Toward Real-time Simulation of Blood-Coil Interaction during Aneurysm Embolization

Yiyi Wei, Stéphane Cotin, Le Fang, Jérémie Allard, Chunhong Pan

Toward Real-time Simulation of Blood-Coil Interaction during Aneurysm Embolization

Yiyi Wei[1,2], Stephane Cotin[1], Le Fang[3], Jeremie Allard[1], and Chunhong Pan[2]

[1]INRIA Lille - North Europe, 59650 Villeneuve d'Ascq, France
[2]LIAMA CASIA, 100190 Beijing, China
[3]LMFA, Ecole centrale de Lyon, 69130, Ecully, France

Abstract. Over the last decade, remarkable progress has been made in the field of endovascular treatment of aneurysms. Technological advances continue to enable a growing number of patients with cerebral aneurysms to be treated with a variety of endovascular strategies, essentially using detachable platinum coils. Yet, coil embolization remains a very complex medical procedure for which careful planning must be combined with advanced technical skills in order to be successful.

In this paper we propose a method for computing the complex blood flow patterns that take place within the aneurysm, and for simulating the interaction of coils with this flow. This interaction is twofold, first involving the impact of the flow on the coil during the initial stages of its deployment, and second concerning the decrease of blood velocity within the aneurysm, as a consequence of coil packing. We also propose an approach to achieve real-time computation of coil-flow bilateral influence, necessary for interactive simulation. This in turns allows to dynamically plan coil embolization for two key steps of the procedure: choice and placement of the first coils, and assessment of the number of coils necessary to reduce aneurysmal blood velocity and wall pressure.

1 Introduction

Detachable coil embolization is a recent interventional technique for treating aneurysms and other blood vessel malformations in the brain and other parts of the body. The procedure uses the vascular network to reach the diseased vessel, starting with the insertion of a catheter (a long, thin and flexible tube) into the femoral artery. This catheter is then advanced through the arterial system until the aneurysm location is reached. Once in position, the physician places several coils through a micro-catheter into the aneurysm. The presence of coils reduces blood flow and wall pressure within the aneurysm, thus creating a favorable hemodynamic environment for thrombus embolization. The formation of a blood clot around the coil blocks off the aneurysm, thus considerably reducing the risk of rupture. Although coil embolization is less invasive than open surgery, such procedures are very difficult to perform and require careful planning and a long experience to minimize the risks for the patient. Yet, even in the case of a successfully performed procedure, the choice of the coil (shape, length, diameter) plays a key role in the long term success of the procedure. In this context, the development of an interactive planning system, allowing interventional radiologists to select different coils and test their behavior in a patient-specific environment, could make a difference. This requires to not only model the behavior of a coil in a patient-specific model of the aneurysm, but also to compute the interaction between a coil and the complex flow occurring within the aneurysm.

Previous work Blood flow dynamics is starting to play an increasingly important role in the assessment of vascular pathologies, as well as in the evaluation of pre- and post-operative status. While angiography has been an integral part of interventional radiology procedures for years, it is only recently that detailed analysis of blood flow patterns has been studied as a mean to assess complex procedures, such as coil deployment. A few studies have focused on aneurysm-related haemodynamics before and after endovascular coil embolization. Groden *et al.* [1] constructed a simple geometrical model to approximate an actual aneurysm, and evaluated the impact of different levels of coil packing on the flow and wall pressure by solving Navier-Stokes equations. Kakalis *et al.* [2] employed patient-specific data to get more realistic flow patterns, and modeled the coiled part, from a static point of view, as a porous medium. As these studies aimed at accurate Computational Fluid Dynamics simulation, they rely on commercial software, and the computation times (dozens of hours in general) are incompatible with interactive simulation or even clinical practice. Generally speaking, accuracy and efficiency are two significant pursuits in numerical calculation, but unfortunately always contradictory.

In the field of Computer Graphics, the main objective is to obtain efficient computations, yet still be able to capture the main structures of fluid motion. Stable fluids approach [3] was a significant milestone, as it brought in fluid advection and the Helmholtz-Hodge decomposition to ensure the mass conservation law. However, this approach relies on a discretization of the Eulerian space by a regular grid, thus making it inappropriate for simulations requiring irregular boundaries, as it is the case in medical applications. Recently, the Discrete Exterior Calculus (DEC) [4] method settled this problem, providing a mean to handle arbitrary meshes. This approach can be considered as an extension of Finite Difference Method to an arbitrary discretization of space. By preserving circulation at a discrete level it also guarantees a certain level of accuracy.

