Full length article

Implementation and comparison of algebraic and machine learning based tensor interpolation methods applied to fiber orientation tensor fields obtained from CT images

J. Blarr [a,*], T. Sabiston [b], C. Krauß [c], J.K. Bauer [d], W.V. Liebig [a,f], K. Inal [b], K.A. Weidenmann [e,f]

[a] Karlsruhe Institute of Technology (KIT), Institute for Applied Materials – Materials Science and Engineering (IAM-WK), Engelbert-Arnold-Straße 4, 76131 Karlsruhe, Germany
[b] University of Waterloo, Department of Mechanical and Mechatronics Engineering, Waterloo, Canada
[c] Karlsruhe Institute of Technology (KIT), Institute of Vehicle System Technology - Division Lightweight Technology, Karlsruhe, Germany
[d] Karlsruhe Institute of Technology (KIT), Institute of Mechanics (IFM), Karlsruhe, Germany
[e] Universität Augsburg, Institute of Materials Resource Management, Augsburg, Germany
[f] Fraunhofer Institute for Chemical Technology ICT, Pfinztal, Germany

ARTICLE INFO

ABSTRACT

Fiber orientation tensors (FOT) are used as a compact form of representing the mechanically important quantity of fiber orientation in fiber reinforced composites. While they can be obtained via image processing methods from micro computed tomography scans (μCT), the specimen size needs to be sufficiently small for adequate resolution – especially in the case of carbon fibers. In order to avoid massive workload by scans and image evaluation when determining full-field FOT distributions for a plaque or a part, e.g., for comparison with process simulations, the possibilities of a direct interpolation of a few measured FOT at specific support points were opened in this paper. Hence, three different tensor interpolation methods were implemented and compared qualitatively with the help of visualization through tensor glyphs and quantitatively by calculating originally measured tensors at support points and evaluating the deviations. The methods compared in this work include two algebraic approaches, firstly, a Euclidean component averaging and secondly, a decomposition approach based on separate invariant and quaternion weighting, as well as an artificial intelligence (AI)-based method using an artificial neural network (ANN). While the decomposition method showed the best results visually, quantitatively the component averaging method and the neural network behaved better (that is for the type of quantitative error assessment used in this paper) with mean absolute errors of 0.105 and 0.114 when calculating previously measured tensors and comparing the components. With each method providing different advantages, the use for further application as well as necessary improvement is discussed. The authors would like to highlight the novelty of the methods being used with small and CT-based tensor datasets.

1. Introduction

Fiber reinforced polymers (FRP) have gained increasing relevance as engineering material in multiple sectors like automotive and aeronautic industry due to their good processability and outstanding specific mechanical properties [1]. The use of thermoplastics instead of thermosets as matrix material has become particularly interesting recently due to better recyclability [2] and faster processing [3]. Carbon fibers as reinforcement material is the more expensive but also much stiffer and stronger option compared to glass fibers [4]. However, a lot of image processing and computational methods are better developed for glass fibers – mostly due to their larger diameter and thereby superior detection possibility in high resolution images – and the density-related better contrast to the matrix material (cf. Hounsfield scale).

Using heterogeneous materials implies the need for advanced characterization methods. The complex microstructure of short and long fiber reinforced polymers (e.g., in contrast to UD laminates) has a high influence on the mechanical behavior of FRP on component level [5]. X-ray μCT scans and consecutive image processing methods have become one of the leading non-destructive testing (NDT) methods. The creation of a 3D digital copy, a so-called "digital twin", of a real

---

part, which is equipped with quantitative information about material, process and possible treatment constitutes an important idea in modern continuous, data-intensive monitoring and early detection of material behavior and failure. As an example, virtual process chain approaches (also called CAE chains) [6] can be mentioned that deal with data transferal across different scales between virtual chain and physical part. In order to quantify the convoluted microstructure of short and long fiber reinforced polymers, various properties have been developed over time. Influence on physical behavior is attributed mostly to the amount of fibers in a specific area or volume compared to the amount of matrix material (fiber volume content, FVC) [7], the length of the fibers (fiber length distributions, FLD) [8,9] and the orientation of the fibers in the material (fiber orientation distributions, FOD). Fiber orientation is generally dependent on the position in the part/plate in the long fiber reinforced thermoplastic (LFT) investigated in this work. The material flow in the underlying compression molding process influences the fiber alignment. Fiber orientation therefore has an enormous influence on the anisotropy of the mechanical properties. Although FOD are classically described by scalar distribution functions, the compact representation of fiber orientation tensors (FOT) has prevailed in process or structural simulations [10,11]. Bauer et al. described the variety of FOT in detail [12]. FOT can be determined both for specimen from finished plaques as well as for describing the initial fiber orientation state in the plastificate [13]. In certain material combination cases, quantities other than fiber orientation tensors also provide a suitable description of the orientation, such as in the case of glass fiber-reinforced sheet molding compounds (SMC). This material has a typical bundle-like structure, so that a description of the mesoscopic homogenized fiber bundles as integration in simulations makes sense [14].

