Advances in Engineering Software 102 (2016) 40–48

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

An analytic meshless enrichment function for handling discontinuities in interactive surgical simulation

Rifat Aras*, Yuzhong Shen, Michel Audette

Modeling, Simulation, and Visualization Engineering, 1300 Engineering & Computational Sciences Bldg Norfolk, VA 23529, United States

ARTICLE INFO

Article history:
Received 22 March 2016
Revised 10 August 2016
Accepted 31 August 2016

Keywords:
Biomechanics
Enrichment functions
Meshless methods
Surgical simulation
Soft tissue cutting

ABSTRACT

For surgical simulation applications, realistic behavioral modeling of soft tissue is considered to be one of the most significant challenges, because biomechanical soft-tissue models need to reflect the correct elastic response, be efficient in order to run at interactive simulation rates, and be able to support operations such as cuts and sutures. For these reasons, having a usable 3D cutting engine is a significant feature for interactive surgery simulation software. Mesh-based solutions, where the connections between the individual degrees of freedom (DoF) are defined explicitly, have been the traditional approach to soft-tissue biomechanics. However, when the problem under investigation in interactive biomechanics contains a simulated surgical gesture that entails a cut that disrupts the connectivity, the underlying mesh structure has to undergo remeshing operation, and most of the time it causes the performance bottleneck in the simulation. Unlike the tightly-coupled nonoverlapping element composition of the mesh-based solutions, this paper builds an analytic enrichment function on top of a loosely-coupled meshless method for constitutive modeling of elastic soft tissues, where arbitrary discontinuities or cuts are applied to the objects in the context of surgical simulation. Enrichment values for a continuous cut interface are computed and stored inside a grid structure that is accessed by individual meshless nodes in order to update their weight functions. The presented analytic enrichment function is efficient to compute and easy to integrate into existing meshless models. The meshless mechanics code and the enrichment-based cut handling functionalities have been implemented within the open-source simulation framework SOFA.

1. Introduction

Medical education traditionally involves the concept of apprenticeship [1], where novices directly learn from experienced doctors, while gradually taking an increasing role in therapy provided to patients to increase their level of expertise. Although classical apprenticeship program constitutes the basis of the medical education field, relying solely on it is not an optimal education strategy as it requires long training hours to bring the expertise to the desired level and also it is difficult to ensure that trainees experience all types of major cases. Computer-based modeling and simulation practices such as virtual reality surgery simulation have begun to make an impact in order to alleviate the aforementioned shortcomings of the traditional medical education.

Computer-based medical modeling and simulation features both anatomical models and therapy models, both of which represent significant challenges. One of the greatest challenges in building complex therapy models is to capture the accurate response of soft tissue [2–4]. For surgical simulator applications, biomechanical models of human soft tissues have to be accurate, efficient enough to be computed in real time, and able to handle topology-altering operations such as cuts and sutures [5]. Volumetric cutting of soft tissue is essential to interactive surgery simulation and convincing implementation of volumetric cutting is still an active research area in surgical simulation.

This paper presents a new approach for treating material discontinuities such as cuts, which is built on top of a point-based rather than mesh-based method. The introduced cuts are handled through mathematical structures called *analytic enrichment functions*, which are essentially functions that are discontinuous across the cut edge yet smoothly varying around the tip of the cut. Enrichment functions also need to be efficient enough to be computed on-the-fly while a cut is continuously being introduced in the simulation domain. The cut that affects the deformable body is represented as a piecewise linear segment, which defines a local cut-centered coordinate system that is used to compute the associated enrichment function. The enrichment values for a continuous cut series are computed and stored in a grid structure that is called the *Enrichment Grid*. The Enrichment Grid also doubles as a

* Corresponding author.
E-mail addresses: rifat.aras@gmail.com, raras001@odu.edu (R. Aras).

http://dx.doi.org/10.1016/j.advengsoft.2016.08.011
0965-9978/

useful data structure in order to accelerate intra-simulation steps such as intersection queries. The methods and the algorithms described in this paper are implemented as an extension to the popular open-source medical simulation framework Simulation Open Framework Architecture (SOFA) [6]. Our contribution to the SOFA codebase, which was previously lacking a volumetric cutting algorithm, is a significant contribution to the open-source surgery simulation community.

The rest of the paper is organized as follows. Section 2 provides a brief overview of meshless methods in general. Section 3 describes various strategies in handling discontinuities in meshless methods. Section 4 presents our contribution, which improves on previous meshless approaches, and introduces the Enrichment Grid data structure. Section 5 provides the point-based deformable object modeling along with the extended enrichment grid approach. Finally, Section 6 concludes the paper by presenting the final remarks.

2. Previous work on tissue deformation and cutting

Deformable modeling of soft tissue is a continuum elasticity problem, whose numeric solution involves the discretization of a continuous domain into discrete elements. Numerous non-physical and physically-based models have been utilized in order to approximate this solution, which typically rely on an underlying mesh structures 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 mass-spring networks [7], finite element methods [8], finite volume methods [9], and finite difference methods [10]. Among these, the finite element method has received particular interest in the biomechanical 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 [11]. In this framework, the author incorporated a technique called condensation, which translates into obtaining a more compact version of the system model by rearranging or eliminating terms of the matrix equations by simplifying a volume into a system of boundary elements. The accuracy of the condensation procedure largely depends on the redistribution quality of the masses; in case of a non-optimal distribution, the solution accuracy can be adversely affected [8]. Moreover, this type of simplification is incompatible with arbitrary cutting.

