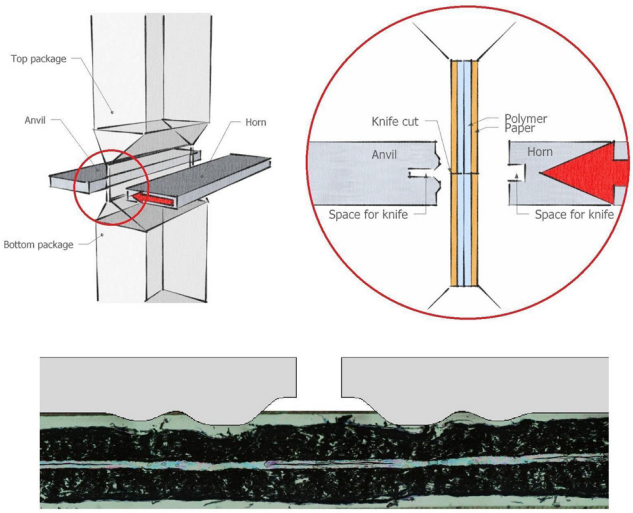
LUND UNIVERSITY



POLYMER FLOW IN PACKAGE SEALING PROCESS - A Coupled Eulerian-Lagrangian Numerical Approach

HENRIK JOHANSSON and JONATHAN OLLAR

Structural Mechanics

Master's Dissertation

Department of Construction Sciences
Structural Mechanics

ISRN LUTVDG/TVSM--11/5175--SE (1-72)
ISSN 0281-6679

POLYMER FLOW IN PACKAGE SEALING PROCESS - A Coupled Eulerian-Lagrangian Numerical Approach

Master's Dissertation by

HENRIK JOHANSSON and JONATHAN OLLAR

Supervisors

Kent Persson, PhD,
Div. of Structural Mechanics

Eskil Andreasson, MSc,
Tetra Pak

Examiner:

Per-Erik Austrell, Assoc. Professor,
Div. of Structural Mechanics

Printed by Wallin & Dalholm Digital AB, Lund, Sweden, May, 2011 ***(PI)***.

For information, address:
Division of Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: http://www.byggmek.lth.se

ABSTRACT

Tetra Pak produces packages designed to contain liquid products. This report is focused on the sealing process of the packages. These are made through compressing and heating of laminated paper through which the plastic laminates melt and merge.

The long term goal for Tetra Pak is to be able to predict the outcome of different setups in the sealing process by the use of virtual models. In this thesis the Coupled Eulerian-Lagrangian formulation in Abaqus 6.10 has been evaluated as a tool for this.

In the journey towards this objective several important findings have been identified, especially concerning material properties of polymer materials during the fluid phase. It has been recognized that viscosity is the primary material parameter to define the resistance of motion for the polymer material. A general method of fitting experimental data to a surface function using the Cross and Arrhenius functions has been defined and the implementation in Abaqus by means of subroutines has also been discussed in the thesis.

Experimental work has been done to achieve a better understanding of how the sealing process, using ultra sonic vibrations, works and also to obtain referential results for this thesis and future work.

The experimental data shows that not only the polymer layer weakens due to heating but also the paperboard layers. It also shows that the deformed geometry of the packaging material, both polymer and paperboard is a direct consequense of the choice of anvil.

PREFACE

This master thesis was carried out at the Division of Structural Mechanics at LTH, Lund University, from June 2010 to January 2011.

First of all we would like to thank Eskil Andreasson at Tetra Pak for his guidance through the entire thesis. His inspiration during weekly meetings has been greatly appreciated. We would also like to thank Kent Persson at the Division of Structural Mechanics for his support, encouragement and paitience through the work.

Also a special thanks to the staff at Tetra Pak, Claes Oveby, Tommy Jönsson, Per CJ Johansson, Magnus Råbe, Nicklas Olsson and Evan Nashe for valuable support and to the staff at the Division of Structual Mechanics for interesting discussions and advice.

Finally, we would like to express our deepest love and gratitude to Rocio and Caroline for their support and understanding throughout our entire education.

Henrik Johansson & Jonathan Ollar
Lund, January 2011

CONTENTS

1. INTRODUCTION

Tetra Pak produces packages designed to contain liquid products e.g. milk and juice. These are low cost every-day products requiring low packaging costs while aspects such as hygiene, design and durability leads to high development and material costs.

In 2009 Tetra Pak sold over 145 billion packages [1]. Such quantities means that even the smallest reduction in production time or material usage leads to great profit. Thus the need for more accurate dimensioning of both package performance and packaging processes are increasing as the need for optimization grows.

Simulating the entire packaging process, from laminated paper and liquid product to finished packaged product is the long-term goal for Tetra Pak. This process involves several branches of physics, often referred to as multiphysics.

The most common methods for simulating such processes are the Finite Element Method (FEM) and the Finite Volume Method (FVM). FEM and FVM is numerical techniques for approximate solutions of partial differential equations. Commercial programs such as Abaqus [14] are widely used to analyze physical problems in solid mechanics, fluid dynamics, thermodynamics etc. using FEM.

This report is focused on the sealing process of the Tetra Pak packages. The sealings are made through compressing and heating of laminated paper through which the plastic laminates melt and merge. This requires interaction between the melted plastic laminates (Fluid mechanics) and the paper (Solid mechanics), a so called Fluid Structure Interaction (FSI) which recently has been implemented in Abaqus. The method for creating such interactions in Abaqus is called Coupled Eulerian Lagrangian (CEL) formulation which will be evaluated in this thesis.

1.1. Problem Formulation

Generally, a continuum mechanics problem may be described in the material or spatial description. The material description, also denoted the Lagrangian description tracks the material particles as they transport. The spatial, also denoted the Eulerian description keeps track of certain domains where material can flow in and out.

The Eulerian description is convenient for fluid dynamics problems where the material flows and the Lagrangian description fails because of mesh distortion. The Eulerian mesh is fixed and therefore better suited. In the case of the sealing process the polymer is expected to flow due to high temperatures and high pressure and is therefore modeled using the Eulerian description. The paperboard however is modeled using the Lagrangian description according to common practice at Tetra Pak.

The main task of the thesis is to evaluate if the recently implemented Coupled Eulerian-Lagrangian formulation in Abaqus is a proper strategy of modelling the the sealing process.

1.2. Objective

The main objective of this thesis is to evaluate the Coupled Eulerian-Lagrangian formulation in Abaqus for simulating the transversal sealing process of Tetra Pak packages. Included in the objective is getting a general knowledge of the sealing process, implementing a material model of the polymer and to gain experience of the Coupled Eulerian-Lagrangian formulations and its applicability to the sealing process.

1.3. Method

To achieve the objective of the thesis, knowledge of the sealing techniques used at Tetra Pak is needed as well as material properties. Theory of fluid mechanics in general and FE-theory will be studied. The main part of the thesis will contain physical experiments and computer simulations of the sealing process.

2. Sealing Technique at Tetra Pak

Tetra Pak produces a wide range of packages suitable for everything from beverages to food. The various packages are designed to meet different demands and is thus of different material compositions.

Generally the packaging material used in the packages is a paperboard laminate. In general there is an outer layer of polymer to protect the printed graphics, next there is a paper layer which has the purpose of providing stability to the package. On the inside of the paper there is another layer of polymer, an aluminum foil layer and yet another polymer layer. The polymer layers have the purpose of protecting the paper and the printing from moist, but they also serve as an adhesive layer in the sealing process.

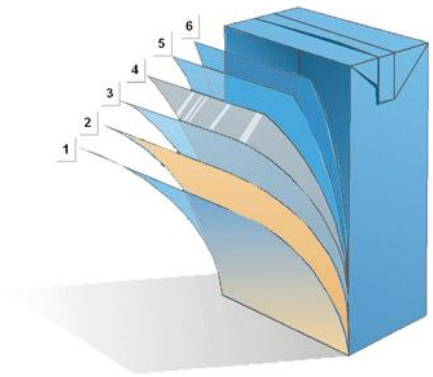


Figure 2.1: Material layers of a Tetra Brik Aseptic package. 1) Polymer 2) Paper 3) Polymer 4) Aluminum 5) Polymer 6) Polymer. [1]

The packaging material produced by Tetra Pak is delivered in rolls to the dairy and put into a filling machine. In the filling machine the packaging material is first sterilized, then folded into a tube and sealed longitudinally. Afterwards the tube is filled with the product and sealed in the transversal direction, making it a closed package as seen in Figure 2.2.

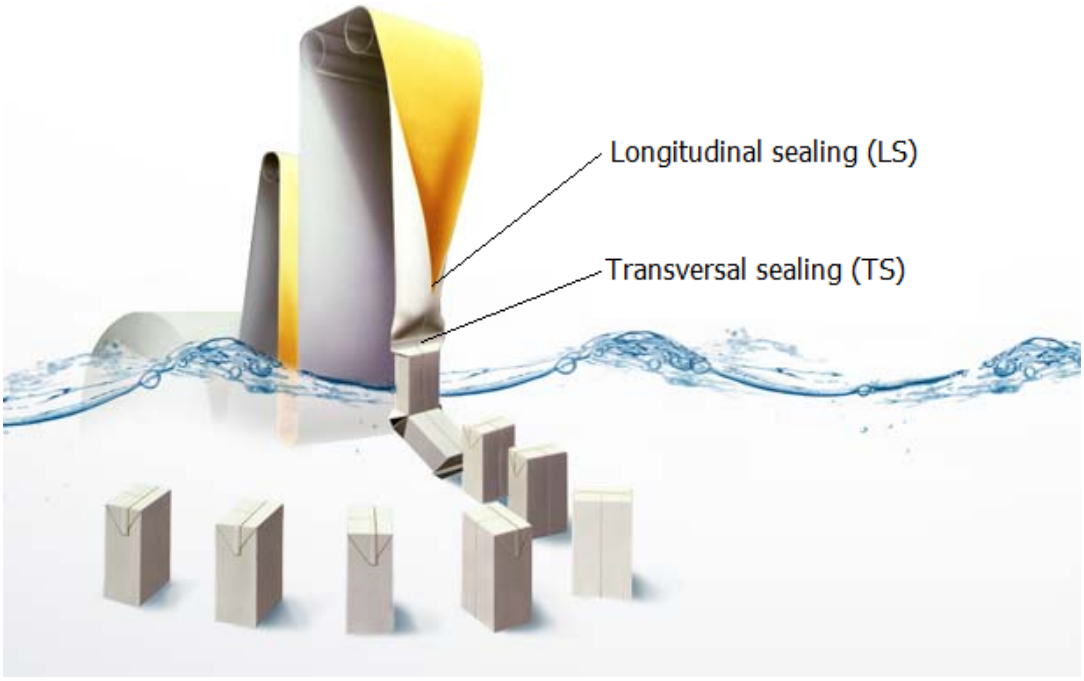


Figure 2.2: Principle sketch of the manufacturing of the longitudinal sealing (LS) and the transversal sealing (TS). [1]

The longitudinal sealing (LS) of the tube is created by overlapping the sheet edges, sealing the outside of one edge to the inside of the other. The inside edge is protected from soaking fluid by a plastic strip which is applied before the LS sealing is made. The transversal sealing (TS) on the other hand is sealed inside to inside by compressing the tube, creating a transversal sealed strip. At middle of the TS, the TS and LS meet which means that at a small area, there are three layers of paper sealed together.

The sealings are made by compressing the two packaging material layers while heating. This causes the plastic laminates to melt and merge. When cooled down the plastic laminates are once again solidified and the package is sealed.

There are different ways of heating the laminates. One way is through a magnetic field, inducing a current in packages containing an aluminum layer. This will cause the aluminum layer to heat up and through conduction the polymer layer will heat up and melt. An other way of heating up the polymer is through ultrasonic vibrations which causes heat.[3]

This report is focused on the TS process using ultrasonic vibrations which is shown in Figure 2.3.

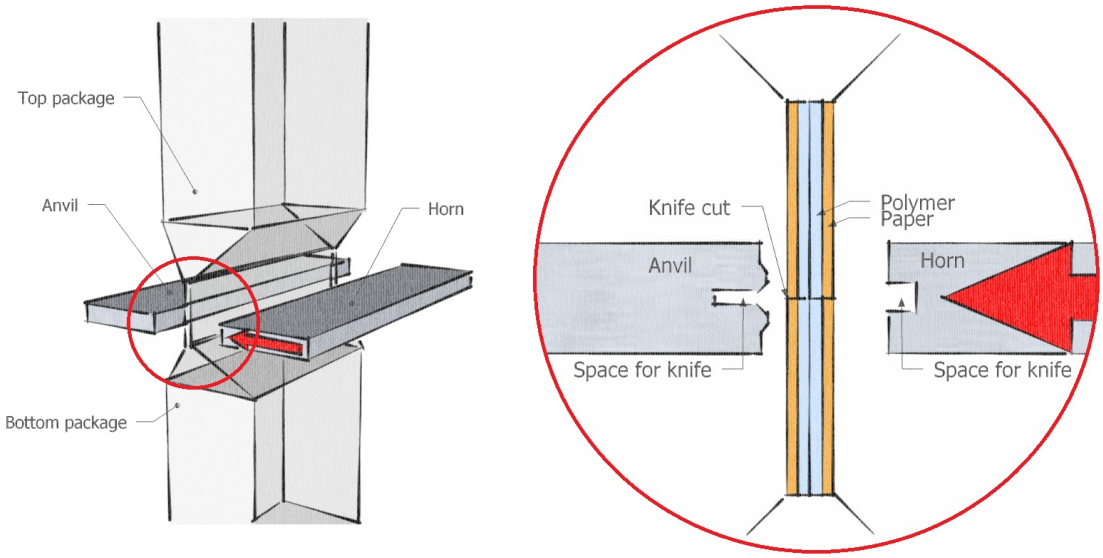


Figure 2.3: Principle sketch of the TS process, showing the different parts.

In this process a tuned horn compresses the packaging material against an anvil. The horn oscillates which causes frictional heat in the packaging material. The geometry of the horn is smooth while the anvil, that exists in various forms, has a special purpose geometry. The geometry is designed to apply pressure on the area which is to be sealed, and preventing the molten polymer to flow out from that area.

In designing the sealing process there are several parameters that influence the end result. Some of these parameters are the temperature of the polymer, the pressure on the packaging material, the anvil geometry and the oscillating time of the horn.

Computer models of the sealing process can be used as an engineering tool in product or machine development. To be able to create accurate models, knowledge of how molten polymer flow when exposed to high temperature and pressure is of great interest. This brings us to the field of rheology, which will be discussed in chapter 4.

3. Experimental Transversal Sealing Tests

In this chapter experimental tests of the transversal sealing process is performed with the main focus on characterizing the polymer flow for various anvil geometries.

The experiment was carried out at Tetra Pak in Lund using a test rig shown in Figure 3.1. The test rig is designed by Tetra Pak for the purpose of analyzing the transversal sealings in various packaging setups by isolating the components from the filling machine needed for the transversal sealing.

The machine is designed to compress a tube, made from folded package material, with a predefined force and generated heat using ultrasonic vibrations of the horn caused by the linear servo actuator. This melts the polymer and thus seals the sheets of the tube together. There are several options for making an user defined experiments such as changing the force compressing the paper over time and the energy input from the linear servo actuator. The output of the test is the measured force and linear servo actuator energy over time as well as the change in distance between the horn and the anvil, measured by two distance sensors shown in Figure 3.1. Visual results can be obtained by making a cut through the sealing and taking photographs by means of a microscope.

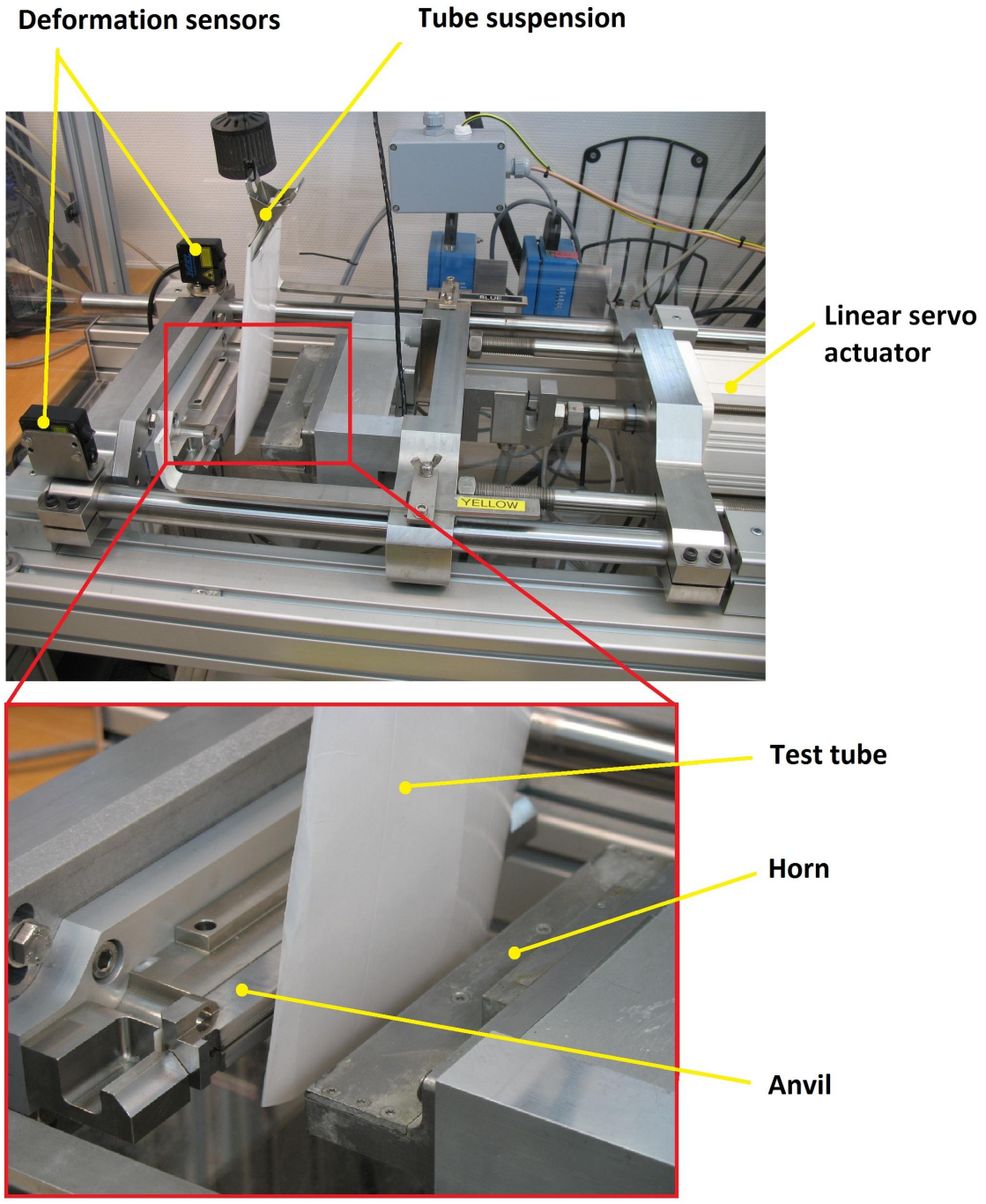


Figure 3.1: Top picture shows an overview of the test rig. Bottom picture shows an enlargement of the sealing area.

3.1. Design of Experiments

The purpose of the experiment was to determine the polymer flow in the sealing process and differences in polymer flow for different anvil geometries. Another goal was to have referential results for both visual and numerical comparison of the computer models.

The available settings for the test rig were the force, anvil geometry, packaging material and the heating energy. The heating energy was controlled by specifying the duration of the oscillations. To track the polymer flow during the sealing process, tests with different durations of oscillations was used. Micrographs of the sealings was made for every test with the aim of tracking the position of the polymer. To investigate the difference in polymer flow due to different anvil geometries, three anvils were used. Also two different packaging materials were used to investigate its effect on the polymer flow.

The experimental setup was planned by means of a so called P-diagram, as shown in Figure 3.2 which allows for experimental testing in a structured way. [5]

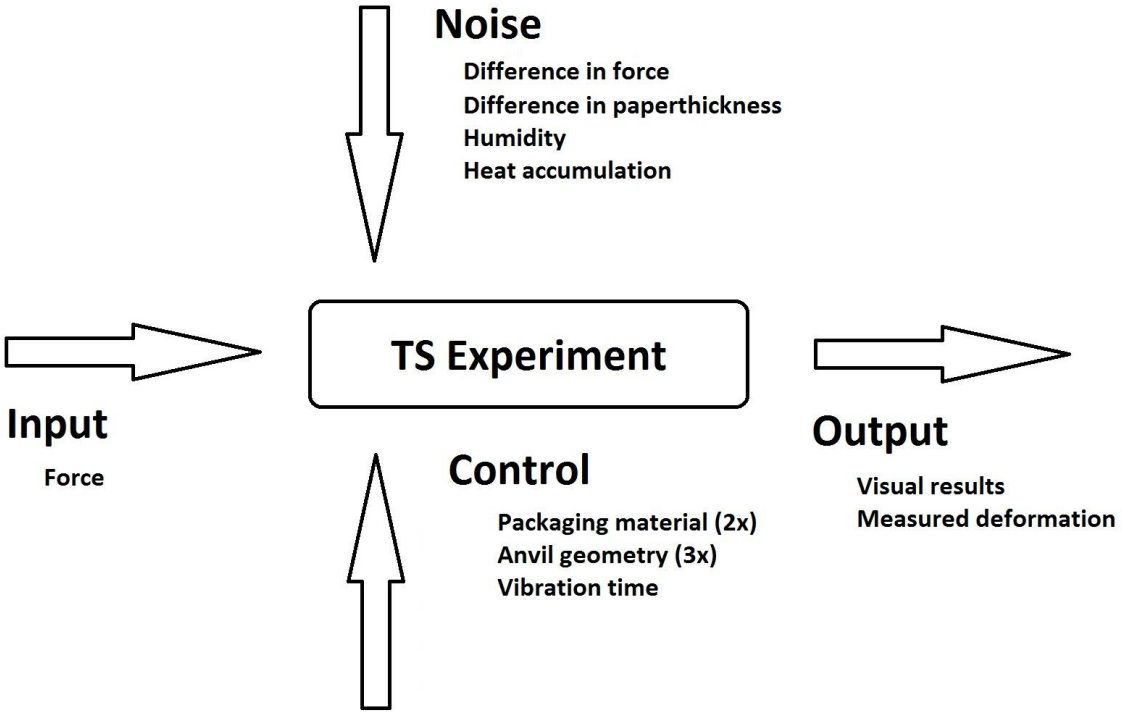


Figure 3.2: P-Diagram.

