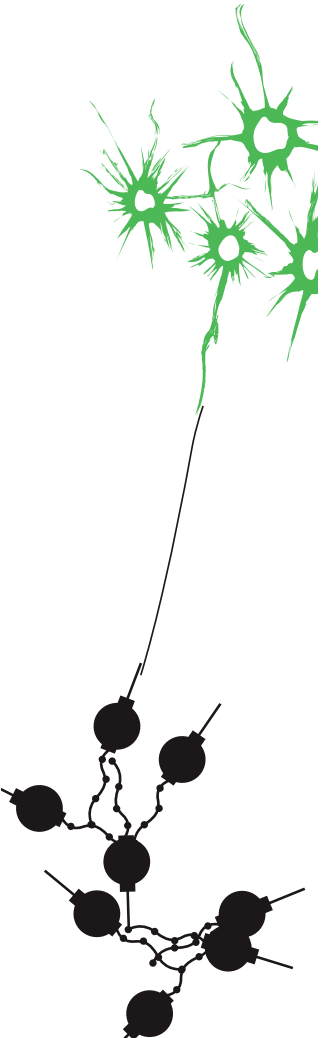


RAM ROBOTICS AND MECHATRONICS

DEFORMATION MODELLING AND GRIPPER CONTACT SIMULATION OF DEFORMABLE OBJECTS

J. (Jordi) Luong

MSC ASSIGNMENT

Committee:
prof. dr. ir. S. Stramigioli
dr. ir. W. Roozing
A. Brugnoli, Ph.D
dr. ir. J.J. de Jong

June, 2022

018RaM2022
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Summary

Grasping and manipulation of deformable objects is still a challenging problem in industrial automation and robotics. To grasp a deformable object the right amount of force needs to be used to on the one hand have a stable grasp and on the other hand prevent damaging the object. Besides, the object will deform when in contact with a grasping finger. Therefore, the behaviour of a deformable object and its contact with a finger needs to be understood. This thesis proposes a mass-spring-damper model with additional constraints for volume preservation and spring deformation limits, and Hunt-Crossley (Hunt and Crossley, 1975) and stick-slip contacts. Experiments with objects made of silicone rubber are conducted to acquire data about the deformation of an object when impacted by an external fingertip. The influence of model parameters on the simulation results are explored and fitted to minimize the error. The results indicate that the proposed model can predict object deformations with a normalized root mean square error of 2.2% in the explored situations. The model can be extended for simulation of grasping deformable objects.

Acknowledgement

First of all, I would like to thank my daily supervisor dr. ir. W. Roozing for his weekly feedback and guidance during my thesis to keep me in the right direction. Furthermore, I want to thank the members of the graduation committee prof. dr. ir. S. Stramigioli, dr. ir. J.J. de Jong and A. Brugnoli Ph.D for providing me feedback and asking questions. Next, I would like to thank the technical staff of the Robotics and Mechatronics department and in particular ing. S. Smits for helping me design and manufacture the experimental set-up and B. Okken for creating and lending me his silicone objects for my experiments. Last but not least, my thanks go to the members of FlexCRAFT and Marel for the discussions and their point of view.

Contents

1 Introduction

1.1 Context

This project is part of the FlexCRAFT program (FlexCRAFT, 2021), a collaboration between Dutch universities and companies, which aim is to research the handling of varying and deformable objects in the agri-food industry. This program consists of four research topics (active perception, world modelling, planning and control, and gripping and manipulation) and three use-cases (food processing, food packaging and greenhouse).

The topic of this MSc project is part of gripping and manipulation of deformable objects. Grasping and manipulation of varying and deformable objects is still a challenging problem in industrial automation and robotics. Although gripping rigid objects is very developed, research in gripping deformable objects is limited but important for further automation. Nowadays, due to the shortage of personnel, it is especially important to automate time-consuming tasks. Grasping deformable and soft objects often requires specialized grippers specifically designed for a certain task. To be more flexible and expand quickly, general grippers are needed that can be used for a wide variety of (deformable) objects and situations. For that reason, state-of-the-art soft and underactuated grippers are currently being researched.

1.2 Problem statement

Grasping and manipulation of objects requires a stable grasp to prevent slippage and pick up and manipulate the object. For rigid objects it is sufficient if the grasping force is large enough to generate enough friction. For deformable objects, however, the right amount of force needs to be used to on the one hand have a stable grasp and on the other hand prevent damaging the object. On top of that, the object will deform when a force is applied, which has to be taken into account when grasping an object.

This thesis takes a step back to understand the behaviour of a deformable object during contact with a gripper finger by creating a model and simulation. Researches on modelling deformable objects have been conducted mainly in the medical sector.

Duan et al. (2013, 2014) created a model of soft tissues for an interactive surgical simulation. The deformable organs are modelled by a volume conserved mass-spring-damper system and the connection between organs are modelled by springs with high stiffness. The interaction between the organs and surgical instruments are taken into account using a position-based method by moving points of the object directly. The simulation has then been evaluated by experiments with real porcine liver and gallbladder.

Mollemans et al. (2003) presented a soft tissue model to predict changes in the human face due to skeletal changes. This model is used in a maxillofacial surgery planning simulation to plan the operation and predict the facial changes. The modelling method used is a mass-spring system with volume preservation.

Pathmanathan et al. (2008) used the finite element method and non-linear elasticity to create a model of a patients breast. The model was used to simulate deformation of the breast shape and predict tumor location.

Previous researches mainly focused on modelling soft tissues/deformable objects, while research on the interaction between a gripper finger and the object is very limited.

1.3 Goals and approach

The goal of this project is to gain more insight in the behaviour of a deformable object when in contact with a finger. In order to accomplish this goal, first literature research has been conducted to get an overview of the research done on grasping deformable objects and modelling methods. Thereafter, a model of deformable objects and contact with their environment is proposed. This model is a mass-spring-damper model with additional constraints for volume preservation and spring deformation limits, and Hunt-Crossley (Hunt and Crossley, 1975) and stick-slip contacts. The model can be used to understand the behaviour of a deformable object when in contact with a finger. Experiments with objects made of silicone rubber are conducted to acquire data about the deformation of an object when impacted by an external fingertip. The influence of model parameters on the simulation results are explored and fitted to minimize the error.

To summarize the steps to be taken to accomplish the project goal, the following objectives have to be achieved:

- Conduct literature research on the grasping and modelling of deformable objects
- Develop a realistic deformable object model and contact model
- Acquire data from experiments to research the behaviour of real objects
- Develop a simulation to simulate the deformation of an object when in contact with a finger and compare with the experiments.

The contributions of this report are therefore as follows:

- Development of a mass-spring-damper model and simulation for deformable objects with external finger contact
- Performing experiments with deformable objects to collect data of the deformations
- Exploration of the influence of model parameters on the simulation error

1.4 Outline

Chapter 2 gives background information and previous researches about grasping and manipulation of deformable objects and modelling methods for (deformable) objects. The paper written for this thesis is shown in Chapter 3. This paper elaborates on the modelling of deformable objects using the mass-spring-damper method and contact model. Furthermore, the simulation and integration method are discussed. On top of that, the paper also explains the design of the experiment to acquire real data and the results of the experiment and simulations. Finally, Chapter 4 discusses the simulation software in a brief overview and Chapter 5 shows some additional results.

2 Background

2.1 Grasping and manipulation of deformable objects

2.1.1 Object types

Objects are often divided into four main categories in research according to their geometry (Sanchez et al., 2018):

linear
Linear objects are objects with one dimension significantly larger than the other two dimensions. This type of objects have either no compression strength, such as ropes, cables and strings, or large strain, for example elastic tubes.

cloth-like
Cloth-like objects have one dimension significantly smaller than the other two dimensions and have no compression strength. Examples of this type of objects are cloth, fabric and curtains.

planar
Planar objects are similar to cloth-like objects, but with compression strength, for instance paper.

solid
Solid objects are three-dimensional objects, for example food or sponges.

Linear, cloth-like and planar type of deformable objects are more researched than solid deformable objects. This is likely because of the computational cost and extra difficulty of the additional dimension of a solid object.

Sanchez et al. (2018) gives an extensive overview of research conducted in grasping and manipulating deformable objects of the four different object types. Linear, cloth-like and planar deformable objects will be mentioned shortly below, whilst solid objects will be more extensively discussed as this will be the focus of this thesis.

Research conducted on linear objects are most often about grasping and manipulation of ropes. Researches on tying a knot with a rope are performed by among others Yamakawa et al. (2008), Vinh et al. (2012), Kudoh et al. (2015) and Yamakawa et al. (2010).

The most researched topics with planar objects are positioning the gripper fingers on the object to get a stable grasp, for example conducted by Guo et al. (2013) and Jia et al. (2011), and manipulating the shape of the object into a desired configuration by Fanson and Patriciu (2010) and Das and Sarkar (2011).

Cloth-like objects are probably the most researched deformable object type. Examples of research are from grasping clothes (Shibata et al. (2009), Monsó et al. (2012)), to manipulating them into the desired configuration (Cusumano-Towner et al. (2011), Doumanoglou et al. (2014)), to folding the clothes (Maitin-shepard et al. (2010), Bersch et al. (2011)).

As mentioned before, in order to successfully grasp a solid deformable object with a gripper, the right amount of force is required to prevent slippage and provide a stable grasp, but without damaging the object. To accomplish this, there are roughly two main approaches. The first ap-

proach is by using a model of the object to compute the optimal gripping force and the second approach are model-free methods.

2.1.2 Model-based grasping

Models of objects can be used to determine the optimal grasping force, which is often the minimal force required to prevent slippage and grasp the object stably. Section 2.2 gives an overview of possible modelling methods for the modelling of deformable objects.

An example is the method developed by Howard and Bekey (1999), making use of a mass-spring-damper model, where a Kelvin-Voigt model (spring and damper in parallel) describes the connection, to determine the required grasping force. In this method, the grasping of the deformable object is iteratively simulated on the model with two parallel grippers. Every iteration, the grasping force is increased until the grasp is stable and the object can be lifted. This grasping was then used as data to train a neural network that estimated the required grasping force to lift an object based on the object's stiffness and damping coefficients.

Another example of model-based grasping is proposed by Lin et al. (2015). This approach makes use of a finite element method model to simulate the deformation based on the displacements of the object due to the gripper. Two gripper fingers squeezes an object, while at each time step the model is used to try to lift the object virtually. When the object can be lifted in simulation, the fingers can stop moving and the real object can be lifted.

2.1.3 Model-free grasping

Model-free methods for grasping of deformable objects mainly uses sensors on the robot to get information about the grasp. These are often tactile, force/pressure or vision/optical sensors. Mostly, the sensors are used to detect slippage after which the gripping force is adjusted accordingly.

Kaboli et al. (2016) developed a method for manipulating deformable objects using tactile sensors. The sensors on the fingers are able to measure the force in three directions. The force sensors are used to estimate the weight of the object to compute the required grasping force by measuring the force in vertical direction. Besides, the sensors are also used to detect slip by measuring movement at the contact points. When slip has been detected, more gripping force is needed.

Other researchers using sensors also use similar methods to increase the grasping force when slip has been detected (Al-Mohammed et al., 2018). Force sensors, which detect slip by monitoring changes of the force, are also used by Hasegawa et al. (2010) and Gunji et al. (2008). Engeberg and Meek (2013) developed a method to detect slippage by measuring high frequency vibrations with vibration sensors, while slip detection with optical sensors are researched by Sani and Meek (2011) and Roberts et al. (2011).

Another approach was conducted by Delgado et al. (2015), who used a robotic hand that consisted of multiple fingers equipped with tactile sensors. This robotic hand was used to grasp and hold a deformable object. They computed the so called deformability ratio, which is a linear relationship between the forces measured by the tactile sensors and the positions of the fingers. With the deformability ratio, the force to be exerted on the object will then be calculated to minimize the deformation of the object.

2.2 Modelling deformable objects

This section describes a few methods that can be used for the modelling of deformable objects.

2.2.1 Mass-Spring-Damper System

A Mass-Spring-Damper System (MSDS) (Zhang et al., 2018; Basloom, 2016; Golec, 2018) represents an object by discretizing the shape into a set of particles in a lattice structure, which are connected by springs and dampers. Each particle is a point-mass on which the spring forces and external forces act. A mass-spring-damper system is relatively simple to implement and has a low computational cost. However, the low cost is in exchange for accuracy. This method is widely used for the simulation of soft tissue deformations for surgical training because the method can be solved real-time.

2.2.2 Position Based Dynamics

The approach of Position-Based Dynamics (Golec, 2018; Müller et al., 2006) is comparable to the mass-spring-damper system, since they are both particle-based methods. The object is also discretized into particles, but the particles are, in contrast to the MSDS, not linked by springs. The positions of the particles are manipulated directly and constraints are used to limit the positions. Stretch and bend constraints can for example be used to control the stretching and bending of the object. Next to that, volume control can be achieved by adding a volume constraint. The physics of this method, however, is not accurate since it is not based on a physical model and can only be used visually, for example in the gaming industry.

2.2.3 Finite Element Method

The Finite Element Method (FEM) (Basloom, 2016; Golec, 2018; Zhang et al., 2018) is a popular method for performing simulations in the science and physics field. The method is widely used for solving continuum mechanics problems and can handle complex geometries, for example for analysis of structures, but also for heat transfer or fluids and many more. FEM discretizes the object into a number of elements of finite size and solves partial differential equations by computing approximations of the real solutions with high accuracy. The disadvantage of the Finite Element Method is that it is very computationally expensive.

2.2.4 Boundary Element Method

The Boundary Element Method (BEM) (Basloom, 2016) discretizes only the boundary of an object, in comparison to FEM, into a set of elements. Since only equations on the surface of the object have to be solved, this method is obviously less computational expensive. BEM is thus also only accurate on the boundary of the object.

2.2.5 Choice

The model will require sufficient accuracy to be useful, causing Position-Based Dynamics to be dropped, since that method is not based on material properties. Also Boundary Element Method will not be used because only results at the boundary is present. The remaining methods are Finite Element Method and Mass-Spring-Damper System. They are both good options for the modelling of deformable objects, where FEM is more accurate but is computationally expensive, and MSDS is faster but less accurate. The model will initially be used to simulate the deformation of objects when in contact with a finger, and later for simulation of grasping and eventually for real-time determining the required gripping force, grasping position or manipulation of the object. With that in mind, MSDS will be the best option for this purpose, considering the potentially shorter simulation time. On top of that, MSDS is simpler to understand and model, with a clearer understanding of what is happening.

3 Paper

Deformation Modelling and Gripper Contact Simulation of Deformable Objects

Jordi Luong
Robotics and Mechatronics
University of Twente
j.luong@student.utwente.nl

***Abstract*—**
Grasping and manipulation of deformable objects is still a challenging problem in industrial automation and robotics. To grasp a deformable object the right amount of force needs to be used to on the one hand have a stable grasp and on the other hand prevent damaging the object. Besides, the object will deform when in contact with a grasping finger. Therefore, the behaviour of a deformable object and its contact with a finger needs to be understood. This paper proposes a mass-spring-damper model with additional constraints for volume preservation and spring deformation limits, and Hunt-Crossley [9] and stick-slip contacts. Experiments with objects made of silicone rubber are conducted to acquire data about the deformation of an object when impacted by an external fingertip. The influence of model parameters on the simulation results are explored and fitted to minimize the error. The results indicate that the proposed model can predict object deformations with a normalized root mean square error of 2.2% in the explored situations.

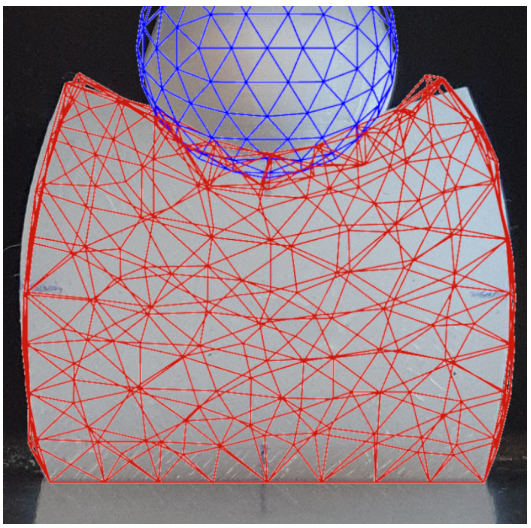


Fig. 1: Simulation result

I. Introduction

Grasping and manipulation of varying and deformable objects is still a challenging problem in industrial automation and robotics. Although gripping rigid objects is very developed, research in gripping deformable objects is limited but important for further automation.

Grasping and manipulation of objects requires a stable grasp to prevent slippage and pick up and manipulate the object. For rigid objects it is sufficient if the grasping force is large enough to generate enough friction. For deformable objects, however, the right amount of force needs to be used to on the one hand have a stable grasp and on the other hand prevent damaging the object. On top of that, the object will deform when a force is applied, which has to be taken into account when grasping an object.

This paper takes a step back to understand the behaviour of a deformable object and its contact with a finger by creating a model. Researches on modelling deformable objects have been conducted mainly in the medical sector. Duan et al. [6], [7] created a model of soft tissues for an interactive surgical simulation. The deformable organs are modelled by a volume conserved mass-spring-damper system and the connection between organs are modelled by springs with high stiffness. The interaction between the organs and surgical instruments are taken into account using a position-based method by moving points of the object directly. The simulation has then been evaluated by experiments with real porcine liver and gallbladder.

Mollemans et al. [12] presented a soft tissue model to predict changes in the human face due to skeletal changes. This model is used in a maxillofacial surgery planning simulation to plan the operation and predict the facial changes. The modelling method used is a mass-spring system with volume preservation.