Regarding fluid/solids interaction, various models have been put forth. For instance, Carlson *et al.* [5] solved this problem by considering rigid objects as if they were made of fluid, and the rigidity was maintained by constraining the velocity field in the object region. Robinson-Mosher *et al.* [6] proposed a two-way solid/fluid coupling method to compute mutual effects separately at each time step. The effect of the solid on the fluid is modeled by taking solid velocity as a boundary condition, while the effect of the fluid on the solid is determined by integrating the fluid pressure along the solid boundary. Considering the dimension of coils and their nest-shaped distribution in an aneurysm, a different computational strategy is required in our case.

Finally, previous work in the area of real-time simulation of interventional radiology procedures mainly focused on training rather than planning (besides the work of [7]) and has been limited to the simulation of flexible devices without considering fluid interaction (see [8], or [9] for instance).

In this paper, we present a novel technique for accurately computing (eventually in real-time) the flow of blood within an aneurysm, as well as the interaction between blood and coils. We rely on the Discrete Exterior Calculus method to obtain an ideal trade-off between accuracy and computational efficiency. We also propose to pre-compute the flow velocity field over a cardiac cycle to achieve real-time computation of the interaction between coils and blood flow. Our results show that our approach can describe the influence between coils and blood flow.

2 Modeling interactions between blood flow and coils

In this section, we first describe the computational approach for computing blood flow in and around an aneurysm (section 2.1). To model the impact of the coil onto the flow (as a change of flow pattern and a decrease of velocity) we then introduce the notion of porous media, where coils are described from a statistical point of view, translating the idea that, after deployment, coils are randomly distributed in the aneurysm. We also show how the reverse effect, *i.e.*, the drag force applied onto the coil due to blood velocity, can be computed (section 2.2). This is particularly important during the first stage of coil deployment.

2.1 Blood motion

Porous media model We divide the fluid domain $\mathscr{D}$ (2D or 3D) into 2 subdomains, a coil-free and a coil-filled subdomain. Blood motion in both subdomains is described by a Navier-Stokes equation of Brinkmann type:

$$\begin{gathered}\frac{\partial(\varphi\rho\boldsymbol{u})}{\partial t} + \rho(\boldsymbol{u}\cdot\nabla)(\varphi\boldsymbol{u}) = -\varphi\nabla p + \mu\triangle(\varphi\boldsymbol{u}) - \frac{\varphi^2\mu}{k}\boldsymbol{u} - \frac{\varphi^3 C_D\rho}{\sqrt{k}}\boldsymbol{u}\,|\boldsymbol{u}| \\ \nabla(\varphi\boldsymbol{u}) = 0\end{gathered} \tag{1}$$

where $\boldsymbol{u}$ is the velocity of the fluid with density ρ and viscosity μ, and p is the pressure. The porosity φ and the permeability k are constitutive characteristics of the porous media, and C_D is the drag factor. The porosity φ describes the volume ratio of fluid to coil-filled subdomain, $\varphi = 1 - \frac{V_{coil}}{V_{an}}$, where V_{coil} is the accumulated volume of all coils, and V_{an} is the volume of the aneurysm. The permeability k measures the fluid conductivity through porous media, $k = \frac{\varphi^3}{cS^2}$, where c is the Kozeny coefficient (for cylinders, $c = 2$), and S is the ratio of the surface area of all coils to the volume of the aneurysm. The drag factor C_D can be derived from the computation of a local Reynolds number. In order to get rid of the pressure term, we take the curl of (1). As density and porosity remain constant in each subdomain, we have

$$\begin{gathered}\frac{\partial\boldsymbol{\omega}}{\partial t} + \mathscr{L}_{\boldsymbol{u}}\boldsymbol{\omega} = \frac{\mu}{\rho}\triangle\boldsymbol{\omega} - \frac{\mu\varphi}{\rho k}\boldsymbol{\omega} - \frac{\varphi^2 C_D}{\sqrt{k}}\nabla\times\boldsymbol{b} \\ \nabla(\varphi\boldsymbol{u}) = 0 \quad \boldsymbol{\omega} = \nabla\times\boldsymbol{u} \quad \boldsymbol{b} = \boldsymbol{u}\,|\boldsymbol{u}|\end{gathered} \tag{2}$$

