Advanced Modeling and Simulation in Engineering Sciences

RESEARCH ARTICLE   **Open Access**

Contact modeling from images using cut finite element solvers

S. Claus[1*] , P. Kerfriden[2,3], F. Moshfeghifar[4], S. Darkner[4], K. Erleben[4] and C. Wong[5]

*Correspondence:
susanne.claus@onera.fr
[1]Onera, 8 Chemin de la Hunière,
91120 Palaiseau, France
Full list of author information is
available at the end of the article

Abstract

This paper presents a robust digital pipeline from CT images to the simulation of contact between multiple bodies. The proposed strategy relies on a recently developed immersed finite element algorithm that is capable of simulating unilateral contact between solids without meshing (Claus and Kerfriden in Int J Numer Methods Eng 113(6):938–966, 2018). It was shown that such an approach reduces the difficulties associated with the digital flow of information from analytically defined geometries to mechanical simulations. We now propose to extend our approach to include geometries, which are not defined mathematically but instead are obtained from images, and encoded in 3D arrays of voxels. This paper introduces two novel elements. Firstly, we reformulate our contact algorithm into an extension of an augmented Lagrangian CutFEM algorithm. Secondly, we develop an efficient algorithm to convert the surface data generated by standard segmentation tools used in medical imaging into level-set functions. These two elements give rise to a robust digital pipeline with minimum user intervention. We demonstrate the capabilities of our algorithm on a hip joint geometry with contact between the femur and the hip bone.

Keywords: Contact, Nonlinear algorithm, LaTIn, Nonconforming finite element, CutFEM, Ghost penalty

Introduction

In this article, we present a modern digital pipeline for the simulation of the mechanical interaction between solid bodies from CT images. The generation of finite element models from medical or engineering images is a very challenging task. The first step is to identify the set of voxels corresponding to the different solid bodies using image segmentation and/or registration. The regional information of the bodies is then typically transformed into a surface representation of each body, e.g. surface triangulations. From the simulation point of view, the challenge is that this surface mesh representation is of low quality and frequently contains badly shaped triangles which are not suitable for finite element analysis. The importance of high quality meshes in biomedical simulations is highlighted in e.g. [14,15]. To obtain accurate and stable solutions, the surface mesh

Claus *et al. Adv. Model. and Simul. in Eng. Sci.* (2021)8:13

Page 2 of 23

needs to be re-meshed, and this boundary operation is followed by interior volume meshing. Unfortunately, meshing is error-prone and often requires time consuming manual interventions and corrections, which is highly cumbersome, and requires valuable experts' time. To address this issue, several approaches are followed in the literature. A first class of mesh mapping methods [33] attempt to automatise mesh generation by deforming reference meshes to fit to specific geometries. Of course, such approaches still require prior meshing work and are limited to relatively small variations around the reference geometry. A second class of approaches aim to eliminate or significantly reduce the reliance on meshing. The most popular instance of these methods is the voxel or image-based finite element method [20,26,31]. In voxel based finite element methods, voxel of the image are directly converted to a hexahedral finite element mesh. Grey-scale values of the voxels are then mapped onto material values such as elasticity moduli. These methods reduce the meshing burden considerably, because the computational grid does not need to be aligned with any object boundary. However body boundaries are represented by "staircases". This causes stress singularities in the finite element simulation, which negatively affects the quality of the results near boundaries. Unfortunately, boundaries are often of key importance to analysts. This is the case in the present contribution, which focusses on the simulation of contact between two elastic bodies. In addition to this drawback of voxel-based methods, one may also find that, depending on the resolution of the image, two bodies may be indistinguishable and merged together into one big block.

In this article, we introduce an alternative approach related to X-FEM [4,19,28,30] and CutFEM [6,22].

We start from surface triangulations generated by a segmentation tool [1]. This surface mesh, which may be of poor quality, is then embedded in a regular and fixed background grid. We now generate signed distance functions for each body, using the regular background grid to represent them discretely. As per usual with level-set-based methods, the surfaces of the bodies will be represented by the zero contour line of these functions. Zero contour lines may intersect the background mesh in an arbitrary manner. This allows for a sharp and smooth (continuous and elements-wise linear in this paper) surface representation which are then used simulate contact and compute contact forces between the elastic bodies.

The background mesh which is used to discretize the unknowns and the level set functions needs to be sufficiently fine in order for the piecewise linear level set function to capture the geometry accurately. For this purpose, we develop and analyse two refinement criteria for adaptive mesh refinement.

The contact problem between elastic bodies will be solved using the CutFEM-LaTIn algorithm recently introduced in [12]. In that paper, the versatility of the algorithm was demonstrated, and it was shown that it converges optimally. In the present article, we formally show that the CutFEM-LaTIn algorithm is an extension of the augmented Lagrangian CutFEM algorithm presented in [7,8]. The CutFEM-LaTIn method uses ghost-penalty stabilization [5] to ensure well-conditioned system matrices. Alternatives to the ghost-penalty method may be employed, e.g. function extension [17,23]. The algorithm is implemented in FEniCS [2].

The aforementioned elements, an efficient surface triangulation to level-set geometrical algorithm, adaptive refinement criteria and a state-of-the art contact simulation algorithm

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13

Page 3 of 23

for contact over implicitly defined geometries, give rise to a stable and robust digital pipeline from CT images to finite element simulations.

This article is organised as follows. In "Frictionless unilateral contact between two deformable solids" section, we detail the CutFEM-LaTIn algorithm used to compute the contact forces between two elastic bodies. In "Digital pipeline from CT-scans to CutFEM biomechanical simulations" section, we describe our algorithm to convert CT-scans to cut finite element biomechanical simulations for multiple bones including two adaptive refinement criteria. In "Unilateral contact between hip and femur bone" section, we show numerical results for contact between a femur and a hip bone.

Frictionless unilateral contact between two deformable solids

In this section, we will describe our contact algorithm between two elastic bodies which occupy the domain Ω^1 and Ω^2, respectively. The domains will be described implicitly using two level set functions. We begin our problem definition in the continuous setting in this subsection, before we describe its discretization of the domains and the definition of the discretization of the unknowns in the next subsection.

We assume that each elastic body is only undergoing small deformations and that they interact through frictionless unilateral contact. A contact problem between more than two bodies is possible in our framework and a contact algorithm for a larger number of solids has been introduced in [12]. However, in this contribution, we restrict ourselves to the contact problem between two bodies (see Fig. 1). Let Ω^b denote our total domain which contains our two elastic bodies. Let $\phi^1 : \Omega^b \to \mathbb{R}$ and $\phi^2 : \Omega^b \to \mathbb{R}$ be the two level set functions that describe the geometry of body 1 and the geometry of body 2, respectively. We then define two domains Ω^1 and Ω^2 by

$$\Omega^1 := \{x \in \Omega^b \mid \phi^1(x) \leq 0 \text{ and } \phi^2(x) > 0\},$$
$$\Omega^2 := \{x \in \Omega^b \mid \phi^2(x) \leq 0\}. \tag{1}$$

The contact interface Γ^c between the two solids, which is assumed to be of non-zero measure, is defined as

$$\Gamma^c = \{x \in \Omega^b \mid \phi^2(x) = 0 \text{ and } \phi^1(x) \leq 0\}. \tag{2}$$

Furthermore, let $\partial\Omega^i$ denote the boundary of Ω^i, $i = 1, 2$ and $\partial\Omega := (\partial\Omega^1 \cup \partial\Omega^2)\backslash\Gamma^c$ the boundary of domain Ω^1 and Ω^2 without Γ^c. We further partition the boundary $\partial\Omega$ into a Neumann part $\partial\Omega^t$ and a Dirichlet part $\partial\Omega^u$ with $\partial\Omega^t \cap \partial\Omega^u = \emptyset$.

The problem of unilateral contact can be formulated as an optimisation problem under inequality constraints. We look for displacement field $\underline{u} := \{\underline{u}^1, \underline{u}^2\} : \Omega^1 \times \Omega^2 \to \mathbb{R}^d \times \mathbb{R}^d$ that minimises the potential energy

$$\mathcal{J}(\underline{u}) := \frac{1}{2}a(\underline{u}, \underline{u}) - l(\underline{u}) \tag{3}$$
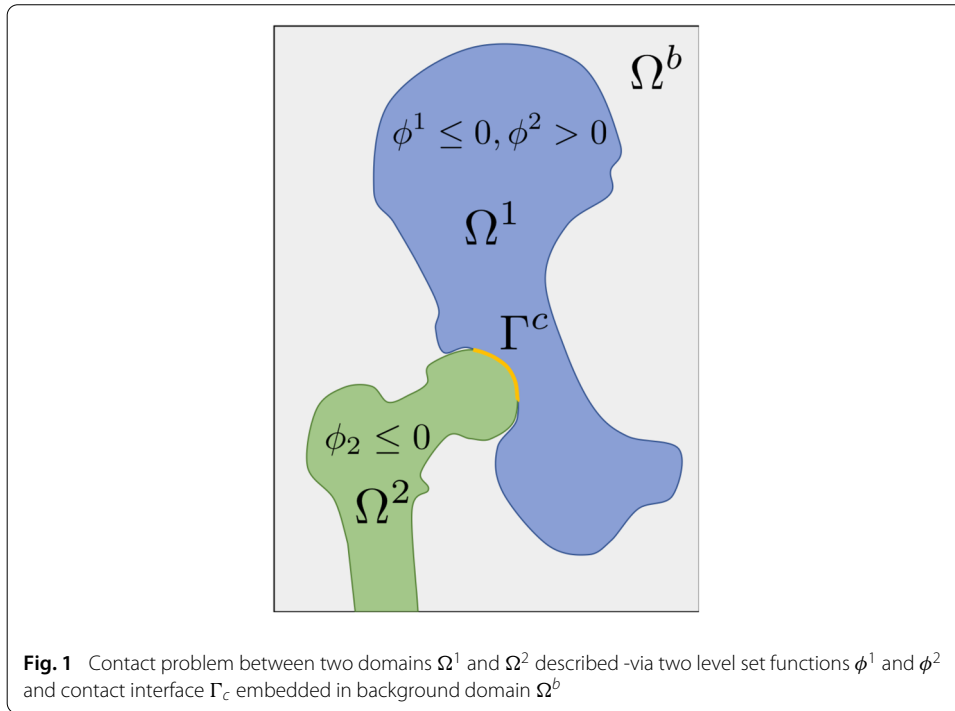
under inequality constraints

$$[\![\underline{u}]\!] \cdot \underline{n}_\Gamma \leq 0 \qquad \text{on } \Gamma^c \tag{4}$$

and Dirichlet boundary conditions

$$\underline{u} = \underline{u}_d \qquad \text{on } \partial\Omega^u. \tag{5}$$

Here, $d = \{2, 3\}$ is the spatial dimension, $\underline{n}_\Gamma$ denotes the normal on Γ^c pointing from Ω^1 to Ω^2, u_d is a known prescribed displacement and $[\![\underline{u}]\!] = \underline{u}^1 - \underline{u}^2$ denotes the jump of $\underline{u}$ across

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13

Page 4 of 23

Fig. 1 Contact problem between two domains Ω^1 and Ω^2 described -via two level set functions ϕ^1 and ϕ^2 and contact interface Γ_c embedded in background domain Ω^b

the contact interface Γ_c. The inequality constraint (4) enforces a zero interpenetration between the domains Ω^1 and Ω^2. In the previous expression, the bilinear form a is defined as

$$a(\underline{u}, \delta\underline{u}) = a^1(\underline{u}^1, \delta\underline{u}^1) + a^2(\underline{u}^2, \delta\underline{u}^2) = \sum_{i=1}^{2} \int_{\Omega^i} \underline{\underline{\sigma}}(\underline{u}^i) : \underline{\underline{\epsilon}}(\delta\underline{u}^i) \, d\Omega \tag{6}$$

where $\underline{\underline{\sigma}}(\underline{u}^i) = \lambda^i \operatorname{tr}(\underline{\underline{\epsilon}}(\underline{u}^i))\boldsymbol{I} + 2\mu_i \underline{\underline{\epsilon}}(\underline{u}^i)$ is the Cauchy-stress tensor, $\underline{\underline{\epsilon}}(\underline{u}^i) = \frac{1}{2}\left(\nabla\underline{u}^i + (\nabla\underline{u}^i)^T\right)$ is the strain tensor, λ^i and μ^i are the two Lamé coefficients, i.e. $\lambda^i = \frac{E^i \nu^i}{(1+\nu^i)(1-2\nu^i)}, \mu^i = \frac{E^i}{2(1+\nu^i)}$, with Young's modulus E^i and Poisson's ratio ν^i, $i = 1, 2$. The linear form l is defined as

$$l(\delta\underline{u}) = l^1(\delta\underline{u}^1) + l^2(\delta\underline{u}^2) = \sum_{i=1}^{2} \int_{\Omega^i} \underline{f} \cdot \delta\underline{u}^i \, d\Omega + \sum_{i=1}^{2} \int_{\partial\Omega^t \cap \partial\Omega^i} \underline{t}_d \cdot \delta\underline{u}^i \, d\Gamma \tag{7}$$

In the following, we set the Poisson's ratios to $\nu^1 = \nu^2 = 0.3$. Here, $\underline{f}$ is a given body force and $\underline{t}_d$ is a given boundary traction.