Pathmanathan et al. [15] used the finite element method and non-linear elasticity to create a model of a patients breast. The model was used to simulate deformation of the breast shape and predict tumor location.

Previous researches mainly focused on modelling soft tissues/deformable objects, while research on the interaction between a gripper finger and the object is very limited. This paper develops a model of deformable objects and contact with their environment. This model is a mass-spring-damper model with additional constraints for volume preservation and spring deformation limits, and Hunt-Crossley [9] and stick-slip contacts. The model can be used to understand the behaviour of a deformable object when in contact with a finger. Experiments with objects made of silicone rubber are conducted to acquire data about the deformation of an object when impacted by an external fingertip. The influence

of model parameters on the simulation results are explored and fitted to minimize the error. The results indicate that the proposed model can predict object deformations with a normalized root mean square error of 2.2% in the explored situations. Fig. 1 shows a comparison of the simulation result of a finger compressing a deformable object with the experiment. The contributions of this paper are as follows:

- Development of a mass-spring-damper model and simulation for deformable objects with external finger contact
- Performing experiments with deformable objects to collect data of the deformations
- Exploration of the influence of model parameters on the simulation error

Sec. II will give background information about modelling deformable objects by giving a short comparison of a mass spring damper system and finite element method as modelling method. Sec. III explains the modelling of deformable objects using the mass-spring-damper method and Sec. IV discusses the contact model. The simulation and integration method are elaborated in Sec. V. The design of the experiment to acquire real data will be explained in Sec. VI and the results of the experiment and simulations and their comparisons are discussed in Sec. VII. This paper will finally be concluded with the conclusion (Sec. VIII) and recommendations (Sec. IX).

II. Background

Two widely used modelling methods for continuously deformable objects are the Mass-Spring-Damper System (MSDS) and Finite Element Method (FEM).

A Mass-Spring-Damper System [3], [8], [21] represents an object by discretizing the shape into a set of particles in a lattice structure, which are connected by springs. Each particle is a point-mass on which the spring forces and external forces act. A mass-spring-damper system is relatively simple to implement and has a low computational cost. However, the low cost is in exchange for accuracy. This method is widely used for the simulation of soft tissue deformations for surgical training because the method can be solved real-time.

The Finite Element Method [3], [8], [21] is a popular method for performing simulations in the science and physics field. The method is widely used for solving continuum mechanics problems and can handle complex geometries, for example for analysis of structures, but also for heat transfer or fluids and many more. FEM discretizes the object into a number of elements of finite size and solves partial differential equations by computing approximations of the real solutions with high accuracy. The disadvantage of the Finite Element Method is that it is very computationally expensive.

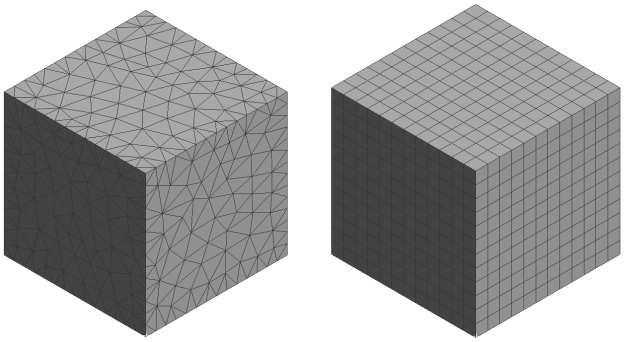
III. Object Model

This section elaborates the modelling of the deformable object using the mass-spring-damper system. Sec. III-A first introduces meshing methods to discretize the object, after which the structure and forces of a mass-spring-damper system are elaborated in Sec. III-B. Thereafter, model parameters that are derived from physical material properties are discussed in Sec. III-C. Finally, additional constraints for the model are introduced in Sec. III-D to alter the behaviour of the model.

A. Meshing

For the modelling of an object using a mass-spring-damper system, the object has to be discretized using a meshing method. In 3D, the most common meshes are tetrahedral (Fig. 2a) and hexahedral (Fig. 2b) meshes [8].

A tetrahedral mesh is more suitable for representing complex objects than a hexahedral mesh. However, a hexahedral mesh gives more accurate results compared to a tetrahedral mesh with the same resolution [11].



(a) Tetrahedral mesh (b) Hexahedral mesh

Fig. 2: 3D meshes

In this work we choose to use tetrahedral meshing, as being able to mesh complex objects will be important for further research.

Fig. 3 introduces the naming of the different parts of a tetrahedral mesh. The geometrical name is shown in regular font and its representation in the model is given in bold face.

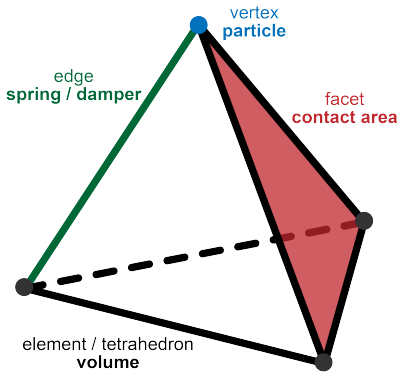


Fig. 3: Nomenclature

B. Mass-spring-damper system

Using a meshing method, the object is discretized in a set of particles p_i, with $i \in \mathbb{N}$, of mass m_i and position $\boldsymbol{x}_i \in \mathbb{R}^3$. The particles are connected with each other by springs with a stiffness constant k, initial length l_0, and in parallel a damper with damping constant d. Fig. 4 visualizes the mass-spring-damper structure of a single tetrahedron.

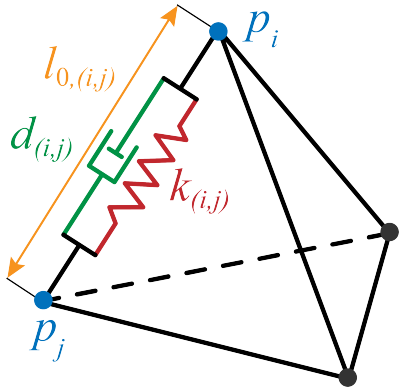


Fig. 4: Mass-spring-damper structure of a tetrahedron

The force $\boldsymbol{f}_i \in \mathbb{R}^3$ acting on the particle and the relation between the acceleration of the particle and the force is:

$$m_i \ddot{\boldsymbol{x}}_i = \boldsymbol{f}_i, \tag{1}$$

where m_i is the mass of particle p_i and $\ddot{\boldsymbol{x}}_i$ is the second time derivative of the position of particle p_i.

The force that acts on the particle is a summation of all the internal spring and damping forces originating from the spring between p_i and p_j, gravitation force and external forces:

$$\boldsymbol{f}_i = \boldsymbol{f}_{\mathrm{s},i} + \boldsymbol{f}_{\mathrm{d},i} + \boldsymbol{f}_{\mathrm{g},i} + \sum \boldsymbol{f}_{\mathrm{e},i} \tag{2}$$

1) Elastic spring forces: The spring force acting on a particle p_i is the summation of spring forces originating from springs that connect p_i to other particles. The spring force is linearly proportional to the elongation of the spring and is given by Hooke's law [7], [6]:

$$\boldsymbol{f}_{\mathrm{s},i} = \sum_j k_{(i,j)}(\|\boldsymbol{x}_j - \boldsymbol{x}_i\| - l_{0,(i,j)}) \cdot \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}, \tag{3}$$

where $k_{(i,j)}$ is the spring stiffness and $l_{0,(i,j)}$ the initial length of the spring connecting p_i and p_j. The value of the spring stiffness will be described in Sec. III-C.2.

2) Damping forces: The damping forces are computed using the viscous damping of the springs and these are given by [6], [7]:

$$\boldsymbol{f}_{\mathrm{d},i} = \sum_j d_{(i,j)} \frac{(\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i)}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \cdot \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}, \tag{4}$$

where $\boldsymbol{v}$ is the velocity of the particle and $d_{(i,j)}$ the damping constant of the damper connecting p_i and p_j. The damping constant is given in Sec. III-C.3.

3) Gravitation: The gravitation force is acting on every particle of the mass-spring-damper system and is given by

$$\boldsymbol{f}_{\mathrm{g},i} = m_i \boldsymbol{g}, \tag{5}$$

where $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}$ is the gravitational acceleration.

4) External forces: External forces, $\boldsymbol{f}_{\mathrm{e},i}$, are forces that are applied on the object. This includes for example gripping forces that are applied by grippers grasping the object or contact forces of the environment.

5) Implementation: Alg. 1 shows the computation of the spring and damping forces. The spring force acting on particle p_i is also applied on particle p_j but negated.

Algorithm 1 Compute spring and damping forces

for each spring connecting particle i and j **do**
$\quad \boldsymbol{f}_{\mathrm{s}} \leftarrow k_{(i,j)}(\|\boldsymbol{x}_j - \boldsymbol{x}_i\| - l_{0,(i,j)}) \cdot \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}$
$\quad \boldsymbol{f}_{\mathrm{d}} \leftarrow d_{(i,j)} \frac{(\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i)}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \cdot \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}$

$\quad \boldsymbol{f}_i \leftarrow \boldsymbol{f}_i + \boldsymbol{f}_{\mathrm{s}} + \boldsymbol{f}_{\mathrm{d}}$
$\quad \boldsymbol{f}_j \leftarrow \boldsymbol{f}_j - \boldsymbol{f}_{\mathrm{s}} - \boldsymbol{f}_{\mathrm{d}}$
end for

C. Physical parameters

A model contains many variables and parameters that need to be chosen. A number of these are based on material or geometric properties. This section describes the parameters that are derived from the object's material and meshing. Deriving good physical parameters of the mass-spring-damper system is required in order to get a useful model with sufficient accuracy. The parameters that have to be derived are the masses of the particles and the stiffness and damping coefficients of the springs and dampers respectively.

1) Mass: The most common method to calculate the point masses of the particles is by dividing the mass of the object over the particles [6], [7], [8]. The mass of each tetrahedron is calculated from its volume and the objects mass density and assume it is equally distributed among its four vertices. The mass m_i of particle p_i is then given by

$$m_i = \sum_{\forall j \in \mathcal{T}_i} \frac{1}{4} \rho V_j, \tag{6}$$

where $\mathcal{T}_i$ is the union of all tetrahedra that contain particle p_i, ρ the mass density of the object and V_j the undeformed volume of tetrahedron j.

2) Stiffness: Obtaining the stiffness of the springs of an object can be done in mainly two ways: parameter identification using experimental data, or analytically, based on physical properties of the object. The parameter identification approach is adjusting the stiffness of the springs according to data acquired from experiments. Golec [8] created an overview of methods and examples

of data-driven parameter identification. This approach will optimize for a specific object under specific circumstances, hence this limits the robustness of a simulation and small changes might change the behaviour. Analytical derivation of stiffness results from object properties such as elasticity and viscosity. Golec [8] gives an overview of different methods to determine the stiffness.

In this work we use the method of Lloyd [10] for calculating the spring stiffnesses. For an object divided into tetrahedra, the spring stiffness of the spring connecting particle p_i and p_j is determined by

$$k_{(i,j)} = \sum_{e \in \mathcal{T}_{(i,j)}} \frac{2\sqrt{2}}{25} l_e E, \tag{7}$$

where $\mathcal{T}_{(i,j)}$ is the set of tetrahedron elements that contain edge (i, j), E the Young's modulus of the object's material and l_e the equivalent edge length of the tetrahedron element given by

$$l_e = (V_e \frac{12}{\sqrt{2}})^{\frac{1}{3}}. \tag{8}$$

The equivalent edge length l_e is required for non regular tetrahedrons (edges are of different length) [6], [7], [10].

3) Damping: The damping can be determined by choosing a damping ratio ζ. The damper can be undamped ($\zeta = 0$), underdamped ($0 < \zeta < 1$), critically damped ($\zeta = 1$) or overdamped ($\zeta > 1$). For a critically damped system and assuming that the system can be transformed into uncoupled single degree of freedom systems, the damping coefficient of the spring connecting p_i and p_j is given by

$$d_{(i,j)} = 2 \frac{\sqrt{k_{(i,j)}(m_i + m_j)}}{l_0}, \tag{9}$$

where $k_{(i,j)}$ is the stiffness, m_i and m_j the mass of p_i and p_j respectively and l_0 the initial length of the spring [8], [7], [6], [14]. The damping coefficient is inversely proportional to the initial length of the spring to ensure similar behaviour for different meshing resolutions [14].

D. Additional constraints

The described linear mass-spring-damper model will be accurate for small deformations but will lack accuracy for larger deformations, which is important for a soft deformable object, due to non-linearity. To change the behaviour of the elongation and compression of springs for larger deformations, spring constraints are introduced in Sec. III-D.2. Another additional model constraint that is applied is an approach to preserve the volume of an object, as discussed in Sec. III-D.1.

1) Volume preservation: Many deformable objects can be considered incompressible, meaning that the volume of the object remains constant when deformation occurs.

For this model the volume preservation method of Mollemans et al. [12] is implemented. This method attempts to preserve the volume by adding a correction force to the particles. The force $\boldsymbol{f}_i$ on particle p_i now becomes

$$\boldsymbol{f}_i = \boldsymbol{f}_{\mathrm{s},i} + \boldsymbol{f}_{\mathrm{d},i} + \boldsymbol{f}_{\mathrm{g},i} + \sum \boldsymbol{f}_{\mathrm{e},i} + \alpha \boldsymbol{f}_{\mathrm{vol},i}, \tag{10}$$

where $\boldsymbol{f}_{\mathrm{vol},i}$ is the correction force and α a scaling constant of which the influence is investigated in Sec. VII-C.3.

The method attempts to compensate for the volume difference by adding a correction force on the four particles of the tetrahedron. Thereby the particles are pushed away from or pulled towards the barycenter of the tetrahedron and the volume of the element will increase or decrease respectively. The force acting on particle p_i is given by [8], [12]

$$\boldsymbol{f}_{\mathrm{vol},i} = \sum_{\forall j \in \mathcal{T}_i} (V_j - V_j^0) \frac{\boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{b}_j}}{\left\| \boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{b}_j} \right\|}, \tag{11}$$

where $\mathcal{T}_i$ is the union of all tetrahedra that contain particle p_i, V_j and V_j^0 the current and initial volume of the tetrahedron respectively, $\boldsymbol{x}_i$ the position of particle p_i and the barycenter of the tetrahedron

$$\boldsymbol{x}_{\mathrm{b}_j} = \frac{1}{4} \sum_{i=0}^{4} \boldsymbol{x}_i. \tag{12}$$

2) Spring constraints: Spring constraints [6], [7], [16] are implemented to correct the position of particles after integration as a post-processing step. These spring constraints introduce non-linearity to the mass-spring-damper system by limiting the elongation or compression of springs for larger deformations.

When large forces are concentrated on the object, the problem of large local deformation might occur, whereby a small region of the object is overly and not realistically stretched or compressed. By introducing constraints for the spring length, this behaviour can be corrected. By limiting the stretching and compression of the springs, the deformation can be extended from a local region to a larger region.

The idea is to find for each particle p_i the mean of the corrections $\Delta \boldsymbol{x}_i$ which satisfy the constraint functions and update the integrated position by

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \frac{1}{n_{\mathrm{constraints}}} \sum_{\epsilon_i} \Delta \boldsymbol{x}_i, \tag{13}$$

where ϵ_i is the set of edges containing particle p_i and $n_{\mathrm{constraints}}$ the number of edges in ϵ_i.

For the over-stretching constraint a parameter τ_{s} is set. When the spring length exceeds $(1 + \tau_{\mathrm{s}})l_0$, where l_0 is the initial length of the spring, the constraint tries to compress the spring again to its maximum length.

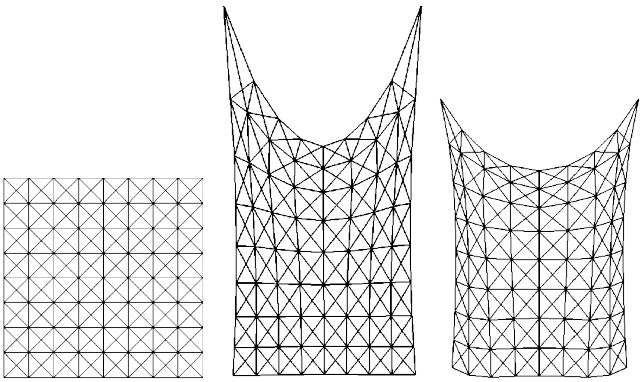
The constraint is given by

$$C_{\mathrm{stretch}} = (1 + \tau_{\mathrm{s}})l_0 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \geq 0. \tag{14}$$

If $C_{\text{stretch}} < 0$, the spring length is larger than the maximum length and the corrections $\Delta \boldsymbol{x}_i$ and $\Delta \boldsymbol{x}_j$ for particles p_i and p_j respectively are computed by

$$\begin{aligned} \Delta \boldsymbol{x}_i &= \frac{1}{2}\, C_{\text{stretch}}\, \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} \\ \Delta \boldsymbol{x}_j &= -\frac{1}{2}\, C_{\text{stretch}}\, \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}. \end{aligned} \tag{15}$$

Fig. 5 illustrates an example of the influence of the over-stretching constraint on the deformation by hanging an object on its top two corners while subject to gravity. Fig. 5a shows the initial mesh and Fig. 5b shows the result when no over-stretching constraint is applied ($\tau_s = \infty$). In this case, large deformation occurs at the corners of the object. In Fig. 5c $\tau_s = 0.1$, where it is clear that the springs at the corners are less deformed and the deformation occurs more globally.