From µCT images, FOT can be obtained using gradient-based methods, such as the structure tensor [15]. While FVC and FLD can also be determined experimentally by either pyrolyzing the matrix material or dissolving it chemically [16–18], another method (outside of calculations with homogenization models) for fiber orientation tensor identification is difficult. Furthermore, the dilemma of sufficient resolution and significant sample size arises: Especially in the case of carbon fibers which typically show diameters of around 5–7 µm (and are hence significantly thinner than glass fibers), a resolution of three to four times the diameter as edge length per voxel (which is necessary to resolve single fibers) amounts to extremely small volumes. In addition to the physical limitation of CT imaging, the scale bridging problem of microstructured materials occurs: a volume too small as specimen would result in resolving orientation too locally and these small volumes often do not represent the microstructure sufficiently well, whereas (if physically possible) a tensor from one scan of a volume too big generalizes too much compared to a meaningful averaging of many local tensors. The size of a representative volume element (RVE) is difficult to find and evaluate in the case of LFT but has been considered before in the form of statistical volume elements (SVE) [19–21]. Assuming small sample sizes, the determination of FOT across an entire plaque (that would allow, e.g., for the comparison with resulting FOT from process simulations or as input variable for structural simulations) would require the scan and consecutive image evaluation of a high amount of samples. However, since full-field information of FOT is highly interesting for analysis and improvement of the preceding process, an alternative to costly scan generation and evaluation was sought. Accordingly, this work is a study of the extent to which a few (in this case nine) measured tensors provide sufficient information to represent the overall plate progression. It is an interpolation question, which, however, gets its difficulty in the non-scalarity of the supporting point information. Moving away from purely mathematical or academic cases of two tensors or the like, this work explicitly aims to evaluate a benchmark of different code-based methods in the context of this engineering application. Therefore, the example used is explicitly a compilation of very dissimilar tensors.

2. State of the art

The fiber orientation state can be described by a probability distribution function $\Psi(\boldsymbol{p})$ providing the probability of finding a fiber in direction of a unit vector $\boldsymbol{p}$. The unit sphere $\mathscr{S}$ can be generated by the set of all possible directions of $\boldsymbol{p}$, so that

$$\mathscr{S} = \{\boldsymbol{p} \in \mathbb{R}^3 \ | \ |p| = 1\}. \tag{1}$$

The orientation distribution function Ψ satisfies three main physical conditions. Firstly, $\Psi(\boldsymbol{p})$ must be periodic as it is impossible to distinguish between end and start point of a fiber, hence $\Psi(\boldsymbol{p})$ is a symmetric function:

$$\Psi(\boldsymbol{p}) = \Psi(-\boldsymbol{p}). \tag{2}$$

In addition, it is normalized, i.e., the integral of $\Psi(\boldsymbol{p})$ over the surface of the unit sphere equals one:

$$\int_{\mathscr{S}} \Psi(\boldsymbol{p}) \, d\boldsymbol{p} = \mathbf{1}. \tag{3}$$

$\Psi(\boldsymbol{p})$ further fulfills the continuity condition regarding the change in Ψ with time for changing fiber orientation, which shall not be elaborated here.

Advani and Tucker [10] then simplified the description of fiber orientation states with the introduction of fiber orientation tensors which are formed by the dyadic product of the vector $\boldsymbol{p}$ multiplied with the distribution function and a subsequent integration over all possible directions.

Due to the periodic nature of Ψ, all odd-order integrals amount to zero while theoretically all even-order tensors are possible; with the second- and fourth-order orientation tensors being commonly used:

$$\boldsymbol{A} = \int_{\mathscr{S}} \Psi(\boldsymbol{p})\boldsymbol{p} \otimes \boldsymbol{p} \, dp, \tag{4}$$

$$\mathbb{A} = \int_{\mathscr{S}} \Psi(\boldsymbol{p})\boldsymbol{p} \otimes \boldsymbol{p} \otimes \boldsymbol{p} \otimes \boldsymbol{p} \, dp. \tag{5}$$

FOT are completely symmetric and their trace equals one.

The distribution function can be recovered from the orientation tensors when they are displayed in a different form (deviatoric versions). An orientation tensor is equivalent to truncating the series by which the distribution function is given by. The distribution function is represented more accurately by higher-order tensors, yet Advani and Tucker [10] argue that the representation of second- and fourth-order tensors is usually sufficient for most applications.

The problem of tensor interpolation can be theoretically avoided for fiber orientation tensors, compared to, for example, stress and strain tensors, by switching to the scalar distribution function. Subdivision into discrete directions and subsequent interpolation would equal a Euclidean interpolation of the scalar-valued function $\bar{\Psi}$. In fact, this corresponds exactly to averaging the tensor components (with weights depending on distance). This can be explained by the fact that an integral is a linear mapping. However, this standard method led to a kind of "artificial" isotropy, which is shown by a change of shape in the manner of a rounding in the representation form of tensor glyphs. This does not necessarily seem to represent a useful averaging, as this issue of so-called tensor swelling arose both in the field of medical technology when interpolating diffusion tensors from MRI images [22–24] and in the case of FOT mapping in process simulation applications [25]. This swelling effect is due to non-monotonic interpolation of the tensor determinant and the Euclidean method does not preserve the positive definiteness. It was part of scientific discussions whether a more isotropic tensor as an interpolation between two anisotropic tensors that point in different directions still constitutes a reasonable behavior of fibers. It would imply that in a region of changing flow direction first some fibers turn and others stay in the original direction until most are turned in the end. The opposite idea that most turn first a bit and later completely into the new direction would rather be indicative of tensors

in the center to be not significantly less anisotropic than the two next to it. As a reference solution, the first method for tensor interpolation implemented and used in this paper is this Euclidean interpolation, which will be called component averaging (CA) from now on.

The complex Riemannian interpolation is another "global" interpolation method [26]. However, if more than two input arguments are used, the underlying computations can only be solved implicitly, which requires an iterative and therefore computationally expensive calculation. Since this paper explicitly seeks an application-oriented method that is as fast and simple as possible while maintaining the highest possible quality, Riemannian interpolation will not be discussed further. Another logarithmic, but explicitly solvable approach was introduced by Arsigny et al. [22]: the Log-Euclidean tensor interpolation method. This method was already considered for FOT by Krauß and Kärger [25].

