INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING
Int. J. Numer. Meth. Biomed. Engng. (2011)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cnm.1446

REVIEW

Modelling pipeline for subject-specific arterial blood flow—A review

Igor Sazonov[1], Si Yong Yeo[1], Rhodri L. T. Bevan[1], Xianghua Xie[2], Raoul van Loon[1] and Perumal Nithiarasu[1,*,†,‡]

[1] *Computational Bioengineering Group, College of Engineering, Swansea University, Swansea SA2 8PP, U.K.*
[2] *Department of Computer Science, Swansea University, Swansea SA2 8PP, U.K.*

SUMMARY

In this paper, a robust and semi-automatic modelling pipeline for blood flow through subject-specific arterial geometries is presented. The framework developed consists of image segmentation, domain discretization (meshing) and fluid dynamics. All the three subtopics of the pipeline are explained using an example of flow through a severely stenosed human carotid artery. In the Introduction, the state-of-the-art of both image segmentation and meshing is presented in some detail, and wherever possible the advantages and disadvantages of the existing methods are analysed. Followed by this, the deformable model used for image segmentation is presented. This model is based upon a geometrical potential force (GPF), which is a function of the image. Both the GPF calculation and level set determination are explained. Following the image segmentation method, a semi-automatic meshing method used in the present study is explained in full detail. All the relevant techniques required to generate a valid domain discretization are presented. These techniques include generating a valid surface mesh, skeletonization, mesh cropping, boundary layer mesh construction and various mesh cosmetic methods that are essential for generating a high-quality domain discretization. After presenting the mesh generation procedure, how to generate flow boundary conditions for both the inlets and outlets of a geometry is explained in detail. This is followed by a brief note on the flow solver, before studying the blood flow through the carotid artery with a severe stenosis. Copyright © 2011 John Wiley & Sons, Ltd.

Received 16 November 2010; Revised 7 March 2011; Accepted 13 March 2011

KEY WORDS: subject-specific modelling; pipeline; image segmentation; surface and volume meshing; boundary layer; fluid dynamics; carotid bifurcation; stenosis

1. INTRODUCTION

The human circulatory system consists of vessels that transport blood throughout the body, providing the tissues with oxygen and nutrients. It is known that vascular diseases, such as stenosis and aneurysms, are often associated with changes in blood flow patterns and the distribution of wall shear stress (WSS). Modelling and analysis of the hemodynamics in the human vascular system improve our understanding of vascular disease, and provide valuable insights, which can help in the development of efficient treatment methods. In recent years, computational methods have been widely used for patient-specific modelling of blood flow in vascular structures [1–11]. Despite the involvement of numerous groups working in this field, and rapid advancement in efficient computational methods, there has been limited applications of computational hemodynamics in clinical

[*] Correspondence to: Perumal Nithiarasu, College of Engineering, Swansea University, Swansea SA2 8PP, U.K.
[†] E-mail: P.Nithiarasu@swansea.ac.uk
[‡] Professor.

practice. This is largely due to the challenges involved in the design of an integrated framework which can robustly and accurately automate the interdisciplinary computational modelling process, which includes image segmentation, mesh generation, and computational fluid dynamics (see Appendix A for a flow chart). In this paper, we describe a semi-automatic framework for image-based computational modelling of blood flow dynamics in arteries, in addition to providing a review.

The general structure of any subject-specific blood flow modelling framework consists of mainly image segmentation, meshing and solution stages. The image segmentation stage depends heavily on the type of image available and its resolution. The image processing is normally followed by a meshing stage in which a domain discretization is carried out. The connection between the image processing and meshing is established via a geometry definition step. Majority of the efforts during the mesh generation process go towards establishing a valid and high-quality surface mesh that is a close representation of the geometry. The surface meshing in general is followed by the boundary layer mesh construction and artery wall meshing (when applicable). The boundary layer and arterial wall volume discretizations are followed by automatic volume meshing of the central flow domain of the artery with unstructured meshes. The flow solver is then used along with appropriately generated boundary conditions to complete the pipeline. The solver may include rigid geometry, flow and structure coupling, Newtonian or non-Newtonian approximation, and may also include multiscale mechanics or turbulence. In this study, our focus is on all the three stages of a pipeline with special emphasis on the first two stages.

One of the main challenges in the computational modelling of hemodynamics is the accurate reconstruction of a blood vessel geometry. Anatomically accurate geometric models of the lumens are essential for realistic flow simulations and analysis. The anatomical information used to reconstruct the geometries are usually provided in the form of medical image data sets (scans) from imaging modalities such as computed tomography (CT) and magnetic resonance (MR) imaging. Manual reconstruction of the geometries can be tedious and time consuming. There is also the issue of variability between the geometries extracted manually by different individuals, and variability of geometries extracted by the same individual on different occasions. In order to allow computational flow modelling to be efficiently applied as a diagnostic or predictive tool, the amount of user intervention required in the process should be reasonably small. Therefore, a robust and efficient method that can be used to accurately segment the geometry from medical image data sets can be very useful and advantageous in the modelling process. The current state-of-the art of medical segmentation methods is explained in Section 1.1.

Assuming that a reasonably accurate reconstruction of the geometry is possible, the next stage of the pipeline would be to generate a surface mesh or meshes to accurately discretize the surface or surfaces of the geometry. Unlike well-defined standard engineering geometries, the reconstructed subject-specific, arterial geometries are often defined by binaries. Thus, an alternative approach is required to that of the standard engineering applications. In Section 1.2, we provide a brief summary on the current methods used for patient-specific surface and volume mesh generation.

Since the fluid dynamics part of the pipeline is one of the widely studied areas, only a very brief account of the method is given in Section 1.3. In Section 1.4, the proposed modelling pipeline is explained in a general sense. Section 2 provides a deformable model-based segmentation method (first part of the pipeline) for blood vessel reconstruction and the meshing methods and cosmetics used in this study are explained in detail in Sections 3 and 4 (second part of the pipeline). In Section 5, the flow modelling steps, including the correct boundary condition generation, boundary condition cosmetics, governing flow equations and their solution method are provided (third and final part of the pipeline). An example flow result obtained for a carotid artery using the proposed modelling pipeline is given in Section 6 and finally, Section 7 draws some conclusions and discusses the challenges and unresolved problems.

1.1. Image segmentation

Although several techniques exist for the segmentation of blood vessels from medical images, it remains an intricate process due to factors such as image noise, partial volume effects, image

DOI: 10.1002/cnm

artefacts, intensity inhomogeneity, and changes in topology. In [12], the coordinate points for the center line of an aortic arch were extracted from volume-rendered MR images. A cubic spline was then used to represent the aortic centerline, and cross-sectional grids were generated on normal planes at equidistant points along the curve. This generated a curved tube with circular cross-section of uniform radius, which is not representative of the geometry of an aorta. In [13], the centerline and diameter information of the vessels were extracted from the image data set, and the vascular model was reconstructed using non-uniform rational B-splines (NURBS). Such techniques may often smooth out geometric information that can be important to the computation of accurate flow dynamics, such as those at bifurcations.

The 3D models of the vascular structures in [14] were reconstructed by extracting the 2D contours of the vessels from the image slices of the MR image data set, and then lofting through the contours to create the surface models of the vessels. The different vessels were then merged using boolean operations in solid modelling. The cross-sections of a particular vessel may however intersect with cross-sections of branching vessels, and the geometry at these positions has to be approximated. Other works [15–19] also reconstructed 3D surface models of the vessels from 2D contours extracted from image slices. This sometimes requires positioning and orienting the 2D contours according to the medial axis of the vessels, and then curve and surface interpolations are used to approximate and reconstruct the surface models. Such 2D contour extraction techniques do not provide control over 3D smoothness.

A simple thresholding technique was used in [20] to extract the binary image of the vessels, and the vascular model was reconstructed using an iso-surface algorithm. The thresholding technique however does not consider the spatial characteristics of the image, and it is sensitive to image noise and inhomogeneous intensity. In [21, 22], region growing algorithms were applied to segment the vascular structures from CT and MR angiography data. The region growing techniques are, in general, sensitive to noise, and can often lead to non-contiguous regions and over-segmentation. In addition, thin structures are often not extracted due to variations in image intensities. The watershed transform was used in [23, 24] to extract the geometry of the carotid arteries. In this approach, the image is interpreted as a landscape or topographic surface, with the pixel intensity representing the elevation of the topographic surface. The regions of local minima on a topographic surface are partitioned by considering water flow towards such regions. The watersheds are the lines that partition these regions. In this way, the image is partitioned into homogeneous regions with the watersheds defining the boundaries of these regions. The watershed transform tends to be sensitive to noise and often produces over-segmentation. It is also difficult for the watershed techniques to extract thin structures and weak object edges.

In [25, 26], a 3D dynamic surface model was used to delineate the boundary of carotid arteries. An initially triangulated model was placed within the interior of the carotid vessels, and an inflation force was applied to deform the model towards the vessel wall. In this method, the inflation force is applied only when the vertices of the model are within the lumen, i.e. at locations with image intensity below a user-specified threshold. An image-based force is further applied to the surface model to better localize the boundary. It may, however, be difficult to select an appropriate threshold value that delineates the vessel wall closely due to inhomogeneous image intensity. This approach is sensitive to noise, and manual editing is often required to move the vertices towards the vessel wall. In [27], a 2D discrete dynamic contour was first used to extract the vessel contours and a dynamic surface model was then inflated to reconstruct the surface model using the binary images of the extracted contours. This however does not consider the 3D geometric information from the image data set. In [28–30], the surface models for each of the vessel branches of the carotid artery were reconstructed independently using a tubular deformable model. Thus, such a method requires a surface merging algorithm to reconstruct the surface model from the triangulated surfaces of the vessel branches. This particular approach requires the axis of each of the vessel branch, which can be established manually by selecting a number of points from image slices to represent the curves of the structure. Owing to the inherent smoothing effect of this technique, high curvature regions such as bifurcations or stenosis may not be modelled accurately. These explicit deformable models represent contours and surfaces parameterically, which requires the tracking of points on

DOI: 10.1002/cnm

the curves and surfaces during deformation. It is therefore difficult for explicit deformable models to deal with topological variations and complex shapes.

Implicit deformable models have been applied in the segmentation of vascular structures in [31–35]. However, many of these techniques use an attraction force field which acts on contours or surfaces only when they are close to the object boundaries. As such, initial contours have to be placed close to the object boundaries, which can be tedious in complex geometries. A constant pressure term such as the one in [36] is often used to monotonically expand or shrink the deformable model towards the image object boundaries, which can easily overwhelm weak object boundaries. In addition, the external force fields used are unidirectional, thus requiring the initial contours to be placed either inside or outside object boundaries, which can be difficult for compact and narrow structures. Many of these techniques are also sensitive to image noise, and have difficulties in extracting deep boundary concavities.

The implicit deformable model [37–39], based on the dynamic interactions between geometries, used in this work can eliminate many of the drawbacks of the existing methods. In particular, the external force field is based on hypothesized geometrically induced interactions between the deformable model and image object boundaries, and is referred to as the *geometric potential force* (*GPF*) *field* [37, 38]. The dynamic geometric interaction force changes according to the relative position and orientation between the geometries of the deformable model and image object. This greatly improves the performance of the deformable model in handling image noise, complex geometries and extreme boundary concavities such as those found in vascular structures. The dynamic vector force proposed in [38] also allows the deformable model to have a strong invariance to initial configurations. In addition, this deformable model can easily deal with arbitrary cross-boundary initializations and weak edges due to the bidirectionality of the external force field. The GPF method is briefly presented in Section 2 along with an example.

1.2. Domain discretization

Despite a significant effort in developing robust patient-specific meshing methods, generation of a valid mesh automatically and rapidly, even today, is a challenge. A valid mesh for cardiovascular flows should be sufficiently fine to capture WSS with minimal error associated with point distribution. Using a coarse and purely unstructured mesh can produce a WSS distribution that is far from a converged solution [40]. Although mesh convergence is not always straightforward to carry out, designing a mesh by taking into account all the necessary factors, including boundary layer [40, 41], is essential to obtain results with high level of confidence.

In a subject-specific modelling pipeline, such as the one proposed here, it is essential to have the smooth transfer of image segmentation results to the meshing stage. In the majority of works reported, this link is often not well defined with some exceptions [29, 35]. Subject-specific meshing is a growing area of research, and currently, there is no universal way of satisfying all the meshing requirements via a single algorithm. Thus, the area is filled with a diverse number of approaches [32, 35, 42–45]. From the available literature, it is clear that the unstructured volume generation currently poses little or no challenge. It is apparent, however, that the interface between the image processing and a valid surface generation, and boundary layer meshing are not satisfactorily addressed. Thus, these two areas are given attention in this work.

Many existing mesh generation methods developed for subject-specific modelling are generally an extension of methods developed for standard engineering applications. In engineering applications, the object boundary can be rigorously defined and described analytically or piecewise analytically, to ensure a pre-defined size and quality of the surface elements. The NURBS patches method is one of the frequently used approaches for complex shapes. A well-defined boundary allows such a method to simplify the construction of a surface mesh and the corresponding cosmetics. This is due to the fact that points can be easily created or moved along the surface in order to improve element quality. However, in patient-specific medical objects, the surface is not well-defined and it may not be easily described analytically without compromising important surface features. Hence, alternative approaches have been used and often essential to approximate the surface of a medical object. The meshing approaches used for constructing surface meshes of medical objects are summarized into four categories below. The first category needs no initial mesh. The remaining three methods require generation of an initial mesh to approximately define the object boundary. The easiest way to build such an initial mesh from a binary object is via the Marching Cubes (MC) method [46]. The MC method produces a consistent surface mesh but such a surface may not be smooth. This mesh may have a significant number of bad quality triangles, which are unlikely to be of the required size. The four categories of the surface meshing methods are described as:

Category (1): Standard NURBS patches may be used [44, 47] as in engineering applications. In such a method, the surface is divided into curvilinear quadrilateral patches. The parameters of the patches (control points) can be tuned to minimize the distances between the initial mesh nodes and the surface. Since the surface is described by NURBS patches, all advantages of the surface mesh generation methods, developed for standard engineering applications, can be employed to the medical geometries. Hence, the challenge here is to develop simple, robust and accurate methods for finding the NURBS parameters for geometries constructed from scans. The surface mesh can then be generated using, for example, an advancing front technique (AFT). Although this method is robust for standard engineering applications, the risk of losing geometrical features is very high and the method can be tedious.

Category (2): In regridding methods [48], the surface mesh with proper element size distribution is built by the AFT. Here, the surface nodes are placed directly on faces of the initial mesh. After a constituent mesh has been generated, the node can be repositioned in order to meet surface smoothness, for example, by the second and higher order interpolation of the surface defined by the initial grid. The challenge here is to develop method that is robust at geometrical abnormalities, such as high curvature parts of the mesh. A set of improvements are proposed in [48] to enhance the robustness of the method.

Category (3): In mapping methods [49], a harmonic mapping of the surface onto a plane domain is performed. The initial mesh is used to compute the local metric of the mapping. The plane domain is triangulated by a high-quality 2D mesh satisfying local sizing criteria defined by the metric. The 2D mesh is then mapped back to the medical object boundary. In elongated objects, such as blood vessels, the element sizing of the plane mesh can vary by many orders. Then the accuracy can be lost because of computation errors. In such cases, the object boundary must be divided into smaller patches to perform the mapping correctly. A simple method of this kind is described in [50]. The mapping of 2D patches comes with additional approximations and it can be time consuming.

Category (4): The fourth type of methods are based on the fact that an initial mesh is already a consistent one and approximately represents the surface. Thus, only mesh-cosmetic operations are required to convert it into a valid mesh for computation. Basic surface mesh cosmetic methods are: edge swapping, edge splitting, or contraction, in combination with smoothing of the nodal distribution [51, 52]. The main difficulty here is to keep the point positions on the surfaces that are not analytically defined. For example, Laplace smoothing [53], a good method for 2D meshes, results in reduced object volume if applied to a 3D surface mesh. Therefore, the Taubing smoothing is preferred [54], which is known to preserve the object volume for spherical shapes. Nevertheless, for cylindrical objects (such as blood vessels), its repeated application may change the surface definition. However, a constrained Taubing smoothing, such as the one proposed in this work, preserves the accuracy of the original surface definition within a prescribed error tolerance.

As seen, the fourth meshing option is also not fully foolproof, but it lends itself to universality with minimal inter user variability. This is especially important for really complex passages such as human nasal cavities [55]. The surface mesh construction method is explained in full detail in Section 3.

Before a fully unstructured volume mesh is generated to fill the inner parts of an artery, it is essential to determine the need for the so-called *boundary layer mesh* in the form of semi-structured layers close to the boundaries. It is increasingly apparent from the available clinical and modelling studies that the link between WSS, oscillatory shear index (OSI) and atherogenesis are very strong

DOI: 10.1002/cnm

[56–60]. Thus, to understand placque initiation and progression, accurate estimation of WSS is essential. One of the major uncertainties in the WSS calculations is associated with the mesh. Converging the time-dependent WSS is not trivial and all efforts should be undertaken to make sure that the WSS calculated has little or no error from the domain discretization. Our previous studies [40] clearly show that the boundary layer mesh, even at low Reynolds numbers, is often essential. Thus, an accurate and fast way of constructing a boundary layer mesh is essential in arterial fluid dynamics.

A small number of attempts have been made in the past to generate boundary layer meshes [32, 61–63]. In [32], a medial axis-based approach was employed to fill a boundary layer with 18 node quadratic wedges. In [63], a boundary layer mesh was generated by a layer-by-layer loop. This may be expensive due to the number of checks carried out when generating every layer. An adaptive method used in [62] may be computationally more expensive than a preassigned boundary layer mesh. In the majority of the studies reported, the inward normal in some form or other was employed. We also employ the surface inward normals for generating prisms in the first instance, and then, the prisms are divided into smaller prisms using a geometrical progression approach, before they are split into tetrahedrons. As explained later in Section 4, the prism size are defined based on the surface mesh and the local diameter or feature size.

Once the boundary layer construction is completed, the remaining unstructured volume meshing is easily carried out by Delaunay mesh generation (see Section 4). The fastest method is the Delaunay point insertion-based method (e.g. [64]) with subsequent 3D mesh cosmetics based on 3D swapping and Laplace smoothing [65]. The centroidal Voronoi tessellation (CVT) method based on Lloyds [66] iteration may also be used for 3D mesh cosmetics [67].

All different aspects associated with the mesh generation are described in detail in Sections 3 and 4. The subtopics of these sections include a valid surface mesh generation, determining the central axis of the artery using a voxel thinning algorithm, cutting the vessels orthogonal to the axes, boundary layer and inner volume construction and different methods of cosmetics required to produce a high-quality mesh.

1.3. Flow solver

The analysis of blood flow through subject-specific arterial geometries is one of the widely researched topics of patient-specific modelling. Fluid flow modelling through static geometries is more or less well established except for large-scale simulations and extremely complex geometries. However, correct application of flow boundary conditions is still an issue. In Section 5, we devote a significant part to discuss the flow boundary conditions. Since any established incompressible flow solver is sufficient to carry out the flow calculations in a static geometry, a review of the methods is not provided here. We employ a matrix-free dual time stepping method in an optimized parallel environment to carry out the calculations [55, 68–72]. The flow equations and the flow solver used here are briefly presented in Section 5.

On the other hand, topics such as fluid–structure interaction has also been studied widely but with marginal success [73–84]. The main problem in fluid–structure interaction studies is the lack of availability of subject-specific, *in vivo* material properties and difficulties associated with imaging and segmentation of arterial walls. One way of overcoming the material property issue is via transiently registering the image and imposing the wall motion as a boundary condition. However, if accurate wall stress distribution is of interest, imposing wall motion may not be the solution.

1.4. Proposed pipeline

While recognizing the fact that intensive research is being carried out in the area of subject-specific modelling, the procedure developed by various groups on individual components may not be easily amalgamated. The routine procedure of segmentation and meshing currently involves intensive user interaction. Although automation of the pipeline heavily depends on the quality of clinical data available, we can at least attempt to produce robust methods that can lead to full automation in the near-future. With this in mind, we have applied a new 3D segmentation method

DOI: 10.1002/cnm

to reconstruct the vascular geometries here [37, 38]. This method is suitable for both MR and CT scanning modalities and uses the level set method [85, 86], which allows topological changes to be handled automatically (Section 2). Once the image reconstruction is completed, an automatic voxel thinning procedure is employed to obtain the axes of the vessel and vessel branches (see Section 3.1). This allows us to automatically truncate the vessels perpendicular to their axes. Special attention is paid in developing a procedure for valid surface mesh generation, using the level set functions, obtained from the image segmentation method (see Section 3.2). An algorithm for cropping the surface mesh is discussed in Section 3.3. All the smoothing methods and the pros and cons of using MC and improvements are discussed in Section 3.4. The surface mesh generation is followed by an automatic procedure to construct the boundary layer mesh on the inner surface of the vessel walls (see Section 4.1). The volume mesh, based on the Delaunay method, is then used to complete the mesh generation step (Section 4.2). Once the mesh is finalized, the flow boundary conditions are decided based on either available flow measurements or assumed flow rate. An accurate reflection of realistic flow values are essential and the method of extracting such conditions from measurements is discussed in detail in Sections 5.1–5.5. The method used for flow solution is given in Section 5.6. The example problem studied is the blood flow through a carotid bifurcation with a severe stenosis (Section 6). The inlet and exit flow conditions may be obtained from ultrasound measurements and in cases where only inlet conditions are available, we have two choices. The first choice is to assume a flow division between the external and ICAs and the second is to employ a Windkessel type of element at the exit. Neither of them are a replacement for an accurate flow measurement. Once the boundary conditions are determined, the flow solver (a finite element (FE) solver is used in the present study) is used to determine the flow and wall quantity distributions, both with respect to time and space.

