A (Near) Real-Time Simulation Method of Aneurysm Coil Embolization

Yiyi Wei, Stéphane Cotin, Jérémie Dequidt, Christian Duriez, Jérémie Allard, Erwan Kerrien

A (Near) Real-Time Simulation Method of Aneurysm Coil Embolization

Yiyi Wei, Stéphane Cotin, Jérémie Dequidt, Christian Duriez,
Jérémie Allard and Erwan Kerrien

Additional information is available at the end of the chapter

1. Introduction

1.1. Aneurysm

An aneurysm is an abnormal widening of a blood vessel. As the vessel widens, it also gets thinner and weaker, with an increasing risk of rupture. Aneurysms are essentially found in the aorta, the popliteal artery, mesenteric artery, and cerebral arteries. Intracranial aneurysms are smaller than other types of aneurysm and mostly saccular. Though most patients do not experience rupture, it can lead to a stroke, brain damage and potential death. The mortality rate after rupture is considerably high: the incidence of sudden death was estimated to be 12.4% and death rate ranged from 32% to 67% after the hemorrhage [16]. Each year, over 12,000 people die in the United States due to rupture of intracranial aneurysms [17].

In order to prevent the rupture, or rerupture, of an aneurysm, several treatments have proved successful: neurosurgical clipping, endovascular coiling and stenting. The aneurysm can be permanently sealed from the normal blood circulation by placing a tiny metal clip across the aneurysm neck. This open surgery requires to perform a craniotomy, which is invasive and associated with risks of complications during or shortly after surgery. In recent years, the development of interventional radiology techniques made it possible for a growing number of patients to be treated with minimally invasive strategies, essentially endovascular coiling. The procedure of coil embolization starts with the insertion of a catheter into the femoral artery, which is then advanced through the arterial system all the way to the location of the intracranial aneurysm. Then the radiologist delivers the filling material (the coils) through a micro-catheter into the aneurysm sac. The presence of coils in the aneurysm reduces blood velocity, and decreases the pressure against the aneurysmal wall, progressively creating a favorable hemodynamic environment for thrombus embolization. Finally, the formation of a blood clot blocks off the aneurysm, thus considerably reducing the risk of rupture. In the case of irregularly-shaped or fusiform aneurysms, or aneurysms with wide necks, stenting of the parent artery can be used in combination with coils. Although endovascular techniques

are less invasive than open surgery, they remain complex and risky to perform, requiring an important expertise and careful planning.

1.2. Computer-based medical simulation

Medical simulation provides a solution to the current need for residency training and procedure planning, by allowing trainees to experience realistic scenarios, and by repeatedly practicing without putting patients at risk. With the ongoing advances in biomechanics, algorithmics, computer graphics, software design and parallelism, computer-based medical simulation is playing an increasingly important role in this area, particularly by providing access to a wide variety of clinical scenarios, patient-specific data, and reduced training cost. Even for experienced physicians, medical simulation has the potential to provide planning and preoperative rehearsal for patient-specific cases. In some cases it can also offer some insight into the procedure outcome. Nevertheless, several fundamental problems remain to be solved for a wide and reliable use of computer-based planning systems in a clinical setup, and in particular for coil embolization. Such a planning system is expected to have a high level of realism and good predictive abilities. In particular we are interested in a prediction of the hemodynamics before and after the procedure, an evaluation of the number of coils required to achieve embolization, and an interactive simulation of coil deployment for rehearsal. This involves the following challenges:

- **Geometry modeling**: although 3D imaging of the patient's vasculature is now widely available under various modalities, extracting the actual geometry of the blood vessels is still an issue due to the limitations in spatial resolution and the presence of noises and artifacts. Particularly, accurate and exact geometry extraction of blood vessel is a challenge in the vicinity of intracranial aneurysms due to the small size and complex shape of the surrounding vascular network.

- *In vivo* **data**: for the purpose of realism, patient-specific *in vivo* data is necessary for biomechanical modeling and hemodynamic simulation, such as mechanical parameters of vessel wall, blood flow rate. Acquisition of *in vivo* data is quite challenging, because imaging techniques are not currently capable to provide images with a high resolution either in space or in time.

- **Fast computation**: the planning system targets at assistance for rapid decision making or rehearsal in a virtual environment. As such it requires fast or even real-time computation of blood flow and blood-structure interaction, which cannot always be guaranteed by modern computers with certain limitations in memory and frequency. Therefore, the need still remains to increase computing speed by optimizing algorithms or proposing alternative numerical methods.

- **Coil modeling**: modeling intracranial coils, which are thin platinum wires, remains challenging, because mechanical parameters are not provided by device manufacturers. As coils are designed to conform to the aneurysm wall, it is also important to compute the multiple contacts (or self-contacts) between the coils and the aneurysm.

1.3. Previous work

Generally speaking, there are three main approaches to obtain hemodynamic data. Experimental techniques have been widely used in clinical analysis, but restricted to idealized

geometries or surgically created structures in animals. However, a better method is *in vivo* analysis, which can be provided through medical imaging. For example, angiography can provide *in vivo* flow data, but only in 1D; Magnetic Resonance Imaging (MRI) data is 3D, but quite noisy. Finally, the computational approach, which can provide a 3D representation of detailed flow patterns in patient-specific geometry, becomes more and more attractive in this area.

To obtain patient-specific geometry, excellent review papers [21] reported on the vast literature that addressed blood vessel segmentation. The diversity in the methods reflected the variety of contexts in which the question of blood vessel segmentation araised, requiring us to be more specific about our expectancies. First, the vessel surface model should be smooth enough for its use in the simulation. Implicit surface representations, where the surface was defined as the zero-level set of a known function f, were arguably well suited [5]. C1-continuous models (with C0-continuous normal) allowed for much smoother sliding contacts. Unwanted friction might occur with polyhedral surfaces [26] or level-sets such as vesselness criteria [11, 34] defined on a discrete grid. Implicit surfaces also offered an improved collision management over parametric surfaces [10]. Indeed, the implicit function value at a point told whether this point was inside or outside the surface, detecting a collision in the latter case. Furthermore, the implicit function gradient gave a natural direction for the contact force used to handle this collision. In many cases, segmentation or vessel enhancement aimed at improving vessel visibility. When quantitative assessment was required, most previous works grounded on the same seminal idea as Masutani [22], which tracked the vessel centerline by fitting local vessel models: graphics primitives [43], convolution kernel [33], or cylinder templates [12]. Both the centerline location and radius estimation might be correct, but such models were unable to accurately cope with irregularities on the vessel surface, especially when the vessel section was not circular (or elliptic). Also, they did not correctly handle pathologies such as aneurysms. Various methods were proposed to improve the vessel delineation in cross-sections. Ray-casting methods were of particular interest, as they were able to extract candidate points at the vessel surface. Indeed, they opened the path to using a wealth of techniques developed by the graphics community to fit an implicit surface to a set of scattered surface points. Radial Basis Functions (RBF) had a promising potential, especially in their variational formulation [37] with the capacity to produce compact models. Closer to our work, Schumann [35] used Multi-level Partition of Unity (MPU) implicits to get a locally defined model. However, RBF or MPU implicit gradients gave appropriate contact directions close to the surface but could mislead contact forces elsewhere. We propose in Section 2 a new and efficient algorithm to model local blood vessels with blobby models [28].

Most computational methods for blood flow simulation relied on the science of Computational Fluid Dynamcis (CFD), and approximated blood flow as a continuous incompressible Newtonian fluid, described by the unsteady incompressible Navier-Stokes equations [23]. For instance, Groden et al. constructed a simple geometrical model by only straight cylinders and spheres to approximate an actual aneurysm, and simulated the flow by solving the Navier-Stokes equations [14]. The geometry model they used could not accurately describe the real patient's case, therefore, had little use in surgery planning for a specific patient. In the last ten years, the successful effort in the research of combining image processing and CFD made it possible to compute patient-specific hemodynamic information in this community. Kakalis et al. employed patient-specific data to get more realistic flow patterns [19]. However, both of their methods, as well as most similar studies, relied on

the CFD commercial software to simulate the flow, and the computational times (dozens of hours in general) were incompatible with interactive simulation or even clinical practice. In order to reach fast computation, several template-based methods were designed [24] [25]. These methods pre-computed hemodynamics on a set of similar template geometries, and then interpolated on a patient-specific geometry instead of fluid simulation. However, they required to set up a pre-computed database, and were only tested on simple artery structures. But for the brain vessels around intracranial aneurysm, the network and shape are much more complicated, so the high accuracy cannot be expected.

More methods for fast computation were proposed in the field of computer graphics, essentially required the results to be visually convincing, but not physically accurate. The stable fluids approach [39] was a significant milestone, as it brought in fluid advection and the Helmholtz-Hodge decomposition to ensure the mass conservation law. However, this approach relied on discretization of Eulerian space using a regular grid, thus making it inappropriate for complex geometry with irregular boundaries, as it is usually the case in anatomical structures. Recently, the Discrete Exterior Calculus (DEC) method, based on unstructured mesh for incompressible fluid simulation, was proposed in [9]. It presented several benefits in terms of stability and computational efficiency. However, the context, in which this method was applied, did not aimed at the practical use in medical simulation. We further assess the stability, accuracy and computational time of this method, more importantly, improve the method for medical applications in (near) real-time.

