Morphoelastic Modelling of Embryonic Growth

Pascalle Wijntjes

Morphoelastic Modelling of Embryonic Growth

by

Pascalle Wijntjes

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday November 6, 2019 at 14:00 PM.

Student number:	4349482	
Project duration:	February 1, 2019 – November 6, 2019	
Thesis committee:	Dr. ir. F.J. Vermolen,	TU Delft, supervisor
	F.W.A. de Pont MSc. LLB.,	COMSOL B.V., supervisor
	Dr. J.L.A. Dubbeldam,	TU Delft

This thesis is confidential and cannot be made public until November 6, 2019.

TU Delft Delft University of Technology

Abstract

In mathematical biology, a morphoelastic model, which combines elasticity and growth, has been used for, e.g., wound healing and tumour growth modelling. Here, it is used to model the growth of an embryo. In order to do so, a growth curve from literature has been fitted to a morphoelastic model which has been extended with a PDE for the normalised cell concentration. In the one dimensional situation, this fitting resulted into a curve with a shape similar to the desired shape. Despite the similar shape, it did not fulfil the total growth requirements; it fell short by a factor 100. Instead, a Gaussian curve was used to represent the cell concentration. In one dimension this gave a similar looking curve, with a relative error of 0.2316 for the first couple of weeks of the gestational age of the embryo, where the length of an embryo is measured form crown-to-rump. For the other weeks, when the length of the embryo is measured from crown-to-heel, a relative error of 0.0507 was observed.

The one dimensional problem is solved using the finite element method with linear basis functions and an Euler backwards time integration method. This is implemented in MATLAB. The output is compared to results of a COMSOL Multiphysics® file created on this topic and verified to be similar.

In three dimensions, the model without the cell concentration contribution is verified. For this, experiments have been done in which a force is applied in various directions separately. All results were similar to the one dimensional model with a pulling force. The model containing cell concentration is not verified in this thesis. Further, the effects of the growth term and mesh choice have been investigated. These calculations have all been done with COMSOL Multiphysics®.

Preface

The thesis in front of you was written by me in order to fulfil the graduation requirements of the master in Applied Mathematics at the TU Delft. For this, I did a research on mathematically modelling of the growth of a human embryo. Between February and November 2019, I was engaged in researching and writing this thesis.

Together with my supervisor from the TU Delft, F. Vermolen, I created the subject of this research. I liked to have social an medical influences in my project. Also, I liked to have a subject about something that I know personally. This last wish was what made me do the project on the topic of embryonic growth. This research can be used in the future to investigate birth defects, which one of my acquaintances is affected by. This also makes it socially and medically relevant.

I would like to thank both of my supervisors for their guidance and support during these months. I also would like to give special thanks to COMSOL Multiphysics® and especially F. de Pont, my supervisor from inside the company. It was great to have the opportunity to work at the office and use the platform for my research. Finally, I would like to thank my friends and family, who always have believed in me.

Enjoy the ride through this thesis!

Pascalle Wijntjes

Delft, Nov. 2019

Contents

List of Symbols

x, y, z	Eularian coordinates
X, Y, Z	Lagrangian coordinates
h	node distance
Ω	three dimensional working space
t	time
T	end time
Δt	time step
u	displacement
v	displacement velocity
L	velocity gradient
ε	infinitesimal strain (tensor)
σ	Cauchy stress (tensor)
c	cell concentration
F	external force
ρ	mass density
F_b	body force
G	growth tensor
C_g	constant growth factor
E	Young's modulus
μ	shear modulus
μ_1	shear viscosity
μ_2	bulk viscosity
ν	Poisson ratio
β	proliferation rate
D	diffusion rate of cells
k	force constant
$\mathbb{I}_3$	identity matrix (3×3)
φ_i	basis function
ψ	test function

List of Figures

List of Tables

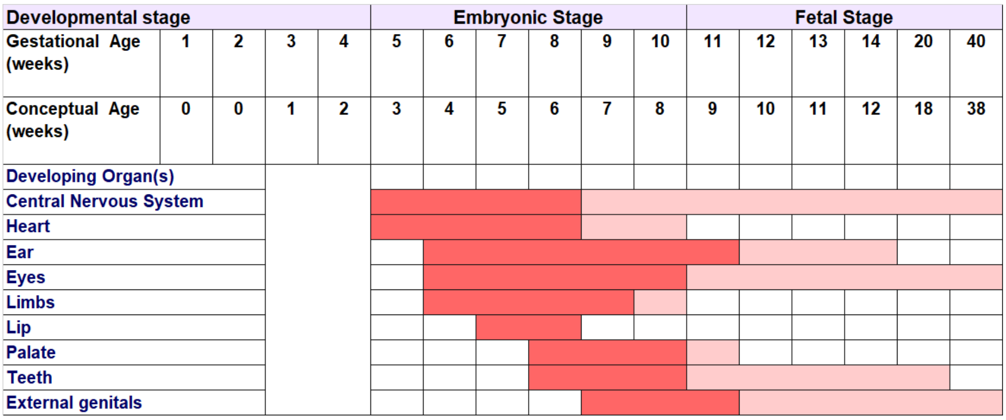
1 Introduction

In the Netherlands, each year around 5 000 babies are born with birth defects [1]. A proportion of them originates from abnormal growth during the embryonic phase. Modelling of the growth and dynamics of the embryo during this phase can give insights in the underlying process and the possible things that can go wrong. With these insights, therapies can be discovered to prevent those birth defects from occurring. The aim of this thesis is to make a contribution to the available models of embryonic growth, from which insights can be derived, by creating a morphoelastic model that simulates the dynamics and the growth of an embryo.

1.1 Embryonic and Fetal Growth

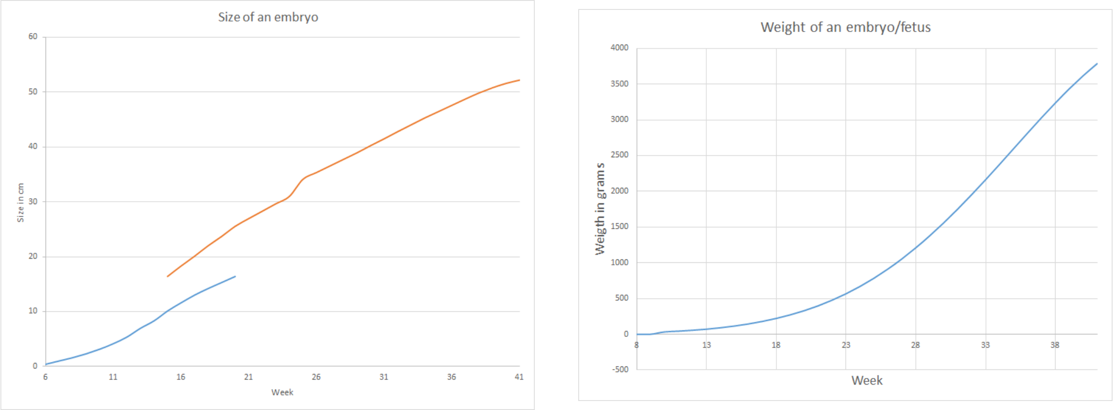
After fertilisation, the cell divides into a cluster of cells. When the cell-cluster is implanted in the uterus, one talks about an embryo and a placenta. In the uterus, the embryo grows to a fetus with the help of the oxygen and nourishments supplied through the placenta [2].

Subsequently the embryo develops into three different layers. After about two weeks (conceptual age), the embryo is shaped like a disc, which grows in the next two weeks to a sphere-like object of around 4 mm in diameter [3]. From this point on, the embryo starts to evolve into a human shape and develops organs and tissues. This time is a crucial moment for the health of the baby. For example, when an organ or limb does not start to grow properly, a handicap can grow, such as anorectal malformations [4, 5], see also Figure 1.1. At the end of this embryonic stage, the embryo is seven to eight weeks old and its crown-to-rump length is about 2.3 cm [3].



Developmental stage					Embryonic Stage						Fetal Stage					
Gestational Age (weeks)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	20	40
Conceptual Age (weeks)	0	0	1	2	3	4	5	6	7	8	9	10	11	12	18	38
Developing Organ(s)																
Central Nervous System					🟥	🟥	🟥	🟥	▨	▨	▨	▨	▨	▨	▨	▨
Heart					🟥	🟥	🟥	🟥	▨	▨						
Ear						🟥	🟥	🟥	🟥	🟥	🟥	▨	▨	▨		
Eyes						🟥	🟥	🟥	🟥	🟥	▨	▨	▨	▨	▨	▨
Limbs						🟥	🟥	🟥	🟥	▨						
Lip							🟥	🟥								
Palate								🟥	🟥	🟥	▨					
Teeth								🟥	🟥	🟥	▨	▨	▨	▨	▨	
External genitals									🟥	🟥	🟥	▨	▨	▨	▨	▨

Figure 1.1: Fetal Development Chart. Red bars show when the different organs are most prone to birth defects. The pink bars show when those organs are sensitive to functional defects and minor malformations [3].



(a) Length curve of fetal growth.

(b) Weight curve of a fetus.

Figure 1.2: Growth curves of a fetus. In 1.2a, the length of the fetus is plotted. The blue line shows the crown-to-rump length of the first stage. The red line presents the crown-to-heel length of the last 2/3rd of the pregnancy. In 1.2b, the weight of the fetus is shown. Later in this thesis, this length curve (1.2a) will be used to compare the obtained results.

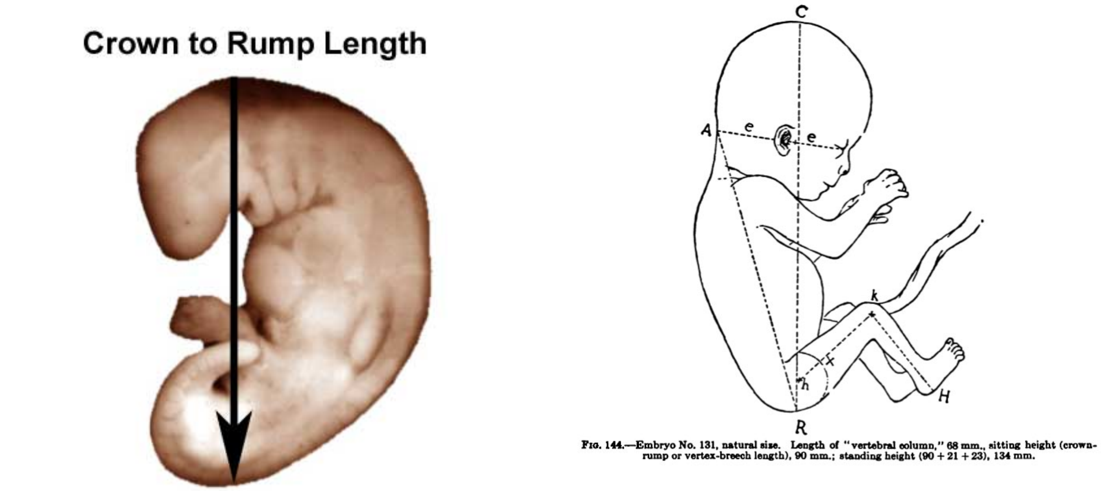


FIG. 144.—Embryo No. 131, natural size. Length of "vertebral column," 68 mm., sitting height (crown-rump or vertex-breech length), 90 mm.; standing height (90 + 21 + 23), 134 mm.

(a) Crown-to-rump length of an embryo [6].

(b) Size of a fetus. C – R is the crown-to-rump length, and C – H the crown-to-heel length [6].

Figure 1.3: These figures show how the length of an embryo and a fetus is measured.

When all the body elements are settled, the whole body can grow until it is ready for birth. For this, all the organs, limbs, and tissues, need to grow to an appropriate size and need to function well. This period, from week eight till week 38/40 of the pregnancy is called the fetal growth period. During this time the fetus grows to a crown-to-heel length of approximately 52 cm and to a weight of around 3500 grams.

In Figure 1.2 it is shown how an embryo develops into a fetus which is ready for birth. The length and corresponding weights are given in those figures. The weight of an embryo is only measured from week 8 on, while the length is already known from week six. During the first couple of weeks, the length is measured as the crown-to-rump length (Figure 1.3a). Thereafter, the length of the embryo is measured from crown-to-heel (Figure 1.3b). In Figure 1.2a this is made clear by different lines. The first period is given in blue, while the second period, which uses another measurement approach, is given in red. These plots have been created from the table in Appendix A.

In this thesis, the term embryo will be used to refer to the unborn child in both the embryonic and fetal phase.

1.2 Previous Research

In the search for a mathematical model for embryonic growth, one will see that this topic has not been studied extensively under the scientists yet. Though, there are some other relevant researches in this area. One can think of mathematical modelling of tumour growth, or wound healing after, for example, burns.

Mathematical research to wound healing was started not long ago by Tranquillo and Murray [7]. With their obtained model, they described the behaviour of tissues mathematically. It describes the (dermal) tissue as an isotropic linear viscoelastic solid. They extended the model by adding various cell properties such as the traction force applied by fibroblasts.

After a while, Olsen et al. [8] modified and extended the model even more. They added myofibroblasts to the model and collagen molecules replaced the extracellular matrix as a model variable.

Both research teams have verified their results by comparing it to experimental data on wound contraction in rats obtained by McGrath and Simon [9]. They could use these data since their models contained isotropic stresses produced by different cells.

Later in time, Hall [10] worked on evolution equations which describe the change of effective strain over time mathematically. This used multiple states of a system: one in which it describes the current strains and one in which it is in relaxation. Therefore, it could be used for modelling the dermal layer over time [11]. In order to work with his results, the effective strains should be small.

The resulting evolution equation has been used by Koppenol [11] to describe the dynamic change of the infinitesimal effective strain over time. In his dissertation, he investigated mathematical models for wound healing in different settings. The last model he introduced was a morphoelastic model using the evolution equation for the effective strain by Hall. Although this thesis is not about wound healing, the model he used there has been used here as well.

Looking into another direction, biological cell behaviour is also described in literature by cellular-automata [12, 13, 14]. Vivas et al. [12] used this type of modelling to describe the cell population behaviour by cell adhesion and proliferation. The cells formed a sphere which was retained through the rest of simulation time.

1.3 Purpose of this Thesis

In this thesis, the main focus is on the mathematical model of morphoelasticity. The growth of an embryo is simulated with this model in MATLAB (only in 1D) and COMSOL Multiphysics®. The models used are based on the one for wound healing described in [10] and [11].

The idea is that, with this model, the growth of an embryo can be simulated as well. With a change of some parameters, effects leading to a birth defect can be generated. In future research, this could be an important aspect for which the models of this thesis could be used. Birth defects could be modelled and, therefore, be understood better.

To achieve this goal, a morphoelastic model is set up and investigated mathematically, see Section 2. The numerical elaborations are shown in Section 3. The three dimensional model is verified by using the one dimensional solutions, which themselves are verified by comparisons with simpler models, e.g., viscoelasticity, and with a manufactured solution. The results and verifications are given in Section 4. Finally, there are some concluding remarks, a discussion on this research and ideas for future research on this topic in Section 5.

The main focus of this thesis is the question *"Is it possible to model the growth of an embryo by using morphoelastics models?"* This research question will be substantiated by the following subquestions:

- What is a morphoelastic model? And where has it been used for before? (Section 1.2, Section 2)
- How does an embryo grow? (Section 1.1)
- Is cell density a good source to stimulate the growth? And how can this be implemented? (Section 2.3)
- What method could be used for solving a morphoelastic problem? (Section 3)

1.4 Physical Properties of an Embryo

The morphoelastic model used in this thesis is based on the physical concept of elasticity. For this model, the physical quantities of the Young's modulus (or elastic modulus), shear modulus, and Poisson ratio are needed. The values for these constants can be measured in various ways, e.g. via a tension or suction test, and on different parts of the body.

A lot of research has been done on the Young's modulus of the human skin. Unfortunately, the obtained results vary widely. The fact that the results vary widely is discussed in [15]. This research mentions that different techniques for measuring and the different positions on the body can give such an effect in the results. Next to that, when getting older, the skin becomes thinner, stiffer, less tense and less flexible, which results in a higher Young's modulus [16, 17], thus it varies per person.

An embryo develops into three layers after a few weeks. At this moment, the skin tissue starts to develop [18]. Therefore, the results of given references on the Young's modulus of the skin are used in this research. After considering all those references, a Young's modulus of 0.23 MPa has been chosen. How this value is established can be seen in Appendix B.

The viscosity values of a human are hard to find in literature. This is substantiated by the fact that Koppenol has estimated the value of the shear viscosity and bulk viscosity for his research [11]. He did the same for the Poisson ratio.

For the Poisson ratio he took 0.49, which is close to a fully incompressible material (ν=0.5). While playing with the skin, one sees that the skin moves along. In its behaviour one could compare the skin to rubber material. Both show large lateral compression under longitudinal tensile load. Rubber has a Poisson ratio close to 0.5 [19]. Therefore, it is chosen to take a Poisson ratio of 0.49 in this thesis as well.

The human body consist of a lot of water. The body gets dryer wile getting older. So, when a child is born the body is the wettest. Therefore, it is chosen to use the viscosity of water, which is 0.6978 mPa s for a temperature of 37 degrees[20].

Lastly, a mass density of 826 kg/m^3 is used. This number is derived from the length, weight and abdominal circumference of the embryo. It turned out that this number is more or less constant over time, and hence assume this value to be constant. The values used can be found in Appendix A.

All other parameters are fitted in this research to get a curve which looks like the growth of an embryo, see Figure 1.2a. This thesis focuses on writing a mathematical model for embryonic growth. This means that the values of the mechanical properties are not the main focus here. It is, however, interesting to investigate this further in future research, such that well-found numbers can be used.

2 Introduction into the Mathematical Models

The skin of a vertebrate, and in particular of a human, is viscoelastic [21, 22, 23]. This can also be seen when you pull your skin, and release it. It swiftly goes back to its original state. It has some memory of how it was before. When doing this experiment with your grandparents, you will notice that it takes a little more time for the skin to recover. This is since the tissue gets stiffer over time, which has also been discovered in Section 1.4. The skin also has anisotropic characteristics [15], however in this thesis anistropy is not taken into account.

When looking at a human embryo, it will also grow, next to stretch. Combining these two phenomena, one could use the theory of morphoelasticity to describe an embryo. That is what has been done in this thesis.

In this section, we dive into the formulas of morphoelasticity, based on the work of Hall [10] and Koppenol [11]. First, it will be described in one dimension. Then the general version is given, which will be used for the three dimensional case. In the end, properties of the cell will be added to the morphoelastic models to create more realistic results.

2.1 One-Dimensional Morphoelastic Model

First a simple model is explored. The simple model consists of only one dimension.

The Morphoelastic PDEs

The model of morphoelasticity is an expanded form of the mathematical description of viscoelasticity, which in itself is an expansion of linear elasticity. How the latter two models are setup, is given in Appendix C. Results of the viscoelastic model will be used to verify results of the morphoelastic model. These results can easily be compared when the growth term in the morphoelastic problem is set equal to zero.

The mathematical model of morphoelasticity is described by [10, 11]:

$$\begin{cases} \rho\left(\dfrac{Dv}{Dt}+v\dfrac{\partial v}{\partial x}\right)-\dfrac{\partial \sigma}{\partial x}=F_b & x\in(x_0,x_L], \quad t\in(0,T]; \\ \sigma=\mu\dfrac{\partial v}{\partial x}+E\varepsilon & x\in(x_0,x_L], \quad t\in(0,T]; \\ \dfrac{D\varepsilon}{Dt}+(\varepsilon-1)\dfrac{\partial v}{\partial x}=-G & x\in(x_0,x_L], \quad t\in(0,T]; \end{cases} \tag{P1}$$

subject to

$$\begin{cases} v(x,0) = 0 & x \in [x_0, x_L]; \\ u(x,0) = 0 & x \in [x_0, x_L]; \\ \sigma = F & x = x_L, \quad t \in (0, T1]; \\ \sigma = 0 & x = x_L, \quad t \in (T1, T]. \end{cases} \tag{BC1}$$

Since the above equations are solved for v and ε, this requires a post-processing step to obtain the displacement u:

$$\begin{cases} \dfrac{Du}{Dt} = v \\ x(t) = X + u(x,t). \end{cases}$$

The displacement indicates how much the initial body has been grown. Together with the initial size, the final length can be calculated. The evolution equation of the infinitesimal strain (tensor) is derived by Hall [10] and given in Appendix D.

The physics are illustrated in Figure 2.1. It is chosen to first look at a rod of length x_L which is fixed on one side ($x_0 = 0\ m$). On the other end, an external force F is applied.

The differential equations require a material derivative (total derivative) due to movement of the domain. Since the equations will be solved in the material frame, this type of derivative is used instead of a partial time derivative in order to measure the time rate of change. This setting will make the calculations easier as no interpolation is needed between time steps and adjacent mesh nodes. How to work in the Eularian setting is described in Section 3 by defining a moving mesh.

One Dimensional Problem Statements

The first setup is the pulling rod problem. This is where one side of a rod is fixed and a force is applied on the other side. For this problem the external force F is nonzero, but the growth tensor G is equal to zero.

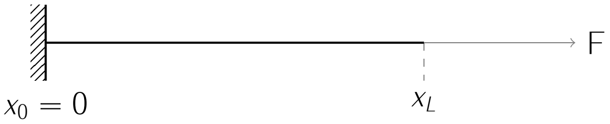


Figure 2.1: The pulling rod problem. In one dimension, the rod is hold on the left side, where $x = 0\ m$. On the other side, $x = x_L$, an external force F is applied.

The second problem statement is when growth comes into the system. This can be done by a nonzero G, and either a zero or nonzero external force F. The drawing of Figure 2.1 is also applicable to this problem statement. The latter option will afterwards be stimulated by cell properties of a human being. This is further explained in Section 2.3.

The growth tensor G is set to be constant or linear dependent on the infinitesimal strain (tensor) ε, as was done in [11].

In all scenarios, except for the stimulated growth, the length of the rod is set to $1\ m$. This means that the value for x_l is $1\ m$. For the stimulated growth, the length is set to $2\ mm$, since growth is first measured from week 6 of the gestational age onwards (Section 1.1). Because of the choice of symmetric growth, the total initial length will be $4\ mm$ as is the average embryo size in week 6. This ensures that the results can be compared to Figure 1.2a.

2.2 Three-Dimensional Morphoelastic Model

Morphoelastic PDEs

When generalising the morphoelastic formulas given in Section 2.1, they become much more complex, especially the formula for the infinitesimal strain (tensor) ε. How these equations are derived can be found in Hall [10]. The equation for the Cauchy stress (tensor) σ is directly derived from the isotropic Hooke's law for the purely elastic part. In addition, the balance of momentum, which is the partial differential equation for the displacement velocity v, is easily transformed into multiple dimensions. Usage of the divergence operator replaces the partial space derivatives.

Below, the three dimensional morphoelastic system corresponding to the x-directional pulling bar problem discussed later on are given. For this, the domain $\Omega \subset \mathbb{R}^3$ is used. It will have a fixed plane ($\partial\Omega_1$) on $x = 0\ m$, and on the opposite side, $x = x_L$ ($\partial\Omega_2$), an external force will be applied. The bar will have a thickness of $0.2 \times 0.2\ m$.

The three dimensional morphoelastic differential equations are given by [10, 11]:

$$\begin{cases} \rho\left(\frac{D\mathbf{v}}{Dt} + \mathbf{v}\nabla\cdot\mathbf{v}\right) - \nabla\cdot\sigma = \mathbf{F_b}, & \mathbf{x}\in\Omega, \quad t\in(0,T]; \\ \sigma = \mu_1\,\mathrm{sym}(L) + \mu_2 tr(\mathrm{sym}(L))\mathbb{I} + \dfrac{E}{1+\nu}\left(\varepsilon + \dfrac{\nu}{1-2\nu}tr(\varepsilon)\mathbb{I}\right), & \mathbf{x}\in\Omega, \quad t\in(0,T]; \\ L = \nabla\mathbf{v}, & \mathbf{x}\in\Omega, \quad t\in(0,T); \\ \frac{\mathcal{D}\varepsilon}{\mathcal{D}t} + (\mathrm{tr}(\varepsilon)-1)\mathrm{sym}(L) = -G, & \mathbf{x}\in\Omega, \quad t\in(0,T); \\ \frac{\mathcal{D}\varepsilon}{\mathcal{D}t} = \frac{D\varepsilon}{Dt} + \varepsilon\,\mathrm{skw}(L) - \mathrm{skw}(L)\varepsilon, & \mathbf{x}\in\Omega, \quad t\in(0,T); \end{cases} \tag{P2}$$

subject to

$$\begin{cases} \mathbf{v}(\mathbf{x}, 0) = \mathbf{0} & \mathbf{x} \in \Omega; \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{0} & \mathbf{x} \in \Omega; \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{0} & \mathbf{x} \in \partial\Omega_1, \quad t \in (0, T]; \\ \sigma \cdot \mathbf{n} = \begin{pmatrix} F & 0 & 0 \end{pmatrix} & \mathbf{x} \in \partial\Omega_2, \quad t \in (0, T1]; \\ \sigma \cdot \mathbf{n} = \mathbf{0} & \mathbf{x} \in \partial\Omega \setminus (\partial\Omega_1 \cup \partial\Omega_2), \quad t \in (T1, T). \end{cases} \tag{BC2}$$

In these equations, the skew-symmetric and the symmetric part of the velocity gradient L are used. These are given by:

$$\text{sym}(L) = \frac{1}{2}(L + L^T)$$
$$\text{skw}(L) = \frac{1}{2}(L - L^T).$$

Note that now, the displacement velocity v and the displacement u are vectors in $\mathbb{R}^3$, and so are the body force F_b, and external force F (given as $\mathbf{F} = \begin{pmatrix} F & 0 & 0 \end{pmatrix}$). For the Cauchy stress (tensor) σ and the infinitesimal strain (tensor) ε the Voigt notation is used. This can be done since we assume to have a symmetric growth tensor, and thus symmetry in the infinitesimal strain (tensor) and the Cauchy stress (tensor). Both are casted in a six dimensional vector instead of a tensor notation. This saves computational time and memory.

