Interactive Blood-Coil Simulation using Discrete Exterior Calculus

Yiyi Wei yywei@nlpr.ia.ac.cn
INRIA, 40 avenue Halley, 59650, Villeneuve d'Ascq, France
Le Fang le.fang@ec-lyon.fr
Stephane Cotin stephane.cotin@inria.fr
Songde Ma masd@most.cn

Abstract

The Discrete Exterior Calculus (DEC), a new method for solving fluid dynamics problems in a computationally efficient way is introduced. It deals with the vorticity-based equations, and updates the vorticity by a backtracking step, which makes the computation circulation-preserving at a discrete level, as well as stable. The pre-computation of operators and the use of larger time steps result in fast simulations. We illustrate this method by simulating a coil embolization procedure used for the treatment of cerebral aneurysms. This complex medical procedure requires careful planning and advanced technical skills in order to be successful. Using the DEC method, the complex blood flow patterns within the aneurysm are computed, and the two-fold interaction between coils and blood is also considered. This approach allows for fast computation of coil-flow bilateral influence, resulting in interactive simulations that can be used for advanced training of interventional radiology procedures.

Keywords: *Discrete Exterior Calculus, vortex method, real-time simulation, aneurysm embolization*

1 Introduction

Vorticity plays an important role in fluid dynamics analysis, as in many cases it is beneficial to describe flow dynamics in terms of the evolution of the vorticity field. Vortex methods for incompressible fluid simulation use vorticity-based formulations of the Navier-Stokes equations, and translate the idea that vorticity elements convect with the fluid velocity. Most of the existing works rely on well-known numerical methods to deal with the vorticity equation, for instance the finite volume method (FVM) or finite difference method (FDM) (see *e.g.* Refs. [1, 2]). However, the development of specific numerical methods dedicated to the vorticity equation could bring additional benefits, in particular in terms of computational efficiency. To this end, we introduce a recent advance in this area, known as the Discrete Exterior Calculus (DEC) method. This method, introduced by Elcott *et al.* [3] in the field of Computer Graphics, presents several benefits in terms of computational efficiency and stability. This technique handles vortices through a backtracking step which enforces the current circulation to be identical to the one at the previous time step. This makes the computation circulation-preserving at a discrete level, i.e. the circulation around any loop remains constant as the loop is advected by the fluid flow, thus complying with Kelvin's theorem. This enforces computational stability, making it possible to use larger time steps, which result in faster simulations. However, the context in which this method has been used essentially required the simulations to be visually convincing. In this paper, we aim at improving and assessing the accuracy of the method, and we illustrate this though the simulation of a complex phenomenon that requires both accurate results and minimal computation times.

1.1 Simulation of coil embolization

Over the last decade, remarkable progress has been made in the field of interventional radiology, a minimally invasive approach to the treatement of many vascular pathologies. Among theses pathologies, the endovascular treatment of aneurysms is one of the most difficult procedures, due to its technical complexity and high risk level. The main idea of coil embolization is to advance a catheter from the femoral artery to the abnormal blood vessel and once properly positioned, several coils (made of plantimum alloys) are released into the aneurysm. A blood clot will form around the coils, reducing blood flow in the aneurysm, thus preventing the aneurysm to further expand and rupture.

In this context, the development of training and planning systems can help decrease the risk of errors, in particular during the learning curve of the physician, or when dealing with complex, rare pathologies [**?**]. However, while significant progress has been made in modeling the anatomy of vascular structures [4] or the mechanical behavior of medical devices [5], there has been little work in the area of blood flow simulation near aneurysms. Yet, blood flow simulation plays a key role in interventional radiology procedures such as coil deployment. A few studies have focused on aneurysm-related haemodynamics before and after the endovascular coil embolization, for instance Groden *et al.* [6] and Kakalis *et al.* [7]. However, while these studies obtained accurate results, they relied on commercial software and the computation times were too large (ten hours or more) to be compatible with interactive simulation or even clinical practice.