The primal constrained optimisation problem (3)–(5) can be reformulated as an unconstrained primal/dual problem, which will be the basis for the development of this paper. In this setting, the solution to the contact problem is found by extremising the Lagrangian

$$\mathcal{L}(\underline{u}, \lambda) := \mathcal{J}(\underline{u}) + b(\lambda, \underline{u}) \tag{8}$$

with respect to primal variable $\underline{u}$, in the space of fields satisfying the Dirichlet boundary conditions, but not the unilateral contact condition, which is now relaxed and enforced by

$$b(\lambda, \underline{u}) = \int_{\Gamma^c} \lambda \left(\llbracket \underline{u} \rrbracket \cdot \underline{n}_\Gamma\right) d\Gamma. \tag{9}$$

Claus *et al. Adv. Model. and Simul. in Eng. Sci.* (2021) 8:13

Page 5 of 23

We reformulate the primal dual formulation in terms of the Karush–Kuhn–Tucker conditions, which consist of the linear variational form

$$a(u, \delta\underline{u}) + b(\lambda, \delta\underline{u}) = l(\delta\underline{u}) \tag{10}$$

for all $\delta\underline{u}$ satisfying $\delta u = 0$ over $\partial\Omega_u$ and the nonlinear constraint on the contact interface Γ^c

$$[\![\underline{u}]\!] \cdot \underline{n}_\Gamma \le 0$$
$$\lambda \ge 0, \tag{11}$$
$$(\underline{u} \cdot \underline{n}_\Gamma)\,\lambda = 0,$$

which can be reformulated as

$$\int_{\Gamma^c} \left(\lambda - \gamma \left[[\![\underline{u}]\!] \cdot n_\Gamma + \frac{1}{\gamma}\lambda \right]_+ \right) \delta\lambda \, d\Gamma = 0 \tag{12}$$

for all $\delta\lambda \in L^2(\Gamma^c)$. Here, γ is a strictly positive scalar and $[x]_+$ denotes the positive part of a scalar quantity $x \in \mathbb{R}$, defined as

$$[x]_+ = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

For a proof of the equivalence of condition (11) and (12) see for example [10,11].

CutFEM primal/dual solver for the contact problem

In this section, we will describe the discretization of the contact problem, which uses level set descriptions of the geometries in a fixed and regular background grid. The boundary of the elastic bodies will be represented by the zero contour lines of level set functions, which will be allowed to intersect the background mesh cells in an arbitrary manner. Each domain Ω^i will be embedded in their own fictitious domain resulting in overlapping fictitious domains along the contact interface. Each $\underline{u}^i$ will obtain its own set of degrees of freedom in its fictitious domain allowing for a multiple valued displacement $\underline{u} = (\underline{u}^1, \underline{u}^2)$ along the interface. This enables jumps in the displacement along the interface.

CutFEM/level-set approximation of the geometry

Let us introduce a triangulation $\mathcal{T}_h$ of the background domain Ω^b. We assume a simple shape for Ω^b, which is easy to mesh with a regular background grid, e.g. a bounding box which contains both elastic bodies. Furthermore, let us introduce the finite element space of continuous piecewise linear functions, i.e.

$$\mathcal{Q}_h := \{\phi \in \mathcal{C}^0(\Omega^b) : \phi|_K \in \mathcal{P}^1(K) \, \forall K \in \mathcal{T}_h\}. \tag{14}$$

We now define a piecewise linear approximation $\phi_h^1 \in \mathcal{Q}_h$ of level set ϕ^1 and $\phi_h^2 \in \mathcal{Q}_h$ of level set ϕ^2, such that for any $i \in \{1, 2\}$, the nodal values of ϕ_h^i coincides with the values of ϕ^i at the locations of the nodes. We now define the approximate physical domains Ω_h^1 and Ω_h^2 as follows:

$$\Omega_h^1 = \{x \in \Omega^b \,|\, \phi_h^1(x) \le 0 \text{ and } \phi_h^2(x) \ge 0\}$$
$$\Omega_h^2 = \{x \in \Omega^b \,|\, \phi_h^2(x) \le 0\} \tag{15}$$

and the piecewise linear approximation of the interface

$$\Gamma_h^c = \{x \in \Omega^b \,|\, \phi_h^1(x) = 0 \text{ and } \phi_h^2(x) \le 0\}. \tag{16}$$

Claus *et al. Adv. Model. and Simul. in Eng. Sci.* (2021) 8:13

Page 6 of 23

Overlapping domain decomposition

Now, we define the set of all elements of $\mathcal{T}_h$ that have a non-zero intersection with Ω_h^i, $i = 1, 2$

$$\hat{\mathcal{T}}_h^i := \{K \in \mathcal{T}_h : K \cap \Omega_h^i \neq \emptyset\}. \tag{17}$$

The domain corresponding to this set, denoted by $\hat{\Omega}_h^i := \bigcup_{K \in \hat{\mathcal{T}}_h^i} K$, is called *fictitious domain*. Furthermore, let us denote all elements which are intersected with Γ_h^c by

$$\hat{\mathcal{G}}_h := \{K \in \mathcal{T}_h \mid K \cap \Gamma_h^c \neq \emptyset\} \tag{18}$$

The domain corresponding to this set is denoted by $\hat{\Gamma}^h := \bigcup_{K \in \hat{\mathcal{G}}_h} K$. This definition will be used to extend interface quantities to a band of elements.

Of key importance to CutFEM methods is the careful tailoring of regularization terms to ensure that the system matrix remains well-conditioned. To this end, we define the set of ghost-penalty element edges for fictitious domain $\hat{\Omega}_h^i$, $i \in \{1, 2\}$ as

$$\hat{\mathcal{F}}_G^i := \{F = K \cap K' : K \in \hat{\mathcal{T}}_h^i \text{ and } K' \in \hat{\mathcal{T}}_h^i, F \cap \hat{\Gamma}^h \neq \emptyset\}. \tag{19}$$

For stabilization purposes still, we further define the set of intersected interface edges

$$\hat{\mathcal{F}}_I := \{F = K \cap K' : K \in \hat{\mathcal{G}}_h \text{ and } K' \in \hat{\mathcal{G}}_h\}. \tag{20}$$

Note our consistent use of notation $\hat{\ }$ to indicate extended quantities.

Fictitious domain and extended interface finite element spaces

We seek the displacement approximation $\underline{\hat{u}}_h = (\underline{\hat{u}}_h^1, \underline{\hat{u}}_h^2)$ of the two-body contact problem in the product space

$$\begin{aligned} \hat{\mathcal{U}}_h &= \hat{\mathcal{U}}_h^1 \times \hat{\mathcal{U}}_h^2, \\ \hat{\mathcal{U}}_h^i &:= \{\underline{\hat{u}} \in \mathcal{C}^0(\hat{\Omega}_h^i) : \underline{\hat{u}}|_K \in \mathcal{P}^1(K) \, \forall K \in \hat{\mathcal{T}}_h^i\}, \quad i = 1, 2. \end{aligned} \tag{21}$$

Note that $\underline{\hat{u}}_h$ is multi-valued. This feature allows for the representation of embedded discontinuities in CutFEM. In all elements that are intersected by the contact interface Γ_h^c, $\underline{\hat{u}}_h^1$ and $\underline{\hat{u}}_h^2$ may take different values at a single point, which results in a non-zero displacement jump between $\hat{u}_h^2$ and $\hat{u}_h^1$.

Next, we define the space of extended interface fields of polynomial order 1 as

$$\hat{\mathcal{W}}_h := \{\hat{W} \in \mathcal{C}^0(\hat{\mathcal{G}}_h) : \hat{W}|_K \in \mathcal{P}^1(K) \, \forall K \in \hat{\mathcal{G}}_h\}. \tag{22}$$

This is another important aspect of the CutFEM approach. Fields on manifolds such as embedded interfaces are represented as the trace of a finite element field defined in $\mathbb{R}^d$. In this paper, this feature will be used to represent the Lagrange multipliers required to enforce contact conditions in a primal/dual setting.

Primal/dual CutFEM solver

We now describe our P1/P1 LaTIn/CutFEM primal/dual algorithms, first introduced in [12]. We seek the displacement field in the linear CutFEM space $\hat{\mathcal{U}}_h$ and the Lagrange multipliers, which correspond to the normal reaction forces from contact, in the spaces of linear interface fields $\hat{\mathcal{W}}_h$. The presentation formalism that we have chosen for this paper allows us to highlight its relationship with the work presented in [7, 8] in the specific case of frictionless unilateral contact (our algorithm extends straightforwardly to more general interface conditions such as frictional contact or elastic damageable cohesive interfaces [25]).

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13

Page 7 of 23

Regularization of the bulk displacement

Let us define the extended and stabilized bilinear form over the product of fictitious domain fields as

$$\hat{a}_h(\underline{\hat{u}}_h, \delta\underline{\hat{u}}_h) = \sum_{i=1}^{2} \hat{a}_h^i(\underline{\hat{u}}_h, \delta\underline{\hat{u}}_h) + \sum_{i=1}^{2} \hat{s}_h^i(\underline{\hat{u}}_h, \delta\underline{u}_h) \tag{23}$$

where for $i \in \{1, 2\}$

$$\hat{a}_h^i(\underline{\hat{u}}_h, \delta\underline{\hat{u}}_h) = \int_{\Omega_h^i} \underline{\underline{\sigma}}(\underline{u}_h^i) : \underline{\underline{\epsilon}}(\delta\underline{u}_h^i) \, d\Omega \tag{24}$$

and the so-called ghost-penalty terms are defined by

$$\hat{s}_h^i(\underline{\hat{u}}_h, \delta\underline{\hat{u}}_h) = \sum_{F \in \hat{\mathcal{F}}_G^i} \int_F \frac{\alpha\, h}{E^i} \, [\![\underline{\underline{\sigma}}(\underline{u}_h^i) \cdot \underline{n}_F]\!] \cdot [\![\underline{\underline{\sigma}}(\delta\underline{u}_h^i) \cdot \underline{n}_F]\!] \, d\Gamma. \tag{25}$$

Here, $[\![x \cdot \underline{n}_F]\!]$ denotes the normal jump of the quantity x over the face, F, defined as $[\![x \cdot \underline{n}_F = x|_K\, \underline{n}_F - x|_{K'}\, \underline{n}_F]\!]$, where $\underline{n}_F$ denotes a unit normal to the facet F with fixed but arbitrary orientation and $\alpha > 0$ is the ghost penalty parameter.

