3D Algorithm for Simulation of Soft Tissue Cutting

Xia Jin, Grand Roman Joldes, Karol Miller, and Adam Wittek

Abstract Modelling and simulation of soft tissue cutting in 3D remain one of the most challenging problems in surgery simulation, not only because of the nonlinear geometric and material behaviour exhibited by soft tissue but also due to the complexity of introducing the cutting-induced discontinuity. In most publications, the progressive surgical cutting is modelled by conventional finite element (FE) method, in which the high computational cost and error accumulation due to re-meshing constrain the computational efficiency and accuracy. In this paper, a meshless Total Lagrangian Adaptive Dynamic Relaxation (MTLADR) 3D cutting algorithm is proposed to predict the steady-state responses of soft tissue at any stage of surgical cutting in 3D. The MTLADR 3D algorithm features a spatial discretisation using a cloud of nodes. With the benefits of no meshing and no re-meshing, the cutting-induced discontinuity is modelled and simulated by adding nodes on the cutting faces and implementing the visibility criterion with the aid of the level set method. The accuracy of the MTLADR 3D cutting algorithm is verified against the established nonlinear solution procedures available in commercial FE software Abaqus.

X. Jin • G.R. Joldes • A. Wittek
Intelligent Systems for Medicine Laboratory, The University of Western Australia, Perth, Australia
e-mail: jinxiach@gmail.com; grand.joldes@uwa.edu.au; adam.wittek@uwa.edu.au

K. Miller (✉)
Intelligent Systems for Medicine Laboratory, The University of Western Australia, Perth, Australia

Institute of Mechanics and Advanced Materials, Cardiff School of Engineering, Cardiff University, Wales, UK
e-mail: karol.miller@uwa.edu.au

A. Wittek et al. (eds.), *Computational Biomechanics for Medicine: Models, Algorithms and Implementation*, DOI 10.1007/978-1-4614-6351-1_6,

1 Introduction

Surgery simulation has great significance in extending surgeons' ability to learn, plan and carry out surgical interventions more accurately and less invasively. Potential applications include surgical simulators for highly realistic surgical training and planning, non-rigid registration in image-guided surgery systems and computer-aided design of medical devices and procedures.

Modelling and simulation of soft tissue cutting remain one of the most challenging problems in surgery simulation. The challenges exist in the complexity of introducing cutting-induced discontinuity and the capability of handling the nonlinear geometric and material behaviour of soft tissue [1–3] while reducing the high computational cost of 3D simulation.

So far, the progressive surgical cutting has been modelled and simulated by subdivision of elements of the volumetric mesh using conventional finite element (FE) method [4–7]. Even when using sophisticated re-meshing technologies, the FE method tends to become unstable and its accuracy deteriorates when the mesh undergoes distortion and fragmentation due to large deformations and cutting [8]. Despite the exploration of speed-up technologies [7, 9, 10], the high computational cost constrains the computational efficiency of FE method in surgical cutting simulation.

In this paper, a meshless Total Lagrangian Aadaptive Dynamic Relaxation (MTLADR) 3D cutting algorithm is developed to provide robust solution for modelling and simulation of soft tissue responses during cutting in 3D. This algorithm belongs to the Element-Free Galerkin (EFG) family and is a generalisation of the 2D version presented in [11]. A cloud of points is used for discretisation and approximation of the deformation field within the continuum which eliminates the need for time-consuming generation of FE meshes and avoids the shortcomings of the distortion and fragmentation of finite elements. In the MTLADR 3D cutting algorithm, the cutting path is geometrically represented as a series of cutting planes. The discontinuities induced by surgical cutting are modelled and traced using nodes with specific level set values and appropriate field values (the nodal displacements and the size of influence domain). The update of the level set values is performed using simple and fast algebraic computations based on the geometry of the cutting path. The effect of cutting is entirely reflected in the changes of the shape and size of the influence domain of the nodes by efficiently implementing visibility criterion with the aid of level set values.

Using adaptive dynamic relaxation, MTLADR 3D cutting algorithm offers a fast convergence to the deformed state of deformation for problems with nonlinear geometric and material behaviour. The application of Total Lagrangian (TL) formulation eliminates the error accumulation due to the stress/strain update associated with the Updated Lagrangian (UL) formulation. The accuracy of the algorithm is well controlled by the stringent convergence criteria of dynamic relaxation. A range of numerical experiments were conducted for verification of the results.