As in the one dimensional model, the material derivative is used. Next to that, the Jaumann derivative, which takes into account the rotation of the body, is used to describe the infinitesimal strain (tensor) ε differential equation. This formula is already given as the last equation in ($P2$).

System Setup

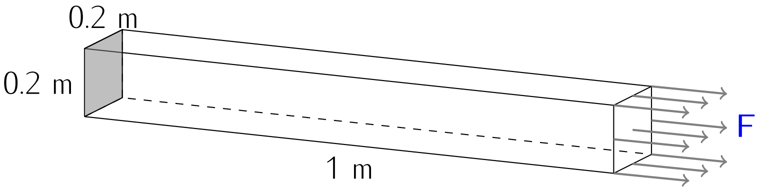
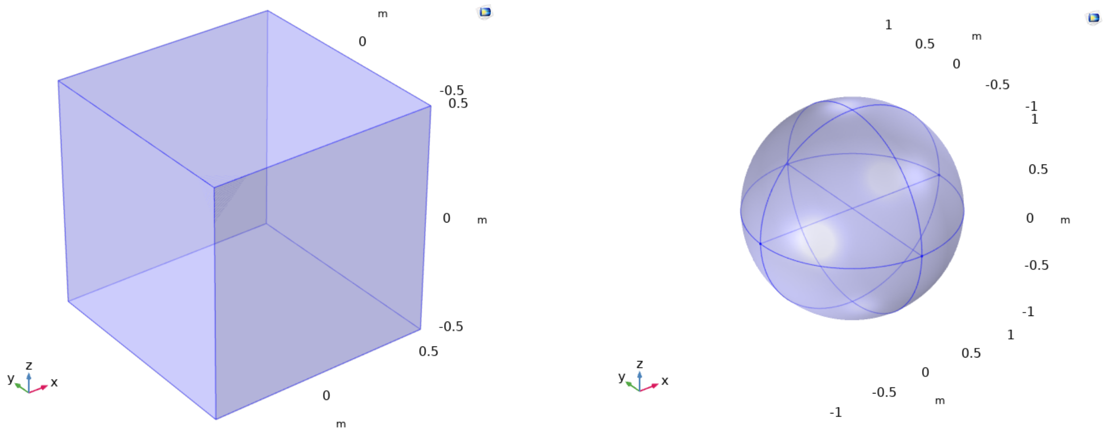


Figure 2.2: The pulling rod problem expanded into its 3D version: the pulling bar problem. The block has a length of 1 meter and its thickness is 0.2×0.2 m. One side (gray) is fixed by a zero constraint and on the opposite side an external force F is applied.

First, the three dimensional version of the pulling and the growing rod problem (the pulling and the growing bar problem) will be investigated. This is mainly for verification. Therefore, the bar will be small in height and width. There has been chosen to examine a bar of 1 m and a height and width of 0.2 m. One squared side is fixed and on the other side an external force F is applied (see Figure 2.2). For the growing bar problem, both with and without an external force have been examined.

Further, a growing cube and sphere will be examined, as shown in Figure 2.3, to look at the impact of using different kinds of meshes (quadrilaterals vs tetrahedrals). Further, growth in multiple directions is implemented. These bodies are starting off as symmetric geometries. The growth tensor used here will also be symmetric. Therefore, and to reduce computation time and safe memory, only 1/8-th of the body will be considered. For this, zero-constraints on v in the normal directions are used on the three symmetry boundaries. This means that the body is forced not to move in those normal directions. The visualisations of the final results of the body shape are then given by using the laws of symmetry.

In the end, cell properties are added to the system to stimulate growth. In order to be able to verify these results, a bar with a smaller size, which is more representative of an embryo, will be investigated. The same length as for the one dimensional case is used for this; 2 mm. The thickness is then said to be 0.1 mm. This stimulant is also tried on a sphere shaped body for a more embryo-like result.



(a) Cube

(b) Sphere

Figure 2.3: Three dimensional setup of growing bodies made with COMSOL Multiphysics$^{\circledR}$ geometry.

2.3 Adding Cell Properties to the Systems

As a last model, a growing rod or bar, without any external forces, but with a body (or internal) force F_b is used. This allows the system to grow by an internal stimulus. It will be stimulated by cellular phenomena, like the cell concentration, proliferation, apoptosis, and so on. These stimuli are used in wound healing models by, e.g., Koppenol [11], but modelled separately. Due to time constraints it has been chosen to use a general formula for this. This PDE then shows the general normalised cell concentration: the normalised amount of cells per unit area. It is added to the system of equations by adding the PDE (P_c) shown below.

$$\begin{cases} \dfrac{Dc}{Dt} + c\nabla \cdot v - \nabla \cdot (D(\mathbf{x},t)\nabla c) = \beta c(1-c), & \mathbf{x} \in \Omega, \quad t \in (0,T); \\ (D(\mathbf{x},t)\nabla c) \cdot \mathbf{n} = 0, & \mathbf{x} \in \partial\Omega; \\ c(\mathbf{x}(0),0) = c_0(\mathbf{x}(0)). & \mathbf{x} \in \Omega \end{cases} \tag{P_c}$$

This equation describes the mass conservation for the cell concentration. It is based on [24], where the diffusion term in its limits can be approximated by the Fisher equation: $D = v^2/4\beta$, with β the proliferation rate. This kind of conservation formula has been used among others in [10, 11, 25]. The difference between (P_c) and the one described in [24] is the source term, which is written as $\beta c(c-1)$ and βc respectively. This is used since in this thesis we deal with a growing body, an embryo, which can and will reach a maximum size. However, a tumour can grow aggressively without boundaries.

Swanson et al. [24], have added chemotherapy to their system, by subtracting kc, with k a constant measure of the effectiveness of the treatment. For this, k must be greater than proliferation rate β, in order to reduce the size of the tumour. In our formula, (P_c), this works similarly. Whenever the cell concentration c reaches the value one, the embryo stops growing. This is a stable asymptotic solution, as can be seen in Figure 2.4.

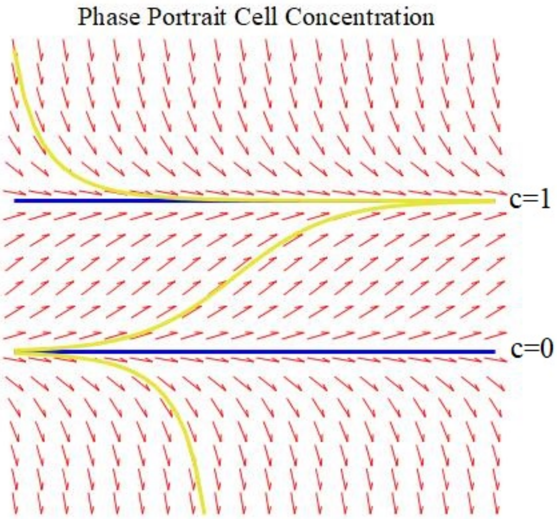


Figure 2.4: Phase portrait of (P_c), where D(x,t)=0. Various phase lines are plotted; $c(0) < 0, 0 < c(0) > 10$, and $c(0) > 1$. One sees that $c = 0$ is an unstable solution and $c = 1$ stable.

The results of this PDE will be implemented as a body force F_b for the displacement velocity v PDE in ($P1$), as in Koppenol [11]. For this the negative spatial derivative of the cell concentration is multiplied by a force constant k , that is:

$$Fb = -k\nabla c. \tag{2.1}$$

According to [23], the internal forces are given by cytoskeletal contraction, actin polymerisation, microtube elongation and shortening, adhesion, differential growth, and swelling of the ECM (extracellular matrix). They also note that cells migrate through the embryo and exchange neighbours to change tissue shape, which motivates the usage of a diffusion equation for the cell concentration.

In order to let the body grow, there need to be a stimulant in the system. Therefore, the cell concentration PDE needs to have non-zero initial conditions. The diffusion equation (P_c) has a Gaussian as homogeneous solution [26]. Therefore a normal distribution, with mean zero and standard deviation s, is used as an initial condition. Hence, the rod, or the three dimensional body, is centred at zero. This point in space will be fixed with a zero constraint for the displacement velocity v. The Gaussian is suppressed by the value of c_0 in order to let the cell concentration be less than one. A schematic one dimensional overview has been given in Figure 2.5.

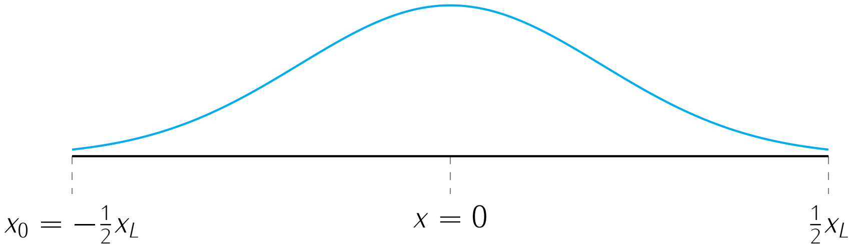


Figure 2.5: The schematic one dimensional growing rod. The rod is stimulated by cell phenomena to grow in both positive and negative x-direction from the centre. Therefore, the centre, $x = 0$, is fixed with a zero constraint. The ends of the rod have a zero-flux bound. This makes it possible for the rod to grow freely, while the centre, and therefore its position, is fixed. The blue curve presents the initial cell concentration, which is implemented as a Gaussian.

2.4 Alternative Idea for the Cell Properties PDE

While solving the growing body problem with the use of internal forces stimulated by cell properties, some problems, which were not solvable in an easy way, were encountered. It is known that the solution to (P_c), is a Gaussian, which depends on the ratio of β and D [24]. Here D is the diffusion rate of cells and tells us how fast the initial Gaussian will be spread out over the domain, or physically how fast the cells will be equally divided over the domain. Further, the proliferation rate β describes how fast the maximum amount of cells at a point or in an area will be reached. So, how fast $c(x,t)$ will be equal to one at the point x.

The growth depends on the spatial derivative of the cell concentration c, as described before. To have a larger growth, this derivative should be larger too. In order to achieve this, the ratio between β and D should give a Gaussian with a large derivative, thus has a small standard deviation.

Since we would like to have a slow growth, it should take about 40 weeks before the embryo reaches a length of 52 cm, the diffusion rate of cells D should be small compared to what is expected physically. Whenever the cells are equally divided over the domain, the spatial derivative gets equal to zero, which results in no further growth, only in cell concentration which will then reach one everywhere at the same time.

On the other hand, the proliferation rate β should not be big either. If this number is too big, the spatial derivative is too large at the beginning of the run time. However, the cells will also be spread faster. The reason for this is that if they reach the maximum for that point, the cells will still divide, but diffuse instantaneously to the sides and push the cells there away, until the existing body is fully filled with cells.

The last difficulty that needs to be dealt with is the Péclet number. This is a ratio of the advection and the diffusion in a differential equation. When the number gets too big, the results start to oscillate. In the PDE (P_c), this ratio consists of the terms β and v, and is given by: $Pe = h * \dfrac{\beta}{2v}$, with h the node distance. This ratio should be below one in order to get a smooth non oscillating result if a standard Galerkin discretisation is used. However, one could play with this number by adding an artificial diffusion term [27].

So, a good balance between the diffusion rate of cells and proliferation rate is needed in order to get a curve lasting for at least forty weeks, for a specific choice of h. Although, a large rate is needed to achieve this, it does not look like reality (a very thick stomach, but almost no head and toes, see Figure 2.6).

Then, two more parameters are involved: force constant k and constant growth factor C_g. Those parameters affect value of the maximum displacement u. Increasing these values gives a larger displacement and thus a larger growth of the body. With this a more realistic starting state could be used to get the same valued curve.

The down side of (P_c) is that it also depends on the displacement velocity v, because of the total derivative. This unfortunately pushes the results down, causing the force constant k and constant growth factor C_g to have less effect. In the end it resulted into a curve shape similar to the desired shape (Figure 1.2a) over time, but a very moderate displacement u of 0.24 cm, which is one hundred times smaller than desired.

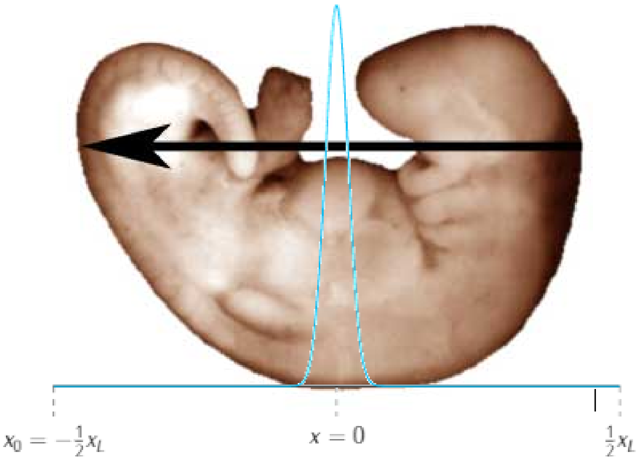


Figure 2.6: The embryonic body together with a tight Gaussian curve. This shows schematically how the cell concentration is presented in the model, which is far from realistic as can be seen in this figure.

Because of this, it is decided to use a different approach with the same idea. Instead of using a PDE for the cell concentration c, one could use a prescribed body force. it is known that (P_c) results in a Gaussian which evolves over time t. Therefore it is chosen to use a standard Gaussian here. This is given as

$$c_1 \left[2\pi s^2\right]^{d/2} \exp\left(-\frac{||\mathrm{x}||^2}{2s^2}\right) * (1 - \exp(-\beta t)) \tag{2.2}$$

where c_1 suppresses the Gaussian such that it does not exceed the value one, s the standard deviation, d the dimension, and β represents here the proliferation rate as well. This rate will cause a delay, starting off with no cells at all. This Gaussian only grows in height, but not in length over time. It therefore does not adjust to the current state. For that, the standard deviation should also be time dependent.

Again the negative spatial derivative of this function will be implemented as the body force F_b, together with the force constant k. In order to verify the results in three dimensions, the following setup has been tested: The body force is described in the length direction of the rod by the negative spatial derivative of the Gaussian. For the other directions, the body force is set to zero.

3 Numerical Method

As shown in the previous chapter, a system of non-linear partial differential equations (PDEs) needs to be solved. To approximate the solution of the problem, a numerical method is used. Theoretical aspects on the numerical methods that are used can be found in [28, 29, 30].

The three classical options for the discretisation of PDEs are: Finite Difference (FDM), Finite Volume (FVM) and Finite Elements methods (FEM). All use finite shaped elements, but all in a different manner. The FDM is an efficient solving method for regular grids. Because of this, rectangular or block shaped elements are used when working with FDM. However, FEM and FVM both use geometrically simple shaped elements instead. Since an embryo is not regularly shaped, FEM and FVM are preferred and the FDM is not considered currently as an option.

A main difference between FVM and FEM, is that FVM is based on local conservation, whereas the FEM only guarantees global conservation. In case of the FVM, this means that whatever goes into a cell or an element on one boundary, should leave this element on another boundary.

For FEM all equations are weighted by a test function ψ and averaged by integrating over the domain. The finally obtained weak form should hold for all test functions $\psi = \psi_h$. In general, FVM is just a special case of FEM, where the test function $\psi = 1$ is chosen and no integration by parts is performed to enlarge the class of possible weak solutions regarding the integrability requirements. Next to that, this corresponding weak form should then hold for all control volumes $\Omega = \Omega_h$.

In the end, it is chosen to work with FEM. This method will shortly be explained in the next paragraph. Furthermore, the application of FEM to the problem described in Section 2 will be described. Afterwards, the engineering platform COMSOL Multiphysics®, which will be used for solving the multidimensional problem, is explained. This platform is created for solving multiphysics problems with the finite element method. This leads to an extra motivation to use FEM, since this allows comparing the results with the certainty that the choice of the numerical method does not contribute to any potential differences.

A property of a multiphysics problem is the existence of coupled partial differential systems in your problem. Accordingly, COMSOL Multiphysics® could be used for the problem stated in this thesis.

3.1 Finite Element Approach

Working with the finite element method gives the user a broad range of choices for modifying the solver. First, a finite element mesh with geometrically simple finite shaped elements is created. Generally, it uses tetrahedral and brick-shaped elements in three dimensions, or triangles and quadrilateral elements in two dimensions. An advantage of FEM is the use of a finer mesh at specific areas of the geometry, such that at, e.g., geometric transitions the solution can accurately be approximated and gradients too.

The next step is to choose what type of basis functions will be needed. For a simple approach one could use linear elements, while for tracking stress over an element, parabolic elements could be more appropriate.

Then, the PDE is mathematically rewritten in a weak form and further elaborated with the use of the basis functions. This results in the so called Galerkin weak form. This can be solved element-wisely.

This will result in a typical sparse matrix for numerical problems, which can be solved with a direct or iterative method.

Below it is described how the finite element method is used for the one dimensional problem given in Section 2.1.

Moving Mesh

The first thing to do when using a numerical method is to define the grid. In one dimension, one can choose between an equidistant grid and a non-equidistant grid. For ease, the first one is chosen as initial mesh for this research. The mesh then looks as in Figure 3.1.

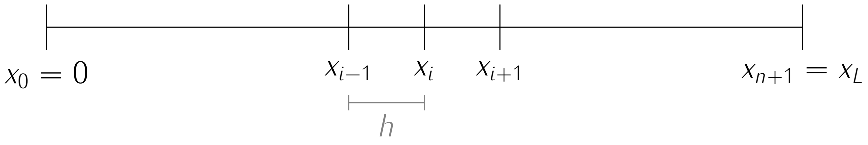


Figure 3.1: One dimensional equidistant mesh

This mesh has a constant node distance h which is defined as $h = x_i - x_{i-1}$.

This type of mesh is easy in use, although the non linearity in the PDEs will be hard to calculate. For this, there is simple modification to the mesh to deal with the non linearity, which is a moving mesh.

A moving mesh is, as it says, a mesh which moves along the results over time. Every time step a new mesh corresponding to the new solution is created. This also means that it is possible that the grid will not be equidistant over time. This type of mesh is often used in fluid dynamics, but also in solids which deform largely. Since the embryo will grow largely as well, this is a second reason to use a moving mesh.

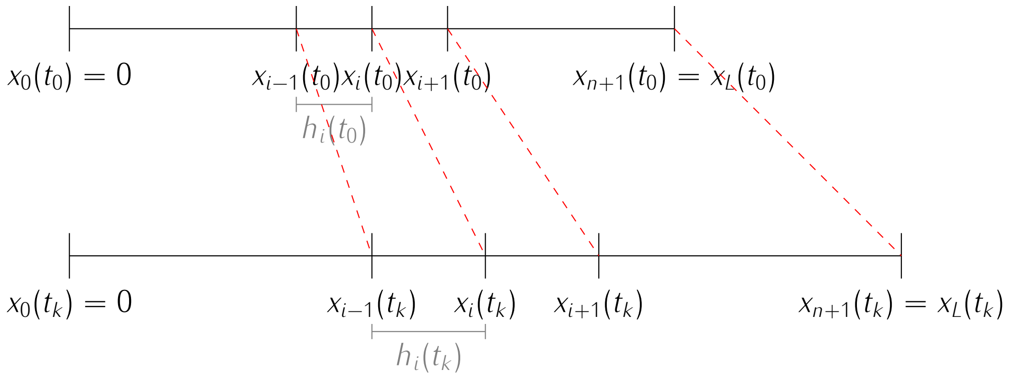


Figure 3.2: An illustration of how a mesh evolves over time when the amount of nodes is preserved. For this, the distance between two nodes also depends on which nodes you are looking at and thus the node distance varies over the mesh by $h_i(t) = x_i(t) - x_{i-1}(t)$.

The idea of a moving mesh is schematically given in Figure 3.2. Every time step the node distance is updated by

$$h_i(t_k) = h_i(t_{k-1}) + \Delta t(v_i(t_k) - v_{i-1}(t_k)). \tag{3.1}$$

with v_i the displacement velocity at node i. This value is available since it is the solution to the velocity PDE (given in ($P1$)). It is important to realise that the node distance h_i is given over an area $[x_{i-1}, x_i]$ where the displacement velocity v_i is given at a certain node i. Hence, the first node velocity needs to be subtracted from the second node velocity $(v_i(t_k) - v_{i-1}(t_k))$.

Basis Functions

Basis functions are functions used within an element to describe the behaviour of the solution. The simplest option is to use piecewise linear basis functions. The solution at both ends of the element are calculated. A linear line is then plotted in between those points to describe the solution behaviour within the element.

For the sake of simplicity it is chosen to use linear basis functions. In this case, the mathematical form of the basis function φ_i is given by

$$\varphi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{h_i} & x \in [x_{j-1}, x_j], \\ \dfrac{x_{j+1} - x}{h_{j+1}} & x \in [x_j, x_{j+1}], \\ 0 & \text{elsewhere.} \end{cases}$$

Here, the node distance h_i is given as in (3.1). For a visualisation, we refer to Figure 3.3.

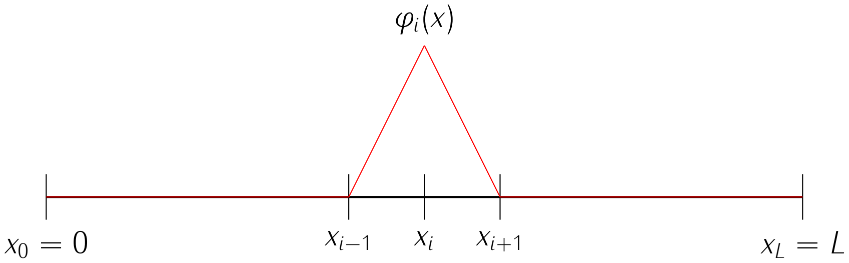


Figure 3.3: Linear basis function φ_i drawn in red.

Galerkin Weak Form

For solving the system of coupled differential equations, given in ($P1$), the Galerkin finite element approach is used. In this thesis the derivation is based on the approach of Boon et al. [31]. Therefore the PDEs are first multiplied by a test function ψ. The method is shown only for the displacement velocity PDE, the balance of momentum:

$$\rho\left(\frac{Dv}{Dt}+v\frac{\partial v}{\partial x}\right)-\frac{\partial\sigma}{\partial x}=F_b,\qquad x\in(x_0,x_L],\quad t\in(0,T].$$

The ε-PDE, the evolution equation for the infinitesimal strain (tensor), is treated analogously. When deriving the velocity PDE into its Galerkin weak form, we assume that the infinitesimal strain (tensor) ε is known and vice versa. This means, a segregated approach in solving the system will be used.

The test function will be taken from a standard Sobolev space; $\psi \in H^1(I_t)$, where $I_t = [x_0, x_L(t)]$ is the domain interval. Subsequently, the equation is integrated over this domain, which gices

$$\int_{I_t}\psi\rho\left(\frac{Dv}{Dt}+v\frac{\partial v}{\partial x}\right)\,dx-\int_{I_t}\psi\frac{\partial\sigma}{\partial x}\,dx=\int_{I_t}\psi F_b\,dx.$$

Next, the Cauchy stress (tensor) σ is partially integrated over the domain. With the knowledge that $\psi(0) = 0$ and from the second boundary condition that $\sigma(x_L) = F$, the external force, one gets

$$\int_{I_t}\psi\rho\left(\frac{Dv}{Dt}+v\frac{\partial v}{\partial x}\right)\,dx+\int_{I_t}\mu\frac{\partial\psi}{\partial x}\frac{\partial v}{\partial x}\,dx=\int_{I_t}\psi F_b-E\varepsilon\frac{\partial\psi}{\partial x}\,dx+\psi(x_L(t))F.$$

As a next step, the product rule for the total derivative,

$$\frac{D\psi v}{Dt}=\psi\frac{Dv}{Dt}+\frac{D\psi}{Dt}v,$$

is used to obtain

$$\int_{I_t} \rho \frac{\partial \psi v}{\partial t} + \rho \frac{\partial}{\partial x}[v \psi v] - \rho \frac{D\psi}{Dt} v \, dx + \int_{I_t} \mu \frac{\partial \psi}{\partial x} \frac{\partial v}{\partial x} \, dx = \int_{I_t} \psi F_b - E \varepsilon \frac{\partial \psi}{\partial x} \, dx + \psi(x_L(t)) F.$$

The first total derivative is written out now.

Applying Gauss theorem (that is the fundamental theorem of calculus in one dimension), results into

$$\int_{I_t} \rho \frac{\partial \psi v}{\partial t} - \rho \frac{D\psi}{Dt} v \, dx + \rho [v \psi v]_{\partial I_t} + \int_{I_t} \mu \frac{\partial \psi}{\partial x} \frac{\partial v}{\partial x} \, dx = \int_{I_t} \psi F_b - E \varepsilon \frac{\partial \psi}{\partial x} \, dx + \psi(x_L(t)) F,$$

with $[v \psi v]_{\partial I_t}$ the boundary integral in one dimension.

