

# Using Region-of-Interest Based Finite Element Modelling for Brain-Surgery Simulation

Kim Vang Hansen, Ole Vilhelm Larsen

Virtual Centre for Health Informatics  
Department of Medical Informatics and Image Analysis  
Aalborg University  
Denmark

**Abstract.** Brain surgery simulation requires a mathematical model of the geometric and elastic properties of the entire brain. To allow for realtime manipulation of the model it is necessary to differentiate the level of accuracy between different subparts of the brain model. A Finite Element Model (FEM) of the brain is presented capable of differentiating the spatial and temporal accuracy in different parts of the model. In a user defined region-of-interest around the surgical target point a dynamic FEM model is used to give high accuracy. The remaining parts of the brain is modelled by a static FEM model having less accuracy. The two models are integrated into one model for the entire brain using Condensation. In the context of our early version of a brain surgery simulator we have tested the condensed model versus a full dynamic model of the brain. Promising results concerning spatial error and execution time are shown.

## 1 Introduction

Surgery simulation [4][1][7][11] have gained growing interest in recent years. The reasons for this are manifold. The success of flight simulators has shown the attainable benefits by letting a trainee practice in a virtual computer environment prior to real world actions [12]. The fast development of computer power now starts to allow interactive manipulation of 3D models representing human organs [4].

The work presented in this paper is motivated from practical needs related to developing a surgery simulator for brain surgery. The aim is to develop a computer-based simulator where a trainee can practice general surgical procedures or where a skilled surgeon can practice surgical details and plan a specific operation prior to performing the specific surgical procedure.

The core of a surgery simulator is a 3D model of the organ of interest and simulation is to a large extent manipulations of the 3D model. One of the fundamental requirements of a surgery simulator is its ability to model deformations of tissue. Surgical procedures often involves pushing around tissue and cutting in tissue which gives rise to deformations of the tissue.

It can be argued that the following three requirements are important in the development of deformable models of human tissue:

- The models should allow for cutting.
- The models should be physically realistic.
- Deformations should be calculated with high speed to allow for real-time simulations.

If cutting should be performed the traditional surface based models can not be used, since these models are empty inside and a cut will not open up for any underlying tissue [1]. Cutting requires volumetric models of the tissue. As a cut is a local procedure, i.e. not effecting the entire organ, the volumetric model must be represented as a local mesh-based model and not as a globally oriented parametric model [5]. The Finite Element Method (FEM) which produces local mesh-based models has been used for different types of surgery simulation, i.e. craniofacial surgery [10], the human leg [1], the liver [4], and the gal bladder [11].

To obtain the most realistic simulation, the model must approximate the physical tissue as well as possible and the deformations should be as similar to deformations of real tissue as possible. We believe that FEM Models, which are models capable of incorporating knowledge of measurable physical properties of tissue, are currently the most promising way to ensure physically realistic deformations.

Real-time simulation is a very challenging requirement. It requires finding the right balance between model size and model accuracy. In brain surgery simulation it is necessary to model the tissue, the blood vessels, nerves, etc. with very high precision, since brain surgery fundamentally is a process of manipulating the details of the brain structure. Models with such a high level of details can only come close to real-time performance if the models are limited in size. In brain surgery simulation it is, however, a problem, that the human brain is a complex organ consisting of several tissue types connected in complicated structural patterns. Surgery at any point in the brain is affected by global properties such as the orientation of the patient relative to the field of gravity, the mass of the brain, the amount of fluid around the brain, the amount of fluid in the internal ventricles, etc., since these properties affect the elastic properties of the brain tissue. Even though brain surgery is almost always performed as minimally invasive surgery, where only a very limited area of the brain is targeted, it is, due to the effect of the global properties, not acceptable to neurosurgeons only to model the area around the target point. It is therefore necessary to develop models for the entire brain, where some parts can be modelled with high precision and other parts with less precision, but still taking into account the elasticity of the brain.

