MASTER

Identification of the Eindhoven Glassy Polymer model parameters using integrated digital image correlation

Joustra, J.A.

Award date:
2019

Eindhoven University of Technology

MSc. Thesis

Department Mechanics of Materials

Identification of the Eindhoven Glassy Polymer model parameters using Integrated Digital Image Correlation

Author
J.A. Joustra
0851110

Supervisors
dr.ir. J.P.M. Hoefnagels
dr.ir. L.C.A. van Breemen

September 12, 2019

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Abstract

Large deformation in polymer materials exhibits complex behavior; e.g., polycarbonate (PC) is viscoelastic, shows strain softening and hardening and is both dependent on time and history. The Eindhoven Glassy Polymer model (EGP) captures the intrinsic material behavior in a constitutive equation utilizing a minimum of 10 material parameters. A method is proposed to obtain these material parameters from a single tensile test using Integrated Digital Image Correlation (IDIC). The identification method of these constitutive model parameters is based on integrating a Finite Element Method (FEM) with Digital Image Correlation (DIC), correlating the EGP model directly to experimentally obtained images to a single inverse material identification method.

A regularization method and an updated scheme to increase robustness has been added to the IDIC routine. Virtual experiments have been conducted, which show an increase in robustness to an erroreous initial guess of a factor 3 compared to earlier work by Neggers et al. (2015). The error in both force and displacement field lowers significantly compared to the initial guess, in which the result is a linear approximation of the true multi-mode behavior of PC, as 10 parameters is not sufficient to fully describe the material behavior. The update on the material parameters lower the residual field of both the images and force data, however, without reaching the convergence criterion, the material parameters still update as the residual field reaches a steady state, contradicting the assumption of uniqueness.

Contents

1 Introduction

Polymer materials are used frequently as a base material in load-bearing applications; therefore, the predictability of the behavior and stresses in complex geometries is essential. The constitutive modeling of glassy polymers is not straightforward since the stress-strain relationship of these materials is highly nonlinear at large strains. Figure 1.1 shows an intrinsic stress-strain curve in compression.

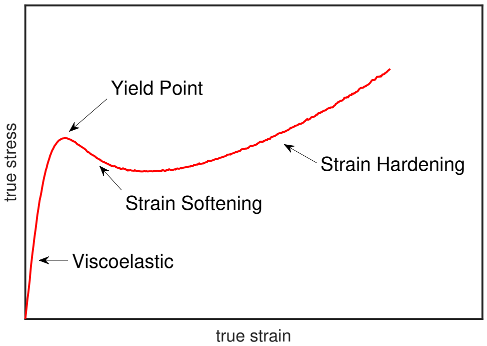


Figure 1.1: *Intrinsic stress-strain curve of a polymer sample under uniaxial compression*

This figure makes a clear distinction of 4 regimes. Initially, the material behaves viscoelastic, which is reversible. Irrecoverable deformation is induced after yielding, after which the resistance of the material to plastic deformation drops due to structural evolution, which is called strain softening [1]. Ultimately, strain hardening occurs, which is caused by the orientation of the molecular network, leading to an increase of resistance to deformation [2].
Moreover, the mechanical behavior of these glassy polymers is time- and temperature-dependent. Polymer chains tend to diffuse to a lower state of entropy, in which the material volume decreases over time. This phenomenon is called physical aging and causes the modulus and yield stress of a polymer to increase over time. Material flow has a reverse effect on the modulus and yield stress, which is, therefore, called mechanical rejuvenation [3]. In all, the observed behavior of polymers is complex. As a result, one of the key challenges in predictive modeling of glassy polymers is measurement of the correct material parameters, in which 10 or more material parameters is no exception [4]. To obtain the material parameters for polycarbonate (PC), Lexan 101R in van Breemen et al. (2009) [4], a plethora of experiments have been conducted, including uniaxial compression, uniaxial extension, planar extension and simple shear tests. All tests have to be conducted at multiple strain rates, leading to an excessive amount of mechanical tests, each sensitive to certain parameters [5, 6].

The advance of full-field, *in-situ*, real-time material testing techniques opens the potential for identification of multiple material parameters in significantly less mechanical tests [7]. The leading principle here is to compare experimentally obtained displacement fields to a numerical equivalent from a FEM. A full-field inverse material identification procedure for complex materials is challenging since not all parameters are equally sensitive to a single test. Moreover, the geometry and boundary conditions of the sample and the FEM need to be identical and the material model should describe the material behavior perfectly.

In literature, J. Neggers proposed a single inverse parameter identification procedure which directly connects a constitutive model with experimentally obtained images using Integrated Digital Image Correlation (IDIC) [8]. Work has been done on the EGP model using virtual experiments and convergence has been reached when initiated with an initial guess within 10% of the true values. This work proposes an alteration to the IDIC routine to increase the robustness against distant initial guesses for the large number of material parameters and extends the work of Neggers towards experimentally obtained images. The ultimate goal is to acquire all material parameters from a single mechanical test.

This paper is structured as follows. Section 2 explains the mathematical framework of IDIC, including the additions to the framework to increase robustness and applicability to experiments. The results of the inverse parameter identification framework is elaborated on in Sections 3 and 4, where at first the results of the virtual experiments are shown, and thereafter, the framework is tested on experimentally obtained data. Finally, a discussion and recommendations for future work is given in Section 5.

2 Methods

2.1 Integrated Digital Image Correlation

Digital Image Correlation (DIC) is commonly used to find a displacement profile $u(\bar{x})$ of a deformed image $g(\bar{x})$ to the reference image $f(\bar{x}_0)$. The main assumption is that captured images contain a pattern which follows the material deformation without affecting the properties. Therefore the brightness of each pixel must remain constant during deformation, known as brightness conservation [9]. A DIC problem is ill-posed since the input is a gray level field (scalar), whereas a displacement field (vector) is the sought result. The solution to this problem is to reduce the number of unknowns that describe this displacement field, usually done by regularizing the displacement field with a set of interpolation functions, which still requires a large number of shape functions, especially for highly non-linear deformation. DIC in its standard form, however, does not exploit known information on the kinematics of the experiment.

Once the displacement field is found, then it can be used to determine the model parameters. To this end, a well-known inverse parameter identification technique is finite element method updating (FEMU) [10]. In FEMU, the difference between the displacement fields of the simulation and experiment is minimized by iteratively updating and optimizing the model parameters. In contrast, IDIC directly correlates experimentally acquired images in a single inverse parameter identification process, which includes a Finite element (FE) simulation based on the known geometry and boundary conditions of the experiment. Utilizing these frameworks, the number of unknowns of a typical DIC problem is reduced to a limited set of material parameters, as stated in Section 2.2.
A single minimization problem is constructed, including all images over time. The leading principle is to minimize the residual, r, of the reference image, f, compared to the deformed images, g, back-deformed to the reference image; as shown in Equation 2.1 [8].

$$r_1(\bar{x}, t, \lambda) = f(\bar{x}, t_0) - \tilde{g}(\phi(\bar{x}, \lambda), t), \tag{2.1}$$

in which $f(\bar{x}, t_0)$ is the image of the reference state at time t_0 and $\tilde{g}(\phi(\bar{x}, t), t)$ is the deformed image deformed back to the reference state with the deformation map ϕ as a function of the material parameters $[\lambda]$. This residual should be equal to the image acquisition noise and interpolation errors when the correct parameters are applied, and the model describes the material behavior accurately. The deformation map ϕ is expressed in Equation 2.2.

$$\phi(\bar{x}, t, [\lambda]) = \bar{x} + \bar{u}(\bar{x}, t, [\lambda]), \tag{2.2}$$

in which $\bar{u}(\bar{x}, t, [\lambda])$ is the displacement field as function of the unknown material parameters acquired from the FE simulation and the reference positions $\bar{x}$. The minimization problem is defined by taking the quadratic residual norm of Equation 2.1 as stated in Equation 2.3.

$$[\lambda]^{opt} = \underset{\lambda}{\operatorname{argmin}}\,(\Phi_1) = \int_\tau \int_\Omega \left[r_1(\bar{x}, t, [\lambda])\right]^2 d\bar{x}dt \tag{2.3}$$

In which τ is the corresponding time sequence of the images and Ω the region of interest (ROI) over which the residual is minimized. The non-linear procedure has to be solved iteratively; The system is linearized and solved with a Newton-Raphson procedure [11]. The linearized system is defined in Equation 2.4.

$$\mathbf{M}\delta[\lambda] = [b] \tag{2.4}$$

$\mathbf{M}$ is the correlation matrix, in which the components are defined in Equation 2.5, $\mathbf{b}$ is the right-hand side member, $\delta[\lambda]$ is the sought iterative update of the material parameters of the FE simulation. The right hand side member should converge towards zero as the residual decreases as shown in Equation 2.6.

$$M_{ij} = \int_\tau \int_\Omega \left(\left(\bar{\nabla} f \cdot \psi_i\right)\left(\bar{\nabla} f \cdot \psi_j\right)\right) d\bar{x}dt \tag{2.5}$$

$$b_j = \int_\tau \int_\Omega \left(\left(\bar{\nabla} f \cdot \psi_j\right) r_1\right) d\bar{x}dt \tag{2.6}$$

In which ∇f is the image gradient and ψ_i are the basis functions, which are defined as the partial derivatives of the displacement field with respect to each of the each material parameters, as shown in Equation 2.7.

$$\psi_i = \frac{\partial u(x, t, [\lambda])}{\partial \lambda_i} \tag{2.7}$$

It is unlikely that these basis functions form a linearly independent basis as the degrees of freedom are the unknown material parameters. However, a linearly independent basis is not required for the minimization algorithm to converge. ψ_i is approximated by choosing a small perturbation ϵ and determining the change of deformation as shown in Equation 2.8.

$$\psi_i \approx \frac{u^k\left(\bar{x}, t, (1+\epsilon)\lambda_i^k\right) - u^k(x, t, \lambda_i^k)}{\epsilon\lambda_i^k} \tag{2.8}$$

The mean value of ψ_i constitutes the sensitivity of the displacement field to the perturbed parameter, λ_i. More information on the perturbation factor is given in Appendix B.5.1 on page 24.

It is of importance that additional force data is acquired otherwise, only stiffness ratios can be determined [8]. From the experiments, the force measured at the clamps can be used as additional input. This is used as a second minimization function, as proposed by, e.g., Neggers et al. [8], in Equation 2.9.

$$r_2 = F_{exp}(t) - F_{sim}(t, [\lambda]) \tag{2.9}$$

This principle is similar to the equivalent of the optical data; by minimizing a residual as a function of time. Force is a single value per time increment, whereas the images contain a 2-dimensional field each time increment, therefore, the force data contains less information. However, for modeling is it crucial that the force corresponds with the deformation as an applied force is one of the most common loading conditions. The force minimization problem by taking the quadratic residual norm of 2.9 is shown in Equation 2.10.

$$[\lambda] = \underset{\lambda}{\operatorname{argmin}}\left(\Phi_2\left([\lambda]\right)\right) = \int_\tau \left(r_2\left(t, [\lambda]\right)\right)^2 dt \tag{2.10}$$

With multiple minimization problems, a weighting parameter ($\alpha \in [0, 1]$) is defined to acquire control on the significance of the force. This parameter combines the two types of measurement data to a single minimization problem. From this, a single linearized system of equations is made, as shown in Equation 2.11. The convergence criterion is based on the 2-norm of the right-hand member.

$$\mathbf{M}_{tot}\delta[\lambda] = [b]_{tot}, \tag{2.11}$$

with,

$$\mathbf{M}_{tot} = \left((1-\alpha)\,\mathbf{M}_1 + \alpha\mathbf{M}_2\right), \tag{2.12}$$

and

$$[b]_{tot} = \left((1-\alpha)\,[b]_1 + \alpha[b]_2\right), \tag{2.13}$$

in which $\mathbf{M_1}$ and $\{\mathbf{b_1}\}$ are equal to the image equivalent seen in Equations 2.5 and 2.6. The force dependent part is defined in Equation 2.14 and 2.15.

$$M_{2(ij)} = \int_\tau (\rho_i\rho_j)\,dt \tag{2.14}$$

$$b_{2(j)} = \int_\tau (\rho_i r_2)\,dt \tag{2.15}$$

In which ρ_i are the force sensitivity maps. The sensitivity of the force to a degree of freedom can be determined similarly as the image basis functions. The definition of ρ_i is defined in Equation 2.16 and linearized in Equation 2.17.

$$\rho_i(t, [\lambda]) = \frac{\partial F\left(t, \lambda_i^k\right)}{\partial\lambda_i^k} \tag{2.16}$$

$$\approx \frac{F^k(t, (1+\epsilon)\lambda_i^k) - F^k(t, \lambda_i^k)}{\epsilon\lambda_i^k} \tag{2.17}$$

A summary of the Integrated Digital Image Correlation routine is given in Figure 2.1.

2.2 A brief description of the EGP model

The EGP model is modeled by the contributions of two mechanisms in parallel. The stress state is split in a visco-elastic contribution related to the intermolecular interactions and the contribution of the molecular network, respectively σ_s and σ_r, as shown in Equation 2.18,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_s + \boldsymbol{\sigma}_r. \tag{2.18}$$

The hardening stress is modeled with a neo-Hookian model, as shown in Equation 2.19 [5].

$$\boldsymbol{\sigma_r} = G_r\tilde{\mathbf{B}}^{\mathbf{d}} \tag{2.19}$$

The intermolecular stress contribution is subdivided in a hydrostatic and a deviatoric part, in which the deviatoric part is modeled with a number of maxwell elements, shown in Equation 2.20 [12].

$$\boldsymbol{\sigma_s} = \kappa\,(J-1)\,\mathbf{I} + \sum\left(G_i\tilde{\mathbf{B}}_{\mathbf{e,i}}\right) \tag{2.20}$$

In here, κ is the bulk modulus, J the determinant of the deformation tensor, representing the volume

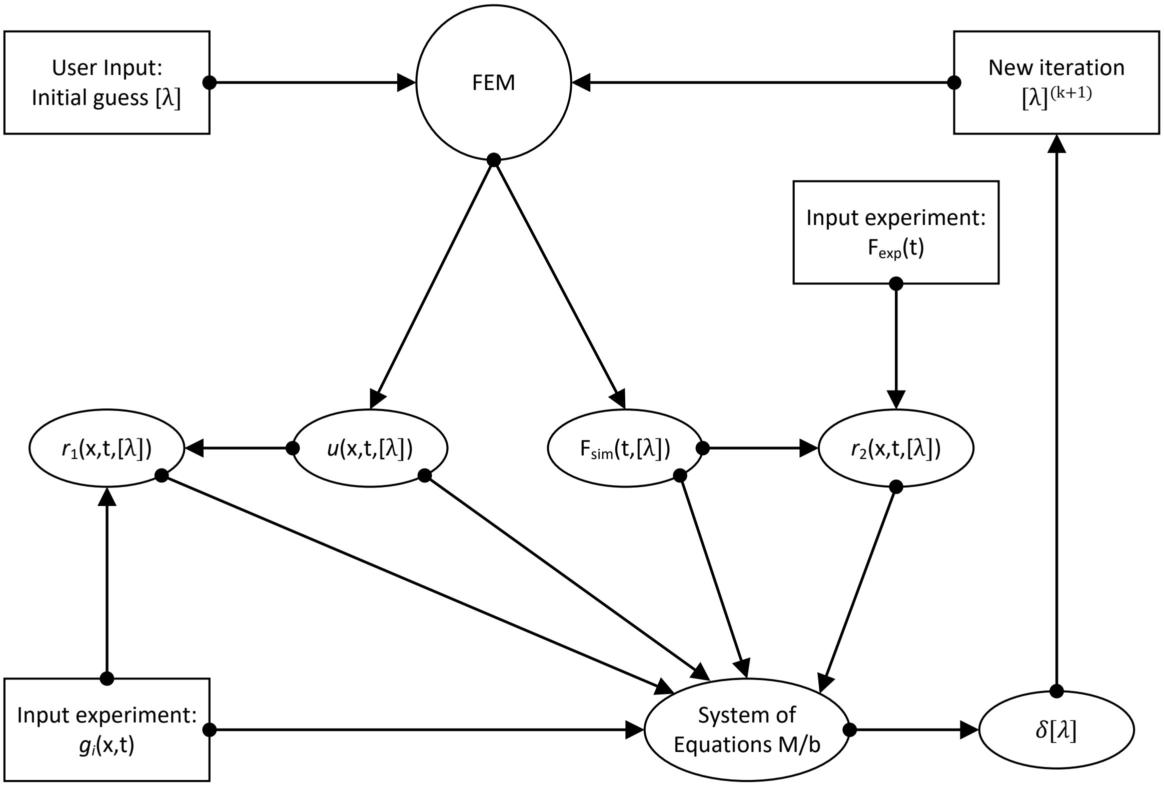


Figure 2.1: *Schematic overview of the IDIC algorithm [8]*

change, G_i is the shear modulus, G_r is the strain hardening modulus and $\tilde{\mathbf{B}}^{\mathbf{d}}_{\mathbf{e,i}}$ the elastic part of the isochoric left Cauchy-Green strain tensor as seen in Equation 2.21.

