

Point-cloud method for image-based biomechanical stress analysis

Jing Qian, Jia Lu*

Department of Mechanical and Industrial Engineering, Center for Computer Aided Design

The University of Iowa, Iowa City, IA 52242-1527, USA

Abstract

This paper introduces a point-cloud method for the stress analysis in biological systems. The method takes a point cloud as the geometric input. Each point represents a small material volume which is assumed to undergo a uniform deformation during motion. The strain at each point is computed from the relative displacements of a set of neighboring points. Delaunay tessellation is utilized to provide the neighboring relation and the point volume. An efficient method is developed to extract point-cloud model from medical images. An aorta inflation problem and a skull impact problem are presented to demonstrate the utility of the method.

Keywords: Point cloud method; discrete gradient method; medical images; stress analysis.

*Corresponding author. Email address: jia-lu@uiowa.edu. Tel: +1-319 3356405. Fax: +1-319 3355669.

1 Introduction

Medical images provide a pixel-resolution point-cloud depiction of the anatomy of biological organs. In the traditional finite element analysis, a point cloud needs to be first converted into a CAD (Computer Aided Design) solid and then meshed into a finite element model. The conversion from is by no means a trivial task. Biological organs often contain intricate geometry features that are difficult to describe in CAD geometry. Even if a CAD model is derived, meshing the solid body may still present a significant challenge. With the advance of mesh generators, it is now possible to generate kinematically admissible tetrahedron meshes for an arbitrary 3D geometry. But it is yet impossible to automatically generate analysis-quality mesh for complicate geometries. Presently mesh generation is semi-automatic at the best, and often user's interference is needed. For biomedical systems, mesh generation remains the most tedious and time-consuming step in the procedure of analysis.

Recently, the present authors developed a family of discrete gradient methods (DGM) [1, 2] for stress analysis. The method is motivated by imaged-based applications; an underlying goal is to perform stress analysis directly on point clouds without converting them into a finite element model. The discrete gradient method computes the strain at a point from the relative displacements in a set of neighboring points. This pointwise gradient is utilized to establish the governing equation from a discrete weak form. The method consists of two essential ingredients, one is the identification of the neighboring set, and the other is the computation of discrete gradient. In [1, 2], we described two approaches of defining neighbors and derived the respective discrete gradient formula. These two formulations pass the constant strain patch test by construction. It has been demonstrated that DGM retains the accuracy and convergence rate of bilinear quadrilateral elements, exhibits a locking-free behavior, and is more tolerant to mesh distortion. The method is easy to implement since it does not construct the continuum assumed solutions and thus bypasses the computation of basis functions, either element-based or meshfree.

More importantly, the method has the potential of lending itself to a fully automated procedure for image-based analysis.

The objective of this paper is to describe such a procedure. The proposed paradigm is markedly different from the voxel finite element method (VFEM) whereby voxels are directly converted into eight-node hexahedral elements [3, 4, 5, 6, 7, 8]. A limitation of the voxel method, as pointed out by [9], lies in the jagged stair-step surfaces. The non-smooth surface can cause numerical problems and sacrifices the algorithmic robustness. To overcome this shortcoming, smoothing techniques were adapted [10, 11]. Algorithms were developed to modify the hexahedrons near the surface while keeping the interior mesh intact [12, 13, 14]. These strategies amend the hexahedron mesh with layers of tetrahedron elements near the boundary. In contrast, the current method is not element-based; the continuum assumed solution is never constructed.

The paper starts with a review of the discrete gradient method. An efficient method for segmenting point clouds from medical images is introduced in Section 3, where the entire procedure pipelining from point-cloud registration to stress analysis is also described. Numerical examples are presented in Section 4 to demonstrate the method.

2 Discrete gradient method

2.1 Gradient interpolation vectors

The starting point of DGM is the approximation of the pointwise gradient of a function using a generalized finite difference. Assuming we can associate each point with a set of neighbors and denoting by N_I the local point set that comprises of the point I and its neighbors, we seek to construct *gradient interpolation vectors* \mathbf{R}_{IJ}^* to approximate the gradient of the function u at the node I by

$$(\nabla^h u)_I = \sum_{J \in N_I} \mathbf{R}_{IJ}^* u_J . \quad (1)$$

For second order problems a linear consistency condition is imposed such that the operator (1) can exactly recover the gradient of any linear function. This condition is satisfied if and only if

$$\sum_{J \in N_I} \mathbf{R}_{IJ}^* = \mathbf{0}, \quad \sum_{J \in N_I} \mathbf{R}_{IJ}^* \otimes \mathbf{x}_J = \mathbf{I}, \quad (2)$$

where \mathbf{x}_J is the coordinate the node J .

In the discrete setting, the domain is initially defined by a set of points. Each point represents a small volume (or area in 2D) around it. In the sequel, we use the term *volume* to mean the Lebesgue measure in the respective dimensions. It is required that

$$\sum_I V_I = V, \quad (3)$$

where V is the total volume of the domain and V_I is the point volume. In [1, 2], we utilized a geometric construction whereby the domain is partitioned into non-overlapping and domain-filling cells, each containing a point. The volume V_I is taken to be the cell volume and hence the condition (3) is satisfied by construction.

In [2], we described a discrete gradient operator over an arbitrary polygon mesh. First, the domain is partitioned into subdivision cells using a generalized subdivision process which for a triangular mesh reduces to the standard barycentric subdivision [15]. The nodal cell consists of the union of the subdivision quadrilaterals. Consequently, the neighbors of a point comprise of the vertices of all polygons sharing that point. An example is shown in Figure 1. The point volume V_I is the sum of the subdivision cells containing the point I . The volume depends smoothly on the nodal coordinates of the neighboring set. It turns out that the vectors $\mathbf{R}_{IJ}^* = \frac{1}{V_I} \frac{\partial V_I}{\partial \mathbf{x}_J}$ satisfy the linear consistency condition (2) and hence they furnish a set of interpolant vectors for the discrete gradient operator defined in equation (1).