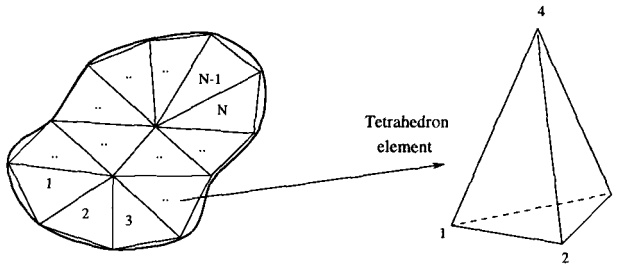
We have previously presented an approach where the model for the entire brain is constructed in order to allow a differentiation of the spatial and temporal accuracy in different parts of the model [8]. The differentiation is obtained by applying a dynamic FEM sub-model with high accuracy to the area around the target point and a static FEM sub-model with less accuracy for the remaining parts. The area with high accuracy is manually pointed out by the user as a special Region-of-interest, ROI. The different models are integrated into one FEM model for the brain using a well known technique called Condensation. In

this paper an overview of the approach is given, and through a line of experiments the approach is evaluated in the context of our early version of a surgery simulator. The experiments are focused on the spatial error and the processing time.

In section 2 the FEM models are described and in section 3 integration of the models using condensation is described. Section 4 and 5 describe and present the results related to experiments conducted to show the performance of the method regarding time consumption and accuracy. Section 6 discusses the results and concludes on the work.

## 2 Modelling by the Finite Element Method

The task of modelling a solid body of matter can be referred to as a continuum problem [9]. In a continuum problem the displacement variable contains an infinitely number of values since it is a function of each generic point in the body. The finite element method reduces the problem to one of a finite number of unknowns by dividing the body into elements and by expressing the displacement field within the element in terms of assumed approximations. Figure 1 shows the discretisation of a solid body into a number of finite elements in the form of tetrahedrons.



**Fig. 1.** Discretisation of the continuum problem into discrete 4 node tetrahedrons.

The tetrahedrons are described by 4 nodes, where the displacement field is described by linear interpolation between nodes. In the finite element representation of a problem, the nodal displacement becomes the finite number of unknowns. Using linear elasticity properties (Hooke's law) to define the relation between applied force  $\mathbf{f}$  and the deformation  $\mathbf{u}$  of the modelled body the equation for each element becomes a linear matrix system:

$$\mathbf{K}_e \mathbf{u}_e = \mathbf{f}_e \quad (1)$$

Since there are 4 nodes in an element each with 3 degrees of freedom  $\mathbf{f}$  and  $\mathbf{u}$  become vectors of  $[12 \times 1]$ , and  $\mathbf{K}$  the *stiffness matrix* is a  $[12 \times 12]$  matrix describing the elastic properties of the element based on material and geometric

properties. Thus, for each element it is possible to define different element properties and then assemble the elements into a complex model.

The elements must be assembled in order to find the properties of the overall system modelled by the mesh of elements. The global matrix equations of the system have the same form as the equations for an element, except the fact that they contain many more terms since they include all nodes. The assembly of the global stiffness matrix becomes,

$$\mathbf{K} = \sum_e t(\mathbf{K}_e) \quad (2)$$

where  $t()$  is a transfer function from element to global node numbers.

## 2.1 Static Finite Element Method

The result of the assembly procedure is a large linear equation system which must be solved with respect to  $\mathbf{u}$ . Since  $\mathbf{u}$  is time independent this system is called a static finite element system

$$\mathbf{K}_{[3N \times 3N]} \cdot \mathbf{u}_{[3N \times 1]} = \mathbf{f}_{[3N \times 1]} \quad (3)$$

where  $N$  is the number of nodes. When modelling large complex structures the number of nodes yields a huge equation system unsuitable for interactive applications [6]. Methods using explicit inversion of the stiffness matrix in order to obtain interactive deformations have already been demonstrated [1].