(a) Initial mesh (b) $\tau_s = \infty$ (c) $\tau_s = 0.1$

Fig. 5: Over-stretching constraint [7]

The over-compressing constraint is comparable to the over-stretching constraint with a parameter τ_{c}. When the spring length is less than $(1 - \tau_{\text{c}})l_0$, the constraint will try to elongate the spring back to the minimum length.

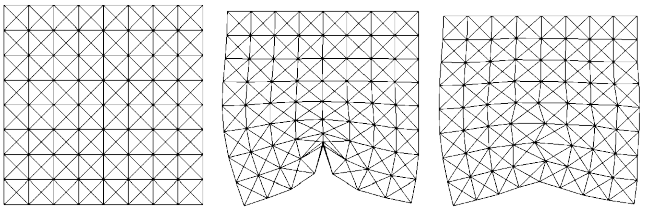
This constraint is given by

$$C_{\text{compress}} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| - (1 - \tau_{\text{c}})l_0 \geq 0. \tag{16}$$

If $C_{\text{compress}} < 0$, the spring length is smaller than the minimum length and the corrections $\Delta \boldsymbol{x}_i$ and $\Delta \boldsymbol{x}_j$ are computed using

$$\begin{aligned} \Delta \boldsymbol{x}_i &= -\frac{1}{2}\, C_{\text{compress}}\, \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} \\ \Delta \boldsymbol{x}_j &= \frac{1}{2}\, C_{\text{compress}}\, \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}. \end{aligned} \tag{17}$$

Fig. 6a depicts the influence of the over-compression constraint with an example. The object is fixed at the top and at the bottom an external force is applied in the center. When no over-compression constraint is applied ($\tau_c = \infty$) the deformation of the object is large and local as shown in Fig. 6b. By applying the over-compression constraint with $\tau_c = 0.1$ the deformation is reduced and more globally.



(a) Initial mesh (b) $\tau_c = \infty$ (c) $\tau_c = 0.1$

Fig. 6: Over-compression constraint [7]

The spring constraint parameters τ_s and τ_c have a significant influence on the deformation properties of the object. The impact of these parameters are explored in Sec. VII-C.2 and Sec. VII-C.1.

3) Implementation: Alg. 2 shows the implementation of the volume preservation correction forces and Alg. 3 the implementation of the spring constraints.

Algorithm 2 Compute volume preservation forces

for each element j **do**
  $V_j \leftarrow \frac{1}{6} \det\left(\begin{bmatrix} \boldsymbol{x}_1 - \boldsymbol{x}_4 & \boldsymbol{x}_2 - \boldsymbol{x}_4 & \boldsymbol{x}_3 - \boldsymbol{x}_4 \end{bmatrix}\right)$
  for each particle i in j **do**
    $\boldsymbol{x}_{\text{b}_j} \leftarrow \boldsymbol{x}_{\text{b}_j} + \frac{1}{4}\boldsymbol{x}_i$
  end for

  for each particle i in j **do**
    $\boldsymbol{f}_i \leftarrow \boldsymbol{f}_i + \alpha(V_j - V_j^0)\frac{\boldsymbol{x}_i - \boldsymbol{x}_{\text{b}_j}}{\|\boldsymbol{x}_i - \boldsymbol{x}_{\text{b}_j}\|}$
  end for
end for

IV. Contact Model

For the contact model, the interacting object is simplified and modelled as a sphere with center position $\boldsymbol{x}_{\text{f}}$ and radius R_{f}. The surface of the deformable object is divided into facets (planar triangles) originating from the tetrahedra on the boundary. The contact model consists of four parts. First, for each facet within range of the finger, it is determined whether there is contact between the finger and facet and what the contact position then would be. When there is contact, the normal and tangential forces are computed. Finally, the total contact force is divided into nodal forces acting on the three particles of the facet.

A. Contact detection

The detection of contact between the finger and facet is done in two ways. First it is checked if the finger is in contact with the face of the facet. When that is not the case, contact on the edge of the facet is considered. If there is also no edge contact, the finger is not in contact with the facet.

Algorithm 3 Apply spring constraints

for number of iterations **do**
  $\Delta \boldsymbol{x}_i \leftarrow \mathbf{0}$
  $\Delta \boldsymbol{x}_j \leftarrow \mathbf{0}$

  for each spring connecting particle i and j **do**
    $C_{\text{stretch}} \leftarrow (1+\tau_{\text{s}})l_0 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$
    if $C_{\text{stretch}} < 0$ **then**
      $\Delta \boldsymbol{x}_i \leftarrow \Delta \boldsymbol{x}_i + \frac{1}{2} C_{\text{stretch}} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$
      $\Delta \boldsymbol{x}_j \leftarrow \Delta \boldsymbol{x}_j - \frac{1}{2} C_{\text{stretch}} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$
    end if

    $C_{\text{compress}} \leftarrow \|\boldsymbol{x}_i - \boldsymbol{x}_j\| - (1-\tau_{\text{c}})l_0$
    if $C_{\text{compress}} < 0$ **then**
      $\Delta \boldsymbol{x}_i \leftarrow \Delta \boldsymbol{x}_i - \frac{1}{2} C_{\text{compress}} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$
      $\Delta \boldsymbol{x}_j \leftarrow \Delta \boldsymbol{x}_j + \frac{1}{2} C_{\text{compress}} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$
    end if
  end for

  for each particle i **do**
    $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \frac{1}{n} \Delta \boldsymbol{x}_i$
  end for
end for

1) Contact on facet: The plane of the facet is given by

$$Ax + By + Cz + D = 0. \tag{18}$$

The normal vector of that plane is

$$\boldsymbol{n} = \begin{bmatrix} A & B & C \end{bmatrix}^{\text{T}}. \tag{19}$$

The equation for the sphere representing the finger f is

$$(x - x_{\text{f}})^2 + (y - y_{\text{f}})^2 + (z - z_{\text{f}})^2 = R_{\text{f}}^2, \tag{20}$$

where x_{f} the x-position and R_{f} the radius of the finger.

The normal vector n passing through the center of the finger is given by the following parametric line equation [1]:

$$L = x_{\text{f}} + At, y_{\text{f}} + Bt, z_{\text{f}} + Ct. \tag{21}$$

The intersection between the line and plane is then found by first substituting Eq. 21 into Eq. 18, resulting in

$$A(x_{\text{f}} + At) + B(y_{\text{f}} + Bt) + C(z_{\text{f}} + Ct) + D = 0. \tag{22}$$

By eliminating t the following expression is found:

$$t = -\frac{Ax_{\text{f}} + By_{\text{f}} + Cz_{\text{f}} + D}{A^2 + B^2 + C^2}. \tag{23}$$

The intersection point on the plane is then

$$\boldsymbol{x}_{\mathbf{c}} = \begin{bmatrix} x_{\text{f}} + At \\ y_{\text{f}} + Bt \\ z_{\text{f}} + Ct \end{bmatrix}. \tag{24}$$

Thereafter, it is checked whether the intersection point lies within the facet and the distance between the finger center and the intersection point is smaller than the finger radius. If these conditions are met, the intersection point is also the contact point between the finger and facet.

2) Contact on edge: To determine the possible contact point on an edge of the facet, the projection with the smallest distance of the finger position $\boldsymbol{x}_{\text{f}}$ onto each of the three edges of the facet are computed with the following formula [13]:

$$\boldsymbol{x}_{\text{c}} = \boldsymbol{x}_{\text{A}} + \frac{(\boldsymbol{x}_{\text{f}} - \boldsymbol{x}_{\text{A}}) \cdot (\boldsymbol{x}_{\text{B}} - \boldsymbol{x}_{\text{A}})}{(\boldsymbol{x}_{\text{B}} - \boldsymbol{x}_{\text{A}}) \cdot (\boldsymbol{x}_{\text{B}} - \boldsymbol{x}_{\text{A}})} (\boldsymbol{x}_{\text{B}} - \boldsymbol{x}_{\text{A}}), \tag{25}$$

where A, B and C are the vertices of the facet.

When the distance between the closest projection and the finger center is smaller than the finger radius, that projection point is also the contact point of the finger on the edge. The normal vector n is then given by the vector between the finger center position and the contact point.

B. Normal force model

The method used for the computation of the normal forces is the Hunt-Crossley model [9]. We provide a brief overview here.

The normal force [5], [19] of a facet j when in contact with a finger is given by

$$\boldsymbol{f}_{\text{n},j} = K\delta_j^n \left(1 + \frac{3(1-\epsilon)}{2} \frac{\dot{\delta}_j}{\dot{\delta}_{0,j}}\right) \boldsymbol{n}_j, \tag{26}$$

where K is the contact stiffness depending on material properties and shape of the finger and object, ϵ the coefficient of restitution, δ_j the penetration depth of the finger into the facet, $\dot{\delta}_{0,j}$ the relative normal velocity between the finger and facet when the contact was initially detected and $\boldsymbol{n}_j$ the direction of the normal force.

For the contact between a sphere (finger) and plane (facet of the object), the contact stiffness K is given by

$$K = \frac{4}{3(\sigma_{\text{f}} + \sigma_{\text{o}})} \sqrt{R_{\text{f}}}, \tag{27}$$

where R_{f} is the radius of the spherical finger and σ_{f} and σ_{o} are expressed by

$$\sigma_{\text{f}} = \frac{1 - \nu_{\text{f}}^2}{E_{\text{f}}}, \quad \sigma_{\text{o}} = \frac{1 - \nu_{\text{o}}^2}{E_{\text{o}}}, \tag{28}$$

where E_{f} and E_{o} are the Young's modulus of the finger and object respectively, and ν_{f} and ν_{o} the Poisson's ratio.

The penetration depth δ_j of the finger into facet j is

$$\delta_j = R_f - \|\boldsymbol{x}_f - \boldsymbol{x}_{\text{c},j}\|, \tag{29}$$

where R_f the finger radius, $\boldsymbol{x}_f$ the finger position and $\boldsymbol{x}_{\text{c},j}$ the contact position. Fig. 7 shows a schematic side view of the finger in contact with a facet.

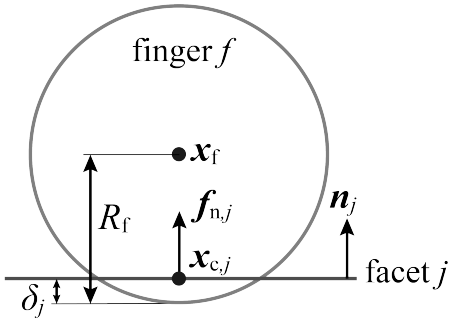


Fig. 7: Side view of the finger in contact with a facet

C. Tangential force model

The tangential force is formulated using a stick-slip model [5], [19] with the sticking force $\boldsymbol{f}_{\mathrm{stick}}$ at low velocities and slipping force $\boldsymbol{f}_{\mathrm{slip}}$ at high velocities.

A transition function is used for a smooth transition between the two forces as given by the following formula for the tangential force:

$$\boldsymbol{f}_{\mathrm{t},j} = S\boldsymbol{f}_{\mathrm{stick},j} + (1-S)\boldsymbol{f}_{\mathrm{slip},j}, \tag{30}$$

where S is the transition function

$$S = e^{-(\boldsymbol{v}_{\mathrm{t},j}^{\mathrm{T}}\boldsymbol{v}_{\mathrm{t},j})/v_{\mathrm{stick}}^2}, \tag{31}$$

where $\boldsymbol{v}_{\mathrm{t},j}$ is the tangential velocity and v_{stick} the velocity of the transition between stick and slip. The transition function S has the following behaviour:

$$S = \begin{cases} 0 & \|\boldsymbol{v}_{\mathrm{t},j}\| \gg v_{\mathrm{stick}} \\ 1 & \|\boldsymbol{v}_{\mathrm{t},j}\| = 0 \end{cases} \tag{32}$$

The tangential velocity is derived as

$$\boldsymbol{v}_{\mathrm{t},j} = \dot{\boldsymbol{x}}_{\mathrm{c},j} - (\boldsymbol{n}_j^T \cdot \dot{\boldsymbol{x}}_{\mathrm{c},j})\boldsymbol{n}_j, \tag{33}$$

where $\dot{\boldsymbol{x}}_{\mathrm{c},j}$ the velocity of the contact point and $\boldsymbol{n}_j$ the normal vector at the contact point.

The slipping force is given by Coulomb's law of friction

$$\boldsymbol{f}_{\mathrm{slip}} = \begin{cases} \boldsymbol{0} & \|\boldsymbol{v}_{\mathrm{t},j}\| = 0 \\ -\mu\|\boldsymbol{f}_{\mathrm{n}}\|\frac{\boldsymbol{v}_{\mathrm{t},j}}{\|\boldsymbol{v}_{\mathrm{t},j}\|} & \|\boldsymbol{v}_{\mathrm{t},j}\| > 0 \end{cases} \tag{34}$$

where μ is the friction coefficient.

The sticking force is given by

$$\boldsymbol{f}_{\mathrm{stick}} = \begin{cases} \boldsymbol{0} & d = 0 \\ \frac{f_{\mathrm{stick}}}{d}\left(\boldsymbol{I} - \boldsymbol{n}_j\boldsymbol{n}_j^{\mathrm{T}}\right)(\boldsymbol{x}_{\mathrm{c},j} - \boldsymbol{x}_{\mathrm{stick},j}) & d > 0 \end{cases} \tag{35}$$

where $\boldsymbol{x}_{\mathrm{c},j}$ the contact position, $\boldsymbol{x}_{\mathrm{stick},j}$ the initial contact position at the moment of first contact, $d = \|\boldsymbol{x}_{\mathrm{c},j} - \boldsymbol{x}_{\mathrm{stick},j}\|$ and

$$f_{\mathrm{stick}} = -k_{\mathrm{stick}}d - c_{\mathrm{stick}}\dot{d}, \tag{36}$$

where k_{stick} and c_{stick} the stiffness and damping coefficient of the stiction model.

D. Nodal forces

The total contact force on a facet is the sum of the normal force and tangential force:

$$\boldsymbol{f}_j = \boldsymbol{f}_{\mathrm{n},j} + \boldsymbol{f}_{\mathrm{t},j} \tag{37}$$

The contact force on each facet then has to be converted into nodal forces, meaning that the force needs to be distributed amongst the three particles of the facet. The nodal forces are computed using the matrix shape function H [20], which is based on the areas of the facet as shown in Fig. 8. The shape function H_j for facet j is

$$H_j = \begin{bmatrix} \frac{A_{1,j}}{A_j} & 0 & 0 & \frac{A_{2,j}}{A_j} & 0 & 0 & \frac{A_{3,j}}{A_j} & 0 & 0 \\ 0 & \frac{A_{1,j}}{A_j} & 0 & 0 & \frac{A_{2,j}}{A_j} & 0 & 0 & \frac{A_{3,j}}{A_j} & 0 \\ 0 & 0 & \frac{A_{1,j}}{A_j} & 0 & 0 & \frac{A_{2,j}}{A_j} & 0 & 0 & \frac{A_{3,j}}{A_j} \end{bmatrix} \tag{38}$$

where A_j is the area of facet j.

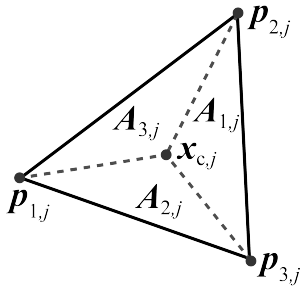


Fig. 8: Division of facet j into three areas by the contact point $\boldsymbol{x}_{\mathrm{c},j}$

The nodal forces are then calculated with

$$\begin{bmatrix} \boldsymbol{f}_{1,j} \\ \boldsymbol{f}_{2,j} \\ \boldsymbol{f}_{3,j} \end{bmatrix} = H_j^{\mathrm{T}}\boldsymbol{f}_j \tag{39}$$

E. Implementation

Alg. 4 shows the implementation of the contact model consisting of a contact detection, normal force model and tangential force model.

V. Simulation

Fig. 9 gives the simulation overview of the mass-spring-damper and contact model. First the object will be meshed for discretization in particles, as discussed in Sec. III-A. Thereafter, the simulation loop will start running. Every time step, several steps are executed. At first, the spring forces (Sec. III-B) and contact forces (Sec. IV) are computed. Subsequently, the correction forces for volume preservation are computed, as described in Sec. III-D.1. After computing the forces, the states are updated by calculating the new position and velocity of the particles and finger(s). The integration is elaborated in Sec. V-A. Next, spring constraints are applied to modify and correct the behaviour of the model as explained in Sec. III-D.

