A tool for the efficient FE-simulation of cardio-vascular flow

Jens Müller (a), Onkar Sahni (a), Kenneth E. Jansen (a), Mark S. Shephard (a) Charles A. Taylor (b)

(a) Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy NY, USA,

(b) James H. Clark Center, Stanford University, Stanford, CA, USA

Summary:

Surgical therapy in the cardio-vascular system aims at restoring blood flow to compromized organs and tissues at physiological conditions. The goal is to avoid reverse flow and low shear areas as these are indicators for recurrent cardiovascular disease. Alternate treatments cannot be tested in the patient and since the surgeon does not have the optimal design tools he traditionally bases the therapy exclusively on medical imaging and his own experience.

In presenting an integrated software tool which assists the physician in constructing and evaluating a combined anatomic/physiologic model to assess the outcome of alternative treatment plans for an individual patient we establish a new paradigm of predictive medicine. The tool integrates medical imaging, geometric modeling, discretization, FEM-based computational fluid dynamics as well as scientific visualization.

This presentation will first summarize our work in the fields of medical imaging and image processing, geometric model generation, mesh generation, numerical simulation and visualization by using concrete physiological examples. We will then focus on the numerical method used and their optimization. The complex 3D geometry of the cardiovascular system and the pulsatile nature of blood flow require an efficient discretization method. We present a SUPG-method using linear tetrahedra for the incompressible Navier-Stokes equations. We increase the method's efficiency by an anisotropic mesh adaptation procedure by around one order of magnitude in terms of computer time and memory requirements. We demonstrate our method using a selected number of physiologic models as for example a stenotic porcine aorta as well as a human aortic model.

Keywords:

Computational blood flow, finite element method, anisotropic mesh adaptivity

1 Introduction

In the predictive medicine paradigm we have proposed [1], a physician would use diagnostic data to reconstruct a model of an individual's anatomy and physiology, and then use simulation techniques, implemented in a Simulation-Based Medical Planning software system, to predict the response of the patient to alternate treatments under different physiologic states. Specifically, physicians could evaluate the efficacy of a treatment plan on the basis of predicted functional data including flow rate, pressure losses, shear stress (correlated to disease processes), and in the future, the probability that clots will form or intimal thickening (leading to blockages in vessels and grafts) will develop.

Clearly, the success of predictive medicine techniques is dependent upon how reliably the mathematical model reflects the actual anatomy and physiology of the individual and on how efficiently the simulations can be performed. Patient-specific anatomic models are derived from 3D imaging techniques including CT and MRI using image segmentation and surface extraction methods. Image-based geometric models can be constructed from the extracted surfaces. The overall patient-specific physiologic model is an abstract model based on the equations governing blood flow and on patient specific diagnostic data, e.g. measurements of blood flow and pressure. Once the patient-specific preoperative model is created, the consequences of treatments can be predicted using numerical methods with postoperative models that reflect the proposed treatments.

Section 2 summarizes the steps we take from acquiring medical imaging data to the geometric vessel model generation. This includes image segmentation and surface extraction as well as image-based geometric modeling. Section 3 discusses the finite element modeling of hemodynamic flow. In section 4 we present the anisotropic mesh adaptation method that we use to enhance the flow solver. In the last section we demonstrate the efficiency of our numerical method by comparing wall shear stress (WSS) values (a challenging but important quantity in the understanding of cardiovascular disease) obtained on uniformly refined meshes to those obtained on anisotropically adapted meshes for various models.

2 Software Frameworks for Image Based Geometric Modeling

In this section we will first discuss image segmentation along with surface extraction and then focus on image based geometric modeling.

2.1 Image segmentation and surface extraction

Geometric image segmentation algorithms rely on the deformation of an initial geometry towards image features. Because these methods can be controlled in a way that produces topologically-closed geometric results which are well-suited to geometric modeling problems, we are actively developing algorithms for image-based model construction using the level set method. One distinguishing characteristic of the level set method is that as a result of its Eulerian perspective, the method can handle arbitrary changes in the topology of evolving geometries. For example, for a 3D level set simulation, when a given segmentation is initialized with multiple seed geometries the level set method accommodates the potential collision and merging of these geometries without modification.

2.2 Image-based geometric modeling

We have developed methods for creating geometric models of the vasculature from volume data sets based on analytic representations of the vessel surfaces. These geometric models of cardiovascular anatomic structures can be created from CT or MR 3D image data, shown in Fig.1, as follows: First, vessel centerlines are obtained by the method described in [2]. Then, a set of 2D slice probes geometrically defined as rectangles imbedded in 3D space are positioned normal to the centerlines. Once a set of slice probes have been defined they can be used to extract a 2D slice of image intensity data from a given volumetric data. Segmentation of the vessel lumens for each 2D data slice is then performed and results in a set of contours that are interpolated with a curve of an appropriate type: circle, ellipse, or spline. A NURB surface is then lofted through the set of curves and capped to create a solid model of a vessel. Once several solid models of vessels have been created they can be joined together to form a single solid model using a union Boolean operation [3]. Geometric modeling operations (skinning, capping and Boolean operations) are performed using the Parasolid (UGS, Inc. St. Louis, MO) geometry kernel integrated into our software and accessed via a generic geometric

