COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING
Commun. Numer. Meth. Engng 2007; **23**:121–134
Published online 14 August 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm.887

Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation

Karol Miller*, †, Grand Joldes, Dane Lance and Adam Wittek

Intelligent Systems for Medicine Laboratory, School of Mechanical Engineering, The University of Western Australia, 35 Stirling Highway, Crawley/Perth WA 6009, Australia

SUMMARY

We propose an efficient numerical algorithm for computing deformations of 'very' soft tissues (such as the brain, liver, kidney etc.), with applications to real-time surgical simulation. The algorithm is based on the finite element method using the total Lagrangian formulation, where stresses and strains are measured with respect to the original configuration. This choice allows for pre-computing of most spatial derivatives before the commencement of the time-stepping procedure.

We used explicit time integration that eliminated the need for iterative equation solving during the time-stepping procedure. The algorithm is capable of handling both geometric and material non-linearities. The total Lagrangian explicit dynamics (TLED) algorithm using eight-noded hexahedral under-integrated elements requires approximately 35% fewer floating-point operations per element, per time step than the updated Lagrangian explicit algorithm using the same elements.

Stability analysis of the algorithm suggests that due to much lower stiffness of very soft tissues than that of typical engineering materials, integration time steps a few orders of magnitude larger than what is typically used in engineering simulations are possible.

Numerical examples confirm the accuracy and efficiency of the proposed TLED algorithm.

Received 29 November 2005; Revised 7 May 2006; Accepted 12 May 2006

KEY WORDS: surgical simulation; soft tissues; finite element method; explicit time integration; total Lagrangian formulation

1. INTRODUCTION

There is wide international concern about the cost of meeting rising expectations for health care, particularly if large numbers of people require currently expensive procedures such as brain surgery.

*Correspondence to: Karol Miller, Intelligent Systems for Medicine Laboratory, School of Mechanical Engineering, The University of Western Australia, 35 Stirling Highway, Crawley/Perth WA 6009, Australia.
†E-mail: kmiller@mech.uwa.edu.au, http://www.mech.uwa.edu.au/ISML/

Contract/grant sponsor: Australian Research Council, Discovery Project; contract/grant numbers: DP0343112, DP0664534

Costs can be reduced by using improved machinery to help surgeons perform these procedures quickly and accurately, with minimal side effects. A novel partnership between surgeons and machines, made possible by advances in computing and engineering technology, could overcome many of the limitations of traditional surgery. By extending the surgeons' ability to plan and carry out surgical interventions more accurately and with less trauma, computer-integrated surgery (CIS) systems could help to improve clinical outcomes and the efficiency of health care delivery. CIS systems could have a similar impact on surgery to that long since realized in computer-integrated manufacturing (CIM).

Mathematical modelling and computer simulation have proved tremendously successful in engineering. Computational mechanics has enabled technological developments in virtually every area of our lives. One of the greatest challenges for mechanists is to extend the success of computational mechanics to fields outside traditional engineering, in particular to biology, biomedical sciences, and medicine [1].

The goal of surgical simulation research is to model and simulate deformable materials for applications requiring real-time interaction. Medical applications for this include simulation-based training, skills assessment and operation planning. A surgical simulator must predict the deformation field within the organ, so that it can be displayed to the user, and the internal forces (stresses), so that reaction forces acting on surgical tools can be computed and conveyed to the user through haptic feedback.

For computational efficiency reasons most researchers in the surgical simulation community use mathematical models based on linear elasticity that allow pre-computation of solutions (Green's functions). These models are incapable of providing realistic predictions of finite deformations of the tissue, because the deformations are assumed to be infinitesimal. Linearity of the material response is also assumed. Consequently, in such models the principle of superposition holds. However, years of experience accumulated by researchers working on the theory of elasticity and the finite element method have shown that the principle of superposition cannot be used for systems undergoing large deformations. In the 1970s, when non-linear finite element procedures were under development, many examples of inadequacy of linear theory were published, see e.g. References [2–4].

Zhuang [5], was probably the first to use updated Lagrangian explicit dynamics for real-time deformable object simulation. Picinbono *et al.* [6] used a similar algorithm, however their implementation of non-linear materials and viscoelasticity was, in our opinion, inferior to what is available within commercial packages, such as LS Dyna [7]. Papers [5, 6] used meshes built of linear tetrahedra, however they did not discuss problems related to volumetric locking. Also Szekely *et al.* [8] used explicit dynamics algorithms. Unfortunately, the details of their methods are not given in the publication.

The efficient finite element algorithm, accounting for both geometric and material non-linearities, that alleviates the problems mentioned above, is presented in the next section. In Section 3, we present three computational examples. Section 4 contains discussion and conclusions.

2. METHODS

2.1. Total Lagrangian formulation of the finite element method

Various spatial discretization schemes are possible while using the finite element method [9]. The great majority (if not all) of commercial finite element programs use the updated Lagrangian

DOI: 10.1002/cnm

formulation, where all variables are referred to the current (i.e. from the end of the previous time step) configuration of the system (Ansys [10], ABAQUS [11], ADINA [12], LS Dyna [13]). Therefore, Cauchy stress and Almansi (or logarithmic) strain are used. The advantage of this approach is the simplicity of incremental strain description. The disadvantage is that all derivatives with respect to spatial co-ordinates must be recomputed in each time step, because the reference configuration is changing. The reason for this choice is historical—at the time of solver development the memory was expensive and caused more problems than actual speed of computations.

The first key idea in the finite element algorithm development is to use the total Lagrangian formulation of the finite element method, where all variables are referred to the original configuration of the system. Second Piola–Kirchoff stress and Green strain are used. The disadvantage of this approach is the complicated description of finite strains resulting from the so-called initial displacement effect. However, the decisive advantage is that all derivatives with respect to spatial co-ordinates are calculated with respect to the original configuration and therefore can be pre-computed. Also, as rotation-invariant second Piola–Kirchoff stress is used, the necessity to rotate (Cauchy) incremental stresses before addition, present in updated Lagrangian formulation, is eliminated. Therefore, the proposed algorithm performs significantly fewer mathematical operations in each time step.