Regularization of the Lagrange multiplier field

The discretized conditions for optimality are the following: Find $(\hat{u}_h, \hat{\lambda}_h) \in \hat{\mathcal{U}}_h \times \hat{\mathcal{W}}_h$ satisfying the Dirichlet boundary conditions such that $\forall (\delta\hat{u}_h, \delta\hat{\lambda}_h) \in \hat{\mathcal{U}}_h \times \hat{\mathcal{W}}_h$, where $\delta\hat{u}_h$ satisfies the homogeneous boundary conditions (and there is no positivity requirement for the Lagrange multipliers),

$$\left\{ \begin{aligned} & \hat{a}_h(\underline{\hat{u}}_h, \delta\underline{\hat{u}}_h) + b_h(\hat{\lambda}_h, \delta\underline{\hat{u}}_h) + \epsilon_\heartsuit\, s_\heartsuit(\hat{\lambda}_h, [\![\delta\underline{\hat{u}}_h]\!] \cdot n_{\Gamma_h}) = l(\delta\hat{u}_h) \\ & \left(\delta\hat{\lambda}_h, \hat{\lambda}_h - \gamma \left[[\![\hat{u}_h]\!] \cdot \underline{n}_{\Gamma_h} + \frac{1}{\gamma}\hat{\lambda}_h \right]_+ \right)_{\Gamma_h^c} + s_\heartsuit(\delta\hat{\lambda}_h, \hat{\lambda}_h) = 0 \end{aligned} \right. \tag{26}$$

where $s_\heartsuit$ is a regularization term that penalizes the jump of Lagrange multipliers across interface element edges, as introduced in [8]:

$$s_\heartsuit(\delta\hat{\lambda}_h, \hat{\lambda}_h) = \beta h^3 \sum_{F \in \hat{F}_I} \int_F [\![\nabla\delta\hat{\lambda}_h \cdot \underline{n}_F]\!] \cdot [\![\nabla\hat{\lambda}_h \cdot \underline{n}_F]\!] \, d\Gamma. \tag{27}$$

Here, $(u, v)_{\Gamma_h^c} = \int_{\Gamma_h^c} uv \, d\Gamma$. And $\epsilon_\heartsuit$ is either 0 or 1. The toggle boolean $\epsilon_\heartsuit$ is introduced for comparison to other references. When $\epsilon_\heartsuit = 1$, the previous problem is equivalent to the extremization of the saddle-type functional

$$\mathcal{M}(\underline{\hat{u}}_h, \hat{\lambda}_h) := \frac{1}{2}\hat{a}_h(\underline{\hat{u}}_h, \underline{\hat{u}}_h) + \frac{\gamma}{2}\left(\left[[\![\hat{u}_h]\!] \cdot \underline{n}_{\Gamma_h} + \frac{1}{\gamma}\hat{\lambda}_h \right]_+, \left[[\![\hat{u}_h]\!] \cdot \underline{n}_{\Gamma_h} + \frac{1}{\gamma}\hat{\lambda}_h \right]_+ \right)_{\Gamma_h^c}$$
$$- \frac{1}{2\gamma}b_h\left(\hat{\lambda}_h, \hat{\lambda}_h \right) - l(\underline{\hat{u}}_h) - s_\heartsuit(\hat{\lambda}_h, \hat{\lambda}_h) \tag{28}$$

with respect to both of its arguments. However, in this contribution, we choose $\epsilon_\heartsuit = 0$. This is to make the present formulation consistent with the developments proposed in [12], were the CutFEM-LaTIn algorithm was introduced in a completely different manner. We do not claim that this choice is better than the alternative.

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13

Page 8 of 23

Domain decomposition formulation

Whilst the previous "jump" formulation (26) couples the CutFEM subdomains, the LaTIn algorithm introduced in [12] implements a nonlinear domain decomposition approach. In this LaTIn-CutFEM algorithm the bulk problem is first solved in each subdomain independently and then interface fluxes and displacements are computed at the contact interface Γ_h^c. We iterate between the bulk problems (linear stage) for $i = 1, 2$ and the interface problem (local stage) until we achieve convergence. In the following, we briefly outline the main steps of the algorithm, a more detailed description can be found in [12].

To reformulate the "jump" formulation (26) into a problem over each fictitious domain $\hat{\Omega}_h^i$, we realize that (26) can be decomposed into

$$\begin{cases} \sum_{i=1}^{2} \hat{a}_h^i(\underline{\hat{u}}_h^i, \delta\underline{\hat{u}}_h^i) + b_h(\hat{\lambda}_h, \delta\underline{\hat{u}}_h^i) = \sum_{i=1}^{2} l^i(\delta\underline{\hat{u}}_h^i), \\ \left(\delta\hat{\lambda}_h, \hat{\lambda}_h - \gamma\left[\![\underline{\hat{u}}_h]\!] \cdot \underline{n}_{\Gamma_h} + \frac{1}{\gamma}\hat{\lambda}_h\right]_+\right)_{\Gamma_h^c} + s_{\heartsuit}(\delta\hat{\lambda}_h, \hat{\lambda}_h) = 0, \end{cases} \tag{29}$$

where b_h decomposes into b_h^1 and b_h^2 as

$$\begin{aligned} b_h(\hat{\lambda}_h, \delta\underline{\hat{u}}_h) &= \int_{\Gamma_h^c} \hat{\lambda}_h([\![\delta\underline{\hat{u}}_h]\!] \cdot \underline{n}_{\Gamma_h}) \, d\Gamma \\ &= \int_{\Gamma_h^c} \hat{\lambda}_h(\delta\underline{\hat{u}}_h^1 \cdot \underline{n}_{\Gamma_h}) \, d\Gamma - \int_{\Gamma_h^c} \hat{\lambda}_h(\delta\underline{\hat{u}}_h^2 \cdot \underline{n}_{\Gamma_h}) \, d\Gamma \\ &=: b_h^1(\hat{\lambda}_h, \delta\underline{\hat{u}}_h^1) - b_h^2(\hat{\lambda}_h, \delta\underline{\hat{u}}_h^2). \end{aligned} \tag{30}$$

Let k denote the iteration steps starting with $k = 0$. We then iterate between the linear stage and the local stage detailed below.

The linear stage In which we solve the bulk problems: Find the displacement $\underline{\hat{u}}_h \in \hat{\mathcal{U}}_h$ such that for all $i = 1, 2$ and for all $\delta\hat{u}_h \in \hat{\mathcal{U}}_h$

$$\hat{a}_h^i(\underline{\hat{u}}_h^{i,k+1}, \delta\underline{\hat{u}}_h^i) + b_h^i(\hat{\lambda}_h^{i,k+1}, \delta\underline{\hat{u}}_h^i) = l^i(\delta\underline{\hat{u}}_h^i) \tag{31}$$

subject to the descent direction for the interface fluxes $\hat{\lambda}_h^{i,k+1} \in \hat{\mathcal{W}}_h$ and the interface displacements $\hat{w}^{i,k+1} \in \hat{\mathcal{W}}_h$, $i = 1, 2$,

$$(\hat{\lambda}_h^{i,k+1} - \hat{\lambda}_h^{i,k+\frac{1}{2}}) - \gamma(\hat{w}^{i,k+1} - \hat{w}^{i,k+\frac{1}{2}}) = 0, \tag{32}$$

where the quantities from the previous half-iterate are known from the local stage and the interface displacements are given by

$$\left(\hat{w}^{i,k+1}, \delta\hat{\lambda}_h\right)_{\Gamma_h^c} = \left(\underline{\hat{u}}^{i,k+1} \cdot \underline{n}_{\Gamma_h}, \delta\hat{\lambda}_h\right)_{\Gamma_h^c}. \tag{33}$$

The local stage In each quadrature point along the interface Γ_h^c, find the interface fields $\{\hat{w}_h^{i,k+\frac{1}{2}}, \hat{\lambda}_h^{1,k+\frac{1}{2}}\}$, $i = 1, 2$, that satisfy the contact condition

$$\hat{\lambda}_h^{i,k+\frac{1}{2}} = \gamma\left[(\hat{w}_h^{1,k+\frac{1}{2}} - \hat{w}_h^{2,k+\frac{1}{2}}) \cdot n_{\Gamma_h} + \frac{1}{\gamma}\hat{\lambda}_h^{1,k+\frac{1}{2}}\right]_+ \tag{34}$$

subject to search directions

$$(\hat{\lambda}_h^{i,k+\frac{1}{2}} - \hat{\lambda}^{i,k}) + \gamma(\hat{w}^{i,k+\frac{1}{2}} - \hat{w}^{i,k}) = 0, \tag{35}$$

for all $i = 1, 2$ and to equality/definition

$$\hat{\lambda}_h^{2,k+\frac{1}{2}} = -\hat{\lambda}_h^{1,k+\frac{1}{2}}. \tag{36}$$

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13

Page 9 of 23

We regularize the interface quantities $\{\hat{w}_h^{i,k+\frac{1}{2}}, \hat{\hat{\lambda}}_h^{1,k+\frac{1}{2}}\}$ by solving the linear interface problem: Find the filtered Lagrange multiplier $\hat{\lambda}_h^{i,k+\frac{1}{2}} \in \hat{\mathcal{W}}_h$ for all $\delta\hat{\lambda}_h \in \hat{\mathcal{W}}_h$ such that

$$\left(\hat{\lambda}_h^{i,k+\frac{1}{2}}, \delta\hat{\lambda}_h\right)_{\Gamma_h^c} + s_{\heartsuit}(\hat{\lambda}_h^{i,k+\frac{1}{2}}, \delta\hat{\lambda}_h) = \left(\hat{\hat{\lambda}}_h^{i,k+\frac{1}{2}}, \delta\hat{\lambda}_h\right)_{\Gamma_h^c}, \tag{37}$$

for $i = 1$ or $i = 2$. The filtered interface jumps $\hat{w}^{i,k+\frac{1}{2}}$ for $i = 1$ and $i = 2$ can then be recovered locally using search direction (35), replacing the "double hat"—unfiltered—Lagrange multiplier by its regularized counterpart.

Digital pipeline from CT-scans to CutFEM biomechanical simulations

In the following section, we will describe our pipeline from CT-images to CutFEM simulations. We will explain the pipeline on scans of the hip joint. Here, we utilise well known classical algorithms including the decomposition of cells into interior, exterior and intersected cells; and the computation of the signed distance field with a combination of a geometrical computation in the near field and fast marching in the far field and extend it from classically one signed distance function to multiple signed distance functions in the context of contact computations.

