

Meshless algorithm for simulation of soft tissue cutting for surgical simulation

Xia Jin^a, Grand Roman Joldes^{a*}, Karol Miller^a, King H. Yang^b, Adam Wittek^a

^aIntelligent Systems for Medicine Laboratory, School of Mechanical and Chemical Engineering, the University of Western Australia, 35 Stirling Highway, Crawley, Perth, WA 6009, Australia

^bBioengineering Center, Wayne State University, 818 W. Hancock, Detroit, MI 48201, USA

*Corresponding Author: E-mail: grand.joldes@uwa.edu.au

Meshless algorithm for simulation of soft tissue cutting for surgical simulation

Abstract: Computation of soft tissue mechanical responses for surgery simulation and image-guided surgery has been dominated by finite element method that utilises a mesh of interconnected elements as a computational grid. Shortcomings of such mesh-based discretisation in modelling of surgical cutting include high computational cost and the need for re-meshing in the vicinity of cutting-induced discontinuity. The Meshless Total Lagrangian Adaptive Dynamic Relaxation (MTLADR) algorithm we present here does not exhibit such shortcomings as it relies on spatial discretisation in a form of a cloud of nodes. The cutting-induced discontinuity is modelled solely through changes in nodal domains of influence, which is done through efficient implementation of the visibility criterion using the level set method. Accuracy of our MTLADR algorithm with visibility criterion is confirmed against the established non-linear solution procedures available in the commercial finite element code Abaqus.

Keywords: surgery simulation; soft tissue cutting; level set; meshless method; dynamic relaxation

1. Introduction

Application of computational biomechanics in medicine includes computing soft tissue mechanical responses. Such computations have been dominated by the finite element method that makes it possible to accurately represent complex geometries of the body organs and account for non-linear constitutive properties of soft tissues and large deformations/strains induced by surgery. Recent examples include implementations of fully non-linear finite element procedures for fast computations of organ deformation for image-guided surgery on graphics processing units (GPUs) (Allard et al., 2012; Joldes et al., 2010a; Taylor et al., 2008) and software frameworks for surgical simulation (Allard et al., 2007; Allard et al., 2012; Joldes et al., 2009b).

Despite advancements in computational biomechanics, modelling and simulation of soft tissue cutting still remain one of the most challenging problems in surgery simulation. The difficulties are how to model surgical cutting and the non-linear geometric and material behaviour exhibited by soft organs (Fung, 1993; Miller et al., 2000; Wittek et al., 2008), and achieve high computation speeds. To increase the computation speed, some researchers modelled surgical cutting on surface meshes and predicted the deformation of soft tissue using mass-spring models (Choi, 2006; Meseure and Chaillou, 1997; Pan et al., 2011; Yuan et al., 2010; Zhang, 2004) or boundary element model (Wang et al., 2006, 2007). Although such simplified approaches made it possible to achieve rapid computation speeds, they suffer from important shortcomings. A surface mesh is only suitable for modelling of cutting on membrane-like structures, such as intestine or gallbladder. Although layered surfaces can be employed, a surface mesh is unable to simulate progressive cutting in depth. Furthermore, while a mass-spring model features simple intrinsic mechanisms and high computational efficiency, it suffers from poor

precision and stability problems because damping for stable mass-spring systems is difficult to determine when topology changes due to cutting. The boundary element solutions have been so far developed for isotropic elastic materials in small deformation regime (Wang et al., 2006, 2007) as the requirement of Green's function limits its ability to deal with non-linear problems.

Progressive surgical cutting has been modelled and simulated by subdivision of elements of the volumetric mesh using conventional finite element (FE) method (Bielser et al., 2004; Bruyns et al., 2002; Courtecuisse et al., 2010; Mor, 2001). Such approach requires sophisticated re-meshing technologies to generate new elements with good aspect ratio and map the field variables from the old mesh to the new mesh. Despite the exploration of speed-up technologies (e.g. implementation on GPUs, sub-structuring models into operational regions where re-meshing is done and non-operational ones that do not require re-meshing), high computational cost and error accumulation due to re-meshing constrain the computational efficiency and accuracy of finite element method in modelling of cutting (Cotin et al., 2000; Courtecuisse et al., 2010; Wu and Heng, 2005).

As one possible solution to avoid re-meshing, (Vigneron et al., 2004) showed the potential of the extended finite element method (XFEM) for simulation of surgical cutting in 2D using linear elasticity assumption. Although XFEM has been proposed in the literature as a general method for modelling discontinuities due to cracks/cuts (Bordas and Moran, 2006; Stolarska et al., 2001), the fact that it requires construction of patient-specific finite element meshes is an important obstacle in surgery simulation. Despite recent progress, that includes octree-based hexahedral mesh generation and mesh refinement (Ebeida et al., 2011; Marechal, 2009; Schneiders, 2000; Zhang and Bajaj, 2006), it is still difficult to mesh 3D highly irregular geometries (e.g. brain) automatically with good quality hexahedral elements that are regarded as

the most effective when dealing with the incompressibility of soft tissues (Doblaré et al., 2005; Ji et al., 2011; Miller et al., 2011). Linear elastic models are used in some simulations to reduce the computational burden (Basdogan et al., 1999; Cotin et al., 2000; Lim et al., 2006; Wang et al., 2006, 2007). However, this simplification deviates from the intrinsic non-linear properties of soft tissue.

There is extensive literature on determining and controlling the discretisation errors of methods of computational mechanics that are applicable in surgery simulation (for comprehensive summaries see (Bathe, 1996) and (Liu, 2003)). However, all the existing techniques used in surgery simulation are still at the exploratory stage as no research group has provided an effective verification or validation of the simulation results so far.

In this paper, we propose a Meshless Total Lagrangian Adaptive Dynamic Relaxation (MTLADR) algorithm to predict the steady-state deformation of soft tissue caused by surgical cutting. Our algorithm belongs to the family of Element-Free Galerkin (EFG) methods which are well suited for solving problems with moving boundaries, such as the modelling of crack propagation (Belytschko et al., 1994; Belytschko et al., 2000; Belytschko and Tabbara, 1996; Rabczuk and Belytschko, 2007). The algorithm features a meshless spatial approximation based solely on nodes. The progressive surgical cutting is modelled by adding and/or splitting nodes on the cutting path and implementing visibility criterion (Belytschko et al., 1996) with the aid of the level set method (Osher and Sethian, 1988). Since building the connectivity between nodes and integration points does not rely on elements, the burden associated with the mesh generation and re-meshing required by the FE method is partly alleviated. With the capability of modelling large deformation and soft tissue non-linear behaviour, MTLADR algorithm has fast convergence to the steady state solution and is generally applicable for both 2D and 3D problems. In this paper,

we explain the surgical cutting modelling method and formulations of solution algorithm (MTLADR) in 2D. The simulation results are verified against well-established finite element procedures in the commercial finite element solver Abaqus (ABAQUS, 2009).

The paper is organised as follows. Section 2 presents the method of simulating surgical cutting. The experiment of simulation and the verification of results are given in Section 3. The conclusions and discussion of future work are in Section 4.

2. Methods

We developed the meshless Total Lagrangian adaptive dynamic relaxation (MTLADR) algorithm to simulate the deformation of soft tissue during surgery. The topology changes introduced by surgical cutting are modelled by adding and/or splitting nodes along the cutting path with the help of the level set method (Osher and Sethian, 1988).

Total Lagrangian formulation (Bathe, 1996; Horton et al., 2010; Miller et al., 2007), in which the calculated variables (such as stresses and strains) are referred to the initial (undeformed) configuration of the analysed continuum, is used as it exhibits the following features:

- The error accumulation due to the stress/strain update associated with the Updated Lagrangian formulation typically used in commercially available finite element solvers (such as e.g. Abaqus and LS-DYNA) is eliminated;
- The shape functions and their spatial derivatives can be pre-computed. This reduces the number of numerical operations (and the computation time) when compared to the Updated Lagrangian formulation (Miller et al., 2007).

In the literature, implicit integration in time domain is often recommended (Bathe, 1996) and used for quasi-static/steady-state problems including surgery simulation (Allard et al., 2007;

Allard et al., 2012; Taylor et al., 2008). Our rationale for application of dynamic relaxation is that, as indicated by our previous research on finite element algorithms for computation of the soft tissue responses (Joldes et al., 2011), it offers an excellent performance in terms of computation speed while preserving accuracy as good as the implicit integration. The solution accuracy is achieved by controlling the error through stringent convergence criteria. Following (Joldes et al., 2011), we used the displacement infinity norm for such criterion. Determining the stable time step for meshless methods with explicit integration in time domain has been addressed in our recent study (Joldes et al., 2012).

2.1. *Governing equations and solution*

In the MTLADR algorithm, the steady-state deformation of soft tissue at any stage of cutting is solved as the steady state part of the general equation of motion, in which an artificial mass proportional damping is included to increase the convergence speed (Underwood, 1983):

$$\mathbf{M}\ddot{\mathbf{U}} + c\mathbf{M}\dot{\mathbf{U}} + \mathbf{F}(\mathbf{U}) = \mathbf{R} \quad (1)$$

where \mathbf{M} is the time-dependent lumped mass matrix; \mathbf{F} is the vector of internal nodal forces; \mathbf{R} is the vector of externally applied nodal loads; \mathbf{U} is the vector of nodal displacements; c is the damping coefficient.

A dynamic relaxation algorithm for solving non-linear finite element problems without cutting has been presented in (Joldes et al., 2009a, 2011). We adapt their strategy of estimating the damping coefficient to our meshless scheme. The nodal displacements at every time step of explicit integration are derived as

$${}^{t+\Delta t}\mathbf{U} = a {}^t\mathbf{M}^{-1}({}^t\mathbf{R} - {}^t\mathbf{F}) + b {}^t\mathbf{U} + (1 - b) {}^{t-\Delta t}\mathbf{U} \quad (2)$$

$$a = \frac{(\rho + 1)^2 \cdot \Delta t^2}{4} \quad (3)$$

$$b = \rho^2 + 1, \quad (4)$$

where ${}^{t+\Delta t}\mathbf{U}$, ${}^t\mathbf{U}$ and ${}^{t-\Delta t}\mathbf{U}$ are the nodal displacement vectors (as field variables) at times $t + \Delta t$, t and $t - \Delta t$ respectively, Δt is the time step, ${}^t\mathbf{M}$ indicates the global time-dependent lumped mass matrix, and ρ is the spectral radius which is adaptively calculated during the iterations:

$$\rho = 1 - \frac{2}{\sqrt{k} + 1} \quad (5)$$

$$k = \frac{4}{\Delta t^2 A_0} \quad (6)$$

$$A_0 \approx \frac{({}^t\mathbf{U} - {}^l\mathbf{U})^T ({}^t\mathbf{F} - {}^l\mathbf{F})}{({}^t\mathbf{U} - {}^l\mathbf{U})^T {}^t\mathbf{M} ({}^t\mathbf{U} - {}^l\mathbf{U})} \quad (7)$$

where ${}^l\mathbf{U}$ is the vector of nodal displacements at a previous time point l close to t ; ${}^t\mathbf{F}$ is the global nodal reaction force vector at time t relative to initial configuration; ${}^l\mathbf{F}$ is the global nodal reaction force vector at the previous time point l relative to initial configuration. More details on the derivation of Equation (7) are presented in (Joldes et al., 2011).

Introducing discontinuities by cutting can lead to spurious jumps and oscillations in the solution. Dynamic relaxation is very effective in damping out these oscillations without the need for any additional treatment.

2.2. *Modelling of surgical cutting in 2D*

Methods for modelling of discontinuity in the displacement and stress fields associated with cuts/cracks are still a hotly debated topic. An extensive review of such methods from the

perspective of the overreaching concept of partition of unity has been recently conducted by (Rabczuk et al., 2010).

In this study, we geometrically represent the progressing surgical cut as a series of line segments (as shown in Figure 1) and adapt the level set method proposed by (Osher and Sethian, 1988) and developed by (Stolarska et al., 2001) to mathematically describe the locations of all nodes and integration points relative to the cutting path. The discontinuities induced by surgical cutting are modelled and traced using nodes with specific level set values and appropriate field values (the nodal displacement and the size of domain of influence). The effect of cutting is entirely reflected in the changes of the shape and size of the influence domain of nodes by efficiently implementing visibility criterion with the help of the level set method.

[Possible location of Figure 1]

2.2.1. *Initialisation of level set values of nodes and integration points*

When the first cut (e.g. T0-T1 in (Figure 1)) is made, we mathematically represent the cutting direction as the zero level set of function $\psi(x, y)$ in Equation (8). The endpoint of a cut is represented as the intersection of the zero level set of function ψ with an orthogonal zero level set of function $\phi(x, y)$ defined in Equation (9):

$$\psi(x, y) = (x - x_{ep}) \frac{V_y}{\|V\|} - (y - y_{ep}) \frac{V_x}{\|V\|} \quad (8)$$

$$\phi(x, y) = (x - x_{ep}) \frac{V_x}{\|V\|} + (y - y_{ep}) \frac{V_y}{\|V\|} \quad (9)$$

where (x, y) is the coordinate of a given point in the problem domain; (x_{ep}, y_{ep}) is the coordinate of the endpoint of the cutting path. V_x, V_y are the components of vector V

representing the cutting direction, and $\|\mathbf{V}\|$ is the length of vector \mathbf{V} .

The values of both level set functions ψ and ϕ are calculated and stored for all the nodes and integration points to indicate their positions relative to the cutting path. The value of the level set function ψ at a given point is the signed-distance from this point to the cutting direction. As illustrated in (Figure 2), the points with positive values of the level set function ψ are all located at the same side of the cutting line segment L and its extension $L1$ while the points with negative values are all at the other side. Zero value of function ψ indicates that the point is located right on the cutting line segment L or its extension line $L1$. Accordingly, the value of the level set function ϕ indicates the signed-distance from a given point to line segment $L2$ which is orthogonal to the cutting path at the endpoint. If a point is judged to be located on the cutting line segment L (zero value of function ψ and negative value of function ϕ), the associated value of function ψ is set to a small positive or negative value in order to allocate the point to one of the sub-domains $S1$ or $S2$ (Figure 2).