The method can be readily extended to tetrahedral or hexahedral meshes. In this case, a tetrahedral element is divided into four polyhedrons of equal volume, as shown in Figure 2. A hexahedral element is divided into eight hexahedrons by connecting the barycenters of the element, the centers of the facets and the middle points of the element

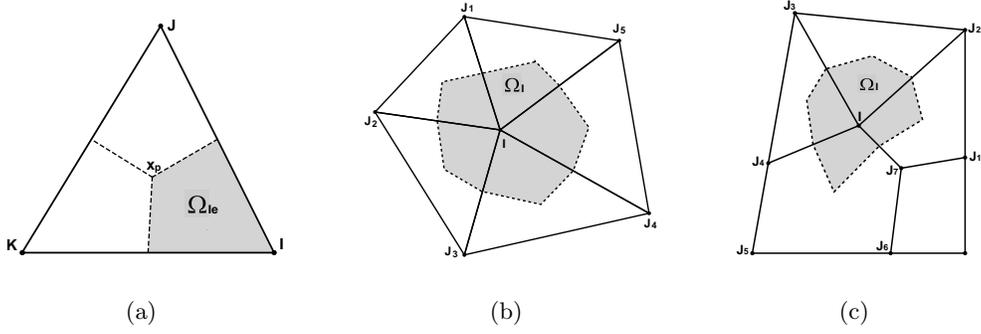


Figure 1: Barycentric subdivision and nodal cell: (a) Triangle subdivision; (b) Nodal cell in a triangle mesh and the nodal set N_I consisting point I and points J_1 through J_5 ; (c) Illustration of a nodal cell in polygon mesh and the nodal set N_I consisting point I and points J_1 through J_7 .

edges. The volume of each subdivision hexahedron can be expressed as a function of the vertices with the aid of finite element isoparametric mapping.

The cell Ω_I is the union of subdivision polyhedrons containing the point I . The cell volume is

$$V_I = \sum_e V_{eI} , \quad (4)$$

where V_{eI} is the volume of the subdivision polyhedrons from element elements e , and the sum runs through the elements connecting to the node I . As in the 2D case, the gradient interpolants are defined by

$$\mathbf{R}_{IJ}^* = \frac{1}{V_I} \frac{\partial V_I}{\partial \mathbf{X}_J} , \quad (5)$$

A key property of this construction is that the point volume V_I is translation-invariant under any rigid body motion, and affine-invariant under any uniform stretching or shearing applied to the points. That is

$$\begin{aligned} V_I(\mathbf{x}_I + \mathbf{a}, \mathbf{x}_{J_1} + \mathbf{a}, \dots, \mathbf{x}_{J_n} + \mathbf{a}) &= V_I(\mathbf{x}_I, \mathbf{x}_{J_1}, \dots, \mathbf{x}_{J_n}) \quad \forall \text{ vector } \mathbf{a} \in \mathbb{R}^3, \\ V_I(\mathbf{F}\mathbf{x}_I, \mathbf{F}\mathbf{x}_{J_1}, \dots, \mathbf{F}\mathbf{x}_{J_n}) &= \det(\mathbf{F})V_I(\mathbf{x}_I, \mathbf{x}_{J_1}, \dots, \mathbf{x}_{J_n}) \quad \forall \text{ linear map } \mathbf{F} : \mathbb{R}^3 \mapsto \mathbb{R}^3, \det \mathbf{F} \neq 0. \end{aligned} \quad (6)$$

It has been proved in [2, Section 2] that the invariance condition (6) implies (2), which in turn guarantees the linear consistency of the gradient operator defined by (1).

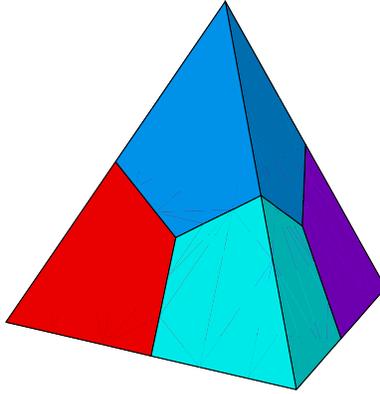


Figure 2: Barycentric subdivision of a tetrahedron

Remarks

1. In a tetrahedron mesh, the gradient defined here is identical to the nodal average gradient [16, 17, 18, 19]. Nevertheless, the current construction is more flexible in the sense that the subdivision does not have to be barycentric. In other words, the volumes of the subdivision polyhedrons do not have to be equal. The formula (5) applies to arbitrary subdivision as long as the point volume V_I obeys the affine invariance condition (6).
2. The derivation of \mathbf{R}_{IJ}^* for a tetrahedral mesh proceeds directly and the details are omitted here. For a hexahedron, the volume of each subdivision element and its derivatives to vertex coordinates can be derived symbolically with the aid of finite element isoparametric mapping. The expressions are lengthy and thus not recorded. For a cubic element, the expression simplifies significantly.
3. The current method bears similarities with the smoothed strain FEM (SFEM) [20, 21, 22], but the computational structure differs substantially. The calculation of nodal gradient in DGM relies on the differentiation of area or volume of a nodal cell, instead of smoothing an underlying continuum strain over the subcells. Remarkably, the strain computation in DGM is purely discrete. By comparing the results shown in [2, 23] and [22], it can be seen that the DGM has the same level of accuracy and

convergence rate as SFEM.

2.2 Discrete gradient method for finite strain elasticity

Let \mathbf{x} represent the position of a material point in the current configuration and let \mathbf{X} be the position of the same material point in the reference configuration. In the discrete setting, the deformation $\Phi : \mathbf{x} = \Phi(\mathbf{X})$ is defined point-wisely by displacement values at nodes. The discrete deformation gradient at the node I can be given by

$$\mathbf{F}_I^h := (\nabla^h \Phi)_I = \sum_{J \in \text{nodes}(\Omega_I)} \mathbf{x}_J \otimes \mathbf{R}_{IJ}^*, \quad (7)$$

where the interpolant \mathbf{R}_{IJ}^* is defined relative to the reference geometry. A variation of the nodal positions induces a corresponding variation of the deformation gradient

$$\delta \mathbf{F}_I^h := (\nabla^h \delta \Phi)_I = \sum_{J \in N_I} \delta \Phi_J \otimes \mathbf{R}_{IJ}^*. \quad (8)$$

The discrete *spatial gradient* of the displacement variation is introduced as

$$(\nabla^h \delta \Phi)_I = (\delta \mathbf{F}_I^h)(\mathbf{F}_I^h)^{-1}. \quad (9)$$