Algorithm 4 Calculate contact forces

for each facet j **do**
  for each finger f **do**
    if $\|\boldsymbol{x}_j - \boldsymbol{x}_f\| \leq cR_f$ **then**
      if contact on facet **then** ▷ facet contact
        $\boldsymbol{x}_{\mathrm{c},j} \leftarrow$ contact point on facet
        $\boldsymbol{n}_j \leftarrow$ normal of facet
      else if contact on edge **then** ▷ edge contact
        $\boldsymbol{x}_{\mathrm{c},j} \leftarrow$ contact point on edge
        $\boldsymbol{n}_j \leftarrow \frac{\boldsymbol{x}_{\mathrm{c},j} - \boldsymbol{x}_f}{\|\boldsymbol{x}_{\mathrm{c},j} - \boldsymbol{x}_f\|}$
      else ▷ no contact
        continue
      end if
      ▷ normal force
      $\delta_j \leftarrow R_f - \|\boldsymbol{x}_f - \boldsymbol{x}_{\mathrm{c},j}\|$
      if $\delta_j \leq 0$ **then**
        $\boldsymbol{f}_{\mathrm{n},j} \leftarrow \boldsymbol{0}$
      else
        $\boldsymbol{f}_{\mathrm{n},j} \leftarrow K\delta_j^n \left(1 + \frac{3(1-\epsilon)}{2} \frac{\dot{\delta}_j}{\dot{\delta}_{0,j}}\right) \boldsymbol{n}_j$
      end if
      ▷ tangential slipping force
      $\boldsymbol{v}_{\mathrm{t},j} \leftarrow \dot{\boldsymbol{x}}_{\mathrm{c},j} - (\boldsymbol{n}_j^T \cdot \dot{\boldsymbol{x}}_{\mathrm{c},j})\boldsymbol{n}_j$
      if $\|\boldsymbol{v}_{\mathrm{t},j}\| = 0$ **then**
        $\boldsymbol{f}_{\mathrm{slip}} \leftarrow \boldsymbol{0}$
      else
        $\boldsymbol{f}_{\mathrm{slip}} \leftarrow -\mu \|\boldsymbol{f}_{\mathrm{n}}\| \frac{\boldsymbol{v}_{\mathrm{t},j}}{\|\boldsymbol{v}_{\mathrm{t},j}\|}$
      end if
      ▷ tangential sticking force
      $d \leftarrow \|\boldsymbol{x}_{\mathrm{c},j} - \boldsymbol{x}_{\mathrm{stick},j}\|$
      if $d = 0$ **then**
        $\boldsymbol{f}_{\mathrm{stick}} \leftarrow \boldsymbol{0}$
      else
        $\boldsymbol{f}_{\mathrm{stick}} \leftarrow \frac{f_{\mathrm{stick}}}{d} \left(\boldsymbol{I} - \boldsymbol{n}_j \boldsymbol{n}_j^{\mathrm{T}}\right) (\boldsymbol{x}_{\mathrm{c},j} - \boldsymbol{x}_{\mathrm{stick},j})$
      end if
      ▷ tangential force
      $\boldsymbol{f}_{\mathrm{t},j} \leftarrow S\boldsymbol{f}_{\mathrm{stick},j} + (1-S)\boldsymbol{f}_{\mathrm{slip},j}$

      $\boldsymbol{f}_j = \boldsymbol{f}_{\mathrm{n},j} + \boldsymbol{f}_{\mathrm{t},j}$
      $\begin{bmatrix} \boldsymbol{f}_{j,1} & \boldsymbol{f}_{j,2} & \boldsymbol{f}_{j,3} \end{bmatrix}^{\mathrm{T}} = H^{\mathrm{T}} \boldsymbol{f}_j$
    end if
  end for
end for

A. Integration method

The mass spring damper system can be written as a system of first order differential equations:

$$\begin{cases} \dot{\boldsymbol{v}} = \frac{\boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{v})}{m}, \\ \dot{\boldsymbol{x}} = \boldsymbol{v}. \end{cases} \tag{40}$$

This system needs to be solved using an integration method.

Both explicit and implicit integration methods [8], [21] are considered. Using explicit integration, the next state is determined explicitly from the current state. The implementation of this kind of integration is simple and computationally inexpensive, since all required values are already known from the current state. The disadvantage however, is that an explicit integration is only conditionally stable and requires a very small time step, resulting in a slow simulation.

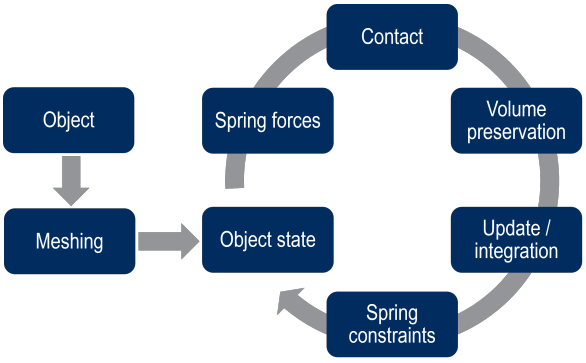


Fig. 9: Simulation overview

An implicit integration method, on the other hand, is unconditionally stable. The future state is computed from the current and future state. Therefore, it needs to solve a system of equations to acquire the next state. This results in a longer computation time for each time step, however, due to the stability of implicit integration the time step can be increased significantly.

1) Implicit Euler: The implicit Euler (backward Euler) scheme [2] is used for the integration of the system. The system of equations to be solved is then given by

$$\begin{cases} \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta x = \boldsymbol{x}_n + \Delta t\ \boldsymbol{v}_{n+1} \\ \boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \Delta v = \boldsymbol{v}_n + \Delta t\ M^{-1} \boldsymbol{f}(t_n, \boldsymbol{x}_{n+1}, \boldsymbol{v}_{n+1}) \end{cases} \tag{41}$$

where M is the mass matrix and $\boldsymbol{f}(t_n, \boldsymbol{x}_{n+1}, \boldsymbol{v}_{n+1})$ is the Taylor expansion

$$\boldsymbol{f}(t_n, \boldsymbol{x}_{n+1}, \boldsymbol{v}_{n+1}) = \boldsymbol{f}_n + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}} \Delta \boldsymbol{v}, \tag{42}$$

where $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}$ are the Jacobians.

By substituting Eq. 41 into Eq. 42 and after rewriting, the following equation is found:

$$\left(M - \Delta t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}} - \Delta t^2 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right) \Delta \boldsymbol{v} = \Delta t \left(\boldsymbol{f}_n + \Delta t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \boldsymbol{v}_n\right). \tag{43}$$

This equation is in the form of a matrix-vector system $A\Delta\boldsymbol{v} = \boldsymbol{b}$. Using the conjugate gradient method this system can be solved by approximating $\Delta\boldsymbol{v}$. Consequently, $\Delta\boldsymbol{x}$ can then also be computed and the position and velocity of the particles can be updated.

2) Conjugate gradient method: The conjugate gradient method [17] is an iterative method for solving large systems of equations of the form $A\boldsymbol{x} = \boldsymbol{b}$, where A a known square matrix, $\boldsymbol{b}$ a known vector and $\boldsymbol{x}$ the solution vector.

This method tries to approach the solution by solving the following minimization problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} A\boldsymbol{x} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}. \tag{44}$$

3) Implementation: Alg. 5 gives the implementation of integration and Alg. 6 the algorithm of the conjugate gradient method.

Algorithm 5 Update / Integrate

for each spring **do**
  compute Jacobians
end for

Solve $A\Delta\boldsymbol{v} = b$

for each particle i **do**
  $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \Delta\boldsymbol{v}_i\, dt$
  $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i\, dt$
  $\boldsymbol{f}_i \leftarrow \boldsymbol{0}$
end for

for each finger f **do**
  update finger
end for

Algorithm 6 Conjugate gradient method

$i \leftarrow 0$
$\boldsymbol{r} \leftarrow \boldsymbol{b} - A\Delta\boldsymbol{v}$
$\boldsymbol{d} \leftarrow \boldsymbol{r}$
$\delta_{\mathrm{new}} \leftarrow \boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}$
$\delta_0 \leftarrow \delta_{\mathrm{new}}$

while $i < i_{\max}$ **and** $\delta_{\mathrm{new}} > \epsilon^2\delta_0$ **do**
  $\boldsymbol{q} \leftarrow A\boldsymbol{d}$
  $\alpha \leftarrow \frac{\delta_{\mathrm{new}}}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{q}}$
  $\Delta\boldsymbol{v} \leftarrow \Delta\boldsymbol{v} + \alpha\boldsymbol{d}$
  $\boldsymbol{r} \leftarrow \boldsymbol{r} - \alpha\boldsymbol{q}$
  $\delta_{\mathrm{old}} \leftarrow \delta_{\mathrm{new}}$
  $\delta_{\mathrm{new}} \leftarrow \boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}$
  $\beta \leftarrow \frac{\delta_{\mathrm{new}}}{\delta_{\mathrm{old}}}$
  $\boldsymbol{d} \leftarrow \boldsymbol{r} + \beta\boldsymbol{d}$
  $i \leftarrow i + 1$
end while

VI. Experiment Design

Experiments are performed to validate and fit the model and simulation. The goal of the experiment is to collect data of the behaviour and deformation of deformable objects when a finger is in contact. Therefore, a set-up is designed to apply finger contact on an object with increasing force. That data can then be used for the validation and parameter fitting of the model.

A. Objects

Several objects are used for the experiment of different materials and shapes.

The following silicone rubber materials are used for the test objects:

1) Ecoflex 00-20
2) Dragonflex 20

The objects are of the following shapes:

1) cuboid, 50 x 50 x 20 mm
2) cylinder, ⌀80 x 53.3 mm

B. Experimental set-up

The experimental set-up consists of a frame and a construction that can move vertically, as shown in Fig. 10. The construction is on the one end attached to a platform for weights to be placed on and on the other end to a force sensor. The other side of the force sensor is connected to a finger holder where fingers can be easily changed. The different fingers are approximations of the fingertips of a gripper and are used for the contact with the objects. The fingers are spherical fingertips with varying radius and a flat plate as shown in Fig. 11.

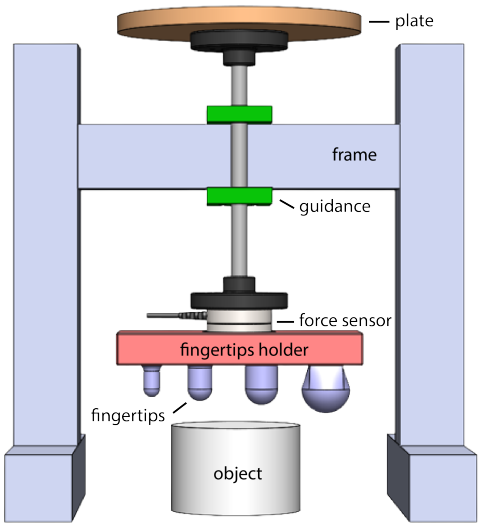


Fig. 10: Experimental set-up

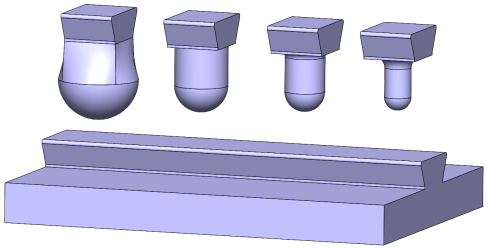


Fig. 11: Different fingers

By putting weight on the platform, a certain force is applied on the object. The force sensor is an extra measurement to make sure there is no unforeseen friction that affects the force.

Experiments are performed by increasing the weight/force and determining the deformation of the object at several points as shown in Fig. 12. Points 0 to 7 are positions on the object and point 8 is the position of the finger. The corresponding points of the simulations will then be compared to these points from the experiment to compute the simulation error. Note however that the position of the points are only evaluated in two directions and there will be no information about the depth of the points in the experiment. The experiment has been repeated for different combinations of objects and fingers.

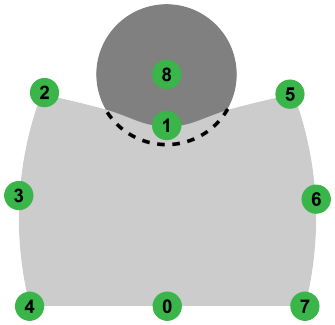


Fig. 12: Points on an object

C. Model parameters fitting

As described in Sec. III-D, a few model parameters needs to be chosen that are not derived from material or geometric properties. The parameters τ_s and τ_c define the spring constraints and limit the stretching or compression of the springs, and α is the impact of the volume preservation correction force. The main interest when choosing these parameters is the effect on the deformation of the object. Using the data from the experiments, these parameters can be fitted by minimizing the error of the model using the points as described above. The parameters of the stick-slip contact model are not explored in-depth as these will impact the friction between the object and finger, and will not influence the deformation behaviour of the object significantly.

D. Error calculation

To calculate the error of the model/simulation in comparison with the experimental data, the root mean square error (RMSE) is computed for the eight points of the object. The formula for the root mean square error is given by

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_{\mathrm{sim},i} - \boldsymbol{x}_{\mathrm{exp},i}\|^2}, \tag{45}$$

where $n = 8$ is the number of points, $\boldsymbol{x}_{\mathrm{sim},i}$ the position of point i of the simulation and $\boldsymbol{x}_{\mathrm{exp},i}$ the same point of the corresponding experiment. The root mean square error is subsequently normalized by dividing by the length of the object:

$$\text{normalized RMSE} = \frac{\mathrm{RMSE}}{l_0}. \tag{46}$$

By assuming the object is incompressible and the volume remains constant, which can be assumed for silicone rubbers if not all directions are constrained [18], the error in volume is computed by

$$\text{volume error} = |V - V_0|, \tag{47}$$

where V_0 the initial volume of the object, and the normalized volume error is

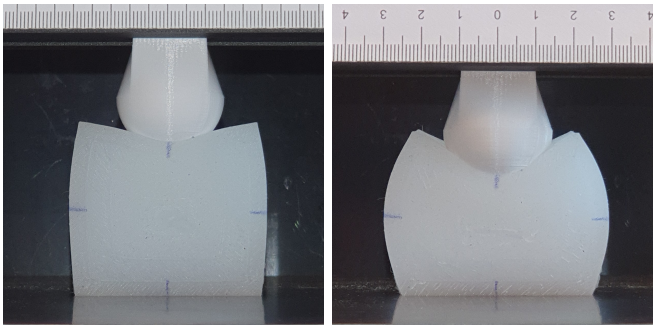
$$\text{normalized volume error} = \frac{\text{volume error}}{V_0}. \tag{48}$$

VII. Results

This section shows some results of a small selection of the experiments and simulations performed.

A. Results experiments

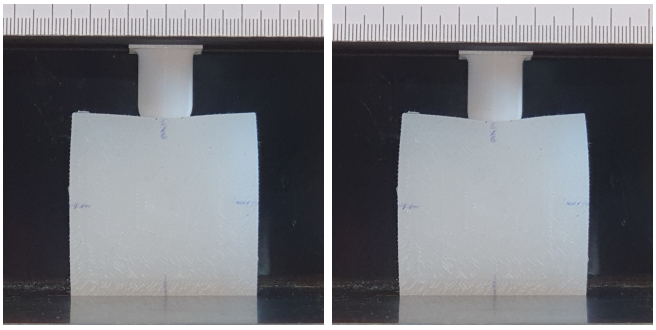
Fig. 13 shows two images of the experiment with the cuboid object of the relatively soft material Ecoflex 00-20. The object is deformed by a larger finger and the results are shown at two different weights.



(a) Weight of 553 g (b) Weight of 1460 g

Fig. 13: Cuboid Ecoflex 00-20 with large finger

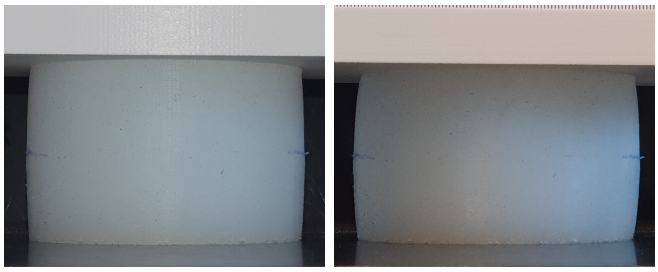
Two results of the cuboid made from the stiffer material Dragonskin 20 are shown in Fig. 14. Here the deformations with a small finger are shown.



(a) Weight of 2159 g (b) Weight of 3176 g

Fig. 14: Cuboid Dragonskin 20 with small finger

Fig. 15 shows the result at two different weights for the cylinder object of Ecoflex 00-20, compressed by a flat plate.



(a) Weight of 1634 g (b) Weight of 3347 g

Fig. 15: Cylinder Ecoflex 00-20 with plate

Since the deformation of the cuboid made of Ecoflex 00-20 in combination with the large finger gives the most interesting results, this scenario will be the main focus of this section. Fig. 16 shows the points, as defined by Fig. 12, of the object for weights ranging from 0 to 1460 grams. Note that the lines between the points are only connecting the points and are not representable for the boundary of the deformed object, as the object boundary will be curved as can be seen in Fig. 13. From Fig. 16 it is clear that for an increasing weight/force the object will be compressed by the finger on the top and the sides will bulge out as expected. It can also be seen that the top corners of the object slowly bend inwards and that the bottom points do not deform, probably due to friction.

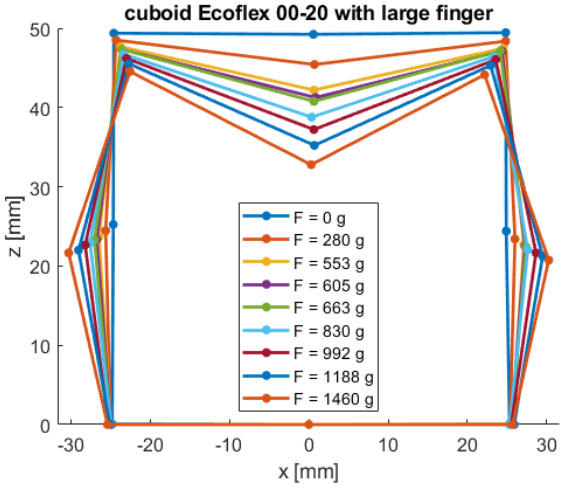


Fig. 16: Points of the object for range of weights

B. Young's modulus

For the simulation, the material property Young's modulus has to be set correctly for the object, since this will affect the stiffness of the deformable object. The Young's modulus is found by using the acquired experimental data from the experiments with the flat plate at different weights, the compression test. For each weight/force the stress σ and strain ϵ are computed which are given by

$$\sigma = \frac{F}{A}, \quad \epsilon = \frac{\Delta l}{l_0}, \tag{49}$$

where F is the force, A the area the force is applied on, Δl the change in length and l_0 the initial length of the object in the direction of the force, as shown in Fig. 17. Fig. 18 shows the stress-strain curve of the three different objects.

