

# Use of Boundary Element Analysis for Simulating Neurosurgery

## **ABSTRACT**

We present the design, implementation and formative evaluation of a virtual-reality simulator for neurosurgical training. Our simulator provides visual and haptic feedback for performing basic surgical actions including prodding, pinching and cutting deformable brain tissue. This is achieved by applying the Boundary Element (BE) method from continuum mechanics, rather than the more conventional Finite Element (FE) method, as this offers potential performance improvements for simulation as it only models the surface of an object rather than its entire internal structure. We introduce BE-based algorithms for our core surgical actions and explain how they have been implemented in real-time through the use of a pre-solution approach combined with an iterative solver that updates the pre-solutions whenever the structure of the three-dimensional (3D) brain model has changed, e.g., following cutting. We describe how our techniques have been built into a simulator that enables two-handed interaction and that can import 3D models that have been generated from Magnetic Resonance Imaging (MRI) scans. We present a formative evaluation in which practicing and trainee neurosurgeons provided initial feedback after using our simulator. Their ratings and comments suggest that simulators such as this can play a useful role in neurosurgical training and that in general terms, they can achieve a

sufficient degree of realism, although further work is required to enhance the realism of pinching and especially cutting and to simulate new effects, most notably bleeding.

## **Keywords**

Neurosurgery Simulation, Deformable Models, Haptics, Boundary Element Method, Visualisation.

## **1. INTRODUCTION**

Standard surgical training consists of the acquisition of theoretical knowledge complemented with practical observation during surgery and, at a suitably advanced stage, performing the relevant procedures on real patients under appropriate supervision. Even experienced surgeons still require training to perform complex, non-routine operations. However, a potentially contradictory situation exists whereby patients' expectations of surgical experience are obviously high, whereas clinical governance requires surgeons' track records to be established before hospitals permit them to operate. To compound the problem, there is increasing fear of litigation among healthcare professionals, along with work-time legislation which limits the hours available for training.

Current practice consists of trainee surgeons utilising cadavers, preserved organs and observing more senior neurosurgeons conducting the relevant procedures. These present obvious limitations, such as inconsistent operational realism, limited patient feedback and intrusion into senior practitioner's time. There is therefore an increasing role for surgical simulation systems for training in order to overcome initial lack of experience (much as is current practice in civil aviation), and for planning of non-routine or more complex techniques.

Various surgical simulators have previously been developed including operation planning tools (Meyer, et al 2001), rigid structure surgery (such as bone removal) (Tsai, et al 2001) and, of more relevance to this paper, procedures on deformable human organs and soft tissue (Delingette, et al 1999,

Moutsopoulos and Gillies, 1997 and Keeve, et al 1998). The use of virtual reality (VR) in surgery is supported by recent developments in real-time interactive graphics and haptics, delivered by increasingly sophisticated input/output devices such as the Phantom force-feedback device from SensAble Technologies(SensAble, 2005, Schijven and Jakimowicz, 2003 and Gallagher and Cates, 2004). However, early surgical simulators have been severely limited in terms of the complexity of the structures and procedures that can be simulated. One of the critical challenges for developing more effective surgical simulators lies in modelling complex interactions with deformable structures, especially the cutting of soft tissue. This provides the key focus for our paper which introduces a virtual reality simulator for training neurosurgeons which is based on the novel use of boundary element analysis rather than the more traditional finite element analysis to model indentation and cutting actions on deformable soft brain tissue.

## **2. THE POTENTIAL OF BOUNDARY ELEMENT ANALYSIS FOR SIMULATION**

Numerical approaches used to simulate real-time deformable models can be divided into two main groups; lumped-mass-spring methods and continuum mechanics methods. Lumped-mass-spring methods are relatively simple to implement but do not strictly adhere to the laws of continuum mechanics which require continuity between adjacent elements. Consequently, they may suffer from lack of realism and instability of the deformable models. Continuum mechanics techniques, namely the Finite Element and Boundary Element techniques, are robust and potentially far more accurate than lumped-mass methods, but require the solution of often thousands of simultaneous equations, and hence are much more computationally expensive.

The most popular method today, the FE method (see, e.g. Bathe, 1982) breaks down a continuum into small areas or volumes whose force-displacement relationships are calculated and then assembled mathematically to simulate the system behaviour. In its simplest form, this approach assumes linear

elastic behaviour and has been well explored for surgical simulation applications (for example Delingette et al, 1999). Nonlinear FE calculations, which are very well-established in engineering applications, have also been applied to a limited number of simulations, for example non-linear elasticity modelling of the simulation of laparoscopic surgery on the liver (Picinbono et al, 2003).

In a recent review of deformable models for surgery simulation, (Meier et al, 2005) have identified the alternative method of BE analysis as being one of the most promising routes to surgical simulation. The BE method, although less widely used in engineering analysis than the FE method, is potentially well suited to virtual reality simulations, since only the surface is discretised into elements or ‘patches’, whereas the FE method requires interior (volume) elements as well as surface discretisation, even though only the surface is visible. A brief overview of the BE approach is given in Appendix A and a detailed description can be found in (**anonomised**).

Two early attempts to use the BE method to support the simulation of deformable objects in VR, published by James and Pai (James and Pai, 1999, 2001) and Monserrat (Monserrat et al, 2001), demonstrated the basic feasibility of this approach. We have built on this baseline, further exploring the potential of BE for surgical simulation, with a particular focus on neurosurgery. An especially challenging aspect of simulating surgery is the real-time simulation of the surgical cutting procedure. Cutting or fracture of a continuum can be modelled using the conventional FE or BE approaches by re-meshing the cut surfaces and re-analysing the new geometry. However, to date, this has not been possible to perform in real-time. A limited number of FE based VR models have recently emerged for modelling suturing (Berkley et al, 2004) and cutting tetrahedral meshes (Bielser et al, 2004). Our work in this project has extended surgical simulation by establishing new BE based approaches to cutting and post-cutting deformations as well as simulation of gravitational loading and self-contact between tissues. The remainder of this paper describes the design, implementation and formative evaluation of a

neurosurgical simulator based on the BE approach, including techniques for simulating pinching, pulling and cutting actions in real-time.

