UNIVERSITY OF OSLO
Faculty of Mathematics and Natural Sciences

On the Shaken Baby Syndrome

Biomechanical experiments and Numerical simulations

MASTER THESIS

for the degree of

MASTER OF SCIENCE

Anna Blechingberg

May 2010

Contents

List of Figures

List of Tables

Acknowledgment

Without the advice and assistance given by many people, this Master's thesis would not have been possible to write. In particular, I would like to thank my supervisors, Kent-André Mardal, Arne Stray-Pedersen, and Joakim Sundnes, for their enthusiasm, motivation, and support.

I also thank my nice and friendly colleagues at Simula Research Laboratory, and at the Forensic Institute of Medicine at Oslo University Hospital. One person I will mention especially is Karen Helene Støverud. Our interesting discussions, and her constant support has brought joy to each day. I also thank Marie Rognes for our interesting discussions on viscoelasticity.

This work had never been done without the help from Frode Strisland and his colleagues at SINTEF, Oslo. He constructed and built the accelerometer bracket with all the accelerometers, and controlled all experiments and measurements. This opened for several possibilities of analyzing the data.

I especially thank Stine Sofies Stiftelse for financial support of our research project. Without their help, we would not have been able to buy the crash test dummy, and no research had been done. I am very grateful that our collaboration will continue in the future.

My family, especially Jenny Blechingberg Friis and Karin Blechingberg, have also supported me for several years. I am so glad for all of you. I wish we could live closer to each other, and not in separate countries.

Last but not least, I am extremely grateful to my dearest, Svein Inge Skandfer Hanssen, for constantly supporting me every second.

Chapter 1

Introduction

1.1 Motivation

Each year, 40-60 infants and children below 3 years of age are admitted to Norwegian hospitals with severe head injuries [68]. The majority are victims of obvious accidents such as falling from heights or traffic incidents, however in 20-30 % of the cases the recognized injuries can not be explained by the stories provided by their caretakers [63, 86], raising a suspicion of abuse. These cases are regularly referred to as Shaken Baby Syndrome (SBS), indicating that violent shaking may be the mechanism of injury. 12-15 infants in Norway are hospitalized for SBS each year, and 2-3 die of their injuries. 50 % of the survivors become severely injured for life.

The term was introduced by Caffey [12], [13] 35 years ago, but is still a heavily debated diagnosis. The symptoms are mainly broken bridging veins leading to subdural bleedings between the hemispheres, diffuse axonal injury (DAI), and retinal bleedings. Researchers, defense attorneys and the public are questioning the evidence that abusive shaking may cause severe head injuries.

Separating shaken baby syndrome from accidental injuries represents a huge challenge and requires multidisciplinary investigation and cooperation (i.e. medical, forensic, biomechanical, mathematical, engineering, and police expertise [78]). The challenges are not only of medical, social and scientific nature, but also involve difficult ethical and judicial considerations. The hospital staff reporting to child protective services and police may be of great prognostic importance for the child since the recurrence rate for child abuse is high. On the other hand, if the police and medical experts fail to reveal the truth, the parents may be falsely acquitted or even worse falsely convicted to imprisonment for child abuse.

In 2005 Bandak proposed that violent shaking could not cause the SBS injuries [6]. He based his conclusions on the analysis of biomechanical experiments. This

led to difficulties for the courts to prove obvious child abuse. Bandak was later criticized for methodological weaknesses [76]. The biomechanical evidence for shaking to cause SBS are weak. Therefore, further biomechanical studies are needed.

1.2 Main topics of the Master's thesis

Before our work with this thesis started, the Forensic Institute of Medicine at University of Oslo had already done shaking experiments with a crash test baby dummy. They had constructed and built an accelerometer bracket which they placed at the center of gravity in the dummy's head (see Section 2.7 for the construction). They let 15 volunteers shake the dummy in a back and forward motion. The neck of the dummy was damaged in the end of our peak acceleration test, so further experiments with the dummy could not be performed.

We discovered in an early stage of the thesis work that the accelerometer positions on the accelerometer bracket were non-optimal, and that it was impossible to compute 3D acceleration from our measurements (see Section 2.7). All previously developed accelerometer configurations from literature demand a too large space for our purpose, or give a larger error at large angular velocities (see Section 2.4). Our acceleration measurements at the top point of the head were several times larger than previous studies (see Section 5.1), so it is of great importance to validate these measurements. 3D acceleration computation gives a possibility for validating accelerometer measurements – measurements from a few of the accelerometers could be used for computing the acceleration in an arbitrary point. If we take this point as the top point of the head, and compare it with the measurements, we could validate our data.

In this Master's thesis we therefore will

1. Describe the biomechanics of a violent baby-shaking event (see Chapter 5).
2. Attempt to validate the acceleration measurements of our shaking experiments (see Chapters 3 and 5).
   (a) Attempt to validate analytically and numerically the accuracy of the vertex accelerometer measurements (see Sections 3.5 and 5.5).
   (b) Integrate the acceleration data for finding the motion and velocity at the vertex (see Sections 3.6 and 5.3).
3. Study and develop new ways of computing 3D acceleration for limited spaces, and as a consequence achieve new possibilities of validating the

acceleration measurements and compute the 3D motion in world space coordinates (see Chapter 3).

   (a) Study the rigid motion formula written as a matrix exponential, and see if it leads to a simpler numerical method for computing 3D acceleration, properly suited for the nature of our problem (see Section 3.3).

   (b) Develop a new accelerometer configuration suited for 3D computations and limited spaces, and study it analytically and numerically (see Sections 3.4 and 5.7).

4. Study how to calculate the displacement and strain of the brain, and the brain/skull/CSF system of interaction (see Chapters 4 and 6).

   (a) Review literature to determine appropriate material parameters (elastic moduli), and identify their range of variation. Is the best choice other elastic moduli values than at impacts? (See Section 4.4.)

   (b) Choose a simplified elasticity FE model that can be carried out within the time limitations for this Master's thesis (see Chapter 4).

   (c) Calibrate the elasticity model against real world experiments. Solve the elasticity equations with the acceleration field of our maximum shaking acceleration, and find the displacement at the brain/skull boundary (see Chapter 6).

Chapter 2

Background

2.1 Anatomy and physiology

The biggest part of the brain, the cerebrum, is egg-shaped and fills up the major space of the skull. The upper (superior) and side (lateral) surfaces are curved and folded in small ridges and valleys (gyri and sulci). The lower (basal) surface of the brain is hollowed by a peak of bone. This bone peak inhibits the brain to move relatively to the skull at the skull base. The cerebrum is largely divided into two halves, the left and right cerebral hemispheres, by a vertical ridge. The hemispheres are conjoined basally with the brain stem. The brain stem connects the cerebrum with the spinal cord through an opening (the foramen magnum) at the base of the skull.

The surface area of the brain (approx. 5 mm thick) is greyish and constitutes the grey matter (cortex) (see Figure 2.1.1). The nerve cells (neurons) are located here. The central part of the brain has a whitish appearance on a cut surface, and is constituted by thin cords (axons) from the neurons covered by isolation material (myelin – yielding the white color) creating communication lines between the neurons in different parts of the cortex and the mid brain and brain stem. During the fetal development of the brain, the number of neurons increase and the surface of the brain develops its gyri and sulci. A newborn baby does not have myelin around all axons. The myelin develops gradually after birth (myelination).

The brain hemispheres are surrounded by three membranes (meninges) – a hard outer membrane called the dura mater, and two soft inner membranes called the arachnoid and the pia mater. Between the arachnoid and the pia mater, there is a thin layer of water like fluid – cerebrospinal fluid (CSF). The CSF is created from blood by specialized blood vessels mainly located in cavities (ventricles) in the interior of the brain, and the produced CSF floats from these cavities through small channels to the surface of the brain and to the CSF layer around the spinal

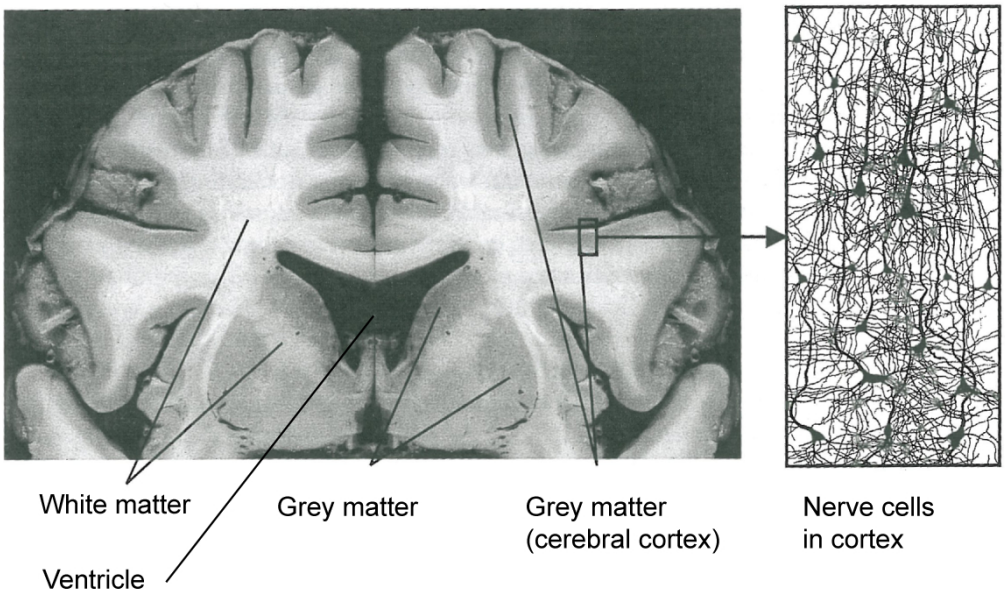

Figure 2.1.1: A vertical cut through the brain. The picture is from [10]

cord.

The brain floats freely in the skull, surrounded by CSF and the soft membranes. When the head is subject to abrupt motion, the brain moves relatively to the skull. The outer hard membrane, the dura, forms a thick mid line ridge between the hemispheres (falx), limiting movement from side to side across the mid line. Also, the dura forms a flat floor at the back part of the skull (tentorium) separating the brain from the brain stem and small brain (cerebellum) which is situated closest to the skull base. The falx hinders subdural bleedings from one hemisphere to cross over to the other hemisphere. This will be important in diagnosing SBS since subdural bleedings in SBS most often are between the hemispheres (inter hemispheric), and bleedings at impacts are placed at the impact area and at the opposite side of the impact (see Section 2.1.1).

As the brain floats freely, there are no structures connecting the brain with the exterior skull except for a few thin blood vessels, called bridging veins, located on the top surface (see Figure 2.1.2). The bridging veins transport blood from the brain through the meninges and further back to the heart. These blood vessels therefore, may be subject to strain when the brain moves relatively to the skull. Rupturing of the bridging veins leads to bleeding (haemorrhage) in the space between the skull and the brain, more precisely in the subdural space (see Figure 2.1.3). Such bleedings are common after severe deceleration impacts such as falling from heights or car collisions. The location of the bleeding indicates the

site of the impact, which is either directly exterior to the bleeding or at the contra lateral site (contre-coup mechanism). Because of the dura fold separating the right and left hemispheres, a subdural bleeding rarely crosses the mid line. The bridging veins are more vulnerable for tearing in the subdural space than in the subarachnoid space since the dura is the hardest of the meninges.

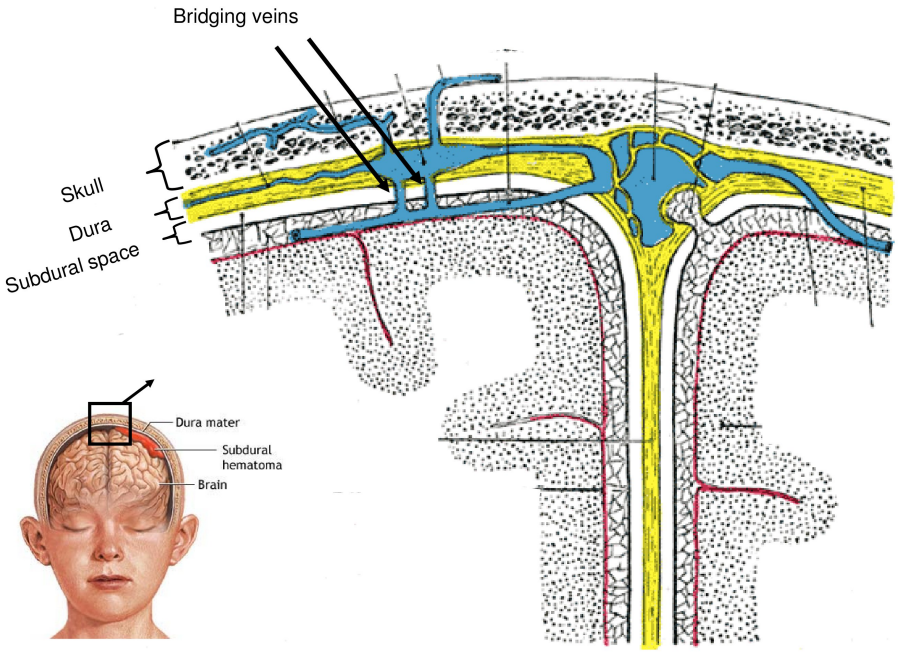

Figure 2.1.2: Brain anatomy with bridging veins at the vertex. The netted area is the subarachnoid space where there flows CSF, and the red lines are the pia mater.

2.1.1 Mechanisms of head injuries

The force that the head experiences at impact or shaking is crucial for predicting the nature of injury. This force follows Newton's second law, $F = ma$ (where F is force, m is mass, and a is the acceleration). Therefore, one could only consider the acceleration. The resulting acceleration field in the head is dependent on the direction of the impact force and the impact position, or in the shaking case – the shape and frequency of the shaking motion. If the impact direction does not point through the head center of gravity (CG), the impact will cause angular acceleration in addition to a *linear acceleration* measured at the CG. The linear acceleration is

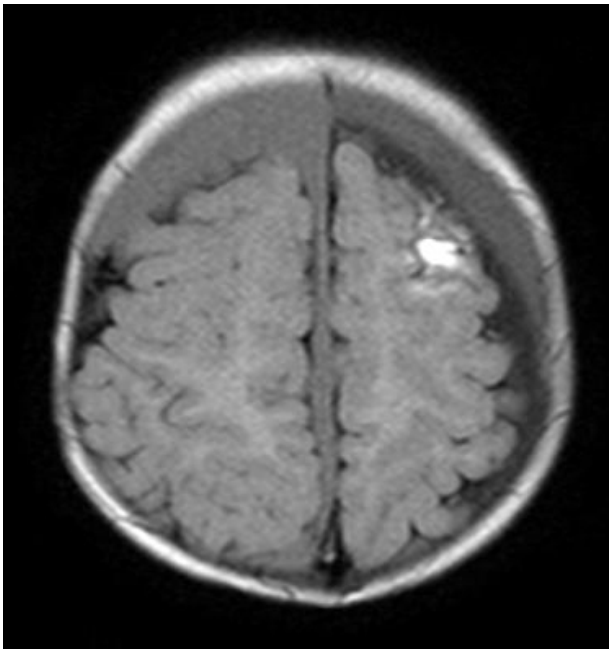

Figure 2.1.3: MR-scan of a 2 months old boy showing subdural hematoma and a contusion injury in the brain. The father told the police he had accidentally lost grip of the baby, resulting in a free fall from > 1,2 meters onto a concrete floor. The medical experts stated that the injuries may have been caused by violent shaking, but a fall could not be ruled out. Further prosecution was dropped. The boy made a full recovery.

defined to be a pure translational acceleration, where the acceleration field has the same magnitude and direction at all points.

If the impact direction points exactly through the CG, the acceleration will only be linear, and the motion will be a pure translation in the beginning. The head's connection with the neck and spine will after a short amount of time cause some angular acceleration and angular velocity in the head.

It is the acceleration field that is injurious to the head [92, 37, 38]. The HIC (Head Injury Criterion) is a number that describes the severity of impact. The HIC is defined as a weighted mean of the integral over the *linear* impact acceleration peak [92]. The purpose with the HIC is to have a number that corresponds well with the level of brain injury. Unfortunately the HIC is not a perfect indicator on the real head injuries. Therefore one needs to develop the HIC to consist not only of the linear acceleration integral.

The angular acceleration is more injurious than the linear acceleration [41] [46]. This is because the shear modulus of the brain is rather low (see Section 4.4.1), and the brain tissue shear will be larger with an angular acceleration than

with a linear acceleration of the same magnitude (see Section 6.1). Kleiven [46] found that the peak change in rotational velocity together with the Head Injury Criterion (HIC) [92] was a significant predictor of injury, and much better predictor than only the HIC. Morison and Minns [67, p. 142] computed numerical simulations of the Shaken Baby Syndrome, and found that "the rotational component of the shaking motion is responsible for approximately 93 % of the bridging vein strain". The rest 7 % were caused by the linear acceleration. Therefore it will be interesting to find the magnitude of the angular acceleration at our shaking peak acceleration (see Section 5.5).

The acceleration results in strains and stresses (see Section 4.1) inside the brain, causing injuries to the nerve cell. It also results in a relative displacement between the skull and the brain that causes bleedings at the brain/skull boundary. The bleedings arising from impacts differ from those in SBS as we will see below.

The time duration of the acceleration peak is also of great importance. For impacts it is around 20 ms, and for shaking it is in the range of 100-170 ms (see Section 5.2). The longer the acceleration duration, the longer time the brain has to move relatively to the skull, augmenting the risk of bleedings. Therefore bleedings at the brain/skull boundary can occur at lower acceleration magnitudes in shaking than in impacts [66, 79].

There are also some other factors that influence the level of head injuries due to certain kinds of impact or shaking [63, p. 58]. We will not describe them here, but refer to [63] for a thorough discussion.

2.1.2 Natural coordinates

We introduce a natural anatomically derived coordinate system of the head. The origo is in the accelerometer bracket (see Section 2.7), and the axes go through the accelerometer positions. The x-axis is in the anteroposterior direction (back-front), the z-axis goes straight up in the inferior-superior direction, and the y-axis is directed from the right to the left side [91].

One often name the x-z-plane: the *sagittal* plane; and the y-z plane: the *coronal* plane.

The *vertex* is the top point of the head. We placed an accelerometer at the vertex measuring in the anteroposterior (x-axis) direction. The coordinate axes can be seen in figure 2.1.4.

2.2 Shaken Baby Syndrome (SBS)

In the US, approximately 1000 children < 3 years of age are annually admitted to the hospital with SBS, resulting in more or less 200 fatalities [45]. Based

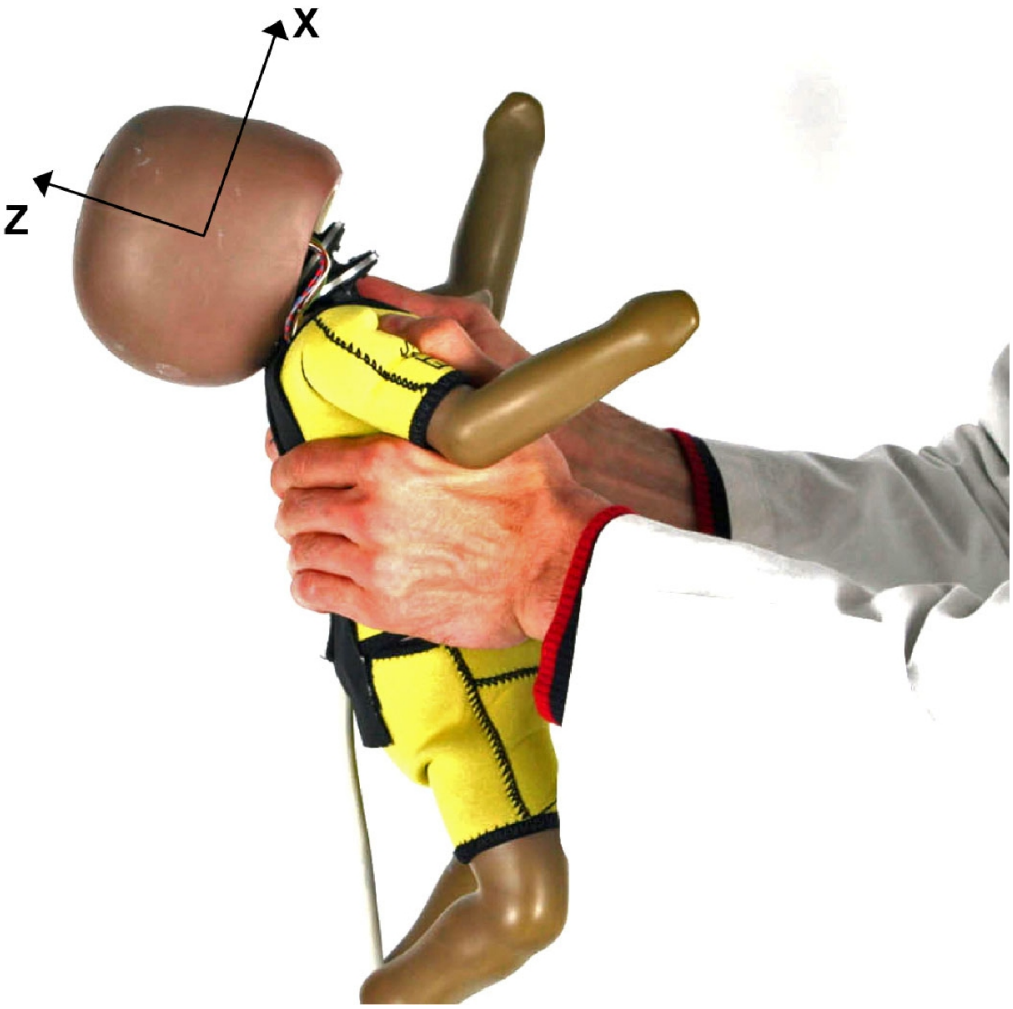

Figure 2.1.4: Shaking the crash test dummy. The coordinate axes x and z are drawn in the picture.

on clinical data from the Institute of Forensic Medicine in Oslo the incidence in Norway is 12-15 infants, with 2-3 (20 %) fatalities annually. Approximately 40-50 % of the survivors sustain severe long-term injuries such as paralysis, blindness and cognitive deficits, whereas 20-30 % survive seemingly unaffected.

The characteristic clinical findings in infants with SBS are *intra-cranial subdural hemorrhage* (bleedings) between the hemispheres (see Figure 2.1.3), *retinal hemorrhages*, and *diffuse axonal injury (DAI)* in the brain tissue. Subdural hemorrhage is explained above (see Section 2.1.1), and the other symptoms will be explained below in this section. Although external signs of abuse may not

be present, 50-90% of SBS patients present with bruises and fractures indicating blunt force abusive injuries [25]. Vigorous shaking in the anteroposterior direction is believed to create shearing forces in the brain causing diffuse axonal injury and brain swelling [86, 63], and relative displacement between the skull and brain causing tearing of bridging veins with subsequent haemorrhage.

Infants with head injuries do not necessarily show signs of direct impact to the head, i.e. bruise marks and fractures. Nevertheless, they present with the SBS symptoms. These findings along with the sometimes dubious testimonies of perpetrators represent clinical evidence that violent shaking is likely to cause such injuries.

Retinal bleeding is observed in up to 50-100 % of allegedly shaken infants [25]. Acceleration-deceleration forces creating shearing of the blood vessels into the eye or a sudden rise in intracranial and intravascular pressure has been proposed as a reason. The eye apple is loosely located in the eye orbita, able to move freely, and attached to the brain through the optical nerve. Typical widespread retinal bleedings are almost exclusively seen in SBS, but retinal haemorrhages can be found in children sustaining accidental head injuries involving severe deceleration/rotation of the head, such as car crashes [15].

Diffuse axonal injury (DAI) is typically seen in frontal car collisions with major whiplash acceleration-deceleration to the head.

2.3 Earlier studies on the Shaken Baby Syndrome

There are several important questions to answer for understanding the origins of SBS. Some of them are

1. What are the differences between injuries caused by shaking or by head impacts? Is shaking actually injurious at all? One would have to compute the displacement at the brain skull boundary, and the shear strain inside the brain and brainstem, to find these answers. It is also of great importance to do further medical research on SBS patients. The displacement at the brain/skull boundary is crucial for determining the risk for subdural and retinal hemorrhages.

2. How are the biomechanics of the baby head motion during shaking – the peak acceleration magnitude, the duration of the acceleration peak, the motion directions, and the position of the rotation axis?

3. How does one measure and compute 3D acceleration for baby dummy heads with the limited amount of interior space? The 3D acceleration is needed for describing the 3D motion completely.

4. Which elastic and numerical models are best suited for simulating and making accurate computer animations of SBS?

A lot of research has been done to reveal some answers to these questions. Some of the biomechanical, numerical, and 3D acceleration studies are shortly reviewed below (see Sections 2.3.1, 2.3.2 and 2.4). For the medical research and a thorough review of most questions and problems with SBS, see [63].

2.3.1 Biomechanical studies

Biomechanical studies have this far failed to reconstruct with plausibility the injury mechanisms of SBS [26, 19, 73, 97]. Critics are questioning the evidence for shaking to cause internal head injuries and propose that intra-cranial haemorrhages in infants without evidence of impact more likely are birth-related or due to pathological conditions [63, 72]. Cory and Jones [19] propose that the head hits the baby torso during shaking, and that the injuries arises from this combination of shaking and impacts (see Section 2.3.1). Minns and Brown [64], and Morison and Minns [67, p. 142 ff.] propose several indications that the SBS symptoms really are caused by shaking.

Only a few biomechanical models of shaking a dummy have been studied [26, 19, 73, 97]. These studies have used dummies with necks possessing one-way hinge joints with no resistance, or short rubber necks. The studies have investigated important properties of the shaking motion. Our biomechanical experiments have profited a lot from these earlier studies, and in our research we try to answer some unsolved questions on the SBS biomechanics.

Both the angular and linear acceleration, in addition to the angular velocity, are needed for computing 3D acceleration in traditional ways, and thus validating the acceleration measurements (see Section 2.4 and Equation (2.4.1)). In Duhaime et al. [26] and Wolfson et al. [97] they only measured the linear acceleration at one point, and did not measure the angular velocity. Cory and Jones [19] measured the linear acceleration at one point in the torso and one point at the vertex of the head, but did not measure the angular velocity. Prange et al. [73] only measured the head angular acceleration around the y-axis in one point, and calculated the angular acceleration from these data.

None of these studies could therefore compute 3D acceleration, and verify the accurateness of their data. Nevertheless, these basic experiments have concluded that shaking alone is unlikely to result in the typical SBS injuries. The authors of the studies argue that the angular acceleration in their experiments is much lower from shaking than from impact.

Duhaime et al. [26] compared different neck constructions of a simple dummy with non-weighted arms and legs. The neck constructions they used were a hinge

neck, and two rubber necks – one thick and one narrow. They only measured the vertex acceleration in the x-axis direction, and shook it in the anteroposterior plane. Their results showed that the rubber necks gave a much lower peak acceleration than the hinge neck. One explanation of this behaviour may be that the rubber necks also accelerated in other directions than in the x-direction.

Cory and Jones [19] studied dummy shaking where they varied the material properties of the dummy. They copied the Duhaime et al. experiments, and tried to find which material properties and shaking patterns, that maximized the peak acceleration. In the end they combined all the worst case properties into one dummy, and reached a high acceleration magnitude. They observed that the dummy head hit the torso of the dummy in these worst cases. They therefore proposed a new explanation and name of the mechanics of shaking – the shaking-impact syndrome. The head-torso impacts were achieved by shaking the dummy down towards the floor and up again.

Prange et al. [73] compared falls with shaking. They used a hinge neck without resistance and limited to motion only in the sagittal plane, and proposed that this construction modeled a worst case. The dummy was shaken back and forth, as in the Duhaime et al. study [26]. They calculated the peak angular acceleration from the angular velocity measurements, and concluded that shaking produced lower peak accelerations.

Wolfson et al. [97] developed a dummy torso with a linear accelerometer attached to the torso. The dummy had no head or neck. Wolfson et al. numerically simulated different neck constructions with parameters varying from a pure hinge neck to a pure two-hinge neck, with different neck stiffnesses in between (a Rigid Body Modeling (RBM)). They arrived at much lower velocities than Duhaime et al. [26] even if they used similar accelerations. The main explanation may be that their RBM model is too simplified, and that one really needs a head on the dummy for more realistic measurements.

Bandak [6] used previously published data on shaking, and calculated the forces acting on the infant neck. His results showed that "the forces far exceeded the limits for structural failure of the cervical spine" [6, p. 71]. Rangarajan and Shams [76] questioned his methodology. They had computed much lower forces with use of the same acceleration data as Bandak. Unfortunately, Bandak had integrated the acceleration over the whole shaking period to reach his high levels of force [5], and did not consider the peak force values.

Furthermore there exist some biomechanical experiments by Jenny et al. [44]. Since we have not been able to obtain the article, we can not review it here.

2.3.2 Numerical simulations of the Shaken Baby Syndrome

Computations and simulations are widely used among engineers, in construction processes and for simulating mechanical and elastic processes. A new research field has appeared in recent decades where one uses engineering methods for simulating body processes in medicine. These computer simulations are needed for a thorough understanding of biomedical processes, for developing new treatments, and for investigations that are impossible to make on a real body due to ethical reasons.

Computer based finite element simulations (FE) of head injuries have developed a lot the last years. Several different three-dimensional finite element models of the adult head exist, for example [14, 47, 94, 49, 98], but only a few 3D models of the child head have been created [79, 61, 52, 65, 22]. Prange and Kiralyfalvi [74], and Couper and Albermani [20] have investigated 2D-slices of the infant head.

Couper and Albermani [20], and Morison [66] simulated SBS with an advanced way of modeling the CSF as a fluid. Roth et al. [79] simulated SBS with solid finite elements for the CSF (see Section 4.7.1 for an explanation of finite elements). Roth et al. and Morison performed 3D simulations, while Couper and Albermani only worked in 2D.

The finite element (FE) simulations made by Roth et al. [79, p.227] using the loading conditions from [73] "show that vigorous shaking can have the same consequence as an impact in terms of subdural bleeding". This is a strong indication that it is too simple to only consider the peak acceleration when concluding if shaking is dangerous or not. It is also necessary to take into account the relative displacement between the skull and brain developed during a whole shaking peak. Roth et al. concluded that this displacement is the same at impact as at shaking, but the time duration of the movement is different.

Morison [66] achieved the same conclusion, even if he only used the low acceleration values from Duhaime et al. [26]. He also found that the bridging veins are especially vulnerable to shaking frequencies in the range of 2 to 5 Hz. These frequencies are typical for all biomechanical SBS experiments.

Couper and Albermani [20] varied the volume of cerebrospinal fluid (CSF) in their 2D mesh, and found that it affected the brain/skull relative displacement a lot. They therefore mean that the CSF volume must be considered while diagnosing SBS.