A completely different approach are so-called decomposition methods. These methods make use of the fact that symmetric positive definite (SPD) tensors can be decomposed into eigenvalues and eigenvectors in spectral decomposition (cf. Eq. (13)). In terms of tensor glyphs, the eigenvalues are responsible for the shape, while the eigenvectors are responsible for the orientation of the tensor in space. Thus, shape and orientation can be weighted (according to various possible distance measures) and interpolated separately and then recomposed into a tensor. This can be done directly via the eigenvalues and eigenvectors, or via detours with the help of other invariants and, for example, quaternions. This method allows in particular the resolution of the swelling effect [23,25]. It should be mentioned here that there is not only one decomposition method, but this must be understood as a kind of umbrella term, which can be executed very differently in the individual steps. The only previous use of this concept for FOT by Krauß and Kärger [25] differs from the one implemented in this work, for example. The exact concepts used for the decomposition method will be explained in detail in the Methods chapter (cf. chapter 4) and have been previously summarized by the authors in [27], but basic considerations about the decomposition method will be elaborated in the following paragraph.

As far as the shape is concerned, a direct interpolation of the eigenvalues would be conceivable. There are also approaches which handle it this way [28]. But Ennis et al. developed the concept of orthogonal invariants [24], which seem to perform very well for physical problems. Each set of invariants decomposes the tensor shape with an orthogonal basis so that the derivatives of these invariants I_i with respect to the (fiber orientation) tensor $\mathbf{A}$ behave as follows:

$$\frac{\delta I_i}{\delta \mathbf{A}} \cdot \frac{\delta I_j}{\delta \mathbf{A}} = 0, i \neq j. \tag{6}$$

Ennis et al. established the so-called K- and R-invariants, which are also used in this paper. They argue that while eigenvalues form a set of orthogonal invariants as well, they have the disadvantage of not isolating essential attributes of tensor shape like size and anisotropy which the sets of K- and R-invariants provide instead.

A direct interpolation of the eigenvalues has still been performed as a test but indeed seemed to distort the results and will not be considered in more detail in this paper.

When it comes to the interpolation of the orientation of the tensors, there are multiple works which focus on the interpolation of rotations in 3D, i.e., elements in the 3D rotation group, which is also called $SO(3)$, as a separate mathematical problem independent of tensors or the shape of the same [29–32]. However, many of these papers focus on the smooth rotation between two or few different orientation states. These methods do not necessarily perform just as well for multiple and spreaded orientation tensors. Typically, the orientation of a general basis in 3D linear algebra is described by Euler angles with respect to a fixed coordinate system. Generally, it must be considered that an orientation can be described by 24 different coordinate systems. This ambiguity is counteracted with the help of conventions, this

includes the determination to use a right-handed system as well as the sorting of the eigenvalues according to magnitude. However, after the conventions still four possible coordinate systems remain to describe an orientation. Depending on the choice it can be influenced whether between two orientations, which, e.g., lie only 20° apart, the interpolated tensor rotates by 10°, which corresponds to the – at least at first – obviously correct option, or by 170°, which would correspond to the mirrored coordinate system. For two tensors between which one wants to interpolate, it therefore makes sense to implement a query and restrict the size of the angle. However, for a set of measured tensors $\mathcal{T}_m$ at multiple supporting points, where the ones further away from the one that is currently to be calculated have less weight but are still included in the calculation of this interpolated tensor, this becomes less obvious. This aspect will be taken up again in the discussion.

A much-investigated method is the orientation interpolation via quaternion. Quaternions as described by Hamilton extend the complex number system and are usually represented in the following form:

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k} \quad \text{with} \tag{7}$$
$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1 \quad \text{(compare complex numbers).}$$

The result is a four-dimensional number system (mathematically: a vector space) with a real part consisting of one real component and an imaginary part consisting of three components, which is also called the vector part. Multiplication of quaternions is noncommutative. Quaternions allow in many cases a computationally elegant description of three-dimensional Euclidean space, especially in the context of rotations. By using quaternions instead of Euler angles the problem of Gimbal lock can be avoided and they are simpler to compose. Compared to the rotation matrices, quaternions are more compact, efficient, and numerically stable. Regardless, not all ambiguities can be circumvented. The unit quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. This means that there is a 2:1 homomorphism from quaternions of unit norm to the 3D rotation group $SO(3)$. In other words, $SO(3)$ is double-covered by quaternions. This sign ambiguity has to be paid attention to when computing a quaternion from the rotation matrix.

Alternative fiber orientation interpolation methods were developed, e.g., by Köbler et al. [33]. They developed a mechanical interpolation method in a surrogate model, i.e., they first calculated the material response for discrete fiber orientations and then used linear interpolation on the fiber orientation triangle (a material model as a function of the orientation triangle). While this yields good results, it is not related to the interpolation of orientations on component level and is therefore of no interest in the application thought of in this work.

The use of artificial intelligence (AI) for fiber orientation tensor interpolation has been explored by Sabiston et al. [34]. The authors used a large number of FOT obtained – as in this work – from µCT images of multiple plates of the same process. This represented their ground truth, which they used to train the artificial neural network (ANN), which was subsequently able to predict the tensor components for plates of this process with less deviation than the variability was between neighboring microstructure units. Since this represents an entirely new, non-physical, nor classically linear-algebraic way, a neural network is used as the third interpolation method in this work. However, in contrast to Sabiston's work, the ANN is trained using only the nine measured tensors considered for all methods for comparison. Then, the remaining 160 are predicted using the trained network. Even though the nine tensors give six values of information each, the use of AI with such a small amount of input data is rare, but it is intended to assess whether this can still produce reasonably useful results, or whether a useful result can be expected with a small additional number of given tensors.