Previous work in the area of real-time or near real-time simulation for interventional radiology mainly focused on training, for instance, [29], [15], [1], [7] and [5], which proposed approaches for modeling either catheter deformation or more general catheter navigation in vascular networks. Real-time simulation of coil embolization was investigated more recently. However, additional difficulties need to be faced: the deformation of the coil is more complex, and the coil is self-colliding during embolization while it also has frictional contact with the vessel wall. A FEM model, based on beam element allowed to adequately capture the deformation of the coil [6]. This model was validated, and an inverse problem was solved to capture the correct rest-shape configuration. Moreover, a solution to contact and self-collision was proposed, which was based on the resolution of Signorini's law and Coulomb friction [4]. A Gaussian deformation, close to the Boussinesq approximation model, allowed to take into account the very small deformation of the aneurysm wall. This approach is developed in more details in Section 3. Recently, a deformation model for the coil, based on inextensible elastic rods was proposed [38], in combination with a geometric approach for coil embolization, based on path-planning [27].

1.4. Our contributions

In our work, we aim at real-time simulations of the blood flow and blood-structure interaction during aneurysm coil embolization in the patient-specific geometry. While the existing studies of aneurysm embolization only concerned the impact of the deployed coil(s) on the blood flow, we also present an effective way to compute the reverse effect of the blood flow on the coil during the medical procedure, and provide higher reality for the simulation of placing coils into the aneurysm compared to the case without considering the interactive force from blood flow.

First of all, a smooth and accurate model of the vessel wall around aneurysm is required. We propose to use a blobby model whose parameterization is constrained so that the implicit function varies with the distance to the vessel surface. Thereafter, a stable blob selection and subdivision process are designed. Also, an original energy formulation is given under closed-form, allowing for an efficient minimization.

Secondly, we present a coil model based on the serially-linked beams that can handle large geometric deformations. Our model can also handle various shape memories in order to simulate different types of coils (like helical, bird-cage or 3D coils). This model is computed efficiently by a backward Euler scheme combined with a linear solver that takes into account the particularities of our model.

Thirdly, in order to achieve accurate and near real-time blood flow simulation, we introduce the DEC method into hemodynamic simulation for the first time. Compared to the DEC method initially applied in the field of computer graphics, we improve the numerical stability by using more advanced backtracking schemes, and more importantly by optimizing quality of the mesh used in the computation. Moreover, a detailed analysis of the results and comparison with a reference software are performed to understand the stability and accuracy of the method, as well as the factors affecting these two aspects.

Finally, based on the DEC method, we propose a framework for real-time simulation of the interventional radiology procedure (Figure 1). We reconstruct 3D models of vascular structures, and propose a real-time finite element approach for computing the behavior of flexible medical devices, such as coils, catheters and guide wires. Then we present the methods for computing contacts between virtual medical devices and virtual blood vessels. Furthermore, we model bilateral interactions between blood flow and deformable medical devices for real-time simulation of coil embolization. Our simulated results of blood-coil interaction show that our approach permits to describe the influence between coils and blood flow during coil embolization, and that an optimal trade-off between accuracy and computation time can be obtained.

2. Geometry modeling

In this section, we present our work on vascular modeling using implicit representations. We show how such models can be obtained from actual patient data (3D rotational angiography, 3DRA).

### 2.1. Local implicit modeling

2.1.1. Implicit formulation as a blobby model

An implicit isosurface generated by a point-set skeleton is expressed as the zero-level set of a function f, a sum of implicit spheres:

$$f(X; p) = T - \sum_{j=1}^{N_b} \alpha_j \phi \left(\frac{|X - C_j|}{\rho_j} \right),$$
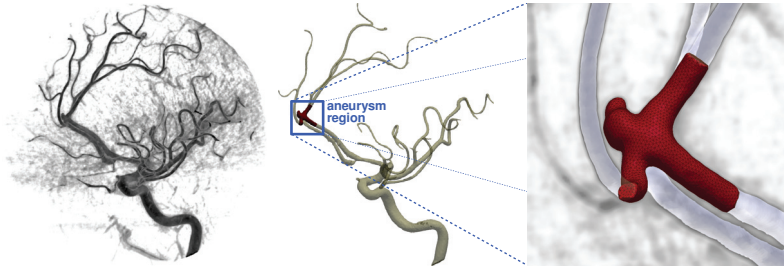
(a) From patient-specific data (left), we reconstruct vessel surface model (middle), and then generate tetrahedral mesh in the region of aneurysm (right).
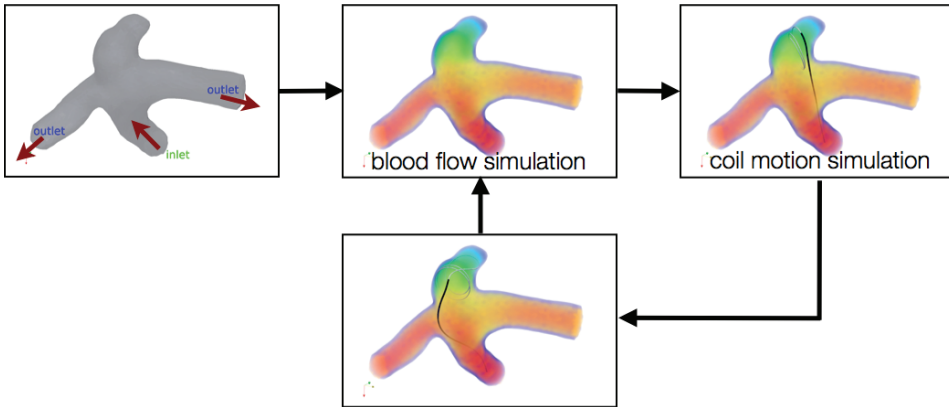
(b) Given the boundary conditions, we simulate the blood flow velocity, from which we compute the interactive force applied on the coil. Until there is a significant increase of coil density, we update the velocity field.

Figure 1. The framework of real-time simulation of coil embolization: (a) mesh generation; (b) simulation of bilateral interaction of blood and medical devices.

where T is the isosurface threshold, $\{\alpha_j\}$ are positive weights, and $\{C_j\}$ is the point set skeleton. Each implicit sphere #j is defined by a symmetric spherical function, centered on C_j, of width ρ_j. The local field function, or kernel, is a function $\phi : \mathbb{R} \to \mathbb{R}^+$, rapidly decreasing to 0 at infinity. Thus, model locality is ensured: an implicit sub-model can be extracted by merely selecting neighboring blobs. For example, all results presented below are produced using the 'Cauchy' kernel [36]:

$$\phi(x) = (1 + x^2/5)^{-2},$$

where the dividing factor 5 normalizes the kernel such that $\phi''(1) = 0$.

Such objets are called differently depending on the kernel used [36]. Our method is not kernel-dependent, and was successfully used with the computationally less efficient Gaussian kernel. Muraki [28] was the first to use this type of model in the context of object reconstruction. Following this seminal work, we will use the terms *blob* for an implicit sphere, and *blobby models* as a generic name for implicit models.
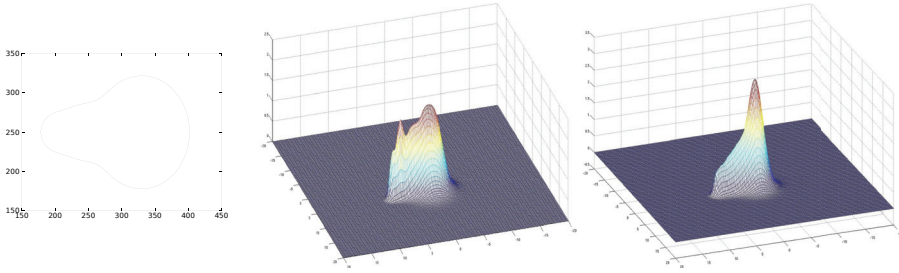
Figure 2. Redundancy in implicit function parameterization: (left): 2D shape to model; middle: implicit modeling using a uniform weight of 1 ($\alpha_j = 1$); right: implicit modeling using weights equal to radii ($\alpha_j = \rho_j$). The implicit function looks more like a distance function in the latter case.

2.1.2. Setting the weights

In the above formulation, there is a redundancy between the weight $\{\alpha_j\}$ and the radii $\{\rho_j\}$. Figure 2 gives an example of a 2D shape (left image) with two different implicit representations ($z = f(x, y)$) only parameterized by the center locations and the radii: for the image in the center, all the weights $\{\alpha_j\}$ were set to 1 ; while on the right we had $\alpha_j = \rho_j, \forall j$. This latter case is particularly interesting when we consider circles of different radii: there is a monotonously increasing relation between the distance function and the implicit function f.

In our particular simulation context, in order to help predict collisions, and have the function give a valid contact force direction, the algebraic value $f(X; p)$ at point X should relate monotonously to the geometric distance of X to the surface.

As a consequence, we set $\alpha_j = \rho_j$. Thereby, as demonstrated in Figure 2, the function gradient gives a valid contact direction anywhere in space (consider the crease in the middle of the shape in the middle image). Meanwhile, redundancy in the parameters of f is also dismissed. This choice for parameterization will also prove useful later to efficiently select the blob to be subdivided.

2.1.3. Energy formulation

Fitting a surface to N points $\{P_i\}_{1 \leq i \leq N_p}$ can be written as an energy minimization problem [28, 42]. We propose to combine three energy terms:

$$\mathcal{E} = \mathcal{E}_d + \alpha \mathcal{E}_c + \beta \mathcal{E}_a$$

, where $(\alpha, \beta) \in \mathbb{R}^{+2}$, and:

1.

$$\mathcal{E}_d = \frac{1}{N_p} \sum_i f(P_i; p)^2$$

, which translates the algebraic relation between data points and the zero-level set. It gives a raw expression of the approximation problem.