None of the SBS FE-simulations considered the brain compressibility during longer time periods of acceleration (see Section 4.4.2).

2.4 A review of different ways of computing 3D acceleration

In this section we present some earlier methods of computing a 3D acceleration field, and investigate properties of the methods important for our shaking simulations. One important characteristic of the SBS motion is changing angular acceleration direction, which will blow up the error of the measurements while the angular acceleration changes direction. Another problem is the very small cavity in the baby dummy head. These problems are not solved in the earlier methods, and is a good motivation for our new 3D acceleration investigations developed in Sections 3.3 and 3.4.

It is important to compute 3D acceleration. It simplifies the validation of external accelerometer measurements, like the vertex measurements, and gives the possibility of describing the 3D motion with use of the integrated angular velocity [23, p. 2-3 ff.] (see Section 3.1).

Essential for the possibility of computing 3D acceleration is the number and position of the accelerometers. The most common and well proven accelerometer configuration is the Nine Accelerometer Package (NAP). The NAP was developed in 1975 by Padgaonkar et al. [70]. It uses nine uniaxial accelerometers in a 3-2-2-2 configuration as shown in Figure 2.4.1.

The governing formula for the NAP is

$$A_p = \ddot{H} + \omega \times (\omega \times \rho_p) + \dot{\omega} \times \rho_p \tag{2.4.1}$$

where A_p is the linear 3D acceleration at the point p in world space coordinates, $\ddot{H}$ is the linear acceleration at the body origo in world space coordinates, ω is the angular velocity vector around the body coordinate axes, $\dot{\omega}$ is the angular acceleration (often called α), and ρ_p is the body coordinates for the point p. The formula is derived in [32, p. 35], and is based on the differentiation of the formula $h = H + \rho_p$ where h is the position of p in world space coordinates, and H is the position of the body origo in world space coordinates.

The NAP is developed to avoid solving the non-linear ODE that arises in (2.4.1). It measures linear $\ddot{H}$ and angular $\dot{\omega}$ acceleration with use of the difference between the NAP origo acceleration and the outer accelerometers. One also needs an integration to compute ω, and thus achieves 3D acceleration computations since the point p is arbitrary. The equations of the NAP are the following:

$$\dot{\omega}_x = (A_{z1} - A_{z0})/2\rho_{y1} - (A_{y3} - A_{y0})/2\rho_{z3} \tag{2.4.2}$$

$$\dot{\omega}_y = (A_{x3} - A_{x0})/2\rho_{z3} - (A_{z2} - A_{z0})/2\rho_{x2} \tag{2.4.3}$$

$$\dot{\omega}_z = (A_{y2} - A_{y0})/2\rho_{x2} - (A_{x1} - A_{x0})/2\rho_{y1} \tag{2.4.4}$$

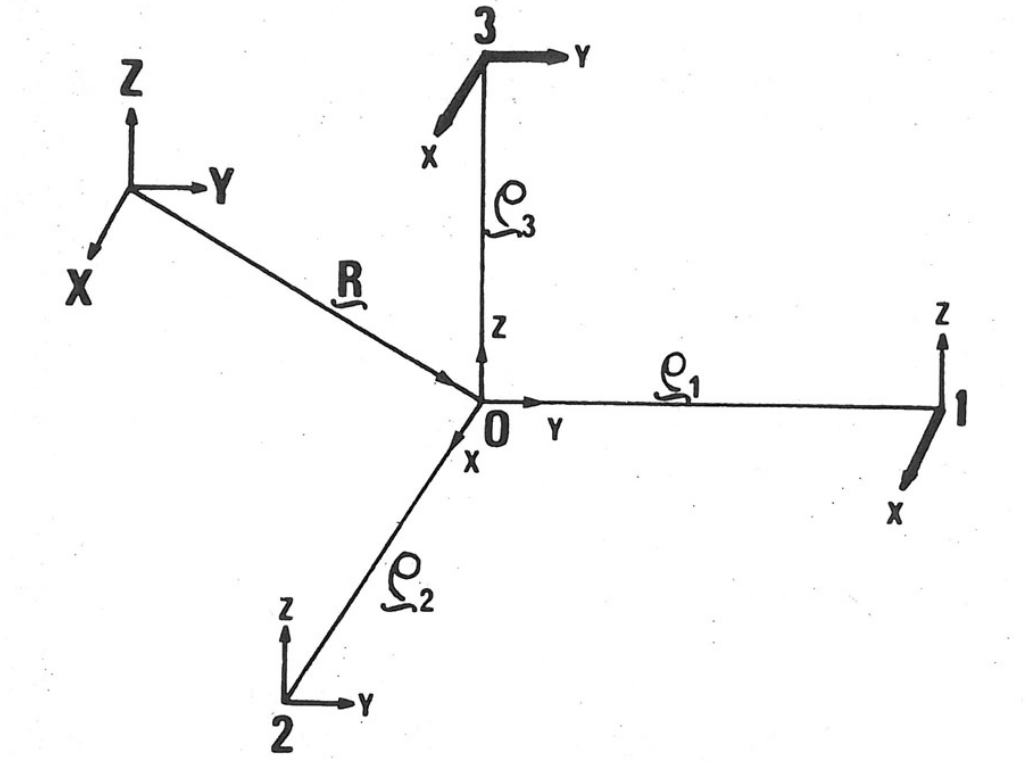

Figure 2.4.1: The nine accelerometers at points 0, 1, 2 and 3 are in the NAP configuration. The three accelerometers at 0 in addition to the three boldfaced arrows at points 1 and 3 are in the 3-2-1 configuration. The figure is printed in [70].

The stability, accuracy, body-to-world-space coordinate transformation, and other aspects of the NAP are further investigated by Plank et al. [71] and DiMasi [23].

It is important that 3D acceleration computations are accurate, do not accumulate error, and are stable. Unfortunately the NAP shows a large error when the angular acceleration changes sign [70, p. 555]. This is of importance in the shaking motion since it is cyclic. The cause of the large error may be that the NAP configuration deals with differences between accelerometers, which are very small near the sign change. The integration of the angular acceleration also needs good initial conditions which may be a problem.

There exist other methods for computing 3D acceleration. They are mostly based on the same equation as the NAP (2.4.1). The most recent methods often use an optimization process to compensate for the noise in the measurements.

Before the NAP was developed one often used a six-accelerometer configuration 3-2-1 (see Figure 2.4.1). Thus one also computed with linear acceleration differences, and needed a relatively large distance between the sensors for minimizing the noise. Unfortunately the configuration resulted in three differential equations (2.4.5) to (2.4.7), in which the numerical solution process accumulated error which severely affected the accuracy of the computed angular accelerations. These equations were in fact unstable [56].

$$\dot{\omega}_x = (A_{z1} - A_{z0})/\rho_{y1} - \omega_y\omega_z \tag{2.4.5}$$
$$\dot{\omega}_y = -(A_{z2} - A_{z0})/\rho_{x2} + \omega_x\omega_z \tag{2.4.6}$$
$$\dot{\omega}_z = (A_{y2} - A_{y0})/\rho_{x2} - \omega_x\omega_y \tag{2.4.7}$$

Seemann and Lustick [84] worked further with the 3-2-1 pattern (see Figure 2.4.1). They developed a method of continuously comparing their integrated values against photo-derived motion, to compensate for the accumulated error. This is impossible to use in our case since it is too time demanding.

Martin et al. [59] developed three configurations where they let the angular velocity components in (2.4.5) to (2.4.7) be measured by a 3-axis gyroscope. They only needed three linear accelerometers in addition to the gyroscope, but additional accelerometers made it possible to reduce the noise with an optimization. In their computations they needed to differentiate the angular velocity to reach the angular acceleration needed in formula (2.4.1) for computing 3D acceleration. In the differentiation process the noise in the angular velocity measurements blew up, and gave unphysical acceleration peaks. They needed to filter their computed results for getting a result nice enough for comparing it with the NAP but in the filtering process the real peak acceleration values were filtered away.

Chu et al. [17] developed a new iterative optimization algorithm based on the exterior 2-2-2 pairs of accelerometers from the NAP configuration (see Figure 2.4.1). Since the position vectors of all accelerometers is orthogonal to the corresponding measuring directions, Chu et al. proposed that they could neglect the centripetal acceleration in their model. They could compute the linear acceleration in their body origo, in addition to the angular acceleration around the origo. However, for computing a 3D acceleration field they would also have needed some information on the velocity, since the angular velocity results in centripetal accelerations. The accelerometer configuration by Chu et al. is used in Greenwald et al. [35] where they placed the accelerometers in helmets of American football players. It was perhaps possible for Greenwald et al. to integrate the angular acceleration and compute the angular velocity, since the initial velocity and impact direction could be derived from video recordings.

In our current accelerometer configuration in Section 2.7, and in our new theoretically constructed accelerometer configuration in Section 3.4, we do not use the

difference between sensors as input in the algorithm, but use the measured values directly. This is of importance for our shaking problem with changing angular velocity and small dummy head cavities.

Neither of the above presented methods compensate for gravity. Gravity is also present as a part of the acceleration measurements, and one often does not know the direction of the gravity vector. So if one computes all six motion degrees of freedom with use of the acceleration measurements, one will get a motion like if the shaker of the baby fell freely towards the earth due to gravity. Of course the gravity part of the acceleration field actually affects the brain (though very little), and it augments the downward velocity. However, it should be subtracted from the acceleration field if one wants to compute the world space coordinates of the shaking motion.

Here follows a summary of the main problems with the earlier 3D acceleration configurations:

3-2-1 Results in an unstable ODE.

NAP Too large for our small head cavity since it is based on differences between the accelerometers. By the same reason it gives a large error when the angular acceleration changes sign.

Seemann/Lustick 3-2-1 Too time demanding, and difficult in 3D motions.

Martin et al. Promising, suites small spaces, but filters away the peak accelerations.

Chu et al. Possible to compute angular acceleration, but one needs additional data on the initial velocity field.

2.5 Impact experiments on post-mortem human subjects

We need some experimental human acceleration data for being able to calibrate our elasticity simulations to real injuries (see Section 4.6.1). Due to ethical and practical reasons it is difficult to measure acceleration of human brain tissue. Only a few investigations on post-mortem human subjects have been done. The most important of these are Hardy et al.'s [37, 38] implantations of neutral density accelerometers in combination with high-speed biplane x-rays to collect data from both the motion and the acceleration. Hardy et al. also studied the brain motion with respect to the skull.

The Hardy et al. tests are performed on post-mortem human heads, which are exposed to several impacts. Each head has got a number, C followed by three digits, and is called a "sample". Hardy et al. hit each sample several times, and record the 3D acceleration with a NAP. The impact tests are numbered T1 to T5, so each test name follows the pattern C379-T4 and so on.

Inside the brain Hardy et al. have implantation two clusters of Neutral Density Targets (NDT). The NDT are visible on x-ray-photos, and the motion of the NDT reveal the displacement. In Hardy et al. (2001) they were not able to compute the strain from the data, but Zou later did that in a simple model [101]. In Hardy et al. (2007) they have computed the strain in the middle of each NDT cluster. These strain values are what we try to reproduce in our elasticity computations for calibrating and validating our model.

In sample C408 from Hardy et al. (2007) the NDT are placed on the surface of the cortex. Sample C408 is therefore especially interesting since the position of the NDT make it possible to carefully measure the relative motion at the brain/skull boundary. The shear strain computed in the C408 sample is a measure of the shear strain close to the brain surface, since Hardy et al. computed the strain at the NDT positions. We compare our computed brain/skull boundary shear strain with the C408 shear strain in Section 6.3.

In all other (2007)-samples the NDT clusters are placed inside the brain. In Hardy et al. (2001) some of the NDT are placed near the surface of the cortex, the rest inside the brain.

2.6 The background for our research project

Before our work with this thesis started, the Forensic Institute of Medicine at University of Oslo had already done shaking experiments with a crash test baby dummy (see Figure 2.6.1). They had constructed and built an accelerometer bracket which they placed at the center of gravity in the dummy's head (see Section 2.7 for the construction). They let 15 volunteers shake the dummy freely in a back and forward motion. The neck of the dummy was damaged during our peak acceleration test, so further experiments with the dummy were not possible.

We discovered in an early stage of the thesis work that the accelerometer positions on the accelerometer bracket were non-optimal, and that it was impossible to compute 3D acceleration from our measurements to validate the vertex acceleration measurements and find the 3D motion (see Section 2.7). All previously developed accelerometer configurations demand a too large space for our purpose, or give a larger error at large angular velocities (see Section 2.4).

Figure 2.6.1: The crash test dummy with the acceleration results.

2.7 Design of the crash test dummy and the accelerometer bracket

We use a standard Q0 crash test dummy from First Technology Safety Systems [90] that represents a 6 weeks old baby. The Q0 dummy is primarily used in automotive passenger safety testing, and we stress that the dummy has not been developed for repeated rigorous head shaking from the manufacturer perspective.

The neck consists of two intermediate disks and three rubber sections to simulate both some of the joints and the elasticity (see Figure 2.7.3). It is a scaled version of an adults neck, since the real elasticity data from a child's neck are not known. This is of course a weakness in the model, since a baby does not have as strong neck ligaments as an adult. The neck allows the rotation axis to move continually, which we have seen is important for simulating the fast whiplash motion at the acceleration peaks in the shaking motion. We believe it is of great importance that the neck model allows the rotation axis to move in a natural way. There are some small holes through the front rubber section of the neck. These holes enables the neck to move more easily in the sagittal plane than in the coronal.

The center of the head contains an accelerometer bracket that also serves as a load for obtaining the natural weight of the head. The bracket was rebuilt to place

six uniaxial accelerometers onto it, one on each side of the bracket (see Figures 2.7.3 and 2.7.2). At the vertex we placed a seventh accelerometer, with a sensing axis in the x-direction.

The sensors that measure in the z-direction are placed onto the horizontal sides of the bracket; the y-sensors onto the remaining sides orthogonal to the y-axis; and the x-sensors onto the last sides; one sensor at each side. All sensors measure in the direction orthogonal to the flat side of the accelerometer. The x_1 sensor is placed at the back side of the bracket, and the x_2 sensor at the front side; the y_1 sensor at the right side, and the y_2 sensor at the left side; the z_1 sensor at the upper side, and the z_2 sensor at the lowest side of the bracket.

We choose a coordinate system with the x-y-plane through all the x-y-sensors, the x-axis pointing forwards through the x-sensors to the nose, and the y-axis from the right to the left side through the y-sensors. The z-axis is orthogonal to these in the upwards direction. This coordinate system is coinciding with the anatomically natural coordinates developed in Section 2.1.2. The origo is called the *body origo*, and the coordinates *body coordinates* (see Figure 2.1.4).

It is easily seen from the positions of the accelerometers that it is impossible to solve for 3D acceleration (see Figure 2.7.1). If we study a pure rotation around the body origo, the sensors will only measure the centripetal acceleration, and will not keep information on the direction of rotation or the magnitude of the angular acceleration. It is impossible to change the coordinate system without at the same time changing the sensing axes of the uniaxial accelerometers since they always in the acceleration formulas have to be parallel to the coordinate axes (see equations (2.4.1) and (3.4.2)). The acceleration vector on the left hand side is divided into parts parallel to the coordinate axes. The accelerometer measurements are inserted at these left hand side acceleration values, so our uniaxial measurements will always be treated as if they are in the same directions as the coordinate axes.

One possibility to get around this problem is to use triaxial accelerometers. Then one could project the measured acceleration onto a new coordinate system better suited for revealing both the centripetal- and the angular acceleration.

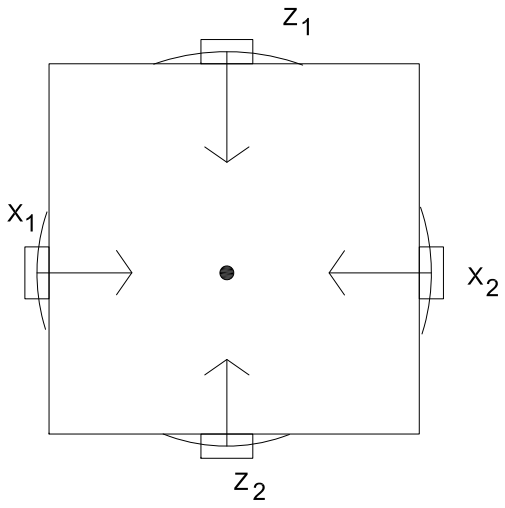

Figure 2.7.1: The accelerometers in the x-z-plane measure only the centripetal acceleration in this case. The rotation is around the black dot, and the sensing axes of the accelerometers are in the direction of the centripetal acceleration only. They can therefore not measure the directions of the rotational acceleration, and it is impossible to calculate the 3D acceleration field from the measurements.

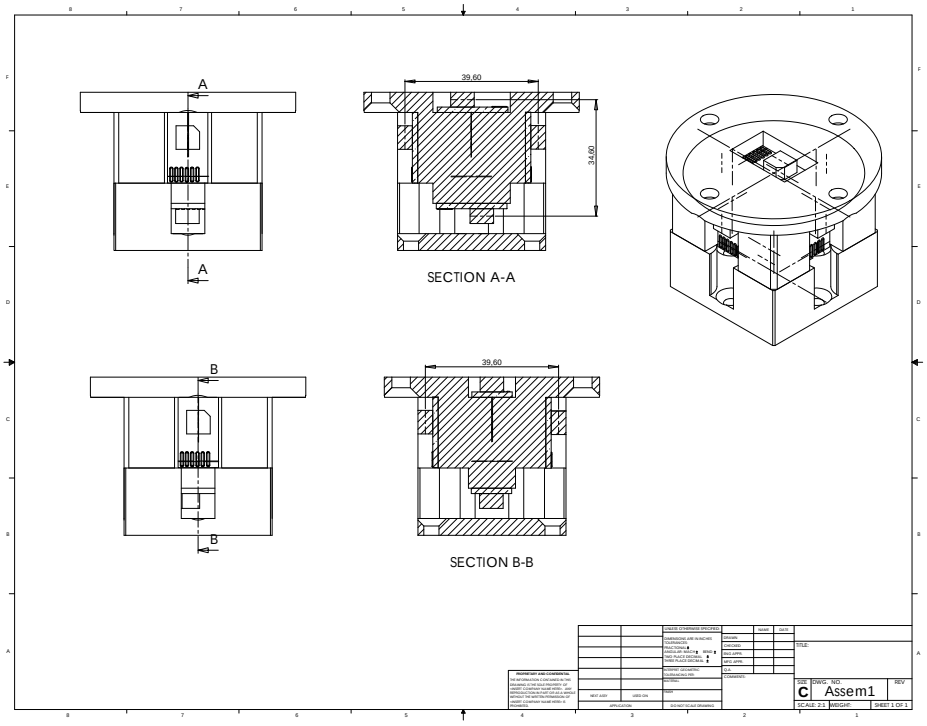

Figure 2.7.2: Drawings of our accelerometer bracket.

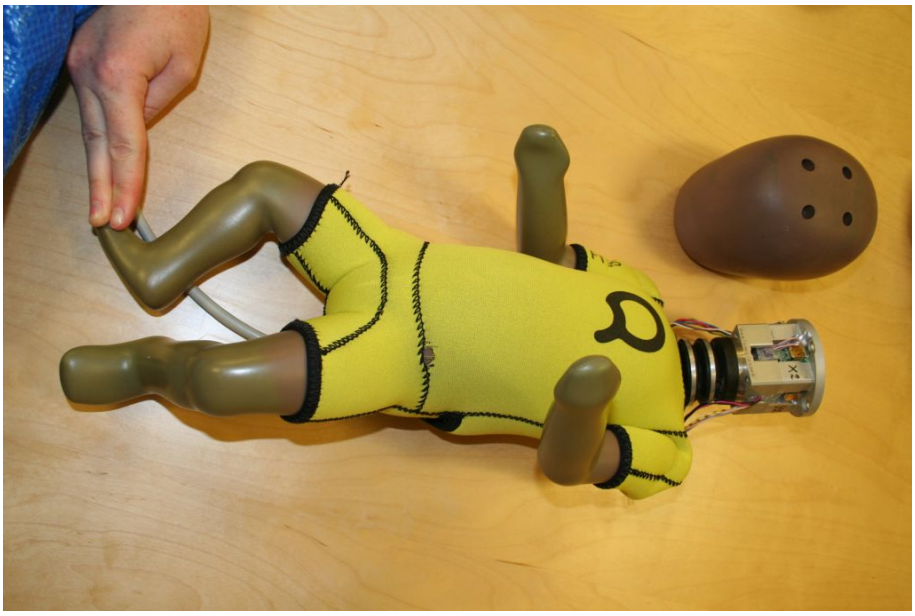

Figure 2.7.3: The crash test dummy with the accelerometer bracket.

Chapter 3

Acceleration methods

3.1 Introduction

In this chapter we will work with the thesis topics 2 and 3 from Section 1.2. We need to validate our accelerometer measurements, develop new ways of computing 3D acceleration suited for small spaces and changing angular acceleration and velocity directions.

Acceleration measurements and analysis are the basis for head injury research. In our investigations we need 3D computations of acceleration. We need to verify that the accelerometers measure the right values. Since human volunteers shake the dummy, every shaking motion differs, and the accelerometers measure different values at each shaking experiment. Thus we can not change the accelerometers between the shakings and hope that they show the same values.

The solution to this problem is to use some of the accelerometers for computing the 3D acceleration field. Then the acceleration at the positions of the remaining accelerometers can be computed and compared with the measured values for a validation.

We also need the 3D motion values for a thorough investigation of the shaking motion behavior. The shaking experiments are recorded on digital video, but the time steps in the video are too large for the possibility of recording the detailed motion at the acceleration peaks.

So we have to compute the 3D *rigid motion* of the skull (see Section 3.2). Usually one computes the 3D acceleration, and uses this to describe the 3D motion with the help of some integration and some conversion computations from body-coordinates to world space coordinates (see Section 2.4).

Unfortunately, the positions of our accelerometers are not optimal for good 3D acceleration computations (see Section 2.7), and due to limitations of space

in the dummy head we can not use the standard method (NAP) for computing 3D acceleration (see Section 2.4). Therefore we study a new theoretical framework, and try to develop an accelerometer configuration and an algorithm that will work better for our case (see Sections 3.3 and 3.4).

3.2 Rigid motion mechanics

In this section we will explain the concepts of a rigid motion. The rotation axis is defined since it will be of importance for our validation of the vertex measurements in Section 3.5. We also explain where the motion, velocity and acceleration max and min occur. This is important for our integrations in Section 3.6. We finish the section with a description of the rotation matrix R.

3.2.1 Rigid motion

The 3D motion of a body is a combination of displacement due to deformation, and displacement due to rigid motion. A *rigid body* shows no deformation, and all inner distances are preserved during motion, and thus the body only moves according to a rigid motion. The brain is clearly no rigid body, but the acceleration bracket of the dummy head is rigid. Therefore, the acceleration measurements follow the laws of a rigid motion.

A rigid motion is a sum of a translation and a rotation:

$$x_p = Rp + b \tag{3.2.1}$$

where p is the coordinate vector before the motion in the body coordinates, and x_p are the new coordinates in a global coordinate system which extends the before-motion body coordinates to the space surrounding the body. We define this coordinate system in 3D as $\mathbb{R}^3_b$, and in 2D as $\mathbb{R}^2_b$. R is a *special orthogonal matrix*, i.e. an orthogonal matrix with determinant +1. R rotates the body around the origo of the body coordinates [77], [27, p. 6]. Furthermore, b is the translation vector that moves the origo of the body coordinates (the *body origo*).

The motion of the dummy head in shaking is a continuous time dependent rigid motion. However, the discrete acceleration measurements subdivide this continuous motion into 5000 finite rigid motions each second. In order to compute the time dependent rigid motion of the head, we need to sum all these finite rigid motions somehow. This summation process is rather complex, and is described in Section 3.3. In the subsections below we only study a finite rigid motion for simplicity.

If we can compute the head rigid motion from each time step, we can apply the motions to the skull as boundary conditions in an elasticity Finite Element (FE) solver (see Section 4.7.1 for an explanation of the Finite Element method).

3.2.2 Rotation axis

The matrix R rotates the body around the body origo. Most often in a rigid motion, the *rotation axis* "experienced" by the body is not through the body origo. One special case of this situation is shown in figure 3.2.1.

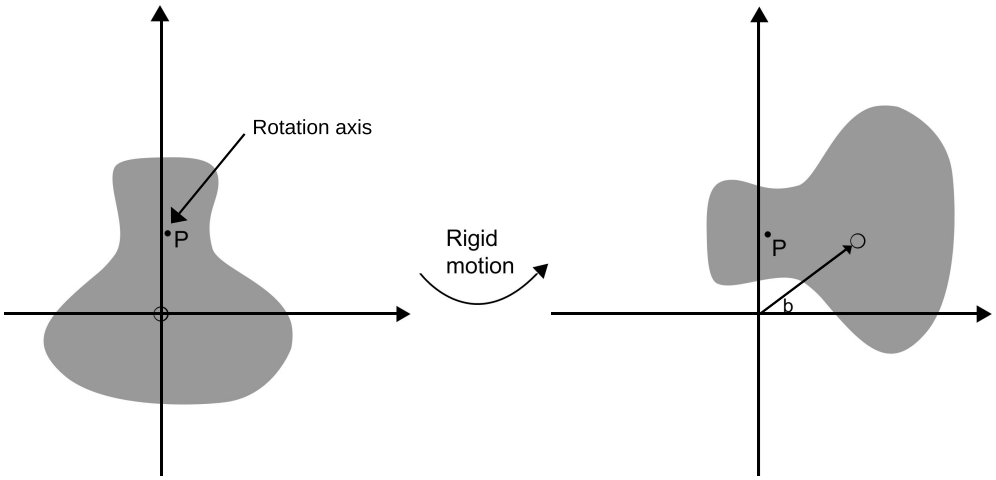

Figure 3.2.1: A rigid motion with P as fix point (rotation axis), and b as translation vector.

In the figure the body actually rotates around a fix point in the smaller body part. We call the perpendicular line through that point for *rotation axis* (Arnold [1, p.126] called it " instantaneous axis of rotation"). This is defined in 2D as the line through the fix point p of the rigid motion (3.2.1) which is orthogonal to the plane of the motion, i.e. we solve the equation $p = Rp + b$ for p, with R and b known. The equation $p = Rp + b$ is equivalent to $(I - R)p = b$. We investigate the determinant of the matrix $(I - R)$ in 2D.

$$\det(I-R) = \det \begin{pmatrix} 1-\cos\theta & \sin\theta \\ -\sin\theta & 1-\cos\theta \end{pmatrix} = 2(1-\cos\theta) \tag{3.2.2}$$

where θ is the angle of the rotation. The determinant in (3.2.2) is clearly non-zero everywhere, except when $\theta = n \cdot 2\pi$, which is equivalent to $R = I$. Since we always have small rigid motions at each time step, the fix point exists if $R \neq I$. The fix point is expressed in $\mathbb{R}^2_b$ (2D) coordinates. The actual rigid motion around

the fix point is easily seen to be a pure rotation since it is a rigid motion, and no translation is experienced by p.

The rotation axis is instantaneous in the sense that it may move with time (a function of time), and that it only exists for each moment. Around the rotation axis all internal body rotation actually occurs, and the centripetal acceleration field $a_c = r\omega^2$ is always directed towards the rotation axis since the angular velocity ω is caused by the changes in body rotations. It is important to understand that the rotation axis is seldom through the body origo, is always measured in $\mathbb{R}^3_b$ or $\mathbb{R}^2_b$ coordinates, and may be placed outside the dummy. It is of course possible to express the rotation axis in world space coordinates, but it is not interesting for us here.

In our shaking experiments we can see the rotation axis move from a place down in the body to a position in the lower part of the brain at the acceleration peaks (see Section 5.6).

Even if the velocity at the rotation axis is zero, the acceleration needs not be zero. It is the acceleration that drives the movement of the rotation axis through time. The rotational acceleration field of a rigid body may also have positions where the acceleration is zero. We define this point in 2D to be the *rotational acceleration center*, also measured in $\mathbb{R}^2_b$ coordinates. One can move this point by adding a rotational or linear acceleration. Sometimes the rotational acceleration center may be placed outside the body. Some Figures of the rotational acceleration center motion are 4.6.2 and 4.6.1.

3.2.3 Correspondence between the velocity, the motion, and the acceleration curves of the shaking motion

There is a kind of phase shift between the acceleration and the velocity which is similar to the simple harmonic motion of a linear spring. The solutions to the spring motion consists of a sum of two trigonometric functions. Our shaking motion is also periodic (but driven by a force) and may be expressed as a larger sum of trigonometric functions – sines and cosines. This is called the Fourier series. Since the acceleration and velocity are derivatives of the trigonometrically expressed motion, this means that the maximum length of a shaking cycle occurs at the maximum absolute acceleration values, that the turning point of the motion is at this maximum, and that the velocity is zero somewhere in the vicinity of this moment. The maximum velocity values will occur where the acceleration changes sign.

We use these facts when we choose initial values and the start- and stop points in our integrations (see Section 3.6).

3.2.4 The rotation matrix

There exist six degrees of freedom in a 3D rigid motion. Three of them come from the translation vector b, and the other three come from the possibility of rotation around all coordinate axes. The representation of the rotational degrees of freedom may be done in several ways. One possibility is to use the angles of the rotation around each body coordinate axis. These angles are called the Bryant angles and they result in a nasty formula with several products of trigonometric functions [96, p. 22] (equation (3.2.3)):

$$R = \begin{pmatrix} c_2c_3 & c_1s_3 + s_1s_2c_3 & s_1s_3 - c_1s_2c_3 \\ -c_2s_3 & c_1c_3 - s_1s_2s_3 & s_1c_3 + c_1s_2s_3 \\ s_2 & -s_1c_2 & c_1c_2 \end{pmatrix} \tag{3.2.3}$$

Where $c_i = \cos\phi_i$ and $s_i = \sin\phi_i$ with ϕ_i the Bryant angle around body coordinate axis number i.

There are two problems with this representation: if $\phi_2 = \pi/2 + n\pi (n = 0, 1 \ldots)$ it is impossible to solve for the other Bryant angles; and it results in complicated equations for solving for the three rotational degrees of freedom if one uses acceleration measurements. Then one needs to differentiate equation (3.2.3) twice. In our problem the rotations are always much smaller than $\pi/2$, but the second problem is harder to solve. So we need a better definition of the rotation matrix R.

We therefore study another definition of the rotation matrix R in Section 3.3 where we use the matrix exponential of a skew symmetric matrix (3 simple degrees of freedom without the use of trigonometric functions). It is much easier to compute the differentiation of the rigid motion with our method than with Bryant angles, and thus we describe the measured 3D acceleration field in direct terms of the rigid motion six-degrees of freedom in world space coordinates.