In the P-diagram the input is the parameters that are not intended to vary from test to test. In this specific series of tests the input was the force over time as shown in Figure 3.3. The force was initially set to 500 N and after 300 ms instantly changed to 4000 N. After another 500 ms the force was released in two steps.

The Control factors in the P-diagram are the parameters that are manually varied from test to test. The energy input was controlled by restricting the maximum time that the linear servo actuator was allowed to generate vibrations. One test was performed without a heat pulse as a reference and another five tests were performed where the heat pulse started after 400 ms as shown in Figure 3.3 and end after 30, 60, 90, 120 or 150 ms to be able to track the polymer flow as it melts.

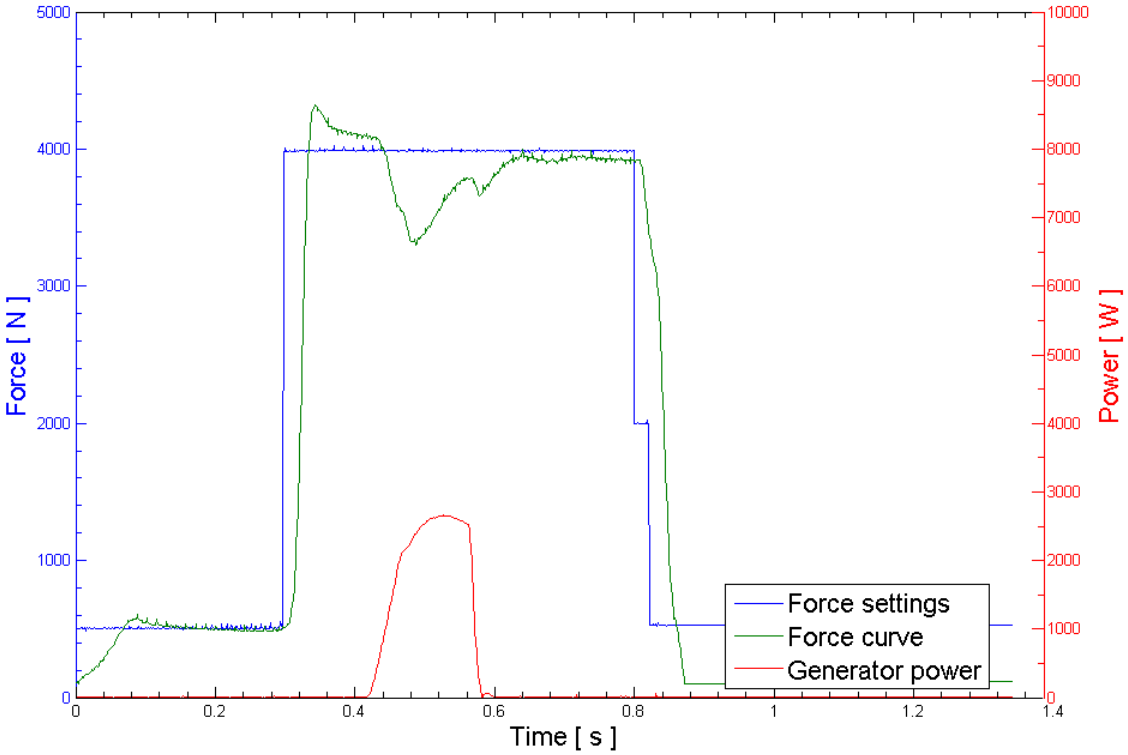


Figure 3.3: Settings for the experiment. The force starts at 500 N and is changed to 4000 N at 300 ms. The oscillations which causes the heat input starts at 400 ms.

In the test, three different anvil geometries shown in Figure 3.4, was used with the aim of detecting characteristic results for each anvil. Two different types of packaging material, denoted P1 and P2, was used to identify the influence of various packaging material on the polymer flow.

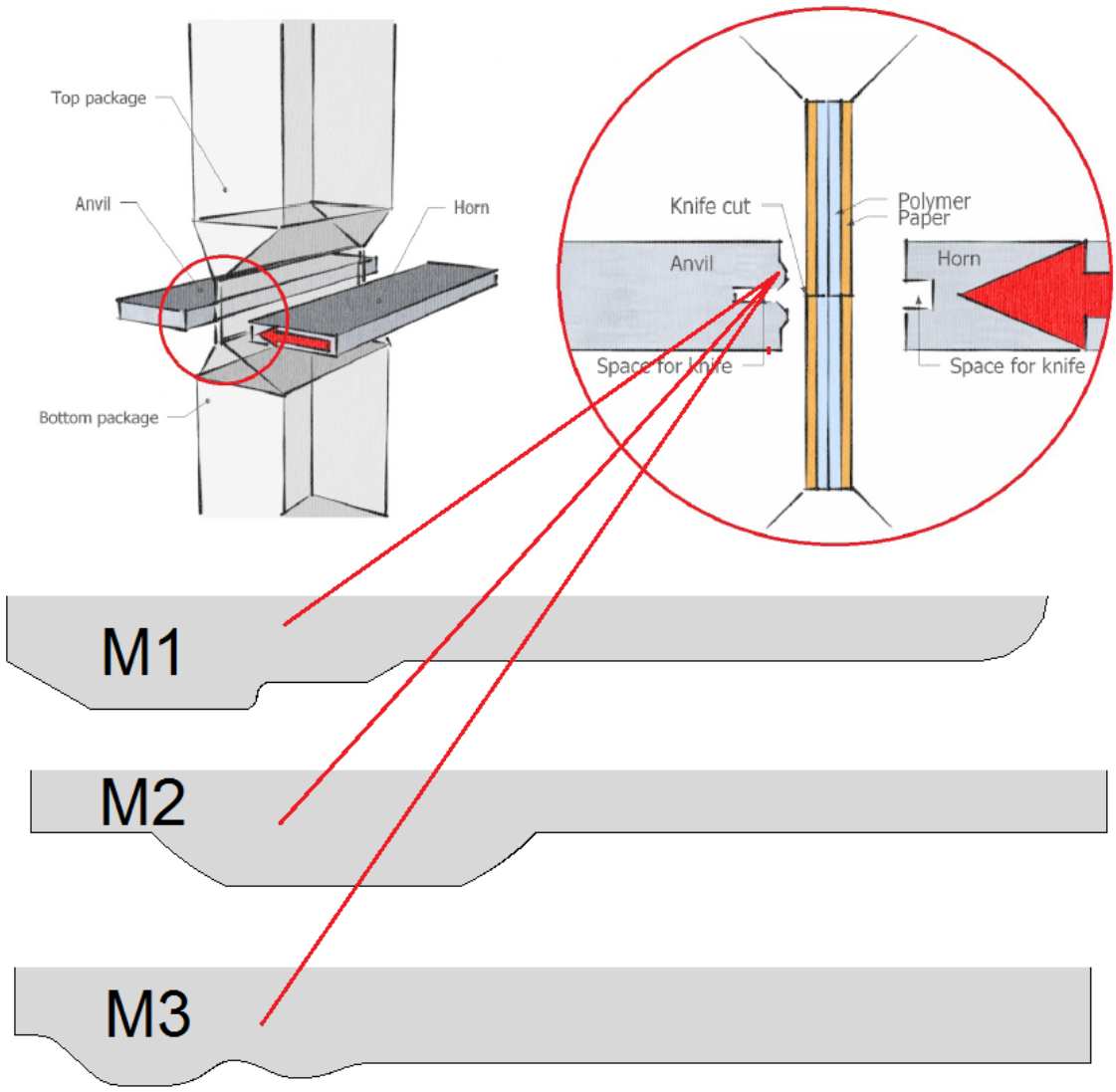


Figure 3.4: Principle sketch of the TS and the geometry of the three anvils used in the experimental testing.

The desired outputs from the tests was visual results as discussed earlier and measured deformation from the deformation sensors.

All tests are affected by noise in some manner. Noise are factors that are not controlled but may influence the result. Identified factors of this kind in this experiment was that the measured force may differ slightly from the force setting, the paper thickness may differ slightly, the humidity in the test location may differ and the test rig might be affected by heat accumulation through frequent testing.

The paper thickness was measured using a machine for this purpose at Tetra Pak. Five measurements were made which showed that the mean thickness of P1 was 448 μm with a standard deviation of 2 μm and the mean thickness of P2 was 466 μm with a standard deviation of 3 μm which means that the paper thickness do deviate but is assumed to have very little influence on the end results.

To minimize the impact of humidity changes the packaging materials were stored in the test location a week prior to the test. The heat accumulation in the rig during the test was negligible due to time consuming preparations between each test.

The experimental planning as described above is shown in Figure 3.5.

#	Anvil	Paper type	Heat pulse [ms]
1	M1	P1	0
2	M1	P1	30
3	M1	P1	60
4	M1	P1	90
5	M1	P1	120
6	M1	P1	150
7	M1	P2	0
8	M1	P2	30
9	M1	P2	60
10	M1	P2	90
11	M1	P2	120
12	M1	P2	150

#	Anvil	Paper type	Heat pulse [ms]
13	M2	P1	0
14	M2	P1	30
15	M2	P1	60
16	M2	P1	90
17	M2	P1	120
18	M2	P1	150
19	M2	P2	0
20	M2	P2	30
21	M2	P2	60
22	M2	P2	90
23	M2	P2	120
24	M2	P2	150

#	Anvil	Paper type	Heat pulse [ms]
25	M3	P1	0
26	M3	P1	30
27	M3	P1	60
28	M3	P1	90
29	M3	P1	120
30	M3	P1	150
31	M3	P2	0
32	M3	P2	30
33	M3	P2	60
34	M3	P2	90
35	M3	P2	120
36	M3	P2	150

Figure 3.5: Experimental planning listed in chronological order.

3.2. Experimental Results

During the experimental tests the energy input from the linear servo actuator dropped continuously as shown in Figure 3.6. This may have caused some uncertainty in the measurements.

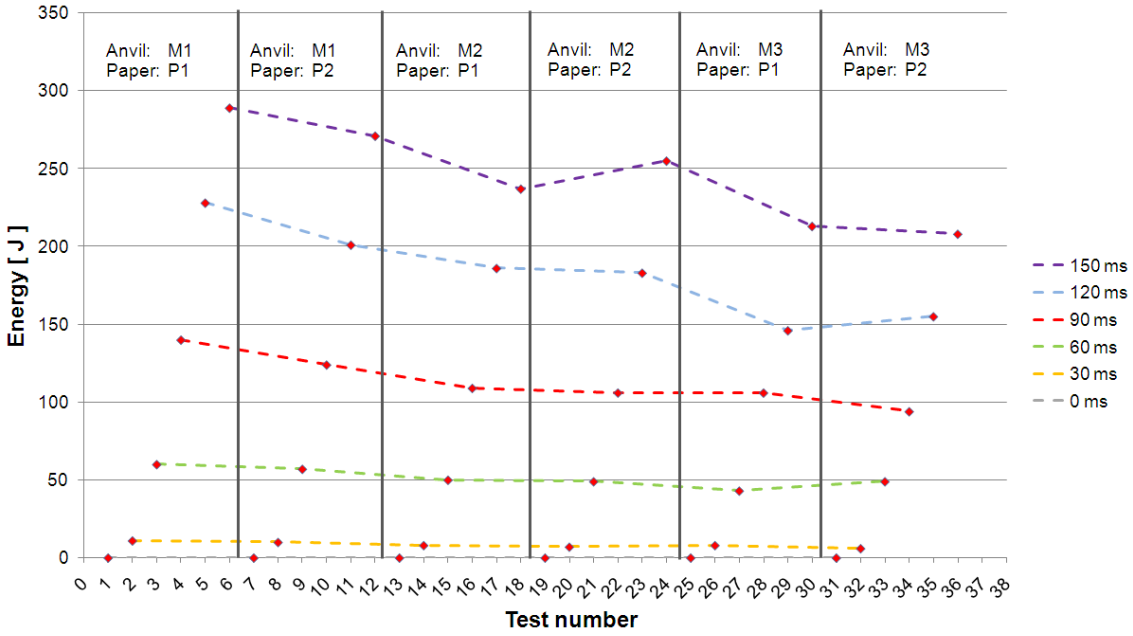


Figure 3.6: Graph of the sealing energy as functions of the test number in chronological order.

As mentioned before, each experiment delivers two kinds of data output. Numerical data from the rig and visual data from the micrographs. Figure 3.7 shows the numerical data from the experiment performed with anvil M1 and paper P1 with no heat pulse and with a heat pulse of 150 ms.

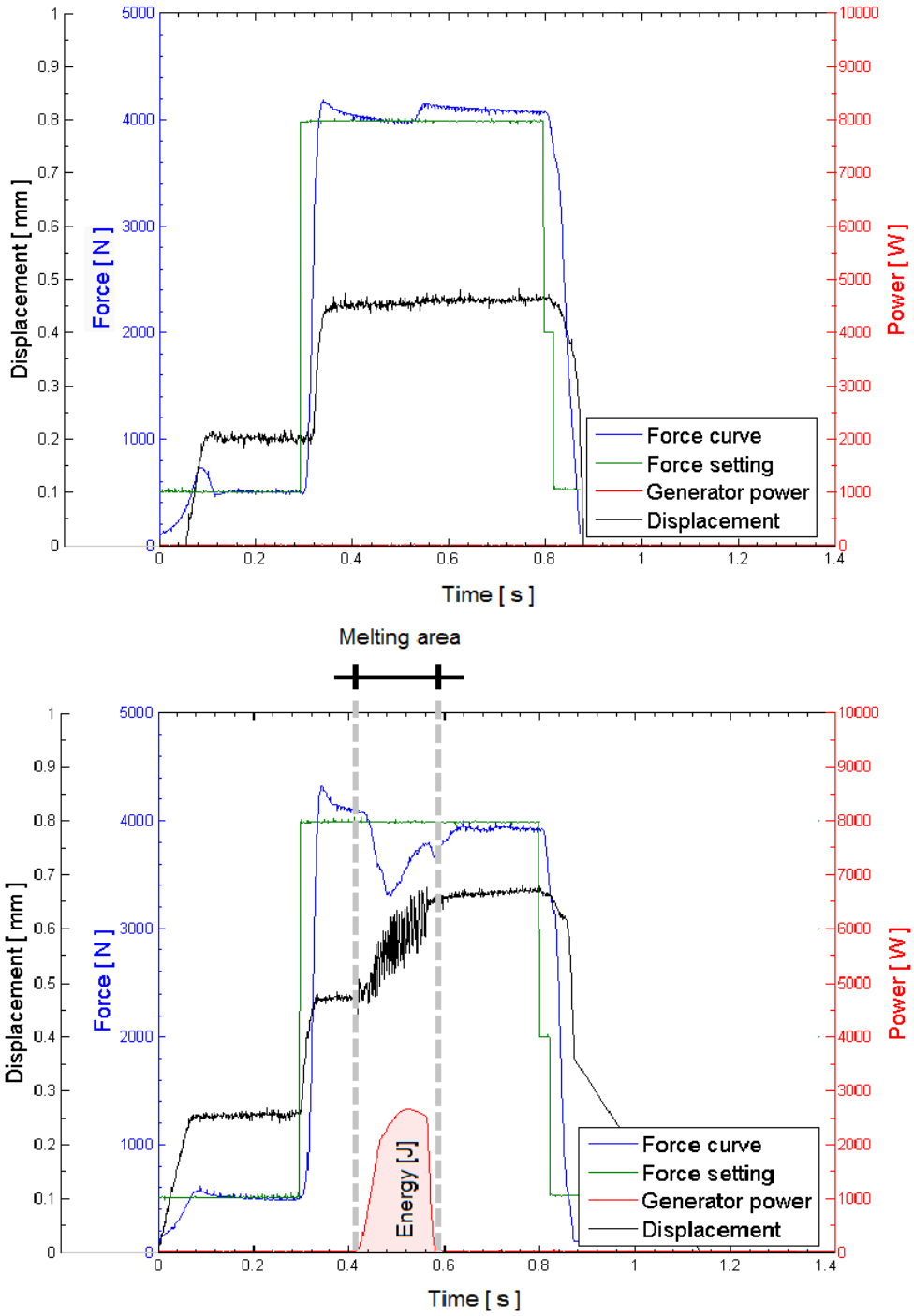


Figure 3.7: Experimental test graphs from anvil M1 with paper P1. The upper graph shows results using no heat pulse and the lower graph shows results using a heat pulse length of 150 ms.

In the upper graph in Figure 3.7 the results with no heat pulse is shown. The graph indicates that the package material is deformed due to both the initial force and the force increase at 300 ms and that the deformation is constant when the force is constant. However when the heat pulse is applied there is additional deformation due to the temperature change of the material in the package material. This can be seen in the lower graph in Figure 3.7.

The deformation increase due to the heating is roughly 0.2 mm which is significantly larger than the thickness of the polymer layer. This means that the deformation is not only because of polymer flow but also because of a deformation increase of the paper layer.

Due to a slow regulation of the force in the test rig the measured force deviates from the force setting. Especially during the heat pulse when heat weakens the package material. This causes the force to reduce for a while until it stabilizes.

The graph also shows the power input over time which can be integrated to obtain the energy in Joule.

The behavior described above is generally the same for all anvils and paper types. To see differences when changing anvil, visual results using micrographs are used.

During the extraction of the results it turned out to be rather complicated to make clear cuts of the test pieces sealed with low energy. The reason is that the test specimen is tightly strapped during the cut which means that small deformations of the specimen will be hard to visualize, even in a microscope. Cuts were made on the specimens sealed with a 150 ms heat pulse with the focus on distinguishing characteristic differences between the anvils.

The microscopic pictures from anvil M1, M2 and M3 using 150 ms heat pulse is shown in Figure 3.8. The pictures show that depending on the anvil geometry the polymer flow differs. Using anvil M2 leads to a more even distribution of the polymer while using M1 and M3 leads to accumulations of the polymer.

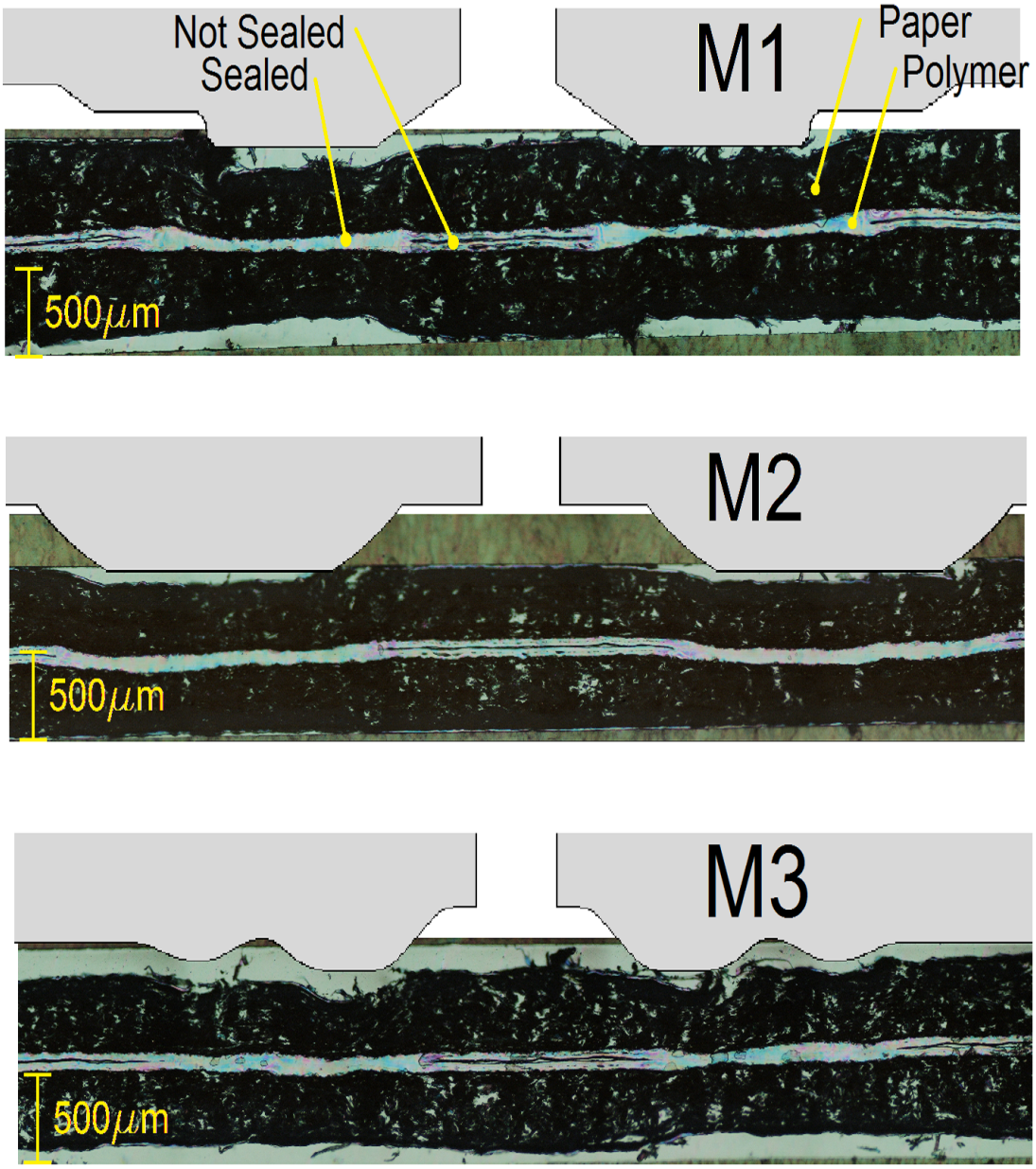


Figure 3.8: Microscopic pictures of the TS for the three anvils M1, M2 and M3 using a heat pulse of 150ms. Areas which are not sealed can be distinguished by a dark line between the polymer layers.