### **3. SIMULATOR REQUIREMENTS**

Neurosurgery is, of course, an extremely complex task that involves many different activities, from the pre-operative planning stage to the surgical procedures themselves. A typical procedure consists of removing the relevant part of the skull and the membrane covering (the Dura Mater). The surgeon then marks out the lines of the desired cut, typically a few centimetres long, providing a guideline or track for subsequent cutting. The surgeon can now cut through the brain down to the target area where more detailed cutting (for example in removing a tumor) occurs. It is this phase of the procedure that provides the initial focus for our simulator in an effort to help trainee surgeons learn the process of cutting through brain tissue to reach a specific target. This is a specialised cutting procedure, however. The surgeon cuts through the tissue in a series of small pinching motions, using diathermic forceps which pass high-frequency electric current between their tips, activated by a foot switch. These first pinch and then effectively melt a small volume of tissue at a time (typically  $1 \text{ mm}^3$ ) which is then sucked away using a suction tube. The surgeon continues to move along the marked line and deepen the cut, possibly utilising retractors for improved access. A skilled surgeon typically makes four such pinches and cuts every second, occasionally stopping to separate the cut surfaces with permanent retractors so as to allow access to the required area. Once the tumor has been located, a similar cutting process is effective in its removal. The whole process may typically last between 10 and 30 minutes. Given these observations, our simulator must meet the following high-level requirements:-

- Simulate the core surgical actions of prodding, pinching and cutting brain tissue.

- Utilise virtual representations of surgical tools in their correct context, including diathermic forceps for cutting, pincers for pinching and retractors for holding open a cut so that it can be deepened.
- Cater for two-handed operation.
- Simulate the correct position and posture for the trainee surgeon.
- Use accurate brain models throughout the simulation.
- Use patient specific data if the procedure pertains to a rehearsal of a specific operation.

In addition, the simulation must offer realistic visual and force feedback to the participant so that they can see the deformations of the brain tissue as they interact and they can also “feel” the resultant interactions. This typically entails maintaining update rates of more than 25Hz for visual feedback and 1000Hz for haptic processing. Stereoscopic vision should also be incorporated to recreate the physical procedure as far as possible.

## **4. SIMULATOR DESIGN**

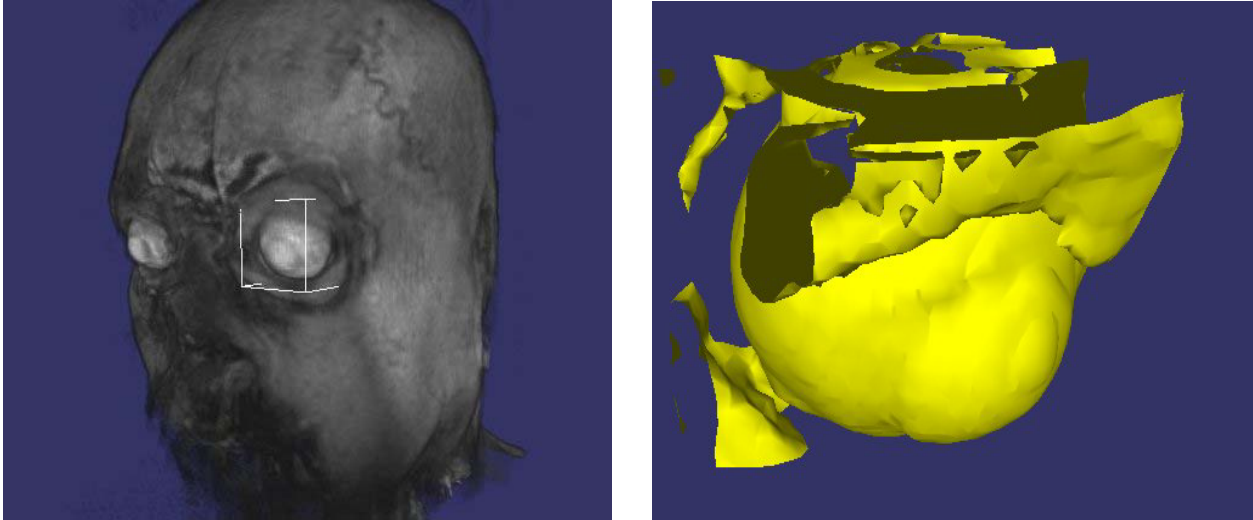
We now describe the design of a simulator that is intended to meet these requirements covering the generation of a 3D model of the relevant area of the brain from MRI data; new techniques for simulating prodding, pinching and cutting in real-time; and the incorporation of these into a physical framework.

### **4.1. Virtual Brain Model Generation**

The first stage of the process of neurosurgical simulation involves acquiring a suitable virtual model of the region of brain to be operated upon, specifically one that represents the prospective patient. Our simulation uses MRI data to generate a three-dimensional graphical model of the brain. A succession of two-dimensional MRI images is combined to form a three-dimensional volume as shown in figure 1

(left). Modern graphics processors can represent this structure internally as a three-dimensional texture. This texture can then be sampled at regular intervals and applied to a number of planar polygons that go to form the volume. The exact number of polygons can be interactively user-controlled, thus allowing sub-sampling of the data, thereby decreasing the computational load at the expense of a less detailed view of the internal structure (i.e. fewer polygons entails fewer visible slices through the dataset and correspondingly less information is conveyed). The user may then select the required region of the brain to be modeled by using the wireframe cube as shown in figure 1 (left).

Once the region has been selected, a solid model is generated by applying a modified version of the Marching Cubes algorithm (Lorenson and Cline, 1987). This algorithm interrogates the data by considering the intensity values at each of the eight corner points of a cube. Should any of these values straddle a pre-determined threshold (supplied by the user) then it is considered to be either side of a “surface” contained within the dataset. Interpolation of these values can then be used to construct an appropriate set of vertices. This technique is then continued throughout the volume to extract a surface representation of the relevant structures (see figure 1 (right) for an example). The model may also be generated at differing levels of geometric detail, by effectively altering the cube size used to sample the data. The greater the cube-size the fewer points get sampled and hence fewer vertices are used in the resulting model. This is particularly pertinent for modeling structures based on Boundary Element techniques, as the processing time increases with the number of elements contained within the model. Therefore provided that the outlying model resembles the organ under simulation to a sufficient degree, a less detailed structure may be successfully utilised. This model is then imported into a commercial 3D modeling package so that finer enhancements, such as texture mapping, may be applied manually.