2. BLOOD VESSEL GEOMETRY EXTRACTION

2.1. A Deformable Model

Extracting geometry from 3D scan data is a pre-requisite for many computational modelling applications in cardiovascular fluid dynamics. Ideally, it should be carried out with minimum user interference and should robustly handle variations and noise interference that exist in the imaging data. Here, we adopt our recently developed level set-based image segmentation technique [37, 38] to extract arterial geometries. An overview of this method and implementation details are given below. Let $I(\boldsymbol{x})$ denote the 3D grey image, where $\boldsymbol{x}=[x, y, z]^{\mathrm{T}}$ is a point in the image domain $\mathscr{D}$. Let Ω be an object to be segmented from this image. We employ the level set method in which the object boundary $\partial\Omega(t)$ is defined through the level set function $\Phi(\boldsymbol{x}, t)$:

$$\partial\Omega(t)=\left\{\boldsymbol{x}:\Phi(\boldsymbol{x}, t)=0\right\}. \tag{1}$$

To compute the level set function, we numerically solve the following PDE for $\Phi(\boldsymbol{x}, t)$ [37, 38]:

$$\frac{\partial\Phi}{\partial t}=\alpha g(\boldsymbol{x})\kappa(\boldsymbol{x}, t)\|\nabla\Phi\|-(1-\alpha)(\boldsymbol{F}(\boldsymbol{x})\cdot\nabla\Phi) \tag{2}$$

where α is a tuning parameter, $g(\boldsymbol{x})=1/(1+\|\nabla I\|)$ is an edge stopping function, $\kappa(\boldsymbol{x}, t)=\nabla\cdot(\nabla\Phi/\|\nabla\Phi\|)$ is the mean curvature of the surface $\Phi=$const, $\boldsymbol{F}(\boldsymbol{x})=[F_x, F_y, F_z]^{\mathrm{T}}$ is the flow function determined by image I. The flow function, which is derived from the image data, acts as an external force on the deformable model, and it is the most critical component in this kind of deformable model formulation.

2.2. Geometrical potential force (GPF)

In our approach, the GPF [37, 38] used is defined as:

$$\boldsymbol{F}(\boldsymbol{x})=\pm\frac{\nabla\Phi(\boldsymbol{x})}{\|\nabla\Phi(\boldsymbol{x})\|}G(\boldsymbol{x}) \tag{3}$$

DOI: 10.1002/cnm

where the scalar potential G is represented by the convolution of the form

$$G(\boldsymbol{x}) = \text{P.V.}\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|^{\lambda+1}} * \nabla I(\boldsymbol{x}) = \text{P.V.}\iint_{\boldsymbol{x}'\in\mathscr{D}} \frac{\boldsymbol{x}-\boldsymbol{x}'}{\|\boldsymbol{x}-\boldsymbol{x}'\|^{\lambda+1}} \cdot \nabla I(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' \tag{4}$$

Here P.V. denotes the principle value and parameter λ depends on the domain dimension: $\lambda = 3$ for a 3D domain and $\boldsymbol{x}'$ indicates a deformable model. Force, $\boldsymbol{F}$, acts in the direction of the gradient of Φ, i.e. always normal to the active surface $\Phi(\boldsymbol{x}, t) = 0$. It is possible to show that in the 2D case, with $\lambda = 2$, this force coincides with a magnetic-active force [39], but in contrast to magnetic force the GPF can be defined and applied to images of any dimension (even to 4D, i.e. time varying 3D scans).

A discrete analogue of the convolution kernel in Equation (4) takes the form

$$\text{P.V.}\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|^{\lambda+1}} = \begin{cases} \boldsymbol{x}/\|\boldsymbol{x}\|^{\lambda+1}, & \boldsymbol{x} \neq \boldsymbol{0} \\ \boldsymbol{0}, & \boldsymbol{x} = \boldsymbol{0} \end{cases} \tag{5}$$

The fastest way to evaluate convolution (Equation (4)) and compute the potential $G_{ijk} = G(x_i, y_j, z_k)$ is to apply the 3D fast Fourier transform (FFT) method. It can be computed prior to the level set function calculation.

2.3. Numerical solution, initial and boundary conditions

Equation (2) is solved numerically by the finite difference method. Let $\boldsymbol{x}_{ijk} = [x_i, y_j, z_k]^{\mathrm{T}} = [i\Delta x, j\Delta y, k\Delta z]^{\mathrm{T}}$ be the grid point, where Δx, Δy and Δz are the grid sizes along the x, y and z directions, respectively (i.e. voxel size in every dimension). Let Δt be a time step and $t_n = n\Delta t$, where $n \geqslant 0$. Having defined all the necessary quantities, the level set function $\Phi(x, y, z, t)$ is given on a time–space grid as:

$$\Phi^n_{ijk} = \Phi(x_i, y_j, z_k, t_n)$$

The forward Euler method is employed here for advancing in time, i.e.

$$\Phi^{n+1}_{ijk} = \Phi^n_{ijk} + \Delta t\{\alpha(g\,\kappa\|\nabla\Phi\|)^n_{ijk} - (1-\alpha)(\boldsymbol{F}\cdot\nabla\Phi)^n_{i,j,k}\} \tag{6}$$

In the curvature term $(g\kappa\|\nabla\Phi\|)$, all the derivatives: $\partial\Phi/\partial x$, $\partial I/\partial x$, $\partial\Phi/\partial y$, etc. are discretized using a central difference approximation: $(\partial\Phi/\partial x)_{ijk} \approx [\Phi_{i+1,j,k} - \Phi_{i-1}, j, k]/(2\Delta x)$, and so on. In the vector force term $(\boldsymbol{F}\cdot\nabla\Phi)$, the derivatives in computing $\boldsymbol{F}$ by Equation (3) are also discretized by the central difference approximation, whereas the derivatives in the second multiple in Equation (4) are approximated using upwind differencing, i.e. backward difference if the corresponding component $\boldsymbol{F}$ is positive and forward difference otherwise. This can be expressed as,

$$\left(F_x\frac{\partial\Phi}{\partial x}\right)^n_{i,j,k} \approx \begin{cases} F^n_x\dfrac{\Phi^n_{i,j,k} - \Phi^n_{i-1,j,k}}{\Delta x} & \text{if } F^n_x > 0 \\ F^n_x\dfrac{\Phi^n_{i+1,j,k} - \Phi^n_{i,j,k}}{\Delta x} & \text{if } F^n_x < 0 \end{cases}$$

$$\left(F_y\frac{\partial\Phi}{\partial y}\right)^n_{i,j,k} \approx \begin{cases} F^n_y\dfrac{\Phi^n_{i,j,k} - \Phi^n_{i,j-1,k}}{\Delta y} & \text{if } F^n_y > 0 \\ F^n_y\dfrac{\Phi^n_{i,j+1,k} - \Phi^n_{i,j,k}}{\Delta y} & \text{if } F^n_y < 0 \end{cases}$$

etc.

To set the initial condition, we have to define the initial surface S_0 somewhere in the vicinity of the blood vessel being investigated. An initial level set function is defined as a signed distance $D[S_0t]$ from the initial surface S_0, i.e.

$$\Phi(x_i, y_j, z_k, 0) = D[S_0]$$

DOI: 10.1002/cnm

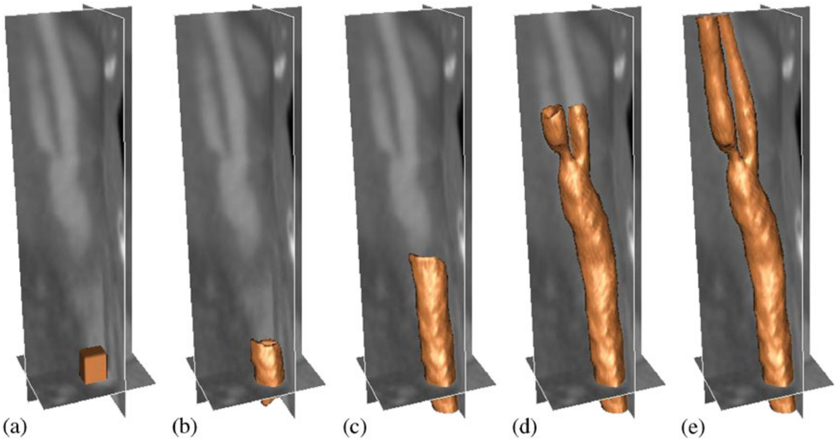

Figure 1. Segmentation of a carotid artery using GPF-based deformable model: (a) initial surface S_0; (b) after 11; (c) after 81; (d) after 187; and (e) after 241 time-steps.

The signed distance can be computed efficiently using the algorithm described in [87]. At the boundaries of the image domain, zero Neumann boundary conditions are imposed. The narrow band approach described in [86] is used to reduce the computational cost in updating the level set function.

During the calculation, function Φ may become irregular. Therefore, it is periodically re-initialized to the signed distance surface set from the current zero level set (surface of the deformable body), $S(t)=\{\boldsymbol{x}:\Phi(\boldsymbol{x},t)=\mathbf{0}\}$. Hence, Φ can be redefined as $\Phi(\boldsymbol{x},t)=D[S(t)]$, which is a smooth function with $S(t)$ being the surface with zero level set.

2.4. An example

In this section, a CT scan (DICOM data set) of a human carotid artery is used to demonstrate the GPF-based segmentation method. The progression of the level set function is depicted in Figure 1. Note that the initial model is positioned across the object boundaries to illustrate the capability of the deformable model to handle arbitrary cross-boundary initializations. The final solution is independent on the initial surface shape (example: sphere, parallelepiped, etc.), which has been tested for a wide range of different initializations [38]. The position of the centre of the initial surface S_0 is prescribed manually, which is the only manual operation in the segmentation process. In contrast, our experience with commercial or open-source software showed that substantially more manual interactions are required for the segmentation of blood vessels from MRI or CT scans of standard clinical quality.

3. SURFACE MESHING

The surface mesh generation procedure has the following steps. The level set functions calculated by the image segmentation method are used to determine the skeleton of the given artery first. This involves determining the approximate axis of the artery and its branches. This is followed by the procedure of generating a valid surface triangulation using again the final level set functions generated by the image processor. After carefully applying cosmetics to the surface mesh, it is clipped at the required length, orthogonal to the vessel axis. Following subsections explain the surface mesh generation steps in detail.

3.1. Skeletonization

In many blood flow applications, it is useful to determine the position of the blood vessel axes, i.e. creating a skeleton of the geometry. It helps in the automatic identification of the geometrical

DOI: 10.1002/cnm

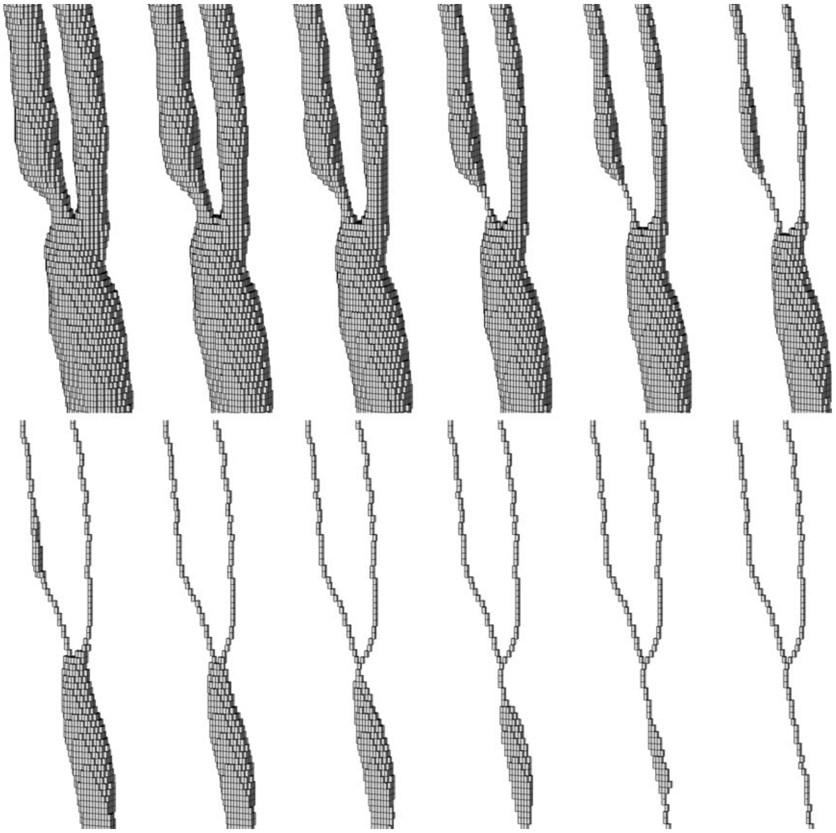

Figure 2. Voxel thinning process. Initial object voxels (top left) and final skeleton (bottom right).

features of the vessel, e.g. stenosis, aneurysms, bifurcations, kinks, etc. It is also helpful for determining the best orientation for the inlet/outlet surface. The following paragraphs explain a systematic way of obtaining a skeleton.

3.1.1. Voxel thinning. Since the image is given in terms of a voxel grid, voxel thinning is the simplest way for a topology-preserving skeletonization [88] (see Figure 2). A binary image can be easily created from the level set functions by allocating unity to the voxels in which level set function is positive (object or foreground points) and by setting the remaining voxels to zero (background voxels).

In the thinning algorithm used, the so-called *simple* points are identified and removed from the image until an image with only one voxel thickness is left along the axis. An object voxel is a *simple* point if the object topology does not change when the voxel is removed, i.e. removal of the point should not lead to disjoint objects (multiple components) and should not create/remove tunnels, cavities, etc. [89]. Several parallel and sequential thinning algorithms are available [88] and we choose reliability of the algorithm over speed as explained below.

Before listing the steps of the algorithm employed, some definitions should be mentioned here. Three main symmetric neighbourhoods for a voxel can be identified, i.e. (1) the set of voxels that share a face (2) the set of voxels that share an edge (3) the set of voxels that share a vertex. The voxels in these sets are referred to as 6-adjacent, 18-adjacent and 24-adjacent, respectively. Now, a 26-adjacent background component is a subset of 26-adjacent foreground voxels that are connected to each other (they form a sub-object). The thinning procedure can now be described in two main stages, i.e.

1. *Locate all boundary voxels*: A voxel is considered to be a boundary voxel if it shares a face with at least one background voxel. Such a voxel can be slid out from the object without the

DOI: 10.1002/cnm

removal of any neighbouring foreground voxels. While marking boundary voxels, we also compute and store the number of background voxels sharing a vertex with every boundary voxel (stored as an array, K). The next step is the removal of boundary voxels in a systematic fashion as discussed below.

2. *Voxel thinning*: Start the thinning process by considering the voxel that has the maximum K value first, and continue in the descending order to the smallest K value. Check whether a boundary voxel is *simple* by considering the $3\times3\times3$ image subset around it using the following steps:

   (i) Check whether its removal changes the number of 26-adjacent foreground components (change from joint to disjoint or disjoint to joint objects).
   (ii) If not, check whether its removal changes the number of 6-adjacent background components.
   (iii) If not, check whether its removal forms a tunnel, i.e. changes its genus g (number of tunnels, Equation (7)).
   (iv) If not, the voxel is *simple* and it is removed from the object and allocated to the background by setting a zero value.

Steps 1 and 2 are repeated and the process is terminated when all simple voxels are moved to the background. Note that the number of iterations required is proportional to the voxel thickness of the object.

The number of tunnels $N_{\mathrm{T}}=g$ of an image is calculated from the Euler characteristics χ that are related to the number of components N_{C} and number of cavities N_{CA} as:

$$\chi=2(N_{\mathrm{C}}+N_{\mathrm{CA}}-N_{\mathrm{T}}) \tag{7}$$

Although there are no cavities in the object considered in this study ($N_{\mathrm{CA}}=0$), the number of components is not zero and it can be computed. Thus, once χ is independently computed (see [90]), the number of tunnels may easily be estimated from Equation (7). A simple procedure for computing the Euler characteristics of a binary image is described in [90] and a slightly different approach is proposed in [91]. Both the methods give identical results. In both methods, a configuration of every $2\times2\times2$ voxel subset in the image is considered, from which the total Euler number is computed.

Typically, on scans of the blood vessels, the inlets and outlets appear at the boundaries of the image box, which may result in some difficulties regarding voxel thinning near the boundaries. The result of a thinning algorithm at the image boundaries depends upon how the neighbours of such boundary voxels are defined. Since there are no voxels beyond the boundary voxel, an additional virtual layer of voxels is added beyond the boundary voxel (Figure 3). Now boundary voxels will have 26 neighbours, of which 9 are virtual voxels.

In the existing studies, the virtual neighbour voxels are often treated as background voxels (Figure 3, left). Hence, all foreground voxels at the image boundary will be marked as boundary voxels and deleted after the first step of the thinning process. With further thinning, the subsequent voxel layers will be deleted and the vessel axis is shortened at the inlet/exit as shown in Figure 4(left). In our proposed approach (Figure 3, right), we mark a virtual voxel as a foreground voxel, if it shares a face with an image boundary voxel at which an inlet/exit exists. However, it is not deleted unless a real foreground voxel, sharing a face with it is deleted. Though the virtual voxels are not included in the final axis construction, they ensure that the axis is not shortened as shown in Figure 4(right).

3.1.2. Axis smoothing. After completing the voxel thinning process, the centres of the remaining voxels are connected to form a ragged curve. This curve is automatically generated in the main artery along with the axes of the branches, if applicable. The curve thus generated is tattered and requires smoothing, i.e. its points $\boldsymbol{p}_i$, which are initially at the centres of the skeleton voxels should be displaced to form a smoother line. A simple way to smooth the lines is to apply a 1D analogue

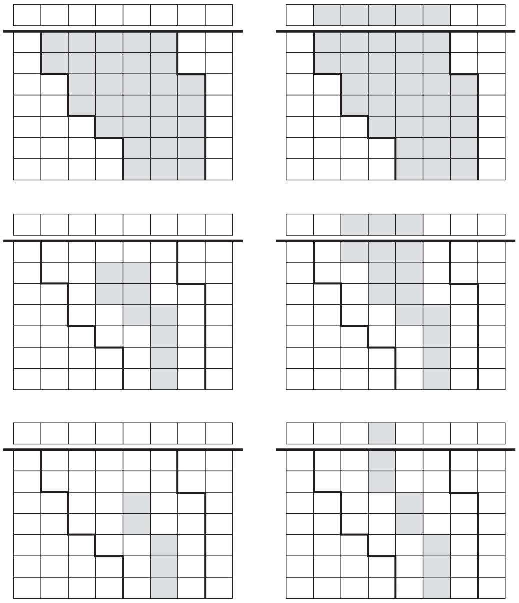

Figure 3. Addition of virtual voxels to the thinning process. Standard method (left) and proposed method (right).