In the Lagrangian setting, the equilibrium is governed by the boundary value problem

$$\begin{aligned} \text{Div } \mathbf{P} + \rho_0 \mathbf{b} &= \rho_0 \ddot{\mathbf{u}}, \quad \mathbf{P} \mathbf{F} = \mathbf{F}^T \mathbf{P}^T \quad \text{in } \Omega_0, \\ \mathbf{P} \mathbf{N} &= \bar{\mathbf{p}} \quad \text{on } \partial \Omega_{0t}, \\ \Phi &= \bar{\Phi} \quad \text{on } \partial \Omega_{0u}, \end{aligned} \quad (10)$$

where \mathbf{P} is the first Piola-Kirchhoff stress, \mathbf{b} is the body force per unit mass, where \mathbf{u} is the displacement vector, \mathbf{N} refers to the unit outwards normal vector on the boundary, $\bar{\mathbf{p}}$ is the applied boundary traction, $\partial \Omega_0$, and the subscript 0 denotes quantities in the reference configuration. The continuum weak form of the boundary value problem is

$$\int_{\Omega_0} \nabla \delta \Phi : \mathbf{P}(\mathbf{F}) dA + \int_{\Omega_0} \delta \Phi \cdot \rho_0 \ddot{\mathbf{u}} dA - \int_{\Omega_0} \delta \Phi \cdot \rho_0 \mathbf{b} dA - \int_{\partial \Omega_{0t}} \delta \Phi \cdot \bar{\mathbf{p}} dS = 0, \quad (11)$$

where $\delta \Phi_I$ is any kinematically admissible virtual displacement.

Having the point-wise discrete gradient, the weak form can be approximated as

$$\sum_I V_{0I} (\nabla^h \delta \Phi)_I : \mathbf{P}(\mathbf{F}_I^h) + \sum_I V_{0I} \delta \Phi_I \cdot \rho \ddot{\mathbf{u}}_I = \sum_I V_{0I} \delta \Phi_I \cdot \rho \mathbf{b}_I + \sum_I A_{tI} \delta \Phi_I \cdot \bar{\mathbf{p}}_I, \quad (12)$$

where A_{tI} stands for the area of the cell surface over which external surface traction is applied. The domain boundary is naturally defined as the union of the exterior boundary of the cells. For a hyperelastic material, the first Piola-Kirchhoff stress is derived from the material strain energy function W via $\mathbf{P}(\mathbf{F}_I) = \frac{\partial W}{\partial \mathbf{F}_I}$. The right-hand side of equation (12) can be collectively written as

$$\sum_I \mathbf{f}_I^{ext} \cdot \delta \Phi_I,$$

and hence $\mathbf{f}_I^{ext} = V_{0I} \rho \mathbf{b}_I + A_{tI} \bar{\mathbf{p}}_I$. The force \mathbf{f}_I^{ext} corresponds to the resultant external force on the cell I . The stress divergence term gives rise to the nodal residual vector \mathbf{f}_I^{int} such that

$$\sum_I \delta \Phi_I \cdot \mathbf{f}_I^{int} = \sum_I V_{0I} (\nabla^h \delta \Phi)_I : \mathbf{P}(\mathbf{F}_I^h).$$

Putting them together, the discrete weak form yields the following set of nonlinear differential equations in time:

$$\mathbf{f}_I^{int} + m_I \ddot{\mathbf{u}}_I - \mathbf{f}_I^{ext} = \mathbf{0} \quad \forall I, \quad (13)$$

where the pointwise mass $m_I = \rho V_{0I}$. The system is solved numerically using a Newmark method in time domain. At each time step, a Newton method is employed to solve the nonlinear algebraic equations. The tangent stiffness matrix is computed in exactly the same manner as the finite element method. Interested readers are referred to [1] for the expressions of the matrices.

2.3 Stability of discrete gradient method

Similar to the nodal strain or the nodal integration methods [16, 17, 24, 25, 26, 27], the discrete gradient method is prone to having spurious singularity or low energy modes arising from stain cancellation in the averaging process. The nature of this kind of instability has been clearly delineated in [28, 23]. Following [28], we adopted a stabilization scheme for

small strain elasticity [23]. The method penalizes the difference between the nodal average strain and the subcell strains by augmenting the strain energy with a penalty term:

$$\int_{\Omega} \boldsymbol{\varepsilon} : \mathbb{D} \boldsymbol{\varepsilon} da \rightarrow \sum_I^N V_I \boldsymbol{\varepsilon}_I : \mathbb{D} \boldsymbol{\varepsilon}_I + \sum_I^N \sum_{e \in \text{elem}(\Omega_I)} \alpha V_{eI} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_{eI}) : \tilde{\mathbb{D}} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_{eI}). \quad (14)$$

Here, $\boldsymbol{\varepsilon}$ is the strain tensor, \mathbb{D} is the fourth-order elasticity tensor, V_{eI} is the volume of subdivision element, $\boldsymbol{\varepsilon}_I$ is nodal average strain and $\boldsymbol{\varepsilon}_{eI}$ is subcell strain, α ($0 \leq \alpha \leq 1$) is a stabilization parameter, and $\tilde{\mathbb{D}}$ is a positive-definite material tensor which could be different from \mathbb{D} . The subcell strain is computed by applying the discrete gradient formula to a subdivision element. Puso and Solberg recommended the following parameters for isotropic elasticity:

$$\alpha = 0.05, \quad \hat{\mu} = \mu, \quad \hat{\lambda} = \min(\lambda, 25\mu).$$

and these parameters were used in the present study. The detail of the formulation is contained in [23]. No stability treatment was implemented for finite strain yet.

3 Point-cloud method

This point cloud method consists of the following steps:

1. Point-cloud registration: converting image pixels into material points.
2. Tessellation: implementing Delaunay triangulation to the extracted point-cloud.
3. Modification: removing spurious boundary triangles or tetrahedron.
4. Analysis: applying the discrete method to the point cloud.
5. Postprocessing: reporting displacement, stress etc. at every point.

The method utilizes Delaunay tessellation to generate a simplex mesh for the purpose of identifying the neighboring set and determining the point volume. Although a mesh is generated, the method is not element-based because element-wise assumed solution is not constructed. The DGM solver is more tolerant to mesh distortions. It has been

demonstrated in [2] that DGM is less sensitive to mesh distortions due to the utility of average strain. In a triangular mesh, for example, the nodal strain turns out to be a weighted average of element strains from the surrounding elements and by construction, a skewed element will carry a relatively smaller weight. For this reason, simplex meshes from Delaunay tessellation are directly permitted.

It is also worth noting that, although Delaunay tessellation is implemented in this study, it is not a pre-requisite for the point cloud method. Unlike a random point set, image data come with a pre-existing adjacent relation between points. For a given point, one can identify its adjacent neighbors from their index in the image array, and this information suffices to establish a neighboring relation. An alternative formulation that bypasses tessellation will be reported elsewhere.

