Comput Mech (2010) 46:71–82
DOI 10.1007/s00466-009-0450-z

ORIGINAL PAPER

# Cardiovascular flow simulation at extreme scale

**Min Zhou · Onkar Sahni · H. Jin Kim · C. Alberto Figueroa · Charles A. Taylor · Mark S. Shephard · Kenneth E. Jansen**

Received: 25 September 2009 / Accepted: 11 November 2009 / Published online: 18 December 2009


**Abstract** As cardiovascular models grow more sophisticated in terms of the geometry considered, and more physiologically realistic boundary conditions are applied, and fluid flow is coupled to structural models, the computational complexity grows. Massively parallel adaptivity and flow solvers with extreme scalability enable cardiovascular simulations to reach an extreme scale while keeping the time-to-solution reasonable. In this paper, we discuss our efforts in this area and provide two demonstrations: one on an extremely large and complex geometry (including many of the major arteries below the neck) where the solution is efficiently captured with anisotropic adaptivity and another case (severe abdominal aorta aneurysm) where the transitional flow leads to extremely large meshes ($O(10^9)$) and scalability to extremely large core counts ($O(10^5)$) for both rigid and deforming wall simulations.

## 1 Introduction

Computational simulations of blood flow have been used to study the cardiovascular system in a variety of applications [31] including the study of the hemodynamics of healthy and diseased blood vessels [5,18,29], the design and evaluation of medical devices [1,13,16,28], and prediction of the outcomes of surgeries [17,26,32]. As computing power and numerical methods advance, simulation-based approaches are expected to become even more extensively used in studying the cardiovascular system. In such studies, there are several challenging aspects that must be considered.

First, we need to represent complex geometries of the cardiovascular system accurately. The aorta starts from the left ventricle and it tapers, bends, and bifurcates into small vessels. The geometric changes directly affect hemodynamic factors such as blood velocity, pressure, and wall shear stress. Further, diseased blood vessels can exhibit highly non-uniform, complex structures, i.e. abdominal aortic aneurysms with high tortuosity, aortic coarctation, thrombus in aortic anerysms, and so forth. Thus, to compute realistic hemodynamic factors in the cardiovascular system, the geometry should be represented accurately. Two cardiovascular applications with complicated geometric models are considered in this study. The first one is a patient-specific "whole" body model composed of 78 arteries extending across most of the body from the neck to the feet. The other application is a patient-specified abdominal aorta aneurysm model.

The reliability and accuracy of simulations is also a strong function of the mesh quality and configuration. The application of reliable numerical simulations requires them to be executed in an automated manner with explicit control of the approximation made. Since there are no reliable a priori methods to control the approximation errors, adaptive methods must be applied where the mesh resolution is determined in a local fashion based on the spatial distribution of the solution and errors associated with its numerical approximation. In the cases of cardiovascular blood flows the use of boundary layer meshes is central to the ability to effectively perform the flow simulations due to their favorable attributes, i.e., high aspect ratio, orthogonal, layered and graded elements near the viscous walls. The function of mesh adaptation is to convert a given mesh into the desired mesh consistent with the mesh improvement information. The techniques

M. Zhou · O. Sahni · H. J. Kim · M. S. Shephard · K. E. Jansen (✉)
SCOREC, Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute,
110 8th St., Troy, NY 12180, USA
e-mail: kjansen@scorec.rpi.edu

C. A. Figueroa · C. A. Taylor
Bioengineering Department, Stanford University,
Stanford, CA 94305, USA

of anisotropic mesh adaptation with boundary layer maintenance developed in [23] have been used in this study.

Besides accurate description of the complicated geometric model and high resolution spacial discretization (mesh), we need to assign boundary conditions which can represent absent parts of the cardiovascular system properly. There are billions of blood vessels in the cardiovascular system and each of the vessels interact with each other due to the closed loop nature of the system. The cardiovascular system is truncated depending on the region of interest and absent parts of the system should be represented with boundary conditions. Otherwise, the computed flow and pressure are not physiologically realistic because the boundary conditions do not consider flow demands, structure, and physiology of the absent upstream/downstream vascular networks of the system. To consider these interactions, more sophisticated boundary conditions have been developed. These boundary conditions use relationships between pressure and flow and assign weak pressure or flow rate boundary conditions as the boundary conditions [9,17,20,34]. Further, blood vessel walls are not rigid but deform in response to blood flow and pressure and models to represent these effects must be included to capture the altered flow and traction fields near the wall, as well as the overall pressure and flow wave propagation phenomena in the system [6–8,19,21,33].

Inclusion of the aforementioned models into cardiovascular simulations often results in a very large scale computation. The ability to solve problems of practical interest in a reasonable time requires the use of massively parallel computers and a flow solver that is both computationally efficient and highly scalable. An implicit finite element flow solver, PHASTA, has demonstrated the ability to meet these performance requirements. PHASTA employs a partition of the mesh into a set of parts distributed over the processing cores for both the formation and implicit solution of the finite element system [24,25]. PHASTA has been shown to be an effective tool for a broad range of applications including cardiovascular simulations.

This paper seeks to apply state of the art modeling at an extreme scale to demonstrate what will become possible in the near future of cardiovascular simulation. To achieve this goal, the paper is organized as follows: Sect. 2 reviews the numerical method and the parallel structure of our implicit solver. Section 3 presents the blood flow simulation in a patient-specific 3D "whole" body model to demonstrate the large scale geometric complexity and adaptivity. Section 4 demonstrates the strong scalability (time-solution compression) of our approach on both a rigid wall and a deformable wall simulation of an abdominal aorta aneurysm (AAA) on very large meshes and very large core counts. Section 5 draws conclusions from this study.

## 2 Numerical methods

To model blood flow in deformable arteries we employ a coupled momentum method for fluid–solid interactions problems as introduced in [8]. In this method, a shear-enhanced membrane model for the vessel walls is used along with monolithic coupling, which results in a robust method. For fluid domain $\Omega$ with its boundary $\Gamma$ and solid domain $\Omega^s$ with its boundary $\Gamma^s$, the following governing equations are considered for velocity $\mathbf{v}$, pressure $p$ and wall displacement $\mathbf{u}$.

Given $\mathbf{f} : \Omega \times (0,T) \to \mathcal{R}^{n_{sd}}$, $\mathbf{f}^s : \Omega^s \times (0,T) \to \mathcal{R}^{n_{sd}}$, $\mathbf{g} : \Gamma_g \times (0,T) \to \mathcal{R}^{n_{sd}}$, $\mathbf{h} : \Gamma_h \times (0,T) \to \mathcal{R}^{n_{sd}}$, $\mathbf{g}^s : \Gamma_g^s \times (0,T) \to \mathcal{R}^{n_{sd}}$, $\mathbf{h}^s : \Gamma_h^s \times (0,T) \to \mathcal{R}^{n_{sd}}$, $\mathbf{v}_0 : \Omega \to \mathcal{R}^{n_{sd}}$, $\mathbf{u}_0 : \Omega^s \to \mathcal{R}^{n_{sd}}$ and $\mathbf{u}_{0,t} : \Omega^s \to \mathcal{R}^{n_{sd}}$, find $\mathbf{v}(\mathbf{x},t)$, $p(\mathbf{x},t)$ and $\mathbf{u}(\mathbf{x}^s,t)$ $\forall \mathbf{x} \in \Omega$, $\forall \mathbf{x}^s \in \Omega^s$, $\forall t \in (0,T)$ such that the following is satisfied

