

Point-based Methods for Medical Modeling and Simulation

Rifat Aras, Yuzhong Shen, and Michel Audette

Department of Modeling, Simulation, and Visualization Engineering, Old Dominion University
Norfolk, VA

(raras001, yshen, maudette)@odu.edu

ABSTRACT

Physically accurate soft tissue models find great use in medical modeling and simulation. When modeled as a continuum, soft tissue models typically rely on mesh-based techniques that are well studied. When the problem involves extreme deformations or discontinuities in the form of either material discontinuities or spatial discontinuities such as cuts and resections, the continuum assumption underlying the mesh-based methods disappears. In these cases, meshless (point-based) methods present several advantages over the former. In this paper, we present a framework for point-based simulation and cutting of soft-tissue to be used in the context of medical surgical simulation setting. The presented framework is implemented as part of the open-source medical simulation library SOFA, which focuses on real-time interactive simulations. The continuum equation of the deformable body is discretized using the meshless moving least square (MLS) based approximation scheme. The cutting operation is realized with a novel easy-to-use intrinsic meshless enrichment technique.

ABOUT THE AUTHORS

Rifat Aras received his B.S. and M.S. degrees in Computer Engineering from Bilkent University, Ankara, Turkey. His research interests include image processing, computer graphics, serious games, and virtual reality. He is currently a Ph.D. candidate of Modeling, Simulation, and Visualization Engineering at Old Dominion University.

Yuzhong Shen received his B.S. degree in Electrical Engineering from Fudan University, Shanghai, China, M.S. degree in Computer Engineering from Mississippi State University, Starkville, Mississippi, and Ph.D. degree in Electrical Engineering from the University of Delaware, Newark, Delaware. His research interests include computer graphics, visualization, serious games, signal and image processing, and modeling and simulation. Dr. Shen is currently an Associate Professor of the Department of Modeling, Simulation, and Visualization Engineering and the Department of Electrical and Computer Engineering of Old Dominion University. He is also affiliated with Virginia Modeling, Analysis, and Simulation Center (VMASC). Dr. Shen is a Senior Member of IEEE.

Michel Audette, PhD, did all of his university studies in Montreal, Canada: his Bachelor's in Electrical Engineering (1986) and PhD in Biomedical Engineering (2002) at McGill University, as well as his Master's in Electrical Engineering (1992) at Ecole Polytechnique. His PhD thesis research dealt with the application of range-sensing, which he introduced to the medical image community, to brain shift estimation for accurate neurosurgical navigation. His interest in medical simulation, particularly patient-specific anatomical modeling for neurosurgery simulation, dates back to postdoctoral research in Tsukuba, Japan from 2001–2005 and Leipzig, Germany from 2006–2008. He has continued this work through a phase I SBIR while at Kitware, Inc. (Chapel Hill, NC; Nov. 2008–June 2011), and through his current position as faculty member of the Modeling, Simulation and Visualization Engineering Department at Old Dominion University (July 2011–present). His main research areas are i) modeling of anatomy and therapy for surgery simulation, with a strong emphasis on neuro- and orthopedic surgery, ii) model-based surgical planning and intraoperative navigation, and iii) the application of areas i and ii for surgical robotics development. He is a strong proponent of the application of open-source software, such as ITK and SOFA, to medical simulation.

Point-based Methods for Medical Modeling and Simulation

Rifat Aras, Yuzhong Shen, and Michel Audette

Department of Modeling, Simulation, and Visualization Engineering, Old Dominion University
Norfolk, VA
([raras001](mailto:raras001@odu.edu), [yshen](mailto:yshen@odu.edu), [maudette](mailto:maudette@odu.edu))@odu.edu

1. POINT-BASED METHODS FOR DEFORMABLE OBJECT MODELING

The ability to model and manipulate deformable objects is essential to many application areas. In order to fulfill the requirements of different use-cases, deformable object modeling has been studied across a range of paradigms. In this section, we are presenting some of the pioneering work related to deformable object modeling. After discussing the adopted point-based discretization of the object and a framework for deformable object modeling, the next section presents a methodology to handle discontinuities of deformable objects in this framework.

1.1. Previous Work

Approaches for modeling object deformation range from non-physical methods to physically realistic methods based on continuum mechanics. The former category of methods makes use of one or more control points or shape parameter values. These are typically adjusted in a user-friendly manner to obtain the desired deformation of the object. Physically-based methods on the other hand, account for the effects of external and internal forces, material properties, and environmental boundary conditions on object deformation.

The early work of Sederberg and Parry (Sederberg & Parry, 1986) presented a technique for deforming solid geometric models in an intuitive free-form manner. In this work, the deformations were based on interpolating trivariate Bernstein polynomials, and could be applied either globally or locally with volume preservation by encapsulating the target object in a lattice of control points. The control points of the enclosing lattice were manipulated in an intuitive way to achieve the desired deformation of the object. Another type of non-physical methods is Shape Matching / Position-Based methods. These are geometrically motivated, and on the contrary to the physically-based methods, they resolve the dynamics of a deformable object through geometric constraints and distances from current to target positions instead of energies and forces. In the work of Müller et al. (Matthias Müller, Heidelberger, Teschner, & Gross, 2005), an object, which is composed of individual particles, kept track of two configurations: the original shape and the current shape. At each time step, target positions for each particle were calculated by matching the original shape of the object to the current shape. Then, the distances between the corresponding particle locations of the matched shape and the current shape were used to pull the particles towards their target positions. Position-based methods are not as accurate as physically-based methods, but they provide visually plausible results, which make them ideal for virtual reality applications and games.