As last, the transport theorem of Reynold (or Leibniz' Rule in one dimension) is aplied to the first term on the left hand side.

$$\rho \frac{d}{dt} \int_{I_t} \psi v dx - \rho \int_{I_t} \frac{D\psi}{Dt} v \, dx \qquad + \int_{I_t} \mu \frac{\partial \psi}{\partial x} \frac{\partial v}{\partial x} \, dx = \int_{I_t} \psi F_b - E \varepsilon \frac{\partial \psi}{\partial x} \, dx + \psi(x_L(t)) F$$

With this, the Galerkin weak form, which is the element-wise form of the above, can be formulated as follows:

Find $v_h \in H^1(I_t)$ such that $\forall \psi_h \in H^1(I_t)$

$$\rho \frac{d}{dt} \int_{I_t} \psi_h v_h dx - \rho \int_{I_t} \frac{D\psi_h}{Dt} v_h \, dx + \int_{I_t} \mu \frac{\partial \psi_h}{\partial x} \frac{\partial v_h}{\partial x} \, dx = \int_{I_t} \psi_h F_b - E \varepsilon \frac{\partial \psi_h}{\partial x} \, dx + \psi(x_L(t)) F. \tag{G_v}$$

Similarly, we obtain the Galerkin weak form for the evolution equation for the infinitesimal strain (tensor):

Find $\varepsilon_h \in H^1(I_t)$ such that $\forall \psi_h \in H^1(I_t)$

$$\frac{d}{dt} \int_{I_t} \psi_h \varepsilon_h \, dx - \int_{I_t} \frac{D\psi_h}{Dt} \varepsilon \, dx = \int_{I_t} \psi_h (\frac{\partial v}{\partial x} - G) \, dx, \tag{G_ε}$$

where ψ, the test function, is from the same test space as described above. In this case, the displacement velocity v is set to be known.

For the cell concentration PDE, the cell concentration c on the right hand side will be taken from the last time step. In this way, the cell concentration c is known for that part of the equation. No extra iterations because of non-linearity are then needed. The Galerkin weak form for the cell concentration c is given below.

Find $c_h \in H^1(I_t)$ such that $\forall \psi_h \in H^1(I_t)$

$$\frac{d}{dt} \int_{I_t} \psi_h c_h \, dx - \int_{I_t} \frac{D\psi_h}{Dt} c_h \, dx + \int_{I_t} D \frac{\partial \psi_h}{\partial x} \frac{\partial c_h}{\partial x} \, dx = \int_{I_t} \psi_h \beta c_h^{t-1} (1 - c_h^{t-1}) \, dx \tag{G_c}$$

Linear system

We continue with the Galerkin weak form for the displacement velocity v, (G_v). The finite element variables are rewritten with the basis function φ_i.

i) $v_h = \sum_{j=1}^{n+1} \xi_j(t)\varphi_j(x(t))$, with ξ the approximation of v_h and $v(t, 0) = 0$.

ii) $\psi_h = \varphi_j(x(t)), \quad i = 1, 2, \dots, n+1$

These are filled in into the Galerkin weak form (G_v). Then the transport property of the basis functions ($\frac{D\varphi_i}{Dt} = 0$)[32] is applied.

$$\frac{d}{dt}\sum_{j=1}^{n+1} \xi_j \rho \underbrace{\int_{I_t} \varphi_i \varphi_j \, dx}_{M_v} + \sum_{j=1}^{n+1} \xi_j \mu \underbrace{\int_{I_t} \varphi_i' \varphi_j' \, dx}_{A} = \underbrace{\int_{I_t} \varphi_i F_b - E\varepsilon\varphi_i' \, dx + \phi_i(x_L(t))F}_{\mathbf{b}_v}$$

With this we get the time dependent system for the displacement velocity

$$\frac{d}{dt}(M_v \xi) + A\xi = \mathbf{b}_v$$

which can be solved in MATLAB with a time integration method. In this thesis, the Euler Backwards method is used. This method has been chosen because of its stability.

A similar system can be found for the infinitesimal strain (tensor) ε and the cell concentration c. The Galerkin Weak form of all the PDEs and the composition of the matrices can be found in Appendix F. In this appendix, there is also a brief elaboration on the idea of time integration in this setting.

Remark: One can note that in both cases, solving the displacement velocity v and solving the infinitesimal strain (tensor) ε, one needs a mass matrix (M) of the new time step, i.e. t_{k+1}, while one only knows the results of $t_0, t_1, \dots, t_k$. This means that the node distance $h_i(t_{k+1})$, at the new time step is also unknown (see Equation (3.1) for its construction). To solve this problem, an additional iteration step has been applied to first solve the displacement velocity v PDE. This means that there will first be iterated over solving this part of the problem, to find an acceptable value for the node distance vector *h. For this, the l_2-norm $||\xi_{t_k}^{n-1} - \xi_{t_k}^{n}||$, with $\xi_{t_k}^{n}$ the approximation for v_h at time t_k at iteration step n to find h_{t_k}, should not be greater than 10^{-12}.*

3.2 COMSOL Multiphysics®

According to the official COMSOL Multiphysics® website [33]: "COMSOL Multiphysics® is a general-purpose simulation software for modelling designs, devices, and processes in all fields of engineering, manufacturing, and scientific research". At the core, COMSOL Multiphysics® is able to setup and solve arbitrary coupled second order PDEs. These can be in a combined coefficient form, general form or weak form.

For each of the built-in physics, a dedicated set of (weak form) PDEs is ready for use. After building or loading the geometry, the end user just has to provide the required material coefficients and has to choose from a set of predefined boundary conditions. This implicates that the software can be used off the shelf for engineering problems where there is no need for mathematical intervention.

For scientific research, the full potential of COMSOL Multiphysics® becomes clear because all predefined PDEs are accessible and can be expanded by the end user. Furthermore, there are dedicated mathematics interfaces to setup user defined PDEs, ODEs, functions, variables and coupling operators.

As a typical example in scope of this research, COMSOL Multiphysics® has built-in viscoelastic models in the *Structural Mechanics* interface, such as the Maxwell model. After filling in the Prony series coefficients, an engineer can compute the relaxation behaviour for an arbitrary structure subject to stresses.

Since viscoelasticity is a special case of a morphoelastic model (see also Appendix C), and since the underlying equations are accessible, the software allows us to expand the viscoelastic equations to account for morphoelasticity by adding terms to the PDE. Instead, it is also possible to start from scratch and setup our own set of morphoelastic PDEs.

The built-in viscoelastic equations in this version of COMSOL Multiphysics® are displacement based. However, the equations used in this research are worked out for the displacement velocity instead. Therefore, it was not possible to adjust the built-in viscoelastic equations to the given morphoelastic equations given in Equation ($P1$) and ($P2$). Therefore, a user defined set of (displacement velocity based) PDEs has been used. Nonetheless, this gave more flexibility with the domain. More on this user defined set of PDEs can be found in the next paragraph.

Adding a PDE to COMSOL Multiphysics®

First of all, COMSOL Multiphysics® does have a lot of in-built physics. The easiest, and safest option is to use one of those when they exist for your problem.

The equations in this thesis are not implemented as a predefined physics interface. Fortunately, COMSOL Multiphysics® offers the possibilities to set up your own PDEs via, e.g., the *Coefficient Form PDE* of the mathematical environment

This physics gives a general equation,

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f,$$

which can be modified to the preferred PDE by modifying the coefficients.

coefficient	v-PDE	ε-PDE	u-PDE
e_a	0	0	0
d_a	ρ	1	1
c	μ	0	0
α	0	0	0
γ	$-E * \varepsilon$	0	0
β	$\rho * 2 * v$	v	v
a	0	0	0
f	F_b	$(1 - \varepsilon) * v_x - G$	v

Table 3.1: list of coefficient values in the one dimensional *Coefficient Form PDE* of COMSOL Multiphysics® corresponding to the one dimensional PDEs ($P1$).

To fill in the coefficients, it is important to realise that COMSOL Multiphysics® *would like to get everything written out. For example, every matrix multiplication (e.g. $A \cdot B$) will have to be done by hand before it can be put in the coefficients. In other words, the coefficients of the PDE only except scalars, e.g.* $a_{11} \cdot b_{11} + a_{12} \cdot b_21 + a_{13} \cdot b_{31} \ldots$.

In the one dimensional case the coefficients are set as in Table 3.1. When working in three dimensions the formulas and with that the coefficients of the COMSOL Multiphysics® *Coeffiecient Form PDE* get more complex. To get a better view on the correctness of the coefficients, the COMSOL Multiphysics® LiveLink™ for MATLAB has been used. More about this can be found in Appendix G.

To specify what happens on the boundaries, one needs boundary conditions. Those exist of two main types: Dirichlet and Neumann. Further, there are mixtures of those two. Nonetheless, COMSOL Multiphysics® has some more frequently used boundary conditions implemented. They are all derived from those two main types.

In COMSOL Multiphysics®, by default a *zero flux* boundary condition is added to all the boundaries of a *physics*. This means that the bounds are free to move and a solution to these boundaries should be found by the use of the PDE, which is basically a zero Neumann condition.

In the problem statement, two boundary conditions are given: a Dirichlet bound on the fixed side and a Neumann bound on the pulling side. In COMSOL Multiphysics® it is chosen to use a *constraint* to fix one boundary and a *flux/source* to apply the external force F on the other, pulling, boundary. These two are both implemented under the *Coefficient Form PDE* for the displacement velocity v. For the other four boundaries and all other PDEs it is sufficient to just use the *zero flux* condition.

Note that in the three dimensional case, more boundaries (surfaces) are available.

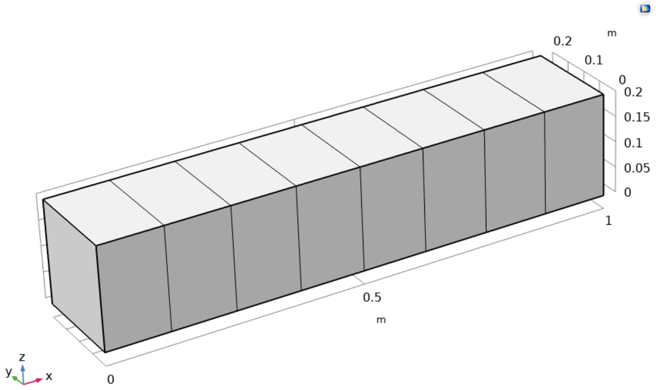


Figure 3.4: Visualisation of a swept mesh with bricks located entirely over the smallest surface. The mesh is made in COMSOL Multiphysics®.

For the remaining surfaces in the v-PDE (in (P2)), it is sufficient to keep the zero flux bound there. Although, it might give a solution in which the bar will move in the y- and/or z-direction instead of only grow into the x-direction. It seems to happen only when the mesh is not chosen properly. Therefore, a recommendation for when working with the bar problem, is to use brick-shaped elements instead of tetrahedrals. Preferably, use bricks which are located entirely over the smallest surface (see Figure 3.4).

Next to boundary conditions, the problem also needs to be initialised to be well defined. The COMSOL Multiphysics® *Coefficient Form PDE* is given in its standard form, with a second time derivative term. Therefore two initial conditions (its real value and the value for the first time derivative when $t = 0$) are asked for. The PDEs are initialised according to the default by COMSOL Multiphysics® by setting both equal to zero. Looking at our problem statement (in Section 2), this is exactly what is wanted. Unless the cell properties are added to the system. This PDE needs a stimulus from the start. Otherwise no growth will enter the system. Therefore, the first initial condition ($c(.,0)$) for this PDE should be modified.

Adding a (Moving) Mesh in COMSOL Multiphysics®

Creating a mesh in COMSOL Multiphysics® is easy at first sight. You get a standard tetrahedral mesh when nothing is changed to the mesh settings. An important option is that one can choose how fine the mesh should be. In general, a finer mesh would give a more accurate result, while it takes longer to solve.

Another nice feature in the mesh settings is that one can tell COMSOL Multiphysics® that you would like to have a finer mesh at some area and broader it somewhere else.

For the one dimensional rod problem, a coarse mesh works fine and fast. In three dimensions this works too, although you need to be careful in selecting the right kind of mesh. Based on experiments, longitudinal *swept* quadrilaterals are recommended, since the movement of the rod should only be in one direction.

When adding the cell properties, one changes the setting of the problem. The initial length will be much shorter (approximately only a few millimetres) and the final displacement should be quite large compared to its initial state (approximately fifty centimetres). For this reason a finer mesh is chosen. Further, it is chosen to have an even *finer* mesh near $x = 0$, since there you will have something like a source of cells, because of the chosen initial condition and PDE solution which are Gaussians. With this mesh, the necessary spatial derivative will be presented better.

As explained in the previous subsection, a moving mesh will be used. This is done, because the object, which is a representation of an embryo, should grow (enormously) over time. In COMSOL Multiphysics® there is an option to use a moving mesh. In the one dimensional setting, one can find this in the *Physics*. Otherwise, one can right click on *Definitions*.

It is necessary to define how the mesh moves. The given problem exists of two main equations: one for the displacement velocity v and one for the infinitesimal strain (tensor) ε. Next to those, there is a DE for the displacement u. The first two need to be solved together, since they are coupled. The third one stands on its own and can be determined by their results. This last PDE, however, gives the displacement u over time t. This can be used in the moving mesh settings, resulting that this PDE should also be solved in the same iteration loop as the two coupled PDEs. For this, a *Prescribed Deformation* with, u_1, u_2, u_3 in X, Y, and Z direction respectively is used.

When a mesh moves, its quality could deteriorate. This mostly happens when (a part of) one element moves faster then another. To keep the quality of the mesh, one could chose to remesh once in a while. When working with the cell properties, it is chosen to use this option in COMSOL Multiphysics®, because the total growth is quite large. This option can be chosen in the *Time-Dependent Solver* of the *Study*. As remesh criterion the *distortion criterion* is used. Different values of this criterion has been tried. When working with the Gaussian to represent the cell concentration of the embryo, COMSOL Multiphysics® will remesh when the distortion criterion exceeds 0.5. The amount of mesh elements after remeshing is the same as before.

Next to the distortion criteria to remesh, COMSOL Multiphysics® *also contains a quality criteria for the remeshing. Although this works fine for the tetrahedral meshes, it does not for the swept brick meshes.*

4 Results

In this section the results of this research are shown. First, the results of the one dimensional pulling rod problem, described in Section 2.1, are given. These results from the morphoelastic model are verified by using the viscoelastic model and by setting the growth tensor to zero (G= 0 in ($P1$)) in the morphoelastic model. This one dimensional model is implemented in MATLAB and filled in into COMSOL Multiphysics®. Therefore, both models are also verified by comparing the results of both programs.

To see if the growth tensor is implemented correctly, the model is also verified by a manufactured solution.

Secondly, the results of the three dimensional cases are given. For this, the pulling bar problem (Section 2.2) will be compared to its one dimensional counterpart, the pulling rod problem. Furthermore, the growing rod will be compared to the growing bar. In this setting, there is a non-zero growth tensor and an external force.

Finally, the results of the growing body, with an internal stimulus, will be briefly discussed, in both the one and the three dimensional setting. All equations can be found in Appendix C.

4.1 One-Dimensional Problem

In the one dimensional setting, the pulling rod problem is examined first. This problem is given in Section 2.1, and its schematic view in Figure 2.1.

For most of the one dimensional calculations the parameter values given in Table 4.1 are used, unless stated otherwise. When an external force F is used in the setting, the force is applied during the time interval of $t \in (0, T/2)$. Then, at $T = T/2$, the rod is loosened at once. However, this could give a non-differentiable point. It would be more realistic to loosen the rod smoothly, by for example using an S-shape function around that time.

The value of the Young's modulus has been chosen to be one GPa, as it is a standard unit for this. The shear modulus and mass density have been chosen such that their effects are visible, so they are of the same order as the Young's modulus. The viscosity gives a damping to the system, while the mass density adds a delay. This results into a new equilibrium (while pulling continues) that gets reached after a while instead of immediately, as is the case for linear elasticity. This may result into a little overshoot while reaching this equilibrium. The behaviour starts to resemble linear elasticity more when the shear modulus approaches zero, see also Appendix H. In this same appendix, the effect of the mass density is shown too.

In the first problems, only the situations without any growth and body force are examined. For these situations, a stability check has been performed. This check can be found in Appendix E. It turns out that in the situations of zero displacement velocity and constant infinitesimal strain (tensor), the system is stable. This is when the maximum displacement has been reached while continuing pulling the rod. The rod will not change size.

	symbol	value	unit
	x_0	0	m
	x_L	1	m
node distance	h	0.1	m
time step	Δt	0.001	s
end time	T	20	s
Young's modulus	E	1e9	Pa
shear modulus	μ	1e9	Pa·s
mass density	ρ	1e9	kg/m^3
body force	F_b	0	N/m
external force	F	1e6	N/m
growth tensor	G	0	1/s
	$u(0, t)$	0	m
	$u(x, 0)$	0	m
	$v(0, t)$	0	m/s
	$v(x, 0)$	0	m/s

Table 4.1: List of values used in the one dimensional problems, unless stated different. The variable h, is the initial equidistant node distance.

4.1.1 Pulling Rod Problem

In Figure 4.1, the schematic visualisation of the pulling rod problem is given.

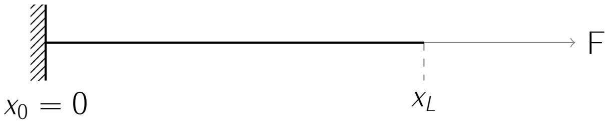
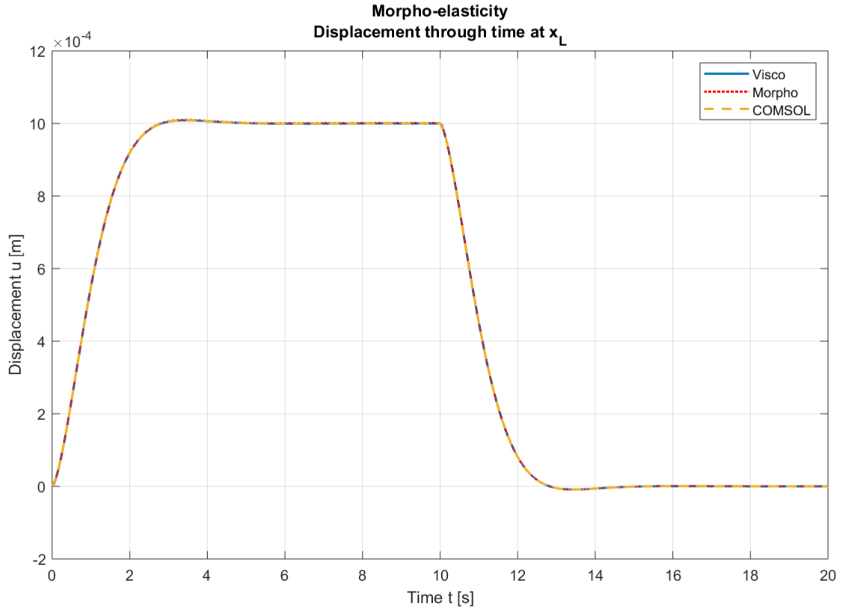
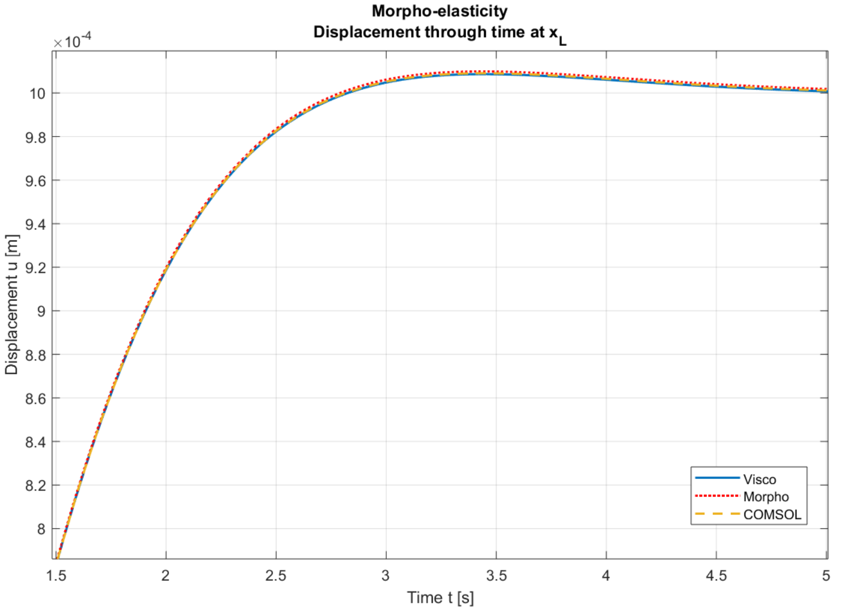


Figure 4.1: The pulling rod problem. In one dimension, the rod is hold on the left side, where $x = 0m$. On the other side, $x = x_l$, an external force F is applied.

The coefficients of the PDE corresponding to this problem ($P1$), are given in Table 4.1. The results to this problem are given in Figure 4.2. This figure shows the displacement u of x_L over time t. For this, three solutions are plotted together: viscoelasticity (*Visco*), morphoelasticity by MATLAB (*Morpho*), and morphoelasticity by COMSOL Multiphysics® (*COMSOL*). Those three curves are plotted together to show that they give similar results. For this the growth tensor G is set equal to zero in the morphoelastic models. It can be seen that the morphoelastic result reproduces the viscoelastic behaviour if the growth tensor is zero.



(a) Full result



(b) Zoomed result, $t \in [1.5, 5]s$

Figure 4.2: Morphoelastic results with a zero growth tensor (G= 0) of both MATLAB (dark blue) and COMSOL Multiphysics® (yellow) compared to viscoelastic results (light blue). The lines are drawn for $x = x_L$.

The viscoelastic result (*Visco*) is used as verification, and verified by comparing it to the linear elastic results, see Appendix H through variation of the damping term μ. It is used as a verification to verify the results of the morphoelastic results of both the MATLAB script (*Morpho*) and the COMSOL Multiphysics® simulation (*COMSOL*). Both make use of a moving mesh.

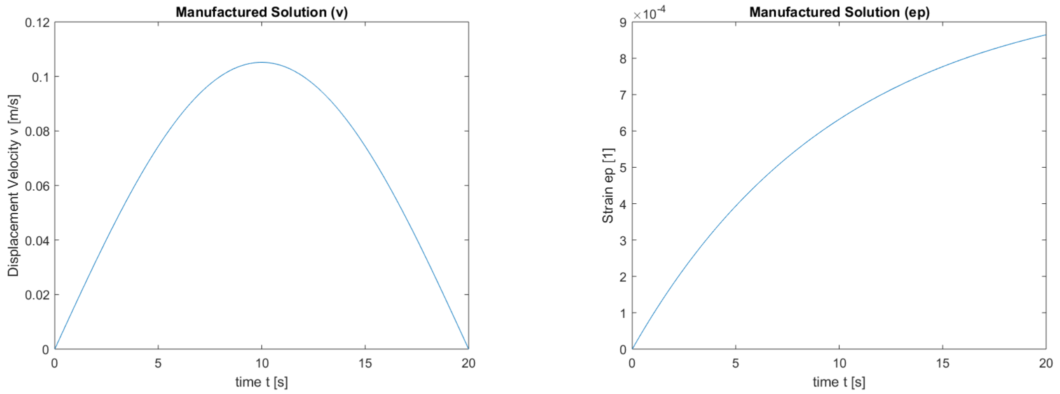
In addition, the COMSOL Multiphysics® program is modified such that the calculation will most likely be the same. To this extent, the time steps taken are fixed on 0.001 s, as in Table 4.1. This can be done in the *Time-dependent solver* by setting the *steps taken by solver* to *manual*. The inter node distance has been set to 0.1 m by putting this as the maximum element size. Importantly, the Euler Backwards method has been used to deal with the time derivatives. Therefore, COMSOL Multiphysics® is forced to use this method too. This is done by using the *BDF* time stepping method with exactly the order 1.

All three results are basically on the same line, with a little deviation at the corner where the force equilibrium gets reached. A zoomed figure is given around this time, Figure 4.2b.

In Appendix H, these same results are shown for two different values for the time step. To see the positions of the curves better, one result is added in every subsequent subfigure.

This shows us that the results are reliable, especially when taking a small time step or when you work in COMSOL Multiphysics® with a moving mesh. This motivates us to use the moving mesh in COMSOL Multiphysics® in the next settings.

4.1.2 Verification by the Use of a Manufactured Solution



(a) Displacement velocity v

(b) Infinitesimal strain (tensor) ε

Figure 4.3: Manufactured solutions of the morphoelastic problem. The functions of equation (4.1) are plotted over time, for $x = 1m$.

The second method of verification used is the method of a manufactured solution. For this method, a solution for the displacement velocity v and the infinitesimal strain (tensor) ε is formulated and by using this solution, the parameters for the growth tensor G, body force F_b, and the external force F are constructed. Those parameter functions are then implemented in the model ($P1$) to determine v and ε again. This should match the created solutions.