1.2 Overview

In this paper, we use the Discrete Exterior Calculus approach to simulate the blood flow around the aneurysm before and after coils insertion. As the coil is relatively randomly distributed in the volume of the aneurysm, we describe its influence as a porous media. The porous terms are added into the governing equation, and solved similarly as calculating advection term. The impact of blood flow on the coils is also considered as drag force. We provide one result of our simulations performed on a patient-specific model. We first explain the discretization and solution of the governing equations in DEC approach (Seciton 2). We rely on this approach to obtain an ideal trade-off between accuracy and computational efficiency in real-time or near real-time computation of blood flow within an aneurysm, as well as the interaction between blood and coils. 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. (Section 4).

2 DEC method

In this section, we first give the governing equations of fluid, with additional terms considering the interaction of solid string materials randomly swirled in the fluid. Then we present the numerical solution of the governing equations by DEC method. Several results and discussion are shown in the end of this section.

2.1 Equations

From a statistical point of view, we introduce the notion of porous media to model the impact of solid strings randomly distributed in the flow (as a change of flow pattern and a decrease of velocity). The fluid domain $\mathscr{D}$ (2D or 3D) is divided into 3 sub-domains: a solid-free, a solid-filled subdomain and a tansitory part between them, which allows the porous parameters variate smoothly accross the first two sub-domains. However, fluid motion in all sub-domains is described uniformly by the Navier-Stokes equation of Brinkmann type in terms of vorticity:

$$\begin{aligned} \frac{\partial \boldsymbol{\omega}}{\partial t} + \mathscr{L}_{\boldsymbol{u}}\boldsymbol{\omega} = \frac{\mu}{\rho}\Delta\boldsymbol{\omega} - \frac{\mu\varphi}{\rho k}\boldsymbol{\omega} - \frac{\varphi^2 C_D}{\sqrt{k}}\nabla \times \boldsymbol{b} \\ \nabla(\varphi \boldsymbol{u}) = 0 \quad \boldsymbol{\omega} = \nabla \times \boldsymbol{u} \quad \boldsymbol{b} = \boldsymbol{u}|\boldsymbol{u}| \end{aligned} \tag{1}$$

where $\boldsymbol{u}$ and $\boldsymbol{\omega}$ are the velocity and vorticity of the fluid with constant density ρ and viscosity μ, and p is the pressure. Lie derivative $\mathscr{L}_{\boldsymbol{u}}\boldsymbol{\omega}$, equal in our case to $\boldsymbol{u} \cdot \nabla\boldsymbol{\omega} - \boldsymbol{\omega} \cdot \nabla\boldsymbol{u}$, is the advection term, and $\boldsymbol{b}$ is the porous term. 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 pores to the total solid-filled sub-domain, $\varphi = 1 - V_s / V_d$, where V_s is the accumulated volume of all solid materials, and V_d is the volume of the domain. The permeability k measures the fluid conductivity through porous media, $k = \varphi^3 / cS^2$, where c is the Kozeny coefficient (for cylinders, $c = 2$), and S is the ratio of the surface area of all solid material to the volume of the domain. The drag factor C_D can be derived from the computation of a local Reynolds number. Note that when $\varphi \to 1$ and $k \to \infty$, the last two terms, related to porous meida, disappear, therefore, (1) is the standard Navier-Stokes equation, corresponding to solid-free region.

2.2 Discretization

DEC provides a vortex method for solving the fluid equations, which discretizes the space as a simplicial complex and a corresponding dual mesh, and defines state variables as discrete forms, *i.e.*, integral values over elements of these two meshes. Therefore the solution complies 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 curl and Laplace, which are defined using basic topological and geometrical operations (see [8], [3] or [9] for details).

The 3D domain $\mathscr{D}$ is discretized as an oriented simplicial complex, *i.e.*, an oriented tetrahedral mesh, referred to as primal mesh. We denote the vertex (0-simplex) set $V = \{v_i\}$, the edge (1-simplex) set $E = \{e_{ij}\}$, the triangle (2-simplex) set $F = \{f_{ijk}\}$, and the tetrahedron set (3-simplex) $T = \{f_{ijks}\}$ ($0 \leqslant i, j, k, s \leqslant |V|$, where i, j, k, s are the serial numbers of vertices). Note that the order of subscripts i, j, k, s indicates the orientation of the simplex.