4. RHEOLOGY OF POLYMERS

This thesis is focused on the polymer flow in the package during the sealing process in which the rheology (study of material flow) of polymers is a key subject.

In the Tetra Pak sealing process both solid and fluid state polymers are involved. Before the heating starts the polymer is solid, then melts as the temperature rises. Capturing the phase transition is crucial to understanding the sealing process.

Polymers at high temperatures will experience fluid behavior. Fluids have numerous characterizing properties but the emphasis in this report is to describe the properties which affects the motion of the polymer, and the corresponding material parameters.

A fluid subjected to shear stress(no matter how small) will, unlike a solid, continuously deform for as long as the shear stress is applied. This means that a fluid at rest can not be subjected to any shear stress. This is called a state of hydrostatic stress and implies that the pressure is uniform in all directions. As an example of this, consider a glass of water in which a certain fluid particle experiences uniform pressure from all directions and is therefore at rest.

A fluid particle subjected to a constant shear stress will however deform at a constant rate. The relation between the shear stress and the shear rate is called the viscosity and will be discussed next.

4.1. Viscosity of Polymers

The viscosity is defined as the deformation resistance of a fluid subjected to forces. Figure 4.1 illustrates a plane 2D case of fluid flow. The fluid flows in the direction of the x-axis with a velocity profile illustrated in the left picture. An infinitesimal element of the fluid is shown in the right figure.

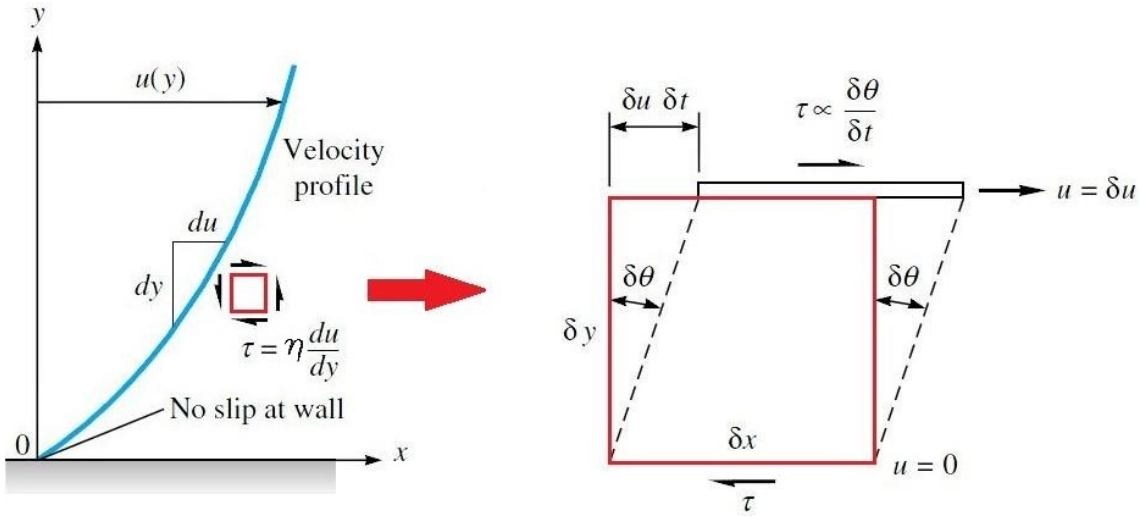


Figure 4.1: Left picture shows plane case of fluid flow, and right Figure shows an infinitesimal part of the fluid flow. [7]

The shear rate is defined as $\dot{\gamma} = \frac{\delta\theta}{\delta t} = \frac{\delta u}{\delta y}$ and is proportional to the shear stress. i.e.

$$\tau = \eta \cdot \dot{\gamma} \tag{4.1}$$

where τ [Pa] is the shear stress, η [Pa s] is the viscosity , and $\dot{\gamma}$ [1/s] is the shear rate.

As previously mentioned the viscosity is a measurement of flow resistance. Eq.(4.1) indicates that higher viscosity equals higher resistance to flow. As an example, water has a viscosity of 0.001 Pa s while syrup has about 10 Pa s. The viscosity is temperature dependent where high temperature equals low viscosity.

The viscosity can also be shear rate dependent. If the viscosity is independent of shear rate, it is said to be Newtonian. Otherwise it is referred to as non-Newtonian. Water for instance is a Newtonian fluid as stirring does not change the viscosity. Paint however is a non-Newtonian fluid as it thins out and flows more easily for stirring at higher rates. This type of thinning non-Newtonian response is called pseudo plastic and a shear thickening viscosity is called Dilatant. Different types of viscosity behaviors are shown in Figure 4.2.

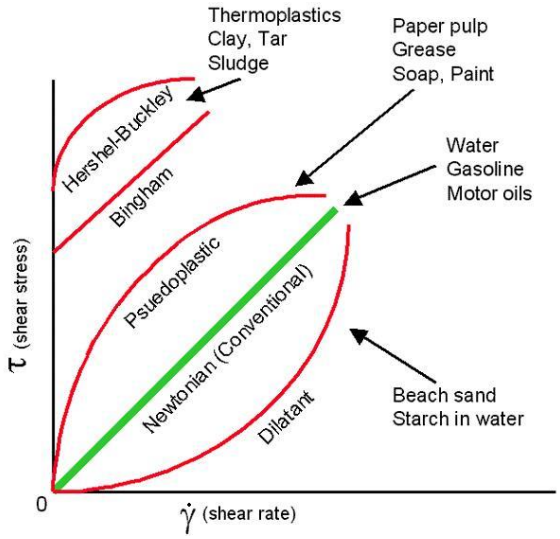


Figure 4.2: Viscosity graph - shear rate dependency. Newtonian and non-Newtonian behavior. The slope of the curves is the viscosity. [20]

Molten polymers have pseudo plastic behavior and the viscosity is then dependent on both temperature and shear rate. There are several available viscosity models such as Cross, Carreau-Yasuda and Herschely-Bulkey for shear rate dependency as well as Arrhenius and Williams-Landell-Ferry for temperature dependency. Since we are dealing with polymers that have a viscosity that depend on both shear rate and temperature, a combination of the Cross- and Arrhenius models provide a good correlation to experimental tests [12].

In Figure 4.3 a typical viscosity shear rate curve of a polymer is shown. To fit data to the Cross model the following equation is used

$$\eta = \eta_\infty + \frac{\eta_0 - \eta_\infty}{1 + (\frac{\eta_0}{\tau^*}\dot{\gamma})^{1-n}} \tag{4.2}$$

Where $\dot{\gamma}$ is the shear rate, η_0 is the viscosity at zero shear rate, η_∞ is the viscosity for infinite large shear rates, n is the Cross rate constant which describes the slope of the curve in the shear thinning region or region where the viscosity change due to the shear rate, $\frac{\eta_0}{\tau^*} = \lambda$ is called the Cross time constant where $\frac{1}{\lambda}$ describes where the curve exits the Newtonian plateau or region where the fluid doesn't change with shear rate and enters the shear thinning region. For practical use η_∞ is often set to zero, since $\eta_0 - \eta_\infty \approx \eta_0$. [2]

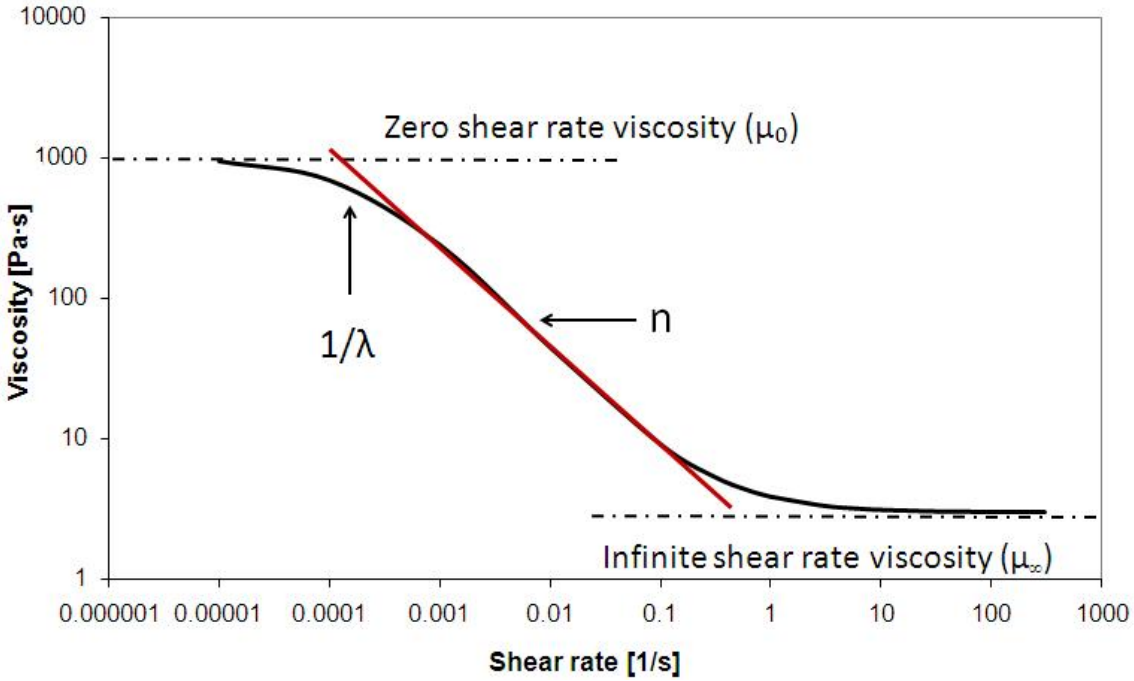


Figure 4.3: Example of Cross model properties.

To make the Cross function dependent of temperature a so called shift function is introduced. The shift function describes the change in zero shear rate viscosity from one reference temperature to a temperature of choice according to [12]

$$a_t = \frac{\eta_0(T_{ref})}{\eta_0(T)} \tag{4.3}$$

Where T is the temperature of choice and T_{ref} is the reference temperature. Inserting eq.(4.3) into (4.2) yields

$$\eta = \left(\frac{\eta_0}{1+(\frac{\lambda\dot{\gamma}}{a_t})^{1-n}}\right)\left(\frac{1}{a_t}\right) \tag{4.4}$$

and a_t is given by the Arrhenius law according to

$$\ln(a_t) = \frac{E_0}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right) \tag{4.5}$$

where T is the temperature of interest measured in Kelvin, $R \approx 8.3144$ is the universal gas constant, E_0 is the activation energy, T_{ref} is the reference temperature measured in Kelvin, a temperature for which viscosity as a function of shear rate is known [12].

4.2. Sealing Process Polymers

In the transversal sealing application there are different types of polymers used. The available data for one of these polymers, shown in Figure 4.4, was the viscosity at temperatures 130, 150 and 170 ^oC as a function of the shear rate.

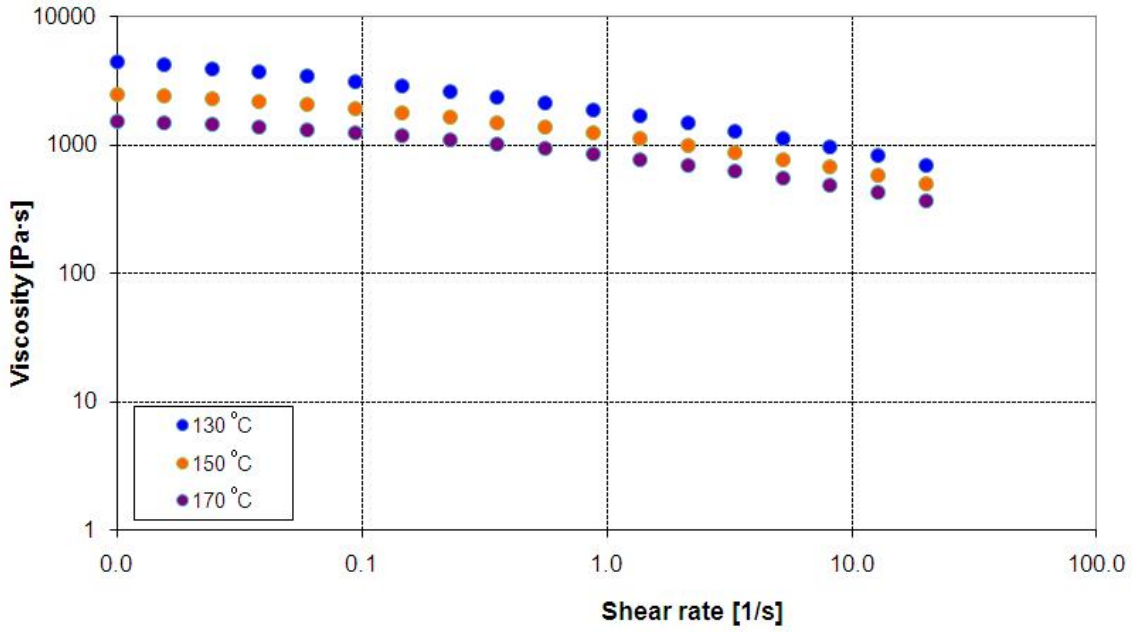


Figure 4.4: Experimental test data of the viscosity vs. shear rate of the sealing polymer at 130, 150 and 170 ^oC.

To fit the experimental test data in Figure 4.4 to the Cross-Arrhenius equation (4.4) the activation energy, was first determined by inserting eq.(4.3) in eq.(4.5) yielding

$$\ln(\eta_{0,ref}) - \ln(\eta_0) = \frac{E_0}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right) \Leftrightarrow \Delta\ln(\eta_0) = \frac{E_0}{R}\Delta\frac{1}{T} \tag{4.6}$$

The activation energy can be obtained by plotting $\ln(\eta_0)$ with respect to $\frac{1}{T}$ where the $\frac{E_0}{R}$ is the determined curve fit to the slope of the curve. The three viscosity curves was individually fitted to the cross function eq.(4.2) to obtain η_0 of each function. The $\ln\eta_0 - \frac{1}{T}$ plot is shown in Figure (4.5).

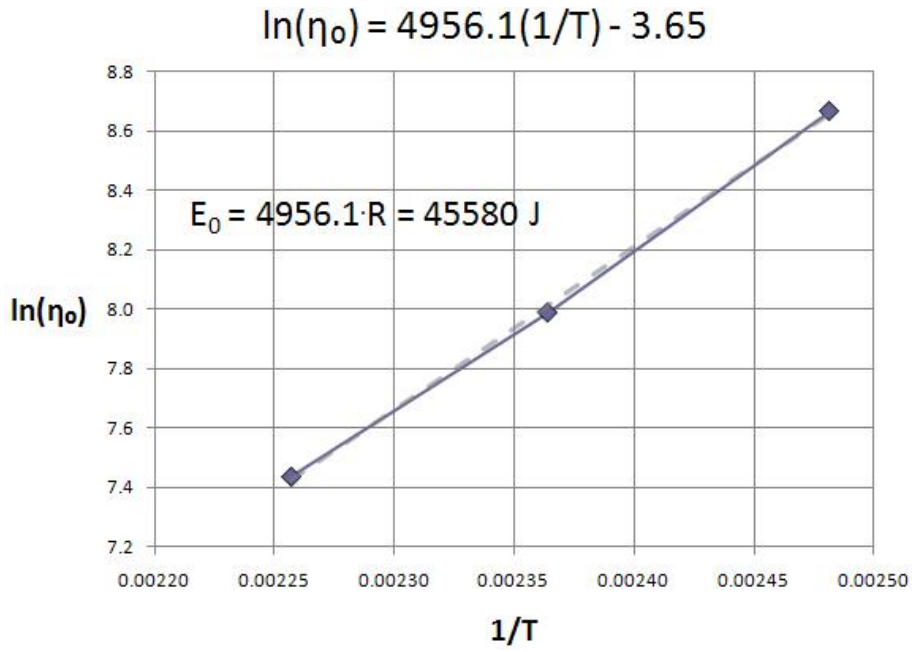


Figure 4.5: $\ln \eta_0$ vs. to $\frac{1}{T}$ for the three known temperatures. The dashed line is a linear fit to these values.

The activation energy was determined to $E_0 = 45580J$ for this sealing polymer. After E_0 was established, it was inserted in eq.(4.4) with the information from the Cross function of the reference temperature, creating a viscosity field shown in Figure 4.6.

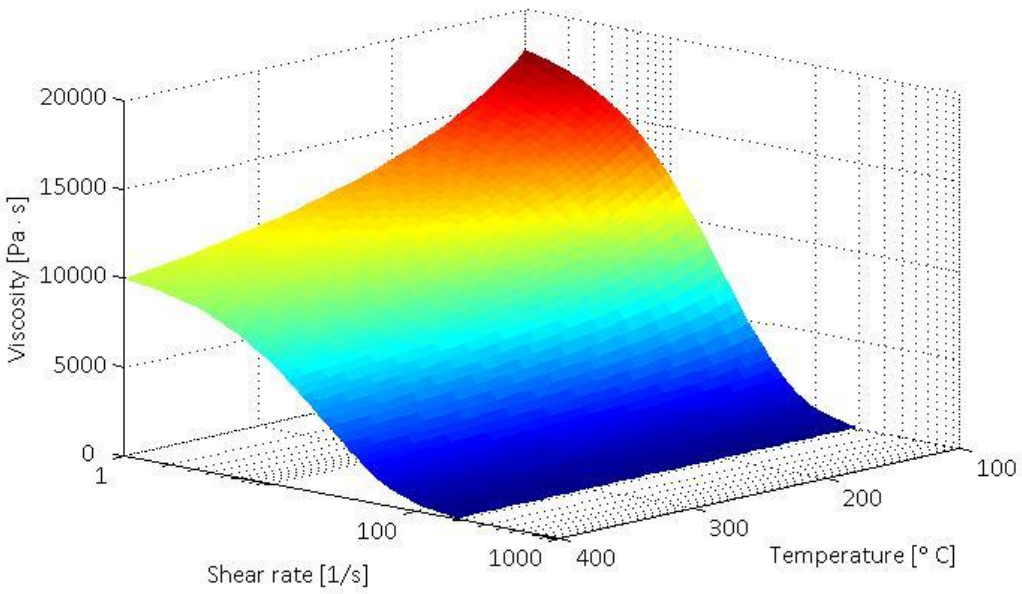


Figure 4.6: Viscosity as a function of shear rate and temperature for a polymer.

To verify the viscosity, the three viscosity curves is once again plotted along with the Cross-Arrhenius function, as shown in Figure 4.7. The value R^2 stated at each curve is the coefficient of determination, which is calculated as

$$R^2 = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \bar{y})^2} \tag{4.7}$$

where y_i is the measured value, $\bar{y}$ is the mean value of y_i and f_i is the calculated value. The R^2 value is a measurement of the compliance of the model and ranges from zero to one where one is a perfect fit.

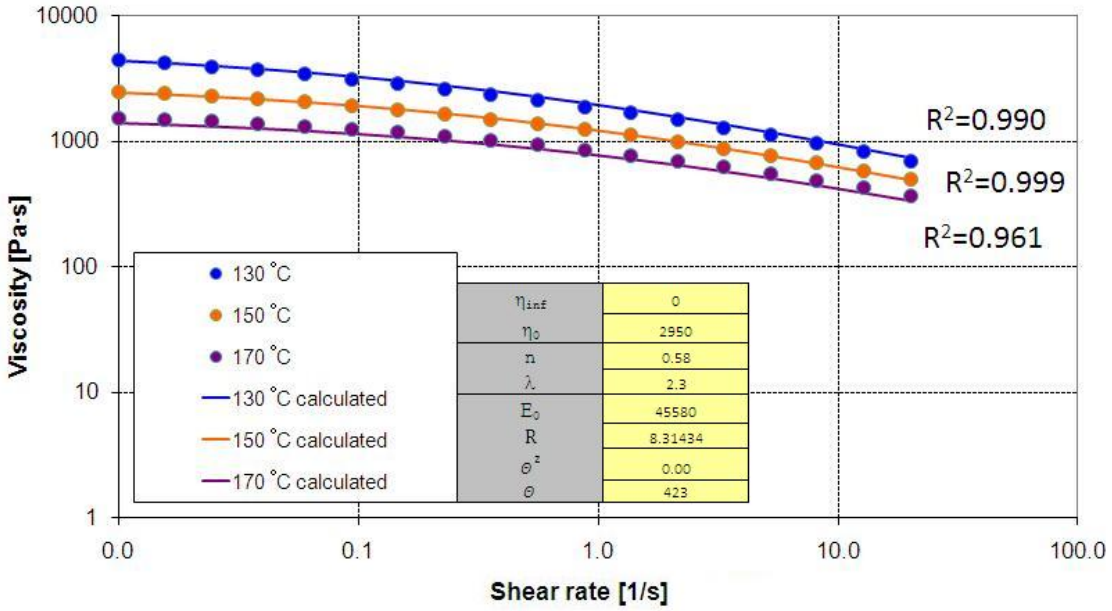


Figure 4.7: Experimental test data and calculated data of the viscosity field at the three temperatures 130, 150 and 170 ^oC. R^2 is the coefficient of determination where 1 is a perfect match.