2.

$$\mathcal{E}_c = \frac{1}{(N_b(N_b - 1))} \sum_{j \neq k} \left(\frac{s\sqrt{\rho_j \rho_k}}{|C_k - C_j|} \right)^{12} - 2 \left(\frac{s\sqrt{\rho_j \rho_k}}{|C_k - C_j|} \right)^6$$

, which is Lennard-Jones energy. Each term is minimal (with value -1) for $|C_j - C_k| = s\sqrt{\rho_j \rho_k}$, being repulsive for blobs closer than this distance, and attractive for blobs further away. It imposes some cohesion between neighboring blobs to avoid leakage where data points are missing, while preventing blobs from accumulating within the model.

3.

$$\mathcal{E}_a = \frac{1}{N_p} \sum_i \kappa(P_i)^2$$

$\kappa(P)$ is the mean curvature. It can be computed in a closed form at any point in space from the implicit formulation [13]

$$\kappa(P) = \frac{\nabla f^t H_f \nabla f - |\nabla f|^2 trace(H_f)}{2|\nabla f|^3}$$

, where ∇f is the implicit function gradient and H_f its Hessian matrix, both computed at point P. This energy smoothes the surface according to the minimal area criterion. In particular, the wavy effect that could stem from modeling a tubular shape with implicit spheres, is reduced.

Behind the rather classical form given above for the energy terms, it is important to notice that the whole energy is known under a closed-form expression. As a consequence, closed-form expressions were derived for its gradients with respect to the blobby model parameters $\{\rho_j\}$ and $\{C_j\}$.

2.1.4. Selection-subdivision

The blob subdivision procedure proposed in the seminal work [28] was exhaustive and time consuming. A blob selection mechanism was added in [42], measuring the contribution of each blob to $\mathcal{E}_d$ within a user-defined window, and choosing the main contributor. We experimentally noted that this technique was prone to favor small blobs, thus focusing on details, before dealing with areas roughly approximated by one large blob. This behavior is caused by this selection mechanism using the algebraic distance to the implicit surface. Our criterion relies upon the geometric distance approximation proposed by Taubin [40]:

$$d_T(P, f_0) = \frac{|f(P; p)|}{|\nabla f(P; p)|},$$

where P is a point and f_0 is the 0-level set of implicit function f, parameterized by p. As a consequence, the point P_{i^*} farthest to the surface is such that:

$$i^* = \arg \max_{1 \leq i \leq N_p} d_T(P_i, f_0).$$

The blob #j^* whose isosurface is the closest to P_{i^*} is selected (according to Taubin's distance d_T). Note that this criterion is valid in large area because we set $\alpha_j = \rho_j$ in the definition of f. The subdivision step then replaces this blob with two new ones. Their width ρ'_{j^*} is chosen such that two blobs, centered on C_{j^*}, of width ρ'_{j^*} would have the same isosurface as one blob centered on C_{j^*}, with width ρ_{j^*} (the formula depends on the kernel). The first new blob is centered on C_{j^*}, while the second is translated by $\rho_{j^*}/10$ towards P_{i^*}.

(a) 2D case: initialization with 4 blobs

(b) 2D case: final result

(c) 3D case: initialization with one blob
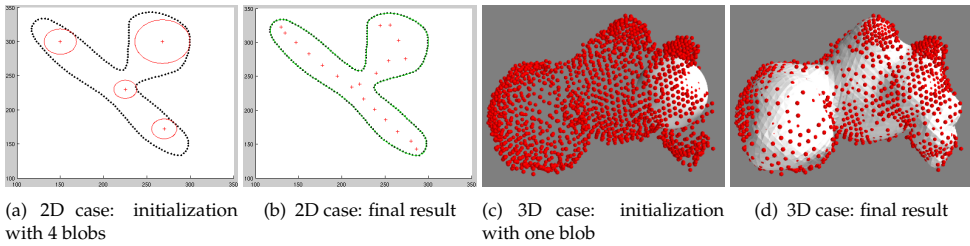
(d) 3D case: final result

Figure 3. Examples of implicit modeling. In the 2D case (two leftmost images), data points are in black, blob centers are red crosses, and the final implicit curve is in green. In the 3D case (two rightmost images), data points are in red and the implicit surface is in white.

2.1.5. Optimization

Such a gradual subdivision procedure may lead to a dramatic increase in the number of blobs, and hence the size of the optimization problem. The locality of the kernel ϕ allows us to focus the optimization onto the newly created pair of blobs. More exactly, only the new blob that is slightly misplaced is optimized, the other blobs remaining constant. The energy is minimized using Polak-Ribiere conjugate gradient (PR) algorithm, taking advantage of the closed-form expressions of both the energy and its gradients. A single minimization loop consists in one PR minimization over the center (3 variables), followed by one on the width (1 variable). In practice, a maximum of 5 loops proved sufficient.

2.2. Overall modeling algorithm

This fitting procedure proved to be very robust to initialization. Figure 3 gives two examples, one in 2D and the other in 3D, of typical initializations and results obtained.

The 2D case (Figure 3(a) and 3(b)) mimics the neighborhood of an aneurysm. Starting from 4 blobs dispatched on the aneurysm and parent vessel (Figure 3(a)), the final result on Figure 3(b) shows 20 blobs whose centers naturally span the skeletal line of the shape while providing a close fit (green line). The 3D case (Figure 3(c) and 3(d)) is an actual bilobated aneurysm. Starting from one blob located at the entrance of the aneurysm (Figure 3(c)), the procedure ends up on Figure 3(d) with 100 blobs closely fitting all bumps and other details in the shape.

Previous works [44] demonstrated that good candidate points at the vessel surface could be obtained by casting rays from center points inside the vessel, and keeping only the locations of minimal directional gradient (blood vessels are bright onto a darker background in angiographic images). The same strategy was followed, except that rays were thrown in directions regularly spaced on the unit 3D sphere. A user-defined threshold was applied to the gradient amplitude to remove extraneous points detected on small intensity variations that could be captured in large areas, such as the aneurysm sac or when rays were cast along the vessel direction.

As a consequence, modeling the vicinity of an aneurysm takes three steps:

1. Manually place initial blobs inside the aneurysm and related blood vessel. Only an approximate radius is needed for these blobs.

2. Extract data points by casting rays from the blob centers

3. Run the above implicit modeling algorithm

Once we get a smooth surface model, meshing the domain is done using the interleaved optimization algorithm based on Delaunay refinement and Lloyd optimization [41], and implemented using the CGAL library [1]. Each refinement step acts on the size of elements, while each optimization step acts on the shape of elements. Using this method, we can also define a size field to obtain anisotropic and non-uniform mesh. We control the fidelity of the generated mesh to the previously obtained surface model, by specifying the distance tolerance between the two surfaces.

3. Modeling of coil deformations during embolization

In this section, we describe the deformation model that is used to capture the mechanical properties of the coil. Moreover, as the behavior of the coil during embolization strongly depends on the contacts with the vessel wall, we presents a constraint-based collision response.

3.1. Coil model

There are different types of detachable coils but most of them have a core made of platinum, and some are coated with another material or a biologically active agent. All types are made of a soft platinum wire of less than a millimeter diameter and therefore are very soft. The softness of the platinum allows the coil to conform to the arbitrary shape of an aneurysm.

The deformation model of the coil is based on the recent work of Dequidt et. al. [6]. Coil dynamics is modeled using serially linked beam elements:

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{p} - \mathbb{F}\left(\mathbf{q}, \mathbf{v}, \mathbf{q}_0\right) + \mathbf{H}\mathbf{f}, \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{(n \times n)}$ gathers the mass and inertia matrices of beams. $\mathbf{q} \in \mathbb{R}^n$ is the vector of generalized coordinates (each node at the extremity of a beam has six degrees of freedom: three of which correspond to the spatial position, and three to the angular position in a global reference frame). The rest position $\mathbf{q}_0$ represents the reference position. Tuning the rest position allows to simulate different families of coil: for instance, helical shape or more complex 3D shape (see Figure 4). $\mathbf{v} \in \mathbb{R}^n$ is the vector of velocity. $\mathbb{F}$ represents internal visco-elastic forces of the beams, and $\mathbf{p}$ gathers external forces. $\mathbf{f}$ is the vector of the contact forces with the aneurysm wall, and $\mathbf{H}$ gathers the contact directions. To integrate this model we use backward Euler scheme with a unique linearization of $\mathbb{F}$ per time step. Moreover the linear solver takes advantage of the nature of our model. All beam elements being serially connected, $\mathbf{F}$ is a tridiagonal matrix with a band size of 12. nd we solve the linear system using the algorithm proposed by Kumar *et al.* [20]. The solution can thus be obtained in $O(n)$ operations instead of $O(n^3)$.
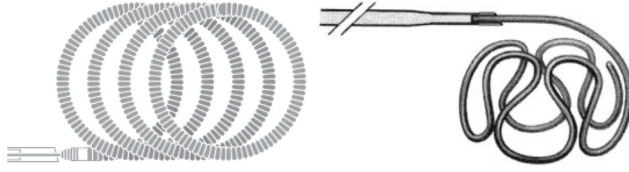
---

[1] http://www.cgal.org/

Figure 4. Example of coils used in our simulations: (left) Boston Scientific helical coil GDC 10; (right) 3D GDC built with omega loops [2].

3.2. Simulation of coil deployment

3.2.1. Modeling contacts with aneurysm walls

Simulating coil embolization requires to accurately model contacts that occur between the coil and the aneurysm wall. The contact model must provide the following features: first, account for the stick and slip transitions that take place during the coil deployment, second include a compliant behavior of vessel wall (we choose the one that is close to Boussinesq model [30]) and finally the friction motion of the coil along the aneurysm wall. For modeling contacts with friction, we use two different laws, based on the contact force and on the relative motion between the coil and the aneurysm walls. The contact law is defined along the normal $\mathbf{n}$ and the friction law, along the tangential $(\mathbf{t}, \mathbf{s})$ space of the contact.