This paper is organised as follows. The methods for modelling and simulation of surgical cutting in 3D are presented in Section 2. The numerical experiments and result verification are given in Section 3. Section 4 contains discussion and conclusions.

2 Methods

2.1 Governing Equation and Solution

The deformed state of soft tissue at any stage of cutting is solved using dynamic relaxation with optimal time stepping to obtain steady-state solution of the deformed continuum [12]. Details of the solution method are given in our previous papers [11, 13, 14].

2.2 Modelling of Surgical Cutting in 3D

In this study, we geometrically represent the progressing surgical cut in 3D as a series of cutting planes as shown in Fig. 1. We adapt the level set method proposed by Osher and Sethian [15] and developed by Stolarska et al. [16] to mathematically describe the signed distances of all the nodes and integration points relative to the cutting path. The discontinuities induced by surgical cutting are modelled and traced using pairs of nodes which are created by adding two new nodes or splitting the existing node at the point of discretisation. Every pair of discretising nodes on the cutting faces is assigned specific level set values in order to be allocated into the opposite sides of the cutting path. The appropriate field values (the nodal

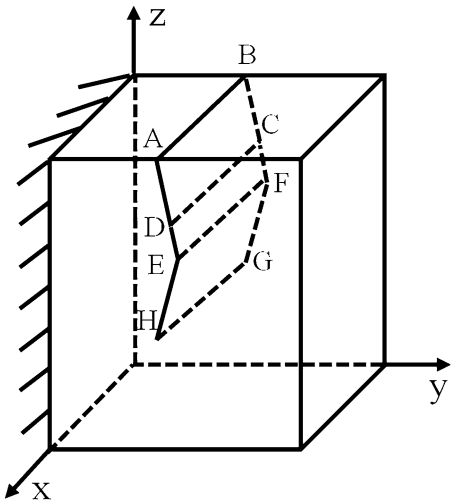

Fig. 1 The cutting path in 3D problem domain is represented by a series of cutting planes (ABCD, CDEF and EFGH)

displacement and the size of domain of influence) of the newly added node can either be interpolated by the existing nodal field parameters or be inherited from the existing node. The effect of cutting is entirely reflected in the changes of the shape and size of the influence domain of the nodes by implementing visibility criterion with the help of the level set values of the nodes and integration points.

2.2.1 Initialisation of the Level Set Values of the Nodes and Integration Points

When the first cut (e.g. the cutting plane ABCD illustrated in Fig. 1) is made, the cutting face and its extension plane are mathematically represented as the zero level set of function $\psi(x,y,z)$ as defined in (1). The bottom line of the cut is defined as the intersection of the cutting face and its orthogonal plane, which is mathematically represented by the zero level set of function $\phi(x,y,z)$ as defined in (2):

$$\psi(x,y,z) = \left(x - x_{\text{ep}}\right)\frac{K_x}{\|K\|} + \left(y - y_{\text{ep}}\right)\frac{K_y}{\|K\|} + \left(z - z_{\text{ep}}\right)\frac{K_z}{\|K\|} \tag{1}$$

$$\phi(x,y,z) = \left(x - x_{\text{ep}}\right)\frac{T_x}{\|T\|} + \left(y - y_{\text{ep}}\right)\frac{T_y}{\|T\|} + \left(z - z_{\text{ep}}\right)\frac{T_z}{\|T\|} \tag{2}$$

where (x,y,z) is the coordinate of a given point in the problem domain; $\left(x_{\text{ep}}, y_{\text{ep}}, z_{\text{ep}}\right)$ is the coordinate of an arbitrary point located at the end-line/endpoint of the cutting plane; K represents the normal vector of the cutting plane; K_x, K_y, and K_z are the x, y, z components of vector K respectively; $\|K\|$ is the length of vector K; T represents the normal vector of the orthogonal plane of the cutting face which points to the cutting direction; T_x, T_y, and T_z are the x, y, z components of vector T respectively; $\|T\|$ is the length of vector T.

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 face. As illustrated in Fig. 2, the points with positive values of the level set function ψ are all located on the same side of the cutting plane P and its extension plane P1, while the points with negative values are all on the other side. Zero value of function ψ indicates that the point is located right on the cutting plane P or its extension plane P1. Following the same principle, the value of the level set function ϕ indicates the signed distance from a given point to the plane P2 which is orthogonal to the cutting face at the end-line of cutting L. If a point is judged to be located on the cutting plane P (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 (Fig. 2).

