Adaptive FE-simulation of cardiovascular flow

J. Müller\textsuperscript{a*}, O. Sahni\textsuperscript{a}, X. Li\textsuperscript{a}, K. E. Jansen\textsuperscript{a}, M. S. Shephard\textsuperscript{a}, C. A. Taylor\textsuperscript{b}

\textsuperscript{a}Scientific Computation Research Center, Rensselaer Polytechnic Institute, 110 8th Street, Troy 12180 NY, USA
\textsuperscript{b}James H. Clark Center, Room E350B, 318 Campus Drive, Stanford, CA 94305-5431, USA

In this study we present an adaptive anisotropic finite element method (FEM) that we have developed and we demonstrate how computational efficiency can be increased when applying the method to the simulation of blood flow in the cardiovascular system. We use the weak SUPG formulation for the transient 3D incompressible Navier-Stokes equations which we discretize by linear finite elements, both for the pressure and the velocity field. Due to the transient periodic nature of the flow in blood vessels an instantaneous mesh adaption in each time step appears less favorable in terms of computer memory and time consumption. Therefore, we use the averaged speed over a cardiac cycle as the underlying field from which we derive our error indicators, namely the Hessians. This quantity is further employed to generate a mesh metric field and mesh modification algorithms are used to anisotropically adapt the mesh according to the desired size field. We demonstrate the efficiency of the method by first applying it to pulsatile flow in a straight cylindrical pipe and then to a pig artery with a stenosis bypassed by a graft. The efficiency of the method is measured in terms of computational savings when we compute the wall shear stresses (WSS), a quantity identified to be important when trying to understand arterial disease. The gain factor in efficiency of our method can be quantified as one order of magnitude and therefore will allow for the simulation of more complicated systems without any loss of accuracy at the same or less computational expenses compared to traditional methods.

\textbf{Keywords:} Computational blood flow, Anisotropic finite element mesh adaptivity, Wall shear stress

1. INTRODUCTION

In recent years, computational techniques have been used increasingly by researchers seeking to understand vascular hemodynamics. Numerous analyses have been performed that range from simplistic two-dimensional models and equivalent circuit or hydraulic models (Začek and Krause, 1996) to more sophisticated three-dimensional models (Lee and Chen, 2002; Hoogstraten et al., 1996; Long et al., 2001). Of particular interest is the work of Perktold et al. (1991) and the work of Steinman et al. (1995). Recent trends suggest the use of more complex geometries in order to represent the anatomy more realistically (Oshima et al., 2001; Buchanan et al., 2003; Zhao et al., 2002; Taylor et al., 1998).

When it comes to the numerical simulation of the physiological phenomena the more complex the geometric models are the higher are the expenses in terms of memory requirement and computational time. Typically, arterial solid models are obtained by processing medical imaging data, such as MRI or CT data. This includes (i) extracting a set of points (a contour) approximating the inside boundary of a vessel on a series of 2D image slices, (ii) interpolating the contour with a curve, (iii) lofting a surface through the interpolated curves and (iv) joining the surfaces together to form a bounded volume. For details, see Taylor et al. (1998).

In order to employ the finite element (FE) method to simulate hemodynamics the model has to be subdivided into a finite number of elements.
The obstacles that have to be overcome when discretizing a domain are twofold. First, a mesh has to be generated that sufficiently represents the geometry of the blood vessel. Considering the fact that the vessel boundary is curved, using curved elements, too, would be desirable. In case only straight sided elements are available, one can assume that the finer the mesh is the better the curved geometry will be approximated.

The second issue which is independent of whether the surface geometry is adequately represented or not, is the question how good the mesh is capable of sufficiently resolving all the desired solution quantities, like the pressure field, the velocity and derivative quantities like the wall shear stresses. Again, one would assume that the finer the mesh is chosen, the better the solution features will be resolved. While this is certainly true when applying a consistent numerical method, like the FEM, the question remains which exact mesh density has to be selected to represent the desired solution field accurately enough.