The contact model, based on Signorini's law, indicates that there is complementarity between the gaps δ^n and the contact forces f^n along the normal direction, that is,

$$0 \leq \delta^n \perp \mathrm{f}^n \geq 0.$$

With the Coulomb's friction law, the contact force lies within a spacial conical region whose height and direction are given by the normal force, giving two complementarity conditions for stick and slip motion:

$$[\delta^t \ \delta^s] = \mathbf{0} \Rightarrow \|[\mathrm{f}^t \ \mathrm{f}^s]\| < \mu \, \|\mathrm{f}^n\| \qquad \text{(stick condition)}$$
$$[\delta^t \ \delta^s] \neq \mathbf{0} \Rightarrow [\mathrm{f}^t \ \mathrm{f}^s] = -\mu \, \|\mathrm{f}^n\| \frac{[\delta^t \ \delta^s]}{\|[\delta^t \ \delta^s]\|} \ \text{(slip condition)}$$

Where the vector $[\delta^t \ \delta^s]$ provides the relative motion in the tangential space and μ represents the friction coefficient.

The obtained complementarity relations could create *singular* events when it changes from one state to another: For instance, when a collision occurs at instant $t^\star$, the velocity $\mathbf{v}(t^\star)$ of the coil, at that point, changes instantaneously. The acceleration could then be ill-defined and we can observe some quick changes in the dynamics. Each friction contact creates three non-holonomic constraints along the normal and tangential directions. Our approach allows for processing simultaneously multiple friction contacts, including self-contacts on the coil.

3.2.2. Simulation steps

The processing of one simulation step begins with solving Equation 1 for all forces except contact forces ($\mathbf{f} = 0$). This *free motion* corresponds essentially to the deformation of the beam elements under gravity and user force input. Once the *free motion* has been computed, collision detection computes the contact points between the coil model and the aneurysm
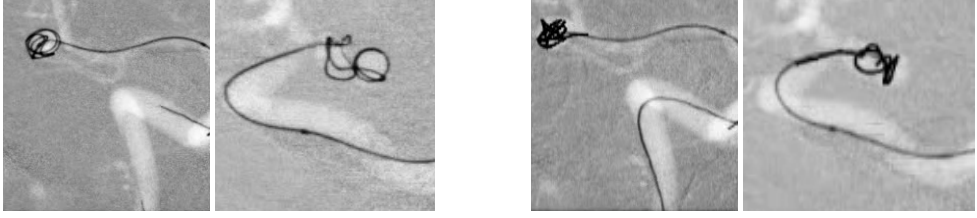
Figure 5. Examples of our simulation results: (left) real coil embolization; (right) our simulated coil embolization with 3D coils.

surface and the points of self-collision. When collisions are detected, the *contact response* is computed. This is a complex aspect that influences greatly the overall behavior of the coil model. To describe the mechanical behavior during contact, the mechanical coupling between the different contact points is modeled. This information is provided by evaluating the compliance matrix in the contact space, called $\mathbf{W}$, for both the coil and the aneurysm. Let's consider a contact α on the node i of the coil (with one constraint along the contact normal $\mathbf{n}$ and two along the tangential friction directions $\mathbf{t}, \mathbf{s}$). $\mathbf{H}_\alpha$ is the matrix of the frame $[\mathbf{n}\ \mathbf{t}\ \mathbf{s}]$. The mechanical coupling of this contact with a contact β (with frame $\mathbf{H}_\beta$) on node j can be evaluated with the following 3×3 matrix:

$$\mathbf{W}_{(\alpha,\beta)} = \mathbf{H}_\alpha^T \left(\frac{\mathbf{M}}{h^2} + \frac{d\mathbb{F}}{hd\mathbf{v}} + \frac{d\mathbb{F}}{d\mathbf{q}} \right)^{-1}_{(i,j)} \mathbf{H}_\beta = \mathbf{H}_\alpha^T \mathbf{C}_{(i,j)} \mathbf{H}_\beta,$$

where $\mathbf{C}_{(i,j)}$ is the 3×3 sub-matrix of global compliance matrix $\mathbf{C}$ (inverse of tangent matrix) at the rows of node i and the columns of node j. For the aneurysm wall, the formulation of the coupling is simpler:

$$\mathbf{W}_{(\alpha,\beta)} = \frac{g(d_{ij})}{e} \mathbf{H}_\alpha^T \mathbf{H}_\beta,$$

where e is an elasticity parameter that is homogeneous to young modulus and $g(d_{ij})$ is a Gaussian function of the distance, defined on the surface, between contact point i and j. The Gaussian function allows a fall-off of the coupling with increasing distance between the contact points. This model is close to the Boussinesq approximation which provides a distribution of the normal contact stress from the elasticity of the surface, around a point of contact [30].

The result of the *contact response* involves finding the friction contact forces that respect Signorini and Coulomb laws. Several works ([18] and [8]) present Gauss-Seidel iterative approaches that solve this problem. The solver needs an evaluation of a global compliance matrix $\mathbf{W}$, which is the sum of the compliances of the coil and the aneurysm wall. It also needs the value of the relative displacement of the contacting points during the free motion $\boldsymbol{\delta}^{\text{free}}$. When the contact forces are found, during the last step, called *contact correction* we compute the motion associated to the contact forces.

In this section, we presented an efficient dynamic coil model, *i.e.*, FEM beams based model for the intrinsic mechanical behavior of the coil, combined with a process for computing collision response (with the aneurysm wall) and auto-collision response. However, even if the results are quite encouraging, as illustrated in Figure 5, the mechanical modeling is not fully complete. Indeed, the behavior of the coil is also greatly influenced by the blood flow.

Moreover, we also aim at predicting the influence of the coils on the hemodynamic status near the aneurysm. As a result, it is important to simulate the flow within the aneurysm, and to propose a method for blood-coil coupling.

4. Modeling of blood flow around intracranial aneurysm

The blood is modeled as an incompressible Newtonian fluid of constant density, $\rho = 1.069 \times 10^3 kg/m^3$, and constant viscosity, $\mu = 3.5 \times 10^{-3} kg/(m \cdot s)$. In this work, we only consider the blood flow near intracranial aneurysms with relatively small Reynolds number ranging from 100 to 1,000 [31], which satisfies the laminar assumption. Thus, it is reasonable to describe the behavior of blood flow using the incompressible Navier-Stokes equations. Additionally, the influence of pulsating vessel on the fluid is ignored; hence vessel walls are assumed to be rigid.

We rely on the Discrete Exterior Calculus (DEC) method for numerically solving the equations, as it offers several benefits for our application. First of all, it is based on unstructured tetrahedral mesh, which is more suitable to describe irregular boundary of anatomical geometries than regular grids. From the tetrahedral mesh, referred as primal mesh, the dual mesh is constructed as follows: dual vertices correspond to the circumcenters of primal tetrahedra, dual edges link dual vertices located on neighbor tetrahedra, and dual faces are surfaces of Voronoi cells of primal vertices, which are dual cells as well. More generally, a dual $(n - p)$-cell is associated to a corresponding p-simplex ($p = 0, 1, 2, 3, n = 3$) as depicted in Figure 7.

Secondly, the orthogonality between primal and circumcentric dual is useful to define the physical variables in fluid (Figure 6), such as flux, which is the face integral of velocity orthogonal to the face, and the vorticity (the circulation per unit area at a point), whose direction is orthogonal to the plane of circulation (the line integral of velocity around a closed curve). The discrete version of these physical quantities is not only defined on points, but represented as scalars attached to the primitives of any dimension on primal or dual mesh, *i.e.*, vertex, edge, face or volume. The scalars are integral values of physical quantity over the primitives (except point-wise variable), and the orientation of each primitive represents the local direction of vector variable. For example, velocity is described as flux on each triangle face (2D primal primitive), so it is primal 2-form, and the orientation of the face indicates whether the fluid flows outward or inward. Vorticity is defined as the integral value over the dual face (2D dual primitive), thus it is dual 2-form. According to the Stoke's theorem, this value equals to the circulation along the boundary of dual face. The orientation of the primal edge gives the direction of vorticity.

