Equation Chapter 4 Section 1 *[Handbook of Real-World Applications in Modeling and Simulations],* edited by [John A. Sokolowski, Catherine M. Banks].
ISBN 0-471-XXXXX-X

Chapter 7

A Review of Mesh Generation for Medical Simulators

by Michel A. Audette, Andrey Chernikov, and Nikos Chrisochoides

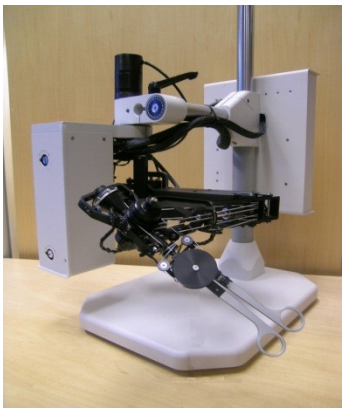
Introduction and Learning Objectives

Medical simulation is the application of computers to synthesizing an anatomical response to a simulated therapy. In particular, surgery simulation uses a software program to synthesize tissue response to virtual surgical tools, typically (but not exclusively) a mechanical response to cutting or manipulation. This behavior can be thought of as a trade-off between material fidelity and computation time, whose weighted emphasis on one or the other can be characterized as a spectrum. At one end of the spectrum we have **predictive simulation**, which consists of highly faithful off-line computations used by expert surgeons to predict the outcome of, and optimize, an intervention, on the basis of an anatomical model of the patient derived from that individual's preoperative image dataset. At the other end of the spectrum, the objective of **interactive simulation** is to offer a means of training surgical residents in order to improve their skill without risk to a real patient, by way of a **haptic device** manipulated by the user to position a virtual surgical tool, while producing a force feedback that simulates tissue resistance and a real-time graphical rendering of an anatomical model at that point in simulated time. Figure 1 illustrates some commonly used haptic devices. Typically, the biomechanics engine used to achieve a response at near-haptic rates (some interpolation is feasible for

haptic rates of 500 Hz or more), in the context of interactive simulation, is less constitutively faithful than that of predictive simulation, although much recent work is devoted to reconciling the conflicting requirements of interactivity and material faithfulness.

Irrespective of whether a medical simulator emphasizes interactivity or predictive computation, the simulation requires an anatomical model on which to carry out its synthesized therapy. For most clinical applications, such a model is not drawn with 3D CAD software, but rather extracted by image analysis from a patient dataset. As a result, the starting point for this model is one or more MR or CT volumes, which in the multi-modal case can be co-registered and resampled, which leads to a volumetric scalar or

(a)



(b)

Figure 1. Commercial haptic devices: (a) Sensable's 6 degree-of-freedom (d.o.f.) Phantom 6S/1.5 (1); (b) MPB Technologies' 7 d.o.f. Freedom 7S (2).

vector image, typically of several hundred voxels along each axis. For example, a 1mm isotropic MR image of the head is usually at least 256 x 256 x 256, which equates with more than 16 million voxels, which in turn precludes efficient computation directly based on raw or segmented image data. In addition, many biomechanical engines require the

decomposition of a geometrically complex body into simple shapes, e.g.: elements, given that the computation itself is typically a matrix equation based on simple, well understood elemental expressions. These requirements, computational efficiency and geometric decomposition, motivate the need for a representation of the anatomy in terms of simple shapes, such as triangles and tetrahedra, a process, which is idealized in 2D in figure 2 and which we describe as **mesh generation**. It is worth noting that in the mesh generation community, the generation of tetrahedra corresponds to **unstructured mesh generation** as contrasted from structured meshes which are typically comprised of hexahedra. The latter elements are not generally used in medical simulation, because this meshing approach requires a significant amount of user interaction (in contrast with tetrahedral

Figure 2. Basic processing pipeline idealized in 2D: (a) raw image data, typically MR and/or CT; (b) segmentation of relevant tissues, e.g. white and grey matter, corticospinal fluid, tumor; (c) tissue boundary meshing: segmentation results of (b) are too dense for practical simulation, requiring a geometrical representation; (d) tissue boundary meshing is completed by inserting internal, or Steiner, points and linking them by edges (and in 3D, faces) and triangles (in 3D, tetrahedra); (e) by adding a camera, virtual tool, and virtual lighting, as well as biomechanics engine, we can apply the anatomical model to a simulation

meshing, which can be automated). Moreover, the subdivision of a hexahedron does not reduce to more hexahedra, which limits their applicability to interactive simulation, whereas a tetrahedron ultimately is divisible into more tetrahedra. However, hexahedra tend to produce stable numerical results and have been applied to predictive simulation.

One way in which hexahedra can be applied to patient-specific simulation involves the warping of a canonical anatomy to the patient's anatomy, which is feasible if the tissue topology is preserved across individuals and the non-rigid transformation well defined.

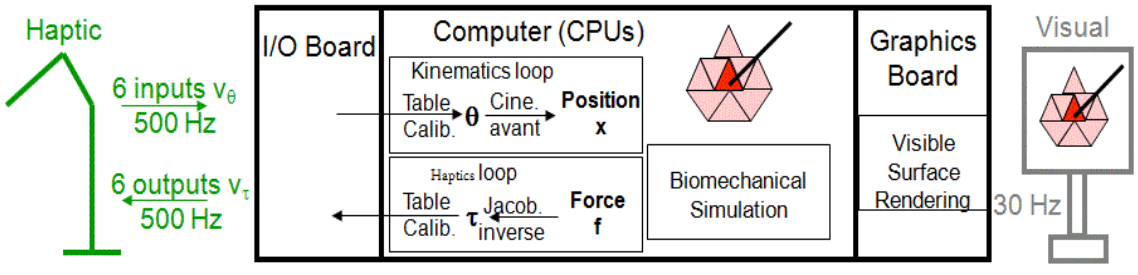
Finally, recent surgery simulation research emphasizes so-called **meshless methods**, these usually require a mesh of some sort as a starting point, if only to identify which points in anatomical volume neighbor each other, and in any event, visual and haptic rendering entails an efficient surface meshing of a relevant isosurface of their numerical solution.

The objective of this chapter is to introduce the reader to leading methods in producing anatomical meshes for medical simulation, in a manner mindful of practical clinical requirements, while also suggesting feasible improvements that are needed for simulation to address these requirements.

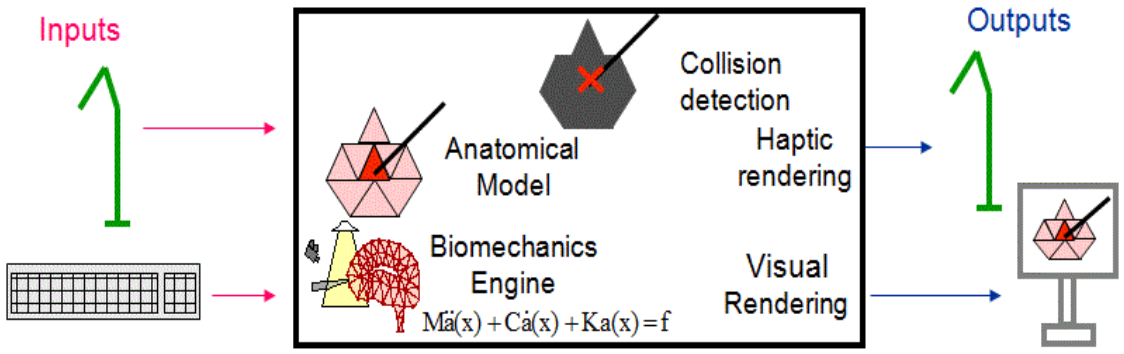
Background - A survey of relevant biomechanics and open-source software

Architecture of an Interactive Medical Simulator

The conflicting requirements of interactive surgery simulation become apparent by considering the architecture of such as system, as well as demands in terms of anatomical modeling and biomechanics needed for clinical realism. Figure 3 illustrates simplified



(a)



(b)

Figure 3. Illustration of interactive surgical simulation architecture: (a) hardware and (b) software.

hardware and software architectures of an interactive simulator. From a hardware standpoint, the haptic device is sampled in real-time at haptic rates (500-1000 Hz) (3), and the joint voltages are input to the computer, by way of an analog input/output board, then converted to joint angles and to position, by forward kinematics (4), in a **kinematics loop** ideally running at that same rate. In addition, it is desirable to convert a feedback force, computed by the biomechanics engine, to joint torques and output voltages through a **haptic loop** running at the same interactive rate. In addition, visual feedback is provided at 30-60 Hz, in a manner that makes use of the **graphical processor unit** (GPU). In addition to rendering anatomical surfaces at video rates, the GPU is increasingly used to accelerate computations other those related to visualization, a subject that we will revisit shortly.

The software architecture features the following:

- an **anatomical model**, typically composed of tetrahedra, encapsulating the anatomy over simulated time;
- a **biomechanics engine** that synthesizes a tissue response to a virtual gesture;

- a **collision detection algorithm** that efficiently determines where the interaction takes place;
- and software engines that respectively handle **haptic** and **visual rendering**.

Mechanics of Tissue Manipulation in Medical Simulation

Biomechanical modeling of tissues is often approached by finding an numerical solution for the displacements, deformations, stresses and forces, as well as possibly other states, such as hydrostatic pressure, in relation to a history of "loading". The approaches for estimating or simulating biomechanicaldeformations are characterized by a trade-off between computational efficiency and material fidelity, and the nature of this trade-off can be viewed as a spectrum between two poles. At the fast but materially approximative end of the spectrum lie **mass-spring systems**. At the other end of the spectrum, computationally slower but more descriptive, we have classical **finite element method (FEM)**, which can characterize even large (finite) deformations and nonlinear elasticity.

A mass-spring system (5) is an approximation of a biomechanical system as a collection of point masses connected by elastic springs, and is derived from the field of computer animation. A mass-spring system is characterized by each node i, having a mass m_i and position $\mathbf{x}_i$, and being imbedded in a mesh where each edge coincides with a spring k. Each node is subject to an equation of the form

$$\text{mid2xidt2+}\gamma\text{ idxidt+gi=fi where} \qquad (1)$$

$$\text{gi(t) = j}\in\text{Nisk and where sk = ckekrkrk .} \qquad (2)$$

In this equation, $\mathbf{s}_k$ represents the force on the kth spring linking the node i to a neighbouring node j. This force is a function of the vector separation of the nodes $\|r_{k=xj-xi}\|$,

of the deformation of the spring ek= ||rk|| - lk and of the characteristics of the spring: its natural length l_k, its stiffness c_k, and its velocity-dependent damping γ_i. The quantity fi is the net external force acting on node i, which may include a surgical tool or the effect of gravity.

The displacement finite elements method numerically solves for unknown displacements, deformations, stresses, forces and possibly other variables of a solid body. An exact solution would require force and momentum equilibrium at all times everywhere in the body, expressed as St dS +Vf dV = 0 and Sx ×t dS +Vx ×f dV = 0 respectively. However, the finite element method replaces this requirement with the weaker one that *equilibrium must be maintained in an average sense over a finite number of divisions of the volume of the body*. These divisions, or *elements*, are simple shapes such as triangles and rectangles for surfaces, and tetrahedra and hexahedra for volumes, and the method relies on estimating the displacement at their vertices, or *nodes*. Discretizing equilibrium equations is based on using Gauss' theorem to restate them as a single integral, called the **Principle of Virtual Work** (PVW), and expressing this integral in a *weak form*, where the integrand is multiplied by a *test function* with compact support over the element, which leads to a linear system featuring a sparse matrix.

The Principle of Virtual Work states that for a virtual displacement applied to the system, static equilibrium requires that the external virtual work must equal the internal work done within the element. The tissue volume is defined as Ω and is subject to boundary conditions. Assuming Cartesian coordinates, the displacement at the node i of a given element is labeled ai= ui vi wiT, while the displacement at any point in Ω is expressed u= ux,y,z v(x,y,z) w(x,y,z)T. The latter displacement is fully determined by the nodal

displacements and by the **shape functions** that govern the interpolation between them. For a tetrahedral element, we have u = INiINjINkINl ae=Nae, where Ni = 1 at node (xi yi zi) but zero elsewhere, and so on, and where a12×1e= aiajakalT is comprised of all nodal displacements within a given tetrahedral element. For a small strain assumption, the relationship between strain ϵ and displacement u is given by the following expression:

$$\epsilon = \epsilon x \epsilon y \epsilon z\, \gamma xy\, \gamma yz\, \gamma xz = \partial u \partial x \partial v \partial y \partial w \partial z \partial u \partial y + \partial v \partial x \partial v \partial z + \partial w \partial y \partial u \partial z + \partial w \partial x =$$

$$BiBjBlBm\ ae \equiv Bae \quad (3)$$

where the shape functions Bi are obtained by deriving INi, and B is designated the *elemental Shape Matrix*.

Foregoing for the sake of brevity the details of the derivation of FEM formulation from PVW, a linear isotropic material whose constitutive properties is expressed as a matrix D(λ , μ), the elemental equilibrium expression is given by (6):

qe= Keae+ fe where Ke=VeBTDBdV and
where fe=-VeNTbdV-VeBTDϵ0dV + VeBT σ 0dV . (4)

We designate *nodal forces* qe, which are statically equivalent to boundary stresses and body forces comprising boundary conditions, *concentrated loads* b acting on the body, and σ(ae, B, σ 0) the *elemental stress*, which is estimated based on the constitutive properties of the tissue.

Summing expressions (4) over the whole volume leads to the familiar FEM formulation:

$$Ka = f, \quad (5)$$

where **Ke** and fe are the *elemental stiffness matrix* and *elemental force vector*, while **K, a** and f represent the system's **stiffness matrix**, having sparse structure, as well as *nodal displacement* and *force vectors*. The unique solution requires one or more boundary

conditions, which modify the stiffness matrix and make it nonsingular. For some dynamic systems, this equation may be modified to further include mass M and damping C effects:

$$Ma + Ca + Ka = f, \qquad (6)$$

The Principle of Virtual Work can be seen as equating internal deformation energy with external energy generated by external forces over a domain Ω:

$$\Omega \delta U \, d\Omega = \Omega f^T \delta u \, d\Omega \qquad (7)$$

where δ indicates the *variation* of a quantity. For material nonlinearity and large geometric deformation, it is common to solve finite elements expressed in terms of a Strain Energy Density (SED) function U, which is a material-related function of invariants of the *Cauchy-Green* deformation tensor $C=F^TF$, where F is the *deformation gradient*. These invariants, which are unchanged under rigid transformation, include $I1 = trC$, $I2 = (trC)^2-(trC^2)$, $I3 = detC$, as well as the Jacobian $J = detF$, where *tr* and *det* represent the trace and determinant.

