An Immersed Boundary Method for Detail-Preserving Soft Tissue Simulation from Medical Images

Christoph J. Paulus, Roland Maier, Daniel Peterseim, and Stéphane Cotin

Abstract Simulating the deformation of the human anatomy is a central element of Medical Image Computing and Computer-Assisted Interventions. Such simulations play a key role in nonrigid registration, augmented reality, and several other applications. Although the Finite Element Method is widely used as a numerical approach in this area, it is often hindered by the need for an optimal meshing of the domain of interest. The derivation of meshes from imaging modalities such as CT or MRI can be cumbersome and time-consuming. In this paper, we use the Immersed Boundary Method (IBM) to bridge the gap between these imaging modalities and the fast simulation of soft tissue deformation on complex shapes represented by a surface mesh directly retrieved from binary images. A high-resolution surface, which can be obtained from binary images using a marching cubes approach, is embedded into a hexahedral simulation grid. The details of the surface mesh are properly taken into account in the hexahedral mesh by adapting the Mirtich integration method. In addition to not requiring a dedicated meshing approach, our method results in higher accuracy for less degrees of freedom when compared to other element types. Examples on brain deformation demonstrate the potential of our method.

C. J. Paulus · S. Cotin (✉)
Inria Nancy Grand Est, Villers-lès-Nancy, France

Université de Strasbourg, ICube Lab, CNRS, Illkirch, France
e-mail: stephane.cotin@inria.fr

R. Maier
Inria Nancy Grand Est, Villers-lès-Nancy, France

University of Augsburg, Augsburg, Germany

D. Peterseim
University of Augsburg, Augsburg, Germany

1 Introduction

Computational models in the field of Medical Image Computing and Computer-Assisted Interventions play an increasingly important role, in areas such as nonrigid registration, augmented reality, or surgical training. In this context, the Finite Element Method (FEM) is often used as the reference numerical approach, and many works have addressed its computational efficiency and accuracy [3, 10, 15].

An alternative approach are mesh-free methods that can be applied likewise, but need a separate surface representation and hence dealing with boundary conditions can become cumbersome. Additionally, mesh-free methods use integration methods which need a volumetric grid and can be computationally expensive. Thus, mesh-free methods are less appealing for our purposes and this work is based on the FEM.

To construct a Finite Element mesh, the general procedure consists of the following steps: The preoperative image is segmented, then a surface mesh is built as the isosurface of the segmented image, and finally a volumetric mesh is constructed in the domain enclosed by the boundary surface. At this stage, the geometrical complexity of anatomical structures makes the generation of volume meshes from a given surface representation a very challenging task.

Volumetric meshing of the domain is almost always done with linear tetrahedral elements and remains a very active area of research [8, 13]. While simulations with tetrahedral elements may lead to numerical issues, such as volumetric locking [1], hexahedral elements suffer less from volumetric locking and yield an equivalent or higher accuracy per degree of freedom. However, hexahedral meshes can hardly be adapted to complex surfaces [7]. Thus, it is advantageous to either extend the partial differential equation (PDE) outside the actual domain in order to use a domain-independent mesh, which is known as the Finite Cell Method [14], or to work with partially filled elements. In the following, we will focus on the latter approach. The Composite FEM [6, 18, 19], for instance, simulates deformations with a coarse hexahedral grid efficiently, while a fine grid is used to modify the shape functions in such a way that they match the boundary conditions to a given accuracy. The fine grid is also utilized for the numerical integration. Tuning the resolution of the coarse grid increases the accuracy of deformations, while the resolution of the fine grid controls the geometric proximity at the beginning of the simulation. Other popular approaches also require two meshes. For example, the Cut Finite Element Method [2] operates on coarse elements but requires a sub-mesh in order to integrate basis functions over the actual domain. The method proposed in [9] is based on two overlapping Finite Element meshes, a coarse mesh that does not resolve any domain boundaries and the one that resolves the boundary. The method is then based on the coarse Finite Element functions enriched by the Finite Element functions of the resolving mesh. Boundary conditions are enforced in integral form with Nitsche's method. The key challenge is to integrate over the cut elements at the boundary and the boundary segments, where an efficient integration method proposed by Mirtich [11] is used. The additional Finite Element functions ensure the accuracy at the boundary but also increase the number of degrees of freedom.

In this paper, we present a Finite Element approach derived from the Immersed Boundary Method (IBM), which was first introduced by Peskin [16] to simulate blood flows: A potentially high-resolution surface mesh is immersed into a coarse simulation mesh. We solve partial differential equations with a regular hexahedral mesh that can automatically be built from the bounding box of a segmented image. To precompute the system matrices, such as the stiffness, the damping, and the mass matrix, our method uses the extension of an integration method dealing with arbitrary polygons [11] and thus with arbitrary geometrical shapes. The efficiency of this integration method yields a fast initialization of our algorithm while the speed depends upon the resolution of the objects surface.