Generally, one assumes the mesh density is adequate when further refinements of the mesh do not significantly change the numerical solution (“mesh independence”), (Zhao et al., 2002; Prakash and Ethier, 2001). However, for certain geometric models that evoke singular behavior of the solution field and especially its derivatives, as is the case for bifurcation points for instance, the numerical FE-solution will indefinitely improve however fine the mesh is, (Müller and Korvink, 2004). In that sense, some caution is appropriate when attempting to obtain a “mesh independent” solution. A much weaker claim, known as a good “convergence rate” of the solution as meshes become finer, would be more appropriate.

To accelerate the convergence behavior mesh adaptive techniques have been proposed and are used for a range of physical problems ranging from solid mechanics through electromagnetics, fluid mechanics and their coupling. Refining the mesh uniformly in order to achieve a better solution would mean to demand computational resources that are up to orders of magnitudes higher than using adaptive techniques (Müller and Korvink, 2004). In view of the complexity of blood vessel models, this is clearly a barrier and an adaptive approach becomes highly desirable when simulating hemodynamics. For a summary of different adaptive techniques, see Ainsworth and Oden (2000) or Verfürth (1996). Common to all the methods is the ability to reduce the computational expenses and at the same time control the accuracy of the solution that is obtained. For blood flow, only recently work has been published indicating significant FE simulation improvement when applying adaptive techniques (Prakash and Ethier, 2001). Key ingredients to such a method consist of the following:

- a posteriori error estimation/indication: estimating or obtaining an indication of the discretization error since the solution and therefore the error are a priori unknown,

- a mesh modification strategy where elements that are identified to have particular high or low (estimated) error are assigned to be modified. The most straightforward way would be to refine (or split) elements with high error and to coarsen elements with low error. Other techniques suggest to consider directional error information, too. Here, error reduction is achieved by stretching element sizes in directions where low error values prevail and by shrinking elements in the direction of high errors. In this study we will follow the latter method.

- The last ingredient are mesh modification techniques. A number of methods have been presented in the literature ranging from conforming splitting and coarsening of simplicial meshes in 2D and 3D, see Rivara (1984) and Bänsch (1991b); to complex mesh operations that were developed only recently and which allow for directional alignment in accordance with a predefined mesh metric fields in 2D (Almeida et al., 2000) and 3D (Li et al., 2003; Li and Shephard, 2003; Pain et al., 2001; Frey and Alauzet, 2003).

A posteriori error estimation is well understood for elliptic problems and to a smaller extent for
parabolic problems as well, it is still unsatisfactorily developed for the time dependent incompressible Navier-Stokes equations. Early results can be found in Bänsch (1991a) and a more recent survey on possible approaches is presented in Barth and Dekoninck (2003). Attempts to improve simulation efficiency of hemodynamics by means of mesh adaptivity up to now are restricted to steady flow (Prakash and Ethier, 2001).

While most of the work on error estimation for transient phenomena also try to incorporate the time discretization error, it seems less advisable to do so for flow that is pulsatile and therefore periodic in time. We propose a method in which high errors are identified by averaging the flow field over a cardiac cycle and then base the error analysis on that averaged field.

In order to be able to extract directional information of the error which in turn can be converted into a mesh metric field we use the Hessian strategy (Remacle et al., 2002), a method where the field's (in our case this will be the speed) second derivatives are used to obtain information on the error distribution. When linear finite elements are used, the interpolation error is equivalent to second order derivatives and there is strong evidence that a large amount of the discretization error is covered by this error indicator, (Fortin, 2000), although this technique is not an a posteriori error estimation in its strict definition. High eigen values of the field’s Hessian matrix defined at a particular node imply high errors and therefore the size of the elements surrounding that node in the direction of the corresponding eigen vector should be decreased. Conversely, low eigenvalues in a particular eigen direction suggest to elongate the elements.

In most hemodynamic studies, the primary quantity of physiological interest is the wall shear stress which contains derivatives of the velocity field (Cheng et al., 2002). It is hypothesized that low wall shear stress affects the localization of atherosclerotic plaques in regions of complex flow (Cheng et al., 2002). As is the case for most finite element methods that are based on a weak formulation, derivative field quantities are usually not as well resolved as the primary field itself. The theoretical nature of this phenomenon is the mathematical fact that while the primary field solution may be bounded everywhere in the simulation domain, its field derivatives may very well not be bounded at all, i.e., they might be of infinite value at certain singularity locations from the weak formulation point of view. In these cases only an appropriate adaptive method is capable to sufficiently recover the singular behavior imposed by the problem formulation (Reddy, 1993). Investigating wall shear stresses therefore not only poses an important challenge from a physiological viewpoint but also the nature of the numerical method at hand has to be properly enhanced by an efficient adaptive method to accurately resolve the fields where otherwise unreasonable values would tempt the analyst to draw the false conclusions.