Increasingly, researchers are reconciling the conflicting requirements of fidelity and efficiency by using algorithmic refinements and high-throughput or parallel hardware such as the graphical processor unit (GPU). Amongst the refinements of the first type include the *Total Lagrangian Explicit Dynamics* (TLED) formulation, featuring an explicit solution of nonlinear finite elements that uses shape functions precomputed (prior to the simulation) from undeformed coordinates (7), and the *reduced basis* adaptation of the TLED (8). Another recent technique, the *Multiplicative Jacobian Energy Decomposition* (MJED) algorithm (9), allows matrix precomputations to be performed

through a decomposition of the strain energy as a sum of terms that are functions of various invariants, so as to avoid matrix inversions and complex derivative expressions.

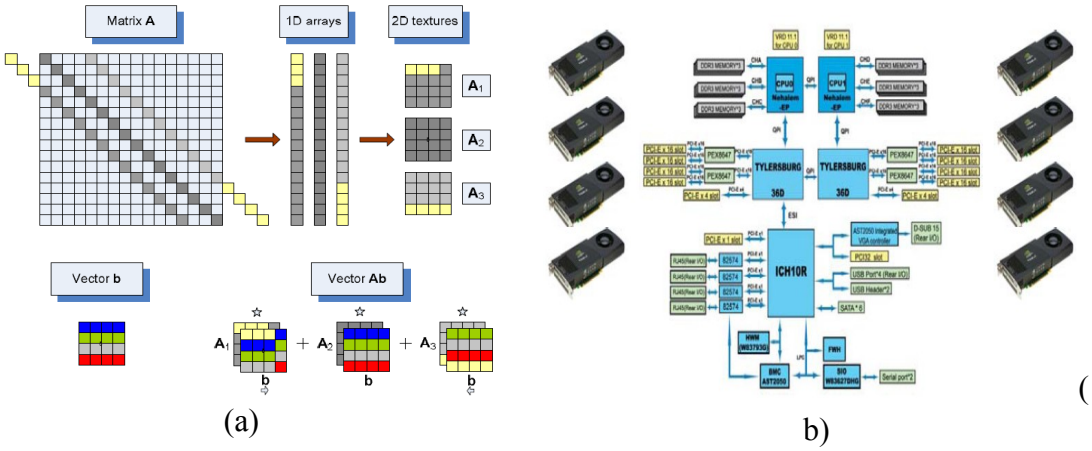


Figure 4. GPU acceleration: (a) Idealized representation of banded matrix in terms of arrays and 2D suitable for texture hardware-based acceleration, reproduced with permission from (10). (b) Architecture of state-of-the-art multi-GPU computer, featuring 8 GPU boards on the same backplane (11) .

Examples of hardware-based acceleration include the application of GPUs to the computation of the TLED (12). The basic idea of GPU acceleration is depicted in figure 4(a): to implement the multiplication of a sparse banded matrix **A** by a vector **b** (10), each band of the matrix is represented by a one-dimensional array that in turn is stored in a 2D texture on the GPU ($\mathbf{A}_1$-$\mathbf{A}_3$). Zeroes are padded at the front or back depending on the position of the band in the matrix. The multiplication **Ab** is computed through a pixel-wise multiplication of each band in **A** with **b**, and the products are added to form the final result. In practice, GPU-based computation produces an order-of-magnitude acceleration over the CPU. Moreover, recent multi-GPU computers (11), such as that depicted in

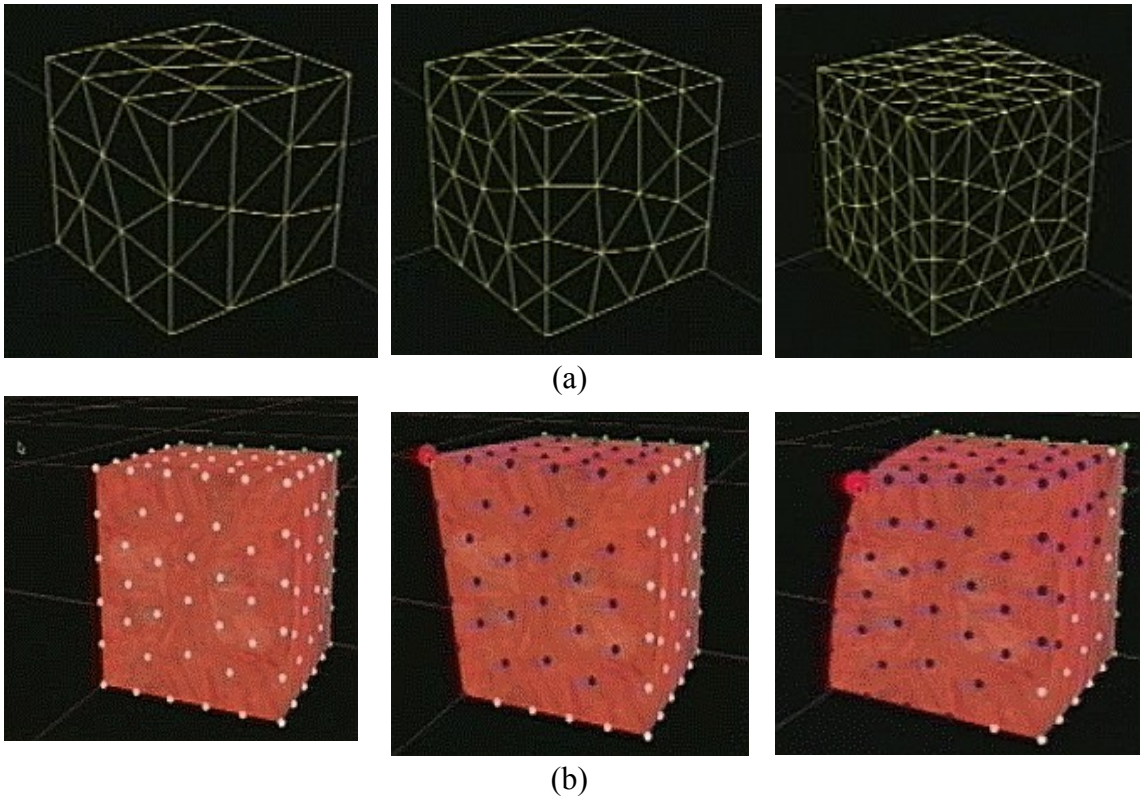


Figure 5. Multi-grid finite elements, computed on a cube with multi-resolution meshing. (a): coarse, medium and fine grids. (b): large-deformation real-time haptic interaction, courtesy of X. Wu (RENCI, Chapel Hill, NC).

figure 4(b), offer promise to scale at low cost this type of acceleration with a large number of GPUs present on one motherboard, in part through judicious choice of PCI-e bus switches.

One of the conflicting requirements that characterize interactive surgery simulation is the computation of a constitutively faithful behavior (e.g.: nonlinear FEM) over relatively large organs, such as the brain, while also modeling fine critical tissues whose preservation is central to the simulation. In short, interactive nonlinear biomechanics (at near-haptic rates) tend to presuppose relatively large elements, with edges in the 10-30 mm range, while small critical tissues are typically comprised of elements of 1mm scale

or even less. A single resolution FE system that would imbed both types of elements could easily be subject to ill conditioning, with dire consequences for its convergence properties. Consequently, these conflicting requirements tend to favor a *multi-grid FE* approach (13), where an anatomical model consists of several meshes of different resolutions, each solved as a separate FE system. Moreover, the coarse level is solved efficiently, due to the limited complexity of its system and to its large critical time step, and then is used to assist middle and/or fine level systems to converge more quickly than otherwise, as depicted in figure 5. Moreover, while Wu suggested medium- and fine-level meshes that spanned the whole volume of interest, in practice only a subvolume is of clinical interest, typically comprising the pathology and the path to it, which limits the complexity of these corresponding FE systems.

Mechanics of Tissue Cutting and Resection in Medical Simulation

Amongst techniques for synthesizing a cut or resection in interactive surgical simulation,

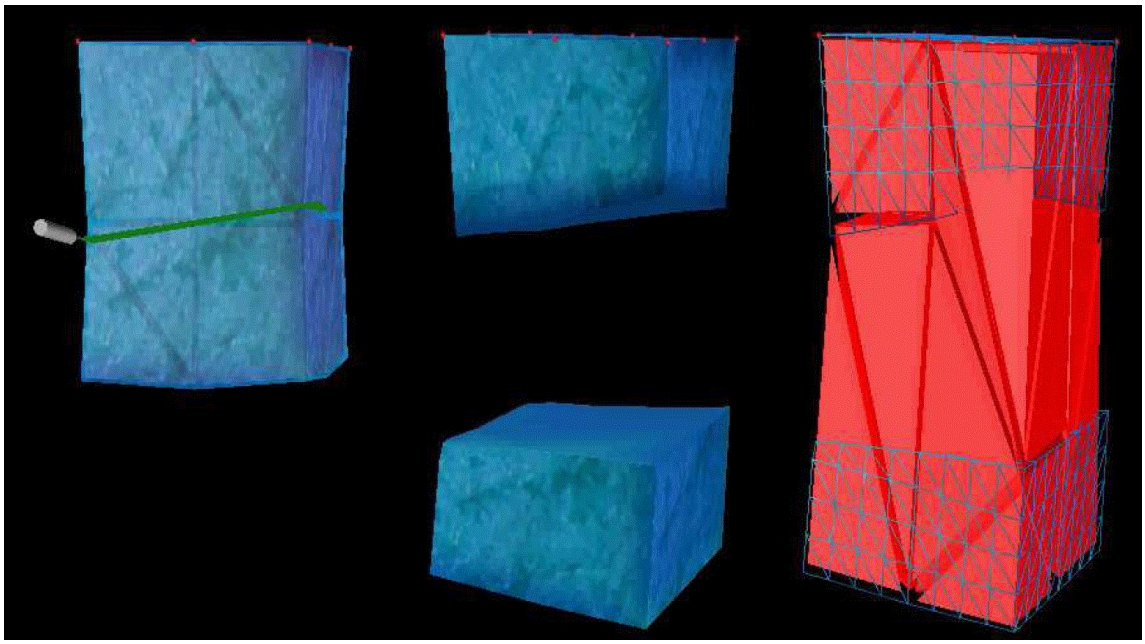


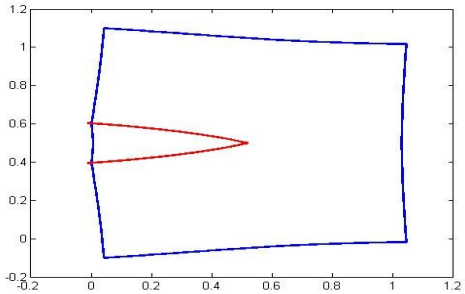
Figure 6. Application of XFEM for simulating a cut to a tetrahedral mesh, courtesy of L.

Jerabkova (Stryker, Germany): (left) – deformable cube fixed at the top which is cut interactively; (middle) the dissected volume falls under the effect of gravity; (right) the underlying elements (usually invisible to the user) are visualized: the enriched elements are stretching between both dissected parts (14).

we cite those based on a tetrahedral or triangulated mesh, such as the **extended finite element method (XFEM).** In this formulation, the shape functions Bi of expression (3) admit a possible discontinuity (14), as depicted in figure 6.

The main alternative to XFEM is based on the meshless formulation, which involves a system of equations derived from point-centered shape functions, as shown in figure 7. Meshless methods discretize partial differential equations, including continuum mechanics expressions, through shape functions with compact support defined on a local cloud of points (or nodes), rather than on an non-overlapping elements. Despite the name that implies that no mesh is involved, the latter approach requires a preliminary meshing that establishes neighboring vertices in the point cloud used in the discretization.

(a) (b)



(c) (d)

Figure 7. Application of meshless methods to simulating cutting by fracture mechanics, courtesy of S. Bordas (Univ. Cardiff, Wales): (a) illustration of point cloud and subdomains associated with each point (2D illustration); (b) point-based test functions, for constructing the weak form of equilibrium expression, and weight functions for

constructing shape functions for interpolation of the point cloud; (c) a propagating crack leads to the insertion of new points; (d) enriched, discontinuity-admitting weight functions (not shown) lead to a numerical solution of a simulated fracture.

Open-source Resources in Medical Simulation

Increasingly, researchers have come to rely on open-source software for implementing state-of-the-art applications. Resources available include in particular the Simulation Open Framework Architecture (15)**,** which features a highly modular suite of C++ classes that implement the basic components of an interactive

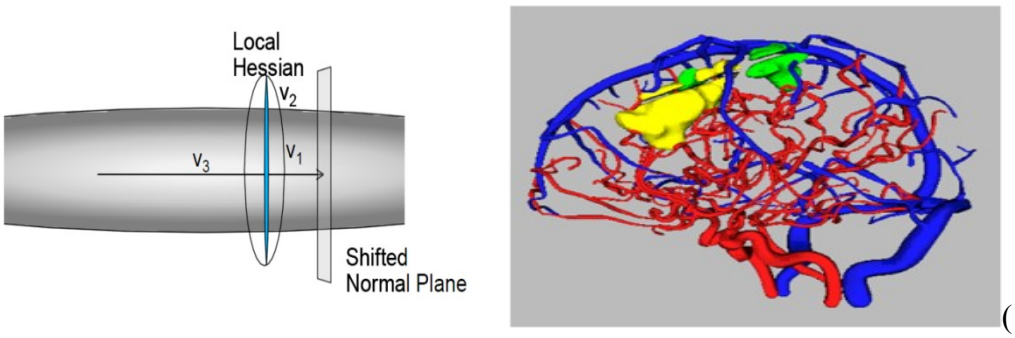
surgery simulator. This toolkit provides separate visualization and biomechanics engines, support for haptic interaction, and collision detection pipeline. In particular, the separate handling of visualization and biomechanics makes it possible to have a rendered triangulated anatomical boundary, or *Visual Model*, that is significantly denser than the tetrahedral *Mechanical Model* that is input to the biomechanics engine, while providing a *Mapping* class that updates the former on the basis of real-time deformations undergone by the latter. A number of other open-source toolkits emphasize haptics and mesh-spring models, such as CHAI 3D (16) and H3D (17).

Finally, an unstated assumption in this discussion is that *we have a starting point of a segmented image volume*, which may be achieved interactively by an expert anatomist, or (semi-)automatically with segmentation algorithm. In general elements represent a homogeneous tissue volume, so it is assumed that each tissue volume is known prior to meshing; because of partial volume effects and imaging noise, the relationship between image intensity and tissue composition is not straightforward: in general, a three-tissue volume does not produce three well separated spikes in the image histogram, but rather

three overlapping Gaussian modes. As a result, the mapping between image intensities and tissues has to be established. The segmentation problem is a highly active area of research within the medical imaging community, and an extensive survey is beyond the scope of this paper. We suggest the following surveys to the reader (18) (19) as well as leading open-source efforts such as Slicer3D (20).