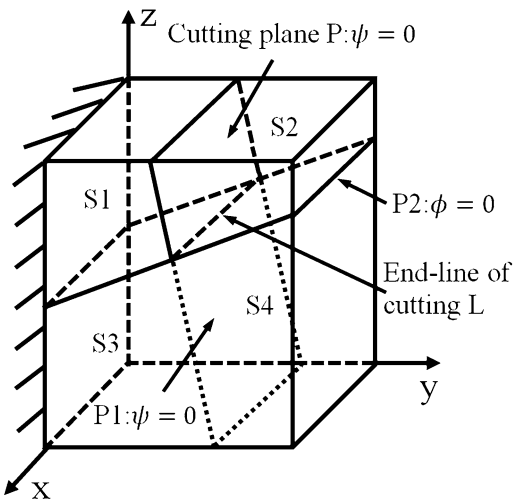

Fig. 2 The initialisation of level set values of functions ψ and ϕ. The 3D problem domain is divided into four subdomains: subdomain S1: $\psi < 0$ and $\phi < 0$; subdomain S2: $\psi > 0$ and $\phi < 0$; subdomain S3: $\psi < 0$ and $\phi > 0$; subdomain S4: $\psi > 0$ and $\phi > 0$. $\psi = 0$ on the cutting plane P and its extension plane P1. $\phi = 0$ on the orthogonal plane P2

2.2.2 Spatial Discretisation of the Cutting Planes

The cutting planes representing the cutting path are discretised using nodes spaced at the average nodal interval of the computational grid. If the computational grid obtained by discretising the geometry of the analysed continuum contains no node at the discretised position of the end-line of the cut, we add one node there with zero values of level set functions ψ and ϕ. If there is no node at the discretised position of the rest of the cutting planes, 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 planes, we split this node into two nodes having the same coordinates.

2.2.3 Update of Level Set Values of Nodes and Integration Points During Progressing Cutting

When cutting progresses from one cutting plane to the next, the level set values of functions ψ and ϕ at the nodes and the 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, z) are the coordinates of a given node or integration point; $(x_{\text{ep}}^n, y_{\text{ep}}^n, z_{\text{ep}}^n)$ and $(x_{\text{ep}}^{n+1}, y_{\text{ep}}^{n+1}, z_{\text{ep}}^{n+1})$ are the coordinates of an arbitrary point located at the end-line/endpoint of the cutting plane at step n and $n+1$ respectively; T^{n+1} denotes the normal vector of the orthogonal plane of the cutting face at step $n+1$ which points in the cutting direction; T_x^{n+1}, T_y^{n+1}, T_z^{n+1} are the x, y, z components of the normal vector T^{n+1}; $\|T^{n+1}\|$ is the length of the normal vector T^{n+1}; K^{n+1} denotes the normal vector of the cutting face at step $n+1$; K_x^{n+1}, K_y^{n+1}, K_z^{n+1} are the x, y, z components of the

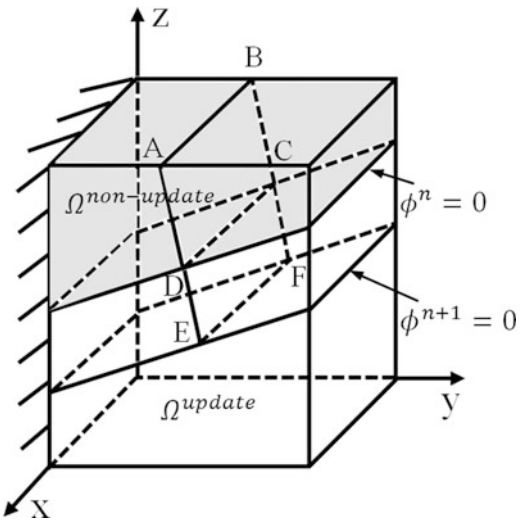

Fig. 3 Progression of cutting without direction change (cutting plane ABCD and CDEF are co-linear). The *shaded space*, where $\phi^n < 0$, is a non-update space $\Omega^{\text{non-update}}$ while the rest of the domain is an update space Ω^{update}

normal vector K^{n+1} respectively; $\|K^{n+1}\|$ is the length of the normal vector K^{n+1}; Ω represents the whole region of the problem domain.