3.3 Method for describing the 3D acceleration with the matrix exponential

In this section we investigate a new formula for finding the six degrees of freedom of the 3D motion, with the help of accelerometer measurements. We want to find out if it results in easier computations which are more suited to the shaking problem with a small head cavity and changing angular velocity.

Since we have a rigid motion at each time step in our acceleration measurements, we have to continuously add the rigid motions to each other, to be able to express the rigid motion in world space coordinates as a function of time. Since the new coordinates after use of the rigid motion formula at each time step (equa-

tion 3.3.1) are always expressed in the body coordinates from before the motion at this time step ($\mathbb{R}^3_b$), we have to be careful when adding the motions.

We define x_{p_i} to be the new coordinates of body coordinate point p at time step i, R_i to be the rotation matrix, and b_i the translation vector, all at time step i and in $\mathbb{R}^3_b$. The rigid motion at time step i thus becomes

$$x_{p_i} = R_i p + b_i \tag{3.3.1}$$

Due to properties of matrix groups and vector groups in $\mathbb{R}^n$ we must use different operators while adding the rotation matrices and the translation vectors. We use matrix multiplication for adding the rotation matrices R_i, but must keep the order of the rotations since rotations are non commutative. We use ordinary addition for the translation vectors b_i , but must also remember to rotate the vector according to the previous rotation R_{i-1} before addition. Notice that $x_{p_i} \neq x_p(t_i)$, $R_i \neq R(t_i)$ and $b_i \neq d(t_i)$ in the following formulas. x_p, R and d are functions of time in *world space coordinates*. This makes it meaningful to talk about $x_p(t_i)$, $R(t_i)$ and $d(t_i)$. x_{p_i}, R_i and b_i are all expressed in *body coordinates* $\mathbb{R}^3_b$.

Thus we can easily get a full description of the whole rigid motion in *world space coordinates* if we know the rigid motion at each time step.

$$x_p(t_i) = R(t_i)p + d(t_i) \quad \Longleftrightarrow \tag{3.3.2}$$

$$x_p(t_i) = (R_i(R_{i-1}(R_{i-2}(\ldots R_1))))p + \left[\sum_{k=1}^{i}(R_{k-1}b_k)\right] \tag{3.3.3}$$

where $x_p(t)$ is expressed in world space coordinates $\mathbb{R}^3_b$, and $R_0 := I$.
$(R_i(R_{i-1}(R_{i-2}(\ldots R_1))))$ is often written as $\prod_{k=1}^{i} R_k$. Two rigid motions added can be seen in Figure 3.3.1.

Notice that we have only one choice of world space coordinate system, it has the same axes and origo as the body coordinate system from time step zero, i.e. the original position. The world space coordinate rigid motion $x_p(t)$ is differentiated twice, and measured by our accelerometers at their body coordinate positions.

For simplicity we work analytically with only one time step rigid motion from now on, and skip the time step notation:

$$x_p = Rp + b \tag{3.3.4}$$

where p are the coordinates before the motion, and x_p are the new coordinates. The rotation matrix R is a *special orthogonal matrix*, which means an orthogonal matrix with determinant +1. [77], [27, p. 6]. The group of special orthogonal matrices of dimension $n \times n$ is written $SO(n)$.

Marsden and Hughes [58, p. 146] describe a rotational motion as the matrix exponential of a skew symmetric matrix. In fact every $R \in SO(n)$, i.e. every

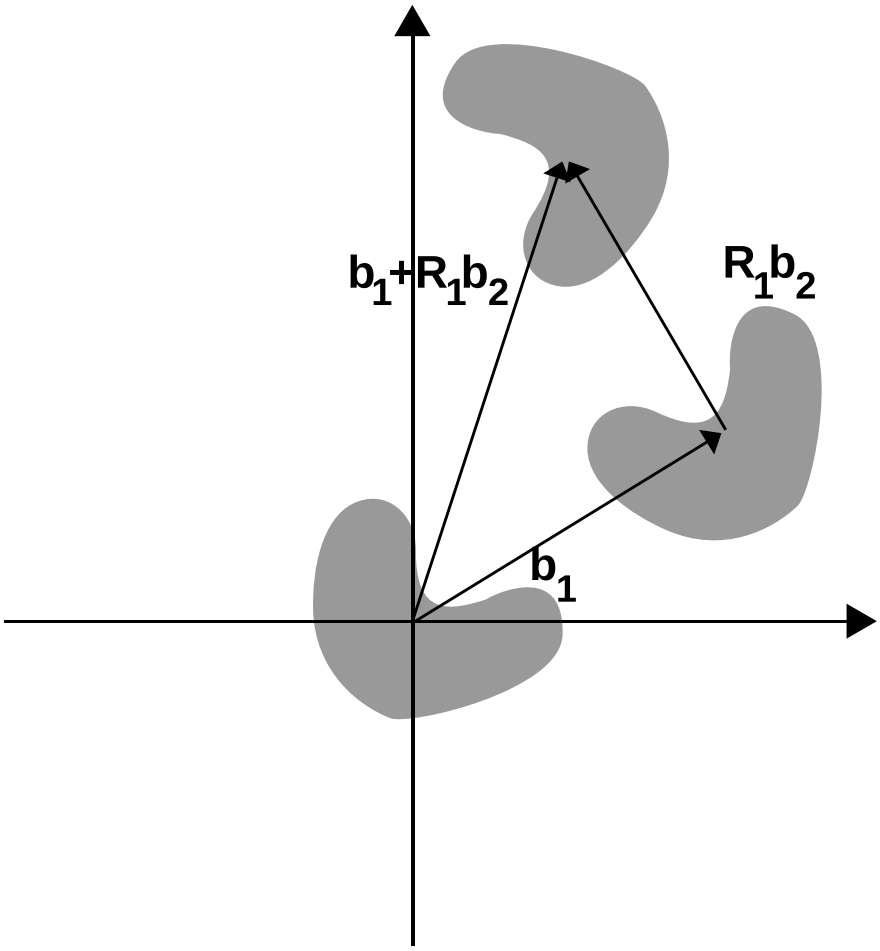

Figure 3.3.1: Two rigid motions added with the translation vectors. The coordinate system is $\mathbb{R}^2_b$.

rotation matrix, can be written as the exponential of a skew symmetric matrix [27, p. 22], and every exponential of a skew symmetric matrix is a member of $SO(n)$ [4, p. 82].

Therefore we can describe every rigid motion as

$$x_p = e^A p + b \tag{3.3.5}$$

where

$$A = \begin{pmatrix} 0 & a_1 & a_2 \\ -a_1 & 0 & a_3 \\ -a_2 & -a_3 & 0 \end{pmatrix} \tag{3.3.6}$$

is a skew symmetric matrix. From this we see that a rigid motion in 3D has 6

degrees of freedom [69, p. 89].

3.3.1 Computing the exponential map

To be able to compute the exponential of a skew symmetric matrix (3.3.6) we first observe that

$$A^2 = \begin{pmatrix} -a_1^2 - a_2^2 & -a_2 a_3 & a_1 a_3 \\ -a_2 a_3 & -a_1^2 - a_3^2 & -a_1 a_2 \\ a_1 a_3 & -a_1 a_2 & -a_2^2 - a_3^2 \end{pmatrix} \tag{3.3.7}$$

and that

$$A^3 = -(a_1^2 + a_2^2 + a_3^2)A = -|x|^2 A \tag{3.3.8}$$

where $x := (a_1, a_2, a_3)^T$.

From (3.3.8) and the exponential formula, we have that

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k = \tag{3.3.9}$$

$$= I + A + \frac{1}{2!}A^2 - \frac{|x|^2}{3!}A - \frac{|x|^2}{4!}A^2 + \frac{|x|^4}{5!}A + \frac{|x|^4}{6!}A^2 - \ldots = \tag{3.3.10}$$

$$= I + A\left(1 - \frac{|x|^2}{3!} + \frac{|x|^4}{5!} - \ldots\right) + A^2\left(\frac{1}{2!} - \frac{|x|^2}{4!} + \frac{|x|^4}{6!} - \ldots\right) = \tag{3.3.11}$$

$$= I + \frac{\sin|x|}{|x|}A + \frac{1 - \cos|x|}{|x|^2}A^2 \tag{3.3.12}$$

Since we know that every matrix can be uniquely decomposed in a symmetric and a skew symmetric part [29, p.32] we see that (3.3.12) is the unique decomposition of a rotation matrix $R \in SO(3)$.

Our accelerometers measure the acceleration 5000 times every second, so the rotation at each time step becomes very small. This leads to small a_1, a_2, a_3, so $\sqrt{(a_1^2 + a_2^2 + a_3^2)} = |x|$ is close to zero and we can approximate the constants in (3.3.12) with their limits as $|x| \to 0$:

$$\frac{\sin|x|}{|x|} \to 1 \tag{3.3.13}$$

$$\frac{1 - \cos|x|}{|x|^2} \to \frac{1}{2}. \tag{3.3.14}$$

We then achieve a much easier expression for our rigid motion

$$x_p = \left(I + A + \frac{1}{2}A^2\right)p + b \tag{3.3.15}$$

where A^2 is the matrix (3.3.7).

Our matrix $I+A$ corresponds to the matrix $I+\Phi$ in (2.6) from Wittenburg [96, p. 22] where Φ is a skew-symmetric matrix with the Bryant angles $(\phi_3, -\phi_2, \phi_1)$ around each coordinate axis as entries. He derives the matrix from the complicated trigonometrically expressed 3D rotation matrix (3.2.3), and he uses only the conditions that $\sin\phi \approx \phi$ and $\cos\phi \approx 1$ for small angles. Actually, he at the same time supposes that the products $\phi_i\phi_j \approx 0$, and this may not be the case at all. The problem is similar to ours, except that our formula with A^2 gives rise to more product terms than in [96, p. 22]. These product terms are hidden in the original trigonometric matrix, and are the difference between $\cos\phi$ and 1, and $\sin\phi$ and ϕ. So our formula expresses the deviation from the skew-symmetric case more visibly than with the formula emerging from the Bryant angles in equation (3.2.3).

3.3.2 Differentiation of the rigid motion

We need an equation at the acceleration level where we can insert our measurements and solve for the six degrees of freedom at each time step. This means that we have to differentiate the combination of (3.3.15) and (3.3.3) twice:

$$a_p(t) = \ddot{R}(t)p + \ddot{d}(t) \Leftrightarrow a_p(t) = \ddot{e^A}(t)p + \ddot{d}(t) \tag{3.3.16}$$

where a_p is the acceleration vector in the body coordinate point p. (We study here the analytical version of the discretized formula (3.3.3) for an easier differentiation process.)

Since $e^A(t)$ is a complicated sum where most terms are products of matrices, we have to make simplifications. It would be meaningless to subdivide the time steps such that $A_iA_j \approx 0$ and write the product $\prod_{k=1}^{i} R_i \approx \sum_{k=1}^{i} A_i$. The differentiation operation where we divide by these new smaller time steps would anyway lead to derivatives of the same size. But we observe that the velocity is only dependent on the motion at the current time step. So we can differentiate the formula $x_p(t) = (I + A + \frac{1}{2}A^2)(t) + b(t)$ while keeping in mind that the discretized version only uses the first time step of this formula at each time step i, and that $A(t)$ and $b(t)$ are different at each time step i.

$$\frac{\partial^2 b}{\partial t^2} = \left(\frac{\partial^2 b_x}{\partial t^2}, \frac{\partial^2 b_y}{\partial t^2}, \frac{\partial^2 b_z}{\partial t^2} \right) \tag{3.3.17}$$

$$\frac{\partial^2 I}{\partial t^2} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.3.18}$$

$$\frac{\partial^2 A}{\partial t^2} = \begin{pmatrix} 0 & \frac{\partial^2 a_1}{\partial t^2} & \frac{\partial^2 a_2}{\partial t^2} \\ -\frac{\partial^2 a_1}{\partial t^2} & 0 & \frac{\partial^2 a_3}{\partial t^2} \\ -\frac{\partial^2 a_2}{\partial t^2} & -\frac{\partial^2 a_3}{\partial t^2} & 0 \end{pmatrix} \tag{3.3.19}$$

where b_x, b_y and b_z are the components of the b vector.

The differentiation of A^2 (3.3.7) is not that beautiful. We have to use the product rule twice:

$$\frac{\partial A_{11}^2}{\partial t} = -2a_1 \frac{\partial a_1}{\partial t} - 2a_2 \frac{\partial a_2}{\partial t} \tag{3.3.20}$$

$$\frac{\partial A_{12}^2}{\partial t} = -a_2 \frac{\partial a_3}{\partial t} - a_3 \frac{\partial a_2}{\partial t} \tag{3.3.21}$$

$$\frac{\partial A_{13}^2}{\partial t} = a_1 \frac{\partial a_3}{\partial t} + a_3 \frac{\partial a_1}{\partial t} \tag{3.3.22}$$

$$\frac{\partial^2 A_{11}^2}{\partial t^2} = -2\frac{\partial a_1}{\partial t}\frac{\partial a_1}{\partial t} - 2\frac{\partial a_2}{\partial t}\frac{\partial a_2}{\partial t} - 2a_1 \frac{\partial^2 a_1}{\partial t^2} - 2a_2 \frac{\partial^2 a_2}{\partial t^2} \tag{3.3.23}$$

$$\frac{\partial^2 A_{12}^2}{\partial t^2} = -2\frac{\partial a_2}{\partial t}\frac{\partial a_3}{\partial t} - a_2 \frac{\partial^2 a_3}{\partial t^2} - a_3 \frac{\partial^2 a_2}{\partial t^2} \tag{3.3.24}$$

and so on.

Since the coefficients a_1, a_2, a_3 is near zero, the contribution from those terms where a_1, a_2, a_3 is a factor will be very small. This means that the terms $\frac{\partial A_{ij}^2}{\partial t}$ is near zero, but from the terms $\frac{\partial^2 A_{ij}^2}{\partial t^2}$ we can get a significant contribution from the terms $\frac{\partial a_i}{\partial t}\frac{\partial a_j}{\partial t}$. These terms measure how fast the rotative part of the motion changes, and may be rather large near the acceleration peaks. Therefore this is a strong indication that we can not use only the contribution from the skew symmetric matrix $\frac{\partial^2 A}{\partial t^2}$ when we compute the peak acceleration at the vertex coordinates. We also have to take into account the first derivative product terms in $\frac{\partial^2 A^2}{\partial t^2}$. Therefore

we approximate the matrix $\frac{\partial^2 A^2}{\partial t^2}$ with

$$\begin{pmatrix} -2\left(\frac{\partial a_1}{\partial t}\frac{\partial a_1}{\partial t}+\frac{\partial a_2}{\partial t}\frac{\partial a_2}{\partial t}\right) & -2\frac{\partial a_2}{\partial t}\frac{\partial a_3}{\partial t} & 2\frac{\partial a_1}{\partial t}\frac{\partial a_3}{\partial t} \\ -2\frac{\partial a_2}{\partial t}\frac{\partial a_3}{\partial t} & -2\left(\frac{\partial a_1}{\partial t}\frac{\partial a_1}{\partial t}+\frac{\partial a_3}{\partial t}\frac{\partial a_3}{\partial t}\right) & -2\frac{\partial a_1}{\partial t}\frac{\partial a_2}{\partial t} \\ 2\frac{\partial a_1}{\partial t}\frac{\partial a_3}{\partial t} & -2\frac{\partial a_1}{\partial t}\frac{\partial a_2}{\partial t} & -2\left(\frac{\partial a_2}{\partial t}\frac{\partial a_2}{\partial t}+\frac{\partial a_3}{\partial t}\frac{\partial a_3}{\partial t}\right) \end{pmatrix}. \tag{3.3.25}$$

We have now achieved the formula

$$\begin{pmatrix} a_{p_x} \\ a_{p_y} \\ a_{p_z} \end{pmatrix} = \begin{pmatrix} 0 & \frac{\partial^2 a_1}{\partial t^2} & \frac{\partial^2 a_2}{\partial t^2} \\ -\frac{\partial^2 a_1}{\partial t^2} & 0 & \frac{\partial^2 a_3}{\partial t^2} \\ -\frac{\partial^2 a_2}{\partial t^2} & -\frac{\partial^2 a_3}{\partial t^2} & 0 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} -\left(\frac{\partial a_1}{\partial t}\frac{\partial a_1}{\partial t}+\frac{\partial a_2}{\partial t}\frac{\partial a_2}{\partial t}\right) & -\frac{\partial a_2}{\partial t}\frac{\partial a_3}{\partial t} & \frac{\partial a_1}{\partial t}\frac{\partial a_3}{\partial t} \\ -\frac{\partial a_2}{\partial t}\frac{\partial a_3}{\partial t} & -\left(\frac{\partial a_1}{\partial t}\frac{\partial a_1}{\partial t}+\frac{\partial a_3}{\partial t}\frac{\partial a_3}{\partial t}\right) & -\frac{\partial a_1}{\partial t}\frac{\partial a_2}{\partial t} \\ \frac{\partial a_1}{\partial t}\frac{\partial a_3}{\partial t} & -\frac{\partial a_1}{\partial t}\frac{\partial a_2}{\partial t} & -\left(\frac{\partial a_2}{\partial t}\frac{\partial a_2}{\partial t}+\frac{\partial a_3}{\partial t}\frac{\partial a_3}{\partial t}\right) \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} \frac{\partial^2 b_x}{\partial t^2} \\ \frac{\partial^2 b_y}{\partial t^2} \\ \frac{\partial^2 b_z}{\partial t^2} \end{pmatrix} \tag{3.3.26}$$

where a_{p_x}, a_{p_y} and a_{p_z} are the linear accelerations in directions x, y and z respectively in point p.

We restate equation (2.4.1) in component form here, and remember that earlier accelerometer configurations are based on this non-linear ODE.

$$a_{p_x} = -p_x(\omega_y^2+\omega_z^2)+p_y(\omega_x\omega_y-\dot{\omega}_z)+p_z(\omega_x\omega_z+\dot{\omega}_y)+a_{0_x} \tag{3.3.27}$$

$$a_{p_y} = p_x(\omega_x\omega_y+\dot{\omega}_z)-p_y(\omega_x^2+\omega_z^2)+p_z(\omega_y\omega_z-\dot{\omega}_x)+a_{0_y} \tag{3.3.28}$$

$$a_{p_z} = p_x(\omega_x\omega_z-\dot{\omega}_y)+p_y(\omega_y\omega_z+\dot{\omega}_x)-p_z(\omega_x^2+\omega_y^2)+a_{0_z} \tag{3.3.29}$$

where a_{0_k} is the linear acceleration in the body origo along the k-axis, and ω_k is the angular velocity around the body coordinates axis k.

Since our b_i-vectors are always the translation vectors of origo, the differentiated b_k components in equation (3.3.26) are the acceleration components of the body origo, i.e. the same as a_{0_k} in (3.3.27) to (3.3.29). If we compare equation (3.3.26) with (3.3.27) to (3.3.29), we see that they are actually the same equations. The definitions

$$\omega_x := -\frac{\partial a_3}{\partial t}, \quad \omega_y := \frac{\partial a_2}{\partial t}, \quad \omega_z := -\frac{\partial a_1}{\partial t} \tag{3.3.30}$$

make the equations coincide. This means that our degrees of freedom a_1, a_2 and a_3 actually are the Bryant angles [96, p. 21] $-\phi_3$, ϕ_2 and $-\phi_1$ respectively.

Our way of writing the rotation matrices R_i as the matrix exponential of a skew symmetric matrix lead to an easy expression which avoid the complicated trigonometric equations in Wittenburg [96], but we can not escape from the physically derived formula (2.4.1) that is the origin of the NAP and the other 3D acceleration methods. Thus we will also get a problem with the non-linear ODE in (3.3.27) to (3.3.29) if we try to solve for 3D acceleration and angular velocities.

3.4 New accelerometer configuration

We saw in Section 3.3.2 that the matrix in the acceleration formula (3.3.26) was not only a skew symmetric matrix but also had some terms consisting of products of angular velocities. These terms turn the equation into a non-linear ODE and complicate the solution for the degrees of freedom that we need for 3D acceleration and the description of the motion. In this section we investigate the possibility of forcing the matrix to be skew symmetric anyway. Then we will not be able to compute the motion afterwards since the computed skew-symmetric matrices will not only consist of the angular acceleration $\dot{\omega}$ (often called α), and the vector $\frac{\partial^2 b_i}{\partial t^2}$ will no longer consist of the linear acceleration in origo, so all values will be non-physical. If we could find an accelerometer configuration that gives a small error while using the skew-symmetric matrix only, the benefits would be the possibility for computing 3D acceleration in small spaces, and a smaller error than the NAP gives when angular velocities are changing directions. If we could achieve 3D acceleration computations from this new configuration, we could later on simulate an artificial NAP, and solve for the motion degrees of freedom afterwards.

Our current accelerometer configuration makes it impossible to solve for 3D acceleration. There are two reasons for this:

1. The sensing axes directions do not discover the tangential acceleration field if we have a rotation around origo (see Section 2.7).

2. It is impossible to choose a coordinate system for the configuration without exterminating some degrees of freedom in equation (3.3.26).

Therefore we need new accelerometer positions where the sensing axes in the best possible way measure both the tangential- and centripetal acceleration, and good coordinates for the sensors that minimize the error in solving for the degrees of freedom. The coordinate system problem will be explained below.

When we force the matrix in (3.3.26) to be skew, we investigate the following transformation

$$a = \begin{pmatrix} 0 & c_1 & c_2 \\ -c_1 & 0 & c_3 \\ -c_2 & -c_3 & 0 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \Leftrightarrow \tag{3.4.1}$$

$$\begin{cases} a_x = +c_1 p_y + c_2 p_z + b_1 \\ a_y = -c_1 p_x + c_3 p_z + b_2 \\ a_z = -c_2 p_x - c_3 p_y + b_3 \end{cases} \tag{3.4.2}$$

The left hand side a is an acceleration vector where we insert our acceleration measurements, and the c_i and b_j are the six degrees of freedom that can be computed if we have at least six equations (i.e. six linear acceleration measurements at good coordinates). These degrees of freedom can afterwards be used for computing acceleration at other points than the measuring ones, and we can thus achieve the 3D acceleration field.

The solvability of our new method is dependent on the coordinates of the accelerometers. It is important that the accelerometer position coordinates for the other axes than the sensing axes are different from zero, since they otherwise will exterminate one of the c_i. It is also important for the accelerometers to measure both parts of the tangential and angular accelerations. Since we have six degrees of freedom we need either six equations from uniaxial measurements, or two 3D equations. Some possibilities could be to use two triaxial accelerometers with as large spacing between them as possible; to use six uniaxial accelerometers at smart positions; or to use a larger amount of sensors combined with an optimization process.

To find a good position for the accelerometers we first change the x- and y-sensors, such that we annihilate fewer degrees of freedom. We also need to have more position coordinates different from zero, in addition to the possibilities of measuring both centripetal and tangential acceleration. Therefore we need to rotate the coordinate system. We suppose that the error is smaller if we choose coordinates as far away as possible from the singular original zero coordinates position. Then the condition number would be rather low [34]. This means that we rotate the coordinates 45 degrees around the y-axis. Fortunately, this also implies that the sensors measure large parts of both the tangential and centripetal acceleration since the sensing axes must follow the coordinate axes in equation (3.4.2) (see Section 2.7). This rotation means that the sensors need to be placed in a 45 degrees angle towards the surface of the accelerometer bracket. Since it is easier to glue the sensor onto the bracket side, we prefer to take the inverse of the rotation afterwards, to physically moving the accelerometers to new positions, but keep the coordinate axes parallel to the surface of the bracket.

The coordinate rotation can be expressed in the following way. Let $\mathcal{E}$ be the standard basis in $\mathbb{R}^3$ and $\mathcal{B}$ be our new basis and let $\{b_1, b_2, b_3\}$ be the coordinates for our basis $\mathcal{B}$ in the standard basis. Then there exists a change-of-coordinates matrix $\underset{\mathcal{B}\leftarrow\mathcal{E}}{P}$ that transforms the coordinates for the standard basis $\mathcal{E}$ into the new coordinates $\mathcal{B}$. This matrix is $\left(\underset{\mathcal{E}\leftarrow\mathcal{B}}{P}\right)^{-1} = (b_1, b_2, b_3)^{-1}$, see [53][p. 269], so it is enough to find coordinates for our new basis $\mathcal{B}$ in the standard basis $\mathcal{E}$. The matrix for our 45 degrees change-of-coordinates is

$$\left(\underset{\mathcal{E}\leftarrow\mathcal{B}}{P}\right)^{-1} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix}^{-1}. \tag{3.4.3}$$

For testing our new configuration we use the measured values from our old configuration. These incorporate noise, and vary between low and high values, and small and large differences between the sensors. The noise should be a good indication of the condition number (and thus the size of the error) of the transformation (3.4.2).

We compute the condition number of the matrix D that solves for $c_1 \dots b_3$ in (3.4.2). If we define $p_{y_{x_1}}$ to be the y-coordinate of the x_1-sensor and so on, we have the following description of D [34]:

$$D = \begin{pmatrix} p_{y_{x_1}} & p_{z_{x_1}} & 0 & 1 & 0 & 0 \\ p_{y_{x_2}} & p_{z_{x_2}} & 0 & 1 & 0 & 0 \\ -p_{x_{y_1}} & 0 & p_{z_{y_1}} & 0 & 1 & 0 \\ -p_{x_{y_2}} & 0 & p_{z_{y_2}} & 0 & 1 & 0 \\ 0 & -p_{x_{z_1}} & -p_{y_{z_1}} & 0 & 0 & 1 \\ 0 & -p_{x_{z_2}} & -p_{y_{z_2}} & 0 & 0 & 1 \end{pmatrix}. \tag{3.4.4}$$

The condition number is an indication of how near the system is to a singular one. See Section 5.7 for the numerical tests of our new accelerometer configuration.

It is also important to validate our new 3D accelerometer configuration and equation (3.4.2) with analytically computed acceleration data before building it.

Another possibility of a new accelerometer configuration is to combine angular velocity measurements with acceleration measurements, to use the angular velocity data in formulas (3.3.27) to (3.3.29) and solve for the other degrees of freedom (i.e. the angular acceleration and the linear acceleration in origo). This would be a further development of Martin et al.'s [59] configurations, where we avoid the blowing-up of the error in the differentiation process of ω. One could

further place a triaxial accelerometer in origo, and compare those measurements with the computed $\frac{\partial^2 b_i}{\partial t^2}$ vector. The values should coincide. The disadvantage could be space problems since gyros that measure the angular velocity are larger than accelerometers.

3.5 Geometrical method for solving for the rotation axis position

In this section we will show that for some important special cases it is possible to solve for the position of the rotation axis. This method opens an alternative way of validating the vertex acceleration measurements. The method is only valid for our current accelerometer configuration (see Figure 2.7.2) where the x_i-sensors' positions are at the prolonged line from the sensing axes. The sensing axes is being changed if we use a new coordinate system (see Section 2.7).

In the following deduction we use the fact that the acceleration measurements at the y-sensors are ≈ 0 in our peak acceleration shaking test, as we see in Figure 3.5.2. It follows that the rotation is two-dimensional and only in the x-z-plane.

If the rotation axis is placed on the z-axis, by symmetry we do not get any difference of the measurements at the x_1- and x_2-sensors (see Figure 3.5.1). Also, if the rotation axis is situated at a relatively long distance from the sensors, we get almost no difference in the measurements since the acceleration is proportional to the distance to the rotation axis, and the angular acceleration is the same at all points of the rigid dummy head:

$$a_{tan} = \alpha r \approx \alpha(r+\varepsilon) \quad \text{and} \quad a_{rad} = \omega^2 r \approx \omega^2(r+\varepsilon) \tag{3.5.1}$$

where α is the angular acceleration, ω is the angular velocity and $r \gg \varepsilon$.

So if we study the acceleration diagrams we can analytically solve for the position of the rotation axis at the time steps where $x_1 \approx x_2$. Since the vertex is situated 57 mm above the x_i-sensors, we get

$$a_{tan} = r\alpha \Rightarrow \tag{3.5.2}$$

$$\begin{cases} a_{x_{vertex}} = (r+0.057)\alpha \\ a_{x_1} = r\alpha \end{cases} \tag{3.5.3}$$

Fortunately we have coinciding x_i measurements at the acceleration peaks of our peak acceleration shaking test (see Figure 3.5.2), so we can use this method for finding the position of the rotation axis at the peaks (see Section 5.5).

It is also possible to construct an algorithm for a numerical solver for the position of the rotation axis. We will not describe the method in detail here since

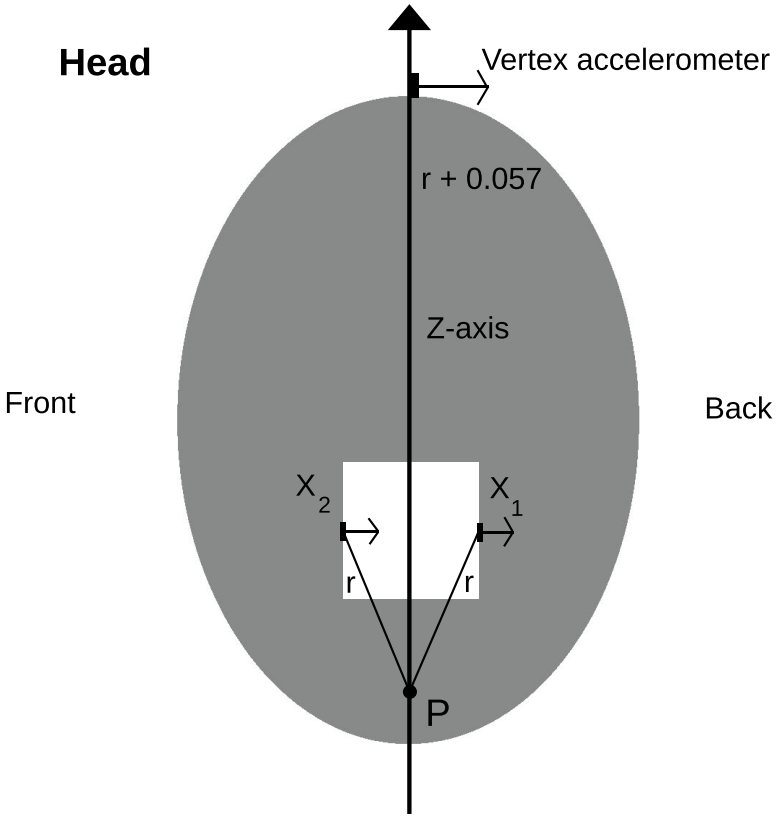

Figure 3.5.1: Symmetry of acceleration measurements when the rotation axis goes through the z-axis.

we did not have time for implementing the method in this thesis. However, it is based on a geometrical argument where we decompose the acceleration at the sensor positions into centripetal and tangential acceleration. The unknowns thus becomes the x- and z-coordinates for the rotation axis position, α and ω^2, and the rotation direction. The latter can be determined mostly from the video, so only four degrees of freedom remain. Since we have five measurements in the x-z-plane, we need only to use the four measurements from the acceleration bracket for the numerical solver. Therefore, we can compare the rotation axis position achieved numerically from the four lower sensors, with the analytically computed rotation axis position where we use the x-sensors and the vertex measurements. If they coincide at the acceleration peak, we have validated our measurements.

