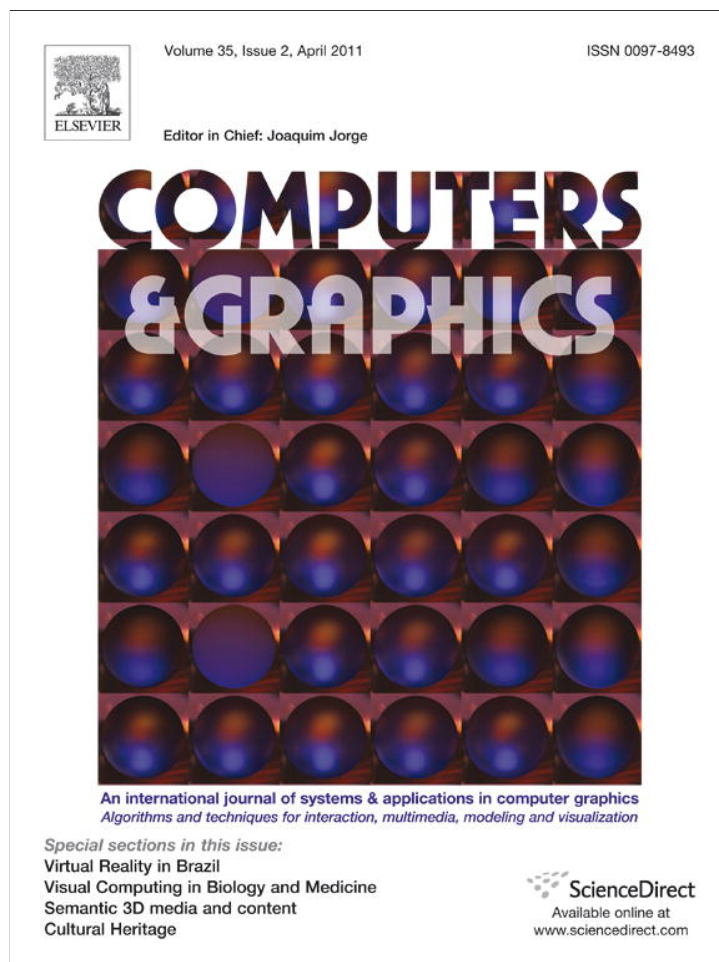


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

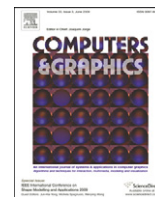
In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Visual Computing in Biology and Medicine

Interactive blood-coil simulation in real-time during aneurysm embolization

Yiyi Wei ^{a,b,*}, Stéphane Cotin ^a, Jérémie Allard ^a, Le Fang ^c, Chunhong Pan ^b, Songde Ma ^b^a Shaman Project, INRIA, IRCICA, 50 avenue de Halley, 59650 Villeneuve d'Ascq, France^b LIAMA CASIA, No.95 Zhongguancun East Road, 100190 Beijing, China^c LMFA, Ecole centrale de Lyon, 69130 Ecully, France

ARTICLE INFO

Article history:

Received 7 October 2010

Received in revised form

4 January 2011

Accepted 14 January 2011

Available online 2 February 2011

Keywords:

Aneurysm embolization

Fluid–solid interaction

Porous media

Drag force

Discrete exterior calculus

ABSTRACT

Over the last decade, remarkable progress has been made in the field of endovascular treatment of aneurysms. Technological advances continue to make it possible for 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 describe a complete process for patient-specific simulations of coil embolization, from mesh generation with medical datasets to computation of coil–flow bilateral influence. We propose a new method for simulating 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 turn 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. Finally, we provide the blood flow simulation results on several aneurysms with interesting clinical characteristics both in 2D and 3D, as well as comparisons with a commercial package for validation. The coil embolization procedure is simulated within an aneurysm, and pre- and post-operative status is reported.

© 2011 Elsevier Ltd. All rights reserved.

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 (Fig. 1) 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 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 [12]. The computer aided system, allowing interventional radiologists to select different coils and test their behavior in a patient-specific environment, 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.

1.1. 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 significant progress has been made in modeling the anatomy of

* Corresponding author at: Shaman Project, INRIA, IRCICA, 50 avenue de Halley, 59650 Villeneuve d'Ascq, France. Tel.: +33 3 62 53 15 77.

E-mail addresses: yiyi511@gmail.com, yywei@nlpr.ia.ac.cn (Y. Wei), stephane.cotin@inria.fr (S. Cotin), jeremie.allard@inria.fr (J. Allard), le.fang@ec-lyon.fr (L. Fang), chpan@nlpr.ia.ac.cn (C. Pan), masd@most.cn (S. Ma).

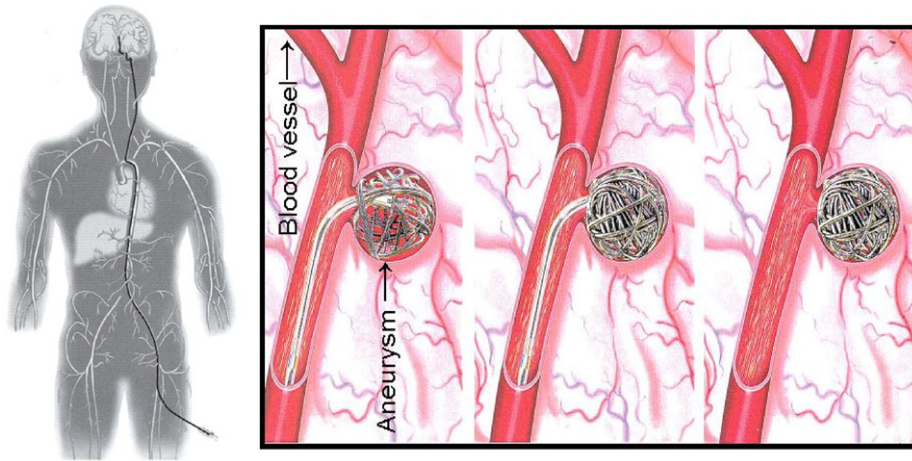


Fig. 1. The endovascular treatment of cerebral aneurysm: insert a catheter into the femoral artery and use the vascular network to reach the diseased vessel, place several coils into the aneurysm through the catheter, and withdraw the catheter.

