe the \( \text{NekTar-3D to DPD-LAMMPS coupling} \).

3.1 Multilevel Communicating Interface

Let us consider a flow problem defined in a computational domain \( \Omega \), which can be subdivided to overlapping or non-overlapping subdomains \( \Omega_j, j = 1, 2, \ldots \). The key feature of
the MCI architecture is the hierarchical decomposition of the default World communicator into subcommunicators. The communicator splitting is done accordingly to the decomposition of domain $\Omega$ into $\Omega_j$, and is performed in several stages as illustrated in Figure 3. At the first stage the topology-oriented splitting is performed, and the processors from different computers or racks are grouped into Level 2 ($L_2$) subcommunicators. In simulations on computer with homogeneous network the $L_2$ communicator is set to be the same as the default communicator. The $L_2$ groups are further subdivided according to the task-decomposition into Level 3 ($L_3$) non-overlapping groups. The communications required for solving tightly coupled problem by either Nektar-3D or DPD-LAPPMS are thus limited to a subset of processors communicating within $L_3$ group.

The $L_3$ sub-communicators used by each solver are further subdivided accordingly to low-level task-parallelism to form the Level 4 ($L_4$) sub-communicators. For example, calculations and communications required for interface conditions are performed on small subsets of processors ($L_4$ groups) mapped to partitions intersected by the interfaces. Data required by the interface conditions is communicated between the roots of the corresponding $L_4$ groups, as illustrated in Figure 4. Such communications are performed only few times at each time step and thus have negligible impact on the performance. The communications between the $L_4$ groups will be covered in more detail in Section 3.2.

### 3.2 Nektar-3D to Nektar-3D coupling

The scalability of effective solvers in large-scale parallel flow simulations based on semi-implicit time-stepping deteriorates as the number of processors increases. One of the big bottlenecks in scalability is solution of linear system, caused by effective preconditioners which often require high volume of communications and are typically not scalable on more than a thousand of processors. The multi-patch domain decomposition implemented in Nektar-3D addresses this issue as follows. A large monolithic domain is subdivided into a series of loosely coupled subdomains (patches) of a size for which good scalability of the parallel solver can be achieved. Once at every time step the data required by the interface conditions is transferred between the adjacent domains, and then the solution is computed in parallel in each patch. Such approach limits the processor count participating in the high volume of blocking and non-blocking communications, which alleviates the scalability limitations and leads to optimal allocation of computational resources. The mathematical formulation of the algorithm is described in [?]; here we focus on the parallel implementation of the algorithm applied to the domain of arteries presented in Figure 1.

Following this geometric domain decomposition of $\Omega$ into four patches $\Omega_j, j = 1, \ldots, 4$, four $L_3$ sub-communicators are created. The size of the $L_3$ sub-communicators is chosen a priori such that solution in each $\Omega_j$ can be obtained within approximately the same wall-clock time at each time-step, and in general it is related to the number of degrees of freedom in each $\Omega_j$. Each patch is subsequently partitioned into non-overlapping partitions in order to apply a parallel solver locally. The boundaries of $\Omega$ are the arterial walls, inlets and outlets. Decomposition of $\Omega$ into four overlapping $\Omega_j$ introduces six artificial interfaces (three inlets and three outlets), where inter-patch conditions must be imposed to enforce continuity in the velocity and pressure fields. To handle operations associated with boundary and interface conditions at each inlet and outlet, the $L_4$ sub-communicators are derived from the corresponding $L_3$. For example, in sub-domain of the right ICA (shown in red in Figure 1) four $L_4$ groups are created: one contains elements facing the inlet of $\Omega$, one contains elements facing the outlet of $\Omega$ and two with elements facing the two interfaces with the adjacent sub-domain. In Figure 4 we illustrate the process of data transfer between the two adjacent patches. The data exchange is a three-step process; at the first step data are gathered on the root of $L_4$; on the second step roots from corresponding two $L_4$ groups communicate over World; at the final step data is scattered from the root of $L_4$ to other ranks.
3.3 NekTarg-3D to DPD-LAMMPS coupling

To couple NekTarg-3D to DPD-LAMMPS we adopt the framework described in [7], where the continuum solver for Navier-Stokes (NS) was coupled to DPD and also to molecular dynamics (MD). The flow domain is decomposed into a number of overlapping regions, in which MD, DPD, or continuum solver can be used. Each subdomain is integrated independently. Coupling among overlapping subdomains is done through BC communications, which is done every \( \tau \) in time progression as shown in figure 5. The time \( \tau \) may correspond to a different number of time steps for distinct multiscale descriptions.