1. Cutting direction does not change. When cutting progresses from one cutting plane 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 Fig. 3. No update is done in the space where $\phi^n < 0$ (indicated as non-update space $\Omega^{\text{non-update}}$ in Fig. 3) while the rest of the analysed domain (where $\phi^n \geq 0$) is an update space

$$\psi^{n+1} = \psi^n \text{ in } \Omega \tag{3}$$

$$\phi^{n+1} = \phi^n \text{ in } \Omega^{\text{non-update}} \tag{4}$$

$$\phi^{n+1}(x,y,z) = (x - x_{\text{ep}}^{n+1}) \frac{T_x^{n+1}}{\|T^{n+1}\|} + (y - y_{\text{ep}}^{n+1}) \frac{T_y^{n+1}}{\|T^{n+1}\|} + (z - z_{\text{ep}}^{n+1}) \frac{T_z^{n+1}}{\|T^{n+1}\|} \text{ in } \Omega^{\text{update}} \tag{5}$$

   where Ω is the union of $\Omega^{\text{non-update}}$ and Ω^{update} regions.
2. Cutting direction changes. If the cutting direction changes, the update region Ω_c^{update} is defined as the space where $\bar{\phi}^n > 0$ while the rest of the analysed domain is defined as the non-update region $\Omega_c^{\text{non-update}}$ (Fig. 4).
   (a) Level set function $\bar{\phi}^n$ is obtained by rotating function ϕ^n around the end-line/endpoints of the cutting plane at step n (e.g. end-line CD of cutting plane ABCD in Fig. 4) until orthogonal to the cutting plane at step $n+1$ (e.g. cutting plane CDEF in Fig. 4):

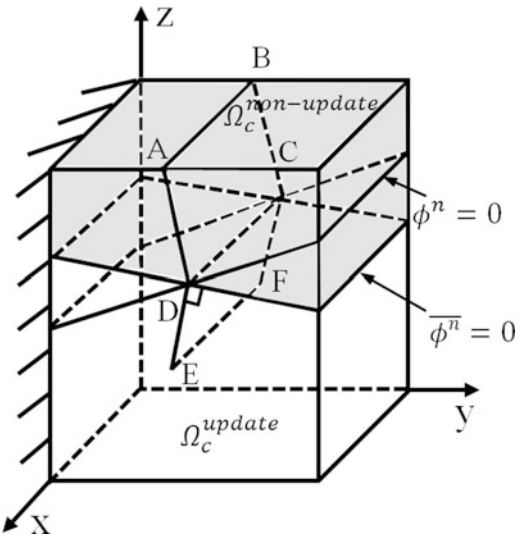

Fig. 4 Progression of cutting with direction change: the cutting proceeds from cutting plane ABCD to cutting plane CDEF with direction change. Level set function $\bar{\phi}^n$ is obtained by rotating level set function ϕ^n until it is orthogonal to the current cutting plane CDEF. The *shaded space*, where $\bar{\phi}^n \leq 0$, is non-update region $\Omega_c^{\text{non-update}}$ while the rest of the problem domain is an update region Ω_c^{update}

$$\bar{\phi}^n(x,y,z) = \left(x - x_{\text{ep}}^n\right)\frac{T_x^{n+1}}{\|T^{n+1}\|} + \left(y - y_{\text{ep}}^n\right)\frac{T_y^{n+1}}{\|T^{n+1}\|} + \left(z - z_{\text{ep}}^n\right)\frac{T_z^{n+1}}{\|T^{n+1}\|} \text{ in } \Omega \tag{6}$$

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

$$\psi^{n+1} = \psi^n \text{in } \Omega_c^{\text{non-update}} \tag{7}$$

$$\psi^{n+1}(x,y,z) = \left(x - x_{\text{ep}}^{n+1}\right)\frac{K_x^{n+1}}{\|K^{n+1}\|} + \left(y - y_{\text{ep}}^{n+1}\right)\frac{K_y^{n+1}}{\|K^{n+1}\|} + \left(z - z_{\text{ep}}^{n+1}\right)\frac{K_z^{n+1}}{\|K^{n+1}\|} \text{ in } \Omega_c^{\text{update}} \tag{8}$$

(c) Re-computation of function ϕ^{n+1} is conducted in the entire analysed space

$$\phi^{n+1}(x,y,z) = \left(x - x_{\text{ep}}^{n+1}\right)\frac{T_x^{n+1}}{\|T^{n+1}\|} + \left(y - y_{\text{ep}}^{n+1}\right)\frac{T_y^{n+1}}{\|T^{n+1}\|} + \left(z - z_{\text{ep}}^{n+1}\right)\frac{T_z^{n+1}}{\|T^{n+1}\|} \text{in } \Omega \tag{9}$$