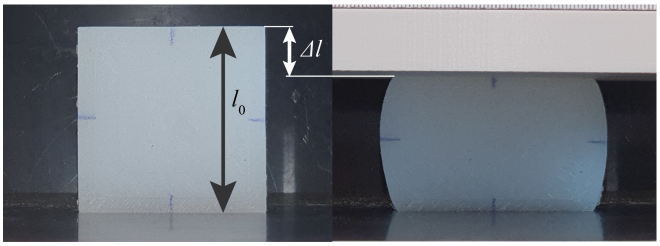


Fig. 17: Compression test

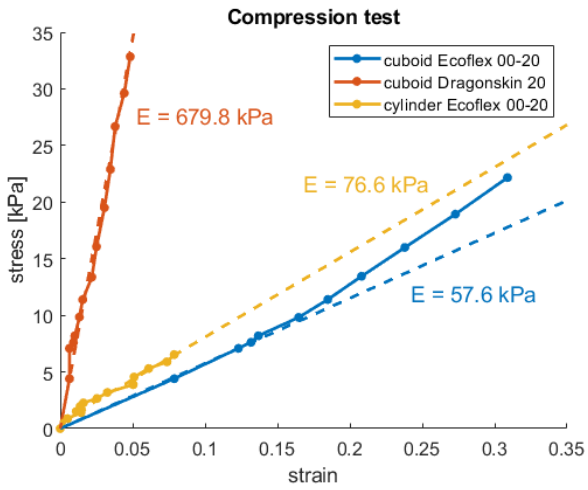


Fig. 18: Results of compression test

The Young's modulus E is then found by fitting a line through the initial linear part of the stress-strain curve (dashed lines in Fig. 18) and computing

$$E = \frac{\sigma}{\epsilon}. \tag{50}$$

Tab. I shows the found Young's moduli of the three objects using data from the compression test.

TABLE I: Young's moduli

Material	Shape	Young's modulus E
Ecoflex 00-20	cuboid	57.6 kPa
Ecoflex 00-20	cylinder	76.6 kPa
Dragonskin 20	cuboid	679.8 kPa

C. Parameters fitting

For the fitting of the model parameters, a large collection of simulation data is acquired for the situation where the Ecoflex 00-20 cuboid is deformed by the large finger with a weight of 992 grams or 9.73 N.

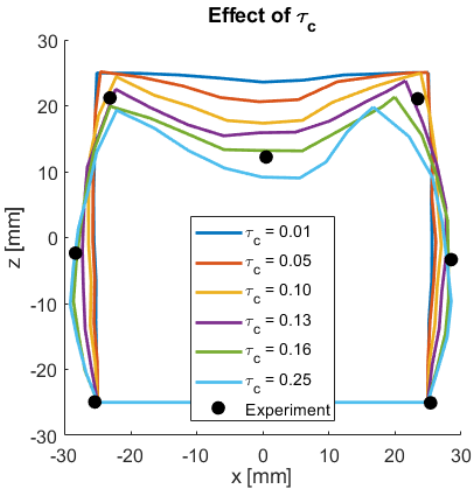
1) Compression constraint parameter τ_c: Simulations have been executed for a range of values for τ_c, ranging from 0.01 to 0.25, where the other two parameters are set to $\tau_s = 0.15$ and $\alpha = 7 \cdot 10^6$. Fig. 19a shows the contours of the object for a few simulations and the points from the experiment. It is clear from this figure that for a low value of τ_c, meaning that the springs are not allowed to compress much, the deformation of the object is also small and does not reach the depth of the experiment. For an increasing value of τ_c the finger deforms the object more and the depth also increases. Meanwhile, when τ_c is set too high, the object is overly compressed. Another behaviour to notice is that the sides of the object bulges out more for increasing compression.

The effect of τ_c is reflected in the error of the simulations compared to the experiment. Fig. 20a shows the normalized root mean square error, showing the error of each simulation as an orange marker and a fitted curve in blue. It can be seen, as expected from Fig. 19a, that the error is at its minimum for values of τ_c where the depth of the deformation is around the point of the experiment. For an increasing or decreasing value, the error will increase.

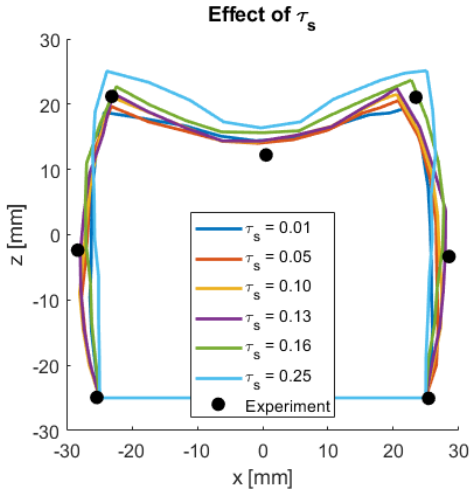
2) Stretching constraint parameter τ_s: To see the effect of τ_s, simulation have been performed, where τ_s ranges from 0.01 to 0.25, $\tau_c = 0.15$ and $\alpha = 7 \cdot 10^6$. The contours of the object after simulation and the points of the corresponding experiment are shown in Fig. 19b. The effect of τ_s is less clear than for τ_c, since the finger compresses the object. However, it can be noticed that τ_s does not have much influence on the compression of the object by the finger. The effect is more visible at the top corners of the object, where for lower values of τ_s the corners are pulled down and inwards by the deformation, since the springs between the contact point and the corners cannot stretch a lot. While for larger values of τ_s the deformation is more local and the corners are less affected.

The normalized RMSE of the simulations are shown in Fig. 20b. It can be concluded that indeed τ_s has less influence on the deformation and thus error than τ_c, as shown by the flatter curve. This could be explained by the fact that the experiment mostly puts the object under compressive stress. For this specific situation the minimum error is achieved at $\tau_s \approx 0.15$ and for lower values the error only increases slightly. However, for values larger than where the minimum error occurs, the error will increase rapidly.

3) Volume preservation parameter α: For the volume preservation coefficient α simulations are conducted for values ranging from 10^5 to 10^8 and $\tau_s = \tau_c = 0.15$. Fig. 21 shows the normalized volume error of the simulations in comparison with the initial volume of the object, assuming the object is incompressible and the volume stays constant. The figure shows that for larger values of α the error decreases, as expected since for higher values the volume correction force is also higher.



(a) $\tau_c \in [0.01, 0.25]$, $\tau_s = 0.15$ and $\alpha = 7 \cdot 10^6$

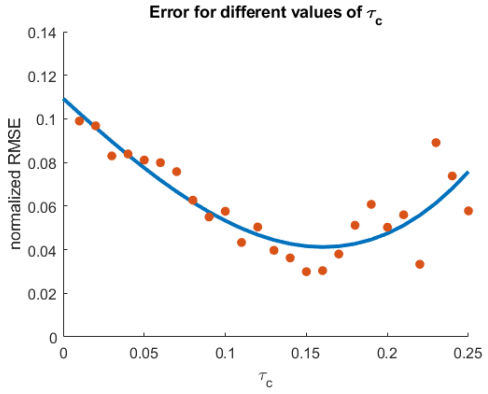


(b) $\tau_s \in [0.01, 0.25]$, $\tau_c = 0.15$ and $\alpha = 7 \cdot 10^6$

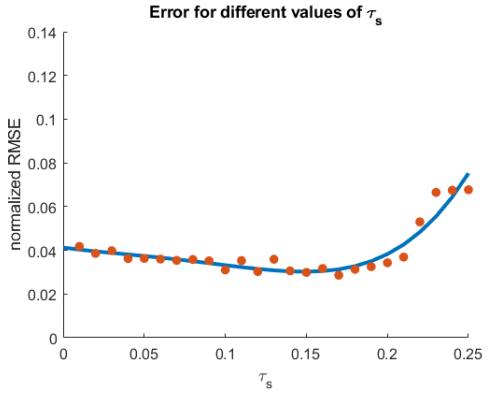
Fig. 19: Simulation results for different values of parameters

4) Parameters τ_s, τ_c and α: In order to find the best fit for the parameters and minimize the error, it is not sufficient to only know the effect of a single parameter. All three parameters influence the behaviour of the object model and thus the error has to be minimized for a set of parameters τ_s, τ_c and α. First, Fig. 22 shows the normalized RMSE for τ_s and τ_c ranging from 0.01 to 0.25 and a fixed α. This plot shows the best combinations of τ_s and τ_c for a specific α. It can be seen that the error is minimal in the center area of the surface, where the value of τ_s and τ_c are around 0.15.

To also include the effect of α, Fig. 23 shows the same plot as before for three values of α. Note that for these plots the number of values for τ_s and τ_c are reduced. Fig. 23a shows a comparable result as Fig. 22, where there is a clear valley around $\tau_s = \tau_c = 0.15$. For $\alpha = 10^7$ and $\alpha = 10^8$, Fig. 23b and Fig. 23c respectively, the valley is elongated towards lower values of τ_s, where τ_c is still 0.15.



(a) $\tau_c \in [0.01, 0.25]$, $\tau_s = 0.15$ and $\alpha = 7 \cdot 10^6$



(b) $\tau_s \in [0.01, 0.25]$, $\tau_c = 0.15$ and $\alpha = 7 \cdot 10^6$

Fig. 20: Errors for different values of parameters

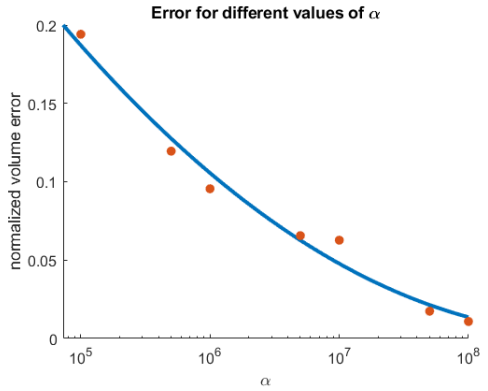


Fig. 21: Error for $\alpha \in [10^5, 10^8]$, $\tau_s = 0.15$ and $\tau_c = 0.15$

D. Simulation results

The best fit of the model parameters is achieved by minimizing the root mean square error and is found for $\tau_s = 0.16$, $\tau_c = 0.155$ and $\alpha = 6 \cdot 10^6$. The RMSE and normalized RMSE are 1.11 mm and 2.2% respectively. Fig. 24 shows two figures comparing the contours of the simulation result with the image of the experiment. It can be seen that the simulated deformed object and the position of the finger mostly matches with the experiment. Fig. 25 shows images of the simulation itself.

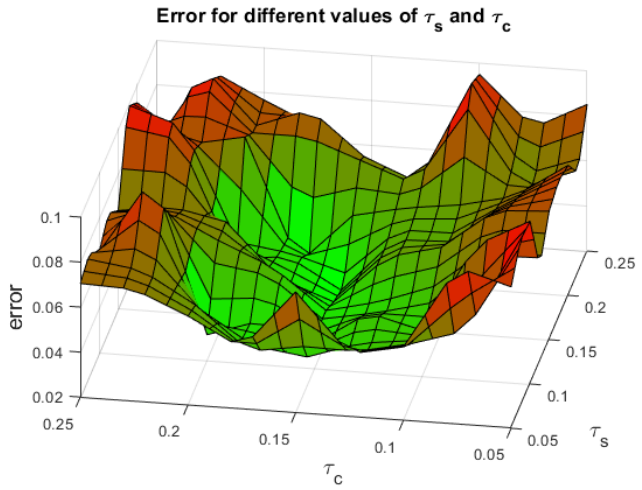


Fig. 22: Error for $\tau_s \in [0.01, 0.25]$, $\tau_c \in [0.01, 0.25]$ and $\alpha = 7 \cdot 10^6$

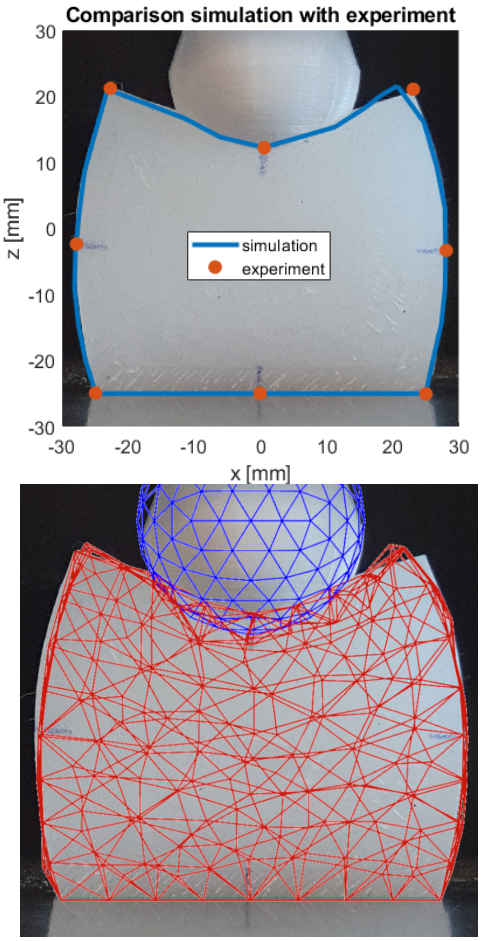
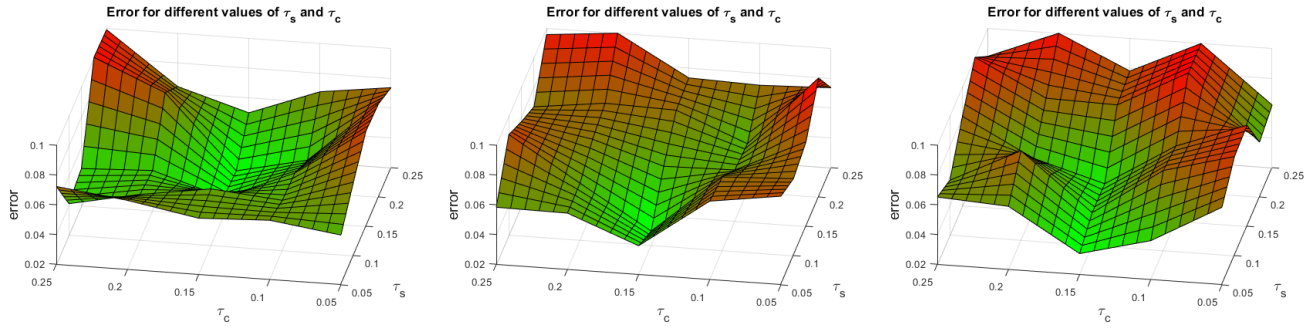


Fig. 24: Comparison of simulation result with experiment

VIII. Conclusions

This paper proposed a model of deformable objects and their contact with fingers. The model is a mass-spring-damper model with additional constraints for volume preservation and spring deformation limits, and Hunt-Crossley [9]



(a) $\alpha = 10^6$ (b) $\alpha = 10^7$ (c) $\alpha = 10^8$

Fig. 23: Errors for different values of α and $\tau_s \in [0.01, 0.25]$, $\tau_c \in [0.01, 0.25]$

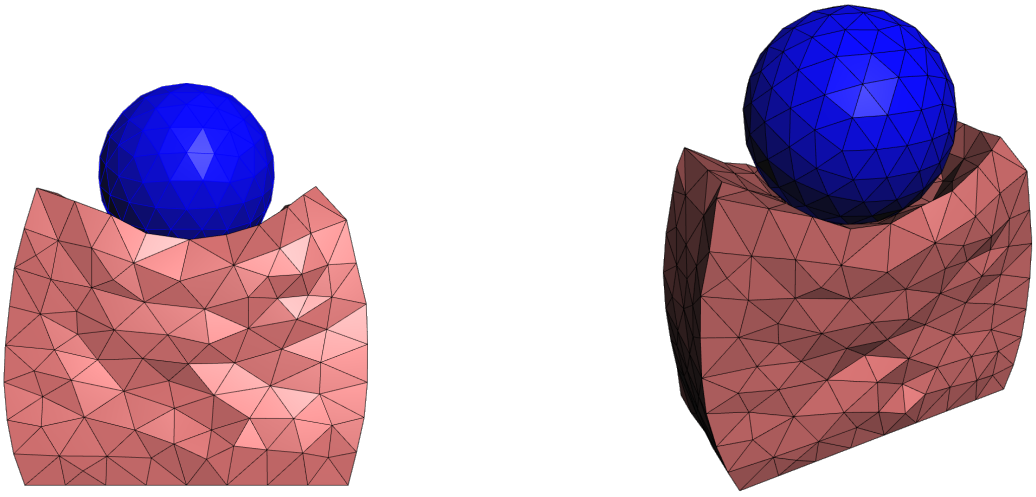


Fig. 25: Images of the simulation with the best fit parameters

and stick-slip contacts. This model is used to understand the behaviour of a deformable object when in contact with a finger. Experiments with objects made of silicone rubber are conducted to acquire data about the deformation of an object when impacted by an external fingertip. The influence of model parameters on the simulation results are explored and fitted to minimize the error. The results indicate that the proposed model can predict object deformations with a normalized root mean square error of 2.2% in the explored situations. The simulation can later be extended by grasping and picking up an object and be used for planning the grasping and manipulation of deformable objects.