3.1 Point cloud registration

We utilize a threshold-based segmentation method to register image points that are supposed to belong to a specific organ. This method is perhaps the most primitive method of image registration. However, it appears to be adequate for our application because only a point cloud is sought. We assume that a material constituent comprise of pixels within a certain range of gray-scale values. The spatial region which contains the segmented organ is first manually identified. Then, pixels within given thresholds are isolated and an initial point cloud is formed. The physical coordinates are inferred from the image resolution (distances between pixels and slices).

Coarser models can be obtained by lumping several pixels into a point. This is implemented using a moving window technique. In 2D a window is a square region containing $n \times n$ pixels. In 3D, a window is a cube which contains $n \times n \times m$ pixels where m denotes the number of image slices in the height direction. The position of the material point in a window is calculated according to

$$\mathbf{x} = \frac{\sum_I^N \mathbf{x}_I \varphi_I}{\sum_I^N \varphi_I}, \quad (15)$$

where \mathbf{x}_I is the coordinate of the center of the pixel I , φ_I is the gray-scale value, and N the number of pixels in the window. Figure 3 illustrates an example of this utility. Note that the reduced cloud in general no longer forms a regular grid.

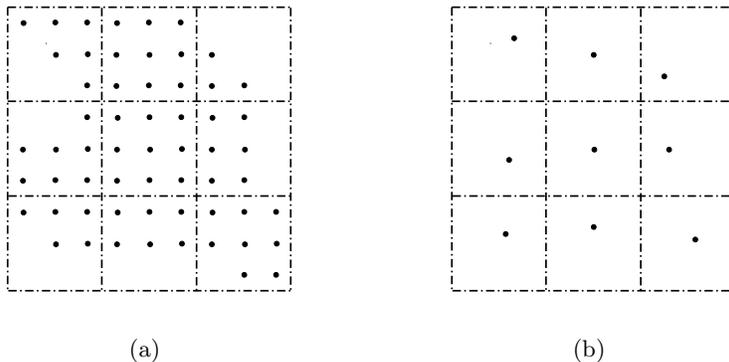


Figure 3: Example of point-grouping. (a) Original point cloud; (b) Reduced cloud.

Figure 4 shows a cross sectional CT (Computed Tomography) scan of an aorta with 0.2 mm resolution. By examining the gray-scale values, it was determined that pixels in the range of 200-255 correspond to the aorta tissue. These points were selected and the resulting pixel-resolution point cloud is shown in Figure 5(a). Notably, the segmentation process requires minimal user interference. A reduced cloud containing 895 points is shown Figure 5(b). It was extracted using 3×3 windows, which are $0.6\text{mm} \times 0.6\text{mm}$ in spatial resolution.

3.2 Tessellation

The open source program Qhull [29] is employed to perform Delaunay tessellation on point clouds. The algorithm is distance-based; for points on a concave boundary the algorithm will produce spurious triangles or tetrahedrons. Figure 7(a) shows Delaunay triangles formed by Qhull on a set of points. It is clearly seen that several points across the bay are wrongly connected, creating spurious triangles that needs to be eliminated.

For an arbitrary cloud, an available method for modifying the boundary of Delaunay mesh is so-called α -shape introduced by Edelsbrunner [30, 31]. α -shapes provide the mathematical framework and a general algorithm to define the hierarchy of point clouds through

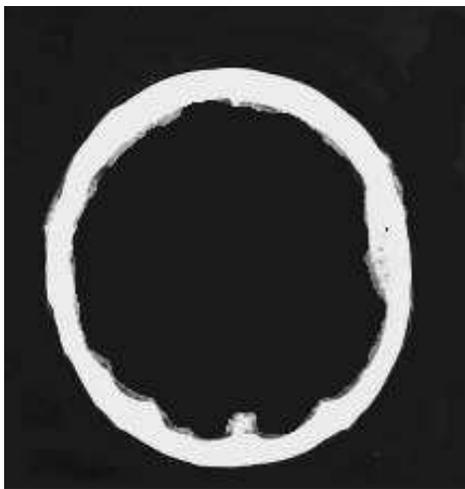


Figure 4: CT scanning image of aorta cross section

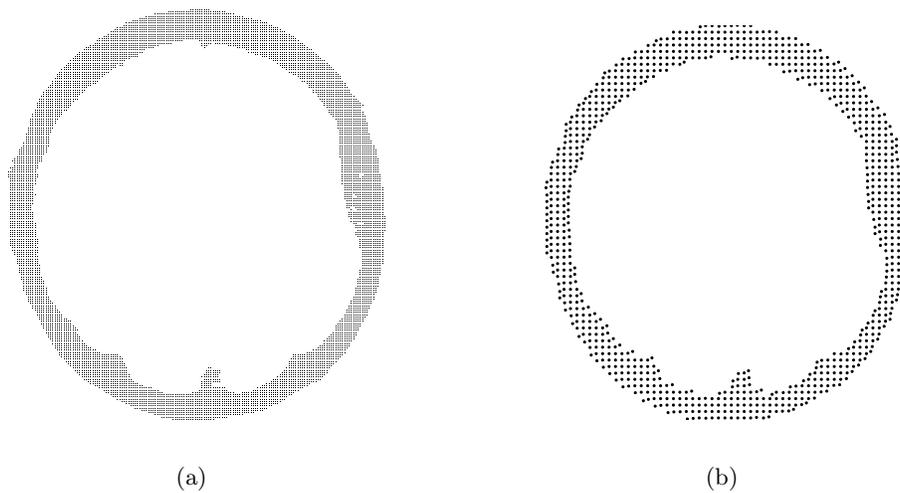


Figure 5: Aorta point clouds extracted from image: (a) Pixel resolution; (b) 9-fold reduction using 3×3 window.

Delaunay triangulation. Some applications of α -shape can be found in [32, 33]. For image-based point clouds there is a pre-existing adjacent relation between the pixels or windows; upon which the modification can be readily implemented. Below, we propose an efficient algorithm for image-based point clouds.