modeling layer. The layer provides the geometric modeling functionality to create curves, surfaces, and solids together, and allows operations on them independent of any particular modeling kernel. We have applied these methods to create patient-specific models from CT and MRI data sets. These pre-operative models can be modified using geometric solid modeling operations to include treatment plans. Figure 1d) shows geometric models created from MRA data and displays a preoperative model and three alternate treatment plans for a case of lower extremity occlusive disease. The pre-operative model was then modified *manually* to reflect the potential treatments. The development of more efficient methods to create alternate pre-operative models is a current research topic.

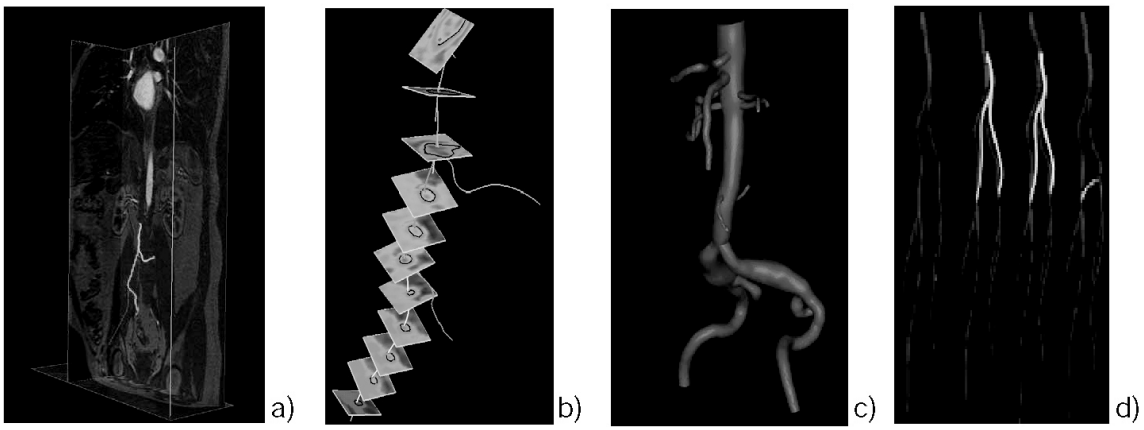

Figure 1 . a) Vessel path construction from MRI data, b) 2D slice probes for contour extraction, c) surface lofting, d) construction of alternate models.

3 FE-Simulation of Blood Flow

This section presents the finite element formulation for the transient incompressible Navier-Stokes equation governing blood flow. We use a stabilized finite element formulation that has been shown to be robust, accurate and stable on a variety of flow problems (see for example [4] and [5]). In particular, we employ the streamline upwind/Petrov-Galerkin (SUPG) stabilization method introduced in [6].

3.1 Governing Equations

The governing equations for blood flow, assuming Newtonian constitutive behavior and rigid blood vessel walls, are the transient incompressible Navier-Stokes equations

$$u_{i,i} = 0, \qquad \rho u_{i,t} + \rho u_j u_{i,j} = -p_{,i} + \tau_{ij,j} + f_i. \tag{1}$$

The variables are: the velocity u_i, the pressure p, the density ρ, and the viscous stress tensor τ_{ij}. The summation convention is used throughout, i.e., sum on repeated indices. For incompressible flow the viscous stress tensor τ_{ij} is modeled by the symmetric strain rate tensor

$$\tau_{ij} = \mu(u_{i,j} + u_{j,i}), \tag{2}$$

where μ is the viscosity. Finally, f_i is a body force or source term, neglected in arterial flow analysis. The above system of equations is supplemented with an appropriate set of boundary conditions that are prescribed on the model boundary of the blood vessels. The no-slip condition is imposed on the vessel walls that are assumed to be rigid and impermeable. A time varying velocity profile, based on physiological values, may be prescribed at the inlet. A zero exit pressure in a weak sense is imposed at the traction-free outlet.

3.2 Flow Solver

Finite element methods are based on the weak form of the governing equations (1) which is obtained by taking the $L^2(\Omega)$-inner product of the entire system with weight functions. Integration by parts is then performed to shift the spatial derivatives onto the weight functions. To derive the finite element discretization from the weak form of (1) discrete weight and solution function spaces must be introduced. Let $\overline{\Omega} \subset \boldsymbol{R}^N$ represent the closure of the physical spatial domain (i.e., $\Omega \cup \Gamma$ where Γ is

the boundary) in *N* dimensions; where only *N*=3 is considered here. The boundary is decomposed into portions with natural boundary conditions, Γ_h, and essential boundary conditions, Γ_g, i.e., $\Gamma_g \cup \Gamma_h$. In addition, $H^1(\Omega)$ represents the usual Sobolev space of functions with square-integrable values and derivatives on Ω. Subsequently Ω is discretized into n_{el} finite elements, $\overline{\Omega}_e$. With this, one can define the discrete solution and weight function spaces for the semi-discrete formulation as:

$$
\begin{aligned}
S_h^k &= \{v \mid v(\bullet,t) \in H^1(\Omega)^N, t \in [0,T], v|_{x\in\overline{\Omega}_e} \in P_k(\overline{\Omega}_e)^N, v(\bullet,t) = g \text{ on } \Gamma_g\}, \\
W_h^k &= \{w \mid w(\bullet,t) \in H^1(\Omega)^N, t \in [0,T], w|_{x\in\overline{\Omega}_e} \in P_k(\overline{\Omega}_e)^N, w(\bullet,t) = 0 \text{ on } \Gamma_g\}, \\
\Pi_h^k &= \{p \mid p(\bullet,t) \in H^1(\Omega), t \in [0,T], p|_{x\in\overline{\Omega}_e} \in P_k(\overline{\Omega}_e)\},
\end{aligned}
\tag{3}
$$

$P_k(\overline{\Omega}_e)$ denoting the space of all polynomials defined on $\overline{\Omega}_e$, complete up to order $k \geq 1$. Let us emphasize that the local approximation space $P_k(\overline{\Omega}_e)$ is the same for both the velocity and pressure variables. This is possible due to the stabilized nature of the formulation to be introduced below. These spaces represent discrete subspaces of the spaces in which the weak form is defined.

The stabilized formulation used in the present work is based on the formulation described in [4]. Given the spaces defined above, the semi-discrete Galerkin finite element formulation is applied to the weak form of the governing equations (1) as: Find $\boldsymbol{u} \in \boldsymbol{S}_h^k$ and $p \in \Pi_h^k$ such that

$$
\begin{aligned}
0 &= B_G(w_i, q; u_i, p) \\
&= \int_\Omega \{w_i(\rho u_{i,t} + \rho u_j u_{i,j} - f_i) + w_{i,j}(-p\delta_{ij} + \tau_{ij}) - q_{,i}u_i\} d\Omega \\
&+ \int_{\Gamma_h} \{w_i(p\delta_{ij} - \tau_{ij})n_j + q u_i n_i\} d\Gamma,
\end{aligned}
\tag{4}
$$

for all $\boldsymbol{w} \in \boldsymbol{W}_h^k$ and $q \in \Pi_h^k$. The boundary integral term arises from the integration by parts and is only carried out over the portion of the domain without essential boundary conditions. Since the standard Galerkin method is well known to be unstable for equal-order interpolation of the velocity and pressure, additional stabilization terms are introduced as follows: Find $\boldsymbol{u} \in \boldsymbol{S}_h^k$ and $p \in \Pi_h^k$ such that

$$
\begin{aligned}
0 = B(w_i, q; u_i, p) = B_G(w_i, q; u_i, p) &+ \sum_{e=1}^{n_{el}} \int_{\overline{\Omega}_e} \{\tau_M(u_j w_{i,j} + q_{,i}/\rho)L_i + \tau_C w_{i,j} u_{j,j}\} d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\overline{\Omega}_e} \{w_i \hat{u}_j u_{i,j} + \overline{\tau} L_j w_{i,j} L_k u_{i,k}\} d\Omega,
\end{aligned}
\tag{5}
$$

for all $\boldsymbol{w} \in \boldsymbol{W}_h^k$ and $q \in \Pi_h^k$. We have used L_i to represent the residual of the *i*th momentum equation

$$
L_i = \rho u_{i,t} + \rho u_j u_{i,j} + p_{,i} - \tau_{ij,j} - f_i. \tag{6}
$$

The second term on the right-hand-side in the first line of the stabilized formulation (5), represents the typical SUPG stabilization added to the Galerkin formulation for the incompressible set of equations (see [7]). The first term in the second line of (5) was introduced in [4] to overcome the lack of mass conservation introduced as a consequence of the momentum stabilization in the continuity equation. The second term on this line was introduced to stabilize this new advective term. The stabilization parameters are described in [5].

To develop a discrete system of algebraic equations, the weight functions w_i and *q*, the solution variables u_i and *p*, and their time derivatives are expanded in terms of the finite element basis functions. Gauß quadrature of the spatial integrals results in a system of first-order, nonlinear differential-algebraic equations. Finally, this system of non-linear ordinary differential equations is discretized in time via a generalized-α time integrator [5], resulting in a non-linear system of algebraic equations. This system is in turn linearized with Newton's method which yields a linear algebraic system of equations that is solved (at each time step) and the solution is updated for each of the Newton iterations. The linear algebra solver of [8] is used to solve the linear system of equations.

