Realistic And Efficient Brain-Skull Interaction Model For Brain Shift Computation

Grand Roman Joldes, Adam Wittek, Karol Miller, Leith Morriss

Intelligent Systems for Medicine Lab.
The University of Western Australia,
35 Stirling Highway
Crawley, WA 6009, AUSTRALIA
grandj@mech.uwa.edu.au

Abstract

In this paper we propose a very efficient contact implementation for modeling the brain-skull interaction. This contact algorithm is specially designed for our Dynamic Relaxation solution method for solving soft-tissue registration problems. It makes possible the use of complex biomechanical models which include different nonlinear materials, large deformations and contacts for image registration. The computational examples prove the accuracy and the computational efficiency of our methods. For a model having more than 50000 degrees of freedom, a complete simulation can be done in less than a minute on a standard personal computer.

Keywords: real time computations, brain-skull contacts, image registration

1 Introduction

Brain deformation during surgery – commonly known as brain shift - is the primary motivation for this study. Deformations within the brain due to brain shift are difficult to monitor in real time as high resolution intra-operative MRI still remains a research rather than a clinical tool. These unknown changes in the location and shape of the brain and associated anatomy present the neurosurgeon with challenges and barriers to safe successful surgery. The "accurate localization of target" has been listed as the first principle in modern neurosurgical procedures [1] and this project aims to make accurate localization of targets more achievable.

Surgery related brain deformations occur for a number of reasons – loss of fluid during a craniotomy, brain edema or physiologic changes [2, 3]. Deformations of up to 10 mm are common in nearly all neurosurgical cranial procedures [4] and can be up to 25 mm in some cases [5]. These deformations make surgery difficult as the neurosurgeon is usually unable to track them using high quality intra-operative medical images. The surgeon may see that the surface of the brain collapsed by 10 mm, but they will not be able to predict the deformation within the brain due to this collapse.

The resolution of intra-operative images is much lower than the one of pre-operative images, thus registration of the accurate pre-operative images to the intra-operative state is required for a complete and accurate visualization. A registration method that leads to physically plausible deformation estimates is the computation of the intra-operative brain deformations using a biomechanical model. Such a method treats brain shift as a solid mechanics problem.

The context of neurosurgery provides a number of constraints for a useful computation of brain deformation. Predominately the two most important constraints are short computation time and high accuracy. The computation time must be very short, so that updates to the model – from intra-operative measuring and imaging – can be immediately shown to the surgeon.

If only partial information about the brain surface can be obtained intra-operatively (i.e. only in the area of craniotomy), the deformation problem can not be solved accurately without considering the interaction between the brain and the skull for the remaining of the surface.

This paper is organized as follows: the problem of brain-skull interaction is analyzed in the next section, the resulting contact algorithm implementation is presented in Section 3, simulation results are presented in Section 4 and the last section contains some discussions and conclusions.

2 Problem Formulation

2.1 Registration As A Solid Mechanics Problem

The process of matching images of the same anatomy in differing modalities or resolutions is termed registration [6]. When the anatomy imaged is rigid (e.g. skeletal structure) only rigid registration is required, which is a simple process of mapping points between two coordinate systems. When the anatomy deforms – as is the case for the brain – more advanced non-rigid registration techniques are required.

Non-rigid registration is required for image-guided surgical procedures, where high resolution pre-operative images are warped to the configuration of lower quality intra-operative images. This has traditionally been achieved through applying image distortion or transformation algorithms to warp images ([7-9]). These methods work well when differences between images are not too large, however the plausibility of the solution can not be guaranteed with purely image based warping. When registering the finite deformations it is instead suggested to consider the registration process as a solid mechanics problem, to produce a solution based on the established principles of continuum mechanics.

The use of biomechanical models was proposed by many researchers. When appropriate nonlinear models and solution methods are used, good registration results are obtained even in case of finite deformations [10-12].

2.2   Interaction Modeling For The Brain-Skull Interface

There are three membranes: dura mater, the arachnoid and pia mater between the brain and skull. The subarachnoid space (SAS) contains cerebrospinal fluid (CSF). This complex structure is presented in Fig. 1 (edited from [13]). During craniotomy CSF can leak freely from the subarachnoid space, creating a gap between the brain and the skull [12].

As the Young's modulus of the skull bone is several orders of magnitude greater than that of the brain tissue we can treat the skull as a rigid body. Therefore it is sufficient to model the brain-skull interaction as a contact between a deformable continuum (the brain) and a rigid body (the skull).