$$
\begin{aligned}
&\rho \mathbf{v}_{,t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nabla \cdot \mathfrak{T} + \mathbf{f} \\
&\nabla \cdot \mathbf{v} = 0 \\
&\rho^s \mathbf{u}_{,tt} = \nabla \cdot \mathfrak{S}^s + \mathbf{f}^s \qquad (1) \\
&\text{where} \quad \mathfrak{T} = \mu(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \\
&\text{and } \mathfrak{S}^s = \mathcal{C} : \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)
\end{aligned}
$$

with fluid–solid interface condition $\forall (\mathbf{x},t) \in \Gamma_{fsi} \times (0,T)$: $\mathbf{t_n} = -\mathbf{t}^s_{\mathbf{n}^s}$, where $\mathbf{t_n}$ is the surface traction vector on the fluid domain and $\mathbf{t}^s_{\mathbf{n}^s}$ is on the solid domain, with $\mathbf{n}$ being unit outward normal at $\Gamma$ while $\mathbf{n}^s$ at $\Gamma^s$. $n_{sd}$ is the number of the spatial dimensions. The boundary $\Gamma$ of the fluid domain is divided into a Dirichlet boundary portion $\Gamma_g$, a Neumann boundary portion $\Gamma_h$, and a (shared) fluid–solid interface portion $\Gamma_{fsi}$ (note $\mathbf{n} = -\mathbf{n}^s$ at $\Gamma_{fsi}$). Similarly, the boundary $\Gamma^s$ of the solid domain is divided into a Dirichlet boundary portion $\Gamma^s_g$, a Neumann boundary portion $\Gamma^s_h$, and a fluid–solid interface portion $\Gamma_{fsi}$. A no-slip condition is used to the fluid–solid interface portion $\Gamma_{fsi}$. Here, vessel walls are assumed to be a thin-walled structure leading to $\Omega^s = \xi \times \Gamma_{fsi}$ and $\Gamma^s_h = \xi \times \partial\Gamma^s_h$, where $\xi$ is the wall thickness. Currently, we assigned a uniform thickness but a non-uniform wall thickness can be also assigned [2]. Furthermore, density $\rho$, dynamic viscosity $\mu$ of the blood, and density $\rho^s$ of the vessel walls are assumed to be constant. $\mathcal{C}$ is a fourth-order tensor of material constants. Additionally, $\mathbf{f}$ and $\mathbf{f}^s$ are external body forces on the fluid and solid domains, respectively.

The standard Galerkin finite element formulation (based on discrete trial solution and weight spaces, $\boldsymbol{\mathcal{S}}^m_h$, $\mathcal{W}^m_h$ and $\mathcal{P}^m_h$, as defined in [36]) is: find $\mathbf{v} \in \boldsymbol{\mathcal{S}}^m_h$ and $p \in \mathcal{P}^m_h$ such that for any $\mathbf{w} \in \boldsymbol{\mathcal{W}}^m_h$ and $q \in \mathcal{P}^m_h$, the following is satisfied

$$
\begin{aligned}
B_G(\mathbf{w},q) = & \int_{\Omega} \left\{ \mathbf{w}\cdot(\rho\mathbf{v}_{,t} + \rho\mathbf{v}\cdot\nabla\mathbf{v} - \mathbf{f}) \right\} d\mathbf{x} \\
& + \int_{\Omega} \nabla\mathbf{w} : (-p\underline{I} + \underline{\tau}) d\mathbf{x} - \int_{\Omega} \nabla q \cdot \mathbf{v} d\mathbf{x} \\
& - \int_{\Gamma_h} \mathbf{w}\cdot\mathbf{h} ds + \int_{\Gamma} q\mathbf{v}\cdot\mathbf{n} ds \\
& + \xi \int_{\Gamma_{fsi}} \left\{ \mathbf{w}\cdot\rho^s\mathbf{v}_{,t} + \nabla\mathbf{w} : \underline{\sigma}^s(\mathbf{u}) - \mathbf{w}\cdot\mathbf{f}^s \right\} ds \\
& - \xi \int_{\partial\Gamma_h^s} \mathbf{w}\cdot\mathbf{h}^s dl \qquad (2)
\end{aligned}
$$

where the acceleration term has been written as the time derivative of the velocity rather than as the second time derivative of the displacement field, since the goal is to express the vessel wall equations in terms of the fluid unknowns to maintain monolithic coupling. Furthermore, to represent the vascular bed downstream of the computational domain, the traction $\mathbf{h}$ at outlets can be calculated using a boundary condition based on prescribed traction, resistance, impedance, or a three-element Windkessel model [34,35]. As the Galerkin method is well known to be unstable for equal-order interpolation of the velocity and pressure, a stabilized semi-discrete finite element method [4,10,30,36] is utilized in this study.

After discretization in space, implicit integration in time is performed using a generalized-$\alpha$ method [11]. The resulting non-linear algebraic equations are linearized to yield a system of equations which is solved using iterative solvers, e.g., GMRES [22] is applied to the linear system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ (where $\boldsymbol{b}$ is the right-hand-side or residual of the equations and $\boldsymbol{A}$ is the left-hand-side or linearized tangent of $\boldsymbol{b}$ with respect to unknowns, both of which are evaluated by numerical integration over all the interior and boundary elements in the mesh, and $\boldsymbol{x}$ is the update in the unknowns that needs to be computed at any given non-linear iteration step). Having described our discretization procedure, we now turn our attention to the parallel implementation that allows us to achieve strong scalability on massively parallel computers.

The aim of the current parallelization procedures is to perform subject-specific flow simulations in a clinically relevant time-frame by achieving (strong) scalability over massive level of parallelism involving hundreds of thousands of cores. All computations are based on a decomposition or partition of the mesh into parts with equal work load. The term part is used to denote a set of mesh entities whereas term partition is used to indicate collection of all parts (i.e., together all the parts within a partition comprise the aggregate mesh). Graph or hypergraph [3,12] based partitioning schemes are currently used as they are more suitable for unstructured meshes. Here, partitioning based on elements

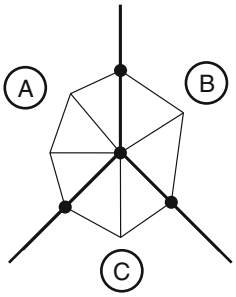


**Fig. 1** *Solid dots* indicate shared *dofs*

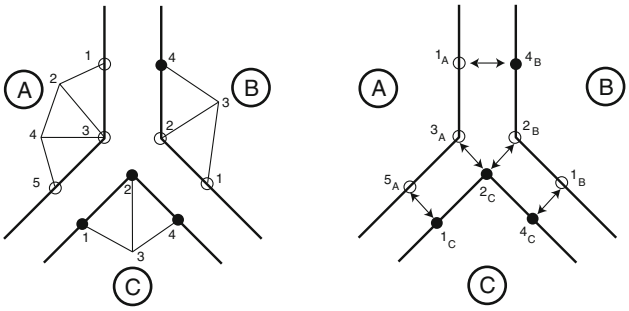


**Fig. 2** Control relationships are established between multiple images of shared *dofs* (*dofs* are locally numbered on each part). *Solid circle* denotes an owner image whereas hollow ones indicate non-owners

is applied as it is natural for equation-formation stage (i.e., during formation of $\boldsymbol{A}$ and $\boldsymbol{b}$), making it highly scalable. Note that during the equation-formation stage the computational load (in any processing core) depends on the elements present in the local part whereas in the equation-solution stage it depends on the degrees-of-freedom (*dofs*), or unknowns in the system of equations, on that part. So long as the *dof* balance is preserved, such a partitioning also maintains the scalability in the equation-solution stage.