vascular structures [25] or the mechanical behavior of medical devices [7], 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 endovascular coil embolization. Groden et al. [13] 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. Castro et al. [4] investigated the effects of unequal physiologic flow conditions on the hemodynamics of anterior communicating artery aneurysms. Kakalis et al. [15] 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. All of these studies assumed the blood to be a Newtonian fluid with constant viscosity. Bernsdorf and Wang [2] used a Lattice Boltzmann solver with a Carreau–Yasuda model to capture non-Newtonian rheology of blood flow in cerebral aneurysms. Cebal et al. [5] reported a sensitivity analysis of hemodynamic characteristics with respect to variations of several variables, such as inflow, flow division, viscosity. As these studies aimed at accurate Computational Fluid Dynamics (CFD) simulation, they rely on commercial software, and the computation times (dozens of hours in general) are incompatible with interactive simulation or even clinical practice. In order to reach fast computation in this application, another fast alternative is to combine patient-specific measurement data and CFD [18], which requires to set a large database of pre-computed flow fields for a given anatomical location. In this paper, we aim at a planning system for endovascular surgeries, which require much faster computation of the blood flow, as well as the interaction between the blood and the device.

Generally speaking, accuracy and efficiency are two significant pursuits in numerical calculation, but unfortunately always contradictory. Lots of computationally efficient techniques have been developed in the field of Computer Graphics, which essentially required the simulations to be visually convincing, but not physically accurate. The stable fluids approach [22] 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, a vortex method for incompressible fluid simulation, the Discrete Exterior Calculus (DEC) was

introduced [10]. It presents several benefits in terms of stability and computational efficiency. 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 through the simulation of a complex phenomenon that requires both accurate results and minimal computation times.

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. This is particularly true in the case of flow in aneurysms, where the bulge along the artery creates vortices which impact both the development of the aneurysm and the embolization procedure. 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 Refs. [19,16] for instance). Many approaches have been proposed on the basis of the FVM or FDM method, but their main drawback in our case remains their computation times, which are often in the order of hours while we aim at a few seconds.

In the field of computer animation, various models have been put forth to study the interaction between fluids and solids. For instance, Carlson et al. [3] solved this problem by considering the 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. [20] 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 has mainly focused on training rather planning (besides the work of Ref. [7]) and has been limited to the simulation of flexible devices (see Ref. [1], or Ref. [9] for instance).

In this paper, we present a novel technique for accurately computing (in real-time or near 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 achieve minimal computation times, but we introduce several improvements as well as a much deeper analysis of the results, in the context of a very different application than initially aimed by the DEC method. First we add extra terms to the Navier–Stokes equation in order to describe the interaction between blood and the deformable solid that corresponds to the coil (Section 2.2). We also improve the numerical stability of the method by using a more advanced interpolation scheme, and more importantly by optimizing the mesh quality used in the computation (Section 2.1). A detailed analysis of the results and comparison with a reference software is also performed to understand the accuracy of the method and the link between accuracy and mesh resolution (Section 3). Our results 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 (Section 4.2).

2. Modeling interactions between blood and coils

In this section, we first present the process used to extract and mesh the aneurysm and adjacent vessels from patient-specific data (Section 2.1). Then we describe the computational approach for computing blood flow in and around an aneurysm (Section 2.2). 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 2.3). This is particularly important during the first stage of coil deployment.

2.1. Mesh generation and boundary conditions

Fig. 2 presents the patient-specific mesh generation process. We start from a volumetric medical dataset where the brain

vessels are visible (such as CTA or 3D rotational angiography), which is then segmented. From the segmented image, we reconstruct a surface model. In our case, we rely on a combination of techniques such as Marching Cubes algorithm followed by a small deformation of the resulting mesh (relaxation) to smooth the surface while adjusting the surface points location using the original image data (and its gradient). As we are only interested in creating a mesh for a region close to the aneurysm (and not for the full vascular network) sometimes only an iso-surface from the measured raw density is sufficient to obtain a surface mesh. Yet it highly depends on the quality of the data, and in some instances more robust algorithms can be necessary, see Ref. [14] for example.

From this surface model, after a plane or a sphere is manually positioned to define and cut the inlet and outlet vessels, a tetrahedral mesh is created. To obtain accurate results, numerical simulations require a high-quality mesh, as measured in our case by the percentage of well-centered elements. While this is relatively easy to obtain in 2D, it is a more challenging problem in 3D.

In this work, we used an interleaved optimization algorithm based on Delaunay refinement and Lloyd optimization [24], as implemented by the CGAL library [6]. Each refinement step acts on the size of the elements, while each optimization step acts on the shape of the elements. With this method, we can also define a size field according to our requirements, such as smaller elements near the boundary. We also control the fidelity of the generated mesh to the previously obtained surface, by specifying the distance tolerance between the two. After several iterations, we obtain 2D triangular meshes with over 98% well-centered elements, and 3D tetrahedral meshes with over 95% well-centered elements.