Consequently, this study focuses on determining a full-field distribution of fiber orientation tensors across an entire carbon long fiber reinforced polyamide 6 plaque by interpolating orientation tensors determined from small samples at specific positions. Therefore, nine

Table 1
Chosen CT parameters for the scans, which the FOT were evaluated from.

Parameter	Chosen value
No. of projections	1950
Acceleration voltage	150 kV
Current	0.25 mA
Exposure/Integration time	500 ms
Frame binning	2
Voxel size	8.57 μm/voxel

samples of 10 mm × 10 mm × 4 mm were scanned in the CT and the orientation tensor was determined for each scanned volume via structure tensor approach implemented by Pinter et al. [35]. Following, the nine computed tensors of second order were interpolated via three different methods to predict 160 additional fiber orientation tensors in-between. The results were compared visually based on their physical meaningfulness. Subsequently, they were compared quantitatively by leaving one of the measured tensors out of the input data provided to each method and computing the originally measured tensor instead. The difference between the computed and the measured tensor is evaluated through the deviation of the absolute component values and via the Frobenius norm. The implemented interpolation methods include Euclidean interpolation (component averaging), a decomposition-based method interpolating the shape of the tensors with the help of orthogonal invariants and the orientation of the tensors using quaternions and a method based on artificial intelligence (AI) with an artificial neural network (ANN). While most of these methods have either been developed from a mathematical or theoretical point of view or – if a physical use-case existed – worked with a high amount of input tensors (in case of the ANN), the application of these methods to sparse but measured CT-based orientation data has not yet been explored to the authors' knowledge.

3. Notation

Symbolic tensor notation is preferred in this work. Scalar values are denoted by standard Latin and Greek letters, e.g., c, λ. Tensors of first order are represented by bold lowercase letters, e.g., $\mathbf{x}, \mathbf{p}$ and bold uppercase letters are used for tensors of second order such as $\mathbf{R}, \mathbf{S}$. Fourth-order tensors are denoted by double-struck letters like $\mathbb{C}, \mathbb{S}$.

Sets, i.e., collections of quantities, are denoted by calligraphic symbols, e.g., $\mathscr{A}$ and are constructed by curly braces. In them, the elements typically are given explicitly or expressed by conditions to be fulfilled by each element contained in the set. The special orthogonal group $SO(3)$ represents all 3D rotations. Four-dimensional quaternions are represented by an arrow-head above the Latin letter, such as in $\vec{q}$.

The terms $tr()$ and $det()$ are the trace and determinant operators respectively, $\|\mathbf{D}\|$ represents the Frobenius norm of the tensor $\mathbf{D}$ defined by $\|\mathbf{D}\| = \sqrt{tr(\mathbf{D}\mathbf{D}^T)}$. The rotation of a tensor is denoted by the Rayleigh Product $\star$.

4. Methods

4.1. Scan acquisition and determination of fiber orientation tensors

The authors used a YXLON-CT precision μCT system with a flat panel PerkinElmer Y.XRD1620 detector with 2048 pixel × 2048 pixel for the acquisition of the microstructural images. The parameters of the performed μCT scans are listed in Table 1. The resulting volumetric images are reconstructed applying the Feldkamp cone-beam algorithm [36].

Subsequently, the reconstructed scans were processed in VG Studio Max 3.4.2 and cropped into regions of interest (ROI). If necessary, brightness and contrast were then adjusted in the ImageJ (FIJI) software. In addition, the individual gray value threshold was determined

for each scan, which was required for the use of the method by Pinter et al. [35] that determined the FOT from the scan data. Pinter et al. implemented their code in C++ with the help of the ITK library. The algorithm makes use of the structure tensor (cf. Eq. (8)) as it performed best out of three different implemented methods:

$$\mathbf{S} = \begin{bmatrix} \left(\dfrac{\delta I}{\delta x}\right)^2 & \dfrac{\delta I}{\delta x}\dfrac{\delta I}{\delta y} & \dfrac{\delta I}{\delta x}\dfrac{\delta I}{\delta z} \\[2ex] \dfrac{\delta I}{\delta y}\dfrac{\delta I}{\delta x} & \left(\dfrac{\delta I}{\delta y}\right)^2 & \dfrac{\delta I}{\delta y}\dfrac{\delta I}{\delta z} \\[2ex] \dfrac{\delta I}{\delta z}\dfrac{\delta I}{\delta x} & \dfrac{\delta I}{\delta z}\dfrac{\delta I}{\delta y} & \left(\dfrac{\delta I}{\delta z}\right)^2 \end{bmatrix}, \tag{8}$$

with I being a discrete function describing the gray-value intensity of the reconstructed image. As the structure tensor is based on the first numerical derivative, it is known to be a robust approach. In the used algorithm, the structure tensor calculation is combined with a Gaussian blur of a width of σ. In this work $\sigma = 0.2$ voxels was chosen as it showed meaningful results across all scans. Another parameter that had to be adjusted before running the algorithm was the mask size. The obtained tensors were averaged by another Gaussian blur representing the image region that is taken into account to determine an orientation at a certain point. It has to be larger than the first one that was used for the derivative and was chosen to two voxels in this work. The algorithm then calculates the smallest eigenvalue and its corresponding eigenvector of the resulting tensor as fiber orientation. The FOT calculated with this algorithm constitute the foundation or ground truth that are fed into the different interpolation methods. They are henceforth called "measured values" implicitly including that – however – these FOT are subject to a certain error as well.