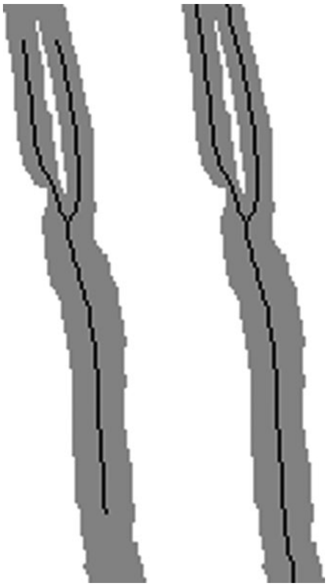

Figure 4. Shortened centroidal line obtained using the standard thinning method (left). The line is not shortened in our approach (right). Here projection of the binary image on the $\{x, z\}$-plane is shown by grey (initial image) and by black (after thinning).

of Laplace smoothing (see Equation (11)), i.e.

$$\boldsymbol{p}_i^{\text{new}} = \boldsymbol{p}_i(1-\lambda) + \tfrac{1}{2}(\boldsymbol{p}_{i-1} + \boldsymbol{p}_{i+1})\lambda, \quad i = 1, \ldots, M-1 \tag{8}$$

where M is the number of points on a curve and λ is a weighting parameter. While smoothing, the first point, $i = 1$, and the last point, $i = M$, are fixed. Note that, in the case of high curvature axis, this method systematically shifts the curve towards its curvature centre. Thus, the 1D analogue of the Taubing smoothing (see Equations (12)–(13)) may be a better option. Even with Taubing

DOI: 10.1002/cnm

smoothing, the curve will shift towards the centre of curvature, but at a much slower pace. To avoid any adverse effect of smoothing, a more robust smoothing method is proposed here based on the smoothing direction of the segments between two neighbouring points. The steps involved in the smoothing process are:

1. Compute the normalized tangent vectors of every segment: $\hat{\boldsymbol{t}}_i = (\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)/\|\boldsymbol{p}_{i+1} - \boldsymbol{p}_i\|$, $i = 1, \ldots, M-1$.
2. Compute the normalized smoothed tangent vector
$$\boldsymbol{t}_i^{\text{new}} = \boldsymbol{t}_i(1-\lambda) + \frac{1}{2}(\boldsymbol{t}_{i+1} + \boldsymbol{t}_{i-1})\lambda, \quad \hat{\boldsymbol{t}}_i^{\text{new}} = \boldsymbol{t}_i^{\text{new}}/\|\boldsymbol{t}_i^{\text{new}}\|, \quad i = 2, \ldots, M-1 \tag{9}$$
where λ is a weighting parameter and it is taken to be 0.5.
3. Compute the endpoint location of the ith segment rotated around its midpoint $\boldsymbol{q}_i = \frac{1}{2}(\boldsymbol{p}_{i+1} + \boldsymbol{p}_i)$ until its direction coincides with $\boldsymbol{t}_i^{\text{new}}$ (Figure 5). Thus, its endpoints take the positions $\boldsymbol{p}_{1,i}^{\text{new}} = \boldsymbol{q}_i - d_i \hat{\boldsymbol{t}}_i^{\text{new}}$, $\boldsymbol{p}_{2,i}^{\text{new}} = \boldsymbol{q}_i + d_i \hat{\boldsymbol{t}}_i^{\text{new}}$, where $d_i = \frac{1}{2}\|\boldsymbol{p}_{i+1} - \boldsymbol{p}_i\|$ are the segment half-lengths. The first and the last segments of the line are not rotated.
4. The new endpoints for every segment are approximated as $\boldsymbol{p}_i^{\text{new}} = \frac{1}{2}(\boldsymbol{p}_{2,i-1}^{\text{new}} + \boldsymbol{p}_{1,i}^{\text{new}})$, $i = 2, \ldots, M-1$ (Figure 5).

Steps 1–4 are repeated typically 4–5 times. The results of smoothing the centroidal lines are shown in Figure 6.

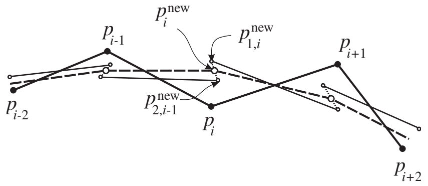

Figure 5. On the line smoothing algorithm. The bold solid line indicates the initial centroidal line, its nodes are shown by closed circles. The bold dashed line indicates the centroidal line after one smoothing step, its nodes are shown by large open circles. Segments rotated around their midpoints are indicated by thin solid lines, their endpoints are shown by small open circles.

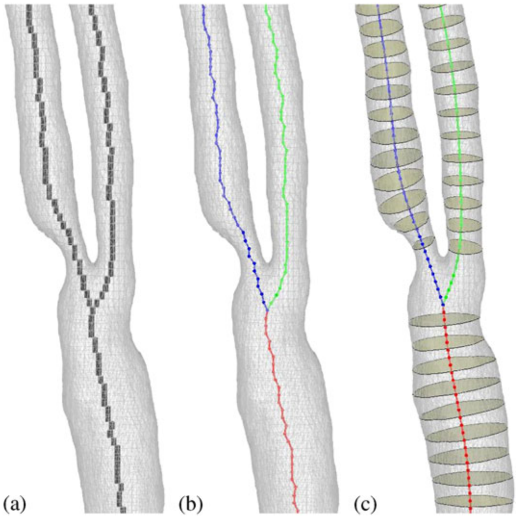

Figure 6. Axis smoothing: (a) Initial voxel skeleton; (b) the initial centroidal lines, circles indicate voxel centres; and (c) the centroidal lines after the smoothing. The planes normal to the lines are shown in yellow. View this figure color online at [10.1002/cnm.1446].

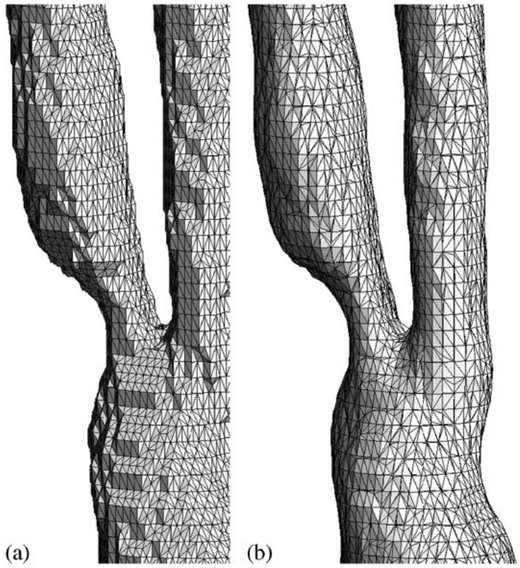

Figure 7. Surface mesh generated by marching cube methods: (a) standard method and (b) advanced method [92].

3.2. Obtaining a consistent surface mesh

Representation of the object surface through a continuous level set function gives an opportunity to employ a advanced MC algorithm as described in [92]. This gives a much smoother surface than the standard MC method [46], without the ambiguities that appear when the standard method is applied (e.g. [93]). The advanced MC method used here gives an isosurface $\Phi(\boldsymbol{x})=0$ on the voxel grid points and linearly interpolates between them, i.e. using piecewise linear functions. In contrast, the standard MC method gives the isosurface of the linearly interpolated non-smooth function $H[\Phi(\boldsymbol{x})]=0.5$, where $H(.)$ is the unit-step Heaviside function. Figure 7 shows the differences between the standard MC method and the method used in this study. As seen, the advanced MC method used here shows a much better quality surface mesh.

3.3. Cropping the surface

Once a satisfactory surface representation is established and the geometry skeleton is generated, the next step is to obtain the valid inlet and exit boundary surfaces. Often clipping is carried out visually but in this work, we provide an automatic cropping framework which uses the surface mesh generated in the previous section and the skeleton constructed in Section 3.1.

3.3.1. Determining the position of inlets/outlets. After smoothing the skeleton, the tangent of the centroidal line varies only slightly from point to point allowing easy determination of cross-sections normal to the centroidal line.To crop the surface mesh, we should indicate a point on the skeleton curve first. This can be done manually by mouse or automatically, say, at a certain distance from a bifurcation point or any other topological change. It is also possible to determine the position that gives the maximal blood vessel length for a given scan. To determine this, the normal cross-section of a closest axis point to the image-box boundary that does not intersect the image-box boundary (Figure 8) should be estimated. The point at which this closest cross-section intersects the axis, $\boldsymbol{P}_k$, is determined by going through the axis points one by one from the edge of the image box.

After point $\boldsymbol{P}_k$ (corresponding to the kth inlet/outlet) on the central line is found, we can compute the tangent $\boldsymbol{t}_k$ to the line at this point, for example, using

$$\hat{\boldsymbol{t}}_i=(\boldsymbol{p}_{i+1}-\boldsymbol{p}_{i-1})/\|\boldsymbol{p}_{i+1}-\boldsymbol{p}_{i-1}\|, \quad i=2,\ldots,M-1. \tag{10}$$

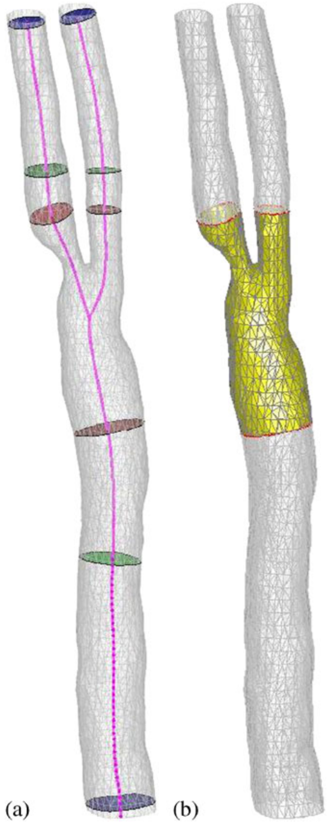

Figure 8. Cropping the surface mesh: (a) automatic determination of inlet/outlet position with different cropping planes and (b) cropped mesh.

The plane $\Pi_k = \{\boldsymbol{x} : (\boldsymbol{x} - \boldsymbol{P}_k) \cdot \boldsymbol{t}_k = 0\}$ passing through $\boldsymbol{P}_k$ normal to $\boldsymbol{t}_k$ is defined for kth inlet/outlet plane.

3.3.2. Cropping the surface mesh. Once the inlet/exit planes are defined, the surface mesh is cropped using the following steps.

1. For every kth inlet/outlet, find the mesh points located on the domain side of interest from the plane defined by Π_k, satisfying the inequality $(\boldsymbol{p} - \boldsymbol{P}_k) \cdot \boldsymbol{t}_k \geqslant 0$ ($\leqslant 0$ for outlets) and flag them.
2. Mark the faces sharing both the flagged points and the points outside the domain.
3. Compute the distances from point $\boldsymbol{P}_k$ to the first vertex of the marked triangles and select the triangle with the smallest distance denoted by f_0 to start the splitting process.
4. Split the selected triangle into three subtriangles using the intersection points created by the cutting plane as new vertices as shown in Figure 9(a)–(b). As seen, the intersection creates a triangle and a quadrilateral. The triangle is retained and the quadrilateral is split into two triangles.

   Remark: every intersection point will be added to the mesh as two different nodes p_4 (filled half-circle) and p'_4 (open half-circle) located at the same position $\boldsymbol{p}_4$. However, only point p_4 forms the part of the final mesh. The same applies to p_5 and p'_5. Thus, if the initial triangle $\{p_1, p_2, p_3\}$ has one point on the domain surface, e.g. p_1 (Figure 9(a)), then triangle $\{p_1, p_5, p_4\}$ is introduced, all vertices of which become the domain surface nodes. The remaining quadrilateral $\{p_2, p_3, p'_4, p'_5\}$ (all vertices of which are outside the domain) is split into two triangles. The choice how to draw a new side: $\{p_2, p'_4\}$ or $\{p_3, p'_5\}$ is made

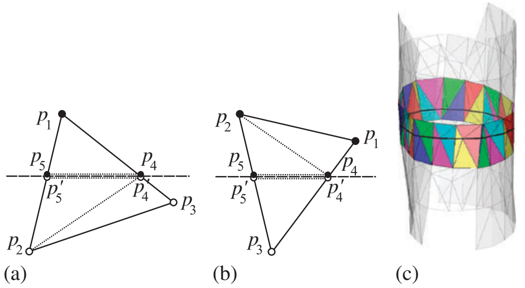

Figure 9. Truncating a surface mesh. (a)–(b) Splitting faces intersected by plane Π_k (dashed line). Marked points are indicated by closed circles, unmarked—by open ones. New sides are indicated by dotted lines. (c) Mesh intersection with plane Π_k (bold black). Marked triangles are indicated as colored (every two neighbouring triangles have different colors). Every triangle is divided into three subtriangles to form a side exactly lying at the intersection plane. View this figure color online at [10.1002/cnm.1446].

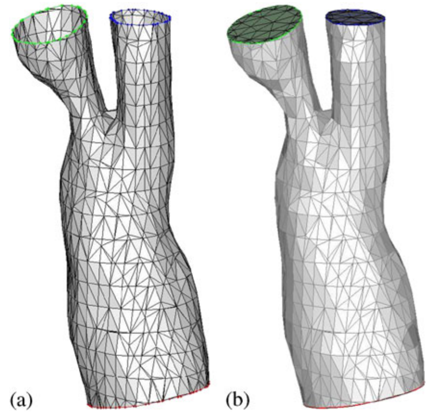

Figure 10. Surface mesh cropping: (a) unclosed mesh after trimming and (b) closed mesh after the triangulation of the inlet and outlet surfaces.

to minimize the maximal angle of two newly formed triangles. In the example shown in Figure 9(a), new faces added are $\{p_2, p_4', p_5'\}$ and $\{p_2, p_3, p_4'\}$. The triangle with two marked vertices on the domain side of interest can be split in an analogous manner as shown in Figure 9(b).

5. Take an edge of the treated triangle in step 4 with a node on the retained surface and another on the outside surface ($\{p_1, p_2\}$ in Figure 9(a) and $\{p_2, p_3\}$ in Figure 9(b)) and find the conjugate triangle containing the same side. If this triangle is f_0 then stop, otherwise go to step 4 and continue.

After this procedure is repeated for every outlet, a mesh is obtained consisting of disjoint components. The triangular elements belonging to the mesh side of interest (shown in yellow in Figure 8(b)) can be identified as those that are flagged. After removing the mesh parts that are outside the surface of interest (shown in grey in Figure 8(b)), an unclosed surface mesh is obtained with the edges on the unclosed side lying on the same plane for every inlet/outlet (Figure 10(a)). The inlets/outlets are then triangulated by any 2D triangulation method. We employ the ‘stitching’ method described in [94]. As a result, a closed mesh is created with flat inlet/outlet surfaces as shown in Figure 10(b).

It is often useful to smooth the contours of the inlets and outlets before the mesh cosmetics procedure is applied. One of the most effective method that does not cause contour shrinkage is the same method which is used to smooth the skeleton lines as described above. An example of such smoothing is shown in Figure 11.

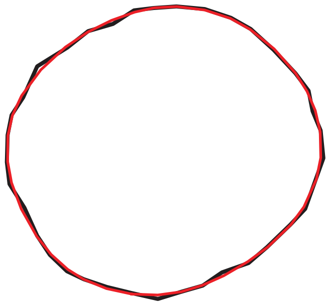

Figure 11. Smoothed outlet contour after 20 steps of smoothing.

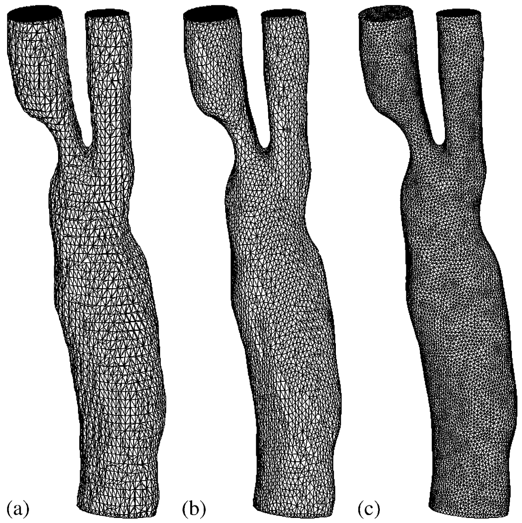

Figure 12. Surface meshes: (a) mid-cut mesh after before mesh cosmetics; (b) after 10 steps of the Taubin smoothing; and (c) after edge splitting/contraction.

3.4. Surface mesh cosmetics

The mesh obtained by the advanced MC method (shown in Figure 12(a)) always contains a large number of ill-shaped elements, i.e. with small aspect ratio defined as $r =$ smallest height/longest side. In addition, the triangular element size can be smaller or larger than the element size required by the flow solver. Therefore, the mesh needs some mesh cosmetics, which creates a surface mesh for the same domain with much better element quality (higher $r_{\min}$) and at a prescribed element size. It is important to note that there are essentially three procedures used to improve the mesh quality. They are:

(i) Mesh smoothing (no topological changes)
(ii) Splitting/contraction of edges
(iii) Edge swapping

Note that these are all standard techniques in mesh cosmetics, but the implementations can be different. Furthermore, these procedures are different for local and global operations or mid-domain and near-ridge applications. Hence, in the following subsections we explain the procedures chosen specifically for patient-specific mesh generation.

3.4.1. Surface mesh smoothing. As a first step in cosmetics, we smooth the surface mesh by moving the nodes along the surface (without changing the mesh topology) in order to increase the element quality (aspect ratio). Note that the constrained Taubing smoothing is preferred here over

DOI: 10.1002/cnm

the more commonly used Laplace smoothing. In standard Laplace smoothing, the position of the point p is redefined by

$$\boldsymbol{p}^{\text{new}} = \frac{1}{d_p} \sum_{i \in \mathscr{P}_p} \boldsymbol{p}_i \tag{11}$$

where $\mathscr{P}_p$ is the set of points contiguous to p, d_p is the number of points in $\mathscr{P}_p$, p is the node numbering index in the mesh, and $\boldsymbol{p}$ its co-ordinates. All points are consecutively moved into their newly computed positions. This iteration may be repeated several times until a specified element quality is reached. However, it is obvious that Laplace smoothing of a convex mesh will monotonically shrink it as the new point position always lies inside the domain.

In the Taubing smoothing technique, the new position is computed in two steps within every iteration, i.e.

$$\boldsymbol{p}^{\text{inter}} = (1-\lambda)\boldsymbol{p} + \lambda \frac{1}{d_p} \sum_{i \in \mathscr{P}_p} \boldsymbol{p}_i \tag{12}$$

$$\boldsymbol{p}^{\text{new}} = (1-\mu)\boldsymbol{p}^{\text{inter}} + \mu \frac{1}{d_p} \sum_{i \in \mathscr{P}_p} \boldsymbol{p}_i^{\text{inter}} \tag{13}$$

where parameters λ and μ satisfy the equation $\lambda^{-1} + \mu^{-1} = \varkappa$. We take $\lambda = 0.6307, \varkappa = 0.1$ then $\mu = -0.6732$ as recommended in [54]. To avoid adverse effects, the following constraint is also added to the Taubing smoothing. If the new position of a surface point is at a distance greater than some maximally allowed value, $h_{\max}$, then the point is displaced by $h_{\max}$ in the direction given by Equations (12)–(13), i.e.

$$\boldsymbol{p}^{\text{new,constr.}} = \boldsymbol{p} + h_{\max} \frac{\boldsymbol{p}^{\text{new}} - \boldsymbol{p}}{\|\boldsymbol{p}^{\text{new}} - \boldsymbol{p}\|} \quad \text{if } \|\boldsymbol{p}^{\text{new}} - \boldsymbol{p}\| > h_{\max} \tag{14}$$

In this work the maximal distance $h_{\max}$ was set as half the voxel size. The mesh after 10 steps of smoothing is shown in Figure 12(b). Note that the Taubing smoothing protects the shape of ball-like objects, but too many iteration steps can disturb cylinder-like objects such as blood vessels.

3.4.2. Topology-based edge swapping. Improving the mesh topology is very likely to have a beneficial effect on the quality of the elements in a mesh (if combined with smoothing). To improve the mesh topology, every edge containing a new point from the splitting/contraction procedure is considered. Let d_i indicate the nodal indices of the nodes of the two triangles that share this edge (4 nodes in total) with i denoting the ith point. The optimal nodal index for a point in a triangular surface mesh is 6 and the aim of the topology based edge swapping is to reduce the deviation of d_i from this number. Hence, for every edge swap, the change of mesh relaxation index, ΔU, is evaluated in accordance with

$$\Delta U = [(m_1-1)^2 + (m_2-1)^2 + (m_3+1)^2 + (m_4+1)^2] - [m_1^2 + m_2^2 + m_3^2 + m_4^2] \tag{15}$$

where $m_i = d_i - 6$ is the deviation from the ideal nodal index [95]. Here, subscripts 1 and 2 are the endpoints of the edge before swapping and 3 and 4 are the endpoints of the edge after swapping. Note that, an edge will only be swapped if $\Delta U < 0$. This procedure is repeated several times and is terminated when swapping of the edges connected to a new point results in an increase in the mesh relaxation index. As mentioned, it is essential to perform local Taubing smoothing to optimally benefit from the topological changes.

3.4.3. Edge splitting/contraction. If the mesh contains a significant number of very short or/and very long edges and the average edge length is not sufficiently close to some predefined reference length as required by the flow solver, then edge splitting and contraction [51, 52] are very useful procedures.