Another technique developed to optimize the fidelity versus efficiency trade-off is the finite element model based on Total Lagrangian Explicit Dynamics (TLED) by Miller et al. [12]. 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. The authors employed central differences-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 entails limitations on the time step size in order to ensure the stability of the system. The authors justified their implementation choice by stating that the relatively lower stiffness (Youngs modulus) value of the soft tissue relaxes the time step limitation considerably compared to the typical simulations involving stiffer 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 method proposed by Marchesseau et al. [13]. 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 method utilized an implicit solver with larger time steps, which has the potential of producing more stable simulations, in application areas that involve haptic interactions. The authors reported computation accelerations of up to five times for the St. Venant Kirchoff materials. TLED and MJED methods both rely on pre-computation of simulation variables in order to achieve faster solutions at each time step. Although useful for modeling the elastic response of the deformable body that does not involve topological changes, these pre-computations at the initial configuration of the simulation object would be invalidated when a topology-changing cut is introduced to the system. In other words, TLED and MJED are undermined by interactive cutting requirements.

Free-form deformation lattices, mass-spring networks, and finite element models (FEM) that are composed of tetrahedral/hexahedral elements are all examples of mesh-based models that result in systems with many degrees of freedom (DoFs) that essentially define the total kinematic state of the modeled object. The aforementioned model examples have one property in common, they all define the connectivity information between the DoFs explicitly. When there is a situation that disrupts this connectivity, such as an introduced discontinuity in the form of a cut, the discretization of the continuum needs to be redefined to handle the changes in the connectivity. Various approaches have been proposed to handle these changes caused by cuts. Courtecuisse et al. [14] presented an FEM-based soft tissue deformation methodology that also supports real-time virtual cutting. In the presence of a cut, the topology of the finite elements comprising the simulation object changes along with the simulation-specific matrices. The topology changes in Courtecuisse's implementation were encoded in three types of topology operations: element removal, element subdivision, and element addition. This work benefitted from a GPU-based parallel implementation in order to ensure interactive operation rates.

More recently, Wu et al. [15] discretized the simulation object by using a semi-regular hexahedral finite element grid. The volume was partitioned using an octree, and the face-adjacent cells of the octree were linked together. The advantage of this discretization is the ability to update the topology of the elements in an efficient way, when a cut is being introduced to the simulation domain, by marking the links between the affected elements as disconnected. The octree was refined dynamically along the cut surface in order to retain fine detailed cuts. The authors employed several approximations of the deformable model, such as the concept of Composite Finite Elements (CFEs), in which smaller neighboring hexahedral elements are grouped together to form larger elements, thus decreasing the number of DoFs significantly. With this CFE-based approximation and a multigrid implicit solver [15], the authors were able to achieve simulation rates of 15 frames per second during the cutting operation.

3. Meshless methods overview

Mesh-based methods such as FEM have been widely used for modeling physical phenomena such as elasticity, heat transfer, and electromagnetism, while relying on the assumption of a continuous domain. However, FEM is 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 or cuts [17], and advanced

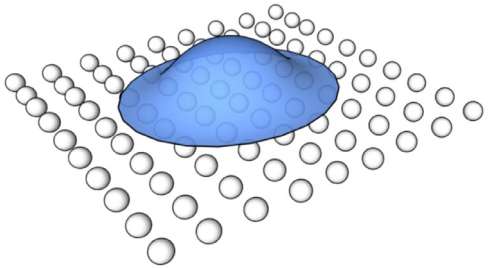

Fig. 1. In meshless methods, the continuum is represented as a set of points, that influence others inside their domains according to a weight function.

material transformations such as melting of a solid or freezing [16]. To address these issues, significant interest has been developed towards a different class of methods for solving differential equations, namely meshless or mesh-free methods [18]. 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 through the domain of the individual points [19]. The domain of a point spatially overlaps several other points as depicted in Fig. 1.

Meshless methods are characterized by the approximation of the field variables such as displacement u. The work of Nayroles et al. [20] used moving least square (MLS) approximations in a Galerkin method, which was later refined by Belytschko et al. [21] and named the Element-Free Galerkin (EFG) method. This class of methods uses shape functions $\phi(x)$ in approximations that are essentially corrected versions of weight functions with compact support

$$u^h(x) = \sum_I \phi_I(x) u_I. \tag{1}$$

Here, $u^h(x)$ is the approximated value of displacement field variable at location x, I is the set of meshless nodes whose influence (value of the associated weight function) at x is positive, and u_I are the displacement values kept at the meshless node set I. A desired property of meshless approximations is that the sum of the shape functions $\phi_I(x)$ for the neighborhood of a given meshless node equal to 1, which is known as the *Partition of Unity* paradigm [18]. The Partition of Unity concept is an important element in meshless methods to improve the consistency of an approximation.

The shape functions $\phi_I(x)$ are obtained by first representing the approximation as the product of a polynomial basis $\mathbf{p}^T(x) = [1 \quad x \quad x^2]$ and a vector of unknown coefficients $\mathbf{a}(x) = [a_0(x) \quad a_1(x) \quad a_2(x)]$

$$\tilde{u}(x_I, x) = \mathbf{p}^T(x_I)\mathbf{a}(x). \tag{2}$$

In this product, x_I are the sample points that contain the values used to reconstruct the continuous function, and x is the point where this reconstructed function is to be evaluated at. With this approximation, a functional is created by taking the weighted sum of square of the approximation error

$$J(x) = \sum_{I=1}^{N} w(x - x_I)[\tilde{u}(x_I, x) - u_I]^2, \tag{3}$$

where $w(x_I - x)$ is the weight value computed using the Euclidean distance between the locations x and x_I with respect to support size d_I of the meshless node I. A typical weight function is the quartic spline weight function [18],

$$w(r) = \begin{cases} 1 - 6r^2 + 8r^3 - 3r^4, & r \le 1 \\ 0, & r > 1 \end{cases}, \tag{4}$$

where r is defined as

$$r = \frac{\|x_I - x\|}{d_I}. \tag{5}$$

After expanding the $\tilde{u}$ term, the functional becomes

$$J(x) = \sum_{I=1}^{N} w(x - x_I)[\mathbf{p}^T(x_I)\mathbf{a}(x) - u_I]^2. \tag{6}$$