2.2. Explicit time integration of discretized equations of motion

The present study focuses on computing the deformation field and internal forces (stresses) within a soft organ during surgical procedure. This requires application of an efficient numerical scheme when integrating equations of equilibrium (or dynamics) in time domain. Such integration can be done using either implicit or explicit methods [14–16]. In implicit methods, the equations of dynamics are combined with the time integration operator, and the displacements are found directly. In explicit methods, on the other hand, at first the accelerations are determined from the equations of dynamics and then integrated to obtain the displacements.

The most commonly used implicit integration methods, such as the Newmarks' constant acceleration method, are unconditionally stable [9]. This implies that their time step is limited only by the convergence/accuracy considerations. However, the implicit methods require solution of a set of non-linear algebraic equations at each time step. Furthermore, iterations need to be performed for each time step of implicit integration to control the error and prevent divergence. Therefore, the number of numerical operations per each time step can be three orders of magnitude larger than for explicit integration [14]. Thus, 'the advantage of implicit method for three-dimensional (transient) problems becomes marginal' [9].

On the other hand, in explicit methods, such as the central difference method, treatment of non-linearities is very straightforward and no iterations are required. The global system of discretized equations of motion to be solved at each time step is

$$\mathbf{M}\ddot{\mathbf{u}}_n + \mathbf{K}(\mathbf{u}_n) \cdot \mathbf{u}_n = \mathbf{R}_n \tag{1}$$

where $\mathbf{u}$ is a vector of nodal displacements, $\mathbf{M}$ is a mass matrix, $\mathbf{K}$ is a stiffness matrix non-linearly dependent on the deformation (because geometrically non-linear procedure, suitable for computing large deformations, is used), and $\mathbf{R}$ is a vector of nodal (active) forces.

Using the central difference integration scheme we obtain:

$$u_{n+1} = u_n + \Delta t_{n+1}\dot{u}_n + 1/2\Delta t_{n+1}^2 \ddot{u}_n \tag{2}$$

DOI: 10.1002/cnm

$$\dot{u}_{n+1} = \dot{u}_n + 1/2\Delta t_{n+1}(\ddot{u}_{n+1} + \ddot{u}_n) \tag{3}$$

$$\mathbf{K}_n \cdot \mathbf{u}_n = \sum_i \mathbf{F}_n^{(i)} = \sum_i \int_{V^{(i)}} \mathbf{B}_n^{\mathrm{T}} \tilde{\mathbf{S}}_n \, \mathrm{d}V \tag{4}$$

and

$$\left(\frac{1}{\Delta t^2}\mathbf{M}\right)\mathbf{u}_{n+1} = \mathbf{R}_n - \sum_i \mathbf{F}_n^{(i)} - \frac{1}{\Delta t^2}\mathbf{M}(\mathbf{u}_{n-1} - 2\mathbf{u}_n) \tag{5}$$

where $\mathbf{B}_n^{\mathrm{T}}$ is the strain–displacement matrix at step n and $V^{(i)}$ is the ith element volume.

The non-linear properties of the tissue are accounted for in the constitutive model (see e.g. References [17–21] for the brain, References [22–24] for liver and kidney) and included in the calculation of nodal reaction forces $\mathbf{F}$. We used lumped (diagonalized) mass matrix $\mathbf{M}$ that multiplies the unknown $\mathbf{u}_{n+1}$. This rendered Equation (5) an explicit formula for the unknown $\mathbf{u}_{n+1}$. Equations (4) and (5) imply that computations are done at the element level eliminating the need for assembling the stiffness matrix $\mathbf{K}$ of the entire model. Thus, computational cost of each time step and internal memory requirements are substantially smaller for explicit than implicit integration. It is worth noting that there is no need for iterations anywhere in the algorithm. This feature makes the proposed algorithm suitable for real-time applications.

However, the explicit methods are only conditionally stable. Normally a severe restriction on the time step size has to be included in order to receive satisfactory simulation results. For example, in car crash simulations conducted with explicit solvers the time step is usually in the order of magnitude of microseconds or even tenths of microseconds [25, 26].

The critical time step is equal to the smallest characteristic length L_e of an element in the mesh divided by the dilatational wave speed c [7, 14, 27]:

$$\Delta t \leqslant \frac{L_e}{c} \tag{6}$$

For the eight-noded hexahedral element, the characteristic length is

$$L_e = \frac{V}{A_{\max}} \tag{7}$$

where $A_{\max}$ is the area of the element's largest face and V is the element's volume.

Stiffness of very soft tissue is very low [19]: e.g. stiffness of the brain is about eight orders of magnitude lower than that of steel. Since the maximum time step allowed for stability is (roughly speaking) inversely proportional to the square root of Young's modulus divided by the mass density, it is possible to conduct simulations of brain deformation with much longer time steps than in typical dynamic simulations in engineering.

2.3. Description of TLED finite element algorithm

Pre-computation stage:

1. Load mesh and boundary conditions.
2. For each element compute the determinant of the Jacobian det($\mathbf{J}$), spatial derivatives of shape functions $\partial\mathbf{h}$ and linear (constant) strain–displacement matrices, ${}^{t}_{0}\mathbf{B}_{L0}$ (notation of Reference [15] is used, where the left superscript represents the current time and the left subscript represents the time of the reference configuration—0 when total Lagrangian formulation is used).
3. Compute and diagonalize (constant) mass matrix ${}^{0}\mathbf{M}$.

DOI: 10.1002/cnm

Initialization:

1. Initialize nodal displacement ${}^0\mathbf{u} = \mathbf{0}$, ${}^{-\Delta t}\mathbf{u} = \mathbf{0}$, apply load for the first time step: forces or/and prescribed displacements: ${}^{\Delta t}R_i^{(k)} \longleftarrow R^{(k)}(\Delta t)$ or/and ${}^{\Delta t}u_i^{(k)} \longleftarrow d(\Delta t)$.