## 2.2 Dynamic Finite Element Method

A dynamic model yields a more physical correct response since it describes the deformation in a temporal context. The displacement vector  $\mathbf{u}$  becomes a function of both place and time. The dynamic system is an extension of the linear static system where mass and damping is added. The equation to be solved becomes:

$$\mathbf{M}\ddot{\mathbf{u}}_n + \mathbf{C}\dot{\mathbf{u}}_n + \mathbf{K}\mathbf{u}_n = \mathbf{f}_n \quad (4)$$

where subscript  $n$  denotes time  $n\Delta t$  where  $\Delta t$  is the size of the time step.  $\mathbf{M}$  is the mass matrix,  $\mathbf{C}$  is the damping matrix. Assuming lumped masses at the nodes and mass proportional damping then,

$$\mathbf{M}_{ii}^e = \rho \frac{V^e}{4}, \quad \mathbf{C}_{ii}^e = \alpha \mathbf{M}_{ii}^e \quad (5)$$

where  $\rho$  is the mass density,  $V^e$  the element volume,  $\alpha$  is a scaling factor and  $i$  is the element nodal number ( $i=1,2,3,4$ ).  $\mathbf{M}$  and  $\mathbf{C}$  are assembled similar to the stiffness matrix  $\mathbf{K}$ .

The dynamic equation is solved for  $\mathbf{u}$  using *the trapezoidal-rule method* to calculate  $\ddot{\mathbf{u}}$  the acceleration and the velocity  $\dot{\mathbf{u}}$  [3].

### 3 Region-of-interest based Finite Element Modelling

To obtain the most precise modelling of the brain for a simulation, the entire brain should be modelled as a dynamic system. This would involve tens of thousands of nodes in the model and for such large models real-time simulation would not be possible.

The main idea behind this paper is to have a user-defined target area for the surgical procedure and then only model a Region-of-interest (ROI) around the target area as a dynamic system. For such small regions real-time simulation is expected to be possible.

As argued in the Introduction the remaining part of the brain is still very important for modelling the target area and can thus not be disregarded. We suggest to model the remaining part as a static model and then, through condensation, integrate the two models into one model.

The integration ensures that the elastic properties of the remaining part affects the dynamic system of the ROI. This means that when pushing tissue around in the target area you not only sense the tissue in the small dynamic model, but also the resistance from tissue in the larger static model. If for example the static models contain non-deformable material like the skull, this will directly be sensed when manipulating tissue in the target area. The information left out to give a simpler system is related to the dynamics of the deformations. The static model can only tell us how the deformation will look eventually after the model have reached a steady state phase, not how the deformation developed over time.

By speeding up of the calculation of the static model real-time simulation is also expected for this model, and thus for the entire condensed model.

Condensation is a technique by which a matrix equation system can be partitioned into several subsystems allowing individual treatment of the subsystems. Condensation is basically done by eliminating some of the nodes from the matrix equation system by pre-calculations [9].

The ROI is described as a dynamic model marked with subscript  $d$

$$\mathbf{M}_d \ddot{\mathbf{u}}_d + \mathbf{C}_d \dot{\mathbf{u}}_d + \mathbf{K}_d \mathbf{u}_d = \mathbf{f}_d \quad (6)$$

The remaining part is modelled using a static model (subscript  $s$ ).

$$\mathbf{K}_s \mathbf{u}_s = \mathbf{f}_s \quad (7)$$

The complete model ordered with the ROI nodes first:

$$\begin{bmatrix} \mathbf{M}_d & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_d \\ \ddot{\mathbf{u}}_s \end{bmatrix} + \begin{bmatrix} \mathbf{C}_d & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_d \\ \dot{\mathbf{u}}_s \end{bmatrix} + \begin{bmatrix} \mathbf{K}_d & \mathbf{K}_{ds} \\ \mathbf{K}_{sd} & \mathbf{K}_s \end{bmatrix} \begin{bmatrix} \mathbf{u}_d \\ \mathbf{u}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_d \\ \mathbf{f}_s \end{bmatrix} \quad (8)$$