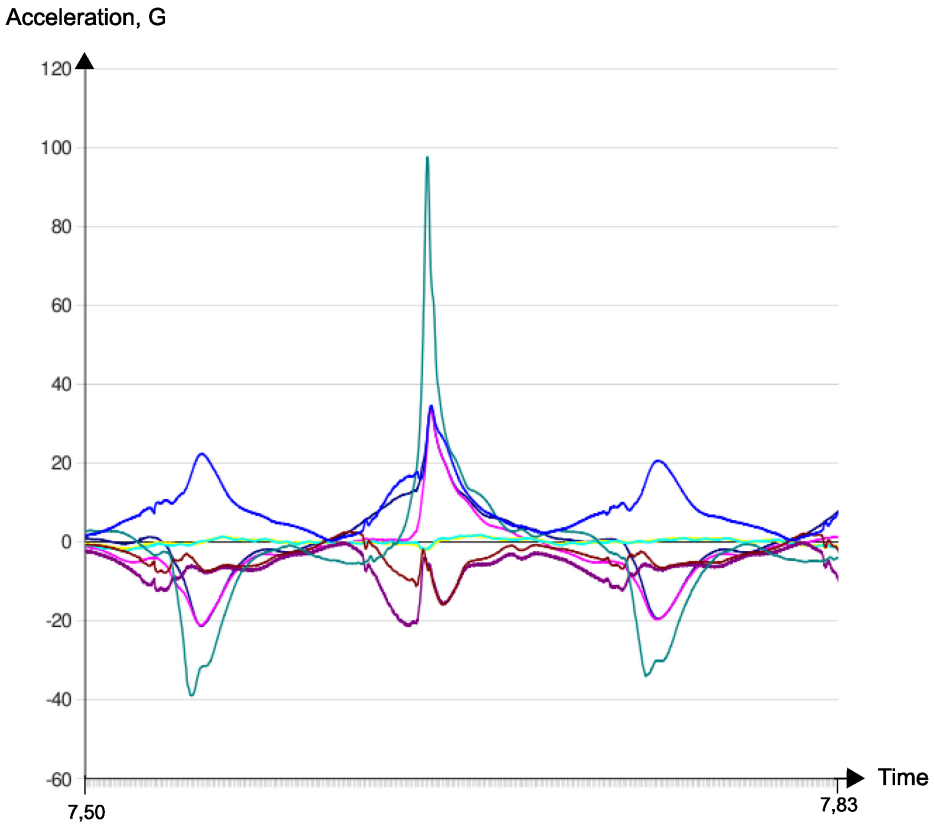

Figure 3.5.2: A typical cut from our peak shaking session. The dark green curve is the vertex measurements (x-direction); the pink and dark blue is the x_1 and x_2 sensors; the cerise and brown are z_1 and z_2; the yellow and light green are y_1 and y_2; and the always positive blue is the absolute acceleration. See Section 2.7 for the accelerometer positions.

In the case of pure translational acceleration the method is not solvable. We need a rather large difference between the sensors to be able to solve it accurately. This difference is present in the vicinity of the acceleration peaks, so a combination of the analytical method described above, and the numerical one here, would make it possible to validate the vertex measurements around the peaks.

3.6 Finding motion and velocity from acceleration values

We are interested in the max velocity of the vertex to compare this with articles by other authors [26, 73, 19]. We also want to compute the length of a shaking cycle. We therefore integrate the measured vertex values twice. We suppose that the velocity is zero at the maximum peak acceleration, and that the max velocity occurs where the acceleration changes sign (see Section 3.2).

We choose to integrate with the trapezoidal rule, since the method smooths the integrated curves, and the time steps are very small. It approximates the curves linearly between the measuring points. It is not necessary to smooth the acceleration values first, since the integration procedure automatically uses the arithmetic middle value between two adjacent points.

The analytic formula for the double integrals from a to b of the vertex acceleration values is

$$\int_a^b \int_a^t f(\tau) d\tau dt \tag{3.6.1}$$

From this equation we see that we need to know two initial values, the velocity and the position. We tested the integration routine with several polynomials and trigonometric functions. See the program *IntegrateTranslationalShaking.py* in the Appendix A.2 for the implementation.

We also wanted to check the computations of the maximum velocity from Duhaime et al. [26]. We simulated some shaking cycles with a cosine function. Our function uses the mean peak tangential acceleration as amplitude, and a time period 0.3 s which we have taken from fig 1 in [26, p. 413]. This time period is longer than 2 times the "time interval" of 106.6 ms from page 413, and should give a larger max velocity than the one in [26, table 6, p. 412]. Our cosine function has less sharp peaks than the curve in fig 1 at page 413, and should thus give a larger contribution to the integrated velocity. The program *DuhaimeVelocitySimulated.py* can be found in the Appendix A.2, and the results from the integrations are reported in Section 5.3.

Chapter 4

Elasticity modeling

In our work we use the linear elasticity equation to model the displacement and shear strain of the brain undergoing different accelerations (see Sections 4.2 and Chapter 6). In this chapter we briefly introduce the linear elasticity theory, and our linear elastic head 2D model. Thereafter we describe some elasticity properties of brain tissue, and describe how we choose the "right" values for our computations among the wide range of existing experimental values.

The linear elasticity equation is only suitable for very small (infinitesimal) deformations. Several recent investigations of brain tissue properties [89] have found that the constitutive laws of brain tissue follow an Ogden- or Mooney-Rivlin nonlinear model. van Dommelen et al. [89, p. 265] propose that brain tissue behaves nearly linear elastic up to a shear strain of order 10^{-2}, and that the nonlinearity appears only at strains larger than this value. We calibrate our elasticity computations against strain values from Zou [101, p.123] which are between 0.07 and 0.1, so a linear elastic model is insufficient for realistic simulations in our case.

We use a circular 2D mesh with a circular brain inside for our elasticity computations. We model the brain tissue as homogeneous and isotropic, and we do not differentiate between white and grey matter, although brain tissue is actually non-homogeneous and anisotropic, and the white and grey matter have different elastic properties. We use only two different materials (subdomains) in our mesh – skull bone and brain tissue. The subdomains can be seen in Figure 4.0.1.

In our FE-mesh we attach the brain to the skull to simplify the computations. Anatomically, there is a layer of water like fluid (CSF) and three membranes between the brain and the skull (see Section 2.1). So a more accurate model would be to simulate the CSF layer as a fluid, or use FSI (Fluid-Solid Interaction) computations. This would give a more correct model for computing the brain/skull displacement important for the Shaken Baby Syndrome.

There are several other drawbacks with the circular two-material mesh. Of course the head and brain are not circular. Inside the brain there are also some

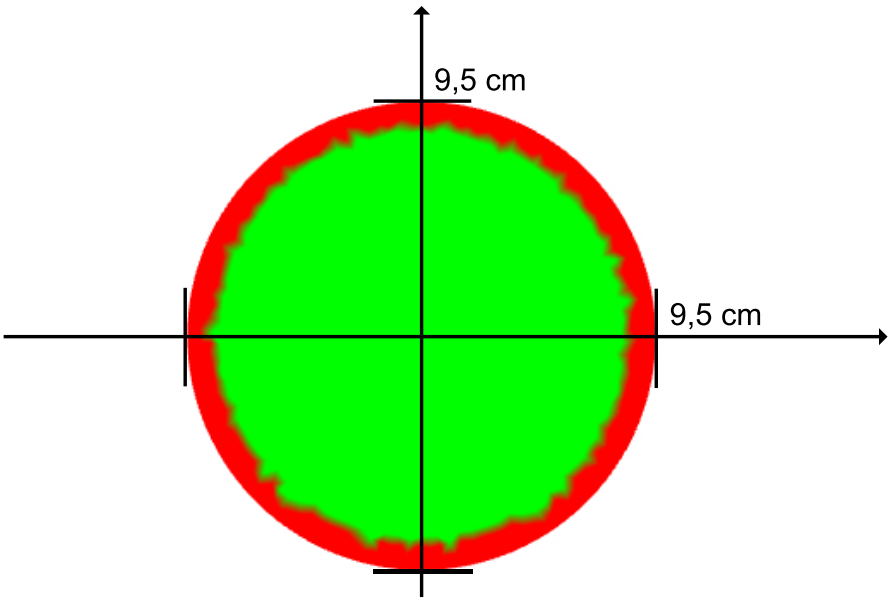

Figure 4.0.1: Subdomains for the skull and brain in our mesh. Green is modeled as brain tissue, and red as skull. Here, the mesh has adult head size. The subdomains are scaled according to the head diameter chosen.

CSF-filled cavities (ventricles). These should be modeled like fluid filled cavities due to the brain's ability to compress the ventricles during accelerations of a certain duration (see Section 4.4.2). The inside of the head also consists of harder structures as the membrane between the brain hemispheres – the *falx cerebri*. In fact the brain is also attached to the skull base due to its connection with the spinal cord, and the peaks of bone at the skull base. It may also be relevant for SBS simulations to model the surface of the brain with gyri and sulci. The sulci protects the brain against high levels of shear strain and strain rate [40] at impacts, but it does not alter the displacement magnitude or the relative brain/skull motion.

Brain tissue is actually viscoelastic [89]. This means roughly that brain tissue can not be regarded as only a solid, but that it also behaves a little bit as a fluid. In our Finite Element (FE) model in Section 4.7.1 and Chapter 6, we do not consider viscoelasticity, but we will further develop our simulations to take into account viscoelasticity after the work with this thesis.

The acceleration values we choose for our elasticity simulations come from our peak acceleration data set. This specific data set shows almost no acceleration in the y-direction, so the forces are approximately uniformly distributed along the

y-axis. In this investigation we therefore only consider a 2D head slice at the x-z-plane [51, p. 496].

4.1 Stress and strain

Elasticity theory deal with how a material deforms when exposed to a force. Two concepts that are important for elasticity are stress, σ, and strain, ε. These can be expressed as 2x2 matrices in two dimensions, and 3x3 matrices in three dimensions, to keep the information in all dimensions. These matrices display special matrix properties (properties of tensors [42]), and are thus named stress tensor and strain tensor respectively. The stress tensor, σ, is a measure of force per unit area and its SI unit is Pascal (Pa). The strain tensor describes the fraction of the length difference after deformation, and the original length, $\Delta L/L_0$, and is thus dimensionless.

The stress tensor entries describe forces acting on arbitrarily oriented surfaces inside the elastic material. The diagonal components of the tensor are called normal stresses, and describe forces perpendicular to a given surface, while the off-diagonal entries are so-called shear stresses that describe forces parallel to the surface. A bit simplified, the normal stresses tend to stretch or compress the material in a given direction, while the shear stresses tend to change the angles between line segments (i.e. twist the material). Similarly, the diagonal components of the strain tensors are the normal strains (stretch and compression), and the off-diagonal entries are the shear strains. A good introduction to linear elasticity and the stress and strain tensor entries, may be found in [51, Chapter 5] (also numerical solutions), [50] and [88][Chapter 1].

As we will see, the brain is more sensitive to shear stress than to normal stress. Therefore angular acceleration forces are more dangerous to the brain than translational forces [41]. (Other stress/strain tensors than those used in linear elasticity also exist [42], which are mostly used in non-linear elasticity computations.)

The equation for the strain tensor in index notation is

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{4.1.1}$$

where u is the displacement field of how much each point is moved after deformation.

A constitutive law describes the relation between different physical quantities. In our case we need a relation between σ and ε. Hooke's law states that the force acting on a material is linearly proportional to the displacement of the material. This means that the stress is proportional to the strain. For an isotropic linear

elastic media, the generalized Hooke's law becomes

$$\sigma = \lambda \mathrm{tr}(\varepsilon(u))I + 2\mu\varepsilon(u) \tag{4.1.2}$$

where λ is a material constant called Lamé's first parameter, and μ is the shear modulus called Lamé's second parameter (see Sections 4.3 and 4.4.1).

4.2 Linear elasticity equations

Newton's second law states that $ma = F$, where F is force, m is mass, and a is the acceleration. Since stress is force divided by area we get (index notation)

$$\rho u_{i,tt} = \sigma_{ij,j} + \rho b_i, \tag{4.2.1}$$

where ρ is density and b_i are the body accelerations. Inserting equation (4.1.2) for σ leads to the linear elasticity *wave* equation (vector notation):

$$\rho \mathbf{u}_{tt} = (\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) + \mu\nabla^2\mathbf{u} + \rho\mathbf{b}. \tag{4.2.2}$$

It means that introduced body loadings distribute as elastic waves into the body. If the displacement acceleration term is negligible, we arrive at the *static* linear elasticity equation [51]:

$$(\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) + \mu\nabla^2\mathbf{u} = -\rho\mathbf{b}. \tag{4.2.3}$$

In our elasticity computations we use the static linear elasticity equation (4.2.3) (see Chapter 6). We want to see if the static linear equation can compute strains of the same magnitude as the computations [101, 38], where they used experimental acceleration values from Hardy et al. [37, 38]. Analytical solutions of the pure rotational elasticity problems [11, 31, 30] suppose that it is the induced shear waves that cause the brain injury, which indicates that a more appropriate model may be (4.2.2) in combination with viscoelastic Lamé parameters.

4.3 The elastic moduli

The constants μ and λ are material parameters that describe the material properties, and therefore differ between different materials. The shear modulus μ, Lamé's second parameter, is a measure of how likely it is for the material to deform if exposed to forces that change the angle between to originally orthogonal lines – shear forces. Lamé's first parameter, λ, is a material constant without physical meaning, but which incorporates important physical quantities like compressibility.

There are several other choices of material parameters for linear elasticity (called elastic moduli) [88, Chpt. 1]. Three of them are the bulk modulus K, Young's modulus E, and Poisson's ratio ν. K describes volumetric elasticity, i.e. how much the material changes volume undergoing uniform compression forces in all directions. E describes tensile elasticity, and ν how much the material changes volume in directions orthogonal to a linear force acting on the material.

The elastic moduli are related to each other in the way that you only need to know two of them, to be able to compute all the other elastic moduli with use of conversion formulas. We discovered, though, that some of the formulas show an unphysical behaviour when the material is nearly incompressible. So one has to be very careful when using the conversion formulas (see Section 4.5).

One material parameter of great relevance for us is the Poisson's ratio ν. It is a measure of compressibility, and never exceeds 0.5, which means complete incompressibility. As we see in Section 4.4, the compressibility of brain tissue differs during different time scales, in different directions of the acceleration force, and with different sizes of the sample. While making mathematical models of the body, it is important to choose the elastic moduli values that are appropriate to the nature of the problem.

The brain is anisotropic, but we assume an isotropic brain in our simulations for simplicity. Therefore, we can use the same elasticity properties in all directions.

4.4 Experimental values of the elastic moduli

The elastic moduli of an adult and an infant *skull* are different, since the baby skull bones have not grown completely together yet [60, 61, 52, 48]. The elastic moduli also vary a lot. Therefore the infant skull acts less stiff than the adult skull as a whole. However, in our problem we shall only test the effect of acceleration inside the head, and do not care about real world impacts on the skull bone. We therefore choose to use adult skull values for the infant simulations as well. The elastic moduli we choose for the skull bone are: Young's modulus (E) = 15 GPa [39] and Poisson's ratio $\nu = 0.22$ [39, 79].

The experimental values of the *brain* tissue elastic moduli vary a lot between different experiments. The main reason for this is different test conditions [89, 16, 43]. Sarron et al. [83, p. 691] and Zou [101, p. 118] have each an interesting table summarizing several experimental values of the elastic moduli, where for example the Young's modulus varies from 10^4 to 10^6 and the Poisson ratio varies from 0.48 to 0.4999. Unfortunately these table values differ while citing the same articles, so one would have to check carefully the original sources. We do not prioritize using time for checking all experimental values, since these vary a lot anyway. We use

instead several different experimental values of the elastic moduli from [101] for our simulations. In the end we choose some appropriate Lamé parameter values that can reproduce the strain values in Hardy et al.'s acceleration experiments with human heads [37, 101].

4.4.1 The shear modulus

The name of the shear modulus can be either μ or G. Both terms are often used in literature. It is most common to talk about the viscoelastic shear modulus as G(t).

The early analytical solutions of brain injury problems used a linear elastic model, and thus they use a non-viscoelastic shear modulus [11, 31]. Zou [101] uses a non-viscoelastic linear elastic model for his first models. He thus needs the shear modulus to be time-independent. He gives some references to the time-independent shear modulus, and claims that Firoozbakhsh and DeSilvas [31] use 13.8 kPa as Bycroft [11], but Bycroft does not present a single number of the shear modulus. Instead Firoozbakhsh and DeSilvas use $32.5 \frac{\text{lb}}{\text{in}^2}$ which is the same as 224 kPa. Since the ratio of the bulk modulus ($\sim 2GPa$) and the shear modulus should be of order 10^4 to 10^5 [87, 31], the adult shear modulus should be somewhere between 200 kPa and 20 kPa. This is rather low compared to other materials. Some examples for the shear modulo are: aluminium 25.5 GPa, polyethylene 0.117 GPa, and rubber 0.0006 GPa.

4.4.2 Poisson's ratio

In crash test research and numerical simulations of impact head injuries, it is standard to model the brain as a nearly incompressible material [39]. The experimental values of Poisson's ratio $0.48 < \nu < 0.4999$ cited in [101, p. 118] [83, p. 691] model the brain tissue nearly incompressible. The time duration of the shaking of a baby may be at least one second, and a usual time duration of an impact is about 15 ms. One may question if the brain is actually compressible during longer acceleration durations than in impact.

An article by Guillaume et al. on Poisson's ratio values [36] is very interesting for our simulations. It is the only experimental investigation of Poisson's ratio with *a whole brain* as a sample, and their conclusion is that the brain is actually *compressible* with a Poisson's ratio of 0.35 during upwards z-accelerations of several seconds and low-magnitude (9-15 g) accelerations. All earlier experiments have only used parts of the brain as a sample.

The explanation for the compressibility results is that the brain is full of blood vessels, and cavities (ventricles) filled with CSF (Cerebrospinal fluid) (see Section 2.1). During an acceleration in the upward direction, the CSF flows out from the ventricles down into and surrounding the spinal cord, or out into the subarachnoid

space between the brain and the skull. The blood is also pressured downwards into the rest of the body. This is the reason why moderate level g-accelerations in this direction make pilots unconscious rather quickly [83].

In their experiments Guillaume et al. [36] remove the outer membranes of the brain before the acceleration tests, but keep the innermost membrane (the pia mater) that is in direct contact with the CSF and protects the brain against unnatural leakages. The ventricles and the space surrounding the brain are filled with a CSF-like fluid, and the fluid is pumped into the brain with a force simulating heart beats.

They use young bovine brains and expose them to centrifugal acceleration in the range of 0.1 to 15 g. This acceleration is in the upwards direction. They accelerate the brains up to a peak value of either 9 or 15 g, and keep this peak acceleration in about 30 seconds before deceleration.

Forensic scientists at the University of Oslo (UiO) have also experienced that a thick bleeding may compress the brain without the brain expanding other places. This is another indication of the brain compressibility. After death, however, the brain acts more compressible due to lower blood pressure, so the real values of Poisson's ratio would be higher than Guillaume et al.'s 0.35.

The outflow of blood and CSF that pilots experience will perhaps not have time to affect the brains of car crash or impact victims, since an impact duration is maximum 0.020 s and the acceleration in airplanes lasts longer. A shaking cycle (up-and-down) lasts about 0.250 s, and a shaking period (several shaking cycles) may last several seconds. This is similar to the pilot's situation. Therefore, it is natural to model the brain as nearly incompressible at shorter time periods, but more compressible at larger acceleration periods like during shaking.

An important difference between the acceleration used in the experimental tests of Guillaume et al. [36] and our acceleration measurements, is that our measurements are from the rigid dummy head, and do not describe the real brain acceleration. Our measurements describe well the skull acceleration, but the brain will experience an acceleration caused by the acceleration of the skull onto the brain-CSF system. This interior acceleration field may differ from our rigid head interior measurements.

The accelerations used in the Guillaume et al. experiments [36] are of the same magnitude, but are in the opposite direction to our z-accelerations (see Section 5.1 and Figure 5.0.1). This will of course result in a different problem, but there are similarities. Since moderate upward g-accelerations can compress the brain, we suppose that our high-level x-accelerations may also compress the brain, and squeeze the CSF and blood out from the ventricles and blood vessels. The downwards z-acceleration (centripetal acceleration) measured in our shaking experiments will accelerate the skull downward towards the brain. The brain will move up towards the skull due to its larger density compared to the CSF. The cen-

tripetal force will force the blood in the skull veins (the extension of the bridging veins) back to the bridging veins at the top of the head. If the acceleration field at the same time has compressed the brain and squeezed out some amounts of blood, then there would be collected a higher amount of blood in the bridging veins than normally. This could cause a larger pressure than normal from the vessel interior onto the vessel wall, with unknown consequences. But it is interesting to notice that the subdural bleedings that occur in SBS are placed exactly where the blood would accumulate due to the centripetal force – at the top of the head (the vertex area).

Perhaps Poisson's ratio and bulk modulus also would be affected by age-dependency (see Section 4.4.3). The channels from the ventricles to the surrounding CSF may differ from the adult, and cause a faster compression of the ventricles due to moderate g-forces.

Further experiments on finding the right Poisson's ratio should focus on finding ν as a function of the time duration of the applied force, and also as a function of the direction of the force and the size of the sample. Further research is also needed for finding out if the bridging veins more easily get ruptured when exposed to a higher interior pressure than under normal conditions. It would also be interesting to expose a whole head to centrifugal forces, and study the blood streams and the blood pressure at the bridging veins.

4.4.3 Age-dependent shear modulus

The elastic moduli of brain tissue are also age-dependent.

The isolation of the nerve axons in the white matter occurs largely the first months after birth – a process called myelination. This means that a layer of fat grows around the axons. Myelination affects the elastic properties. Only three different studies of experimental pediatric brain tissue values exist [87, 75, 33]. These measure the time dependent viscoelastic shear modulus G(t).

The first investigation by Thibault and Margulies [87, p.1125] showed that the pediatric shear modulus was *smaller* than the adult value. The limitations with the experiment were that it was not statistically significant and measured small strain only. The next two investigation by Prange and Margulies [75] and Gefen et al. [33] showed that the pediatric shear modulus was in fact *larger* than the adult values.

In our FE computations (see Section 4.7.1 and Chapter 6) we did not consider viscoelasticity. Since all experimental infant brain tissue values are a measure of the viscoelastic shear modulus, we have scaled the adult linear elastic shear moduli according to the time zero shear modulus quotient of infant/adult experimental results, and thus achieved three different scalings: 0.667 from [87], 2.434 from [75], and 1.9 from [33].

The interface between unmyelinated and myelinated white matter may also undergo larger strain due to the differences in stiffness. We see this behaviour in our FE computations at the boundary of the attached brain where the mesh changes stiffness (see Chapter 6). Therefore, one has to be careful when drawing conclusions that the stiffer infant brain can resist strain better than the adult brain.

Roth et al. [79] make numerical simulations of baby shaking, and use a bulk modulus (K) of only 2.11 MPa. They refer to a bulk modulus value from Thibault et Margulies [87], but there they have not investigated the infant bulk modulus at all. The adult value of K lies in the range of 2.1 GPa to 2.5 GPa [89], and an often used value is 2.3 GPa from [8]. It may seem like Roth et al. have found the infant brain to be more compressible, but we actually think it is a typo. In another article on child head injuries by the same authors [82] they use K = 2.11 GPa as in the adult case. The bulk modulus should be of size 10^5 bigger than the shear modulus [87, 31], and this is also an indication of 2.11 MPa being 1000 times to small.

4.5 Computations with the conversion formulas

As mentioned above, one only needs any pair of elastic moduli to compute the others. The formulas for doing this are called conversion formulas, and good references are [88, 93]. In this section we reveal and solve an important unphysical behaviour of some of the conversion formulas, when ν is close to 0.5.

In our elasticity simulations we need values of μ and λ (Equation (4.2.3)). First we compute μ from experimental values of E and ν [101, p. 118] (see Table 4.5.1 and the program *LinearElasticitySimulator.py* in the Appendix A.1). Below we will explain an optimal order for computing the elastic moduli needed for our elasticity simulations.

Young's modulus E (kPa)	Poisson's ratio ν
10	0.4999
66.7	0.499
66.7	0.48
80	0.49
250	0.49
675	0.48
1000	0.48

Table 4.5.1: Experimental values of the elastic moduli used in the elasticity computations in Section 6.2.

In the experimental values in Table 4.5.1, the brain is found to be almost incompressible, and the Poisson's ratio is measured to between 0.48 and 0.4999. (For complete references, see [101, p. 118].)

The reason for us using nearly incompressible values of ν in the *conversion formulas* is the following: We calibrate our computations against real world impact experiments from Hardy et al. [37] later on, and compute the shear strain in order to compare it to the computations by Zou [101]. It is therefore very important that our shear modulus is computed with the use of a nearly incompressible ν when testing against such short accelerations as those of Hardy et al. The nearly incompressible experimental ν-values are found in investigations of parts of the brain, so a shear modulus computed from those values would simulate the brain tissue's ability to fall into shear strain, and that is of interest for us in our calibration procedure. Thus, we choose to compute μ directly from the experimental values of E and the nearly incompressible ν-values, but to compute λ from our μ and a more compressible Poisson's ratio. From the numerical point of view, we need a more compressible brain to reduce locking problems [2, 9] that otherwise occur when using Lagrangean elements in the Finite Element (FE) method (see Section 4.7.1). Most important for the compressible λ-choice is the fact that the brain acts more compressible during acceleration over larger time durations as for example shaking (see the Poisson's ratio Section 4.4.2 above). To check that we reduce the numerical locking effects, we compute the resulting Poisson's ratio from our μ and λ and assure that it is not to close to 0.5.

Which conversion formulas can be used when ν is nearly incompressible? Unfortunately we discovered that some of the conversion formulas show a non-realistic strange behaviour when $\nu \sim 0.49$, and thus affect our computations of μ. In our first 1 g acceleration computations, we computed μ from experimental values of K and ν. By using realistic values of K and ν, we got a completely non-realistic value of μ: $4.6 \cdot 10^7$. It should have been of size 10^4.

For investigating this behaviour we wrote a program which tested several conversion formulas until we had identified the problem nature (*LameParameters.py* in the Appendix A.3). We also tested the conversion formulas analytically to assure that the floating point representation of numbers did not cause the problem. The cause of the non-realistic behaviour was found to be the factor $c = 1 - 2\nu$ that appears in several of the conversion formulas (see the formulas in the Appendix program). We therefore investigated the array $\nu = 0.49, 0.499, 0.4999, 0.49999$. For each new decimal number 9 that is added to ν, c becomes ten times smaller and thus reduce the computed elastic modulus with a factor ten (see Table 4.5.2).

All formulas using the factor $1 - 2\nu$ should therefore be avoided when using nearly incompressible values of ν. As we see from the formulas in the program below, it is much safer to compute μ from E and ν instead. We choose this conversion

Input Poisson's ratio ν	Computed shear modulus μ (Pa)
0.4	$4.92857 \cdot 10^8$
0.49	$4.63087 \cdot 10^7$
0.499	$4.60307 \cdot 10^6$
0.4999	$4.60031 \cdot 10^5$
0.49999	$4.60003 \cdot 10^4$

Table 4.5.2: Values of μ computed from K and ν in the program *LameParameters.py*. K is constantly $2.3 \cdot 10^9$ Pa.

formula in the static linear elasticity simulations (*LinearElasticitySimulator.py*). As we see below, this way of computing μ behaves physically correct with a large ν. The results from the program are as follows:

```
Here E is constantly 150e3
Nu =  0.4
Mu = 53571.4

Nu =  0.49
Mu = 50335.6

Nu =  0.499
Mu = 50033.4

Nu =  0.4999
Mu = 50003.3

Nu =  0.49999
Mu = 50000.3
```

This is a more reliable result physically.

4.6 Acceleration field

We need an acceleration field for use in our elasticity simulations in chapter 6. We use acceleration data from Hardy et al. [37] when we calibrate our FE-solver against real-world experiments and the strain computed by Zou [101]. We use adult human experimental test values for calibrating our FE solver since there are no experiments with infant subjects.

There is one main drawback with using the acceleration data from Hardy et al. [37, 38] for modeling brain elasticity. The acceleration in Hardy et al. is measured with a rigid NAP configuration rigidly attached to the skull, so the NAP thus measures the skull acceleration. Our acceleration bracket is also rigidly attached to the skull. So even if we measure a downwards z-acceleration in the middle of the head, this will not be the true acceleration if there is a brain and CSF inside the skull. A downwards z-acceleration will force the skull downwards, but the brain will be stretched upwards first until it will be pushed downwards due to contact with the membranes surrounding the brain, or due to some flow/turbulence phenomena of the CSF. Therefore, the real acceleration values in the brain will be close to the rigidly measured values, but there will be a delay (phase shift) in the real acceleration values.

We do not take into account the angular velocity in the static elasticity simulations. This is because we assume that the motion changes direction at the acceleration peak time step (see Section 3.2.3). Therefore, the angular velocity ω (and thus the centripetal acceleration $a_c = r\omega^2$) is nearly zero (see Section 3.2.3). This can also be seen in our shaking experiments (see Figure 3.5.2). This is also the case in the impact simulations of Hardy et al.'s (2001) [37], where the angular velocity mostly arises after the angular acceleration.

When we calibrate our elasticity computations against Zou's strain values, we use the same acceleration values as Zou uses for his 2D simulations in the sagittal plane [101, p. 123] (see Section 2.1.2 for a definition of sagittal). He computes the strain values, and it is interesting for us to compare our results with his, and calibrate our FE-solver to his strain results. Zou uses acceleration values from Hardy et al. (2001) [37, p. 7], but in his table Zou writes the resultant acceleration. This is the length of the 3D linear acceleration vector and the 3D angular acceleration vector respectively. We do not know if he uses this resultant for his sagittal plane simulations. On page 73 he writes "Note that only the experimental data in the sagittal plane are used for the planar model" and this should be a more correct method for both ours and Zou's 2D sagittal plane model. Zou also uses data from tests C383-T3 and C383-T4 from Hardy et al. (2001). These tests show a large angular acceleration around the z-axis, which we do not have in our peak acceleration shaking experiment. Therefore we disregard these tests in our simulations.