It is chosen to use the functions

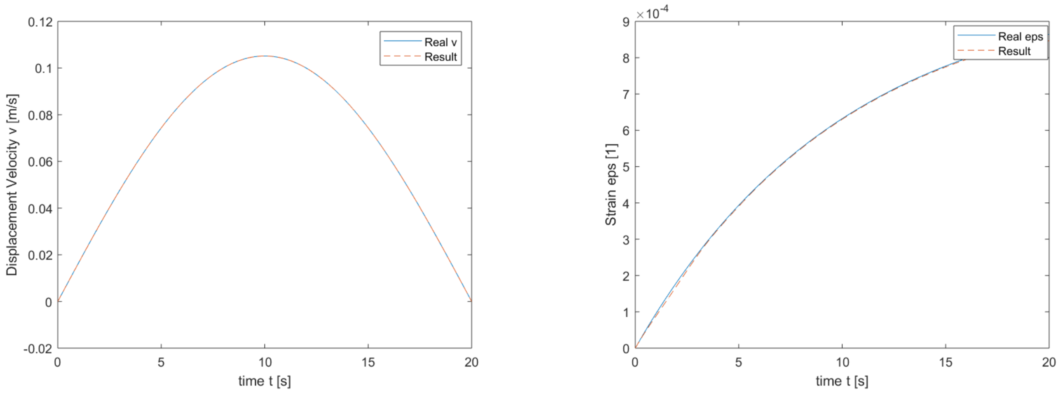
$$\begin{aligned} v(x,t) &= (e^{\alpha x^2} - 1)\sin \beta t; \\ \varepsilon(x,t) &= \varepsilon_0 \left(1 - e^{-\gamma t}\right) \end{aligned} \tag{4.1}$$

with the values

$$\begin{aligned} \alpha &= 0.1 \\ \beta &= 0.05\pi \\ \gamma &= 0.1 \\ \varepsilon_0 &= 0.001. \end{aligned}$$

This gives the curves shown in Figure 4.3.

These functions are implemented in the PDE ($P1$) to determine a function for G, F_b and F. This is done by using the symbolic environment of MATLAB, together with the derivative function `diff(.)`. The functions v and ε, (4.1), are created with the



(a) Displacement velocity v

(b) Infinitesimal strain (tensor) ε

Figure 4.4: The manufactured solution (blue) and the COMSOL Multiphysics® generated solution (red dashed) of the morphoelastic problem. The results and functions are plotted over a time interval of 20 seconds, and at the end point of the rod $x = 1m$.

symbols x, t. Then the parameters are determined as follows:

$$F_b = \rho \cdot \frac{\partial v}{\partial t} + 2\, v\, \frac{\partial^2 v}{\partial x^2} - \frac{\partial \varepsilon}{\partial x};$$
$$G = -\left(\frac{\partial \varepsilon}{\partial t} + v\, \frac{\partial \varepsilon}{\partial x} + (\varepsilon - 1)\frac{\partial v}{\partial x} \right);$$
$$F = \mu\, \frac{\partial v}{\partial x} + E \cdot \varepsilon.$$

The MATLAB-output is thereafter plugged in for the corresponding *variable* in COMSOL Multiphysics®.

For this, only the model in COMSOL Multiphysics® has been used. This is since in the previous subsection, the results of MATLAB and COMSOL Multiphysics® were close to each other. Also, COMSOL Multiphysics® will be used in three dimensional setting, where MATLAB will not.

Figure 4.4 shows the COMSOL Multiphysics® results together with the formulation of v or ε. In this figure one can see that the lines are close to, or even on each other. With this we conclude that the model has been implemented correctly.

4.1.3 The Effect of Two Different Growth Terms

In the end, we would like to have an autonomously growing body. This means that its starting point will never be reached again, since the body will grow, instead of (only) get stretched.

To first see the effect of the growth tensor G, a constant growth tensor has been chosen. A second choice is a growth tensor depending on the infinitesimal strain (tensor) ε. This has been chosen since it is also used by Koppenol [11] to describe the growth tensor.

Note that an external force is still applied on the system to stimulate the growth, which is especially needed in the second case.

Linear Growth Finally, a growth tensor will be added to the system. To this end, first a constant growth tensor has been chosen of $10^{-5}\ s^{-1}$. With a constant growth tensor, one would expect a linear increase in displacement u. In Figure 4.5 one can see that this indeed happens.

Strain Dependent Growth Another case is to look at a strain dependent growth tensor. This has also been done by Koppenol [11]. Here, it is chosen to use $G = \alpha\varepsilon$, with $\alpha = 10^{-2}\ s^{-1}$ as the morphoelastic change (i.e., the rate at which the effective strain changes actively over time)[11]. With this, one would expect to reach a new equilibrium after a while. This will briefly be explained.

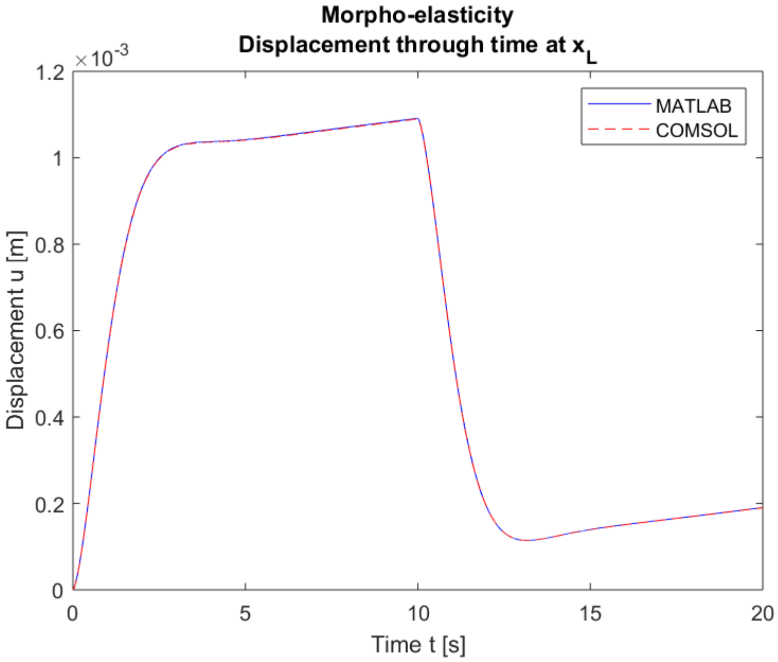


Figure 4.5: Morphoelastic results of the pulling rod problem with a constant growth tensor in the rod. The line is drawn for $x = x_L$.

From the problem statement ($P1$) one can note that the infinitesimal strain (tensor) ε is linked to the Cauchy stress (tensor) σ, and therefore the external force F. When one loosens the rod, this force is equal to zero, and so there should be no stress after this point. This results into a displacement velocity v with an opposite sign of the infinitesimal strain (tensor). Logically, the infinitesimal strain (tensor) and displacement velocity should both get equal to zero. Then the growth rate is given by $10^{-2}\varepsilon$, as stated before. As a result, there is no current growth and an equilibrium has been reached.

The obtained results are shown in Figure 4.6. In this case the results of the MATLAB model agree with the results of the COMSOL Multiphysics® model. In both results, one can see that indeed a new equilibrium is found after about 15 seconds.

Interesting to remark is the fact that the rod keeps on growing between $5-10$ s. This is explainable in the same way the new equilibrium is explained. In this interval, an external force is applied on the rod. This provides stress in the end node, x_l. This means that the Cauchy stress (tensor) σ is non-zero. The displacement velocity v together with the infinitesimal strain (tensor) ε are therefore still present. Subsequently, the growth tensor G is non-zero. Thus, the infinitesimal strain (tensor) ε is still varying over time. This results into a small increase in displacement velocity and thus in the displacement, which shows the total growth of the system relative to

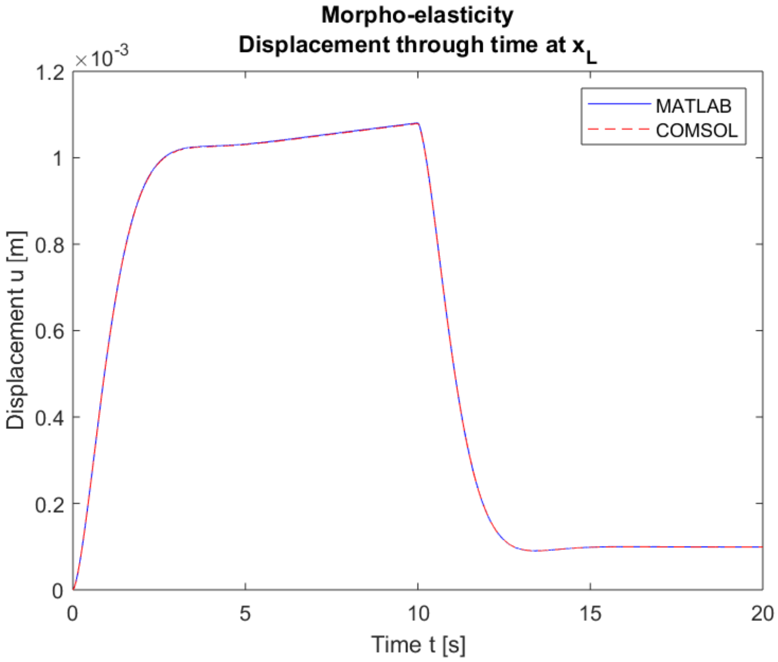


Figure 4.6: Results of the morphoelastic model of the pulling rod problem with an strain dependent growth tensor in the rod. The line is drawn for $x = x_L$

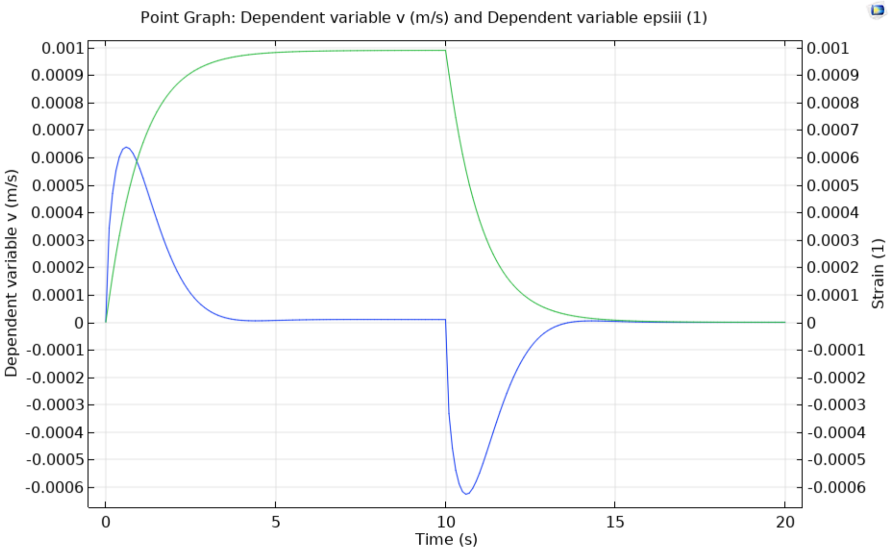


Figure 4.7: Infinitesimal strain (tensor) (green, epsi) and displacement velocity (blue, v) plotted over time for the growing rod problem with an infinitesimal strain (tensor) dependent growth tensor. For both variables, the end point of the rod, $x = x_l$, is plotted.

its initial state.

This added growth during the time interval that was mentioned earlier does not influence the Cauchy stress (tensor) and therefore it could reach a new equilibrium after switching off the applied external force F. The growth effect, while pulling and the maximum force has not been reached yet, is undone by the time the final equilibrium (after loosening the rod) is reached, because it has the same effect after releasing the rod.

The explanations above can also be illustrated by plotting the infinitesimal strain (tensor) and displacement velocity in one figure, see Figure 4.7.

4.1.4 Results with Added Cell Properties

A more realistic setting is to stimulate growth from the inside. For this reason an extra PDE (P_c) is added to the system. This PDE describes the behaviour of the cells, but can be seen as a description of the cell concentration.

For the results given in Figure 4.8, the physical parameter values given in Section 1.4 are used. The other four parameters are found by trial and error. The values are given in Table 4.2 and 4.3.

A *Gaussian Pulse* is used as initial condition. It has been given a standard devia-

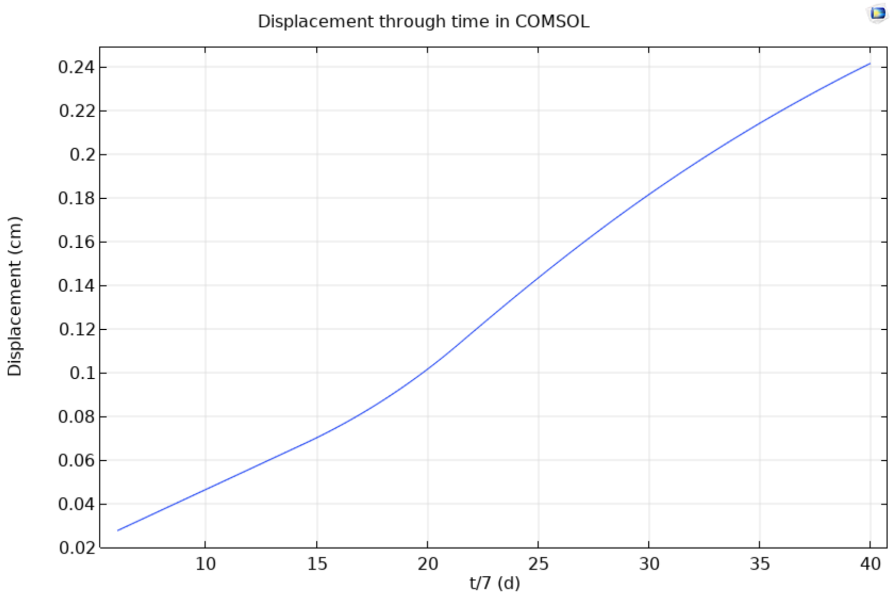


Figure 4.8: Results of the model of morphoelasticity with a cell concentration PDE growth stimulus. The curve shows the displacement of the end node of an initially $2mm$ long rod. Therefore, the results should be multiplied by 2 to get the total growth of the one dimensional embryo representation. To get the total length, the initial length should be added too.

Parameter	Value
k	10
D	0.0000123 $[cm^2/day]$
C_g	1 $[1/s]$

Table 4.2: Parameter values used for solving the problem with the morphoelastic model and cell concentration.

start (d)	end (d)	Function ($1/day$)
42	100	0.000013
100	150	0.013
150	300	0.00013

Table 4.3: The variation of the proliferation rate β over time

tion of $5 \cdot 10^{-7}$ and is divided by a factor 10^6 to keep the maximum cell concentration within the domain [0,1].

4.1.5 Results on the Alternative Idea for the Cell Concentration

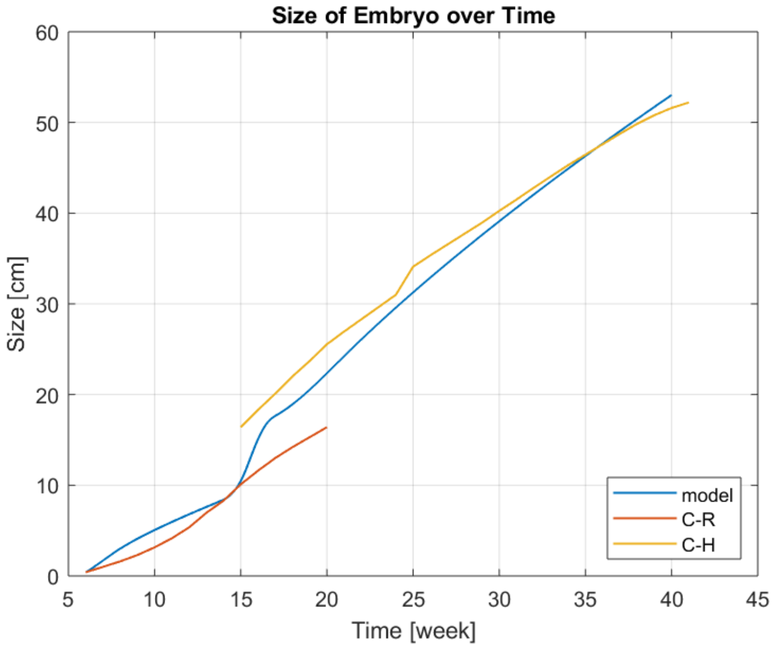


Figure 4.9: Total size over time of the initial $2mm$ long rod, representing the embryo in one dimension. It has a fixed end for symmetry, and a zero flux end. The growth is stimulated by the spatial derivative of a Gaussian which represents the cell concentration. The time t axis is in weeks.

As it was discussed in Section 2.3, an alternative idea will be investigated too. For this Equation (2.2) is used as stimulus. It has been shrunk by a factor 10 000 to get a curve below one, which can grow. The disadvantage of using this Gaussian is that its standard deviation does not change in time. With a chosen standard deviation of $5 \cdot 10^{-4}$ m, it starts as a nice curve on the initial length. This results into the behaviour that new cells will be created at the midpoint of the (symmetric) rod. Those cells will be pushed to the side and this will result in growth. However, it never reaches the maximum amount of cells in grown parts ($x > 2$ mm). The physical parameters have the value given in Section 1.4.

The cell proliferation rate β changes discretely over time. This has been done to be able to represent the lengths based on both the measurements methods that are used for measuring the length of an embryo (crown-to-rump and crown-to-heel). In this way, the curve will look more like the one in literature (Figure 1.2a). For the proliferation rate a *piecewise* function is used and filled in as in Table 4.4.

Secondly, the force constant k, see (2.1), changes over time in a similar manner as the proliferation rate. A *piecewise* function has been used as well for this with the values in Table 4.5.
Solving this system of PDEs resulted in the blue graph plotted in Figure 4.9. The figure shows two other lines. The red and yellow lines are the ones from literature and show the crown-to-rump length of the first weeks and the crown-to-heel length of the last twenty five weeks, respectively.

The discrete changes in proliferation rate β and force constant k result in a nod around week 15-16. This is about the time where the method of measurement is switched from crown-to-rump to crown-to-heel. At that moment it looks like the embryo has gained in size enormously in Figure 4.9, which just represents the change of measurement method.

In order to analyse the the accuracy of the model, the relative error is calculated. In the first fifteen weeks the model has a relative error of 0.2316. The next 25 weeks give a relative error of 0.0507.

4.2 Three-Dimensional Problem

A more realistic setting for a growing embryo is to work in three dimensions. To check the formula's and their implementation in COMSOL Multiphysics®, the results are compared to the one dimensional results. This can only be done on a similar situation.

start (d)	end (d)	Function ($1/day$)
6*7	110	0.013*0.005
110	300	0.013*.3

Table 4.4: The variation of the proliferation rate β over time.

start (d)	end (d)	Function
6*7	110	500
110	300	50

Table 4.5: The variation of the force constant k over time.

Method	absolute error	relative error
C-R	3.9026	0.2316
C-H	9.6499	0.0507

Table 4.6: The absolute and relative error of the model results. The errors are given for the weeks 6 – 15, where the crown-to-rump measurement is used and the second period, weeks 16 – 40, when the measurements are done between crown-to-heel.

Therefore, a pulling bar and growing bar setting have been examined.

After this has been verified, the geometry will be changed into a unit block and sphere. For those, the mesh type has been investigated.

Lastly, some words are given on the growing body without an external force, but a stimulus from inside the body, by using cell division phenomena.

For the comparison between the one and the three dimensional case the parameter values given in Table 4.7 and 4.8 are used.

It is chosen to use a mass density which is closer to reality, while working with the given Young's modulus. As it can be seen, the bulk viscosity μ_2 and Poisson ratio ν are both set to zero. This is chosen since in the pulling rod or bar problem, there should be no influence from the other directions than the pulling direction.

4.2.1 Verification by the Use of the One-Dimensional Results

As a first check on the generalised formulas of ($P2$), the results will be compared to their one dimensional counterpart.

For this, the pulling and growing rod problems are transformed into three dimensional pulling and growing bar problems. The schematic drawing is repeated in Figure 4.10.

The bar is placed with one edge on the x-axis, such that a fixed corner is on (0 0 0). The other edges are in the positive y-, and z-direction.

Name	Expression	Value	Description
μ	1e9 [Pa*s]	1E9 Pa·s	shear modulus
E	1e9 [Pa]	1E9 Pa	Young's modulus
ρ	8e3 [kg/m^3]	8000 kg/m^3	mass density
C_g	0 [1/s]	0 1/s	constant growth factor
F_b	0 [Pa/m]	0 N/m^3	body force
T	20 [s]	20 s	end time
Δt	0.1 [s]	0.1 s	time step

Table 4.7: Parameter list of the one dimensional pulling rod problem, for comparison to the three dimensional case.

Name	Expression	Value	Description
μ_1	1e9 [Pa*s]	1E9 Pa·s	shear viscosity
μ_2	0 [Pa*s]	0 Pa·s	bulk viscosity
E	1e9 [Pa]	1E9 Pa	Young's modulus
ν	0 [1]	0	Poisson ratio
ρ	8e3 [kg/m^3]	8000 kg/m^3	mass density
C_g	0 [1/s]	0 1/s	constant growth factor
F_b1	0 [Pa/m]	0 N/m^3	body force, x direction
F_b2	0 [Pa/m]	0 N/m^3	body force, y direction
F_b3	0 [Pa/m]	0 N/m^3	body force, z direction
T	20 [s]	20 s	end time
Δt	0.1 [s]	0.1 s	time step

Table 4.8: Parameter list of the three dimensional pulling bar problem.

In Figure 4.11, the results are presented. The red striped line shows the displacement-curve of the outermost upper right corner of the bar; (1 0.2 0.2)m. The curve is close on the one dimensional result (at $x = x_l$).

A three dimensional visualisation of the result is illustrated in Figure 4.12. The coloured slice shows how much that point x-directionally is moved from its starting point, blue meaning the least and red the most. The result is scaled by a factor of 100 to get a visible displacement. In Figure 4.11 one sees that the displacement u would otherwise be 1 mm, which is barely visible. From Figure 4.12, the one dimensional nature is clearly visible.

The bar has eventually also been rotated. It is placed on the y- and z-axis and also hovering in the space. For this scenario, the bar was rotated 45° around the z-axis, and 45° around the $\begin{pmatrix}1 & -1 & 0\end{pmatrix}$ line. Results for these rotations are given in Appendix I.

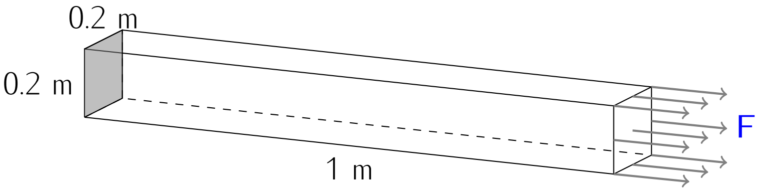


Figure 4.10: The pulling rod problem expanded into its 3D version: the pulling bar problem. The block has a length of 1 meter and its thickness is 0.2×0.2 m. One side (gray) is fixed by a zero constraint and on the opposite side an external force F is applied.

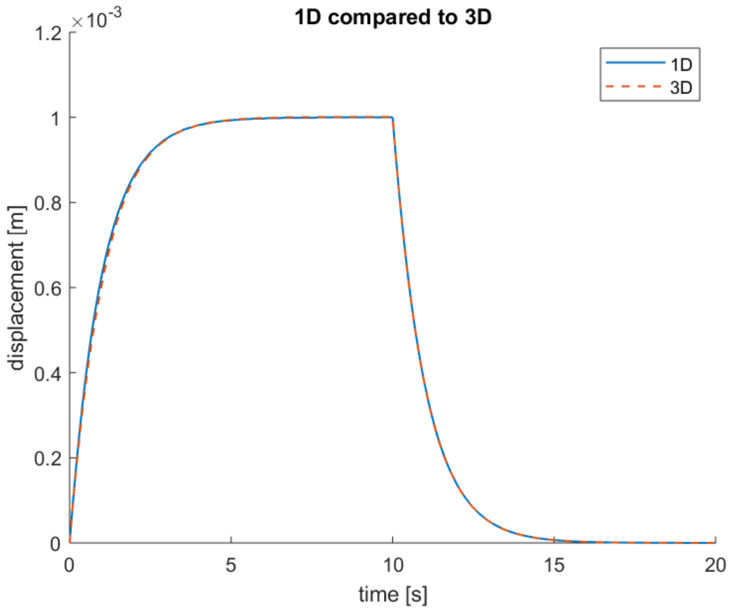


Figure 4.11: The displacement-curve of the one dimensional pulling rod problem (blue) compared to the one resulting from the three dimensional pulling bar problem (red, striped). The one dimensional result is drawn for $x = x_L$ and the three dimensional results for $x = (1\ 0.2\ 0.2)$. The bar is placed with one edge on the x-axis.