![Figure 5: A schematic of the time progression in different subdomains.](image)

To setup a multiscale problem with multiple descriptions we are required to define length and time scales. In principle, the choice of spatiotemporal scales may be flexible, but it is limited by various factors such as method applicability (e.g., stability, flow regime) and problem constraints (e.g., temporal resolution, mesoscale/microscale phenomena). For example, a unit of length \( (L_{NS}) \) in NS domain corresponds to \( 1 \) mm, while a unit of length \( (L_{DPD}) \) in DPD is equal to \( 5 \) \( \mu \)m in order to adequately capture platelet aggregation phenomena. In addition, fluid properties (e.g., viscosity) in different descriptions may not necessarily be the same in various method's units. To glue different descriptions together we need to match consistently non-dimensional number/s, which are characteristic for a certain flow, as an example Reynolds and Womersley numbers in our blood flow problem. The following formula provides velocity scaling between NS and DPD subdomains and implies consistency of Reynolds number

\[
\frac{v_{DPD}}{v_{NS}} = \frac{L_{NS}}{L_{DPD}} \frac{v_{DPD}}{v_{NS}}.
\]  

where \( v_{NS} \) and \( v_{DPD} \) are the kinematic fluid viscosities the NS and DPD regions. The time scale in each subdomain is defined as \( t \sim L^2/\nu \) and is governed by the choice of fluid viscosity. In our simulations we selected that single time step in NekTarg-3D solver (\( \Delta t_{NS} \)) corresponds to 20 time steps in DPD (\( \Delta t_{DPD} \)). The data exchange between the two solvers occurs every \( \tau = 10\Delta t_{NS} = 200\Delta t_{DPD} \sim 0.6344 \) s.

The methodology developed in [7] has been applied to steady flow problems in simple geometries, while here we consider an unsteady flow in the domain with complex geometry. In the coupled continuum-atomistic simulation we employ the domain of Circle of Willis (CoW) \( \Omega_C \) with an insertion of an additional domain \( \Omega_A \) inside the aneurysm as depicted in Figure 1. The methodology proposed in the current study allows placement of several overlapping or non-overlapping atomistic domains coupled to one or several continuum domains to simulate the local flow dynamics at meso- and micro-vascular scales.

Let \( \Gamma_I \) denote the boundaries of \( \Omega_A \), where interface conditions are imposed. The \( \Gamma_I \) are discretized by triangular elements \( T \) as presented in Figure 1. To couple the atomic and continuum domains, the following steps are performed at the preprocessing:

1) Processors assigned to \( \Omega_A \) and mapped to partitions intersecting the \( \Gamma_I \) are forming an LA subcommunicator.

2) The coordinates of \( T \) mid-points are sent from the root of \( L3 \) of \( \Omega_A \) to the \( L3 \) roots of each \( \Omega_{Ci} \), where continuum solver is applied.

3) The \( L3 \) roots of continuum domains not overlapping with \( \Gamma_I \) report back to the \( L3 \) root of \( \Omega_A \) that coordinates of \( T \in \Gamma_I \) are not within the boundaries of those domains. If coordinates of \( T \) are included in a particular \( \Omega_{Ci} \), then a new LA group is derived from \( L3 \) of this \( \Omega_{Ci} \). This LA group consists of the processes mapped to partitions of \( \Omega_{Ci} \) including the \( T \) coordinates. \( L3 \) root of \( \Omega_{Ci} \) signals to the \( L3 \) root of \( \Omega_A \) that communication between the LA groups of \( \Omega_{Ci} \) and \( \Omega_A \) should be established in order to allow data transfer between \( \Omega_{Ci} \) and \( \Omega_A \). From this point the communication between \( \Omega_A \) and relevant \( \Omega_{Ci} \) is performed between the \( LA \) roots from both sides.