Segmentation methods fall into three categories: *region*, *boundary*, and *atlas*-based.

- Region-based methods identify tissues by considering the intensity of each voxel, and include region growing, clustering, and Bayesian approaches such



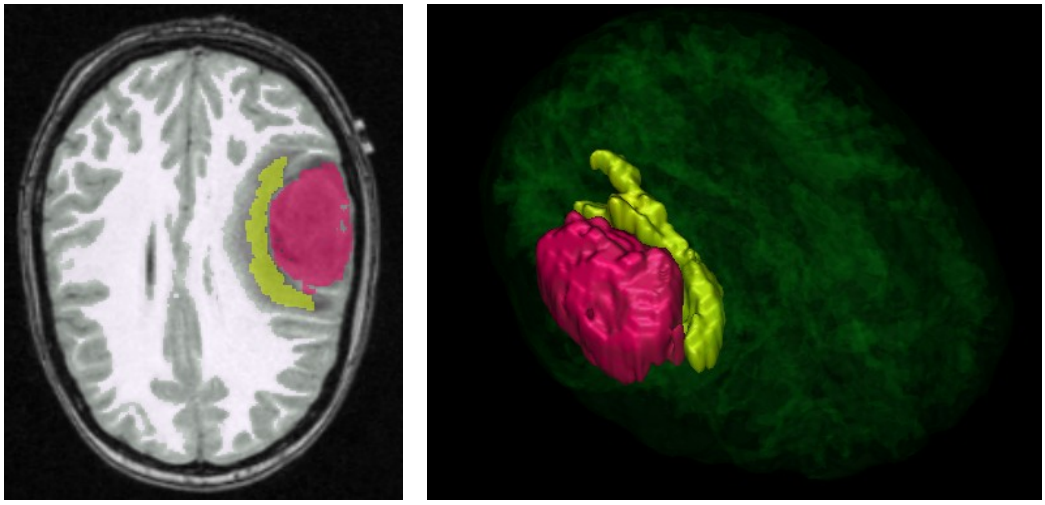
(a) (b)

Figure 8. Region-based approach to detecting blood vessels in MR angiography data: (a) orientation of principal axes of local Hessian-based 3D ridge operator; (b) application to identifying cerebral blood vessels (21), courtesy of S. Aylward (Kitware), with labeled tumor and oedema also depicted.

as Expectation Maximization. One illustration of a region-based approach is shown in figure 8. This technique combines a tube-like structure detection operator, based on an eigenanalysis of the local Hessian of the image, and medial axis tracking of these locally bright tubes that coincide with blood vessels, which tracks in the direction the eigenvector corresponding to the smallest of three eigenvalues (21).

- Boundary-based segmentation methods identify the anatomical boundary, and include active surface models whose dynamic behavior encapsulated in a partial differential equation in time and space or a physical equilibrium expression that features a number of pseudo-forces. These generally correspond respectively to *implicit,* or *level set-based* (22), surface models and *explicit, physically based* models that include the *simplex* mesh (23).

  These surface models may also incorporate a component that accounts for *shape statistics* (24), which nudges the surface shape towards consistency with population statistics of the boundary of an anatomical structure. The simplex surface model is closely related to the surface meshing problem, which will be revisited in Modeling Paradigms section.



(a) (b)

Figure 9. Atlas-based minimally supervised approach (25) to detecting tumors in magnetic resonance imaging data: (a) axial slice of MRI data with segmentation results of tumor and oedema; (b) surface rendering of tumor and oedema overlaid on volume rendering of the brain, courtesy of M. Prastawa (Univ. Utah).

- Finally, atlas-based methods use a digital atlas of a structure or organ, which makes explicit various tissue or functional classes and which is warped to patient data (26). By proceeding this way, we can express tissue classification as a non-rigid registration exercise, which in turn may require the entry of a seed point from a sub-volume where the relationship with the atlas is undefined, such as a brain tumor (25), shown in figure 9.

Theory - The Impact of Element Quality and Size on Simulation

In general, the objective of the mesh generation stage is to produce a description of the anatomy suitable for numerical analysis and that consists of high-quality elements, where each element typically overlaps one type of tissue, and where the quality of an element correlates with the absence of edges of small relative size (in relation to the other edges of the element) and of small **dihedral angles** (defined as the angle between two planes subsuming two triangular faces). A characterization of element quality, as it relates to finite elements, was proposed in (27) and will be summarized in this section.

For finite elements applications, the **condition number** of the stiffness matrix should be kept as small as possible. ***Poorly conditioned matrices affect linear equation solvers by slowing down their convergence or introducing large round-off errors in the result.*** Element shape highly influences matrix conditioning, with small angles being deleterious to the result. The relationship between the shape of the element and matrix conditioning depends on the partial differential equation being solved, as well as the basis and test functions used to discretize it. Given a *d*-dimensional problem, where in the tetrahedral

case *d=3*, the finite element method builds, for each i^{th} element, a $(d+1)\times(d+1)$ *elemental stiffness matrix* K_i. These matrices are assembled into a global stiffness matrix K whose size is determined by the total number of elements. The convergence of the FE problem is made worse by K's condition number $\kappa = \lambda_{maxK} \lambda_{minK}$, where λ_{maxK} and $\lambda_{minK}$are the largest and smallest eigenvalues of K. While the eigenvalue $\lambda_{minK}$is not strongly influenced by elemental shape, λ_{maxK} ***can be made arbitrarily large by a single badly shaped element.*** For the *i*th tetrahedral element, its elemental stiffness, labeled Ke in expression (4), is given by the following expression, where we drop the *e* superscript for simplicity:

$$
Ki = A
\begin{matrix}
\omega 1\cdot \\
\omega 1 \\
\omega 1\cdot \\
\omega 2 \\
\omega 1\cdot \\
\omega 3 \\
\omega 1\cdot \\
\omega 4 \\
\omega 2\cdot \\
\omega 1 \\
\omega 3\cdot \\
\omega 1 \\
\omega 2\cdot \\
\omega 2 \\
\omega 3\cdot \\
\omega 2
\end{matrix}
$$

ω 2·

ω 3

ω 3·

ω 3

ω 2·

ω 4

ω 3·

ω 4

ω 4·

ω 1

ω 4·

ω 2

ω 4·

ω 3

ω 4·

ω 4 (8)

where in expression (8), ω kp= VkV denotes the **barycentric coordinates** of a point *p* within the i^{th} tetrahedral element t_i, where V represents the volume of this element, and Vk*(p)* represents the volume of the new tetrahedron formed by replacing vertex v_k by *p*. The *gradient of the barycentric coordinate* is

ω k(*p*) =

ω k, *which does not vary with p*, is a function of the *altitude* of each vertex a_k. As shown in figure 10, we label f_k the *(d-1)*-dimensional face t_i opposite vertex v_k. The altitude a_k of this vertex v_k is the *shortest distance from it to the plane that includes* f_k. The gradient

ω_k is a vector of length $1/a_k$, directed into t_i in a manner orthogonal to f_k. This development leads to the following expression for the element stiffness matrix:

$$K_i = 16 \left(\sum_{1 \neq i<j \neq 1} l_{ij} \cot \theta_{ij} - l_{34} \cot \theta_{34} \right) - l_{24} \cot \theta_{24} - l_{23} \cot \theta_{23} - l_{34} \cot \theta_{34} - l_{24} \cot \theta_{24} \left(\sum_{2 \neq i<j \neq 2} l_{ij} \cot \theta_{ij} - l_{14} \cot \theta_{14} \right) - l_{14} \cot \theta_{14} \left(\sum_{3 \neq i<j \neq 3} l_{ij} \cot \theta_{ij} \right) - l_{13} \cot \theta_{13} - l_{12} \cot \theta_{12} - l_{23} \cot \theta_{23} - l_{13} \cot \theta_{13} - l_{12} \cot \theta_{12} \left(\sum_{4 \neq i<j \neq 4} l_{ij} \cot \theta_{ij} \right), \quad (9)$$

where the variables l_{mn} represent the lengths of the edges between all vertex pairs v_m-v_n, while the variables θ_{mn} represent the corresponding dihedral angles about those edges. If a dihedral angle approaches 0 degrees, its cotangent approaches infinity, and so does the resulting large eigenvalue $\lambda_{\max K}$. In other words, flat tetrahedra, with one or more dihedral angles approaching 0, cause the largest eigenvalue of the stiffness matrix to explode numerically, which in turn undermines the convergence of the FE linear system.

In addition, explicit solvers such as that used in the TLED discussed above, require for the sake of stable computation a *critical time step* that is proportional to the length of the smallest edge of the whole mesh. As a result, one of the objectives in mesh generation is

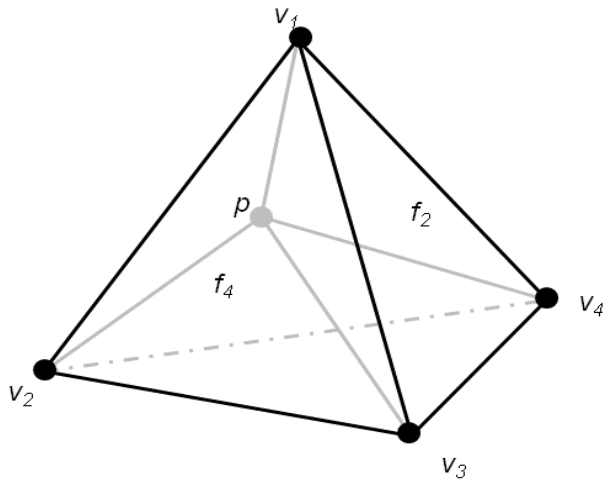


Figure 10. Illustration of tetrahedral geometry, adapted from (27), which illustrates a point whose barycentric coordinates are being computed, vertices v_k, and opposing outer

faces of two of these vertices.

control of the size of elements, in relation to enabling interactive medical simulation and limiting the complexity of off-line predictive simulation. In other words, a mesh whose smallest edge is 1 mm long will yield a critical time step that is ten times smaller than a mesh whose smallest edge is 10 mm long. Moreover, given that the volume of an equilateral tetrahedron of edge length l is given by

$$V = 212\ l3 \qquad (10)$$

and that the number of elements is therefore inversely proportional to edge length, the complexity (or size of the FE system) also decreases as a cubic function of the increase in edge length. For example, the average brain volume of a human male is 1260 cm^3; representing it with regular tetrahedra of 1 mm edges equates with 1260 $cm^3/1.18 \times 10^{-4}$ $cm^3 \approx 1.07 \times 10^7$ tetrahedra, while enforcing an edge length of 10 mm can produce about 10,700 elements, a reduction of 3 orders of magnitude of the FE system. Clearly, control over the resolution of the mesh is highly material to the achievement of an interactive surgery simulator, especially if i) the biomechanics incorporate a strongly predictive aspect, and ii) inherently volumetric processes, such as tumor resection, need to be modeled explicitly.

For some anatomical structures, ***the ideal modeling may not be tetrahedral but a collection of triangular shell elements, or even a piecewise-linear collection of beam elements.*** For example, consider large blood vessels, iatrogenic trauma to which should be penalized by the simulation, and which may require microsurgical suturing to correct. This type of simulation typically requires modeling that faithfully reproduces thin vessel walls. In this particular case, the thickness of the wall is less than 1 mm, while an

interactive simulation may require shell elements whose edges are significantly longer than 1 mm. In other words, it does not make sense to model thin-walled manifold structures with tetrahedra, but rather with shell elements, in which case the condition number is not sensitive to the small thickness of the vessel wall. In this context, Shewchuk's analysis of triangular elements, which mirrors the analysis of tetrahedral quality described above, also suggests that triangular shell elements should have no highly acute angles. The stiffness of the triangular element is given by the gradient of the barycentric coordinates in the triangle:

$$K_i = A \begin{matrix} \omega_1 \cdot \\ \omega_1 \\ \omega_1 \cdot \end{matrix} \begin{matrix} \omega_2 \\ \omega_1 \cdot \\ \omega_3 \\ \omega_2 \cdot \\ \omega_1 \\ \omega_3 \cdot \\ \omega_1 \\ \omega_2 \cdot \\ \omega_2 \\ \omega_3 \cdot \end{matrix} \begin{matrix} \omega_2 \\ \omega_2 \cdot \\ \omega_3 \\ \omega_3 \cdot \end{matrix}$$

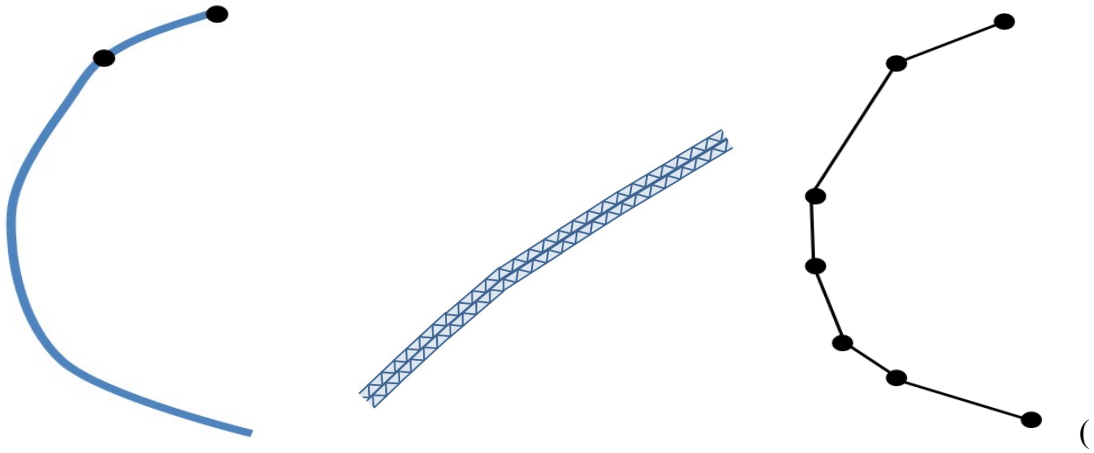
$$\omega_3 , \quad (11)$$

which in turn leads to an expression in terms of the angles of the triangular element:

$$K_i = 12 \cot\theta_2 + \cot\theta_3 - \cot\theta_3 - \cot\theta_2 - \cot\theta_3 \cot\theta_1 + \cot\theta_3 - \cot\theta_1 - \cot\theta_2 - \cot\theta_1 \cot\theta_1 + \cot\theta_2 \quad (12)$$

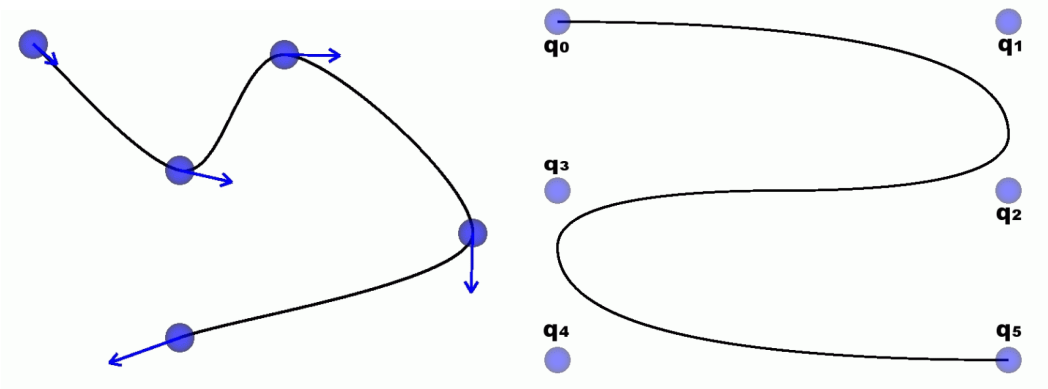
A similar development as above relates the convergence properties of triangular shell-based finite elements with small eigenvalues in expression (12), which are the result of a large angle θ_i.