[Possible location of Figure 2]

2.2.2. *Spatial discretisation of the cutting path*

The endpoint of the cut and the rest of the line segments representing the cutting path are discretised using nodes. If the computational grid obtained by discretising the geometry of the analysed continuum contains no node at the endpoint of the cut, we add a node there with zero values of level set functions ψ and ϕ (Figure 3a). The rest of the cutting segments are discretised using nodes spaced at the average nodal interval of the computational grid. If there is no node at the discretised position, two nodes (having the same coordinates) are added at this position. If there is an existing node at the discretised position or elsewhere on the cutting path, we split this

node into two nodes having the same coordinates.

Every two nodes added (or split) at the same position are allocated on opposite sides of the cutting path by appropriate values of the level set functions (Figure 3b,c). For one node, the level set function ψ is set to a positive value smaller than the minimum size of the influence domain, while for the other one, the function ψ is set to a negative value of the same magnitude. Both nodes are assigned with the same negative value of the level set function ϕ with magnitude less than the length of the cutting segment. Imposing the limit of the minimum size of the influence domain on the magnitudes of the assigned level set values ensures that the nodes fall into the confined area in which the nodal influence domains should be re-shaped to account for discontinuities introduced by cutting.

[Possible location of Figure 3]

Besides the level set values, appropriate field values are assigned to the newly added nodes. If the node is created at a new position, its displacements are interpolated (from the adjacent nodes) by Moving Least Squares (MLS) approximation and the size of its domain of influence is set to the average size of the influence domains of the surrounding nodes. If the node is created by splitting of an existing node, its displacements and the size of influence domain are inherited from the existing node.

2.2.3. *Update of level set values of nodes and integration points while cutting progresses*

When cutting progresses from one point to the next one, the level set values of functions ψ and ϕ at the nodes and integration points need to be updated. In the following algorithm, ψ^n and ϕ^n denote the values of functions ψ and ϕ at step n ; ψ^{n+1} and ϕ^{n+1} denote the values of functions ψ and ϕ at step $n + 1$; (x, y) is the coordinate of a given node or integration point in

the problem domain; (x_{ep}^n, y_{ep}^n) and $(x_{ep}^{n+1}, y_{ep}^{n+1})$ are the coordinates of the cut endpoint at steps n and $n + 1$ respectively; V_x^{n+1}, V_y^{n+1} are the components of the cutting vector \mathbf{V}^{n+1} at step $n + 1$; vector \mathbf{V}^{n+1} is defined by points (x_{ep}^n, y_{ep}^n) and $(x_{ep}^{n+1}, y_{ep}^{n+1})$.

(1) *Cutting direction does not change.* When cutting progresses from one endpoint to the next without direction change, the values of level set function ϕ of the nodes and integration points whose $\phi^n \geq 0$ need to be updated. The update process is illustrated in (Figure 4). No update is done in the area where $\phi^n < 0$ (indicated as non-update area $A^{non-update}$ in (Figure 4)) while the rest of the analysed domain (where $\phi^n \geq 0$) is an update area A^{update} :

$$\psi^{n+1} = \psi^n \quad \text{in } A = A^{non-update} \cup A^{update} \quad (10)$$

$$\phi^{n+1} = \phi^n \quad \text{in } A^{non-update} \quad (11)$$

$$\phi^{n+1}(x, y) = (x - x_{ep}^{n+1}) \frac{V_x^{n+1}}{\|\mathbf{V}^{n+1}\|} + (y - y_{ep}^{n+1}) \frac{V_y^{n+1}}{\|\mathbf{V}^{n+1}\|} \quad \text{in } A^{update} \quad (12)$$

where A is the union of $A^{non-update}$ and A^{update} areas.

[Possible location of Figure 4]

(2) *Cutting direction changes.* If the cutting direction changes, the update region A_c^{update} is defined as the area where $\overline{\phi^n} > 0$ while the rest of the analysed domain is defined as the non-update region $A_c^{non-update}$ (Figure 5).

a) Level set function $\overline{\phi^n}$ is obtained by rotating function ϕ^n around the cutting path endpoint (x_{ep}^n, y_{ep}^n) (point T2 in (Figure 5)) until orthogonal to the cutting vector \mathbf{V}^{n+1} :

$$\overline{\phi^n}(x, y) = (x - x_{ep}^n) \frac{V_x^{n+1}}{\|\mathbf{V}^{n+1}\|} + (y - y_{ep}^n) \frac{V_y^{n+1}}{\|\mathbf{V}^{n+1}\|} \quad \text{in } A \quad (13)$$

where A is defined in Equation (10).

b) Following changes in the cutting direction, the values of function ψ^{n+1} of the nodes and integration points in the update region A_c^{update} are re-computed while no re-computation is needed in the non-update region $A_c^{non-update}$:

$$\psi^{n+1}(x, y) = (x - x_{ep}^{n+1}) \frac{V_y^{n+1}}{\|V^{n+1}\|} - (y - y_{ep}^{n+1}) \frac{V_x^{n+1}}{\|V^{n+1}\|} \quad \text{in } A_c^{update} \quad (14)$$

$$\psi^{n+1} = \psi^n \quad \text{in } A_c^{non-update} \quad (15)$$

c) Re-computation of function ϕ^{n+1} is done in the entire analysed area A :

$$\phi^{n+1}(x, y) = (x - x_{ep}^{n+1}) \frac{V_x^{n+1}}{\|V^{n+1}\|} + (y - y_{ep}^{n+1}) \frac{V_y^{n+1}}{\|V^{n+1}\|} \quad \text{in } A . \quad (16)$$

[Possible location of Figure 5]

2.2.4. Update of the influence domain of the nodes affected by cutting

The update of the influence domains of any node in the vicinity of the cutting path is implemented by finding and eliminating the points that are no longer influenced by the node due to the cutting path. In the following algorithm, ψ_i and ϕ_i denote the values of the level set functions ψ and ϕ at node i while ψ_p and ϕ_p denote the values of level set functions ψ and ϕ at a point P in the influence domain of node i .

(1) The influence domain of a node, which is defined as a circle in our algorithm, is updated if it intersects the cutting path (Figure 6). The selection criterion for the node whose influence domain needs to be updated is

$$|\psi_i| \leq R_i \text{ and } \phi_i < R_i \quad (17)$$

where R_i is the size (radius) of the influence domain of node i .

[Possible location of Figure 6]

(2) The point is kept in the influence domain of the node if the node and the point are both located at the same side of the cutting path:

$$\psi_i \times \psi_p \geq 0 \quad (18)$$

(3) If the point and the node are located at the opposite sides of the cutting path,

$$\psi_i \times \psi_p < 0 \quad (19)$$

additional criteria need to be checked to decide whether the point is to be kept or eliminated from the influence domain of the node:

(a) As illustrated in (Figure 7), the first selection criterion for the points that need to be eliminated from the influence domain of the node is: the node and the point are located in different subdomains S1 and S2, i.e. one of them is located in sub-domain S1 while another one is within sub-domain S2 or on line segment L2. The above selection criterion is expressed as (see also Figure 7):

$$(\phi_i \leq 0 \text{ AND } \phi_p < 0) \text{ OR } (\phi_i < 0 \text{ AND } \phi_p = 0) \quad (20)$$

(b) If the criterion (a) expressed by Equation (20) is not satisfied, there are only two alternatives for position of the point in relation to the node (Figure 7): either one is located in sub-domain S1 and another one is in sub-domain S4 or one is in sub-domain S2 and another one is in sub-domain S3. Therefore, we have to geometrically judge if the line segment linking the node and the point intersects the cutting path. If the intersection occurs, the point is removed from the influence domain of the node.

(4) The following updates are made to account for the changes in influence domains:

- For the points that are eliminated from the influence domains of the nodes, the shape functions and their derivatives are re-calculated.

- The global mass matrix is updated.

[Possible location of Figure 7]

The update procedure proposed in this section works well for direction changes of less than 90 degrees. If more abrupt changes in cutting direction have to be simulated, the procedure needs to be modified or a different approach for defining the discontinuity may be needed. The drawback, however, appears to be insignificant as surgical techniques employing cuts with abruptly changing direction are not reported in commonly used surgery textbooks (Greenfield and Mulholland, 2010; Westra et al., 2002).

2.3. *Description of MTLADR solution algorithm for surgical cutting simulation*

The complete Meshless Total Lagrangian Adaptive Dynamic Relaxation (MTLADR) algorithm adapted for surgical cutting simulation can be described as follows:

(a) Pre-computation and initialisation:

- Compute the shape functions and their derivatives for each integration point. A regular grid of square integration cells is used for Gaussian quadrature;
- Specify the time step for explicit integration in time domain and apply mass scaling to ensure the integration stability;
- Create global lumped mass matrix by evenly distributing the scaled mass carried by each integration point to the nodes that influence it;
- When simulating cutting of the deforming continuum, subject the continuum to loading through essential boundary conditions (i.e. by prescribing the displacements on the boundary) and compute the deformations within continuum.

For each progressive cut:

(b) Cutting:

- Update the shape functions and their derivatives at the integration points affected by cutting (section 2.2.4);
- Determine the critical time step and update the mass scaling coefficients at the integration points;
- Update the scaled global lumped mass matrix.

(c) Relaxation:

- Initialise the spectral radius for dynamic relaxation with a value close to 1 (e.g. 0.998), which attenuates the high-frequency oscillations in the solution but allows deformation propagation (small damping) (Joldes et al., 2009a);
- For each time step of explicit integration:
 - Update ${}^{t-\Delta t}\mathbf{U} = {}^t\mathbf{U}$ and ${}^t\mathbf{U} = {}^{t+\Delta t}\mathbf{U}$;
 - Compute the global nodal reaction forces as described in (Horton et al., 2010);
 - Compute the nodal displacements at next time step (Equations 2 - 4);
 - Impose the essential boundary conditions;
 - Re-calculate the spectral radius (Equations 5 - 7). The spectral radius used in computations is smoothed in order to eliminate large jumps in its value.
 - Update ${}^l\mathbf{U}$ and ${}^l\mathbf{F}$ using the computed nodal displacements and global nodal reaction forces.
 - Following the update of ${}^l\mathbf{U}$ and ${}^l\mathbf{F}$, the spectral radius is regarded as stabilised if the difference between the calculated spectral radius ρ and the currently used one is less than a specified threshold.

- If the convergence rate becomes stabilised, check the convergence criteria (Joldes et al., 2011). The time iteration is terminated if the convergence criterion is met.

The computed nodal displacements at the end of Dynamic Relaxation in a Meshless method are in fact field variables. To compute the real displacements at any point (including the nodes), an interpolation needs to be performed using the meshless shape functions.

3. Numerical results and algorithm verification

3.1. *Modelling and simulation of soft tissue cutting in 2D using MTLADR algorithm*

We simulate cutting in a deforming continuum having soft tissue-like material behaviour. The deformation preceding cutting is induced by elongating a square specimen of size of 0.1m by 0.1m as illustrated in (Figure 8). One edge ($x=0$ m) of the specimen is rigidly constrained while the opposite one ($x=0.1$ m) is elongated by 0.02 m along the x-axis. The specimen is discretised into nodes. The number of nodes (6561) was selected based on the results of convergence analysis summarised in (Figure 9) and Table 1. Because of the specimen's rectangular shape, we distributed the nodes regularly. However, as indicated by (Horton et al. 2010), the Total Lagrangian meshless algorithm with MLS shape functions can be also applied for irregular node placements.

Various non-linear material models can be used in the MTLADR algorithm through application of different formulae for calculation of the second Piola-Kirchhoff stress at the integration points. In our simulations we used the Neo-Hookean material model and modelled soft tissue as a soft and nearly incompressible continuum (Young's modulus of $E = 3000$ Pa, Poisson's ratio of $\nu = 0.49$, mass density of $\rho = 1000\text{kg/m}^3$). As we verify the MTLADR algorithm against the established non-linear static solution procedures available in the commercial finite element code Abaqus (ABAQUS, 2009), we use the same form of the Neo-

Hookean strain energy potential as the one in Abaqus:

$$U = \frac{\mu_0}{2}(\bar{I}_1 - 3) + \frac{k_0}{2}(J - 1)^2 \quad (21)$$

$$\bar{I}_1 = J^{-1} \cdot I_1 \quad \text{for 2D} \quad (22)$$

where μ_0 is the initial shear modulus; k_0 is the initial bulk modulus; \bar{I}_1 is the first deviatoric strain invariant; J is the volumetric change which equals to the determinant of the deformation gradient; I_1 is the first invariant of the right Cauchy deformation tensor. From Equations (21) and (22), we derive the second Piola-Kirchhoff stress tensor \mathbf{S} for 2D problems as

$${}^t_0\mathbf{S}_{ij} = \mu_0 J^{-1} \left(\delta_{ij} - \frac{1}{2} I_1 {}^t_0\mathbf{C}_{ij}^{-1} \right) + k_0 J (J - 1) {}^t_0\mathbf{C}_{ij}^{-1} \quad i = 1, 2; j = 1, 2 \quad (23)$$

where δ_{ij} is the Kronecker Delta tensor; \mathbf{C}_{ij} is the right Cauchy-Green deformation tensor.

[Possible location of Figure 8]

Cutting simulation is carried out along the pre-planned path on the deforming (stretched) specimen model (Figure 8). At any stage of cutting, the steady-state deformations within the specimen are predicted by the proposed MTLADR algorithm. The deformed shape of the specimen before the change of cutting direction is shown in (Figure 10a and 10b). The specimen at the end of cutting, i.e. after the change in cutting direction is introduced, is shown in (Figure 10c). The total reaction force in x-axis direction (at the specimen's edge $x=0$ m) and strain energy at the end of cutting, which formed the basis of the convergence analysis, are reported in (Table 1).

[Possible location of Figure 9]

[Possible location of Figure 10]

[Possible location of Table 1]

3.2. *Reference solution for MTLADR algorithm verification*

As discussed in Introduction, there are a number of difficulties (such as the need for re-meshing) when attempting to simulate cutting using the finite element method. Therefore, when obtaining the reference solution for MTLADR algorithm verification using the established non-linear static solution procedures available in the commercial finite element code Abaqus, we do not directly simulate cutting but instead apply elongation to a pre-cut specimen model. The edges of the elements were aligned and separated along the edges of the cut (Figure 11). The specimen dimensions are the same as those of the MTLADR model. Implicit integration and standard linear quadrilateral plain strain elements with hybrid formulation are used in the Abaqus analysis. The boundary conditions are the same as the ones applied in the model solved using the MTLADR algorithm.