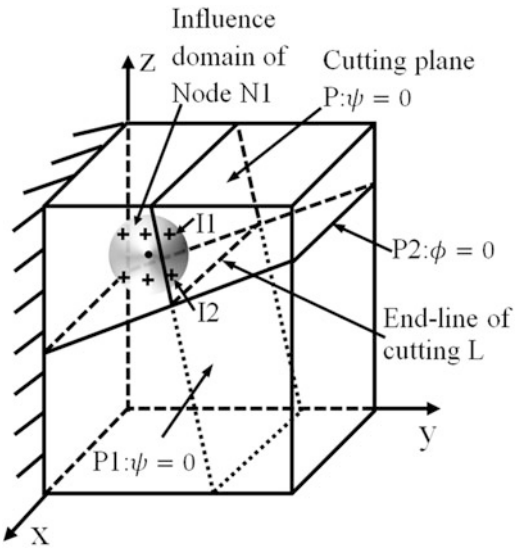

Fig. 5 The influence domain of node N1 intersects the cutting plane P; points I1 and I2 should be eliminated from the influence domain of this node

2.2.4 Update of Influence Domains During Progressing Cutting

The update of the influence domains of every node in the vicinity of the cutting faces 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_{pt} and ϕ_{pt} denote the values of level set functions ψ and ϕ at an integration point/node in the influence domain of node i.

1. The influence domain of a node, which is defined as a sphere in our algorithm, is updated if it intersects the cutting path (Fig. 5). The selection criterion for the node whose influence domain needs to be updated is

$$|\psi_i| \leq R_i \quad \text{AND} \quad \phi_i < R_i \tag{10}$$

   where R_i is the size (radius) of the influence domain of node i.
2. If the point and the node are located at the opposite sides of the cutting faces and the line segment linking the node and the point intersects the cutting plane (Fig. 5),

$$\psi_i \times \psi_{\text{pt}} < 0 \quad \text{AND} \quad \phi_{\text{pt}} < -\phi_i \times \left| \frac{\psi_{\text{pt}}}{\psi_i} \right| \tag{11}$$

   the point is removed from the influence domain of the node. Otherwise, the node remains to be the supporting node of the point for interpolation.

3. 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 3D mass matrix is updated.

3 Numerical Results and Algorithm Verification

3.1 Numerical Experiment

The large deformation preceding cutting is simulated by the elongation of a cube ($0.1 \times 0.1 \times 0.1\,\text{m}$). One face of the cube (at $y = 0\,\text{m}$) is rigidly constrained in x,y,z axes directions. The opposite face (at $y = 0.1\,\text{m}$) is elongated by 20% of the initial length along y axis while the displacements in x and z axis direction are constrained. The cube is spatially discretised into nodes to create the computational grid for meshless algorithm. The soft tissue-like material property of the specimen is modelled as soft and nearly incompressible continuum using Neo-Hookean material model (Young's modulus $= 3,000\,\text{Pa}$, Poisson's ratio $= 0.49$, mass density $= 1,000\,\text{kg/m}^3$) that approximates the mechanical response of brain tissue [2]. Cutting is conducted in the deformed soft tissue-like specimen via three sequential steps with arbitrary direction as shown in Fig. 6a, b, and c respectively. The strain energy, the reaction force at the rigidly constrained face of the specimen, and the nodal displacements are predicted at every stage of the simulation.

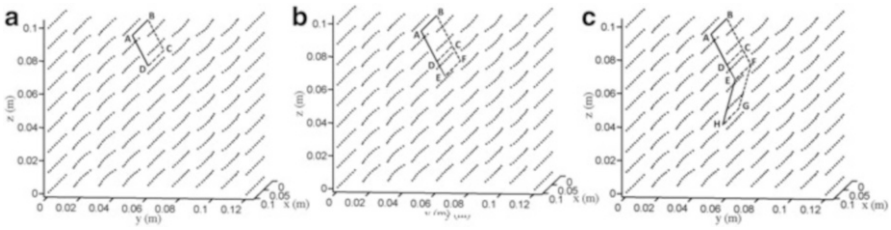

Fig. 6 Model for verification of the MTLADR 3D cutting algorithm (shown using 729 nodes for discretisation). Cutting is carried out in the stretched specimen of soft tissue-like material along the pre-defined path shown using thick line segments. (**a**) The first cutting plane ABCD: A(0.0953,0.05,0.095), B(0.0047,0.05,0.095), C(0.005,0.06,0.077) and D(0.095,0.06,0.077). Cutting starts with arbitrary angle; (**b**) The second cutting plane CDEF: E(0.095,0.065,0.068) and F(0.005,0.065,0.068). Cutting progresses without direction change; (**c**) The third cutting plane EFGH: G(0.0053,0.058,0.041) and H(0.095,0.058,0.041). Cutting progresses with direction change. Dimensions are in meters (m)