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 ($p = 0, 1, 2, 3, n = 3$ for 3D mesh) as depicted in Figure 1. As a result, in n-D space, the number of dual $(n-p)$-cell is the same as that of primal p-simplex.

We construct these two meshes, primal and dual, in view of description and transition of problem variables, which are defined as discrete p-forms, *i.e.*, scalars associated to p dimension primitives (p-cells) of either the primal or dual mesh. In fluid, velocity is described as flux, the mass of fluid passing through each edge per unit time. Thus it's a discrete 2-form $\boldsymbol{U}$, represented as a vector of size $|F|$. Similarly, the porous term $\boldsymbol{b}$ is defined as a discrete 2-form $\boldsymbol{B}$. In continuum fluid dynamcis, vorticity measures the local angular rate of rotation, and is defined as the circulation per unit area at a point. Accordingly, we describe discrete vorticity $\boldsymbol{\Omega}$ through the integral around dual faces, as a discrete dual 2-form, represented as a vector of size $|E|$.

2.3 Operators

All the vector calculus operators involved in our computation can be derived from two types of 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 the transpose of the signed incidence matrix, while 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, that is, $(\star_0)_{vv} = |v^\star| / |v|$, $(\star_1)_{ee} = |e^\star| / |e|$, $(\star_2)_{ff} = |f^\star| / |f|$, $(\star_3)_{tt} = |t^\star| / |t|$ (The superscript * is dual element of the primal one, and the operator $|\cdot|$ denotes the volume of the element). The transition between discrete forms through these fundamental operators are illustrated in Figure 1. Note that the element of star operators could be negative if the circumcenter locates outside the tetrahedron. The negative elements may create unstability of the computation. We measure the mesh quality by the percentage of the well-centered tetrahedra, and always use a mesh with over 90% quality.

Since variables and their transitions are delicately defined, we can easily get $\boldsymbol{\Omega}$ from $\boldsymbol{U}$ as follows: $\star_2$ transforms $\boldsymbol{U}$ on primal faces to $\star_2\boldsymbol{U}$ on dual edges, and then d_1^t makes a sum on each dual faces by accumulating $\star_2\boldsymbol{U}$ on all incident dual edges, *i.e.*, $\boldsymbol{\Omega} = d_1^t \star_2 \boldsymbol{U}$, a dual 2-form. And this also