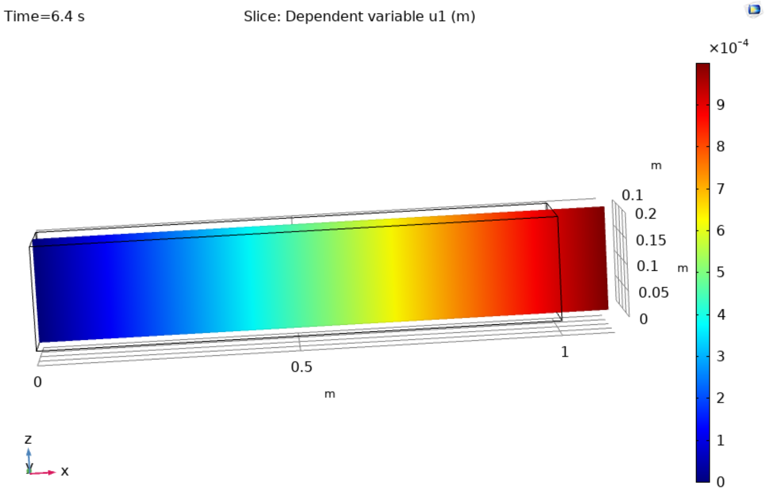


Figure 4.12: Visualisation of the pulled bar, created with COMSOL Multiphysics®. The coloured bar shows the displacement u_1 (x-directional displacement) of that point. The colour blue represents the least movement, while the colour red represents the most movement. It is plotted at $t = 6.4\ s$ in order to plot the maximum displacement. The scale is 100x.

4.2.2 Added Growth

In the section before, it is shown that the three dimensional problem is implemented correctly by comparing the results to the one dimensional case. Therefore, the effect of different kinds of growth factors and the influence of a mesh can be examined.

Growing Bar Again, a bar of which one edge is situated on the x-axis, with a corner of the fixed plane on (0 0 0), is examined. It is said to grow in x-direction only. Therefore, only G_{11} has been given a value. Also, the Poisson ratio and bulk viscosity are set equal to zero.

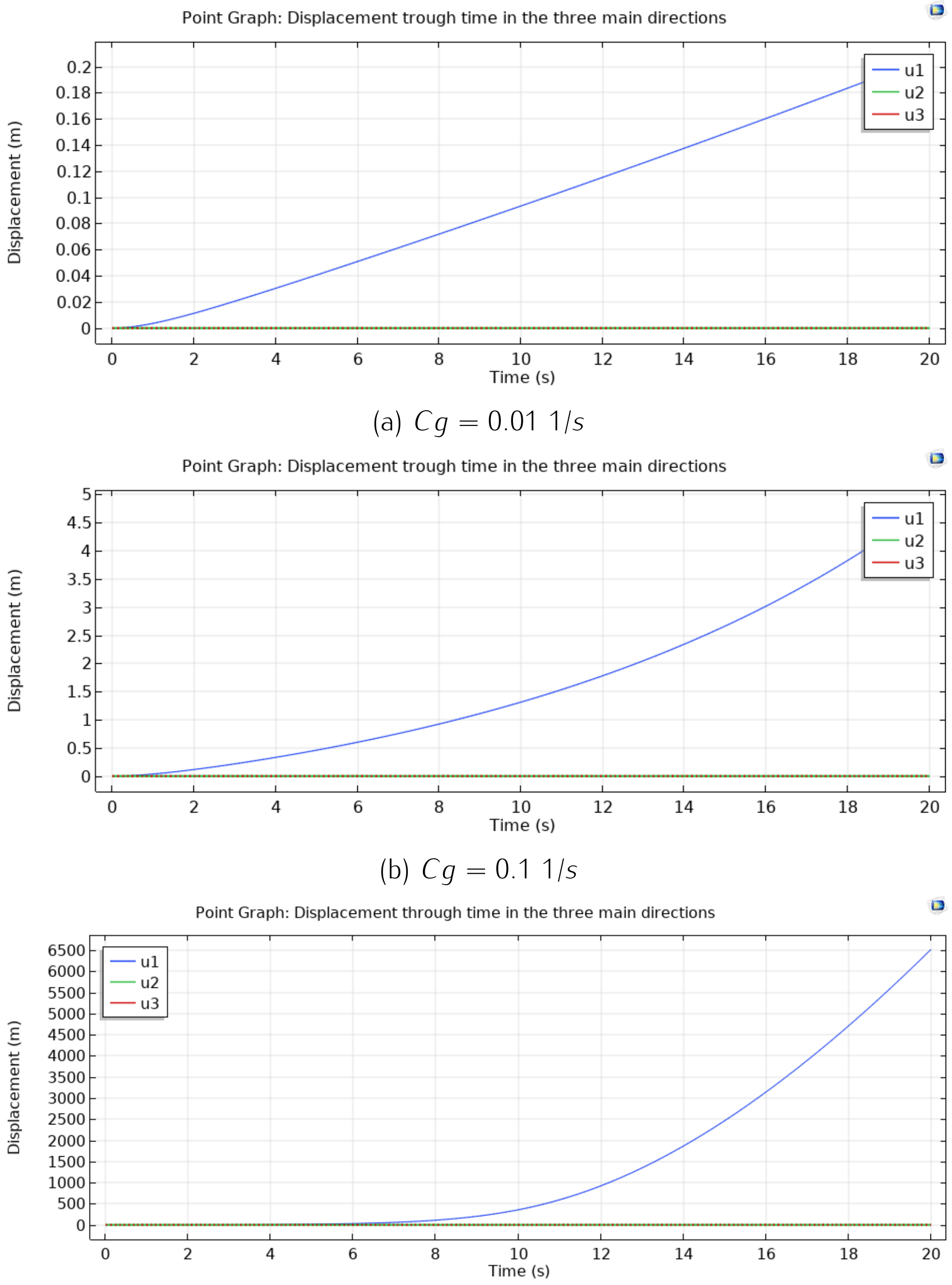
The setup has been tested in multiple directions as was done for the pulling bar situation. For this, both the linear growth as the strain dependent growth results in one dimension (Section 4.1.3) has been used for verification. About this, no further details are given.

Constant growth in the x-direction, $G_{11} = C_g$, is added to the system. Three cases are investigated:

1. $C_g = 0.01$ $1/s$
2. $C_g = 0.1$ $1/s$
3. $C_g = 1$ $1/s$

For all cases no external force has been put in the system. The results of these problems are given in Figure 4.13. One can notice that the growth and the displacement are mainly in the x-direction (u_1). However, in Appendix I one can note that for the extreme constant growth factor of 1 s^{-1} the bar eventually will bend. Though, in this time frame of 20 s the bending is negligible compared to the x-directional displacement.

The system of equations (P2), described in Section 2, is tested by extreme growth factors, e.g. $C_g = 1$ s^{-1}. The geometry of the bent solution used a swept mesh in the longitudinal direction. The mesh does not influence the results much. Whenever a tetrahedral mesh is used or when the mesh is swept over its width or height, the bending is much greater in the results. By using an appropriate scaling on the unknowns, this effect disappears. These bendings, however, could also be explained by the fact that the formulas are only defined for small displacements u. In this scenario, the bar grows 6 km.
Further is observed that the fully coupled solver is not able to solve for all these big constant growth factors. It automatically took extremely small time steps of about 10^{-5} and smaller. A segregated solver is therefore used instead.

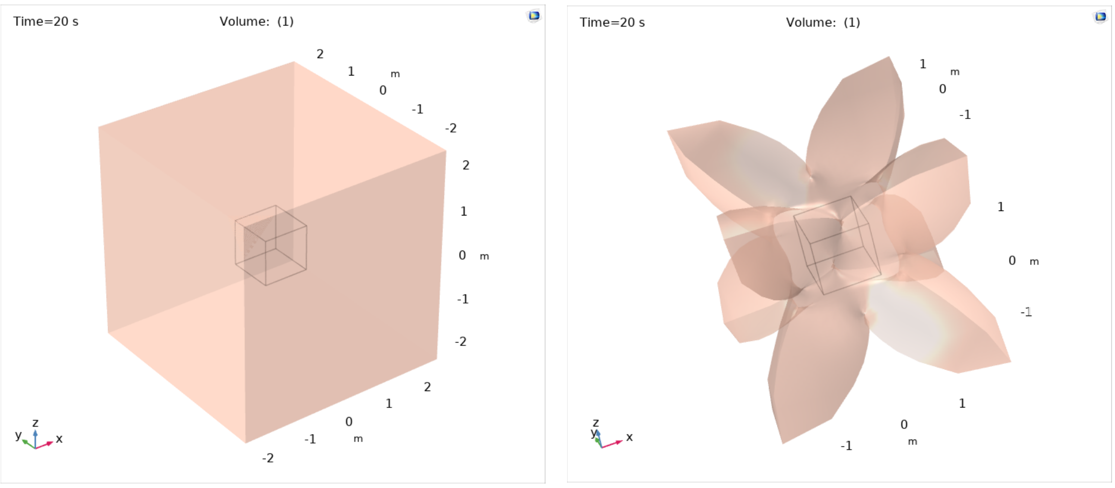


(a) $Cg = 0.01\ 1/s$

(b) $Cg = 0.1\ 1/s$

(c) $Cg = 1\ 1/s$

Figure 4.13: Displacement u through time for various values of C_g, the constant growth factor in x-direction. The displacement in the main directions is plotted in each figure.



(a) Constant growth

(b) Growth by norm

Figure 4.14: A visualisation of the three dimensional growing unit cube. In (a) it is grown by a constant growth factor of 0.1/s. In (b) the constant growth factor is given by the norm of **x**, in other words, the distance from a point in space to the origin.

Growing Block Gradually, a more realistic shape will be used. A unit cube (Figure 2.3a) will be grown symmetrically in all directions, stimulated by an external force. For this, the cube is centred at (0 0 0). This point is fixed with a zero-constraint on the displacement velocity in all directions. This keeps the block from moving instead of growing.

The outer boundaries get an equal *flux source*, which is the external force. Because of symmetry, the inner boundaries get a zero-constraint on the displacement velocity in the outward directed normal.

For this geometry, two growth tensors have been tried on to see their effects.

1. Constant growth: a constant growth tensor, with C_g= 0.1 s^{-1}.

2. Growth by norm: a growth tensor depending on the distance to its centre: $0.1 \cdot (x^2 + y^2 + z^2)\ s^{-1}m^{-2}$.

This gives the entries of the main diagonal of the growth tensor. The off-diagonals are set to zero.

Note that the growth and displacement will now be in all directions. Therefore, a longitudinal, vertically and horizontally swept mesh is used (a combination of a *mapped* and *swept* mesh).

In Figure 4.14 the results of both growth tensors are given. Figure 4.14a shows the results for the constant growth factor. In all directions, the block grows evenly to more than two meters. This corresponds to the results in Figure 4.13b.

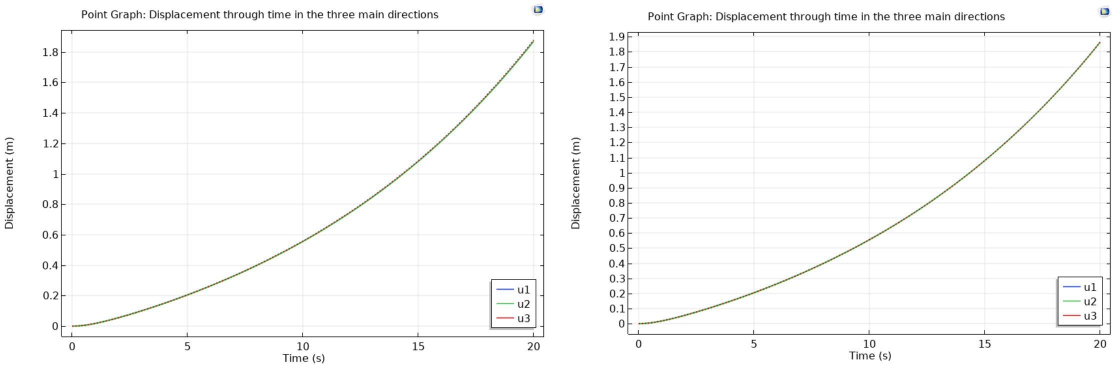
The second figure, 4.14b, shows a distance dependent growth factor. The growth factor is given by the norm of **x**, which is the distance between a point **x** and the origin. One can see that the corners grows the fastest, while the midpoints of the edges do not move much. When running for a longer time, the spikes, which can be seen in the solution (Figure 4.14b), will go through each other, which indicates that the model is not functioning correctly.

Growing Sphere To make an even more realistic shape, a sphere with a radius of 0.5 m has been examined. In this way, the results could be compared to the results of the block. For this problem, the same constant growth factors as for the growing block are used. Tetrahedral elements are used for the sphere.

Next to that, the behaviour is examined when the bulk viscosity μ_2 and Poisson ratio ν are taken into account. It is chosen to use a bulk viscosity of the same value as the shear viscosity, 10^9 $Pa \cdot s$, and a Poisson ratio of about rubber, 0.49.

In Figure 4.15 the displacement of both the growing sphere and the growing block are given. For this the end points of the three main axis are used. The displacement of the corresponding axis is then plotted. One can see that the growth is similar for both objects.

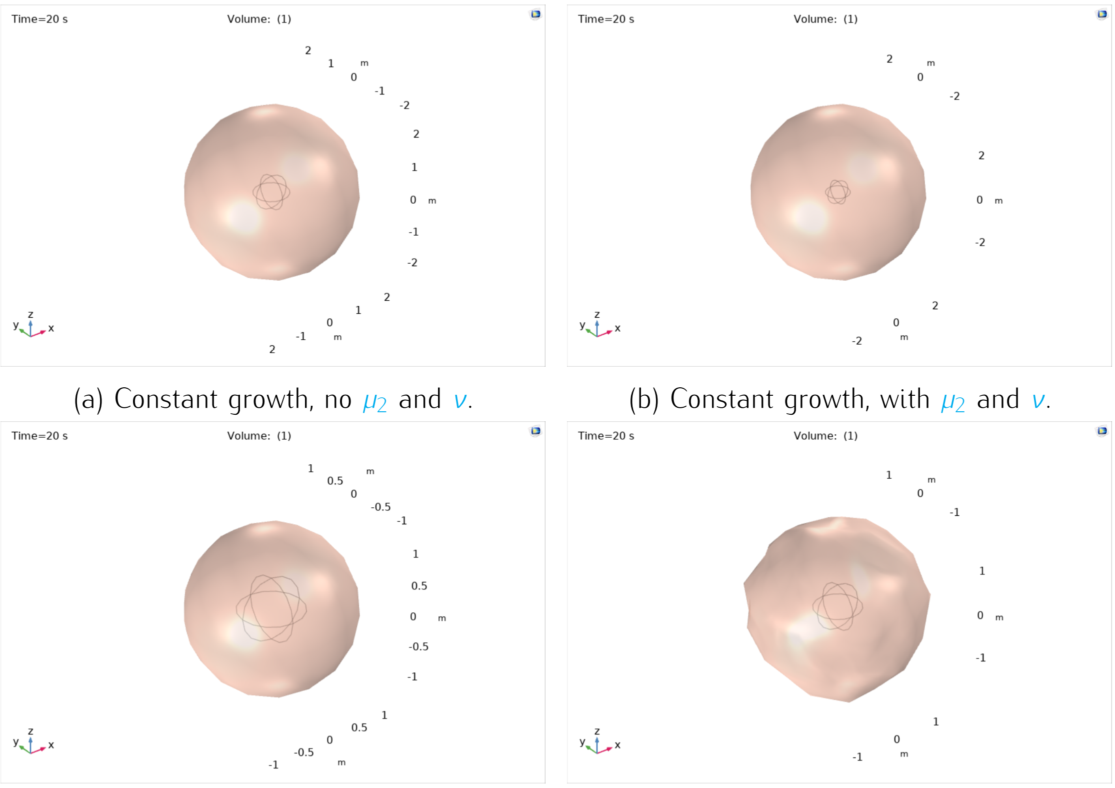
Figure 4.16 shows the results of the two different growth tensors, both with and without interdependence of the growth directions. One can see that without the interdependence, the sphere grows faster in all directions. Both methods keep the sphere shape regardless of the choice of including the interdependency. Although, in Figure 4.16d the sphere has gotten wobbly, due to the coarser mesh.



(a) Growing block.

(b) Growing Sphere

Figure 4.15: Displacement of both the growing block and the growing sphere without interdependence of the growth directions are given. The displacement of the end nodes of the three main axis are plotted. The displacement direction of the corresponding axis is plotted.



(a) Constant growth, no μ_2 and ν.

(b) Constant growth, with μ_2 and ν.

(c) Growth by norm, no μ_2 and ν.

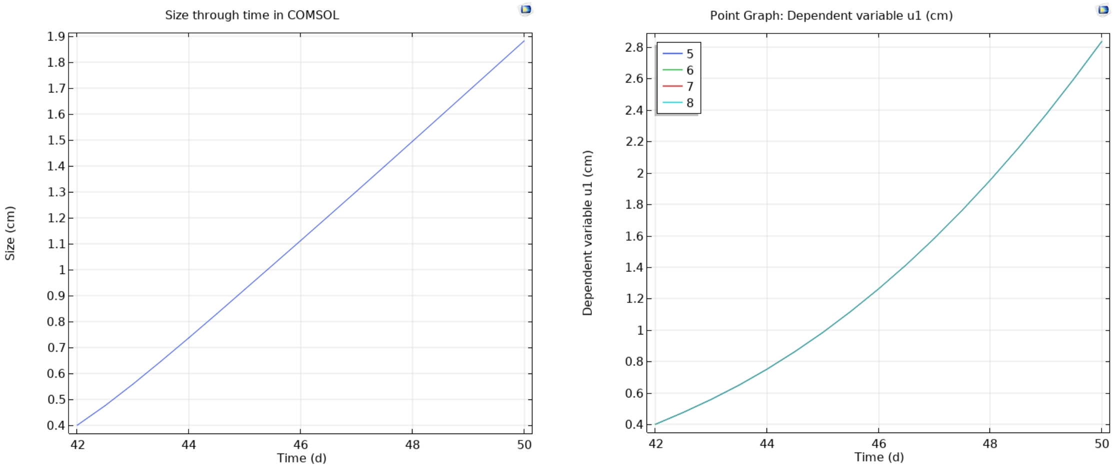
(d) Growth by norm, with μ_2 and ν.

Figure 4.16: Results of the growing sphere with different kinds of growth tensors, existing of a constant C_g of $0.1/s$ and given by the norm of $\mathbf{x}$. Both have been examined with and without the interdependence of the growth directions, by using either a positive or a zero bulk viscosity μ_2 and the Poisson ratio ν. For all these calculations a *coarser* mesh is used.

4.2.3 Results with Added Cell Properties

The idea is to verify the three dimensional problem containing the cell concentration and without any external force by the one dimensional results on this topic. Whereafter the situation can be extended to a sphere-like object which grows in multiple directions. Since the results while using the cell concentration PDE in the system were not realistic, this has not been looked at in multiple dimensions. Therefore, only the alternative idea of implementing a Gaussian to represent the cell concentration has been considered.

Alternative Idea The pulling bar model, that has been verified before, is modified by removing the external force and adding a body force generated by a cell concentration. Unfortunately, this final model gives an error of no convergence after a few time steps of which the amount varies per setting. Various settings in COMSOL Multiphysics® have been tried to solve this problem, however, no setting without this problem has been found yet. It can be seen that before the time step at which the program runs into a non convergence error, the results already start to deviate in the first time step from the results in one dimension. More on this can be found in the discussion: Section 5.1.



(a) 1D

(b) 3D

Figure 4.17: (a) shows the one dimensional displacement results when using a Gaussian to represent the cell concentration of the embryo. (b) shows a three dimensional result, seen from all four corners of the outermost plane (end plane of the rod). Both figures are given in a time frame of 8 days, starting from week 6 (day 42) till day 50.

4.3 Summary of the Results

Dim.	Geometry	COMSOL Multiphysics®	MATLAB	F	G	F_b	c-PDE	Correct
1D	Rod (1m)	✓	✓	✓	x	x	x	✓
1D	Rod (1m)	✓	✓	✓	✓	x	x	✓
1D	Rod (2mm)	✓	x	x	✓	✓	✓	?*
1D	Rod (2mm)	✓	x	x	✓	✓	x	✓
3D	Bar (1m)	✓	x	✓	x	x	x	✓
3D	Bar (1m)	✓	x	✓	✓	x	x	✓**
3D	Cube	✓	x	✓	✓	x	x	✓
3D	Sphere	✓	x	✓	✓	x	x	✓
3D	Bar (2mm)	✓	x	x	✓	✓	x	x***

Table 4.9: List of experiments, their properties and whether their results are verified to be correct.

* Since the end size of the embryo was not close to the values in literature, it has been decided not to put effort in verifying this method.

** Although not given in the results section, it has been verified by using the same constant growth factor C_g as in one dimension.

*** Deviation in first time step and a no convergence error is encountered.

5 Conclusion

A morphoelastic model is simply a mathematical description of the elasticity combined with growth. The generalised model is given in ($P2$).

This model is used to describe the growth of an embryo, mainly in one dimension. According to literature, the embryo size is first measured from crown-to-rump and after a couple of weeks this measurement method is switched to crown-to-heel. The resulting curves from the observations are shown in Figure 1.2a. Figure 1.3 shows the two lines ot the measurement methods over the embryonic body.

To get this type of curve, a PDE for the cell concentration c is used in this thesis. This worked partly in one dimension; the size of the embryo only got 100 times too small. Instead a Gaussian has been used as a replacement. In one dimension this showed a comparable curve with a relative error of 0.2316 and 0.0507 for the crown-to-rump and crown-to-heel measurement periods respectively.

These results are obtained by using a finite element approach. The resulting system of linear equations is given in Section 3.

To conclude, a morphoelastic model approximating the embryo growth in one dimension has been constructed in this thesis. However, the current results of the three dimensional model indicate that some modifications have to be made to improve the model. Suggestions for further research on modifications that have to be taken into account to improve this model in three dimensions are given in Section 5.1.

5.1 Discussion

During this research some observations were encountered which lead to proposals for improvement. Those observations and proposals are discussed below. First, the one dimensional model containing the cell concentration is discussed. Then the corresponding three dimensional model. And lastly, some general remarks on this research are given together with different theories that could be applied on this topic as well.

One Dimensional Model with the Cell Concentration By adding a PDE, describing the cell concentration (P_c), to the system, the mathematical model obtained a more realistic approach to capture the growth of an embryo. With this the model also incorporated the cell concentration, which is a biological phenomenon. With the cell concentration a body force is described in order to have an internal stimulus. Therefore, the external force was removed from the system. Since I was unable to fit the parameters of the cell concentration PDE in order to get a growth curve of the order of magnitude as in literature (Figure 1.2a), it raises the question whether adding just a single PDE, describing the phenomena of the cells, is sufficient to model the growth curve of an embryo.

In one dimension, this PDE (see Equation (P_c) in Section 2.3) gives a final growth which is too small (around 0.24 cm instead of 26 cm, since only half of the body is considered). The difficulty was to fit the parameters correctly. By this observation, it might be that other factors should be taken into account as well.

In the mathematical section, Section 2, it is already explained what effect all parameters in the equation have. A short recap on which restrictions on parameter choices we have to deal with is given next: the relation between the proliferation rate β and diffusion rate of cells D give the standard deviation of the resulting Gaussian. Both quantities cannot be too low or too high in magnitude, since this effects the time interval in which the body grows. Furthermore, the ratio of the proliferation rate β and the displacement velocity v (Péclet number) must be taken into account in order to get a stable, non oscillating, result. The displacement velocity is present in the equation since we work with a moving mesh. However, this results in a smaller impact of the force constant k and the constant growth factor C_g, which both effect the maximum growth. Removing this dependency on the displacement velocity in this PDE results in a larger growth.

This interplay of the parameters seems to have the unpleasant effect of not being able to get the growth curve from literature. Therefore, the model used in this thesis might be missing indispensable (cellular) phenomena for describing embryonic growth in a mathematical way. Koppenol successfully used this additional equation (P_c) for constructing a wound healing model [11]. However, he added four equations with this structure instead of one. Each of those describe one cell type, such as: fibroblasts, myofibroblasts, a generic signalling molecule, and collagen molecules. Because of his results, it is believed that it should be possible to use this type of PDE in order to describe the cell concentration.

This contributes to a second question: is the time interval of 40 weeks too long, or is the maximum displacement of 52 cm too much for this model? Koppenol [11] looked at a situation of a wound of about 4 cm in diameter within a two dimensional frame, representing the skin and the wound, of [-4:4, -4:4] cm. The total growth is, compared to the starting situation of the amount of skin, many times smaller then needed to model the growth of an embryo. This question is also motivated by the fact that the current, Hookean based morphoelastic model can only be used for small displacements [10]. However, it is not said what small displacements are. In Section 4, the model has been tested for extreme valued constant growth factors. The model results still looked stable, although the displacement was very large: 6 km. With this, it seems that the maximum displacement does not play a role in this. Therefore, it is still an unanswered question why using the c-PDE (P_c) does not work properly.

One could try to fit the Gaussian used as an alternative idea to represent the cell concentration by modifying the parameters of the PDE. Another idea is to put this model in MATLAB as well and see if the same problems occur. A last recommendation for this problem is to try to look at it from a different angle. An example is to look at adding a formula for the chemicals present in the uterus. One of the chemicals involved in embryonic growth is BMP4. This affects the way of growing, In the sense that it tells cells to which part of the body they belong, such as: arms, legs and organs [34].

Another idea for a follow-up research, is to fit two curves instead of one to represent both embryo measurement methods separately. It would be nice to see if the same biological and/or physical parameter values can be used for both curves, while both curves have a different initial length and start time.