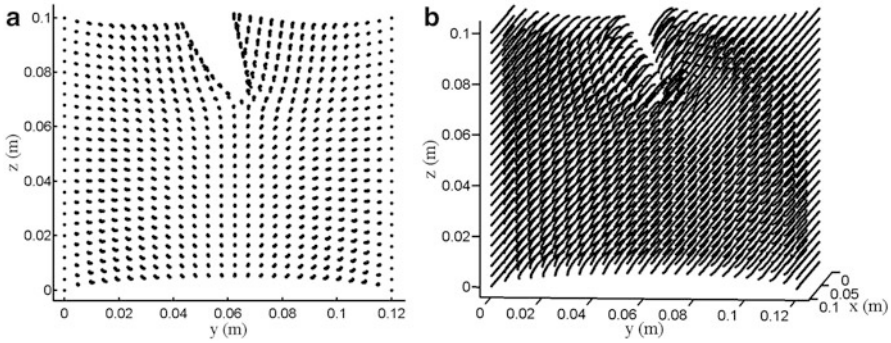

Fig. 7 The deformed model of the 3D specimen of soft tissue-like material before the change of cutting direction obtained using the MTLADR 3D cutting algorithm (converged solution shown using 17,576 nodes): (**a**) the projection of the deformed model on the YOZ plane; (**b**) 3D view of the deformed model

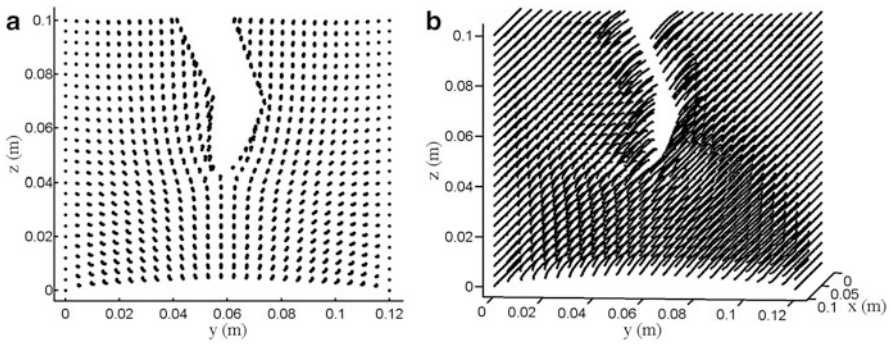

Fig. 8 The deformed model of the 3D specimen of soft tissue-like material at the end of 3-step arbitrarily angled cutting obtained using the MTLADR 3D cutting algorithm: (**a**) the projection of the deformed model on the YOZ plane; (**b**) 3D view of the deformed model

3.2 Numerical Results

Careful convergence studies were conducted. The deformed state of the specimen before the change of cutting direction (converged solution) is shown in Fig. 7. The deformed shape of the specimen at the end of cutting (converged solution), i.e. after the change in cutting direction is introduced, is shown in Fig. 8.

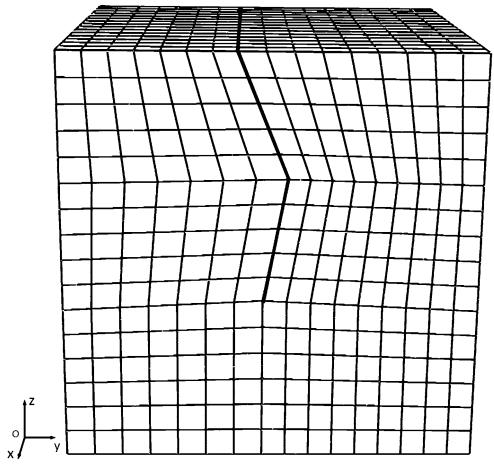

Fig. 9 The mesh of elements of the FE model analysed to obtain the reference solution in Abaqus (18,133 nodes). The edges of the elements were aligned and separated along the pre-cut face, which is shown as *thick line* segments, in order to assure the good aspect ratio of the elements

3.3 Reference Solution for Verification of MTLADR 3D Cutting Algorithm

The results of MTLADR 3D cutting algorithm were verified against the well-tested nonlinear FE analysis procedure available in commercial software Abaqus 6.10–1. As simulation of cutting is not supported by Abaqus, we computed only the deformed configuration of the specimen with pre-introduced discontinuity corresponding to the cutting path (Fig. 9).