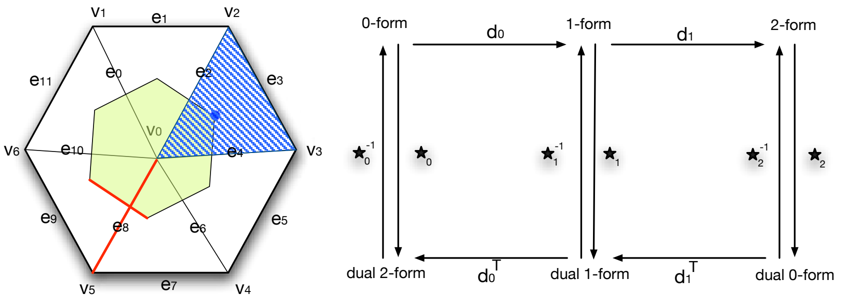

Fig. 1: (Left): two-dimensional mesh and its dual mesh, (right): transition graph between variables at primal and dual primitives, and operators describing the transitions.

where $\boldsymbol{\omega}$ is the vorticity, $\mathscr{L}_{\boldsymbol{u}}\boldsymbol{\omega}$ is the Lie derivative, equal in our case to $\boldsymbol{u} \cdot \nabla\boldsymbol{\omega} - \boldsymbol{\omega} \cdot \nabla\boldsymbol{u}$, and $\boldsymbol{b}$ is the porous term.

Numerical solution As mentioned previously, the complex shape of aneurysms requires an unstructured grid to describe the geometry. We rely on the DEC method to provide a numerical framework for solving the fluid equations, by discretizing the space as a simplicial complex, and computing its dual complex. State variables are defined as discrete forms, *i.e.*, integral values over elements of these two meshes, complying with conservation laws at a discrete level, which is a key point to get accurate and stable results. Since the mesh is static (Eulerian approach), computation efficiency is obtained by pre-computing several vector calculus operators such as gradient and Laplace, which are defined using basic topological and geometrical operations (see [4] or [10] for details).

Discretization In the following part, we limit our descriptions and illustrations to the 2D case for simplicity, but definitions in 3D are similar. The domain $\mathscr{D}$ is discretized as an oriented simplicial complex, *i.e.*, a triangulation, referred to as primal mesh. We denote the vertex set $V = \{v_i\}$, the edge set $E = \{e_{ij}\}$, and the triangle set $T = \{t_{ijk}\}(0 \leqslant i, j, k \leqslant |V|)$.

The dual mesh of a triangulation is constructed as follows: dual vertices correspond to the circumcenters of primal triangles, dual edges link dual vertices located on neighbor triangles, and dual faces are defined as Voronoi cells. More generally, a dual $(n-p)$-cell is associated to a corresponding p-simplex ($p = 0, 1, 2, n = 2$ for 2D mesh) as depicted in Fig. 1. Problem variables are defined as discrete p-forms, *i.e.*, scalars associated to p dimension primitives (p-cells) of either the primal or dual mesh.