Some authors have tried to model the brain-skull interaction as a sliding contact with no separation, in which the nodes on the brain surface can move only tangentially to the skull surface [14]. In such case the brain can not move towards the skull or separate from it. Considering the anatomical structure of the brain-skull interface and based on comparisons between pre-operative and intra-operative MRI images, we consider this is not an appropriate approach.

Other authors have applied displacements over the entire surface of the brain, to match the deformation of the surface to the intra-operative images [15, 16]. Although this is a realistic approach from the modelling point of view, the problem of obtaining the displacements of the entire brain surface intra-operatively remains.

The following assumptions are made in order to simplify the contact problem:
- If the skull is considered rigid and fixed, then deformation of this body is irrelevant. Only consideration of brain deformation is required.
- As lubrication is present, friction is low and sliding of the brain on the skull occurs – frictionless contact conditions are the simplest representation of sliding contact.
- Separation of brain from skull is allowed.
- Only the deformation of the brain is of interest for registration purposes – thus the contact force is not specifically of interest. The brain deformation is formulated as a "displacement – zero traction problem", as only displacement constrains are prescribed and no surface tractions are applied. This leads to a smaller influence of the material constitutive model on the simulation results [17].
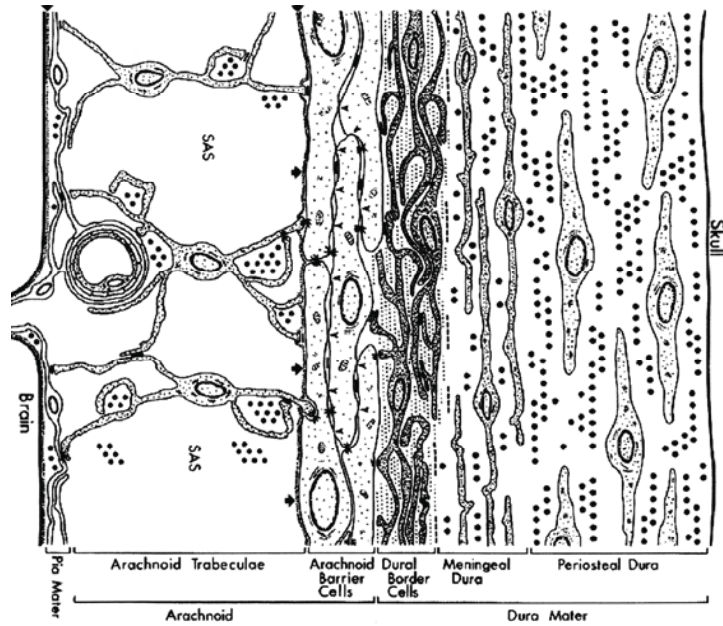
Fig. 1. The structure of the brain-skull interface, adapted from [13]

When selecting the best contact formulation we must also consider the solution method used for solving the finite element problem. We use Dynamic Relaxation [18] for finding the deformed state of our biomechanical model. This is an explicit method in which the position of the brain nodes is updated at every time step.

The simplest contact formulation for the brain-skull interaction, that accounts for the points discussed above, would be a finite sliding, frictionless contact between a deformable object (the brain) and a rigid surface (the skull). This can be implemented as a kinematic constraint type of contact that does not require the computation of any contact forces at the interface. A similar approach was proposed in [19], but no details are given regarding the contact algorithm and the simulations are performed using a commercial software (Abaqus).

There are many interaction (contact) handling algorithms available in commercial software, but there are some problems in using them: a large number of parameters (that influence the contact behaviour and the accuracy of the results) and long computation time. The contact algorithm we present has no configuration parameters (does not require the computing of contact forces) and is very fast, with the speed almost independent of the mesh density for the skull surface.

The main parts of the contact algorithm are: detection of nodes on the brain surface (also called the slave surface) which have penetrated the skull surface (master surface) and the displacement of each slave node that has penetrated the master surface to the closest point on the master surface.

# 3 Contact Algorithm Implementation

3.1 Detecting Penetration

The surfaces of the anatomical structures of segmented brain images are typically discretized using triangles; therefore we consider the skull surface as a triangular mesh. We will call each triangle surface a "face", the vertices - "nodes" and the triangle sides - "edges".