During the time-stepping scheme the velocity field computed by the continuum solver is interpolated onto the predefined coordinates and are transferred to the atomistic solver.

The inherited feature of the DPD-LAMMPS solver is that it is capable to replicate the computational domain and solve an array of problems defined in the same domain but with different random forcing. Averaging solutions obtained at each domain replica improves the accuracy of the statistical analysis. In order to preserve this capability of DPD-LAMMPS to concurrently obtain several realizations without introducing additional complexity into \( \Omega_A \) – \( \Omega_C \) data exchange we need to design a computational algorithm that will seamlessly collect or distribute data required for the interface conditions over all replicas of \( \Omega_A \) and transfer it via one p2p communication to a process from \( \Omega_C \). To accommodate this requirement the communicating interface is constructed as follows. Let us consider \( N_A \) replicas, consequently the \( L3 \) group associated with \( \Omega_A \) is further subdivided into \( N_A \) non-overlapping groups \( L3_{ji} \), \( j = 1, ..., N_A \) as illustrated in Figure 6. Each replica is partitioned in order to apply a parallel DPD-LAMMPS solver locally, and \( LA \) groups are derived from each \( L3_{ji} \). The \( LA \) group of \( L3_{ji} \) is then considered as a master and \( LA \) groups of \( L3_{j}, j > 2 \) are the slaves. The master \( LA \) communicates with the process of corresponding \( \Omega_C \) and broadcast or gather data from or to the slaves, as illustrated in Figure 6.
3.4 Processing non-stationary atomic data

Ensemble average solution \( \bar{u}(t, x) \) and thermal fluctuations \( u(t, x) \) of the velocity field are two very important characteristic in analysis the atomistic simulation. However, computing these two parameters in non-stationary process is extremely difficult.

In stationary flow simulations the average solution \( \bar{u}(x) \) is typically computed by sampling and averaging the trajectories of the particles over a subdomain (bin) \( \Omega_p \) and over a very large time interval. In non-stationary flow simulations, an ensemble average \( \bar{u}(t, x) \) is required, but it is not obvious how to define a time interval \( T \gg \Delta t \) over which the solution can be averaged. It is possible to perform phase averaging, if the flow exhibits a limit cycle and integrate the solution over a large number of cycles. Constructing the ensemble based on number of realizations \( N_t \) improves the accuracy by a factor of \( \sqrt{N_t} \). In our simulation we utilized about 130K compute cores by the atomistic solver, doubling the number of realisations would require more computational resources than available, while resulting in only \( \sqrt{2} \) accuracy improvement.

To this end, we have developed a window proper orthogonal decomposition (WPOD) of general atomistic simulations that leads to a significant reduction of the computational load and enhances the quality of the numerical solutions. WPOD applied to the atomistic data computed in stochastic simulations helps to extract information on collective and correlated in time and space motion of particles. We will employ the method of snapshots \([7]\) and extend it to analyze a certain space-time window adaptively. A similar extension but for continuum based systems was presented in \([7]\), where WPOD was employed for the analysis of intermittent laminar-turbulent flows.

In Figure 7 we present the results of DPD simulations of healthy and diseased RBCs. The WPOD was applied as a co-processing tool performing spectral analysis of the velocity field to compute \( \bar{u}(t, x) \) and \( u(t, x) \).