Image segmentation and surface triangulation of bone geometries

In the first step, we reconstruct a patient-specific hip joint geometry based on computed tomography (CT) images (see Fig. 2). The images are obtained from the cancer imaging archive (TCIA) [27] and transferred to the 3D slicer software for the hip joint segmentation [1,18].
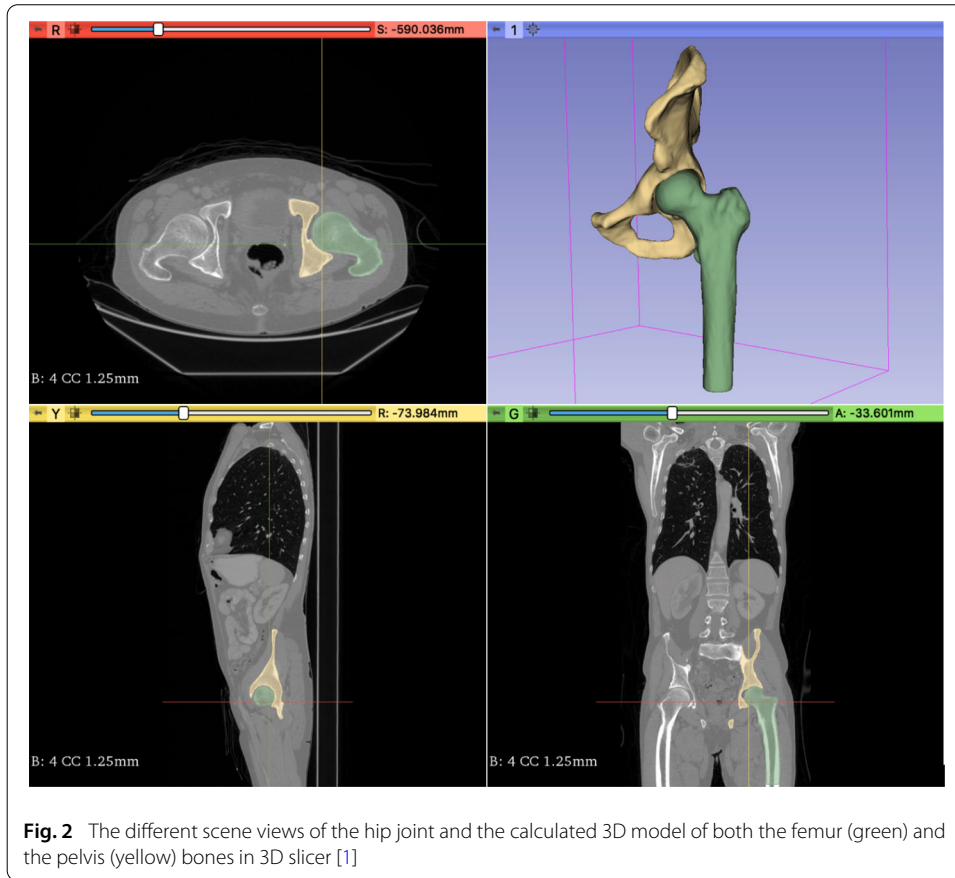
We use multiple landmarks in a few slices to define the femur, the pelvis, and the background regions. Using these landmarks, all the image pixels are labelled and the corresponding 3D volumes are calculated (Fig. 2). Further, we refine the bone models to eliminate rough surfaces, holes, and irrelevant connected tissues. We extract the surface of each bone as a triangulated surface mesh and export them as STL-files. These STL-files are used as the basis for the signed distance function calculation described in the following.

Proposed surface triangulation to level-set algorithm

The surface triangulations, read from the STL-files, are first embedded in a regular background grid and then signed distance functions are computed on this background grid. The following steps describe this surface triangulation to level set function algorithm.

Step 1: Determine Bounding Box Mesh In the first step, we determine a bounding box domain, denoted by Ω^b, which contains both the surface triangulation of the hip bone, denoted by $\mathcal{S}_h^1$, and the surface triangulation of the femur bone, denoted by $\mathcal{S}_h^2$. This bounding box domain, Ω^b, is defined by the minimum and maximum coordinates, $\boldsymbol{x}_{min} = (x_{min}, y_{min}, z_{min})$ and $\boldsymbol{x}_{max} = (x_{max}, y_{max}, z_{max})$, of all vertex coordinates in both surface triangulations $\mathcal{S}_h^1$ and $\mathcal{S}_h^2$. The resulting domain is given by $\Omega^b = [x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$. We then generate a uniform and regular tetrahedral mesh, denoted by $\mathcal{T}_h^b$, on the background box domain Ω^b.

Step 2: Collision detection between the surface triangulations and the background mesh In the second step, we detect all collisions between the surface triangles and the

Claus *et al. Adv. Model. and Simul. in Eng. Sci.* (2021) 8:13

Page 10 of 23

Fig. 2 The different scene views of the hip joint and the calculated 3D model of both the femur (green) and the pelvis (yellow) bones in 3D slicer [1]

background mesh tetrahedra. A collision between a surface triangle and a background mesh tetrahedron (cell) means that the surface triangle and the tetrahedron have a non-empty intersection. An efficient way to detect the collision between each surface triangle and the background mesh cells is to use Axis Aligned Bounding Box (AABB) trees. In this method, instead of testing each background mesh cell for an intersection with each surface triangle $F_j \in \mathcal{S}_h^i$, we detect intersections between two hierarchical structures of bounding boxes (AABB trees, for more details see [16,24,29]). This collision detection between an AABB tree for the surface mesh, $\mathcal{S}_h^i$, $i = 1, 2$, and the AABB tree for the background mesh, $\mathcal{T}_h^b$ yields a set of tetrahedra that have a bounding box which collides with the bounding box around the surface triangle $F_j \in \mathcal{S}_h^i$. We then test among this set of tetrahedra with colliding bounding boxes if the surface triangle indeed intersects with the tetrahedra using geometric predicates (triangle–tetrahedra intersection [16]).

We express the results of this collision detection in a map between the index of each surface triangle, j for surface triangle $F_j \in \mathcal{S}_h^i$, $i = 1, 2$, and the indices of the background mesh tetrahedra, k_m, it intersects (collides) with. Formally, these maps, one for each surface triangulation, are defined as

$$\chi^i : \mathcal{S}_h^i \to \mathcal{T}_h^b, \tag{38}$$

$$j \to \{k_1, k_2, \dots, k_{N_j}\}, \tag{39}$$

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13
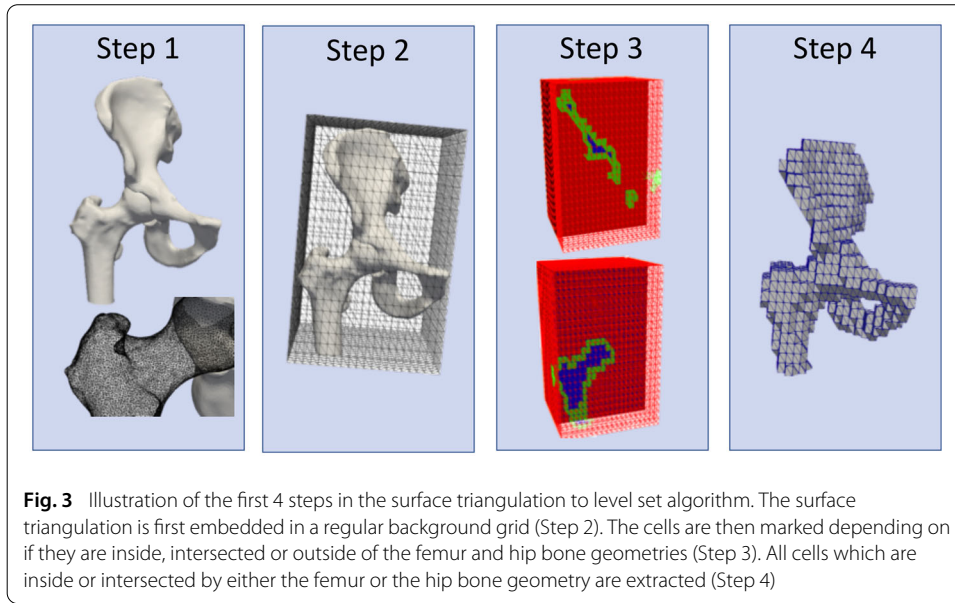
Page 11 of 23

where $i = 1, 2$, j denotes the index of the surface triangle $F_j \in \mathcal{S}_h^i$ and runs from 1 to the number of surface triangles in $\mathcal{S}_h^i$ and k_m, $m = 1, \ldots, N_j$, denotes the index of the background mesh tetrahedra that the surface triangle j intersects with. Note that, one surface triangle in general intersects several tetrahedra in the background mesh. The number of tetrahedra intersected by surface triangle with index j is denoted by N_j.

In addition to these mappings, we also construct the inverse map from the index of the background mesh tetrahedra to the index of the surface triangles that intersect the tetrahedra defined as

$$
\begin{aligned}
\chi^{i,-1} : \mathcal{T}_h^b &\to \mathcal{S}_h^i, \\
k &\to \{j_1, \ldots, j_{N_k}\}.
\end{aligned}
\tag{40}
$$

Here, k denotes the index of the background mesh tetrahedron, and j_m, $m = 1, \ldots, N_k$ denotes the index of the surface mesh triangle $F_{j_m} \in \mathcal{S}_h^i$ that has a non-empty intersection with the background tetrahedron k and N_k is the number of surface triangles which intersect the background mesh tetrahedron with index k. These two maps are crucial for the next steps.
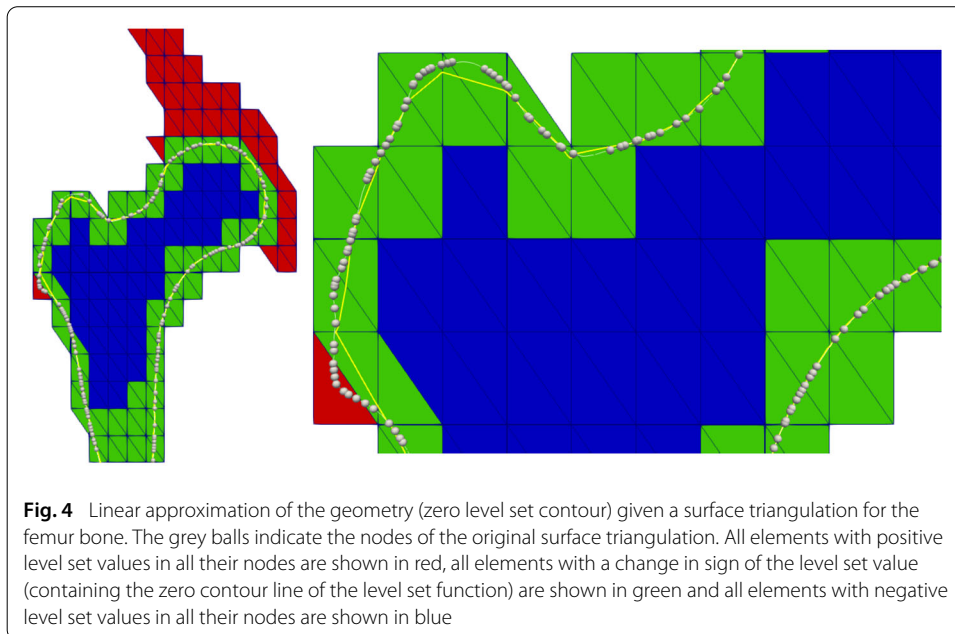
Step 3: Decompose cells in the background mesh into interior cells, intersected cells and exterior cells We give each background mesh tetrahedron, a set of two marker values, $M = (M_1, M_2)$, $M_1 \in \{0, 1, 2\}$, $M_2 \in \{0, 1, 2\}$ depending on if the tetrahedron is inside, intersected or outside by femur or hip bone surface triangulations, $\mathcal{S}_h^i$, $i = 1, 2$. We start with all cells marked as $M = (0, 0)$ and then mark the tetrahedra intersected by $\mathcal{S}_h^i$ using the inverse mapping $\chi^{i,-1}$. From the mapping, we determine the indices of all background mesh tetrahedra that have a non-zero intersection with the surface triangulation $\mathcal{S}_h^i$, $i = 1, 2$ and mark these cells with 1 ($M_1 = 1$ if tetrahedron intersects with cells in $\mathcal{S}_h^1$ and $M_2 = 1$ if tetrahedron intersects with cells in $\mathcal{S}_h^2$). The last step in the mesh marking is to determine all cells outside of the femur bone surface triangulation ($M_1 = 2$) and outside of the hip bone surface triangulation ($M_2 = 2$). To achieve this, we assume that the background mesh cells are clearly separated by a band of intersected background mesh elements into mesh cells inside the femur or hip bone and into mesh cells outside. Starting with a tetrahedron in a corner of the background mesh, for which we are sure that the cell is outside of both the hip and femur bone geometries, we mark this corner cell with $M = (2, 2)$. We then find all cells connected to this starting tetrahedron and see if any of the connected tetrahedron is marked with $M_i = 1$. If the neighbouring cell is marked with $M_i = 1$, we leave the mark at $M_i = 1$ if it is marked with $M_i = 0$, we set $M_i = 2$ and find all cells connected to the new cell we just marked with $M_i = 2$. This way, we find all cells that are connected to the outside corner cell without crossing the band of elements which are marked as intersected. For more details on a mesh marking method for connected domains see [13]. Note that the presented technique is similar to the classical coloring technique employed in Delaunay mesh generation [9]. Alternatively to exploiting connectivities of inside and outside domains, *ray casting techniques* can be used in which the number of intersections of a ray originating in a background mesh cell with the surface mesh can determine if a cell is inside or outside (see e.g. [3]). The decomposition into intersected, interior and exterior cells will be crucial in the next steps.