Time stepping:

Loop over elements:

1. Take element nodal displacements from the previous time step.
2. Compute deformation gradient ${}^t_0\mathbf{X}$.
3. Calculate full strain–displacement matrix:

$$ {}^t_0\mathbf{B}_L^{(k)} = {}^t_0\mathbf{B}_{L0}^{(k)}\, {}^t_0\mathbf{X}^{\mathrm{T}} \tag{8} $$

   This matrix accounts for initial displacement effect.
4. Compute second Piola–Kirchoff stress (vector) ${}^t_0\tilde{\mathbf{S}}$ at integration points.
5. Compute element nodal reaction forces

$$ {}^t\hat{\mathbf{F}}^{(m)} = \int_{0_V} {}^t_0\mathbf{B}_L^{\mathrm{T}}\, {}^t_0\tilde{\mathbf{S}}\, \mathrm{d}^0V \tag{9} $$

   using Gaussian quadrature.

Making a (time) step:

1. Obtain net nodal reaction forces at time t, ${}^t\mathbf{F}$.
2. Explicitly compute displacements using central difference formula

$$ {}^{t+\Delta t}u_i^{(k)} = \frac{\Delta t^2}{M_k}\,{}^t({}^tR_i - {}^tF_i^{(k)}) + 2^tu_i^{(k)} - {}^{t-\Delta t}u_i^{(k)} \tag{10} $$

   where M_k is a diagonal entry in kth row of the diagonalized mass matrix, R_i is an external nodal force, and Δ_t is the time step.
3. Apply load for the next step: ${}^{t+\Delta t}R_i^{(k)} \longleftarrow R^{(k)}(t+\Delta t)$ or/and ${}^{t+\Delta t}u_i^{(k)} \longleftarrow d(t+\Delta t)$.

As can be seen, there is no need for solution of coupled equations anywhere in the algorithm. Computationally most expensive parts of the algorithm are the evaluation of the full strain–displacement matrix (Point 3 of the loop over elements) and of second Piola–Kirchoff stresses (Point 4 of the loop over elements). These computations must be conducted at every integration point. That is why to achieve high computational efficiency under-integrated elements should be used. For 3D problems the most efficient elements are eight-noded hexahedra with a single integration point (for an excellent description of the implementation of this element see LS Dyna Manual [7]). For this element the integral (9) can be simply evaluated as

$$ {}^t\hat{\mathbf{F}}^{(m)} = 8\,{}^t_0\mathbf{B}_L^{\mathrm{T}}\, {}^t_0\tilde{\mathbf{S}}\det({}^0\mathbf{J}) \tag{11} $$

As, for stability reasons, rather small time steps are used, the strain–displacement matrix does not change much during a few time steps. Our experience shows that it is sufficient to update it every about 10 time steps. This leads to substantial computational efficiency improvement.

Accurate evaluation of stresses at integration points is essential for precise prediction of internal forces within the organ and computation of reaction forces acting on surgical tools. Therefore, appropriate constitutive models need to be applied. In case of hyperelastic models the second

DOI: 10.1002/cnm

Piola–Kirchoff stress can be evaluated as a derivative of the energy function with respect to Green strain. When hyper-viscoelastic models, such as the Ogden-based one developed for that brain in Reference [19], are selected, an efficient recursive scheme should be implemented. The implementation method is explained in detail in Reference [28].

3. RESULTS

3.1. Stability analysis

Stability analysis of explicit integration schemes is very well developed for sets of linear differential equations [29]. Most of the results are approximately valid for non-linear analysis as well. Nevertheless, we have conducted numerical experiments to ascertain the stability limits of the proposed algorithm.

We used 10 cm side cube meshed by 1000 eight-noded, single-integration-point hexahedra for the test problem, as it makes defining boundary conditions and generating hexahedral meshes simple and has similar dimensions to the human brain. No hourglass control was used. The cube is deformed by constraining one face, whilst the opposite face is displaced. The boundary conditions are: constrained face: $\Delta x = \Delta y = \Delta z = 0$; displaced face: $\Delta x = \Delta z = 0$, $\Delta y = d(t)$ (Figure 1). Mass density of $1000\,\mathrm{kg/m^3}$ and Neo-Hookean material model with Young's modulus in undeformed state equal to 3000 Pa and Poisson's ratio 0.49, which is approximately representative for the human brain, were used. The speed of sound in the bulk material is, therefore, approximately 7 m/s. For a mesh with 10 cubic elements per side, the critical time step, as predicted by a linear theory [29] is

$$\Delta t_{\mathrm{crit}} = \frac{L}{c} \approx 0.0013\,\mathrm{s}$$

To obtain experimental verification of this prediction we computed the displacement of all nodes with varying time steps. Above the critical time step the displacement increased dramatically as the algorithm lost stability, Figure 2. Figure 3 shows the axial force during axial elongation of the cube computed with the time step size very close to the critical time step. Oscillations in the force can be seen as the algorithm becomes unstable.

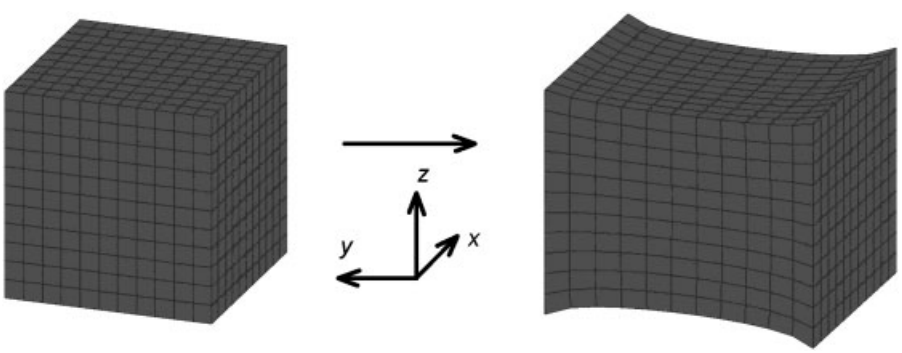

Figure 1. Constrained elongation of a cube.