Thirdly, based on the discretization of space and variables, discrete vector calculus operators, such as curl, divergence, Laplace operators, can be easily expressed by two basic operators, the discrete differentials d and the Hodge stars $\star$. The former, d_p, maps p-forms to $(p + 1)$-forms on the primal mesh, represented by signed incidence matrix. The transpose of this matrix operates similarly on the dual mesh. The latter, $\star_p$, maps from primal p-forms to dual $(n - p)$-forms, represented by a diagonal matrix whose element equals to the volume ratio between the corresponding dual and primal elements. And the inverse matrix maps in the opposite (Figure 7).
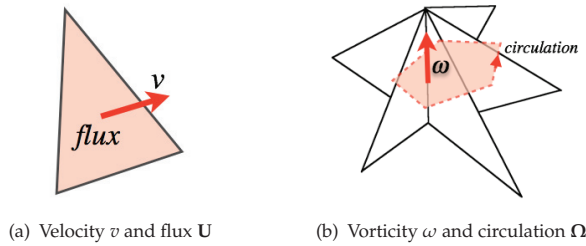
(a) Velocity v and flux $\mathbf{U}$ (b) Vorticity ω and circulation Ω

Figure 6. Discretization of physical variables in fluid: (a) velocity; (b) vorticity.
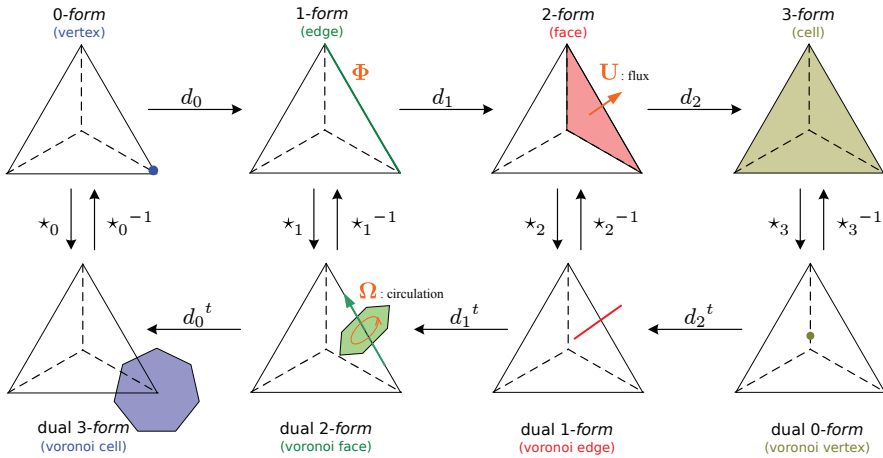

Figure 7. The duality of primal and dual mesh: the first line shows the primal simplex, whose dual elements are below. Physical variables $\mathbf{U}$ and Ω, defined as discrete forms, can be transferred by two fundamental operators d and $\star$.

Finally, it applies the same idea of Stable Fluids [39], a semi-Lagrangian method, which is much more stable than traditional Eulerian methods, and allows larger time steps and improves the computational efficiency. But instead of dealing with velocity field, the DEC method uses vorticity-based formulations (Equation 2), and preserves the circulation at a discrete level. As a result, to some degree, it overcomes the issue of numerical diffusion, and results in higher accuracy.

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = -\mathscr{L}_{\boldsymbol{v}}\boldsymbol{\omega} + \frac{\mu}{\rho}\Delta\boldsymbol{\omega}$$
$$\nabla \boldsymbol{v} = 0 \quad \boldsymbol{\omega} = \nabla \times \boldsymbol{v},$$

(2)

where $\boldsymbol{\omega}$ is the vorticity, Lie derivative $\mathscr{L}_{\boldsymbol{v}}\boldsymbol{\omega}$ (in this case equal to $\boldsymbol{v} \cdot \nabla\boldsymbol{\omega} - \boldsymbol{\omega} \cdot \nabla\boldsymbol{v}$), is the advection term, and $\frac{\mu}{\rho}\Delta\boldsymbol{\omega}$ is the diffusion term.

Simply speaking, the advection term describes the idea that the local spin is pushed forward along the direction of the velocity, which is consistent with Kelvin's circulation theorem: the circulation around a closed curve moving with the fluid remains constant with time. In this
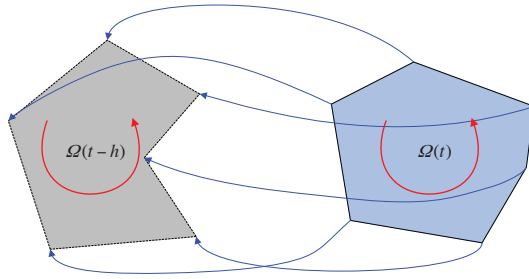
Figure 8. Backtracking. At the current time step $t_n = t$, given the velocity field, each dual vertex (on the right) is backtracked to its position at the previous time step $t_{n-1} = t - h$ (on the left). The circulation around the loop of dual face boundary at time t_n is forced to be equal to the circulation of the backtracked loop at time t_{n-1}, *i.e.*, $\Omega(t) = \Omega(t - h)$.

approach, the discrete vorticity is conserved by extending Kelvin's theorem to the discrete level: the circulation around the loop of each dual face's boundary keeps constant as the loop is advected by fluid flow. So we run a backtracking step (Figure 8) to find out where the current dual face comes from, and accumulate the circulation around the backtracked dual face, and then assign this value to the current one. This step makes the computation circulation-preserving at a discrete level, as well as stable, because the maximum of the new field is never larger than that of the previous field. For the diffusion term, linear solver is used, and an implicit scheme is chosen for the purpose of stability (Equation 3).

$$\omega_{n+1} - \omega_n = \frac{\mu h}{\rho} L \omega_{n+1}, \tag{3}$$

where h is the span of time step, and L is the Laplace operator.

5. Blood-coil interaction

In this section, we illustrate how we simulate the bilateral interaction between coils and blood flow. First, we describe how we compute the impact of the coil onto the blood flow by adding extra terms to the Navier-Stokes equation. Second, we explain how the drag force applied by blood flow onto the coil is computed. Finally, we combine the fluid and structure systems by a loosely-coupled strategy for real-time simulation of coil embolization.

5.1. Porous media model

The typical diameter of coils chosen for intracranial aneurysms ranges from $0.2mm$ to $0.4mm$, which is quite small compared to the aneurysm size (0.02 to 0.11 times of the intracranial aneurysm in diameter). And after inserted into aneurysm, they are randomly distributed, forming the shape of a twisted nest (Figure 9). Considering the relatively small dimension of coils and their random distribution in aneurysm, coils are modeled, from a statistical point of view, as porous media in aneurysm.

We divide the fluid domain $\mathscr{D}$ into 3 sub-domains, a coil-free and a coil-filled sub-domain, as well as a transitory sub-domain between them, which allows the porous parameters to vary
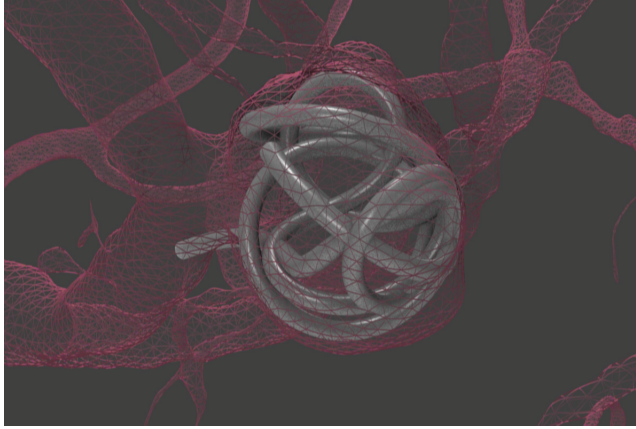
Figure 9. The shape of detachable coils after deployment.

smoothly between the first two sub-domains. However, blood motion in all sub-domains is described uniformly by the Navier-Stokes equations (vorticity-based) of Brinkmann type:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathscr{L}_{\boldsymbol{v}}\boldsymbol{\omega} = \frac{\mu}{\rho}\Delta\boldsymbol{\omega} - \frac{\mu\varphi}{\rho k}\boldsymbol{\omega} - \frac{\varphi^2 C_D}{\sqrt{k}}\nabla \times \boldsymbol{b},$$

$$\nabla(\varphi\boldsymbol{v}) = 0 \quad \boldsymbol{\omega} = \nabla \times \boldsymbol{v} \quad \boldsymbol{b} = \boldsymbol{v}\,|\boldsymbol{v}|,$$

$$(4)$$

where three more parameters are used to describe the properties of porous media: porosity φ, permeability k and drag factor C_D. Porosity φ describes the volume ratio of pores to the total coil-filled sub-domain, $\varphi = 1 - V_{coil}/V_{sac}$, where V_{coil} is the accumulated volume of all coils, and V_{sac} is the volume of the aneurysm sac. The permeability k measures the fluid conductivity through porous media, $k = \varphi^3/cS^2$, where c is the Kozeny coefficient related to the micro-shape of the porous media (for coils, the value of cylinders is chosen, $c = 2$), and S is the ratio of the surface area of all coils to the volume of porous region V_{sac}. The drag factor C_D can be derived from the local Reynolds number. Note that when $\varphi \to 1$ and $k \to \infty$, these porous terms disappear, therefore, Equation 4 is identical to Equation 2, within the coil-free region. The computation of the extended porous terms takes little extra computational time (less than 1% of computing the other terms) when using the DEC method.

5.2. Drag force

In the existing simulations of aneurysm embolization, the interactive force between blood and coil was only studied for the blood from a global view, while the local reacting force on coils during the deployment process was ignored. In fact, the last term of Equation 4 is a description of the interactive force, but treated as an averaged quantity. When computing the reaction on the coil, we apply its local version, which is the drag force of flow over cylinder, since the coil is considered to consist of serially linked cylinder segments:

$$\boldsymbol{F_D} = \frac{1}{2}C_D\rho\boldsymbol{v}_\perp\,|\boldsymbol{v}_\perp|\,A|l,$$

where $v_\perp$ is the velocity orthogonal to the coil, A is the cross-sectional area of the coil, l is the length of one short cylinder segment, and F_D is the drag force applied on this segment. The velocity parallel to the coil is neglected, since it only produces shear force on the coil, which is insignificant compared to the drag force, and has little impact on the movement of the coil in the blood. Hence, the reacting force on the coil only depends on local fluid velocity.