We base our penetration detection algorithm on the closest master node (nearest neighbor) approach [20]. The basic algorithm is as follows:

- For each slave node P:
  - Find the closest master node C (global search)
  - Check the faces and edges surrounding C for penetration (local search)

To improve the computation speed, the global search phase is usually implemented using bucket sort [20]. A good description of this searching algorithm is given in [21]. In our implementation the size of the buckets used for the global search is different in the three directions. For each direction, this size is given by half of the maximum size of all master edge projections on that direction. This ensures that the number of nodes in each bucket is minimal while there are no buckets for which a closest node can not be found.

The next step (local search), for a slave node P, aims at finding the closest node R on the master surface, on the faces or edges surrounding node C (Fig. 2). Once the closest point on the master surface is identified, the penetration is detected by checking the sign of the scalar product RP·**n**, with **n** the inside normal to the master surface in R. For an edge or a node the normal is defined as the sum of the normal vectors of adjacent faces.
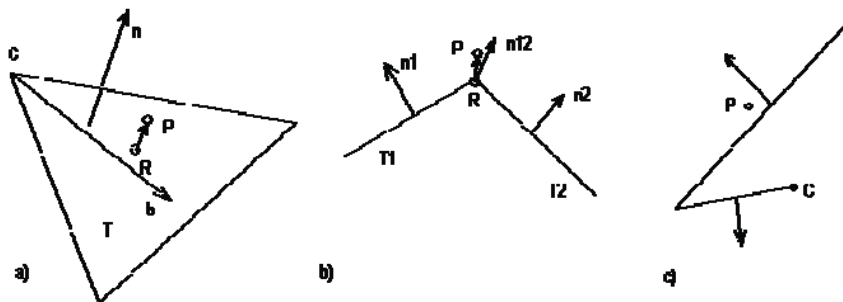
Fig. 2. Local search a) Penetration of a face b) Penetration of an edge c) Penetration of a face that is not connected to the closest node

Consider a triangular face T that contains node C, and the projection R of slave node P on the face (Fig. 2.a). If R is outside the triangle T, the face is discarded, otherwise the distance to the face is [PR]. In order to improve the speed of the search, only the faces for which CP·**b** > 0 are checked, with **b** being the bisector of angle C in triangle T.

When P projects outside triangle T, it can project on one of the adjacent triangles or it can project on the common edge between two adjacent triangles, as shown in Fig. 2.b (seen along the common edge). Therefore all the edges containing the closest master node C must also be checked.

Another possibility is that the node does not project inside any of the edges either and the closest node itself is the closest point on the master surface.

In most of the cases, the basic tests presented above are sufficient for identifying the closest point on the master surface. Nevertheless, there are also special cases that must be considered, when the closest point on the master surface is not on the faces and edges adjacent to C. A simple case is presented in Fig. 2.c for a two-dimensional situation. In a tri-dimensional setting the situation is more complex and such cases are more likely to occur even without having such sharp corners.

In commercial software this problem is solved by searching for the closest face or edge on the master surface instead of searching for the closest master node [20]. This search is time consuming even if bucket sort is used. Therefore our proposal for handling these special cases is to make an analysis of the master surface and identify, for each node C, all the faces and edges that are possible to be penetrated by a slave node P in the case C is the closest master node to P. This analysis is done based on geometrical considerations as explained in the next section. The identified faces and edges are kept in a list for each master node C and they are checked in addition to the faces and edges that contain C when the local search is performed.

In some cases the slave node P is too far from the closest master node C to penetrate any face or edge that contains C. If d is the maximum penetration possible in any given time step and

$$CP^2 > d^2 + r^2 \tag{1}$$

then the basic tests are skipped and only the additional tests are done. In the above relation r is the radius of influence of node C, being equal with the maximum length of all master surface edges containing C.

3.2   Finding additional edges and faces that must be checked

Consider an edge AB and a node on the master surface C (Fig. 3.a). We must check if it is possible for a slave node to be closer to C than to A or B but to have penetrated AB. In triangle ABC, the location of nodes that are closer to C than to A and B (R) is delimited by the lines OP and ON, where O is the center of the circumscribed circle and P and N are the middle of edges AC and BC. In space, R is delimited by two planes perpendicular on ABC and containing OP and ON respectively. The following tests are made for edge any edge AB that does not contain C and is not part of the same master triangle as C:

- If [CM] < [AM] or ([CM] > [AM] and [OM] < d) then AB is added to the list for node C. These conditions are equivalent to the edge AB crossing or being very close (less than d) to R.