5. THEORY

Continuum mechanics is the analysis of the kinematic and mechanical behavior of materials modeled on the assumption that the mass is continuous instead of discrete particles and voids. There are two major fields in Continuum mechanics, namely solid mechanics and fluid mechanics which are based on the basic principles of mass-, momentum- and energy conservation. These equations are the same irrespective of the continuum field. By introducing a constitutive law the material behavior is defined and the general equations are reduced accordingly.

5.1. Eulerian and Lagrangian Description

In continuum mechanics there are two available descriptions in analysis of continuous media. The Lagrangian- and the Eulerian formulation, Figure 5.1.

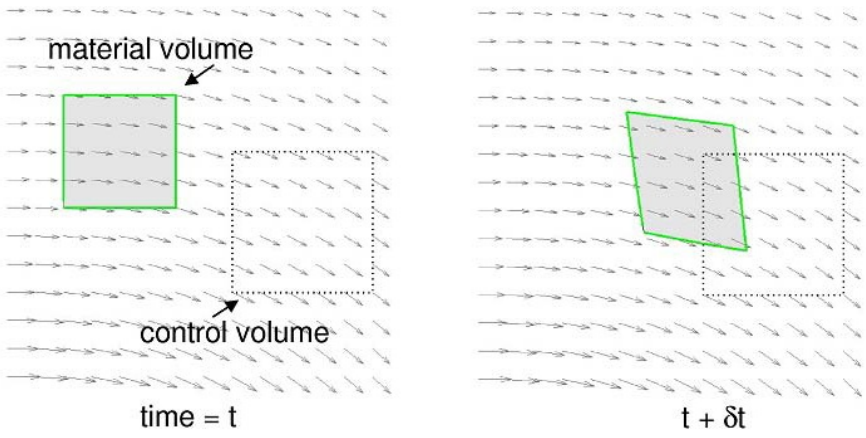


Figure 5.1: Lagrangian (Material) description and Eulerian (Spatial) description in a velocity field. The Lagrangian volume deforms with the material while the Eulerian volume is fixed. [8]

In the Lagrangian (or material) formulation all equations are in terms of a referential, or initial, configuration. This can be thought of as following a material point as it moves in

space with time. It is convenient to use the Lagrangian formulation in solid mechanics since the material follows the deformation and constitutive laws often are in terms of total deformation.

In the Eulerian (or spatial) description the equations are in terms of a current position. Often you have a certain area of interest and views what happens in that particular area. This is often used in fluid mechanics where constitutive laws are deformation rate dependent (c.f. section 3.1) and independent of the total deformation.[6]

In finite element modeling there are also advantages and disadvantages of the different formulations. The main difference is in the mesh where the nodes of the Lagrangian mesh is coincident with the material points while the Eulerian nodes are fixed in space. Eulerian meshes can therefore be used for analysis involving large deformation without remeshing while Lagrangian meshes can't due to distortion (see fig 5.2). This is the main reason for use of the Eulerian formulation in fluid mechanics, where the material flows through the mesh.

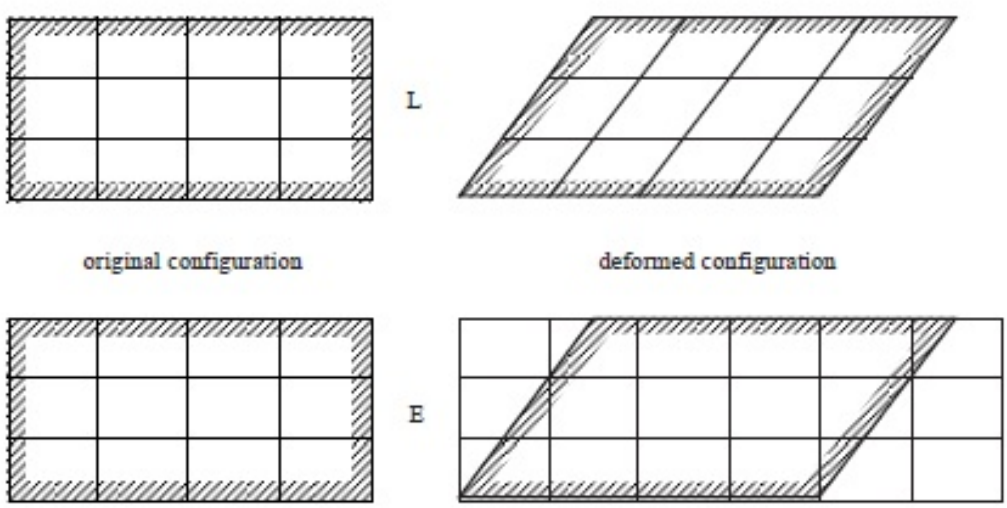


Figure 5.2: Lagrangian (L) and Eulerian (E) description of 2D pure shear scenario, where Lagrangian mesh develops distortion but the Eulerian mesh does not. [4]

In solid mechanics boundary conditions such as surface pressure, point loads or constraints are often used. As Lagrangian mesh nodes coincides with the material nodes, Lagrangian boundary's will coincide with material boundary's, whereas Eulerian boundary's in general will not. This is the main reason for using the Lagrangian formulation in Solid mechanics.[4]

5.2. The Coupled Eulerian-Lagrangian Formulation

In the Tetra Pak sealing process solid (paper) and fluid (molten polymer) materials interact. The solid material is preferably modeled as a Lagrangian region while the fluid material will be heavily distorted and thus must be modeled using the Eulerian formulation.

Coupling between fluid and solid mechanics, so called Fluid Structure Interaction (FSI) has been implemented in several commercial finite element programs. Abaqus uses a technique called Coupled Eulerian Lagrangian(CEL) formulation where contact conditions are applicable between Eulerian and Lagrangian regions.

The CEL formulation allows interaction between Lagrangian and Eulerian regions by enforcing contact.

In Abaqus only a three dimensional Eulerian element exists, namely the EC3D8R which is a Eulerian Continuum 3D element with reduced integration. Abaqus keeps track of the volume fraction of fluid in each element as shown in Figure 5.3.

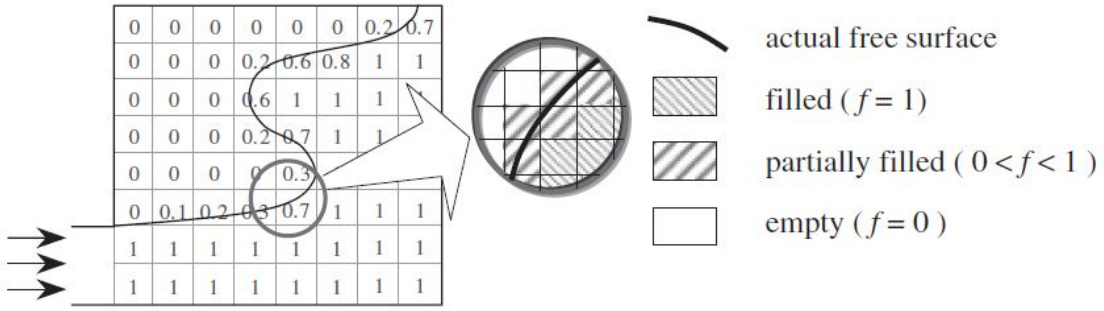


Figure 5.3: The volume of fluid method which shows the fraction of fluid in each element. [11]

This method is called the volume of fluid (VOF) method and the key is to be able to determine the location of the free surface of the fluid. When the volume fraction of the fluid in an element is between zero and one the element is partially filled and thus the surface of the fluid must be in that element. The surface of the fluid is then approximated with the information for the element and the surrounding elements. This is calculated for each time increment in the analysis.

The general calculation procedure is shown in Figure 5.4. First a Lagrangian approach is used where the mesh deforms, after which the material flow between the connecting elements is calculated. The mesh is then restored and the volume fraction is updated.

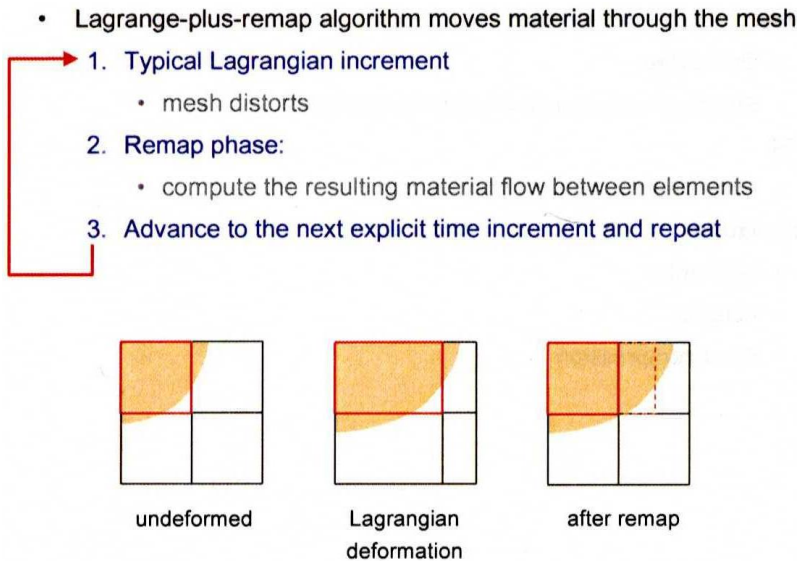


Figure 5.4: General calculation procedure of Eulerian elements in Abaqus. Lagrangian computation of deformation, calculation of material flow between elements and remapping of mesh with updated volume fraction. [14]

5.3. Explicit Solution

The Coupled Eulerian-Lagrangian Formulation requires the use of an Explicit analysis procedure which has certain aspects to enlighten.

The finite element discretization for a dynamic case can be written on the form

$$\mathbf{M}\ddot{\mathbf{u}} = \mathbf{P} - \mathbf{I} \tag{5.1}$$

where $\mathbf{M}$ is the Mass matrix, $\ddot{\mathbf{u}}$ the accelerations, $\mathbf{P}$ the external forces and $\mathbf{I}$ is the internal forces.

This is a set of differential equations which can be transformed to algebraic equations using the Newmark time integration scheme according to

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta t\dot{\mathbf{u}}_n + \frac{\Delta t^2}{2}\left[(1-2\beta)\ddot{\mathbf{u}}_n + 2\beta\ddot{\mathbf{u}}_{n+1}\right] \\ \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + \Delta t\left[(1-\gamma)\ddot{\mathbf{u}}_n + \gamma\ddot{\mathbf{u}}_{n+1}\right] \end{aligned} \tag{5.2}$$

where n denotes the current state where all quantities are known, while n+1 denotes the next step. β and γ are parameters to be chosen depending on the wanted solution procedure.

Choosing the parameters $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ results in a system of coupled equations which is computationally expensive. This is called an implicit procedure and it is unconditionally stable.

Setting $\beta = 0$ and $\gamma = \frac{1}{2}$ together with a lumped mass matrix results in uncoupled equations which can be solved separately at low computational cost. This is called an explicit procedure. This kind of procedure is only conditionally stable and to obtain a stable solution the time increment must be sufficiently small. This makes explicit analysis suitable for short load durations.[9, 10, 14]

6. Fluid Structure Interaction Simulations in Abaqus

This chapter is dedicated to bringing up topics of interest to Abaqus users who wish to use the Coupled Eulerian Lagrangian formulation for Fluid Structure Interaction.

Further on, this chapter will focus on specific topics related to CEL which have been discovered during the work. Some topics can be found in the Abaqus manual [14] and others are experience from trial and error testing. It is possible to work with different systems of consistent units in Abaqus but in this report SI units have been used.

6.1. Time Aspects in Abaqus/Explicit

Abaqus/Explicit solves the continuity, motion and energy equations, reduced by the constitutive law and the equation of state in an explicit manner, cf. chapter 5.

To obtain a stable solution the critical time incremenent is calculated automatically by Abaqus/Explicit but by manipulating its controlling factors the total analysis time may be reduced.

The time increment in Abaqus/Explicit can be estimated by the equation

$$\Delta t = min(L/C_d) \tag{6.1}$$

where L is the characteristic element length, which approximately is the smallest element edge, and C_d is the dilatational wave speed of the material. This estimation does not take into account the effects of damping or contact conditions. In CEL this turns out to be an overestimation. Throughout this thesis the the stable time increment has been estimated through data checks in the job module in Abaqus.

The parameters affecting the time increment has been found through parameter variation of small models. Those parameters are the mesh size, density, reference sound speed and the viscosity. According to Abaqus [17] the contact conditions will also affect the time increment.

Thus, if the time increment is to small, very long computational time will be the result. This can be controlled by changing some model parameters. First of all a coarse mesh size reduces the analysis time. So does an increase of density or decrease of reference sound speed or even better a combination of the two since it turns out through parameter variation that there are combined effects. Also a decrease of the viscosity will in some cases reduce the analysis time.

It should be noted that altering these properties will affect the physical results of the analysis. In some cases the increment will be very small due to small models or high viscosity. In those cases the physical results may have to be compromised to be able to run the analysis within a reasonable amount of time.

The viscosity impact on the analysis time when running a small model can be seen in Figure 6.1. Note that this only applies to a certain model and may change depending on, for instance, the size of the model.

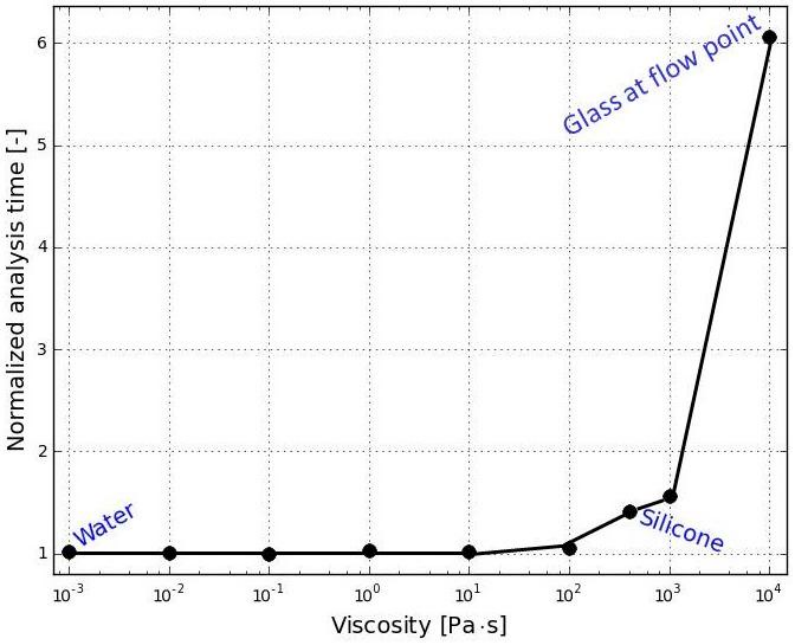


Figure 6.1: Analysis time at different viscosities for a small model. Viscosities below 100 Pa s hardly affects the analysis time. Above 100 Pa s a slight change in viscosity radically changes the analysis time.

6.2. Eulerian Region

The fluid region is defined by creating an Eulerian part in the part module of Abaqus. When defining the geometry of the part one should keep in mind that the region is fixed in space and does not deform as a regular Lagrangian mesh. Thus the entire volume that

may contain material at some point during the analysis and that is of interest should be modeled. To ensure contact with Lagrangian surfaces the Lagrangian surface should be within the Eulerian region with an overlap of at least one element.

6.3. Meshing

In Abaqus there is only one available element for Eulerian analysis, the EC3D8R element. Viscous hourglass control should be turned off and Linear and Quadratic bulk viscosity should be set to zero unless damping is to be introduced. The general guidelines from Simulia is that the Eulerian mesh size should be at least 3 times smaller then the smallest feature of interest in the Lagrangian mesh.[18]

6.4. Initial Fluid Location

To define the initial position of the Fluid in the Eulerian part it is a good idea to create a partition of that area and to make a set of this partition. The assignment of the material to the set is done by creating a predefined field in the Load module according to Figure 6.2. To completely fill the region of the model with material, choose 1 for material (second column) and 0 for void (third column).

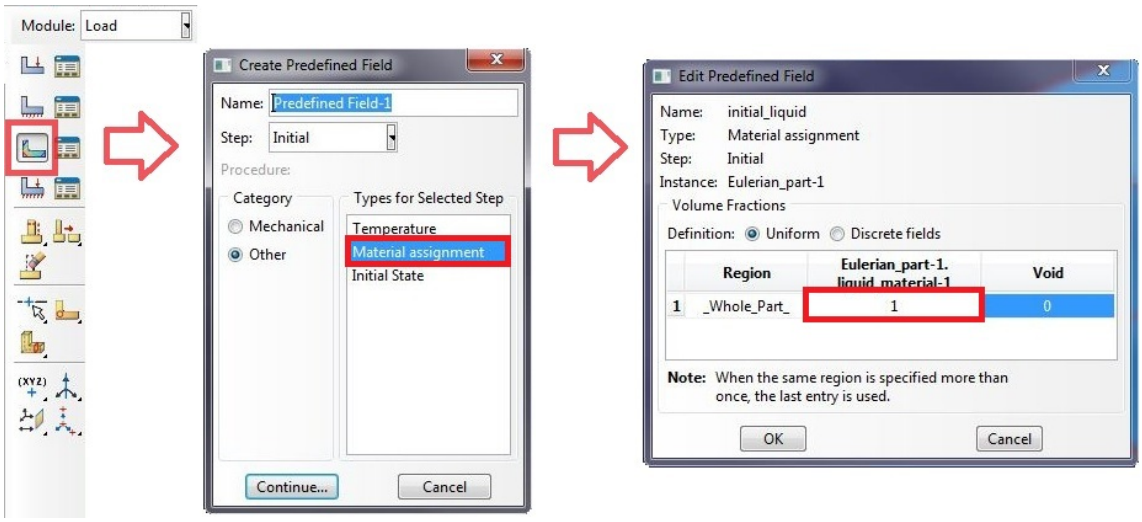


Figure 6.2: Specifying the initial location of the fluid by creating a predefined field.

If the geometry is complex it can be difficult to create sets matching the initial position of the fluid, then one can make use of the Volume Fraction Tool. The Volume Fraction Tool requires that the Eulerian part is already meshed. This is because it assigns material

to each element in the mesh. To specify the initial fluid location, specify a new part (an Eulerian or Lagrangian part) with the same shape and size as the initial location of the fluid. This will be used as a reference part instance. Assemble and constraint it to the correct position. In the load module choose Tools > Discrete Field > Volume Fraction Tool. Choose the Eulerian part instance and then the reference part instance. Fill in the volume fraction dialog shown in Figure 6.3. This will create a discrete field.

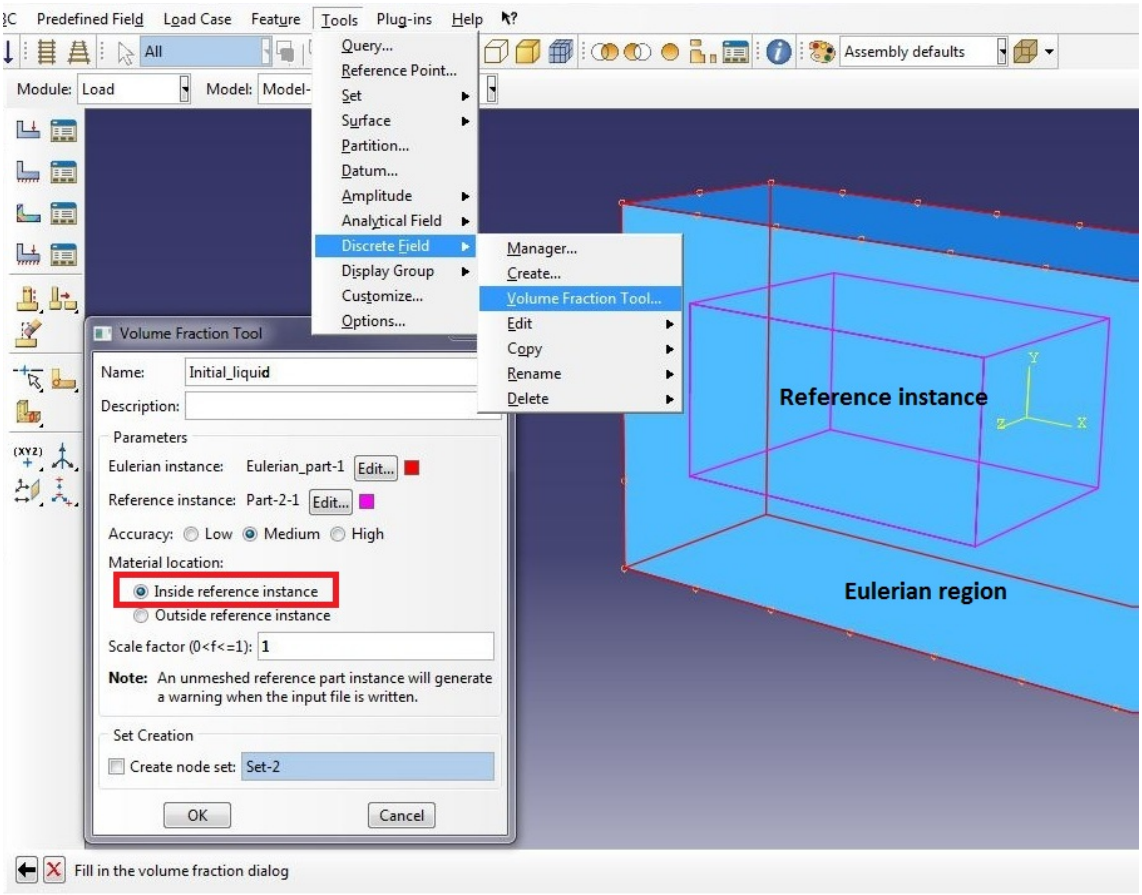


Figure 6.3: The volume fraction dialog assigns initial fluid element wise.