4 Anisotropic Mesh Adaptation

The accuracy of the numerical solution depends on the spatial discretization of the physical domain, i.e., on the process of subdividing the domain into a finite number of elements, also referred to as the mesh. In general, the desired element sizes in different directions are influenced by the physical and geometric features of the problem which can vary significantly. In many physical problems, including blood flow, the solution exhibits strong anisotropic features creating a demand for elements which are aligned with the solution's anisotropy. In realistic cases such information, required to compute the desired solution field to an acceptable level of accuracy, is unknown *a priori*. An efficient approach to overcome this difficulty is to apply an iterative adaptive procedure where the errors introduced due to spatial discretization are controlled within a specified tolerance. An anisotropic adaptive procedure modifies the mesh in a way such that the local mesh resolution becomes adequate in all directions.

In this section, we describe the anisotropic adaptive procedure that we employ. We describe the basis for the Hessian strategy, a method suited when using linear finite elements, and review the concept of mesh metric tensors which is used to represent the desired mesh anisotropy. We then provide the details of the anisotropic mesh size field computation.

4.1 General Components

An adaptive method involves a feed-back process that evaluates the quality of the computed solution using *a posteriori* error estimation. To control the discretization errors mesh modification procedures are applied to change the local mesh resolution. The key ingredients of an adaptive meshing method include:

- A posteriori error estimation/indication: Estimating and/or obtaining an indication of the discretization error based on the quality of the computed solution. See [9] or [10] for a survey.

- Size field construction: Transformation of the error information into a size field information that describes the desired mesh resolution over the domain.

- Modifying strategy: Altering the mesh based on the size field information using local mesh modifications [11,12] or global remeshing [13,14].

The above components are general enough to include anisotropic mesh adaptation techniques provided each of them incorporates appropriate directional information. The remainder of this section elaborates on these key components, except the last item which has been described in [15,16].

4.2 Hessian Strategy

To obtain directional information of the error we use the Hessian strategy [17], a method where the field's second derivatives are used to extract information on the error distribution. This directional information is converted into a mesh metric field which prescribes the desired element size and orientation. Recall that a function which is sufficiently smooth can be expanded into a Taylor series. When trying to interpolate that function with a piecewise linear function, the interpolation error will have a lowest order error term proportional to the second derivatives of the function, which covers a large portion of the discretization error [18]. The interpolation error $\| e \|_{\infty,K}$ in 3D in the L_∞ norm defined on an element *K*, given the solution is sufficiently regular, then can be measured as follows [19]:

$$\| e \|_{\infty,K} \le c_1 \max_{x \in K} \max_{\boldsymbol{v} \in K} \left\langle \boldsymbol{v}, \left| H(x) \right| \boldsymbol{v} \right\rangle \le c_1 \max_{x \in K} \max_{\boldsymbol{e} \in E_K} \left\langle \boldsymbol{e}, \left| H(x) \right| \boldsymbol{e} \right\rangle \tag{7}$$

where c_1 is a constant independent of element parameters, $\boldsymbol{v}$ is any vector contained in the element, E_K is the set of element edges and $|H|$ is the absolute value of the Hessian matrix of the solution, i.e., consists of absolute eigenvalues.

The Hessian strategy involves the computation of the symmetric matrix of second derivatives that can be decomposed as $H = R \Lambda R^T$, where R is the eigenvector matrix and $\Lambda = diag(\lambda_k)$ is the diagonal matrix of eigenvalues (*k*=1,2,3 in 3D). The directions associated with the eigenvectors $\boldsymbol{p}_k$ are referred to as principal directions and the eigenvalues λ_k are then equivalent to the second derivatives along the local principal directions. The strategy is based on the idea that a high magnitude of an eigenvalue implies a high error in the direction associated with the corresponding eigenvector, so a small element

size would be desired in this direction. Conversely, a low eigenvalue magnitude in a particular eigendirection suggests that the element size can be large in this direction.

4.3 Mesh Metric Field

To perform anisotropic mesh adaptation requires a way to define the desired element size distribution over the domain. Mesh metric tensors, symmetric and positive definite, are used to represent a size field defining the desired mesh anisotropy at a point (see for example, [20]). The associated quadratic form $\langle \boldsymbol{x}, M\boldsymbol{x} \rangle = 1$ defines a mapping of an ellipsoid in the physical space into a unit sphere in the *transformed/metric* space. In other words, any vector $\boldsymbol{x}$ at a point *P* assumes a unit value where distances are measured in the metric space. The stated goal of the mesh adaptation algorithm is to yield a mesh with regular elements in the metric space where each edge $\boldsymbol{e}$ must satisfy the following relation:

$$\langle \boldsymbol{e}, M\boldsymbol{e} \rangle = 1. \tag{8}$$

For further details on mesh modifications and element quality measures in the transformed space see references [15,16]. The same references also provide the details of the discretization of the mesh metric field over the domain along with its implementation.

4.4 Size Field Computation

A crucial step in the process is the construction of a size field based on the Hessian that can be input to the mesh adaptation module. The key point in the construction of a size field is to attempt to uniformly distribute the estimated error in all directions. To achieve a suitable mesh resolution in different directions, a uniform distribution of local errors is applied in the principal directions which leads to $h_k^2 |\lambda_k| = \varepsilon$, where ε is the user specified tolerance for the error and h_k is the desired size in the *k*th principal direction. To compute the Hessian matrix we reconstruct the second derivatives at each node by using the derivative information of the computed solution from the patch S of all elements K surrounding a node. In the first step we recover the gradient at node *i* by taking the volume weighted average of gradients on elements in the patch S_i. The same procedure is applied to each term of second derivatives to obtain the recovered Hessian matrix.