For a node C and a face $T_1T_2T_3$ on the master surface, the location of nodes that are closer to C than to T_1, T_2 or T_3 (R) is delimited by 3 planes P_i which are perpendicular at the midpoint M_i to segments CT_i ($i = 1,2,3$). These planes all contain point O which is the center of the sphere circumscribed to the tetrahedron $CT_1T_2T_3$. G_1, G_2, G_3 and R are the projections of O to the faces of this tetrahedron (Fig. 3.b).

If n is the normal to the face pointing in the direction of C, we build the points D_1, D_2 and D_3 by displacing T_1, T_2 and T_3 in the direction of n by distance d. We name E_1, E_2 and E_3 the middle of the edges of the triangle $D_1D_2D_3$ and with O_1 the center of the circumscribed circle for the same triangle. The following tests are made for each master triangle $T_1T_2T_3$ which does not have C as a node:

- It is easy to show that if $[CR] > 2*[T_1R]$ then $\boldsymbol{R}$ can not intersect the interior of the triangle $T_1T_2T_3$, and therefore the face is discarded
- Consider the set of points $\boldsymbol{S} = \{\ D_1, D_2, D_3, E_1, E_2, E_3, O_1\}$. O_1 is included in $\boldsymbol{S}$ only if it is in the interior of triangle $D_1D_2D_3$. If any of these points is on the same side of all three planes $\boldsymbol{P}_i$ as C then the face is added to the list of additional faces to check for node C. From the geometry, these tests are equivalent to:

$$M_iS \cdot M_iC > 0, \quad i = 1,2,3 \qquad\qquad (2)$$

with S being any point from $\boldsymbol{S}$.
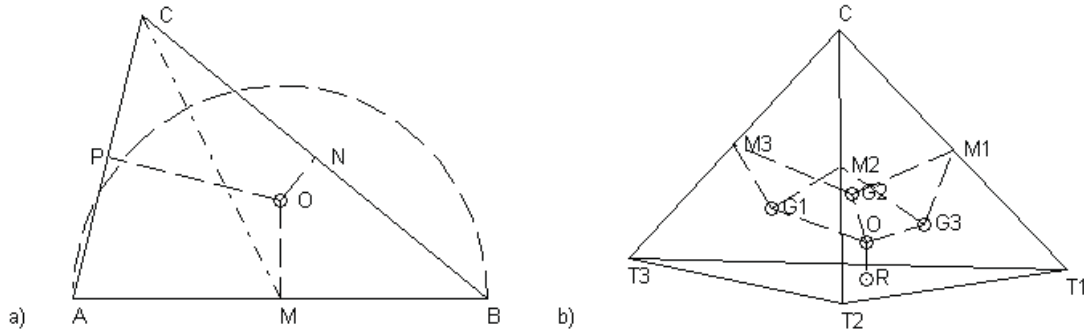
Fig. 3. Detection of additional edges (a) and triangles (b) to check for node C

When the relation between nodes and edges or faces from the master surface is studied, bucket sort is used for decreasing the computation time. The edges are organized in buckets based on their middle point and the size of the buckets in all three directions is equal with half of the maximum edge length. The faces are organized in buckets based on the centers of their circumscribed circles and the size of the buckets is given by the maximum radius of these circles.

3.3 The complete algorithm

The basic contact algorithm is as follows:

1. Preprocessing stage:

   - Study master surface and create lists with additional edges and faces to check for each master node;
   - Pre-compute all dimensions related to the master surface that are needed in the local search stage (such as normal directions, lengths, bisectors, etc.)
   - Distribute master nodes into buckets;