In order to minimize the approximation error, the derivative of this functional with respect to the unknown coefficients $\mathbf{a}(x)$ is taken and set to zero. This operation results in the following equation

$$\sum_{I=1}^{N} w(x - x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I)\mathbf{a}(x) = \sum_{I=1}^{N} w(x - x_I)\mathbf{p}(x_I)u_I. \tag{7}$$

Using this equation, we can solve for the unknown coefficients $\mathbf{a}(x)$ and rewrite the approximation as

$$u^h(x) = \mathbf{p}^T(x)[\mathbf{A}(x)]^{-1} w(x - x_I)\mathbf{p}(x_I)\mathbf{u}, \tag{8}$$

where $\mathbf{A}(x)$ is called the moment matrix and defined by

$$\mathbf{A}(x) = \sum_{I=1}^{N} w(x - x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I). \tag{9}$$

The shape functions are therefore equal to

$$\phi_I(x) = \mathbf{p}^T(x)[\mathbf{A}(x)]^{-1} w(x - x_I)\mathbf{p}(x_I). \tag{10}$$

The consistency of the MLS approximation scheme depends on the order and completeness of the chosen basis function, an important consideration to ensure stability. If the basis function used in the approximation is a complete polynomial of order k, then the MLS approximation is said to be kth-order consistent. In other words, an approximation that is kth-order consistent can reproduce a kth-order polynomial exactly. Another technique that has used the MLS approximation is the work of Müller et al. [22]. In their framework, the authors calculated the spatial derivatives of the deformation gradient only at the particle locations. This approach is similar to the meshless point collocation methods [23] that discretize the differential equations only at the meshless nodes. A typical characteristic of meshless point collocation methods that differentiate them from other meshless methods is their truly meshless nature as they do not require an underlying mesh structure for field variable approximation or spatial integration. As described above in the meshless method steps, instead of converting the governing differential equations into their weak form and integrating over a sub-domain, point collocation methods directly discretize the strong-form of the governing differential equation at the meshless nodes. The advantage of the point collocation methods is their computational efficiency as the shape functions do not need to be evaluated at the integration points, at the expense of difficulty in imposing natural boundary conditions, where the field variables take the specified values. The technique described by Müller et al. is capable of simulating a wide range of material properties from very stiff materials to soft ones, while also being able to handle plastic deformations as well. Another point collocation method study in the area of virtual minimally invasive surgery simulation is the work of Banihani et al. [28] that utilizes a proper orthogonal decomposition (POD) model along with a MLS approximation scheme to obtain real-time solutions to nonlinear hyperelastic soft-tissue simulation problems.

4. Convergence study through Hertzian contact theory

The Hertzian theory of non-adhesive elastic contact [29] defines analytical solutions for the interaction of elastic half-spaces with simple shapes in terms of applied force and object indentation. For

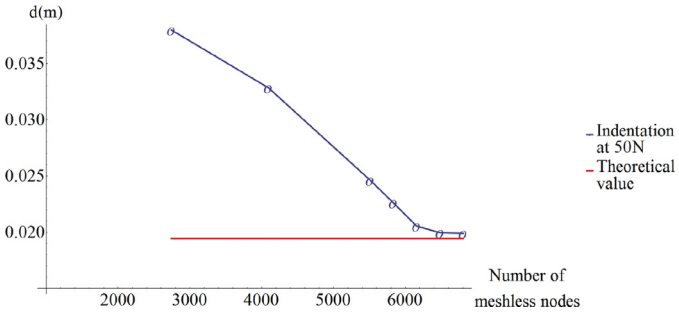

Fig. 2. Convergence of the indentation value with increasing number of meshless nodes.

example, the amount of indentation of an elastic half-space under a spherical load is given by

$$f = \frac{4}{3}E^*\sqrt{r}d^{\frac{3}{2}} \tag{11}$$

where f is the vertical force applied on the spherical load, r is the radius of the spherical load, d is the indentation amount, and E^* is the combined Youngs modulus of the two materials and calculated using the Youngs moduli (E_1, E_2) and Poissons ratios (v_1, v_2) of the two materials as

$$\frac{1}{E^*} = \frac{1-v_1^2}{E_1} + \frac{1-v_2^2}{E_2}. \tag{12}$$

The Hertzian theory assumes 1) small strains within the elastic material, 2) much smaller area of contact compared to the areas of the objects in contact, and 3) continuous and frictionless contact surfaces. There have been numerous finite element analysis studies about the Hertzian theory that use both research and commercial finite element code [30–33].

For the meshless collocation method with nodal integration, an explicit time integration scheme was used with a time step of 0.001 s without any stability problem. For the SOFA FEM implementation, implicit integration with a time step of 0.01 s or greater were used. The calculations were performed within the SOFA application on a single Intel Core i5 CPU running at 2.67 GHz with 16GB of RAM under Windows 7 operating system. The SOFA FEM implementation took 195ms of calculations per time step, whereas the meshless method consumed 20.11ms for calculations per time step. Therefore, the meshless collocation implementation in SOFA (along with other SOFA related operations such as collision detection) is roughly 25 times slower than the real-time operation, which is slightly better than the 30 times slower performance reported by the Meshless TLED algorithm [34]. The calculation speed of the meshless collocation algorithm is governed by the number of particles and the number of neighbors assigned to each particle.

A mesh convergence study for the meshless collocation method was also performed by investigating the convergence of the indentation amount to the theoretical value for a fixed amount of force (Fig. 2). After around 6000 particles, the indentation value converges to the theoretical indentation value.