5.3. Fluid-structure coupling

For integrating the two models (coil and blood flow) in one single frame, we design a loosely-coupled strategy with the assumption that the simulation is performed over a series of identical cardiac cycles. Given the periodically time-varying boundary conditions at the inlet and outlet vessels around aneurysm, we solve the Navier-Stokes equations of Brinkmann type with a constant coil packing density by the DEC approach, and obtain the velocity at each tetrahedron center of the mesh at each time step. Then these velocity values are used to interpolate the velocity at the positions of coil segments and apply appropriate drag forces on the coil. The coil can provide real-time feedback inside the aneurysm at any time step during embolization. In this stage, we assume that a small segment of inserted coil does not change the blood flow. Until there is a significant increase of the coil density, the velocity of blood flow is recomputed at the new level of coil packing density. In this stage, we only care about the coil density, but not the coil shape. So the blood velocity field can be pre-computed and stored for real-time simulation of coil deployment. Although we use pre-computation of blood velocity, this process can be done in several minutes to simulate one cardiac cycle at five levels of coil density in our simulation. This is still essential for interactive planning.

The main benefit of the loosely-coupled approach is the relatively independency between the two systems, which allows different time resolution in two systems, independent real-time strategies, as well as pre-computation of the blood flow over one cardiac cycle. For the purpose of real-time refresh rate, we consider using relatively coarse mesh to reduce the size of the linear systems to be solved, and using large time steps to lessen the iterations necessary to simulate one second. As in other applications where real-time computation is sought, the objective is then to reach the best trade-off between accuracy and computational time.

6. Simulations and results

6.1. Blood flow simulation

We have performed comparative tests against FLUENT software [2] both in 2D and 3D space. In this case, we mostly aim at validating numerical accuracy of the DEC method rather than the ability to precisely describe the actual blood flow features near aneurysm, due to the present difficulties in *in vivo* analysis of velocity, particularly in the small cerebral vessels. Each group of the comparison between DEC and FLUENT consists of blood flow simulation on several identical meshes with the same geometry but different resolution.

The first group of simulations using a 2D aneurysm model (generated from the profile of a patient-specific geometry) is performed on three meshes composed of 210177, 19753 and 2160 triangles respectively. The contours of velocity magnitude and streamlines computed by DEC and FLUENT are shown in Figure 10. The unit of velocity displayed in all the figures is mm/s

---

[2] FLUENT is a commercial product of ANSYS (http://ansys.com), widely used in industries.
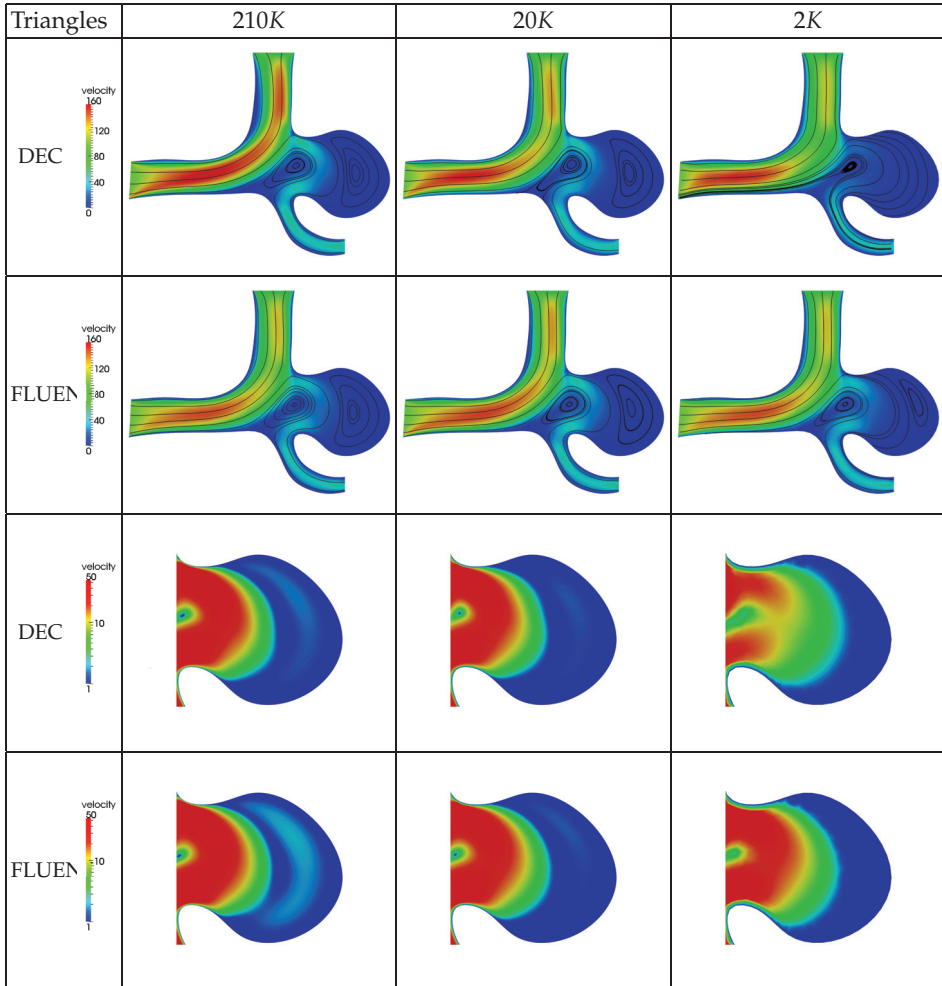
Figure 10. Comparison of velocity field on 2D aneurysm model. The contours of velocity magnitude and streamlines in the whole region (the first two rows) and in the sac (the last two rows) are computed by DEC and FLUENT on the identical meshes of three different resolution.

by default. Besides, we plot the profiles of the velocity magnitude over two lines across the inlet and the aneurysm respectively in Figure 11. When the mesh resolution decreases by 10 times, the contours and profiles are almost the same, and show the similarity between the two methods. In addition, the streamlines show a strong agreement for flow patterns and vortex structures in terms of the positions of the vortex centers. Only when we reduce the mesh resolution by 100 times, the result computed by DEC loses some detail information inside the sac; less fluid flows into the sac, and the vortex inside the sac is missing. But the main features of the velocity field still remain the same, such as the flow pattern (characterized by the streamlines) and the variation of the velocity field in space (characterized by the profiles).
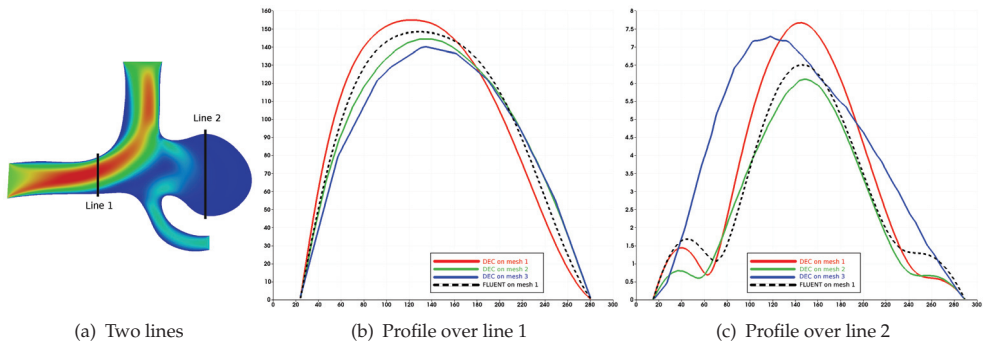
(a) Two lines   (b) Profile over line 1   (c) Profile over line 2

Figure 11. Comparison of velocity profiles over two lines across the inlet and aneurysm

The second group of simulations is performed similarly on two meshes of a 3D patient-specific aneurysm with a wide neck. In Figure 12, the streamlines show the similar movement of blood. In both cases there is a low-speed region surrounded by high-speed flows observed on slice 1, which represents a local vortex in this region. There are two obvious vortices in the FLUENT result, reflected both by the low-speed regions observed on slice 2 and the swirls of streamlines, but in the DEC result on the coarser mesh, the smaller vortex is not that obvious. These results show that the DEC method can capture large-scale flow structures, while small-scale differences exist between the two approaches. Considering both accuracy and computational efficiency, we choose the lower-resolution mesh of 34029 tetrahedra for the simulation of coil embolization, and the computational time of simulating $1s$ real time is $44s$ on an Intel i7 $3.33GHz$ processor.

6.2. Real-time simulation of coil embolization

We predict the embolization outcome for an intracranial aneurysm with a small sac of volume $132.1mm^3$ (usually the sac volume of an intracranial aneurysm varies from 100 to $1000mm^3$) and a wide neck of dimension $7.0mm$. From the clinical experience, the final coil packing density is around 25%, i.e., the porosity φ is around 75%. In Figure 13, we show the blood flow without coils, with 10% and 25% volume filled with coils. The velocity magnitude contours are compared on two sections, crossing the neck and the sac respectively. From the comparison on the neck section, we can see that every incremental increase in coil packing density is accompanied by a decrease in cross-neck flow rate. Additionally on the sac section, after inserting the first coil, the velocity magnitude over the whole sac region has been reduced, which creates a favorable hemodynamic environment for the deployment of the following coils.