Free-form deformation and the Shape Matching technique are approximate and simple methods for deforming solid objects; however they lack the physical basis, and this is grounds for excluding them as options for realistic simulations with medical focus. An alternative to non-physical approximations is to use mass-spring models (Waters & Terzopoulos, 1990) and membrane based approximations that utilize spring networks (Van Gelder, 1998). A mass-spring network is composed of vertices and edges, in which each edge is realized as a spring that connects vertices pair-wise, and each vertex is idealized as a point mass. Although these constructs employ physical equations like Hooke's law, it is difficult to reproduce specific elastic material properties even with very careful distribution of spring stiffness throughout the network. Moreover, being stiff systems, mass-spring networks suffer from poor numerical stability.

A continuum model typically relies on an underlying mesh structure either in 2D or 3D depending on the nature and the requirements of the problem. A breadth-first classification of mesh-based continuum models includes Finite Element Methods (Bathe, 1996), Finite Volume Methods (Eymard, Gallouët, & Herbin, 2000), and Finite Difference Methods (Mitchell & Griffiths, 1980). Among these, the finite element method has received particular interest in the biomechanic modeling community.

The early work of Bro-Nielsen discussed a fast adaptation of finite element modeling to satisfy speed and robustness requirements in a surgical simulation setting (Bro-Nielsen, 1998). The body part was modeled as a 3D linear elastic solid that consisted of particles, which were deformed into a new shape when forces were applied to the elastic solid. In this work, the author incorporated a technique called condensation. In the finite element modeling context, condensation translates into obtaining a more compact version of the system model by rearranging or eliminating the terms of the matrix equations. For example, for a single element, the displacement degrees of freedom at a node in the internal region of the element can be condensed out because they are not used in the interelement continuity definition (Bathe, 1996). In a macro level, for a volumetric finite element system, masses of the internal nodes can be lumped to the surface nodes, and the equations can be arranged accordingly to only consider finite element nodes on the surface of the model. Accuracy of the condensation procedure largely depends on the redistribution quality of the masses, in case of a non-optimal distribution, solution accuracy can be adversely affected (Bathe, 1996).

A number of recent techniques have addressed the fidelity versus efficiency trade-off. Another important work in the area is the finite element model based on Total Lagrangian Explicit Dynamics (TLED) by Miller et al. (Miller, Joldes, Lance, & Wittek, 2007). The difference between the TLED based finite element model and other approaches is the former's use of the original reference configuration of the object to calculate the stress and strain tensors during a simulation step. As a result of expressing computations in the reference coordinates, the authors were able to pre-compute spatial derivatives. The pre-computation of the spatial derivatives leads to efficiency in terms of computational resources, while being capable of handling geometric and material non-linearities. In their work, the authors employed the central difference based explicit integration rather than the implicit integration scheme. With this choice, they were able to avoid solving the set of non-linear algebraic equations that are required by the implicit integration at each time step. However, the use of explicit integration brings limitation on the time step size. The authors justified their implementation choice by stating that the relatively lower stiffness (Young's modulus) value of the soft tissue relaxes the time step limitation considerably compared to the typical simulations involving more stiff material like steel or concrete.

Another attempt to increase the computational efficiency of the elastic model in the context of interactive simulation was discussed in the work of Marchesseau et al. (Marchesseau, Heimann, Chatelin, Willinger, & Delingette, 2010). The authors presented a new discretization method called Multiplicative Jacobian Energy Decomposition (MJED), which allows the simulation to assemble the stiffness matrix of the system faster than the traditional Galerkin FEM formulation. The authors reported computation accelerations of up to five times for the St. Venant Kirchhoff materials.

1.2. Continuum Elasticity Theory

Continuum elasticity theory describes the mechanical behavior of the material that is modeled as a continuous medium rather than as discrete particles. The matter is assumed to be continuously distributed and fills the entire region of space that it occupies. Compared to simpler methods, continuum-based approaches offer a significant advantage, which is the convergence of the discrete solution to the continuous solution as the granularity of the discretization goes to zero. In addition to this consideration, the material properties are represented with well-established rheological properties such as Young's modulus and Poisson's ratio, in contrast to the ad-hoc fine-tuning requirement of stiffness values of the mass-spring networks.