Since each element is uniquely assigned to a single part, *dofs* are shared on inter-part boundaries, see solid dots in Fig. 1, whereas a non-shared *dof* resides solely on one part (i.e., interior to a part). Therefore, the communication effort related to shared *dofs* is peer-to-peer, i.e., required only among interacting neighboring parts, and grows with the number of *dofs* shared at inter-part boundaries. A concept of a *partition-graph* is currently used to describe the parts, and their interactions, within a partition. Each partition-graph vertex represents a part whereas each partition-graph edge represents interaction between a pair of parts sharing *dofs*. The interaction between neighboring parts is defined based on shared *dofs*, as shown in Fig. 1, where every part sharing it has an image of it. One image among all images of a shared *dof* is assigned the *owner* thereby making all the other images to be non-owners, see Fig. 2. Such a control relationship for shared *dofs* allows each owner image to be "in-charge" for data accumulation and update and in turn limits communications between owner and non-owners

(i.e., non-owner images do not communicate to each other). Furthermore, data exchange is done only for vector entries (e.g., in $\boldsymbol{b}$), since the assembly in entries of $\boldsymbol{A}$ is performed at the local level without regard to the neighboring parts. An entry is considered complete when it has the same value in a partitioned (or parallel) case as it would have in the unpartitioned case. To obtain complete values, a two pass communication strategy is applied as described in [24].

After forming $\boldsymbol{A}$ and $\boldsymbol{b}$, and obtaining complete values in $\boldsymbol{b}$, the next step involves finding the solution update vector $\boldsymbol{x}$. Krylov iterative solution techniques are currently used (e.g., GMRES [22]), where repeated products of $\boldsymbol{A}$ with a series of vectors (say, $\boldsymbol{p}$) are carried out to construct an orthonormal basis that is used to approximate $\boldsymbol{x}$. In this series, the outcome of any matrix–vector product is another vector $\boldsymbol{q}$ ($=\boldsymbol{A}\boldsymbol{p}$) which is used to derive the subsequent vector in the series while the first vector in the series is derived from the residual vector $\boldsymbol{b}$ (which contains complete values). Due to the distributive property of $\boldsymbol{A}\boldsymbol{p}$ product, the resulting vector $\boldsymbol{q}$ will contain on-part (incomplete) values. To obtain complete values in $\boldsymbol{q}$, two-pass communication strategy is used as before. Before the next product in the series orthonormalization is required, where the norm of vector $\boldsymbol{q}$, and its dot-product with previous vectors, are computed. The on-part or local dot-product is first computed which is followed by a global sum (based on collective communication) to obtain a complete dot-product. This series of steps leads to an orthonormal basis of vectors to approximate $\boldsymbol{x}$ and marks the end of a single non-linear iteration step. In a transient, non-linear problem, multiple non-linear iterations are performed at each time step to obtain a converged (non-linear) solution at that time step before advancing to the next time step.

## 3 "Whole" body simulation

In this first set of simulations, we demonstrate the capability of our methods by applying them to compute physiologically realistic blood flow and pressure in a subject-specific "whole" body model within a short time frame. The "whole" body model is constructed from computer tomography (CT) data, which covered the entire body from the neck to the feet of a female patient. Figure 3 shows the volume rendering of the CT data and the constructed geometric model projected into an iso-surface image of the CT data.

The layout of the "whole" body model is illustrated in Fig. 4, along with the labels of sections where the velocity magnitude are reported. The figure also shows the inflow waveform and the three-element Windkessel model [34,35] assigned to each outlet. The geometric model consists of seventy eight arteries, starting from the root of the aorta and including major branch vessels and main branches of the upper-body and lower-body vasculature. The discretization

**Fig. 3** Volume rendering of CT data (*left*) and the geometric model in iso-surface of CT data (*right*)

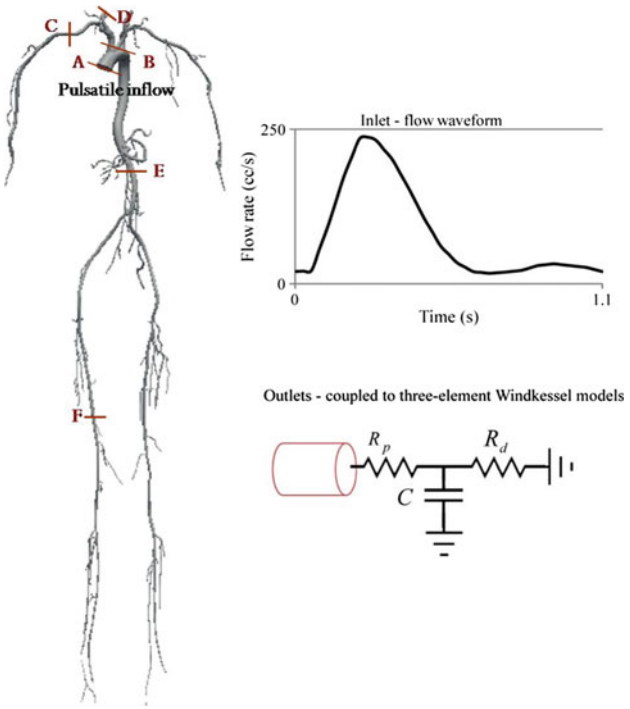


**Fig. 4** The "whole" body model with the labels of sections (*left*) and the boundary conditions (*right*)

of this geometric model is challenging because it is a large complex model that spans a large range of vessel diameters. The length of the model is roughly 150 cm whereas the diameter varies from 3 cm at the aortic inlet to 0.2 cm in small branch vessels. To generate an efficient finite element mesh, we need an adaptive tool that considers non-uniform complex geometries and solution fields. Additionally, versatile boundary conditions are needed to obtain realistic flow and

pressure waveforms in this model. This model consists of one inlet, 78 outlets, and the vessel walls. If the boundary conditions of the inlet and outlets are not properly set based on the flow demands of the downstream vessels absent in the geometric model, the flow distribution will be determined solely by the resistance of the geometry and lead to unphysiologic flow and pressure distribution.

For the inlet, a Womersley velocity profile [37] with a mean flow rate of 83 cc/s (Fig. 4) is assigned. The Womersley number is 5.6 with a cardiac cycle of 1.1 s. We solve for blood velocity and pressure for four cardiac cycles until the periodic solution fields are obtained. Results from the last cardiac cycle are presented. A uniform time step size of 1 ms is used for all the simulations. The viscosity and the density of the blood are assumed to be $\mu = 0.04$ dynes s/cm$^2$ and $\rho = 1.06$ g/cm$^3$, respectively. No-slip conditions are assigned on the blood vessels assuming rigid walls and a three-element Windkessel model is applied at the outlet surfaces to represent the downstream vascular networks absent in the geometric model using an analogy between the electric circuit and the blood flow in the cardiovascular system (Fig. 4). Flow distribution to each outlet was determined based on literature data [38] and we adjusted the parameter values of the three-element Windkessel model for different outlets to match both the flow distribution and the measured brachial artery pulse pressure [14,27] at the level of the descending thoracic aorta. Note that the computed mean pressure of the outlets was smaller than the mean pressure of the brachial artery because of the pressure drops due to the resistance to blood flow in the geometric model. Note that the walls will be assumed rigid in this case since a deforming wall simulation of such a long geometry would require a foundation model which is still under development.

