INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2001; **51**:985–1008

From medical images to anatomically accurate finite element grids

Juan Raul Cebral[†] and Rainald Löhner[*,‡]

School of Computational Sciences, George Mason University, M.S. 4C7, 4400 University Drive, Fairfax, VA 22030, U.S.A.

SUMMARY

The successful application of computational modelling of blood flow for the planning of surgical and interventional procedures to treat cardiovascular diseases strongly depends on the rapid construction of anatomical models. The large individual variability of the human vasculature and the strong dependence of blood flow characteristics on the vessel geometry require modelling on a patient-specific basis. Various image processing and geometrical modelling techniques are integrated for the rapid construction of geometrical surface models of arteries starting from medical images. These discretely defined surfaces are then used to generate anatomically accurate finite element grids for hemodynamic simulations. The proposed methodology operates directly in 3D and consists of three stages. In the first stage, the images are filtered to reduce noise and segmented using a region-growing algorithm in order to obtain a properly defined boundary of the arterial lumen walls. In the second stage, a surface triangulation representing the vessel walls is generated using a direct tessellation of the boundary voxels. This surface is then smoothed and the quality of the resulting triangulation is improved. Finally, in the third stage, the triangulation is subdivided into so-called discrete surface patches for surface gridding, the desired element size distribution is defined and the finite element grid generated. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: grid generation; hemodynamics; medical images; finite element modelling

1. INTRODUCTION

Detailed knowledge of the blood flow in arteries and its relationship to the geometry of the vessels is required to understand the role of hemodynamics in the development of arterial diseases [1, 2] and to design, optimize and plan surgical procedures to treat these diseases [3]. Computer simulations can help to design less invasive and more precise procedures, to advance the quality and cost-effectiveness of the therapy and to provide better training for physicians.

*Correspondence to: Rainald Löhner, M.S. 4C7, School of Computational Sciences, George Mason University, Fairfax, VA 22030-4444, U.S.A.
[†]E-mail: jcebral@gmu.edu
[‡]E-mail: rlohner@gmu.edu

Received 26 November 1999
Revised 6 October 2000

Even though experimental research has made significant advances to the understanding of vascular diseases and long-term consequences of surgical interventions, it suffers from several limitations. Experimental flow model studies are expensive and time consuming, working with fluids with the same rheological properties of blood is difficult, and the level of detail that can be extracted from an experimental investigation is usually limited to a handful of pressure and velocity profile measurements [4, 5]. Recent advances in the cardiovascular imaging technology have enabled not only to obtain *in vivo* anatomical data but also physiologic measurements from different modalities such as Doppler ultrasound and magnetic resonance imaging (MRI). Phase-contrast MRI techniques allow to measure 3D time-dependent velocity fields [6]. However, these measurements are usually limited to a few planes perpendicular to the vessel axis, and do not provide other flow quantities of interest such as pressure distributions, wall shear stress and particle residence times. In many instances, only the velocity component normal to the sampling plane is imaged in order to reduce the scanning time and the discomfort of the patient.

The analytical solution of the partial differential equations (PDEs) governing the flow of blood in arteries is limited by the fact that only a few closed-form solutions can be found [7]. They typically involve extremely simple geometries such as straight tubes, simplified shear stress models and simple boundary conditions. Certainly, analytical solutions cannot be found for curved vessels with bifurcations where flows are extremely complex compared to those in unbranched straight segments.

Computational fluid dynamics (CFD) techniques have been increasingly used over the past few years in order to gain insight into the arterial hemodynamics [8–10]. Numerical simulations deliver detailed information over the complete flow field, including scalar, vector and tensor quantities such as pressure, velocity, shear stress and particle residence times. The ability to quickly change simulation and model parameters makes the CFD simulation methodology very useful for surgical and interventional planning.

Numerous two-dimensional hemodynamic calculations have been reported. However, these are highly inadequate for arterial blood flow. Many three-dimensional, transient calculations have been performed, most of them restricted to idealized geometries [8–14]. The goal of these studies is to gain basic understanding of the role of hemodynamics in the genesis and development of arterial diseases. However, the objective of the present paper is to model *patient-specific* blood flow in order to plan and evaluate different surgical and interventional procedures. For these purposes, it is mandatory to adopt anatomically realistic geometrical models on an individual basis, since individual variability of arterial geometry is the rule rather than the exception, and blood flow characteristics strongly depend on the vessel geometry.

Previous efforts towards the construction of realistic geometrical models suitable for numerical calculations have concentrated on surface reconstruction from either digitized experimental models or from medical images. The first strategy requires the construction of anatomically accurate experimental models, followed by optical digitalization and surface reconstruction from unorganized data points [9, 15], an expensive and time-consuming procedure. A variety of approaches have been taken for solving the problem of reconstructing the surfaces of the vessels from medical images:

Contour detection: Each 2D slice is independently segmented using either grey-scale thresholding [16] or edge detection schemes [17]. The lumen contours are then smoothed, stacked and interpolated between slices [10, 18–21]. Although there are many contour and edge detection image-processing operators [17, 19], there is still no algorithm that can automatically

extract region boundaries perfectly from medical images. These methods are well suited for cases in which the arterial axis is nearly perpendicular to the image slices. However, in more complex cases such as the brain vasculature where many branches emanate in different directions, the errors associated to the interpolation and smoothing may become very large.

Iso-surface method: In these straightforward 3D methods, an iso-surface of intensity is directly extracted from the images [22]. High-intensity non-vascular tissue may be eliminated using connectivity criteria. The major drawback is that the iso-intensity level must be chosen by trial-and-error, and that a single iso-intensity value may not be valid for the entire vessel tree.

Region growing segmentation: These techniques classify all the points in a 3D image as inside or outside of the vessels, based on the topological connectivity of the image segments [17, 23]. Similar approaches have recently been applied to the automatic segmentation of the cerebral cortex from MRI data with great success [24]. A problem associated to this class of methods is that for small vessels, relatively small imperfections in the image may result in relatively large discontinuities in the vascular region, due to an incomplete interconnectivity.

Deformable models: These methods are based not only on the image intensity but also on *a priori* knowledge of the shape of the vessels. Adaptive deformable models have been applied to the segmentation of vessels and aneurysms from MRA and CTA [25–27]. In this case, an initial shape is iteratively deformed by solving an equivalent elastic problem driven by the image intensity gradient. The major drawback is that the initial shape must have the same topology as the reconstructed boundary and that in many instances it requires substantial human intervention.

The present paper describes the integration of various image processing and geometrical modelling techniques in order to quickly generate geometrical models (surface triangulations) of arteries starting from medical images. These discretely defined surfaces are then used to generate anatomically accurate finite element grids for hemodynamic simulations via the advancing front technique. The proposed methodology consists of three stages: image processing, geometrical modelling and grid generation. The aim of the first stage (image processing) is to obtain a properly defined boundary of the arterial lumen walls in the region of interest. In the second stage, the tessellation or iso-surface obtained from the first stage is made topologically consistent, smoothed, and the quality of the resulting triangulation is improved. Finally, in the third stage, the triangulation is subdivided into so-called discrete surface patches for surface gridding, the desired element size distribution is defined and the grid generated.

2. IMAGE PROCESSING

High contrast medical images such as computed tomography arteriography (CTA) and magnetic resonance angiography (MRA) are obtained by injecting a contrast agent in order to enhance the contrast between the artery and the surrounding tissues. The volumetric images consist in a series of m equally spaced 2D slices of $n \times n$ pixels. Either 1 byte (8 bits per pixel) or 2 bytes (12 or 16 bits per pixel) are used to specify the intensity or grey level of each pixel. This series of slices are combined to form a 3D image composed of $n \times n \times m$ voxels. Note that, in general, the voxels are not cubes since the distance between slices may not be the same as the pixel size in each 2D image. The intensity value $I(\mathbf{x})$ of a voxel located at position $\mathbf{x}$ is first converted to floating point numbers in the interval [0,1]. Then,

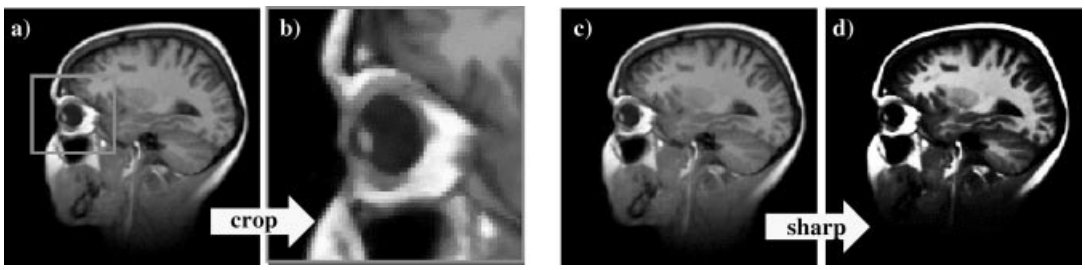

Figure 1. Application of the crop and sharpening operators: (a) definition of the region of interest; (b) cropped image; (c) original image; (d) sharpened image.

several image processing operators are applied in turn in order to reduce noise present in the image, enhance the contrast, and segment the arteries:

Crop: The cropping operator allows the definition of a region of interest R (a 3D box). The voxels inside the volume of interest are kept, while those outside are removed. Figures 1(a) and 1(b) show an example of the application of the crop operator to an MRI of the head. Mathematically, this operator can be represented as

$$I_0(\mathbf{x}) = \begin{cases} I(\mathbf{x}) & \text{if } \mathbf{x} \in R \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Contrast (*Sharpening*): The contrast filter enhances the intensity difference between the lighter and darker voxels of the image. Defining $\theta = [I_0(\mathbf{x}) - 1/2]\pi$ the contrast operator can be written as

$$I_1(\mathbf{x}) = \begin{cases} 1 & \text{if } I^*(\mathbf{x}) > 1 \\ 0 & \text{if } I^*(\mathbf{x}) < 0, \text{ where } I^*(\mathbf{x}) = I_0(\mathbf{x}) + \frac{1}{2}[\sin(\theta + 1) - I_0(\mathbf{x})] \\ I^*(\mathbf{x}) & \text{otherwise} \end{cases} \tag{2}$$

Region growing: This algorithm generates a connected region that represents the desired blood vessel. Staring with a seed voxel $\mathbf{x_0}$ (selected interactively), an advancing front-like procedure is used to mark all the voxels topologically connected to the seed and with intensity in the same range R_I as the seed, as illustrated in Figures 2(a) and 2(b). Denoting $N_0(\mathbf{x})$ the six closest neighbours of $\mathbf{x}$, the region growing algorithm can be summarized as

Initialize $I_2(\mathbf{x}) = 0 \ \forall \mathbf{x}$
Insert $\mathbf{x_0}$ into a Queue (the front)
While (Queue not empty)
  Extract $\mathbf{x}$ from Queue
  For ($\mathbf{x}' \in N_0(\mathbf{x})$) If ($I_1(\mathbf{x}') \in R_I$) Then insert $\mathbf{x}'$ in Queue
  Set $I_2(\mathbf{x}) = I_1(\mathbf{x})$
End While

The successful application of the region growing segmentation depends on the proper specification of the seed voxel $\mathbf{x_0}$ and the definition of the intensity range R_I. The intensity range R_I can be defined in a variety of ways. The simplest intensity range definition used in this

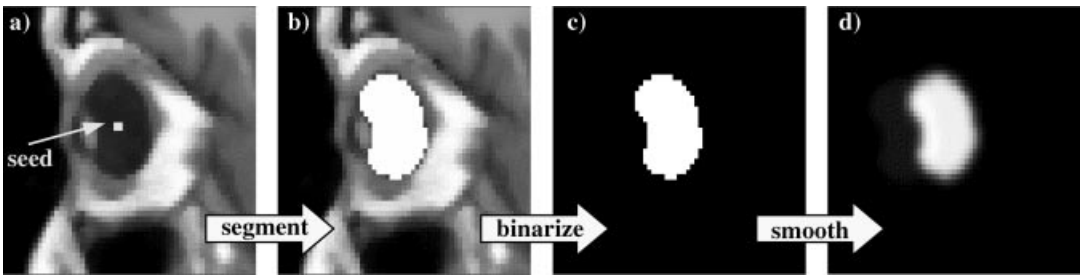

Figure 2. Region growing, binarization and blurring operators: (a) selection of the seed for the region growing scheme; (b) segmented image; (c) binarized image; (d) blurred image.

work is $R_I = [I_{\min}, I_{\max}]$ where $I_{\min}$ and $I_{\max}$ should enclose the intensity of all the voxels in the desired artery. An estimation of these parameters cannot be done *a priori*, it must be done on a trial and error basis for each particular image. This estimation is typically done as follows. First, select a low intensity voxel inside the desired artery as the seed. Then, define $I_{\min} = \alpha I(\mathbf{x_0})$ and $I_{\max} = \beta I(\mathbf{x_0})$ where α is usually 0.9, i.e. only a 10 per cent change in the intensity below the intensity of the seed is allowed. Since the high-intensity voxels usually represent vessels in MRA or CTA images, $I_{\max}$ is defined as 1.0. Thus, all voxels with intensity above $I_{\min}$ are kept. Other alternatives that attempt to automate the definition of the intensity range are still under investigation.

Binarization: A bilevel image is produced by clamping all pixels in a specified intensity interval to 1 (white) and the rest to 0 (black):

$$I_3(\mathbf{x}) = \begin{cases} 1 & \text{if } I_2(\mathbf{x}) \in [I_{\min}, I_{\max}] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

After the binarization operator is applied, white voxels represent the interior of the vessels and black voxels the outside, as illustrated in Figure 2(c).

Image smoothing (*Blur*): Image smoothing is performed with a linear convolution operator:

$$I_4(\mathbf{x}) = \sum_{x' \in N_1(x)} w(\mathbf{x}') I_3(\mathbf{x}') \tag{4}$$

where $w(\mathbf{x}')$ are the weights assigned to each of the neighbours $\mathbf{x}'$ of $\mathbf{x}$ contained in $N_1(\mathbf{x})$, a $3 \times 3 \times 3$ window centered at $\mathbf{x}$. The six nearest neighbours of $\mathbf{x}$ receive a weight of 2 and the rest of the neighbours in the $3 \times 3 \times 3$ window receive a weight of 1. The weight for the voxel at $\mathbf{x}$ is -26. The application of the blurring operator is exemplified in Figure 2(d).

Statistical filter: The aim of this operator is to eliminate inconsistencies in the labelling (black or white) of neighbouring voxels. These inconsistencies may arise due to noise in the original images. For instance, suppose white voxels represent the inside of an artery and black voxels the outside. Then, isolated black voxels may appear inside the white artery, completely surrounded by white voxels.

The statistical filtering proceeds as follows. First, black voxels with more that 3 white nearest neighbours are painted white (added to the artery). Second, white voxels with less than 3 white nearest neighbours are painted black (removed from the artery). Finally, black nearest neighbours of white voxels are painted white if they are also nearest neighbours of

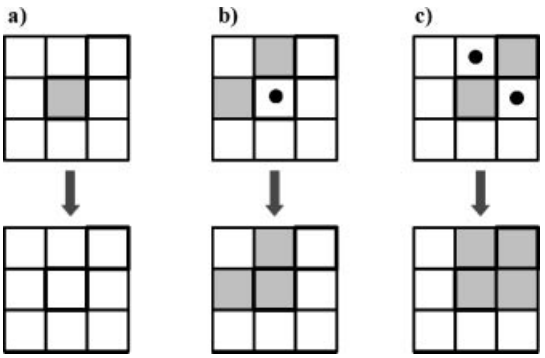

Figure 3. Statistical filtering operations: (a) step 1; (b) step 2; (c) step 3.

another white voxel. These three operations are schematically represented in Figure 3. Note that it makes sense to apply these statistical filters only after region growing and binarization.

Voxel painting: The interactive painting of voxels can be useful to get rid of some details present in the original image which are not to be taken into account in the CFD simulation, to cut arterial branches at desired places, to eliminate inconsistencies not removed by the statistical filters, or to add new features to the anatomy. For instance, fictitious aneurysms, stenosis or bypass grafts can be included by painting voxels in three dimensions. Obviously, this operation requires user intervention and prevents automation. Fortunately, in most of the cases the reconstruction of the anatomical geometry does not require voxel painting.

3. GEOMETRY MODELLING

Once the medical images have been processed as described above, the geometry of the vascular vessels is reconstructed by applying the following geometry modelling techniques:

3.1. Tessellation

The reconstruction of the surface triangulation representing the geometry of the desired vessels from the segmented image is done using two alternative approaches:

Iso-surfacing: In this case, each voxel is first split into 6 tetrahedra as shown in Figure 4(a) and the intensity is thought of as being specified at the nodes of these elements. Then, an iso-surface of intensity is computed assuming a linear variation of intensity along the edges of the tetrahedral elements. The edges crossed by the desired iso-intensity value are marked, and the points that will comprise the final triangulation are introduced along them. In a second pass over the elements, the number of edges marked is counted, and either one (3 marked edges) or two triangles (4 marked edges) are introduced, as shown in Figure 4(b). The triangles are oriented so that their normal points in the direction of increasing value of intensity. The selection of different iso-intensity values does not greatly affect the final surface tessellation. A typical value of 0.5 is used for binarized images.

Direct tessellation: A second alternative that tends to work better with the smoothing algorithm described below, consists in introducing two (or four) triangles in each of the

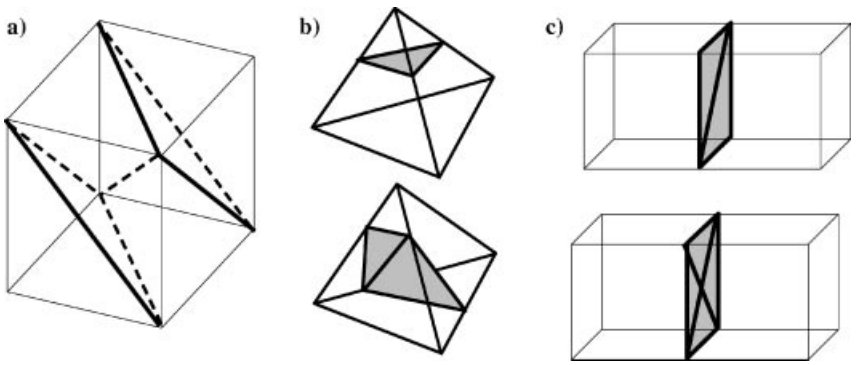

Figure 4. Surface reconstruction: (a) splitting voxels into tetrahedra; (b) construction of iso-surface triangulation; (c) direct tessellation of voxels using two and four triangles per face.

rectangular faces separating voxels with different labels. This approach is illustrated in Figure 4(c). After generating all the triangles, duplicated points are deleted. Points that are separated by a distance less than a specified tolerance are considered as duplicated. This tolerance is selected as 10^{-8} units of length. Using an octree data structure, the number of operations to search for the points close to a given point is of order $n\log(n)$, where n is the total number of points.