Our approach is very similar to the one presented in [9]. Both methods use Nitsche's method to apply boundary conditions and Mirtich's integration method to integrate over boundary elements. The key advantage of our method, however, is that we do not need a second volumetric mesh to resolve the boundary and only use the basis functions of the coarse volumetric mesh. While this difference results in less accuracy at the boundary itself, the information gathered from the exact integration is enough to efficiently simulate the coarse behavior of an object. The fewer degrees of freedom are particularly appealing since we aim for dynamic simulations in real time. Other methods, such as [9], that resolve the boundary are less suitable for dynamic simulations because of the higher number of degrees of freedom which results in larger linear systems to be solved in every time step. The main difference compared to the Composite FEM or the Cut FEM lies in the fact that no additional refined mesh is needed to resolve complicated structures. Moreover, those methods are mainly based on distance functions to characterize the boundary and not on a surface mesh.

To further increase the speed of the dynamic simulation, we enrich the IBM with the corotational approach, preventing recomputations of the system matrices and thus allowing for real-time computation. The stability of our approach is ensured by the removal of elements that are only filled with a small portion of the complete cube. Evaluations reveal the improved accuracy per degree of freedom of our approach when comparing to the conventional methods, particularly for objects with complex geometries. Thus, our method has an improved efficiency which decreases computational costs. Since the meshing with different resolutions is not an issue, it can be adapted to the power of the device performing the computation. This underlines the importance for the simulation of surgical interventions.

Our paper is structured as follows: After a brief overview of the Immersed Boundary Method from which we derived our approach, Sect. 2 explains the core of the method. Then we present a numerical comparison against other techniques in Sect. 3.1. Finally, we apply our method on a complex brain geometry to simulate a brain shift (Sect. 3.2), and conclude in Sect. 4 discussing future extensions.

2 Method

This section explains an Immersed Boundary Method (IBM) [17, 20] for dynamic simulations, which has been adapted using an extension of an efficient integration method [11]. This approach allows embedding a discrete representation of complex, highly detailed surfaces into the hexahedral simulation mesh, which is potentially coarse and sparse. With the corotational approach, the method keeps its computational efficiency over time. Image processing methods can be used to provide the surface representation. Then our approach can transform this representation directly to a fast, accurate, patient-specific biomechanical model. Circumventing cumbersome and time-consuming volume meshing algorithms thus makes our approach a competitive tool for medical simulations.

Figure 1 summarizes our approach, separating the theory (Sect. 2.1) from the algorithmic details and the numerical integration method (Sect. 2.2).

2.1 From a Continuous Problem to a Discrete Formulation

In order to allow for reasonable computation times while providing accurate deformations, we consider the deformation of objects with initial geometry Ω on

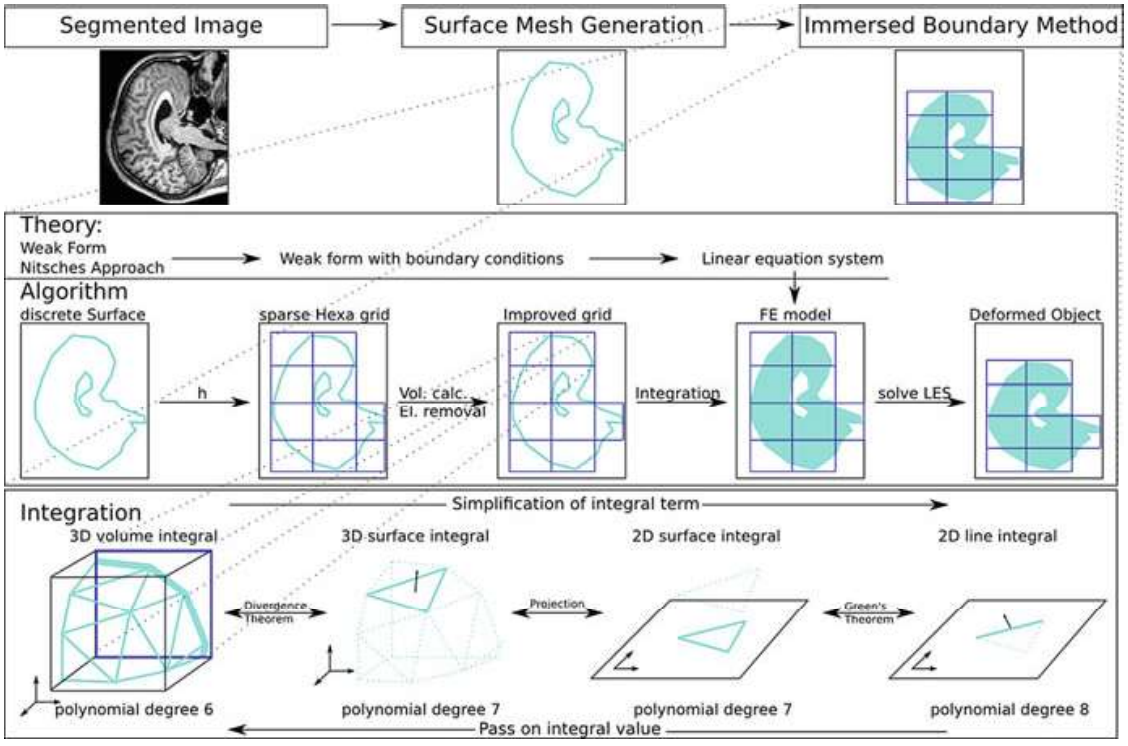

Fig. 1 Overview of our method: general work flow (*top*) and the particular elements of our method (*middle*) with focus on one of the main contributions: the integration