Two-dimensional triangular tessellation

For point arrays derived from images, each window (or pixel) is always surrounded by a number of adjacent windows (pixels). One can identify all points in adjacent windows as the *adjacent points*. For example, in Figure 6, the point A is fully surrounded and has eight adjacent points while the point B has seven. With this adjacent relationship, one can define an admissible triangle by the criterion that at least one of its vertices is simultaneously adjacent to the other two. Conversely, if none of the vertices is adjacent to the other two points, the triangle is deemed spurious and will be removed. Applying this procedure to the point cloud in 7(a), several triangles in the bay and on the upper boundary were deleted, resulting in the modified mesh in Figure 7(b). Note that, in this process, the boundary of the domain is automatically formed.

Prior to tessellation, any “hanging point” must be deleted. These are the points that are not properly connected to the rest of the cloud. In two-dimensional case, if a point has less than two adjacent points, it is a hanging point and should be removed from the cloud.

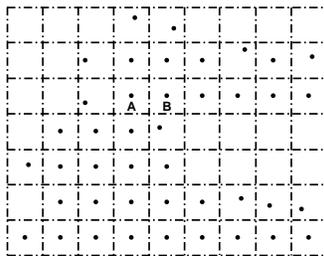


Figure 6: Adjacent points.

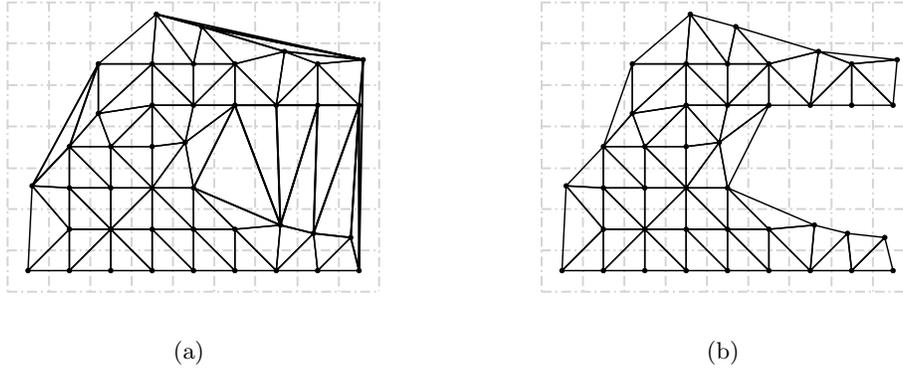


Figure 7: Delaunay triangulation of point clouds: (a) Before removing spurious triangles; (b) After removing spurious triangles.

Three-dimensional tetrahedral tessellation

The same logic applies to a 3D tetrahedron mesh, in which case a material point have at most 26 adjacent points. Any point having less than three adjacent points is deemed a hanging node and will be deleted prior to tessellation. We adapted a criterion for an admissible tetrahedron such that at least one of its vertices is simultaneously adjacent to the other three. Note that, according to this criterion, an admissible tetrahedron could contain facets of which none of the vertices are adjacent to each other. It is possible to adapt more stringent criteria, for example, requiring all facets containing mutually adjacent points, but we found that the present criterion appears to be adequate, at least for the applications considered in this work. In particular, it results in reasonably good boundary depiction for the skull model discussed later.

4 Examples

4.1 Static analysis: inflation of aorta

We use this example to make a compare with finite element method. The point-cloud in Figure 5(b) is assumed to represent the stress free configuration of an aorta cross section. Although aortic tissue exhibits anisotropic behavior, for demonstration purpose, we model

the wall tissue by an isotropic hyperelastic energy function

$$W = \frac{\mu_1}{2}(I_1 - 3) + \frac{\mu_2}{4}(I_1 - 3)^2 ,$$

where $I_1 = \text{tr}(\mathbf{F}^T \mathbf{F})$, \mathbf{F} is the deformation gradient and $J = \det \mathbf{F}$. A penalty energy $\frac{\kappa}{2}(\log(J))^2$ with a large value κ is augmented to the energy function to introduce nearly incompressibility. The following numerical values are used:

$$\mu_1 = 100, \quad \mu_2 = 5, \quad \kappa = 1000000 .$$

We ignore the residual stress which is known to exist in aorta. Plane strain condition is assumed, and in-plane displacement boundary conditions just enough to eliminate rigid body motion are imposed. A 100mmHg follower pressure is applied on the lumen surface.

For comparison, a FE model of comparable mesh density consisting of four-node mixed element [34] was introduced. Figure 8 shows the distributions of the von Mises stress from the two models. Since the nodes in the two models do not correspond, there is no effective way to make a node-wise comparison. But by a visual inspection, the stress results agree well. It should be noted that nodal stresses in the point cloud model are taken directly from the pointwise stress value without smoothing, and yet, the stress distribution appears to be reasonably smooth. In addition, as also demonstrated in previous works, the discrete method appears to be locking-free in the limit of incompressibility and no additional algorithmic treatment is necessary for dealing with nearly incompressible materials.

4.2 Dynamics test: transient response of a cantilever beam

While the behavior of DGM in static analysis has been thoroughly investigated in early studies [1, 2], the dynamics capability was tested only in modal analysis [23]. As a precursor to impact analysis, here we test the transient capability using the cantilever system in Figure 9(a). The beam has a length 10 and a width 0.2, loaded by a tip force $F = \sin 25t$. The material is elastic, with $E = 5 \times 10^4$, $\nu = 0.3$, and mass density $\rho = 1.0 \times 10^{-6}$. The transient response is obtained using the Newmark method of $\beta = 0.5$ and $\gamma = 0.5$. Figure

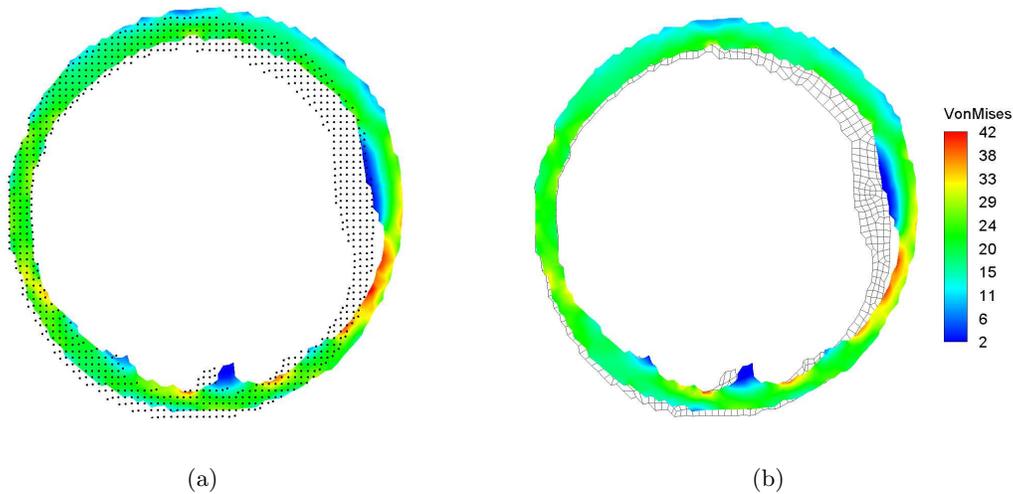


Figure 8: Von Mises stress in aorta cross section. (a) Point cloud model; (b) FEM model.