In the following we briefly review the WPOD methodology. The WPOD is a spectral analysis tool based on transformation of the velocity field into orthogonal temporal and spatial modes: \( \bar{u}(t, x) = \sum_{k=1}^{N_{pod}} \alpha_k(t) \phi_k(x) \). The temporal modes are computed as the eigenvectors of correlation matrix constructed from the inner product of velocity fields (snapshots) computed at different times. The velocity field snapshots are computed by sampling (averaging) data over short time-intervals, typically \( N_{ts} = [50 500] \) time-steps. The data is sampled over spatial bins of a size comparable to the cut off radius \( r_c \). To compute the \( \bar{u}(t, x) \) we analyse the eigenspectrum of the correlation matrix, specifically the convergence rate of its eigenvalues \( \lambda_k \). The high modes represent small-scale features with very short correlation time, i.e., the thermal fluctuations, and convergence rate of the high modes is very slow. In contrast, the \( \lambda_k \) for the low modes converge very fast. We separate the POD eigenspectrum based on the convergence rate of the modes. We can then compute the ensemble average from the low most energetic modes while the fluctuations \( u(t, x) \) from the high slowly decaying modes. The number of POD modes \( k \) required to compute \( \bar{u}(t, x) = \sum_{k=1}^{k} \alpha_k(t) \phi_k(x) \) is determined adaptively by analysing the eigenspectrum.

In Figure 8 we plot the POD eigenspectrum for two velocity components in unsteady flow simulation. The difference in the convergence rate of the low-order (correlated) modes and high-order (uncorrelated) modes is clear. Computed with WPOD \( \bar{u}(t, x) \) was about one order of magnitude more accurate than computed with standard averaging procedure. Comparable accuracy was achieved by performing 25 concurrent realizations, which demanded 25 times more computational resources. The smoother velocity field reconstructed with the WPOD allows better accuracy in predicting the mean wall shear stress, which is a very important quantity in biological flows.

3.5 Performance tuning and optimization

The exploit advantages of specific architectural differences of modern supercomputers, we have carried out performance tuning of our computational kernels as well as communication strategy to maximize the messaging rate. Specifically, our single core performance tuning is based on the fact that
most microprocessors provide additional floating point hardware unit, which maximizes the floating point performance rate by executing SIMD instructions. In such, based on AMD Opteron microprocessors, the Cray XT5 supercomputer can execute SIMD SSE instructions. Similarly, based on PowerPC 450 microprocessor with special Double Humer floating point unit, the Blue Gene/P supercomputer can execute double FPU instructions. Both types of microprocessors set certain restrictions on the usage of SIMD instructions (see details in References [2, 10]), which we address as follows. Specifically, we focus on the following optimization tasks: a) proper data alignment is enforced by the use of posixa_memalign call to guarantee the memory allocation with 16 bytes alignment for the most important data structures; and b) a number of the kernel routines with high flop rate depends on the participating vectors in use as long as possible.

Blue Gene/P compute nodes are connected with three networks that the application may use: a 3D torus network that provided point-to-point (p2p) messaging capability, a collective network which implements global broadcast-type operations, and a global interrupt network for fast barrier synchronizations.

On the 3D torus, packets are routed on an individual basis with either deterministic or adaptive routing. With deterministic routing, all packets between a pair of nodes follow the same path along X,Y,Z dimensions in that order. With adaptive routing, each packet can choose a different path based on the load on the torus router ports. The BG/P architecture also has a Direct Memory Access (DMA) engine to facilitate injecting and receiving packets to/from the network. To maximize the messaging rate, all 6 links of the torus can be used simultaneously.

The Blue Gene/P specific implementation of the message exchange library is implemented in $NekTor$ based on the information provided by the personality structure, such as the torus coordinates (X,Y,Z) of the node, the CPU id number $T$ within the node, and the coordinates of the p2p targets. In particularly communication intensive routines, such as a parallel block-sparse matrix-vector multiplication, we create a list of communicating pairs and schedule the communications so that at each time, the node have at least 6 outstanding messages targeted all directions of the torus simultaneously. The incoming messages are processed on the “first come, first served” basis. For partitioning of the computational domain into non-overlapping partitions we employ the METIS_PartGraphRecursive routine of the METIS library [10]. In unstructured meshes a relatively high number ($O(10) - O(100)$) of adjacent elements sharing vertex, edge and face may exist, hence the large volume of p2p communications. To minimize the communication between partitions we provide to METIS the full adjacency list including elements sharing only one vertex. The weights associated with the links are scaled with respect to the number of shared degrees of freedom per link. According to our measurements, the topology-aware p2p communication algorithm reduces the overall run time for the application by about 3 to 5% while using 1024 to 4096 compute cores of Blue Gene/P. In Table 2 we compare the computational time required in simulations of a turbulent flow in a carotid artery where: a) partitioning considers only elements sharing the face degrees of freedom and b) all neighbor elements are taken into account.