5. Handling discontinuities in meshless methods

In engineering problems, discontinuities are a common occurrence. In these cases, the continuum assumption of the elastic theory is undermined, which typically requires 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. The cuts that modify the elastic response of the deformable objects are called strong discontinuities in the continuum elasticity mechanics field. In meshless methods, there are three

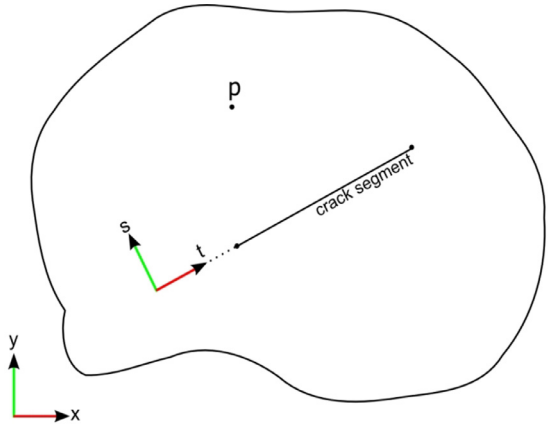

Fig. 3. The discontinuity caused by a cut segment is defined at the local coordinate frame of the cut segment.

main classes of techniques to treat discontinuity of the field variable (displacement). These techniques are: (1) modification of the weight function $w(x)$ of the affected meshless nodes, (2) intrinsic enrichment of the basis $\mathbf{p}^T(x)$ of the approximation, and (3) techniques based on extrinsic enrichment [16]. The *visibility* method is an example of techniques that modify the weight function. In this method, the cut segment is treated as an opaque object and the influence of a node on a point in the domain is decided by shooting a ray between the node and the point in question, and testing whether the ray intersects with the cut segment or not. Although it is simple in nature, this method can lead to incorrect discontinuity calculations along the rays that pass through the tips of the cut. 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 tip of the cut and by calculating the influence of a node on a point via the ray length. The diffraction method requires complex computations of the bending rays and its extension into three dimensions is even more complex [24].

Enrichment functions have been introduced to the classical FEM approach by Moes et al. [25] and incorporated into several studies afterwards. Kaufmann et al. [26] presented a method that employed harmonic enrichment functions for simulating detailed cutting of thin shells. These functions were computed by solving a Laplace equation that is subject to boundary conditions that correspond to the cut, and stored in 2D texture elements in higher resolutions than the underlying finite element discretization. Despite being a 2D enrichment technique, the high computational cost of the solution of the Laplace equation prevented this work from achieving interactive rates.

In contrast with the previous approach, Duflot and Nguyen-Dang [35] proposed an analytical enrichment function in 2D, which was later on developed by Barbieri et al. [27] as an analytical enrichment function in 2D that is easy to compute and extend into 3D, based on a distance function calculated on the local coordinate frame of the cut segment. Their method processed cuts as piecewise-linear segments and calculated the absolute distance of a meshless node to these segments. The enrichment function obtained from the distance field was then multiplied with the weight kernel of the node. Based on an analytic formulation, this approach not only required less computation compared to the competing techniques such as the visibility criterion, but also offered greater potential for computationally efficient extension to 3D by modifying the distance function to support the extra dimension. As depicted in Fig. 3, the distance function is computed in the local coordinate system of the cut piece. The 2D distance function for a

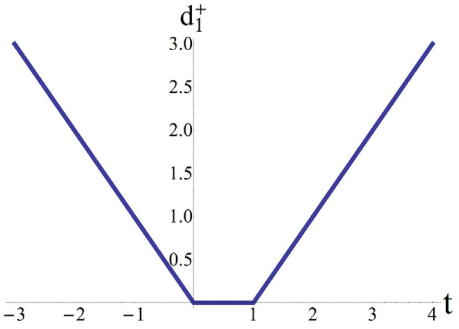

Fig. 4. The function $d_1^+(t)$ is calculated for a cut segment defined from $t_1 = 0$ to $t_2 = 1$.

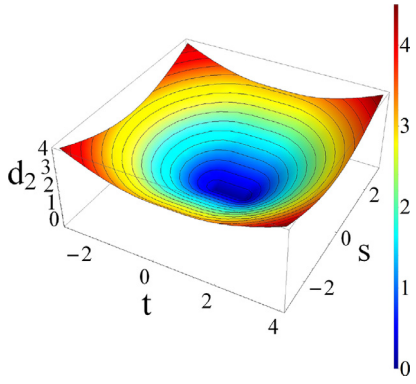

Fig. 5. Three dimensional plot of the distance function d_2.

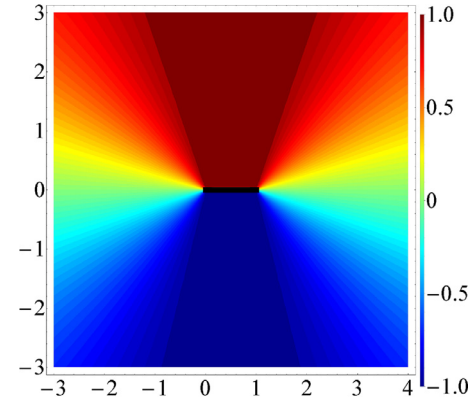

Fig. 6. Contour plot of the discontinuous φ function.