4.2. Interpolation methods

4.2.1. General

In general, all interpolation methods were mainly implemented in Python 3.8.

Symmetric positive definite (SPD) tensors can be visualized as superquadric tensor glyphs [37,38]. This method will be used in this work as it constitutes a descriptive and interpretable way of assessing the success of the different implemented interpolation methods. The authors implemented the visualization in Matlab R2020b with the help of the "plotDTI" function of the fanDTasia ToolBox by Barmpoutis et al. [39].

The overall idea of all three implemented interpolation methods is to get FOT values for the 160 positions in the plaque that are missing, from calculations with the measured nine FOT at the given grid positions. Determining values between a set of measured values, here the set $\mathscr{T}_m = \{\mathbf{UL}, \mathbf{UM}, \mathbf{UR}, \mathbf{ML}, \mathbf{MM}, \mathbf{MR}, \mathbf{LL}, \mathbf{LM}, \mathbf{LR}\}$ (respectively denoting "Upper Left", "Upper Middle", "Upper Right", "Middle Left", …, "Lower Left", etc.) of measured FOT computationally, based on the set of measured values, describes a classical interpolation problem. An interpolation scheme $\bar{\phi}$ is defined as a mapping f, which connects its arguments, on the one hand a set of $N \geq 1$ discrete values ϕ_i and on the other hand their associated weights $w_i \in [0; 1]$:

$$\bar{\phi} = f(\phi_i, w_i) \tag{9}$$

As a weight function, multiple options are conceivable with the possibly simplest being Shepard's inverse distance weighting (IDW)

$$w_I(\mathbf{x}) = \frac{1}{d(\mathbf{x}, \mathbf{x}_i)^p}, \tag{10}$$

as an explicit approach with $\mathbf{x}$ denoting an arbitrary point that shall be interpolated, $\mathbf{x}_i$ being a known interpolating point and d is the given distance from the known point $\mathbf{x}_i$ to the unknown point $\mathbf{x}$. p is a positive real number, called the "power parameter". Weight decreases as distance increases from the interpolated points. Greater values of p assign greater influence to values closest to the interpolated point, which results in nearly constant interpolated values for large values of p.
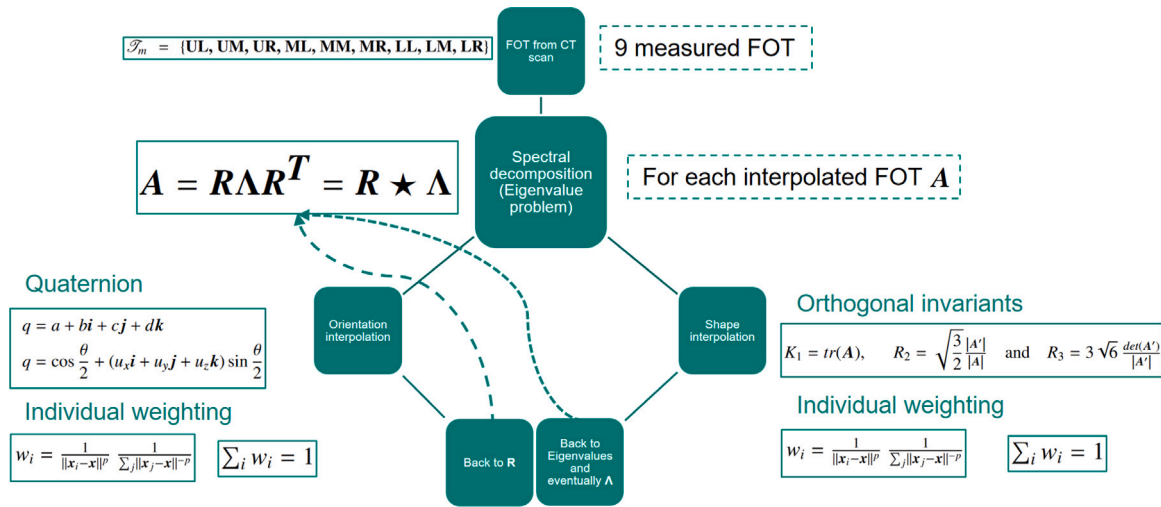
Fig. 1. Overview of the concept of the implemented decomposition method.

4.2.2. Orientation distribution function (ODF) and component averaging method

Recalling the definition of second-order orientation tensors as described by Advani and Tucker [10], Kanatani [11],

$$A = \int_{\mathscr{S}} \Psi(p) p \otimes p \, dp, \qquad (11)$$

with $\mathscr{S}$ being the unit sphere and dp the surface element on it, as well as p being the unit vector for the direction of the fibers, it appears that A is linear in $\Psi(p)$. Assuming the surface can be divided into two equally sized areas $\mathscr{S}_1$ and $\mathscr{S}_2$ with two distribution functions $\Psi_1(p)$ and $\Psi_2(p)$ and $\Psi_i(n) = \frac{1}{2}(\Psi_1(n) + \Psi_2(n))$ holds, this means that $A_i = \frac{1}{2}(A_1 + A_2)$ is exact, as integration is a linear functional and as an integral domain can always be divided into sub-intervals. This further implies that a direct averaging of the orientation function is equivalent to an averaging of the components of the orientation tensors.