Fig. 3 Illustration of the first 4 steps in the surface triangulation to level set algorithm. The surface triangulation is first embedded in a regular background grid (Step 2). The cells are then marked depending on if they are inside, intersected or outside of the femur and hip bone geometries (Step 3). All cells which are inside or intersected by either the femur or the hip bone geometry are extracted (Step 4)

Step 4: Extract ficititious domain   In this step, we disregard the cells outside the femur and hip bone geometries. The cells outside of both the femur and the hip bone have no physical meaning and are not of interest in our simulations. We disregard the outside cells by constructing a new background mesh, $\mathcal{T}_h^* \subset \mathcal{T}_h^b$, with all cells from the original background mesh which are marked with $M_1 = 0$ or $M_2 = 0$ and $M_1 = 1$ or $M_2 = 1$. In extracting the cells, we keep track of the mapping of the cell indices of this new "submesh" to the original background mesh to update the mappings $\chi^{i,-1}$ (to map from the surface triangle index to the new indices of the fictitious domain background mesh, $\mathcal{T}_h^*$). We also keep the mesh cell marker. The new background mesh, $\mathcal{T}_h^*$, forms a "Cube"-like mesh which fully contains the hip joint geometry (as shown in Fig. 3).

Step 5: Compute the signed distance functions to the femur and to the hip bone surface meshes.   In this step, we determine two signed distance functions: one to the hip surface mesh, denoted by ϕ_h^1, and one to the femur bone surface mesh denoted by ϕ_h^2. We compute the distance to surface mesh $\mathcal{S}_h^i$ in the following way

1. **Initialization phase.** In the first step, we use a geometry based approach to compute the unsigned distance of all nodes of the intersected tetrahedra to the surface mesh, as described in the following. Using the mapping $\chi^{i,-1}$, we iterate over all cells that are intersected by the surface mesh $\mathcal{S}_h^i$. For each intersected cell, we iterate over its vertex nodes with coordinates $\mathbf{v_p^j}$, $j = 1, 2, 3, 4$. We compute the distance from each vertex node to all surface triangles, which the current element intersects (given by map $\chi^{i,-1}$). This is done by calculating the unsigned distance of a point ($\mathbf{v_p^j}$) to a triangle [16]. Among all distances determined for a vertex node, we keep the shortest distance and save the value in the degree of freedom of the level set finite element function corresponding to the vertex node given by a vertex to dof mapping (for a piecewise linear level set function). Note here, that vertex nodes are often iterated over several times (when they belong to several intersected tetrahedra) and we always keep the shortest unsigned distance value of all computed distances.

Fig. 4 Linear approximation of the geometry (zero level set contour) given a surface triangulation for the femur bone. The grey balls indicate the nodes of the original surface triangulation. All elements with positive level set values in all their nodes are shown in red, all elements with a change in sign of the level set value (containing the zero contour line of the level set function) are shown in green and all elements with negative level set values in all their nodes are shown in blue

2. After initialising all values of the level set at the nodes of all intersected tetrahedra (previous step), we use a fast marching method [32] to compute the approximate unsigned distance of all remaining nodes in the background mesh. In a first step, we iterate over all nodes with known distance value and determine all nodes connected to these nodes via an edge (neighbouring node). We save these connected (neighbouring) nodes in a set called active set. We then compute the distance of these nodes in the active set to the known values, as follows. For each node in the active set, we iterate over all tetrahedra that share this node. For each tetrahedron, we compute the distance of the active node to known values in the tetrahedron (if any). There are four possible cases, firstly, the tetrahedron contains three known values [point (active node)—triangle distance], secondly, the tetrahedron contains two known values [point (active node)—edge (between known nodes) distance]; thirdly, the tetrahedron contains one known value [point (active node)—point (known node) distance] or the tetrahedron contains no known values (no distance computation). Once all nodes in the active set have their distance values (smallest of all computed distances), we determine the next set of nodes connected to all known nodes. We repeat this process until all unsigned distance values in all nodes have been computed. For more details on the employed fast marching method see [21].

3. In a last step, we convert the unsigned distance function computed in the previous steps to a signed distance function using the cell marker computed in step 3. In step 3, we have marked all tetrahedra enclosed by the surface mesh with 0, all cells outside with 2 and all intersected cells with 1. We want negative level set values inside our bone geometries and a positive sign outside such that the zero contour of the level set function represents a linear approximation of the surface mesh (see Fig. 4) and the normalised gradient of the level set represents a normal pointing from inside the bone to the outside. To achieve this, all that is needed is to assign negative signs to all values of the level set in nodes for cells marked with 0.

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13

Page 14 of 23

Following steps 5.1 to 5.3, for each $\mathcal{S}_h^i$, $i = 1, 2$, we obtain the piecewise linear signed distance functions ϕ_h^1 and ϕ_h^2, which provide a linear approximation of the femur and hip bone geometries, respectively. In the next section, we will use these signed distance functions to compute stress distributions in the bones.
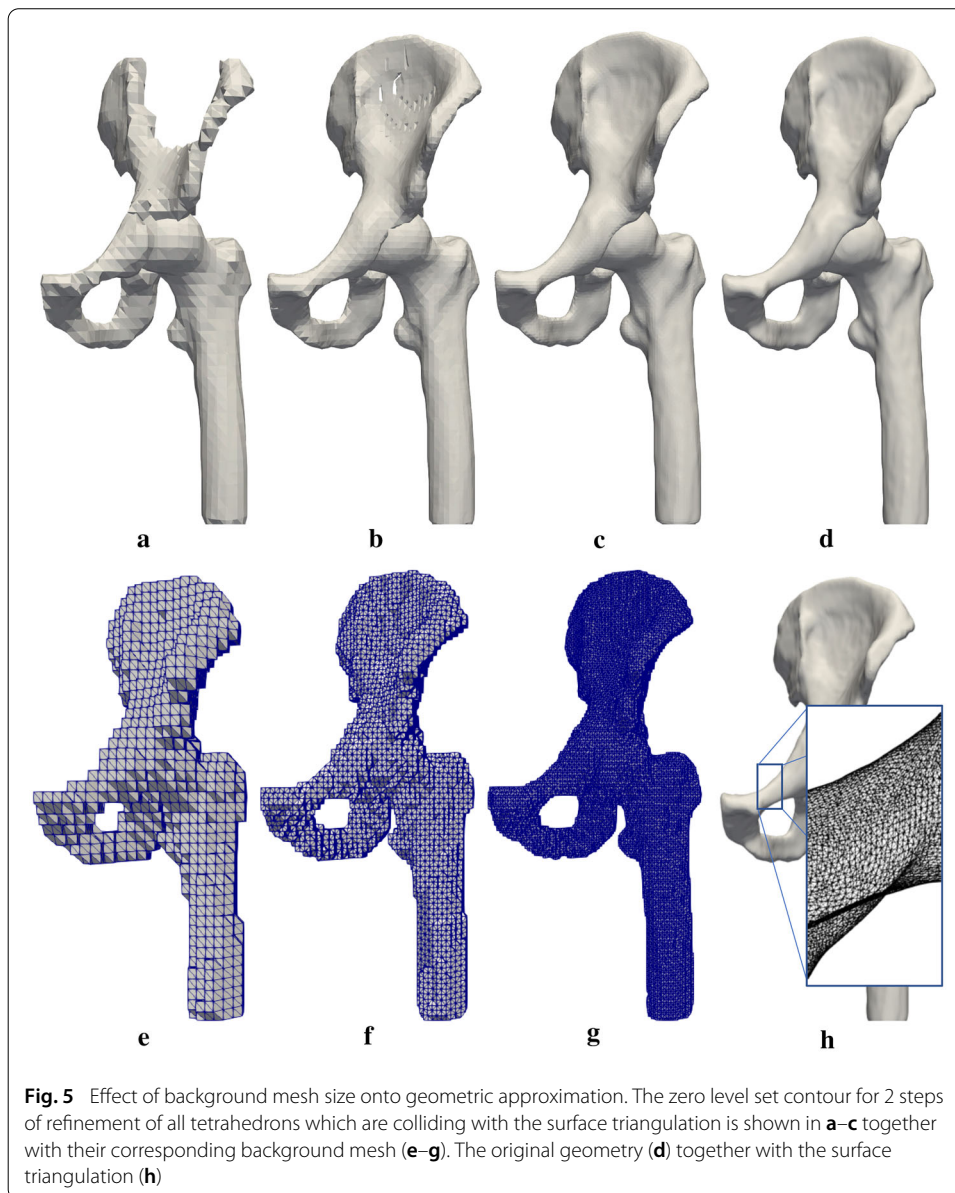
Adaptation of the background mesh to guarantee sufficient geometric resolution

As described in the previous section, we compute a piecewise linear signed distance function to represent the surface triangulation geometry. This piecewise linear representation is close to the surface geometry in planar areas. However, in areas of high curvature or in areas with fine features, we need to ensure that the background mesh size is fine enough to capture these features with our piecewise linear approximation. Here, the finest mesh size is dictated by the thinnest parts of the geometry. In the piecewise linear approximation, the thinnest parts needs to be resolved by at least two cells.

In this section, we will first make some heuristic observations about the quality of the level set approximation for the hip and femur bone geometry for a refinement of all background mesh cells which are colliding with the surface triangulations of the two bones. Secondly, we will use these heuristic observations to suggest two refinement criteria to automatically choose only some of the colliding elements to be refined and to generate a level set approximation that captures the geometry with sufficient accuracy.