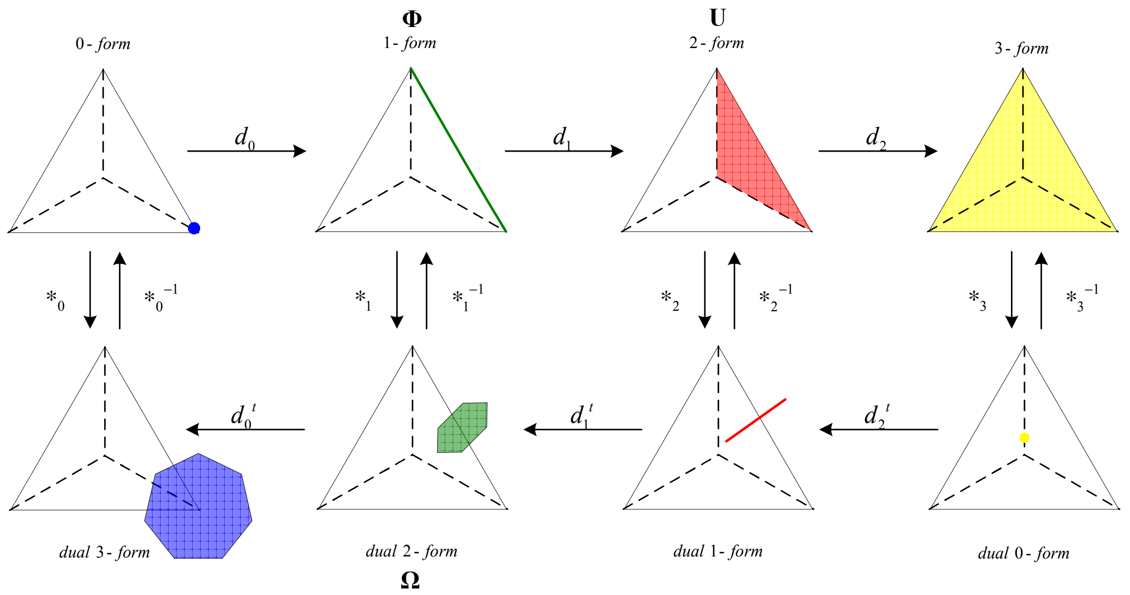

Figure 1: The duality of primal and dual mesh: The first line shows the primal simplex, whose dual elements are below. Physical variables $\boldsymbol{\Phi}$, $\boldsymbol{U}$ and $\boldsymbol{\Omega}$, defined as discrete forms, can be transfered by two fundamental operators d and $\star$.

explains how we build $\nabla\times$ by the basic operators. Following similar principles, the operators (div $\nabla\cdot$, curl $\nabla\times$, Laplace Δ) used in equation (1) can be constructed from d and $\star$: $\nabla\cdot = d_2$, $\nabla\times = d_1^t\star_2$, $\Delta = d_1^t \star_2 d_1 + \star_1 d_0 \star_0^{-1} d_0^t \star_1$.

2.4 Solving Fluid Equations

The vorticity-based equation (1) consists of three ingredients, an advection term $\mathscr{L}_{\boldsymbol{u}}\boldsymbol{\omega}$, a viscous term $\Delta\boldsymbol{\omega}$, and last two porous ones. Simply speaking, the advection term 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 [3]. In this 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 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 viscous term and porous terms, linear solution are used, and an implictit scheme could be chosen for the purpose of stability.

After updating the vorticity, the recovery of velocity field $\boldsymbol{u}$ from vorticity $\boldsymbol{\Omega}$ is needed for backtracking in the next step. Considering $\boldsymbol{\Omega} = d_1^t \star_2 \boldsymbol{U}$, flux $\boldsymbol{U}$ can be obtained directly via a Poisson equation [3]. Here we also add boundary constraints into the equation by the method of Lagrange multipliers. And then we find out the unique velocity vector $\boldsymbol{u}_t$ at each dual vertex whose projection along the incident dual edges is consistent with the flux of the corresponding primal edges. See Algorithm 1 for an overview of the computation process.

***Algorithm 1*: Pseudocode of Fluid Computation**

```
//Load mesh and compute the operators
$\boldsymbol{C} \leftarrow d_1^t\star_2$
$\boldsymbol{L} \leftarrow d_1^t \star_2 d_1 + \star_1 d_0 \star_0^{-1} d_0^t \star_1$
$\boldsymbol{L'} \leftarrow$ LagrangeMultiplier($L$)

//Time stepping $h$
```

```
loop
    //advect vorticities
    $\hat{\boldsymbol{c}} \leftarrow$ backtrackTetrahedronCenters($h$)
    $\hat{\boldsymbol{v}} \leftarrow$ interpolateVelocity($\hat{\boldsymbol{c}}$)
    for each dual face $f$
        $\boldsymbol{\Omega}_f \leftarrow 0$
        for each dual edge $(i,j)$ on the boundary of $f$
            $\boldsymbol{\Omega}_f \leftarrow \boldsymbol{\Omega}_f + \frac{1}{2}(\hat{\boldsymbol{v}}_i + \hat{\boldsymbol{v}}_j) \cdot (\hat{\boldsymbol{c}}_i - \hat{\boldsymbol{c}}_j)$
    //add porous terms
    for each dual face $f$ in the coil-filled subdomain
        $\boldsymbol{\Omega}_f \leftarrow \frac{\rho k}{\rho k + \mu\varphi h}\boldsymbol{\Omega}_f + \frac{\varphi^2 C_D}{\sqrt{k}} h\, (\boldsymbol{C} \times \boldsymbol{B})_f$
    //add viscous term
    $\boldsymbol{\Psi} \leftarrow$ linearSolver( $(\star_1 - \mu \boldsymbol{L} h)\boldsymbol{\Psi} = \boldsymbol{\Omega}$) )
    $\boldsymbol{\Omega} \leftarrow \star_1 \boldsymbol{\Psi}$
    //recover flux from vorticity
    $\tau \leftarrow$ setVaringBoundaryConditions()
    $(\boldsymbol{\Phi} \mid \lambda) \leftarrow$ linearSolver( $\boldsymbol{L}'(\boldsymbol{\Phi} \mid \lambda) = (\boldsymbol{\Omega} \mid \tau)$ )
    $\boldsymbol{U} = d_1 \boldsymbol{\Phi}$
    //compute velocity at triangle centers and porous term at edges
    for each tetrahedron $t$
        $\boldsymbol{u}_t \leftarrow$ linearSolver( $\boldsymbol{A}_t \boldsymbol{u}_t = \boldsymbol{U}_t$ )        // $\boldsymbol{A}_t$ is a project matrix of $t$
    $\boldsymbol{B} \leftarrow$ integrate($\boldsymbol{u}$)
end loop
```