IX. Recommendations

To extend and improve the model and simulation a few recommendations are proposed.

First of all, several additions to the contact model are presented. Currently, the finger is simplified to a sphere with a center position and radius for easy and fast contact detection. To account for fingers with other geometries, the contact model can be extended with for example cubic or cuboid fingers. Therefore, the contact detection and contact force will change. The next step would be to combine multiple simple geometries to create a more complex finger. Optimally, the finger is also represented by a meshed (deformable) object, however that would increase the computation time drastically, since a large amount of possible contacts have to be checked and computed. Another contact improvement is to add collision of the object with the ground or table, allowing the object to fall and slide on the ground. The last contact addition is self-collision to prevent the object from moving into itself.

For the volume preservation a more complex method could be used that also depends on material properties. Baudet et al. [4] proposed a method to take into account the Young's modulus and Poisson ratio for determining the spring stiffness and correction forces for volume preservation. Golec [8] gives an overview of other methods for volume preservation.

Friction of the contact model between the object and finger is important for grasping the object. In order to extend the simulation to grasping and manipulation and create a reliable model, the friction behaviour has to be investigated in more details. Therefore, it is recommended to conduct experiments to find the friction coefficients, and the stick-slip behaviour and parameters.

The meshing method and resolution will influence the behaviour of the model. To find the optimal mesh, as far

as possible since that will differ for each object, the effect of meshing (e.g. tetrahedral/hexahedral/other geometries or structured/unstructured) and resolution of the mesh might be investigated.

REFERENCES

[1] "Sphere and plane intersection," n.d. [Online]. Available: http://www.ambrsoft.com/TrigoCalc/Sphere/SpherePlaneIntersection_.htm

[2] D. Baraff and A. Witkin, "Large steps in cloth simulation," *Proceedings of SIGGRAPH*, vol. 98, 06 2001.

[3] H. Basloom, "A survey on physical methods for deformation modeling," *International Journal of Scientific & Technology Research*, vol. 5, pp. 59–64, 2016.

[4] V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara, "Integrating tensile parameters in hexahedral mass-spring system for simulation," 01 2009.

[5] D. Dopico Dopico, A. Luaces, M. Gonzalez, and J. Cuadrado, "Dealing with multiple contacts in a human-in-the-loop application," *Multibody System Dynamics*, vol. 25, pp. 167–183, 02 2011.

[6] Y. Duan, W. Huang, H. Chang, W. Chen, K. Toe, J. Zhou, T. Yang, J. Liu, S.-K. Teo, C. Lim, Y. Su, C.-K. Chui, and S. Chang, "Modeling and simulation of soft tissue deformation," 2013.

[7] Y. Duan, W. Huang, H. Chang, W. Chen, J. Zhou, S.-K. Teo, Y. Su, C.-K. Chui, and S. Chang, "Volume preserved mass-spring model with novel constraints for soft tissue deformation," *IEEE journal of biomedical and health informatics*, vol. 20, 2014.

[8] K. Golec, "Hybrid 3d mass spring system for soft tissue simulation." 2018.

[9] K. H. Hunt and F. E. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *Journal of Applied Mechanics*, vol. 42, pp. 440–445, 1975.

[10] B. Lloyd, G. Székely, and M. Harders, "Identification of spring parameters for deformable object simulation," *IEEE transactions on visualization and computer graphics*, vol. 13, pp. 1081–94, 2007.

[11] M. Mir, "Additive manufacturing simulation - tetrahedral vs. hexahedral mesh," 2021. [Online]. Available: https://www.additive-lab.com/post/additive-manufacturing-simulation-tetrahedral-vs-hexahedral-mesh

[12] W. Mollemans, F. Schutyser, J. V. Cleynenbreugel, and P. Suetens, "Tetrahedral mass spring model for fast soft tissue deformation," 2003.

[13] MonkeyProof Solutions, "How to calculate the shortest distance between a point and a line?" 2016. [Online]. Available: https://monkeyproofsolutions.nl/wordpress/how-to-calculate-the-shortest-distance-between-a-point-and-a-line/

[14] C. Paloc, F. Bello, R. Kitney, and A. Darzi, "Online multiresolution volumetric mass spring model for real time soft tissue deformation," 09 2002, pp. 219–226.

[15] P. Pathmanathan, D. J. Gavaghan, J. P. Whiteley, S. J. Chapman, and J. M. Brady, "Predicting tumor location by modeling the deformation of the breast," *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 2471–2480, 2008.

[16] X. Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behavior," 1995.

[17] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Tech. Rep., 1994.

[18] D. Steck, J. Qu, S. Baghbani Kordmahale, D. Tscharnuter, A. Muliana, and J. Kameoka, "Mechanical responses of ecoflex silicone rubber: Compressible and incompressible behaviors," *Journal of Applied Polymer Science*, vol. 136, 2018.

[19] L. Zaidi, B. C. Bouzgarrou, S. Laurent, and Y. Mezouar, "Modeling and analysis of 3d deformable object grasping," in *2014 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD)*, 2014, pp. 1–8.

[20] L. Zaidi, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, and L. Sabourin, "Model-based strategy for grasping 3d deformable objects using a multi-fingered robotic hand," *Robotics and Autonomous Systems*, vol. 95, pp. 196–206, 2017.

[21] J. Zhang, Y. Zhong, and C. Gu, "Deformable models for surgical simulation: A survey," *IEEE Reviews in Biomedical Engineering*, vol. 11, pp. 143–164, 2018.

4 Software overview

The simulation software with the model is implemented in C++ of which the class diagram is shown in Figure 4.1. This diagram gives the structure of the classes of the implementation.

4.1 Simulator

The `Simulator` represents the simulation, which stores the `Object`, one or multiple `Finger`s, optionally a `Viewer` to visualize the simulation and simulation parameters (volume preservation constant α, over-stretching and over-compression constraint parameters τ_s and τ_c respectively) and the simulation time step. The `Simulator` is also responsible for the flow of the simulation by running the simulation loop. For each time step, at first the spring forces and the contact forces are computed. Thereafter, the correction forces for volume preservation are calculated and the states are updated by computing the new position and velocity of the particles and finger(s). After the integration, the spring constraints are applied.

4.2 Object

The `Object` class represents the deformable object and is a container for all parts representing the discretized object for a mass-spring-damper system. The `Object` has a mass density, Young's modulus and Poisson's ratio and consists of `Particle`s, `Spring`s, `Element`s and `Facet`s. The mesh is generated by the library TetGen (Si, n.d.), which divides a geometry into tetrahedra using the Delaunay triangulation algorithm.

Particle

A `Particle` represents a mass point of the discretized object. The `Particle` keeps track of its mass, position, velocity, acceleration, force and whether it is fixed.

Spring

A `Spring` represents the spring and damper in parallel between two `Particle`s. A `Spring` has a stiffness and damping value, and an initial length.

Element

An `Element` represents a tetrahedron and contains four `Particle`s and six `Spring`s. Each `Element` has a current and initial volume, equivalent length, mass density and Young's modulus.

Facet

A `Facet` represents a triangular planar face of a tetrahedron that lies on the boundary of the object and consists of three `Particle`s. The `Facet` keeps track of the contact between a finger and itself, assuming each facet can at most be in contact with one finger.

4.3 Finger

A `Finger` is a simplification of a gripper finger in a sphere. A `Finger` has the properties mass, radius, Young's modulus and Poisson's ratio, and position, velocity, acceleration and force. A `Finger` can be actuated in two ways. The first method is force-controlled, whereby the finger moves due to a force acting on it or gravity. The second method is velocity-controlled, where the finger moves at a certain speed.

4.4 `Viewer`

The `Viewer` visualizes the `Object`, `Fingers`, contact points and forces using the Easy3D (Nan, 2021) library.

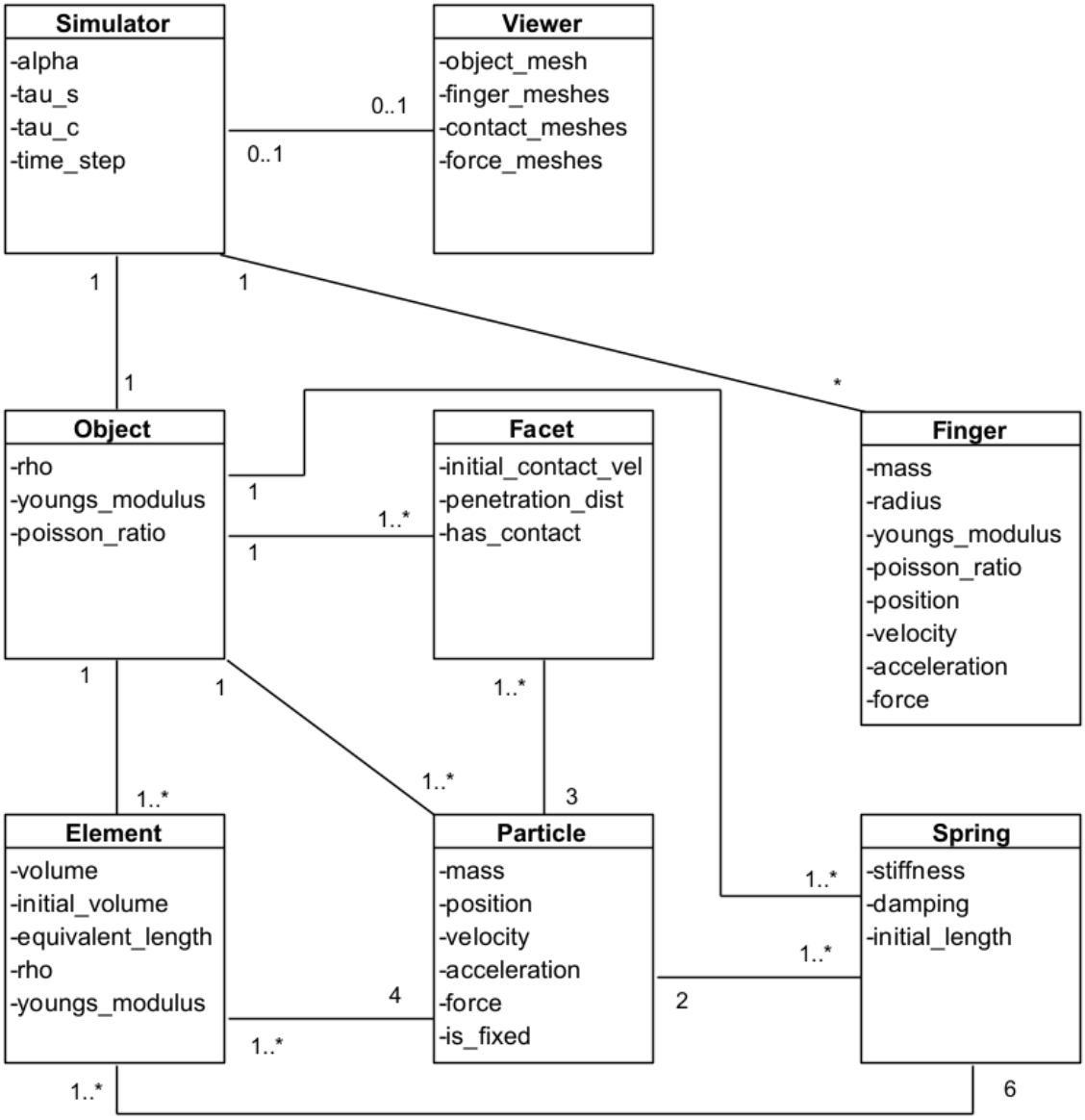


Figure 4.1: Class diagram of the software

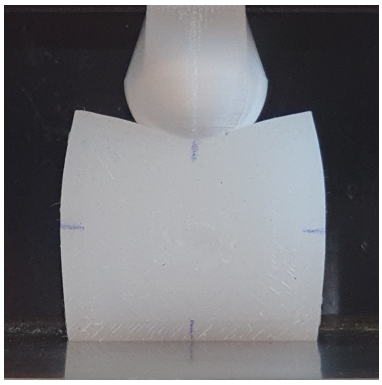
5 Results

Section 5.1 shows some additional results of the experiments and Section 5.2 some of the simulation results.

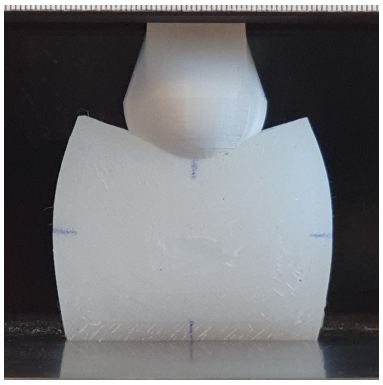
5.1 Experiment results

5.1.1 Cuboid Ecoflex 00-20

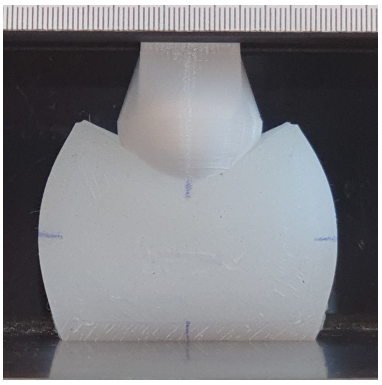
Figure 5.1 visualizes results of the experiment with the cuboid object made of Ecoflex 00-20 with the large finger compressing the object. The experiment is performed for weights up to 1.46 kg. Figure 5.1a to Figure 5.1c shows the deformation of the object at three different weights. Figure 5.1d shows the eight points of the object after deformation for different weights. Note that the lines between the points are only connecting the points and are not representable for the boundary of the deformed object. It can be seen that for an increasing weight/force the object will be compressed by the finger on the top and the sides will bulge out as expected. It can also be seen that the top corners of the object slowly bend inwards and that the bottom points do not deform, probably due to friction. Figure 5.1e shows the x and z positions of the points, where the same behaviour can be seen.



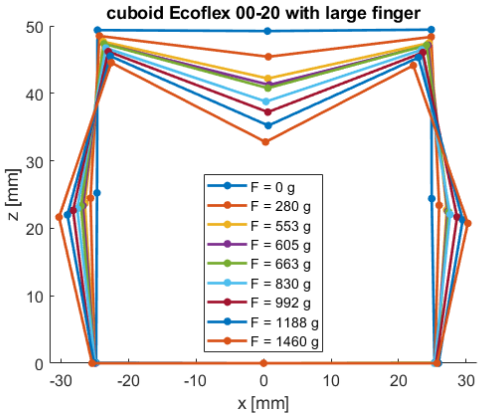
(a) Weight of 605 g



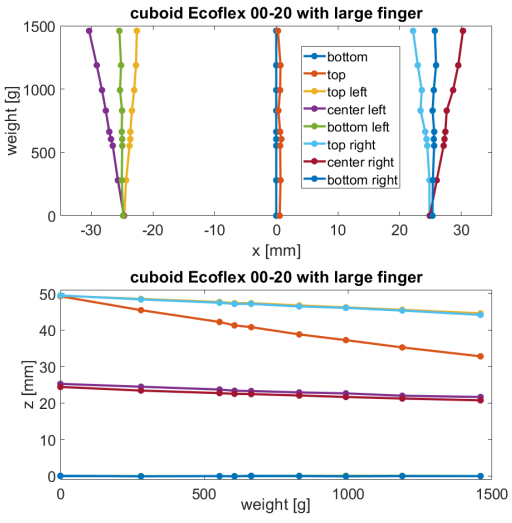
(b) Weight of 992 g



(c) Weight of 1460 g



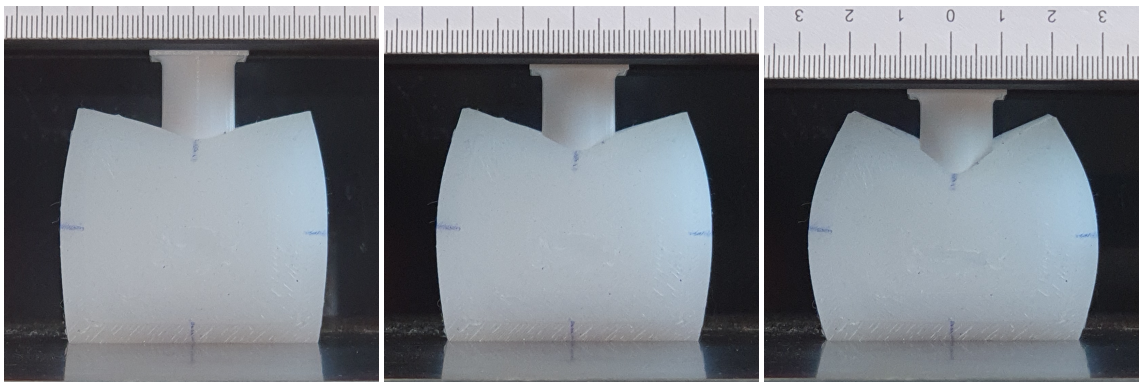
(d) Points of experiments



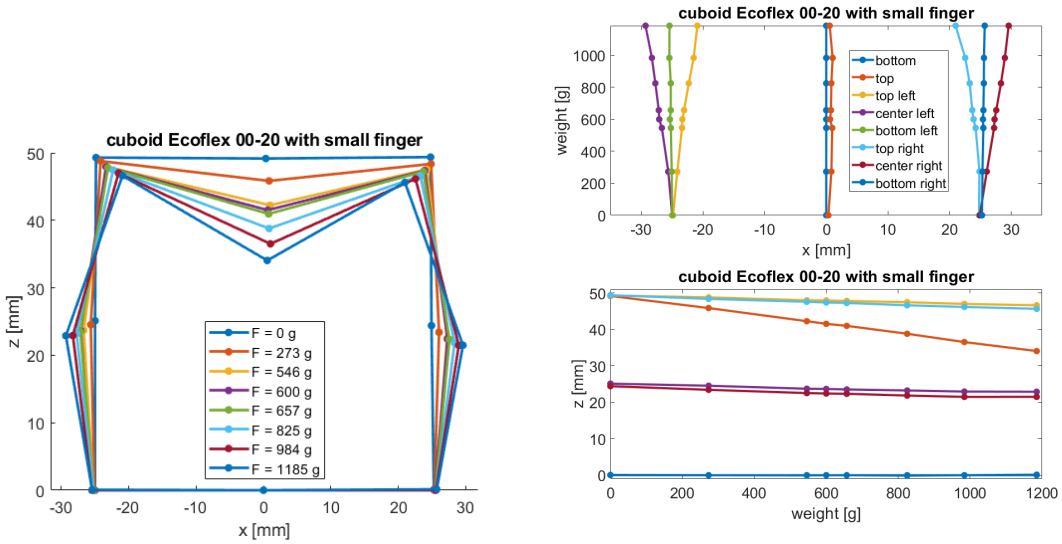
(e) x and z positions of the points

Figure 5.1: Cuboid Ecoflex 00-20 with large finger

Figure 5.2 shows similar images and plots as Figure 5.1 for the situation where the same object is compressed by a smaller finger. The results are comparable to the large finger, but with a more local deformation due to the smaller contact area of the smaller finger.