To construct an efficient finite element mesh, we initially created a fairly coarse unstructured mesh with two semi-structured layers of the constant thickness of 0.004 cm within the boundary layer. Figure 5 presents magnified views for different arteries along with those of the adapted mesh. Due to the high ratio between the model's total height and the diameter of small arteries (150/0.3 cm = 500), we varied the initial local mesh size for different blood vessels from 0.03 to 0.1 cm on the basis of the outlet diameter. The initial mesh had 1.86 million nodes and 7.1 million elements but is still relatively uniform and isotropic in the interior volume and does not have high resolution on small blood vessels (e.g., Radial artery in Fig. 5 shows only tens of triangles on the outlet surface).

The solution based, anisotropic, boundary layer mesh adaptation was used to improve the mesh quality. After four sets of flow computation and mesh modification, we obtained an adapted mesh with 9.0 million nodes and 42.8 million elements. Figure 5 depicts magnified views of selected arteries in comparison with the initial mesh. The effect of flow

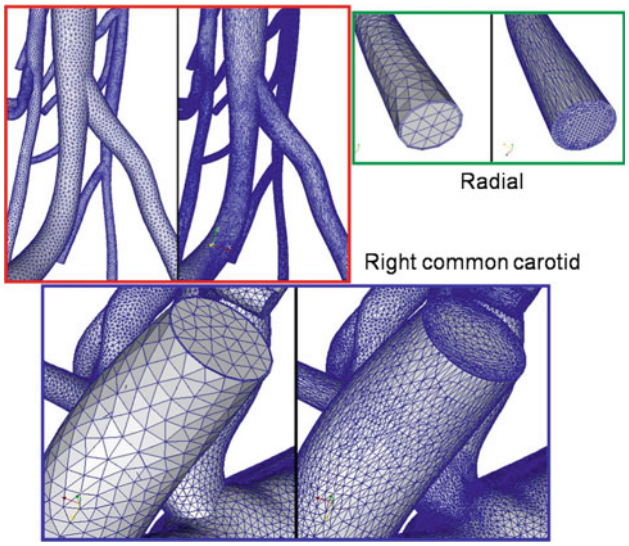


**Fig. 5** The comparison of initial and adapted meshes

features can be clearly seen on the adapted meshes. Elongated elements aligned in the flow direction are created in the regions where one velocity component is varying sharply relative to the other ones such as near the vessel walls due to boundary layers. Isotropic elements are created where flow variation in each direction is relatively similar, for example, in branching regions. The minimum local size in the adapted mesh is 0.004 cm while the maximum size is 0.8 cm. The adaptation increased the mesh size by a factor of six (42 million/7 million). However, the local resolution was increased up to a factor of 25 (0.1/0.004 cm) in some regions. To achieve the same level of resolution, uniform refinement will increase the mesh by $25 \times 25 \times 25 = 15.6$ thousand times in the 3D model. This is one of the advantages of the anisotropic adaptation. Moreover, we observe that in the case of an adapted boundary layer mesh, the layered and graded elements are maintained near the walls. The transition of the elements at the interface between the layered part and the interior tetrahedral part of the mesh is also smooth in the adapted mesh, avoiding mesh based artifacts in the solution due to a rapid change in the element size. Note that, the newly created nodes are projected ("snapped") onto the solid model surface to improve the geometric approximation.

Figure 6 shows the distribution of the pressure fields of the "whole" body model at peak systole, early diastole, and end diastole. At peak systole, the aortic pressure of the model reaches 120 mmHg (left in Fig. 6). After the aortic valve is closed in early diastole, aortic inflow is zero and the pressure decreases. The pressure of the upper body drops below 100 mmHg and the pressure of the lower body is higher than the pressure of the aorta due to the deceleration of the flow (middle in Fig. 6). The pressure continues to decrease until the end of diastole and the pressure in the outlets surfaces reaches the minimum value of 60 mmHg (right in Fig. 6).

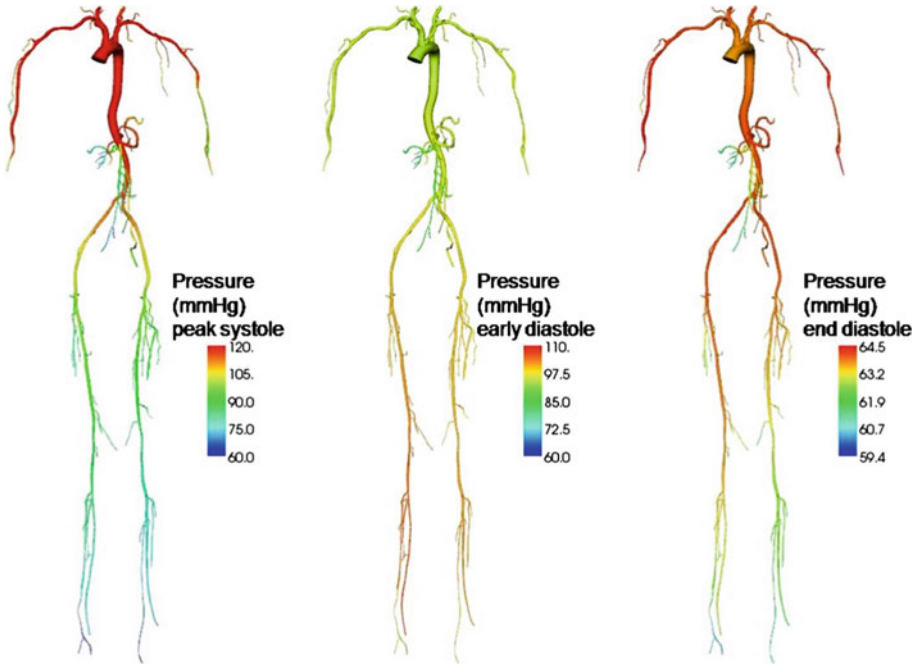


**Fig. 6** Pressure contours of the "whole" body model using the adapted mesh at peak systole, early diastole, and end diastole

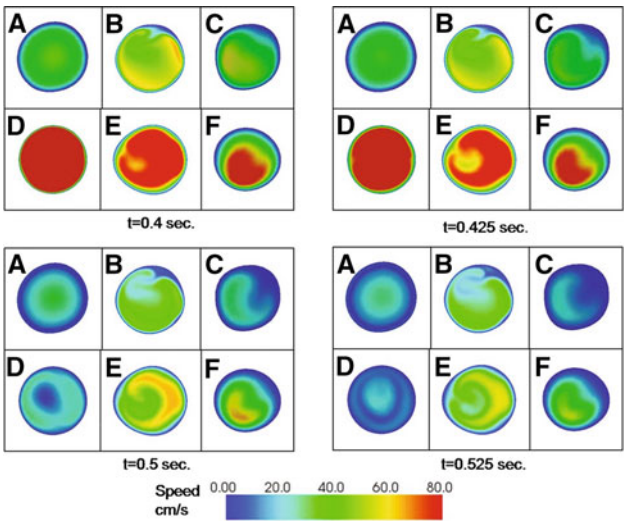


**Fig. 7** Velocity magnitude on selected cut-planes (defined in Fig. 4) obtained on the adapted mesh at four instants

In Fig. 7, velocity magnitude on six selected cut-planes (see Fig. 4) are shown at four instants of a cardiac cycle. When the inflow is decelerating, some cut-planes experience complex flow structures due to the complex geometry and the interactions of the boundary conditions. Small scale flow features are developed at this phase and the adapted mesh demonstrates its ability to capture them.