Thus, the algorithm multiplies the components of each measured FOT by a weight that depends on its distance from the tensor being calculated.

As mentioned before, Shepard's inverse distance weighting method is used as weight function in all three methods with $p = 2$:

$$w_i = \frac{1}{\|x_i - x\|^p} \; \frac{1}{\sum_j \|x_j - x\|^{-p}}. \qquad (12)$$

Compared to Eq. (10), Eq. (12) features a necessary normalization factor.

4.2.3. Decomposition-based method

The method, which uses spectral decomposition of tensors, is shown schematically in Fig. 1 and is described in the following.

For the chosen decomposition approach, the shape and orientation of the tensors are to be interpolated separately. Therefore, the well-known spectral decomposition resulting from the Eigenvalue problem is used:

$$A = R \Lambda R^T = R \star \Lambda \qquad (13)$$

Λ denotes the tensor containing the Eigenvalues on the principal diagonal and R is defined as the orthogonal rotation matrix consisting of the normalized Eigenvectors.

Orientation

The rotation matrix R can be interpreted as a rotation around a rotation axis and can therefore be transformed into a quaternion as described in the state of art:

$$q = a + bi + cj + dk,$$
$$q = \cos\frac{\theta}{2} + (u_x i + u_y j + u_z k)\sin\frac{\theta}{2}, \qquad (14)$$

with rotation axis: $u = (u_x, u_y, u_z)^T$ and rotation angle θ.

The quaternion is calculated from the given rotation matrix R via:

$$t = tr(R), \quad r = \sqrt{1+t} \quad \text{and} \quad a = \frac{r}{2} \quad \text{with}$$
$$b = sgn(R_z y - R_y z)\left|\frac{1}{2}\sqrt{1 + R_x x - R_y y - R_z z}\right|,$$
$$c = sgn(R_x z - R_z x)\left|\frac{1}{2}\sqrt{1 - R_x x + R_y y - R_z z}\right|,$$
$$d = sgn(R_y x - R_x y)\left|\frac{1}{2}\sqrt{1 - R_x x - R_y y + R_z z}\right|. \qquad (15)$$

This is followed by the actual interpolation: $q_{ges} = \sum_i w_i q_i$ with weights: $\sum_i w_i = 1$ and the retransformation of q_{ges} in R:

$$R = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2(bc - ad) & 2(bd + ac) \\ 2(bc + ad) & a^2 - b^2 + c^2 - d^2 & 2(cd - ab) \\ 2(bd - ac) & 2(cd + ab) & a^2 - b^2 - c^2 + d^2 \end{bmatrix}. \qquad (16)$$

Projector interpolation

An alternative approach for orientation interpolation, circumventing the sign-ambiguity, is based on the bi-orthogonal projector decomposition with

$$P_i = \pm p_i \otimes \pm p_i = \frac{\partial A}{\partial \gamma_i} \qquad (17)$$

Gahm and Ennis [40] introduced this method as *Dyadic interpolation*, since each projector P_i can be expressed as dyadic product of its underlying eigenvector p_i. Subsequently, the projectors are interpolated separately and composed into a non-orthogonal deformation tensor F. Finally, polar or singular value decomposition yield the interpolated orientation tensor R. Details on the implementation can be found in [40]. It is noteworthy, however, that this approach will fail, if the input projectors are antipodal.

Shape

For the interpolation of the shape, three linear independent invariants are formed of each tensor and interpolated separately. Of the already mentioned K- and R-invariants [24] K_1, R_2, and R_3 will be used (comparable to [23]):

$$K_1 = tr(A), \quad R_2 = \sqrt{\frac{3}{2}}\frac{|A'|}{|A|} \quad \text{and} \quad R_3 = 3\sqrt{6}\frac{det(A')}{|A'|}. \qquad (18)$$

A' is the deviatoric part of A.

Even though K_1 and R_2 are not orthogonal (cf. appendix of [24]), the use of K_1 can be justified by ensuring that the trace of the orientation tensor is one. It is not necessarily essential to have orthogonal invariants for this specific case of application.

The invariants are then interpolated individually: $K_{1ges} = \sum_i w_i K_{1i}$, $R_{2ges} = \sum_i w_i R_{2i}$ and $R_{3ges} = \sum_i w_i R_{3i}$. The weights stay the same: $\sum_i w_i = 1$.

From the interpolated invariants, the following formula was used to calculate the associated eigenvalues (cf. [23]):

For $i = 1, 2, 3$ holds: $\lambda_i = \frac{1}{3}K_1 + \frac{2K_1 R_2}{3\sqrt{3-2R_2^2}} \cos(\frac{\cos^{-1}(R_3)+P_i}{3})$, with $P_i = 0, 2\pi, -2\pi$

With these eigenvalues, Λ can then be created again.

4.2.4. Artificial neural network (ANN) method

The artificial neural network used in this study is based on the idea and implementation of Sabiston et al. [34]. Just like for the other two methods, the goal of the neural network is to determine a FOT for each specified point of the 160 missing positions within the plaque. For the ANN, the nine x, y (and z) coordinates of the given FOT were normalized (divided by 14 since there are 13 rows and columns of FOT). These $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ represent the input data. The output data for the ANN are the respective components of the nine given orientation tensors of second order. Since these components are already between -1 and 1, this data does not need to be normalized. The coordinates were read in as one .csv file as input and the components separated by a_{11}, a_{33}, a_{12}, a_{13} and a_{23} as five separate .csv files as output. There are only five independent components instead of the usual six independent ones for symmetric tensors, since orientation tensors are subject to an additional condition that the trace of the tensor must add up to 1:

$$\sum_{i=1}^{3} a_{ii} = 1. \tag{19}$$

Therefore, only a_{11} and a_{33} were fed into the network as output parameters and a_{22} was determined via $a_{22} = 1 - a_{11} - a_{33}$. The choice was made explicitly, according to the findings of Sabiston et al. [34], to use only one in-plane coordinate and the through-thickness coordinate (a_{22} and a_{33} would have worked analogously as well) in order to reduce the error and to satisfy Eq. (19). This is due to the two in-plane coordinates being significantly larger than the through-thickness coordinate, which in turn meant that a_{11} and a_{22} alone could get above 1 if they were both predicted.