[Possible location of Figure 11]

As in both models (solved using the MTLADR algorithm and Abaqus) soft tissue is considered as a hyper-elastic material, the predicted deformations are independent of the loading history. This implies that despite the differences in method of introducing the discontinuity in the model using our MTLADR algorithm (cutting directly modelled) and reference model solved using Abaqus (the discontinuity introduced before deformation is applied), the steady state solution at the end of cutting should be the same.

H-refinement (mesh density increase) is used to obtain a converged reference solution in the model solved using Abaqus. As shown in (Table 2), the reaction force changes only by 0.3% and the strain energy remains virtually constant when the number of nodes increases from 7813 to 31572. Therefore, the solution for 7813 nodes is taken as the benchmark for verification. The deformed state (reference solution) obtained using Abaqus is shown in (Figure 12).

[Possible location of Table 2]

[Possible location of Figure 12]

3.3. Results of MTLADR algorithm verification

The reaction force, strain energy and nodal displacements predicted when modelling surgical cutting using the proposed MTLADR algorithm agree well with the reference solution obtained using the commercial finite element solver Abaqus.

The relative difference for the reaction force and strain energy calculated using the MTLADR algorithm and Abaqus solver at the end of cutting is 0.07% and 0.8% respectively. This is a very good agreement, considering that we used a grid of regular background integration cells for integration in the MTLADR algorithm, which is a source of integration errors along the cutting path, as the integration cells do not align with the geometry. Nevertheless, such regular integration cells are very easy to generate and do not require any mesh to be created.

To enable verification of the predicted deformations, the nodal displacements obtained using the MTLADR algorithm were calculated (through interpolation using the MLS shape functions) for the nodal positions of the Abaqus model. For the nodal displacement magnitudes, the maximum absolute difference between the results obtained using the MTLADR algorithm

and Abaqus is 0.31mm (1.56% of the imposed elongation) and the average difference (averaging over all model nodes) is only 0.018mm (0.09% of the imposed elongation) (Figure 13).

As illustrated in (Figure 13b), the absolute difference at most of the nodal positions (93.82%) is less than 0.05mm (0.25% of the imposed elongation) while it slightly increases to within 0.1mm (0.5% of the imposed elongation) for another 5.27% of nodal positions. At only few nodal positions (0.91%), the absolute difference ranges from 0.1 mm to 0.31 mm. Considering that the accuracy of the state-of-art image-guided neurosurgery techniques is not better than 1mm (Bucholz et al., 2004), the accuracy of the proposed MTLADR algorithm can be regarded as satisfying the requirements of computer-integrated surgery.

[Possible location of Figure 13]

4. Conclusions and discussion

We developed the Meshless Total Lagrangian Adaptive Dynamic Relaxation (MTLADR) algorithm, which belongs to the Element-Free Galerkin (EFG) family, for predicting the steady-state deformation of soft tissue at any stage of cutting. The features of the MTLADR algorithm include: (1) the effective modelling of surgical cutting using nodes with the aid of the level set method; (2) the Total Lagrangian formulation that makes it possible to conduct the most time consuming computations before the start of time-critical simulations; (3) the adaptive Dynamic Relaxation that facilitates fast convergence to the deformed (steady) state; (4) the capability of modelling large deformations and non-linear material properties of soft tissues exhibited in surgery. The MTLADR algorithm can generally be used to solve both 2D and 3D problems, although only 2D examples are presented in this paper.

With the benefits of no meshing and no re-meshing, we model 2D surgical cutting by creating a pair of nodes at each discretised position on the cutting path and introducing one node at the endpoint of the cut. The nodes of each nodal pair on the cutting path are allocated to opposite sides of the cutting path by using specific level set values. For explicit time integration, the field variable values at these nodes are either inherited or computed (using MLS interpolation) from the surrounding nodes. The effect of cutting is entirely reflected through the changes in the domain of influence of the nodes by efficiently implementing the visibility criterion using the level set method.

Numerical experiments are carried out to predict the behaviour of a deformed (stretched) specimen of soft tissue-like material during cutting. The computed reaction force, strain energy and nodal displacements exhibit good agreement with the reference solution obtained using the well established non-linear static solution procedures available in the commercial finite element software Abaqus.

Further investigation is being undertaken to extend the proposed algorithms to modelling and simulation of soft tissue cutting in 3D. Maintaining high computational speed despite increase in the computational cost associated with 3-D modelling will be possible with the help of parallel computation on graphics processing unit (GPU). The literature suggests that this can be achieved with ease (Joldes et al., 2010a; Joldes et al., 2010b).

Acknowledgments: The first author was supported by William & Marlene Schrader Postgraduate Scholarship during the completion of this research. The financial support of the Australian Research Council (Discovery Grants No. DP1092893 and No. DP120100402) and the National Health and Medical Research Council (Grant No. APP1006031) is gratefully acknowledged.

References

ABAQUS. 2009. ABAQUS Theory Manual Version 6.9. Providence, RI: Dassault Systèmes Simulia Corp.

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delignette, H., Grisoni, L., 2007. SOFA - an open source framework for medical simulation. In Proc. of Medicine Meets Virtual Reality (MMVR 15), Long Beach, CA, USA, pp. 1-6.

Allard, J., Courtecuisse, H., Faure, F., 2012. Chapter 21 - Implicit FEM Solver on GPU for Interactive Deformation Simulation, in: Wen-mei, W.H. (Ed.), GPU Computing Gems Jade Edition. Morgan Kaufmann, Boston, pp. 281-294. ISBN 978-0-12-385963-1.

Basdogan C, Ho CH, Srinivasan MA. 1999. Simulation of tissue cutting and bleeding for laparoscopic surgery using auxiliary surfaces. In: Medicine meets virtual reality. IOS Press.

Bathe K-J. 1996. Finite Element Procedures. New Jersey: Prentice-Hall.

Belytschko T, Krongauz Y, Organ D, Flaming M, Krysl P. 1996. Meshless methods: An overview and recent developments. Computer Methods in Applied Mechanics and Engineering. 139:3-47

Belytschko T, Lu YY, Gu L. 1994. Element-free Galerkin methods. International Journal for Numerical Methods in Engineering. 37:229-256

Belytschko T, Organ D, Gerlach C. 2000. Element-free Galerkin methods for dynamic fracture in concrete. Computer Methods in Applied Mechanics and Engineering. 187:385-399

Belytschko T, Tabbara M. 1996. Dynamic fracture using element-free Galerkin methods. International Journal for Numerical Methods in Engineering. 39:923-938

Bielser D, Glardon P, Teschner M, Gross M. 2004. A state machine for real-time cutting of tetrahedral meshes. Graphical Models. 66:398-417

Bordas S, Moran B. 2006. Enriched finite elements and level sets for damage tolerance assessment of complex structures. Engineering Fracture Mechanics. 73(9):1176-1201. doi: 10.1016/j.engfracmech.2006.01.006

Bruyns CD, Senger S, Menon A, Montgomery K, Wildermuth S, Boyle R. 2002. A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh-cutting using virtual tools. *Journal of Visualization and Computer Animation*. 13:21-42

Bucholz R, MacNeil W, McDurmott L. 2004. The operating room of the future. *Clin Neurosurg.* 51:228-237

Choi KS. 2006. Interactive cutting of deformable objects using force propagation approach and digital design analogy. *Computers & Graphics*. 30:233-243

Cotin S, Delingette H, Ayache N. 2000. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. *The Visual Computer*. 16:437-452

Courtecuisse H, Jung H, Allard J, Duriez C. 2010. GPU-based real-time soft tissue deformation with cutting and haptic feedback. *Prog Biophys Mol Biol*. 103:159-168

Doblaré M, Cueto E, Calvo B, Martínez MA, Garcia JM, Cegonino J. 2005. On the employ of meshless methods in biomechanics. *Computer Methods in Applied Mechanics and Engineering*. 194:801-821

Ebeida MS, Patney A, Owens JD, Mestreau E. 2011. Isotropic conforming refinement of quadrilateral and hexahedral meshes using two-refinement templates. *International Journal for Numerical Methods in Biomedical Engineering*. 88:974–985

Fung YC. 1993. Biomechanics. *Mechanical Properties of Living Tissues*. Second ed. New York: Springer-Verlag.

Greenfield, L.J., Mulholland, M.W., 2010. *Greenfield's Surgery Scientific Principles and Practice*, Fifth ed. Wolters Kluwer Health/Lippincott Williams & Wilkins, Philadelphia. ISBN 9781605473550.

Horton A, Wittek A, Joldes GR, Miller K. 2010. A Meshless Total Lagrangian Explicit Dynamics algorithm for surgical simulation. *International Journal for Numerical Methods in Biomedical Engineering*. 26(8):977-998

Ji S, Ford JC, Greenwald RM, Beckwith JG, Paulsen KD, Flashman LA, McAllister TW. 2011. Automated subject-specific, hexahedral mesh generation via image registration. *Finite Elements in Analysis and Design*. 47(10):1178-1185

Joldes GR, Wittek A, Miller K. 2011. An adaptive Dynamic Relaxation method for solving nonlinear finite element problems. Application to brain shift estimation. *International Journal for Numerical Methods in Biomedical Engineering.* 27(2):173-185. doi 10.1002/cnm.1407

Joldes GR, Wittek A, Miller K. 2009a. Computation of intra-operative brain shift using dynamic relaxation. *Computer Methods in Applied Mechanics and Engineering.* 198(41-44): 3313-3320. doi 10.1016/j.cma.2009.06.012

Joldes GR, Wittek A, Miller K. 2010a. Real-time nonlinear finite element computations on GPU - Application to neurosurgical simulation. *Computer Methods in Applied Mechanics and Engineering.* 199(49-52):3305-3314. doi 10.1016/j.cma.2010.06.037

Joldes GR, Wittek A, Miller K. 2010b. Real-time nonlinear finite element computations on gpu – handling of different element types, in: Wittek, A, Nielsen, P.M.F., Miller, K. (Eds) *Computational Biomechanics for Medicine,, Soft Tissues and Musculoskeletal System:* 73-80, Springer, New York, ISBN 978-1-4419-9618-3.

Joldes GR, Wittek A, Miller K. 2012. Stable time step estimates for mesh-free particle methods. *International Journal for Numerical Methods in Engineering.* doi 10.1002/nme.4290

Joldes GR, Wittek A, Miller K. 2009b. Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation. *Medical Image Analysis* 13(6):912-919. doi 10.1016/j.media.2008.12.001

Lim Y, Hu J, Chang C, Tardella N. 2006. Soft tissue deformation and cutting simulation for the multimodal surgery training. Paper presented at: 19th IEEE Symposium on Computer-Based Medical Systems, Salt Lake City, UT

Marechal L. 2009 Advances in Octree-Based All-Hexahedral Mesh Generation: Handling Sharp Features. *Proceedings of the 18th International Meshing Roundtable:* 65-84, Springer-Verlag, October 25-28..

Meseure P, Chaillou C. 1997 Deformable body simulation with adaptive subdivision and cuttings. *Proceedings of the WSCG'97; Plzen:* 361-370, 10-14 February

Miller K, Chinzei K, Orssengo G, Bednarz P. 2000. Mechanical properties of brain tissue in-vivo: experiment and computer simulation. *Journal of Biomechanics* 33:1369-1376.

Miller K, Joldes GR, Lance D, Wittek A. 2007. Total Lagrangian Explicit Dynamics finite element algorithm for computing soft tissue deformation. *Communications in Numerical Methods in Engineering*. 23:121-134

Miller K, Wittek A, Joldes GR. 2011. Biomechanical modelling of the brain for computer-assisted neurosurgery. In: *Biomechanics of the Brain*. New York: Springer. p. 111-136.

Mor AB 2001. Progressive cutting with minimal new element creation of soft tissue models for interactive surgical simulation [PhD]. Pittsburgh (PA): Carnegie Mellon University.

Osher S, Sethian JA. 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*. 79:12-49

Pan JJ, Chang J, Yang XS, Zhang JJ, Qureshi T, Howell R, Hickish T. 2011. Graphic and haptic simulation system for virtual laparoscopic rectum surgery. *International Journal of Medical Robotics and Computer Assisted Surgery*. 7:304-317

Rabczuk T, Belytschko T. 2007. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering*. 196:2777-2799

Rabczuk T, Bordas S, Zi G. 2010. On three-dimensional modelling of crack growth using partition of unity methods. *Computers & Structures*. 88(23-24):1391-1411. doi 10.1016/j.compstruc.2008.08.010

Schneiders R. 2000. Octree-based hexahedral mesh generation. *International Journal of Computational Geometry and Applications*. 10(4): 383-398.

Stolarska M, Chopp DL, Moes N, Belytschko T. 2001. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering* 51:943-960

Taylor ZA, Cheng M, Ourselin S. 2008. High-Speed Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units. *IEEE Transactions on Medical Imaging*. 27(5):650-663

Underwood P. 1983. Dynamic Relaxation. In: *Computational Methods for Transient Analysis*. Amsterdam: New-Holland: 245-265.

Vigneron LM, Verly JG, Warfield SK. 2004, On extended finite element method (XFEM) for modelling of organ deformations associated with surgical cuts. *Proceedings of the ISMS 2004, LNCS 2078: 311-318*. Springer-Verlag, Berlin Heidelberg.

Wang P, Becker AA, Jones IA, Glover AT, Benford SD, Greenhalgh CM, Vloeberghs M. 2007. Virtual reality simulation of surgery with haptic feedback based on the boundary element method. *Computers and Structures*. 85:331-339

Wang P, Becker AA, Jones IA, Glover AT, Benford SD, Greenhalgh CM, Vloeberghs M. 2006. A virtual reality surgery simulation of cutting and retraction in neurosurgery with force-feedback. *Comput Methods Programs Biomed*. 84:11-18

Westra, W.H., Phelps, T.H., Hruban, R.H., Isacson, C., 2002. *Surgical Pathology Dissection: An Illustrated Guide*. Springer. ISBN 0-387-95559-3.