a macroscopic scale, which is commonly referred to as the continuum approach. We denote the displacement of a point $X \in \Omega$ to the point x by $u(X,\tau) = x(X,\tau) - X$ at a time $\tau \geq 0$. Using Cauchy's stress tensor σ, the gravity g, and density ρ of the considered material, one can state Cauchy's equation of motion for the dynamic scenario as $\operatorname{div}(\sigma(u)) + \rho g = \rho\ddot{u}$. Additionally, we define boundary conditions fixing the displacement at $u = u_0$ on $\Gamma_D \subseteq \partial\Omega$ and the boundary force or traction at $t(u) = \sigma(u)n = t_0$ on $\Gamma_N = \partial\Omega \backslash \Gamma_D$, where n is the outer normal of our domain Ω. The equation of motion is usually transformed to an integral equation, the so-called weak formulation. In contrary to the standard approach, the IBM incorporates the boundary terms, that is, as well the Dirichlet boundary condition, in this form. Violations of the conditions are penalized by additional integral terms and using the stabilization parameter γ_D. This is referred to as Nitsche approach [12]. Thus, we search a displacement function u that satisfies

$$\begin{aligned} &\int_\Omega \rho\ddot{u} \cdot v \, dV + \int_\Omega \sigma(u) : \varepsilon(v) \, dV - \int_\Omega \rho g \cdot v \, dV \\ &\quad - \int_{\Gamma_N} t_0 \cdot v \, dA - \int_{\Gamma_D} t(u) \cdot v \, dA - \int_{\Gamma_D} u \cdot t(v) \, dA \\ &\quad + \int_{\Gamma_D} u_0 \cdot t(v) \, dA - \gamma_D \int_{\Gamma_D} (u_0 - u) \cdot v \, dA = 0 \end{aligned} \tag{1}$$

for all test functions v and all times $\tau \geq 0$. The terms in the first row and the first term in the second row relate to Cauchy's equation of motion and the remaining terms are introduced as a result of the Nitsche approach. The linear elasticity theory defines the linear relationship between the Cauchy stress tensor σ and the strain tensor ε. Since the linearized Green-Lagrange strain is nonzero for rigid body motions such as rotations, we use the corotated strain tensor $\varepsilon(u) = S - I$ with the symmetric stretch matrix S, which is obtained from the deformation tensor.

In order to discretize the object, the surface $\partial\Omega$ is replaced by polygons of arbitrary shape representing a closed, manifold surface $\partial\Omega_h$ potentially of high resolution, that can be obtained by a marching cube algorithm or other image processing methods. Likewise, we replace the boundary domains Γ_D, Γ_N by their discrete counterpart $\Gamma_{D,h}, \Gamma_{N,h}$. Then we overlap the object Ω_h enclosed in $\partial\Omega_h$ with a regular grid of hexahedral elements Ω_e s.t. $\Omega_h \subset \bigcup_e \Omega_e$, and call the hexaeder corner nodes P_i. Note, that the standard Finite Element approach requires equality, that is, $\Omega_h = \bigcup_e \Omega_e$, which is a strong restriction for the choice of the hexahedral elements. We define the mesh size parameter as $h = \max_e(\max_{X_1,X_2\in\Omega_e} |X_1 - X_2|)$, which indicates the deformation accuracy of the hexahedral mesh and is independent of the resolution of the surface mesh. The element-wise trilinear shape functions φ_i of the conventional FEM allow to replace the displacement function by its discrete counterpart $u \approx \sum_i \varphi_i u_i$, with the displacements u_i at the node P_i, forming the vector of displacements $\mathbf{u}(\tau)$ that depends on time. With the Galerkin approach, the integral equation transforms to the semi-discrete equation

$$\begin{aligned}&\rho\mathbf{M}\ddot{\mathbf{u}}(\tau)+\mathbf{D}\dot{\mathbf{u}}(\tau)+\mathbf{K}\mathbf{u}(\tau)-\mathbf{B}_D^T\mathbf{u}(\tau)-\mathbf{B}_D\mathbf{u}(\tau)+\gamma_D\mathbf{M}_D\mathbf{u}(\tau)\\&\quad=\mathbf{M}_N\mathbf{t_0}-\mathbf{B}_D\mathbf{u_0}+\rho\mathbf{M}_\Omega\mathbf{g}+\gamma_D\mathbf{M}_D\mathbf{u_0}\end{aligned}\tag{2}$$