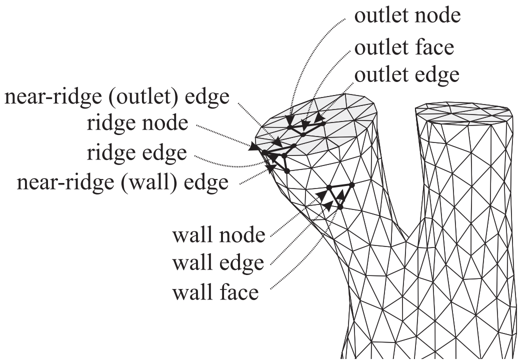

Figure 13. Meshing terminologies. Outlet faces are shown in grey.

If an edge $\{p_1, p_2\}$ is very short, it can be contracted into a single node p^{new}. The nodal index d_{new} (number of contiguous nodes) of the new node p^{new} is often significantly larger than the optimal value of 6 ($d_{\text{new}} = d_1 + d_2 - 4$ where $d_{1,2}$ are nodal indexes of endpoints p_1, p_2 of the initial edge). In such cases, the local mesh topology is improved by swapping edges attached to node p^{new}. This procedure is described in Section 3.4.2.

If the edge is too long, it can be divided by inserting a new node p^{new} approximately at the midpoint of the edge. The resulting nodal index is $d_{\text{new}} = 4$, which is less than optimal. In this procedure the local mesh topology is improved by swapping edges surrounding the node p^{new}. In both edge contraction and splitting, the biggest challenge is placing the new point on the domain surface, due to the fact that the object boundary is not explicitly defined. A simple and effective formula for placing a new point is described in Appendix B. The following paragraphs explain the edge splitting and contraction of ridge and near-ridge edges. The meshing terminologies used here are shown in Figure 13.

The edges that are shared by both a wall face and an inlet/outlet face are ridge edges. Their endpoints are ridge points. During edge contraction, a newly created point (which is a ridge point) is positioned in an identical fashion to the procedure described above and in Appendix B. The only difference is that the normals to the endpoints are computed by averaging the normals to the wall face containing the given ridge point. This normal is then projected onto the plane of the inlet/outlet surface when required. The mesh topology improvement is performed separately for inlet/outlet edges and wall edges containing a new ridge point. For a ridge edge the optimal nodal index is 2 (2 wall edges, 2 outlet edges and 2 ridge edges, total 6) from the perspective of either wall or inlet/outlet surface. Thus, swapping must be performed such that at least one wall edge and one outlet edge are connected to a certain ridge point.

If an edge contains one ridge point and it is not a ridge edge, we call it a near-ridge edge (Figure 13). When contracting a near-ridge edge, the new point will always be placed at the position of the ridge endpoint. If both endpoints of an edge are ridge points and both triangles sharing this edge are wall triangles, the local mesh topology is not good, i.e. there is a ridge point which is not attached to a wall edge. In such situations, the local topology must be improved by edge swapping (see above) before edge contraction. The same holds for an inlet/outlet edge with two ridge endpoints. The edge swapping/splitting/contraction in the near-ridge part of the mesh needs special care in marking the resulting edges and nodes correctly.

3.4.4. Mesh refinement. If the element size required is smaller than the initial element size (determined by the voxel size), the edge splitting procedure could become computationally expensive. It is often more efficient to obtain a coarser mesh first (e.g. double the element size) using the splitting/contraction method, and then to refine the mesh by splitting every triangle into four. This can be performed by splitting every edge into two and inserting a new node at the mid-point of every edge. The method for placing the new points is described in Appendix B and it allows us

DOI: 10.1002/cnm

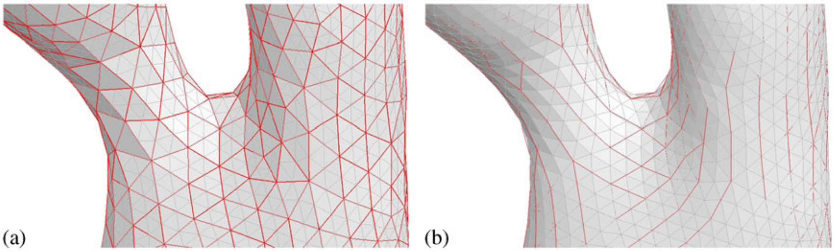

Figure 14. Refining the surface mesh: (a) standard method of point placement at edge midpoints and (b) new points are placed in accordance with the method described in Appendix B.

to create a much smoother mesh as shown in Figure 14. Figure 14(a) shows the standard point placement method and Figure 14(b) shows the mesh produced by the new point placement method proposed in Appendix B. As seen, the new method gives a much smoother surface mesh.

4. VOLUME MESHING

4.1. *Boundary layer meshing*

To resolve the viscous boundary layer in flow studies, a special structure of the near-wall volumetric mesh is required. The elements built in the direction of inward normal to the wall should be essentially smaller than the element size in the inner volume of an artery. The short edge of a near-boundary tetrahedra should be directed normal to the wall [40]. Such a quasi-structured, near-boundary mesh consists of N sub-layers with decreasing thickness towards the boundary. The ratio of the thickness between every neighbouring sub-layer towards the wall can be defined as $f<1$ (see Equation (17)). The mesh can be generated if we build triangular prisms at every face, divide every prism into N smaller prisms with height ratio f between two neighbouring prisms. If we need a tetrahedron mesh, then every triangular prism can be split into three tetrahedra (Figure 15). Following sections provide some details on the boundary layer meshing procedure.

4.1.1. Thickness of the boundary layer mesh and its sub-layers. The boundary layer mesh should gradually change the element size from the inner volumetric mesh element size (see below), which we denote as h, to a much smaller size at the boundary. The minimal sub-layer thickness $h_{\min}$ and total thickness of the boundary layer mesh h_{b} are defined as:

$$h_{\min}=hf^N, \quad h_{\mathrm{b}}=h(f+\cdots+f^N)=hf\frac{1-f^N}{1-f} \tag{16}$$

The number of sub-layers N and the thickness ratio f can be chosen experimentally depending on the problem studied. In [40] we show that $N=8$–10 seems sufficient to capture the shear stresses in a carotid artery. We use the following definition for f:

$$f=(1+N^{-\beta})^{-1} \tag{17}$$

where β is a tuning parameter. We take $\beta=0.452$, then $h_{\mathrm{b}}/h=2$ for $N=6$ [40].

4.1.2. Adjustment of the boundary layer near rapid variation in lumen area. The local boundary layer thickness must be decreased in narrow blood vessel branches or near a stenosis. It should also be decreased near high curvature parts in the surface mesh, especially near convex parts. Otherwise, self-interaction may occur in the boundary layer mesh. Therefore, three stages of the boundary-layer thickness correction are used here.

First, the distance from a node p to the opposite side of the surface is determined. This distance is determined as the distance between the point p and a point within a semi-infinite cylinder of

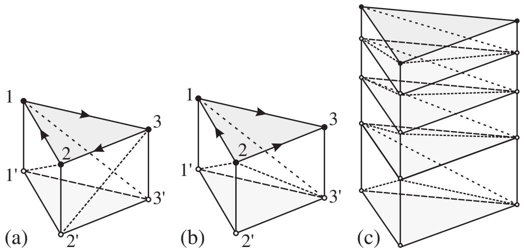

Figure 15. Splitting a prism into tetrahedra: (a) invalid case; (b) the prism is divided into three tetrahedra $\{1,2,3,3'\}$, $\{1,2,3',1'\}$, $\{1',2',3',2\}$; and (c) dividing a prism into smaller prisms and then into tetrahedra. The surface triangle is shaded darker, triangles of the sub-layers are shaded lighter. Wall surface points are indicated by closed circles, inner points—by open circles.

radius h with the axis identical to that of the inward normal, $\boldsymbol{n_p}$, built at the node p. The origin of the cylinder is the point p. To avoid identifying the neighbouring points of a cured wall, we only search for points beyond a minimum cylinder length of $2h$. If no points are found within the cylinder, the search is carried out within a cylinder of larger radius, and so on, until the point is found. If distance, l, to this point is less than $4h$ then the local boundary layer thickness is reduced to $h_\mathrm{b}=l/4$. Note that this option can be switched off if it is known *a priori* that there are no narrowing in the domain. Other alternatives such as medial axis or using a cone to identify the point on the opposite side is possible. However, all the methods have limitations.

Second, the mean dot-product of the unit normal vector $\hat{\boldsymbol{n}}_p$ with unit normal vectors of contiguous surface nodes is computed. The local boundary layer mesh thickness is decreased by this factor, i.e.

$$h'_b = h_\mathrm{b}\frac{1}{d_p}\sum_{i\in\mathscr{P}_p}(\hat{\boldsymbol{n}}_p\cdot\hat{\boldsymbol{n}}_i). \tag{18}$$

Third, to make the method robust, the distances between the ends of the normal vectors $\boldsymbol{p}+h'_b\hat{\boldsymbol{n}}_p$ and all $\boldsymbol{p}_i+h'_b\hat{\boldsymbol{n}}_i$, $i\in\mathscr{P}_p$ are calculated. If any of the distances is less than half the size of the corresponding edge on the surface mesh, $\|\boldsymbol{p}-\boldsymbol{p}_i\|$, the thickness is proportionally reduced to obtain the minimal distance $\|(\boldsymbol{p}_p+h'_b\hat{\boldsymbol{n}}_{\boldsymbol{p}})-(\boldsymbol{p}_i+h'_b\hat{\boldsymbol{n}}_i)\|$.

4.1.3. Arranging the edge directions. When splitting a triangle prism into three tetrahedra, every quadrilateral face is divided into two triangles by inserting diagonals as shown in Figure 15(a,b), e.g. in face $\{2,3,3',2'\}$, the dividing diagonal can be either $\{2',3\}$ or $\{2,3'\}$. If two prisms in the mesh share the same quadrilateral face then the dividing diagonal should be the same. Otherwise the 3D mesh will not be consistent.

In addition, the dividing diagonals should lead to three tetrahedrons and can therefore not be chosen arbitrarily and the procedure used should be well-defined. In the proposed procedure, a direction is assigned to every edge of the surface mesh, i.e. the endpoints of every edge are ordered (first and second). Every edge of the surface mesh corresponds to a quadrilateral face shared by two prisms. If the diagonal of a quadrilateral face starts at the first point of a given edge on the surface, it ends at the second point of the corresponding edge at the next layer, as shown in Figure 15(a),(b). Thus, as shown in Figure 15(a) choosing all three edges in the same clockwise (or counterclockwise) direction will result in difficulties. An algorithm to arrange the edge directions in a correct manner is described below.

Define a triangle as invalid if all its edges have the same directions and define it as valid otherwise. Call an edge shared by invalid and valid triangles as 'unflippable' if, after flipping its direction, the valid triangle becomes invalid. The aim of the algorithm is to remove all invalid triangles by flipping some edge directions. To do this, the following steps are used:

(i) Find and mark all invalid triangles in the mesh.
(ii) Take a marked triangle, find an edge shared with another invalid triangle (if exists) and flip its direction and unmark both triangles and go to (iv).

(iii) If no such edge exists, find an ‘unflippable’ edge and flip its direction. If more than one ‘unflippable’ edge is found, then flip the direction of a randomly chosen edge (its number should be selected by a random number generator). Unmark the triangle under consideration, and mark the adjacent one.

(iv) If any triangle is still marked, go to (ii) otherwise terminate.

Despite the fact that we cannot provide a rigorous proof that this algorithm works for an arbitrary mesh, it has not failed for the cases we studied so far.

4.1.4. Defining and combing the normals. The unit normal vector is computed for every surface point using Equation (B2). For the ridge, the normal is computed by summing the unit normal vectors of only the wall elements containing this node. The ridge normal vector is then projected onto the plane of the corresponding inlet/outlet surface and normalized after that.

If the surface mesh is not sufficiently smooth, the normals of the neighbouring points may not be very close to each other. The following ‘combing’ procedure smooths the variation in the normal directions. It is similar to the Laplace smoothing (Equation (11)) given by

$$\boldsymbol{n}^{\text{new}} = \hat{\boldsymbol{n}}(1-\lambda) + \frac{1}{d_p}\sum_{i\in\mathscr{P}_p} \hat{\boldsymbol{n}}_i\lambda, \quad \hat{\boldsymbol{n}}^{\text{new}} = \frac{\boldsymbol{n}^{\text{new}}}{\|\boldsymbol{n}^{\text{new}}\|}. \tag{19}$$

where λ is a weighting parameter. In this work, it was taken equal to unity, but in geometries with high curvature values, other values may be optimal. The iterations can be repeated several times, to obtain a smoothed surface normals.

4.1.5. Boundary-layer mesh generation. To generate the boundary layer part of the volumetric mesh, we take every node of the surface mesh and build a valid inward normal as explained previously. An inner triangular face is constructed corresponding to every surface triangle, at a distance of a given boundary layer thickness. Then, every segment connecting a surface point and a corresponding inner point is divided into a number of smaller segments using geometric progression according to $f^N : f^{N-1} : \cdots : f^2 : f$, to build N triangular prisms. Finally, every prism is divided into three tetrahedra. Figure 15(c) shows the procedure of splitting the prisms into tetrahedrons. The boundary layer meshing procedure may be summarized as:

1. Remove inlet/outlet parts of the surface mesh.
2. Decide the number of sub-layers, N, in the boundary-layer mesh and the thickness ratio f. The latter factor can be computed via Equation (17).
3. Compute the total thickness via Equation (16).
4. Compute inward normals to every point in accordance with (B2).
5. ‘Comb’ the normals (see Section 4.1.4).
6. Correct local boundary-layer thickness (shrink or expand) according to the domain cross-sectional variations (see Section 4.1.2).
7. Find positions of the inner surface points (for both wall and ridge points). Smooth the inner surface mesh by Taubing smoothing algorithm (Equations (12))–(13).
8. To close the inlets/outlets, the stitching method [94] is used.
9. Generate the quasi-structured, 3D tetrahedral mesh by dividing the prisms into smaller prisms first and then by dividing a smaller prism into three tetrahedra.

If the mesh without boundary layer has to be generated, the algorithm presented in the following section for inner volume mesh generation can be applied directly to the surface mesh.

4.2. Inner volume mesh

Once the boundary layer mesh generation is completed, a semi-structured, tetrahedral mesh occupies a small region along the vessel walls. Thus, in addition to exterior surfaces, an interior surface at the edge of the boundary layer exists. This interior surface mesh is open at the inlets/outlets. To close the inlets/outlets, the stitching method [94] is used. After improving the quality of the

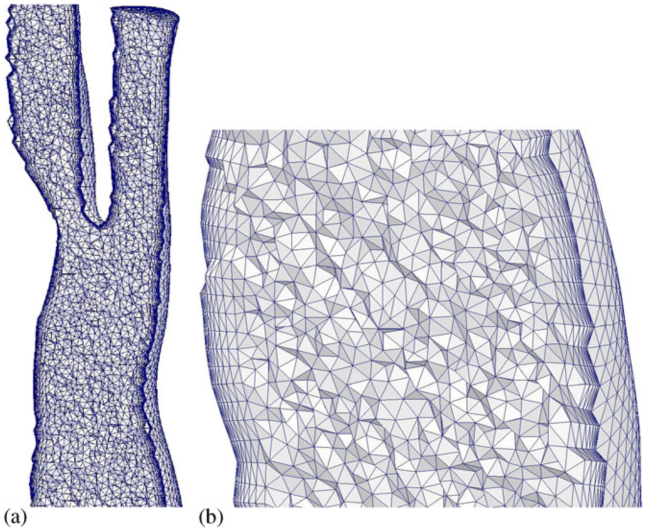

Figure 16. Volume mesh generated for a carotid artery.

new surface mesh at the inlets/outlets using Taubing smoothing, we use an in-house developed code [64] based on a Delaunay triangulation to insert the points to generate a 3D volume mesh of tetrahedrons. Subsequently, 3D mesh cosmetics based on 3D swapping, and smoothing is applied. As mentioned, the quality of the 3D mesh depends on the quality of the surface mesh. In all the carotid arteries that we have reconstructed so far, the smallest tetrahedral aspect ratio for a specific geometry lies between 0.01 and 0.1 (with $\sqrt{2/3}$ being the optimal aspect ratio). This result excludes the boundary layer mesh for obvious reasons. A portion of the volume mesh cut by a plane passing approximately along the centroidal line is shown in Figure 16. The minimum aspect ratio of unstructured tetrahedra mesh observed here is 0.08.

5. NUMERICAL MODELLING

In this section, the flow boundary conditions at the inlet and exits of an artery are derived. In addition to analysing the Womersley and Poiseuille flow profiles, transient numerical calculation of flow profiles is also explained in detail. In addition to the velocity profiles, flow rate division, pressure gradient variation and flow cosmetics are also explained in this section. Finally, a brief summary of the flow solver is also provided.

5.1. Equations for velocity profile

Consider a time varying flow in a pipe of arbitrary cross-section with a rigid wall. Let z be the coordinate along the pipe axis and x and y the coordinates of the cross-section. The flow velocity $\boldsymbol{u}$ and pressure p obey the Navier–Stokes and continuity equations:

$$\frac{\partial}{\partial t}\boldsymbol{u}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}+\frac{1}{\rho}\nabla p=v\nabla^2\boldsymbol{u} \tag{20}$$

$$\nabla\cdot\boldsymbol{u}=0 \tag{21}$$

where ρ is the density, v is the kinematic viscosity. The velocity vanishes at the pipe wall.

If the pipe is straight and its cross-section is constant then it can be shown that the velocity and the pressure gradient ∇p have only axial components: $\boldsymbol{u}=[0,0,u(x,y,t)]^{\mathrm{T}}$, $\nabla p=[0,0,p_z(t)]^{\mathrm{T}}$. Moreover, the pressure gradient depends on time t only. Thus, Equation (21) is satisfied automatically, and from Equation (20) we obtain a single linear parabolic inhomogeneous PDE as a function of z velocity component $u(x, y, t)$ as:

$$-\frac{\partial}{\partial t}u + v\nabla_{\perp}^2 u = \frac{p_z(t)}{\rho}, \quad \nabla_{\perp} = [\partial_x, \partial_y]^{\mathrm{T}}. \tag{22}$$

Let the flow be periodic with a period T: $u(t+T) = u(t)$, $p(t+T) = p(t)$. By applying the Fourier transform, we represent the flow as a sum of harmonic components as

$$u(x, y, t) = \sum_{n=-\infty}^{\infty} \tilde{u}_n(x, y)\mathrm{e}^{\mathrm{i}\omega_n t}, \quad p_z(t) = \sum_{n=-\infty}^{\infty} (\tilde{p}_z)_n \,\mathrm{e}^{\mathrm{i}\omega_n t} \tag{23}$$

where $\omega_n = 2\pi n/T$, $\mathrm{i} = \sqrt{-1}$. Substituting (23) into (22) gives a set of uncoupled PDEs for every harmonic component as:

$$-\mathrm{i}\omega_n \tilde{u}_n + v\nabla_{\perp}^2 \tilde{u}_n = \frac{(\tilde{p}_z)_n}{\rho}, \quad n = \ldots, -2, -1, 0, 1, 2, \ldots.$$

Thus, for every harmonic component, we have a 2D boundary value problem: the inhomogeneous Helmholtz equation (the Poisson equation if $n = 0$) in the domain Ω with the Dirichlet boundary conditions, i.e.

$$\nabla_{\perp}^2 \tilde{u}_n + k_n^2 \tilde{u}_n = f_n, \quad \{x, y\} \in \Omega \tag{24}$$

$$\tilde{u}_n = 0, \quad \{x, y\} \in \partial\Omega \tag{25}$$

Here $k_n = \sqrt{-\mathrm{i}\omega_n/v} = \sqrt{\omega_n/2v}(1-i)$ is Stokes's viscous wave number, $f_n = (\tilde{p}_z)_n/(v\rho)$ is the r.h.s.

In the case of a circular pipe of radius a: $\Omega = \{x, y : x^2 + y^2 < a^2\}$, the boundary value problem (24)–(25) admits an analytical solution

$$\tilde{u}_n = \tilde{U}_n \frac{1 - J_0(k_n r)/J_0(k_n a)}{1 - 1/J_0(k_n a)}, \quad \tilde{U}_n = \frac{(\tilde{p}_z)_n}{i\omega_n \rho}\left[1 - \frac{1}{J_0(k_n a)}\right], \quad n \neq 0 \tag{26}$$

$$\tilde{u}_0 = \tilde{U}_0 \left(1 - \frac{r^2}{a^2}\right), \quad \tilde{U}_0 = \frac{(\tilde{p}_z)_0}{v\rho}, \quad n = 0 \tag{27}$$

which becomes a Womersley profile [96] for $n \neq 0$ and Poiseuille's profile for $n = 0$. Here $J_0(.)$ is the Bessel function of order zero, $\tilde{U}_n$ is the velocity of the nth harmonic component at the centre of the circular domain and $|k_n a|$ is the Womersley number.