9(b) shows the tip displacement from DGM and FEM. It is evident that, while the phases are in good agreement, DGM over-predicts the tip displacement. This finding is consistent with the theoretical analysis on similar discrete methods [20, 21].

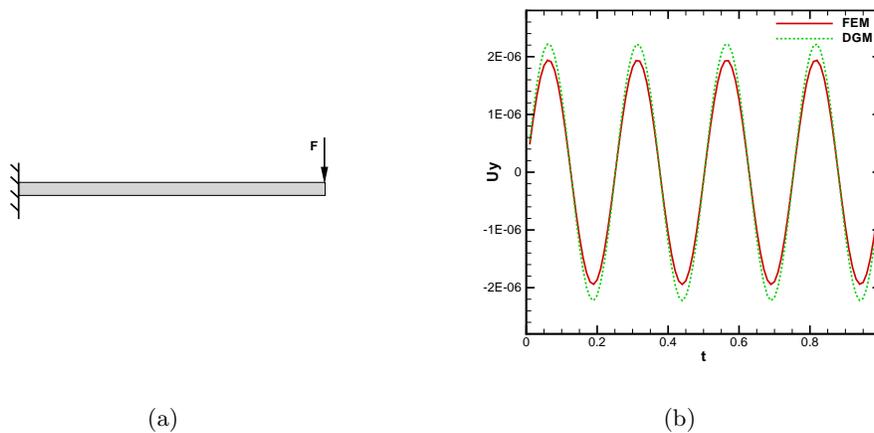


Figure 9: Cantilever beam subject to time dependant load: (a) Model; (b) Tip displacement.

4.3 3D large deformation test: twist of a cantilever beam

A cantilever beam subjected to large twist is simulated using both DGM and finite element method. As shown in Figure 10, the free end of the $8 \times 1 \times 1$ beam is subjected to an end-displacement that produces a rigid rotation of 360° . The beam is made of hyperelastic

material described by the energy function

$$W = \frac{\mu}{2}(I_1 - 2 \log J - 3) + \frac{\kappa}{2}(\log J)^2,$$

where $\mu = 192.3, \kappa = 288.5$. The two models have the same nodal density and the finite element model contains 270 brick elements. The deformed configurations of the beam are

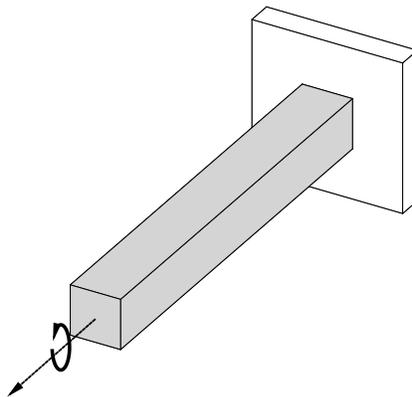


Figure 10: Three dimensional cantilever beam subject to an end-twist.

presented in Figure 11. As seen from the figure, the deformations predicted from the two methods agree well with each other. To provide a quantitative assessment, the torques as a function of the twisting angle are computed and compared in Figure 12. Evidently, the results are in excellent agreement.

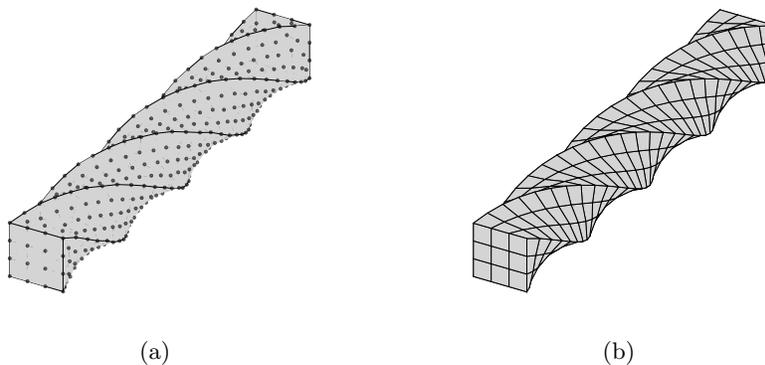


Figure 11: Deformed configurations at 360° twist: (a) DGM; (b) FEM.

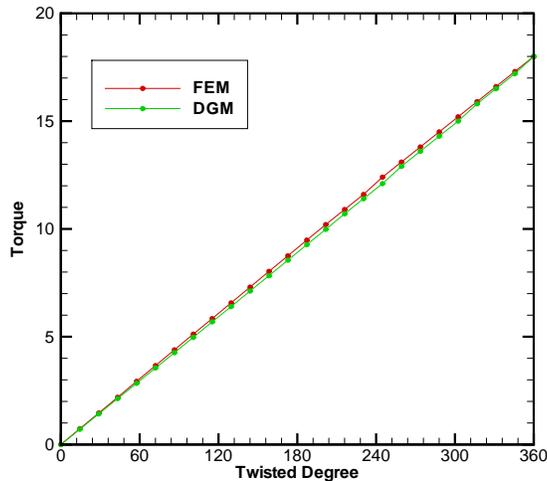


Figure 12: Comparison of torques.

4.4 Impact analysis: stress wave in skull

The propagation of stress wave in a three-dimensional skull under a point impact is presented. To obtain a point-cloud, a life-size plastic skull was scanned by a CT with 0.4 mm resolution between pixels and slices. The entire image stack contains 400 CT images. Each image has 512×512 pixels. A $4 \times 4 \times 4$ window is utilized in point extraction. Figure 13 shows different views of the three dimensional point-cloud. The registration is fully automatic.

Although it is possible to estimate density and elastic modulus of the material at any particular location with the CT data [35], we assume that the material is homogeneous, and isotropic linear elastic with the properties $E = 11$ GPa, $\nu = 0.35$, and $\rho = 1.6 \times 10^3$ Kg/m³. A transient pressure is applied over a small region in the forehead. Again, Newmark method is utilized for time integration. The predicted stress wave is depicted in Figure 14. Clearly, a propagating stress wave is captured. The problem was solved using the point cloud method alone, and no comparison was made with FEM.