The article is organized as follows. We provide a brief introduction into the numerical method that we use to solve the hemodynamic flow in the proceeding section. Then, we give an overview of the error indication employed together with the basic concepts of the mesh modification techniques. The third section encompasses an in depth study of wall shear stress computation employing our adaptive method. We first demonstrate the efficiency of our method by applying it to a steady flow case in a straight cylindrical pipe. Here, we compare the computed WSS with the analytical values. We then increase the complexity of the problem by extending it to pulsatile flow for the same geometric configuration. Finally, we apply the method to a physiological model, that is, a pig artery stenosis with a bypass graft. For the same model, previous studies showed a satisfactory agreement between experimental and numerical simulations (Ku et al., 2002).

In each of the cases, we compare the adaptive method to a non adaptive method. In the latter method the solution quality is attempted to be improved by using a series of different uniform meshes, meaning that element sizes (in terms of edge lengths) for a given mesh are roughly the same. It turns out that by applying our method huge savings in terms of computer time and memory can be made. Furthermore, we give evidence that more realistic and therefore more complex physiological models are only accessible for nu-
numerical simulation (assuming the computing devices at hand are reasonably powerful) when our method is applied.

2. SIMULATION METHODS

2.1. The Governing Equations

The governing equations for flow are the well known incompressible Navier-Stokes equations, see Jansen (1997, 1999) for details,

\[
\rho \frac{du^i}{dt} + \rho u^j u_{i,j} = -p_i + \tau_{i,j,j} + f_i
\]  

(1)

\[
u_{i,i} = 0.
\]  

(2)

The variables are the velocity \( u_i \), the pressure \( p \), the density \( \rho \), and the stress tensor \( \tau_{ij} \). For incompressible flow, indicated by equation (2), the divergence of the velocity is zero, causing the stress tensor \( \tau_{ij} \) to be simply the symmetric strain rate tensor. Hence

\[
\tau_{ij} = \rho (u_{i,j} + u_{j,i}).
\]  

(3)

Finally \( f \) is a source term, such as gravity or the force due to a surface tension. This term is typically neglected in arterial flow analysis.

To proceed with the finite element discretization of the weak form of the Navier-Stokes equations (1,2), we first introduce the discrete weight and solution spaces that are used. Let \( \Omega \subset \mathbb{R}^N \) represent the closure of the physical spatial domain (i.e. \( \Omega \cup \Gamma \) where \( \Gamma \) is the boundary) in \( N \) dimensions; only \( N = 3 \) is considered here. The boundary is decomposed into portions with natural boundary conditions, \( \Gamma_h \), and essential boundary conditions, \( \Gamma_g \), i.e., \( \Gamma = \Gamma_g \cup \Gamma_h \). In addition, \( H^1(\Omega) \) represents the usual Sobolev space of functions with square-integrable values and derivatives on \( \Omega \). Subsequently \( \Omega \) is discretized into \( n_{el} \) finite elements, \( \Omega_e \). With this, we can define the discrete trial solution space for the velocity

\[
S_h = \{ v|v(\cdot, t) \in H^1(\Omega)^N, t \in [0,T], \}

(4)

\[
v|_{x \in \Omega_e} \in P_k(\Omega_e)^N, v(\cdot, t) = g \text{ on } \Gamma_g \},
\]  

and weight space for the semi-discrete formulation

\[
W_h = \{ w|w(\cdot, t) \in H^1(\Omega)^N, t \in [0,T], \}

(5)

\[
w|_{x \in \Omega_e} \in P_k(\Omega_e)^N, w(\cdot, t) = 0 \text{ on } \Gamma_g \}.
\]