Wittek A, Dutta-Roy T, Taylor ZA, Horton A, Washio T, Chinzei K, Miller K. 2008. Subject-specific non-linear biomechanical model of needle insertion into brain. *Computer Methods in Biomechanics Biomedical Engineering* 11(2):135-146

Wu W, Heng PA. 2005. An improved scheme of an interactive finite element model for 3D soft-tissue cutting and deformation. *Visual Computer*. 21:707-716

Yuan ZY, Ding YH, Zhang YY, Zhao JH. 2010. Real-time simulation of tissue cutting with CUDA based on GPGPU. *Advanced Materials Research*. 121-122:154-161

Zhang H. 2004. Simulating tissue dissection for surgical training [MSc]. Burnaby (BC): Simon Fraser University.

Zhang Y, Bajaj C. 2006. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*. 195(9-12):942-960.

Table 1. The study of spatial convergence of MTLADR 2D cutting algorithm: comparison of the strain energy and the reaction force when the nodal density (indicated by the number of nodes) is increased.

Number of nodes	121	441	1681	6561	25921
Strain energy [J]	0.54382	0.52351	0.51358	0.50663	0.50546
Reaction force [N]	49.5437	47.9239	47.1165	46.531	46.4683

Table 2. The H-refinement results for the model shown in (Figure 11) obtained using the non-linear static solution procedures available in Abaqus. Comparison of the reaction force and strain energy when the mesh density (indicated by the number of nodes) is increased.

Number of nodes	144	515	1925	7813	31572
Strain energy [J]	0.52627	0.51547	0.509889	0.506979	0.505576
Reaction force [N]	49.2445	47.9239	47.2442	46.9095	46.7666

List of figures:

Figure 1. The cutting path in the problem domain is represented by a series of line segments (T0-T1, T1-T2 and T2-T3).

Figure 2. The initialisation of level set values of functions ψ and ϕ for nodes/integration points in the problem domain: $\psi = 0$ on L and L1 and $\phi = 0$ on L2.

Figure 3. Illustration of the cutting path discretisation using nodes. After discretisation, the nodes are divided by the level set value of function ψ : (a) $\psi = 0$ (cutting path); (b) $\psi > 0$; (c) $\psi < 0$.

Figure 4. Progression of cutting without direction change (cutting line segments T0-T1 and T1-T2 are co-linear). The shaded area, where $\phi^n < 0$, is a non-update area $A^{non-update}$ while the rest of the domain is an update area A^{update} .

Figure 5. Progression of cutting with direction change: the cutting proceeds from point T2 to T3 with direction change. Level set function $\bar{\phi}^n$ is obtained by rotating level set function ϕ^n until it is orthogonal to the current cutting vector V^{n+1} . The shaded area, where $\bar{\phi}^n \leq 0$, is a non-update area $A_c^{non-update}$ while the rest is an update area A_c^{update} .

Figure 6. The influence domain of node N1 intersects the cutting line L; points P1 and P2 should be eliminated from the influence domain of this node. The influence domain of node N2 goes through the cutting endpoint T only, so it does not need an update.

Figure 7. Update of influence domains of the nodes affected by cutting. Points P3 and P4 are removed from the influence domain of node N (step 3.a of the algorithm for updating nodal influence domains). According to step 3.b of the algorithm for updating nodal influence domains, point P5 must be removed from the domain of influence of node N while points P6 and P7 are kept.

Figure 8. Model for verification of the MTLADR algorithm. Cutting is carried out in the stretched specimen of soft tissue-like material along the pre-defined path shown using thick line segments. Dimensions are in metres (m).

Figure 9. Convergence study: variation of strain energy with the number of nodes used for discretisation. Comparison between MTLADR and Abaqus.

Figure 10. Results obtained using the MTLADR algorithm showing different stages of cutting. Dimensions are in metres (m).

Figure 11. The initial configuration of the finite element model with 7813 nodes built using the commercial finite element code Abaqus. Dimensions are in metres (m).

Figure 12. The deformed state of the finite element model with 7813 nodes. The results are used as the reference solution for verification of the MTLADR algorithm. Dimensions are in metres (m).

Figure 13 Results of verification the proposed MTLADR algorithm. Distribution of the absolute difference between the displacement magnitudes computed using the MTLADR algorithm and the reference results from the established non-linear static solution procedures available in the Abaqus commercial finite element code: a) Spatial distribution. b) Statistical distribution. The displacements are at the nodal positions of the model (shown in Figure 11) built in Abaqus.

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delingette, H., Grisoni, L., 2007. SOFA - an open source framework for medical simulation., Medicine Meets Virtual Reality (MMVR 15), Long Beach, CA, USA, pp. 1-6.

Allard, J., Courtecuisse, H., Faure, F., 2012. Chapter 21 - Implicit FEM Solver on GPU for Interactive Deformation Simulation, in: Wen-mei, W.H. (Ed.), GPU Computing Gems Jade Edition. Morgan Kaufmann, Boston, pp. 281-294. 978-0-12-385963-1.

Basdogan, C., Ho, C.H., Srinivasan, M.A., 1999. Simulation of tissue cutting and bleeding for laparoscopic surgery using auxiliary surfaces, in: Westwood, J.D. (Ed.), Medicine meets virtual reality. IOS Press.

Bathe, K.-J., 1996. Finite Element Procedures. Prentice-Hall.

Belytschko, T., Krongauz, Y., Organ, D., Flaming, M., Krysl, P., 1996. Meshless methods: An overview and recent developments. Computer Methods in Applied Mechanics and Engineering 139, 3-47.

Belytschko, T., Lu, Y.Y., Gu, L., 1994. Element-free Galerkin methods. International Journal for Numerical Methods in Engineering 37, pp. 229-256.

Belytschko, T., Organ, D., Gerlach, C., 2000. Element-free Galerkin methods for dynamic fracture in concrete. Computer Methods in Applied Mechanics and Engineering 187, 385-399.

Belytschko, T., Tabbara, M., 1996. Dynamic fracture using element-free Galerkin methods. International Journal for Numerical Methods in Engineering 39, 923-938.

Bielser, D., Glardon, P., Teschner, M., Gross, M., 2004. A state machine for real-time cutting of tetrahedral meshes. Graphical Models 66, 398-417.

Bordas, S., Moran, B., 2006. Enriched finite elements and level sets for damage tolerance assessment of complex structures. Engineering Fracture Mechanics 73, 1176-1201.

Bruyns, C.D., Senger, S., Menon, A., Montgomery, K., Wildermuth, S., Boyle, R., 2002. A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh-cutting using virtual tools. Journal of Visualization and Computer Animation 13, 21-42.

Bucholz, R., MacNeil, W., McDurmott, L., 2004. The operating room of the future. Clinical Neurosurgery 51, 228-237.

Choi, K.S., 2006. Interactive cutting of deformable objects using force propagation approach and digital design analogy. Computers & Graphics 30, 233-243.

Cotin, S., Delingette, H., Ayache, N., 2000. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. The Visual Computer 16, 437-452.

Courtecuisse, H., Jung, H., Allard, J., Duriez, C., 2010. GPU-based real-time soft tissue deformation with cutting and haptic feedback. Prog. Biophys. Mol. Biol. 103, 159-168.

Doblaré, M., Cueto, E., Calvo, B., Martínez, M.A., Garcia, J.M., Cegonino, J., 2005. On the employ of meshless methods in biomechanics. Computer Methods in Applied Mechanics and Engineering 194, 801-821.

Ebeida, M.S., Patney, A., Owens, J.D., Mestreau, E., 2011. Isotropic conforming refinement of quadrilateral and hexahedral meshes using two-refinement templates. International Journal for Numerical Methods in Biomedical Engineering 88, 974-985.

Fung, Y.C., 1993. Biomechanics. Mechanical Properties of Living Tissues, Second ed. Springer-Verlag, New York.

Greenfield, L.J., Mulholland, M.W., 2010. Greenfield's Surgery Scientific Principles and Practice, Fifth ed. Wolters Kluwer Health/Lippincott Williams & Wilkins, Philadelphia. ISBN 9781605473550.

Horton, A., Wittek, A., Joldes, G.R., Miller, K., 2010. A Meshless Total Lagrangian Explicit Dynamics Algorithm for Surgical Simulation. *International Journal for Numerical Methods in Biomedical Engineering* 26, 977-998.

Ji, S., Ford, J.C., Greenwald, R.M., Beckwith, J.G., Paulsen, K.D., Flashman, L.A., McAllister, T.W., 2011. Automated subject-specific, hexahedral mesh generation via image registration. *Finite Elements in Analysis and Design* 47, 1178-1185.

Joldes, G.R., Wittek, A., Miller, K., 2009a. Computation of intra-operative brain shift using dynamic relaxation. *Computer Methods in Applied Mechanics and Engineering* 198, 3313-3320.

Joldes, G.R., Wittek, A., Miller, K., 2009b. Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation. *Med. Image Anal.* 13, 912-919.

Joldes, G.R., Wittek, A., Miller, K., 2010a. Real-Time Nonlinear Finite Element Computations on GPU - Application to Neurosurgical Simulation. *Computer Methods in Applied Mechanics and Engineering* 199, 3305-3314.

Joldes, G.R., Wittek, A., Miller, K., 2010b. Real-time nonlinear finite element computations on gpu – handling of different element types, in: Wittek, A., Nielsen, P.M.F., Miller, K. (Eds.), *Computational Biomechanics for Medicine V Workshop*, MICCAI, Beijing, pp. 69-77.

Joldes, G.R., Wittek, A., Miller, K., 2011. An adaptive Dynamic Relaxation method for solving nonlinear finite element problems. Application to brain shift estimation. *International Journal for Numerical Methods in Biomedical Engineering* 27, 173-185.

Joldes, G.R., Wittek, A., Miller, K., 2012. Stable time step estimates for mesh-free particle methods. *International Journal for Numerical Methods in Engineering* Accepted on January 4th, 2012.

Lim, Y., Hu, J., Chang, C., Tardella, N., 2006. Soft tissue deformation and cutting simulation for the multimodal surgery training, 19th IEEE Symposium on Computer-Based Medical Systems, Salt Lake City, UT pp. 635-640.

Liu, G.R., 2003. Mesh Free Methods: Moving Beyond the Finite Element Method. CRC Press, Boca Raton.

Marechal, L., 2009. Advances in Octree-Based All-Hexahedral Mesh Generation: Handling Sharp Features, *Proceedings of the 18th International Meshing Roundtable*, pp. 65-84.

Meseure, P., Chaillou, C., 1997. Deformable body simulation with adaptive subdivision and cuttings, *WSCG'97*, Pizen, pp. 361-370.

Miller, K., Chinzei, K., Orssengo, G., Bednarz, P., 2000. Mechanical properties of brain tissue in-vivo: experiment and computer simulation. *Journal of Biomechanics* 33, 1369-1376.

Miller, K., Joldes, G.R., Lance, D., Wittek, A., 2007. Total Lagrangian Explicit Dynamics Finite Element Algorithm for Computing Soft Tissue Deformation. *Communications in Numerical Methods in Engineering* 23, 121-134.

Miller, K., Wittek, A., Joldes, G.R., 2011. Biomechanical modelling of the brain for computer-assisted neurosurgery, in: Miller, K. (Ed.), *Biomechanics of the Brain*. Springer, New York, pp. 111-136. 978-1-441999-96-2.

Mor, A.B., 2001. Progressive cutting with minimal new element creation of soft tissue models for interactive surgical simulation. Carnegie Mellon University, Pittsburgh (PA).

OSher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 79, 12-49.

Pan, J.J., Chang, J., Yang, X.S., Zhang, J.J., Qureshi, T., Howell, R., Hickish, T., 2011. Graphic and haptic simulation system for virtual laparoscopic rectum surgery. *International Journal of Medical Robotics and Computer Assisted Surgery* 7, 304-317.

Rabczuk, T., Belytschko, T., 2007. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering* 196, 2777-2799.

Rabczuk, T., Bordas, S., Zi, G., 2010. On three-dimensional modelling of crack growth using partition of unity methods. *Computers & Structures* 88, 1391-1411.

Schneiders, R., 2000. Octree-based hexahedral mesh generation. *International Journal of Computational Geometry and Applications* 10, 383-398.

Stolarska, M., Chopp, D.L., Moes, N., Belytschko, T., 2001. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering* 51, 943-960.

Taylor, Z.A., Cheng, M., Ourselin, S., 2008. High-Speed Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units. *IEEE Transactions on Medical Imaging* 27, 650-663.

Underwood, P., 1983. Dynamic Relaxation, in: Belytschko, T., Hughes, T.J.R. (Eds.), *Computational Methods for Transient Analysis*. New-Holland, Amsterdam, pp. 245-265.

Vigneron, L.M., Verly, J.G., Warfield, S.K., 2004. On extended finite element method (XFEM) for modelling of organ deformations associated with surgical cuts, in: D., C.S.a.M. (Ed.), *ISMS 2004, LNCS*. Springer-Verlag Berlin Heidelberg, pp. 134-143.

Wang, P., Becker, A.A., Jones, I.A., Glover, A.T., Benford, S.D., Greenhalgh, C.M., Vloeberghs, M., 2006. A virtual reality surgery simulation of cutting and retraction in neurosurgery with force-feedback. *Comput. Methods Programs Biomed.* 84, 11-18.

Wang, P., Becker, A.A., Jones, I.A., Glover, A.T., Benford, S.D., Greenhalgh, C.M., Vloeberghs, M., 2007. Virtual reality simulation of surgery with haptic feedback based on the boundary element method. *Computers and Structures* 85, 331-339.

Westra, W.H., Phelps, T.H., Hruban, R.H., Isaacson, C., 2002. *Surgical Pathology Dissection An Illustrated Guide*. Springer. ISBN 0-387-95559-3.

Wittek, A., Dutta-Roy, T., Taylor, Z.A., Horton, A., Washio, T., Chinzei, K., Miller, K., 2008. Subject-specific non-linear biomechanical model of needle insertion into brain. *Comput. Methods Biomech. Biomed. Engin.* 11, 135-146.

Wu, W., Heng, P.A., 2005. An improved scheme of an interactive finite element model for 3D soft-tissue cutting and deformation. *Visual Computer* 21, 707-716.