where subscript  $ds$  and  $sd$  refer to the interaction between the two models. In expanded form these equations become

$$\mathbf{M}_d \ddot{\mathbf{u}}_d + \mathbf{C}_d \dot{\mathbf{u}}_d + \mathbf{K}_d \mathbf{u}_d + \mathbf{K}_{ds} \mathbf{u}_s = \mathbf{f}_d \quad (9)$$

$$\mathbf{K}_{sd} \mathbf{u}_d + \mathbf{K}_s \mathbf{u}_s = \mathbf{f}_s \quad (10)$$

When equation (10) is solved for  $\mathbf{u}_s$  and the result is substituted into (9), we obtain

$$\begin{aligned} \mathbf{M}_d \ddot{\mathbf{u}}_d + \mathbf{C}_d \dot{\mathbf{u}}_d + \mathbf{K}_d \mathbf{u}_d - \mathbf{K}_{ds} \mathbf{K}_s^{-1} \mathbf{K}_{sd} \mathbf{u}_d &= \mathbf{f}_d - \mathbf{K}_{ds} \mathbf{K}_s^{-1} \mathbf{f}_s \\ \text{or} & \\ \mathbf{M}_d \ddot{\mathbf{u}}_d + \mathbf{C}_d \dot{\mathbf{u}}_d + \tilde{\mathbf{K}} \mathbf{u}_d &= \tilde{\mathbf{f}} \end{aligned} \quad (11)$$

where  $\tilde{\mathbf{K}} = \mathbf{K}_d - \mathbf{K}_{ds} \cdot \mathbf{K}_s^{-1} \cdot \mathbf{K}_{sd}$  and  $\tilde{\mathbf{f}} = \mathbf{f}_d - \mathbf{K}_{ds} \cdot \mathbf{K}_s^{-1} \cdot \mathbf{f}_s$ . The inversion of  $\mathbf{K}_s$  can be done prior to the simulation and will not influence the total complexity of the condensed model.

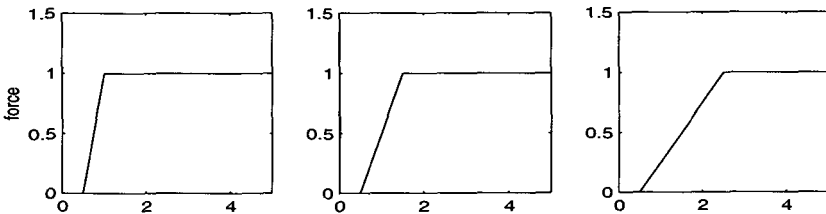
## 4 Experimental Design

The experimental work is aimed to measure the spatial error introduced and the processing time involved in computing the deformation.

For the experiments only a simplified model of the brain is used. The very precise model with detailed information about the surface structure, blood vessels, nerves, etc. is still under construction. The deformation used in all experiments is caused by two forces applied on both sides of an opening in the surface pulling the sides apart, see figure 4.