Figure 8 demonstrates temporal variation of the pressure and the flow rate at various outlets. A phase lag of pressure relative to flow is observed, which is realistic. The computed flow and pressure waveforms are physiologically realistic by assigning proper boundary conditions at the outlet surfaces.

Figure 9 shows the computed wall shear stress fields at peak systole, early diastole, and end diastole. At peak systole, the wall shear stress is high due to the high flow from the aortic inlet. The wall shear stress at end diastole is low due to the low aortic inflow. However, the transiency in wall shear stress mostly affected the aorta and big vessels and not the small vessels due to the differences in the diameter.

Figure 10 shows the wall shear stress of the aorta in the mesenteric region in a magnified view. Note that the wall shear stress was plotted in the same scale ($0-20$ dynes/cm$^2$). In the magnified view, we see a smooth distribution of wall shear stress on the model, which demonstrates another benefit of using the adaptive boundary layer mesh [23].

Figure 11 compares the wall shear stress over vessel walls in a magnified view obtained on the initial ($M_0$) and the adapted mesh ($M_4$) at peak systole. Obvious differences are observed between these two meshes. The wall shear stress on the initial mesh already captured the overall distribution patterns, but it is not as smooth as the field on the adapted mesh and is over predicted in some regions.

The simulation of the adapted mesh were performed on Kraken (Cray XT5) at NICS, using 2,048 processing cores and the simulation took 90 min (3,000 CPU h) for one cardiac cycle. These results indicate that we can run cardiovascular simulations with complex geometry and flow characteristics such as the "whole" body model in a reasonable time frame because the flow solver has strong scalability (see the following section), which is very important for time critical simulations.

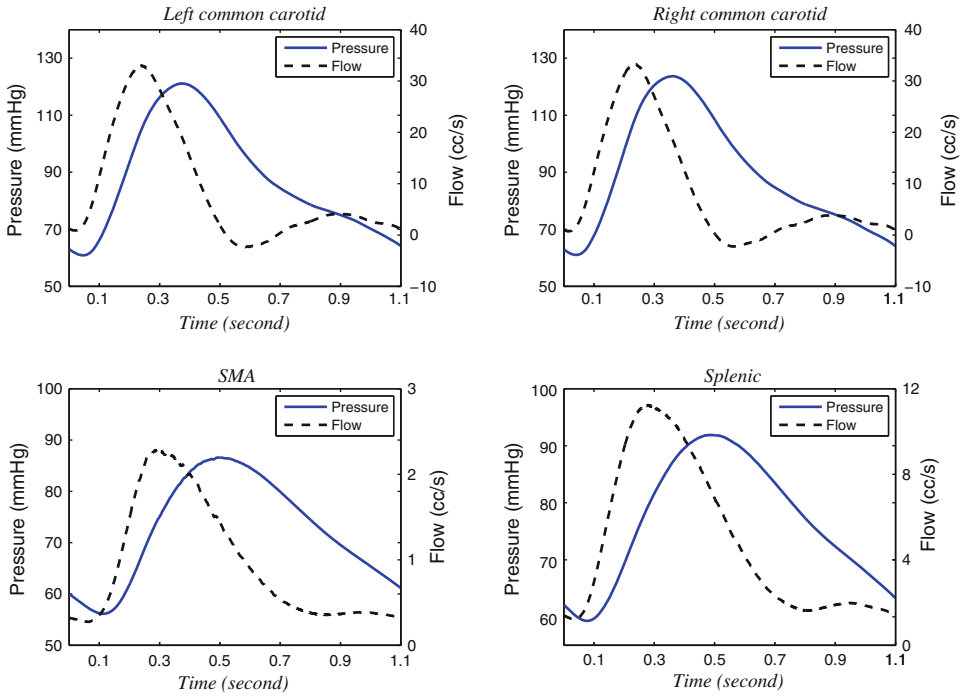


**Fig. 8** Pressure and volumetric flow rate computed on the adapted mesh at various outlets in a "whole" body model. SMA stands for Superior Mesenteric Artery

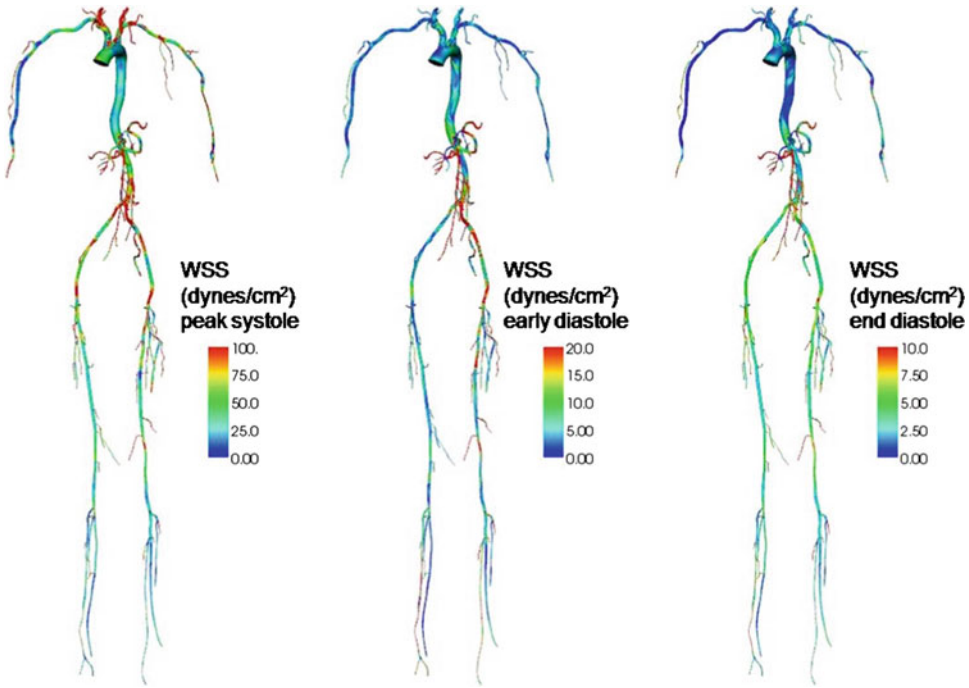


**Fig. 9** Wall shear stress of the "whole" body model using the adapted mesh at peak systole, early diastole, and end diastole

## 4 Scaling study of AAA model with FSI

In this section, we consider an abdominal aorta aneurysm (AAA) model which is shown in Fig. 12. Simulations of this model have been carried out for multiple cardiac cycles in a previous study [15]. This study has shown that the unsteady pulsatile flow, combined with the complex aneurysm shape, leads to unstable free shear layers that undergo transition during the deceleration phase of the cardiac cycle (post-peak of systole). Obtaining a grid-independent study of this transitional flow requires several cycles of anisotropic adaptivity. While grid independence may be obtained for some