3 Validation

In order to validate the DEC method, a two-dimensional test simulation of blood flow on two meshes with the same geometry but different resolution is performed by comparing with the results of FLUENT. Laminar flow with flux boundary conditions at both inlet and outlet sections is considered. The viscosity of blood is strong so that the corresponding Reynolds number is small enough for the laminar assumption. The first two images in Figure 2 are the contours of velocity magnitude and streamlines computed by the DEC method and FLUENT software respectively on the identical mesh of 210,177 triangles, while the last image is the result by DEC on a coarser mesh of only 19,753 triangles. This comparison shows the similarity between the two methods, even when we reduce the number of mesh elements by 10 times. The streamlines show perfect agreement for the vortex structures, including the position of the vortex center in the aneurysm. Besides, we have also investigated other physical variables, which are also in good agreement between DEC and FLUENT. It should be noted that the method is more sensitive to the mesh quality than the mesh resolution, as numerical unstabilities may occur when elements of the star operator become negative (if the circumcenter is located outside an element). We measure the mesh quality by the percentage of well-centered tetrahedra, and in the results presented here, all meshes have at least 90% of well-centered elements. Although the DEC method does not have high order accuracy, we did not find obvious numerical dissipation from this test case. These results illustrate the potential of the proposed method, especially in the eld of medical imaging or intervention planning, where computational efficiency is more important than small-scale details.

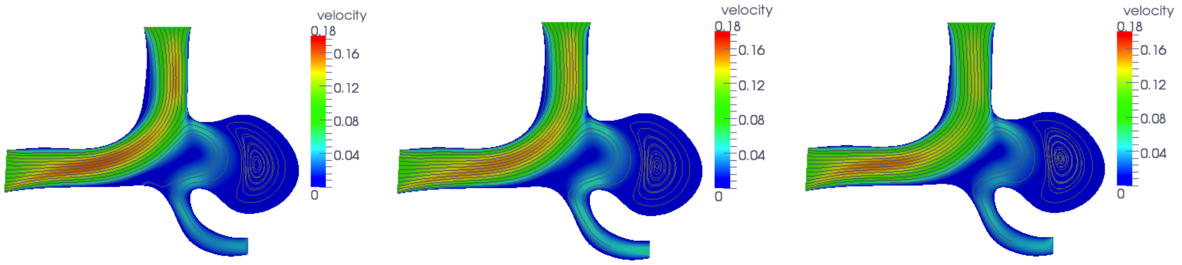

Figure 2: Comparison between simulation results: (left) simulation based on DEC method, (mid) simulation from a commercial software. Identical meshes (with 210,177 triangles) and boundary conditions were used in both methods. (right) simulation based on DEC method on a coarser mesh (with only 19,754 triangles).