DOI: 10.1002/cnm

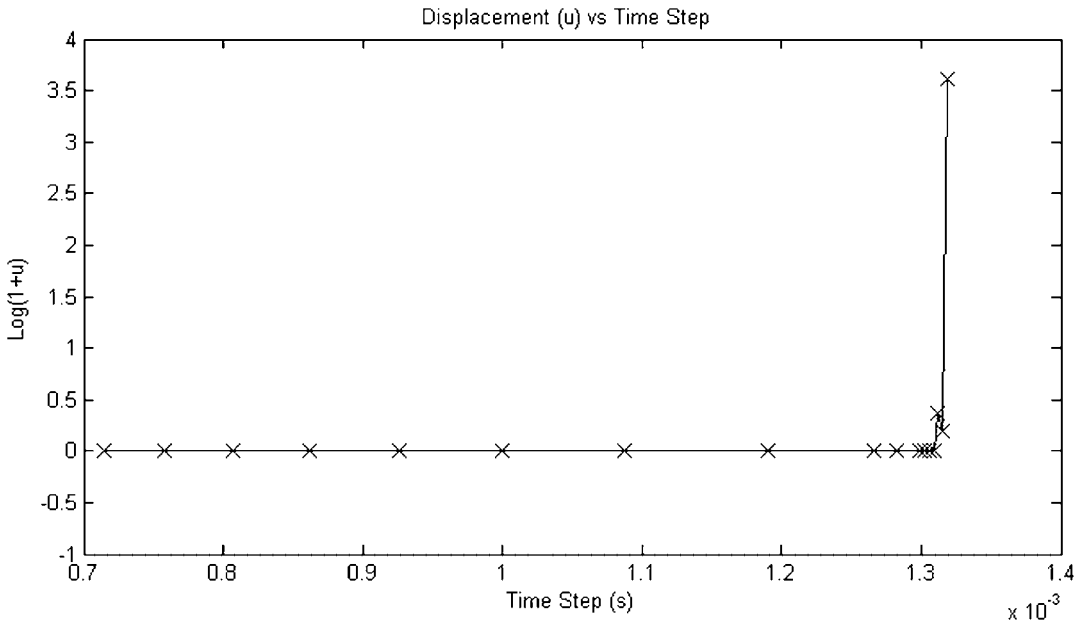

Figure 2. Logarithm of net nodal displacement with time step for 10 elements per side mesh.

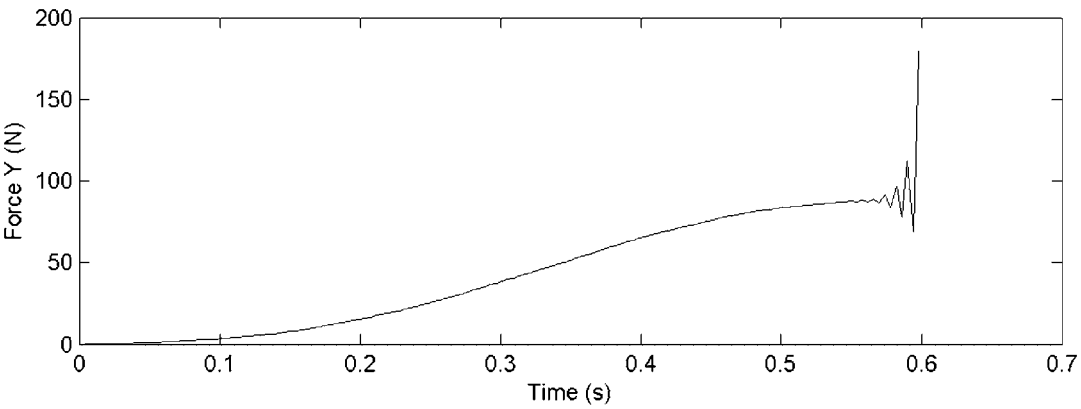

Figure 3. Force versus time showing the onset of oscillations due to time step nearing the critical time step.

Therefore, it should be ensured that the time step does not exceed the critical time step predicted by the linear theory.

3.2. Numerical examples

In all numerical examples hexahedral meshes with and without hourglass control were used. The hourglass control algorithm uses artificial stiffness to treat the zero energy modes of the