Operators All the vector calculus operators involved in our computation can be derived from two types of fundamental operators: the discrete differentials d and the hodge stars $\star$. The former (d) maps between discrete forms on the same mesh, represented by the transpose of the signed incidence matrix, while the latter ($\star$) transforms discrete forms from one mesh to the other, represented by a diagonal matrix whose element equals to the volume ratio between the corresponding dual and primal elements (see Fig. 1).

```
$C \leftarrow d_0^t \star_1$                                   // Load mesh and compute the operators
$L \leftarrow d_0^t \star_1 d_0$
$L' \leftarrow$ LagrangeMultiplier($L$)
loop                                                            // Time stepping $\triangle t$
    $\hat{\mathbf{c}} \leftarrow$ backtrackTriangleCenters($\triangle t$)           // advect vorticities
    $\hat{\mathbf{v}} \leftarrow$ interpolateVelocity($\hat{\mathbf{c}}$)
    for each dual face $f$
        $\mathbf{\Omega}_f \leftarrow 0$
        for each dual edge $(i, j)$ on the boundary of $f$
            $\mathbf{\Omega}_f \leftarrow \mathbf{\Omega}_f + \frac{1}{2}(\hat{\mathbf{v}}_i + \hat{\mathbf{v}}_j)(\hat{\mathbf{c}}_i - \hat{\mathbf{c}}_j)$
    for each dual face $f$ in the coil-filled subdomain         // add porous terms
        $\mathbf{\Omega}_f \leftarrow \frac{\rho k}{\rho k + \mu \varphi \triangle t}\mathbf{\Omega}_f + \frac{\varphi^2 C_D}{\sqrt{k}} \triangle t \; (C \times \mathbf{B})_f$
    $\mathbf{\Psi} \leftarrow$ linearSolver( $(\star_0 - \mu L \triangle t)\mathbf{\Psi} = \mathbf{\Omega}$) )   // add viscous term
    $\mathbf{\Omega} \leftarrow \star_0 \mathbf{\Psi}$
    $\tau \leftarrow$ setBoundaryConditions($t$)              // recover flux from vorticity
    $(\mathbf{\Phi} \mid \lambda) \leftarrow$ linearSolver( $L'(\mathbf{\Phi} \mid \lambda) = (\mathbf{\Omega} \mid \tau)$ )
    $\mathbf{U} = d_0 \mathbf{\Phi}$
    foreach triangle $f$ do $\mathbf{v}_f \leftarrow A_f^{-1} \mathbf{U}_f$     // velocity at triangle centers
    $\mathbf{B} \leftarrow$ integrate($\mathbf{v}$)
```

Fig. 2: Pseudocode of Fluid Computation

Solving Fluid Equations In DEC, velocity is described as flux, *i.e.*, the mass of fluid passing through each edge per unit time. Thus it's a discrete 1-form $\mathbf{U}$, represented as a vector of size $|E|$. Similarly, the porous term $\boldsymbol{b}$ is defined as a discrete 1-form $\mathbf{B}$. In continuum fluid dynamics, vorticity measures the local angular rate of rotation, defined as the circulation per unit area at a point. Accordingly, we describe discrete vorticity $\mathbf{\Omega}$ through the integral over faces, and a natural choice is the dual face, in that we can easily get these values from $\mathbf{U}$ as follows: $\star_1$ transforms $\mathbf{U}$ on primal edges to $\star_1\mathbf{U}$ on dual edges, and then d_0^t sums it on each dual faces by accumulating $\star_1\mathbf{U}$ on all incident dual edges, *i.e.*, $\mathbf{\Omega} = d_0^t \star_1 \mathbf{U}$, a dual 2-form. This explains how $\nabla\times$ is built by the basic operators. Following similar principles, all operators (grad $\nabla\cdot$, curl $\nabla\times$, Laplace $\triangle$) used in (2) can be constructed from d and $\star$: $\nabla\cdot = d_1$, $\nabla\times = d_0^t\star_1$, $\triangle = d_0^t \star_1 d_0$.

The vorticity-based equation (2), simply speaking, describes the idea that the local spin is pushed forward along the direction of the velocity. This is consistent with Kelvin's circulation theorem: the circulation around a closed curve moving with the fluid remains constant with time [10]. In our 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. We implement this by *backtracking* all the loops into the previous time step [3]. Once the vorticity have been computed, the flux is recovered by solving the Laplace equation, taking into account the varying boundary conditions. Finally, the velocity at each triangle centers is computed using the property that there is a unique vector at each dual vertex whose projection along the incident dual edges is consistent with the flux of the corresponding primal edges. This defines a projection matrix A_f for each triangle f. The final computation process is shown in Fig. 2.