Finally, anatomical structures that benefit from beam elements are essentially curvilinear, in comparison with the scale of the simulation overall: e.g. thin cranial nerves modeled in conjunction with a brain model. Consider the synthetic example of figure 11, which depicts a curvilinear structure of comparable dimensions to a cranial nerve or narrow blood vessel: let's suppose that we have a curved structure of a radius of 0.5 mm that is 20 mm of arclength. With a fully three-dimensional approach, the avoidance of acute dihedral angles in expression 9 would entail a large number of tetrahedra: the volume of this structure is 20 mm x Π x $(0.5 \text{ mm})^2 = 15.7 \text{ mm}^3$, while each tetrahedron which would have edges of roughly 0.5 mm, and a volume of 0.0147 mm^3, using expression (10). The tetrahedral approach would then produce roughly 1065 elements, and the time step would

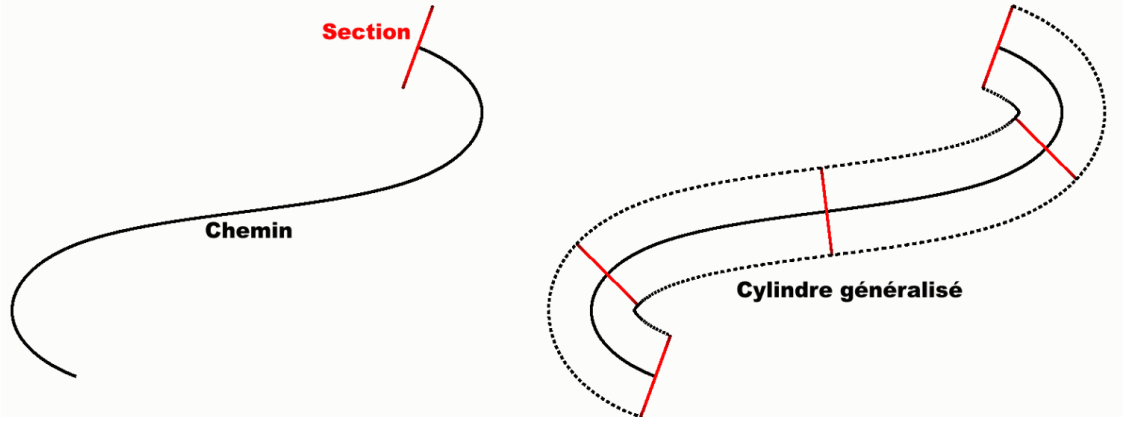


(a) (b) c) (

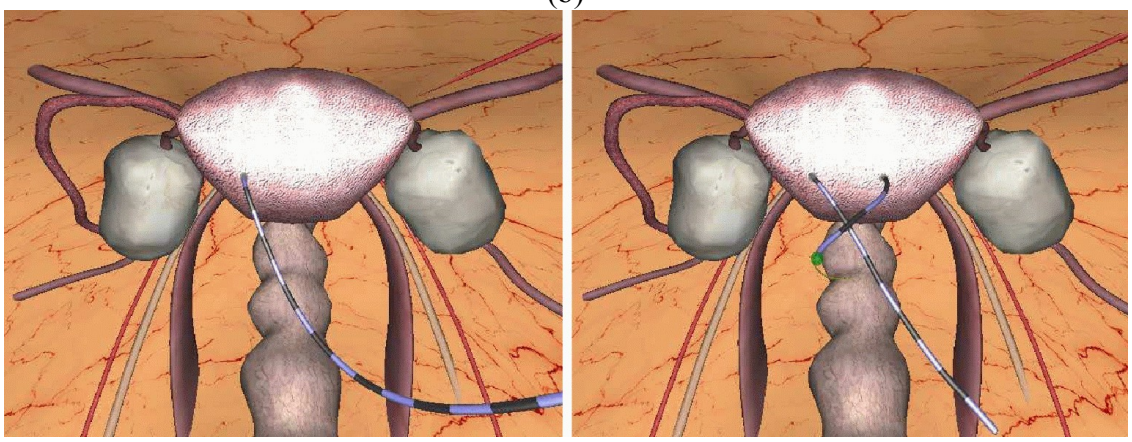
Figure 11. Application of 1D elements to curvilinear features: (a) synthetic nerve of 20mm arclength; (b) for the segment between two points in (a), the tetrahedralization is idealized in - assuming tets of 0.5mm edge length, (b) amounts to more than 1000 tetrahedra, while in (c) a piecewise-linear beam element representation uses six elements.



(a)



(b)



(c)

Figure 12. Application of 1D representation to medical simulation: (a) Catmull-Rom spline, left, and uniform B-spline, right; (b) curvilinear path, left, and generalized cylinder defined on this path, right; (c) application of generalized cylinder elastic model to suturing simulation, courtesy of J. Lenoir (SimQuest).

need to account for edges of 0.5 mm. By contrast, the 1D element approach, exemplified by the piecewise-linear beam elements of this example, would typically lead to a much more compact representation, as shown in figure 11, and each time step would reflect edges far longer than 0.5mm.

Not surprisingly, 1D elements are used to model thread in interactive suturing simulation (28), as depicted in figure 12, as well as to modeling catheters in angioplasty simulation (29). In contrast with the simple C^0 piecewise-linear example suggested in figure 11(c), Lenoir exploits a formalism based on splines, including either *Catmull-Rom*, or uniform as well as non-uniform *B-Splines*. The path traced by these splines achieves a material or tissue representation of high levels of continuity (C^1 and C^2). This path is imbedded in a general cylinder, which endows the curve with a width in 2D or a cross-sectional radius

in 3D, which serves as basis for an elastic model parameterized according to the arclength of the path.

Modeling Paradigms – Methods for Mesh Generation

Generally speaking, there are two types of mesh generation schemes: structured and unstructured. Methods for producing structured grids are based on rules for geometrical grid-subdivision and mapping techniques (30). Structured grids have a regular topology where the neighborhood relation between all points is captured with a two or three-dimensional array. If the nodes can be ordered into a regular array, where nodes *(i,j,k)* and *(i,j,k+1)* are considered neighbors, then the grid is described as structured. Otherwise, the grid is categorized as unstructured.

Structured Tetrahedral Mesh Generation

Generally, structured meshing describes quadrilateral and hexahedral meshing in 2D and 3D respectively, characterized by interior nodes having a constant number of incident elements. Typically, the elaboration of these methods involves manual interaction on the part of the user to produce a template (30), as well as a procedure for warping this template to patient data. IA-FEMesh is an open-source interactive software program that enables the development of structured hexahedral meshes (31). Figure 13 illustrates the application of this program to the development of practical anatomical meshes.

Unstructured Tetrahedral Mesh Generation

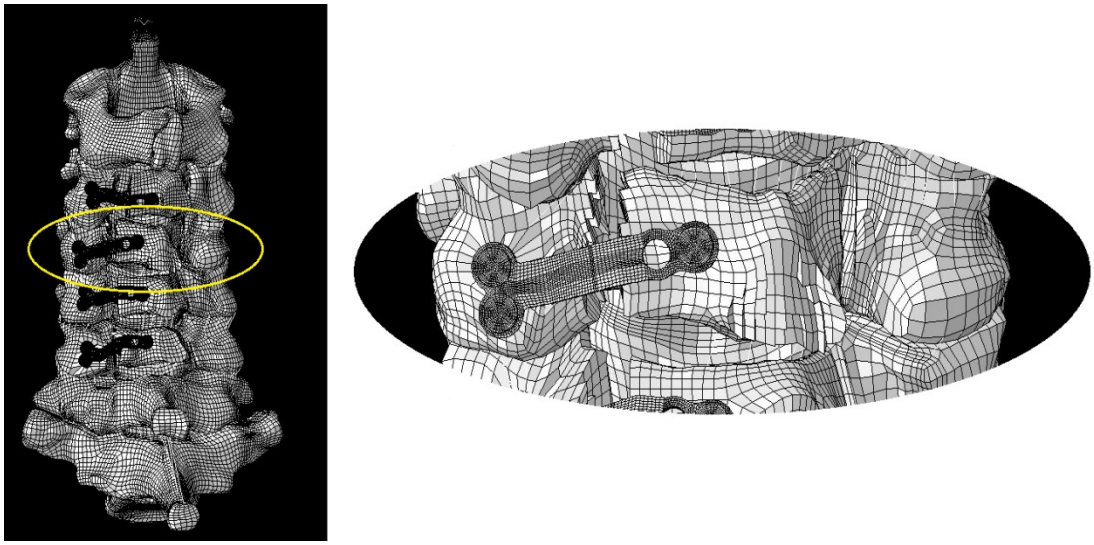
In contrast, unstructured mesh generation relaxes the node valence requirement, so that any number of elements can meet at a given node. Unstructured tetrahedral mesh generation falls under four categories (32):

- **Octree-based**

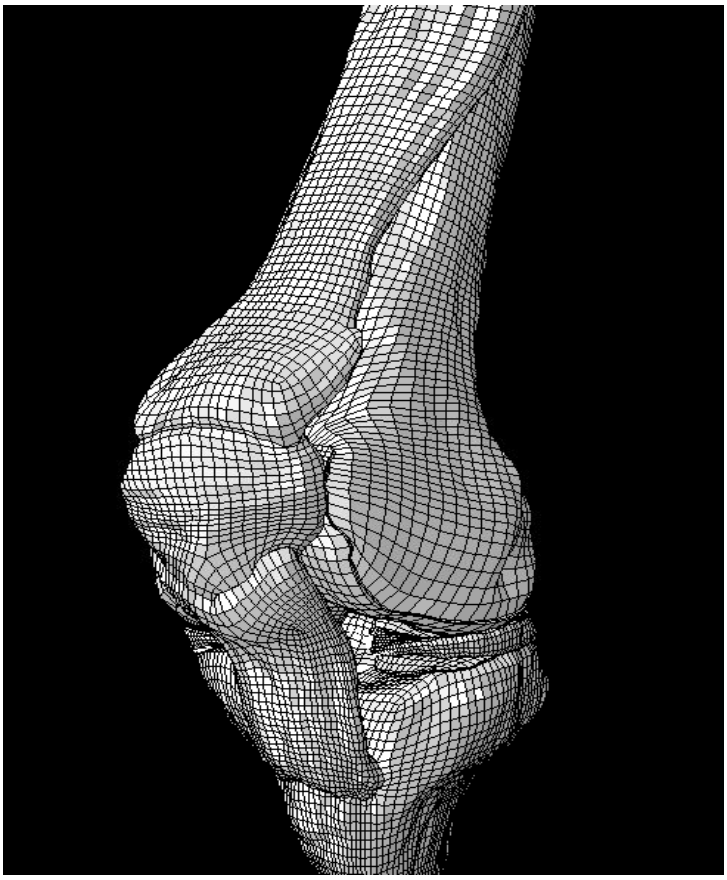
- **Delaunay**
- **Advancing Front**
- **Optimization-based**

Octree-based Unstructured Tetrahedral Mesh Generation

The octree method (33) (34) entails a recursive subdivision of the tissue volume into contiguous cubes that are fully or partially overlapped by this tissue. Irregular cells are created where the cubes intersect the boundary of the volume, which typically requires a large number of surface intersection computations. Subsequently, both irregular cells on

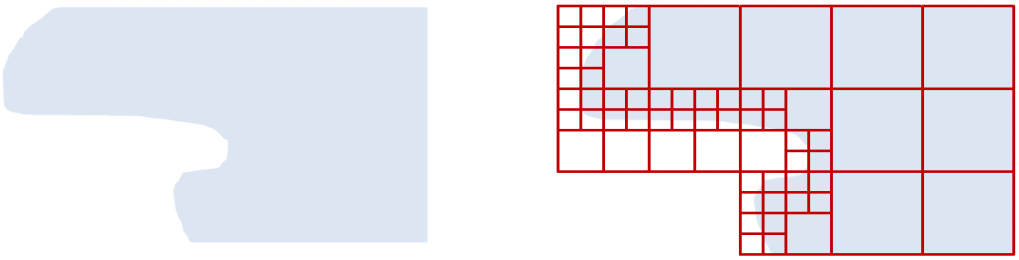


(a)



(b)

Figure 13. Orthopedic applications of structured meshing, courtesy of N. Grosland (Univ. Iowa): (a) cervical laminoplasty; (b) knee model.
generally means triangulated and tetrahedral mesh generation in 2D and 3D respectively.



(a) (b)

Figure 14. 2D analogy of octree-based spatial occupancy: quadtree; (a) section of a 2D tissue; (b) quadtree-based decomposition of three levels.

the boundary and internal regular cells give rise to tetrahedral. In general, this method does not use as input a triangulated surface boundary, but derives the tetrahedra explicitly from cubic cell occupancy. This representation then leads to a tetrahedralization inserting vertices at each inner octree node, as well as at intersections between octree edges and the tissue boundary. A simple 2D analogy, featuring the quadtree to implement spatial decomposition, appears in figure 14. To ensure that element size does not change too dramatically, a rule enforces a maximum difference of one level between adjacent cells of the octree subdivision. Post-processing can be used to improve element quality.