DOI: 10.1002/cnm

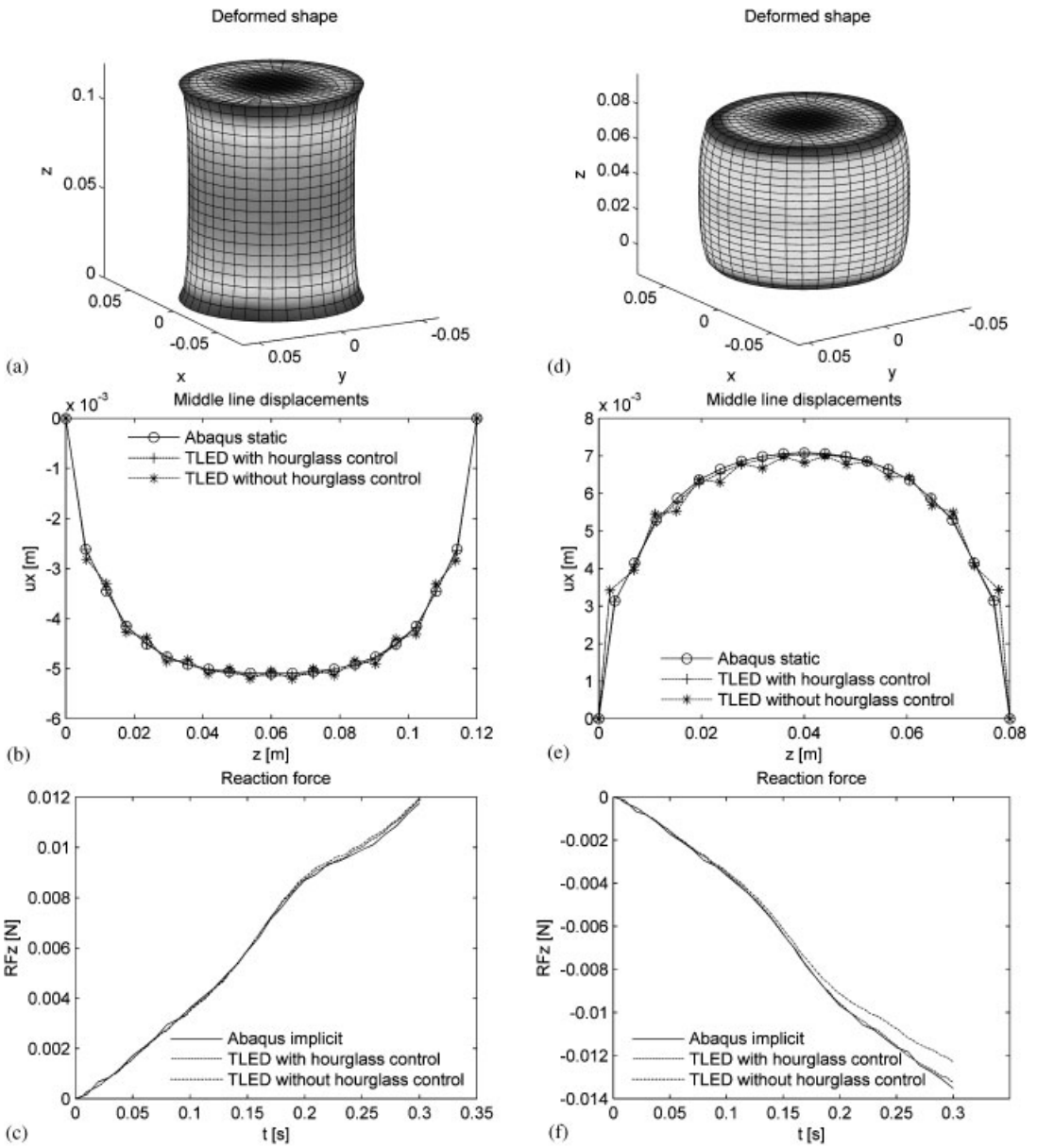

Figure 4. Extension and compression of a cylinder.

under-integrated hexahedral elements, as described in Reference [30]. Results obtained with the proposed algorithm programmed in C [31] were compared to the ones obtained with ABAQUS Standard Version 6.5-1 finite element analysis package [11].

The computed deformed shape of the object was compared with the one obtained using ABAQUS Static. The computed time evolution of the reaction force in one node was compared with the one

DOI: 10.1002/cnm

obtained using ABAQUS Dynamic/Implicit. In both cases default parameters of the ABAQUS solver were used. A Neo-Hookean material model with Young's modulus in undeformed state equal to 3000 Pa, Poisson's ratio 0.49 and mass density of 1000 kg/m^3 was considered in all cases. A fully integrated eight-noded linear brick, hybrid, constant pressure element was used in ABAQUS. The maximum displacement was 0.02 m in all cases and was applied using a loading curve given by

$$d(t) = (10 - 15t + 6t^2)t^3 \tag{12}$$

where t is the relative time (varying from 0 to 1).

3.2.1. Extension and compression of a cylinder. The cylinder is deformed by constraining one face, whilst the opposite face is displaced. The boundary conditions are: constrained face: $\Delta x = \Delta y = \Delta z = 0$; displaced face: $\Delta x = \Delta y = 0$, $\Delta z = d(t)$. The displacements are presented for a line of nodes found at the intersection of the plane $y = 0$ with the lateral surface of the cylinder. The reaction forces are presented for the node found in the middle of the displaced face, Figure 4. The results obtained with the proposed total Lagrangian explicit dynamics algorithm (TLED) are essentially the same as those obtained with ABAQUS indicating the appropriateness of the developed methods. It is also clear that hourglass control is necessary to maintain the accuracy. Therefore, in the remaining examples hourglass control was used.

3.2.2. Simple shear of a cube. The cube is deformed by constraining one face, whilst the opposite face is displaced. The boundary conditions are: constrained face: $\Delta x = \Delta y = \Delta z = 0$; displaced face: $\Delta y = \Delta z = 0$, $\Delta x = d(t)$. The displacements are presented for a line of nodes found at the intersection of the plane $y = 0.05$ with the lateral surface of the cube. They agree well with those computed with ABAQUS Static. The reaction forces are presented for the node found in the middle of the displaced face, Figure 5.

3.2.3. Indentation of an ellipsoid. The ellipsoid is deformed by constraining the lower half, whilst four nodes are displaced in the z-direction. This is a rough approximation of indentation of the brain [32]. The boundary conditions are: constrained nodes: $\Delta x = \Delta y = \Delta z = 0$; displaced nodes: $\Delta x = \Delta y = 0$, $\Delta z = d(t)$. The displacements are presented for a line of nodes found at the intersection of the plane $y = 0$ with the surface of the ellipsoid. The reaction forces are presented for one of the displaced nodes, Figure 6.

3.2.4. Influence of strain–displacement matrix update frequency. The strain–displacement matrix was updated at different number of time-step intervals for the ellipsoid indentation experiment and the obtained displacements, reaction forces and average time step were compared. The average time step was obtained by dividing the total computation time by the number of steps (1000 in this case). The hourglass control algorithm takes about 5.9 ms each step (using a standard 3.2 GHz Pentium 4 computer). The nodal displacements are presented for the node marked with a cross in the middle line displacements (Figure 7). The reaction forces are presented for one of the displaced nodes.