**Figure 1:** Volume representation of MRI scan data showing selection cube (left) and the resulting solid model of the selected volume (in this case an eye) derived using the Marching Cubes algorithm (right)

## 4.2 Software Simulation of Prodding, Pinching and Cutting

We have made advances in applying the BE technique to produce algorithms to simulate the key actions of prodding deformable tissue, cutting it, applying a retractor tool to a cut area, and modelling the deformation of the tissue as a result of self-contact. The mathematical details of these algorithms from a BE perspective can be found in (**anonimised**). We have also developed techniques for implementing our algorithms with sufficient performance to support real-time simulation.

### *Simulation of prodding and pulling*

We have developed a BE algorithm for simulating the prodding of deformable brain tissue. This also enables us to simulate pinching and pulling, as these are essentially the same as prodding but in the reverse direction. The brain tissue is modelled as a fine mesh of triangular polygons. In practice, the surgical operation is focused on only a small part of the brain. In order to reduce the total number of degrees of freedom to be considered, the surface of the object can be divided into several sub-regions, with only one sub-region (the part being operated on) being finely meshed while the other sub-region

can remain relatively coarse. Uneven meshing is a common modelling strategy in BE, or indeed FE, analysis which requires that the mesh sizes are gradually, not abruptly, changed.

Before the physical deformation effects of the virtual brain can be modelled, its interactions with a typical surgical tool (forceps, etc.) must be considered. A simple contact algorithm has been developed to represent the contact of a single-point surgical implement with the discretised surface of the brain model. A ray-plane intersection test is utilised to determine the exact element that is under contact and the penetration distance. The resulting penetration displacement components are then fed into the BE model. Two sets of data are calculated using the BE model in order to model prodding and pulling of the structure: the vertex displacements (and hence new vertex positions) for all visible vertices in the model (in order to provide the information for visual rendering), and the force components required to cause the imposed displacement, which are fed back to the haptic device as reaction forces.

#### *Development of a pre-solution approach to enable real-time performance*

As noted previously, we estimate that the visual refresh rate of the simulator needs to be at least 25Hz so as to give a visually acceptable appearance, while the haptic rendering rate needs to be around 1000 Hz to give an acceptable feel. However, the solution of the BE system equations takes significantly longer than the haptic update rate of 1000Hz would allow; in other words realistic haptic interaction cannot be achieved by direct solution. For a linear system and where deformations are small compared to the size of the model structure, we can exploit the superposition principle as follows. When a surgery tool contacts an organ, it can be represented as a non-zero displacement being applied on the deformable model. Prior to the real-time simulation, a look-up table containing a set of pre-solutions can be created by applying a hypothetical unit displacement on all surface nodes (where a node is the midpoint of a single triangular element in this case) in turn and solving the system of BE equations for the structure without gravitational loads (see Appendix A for further details). The deformation at a node due to a

given displacement is then a straight forward linear transformation of the corresponding unit displacements. This approach is similar to the “pre-computed Green’s function” approach described by James and Pai (James and Pai, 2001) (who also use wavelet compression to reduce storage requirements, a technique not used here), and is analogous to the “fast finite element” approach of Bro-Nielson (Bro-Nielson, 1998).

Calculating the look-up table of pre-solutions uses an iterative method. It is a computationally expensive process, requiring of the order of  $2N^2$  floating point operations per iteration for a structure with  $N$  degrees of freedom (where the number of degrees of freedom is 3 times the number of elements in the virtual model), and typically requiring 25 iterations to find one set of pre-solutions. However, once available, deformations due to prodding can then be calculated by a simple table lookup and three multiplications, requiring of the order of  $3N$  floating point operations at a rate of approximately 1000 HZ and so can achieve sufficient performance for real-time graphical and haptic rendering.

#### *Simulation of real-time contact between a retractor (such as a flat spatula) and the brain*

We have also developed algorithms to detect and identify nodes/elements contacting the tool plane during the movement of a retractor. In the simulation, the retractor is inserted into the gap produced by a cut to separate brain tissues to allow for deeper cutting, or is used to lift up a part of the brain tissue. An algorithm has been developed which considers the different cases which can occur when elements lie close to the retractor. Essentially each triangle/retractor contact is treated as a separate indenter/contact problem in a similar manner to the simple contact algorithm described earlier; while this is clearly a simplification of the true situation, we anticipate it should provide sufficient realism without the need for iteration which would slow down the simulation significantly. With a reasonable number of contacting elements 1000Hz haptic rendering can still be achieved.

### *Simulation of Cutting*

This has been the most challenging part of our work to date, and aims to simulate the cutting process in neurosurgery, in which small amounts of brain material are eliminated using diathermic forceps (i.e., in which the surgeon does not cut with a scalpel, but rather pinches a small amount of tissue between forceps and applies an electrical current to melt it). In principle, the use of the BE method allows the creation of cuts without any prior knowledge of the direction in which the cut can proceed through the brain, since the interior of the brain is initially unmeshed. Algorithms have been developed for creating a cut in previously-uncut brain material, involving the creation of new nodes and elements. The cut can also be extended incrementally both in length and depth. Each step of cut involves the geometrical change of the model and the topological change of the model's mesh and subsequently the updating of the expanded BE system matrices to reflect the new structure that has been generated.

### *Incremental update of the pre-solution table*

A key challenge is to support real-time simulating of cutting, which in this case means being able to simulate up to 4 cuts a second which is the rate at which an experienced surgeon might feasibly work for short bursts (a trainee would probably work at a slower rate). Our cutting algorithms can update the BE model at this rate. However, the introduction of a cut involves changes to the structure of the brain being modelled, and hence the associated pre-solutions based on the original model are, strictly speaking, no longer accurate and have to be updated if subsequent prodding and pulling are to be simulated. This is potentially a time consuming process as noted previously and cannot currently be achieved in real-time, leading us to develop the following additional techniques:

(a) *Approximated update to pre-solution*: in order to allow the simulation to continue without disruption after cutting, an approximate update is performed in real-time during each cutting step by interpolating displacements from either side of the cut so that the newly-created surfaces within the cut can have

plausible deformations. However, this does not take account of the structural weakening of the brain after material has been removed, this will particularly affect the deformations of the area near the cut.