If the inlet/outlet is close to a circular shape or if a cylindrical extension is added to the inlet/outlet then Womersleys solution can be applied directly. In [3] a method is proposed to map the Womersley profile from a circular to a non-circular cross-section. This is acceptable as long as the cross-section is close to a circular shape. For an arbitrary inlet/outlet with a non-circular cross-section, which is often the case (Figure 11), the boundary value problem can be numerically solved. For example, using a standard FE method and the surface triangulation of the inlet/outlet. The presence of the boundary layer mesh increases the accuracy of the numerical solution. Comparison between the analytical and numerical solution in a cylindrical pipe of radius $a = 1$ is presented in Figure 17 for Womersley numbers of 0 (a,b) and 11.5 (c,d). Here $|k_n a| = 11.5$ corresponds to the 15th harmonic component in a carotid artery with an inlet of radius $a = 0.23\,\mathrm{cm}$ (then πa^2 equal to the inlet area), $v = 0.04$, $T = 0.92\,\mathrm{s}$. As seen, the difference between the analytical and numerical solutions is negligible.

5.2. *Obtaining a time varying inlet velocity profile*

At present, it is not easy to measure the pressure gradient *in vivo*. Ultrasonic measurements allow obtaining the velocity waveform $U(t)$ at the axis of a blood vessel where the velocity is maximum [97]. Assuming that the waveform is a periodic function with period T, i.e.

$$U(t) = U(t+T)$$

DOI: 10.1002/cnm

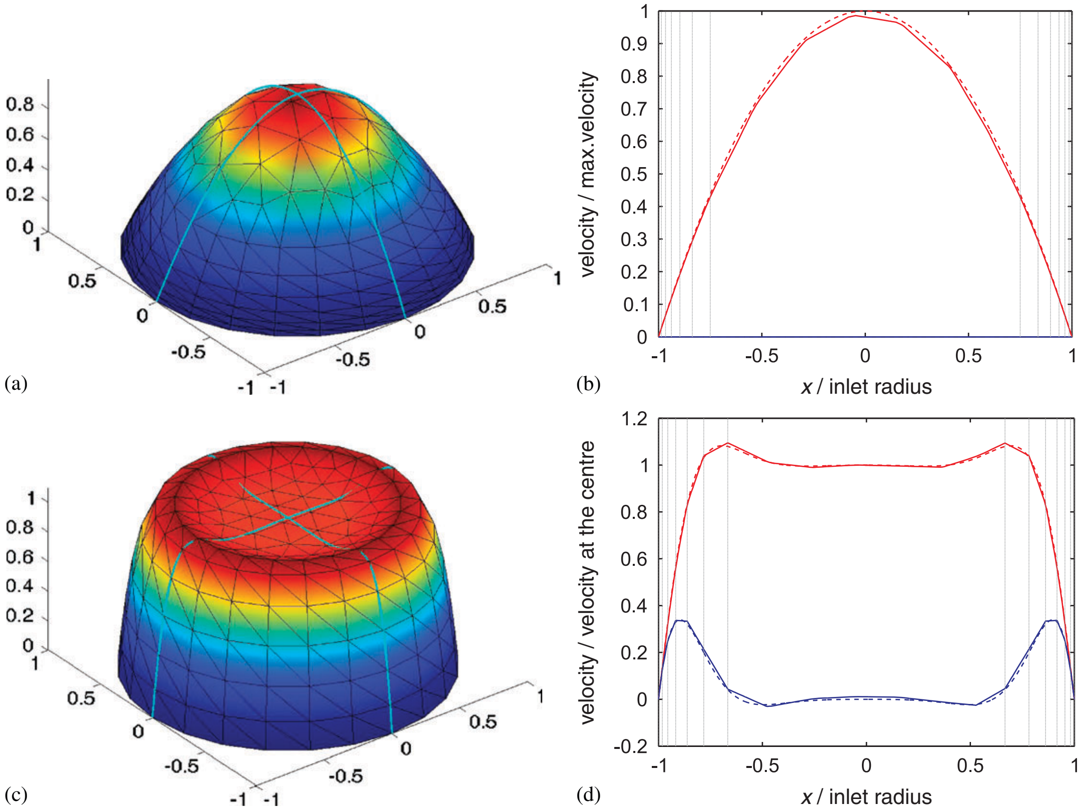

Figure 17. Velocity profile $\tilde{u}_n/\tilde{U}_n$ computed for $k_n a = 0$ (a,b) and $|k_n a| = 11.5$ (c,d) for a circular pipe of radius unity. (a,b)—only the real part of the velocity is shown, (c,d)—real part is indicated by red, the imaginary part—by blue. Womersley's profile is shown by cyan lines (a,c) (real part only) and by dashed lines in (b,d). Sub-layers of the boundary layer mesh are shown by black dotted lines in (b,d). View this figure color online at [10.1002/cnm.1446].

we can expand it into the Fourier series as

$$U(t) = \sum_{n=-\infty}^{\infty} \tilde{U}_n \, \mathrm{e}^{\mathrm{i}\omega_n t} \tag{28}$$

If the inlet is circular, we can substitute complex amplitudes $\tilde{U}_n$ directly into the left half of relations (26)–(27) without computing $(\tilde{p})_n$. Here, we take the waveform from [97] and the details are described in Section 5.4. In the case of a non-circular inlet, the procedure of obtaining inlet velocity profile $u(x, y, t)$ from the inlet velocity waveform $U(t) = u(x_c, y_c, t)$ (with (x_c, y_c) the point where the velocity profile reaches its maximal value) is given below.

1. Compute complex amplitudes by the FFT method, i.e.

$$\tilde{U}_n = \int_0^{\mathrm{T}} U(t)\mathrm{e}^{-\mathrm{i}\omega_n t}, \quad \omega_n = \frac{2\pi}{T}, \quad n = 0, \ldots, N \tag{29}$$

   A sufficient number of harmonic components, $N+1$, should be used to resolve the waveform accurately. For the velocity profile used in this work, we take $N = 24$. The corresponding data are listed in Appendix C.

2. Solve $N+1$ boundary value problems numerically

$$\nabla_\perp^2 \tilde{u}_n + k_n^2 \tilde{u}_n = -1, \quad \{x, y\} \in \Omega \tag{30}$$

$$\tilde{u}_n = 0, \quad \{x, y\} \in \partial\Omega \tag{31}$$

To do this, consider a surface mesh of an inlet/outlet and transform the mesh to the horizontal plane. This is done by rotating the plane with respect to the barycentre of the inlet/exit. After rotation, the z co-ordinates of all inlet/outlet nodes will be the same. Now, along with the x, y co-ordinates of the nodes and the mesh connectivities, we obtain a plane 2D mesh that is used in the numerical solution. Note that there is no necessity to compute $\tilde{u}_n$ for negative n, i.e. $\tilde{u}_{-n} = \tilde{u}_n^*$ as $k_{-n}^2 = (k_n^2)^*$.

The value on the r.h.s. of Equation (30) is not important since the solution to the problem will be normalized. The only condition is that the r.h.s must be a non-zero constant. We use a negative value for the r.h.s. to get a positive solution for the zeroth harmonics ($n = 0$). For other harmonics, the solution is complex due to the fact that parameter k_n^2 is complex.

3. Find a node p_c where $\tilde{u}_0$ is maximum: find x_c, y_c: $\tilde{u}_0(x_c, y_c) \to \max$. This node's position is the point of interaction of the blood vessel axis with the inlet. We refer to this node as an axial node.
4. Normalize the solutions using the value at the axial node

$$\tilde{v}_n(x,y) = \frac{\tilde{u}_n(x,y)}{\tilde{u}_n(x_c,y_c)}, \quad n = 0, \ldots, N$$

5. Compute the inlet velocity profile at every instant using

$$u(x,y,t) = \sum_{n=-N}^{N} \tilde{U}_n \tilde{v}_n(x,y) \mathrm{e}^{\mathrm{i}\omega_n t} = U_0 \tilde{v}_0(x,y) + 2\sum_{n=1}^{N} \mathrm{Re}[\tilde{U}_n \tilde{v}_n(x,y) \mathrm{e}^{\mathrm{i}\omega_n t}]$$

The computed profiles at different instants are shown in Figure 18(a) and the exit velocity profiles are shown in 18(b). Figure 19 shows the pulse velocity wave form.

Note that the gradient pressure waveform can also be evaluated. Let $\tilde{u}_n$ be the solutions to the boundary value problem (30)–(31). Then, $\tilde{u}_n(-\tilde{p}_z)_n/(\nu\rho)$ is the solution to the boundary value problem (24)–(25). Thus, we can write

$$U(t) = \sum_{n=-\infty}^{+\infty} \tilde{U}_n \mathrm{e}^{\mathrm{i}\omega_n t} = \sum_{n=-\infty}^{+\infty} \frac{-(\tilde{p}_z)_n}{\rho\nu} \tilde{u}_n \mathrm{e}^{\mathrm{i}\omega_n t}$$

From this we derive

$$(\tilde{p}_z)_n = -\rho\nu \frac{\tilde{U}_n}{\tilde{u}_n} \quad \text{and} \quad p_z(t) = -\rho\nu \sum_{n=-\infty}^{+\infty} \frac{\tilde{U}_n}{\tilde{u}_n} \mathrm{e}^{\mathrm{i}\omega_n t}.$$

The pressure gradient waveform at the inlet is shown in Figure 20.

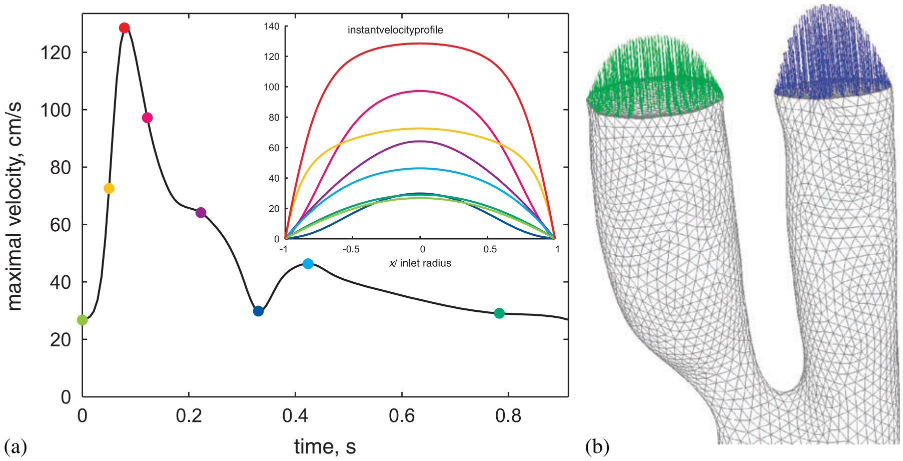

Figure 18. (a) Velocity pulse wave form and velocity profile at the inlet (colour of the line corresponds to the instant indicated by the circle of the same colour) and (b) Outlet velocity vectors. View this figure color online at [10.1002/cnm.1446].

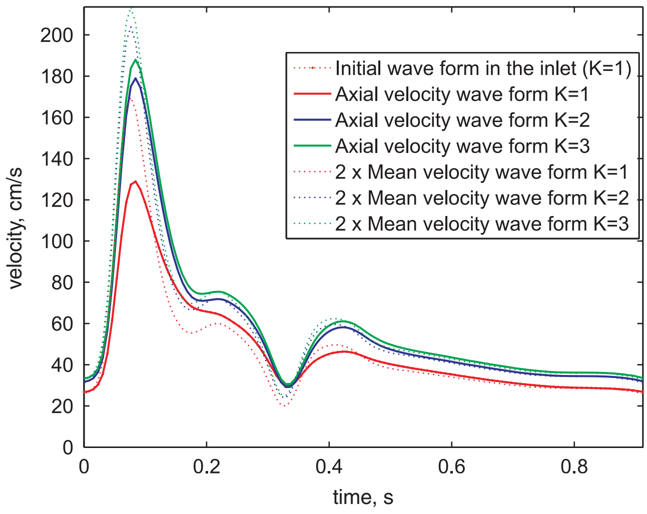

Figure 19. Velocity waveforms. Initial pulse at the inlet ($K=1$ dotted line), pulse formed by 24 harmonics (solid red ($K=1$): almost covers the initial pulse), pulses at outlets ($K=2$ blue, $K=3$ green). The lines with twice the mean velocity (maximal velocity in Poiseuille's flow for an identical flow rate) are also shown. View this figure color online at [10.1002/cnm.1446].

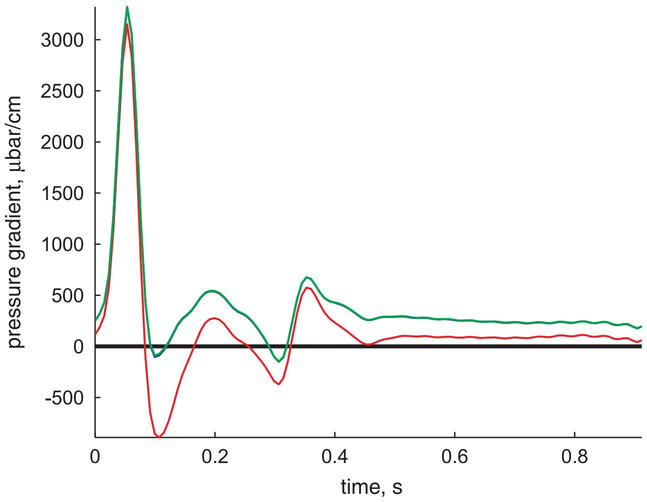

Figure 20. Pressure gradient waveform: red (inlet), blue, green (outlets). View this figure color online at [10.1002/cnm.1446].

5.3. *The flow rate and outlet velocity wave forms*

To compute the flow rate at an inlet or outlet, we can use the following equations:

$$W(t)=\iint_{\Omega} u(x,y,t)\,\mathrm{d}x\,\mathrm{d}y=\iint_{\Omega}\sum_{n=-\infty}^{+\infty}\tilde{U}_n\tilde{v}_n\mathrm{e}^{\mathrm{i}\omega_n t}=\sum_{n=-\infty}^{+\infty}\tilde{U}_n\tilde{w}_n\mathrm{e}^{\mathrm{i}\omega_n t} \tag{32}$$

$$\tilde{w}_n=\iint_{\Omega}\tilde{v}_n(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{33}$$

Alternatively, it is also possible to expand flow rate $W(t)$ using a Fourier series as:

$$W(t)=\sum_{n=-\infty}^{+\infty}\tilde{W}_n\mathrm{e}^{\mathrm{i}\omega_n t},\quad \tilde{W}_n=\int_0^T W(t)\mathrm{e}^{-\mathrm{i}\omega_n t},\quad \omega_n=\frac{2\pi}{T} \tag{34}$$

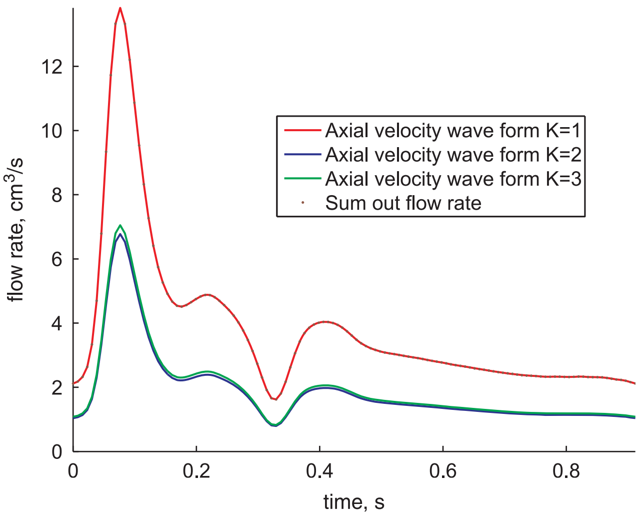

Figure 21. Flow rate wave forms.

Comparing (32) and (34), we obtain the relationship between the axial velocity and the flow rate harmonic complex amplitudes, i.e.

$$\tilde{W}_n = \tilde{U}_n \ \tilde{w}_n \tag{35}$$

This flow rate variation in time is shown in Figure 21.

Now, we consider the following problem. An axial waveform $U^{\text{inlet}}(t)$ is given and the flow rate is divided into specified proportions beyond the bifurcation. If the flow rate at an outlet is $\mu^{\text{outlet}}(<1)$ times the flow rate at the inlet, at every instant, find the axial velocity waveform at the outlet $U^{\text{outlet}}(t)$. The following steps may be used to compute the outlet velocity:

(1) Compute the inlet axial velocity harmonics (=complex amplitudes)

$$\tilde{U}_n^{\text{inlet}} = \int_0^T U^{\text{inlet}}(t)\text{e}^{-\text{i}\omega_n t}$$

(2) Compute the inlet flow rate harmonics

$$\tilde{W}_n^{\text{inlet}} = \tilde{w}_n^{\text{inlet}} \tilde{U}_n^{\text{inlet}} \quad \text{where } \tilde{w}_n^{\text{inlet}} = \int_\Omega \tilde{v}_n^{\text{inlet}}(x,y)\,\text{d}x\,\text{d}y, \ \ \tilde{v}_n^{\text{inlet}}(x,y) = \frac{\tilde{u}^{\text{inlet}}(x,y)}{\tilde{u}^{\text{inlet}}(x_c^{\text{inlet}}, y_c^{\text{inlet}})}$$

where $\tilde{u}^{\text{inlet}}(x,y)$ is the solution to the boundary value problem (Equations (30)–(31)) at the inlet.

(3) Compute the outlet flow rate harmonics

$$\tilde{W}_n^{\text{outlet}} = \mu^{\text{outlet}} \tilde{W}_n^{\text{inlet}}$$

(4) Compute the outlet axial velocity harmonics

$$\tilde{U}_n^{\text{outlet}} = \frac{\tilde{W}_n^{\text{outlet}}}{\tilde{w}_n^{\text{outlet}}} = \mu^{\text{outlet}} \frac{\tilde{w}_n^{\text{inlet}} \ \tilde{U}_n^{\text{inlet}}}{\tilde{w}_n^{\text{outlet}}}$$

where

$$\tilde{w}_n^{\text{outlet}} = \int_\Omega \tilde{v}_n^{\text{outlet}}(x,y)\,\text{d}x\,\text{d}y, \quad \tilde{v}_n^{\text{outlet}}(x,y) = \frac{\tilde{u}^{\text{outlet}}(x,y)}{\tilde{u}^{\text{outlet}}(x_c^{\text{outlet}}, y_c^{\text{outlet}})}$$

Note that $\tilde{u}^{\text{outlet}}(x,y)$ is the solution to the boundary value problem (Equations (30)–(31)) at the outlet. Thus we need $\tilde{v}_n^{\text{inlet}}, \tilde{v}_n^{\text{outlet}}$ to compute $\tilde{w}_n^{\text{inlet}}, \tilde{w}_n^{\text{outlet}}$.

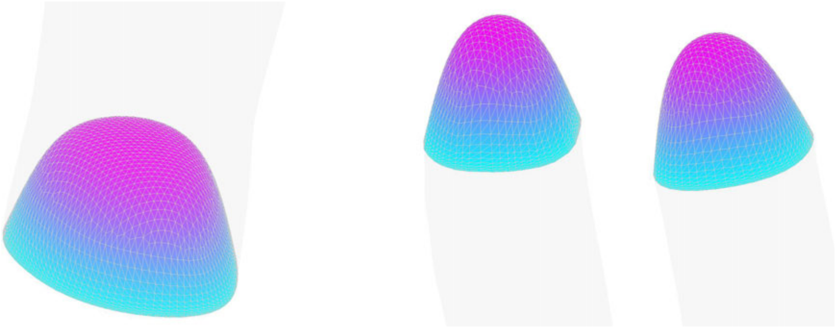

Figure 22. Velocity profile at maximum flow rate, left: inlet, right: outlets.

The outlet velocity waveform at its axis is

$$U^{\text{outlet}}(t)=\sum_{n=-\infty}^{+\infty}\tilde{U}_n^{\text{outlet}}\mathrm{e}^{\mathrm{i}\omega_n t}=\mu^{\text{outlet}}\sum_{n=-\infty}^{+\infty}\tilde{U}_n^{\text{inlet}}\frac{\tilde{w}_n^{\text{inlet}}}{\tilde{w}_n^{\text{outlet}}}\mathrm{e}^{\mathrm{i}\omega_n t} \tag{36}$$

Once the velocity wave form is computed, the corresponding outlet velocity profile can be calculated as:

$$u^{\text{outlet}}(x,y,t)=\sum_{n=-\infty}^{+\infty}\tilde{U}_n^{\text{outlet}}\tilde{v}_n^{\text{outlet}}(x,y)\mathrm{e}^{\mathrm{i}\omega_n t}=\mu^{\text{outlet}}\sum_{n=-\infty}^{+\infty}\frac{\tilde{w}_n^{\text{inlet}}\,\tilde{U}_n^{\text{inlet}}}{\tilde{w}_n^{\text{outlet}}}\tilde{v}_n^{\text{outlet}}(x,y)\mathrm{e}^{\mathrm{i}\omega_n t} \tag{37}$$

The calculated velocity profiles at the inlet and outlets are shown in Figure 22.