Lastly, a comment on the alternative idea of adding cell properties to the system. In this thesis it is chosen to use (2.2), a Gaussian which grows over time. However, the standard deviation is not time dependent, and thus the Gaussian does not adjust to the current state of the body. In order to let the Gaussian move along, one could multiply the standard deviation s simply by the time t, which will also make the Gaussian satisfy a diffusion equation. Though, it will probably not spread accordingly with the displacement velocity. For this reason, one should try out which (time dependent) value for the standard deviation works best.

Three Dimensional Model with the Cell Concentration Contrary to the one dimensional case, the three dimensional situation containing a Gaussian representing the cell concentration is not verified. This model setup, unfortunately, results in a 'no convergence error'. Furthermore, it can be seen that the results, before the error is encountered, deviate from the ones in the one dimensional setting (Figure 4.17). This indicates that already something went wrong in the first time step and that it is not a result of accumulating (numerical) instabilities over time. Therefore, we have to consider that something is going wrong with how the equations are solved at an individual time step. This thought is because of the fact that the verified growing bar model has been adjusted to the one with a Gaussian representing the cell concentration.

Multiple settings within COMSOL Multiphysics® have been tried on this problem. First, it seemed like the mesh was not translated correctly over time. In this problem, a moving mesh was used in combination with a remeshing tool. The mesh was remeshed after the given distortion condition was fulfilled. At every time step, the current mesh was plotted. After some steps, the mesh seemed not to be okay anymore. Multiple elements were drawn through each other. Although, it seems like a big problem, it turned out not to be. The remeshed meshes within the *mesh* section all looked fine, as straight bricks. The plot however did not draw the mesh as the mesh was given. This was repaired by turning the *smoothing* off in the figures. Hence, the meshes where fine, the fact that they looked ill-shaped was caused by a post-processing error.

Together with the COMSOL Multiphysics® support, at least the following options have been modified: time stepping, and the tolerance; the initial mesh size, a moving mesh described by the edges instead of on the full geometry, and different remeshing properties (distortion, mesh quality, both for various values); discretisation order of the physics, and extra constraints on the physics to specify not to bend; the update frequency of the Jacobian, and the damping factor of the newton iterations in the fully coupled solver; scaling of the dependent variables; and lastly, the direct solver type (MUMPS vs PARADISO). Furthermore, the system has been tested when multiplying the body force by zero. Then the expected zero solution is found.

After having done all these modifications in order to reach an appropriate result, the following questions arise: Is the model implemented right? Is the chosen solver appropriate for the mathematical description which includes the cell concentration equation? And why does the first time step already deviate from the one dimensional case?

That the physics are implemented right, is partly discussed in Section 4.2. The difference to the situation described in that section is that we are now looking at the problem containing a body force, described by a Gaussian which represents the cell concentration, and without an external force. So the *flux/source* bound has been replaced by a *zero flux* bound in order to allow the bound to move outwards. Next to that, the body force in the v-PDE/physics is changed to the negative x-directional derivative of the suppressed Gaussian, which represents the cell concentration. This is done by a copy of the analytical formulas from the one dimensional COMSOL Multiphysics® model. Therefore, this hints at the correct implementation. As said before, when multiplying the body force by zero, the zero solution is found, which additionally sustains the claim that the implementation is correct.

The non linear solver in COMSOL Multiphysics® performs Newton iterations. This can be modified in different aspects, such as its damping factor. A Newton method needs a starting point. It could be that the chosen starting point is too far away from the solution, which can cause that the method will not converge. Another approach is to first use a couple of Picard iterations and afterwards start the Newton iterations from that point on. Picard iterations are generally slow, but computationally cheap. Because of its slow convergence, a few steps in the beginning of the non linear iterations can be helpful for the Newton approach in order to get a reasonable starting point. Therefore, I suggest to try this option as well. The down side is that COMSOL Multiphysics® does not contain this type of iteration method. Next to Newton methods, it only contains a *double dogleg* solver. This however, can only be used for stationary problems.

Looking at the results again (Figure 4.17), it seems that the curve is increasing too quickly, so a too big displacement could have been reached at the point the solver gives an error. In order to avoid the error, one should investigate what goes wrong in the first time step and how this accumulates through the next steps.

General Remarks First of all, it might have been a good idea to try the model in two dimensions as well. Results can be interpreted easier, both with the parameters and physics. Further, it would have been nice to see if the Gaussian representation of the cell concentration would have worked out in two dimensions, or if it already got stuck there.

For this research, only one mechanical physical value of the skin is used. More research on the mechanics of the skin or of an embryo would be necessary to make a more realistic mathematical model.

The one dimensional results have been verified by a one dimensional growth curve found in literature. In three dimensions, it would be nice to have a(n) (exact) three dimensional biological growth model of an embryo. For this, one could for example consider the three dimensional model discussed in [34]. This paper shows that an embryo already in an early stage cannot be described by symmetric models, which was a simplification adopted in this thesis because of computation time in combination with the preferred mesh size.

As a final idea, it could be interesting to dive into the theory of tensegrity, which provides an alternative way of modelling human tissue and its dynamics and growth. This principle describes relations between body parts in their tension. It comes from the idea that all body parts are connected to each other. Bones are attached to the fascia and so they are connected to other bones through your body. In this way, imbalances in the body can be described from places that are counter-intuitive since they are located somewhere else in the body then where you, e.g., feel the pain. This model describes generally how all the forces are divided over the body in order to stay in balance. Modelling growth can also be done using this model. In light of this research on embryonic growth, in the uterus, the embryo adapts to compressive forces, which influence the growth. More information about (bio)tensegrity can be found in [35] and [36].

Appendix

A Embryonic Growth Table

Table A.1: Crown-to-rump and Crown-to-heel length of an embryo in centimetres given per week according to two studies; the weight of an embryo according to these two studies in grams given per week; the Abdominal Circumference of an embryo in centimetres is given per week. All are measured for the gestational age of the embryo.

Week	C-R[3]	C-R[6]	C-H[3]	C-H[6]	Weight[3]	Weight[6]	AC[37]
	[cm]	[cm]	[cm]	[cm]	[g]	[g]	[cm]
6	0.4	–	–	–	–	–	–
7	1	–	–	–	–	–	–
8	1.6	1.6	–	–	1	1	–
9	2.3	2.3	–	–	2	2	–
10	3.2	3.1	–	–	35	4	–
11	4.2	4.1	–	–	45	7	–
12	5.3	5.4	–	–	58	14	–
13	6.5	7.4	–	–	73	23	–
14	7.9	8.7	–	–	93	43	–
15	–	10.1	16.41	–	117	70	–
16	–	11.6	18.3	–	146	100	10.4
17	–	13	20.1	–	181	140	11.4
18	–	14.2	22	–	223	190	12.5
19	–	15.3	23.7	–	273	240	13.5
20	–	16.4	25.5	25.60	331	300	14.5
21	–	–	27.2	26.7	399	360	15.5
22	–	–	28.8	27.8	478	430	16.5
23	–	–	30.4	28.9	568	501	17.5
24	–	–	32	30	670	600	18.5
25	–	–	33.6	34.6	785	660	19.5
26	–	–	35.1	35.6	913	760	20.5
27	–	–	36.5	36.6	1055	875	21.4
28	–	–	37.9	37.6	1210	1005	22.4
29	–	–	39.3	38.6	1379	1153	23.3
30	–	–	40.6	39.9	1559	1319	24.2
31	–	–	41.9	41.1	1751	1502	25.1
32	–	–	43.2	42.4	1953	1702	25.9
33	–	–	44.4	43.7	2162	1918	26.8
34	–	–	45.6	45	2377	2146	27.6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Week	C-R[3]	C-R[6]	C-H[3]	C-H[6]	Weight[3]	Weight[6]	AC[37]
	[cm]	[cm]	[cm]	[cm]	[g]	[g]	[cm]
35	–	–	46.7	46.2	2595	2383	28.4
36	–	–	47.8	47.4	2813	2622	29.2
37	–	–	48.9	48.6	3028	2859	29.9
38	–	–	49.9	49.8	3236	3083	30.6
39	–	–	50.9	50.7	3435	3288	31.3
40	–	–	52	51.2	3619	3462	31.9
41	–	–	52.7	51.7	3787	3597	32.5

Table A.1: [Cont.] Crown-to-rump and Crown-to-heel length of an embryo in centimetres given per week according to two studies; the weight of an embryo according to these two studies in grams given per week; the Abdominal Circumference of an embryo in centimetres is given per week. All are measured for the gestational age of the embryo.

B The Young's Modulus of an Embryo

The Young's modulus is a physical value for the elasticity of a material. For the human skin, different measurement methods are available: tension, suction, indentation, and torsion are the most commonly used methods. Next to this, the measurement can take place on different locations on the skin. All these have influence on the results. Therefore, the numbers in literature vary widely.

Reference	Method	Location	Age	Young's modulus
Jansen (1958)	Tension	Abdomen	0–99	2.9–54.0 MPa
Dunn (1958)	Tension	Abdomen/Thorax	47–86	18.8 MPa
Vogel (1987)	Tension	Various	0–90	15–150 MPa
Jacquemoud (2007)	Tension	Forhead/Arm	62–98	19.5–87.1 MPa
Agache (1980)	Torsion	Back	3–89	0.42–0.85
Diridollou (1998)	Suction	Forhead/Arm	20–30	0.12–0.25 Mpa
Khatyr (2004)	Tension	Tibia	22-68	0.13–0.66 MPa
Pailler-Mattei (2008)	Indentation	Arm	30	0.0045–0.008 MPa
Zahouari (2009)	Indentation/ Static Friction	Arm	55–70	0.0062–0.0021 MPa
Annaidh (2012)	Tensile	Back	81–97	83±34.9 MPa
Pawlaczyk (2013)	Torsion	-	-	0.42–0.85 MPa
	Mechanical	-	-	4.6–20 MPa
	Suction	-	-	0.05–0.15 MPa

Table B.1: The Young's modulus of the skin from various studies together with the age range, method and location on the skin. The first ten are from [15], the last three values from [16].

First the data is separated into two groups: greater and less than age 40. When given an age range, the first given modulus will be represented by the youngest age. This since the skin get stiffer over time and therefore the Young's modulus will get bigger.

In this thesis, we will look at an embryo, which is an unborn child. Therefore, only the group less than 40 years old is considered. Of this group, the outliers will be deleted. Those are having the value of 0.00035 MPa and 0.00625 MPa.

An average of the remaining values is then used as the Young's modulus of the embryo. This value is 0.23 MPa.

C Partial Differential Equations

In this appendix, an overview of the used PDEs is given. There are PDEs for the displacement u, the displacement velocity v, the infinitesimal strain (tensor) ε and the cell concentration c.

C.1 One-Dimensional PDEs

Linear Elasticity

$$\begin{cases} -Eu_{xx} = F_b; & x \in (x_0, x_L] \\ u(0) = a & a \in \mathbb{R} \\ E\dfrac{du}{dx} = F; & x = x_L \end{cases} \tag{P_L}$$

With F an external force applied to the system, and $x_0, x_L \in \mathbb{R}$ defining the length and position of the rod. The Cauchy stress (tensor) and infinitesimal strain (tensor) are given by

$$\begin{cases} \sigma = E\epsilon; \\ \epsilon = \dfrac{du}{dx}. \end{cases}$$

For this system an exact solution can be found. The exact solution for the displacement is given by a linear polynomial: $u(x) = F/E * x$ for $x \in [0, 1]$. Next to this, the solution for the Cauchy stress (tensor) is then given by: $\sigma(x) = F$.

Viscoelasticity The full viscoelastic problem is given below. To make it quasi-static, the second time t derivative of the displacement u is set to zero.

$$\begin{cases} \rho\dfrac{\partial^2 u}{\partial t^2} - \dfrac{\partial \sigma}{\partial x} = F_b & x \in (x_0, x_L], \quad t \in (0, T] \\ \sigma = \mu\dfrac{\partial \varepsilon}{\partial t} + E\varepsilon & x \in (x_0, x_L], \quad t \in (0, T] \\ \varepsilon = \dfrac{\partial u}{\partial x} & x \in (x_0, x_L], \quad t \in (0, T] \\ \dfrac{\partial u}{\partial t} = v & x \in (x_0, x_L), \quad t \in (0, T] \\ v(x, 0) = 0 & x \in [x_0, x_L] \\ u(x, 0) = 0 & x \in [x_0, x_L] \\ u(0, t) = 0 & t \in [0, T] \\ \sigma = F & x = x_L, \quad t \in (0, T1] \\ \sigma = 0 & x = x_L, \quad t \in (T1, T] \end{cases} \tag{P_V}$$

This system can be rewritten since Equations 2 – 4 can be implemented in Equation 1. As it is preferred to start in rest, $u(0, x)$ is set to zero. A force F will then be applied until a time $T1$ has been reached.

Morphoelasticity The morphoelastic problem below is subject to the boundary and initial conditions of Equation (P_V).

$$\begin{cases} \rho\left(\dfrac{Dv}{Dt} + v\dfrac{\partial v}{\partial x}\right) - \dfrac{\partial \sigma}{\partial x} = F_b & x \in (x_0, x_L], \quad t \in (0, T]; \\ \sigma = \mu\dfrac{\partial v}{\partial x} + E\varepsilon & x \in (x_0, x_L], \quad t \in (0, T]; \\ \dfrac{D\varepsilon}{Dt} + (\varepsilon - 1)\dfrac{\partial v}{\partial x} = -G & x \in (x_0, x_L], \quad t \in (0, T]; \end{cases} \tag{P_M}$$

This requires a post-processing step to obtain the displacement u.

$$\begin{cases} \dfrac{Du}{Dt} = v & x \in (x_0, x_L], \quad t \in (0, T]; \\ x(t) = X + u(x, t) & x \in [x_0, x_L], \quad t \in [0, T] \end{cases}$$

Here, $x(t)$ is the current position of a point in your body seen from the new frame, X is the starting position in the original frame and u is the displacement.

These equations all use the material derivative, which is given by

$$\begin{aligned} \frac{Du}{Dt} &= \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{\partial x}{\partial t} \\ &= \frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} \end{aligned}$$

and can be applied in a similar way to all of them.

Cell Concentration

$$\begin{cases} \dfrac{\partial c}{\partial t} + c\dfrac{\partial v}{\partial x} - \dfrac{\partial}{\partial x}\left(D(x, t)\dfrac{\partial c}{\partial x}\right) = \beta c(1 - c) & x \in [x_0, x_L], \quad t \in (0, T]; \\ D(x, t)\dfrac{\partial c}{\partial x} = 0, & x = 0, x = x_L \quad t \in (0, T]; \\ c(x(0), 0) = c_0 \cdot N(0, s), & x \in [0, x_L] \end{cases} \tag{P_c}$$

Alternative to the Cell Concentration PDE

$$c_1\left[2\pi s^2\right]^{1/2} \exp\left(-\frac{|x|^2}{2s^2}\right) * (1 - \exp(-\beta t)) \tag{C.1}$$

C.2 Three-Dimensional PDEs

Morphoelasticity

$$\begin{cases} \rho\left(\frac{D\mathbf{v}}{Dt} + \mathbf{v}\nabla\cdot\mathbf{v}\right) - \nabla\cdot\sigma = \mathbf{F_b}, & \mathbf{x}\in\Omega, \quad t\in(0,T]; \\ \sigma = \mu_1\,\mathrm{sym}(L) + \mu_2 tr(\mathrm{sym}(L))\mathbb{I} + \dfrac{E}{1+\nu}\left(\varepsilon + \dfrac{\nu}{1-2\nu}tr(\varepsilon)\mathbb{I}\right), & \mathbf{x}\in\Omega, \quad t\in(0,T]; \\ L = \nabla\mathbf{v}, & \mathbf{x}\in\Omega, \quad t\in(0,T); \\ \frac{\mathcal{D}\varepsilon}{\mathcal{D}t} + (\mathrm{tr}(\varepsilon)-1)\mathrm{sym}(L) = -G, & \mathbf{x}\in\Omega, \quad t\in(0,T); \\ \frac{\mathcal{D}\varepsilon}{\mathcal{D}t} = \frac{D\varepsilon}{Dt} + \varepsilon\,\mathrm{skw}(L) - \mathrm{skw}(L)\varepsilon, & \mathbf{x}\in\Omega, \quad t\in(0,T); \end{cases} \tag{P_M}$$

subject to

$$\begin{cases} \mathbf{v}(\mathbf{x},0) = \mathbf{0} & \mathbf{x}\in\Omega; \\ \mathbf{u}(\mathbf{x},0) = \mathbf{0} & \mathbf{x}\in\Omega; \\ \mathbf{u}(\mathbf{x},t) = \mathbf{0} & \mathbf{x}\in\partial\Omega_1, \quad t\in(0,T]; \\ \sigma\cdot\mathbf{n} = \begin{pmatrix} F & 0 & 0 \end{pmatrix} & \mathbf{x}\in\partial\Omega_2, \quad t\in(0,T1]; \\ \sigma\cdot\mathbf{n} = \mathbf{0} & \mathbf{x}\in\partial\Omega\setminus(\partial\Omega_1\cup\partial\Omega_2), \quad t\in(T1,T). \end{cases} \tag{BC_M}$$

Cell Concentration

$$\begin{cases} \dfrac{Dc}{Dt} + c\nabla\cdot v - \nabla\cdot(D(\mathbf{x},t)\nabla c) = \beta c(1-c), & \mathbf{x}\in\Omega, \quad t\in(0,T); \\ (D(\mathbf{x},t)\nabla c)\cdot\mathbf{n} = 0, & \mathbf{x}\in\partial\Omega; \\ c(\mathbf{x}(0),0) = c_0(\mathbf{x}(0)). & \mathbf{x}\in\Omega \end{cases} \tag{P_c}$$

D Derivation of the One-Dimensional Evolution Equation of the Infinitesimal Strain

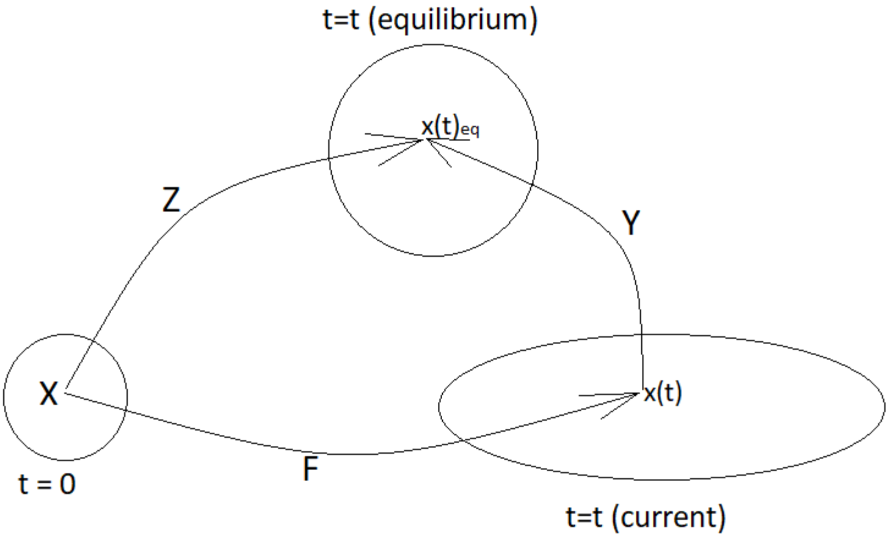


Figure D.1: Transformation of a body over time. Its original state, current state and equilibrium state corresponding to the current state.

This section is based on the calculations made in Hall [10]. Here, only the one dimensional case is described. The zero stress deformation gradient Y is given by $\frac{\partial x_{eq}}{\partial x}$. The infinitesimal strain (tensor) ε is written as $\varepsilon = \frac{L(t) - L(0)}{L(t)}$, where $L(t)$ is the length of the rod at time t. This can be transformed into $\varepsilon = 1 - Y$, and thus

$$Y = 1 - \varepsilon. \tag{D.1}$$

In a same way as for Y, F and Z can be described as $\frac{\partial x}{\partial X}$ and $\frac{\partial x_{eq}}{\partial X}$ respectively. In this way, F can be transformed into $\frac{\partial x}{\partial x_{eq}} \frac{\partial x_{eq}}{\partial X} = Y^{-1}Z$. If we look at the change in Z : $\frac{D\Delta Z}{Dt} = g(x,t)\Delta x$, for small values of Δx.

$$\frac{D\Delta Z}{Dt} = \frac{DZ}{Dt}\Delta X = \frac{D(YF)}{Dt}\Delta X = \frac{D(Y\Delta x)}{Dt} = g\Delta x \tag{D.2}$$

In this equation, $g = g(x,t)$ has units of inverse time and represents the rate of growth at point x. This equation can be converted to

$$\frac{D(YF)}{Dt} = gF.$$

Multiplying this by F^{-1} and working out de total derivatives gives

$$\begin{aligned} &\frac{\partial Y}{\partial t} + \frac{\partial Y}{\partial x} v + Y \frac{\partial v}{\partial X} F^{-1} = g \\ \Leftrightarrow \quad &\frac{\partial Y}{\partial t} + \frac{\partial (Yv)}{\partial x} = g. \end{aligned} \tag{D.3}$$

These steps are allowed since Δx is small.

Now we substitute Equation (D.1) into (D.3) and obtain

$$\begin{aligned} &\frac{\partial}{\partial t}(1-\varepsilon) + \frac{\partial}{\partial x}(v(1-\varepsilon)) = g \\ \Leftrightarrow \; &-\frac{\partial}{\partial t}\varepsilon + (1-\varepsilon)\frac{\partial}{\partial x} v + v\frac{\partial}{\partial x}(1-\varepsilon) = g \\ \Leftrightarrow \; &\frac{\partial}{\partial t}\varepsilon + (\varepsilon - 1)\frac{\partial v}{\partial x} + v\frac{\partial \varepsilon}{\partial x} = -g \\ \Leftrightarrow \; &\frac{\partial}{\partial t}\varepsilon + \frac{\partial x}{\partial t}\frac{\partial \varepsilon}{\partial x} + (\varepsilon - 1)\frac{\partial v}{\partial x} = -g \\ \Leftrightarrow \; &\frac{D\varepsilon}{Dt} + (\varepsilon - 1)\frac{\partial v}{\partial x} = -g \end{aligned}$$

which results in the infinitesimal strain (tensor) equation used for morphoelasticity.

E Stability

Consider the one dimensional morphoelastic problem stated in Section 2 with zero growth and zero body force:

$$\begin{cases} \rho \dfrac{\partial v}{\partial t} + 2\rho v \dfrac{\partial v}{\partial x} - \mu \dfrac{\partial^2 v}{\partial x^2} - E \dfrac{\partial \varepsilon}{\partial x} = 0 \\ \dfrac{\partial \varepsilon}{\partial t} + (\varepsilon - 1) \dfrac{\partial v}{\partial x} + v \dfrac{\partial \varepsilon}{\partial x} = 0. \end{cases} \tag{S_1}$$

The boundary conditions are given by:

$$\begin{cases} v(0,t) = 0 \\ \dfrac{\partial v}{\partial x}(1,t) = 0. \end{cases}$$

In this appendix, the stability of (S_1) around the equilibrium points

$$(v, \varepsilon) = (0, \varepsilon_0), \;\; \varepsilon_0 \in \mathbb{R},$$

is investigated. In order to do so, this system is linearised around the equilibrium points where $(v, \varepsilon) = (\delta_1 \hat{v}, \varepsilon_0 + \delta_2 \hat{\varepsilon}), \;\; \delta_{1,2} \in \mathbb{R}$ have been used as perturbations. The linearised system is given by:

$$\begin{cases} \dfrac{\partial \hat{v}}{\partial t} - \dfrac{\mu}{\rho} \dfrac{\partial^2 \hat{v}}{\partial x^2} - \dfrac{E}{\rho} \dfrac{\partial \hat{\varepsilon}}{\partial x} = 0 \\ \dfrac{\partial \hat{\varepsilon}}{\partial t} + (\varepsilon_0 - 1) \dfrac{\partial \hat{v}}{\partial x} = 0. \end{cases} \tag{S_2}$$

The following general solutions will be used next:

$$\begin{aligned} v(x,t) &= e^{\lambda_v t + ikx} \\ \varepsilon(x,t) &= e^{\lambda_\varepsilon t + ikx}. \end{aligned}$$

In the end, it turns out that the frequencies k do not have any influence on the stability. Therefore, they are chosen to be equal. The system (S_1) is stable if $Re(\lambda_v) \leq 0$, and $Re(\lambda_\varepsilon) \leq 0$.

These general solutions are substituted into the linearised system (S_2):

$$\begin{cases} \lambda_v \, e^{\lambda_v t + ikx} + k^2 \, \dfrac{\mu}{\rho} \, e^{\lambda_v t + ikx} - ik \, \dfrac{E}{\rho} \, e^{\lambda_\varepsilon t + ikx} = 0 \\ \lambda_\varepsilon \, e^{\lambda_\varepsilon t + ikx} + (\varepsilon_0 - 1) \, ik \, e^{\lambda_v t + ikx} = 0. \end{cases} \tag{S_3}$$

Division by $e^{\lambda_v t+ikx}$ in the first equation and by $e^{\lambda_\varepsilon t+ikx}$ in the second leads to

$$\begin{cases} \lambda_v + k^2\,\frac{\mu}{\rho} - ik\,\frac{E}{\rho} e^{(\lambda_\varepsilon - \lambda_v)t} = 0 \\ \lambda_\varepsilon \ + (\varepsilon_0 - 1)\; ik\; e^{(\lambda_v - \lambda_\varepsilon)t} = 0. \end{cases} \tag{S_4}$$

Looking only at the real terms of those equations gives

$$\begin{cases} Re(\lambda_v) = -k^2\,\frac{\mu}{\rho} \leq 0 \\ Re(\lambda_\varepsilon) = 0. \end{cases}$$

Thus, the system (S_1) is stable.