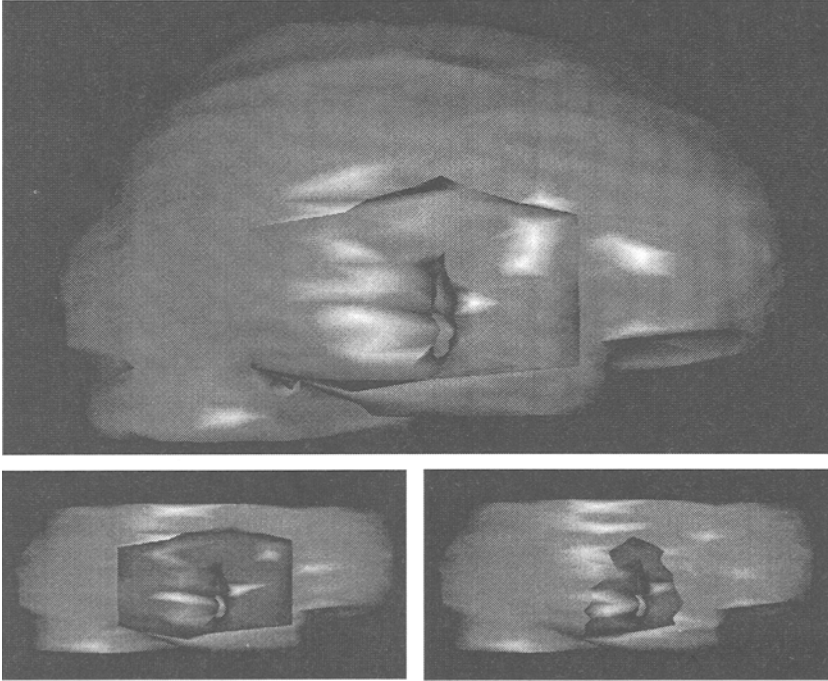
**Measuring the spatial accuracy.** Assuming that the full dynamic system calculates the correct deformation, it is possible to measure the spatial error of the condensed model by measuring the spatial difference between the result of a deformation for the condensed models and the full dynamic model. The spatial difference is measured as the Euclidean distance between all common nodes in the models, i.e. the nodes in the ROI.

As it is expected that the spatial error is related to the dynamics of the system, the force has been applied to the system in three different ways. As shown in figure 2 the situations are ramp functions with different rise times i.e. 0.5, 1, and 2 seconds. The applied force has been determined experimentally as a force giving realistic deformations of the brain models when viewed in the simulator. The duration of the test is 200 iterations or 10 seconds.



**Fig. 2.** The forces are ramp functions with a risetime of 0.5, 1 and 2 seconds.

To determine which effect the size of the ROI will have on the accuracy and processing time two setups were made. The full dynamic model contains 799 nodes, setup 1 is a condensed model of 306 nodes and setup 2 contains 228 nodes.



**Fig. 3.** The models used in the experiments. The full dynamic model (*top*), the condensed model of setup 1 (*left*) and the condensed model of setup 2 (*right*). The dark areas illustrate the dynamic part of the model.

**Measuring the time consumption.** The time used to calculate the deformation at each time step is measured. Based on the iteration-time a frame rate is calculated.

The remaining test parameters are described in table 1. Since finite element modelling of brain tissue is a fairly new area, the material parameters are not described in literature. As an approximation the brain tissue is modelled with the material parameters of rubber.

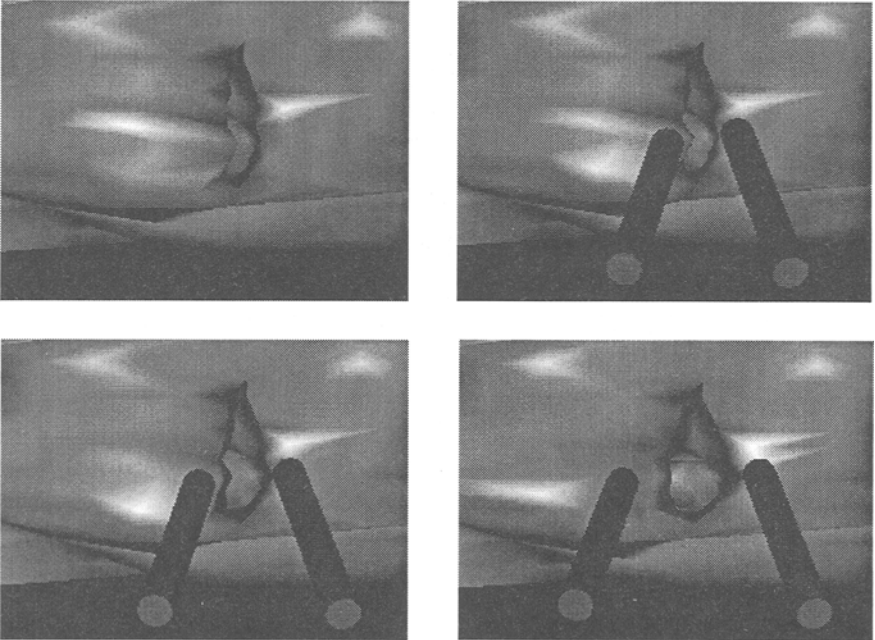
The system was implemented on a Silicon Graphics ONYX with four MIPS R10000 processors and a RealityEngine II graphical interface. Notice that the deformations are calculated using a single processor, parallel features are only used to separate rendering from the actual simulation.