2. At the end of each Dynamic Relaxation step, for each slave node P:

   - Identify the bucket containing P and search for the closest master node C in that bucket and all the surrounding buckets;
   - Find the closest point on the master surface, R, by searching the master edges and faces that contain C and the additional master edges and faces related to node C
   - Check for penetration, using the normal to the master surface in R;
   - If penetration is detected, move the slave node P to the point R
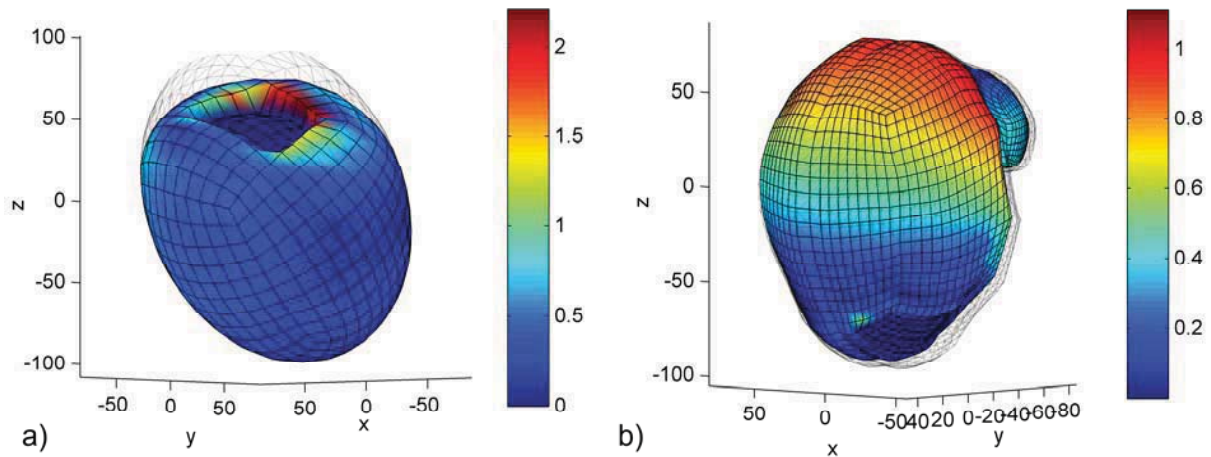
4 Simulation results

In order to assess the performance of the algorithm we performed simulations using our implementation of the contact algorithm (combined with Dynamic Relaxation as a solution method) and the commercial software package LS-Dyna [20] and compared the results.

The same loading conditions and material models were used in both cases. The loading consisted in displacements applied to the nodes from the craniotomy area using a smooth loading curve. Neo-Hookean material models were used for the brain tissue and for the tumor and a linear elastic model was used for the ventricles. In order to obtain the steady state solution, the oscillations were damped away using both mass and stiffness proportional damping in LS-Dyna.

In a first experiment, we displaced an ellipsoid (made of a nonlinear Neo-Hookean material) with the approximate size of a brain inside another ellipsoid simulating the skull. The maximum displacement applied was 40 mm. The average difference in the nodal displacement field between our simulation and the LS-Dyna simulation was less than 0.12 mm (Fig. 4.a).

In another experiment we performed the registration of a patient specific brain shift. LS-Dyna simulations for this case have been done previously and the results were found to agree well with the real deformations [11]. We performed the same simulations using Dynamic Relaxation and our contact algorithm. The average difference in the nodal displacement field was less than 0.2 mm (Fig. 4.b).

For a master surface consisting of 1993 nodes and 3960 triangular faces and a slave surface having 1749 nodes, the computation time dedicated to the contact handling for 1000 time steps is about 3.2 s on a standard 3 GHz Intel® Core™ Duo CPU system.

Fig. 4. Displacement differences (in millimeters) between our results and LS-Dyna simulations are presented using color codes. The transparent mesh is the master contact surface.

It is worth noting that if we refine the master surface and increase the number of triangles 4 times (to 15840), the computation time for 1000 time steps increases to 3.8 s. Therefore the computation time is almost independent of the number of triangles on the master surface. This happens because we use bucket sort with the bucket size depending on the dimensions of the triangles belonging to the master surface.

For the brain shift simulation a mesh with 16710 nodes and 15050 elements was used. The computation time for 1000 time steps was about 12 s and less than 3000 time steps are needed to reach the steady state solution. Therefore we need less than one minute for a complete brain shift simulation.

5   Discussion and conclusions

We presented in this paper a very simple and efficient contact algorithm that can be used for simulating the brain-skull interaction in a biomechanical model, when combined with an explicit solution algorithm – Dynamic Relaxation.

The surface representing the skull is considered rigid and therefore it can be analyzed pre-operatively and many quantities needed for handling the contact can be pre-computed. No parameters are needed for defining the contact (contact thickness, stiffness, etc.), as it only imposes kinematic restrictions on the movement of the brain nodes. The brain nodes are prevented from penetrating the skull, but they can slide along or separate from it.

By imposing only kinematic restrictions, no contact forces need to be computed. Although the contact forces can be extracted from the strains occurring in the brain elements, they are not of interest in our application. The absence of any forces applied on the brain surface leads to a smaller influence of the material constitutive model parameters on the simulation results.