The pressure space is given by

\[
P_h = \{ p|p(\cdot, t) \in H^1(\Omega), t \in [0,T], \}

(6)

\[
p|_{x \in \Omega_e} \in P_k(\Omega_e) \},
\]  

where \( P_k(\Omega_e) \) is the space of all polynomials defined on \( \Omega_e \), complete to order \( k \geq 1 \). 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 that described by Taylor et al. (1998). Given the spaces defined above, we can state the semidiscrete Galerkin finite element formulation applied to the weak form of (1) as: Find \( u \in S_h \) and \( p \in P_h \) such that

\[
B(w, q; u, p) = 0
\]  

(7)

where

\[
B(w, q; u, p) = \int_\Omega \{ w_i (u_i + u_j u_{i,j} - f_i)
\]

\[
+ w_{i,j} (-p \delta_{ij} + \tau_{ij}) - q_{i} u_{i} \} dx
\]

\[
+ \int_{\Gamma} \{ w_i (p \delta_{im} - \tau_{im}) + q u_{i} \} ds
\]  

(8)

for all \( w \in W_h \) and \( q \in P_h \). 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. We add stabilization terms (Franca and Frey, 1992), that allow us to choose the same local approximation space, \( P_k(\Omega_e) \), for both the velocity and pressure variables.

To develop a discrete system of 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 (typically piecewise polynomials; all calculations described herein were performed with a linear basis). Since we have a non-linear, time dependent system of equations, Gauss quadrature of the spatial integrals results in a system of first-order, non-linear ordinary differential equations.

Finally this system of non-linear ordinary differential equations is discretized in time via a generalized-alpha time integrator (Jansen et al., 1999) 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 which in turn is solved (at each
time step) and the solution is updated for each of the Newton iterations. The linear algebra solver of Shakib (1999) is used to solve the linear system of equations.

2.2. Error indication

The numerical solution that we obtain through the method described above is basically subject to two kind of errors: first, insufficiency in the approximation of the geometric model, and second, the discretization error itself. To render the second type more precisely, consider a model that can be perfectly represented by any given mesh, e.g. a straight sided quadrilateral. Then, even in this case, numerical FE solutions generally carry errors that exclusively depend upon the coarseness of the mesh.

This section deals with identifying the latter. It is well known, that a function which is sufficiently smooth can be developed into a Taylor series. When trying to interpolate that function with a piecewise linear function the error will always be of the order of the second derivative of the function. Therefore, if we assume that the “true” finite element solution (meaning the one that is defined through the weak form) is smooth enough and assuming that our finite element solution is a linear interpolation to that (which is not always the case, however) a good error measure would be to calculate the second derivatives to the discrete solution.

The discrete solution is piecewise linear on each element, thus the first and the second derivatives for each node $i$ have to be reconstructed. The reconstruction that we employ uses the derivatives of the patch $S_i$ of all elements $T$ surrounding node $i$. For example, the recovered gradient of a given field component $u$ on node $i$ can be computed as

$$\text{grad}_i(u) = \frac{1}{\text{Vol}(S_i)} \sum_{T \in S_i} \text{Vol}(T) \nabla u|_T.$$ (9)

The same procedure applied to the second field derivatives yields the Hessian matrix. We should note that values that are reconstructed on the mesh boundary in this way usually do not very well represent the “real” Hessians, therefore an extrapolation technique is used to project interior values onto the nodes that lie on the boundary.

A mesh metric field then is obtained by calculating the eigen space of the Hessian matrix. A high eigen value in a particular eigen direction for a given node indicates that the lengths of element edges in the vicinity of that node (aligned with that direction) would have to be decreased to reduce the error. Conversely, a low eigenvalue would suggest to increase edge lengths since the solution in that direction is already sufficiently resolved.

Truncation values $h_{\text{min}}$ and $h_{\text{max}}$ for the eigen values have to be specified to reflect the fact that infinitely large eigenvalues would result in infinitely small mesh sizes and zero eigenvalues would yield infinitely long elements. The modified eigen values of the Hessian matrix then read

$$\tilde{\lambda}_i = \min\left(\max\left(c |\lambda_i|, \frac{1}{h_{\text{max}}^2}, \frac{1}{h_{\text{min}}^2}\right), 1\right)$$ (10)

where $c$ is a constant reflecting the overall interpolation error. The mesh metric field $M$ is obtained by multiplying the matrix of the modified eigen values with the matrix $R$ of eigen vectors

$$M = R\tilde{\Lambda}R^{-1}, \quad \tilde{\Lambda} = \begin{pmatrix} \tilde{\lambda}_1 & 0 & 0 \\ 0 & \tilde{\lambda}_2 & 0 \\ 0 & 0 & \tilde{\lambda}_3 \end{pmatrix}.$$ (11)

The goal then is for each edge $e$ of the mesh, to assume unit value in the space where distances are measured by the metric $M$. Expressed in terms of a quadratic form, this is equivalent to

$$(e, M_e e) = 1$$ (12)

where $M_e$ is the mesh metric field of the node adjacent to the given edge. We also refer to this operation as the measuring of sizes in the transformed space.