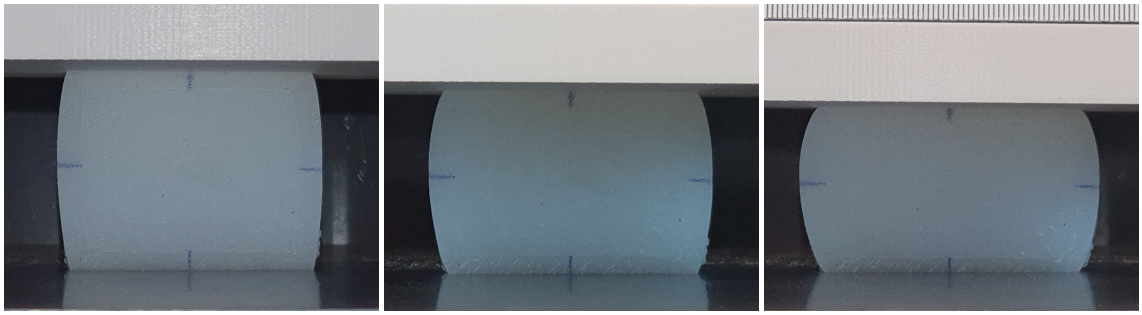
(a) Weight of 600 g **(b)** Weight of 825 g **(c)** Weight of 1185 g



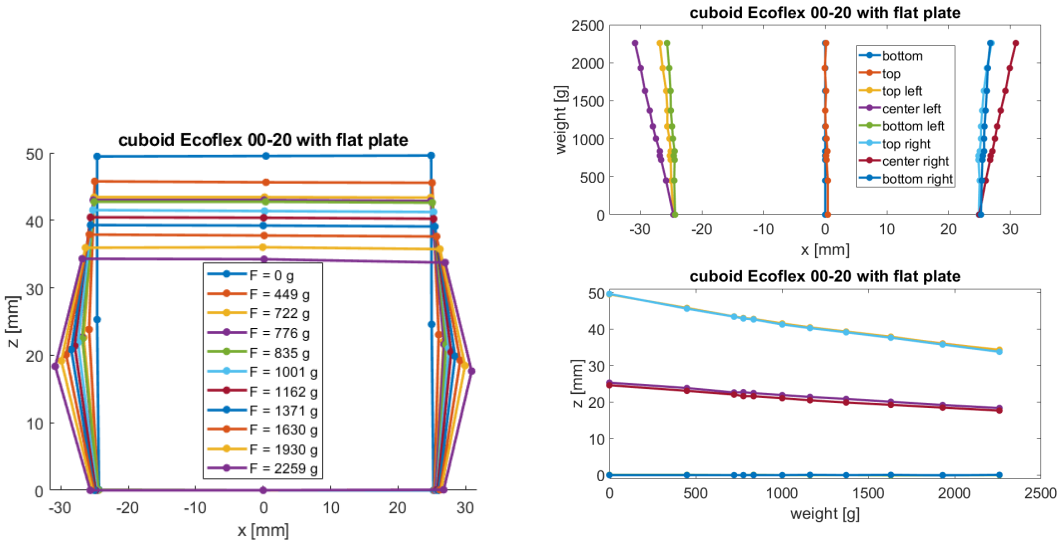
(d) Points of experiments **(e)** x and z positions of the points

Figure 5.2: Cuboid Ecoflex 00-20 with small finger

Figure 5.3 shows the experiment results of a flat plate compressing the cuboid object. As can be expected, by increasing the weight, the object compresses and bulges out more.



(a) Weight of 1001 g **(b)** Weight of 1630 g **(c)** Weight of 2269 g

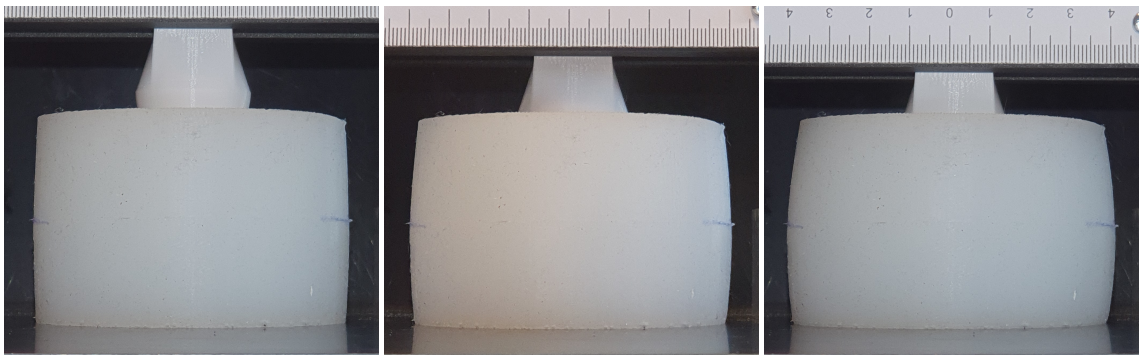


(d) Points of experiments **(e)** x and z positions of the points

Figure 5.3: Cuboid Ecoflex 00-20 with flat plate

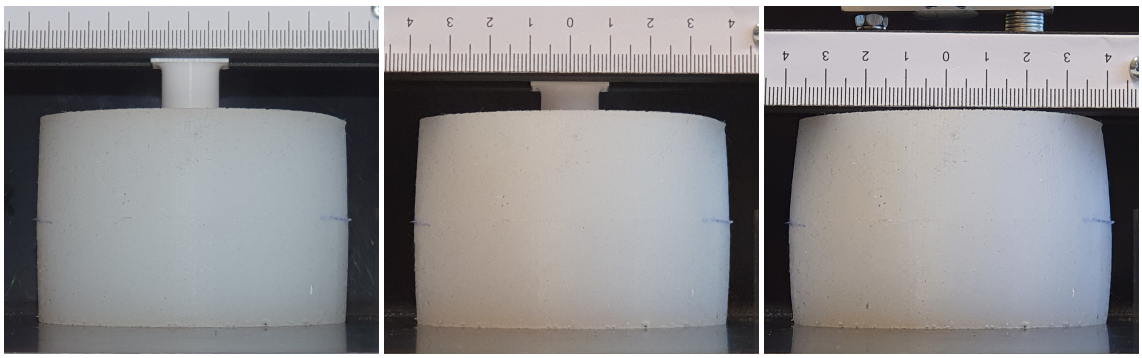
5.1.2 Cylinder Ecoflex 00-20

Figure 5.4 to Figure 5.6 shows images of the experiments on the cylinder object of the material Ecoflex 00-20. The large finger, small finger and flat plate are respectively pushing on the object. It can be seen that the fingers do not compress the object significantly and that there is only a slight deformation visible in the width of the object. The large and small fingers deform the object very locally, resulting in the fingers being enclosed by the object.



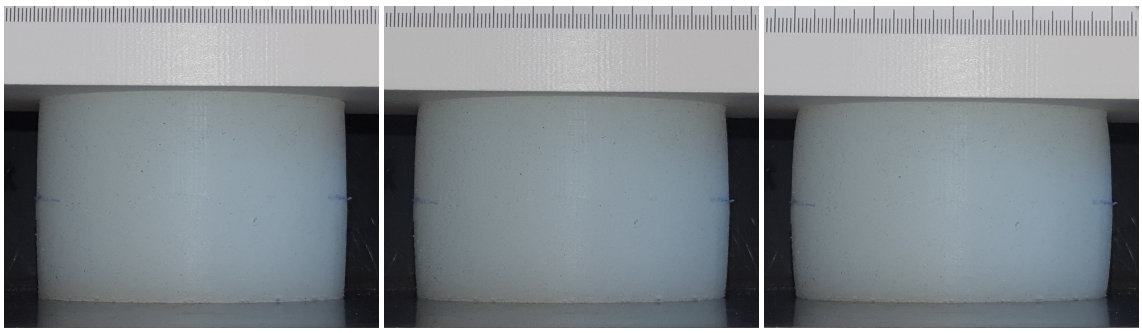
(a) Weight of 992 g **(b)** Weight of 2170 g **(c)** Weight of 3183 g

Figure 5.4: Cylinder Ecoflex 00-20 with large finger



(a) Weight of 600 g **(b)** Weight of 1187 g **(c)** Weight of 2158 g

Figure 5.5: Cylinder Ecoflex 00-20 with small finger



(a) Weight of 1160 g **(b)** Weight of 1984 g **(c)** Weight of 3019 g

Figure 5.6: Cylinder Ecoflex 00-20 with flat plate

5.2 Simulation results

Figure 5.7 shows the contours of the deformed object from simulations during the model parameters exploration mentioned in Chapter 3. The white markers are points of the object from the experiment.

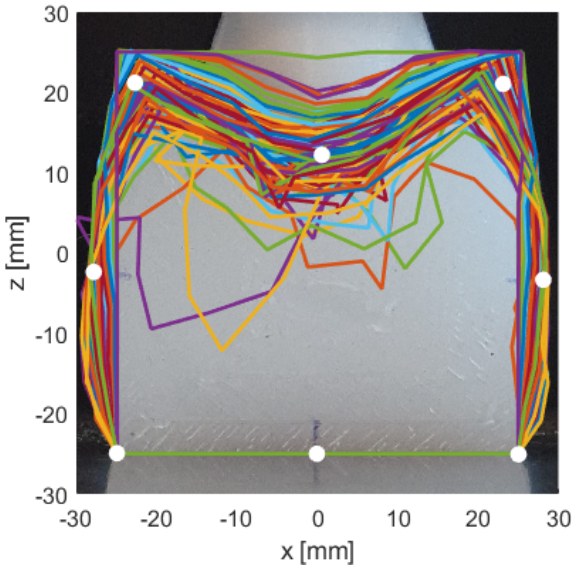
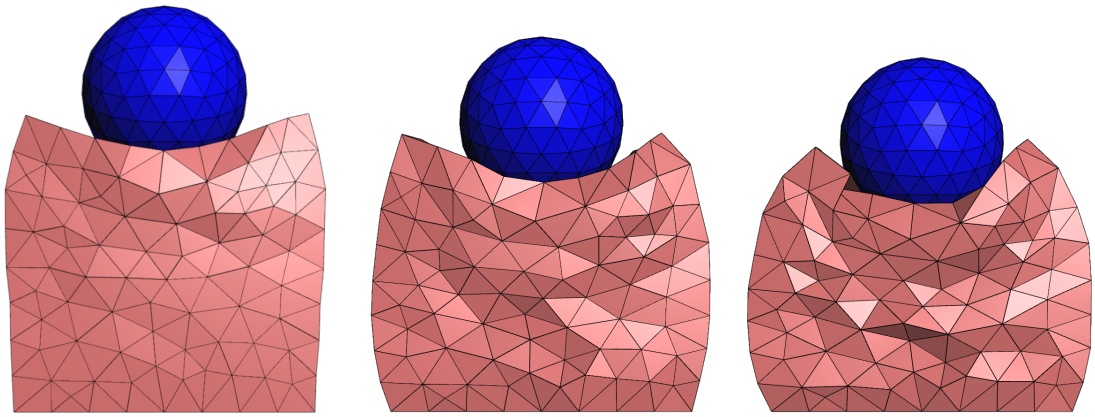


Figure 5.7: Simulation results for a wide range of parameters

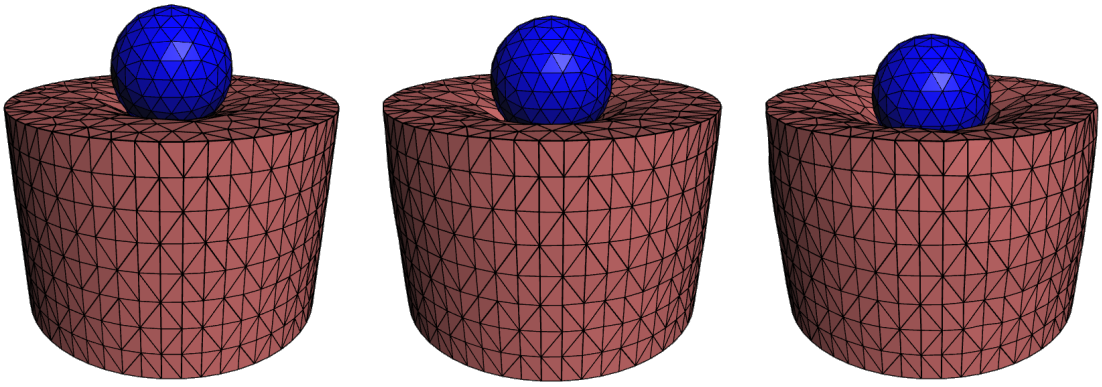
In Figure 5.8 the same object (cuboid Ecoflex 00-20) is compressed by the large finger with three different weights. The model parameters are set to the same value (as found in Chapter 3), only the weight of the finger has been changed. The results of the simulations are in accordance with the experiments in Figure 5.1.



(a) Weight of 605 g **(b)** Weight of 992 g **(c)** Weight of 1460 g

Figure 5.8: Simulation of cuboid Ecoflex 00-20 with large finger

The same simulation is performed on the cylinder object with the large finger. Figure 5.9 shows the deformation at three different weights.



(a) Weight of 992 g **(b)** Weight of 2170 g **(c)** Weight of 3183 g

Figure 5.9: Simulation of cylinder Ecoflex 00-20 with large finger

Figure 5.10 visualizes the deformation of the object at three time steps during a grasping simulation. The cuboid object is being pinched by two fingers that are slowly moving towards each other. This kind of simulation can help understanding the behaviour of the object during a grasp and find the optimal gripping position.

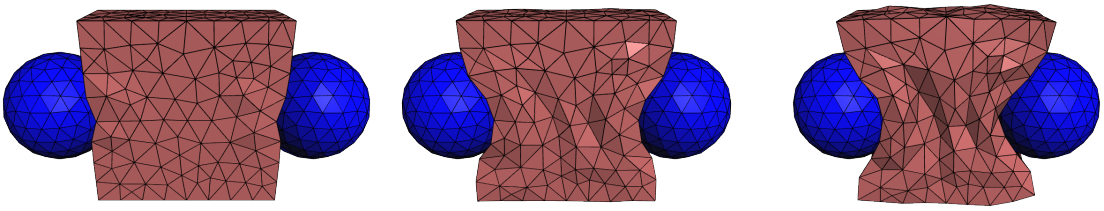


Figure 5.10: Simulation of two fingers pinching cuboid Ecoflex 00-20 at three moments

6 Conclusions

This thesis proposed a model of deformable objects and their contact with fingers. The model is a mass-spring-damper model with additional constraints for volume preservation and spring deformation limits, and Hunt-Crossley Hunt and Crossley (1975) and stick-slip contacts. This model is used to understand the behaviour of a deformable object when in contact with a finger. Experiments with objects made of silicone rubber are conducted to acquire data about the deformation of an object when impacted by an external fingertip. The influence of model parameters on the simulation results are explored and fitted to minimize the error. The results indicate that the proposed model can predict object deformations with a normalized root mean square error of 2.2% in the explored situations. The simulation can later be extended by grasping and picking up an object and be used for planning the grasping and manipulation of deformable objects.

Appendices

A Experimental set-up

Figure A.1 shows the set-up for the experiment.