To compute the blood motion, the flux through all the boundaries must be formulated. In this paper, we assume the vessel walls to be rigid, thus no-slip and no-penetration conditions are applied, and flux and velocity are set zero at the vessel walls. At the inlet and outlets, we have different conditions for two experiments, *i.e.*, a sinusoidally pulsatile flow for the simulation of coil embolization, and non-varying flow (the mean value of the pulsatile flow) for the validation against another numerical solution. For the inlet, we choose these values based on the measurements from magnetic resonance imaging [17]. As the blood is assumed to be incompressible, the total flux at the outlet(s) should be equal to inflow. If there are more than one outlet, the distribution of the blood flow is calculated by

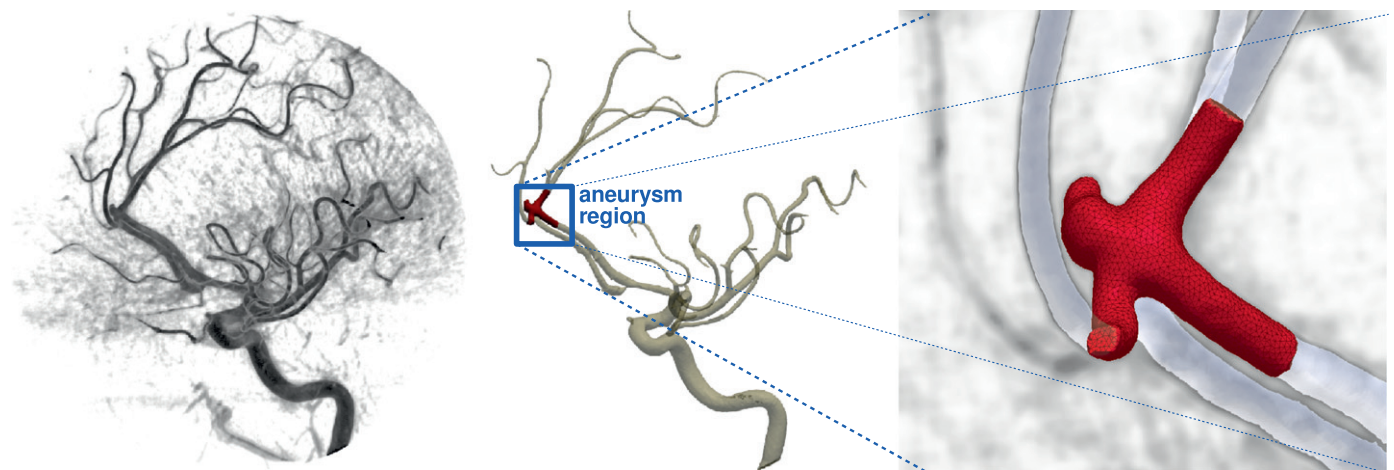


Fig. 2. Mesh generation: (left) patient-specific data, (middle) reconstructed vessels surface model, (right) tetrahedral mesh generated in the region of the aneurysm.

Bernoulli's principle in the assumption that the pressure is identical at two outlets.

2.2. Blood motion

2.2.1. Porous media model

We divide the fluid domain \mathcal{D} (2D or 3D) into 3 sub-domains, a coil-free, a coil-filled sub-domain and a transitory part between them, which allows the porous parameters variate smoothly across the first two sub-domains. However, blood motion in all sub-domains is described uniformly by a Navier–Stokes equation of Brinkmann type:

$$\nabla(\varphi \mathbf{u}) = 0,$$

$$\frac{\partial(\varphi \rho \mathbf{u})}{\partial t} + \rho(\mathbf{u} \cdot \nabla)(\varphi \mathbf{u}) = \varphi \nabla p + \mu \Delta(\varphi \mathbf{u}) - \frac{\varphi^2 \mu}{k} \mathbf{u} - \frac{\varphi^3 C_D \rho}{\sqrt{k}} \mathbf{u} |\mathbf{u}| \quad (1)$$

where \mathbf{u} is the velocity of the fluid with density ρ and viscosity μ , and p is the pressure. The symbol ∇ is the vector of spatial partial derivatives, more precisely, $\nabla = (\partial/\partial x, \partial/\partial y)$ in 2D or $(\partial/\partial x, \partial/\partial y, \partial/\partial z)$ in 3D. We also use the notation $\Delta = \nabla \cdot \nabla$. 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 coil-filled sub-domain, $\varphi = 1 - V_{coil} / V_{an}$, 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, $k = \varphi^3 / cS^2$, 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. Note that when $\varphi \rightarrow 1$ and $k \rightarrow \infty$, the last two terms, related to porous media, disappear, therefore, (1) is the standard Navier–Stokes equation, corresponding to coil-free region.

In order to get rid of the pressure term, we take the curl of (1). As density and porosity remain constant in each sub-domain, we have

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

$$\nabla(\varphi \mathbf{u}) = 0 \quad \boldsymbol{\omega} = \nabla \times \mathbf{u} \quad \mathbf{b} = \mathbf{u} |\mathbf{u}| \quad (2)$$

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

2.2.2. 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 (Fig. 3). 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 curl and Laplace, which are defined using basic topological and geometrical operations (see [8] or [10] for details).