Looking at the results of the pulling rod problem in this thesis, one can assume that the real parts of the eigenvalues λ_v and λ_ε are equal. In this problem, one pulls a rod with constant force. This force is applied for a limited time. After that time, the rod is loosened. The rod will reach a maximum displacement corresponding to its mechanical properties and the applied force. The results show that this maximum displacement is constant. No oscillations are found, neither with smaller time steps. With this, one can conclude that the real part of the eigenvalues are equal, and thus zero. This is since the last terms of both equations in (S_4):

$$- ik\,\frac{E}{\rho} e^{(\lambda_\varepsilon - \lambda_v)t} \qquad \text{and}$$
$$(\varepsilon_0 - 1)\; ik\; e^{(\lambda_v - \lambda_\varepsilon)t},$$

should be constant over time when no oscillations are found.

F Numerics of this Thesis

F.1 Velocity Displacement PDE

Galerkin Weak Form

$$\text{Find } v_h \in H^1(I_t) \text{ such that } \forall \psi_h \in H^1(I_t)$$

$$\begin{aligned}\rho\frac{d}{dt}\int_{I_t}\psi_h v_h dx - \rho\int_{I_t}\frac{D\psi_h}{Dt}v_h\,dx + \int_{I_t}\mu\frac{\partial\psi_h}{\partial x}\frac{\partial v_h}{\partial x}\,dx = \\ \int_{I_t}\psi_h F_b - E\varepsilon\frac{\partial\psi_h}{\partial x}\,dx + \psi(x_L(t))F.\end{aligned} \tag{G_v}$$

Linear System

i) $v_h = \sum_{j=1}^{n+1}\xi_j(t)\varphi_j(x(t))$, with ξ the approximation of the displacement velocity v_h and $v(t,0) = 0$.

ii) $\psi_h = \varphi_j(x(t)), \quad i = 1, 2, \ldots, n+1$

Substitution of the above equations into the Galerkin weak form (G_v), results into

$$\frac{d}{dt}\sum_{j=1}^{n+1}\xi_j\underbrace{\rho\int_{I_t}\varphi_i\varphi_j\,dx}_{M_v} + \sum_{j=1}^{n+1}\xi_j\underbrace{\mu\int_{I_t}\varphi_i'\varphi_j'\,dx}_{A} = \underbrace{\int_{I_t}\varphi_i F_b - E\varepsilon\varphi_i'\,dx + \phi_i(x_L(t))F}_{\mathbf{b}_v}.$$

In short notation:

$$\frac{d}{dt}(M_v\xi) + A\xi = \mathbf{b}_v.$$

The mass matrix, stiffness matrix and load vector are given by the following.

$$M_v = \frac{\rho}{6}\begin{pmatrix} 2h(t)_1 + 2h(t)_2 & h(t)_2 & & & \\ h(t)_2 & 2h(t)_2 + 2h(t)_3 & h(t)_3 & & \\ & \ddots & \ddots & \ddots & \\ & & h_n & 2h(t)_n + 2h(t)_{n+1} & h(t)_{n+1} \\ & & & h(t)_{n+1} & 2h(t)_{n+1} \end{pmatrix};$$

$$A = \mu \begin{pmatrix} \frac{1}{h(t)_1} + \frac{1}{h(t)_2} & -\frac{1}{h(t)_2} & & & \\ -\frac{1}{h(t)_2} & \frac{1}{h(t)_2} + \frac{1}{h(t)_3} & -\frac{1}{h(t)_3} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h(t)_n} & \frac{1}{h(t)_n} + \frac{1}{h(t)_{n+1}} & -\frac{1}{h(t)_{n+1}} \\ & & & -\frac{1}{h(t)_{n+1}} & \frac{1}{h(t)_{n+1}} \end{pmatrix};$$

$$b_v = h(t)F_b - E\int_{I_t} \varepsilon\varphi' dx + F\mathbf{e}_{n+1}$$

Here $\mathbf{e}_{n+1}$ represents the unit vector, with only a one at the (n+1)th position. The leftover integral in the load vector will be described in detail below.

For every element of the mesh, the integral will have a different result. First we look at an inner node x_i. Since the space derivative of the basis function is not depending on the position x, we can drag this term out of the integral.

$$\begin{aligned} -E\int_{I_t} \varepsilon\varphi_i' \, dx &= -E\int_{x_{i-1}}^{x_{i+1}} \varepsilon\varphi_i' \, dx \\ &= -E\varphi_i' \int_{x_{i-1}}^{x_{i+1}} \varepsilon \, dx \\ &= -E\varphi_i' \sum_{k=i}^{i+1} \frac{\varepsilon_{k-1} + \varepsilon_k}{2} \cdot h_k \quad \text{(trapezoidal rule)} \\ &= E(h_i\varphi_i' \frac{\varepsilon_{i-1} + \varepsilon_i}{2} + h_{i+1}\varphi_i' \frac{\varepsilon_i + \varepsilon_{i+1}}{2}) \\ &= \frac{E}{2}(\varepsilon_{i+1} - \varepsilon_{i-1}) \end{aligned}$$

Since we only look at the segment around node x_i, the sum starts at i and ends at $i+1$. All others are equal to zero (see the description of the basis function in Section 3.1).

$$\mathbf{b}_v = \begin{pmatrix} h_1(t)F_b - \frac{E}{2}(\varepsilon_2 - \varepsilon_0) \\ h_2(t)F_b - \frac{E}{2}(\varepsilon_3 - \varepsilon_1) \\ \vdots \\ h_n(t)F_b - \frac{E}{2}(\varepsilon_{n+1} - \varepsilon_{n-1}) \\ h_{n+1}(t)F_b - \frac{E}{2}(\varepsilon_n + \varepsilon_{n+1}) + F \end{pmatrix}$$

Time Integration For this PDE the Euler backwards method is used.

$$\frac{M_v^{k+1}\xi^{k+1} - M_v^k \xi^k}{\Delta t} + A^{k+1}\xi^{k+1} = \mathbf{b}_v^k$$
$$\Rightarrow (M_v^{k+1} + \Delta t A^{k+1})\xi^{k+1} = M_v^k \xi^k + \Delta t \mathbf{b}_v^k$$

To be able to solve the above system easily, $(M_v^{k+1} + \Delta t A^{k+1})$ should be invertible. To show that this is true, it will be proved that this is symmetric positive definite (SPD).

Proof of its Symmetric Positive Definiteness Symmetry of M_v, A, and $M_v + \Delta t$ follows immediately from their definitions. It remains to prove the positive definiteness. For this, first we prove that M_v is positive definite, that is: $\mathbf{x}^T M_v \mathbf{x} > 0 \quad \forall \mathbf{x} \neq \mathbf{0}$. We get:

$$\mathbf{x}^T M_v \mathbf{x} = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} x_i m_{i,j} x_j = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} x_i \, \rho \int_{I_t} \varphi_i \varphi_j \, d\hat{x} \, x_j$$
$$= \rho \int_{I_t} \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} x_i \varphi_i \varphi_j x_j \, d\hat{x} = \rho \int_{I_t} (\sum_{i=1}^{n+1} x_i \varphi_i)^2 \, d\hat{x} \; > \; 0.$$

Further, we treat A similarly. It follows that

$$\mathbf{x}^T A \mathbf{x} = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} x_i a_{i,j} x_j = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} x_i \, \mu \int_{I_t} \nabla \varphi_i \cdot \nabla \varphi_j \, d\hat{x} \, x_j$$
$$= \mu \int_{I_t} |\nabla (\sum_{i=1}^{n+1} x_i \varphi_i)|^2 \, d\hat{x} \overset{*}{\geq} \alpha\mu \int_{I_t} (\sum_{i=1}^{n+1} x_i \varphi_i)^2 \, d\hat{x} \; > \; 0$$

* Poincaré's inequality, with $\alpha > 0$.

Herewith

$$\mathbf{x}^T (M_v + \Delta t A)\mathbf{x} = \mathbf{x}^T M_v \mathbf{x} + \Delta t \mathbf{x}^T A \mathbf{x} > 0 \, \forall x \neq \mathbf{0}.$$

Hence $(M_v + \Delta t A)$ is SPD. □

This together results in that $M_v^{k+1} + \Delta t A$ is SPD and moreover it is invertible. So the above system can be solved for ξ^{k+1} by the use of the Euler backwards method and the MATLAB backslash solver.

F.2 Infinitesimal Strain PDE

Galerkin Weak Form

$$\text{Find } \varepsilon_h \in H^1(I_t) \text{ such that } \forall \psi_h \in H^1(I_t)$$

$$\frac{d}{dt}\int_{I_t} \psi_h \varepsilon_h \, dx - \int_{I_t} \frac{D\psi_h}{Dt}\varepsilon_h \, dx = \int_{I_t} \psi_h\Big(\frac{\partial v}{\partial x} - G\Big)\, dx. \tag{G_ε}$$

Linear System

i) $\varepsilon_h = \sum_{j=0}^{n+1} \zeta_j(t)\varphi_j(x(t))$, with ζ the approximation of the infinitesimal strain (tensor) ε_h.

ii) $\psi_h = \varphi_i(x(t)), \quad i = 0, 2, \ldots, n+1$

Together with the transport property of basis functions, the system is given by

$$\frac{d}{dt}\sum_{j=1}^{n+1} \zeta_j \underbrace{\int_{I_t} \varphi_i \varphi_j \, dx}_{M_\varepsilon} = \underbrace{\int_{I_t} \varphi \left(\frac{\partial v}{\partial x} - G\right) \, dx}_{\mathbf{b}_\varepsilon}.$$

In short notation, this is given by:

$$\frac{d}{dt}(M_\varepsilon \zeta) = \mathbf{b}_\varepsilon.$$

The mass matrix for the infinitesimal strain (tensor) equation is given by

$$M_\varepsilon = \frac{1}{6}\begin{pmatrix} 2h(t)_1 + 2h(t)_2 & h(t)_2 & & & \\ h(t)_2 & 2h(t)_2 + 2h(t)_3 & h(t)_3 & & \\ & \ddots & \ddots & \ddots & \\ & & h(t)_n & 2h(t)_n + 2h(t)_{n+1} & h(t)_{n+1} \\ & & & h(t)_{n+1} & 2h(t)_{n+1} \end{pmatrix}$$

and the load vector by

$$\mathbf{b}_\varepsilon = \left(\frac{\partial v}{\partial x} - G \cdot \mathbf{1}\right) h(t),$$

where $\mathbf{1}$ is a vector filled with all ones.

For implementing this load vector, one needs the spatial derivative of the displacement velocity v. This, however, is not a result of the velocity system, though one can use this result in combination with a difference scheme to approximate its spatial derivative. Lets say $v = \begin{pmatrix} 0 & v_1 & v_2 & \dots & v_{n+1} \end{pmatrix}$, where v_i is the velocity in node i on the mesh. Then the spatial derivative in node i at time t can be approximated by

$$\frac{v_i(t) - v_{i-1}(t)}{h_i(t)},$$

with $h_i(t)$ the distance between node i and $i-1$. For the first and last equation for the load vector $\mathbf{b}_\varepsilon$, an upwind and backward scheme are respectively used. For the infinitesimal strain (tensor) PDE there are no boundary conditions needed. However, this means that the values on the boundary aren't known and thus need to be determined by solving the PDE.

Time Integration

$$\begin{aligned} \frac{M_\varepsilon^{k+1}\zeta^{k+1} - M_\varepsilon^k \zeta^k}{\Delta t} &= \mathbf{b}_\varepsilon^k \\ \Rightarrow M_\varepsilon^{k+1}\zeta^{k+1} &= M_\varepsilon^k \zeta^k + \Delta t \mathbf{b}_\varepsilon^k \end{aligned}$$

Because the mass matrix is SPD, this system in the spatial dimension can easily be solved with the MATLAB backslash solver.

F.3 Cell Concentration PDE

Galerkin Weak Form

$$\text{Find } c_h \in H^1(I_t) \text{ such that } \forall \psi_h \in H^1(I_t)$$

$$\frac{d}{dt}\int_{I_t} \psi_h c_h \, dx - \int_{I_t} \frac{D\psi_h}{Dt} c_h \, dx + \int_{I_t} D \frac{\partial \psi_h}{\partial x}\frac{\partial c_h}{\partial x} \, dx = \int_{I_t} \psi_h \beta c_h (1 - c_h) \, dx. \qquad (G_c)$$

Further use of this equation will be in the explicit form. This means that the right hand side will be taken from the last time step. With that the values for c_h are known.

Linear System

i) $c_h = \sum_{j=0}^{n+1} \eta_j(t)\varphi_j(x(t))$, with η the approximation of the cell concentration c_h.

ii) $\psi_h = \varphi_i(x(t)), \quad i = 0, 2, \dots, n+1$

Together with the transport property of basis functions, the system is given by

$$\frac{d}{dt}\sum_{j=1}^{n+1} \eta_j(t) \underbrace{\int_{I_t} \varphi_i \varphi_j \, dx}_{M_c} + \sum_{j=1}^{n+1} \eta_j(t) \underbrace{\int_{I_t} D \frac{\partial \varphi_i}{\partial x}\frac{\partial \varphi_j}{\partial x} \, dx}_{A_c} = \underbrace{\int_{I_t} \varphi_i \beta c^{t-1}(1 - c^{t-1}) \, dx}_{\mathbf{b}_c}.$$

One gets the short notation system

$$\frac{d}{dt}(M_\varepsilon \eta) + A_c \eta = \mathbf{b}_c.$$

The mass matrix is exactly the same as for the infinitesimal strain (tensor) equation. The stiffness matrix is similar to the one for the velocity equation, only this time it is multiplied by D, instead of μ. Last, the load vector by

$$\mathbf{b}_c = \beta c^{t-1}(1 - c^{t-1})h(t).$$

Time Integration The Euler Backwards method is used, while considering the fact that the right hand side will be taken from the last time step.

$$\begin{aligned} \frac{M_c^{k+1}\eta^{k+1} - M_c^k \eta^k}{\Delta t} + A_c^{k+1}\eta^{k+1} &= \mathbf{b}_c^k \\ \Rightarrow (M_c^{k+1} + \Delta t A_c^{k+1})\eta^{k+1} &= M_c^k \eta^k + \Delta t \mathbf{b}_c^k \end{aligned}$$

G COMSOL Multiphysics® LiveLink™ for MATLAB

The COMSOL Multiphysics® LiveLink™ for MATLAB has been used to reduce the risk of implementation errors. It has been used for automatically filling in the components of COMSOL Multiphysics® *PDE Coefficient Form* when working in three dimensions. For this, the formulas given in Section 2 are implemented in MATLAB by the use of the symbolic environment, and transformed via the COMSOL Multiphysics® LiveLink™ to a COMSOL Multiphysics® model.

In this subsection, it is briefly explained how the symbolic environment of MATLAB is used, and how this is implemented in COMSOL Multiphysics® by using the LiveLink™.

it is a good idea for first time users of the COMSOL *Multiphysics® LiveLink™ for* MATLAB *is to save a* COMSOL *Multiphysics® model as an '.m' file and open it in* MATLAB *afterwards. In this way, the* MATLAB *commandos to call* COMSOL *Multiphysics® commandos are shown.*

G.1 The MATLAB Symbolic Environment

The symbolic environment of MATLAB can do mathematical calculations with symbolic variables instead of just numbers. Symbolic equations are also used in COMSOL Multiphysics®. However, every matrix equation should be written out per matrix element. This has to be done since symbolic matrix equations are not available in the standard interface. It can, however, be done with the physics builder interface. The PDEs given in (*P*2) are very complex. For this reason, the symbolic environment of MATLAB in combination with the COMSOL Multiphysics® LiveLink™ for MATLAB would be a less error prone method to get all the formulas in a correct fashion in a COMSOL Multiphysics® model, than filling in the coefficients in COMSOL Multiphysics® directly.

As an example of how the symbolic environment works and how to use the LiveLink™, an example will be given using the symbolic expression of the Cauchy stress (tensor) σ.

$$\sigma = \mu_1\, sym(L) + \mu_2 tr(sym(L))\mathbf{I} + \frac{E}{1+nu}\left(\varepsilon + \frac{\nu}{1-2\nu} tr(\varepsilon)\mathbf{I}\right)$$

The equation for the Cauchy stress (tensor) σ consists of four interesting parts: $sym(.)$, $tr(.)$, L and ε, which will be described in the sense of the symbolic environment below. For $tr(.)$, the int-built the MATLAB function `trace(.)` is used to get the trace of a squared matrix.

For the symmetric part of a matrix ($sym(.)$), a small function has been written: `symm = @(L) 1/2*(L+L.')`. Note that the name of this function consists of a double 'm', since `sym(.)` already exists in MATLAB and constructs an object of the symbolic class from its input. This however has already been used (to create symbols), and overwriting base functions is general bad practice. Further note that the transpose of the matrix is taken with the notation `.'`, otherwise the conjugate transpose is taken, which is not needed in our case. Though, the conjugate transpose would not give any troubles because only real numbers are worked with. The MATLAB symbolic environment will write down how to get the conjugate transpose which is larger than necessary. So the real numbers transpose is used instead. Now the symmetric part of a function can be called by `symm(.)`.

The values of the variables L and ε are unknown (those are results of the PDE), therefore those matrices should also be given in symbolic form. For the infinitesimal strain (tensor) ε the self-made function `createMatrix_sym('var',n,m)` is used. This function creates an $n \times m$ matrix of variable '*var*'. The matrix entries will be named as follows: the entry of row i and column j will be called `vari_j`. This is easily done by using the commando `sym('var',[n m])`. Then the underscores are deleted by the use of `sym(strrep(string(mat),'_',''))`. In this way the 3×3 matrix for the infinitesimal strain (tensor) ε, knowing that the symbol 'ε' cannot be implemented in both MATLAB as COMSOL Multiphysics®, has the form:

$$\begin{bmatrix} E11 & E12 & E13 \\ E21 & E22 & E23 \\ E31 & E32 & E33 \end{bmatrix}$$

The infinitesimal strain (tensor) ε is said to be symmetric, and so the lower triangle is replaced by the transposed upper triangle `triu(A,0) + triu(A,1).'` to save memory.

$$\begin{bmatrix} E11 & E12 & E13 \\ E12 & E22 & E23 \\ E13 & E23 & E33 \end{bmatrix}$$

Finally, the symbolic formulation of the velocity gradient L is described. One first needs to know how to write a derivative in COMSOL Multiphysics®. There are two ways to describe, e.g, a spatial derivative of u:

1. `d(u,x)`;

2. `ux`.

The second option can only be used for dependent variables, such as the displacement u. For other type of variables, the first must be used. Since the velocity gradient contains spatial derivative of the dependent variable for the displacement velocity v, it is chosen to use the second option. To create the velocity gradient matrix, a function has been written called `grad3D_sym(A,isVar)`. A can be either a matrix ('isVar' is false) to which only the derivative form needs to be added, or a variable ('isVar' is true) of which the whole matrix including its derivatives needs to be created.

After a simple matrix (such as for the infinitesimal strain (tensor)) has been created, the spatial derivative notation is added. For this, a matrix for the spatials has been made via the `repmat(.)` function of MATLAB. Then the two matrices are added as characters. Consequently, the unnecessary parts are deleted and the following matrix is created.

$$\begin{bmatrix} v1x & v1y & v1z \\ v2x & v2y & v2z \\ v3x & v3y & v3z \end{bmatrix}$$

In the end, this matrix needs to be transposed to get the correct form for the velocity gradient L.

Note that the function `grad3D_sym(.)` *only works if the variable 'A' is lower in the alphabet than x, y and z, since this is how this symbolic function and* `char(.)` *works when adding variables (B + A always becomes A + B).*

To save some memory while solving in COMSOL Multiphysics®, the Voigt notation is used. This can be used since the infinitesimal strain (tensor) ε and so the Cauchy stress (tensor) σ are said to be symmetric. In this way, there are three fewer unknowns to solve. For this, a function called `voigtNotation_sym(.)` is implemented in MATLAB. It rewrites a symmetric matrix to a vector of only the elements of its upper triangle. The first elements of the vector are the diagonal elements of the matrix.

With all the above knowledge, the symbolic expression of the Cauchy stress (tensor) σ can be implemented. With similar tricks, the other formulas of ($P2$) are written out by the use of the symbolic environment of MATLAB.

G.2 Usage of the COMSOL Multiphysics® LiveLink™ for MATLAB

After all the preliminary work, such as creating the correct matrices has been done, it is implemented in COMSOL Multiphysics® with the use of the LiveLink™ for MATLAB.

The order in this MATLAB file is the same as how you should create a COMSOL Multiphysics® file. First, a model is selected and created. Subsequently a geometry is added, physics are added and a mesh is chosen. Of course more complex ideas can be added too this way, such as a parameter and variable list, but extra functions can also be added under *definitions*.

For every part, it first needs to be created before it can be adjusted. For example, if one adds a bar to the geometry, the the following lines of code are used.

```
1 model.component('comp1').geom.create('block', 3);
2 model.component('comp1').geom('block').label('Geometry: Bar');
3 model.component('comp1').geom('block').create('blk1', 'Block');
4 model.component('comp1').geom('block').run;
```

To the model component '*comp1*' a three dimensional geometry called or tagged '*block*' is added in line 1. As a next step, this geometry will get the label '*Geometry: Bar*', which can be handy when COMSOL Multiphysics® is opened. Then a standard unit cube is created and named '*blk1*'. To use the geometry throughout the code, it needs to be ran.

In a same way physics can be added to the file. When adding a *PDE Coefficient Form* to COMSOL Multiphysics®, a standard Poisson equation is already filled in. To avoid this, the entries of the diffusion (c), source (f), and mass (da) terms are set equal to zero. For instance, the mass term:

```
nv = length(V);
for i = 0:(nv-1)
    model.component('comp1').physics('v').feature('cfeq1').
        setIndex('da', 0, i*(nv+1)); %matrix da
end
```

Then, the values are added to the right coefficient. Again, the mass coefficient, for the displacement velocity v PDE, is shown as example.

```
for i = 0:(nv-1)
    model.component('comp1').physics('v').feature('cfeq1').
        setIndex('da', 'rho', i*(nv+1));
end
```

One thing to note is that this LiveLink^{TM} commandos would like to have strings as inputs, and thus no symbols. Therefore, all the symbolic terms are transformed to strings, by using the function `string()`.

As a last step, the boundary conditions are added. These conditions, however, differ between the cases; do we work on the pulling bar problem or the growing problem? The below example is of the pulling bar problem, where a force is applied on one boundary.

```
model.component('comp1').physics('v').create('flux1',
    'FluxBoundary', 2);
model.component('comp1').physics('v').feature('flux1').
```

```
    setIndex('g', '1e6*(t<=t_end/2) [Pa]', 0);
model.component('comp1').physics('v').feature('flux1').
    selection.set([6]);
```

First, the boundary condition of a *flux/source* is implemented for a two dimensional space, a surface boundary, and named '*flux1*'. Then, the value of this force is added to this boundary condition '*flux1*'. Lastly, it is set to be only applicable on boundary number 6.