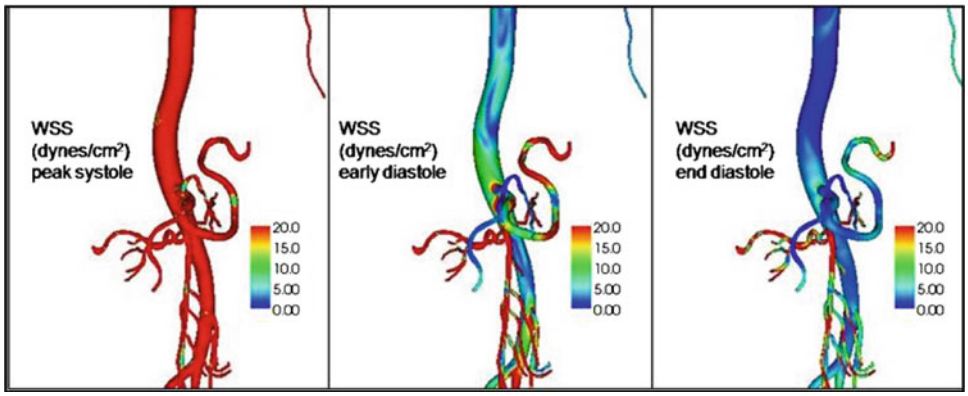


**Fig. 10** Wall shear stress of the descending thoracic aorta using the adapted mesh at peak systole, early diastole, and end diastole

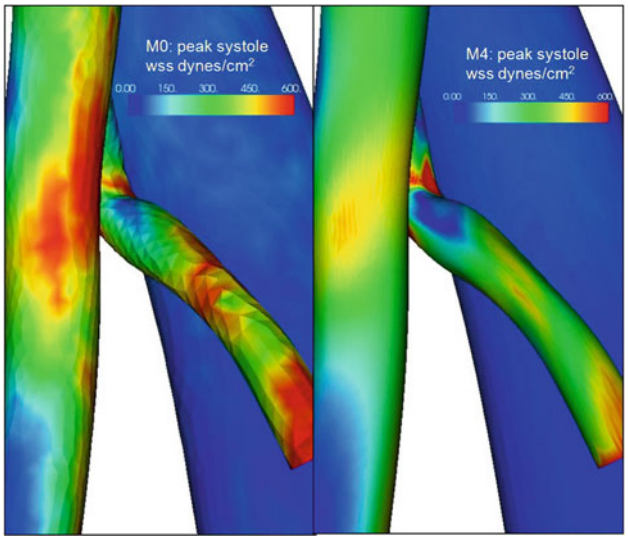


**Fig. 11** Comparison of wall shear stress on the initial and the adapted meshes at peak systole

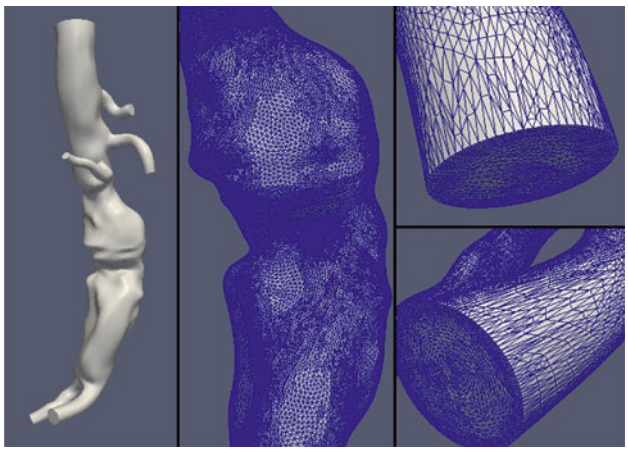

**Fig. 12** Geometry and a mesh of an AAA model

quantities (e.g., flow splits) on meshes with $O(10^7)$ elements, other, more challenging quantities (e.g., wall shear stress) appear to require $O(10^8)$ elements. When an adaptive scheme is successful, it equally distributes error among all the elements. At this point, the next adaptation cycle would subdivide all existing elements leading to an eightfold increase in mesh size which also happens to be the gold standard for grid independence if the solution change, in the quantity of interest, is made acceptably small. Typically, such simulations are out of reach for problems of this complexity but the extreme scalability of our solver combined with coming computer architecture appears poised to make such simulations feasible. To be clear, we do not yet have sustained access to the machines that would be necessary to run multiple cardiac cycles on such a large mesh but the purpose of the studies in this section is to demonstrate scalability of our software on short runs and thereby establish feasability of such work as these computers become more available. To this end, a mesh with 1.07 billion elements is considered which is obtained from adaptive refinement of a 133 million element mesh that was itself the result of several cycles of anisotropic adaptive refinement of [23]. The refinement was performed on 2,048 processing cores on NICS Kraken (Cray XT5) and the process took less than half a minute. In this study, we focus on the strong scalability (time-solution compression) of the methodology in use, which is required for time critical simulations, for example, surgical planning. We also compare the absolute time usage and scaling factors for this model using two models of the vessel walls: deformable wall (FSI) and rigid wall.

In this case, the scaling studies are performed on the near-petascale system, Intrepid (IBM BG/P system) at ALCF, ANL. The number of cores covered range from 4,096 to 163,840, i.e., from 1 rack to all 40 racks of BG/P covering the full system-scale.

One fundamental requirement of scalable finite element solvers is the balance of the work load on each part. Graph and hypergraph based partitioning schemes are currently used as they are more suitable for unstructured mesh calculations that process irregular structures and connections. Zoltan

**Table 1** The average number of regions and vertices of a 1.07 billion element mesh of an AAA model, along with the imbalance ratios

| Num. of parts | Ave. num. Rgns | Ave. num. vtxs | Rgn. imbal. | Vtx. imbal. |
|---|---|---|---|---|
| 4,096 (base) | 261,667 | 48,054 | 1.0549 | 1.0736 |
| 8,192 | 130,834 | 24,732 | 1.0549 | 1.0841 |
| 16,384 | 65,417 | 12,822 | 1.0549 | 1.0990 |
| 32,768 | 32,708 | 6,707 | 1.0549 | 1.1210 |
| 65,536 | 16,354 | 3,550 | 1.0549 | 1.1719 |
| 98,304 | 10,903 | 2,465 | 1.0549 | 1.2117 |
| 131,072 | 8,177 | 1,908 | 1.0549 | 1.1782 |
| 163,840 | 6,542 | 1,561 | 1.0554 | 1.1950 |

**Table 2** Strong scaling results on an AAA model with rigid wall and with O($10^9$) elements up to 163,840 cores on Intrepid:IBM BG/P (execution time is measured in seconds)

| Intrepid: 1.07$B$ element | | Eqn. form. | | Eqn. soln. | | Total | |
|---|---|---|---|---|---|---|---|
| Num. of core | Avg. elem./core | Time | s-factor | Time | s-factor | Time | s-factor |
| 4,096 (base) | 261,600 | 388.68 | 1 | 456.24 | 1 | 844.92 | 1 |
| 8,192 | 103,800 | 194.48 | 1.00 | 233.86 | 0.98 | 428.34 | 0.99 |
| 16,384 | 65,400 | 97.48 | 1.00 | 120.90 | 0.94 | 218.38 | 0.97 |
| 32,768 | 32,700 | 49.01 | 0.99 | 62.33 | 0.91 | 111.34 | 0.95 |
| 65,536 | 16,350 | 24.59 | 0.99 | 33.70 | 0.85 | 58.29 | 0.91 |
| 98,304 | 10,900 | 16.42 | 0.99 | 23.56 | 0.81 | 39.98 | 0.88 |
| 131,072 | 8,175 | 12.37 | 0.98 | 18.65 | 0.76 | 31.02 | 0.85 |
| 163,840 | 6,540 | 10.27 | 0.95 | 15.34 | 0.74 | 25.61 | 0.82 |