Firstly, Fig. 5a, b show the effect of choosing a background mesh, which is too coarse to represent the fine features in the hip bone geometry. The piecewise linear level set approximation cannot capture the fine bone structures in the upper hip bone and in the acetabulum and instead these areas are represented as holes in the geometry. Refining the mesh in the elements, which are colliding with both surface triangulation (as identified by the markers in the algorithm introduced in the previous section) until all features are sufficiently represented, can solve this problem. Figure 5, shows the zero contour line of the level set function for two such background mesh refinement steps (a–c) together with their background mesh (e–g) and the original surface triangulation (d, h). We can see in Fig. 5 that the level set approximation leaves large holes in the hip bone for the initial background mesh. After the first refinement step, we still observe some small holes in the thin part of the upper hip bone which is finally completely closed in the second refinement step.
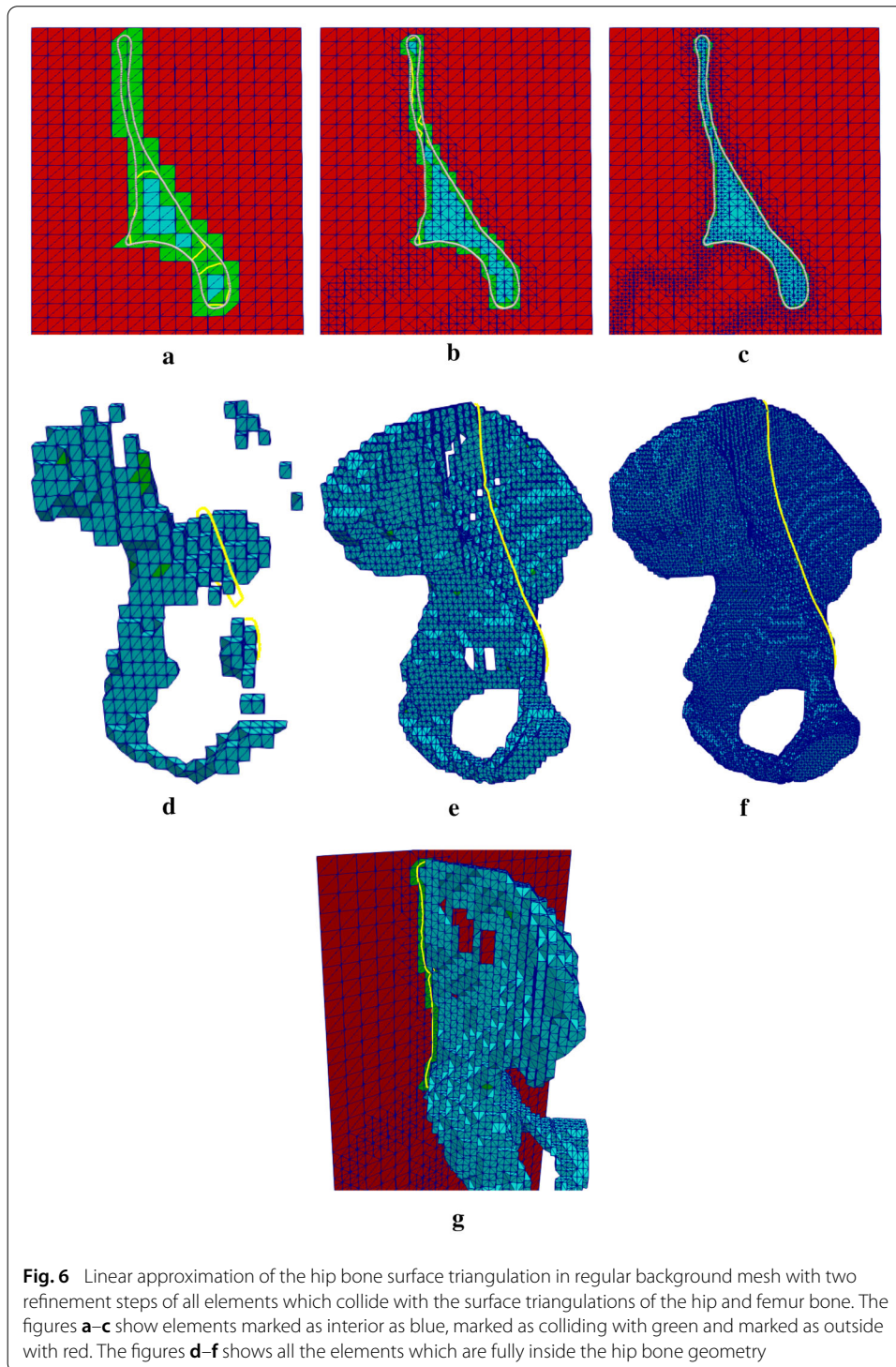
The appearance of these holes in the hip bone geometry are due to sections of the hip bone geometry with no connectivity to interior cells as displayed in Fig. 6. Figure 6a–c shows a slice (shown in Fig. 6g) through the hip bone through a part of the bone with the holes. Elements which are colliding with the surface triangulation of the hip bone or the femur bone are displayed in green, elements which are outside the hip bone are shown in red and elements fully inside the hip bone are shown in blue. The zero level set contour line is shown in yellow and the surface triangulation is shown by spheres in the nodes of the surface triangulation. We observe that in the thin part of the hip bone, elements are detected as colliding but there is no element which is fully inside the hip bone geometry (see Fig. 6d–f). The computed distance function to the surface triangulation only gets assigned a negative sign for all nodes of elements that are fully inside. As the thin section does not contain any elements that are fully inside, the level set function does not change sign and hence there is no zero contour intersecting these elements. This yields the holes.

Claus *et al. Adv. Model. and Simul. in Eng. Sci.* (2021) 8:13

Page 15 of 23

Fig. 5 Effect of background mesh size onto geometric approximation. The zero level set contour for 2 steps of refinement of all tetrahedrons which are colliding with the surface triangulation is shown in **a**–**c** together with their corresponding background mesh (**e**–**g**). The original geometry (**d**) together with the surface triangulation (**h**)

By refining all background elements which are colliding with the surface triangulations, we can close these holes and improve the geometrical approximation. We refine until the zero level set contour is close to the original surface geometry. Here, two refinements were found to be sufficient.

As the femur bone does not contain so many small features, it is much easier approximated by a piecewise linear level set approximation in a coarse background mesh (see Fig. 7). Figure 7 displays the linear approximation of the zero contour line of the level set function in yellow and the nodes of the surface triangulation as spheres in a slice through the femur bone. We observe that the zero contour line approximates the femur shape in the slice very well. Only for the coarsest mesh, we can observe a clear difference between the femur bone and the surface triangulation in areas of high curvature.

Hence, we heuristically observed two major challenges in the resolution of the surface triangulation: the appearance of holes and the approximation of areas with high curvature.

Claus *et al. Adv. Model. and Simul. in Eng. Sci.*(2021)8:13

Page 16 of 23

Fig. 6 Linear approximation of the hip bone surface triangulation in regular background mesh with two refinement steps of all elements which collide with the surface triangulations of the hip and femur bone. The figures **a**–**c** show elements marked as interior as blue, marked as colliding with green and marked as outside with red. The figures **d**–**f** shows all the elements which are fully inside the hip bone geometry

We develop two refinement criteria to address these two challenges. One to identify cells, in which the level set approximation does not change its sign even though the surface triangulation collides with the element (e.g. holes) and one to identify cells in which the zero contour line of the level set is far away from the surface triangulation (e.g. high curvature).
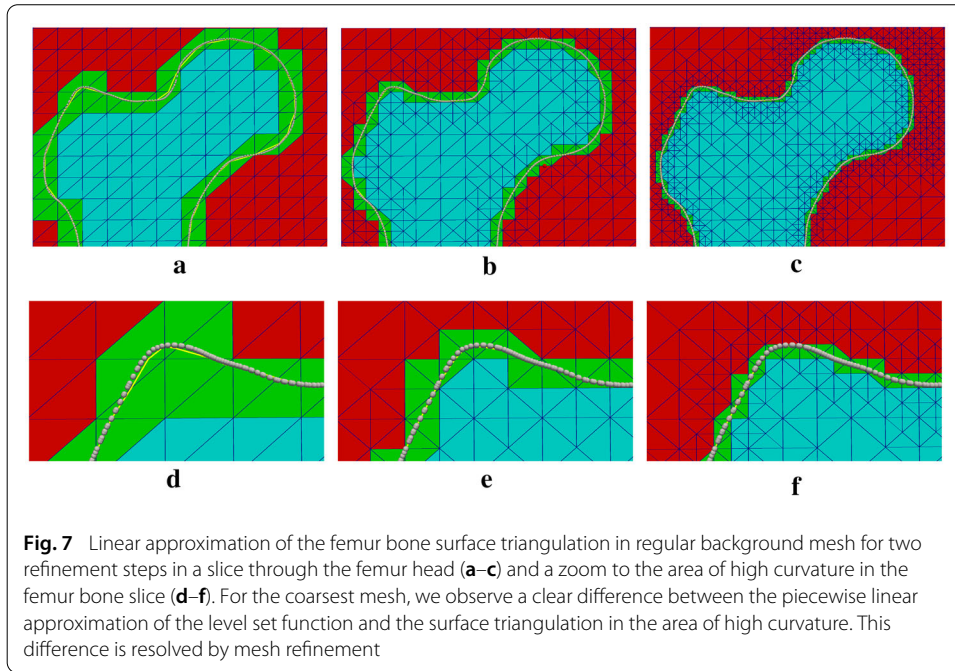
Fig. 7 Linear approximation of the femur bone surface triangulation in regular background mesh for two refinement steps in a slice through the femur head (**a**–**c**) and a zoom to the area of high curvature in the femur bone slice (**d**–**f**). For the coarsest mesh, we observe a clear difference between the piecewise linear approximation of the level set function and the surface triangulation in the area of high curvature. This difference is resolved by mesh refinement

The first refinement criterion is to mark all cells for refinement which are colliding with the surface triangulation but in which the computed level-set function does not change sign. We refer to this refinement criterion as "adaptive sign criterion".

For the second refinement criterion, we determine for each background mesh cell which collides with at least one surface triangulation what the maximum absolute level set value is at all nodes of all surface triangles which collide with the background cell. For example, let us assume background mesh cell j collides with surface triangle k, l and m. We then evaluate the level set function value in each node of the surface triangles k, l and m, which will give us the distance of each node of the surface triangles to the zero level set contour. We take the maximum absolute level-set/distance value of all these node values and store this value for cell j, denoted by d_{max}^{j}. We repeat this process for all cells that collide with surface triangles. We then determine the global maximum value among all these cells, denoted by d_{max}^{global},

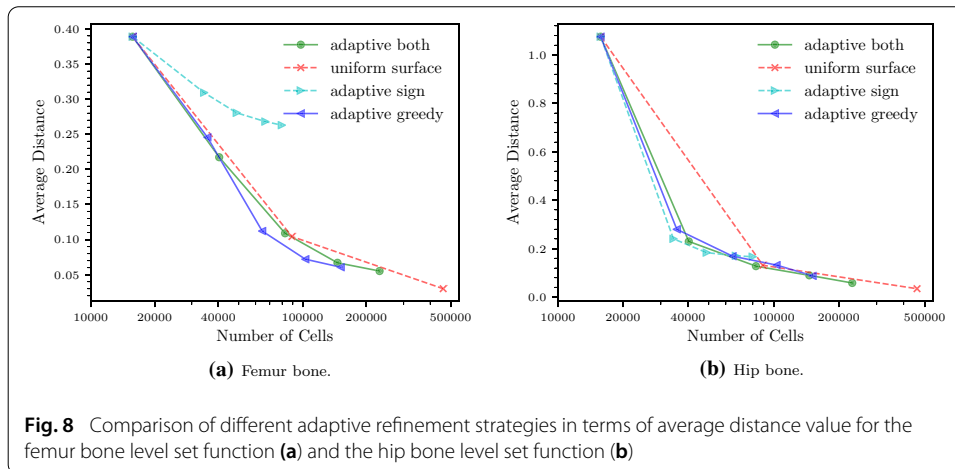$$d_{max}^{global} = \max_{j} d_{max}^{j} \tag{41}$$

where j is a list of cell indices of cells that collide with the surface triangulation. We mark all cells for refinement which have a maximum distance value larger than a threshold ϵ times the global maximum, with $0 < \epsilon < 1$, i.e.

$$d_{max}^{j} > \epsilon \, d_{max}^{global} \tag{42}$$

for each cell j which collides with the surface triangulation. We refer to the second refinement criterion as "adaptive greedy criterion".

To evaluate the quality of the level set approximation with respect to the surface triangulation, we calculate the average distance value, d_{avg}, over all nodes of the surface triangulation, i.e.

$$d_{avg} = \frac{\sum_{k} |\phi^{i}(\boldsymbol{x_k})|}{N_k} \tag{43}$$

Fig. 8 Comparison of different adaptive refinement strategies in terms of average distance value for the femur bone level set function **(a)** and the hip bone level set function (**b**)

where $\boldsymbol{x}_k$ denotes the coordinate of a node in the surface triangulation, $k = [0, 1, \ldots, N_k]$ is the node index, and N_k is the total number of nodes in the surface triangulation.