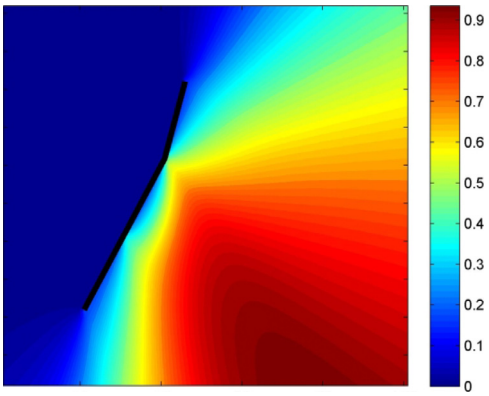

Fig. 7. When the enrichment values from individual segments are multiplicatively applied, the overall enrichment results in incorrect modification of meshless weight functions.

given point (x, y), can therefore be computed in terms of the local coordinates (t, s) as

$$d_2(x, y) = \sqrt{d_1^+(t)^2 + s^2}, \tag{13}$$

where $d_1^+(t)$ is defined as

$$d_1^+(t) = \frac{d_s(t) + |d_s(t)|}{2}. \tag{14}$$

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

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

where t_1 and t_2 are the endpoints of the cut segment in the cuts local coordinate system (Fig. 4).

In order to obtain an enrichment function that is sharply discontinuous across the cut segment, but smoothly varies from one side to the other side around the tips of the cut segment, we can take the partial derivative of the distance function d_2 (Fig. 5) with respect to the normal coordinate axis s,

$$\frac{\partial d_2}{\partial s} = \frac{s}{d_2} = \varphi, \tag{16}$$

and obtain the discontinuous function φ across the segment that is 1 on one side of the cut and -1 on the other side and varies smoothly around the cut (Fig. 6). The partial derivative of the distance function d_2 with respect to the coordinate that is vertical to the cut segment results in a constant 1 value above the cut segment and a constant -1 value below the cut segment, as d_2 has a constant variation at these regions along the vertical direction. Outside the regions immediately above and below the cut segment though, the function d_2 varies smoothly with respect to the vertical direction. The actual enrichment function is obtained by first translating the φ function and then scaling it down so that it ranges from 0 to 1,

$$h = \frac{\varphi + 1}{2}. \tag{17}$$

Designed for elastic isotropic deformable bodies modeled with meshless methods, the Barbieri technique handled multiple cut segments by simply multiplying their enrichment functions consecutively as

$$h = \Pi_i h_i, \tag{18}$$

where h_i is the enrichment computed for the cut segment i, and h is the final enrichment function for the meshless node that is affected by the cut segments. Although the multiplicative application is simple to implement, it can result in incorrect modification of meshless weight functions, therefore decreasing the stability of the simulation as depicted in Fig. 7.

6. Extended enrichment grid

As described above, the weight functions of the close meshless nodes are modified incorrectly when consecutive enrichments are applied through multiplication. In order to address these issues, an extension of the distance function-based enrichment technique is proposed to support consecutive discontinuity fronts in a correct way. In this extended technique, the enrichment values for multiple cuts are evaluated inside a grid structure, named the Enrichment Grid. For each grid point, the corresponding enrichment value is calculated in a manner 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

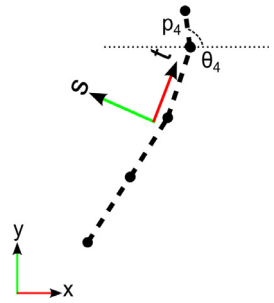

Fig. 8. The common coordinate system is calculated from four cut segments.

cut segment. The first step in the enrichment grid algorithm is to define a common coordinate system for calculating the regions of the grid points. In 2D, this common coordinate system is defined by the enrichment origin p_0 with coordinates (x_0, y_0) as well as the angle θ_0 between the horizontal axis of the common coordinate system and the positive x-axis of the world coordinate system as depicted in Fig. 8. The coordinate system is updated with each propagating cut as

$$\mathbf{p}_0 = \frac{\sum_{l=1}^{n} w_l \mathbf{p}_l}{\sum_{l=1}^{n} w_l}, \tag{19}$$

and

$$\theta_0 = \frac{\sum_{l=1}^{n} w_l \theta_l}{\sum_{l=1}^{n} w_l}, \tag{20}$$

where n is the number of cut segments, w_l is the associated weight with the cut segment l, which is typically the length of the segment in 2D problems, and $\mathbf{p}_l$ and θ_l are the center point and horizontal angle of the lth 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_l, y_l) are rigidly transformed into this new coordinate system to obtain new coordinates t and t_l by

$$t = cos(\theta)(x - x_0) + sin(\theta)(y - y_0) \tag{21}$$

and

$$t_l = cos(\theta)(x_l - x_0) + sin(\theta)(y_l - y_0). \tag{22}$$

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

$$d_1^+(t) = (t_0 - t) \cdot H(t_0 - t) + (t - t_n) \cdot H(t - t_n), \tag{23}$$

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 (Fig. 9).

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 of the cut segment l when $t' > t_{l-1} \& t' < t_l$. 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 (Fig. 10).

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 l, the s-coordinate (Fig. 11) is calculated as

$$s = -sin(\theta_l)(x - x_l) + cos(\theta_l)(y - y_l). \tag{24}$$

The extended enrichment function can be calculated with the (t, s) coordinates of the grid points. Using these values,

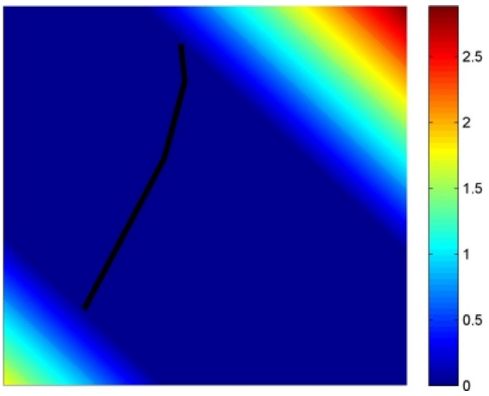

Fig. 9. Cut segments are processed as a series instead of individual processing. The series of cut segments are used to derive a common coordinate system. The function d_1^+ is calculated with respect to this common coordinate system.