After an octree is constructed and filled with well-shaped tetrahedra, there are various approaches to proceed with further mesh improvement. Let us consider the first case, when the resolution of the octree is significantly more coarse then the resolution of the image. In this case, the geometric fidelity, measured as Hausdorff distance between the boundary of the mesh and the boundaries of the tissues in the image, is often not satisfactory. A common approach to improve boundary fidelity is to move the nodes of the mesh to some positions corresponding to the tissue boundaries to improve the fidelity (35) (36) (37). In the second case, the resolution of the octree matches the resolution of the structures in the image. In this scenario, the fidelity can be acceptable, however the

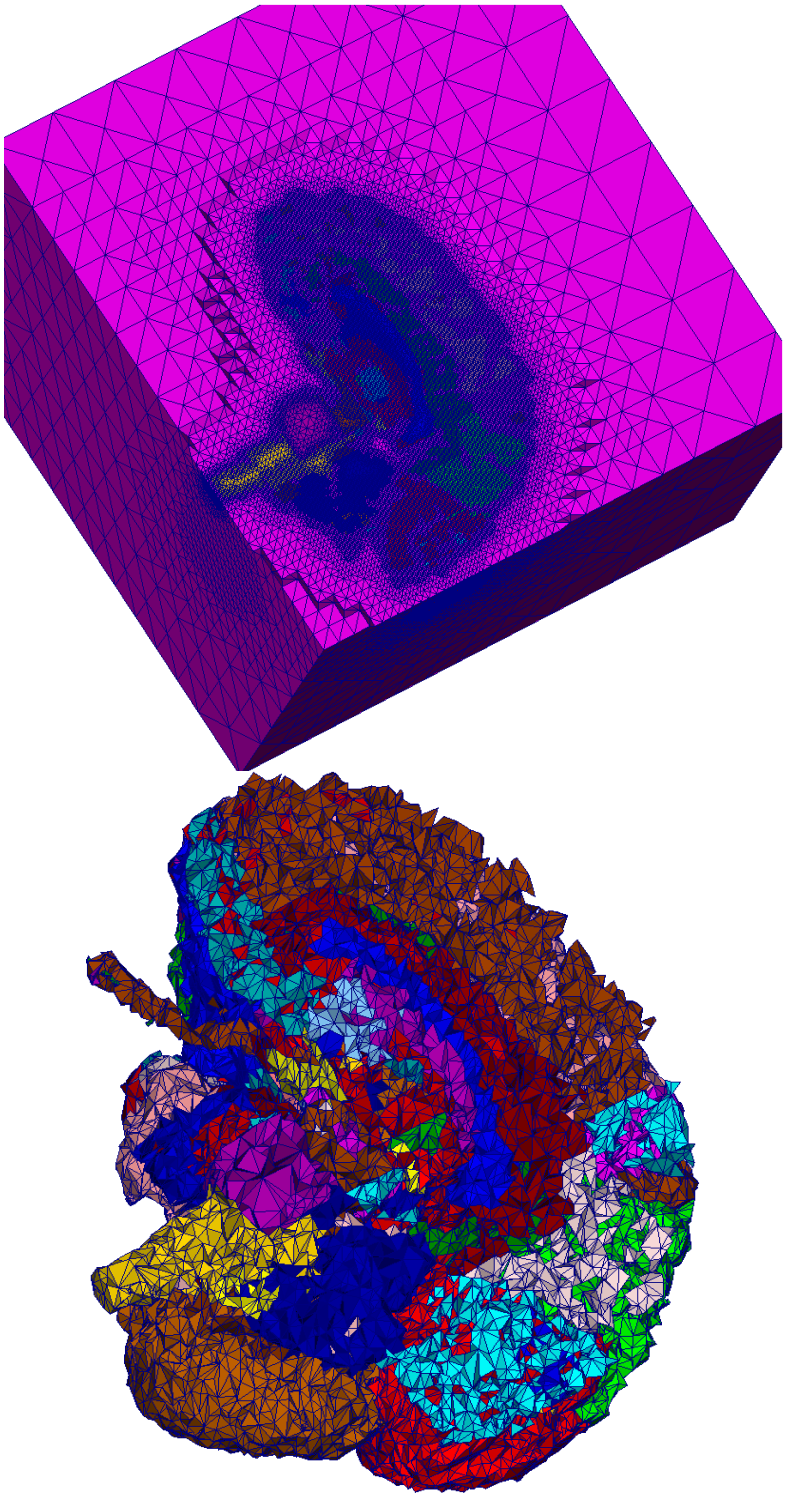
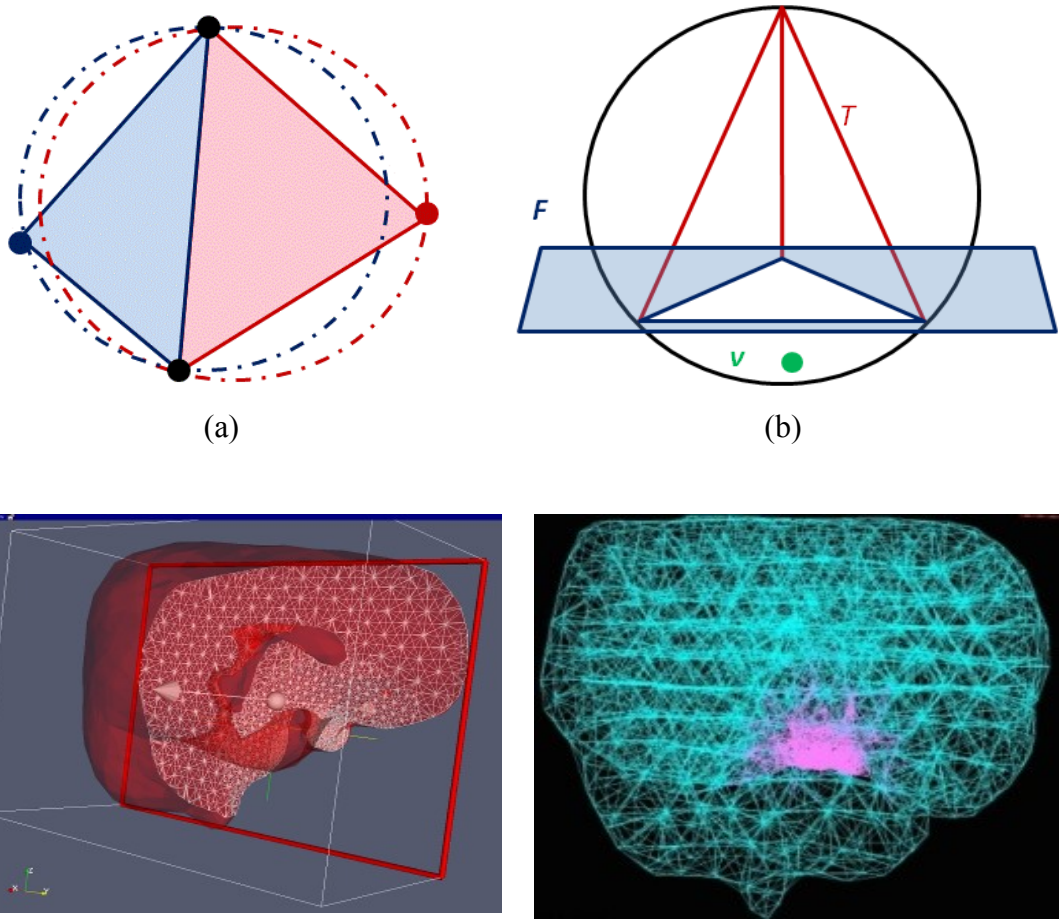


Figure 15 Octree-based tetrahedral meshing of the brain. (a) An octree-based multi-tissue tetrahedral mesh of a brain atlas (38) and the surrounding space. Right: the final mesh obtained with the Lattice Decimation method which conforms to the bounds of 2 voxels on the fidelity to the tissue boundaries and 15 degrees minimum dihedral angle (39).

number of tetrahedra in the mesh is generally very high. To reduce the number of tetrahedra in the mesh, a decimation approach was developed which, while preserving fidelity and quality bounds, makes the mesh much more sparse through the use of edge contraction operations (39). See Figure 15 for an illustration.

Delaunay Unstructured Tetrahedral Mesh Generation

The most widely used tetrahedralization technique is the method based on the Delaunay



(a) (b)

(c)

Figure 16 Delaunay tetrahedralization. (a) two neighboring triangular elements fulfilling the Delaunay "empty circle" criterion, adapted from (32) ; (b) *constrained Delaunay* tetrahedron *T* (see text below), adapted from (40); (c) topologically faithful Delaunay tetrahedralization of radially varying density for pituitary surgery simulation:

visualization based on planar intersection and rendering of individual edges (reproduced with permission from (41)).

partition of space, otherwise known as the “empty sphere” criterion. The basic idea, illustrated in figure 16 (a) with a 2D “empty circle” equivalent, is that no node can be contained in the circumsphere of a tetrahedron not incident to it. However, this criterion is not used as an algorithm for generating a mesh, but rather for determining which subset of a point cloud should be connected to form tetrahedra. A number of point insertion methods have been proposed for Delaunay methods. As suggested in (32), the simplest point insertion approach is to define a regular grid of points covering the domain at the desired density. This density can be a spatially varying size function defined by the user as opposed to a constant. A mesh of varying density was proposed by the first author for representing the brain for endoscopic pituitary surgery simulation, based on a radial edge size function defined from a point centered in the pituitary gland, as illustrated in figure 16 (c). A variation of this density- driven insertion approach involves the recursive insertion of points into the centroid of

each tetrahedron provided that the size objective function is not violated (42). Alternate methods for inserting internal points include the following:

- the *center of element circumcircles in 2D and circumspheres in 3D*, also known as Garanteed-Quality mesh generation (43) because of a minimum bound on element angles that can be achieved with a prescribed order of insertion, and
- point insertion based on the Voronoi-segment, which is defined as the line segment between two circumcircles/spheres of contiguous triangular/tetrahedral elements (44).

- So-called *selection balls* (or disks) within the circumscribed spheres of poorly shaped tetrahedra, or within the equatoral and diametral spheres of encroached boundary elements (45) (46) (47). Instead of the previous approaches based on one or two specific positions for the new points, this generalized approach offers whole three-, two-, and one-dimensional regions with infinite number of points. These regions allow for the development of multiple point insertion strategies, all within a single theoretical framework endowed with proofs of termination and good grading.

As pointed out in (40), Delaunay methods tend to produce convex tetrahedralizations, especially if we constrain them to use as few elements as possible for the sake of interactivity, while polygonal surface boundaries, also known as piecewise linear complexes (PLC), often feature concavities. Moreover, these input boundaries can feature arbitrarily small dihedral angles, which high-quality tetrahedral meshing tries to prevent. Consequently, in addition to the point insertion strategy, Delaunay methods also can be differentiated according to their policy in relation to preserving the input PLC in the output tetrahedralization, versus maintaining element quality, as follows.

- *Conforming Delaunay.* A tetrahedralization $\boldsymbol{T}$ is said to *conform* to a PLC $\boldsymbol{C}$ if for any face of $\boldsymbol{C}$ is a union of faces of $\boldsymbol{T}$. In this approach, vertices are inserted into the mesh while still maintaining the Delaunay property and until it conforms to the boundaries (48) (49). The precise choice of where to insert additional vertices to obtain boundary conformity is not obvious, especially in relation to guaranteeing termination, as sharply edged boundaries and corners can induce cascading additions of Steiner points. This problem is alleviated somewhat through the definition of protected areas near the PLC that restricts the

geometrical requirements for inserting Steiner points. Nonetheless, these methods tend to produce a large number of small tetrahedra, while the need for good-quality elements and the need for boundary conformity appear to be conflicting requirements (40). There are also parallel Delaunay mesh refinement methods with proven conformity of the concurrently refined subdomains (46) (50).

- *Almost Delaunay.* The second approach to reconciling a Delaunay tetrahedralization with input PLC is termed *almost Delaunay* approach (51). This method is similar to the conforming approach, except that the vertex insertion near the PLC boundary does not necessarily fulfill the Delaunay property. Once the PLC boundary is fully recovered by the outer faces of the tetrahedralization, the algorithm uses topological flips to regain the Delaunay property. They too can fall prey to either creating a large number of tetrahedra or very short edges.
- *Constrained Delaunay.* Constrained Delaunay Tetrahedralizations, or CDTs, are not fully Delaunay but retain some of the qualities of Delaunay Tetrahedra: they control interpolation error, while preserving high quality meshes under refinement. Moreover, they tend to require fewer vertex insertions and therefore afford better control over mesh resolution, without creating short edges. A CDT is a tetrahedralization where every tetrahedron T is *constrained Delaunay*, which is defined as follows:
    - T *respects* the PLC $\boldsymbol{C}$: there exists no segment of $\boldsymbol{C}$ that is cut in two by T, and T does not penetrate $\boldsymbol{C}$ from one side of a facet to the other.
    - There is a *circumsphere* S of T such that there is no vertex v of $\boldsymbol{C}$ that falls inside S and that is *visible from any point in the relative interior* of T. A

constrained Delaunay tetrahedron appears in figure 16 (b): the intersection of T with a facet F of $\boldsymbol{C}$ is a face of T, so this tetrahedron respects $\boldsymbol{C}$. The circumsphere of T encloses a vertex v, but v is not visible from any point in the interior of T, although visible from points on the boundary of T.

Finally, Delaunay-based mesh generation approaches for medical images can be classified according to how tissue boundary is processed. The first group begins by constructing a piecewise linear complex coinciding with the boundaries of the tissues, and then uses a CAD-oriented algorithm (such as the one implemented in Tetgen software (52)) to mesh the interior (53) (54). This approach, while enjoying the benefit of using existing algorithms, needs to solve a difficult problem of constructing an initial surface which will lead to a tetrahedral mesh with good fidelity, element shape, and number of elements. The other approach is based on simultaneous construction of mesh surface and interior (55).

Advancing Front Unstructured Tetrahedral Mesh Generation

Starting from the tissue boundary, new vertices are added by a local heuristic to ensure

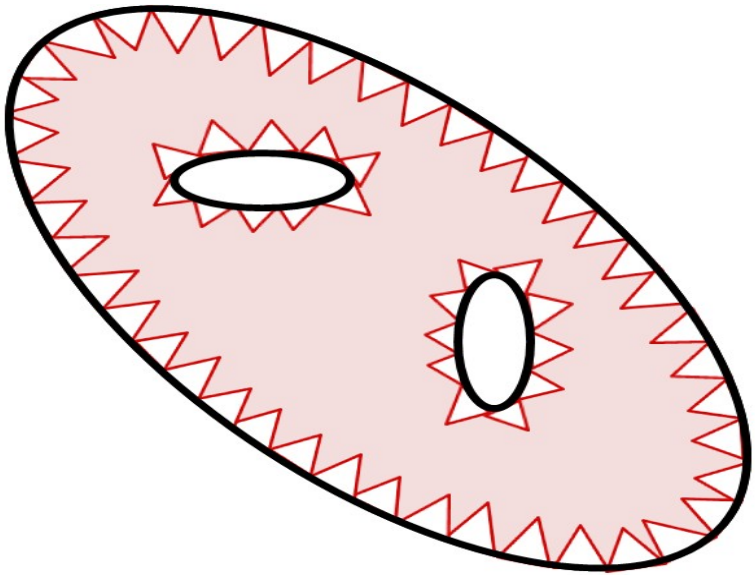


Figure 17. Two-dimensional analogy for advancing front method, after one iteration, based on simple ellipsoid with two holes; adapted from (32).

that the generated tetrahedra have acceptable shapes and sizes and conform to the desired size objective (56). An active front is maintained where new tetrahedra are formed. Figure 17 shows a simple two dimensional analogy, where triangles have been formed at the boundary. As the algorithm progresses, the front will advance to fill the remainder of the area with triangles. In three-dimensions, for each triangular facet on the front, an ideal location for a fourth node is computed. Mesh optimization steps can also be performed to further improve the quality.