2.3. Mesh modifications

Given the mesh metric field $M$ defined over the domain, the ideal goal is to apply local mesh modification operators to yield a mesh with all edges satisfying equation (12). However, the fact that we cannot pack unit equilateral tetrahedra to satisfy a constant unit mesh metric field indicates that a perfect match to equation (12) is impossible. Therefore, we relax that criteria and
consider a mesh satisfying the metric field if, in the transformed space, (i) all its measuring edge lengths fall into an interval close to one, referred to as \([L_{\text{low}}, L_{\text{up}}]\), and (ii) no sliver element exists. A restriction, \(L_{\text{low}} \leq 0.5L_{\text{up}}\), in selecting the interval is applied. This is to ensure that the measuring size of two new edges created by a bisection will not be shorter than \(L_{\text{low}}\) so that oscillation between subdivision and coarsening is prevented. The overall mesh modification procedure consists of (Li and Shephard, 2003):

- Incrementally reduce the maximal measuring edge length in the transformed space until below \(L_{\text{up}}\) by marking a set of longest mesh edges and subdividing all elements bounding these edges in each iteration;
- Collapse as much mesh edges shorter than \(L_{\text{low}}\) (in the transformed space) as possible to coarsen or fix up the mesh;
- Eliminate sliver elements with respect to the mesh size field to improve connectivity.

The subsections that follow illustrate the concepts of the major mesh modification procedures in two dimensions. Similar rules hold for the slightly more complex 3D case.

2.3.1. Subdivision of elements

The first mesh modification operator is the element subdivision. Given a set of edges to be refined, it consists in modifying a cavity \(C\) (or cavities) composed of elements neighboring these edges by replacing these elements with nested and subdivided ones, see Figure 1.

![Figure 1. Element subdivision in 2D.](image1)

2.3.2. Edge collapsing

The second mesh modification operator is edge collapsing. If an edge is too short (< \(L_{\text{low}}\)) we decide to remove it. The operator modifies a cavity \(C\) composed of all the triangles (tetrahedra in 3D) neighboring one end vertex of the short edge (see Figure 2). It can be seen as pulling one end vertex of the edge to another.

![Figure 2. Edge collapsing in 2D.](image2)

2.3.3. Edge swapping

This operation consists in modifying a cavity \(C\) composed of two neighboring elements \(e_1\) and \(e_2\) by swapping the edge and replacing \(e_1\) and \(e_2\) by \(e_3\) and \(e_4\) as represented in Figure 3.

![Figure 3. Edge swapping in 2D. An edge separating two triangles is replaced by the opposite edge.](image3)

2.3.4. Projecting to the model boundary

Once the mesh size field requires that a new node be introduced, by subdividing an element that is on the mesh boundary, the algorithm takes care that the new node lies on the model boundary as well (Li et al., 2003). At the same time we avoid nodes being removed from the mesh when they lie on the boundary, see Figure 4. Hereby, each time a modification procedure is invoked, we assure that the approximation of the model by the mesh only can improve.

![Figure 4. Projecting to the model boundary.](image4)
2.3.5. Easy solution transfer

We use a callback mechanism which is invoked each time a mesh modification takes place. That is, once an element is split by dividing its edges a list of elements of the old mesh as well as elements of the new mesh are handed in to the callback function which then accesses the solution values attached to the old mesh and transfers these values according to a user defined interpolation rule onto the new mesh, see Figure 5. This is of particular importance when the mesh is modified during a time stepping sequence within the solving process.

\[ t_i = -p \delta_{ij} + \tau_{ij} n_j \]  

(13)

where \( n_j \) are the components of the normal \( n \) to the surface. The WSS magnitude then is defined, on each point on the surface, as

\[ t_w = |t_w| = |t - (t \cdot n) \cdot n|, \]  

(14)

that is, the magnitude of the traction vector’s component that is tangential to the surface.