hypergraph [3] is utilized in this section to provide balanced partitions for the finite element solver scaling studies. The average number of mesh regions and vertices for each partition along with imbalance ratios are given in Table 1. A wide range of partitions (with an associated wide range of average number of elements per part) is considered. Specifically, the average number of elements per part varies from as high as above two hundred thousand (required for large simulations on low processor counts) on a partition with 4,096 parts to as few as several thousand (useful for time critical runs) on a partition with 163,840 parts. When a fixed-size problem is spread to more and more parts, the mesh regions are still well balanced since we define mesh regions (finite elements) to be graph nodes (partition objects). However, the vertex imbalance ratio increases which adversely affects the scalability of the flow solver (equation solution phase). When the mesh is partitioned to 163,840 parts, the average number of mesh regions per part is about six thousand and the vertex imbalance goes up to 19.5%. This impacts scaling since the part with the most vertices has 19.5% more work to do than the average part in the equation solution phase, which slows down all the other processors. Methods based on iterative load balancing operations between neighboring parts are under development to improve the partition and in turn parallel performance [39]. It is also worth noting that the average number of vertices per part (column 3 of Table 1) does not decrease at the same rate as the regions when a fixed-size problem is spread into more and more parts due to an increase in mesh vertices shared at inter-part boundaries which is a second source of degradation to the scaling.

The scaling study of the finite element solver is performed on Intrepid using all four cores on each compute node, and allowing one part on each processing-core. A Womersley velocity profile [37] is applied at the inlet and both rigid or deformable walls are considered. At the outlets a three-element Windkessel model [34,35] is applied. As stated above, this study is meant only to establish the feasibility of such simulations and therefore involves only 20 time steps. From experience, we know that this is sufficient to establish the relative time usage for different partitions. This time usage is measured and the scaling factor is computed (i.e., $s\_factor = t_{base} \times np_{base}/(t_i \times np_i)$) and shown for both the rigid wall model in Table 2 and for the deformable wall model in Table 3.

In both tables, we measure the execution time in terms of equation formation and equation solution so as to understand the effect of partitioning on scaling, specifically due to the imbalance in mesh vertices. Both tables also provide the

**Table 3** Strong scaling results on an AAA model with FSI and with O($10^9$) elements up to 163,840 cores on Intrepid:IBM BG/P (execution time is measured in seconds)

| Intrepid: 1.07$B$ element | | Eqn. form. | | Eqn. soln. | | Total | | Percentage of time on bdry. elem. |
|---|---|---|---|---|---|---|---|---|
| Num. of core | Avg. elem./core | Time | s-factor | Time | s-factor | Time | s-factor | |
| 4,096 (base) | 261,600 | 402.21 | 1 | 460.83 | 1 | 863.04 | 1 | 2.1 |
| 8,192 | 103,800 | 202.92 | 0.99 | 236.28 | 0.98 | 439.20 | 0.99 | 2.5 |
| 16,384 | 65,400 | 101.94 | 0.99 | 122.67 | 0.94 | 224.63 | 0.96 | 2.8 |
| 32,768 | 32,700 | 51.67 | 0.97 | 63.03 | 0.91 | 114.70 | 0.94 | 2.9 |
| 65,536 | 16,350 | 26.46 | 0.95 | 34.05 | 0.85 | 60.51 | 0.89 | 3.7 |
| 98,304 | 10,900 | 17.59 | 0.95 | 23.82 | 0.81 | 41.41 | 0.88 | 3.5 |
| 131,072 | 8,175 | 13.43 | 0.94 | 18.85 | 0.76 | 32.28 | 0.84 | 3.9 |
| 163,840 | 6,540 | 11.13 | 0.90 | 15.35 | 0.75 | 26.53 | 0.81 | 3.5 |

scaling factor and execution time (including for both work components). In the rigid wall simulation, we observe a near-perfect strong scaling (95% or above) for equation-formation stage all the way to the full system-scale. On the other hand, strong scaling for equation-solution stage drops below 90% at 65,536 cores and onwards. The strong scaling for both work components together is at 90% or above till 65,536 cores and drops to 82% while using 163,840 cores or all the 40 racks of BG/P in Intrepid. Therefore, with 20 time steps in this case, each step takes only 1.25 s for an implicit solve (over 163,840 cores) on an adapted mesh with over 1 billion elements.

In terms of computational time we see that the deformable wall simulation only modestly increases the computational time, e.g., 3.5% in equation formation and 1% in equation solution on the partition with 4,096 parts. Furthermore, this work does not harm scalability as can be seen by comparing the scaling factors where we observe only a slight degradation in the deformable simulations (0.82 as compared to 0.81 at 160k cores). This is likely due to the boundary element imbalance since this is where the deformable wall contributions are calculated. However, the total number of boundary elements is 3.1 million, which is very small compared to the total number of interior elements (0.29%). Assessing the imbalance of boundary elements is complicated. On one hand, it might be viewed as large since, for example, the average number of boundary element is 192 on the partition with 16,384 parts while the part with the most boundary elements contains 4,087 boundary elements. This would seeem to suggest a problem until one realizes that even on this part, the boundary elements are only 6.7% of the interior elements. Clearly, even this massive boundary element imbalance is well sheltered by the relatively larger interior element integration work because no parallel synchronization steps are required between these two work components. The percentage of time used on boundary element integration is given in the last column of Table 3. Clearly, while no attempt was made to balance the work of the boundary elements explicitly, the modest degradation in scalability of that work component has a very small impact on the scalability of the overall flow solve. More sophisticated FSI models will of course face greater challenges. Specifically, loosely coupled schemes will likely have to take greater care that each component is well balanced if separate solves are performed on each component (introducing a communication barrier that prevents the sheltering described above). Still, so long as the volume integration work dominates, the prospects are good.

## 5 Conclusions

Cardiovascular simulation is advancing rapidly to include more physiologically realistic geometry, boundary conditions, and coupled structural response. A framework to address these issues was described and demonstrated on two problems. While neither application was presented as a complete physiological study, both provided demonstrations of the future potential for "extreme scale" computing. Specifically, the "whole body" simulation demonstrated the potential for anisotropic adaptivity to match local solution resolution needs and provide a dramatic reduction in computational effort relative to isotropic meshing approaches. The AAA case further demonstrated that, as compute core counts continue to climb, not only can much larger problems be solved but the solution time can also be dramatically compressed if the solver can scale to very large core count with a modest number of elements per processor. This was demonstrated not only for rigid wall simulations but also for deformable wall simulations for the first time.

**Acknowledgments** This work is supported in part by the US Department of Energy under grant DE-FC02-06ER25769 and in part by the National Science Foundation under grant 0749152. Computing resources used were provided by support from NSF through Teragrid resources including systems at TACC, PSC and NICS. Computing

resources used at the Rensselaer Computational Center for Nanotechnology Innovations was funded by the State of New York, RPI and IBM. This research also used computing resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the US Department of Energy under contract DE-AC02-06CH11357. The solution presented herein made use of the linear algebra solution library provided by Acusim Software Inc. and meshing and geometric modeling libraries by Simmetrix Inc. The authors would like to thank the following individuals for their kind assistance: Michael Schrameyer, Gilwoo Choi, Ankit Garg, and Jessica Shih.