In continuum elasticity, deformation of a body is analyzed by studying the three major quantities *displacement*, *strain*, and *stress*. These quantities can be formulated either with the Lagrangian formulation, or the Eulerian formulation. Eulerian formulation is usually used in the analysis of fluid mechanics problems, where the attention is mostly focused on the motion of the material through a regular grid of volume (Bathe, 1996). Therefore, following the Lagrangian formulation, configurations are defined as sets containing the positions of all particles of the continuum. When the object undergoes deformation, the *current* configuration of the body is changing continuously. It is natural to keep the original positions of the particles, called the *reference* configuration, and define the deformation as the transformation of a body from its *reference* configuration to its *current* configuration (Figure 1).

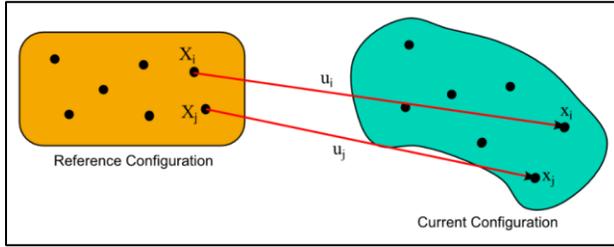


Figure 1. In continuum elasticity, a deformable body is represented in two configurations.

Strain, in the Lagrangian formulation, is a unitless measure of relative deformation representing the displacement between the particles of the current configuration and the reference configuration. Any strain of a solid material generates an internal elastic stress, which expresses the restoring internal forces that the neighboring particles exert on each other. The relation between stress and strain quantities is defined by the so called constitutive equations. In purely elastic materials the deformation of the object is recovered to its reference configuration after the stress is removed.

In the continuum elasticity modeling scheme, a deformable object is defined in its reference configuration with additional material parameters that are part of the underlying constitutive equations. The points inside the domain of the deformable body at its reference configuration are located by material coordinates $\mathbf{X} = (X, Y, Z)$, whereas the points at the current configuration are located by world coordinates $\mathbf{x} = (x, y, z)$. The displacement vector field is therefore defined as

$$\mathbf{u}(\mathbf{X}) = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} x - X \\ y - Y \\ z - Z \end{bmatrix} = \mathbf{x} - \mathbf{X}. \quad (1)$$

The elastic strain tensor ϵ is computed from the spatial derivatives of the displacement field $\mathbf{u}(\mathbf{X})$. ϵ is a 2nd-rank tensor in three dimensions, therefore having 3×3 symmetric elements

$$\epsilon = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{xy} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{xz} & \epsilon_{yz} & \epsilon_{zz} \end{bmatrix}. \quad (2)$$

The components of ϵ are computed by using Green's nonlinear strain tensor equation

$$\epsilon = \nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u}^T \nabla \mathbf{u}, \quad (3)$$

where the gradient of the continuous displacement vector field $\nabla \mathbf{u}$ is essentially the derivatives of (u_x, u_y, u_z) with respect to (X, Y, Z) arranged in Jacobian format

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u_x}{\partial X} & \frac{\partial u_x}{\partial Y} & \frac{\partial u_x}{\partial Z} \\ \frac{\partial u_y}{\partial X} & \frac{\partial u_y}{\partial Y} & \frac{\partial u_y}{\partial Z} \\ \frac{\partial u_z}{\partial X} & \frac{\partial u_z}{\partial Y} & \frac{\partial u_z}{\partial Z} \end{bmatrix}. \quad (4)$$

The stress quantity is the directional force per unit area applied to a plane. Like the strain measurement, stress is also represented by a 3×3 tensor, relating the three-dimensional force vector with the normal of the plane

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix}. \quad (5)$$

The force per area applied to a plane with normal \mathbf{n} is then obtained by multiplying the stress tensor σ by \mathbf{n} .

Both the strain and stress tensors are symmetric with six independent components each. For the selection of the stress tensor, the concept of being work-conjugate becomes important. A strain tensor and a stress tensor are said to be work-conjugate when the stress measure increments multiplied by the strain measure increments give the rate of work done by the forces. Work-conjugate property is a requirement for stability and correct solutions for certain types of problems (Ji, Waas, & Bazant, 2010). For the Green-Lagrangian strain tensor, an appropriate stress tensor is

the second Piola-Kirchoff stress tensor (Bathe, 1996). For an isotropic linear-elastic material, the strain tensor is mapped to the stress tensor by the 6×6 \mathbf{C} matrix that approximates the material properties and its elements are composed of two independent coefficients, Young's Modulus (E) describing the elastic stiffness, and Poisson's Ratio (ν) describing the amount of volume conserved when the material is undergoing deformation

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{zx} \end{bmatrix} \quad (6)$$

The internal elastic forces can be derived through the so-called *strain energy formulation*. When the continuum is undergoing deformation, the energy stored in the system is called strain energy. For isotropic linear materials this energy term can be written as

$$U = \frac{1}{2} \sigma \cdot \epsilon, \quad (7)$$

where $\sigma \cdot \epsilon$ is defined as

$$\sigma \cdot \epsilon = \sum_{i=1}^3 \sum_{j=1}^3 \sigma_{ij} \cdot \epsilon_{ij}. \quad (8)$$

Strain energy density is a function of the displacement field, therefore, the elastic force per unit volume is the negative directional derivative of the strain energy density with respect to the displacement field.