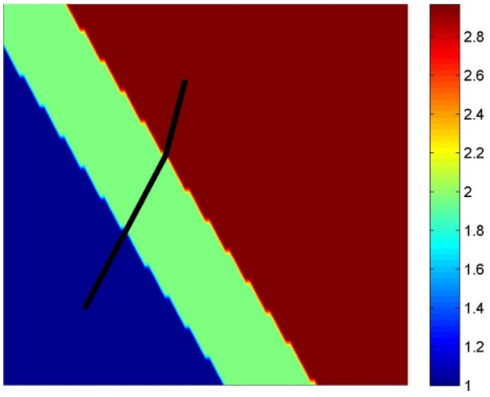

Fig. 10. The region values are assigned for individual cut segments with respect to the common coordinate system.

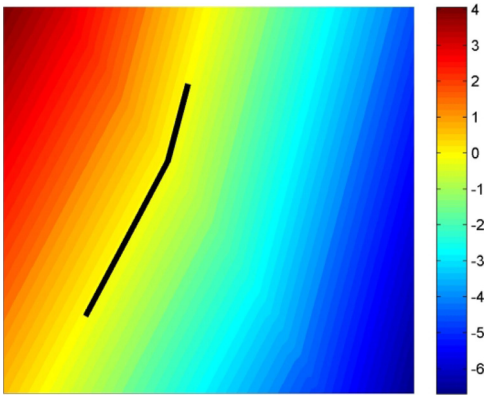

Fig. 11. For each grid location, the s-coordinates, which essentially represent the signed vertical distance of a point to the cut, are calculated according to the assigned region value.

$d_2(x, y)$ is calculated and its partial derivative with respect to the s-coordinates is taken to obtain the extended enrichment function (Fig. 12).

The extended enrichment technique with enrichment grids provides a computationally efficient way of handling multiple discontinuity fronts. Moreover, unlike the multiplicative approach, it modifies the weight function of the affected meshless nodes in a way such that stability problems are avoided.

Although, the enrichment technique is described in two dimensions, it can be extended into three dimensions through changes in the defined entities. For example, the piecewise-linear cut seg-

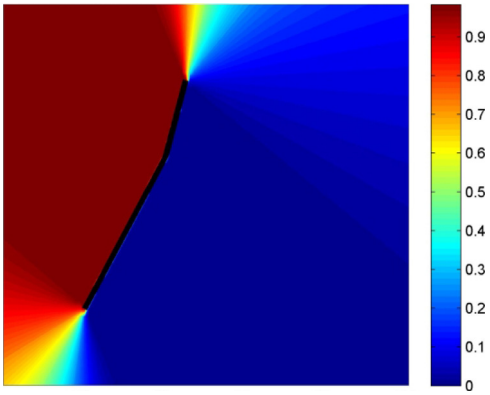

Fig. 12. The extended enrichment function for sequential cut segments. Unlike the original multiplicative approach, the enrichment function modifies the weight functions correctly to prevent instabilities.

1: **procedure** COMPUTE2DEXTENDEDENRICHMENT
2: **initialize** enrichment grid points (x,y)
3: **input** the first point of cut segment (x_{l0},y_{l0})
4: **while** the cut is not finished
5: **input** the next point of cut segment (x_{li},y_{li})
6: update the common coordinates defined by (p_0, Θ_0) in polar coordinate system or (t,s) in cartesian coordinate system
7: **foreach** grid point (x,y) and cut segment point (x_{li},y_{li})
8: transform (x,y) and (x_{li},y_{li}) into common coordinates (t,s)
9: calculate $d_1^+(t)$ values of the transformed grid points
10: **assign** cut segment regions for the grid points
11: calculate s values of the transformed grid points
12: calculate $d_2(x,y)$ values and the corresponding extended enrichment values of the transformed grid points

Fig. 13. The pseudocode for the extended enrichment function calculation in 2D.

ments are replaced with triangle strips and the 2D-array representation of the enrichment grid structure is replaced with a 3D-lattice structure with coordinate points (x, y, z). Likewise, the common coordinate system in 2D is replaced with the common plane in 3D. The distance function $d_3(x, y, z)$ is defined as

$$d_3(x,y,z) = \sqrt{d_2^{+2} + s^2}. \tag{25}$$

Here, d_2^+ is obtained by first projecting the grid point (x, y, z) and the associated cut triangle on to the common plane and then calculating the absolute distance of the projected point to the projected triangle. This absolute distance gets positive values outside of the projected triangle and 0 otherwise. The other term, s, is the distance of the grid point to the associated triangle along the triangle's normal direction.

After d_3 is computed, similar to the 2D case, the partial derivative of d_3 with respect to the normal direction s is taken to get the discontinuous function φ as Fig. 13

$$\frac{\partial d_3}{\partial s} = \frac{s}{d_3} = \varphi. \tag{26}$$

7. Results

Presented meshless method and the extended enrichment function were implemented as a plug-in component within the SOFA library using the library's object-oriented extension schema. The modular construct of the library allowed the division of the presented concepts into separate modules. For example, the meshless method was implemented as a *Behavior Model*, whereas the Enrichment Grid structure was suitably implemented as a *Container* module. In SOFA, the modules are consumed and linked together through hierarchical XML-based files called *Scene Files*. Scene Files can be used to setup links between modules so that information output from a module can be accessed from the linked one. A use case for such a requirement in our implementation is the information transfer from the module that controls the haptic device to our Enrichment Grid module.

In this section, the results for the point-based deformable object modeling approach and the cutting operation with the Extended Enrichment Grid are presented by modeling a sample object in the form of a synthetic rectangular block. The meshless nodes shown in Fig. 14 correspond to the Behavior Model of the simulation object (Fig. 15).