From these results it can be seen that the strain–displacement matrix can be updated at time step intervals greater that 10, but the improvement in computation time for a greater update interval is minimal and the accuracy of the results starts to decrease (Table I).

DOI: 10.1002/cnm

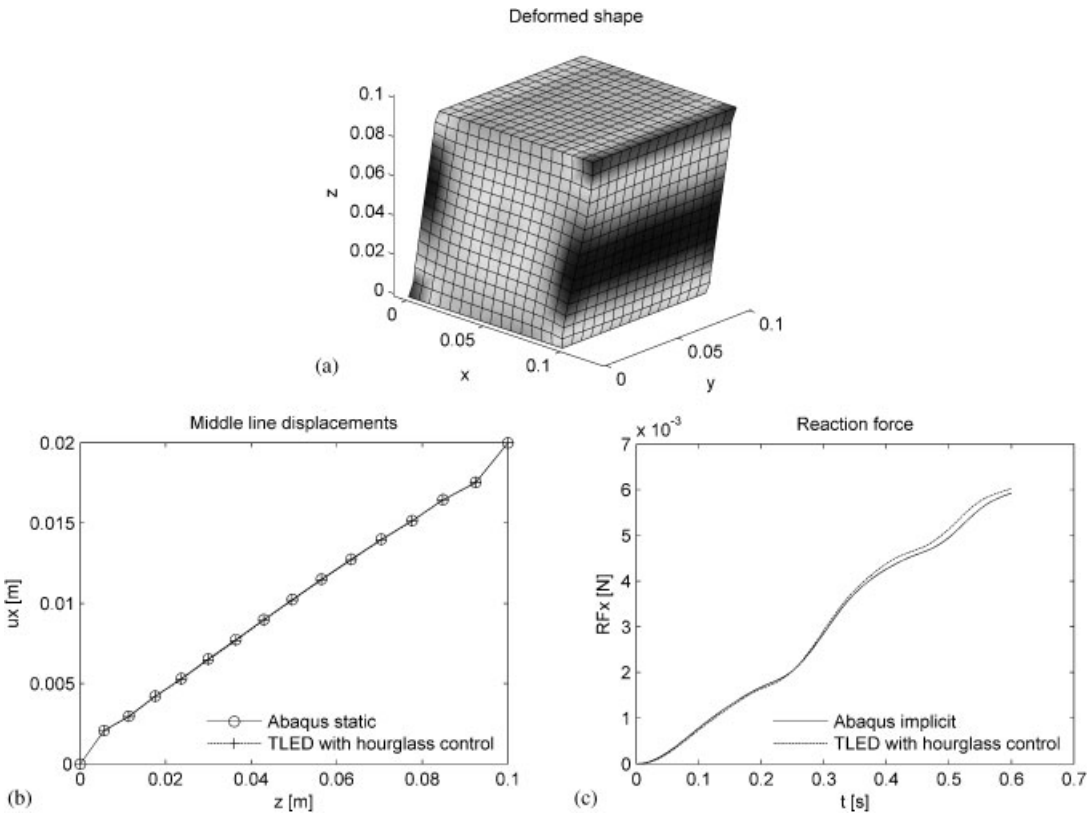

Figure 5. Simple shear of a cube.

Detailed comparison of floating point operation count per element, per time step shows that in time steps when the strain–displacement matrix is updated the number of floating point operations per element for our algorithm is 53 lower than in the explicit dynamics algorithm using updated Lagrangian formulation (as described in Reference [15]). In time steps when the strain–displacement matrix is not updated there are 773 fewer operations. As the strain–displacement matrix needs to be updated every about 10 time steps significant computational savings are evident. The average (over 10 time steps) number of floating-point operations per time step is approximately 35% lower in the proposed algorithm as compared to the one based on updated Lagrangian formulation.

4. DISCUSSION AND CONCLUSIONS

We presented an efficient algorithm based on the total Lagrangian formulation of the finite element method combined with explicit time integration. The TLED algorithm is capable of handling geometric and material non-linearities.

DOI: 10.1002/cnm

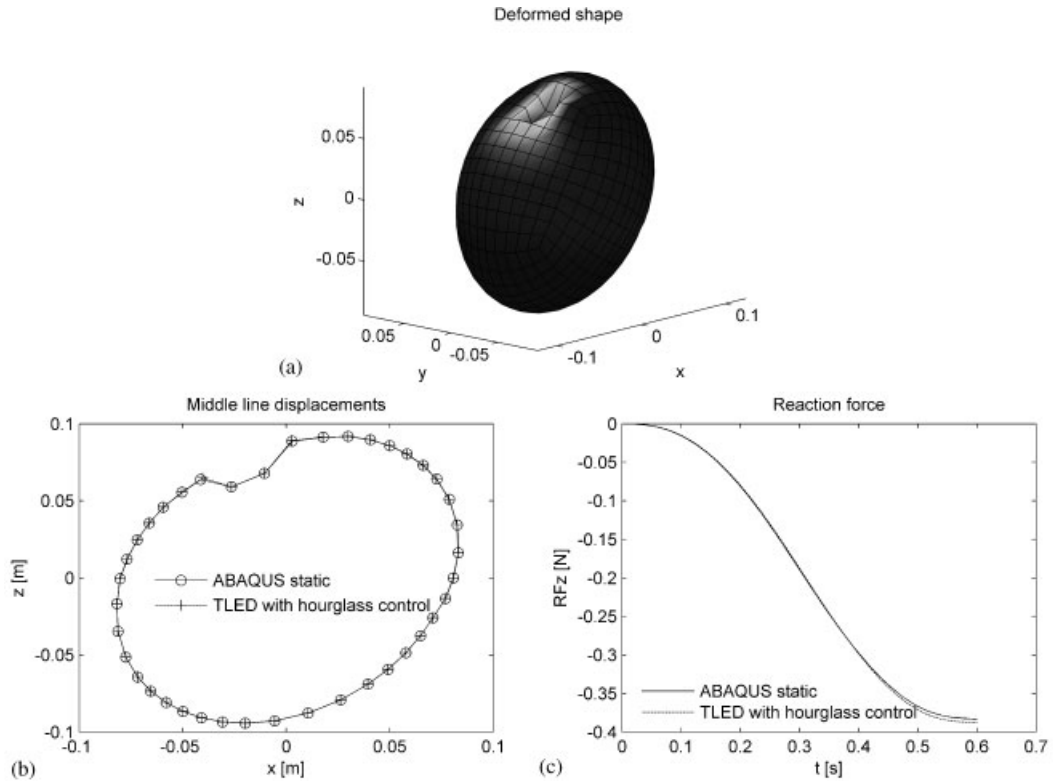

Figure 6. Indentation of an ellipsoid.