1.3. Point-Based Discretization

Mesh-based discretization techniques such as FEM have dominated the field of computational mechanics in the past several decades. They have been widely used for modeling physical phenomena such as elasticity, heat transfer, and electromagnetism and they heavily rely on the assumption of a continuous domain. FEM is also not well suited to problems involving extreme mesh distortions that result in degenerate element shapes, moving discontinuities that do not align with the element edges such as propagating cracks, and advanced material transformations such as melting of a solid or freezing. To address these issues, significant interest has been developed towards a different class of methods for solving PDEs, namely meshless or mesh-free methods (Chen, Lee, & Eskandarian, 2006; Nguyen, Rabczuk, Bordas, & Duflo, 2008). Mesh-based methods divide the deformable body into tightly connected finite-sized elements. Meshless methods, on the other hand, represent a deformable object by a set of points, whose influence is distributed around them by a weight function (Figure 2).

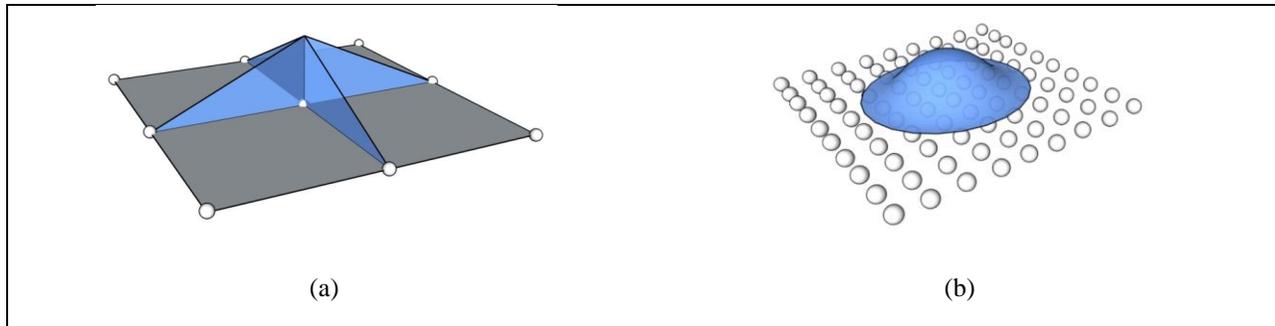


Figure 2. Influence of a node to its neighbors in (a) mesh-based techniques, and in (b) meshless techniques.

The very first meshless approach dates back to 1977 (Lucy, 1977) and proposed a smoothed particle hydrodynamics (SPH) method that was used to model theoretical astrophysical phenomena such as galaxy formation, star formation, stellar collisions, and dust clouds. Its Lagrangian formulation allowed diverse usage areas besides astrophysics such as fluid flow, ballistics, volcanology, and oceanography (Sukumar, 2002). The discrete SPH form can be written as

$$u(x) = \sum_I w_I(x) u_I \Delta V_I \quad (9)$$

where $u(x)$ is the approximation of the field variable at the independent variable location x , I is the set of nodes that has the location x inside their domain of influence, $w_I(x)$ is the weight of the node I at x , u_I is the value of the field variable at the node I , and ΔV_I is the size of the domain of the node I .

Although the SPH method eliminates the necessity of a mesh structure and allows the solution of unbounded problems, it also has its limitations. Because of its approximation scheme based only on the weight function, it fails to reproduce even first order polynomials, resulting in severe consistency problems (Chen et al., 2006). To alleviate this problem, methods that utilize moving least squares (MLS) approximations have been developed. The first work that used MLS approximations in a Galerkin method is the work of Nayroles et al. (Nayroles, Touzot, & Villon, 1992), which was refined by Belytschko et al. (Belytschko, Lu, & Gu, 1994) and named Element-Free Galerkin (EFG) method. This class of methods, different from the SPH method, use shape functions in approximations that are essentially corrected versions of compact supported weight functions

$$u(x) = \sum_I \phi_I(x) u_I. \quad (10)$$

Here, the sum of shape functions $\phi_I(x)$ for a given approximation equals to 1, which is known as the *partition of unity* paradigm (Nguyen et al., 2008).

The shape functions are obtained by first representing the approximation as a product of a polynomial basis and a vector of unknown coefficients. Then, a functional is created by taking the weighted sum of square of the approximation error. By taking the derivative of this functional with respect to the unknown coefficients and setting it to zero for minimizing the approximation error, we obtain a set of equations that are reorganized to solve for the MLS shape functions. The order of consistency of the MLS approximation scheme depends on the order and completeness of the used basis function. If the basis function used in the approximation is a complete polynomial of order k , then the MLS approximation is said to be k^{th} order consistent. This fact makes the MLS based approximations more consistent than the SPH method.

Another technique that has used the MLS approximation is the work of Mueller et al. (M. Müller et al., 2004), which forms the basis of the point-based method discussed in this paper. In the former work, the authors calculated the spatial derivatives of the deformation gradient only at the particle locations similar to the meshless collocation methods. This technique is capable of simulating a wide range of material properties from very stiff materials to soft ones, while also handling plastic deformations as well.