$$\tilde{\mathbf{B}}_e = J^{-2/3}\mathbf{F} \cdot \mathbf{F}^T \tag{2.21}$$

The evolution of J and $\tilde{\mathbf{B}}_{\mathbf{e,i}}$, in which the latter is modeled as the Jaumann derivative of $\tilde{\mathbf{B}}_e$, is given by Equations 2.22 and 2.23 [5], in which the subscript 'e' stands for elastic, the tilde for isochoric, 'p' for plastic and 'd' for deviatoric.

$$\dot{J} = Jtr(\mathbf{D}) \tag{2.22}$$

$$\mathring{\tilde{\mathbf{B}}}_{\mathbf{e,i}} = (\tilde{\mathbf{L}} - \mathbf{D}_{\mathbf{p,i}}) \cdot \tilde{\mathbf{B}}_{\mathbf{e}} + \tilde{\mathbf{B}}_{\mathbf{e}} \cdot (\tilde{\mathbf{L}}^c - \mathbf{D}_{\mathbf{p,i}}) \tag{2.23}$$

where $\tilde{\mathbf{L}}$ is the isochoric velocity gradient tensor and $\mathbf{D}_{\mathbf{p,i}}$ is the plastic deformation gradient tensor. The plastic deformation rate is dependent on the viscosity, which is modeled with a modified Eyring equation. The plastic deformation rate tensor is given in Equation 2.24.

$$\mathbf{D}_{\mathbf{p,i}} = \frac{\boldsymbol{\sigma^d_{s,i}}}{2\eta(\bar{\tau}, p, S)} \tag{2.24}$$

The viscosity in Equation 2.24 is dependent on the total equivalent stress, $\bar{\tau}$, hydrostatic pressure, p, according to

$$\bar{\tau} = \sqrt{\frac{1}{2}\sigma^d_s : \sigma^d_s}, \qquad p = \frac{1}{3}tr(\sigma) \tag{2.25}$$

The modified Eyring viscosity is shown in Equation 2.26.

$$\eta\left(\bar{\tau}, p, S\right) = \eta_{0,1} \frac{\bar{\tau}/\tau_0}{sinh\left(\bar{\tau}/\tau_0\right)} \exp(\frac{\mu p}{\bar{\tau}}) \exp(S(\gamma)) \tag{2.26}$$

The parameters in this equation are the zero shear viscosity, η_0, the characteristic stress, τ_0 and the pressure dependency μ. The thermodynamic state is defined in Equation 2.27 [4].

$$S(\gamma) = S_a \cdot \frac{\left((1 + (r_0 \cdot \exp(\gamma))^{r_1}\right)^{\frac{r_2-1}{r_1}}}{\left(1 + r_0^{r_1}\right)^{\frac{r_2-1}{r_1}}} \tag{2.27}$$

The softening is defined with three constants, r_0, r_1 and r_2. These parameters define the shape of the softening of the material. The thermodynamic state parameter, S_a, determines the height of the yield stress, which is affected by aging and rejuvenation. This leads to a minimum of 10 independent parameters that need to be identified, when utilizing a single Maxwell model. Table 2.1 gives an overview of these material parameters including reference values for PC from literature [4].

Table 2.1: *Material parameters in the EGP model*

	Description	Value	Unit
κ	Bulk modulus	3750	[MPa]
G_r	Hardening modulus	26	[MPa]
G_i	Shear modulus	634	[MPa]
τ_0	Characteristic stress	0.7	[MPa]
$\eta_{0,i}$	Zero shear viscosity	$1.47 \cdot 10^{11}$	[MPa$\cdot s$]
μ	Pressure dependency	0.008	[-]
r_0	Softening fitting	0.965	[-]
r_1	Softening fitting	50	[-]
r_2	Softening fitting	-3	[-]
S_a	Thermodynamic state	27	[-]

Figure 2.2 shows a simplified sketch of the influences of the EGP parameters on the stress-strain response of a glassy polymer. This is a single-mode representation of the true multi-mode behavior of glassy polymers.

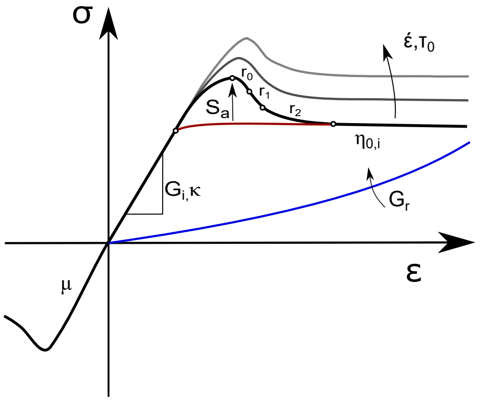


Figure 2.2: *Simplified overview of the influences of the the EGP parameters [8]*

For material characterization, polycarbonate (PC) is used. PC is a glassy polymer which has a large post-yield window, therefore all EGP parameters can be excited, moreover, the material parameters for PC are well known [4].

2.3 Sensitivity and uniqueness

As can be seen in Figure 2.2, the EGP parameters are not all active simultaneously at a material point. However, an error in for instance the moduli affects residual fields of all time increments since the error in force and displacement propagates as seen by combining Equations 2.1 and 2.3. Therefore an adjustment is made to the formulation of the residual, which is altered to the difference between two consecutive frames back-transformed to the reference image. The updated approach can be seen in Equation 2.28.

$$r_{1,t}(\bar{x}, [\lambda]) = \tilde{g}(\phi(\bar{x}, \lambda, t-1)) - \tilde{g}(\phi(\bar{x}, \lambda), t) \quad (2.28)$$

To gain a consistent scheme, the derivative of r to the perturbed parameter is rewritten as:

$$L = \frac{\partial r}{\partial \lambda_i} = \frac{\partial r}{\partial u}\frac{\partial u}{\partial \lambda_i} \quad (2.29)$$

$$= \bar{\nabla}\tilde{g}_{t-1}\frac{\partial u(t-1)}{\partial \lambda_i} - \bar{\nabla}\tilde{g}_t\frac{\partial u(t)}{\partial \lambda_i} \quad (2.30)$$

$\bar{\nabla}\tilde{g}_{t-1}$ can be considered equal to $\bar{\nabla}\tilde{g}_t$ for small deformations as both are backtransformed images to the reference frame. Equation 2.30 can be rewritten as:

$$L = \bar{\nabla}\tilde{g}_t\psi_i, \quad (2.31)$$

with,

$$\psi_i = \frac{\partial u(x, t-1, \lambda)}{\partial \lambda_i} - \frac{\partial u(x, t, \lambda)}{\partial \lambda_i}. \quad (2.32)$$

With this approach, the residual is unique per increment instead of a summation of all residuals over time. This moreover decreases the requirement of brightness conservation for all frames between the first and last frame, making it less affected by pattern degradation.

All 10 parameters in the EGP model have a different effect on the deformation behavior and reaction force. Figure 2.3 shows the change in displacement data for each perturbed parameter on a single x-t plane, located at the plane with maximum deformation. The sample for the numerical experiment, including a pattern and the x-t plane is shown in Figure 2.4. The sensitivity maps are scaled to their h

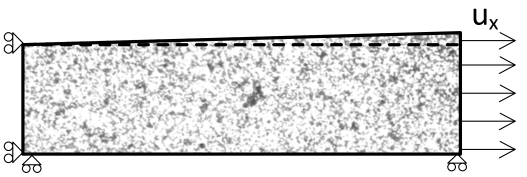


Figure 2.4: *Numerical experiment geometry including pattern the x-plane (dashed line) for sensitivity analysis. Only $^1/_4{}^{th}$ of the tensile bar is modeled to reduce computational time.*

maximum absolute value of $\partial u/\partial\lambda$ (i.e. scaling factor $\bar(S_f)$).

The shape of the sensitivity fields of parameter perturbations can be similar as seen in Figure 2.3, τ_0 and $\eta_{0,1}$ have the same shape for example, but a different sensitivity factor. This can be an indication of non-uniqueness. The sensitivity can be mapped in a single sensitivity map, the values in the M-matrix derived in Equation 2.5 give a representation of these cross-terms and are shown in Figure 2.5.

In this figure, the on-diagonal values give a measure of the sensitivity of the parameter, the off-diagonal values are a measure of the similarity between two parameters. The parameter $r1$ on position 8, for example, has a relatively low sensitivity, hence the low scaling factor in Figure 2.5, however, the shape of the sensitivity field is quite similar to the negative field of r_2, on position 9. The scaling factor of r_2 is larger, therefore, position [8,9] and [9,8] are larger than the on-diagonal value of r_1, but lower than the on-diagonal value of r_2, the red dot in the middle indicates that the value is negative. Note that the scaling bar of the

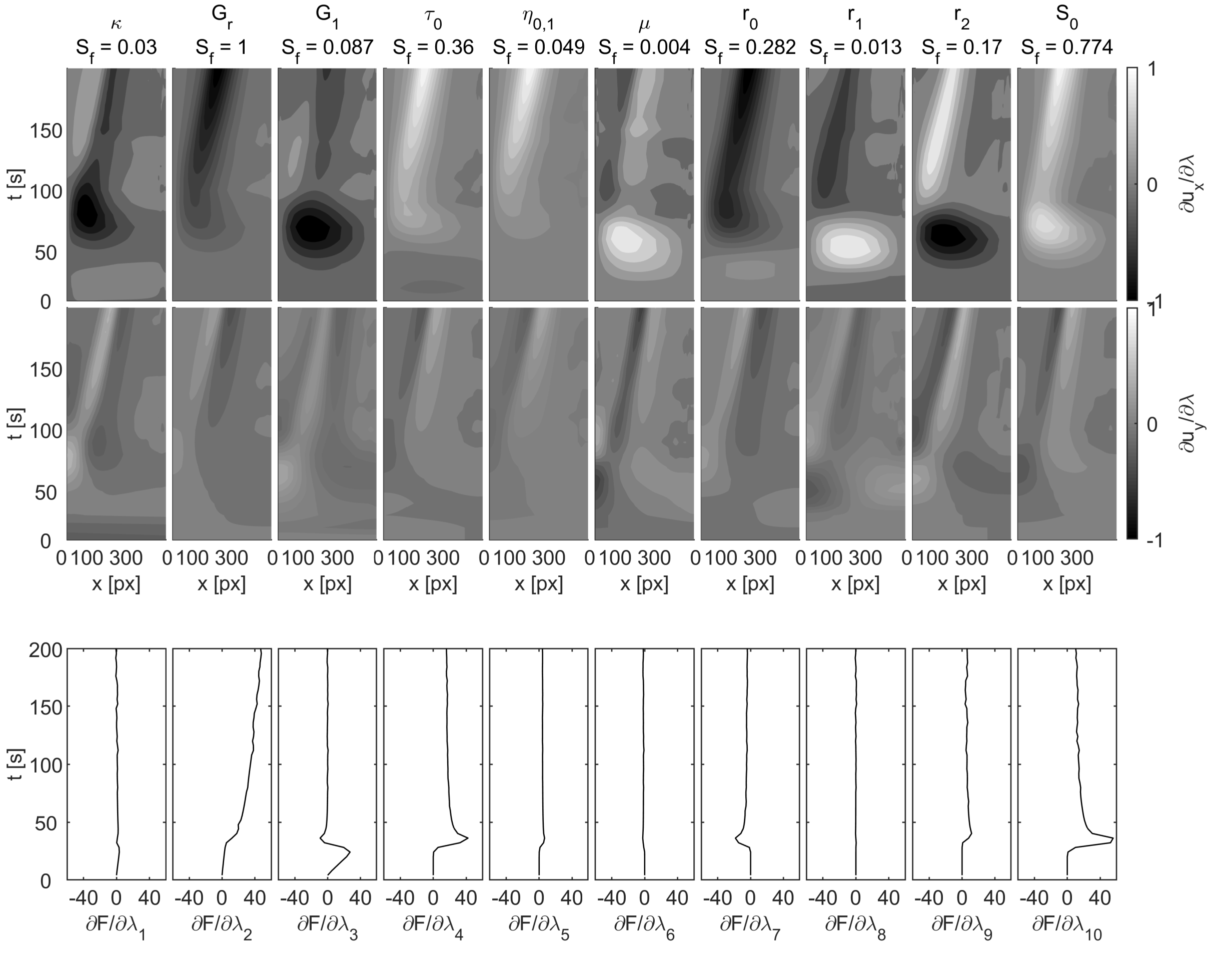


Figure 2.3: *The effect of a parameter perturbation (10%) on the displacement field and force*

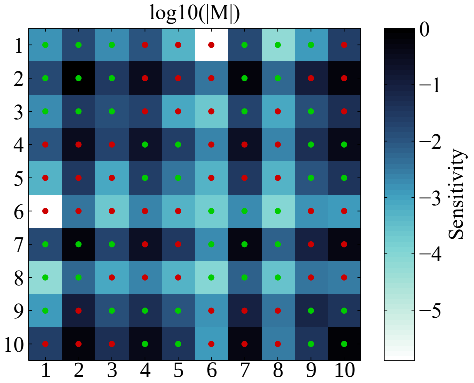


Figure 2.5: *Sensitivity map of the virtual experiment with parameters in the same order as Table 2.1*

sensitivity is on a logarithmic scale and is scaled to the highest sensitivity value.

A challenge with this cross sensitivity is that values with lower sensitivities but high off diagonal values can tend to overcompensate for an error in the residual field. For instance, looking at $\partial u_x/\partial x$, an increase of 10% of τ_0 has the same effect as an increase of 79% of $\eta_{0,1}$, but can also be compensated by a decrease of 3% of G_r. Therefore, parameters with high off-diagonal sensitivity values could indicate non-uniqueness and are highly susceptible to run away, which is troublesome as they can move towards non-physical values, in which the FEM simulation seizes to converge. A solution to this phenomenon is to dampen the insensitive parameters for the first iterations. The regularization method used was first proposed by Tikhonov et al. for ill-posed systems [13]. Tikhonov regularization is commonly applied if the a priori information is insufficient to single out a set of well-posedness. The formulation of the applied Tikhonov regularization to Equation 2.4 is given in Equation 2.33

$$[\mathbf{M} + \alpha_T \mathbf{I}]\delta[\lambda] = \{\mathbf{b}\}, \tag{2.33}$$

in which $\mathbf{I}$ is the identity matrix of equal size as $\mathbf{M}$ and α_T a scalar to obtain additional stiffness to $\mathbf{M}$. α_T is obtained by an eigenvector decomposition and is defined equal to the eigenvalues of the corresponding vectors. Ten values are found and α_T is released stepwise as the update of λ decreases.

3 Virtual Experiments

To attempt to identify all EGP parameters accurately, it is of importance that all material parameters are excited and have an influence on either the observed pattern deformation or the response in force. As a PC sample deforms, a neck grows stably due to the combination of strain softening and strain hardening. The area around the neck, moreover, creates a gradient in strain and strain rate. Therefore, it is assumed that a simple tensile test is sufficient to capture all EGP parameters.

As proof of concept, virtual experiments are conducted similar to the work of Neggers [8]. The advantage of virtual experiments is that the exact material parameters can be chosen beforehand and can be compared to the converged values. The material parameters are chosen to match the reference values for PC [4]. Figure 2.5 shows the sensitivity map for the simple geometry, as shown in Figure 2.4. However, this geometry could be optimized for a better sensitivity distribution.

3.1 Geometry Optimization

The geometry of the specimen influences the deformation behavior and therefore, the sensitivity of the parameters to the deformation. The goal of a different geometry is to optimize the sensitivity field to have less cross-sensitivity; these off-diagonal values in the sensitivity matrix can cause instabilities and non-uniqueness. Virtual experiments are used to attempt to optimize sensitivity.

A 3D approach is needed as the thickness of the sample is not constant in loading, and plane stress functionality is not implemented in the EGP user subroutine HYPELA2 in the finite element package MSC Marc. At least 3 linear elements are needed in the thickness direction to allow for a gradient in strain.