For testing the validity of our FE-solver we use some acceleration values from Hardy et al. (2007) [38] and Hardy et al. (2001) [37]. There are problems with using all test values from [38] since the accelerations are mostly too large for making good simulations with linear elasticity only, and all acceleration and strain values from all tests are not printed in the article [38].

We also want to simulate the peak of our shaking experiments, and construct a static acceleration field for this purpose.

All acceleration values we use for our elasticity simulations can be found in Section 4.6.1.

We have to translate the angular- and linear acceleration to a x-z-coordinate field before using them in the elasticity computations, since our mesh is a slice in the x-z-plane. This is most easily done with the equations (3.3.27) to (3.3.29) which we restate here.

$$a_{p_x} = -p_x(\omega_y^2+\omega_z^2)+p_y(\omega_x\omega_y-\dot{\omega}_z)+p_z(\omega_x\omega_z+\dot{\omega}_y)+a_{0_x} \quad (4.6.1)$$

$$a_{p_y} = p_x(\omega_x\omega_y+\dot{\omega}_z)-p_y(\omega_x^2+\omega_z^2)+p_z(\omega_y\omega_z-\dot{\omega}_x)+a_{0_y} \quad (4.6.2)$$

$$a_{p_z} = p_x(\omega_x\omega_z-\dot{\omega}_y)+p_y(\omega_y\omega_z+\dot{\omega}_x)-p_z(\omega_x^2+\omega_y^2)+a_{0_z} \quad (4.6.3)$$

The p_i are here the coordinates for the node in the mesh, and a_{p_i} are the corresponding linear accelerations at the node.

The linear acceleration a_{0_i} works as a translation vector changing the rotational acceleration center. This means that if we have two acceleration fields with the same angular acceleration and angular velocity, there will always exist linear acceleration vectors that can transform the acceleration fields into each other. Some figures that show this behaviour are 4.6.1 and 4.6.2.

Hardy et al. (2007) [38, p. 6] use a point in the sinuses as origo, and this means that our choice of origo is rather close to theirs. Hardy et al. (2001) use an external NAP which they attach to the vertex. Therefore, we should actually move our coordinate system for the mesh before applying their values. In their acceleration value table [37, p. 7], however, they write "Linear c.g. acceleration (g)" so it seems like they have computed the acceleration in the center of gravity. Therefore we use their values directly.

4.6.1 Acceleration values used in the FE elasticity simulations

The resultant acceleration values from Zou [101, p. 123] are in Table 4.6.1. We use these values for calibrating our FE-solver against his computed strain. (For an explanation of the sample- and test-numbers, see Section 2.5.)

For our elasticity tests we also use some impact acceleration values from Hardy et al. (2001) [37, p. 7] (see Table 4.6.2). All of their tests are impacts in the sagittal plane, and the max linear acceleration thus occurs in the x-direction, and the max angular acceleration is around the y-axis.

The acceleration values we use from Hardy et al. (2007) [38] (see Table 4.6.3) are one low acceleration test where we know the strain values (sample C241), in addition to some tests where the Neutral Density Targets (NDT) are placed on the surface of the cortex – the tests with sample C408. Sample C408 is especially interesting since the position of the NDT makes it possible to carefully measure the

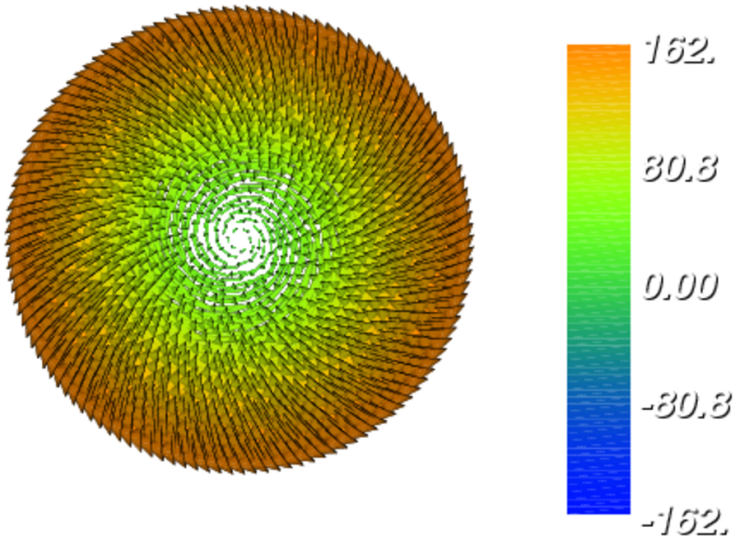

Figure 4.6.1: Rotational acceleration field (m/s^2) without linear acceleration in origo.

Test	Linear resultant acceleration (g)	Angular resultant acceleration (rad/s/s)	Computed linear elastic shear strain
C755-T2	21.8	1753.2	0.079
C755-T3	24.4	1948.7	0.073
C755-T5	12.1	803.9	0.035
C383-T1	62.2	3033.3	0.102

Table 4.6.1: Acceleration resultant values from Hardy et al. (2001) [37] used in Zou [101, p. 123]. The shear strain is computed by Zou.

relative motion at the brain/skull boundary (see Section 2.5). Many of the Hardy et al. (2007) tests were not in the sagittal plane which we need for our simulations. The displacement between the skull and brain for sample C408 is only printed for test T4 [38, p. 53], but that test also shows a very large angular acceleration around the z-axis [38, table 6] so it does not suit our problem perfectly.

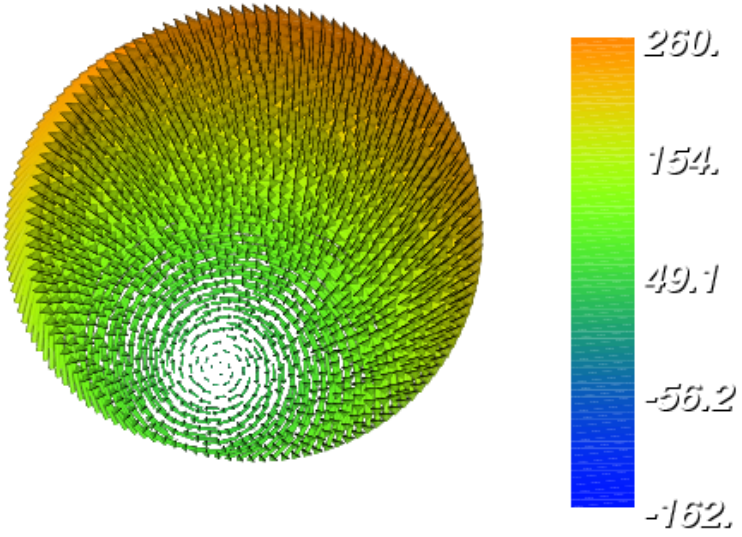

Figure 4.6.2: The same angular acceleration as in figure 4.6.1, but with an added linear acceleration of 10 g. We see that the rotational acceleration center is moved. Unit is m/s^2.

Test	Linear max x-acceleration (g)	Angular y-acceleration (rad/s/s)
C731-T2	20	1227
C731-T3	27	1695
C755-T2	21	1882
C755-T3	23	1995
C755-T4	27	2681
C755-T5	12	843
C381-T1	46	2592

Table 4.6.2: Acceleration values from Hardy et al. (2001) [37, table 2]

Shaking peak acceleration field

The acceleration values used for simulating the shaking peaks from our data set are approximated such that a baby sized head mesh gets an acceleration at the vertex of the right size, and a rotation acceleration center near the real position of the rotation axis (see Section 5.5). We suppose here that the rotation axis is placed close to the point where the acceleration field is zero – the rotational acceleration

Test	Linear max x-acceleration (g)	Angular y-acceleration (rad/s/s)	Max shear strain
C241-T3	53	2629	0.026
C408-T3	43	5980	0.014
C408-T4	154	7506	0.020

Table 4.6.3: Acceleration and max shear strain from Hardy et al. (2007) [38]. Remember that the shear strain is computed at the (NDT) positions.

center (see Section 3.2.2).

The max acceleration at the vertex is 105 g = 1038 m/s^2, and the rotation axis is at that moment placed slightly below the x-sensors (see Section 5.5). The acceleration at the x-sensors are 30 g = 294 m/s^2 (see Section 5.1). Our estimate of the acceleration field can be seen in Figure 4.6.3, and we see there that the acceleration at the x-sensor positions on the accelerometer bracket is actually approximately 290 m/s^2 as measured. Another way of finding the acceleration field from our measurements would be to use an optimization process for finding a field agreeing with our measured values. This could be repeated at each time step.

Especially interesting in our choice for the shaking peak acceleration field, is the angular acceleration magnitude needed for being able to place the rotation axis close to the right position: 10500 rad/s^2. This is above the magnitude of several high-impact tests from Hardy et al. (2007) [38], and should cause a large rotational force at the bridging veins.

4.7 Numerical methods for elasticity

The most common numerical method for modeling head injuries is the Finite Element (FE) method [62, 55, 39, 55, 47, 46, 79, 20] . Zou is an exception [101] and develops an analytic model where the brain is attached to the skull with springs. He uses numerical methods for computing the strain, and he validates his results by comparing it with the Kleiven model [101, p. 189] and the displacement data from Hardy et al. (2001) [37].

4.7.1 The Galerkin Finite Element method

We apply a class of FE methods called Galerkin [51, 9]. The main ideas are the following. First, to subdivide the domain into smaller domains with easier geometry (*finite elements*) for solving the equation over each element. Second, to project solution onto a discrete set of functions on the element (*basis functions*). Third,

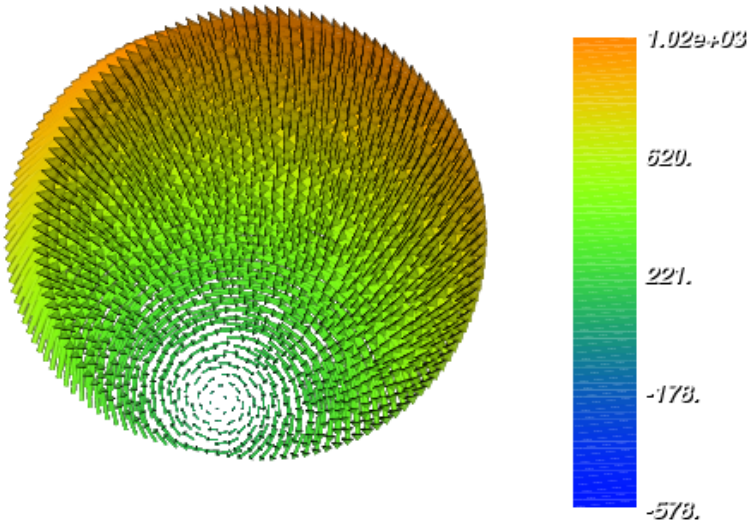

Figure 4.6.3: The acceleration field used for simulating our peak acceleration in our shaking experiments. The angular acceleration is chosen to be 10500 rad/s^2, and the linear acceleration is 45 g. The head diameter is here 0.11 m.

to multiply the resulting discrete PDE with an arbitrary *test function* for turning it into a minimization problem of the residual error [28, p. 513] [51, p. 142]. Fourth, to add all data (*assemble*) from each element into a large matrix where careful numbering of the entries corresponds to the numbering of all elements, and use the matrix for solving for the solution vector linearly if the original PDE is linear. In the Galerkin's method, the test functions belong to the same space as the basis functions. Our mesh with the triangular elements can be seen in Figure 4.7.1.

After multiplying with the test function in the Galerkin method, it is common to use integration by parts for simplifying the results. The equation one achieves is called the *weak formulation* (or simpler – the weak form). It is the weak form that is the input to the FE solver DOLFIN in the FEniCS software that we use (see Section 4.8). The weak form of the static elasticity equation (4.2.3) with isotropic material and only homogeneous Dirichlet boundary conditions on $\delta\Omega$ is

Find $\mathbf{u}$ in $\mathbf{H}^1(\Omega)$ such that $\mathbf{u}|_{\delta\Omega} = 0$ and

$$a(\mathbf{u},\mathbf{v}) = -\int_\Omega \rho\mathbf{b}\cdot\mathbf{v}dx \tag{4.7.1}$$

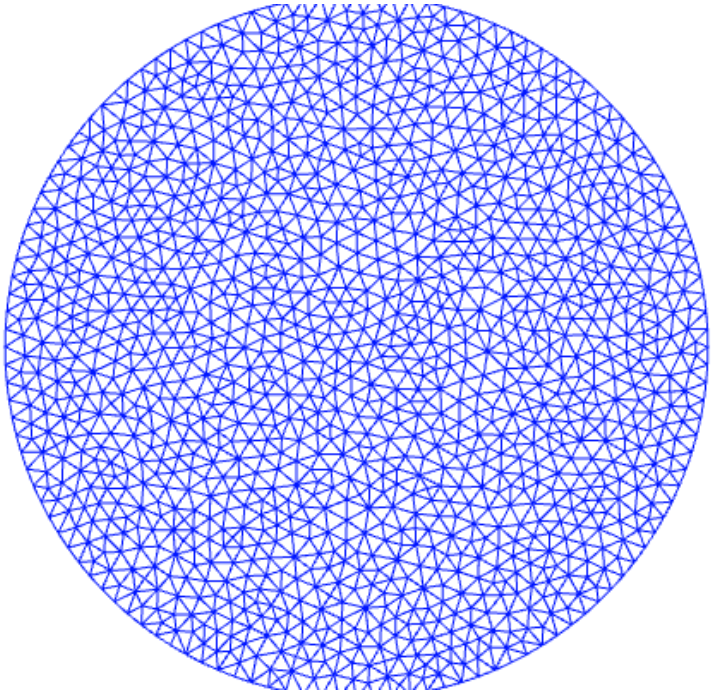

Figure 4.7.1: Our mesh used in the elasticity computations. The plot unfortunately cut the top and bottom of the mesh – it is in fact circular. A scaled version is used for the baby head simulations.

for all $\mathbf{v} \in \mathbf{V}$, where

$$a(\mathbf{u},\mathbf{v}) := \int_\Omega \lambda \nabla\cdot\mathbf{u}\nabla\cdot\mathbf{v} + 2\mu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})dx \tag{4.7.2}$$

$$\mathbf{V} := \{\mathbf{v} \in \mathbf{H}^1(\Omega) : \mathbf{v}|_{\delta\Omega} = \mathbf{0}\} \tag{4.7.3}$$

and $\mathbf{b}$ is the body acceleration. The weak form has a unique solution due to the coercivity of $a(\cdot,\cdot)$ established by Korn's inequality [9, Chpt 11]

Please note that in [85, formula 2.2.23] μ actually means 2μ which is the same as $2G$ (see page 13 in [85]). See Section 4.3 for an explanation of the constants (the Lamé parameters).

The discretization process (automatic in DOLFIN) is to let $u := \sum_{j=1}^{n} u_j N_j(x)$ where n is the number of nodes, and let $v := N_i,\ i = 1,...,n$ resulting in n numbers of equations to solve. After that one solves the equation over each element where $n := \tilde{n} =$ number of nodes in the element, and assembles the data.

The elements we use are Lagrangean triangle elements of order one. This means that the basis functions are linear, the number of nodes are three in each element and are placed at the corners. Computations with Lagrangean elements

run into problems (locking) if the elastic material is nearly incompressible [3, 2]. The locking means that one get unphysically small deformations when Poisson's ratio is close to 0.5. In our case we use a slightly compressible brain for reducing locking effects.

A lot of other elements for use in the FE method also exist. These elements are developed for special kinds of computational problems. We can avoid the locking problem with use of mixed elements instead [9], or some other specialized elements [57].

Our acceleration measurements obviously use Lagrangian coordinates, which means that the accelerometers have the same body coordinates throughout all motion. These Lagrangian coordinates simplify our computations a lot, since we can think of the head as non-moving, while the accelerating force varies. Thus we get homogeneous Dirichlet boundary conditions instead of a rigidly moving boundary. The limitations of this view is, as mentioned above in Section 4.6, that the brain is not rigid, but the acceleration data is measured on a rigid skull or bracket.

The acceleration measured in the dummy head is actually emerging from the global coordinates in the rigid motion (see Section 3.3). Thus, we use an acceleration field based on the measurements, as the external force in the elasticity modeling. Then we achieve the static linear elasticity equation (4.2.3) with homogeneous Dirichlet BC

$$(\lambda+\mu)\nabla(\nabla\cdot\mathbf{u})+\mu\nabla^2\mathbf{u}=-\rho\mathbf{b} \tag{4.7.4}$$

$$u=0, \quad x\in\delta\Omega \tag{4.7.5}$$

where $\mathbf{b}$ is the acceleration field measured in the head. If one takes into account the time-dependency of the elasticity waves, one simply adds the term $\rho\mathbf{u}_{tt}$ on the right hand side. (Hereafter we define $u := \mathbf{u}$ for simplicity.)

It is easier to verify our simulations with use of acceleration input as in equations (4.7.4) than with use of a rigid motion of the boundary, since the experimental values that exist are in the form of acceleration and angular velocity [37, 38].

To be able to choose the right values of the elasticity parameters, we have to compare the results of the numerical model with real world experiments. Then we are able to scale or adjust the parameter values for simulating the experiments as correctly as possible. So we test our elasticity simulations with the different values of E and ν discussed in Section 4.5, and choose an optimal value of μ while comparing it with the real world shear strain values. Afterwards we validate that our modeled brain is compressible, and compute λ with use of a more compressible value of ν before solving the elasticity equation to reduce locking effects (see Chapter 6). This is done in the program *LinearElasticitySimulator.py*

printed in Appendix A.1. The program has the possibility of testing different acceleration fields, head size, Lamé parameters, age-dependent elastic moduli, and also computing the displacement, shear strain, and max shear strain. The default acceleration field is our peak acceleration field developed for a baby sized head (see Section 4.6.1).

4.8 Program packages

The basis programming language for our computer programs is Python (*python.org*). We frequently use the packages Numpy (*http://sourceforge.net/projects/numpy/*) for numerical computing with arrays, and Matplotlib (*http://matplotlib.sourceforge.net/*) for plotting our results. For solving our acceleration equation in chapter 3 we use the symbolic solver Swiginac (*http://swiginac.berlios.de/*). Our elasticity simulations are done with the use of the FEniCS project [24], and especially the main interface to FEniCS – DOLFIN.

Chapter 5

Acceleration – results and discussion

In this chapter we present some results from the methods presented in Chapter 3. First we describe some important characteristics of our shaking peak acceleration data set, and compare the results with other biomechanical experiments from the literature. After that, we try to understand the shaking motion of this specific data set. The chapter ends with some numerical properties of our new accelerometer configuration from Section 3.4.

The main investigation of this thesis is based on our peak acceleration shaking data set. The shaker managed to get the highest peak acceleration in this shaking session, so the test is a model for a worst case scenario (see Figure 5.0.1). The shaker did not shake the dummy in the forwards-back direction as in Duhaime et al. and Prange et al. [26, 73], but shook it upwards-down. Cory and Jones [19] also found that this shaking direction gives the highest peak acceleration magnitude.

5.1 Peak acceleration magnitude

Our peak x-direction acceleration magnitude at the vertex is 105 g, and the peak x-sensor acceleration at the accelerometer bracket is 30 g. Duhaime et al. [26] only mention the mean peak acceleration, which they have found to be 9.3 g. Prange et al. [73] measured the angular velocity only, and computed the acceleration afterwards. It is difficult to read the acceleration values in their article due to low resolution. Cory and Jones [19] have achieved 177 g as peak tangential acceleration. This peak occurs when the head hits the back and chest – a shaking-impact.

We can compute our angular acceleration at the acceleration peak, with help from the rotation axis position from Section 5.5, where the rotation axis is found to be situated 0.023 m below the x-sensors. The radius from the vertex accelerometer to the rotation axis position is at that moment 0.023 + 0.0579 m. By using the

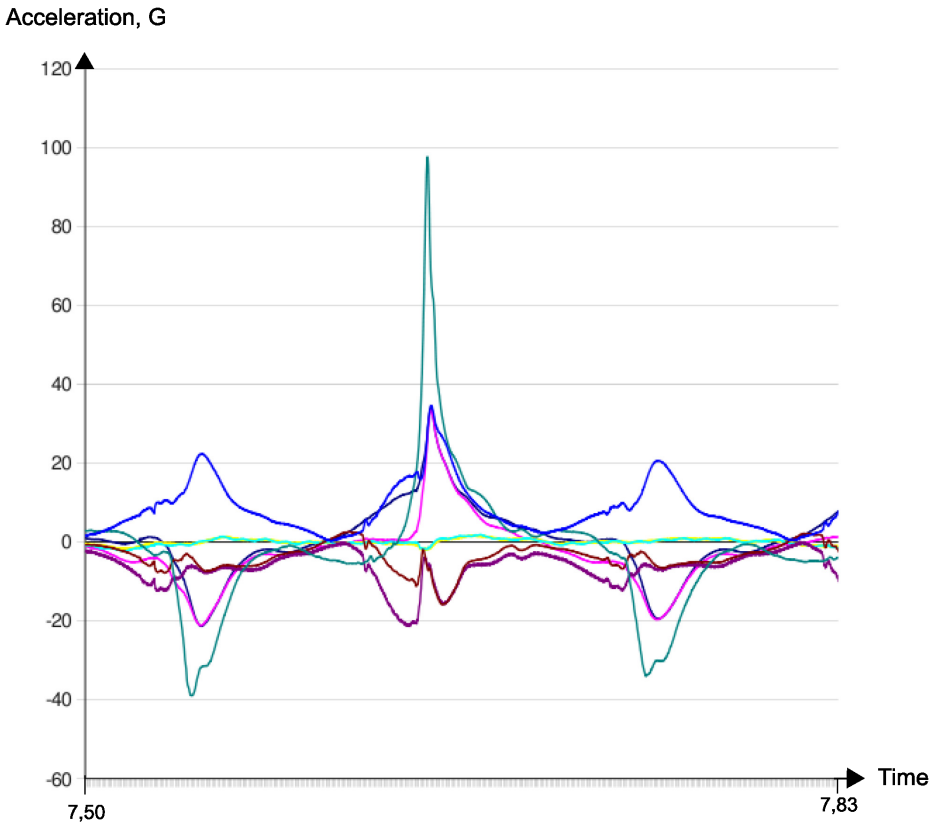

Figure 5.0.1: A typical cut from our peak acceleration shaking session. (The same figure as 3.5.2.) The dark green curve is the vertex measurements (x-direction); the pink and dark blue is the x_1 and x_2 sensors; the cerise and brown are z_1 and z_2; the yellow and light green are y_1 and y_2; and the always positive blue is the absolute acceleration. See Section 2.7 for the accelerometer positions.

formula $a_t = r\alpha$, we get an angular acceleration of 12900 rad/s^2. Notice that this angular acceleration is higher than the angular acceleration used for our shaking peak acceleration field in Section 4.6.1. The reason for this is that we also need a "false" linear acceleration in our acceleration field, for changing the *rotational acceleration center* from the origo to the position of the *rotation axis*. The real acceleration field should be simulated with an origo in the rotation axis position, and with the angular acceleration of 12900 rad/s^2. The angular acceleration is actually larger than in high-acceleration impact experimental tests by Hardy et al. [38].

We also compute the mean angular acceleration from Duhaime et al. [26]. We

use their mean peak tangential acceleration, and their estimation of the rotation axis position 3.3 cm below the skull base. However, we are unable to reproduce their angular acceleration in [26, Table 6, p. 412].

The highest level of peak acceleration in our experiment was measured at the upper turning point.

The acceleration in our peak acceleration shaking test in the z-direction is almost never positive, and always directed downwards into the body. This is natural, since we have a centripetal force in the shaking motion. The magnitude of the z-acceleration is between 2 and 20 g (see Figure 5.0.1).

5.2 Time duration of the acceleration peak

In this section "acceleration peak" means the whole time period from almost zero acceleration, through the maximum acceleration moment, and ending at the almost zero acceleration time step. In most other sections of this thesis, "acceleration peak" means the moment (time step) of the maximum acceleration of the vertex. The meaning of the words will be clear from the context.

Our time duration of the acceleration peak is 54 ms. We measure the time duration as the difference from zero acceleration before the peak, and until it has decreased back to 1 g. After the vertex has reached the 1 g acceleration, it enters a rather long "resting period" of low-value acceleration (see Figure 5.0.1). The graphs of shaking in neither Duhaime et al. [26] nor Prange et al. [73] show these resting periods behaviour.

The vertex acceleration in our measurements increases much faster than the x-sensor measurements at the accelerometer bracket. The time duration for the x_1 sensor's corresponding acceleration peak is 87 ms. The explanation for this inconsistence is that the vertex actually accelerates in the negative direction, at the start of the x_1 sensor's positive acceleration peak. This behaviour is explained in Section 5.6.

Now we compare with shaking peak durations from earlier biomechanical experiments. The Duhaime et al. [26] mean time peak duration is longer than ours, 107 ms. This is natural since we study only our worst case in this thesis, and Duhaime et al. study the mean value. Prange et al. [73] also report a longer time duration of the acceleration peak – 180 ms.

Perhaps it is better to compare the frequencies of the shaking cycles than the time duration of the peaks, since it is so difficult to decide where the peak in our case starts and ends. The frequency in our peak acceleration test is 5 Hz. It would also be interesting to compare the length of the "resting periods" between the peaks, and the derivative of the acceleration measurements.

Cory and Jones [19] observed impacts between the skull and the body in their worst case tests. We can not completely exclude the possibility of a shaking-impact in our shaking experiment. However, we did not observe impacts at all, and it can not be seen on the video tape. We also assume that the acceleration will change direction at all x-sensors at the same moment, if we have an impact. The opposite behaviour is present in our case.

5.3 Velocity and length of shaking cycle

We integrate the vertex acceleration for finding the max velocity and the length in space of the shaking (see Sections 3.6 and 3.2.3). The max tangential velocity is 8 m/s, and occurs at the time step where the acceleration is zero, between the positive and negative acceleration peaks. The max length of the vertex motion curve is 71 cm. The velocity and length can be seen in Figures 5.3.1 and 5.3.2.

We compare our velocity results with the velocity from Duhaime et al. [26, p. 412]. The following computations are based on the formulas for the relation between angular and tangential acceleration, and angular and tangential velocity – $a_t = r\alpha$ and $v_t = r\omega$. Duhaime et al. have computed the angular acceleration with the use of a hypothetical rotation axis 12.3 cm below the vertex, and integrated the angular acceleration for reaching an angular velocity of 60.68 rad/s. We compute the Duhaime et al. tangential velocity from their angular velocity (and with their estimated rotation axis), and get 7.5 m/s, very close to our result. However, we also simulate their shaking tangential acceleration curves with a sinusoidal curve, developed from their mean peak acceleration data and the frequency in [26, fig. 1] (see Section 3.6). We integrate the simulated shaking curve from the peak moment to the zero velocity time step, and get a max tangential velocity of only 4.3 m/s. Since Duhaime et al. used a rather large distance to the rotation axis when computing the angular acceleration, they should have got low angular acceleration values, which would have given low angular velocity in their integration. We do not understand this inconsistency of the Duhaime et al. velocity magnitudes. However, converting between angular and tangential accelerations with the use of hypothetical rotation axis position as in Duhaime et al. [26] may cause the problem. Wolfson et al. [97, p. 65] has used the same size of acceleration as in Duhaime et al., but he has also computed lower velocities. The sinusoidal simulation of the Duhaime et al. shaking can be seen in Figure 5.3.3.

Prange et al. [73], did not compute the tangential velocity. They only measured the angular velocity. There is no data in their article on the distance from the vertex gyro, to the hinge rotation axis, so we can not compute their tangential velocity.

Cory and Jones [19] report a peak tangential head velocity between 7 and 10

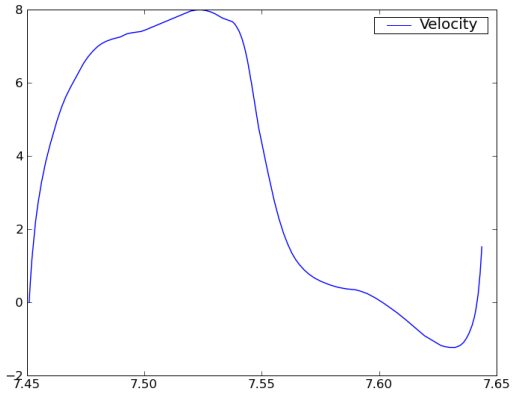

Figure 5.3.1: The vertex velocity for the time between two subsequent acceleration peaks from our peak acceleration data. We have not subtracted the gravity from the measurements. Therefore the velocity decreases to slowly. However, the max velocity occurs at the same moment as the zero acceleration. The x-axis unit is seconds (s).

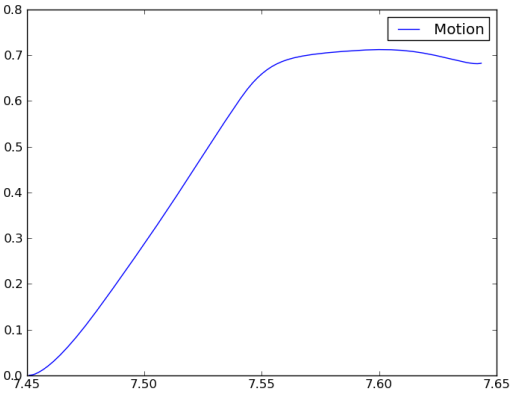

Figure 5.3.2: The motion for the vertex for the time between two subsequent acceleration peaks from our peak acceleration data. We have not subtracted the gravity from our measurements. Therefore the max real motion is lower, and occurs earlier than in the figure. The x-axis unit is seconds (s).

m/s, which is close to ours.

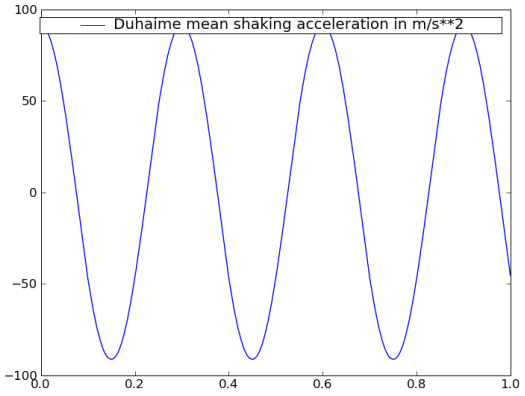

Figure 5.3.3: Our simulations of the mean Duhaime et al. [26] vertex acceleration. The x-axis unit is seconds (s).