Figure 8 shows the average distance value of the level set function to the surface triangulation versus the number of cells for the different adaptive refinement criteria. We compare the refinement of all cells that are colliding with the interface (dashed, red) with the sign refinement criterion (dashed, cyan), the greedy refinement criterion with a parameter of $\epsilon = 0.5$ and the combination of both sign and greedy criteria. Figure 8a shows the average distance of the level set function to the surface triangulation for the femur bone and Fig. 8b shows the average distance for the hip bone. For each bone the refinement criteria have very different effects. For the smooth femur bone, the sign criterion, which will refine in the area of holes or very high curvature does not refine as efficiently as greedy or uniform refinement with respect to the average distance. For the hip bone on the other hand, the sign criterion reduces the average distance between the level set function and the surface triangulation most efficiently. This is what we have heuristically observed for the hip bone before. The hip bone has a lot of fine feature and areas of high curvature and this is efficiently resolved by the sign criterion. The greedy criterion performs best for the femur bone and yields lower average distances than uniform refinement for the same number of elements. The greedy algorithm for the hip bone performs slightly less well for the hip bone than the sign criterion up until 70000 cells by which the greedy criterion yields better refinement than the sign criterion. The combination of both sign and greedy criterion yields the lowest average distance to the surface triangulation for the hip bone geometry.
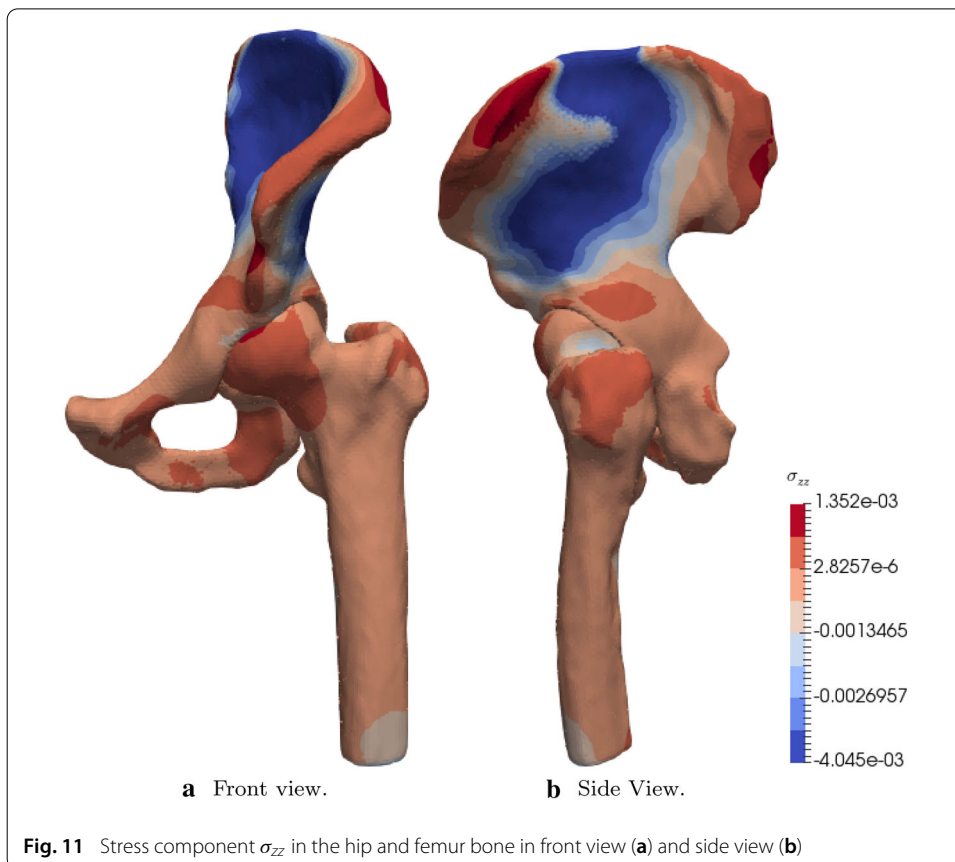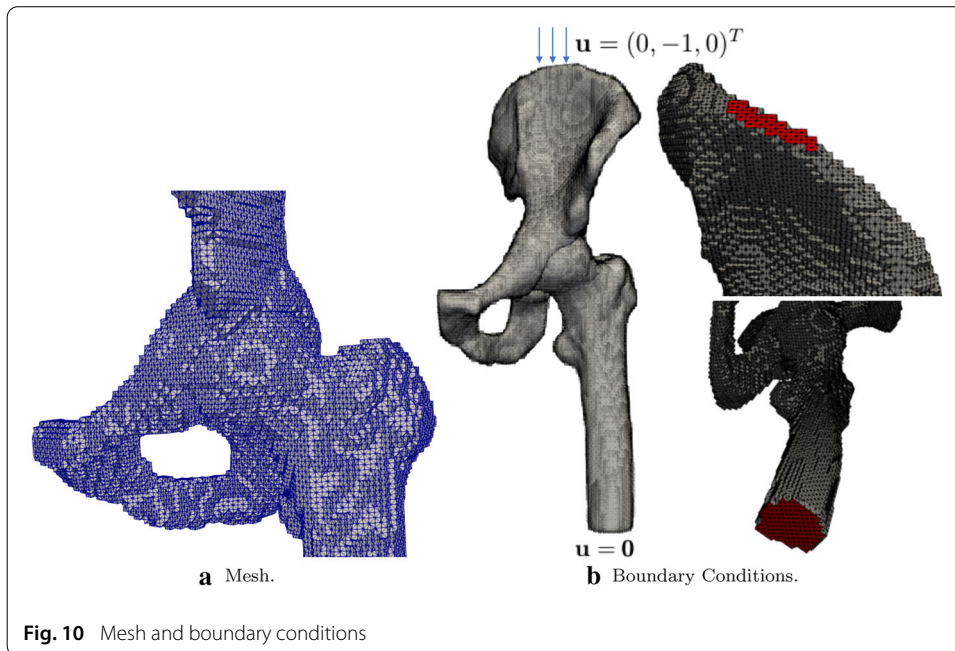
Unilateral contact between hip and femur bone

In this section, we use our LaTIn/CutFEM algorithm described in section to compute the stress distribution in the hip and the femur bone in unilateral contact. All of our software developments are based on FEniCS [2]. In [12], we have shown that our LaTIn/CutFEM algorithm was stable and converged optimally for a range of carefully selected benchmark problems. This contribution focuses on showing that our contact algorithm performs well for geometries obtained from images. In this section, we use the level set geometry description for the femur and the hip bone obtained from the algorithm described in the previous section and compute a unilateral contact problem on the bones.

Claus *et al. Adv. Model. and Simul. in Eng. Sci.* (2021) 8:13

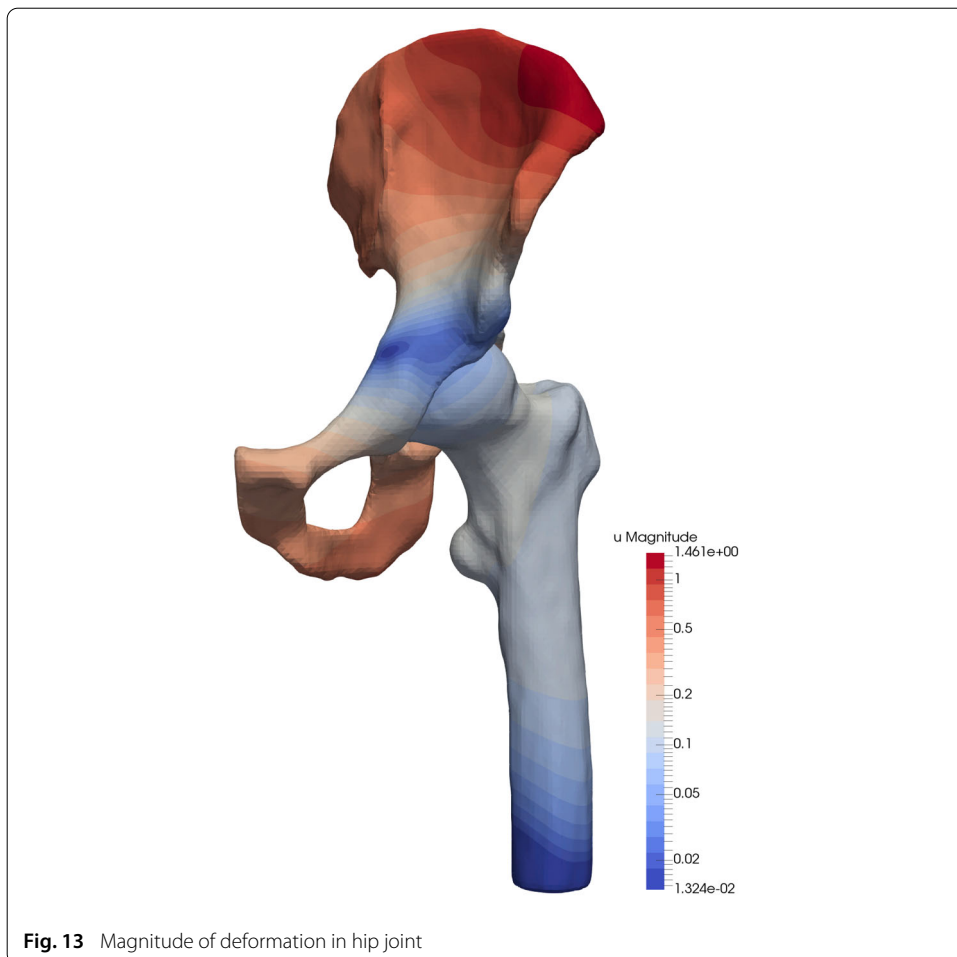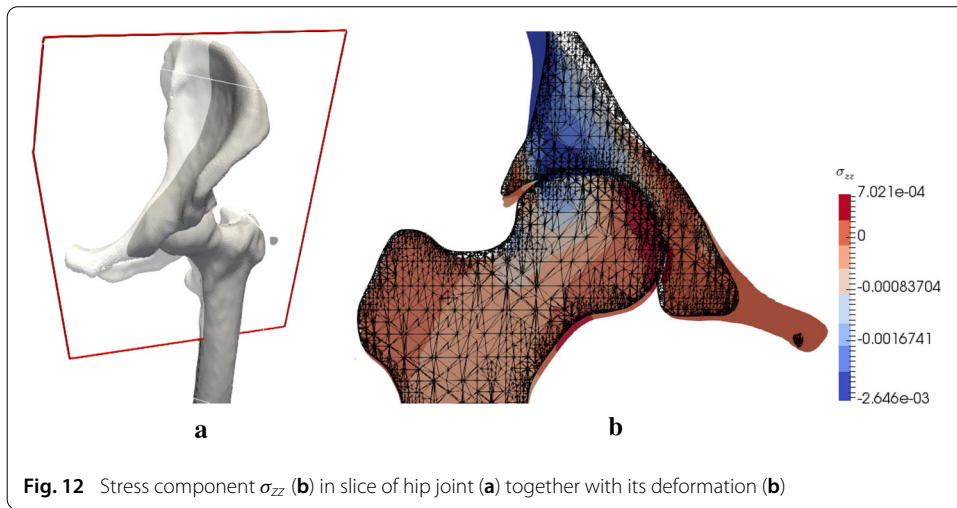Page 19 of 23

Fig. 9 Contact interface
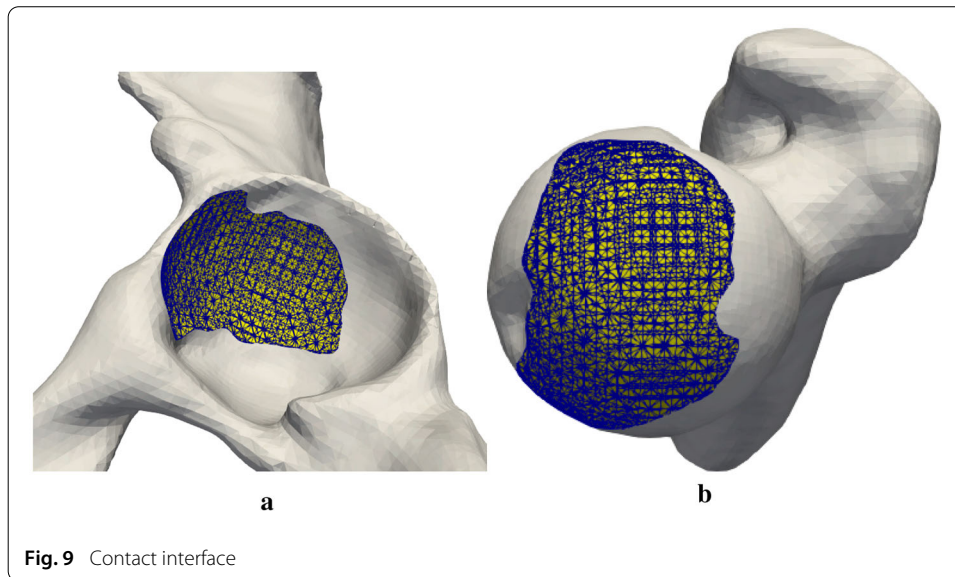
We bring the femur and the hip bone into contact by moving the femur bone geometry slightly into the acetabulum to create a clear contact interface in between the bones. The resulting contact interface is displayed in Fig. 9. Here, we need to predefine the contact interface as our algorithm currently does not include a contact detection algorithm and our algorithm is only valid for small deformations. The extension of the algorithm to include large deformations and contact detection will be addressed in a future publication but is out of the scope of this contribution.