Firstly, a geometry with a single hole is used. A 10 mm wide specimen with a hole of 3 mm is loaded in axial direction. Loading a specimen with a hole initiates a more complex stress distribution due to stress concentrations. Figure 3.2 shows the combined sensitivity map of the image and force data. The sensitivity of this experiment is in general lower than the simple notched tensile test; therefore this test has been excluded. Other geometries, such as

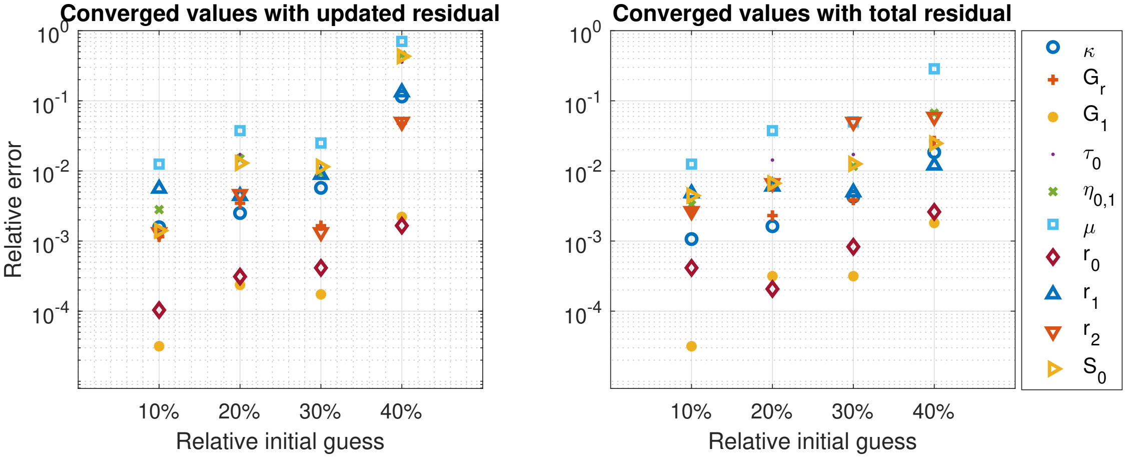


Figure 3.1: *Comparison of the updated residual and total residual on numerical convergence of all 10 EGP parameters*

a specimen with two holes, differently sized holes and a wide sample clamped on both sides. All results, however, show a sensitivity map with lower values than the initial experiment and, most importantly, no convergence is reached in the IDIC routine. The sensitivity maps of these tests are found in Appendix C.

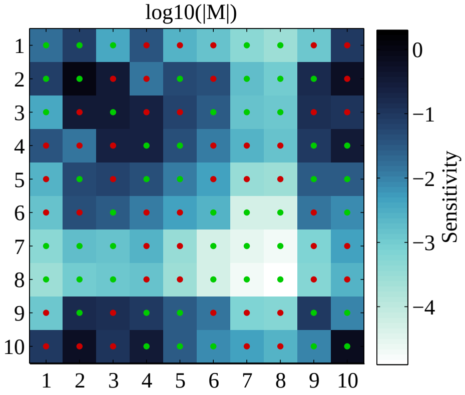


Figure 3.2: *Sensitivity map of a tensile bar with a single hole*

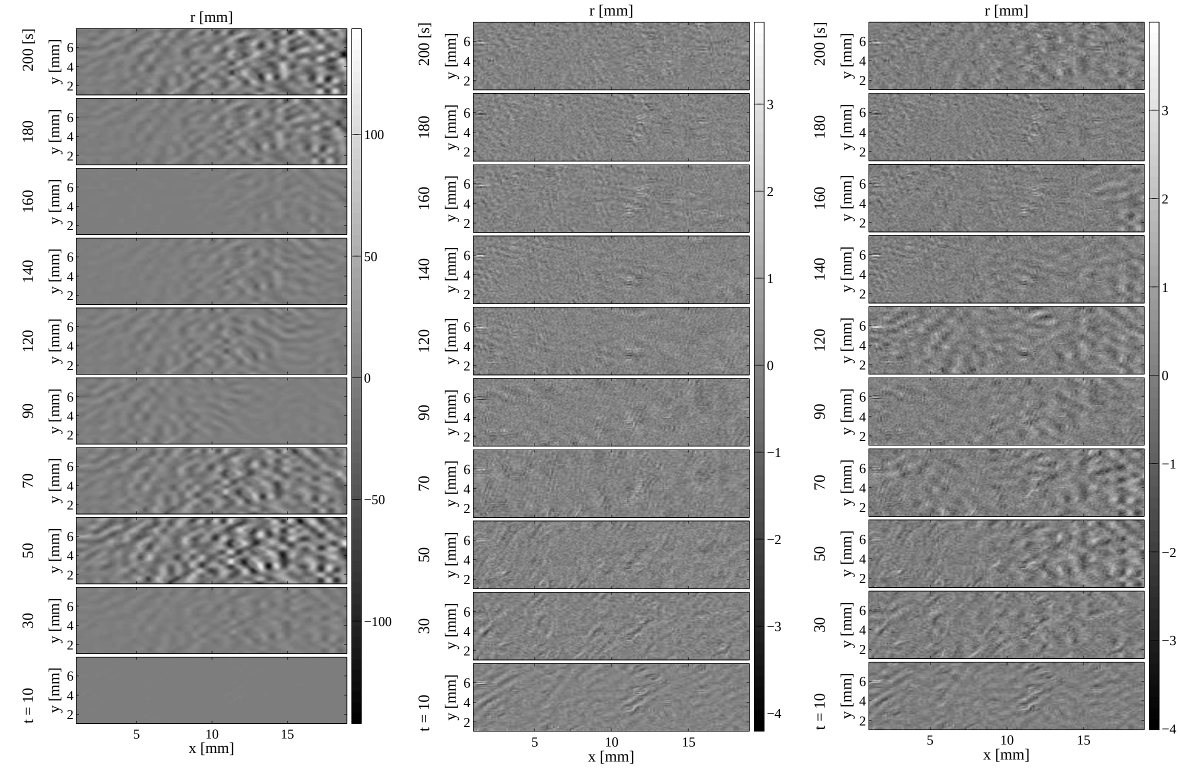
Therefore, the simple initial loading geometry has been chosen to use for the experiments as seen in Figure 2.4.

3.2 Results of virtual experiments

The applied Newton-Raphson scheme is known for excellent convergence properties when close to the solution of a convex, unique problem [14]. However, with 10 parameters, the probability of having local minima is significant. To test the robustness against various initial guesses with the proposed method in Section 2.3, this virtual experiment is initiated at different initial guesses, in which all parameters are multiplied by a factor. Figure 3.1 shows the converged values of the material parameters at various initial guesses for both the updated residual and the total residual.

For all initiated IDIC routines with the updated residual scheme up to 30%, all parameters converge within 2% of the exact values with a convergence criterion of $|\mathbf{b}| = 10^{-3}$, except for the pressure dependent parameter μ and fitting parameter r_2 at an offset of 30%. As can be seen from Figure 2.3 and 2.5, the sensitivity of this pressure dependency is the lowest of all parameters. The cross-terms of multiple parameters are higher than the diagonal term, which means that these parameters have similar deformation behavior, but this deformation behavior results from a smaller perturbation of another parameter than that of the on-diagonal parameter itself. Figure 3.3 shows the residual field of a non-converged sample and the converged residual for an offset of 30%.

The result of the initiated IDIC routine with a 40% offset is an example of a local minimum. This can be seen from the residual field in Figure 3.3. For the 30% offset case, the residual field is minimized to interpolation errors, and only a slight pattern can be recognized. The force residual is decreased to an almost negligible 3N compared to the



(a)

(b)

(c)

Figure 3.3: *Residual fields of virtual experiment: Initial guess at a 30% offset* ***(a)****, Residual after convergence from 30% offset* ***(b)****, Residual after convergence from 40% offset; converged to a local minimum* ***(c)****. Note the difference in scale bars.*

initial maximum error of 94N, as seen in Figure 3.4.

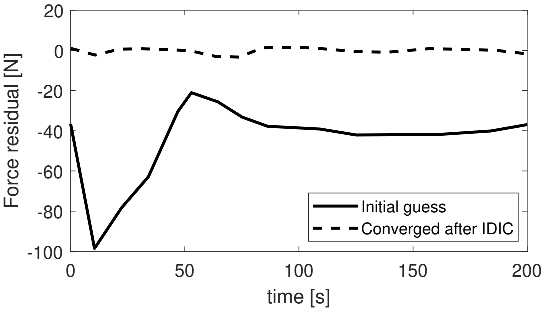


Figure 3.4: *The force residual at an initial guess 30% off and after convergence of the IDIC routine*

In previous work of Neggers et al. [8], convergence has been reached for an initial guess 10% off of all 10 parameters. With the applied Tikhonov regularization and updated image scheme, convergence is reached when initiated at an offset 30% of the reference. Equally good results are acquired with the total residual as proposed by Neggers [8]; however, the updated residual method is preferred in the experiments, as explained in Section 4.1. From these numerical results it can be concluded that the updated IDIC and Tikhonov regularization both work and that the regularization method makes the optimization routine more robust. These results provide a promising basis to move towards experiments, which are discussed in the next chapter.

4 Material Characterization

Experiments are bound to become more complicated than the numerical equivalent. This chapter firstly covers the applied methods for the experiments, after that a discussion on the complications and solutions. Lastly, the results from the IDIC routine is discussed.

4.1 Experimental Sample

A simple geometry PC sample is made, a geometry equal to the virtual experiments is not possible due to the sharp edge in the middle, see Figure 2.4, which causes the sample to fail. In the new sample, a smooth edge is created by milling a dog bone shape. Figure 4.1 shows the dimensions of the sample. All values here are in mm. The thickness of the sample is 1 mm.

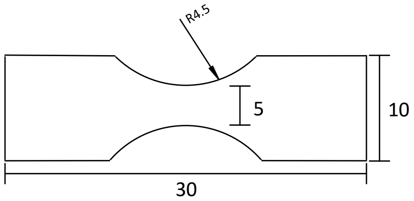


Figure 4.1: *Dimensions of the PC sample in experiments*

The minimum width of this sample is 5 mm; therefore, the maximum stress is in the center, ensuring necking to occur in this specific location, which is also the case in the FE simulation. The IDIC routine is highly sensitive to the exact dimensions of the sample. The FEM model, therefore, is not defined with the exact dimensions stated in Figure 4.1. The model is derived directly from the acquired images, this is further explained in Appendix B.2 on page 20. A speckle pattern is applied to the sample by using graphite powder and white chalk powder. Applying paint is not feasible for these samples since the large deformations cause the pattern to crack. No discontinuities are found when the material is loaded to the point of crack initiation with the applied powder, which is at a global strain of approximately 100%. However, the pattern is darkening at the large deformations in and around the neck.

4.2 Experimental Setup

An in-situ micro tensile stage is used to conduct a tensile test on the sample. A camera mounted on a microscope is used to acquire images during deformation. The tensile stage provides additional data on the applied force. The sample is clamped in a sandwich structure with two screws and loaded with a displacement control at $20\mu m$ per second. The experimental setup is shown in Figure 4.2.

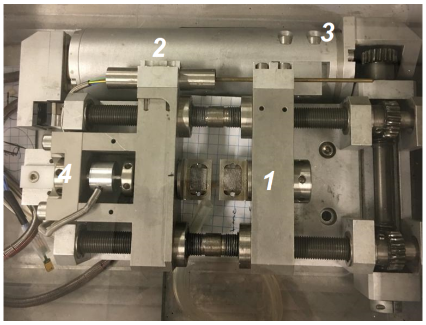


Figure 4.2: *Experimental setup; 1 Clamp; 2 Parallel Extensiometer; 3 Actuator; 4 Load Cell of 500N*

The loading geometry is 3-dimensional, therefore, there is a contraction in the out-of-plane direction. The thickness of the polycarbonate sample after the test is reduced from 1 to 0.75mm, which is a working distance change of 0.125mm relative to the camera. At a working distance of 50mm, which is a 2.5% change in depth, this change can cause the image to defocus. A countermeasure for this effect could be to reduce the thickness of the sample, however, a thicker sample yields more volumetric information, and is thus more sensitive to both the bulk modulus and pressure dependency μ. Therefore, the Zeiss Discovery.V20 stereo-microscope is used, which has a forgiving depth of field [15].

4.3 Boundary conditions

The boundary conditions of the FEM model are determined directly from a DIC analysis of the experimental images. The acquired displacements are applied to the respective nodes on the right and left edge of the sample similarly to Ruybalid (2019) [16]. The boundary conditions are assumed constant through the thickness of the sample. The application of the boundary conditions are explained in more detail in Appendix B.3 on page 20.

4.4 EGP accuracy with 10 parameters

A single-mode EGP model only captures the intrinsic post-yield behavior of glassy polymers accurately. Van Breemen et al. [4] extends the EGP model to a multi-mode model and show that 17

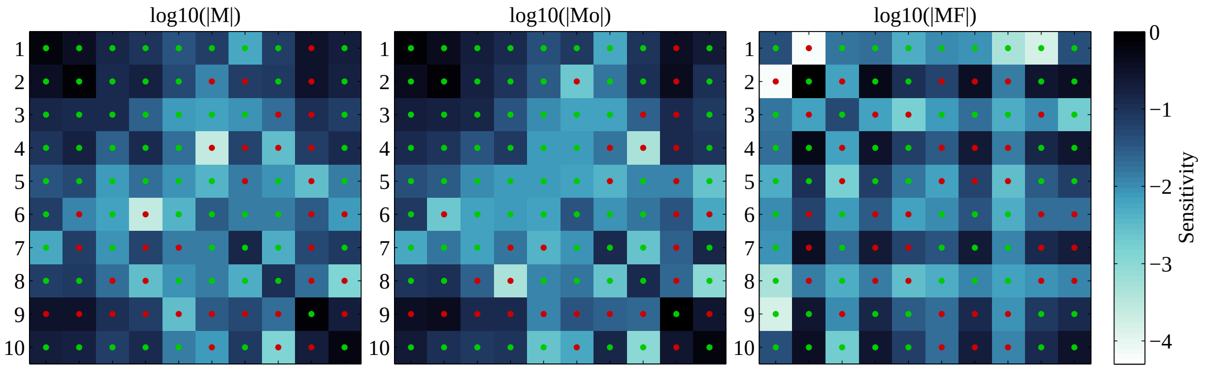


Figure 4.3: *Sensitivity map of a dog bone shape test case, with Mo the sensitivity of the image , MF the sensitivity of the force and M the combined sensitivity fields with α_T*

parameters for the shear modulus spectrum is sufficient to describe the material. However, to reduce the number of variables in the IDIC routine, a single-mode EGP model is used in this routine; therefore, an inaccuracy is expected of the viscoelastic parameters (G_1, κ). The single-mode approximation versus the true response is shown in Figure 4.4.

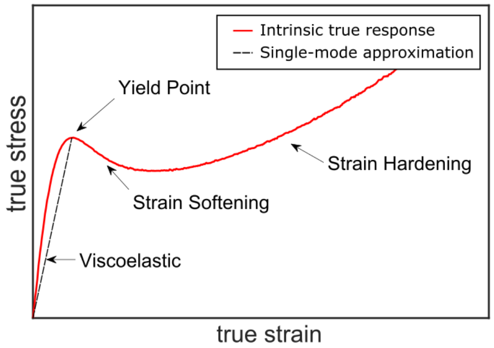


Figure 4.4: *Stress strain relationship of a single mode model versus the true response*

4.5 Convergence and Sensitivity

Convergence could be rather troublesome since the sensitivity of all parameters is different; furthermore, they strongly influence each other. This could imply that if a parameter does not converge, other parameters also seize to converge. Figure 4.3 shows the sensitivity to the parameters, in which Mo is the sensitivity matrix of the image field, and MF is the sensitivity matrix of the Force increments. As can be seen, the sensitivity matrix has significant off-diagonal values. This indicates that these parameters are coupled, which could indicate that the solution is non-unique.

4.6 EGP parameters from a single tensile experiment

The results after convergence of the initiated IDIC routine are shown in Table 4.1. The code was initiated at the reference values known for PC, which are stated in this table. All 10 parameters are free to update; however, due to the Tikhonov regularization, the parameters with lower sensitivities are gradually released. Figure 4.5 shows the convergence behavior of the IDIC routine, the mean residual of both the image and force, the relative average update of the parameters and the mean of the b-vector, as function of the iteration count.