where the terms with the matrix $\mathbf{B}_D$ relate to the integral terms in (1) over the Dirichlet boundary Γ_D and are due to the method of Nitsche. The stiffness matrix $\mathbf{K}$ is constructed similarly to the conventional FEM, but integrals are computed over the actual domain $\Omega_h \neq \bigcup_e \Omega_e$. $\mathbf{K}$ relates to the second term in (1). Similarly, the mass matrices are adapted to the IBM, that is, $\mathbf{M}_{Y,ij} = \int_Y \varphi_i\varphi_j$ for $Y = \Gamma_{D,h}, \Gamma_{N,h}, \Omega_h$. Note that the term $\mathbf{D}\dot{\mathbf{u}}(\tau)$ in (2) is artificial and does not correspond to any term in (1). However, the introduction of the damping term is reasonable in the context of dynamic simulation, as damping effects occur in dynamic motions. We choose the damping term to be constant and obtain it using the mass and the stiffness matrix. Such a representation is called modal or Rayleigh damping and can be expressed as

$$\mathbf{D}=\alpha\mathbf{M}+\beta\mathbf{K}.$$

In the general case, an experimental verification of the parameters α and β is essential for real applications [21].

Solving this system of linear equations yields the displacements $u_i(\tau)$ for any time $\tau \geq 0$ of the object Ω at the points P_i, which are propagated to the surface $\partial\Omega$ using the trilinear shape functions and the initial barycentric coordinates of the surface in the hexahedral mesh. Since (2) is still a continuous formulation with respect to time, the next step to obtain a fully discrete model consists in applying a time-stepping scheme with step size $\Delta\tau$. Here, we use the implicit Euler scheme

$$\begin{aligned}\dot{\mathbf{u}}_{\mathbf{n+1}} &= \dot{\mathbf{u}}_{\mathbf{n}}+\Delta\tau\,\ddot{\mathbf{u}}_{\mathbf{n+1}},\\ \mathbf{u}_{\mathbf{n+1}} &= \mathbf{u}_{\mathbf{n}}+\Delta\tau\,\dot{\mathbf{u}}_{\mathbf{n+1}}.\end{aligned}\tag{3}$$

Inserting the scheme into (2) leads to a system of linear equations that have to be solved in every time step. For better readability, we only state the equation for the $(n+1)$th time step under the additional assumption $u_0 \equiv 0$.

$$\begin{aligned}&\left(\rho\mathbf{M}+\Delta\tau\mathbf{D}+\Delta\tau^2(\mathbf{K}-\mathbf{B}_D^T-\mathbf{B}_D+\gamma_D\mathbf{M}_D)\right)\ \dot{\mathbf{u}}_{\mathbf{n+1}}\\&\quad=\Delta\tau\mathbf{f_0}+\left(\rho\mathbf{M}-\Delta\tau\,(\mathbf{K}-\mathbf{B}_D^T-\mathbf{B}_D+\gamma_D\mathbf{M}_D)\right)\ \mathbf{u_n},\end{aligned}\tag{4}$$

where $\mathbf{f_0}=\rho\mathbf{M}\mathbf{g}+\mathbf{M}_N\mathbf{t_0}$. Finally, $\mathbf{u}_{\mathbf{n+1}}$ can be calculated using the equations in (3).

2.2 Numerical Considerations

The construction of the matrices in the system of linear equations (2) is one of the key challenges of the IBM. To integrate, the domain is split into the volumes of the hexahedral elements. For completely filled elements with $\Omega_e \cap \partial\Omega = \emptyset$, Gauss points commonly yield the integral value. For elements intersecting with the boundary, standard integration techniques do not work as the integration domain might have an arbitrary complexity: The integration domains are either the boundaries intersected with an element, that is, $\Gamma_{D,h} \cap \Omega_e$ or $\Gamma_{N,h} \cap \Omega_e$, or the intersection of the object with the hexahedral elements, that is, $\Omega_h \cap \Omega_e$. We integrate exactly over these domains, by using an integration approach depicted in Fig. 1(bottom), which has been proposed for polynomials up to degree two [11]. The integrands in the last subsection are polynomial functions where the sum of the polynomial degrees is maximally six, due to the multiplication $\varphi_i \varphi_j$ of the trilinear shape functions φ_i and φ_j to compute the mass matrices. In the following, we use the space of polynomials $\mathbb{P}_M(\mathbb{R}^n)$ that acts on $\mathbb{R}^n$, constructed with monomials, whose sum of degrees is smaller than or equals M. Thus the trilinear shape functions fulfill $\varphi_i \in \mathbb{P}_3(\mathbb{R}^3)$ and we need to calculate integrals for $\mathbb{P}_6(\mathbb{R}^3)$. For this, we extend [11] to arbitrary polynomial degrees: First, the divergence theorem transforms the volume integral to a sum of integrals of $\mathbb{P}_7(\mathbb{R}^3)$ over polygons in 3D. Then these integrals are projected onto the plane with the biggest surface and one needs to calculate $\mathbb{P}_7(\mathbb{R}^2)$ over polygons in 2D. Finally, these integrals are simplified using Greens' theorem to $\mathbb{P}_8(\mathbb{R}^2)$ over lines in 2D, which can be integrated analytically using binomial coefficients and the positions of the line start and end. For the boundary matrices, the first step is left out but the subsequent steps stay the same with the polynomial degree reduced by one.