## References

1. Bazilevs Y, Gohean JR, Hughes TJR, Moser RD, Zhang Y (2009) Patientspecific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the jarvik 2000 left ventricular assist device. Comp Methods Appl Mech Eng 198:3534–3550
2. Bazilevs Y, Hsu M-C, Benson DJ, Sankaran S, Marsden AL (2009) Computational fluidstructure interaction: methods and application to a total cavopulmonary connection. Comput Mech (submitted)
3. Boman E, Devine K, Fisk LA, Heaphy R, Hendrickson B, Leung V, Vaughan C, Catalyurek U, Bozdag D, Mitchell W (1999) Zoltan home page. http://www.cs.sandia.gov/Zoltan
4. Brooks AN, Hughes TJR (1982) Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comp Methods Appl Mech Eng 32:199–259
5. Cebral JR, Castro MA, Burgess JE, Pergolizzi RS, Sheridan MJ, Putman CM (2005) Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics models. AJNR Am J Neuroradiol 26(10):2550–2559
6. Farhat C, Geuzaine P (2004) Design and analysis of robust ale time-integrators for the solution of unsteady flow problems on moving grids. Comp Methods Appl Mech Eng 193:4073–4095
7. Fernandez MA, Le Tallec P (2003) Linear stability analysis in fluid–structure interaction with transpiration. Part ii: numerical analysis and applications. Comp Methods Appl Mech Eng 192:4837–4873
8. Figueroa CA, Vignon-Clementel IE, Jansen KE, Hughes TJR, Taylor CA (2006) A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. Comp Methods Appl Mech Eng 195(41–43):5685–5706
9. Formaggia L, Gerbeau JF, Nobile F, Quarteroni A (2002) Numerical treatment of defective boundary conditions for the Navier–Stokes equations. SIAM J Numer Anal 40(1):376–401
10. Franca LP, Frey SL (1992) Stabilized finite element methods ii. the incompressible Navier–Stokes equations. Comp Methods Appl Mech Eng 99(2–3):209–233
11. Jansen KE, Whiting CH, Hulbert GM (2000) Generalized-$\alpha$ method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. Comp Methods Appl Mech Eng 190(3–4):305–319
12. Karypis G, Kumar V (1999) Parallel multilevel k-way partitioning scheme for irr. graphs. SIAM Rev 41:278–300
13. LaDisa JF, Olson LE, Molthen RC, Hettrick DA, Pratt PF, Hardel MD, Kersten JR, Warltier DC, Pagel PS (2005) Alterations in wall shear stress predict sites of neointimal hyperplasia after stent implantation in rabbit iliac arteries. Am J Phys Heart Circ Phys 288:H2465–H2475
14. Laskey WK, Parker HG, Ferrari VA, Kussmaul WG, Noordergraaf A (1990) Estimation of total systemic arterial compliance in humans. J Appl Physiol 69(1):112–119
15. Les AS, Shadden SC, Figueroa CA, Park JM, Tedesco MM, Herfkens RJ, Taylor CA, Dalman RL (2009) Quantification of hemodynamics in abdominal aortic aneurysms during rest and exercise using magnetic resonance imaging and computational fluid dynamic. Ann Biomed Eng (submitted)
16. Li Z, Kleinstreuer C (2005) Blood flow and structure interactions in a stented abdominal aortic aneurysm model. Med Eng Phys 27(5):369–382
17. Migliavacca F, Balossino R, Pennati G, Dubini G, Hsia TY, de Leval MR, Bove EL (2006) Multiscale modelling in biofluidynamics: application to reconstructive paediatric cardiac surgery. J Biomech 39(6):1010–1020
18. Perktold K, Peter R, Resch M (1989) Pulsatile non-newtonian blood flow simulation through a bifurcation with an aneurysm. Biorheology 26(6):1011–1030
19. Peskin CS, McQueen DM (1995) A general method for the computer simulation of biological systems interacting with fluids. Symp Soc Exp Biol 49:265–276
20. Quarteroni A, Ragni S, Veneziani A (2001) Coupling between lumped and distributed models for blood flow problems. Comp Vis Sci 4(2):111–124
21. Quarteroni A, Tuveri M, Veneziani A (2000) Computational vascular fluid dynamics: problems, models and methods. Comp Vis Sci 2(4):163–197
22. Saad Y, Schultz MH (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7:856–869
23. Sahni O, Müller Y, Jansen KE, Shephard MS, Taylor CA (2006) Efficient anisotropic adaptive discretization of the cardiovascular system. Comp Methods Appl Mech Eng 195:5634–5655
24. Sahni O, Zhou M, Shephard MS, Jansen KE (2009) Scalable implicit finite element solver for massively parallel processing with demonstration to 160k cores. In: Proceedings of IEEE/ACM SC'09, Finalist paper for the Gordon Bell Prize
25. Shephard MS, Jansen KE, Sahni O, Diachin LA (2007) Parallel adaptive simulations on unstructured meshes. J Phys Conf Ser 78. doi:10.1088/1742-6596/78/1/012053
26. Soerensen DD, Pekkan K, de Zelicourt D, Sharma S, Kanter K, Fogel M, Yoganathan AP (2007) Introduction of a new optimized total cavopulmonary connection. Ann Thorac Surg 83(6):2182–2190
27. Stergiopulos N, Segers P, Westerhof N (1999) Use of pulse pressure method for estimating total arterial compliance in vivo. Am J Physiol Heart Circ Physiol 276(2):H424–H428
28. Stuhne GR, Steinman DA (2004) Finite-element modeling of the hemodynamics of stented aneurysms. J Biomech Eng 126(3):382–387
29. Tang BT, Cheng CP, Draney MT, Wilson NM, Tsao PS, Herfkens RJ, Taylor CA (2006) Abdominal aortic hemodynamics in young healthy adults at rest and during lower limb exercise: quantification using image-based computer modeling. Am J Physiol Heart Circ Physiol 291(2):H668–H676
30. Taylor CA, Hughes TJR, Zarins CK (1998) Finite element modeling of blood flow in arteries. Comput Methods Appl Mech Eng 158(1–2):155–196
31. Taylor CA, Draney MT (2004) Experimental and computational methods in cardiovascular fluid mechanics. Annu Rev Fluid Mech 36:197–231
32. Taylor CA, Draney MT, Ku JP, Parker D, Steele BN, Wang K, Zarins CK (1999) Predictive medicine: computational techniques in therapeutic decision-making. Comp Aided Surg 4(5):231–247
33. Tezduyar TE, Behr M, Liou J (1992) A new strategy for finite element computations involving moving boundaries and

interfaces -the deforming-spatial-domain/space–time procedure: I. the concept and the preliminary numerical tests. Comput Methods Appl Mech Eng 94:339–351
34. Vignon-Clementel IE, Figueroa CA, Jansen KE, Taylor CA (2008) Outflow boundary conditions for three-dimensional simulations of non-periodic blood flow and pressure fields in deformable arteries. Comput Meth Biomech Biomed Eng (submitted)
35. Vignon-Clementel IE, Figueroa CA, Jansen KE, Taylor CA (2006) Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries. Comp Methods Appl Mech Eng 195(29–32):3776–3796
36. Whiting CH, Jansen KE (2001) A stabilized finite element method for the incompressible Navier–Stokes equations using a hierarchical basis. Int J Numer Meth Fluids 35:93–116
37. Womersley J (1955) Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. J Physiol 127:553–563
38. Zamir M, Sinclair P, Wonnacott TH (1992) Relation between diameter and flow in major branches of the arch of the aorta. J Biomech 25(11):1303–1310
39. Zhou M (2009) Petascale adaptive computational fluid dynamics. Ph.D. thesis, Rensselaer Polytechnic Institute