To assign the material to this discrete field follow the procedure in Figure 6.3 but choose discrete field in the last step.

After this is done the reference part instance must be deactivated by right clicking the instance in the module database tree and choosing suppress. If the Eulerian part is remeshed, the volume fraction tool will have to be used again.

6.5. Fluid Material

Two major types of fluid material can be used in CEL analysis, fluids and gases. This report is about polymer flow modeled as fluid material and thus only fluid material will be considered.

Fluid material can be defined by the material parameters density, an equation of state and the viscosity of the fluid.

The equation of state used for fluids is the Hugoniot formulation $U_s - U_p$ option where C_0 is the reference sound speed. Water for instance has a reference sound speed of about 1500m/s. s and gamma0 is set to zero, which is recommendations from Abaqus.

Only Newtonian (i.e not shear rate dependent) and non temperature dependent viscosity can be defined in the Abaqus CAE interface. For shear rate- and temperature dependent viscosity one can make use of the subroutine VUVISCOSITY.

6.6. Viscosity User Subroutine VUVISCOSITY

The user may write subroutines in Fortran to define certain characteristics in Abaqus. By using subroutines, one can implement arbitrary material models or save data in structured ways.

The subroutine VUVISCOSITY in Abaqus enables user coding of material models for the viscosity. The subroutine provides parameters such as time, shear rate and temperature which is updated through the analysis. These parameters can be used in the material model to describe the viscosity as a function of time, shear rate or temperature or both.

Temperature degrees of freedom are not yet implemented in CEL analysis which means that the temperature variable in VUVISCOSITY can not be used until it is included in future releases of Abaqus.

An example code where the Cross function as described in chapter 3 is implemented in VUVISCOSITY is shown in Appendix C.1.

When using a subroutine, the Fortran file is submitted along with the input file. In the CAE this is done by just appending the Fortran file when creating a job. When using input files to submit jobs on clusters the keyword

```
* viscosity, def = user
```

must be added to the input file.

6.7. Fluid Structure Interaction

To enforce contact between the Eulerian and Lagrangian region a general contact property is assigned in the interaction module according to Figure 6.4. This contact is a penalty contact, which apply forces preventing parts from entering each other.

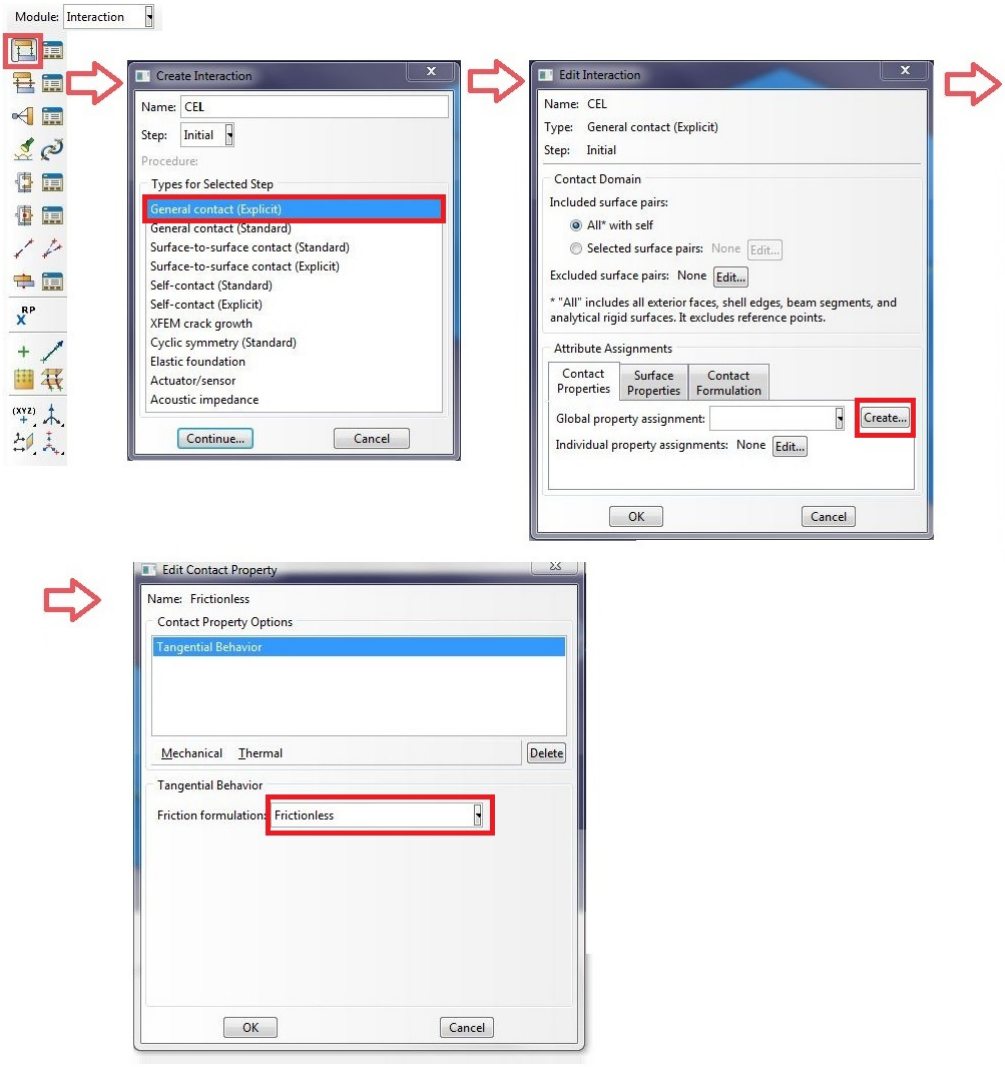


Figure 6.4: Enforcing frictionless contact between the Eulerian and Lagrangian region.

Penetration of the fluid media through a contact surface can occur if the model includes sharp edges or in case of a coarse mesh. This is because a volume fraction below 0.5 suppresses the contact between the Eulerian and Lagrangian element as shown in Figure 6.5.

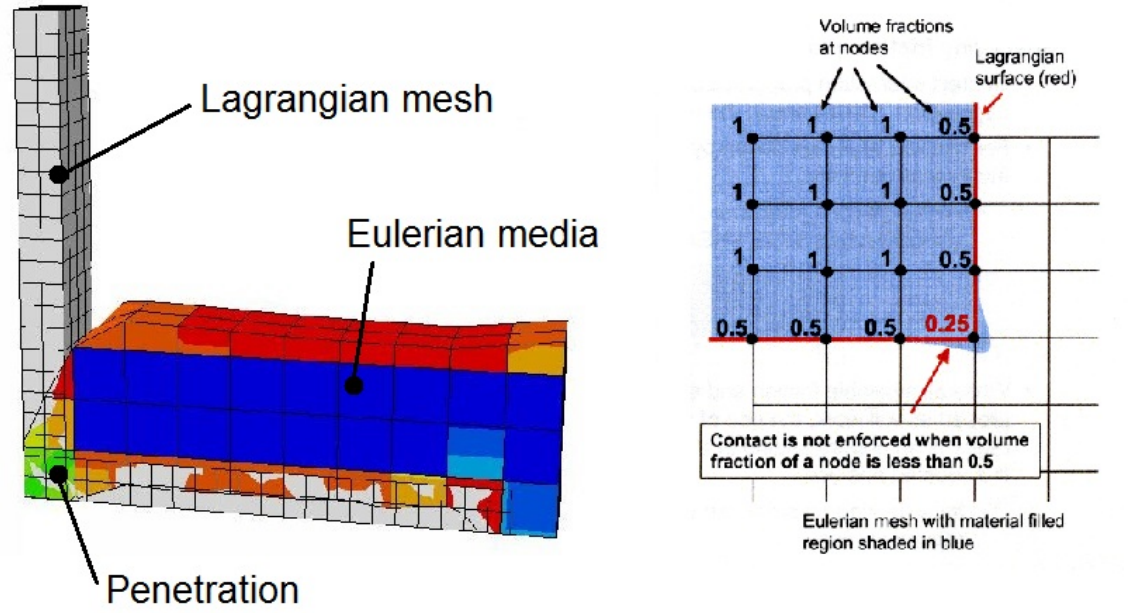


Figure 6.5: Left Figure shows penetration in a simple fluid structure interaction model. Right Figure shows that a volume fraction below 0.5 suppresses contact. [15]

6.8. Boundary Conditions and Loads

The standard boundary condition is free flow which allows the material to flow across the boundary freely. It also means that if a material is specified in contact with a boundary, the boundary will have free inflow of that material.

To prevent in or outflow over a boundary, the velocity can be prescribed to zero. Prescribed in (or outflow) can be defined in the same manner.

6.9. Submitting an Analysis Job

There are two ways of submitting a job. Either on a cluster or on the local computer. CEL analysis is very time consuming, thus it is highly recommended to use clusters.

As discussed earlier CEL analysis calculates a lot of time increments and because of this it is recommended to use double precision for more accurate results.

6.10. Visualization

To visualize the fluid, choose to plot deformed contours: EVF_VOID. Then open the View Cut Manager and check the box EVF_VOID and set the value to 0.5. This will hide all elements not containing at least 50% fluid. In the result options set Averaging threshold to 100% and open the common plot options > other > translucency and apply 70% translucency. This will result in a clear visualization.

7. Numerical Transversal Sealing Simulations in Abaqus

The intention of the work reported in this chapter is to show how to create a model of the transversal sealing process in Abaqus using the Coupled Eulerian-Lagrangian formulation, and to evaluate whether CEL is well suited for simulation of the TS process. Unfortunately there have been many obstacles in creating a realistic model of the sealing process which is why the first part of this chapter will discuss the limitations of analyzing the TS process using CEL and Abaqus/Explicit in general.

7.1. Limitations

Much of what is enlightened here has also been raised in chapters 5 and 6 but is here discussed as application specific.

The polymer layers that are studied has a thickness of about 50 μm . The layers are compressed and due to this the thickness will decrease with time. In the real application the polymer layer is often compressed until the thickness of the polymer layer is close to zero. This means that using a Lagrangian mesh in an explicit solution is out of the question since the element length and thus the time increment would also tend to zero.

Using a Eulerian mesh solves the problem of decreasing element lengths since it is fixed in space through the analysis. Yet, the element length will be very small, considering that there must be at least two elements, preferably three or four in the thickness direction of the polymer layers to capture the behavior.

Problems with small time increments can usually be solved by introducing higher mass in the system which increases the stable time increment cf. Chapter 6.1. The requirement for such an assumption to not have large physical effects is that the material is not rate dependent, i.e. that the material model does not depend on velocities. Unfortunately all viscous media does. This means that introducing higher density when simulating fluids using CEL will result in exaggerated forces.

During the TS process the polymer undergoes phase transitions. First the polymer is in solid form and as the temperature reaches the melting point of the polymer it will change phase to fluid media. This process is then reversed as the sealing cools off.

To capture this in the computer analysis, the media would first have to be elastic and then transform to viscous with temperature. Since temperature degrees of freedom is not yet available in CEL analysis this is not possible.

One must also be careful using high viscosities because of the impact on the computational cost discussed in chapter 6.

Even though there are several obstacles in creating a realistic model of the TS process, a model was created using a constant viscosity of 100 Pa s, mass scaling and a rough mesh. All which has been declared as unphysical in the above text.

7.2. Abaqus/CEL TS Model

The FSI model of the transversal sealing process was created using Abaqus/CAE 6.10-2 with the implemented CEL formulation, cf Chapter 5.2 and 6.

Initially, the model was intended to be created solely as a Fluid Structure Interaction model using CEL. The polymer however can not be considered as a fluid during the entire time period since polymers at low temperatures are elastic.

Therefore the sealing process will be modeled in two steps as indicated in Figure 7.1. In the first part the polymer will be modeled as a solid (Lagrangian part) with elastic behavior, after which the deformed geometry from the solid phase simulation will be used as input to the fluid phase simulation. The polymer will be switched to an Eulerian part and the analysis will be carried out using the CEL formulation.

This is a rough representation of the phase transition from solid to liquid behavior of the polymer, and a major error may be introduced. This error is due to that the the deformed geometry obtained from the solid simulation is stress free, but in reality this is not the case. The latter part of the process, containing the unloading phase will not be considered since there has been found no way to transfer the Eulerian material positions back to a Lagrangian geometry in Abaqus.

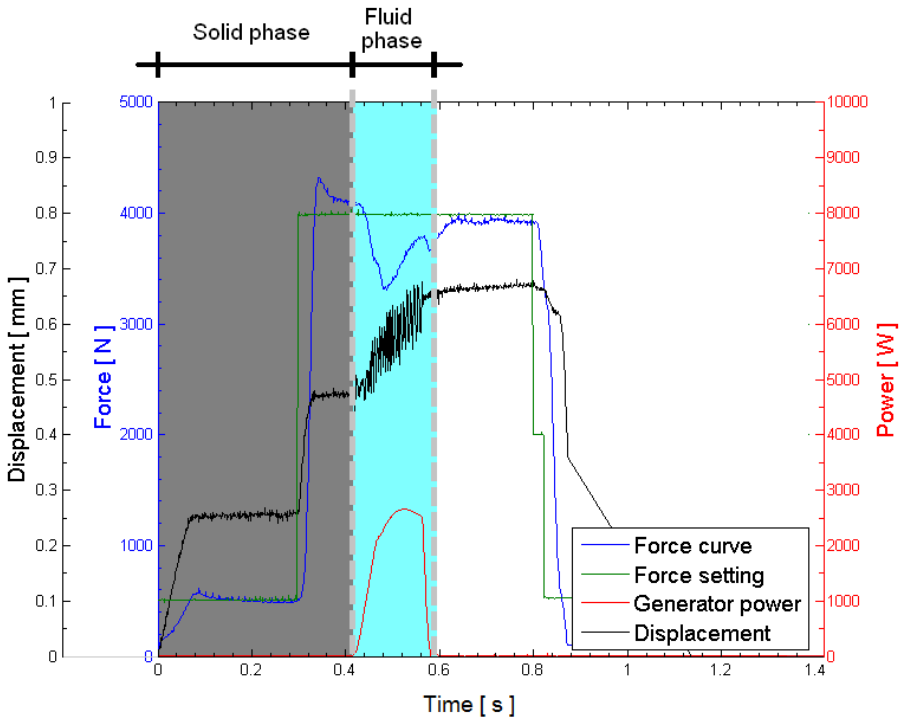


Figure 7.1: Analysis divided into two parts. Solid phase(gray) and Fluid phase(turquoise).

7.2.1. Geometry

The only available element for Eulerian analysis is the EC3D8R element which is an eight node, linear, 3D element with reduced integration which means that the model must be three dimensional.

To decrease computational cost only a small strip of the seal geometry was considered and due to symmetry, shown in Figure 7.2, only one half of the model was modelled.

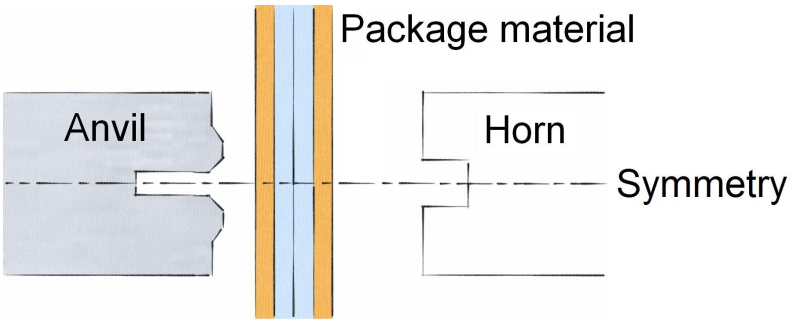


Figure 7.2: Symmetry of the model.

The geometry of the three anvils were modeled as shown in Figure 7.3. Only the bottom face of the anvil was expected to be in contact with other parts and thus to further decrease the computational cost, the anvils were modeled as discrete rigid bodies since negligible deformations of the anvils was expected.

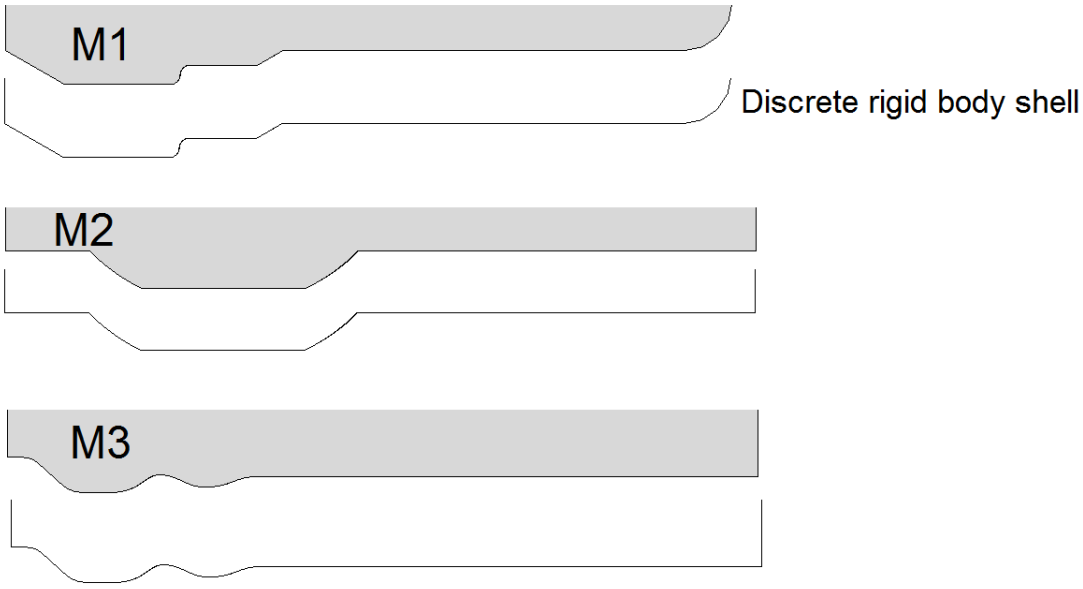


Figure 7.3: The three anvils are modeled as discrete rigid body shells.

The packaging material which consists of several layers, cf. chapter 2, was for simplicity modeled as two layers, namely the paper and the polymer as shown in figure 7.4.

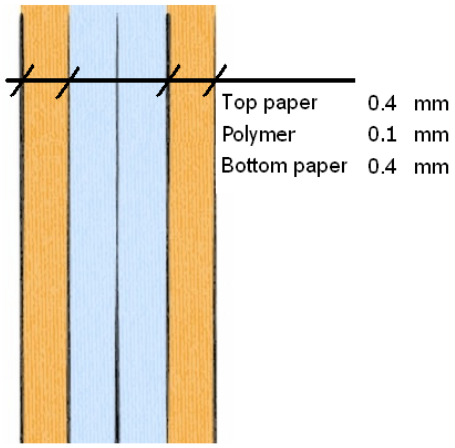


Figure 7.4: Simplified model of two opposite sheets of packaging material.

7.2.2. Simulation of the Solid Phase of the Polymer

The solid phase simulation is modeled according to Figure 7.5 where both the paper and polymer are modelled as deformable (Lagrangian) parts. The constitutive behavior of the paper is modeled using a simplified material model, which captures the compressional behavior of the paper. The material of the polymer is modeled as a linear isotropic material with Young's modulus 200 MPa and the Poisson's ratio has been set to zero for both materials

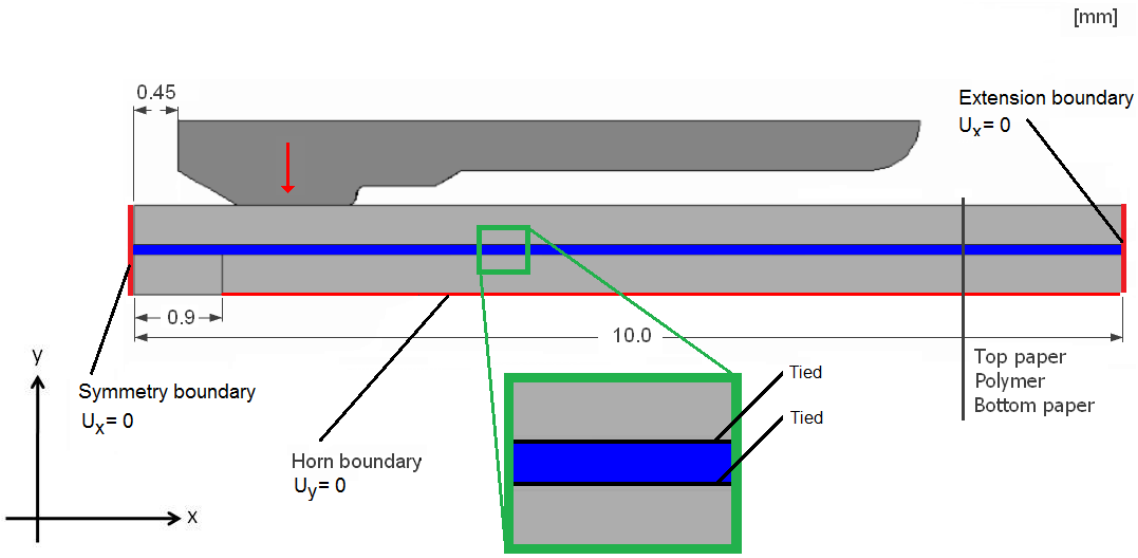


Figure 7.5: FE-model of the solid phase of the polymer using anvil M1.