The algorithm's accuracy for large strain non-linear elastic behaviour was validated using a reputable commercial software ABAQUS [11]. The average number of floating-point operations per element per time step is 35% lower than for the similar implementation of the algorithm based on updated Lagrangian formulation.

Our implementation of the algorithm allows the computation of a step for a 6000-element 6741-node hexahedral mesh in about 16 ms on a standard 3.2 GHz Pentium 4 system using Windows XP operating system. As our results indicate that time steps as large as of 0.001 s are indeed possible, the algorithm constitutes a step towards a real-time surgical simulation. Additional advantageous feature of the explicit dynamics algorithm is that the computations are conducted at the element level and therefore the algorithm can be easily parallelized.

In Reference [33] we used the mesh of about 15 000 hexahedra to compute the craniotomy-induced brain shift for application in image-guided surgery. Regardless of improvements in calculation speed proposed in this paper, real-time computations of meshes of such size will require much faster computational hardware.

A number of challenges must be met before CIS based on computational biomechanical models can become as widely used as CIM systems are now. As we deal with individual patients, methods to produce patient-specific models quickly and reliably must be improved. Substantial progress in

DOI: 10.1002/cnm

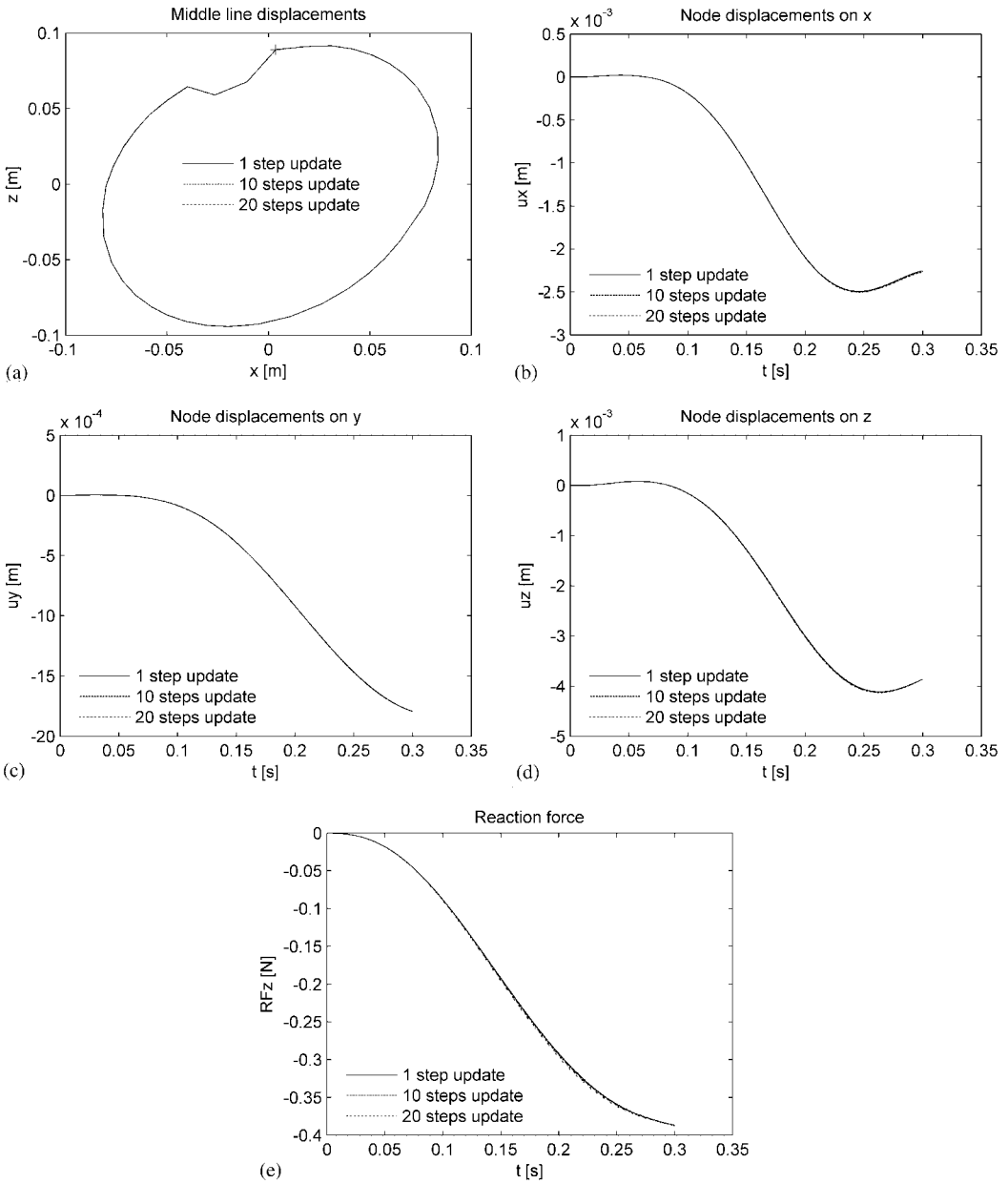

Figure 7. Influence of strain–displacement matrix update frequency.

automatic meshing methods is required. Implementation of algorithms for parallel computations on networks of processors, and harnessing the computational power of graphics processing units provide a challenge for coming years.