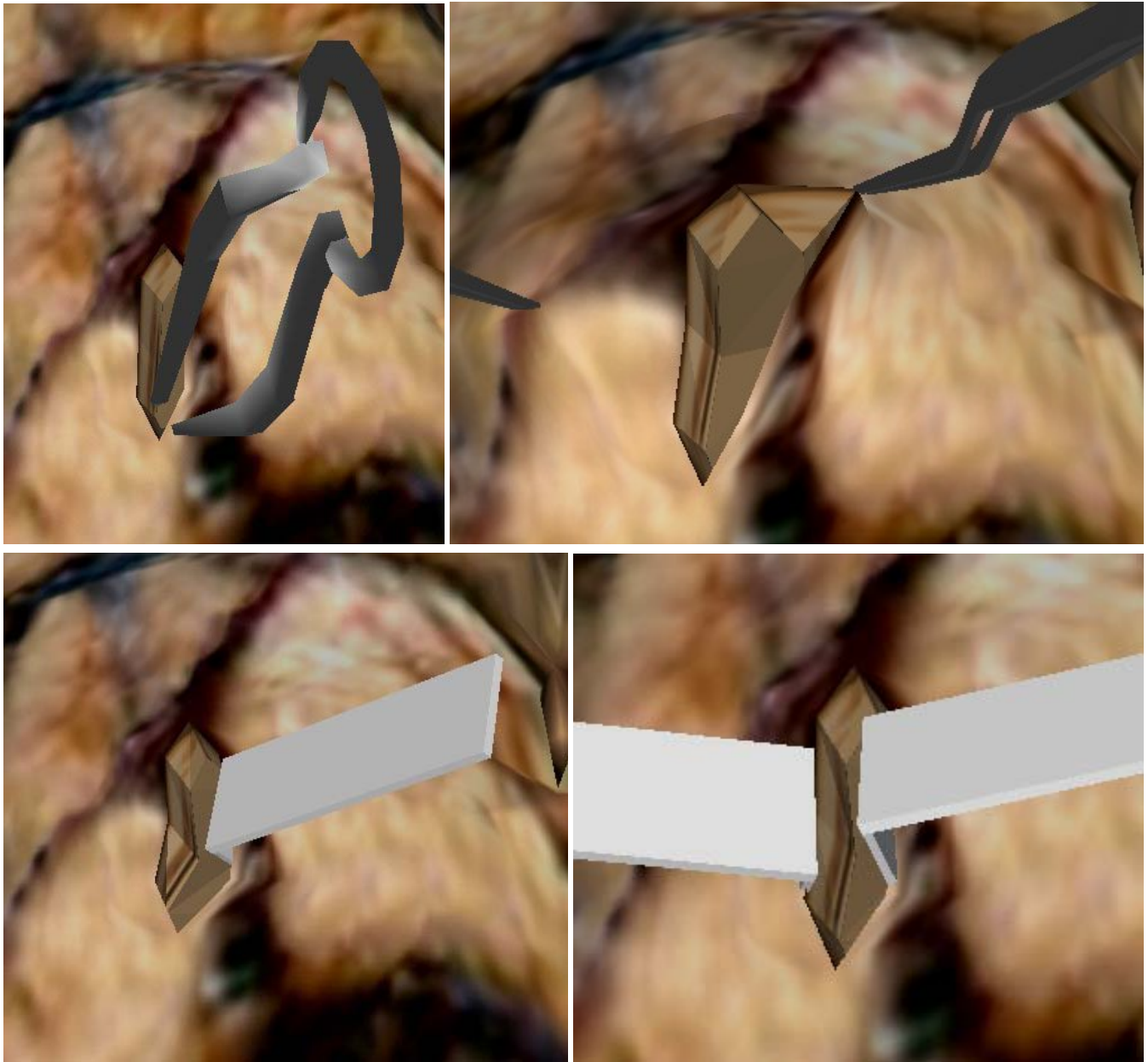
(b) *Real-time multithreaded and parallelisable boundary element analysis*: upon successful completion of a cut, a separate background task is initiated in a differing thread from the main program. The responsibility of this task is to organise and obtain solutions for the affected elements, this will entail initially obtaining solutions for the newly created elements within or upon the cut surfaces and then subsequently for those elements affected by the cut. This approach is taken with the assumption that the greatest divergence from the original solution will occur at the newly created cut surfaces and become less considerable as the distance from the cut increases. It also assumes that the user is most likely to next interact with the model close to where they have just cut. The process of updating the element solutions is achieved by utilising an iterative solver, using the interpolated pre-solution as the starting point for the iterative process and therefore helping to minimise the solution times. Whilst the background task is performing the iterative updates the approximate (interpolated) solutions are used until the update has been completed. Upon completion the original solution matrix can then be updated to reflect the updated object structure. The background thread can then continue updating elements nearer to the cut surface. Should subsequent cuts occur during the updating process the list of affected elements that requires updating is amended accordingly and the iterative process continues. We anticipate that the structuring of the code to support this multi-threaded approach will allow for the parallelisation of certain aspects of the system, which will be investigated in due course.

#### *Simulation of self-contact between two adjacent surfaces of the brain*

This arises, for example, from the newly-created surface of a cut touching under gravitational or external loading, and has been investigated and implemented. It should be noted that self-contact is very different from the spatula contact which assumes that the spatula is a rigid non-deformable body. It is found that

at present the self-contact executes too slowly to be useful in the real-time simulation. Although the material of the object is assumed linear and friction is ignored, contact problems are inherently non-linear because the contact area does not change linearly with the applied load. In practice, the actual contact area is unknown in advance and can only be estimated at first, so that achieving the correct contact status may require many iterations. In each iteration, first the system matrices must be modified according to the contact conditions between contacted elements, and then the system equations must be solved. Finally, a check must be made from the obtained results to modify the contact area which was estimated at first or determined from the last iteration. Using an advanced solver may reduce the solution time down to a short period (half a second or less), but the multiple iterations necessary will require several seconds making real-time calculation impractical. Furthermore, to be accurate the contact analysis requires mesh refinement of the contacting surfaces and this will cause more difficulties and add to the overall processing time. It is therefore concluded that, with current computing power, self-contact is not currently fast enough to be a real-time computation and this remains a challenge for further research.

Figure 2 provides a selection of screenshots from our simulator showing the actions of making a cut, pinching the area around a cut, and applying retractors.