Yuan, Z.Y., Ding, Y.H., Zhang, Y.Y., Zhao, J.H., 2010. Real-time simulation of tissue cutting with CUDA based on GPGPU. *Advanced Materials Research* 121-122, 154-161.

Zhang, H., 2004. Simulating tissue dissection for surgical training. Simon Fraser University, Burnaby (BC).

Zhang, Y., Bajaj, C., 2006. Adaptive and Quality Quadrilateral/Hexahedral Meshing from Volumetric Data. Computer Methods in Applied Mechanics and Engineering 195, 942-960.

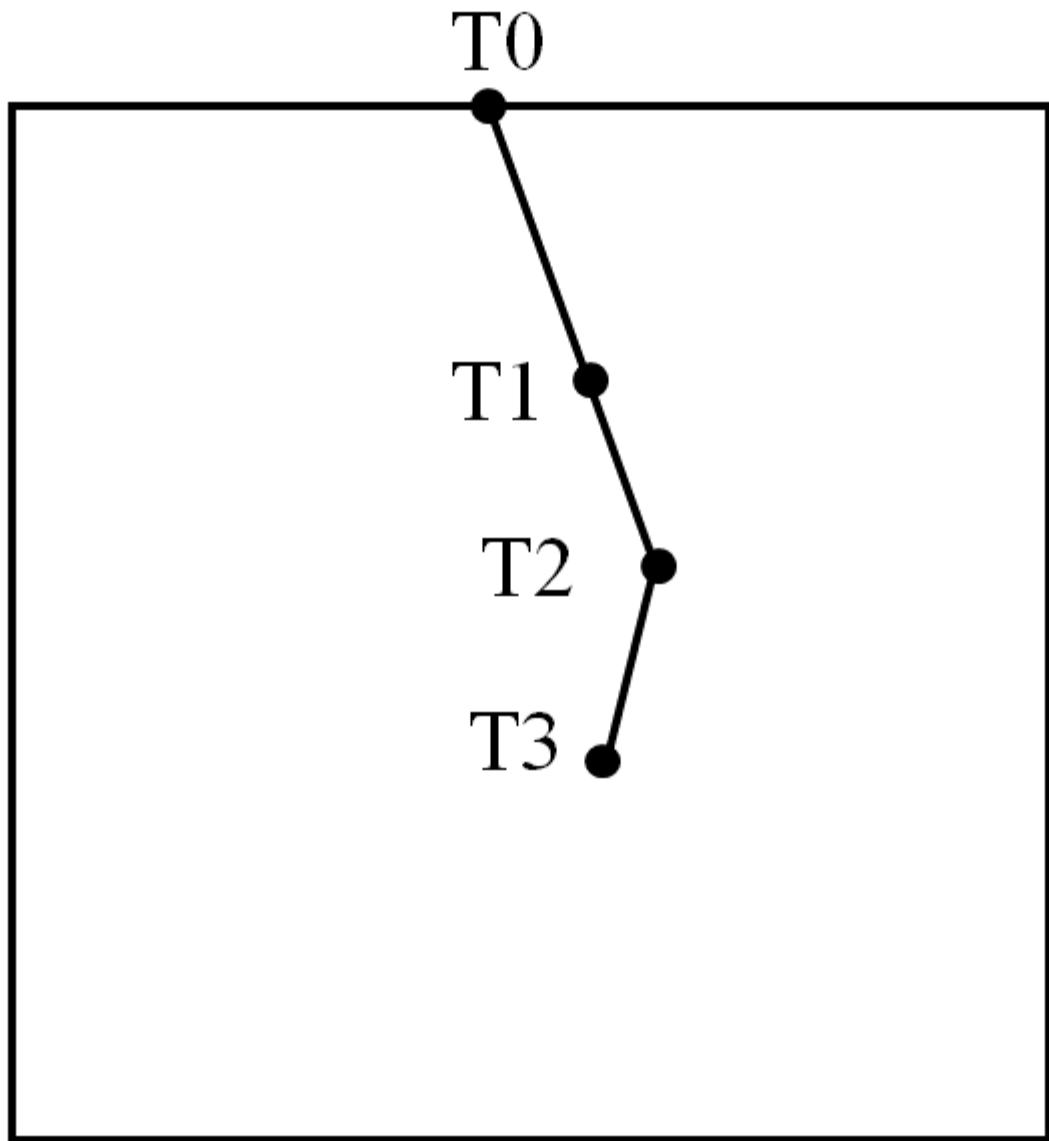


Figure 1

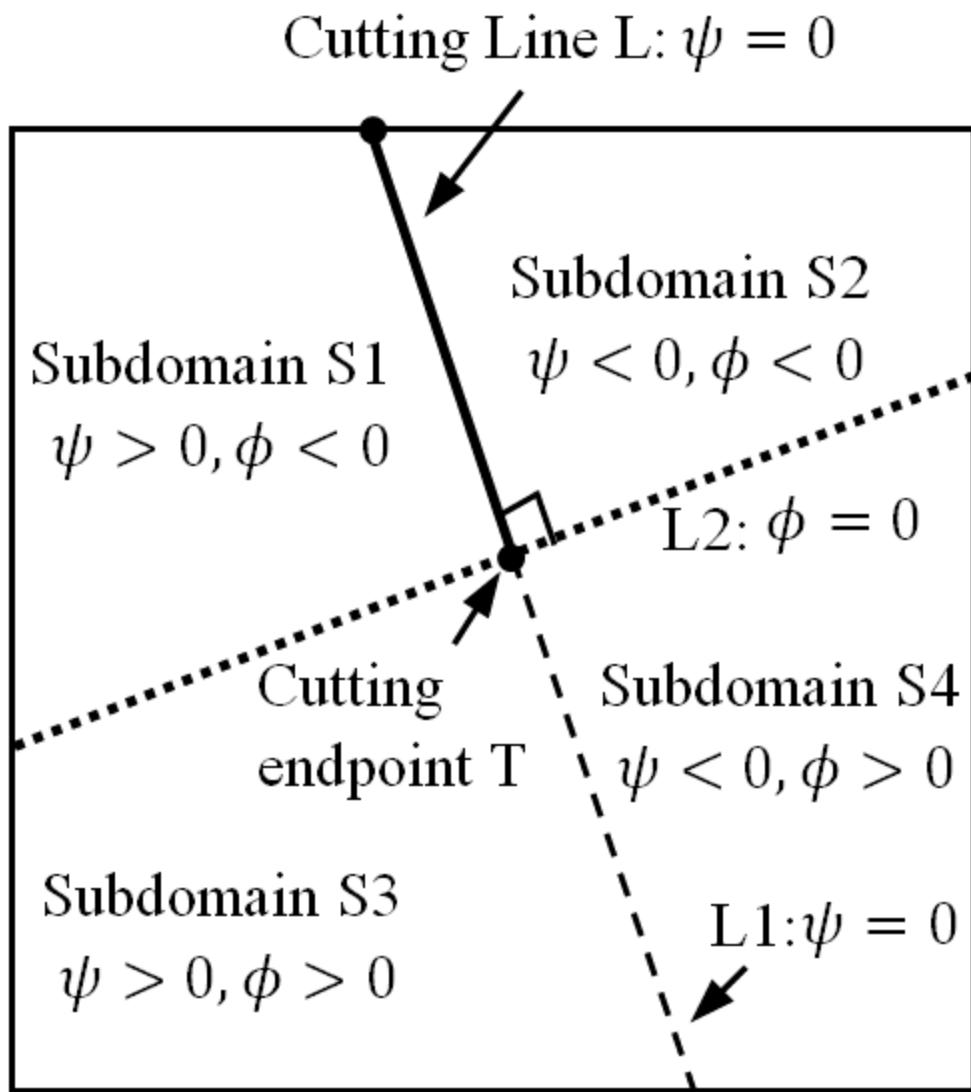


Figure 2

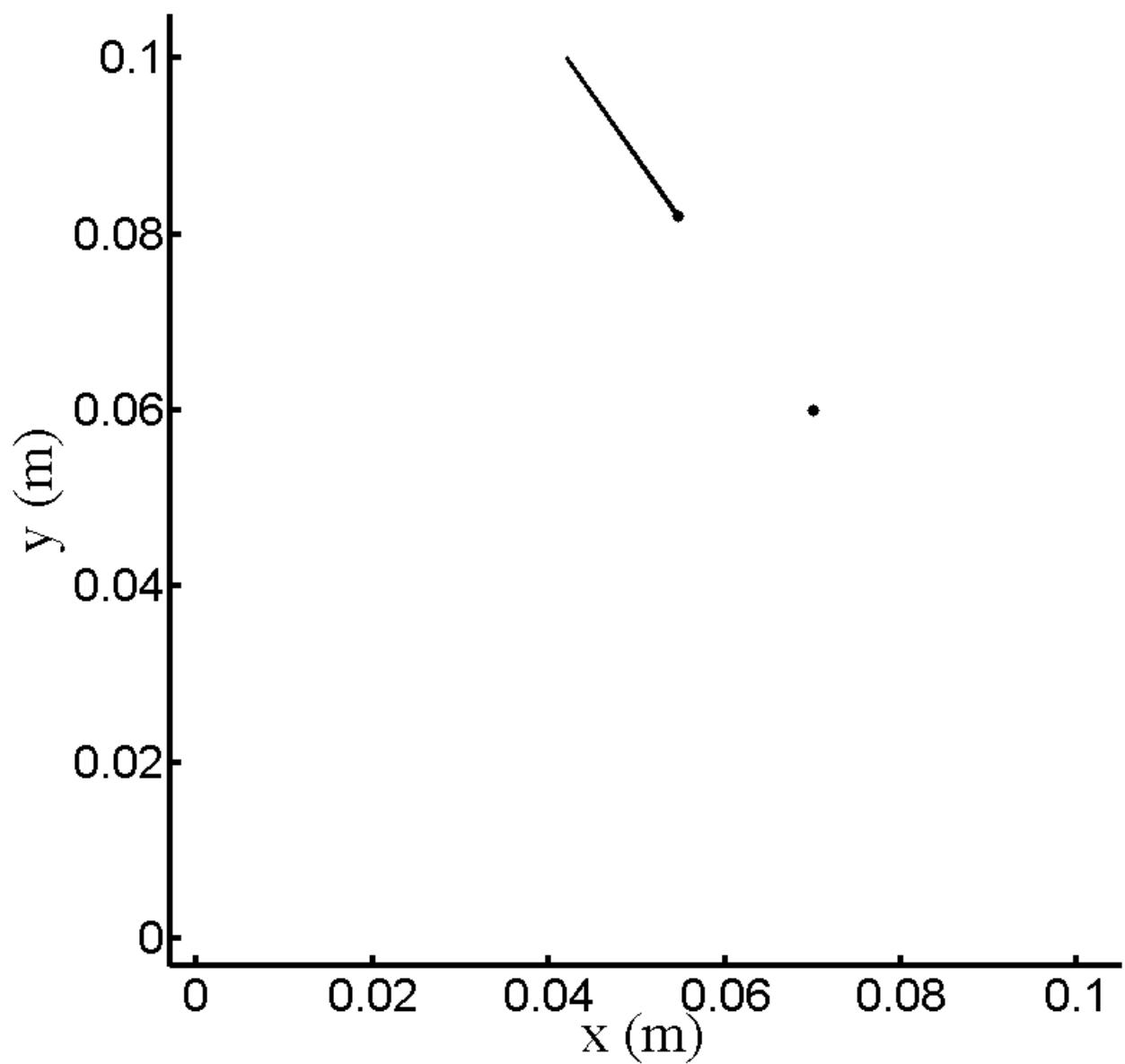


Figure 3a

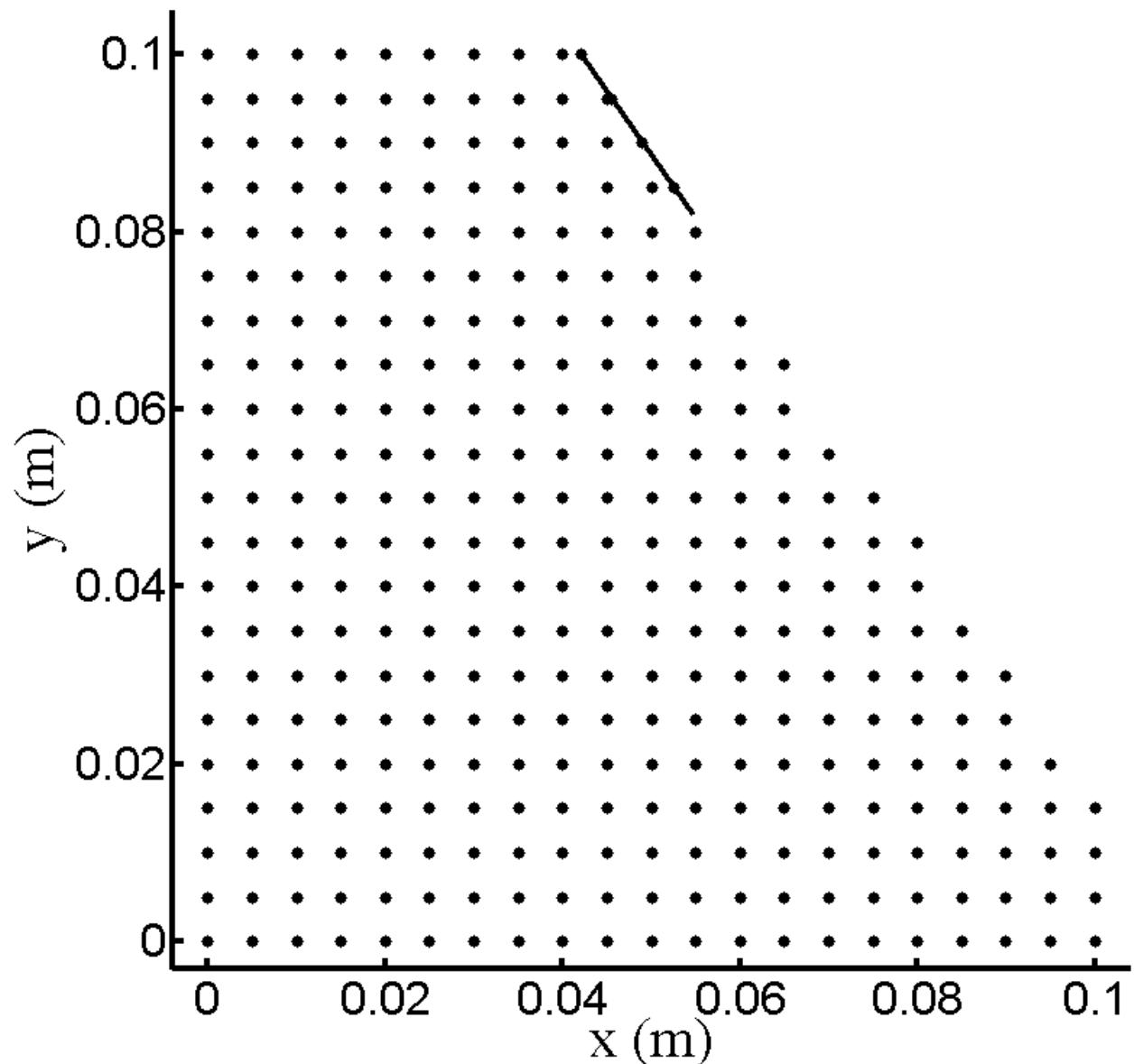