**Table 1.** Parameter settings used during all tests. The rubber lamé constants  $\lambda$  and  $\mu$  defining the material properties for rubber can be found in [2].

$\lambda$	$\mu$	$\rho$	$\alpha$	Time step
0.4	0.0012	1.0	0.8	0.05
$\frac{10^5 \text{ kg}}{\text{cm}^2}$	$\frac{10^5 \text{ kg}}{\text{cm}^2}$	$\frac{\text{g}}{\text{cm}^3}$		sec.

## 5 Experimental Results

Figure 4 visualises how the deformation used in the line of experiments develops over time. The cylindrical objects are representing spatulas capable of pushing tissue around. The time spent on the deformation is either 0.5, 1 or 2 seconds depending on the ramp functions in use (see figure 2).



**Fig. 4.** This figure shows the deformation of the brain model as the forces increase according to the ramp function (figure 2).

**Spatial accuracy.** Figure 5 shows the mean and max. error for the condensed model with setup 1 compared to the full dynamic model.

We notice that the mean steady state error (above 100 iterations) is below 0.2 mm for all three ramps functions. There are, however, significant differences



between the three ramps when looking at the errors occurring shortly after the force has been applied. For ramp 1 and 2 the error rapidly climbs to a peak value of 0.4 and 0.3 respectively. From this peak it is gradually lowered to the steady state error. For ramp 3 it slowly increases until it reaches its maximum value at the steady state error.