The skull surface is considered to be a C0 triangular mesh, as this leads to a fast method for detecting penetration. If quadrilateral elements were present in this mesh, they can easily be split into two triangular elements. Because this surface is not smooth, it can be argued that high frequency vibrations will be introduced in the solution. In commercial codes such vibrations are handled using contact damping or by smoothing the surface (see [20]). Our solution algorithm naturally damps all the high frequency vibrations [18], therefore no additional effort is needed for handling these vibrations.

Combining Dynamic Relaxation with this contact implementation we can perform a brain shift simulation in less than a minute on a normal PC, for a model having over 50000 degrees of freedom. Therefore, we are one step closer to intra-operative brain shift simulation.

6 Bibliography

1. Nakaji, P. and Speltzer, R.F.: The Marriage of Technique, Technology, and Judgement. Innovations in Surgical Approach (2004) 51: p. 177-185

2. Himpens, J., Leman, G., and Cadiere, G.B.: Telesurgical laparoscopic cholecystectomy. Surgical Endoscopy (1998) 12: p. 1091

3. Hata, N., et al.: Three-dimensional optical flow method for measurement of volumetric brain deformation form intraoperative MR images. Journal of Computer Assisted Tomography (2000) 24: p. 531-538

4. Soza, G., et al.: Determination of the elasticity parameters of brain tissue with combined simulation and registration. International Journal of Medical Robotics and Computer Assisted Surgery (2005) 1: p. 87-95

5. Nabavi, A., et al.: Serial intraoperative magnetic resonance imaging of brain shift. Neurosurgery (2001) 49: p. 75-85

6. Lavallée, S.: Registration for Computer Integrated Surgery: Methodology, State of the Art., Computer-Integrated Surgery, 1995, Cambridge, Massachusetts: MIT Press

7. Beauchemin, S.S. and Barron, J.L.: The computation of optical flow. ACM Computing Surveys (1995) 27(3): p. 433-467

8. Viola, P. and Wells III, W.M.: Alignment by maximization of mutual information. International Journal of Computer Vision (1997) 24(2): p. 137-154

9. Toga, A. and Mazziotta, J.: Brain Mapping - the methods. Academic Press, Sydney (2002)

10. Wittek, A., et al.: Brain shift computation using a fully nonlinear biomechanical model, 8th International Conference on Medical Image Computing and Computer Assisted Surgery MICCAI 2005, 2005, Palm Springs, California, USA

11. Wittek, A., et al.: Patient-Specific Model of Brain Deformation: Application to Medical Image Registration. Journal of Biomechanics (2007) 40: p. 919-929

12. Hu, J., et al.: Intraoperative brain shift prediction using a 3D inhomogeneous patient-specific finite element model. Journal of Neurosurgery (2007) 106: p. 164-169

13. Haines, D.E., Harkey, H.L., and Al-Mefty, O.: The "subdural" space: A new look at an outdated concept. Neurosurgery (1993) 32: p. 111-120

14. Lunn, K.E., et al.: Data-guided brain deformation modeling: evaluation of a 3-d adjoint inversion method in porcine studies. IEEE transactions on bio-medical engineering (2006) 53: p. 1893-1900

15. Ferrant, M., et al.: Serial registration of intraoperative MR images of the brain. Medical Image Analysis (2002) 6(4): p. 337-359

16.   Warfield, S.K., et al.: Real-time registration of volumetric brain MRI by biomechanical simulation of deformation during image guided surgery. Computing and Visualization in Science (2002) 5: p. 3-11

17.   Wittek, A., Hawkins, T., and Miller, K.: On the unimportance of constitutive models in computing brain deformation for image-guided surgery. Biomechanics and modeling in mechanobiology (2008) published online

18.   Underwood, P.: Dynamic Relaxation, in Computational Methods for Transient Analysis, Editor: T. Belytschko and T.J.R. Hughes (1983) New-Holland: Amsterdam. p. 245-265

19.   Skrinjar, O., Nabavi, A., and Duncan, J.: Model-driven brain shift compensation. Medical Image Analysis (2002) 6: p. 361-373

20.   Hallquist, J.O.: LS-DYNA Theory Manual. Livermore Software Technology Corporation, Livermore, California 94551 (2005)

21.   Sauve, R.G. and Morandin, G.D.: Simulation of contact in finite deformation problems – algorithm and modelling issues. International Journal of Mechanics and Materials in Design (2004) 1: p. 287–316