Discretization: The 3D domain \mathcal{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) $T = \{t_{ijkl}\}$ ($0 \leq i, j, k, s \leq |V|$, 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 Fig. 3. 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. Physical quantities 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 triangle (primal 2-cell) per unit time. Thus it is a discrete 2-form \mathbf{U} , represented as a vector of size $|F|$. Similarly, the porous term \mathbf{b} is defined as a discrete 2-form \mathbf{B} . In continuum fluid dynamics, 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 (dual 2-cells), as a discrete dual 2-form, represented as a vector of size $|E|$.

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 ,

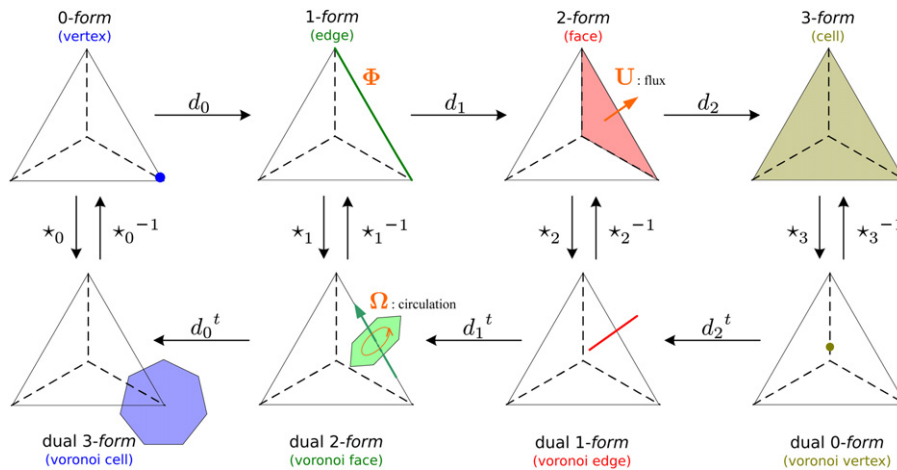
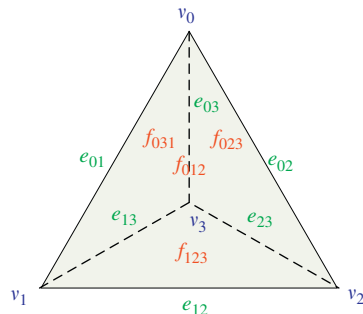


Fig. 3. The duality of primal and dual mesh: The first line shows the primal simplex, whose dual elements are below. Physical variables Φ , \mathbf{U} and $\boldsymbol{\Omega}$, defined as discrete forms, can be transferred by two fundamental operators d and \star .



0-simplex	$\{v_0, v_1, v_2, v_3\}$
1-simplex	$\{e_{01}, e_{02}, e_{03}, e_{12}, e_{13}, e_{23}\}$
2-simplex	$\{f_{012}, f_{023}, f_{031}, f_{123}\}$
3-simplex	$\{t_{0123}\}$

d_0	v_0	v_1	v_2	v_3
e_{01}	-1	1		
e_{02}	-1		1	
e_{03}	-1			1
e_{12}		-1	1	
e_{13}		-1		1
e_{23}			-1	1

d_1	e_{01}	e_{02}	e_{03}	e_{12}	e_{13}	e_{23}
f_{012}	1	-1		1		
f_{023}		1	-1			1
f_{031}	-1		1		-1	
f_{123}				1	-1	1

d_2	f_{012}	f_{023}	f_{031}	f_{123}
t_{0123}	1	1	1	1

Fig. 4. Simplex sets and differential operators in the case of a single tetrahedron.

maps p -forms to $(p+1)$ -forms on the primal mesh, represented by the transpose of the signed incidence matrix (an example is given in Fig. 4), 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 \star is dual element of the primal one, and the operator $|\cdot|$ denotes the volume of the element.) Once we have the Hodge star operators, the variables can be transferred from primal p -cell to dual $(n-p)$ -cell. For example, flux \mathbf{U}_f through the triangle f (primal 2-cell), can be transferred to the integral value of velocity on the corresponding dual edge f^\star (dual 1-cell), $|f^\star| \star (\mathbf{U}_f / |f|)$, i.e., $(\star_2)_{ff} \star \mathbf{U}_f$. Note that the element of star operators could be negative if the circumcenter locates outside the tetrahedron. The negative elements may create divergence or instability of the computation. We measure the mesh quality by the percentage of the well-centered tetrahedra, and always use amesh with over 90% quality. In the following subsection we will discuss how to improve the mesh quality. The transition between discrete forms through these fundamental operators are illustrated in Fig. 3.

Since variables and their transitions are delicately defined, we can easily get Ω from \mathbf{U} as follows: \star_2 transforms \mathbf{U} on primal faces to $\star_2 \mathbf{U}$ on dual edges, and then d_1^t makes a sum on each dual faces by accumulating $\star_2 \mathbf{U}$ on all incident dual edges, i.e., $\Omega = d_1^t \star_2 \mathbf{U}$, a dual 2-form. And this also 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 Eq. (2) can be constructed from d and \star : $\nabla \cdot = d_2$, $\nabla \times = d_1^t \star_2$, $\Delta = d_1^t \star_2 d_1 + \star_1 d_0 \star_0^{-1} d_0^t \star_1$.

Solving fluid equations: The vorticity-based equation (2) consists of three ingredients, an advection term $\mathcal{L}_u \omega$, a viscous term $\Delta \omega$, 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 [10]. 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 solutions are used, and an implicit scheme could be chosen for the purpose of stability.