Node distribution is the first step in the presented point-based discretization algorithm, which supports both regular and hierarchical distribution of the nodes through the simulation domain. In the case of a simulation domain with a regular geometric shape, a regular distribution of the nodes is the natural choice. On the other hand, if the simulation domain has a complex geometry, which is the general case, a regular distribution becomes inapplicable. In this case, we sample the volumetric simulation domain bounded by the complex boundary surface by hierarchically sampling the volume. In the work of Pauly et al. (Pauly et al., 2005), the authors used a balanced octree data structure to distribute the nodes inside the volume. In our approach, we first tetrahedralize the simulation domain with well-established computational geometry libraries like TetGen (Si, 2006) and CGAL (Boissonnat, Devillers, Teillaud, & Yvinec, 2000) and then use the set of vertices of the tetrahedra as the meshless node locations. In this way, similar to graded finite element meshing techniques, we are able to obtain a higher node density close to the domain boundary and fewer nodes towards the interior of the volume where the material is homogeneous.

Meshless methods represent a deformable body by a cloud of particles with overlapping support domains. Quantities such as mass, volume, support size, strain, and stress are stored and updated per particle for the duration of the simulation. In this work, the support domains of the particles are spherical and their radii are computed by finding the average distance of the central node to its k -nearest neighbors. For efficient neighborhood search purposes, a k -d tree data structure is used

Weight (kernel) function in the meshless method context describes the way meshless nodes affect each other and how the material values of the continuum such as mass, volume, and density are distributed among the nodes. The neighboring particles that fall inside the support domain of a central particle are weighted using the polynomial kernel function

$$w(r_{ij}) = \begin{cases} 1 - 3r_{ij}^2 + 3r_{ij}^4 - r_{ij}^6, & r_{ij} \leq 1 \\ 0, & r_{ij} > 1 \end{cases} \quad (11)$$

with $r_{ij} = \frac{\|\mathbf{x}_j - \mathbf{x}_i\|}{h_j}$, where \mathbf{x}_j and \mathbf{x}_i are the current locations of the neighboring and central particles, respectively and h_j is the support radius of the central particle j (Figure 3).

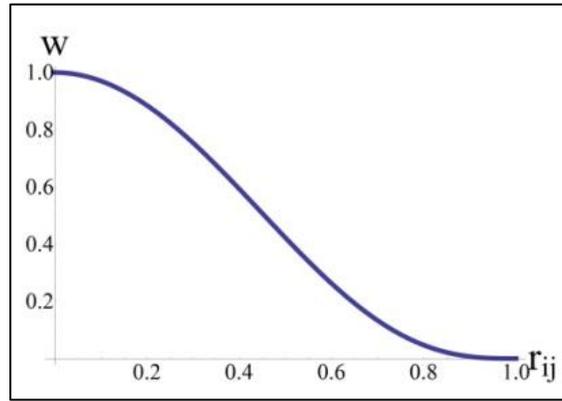


Figure 3. The weight kernel used in our MLS approximations. Weight functions are critical to solution accuracy and stability. Weight functions should be continuous and positive in their support.

The mass and density of a meshless node are assigned at the beginning and kept fixed throughout the simulation. The mass values are initialized with

$$m_i = s \bar{r}_i^3 \rho, \quad (12)$$

where ρ is the material density value, \bar{r}_i is the average distance of the i^{th} node to its k -nearest neighbors, and s is a scaling factor that is chosen so that the average of the assigned densities is close to the actual material density. The assigned mass value of a meshless node is distributed around the node with the kernel function. Therefore the density of a meshless node is calculated after the mass allocation step by taking the weighted average of the masses of the neighboring nodes

$$\rho_i = \sum_j m_j w(r_{ij}). \quad (13)$$

In our approach, spatial integration is performed through the nodal integration technique. Compared to other spatial integration techniques that utilize a background mesh or grid with multiple integration points per region, nodal integration is fast and efficient with the added disadvantage of decreased stability. We calculate the spatial derivatives of the deformation gradient only at the particle locations similar to the meshless collocation methods.

2. HANDLING DISCONTINUITIES IN POINT-BASED CONTINUUM MODELS

In engineering problems, discontinuities are frequently found. In these cases, the continuum assumption of the elastic theory is undermined, which typically require special treatment to ensure the correct solution to the system. Discontinuities may be caused when the continuum domain is composed of different material types or when there is a spatial gap in the continuum such as a cut.

2.1. Overview

In meshless methods, there are three main classes of techniques to treat discontinuity of the field variable (displacement). These techniques are: (1) modification of the weight function, (2) intrinsic enrichment of the basis of the approximation, and (3) the techniques based on extrinsic enrichment. Discontinuity treatment in meshless methods has been studied within a wide range of approaches such as visibility criterion, diffraction / transparency methods, and H- and P-Refinements (Chen et al., 2006).