When a cutting operation is performed, an Enrichment Grid is generated for the cut, which is used to update the behavior model of the deformable object by modifying the weight functions of the affected meshless nodes (Fig. 16).

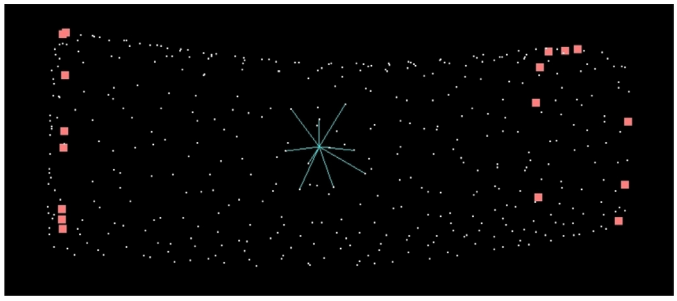

Fig. 14. The meshless node locations of the deformable block. This simulation object has 436 DoFs and simulated at around 125 FPS under the effect of gravitational forces.

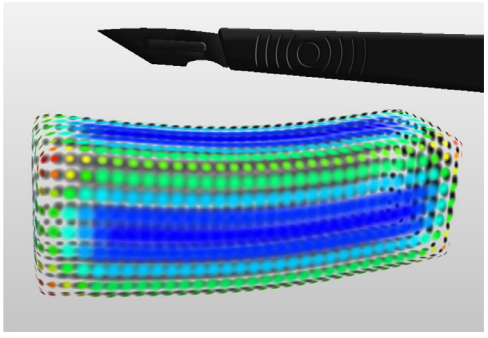

Fig. 15. The rectangular block object that is modeled with the meshless method and deformed with the gravity. This simulation object has 436 DoFs and simulated at around 105 FPS.

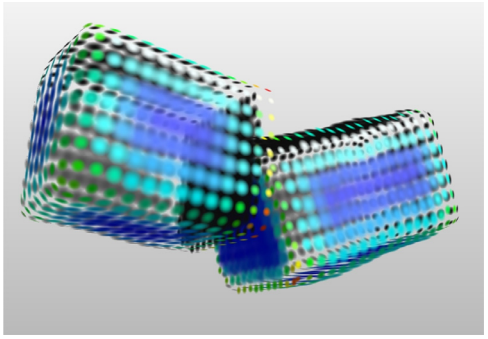

Fig. 16. The deformable block is cut, which results in the update of the behavior model.

8. Conclusion

In the surgical simulation context, an important functionality that the soft tissue models need to have is the cutting operation. For the mesh-based models however, the cutting operation is particularly problematic and usually becomes the bottleneck of the simulation in terms of performance. This paper discussed a novel way of handling piecewise cut segments in 2D, while also demonstrating its extension into 3D. The approach was achieved by refining the application of the mathematical enrichment function described earlier by Barbieri et al. [27]. The enrichment grid structure that is proposed in this work allowed the handling of consecutive cut segments in a correct way that prevented the occurrence of computational instabilities. Another advantage of the enrichment grid approach is that it is possible to utilize this structure as a spatial query accelerator in order to increase the performance of steps such as finding the meshless nodes that are affected by a cut.