After updating the vorticity, the recovery of velocity field \mathbf{u} from vorticity Ω is needed for backtracking in the next step. Considering $\Omega = d_1^t \star_2 \mathbf{U}$, flux \mathbf{U} can be obtained directly via a Poisson equation [10]. Here we also add boundary constraints into the equation by the method of Lagrange multipliers. And then we find out the unique velocity vector \mathbf{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. Main steps of the fluid computation using DEC.

```

//Load mesh and compute the operators
C ← d1^t * 2
L ← d1^t * 2 d1 + * 1 d0 * 0^-1 d0^t * 1
L' ← LagrangeMultiplier(L)

//Time stepping h
loop
//advect vorticities
c-hat ← backtrackTetrahedronCenters(h)
v-hat ← interpolateVelocity(c-hat)
for each dual face f
    Omega_f ← 0
    for each dual edge (i, j) on the boundary of f
        Omega_f ← Omega_f + 1/2 (v-hat_i + v-hat_j) · (c-hat_i - c-hat_j)
//add porous terms
for each dual face f in the coil-filled sub-domain
    Omega_f ← (rho*k / (rho*k + mu*phi*h)) * Omega_f + (phi^2 * C0 / sqrt(k)) * h * (C × B)_f
//add viscous term
Psi ← linearSolver(( * 1 - mu * Lh ) Psi = Omega)
Omega ← * 1 Psi
//recover flux from vorticity
tau ← setVaringBoundaryConditions()
(Phi | lambda) ← linearSolver(L' (Phi | lambda) = (Omega | tau))
U = d1 Phi
//compute velocity at triangle centers and porous term at edges
for each tetrahedron t
    u_t ← linearSolver(A_t u_t = U_t) // A_t is a project matrix of t
    B ← integrate(u)
end loop
    
```

2.3. 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 Eq. (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:

$$\mathbf{F}_D = \frac{1}{2} C_D \rho \mathbf{u}_\perp |\mathbf{u}_\perp| A dh, \quad (3)$$

where \mathbf{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 model of the coil 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. Validation

In order to assess the overall approach, we have performed a series of validation tests for both the coil and the blood flow models. The coil model has been validated against an actual coil, that was scanned at different steps of its deployment. This study was reported in [7] and showed a very good similarity between the simulated coil behavior and the real coil deformation. The main difference with our previous work is that we have further improved the computation time for the coil model, in particular using a GPU-based version of the block tri-diagonal band linear solver. Typical computation times are less than 5 ms per time step for a coil model with 200 nodes.

To assess blood flow computation using the DEC method, we performed two series of comparative tests against FLUENT¹ which is one of the leading software in the field of fluid dynamics. In this case, we mostly aim at validating the numerical accuracy of the DEC method than the actual ability to accurately describe blood flow patterns near the aneurysms. Such an analysis is very complex to achieve, in particular in very small vessels such as the ones in the brain. The first comparison consists of a 2D simulation of blood flow on two meshes with the same geometry but a 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. And we pick up two interesting geometries of the aneurysm, one with a large neck and the other with a small neck. The first two columns in Fig. 5 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 (first row) and 35,688 triangles (second row), while the last image is the result by DEC on a coarser mesh of only 19,753 triangles (first row) and 8944 triangles (second row). 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. Although the DEC method does not have high order accuracy, we did not find any obvious numerical dissipation from this test case. It should be noted that the method is more sensitive to the mesh quality than the mesh resolution, as

numerical instabilities may occur when elements of the star operator become negative (if the circumcenter is located outside an element). In the results presented here, all meshes have at least 90% of well-centered elements.

We also performed a 3D comparison. The computational domain has 32,453 tetrahedral elements, which is much coarser if comparing with the 2D cases. As a result, we show the comparisons of the velocity magnitude in Fig. 6. Although the mesh is very coarse, similar flow structures are captured. In both cases there is a low-speed region surrounded by high-speed flows, which represents a local vortex in this section. But obviously, we found less blood flows into the sac of the aneurysm by the DEC method, and also a difference of kinetic energy exists between the two approaches. In addition, we can also find the streamlines in Fig. 7, which show the similar movement of blood. These results show that the DEC method can also capture the large-scale flow structure, even on a very coarse mesh. However, small-scale differences exist between the two approaches, which may stem from numerical dissipation, but could be reduced by using finer meshes and higher-order interpolation methods (but increase the computation time, which is an issue when dealing with applications that require real-time or near real-time computation).

When real-time simulation is required (for training, intervention planning, or augmented reality), a trade-off between computation time and accuracy has to be found. The accuracy is highly dependent on the mesh resolution. As the resolution decreases (*i.e.*, the average edge size increases), the kinetic energy decreases in both DEC and FLUENT results, but less rapidly for the DEC case. In our case, computational efficiency is more important than small-scale details. In this context, the DEC method offers some advantages over other methods, in particular it is less sensitive to the mesh resolution. As mesh resolution and computation times are related, we can run faster simulations with the DEC method. While these results are in favor of our approach, they only rely on comparisons between two numerical approaches. To further prove the reliability of our results, other methods could be used in the comparison, but ultimately, the validation should involve a comparison with actual blood flow data. Such information can be obtained using MRI or angiograms [23].

In order to show the effect of coil embolization, another simulations are performed on an aneurysm of particular interest, with a large sac of volume $8.976 \times 10^{-7} \text{ m}^3$ and a wide neck of dimension $8.2 \times 10^{-3} \text{ m}$. Such aneurysms are difficult to treat, as coils might be pushed out by the blood flow during deployment. Fig. 8 shows the velocity magnitude contours before and after placement of coils of total length 0.3 m ($\varphi = 83.2\%$). The decrease of velocity magnitude is obvious and in accordance with recent results [15].