Figure 3b

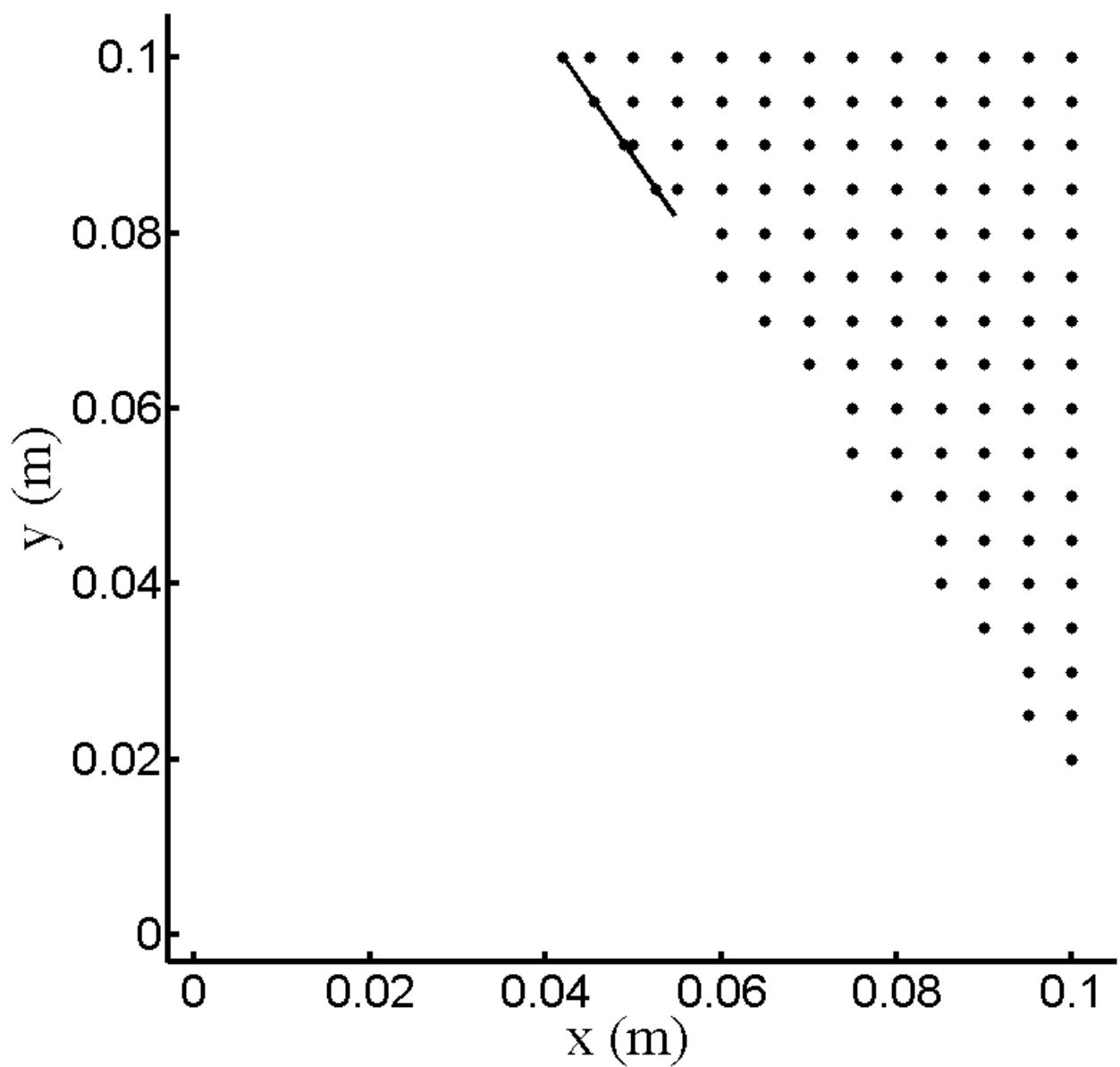


Figure 3c

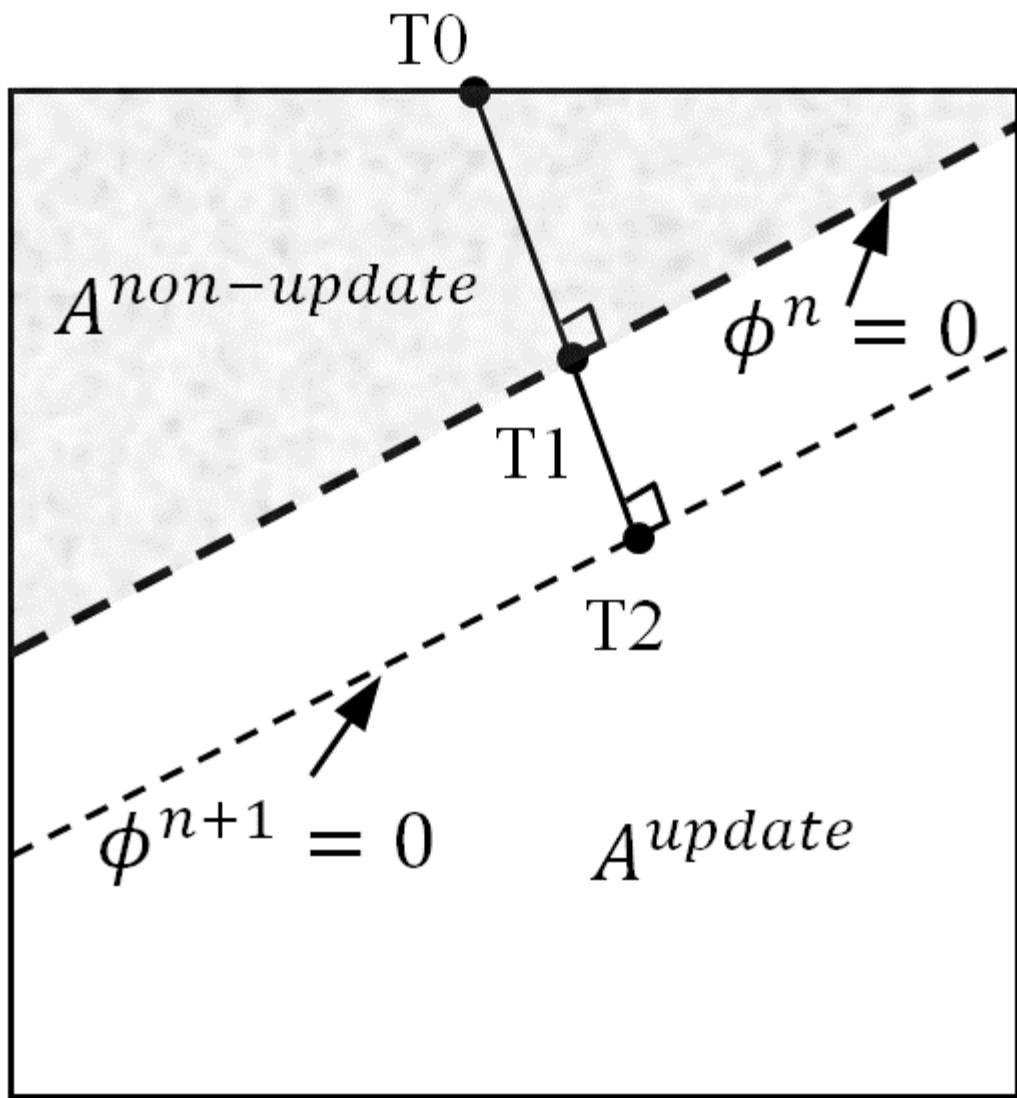


Figure 4

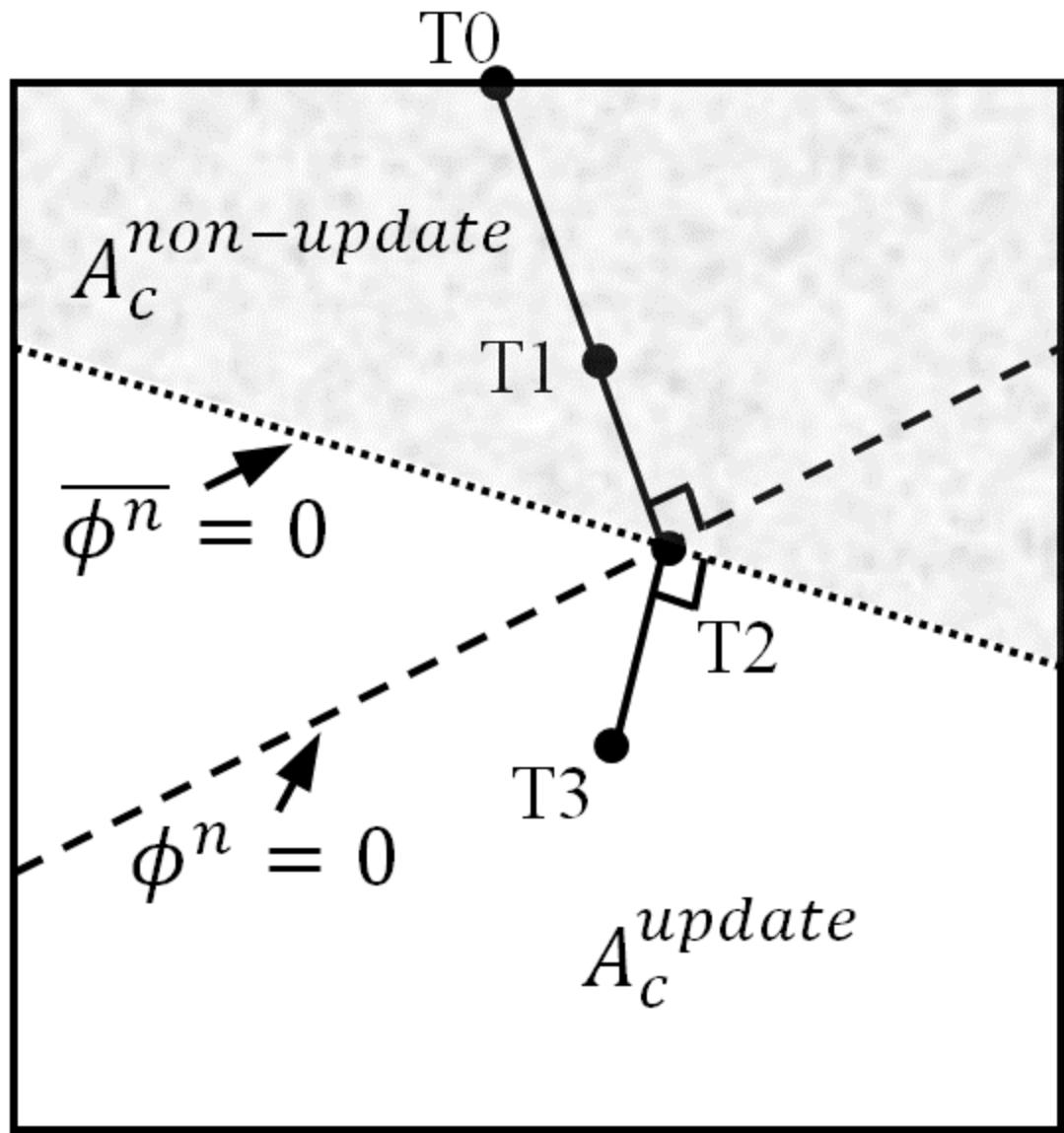


Figure 5

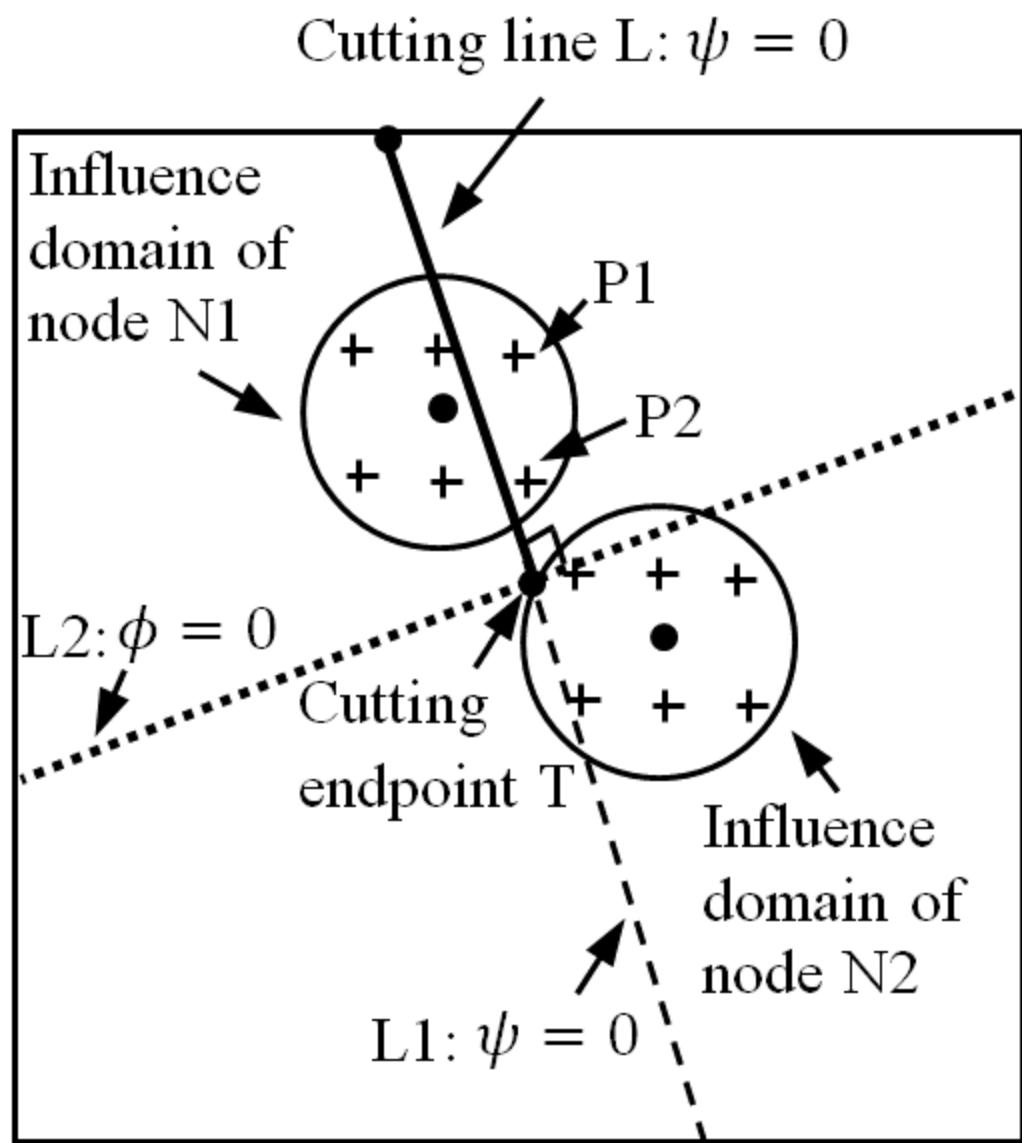


Figure 6

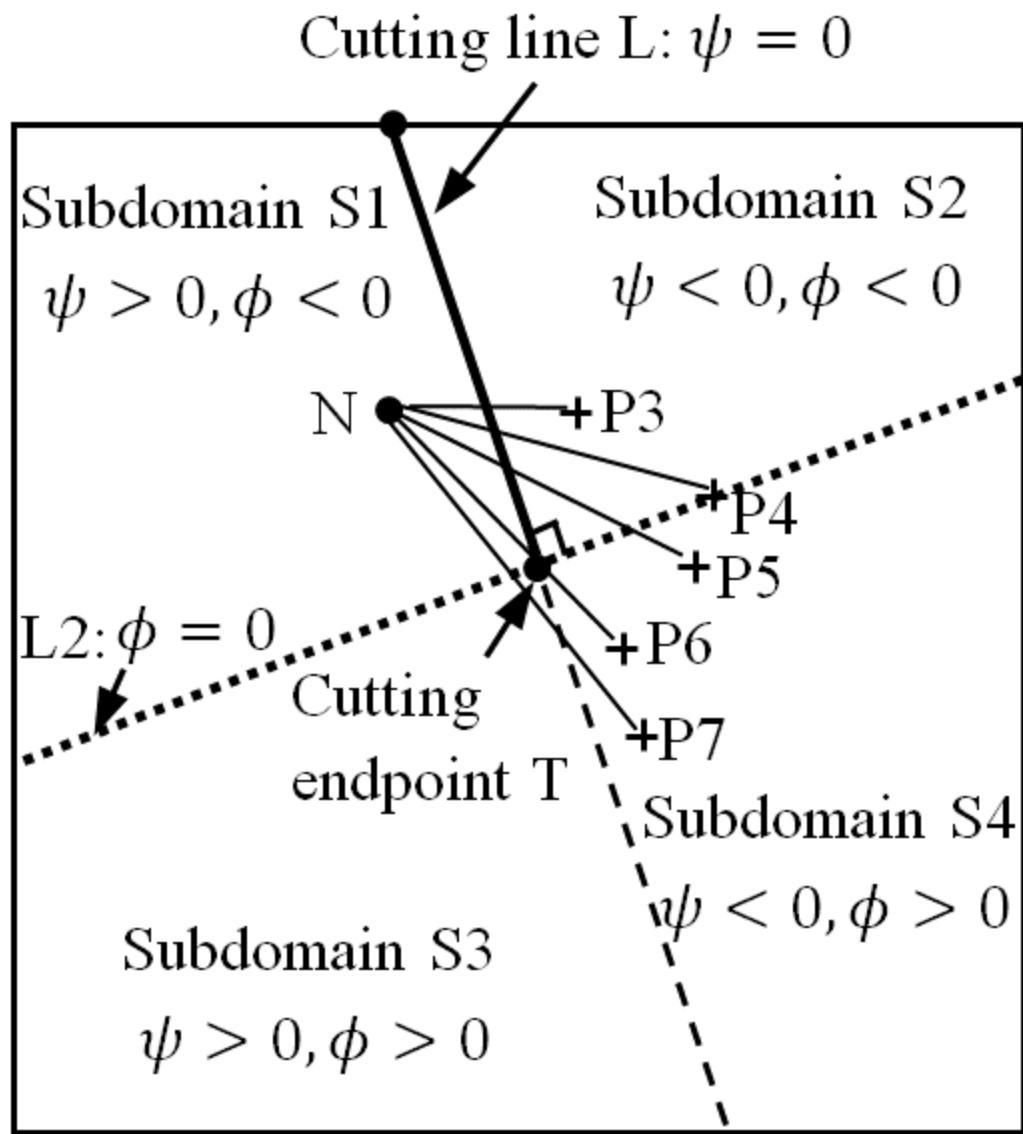


Figure 7

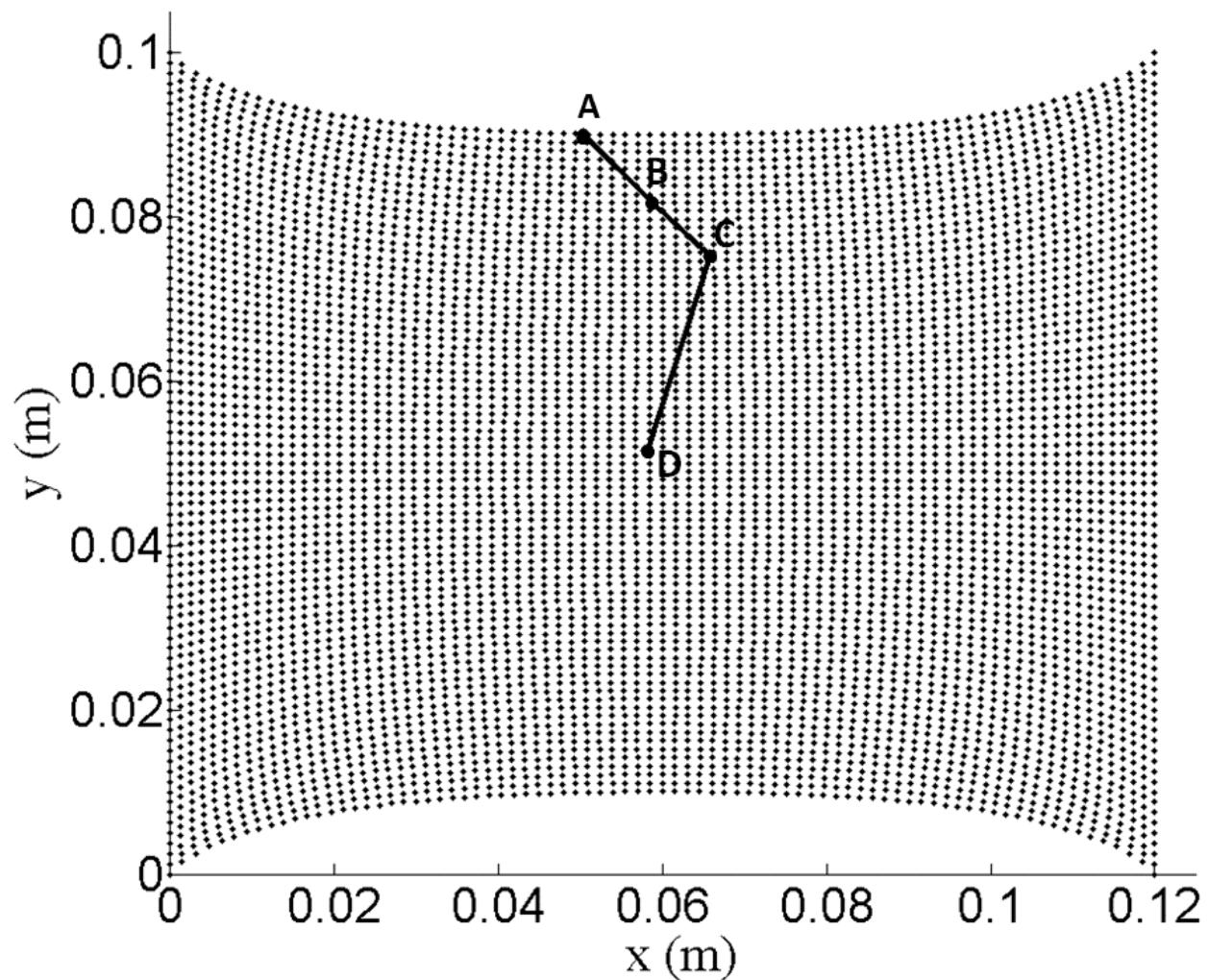


Figure 8