3.2. Smoothing

Since voxel faces are parallel to the co-ordinate axes, the tessellated surface tends to be jagged at a scale of a single voxel, even if the image is first blurred and then a true iso-surface extracted. Therefore, it is necessary to smooth the surface triangulation before attempting the generation of the finite element mesh. The need of smoothing the surfaces reconstructed from *in vivo* MR images in order to be used in computational studies has been presented and discussed by Moore *et al.* [27]. It is known that the application of a straight Laplacian smoothing does not conserve the volume, it shrinks the geometry. Thus, a modified version of a non-shrinking smoothing algorithm developed for computer graphics [28] is used. In the usual Laplacian smoothing, the new coordinates are calculated as

$$\mathbf{x}_i' = \mathbf{x}_i + \lambda \Delta \mathbf{x}_i \tag{5}$$

where

$$\Delta \mathbf{x}_i = \sum_{j \in N(i)} w_{ij}(\mathbf{x}_j - \mathbf{x}_i) \tag{6}$$

The weights are given by

$$w_{ij} = \frac{\phi(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{k \in N(i)} \phi(\mathbf{x}_i, \mathbf{x}_k)} \tag{7}$$

where $N(i)$ is the set of points connected to i and $\phi(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^{-1}$ (other forms can be used). This smoothing operator does not preserve the volume of the original triangulation.

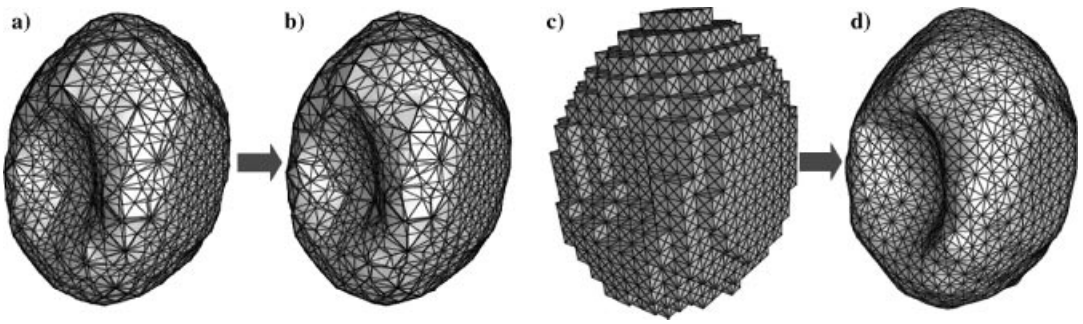

Figure 5. Comparison of iso-surfacing and direct tessellation: (a) iso-surface triangulation; (b) smoothed iso-surface; (c) directly tessellated surface; (d) smoothed direct tessellation surface.

It tends to shrink the mesh ($\lambda > 0$). This problem is solved by performing a second 'un-shrinking' step:

$$\mathbf{x}_i'' = \mathbf{x}_i' + \mu \Delta \mathbf{x}_i' \tag{8}$$

It can be demonstrated that choosing $\mu < -\lambda$ and iterating these two steps for a number of times is equivalent to a band pass filter (see Reference [28] for further details on how to choose these constants). Even though this algorithm provides an un-shrinking smoothing, it does not guarantee that triangles will not cross each other. In fact, it was found that in some cases with small vessel branches triangles did cross each other. Therefore, a modification was added to the scheme: besides considering the points connected to i in $N(i)$ all points contained in a sphere of radius equal to the maximum edge connected to i were considered. The 'force' exerted by these extra points was scaled with a parameter, which can be different in the first (shrinking) and second (un-shrinking) steps. This simple modification avoids the crossing of triangles in many cases and yields a better quality surface grid. The scaling parameter typically used is of the order of 0.3. Thus, the contribution of these extra points to the total force is 30 per cent of that of the points connected to i.

As mentioned earlier, this smoothing technique tends to work better with meshes obtained by direct tessellation of the voxel faces than with iso-surfaces, since in the former case the edge size distribution is uniform. An example is presented in Figure 5.

Figure 5(a) shows the surface triangulation obtained using the iso-surfacing technique, and Figure 5(b) the same surface after smoothing. Figure 5(c) shows the surface triangulation obtained using the direct tessellation algorithm, and Figure 5(d) the same surface after smoothing. Both the surfaces of Figures 5(b) and 5(d) are smooth, i.e. the triangle normals do not change abruptly from neighbour to neighbour. However, the quality of the triangulation is higher for the surface of Figure 5(d). Element quality can be defined from the deviation from equilateral triangles, i.e. deviation of the maximum angle from 60°.

3.3. *Connected components*

In some cases, the tessellated surface may be composed of several groups of disconnected surface triangulations. For example, in Figure 6 the two eyeballs are segmented simultaneously using the region growing algorithm with two seeds. The tessellated surface then consists of

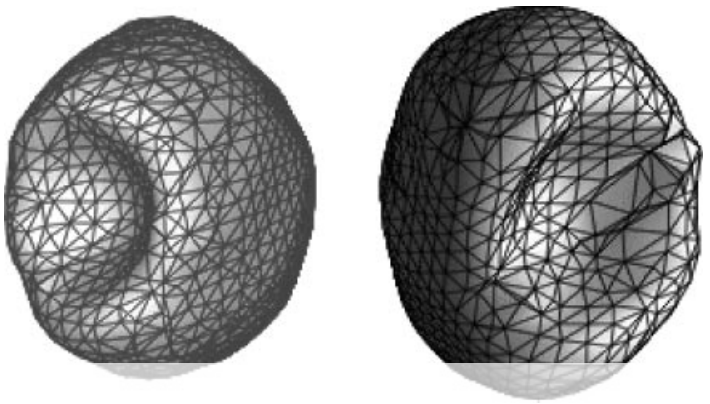

Figure 6. Surface triangulation composed of two groups of topologically connected elements.

two topologically disconnected triangulations, corresponding to the left eye and the right eye, respectively.

The following algorithm is used to classify a list of triangles into groups that are topologically connected. The advancing front-like scheme that assigns neighbour triangles to the same group can be summarized as follows:

```
Set group(ie) = 0 for ie = 0,nelem
Loop until all elements classified
  Get next unclassified element (ie)
  Insert ie into a Queue (front)
  Loop until the Queue is empty
    Extract first element from the Queue (ie)
    Loop over neighbours of ie (je)
      Insert je into the Queue
    End loop
    Set group(ie) = igroup
  End loop
  Igroup = igroup + 1
End loop
```

3.4. *Cutting branches*

In cases where only a portion of the reconstructed arterial tree is to be simulated, it is necessary to cut several branches from the tessellated surface. This is done using a level set approach. First, the geodesic distances to a source point (far from the location of the cut) are computed. In general, the contours of geodesic distance tend to be perpendicular to the axis of the vessel (medial axis) [29]. Thereafter, an average plane (normal and centroid) is calculated from the elements cut by the specified geodesic contour level. This plane is then used to cut the triangles touched by the contour level in question. In this way, planar cuts that are nearly perpendicular to the vessel axis can be computed with a single mouse click

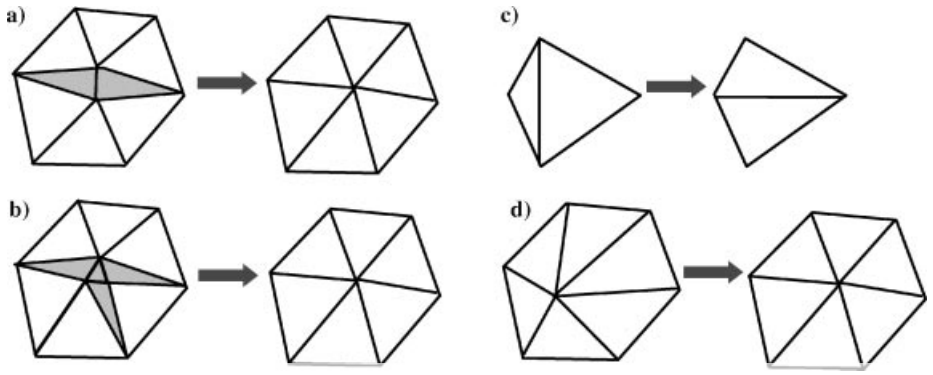

Figure 7. Mesh improvement operations: (a) removal of distorted elements by edge collapse; (b) removal of small elements by edge collapse; (c) diagonal swapping; (d) element smoothing.

(to select the cutting location). Alternatively, arterial branches are cut by specifying a cut plane (normal and point on the plane) and a bounding box. Only surface elements inside the bounding box are cut by the plane. This option requires more work by the user in order to find the correct plane definition and bounding box. However, it can be useful in cases where the contours of geodesic distance are not perpendicular to the medial axis.