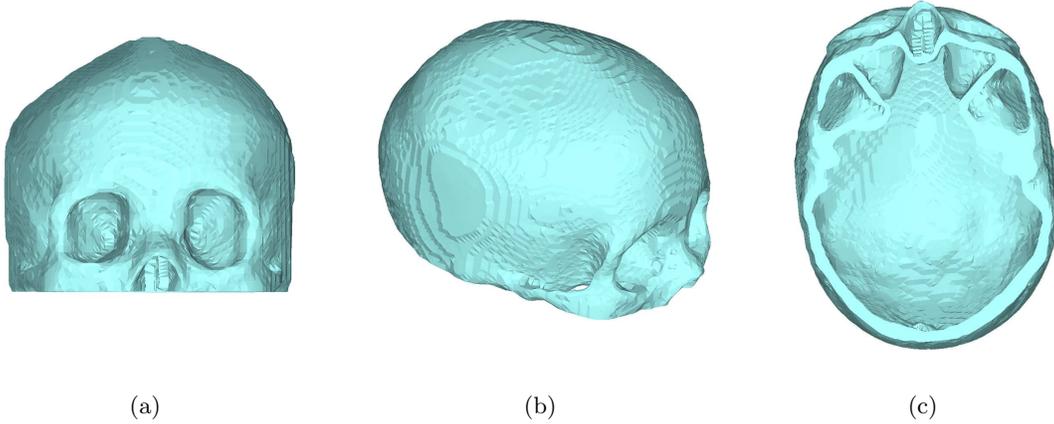


Figure 13: Three dimensional skull model extracted from CT images: (a) Front view; (b) Isometric view; (c) Bottom view.

5 Concluding Remarks

We introduced a point cloud method for the stress analysis in biological systems. The method takes as the geometric input a point cloud derived from medical images, and utilizes DGM as the mechanical solver. The latter computes the pointwise strain using discrete differentials involving the nodal displacements of a set of neighboring points. The neighboring relations as well as the point volumes are provided by Delaunay tessellation. Despite the utility of tessellation, the method is not element-based because element-wise assumed solution is never constructed. A distinct feature of this method is that the entire process can be carried out with minimal user interference or even fully automatically. This is significant for applications where a timely analysis is desired. We also demonstrated in this and previous studies that the discrete solver has comparable accuracy as low order finite elements.

A limitation of the present method lies in the use of Delaunay tessellation. Although utilized in this study, Delaunay tessellation is not a pre-requisite for the method. To set up the discrete equations, the essential information one needs is a neighboring relation and the point volume. These pieces of information can be inferred from medical images and therefore a truly point-based analysis is possible. Research in this direction is underway

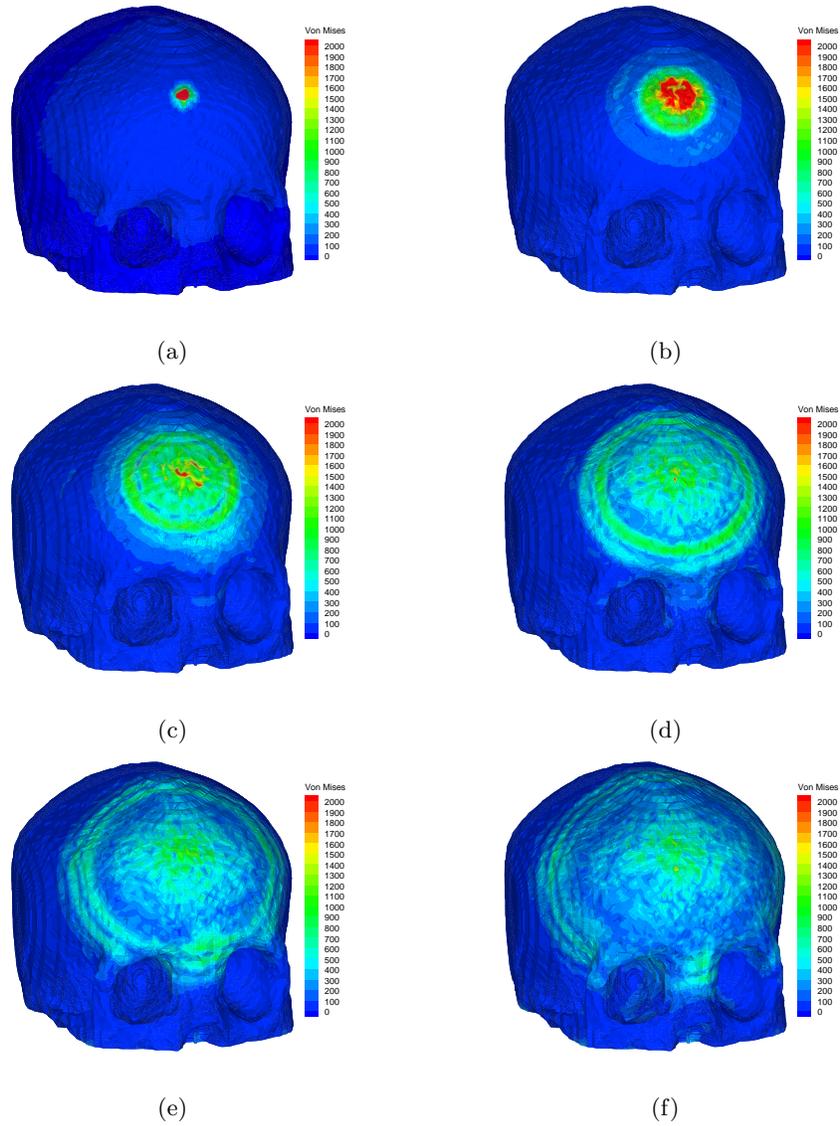


Figure 14: Von Mises stress wave propagation: (a) $t = 5.0E10^{-5}$ Sec.; (b) $t = 5.5E10^{-5}$ Sec.; (c) $t = 6.0E10^{-5}$ Sec.; (d) $t = 6.5E10^{-5}$ Sec.; (e) $t = 7.0E10^{-5}$ Sec.; (f) $t = 7.5E10^{-5}$ Sec.

and will be reported in a forthcoming paper.

Acknowledgment The work was partially supported by the National Science Foundation under the grant CMS 03-48194 to the University of Iowa.