Then we show the inverse influence of the blood flow on the coil deployment. Figure 14 displays the simulated process of placing the first coil into the small aneurysm. After the catheter is advanced in the vascular network to reach the position of the aneurysm, the coil is delivered through the catheter and inserted into the aneurysm. The contact among the catheter, the coil and the vessel wall, as well as the interaction of the blood flow are all included in this simulation. The pre-computation of blood velocity covering one cardiac cycle at 5 levels of coil density is accomplished in less than 5 minutes, and then using the pre-computed velocity field, the simulation of coil deployment is performed in real time.
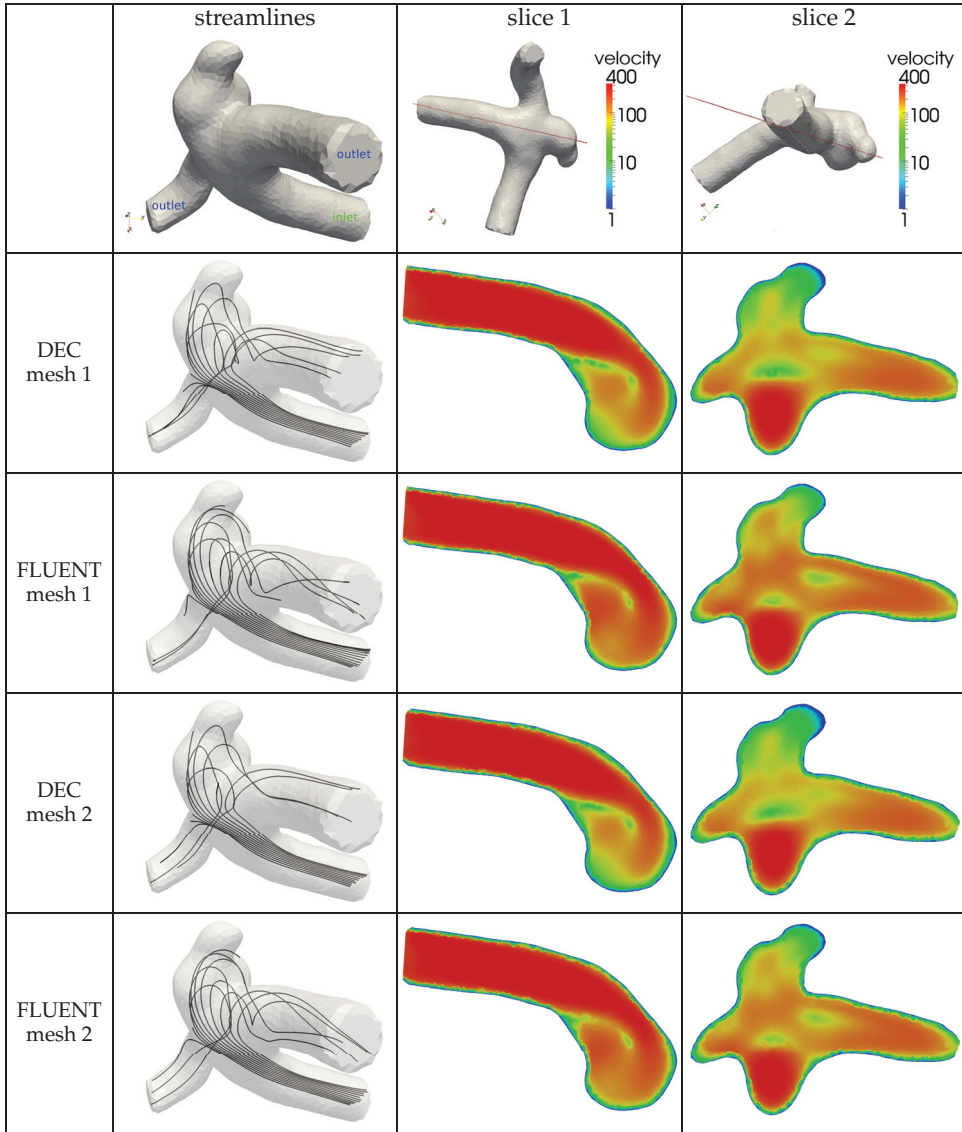
Figure 12. Comparison of velocity field on the 3D patient-specific aneurysm model. The first row displays the geometry of the model and the position of two slices chosen for comparing velocity magnitude contours. In the following rows, streamlines and contours of velocity magnitude are computed by DEC and FLUENT on the identical meshes, which are mesh 1 of 55711 tetrahedra and mesh 2 of 34029 tetrahedra.
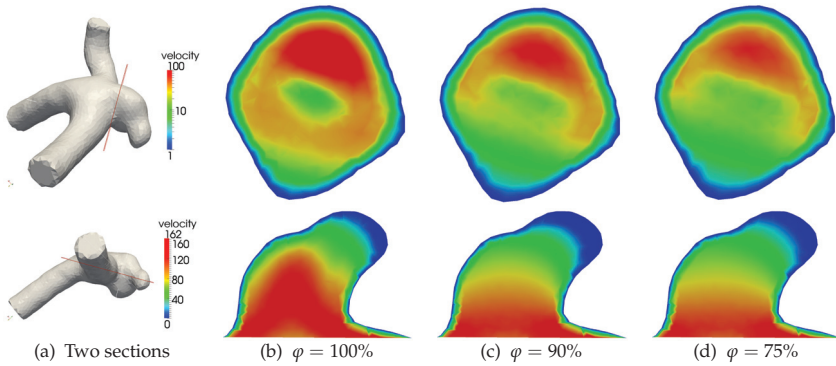
(a) Two sections	(b) $\varphi = 100\%$	(c) $\varphi = 90\%$	(d) $\varphi = 75\%$

Figure 13. Coil embolization of a small aneurysm. The velocity magnitude on two sections (a) before the embolization (b), after the first coil deployed (c), and after the final coil deployed (d).
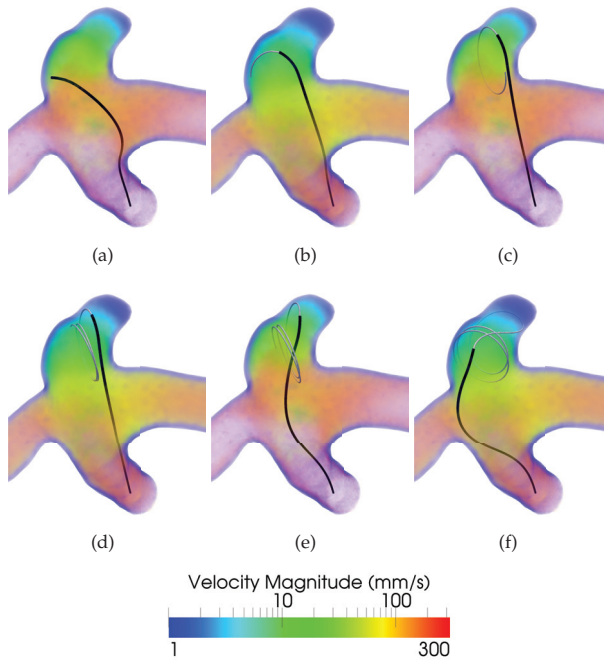

Figure 14. Simulation of coil embolization: (a) The catheter (black) reaches at the aneurysm neck through vessels. (b)-(f) The first coil (silver) is delivered by the catheter and inserted into the aneurysm. The colorful volume displays the periodically varying velocity field.

7. Conclusion and perspectives

7.1. Conclusion

We proposed an approach to accurate and real-time simulation of coil deployment, as well as prediction of coil embolization outcome in a patient-specific environment for clinical planning and rehearsal.

- The geometry of the aneurysm and its parent vessel has been reconstructed as an implicit surface. The proposed blobby model is particularly adapted to simulation; it enables to model complex shapes, as smooth surfaces, with a relatively small number of primitives. Our model improves upon previous works in two ways. First the constrained parameterization closely relates the implicit function to the distance field to the surface, and stabilizes the blob selection and subdivision process. Second, reconstruction is expressed as an energy minimization problem. The closed-form expression given for the energy enables to leverage efficient minimization algorithms with derivatives for faster and more accurate results.

- We have achieved real-time simulation of flexible medical devices, such as the catheter or coil by using a model based on beams that can handle various rest shapes and large geometric deformations. Special care is taken to make this model computationally robust and efficient by using dedicated linear solver and time-integration scheme. Our contributions also include an accurate contact model to handle the collisions between the coil and the vascular surface. This allows computation times of about 2 ms for a coil composed of 100 beam elements.

- While most previous work on hemodynamic simulation aimed at accurate results and required dozens of hours computational time, we have achieved fast simulation of blood flow using the DEC method, which is initially developed in the field of computer graphics. However, a much deeper analysis and improvement has been made for medical application.

- Bilateral interaction between coil and blood flow has been first studied in our work for planning two key steps of the procedure. Firstly, prediction of hemodynamic status after deployment is performed by modeling the inserted coils as porous media. Secondly, the reciprocal force, *i.e.*, the impact of the flow onto the coil, is modeled as drag force and applied on the coil during deployment for interactive choice and placement of the first coil. The results show that our approach permit real-time simulation of the interaction between coils and blood flow during coil embolization.

7.2. Perspectives

Regarding future work, first of all, we are further improving the computational efficiency, since real-time simulation has not been achieved when performed on a mesh consisting of over 8,000 elements. We still want to more deeply investigate various computational strategies to obtain real-time computation by using more advanced numerical schemes, such as parallel implementation on GPU of the backtracking step, optimization of linear solvers by the preconditioning technique coupled with a GPU-based conjugate gradient implementation [3]. Additionally, more advanced techniques to generate a multi-resolution mesh, such as adaptive refinement [32], could also be a solution, as it maximizes the result accuracy while minimizing the computational effort.

Of course, we acknowledge that further validation is required, both on the DEC method and on other elements of the simulation. However, comparison between simulated results and *in vivo* data remains difficult due to the limitations of current medical imaging techniques. We are starting to investigate the use of fluoroscopic imaging to assess the simulation of medical devices used in interventional radiology procedures. Regarding the specific validation of the flow computation, a possible direction is MR imaging which, under some modalities, can measure flow patterns. Although quite noisy, such data could provide interesting insights and help validate our results.