The visibility method is an example for the techniques that modify the weight function. In this method, the cut segment is treated as an opaque object and the influence of a node to another one is decided by drawing a line between the two, and testing whether the line intersects with the cut segment or not. Although being simple in nature, this method can lead to incorrect discontinuity calculations along the lines that pass through the cut tips. Another disadvantage of this method is that it cannot be used to treat non-convex boundaries. The diffraction method follows the same steps as the visibility criterion, but improves the technique by passing the ray around the cut tip and calculating the influence between two nodes via the ray length. The diffraction method requires complex computations of the bending rays and the technique's extension into three dimensions is even more complex. The transparency method addresses these shortcomings by simply decreasing the opacity of the cut near its tip. The transparency method is easier to extend into three dimensions (Organ, Fleming, Terry, & Belytschko, 1996).

2.2. Modeling Discontinuities in 2D

Barbieri et al. (Barbieri, Petrinic, Meo, & Tagarielli, 2011) proposed an enrichment technique based on distance function for handling discontinuities with multiple boundaries. Their method processes cuts as piecewise linear segments and calculates the absolute distance of a meshless node to these segments. The enrichment function obtained from the distance field is then multiplied with the weight kernel of the node. Compared to competing techniques like the visibility criterion, this approach requires less computation and also easily extensible to three dimensions. The distance function is computed in the local coordinate system of the cut piece.

The 2D distance function for a given coordinate (x, y) , can therefore be computed in terms of the local coordinates (s, t) as

$$d_2(x, y) = \sqrt{d_1^+(t)^2 + s^2} \quad (14)$$

where $d_1^+(t)$ is the positive part of the 1D signed distance function $d_s(t)$ for a 1D segment (in the local coordinates) and defined as

$$d_s(t) = \left| t - \frac{t_1 + t_2}{2} \right| - \left| \frac{t_1 - t_2}{2} \right| \quad (15)$$

where t_1 and t_2 are the endpoints of the cut segment in the cut's local coordinate system and

$$d_1^+(t) = \frac{d_s(t) + |d_s(t)|}{2}. \quad (16)$$

When we take the partial derivative of this distance function with respect to the normal coordinate axis s , we obtain a discontinuous function across the segment that is 1 on one side of the crack and -1 on the other side and varies smoothly around the cut.

This technique is easy to implement and also applicable to the existing methods to extend their functionalities. One shortcoming of the technique is the approach it takes on handling multiple cracks. Let h_i be the enrichment value for the i -th crack on a node I , the cumulative enrichment value for the node I that is in the vicinity of n cracks is given by $h = \prod_{i=1}^n h_i$.

Multiplicative application of consecutive enrichments is a practical approach and requires minimum amount of processing, but it may also lead to incorrect weight function modifications and therefore fatal instabilities in the simulation (Figure 4).

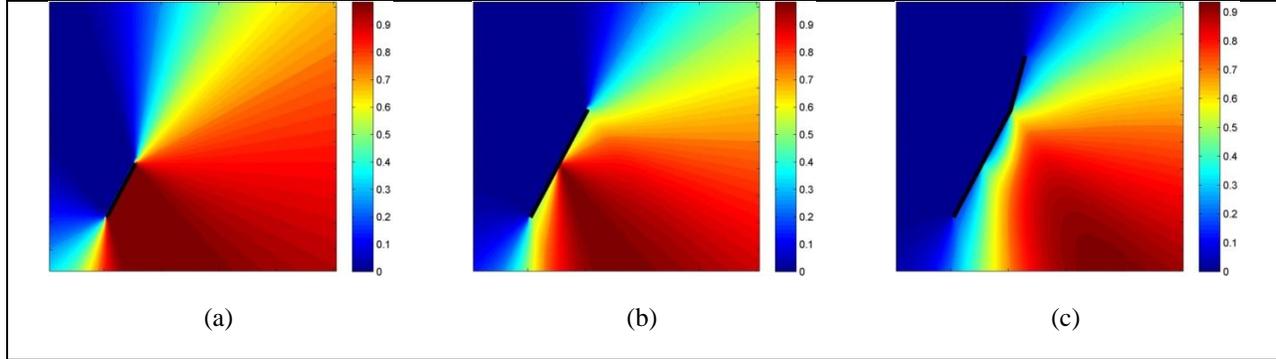


Figure 4. The problem with the original multiplicative enrichment approach. The weight functions of nodes in vicinity of multiple cuts are modified incorrectly.

In order to address these issues, the distance function-based enrichment technique is extended to support consecutive discontinuity fronts in a correct way. In the extended technique, the enrichment values for multiple cuts are evaluated inside a grid structure, therefore conveniently taking the name enrichment grid. For each grid point, the corresponding enrichment value is calculated similar to the original distance function-based technique, though, instead of a multiplicative approach, each grid point is assigned to a specific cut segment region and its enrichment value is calculated with respect to this specific crack segment.