References

[1] Ashley EA. Medical education-beyond tomorrow? Med Edu 2000;34:455–9.
[2] Loftin B. Med school 1.0: can computer simulation aid physician training. Quest 2002;5:16–19.
[3] Grush L. Surgical theater: flight simulation technology helps surgeons prep for surgery. 2014. Available: http://www.foxnews.com/health/2014/04/02/surgical-theater-flight-simulation-technology-helps-surgeons-prep-for-surgery/.
[4] Ota D, Loftin B, Saito T, Lea R, Keller J. Virtual reality in surgical education. Comput Biol Med 1995;25:127–37.
[5] Chen DT, Kakadiaris IA, Miller MJ, Loftin RB, Patrick C. Modeling for plastic and reconstructive breast surgery. In: Medical image computing and computer-assisted intervention MICCAI 2000; 2000. p. 1040–50.
[6] Allard J, Cotin S, Faure F, Bensoussan PJ, Poyer F, Duriez C. SOFA - an open source framework for medical simulation. Medicine meets virtual reality; 2007.
[7] Nedel LP, Thalmann D. Real time muscle deformations using mass-spring systems. In: Computer graphics international, proceedings, 1998; 1998. p. 156–65.
[8] Bathe K-J. Finite element procedures, 2. Prentice Hall Englewood Cliffs; 1996.
[9] Eymard R, Gallout T, Herbin R. Finite volume methods. In: Handbook of numerical analysis, 7. Elsevier; 2000. p. 713–1018.
[10] Mitchell AR, Griffiths DF. The finite difference method in partial differential equations, 1980. Chichester, Sussex, England and New York: Wiley-Interscience; 1980. p. 281.
[11] Bro-Nielsen M. Finite element modeling in surgery simulation. Proc IEEE 1998;86:490–503.
[12] Miller K, Joldes G, Lance D, Wittek A. Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. Commun Numer Methods Eng 2007;23:121–34.
[13] Marchesseau S, Heimann T, Chatelin S, Willinger R, Delingette H. Multiplicative jacobian energy decomposition method for fast porous visco-hyperelastic soft tissue model. In: Medical image computing and computer-assisted intervention MICCAI 2010. Springer; 2010. p. 235–42.
[14] Courtecuisse H, Jung H, Allard J, Duriez C, Lee DY, Cotin S. GPU-based real-time soft tissue deformation with cutting and haptic feedback. Prog Biophys Mol Biol 2010;103:159–68.
[15] Wu J, Westermann R, Dick C. Real-time haptic cutting of high-resolution soft tissues. Stud Health Technol Inform 2014;196:469–75.
[16] Chen Y, Lee JD, Eskandarian A. Meshless methods in solid mechanics. Berlin: Springer Verlag; 2006.
[17] Liu GR. Meshfree methods: moving beyond the finite element method. CRC Press; 2010.
[18] Nguyen VP, Rabczuk T, Bordas S, Duflot M. Meshless methods: a review and computer implementation aspects. Math Comput Simul 2008;79:763–813.
[19] Li H, Mulay SS. Meshless methods and their numerical properties. CRC Press; 2013.
[20] Nayroles B, Touzot G, Villon P. Generalizing the finite element method: Diffuse approximation and diffuse elements. Comput Mech 1992;10:307–18.
[21] Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. Int J Numer Methods Eng 1994;37:229–56.
[22] Mller M, Keiser R, Nealen A, Pauly M, Gross M, Alexa M. Point based animation of elastic, plastic and melting objects. In: ACM SIGGRAPH/eurographics symposium on computer animation; 2004. p. 141–51.
[23] Zhang X, Liu XH, Song KZ, Lu MW. Least-squares collocation meshless method. Int J Numer Methods Eng 2001;51:1089–100.
[24] Organ D, Fleming M, Terry T, Belytschko T. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. Comput Mech 1996;18:225–35.
[25] Mos N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. Int J Numer Methods Eng 1999;46:131–50.
[26] Kaufmann P, Martin S, Botsch M, Grinspun E, Gross M. Enrichment textures for detailed cutting of shells. ACM Trans Graphics 2009;28:50.
[27] Barbieri E, Petrinic N, Meo M, Tagarielli VL. A new weight-function enrichment in meshless methods for multiple cracks in linear elasticity. Int J Numer Methods Eng 2011:177–95.
[28] Banihani S, Rabczuk T, Almomani T. POD for real-time simulation of hyperelastic soft biological tissue using the point collocation method of finite spheres. Math Prob Eng 2013;2013:9. Article ID 386501.
[29] Hertz H. Über die Berührung fester elastischer körper. On the contact of elastic solids. Journal fr die Reine und Angewandte Mathematik 1882;1882:156–71.
[30] Franke D, Dster A, Rank E. The pversion of the FEM for computational contact mechanics. Proc Appl Math Mech 2008;8:10271–2.
[31] Schwarzer N, Djabella H, Richter F, Arnell R. Comparison between analytical and FEM calculations for the contact problem of spherical indenters on layered materials. Thin Solid Films 1995;270:279–82.
[32] Franke D, Dster A, Nbel V, Rank E. A comparison of the h-, p-, hp-, and rp-version of the FEM for the solution of the 2d hertzian contact problem. Comput Mech 2010;45:513–22.
[33] Pennec F, Achkar H, Peyrou D, Plana R, Pons P, Courtade F. Verification of contact modeling with COMSOL multiphysics software. Sixth EUROSIM congress on modelling and simulation, Ljubljana, Slovenia; 2007.
[34] Horton A, Wittek A, Joldes GR, Miller K. A meshless total lagrangian explicit dynamics algorithm for surgical simulation. Int J Numer Methods Biomed Eng 2010;26:977–98.
[35] Duflot M, Nguyen-Dang H. A meshless method with enriched weight functions for fatigue crack growth. Int J Numer Methods Eng 2004;59(14):1945–61.

Dr. Rifat Aras received B.S. and M.Sc. degrees in Computer Science Department from Bilkent University, Ankara, Turkey in 2005 and 2008 respectively. He received his Ph.D. degree in Modeling, Simulation, and Visualization Department from Old Dominion University, Norfolk, VA in 2014. His research interests include computer graphics, physically-based modeling and simulation, and virtual reality.

Dr. Yuzhong Shen received his B.S. degree in Electrical Engineering from Fudan University, Shanghai, China in 1990, M.S. degree in Computer Engineering from Mississippi State University, Starkville, Mississippi in 2000, and Ph.D. degree in Electrical Engineering from the University of Delaware, Newark, Delaware in 2004. His research interests include visualization and computer graphics, signal and image processing, and modeling and simulation. Dr. Shen is currently an Associate Professor of the Department of Modeling, Simulation, and Visualization Engineering at Old Dominion University, with joint appointment with the Department of Electrical and Computer Engineering. Prior to joining Old Dominion University, Dr. Shen worked as an Engineer and a Senior Engineer with Weifang Hua-Guang Technologies, China (1990–1998), a Research Assistant with National Science Foundation Engineering Research Center for Computational Field Simulation at Mississippi State University (1998–2000), a Research Assistant with the Department of Electrical and Computer Engineering at the University of Delaware (2000–2004), and a Senior Research Scientist with Virginia Modeling, Analysis, and Simulation Center at Old Dominion University (2004–2006).

Dr. Michel Albert Audette received the B.Eng. (Electrical) degree from McGill University, in 1986, the M.Eng. degree (Electrical) from Ecole Polytechnique in 1993, and the Ph.D. (Biomedical Engineering) from McGill in 2002, all in Montreal, Canada. His industry experience includes flight simulation from 1986 to 1988, welding automation from 1991 to 1994, neurosurgical navigation (part-time) from 1995 to 1997, as well as open-source image analysis software from 2008 to 2011. He also did postdoctoral research at the National Institute of Advanced Industrial Science and Technology (AIST) in Tsukuba, Japan from 2001 to 2005 and at Innovation Center Computer Assisted Surgery (ICCAS) in Leipzig, Germany from 2006 to 2008. He has patents in US and Japan on surgery planning. Since July 2011, he has been employed as assistant professor in Old Dominion University's Department of Modeling, Simulation and Visualization Engineering. His research interests include medical simulation, medical image analysis, therapy planning, all three with an emphasis on neuro- and orthopedic surgery, as well as other clinical applications of musculoskeletal modeling, in addition to imaging and simulation applications for military well-being.