2.2 Coil motion

In the existing simulations of aneurysm embolization, the interactive force between blood and coil was only studied for the blood, while the reacting force on coils was ignored. In fact, the last term of (1) 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 a cylinder:

$$F_D = \frac{1}{2} C_D \rho \boldsymbol{u}_\perp \left| \boldsymbol{u}_\perp \right| A dh, \tag{3}$$

where $\boldsymbol{u}_\perp$ is the velocity orthogonal to the coil, A is the cross-sectional area of the coil, dh is the length of the coil section. 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. Hence, the reacting force on the coil only depends on local fluid velocity. The coil model is based on the work of Dequidt *et. al.* [7] where coils are modeled as a series of serially-linked beam elements. The deformation of the structure is computed using a finite element approach which can be optimized for real-time computation by taking advantage of the structure (tri-diagonal band) of its stiffness matrix.

3 Real-time simulation

In this section, we show how the two previously described models can be combined and used in a real-time simulation. For this we assume the simulation is performed over a series of identical cardiac cycles. Periodically time-varying boundary conditions are set at the inlet and outlet vessels around the aneurysm for a duration covering a complete cardiac cycle. Using the method described in section 2.1 we compute and store the velocity field for multiple time steps within one cardiac period. This process can be done for different densities of coil in the aneurysm. This database of velocity fields can then be used to interpolate the velocity at the position of coil segments and apply appropriate drag forces. It can also provide real-time feedback, at any step of the embolization, about blood velocities inside the aneurysm. It should also be noted that most of the computation time of our method comes from solving two linear sparse systems of equations. Many numerical techniques can be used to improve the efficiency of this process. When dealing with relatively small systems a direct inversion of the matrices is often the best approach. The inversion of these matrices can be performed only once, during initialization. This ensures real-time or near real-time computation (which is typically the case when dealing with two-dimensional problems). For larger systems, we rely on the offline computation of operators and a pre-factorization of the sparse linear systems. This is however not always sufficient to achieve real-time computation when high accuracy is needed.

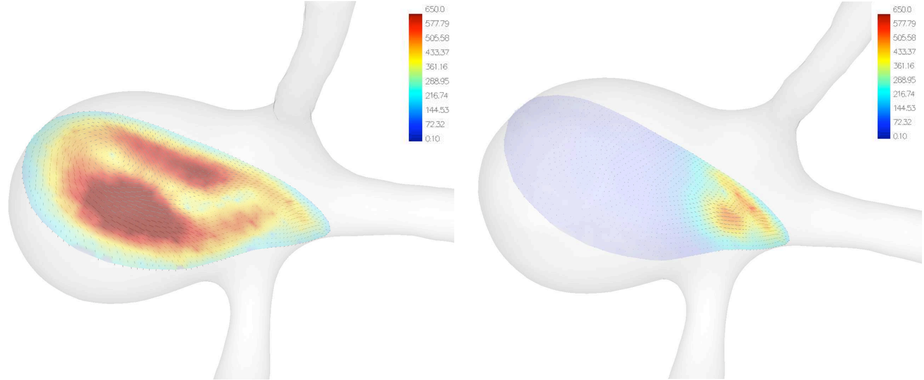

Fig. 3: (Left): velocity magnitude in aneurysm with no coil; (Right): with 16.8% of the volume filled with coils. Aneurysm model composed of of $18K$ tetrahedral elements.

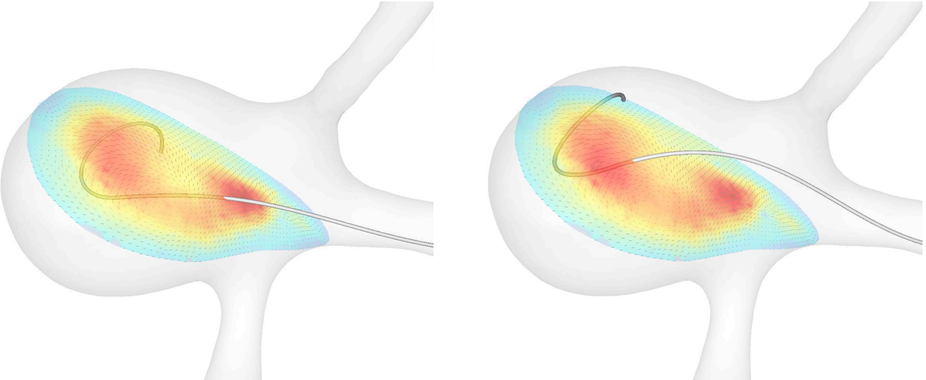

Fig. 4: (Left): Coil shape when the effect of blood flow is not taken into consideration; (Right): Resulting shape when considering flow forces on the coil.