References

- [1] J. Lu, J. Qian, and WM. Han. Discrete gradient method in solids mechanics. *International Journal for Numerical Methods in Engineering*, 74:619–641, 2008.
- [2] J. Lu and J. Qian. Discrete gradient method over polygon mesh. *International Journal for Numerical Methods in Engineering*, 78:505–527, 2009.
- [3] D. P. Fyhire, M. S. Hamid, R. F. Kuo, and S. M. Lang. Direct three-dimensional finite element analysis of human vertebral cancellous bone. *Transactions of 38th Annual Meeting of Orthopaedic Research Society*, 1992.
- [4] S. J. Hollister, J. M. Brennan, and N. Kikuchi. A homogenization sampling procedure for calculating trabecular bone effective stiffness and tissue level stress. *Journal of Biomechanics*, 27:433–444, 1994.
- [5] J. H. Keyak, J. M. Meagher, H. B. Skinner, and C. D. Mote. Automated three-dimensional finite element modelling of bone: a new method. *Journal of Biomedical engineering*, 12:389–397, 1990.
- [6] J. H. Keyak, , and H. B. Skinner. Three dimensional finite element modelling of bone: effects of element size. *Journal of Biomedical engineering*, 14:483–489, 1992.
- [7] J. H. Keyak, , M. G. Fourkas, J. M. Meagher, and H. B. Skinner. Validation of an automated method of three dimensional finite element modelling of bone. *Journal of Biomedical engineering*, 15:505–509, 1993.

- [8] B. van Rietbergen, H. Weinans, R. Huiskes, and A. Odgaard. A new method to determine trabecular bone elastic properties and loading using micromechanical finite-element models. *Journal of Biomechanics*, 28:69–81, 1995.
- [9] L. W. Marks and T. N. Gardner. The use of strain energy as convergence criterion in the finite element modelling of bone and the effect of model geometry on stress convergence. *ASME Journal of Biomedical Engineering*, 15:474–476, 1993.
- [10] D. L. A. Camacho, R. H. Hopper, G. M. Lin, and B. S. Myers. An improved method for finite element mesh generation of geometrically complex structures with application to the skullbase. *Journal of Biomechanics*, 30:1067–1070, 1997.
- [11] S. K. Boyd and R. Müller. Smooth surface meshing for automated finite element model generation from 3D image data. *Journal of Biomechanics*, 39:1287–1295, 2006.
- [12] D. P. Frey, B. Sarter, and M. Gauthere. Fully automated mesh generation for 3-d domains based upon voxel data sets. *International Journal for Numerical Methods in Engineering*, 37:2735–2753, 1994.
- [13] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 21:163–169, 1987.
- [14] R. Müller and P. Rügsegger. Three-dimensional finite element modeling of non-invasively assessed trabecular bone structures. *Journal of Medical Engineering and Physics*, 17:126–133, 1995.
- [15] F. H. Croom. *Basic concepts of algebraic topology*. Springer, Berlin, 1978.
- [16] J. Bonet and A. J. Burton. A simple average nodal pressure tetrahedral element for incompressible and nearly incompressible dynamic explicit applications. *Communications in Numerical Methods in Engineering*, 14:437–449, 1998.
- [17] C. R. Dohrmann, M. W. Heinstein, J. Jung, S. W. Key, and W. R. Witkowski. Node-based uniform strain elements for three-node triangular and four-node tetrahedral

- meshes. *International Journal for Numerical Methods in Engineering*, 47:1549–1568, 2000.
- [18] J. Bonet, H. Marriott, and O. Hassan. An averaged nodal deformation gradient linear tetrahedral element for large strain explicit dynamic applications. *Communications in Numerical Methods in Engineering*, 17:551–561, 2001.
- [19] M. A. Puso and J. Solberg. A stabilized nodally integrated tetrahedral. *International Journal for Numerical Methods in Engineering*, 67:841–867, 2006.
- [20] G. R. Liu, K. Y. Dai, and T. T. Nguyen. A smoothed finite element method for mechanics problems. *Computational Mechanics*, 39:859–877, 2007.
- [21] G. R. Liu, T. T. Nguyen, K. Y. Dai, and K. Y. Lam. Theoretical aspects of the smoothed finite element method. *International Journal for Numerical Methods in Engineering*, 71:902–930, 2007.
- [22] N. X. Hung, S. Bordas, and N. D. Hung. Smooth finite element method: convergence, accuracy and properties. *International Journal for Numerical Methods in Engineering*, 74:175–208, 2007.
- [23] J. Qian and J. Lu. A stabilized formulation for discrete gradient method. *Communications in Numerical Methods in Engineering*, DOI:10.1002/cnm.1337, 2009.
- [24] P. Krysl and B. Zhu. Locking-free continuum displacement finite elements with nodal integration. *International Journal for Numerical Methods in Engineering*, 76:1020–1043, 2008.
- [25] J. S. Chen, C. T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.
- [26] J. S. Chen, S. Yoon, and C. T. Wu. Non-linear version of stabilized conforming

- nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 53:2587–2615, 2002.
- [27] J. W. Yoo, B. Moran, and J. S. Chen. Stabilized conforming nodal integration in the natural-element method. *International Journal for Numerical Methods in Engineering*, 60:861–890, 2004.
- [28] M. A. Puso, J. S. Chen, E. Zywickz, and W. Elmer. Meshfree and finite element nodal integration methods. *International Journal for Numerical Methods in Engineering*, 74:416–446, 2008.
- [29] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22:469–483, 1996.
- [30] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *Information Theory*, 29:551–559, 1983.
- [31] H. Edelsbrunner and E. P. Mucke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43–72, 1994.
- [32] E. Cueto, B. Calvo, and M. Doblare. Modelling three dimensional piece-wise homogeneous domains using the α -shape-based natural element method. *International Journal for Numerical Methods in Engineering*, 54:871–897, 2002.
- [33] E. Cueto, J. Cegonino, B. Calvo, and M. Doblare. On the imposition of essential boundary conditions in natural neighbour galerkin methods. *Communications in Numerical Methods in Engineering*, 19:361–376, 2003.
- [34] J. C. Simo and M. S. Rifai. A class of mixed assumed strain methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering*, 29:1595–1638, 1990.
- [35] D. R. Cater and W. C. Hayes. The compressive behavior of bone as a two-phase porous structure. *Journal of Bone Joint Surgery*, 59:954–962, 1977.