3.5. Mesh improvement

The quality of the elements of the surface triangulation can be very poor after smoothing and cutting, i.e. very distorted elements, with very large maximum angles may be present. Thus, several mesh improvement operations are applied. They are schematically summarized in Figure 7. The following operations are applied:

Edge collapse: An edge-collapsing algorithm is used to remove highly distorted or very small elements. First, a loop over the elements is carried out and the ratios of the element edge-lengths are formed. An edge is then marked for collapse if the ratio is above a specified tolerance, i.e. if

$$r = \frac{\text{length}(\text{edge}_i)}{\text{length}(\text{edge}_j)} > \text{tolerance} \tag{9}$$

then edge_i is marked for collapse. The typical value used for the tolerance is 10, which corresponds to an angle opposite to the edge i of approximately 5°.

Then, in a second pass over the elements, marked edges are collapsed and the two adjacent elements deleted, as shown in Figure 7(a). If all three edges of an element have been marked, they are all collapsed and the element and its three neighbours are deleted, as shown in Figure 7(b).

Before an edge is actually collapsed, all the elements connected to this edge are tested for a change in the normals. If the inner product of the element normals before and after the collapse is above a certain tolerance the collapse is not carried out (Figure 8(a)). Typically, changes in the element normal direction of more than 10° are not allowed.

Diagonal swapping: Adjacent triangles are tested for possible diagonal swaps. If the maximum angle of the new configuration is smaller than that of the current configuration, the

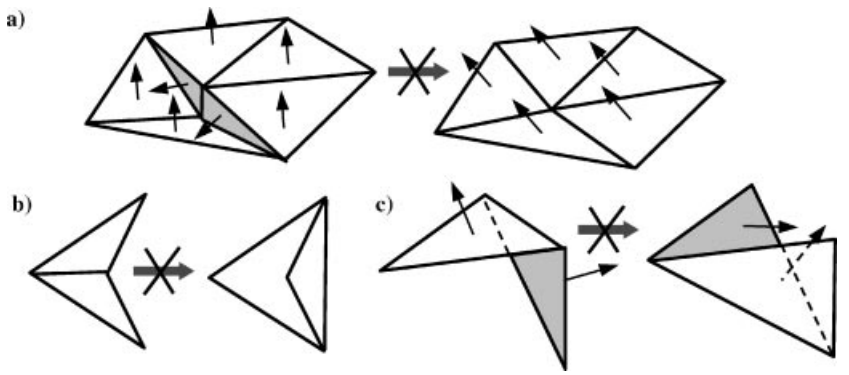

Figure 8. Cases where edge collapse and diagonal swaps are prevented: (a) large change in element normals prevents edge collapse; (b) diagonal swap creates negative elements; (c) large change in element normals prevents diagonal swapping.

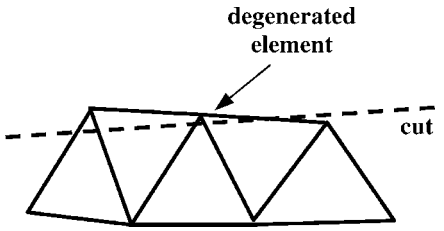

Figure 9. Generation of degenerated elements when cutting the geometry and deleting duplicated points.

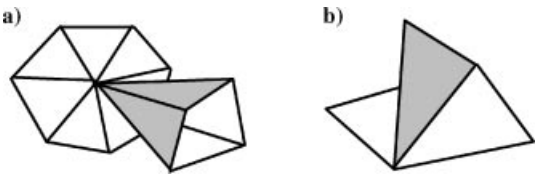

Figure 10. Topological defects: (a) point surrounded by more than one closed circuit; (b) edge connected to more than two triangles.

diagonal is swapped (see Figure 7(c)). If the swapping creates negative elements (Figure 8(b)) or if the inner product of the element normals before and after the swapping differ more than a specified tolerance (Figure 8(c)), the swapping is not performed. As in the previous case, changes in the element normals of more than 10° are prevented.

Element smoothing: In some cases, once the surface mesh has been smoothed as explained before and improved via edge collapse and diagonal swapping it is useful to move the points to the centroid of the polygon formed by all the triangles surrounding the point (see Figure 7(d)). Even though this operation is not volume preserving, it was found that the shrinkage is below the pixel resolution if it is applied after the non-shrinking smoothing scheme and provides a considerable improvement in the element quality.

Deletion of degenerate elements: Elements with repeated nodes are called degenerated elements. They may come about after cutting the surface triangulation and deleting duplicated points. For instance, consider the situation illustrated in Figure 9. If the cutting plane is very close to one of the nodes of an element, it may happen that after cutting and removing duplicated points a degenerated element is generated.

Degenerated elements are detected by testing if the nodes are repeated. Once identified, degenerated elements are removed from the surface triangulation.

Topology corrections: In some cases, such as two small vessels close to each other, the tessellated surface may contain topological defects that need to be identified and corrected (see Figure 10). These defects make the triangulation a non-manifold surface. For instance, one and only one circuit joining the centroids of the elements connected to a point must be found for each point. In cases where more than one of these circuits is found the point is duplicated and moved to the centroid of the circuit. The circuits are identified as follows. An element connected to the point in question is marked. Then, neighbours of this element that are also connected to the point are marked. This process is repeated until all the elements connected to the first element have been marked. All these elements form a loop around the given point. If more elements that have not been marked are connected to the point, a new circuit around the point must exist, and a topological defect has been found.

Another typical topological defect is an edge connected to more than two triangles. These cases are usually avoided by applying the statistical filter described above before the surface tessellation.

4. MESH GENERATION

In order to generate a finite element mesh suitable for CFD calculations, the required element size and shape distributions are specified using background grids and sources. The element sizes can also be automatically specified using adaptive background grids [30, 31]. The radius of curvature of the surface is used as an indicator for the *h*-refinement of the elements of the background grid. The radius of curvature is computed from the change in the normal direction between adjacent elements, i.e. the distance to the point where the lines formed along the normal directions intersect. If the radius of curvature at a given point on the surface is smaller than the element size, then the background grid element containing the point is *h*-refined and the element size reduced.

The usual approach to generate grids over domains defined by discrete surface triangulations is to first fit analytical patches such as B-Splines or NURBS to describe the geometry, and then mesh over these patches.

However, we prefer to operate directly on the discrete surfaces. In order to preserve features present in the given surface triangulation, a feature detection algorithm is used to define discrete surface patches [32]. If the angle between the normals of two adjacent triangles exceeds a certain tolerance, the common edge between the elements is defined as a *ridge*. *Corners* are defined as points that are attached to:

(a) only one ridge;
(b) more than two ridges; or
(c) two ridges with considerable deviation of unit side vectors, i.e. the angle between two ridges exceeds a given tolerance.

Between corners, the ridges form *discrete lines*. These discrete lines either separate or are embeded completely (i.e. used twice) in *discrete surface patches*, as shown in Figure 11.

Once the discrete surface patches have been defined, a new surface grid that conserves the identified features is generated by re-triangulating the tessellated surface, using an advancing front technique [31]. Points and sides are first introduced along the discrete lines. These form the initial front. Then, each patch is re-triangulated in turn. Starting from the smallest side

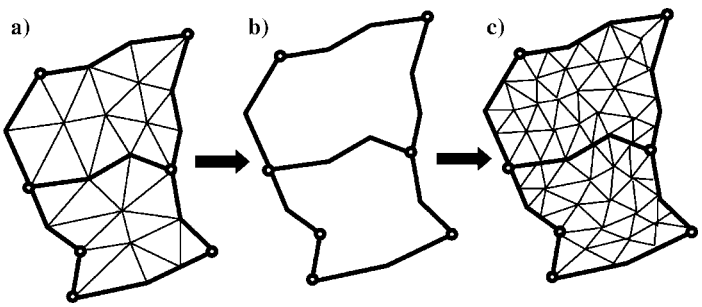

Figure 11. Feature detection algorithm for surface re-triangulation: (a) original triangulation, detected ridges and corners; (b) discrete surface patches; (c) re-triangulated surface preserving identified features.

in the front, new points are introduced to form triangles on the discrete surface patch. The generated surface points can either follow the underlying surface triangulation exactly (i.e. linear interpolation), or can be placed on a local parabolic element. This latter option is commonly used, as the CFD grids typically used are much finer than the underlying triangulation defining the surface. The new surface triangulation has the desired element size and shape distribution specified by the background grid and sources. For implementation details of the surface gridding technique the interested reader is referred to [32]. Finally, after remeshing the surface, the volume mesh is generated using again an advancing front method. In this case the newly generated surface mesh constitutes the initial front and new points are introduced in the interior of the domain to form tetrahedral elements.

The tolerance used to define ridges and corners in the feature detection algorithm, depends on which features one wants to preserve in the final mesh. If most of the details in the original triangulation are to be conserved, a typical value used for the angle tolerance is 10°. However, as this tolerance is increased, fewer ridges are detected and the final mesh is much smoother since the grid generator meshes over these details. In the limit, when any angle is allowed and no ridges are found, all the small details in the original mesh are neglected. In many cases this is desirable since the details come from imperfections in the initial triangulation due to the original voxelization of the vessels.

5. EXAMPLES

The following series of examples serve two main purposes:

(a) Illustrate the complete process of generating CFD grids starting from medical images of two kinds (CT and MRA).
(b) Show that the generated meshes can be used for CFD calculations of hemodynamics over complex anatomical geometries.