The first step in the enrichment grid algorithm is to set a global coordinate system for calculating the regions of the grid points. In 2D, the coordinate system is defined by enrichment origin point \mathbf{p}_0 with coordinates (x_0, y_0) and θ_0 that is the angle between the horizontal axis of the coordinate system and the positive x-axis. The coordinate system is updated with each propagating cut as

$$\mathbf{p}_0 = \frac{\sum_{i=1}^n w_i \mathbf{p}_i}{\sum_{i=1}^n w_i}, \quad \theta_0 = \frac{\sum_{i=1}^n w_i \theta_i}{\sum_{i=1}^n w_i} \quad (17)$$

where n is the number of segments, w_i is the associated weight with the cut segment i , which is typically the length of the segment in 2D problems, and \mathbf{p}_i and θ_i are the center point and horizontal angle of the i -th cut segment respectively. After setting the global coordinate system for the series of cut segments, each grid point with coordinates (x, y) as well as the endpoints of the cut segments (x_i, y_i) are translated into this new coordinate system to obtain new coordinates (t) and (t_i) by

$$\begin{aligned} t &= \cos(\theta) (x - x_0) + \sin(\theta) (y - y_0) \\ t_i &= \cos(\theta) (x_i - x_0) + \sin(\theta) (y_i - y_0). \end{aligned} \quad (18)$$

With these translated points, the modified $d_1^+(t)$ function is now defined as

$$d_1^+(t) = (t_0 - t) \cdot H(t_0 - t) + (t - t_n) \cdot H(t - t_n), \quad (19)$$

where t_0 and t_n are the t -coordinates of the first and last points of the cut segment series, and H is the Heaviside step function.

The next step to calculate the distance function $d_2(x, y)$ is to set the s -coordinates of the grid points. This is achieved by assigning a cut segment region for each of the grid points by comparing their t -coordinates against the t -coordinates of the cut segment endpoints. A grid point with t -coordinate t' is set to be in the region i when $t' > t_{i-1}$ & $t' < t_i$. For grid points whose t -coordinates are smaller than t_0 and larger than t_n , their regions are set to the first and last regions respectively (Figure 5).

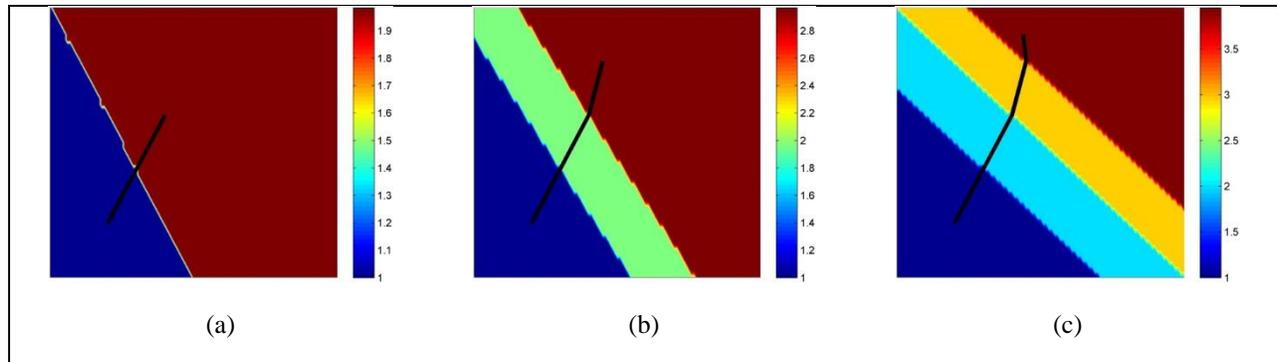


Figure 5. Region assignments for multiple cut segments.

After assigning the region values for the grid points, the s -coordinates are calculated by finding the vertical distance of the grid point to the assigned cut segment. For a grid point with coordinates (x, y) and assigned region i , the s -coordinate is calculated as

$$s = -\sin(\theta_i)(x - x_i) + \cos(\theta_i)(y - y_i). \quad (20)$$

The (t, s) coordinates of the grid points are enough to calculate the extended enrichment function. Using these values, $d_2(x, y)$ is calculated as in equation (14) and its partial derivative with respect to the s -coordinates is taken to obtain the extended enrichment function (Figure 6).

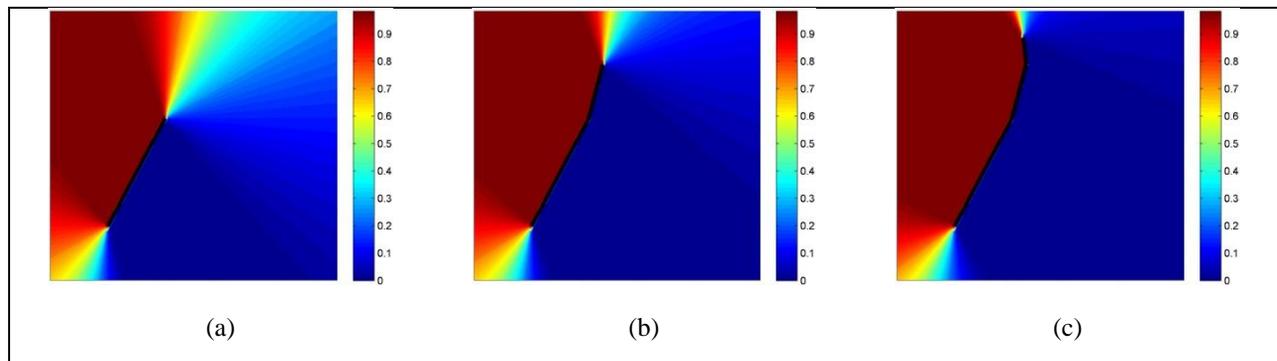


Figure 6. Visualization of the extended enrichment function. Unlike the original approach, this function modifies the weight functions correctly.