A mesh metric tensor is then obtained at each node by calculating a scaled eigenspace of the recovered Hessian matrix as $\overline{M} = R\overline{\Lambda}R^T$, where R is the eigenvector matrix and $\overline{\Lambda} = \Lambda / \varepsilon$ is the diagonal matrix of scaled eigenvalues. Truncation values $h_{\min}$ and $h_{\max}$ for mesh sizes are specified to limit the eigenvalues. One reason for truncating the element size, in terms of edge lengths, is to avoid singular metrics. For example, it is necessary to apply $h_{\max}$ in case an eigenvalue is zero (or close to zero) in the direction where the solution does not vary. The modified eigenvalues of the Hessian matrix then become:

$$\tilde{\lambda}_k = \min(\max(\varepsilon^{-1}|\lambda_k|, \frac{1}{h_{\max}^2}), \frac{1}{h_{\min}^2}) \quad \text{(k=1,2,3).} \tag{9}$$

The final mesh metric field is constructed at each node through multiplication of the diagonal matrix of modified eigenvalues $\tilde{\Lambda} = diag(\tilde{\lambda}_k)$ with the matrix R of eigenvectors: $M = R\tilde{\Lambda}R^T$.

5 Results

5.1 Stenotic Porcine Aorta

We first study the performance of the adaptive method by applying it to the simulation of pulsatile flow in a porcine aorta with a stenosis and a bypass graft. The dimensions of the model are approximately 10*cm* in length while the vessel's diameter at the inflow is approximately 1.6*cm*. Blood was modelled as an incompressible Newtonian fluid with a constant density of 1.06 g/cm^3 and a constant viscosity of 0.04 *dyn s/cm*^{2}. The geometric model is shown in Fig. 2a). The inlet flow is prescribed as a pulsatile Womersley flow profile that is based upon PC-MRI through-plane flow rate data, see [21]. Figure 2b) shows the velocity at an instant during the cardiac cycle and the inset depicts the velocity profile at a point near the centre of the inflow plane for one cardiac cycle (together with the instant the velocity snapshot is taken at).

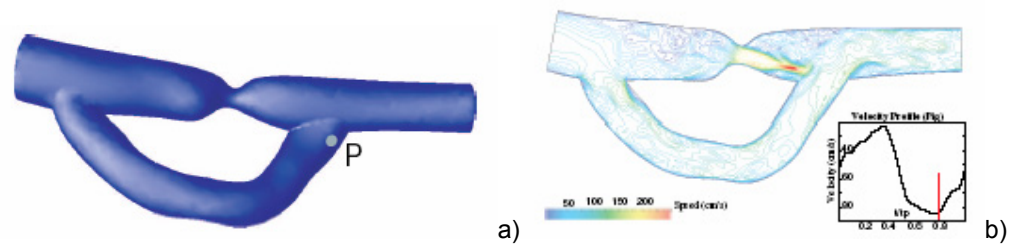

Figure 2. Model and instantaneous flow field of a stenotic porcine aorta together with inflow profile.

We obtain simulation results on five different meshes, four of them being uniform and one being an anisotropically adapted mesh. The latter was obtained by first computing the solution on the coarsest uniform mesh, then calculating the interpolation error in terms of the Hessians of the averaged speed field. The Hessians then are converted into a mesh metric field that in turn is handed over to the mesh modification tool box. Figure 3a) shows a clip plane through the adapted mesh and Fig. 3b), bottom, its surface. On top of Fig. 3b) we show the surface of a *uniform* mesh that consists of approximately the same number of nodes as the adapted mesh.

The clip plane parallel to the flow direction in Fig. 3a) of the main vessel further illustrates the effect of the mesh modification procedure. We observe that slender and elongated elements were created in regions where flow speed and also second derivatives are highest, such as near the vessel walls (boundary layer effect). This becomes more obvious when comparing Fig. 3a) and the flow field at an instant, depicted in Fig. 2b. Other regions where major mesh adaptation takes place (see the zooms in Fig. 3a)) are the stenosis area, the re-entrant corners of the model and the main artery where the re-directed flow impacts on the vessel wall. Typically, element sizes are predominantly small but isotropic in the re-entrant corner, reflecting the well known fact that solution behavior near that region tends to be nearly singular. Small isotropic elements also dominate the stenosis area, accounting for strongly varying flow in that region; for example, back-flow occurs within a particular interval during the cycle.