5.4 Influence of the neck elasticity

We shook the dummy so violently that we got a rift in the rubber at the front side of the neck. When the front side broke up and the steel string that held it together lay open, the acceleration actually diminished. Before we got the rift we could reach a peak acceleration of 30 g at the x_1 and x_2 sensors. After the rift, we could only reach accelerations up to 15 g. So the elasticity of the rubber neck actually increases the peak acceleration while shaking.

The authors of the earlier biomechanical models [26, 73, 19, 97] believe that the worst case acceleration in SBS happens when the neck is close to a hinge, with no elasticity in the neck. This is also the assumption in [18]. This is not the case in our simulations. One explanation of why the hinge neck gave highest acceleration in earlier experiments may be that the hinge limits the motion to the x-z-plane only. Therefore, no energy is used for accelerating around other axes than the y-axis. Our neck may also make it easier for rotation in the x-z-plane due to the holes at the front rubber section (see Section 2.7).

Another explanation may be that with a hinge neck without resistance, one often gets impacts of the head onto the torso. The acceleration peaks with a hinge neck therefore also gets contributions from the impacts, which may not be the case with real and perhaps more resistant baby necks. However, the real baby neck elasticity and mobility is not known.

5.5 Validation of the acceleration measurements

The analytical solution of the rotation axis position at the acceleration peak is computed with the methods from Section 3.5. It is found to be 0.023 m below the x_i sensors, and 8 cm below the vertex accelerometer. This means that the rotation axis is in the lowest part of the accelerometer bracket at the acceleration peak, i.e. inside the lower parts of the brain. A rotation axis inside the brain leads to a higher shear strain than if the rotation axis had been more far away.

The numerical method for finding the rotation axis position with use of the accelerometer measurements from the accelerometer bracket sensors only, requires more time for implementation. We will do this in our further research.

The length (in space, not in time) of the shaking cycle is another indication of the validity of our measurements. The double integral of the vertex accelerometer measurements shows that the max length that the vertex travels between two positive acceleration peaks (one cycle) is 71 cm (see Sections 3.6 and 3.2.3). This suites well with what we see in the video recordings of the shaking. The largest contribution to the integral comes from the acceleration peaks. If the vertex accelerometer had measured a too large acceleration, we would have got a too large distance in space between the acceleration peaks.

5.6 Understanding the shaking motion

The acceleration curves in Figure 5.0.1 show an interesting difference between the accelerometers x_1 and x_2, and z_1 and z_2, which may be explained by a motion of the rotation axis into the accelerometer bracket. The x_1 and x_2 accelerations differ with 10 g, during a time duration of 30 ms, just before the positive acceleration peak. As we can see in Figure 5.6.1, the position of the rotation axis close to the sensor bracket causes different centripetal accelerations (and also different tangential accelerations not showed in the picture) at the sensors. This will surely result in different uniaxial measurements along the accelerometer sensing axes. One exception is when the rotation axis is situated on the z-axis (see Section 3.5), which leads to coinciding measurements of the x-sensors.

At the same time period as the x_1 and x_2 difference, we see that the acceleration at the vertex is negative (see Figure 5.0.1), which means that the vertex is still accelerating upwards, while the wave motion of the flexible neck (induced by the chest moving downwards), causes the lower part of the head to accelerate downwards. When the torso starts to move downwards (and the head is still moving upwards), the rotation axis is continuously moving from somewhere down in the neck, up into the interior of the head. This movement is due to the wave motion in the neck caused by the torso's new direction of motion. So the vertex is still mov-

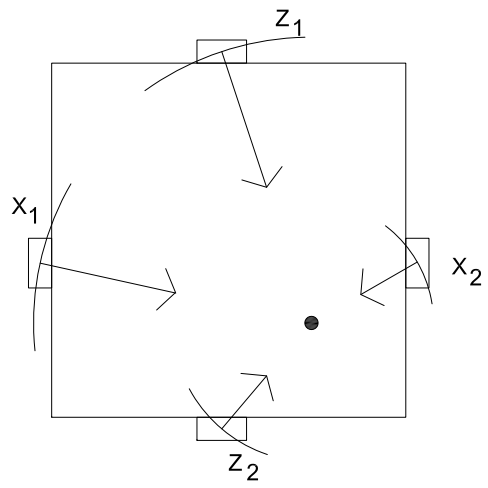

Figure 5.6.1: Different centripetal acceleration magnitudes at the accelerometers, with the rotation axis inside the bracket (the black dot).

ing upwards, and the neck is moving downwards, until the neck is bent enough for the inner head rotation to end, and the whole head moves in the same direction as the torso. The rapid positive acceleration increment that follows at the vertex, lasts only 15 ms. At the acceleration peak, the x_1 and x_2 sensors synchronize and reach the same magnitudes, since the rotation axis at that moment is placed on the z-axis (see Section 3.5). The moment where the neck is bent maximally, and the head direction of motion has to change in a very short amount of time, gives the maximum acceleration measurements. The explanation above describes a fast whiplash motion of the head, with a naturally moving rotation axis.

5.7 New accelerometer configuration

In this section we present some numerical simulations with our new accelerometer configuration from Section 3.4.

We used the acceleration measurements from the accelerometer bracket to simulate an acceleration at the vertex. Since our new coordinates rotate the sensing axes of the new "virtual" accelerometers 45 degrees compared to the old sensing axes, our vertex simulations will certainly not coincide with the real vertex

measurements. We only hope to get a similar shape of the computed vertex "acceleration", without too much noise. The result of our vertex acceleration computations can be seen in Figure 5.7.1. As we see, the computations blow up the noise from the measurements a bit, but not too much. The shape is also similar to the real vertex acceleration. Our new configuration shows an error where the angular acceleration changes sign, but not as large as the NAP developed in Padgaonkar et al. [70, p.555]. The error in our computations is of the same size at all time steps as it should be, since the condition number is only dependent on the coordinates of the accelerometers (see Section 3.4).

The noise in the measurements is between 0.02 and 0.4 g – a different size of noise from different accelerometers. This error is blown up to 1 g in our computations, and is a sign of a condition number size much larger than 1. The condition number is computed in the program *ConditionNumber.py* (see the Appendix A.4.1), and is found to be 98.7. This is a rather high value, and shows that the coordinates of our new accelerometer configuration developed in Section 3.4 are non-optimal. We need to further study the coordinate positions of the accelerometers, to find a configuration with a lower condition number. One way of improving the coordinates would be to move all sensors away from the coordinates axes, in order to prevent the annihilation of the degrees of freedom. In our configuration we let the x-sensors to be placed on the y-axis (see Section 3.4).

However, since the noise in the measurements is low, the computed acceleration in our case has also a reasonably sized noise. The noise of the computations may be reduced further by smoothing the measured accelerations first. If we also smooth the resulting acceleration, we can reach an acceptable size of the error.

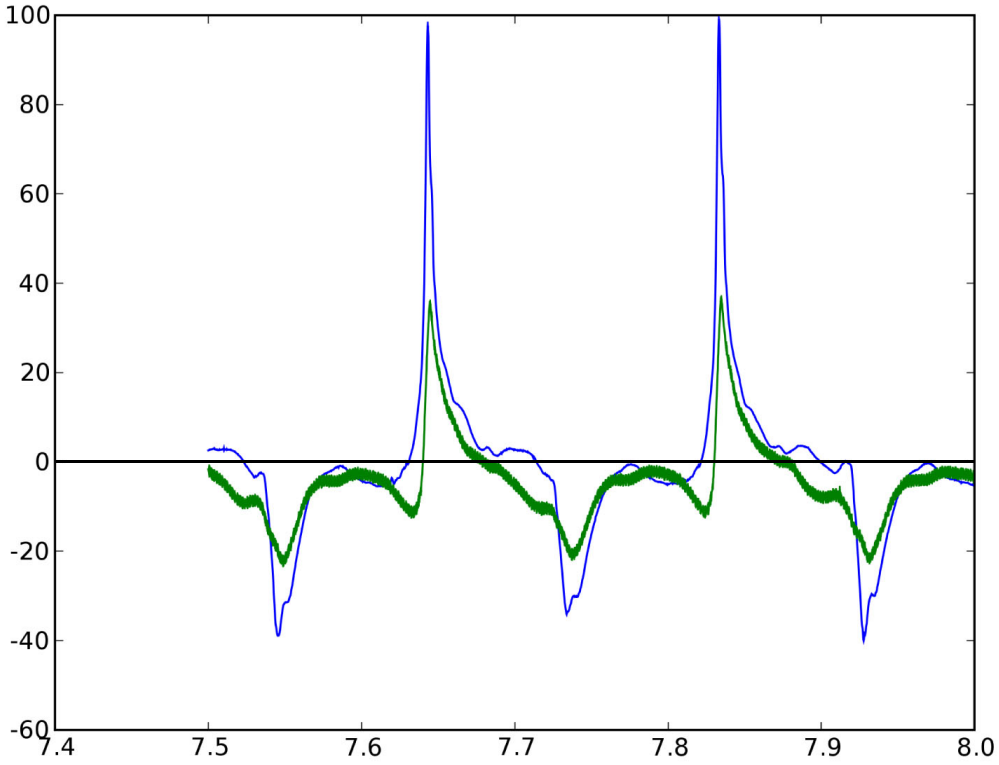

Figure 5.7.1: Computed values of vertex acceleration (green) in new coordinate system from Section 3.4, plotted against our measured values from the vertex (blue) in the old coordinate system. It shows a coinciding shape, and a not too large noise, not even at the zero acceleration points where the angular velocity direction changes. The noise is of the same size at all time steps, since the condition number of the matrix (3.4.4) only is dependent on the accelerometer positions. (The acceleration values are in multiples of 1 g.)

Chapter 6

Elasticity – results and discussion

In this chapter we present the results of our elasticity FE computations, and compare them with results from literature. Our first goal here is to calibrate our FE-solver against known shear strain values from human impact experiments. We will choose values of the Lamé parameters such that we can reproduce the known strain values. Then we will be able to simulate the shaking acceleration peak more realistically than without a calibration. We finish the chapter with displacement and strain computations with use of our peak shaking acceleration field.

The main program we use here are *LinearElasticitySimulator.py* (see the Appendix A.1). In our first simulations we use only adult material parameters, in order to be able to compare the results with the real world experiments by Hardy et al. [37, 38] and the shear strain computations by Zou [101, p. 123].

Several elastic factors are found to correlate with the severity of injury. Some of these are strain, strain rate, maximum pressure and Von Mises effective stress [46]. In our computations we compute the displacement and shear strain. In this chapter, *displacement* means the norm of the displacement vector field.

The shear strain in the figures in this chapter show a non-smoothly distributed strain. The main reason for this is an interpolation error in DOLFIN [24].

6.1 Effect of angular acceleration

We start with choosing the shear modulus μ to be the same as Zou uses in his linear elasticity simulations, so $\mu = 50.3$ kPa. We thereafter try with different sizes of angular and linear acceleration, to see if the angular acceleration in our simulations causes more strain than the linear acceleration – as we expect it to from physical reasoning (see Section 2.1.1). We see in Table 6.1.1 that our simulations confirm this physical behaviour. (Linear acceleration is defined in Section 2.1.1.)

6.2 Suitable values of the Lamé parameters

In this section we compute the shear strain for different values of the brain tissue elastic moduli. We want to check if our numerical FE model shows a reasonable behaviour when varying these parameters.

These simulations are based on the experimental values of the elastic moduli E and ν in Table 4.5.1. We see from the results in Table 6.2.1 that a larger value of E results in a larger value of μ, that results in a smaller shear strains as expected. (See Sections 4.5 and 4.7.1 for a discussion on the methods behind these results).

We also test the elasticity program with different values of Poisson's ratio ν. The results can be seen in Table 6.2.2. They show the expected phenomena, that the shear strain becomes larger with a more compressible brain.

Max acceleration m/s^2	Linear acceleration (g)	Angular acceleration rad/s^2	Max shear strain
104.5	0	1100	0.019
104.5	104.5	0	0.0095
184	12	700	0.021
183	10	900	0.023
183	8	1100	0.024
182	6	1300	0.026

Table 6.1.1: Max shear strain for different angular and linear accelerations. We see that the angular acceleration causes more shear strain than the linear acceleration.

Young's modulus E (kPa)	Poisson's ration ν	Computed μ (Pa)	Max shear strain
10	0.4999	3334	0.33
66.7	0.499	$2.225 \cdot 10^4$	0.049
66.7	0.48	$2.253 \cdot 10^4$	0.048
80	0.49	$2.685 \cdot 10^4$	0.041
250	0.49	$8.389 \cdot 10^4$	0.013
675	0.48	$2.28 \cdot 10^5$	0.0048
1000	0.48	$3.378 \cdot 10^5$	0.0032

Table 6.2.1: Max shear strain for different experimental values of the brain tissue elastic moduli. We used experimental E and ν values from Table 4.5.1 for our μ computations. The acceleration used in these computations is 1100 rad/s^2 (angular) and 4.5 g (linear).

1 g simulations

We also simulate a 1 g linear acceleration with a chosen $\mu = 25000$ Pa which is in the middle of the computed μ values in Table 6.2.1. 1 g acceleration is what humans experience each day on earth. We see in Figure 6.2.1 that an adult head gets a maximum shear strain value of 0.0018 in our computations. The acceleration field is directed straight to the left in the figure. We guess that these shear strain values are too large. The reason for the unrealistic shear strain seems to be the attachment of the brain to the skull in our simulations, since the shear strain is largest close to the walls parallel to the acceleration direction. The corresponding displacement can be seen in Figure 6.2.2. The max displacement is 0.1 mm, and may be of the real size.

6.3 Calibrating with real world adult experiments

In order to assure that our computed shear strains and brain/skull displacements have an acceptable size compared with real world experiments, we need to choose values of the elastic moduli that can reproduce some known experimentally achieved results. We therefore run some of Zou's simulations (see Table 4.6.1) in order to compare them with his shear strain results, and let μ vary in the range of Table 6.2.1. This leads to an indication on which value for μ we could choose to reproduce max shear strain magnitudes close to Zou's results.

For a brain tissue shear modulus μ of 25 kPa, we get slightly too large shear strain values. The correspondence with the Zou values is not linear. Our shear strain values grow faster than Zou's as the acceleration increases. One typical picture of our shear strain is shown in Figure 6.3.1 with the corresponding acceleration field in Figure 6.3.2. It shows that the shear strain is largest near the surface of the brain. The same behaviour is not present in Hardy et al. (2007). They found in their tests that the shear strain is low near the surface. The large shear strain we get in our simulation is therefore caused by the attachment of the brain to the

Poisson's ratio	Max shear strain
0.44	0.044
0.40	0.049
0.35	0.054
0.30	0.058

Table 6.2.2: Max shear strain for different experimental values of Poisson's ratio ν. We used a brain tissue μ value of 25 kPa. The acceleration used in these computations is 1100 rad/s^2 (angular) and 4.5 g (linear).

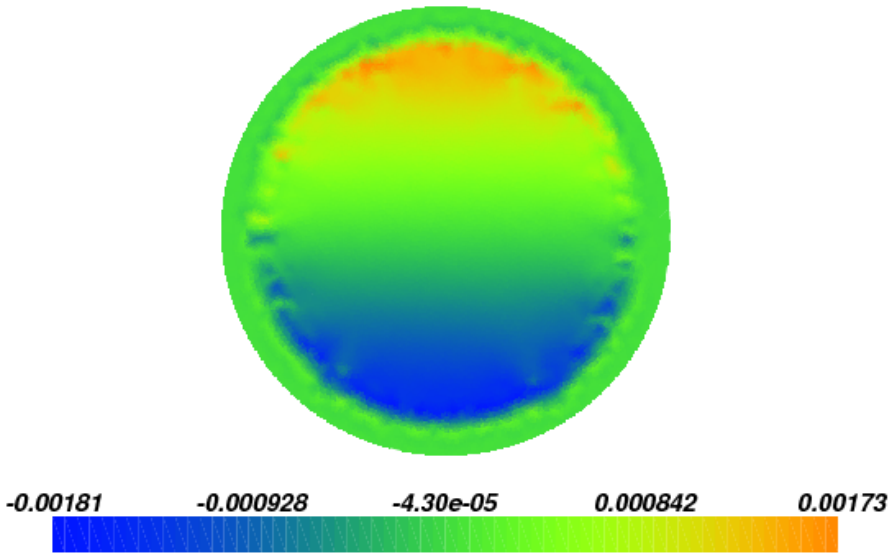

Figure 6.2.1: Shear strain for an adult head and 1g linear acceleration. The acceleration force is directed straightly to the left side.

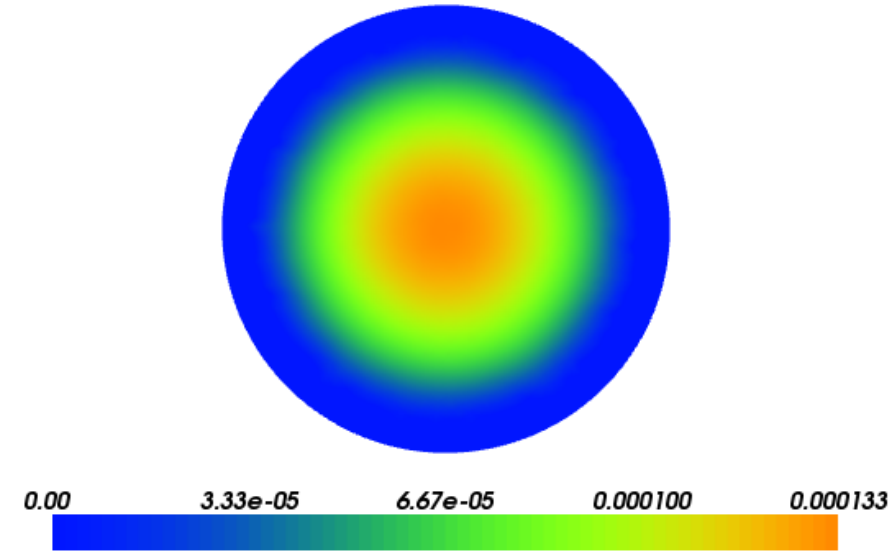

Figure 6.2.2: Displacement for an adult head and 1g linear acceleration. The acceleration force is directed straightly to the left side.

skull. This is in accordance with Wittek and Omori [95]. They showed that an attached brain/skull model, like the one we use, will cause unnaturally large strain values at the brain/skull boundary at a time period of size 20 ms. With larger time computations and when using the elastic wave equation, the large shear transports into the brain.

It turns out that when we choose μ = 40 kPa, we can reproduce nearly the same max shear strain values as Zou [101, p.123] did with μ = 50.3 kPa (see Table 6.3.1). Therefore, we choose $\mu = 40$ kPa from now on, as a *default adult shear modulus* for the brain tissue. This μ value makes it possible for us to reproduce shear strain values close to Zou's [101], and is in the range of the experimental values in Table 6.2.1.

Linear acceleration (g)	Angular acceleration rad/s^2	Our max shear strain	Zou's max shear strain
21.8	1753.2	0.059	0.079
24.4	1948.7	0.065	0.073
12.1	803.9	0.029	0.035
62.2	2745.7	0.12	0.102

Table 6.3.1: Shear strain for different tests taken from Zou [101, p.123]. Our computed values are in the third column. We choose $\mu = 40$ kPa for our simulations. Zou uses $\mu = 50.3$ kPa.

Now we go on and simulate some experiments from Hardy et al. (2007) (see the acceleration values in Table 4.6.3 and the strain results in Table 6.3.2). We have now calibrated our simulator against the Zou strain values, and want to see if we can reproduce the strain- and displacement values from Hardy et al.'s experiments. (Remember in the following that C241 is a sample with the NDT positions inside the upper part of the brain, and C408 has the NDT positions very close to the upper brain surface (see Section 2.5).) Our tests with sample C408 give, as we expect, too large shear strain values near the brain-skull boundary (see Table 6.3.2 and Figure 6.3.3). The Hardy et al. shear strains close to the brain surface (sample C408) are ten times smaller than our results (see Table 6.3.2). The explanation is again the attachment of our brain to the skull [95].

One important detail in our results is that our computed shear strain values at the position of the NDT in sample C241 are rather accurate (see Figure 6.3.5). This is an indication that our computed shear strain inside the brain may be of the right size, even if it is too large close to the brain/skull boundary. Of course one should further compare our computational results with a higher number of real

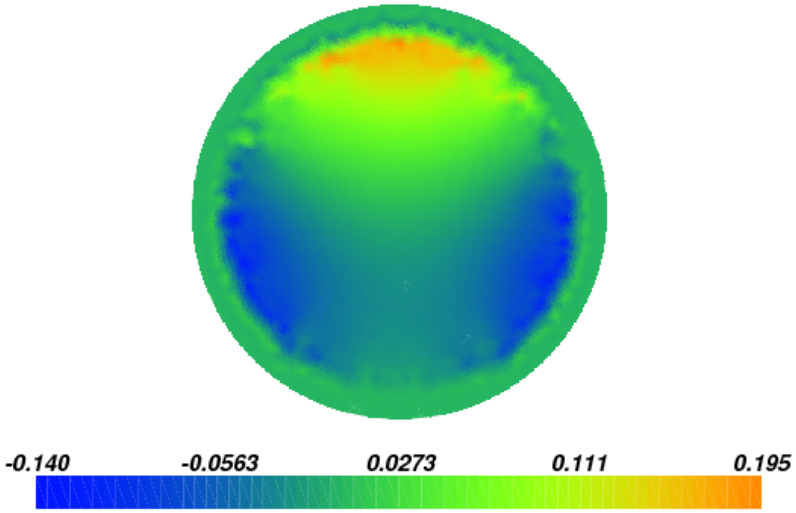

Figure 6.3.1: Typical shear strain for a combination of linear (62.2 g) and angular (2746 rad/s^2) acceleration. The corresponding acceleration field can be seen in Figure 6.3.2. The brain tissue shear modulus μ is here chosen as 25kPa.

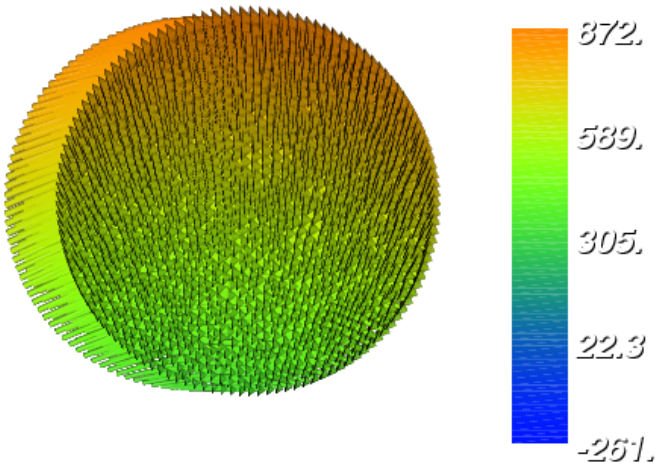

Figure 6.3.2: Acceleration for a combination of linear (62.2 g) and angular (2746 rad/s^2) acceleration. The corresponding shear strain can be seen in Figure 6.3.1.

world low-scaled acceleration tests, in order to assure that this behaviour is not a coincidence. However, the Hardy et al. (2007) tests most often have a too large acceleration magnitude for our linear elasticity computations to be accurate.

Test	C214-T3	C408-T3	C408-T4
Our max shear strain on the entire grid	0.11	0.17	0.32
Our computed max shear strain at the NDT positions	0.03	0.17	0.32
Hardy et al.'s max shear strain at the NDT positions	0.026	0.014	0.020

Table 6.3.2: Shear strain for different tests taken from Hardy et al. (2007) [38]. Our computed values are in the second and third columns. The tests with sample C241 compute the shear strain inside the upper part of the brain, and the tests with sample C408 near the surface of the cortex. The third and fourth row should coincide if our simulations are good. Our max shear strain values are computed across the entire grid. (We choose a brain tissue $\mu = 40$ kPa for our simulations.)

The *displacement* at the brain/skull boundary for acceleration test C408-T4 is also measured by Hardy et al. [38, p. 53] (see Section 2.5 for an explanation). The displacement we have computed for C408-T4 can be seen in Figure 6.3.4. The displacement should be around 10 mm at the brain/skull boundary, but we got only a displacement of 3.5 mm. The explanation for this result may again be the attachment of the brain to the skull, in addition to the other limitation of our model described above in Chapter 4. Since we do not consider time dependency in our elasticity model, this may also be a reason for the low displacement values. It is anyway an important result that we get a displacement of several millimeters. In earlier experiments by Hardy et al. (2001) [37] they found that the displacement peaks were of order ± 5 mm and that increasing acceleration did not augment the displacement at the brain/skull boundary.

Finally we validate our simulations against the experiments from Hardy et al. (2001) [37]. Now we use only the measured rotation around the y-axis, and the linear x-axis acceleration as explained in Section 4.6. Zou seems to have used the resultant acceleration for his strain computations. We see from Table 6.3.3 that the shear strain results closely match the severity of impacts. The shear strain values seem of reasonable size compared to the values measured on human volunteers in [7].

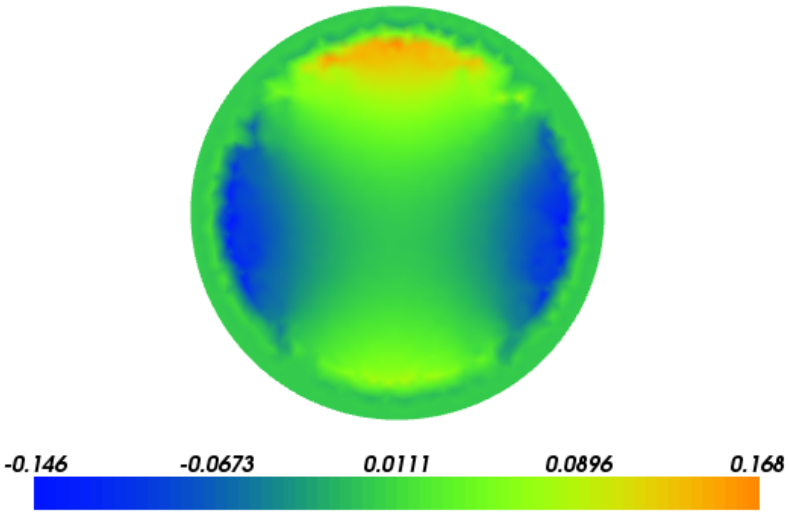

Figure 6.3.3: Our computed shear strain for Hardy et al. test C408-T3. The NDT are implantations close to the brain/skull boundary. Our computations show a shear strain 10 times larger than the real shear strain close to the brain surface.

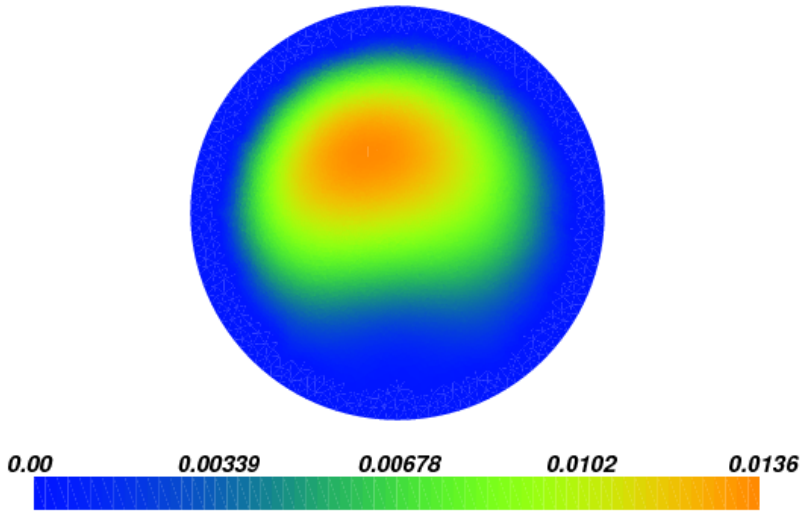

Figure 6.3.4: Our computations of the displacement of test C408-T4 from Hardy et al. (2007) [38]. The displacement at the brain/skull boundary should be 10 mm, but is much lower.

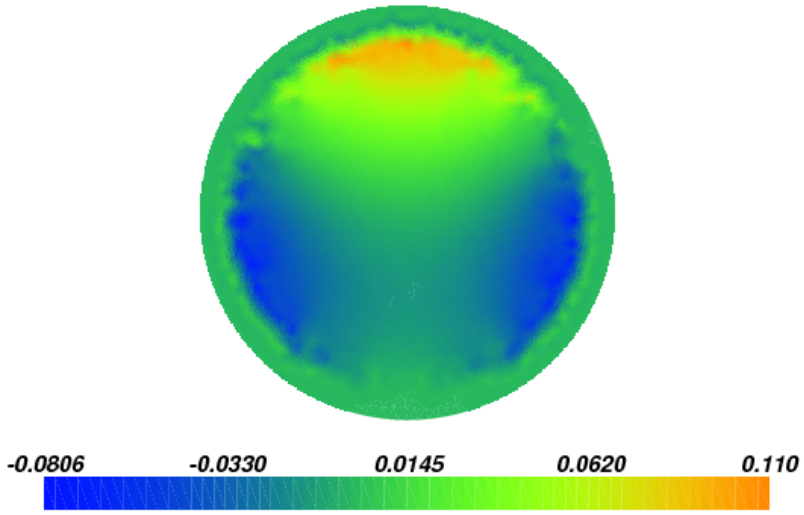

Figure 6.3.5: Our computed shear strain for Hardy et al. test C241-T3. The NDT are implantations inside the upper part of the brain. Our computed shear strain magnitude at the NDT position is close to the real values.

Test	Linear max x-acceleration (g)	Angular max y-acceleration (rad/s^2)	Max shear strain
C731-T2	20	1227	0.046
C731-T3	27	1695	0.063
C755-T2	21	1882	0.061
C755-T3	23	1995	0.065
C755-T4	27	2681	0.083
C755-T5	12	843	0.030
C381-T1	46	2592	0.10

Table 6.3.3: Our computed shear strain with use of acceleration values from the human experiments by Hardy et al. (2001) [37, table 2]. The test numbers C731-T2 (and so on) are explained in Section 2.5. They are accelerations measured in different impact tests on post-mortem human heads.

6.4 Child head simulations

When we have calibrated our FE-solver against reasonably sized strain values for adult head experiments, it is time to go on towards our main goal – simulate child head injuries, and especially our shaking acceleration peak.

First we try with different head sizes and adult elastic moduli, to assure that a smaller head gives less injuries as found in [47]. Our results show that we get less shear strain for a baby sized head at the same accelerations (see Table 6.4.1).

Linear x-acceleration (g)	Angular y-acceleration (rad/s^2)	Max shear strain (adult head size)	Max shear strain (child head size)
20	1700	0.056	0.024
30	2000	0.073	0.032
70	4000	0.16	0.07

Table 6.4.1: Different shear strain and displacement values for adult and baby head size. The brain tissue shear modulus is 40 kPa, and we only vary head size and acceleration. The adult head diameter is 19 cm, and the baby head diameter is 11 cm (the same as our crash test dummy).