Before put to wide clinical utilization, our method should be further improved. The boundary conditions obtained from patient-specific real data will improve the realism of the simulated results. To obtain such real data, we can benefit from the imaging techniques or the state-of-the-art medical instruments, such as the ultra-miniature pressure catheter, which is a so small sensor on the catheter that it does not significantly change the blood flow. Furthermore, standards of assessment and metrics of evaluation should be set up to offer doctor the information on the minimal number of coils required to achieve embolization in long term, and to evaluate the doctor's performance on the procedure.

Finally, when using stent-assisted coiling, the stent acts as scaffolding, and prevents the coils from falling out. Our work could be easily extended to simulate the outcome by taking the collision contact between coils and the stent into account.

Acknowledgments

Author details

Yiyi Wei, Stéphane Cotin, Jérémie Dequidt, Christian Duriez, Jérémie Allard
Shacra Team, INRIA, France

Erwan Kerrien
Magrit Team, INRIA, France

8. References

[1] Alderliesten, T. [2004]. *Simulation of Minimally-Invasive Vascular Interventions for Training Purposes*, PhD dissertation, Utrecht University.

[2] Cloft, H. J., Joseph, G. J., Tong, F. C., Goldstein, J. H. & Dion, J. E. [1999]. Use of three-dimensional guglielmi detachable coils in the treatment of wide-necked cerebral aneurysms, *American Journal of Neuroradiology* .

[3] Courtecuisse, H., Allard, J., Duriez, C. & Cotin, S. [2011]. Preconditioner-based contact response and application to cataract surgery, *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011* pp. 315–322.

[4] Dequidt, J., Duriez, C., Cotin, S. & Kerrien, E. [2009]. Towards interactive planning of coil embolization in brain aneurysms, *Proceedings of the 12th International Conference on Medical Image Computing and Computer-Assisted Intervention: Part I*, MICCAI '09, Springer-Verlag, Berlin, Heidelberg, pp. 377–385.

[5] Dequidt, J., Lenoir, J. & Cotin, S. [2007]. Interactive contacts resolution using smooth surface representation, *Proc. Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, pp. 850–857.

[6] Dequidt, J., Marchal, M., Duriez, C., Kerien, E. & Cotin, S. [2008]. Interactive simulation of embolization coils: Modeling and experimental validation, *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008* pp. 695–702.

[7] Duriez, C., Cotin, S., Lenoir, J. & Neumann, P. F. [2006]. New approaches to catheter navigation for interventional radiology simulation, *Computer Aided Surgery* 11: 300–308.

[8] Duriez, C., Dubois, F., Kheddar, A. & Andriot, C. [2006]. Realistic haptic rendering of interacting deformable objects in virtual environments, *IEEE Transactions on Visualization and Computer Graphics* 12(1): 36–47.

[9] Elcott, S., Tong, Y., Kanso, E., Schröder, P. & Desbrun, M. [2007]. Stable, circulation-preserving, simplicial fluids, *ACM Transactions on Graphics (TOG)* 26(1): 4.

[10] Frangi, A., Niessen, W., Hoogeveen, R., van Walsum, T. & Viergever, M. [1999]. Model-based quantitation of 3-D magnetic resonance angiographic images, *IEEE Trans. Med. Imaging* 18(10): 946–956.

[11] Frangi, A., Niessen, W., Vincken, K. & Viergever, M. [1998]. Multiscale vessel enhancement filtering, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 1998*, number 1496 in *LNCS*, pp. 130–137.

[12] Friman, O., Hindennach, M., Kühnel, C. & Peitgen, H.-O. [2010]. Multiple hypothesis template tracking of small 3D vessel structures, *Medical Image Analysis* 14(2): 160 – 171.

[13] Goldman, R. [2005]. Curvature formulas for implicit curves and surfaces, *Computer Aided Geometric Design* 22: 632–658.

[14] Groden, C., Laudan, J., Gatchell, S. & Zeumer, H. [2001]. Three-dimensional pulsatile flow simulation before and after endovascular coil embolization of a terminal cerebral aneurysm, *Journal of Cerebral Blood Flow & Metabolism* 21(12): 1464–1471.

[15] Hoefer, U., Langen, T., Nziki, J., Zeitler, F., Hesser, J., Mueller, U., Voelker, W. & Maenner, R. [2002]. Cathi - catheter instruction system, *Computer Assisted Radiology and Surgery (CARS)*, Paris, France, pp. 101 – 06.

[16] Hop, J., Rinkel, G., Algra, A. & van Gijn, J. [1997]. Case-fatality rates and functional outcome after subarachnoid hemorrhage: a systematic review, *Stroke* 28(3): 660–664.

[17] Huang, J. & van Gelder, J. [2002]. The probability of sudden death from rupture of intracranial aneurysms: a meta-analysis, *Neurosurgery* 51(5): 1101.

[18] Jourdan, F., Alart, P. & Jean, M. [1998]. A gauss-seidel like algorithm to solve frictional contact problems, *Comp. Meth. in Appl. Mech. and Engin.* pp. 33–47.

[19] Kakalis, N., Mitsos, A., Byrne, J. & Ventikos, Y. [2008]. The haemodynamics of endovascular aneurysm treatment: a computational modelling approach for estimating the influence of multiple coil deployment, *Medical Imaging, IEEE Transactions on* 27(6): 814–824.

[20] Kumar, S. & Petho, A. [1993]. An algorithm for the numerical inversion of a tridiagonal matrix, *Communications in Numerical Methods in Engineering* 9(4): 353–359.

[21] Lesage, D., Angelini, E., Bloch, I. & Funka-Lea, G. [2009].   A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes, *Medical Image Analysis* 13(6): 819–845.

[22] Masutani, Y., Masamune, K. & Dohi, T. [1996]. Region-growing based feature extraction algorithm for tree-like objects, *Visualization in Biomedical Computing*, Vol. 1131 of *LNCS*, Springer, pp. 159–171.

[23] Mazumdar, J. [1992]. *Biofluid mechanics*, World Scientific Pub Co Inc.

[24] McGregor, R., Szczerba, D., Muralidhar, K. & Székely, G. [2009].   A fast alternative to computational fluid dynamics for high quality imaging of blood flow, *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009* pp. 124–131.

[25] McLeod, K., Caiazzo, A., Fernández, M., Mansi, T., Vignon-Clementel, I., Sermesant, M., Pennec, X., Boudjemline, Y. & Gerbeau, J. [2010]. Atlas-based reduced models of blood flows for fast patient-specific simulations, *Statistical Atlases and Computational Models of the Heart* pp. 95–104.

[26] Montagnat, J. [1999]. *Deformable Modelling for 3D and 4D Medical Image Segmentation*, PhD thesis, Nice – Sophia Antipolis University, Nice, France.

[27] Morales, H., Larrabide, I., Kim, M., Villa-Uriol, M., Macho, J., Blasco, J., Roman, L. & Frangi, A. [2011].   Virtual coiling of intracranial aneurysms based on dynamic path planning, *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011* pp. 355–362.

[28] Muraki, S. [1991].   Volumetric shape description of range data using Blobby Model, *SIGGRAPH Comput. Graph.* 25: 227–235.

[29] Nowinski, W. & Chui, C. [2001]. Simulation of interventional neuroradiology procedures, *MIAR*, pp. 87 – 94.

[30] Pauly, M., Pai, D. & Leonidas, G. [2004].   Quasi-rigid objects in contact, *Proceedings of ACM SIGGRAPH Symposium on Computer Animation*, pp. 109–119.

[31] Penney, D. [2002]. Hemodynamics.

[32] Prakash, S. & Ethier, C. [2001].   Requirements for mesh resolution in 3d computational hemodynamics, *Journal of biomechanical engineering* 123: 134.

[33] Preim, B. & Oeltze, S. [2008]. 3d visualization of vasculature: An overview, *Visualization in Medicine and Life Sciences*, Mathematics and Visualization, Springer, pp. 39–59.

[34] Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S. & Kikinis, R. [1998]. 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images, *Medical Image Analysis* 2: 143–168.

[35] Schumann, C., Neugebauer, M., Bade, R., Preim, B. & Peitgen, H.-O. [2008].   Implicit vessel surface reconstruction for visualization and CFD simulation, *Int. J. of Computer Assisted Radiology and Surgery* 2(5): 275–286.

[36] Sherstyuk, A. [1999]. Kernel functions in convolution surfaces: A comparative analysis, *The Visual Computer* 15(4): 171–182.

[37] Slabaugh, G., Dinh, Q. & G., U. [2007].   A variational approach to the evolution of radial basis functions for image segmentation, *Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

[38] Spillmann, J. & Harders, M. [2009]. Inextensible elastic rods with torsional friction based on lagrange multipliers, *Computer Animation and Virtual Worlds* 21(6): 561–572.

[39] Stam, J. [1999]. Stable fluids, *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., pp. 121–128.

[40] Taubin, G. [1991]. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Trans. on PAMI* 13: 1115–1138.

[41] Tournois, J., Wormser, C., Alliez, P. & Desbrun, M. [2009]. Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation, *ACM Transactions on Graphics (TOG)* 28(3): 1–9.

[42] Tsingos, N., Bittar, E. & Cani, M.-P. [1995]. Implicit surfaces for semi-automatic medical organ reconstruction, *Computer Graphics Internat. (CGI'95)*, pp. 3–15.

[43] Tyrrell, J., di Tomaso, E., Fuja, D., Tong, R., Kozak, K., Jain, R. & Roysam, B. [2007]. Robust 3-D modeling of vasculature imagery using superellipsoids, *IEEE Trans. Med. Imag.* 26(2): 223–237.

[44] Wink, O., Niessen, W. & Viergever, M. [2000]. Fast delineation and visualization of vessels in 3-D angiographic images, *IEEE Trans. Med. Imag.* 19(4): 337–346.