4. Real-time simulation of coil embolization

In this section, we show how the two previously described models can be combined and used in a real-time simulation of coil embolization. Using the method described in Section 2.2 we show how it is possible to achieve real-time simulation when relying on “coarse” meshes (*i.e.*, meshes with less than 20,000 elements) and how a pre-computation scheme can be used when higher accuracy of the solution is sought. Typically, real-time computation can be achieved with a good accuracy for applications where real-time simulation is required (training system for instance), yet interactivity is still possible in other applications, but with a different approach described in the second part of this section.

¹ FLUENT is a commercial product of ANSYS, <http://ansys.com>

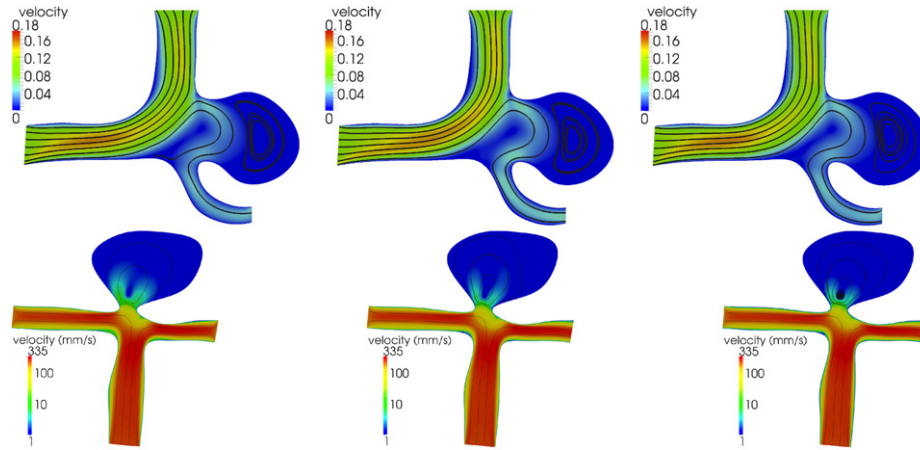


Fig. 5. Comparison between 2D simulation results. In each row, results are based on identical boundary conditions but different mesh resolutions (the first two columns rely on an identical fine mesh, while the last result uses a different, coarser mesh). Left: simulation based on DEC method, middle: simulation using FLUENT, a commercial CFD software, right: simulation based on DEC method on a coarser mesh. Top row: fine mesh resolution consists of 210,177 triangles while the coarse mesh only contains 19,754 triangles. Bottom row: fine mesh resolution consists of 35,688 triangles while the coarse mesh only contains 8944 triangles. We can see that nearly identical results are obtained between the proposed method and the commercial software when using a fine mesh, but also that significantly decreasing the mesh resolution has little impact on the solution computed by DEC.

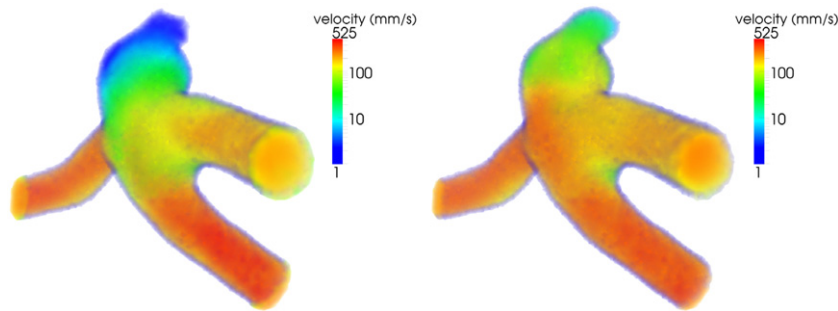


Fig. 6. Comparison between 3D simulation results using velocity magnitude by volume rendering: (left) simulation based on DEC method, (right) simulation from FLUENT (a commercial software). Identical meshes (with 32,453 tetrahedral elements) and boundary conditions were used in both methods.

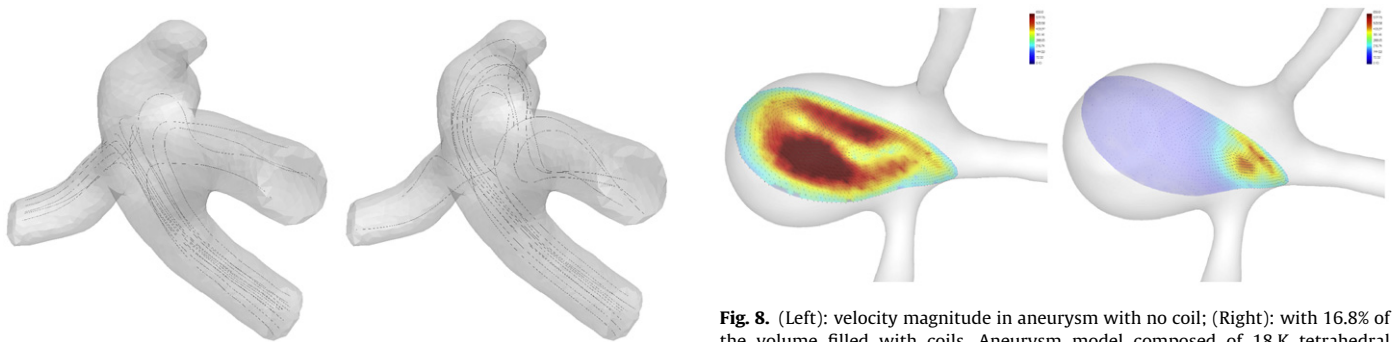


Fig. 7. Comparison between 3D simulation results using streamlines: (left) simulation based on DEC method, (right) simulation from FLUENT (a commercial software). Identical meshes (with 32,453 tetrahedral elements) and boundary conditions were used in both methods.