Regarding the constitutive model, we use a corotational approach rather than a simple linear strain tensor. This also permits to expand the range of applications of the method. Identically to the conventional approach a rotation matrix $\mathbf{R}_n$ is calculated based on the deformed configuration $\mathbf{u}_n$ of each hexahedral element and incorporated into the system of equations by using $\mathbf{K}_n = \mathbf{R}_n \mathbf{K}_0 \mathbf{R}_n^T$, but it is not applied to the other matrices in (2). The rotation matrix in the undeformed configuration equals the identity matrix and is updated when the system of linear equations yields new positions at a time step. In contrary to hyperelastic approaches, the stiffness matrix does not have to be recalculated for every time step, but is updated by multiplying the rotation matrices from both sides. Note that partially filled elements can result in elevated computational costs when integrating for a recalculated matrix, while the calculation of the rotation matrix and the multiplication on the stiffness matrix from both sides are computationally cheap. Thus, the corotational approach is particularly interesting for contexts where the speed of calculation matters, for example, for simulations in real time. Moreover, the corotational approach combines nonlinear characteristics with the simplicity of the

stress-deformation relationship. However, despite our choices for the corotational approach, we want to emphasize that the IBM we propose is not limited to small strain problems.

Finally, we maintain the numerical stability by ignoring hexahedral elements which are only partially inside the domain. For that, we remove elements where the volume of the integration domain is under 5% of the volume of the hexahedral element. Ignoring these elements has nearly no impact on the end result, improves stability, and leads to a slightly smaller system of linear equations.

In summary, the construction of the system matrices and thus the choice of the integration method impact the initialization time of our algorithm, while the corotational approach reuses the system matrices in order to prevent such time-consuming procedures in every time step. To conclude, our method allows for

1. A fast initialization depending upon the number of edges in the high-resolution surface mesh
2. A fast and stable simulation that depends upon the size and the quality of the linear system of equations which is based on the coarse simulation grid

3 Results

Here, we present the results of an implementation of our approach into the open-source framework SOFA [4]: First, we perform a convergence analysis that compares our algorithm to the classical FEM in beam compression and bending scenarios. Then, we apply the method to simulate a brain shift, a deformation of the brain often observed during neurosurgery.

3.1 Cylinder Compression and Bending Scenarios

To numerically compare our method against the classical FEM, we simulate the compression and bending of beams under gravity while being fixed at one end. The classical bending test results in a deflection, which requires a geometrically nonlinear model of deformation. Thus this example shows one of the motivations to use the corotational approach presented in the previous section.

We consider different cross sections (see Fig. 2): a circle and a cross section that has convolutions that are similar to the surface of a brain, see Fig. 4 bottom. The beam cross sections of the compression example have a radius $r = 0.06\,\mathrm{m}$, which is expanded $l = 0.1\,\mathrm{m}$ in the length, while for the bending example we have $r = 0.02\,\mathrm{m}$ and $l = 0.2\,\mathrm{m}$. We choose a Young's modulus of $E = 3000\,\mathrm{Pa}$ and a Poisson's ratio of $\nu = 0.49$ for the compression example and $E = 1\,\mathrm{MPa}$ and $\nu = 0.4$ for the bending to represent deformations close to the application in a brain shift.