3.4 Comparison of MTLADR 3D and Abaqus Solutions

The strain energy, reaction force and nodal displacements obtained using the MTLADR 3D cutting algorithm agree well with the reference solution obtained using the commercial FE solver Abaqus. The relative differences for strain energy and the reaction force between the two methods are 0.34% and 0.29% respectively. To enable verification of the predicted deformations, the nodal displacements obtained using the MTLADR 3D cutting algorithm were re-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 3D cutting algorithm and Abaqus is 0.78 mm (3.9% of the imposed elongation) and the average difference (averaging over all model nodes) is only 0.039 mm (0.19% of the imposed elongation). The absolute difference at most of the nodal positions (95.76%) is less than 0.1 mm (0.5% of the imposed elongation).

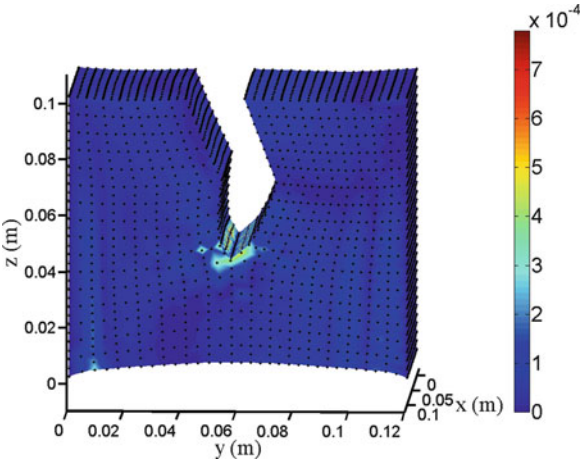

Fig. 10 Verification results for the proposed MTLADR 3D cutting algorithm. Distribution of the absolute differences between the nodal displacement magnitudes computed using the MTLADR 3D cutting algorithm and the reference results from the established nonlinear static solution procedures available in commercial finite element software Abaqus

As shown in the distribution of the absolute differences demonstrated in Fig. 10, the maximum differences are located at the vicinity of the endpoints of the cutting plane while the differences in the other areas are extremely small (well below 0.1 mm). It is probable that the maximum differences may be caused by the deterioration of the finite element solution accuracy caused by mesh distortion at large strains in the vicinity of the endpoints of the cutting plane rather than by our meshless algorithm. Considering that the accuracy of the state-of-the-art image-guided neurosurgery techniques is not better than 1 mm [17], the accuracy of the proposed MTLADR 3D cutting algorithm can be regarded as satisfying the requirements of computer-integrated surgery.

4 Discussion and Conclusions

We developed the MTLADR 3D cutting algorithm, which belongs to the EFG family, to predict the steady-state responses of soft tissue at any stage of surgical cutting in 3D. The algorithm is capable of modelling both large deformations and nonlinear material properties of soft tissues that are necessary in simulating surgery. Since the spatial discretisation is in a form of a cloud of nodes, the burdensome mesh generation and re-meshing in 3D required by the FE method are effectively alleviated. The application of TL formulation eliminates the accumulation of errors due to the reference frame update that occurs in UL formulation, which is typically utilised in commercial FE codes. The privilege of pre-computing the shape functions and their spatial derivatives greatly reduces the number of numerical operations as compared to UL formulation. Furthermore, the strategy of dynamic relaxation offers excellent performance in terms of computation speed while preserving good

accuracy by controlling errors using stringent convergence criteria. These features make our MTLADR method faster than FEM using tetrahedra.

The trajectory of cutting in 3D is geometrically represented as a series of cutting planes. The cutting-induced discontinuity is modelled by creating a pair of nodes at each discretised position of the cutting faces and introducing one node at each discretised position of the end-line of the cutting. The nodes of each nodal pair on the cutting faces are allocated to the opposite sides of the cutting path by using specific level set values. For explicit time integration, the field variable values at these nodes (i.e. the nodal displacements at the current and previous time step and the size of the influence domain) are either inherited or interpolated (using MLS shape functions) from the surrounding nodes. The effect of the cutting in the deformation of the continuum is modelled solely through the changes in the nodal domains of influence, which is effected through the efficient implementation of the visibility criterion using the level set method.