3. SIMULATION RESULTS

To demonstrate the efficiency of our method we first apply it to stationary flow in a cylindrical straight pipe. This example suits well to explicitly quantify the major improvements that we can achieve with our method. The symmetry of the geometry and therefore the flow profile allow to easily measure computational savings while comparing numerical solutions obtained on different meshes to an analytical value. The quantity of interest is the wall shear stress, defined on any point on the domain boundary. WSS has proven to be an important physiological quantity since low values are presumed to be related to arteriosclerotic disease (Cheng et al., 2002). The same authors define the WSS via the surface traction vector \( t \) whose components are given as

\[ \tau_{ij} = \nu \begin{pmatrix} 0 & 0 & w_x \\ 0 & 0 & w_y \\ w_x & w_y & 0 \end{pmatrix}. \]  

(15)
Then, since the $w$-velocity profile is quadratic in each plane perpendicular to the cylinder axis, its derivatives are linear, i.e., $w_x = c \, x$, $w_y = c \, y$. Thus, the wall shear stress is assuming a constant value of $t_w = |c \, \nu|$ all over the cylinder wall. In our example, we have set the velocity inflow profile such that values of $c = 2.0$ and $\nu = 1.0 \times 10^{-2}$ are assumed.

We try to numerically approximate that value by two different approaches. First, we obtain results on a series of successively refined uniform meshes, hereby reflecting the fact that the values should become the more accurate the finer the mesh is. Second, we try to more efficiently calculate the WSS by applying error estimation/indication and anisotropic mesh adaption. Therefore we compute the flow field on the coarsest uniform mesh (3021 nodes), then reconstruct the second derivatives of the speed field, which in turn is transformed into a mesh metric field according to (11). Then, the metric field is fed into the mesh modification tool box which, after a number of iteration cycles produces a highly anisotropic mesh. Fig. 7 shows the surfaces of three uniform meshes and an anisotropically refined mesh according to the method described in section 2.

Table 1 shows the mean values and the standard deviation for the WSS which is interpolated on the surface mesh along a cut plane perpendicular to the cylinder’s $z$-axis, as indicated in Fig. 6. The mean values together with the error margins $\sigma$ coincide with the analytically expected value in all four cases. Fluctuations of the WSS obtained on the uniform meshes are the smaller the finer the mesh is. We observe that we achieve almost the same degree of accuracy in terms of the standard deviation when using an anisotropically adapted mesh of 1258 vertices as compared to a uniform mesh of 18184 vertices. Figure 8 shows the spatial distribution of the WSS along the circumference for a constant $z$-value.

<table>
<thead>
<tr>
<th>Mesh type</th>
<th># nodes</th>
<th>mean WSS</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>uniform</td>
<td>3021</td>
<td>1.968E-2</td>
<td>1.360E-3</td>
</tr>
<tr>
<td>uniform</td>
<td>6939</td>
<td>1.969E-2</td>
<td>0.623E-3</td>
</tr>
<tr>
<td>uniform</td>
<td>18184</td>
<td>1.979E-2</td>
<td>0.272E-3</td>
</tr>
<tr>
<td>anisotropic</td>
<td>1258</td>
<td>1.986E-2</td>
<td>0.380E-3</td>
</tr>
</tbody>
</table>

We try to further quantify the amount of savings that we gain through the adaptive method. We could claim, by considering the standard deviation as a measure, a factor that is between 5.5 and 14.5 when measuring computational expenses in terms of degrees of freedom.