The flow calculations presented were carried out using a time-dependent finite element incompressible Navier–Stokes solver for unstructured grids, based on a pressure projection scheme [33, 34]. The advection terms are discretized in time explicitly and the viscous fluxes implicitly. At each time-step, a Laplace equation is solved for the pressure using a preconditioned

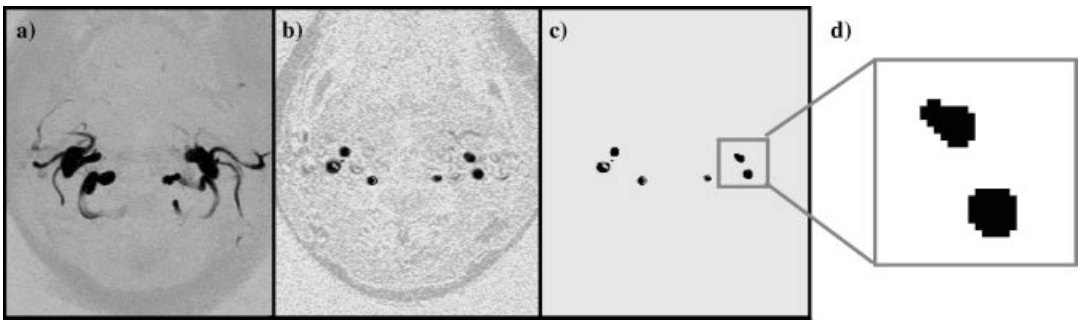

Figure 12. CTA image processing: (a) MIP of the original 3D image; (b) original middle slice ($z = 25$); (c) middle slice after sharpening; (d) detail of middle slice after region growing segmentation of the left carotid artery.

conjugate gradient matrix solver. The arterial walls are assumed rigid and the blood rheology is approximated by a Newtonian fluid of constant (laminar) viscosity $\mu = 0.04$ P and density $\rho = 1.0\,\mathrm{g/cm^3}$. Unsteady flow solutions were obtained by prescribing pulsating boundary conditions from flux–rate curves and using the Womersley velocity profile for fully developed flows [35]. All calculations were performed in cgs units, which is reflected in the labels of the figures showing the results. We emphasize once again that the purpose of the paper is not to discuss in detail a particular hemodynamic simulation, but the rapid construction of grids from anatomical images. We therefore keep the hemodynamic discussion of each case to a minimum, and only provide the essential data.

5.1. Carotid artery

In this case the geometry of the carotid artery is reconstructed from a CTA image composed of 50 slices of 256×256 pixels with 8 bits per pixel. The image processing pipeline is illustrated in Figure 12. Figure 12(a) shows the maximum intensity projection (MIP) of the original 3D image and Figure 12(b) the original middle slice ($z = 25$). The middle slice after application of the sharpening operator is shown in Figure 12(c) and a detail of the middle slice after segmentation of the left carotid artery using the region growing algorithm is shown in Figure 12(d).

Once the image was segmented, a surface tessellation was created by introducing two triangles at the faces of the boundary voxels, as explained before. This surface was the smoothed using the non-shrinking scheme presented earlier. The initial surface tessellation is shown in Figure 13(a). The same surface after smoothing is shown in Figure 13(b).

This geometrical model was then used to generate a finite element mesh. The discrete lines detected using an angle tolerance of 10° to define ridges are shown in Figure 13(c). The final finite element mesh was composed of approximately 1.4 M tetrahedral elements and 75K triangles on the surface. Figure 13(d) shows the surface of this volumetric finite element mesh.

A CFD simulation of the pulsating flow in the left carotid artery was then performed using this finite element mesh. At the common carotid artery (CCA) the velocity profile was prescribed from the Womersley solution corresponding to the volume–flux curve presented

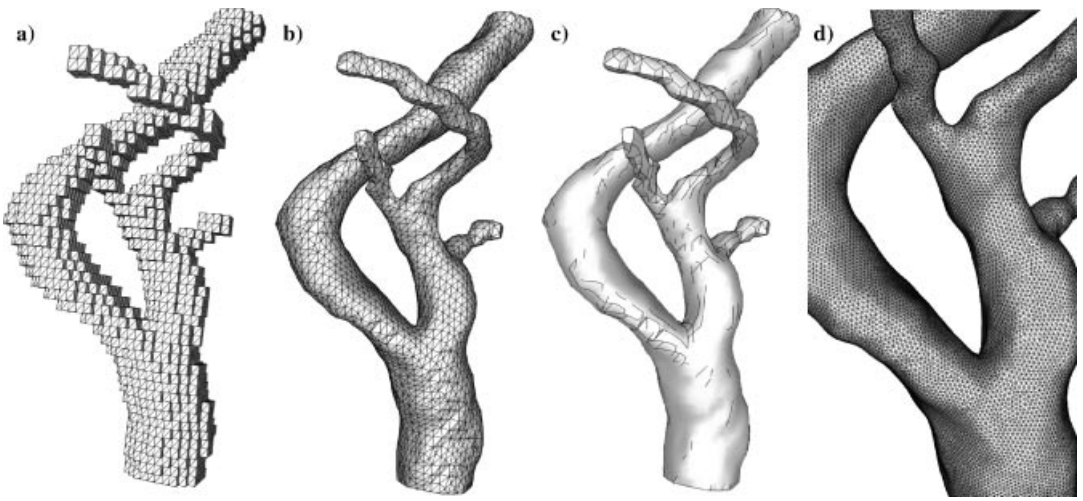

Figure 13. Construction of a finite element mesh for the left carotid artery: (a) initial surface tessellation; (b) smoothed surface triangulation; (c) detected discrete lines and surface patches; (d) detail of the surface of the final volumetric finite element grid.

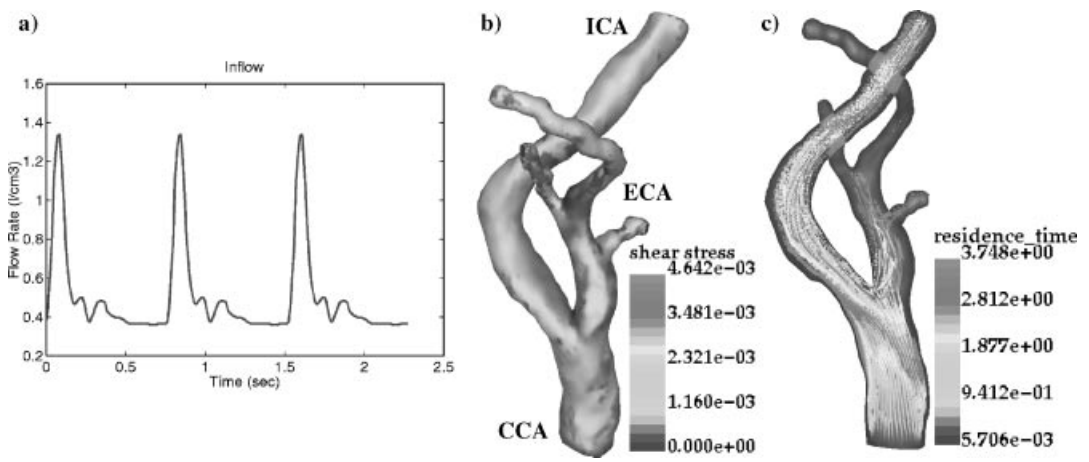

Figure 14. CFD flow calculation: (a) prescribed volume–flux at the CCA; (b) wall shear stress distribution at the peak of the flow curve; (c) particle traces coloured with residence time.

in Figure 14(a). Traction-free boundary conditions were specified at the outlet of the internal carotid artery (ICA) and the exits of the three branches of the external carotid artery (ECA).

The flow field obtained with under these conditions is presented in Figure 14. The instantaneous wall shear stress distribution at the peak of the flow curve is shown in Figure 14(b). It can be seen that the shear stress is lowest at sites of arterial branching and highest in

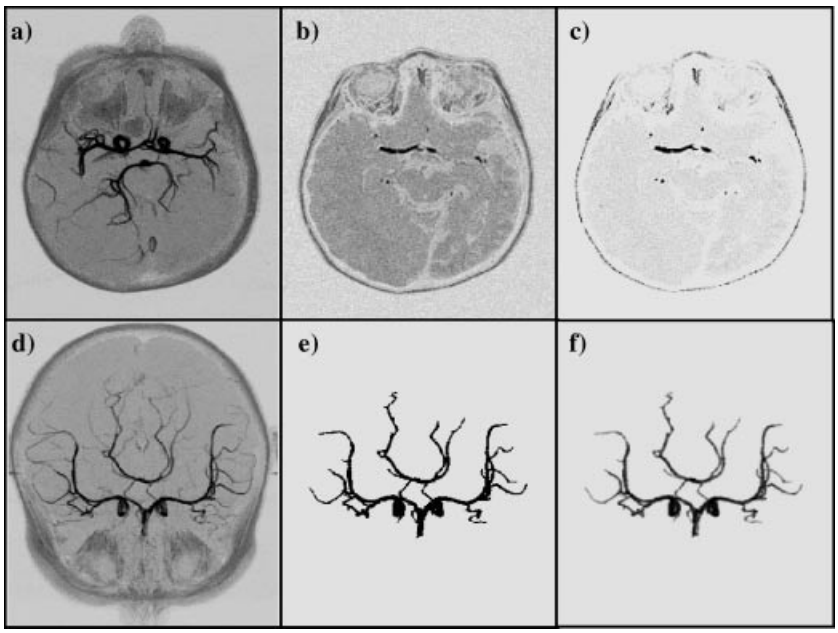

Figure 15. MRA image processing: (a) original MIP for patient 1; (b) slice at $z = 32$ for patient 1 before processing; (c) slice at $z = 32$ for patient 1 after sharpening; (d) original MIP for patient 2; (e) MIP of segmented image; (f) MIP of segmented image after smoothing.

regions of strong secondary flows or geometrical constrictions. Figure 14(c) shows particle traces coloured according to residence times. Secondary flow patterns can be observed at the bifurcation of the CCA and along the ICA.