Optimization-based Unstructured Tetrahedral Mesh Generation

In general contrast to the preceding approaches, variational approaches express tetrahedral meshing as a functional that is interpreted in terms of energy, based on calculus of variations, and where the optimal mesh results from an iterative minimization of this functional, through vertex displacements and connectivity changes of the mesh. The main difficulties of the early implementations of this type of method included their susceptibility to local minima, as a result of highly non-convex functionals, and as well as

to producing non-compact, sliver-like tetrahedra (57). Successful methods in this category include the *Variational Tetrahedral Meshing* proposed in (57) and *Centroidal Voronoi Diagram*-based variational method (58).

Unstructured Surface Mesh Generation

The representation of surface boundaries by unstructured meshes typically involves the identification of the polygonal intersection of a tissue volume with a regular grid, also known as *isosurface extraction*, in the manner pioneered by the *Marching Cubes* method (59), which can give rise to fifteen topologically different cases, as depicted in figure 18.

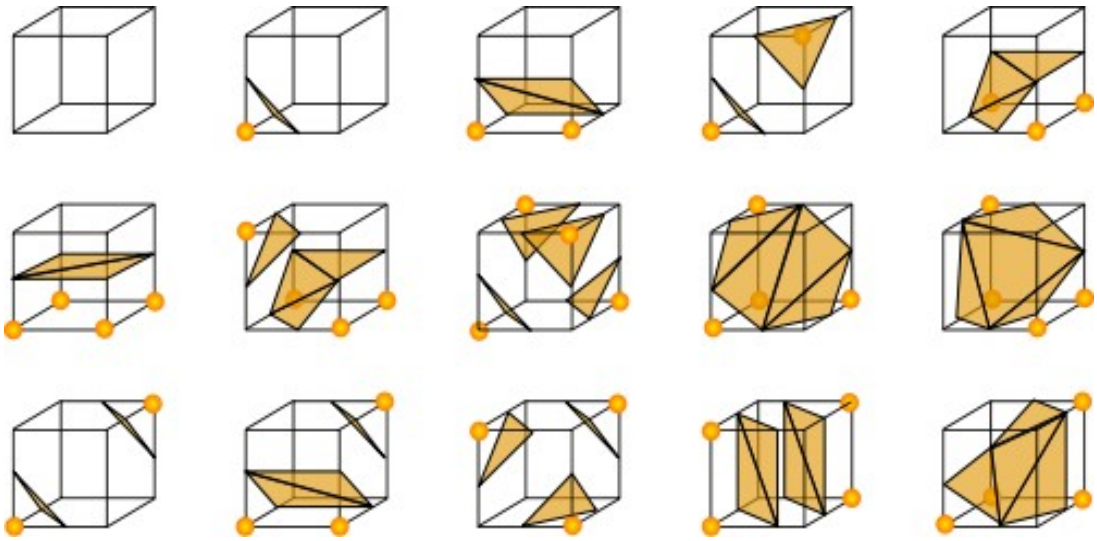


Figure 18. Illustration of isosurface intersection cases of Marching Cubes (reproduced from freely licensed media repository of (60)).

These polygons are then stitched together to produce a polygonal surface, which can be refined into purely triangulated surface. The Marching Cubes method subsequently lead to a number of refinements based on other types of grids, namely the Marching Tetrahedra (61). For a survey of isosurface extraction, the reader is referred to (62).

The results of these methods tend to be extremely dense and require decimation in order to produce a surface that is usable in simulation, especially interactive simulation.

However, most of the early decimation methods were conceived to meet the requirements of visualization of a static scene, rather than simulation based on a dynamically deformable model, so that they tend to produce very short edges at high-curvature areas.

By contrast, the discrete active surface model known as the simplex model, introduced in (23), offers a means of precisely controlling the density of the final mesh, as well as a number of other advantages. The simplex is a physically based discrete surface model featuring a set of vertices, with each vertex having a mass as well as three neighbors, as

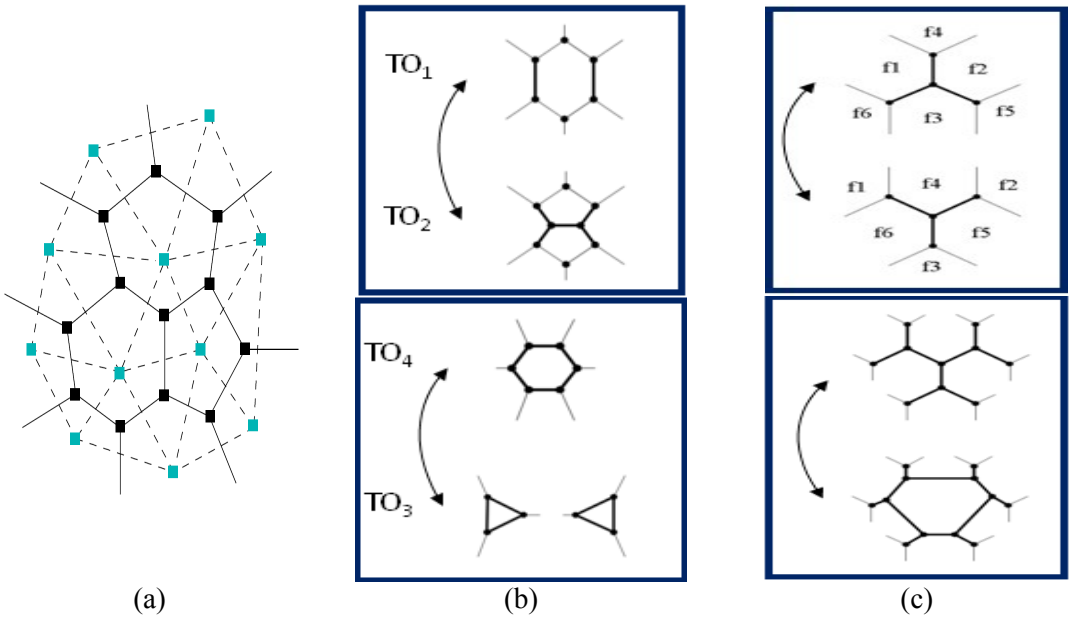


Figure 19. Topology of simplex model. (a) 3-neighbour vertex topology, and duality with triangulated mesh; (b) two topological operators of the Delingette simplex model; (c) two of several topological macro-operators introduced in the Gilles simplex model, courtesy of B. Gilles (INRIA, Grenoble, France).

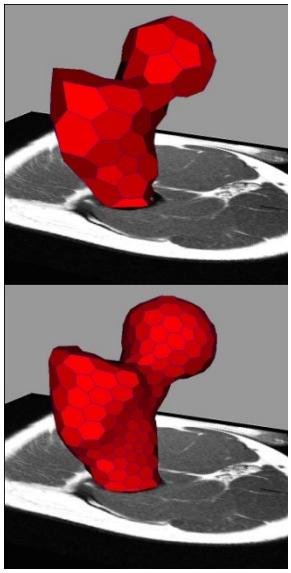
shown in figure 19 (a), and where the motion of each vertex is governed by a Newtonian law of motion:

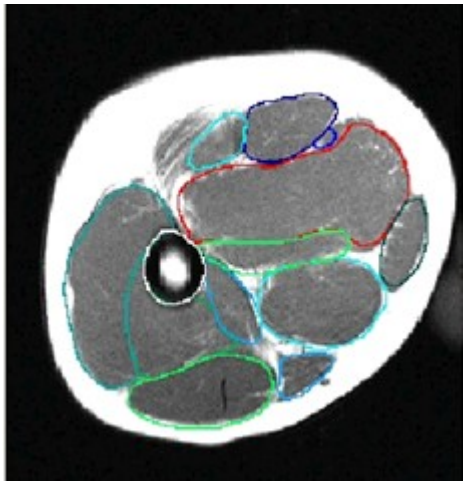
$$m\,\partial^2 pi/\partial t^2 + \gamma\,\partial pi/\partial t - \alpha\,fintpi = \beta\,fext(pi) \quad (13)$$

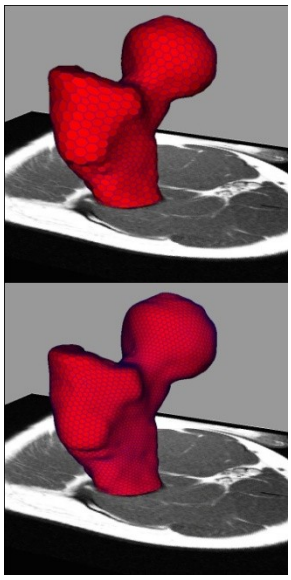
$\mathbf{F}_{int}$ is the sum of internal forces, which enforces geometric continuity, while $\mathbf{F}_{ext}$ represents summed external forces. The variable *m* denotes the vertex mass, γ the damping factor, and α and β are weight factors balancing internal and external forces. It features forces that assure i) *internal stabilization* in $\mathbf{F}_{int}$, ii) *balloon inflation and image attraction forces*, and iii) *topological operators* (see figure 19 (b)) that enable the insertion and deletion of individual edges as well as provide resolution control. A Simplex-based solution was proposed by the first author to achieve controlled-resolution decimation of brain meshes for (41).

In addition, a number of highly descriptive features were proposed in (63) for musculoskeletal applications, emphasizing multi-surface modeling (figure 22) and also included the following:

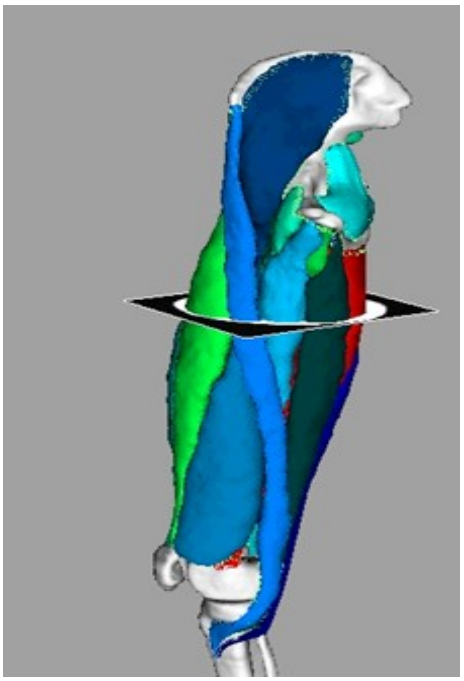
Topological operators that produce hexagonal faces (figure 19 (c)), enabling







(a)



(b)

Figure 20. Recent innovations of Simplex model: (a) multi-resolution meshing; (b) multi-surface coupled model, courtesy of B. Gilles (INRIA, Grenoble, France).

- resolution control and *multi-resolution surface meshes*;
- *spline-based contours for inter-surface attachments*.
- *Medial axis representation of the surface interior*, coupling forces between medial axis and surface model, and medial-axis based inter-surface collision handling.

Beyond the above refinements for multi-resolution and multi-surface modeling, recent work (24) also endowed the Simplex with a statistical-shape energy functional, as depicted in figure 21:

$$E_{shape} = (\vec{s} - \vec{s}_t)^T \Sigma_{\perp}^{-1} (\vec{s} - \vec{s}_t), \quad (14)$$

the regularized covariance matrix. This term models the universe of possible object

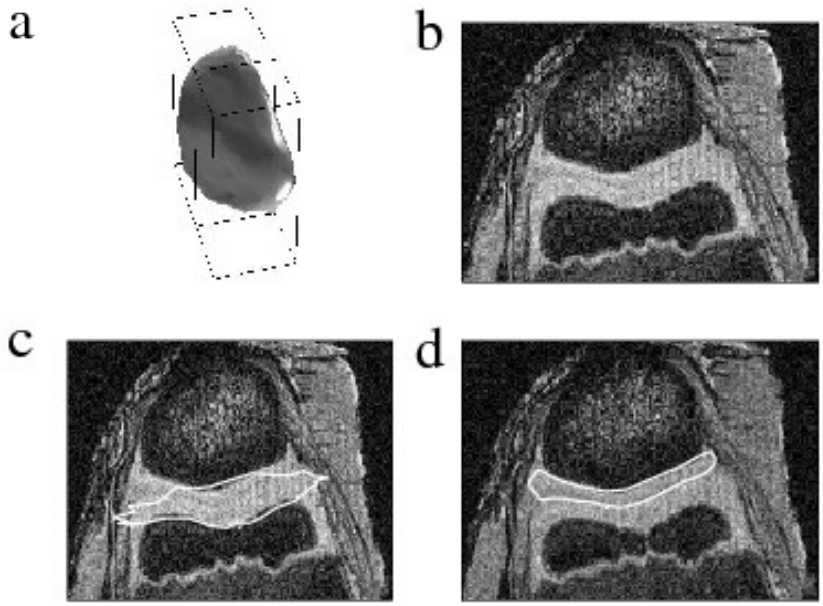


Figure 21. Shape statistics-aware simplex mesh: 3D patellar cartilage model: (a) 3D rendering; (b) MR axial slice (note: low contrast); axial slice of Simplex model (c) without and (d) with shape statistics, reproduced with permission from (24).

where S is the Simplex model, $\mathbb{S}$ is the average Simplex shape, and $\Sigma_{\perp}^{-1}$ is the inverse of shapes as a Gaussian distribution. A minimization of the expression in (14) is used to modify the evolution equation in (13) to nudge the surface model towards the average shape, in a manner inversely weighted by second-order statistics.