5.4. *The wave form*

In [97], more than 3000 recorded waveforms in carotid arteries have been analyzed. Finally, two wave forms were presented: one as an average of all waveforms and another as a synthetic waveform obtained by fitting a cubic spline from some selected critical points of the pulse. We choose not to use them as they are (i) not periodic and (ii) after averaging, the pulse becomes smoother. Therefore, we take the synthetic waveform and adjust its parameters to the pulse of 5 beats depicted in Figure 3(a) of [97]. As a result, we obtain a realistic maximal acceleration (about $3000\,\mathrm{cm\,s^{-2}}$) as shown in Figure 3(b) of [97].

5.5. *Computing velocity directions at inlet and outlet*

If the blood vessel cross-section varies significantly near an inlet or outlet, the wall element faces may not be orthogonal to the inlet/outlet faces. Thus, if the velocity vectors on the inlet/outlet surfaces are placed normal to the plane, some parasitic flow separation may occur near the inlet/outlet. To avoid this, it may be beneficial to direct the near-wall vectors parallel to the walls and then progressively make the velocity vectors perpendicular to the inlet/exit plane towards the barycentre. To do this, we first compute the tangent vectors to the wall in all ridge nodes. To compute tangents, we need to compute the normal vectors of the wall ridge nodes first (Equation (B2)). The ridge node normals may be calculated as:

$$\boldsymbol{n}_p^{\mathrm{w}}=\sum_{f\in\mathscr{F}_p^{\mathrm{w}}}\hat{\boldsymbol{n}}_f,\quad \hat{\boldsymbol{n}}_p^{\mathrm{w}}=\frac{\boldsymbol{n}_p^{\mathrm{w}}}{\|\boldsymbol{n}_p^{\mathrm{w}}\|} \tag{38}$$

where $\mathscr{F}_p^{\mathrm{w}}$ is a set of wall faces containing ridge point p. Let $\hat{\boldsymbol{N}}_k$ be a normal vector to the kth outlet (which is under consideration). Then, we draw a vector $\boldsymbol{t}_p^{\mathrm{w}}$ orthogonal to $\hat{\boldsymbol{n}}_p^{\mathrm{w}}$ laying in the

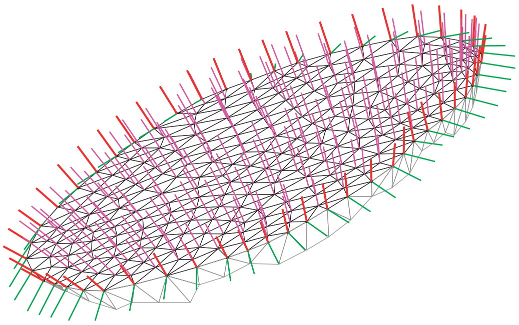

Figure 23. Velocity directions at an outlet: green—normal unit vectors to the wall, red—tangent unit vectors along the wall, magenta—'combed' outlet velocity directions.

plane formed by span of vectors $\hat{\boldsymbol{N}}_k$ and $\hat{\boldsymbol{n}}_p^{\mathrm{w}}$, i.e.

$$\boldsymbol{t}_p^{\mathrm{w}} = \hat{\boldsymbol{N}}_k - \hat{\boldsymbol{n}}_p^{\mathrm{w}}(\hat{\boldsymbol{N}}_k \cdot \hat{\boldsymbol{n}}_p^{\mathrm{w}}), \quad \hat{\boldsymbol{t}}_p^{\mathrm{w}} = \frac{\boldsymbol{t}_p^{\mathrm{w}}}{\|\boldsymbol{t}_p^{\mathrm{w}}\|} \tag{39}$$

On all non-ridge outlet points, we set $\hat{\boldsymbol{t}}_p = \hat{\boldsymbol{N}}_k$ initially. We then perform iterations analogous to Laplace smoothing and identical to the combing of the normal vectors described previously. The result of combing is presented in Figure 23. To prepare this picture, we made a special cut of the mesh immediately after the stenosis (in the ICA), where the cross-section rapidly varies.

5.6. *Flow solution*

The incompressible Navier–Stokes equations governing flow in large arteries are given by Equations (20) and (21). With appropriate boundary and initial conditions, the system of equations is closed. These equations can be solved in a variety of fashion. In the present work, we adopted the explicit characteristic-based split (CBS) method [3, 55, 68–70, 72, 98–100]. The linear FEs are used in the spatial discretization. In the explicit CBS scheme, the governing equations are solved in three steps. They are:

Step 1: Intermediate momentum

$$\left(\frac{\Delta\tilde{\boldsymbol{u}}}{\Delta t} + \boldsymbol{u}^n.\nabla\boldsymbol{u}^n\right) = v\nabla^2\boldsymbol{u}^{n+\theta_1} + \frac{\Delta t}{2}\boldsymbol{u}\cdot\boldsymbol{\Delta R}^n \tag{40}$$

where $\Delta\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{u}} - \boldsymbol{u}^n$, $\tilde{\boldsymbol{u}}$ is an intermediate velocity field and $\boldsymbol{R}^n = \boldsymbol{u}^n.\nabla\boldsymbol{u}^n$ is the stabilizing characteristic Galerkin term [101].

Step 2: Pressure calculation

$$(1-\theta_2)\left(\frac{1}{\beta^2}\right)^n \frac{\Delta p}{\Delta t} - \Delta t\nabla^2 p^{n+\theta_2} = -\nabla\cdot\tilde{\boldsymbol{u}} \tag{41}$$

Step 3: Momentum correction

$$\frac{\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}}{\Delta t} = -\Delta p^{n+\theta_2} \tag{42}$$

In the above equations θ_1 and θ_2 vary between 0 and 1 depending on whether equations are treated explicitly, implicitly or semi-implicitly. All the results given in this work are generated by

DOI: 10.1002/cnm

assuming $\theta = \theta_1 = \theta_2 = 0$. The semi-discrete form of flow field equations (Equations (40)–(42)) is integrated in space using the standard Galerkin FE method. For further details on the discretization and solution methodology, refer to [68, 99].

6. FLOW RESULTS

In this section, analysis of flow in a severely stenosed carotid artery is presented. The geometry used here is generated using the image segmentation method presented in Section 2, the surface mesh used is obtained by employing all the techniques discussed in Section 3, the boundary layer and the inner, unstructured volume meshes are generated using the approaches discussed in Section 4. The transient flow boundary conditions are generated using the methods outlined in Section 5. Finally, the fluid flow calculations are carried out using the algorithm presented in Section 5.6. The accuracy of the transient fluid dynamics code employed in this work has been previously verified on a number of problems, including biomedical cases [3, 40, 41, 70, 72, 100]. Therefore, demonstration of the accuracy of the scheme is not repeated here.

The carotid artery geometry used here was obtained from the Heart and Lung Centre, at Wolverhampton Hospital. The carotid artery contains a severe stenosis within the ICA immediately downstream of the bifurcation. For the numerical simulation, the mesh is clipped at the mid-height of the common carotid artery (CCA) and maximal lengths of both internal and external carotid arteries (ECAs) are retained (Figure 8(a)). The final mesh is depicted in Figure 24. This mesh consists of 4 126 777 linear tetrahedral elements and 708 191 nodes with 10 structured boundary layers. The stenosis occludes approximately 65% of the ICA. The CCA and the ECA, however, are typical healthy arteries. A flow division of 50–50% between ECA and ICA is assumed. In normal circumstances, a flow division of 40–60% between ECA and ICA may be used. Since the ICA is partially occluded in the geometry used here, we assumed a 50–50% flow division.

To examine the complex flow patterns within the stenosed artery and bifurcation in greater detail, the flow is visualized through a number of slices in Figure 25. Here, the velocity vectors for the representative phases of the cardiac cycle (mid acceleration, peak flow, mid deceleration, dicrotic notch) are shown. From the figure, it is clear that the stenosis has a strong influence on the downstream flow. This is shown by the flow separation and recirculation between the bifurcation and at least up to slice J. As a result, low WSS would be expected downstream of the stenosis, marking a potential region of further plaque formation. This is in good agreement with Smedby [102], who found that plaque was more likely to grow downstream of a stenosis. Within the CCA and ICA, the flow at mid acceleration and peak flow is typically well distributed within the slice,

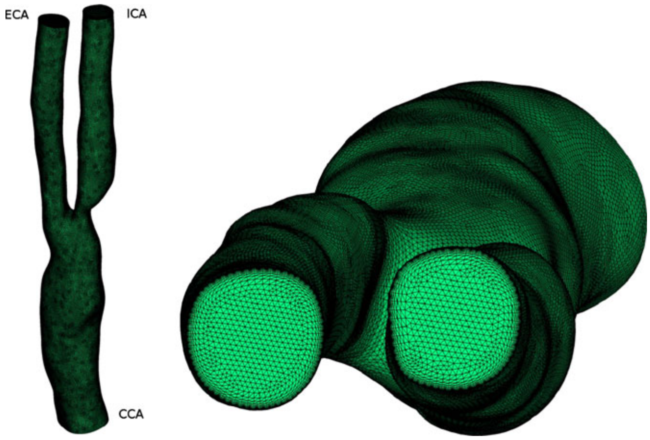

Figure 24. Mesh used for the computation.

DOI: 10.1002/cnm

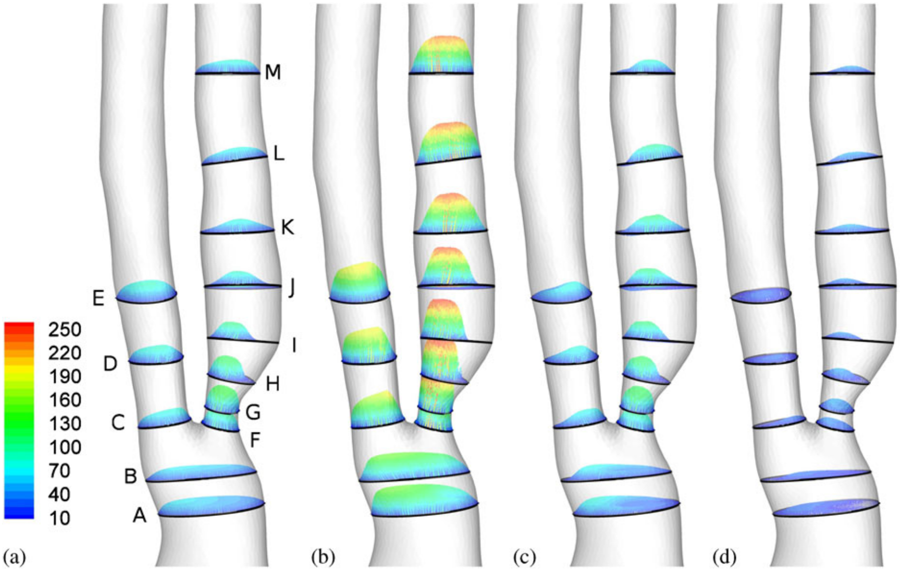

Figure 25. 3d velocity distribution within 13 slices (cm/s): (a) mid accl; (b) peak; (c) mid decel; and (d) dicrotic notch.

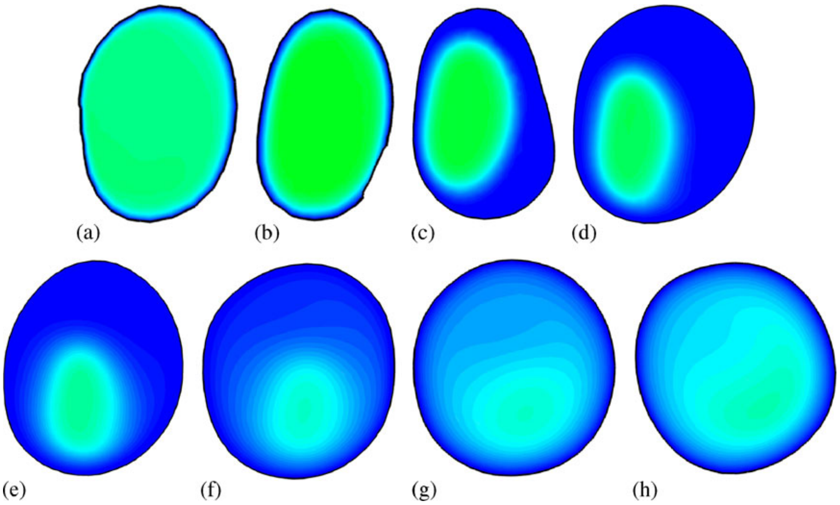

Figure 26. Streamline velocity magnitude at eight cross sections within the ICA at mid acceleration (cm/s): (a) slice F; (b) slice G; (c) slice H; (d) slice I; (e) slice J; (f) slice K; (g) slice L; and (h) slice M.

although in the ECA the flow is skewed towards the inner wall. It is during deceleration that the skewed nature becomes more obvious, with larger velocity values found closer to the inner wall in the ECA. To further establish the flow conditions within the ICA, the slices are now presented for three time instances. The mid acceleration time instance in Figure 26, peak flow in Figure 27, and during deceleration in Figure 28. From the three figures, it is possible to see the evolution of the flow during systole. During the mid-acceleration stage, flow maintains a strong jet from slice F to slice J, before it begins to diffuse and by slice M reattachment has occurred. In contrast, during peak flow the length of the jet has extended such that it impacts the posterior wall and flow is skewed towards this wall even in slice M. During peak flow, the predicted overall peak velocity is 266 cm/s within slice G and even in slice M, the velocity readily exceeds 200 cm/s. Within

DOI: 10.1002/cnm

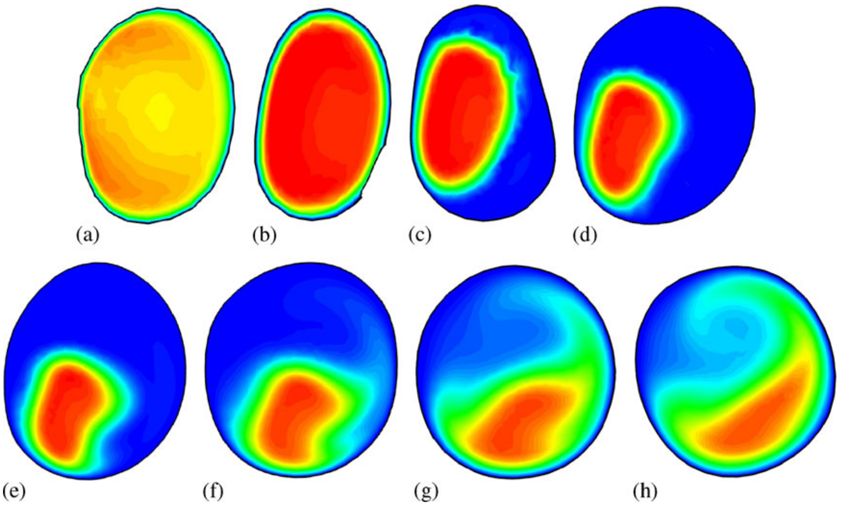

Figure 27. Streamline velocity magnitude at eight cross sections within the ICA at peak flow (cm/s): (a) slice F; (b) slice G; (c) slice H; (d) slice I; (e) slice J; (f) slice K; (g) slice L; and (h) slice M.

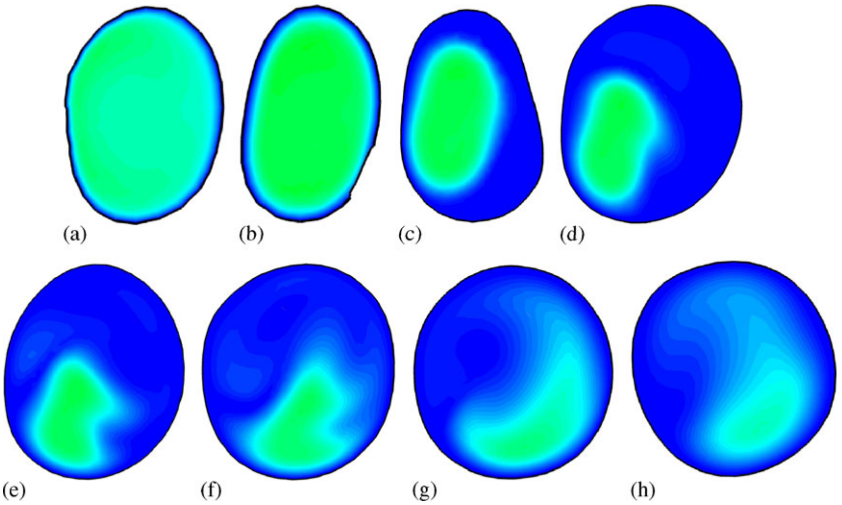

Figure 28. Streamline velocity magnitude at eight cross sections within the ICA at mid deceleration (cm/s). (a) slice F; (b) slice G; (c) slice H; (d) slice I; (e) slice J; (f) slice K; (g) slice L; and (h) slice M.

the deceleration phase, the influence of the jet is still strong, although secondary flow is observed downstream of the stenosis.

Low WSS is one of the indicators of the location of atherogenesis [56, 60, 103–108]. The work [109] defines the low shear stress as any value less than $5\,\mathrm{dyne/cm^2}$ and this work also defines an athero-protective region as the region where the WSS exceeds $25\,\mathrm{dyne/cm^2}$. While instantaneous WSS values can provide individual snapshots of potential atherogenesis and athero-protective regions, the time-averaged WSS is employed to provide information over the full cardiac cycle. The time-averaged WSS can be determined using

$$\tau_{\mathrm{abs}} = \frac{1}{T}\int_0^T \|\mathbf{t_s}\|\,\mathrm{d}t. \tag{43}$$

DOI: 10.1002/cnm

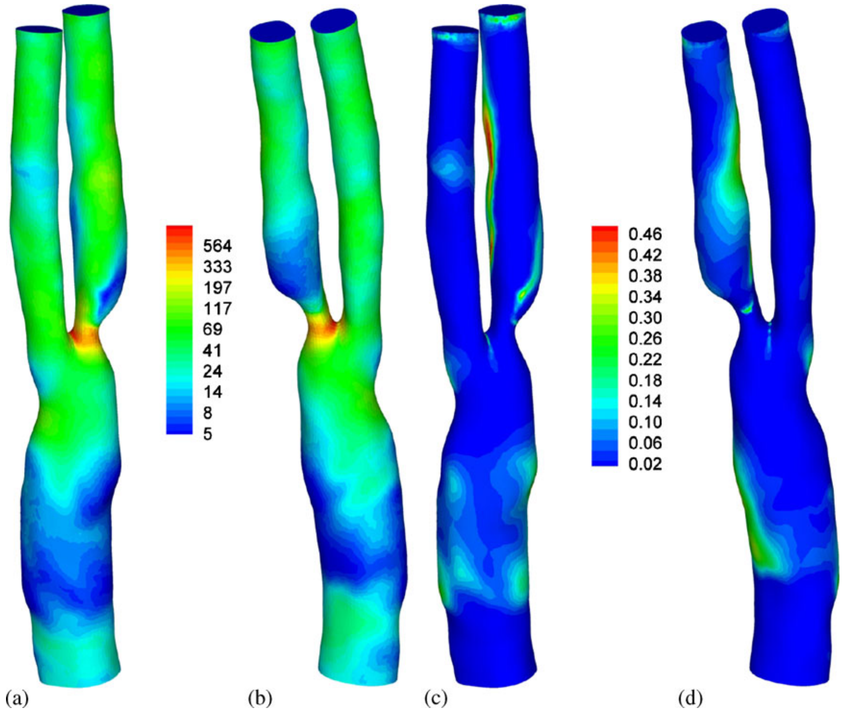

Figure 29. Haemodynamic wall parameter distributions. (a) and (b) Time-averaged WSS (dyne/cm^2); (c) and (d) OSI. (a and c) Anterior and (b and d) posterior.

where $\mathbf{t}_s = \mathbf{t} - (\mathbf{t}\cdot\mathbf{n})\mathbf{n}$ is the surface traction vector and the traction vector $\mathbf{t}$ is calculated from the Cauchy stress tensor $\boldsymbol{\sigma}$ and the surface normal vector $\mathbf{n}$ using $\mathbf{t} = \boldsymbol{\sigma}\cdot\mathbf{n}$. The OSI, originally introduced in [56] and adapted for general three dimensional use in [110], is also used in this work to quantify the transient shear stress dynamics experienced by the endothelial cells. The OSI is a ratio of the absolute WSS and the mean WSS. The mean WSS is calculated as the magnitude of the time-averaged surface traction vector $\tau_{\text{mean}} = \|(1/T)\int_0^T \mathbf{t}_s\,\mathrm{d}t\|$ and the OSI is defined by as [110]:

$$\text{OSI} = \frac{1}{2}\left(1 - \frac{\tau_{\text{mean}}}{\tau_{\text{abs}}}\right) \tag{44}$$

The logarithmic time-averaged WSS distribution is given in Figure 29 along with the OSI distribution. As expected, the peak time-averaged WSS occurs within the ICA in the vicinity of the flow divider. The predicted peak WSS of 805 dyne/cm^2 is exceptionally high and damaging. Upstream of the stenosis apex lies a band of high time-averaged WSS, across the ICA (400–500 dyne/cm^2). This is also greater than the threshold value of 315 dyne/cm^2 that according to Holme *et al.* [111] was sufficient to induce platelet activation and enhanced platelet thrombus formation. Immediately downstream of the stenosis lies a small region experiencing time-averaged WSS of less than 5 dyne/cm^2 and a bigger region experiencing less than 10 dyne/cm^2. This region is indicative of a potential location for further plaque formation and it corresponds to the region experiencing flow separation. Away from the stenosis, majority of the ICA experiences a WSS value less than 40 dyne/cm^2, although some regions experience values exceeding 100 dyne/cm^2 along the posterior wall. In the ECA, the time-averaged WSS values range between 20 and 50 dyne/cm^2. In contrast, approximately 20% of the CCA is predicted to experience a time-averaged WSS of less than 5 dyne/cm^2 with the vast majority experiencing less than 25 dyne/cm^2.