This way of working with the combination of COMSOL Multiphysics® *and* MATLAB *has shown off its merits. The implementation worked amazing and the results, compared to the one dimensional cases, look excellent. Even after rotating the geometry into multiple positions, the results were as they should. For this see Appendix I.*

H One Dimensional Results

H.1 Viscoelastic Results Compared to Linear Elasticity

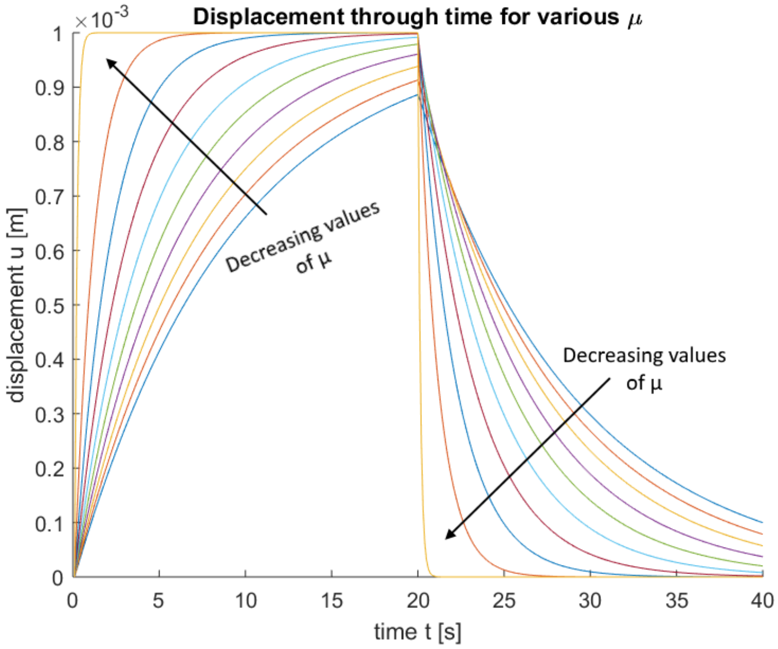
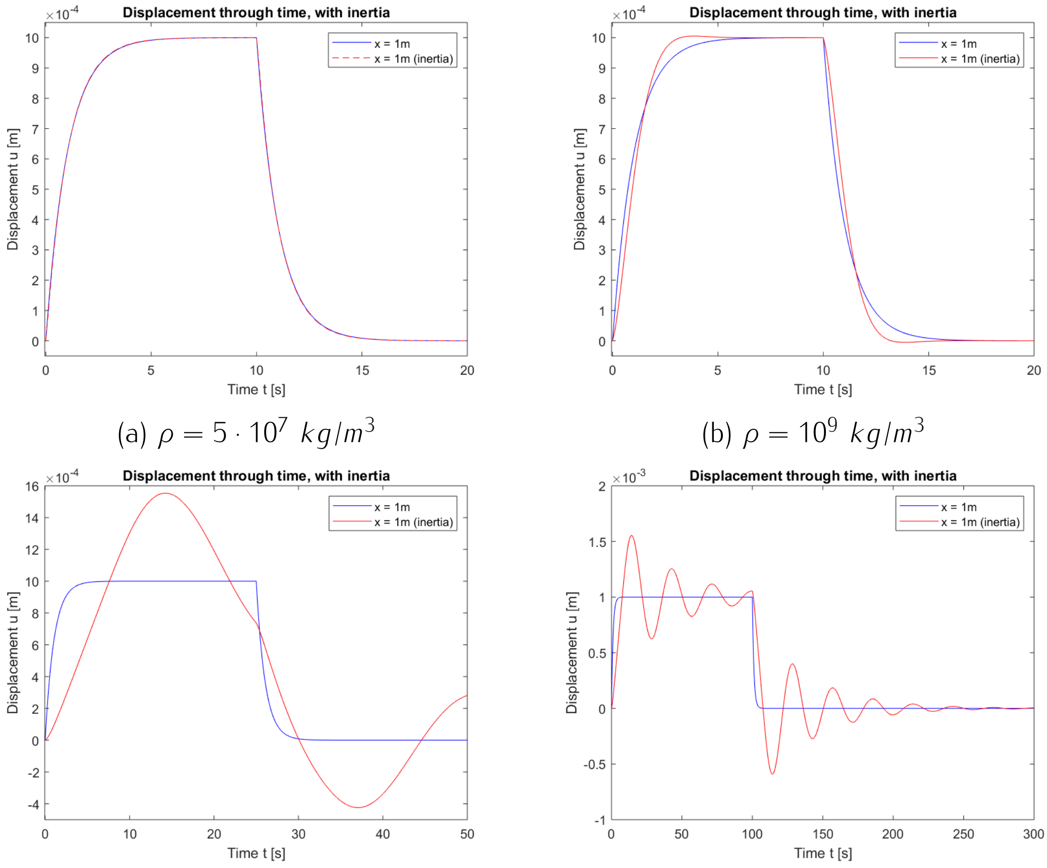


Figure H.1: Results of applying the viscoelastic model with different values of the viscosity μ. The end point of the rod with length one is plotted.

To see the effect of the viscosity, simulations over the first 40s have been shown. An external force is applied for the first 20s. In this examined system, no mass density is present. Only the effect of the shear modulus has been considered. The arrows show the decrease in shear modulus. The outer yellow line shows when the viscosity μ is near zero. This looks like linear elasticity. The maximum displacement when pulling by a constant external force has been reached (almost) immediately.

H.2 Viscoelasticity; Varying the Mass Density ρ



(a) $\rho = 5 \cdot 10^7 \ kg/m^3$

(b) $\rho = 10^9 \ kg/m^3$

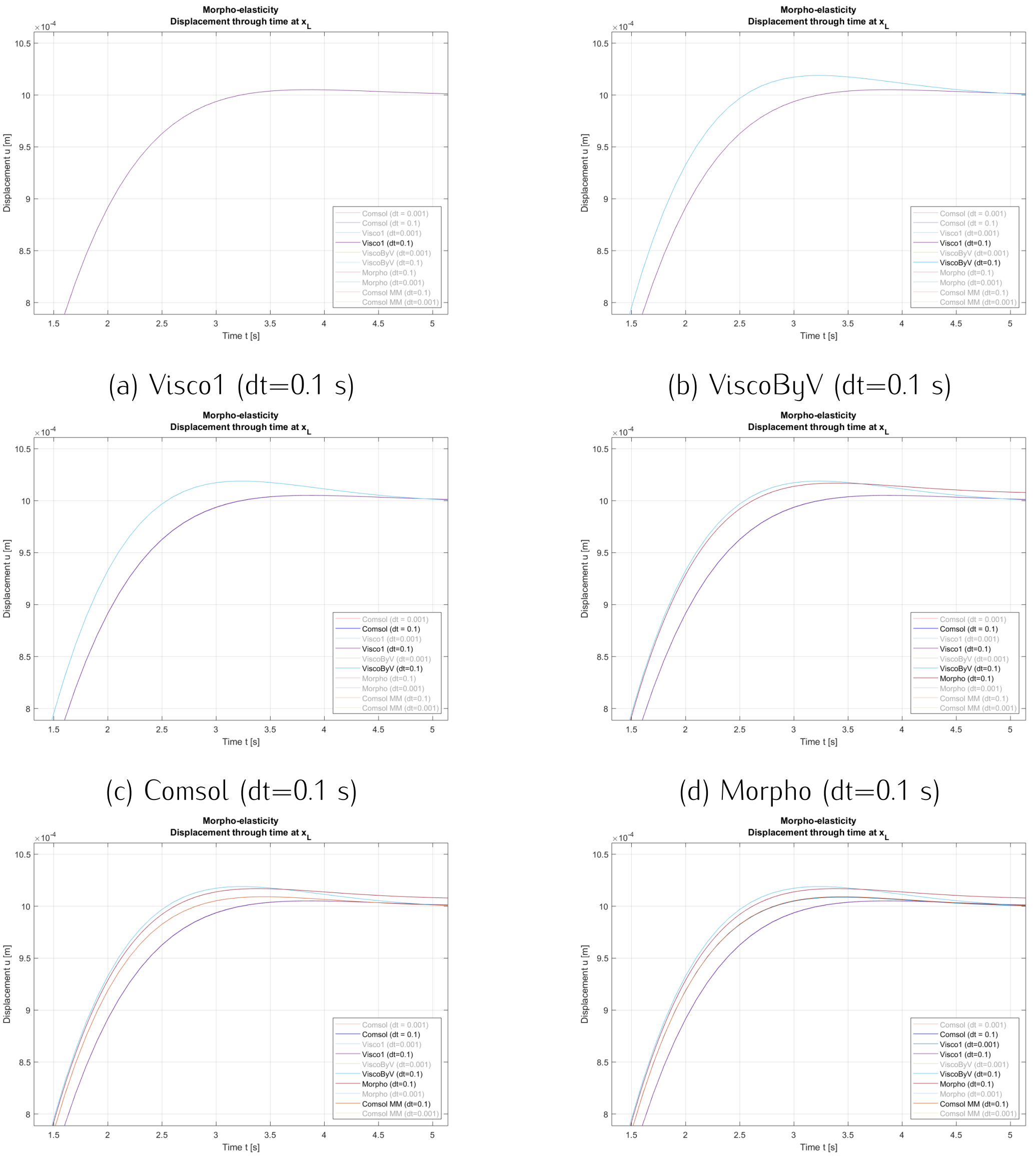
(c) $\rho = 5 \cdot 10^{10} \ kg/m^3$

(d) $\rho = 5 \cdot 10^{10} \ kg/m^3, \ T = 300s.$

Figure H.2: Results of applying the viscoelastic model with different values of the mass density ρ. The end point of the rod with length one is plotted.

The effects of the mass density ρ are shown in the figures above. In the case of a mass density with a higher order than the order of the Young's modulus E and the shear modulus μ, as in Figure H.2c, the behaviour of the system appears to be unstable. However, from Figure H.2d one can conclude that the oscillating effect gets damped and the solution is stable.

H.3 Viscoelastic Results vs. Morphoelastic Results



(a) Visco1 (dt=0.1 s)

(b) ViscoByV (dt=0.1 s)

(c) Comsol (dt=0.1 s)

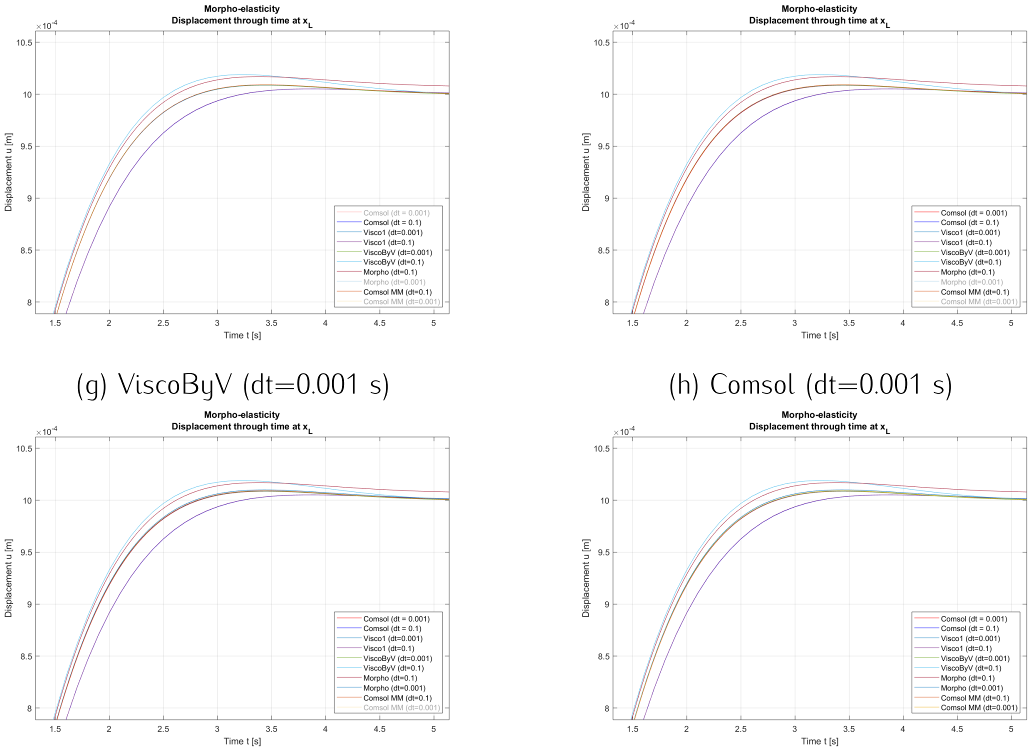
(d) Morpho (dt=0.1 s)

(e) Comsol MM (dt=0.1 s)

(f) Visco1 (dt=0.001 s)

Figure H.3: A comparison of the different solving methods. Visco1, ViscoByV, Comsol, Morpho, and Comsol MM are all plotted for two time steps (dt=0.1 s and dt=0.001 s), over a time of 50 seconds. Every figure (a) – (j) one line is added, which line is mentioned in their caption.

In the figures in this appendix, the zoomed results of a MATLAB viscoelastic model are compared to a MATLAB morphoelastic model and the morphoelastic model in COMSOL



(g) ViscoByV (dt=0.001 s)

(h) Comsol (dt=0.001 s)

(i) Morpho (dt=0.001 s)

(j) Comsol MM (dt=0.001 s)

Figure H.3: (Cont.) A comparison of the different solving methods. Visco1, ViscoByV, Comsol, Morpho, and Comsol MM are all plotted for two time steps (dt=0.1 s and dt=0.001 s), over a time of 50 seconds. Every figure (a) – (j) one line is added, which line is mentioned in their caption.

Multiphysics®.

Two solving techniques are used for the viscoelastic results. Both make use of a order reduction in the time derivative. The first method is to write the Euler Backwards time integration method in a system and to solve for two variables (the displacement u and the displacement velocity v). The results from this method are called *Visco1*. The second approach is the same as has been done and described for the morphoelastic problem, using a post processing step to find the displacement. This result is called *ViscoByV* in the figures above.

For the results created with COMSOL Multiphysics®, it is chosen to compare both morphoelastic calculation without a moving mesh (*Comsol*) and with a moving mesh (*Comsol MM*). Lastly, *Morpho* shows the results of the written MATLAB program to solve the morphoelastic model.

A first look shows that *Visco1* and *Comsol* have a slight undershoot, while *Vis-*

coByV and *Morpho* have a slight overshoot when using a bigger time step. Further, it is interesting that the *Comsol MM* results are in between, just on the line where all the results will be converging to when using a smaller time step.

I Three Dimensional Results

I.1 Rotations of the Pulling Bar Problem

In this appendix a verification of the three dimensional setup is given. The pulling bar setup is compared to the pulling rod problem. For this the displacement in the longitudinal direction of the bar is calculated for different directional states of the bar:

- placed with an edge on the x-axis (3Dx)
- placed with an edge on the y-axis (3Dy)
- placed with an edge on the z-axis (3Dz)
- placed with an edge on the x-axis, then rotated in space by 45° in the xy-plane and thereafter rotated by 45° in the plane that the longitudinal direction of the bar makes with the z-axis (3Drot)

All the results are plotted together in the figure below together with the one dimensional result. One can see that the results overlap.

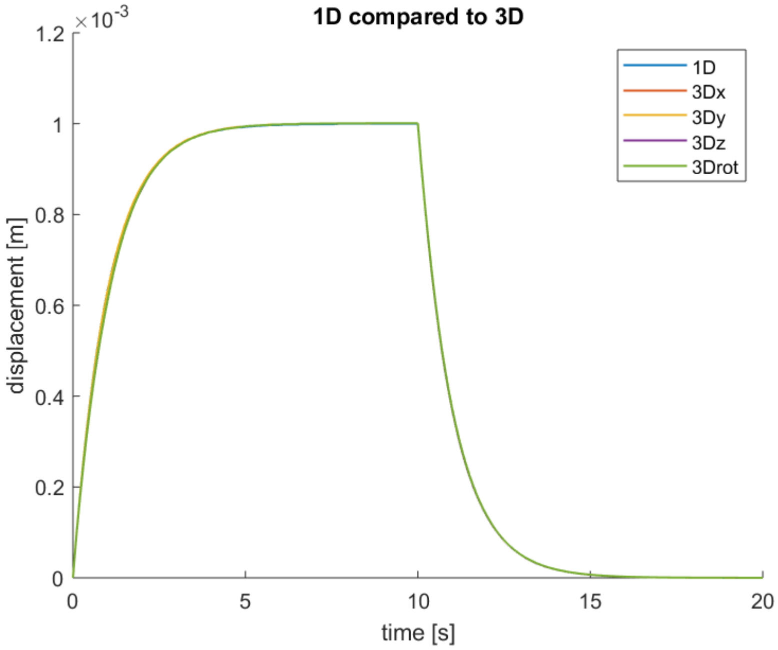


Figure I.1: One dimensional pulling rod results (blue) compared to the three dimensional pulling bar results.

I.2 Bar Bending

For the extreme growth trials, a bending takes place after an amount of time. This can be seen most easily for the constant growth factor of 1 s^{-1}. To show this, the y- and z-directional displacement is given in Figure I.2. Further, it can be seen that the solving from approximately that point of bending in time on gets tougher. Smaller time steps are taken (see Figure I.3).

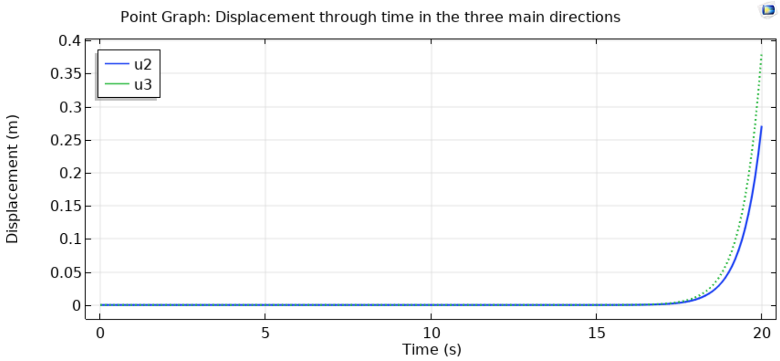


Figure I.2: A constant growth factor of 1 s^{-1} has been used for this result. Only the displacement in the y- and z-direction are drawn. However, the growth tensor only had values in the x-direction. One can see that the bar will eventually bend. Although the bar will bend, the bending is minimal compared to the x-directional displacement.

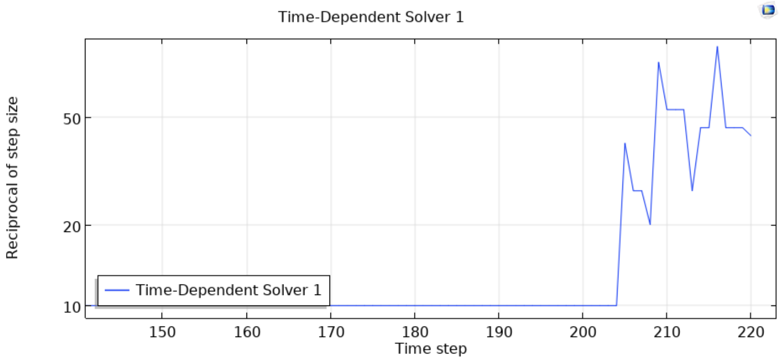


Figure I.3: One can see that solving gets tougher, because in the end, the solver takes bigger time steps.

References

[1] M. Grevinga, Y. Schönbeck, A. D. Hindori-Mohangoo, M. E. B. Reijnders, and S. B. Detmar, "Aangeboren afwijkingen in nederland 2010-2016: Gebaseerd op de landelijke perinatale registraties," tech. rep., TNO, 2018.

[2] J. Stone, K. Eddleman, and M. Duenwald, *Pregnancy for Dummies*. John Wiley & Sons, 4th ed., 2014.

[3] F. Mark A Curran, M.D., "Fetal development." http://perinatology.com/Reference/Fetal%20development.htm, 03 2019. Online article about the fetal development, described per week.

[4] Maag Lever Darm Stichting, "Anusatresie." https://www.mlds.nl/ziekten/anusatresie/, 08 2019.

[5] Vereniging Anusatresie. https://www.anusatresie.nl/, 08 2019.

[6] M. Hill, "Embryology." Crown-Rump Length, 08 2019. Retrieved from https://embryology.med.unsw.edu.au/embryology/index.php/Crown-Rump_Length.

[7] R. T. Tranquillo and J. D. Murray, "Continuum model of fibroblast-driven wound contraction: inflammation-mediation," *Journal of theoretical biology*, vol. 158, no. 2, pp. 135–172, 1992.

[8] L. Olsen, J. A. Sherratt, and P. K. Maini, "A mechanochemical model for adult dermal wound contraction and the permanence of the contracted tissue displacement profile," *Journal of theoretical biology*, vol. 177, no. 2, pp. 113–128, 1995.

[9] M. M and S. R, "Wound geometry and the kinetics of wound contraction," *Plast Reconstr Surg*, vol. 72, pp. 66–72, 1983.

[10] C. L. Hall, *Modelling of some biological materials using continuum mechanics*. PhD thesis, Queensland University of Technology, 2008.

[11] D. Koppenol, *Biomedical Implications from mathematical models for the simulation of dermal wound healing*. PhD thesis, TU Delft, 2017. https://doi.org/10.4233/uuid:cdc7392f-4ac6-404c-9615-dc425f67efae.

[12] Vivas, J., Garzón-Alvarado, D., and Cerrolaza, M., "Modeling cell adhesion and proliferation: a cellular-automata based approach," *Advanced modeling and simulation in engineering sciences*, vol. 2, no. 32, 2015.

[13] Á. Monteagudo and J. Santos, *A Cellular Automaton Model for Tumor Growth Simulation*, pp. 147–155. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[14] S. Shahmoradi, F. N. Rahatabad, and K. Maghooli, "A stochastic cellular automata model of growth of avascular tumor with immune response and immunotherapy," *Informatics in Medicine Unlocked*, vol. 12, pp. 81–87, Elsevier.

[15] A. N. Annaidh, K. Bruyère, M. Destrade, M. D. Gilchrist, and M. Otténio, "Characterization of the anisotropic mechanical properties of excised human skin," *Journal of the mechanical behavior of biomedical materials*, vol. 5, no. 1, pp. 139 – 148, 2012.

[16] M. Pawlaczyk, M. Lelonkiewicz, and M. Wieczorowski, "Age-dependent biomechanical properties of the skin," *Advances in Dermatology and Allergology/Poste py Dermatologii i Alergologii*, vol. 30, no. 5, pp. 302 – 306, 2013.

[17] P. G. Agache, C. Monneur, J. L. Leveque, and J. De Rigal, "Mechanical properties and young's modulus of human skin in vivo," *Archives of dermatological research*, vol. 269, no. 3, pp. 221 – 232, 1980.

[18] M. S. Hu, M. R. Borrelli, W. X. Hong, S. Malhotra, A. T. Cheung, R. C. Ransom, R. C. Rennert, S. D. Morrison, H. P. Lorenz, and M. T. Longaker, "Embryonic skin development and repair," *Organogenesis*, vol. 14, no. 1, pp. 46 – 63, 2018.

[19] P. Mott and C. Roland, "Limits to poisson's ratio in isotropic materials," *Physical review B*, vol. 80, no. 13, p. 132104, 2009.

[20] L. Huber, R. A. Perkins, A. Laesecke, D. G. Friend, J. V. Sengers, M. J. Assael, I. N. Metaxa, E. Vogel, R.Mares, K. Miyagaw, "New international formulation for the viscosity of h2o," *Journal of Physical and Chemical Reference Data*, vol. 38, 2009.

[21] J. M. Benítez and F. J. Montáns, "The mechanical behavior of skin: Structures and models for the finite element analysis," *Computers & Structures*, vol. 190, pp. 75–107, 2017.

[22] R. W. Ogden and G. A. Holzapfel, *Mechanics of biological tissue*. Springer, 2006.

[23] D. Ambrosi, G. A. Ateshian, E. M. Arruda, S. Cowin, J. Dumais, A. Goriely, G. A. Holzapfel, J. D. Humphrey, R. Kemkemer, E. Kuhl, *et al.*, "Perspectives on biological growth and remodeling," *Journal of the Mechanics and Physics of Solids*, vol. 59, no. 4, pp. 863–883, 2011.

[24] K. R. Swanson, C. Bridge, J. Murray, and E. C. Alvord Jr, “Virtual and real brain tumors: using mathematical modeling to quantify glioma growth and invasion,” *Journal of the neurological sciences*, vol. 216, no. 1, pp. 1–10, 2003.

[25] D. C. Koppenol, F. J. Vermolen, F. B. Niessen, P. P. van Zuijlen, and K. Vuik, “A biomechanical mathematical model for the collagen bundle distribution-dependent contraction and subsequent retraction of healing dermal wounds,” *Biomechanics and modeling in mechanobiology*, vol. 16, no. 1, pp. 345–361, 2017.

[26] R.Haberman, *Applied Partial Differential Equations*. Pearson Prentice Hall, 5th ed., 2013. ISBN: 9780321828972.

[27] COMSOL Inc., *COMSOL Multiphysics Reference Manual*, 2018. COMSOL Multiphysics® version 5.4.

[28] C. Vuik, F.J. Vermolen, M.B. van Gijzen, M.J. Vuik, *Numerical methods for ordinary differential equations*. Delft Academic Press, 2nd ed., 2016. ISBN: 9789065623737.

[29] J. van Kan, A. Segal, F.J. Vermolen, *Numerical methods in scientific computing*. Delft Academic Press, 2nd ed., 2014. ISBN: 9789065623638.

[30] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Computational Differential Equations*. Studentlitteratur. ISBN: 91-44-49311-8.

[31] W. Boon, D. Koppenol, and F. Vermolen, “A multi-agent cell-based model for wound contraction,” *Journal of biomechanics*, vol. 49, no. 8, pp. 1388–1401, 2016.

[32] G. Dziuk and C. M. Elliott, “Finite element methods for surface pdes,” *Acta Numerica*, vol. 22, pp. 289–396, 2013.

[33] COMSOL Inc. https://www.comsol.com/products, 10 2019.

[34] M. Simunovic, J. J. Metzger, F. Etoc, A. Yoney, A. Ruzo, I. Martyn, G. Croft, D. S. You, A. H. Brivanlou, and E. D. Siggia, “A 3D model of a human epiblast reveals BMP4-drive symmetry breaking,” *Nature cell biology*, vol. 21, no. 7, p. 900, 2019.

[35] S. Levin, *Tensegrity: The New Biomechanics*, pp. 69–80. 01 2006.

[36] L. van Renterghem, “Biotensegrity, a structural model for the human body as a unity, body and mind,” 2017.

[37] N. Shinozuka, N. Akamatsu, S. Sato, T. Kanzaki, H. Takeuchi, M. Natori, Y. Chiba, and T. Okai, “Ellipse tracing fetal growth assessment using abdominal circumference: Jsum standardization committee for fetal measurements,” *Journal of Medical Ultrasound*, vol. 8, pp. 87–94, 01 2000.