The ANN consists of an input layer, where the normalized input coordinates of the $i = 9$ different points are given and two hidden layers with $n = 48$ neurons. The output of the first hidden layer is the input for the second hidden layer. In the output layer the five independent tensor components $a_{11}, a_{33}, a_{12}, a_{13}$ and a_{23} are predicted for the given $i = 9$ points. The structure of the ANN can be seen in Fig. 2.

The optimizer used is the stochastic gradient descent (SGD) which is an iterative method for optimizing an objective function $Q(w)$ with suitable smoothness properties. Thereby, after choosing an initial vector of parameters w and a learning rate η, two steps are repeated until an approximate minimum is obtained: The samples in the training set are randomly shuffled and $w := w - \eta \nabla Q_i(w)$ is set for $i = 1, 2, \dots, n$ (with Q_i being the summand function typically associated with the ith observation in the data set (used for training)). The loss function chosen is the Mean Absolute Error (MAE), which is defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} = \left| a_{jk(i)} - \hat{a}_{jk(i)} \right|, \tag{20}$$

with n being the number of samples, $a_{jk(i)}$ being the value of the orientation tensor component at the sample location and $\hat{a}_{jk(i)}$ being the predicted value of the orientation tensor component at that sample location. MAE was preferred as error metric over percentage error since many values (especially the off-axis and a_{33} components) are close to zero. Therefore percentage errors tend to become quite large.

Table 2
Parameters of the neural network.

Parameter	Value
Software	Keras 2.4.3 in Python 3.8.5
Optimizer	Stochastic gradient descent (SGD)
Loss function	Mean Absolute Error (MAE)
Layers	2 hidden (plus 1 input and 1 output layer)
Neurons per layer	48
Training data	75%
Validation set	25%
Epochs	100.000

Additionally, outliers seem to be filtered out better by using MAE than by a quadratic error metric like root mean squared error which is more likely to result in overfitting and being biased towards outliers respectively. Additionally, a soft sign activation function is used in the model as it is able to calculate negative numbers and behaves differently in terms of saturation (compared to, e.g., the hyperbolic tangent) because of its smoother asymptotes (polynomial instead of exponential) [41]. However, this of course impacts the amount of epochs required for training as it does not saturate as quickly. The soft sign activation function is given as

$$o = \frac{i}{1 + |i|}, \tag{21}$$

where i is the input to the function and o is the output of the function.

Furthermore, a bias was placed on the loss weights (w) of the outputs of the ANN in order to give more weight to the in-plane orientations. The biases are 0.4 for a_{11}, and 0.15 for all four other components, adding up to 1.

The classical data validation split of 25% is used in the study. The high amount of 100.000 epochs, i.e., times the neural network iteratively trains the weights for each neuron to optimize the outputs from the given input steps, was chosen. While increasing the epochs normally reduces the error, it can also evoke overtraining and leads to longer calculation times. The authors will elaborate on that in the discussion.

All chosen parameters of the ANN are summarized in Table 2. Once the model is trained, a .csv-file with all 160 normalized coordinates – apart from the nine the network was trained with – is given to the trained ANN, to predict the components of the missing FOT.

5. Material

In this study, carbon fiber reinforced polyamide 6 (PA6) is investigated. This material is manufactured in the so-called "long-fiber thermoplastic direct process" introduced by Krause et al. [42]. The schematic process can be seen in Fig. 3.