As can be seen from Table 4.1, the converged bulk modulus is lower, and the shear modulus is higher than the reference value. As the loading is in axial direction, a Youngs-modulus and Poisson ratio can be determined and compared using the linear isotropic relations of Equation 4.1.

$$E = \frac{9\kappa G}{3\kappa + G} \quad , \quad \nu = \frac{3\kappa - 2G}{2(3\kappa + G)} \tag{4.1}$$

The reference and identified parameters of the Youngs-modulus are respectively: 1.8 GPa and 2.09 GPa, which are typical values for PC. The reference Poisson ratio is 0.42, while a value of 0.39 has been identified. These values are within the experimental uncertainty of the methods with which the

Table 4.1: *Material parameters converged from the IDIC routine compared to the reference values*

Symbol	IDIC	Reference	Unit
κ	3056	3750	[MPa]
G_r	31.85	26	[MPa]
G_1	753	634	[MPa]
τ_0	0.785	0.7	[MPa]
η_0	$3.72 \cdot 10^{10}$	$1.47 \cdot 10^{11}$	[MPa$\cdot s$]
μ	0.0085	0.008	[-]
r_0	0.9215	0.965	[-]
r_1	51.96	50	[-]
r_2	-2.798	-3	[-]
S_a	33.89	27	[-]

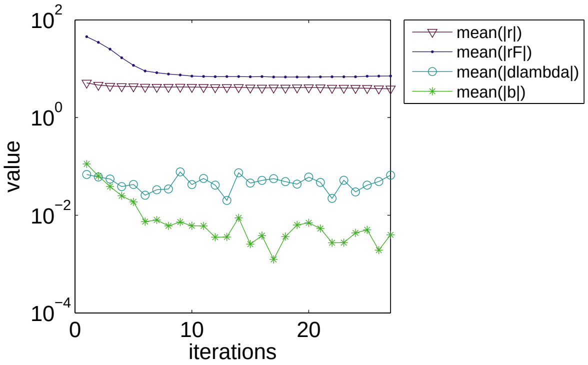


Figure 4.5: *The average residuals, parameter updates and b values plotted over the iteration count*

reference values have been determined. Moreover, considering the EGP model with a single-mode approximation was used for the identification, an inaccuracy in the linear elastic material parameters is expected is expected, as explained in Section 4.4. Another noticeable difference in the reference values and the results of the IDIC routine is the hardening parameter G_r and the thermodynamic state S_a. As seen in Figure 4.6, the force is not identified completely accurate, although the error in the force improved greatly compared to the reference values. A difference in yield stress is found, and more hardening is seen than in the experimental force data. The residual of the deformation field, however, is low. Both parameters have a high sensitivity on both the deformation field and force data as shown in Figure 2.5; however, these parameters move towards a minimum in deformation residual rather than a minimum in force residual, since the sensitivity of the image is stronger. Furthermore, the α value used in the IDIC routine is 0.1, which gives more weight on the image residual. Therefore, it can be concluded that the minimum of the force residual and image residual both have different material parameter combinations.

Figure 4.7 shows the parameter value relative to the initial guess. A steady-state is found in the image and force residual; however, the parameters update freely as the regularization method has stopped after iteration 17. The residuals both are nearly unaffected by the change of the parameters which indicates that multiple parameter combinations results in almost the same deformation and force response. It can be concluded that the solution to this IDIC routine is not fully unique.

The results of the IDIC routine at an initial guess 10% lower than the reference values shows strongly different results. The average residual of both force and the image are similar to the results of the routine initiated at the reference values; however, the material parameters differ significantly to the previously obtained results, as shown in Table 4.2.
An unexpected outlier is the bulk modulus. The bulk modulus of a material is a measure to determine how much the sample compresses under external pressure; therefore, a tensile test is not the most appropriate experiment to determine this parameter. For future research, it is recommended

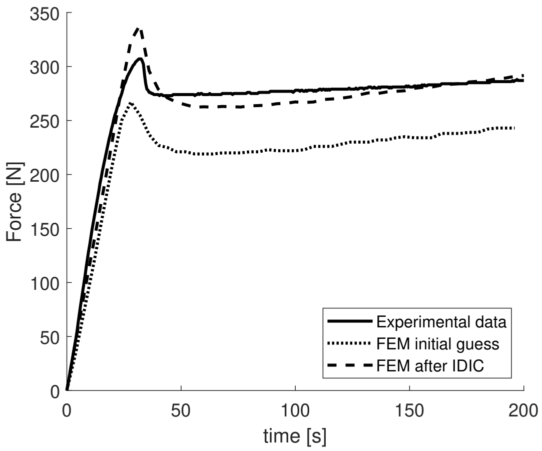


Figure 4.6: *Force time diagram of the experiment, initial guess and converged after the IDIC routine*

Table 4.2: *Material parameters initiated at 10% lower than reference values from the IDIC routine compared to the reference values*

Symbol	IDIC	Reference	Unit
κ	1614	3750	[MPa]
G_r	34.66	26	[MPa]
G_1	858.37	634	[MPa]
τ_0	0.808	0.7	[MPa]
η_0	$6.254 \cdot 10^8$	$1.47 \cdot 10^{11}$	[MPa$\cdot s$]
μ	0.00895	0.008	[-]
r_0	0.9162	0.965	[-]
r_1	49.12	50	[-]
r_2	-3.39	-3	[-]
S_a	40.8	27	[-]

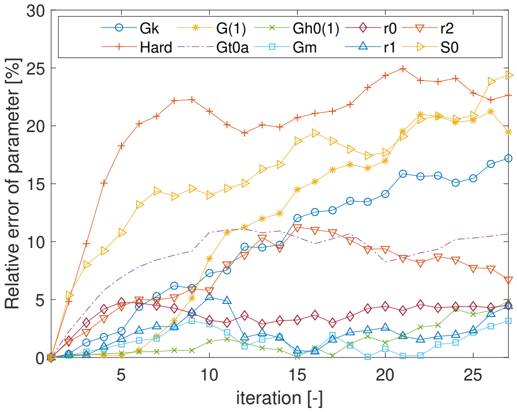


Figure 4.7: *Parameter update of IDIC routine*

to obtain a set of optimized experiments, such that all parameters are activated. Adding compression test, for example, increases the sensitivity for both the bulk modulus and the pressure dependent parameter μ.

Interestingly, the same IDIC routine has been initiated without the regularization method; all parameters directly have a maximum update and have runaway parameters after 3 iterations. Without the updated approach explained in Section 2.3, the residual field from the initial guess is too large due to pattern darkening, the update is too large initially and these cause runaway or infeasible parameter combinations, like with the simulation without the regularization method. The experimental displacement data is obtained from a global DIC analysis in the region of interest, as shown in Figure 4.8. This field is compared to the nodal data of the FEM model at the first and last iteration. The displacement data between the nodes are interpolated linearly to compare the fields. It can be seen that the difference in both x- and y-displacement significantly decreases with the IDIC routine. The highest error is found in the x-direction in the area in front of the growing neck, for which no plastic deformation has occurred as the yield stress has not been reached yet. Therefore, the leading parameters are the shear and bulk modulus, which are insufficient to describe the viscoelastic behavior of PC as discussed before.

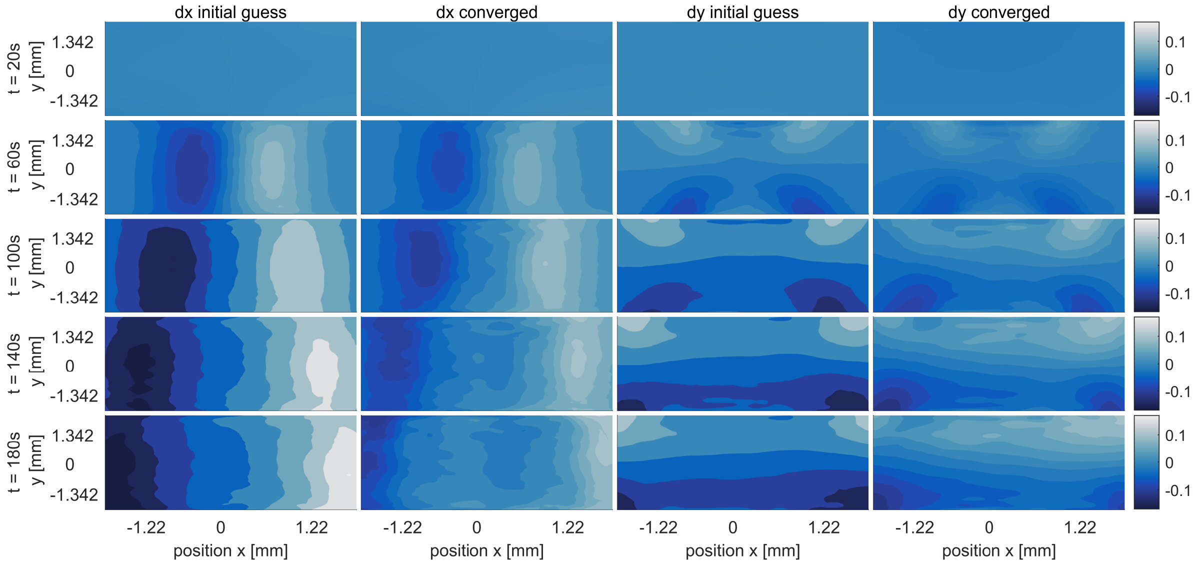


Figure 4.8: *Differences in x and y displacement of DIC and FEM results before and after IDIC routine, initiated at the reference values*

5 IDIC for EGP Discussion and Recommendations

Typical identification of the EGP model parameters requires an excessive amount of experiments, each using different samples. The identification can, however, be troublesome due to the history dependency that can vary for each sample. The work of Neggers et al. [8] shows with his proposed method that the material properties can be identified from a single virtual experiment providing a reasonably accurate initial guess of 10%. With the proposed regularization method the robustness to the initial guess has been increased to 30%.

The altered IDIC routine, including Tikhonov regularization and an updated residual scheme applied to experimentally obtained images, shows convergence and is robust towards runaway parameters. The convergence, however, is limited, the residual of both the image and force reach a steady-state. In this steady state, the parameters still update freely, which could indicate that this minimization problem is non-unique. The EGP model with 10 parameters can only describe the intrinsic post-yield behavior; therefore the viscoelastic regime cannot be identified accurately with 10 parameters. To add 16 more parameters to the optimization routine would most likely not be possible as the sensitivity of the G_1 parameter is then split into 17 separate parts and thus be significantly lower, increasing the chance of non-uniqueness and increasing the computational time. To accurately describe the viscoelasticity, the best approach will most likely be to obtain the viscoelastic spectrum from the traditional material model fitting, after which these parameters can be locked from the IDIC routine, however, this abandons the idea of the single inverse parameter identification method.

No actual minimum in residual is found with the single-mode model. With experiments optimized to gain more sensitivity for the pressure dependent parameter μ and zero shear viscosity η_0 it might be possible to gain a more accurate description of these parameters. In recent work of Jan Neggers [17], a multi-experiment approach is proposed. With this approach, multiple experiments can be assessed simultaneously; this offers a potential for a combination of experiments to optimize specific parameter sensitivities.

6 Conclusions

The IDIC routine has been altered to stabilize the 10 parameters used in the EGP model. Tikhonov regularization is added to initially lower the weight of relatively insensitive parameters and releases gradually. In previous work, an offset of the initial guess with a maximum of 10% was proven to converge, with the regularization method this has been increased to 30% for virtual experiments, in which the model perfectly describes the material behavior.

An updated scheme for the image residual is added to decrease the requirement of brightness conservation of the images. Both alterations to the IDIC routine are necessary to avoid runaway parameters and instabilities.

From the IDIC procedure with the EGP model on experimentally obtained images can be concluded that the routine works, and the resulting parameter combination describes material behavior more accurately than the values from literature. Residual fields on both force and displacement decrease with the iterations. These residuals, however, reach a steady-state, in which the parameters still updating. The most feasible explanation for this behavior is that the single-mode EGP model approximation with 10 parameters has multiple solutions preserving the same error in both displacement and force.

References

[1] L.E. Govaert, P.H.M. Timmermans, and W.A.M. Brekelmans. *The Influence of Intrinsic Strain Softening on in Polycarbonate: Modeling and Experimental Validation.* Journal of Engineering Materials and Technology, 2000.

[2] L.C.E. Struik. *The mechanical and physical ageing of semicrystalline polymers.* Plastics and Rubber Research Istitute TNO, 1987.

[3] F.R. Schwarzl and F. Zahradnik. *The time temperature position of the glass-rubber transition of amorphous polymers and the free volume.* Rheologica Acta, 1980. Volume 19, issure 2, pp 137-152.

[4] L.C.A. van Breemen, E.T.J. Klompen, and H.E.H Meijer. *Extending the EGP constitutive model for polymer glasses to multiple relaxation times.* Journal of the Mechanics and Physics of Solids, 2011.

[5] T.A. Tervoort, E.T.J. Klompen, and L.E. Govaert. *A multi-mode approach to finite, three-dimensional, nonlinear viscoelastic behavior of polymer glasses.* The Society of Rheology, Inc. J. Rheol., 1996.

[6] T.A. Tervoort, R.J.M. Smit, W.A.M. Brekelmans, and L.E. Govaert. *A Constitutive Equation for the Elasto-Viscoplastic Deformation of Glassy Polymers.* Mechanics of Time-Dependent Materials, 1996.

[7] S. Avril, M. Bonnet, A.S. Bretelle, M. Grédiac, F Hild, P. Ienny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, and F. Pierron. *Overview of Idenfitication Methods of Mechanical Parameters Based on Full-field Measurements.* Experimental Mechanics, 2008. 48, 381-402.

[8] J. Neggers, J.P.M. Hoefnagels, and M.G.D. Geers. *Time-resolved integrated digital image correlation.* International journal for numerical methods in engineering, 2015.

[9] C. Fennema and W Thompson. *Velocity determination in scenes containing several moving objects.* Computational Graphics and Image Processing, 1979.

[10] S. Cooreman, D. Lecompte, H. Sol, J. Vantomme, and Debruyne D. *Elasto-plastic material parameter identification by inverse methods: Calculation of the sensitivity matrix.* International Journal of Solids and Structures, 2007.

[11] B.D. Lucas and T. Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision.* Intl Joint Conf on Artificial Intelligence, 1981.

[12] T.A. Tervoort, E.T.J. Klompen, and L.E. Govaert. *A multi-mode approach to finite, three-dimensional, nonlinear viscoelastic behavior of polymer glasses.* Journal of Rheology, 1996.

[13] A.N Tikhonov, A.V. Goncharsky, V.V. Stepanov, and A.G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems.* Springer-Science+Business Media, BV, 1990.

[14] H. Ahmadian, J.E. Mottershead, and M.I. Friswell. *Regularisation methods for finite element model updating.* Mechanical Systems and Signal Processing, 1998.

[15] Stereo discovery.v20. `https://applications.zeiss.com/C125792900358A3F/0/CFE40039745C6C41C125790600481655/$FILE/46-0128_e.pdf`. Accessed:11-01-2019.

[16] A.P. Ruybalid, J.P.M. Hoefnagels, O. van der Sluis, M.P.F.H.L. van Maris, and Geers M.G.D. *Mixed-mode cohesive zone parameters from integrated digital image correlation on micropraphs only.* International Journal of Solids and Structures, 2019.

[17] J. Neggers, F. Mathieu, F. Hild, and F. Roux. *Simultaneous Full-Field Multi-Experiment Identification.* Mechanics of Materials, 2019. Accepted Manuscript.

TU/e Technische Universiteit
Eindhoven
University of Technology

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

19/09/2019

Name

J.A. Joustra

ID-number

0851110

Signature

Submit the signed declaration to the student administration of your department.

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

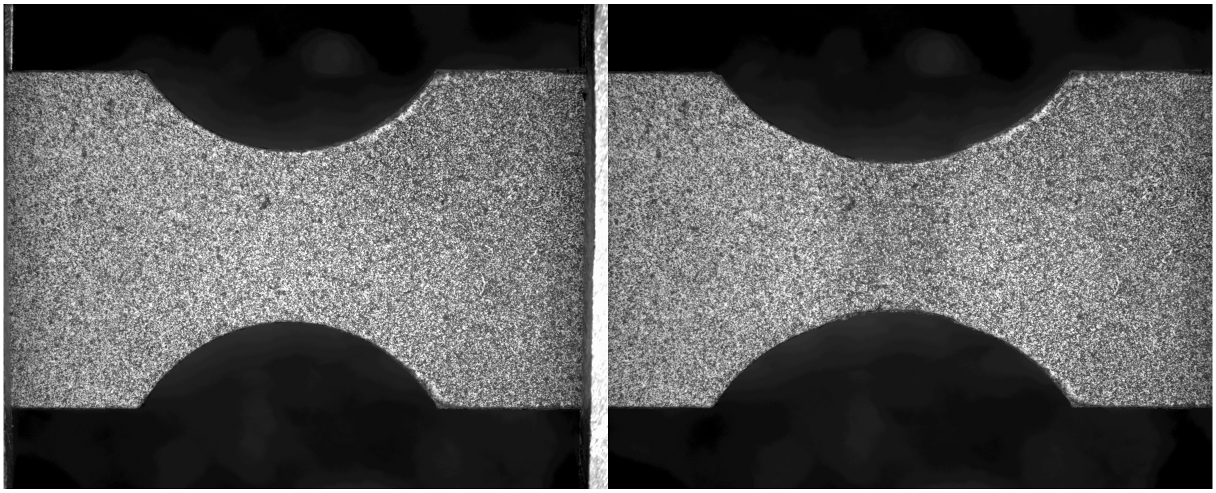
January 15 2016

A Experiments

At first, experiments need to be conducted. This can be done in various ways. A tensile bar is mounted in a tensile tester and placed under a microscope. The surface area of this tensile bar can have any arbitrary shape; the thickness, however, needs to be known precisely. This experiment is conducted with a stereo-microscope to utilize a large depth of field. The microscope is equipped with a camera to acquire images of the deformed and undeformed states. Secondary force data is obtained with a load cell in the tensile tester. All images are saved in a jpg format, and the force is stored in a DDS file. This section will cover various experimental challenges and methods.