We should note that this gain factor has to be reduced when taking into account that first an initial solution has to be calculated on the coarsest uniform mesh. On the other hand, we also would like to point out that the gain factors given above hold for the best case scenario where the flow solver scales linearly with the number of degrees of freedom. In practice, the computer time versus the number of nodes increases by a higher order than linear and our gain factor becomes even larger.
3.1.2. Pulsatile flow

An improvement towards a more physiologically realistic flow situation is to study pulsatile flow in a straight cylindrical pipe. Here, we use the same model that we have used for the stationary case. The inflow boundary condition is assumed to be a Womersley profile (Womersley, 1955), as depicted in Fig. 9 for a point near the center of the inlet boundary disk. Again, we apply zero (no-slip) velocity conditions on the cylinder wall and zero natural pressure on the outlet. Similarly to the stationary case, the only non-zero velocity component is the $z$-component. This component varies periodically in time at each location within the domain, reflecting the pulsatile nature of the flow. We analyze the WSS both at a given instant during a time cycle and temporally averaged over the period. The latter is also known as the wall shear stress magnitude, defined as

$$ t_{mag} = \frac{1}{T} \int_0^T |t_w| dt $$

(16)

where $T$ is the time period of the cycle. $t_{mag}$ is discretely approximated in time by the sum over the values at each time step.

As in the stationary case we compare results obtained by a series of uniformly refined meshes (as shown in Figure 7) to the results obtained on an anisotropic mesh. Here, in contrast to the time-independent flow, it is not advisable to use the speed field at an arbitrary instant during the cycle to compute the second field derivatives. Instead, we use the time averaged speed field to compute the Hessian values. As can be expected, the mesh after adaption looks similar to the one depicted in Fig. 7. Table 2 lists the time averaged $t_{mag}$ at a location of constant $z$. In this exam-

<table>
<thead>
<tr>
<th>Mesh type</th>
<th># nodes</th>
<th>mean $t_{mag}$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>uniform</td>
<td>3021</td>
<td>5.030</td>
<td>1.349</td>
</tr>
<tr>
<td>uniform</td>
<td>6939</td>
<td>3.293</td>
<td>0.527</td>
</tr>
<tr>
<td>uniform</td>
<td>18184</td>
<td>2.963</td>
<td>0.177</td>
</tr>
<tr>
<td>anisotropic</td>
<td>1218</td>
<td>3.061</td>
<td>0.366</td>
</tr>
</tbody>
</table>

Table 2

WSS magnitude ($t_{mag}$) mean values and standard deviation $\sigma$ for pulsatile flow in the cylindrical pipe.
Figure 9. Velocity profile at a point near the center of the inflow face for a cycle.

We further illustrate this by presenting the time dependent behavior of the WSS at a specific point on the wall in Fig. 10. While the shape of the curves follow the same pattern within a cycle, the magnitude is far too high for the coarser meshes. Assuming that the solution fields have converged sufficiently on the finest uniform mesh we can state that the anisotropic mesh is capable to adequately resolve the time dependent behavior of the fields, too.

3.2. Pig aorta

As the last but practically most relevant application we study the performance of our method by applying it to the simulation of pulsatile flow in a pig aorta with a stenosis and a bypass graft. The dimensions of the model are approximately 10cm in length while the vessels diameter at the inflow is approximately 1.6cm. Blood was modeled as an incompressible Newtonian fluid with a constant density of 1.06g/cm³ and a constant viscosity of 0.04dyn s/cm². The geometric model is shown in Fig. 11. The inlet flow is prescribed as pulsatile Womersley flow profile that is based upon PC-MRI through-plane flow rate data, see Ku et al. (2002). Figure 12 shows the velocity at an instant during the cardiac cycle and the inset depicts the velocity profile at a point near the center of the inflow plane for one cardiac cycle (together with the instant the velocity snap-
Figure 13. Three successively refined uniform meshes and one anisotropically adapted mesh.

shot is taken at). Similar to the straight pipe cases we obtain simulation results on five different meshes, four of them being uniform and one being an adaptively refined mesh. Again, 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 7 shows four of the meshes used in this simulation together with their total number of vertices. We have cut the adapted mesh open along a clip plane parallel to the flow direction of the main vessel in order to further illustrate the effects of the mesh modification procedure, see Fig. 14. We observe that slender and elongated elements were created in regions where flow speed and also second derivatives are highest, such as near the the vessel walls (boundary layer effect). This becomes more obvious when comparing Figs. 14 and the flow field at an instant, see12. Other regions where major mesh adaption takes place (see the zooms in Fig. 14) are the stenosis area, re-entrant corners of the model and in the main artery where the re-directed flow impacts on the vessel wall. Typically, element sizes are predominately small but isotropic in the re-entrant corner, reflecting the well known fact that solution behavior there tends to be singular. Small isotropic elements also dominate the stenosis area, accounting for strongly varying flow in that region; for exam-

Figure 12. Flow profile during a cycle (inset) and velocity field at an instant. Bright shades indicate high speed.
ple, back-flow occurs within a particular interval during the cycle.