DOI: 10.1002/cnm

The OSI distribution indicates that the regions experiencing low time-averaged WSS in the CCA are coincident with the elevated OSI values. In the ICA, the regions of elevated or high OSI are extensive. A region extends from above the stenosis down the entire length of the inner wall to the ICA outlet. Another region of elevated OSI extends from near the stenosis apex along the outer wall of the ICA. In the ECA, only a small region experiences elevated OSI although this is located distal to the flow divider along the outer wall which is reminiscent of the regions experiencing low time-averaged WSS in 3 out of 4 of the carotids studied by Kaazempur-Mofrad *et al.* [105]. This region experiences time-averaged WSS of between 5 and 7 dyne/cm^2.

From the OSI distribution, it is clear that the disturbed flow is found within the ICA. With the flow separation inducing a low time-averaged WSS region immediately downstream of the stenosis, the potential for the stenosis to grow further in the downstream direction is high. Several small regions within the CCA experience low time-averaged WSS and elevated OSI making this a potential region of atheroma. The high time-averaged WSS values are also a potential issue, either through inducing further platelet deposition, thrombus formation or possible rupture.

7. CONCLUSION

A patient-specific modelling framework was presented in detail and all the aspects of this framework were thoroughly analysed. The major objective of the work was to present different aspects of the modelling pipeline that need little user interaction. The image processing method presented is not restricted by initialization invariance, and thus, easy to automate and need only an initial input from the user. Since the image segmentation is an implicit deformable model, topology changes in the blood vessel are automatically handled. As expected, generating a valid mesh for an accurate flow calculation involves a large number of stages. Since the image processor outputs level set functions, it was straightforward to use a advanced MC method to generate an initial surface mesh. However, cosmetics and refinement/derefinement of the surface mesh should be carried out with care to avoid any significant changes to the initial surface. The constrained Taubing smoothing was a good example of the care the meshing requires. Every aspect of the meshing and related techniques such as skeletonization, cropping, boundary layer meshing and general volume meshing were carried out with great care. Although the meshing procedure presented was lengthy in nature, it may be automated provided that there is a sufficient number of checks in the code. The flow solution discussed captured all essential flow physics normally expected for such a geometry. Since the solution was obtained on a static geometry the effects of wall motion on the mesh and flow solution are unknown.

The authors are not underestimating the challenge of constructing a valid geometry and mesh from a standard, clinical quality scan. For scans with poor quality and/or with aftereffects, the deformable model-based image processing may need additional support from emerging statistical methods such as ‘prior’ driven segmentation or graph-cuts. It is also important to investigate the procedures for merging both image segmentation and meshing together into a single step. This may be possible by evolving a surface mesh or collection of surface points, along with the evolving deformable model. The flow solution needs to include the wall dynamics if the impact of the wall motion is to be studied. To do this, establishing a solution pipeline from dynamic images is essential. Such a pipeline would include image registration to determine the wall motion. The registered motion should then be translated to a moving mesh and to a solution algorithm that can deal with moving boundaries. This idea is not new but it has not been systematically investigated in the literature.

APPENDIX A: FLOW CHART FOR THE PIPELINE

Figure A1 shows the proposed semi-automatic modelling pipeline for blood flow through arteries.

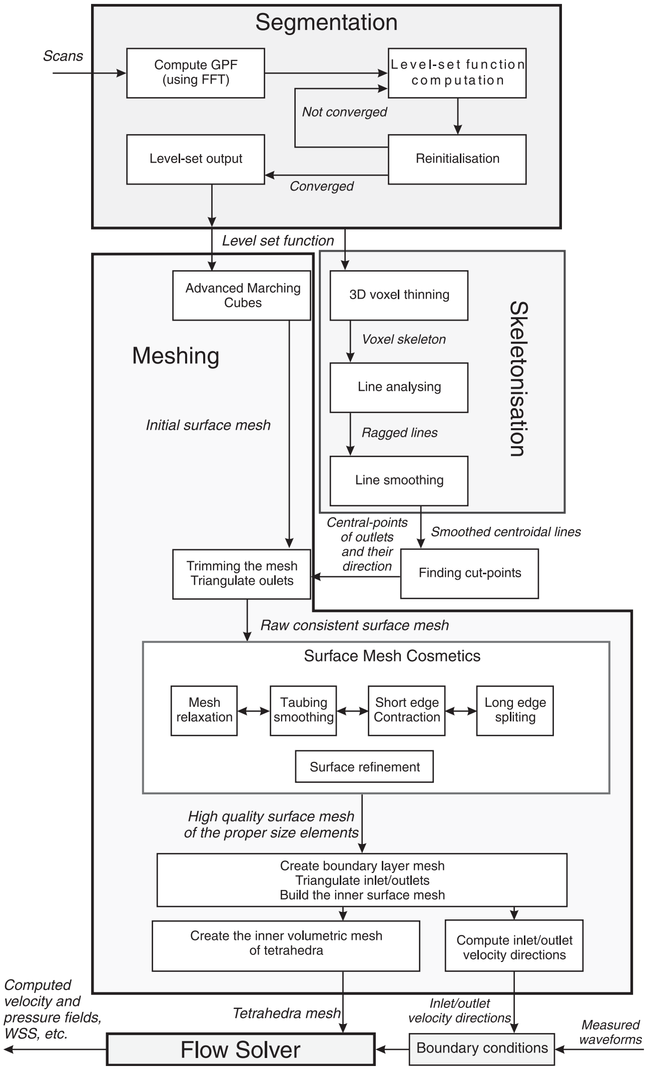
Segmentation
Scans
Compute GPF (using FFT)
Level-set function computation
Not converged
Level-set output
Reinitialisation
Converged
Level set function
Advanced Marching Cubes
3D voxel thinning
Meshing
Voxel skeleton
Line analysing
Skeletonisation
Initial surface mesh
Ragged lines
Line smoothing
Central-points of outlets and their direction
Smoothed centroidal lines
Trimming the mesh Triangulate oulets
Finding cut-points
Raw consistent surface mesh
Surface Mesh Cosmetics
Mesh relaxation
Taubing smoothing
Short edge Contraction
Long edge spliting
Surface refinement
High quality surface mesh of the proper size elements
Create boundary layer mesh
Triangulate inlet/outlets
Build the inner surface mesh
Create the inner volumetric mesh of tetrahedra
Compute inlet/outlet velocity directions
Computed velocity and pressure fields, WSS, etc.
Tetrahedra mesh
Inlet/outlet velocity directions
Measured waveforms
Flow Solver
Boundary conditions

Figure A1. Flow chart of the pipeline.

APPENDIX B: POINT PLACEMENT

An edge $e=\{p_1, p_2\}$ of a mesh is given and point $\boldsymbol{p}_\xi$ is an arbitrary point on the edge. Here p_1 and p_2 are two contiguous nodes (Figure B1) of the mesh with coordinates $\boldsymbol{p}_1, \boldsymbol{p}_2$. Now, $\boldsymbol{p}_\xi$ may be expressed as (a similar method is proposed in [112]):

$$\boldsymbol{p}_\xi = \boldsymbol{p}_1(1-\xi)+\boldsymbol{p}_2\xi, \quad 0\leqslant\xi\leqslant 1. \tag{B1}$$

In order to find the position of the point $\boldsymbol{p}_S(\xi)$ on the mesh surface S, following steps are adopted in this study.

1. Determine the normal direction to surface S at points $\boldsymbol{p}_1, \boldsymbol{p}_2$. We approximate it by averaging unit normal vectors $\hat{\boldsymbol{n}}_f$ to all faces containing this point (denote set of such faces as $\mathscr{F}_p$).

$$\boldsymbol{n}_p = \sum_{f\in\mathscr{F}_p} \hat{\boldsymbol{n}}_f, \quad \hat{\boldsymbol{n}}_p = \frac{\boldsymbol{n}_p}{\|\boldsymbol{n}_p\|} \tag{B2}$$

DOI: 10.1002/cnm

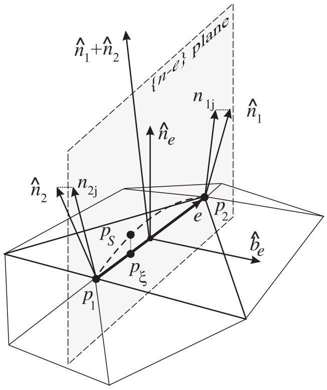

Figure B1. To the placing a new point onto the surface S.

Compute unit normal vectors $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$ to the edge endpoints $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, respectively, by Equation (B2). Thus, we have

$$\boldsymbol{n}_1=\sum_{f\in\mathscr{F}_{p_1}}\hat{\boldsymbol{n}}_f,\quad \hat{\boldsymbol{n}}_1=\frac{\boldsymbol{n}_1}{\|\boldsymbol{n}_1\|} \tag{B3}$$

$$\boldsymbol{n}_2=\sum_{f\in\mathscr{F}_{p_2}}\hat{\boldsymbol{n}}_f,\quad \hat{\boldsymbol{n}}_2=\frac{\boldsymbol{n}_2}{\|\boldsymbol{n}_2\|} \tag{B4}$$

2. Find the normal vector to the surface at the edge e. This can be calculated using

$$\boldsymbol{n}=\hat{\boldsymbol{n}}_1+\hat{\boldsymbol{n}}_2 \tag{B5}$$

Vectors $\boldsymbol{e}=\boldsymbol{p}_2-\boldsymbol{p}_1$ and $\boldsymbol{n}$ form a plane (the $\{e\text{-}n\}$ plane) with the unit normal

$$\hat{\boldsymbol{b}}_e=\frac{\boldsymbol{e}\times\boldsymbol{n}}{\|\boldsymbol{e}\times\boldsymbol{n}\|} \tag{B6}$$

3. Note that in general case $\boldsymbol{n}\cdot\boldsymbol{e}\neq 0$. Therefore, we compute an unit vector $\hat{\boldsymbol{n}}_e$ lying in the $\{e-n\}$ plane orthogonally to the edge e as

$$\hat{\boldsymbol{n}}_e=\hat{\boldsymbol{b}}_e\times\hat{\boldsymbol{t}}_e \tag{B7}$$

where $\hat{\boldsymbol{t}}_e=\boldsymbol{e}/\|\boldsymbol{e}\|$ is a unit vector along the edge (Figure B1).

4. Compute projection of $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$ onto the $\{e-n\}$ plane, i.e.

$$\boldsymbol{n}_{1j}=\hat{\boldsymbol{n}}_1-(\hat{\boldsymbol{n}}_1\cdot\hat{\boldsymbol{b}}_e)\hat{\boldsymbol{b}}_e,\quad \hat{\boldsymbol{n}}_{2j}=\boldsymbol{n}_2-(\hat{\boldsymbol{n}}_2\cdot\hat{\boldsymbol{b}}_e)\hat{\boldsymbol{b}}_e \tag{B8}$$

5. Find equation for the curve in the $\{e-n\}$ plane passing through points $\boldsymbol{p}_1,\boldsymbol{p}_2$ normal to $\boldsymbol{n}_{1j},\boldsymbol{n}_{2j}$ using,

$$\boldsymbol{p}_S(\xi)=\boldsymbol{p}_\xi+\hat{\boldsymbol{n}}_e\ Z(\xi). \tag{B9}$$

The function $Z(\xi)$ in the above equation is approximated by a cubic polynomial,

$$Z(\xi)=z_0+z_1\xi+z_2\xi^2+z_3\xi^3. \tag{B10}$$

The equations for determination of $z_0,\ldots,z_3$ can be obtained using the following procedure.

The curve passes through $\boldsymbol{p}_1, \boldsymbol{p}_2$ at $\xi = 0$ and $\xi = 1$, respectively, and hence we have

$$z_0 = 0 \quad \text{and} \quad z_0 + z_1 + z_2 + z_3 = 0 \tag{B11}$$

Compute the tangent vector to this curve (prime denotes derivative with respect to ξ) as:

$$\boldsymbol{t}(\xi) = \boldsymbol{p}'(\xi) = \boldsymbol{e} + \hat{\boldsymbol{n}}_e \; z'(\xi) \tag{B12}$$

Conditions of orthogonality at $\xi = 0$ and $\xi = 1$ are, respectively,

$$\boldsymbol{t}(0) \cdot \boldsymbol{n}_{1j} = 0 \quad \text{and} \quad \boldsymbol{t}(1) \cdot \boldsymbol{n}_{2j} = 0$$

which gives

$$(\boldsymbol{e} \cdot \boldsymbol{n}_{1j}) + (\hat{\boldsymbol{n}}_e \cdot \boldsymbol{n}_{1j}) \; Z'(0) = 0 \Longrightarrow Z'(0) = -\frac{\boldsymbol{e} \cdot \boldsymbol{n}_{1j}}{\hat{\boldsymbol{n}}_e \cdot \boldsymbol{n}_{1j}} \tag{B13}$$

$$(\boldsymbol{e} \cdot \boldsymbol{n}_{2j}) + (\hat{\boldsymbol{n}}_e \cdot \boldsymbol{n}_{2j}) \; Z'(1) = 0 \Longrightarrow Z'(1) = -\frac{\boldsymbol{e} \cdot \boldsymbol{n}_{2j}}{\hat{\boldsymbol{n}}_e \cdot \boldsymbol{n}_{2j}}. \tag{B14}$$

As soon as $Z'(0) = z_1$ and $Z'(1) = z_1 + 2z_2 + 3z_3$, we obtain a system of linear algebraic equations

$$\begin{aligned} z_0 &= 0 \\ z_0 + z_1 + z_2 + z_3 &= 0 \\ z_1 &= -Y_1 \\ z_1 + 2z_2 + 3z_3 &= -Y_2 \end{aligned} \tag{B15}$$

where

$$Y_1 = \frac{\boldsymbol{e} \; \boldsymbol{n}_{1j}}{\hat{\boldsymbol{n}}_e \; \boldsymbol{n}_{1j}} \tag{B16}$$

$$Y_2 = \frac{\boldsymbol{e} \; \boldsymbol{n}_{2j}}{\hat{\boldsymbol{n}}_e \; \boldsymbol{n}_{2j}} \tag{B17}$$

The solution to (B15) is

$$z_0 = 0, \quad z_1 = -Y_1, \quad z_2 = 2Y_1 + Y_2, \quad z_3 = -Y_1 - Y_2 \tag{B18}$$

Finally,

$$\boldsymbol{p}_S(\xi) = [\boldsymbol{p}_1(1-\xi) + \boldsymbol{p}_2\xi] + \hat{\boldsymbol{n}}_e[-Y_1\xi + (2Y_1 + Y_2)\xi^2 - (Y_1 + Y_2)\xi^3] \tag{B19}$$

In the case of mid-point $\xi = 0.5$, this formula turns into

$$\boldsymbol{p}_S(0.5) = \boldsymbol{p}_{1/2} + \hat{\boldsymbol{n}}_e[\tfrac{1}{8}Y_2 - \tfrac{1}{8}Y_1] \tag{B20}$$

where $\boldsymbol{p}_{1/2} = \frac{1}{2}(\boldsymbol{p}_1 + \boldsymbol{p}_2)$ is the mid-point of the edge e.

APPENDIX C: TABLE OF WAVE FORM HARMONICS

The data listed here in Table CI are from [97]. Here, n is the number of harmonic components, ω_n is the circular frequency (in Hz) for every component (for $T = 0.92\,\text{s}$), complex amplitudes $\tilde{U}_n$ are given in polar form $\tilde{U}_n = |\tilde{U}_n| \exp\{\mathrm{i} \arg(\tilde{U}_n)\}$.

Table CI. Flow harmonics.

n	$\omega_n/2\pi$	$\lvert\tilde{U}_n\rvert$	$\arg\tilde{U}_n$
0	0.00000	46.92637	0.00000
1	1.08755	21.52413	−1.17575
2	2.17510	17.61459	−1.63996
3	3.26264	12.14771	−2.41262
4	4.35019	6.76785	−2.59232
5	5.43774	9.01330	−2.86544
6	6.52529	8.01553	2.57922
7	7.61283	4.42009	2.02713
8	8.70038	3.57114	1.92789
9	9.78793	3.43206	1.38781
10	10.87548	2.32722	0.77936
11	11.96302	1.42878	0.62628
12	13.05057	1.58177	0.35280
13	14.13812	1.29647	−0.36075
14	15.22567	0.71691	−0.75220
15	16.31321	0.64928	−0.87985
16	17.40076	0.55891	−1.45062
17	18.48831	0.35657	−1.87787
18	19.57586	0.27657	−2.05181
19	20.66340	0.25372	−2.46117
20	21.75095	0.18541	−2.99914
21	22.83850	0.11847	3.00617
22	23.92605	0.10273	2.74036
23	25.01359	0.07701	2.23015
24	26.10114	0.04913	1.92394

ACKNOWLEDGEMENTS

This work is supported by EPSRC grants D070554 and H024271. This work is also partially supported by the Leverhulme Trust grant F/00391/R. The anonymized CT images used in this study are from Heart and Lung Centre of Wolverhampton Hospital. The support of Mr Heyman Luckraz in this regard is gratefully acknowledged.

REFERENCES

1. Steinman DA. Image-based computational fluid dynamics modeling in realistic arterial geometries. *Annals of Biomedical Engineering* 2002; **30**(4):483–497.
2. Cebral JR, Hernandez M, Frangi AF. Computational analysis of blood flow dynamics in cerebral aneurysms from cta and 3d rotational angiography image data. *International Congress on Computational Bioengineering*, Zaragosa, Spain, 24–26 September 2003; 191–198.
3. Mynard JP, Nithiarasu P. A 1d arterial blood flow model incorporating ventricular pressure, aortic valve and regional coronary flow using the locally conservative Galerkin (lcg) method. *Communications in Numerical Methods in Engineering* 2008; **24**:367–417.
4. Taylor CA, Figueroa CA. Patient-specific modeling of cardiovascular mechanics. *Annual Review of Biomedical Engineering* 2009; **11**:109–134.
5. Taylor CA, Steinman DA. Image-based modeling of blood flow and vessel wall dynamics: applications, methods and future directions. *Annals of Biomedical Engineering* 2010; **38**:1188–1203.
6. Gambaruto AM, Moura A, Sequeira A. Topological flow structures and stir mixing for steady flow in a peripheral bypass graft with uncertainty. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:926–953.
7. Nithiarasu P, Löhner R. Special issue on patient-specific computational modelling. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:1–2.
8. Radaelli AG, Peiro J. On the segmentation of vascular geometries from medical images. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:3–34.
9. Gee MW, Foerster Ch, Wall WA. A computational strategy for prestressing patient-specific biomechanical problems under finite deformation. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:52–72.
10. Mut F, Aubry R, Löhner R, Cebral JR. Fast numerical solutions of patient-specific blood flows in 3D arterial systems. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:73–85.