4 Results

Our simulations are performed on an aneurysm of particular interest, with a large sac of volume $8.976 \cdot 10^{-7} m^3$ and a wide neck of dimension $8.2 \cdot 10^{-3} m$. Such aneurysms are difficult to treat, as coils might be pushed out by the blood flow during deployment. The aneurysm geometry is obtained from CT images, and then discretized into a tetrahedral mesh, consisting of $18K$ elements.

The parameters of our model were set as follows: blood density: $1069 kg/m^3$, blood viscosity: $0.0035 kg/ms$, drag factor C_D: 2.2, coil radius: $0.36 \cdot 10^{-3} m$. Fig. 3 shows the velocity magnitude contours before and after placement of coils of total length $0.3m$ ($\varphi = 83.2\%$). The decrease of velocity magnitude is obvious and in accordance with recent results [2]. But we need only 25s of simulation for a cardiac cycle of 1s (using a mesh of 18k tetrahedra and a time step of 0.002s). Using a pre-computed time-varying velocity field for a complete cardiac cycle, we simulated the influence of the flow on a typical coil (radius of $0.36 \cdot 10^{-3} m$, Young modulus of $10^9 Pa$, length of $10cm$) discretized using 100 elements. The resulting simulation, illustrated in Fig. 4, was performed in real-time.

5 Conclusion

In this paper we present a method for efficiently and accurately computing blood flow in aneurysms. More importantly we introduce parameters to account for the presence of coils in the aneurysm in order to model their impact on the flow. We also model the reciprocal effect, i.e. the impact of the flow onto the coil. Both aspects are essential in the context of coil embolization planning. Our method is significantly faster than previous approaches while providing similar results. We have assessed our computation on an aneurysm presenting interesting clinical characteristics.

Regarding future directions, we acknowledge that further validation is required, however experimental data on flow patterns in aneurysms is challenging to obtain. We also want to investigate more deeply various computational strategies to obtain real-time (or near real-time) computation by using more advanced numerical schemes. Also, using the DEC method, several steps of our computation depend only on topological neighbors. This could be leveraged to provide a parallel implementation on GPU.

References

1. Groden, C., Laudan, J., Gatchell, S., Zeumer, H.: Three-dimensional pulsatile flow simulation before and after endovascular coil embolization of a terminal cerebral aneurysm. J Cereb Blood Flow Metab **21**(12) (2001) 1464–1471
2. Kakalis, N., Mitsos, A.P., Byrne, J.V., Ventikos, Y.: The haemodynamics of endovascular aneurysm treatment: a computational modelling approach for estimating the influence of multiple coil deployment. IEEE Trans Med Imaging **27**(6) (2008) 814–824
3. Stam, J.: Stable fluids. In: Proceedings of SIGGRAPH 1999, ACM (1999) 121–128
4. Desbrun, M., Kanso, E., Tong, Y.: Discrete differential forms for computational modeling. In: ACM SIGGRAPH Courses, Chapter 7. (2005)
5. Carlson, M., Mucha, P.J., Turk, G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. In: ACM Trans. Graph. (2004) 377–384
6. Robinson-Mosher, A., Shinar, T., Gretarsson, J., Su, J., Fedkiw, R.: Two-way coupling of fluids to rigid and deformable solids and shells. ACM Trans. Graph. **27**(3) (2008) 1–9
7. Dequidt, J., Marchal, M., Duriez, C., Kerien, E., Cotin, S.: Interactive simulation of embolization coils: Modeling and experimental validation. In: MICCAI. Volume 1. (2008) 695–702
8. Alderliesten, T.: Simulation of Minimally-Invasive Vascular Interventions for Training Purposes. PhD dissertation, Utrecht University (2004)
9. Duriez, C., Cotin, S., Lenoir, J., Neumann, P.F.: New approaches to catheter navigation for interventional radiology simulation. Computer Aided Surgery **11** (2006) 300–308
10. Elcott, S., Tong, Y., Kanso, E., Schroder, P., Desbrun, M.: Stable, circulation-preserving, simplicial fluids. In: ACM SIGGRAPH 2006 Courses. (2006) 60–68