5.2. Circle of Willis

In this example, the anatomy of the cerebral arteries comprising the circle of Willis is modelled from two MRA data sets corresponding to two different patients. The original contrast-enhanced MRA images consist in a stack of 64 slices of 512×512 pixels. Grey levels were specified using 16 bits per pixel.

The objective of these CFD calculations is the study of the blood flow near the anterior cerebral artery, which is a common place for the development of aneurysms. A common practice during surgical procedures to treat these aneurysms is to occlude a vessel in order to reduce blood flow into the aneurysm. However, the closure of a vessel can lead to reduced blood flow in other regions of the brain, causing ischemic stress and eventually a stroke. On the other hand, it may happen that clipping certain arteries does not actually reduce the flow of blood as expected, due to the anatomical redundancy of the circle of Willis. Therefore, it is of great value for surgical planning to calculate the redistribution of blood when an arterial clipping is carried out.

The processing of the MRA images for both patients is illustrated in Figure 15. The top row corresponds to patient 1 and the second row to patient 2. Figure 15(a) shows the MIP projection of the original data-set for patient 1. Figure 15(b) shows a slice at $z = 32$ for patient

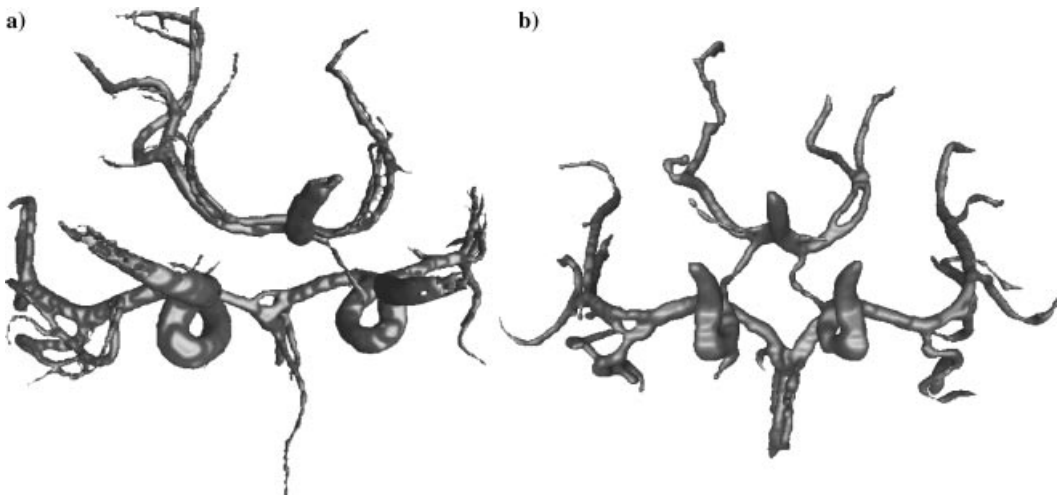

Figure 16. Reconstructed anatomical models after tessellation and smoothing for: (a) patient 1; (b) patient 2.

1 before image processing and Figure 15(c) the same slice after sharpening. For patient 2, Figures 15(d)–15(f) show the MIP projection of the original images, the image after region growing segmentation and the segmented image after smoothing, respectively.

The surface model for patient 1 was created using the direct tessellation scheme applied the segmented image. This model was then smoothed using the modified version of the non-shrinking algorithm described earlier in order to avoid triangle intersections in the smallest branches. This surface model is shown in Figure 16(a). For patient 2, it was found that smoothing (blurring) the image before tessellation yielded a better surface triangulation, i.e. smoother and with better element quality, than tessellating directly the segmented image. The smoothed surface tessellation thus constructed is shown in Figure 16(b). In these models, as well as in the MIP of the original images, the large variability of the cerebrovascular anatomical characteristics from individual to individual can be observed. For instance, in patient 1 the left posterior communicating artery is missing, leading to an incomplete circle of Willis. This example shows the importance of the anatomical modelling on an individual basis for the planning of surgical and interventional procedures.

Several branches were cut from these models using the contours of geodesic distance as explained before. The anterior part of the circle of Willis for patient 1 is shown in Figure 17(a), while the complete circle of Willis for patient 2 is shown in Figure 17(b). Some of the major arterial branches are identified in Figure 17.

These final models were then used to generate a volumetric finite element mesh for each patient. In both cases, a single discrete surface patch was defined for the whole surface, i.e. no ridges were allowed by specifying a large angle tolerance for the variation of the normals between adjacent triangles. Closed surfaces were obtained by defining planar surfaces closing the ends of the arterial branches that were cut. A uniform element size distribution was specified using a background grid, and a few sources added in order to reduce the element size along small arterial branches.

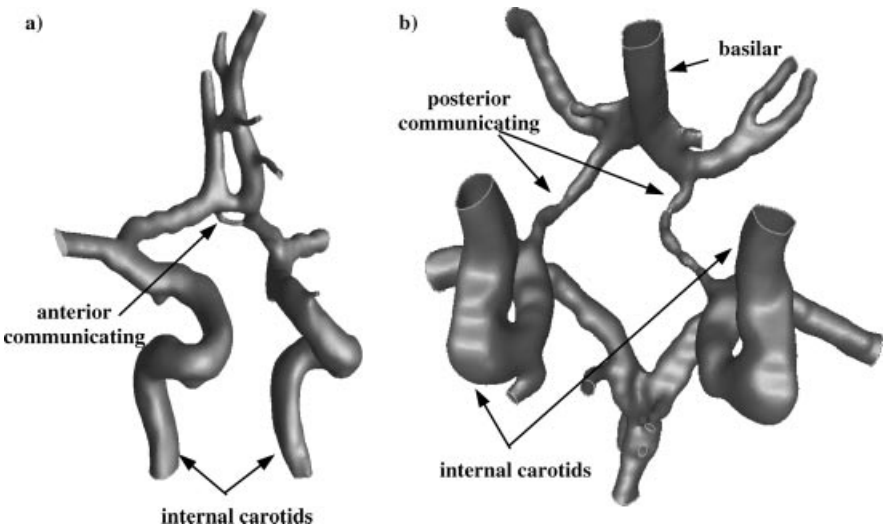

Figure 17. Final surface models after cutting branches: (a) anterior part of the circle of Willis for patient 1; (b) complete circle of Willis for patient 2.

The final finite element mesh for patient 1 contained approximately 800K tetrahedral elements and 60K triangles on the surface. For patient 2, the final grid contained roughly 4M tetrahedra and 175K triangles on the surface. The surface of the finite element grids for patients 1 and 2 are shown in Figures 18(a) and 18(b), respectively.

As mentioned before, these finite element grids were used to investigate the changes in the flow patterns and flow re-distribution precipitated by the occlusion of a cerebral artery. At the time when these calculations were performed, no physiologic flow measurements were available. Thus, idealized or 'typical' flow-rate curves were prescribed at the entrance of both internal carotid arteries and the basilar artery (Figure 14(a)). At these inflow boundaries, the velocity profile was derived from the Womersley solution for a pulsating flow in a circular pipe [35]. At the outflow boundaries, a vascular bed model was adopted [36]. This model assumes that all brain arteries bifurcate a number of times until the capillary bed is reached, were the pressure is constant. Then, a pressure boundary condition of the form $p = p_0 + RQ$ is imposed, where p_0 is the pressure at the capillary bed, Q if the volume–flux through the boundary and R represents the flow resistance up to the capillary bed. The values of R can be deduced from the area of the exit boundary and the distance to the capillary bed, assuming a geometric similarity of arterial bifurcations. For a more detailed discussion of these flow models see [36].

The computed flow patterns before and after clipping the left anterior cerebral artery for patient 1 are shown in Figures 19(a) and 19(b), respectively. Instantaneous streamlines emanating from the entrance of the two internal carotid arteries at time $= 0$ are coloured according to the velocity magnitude. Significant changes in the flow pattern can be clearly seen. In particular, before the clipping is carried out there is very little flow through the anterior communicating artery. However, after clipping, streamlines originating in the right internal carotid artery pass through the anterior communicating artery, where there is a significant

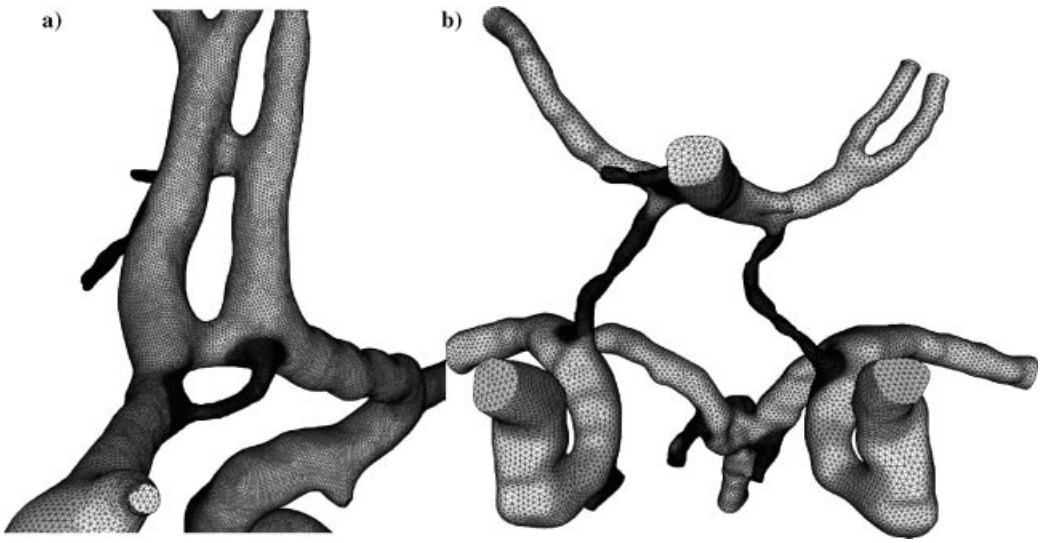

Figure 18. Surface of the finite element grids generated from the surface models of: (a) patient 1; (b) patient 2.