3. CONCLUSION AND FUTURE WORK

Soft tissue models typically rely on mesh-based techniques that are well studied with a strong basis. When the problem involves extreme deformations or artificial discontinuities such as material discontinuities or cracks and cuts, the continuum assumption underlying the mesh-based methods disappears. In these cases, we advocate the use of point-based methods for soft-tissue simulation in the context of medical surgical simulation setting. In order to handle discontinuities such as a cut, we have extended the enrichment technique with enrichment grids to provide a computation cost-efficient way of handling multiple discontinuity fronts. Our extended technique, unlike the multiplicative approach, modifies the weight function of the affected meshless nodes in a way such that stability problems are avoided. The proposed technique is easy to extend into three dimensions.

4. REFERENCES

Barbieri, E., Petrinic, N., Meo, M., & Tagarielli, V. L. (2011). A new weight-function enrichment in meshless methods for multiple cracks in linear elasticity. *International Journal for Numerical Methods in Engineering*, n/a-n/a. doi: 10.1002/nme.3313

- Bathe, K.-J. (1996). *Finite element procedures* (Vol. 2): Prentice hall Englewood Cliffs.
- Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2), 229-256.
- Boissonnat, J.-D., Devillers, O., Teillaud, M., & Yvinec, M. (2000). *Triangulations in CGAL*. Paper presented at the Proceedings of the sixteenth annual symposium on Computational geometry.
- Bro-Nielsen, M. (1998). Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3), 490-503.
- Chen, Y., Lee, J. D., & Eskandarian, A. (2006). *Meshless methods in solid mechanics*: Springer Verlag.
- Eymard, R., Gallouët, T., & Herbin, R. (2000). Finite volume methods. *Handbook of numerical analysis*, 7, 713-1018.
- Ji, W., Waas, A. M., & Bažant, Z. P. (2010). Errors caused by non-work-conjugate stress and strain measures and necessary corrections in finite element programs. *Journal of Applied Mechanics*, 77(4), 044504.
- Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82, 1013-1024.
- Marchesseau, S., Heimann, T., Chatelin, S., Willinger, R., & Delingette, H. (2010). Multiplicative jacobian energy decomposition method for fast porous visco-hyperelastic soft tissue model *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2010* (pp. 235-242): Springer.
- Miller, K., Joldes, G., Lance, D., & Wittek, A. (2007). Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in numerical methods in engineering*, 23(2), 121-134.
- Mitchell, A. R., & Griffiths, D. F. (1980). *The finite difference method in partial differential equations*. Chichester, Sussex, England and New York, Wiley-Interscience, 1980. 281 p.
- Müller, M., Heidelberger, B., Teschner, M., & Gross, M. (2005). *Meshless deformations based on shape matching*. Paper presented at the ACM Transactions on Graphics (TOG).
- Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M., & Alexa, M. (2004). *Point based animation of elastic, plastic and melting objects*. Paper presented at the ACM SIGGRAPH/Eurographics symposium on Computer animation.
- Nayroles, B., Touzot, G., & Villon, P. (1992). Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational mechanics*, 10(5), 307-318.
- Nguyen, V. P., Rabczuk, T., Bordas, S., & Duflot, M. (2008). Meshless methods: a review and computer implementation aspects. *Mathematics and Computers in Simulation*, 79(3), 763-813.
- Organ, D., Fleming, M., Terry, T., & Belytschko, T. (1996). Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational mechanics*, 18(3), 225-235.
- Pauly, M., Keiser, R., Adams, B., Dutra, P., Gross, M., & Guibas, L. J. (2005). Meshless animation of fracturing solids. *ACM Trans. Graph.*, 24(3), 957-964. doi: 10.1145/1073204.1073296
- Sederberg, T. W., & Parry, S. R. (1986). Free-form deformation of solid geometric models. *ACM Siggraph Computer Graphics*, 20(4), 151-160.
- Si, H. (2006). TetGen: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. *Weierstrass Institute for Applied Analysis and Stochastic*, Berlin, Germany.
- Sukumar, N. (2002). The Natural Element Method (NEM) in Solid Mechanics. Retrieved 08/01/2012, from <http://dilbert.engr.ucdavis.edu/~suku/nem/>
- Van Gelder, A. (1998). Approximate simulation of elastic membranes by triangulated spring meshes. *Journal of graphics tools*, 3, 21-42.
- Waters, K., & Terzopoulos, D. (1990). *A physical model of facial tissue and muscle articulation*. Paper presented at the Visualization in Biomedical Computing.