4 Simulation of Coil Embolization

Computaitonal Fluid Dynamics (CFD) 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. As an application of DEC in medical procedures, we simulate the process of planting coils into an aneurysm, and evaluate the embolization effect. In this section, we describe the models of blood, coil and their two-fold interactions as well. We also present several simulation results.

Blood flow is assumed to be a Newtonian fluid with constant density $\rho = 1069kg/m^3$ and constant dynamic viscosity $\mu = 0.0035kg/m\cdot s$, and its behavior is modeled by NS equations. The model of the coil is based on the work of Dequidt *et. al.* [5] 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.

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, which is particularly important during the first stage of coil deployment. The interaction we consider in this paper 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. In fact, the last term of Navier-Stokes equation of Brinkmann type 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:

$$F_D = \frac{1}{2}C_D\rho\vec{u}_\perp|\vec{u}_\perp|Adh, \tag{2}$$

where C_D is the drag coefficient, $\vec{u}_\perp$ is the velocity orthogonal to the coil, A is the cross-sectional area of the coil, and dh is the length of the coil section. The velocity component parallel to the coil is neglected, since it only produces shear force on the coil, which is insignificant for the deformation compared to the drag force. Hence, the reacting force on the coil only depends on local fluid velocity.

On the other hand, the impact of the coil onto the flow is considered as porous media in the fluid, where coils are described from a statistical point of view, translating the idea that, after deployment, coils are randomly distributed in the aneurysm. As a result, we apply the model and method describe in section 2 to simulate the blood flow with the influence of the deployed coils within the aneurysm.

To combine and use the models above in our integrate system, we assume that 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. Solving the equation (1) by DEC method, we first 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.

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 $8.2\cdot 10^{-3}m$. The corresponding Reynolds number is smaller than 100, thus the flow is laminar. Such aneurysms are difficult to be treated, as coils might be

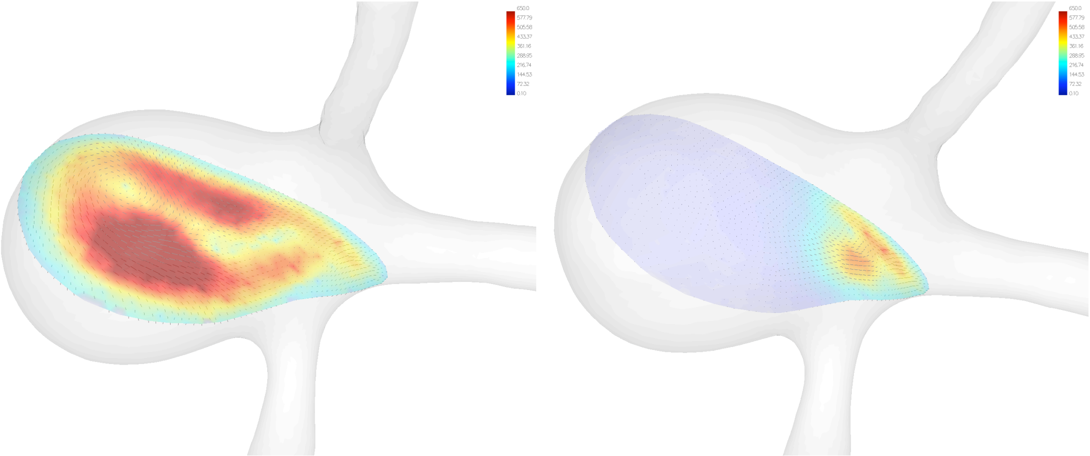

Figure 3: (Left): velocity magnitude in aneurysm with no coil; (Right): with 16.8% of the volume filled with coils. Aneurysm model composed of of $18K$ tetrahedral elements.