**Figure 2:** Simulating cutting (top left), pinching and pulling the tissue around a cut (top right) and applying retractors (bottom left and right)

### 4.3 The Physical Design of our Simulator

We have embedded our software simulator within a physical framework that is intended to provide an appropriate experience for the neurosurgeon as shown in figure 3. This incorporates two Phantom devices to support two handed operation (currently utilising Phantom Omni devices). Stereo graphical

rendering is realised by using 3D stereoscopic glasses from eDimensional (eDimensional, 2005). Finally, we have realised a physical configuration whereby the surgeon looks downwards onto the virtual scene, which appears to lie behind a semi-silvered mirror, which reflects the image from the monitor. They can therefore use their hands in the appropriate operating posture, thus reproducing the arrangement of a physical operation. The use of a semi-silvered mirror allows the coincidence of the virtual surgical tools and the haptic devices. For convenience, this equipment is then integrated and mounted on a purpose-built wheeled trolley.



**Figure 3:** The physical design of our simulator (left) and typical operating position (right)

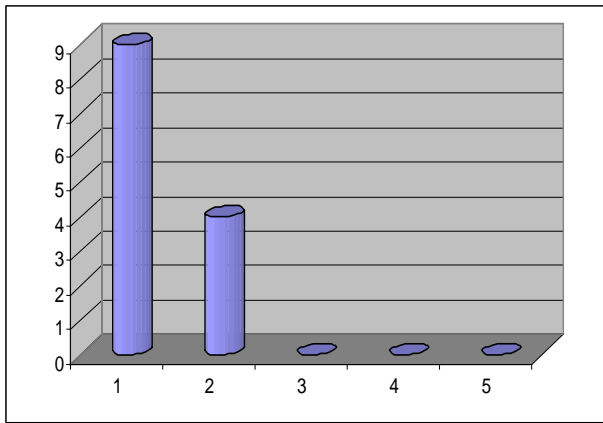
## 5. A FORMATIVE EVALUATION

We have conducted an initial formative evaluation of our implementation. This began with two initial sessions undertaken within the ECMT (European Continuing Medical Training) courses at the Queen's Medical Centre (QMC) in Nottingham in the UK in which the system was demonstrated to more than twenty-five neurosurgical-related staff (ranging from neurosurgeons to theatre nurses) and from which initial feedback was obtained. A refined version of the simulator was then evaluated at a subsequent

session with 13 participants who were either consultant or trainee neurosurgeons. Although this may sound like a small number, it needs to be borne in mind that neurosurgeons are a scarce resource – we estimate that there are approximately 150 operating in the UK at present.

Our evaluation is formative in that participants tried the simulator for a relatively short period of time (a few minutes each) before giving feedback through a short questionnaire which gathered their opinions about its realism (both graphically and haptically) and potential improvements. It should also be noted that the evaluation only addressed prodding and pinching and making a few cuts, but not the use of the retractor tool as this was not fully finalized at the time. However, given the great expertise of our participants, this still provided us with valuable data as to the general effectiveness of our techniques as well as useful suggestions for further work. The outcomes of the evaluation are presented and interpreted in the following sections.

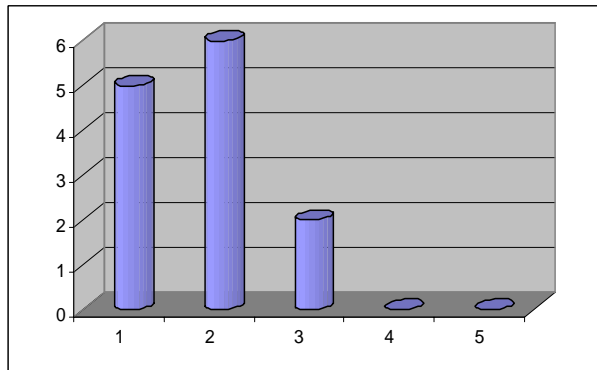
Figure 4 (overleaf) summarises the responses to four general questions that addressed the overall usability and appropriateness of our simulator. On a five-point scale, participants all either strongly agreed or agreed that the simulator was easy to use. No participants found the simulator to be uncomfortable or very uncomfortable, or found the stereo-viewing to be unrealistic or very unrealistic. Finally, participants unanimously agreed that the simulator could help in understanding basic surgical acts. This feedback suggests to us that simulators based on this technology do indeed have to potential to assist with surgical training.



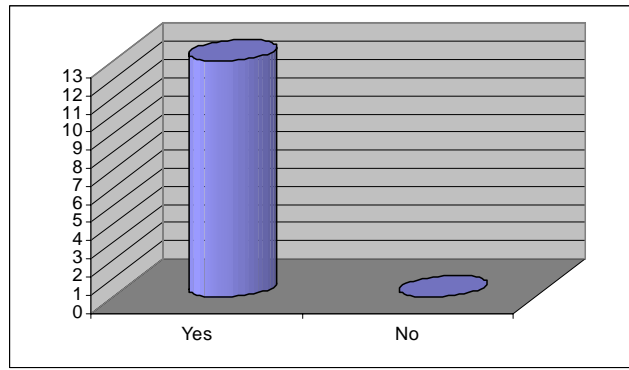
In general, how easy was the simulator to use?  
 (1 very easy, 5 very hard)  
 Mean = 1.3, Standard deviation = 0.48



How comfortable was the physical set-up?  
 (1 very comfortable, 5 very uncomfortable)  
 Mean = 1.54, Standard Deviation = 0.66



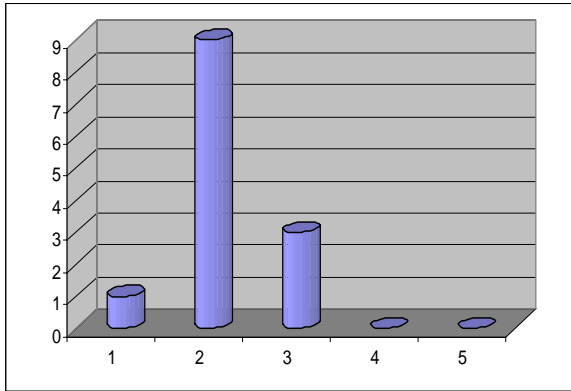
How realistic was the stereo viewing?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 1.77, Standard Deviation = 0.73



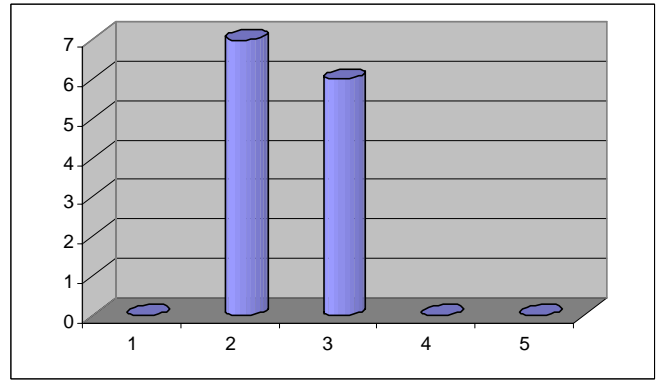
Could the simulator help you understand basic surgical acts?