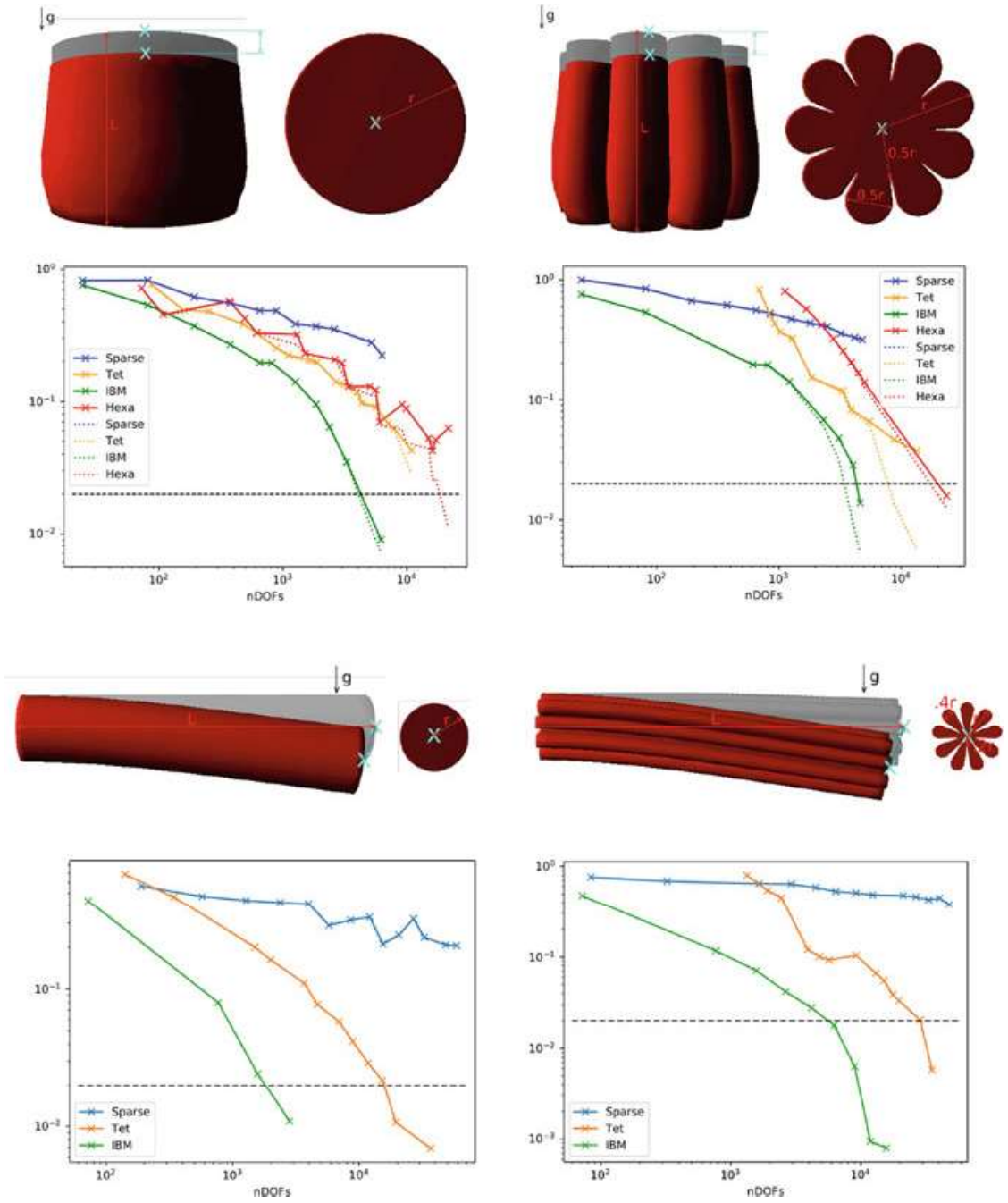

Fig. 2 Compression (*top*) and bending (*bottom*) examples with different cross sections (*left/right*): *Top:* Setup of the example in front (*left*) and top (*right*) views with the initial (*gray*) and final (*red*) configuration of the beam and the point at which we measure the distance in *turquoise*; *Bottom:* Convergence analysis, with the comparison of only the z values as dashed curves and the dashed black line depicts a relative error of 2%

We compare our method to the conventional FEM using tetrahedral elements, obtained using the open-source meshing library GMSH [5]. For tetrahedral elements with four nodes, incompressible behavior can yield stiff behavior, called volumetric locking [1]. Thus, we also compare to the conventional FEM using non-cuboid irregular hexahedral elements in the compression example. Since, to the best of

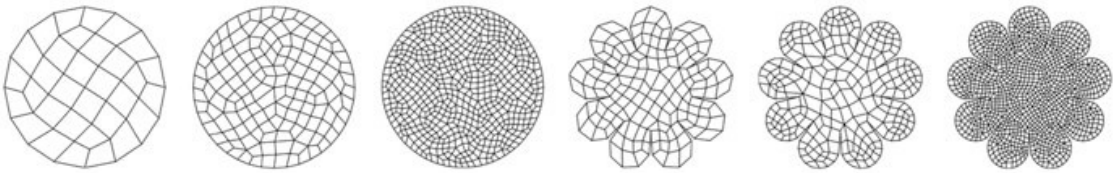

Fig. 3 Cross section of the first example meshed with quadrilaterals of different resolutions

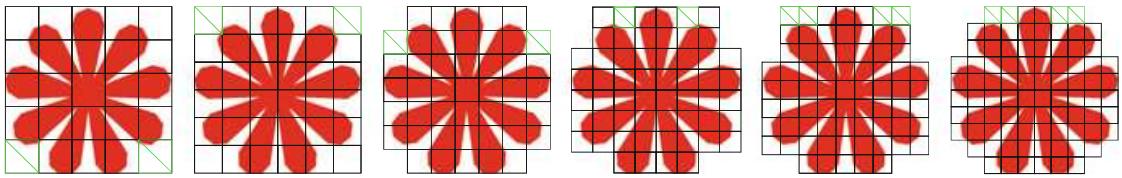

Fig. 4 Bending example: Several chosen resolutions of simulations with a regular hexahedral topology, showing the two-dimensional cross section with the grid: for the sparse grid, all displayed elements are simulated as completely filled, while for the IBM *green* elements are deleted (since they have less than 5% of the hexahedral volume) and boundary elements are partially filled

our knowledge, available meshing algorithms cannot provide hexahedral meshes, we first mesh a circle with quadrilaterals using GMSH (see Fig. 3) and then extrude them to hexahedral elements. To show the effect of regular partially filled elements, we compare as well to a sparse regular hexahedral grid: Similar to our idea, a surface is overlaid by a regular grid and elements outside of the topology are removed. Contrary to our approach, all elements are filled completely, even elements that are filled only to a small part (see Fig. 4).