A.1 Speckle pattern

A pattern for DIC cannot be applied directly to the PC sample since it is transparent. Firstly, the sample needs to be opaque. Since a polycarbonate sample necks, the light will refract, which affects the light intensity map, and the pattern will not hold. This is done by a white chalk powder spray (”Standard-Chek Eindringprüfsystem Medium Nr.3 Entwickler weiß”). With a graphite powder, the speckle pattern is applied.
With the current chalk powder and graphite powder combination, the speckle pattern of the sample does not hold perfectly, no cracks or irregularities occur; however, the pattern in and around the neck darkens as can be seen from Figure A.1.



(a) **(b)**

Figure A.1: ***(a)*** *Undeformed reference image.* ***(b)*** *Plasticly deformed and darkened at the neck.*

Since the deformed image is continuously back-transformed to the reference image in the IDIC algorithm, the error in the residual will grow throughout the images. The solution described in Equation 2.28 limits this effect to a single frame.

A.2 Experimental challenges

Experimental data is subjected to noise and other complications; the effects and severity will be explained and analyzed in this section.

A.2.1 Out of plane deformation

A PC sample in tension beyond yield will neck and will show a contraction of about 25%; this will result in a change of about 0.125 mm relative to the camera. The stereo microscope used has a large depth of field, in which the change in focus is negligible.

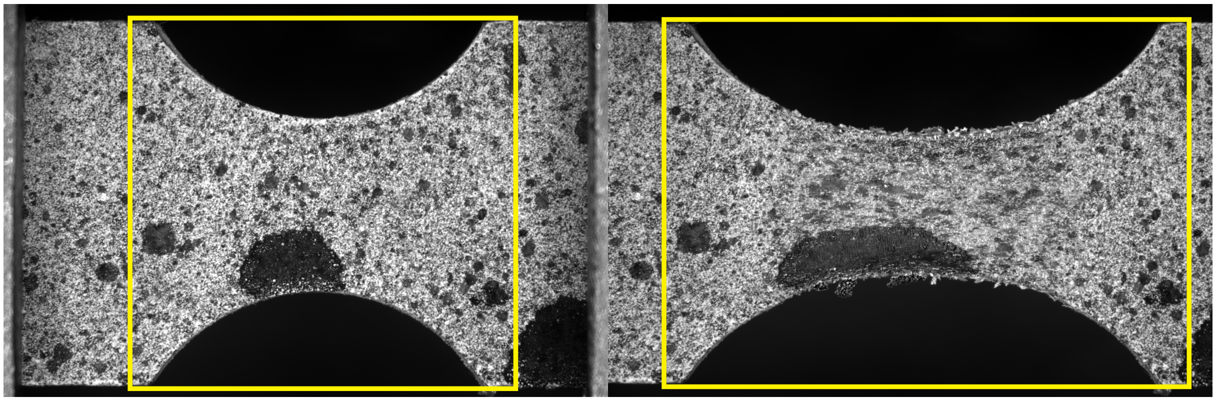
As for the deformation fields, the change in depth cannot be used in the IDIC routine without information of the thickness. This information could, in theory, be gathered with the sensofar profilometry. However, in this work, the deformation field is limited to 2D. Since the Python script only uses the x and y plane of the FEM, we are in both the experiment and simulation looking in a 2D space, which should not give any complications as it didn't with the virtual experiments, which are constrained in the same way as the experiments.

B Step by Step Experimental IDIC Routine

After the images and force data of the experiments are acquired, a couple of steps are needed to be able to run the IDIC routine. For experiments, this Appendix will give a step by step explanation, including Matlab codes used.

B.1 Convert images to mat file

Firstly, the images of the experiment need to be compatible with the IDIC routine. Firstly it is advised to crop the images to take up less space. Of importance is that the whole sample region of interest is visible, undeformed, and deformed. An example is shown in Figure B.1.



(a)

(b)

Figure B.1: *(a) Undeformed image with region of interest highlighted (b) Deformed image with region of interest highlighted*

A code of Andre Ruybalid is used to create a .mat file containing the images in a manner so the IDIC routine can use it. Additional information can be added by adding a .dds file from the tensile tester, which contains force and time data. New functionality is added to add a blur to the images, which might be in favor of convergence as the interpolation of the IDIC routine can cause problems when not applied, and gradients are large. That the pixel size needs to be spot on, when this is not the case, the output of the IDIC routine will be wrong if it converges at all. The origin of the image is now set at the (0,0) coordinate as that works together with the next section. Furthermore, the time-offset of the DDS file has to be determined as the image acquisition software, and the force logger does not operate simultaneously. Further options are described in notes in the code below.

B.2 Get FEM model from images

Instead of using the known geometry of the specimen, it is now possible to create a mesh directly from the image instead. The question: '*Graphical nodal selection (y/n)?*' will show up, this gives a choice to manually add data points to the image by using the data cursor and holding shift while clicking, a contour can be acquired. This contour is then loaded in a .proc file for Marc Mentat. Lines will be added between the data points automatically in the same order as the manually selected data points. If the data points are known of beforehand, this can also be loaded in cursor_info. Figure B.2 shows the result of the Matlab code after manually adding data points.

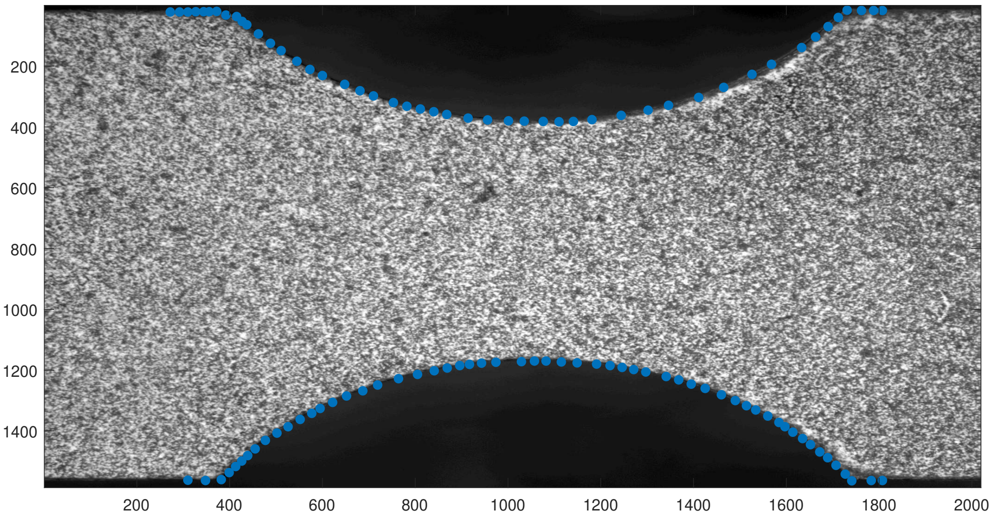


Figure B.2: *The FEM contour manually applied to the image*

A mesh can then be generated manually in 3d. The sample thickness is assumed constant in the sample. Figure B.3 shows a 2d mesh acquired from the experimental images; this mesh is extruded to match the sample thickness. The Matlab code is shown below.

B.3 Apply Boundary conditions from DIC

The boundary conditions on the left and right side of the sample do not match the experimental input due to slip and the mechanical deformation of the sample. Therefore a global DIC algorithm is used to determine these boundary conditions[16]. The region of interest is chosen to match the edge of the mesh on both sides of the sample. Since the strain in these parts is expected to be in the visco-elastic regime, a second-order b-spline fit suffices, which is subdivided in a 3x15 raster. The residual of the DIC code is shown in Figure B.4. The residual is determined from the back-transformed image to the reference; the pattern is darkened slightly; therefore, the residual is positive on most of the area.
The deformation determined from the DIC is extracted to an array for each pixel on the left and right side of the mesh and interpolated for the nodal positions. The deformation through the thickness of the sample in x and y direction is considered constant.
For the integration of the DIC and FEM boundary conditions, scripts of Andre Ruybalid is used: ubuild.m; isnear.m and nodecoord.m. It the chosen left and right side of the FEM, chosen in Section B.2, are in the DIC results, a prompt will show up with the question: '*Graphical nodal selection (y/n)?*', if y (yes) is chosen, a similar screen to Section B.2 shows up. Again all nodes chosen and saved to cursor_info will be selected to add boundary conditions. If there are a lot of nodes to be described,

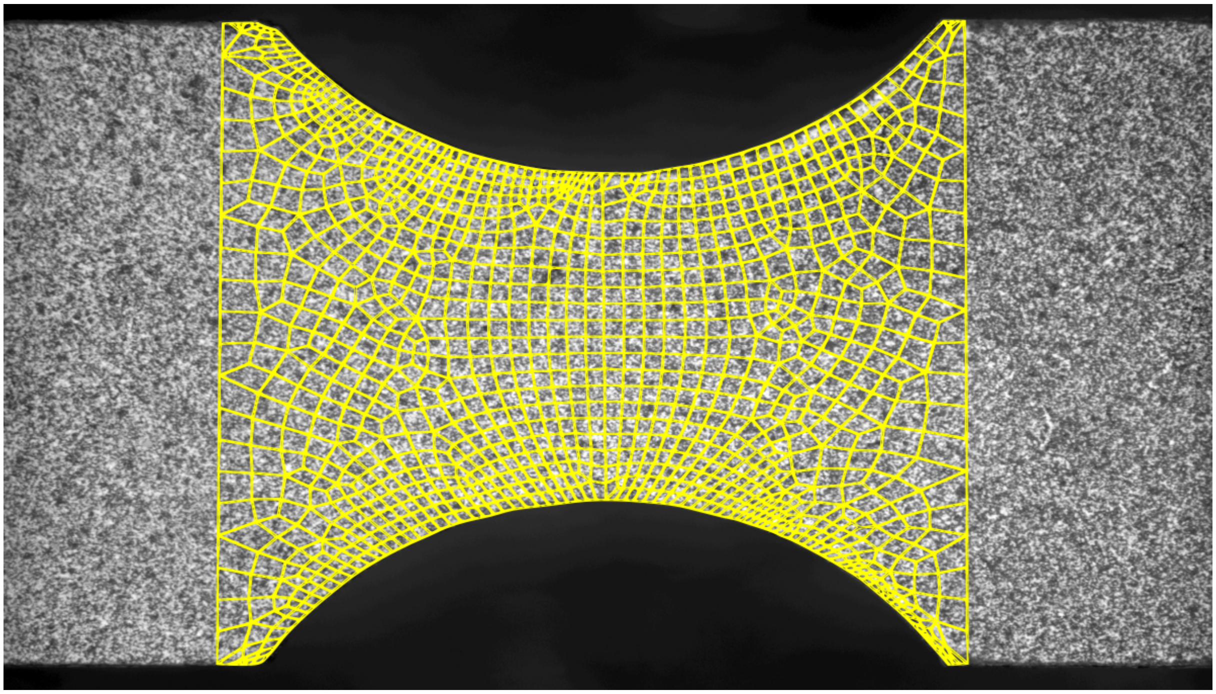


Figure B.3: *A mesh overlay on an acquired image*

it could be hard to converge, which will be explained in the next section, the Matlab scripts are shown below.

B.3.1 Boundary conditions: Numerical complications

Boundary conditions on all nodes on the left- and right edge of the tensile bar adds constraints to the stiffness matrix, which can cause the system to be unable to converge. The nodes hare linked such that there are fewer boundary conditions. By using a quadratic shell element on the edges of the tensile bar and using an insert in Marc Mentat, the nodes along the edge of the sample follow the shell element. In Figure B.5 is the evolution of deformation over time, along with the edge nodes shown.
From this Figure can be concluded that a quadratic profile as boundary condition suffices.

B.4 FEM Settings

The EGP model is highly nonlinear and complex, sometimes, especially when updating the parameters constantly, the model seizes to converge, which yields the complete IDIC routine to stop working. Therefore it is of importance that the FEM and settings are as stable as possible. An overview of the settings used will be given in this Appendix. The Tables & Coord. Syst. and Boundary conditions will be skipped as these are discussed in section B.3.

B.4.1 Geometry and mesh

The stability is dependent on the mesh; therefore, a stable mesh is required. Especially when the nodes placed manually, there could be outliers which could make the auto mesh function to make distorted elements or elements with bad aspect ratio. This should be minimized as much as possible. If the mesh is unstable, another possibility is to fit a polynomial through the top and bottom half-circle of the dog bone shape. The shape of the elements should be 8 node brick elements (hex8). Tetrahedral should be

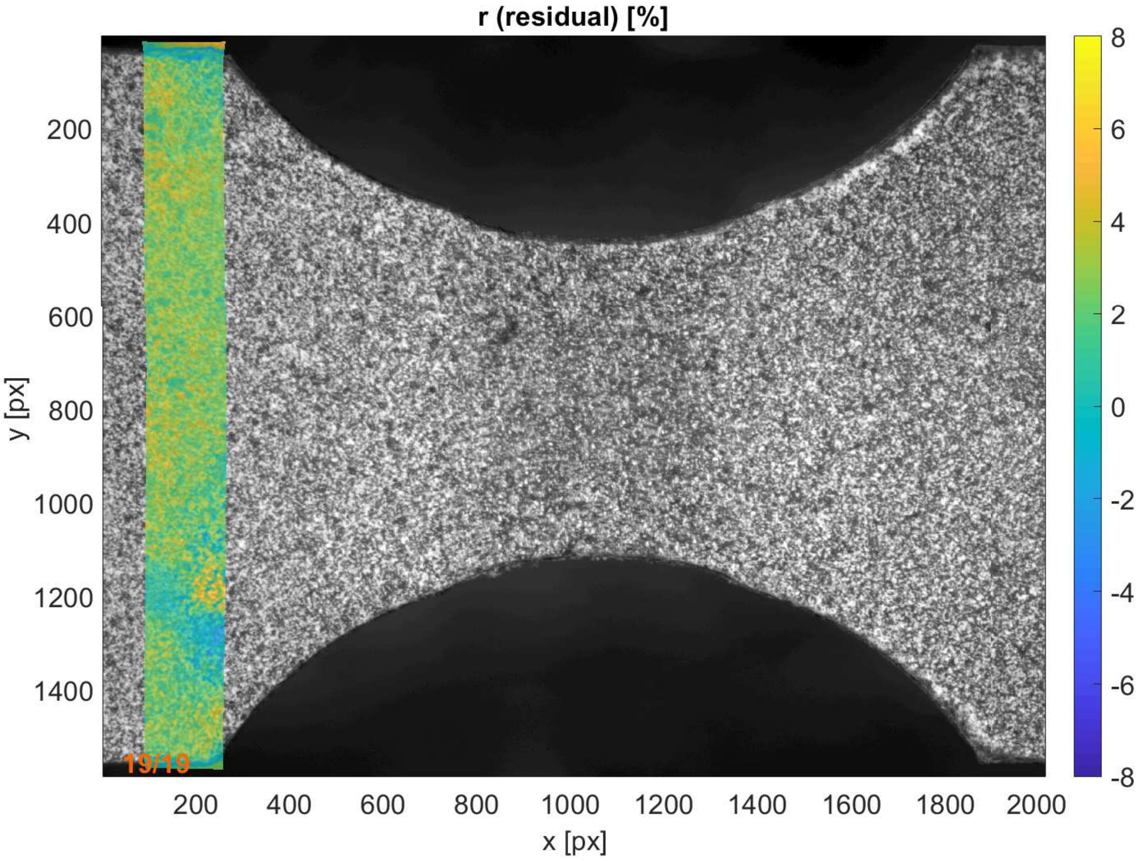


Figure B.4: *Image of experiment with overlay of the residual after convergence of the DIC algorithm*

avoided as these are inherently stiff by nature.

B.4.2 Geometric and Material Properties

As discussed in Section B.3.1, there will be two types of elements in the FE model, 3D solid (mech_three_solid) and 3D shell (mech_three_shell) elements, these need to be assigned to the respective elements to make the model work. The material models used are the user subroutine Hypela2 for the EGP model and a compliant linear elastic Isotropic material model for the shell elements. It is of importance that a low stiffness is used as stress will otherwise be transferred through the sample orthogonal to the loading direction.