One major disadvantage of scaling an adult head geometry is that it does not model the real child head injuries. Roth et al. found that one must take into account the specialties of a child head anatomy when modeling child head impacts [81] [80]. But Danelson et al. showed in [21] that it was mostly the size of the baby head that affected the computed strain – the difference in shape had little influence.

6.4.1 Simulating the shaking peak

In this section we simulate our peak acceleration field, and solve the linear elasticity equations with different scalings of the brain tissue shear modulus as described in Section 4.4.3. Thibault and Margulies [87] found that the infant shear modulus was less than the adult shear modulus (softer brain tissue), and Prange and Margulies [75] and Gefen et al. [33] meant that the infant shear modulus was larger than the adult's (stiffer brain tissue). We run our simulations with scalings according to all three experimental values (see Table 6.4.2).

In the previous simulations we always chose a Poisson's ratio ν of 0.44 to reduce the influence of locking effects. We needed a rather high ν value for simulating impacts, and for calibrating the simulations against the impact real world experiments by Hardy et al. In our shaking simulations we will use a Poisson's

ratio of 0.40. In Section 4.4.2 we explained why Guillaume et al. [36] have found that the Poisson's ratio was 0.35 during longer time durations of acceleration, but these experiments were post-mortem and therefore too low, so we choose a Poisson's ratio $\nu = 0.40$ for our shaking simulations.

Shear modulus scaling	[87] Thibault and Margulies	[75] Prange and Margulies	[33] Gefen et al.
Max shear strain	0.17	0.048	0.062
Max displacement (mm)	4.1	1.1	1.5

Table 6.4.2: Static linear elasticity solved for the shaking acceleration peaks. We use the shaking peak acceleration field from Section 4.6.1, i.e. angular acceleration = 10.5 krad/s^2 and linear acceleration = 45 g.

The results for the shaking peak simulations are shown in Table 6.4.2. The shear strains and displacements are of rather low magnitude, despite the high acceleration. The main explanation is the small sized baby head. The displacement at the brain/skull boundary is also rather low (see Figure 6.4.1). The main cause of a low displacement at the brain/skull boundary is again the attached brain [95], and the choice of the static elasticity equation instead of a time dependent one. As found in Roth et al. [79] and Morison [66], the large time duration of the shaking acceleration peak gives as large brain/skull displacement as in impacts. However, we must also remember that our acceleration peak magnitudes are in the size of impacts, while Roth et al. and Morison used much lower acceleration values for their simulations.

Couper and Albermani [20] model the Shaken Baby Syndrome with a non-linear FE-solver, and the computations are limited to 2D only, as in our case (see Section 2.3.2). Their main conclusions are that it is important to model the CSF as a fluid, and that the volume of CSF is critical for the magnitude of the relative brain/skull motion and the strain in the brain tissue. Larger subarachnoid CSF volume gives a larger brain/skull relative displacement. Their simulations again show the importance of not using an attached brain, not even with solid elements with fluid properties for the CSF layer. It is important, though, not to conclude that a naturally larger space between the brain and the skull causes broken bridging veins in shaking more easily. The bridging vein strain is dependent on both the original length of the vein, and on the displacement size.

Since our peak acceleration shear strain in Table 6.4.2 was rather low, it may be that with a better simulation of the CSF as a fluid, we will get small enough (infinitesimal) strain for the linear elasticity model to be accurate in the shaking simulations. However, we must be careful here; perhaps a longer time duration of

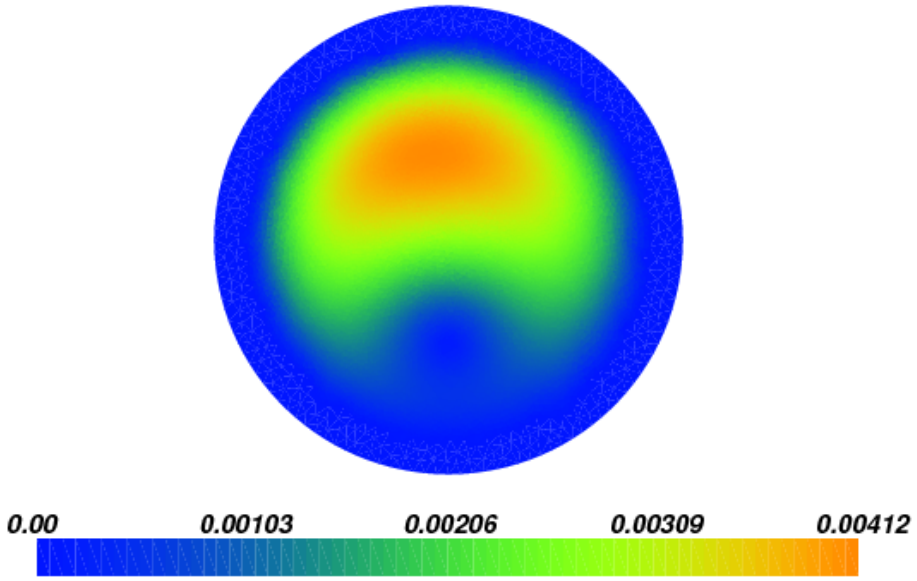

Figure 6.4.1: Displacement for our shaking peak simulations with a baby head size and shear modulus scaling from Thibault and Margulies [87].

the acceleration peak in shaking will blow up the strain to finite size, which means we have to use non-linear constitutive equations for the shaking simulations.

Couper and Albermani [20] model the myelinated white matter differently from the unmyelinated. Myelinated white matter is less stiff than the unmyelinated (see Section 4.4.3). This leads to important shear stresses at the borders of different regions. However, it is very difficult to distinguish clearly between myelinated and unmyelinated regions since no clearly visible borders actually exists. It would be interesting to further develop our model to a more complex geometry, with material differences as in Couper and Albermani.

Roth et al. [79] use the shaking acceleration from Prange et al. [73] for a non-linear FE solver, and compare shaking with impacts. The acceleration peaks have a smaller magnitude in [73] than in our max peak acceleration. Roth et al. anyway show that the relative displacement between the skull and brain in shaking can be as large as in impact, but on a different time scale. Since the bridging veins are not sensitive to strain rate [54], the longer time duration and lower strain rate do not reduce the risk for ruptured bridging veins. One limitation of Roth et al.'s simulations is that they use only a linear velocity when simulating impacts, and only an angular velocity when simulating shaking. We use a combination of linear

and angular acceleration. As seen in the real world experiments by Hardy et al. [37, 38], the angular acceleration and velocity are often large also in impacts. It depends on where the impact hits the skull. Roth et al. use a rotation axis position at cervical vertebrae 5 and 6, that is a rather long distance from the skull base, so their angular acceleration is fairly similar to a linear acceleration except for the lack of a centripetal force.

Chapter 7

Conclusion and further research

We summarize here the main results of this thesis, and give some suggestions on further research.

7.1 Summary of results

First we state again the main topics for the thesis (see Section 1.2), and present the main results at the corresponding number of the list.

1. Describe the biomechanics of a violent baby-shaking event (see Chapter 5).

   Results
   - We got a high peak vertex acceleration of 105 g, and a short peak duration. The frequency of the shaking was 5 Hz.
   - The rotation axis moved continuously during shaking. Right before the acceleration peak, the vertex accelerated in the opposite direction of the x-sensor's acceleration. The acceleration peaks in the x-direction had different shapes at different positions of the head, and the reason for this was the motion of the rotation axis into the lower part of the head.
   - We also got a high magnitude z-acceleration of between 3 and 20 g during the whole shaking cycle.
   - Our peak velocity was 8 m/s.

2. Attempt to validate the acceleration measurements of our shaking experiments (see Chapters 3 and 5).

   (a) Attempt to validate analytically and numerically the accuracy of the vertex accelerometer measurements (see Sections 3.5 and 5.5).

(b) Integrate the acceleration data for finding the motion and velocity at the vertex (see Sections 3.6 and 5.3).

Results
- We found that the rotation axis was placed 2.3 cm below the x-sensors (right below the bracket) at the peak acceleration. This is a reasonable result, since the accelerations measured at the accelerometer bracket differed a lot in the vicinity of the peaks. This is an indication of a rotation axis position close to the bracket.
- The max distance traveled by the vertex during one shaking cycle was 71 cm. This is reasonable compared to the video recordings, and is a strong indication of the accurateness of the vertex acceleration measurements.

3. Study and develop new ways of computing 3D acceleration for limited spaces, and as a consequence achieve new possibilities of validating the acceleration measurements and compute the 3D motion in world space coordinates (see Chapter 3).

   (a) Study the rigid motion formula written as a matrix exponential, and see if it leads to a simpler numerical method for computing 3D acceleration, properly suited for the nature of our problem (see Section 3.3).

   (b) Develop a new accelerometer configuration suited for 3D computations and limited spaces, and study it analytically and numerically (see Sections 3.4 and 5.7).

   Results
   - When we studied the rigid motion with a rotation matrix written as a matrix exponential, we achieved the same formula as the geometrically derived equation (2.4.1) used for several decades in acceleration computations.
   - We constructed a new theoretical accelerometer configuration and a corresponding computer program for 3D acceleration computations, suited for small spaces. Computations with the new accelerometer positions showed an acceptably sized error.

4. Study how to calculate the displacement and strain of the brain, and the brain/skull/CSF system of interaction (see Chapters 4 and 6).

   (a) Review literature to determine appropriate material parameters (elastic moduli), and identify their range of variation. Is the best choice other elastic moduli values than at impacts? (See Section 4.4.)

(b) Choose a simplified elasticity FE model that can be carried out within the time limitations for this Master's thesis (see Chapter 4).

(c) Calibrate the elasticity model against real world experiments. Solve the elasticity equations with the acceleration field of our maximum shaking acceleration, and find the displacement at the brain/skull boundary (see Chapter 6).

Results

- We found that linear elasticity computations with our simple 2D head slice, could reproduce shear strains of the same size as real world experimental values. The model also behaved reasonably well when altering the elastic moduli, the acceleration field, and the head size. However, it showed too large concentrations of the shear strains at the brain/skull boundary compared to human experiments. The reason was probably the attachment of the brain to the skull.
- Our model showed a too small displacement at the brain/skull boundary. The main causes are probably that we did not consider the time duration of the acceleration, and we modeled an attached brain.
- The simulation of the acceleration peak in shaking gave a small displacement at the brain/skull boundary, and a rather low shear strain. The limitations of our model make these results unreliable.

7.2 Future developments of our research

There are of course several limitations of the current study and future research is needed. Our current plans for the future is:

We would like to

- construct new experiments with the crash test dummy, for perhaps excluding a shaking-impact (see Section 5.2).
- implement the numerical method for finding the rotation axis position (see Section 3.5). Afterwards we would like to compare this position with the analytical rotation axis position from Section 5.5, and thus validate completely the vertex acceleration measurements.
- compute the analytical position of the rotation axis at all acceleration peaks, and check if the different peaks have a similar shape.

- develop further the FE head model to a more anatomically correct geometry, with a better simulation of the CSF and the brain tissue. We should compare the model with variations among patients, and attempt to decide the uncertainty of the model.

- use a time dependent elasticity equation for our FE simulations, where we insert the 3D rigid motion as boundary conditions. This will incorporate the longer time durations of the acceleration peaks in shaking. The model could also be expanded with inserting the bridging veins and their elastic properties into the model. It may be natural to use some of the methods by Zoghi-Moghadam and Sadegh [100, 99], where they first model the displacement at the brain/skull boundary in a global head model, and later model the neighborhood of the bridging veins locally, for careful computations of the bridging veins strains.

7.3 Further research

Further research is also needed in the following areas.

1. New biomechanical shaking experiments are needed with more realistic anthropomorphic test dummies. One should compute the 3D acceleration, and find the position of the rotation axis.

2. Better models and numerical methods are needed for simulating infant head injuries.

3. One needs to determine the uncertainty properties of the different parameters and measurements involved in both the acceleration investigations and the FE simulations.

4. It would also be interesting with more research on the elastic properties of the body tissues involved in the SBS injuries.

    - The infant brain tissue properties have not been established completely yet, and diverge a lot between different studies (see Section 4.4.3). Further research is needed here.
    - Further experiments on Poisson's ratio of the brain, should focus on finding ν as a function of the time duration of the applied force, and also as a function of the direction of the force and the size of the sample (see Section 4.4.2).

- It would also be interesting to expose a whole head to centrifugal forces, and study the blood streams and the blood pressure at the bridging veins, in addition to the pressure gradient in the CSF surrounding the bridging veins. Further research is also needed for finding out if the bridging veins more easily get ruptured when exposed to a higher interior pressure than under normal conditions (see Section 4.4.2).

Appendix A

Appendix - computer programs

A.1 Elasticity computations

First, the program used in the elasticity computations in Section 6, *LinearElasticitySimulator.py*.

```
"""This program solves the equations of static linear elasticity
for an adult or child 2D head circle, with rotational and translational
accelerations.

It also computes the shear strain, and plots the displacement and the shear strai

It tests different values of lame parameters, and all other values can be
varied from the command line
or manually.

Mu = G trough the whole program.

It works with dolfin version 0.9.4

"""

from dolfin import *
import numpy
import sys, getopt, string
from LameParameterFunctions import *

# Set default values of acceleration, adult or child cases
# Default acceleration suites our max for baby dummy shaken (baby head size)
```

```
linear_acc = 45 # linear acceleration in g
alpha = 10500 # angular acceleration in radians/s^2
adult = True # Baby or adult
baby_values = 0 # Choice of scaling value. May take 0, 1, 2, or 3 as input.
head_diam = 0
mu_ = 40000 # Default calibrated with Zou shear strain values
nu_ = 0.44 # Compressible brain default.
userDefinedMu = False # Override the E/nu test values

# Import optional command line arguments
options, args = getopt.getopt(sys.argv[1:], '',
                              ['l=', 'linear_acc=', 'a=', 'angular_acc=',
      'alpha=', 'adult=', 'b=', 'baby_values=',
      'mu=', 'nu=', 'userDefinedMu=', 'u='])
for option, value in options:
    if option in ('--l', '--linear_acc'):
        linear_acc = float(value)
    elif option in ('--a', '--angular_acc', '--alpha'):
        alpha = float(value)
    elif option in ('--adult'):
        adult = eval(value)
    elif option in ('--b', '--baby_values'):
        baby_values = int(value)
    elif option in ('--mu'):
        mu_ = float(value)
        userDefinedMu = True
    elif option in ('--nu'):
        nu_ = float(value)
    elif option in ('--userDefinedMu', '--u'):
        userDefinedMu = eval(value)
        if userDefinedMu == True:
            print '\nUses default mu value or command line input mu value\n'

if (adult==True):
    head_diam = 0.19 # Adult head
else:
    head_diam = 0.11 # Baby size of head



print '\nWe use a head diameter of %.2g m.\n' % head_diam
```

```
# Load mesh
mesh = Mesh("circle.xml.gz")

# Finds radius of mesh
def find_radius(mesh):
    coord =  mesh.coordinates()
    temp_max = []
    temp_min = []
    i = 0
    while i < len(coord):
        ma = max(coord[i])
        mi = min(coord[i])
        temp_max.append(ma)
        temp_min.append(mi)
        i += 1
    print 'i: ', i
    print 'Max coordinates of mesh: ', max(temp_max)
    print 'Min coordinates of mesh: ', min(temp_min)

# Radius of UnitCircle; == 0.5 m (Needs to scale for correct dimensions in meter)
# find_radius(mesh)

# Scale mesh according to size of head
scaling_factor = head_diam/2/0.5
mesh.coordinates()[:] = mesh.coordinates()*scaling_factor
#find_radius(mesh)
#sys.exit(1)


# Define function space
V = VectorFunctionSpace(mesh, "CG", 1)
W = FunctionSpace(mesh, "CG", 1)


# Sub domain for brain tissue
brain_diam2 = (head_diam/2)**2 - ((head_diam/2)**2)/5 # Brain radius squared
class InnerStuff(SubDomain):
    def inside(self, x, on_boundary):
        if x[0]**2 + x[1]**2  < brain_diam2:
            return True
```

```
        return False


# Sub domain for skull boundary conditions
class Skull(SubDomain):
    def inside(self, x, on_boundary):
         return on_boundary


# Create subdomains for brain/skull
sub_domains = MeshFunction("uint", mesh, 0)
innerstuff = InnerStuff()
innerstuff.mark(sub_domains, 1)

#plot(sub_domains)
#interactive()
#sys.exit(1)



# Expression for rotational and linear acceleration force. 2D vector field.
class Acceleration(Expression):
    def eval(self, value, x):
        value[0] = alpha*x[1] + linear_acc*9.82
        value[1] = -alpha*x[0]


# Define density
ro_brain = 1.05*1000 #
# From http://en.allexperts.com/q/Molecular-Biology-1353/density-brain.htm
ro_skull = 1.3*1000 # HoThesis
ro_values = numpy.zeros(len(sub_domains.values()))
for i in range(len(sub_domains.values())):
    val = sub_domains.values()[i]
    if val == 1: # Brain tissue
        ro_values[i] = ro_brain
    else: # Skull bone
        ro_values[i] = ro_skull
ro = Function(W)
ro.vector()[:] = ro_values
```

```
# Read lame parameter values from file
def read():
    filename = "BrainLameParameters.txt"
    file = open(filename)
    lines = file.readlines()
    list_of_numbers = []
    for line in lines[3:]:
        # Applying the function float to each member of the sequence
        numbers = map(float, string.split(line))
        list_of_numbers.append(numbers)
    file.close()
    return list_of_numbers



# Define Elastic moduli

mu_skull = 0
lambda_skull = 0
mu_brain = 0
lambda_brain = 0

# Use the same skull parameters fo child and adult since
# we don't have impact to skull.
# Skull Young's modulus (E) = 15 GPa, HoThesis
e_skull = 15e9
nu_skull = 0.22 # From Roth et al., HoThesis
k_skull = e_skull/(3*(1 - 2*nu_skull))       # K = E/(3*(1 - 2*nu)), 8.93 GPa
# Convertion formulas from Wikipedia
mu_skull = 3*k_skull*((1 - 2*nu_skull)/(2*(1 + nu_skull)))
lambda_skull = 3*k_skull*nu_skull/(1 + nu_skull)


def set_lameParameters(E, nu):
    '''Function for setting the lame parameters lambda and mu (also named G).
    '''
    if (userDefinedMu == False): # use loop over E/nu experimental values
        print '\nE = %.4g and nu = %.4g' % (E, nu)
```

```
mu_brain = E_nu_to_G(E, nu) # Compute correct adult mu
               # with function from LameParameterFunction.py
nu_brain = nu_ # Need a compressible brain for avoiding locking

# Override mu values with command line argument or default value
if (userDefinedMu == True):
    mu_brain = mu_ # Command line argument. Default 40 kPa
    print '\nUser defined shear modulus mu'



if (adult==True):
    print 'Adult shear modulus chosen\n'

elif(adult==False): # Scale adult values according to child values
    print 'Baby values chosen'
    if (baby_values==0): #Adult values anyway
        print 'Baby sized head, but adult shear modulus\n'
    elif (baby_values==1): # Thibault/Margulies scaling from 1998
        print 'Thibault/Margulies scaling from 1998\n'
        mu_brain = 0.667*mu_brain # Scale adult mu_brain

    elif (baby_values==2): # Prange/Margulies scaling from 2002
        print 'Prange/Margulies scaling from 2002\n'
        mu_brain = 2.434*mu_brain

    elif (baby_values==3): # Gefen et al. scaling from 2003
        print 'Gefen et al. scaling from 2003\n'
        mu_brain = 1.9*mu_brain

lambda_brain = G_nu_to_lamda(mu_brain, nu_brain)


print "Mu brain: ", mu_brain
print "Lambda brain: ", lambda_brain

# Test if Poissons ratio is small enough for a compressible brain
nu_test = G_lamda_to_nu(mu_brain, lambda_brain)
if (nu_test < 0.46):
    print '\nCompressible brain succeded. nu = %.3g\n' % nu_test
else:
```

```
            print '\nCompressible brain failed. nu = %.3g\n' % nu_test
            sys.exit(1)


    mu_values = numpy.zeros(len(sub_domains.values()))
    lambda_values = numpy.zeros(len(sub_domains.values()))
    for i in range(len(sub_domains.values())):
        val = sub_domains.values()[i]
        if val == 1: # Brain tissue
            mu_values[i] = mu_brain
            lambda_values[i] = lambda_brain
        else: # Skull bone
            mu_values[i] = mu_skull
            lambda_values[i] = lambda_skull

    # This works well with dolfin version 0.9.4
    lamda = Function(W)
    lamda.vector()[:] = lambda_values
    mu = Function(W)
    mu.vector()[:] = mu_values

    # print 'Plots lambda'
    # plot(lamda, warpscalar=False) # projects in 2D
    # interactive()
    # sys.exit(1)
    # plot(mu)
    # interactive()
    # print 'Plots density ro: '
    # plot(ro)
    # interactive()
    # sys.exit(1)
    return lamda, mu, mu_brain # Return Functions and scalar value of mu_brain


# Stress tensor
def sigma(v):
    return 2.0*mult(mu,epsilon(v))
+ mult(lamda,mult(trace(epsilon(v)), Identity(2)))

# Eulerian strain tensor
def epsilon(v):
```

```
    return 0.5*(grad(v) + transpose(grad(v)))


# Define acceleration field
f = Acceleration(V=V) # Acceleration force field

# Plot acceleration
#print '\nPlots acceleration field:\n'
#f_int = interpolate(f,V)
#plot(f_int)
#interactive()
#print 'Max acceleration: ' , max(f_int.vector())
print '\nAngular acceleration= %.04g radians/s^2, and \
 \nlinear acceleration= %.04g g\n' % (alpha, linear_acc)
#sys.exit(1)

# Homogenous boundary condition
boundary_function = Expression('0.0', V=W)
bc = DirichletBC(V, boundary_function, Skull())

# Read from elastic moduli file.
parameters = read()

# Initialise list for storing max shear strain
MaxShearStrain = []

# File for saving max shear strain
shear_file = open('MaxShearStrain.dat','w')
shear_file.write('# Linear acc %.3g g and angular acc %.3g r/s^2\n' %
  (linear_acc, alpha))
shear_file.write('# Head diameter = %.2g, parameter scaling rule is %g \n\n' %
  (head_diam, baby_values))
shear_file.write('# Data below are: \tmu_brain \tE \tnu \tmax shear strain \n\n')



# Here is a loop over all different values of E and nu from file
for i in range(len(parameters)):
    E = parameters[i][0]
    nu = parameters[i][1]
```

```
# Contruct lamda and mu Functions
lamda, mu, mu_scalar = set_lameParameters(E, nu)
# Uses only E, nu if UserDefinedMu == False

# Define variational form of static linear elasticity
# From Shaw page 20, Eriksson page 523, Brenner page 282.
v = TestFunction(V)
u = TrialFunction(V) # Relative displacement as vector field in 2D

a = 2*mu*inner(epsilon(v), epsilon(u))*dx + lamda*div(u)*div(v)*dx
L = -ro*dot(v, f)*dx

problem = VariationalProblem(a, L, bc)
sol = problem.solve()

# Plot relative displacement
print '\nPlots relative displacement field:\n'
plot(sol, mode="displacement")
interactive()

# Compute the strain tensor
strain = epsilon(sol)
shear_object = strain[0][1]

# Project shear strain onto W
# Need to define variational form for evaluating trace of strain tensor
s = TrialFunction(W)
v = TestFunction(W)
a_strain = s*v*dx
L_strain = shear_object*v*dx
probl_strain = VariationalProblem(a_strain, L_strain)
shear_strain = probl_strain.solve()

print '\nPlots shear strain\n'
plot(shear_strain, warpscalar = False)
interactive()


# Find max shear strain
ma = max(shear_strain.vector())
mi = min(shear_strain.vector())
```

```
    mi = abs(mi)
    m = max(ma, mi)
    print '\nMax absolute shear strain = %.4g\n' % m

    # If user defined mu or nu, leave loop
    if (userDefinedMu == True):
        shear_file.close()
        sys.exit(1)


    MaxShearStrain.append(m)

    print '\nMax shear strain = %.4g for E = %.4g and nu = %.4g\n' % (m, E, nu)

    # Save Max shear strain to file to compare with
    # Hardy et al.'s experiments for adults
    shear_file.write('%10.4g %10.4g %12.4g %10.6g \n' % (mu_scalar, E, nu, m))


# Print max shear strain
print 'Max shear strain list: \n', MaxShearStrain
shear_file.close()
```

A.2 Integration programs

Here follows the program from Section 3.6 used in Section 5.3. It contains integration tests, and computes the velocity and shaking length between two positive subsequent acceleration peaks.

```
import string
from numpy import *
from pylab import *
import sys
from math import pi

'''
Integrate the vertex acceleration two times.
Check the maximum velocity.
'''
```

```
def read(start = 0, stop = 0):
    '''
    Read vertex acceleration values. Return list of values.
    '''
    ofile = open('shake_torleiv_test2.lvm')
    lines = ofile.readlines()
    if start==0 and stop==0:
        start = 0
        stop = len(lines) # Stop index of integration + 1

    Vertex = []
    Time = []
    i = 1
    startline = map(float, string.split(lines[start]))
    time = startline[0]
    print 'Time start: ', time
    for line in lines[start:stop]:
        numbers = map(float, string.split(line))
# nine numbers, time, x1, x2, y1, y2, z1, z2, vertex, abs
        vertex = numbers[7]
        time += 0.0002
        Time.append(time)
        Vertex.append(vertex)
    return Time, Vertex


def integrate(liste, h=0.0002):
    '''
    Integrate with composite trapezoidal rule and default h=0.0002.
    Works both with a list and a python array.
    '''
    print 'Start value in integration: ' , liste[0]
    print 'Stop value: ' , liste[-1]
    #print len(list)
    print 'h = ', h
    vertex = 0
    integratedList = [0.0]
    for i in range(1, len(liste)):
        vertex += (liste[i-1] + liste[i])*(h/2)
        integratedList.append(vertex)
    return integratedList
```

```
def test1():
    '''Tests the integration with cos(x)'''
    a = linspace(0, 2*pi, 2000)
    a = cos(a)
    print 'cos(0): ', a[0]
    print 'cos(2pi): ', a[-1]
    print 'Plots cos(x)'
    plot(a)
    integration_interval = 2*pi/1999
    print 'Integrates cos(x) and plots the result'
    b = integrate(a, h=integration_interval)
    print 'max of sin(x) integrated: ', max(b)
    plot(b)
    show()
    sys.exit(1)

def test2():
    '''Tests the integration with x**2'''
    time = linspace(0, 2 , 9999)
    a = time**2
    print 'Plots x**2'
    #plot(a) # Looks great
    #show()
    integration_interval = 0.0002
    print 'Integrates x**2 and plots the result'
    b = integrate(a, h=integration_interval)
    # Test with x**3/3
    c = (time**3)/3
    plot(b, label='Integrated array')
    plot(c, label='Analytical integration')
    legend()
    show()
    sys.exit(1)

# Tests the integration function
# test1() # Works perfect with cos(x)
# test2() # Works perfect with x**2

Time, VertexList = read()
```

```
# We see that all data is from shaking
# plot(VertexList)
# show()
# sys.exit(1)

# Finds maximum acceleration and one integration cycle
'''
m = max(VertexList)
print 'Max: ', m # 105.13g
index = VertexList.index(m)
print 'Time step: ', index # 37254, line 37255; Time step 37255
sys.exit(1)
'''
# Next acceleration maximum are at line 38218, i.e. index 38217


# Read acceleration data from one cycle of shaking
Time, VertexAcc = read(start = 37254, stop = 38218)
print 'Start acceleration value: ', VertexAcc[0]
print 'Stop acceleration value: ', VertexAcc[-1]

# Subtract one g (of course not the true direction of gravity)
a = array(VertexAcc)
# a = a-1

# Multiply with 9.82 m/(s*s) for values in SI
a = a*9.82
print 'Max acceleration in SI units: ', a[0]
SI_VertexAcc = a.tolist()



# Integrate once for velocity, initial conditions zero at max acceleration
print 'Integrates acceleration'
VertexVel = integrate(SI_VertexAcc)

# Find max velocity, 8.0 m/s in this cycle
maxVel = max(VertexVel)
print 'Max velocity: ', maxVel

# Find if zero acceleration happens at maks velocity - IT DOES
```

```
'''
ind = VertexVel.index(maxVel)
print 'Index max velocity: ', ind # 363
zero_acc = SI_VertexAcc[ind]
print 'Acceleration at max velocity in m/s: ', zero_acc
indexzero_acc = SI_VertexAcc.index(zero_acc)
print 'Index of this acceleration: ', indexzero_acc # The same
# Wants to find the value in g from original file
a0_in_g = VertexList[37617] # Index 363 + 37254
a0computed = zero_acc/9.82
print 'Acceleration in g at max velocity from original data, %g, \
      and computed from m/s list, %g.' \
      % (a0_in_g, a0computed)
IndexList = []
i = 37600
while i < 37630:
    IndexList.append(i)
    i+=1
plot(IndexList, VertexList[37600:37630], \
      label='Acceleration around max velocity') # Zero at index 37617
legend()
show()
'''

# Integrate once more for the curve length of vertex motion,
# Initial position zero
VertexMotion = integrate(VertexVel)

# Plot acceleration and velocity
#plot(SI_VertexAcc)
#show()

plot(Time, VertexVel, label='Velocity')
legend()
show()
# sys.exit(1)

# Plot velocity and position
#plot(VertexVel, label='Velocity')
#hold(True)
#plot(VertexMotion, label='Motion')
```

```
#legend()
#show()
```

Here follows the program *DuhaimeVelocitySimulated.py* from Sections 3.6 and 5.3.

```
'''A test that simulates a shaking cycle of Duhaime, and computes the velocity'''
import string
from numpy import *
from pylab import *
import sys
from math import pi

def integrate(liste, start=0, stop=375, h=0.0002):
    '''
    Integrate with composite trapezoidal rule and default h=0.0002.
    '''
    print 'Start value in integration: ' , liste[start]
    print 'Stop value: ' , liste[stop]
    #print len(list)
    print 'h = ', h
    vertex = 0
    integratedList = [0.0]
    for i in range(start+1, stop):
        vertex += (liste[i-1] + liste[i])*(h/2)
        integratedList.append(vertex)
    return integratedList

# Divide cos period with length of cycle 0.3s
# from diagram page 413 Duhaime et al.
# A longer period than if using "time interval" 106.6 s from page 413.
a = 2*pi/.3

# Construct time array with time step 0.0002
time = linspace(0,1,5001)
# Construct Duhaime mean shaking cycle
# Amplitude 91.23 m/s^2 = mean peak acc from table 6 on page 412 (9.29 G),
# times 9.82 m/s^2
duhaime = 91.23*cos(a*time)
# Less pointed peak than Duhaime, i. e. gives larger velocity
```

```
#plot(time, duhaime, label='Duhaime mean shaking acceleration in m/s**2')
# Looks fine, 3.3 cycles a second
#legend()
#show()
#sys.exit(1)

# Time to first zero acceleration is 0.3/4 = 0.075,  index 0.075/0.0002 = 375
# print duhaime[375] # correct = zero

# Find max velocity:
# Integrate from start to zero acceleration
duh_vel = integrate(duhaime)
m = max(duh_vel) # 4.36 m/s
print 'Max Duhaime velocity of shaking cycle from diagram page 413: ', m
plot(duh_vel)
show()
```