**Figure 4: General questions about the design and potential utility of the simulator**

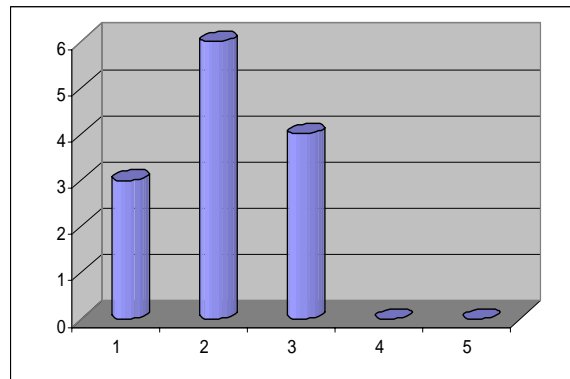
Figure 5 summarises the responses to three questions that focused on the visual realism of the simulation. Our participants generally agreed that the simulation of prodding looked sufficiently realistic (no participant disagreed or strongly disagreed). They also agreed that prodding felt realistic, although less strongly. They also tended to agree that prodding was easy to perform, although less strongly again.



How realistic did the brain look while prodding?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 2.15, Standard deviation = 0.55



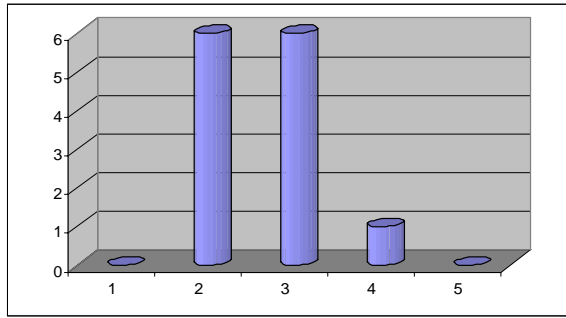
How realistic did prodding the brain feel?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 2.46, Standard deviation = 0.52



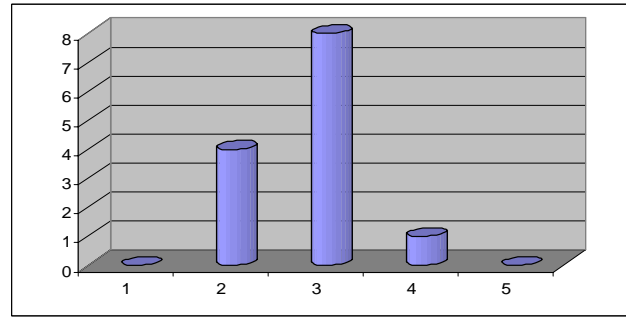
How easy was prodding the brain?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 2.07, Standard deviation = 0.76

**Figure 5: Questions about the simulation of prodding the brain**

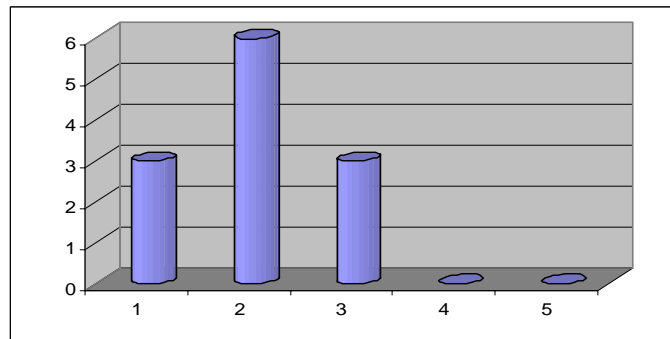
Asking similar questions for pinching revealed a different pattern of responses (figure 6). In this case, the large majority of participants were equally split between agreement that pinching was visually realistic or being neutral about this. However, a few disagreed and no participant strongly agreed. Participants were much more neutral again about whether pinching felt sufficiently realistic. They did, however, tend to agree that pinching was easy to perform.



How realistic did the brain look while pinching?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 2.62, Standard deviation = 0.65



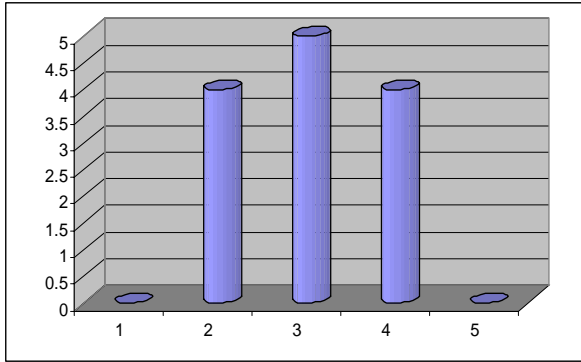
How realistic did pinching the brain tissue feel?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 2.77, Standard deviation = 0.6



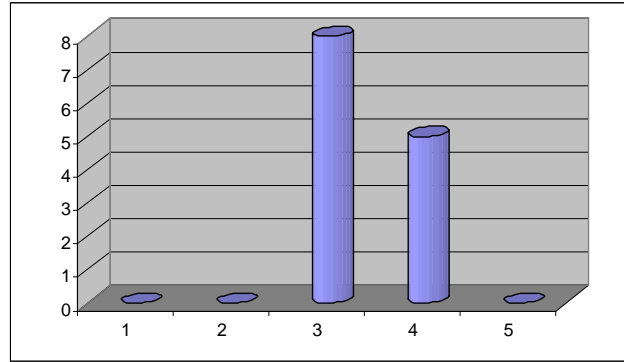
How easy was pinching the brain?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 2.0, Standard deviation = 0.74