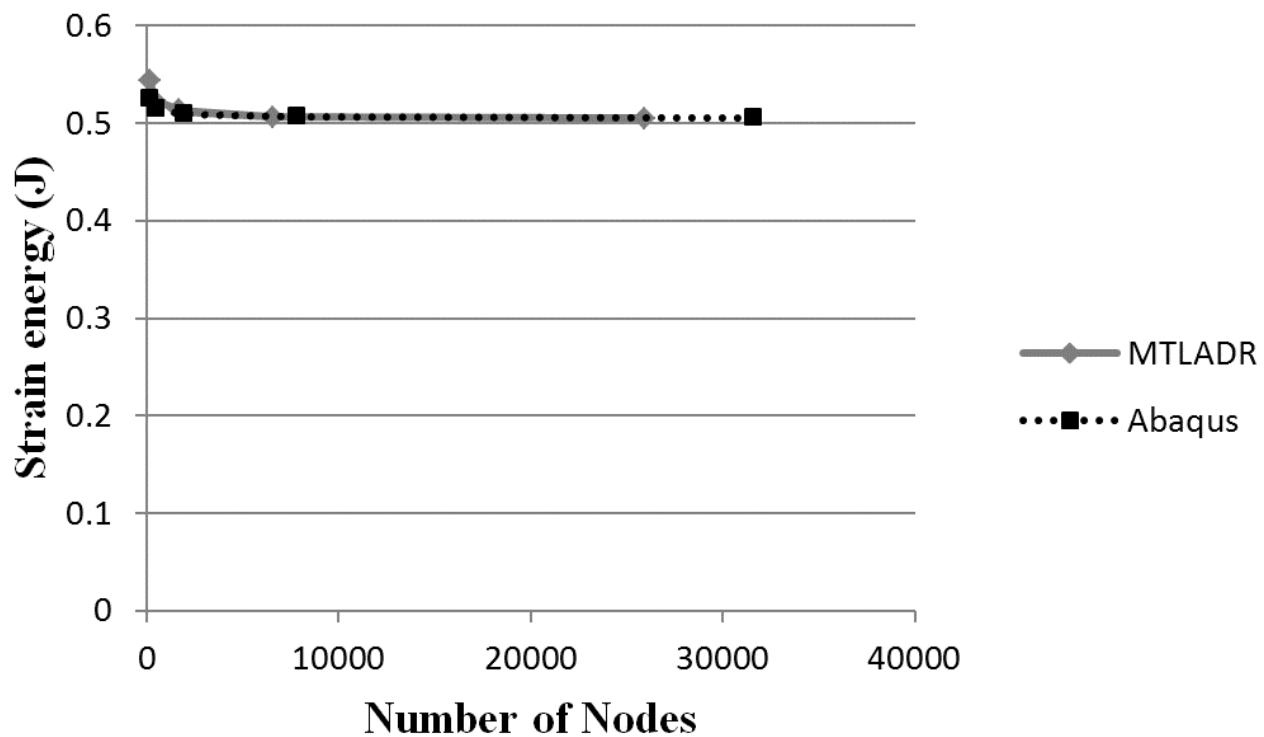


Figure 9

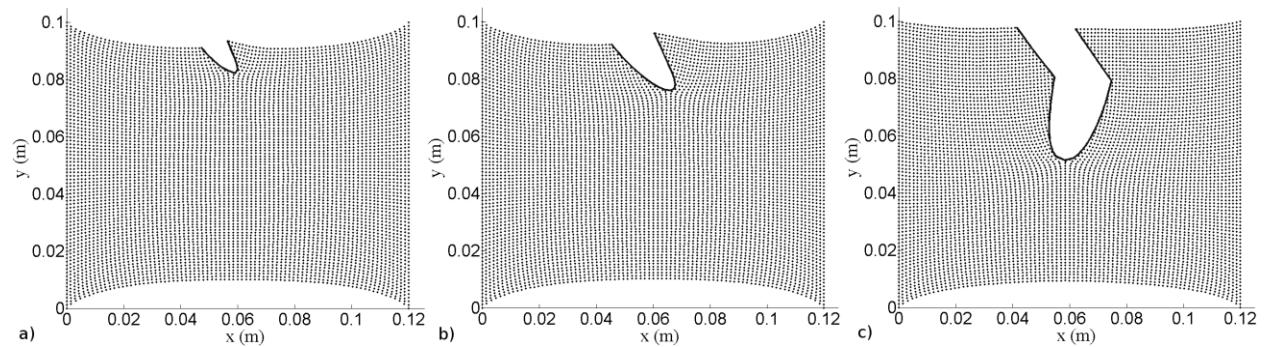


Figure 10

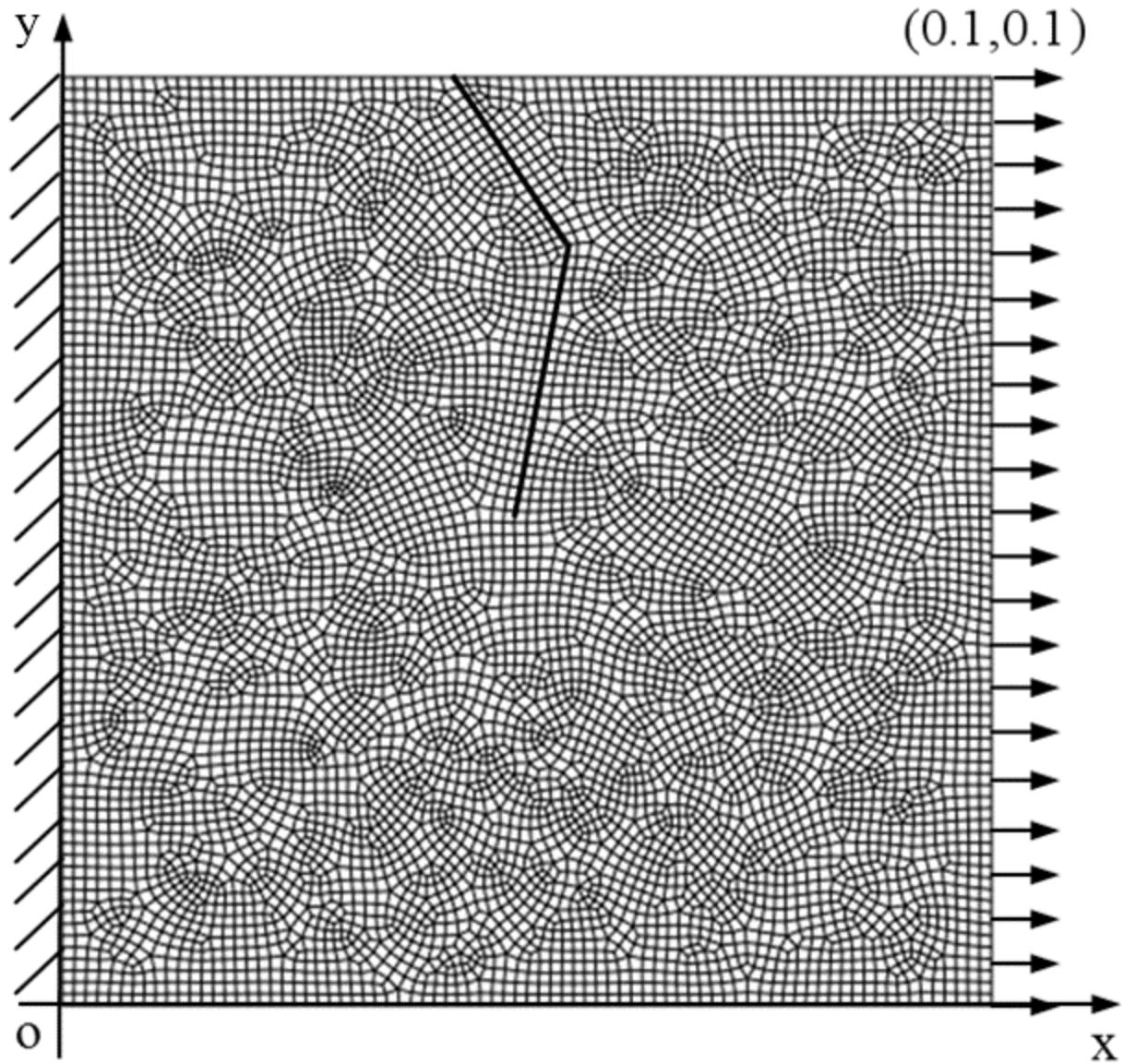


Figure 11

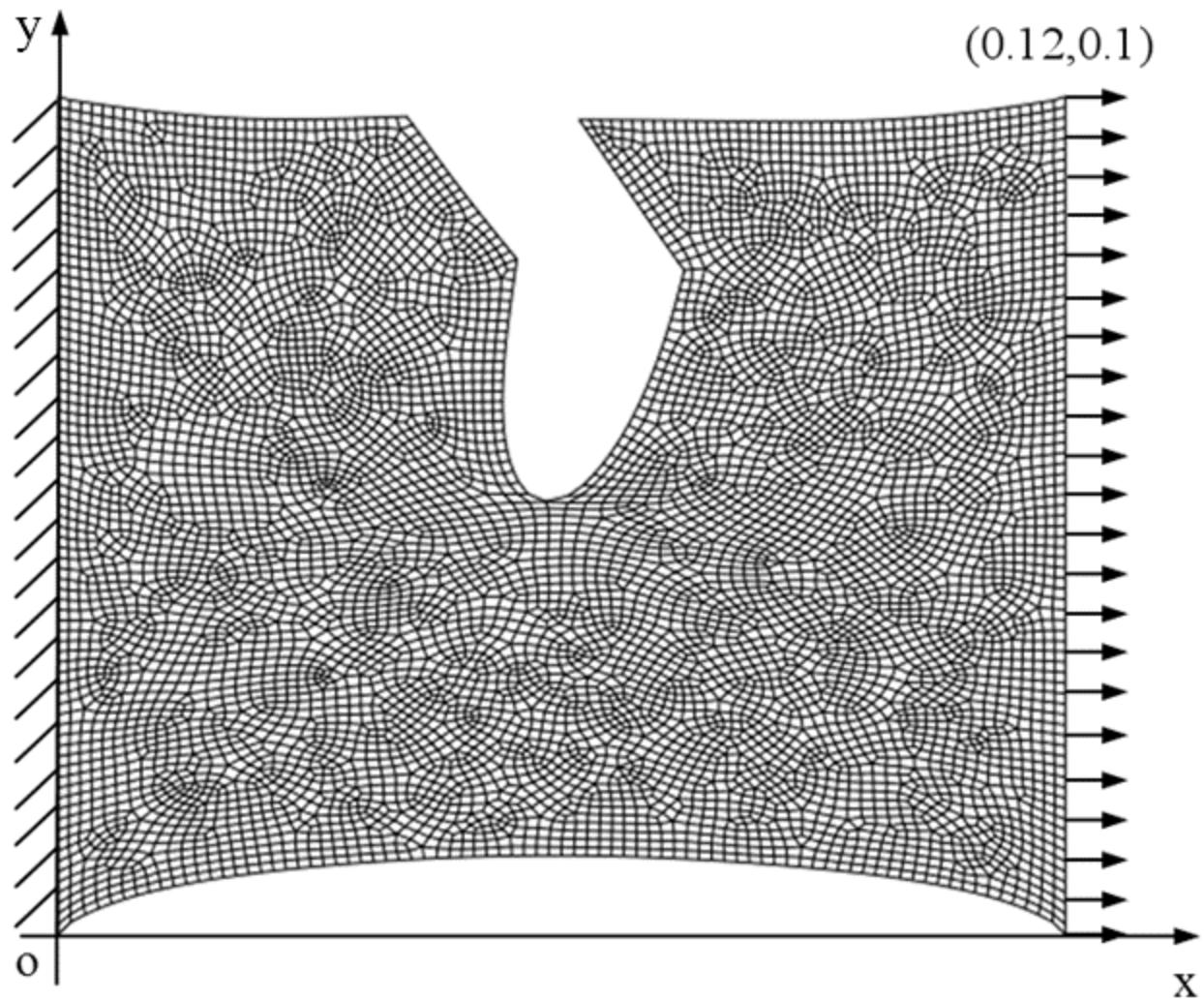


Figure 12

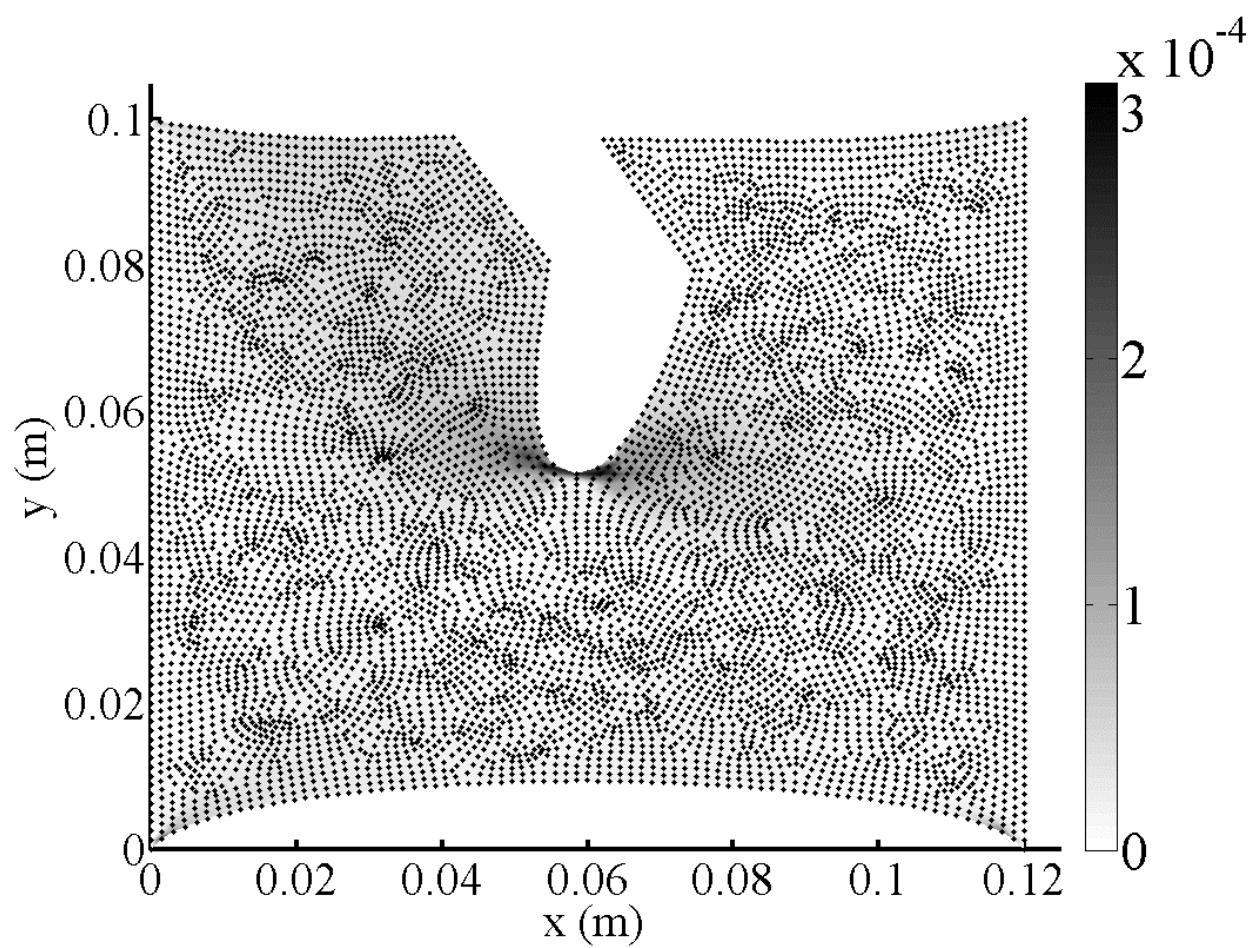


Figure 13a

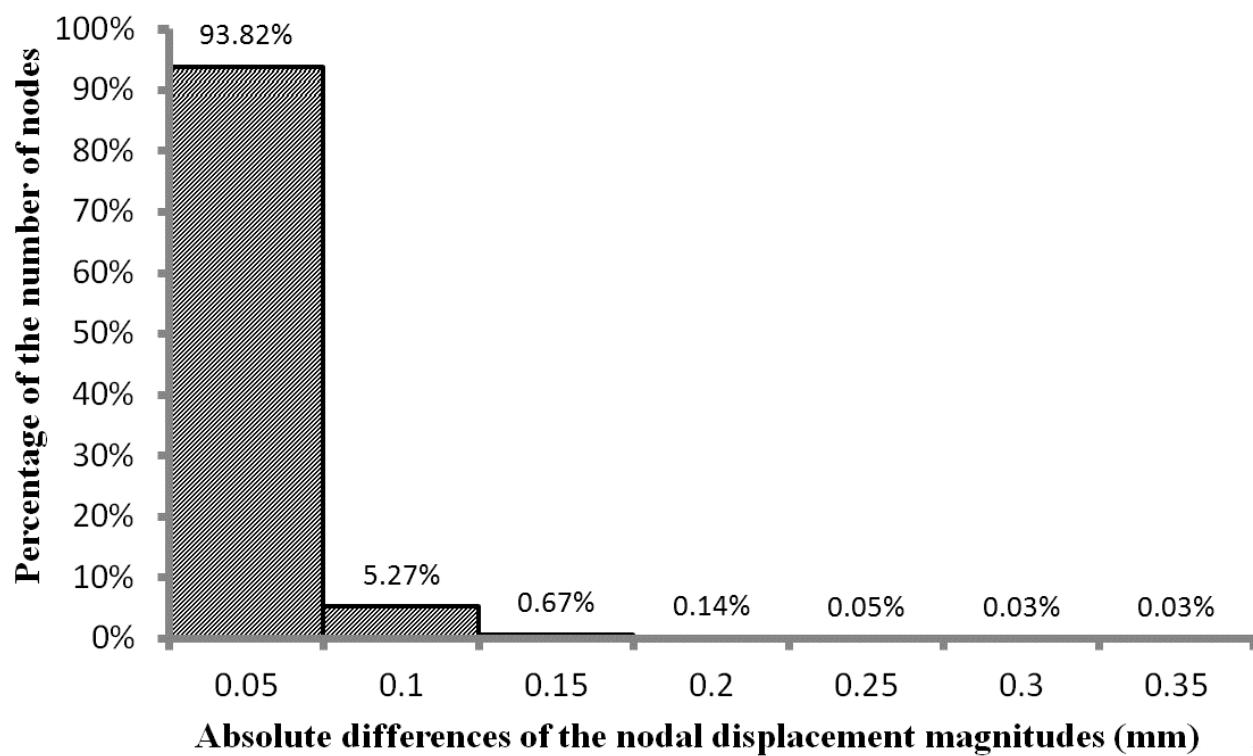


Figure 13b