We analyze the transient behavior of the WSS in two different ways. First, we compare the WSS distribution for different meshes at the same instant t during the cardiac cycle with period tp. Fig. 4a) shows the WSS on the model's surface at instant t/tp = 0.8. Secondly, in Fig. 4b) we compare the time dependent behavior of the WSS for the different meshes at one particular point on the vessel wall, located near the downstream vessel branching and labeled as P (depicted in Fig. 2a)). The WSS are computed on a series of uniform meshes ranging from 3456 nodes to over 857K nodes and on an adapted mesh consisting of 42196 nodes. While the WSS values near the stenosis are much higher than at the other locations, we observe that in both cases the values heavily depend on the number of degrees of freedom that are used for their computation, see Fig. 4. On the coarser meshes the calculated WSS generally is too high, whereas the values tend to converge for the finer meshes. Even though the convergence is not uniform during the entire cardiac cycle we do observe convergence patterns in parts of the cycle. For location P, see Fig. 2, assuming that the finest uniform mesh of over 857K nodes (corresponding to over 4.7 million tetrahedra) sufficiently resolves all the flow features including the derivative quantities, we observe that the anisotropically adapted mesh follows the pattern of the finest uniform mesh well in the first half of the cycle but is of slightly lower magnitude after the peak in the cycle's second half.

The convergence behavior can be regarded sufficient for the uniform meshes, i.e. the mesh consisting of around 217K nodes and the mesh consisting of 857K nodes yield roughly the same results all over the cycle. The values of the anisotropically adapted mesh with 42K nodes follow those of the finest uniform mesh for the most part. The anisotropic mesh even seems to capture the WSS pattern better than the 217K mesh at the beginning of the cycle. Minor inconsistencies occur in the second half of the cycle. Considering the complex flow pattern resulting in a convergence behavior which cannot be fully classified as spatially and temporally uniform and that is further exacerbated by our attempt to reduce errors in WSS in each point on the surface, we still are able to claim considerable gains in computational time. Even though stringent margins for a gain factor do not seem appropriate for this specific case we do observe that results are far better than those obtained on uniform meshes of twice the number of nodes.

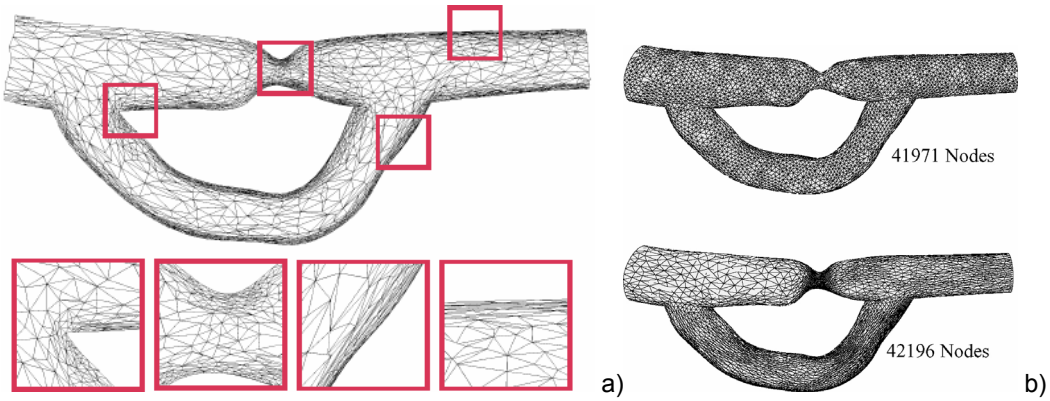

Figure 3. Clip plane through anisotropically adapted mesh (42196 nodes) with zooms as well as uniform and adapted surface meshes.

And, they are quite close to those obtained on meshes of either 217K or 857K nodes, which supports our claim to gain savings of around an order of magnitude with our method.

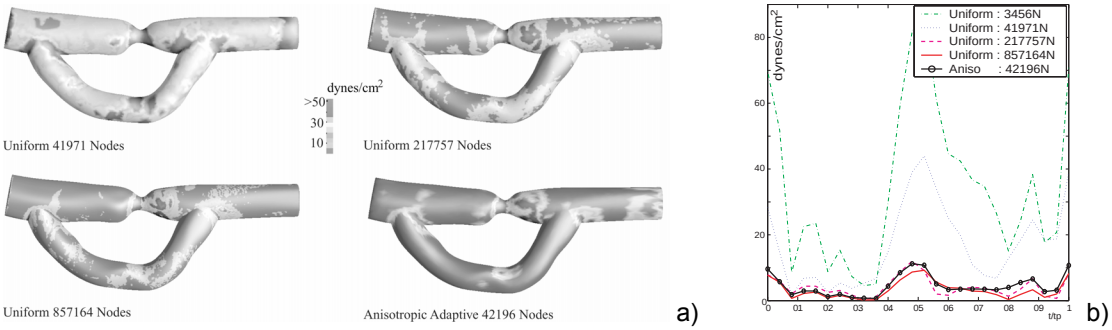

Figure 4. Wall shear stress computed on different meshes. Instantaneous (a) and time dependent (b).