The different results are compared by computing the relative error $d = |\bar{p}_{\text{ref},n} - \bar{p}_n|/|\bar{p}_{\text{ref},n} - \bar{p}_0|$, where $|\cdot|$ is the Euclidean norm, $\bar{p}_{\text{ref},n}$ is the reference beam tip in the middle of the beam, and $\bar{p}_n$ is the tip position for a given simulation after a few simulation steps n (we use subscript 0 for the initial position). The points are depicted as turquoise crosses in Fig. 2.

The compression example uses a reference solution composed of hexahedra (40,887 and 46,242 degrees of freedom), while the bending example uses a tetrahedral reference solution (592,899 and 514,047 degrees of freedom). The high resolution of the chosen reference solutions allows neglecting possible locking error. The results are summarized in a convergence analysis depicted in Fig. 2 and Table 1 compares the number of degrees of freedom for an accuracy of 2%. As expected, our approach yields a higher accuracy per degree of freedom than the existing approaches. For the complex geometry with the convolutions in the cross section, the effect is amplified in the second example. When comparing to tetrahedral meshes, our method is particularly interesting for a small number of degrees of freedom. The convergence analysis of the compression example shows for our approach that the error in the z direction depicted as a dashed line and the relative error are approximately the same. Thus, despite the potentially asymmetric (see Fig. 4) removal of elements for the immersed boundary method, the impact on the complete mesh stays reasonably low and the deformation is symmetric.

Table 1 Comparison of the number of degrees of freedom for an accuracy of 2%

Example	Degrees of freedom			Factor Dofs IBM vs.	
	Tetrahedra	Hexahedra	IBM	Tetrahedra	Hexahedra
Compression	–	–	4956	–	–
Compression with convolutions	–	22,892	4368	–	$\frac{22892}{4368} \approx 5.2$
Compression—only z direction	–	18,984	4809	–	$\frac{18984}{4809} \approx 3.9$
Compression with convolutions—only z direction	8363	22,326	3578	$\frac{8363}{3578} \approx 2.3$	$\frac{22326}{3578} \approx 6.2$
Bending	15,924	–	1969	$\frac{15924}{1969} \approx 8.1$	–
Bending with convolutions	28,871	–	5813	$\frac{28871}{5813} \approx 5$	–

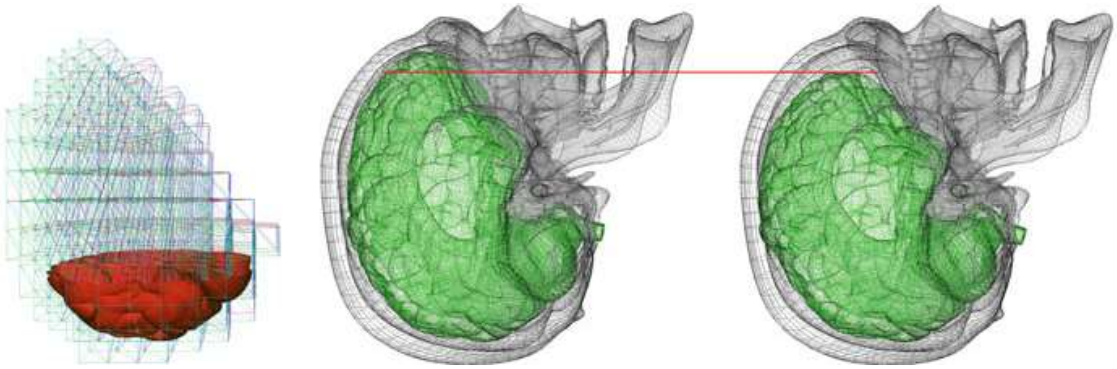

Fig. 5 Simulation of brain shift using a detailed surface mesh embedded into an hexahedral grid. Boundary conditions are applied onto the exact surface, not the grid *(left)*

3.2 Brain Shift Simulation

In order to assess our method in the medical context, it has been applied to simulate a brain deformation, frequently occurring after a craniotomy, see Fig. 5. The brain surface mesh is highly detailed, with 88,580 triangles and 44,261 points, and contains a lot of anatomical details due to the presence of sulci and gyri. The bounding box of the surface mesh is then simply subdivided into a regular grid with 9, 11, and 11 points in each principal direction, resulting in a sparse grid with 675 nodes P_i. Dirichlet boundary conditions are then applied to the actual surface (and not to the nodes of the grid) as explained in Sect. 2 and depicted in Fig. 5 (left). Other interactions such as collision between the scull and the brain are disregarded. The convolutions are in most cases inside one element or two neighboring elements that are connected, which allows to handle auto collisions without external algorithms, iff the elements should not invert themselves. Since Dirichlet boundary conditions in the standard FEM are applied to the nodes and not on a part of a face as in our approach, no comparisons to existing approaches were performed. Using a Young's modulus $E = 3000\,\text{Pa}$, a Poisson's ratio of 0.49, and a mass density of $1027\,\text{kg/m}^3$, the computation of the brain shift did not involve complex volumetric meshing and has been performed using the damping parameters $\alpha = \beta = 0.1$.