We analyze the transient behavior of the WSS at two particular points that are located in distinguished regions of the vessel boundary. The first one is located directly in the stenosis part (labelled A) whereas the second is located near the downstream vessel branching and is labelled B (both depicted in Fig. 11). The WSS are computed on a series of uniform meshes ranging from 3456 nodes to over 800K nodes and on an adapted mesh consisting of 42196 nodes.

While the WSS values near the stenosis are much higher than at the other location, we observe that in both cases the values heavily depend on the number of degrees of freedom that are used for their computation, see Figs. 15 and 16. On the coarser meshes the calculated WSS generally is too high, whereas the values tend to converge for the larger meshes. Even though the convergence is not uniform during the cardiac cycle we do observe convergence patterns that are similar to that of the pulsatile flow in the straight pipe case. For location A, see Fig. 15, assuming that the finest uniform mesh of over 800K nodes (corresponding to over 4 million tetrahedra) sufficiently resolves all the flow features including the derivative quantities, we observe that the anisotropically refined mesh follows the pattern of the finest uniform mesh well in the first half of the cycle but is of lower magnitude after the peak in the cycle’s second half. As of now, we are not able to determine whether this implies even a further improved approximation compared to the solution obtained on the uniform mesh or an insufficient approximation.

The situation is slightly different for location B, see Fig. 16. The convergence behavior can be very well approved for the uniform meshes, i.e. the mesh consisting of around 200K nodes and the mesh consisting of 800K nodes yield roughly the same results all over the cycle. The values of the anisotropically adapted mesh follow those of the finest uniform mesh for the most part. The anisotropic mesh even seems to catch the WSS pattern better than the 200k 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 gainings 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. And, they are not too far away from those obtained on meshes of either 200K or 800K nodes. Therefore, as a conclusion, claiming a factor of one order of magnitude in computational savings in this case study, too, would not be an exaggeration.

Figure 14. Clip plane through mesh of pig aorta.

4. Summary and Discussion

We have presented a method by which hemodynamic FE-simulations can be enhanced. 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
Figure 15. WSS during a cycle at a point in cut-plane A.

Figure 16. WSS during a cycle at a point in cut-plane B.

a more accurate solution at considerably reduced computational costs.

This approach is novel in computational hemodynamics where it is generally assumed that the solution field the analyst is interested in is accurate enough 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 branching 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. In pursuing this traditional path, a huge amount of computational resources are wasted.

We have demonstrated how computational resources can be saved and at the same time numerical accuracy is increased. We have compared the quality of the solution that is obtained when employing our adaptive method to the solutions obtained on a series of uniformly refined meshes.

For a straight pipe, we first study how the computed WSS converges pointwise to the value that can be expected from the analytical theory. Then, we study the convergence behavior for the same field and the same geometry when pulsatile inlet boundary conditions are imposed. Here, we also investigate the transient behavior of the WSS at arbitrary locations on the domain boundary. In both cases, our method proves to be highly efficient and a gain factor in terms of degrees of freedom of approximately one order of magnitude can be achieved.

Accordingly we try to transfer the results to the simulation of a more realistic physiologic model, namely a pig aorta with a bypassed stenosis. We aim at obtaining pointwise convergence of the WSS at two arbitrarily chosen locations on the domain boundary. The first being right in the stenosis, another near the vessel branching. For these two locations we compute the transient behavior of the WSS. The complex nature of the flow in this model requests more flexibility when it comes to the quantification of the gain factors that we are able to achieve. While that may not be of one order of a magnitude at each location in the spatial and temporal simulation domain, we still can observe significant savings when we...
employ the anisotropic adaptivity.

Future work will incorporate a more stringent error analysis to further accelerate the efficiency of the adaptive method. We will also explore the possibility to include compliant vessel walls into our FE-model. Eventually, we will be able to extend the simulation to larger and therefore more realistic hemodynamic systems.

ACKNOWLEDGEMENTS

We greatly acknowledge the support of this work by NSF grant ACI-0205741.

References