5.2 Descending Human Aorta with Iliacs

We present result for a reduced human descending aorta model including the right and left iliacs, as shown in Fig. 5. Again we will compare WSS values obtained on a series of uniformly adapted meshes to those obtained on an anisotropically adapted mesh, of which we show clip planes at varoius locations in Fig. 5. We observe that anisotropic elements aligned with the flow are created. All the locations shown exhibit significant anisotropy in all directions: The zooms at the bottom right of Fig. 5 display anisotropy in flow direction while the cross sections on top display anisotropy in azimuthal and radial direction. Furthermore, we observe a higher mesh density downstream in the iliacs.

In Fig. 6 we analyze the time dependent WSS behavior for a particular point on the vessel surface for different meshes. Results are obtained on uniform meshes consisting of around 6K, 33K, 180K and 1.1M nodes as well as on an adapted mesh of around 99K nodes. We observe that the mesh of 6K nodes is far too coarse to capture the WSS behavior while the values of the finest uniform mesh tend to have converged and coincide with the values computed on the adapted mesh, hereby suggesting savings of around an order of magnitude.
At this point we would like to point out that at other locations on the wall we observe similar WSS patterns, however, there are locations where the convergence behavior is less obvious. On the other hand, we would like to emphasize that the WSS is a derivative field quantity and therefore is subjected to a lower convergence rate than the primary field quantities. To reproduce accurate spatial and temporal WSS values for pulsatile flow simulations therefore is particularly demanding. Improving WSS accuracy by more sophisticated adaptation techniques is a current research topic.

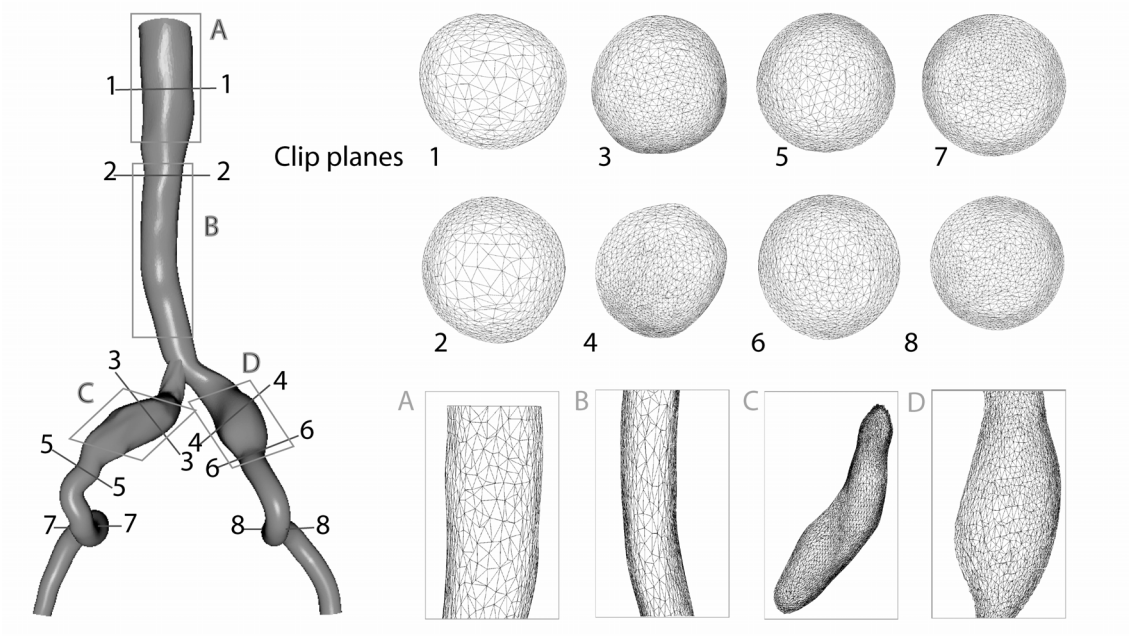

Figure 5. Human aortic model togther with segments of the adapted mesh.

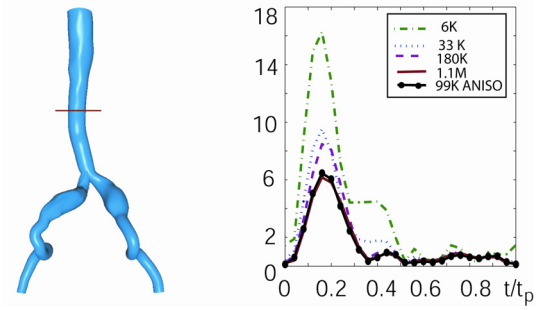

Figure 6. WSS for Human aortic model for different meshes, uniform and anisotropic, together with location.

6 Summary

We have presented a software framework that is designed to support our new paradigm of predictive medicine where the physician uses computational tools to predict and evaluate the outcome of alternate cardiovascular surgical treatments. The tool includes medical imaging and image processing, geometric model generation, mesh generation, numerical simulation and visualization.