A.3 Conversion formulas

Parts of *LameParameters.py* used in Section 4.5 are printed below:

```
# Methods using conversion formulas from wikipedia (G = mu)
def E_nu_to_G(E, nu):
    mu = E/(2*(1 + nu))
    return mu

def G_nu_to_lamda(G, nu):
    lamda = (2*G*nu)/(1 - 2*nu)
    return lamda

def E_nu_to_lamda(E, nu):
    lamda = (E*nu)/((1 + nu)*(1 - 2*nu))
    return lamda

def K_G_to_nu(K, G):
    nu = (3*K - 2*G)/(2*(3*K + G))
    return nu

def K_nu_to_G(K, nu):
    G = 3*K*((1 - 2*nu)/(2*(1 + nu)))
```

```
    return G

def K_nu_to_lamda(K, nu):
    lamda = 3*K*nu/(1 + nu)
    return lamda

def E_nu_to_K(E, nu):
    K = E/(3*(1-2*nu))
    return K

def K_nu_to_E(K, nu):
    E = 3*K*(1 - 2*nu)
    return E

def E_K_to_G(E, K):
    G = 3*K*E/(9*K - E)
    return G


# Test for checking behaviour of original formula first used for computing mu:
# (K, nu to mu) with nearly incompressible $\nu$ valus.
def K_nu_test():
    K = 2.3e9
    nu = 0.4
    print 'Here K is constantly 2.3e9'
    print 'Nu = ', nu
    G = K_nu_to_G(K, nu)
    print 'Mu = %.6g' % G + '\n'

    for i in range(1,5):
       nu = nu + eval('9e%i' % (-1 - i))
       print 'Nu = ', nu
       G = K_nu_to_G(K, nu)
       print 'Mu = %.6g' % G + '\n'

K_nu_test()

# Test for checking the behaviour of ((E, nu) to G) in both variables
def E_nu_test():
    E = 150e3
    nu = 0.4
```

```
print 'Here E is constantly 150e3'
print 'Nu = ', nu
G = E_nu_to_G(E, nu)
print 'Mu = %.6g' % G + '\n'

for i in range(1,5):
   nu = nu + eval('9e%i' % (-1 - i))
   print 'Nu = ', nu
   G = E_nu_to_G(E, nu)
   print 'Mu = %.6g' % G + '\n'

nu = 0.49
E = 10e3
print 'Here nu is constantly 0.49'
print 'E = ', E
G = E_nu_to_G(E, nu)
print 'Mu = %.6g' % G + '\n'

for i in range(5):
    E += 50e3
    print 'E = ', E
    G = E_nu_to_G(E, nu)
    print 'Mu = %.6g' % G + '\n'
```

A.4 New accelerometer configuration

Here follows the program where we use our new accelerometer coordinates from Section 3.4 for computing a "vertex" acceleration simulation showed in Figure 5.7.1.

```
import string
import numpy as n
from swiginac import numeric, symbol, matrix, lsolve

'''
Computes the acceleration values at the new coordinate's vertex, with an array
of six accelerometer measurements. Saves and plots the results.
Compares with measured values.

The change-of-basis-matrix turns the x-y-plane 45 degrees around y-axis.
```

```
The y-axis is along the x-sensors.

The solution only oscillates mildly, with an error of 1 g.
'''


def read():
    filename = "shake_torleiv_test2.dat"
    file = open(filename)
    lines = file.readlines()
    list_of_numbers = []
    for line in lines[1:-2]:
        #Applying the function float to each member of the sequence
        numbers = map(float, string.split(line))
        list_of_numbers.append(numbers)
    return list_of_numbers


def compute(list_of_numbers):
#   Assume a rigid transformation on the form
#   x =  c1 l  + c2 m + b1
#   y = -c1 k  + c3 m + b2
#   z = -c2 k  - c3 l + b3

    C = matrix(3,3)
    c1,c2,c3 = symbol("c1"), symbol("c2"), symbol("c3")
    C[0,1] =  c1; C[0,2] =  c2
    C[1,0] = -c1; C[1,2] =  c3
    C[2,0] = -c2; C[2,1] = -c3

    b1,b2,b3 = symbol("b1"), symbol("b2"), symbol("b3")
    b = matrix(3,1,[b1,b2,b3])

    k,l,m = symbol("k"), symbol("l"), symbol("m")
    xi = matrix(3,1,[k,l,m])

    x = C*xi + b

    #Coordinates in standard basis E in meter
    #Origo is placed at x-y-sensors center
    #Axes are parallell with sensor cube
```

```
#Accurate coordinates for the sensor placement
#x and y sensors is in x-y-plane

#x-sensors on y-axis, and ysensors on x axis

x1coorE = n.matrix([[-.0024],[-0.0198],[0]])
x2coorE = n.matrix([[.0024],[0.0198],[0]])
y1coorE = n.matrix([[0.0198],[-.0024],[0]])
y2coorE = n.matrix([[-0.0198],[.0024],[0]])
z1coorE = n.matrix([[-0.00234],[-0.00455],[.01275]])
z2coorE = n.matrix([[0.00044],[-0.00025],[-.02515]])
vertexcoorE = n.matrix([[0],[0],[.0579]])

#coordinates in new basis
x1coorB = EtoB(x1coorE)
x2coorB = EtoB(x2coorE)
y1coorB = EtoB(y1coorE)
y2coorB = EtoB(y2coorE)
z1coorB = EtoB(z1coorE)
z2coorB = EtoB(z2coorE)
vertexcoorB = EtoB(vertexcoorE)

X1 = []
X2 = []
resultVertex = []
tidArray = []
vertexMeasured = []

t=0
oMfile = open('VertexMeasuredCompute8.dat','w')
oCfile = open('VertexComputedCompute8.dat','w')
oMfile.write('# Time versus measured values at the vertex\n')
oCfile.write('# Time versus computed values at the vertex\n')

for li in list_of_numbers:
    (tid, x1,x2,y1,y2,z1,z2, vertex, absolutt) = li
    tid = 0.0002*t
    tidArray.append(tid) #The real timesteps
    vertexMeasured.append(vertex)
    X1.append(x1)
    X2.append(x2)
```

```
        t+=1
        eq1 = (x[0] == x1).subs ([k == x1coorB[0,0], l == x1coorB[1,0], \
  m == x1coorB[2,0]])
        eq2 = (x[0] == x2).subs ([k == x2coorB[0,0], l == x2coorB[1,0], \
  m ==  x2coorB[2,0]])
        eq3 = (x[1] == y1).subs ([k == y1coorB[0,0], l == y1coorB[1,0], \
  m ==  y1coorB[2,0]])
        eq4 = (x[1] == y2).subs ([k == y2coorB[0,0], l == y2coorB[1,0], \
  m ==  y2coorB[2,0]])
        eq5 = (x[2] == z1).subs ([k == z1coorB[0,0], l == z1coorB[1,0], \
  m ==  z1coorB[2,0]])
        eq6 = (x[2] == z2).subs ([k == z2coorB[0,0], l == z2coorB[1,0], \
  m ==  z2coorB[2,0]])

        equations = [eq1, eq2, eq3, eq4, eq5, eq6]
        variables = [c1,c2,c3,b1,b2,b3]
        # Find all variables and save in symbolic list
        sol = lsolve(equations, variables)
        result = x.subs([k == vertexcoorB[0,0], l == vertexcoorB[1,0], \
  m == vertexcoorB[2,0]]).subs(sol).evalf()
        resultVertex.append(result[0,0])

        #Write to file
        oMfile.write('%g\t%.7e\n' % (tid, vertex))
        oCfile.write('%g\t%.7e\n' % (tid, result[0,0]))


    oMfile.close()
    oCfile.close()

    #Plot the solution
    from pylab import *
    plot(tidArray[37500:40000], vertexMeasured[37500:40000], 'b-', \
  label='Measured vertex values, blue')
    hold(True)
#    plot(tidArray[37500:40000], X2[37500:40000], 'r-.', \
  label='Measured x2 values, red')
 #   hold(True)
    plot(tidArray[37500:40000], resultVertex[37500:40000], 'g:', \
  label='Computed vertex-values, green')
#    legend()
```

```
    show()

def EtoB(x):#Turns the x-yplane 45 degrees around y-axis
    BtE = n.matrix([[1./n.sqrt(2.), 0., 1./n.sqrt(2.)],\
                    [0., 1., 0.],\
                    [-1./n.sqrt(2.), 0., 1/n.sqrt(2.)]])#orthonormal basis
    EtB = BtE.I #inverse
    return EtB*x #new coordinates


if __name__ == '__main__':
    list_of_numbers = read()
    compute(list_of_numbers)
```

A.4.1 Condition number

Here follows the program for computing the condition number (*ConditionNumber.py*) in Section 5.7.

```
'''Compute the condition number of the matrix Coord'''
import sys
import numpy as n

# Coordinates for accelerometer positions
# Accurate coordinates with changed x-y-axes
x1 = n.matrix([[-.0024],[-0.0198],[0]])
x2 = n.matrix([[.0024],[0.0198],[0]])
y1 = n.matrix([[0.0198],[-.0024],[0]])
y2 = n.matrix([[-0.0198],[.0024],[0]])
z1 = n.matrix([[-0.00234],[-0.00455],[.01275]])
z2 = n.matrix([[0.00044],[-0.00025],[-.02515]])


#Turns the x-yplane 45 degrees around y-axis
# New basis B
def EtoB(x):
    BtE = n.matrix([[1./n.sqrt(2.), 0., 1./n.sqrt(2.)],\
                    [0., 1., 0.],\
                    [-1./n.sqrt(2.), 0., 1/n.sqrt(2.)]])#orthonormal basis
    EtB = BtE.I #inverse
    return EtB*x #new coordinates
```

```
# Convert coordinates to the new accelerometer configuration
x1coorB = EtoB(x1)
x2coorB = EtoB(x2)
y1coorB = EtoB(y1)
y2coorB = EtoB(y2)
z1coorB = EtoB(z1)
z2coorB = EtoB(z2)


# Extract numbers from coordinate vectors
y_x1 = float(x1coorB[1][0])
z_x1 = float(x1coorB[2][0])
y_x2 = float(x2coorB[1][0])
z_x2 = float(x2coorB[2][0])
x_y1 = float(y1coorB[0][0])
z_y1 = float(y1coorB[2][0])
x_y2 = float(y2coorB[0][0])
z_y2 = float(y2coorB[2][0])
x_z1 = float(z1coorB[0][0])
y_z1 = float(z1coorB[1][0])
x_z2 = float(z2coorB[0][0])
y_z2 = float(z2coorB[1][0])

# Define matrices and unnessecary test vector for
# the lstsq range command
Coord = n.matrix([[y_x1, z_x1, 0, 1, 0, 0],\
                  [y_x2, z_x2, 0, 1, 0, 0],\
                  [-x_y1, 0, z_y1, 0, 1, 0],\
                  [-x_y2, 0, z_y2, 0, 1, 0],\
                  [0, -x_z1, -y_z1, 0, 0, 1],\
                  [0, -x_z2, -y_z2, 0, 0, 1]])
#print Coord

b = n.matrix([[1],[1],[1],[1],[1],[1]])


# Compute the singular value decomposition of
# the matrix Coord
from numpy.dual import svd
U,s,Vh = svd(Coord)
```

```
# Need to disregard too small values (Lay page 469)
S = list(s)
i = True
while i:
    print 'S = ', S

    if S[-1] < 1e-12:
        print 'last value of S = ',S[-1]
        del S[-1]
    else:
        i = False

minS = S[-1]
cond1 = (max(s)/minS)
print 'Condition number of matrix', cond1
print 'max singular value= ', max(s)
print 'first singular value= ', s[0]
print 'min singular value= ', minS

from numpy.dual import lstsq
(xvector, resids, rank, sing) = lstsq (Coord, b)
print '\nRank of matrix =  ', rank, '\n'
print sing
# Gives the same singular value decomposition
# as the svd command
```

Bibliography

[1] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*. Springer Verlag, 1978.

[2] I Babuška and M Suri. Locking effects in the finite element approximation of elasticity problems. *Numer Math*, 62:439–463, 1992.

[3] I Babuška and M Suri. On locking and robustness in the finite element method. *Siam J Numer Anal*, 29(5):1261–1293, 1992.

[4] Andrew Baker. *Matrix groups, An introduction to Lie group theory*. Springer Verlag, 3rd printing edition, 2006.

[5] F Bandak. Response to the letter to the editor. *Forensic Science International*, 164:282–283, 2006.

[6] F A Bandak. Shaken baby syndrome: A biomechanics analysis of injury mechanisms. *Forensic Science International*, 151:71–79, 2005.

[7] P Bayly, T Cohen, E Leister, D Ajo, E Leuthardt, and G Genin. Deformation of the human brain induced by mild acceleration. *Journal of Neurotrauma*, 22(8):845–856, 2005.

[8] D W A Brands, G W M Peters, and P H M Bovendeerd. Design and numerical implementation of a 3-d non-linear viscoelastic constitutive model for brain tissue during impact. *Journal of Biomechanics*, 37:127–134, 2004.

[9] Susanne C brenner and L Ridgway Scott. *The mathematical theory of finite element methods*. Springer-Verlag, 2002.

[10] Per Brodal. *Sentralnervesystemet*. Universitetsforlaget AS, 3. utgave edition, 2001.

[11] G N Bycroft. Mathematical model of a head subjected to an angular acceleration. *J Biomechanics*, 6:487–495, 1973.

[12] J Caffey. On the theory and practice of shaking infants. its potential residual effects of permanent brain damage and mental retardation . *Am J Dis Child*, 124:161–169, 1972.

[13] J Caffey. The whiplash shaken infant syndrome: manual shaking by the extremities with whiplash-induced intracranial and intraocular bleedings, linked with residual permanent brain damage and mental retardation. *Pediatrics*, 54:396–403, 1974.

[14] B Canaple, G Rungen, P Drazetic, E Markiewicz, and D Cesari. Towards a finite element head model used as a head injury predictive tool. *Int J Crashworthiness*, 8:41–52, 2003.

[15] M E Case, M A Graham, T C Handy, J M Jentzen, and J A Monteleone. Position paper on fatal abusive head injuries in infants and young children. *Am J Forensic Med Pathol*, 22(2):112–122, 2001.

[16] S Cheng, E C Clarke, and L E Bilston. Rheological properties of the tissues of the central nervous system: A review. *Medical engineering and physics*, 30:1318–1337, 2008.

[17] J J Chu, J G Beckwith, J J Crisco, and R M Greenwald. A novel algorithm to measure linear and rotational head acceleration using single-axis accelerometers. *Journal of Biomechanics*, 39 (Suppl 1):S534, 2006.

[18] B Coats and S S Margulies. Potential for head injuries in infants from low-height falls. *J Neurosurg Pediatrics*, 2:321–330, 2008.

[19] C Z Cory and M D Jones. Can shaking alone cause fatal brain injury? a biomechanical assessment of the duhaime shaken baby syndrome. *Med Sci Law*, 43(4):317–333, 2003.

[20] Z Couper and F Albermani. Infant brain subjected to oscillatory loading: material differentiation, properties, and interface conditions. *Biomechanics and modeling in mechanobiology*, 7(2):105–125, 2008.

[21] K A Danelson, C P Geer, J D Stitzel, D E Slice, and E G Takhounts. Age and gender based biomechanical shape and size analysis of the pediatric brain. *Stapp Car Crash Journal*, 52:59–81, November 2008.

[22] K D De Santis Klinich and G M Hulbert. Estimating infant head injury criteria and impact response using crash reconstruction and finite element modeling. *Stapp Car Crash J*, 46:165–194, 2002.

[23] Frank DiMasi. *Tranformation of Nine-Accelerometer-Package (NAP) Data for Replicating Headpart Kinematics and Dynamic Loading*. U.S. Department of Transportation. National Highway Traffic Safety Administration, August 1995.

[24] DOLFIN. The fenics project. *http://fenics.org/wiki/FEniCS_Project.*

[25] A-C Duhaime, C W Christian, L B Rorke, and R A Zimmerman. Nonaccidental head injury in infants - the "shaken-baby syndrome". *The New England Journal of Medicine*, 338(25):1822–1829, June 18 1998.

[26] A-C Duhaime, T A Gennarelli, L E Thibault, D A Bruce, S S Margulies, and R Wiser. The shaken baby syndrome. a clinical pathological, and biomechanical study. *J Neurosurg*, 66:409–415, 1987.

[27] J J Duistermaat and J A C Kolk. *Lie groups*. Springer Verlag, 2000.

[28] K Eriksson, D Estep, P Hansbo, and C Johnson. *Computational Differential Equations*. Studentlitteratur, Lund, 1996.

[29] Howard Eves. *Elementary matrix theory*. Dover publications, 1980.

[30] K K Firoozbakhsh and C N DeSilva. Axisymmetric wave propagation in composite viscoelastic-elastic spheres: a head injury model. *Mathematical biosciences*, 26:241–266, 1975.

[31] K K Firoozbakhsh and C N DeSilva. A model of brain shear under impulsive torsional loads. *J Biomechanics*, 8:65–73, 1975.

[32] Y C Fung. *Biomechanics. Motion, Flow, Stress and Growth.* Springer-Verlag, 1990.

[33] A Gefen, N Gefen, Q Zhu, R Raghupathi, and S S Margulies. Age-dependend changes in material properties of the brain and braincase of the rat. *Journal of Neurotrauma*, 20(11):1163–1177, 2003.

[34] Philip E Gill, Walter Murray, and Margaret H Wright. *Numerical linear algebra and optimization. Volume 1*. Addison-Wesley Publishing Company, 1991.

[35] R M Greenwald, J T Gwin, J J Chu, and J J Crisco. Head impact severity measures for evaluating mild traumatic brain injury risk exposure. *Neurosurgery*, 62(4):789–798, April 2008.

[36] A Guillaume, D Osmont, D Gaffie, J C Sarron, and P Quandieu. Effects of perfusion on the mechanical behavior of the brain exposed to hypergravity. *Journal of Biomechanics*, 30(4):383–389, 1997.

[37] W N Hardy, C D Foster, M J Mason, K H Yang, and A I King. Investigation of head injury mechanisms using neutral density technology and high-speed biplanar x-ray. *Stapp Car Crash Journal*, 45:1–32, November 2001.

[38] W N Hardy, M J Mason, C D Foster, C S Shah, J M Kopacz, K H Yang, A I King, J Bishop, M Bey, W Anderst, and S Tashman. A study of the response of the human cadaver head to impact. *Stapp Car Crash J.*, 51:17–80, October 2007.

[39] J Ho. *Generation of Patient Specific Finite Element Head Models*. PhD thesis, Division of Neuronic Engineering, School of Technology and Health, Royal Institute of Technology, Stockholm, 2008.

[40] J Ho and S Kleiven. Can sulci protect the brain from traumatic injury? *Journal of Biomechanics*, 42:2074–2080, 2009.

[41] A H S Holbourn. Mechanics of head injuries. *The Lancet*, 242(6267):438–441, 1943.

[42] G A Holtzapfel. *Nonlinear solid mechanics. A continuum approach for engineering*. John Wiley & Sons, Ltd, 2000. Reprinted with corrections 2006.

[43] M Hrapko, J A W van Dommelen, G W M Peters, and J S H M Wismans. The influence of test conditions on characterization of the mechanical properties of brain tissue. *J Biomech Eng*, 130:031003, June 2008.

[44] C Jenny, T Shams, N Rangarajan, and T Fukuda. Development of a biofidelic 2.5 kg infant dummy and its application to assessing infant head trauma during violent shaking. *In:Proceedings of the 30th International Workshop on Injury Biomechanics Research, Ponte Vedra Beach, Florida*, pages 129–143, 2002.

[45] H T Keenan and S L Bratton. Epidemiology and outcomes of pediatric traumatic brain injury. *Dev Neurosci*, 28(4-5):256–263, 2006.

[46] S Kleiven. Predictors for traumatic brain injuries evaluated through accident reconstructions. *Stapp Car Crash Journal*, 51:81–114, October 2007.

[47] S Kleiven and H von Holst. Consequences of head size following trauma to the human head. *Journal of Biomechanics*, 35:153–160, 2002.

[48] T J Kriewall, G K McPherson, and A C Tsai. Bending properties and ash content of fetal cranial bone. *J Biomechanics*, 14:73–79, 1981.

[49] S Kumaresan, S Radhakrishnan, and N Ganesan. Generation of geometry of closed human head and discretisation for finite element analysis. *Med Biol Eng Comput*, 33:349–353, 1995.

[50] L D Landau and E M Lifshitz. *Theory of Elasticity*, volume 7 of *Course of theoretical physics*. Pergamon Press, 3rd edition edition, 1986.

[51] Hans Petter Langtangen. *Computational Partial Differential Equations, Numerical methods and Diffpack Programming*. Springer Verlag, second edition edition, 2003.

[52] R J Lapeer and R W Prager. Fetal head moulding: finite element analysis of a fetal skull subjected to uterine pressures during the first stage of labour. *Journal of Biomechanics*, 34:1125–1133, 2001.

[53] David C Lay. *Linear Algebra and its applications*. Addison-Wesley publishing company, second edition edition, 2000.

[54] M-C Lee and R C Haut. Insensitivity of tensile failure properties of human bridging veins to strain rate: implication in biomechanics of subdural hematoma. *J Biomechanics*, 22(6/7):537–542, 1989.

[55] A Levchakov, E Linder-Ganz, R Raghupathi, S S Margulies, and A Gefen. Computational studies of strain exposures in neonate and mature rat brains during closed head impact. *Journal of Neurotrauma*, 23(10):1570–1580, 2006.

[56] Y K Liu (Ed.). Letter to the editor on: Measurement of angular acceleration of a rigid body using linear accelerometers by padgaonkar et al. *Journal of Applied Mechanics*, 43:377–378, 1976.

[57] K A Mardal, X-C Tai, and R Winther. A robust finite element method for Darcy-Stokes flow. *SIAM J. Numerical Analysis*, 40:1605–1631.

[58] Jerrold E Marsden and Thomas J R Hughes. *Mathematical foundations of elasticity*. Dover Publications, 1983.

[59] P G Martin, G W Hall, J R Crandall, and W D Pilkey. Measuring the acceleration of a rigid body. *Shock and vibration*, 5(4):211–224, 1998.

[60] G K McPherson and T J Kriewall. The elastic modulus of fetal cranial bone: A first step towards an understanding of the biomechanics of fetal head molding. *J Biomechanics*, 13:9–16, 1980.

[61] G K McPherson and T J Kriewall. Fetal head molding: An investigation utilizing a finite element model of the fetal parietal bone. *J Biomechanics*, 13:17–26, 1980.

[62] K Miller, A Wittek, G Joldes, A Horton, T Dutta-Roy, J Berger, and L Morriss. Modelling brain deformations for computer-integrated neurosurgery. *Communications in numerical methods in engineering*, DOI: 10.1002/cnm, 2009.

[63] R A Minns and J K Brown, editors. *Shaking and other non-accidental head injuries in children*, volume 162 of *Clinics in Developmental Medicine*. Mac Keith Press, 2005.

[64] R A Minns and Brown J K. *Shaking and other non-accidental head injuries in children*, volume No. 162 of *Clinics in Developmental Medicine*, chapter 1, pages 1–105. Mac Keith Press, 2005.

[65] K Mizuno. Development of three-year old child human fe model. *IRCOBI Conference, Graz, Switzerland*, pages 335–336, 2004.

[66] C N Morison. *The dynamics of shaken baby syndrome*. PhD thesis, School of Manufacturing and Mechanical Engineering, The University of Birmingham, June 2002.

[67] C N Morison and R A Minns. *Shaking and other non-accidental head injuries in children*, volume No. 162 of *Clinics in Developmental Medicine*, chapter 2, pages 106–146. Mac Keith Press, 2005.

[68] M C Myhre, J Grogaard, G A Dyb, L Sandvik, and M Nordhov. Traumatic head injury in infants and toddlers. *Acta Paediatrica*, 96(8):1159–1163, 2007.

[69] Waldyr Muniz Oliva. *Geometric mechanics*. Springer Verlag, 2002.

[70] A J Padgaonkar, K W Krieger, and A I King. Measurement of angular acceleration of a rigid body using linear accelerometers. *Transactions of the american society of mechanical engineers*, 42:552–556, 1975.

[71] G Plank, H Weinstock, M Coltman, and H Lee. *Methodology for the calibration of and data acquisition with a six-degree-of-freedom acceleration*

measurement device. U.S. Department of transportation, National highway traffic safety administration, June 1989.

[72] J Plunkett. Fatal pediatric head injuries caused by short-distance falls. *Am J Forensic Med Pathol*, 22(1):1–12, 2001.

[73] M T Prange, B Coats, A-C Duhaime, and S S Margulies. Anthropomorphic simulations of falls, shakes, and inflicted impacts in infants. *J Neurosurg*, 99:143–150, 2003.

[74] M T Prange and G Kiralyfalvi. Pediatric rotational inertial brain injury: the relative influence of brain size and mechanical properties. *Proceedings of the 43rd Stapp Car Crash Conf, SAE*, pages 333–341, 1999.

[75] M T Prange and S S Margulies. Regional, directional, and agedependent properties of the brain undergoing large deformation. *Trans ASME J Biomech Eng*, 124:244–252, 2002.

[76] N Rangarajan and T Shams. Letter to the editor. *Forensic Science International*, 164:280–281, 2006.

[77] Elmer G Rees. *Notes on geometry*. Springer Verlag, 4th printing edition, 2000.

[78] T O Rognum, M Arnestad, T Bajanowski, J Banner, P Blair, A Borthne, R Byard, P Gaustad, M Gregersen, J Grogaard, E Holter, C Isaksen, J Jorgensen, H Krous, C de Lange, I Moore, J Morland, S Opdal, P Rasten-Almqvist, M Schlaud, P Sidebotham, K Skullerud, G Stoltenburg-Didinger, A Stray-Pedersen, L Sveum, and A Vege. Consensus on diagnostic criteria for the exclusion of sids. *Scand J of Forens Sci*, 9(3/4):62–73, 2003.

[79] S Roth, J-S Raul, B Ludes, and R Willinger. Finite element analysis of impact and shaking inflicted to a child. *International journal of legal medicine*, 121:223–228, 2007.

[80] S Roth, J-S Raul, J Ruan, and R Willinger. Limitation of scaling methods in child head finite element modelling. *Int J vehicle Safety*, 2(4):404–421, 2007.

[81] S Roth, J-S Raul, and R Willinger. Biofidelic child head fe model to simulate real world trauma. *Computer methods and programs in biomedicine*, 90:262–274, 2008.

[82] S Roth, J S Raul, and R Willinger. Finite element analysis of child head injuries in the field of child abuse. *Computer methods in biomechanics and biomedical engineering*, 11(1):199–201, 2008.

[83] J-C Sarron, C Blondeau, A Guillaume, and D Osmont. Identification of linear viscoelastic constitutive models. *Journal of Biomechanics*, 33:685–693, 2000.

[84] M R Seemann and L S Lustick. Combination of accelerometer and photographically derived kinematic variables defining three-dimensonal rigid body motion. *Biomechanics cinematography*, 291:133–140, 1981.

[85] S Shaw. *Finite element and discrete time methods for continuum problems with memory and applications to viscoelasticity*. PhD thesis, Department of Mathematics and Statistics, Brunel University, November 1993.

[86] A Stray-Pedersen and T O Rognum. Shaken baby syndrome: Commentary on current debate. *Scand J of forensic sciences*, 2:79–83, 2006.

[87] K L Thibault and S S Margulies. Age-dependent material properties of the porcine cerebrum: Effect on pediatric inertial head injury criteria. *Journal of biomechanics*, 31:1119–1126, 1998.

[88] N W Tschoegl. *The phenomological theory of linear viscoelastic behaviour. An introduction.* Springer-Verlag, 1989.

[89] J A W van Dommelen, M Hrapko, and G W M Peters. *Mechanosensitivity of the nervous system*, volume Vol 2 of *Mechanosensitivity in Cells and Tissues*, chapter 12, pages 249–279. Springer Science+Business media B. V., 2009.

[90] C D Waagmeister. *Q0 dummy users guide.* First technology safety systems, http://www.ftss.com/.

[91] Wikipedia. *http://en.wikipedia.org/wiki/Anatomical_terms_of_location #Medical_.28human.29_directional_terms.*

[92] Wikipedia. Hic. *http://en.wikipedia.org/wiki/Head_injury_criterion.*

[93] Wikipedia. Lamé parameters. *http://en.wikipedia.org/wiki/Lamé_parameters.*

[94] R Willinger and D Baumgartner. Human head tolerance limits to specific injury mechanisms. *Int J Crashworthiness*, 8(6):605–617, 2003.

[95] A Wittek and K Omori. Parametric study of effects of brain-skull boundary conditions and brain material properties on responses of simplified finite element brain model under angular acceleration impulse in sagittal plane. *JSME International Journal*, 46(4):1388–1399, 2003. Series C.

[96] Jens Wittenburg. *Dynamics of systems of rigid bodies*. B G Teubner Stuttgart, 1977.

[97] D R Wolfson, D S McNally, M J Clifford, and M Vloebergs. Rigid-body modelling of shaken baby syndrome. *J Engineering in Medicine*, Proceedings of the Institution of Mechanical Engineers, 219, Part H:63–70, 2005.

[98] C Zhou, T B Khalif, and A I King. A new model comparing impact responses of the homogeneous and inhomogeneous human brain. *Stapp Car Crash Conference Proceedings*, pages 121–137, 1995.

[99] M Zoghi-Moghadam. *The role of CSF and SAS trabeculae in head/brain injuries: A new local/global and solid/fluid model.* PhD thesis, The City University of New York, 2006.

[100] M Zoghi-Moghadam and A M Sadegh. Global/local head models to analyse cerebral blood vessel rupture leading to asdh and sah. *Computer Methods in Biomechanics and Biomedical Engineering*, 12(1):1–12, February 2009.

[101] H Zou. *Analysis and modeling of the biomechanics of brain injury under impact*. PhD thesis, The Ohio State University, 2007.