**Figure 6: Questions about the simulation of pinching the brain**

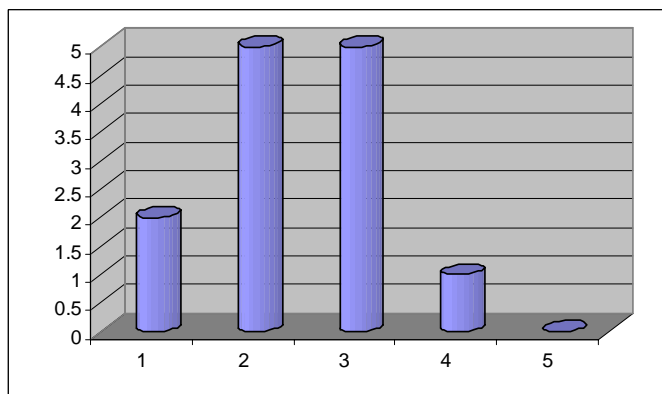
In their comments, one participant observed that the “brain is too robust when stretching and pinching”; another suggested that “the ability to demonstrate irreversible damage from overstretching” would be a useful improvement and a third noted that “strength of tissues” was a factor. We note that it is possible to achieve far greater deformations of the tissue by pinching and pulling than it is by pushing in our simulator, so that pinching is more likely to reveal problems with less realistic simulation of tissue elasticity and behaviour such as tearing, which might explain the above comments and a possible loss of realism.



How realistic did the brain look while cutting?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 3.0, Standard deviation = 0.82



How realistic did cutting the brain tissue feel?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 3.38, Standard deviation = 0.51



How easy was cutting the brain?  
 (1 very realistic, 5 very unrealistic)  
 Mean = 2.38, Standard deviation = 0.87

**Figure 7: Questions about the simulation of cutting the brain**

Figure 7 shows the responses to the same questions, but this time about our simulation of cutting. Participants were neutral about the visual appearance of cutting, with equal numbers being favourable and unfavourable. They were either neutral or disagreed that cutting felt sufficiently realistic. On average though they still tended to agree that cutting was easy, although quite a few were neutral and a few disagreed. Several participants suggested that cutting required improvement, both visually and also in terms of removing delay (there was an initial delay in making the first incision due to implementation issues that are currently being investigated). One observed that “Cutting mode [is] delayed and not

easily directed”, another suggested working on a “faster cutting experience”, a third commented that “improved visuals on cutting would be biggest improvement, and less time delay on cutting”.

Other comments and suggestions concerned wanting to see more detail, especially being able to simulate bleeding: “feedback effects, e.g. bleeding, swelling”, “realistic bleeding”, “add secondary effects such as bleeding”, “pathology within the brain after surface breached. Able to simulate bleeding”, and “variety of operations, surgical hazards, e.g. bleeding”. Clearly simulating bleeding is a key requirement – and indeed a major challenge – for future work.

In summary, preliminary feedback from experts suggest that simulators based on the simulation techniques and hardware described in this paper can achieve a sufficient level of realism to be able to play a useful role within surgical training. Participants’ responses to our questions, both rating and freeform comments, suggest that the simulation of prodding both looked and felt realistic; that they were more neutral about pinching, perhaps due to being able to make large stretching deformations; and that cutting, while functional, does require greater work in terms of look, feel and performance.

## **6. SUMMARY**

In summary, we have presented the design and initial implementation of a neurosurgical simulator. Key features of our simulator are being able to import 3D brain models that are derived from MRI scan data; the simulation of prodding and cutting actions on these models, realised using the Boundary Element analysis technique from computational continuum mechanics, rather than the more conventional Finite Element analysis method; and integration of this simulation with physical hardware that includes stereo graphical rendering and force feedback. Formative evaluation of this simulator by practicing and trainee neurosurgeons suggest that simulators such as this can achieve a sufficient degree of realism in the near future to play a useful role in neurosurgical training. However, a great deal of work is still required to

improve realism, simulate additional effects and simulate more complex procedures with far more detail models, but still in real time. Beyond this, more detailed evaluation is also required involving evaluation when used as part of training rather than through demonstrations as has been the case to date.

## 7. ACKNOWLEDGEMENT

The authors wish to gratefully acknowledge the support of the UK's Engineering and Physical Sciences Research Council (EPSRC) through research grant GR/R84030.

## 8. REFERENCES

Bathe, K.J. (1982) Finite element procedures in engineering analysis, Prentice-Hall, New Jersey.

Anonomised reference
----------------------

Bro-Nielsen, M. (1998) "Finite element modelling in surgery simulation", Proc. IEEE, 86 490-503.1.

Bro-Nielsen, M. (1997) "Fast finite elements for surgery simulation", Studies in Health Technology and Informatics, 39, 395-400.

Delingette, H; Cotin, S; Ayache, N. (1999) "Efficient linear elastic models of soft tissues for real-time surgery simulation", Studies in Health Technology and Informatics, 62, 100-101.

e-Dimensional (2005). Retrieved 25<sup>th</sup> November 2005 from <http://www.edimensional.com>.

James D, and Pai DK. (2001) "A United Treatment of Elastostatic and Rigid Contact Simulation for Real Time Haptics", Haptics-e, the Electronic Journal of Haptics Research, 2(1).

Keeve, E; Girod, S; Kikinis, R; Girod, B. (1998) "Deformable modelling of facial tissue for craniofacial surgery simulation", Computer Aided Surgery: Official Journal of the International Society for Computer Aided Surgery, 3(5), 228-238.

William E. Lorensen and Harvey E. Cline. (1987) Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Proceedings of SIGGRAPH '87, Vol. 21, No. 4, pp. 163-169.

U. Meyer, T. Stamm, N. Meier & U. Joos. (2001) First experience with a public domain computer-aided surgical system, Brit. J. Oral and Maxillofacial Surg., 40 96-104.

Monserrat, C., Meier, U., Alcaniz, M., Chinesta, F and Juan, M.C. (2001) "A new approach for the real-

time simulation of tissue deformations in surgery simulation”, *Comp Meth Prog in Biomedicine*, 64, 77-85.

K Moutsopoulos and D Gillies. (1997) “Deformable models for laparoscopic surgery simulation“, *Computer Networks and ISDN Systems*, 29, 1675-1683.

SensAble Technologies Limited (2005). Retrieved 25<sup>th</sup> November 2005 from <http://www.sensable.com>.

Trevelyan J. (1994) “Boundary Elements for Engineers, Theory and Applications”, Computational Mechanics publications, Southampton.