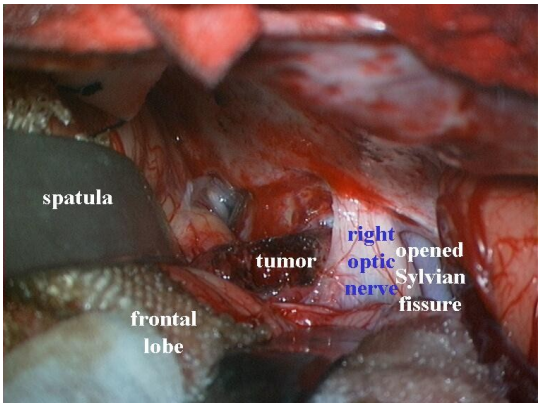
In addition to the Simplex-based approach, a variational formulation to surface decimation was proposed by Valette, based on the notion of Centroidal Voronoi Diagram (CVD) (64). This formulation is characterized by a partitioning, or *clustering*, of the input mesh, with individual clusters positioned according to a user-provided choice of mesh density, which in turn leads to an energy functional for spatially distributing clusters over the original surface. This technique is revisited in the Case Studies section. Recently, this

method was also adapted to deal with implicit surfaces, which typically arise in the application of meshless methods (65).

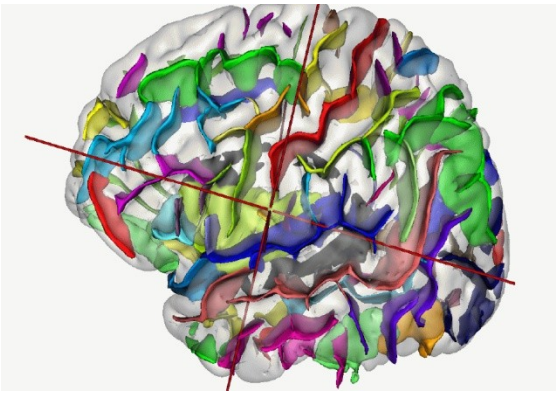
The implications of these surface meshing refinements to tetrahedral meshing are profound, especially those methods that preserve the integrity of the original triangulated tissue boundary. In conjunction with surface meshing techniques based on the simplex model, ***tetrahedral meshing can now exploit boundary meshing methods that enforce precise resolution control, multi-tissue boundary consistency as well as conformity to population shape statistics.*** This development, in our opinion, portends a revolution in anatomical modeling, with equally important implications for medical simulation, especially in conjunction with the application of multi-grid finite elements.

Case Studies

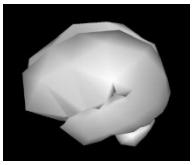
The first author has collaboratively applied the CVD surface meshing method (66), described above, to controlled resolution surface meshing of the brain with strict requirements for fidelity to the input surface. In particular, this surface emphasizes separability of the mesh about one Sylvian fissure in conjunction with the on-going

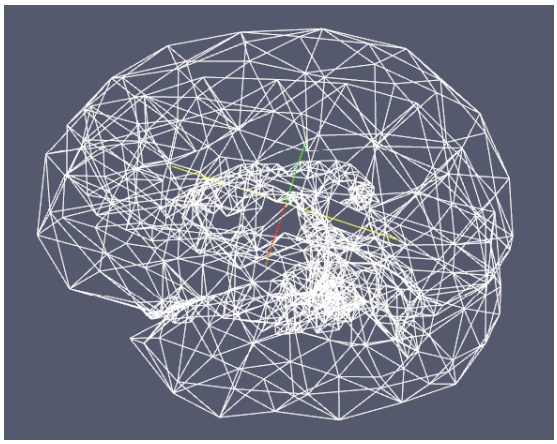


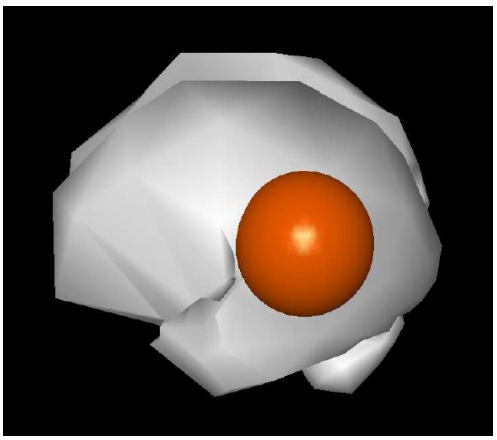
(a)



(b)







(c) (d)

Figure 22. Illustration of approach specific, sulcal-separable brain surface mesh decimation by CDV, based on on-going work of first author: (a) illustration of access to deep-seated tumor by pterional approach, featuring tissue being parted about the Sylvian fissure and critical tissues, courtesy of C. Trantakis (Leipzig); (b) illustration of BrainVisa sulcal labeling output; (c) wireframe rendering of ACV surface mesh; (d) simulated deformation of tetrahedral model through haptic interaction with TLED biomechanics engine (inset: undeformed tetrahedral model).

development of a neurosurgery simulator based on the pterional approach (66), as depicted in figure 22. The identification of the Sylvian fissure is achieved with a publicly available sulcal labeling software (67) (68). This fissure was found to be easily be obliterated by decimation methods whose final result is not close to the input surface mesh (69), such as the decimation available through the Computational Geometry Algorithm Library (70), even if the resulting mesh was otherwise high-quality. Figure 22 (d) illustrates preliminary results in TLED-based haptic interaction with tetrahedral model, which was produced by inputting the CVD surface to Tetgen. The development

of this model is produced in conjunction with clinical specifications through the development of approach- and pathology-specific neurosurgical ontologies (71).

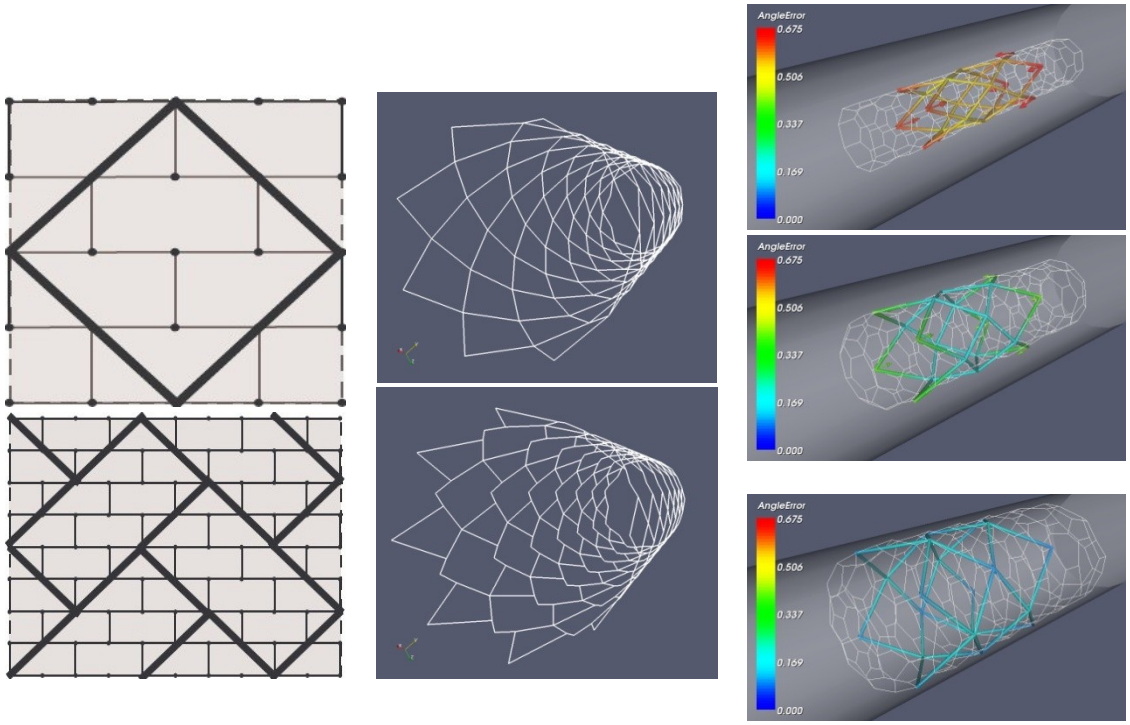


Figure 23. Application of simplex mesh to stent design. (a) Basic topology - top: regular simplex mesh (thin lines with vertices) imbedded with strut mesh (thick lines); bottom: typical strut mesh pattern; (b) two stent designs; (c) screen shots of deployment in vessel. Images courtesy of I. Larrabide and A. Frangi (Univ. Pompeu Fabra, Barcelona, Spain.)

Beyond the modeling of anatomical surfaces, the simplex mesh has also been applied to the design of stents and the simulation of their deployment, as depicted in figure 23 (72). This stent simulation was achieved by first modeling the simplex with rectangular faces, while still preserving the three-neighbor connectivity, and then imbedding a coarser strut mesh within the simplex mesh. In addition, as shown in figure 23 (b), this imbedded simplex-strut mesh pattern was given a cylindrical topology. The cylindrical model could then be represented with an equilibrium equation that accounted for both simplex and strut meshes. Expression (13) was restated as

$$pit+1 = pit+1 - \gamma\,(pit-pit-1) + \alpha\, fint(pit) + \beta\, fext(pit)\,, \qquad (15)$$

which was expanded to account for shape-constraining forces needed to model the stent:

$$pit+1 = pit+1 - \gamma\, pit-pit-1 + \chi\, fl(pbt) + \Phi\, fapbt + fspit + fexp(pit) + fext(pit)\,. \qquad (16)$$

In this model, parameters α and β are considered equal to 1, and the equilibrium expression applies to both the simplex vertices pit and the stent points pbt. In addition, χ and Φ are factors that balance the forces that determine the strut mesh length and angle forces fl and fa respectively, while fs and fexp are simplex mesh smoothing and stent expanding forces. The result of the stent-based vessel opening was input to a computational fluid dynamics simulation of the blood flow, which in turn is validated against phantoms.

Summary

This chapter surveyed tetrahedral and surface meshing techniques, emphasing those most relevant to medical simulation. It is clear that medical simulation is still in its infancy, and that improvements to methods for producing anatomical meshes will be a key issue in making simulators more clinically relevant, including enabling the computation of a patient-specific suite of models for clinical applications of interest. The introduction provided the reader with an understanding of leading methods of biomechanical engines used in the simulation of manipulation and cutting of tissues. A theoretical justification of the relevance of high-quality elements was described. Finally, the leading four categories of tetrahedral meshing were surveyed. Leading techniques for producing controlled resolution surface meshes, based on the simplex model and clustering, were also presented, with the former also offering methods for dealing with multi-tissue boundaries and with population shape statistics.

Key Terms

Medical simulation:

predictive simulation: medical simulation whose emphasis is high-quality computations, typically used by expert surgeons to optimize their procedure.

interactive simulation: medical simulation whose emphasis is real-time response, usually to a surgical gesture input through a haptic device by a young surgeon in the course of training.

kinematics loop: the loop in haptic processing that converts input voltages from the device into input joint angles and input tool position.

haptics loop: the loop in haptic processing that converts output force computed by the biomechanics engine into output joint torques and output voltages to the motors of the device.

Meshing:

mesh generation: the decomposition of a tissue volume into simple shapes, suitable for finite elements, such as hexahedra or tetrahedra.

unstructured mesh generation refers to this decomposition, whose vertices are inserted in an irregular pattern rather than a grid, which typically results in tetrahedra.

dihedral angle: the angle between two planes subsuming two triangular faces meeting at a vertex; there are six dihedral angles in a tetrahedron.

Mechanics:

finite elements method (FEM): a numerical simulation based on decomposition of non-overlapping elements, where a weak form of an equilibrium expression is discretized through shape functions defined on tetrahedral or triangular elements.

extreme finite elements method (XFEM) is a variant of FEM that features shape functions that admit a discontinuity, in contrast to the preceding that is based on continuous interpolants.

Principle of Virtual Work: principle describing an equilibrium where applied forces and constraint forces on a mechanical system balance such that the system remains static, and according to which virtual work of these forces also balance out; the weak form of this principle is the foundation of FEM.

shape function: interpolating function of an element that relates nodal quantities such as displacements to quantities evaluated at any position within the element, based on internal coordinates defined with respect to the nodes of the element; these functions are assembled into a matrix for the whole volume, and the stiffness matrix is obtained from this shape matrix.

condition number: for a nonsingular linear system, e.g. $\mathbf{Ax} = \mathbf{b}$, a number that indicates the sensitivity of the solution $\mathbf{x} = \mathbf{A}^{-1}\,\mathbf{b}$, based on the factor of the largest to the smallest eigenvalues of **A**, to small changes in **b**, which in turn is predictive of the system's numerical convergence to a solution.

meshless methods: continuum mechanics simulation that is based on a volumetric decomposition into overlapping spheres of interest defined over a cloud of points, where a weak form of a equilibrium expression is discretized through shape functions defined on these overlapping spheres.

mass-spring systems are an elastic sparse representation of a volume, in contrast to the densely interpolative schemes described above: this representation consists of point-masses linked by linear springs.

Unstructured Tetrahedral meshing

Octree-based meshing: Tetrahedralization based on a hierarchical decomposition of space into octrees.

Delaunay meshing: Tetrahedralization based on the insertion of points is carried out in way which is optimal in terms of a containment sphere.

Advancing-front meshing: Tetrahedralization based on the iterative addition of tetrahedra, starting from tissue boundaries.

Optimization-based meshing: Tetrahedralization achieved through the expression of the meshing problem as a functional based on calculus of variations, suitable for optimization.

Unstructured Surface Meshing

Simplex mesh: Active discrete surface model characterized by each vertex having three neighbours, and by an equilibrium relation between pseudo-forces that govern the behavior of the model.

Exercises

1. Is it feasible to input a segmented volume into a numerical simulation? If so, why, and if not, why not?
2. Is it feasible to input a raw image volume into a tetrahedralization method? If so why, and if not, why not?
3. If we are representing the anatomy of interest by tetrahedral meshes of several levels of resolution, should we model the whole volume coinciding with the organ at each level, as was illustrated in figure 4?
4. If we must represent clinically relevant blood vessels by elements, what are the relative merits of using tetrahedra versus shell elements?
5. Contrast the main methods for decomposing a volume into tetrahedral elements.
6. Describe the simplex surface model. The surface model is formally known as a 2-simplex, where an n-simplex is such that (n+1) is the number of vertex neighbors. If this 2-simplex is known as geometrically dual to a triangulation, what is the value of *n* for a *n*-simplex that is dual to a tetrahedron?

References

1. Sensable. [Online] http://www.sensable.com/.
2. **MPB Technologies, Inc.** F7S Force Feedback Hand Controller. [Online] http://www.mpb-technologies.ca/mpbt/haptics/hand_controllers/scissors/description.html.