The accuracy of the proposed MTLADR 3D cutting algorithm is verified against the well-established nonlinear solution procedures available in the commercial finite element software Abaqus. Numerical experiments were carried out to predict the behaviour of a deformed (stretched) specimen of soft tissue-like material during surgical cutting in 3D. The computed strain energy, reaction force, and the nodal displacements agree very well with the reference solution obtained using the nonlinear static solution procedures in Abaqus.

The key factor determining the computational cost of the proposed MTLADR 3D cutting algorithm is the number of iterations associated with dynamic relaxation rather than modelling of cuts through the visibility criterion and level set method. We have already completed a prototype implementation of the dynamic relaxation part of our 3D cutting algorithm on graphics processing unit (GPU) (from within Matlab) and achieved more than two orders of magnitude improvement in computational speed in comparison with the Matlab implementation on CPU alone. We are confident that our approach is sufficiently computationally efficient to offer the prospect of surgical simulation within the time constraints of the operating theatre.

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

References

1. Fung, Y.C.: Biomechanics: Mechanical Properties of Living Tissues. Springer, New York (1993)
2. Miller, K., Chinzei, K., Orssengo, G., Bednarz, P.: Mechanical properties of brain tissue in-vivo: experiment and computer simulation. J. Biomech. **33**, 1369–1376 (2000)

3. Wittek, A., Dutta-Roy, T., Taylor, Z., Horton, A., Washio, T., Chinzei, K., Miller, K.: Subject-specific non-linear biomechanical model of needle insertion into brain. Comput. Meth. Biomech. Biomed. Eng. **11**, 135–146 (2008)
4. Mor, A.B.: Progressive Cutting with Minimal New Element Creation of Soft Tissue Models for Interactive Surgical Simulation. Carnegie Mellon University, Pittsburgh (2001)
5. Bruyns, C.D., Senger, S., Menon, A., Montgomery, K., Wildermuth, S., Boyle, R.: A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh-cutting using virtual tools. J. Visualization Comput. Anim. **13**, 21–42 (2002)
6. Bielser, D., Glardon, P., Teschner, M., Gross, M.: A state machine for real-time cutting of tetrahedral meshes. Graph. Model. **66**, 398–417 (2004)
7. Courtecuisse, H., Jung, H., Allard, J., Duriez, C.: GPU-based real-time soft tissue deformation with cutting and haptic feedback. Prog. Biophys. Mol. Biol. **103**, 159–168 (2010)
8. Belytschko, T., Tabbara, M.: Dynamic fracture using element-free Galerkin methods. Int. J. Numer. Meth. Eng. **39**, 923–938 (1996)
9. Cotin, S., Delingette, H., Ayache, N.: A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. Vis. Comput. **16**, 437–452 (2000)
10. Wu, W., Heng, P.A.: An improved scheme of an interactive finite element model for 3D soft-tissue cutting and deformation. Vis. Comput. **21**, 707–716 (2005)
11. Jin, X., Joldes, G.R., Miller, K., Yang, K.H., Wittek, A.: Meshless algorithm for soft tissue cutting in surgical simulation. Comput. Meth. Biomech. Biomed. Eng. 12 pages (in press) (2012)
12. Underwood, P.: Dynamic relaxation. In: Belytschko, T., Hughes, T.J.R. (eds.) Computational Methods for Transient Analysis, pp. 245–265. Elsevier Science Publishers B. V., Amsterdam, the Netherlands (1983)
13. Joldes, G.R., Wittek, A., Miller, K.: An adaptive dynamic relaxation method for solving nonlinear finite element problems. Application to brain shift estimation. Int. J. Numer. Meth. Biomed. Eng. **27**, 173–185 (2011)
14. Joldes, G.R., Wittek, A., Miller, K.: Computation of intra-operative brain shift using dynamic relaxation. Comput. Meth. Appl. Mech. Eng. **198**, 3313–3320 (2009)
15. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**, 12–49 (1988)
16. Stolarska, M., Chopp, D.L., Moes, N., Belytschko, T.: Modelling crack growth by level sets in the extended finite element method. Int. J. Numer. Meth. Eng. **51**, 943–960 (2001)
17. Bucholz, R., MacNeil, W., McDurmont, L.: The operating room of the future. Clin. Neurosurg. **51**, 228–237 (2004)