Figure 10 shows the fictitious domain mesh used for the computation, which consists of 460.985 cells yielding 288.426 number of degrees of freedom for the P1 displacement. As boundary conditions, we apply homogenous Dirichlet boundary conditions for the displacement on the bottom of the femur bone and we apply a displacement of $\mathbf{u} = (0, 0, -1)^T$ on the boundary facets marked in red in Fig. 10 on top of the hip bone. We set the Young's moduli for the hip and femur bone to $E^1 = E^2 = 1$, and the penalty parameters to $\alpha = \beta = 0.1, \gamma = 1$.

Figure 11 shows the stress component σ_{zz} for the hip and femur bone. We observe that under our artificial boundary conditions, the thin part of the hip bone is under compression and the rest of the hip bone is under tension. The femur bone is mainly under tension except for a small area in the femoral neck and in the contact region. Looking at a slice through the hip joint as shown in Fig. 12 contact forces transmit the compression force to the top of the femoral head. Figure 12b shows the deformation from the slice in the reference configuration to the slice in the deformed configuration. We observe that the femur bone deforms very little while the hip bone shows a much larger deformation. The magnitude of the deformation is shown in Fig. 13, like seen in the slice the femur bone deforms very little while the hip bone deforms significantly away from the area of contact between the femur and hip bone. The femur bone stabilises the hip bone in the contact region but the hip bone is free to move away from the contact interface. Figure 14 shows the impact of the penalty parameter β, which is used to stabilise the Lagrange multiplier, on the contact pressure. We clearly see that for $\beta = 0$ the contact pressure is strongly oscillating. In [12], we showed that these oscillations grow unbounded with increasing

Claus *et al. Adv. Model. and Simul. in Eng. Sci.* (2021)8:13

Page 20 of 23

a Mesh.   **b** Boundary Conditions.

Fig. 10 Mesh and boundary conditions

a Front view.   **b** Side View.

Fig. 11 Stress component σ_{zz} in the hip and femur bone in front view (**a**) and side view (**b**)

Fig. 12 Stress component σ_{zz} (**b**) in slice of hip joint (**a**) together with its deformation (**b**)

Fig. 13 Magnitude of deformation in hip joint

number of LaTIn iterations. For $\beta > 0$, we observe a smooth contact pressure. For very large β ($\beta = 10$), gradients on the contact pressure are over penalised, which yields an overly stiff interface field that moves like a block.

Fig. 14 Impact of penalty parameter β onto the contact pressure
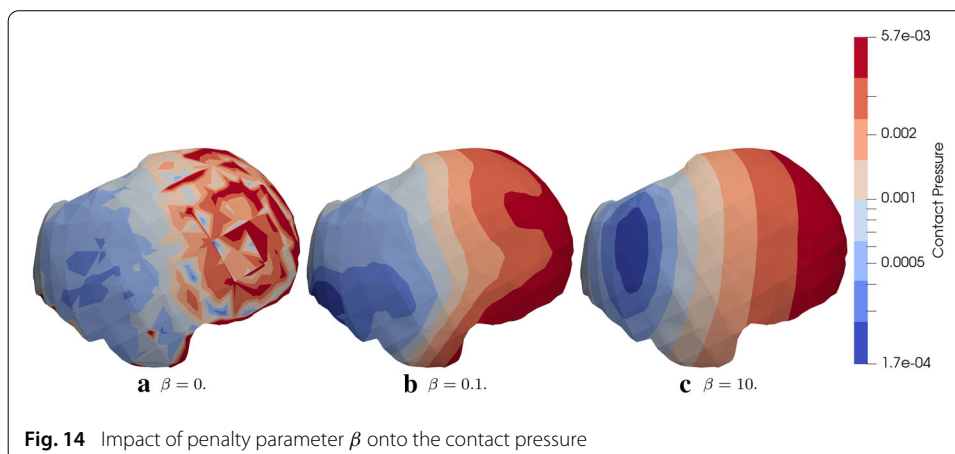
Conclusions

In this article, we have first presented a reformulation of our LaTIn/CutFEM algorithm in terms of an augmented Lagrangian approach for two elastic bodies. We have then detailed our robust algorithm to compute level set representations on finite element meshes for multiple bodies from segmented images. We have discussed in detail on how to adapt the finite element mesh to guarantee a good approximation of the level set function to the surface triangulation. We have developed and analysed two refinement criteria for adaptive mesh refinement. Finally, we have concluded our article with numerical results of a unilateral contact computation between a femur and a hip bone.

Our contact algorithm is currently limited to small deformation. In a future contribution, we are planning to extend our approach to large deformation.

Declarations

Competing interests
The authors declare that there are no competing interests.

Author details
[1]Onera, 8 Chemin de la Hunière, 91120 Palaiseau, France, [2]Mines ParisTech, PSL Research University, MAT–Centre des Matériaux, CNRS UMR 7633, BP 87 91003, Evry, France, [3]School of Engineering, Cardiff University, The Parade, Cardiff CF243AA, UK, [4]Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark, [5]Department of Orthopaedic Surgery, Hvidovre University Hospital, Kettegaards Alle 30, 2650 Hvidovre, Denmark.

References
1. 3D Slicer. https://www.slicer.org/, 2019.
2. Alnæs Martin, Blechta Jan, Hake Johan, Johansson August, Kehlet Benjamin, Logg Anders, Richardson Chris, Ring Johannes, Rognes Marie E, Wells Garth N. The fenics project version 1.5. Archive of Numerical Software. 2015;3(100):1.
3. Baumgärtner D, Wolf J, Rossi R, Dadvand P, Wüchner R. A robust algorithm for implicit description of immersed geometries within a background mesh. Advanced Modeling and Simulation in Engineering Sciences. 2018;5(1):21.
4. Bui HP, Tomar S, Bordas SPA. Corotational cut finite element method for real-time surgical simulation: Application to needle insertion simulation. Computer methods in applied mechanics and engineering. 2019;345:183–211.
5. Burman E. Ghost penalty. Comptes Rendus Mathematique. 2010;348(21–22):1217–20.
6. Burman E, Claus S, Hansbo P, Larson MG, Massing A. CutFEM: Discretizing geometry and partial differential equations. International Journal for Numerical Methods in Engineering. 2015;104(7):472–501.
7. Burman E, Hansbo P. Deriving robust unfitted finite element methods from augmented Lagrangian formulations. In: Geometrically Unfitted Finite Element Methods and Applications, Springer; 2018.
8. Burman E, Hansbo P, Larson MG. Augmented Lagrangian finite element methods for contact problems. ESAIM: Mathematical Modelling and Numerical Analysis. 2019;53(1):173–95.
9. Cheng SW, Dey TK, Shewchuk J. Delaunay Mesh Generation. : Chapman and Hall/CRC; 2016.
10. Chouly Franz, Hild Patrick. A Nitsche-based method for unilateral contact problems: Numerical analysis. SIAM Journal on Numerical Analysis. 2013;51(2):1295–307.
11. Claus S, Bigot S, Kerfriden P. CutFEM Method for Stefan-Signorini Problems with Application in Pulsed Laser Ablation. SIAM Journal on Scientific Computing. 2018;40(5):B1444–69.
12. Claus S, Kerfriden P. A stable and optimally convergent LaTIn-CutFEM algorithm for multiple unilateral contact problems. International Journal for Numerical Methods in Engineering. 2018;113(6):938–66.
13. Dapogny Charles, Frey Pascal. Computation of the signed distance function to a discrete contour on adapted triangulation. Calcolo. 2012;49(3):193–219.
14. Duprez M, Bordas SPA, Bucki M, Bui HP, Chouly F, Lleras C, Lobos V, Lozinski A, Rohan P-Y, Tomar S. Quantifying discretization errors for soft tissue simulation in computer assisted surgery: A preliminary study. Applied Mathematical Modelling. 2020;77:709–23.
15. Duprez M, Lleras V, Lozinski A. Finite element method with local damage of the mesh. ESAIM: Mathematical Modelling and Numerical Analysis. 2019;53(6):1871–91.
16. Ericson Christer. Real-Time Collision Detection. : CRC Press; 2004.
17. Fabre M, Pousin J, Renard Y. A fictitious domain method for frictionless contact problems in elasticity using nitsche's method. The SMAI journal of computational mathematics. 2016;2:19–50.
18. Fedorov Andriy, Beichel Reinhard, Kalpathy-Cramer Jayashree, Finet Julien, Fillion-Robin Jean-Christophe, Pujol Sonia, Bauer Christian, Jennings Dominique, Fennessy Fiona, Sonka Milan, Buatti John, Aylward Stephen, Miller James V, Pieper Steve, Kikinis Ron. 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network. Magnetic resonance imaging. 2012;30(9):1323–41.
19. Fries Thomas-Peter, Belytschko Ted. The intrinsic XFEM: A method for arbitrary discontinuities without additional unknowns. International journal for numerical methods in engineering. 2006;68(13):1358–85.
20. Giovannelli Luca, Ródenas JJ, Navarro-Jiménez JM, Tur M. Direct medical image-based Finite Element modelling for patient-specific simulation of future implants. Finite Elements in Analysis and Design. 2017;136:37–57.
21. Groß S, Reusken A. Numerical Methods for Two-Phase Incompressible Flows. Berlin: Springer; 2011.
22. Hansbo Anita, Hansbo Peter. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. Computer Methods in Applied Mechanics and Engineering. 2004;193(33–35):3523–40.
23. Haslinger J, Renard Y. A new fictitious domain approach inspired by the extended finite element method. SIAM Journal on Numerical Analysis. 2009;47(2):1474–99.
24. Johansson August, Kehlet Benjamin, Larson Mats G, Logg Anders. Multimesh finite element methods: Solving PDEs on multiple intersecting meshes. Computer Methods in Applied Mechanics and Engineering. 2019;343:672–89.
25. Kerfriden P, Allix O, Gosselet P. A three-scale domain decomposition method for the 3D analysis of debonding in laminates. Computational Mechanics. 2009;44(3):343–62.
26. Keyak JH, Meagher JM, Skinner HB, Mote CD Jr. Automated three-dimensional finite element modelling of bone: A new method. Journal of biomedical engineering. 1990;12(5):389–97.
27. Kirk S, Lee Y, Lucchesi FR, Aredes ND, Gruszauskas N, Catto J, Lemmerman J. Radiology Data from The Cancer Genome Atlas Urothelial Bladder Carcinoma [TCGA-BLCA] collection; 2016.
28. Legrain G, Chevaugeon N, Dréau K. High order X-FEM and levelsets for complex microstructures: Uncoupling geometry and approximation. Computer Methods in Applied Mechanics and Engineering. 2012;241:172–89.
29. Massing André, Larson Mats G, Logg Anders. Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions. SIAM Journal on Scientific Computing. 2013;35(1):C23–47.
30. Moës N. Extended finite element method for cohesive crack growth. Engineering Fracture Mechanics. 2002;69:813–33.
31. Ruess Martin, Tal David, Trabelsi Nir, Yosibash Zohar, Rank Ernst. The finite cell method for bone simulations: Verification and validation. Biomechanics and Modeling in Mechanobiology. 2012;11(3):425–37.
32. Sethian James A. Fast marching methods. SIAM review. 1999;41(2):199–235.
33. Viceconti Marco, Bellingeri Luisella, Cristofolini Luca, Toni Aldo. A comparative study on different methods of automatic mesh generation of human femurs. Medical engineering & physics. 1998;20(1):1–10.

Publisher's Note