M Tsai, M Hsieh and S Jou. (2001) “Virtual reality orthopaedic surgery simulator”, *Computers in Biology and Medicine*, 31, 333-351.

Anonomised reference
----------------------

Anonomised reference
----------------------

Zhuang, Y and Canny, J. (2000) “Haptic interaction with global deformations”, *Proc. IEEE Int Conf on Robotics and Automation*, 3, 2428-2433.

Zienkiewicz, O.C. and Taylor, R.L.(2000) “The finite element method”, Butterworh-Heinemann, Oxford, 2000.

## List of Figures

Figure 1 - Volume representation of MRI scan data showing selection cube (left) and the resulting solid model of the selected volume (in this case an eye) derived using the Marching Cubes algorithm (right).

Figure 2 - Simulating cutting (top left), pinching and pulling the tissue around a cut (top right) and applying retractors (bottom left and right).

Figure 3 - The physical design of our simulator (left) and typical operating position (right).

Figure 4 - General questions about the design and potential utility of the simulator.

Figure 5 - Questions about the simulation of prodding the brain.

Figure 6 - Questions about the simulation of pinching the brain.

Figure 7 - Questions about the simulation of cutting the brain.

## APPENDIX A: BRIEF OVERVIEW OF THE BOUNDARY ELEMENT FORMULATION AND OUR PRESOLUTION APPROACH

BE analysis relies upon the existence of surface integral equations to model the continuum behaviour. A displacement integral equation describes the displacement at a source point  $p$  within the domain of the problem due to tractions  $t$  and displacements  $u$  on a surface point  $Q$  on the surface  $S$  of a volume  $V$  as well as the contribution of body forces  $f$  as follows:

$$u_i(p) + \int_S T_{ij}(p, Q) u_j(Q) dS(Q) = \int_S U_{ij}(p, Q) t_j(Q) dS(Q) + \int_V U_{ij}(p, q) f_j(q) dV(q) \quad (1)$$

where  $U_{ij}$  and  $T_{ij}$  are called the displacement and traction kernels, respectively, and are functions of the geometry and material properties but independent of the boundary conditions.

Using Green's theorems, the volume integral in equation (1) can be transformed into a surface integral, thus preserving the surface-only modelling capability of the BE method. Furthermore, the surface is discretized into elements, and the displacements and tractions over each element are described by an interpolation shape function. For simplicity, constant triangular surface elements are used in this work where the displacement is assumed to be linear over each element and the elements can be represented by a node placed at the centroid of the triangle (so that the numbers of elements and nodes are equal as far as the system equations are concerned). The BE software can also be easily adapted for linear and quadratic elements. By taking each node on the boundary in turn as the load point  $p$  and performing the numerical integrations over all boundary elements in equation (1), a set of linear algebraic equations is obtained as follows:

$$[A] \{u\} = [B] \{t\} + \{R_G\} \quad (2)$$

where  $[A]$  and  $[B]$  are matrices that contain the numerical integrations involving  $T_{ij}$  and  $U_{ij}$ , respectively, and  $\{R_G\}$  contains the gravitational load which is obtained via the transformation from volume to surface integral of the last term in (1). Further manipulation yields

$$[A^*] \{x\} = \{R_{bc}\} + \{R_G\} \quad (3)$$

where  $[A^*]$  is the solution matrix,  $\{x\}$  contains all unknown variables (displacements or tractions), and  $\{R_{bc}\}$  contains all prescribed values of displacements and tractions. In the surgery simulation these prescribed displacements and tractions are caused by surgical tools.

Alternatively, (3) (excluding the gravitational load) can be expressed as:

$$[A^*] \{x\} = [B] \{t\} + \{R_{Ubc}\} \quad (4)$$

Here we have separated the boundary conditions into two parts: one due to displacement  $\{R_{Ubc}\}$  and the other due to traction  $[B]\{t\}$ . The prescribed displacements are usually zeros to prohibit rigid body movement so that the right-hand vector  $\{R_{Ubc}\}$  is zero. In such a case, the unknown variables can be expressed as follows:

$$\{x\} = \left( [A^*]^{-1} [B] \right) \{t\} = [G] \{t\} \quad (5)$$

where  $[G] = [A^*]^{-1} [B]$ .

It should be noted that all the above assumes linear elastic behaviour of the object being modelled. This assumption has been used in almost all existing real-time BE and FE deformable models (BE and FE), since the extension to non-linear material behaviour is currently computationally too costly to be implemented in real-time.

*Development of a pre-solution approach:* The solution of equation (4) for each value of applied traction typically takes significantly longer than the haptic update rate of 1000Hz would allow; hence realistic haptic interaction cannot be achieved by direct solution. Instead, an array of all possible interactions between an indenter (e.g. a surgical tool) and the brain surface is created off-line and stored as a pair of “pre-solution arrays”. The resulting displacements and feedback forces due to an imposed displacement vector  $\{dx, dy, dz\}^T$  on the  $J$ 'th element are reconstructed by superposition using the following equations, noting that the three components of displacement and force on the  $J$ th element are respectively stored in columns  $(3J-2)$ ,  $(3J-1)$  and  $(3J)$  in the displacement pre-solution matrix  $[U_p]$  and in rows  $(3J-2)$ ,  $(3J-1)$  and  $(3J)$  of the 1D pre-traction solution column vector  $[T_p]$ :

$$disp(i) = d_x U_p(i, 3J-2) + d_y U_p(i, 3J-1) + d_z U_p(i, 3J), \quad i = 1, 3N$$

(6)

$$Force(i) = Area(d_x T_p(3(J-1) + i, 1) + d_y T_p(3(J-1) + i, 2) + d_z T_p(3(J-1) + i, 3)), \quad i = 1, 3$$

(7)

where  $N$  is the number of elements and nodes. This stage was a necessary (if incremental) stepping stone to creating a real-time simulation involving haptic feedback. In the present BE implementation,  $[U_p]$  and  $[T_p]$  are found by repeated solution of equation (5) for cases of an applied unit displacement in the three Cartesian directions at each of the nodes, determining the force components necessary to cause these displacements then calculating the nodal displacements over the whole structure. This approach is similar to the “pre-computed Green’s function” approach described by James and Pai [2001] (who also use wavelet compression to reduce storage requirements, a technique not used here), and is also analogous to the “fast finite element” approach described by Bro-Nielson [1998].