We have further presented a method by which hemodynamic finite element simulations can be enhanced. We have demonstrated how computer time can be reduced and at the same time numerical accuracy increased. The method is based on mesh adaptivity and uses error indication together with mesh modification techniques. The error indicator uses interpolation estimates that yield information on the direction in which the mesh should be refined and in which direction it can be coarsened in order to obtain a more accurate solution at considerably reduced costs, at around one order of magnitude in terms of computer time and memory. This approach is novel in computational hemodynamics where it is generally assumed that the solution field the analyst is interested in is accurate enough when it does not change significantly if the underlying computational mesh is uniformly refined. This traditional approach is far too costly in terms of computer resources and disregards the effect of possible singular values of the solution field and particularly its derivatives, on the convergence behavior of the solution. Singularities of that kind are rooted in the finite element formulation for a given model, especially for complex geometries such as blood vessels. Moreover, the

non-adaptive approach neglects anisotropic behavior of the solution fields such that a fine mesh density is used where a coarse density would be sufficient.
Besides efforts to improve the overall patient specific models, future work will incorporate a more stringent error analysis to further accelerate the efficiency of the adaptive method. We will also extend the simulation to larger and therefore more realistic hemodynamic systems.

7 Acknowledgements

We gratefully acknowledge the support of this work by NSF grant ACI-0205741. This work was facilitated through an allocation of advanced computing resources by NPACI (NRAC program) through the support of the National Science Foundation. We would also like to thank the Deutsche Forschungsgemeinschaft DFG for making the presentation of this work possible through a travel grant.

8 References

[1] Taylor, C.A., et al., *Predictive Medicine: Computational Techniques in Therapeutic Decision-Making.* Computer Aided Surgery, **4**, 1999, p. 231-247
[2] Paik, D.S., et al., *Automated Flight Path Planning for Virtual Endoscopy*. Medical Physics. **25**, 1998, p. 629-637
[3] Hoffman, C.M., *Geometric and Solid Modeling: An Introduction*. Morgan Kaufman: San Mateo, CA, 1989
[4] Taylor C.A., Hughes T.J.R, Zarins C.Z, *Finite element modeling of blood flow in arteries.* Comp. Meth. Appl. Mech. Engrg. **158**, 1998, p.155-196
[5] Whiting, C.H, Jansen, K.E, *A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis.* International Journal of Numerical Methods in Fluids, **35,** 2001, p.93–116
[6] Brooks, A.N, Hughes, T.J.R, *Streamline upwind / Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comp. Meth. Appl. Mech. Engng. 32, 1982, p.199–259
[7] Franca, L.P., Frey, S., *Stabilized finite element methods: II. The incompressible Navier-Stokes equations*. Comp. Meth. Appl. Mech. Engng. **99**, 1992, p.209–233
[8] Shakib, F., http://www.acusim.com
[9] Ainsworth, M., Oden, J.T, A Posteriori Error Estimation in Finite Element Analysis, John Wiley & Sons, New York, 2000
[10] Verfürth, R., A Review of Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques, Teubner-Wiley, Stuttgart, 1996
[11] Bänsch, E., *Local refinements in 2 and 3 dimensions*. Impact of Comp. in Sci. and Engng. **3**, 1991, p.181–191
[12] de Cougny, H.L., Shephard, M.S, *Parallel refinement and coarsening of tetrahedral meshes*. Int. J. Numer. Meth. Engng. **46**, 1999, p.1101–1125.
[13] George, P.-L., Borouchaki, H., Laug, P., *An efficient algorithm for 3D adaptive meshing*. Advances in Engineering Software **33**, 2002, p.377–387
[14] George, P.-L., *Tet meshing: construction, optimization and adaptation*, in: Proc. 8th Internatinal Meshing Roundtable, South Lake Tao, CA, USA, 1999
[15] Li, X., Shephard, M.S., Beall, M.W. *3D anisotropic mesh adaptation by mesh modifications*. Comp. Meth. Appl. Mech. Engng. 2005, submitted.
[16] Li, X., *Mesh modification procedures for general 3-d non-manifold domains*, Ph.D. thesis, Rensselaer Polytechnic Institute (Aug. 2003).
[17] Kunert, G. *Toward anisotropic mesh construction and error estimation in the finite element method*. Numer. Methods Partial Differential Equations **18**, 2002, p. 625–648.
[18] Ciarlet, P.G., The finite element method for elliptic problems, North-Holland, Amsterdam, 1978.
[19] Frey, P.J., Alauzet, F., *Anisotropic mesh adaptation for transient flows simulations*, in: Proc. 12th Internatinal Meshing Roundtable, Santa Fe, NM, USA, 2003.
[20] Borouchaki, H., George, P.-L., Mohammadi B., *Delaunay mesh generation governed by metric specifications*. Part II. Applications, Finite Elements in Analysis and Design 25, 1997, p.85–109
[21] Ku, J., Draney, M., Arko, F.R., Lee, W., Chan, F., Pelc, N., Zarins, C, Taylor, C.A., *In vivo validation of numerical prediction of blood flow in arterial bypass grafts*, Ann. Biomed. Eng. **30**, 2002, p.743-752.