4 Conclusion and Perspectives

In this paper, we have introduced an adaptation of the Immersed Boundary Method, generally used for fluid dynamics problems, to the context of patient-specific simulation of soft tissue deformation. The benefit of our method over conventional Finite Element Methods lies in the ability to handle complex geometries while using a regular, relatively coarse, hexahedral unfitted mesh. In particular, no auxiliary mesh is required. The complexity of the nonstandard numerical integration over the cut elements remains proportional to the number of surface elements. The number of degrees of freedom and thus the size of the linear system in every time step are only proportional to the number of coarse elements and independent of the number of geometric features of the boundary. In contrast to the standard Finite Element approach, fixed displacement boundary conditions can be applied to the high-resolution surface mesh and are propagated to the potentially coarse hexahedral simulation mesh. These different advantages make the method a promising approach for building patient-specific coarse simulations in an automated way.

Yet, we believe this is only a first step toward a new way of constructing Finite Element simulations over complex domains defined in medical images. As the underlying integration can deal with holes in the surface, the approach has the built-in potential to compensate for incomplete surface reconstruction due to missing or wrong data. This could make the method even more robust and adapted to medical images. Furthermore, the fact that dynamic simulations are directly obtained from surface meshes segmented from medical images and do not require dedicated volumetric meshing techniques could make the method a very valuable and user-friendly tool in the context of medical simulation.

Acknowledgements Daniel Peterseim was supported by DFG-SPP 1748 under the project *Adaptive isogeometric modeling of discontinuities in complex-shaped heterogeneous solids*.

References

1. Benzley SE, Perry E, Merkley K, Clark B, Sjaardama G (1995) A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis. In: Proceedings, 4th international meshing roundtable, vol 17, pp 179–191
2. Burman E, Claus S, Hansbo P, Larson MG, Massing A (2015) Cutfem: discretizing geometry and partial differential equations. Int J Numer Methods Eng 104(7):472–501
3. Cotin S, Delingette H, Ayache N (1999) Real-time elastic deformations of soft tissues for surgery simulation. IEEE Trans Vis Comput Graph 5(1):62–73
4. Faure F, Duriez C, Delingette H, Allard J, Gilles B, Marchesseau S, Talbot H, Courtecuisse H, Bousquet G, Peterlik I et al (2012) Sofa: a multi-model framework for interactive physical simulation. In: Soft tissue biomechanical modeling for computer assisted surgery. Springer, pp 283–321
5. Geuzaine C, Remacle J-F (2009) GMSH: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. Int J Numer Methods Eng 79(11): 1309–1331

6. Hackbusch W, Sauter SA (1997) Composite finite elements for the approximation of PDEs on domains with complicated micro-structures. Numer Math 75(4):447–472
7. Ji S, Ford JC, Greenwald RM, Beckwith JG, Paulsen KD, Flashman LA, McAllister TW (2011) Automated subject-specific, hexahedral mesh generation via image registration. Finite Elem Anal Des 47(10):1178–1185
8. Liseikin VD (2009) Grid generation methods. Springer, Berlin
9. Massing A, Larson MG, Logg A (2013) Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions. SIAM J Sci Comput 35(1): C23–C47
10. Miller K (1999) Constitutive model of brain tissue suitable for finite element analysis of surgical procedures. J Biomech 32(5):531–537
11. Mirtich B (1996) Fast and accurate computation of polyhedral mass properties. J Graph Tools 1(2):31–50
12. Nitsche J (1971) Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, vol 36. Springer, Berlin, pp 9–15
13. Owen SJ (1998) A survey of unstructured mesh generation technology. In: International Meshing Roundtable, pp 239–267
14. Parvizian J, Düster A, Rank E (2007) Finite cell method. Comput Mech 41(1):121–133
15. Paulus CJ, Haouchine N, Kong S-H, Soares RV, Cazier D, Cotin S (2016) Handling topological changes during elastic registration: application to augmented reality in laparoscopic surgery. Int J Comput Assist Radiol Surg 12(3):461–470
16. Peskin CS (1972) Flow patterns around heart valves: a numerical method. J Comput Phys 10(2):252–271
17. Peskin CS (2002) The immersed boundary method. Acta Numer 11:479–517
18. Peterseim D, Sauter SA (2008) The composite mini element - coarse mesh computation of stokes flows on complicated domains. SIAM J Numer Anal 46(6):3181–3206
19. Rech M, Sauter S, Smolianski A (2006) Two-scale composite finite element method for Dirichlet problems on complicated domains. Numer Math 102(4):681–708
20. Rüberg T, Cirak F, García Aznar JM (2016) An unstructured immersed finite element method for nonlinear solid mechanics. AMSES 3(1):22
21. Wriggers P (2008) Nonlinear finite element methods. Springer, New York