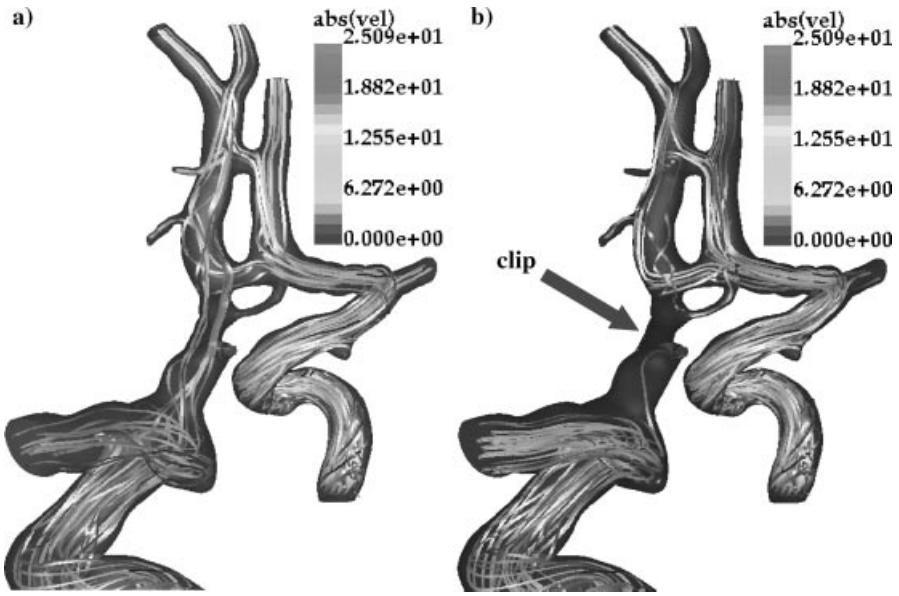

Figure 19. Visualization of the flow pattern for patient 1 using instantaneous streamlines at time = 0 coloured according to velocity magnitude: (a) before clipping; (b) after clipping the left anterior cerebral artery.

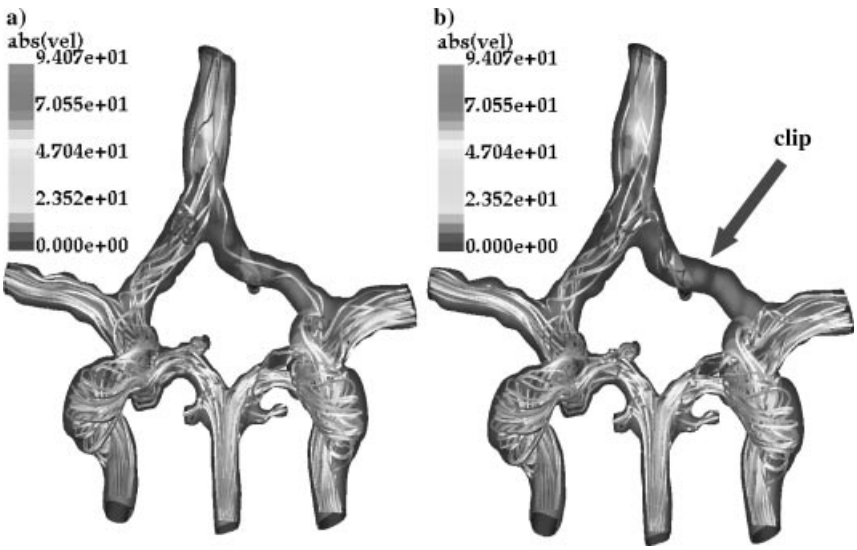

Figure 20. Visualization of the flow pattern for patient 2 using instantaneous streamlines at time = 0 coloured according to velocity magnitude: (a) before clipping; (b) after clipping the left anterior cerebral artery.

increase in the volume–flux. This result was to be expected since the communicating arteries usually play a passive role. They provide an alternative circuit in the event of the occlusion of one of the arteries feeding the circle of Willis.

A similar flow calculation is presented in Figure 20 for patient 2. As in the previous case, the flow patterns before and after clipping the anterior cerebral artery are visualized using instantaneous streamlines at time = 0, coloured according to the velocity magnitude. The computed flows are quite complex and secondary flows due to the vessel curvature can be clearly seen.

The quantitative changes in the volume–flux through each of the outflow boundaries are presented in Figure 21(a). The volume–flux before and after clipping is shown for each arterial branch in the model. Again as in the previous case, significant flow changes take place in some arterial branches when the anterior cerebral artery is temporarily occluded. The redundancy of the cerebral arterial system provided by the circle of Willis, and the large individual variability of the cerebral vasculature make the predictions of the outcome of maneuvers such as the one considered here very difficult. However, the predictability provided by the CFD modelling of blood flow in cerebral arteries, as exemplified here, makes this methodology very useful for planning surgical and interventional procedures on a patient-specific basis.

It is also interesting to observe the instantaneous wall shear stress distribution, shown in Figure 21(b). Regions of low shear stress are usually indicative of places where arterial diseases such as aneurysms develop. It can be seen that low shear stresses are obtained at the site of arterial branching and also in the communicating arteries, where there is low flow under normal conditions. This result is encouraging since it is well known that aneurysms usually develop at the sites of arterial bifurcations [37].

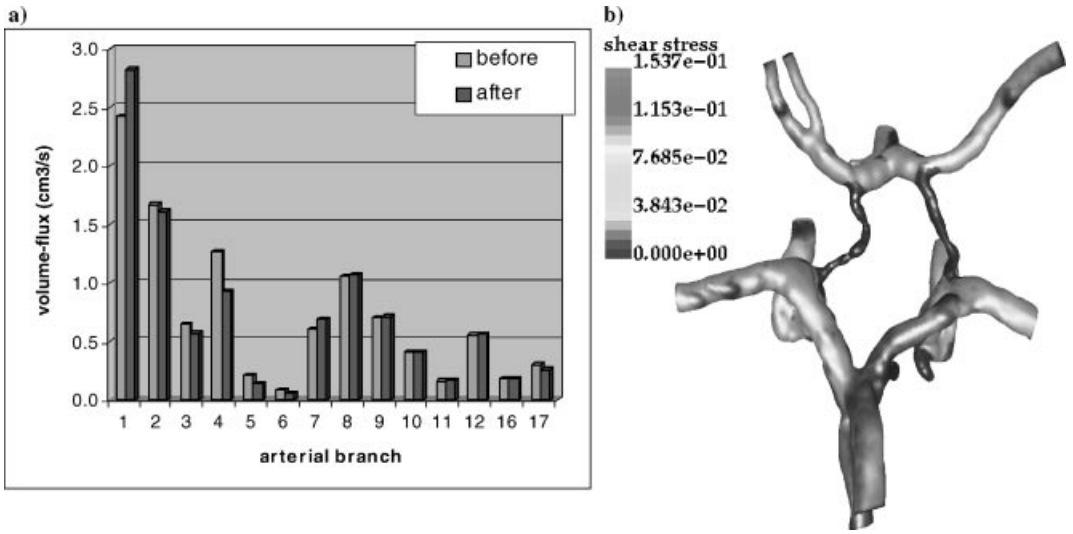

Figure 21. CFD results for patient 2: (a) changes in volume–flux through each arterial branch when the anterior cerebral artery is clipped (cm^3/s); (b) wall shear stress distribution under normal conditions (no clipping).

6. DISCUSSION

Various image processing and geometrical modelling techniques were integrated in order to quickly generate geometrical models (surface triangulations) of arteries starting from medical images. These discretely defined surfaces are then used to generate anatomically accurate finite element grids for hemodynamic simulations via the advancing front technique.

The proposed technique consists of three stages: image processing, geometrical modelling and grid generation. The aim of the first stage (image processing) is to obtain a properly defined boundary of the arterial lumen walls in the region of interest. In the second stage, the tessellation or iso-surface obtained from the first stage is made topologically consistent, smoothed, and the quality of the resulting triangulation is improved. Finally, in the third stage, the triangulation is subdivided into so-called discrete surface patches for surface gridding, the desired element size distribution is defined and the grid generated.

The examples presented illustrate the process of construction of proper CFD grids starting from high contrast medical images (CTA, MRA). Moreover, the examples show that the generated finite element meshes can be used to conduct hemodynamic calculations in fairly complex anatomical geometries. The total time required by the present methodology to generate the CFD meshes starting from the medical images is of the order of a few hours. User interaction is only needed in the segmentation stage (to specify seeds and intensity ranges), and in the geometry modelling stage (to cut arterial branches and define boundary conditions). The mesh improvement steps as well as the surface re-gridding and volume grid generation are completely automatic and can run in parallel on shared memory computers. The time spent in these last stages depends only on the grid sizes and the number of processors employed.