The symmetry boundary is modeled as a restricted motion in the x-direction. The horn is not modelled, instead a prescribed boundary condition is applied representing the horn. The model has a boundary condition to the right which also is a restricted displacement in the x-direction. The whole model is also restricted in the z-direction as the model represents a small slice of what is seen as a infinite long sealing. The displacement of the anvil is controlled by a reference point which is restricted in the z and x direction as well as rotations and assigned a motion in the negative y direction according to Figure 7.6. The curve controlling this displacement is adopted from the measurements but has been smoothed due to numerical stability issues.

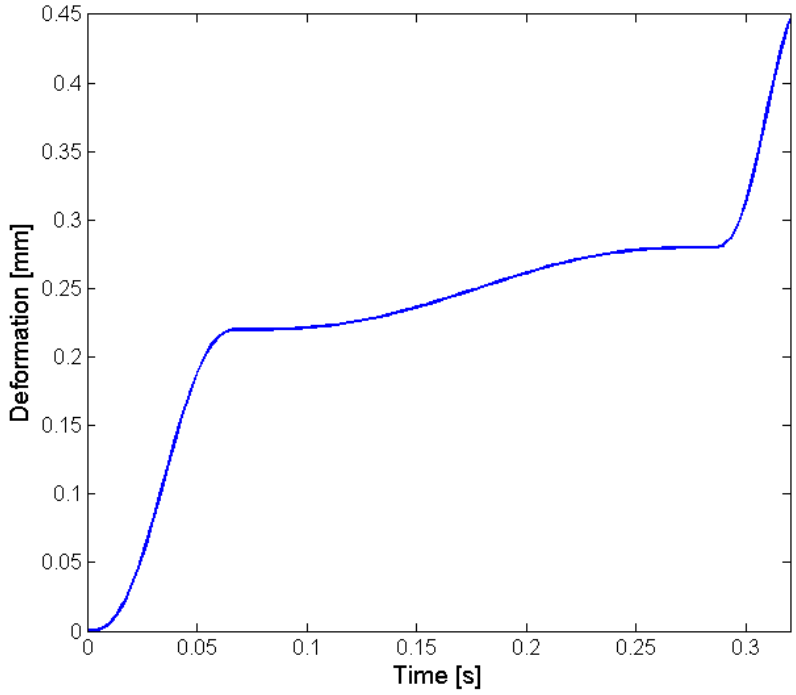


Figure 7.6: Simplification of deformation curve from experimental testing.

The analysis was carried out at a loading rate that was increased by a factor of 100, preventing long analysis time. Frictionless general contact was prescribed between the anvil and the top paper. General contact uses penalties to enforce contact. This means that it applies a force which depends on the overclosure of the nodes. This means that overclosure will always occur. The default penalty was not strong enough and thus a scaling was required. A scaling of 9 times the default penalty seemed to be enough and were used. The effects of using scaled penalties has not been studied in this work.

7.2.3. Simulation of the Fluid Phase of the Polymer

The geometry of the fluid phase simulation was extracted from the results of the Solid phase simulation. This was done by importing the deformed mesh from the odb file. In Abaqus, an imported mesh is called an orphan mesh and has few changeable attributes. By transforming the orphan mesh into a part using the Abaqus commands PartFromSection3DMeshByPlane and Part2DGeomFrom2DMesh the orphan mesh is first transformed into a 2D orphan mesh and then into a 2D shell part. By using the sketch from this part a 3D deformable part can be created. The script file used for this can be found in Appendix B.1.

The fluid polymer was modeled as a material with a constant viscosity of 100 Pa s partly because temperature degrees of freedom is not yet implemented in CEL analysis and partly because high viscosity models are very time consuming. The Cross-Arrhenius implementation discussed in chapter 4 was not used but may be used in the next version of Abaqus where temperature degrees of freedoms are introduced.

The boundary conditions for the fluid model was the same as in the solid phase simulation except that there was no tie between the paper and polymer since it was considered as a fluid. Velocity boundary conditions in the x and z-direction have been set as symmetry conditions for the Eulerian region in accordance with chapter 6.

To reduce the analysis time, the actual process time was decreased by a factor of 10000. The analysis time was 5.5 minutes. The full scale model would occupy 8 cpus for 38 days despite all assumptions made. The contact penalty was scaled in the same way as in the solid model.

In the analysis, the anvil was deformation controlled and a smoothed curve of the measured deformations from the experiments was used as input, as shown in Figure 7.7.

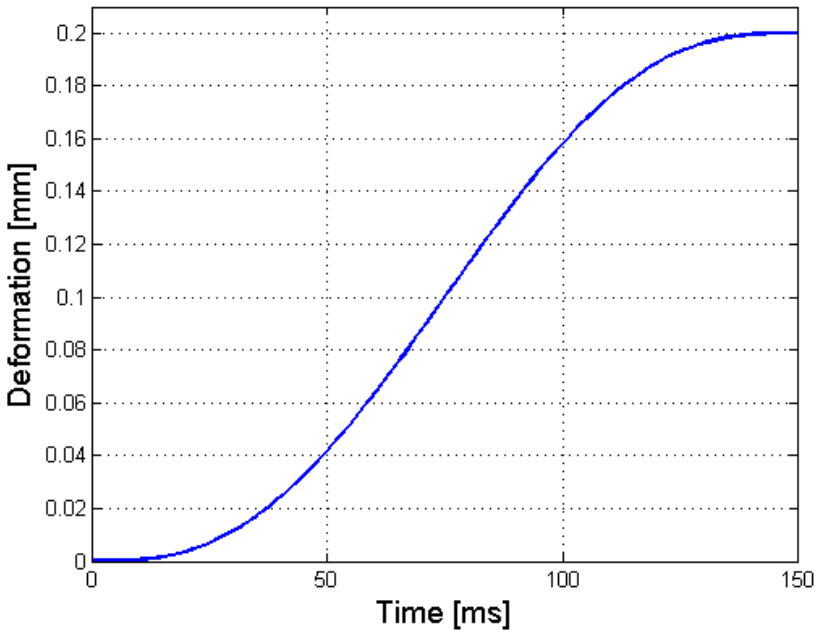


Figure 7.7: Simplification of deformation curve from experimental testing.

The results from both the solid and the fluid analysis is shown in Figure 7.8.

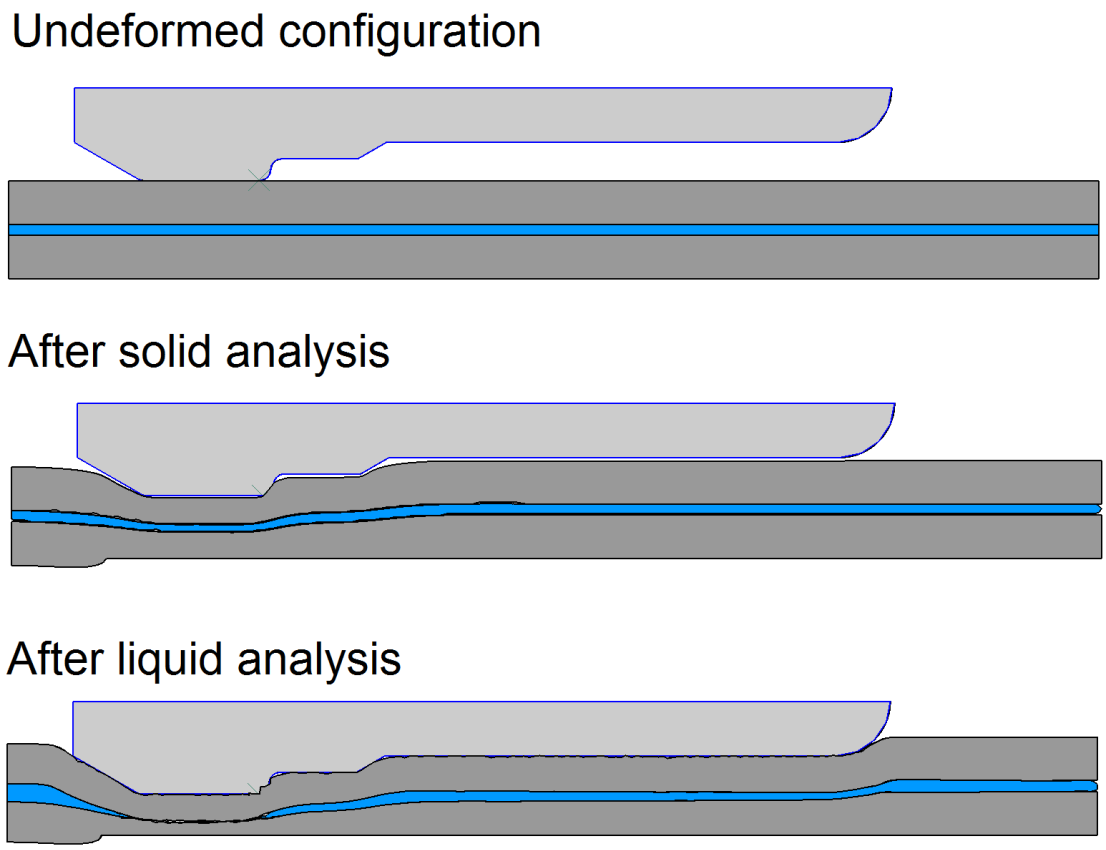


Figure 7.8: Undeformed geometry and results from the solid and the fluid analysis.

8. CONCLUSIONS

Throughout this work, the main objective was to investigate whether the Coupled Eulerian-Lagrangian formulation implemented in Abaqus is well suited for simulating the full sealing process where the packaging material structure is compressed and heated.

In the journey towards this objective several important findings have been identified, especially concerning material properties of polymer materials during the fluid phase.

It has been recognized that the viscosity is the primary material parameter to define the resistance of motion for the polymer material. A general method of fitting experimental data of shear rate and temperature dependent viscosity to a surface function using the Cross and Arrhenius functions has been developed and the implementation in Abaqus by means of subroutines has also been discussed in the thesis.

Experimental work was made to achieve a better understanding of the sealing process, using ultra sonic vibrations, and also to obtain referential results for the analyses and for future work.

The experimental data shows that not only the polymer layer weakens due to heating but also the paperboard layers. It also shows that the deformed geometry of the packaging material, both polymer and paperboard is a direct consequence of the choice of anvil.

Much of the work in the thesis has been devoted to understanding how and if the CEL formulation is suitable for transversal sealing process simulations.

The CEL formulation is a tool for analyzing fluid structure interaction by allowing contact conditions between Lagrangian and Eulerian regions. In other words, fluids can interact with solids which was why the method was chosen by Tetra Pak for this thesis.

During the thesis several obstacles, which made it impossible to make a physically accurate analysis, were encountered.

The main problem was the time increment which tends to zero for very small geometries such as found in the sealing process. The problem is due to using an explicit solver. Explicit solvers needs small time increments for the solution to be numerically stable. When using CEL, there has been found no way around the problem other then scaling the mass or size of the model, but this changes the results.

There has neither been found a way to simulate the phase transition without changing the physics of the process. At the very end of this work an idea of adding an elastic shear modulus came up. Possibly, it is a way of solving the phase transition problem. There was however no time to investigate that possibility within the time frame of the thesis.

The general conclusion of this part of the thesis is that the limitations of the CEL formulation for small scale geometry, high viscosities or phase transitions is so critical that at this time there has not been found a good way to simulate the full sealing process without compromising the physics.

9. SUGGESTIONS FOR FURTHER WORK

Several findings were done during the FE-model development in this thesis. The material model for the polymer needs further work. Ways of capturing the solid phase and the transition between solid and fluid phase needs to be investigated. Further more experimental verification of the fluid phase material model is needed.

To be able to create a realistic model of the sealing process using CEL the analysis must be carried out without scaling the time which would take roughly 38 days at 8cpu's. The problem of small time increments must be solved otherwise, perhaps by scaling quantities with knowledge of how the physics is changing and compensate for this in the results.

Furthermore one should be able to implement the material model as soon as temperature degrees of freedom is introduced in Abaqus 6.11.

A material model for the thermal coupling should also be considered as the high temperature of the sealing process surely affects both the polymer and the paper.

Probably the best way of gaining knowledge of the sealing process and creating a realistic model is to take a step back and examine each subprocess of the sealing process separately.

As discussed earlier the material models needs to be further developed and verified. Also knowledge of how the heat is generated and transported in both the paperboard and the polymer is needed.

When all subprocesses are known, the work of putting these together can be initiated. How the coupling should be done and by means of which software is also an issue of interest.

BIBLIOGRAPHY

[1] Tetra Pak homepage, *Tetra pak in figures (Sept 2010)*. Available from: http://www.tetrapak.com

[2] Rheology school, *Making Use Of Models (Oct 2010)*. Available from: http://www.rheologyschool.com/

[3] *Internal Tetra Pak report*.

[4] Belytschko T, *Non Linear Continua Equations*. 1998

[5] Johansson Consulting *Six Sigma Black Belt Training*.

[6] Holzapfel G.A, *Nonlinear solid mechanics - A continuum approach for engineering*. 1998

[7] White F.M, *Fluid Mechanics - Sixth Edition*. 2008

[8] Price J.F, *Lagrangian and Eulerian Representations of Fluid Flow: Kinematics and the Equations of motion*. 2006

[9] Ottosen N.S, Ristinmaa M, *The Mechanics of Constitutive Modeling*. 2005

[10] Krenk S, *Non-Linear Modeling and Analysis of Solids and Structures*. 2009

[11] Min Soo Kim, Woo Il Lee, *A new VOF-based numerical scheme for the simulation of fluid flow with free surface. Part I: New free surface-tracking algorithm and its verification*. 2003

[12] Helleloid G.T, *Morehead Electronic Journal of Applicable Mathematics, Issue 1 - CHEM-2000-01 On the Computation of Viscosity-Shear Rate Temperature Master Curves for Polymeric Liquids.* University of Wisconsin - Madison

[13] Polymer technology, *Rheology of fluids (Sept 2010)*. Available from: http://polymer.w99of.com

[14] Dassault Systemes Simulia Corp, *Abaqus Manual, Version 6.10*. 2010

[15] Dassault Systemes Simulia Corp, *Abaqus/Explicit: Advanced Topics & Coupled Eulerian-Lagrangian Analysis*. Presented at Tetra Pak, Lund, October 6-9 2008

[16] Dassault Systemes Simulia Corp, *Answer ID 3765 available from Abaqus support*.

[17] Dassault Systemes Simulia Corp, *Answer ID 4466 available from Abaqus support*. 2010

[18] Dassault Systemes Simulia Corp, *Answer ID 3523 available from Abaqus support*. 2010

[19] Sharat Prasad, Engineering Specialist, Dassault Systemes. 13 October, 2010

[20] Pacific Northwest National Laboratory, Available from: http://www.technet.pnl.gov

A. Abaqus Input Files

A.1. Solid Model Input File

```
*Heading
** Job name: TS_solid_P1_M1 Model name: Model-1
** Generated by: Abaqus/CAE 6.10-2
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Anvil
*End Part
**
*Part, name=Anvil-disp
*End Part
**
*Part, name="Bottom paper"
*End Part
**
*Part, name="Eulerian region"
*End Part
**
*Part, name="Eulerian region-2"
*End Part
**
*Part, name="Top paper"
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name="Bottom paper-1", part="Bottom paper"
*Node
      1, 0.000899999985,           0.,           0.
...
3618, 0.000950000016,  4.99999987e-05,           0.
*Element, type=C3D8R
  1,   94,  102,  966,  847,    1,    2,   13,   46
...
1600,  102,    2,    1,   94, 2351,  846,  485, 3618
*Nset, nset=paperboundary
   1,   2,  10,  11, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496
 ...
 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846
*Elset, elset=paperboundary, generate
  152,  1600,     8
** Section: Hyperfoam
*Solid Section, elset=_paper_part, material=Hyperfoam
,
```

```
*End Instance
**
*Instance, name="Eulerian region-1", part="Eulerian region"
 -2.76471553983809e-20, 0.000150000000000003,           0.
*Node
      1, 0.00999999978, 0.000349999988,  2.49999994e-05
...
   4008,           0., 0.000300000014,           0.
*Element, type=C3D8R
   1,    9,   10,   14,   13,    1,    2,    6,    5
...
1500, 4003, 4004, 4008, 4007, 3995, 3996, 4000, 3999
** Section: Plastic
*Solid Section, elset=_polymer_part, material=Plastic
,
*End Instance
**
*Instance, name="Top paper-1", part="Top paper"
          0., 0.000499999999999999,           0.
*Node
      1, 0.00999999978, 0.00039999999,  2.49999994e-05
...
   3618,           0.,           0.,           0.
*Element, type=C3D8R
   1,   19,   20,   29,   28,    1,    2,   11,   10
...
1600, 3608, 3609, 3618, 3617, 3590, 3591, 3600, 3599
*End Instance
**
*Instance, name="Eulerian region-2-1", part="Eulerian region-2"
          0., 0.000100000000000001,           0.
*Node
      1, 0.00999999978, 0.000349999988,  2.49999994e-05
...
  24024,           0., 0.000300000014,           0.
*Element, type=C3D8R
    1,    25,    26,    32,    31,     1,     2,     8,     7
...
15000, 24017, 24018, 24024, 24023, 23993, 23994, 24000, 23999
** Section: Plastic
*Solid Section, elset=_polymer_part_2, material=Plastic
,
*End Instance
**
*Instance, name=Anvil-1, part=Anvil
0.00044764297739604, 0.00279741988391855,           0.
*Node
      1, 0.00180235703, -0.00179741986,  2.49999994e-05
...
    288, 0.00185235706, -0.00171081733,           0.
*Element, type=R3D4
1,  1, 23, 26,  4
...
143, 286,   1,   4, 287
*Node
    289, 0.00170235697, -0.00189741992,  2.49999994e-05
*Nset, nset=Anvil-1-RefPt_, internal
289,
*Elset, elset=Anvil-1, generate
   1,  143,    1
*End Instance
**
*Instance, name=Anvil-disp-1, part=Anvil-disp
0.00044764297739604, 0.00279741988391855,           0.
*Node
      1, 0.00179474498, -0.00183568825,  2.49999994e-05
...
```

```
    110, 0.00260235695, -0.00169741991,           0.
*Element, type=S3
 1,  10,   4,   9
...
84, 108, 106, 107
*End Instance
**
*Nset, nset=_PickedSet48, internal, instance="Bottom paper-1"
   3,   4,   6,   7,   9,  10,  11,  12,  81,  82,  83,  84,  85,  86,  87,  95
...
 119, 120, 121, 122
*Nset, nset=_PickedSet48, internal, instance="Top paper-1"
    1,    2,    3,    4,    5,    6,    7,    8,    9,   10,   11,   12,   13,   14,   15,   16
...
 3615, 3616, 3617, 3618
*Elset, elset=_PickedSet48, internal, instance="Bottom paper-1"
  18,  36,  54,  72,  90, 108, 126, 144, 145, 146, 147, 148, 149, 150, 151, 152
*Elset, elset=_PickedSet48, internal, instance="Top paper-1"
    1,    2,    3,    4,    5,    6,    7,    8, 1593, 1594, 1595, 1596, 1597, 1598, 1599, 1600
*Nset, nset=_PickedSet98, internal, instance="Eulerian region-1"
    1,    2,    3,    4,    5,    6,    7,    8, 4001, 4002, 4003, 4004, 4005, 4006, 4007, 4008
*Nset, nset=_PickedSet98, internal, instance="Eulerian region-2-1"
     1,     2,     3,     4,     5,     6,     7,     8,     9,    10,    11,    12,    13,    14,    15,    16
...
 24009, 24010, 24011, 24012, 24013, 24014, 24015, 24016, 24017, 24018, 24019, 24020, 24021, 24022, 24023, 24024
*Elset, elset=_PickedSet98, internal, instance="Eulerian region-1"
    1,    2,    3, 1498, 1499, 1500
*Elset, elset=_PickedSet98, internal, instance="Eulerian region-2-1"
     1,     2,     3,     4,     5,     6,     7,     8,     9,    10,    11,    12,    13,    14,    15, 14986
 14987, 14988, 14989, 14990, 14991, 14992, 14993, 14994, 14995, 14996, 14997, 14998, 14999, 15000
*Nset, nset=_PickedSet99, internal, instance="Eulerian region-1"
    1,    2,    3,    4,    6,    7,    8,    9,   10,   11,   12,   14,   15,   16,   17,   18
...
 4006, 4007, 4008
*Nset, nset=_PickedSet99, internal, instance="Eulerian region-2-1"
     1,     2,     3,     4,     5,     6,    19,    20,    21,    22,    23,    24,    25,    26,    27,    28
...
 24001, 24002, 24003, 24004, 24005, 24006, 24019, 24020, 24021, 24022, 24023, 24024
*Elset, elset=_PickedSet99, internal, instance="Eulerian region-1", generate
    1,  1500,     1
*Elset, elset=_PickedSet99, internal, instance="Eulerian region-2-1"
     1,     2,     3,     4,     5,    11,    12,    13,    14,    15,    16,    17,    18,    19,    20,    26
...
 14975, 14981, 14982, 14983, 14984, 14985, 14986, 14987, 14988, 14989, 14990, 14996, 14997, 14998, 14999, 15000
*Nset, nset=_PickedSet122, internal, instance="Bottom paper-1"
   1,   2,  10,  11, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496
...
 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846
*Elset, elset=_PickedSet122, internal, instance="Bottom paper-1", generate
  152,  1600,     8
*Nset, nset=_PickedSet123, internal, instance="Bottom paper-1", generate
    1,  3618,     1
*Nset, nset=_PickedSet123, internal, instance="Top paper-1", generate
    1,  3618,     1
*Elset, elset=_PickedSet123, internal, instance="Bottom paper-1", generate
    1,  1600,     1
*Elset, elset=_PickedSet123, internal, instance="Top paper-1", generate
    1,  1600,     1
*Nset, nset=_PickedSet132, internal, instance=Anvil-1
 289,
*Elset, elset=__PickedSurf108_S4, internal, instance="Eulerian region-2-1", generate
     5,  15000,      5
*Surface, type=ELEMENT, name=_bottompolymer_tie, internal
__PickedSurf108_S4, S4
*Elset, elset=__PickedSurf109_S4, internal, instance="Top paper-1", generate
    8,  1600,     8
*Surface, type=ELEMENT, name=_toppaper_tie, internal
```

```
__PickedSurf109_S4, S4
*Elset, elset=__PickedSurf110_S6, internal, instance="Eulerian region-1", generate
    1,  1498,     3
*Surface, type=ELEMENT, name=_toppolymer_tie, internal
__PickedSurf110_S6, S6
*Elset, elset=__PickedSurf140_S6, internal, instance="Bottom paper-1", generate
  145,  1593,     8
*Elset, elset=__PickedSurf140_S1, internal, instance="Bottom paper-1", generate
 127,  144,    1
*Surface, type=ELEMENT, name=_bottompaper_tie, internal
__PickedSurf140_S6, S6
__PickedSurf140_S1, S1
*Elset, elset=_Surf-1_S4, internal, instance="Eulerian region-1", generate
    3,  1500,     3
*Surface, type=ELEMENT, name=Surf-1
_Surf-1_S4, S4
*Elset, elset=_Surf-2_S6, internal, instance="Eulerian region-2-1", generate
     1,  14996,      5
*Surface, type=ELEMENT, name=Surf-2
_Surf-2_S6, S6
*Rigid Body, ref node=Anvil-1.Anvil-1-RefPt_, elset=Anvil-1.Anvil-1
** Constraint: Bottom_tie
*Tie, name=Bottom_tie, adjust=yes
_bottompolymer_tie, _bottompaper_tie
** Constraint: Top_Tie
*Tie, name=Top_Tie, adjust=yes
_toppolymer_tie, _toppaper_tie
** Constraint: disp_body
*Display Body, instance=Anvil-disp-1
Anvil-1.289,
*End Assembly
*Amplitude, name=Amp-2, definition=SMOOTH STEP
          0.,           0.,       0.0007,      0.00022,      0.00285,      0.00028,      0.00333,      0.00047
**
** MATERIALS
**
*Material, name=Hyperfoam
*Density
100000.,
** (Massscaled)
*Hyperfoam, testdata
*Uniaxial Test Data
     4.48e+07,   0.6,    0.
   -5.143e+06,  -0.1,    0.
  -1.1174e+07, -0.15,    0.
  -2.3047e+07, -0.25,    0.
   -3.918e+07,  -0.3,    0.
  -5.8665e+07, -0.35,    0.
  -7.8569e+07, -0.38,    0.
 -1.06656e+08, -0.42,    0.
 -2.17907e+08,  -0.5,    0.
 -4.54272e+08,  -0.6,    0.
*Material, name="Linear elastic"
*Density
100000.,
** (Massscaled)
*Elastic
 3e+08,0.
*Material, name=Plastic
*Density
100000.,
** (Massscaled)
*Elastic
 2e+08,0.
**
** INTERACTION PROPERTIES
**
```

```
*Surface Interaction, name=general
**
** BOUNDARY CONDITIONS
**
** Name: Euelrian X-led Type: Displacement/Rotation
*Boundary
_PickedSet98, 1, 1
** Name: Euelrian Z-led Type: Displacement/Rotation
*Boundary
_PickedSet99, 3, 3
** Name: Lagr_X-led Type: Displacement/Rotation
*Boundary
_PickedSet48, 1, 1
** Name: Lagr_Y-led Type: Displacement/Rotation
*Boundary
_PickedSet122, 2, 2
** Name: Lagr_Z-led Type: Displacement/Rotation
*Boundary
_PickedSet123, 3, 3
** ----------------------------------------------------------------
**
** STEP: dyn_expl
**
*Step, name=dyn_expl
*Dynamic, Explicit
, 0.00333
*Bulk Viscosity
0., 0.
**
** BOUNDARY CONDITIONS
**
** Name: Rigid_body_motion Type: Displacement/Rotation
*Boundary, amplitude=Amp-2
_PickedSet132, 1, 1
_PickedSet132, 2, 2, -1.
_PickedSet132, 3, 3
_PickedSet132, 4, 4
_PickedSet132, 5, 5
_PickedSet132, 6, 6
**
** INTERACTIONS
**
** Interaction: Int-1
*Contact, op=NEW
*Contact Inclusions, ALL EXTERIOR
*Contact Property Assignment
 ,  , general
*Contact controls assignment, type=scale penalty
surf-1,surf-2,9
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=1, time marks=NO
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT, number interval=100
**
** HISTORY OUTPUT: Energy
**
*Output, history
*Energy Output
ALLAE, ALLCD, ALLCW, ALLDC, ALLDMD, ALLFD, ALLIE, ALLKE, ALLMW, ALLPD, ALLPW, ALLSE, ALLVD, ALLWK, ETOTAL
*End Step
```