DOI: 10.1002/cnm

Table I. Influence of strain–displacement matrix update frequency on computation time.

Update frequency (steps)	Average step time (ms)
1	22.6
5	16.5
10	15.7
20	15.4

ACKNOWLEDGEMENTS

The financial support of the Australian Research Council (Grant Nos. DP0343112, DP0664534) is gratefully acknowledged. The second author was an Endeavor IPRS Scholar in Australia during the completion of this research. The first author would like to thank Professors Ivo Babuska, Graham Carey, Thomas Hughes and Leszek Demkowicz for very helpful discussion while visiting the University of Texas at Austin.

REFERENCES

1. Oden JT, Belytschko T, Babuska I, Hughes TJR. Research directions in computational mechanics. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:913–922.
2. Carey GF. A unified approach to three finite element theories for geometric nonlinearity. *Journal of Computer Methods in Applied Mechanics and Engineering* 1974; **4**(1):69–79.
3. Martin HC, Carey GF. *Introduction to Finite Element Analysis*: *Theory and Application*. McGraw-Hill Book Co: New York, 1973.
4. Oden JT, Carey GF. *Finite Elements*: *Special Problems in Solid Mechanics*. Prentice-Hall: Englewood Cliffs, NJ, 1983.
5. Zhuang Y. Real-time simulation of physically realistic global deformations. *Ph.D. Thesis*, Department of Computer Science, University of California, Berkley, 2000.
6. Picinbono G, Delingette H, Ayache N. Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models* 2003; **65**:305–321.
7. Hallquist JO. *LS-DYNA Theoretical Manual*. Livermore Software Technology Corporation: Livermore, California, 1998.
8. Szekely G, Brechbuhler C, Hutter R, Rhomberg A, Ironmonger N, Schmid P. Modelling of soft tissue deformation for laparoscopic surgery simulation. *Medical Image Analysis* 2000; **4**:57–66.
9. Belytschko T. An overview of semidiscretization and time integration procedures. In *Computational Methods for Transient Analysis*, Belytschko T, Hughes TJR (eds). North-Holland: Amsterdam, 1983; 1–66.
10. http://www.ansys.com/services/documentation/manuals.htm, ANSYS documentation. Accessed: 18/10/2005.
11. ABAQUS. *ABAQUS Online Documentation*: *Version 6.5-1*. Hibbitt, Karlsson & Sorensen, Inc., 2004, http://www.abaqus.com/products/products_documentation.html
12. www.adina.com, ADINA home page. Accessed: 22/11/2005.
13. LS-DYNA. *Keyword User's Manual. Version 970*. Livermore Software Technology Corporation: Livermore, California, 2003.
14. Belytschko T. A survey of numerical methods and computer programs for dynamic structural analysis. *Nuclear Engineering and Design* 1976; **37**:23–34.
15. Bathe K-J. *Finite Element Procedures*. Prentice-Hall: Englewood-Cliffs, NJ, 1996.
16. Crisfield MA. Non-linear dynamics. In *Non-linear Finite Element Analysis of Solids and Structures*. Wiley: Chichester, 1998; 447–489.
17. Miller K, Chinzei K. Constitutive modeling of brain tissue; experiment and theory. *Journal of Biomechanics* 1997; **30**(11/12):1115–1121.
18. Miller K. Constitutive model of brain tissue suitable for finite element analysis of surgical procedures. *Journal of Biomechanics* 1999; **32**:531–537.
19. Miller K, Chinzei K. Mechanical properties of brain tissue in tension. *Journal of Biomechanics* 2002; **35**:483–490.

DOI: 10.1002/cnm

20. Bilston L, Liu Z, Phan-Tiem N. Large strain behaviour of brain tissue in shear: some experimental data and differential constitutive model. *Biorheology* 2001; **38**:335–345.
21. Prange MT, Margulies SS. Regional, directional, and age-dependent properties of the brain undergoing large deformation. *Journal of Biomechanical Engineering* (ASME) 2002; **124**:244–252.
22. Liu ZZ, Bilston LE. Large deformation shear properties of liver tissue. *Biorheology* 2002; **39**(6):735–742.
23. Farshad M, Barbezat M, Fleler P, Schmidlin F, Graber P, Niederer P. Material characterization of the pig kidney in relation with the biomechanical analysis of renal trauma. *Journal of Biomechanics* 1999; **32**(4):417–425.
24. Miller K. Constitutive modelling of abdominal organs. *Journal of Biomechanics* 2000; **33**:367–373.
25. Brewer JC. Effects of angles and offsets in crash simulations of automobiles with light trucks. In *17th Conference on Enhanced Safety of Vehicles*, 2001, Amsterdam, Netherlands.
26. Kirkpatrick SW, Schroeder M, Simons JW. Evaluation of passenger rail vehicle crashworthiness. *International Journal of Crashworthiness* 2001; **6**(1):95–106.
27. Cook RD, Malkus DS, Plesha ME. Finite elements in dynamics and vibrations. In *Concepts and Applications of Finite Element Analysis*. Wiley: New York, 1989; 367–428.
28. Feng WW. A recurrence formula for viscoelastic constitutive equations. *International Journal of Non-Linear Mechanics* 1992: **27**(4):675–678.
29. Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1987.
30. Flanagan DP, Belytschko T. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering* 1981; **17**:679–706.
31. http://msdn.microsoft.com/netframework, Visual Studio. NET Combined Collection, Accessed: 22/11/2005.
32. Miller K, Chinzei K, Orssengo G, Bednarz P. Mechanical properties of brain tissue in-vivo: experiment and computer simulation. *Journal of Biomechanics* 2000; **33**:1369–1376.
33. Wittek A, Miller K, Kikinis R, Warfield SK. Patient-specific model of brain deformation: application to medical image registration. *Journal of Biomechanics* 2006 (accepted in February 2006).

DOI: 10.1002/cnm