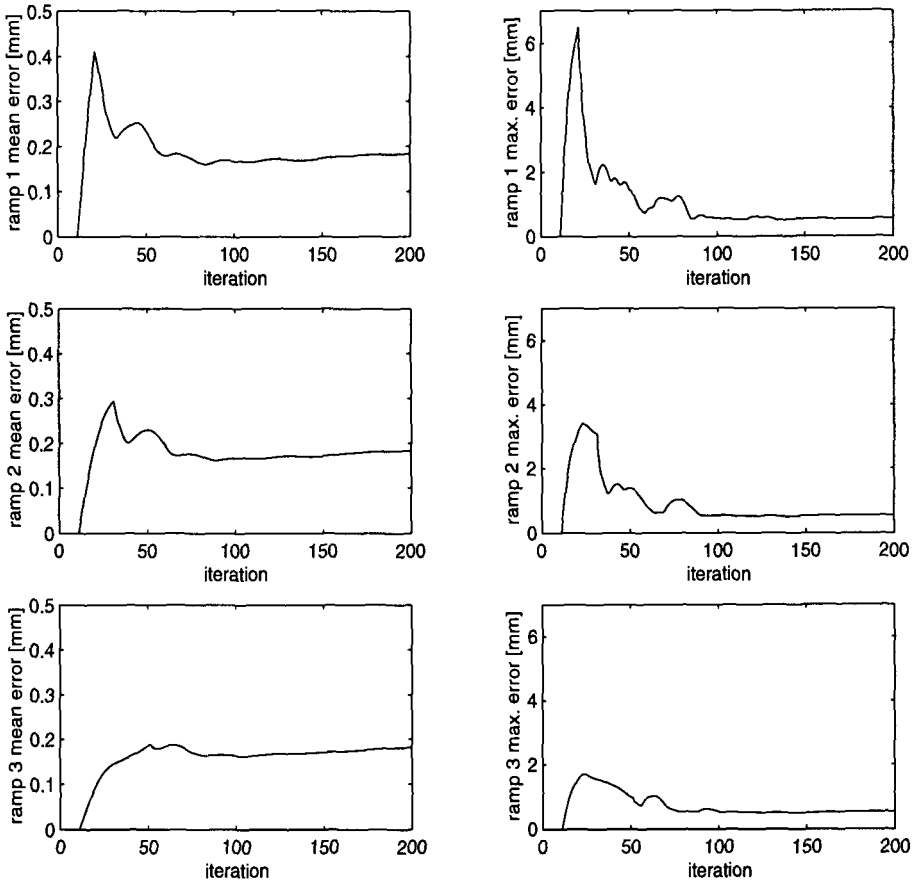


Fig. 5. The mean error and max. error for setup 1.

The maximum error for the condensed model follows a common pattern for the three ramp functions. It peaks shortly after the force is applied with values of approximately 6 mm, 4 mm and 2 mm respectively. Within the next 50 iterations the max. error is lowered to less than 1 mm.

In figure 6 the mean and max errors for the condensed model with setup 2 are shown. Only results with ramp 2 are shown, as all three ramps followed the

same pattern represented by ramp 2. The pattern is a performance very close to that of setup 1 but with a minor increase in the max. error. The most significant change is the duration of the time interval where the max error occurs. The max. error has not one but several peaks in the first 100 iterations.

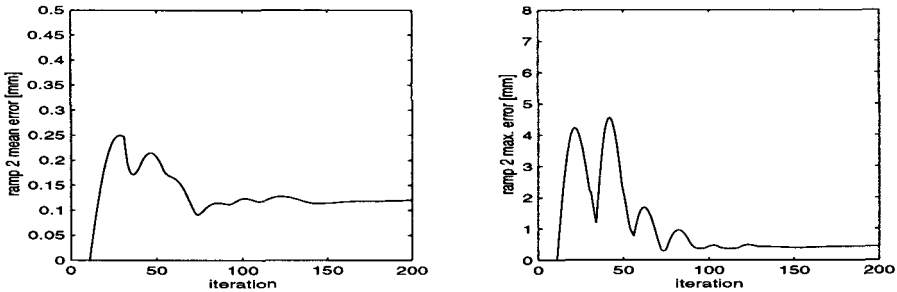


Fig. 6. The mean error and max. error for setup 2 (ramp 2 only).

**Time consumption.** The time consumption was measured as the iteration time, that is, the calculations needed for each time step in order to calculate the new deformations. The iteration times shown in table 2 are the mean of 1000 iterations, the frame rate tells how many times the deformations can be updated per second.

**Table 2.** Time consumption during the tests. The second column shows the iteration time which is a mean of 1000 iterations. The third column shows the according frame rate.

Model	nodes	Iteration [sec.]	Frame rate [Hz]
Full dynamic	799	0.440	2
Setup 1	306	0.071	14
Setup 2	228	0.037	27

## 6 Discussion and Conclusion

Examining the results from the test for spatial accuracy it is characteristic that the condensed model has a very small mean error. The mean performances of the condensed model and the full dynamic model are therefore very similar. The similarity is obtained because the condensation allows the dynamic model to

sense the elastic properties of the larger part of the brain modelled by the static model.

Bearing in mind that real-time performance of the simulator is of interest it is important that the condensed model not only gives small mean errors but at the same time a significant reduction in processing time.