A.2. Liquid Model Input File

```
*Heading
** Job name: TS_visc_P1_M1 Model name: Model-1
** Generated by: Abaqus/CAE 6.10-2
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Anvil
*End Part
**
*Part, name=Anvil-disp
*End Part
**
*Part, name="Bottom Paper"
*End Part
**
*Part, name=Eulerian
*Surface, type=EULERIAN MATERIAL, name=polymer-1
polymer-1
*End Part
**
*Part, name="Top Paper"
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Anvil-1, part=Anvil
0.00044764297739604, 0.00248640988391855,           0.
*Node
      1, 0.00180235703, -0.00179741986,  2.49999994e-05
...
    856, 0.00188284798, -0.00169934134,           0.
*Element, type=R3D4
1,  1, 23, 36,  4
...
427, 849,   1,   4, 850
*Node
    857, 0.00170235697, -0.00189741992,  2.49999994e-05
*Nset, nset=Anvil-1-RefPt_, internal
857,
*Elset, elset=Anvil-1, generate
   1,  427,    1
*End Instance
**
*Instance, name=Anvil-disp-1, part=Anvil-disp
0.00044764297739604, 0.00248640988391855,           0.
*Node
      1, 0.00179474498, -0.00183568825,  2.49999994e-05
...
    110, 0.00260235695, -0.00169741991,           0.
*Element, type=S3
 1,  10,   4,   9
...
84, 108, 106, 107
*End Instance
**
*Instance, name="Bottom Paper-1", part="Bottom Paper"
*Node
      1, 0.00701939687,           0.,           0.
...
```

```
   3842,  0.006480732, 0.00016022168,  2.99999992e-05
*Element, type=C3D8R
   1,  922,  694,  244,  243, 2843, 2615, 2165, 2164
...
1713, 1886, 1449, 1920, 1921, 3807, 3370, 3841, 3842
** Section: Paper_sect
*Solid Section, elset=_PickedSet4, material="Linear elastic"
,
*End Instance
**
*Instance, name="Top Paper-1", part="Top Paper"
*Node
      1, 0.00875478052, 0.000619949307,           0.
...
   3766, 0.000673354371, 0.00057974353,  2.99999992e-05
*Element, type=C3D8R
   1,  692,    3,    1,  691, 2575, 1886, 1884, 2574
...
1674, 1312, 1882, 1640,  414, 3195, 3765, 3523, 2297
** Section: Paper_sect
*Solid Section, elset=_PickedSet4, material="Linear elastic"
,
*End Instance
**
*Instance, name=Eulerian, part=Eulerian
    0.005125,   0.00113899,           0.
*Node
      1, -0.00512500014, -0.00100000005,  2.99999992e-05
...
  14028, 0.0048750001, -0.00039999999,           0.
*Element, type=EC3D8R
   1,    43,    44,    65,    64,     1,     2,    23,    22
..
 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614
*Elset, elset=_PickedSet231, internal, instance="Bottom Paper-1"
   10,   11,   12,   13,   34,   36,   39,   40,   41,   42,   43,   44,   54,   55,   56,   57
...
 1061, 1074, 1082
*Nset, nset=_PickedSet232, internal, instance="Bottom Paper-1", generate
    1,  3842,     1
*Nset, nset=_PickedSet232, internal, instance="Top Paper-1", generate
    1,  3766,     1
*Elset, elset=_PickedSet232, internal, instance="Bottom Paper-1", generate
    1,  1713,     1
*Elset, elset=_PickedSet232, internal, instance="Top Paper-1", generate
    1,  1674,     1
*Nset, nset=_PickedSet238, internal, instance=Eulerian
     1,     2,     3,     4,     5,     6,     7,     8,     9,    10,    11,    12,    13,    14,    15,    16
...
 14025, 14026, 14027, 14028
*Elset, elset=_PickedSet238, internal, instance=Eulerian
    1,    2,    3,    4,    5,    6,    7,    8,    9,   10,   11,   12,   13,   14,   15,   16
   17,   18,   19,   20, 6641, 6642, 6643, 6644, 6645, 6646, 6647, 6648, 6649, 6650, 6651, 6652
 6653, 6654, 6655, 6656, 6657, 6658, 6659, 6660
*Nset, nset=_PickedSet239, internal, instance=Eulerian, generate
     1,  14028,      1
*Elset, elset=_PickedSet239, internal, instance=Eulerian, generate
    1,  6660,     1
*Nset, nset=_PickedSet240, internal, instance=Anvil-1
 857,
*Nset, nset=_PickedSet268, internal, instance=Eulerian, generate
     1,  14028,      1
*Elset, elset=_PickedSet268, internal, instance=Eulerian, generate
    1,  6660,     1
*Nset, nset=_PickedSet273, internal, instance=Eulerian, generate
     1,  14028,      1
*Elset, elset=_PickedSet273, internal, instance=Eulerian, generate
```

```
    1,  6660,     1
*Nset, nset=_PickedSet274, internal, instance=Eulerian, generate
     1,  14028,      1
*Elset, elset=_PickedSet274, internal, instance=Eulerian, generate
    1,  6660,     1
*Nset, nset=_PickedSet275, internal, instance=Eulerian, generate
     1,  14028,      1
*Elset, elset=_PickedSet275, internal, instance=Eulerian, generate
    1,  6660,     1
*Nset, nset=_PickedSet276, internal, instance=Eulerian
     1,    21,    22,    42,    43,    63,    64,    84,    85,   105,   106,   126,   127,   147,   148,   168
...
 13945, 13965, 13966, 13986, 13987, 14007, 14008, 14028
*Elset, elset=_PickedSet276, internal, instance=Eulerian
    1,   20,   21,   40,   41,   60,   61,   80,   81,  100,  101,  120,  121,  140,  141,  160
...
 6561, 6580, 6581, 6600, 6601, 6620, 6621, 6640, 6641, 6660
*Elset, elset=_Surf-1_SPOS, internal, instance=Anvil-1, generate
   1,  427,    1
*Surface, type=ELEMENT, name=Surf-1
_Surf-1_SPOS, SPOS
*Elset, elset=_Surf-2_S4, internal, instance="Top Paper-1"
    4,    5,    9,   11,   12,   15,   20,   30,   43,   44,   46,   47,   65,   68,   70,   71
...
  941,  962,  966,  995, 1033, 1367
*Elset, elset=_Surf-2_S6, internal, instance="Top Paper-1"
   7,   8,  10,  14,  17,  19,  22,  24,  26,  27,  28,  29,  31,  32,  33,  67
...
 403, 406, 409, 491, 493, 494, 495, 497, 565, 693, 796, 827, 876, 887
*Elset, elset=_Surf-2_S5, internal, instance="Top Paper-1"
   35,  116,  130,  216,  219,  221,  367,  392,  411,  415,  431,  443,  445,  450,  461,  463
  999, 1001, 1002, 1013, 1016, 1018, 1022, 1025, 1026
*Elset, elset=_Surf-2_S3, internal, instance="Top Paper-1"
  114,  132,  133,  134,  136,  211,  212,  213,  215,  217,  218,  220,  222,  405,  407,  410
...
  577,  751,  803,  804,  824,  908,  939, 1000, 1019, 1032, 1055
*Surface, type=ELEMENT, name=Surf-2
_Surf-2_S4, S4
_Surf-2_S6, S6
_Surf-2_S5, S5
_Surf-2_S3, S3
*Elset, elset=_Surf-3_S4, internal, instance="Top Paper-1"
    1,   37,   39,   52,   53,   54,   55,   57,   58,   62,   79,   82,  102,  104,  180,  194
...
  976,  978,  980,  982,  990, 1043, 1045, 1049, 1626
*Elset, elset=_Surf-3_S6, internal, instance="Top Paper-1"
  36,  38,  49,  51,  56,  60,  61,  81,  83,  97,  98, 100, 101, 103, 105, 106
...
 544, 545, 546, 547, 549, 550, 551, 689, 715, 721, 724, 838
*Elset, elset=_Surf-3_S5, internal, instance="Top Paper-1"
  59, 932, 984
*Surface, type=ELEMENT, name=Surf-3
_Surf-3_S4, S4
_Surf-3_S6, S6
_Surf-3_S5, S5
*Elset, elset=_Surf-4_S6, internal, instance="Bottom Paper-1"
  223,  269,  306,  317,  343,  353,  354,  355,  356,  357,  365,  814,  994, 1692
*Elset, elset=_Surf-4_S4, internal, instance="Bottom Paper-1"
 221, 304, 305, 307, 314, 315, 316, 318, 342, 351, 352, 361, 362, 364, 366, 561
 996,
*Elset, elset=_Surf-4_S3, internal, instance="Bottom Paper-1"
   19,   20,   21,   23,   25,   28,   29,   30,   31,   48,   49,   70,   82,   87,   89,   92
...
 1017, 1019, 1024, 1029, 1046
*Elset, elset=_Surf-4_S5, internal, instance="Bottom Paper-1"
   1,   3,   4,   6,  17,  27,  68,  69,  75,  76,  81,  86,  88,  90,  91, 102
...
```

```
 476, 480, 482, 757
*Surface, type=ELEMENT, name=Surf-4
_Surf-4_S6, S6
_Surf-4_S4, S4
_Surf-4_S3, S3
_Surf-4_S5, S5
*Rigid Body, ref node=Anvil-1.Anvil-1-RefPt_, elset=Anvil-1.Anvil-1
** Constraint: disp_body
*Display Body, instance=Anvil-disp-1
Anvil-1.857,
*End Assembly
**
** ELEMENT CONTROLS
**
*Section Controls, name=EC-1, hourglass=VISCOUS
0., 1., 1., 0., 0.
*Amplitude, name=Amp-1, definition=SMOOTH STEP
          0.,           0.,      1.5e-05,       0.0002
**
** MATERIALS
**
*Material, name=Hyperfoam
*Density
100000.,
** (Massscaled 100 times)
*Hyperfoam, testdata
*Uniaxial Test Data
     4.48e+07,   0.6,    0.
   -5.143e+06,  -0.1,    0.
  -1.1174e+07, -0.15,    0.
  -2.3047e+07, -0.25,    0.
   -3.918e+07,  -0.3,    0.
  -5.8665e+07, -0.35,    0.
  -7.8569e+07, -0.38,    0.
 -1.06656e+08, -0.42,    0.
 -2.17907e+08,  -0.5,    0.
 -4.54272e+08,  -0.6,    0.
*Material, name="Linear elastic"
*Density
100000.,
** (Massscaled 100 times)
*Elastic
 3e+08,0.
*Material, name=Plastic
*Density
1000.,
*Elastic
 2e+08, 0.4
*Material, name=Polymer
*Density
1000.,
*Eos, type=USUP
1500.,0.,0.
*Viscosity
100.,
**
** INTERACTION PROPERTIES
**
*Surface Interaction, name=general
**
** BOUNDARY CONDITIONS
**
** Name: Euler_X-led Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
_PickedSet238, 1, 1
** Name: Euler_Z-led Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
```

```
_PickedSet239, 3, 3
** Name: Eulerian-y Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
_PickedSet276, 2, 2
** Name: Lagr_X-led Type: Displacement/Rotation
*Boundary
_PickedSet230, 1, 1
** Name: Lagr_Y-led Type: Displacement/Rotation
*Boundary
_PickedSet231, 2, 2
** Name: Lagr_Z-led Type: Displacement/Rotation
*Boundary
_PickedSet232, 3, 3
**
** PREDEFINED FIELDS
**
** Name: Polyinit   Type: Material assignment
*Initial Conditions, type=VOLUME FRACTION
** ----------------------------------------------------------------
**
** STEP: dyn_expl
**
*Step, name=dyn_expl
*Dynamic, Explicit
, 1.5e-05
*Bulk Viscosity
0., 0.
**
** BOUNDARY CONDITIONS
**
** Name: Rigid_body_motion Type: Displacement/Rotation
*Boundary, amplitude=Amp-1
_PickedSet240, 1, 1
_PickedSet240, 2, 2, -1.
_PickedSet240, 3, 3
_PickedSet240, 4, 4
_PickedSet240, 5, 5
_PickedSet240, 6, 6
**
** INTERACTIONS
**
** Interaction: Int-1
*Contact, op=NEW
*Contact Inclusions, ALL EXTERIOR
*Contact Property Assignment
 ,  , general
*Contact controls assignment, type=scale penalty
surf-1,surf-2,9
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=1, time marks=NO
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT, number interval=100
**
** HISTORY OUTPUT: Energy
**
*Output, history
*Energy Output
ALLAE, ALLCD, ALLCW, ALLDC, ALLDMD, ALLFD, ALLIE, ALLKE, ALLMW, ALLPD, ALLPW, ALLSE, ALLVD, ALLWK, ETOTAL
*End Step
```

B. Abaqus Script Files in Python

B.1. Orphan Mesh to 3D Deformable Part

```
################################## BOTTOM PAPER-1 ##############
from abaqus import *
from abaqusConstants import *
from caeModules import *
#
# 3d Orphan to 2d Orphan
mdb.models['Model-1'].PartFromSection3DMeshByPlane('BOTTOM PAPER-2',
mdb.models['Model-1'].parts['BOTTOM PAPER-1'],(0,0,1e-5),(0,0,1),(1,0,0))
#
# 2d Orphan to Shell geometry
mdb.models['Model-1'].Part2DGeomFrom2DMesh('BOTTOM PAPER-3',
mdb.models['Model-1'].parts['BOTTOM PAPER-2'],0)
#
# Open sketch
p = mdb.models['Model-1'].parts['BOTTOM PAPER-3']
s1 = p.features['Shell planar-1'].sketch
mdb.models['Model-1'].ConstrainedSketch(name='__edit__', objectToCopy=s1)
#
# Save sketch
s2 = mdb.models['Model-1'].sketches['__edit__']
g, v, d, c = s2.geometry, s2.vertices, s2.dimensions, s2.constraints
s2.setPrimaryObject(option=SUPERIMPOSE)
p.projectReferencesOntoSketch(sketch=s2,
    upToFeature=p.features['Shell planar-1'], filter=COPLANAR_EDGES)
mdb.models['Model-1'].ConstrainedSketch(name='Bottom Paper', objectToCopy=s2)
s2.unsetPrimaryObject()
del mdb.models['Model-1'].sketches['__edit__']
#
# Create 3d deformable part
s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=0.01)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.sketchOptions.setValues(decimalPlaces=4)
s.setPrimaryObject(option=STANDALONE)
#
# Use sketch
s.retrieveSketch(sketch=mdb.models['Model-1'].sketches['Bottom Paper'])
p = mdb.models['Model-1'].Part(name='Bottom Paper', dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['Bottom Paper']
p.BaseSolidExtrude(sketch=s, depth=3.333333e-5)
s.unsetPrimaryObject()
del mdb.models['Model-1'].parts['BOTTOM PAPER-1']
del mdb.models['Model-1'].parts['BOTTOM PAPER-2']
del mdb.models['Model-1'].parts['BOTTOM PAPER-3']
#########################################################################
```

C. ABAQUS SUBROUTIN IN FORTRAN

C.1. VUVISCOSITY using Cross-Arrhenius Viscosity

```
      subroutine vuviscosity (
C Read only -
     *     nblock,
     *     jElem, kIntPt, kLayer, kSecPt,
     *     stepTime, totalTime, dt, cmname,
     *     nstatev, nfieldv, nprops,
     *     props, tempOld, tempNew, fieldOld, fieldNew,
     *     stateOld,
     *     shrRate,
C Write only -
     *     viscosity,
     *     stateNew )
C
      include 'vaba_param.inc'
C
      dimension props(nprops),
     *  tempOld(nblock),
     *  fieldOld(nblock,nfieldv),
     *  stateOld(nblock,nstatev),
     *  shrRate(nblock),
     *  tempNew(nblock),
     *  fieldNew(nblock,nfieldv),
     *  viscosity(nblock),
     *  stateNew(nblock,nstatev)
C
      character*80 cmname
C
C VARIABLES
C
n0 = 2950
n = 0.58
lambda = 2.3
E0 = 45580
R = 8.31434
absZero = -273
refTemp = 150
Tmax = 300
heatTime = 0.2
coolTime = 0.3
roomTemp = 20
coolTemp = -180
maxVisc = 200
C
C VISCOSITY SET ELEMENT WISE.
C
do k = 1, nblock
C
C  HEATINGPHASE  linear equation T=(Tmax-Roomtemp)/heattime*stepTime+Roomtemp
C
```

```
if stepTime < heatTime then
test = (n0/(1+(lamda*shrRate(k)/exp(((E0/R)*(1/(refTemp-absZero)-1/((Tmax-roomTemp)/heatTime*stepTime+roomTemp-absZero)))))**
C
C MAXIMUM VISCOSITY CHECK, BECAUSE OF VISCOSITYS INPACT ON THE TIME INCREMENT
C
if test > maxVisc then
viscosity(k) = maxVisc
else
viscosity(k) = test
end if
C
C  COOLINGPHASE   linear equation T=Tmax+Cooltemp/cooltime*(stepTime-heattime)
C
else if stepTime < heatTime then
viskosity(k)= (n0/(1+(lamda*shrRate(k)/exp(((E0/R)*(1/(refTemp-absZero)-1/(Tmax+coolTemp/coolTime*(stepTime-heatTime)-absZero
C
C MAXIMUM VISCOSITY CHECK, BECAUSE OF VISCOSITYS IMPACT ON THE TIME INCREMENT
C
if test > maxVisc then
viscosity(k) = maxVisc
else
viscosity(k) = test
end if
end if
end do
C
      return
      end
```

D. Experimental Data

D.1. Test Graphs from the Sealing Process

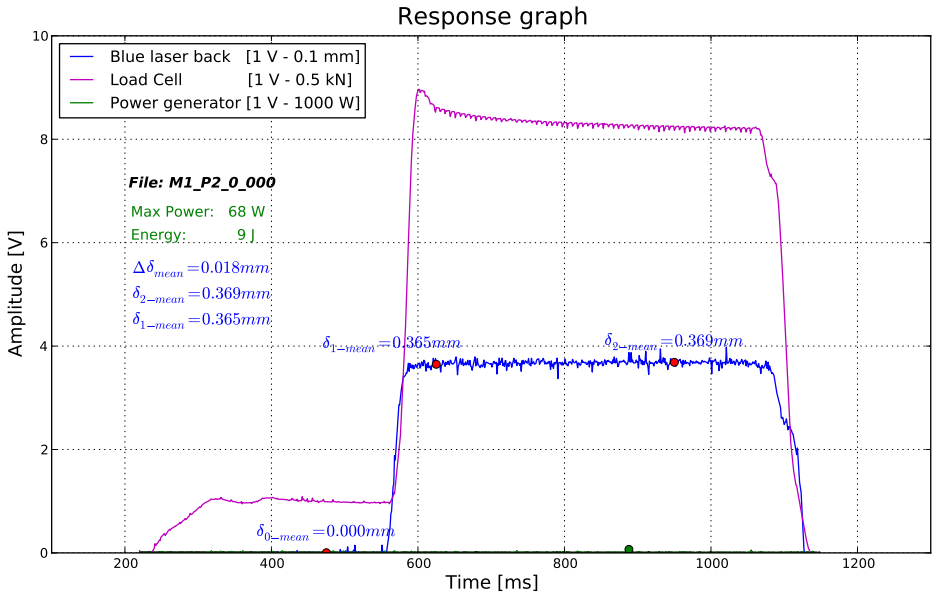


Figure D.1: Results using anvil M1, paper P2 and no heatpulse.