11. Takizawa K, Christopher J, Tezduyar TE, Sathe S. Space–time finite element computation of arterial fluid–structure interactions with patient-specific data. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:101–116.
12. Mori D, Yamaguchi T. Construction of the cfd model of the aortic arch based on mr images and simulation of the blood flow. *International Workshop on Medical Imaging and Augmented Reality*, Hong Kong, 10–12 June 2001; 111–116.
13. Tokuda Y, Song M-H, Ueda Y, Usui A, Toshiaki A, Yoneyama S, Maruyama S. Three-dimensional numerical simulation of blood flow in the aortic arch during cardiopulmonary bypass. *European Journal of Cardio-thoracic Surgery* 2008; **33**(2):164–167.
14. Wang KC, Dutton RW, Taylor CA. Improving geometric model construction for blood flow modeling. *IEEE Engineering in Medicine and Biology Magazine* 1999; **18**(6):33–39.
15. Xu XY, Long Q, Collins MW, Bourne M, Griffith TM. Reconstruction of blood flow patterns in human arteries. *Proceedings of the Institution of Mechanical Engineers H*, *Journal of Engineering in Medicine* 1999; **213**(5):411–421.
16. Augst AD, Barratt DC, Hughes AD, McG Thom SA, Xy XY. Various issues relating to computational fluid dynamics simulations of carotid bifurcation flow based on models reconstructed from three-dimensional ultrasound images. *Proceedings of the Institution of Mechanical Engineers H*, *Journal of Engineering in Medicine* 2003; **217**(5):393–403.
17. Younis HF, Kaazempur-Mofrad MR, Chan RC, Isasi AG, Hinton DP, Chau AH, Kim LA, Kamm RD. Hemodynamics and wall mechanics in human carotid bifurcation and its consequences for atherogenesis: investigation of inter-individual variation. *Biomechanics and Modeling in Mechanobiology* 2004; **3**(1):17–32.
18. Giordana S, Sherwin SJ, Peiro J, Doorly DJ, Papaharilaou Y, Caro CG, Watkins N, Cheshire N, Jackson M, Bicknall C, Zervas V. Automated classification of peripheral distal by-pass geometries reconstructed from medical data. *Journal of Biomechanics* 2005; **38**(1):47–62.
19. Peiro J, Sherwin SJ, Giordana S. Automatic reconstruction of a patient-specific high-order surface representation and its application to mesh generation for cfd calculations. *Medical and Biological Engineering and Computing* 2008; **46**(11):1069–1083.
20. Nanduri JR, Pino-Romainville FA, Celik I. Cfd mesh generation for biological flows: geometry reconstruction using diagnostic images. *Computers and Fluids* 2009; **38**(5):1026–1032.
21. Yi J, Ra JB. A locally adaptive region growing algorithm for vascular segmentation. *International Journal of Imaging Systems and Technology* 2003; **13**(4):208–214.
22. Sekiguchi H, Sugimoto N, Eiho S, Hanakawa T, Urayama S. Blood vessel segmentation for head mra using branch-based region growing. *Systems and Computers in Japan* 2005; **36**(5):80–88.
23. Abdel-Dayem AR, El-Sakka MR, Fenster A. Watershed segmentation for carotid artery ultrasound images. *IEEE International Conference on Computer Systems and Applications*, Cairo, Egypt, 3–6 January 2005.
24. Ding S, Tu J, Cheung CP, Beare R, Phan T, Reutens D, Thien F. Geometric model generation for cfd simulation of blood and air flows. *International Conference on Bioinformatics and Biomedical Engineering*, 2007; 1335–1338.
25. Ladak HM, Milner JS, Steinman DA. Rapid three-dimensional segmentation of the carotid bifurcation from serial mr images. *Journal of Biomechanical Engineering* 2000; **122**(1):96–99.
26. Gil JD, Ladak HM, Steinman DA, Frenster A. Accuracy and variability assessment of a semiautomatic technique for segmentation of the carotid arteries from three-dimensional ultrasound images. *Medical Physics* 2000; **27**(6):1333–1342.
27. Steinman DA, Thomas JB, Ladak HM, Milner JS, Rutt BK, Spence JD. Reconstruction of carotid bifurcation hemodynamics and wall thickness using computational fluid dynamics and mri. *Magnetic Resonance in Medicine* 2002; **47**(1):149–159.
28. Yim PJ, Cebral JJ, Mullick R, Marcos HB, Choyke RL. Vessel surface reconstruction with a tubular deformable model. *IEEE Transactions on Medical Imaging* 2001; **20**(12):1411–1421.
29. Cebral JR, Lohner R, Soto O, Choyke PL, Yim PJ. Patient-specific simulation of carotid artery stenting using computational fluid dynamics. *International Conference on Medical Image Computing and Computer Assisted Intervention*, The Netherlands, 14–17 October 2001; 153–160.
30. Cebral JR, Summers RM. Tracheal and central bronchial aerodynamics using virtual bronchoscopy and computational fluid dynamics. *IEEE Transactions on Medical Imaging* 2004; **23**:1021–1033.
31. Nilsson B, Heyden A. A fast algorithm for level set-like active contours. *Pattern Recognition Letters* 2003; **24**(9):1311–1337.
32. Antiga L, Ene-Iordache A, Remuzzi B. Computational geometry for patient-specific reconstruction and meshing of blood vessels from mr and ct angiography. *IEEE Transactions on Medical Imaging* 2003; **22**(5):674–684.
33. Deschamps T, Schwartz P, Trebotich D, Colella P, Saloner D, Malladi R. Vessel segmentation and blood flow simulation using level-sets and embedded boundary methods. *Computer Assisted Radiology and Surgery* 2004; **1268**:75–80.
34. Svensson J, Gardhagen R, Heiberg E, Ebbers T, Loyd D, Lanne T, Karlsson M. Feasibility of patient specific aortic blood flow cfd simulation. *International Conference on Medical Image Computing and Computer Assisted Intervention*, Copenhagen, Denmark, 1–6 October 2006; 257–263.
35. Antiga L, Piccinelli M, Botti L, Ene-Iordache B, Remuzzi A, Steinman DA. An image-based modeling framework for patient-specific computational hemodynamics. *Medical and Biological Engineering and Computing* 2008; **46**(11):1097–1112.

36. Malladi R, Sethian JA, Vemuri BC. Shape modelling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1995; **17**(2):158–175.
37. Yeo SY, Xie X, Sazonov I, Nithiarasu P. Geometric potential force for the deformable model. *British Machine Vision Conference*, London, 8 September 2009.
38. Yeo SY, Xie X, Sazonov I, Nithiarasu P. Geometrically induced force interaction for three-dimensional deformable models. *IEEE Transactions of Image Processing* 2010; DOI: 10.1109/TIP.2010.2092434.
39. Xie X, Mirmehdi M. Mac: magnetostatic active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2008; **30**:632–646.
40. Bevan RLT, Nithiarasu P, Van Loon R, Sazonov I, Luckraz H, Garnham A. Application of a locally conservative Galerkin (lcg) method for modelling blood flow through a patient-specific carotid bifurcation. *International Journal for Numerical Methods in Fluids* 2010; **64**:1274–1295.
41. Bevan RLT, Sazonov I, Saksono PH, Nithiarasu P, Van Loon R, Luckraz H, Ashraf S. Patient-specific blood flow simulation through an aneurysmal thoracic aorta with a folded proximal neck. *International Journal for Numerical Methods in Biomedical Engineering* 2011; DOI: 10.1002/cnm.1425.
42. Cebral JR, Löhner R. From medical images to anatomically accurate finite element grids. *International Journal for Numerical Methods in Engineering* 2001; **51**:985–1008.
43. Ito Y, Shum PC, Shih AM, Soni BK, Nakahashi K. Robust generation of high-quality unstructured meshes on realistic biomedical geometry. *International Journal for Numerical Methods in Engineering* 2006; **65**(6):943–973.
44. Bekkers EJ, Taylor CA. Multiscale vascular surface model generation from medical imaging data using hierarchical features. *IEEE Transactions on Medical Imaging* 2008; **27**(3):331–341.
45. Bazilevs Y, Hsu MC, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen JG. A fully-coupled fluid–structure interaction simulation of cerebral aneurysms. *Computational Mechanics* 2010; **46**(1):3–16.
46. Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics* 1987; **21**(4):163–169.
47. Zhang Y, Bazilevs Y, Goswamia S, Bajajb CL, Hughes TJR. Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(29–30):2943–2959.
48. Löhner R. Regridding surface triangulations. *Journal of Computational Physics* 1996; **126**:1–10.
49. Remacle J-F, Geuzaine C, Compère G, Marchandise E. High-quality surface remeshing using harmonic maps. *International Journal for Numerical Methods in Engineering* 2010; **83**:403–425.
50. Marchandise E, Compere G, Willemet M, Bricteux G, Geuzaine C, Remacle JF. Quality meshing based on STL triangulations for biomedical simulations. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:876–889.
51. Wang D, Hassan O, Morgan K, Weatherill NP. EQSM: an effective high quality surface grid generation method based on remeshing. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5621–5633.
52. Wang D, Hassan O, Morgan K, Weatherill NP. Enhanced remeshing from stl foles with applications to surface grid generation. *Communications in Numerical Methods in Engineering* 2007; **23**:227–239.
53. Hermann LR. Laplacian-isoparametric grid generation scheme. *Journal of the Engineering Mechanics Division of the American Society of Civil Engineers* 1976; **102**:749–756.
54. Taubing G. A signal approach to fair surface design. *Proceedings of SIGGRAPH*, Los Angeles, CA, U.S.A., 6–11 August 1995; 351–358.
55. Saksono PH, Nithiarasu P, Sazonov I, Yeo S-Y. Computational flow studies in a subject-specific human upper airway using a one-equation turbulence model. Influence of the nasal cavity. *International Journal for Numerical Methods in Engineering* 2011; DOI: 10.1002/nme.2986.
56. Ku DN, Giddens DP, Zarins CK, Glagov S. Pulsatile flow and atherosclerosis in the human carotid bifurcation—positive correlation between plaque location and low and oscillating shear-stress. *Atherosclerosis* 1985; **5**(3):293–302.
57. Nagel T, Resnick N, Dewey CF, Jr., Gimbrone Jr MA. Vascular endothelial cells respond to spatial gradients in fluid shear stress by enhanced activation of transcription factors. *Arterisoscelerosis*, *Thrombosis*, *and Vascular Biology* 1999; **19**:1825–1834.
58. Hwang J, Saha A, Boo YC, Sorescu GP, McNally JS, Holland SM, Dikalov S, Giddens DP, Griendling KK, Harrison DG, Jo H. Oscillatory shear stress stimulates endothelial production of from p47phox-dependent nad(p)h oxidases, leading to monocyte adhesion. *Journal of Biological Chemistry* 2003; **278**:47291–47298.
59. Hwang J, Ing MH, Salazar A, Lassègue B, Griendling K, Navab M, Sevanian A, Hsiai TK. Pulsatile versus oscillatory shear stress regulates nadph oxidase subunit expression. *Circulation Research* 2003; **93**:1225–1232.
60. Nguyen KT, Clark CD, Chancellor TJ, Papavassiliou DV. Carotid geometry effects on blood flow and on risk for vascular disease. *Journal of Biomechanics* 2008; **41**:11–19.
61. Zhang Y, Wang W, Liang X, Bazilevs Y, Hsu M-C, Kvamsdal T, Brekken R, Isaksen JG. High-fidelity tetrahedral mesh generation from medical imaging data for fluid–structure interaction analysis of cerebral aneurysms. *CMES—Computer Modeling in Engineering and Sciences* 2009; **42**:131–150.
62. Sahni O, Jansen KE, Taylor CA, Shephard MS. Automated adaptive cardiovascular flow simulations. *Engineering with Computers* 2009; **25**:25–36.
63. Dyedov V, Einstein DR, Jiao X, Kuprat AP, Carson JP, del Pin F. Variational generation of prismatic boundary-layer meshes for biomedical computing. *International Journal for Numerical Methods in Engineering* 2009; **79**(8):907–945.

64. Weatherill NP, Hassan O. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering* 1994; **37**:2005–2039.
65. Freitag L, Ollovier-Gooch C. Tetrahedral mesh improvement using face swapping and smoothing. *International Journal for Numerical Methods in Engineering* 1997; **40**:3979–4002.
66. Lloyd S. Least square quantization in PCM. *IEEE Transactions on Information Theory* **28**:29–137.
67. Du Q, Faber V, Gunzburger M. Centroidal Voronoï tessellations: applications and algorithms. *SIAM Review* 1999; **41**(4):637–676.
68. Nithiarasu P. An efficient artificial compressibility (ac) scheme based on the characteristic based split (cbs) method for incompressible flows. *International Journal for Numerical Methods in Engineering* 2003; **56**:1815–1845.
69. Nithiarasu P. A simple locally conservative Galerkin(lcg) finite element method for transient conservation equations. *Numerical Heat Transfer*, *Part B Fundamentals* 2004; **46**:357–370.
70. Nithiarasu P, Liu C-B. An explicit characteristic based split (cbs) scheme for incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2961–2982.
71. Nithiarasu P. A matrix free fractional step method for static and dynamic compressible solid mechanics. *International Journal for Computational Methods in Engineering Science and Mechanics* 2006; **7**:369–380.
72. Nithiarasu P, Liu C-B, Massarotti N. Laminar and turbulent flow through a model human upper airway. *Communications in Numerical Methods in Engineering* 2007; **23**:1057–1069.
73. van Loon R, van de Vosse FN. Special Issue: Fluid–structure interaction in biomedical applications. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:273–275.
74. Degroote J, Swillens A, Bruggeman P, Haelterman R, Segers P, Vierendeels J. Simulation of fluid structure interaction with the interface artificial compressibility method. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:276–289.
75. Wang T, Pan TW, Glowinski R. A fictitious domain method for simulating viscous flow in a constricted elastic tube subject to a uniform external pressure. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:290–304.
76. Kuettler U, Gee M, Foerster Ch, Comerford A, Wall WA. Coupling strategies for biomedical fluid–structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:305–321.
77. Wolters BJBM, Rutten MCM, Schurink GWH, van de Vosse FN. Computational modelling of endoleak after endovascular repair of abdominal aortic aneurysms. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:322–335.
78. Torii R, Oshima M, Kobayashi T, Takagi K, Tezduyar TE. Influence of wall thickness on fluid–structure interaction computations of cerebral aneurysms. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:336–347.
79. Einstein DR, Del Pin F, Jiao X, Kuprat AP, Carson JP, Kunzelman KS, Cochran RP, Guccione JM, Ratcliffe MB. Fluid–structure interactions of the mitral valve and left heart: comprehensive strategies, past, present and future. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:348–380.
80. Yin M, Luo XY, Wang TJ, Watton PN. Effects of flow vortex on a chorded mitral valve in the left ventricle. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:381–404.
81. van Loon R. Towards computational modelling of aortic stenosis. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:405–420.
82. Vigmostad SC, Udaykumar HS, Lu J, Chandra KB. Fluid–structure interaction methods in biological flows with special emphasis on heart valve dynamics. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:435–470.
83. Crowl LM, Fogelson AL. Computational model of whole blood exhibiting lateral platelet motion induced by red blood cells. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:471–487.
84. Kamada H, Tsubota Ki, Nakamura M, Wada S, Ishikawa T, Yamaguchi T. A three-dimensional particle simulation of the formation and collapse of a primary thrombus. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:488–500.
85. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on the Hamilton–Jacobi formulation. *Journal of Computational Physics* 1988; **79**:12–49.
86. Sethian JA. *Level Set Methods and Fast Marching Methods*: *Evolving Interfaces in Computational Geometry*, *Fluid Mechanics*, *Computer Vision*, *and Material Science*. Cambridge University Press: Cambridge, U.K., 1999.
87. Felzenszwalb PF, Huttenlocher DP. Distance transforms of sampled functions. *Cornell Computing and Information Science, TR2004-1963*, 2004.
88. She FH, Chen RH, Gao WM, Hodgson PH, Kong LX, Hong HY. Improved 3d thinning algorithms for skeleton extraction. *Digital Image Computing*: *Techniques and Applications* 2009; **0**:14–18.
89. Morgenthaler DG. Three dimensional simple points: serial erosion, parallel thinning and skeletonization. *Technical Report tr-1005*, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, 1981.
90. Lee CN, Poston T, Rosenfeld A. Winding and euler numbers for 2d and 3d images. *Computer Vision*, *Graphics and Image Processing*: *Graphical Models and Image Processing* 1991; **53**(6):522–537.
91. Huang A, Liu H-M, Lee Ch-W, Tang Ch-Y, Tsang Y-M. On conside 3d simple point characterizations: a marching cubes paradigm. *IEEE Transaction on Medical Imaging* 2009; **28**:43–51.
92. Chernyaev EV. Marching cubes 33: construction of topologically correct isosurfaces. *Technical Report cern cn 95-17*, CERN, 1995.

DOI: 10.1002/cnm

93. Nielson GM, Hamann B. The asymptotic decider: resolving the ambiquity in marching cubes. *Proceedings of Visualiration91*. IEEE Computer Society Press: Silver Spring, MD, 1991; 83–90.
94. Sazonov I, Wang D, Hassan O, Morgan K, Weatherill NP. A stitching method for the generation of unstructured meshes for use with co-volume solution techniques. *Computer Methods in Applied Mechanics and Engineering Corresponding* 2006; **195**(13–16):1826–1845.
95. Frey WH, Field DA. Mesh relaxation: a new technique for improving triangulation. *International Journal for Numerical Methods in Engineering* 1991; **31**:1121–1133.
96. Womersley JR. Method for the calculation of velocity, rate flow, and viscous drag in arteries when the pressure gradient is known. *Journal of Physiology* 1955; **127**(3):553–563.
97. Holdsworth DW, Norley CJD, Frayne R, Steinman DA, Rutt BK. Characterization of common carotid artery blood-flow waveforms in normal human subjects. *Physiological Measurement* 1999; **20**(3):219–240.
98. Nithiarasu P. An arbitrary lagrangian eulerian (ale) method for free surface flow calculations using the characteristic based split (cbs) scheme. *International Journal for Numerical Methods in Fluids* 2005; **48**:1415–1428.
99. Zienkiewicz OC, Taylor RL, Nithiarasu P. *The Finite Element Method for Fluid Dynamics*. Elsevier Butterworth-Heinemann: Burlinton, MA, 2005.
100. Nithiarasu P, Hassan O, Morgan K, Weatherill NP, Fielder C, Whittet H, Ebden P, Lewis KR. Steady flow through a realistic human upper airway geometry. *International Journal for Numerical Methods in Fluids* 2008; **57**:631–651.
101. Zienkiewicz OC, Codina R. A general algorithm for compressible and incompressble flow—part I. The split, characteristic based scheme. *International Journal for Numerical Methods in Fluids* 1995; **20**:869–885.
102. Smedby O. Do plaques grow upstream or downstream? An angiographic study in the femoral artery. *Arteriosclerosis Thrombosis and Vascular Biology* 1997; **17**:912–918.
103. Glor FP, Long Q, Hughes AD, Augst AD, Ariff B, Thom SAMcG, Verdonck PR, Xu XY. Reproducibility study of magnetic resonance image-based computational fluid dynamics prediction of carotid bifurcation flow. *Annals of Biomedical Engineering* 2003; **31**:142–151.
104. Papathanasopoulou P, Zhao S, Köhler U, Robertson MB, Long Q, Hoskins P. MRI measurement of time-resolved wall shear stress vectors in a carotid bifurcation model, and comparison with CFD predictions. *Journal of Magnetic Resonance Imaging* 2003; **17**:153–162.
105. Kaazempur-Mofrad MR, Isasi AG, Younis HF, Chan RC, Hinton DP, Sukhova G, LaMuraglia GM, Lee RT, Kamm RD. Characterization of the atherosclerotic carotid bifurc ation using MRI finite element modeling, and histology. *Annals of Biomedical Engineering* 2004; **32**:932–946.
106. Urquiza SA, Blanco PJ, Vénere MJ, Feijóo RA. Multidimensional modelling for the carotid artery blood flow. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:4002–4017.
107. Jeays AD, Lawford PV, Gillott R, Spencer P, Barber DC, Bardhan KD, Hose DR. Characterisation of the haemodynamics of the superior mesenteric artery. *Journal of Biomechanics* 2007; **40**:1916–1926.
108. Lee KE, Parker KH, Caro CG, Sherwin SJ. The spectral/hp element modelling of steady flow in non-planar double bends. *International Journal for Numerical Methods in Fluids* 2008; **57**:519–529.
109. Malek AM, Alper SL, Izumo S. Hemodynamic shear stress and its role in atherosclerosis. *JAMA* 1999; **282**:2035–2042.
110. He X, Ku DN. Pulsatile flow in the human left coronary artery bifurcation: average conditions. *Journal of Biomechanical Engineering* 1996; **118**:74–82.
111. Holme PA, Orvim U, Hamers MJAG, Solum NO, Brosstad FR, Barstad RM, Sakariassen KS. Shear-induced platelet activation and platelet microparticle formation at blood flow conditions as in arteries with a severe stenosis. *Arteriosclerosis Thrombosis and Vascular Biology* 1997; **17**:646–653.
112. de Putter S, Laffargue F, Breeuwer M, van de Vosse FN, Gerritsen FA. Computational mesh generation for vascular structures with deformable surfaces. *International Journal of Computer Assisted Radiology and Surgery* 2006; **1**(1):39–49.

DOI: 10.1002/cnm