4.1. Real-time simulation on coarse meshes

We have seen in Section 3 that a higher mesh resolution permits to achieve a higher level of accuracy in the solution, but that the error induced by a lower resolution mesh is limited. As in other applications where real-time computation is sought, the objective is then to reach the best trade-off between accuracy and computation time. In our method, as in other numerical techniques such as the finite element method, the size of the matrices directly relates to the number of elements in the mesh (see *Algorithm 1*). Two linear systems, for the diffusion and

Fig. 8. (Left): velocity magnitude in aneurysm with no coil; (Right): with 16.8% of the volume filled with coils. Aneurysm model composed of 18 K tetrahedral elements.

Laplacian computations, need to be solved at each time step. In our approach, solving them is the most time consuming step, the second most time consuming step being the backtrack step (see Section 2.2.2). In order to reach real-time or interactive rates in the simulation, we then need to limit the number of elements in the mesh *and* rely on optimized linear solvers.

Many numerical techniques can be used to improve the efficiency of this process. Both the diffusion and Laplacian matrices specifying the systems of equations that need to be solved are constant over time. Therefore, we rely on a direct solver which factorizes the matrices only once, during initialization. Within the simulation itself, the solution of each system of equations is computed by efficient sparse triangular solve

operations using the pre-computed factorized matrices. In our experiments we used the multi-threaded sparse symmetric indefinite solver in the Pardiso [21] library to factorize and solve the diffusion and Laplacian matrices. In addition to the computational efficiency, the ability to support large time steps while remaining stable must be considered to achieve interactive performances. Due to the considerable local variations that can appear in the flow inside blood vessels and aneurysms with complex geometries, the precision of the time integration scheme used for the advection stage is critical. Many solutions are available and have been studied, from simple single-step Euler integration scheme, to adaptive multi-step methods and higher-order schemes. In our experiments, we found the third-order Runge–Kutta scheme to provide an interesting trade-off between accuracy and computational cost. It is able to increase the achievable time step compared to single-step methods by a factor of four to five times. Using a more advanced method would not be able to improve this result significantly, as other stages in the algorithms are now the limiting factor.

4.2. Real-time simulation on fine meshes

To achieve interactivity in the simulation, we compute and store the velocity field for multiple time steps within one cardiac period and 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. A cardiac cycle is decomposed in 20–30 steps (one cardiac cycle lasts about 1 s). For each step, the solution is computed for a specific coil density. This process is repeated for different densities of coil in the aneurysm. This database of velocity fields

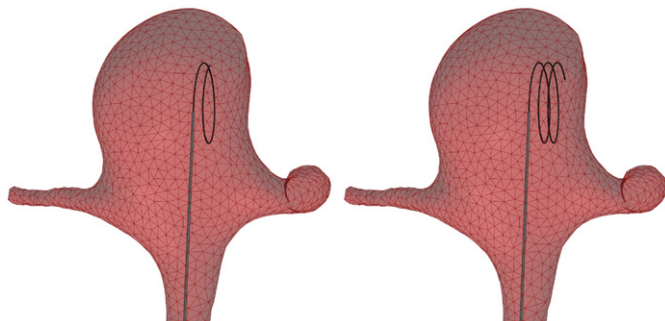


Fig. 9. Simulation of coil deployment without taking into account the influence of blood flow. Even a very small coil stays within the aneurysm as there is no flow to drag it out.

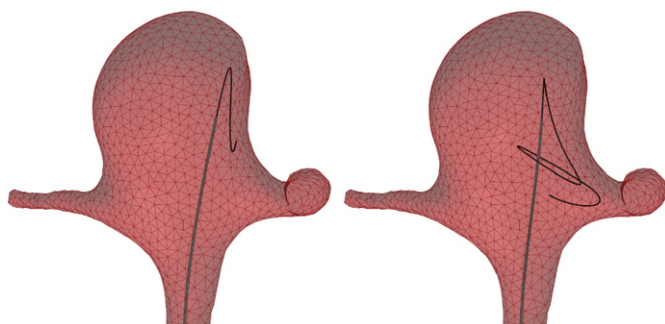


Fig. 10. Near real-time simulation of coil deployment while accounting for the influence of blood flow. The coil changes the flow distribution in the aneurysm while the blood flow influences the coil motion.

(typically 100 sets) can then be used to interpolate the blood velocity at the position of each 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.

Figs. 9 and 10 present a simulation of coil deployment respectively without and with the influence of the blood flow. While the simulation without flow is unrealistic, accounting for blood flow influence provides a simulation and overall behavior much closer to what takes place in an actual procedure (e.g. if the coil loop diameter is too small compared to the aneurysm size, it gets pushed out of the aneurysm by the flow). By combining patient-specific aneurysm geometry with accurate coil and flow models, such coupled simulation has the potential to become accurate enough to support medical planning applications.

On a Intel i7 3.33 GHz processor, one simulation time step of 5 ms required an average of 44 ms to compute with a tetrahedral mesh containing 21,576 elements. This allows to compute one cardiac cycle in less than 10 s, and the full database in only a few minutes. Comparing our computation times with FLUENT, which we used for assessing the accuracy, we noticed a $\times 2$ to $\times 3$ speed-up using DEC, depending on the mesh resolution. Combining this result with the fact the DEC method is less sensitive to the reduction in mesh density, it seems that this approach has some real benefits when computation time is an important constraint, without sacrificing accuracy.