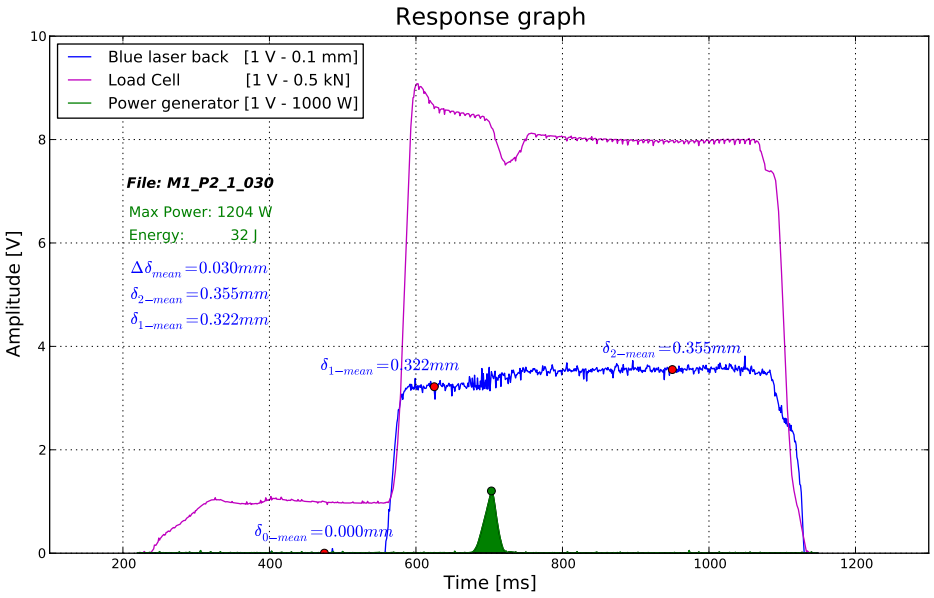


Figure D.2: Results using anvil M1, paper P2 and a 30 ms heatpulse.

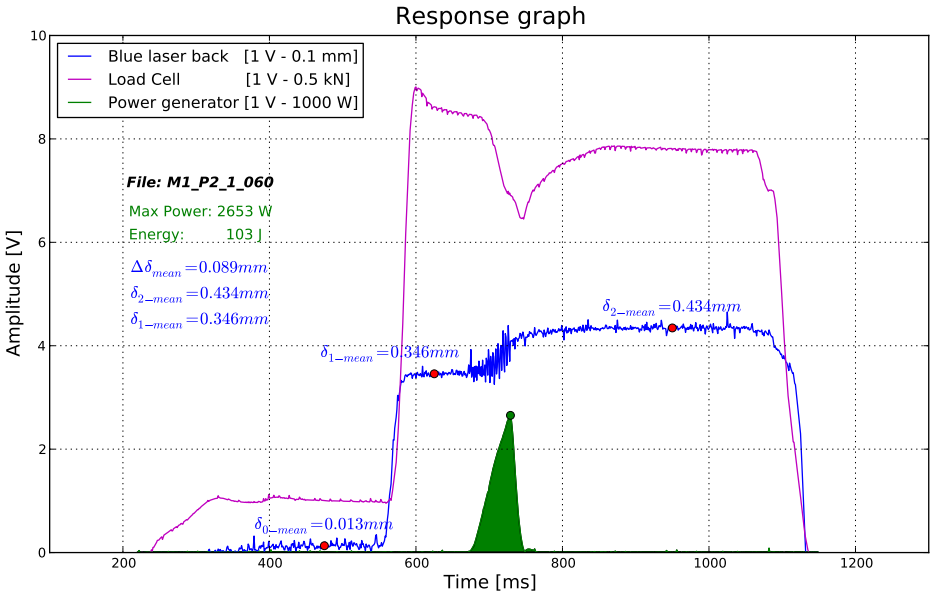


Figure D.3: Results using anvil M1, paper P2 and a 60 ms heatpulse.

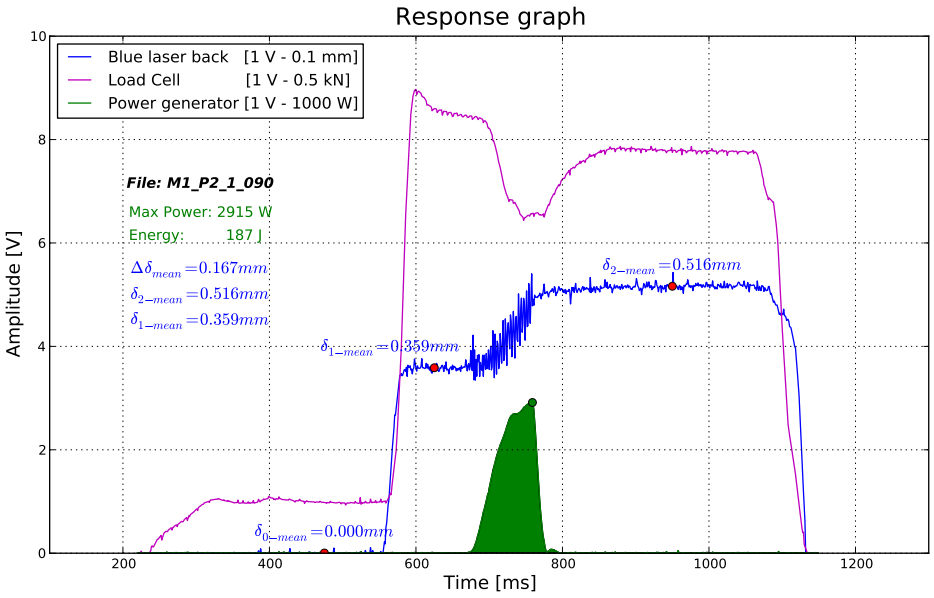


Figure D.4: Results using anvil M1, paper P2 and a 90 ms heatpulse.

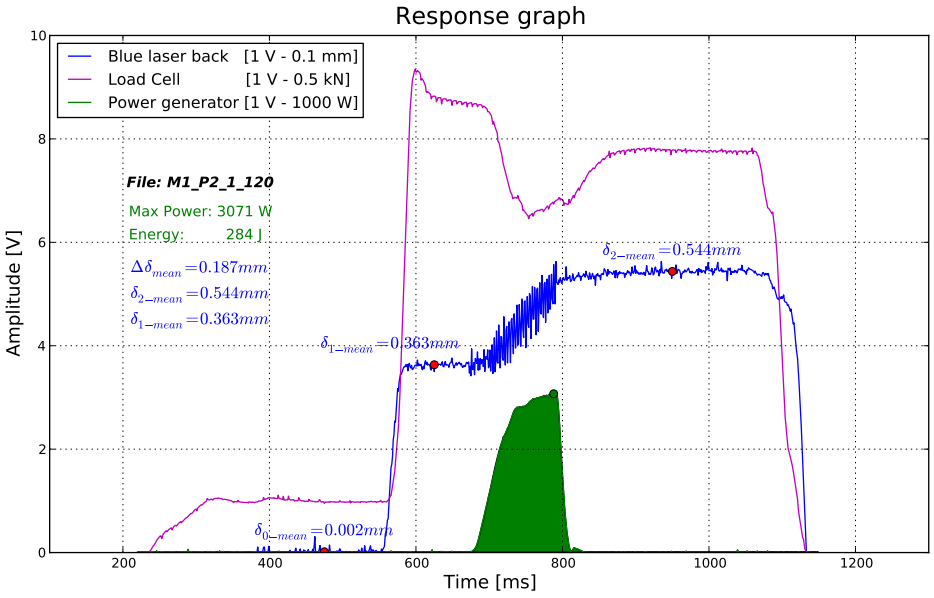


Figure D.5: Results using anvil M1, paper P2 and a 120 ms heatpulse.

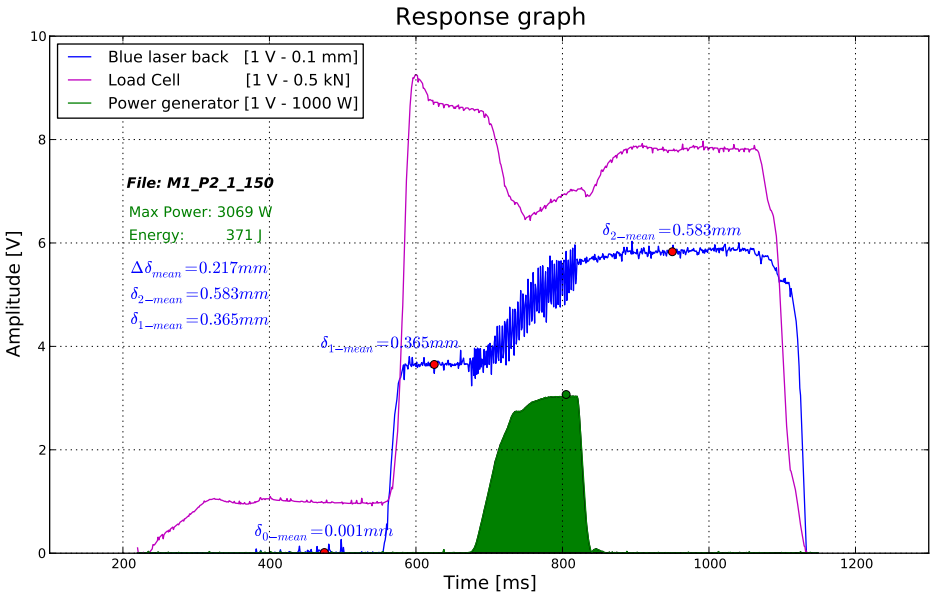


Figure D.6: Results using anvil M1, paper P2 and a 150 ms heatpulse.

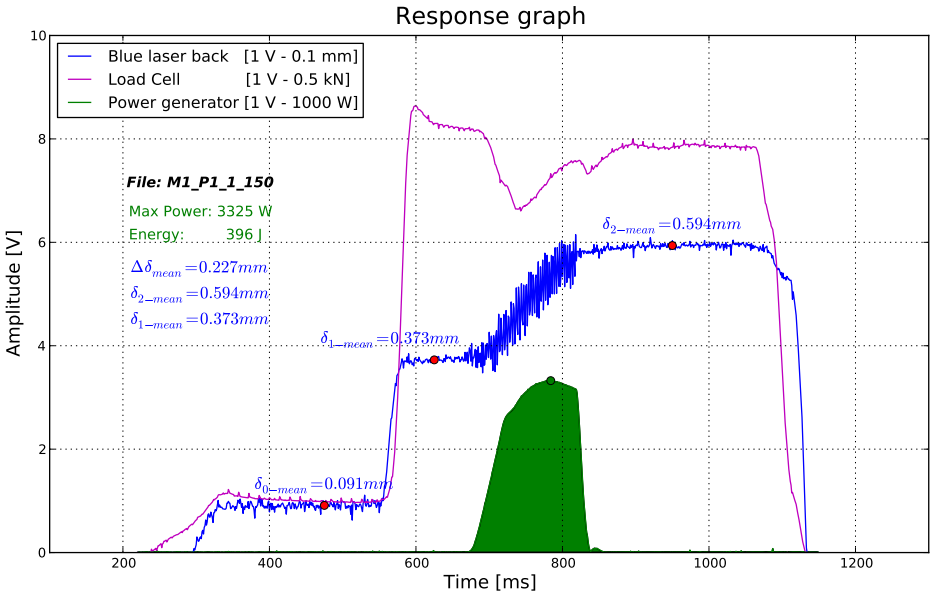


Figure D.7: Results using anvil M1, paper P1 and a 150 ms heatpulse.

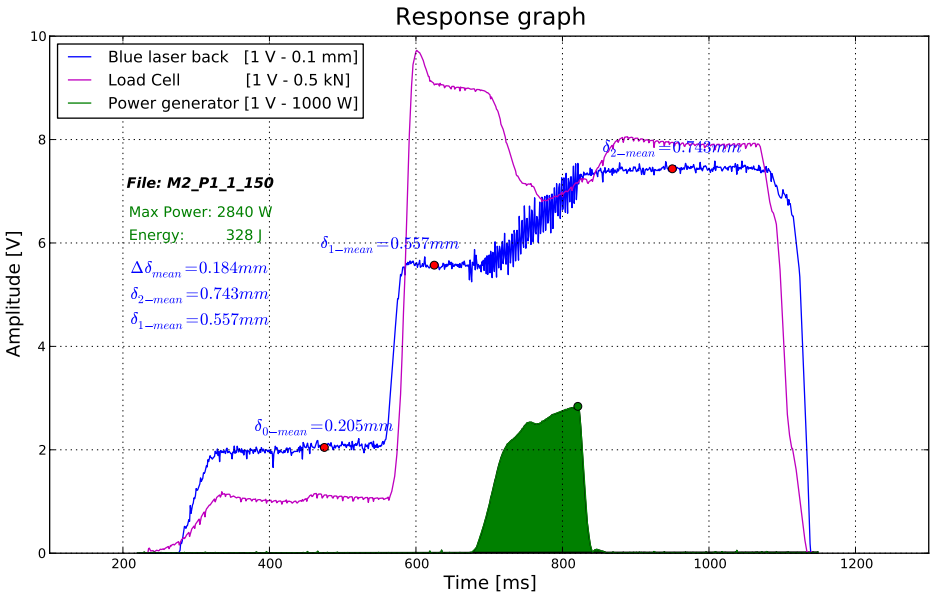


Figure D.8: Results using anvil M2, paper P1 and a 150 ms heatpulse.

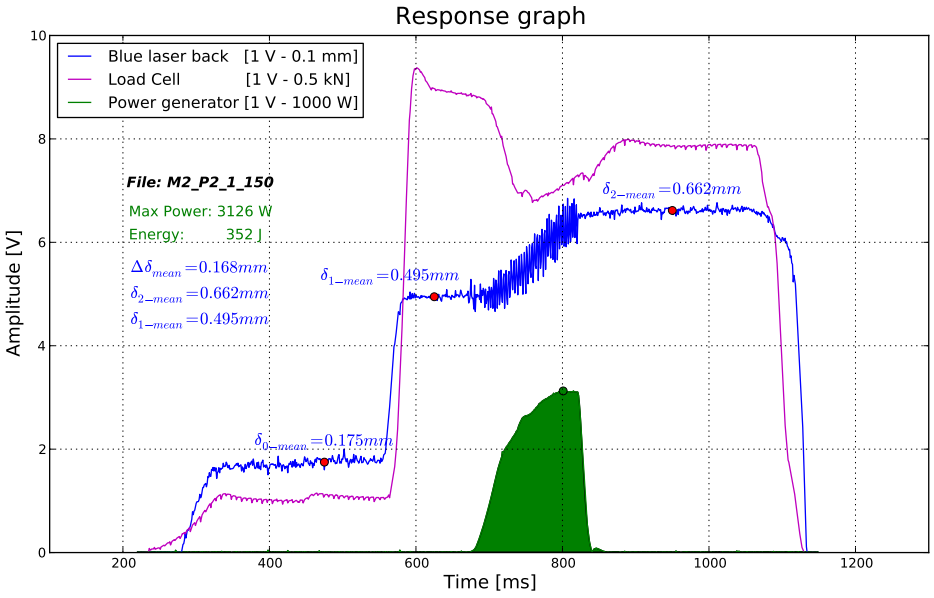


Figure D.9: Results using anvil M2, paper P2 and a 150 ms heatpulse.

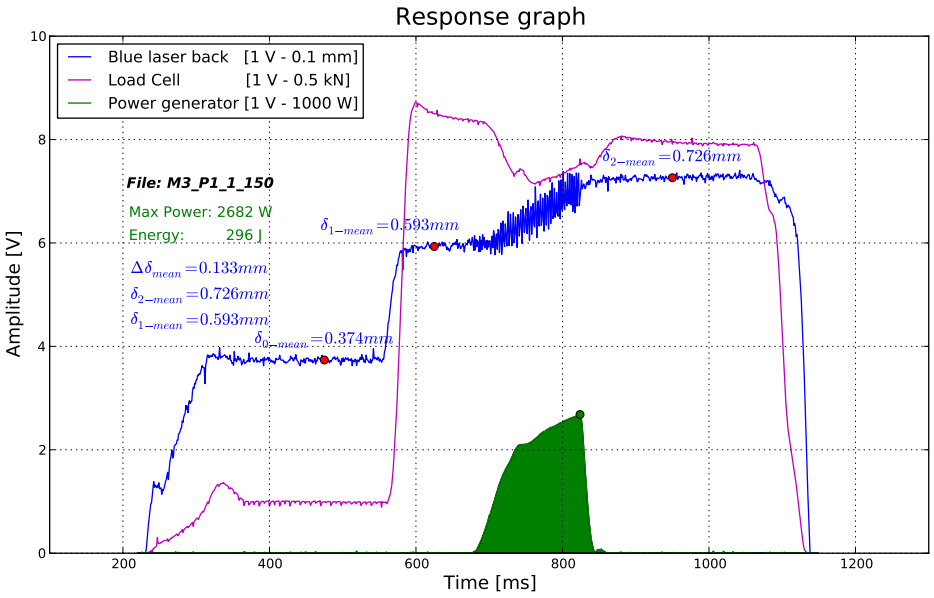


Figure D.10: Results using anvil M3, paper P1 and a 150 ms heatpulse.

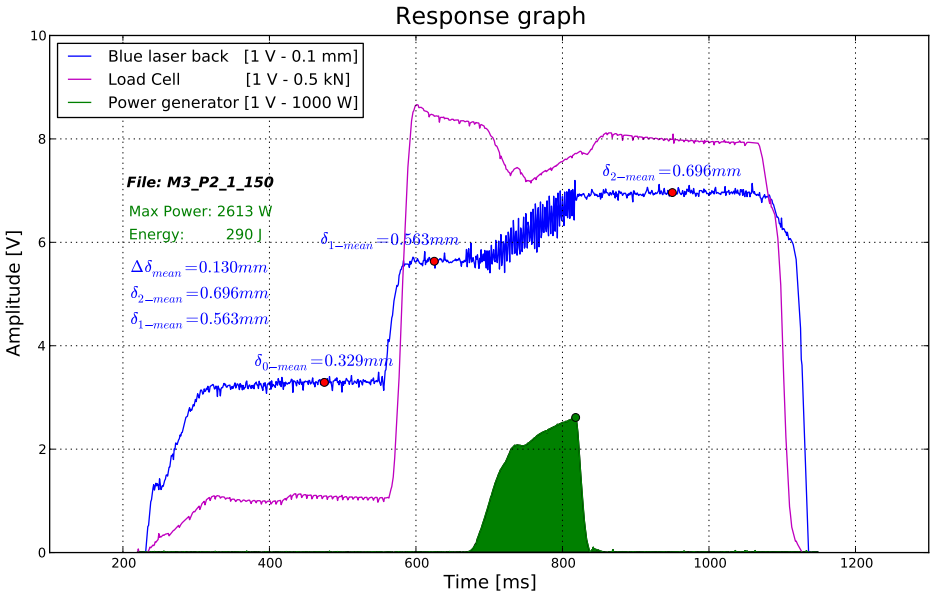


Figure D.11: Results using anvil M3, paper P2 and a 150 ms heatpulse.