B.4.3 Nodal ties

The nodes on both sides of the sample should follow the shape of the shell element, therefore links are made:
Links → Inserts → New → Host Entities - elements (select shell element) → Embedded Entities - Nodes (select all corresponding nodes) → Tolerance (0.0001)

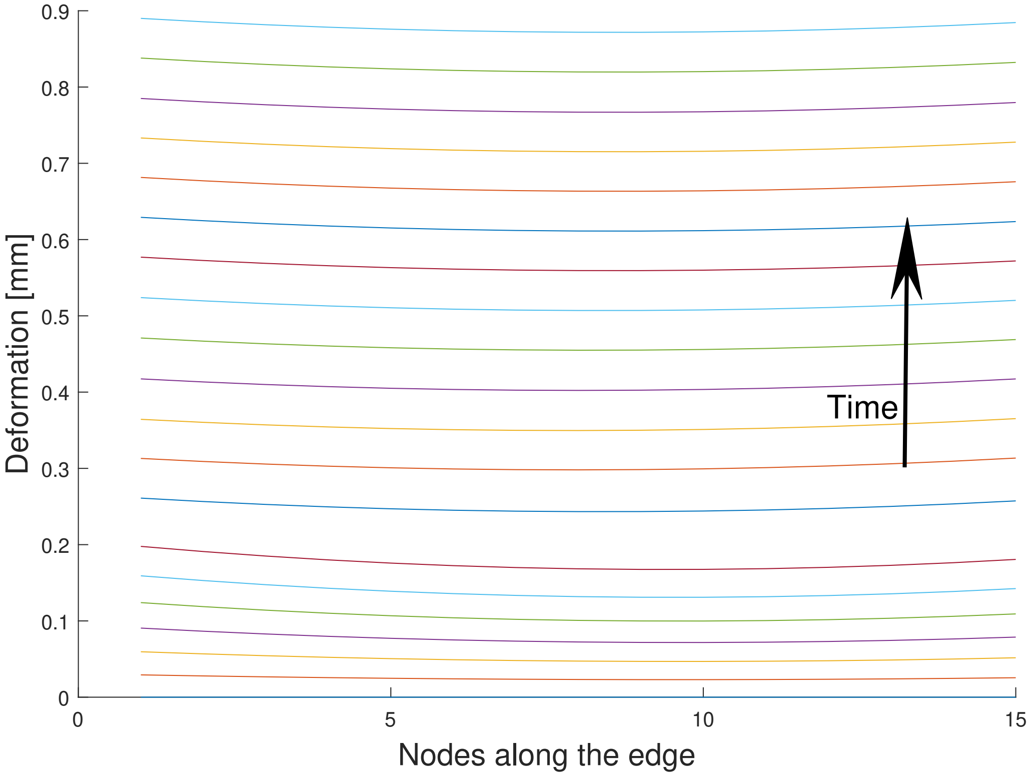


Figure B.5: *Evolution of deformation of nodes along the edge of the sample*

B.4.4 Loadcases

A quasi-static load case is performed in which the load case time is equal to the time of the experiment; the number of steps should equal the number of images added to the matfile in immat.m. In the iterative procedure, full Newton-Raphson can be used. Contribution of initial stress to stiffness can be Full or on Tensile stress, in which the latter is found more stable. The maximum number of Recycles is free to be chosen; in this work, a maximum of 50 iterations is allowed. Cutbacks are capable of cutting time back to decrease the deformation step, which can help an iteration to converge.
An important setting is found in *Convergence Testing*, the default setting of Marc Mentat is set on a convergence criterion of 10% on the force residual. In these simulations, a value of 0.1% is used, which means that the residual force can be up to 0.3 Newton off.

B.4.5 Jobs

In Jobs a couple of parameters need to be changed for the EGP model to work:

Job Parameters → User Data Memory Allocation - 1000000 → # state variables - 274 for PC → OK
Analysis options → Nonlinear Procedure - Large strain → Advanced Options → Large strain - Updated Lagrange - □ Allow Switch to Total Lagrange - OK - OK

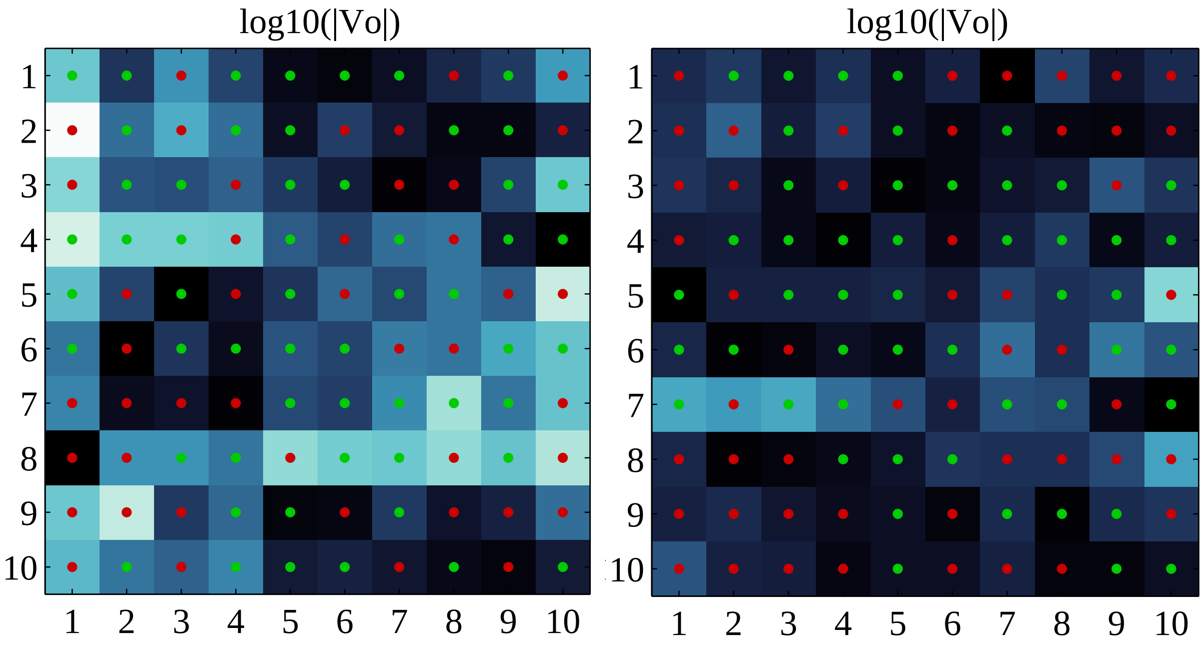
To run and test the FEM, press run, add the user subroutine file egpm.f and submit job. The model should be saved as a DAT file executable in Marc, to be used in the IDIC routine. This is saved in the right way by pressing *Advanced Job Submission* → *Write Input File.* The file should be renamed idic_job.dat and placed in the qsub map of the IDIC routine.

B.5 Run IDIC routine

A couple of options can be given in the main idic.m file. Table B.1 gives an overview of all options available in idic.m.

B.5.1 Perturbation factor

A 3D model of a dogbone shape tensile bar can converge with a relative force tolerance of 0.001, which means that the FEM is on average 0.1% off with the numerical estimate. With lower tolerances, the dogbone shape does not converge. Since the displacement field is not perfectly correct, the sensitivity of the parameters cannot be determined correctly if this displacement field is in the order of the numerical error. The comparison of a relatively high to a too low perturbation factor can be seen in Figure B.6.



(a)

(b)

Figure B.6: ***(a)*** *A sensitivity eigenvalue value decomposition with a perturbation factor larger than the numerical error,* ***(b)*** *The same FEM simulation with a perturbation factor lower than the numerical error.*

As can be seen from the above Figure, the sensitivity map of a simulation with a too low perturbation factor has same eigenvectors, which cannot converge any further and is more likely to be subjected to runaway variables. This can, in turn, cause infeasible variable combinations to occur. On the contrary, a too high perturbation factor can cause interpolation instead of extrapolation. This can cause a variable to surpass the optimum value and flip back and forth without a chance of convergence for nonlinear cases like this.

From this can be concluded that the numerical precision of the EGP FEM simulation could lead to insensitivities or instabilities of the IDIC script. Therefore it is not expected to acquire the same accuracy for all parameters.

Option name	**Description**
comp	Sets the operating system; Linux/windows/cluster; EGP only works on Linux/cluster
cores	amount of cores used in the calculation; cluster automatically sends jobs in queuing system
pbsoptions	if you want to run the marc mentat model in parallel, it is possible to add more ppn (cores) to it
run_marc/run_python	Directories of marc mentat and python
lambda(:,1)	The initial guess of the parameters
lambdascale(:,1)	Scaling of these parameters (i.e. 3 = lambda * 10^1)
lambdastr(:,1)	Name of the material parameter
lambda_lock	vector of parameters that are not used in the correlation (i.e. [2 5 10] means parameters 2,5 and 10 are locked)
soft_lock	vector of parameters that are coupled (i.e. [0 1 0 5 0 0 0 0 0 0] means that parameter 2 is the same as 1 and 4 equal to 5.)
libpath	path of idic library (standard set to [pwd '/lib_idic']
resumefile	resume from previous iterations (set to check_latest_resumefile)
perturbe	perturbation factor of lambda at the start of the routine
minperturbe	minimum perturbation factor of lambda
convcrit	convergence criterion norm|b|<convcrit
maxiter	maximum number of iterations
LaterLock	option to lock the parameter if b is small enough (now set to convcrit*0.1)
Tigonovstart	starting position of Tikhonov regularization ∈ 1∽Ndof with lower numbers is heavier dampening, if 0 it is turned off
waittime	Stabilityvariable, this is the time allowed between jobs to be completed in seconds
reglimit	limit the maximum update to lambda*reglimit
roi	the regeon of interest in mm (important to have this inside of the marc mentat model [minx maxx miny maxy]
pausetime	mainly for cluster, it is the time between checks if the jobs are done
normalize Mo and MF	true if force data is taken into account
alpha	value ∈ 0∽1 to determine severity of force data
usegradgt	option to use the gradient of the backtransformed image as well as the gradient of the initial image
use_gt	This is the option to use the updated idic routine, the difference between two images are taken instead of all images back to the reference frame
sym_x, sym_y	for virtual experiments, if there is a symmetry line
basename	The basename for the saved files
cmap	colormap for images that involve a residual field
Nplot_y Nplot_t	number of plots for respectively the sensitivity maps and residual field
storepdf storepng	store figures true or false
storemat	a resumeskipfile, this takes a long time to save and can take a lot of space, recommended to not have this on for large models
logfile stsfile intfile	filenames for logging and status

Table B.1: *Options in the main idic.m file*

C Sensitivity maps for different experiments

Two holes

The Tikhonov regularization is turned on in all these simulations; therefore the combined sensitivity and eigenvalue matrices have a single diagonal value, this should be ignored.

For the double holed sample, similar results to the single hole are found. However slightly worse sensitivity is found for r_0 and r_1. Slightly better sensitivity is seen for G_1, the shear modulus here is seen more directly as the holes are slightly skew, a shear contribution is clearly visible.

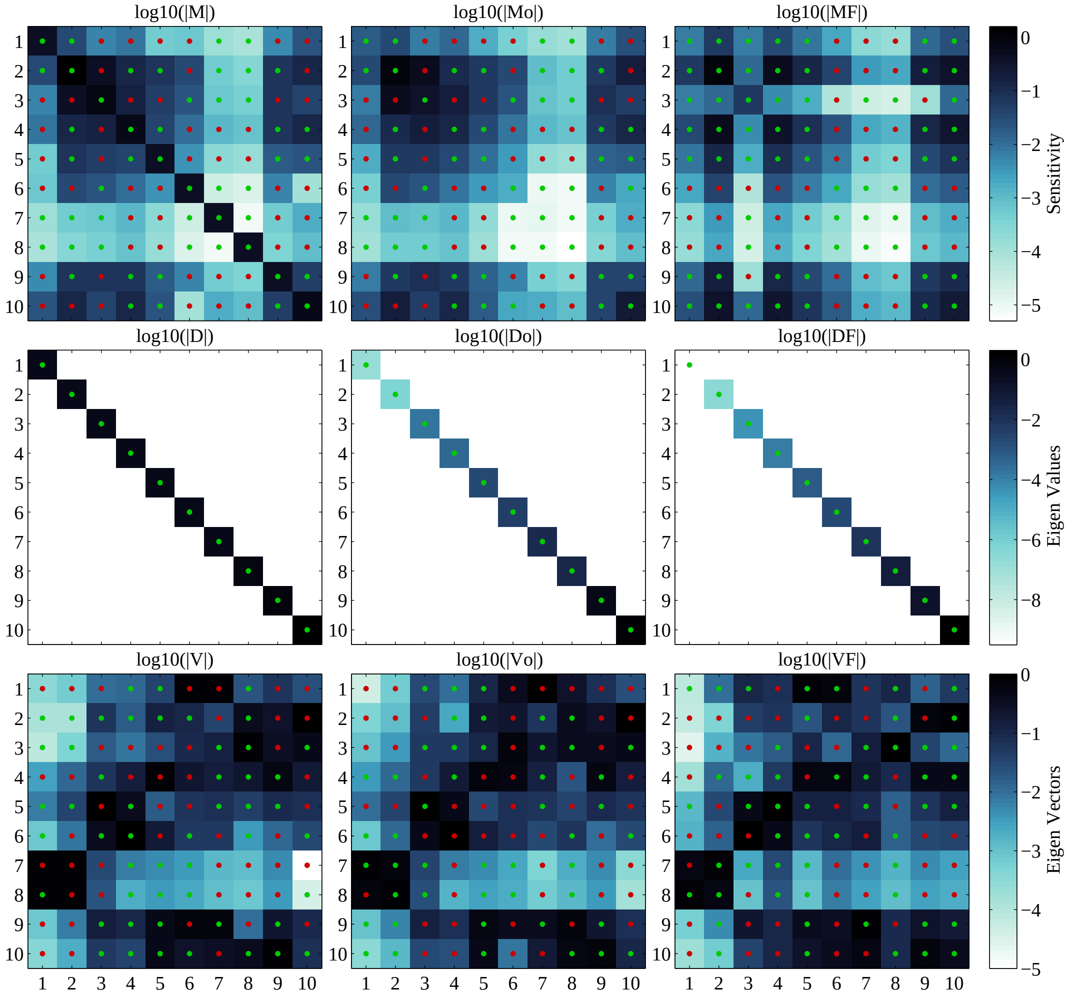


Figure C.1: *Sensitivity field of a tensile bar with two holes*

Smaller hole

A smaller hole than in the first case is used, 1mm compared to the 3mm before. The results are shown below in Figure C.2. Only a slight improvement has been found in r_0. With the initiated IDIC routine with 10% offset, no convergence is reached.