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. 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. Moreover, an intuitive 3D visualization of flow data could also be interesting and helpful for the surgery [11].

Acknowledgments

We would like to thank Frederick Roy and Chunhong Pan for their valuable help in this research. We are also very grateful to Pierre Alliez and Stephane Tayeb from INRIA for their help using CGAL and for improving CGAL Mesh to answer some of our needs. 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] Alderliesten T. Simulation of minimally-invasive vascular interventions for training purposes. PhD dissertation, Utrecht University; 2004.
- [2] Bernsdorf J, Wang D. Non-newtonian blood flow simulation in cerebral aneurysms. *Computers and Mathematics with Applications* 2009;58(5):1024–9. Mesoscopic methods in engineering and science.
- [3] Carlson M, Mucha PJ, Turk G. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics* 2004;23:777–84.
- [4] Castro M, Putman C, Cebal J. Patient-specific computational fluid dynamics modeling of anterior communicating artery aneurysms: a study of the sensitivity of intra-aneurysmal flow patterns to flow conditions in the carotid arteries. *American Journal of Neuroradiology* 2006;27:2061–8.
- [5] Cebal J, Castro M, Appanaboyina S, Putman C, Millan D, Frangi A. Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: technique and sensitivity. *IEEE Transactions on Medical Imaging* 2005;24(4):457–67.
- [6] CGAL <<http://www.cgal.org/>>; 2010.
- [7] Dequidt J, Marchal M, Duriez C, Kerien E, Cotin S. Interactive simulation of embolization coils: modeling and experimental validation. In: *MICCAI*, vol. 1; 2008. p. 695–702.
- [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] Duriez C, Cotin S, Lenoir J, Neumann PF. New approaches to catheter navigation for interventional radiology simulation. *Computer Aided Surgery* 2006;11:300–8.
- [10] 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. p. 60–8.
- [11] Rocco Gasteiger, Mathias Neugebauer, Alexandra Baer, Christoph Kubisch, Bernhard Preim. Adapted surface visualization of cerebral aneurysms with embedded blood flow information. In: *Eurographics workshop on visual computing for biology and medicine*; 2010.
- [12] Gould D. Interventional radiology simulation: prepare for a virtual revolution in training. *Journal of Vascular Interventional Radiology* 2007;18(4):483–90.
- [13] Groden C, Laudan J, Gatchell S, Zeumer H. Three-dimensional pulsatile flow simulation before and after endovascular coil embolization of a terminal cerebral aneurysm. *Journal of Cerebral Blood Flow and Metabolism* 2001;21(12):1464–71.
- [14] Hernandez M, Frangi AF. Non-parametric geodesic active regions: Method and evaluation for cerebral aneurysms segmentation in 3dra and cta. *Medical Image Analysis* 2007;11(3):224–41. <http://www.sciencedirect.com/science/article/B6W6Y-4N4J341-1/2/6f9196c3d849902495a3ea4c6b11d1dd>.
- [15] Kakalis N, Mitsos AP, Byrne JV, Ventikos Y. The haemodynamics of endovascular aneurysm treatment: a computational modelling approach for estimating the influence of multiple coil deployment. *IEEE Transactions on Medical Imaging* 2008;27(6):814–24.
- [16] Liu C. Numerical solution of three-dimensional Navier–Stokes equations by a velocity–vorticity method. *International Journal for Numerical Methods in Fluids* 2001;35:533–57.
- [17] Marshall I, Papatheanasopoulou P, Wartolowska K. Carotid flow rates and flow division at the bifurcation in healthy volunteers. *Physiological Measurement* 2004;25(3):691.
- [18] McGregor R, Szczerba D, Muralidhar K, Székely G. A fast alternative to computational fluid dynamics for high quality imaging of blood flow. In: Yang G-Z, Hawkes D, Rueckert D, Noble A, Taylor C, editors. *Medical image computing and computer-assisted intervention-MICCAI 2009*, vol. 5761. Berlin/Heidelberg: Springer; 2009. p. 124–31.
- [19] Pascazio G, Napolitano M. A staggered-grid finite volume method for the vorticity–velocity equations. *Computers & Fluids* 1996;25(4):433–46.
- [20] Robinson-Mosher A, Shinar T, Gretarsson J, Su J, Fedkiw R. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics* 2008;27(3):1–9.
- [21] Schenk O, Gärtner K. On fast factorization pivoting methods for symmetric indefinite systems. *Electronic Transactions on Numerical Analysis* 2006;23:158–79.
- [22] Stam J. Stable fluids. In: *Computer graphics proceedings. SIGGRAPH*; 1999. p. 121–8.
- [23] Sun Q, Groth A, Bertram M, Waechter I, Bruijns T, Hermans R, et al. Phantom-based experimental validation of computational fluid dynamics simulations on cerebral aneurysms. *Medical Physics* 2010;37(9):5054–65.
- [24] Tournis J, Wormser C, Alliez P, Desbrun M. Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. In: *Proceedings of the 16th international meshing roundtable*, vol. 1B(3); 2008. p. 83–101.
- [25] Wu X, Luboz V, Krissian K, Cotin S, Dawson S. Segmentation and reconstruction of vascular structures for 3d real-time simulation. *Medical Image Analysis* 2011;15(1):22–34.