<table>
<thead>
<tr>
<th>N cores</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1181.06</td>
<td>654.94</td>
<td>381.53</td>
<td>238.05</td>
</tr>
<tr>
<td>b</td>
<td>1171.82</td>
<td>638.00</td>
<td>361.65</td>
<td>219.87</td>
</tr>
</tbody>
</table>

Table 2: BG/P Simulations with two partitioning strategies: CPU-time (seconds) required for 1000 time-steps.

### 4. RESULTS

In the following we present the performance of $NekTor$ in unsteady 3D flow simulations. The computations have been carried on IBM BlueGene/P [?] (BG/P), CRAY-XT5 [?] and Sun Constellation Linux Cluster [?] computers. The superior parallel performance in simulations with $NekTor$-3D and multipatch decomposition compared to a single patch
is presented in [?]. The accuracy of the method is also discussed in the same publication. Here we present scaling obtained in solving macro-scale flow problems with very large number of degrees of freedom.

4.1 Performance

First, we consider a 3D flow problem in a very large computational domain subdivided into multiple patches $\Omega_i$, $i = 1, \ldots, N_p$. The solution is computed by coupling multiple instances of $\texttt{NekTAR-3D}$ as was described in Section 3.2. Each $\Omega_i$ is composed of 17,474 tetrahedral elements, while the one element-wide overlapping regions contain 1,114 tetrahedral elements. In each spectral element the solution was approximated with polynomial expansion of order $P = 10$. The parallel efficiency of the method is evaluated by solving an unsteady flow problem. The weak scaling obtained on BlueGene/P (4 cores/node) and Cray XT5 (8 cores/node) is presented in Table 3 and the strong scaling obtained on BG/P is presented in Table 4. On both computers, good and similar scaling is observed. The performance of $\texttt{NekTAR}$ was also measured on up-to 122,880 cores of BG/P using the same computational domain as in the previous example with spatial resolution $P = 6$. In the weak scaling tests performed on 49,152 and 122,880 cores (16 and 40 patches, respectively; 3072 cores per patch) the code achieved 92.3% parallel efficiency. The performance of $\texttt{NekTAR}$ was also benchmarked in simulations with 40 patches on 96,000 cores of the Cray XT5 with 12 cores per node. We used polynomial order $P = 12$, and the number of degrees of freedom was about 8.21 billions. The wall clock time measurements were exactly as expected: approximately 610 seconds per 1000 time steps.

<table>
<thead>
<tr>
<th>$N_p$ (DOF)</th>
<th>total # of cores</th>
<th>CPU-time [s] /1000 steps</th>
<th>weak scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlueGene/P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 (0.384B)</td>
<td>6,144</td>
<td>650.67</td>
<td>reference</td>
</tr>
<tr>
<td>8 (1.038B)</td>
<td>16,384</td>
<td>665.23</td>
<td>95%</td>
</tr>
<tr>
<td>16 (2.085B)</td>
<td>32,768</td>
<td>703.4</td>
<td>92%</td>
</tr>
<tr>
<td>Cray XT5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 (0.384B)</td>
<td>6,144</td>
<td>462.3</td>
<td>reference</td>
</tr>
<tr>
<td>8 (1.038B)</td>
<td>16,384</td>
<td>477.2</td>
<td>96.9%</td>
</tr>
<tr>
<td>16 (2.085B)</td>
<td>32,768</td>
<td>505.1</td>
<td>91.5%</td>
</tr>
</tbody>
</table>

Table 3: BlueGene/P (4 cores/node) and Cray XT5 (8 cores/node): weak scaling in flow simulation with $N_p = 3, 8$ and 16 patches.