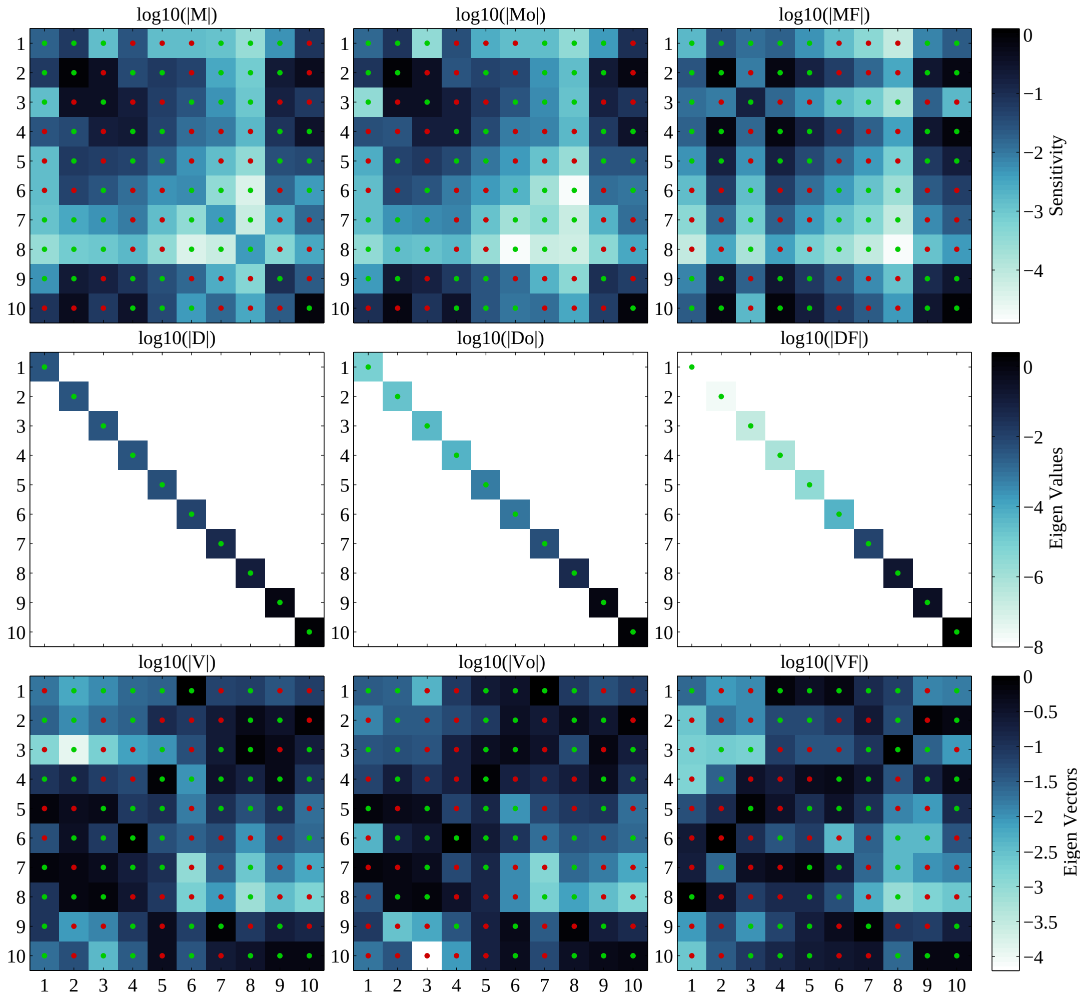


Figure C.2: *Sensitivity field of a tensile bar with a single hole, smaller than the previous*

Clamped sample loaded in wide direction

The EGP model has been shown to work to great extend on a fully clamped sample that is loaded in the wide direction, as shown in Figure C.4. The sensitivity field of the clamped sample is shown below. As with the samples with the holes, r_0 and r_1 have a negligible sensitivity. Additionally, G_1 is lower than with the other samples. The IDIC routine did not converge with a 10% offset; therefore, this geometry is not used.

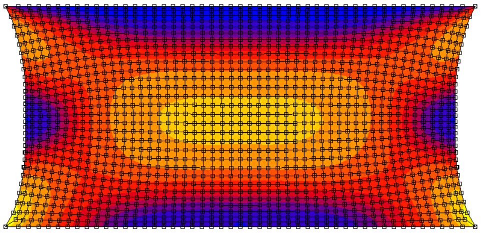


Figure C.3: *Strain field ϵ_{yy} of a tensile bar clamped and loaded in the wide direction*

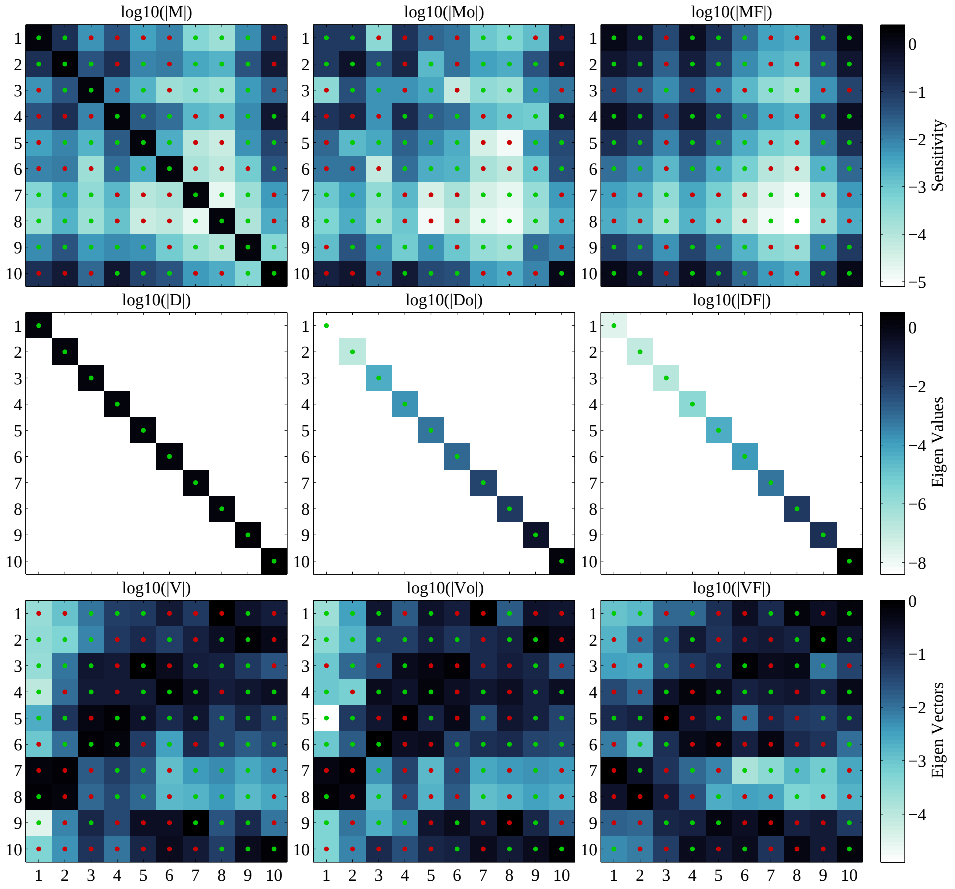


Figure C.4: *Sensitivity field of a tensile bar clamped and loaded in the wide direction*

D Alterations to the IDIC routine

For compatibility and routine alterations, the IDIC script needs to be altered. The base code is copied from J.G. (Janko) Stellaard (Version 2.01 - 01-12-2016). This section will give more detail on what changes are made to the scripts and what functionalities are added.

D.1 Compatibility

As the EGP model is not a standard material model in Marc Mentat, but a user subroutine, a few alterations are needed to make the marc Mentat model work. After the FEM has finished, the code will remain the same.

In the main IDIC.m file, the material parameters are updated to a vector of 10 long. The following order needs to be maintained to work with the upcoming codes.

```
% ================================================
% Initial Guess
% ================================================

% initial guess               % scaling factor          % descriptive string
lambda(1,1) = 3.750;          lambdascale(1,1)  = 3;    lambdastr{1,1}  = 'Gk';
lambda(2,1) = 2.6;            lambdascale(2,1)  = 1;    lambdastr{2,1}  = 'Hard';
lambda(3,1) = 6.34;           lambdascale(3,1)  = 2;    lambdastr{3,1}  = 'G(1)';
lambda(4,1) = 7;              lambdascale(4,1)  = -1;   lambdastr{4,1}  = 'Gt0a';
lambda(5,1) = log(1.47e11);   lambdascale(5,1)  = 0;    lambdastr{5,1}  = 'Gh0(1)';
lambda(6,1) = 8;              lambdascale(6,1)  = -2;   lambdastr{6,1}  = 'Gm';
lambda(7,1) = 9.65;           lambdascale(7,1)  = -1;   lambdastr{7,1}  = 'r0';
lambda(8,1) = 5;              lambdascale(8,1)  = 1;    lambdastr{8,1}  = 'r1';
lambda(9,1) = -3;             lambdascale(9,1)  = 0;    lambdastr{9,1}  = 'r2';
lambda(10,1) = 2.7;           lambdascale(10,1) = 1;    lambdastr{10,1} = 'S0';
```

Note that λ_5 or η_0 is on a logarithmic scale, this is the case since the sensitivity of a perturbation of for instance 10^{-1} is low and since the value could easily vary in the range of 10^8 to 10^{14}, this gives the model more freedom to go towards the respective value.

The EGP model needs to be copied to the working directory of the cluster node, in the main map, the folder containing the EGP model is added and copied to each job submission folder.

```
%% Fem_EGP_copy
% Jesse Joustra
% This function will copy the egp model to the right directory
function Fem_EGP_copy(jobdir)

    copyfile([cd '/EGP/'],[cd '/' jobdir]);
```

Also, the material parameters need to be updated and placed in this folder. A matpar file (extensionless) needs to be in the main IDIC map and is updated with the following code.

Listing 1: *fem_EGP_datupdate.m*

```
%% FEM UPDATE FOR EGP MODEL
function dat = fem_EGP_datupdate(lambda,datfile,jobdir)

lambda(5,1) = exp(lambda(5,1));

filename = 'matpar';
fid = fopen(filename,'r');
```

```
D = textscan(fid,'%s','Delimiter','\n','whitespace','');

Matline1 = 26;

C = D;
C{1,1}(Matline1+0) =  {['      parameter (         Hard    = ' num2str(lambda(2,1)) 'd0, ']};
C{1,1}(Matline1+1) =  {['     +                    S0      = ' num2str(lambda(10,1)) 'd0, ']};
C{1,1}(Matline1+2) =  {['     +                    r0      = ' num2str(lambda(7,1)) 'd0, ']};
C{1,1}(Matline1+3) =  {['     +                    r1      = ' num2str(lambda(8,1)) 'd0, ']};
C{1,1}(Matline1+4) =  {['     +                    r2      = ' num2str(lambda(9,1)) 'd0, ']};
C{1,1}(Matline1+5) =  {['     +                    Gt0a    = ' num2str(lambda(4,1)) 'd0, ']};
C{1,1}(Matline1+6) =  {['     +                    A0ab    = 0.000000d0, ']};
C{1,1}(Matline1+7) =  {['     +                    Gt0ab   = 0.000000d0, ']};
C{1,1}(Matline1+8) =  {['     +                    Gm      = ' num2str(lambda(6,1)) 'd0, ']};
C{1,1}(Matline1+9) =  {['     +                    Gk      = ' num2str(lambda(1,1)) 'd0, ']};
C{1,1}(Matline1+13) =  {['      G(1) = ' num2str(lambda(3,1)) 'd+00 ']};
C{1,1}(Matline1+15) =  {['      Gh0(1) = ' num2str(lambda(5,1)) 'd+00 ']};

fid = fopen([cd '/' jobdir '/' filename],'w+t');

fprintf(fid,'%s\n',C{1:1}{:});
fclose(fid);

dat = datfile;
```

To run the material model from the cluster it is important that the user subroutine is called from the .pbs file. If the job is saved in marc mentat using *Jobs* → *Run* → *Advanced Job Submission* → *Write Input File*, the model should work.

```
% Small alteration to the runstring in .PBS for egpm.f
\$marcbin -jid "\$datfile" -q f -u egpm.f -v n -nthread 1 -nps \$ncpus
```

D.2 Updated IDIC

basisfunctions (updated IDIC)
Lmatrix (updated idic)
sol_grad

D.3 Tikhonov regularization

To obtain the α, as stated in Equation 2.33, an eigenvalue decomposition is first made. The working principle is to use the eigenvalues as α while gradually decreasing the size, going down stepwise as the update of lambda decreases. With 10 parameters, there are 10 eigenvectors and eigenvalues, in the main idic.m, the starting position of Tikhonov can be chosen to be 1 Ndof; if 0 is chosen, the regularization is turned off. If the average update of λ is smaller than 4%, the Tikhonov regularization will go down one step in the next iteration. The code is shown below and applied to sol_solvedlambda.m.

```
% TIGONOV regularizATION : M = M+aI --> Dampens the insensitive parameters
if Tigonov > 0

        if Tigonov < NIfree+1
            [V,  D ] = eig(M);
            Mtemp = M;
            Dtemp = D(Ifree,Ifree);
            Mtemp(Ifree,Ifree) = Mtemp(Ifree,Ifree) + eye(length(Ifree)) ...
            *D(NIfree-Tigonov+1,NIfree-Tigonov+1);
            dlambda_temp = Mtemp\b;
            Percentage_Change = 100*dlambda_temp./lambda;

            if mean(abs(Percentage_Change)) < 2
                Tigonov = Tigonov +1;
                if Tigonov > NIfree
                     Tigonov = 0;
                end
            end
            if Tigonov <= NIfree
                if Tigonov ~= 0
                regfactor = Dtemp(NIfree-Tigonov+1,NIfree-Tigonov+1);
                M(Ifree,Ifree) = M(Ifree,Ifree) + eye(NIfree)*regfactor;
                Mb.M = M;
                itdata.Mb = Mb;
                printlog(logfile,'Tigonov regularization; Alpha: %10.3e; ...
                Position on eigenvalue matrix: %d of maximum of %d',regfactor, ...
                 Tigonov, NIfree);
                end
            else
                printlog(logfile,'Tigonov regularization has done its job, ...
                 now turned off');
            end

        end

end

inp.Tigonov = Tigonov;
```

D.4 Minor alterations to increase the stability of the model

Computing on the cluster can be a time saver and a time consumer. Without the cluster, the code would run for many months, which is not an option for the limited timespan. Parallel computing is used on the supercomputer of the Technical University Of Eindhoven to reduce runtime. This cuts the computation time with an order 10. However, this is a substantial job, easily surpassing 10-gigabyte file, in which the cluster sometimes struggles to handle. (Mostly due to network speed) Therefore some minor alterations to the routine are made to automatically restart jobs that are stuck or completely restart an iteration.

If a job gets stuck, it will be restarted in fem_checkstat.m:

```
                if all(jstatus(k,:) < 0)
                        printlog(logfile,'!!! job %d failed !!! ',k);
                        printlog(logfile,'This single job restarting in 2 minutes!');
                        pause(120)
                        fem_single_submit_cluster(k,itdata,inp);
                        jstatus(k,:) = [0 0 0 0];
                end
```

It might also be the case that data is lost; it is hard to check whether this is the case. If this is the case, the routine will still be up, but nothing will happen. If nothing changes for more than 1,5 hour, the IDIC routine will be restarted in fem_checkstat.m:

```
            if jstat(4) > ncjobs
                if exist('timerstart') == 0
                    timerstart = tic;
                    tic;
                    printlog(logfile,'First job done --> Starting timer');
                end
                timereval = toc;
                if timereval > waittime
                    printlog(logfile, 'Script took more than %d seconds for all simulations to complete',
                    printlog(logfile, 'Dont worry, script is restarting!')
                    run('idic.m')
                end
            end
```

Stability (alterations to restart the script)
sol_refdisplacement (For large files to check if the job is truly done)