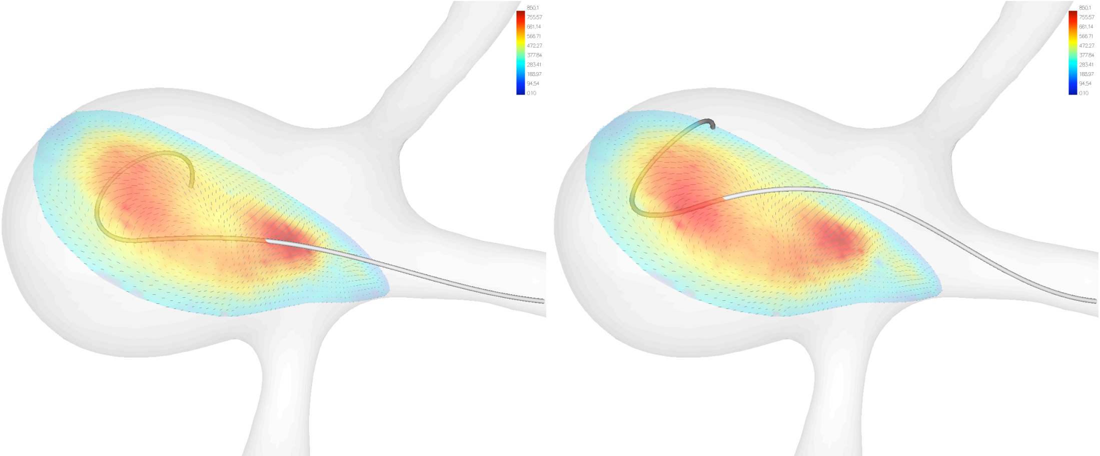

Figure 4: (Left): the coil shape without the effect of blood flow; (Right) with flow forces.

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.

Figure 3 shows the velocity magnitude contours before and after the placement of coils. The decrease of velocity magnitude is obvious and in accordance with recent results [7]. Note that in this case, 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 $109Pa$, length of $10cm$) discretized using 100 elements. The resulting simulation, illustrated in Figure 4, was performed in real-time.

5 Conclusion

In this contribution we introduce the DEC 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. Comparisons with results obtained from commercial software show that this method is accurate, but also presents interesting properties in terms of scalability. The numerical error introduced by using a coarser mesh is limited, allowing for fast computation while minimizing the impact on the accuracy.

Regarding future directions, we want to further investigate and hopefully improve the current limitations of DEC, such as its sensitivity to the mesh quality, the accuracy in the backtracking step and interpolation method. Besides, we acknowledge that further validation is required, both on DEC method and the medical simulation. 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.

Acknowledgments

We would like to thank Jeremie Allard, Frederick Roy for their valuable help in this research. This work is supported by LIAMA, the Sino French Lab in Computer Science, Automation and Applied Mathematics, and INRIA, the French National Institute for Research in Computer Science and Control.

References

[1] Pascazio, G., Napolitano, M.: A staggered-grid finite volume method for the vorticity-velocity equations. Computers & Fluids **25**(4) (1996) 433–446

[2] Liu, C.: Numerical solution of three-dimensional Navier-Stokes equations by a velocity-vorticity method. International Journal for Numerical Methods in Fluids **35** (2001) 533–557

[3] Elcott, S., Tong, Y., Kanso, E., Schroder, P., Desbrun, M.: Stable, circulation-preserving, simplicial fluids. In: SIGGRAPH 06: ACM SIGGRAPH 2006 Courses, New York, NY, USA, ACM (2006) 60–68

[4] Wu, X., Luboz, V., Krissian, K., Cotin, S., , Dawson, S.: Segmentation and reconstruction of vascular structures for 3d real-time simulation. Medical Image Analysis (2010)

[5] 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

[6] 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 Dec) 1464–1471

[7] 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

[8] Desbrun, M., Kanso, E., Tong, Y.: Discrete differential forms for computational modeling. In: ACM SIGGRAPH Course Notes on Discrete Differential Geometry, Chapter 7. (2005)

[9] Wei, Y., Cotin, S., Fang, L., Allard, J., Pan, C., Ma, S.: Toward real-time simulation of blood-coil interaction during aneurysm embolization. In: MICCAI. (2009)