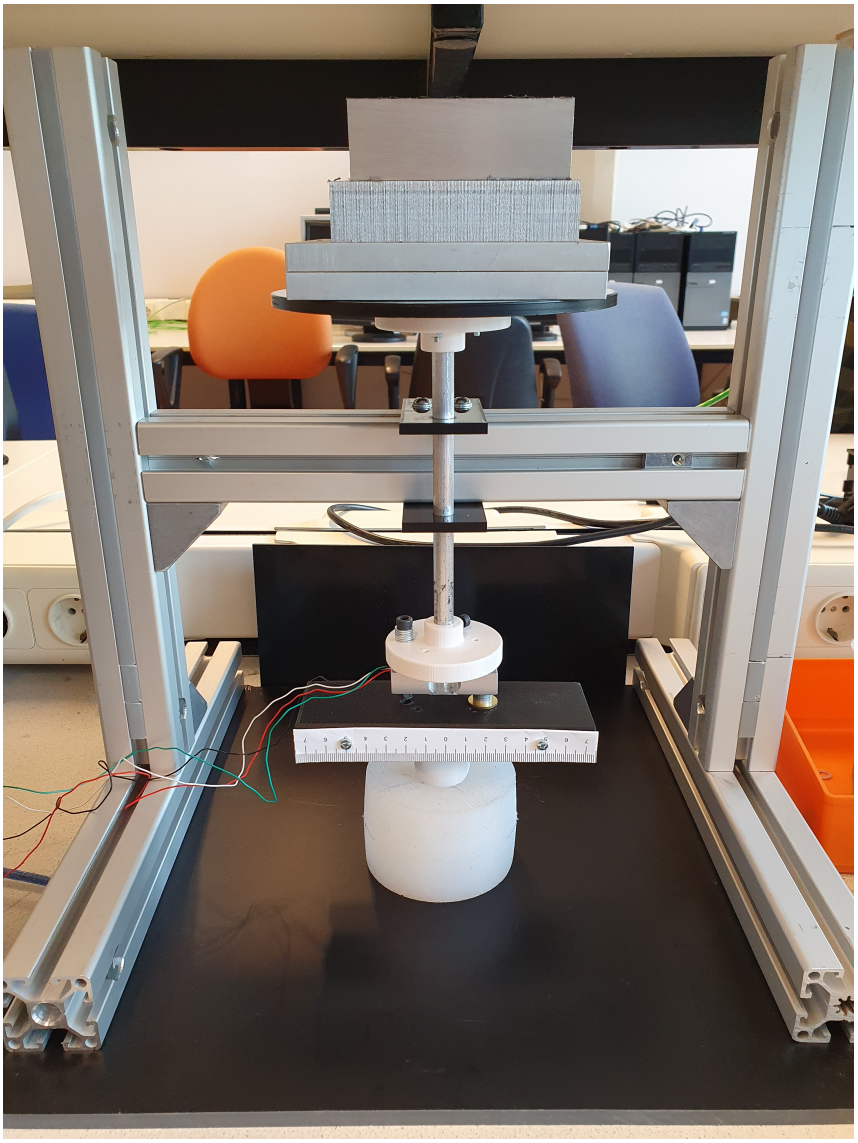


Figure A.1: Experimental set-up

B Try-outs

In order to get more insight in the modelling of deformable objects and test the feasibility and software, simple or less complex models have been implemented.

B.1 State-space in MATLAB

First simple mass-spring-damper systems in 2D have been created in a state-space model in MATLAB. A state-space model is described in the form

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} \end{aligned} \tag{B.1}$$

where $\boldsymbol{x}$ is the state (position and velocity of each particle), $\boldsymbol{u}$ the input (external forces) and $\boldsymbol{y}$ the output.

These models gave fast and good initial results, but a state-space representation is very limited for simulation. The reason for this is that the complete system is completely described initially and cannot depend on external factors such as gripper interactions.

B.2 ODE in MATLAB

To solve the limitation of the state-space models, deformable object models in 2D have been rewritten in ordinary differential equations. Therefore, a second-order differential equation has to be rewritten in two first order differential equations for each particle. This results in

$$\begin{aligned} \frac{d\boldsymbol{y}}{dt} &= f(t, \boldsymbol{y}) \\ \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \end{bmatrix} \end{aligned} \tag{B.2}$$

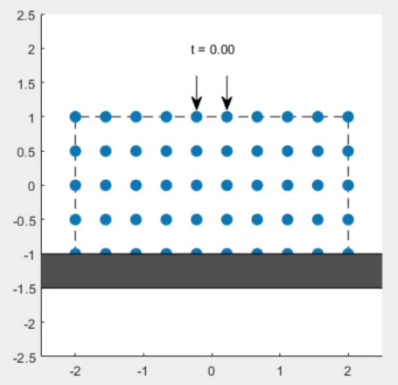
where t is the time and $\boldsymbol{x}$ the positions of the particles.

The differential equations are then solved by the ODE-solvers of MATLAB. This method gave good results as well, with more freedom to interact with the object. However, when extending to 3D, more particles or more complex situations, the description of the state in one vector $\boldsymbol{y}$ is messy and confusing.

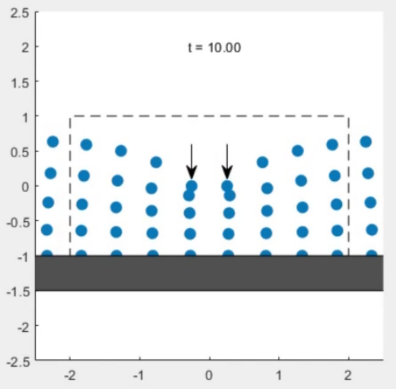
Figure B.1 shows a simulation where the object rests on a fixed table while a force is applied on the top. Figure B.2 illustrates the simulation with a parallel gripper squeezing the object, where Figure B.2a and Figure B.2b shows the state before and after compression of the object by the gripper respectively.

B.3 OOP in Python/C++

To account for the complexity, the model and simulation have been rewritten in Python and C++ in an Object-Oriented Programming (OOP) manner. OOP is a method to structure a software program by introducing classes and objects. A class is a description of an object or thing with certain properties and methods or functions to execute some action. Multiple instances, called objects, with their own data, can then be created from a class.

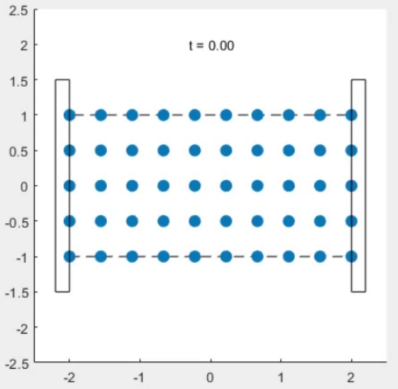


(a) Object at time = 0 s

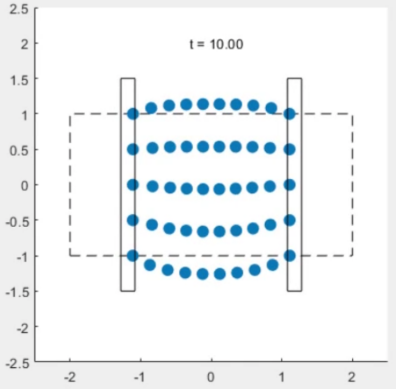


(b) Object at time = 10 s

Figure B.1: 2D simulation of an object on a table with a force applied on the top



(a) Object at the beginning



(b) Object after compression

Figure B.2: 2D simulation of an object being compressed by a parallel gripper

The feasibility and structure of the algorithm have first been implemented and tested in Python. In order to get more speed and more extensive possibilities for mesh generation, the simulation is finally implemented in C++, which is discussed in Chapter 4.

Bibliography

(n.d.), Sphere and plane intersection.
http://www.ambrsoft.com/TrigoCalc/Sphere/SpherePlaneIntersection_.htm

Al-Mohammed, M., Z. Ding, P. Liu and A. Behal (2018), An Adaptive Control Based Approach for Gripping Novel Objects with Minimal Grasping Force, *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pp. 1040–1045.

Baraff, D. and A. Witkin (2001), Large Steps in Cloth Simulation, *Proceedings of SIGGRAPH*, **vol. 98**, doi:10.1145/280814.280821.

Basloom, H. (2016), A Survey On Physical Methods For Deformation Modeling, *International Journal of Scientific & Technology Research*, **vol. 5**, pp. 59–64.

Baudet, V., M. Beuve, F. Jaillet, B. Shariat and F. Zara (2009), Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation.

Bersch, C., B. Pitzer and S. Kammel (2011), Bimanual robotic cloth manipulation for laundry folding, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1413–1419, doi:10.1109/IROS.2011.6095109.

Cusumano-Towner, M., A. Singh, S. Miller, J. F. O'Brien and P. Abbeel (2011), Bringing clothing into desired configurations with limited perception, in *2011 IEEE International Conference on Robotics and Automation*, pp. 3893–3900, doi:10.1109/ICRA.2011.5980327.

Das, J. and N. Sarkar (2011), Autonomous Shape Control of a Deformable Object by Multiple Manipulators, *Journal of Intelligent and Robotic Systems*, **vol. 62**, pp. 3–27, doi:10.1007/s10846-010-9436-5.

Delgado, A., C. A. Jara, D. Mira and F. Torres (2015), A tactile-based grasping strategy for deformable objects' manipulation and deformability estimation, in *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 02, pp. 369–374.

Dopico Dopico, D., A. Luaces, M. Gonzalez and J. Cuadrado (2011), Dealing with multiple contacts in a human-in-the-loop application, *Multibody System Dynamics*, **vol. 25**, pp. 167–183, doi:10.1007/s11044-010-9230-y.

Doumanoglou, A., A. Kargakos, T.-K. Kim and S. Malassiotis (2014), Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning, in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 987–993, doi:10.1109/ICRA.2014.6906974.

Duan, Y., W. Huang, H. Chang, W. Chen, K. Toe, J. Zhou, T. Yang, J. Liu, S.-K. Teo, C. Lim, Y. Su, C.-K. Chui and S. Chang (2013), Modeling and Simulation of Soft Tissue Deformation.

Duan, Y., W. Huang, H. Chang, W. Chen, J. Zhou, S.-K. Teo, Y. Su, C.-K. Chui and S. Chang (2014), Volume Preserved Mass-Spring Model with Novel Constraints for Soft Tissue Deformation, *IEEE journal of biomedical and health informatics*, **vol. 20**.

Engeberg, E. D. and S. G. Meek (2013), Adaptive Sliding Mode Control for Prosthetic Hands to Simultaneously Prevent Slip and Minimize Deformation of Grasped Objects, *IEEE/ASME Transactions on Mechatronics*, pp. 376–385.

Fanson, R. and A. Patriciu (2010), Model based deformable object manipulation using linear robust output regulation, in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 496–501, doi:10.1109/IROS.2010.5650163.

FlexCRAFT (2021), FlexCRAFT.
https://flexcraftprogram.com/

Golec, K. (2018), Hybrid 3D Mass Spring System for Soft Tissue Simulation.

Gunji, D., Y. Mizoguchi, S. Teshigawara, A. Ming, A. Namiki, M. Ishikawaand and M. Shimojo (2008), Grasping force control of multi-fingered robot hand based on slip detection using tactile sensor, in *IEEE International Conference on Robotics and Automation*, pp. 2605–2610.

Guo, F., H. Lin and Y.-B. Jia (2013), Squeeze grasping of deformable planar objects with segment contacts and stick/slip transitions, in *2013 IEEE International Conference on Robotics and Automation*, pp. 3736–3741, doi:10.1109/ICRA.2013.6631102.

Hasegawa, H., Y. Mizoguchi, K. Tadakuma, A. Ming, M. Ishikawa and M. Shimojo (2010), Development of intelligent robot hand using proximity, contact and slip sensing, in *IEEE International Conference on Robotics and Automation*, pp. 777–784.

Howard, A. and G. Bekey (1999), Intelligent learning for deformable object manipulation, in *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No.99EX375)*, pp. 15–20.

Hunt, K. H. and F. E. Crossley (1975), Coefficient of Restitution Interpreted as Damping in Vibroimpact, *Journal of Applied Mechanics*, **vol. 42**, pp. 440–445.

Jia, Y.-B., F. Guo and J. Tian (2011), On two-finger grasping of deformable planar objects, in *2011 IEEE International Conference on Robotics and Automation*, pp. 5261–5266, doi:10.1109/ICRA.2011.5980565.

Kaboli, M., K. Yao and G. Cheng (2016), Tactile Based Manipulation of Deformable Objects with Dynamic Center of Mass, in *IEEE-RAS International Conference on Humanoid Robots*.

Kassama, L. S. and M. Ngadi (2016), Shrinkage and Density Change of De-Boned Chicken Breast during Deep-Fat Frying, *Food and Nutrition Sciences*, **vol. 7**, pp. 895–905.

Kudoh, S., T. Gomi, R. Katano, T. Tomizawa and T. Suehiro (2015), In-air Knotting of Rope by a Dual-arm Multi-finger Robot, in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6202–6207, doi:10.1109/IROS.2015.7354262.

Lin, H., F. Guo, F. Wang and Y.-B. Jia (2015), Picking up a soft 3D object by "feeling" the grip, *The International Journal of Robotics Research*, **vol. 34**, pp. 1361 – 1384.

Lloyd, B., G. Székely and M. Harders (2007), Identification of Spring Parameters for Deformable Object Simulation, *IEEE transactions on visualization and computer graphics*, **vol. 13**, pp. 1081–94.

Maitin-shepard, J., M. Cusumano-towner, J. Lei and P. Abbeel (2010), Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in *In International Conference on Robotics and Automation (ICRA*.

Mir, M. (2021), Additive Manufacturing Simulation - Tetrahedral Vs. Hexahedral Mesh. https://www.additive-lab.com/post/additive-manufacturing-simulation-tetrahedral-vs-hexahedral-mesh

Mollemans, W., F. Schutyser, J. V. Cleynenbreugel and P. Suetens (2003), Tetrahedral Mass Spring Model for Fast Soft Tissue Deformation.

MonkeyProof Solutions (2016), How to calculate the shortest distance between a point and a line? https://monkeyproofsolutions.nl/wordpress/how-to-calculate-the-shortest-distance-between-a-point-and-a-line/

Monsó, P., G. Alenyà and C. Torras (2012), POMDP approach to robotized clothes separation, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1324–1329, doi:10.1109/IROS.2012.6386011.

Müller, M., B. Heidelberger, M. Hennix and J. Ratcliff (2006), Position Based Dynamics.

Nan, L. (2021), Easy3D: a lightweight, easy-to-use, and efficient C++ library for processing and rendering 3D data, *Journal of Open Source Software*, **vol. 6**, p. 3255, doi:10.21105/joss.03255.

Ogneva, I. V., D. V. Lebedev and B. S. Shenkman (2010), Transversal Stiffness and Young's Modulus of Single Fibers from Rat Soleus Muscle Probed by Atomic Force Microscopy.

Paloc, C., F. Bello, R. Kitney and A. Darzi (2002), Online Multiresolution Volumetric Mass Spring Model for Real Time Soft Tissue Deformation, pp. 219–226, ISBN 978-3-540-44225-7, doi:10.1007/3-540-45787-9_28.

Pathmanathan, P., D. J. Gavaghan, J. P. Whiteley, S. J. Chapman and J. M. Brady (2008), Predicting Tumor Location by Modeling the Deformation of the Breast, *IEEE Transactions on Biomedical Engineering*, **vol. 55**, pp. 2471–2480.

Provot, X. (1995), Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior.

Roberts, L., G. Singhal and R. R. Kaliki (2011), Slip detection and grip adjustment using optical tracking in prosthetic hands, in *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2011, Boston, MA, USA, August 30 - Sept. 3, 2011*, IEEE, pp. 2929–2932.

Sanchez, J., J. A. Corrales Ramon, B. C. BOUZGARROU and Y. Mezouar (2018), Robotic Manipulation and Sensing of Deformable Objects in Domestic and Industrial Applications: A Survey, *The International Journal of Robotics Research*, **vol. 37**, pp. 688 – 716.

Sani, H. N. and S. G. Meek (2011), Characterizing the performance of an optical slip sensor for grip control in a prosthesis, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1927–1932.

Shewchuk, J. R. (1994), An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, Technical report.

Shibata, M., T. Ota and S. Hirai (2009), Wiping motion for deformable object handling, in *2009 IEEE International Conference on Robotics and Automation*, pp. 134–139, doi:10.1109/ROBOT.2009.5152448.

Si, H. (n.d.), TetGen. A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator. http://wias-berlin.de/software/index.jsp?id=TetGen&lang=1

Steck, D., J. Qu, S. Baghbani Kordmahale, D. Tscharnuter, A. Muliana and J. Kameoka (2018), Mechanical responses of Ecoflex silicone rubber: Compressible and incompressible behaviors, *Journal of Applied Polymer Science*, **vol. 136**.

Vinh, T. V., T. Tomizawa, S. Kudoh and T. Suehiro (2012), A new strategy for making a knot with a general-purpose arm, in *2012 IEEE International Conference on Robotics and Automation*, pp. 2217–2222, doi:10.1109/ICRA.2012.6224852.

Yamakawa, Y., A. Namiki and M. Ishikawa (2010), Motion planning for dynamic knotting of a flexible rope with a high-speed robot arm, in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 49–54, doi:10.1109/IROS.2010.5651168.

Yamakawa, Y., A. Namiki, M. Ishikawa and M. Shimojo (2008), Knotting manipulation of a flexible rope by a multifingered hand system based on skill synthesis, in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2691–2696, doi:10.1109/IROS.2008.4650802.

Zaidi, L., B. C. Bouzgarrou, S. Laurent and Y. Mezouar (2014), Modeling and analysis of 3D deformable object grasping, in *2014 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD)*, pp. 1–8, doi:10.1109/RAAD.2014.7002259.

Zaidi, L., J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar and L. Sabourin (2017), Model-based strategy for grasping 3D deformable objects using a multi-fingered robotic hand, *Robotics and Autonomous Systems*, **vol. 95**, pp. 196–206, ISSN 0921-8890,

doi:https://doi.org/10.1016/j.robot.2017.06.011.

Zhang, J., Y. Zhong and C. Gu (2018), Deformable Models for Surgical Simulation: A Survey, *IEEE Reviews in Biomedical Engineering*, **vol. 11**, pp. 143–164, doi:10.1109/RBME.2017.2773521.