E MATLAB codes

E.1 Image to mat file

Listing 2: *immat.m*

```
% store images in a .mat-file as to be used for the IDIC or FEMU routine.
%
% clear the image files
% clear m n x y f inc F
%
% Make sure the pixel size (pxs) is defined correctly under OPTIONS
%
% Select the origin in image, or another known coordinate, using mouse input.
%
% Andre Ruybalid 2018 (c)
% ####################################################################

clear all; clc;

% ==========
% OPTIONS
% ==========
ddsdir = [pwd];  % directory containing .dds file
ddsfile = '02_07_dogbone2';
imdir = [pwd filesep '02_07_Dogbone_cropped'];     % directory containing images
reftime = 1;                   % determine the time increment at which the
                               % force is first 0 (or nearest to 0)
timesteps = [1:1:50];          % select the time increments of the 'deformed'
                               % image and force data
%pxs = [0.01608 0.01608];       % pixel size of the images
pxs = [0.0061 0.0061];
crp = 0;                       % [ymin ymax xmin xmax] for trimming image,
                               % crop = 0 --> do not crop
blur = 1;                      % add image blurring (true/false)
windowsize = 4;                % Windowsize of the blur
grad = 0;                      % gradient image (true/false)
store = 1;                     % save a .mat file (true/false)
matname = '02_07_Dogbone';     % set the name of the to-be-saved .mat file

% Set coords of origin in marc mentat model
% If empty -> Visual popup will come up

x0 = 0;
y0 = 0;
force0 = 1;
time_bet_pics = 4;                %Time between pictures
freq_f = 2;                       %Frequency of data acquisition forces

% ==============
% END of OPTIONS
% ==============

% #####################################################################

% % ==========================================================
% % Force Array (read .dds) in case force is needed for IDIC. Otherwise,
% % fill array with ones equal in length of images.
% % =========================================================

F = zeros(length(timesteps),1);

% load data files
```

```
files_ten = dir([ddsdir filesep ddsfile '.dds']);
filenames = {files_ten.name}';
curvenum = cell(length(filenames),1);

% read the .dds file to extract the force column
fid = fopen([ddsdir filesep filenames{1}]);
data = textscan(fid, '%s%s%s%s%s%s%s%s', 'delimiter', ',', 'headerlines', 104);

% extract time, force, and elongation data
time = zeros(length(data{1}),1);
for i = 1: length(data{1})
    col1 = textscan(data{1}{i}, '%s %s', 'delimiter', '=');
    time(i) = str2double(col1{2});
end
force = str2double(data{2}(:));                         % [N]
elong = str2double(data{3}(:));                         % [um]

% % set the compressive force to zero
force = force-force(force0);

% extract the data that corresponds to the captured images (deformed only)
forces = force(force0:freq_f*time_bet_pics:freq_f*time_bet_pics*length(timesteps))

% CORRECT THE FORCE BY FACTOR 2 (incorrect calibration settings used for
F = (forces-force(1));

% ===========================
% IMAGE SELECTION
% ===========================
% reftime =  min(time(force == 0));

% list images in a certain directory
files = dir([imdir filesep '*.png']);

% store the reference image seperately (corresponding to the 0-force time
% increment)
ftemp2 = double(imread([imdir filesep files(nearest(reftime)).name]));
sizeftemp = size(ftemp2);

if sizeftemp(3) == 3
    ftemp = ftemp2(:,:,1);
else
    ftemp = ftemp2;
end

% -------------------------------------
% Image Processing of reference image
% -------------------------------------
% crop the image
if crp ~= 0
    ftemp = ftemp(crp(1):crp(2),crp(3):crp(4));
end

% smooth the image
if blur == 1;
    % apply a smoothing kernel to blur the pattern
    avg        = ones(windowsize)/windowsize^2;
    h = fspecial('average',windowsize)
    fblur = imfilter(ftemp, h);

    f = fblur;
elseif blur == 0
    f = ftemp;
end
% take the image gradient and add to image
```

```
if grad == 1
    % take the gradient of the image
    fg = gradient(f);

    % add the blurred and gradient images to obtain the eventual image
    fadd = f + fg./max(fg(:));
    f = fadd./max(fadd(:));
end


% -----------------------------------
% Image Processing of deformed image
% -----------------------------------
% loop over the deformed image files, convert to gray-values, and store in structure
for i = 1:length(timesteps)
    timeinc = timesteps(i);
    gtemp2 = (imread([imdir filesep files(timeinc).name]));

    sizeftemp = size(gtemp2);

    if sizeftemp(3) == 3
        gtemp = gtemp2(:,:,1);
    else
        gtemp = gtemp2;
    end

    % crop the image
    if crp ~= 0
        gtemp = gtemp(crp(1):crp(2),crp(3):crp(4));
    end

    if blur == 1;
        % apply the smoothing kernel
        gblur = imfilter(gtemp, h);
        inc(i).g = gblur;
    elseif blur == 0
        inc(i).g = gtemp;
    end

    if grad == 1
        % take the gradient of the image
        gg = gradient(gblur);

        % add the blurred and gradient images to obtain the eventual image
        gadd = inc(i).g + gg./max(gg(:));

        % store in structure for saving
        inc(i).g = gadd./max(gadd(:));
    end
end

% discretize the spatial domain in terms of length units
check = input(’Is the pixel size correct? (1/0) \n’);
[n m] = size(f);
x = pxs(1).*(0:m-1)’;
y = pxs(2).*((n-1):-1:0)’;       % flip the y-coordinate, to make
                                 % compatible with MSC.Marc model

% set the origin in the image to correspond with the origin in the model
if exist(’x0’)
else
    figure, imagesc(x,y,f), colormap(’gray’);
    [x0 y0] = ginput(1);
end
x = x - x0;
```

```
y = y - y0;

% ==================
% Save image
% ==================
if store == 1
    % save the images
    save([imdir filesep matname '.mat'], 'F', 'f', 'inc', 'm', 'n', 'x', 'y','-v7.3');
end

for i = 1:length(inc)
    imagesc(inc(i).g)
    colormap('gray')
    colorbar
    pause(0.4)
end
```

E.2 Image to mesh contour

Listing 3: *Image2Mesh.m*

```
%% Images 2 mesh
% Create a  mesh contour for marc mentat
%
% Make sure that the image size is identical to immat.m
%
% Jesse Joustra (2019)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc; clear all; close all;


%% OPTIONS

imagename = '02_dogbone_t003_c.png';                         % Name of figure
procname  = 'procfile';
pixsize   = 0.0061;                                          % mm per pixel

%% END OPTIONS

A = imread(imagename);                                       % Read image values
A = A(:,:,1);
A = flip(A);                                                 % Flip image to match
[x,y]   = meshgrid(1:size(A,1),1:size(A,2));                 % Make pixel mesh


% ==========================================
% select nodes for writing .proc-file with
% ==========================================
% prompt for graphical node selection
reply = input('Graphical nodal selection (y/n)?', 's');
if reply == 'y'
    % plot the nodes
    figure
    scat = imagesc(A); colormap('gray')
    %set(scat,'sizedata',20)
    set(gca, 'xminorgrid','on','yminorgrid','on')
    view([0 90])

    % graphically select BC-nodes
    msgbox('select the BC-nodes. Export datatips to "cursor_info"')

    % wait until the graph is closed before continuing
    waitfor(scat);
else
    load('cursor_info2');
end

nodalvalues = zeros(size(length(cursor_info),2));
pixelvalues = zeros(size(length(cursor_info),2));

%cursor_info = flip(cursor_info);
%% for making data points
for i = 1:length(cursor_info)
    pixelvalues(i,1) = cursor_info(i).Position(1);
    pixelvalues(i,2) = cursor_info(i).Position(2);
    nodalvalues(i,1) = cursor_info(i).Position(1)*pixsize-pixsize;
    nodalvalues(i,2) = cursor_info(i).Position(2)*pixsize-pixsize;
end

dim = size(y)
%nodalvalues(:,2) = dim(2)*pixsize-nodalvalues(:,2);
```

```

%% Analyse the points
figure
imagesc(A); colormap('gray');
hold on
scatter(pixelvalues(:,1),pixelvalues(:,2))

%% Proc writing
% create a new .proc file to which the Mentat-commands are printed
procfile = fopen([pwd filesep procname '.proc'],'w+');
labels = ['x' 'y'];

% loop over the number of BC-nodes
for j = 1:length(nodalvalues)
    % string to add node
    nodeadd = ['*add_points ' num2str(nodalvalues(j,1)) ' ' num2str(nodalvalues(j,2)) ' 0\n'];
    % save name to procfile
    fprintf(procfile, nodeadd);
end

for j = 1:(length(nodalvalues)-1)
    lineadd = ['*add_curves ' num2str(j) ' ' num2str(j+1) '\n'];
    fprintf(procfile, lineadd);
end
```

E.3 DIC results to boundary conditions

Listing 4: *ubuild2_2.m*

```
% Create input tables as .proc files for MSC.Mentat from GDIC displacement
% data.
%
% Necessary files:
%   1. .dat-file containing an MSC.Marc model
%   2. D-structure from GDIC analysis by GUI of Jan Neggers
%   3. .mat-file containing image structures "inc.g", "f", "x", "y"
%
% Andre Ruybalid (c)                        2018
% =========================================================================
clear all

% =========================================================================
% NECESSARY FILES and OPTIONS
% -------------------------------------------------------------------------
% model (.dat) data
% ----------------------
datpath = pwd;
datfile = '02_07_Dogbone';

% gdic data (D-structure)
%--------------------------
gdicpath = [pwd filesep 'GDIC_Base_Code'];
gdicfile = 'BC_left_02_07';

% image directory and file (.mat)
% ----------------------------------
imagepath = pwd;
imagefile = '02_07_Dogbone';

% set the directory and filename of the .proc file
%--------------------------------------------------
pathname = [pwd filesep 'GDIC_Base_Code'];
filename = ['BC_' gdicfile];

% time increments for table written in proc-file
timeincs = [4:4:200];

% --------------------------------------------------------------------------
% END OPTIONS
% =========================================================================

% -----------------------------------
% Load files and initizalize some data
% -----------------------------------
% load the saved .mat-file containing D
load([gdicpath filesep gdicfile '.mat'])
D = bgdic;

% load image data
load([imagepath filesep imagefile '.mat'])
pxs = x(2)-x(1);

% determine ROI coordinates
if exist('D')
    xroi = x(D.cor(1).xroi);
    yroi = y(D.cor(1).yroi);
elseif exist('X')
    xroi = X(1,:);
    yroi = Y(1:6,1);
end
```

```

% EXTRACT NODAL COORDINATES OF ROI
% -------------------------------------------------------------------------
% read the node numbers and coordinates from a .dat-file
[datdata, nodcoord] = nodecoord([datpath filesep datfile '.dat']);

% find the indices of nodal coordinates within ROI
[I,J] = find(nodcoord(:,2) >= xroi(1) & nodcoord(:,2) <= xroi(end) ...
    & nodcoord(:,3) >= yroi(end) & nodcoord(:,3) <= yroi(1));

% extract the node numbers and coordinates of the ROI
nodenum = nodcoord(I,1);
nodalx = uniquetol(nodcoord(I,2));
nodaly = uniquetol(nodcoord(I,3));
%nodalz = uniquetol(nodcoord(I,4));

% INTERPOLATE THE GDIC-FIELDS
% -------------------------------------------------------------------------
% for image (GDIC) data
[Xnode, Ynode] = meshgrid(nodalx, flipud(nodaly));
if exist('D')
    Uxnode = zeros(size(Xnode,1), size(Xnode,2), length(D.cor));
    Uynode = zeros(size(Xnode,1), size(Xnode,2), length(D.cor));

    for i = 1:length(D.cor)
        Uxnode(:,:,i) = griddata(xroi, yroi, pxs*double(D.cor(i).U1), Xnode, Ynode, 'cubic');
        Uynode(:,:,i) = griddata(xroi, yroi, -pxs*double(D.cor(i).U2), Xnode, Ynode, 'cubic');
    end

% for VE-data
elseif exist('Uxref')
    Uxnode = zeros(size(Xnode,1), size(Xnode,2), length(timeincs)-1);
    Uynode = zeros(size(Xnode,1), size(Xnode,2), length(timeincs)-1);

    for i = 1:(length(timeincs)-1)
        Uxnode(:,:,i) = griddata(X, Y, Uxref(:,:,timeincs(1+i)), Xnode, Ynode, 'cubic');
        Uynode(:,:,i) = griddata(X, Y, Uyref(:,:,timeincs(1+i)), Xnode, Ynode, 'cubic');
    end
end

%% Alternative coding --> Jesse

nodinfo = cell(length(nodenum), 3);

for i = 1:length(nodenum)
    Inx = find(isnear(nodcoord(I(i),2), Xnode(1,:),1e-6));
    Iny = find(isnear(nodcoord(I(i),3), Ynode(:,1),1e-6));
    nodinfo{i,1} = nodcoord(I(i),1);
    for j = 1:size(Uxnode, 3)
        Uxtemp(1,j) = Uxnode(Iny,Inx,j);
        Uytemp(1,j) = Uynode(Iny,Inx,j);
    end
    nodinfo{i,2} = [0 Uxtemp];
    nodinfo{i,3} = [0 Uytemp];
end


% ==========================================
% select nodes for writing .proc-file with
% ==========================================
% prompt for graphical node selection
reply = input('Graphical nodal selection (y/n)?', 's');
if reply == 'y'
    % plot the nodes
    figure
    scat = scatter3(nodcoord(I,2), nodcoord(I,3), nodcoord(I,1), 's');
    %set(scat,'sizedata',20)
    set(gca, 'xminorgrid','on','yminorgrid','on')
    view([0 90])
    %set(gcf, 'position', [51 310 1646 481])

    % graphically select BC-nodes
    msgbox('select the BC-nodes. Export datatips to "cursor_info".' ...
        ' Close graph afterwards')

    % wait until the graph is closed before continuing
    waitfor(scat);

    nodeselect = zeros(size(length(cursor_info),3));

    cursor_info = flip(cursor_info);
    for i = 1:length(cursor_info)
        nodeselect(i,1) = cursor_info(i).Position(3);
        nodeselect(i,2) = 1;        % x-BC active
        if i == 1
            nodeselect(i,3) = 1;        % y-BC active
        elseif i == length(cursor_info)
            nodeselect(i,3) = 1;        % y-BC active
        else
            nodeselect(i,3) = 0;        % y-BC active --> Not in the middle
        end
    end
end



% ========================================
% BC-WRITING TO .proc-file
% ========================================
% create a new .proc file to which the Mentat-commands are printed
procfile = fopen([pathname filesep filename '.proc'],'w+');

labels = ['x' 'y'];

% loop over the number of BC-nodes
for j = 1:length(nodeselect)
    % loop over the dimensions
    for i = 1:2;
        if nodeselect(j, i+1) ~= 0
            % find the selected data in the nodinfo cell
            index = find([nodinfo{:,1}] == nodeselect(j,1));

            % node number and directional label for labeling/naming of tables
            nnum = nodinfo{index,1};

            % add the appropriate string to create a new table with time as the
            % independent variable
            nwtabstr = ['*new_md_table 1 1\n*set_md_table_type 1\ntime\n*table_name ' ...
                num2str(nnum) labels(i) '\n*table_add\n'];
            fprintf(procfile, nwtabstr);
            Uproc = [timeincs' nodinfo{index,i+1}'];
            fprintf(procfile, '%i %d\n', Uproc');

            % add the table as a boundary condition to a specific node and add the BC
            % to the loadcase
            bcaddstr = ['*new_apply *apply_type fixed_displacement\n*apply_dof ' ...
                labels(i) ' *apply_dof_value ' labels(i) '\n*apply_dof_table ' labels(i) ...
                '\n' num2str(nnum) labels(i) '\n*add_apply_nodes\n' num2str(nnum) ...
                '\n# | End of List\n*apply_name ' num2str(nnum) labels(i) ...
                '\n*add_loadcase_loads ' num2str(nnum) labels(i) '\n'];
            fprintf(procfile, bcaddstr);
        end
    end
end

% SAVE .mat-file with relevant data
if exist('D')
    save([gdicpath filesep filename '.mat'], 'D', 'nodinfo', 'Xnode', ...
        'Ynode', 'pxs', 'Uxnode', 'Uynode','nodeselect', 'timeincs')
elseif exist('Uxref')
    save([gdicpath filesep gdicfile '.mat'], 'nodinfo', 'Xnode','Ynode', ...
        'pxs', 'Uxnode', 'Uynode','nodeselect','Uxref','Uyref','X','Y','timeincs')
end
```

E.3.1 determine nodal coordinates

Listing 5: *nodecoord.m*

```
% This file reads an MSC.Mentat produced .dat-file and extracts
% the node numbers and nodal coordinates.
%
% (c) Andre Ruybalid
%                                                    August 2016
% ==============================================================
% ==============================================================

function [datdata, nodcoord] = nodecoord(file)
% READ NODAL INFORMATION
% -------------------------------------------------------------------------
% scan the .dat-file
datid = fopen(file);
datdata = textscan(datid,'%s', 'whitespace','', 'delimiter','\n');

% select the line at which the nodal coordinate information begins
line =  strmatch('coordinates',datdata{1});

% read the total number of coordinates
nnod = str2double(datdata{1}{line+1}(11:20));

% store the nodal information (string) in a new array
nodstring = datdata{1}(line+2:line+1+nnod);

% fill array with doubles: [node number, x, y, z]
nodcoord = zeros(length(nodstring), 3);
for i = 1:nnod
    nodcoord(i,1) = str2double(nodstring{i}(1:10));

    % convert the x-coordinate string-part to floating point and fill array
    nodcoord(i,2) = datfloat(nodstring{i}(11:30));

    % convert the y-coordinate string-part to floating point and fill array
    nodcoord(i,3) = datfloat(nodstring{i}(31:50));

end
```

E.3.2 Check for nodes to be near

Listing 6: *isnear.m*

```
function tf=isnear(a,b,tol)
%ISNEAR True Where Nearly Equal.
```

```
% ISNEAR(A,B) returns a logical array the same size as A and B that is True
% where A and B are almost equal to each other and False where they are not.
% A and B must be the same size or one can be a scalar.
% ISNEAR(A,B,TOL) uses the tolerance TOL to determine nearness. In this
% case, TOL can be a scalar or an array the same size as A and B.
%
% When TOL is not provided, TOL = SQRT(eps).
%
% Use this function instead of A==B when A and B contain noninteger values.

% D.C. Hanselman, University of Maine, Orono, ME 04469
% Mastering MATLAB 7
% 2005-03-09

%-----------------------------------------------------------------------
if nargin==2
%    tol=sqrt(eps);        % original version
        tol=sqrt(eps(max(abs(a),abs(b))));      %  my version. sqrt seems like 'overdamping'
end

% when in need of optimization, drop the error checks
if ~isnumeric(a) || isempty(a) || ~isnumeric(b) || isempty(b) ||...
   ~isnumeric(tol) || isempty(tol)
   error('Inputs Must be Numeric.')
end
if any(size(a)~=size(b)) && numel(a)>1 && numel(b)>1
   error('A and B Must be the Same Size or Either can be a Scalar.')
end

% main line
tf=abs((a-b))<=abs(tol);
```