The experiments also show where the price for the reduction is being paid. It is in the dynamic properties of the condensed model. The experiments show max errors up to 6 mm shortly after the force has been applied rapidly. These errors are most likely caused by a mis-match between the deformations of the static and the dynamic model shortly after force impact. In the dynamic model the deformation develops over time. In the static model the final result of the deformation is determined at the instant where the force is applied. The deformation of the static model is therefore presented to the dynamic model prior to the point in time where it would have occurred, had it been modelled as a dynamic model.

However, these problems are only significant for abrupt manipulations, and as large max. errors occur together with small mean errors, only few nodes have been affected by the errors. How these max. errors affect the usability of the condensed model in a surgery simulation needs to be further investigated. A preliminary evaluation by the neurosurgeon related to our surgical simulator team calls for no major concern when used in a realistic simulation situation.

The experiments with the condensed model with different sized dynamic sub-models show that it is possible to develop condensed models with very small dynamic parts and still maintain very good spatial accuracy. As the size decreases, the sensitivity to rapid changes in the force is increased.

Concerning time consumption the results closely reflect the complexity of the models. For the very simple brain model (low node density) used in the experiments the condensed models comes close to real-time performance. As the node density for a realistic brain model is expected to be considerably higher, our dream of a real-time simulator can benefit from the ongoing rapid increase in processing power on the computer market.

Based on the line of experiments presented in this paper it can be concluded that a condensed model with both dynamic and static sub-models has interesting properties for use in brain surgery simulation. It allows for the elastic properties of the entire brain to be taken into account when modelling deformation in a Region-of-interest around the surgical target point. The condensed model has a complexity significantly lower than a dynamic model for the entire brain and therefore the processing time is much lower and realtime performance is coming closer.

The next step in developing a simulator for brain surgery is to enhance the physical realism of the model. Material parameters for brain tissue, blood vessels etc. need to be determined, and the appropriate node density must be computed. Once these problems are overcome, a new line of experiments can determine how close we are to realtime simulation with realistic models.

## References

1. Morten Bro-Nielsen and Stephane Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum*, 15(3):57–66, 1996.
2. Philippe G. Ciarlet. *Mathematical Elasticity, Volume I: Three-Dimensional Elasticity*. Elsevier Science Publisher B.V., 1988.
3. Robert D. Cook, David S. Malkus, and Michael E. Plesha. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, 1989.
4. S. Cotin, H. Delingette, and N. Ayache. Volumetric deformable models for simulation of laparoscopic surgery. In *Computer Assisted Radiology (CAR' 96)*, 1996.
5. S. A. Cover, N. F. Ezquerro, J. F. O'Brien, R. Rowe, T. Gadacz, and E. Palm. Interactively deformable models for surgery simulation. *IEEE Computer Graphics and Applications*, 13:68 – 75, 1993.
6. Sarah F. F. Gibson. 3D ChainMail: a Fast Algorithm for Deforming Volumetric Objects. <http://www.merl.com/reports/index.html>, 1996.
7. Sarah Gibson et. al. Volumetric object modeling for surgical simulation. *Medical Image Analysis*, 2:121 – 132, june 1998.
8. Kim Vang Hansen, Martin Stumpf Eskildsen, and Ole Vilhelm Larsen. Region-of-interest based finite element modelling of the human brain - an approach to brain-surgery simulation. In *Proceedings of the 14th International Conference on Pattern Recognition, (ICPR'98)*, August 1998.
9. Kenneth H. Huebner. *The Finite Element Method for Engineers*. John Wiley & Sons, 1975.
10. Erwin Keeve, Sabine Girod, and Bernd Girod. Craniofacial surgery simulation. Telecommunications Institute, University of Erlangen-Nuremberg, German, 1996.
11. U. Kuhn, Kühnapfel, H.-G. Krumm, and B. Neisius. The Karlsruhe Endoscopic Surgery Trainer - A "Virtual Reality" based Training System for Minimal Invasive Surgery. *CAR*, 1996.
12. Richard M. Satava. Advanced simulation technologies for surgical education. *Bulletin of the American College of Surgeons*, 81, 1996.