3. **Burdea G, Coiffet P.** *Force and Touch Feedback for Virtual Reality, 2nd ed.* s.l. : Wiley, 2003.
4. **Craig JJ.** *Introduction to Robotics: Mechanics and Control, 3rd ed.* s.l. : Prentice-Hall, 2004.
5. *A Physical Model of Facial Tissue and Muscle Articulation Derived from.* **Waters K.** 1992. Proc. Visual. in Biomed. Comput.- SPIE Vol. 1808. pp. 574-583. Proc. Visual. in Biomed. Comput.- SPIE Vol. 1808.
6. **Zienkiewicz O.C., Taylor R.L.** *The Finite Element Method, 4th ed., vol. 1 & 2,.* s.l. : McGraw-Hill, 1991.
7. *Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation.* **Miller K 06, Joldes G, Lance D, Wittek A.** 2006 йил, Communications in Numerical Methods in Engineering, pp. 23:2, pp 121 - 134.
8. *Real-Time Surgical Simulation Using Reduced Order Finite Element Analysis.* **Taylor Z.A.10, Crozier S. & Ourselin S.** s.l. : Springer, 2010. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010 Lecture Notes in Computer Science. pp. 388-395.
9. *Fast porous visco-hyperelastic soft tissue model for surgery simulation: application to liver surgery.* **Marchesseau S, Heimann T, Chatelin S, Willinger R, Delingette H.** 2010, Progress in Biophysics and Molecular Biology,, pp. 103(2-3):185-196.
10. **Mosegaard TS, Sorensen J,.** An Introduction to GPU Accelerated Surgical Simulation. *Biomedical Simulation - Lecture Notes in Computer Science.* Berlin : Springer, 2006, pp. 93-104.
11. **RenderStream.** RenderStream High Performance Computing. [Online] http://www.renderstream.com/HPC.html.
12. *Real-Time Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units.* **Taylor Z.A.07, Cheng M., Ourselin S.** Brisbane : Lecture Notes in Computer Science, Springer, 2007. Medical Image Computing and Computer-Assisted Intervention – MICCAI. pp. 701-708.
13. *Multigrid Integration for Interactive Deformable Body Simulation.* **Wu X., Tendick F.** Berlin : Lecture Notes in Computer Science, Springer, 2004. Medical Simulation. pp. 92-104.
14. **Jerabkova L.** *Interactive Cutting of Finite Elements based Deformable Objects in Virtual Environments.* Aachen, Germany : Ph.D. Thesis, RWTH Aachen University, 2007.
15. **SOFA.** SOFA, Simulation Open Framework Architecture. [Online] http://www.sofa-framework.org/home.
16. **CHAI 3D.** CHAI 3D. [Online] http://chai3d.org/.
17. **H3D.** Open Source Haptics. [Online] http://www.h3dapi.org/.
18. **Hu YC, Grossberg MC, Mageras G.** Survey of Recent Volumetric Medical Image Segmentation Techniques. [book auth.] Barros de Mello CA. *Biomedical Engineering.* s.l. : InTech, 2009, pp. http://www.intechopen.com/source/pdfs/8807/InTech-Survey_of_recent_volumetric_medical_image_segmentation_techniques.pdf.
19. *Current Methods in Medical Image Segmentation.* **Pham DL, Xu C, Prince JL,.** 2000, Annual Review of Biomedical Engineering, , pp. vol. 2, pp. 315-338.
20. **Slicer 3D.** 3D Slicer v3.6. [Online] http://www.slicer.org/.

21. *Initialization, Noise, Singularities, and Scale in Height Ridge Traversal for Tubular Object Centerline Extraction.* **Aylward SR, Bullitt E.** Feb. 2002, IEEE Trans. Med. Imag., pp. 21(2): 61-75.
22. *Shape modeling with front propagation: a level set approach .* **Malladi R, Sethian JA, Vemuri BC,.** Feb 1995 , Pattern Analysis and Machine Intelligence, IEEE Transactions on , pp. 17 (2), 158 - 175 .
23. *General Object Reconstruction Based on Simplex Meshes.* **Delingette H.** 1999, Int. J. Comput. Vis., pp. 32(2):pp 111-146.
24. *Simplex Mesh Diffusion Snakes: Integrating 2D and 3D Deformable Models and Statistical Shape Knowledge in a Variational Framework.* **Tejos C, Irarrazaval P, Carrdenas-Blanco A.** October 2009, Int. J. Comput. Vision , pp. 85(1) , 19-34. .
25. *A Brain Tumor Segmentation Framework Based on Outlier Detection.* **Prastawa M, Bulllitt E, Ho S, Gerig G.** 2004, Medical Image Analysis (MedIA)., pp. Vol 8, Issue 3, Pages 275-283.
26. *A bayesian model for joint segmentation and registration.* **Pohl K.M., Fisher J., Grimson W.E.L., Kikinis R., Wells W.M.** 2006, NeuroImage, pp. 31(1), pp. 228-239, .
27. *What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures.* **Shewchuk J.** Ithaca, New York : Sandia National Laboratories, September 2002. Eleventh International Meshing Roundtable . pp. pages 115-126.
28. *A Suture Model for Surgical Simulation.* **Lenoir J., Meseure P., Grisoni L., Chaillou C.** 2004 . International Symposium on Medical Simulation. pp. 105-113.
29. *Interactive physically-based simulation of catheter and guidewire.* **Lenoir J., Cotin S., Duriez C., Neumann P.** 2006, Computer & Graphics, pp. 416-422.
30. *IA-FEMesh: An open-source, interactive, multiblock approach to anatomic finite element model development.* **Grosland NM, Shivanna KH, Magnotta VA, Kallemeyn NA, DeVries NA, Tadepalli SC, Lisle C.** 2009, Comput Methods Programs Biomed., pp. April;94(1): 96–107.
31. **IA-FEMesh.** IA-FEMesh - Center for Computer Aided Design. [Online] http://www.ccad.uiowa.edu/mimx/IA-FEMesh/.
32. *A survey of unstructured mesh generation technology.* **Owen, S.** 1998. Proceedings of the 7th International Meshing Roundtable. pp. pp. 239-267 .
33. *Three-Dimensional Mesh Generation by Modified Octree Technique.* **Yerry MA, Shephard MS.** 1984, Int. J. Numerical Methods in Engineering, pp. vol 20, pp.1965-1990.
34. *Three-Dimensional Mesh Generation by Finite Octree Technique.* **Shephard MS, Georges MK.** 1991, Int. J. Numerical Methods in Engineering,, pp. vol 32, pp. 709-749.
35. *Isosurface Stuffing: Fast Tetrahedral Meshes with Good Dihedral Angles.* **Labelle F, Shewchuk JR.** 2007, ACM Trans. Graphics, pp. 26(3):57.1-57.10.
36. *Mesh Deformation-based Multi-tissue Mesh Generation for Brain Images.* **Liu Y, Foteinos P, Chernikov A, Chrisochoides N.** 2011, Engineering with Computers, p. in press.
37. *Tetrahedral Mesh Generation for Medical Imaging.* **Fedorov A, Chrisochoides N, Kikinis R, Warfield SK.** 2005. Med. Image Comput. & Comput. Assist. Interv. .
38. **Talos I, Jakab M, Kikinis R, Shenton M.** SPL-PNL Brain Atlas. [Online] http://www.slicer.org/publications/item/view/1265.

39. *Tetrahedral Image-to-Mesh Conversion for Biomedical Applications.* **Chernikov A., Chrisochoides N.** s.l. : ACM, 2011. ACM Conf. Bioinformatics, Computational Biology and Biomedicine. pp. 125-134.
40. *Constrained Delaunay Tetrahedralizations and Provably Good Boundary Recovery.* **Shewchuk JR 02a.** 2002. Eleventh International Meshing Roundtable. pp. 193--204.
41. *M.A. Audette, H. Delingette, A. Fuchs, O. Astley and K. Chinzei, A. "Topologically Faithful, Tissue-guided, Spatially Varying Meshing Strategy for Computing Patient-specific Head Models for Endoscopic Pituitary Surgery Simulation".* **Audette MA 07, Delingette H, Fuchs A, Astley O and Chinzei K.** Jan. 2007, Journal of Computer Aided Surgery, pp. 12(1): 43–52.
42. *Efficient Three-dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints.* **Weatherill NP, Hassan O.** 1994, Int. J. Numerical Methods in Engineering, , pp. vol 37, pp.2005-2039.
43. **Chew LP.** *Guaranteed-Quality Triangular Meshes.* s.l. : Cornell University, 1989. Department of Computer Science Tech Report 89-983.
44. *Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson.* **Rebay S.** 1993, J. Computational Physics, , pp. vol. 106, pp.125-138.
45. *Three-Dimensional Semi-Generalized Point Placement Method for {Delaunay} Mesh Refinement.* **Chernikov AN 07, Chrisochoides NP.** Seattle, WA : Springer, 2007. Proc. 16th Int. Meshing Roundtable. pp. 25-44.
46. *Parallel Guaranteed Quality {Delaunay} Uniform Mesh Refinement.* **Chernikov AN 06, Chrisochoides NP.** 2006, {SIAM} J. Scientific Computing, pp. 1907-1926.
47. *Fully Generalized Two-Dimensional Constrained {Delaunay} Mesh Refinement.* **Foteinos PA, Chernikov AN, Chrisochoides NP.** 2010, SIAM J. Scientific Computing, pp. 2659-2686.
48. *A point-placement strategy for conforming Delaunay tetrahedralization.* **M. Murphy, D. M. Mount, and C. W. Gable.** 2001, Int. J. Computational Geometry and Applications, pp. 11(6):669-682.
49. *Conforming Delaunay Triangulations in 3D.* **D. Cohen-Steiner, E.C. Verdiere and M. Yvinec,.** 2004, Computational Geometry: Theory and Applications, pp. 28 (2–3), pp. 217–233.
50. *Three-Dimensional {Delaunay} Refinement for Multi-Core Processors.* **Chernikov AN 08, Chrisochoides NP.** Island of Kos, Greece : ACM Press, 2008. Proc. 22nd Annual Int. Conf. on Supercomputing. pp. 214-224.
51. *Approximating Constrained Tetrahedrizations.* **Hazlewood C.** 1993, 1993, pp. 10:67–87.
52. **Tetgen.** TetGen - A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator. [Online] http://tetgen.berlios.de/.
53. *Tetrahedral Mesh Generation for Non-Rigid Registration of Brain MRI: Analysis of the Requirements and Evaluation of Solutionsof Solutions.* **Fedorov A 08, Chrisochoides N.** 2008. Proc. 17th International Meshing Roundtable . pp. 55-72.
54. *An Evaluation of Tetrahedral Mesh Generation for Non-Rigid Registration of Brain MRI.* **Foteinos P 10b, Liu Y, Chernikov A, Chrisochoides N.** 2010. Proceedings of the Computational Biomechanics for Medicine V, Medical Image Computing and Computer Assisted Intervention (MICCAI) Workshop,. pp. 126-137.

55. *Guaranteed Quality Tetrahedral Delaunay Meshing for Medical Images,.* **Foteinos P 10a, Chernikov A, Chrisochoides N.** 2010. Proc. 7th International Symposium on Voronoi Diagrams in Science and Engineering. pp. pages 215-223.
56. *Progress in Grid Generation via the Advancing Front Technique.* **Lohner R.** 1996, Engineering with Computers,, pp. vol 12, pp.186-210.
57. *Variational tetrahedral meshing.* **Alliez P., Cohen-Steiner D., Yvinec M., Desbrun M.** 2005, ACM Transactions on Graphics , pp. 24(3), 617–625 .
58. *Variational tetraedral mesh generation from discrete volume data,.* **Dardenne J., Valette S., Siauve N., Prost R.** 2009., The Visual Computer (proceedings of CGI 2009), , pp. Volume 25, no. 5, pages 401-410, May.
59. *Marching Cubes: A high resolution 3D surface construction algorithm.* **Lorensen WE, Cline HE.** July 1987, Comput. Graph., pp. Vol. 21, Nr. 4.
60. **Wikimedia Commons.** Wikimedia Commons. [Online] http://commons.wikimedia.org/wiki/Main_Page.
61. *Regularised marching tetrahedra: improved iso-surface extraction.* **Treece G. M., Prager R.W., Gee A.H.** 1999, Computers and Graphics, , pp. 23 (4). pp. 583-598.
62. *A survey of the marching cubes algorithm.* **Newman T.S., Yi H.** 2006, Computers & Graphics , pp. 30: 854–879.
63. *Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations.* **Gilles B, Magnenat-Thalmann N.** 2010 Jun, Med Image Anal. , pp. 14(3):291-302. Epub 2010 Mar 1.
64. *Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi Diagrams,.* **Valette S., Chassery J.-M., Prost R.** 2008, IEEE Transactions on Visualization and Computer Graphics, , pp. Volume 14, no. 2, pages 369-381,.
65. *Variational implicit surface meshing, .* **A. Gelas, S. Valette, R. Prost and W. L. Nowinski.** 2009, Computers and Graphics (proc. IEEE International Conference on Shape Modeling and Applications SMI, pp. Volume 33, no. 3, pages 312-320.
66. *Approach-guided controlled resolution brain meshing for FE-based interactive neurosurgery simulation.* **Audette M 11c, Riviere D, Ewend M, Valette S.** Toronto : Sprenger, 2011. MICCAI MeshMed Workshop. p. accepted.
67. *Automatic recognition of cortical sulci of the human brain using a congregation of neural networks.* **Rivière D, Mangin JF, Papadopoulos-Orfanos D, Martinez JM, Frouin V, Régis J.** 2003 Dec, Med Image Anal., pp. 7(4):403-16.
68. *A framework to study the cortical folding patterns .* **Mangin JF, Rivière D, Cachia A, Duchesnay E, Cointepas Y, Papadopoulos-Orfanos D, Scifo P, Ochiai T, Brunelle F, Régis J.** 2004, NeuroImage, pp. 23: S129–S138.
69. *Fast and memory efficient polygonal simplification.* **Lindstrom P, Turk G.** 1998. , IEEE Visualization, pp. 279-286.
70. **CGAL.** Computational Geometry Algorithms Library. [Online] http://www.cgal.org.
71. *The application of textbook-based surgical ontologies to neurosurgery.* **Audette MA 11b, Yang H, Enquobahrie A, Finet J, Jannin P, Ewend M.** 2011, Int J Comput. Ass. Radiol. & Surg., pp. 6 (Suppl 1):S138–S143.
72. *Fast virtual deployment of self-expandable stents: Method and in vitro evaluation for intracranial aneurysmal stenting.* **Larrabide I., Kim M., Augsburger L., Villa-Uriol M.C., Rüfenacht D., Frangi A.F.** s.l. : Med Image Anal., 2010, Vol. ahead of print.

Further Reading

1. *What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures.* **Shewchuk J.** Ithaca, New York : Sandia National Laboratories, September 2002. Eleventh International Meshing Roundtable . pp. pages 115-126.
2. *Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations.* **Gilles B, Magnenat-Thalmann N.** 2010 Jun, Med Image Anal. , pp. 14(3):291-302. Epub 2010 Mar 1

Acknowledgements

The authors wish to thank the all of the researchers and publications who generously contributed images to this chapter.