More automation of the image segmentation and geometry modelling techniques is required to reduce the total times further.

Each of the three stages: image processing, surface reconstruction and grid generation can lead to errors that can degrade the fidelity of the final surface used for CFD calculations. Studies by Moore *et al.* [22] have shown that even small errors in the surface definition can have a significant effect on the shear stress distribution for steady flows, but less influence for pulsatile flows. In the sequel, we discuss the possible sources of errors in each of the stages in turn.

Any image segmentation method attempts to determine which voxels are inside and outside of the vessels, i.e. detection of the vessel boundary. Several imaging artifacts make this a difficult task. In the particular case of MRA, these artifacts include those related to flow causing non-uniform lumen signal, magnetic field inhomogeneity, imperfect timing of the arrival of the contrast agent, and the partial volume effect. Image noise is also considerable and image resolution is low relative to the vessel diameter. The image-sharpening operator described aims to reduce the noise while preserving the location of the vessel boundaries. After sharpening a high contrast image, the choice of the intensity range for the region growing algorithm presented does not greatly affect the results. The accuracy of the surface reconstruction of vessels is limited primarily by the resolution of the image. Another factor is the lack of definition of the distal extent of the vascular tree. The number and extent of small vessels detected depends on the acquisition method.

The largest possible error during the surface reconstruction stage is due to smoothing. The smoothing procedure described was chosen explicitly to mitigate smoothing errors, i.e. conserve the volume. The error incurred during this stage is of the order of the voxel size, since all points deviate at most half a voxel from the iso-surface obtained from the image processing stage.

The surface grids typically used for CFD simulations are considerably finer than the underlying triangulation defining the surface. The error here could result from an exact projection of the new points to the underlying triangulation. The quadratic reconstruction procedure attempts to reduce this error, which is again of the order of the voxel size.

At this stage, it is not possible to determine the combined effect of these errors, i.e. the global error in the surface reconstruction and grid generation. We estimate that the combined error will not be larger than one voxel size, but a convincing proof of this hypothesis awaits further investigations.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. M. Venere (Universidad del Centro de la Provincia de Buenos Aires, Argentina) and Dr. F. Calamante (MR Unit, Great Ormond Street Hospital for Children, London, U.K.) for providing the CTA and MRA images, respectively.

REFERENCES

1. Bassiouny HS, White S, Glagov S, Choi E, Giddens DP, Zarins CK. Anastomotic intimal hyperplasia: mechanical injury or flow induced. *Journal of Vascular Surgery* 1992; **15**:708–717.
2. Giddens DP, Tang TD, Loth F. Fluid mechanics of arterial bifurcations. In *Biological Flows*. Jaffrin MY, Caro C (eds), Plenum Press: New York, 1995: 51–68.

3. Taylor CA, Draney MT, Ku JP, Parker D, Steele BN, Wang K, Zarins CK. Predictive medicine: computational techniques in therapeutic decision-making. *Computer Assisted Surgery* 1999; **4**:231–247.
4. Sugawara M, Kajiya F, Kitabatake A, Matsuo H (eds), *Blood Flow in the Heart and Large Vessels*. Springer: Tokyo, Berlin, New York, 1989.
5. Buckley JA, Shifrin RY, Paik DS, Rubin GD. Automated volumetric quantification of blood vessels using MR angiography. *Proceedings of the RSNA* 1998; 475.
6. Fredrickson JO, Irarrazabal P, Pelc NJ. Quantitative 3D time-resolved phase contrast MR imaging. *SMR Workshop*: *Cardiovascular MRI*: *Present and Future*, 1994; 25.
7. Mazumdar J. *Biofluid Mechanics*. World Scientific: Singapore, 1992.
8. Perktold K, Rappitsch G. Computer simulation of arterial blood flow. Vessel diseases under the aspect of local hemodynamics. In *Biological Flows*. Jaffrin MY, Caro C (eds), Plenum Press: New York, 1995; 83–114.
9. Ma P, Li X, Ku DN. Convective mass transfer at the carotid bifurcation. *Journal of Biomechanics* 1997; **30**(6):565–571.
10. Taylor CA, Hughes TJR, Zarins CK. Finite element modelling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering* 1998; **158**:155–196.
11. Perktold K, Peter RO, Resch M, Langs G. Pulsatile non-Newtonian blood flow in three-dimensional carotid bifurcation models: a numerical study of flow phenomena under different bifurcation angles. *Journal of Biomedical Engineering* 1991; **13**:507–515.
12. Doorly DJ, Peiró J, Sherwin SJ, Shah O, Caro CG, Tarnawski M, McLean M, Dumoulin C, Axel L. Helix and model graft flows: MRI measurement and CFD simulations. *ASME Paper* FEDSM97-3423 ASME FED Meeting, Vancouver, Canada, June 22–26, 1997.
13. Lei M, Kleinstreuer C, Archie JP. Hemodynamic simulations and computer-aided design of graft-artery junctions. *Journal of Biomechanical Engineering* 1997; **119**:343–348.
14. Santamarina A, Weydahl E, Siegel Jr JM, Moore Jr JE. Computational analysis of flow in a curved tube model of the coronary arteries: effects of time-varying curvature. *Annals of Biomedical Engineering* 1998; **26**: 944–954.
15. Perktold K, Hofer M, Karner G, Trubel W, Schima H. Computer simulation of vascular fluid dynamics and mass transport: optimal design of arterial bypass anastomoses. *Proceedings of ECCOMAS 98*, vol. 2, Wiley: New York, 1998; 484–489.
16. Moore JA, Steinman DA, Ethier CR. Computational blood flow modelling: errors associated with reconstructing finite element models from magnetic resonance images. *Journal of Biomechanics* 1998; **31**:179–184.
17. Klette R, Zamperoni P. *Handbook of Image Processing Operators*. Wiley: New York, 1996.
18. Finnigan P, Hathaway A, Lorensen W. Merging CAT and FEM. *Mechanical Engineering* 1990; **112**(7):32–38.
19. Johnson CR, MacLeod RS, Schmidt JA. Software tools for modelling, computation, and visualization in medicine. *CompMed 94 Proceedings*, World Scientific, 1995.
20. Quarteroni A, Tuveri M, Veneziani A. Computational vascular fluid dynamics: problems, models and methods. *Report EPFL/DMA 11.98*, 1998.
21. Moore JA, Steinman DA, Holdsworth DW, Ethier CR. Accuracy of computational hemodynamics in complex arterial geometries reconstructed from magnetic resonance imaging. *Annals of Biomedical Engineering* 1999; **27**:32–41.
22. Cline HE, Lorensen WE, Souza SP, Jolesz FA, Kikinis R, Gerig G, Kennedy TE. 3D surface rendered MR images of the brain and its vasculature. *JCAT* 1991; **15**:344–351.
23. Lei T, Udupa JK, Saha PK, Odhner D. MR angiographic visualization and artery-vein separation. *Proceedings of SPIE* 1999; **3658**:58–66.
24. Dale AM, Fischl B, Sereno MI. Cortical surface-based analysis I: segmentation and surface reconstruction. *NeuroImage* 1999; **9**(2):179–194.
25. Burdin V, Roux C. Surface segmentation of long bone structures from CT images using a deformable contour model. *Proceedings of 16th Annual International Conference IEEE Engineering in Medicine and Biology Society*, Baltimore, November 3–6. 1994.
26. McInerney T, Terzopoulos D. Medical image segmentation using topologically adaptable surfaces. *Proceedings of CVRMed '97*, Grenoble, France, 1997.
27. Bulpitt AJ, Berry E. Spiral CT of abdominal aneurysms: comparison of segmentation with an automatic 3D deformable model and interactive segmentation. *Proceedings of SPIE*, *Medical Imaging* 1998; **3338**:938–946.
28. Taubin G. A signal processing approach to fair surface design. *Computer Graphics Proceedings* 1995; 351–358.
29. Lazarus F, Verroust A. Level set diagrams of polyhedral objects. *Rapport de Recherche INRIA* No. 3250, 1997.
30. Löhner R. Extensions and improvements of the advancing front grid generation technique. *Communications in Numerical methods and Engineering* 1996; **12**:683–702.
31. Löhner R. Automatic unstructured grid generators. *Finite Elements in Analysis and Design* 1997; **25**:111–134.
32. Löhner R. Regridding surface triangulations. *Journal of Computational Physics* 1996; **126**:1–10.
33. Ramamurti R, Löhner R. A parallel implicit incompressible flow solver using unstructured meshes. *Computers and Fluids* 1996; **5**:119–132.

34. Löhner R, Yang C, Oñate E, Idelssohn S. An unstructured grid-based, parallel free surface solver. *Applied Numerical Mathematics* 1999; **31**:271–293.
35. Womersley JR. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *Journal of Physiology* 1955; **127**:553–563.
36. Cebral JR, Löhner R, Burgess JE. Computer simulation of cerebral artery clipping: relevance to aneurysm neuro-surgery planning. *Proceedings of the ECCOMAS 2000*, Barcelona, Spain, September 11–14, 2000.
37. De Groot J. *Correlative Neuroanatomy*. Appleton & Lange, Conn., 1991.