<table>
<thead>
<tr>
<th>$N_p$ (DOF)</th>
<th>total # of cores</th>
<th>CPU-time [s] /1000 steps</th>
<th>strong scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (0.384B)</td>
<td>3,072</td>
<td>996.98</td>
<td>reference</td>
</tr>
<tr>
<td>3 (0.384B)</td>
<td>6,144</td>
<td>650.67</td>
<td>(76.6%)</td>
</tr>
<tr>
<td>8 (1.038B)</td>
<td>8,192</td>
<td>1025.33</td>
<td>reference</td>
</tr>
<tr>
<td>8 (1.038B)</td>
<td>16,384</td>
<td>685.23</td>
<td>(74.8%)</td>
</tr>
<tr>
<td>16 (2.085B)</td>
<td>16,384</td>
<td>1048.75</td>
<td>reference</td>
</tr>
<tr>
<td>16 (2.085B)</td>
<td>32,768</td>
<td>703.4</td>
<td>74.5%</td>
</tr>
</tbody>
</table>

Table 4: BlueGene/P (4 cores/node): strong scaling in flow simulation with $N_p = 3, 8$ and 16 patches.

In Table 5 we present the strong scaling of $\texttt{NekTAR}$ in coupled simulations of platelets aggregation.

<table>
<thead>
<tr>
<th>$N_{core}$</th>
<th>CPU-time [s]</th>
<th>efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlueGene/P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28,672</td>
<td>3205.58</td>
<td></td>
</tr>
<tr>
<td>61,440</td>
<td>1399.12</td>
<td>107%</td>
</tr>
<tr>
<td>120,976</td>
<td>665.79</td>
<td>102%</td>
</tr>
<tr>
<td>Cray XT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17,280</td>
<td>2193.66</td>
<td></td>
</tr>
<tr>
<td>34,560</td>
<td>762.99</td>
<td>144%</td>
</tr>
<tr>
<td>93,312</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: BlueGene/P (4 cores/node) and Cray XT5 (12 cores/node): strong scaling of $\texttt{NekTAR}$ in coupled flow simulation in the domain of Figure 1. $N_{core}$ - number of cores assigned to the DPD-LAMMPS solver. Number of cores assigned to $\texttt{NekTAR}$-3D was fixed: 4,096 on BG/P and 4116 on CRAY XT5. CPU-time - time required for 4000 DPD-LAMMPS time-steps (200 $\texttt{NekTAR}$'s time-steps). Total number of DPD particles: 823,079,981.

4.2 Coupled continuum-atomistic simulations

In the coupled continuum-atomistic simulation we employ the domain of CoW with an insersion of an additional sub-domain $\Omega_4$ inside the aneurysm as depicted in Figure 1. The blood clot formation typically starts close to the bottom of an aneurysm, hence the location of $\Omega_4$. We employ physiological flow characteristics: Reynolds number $Re = 394$ and Womersley number $W = 3.7$. We also employ patient specific flow boundary conditions at the four inlets and RC boundary conditions at all outlets of $\Omega_C$. The volume of $\Omega_4$ is $3.93 \text{mm}^3$, it interfaces the continuum domain at five planar surfaces $\Gamma_k, k = 1, \ldots, 5$ and its sixth surface $\Gamma_{wall}$ overlaps with the aneurysm’s wall. The size and velocities imposed at $\Gamma_{wall}$ have been scaled up to ... the main parameters in the DPD simulations are: Fedosov.

In Figure 9 we show the continuity of the velocity field at continuum-continuum and continuum-atomistic interfaces. In Figure 10 we present the clot formation process.

5. CONCLUSIONS

6. ACKNOWLEDGMENTS

This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

7. ADDITIONAL AUTHORS

8. REFERENCES

[4] M. D., A. L., and M. B. Knife-edge scanning microscopy for imaging and reconstruction of...