The LFT-D process is characterized by its inline compounding of the matrix polymer (in this work PA6) in a regular extruder at about 230 °C. The polymer melt is then fed through a film-die into the second extruder which is a twin-screw-device and which is used as mixing unit. The (through a specific nozzle) ejected plastificate is subsequently manually inserted in a Dieffenbacher compression molding press which prevents a reheating of semi-finished products that is necessary in other known LFT production processes. It is then pressed and cooled in the tool which has dimensions of 400 mm × 400 mm and a temperature of about 80 °C –90 °C. The variation of the plastificates in height and size as well as the at times uneven temperature in the tool, which was measured by the authors mid-process, might be important contributors to variations in the flow behavior of the melt and therefore the fiber orientation behavior as a function of position. This aspect is elaborated in the discussion.
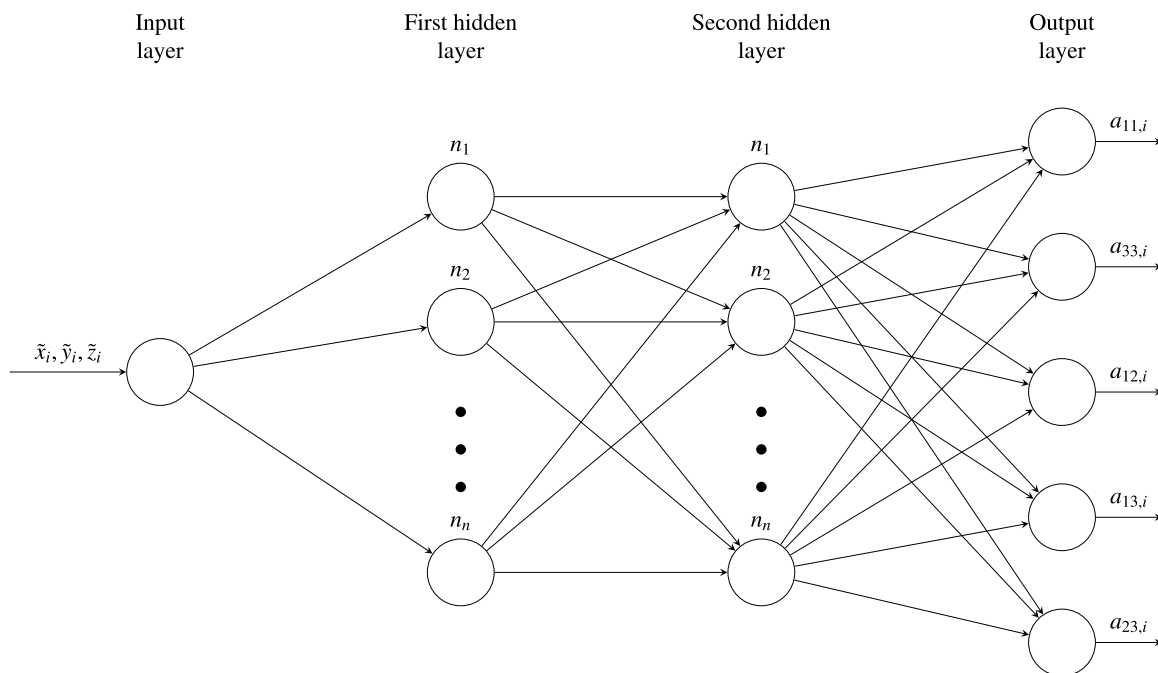
Input
layer

First hidden
layer

Second hidden
layer

Output
layer

n_1

n_2

n_n

n_1

n_2

n_n

$\tilde{x}_i, \tilde{y}_i, \tilde{z}_i$

$a_{11,i}$

$a_{33,i}$

$a_{12,i}$

$a_{13,i}$

$a_{23,i}$

Fig. 2. Schematic concept of the ANN used in this work.

Polymer resin

Modifiers + additives

Carbon/Glass fiber rovings

Compounding
extruder

Mixing extruder (Twin-screw)

LFT charge

Compression molding
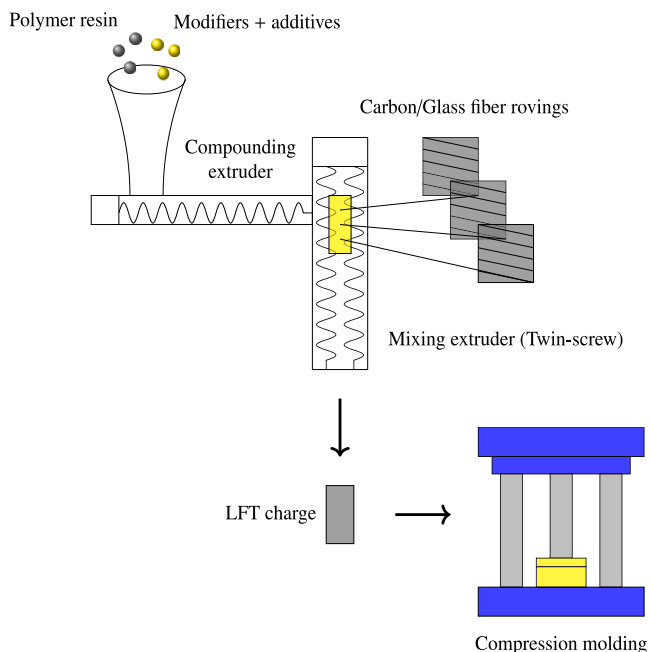
Fig. 3. Schematic representation of the LFT-D process.

Fig. 4. Left: Photo of a plastificate/initial charge that is inserted in the press. Right: Image of one of the plaques with approximate areas of sample withdrawal in transparent orange and the area of insertion of the plastificate/initial charge in the white patterned area.

Out of one of these manufactured plaques nine rectangular samples of 10 mm × 10 mm × 4 mm were cut via waterjet cutting (cf. Fig. 4).

The right-hand coordinate system in the lower left corner of the plate is also valid for all following images with tensor glyphs. It is shifted slightly to the upper right for better visibility, but should actually have its origin at the outer corner of the plate. The dimensions of the small samples resulted from the consideration of two main aspects: How small do the samples have to be in order to allow a sufficiently fine resolution of the fibers and how large do they have to be in order not to cut off the long fibers too much and thus possibly distort the result

of the detectable orientation tensors. This question plays somewhat into the aforementioned conflict point of RVE in the area of fiber-reinforced or generally heterogeneously microstructured materials. The first question was clarified by a series of experiments: several scans with different resolutions and therefore sample dimensions were made and generally compared in their image quality. While single fiber detection (which is required for example for image recognition of fiber length distributions) requires an amount of 3–4 voxels over the fiber thickness and thus in the case of carbon fibers approximately a resolution between 1–2 µm/voxel, the fiber orientation algorithm works on voxel data without any connectivity. Hence, it is not necessary to separate fibers and matrix in a pre-processing step and a resolution of one voxel over the thickness is reasonable. As described above, a resolution of 8.57 µm/voxel was chosen in this work. The second question was answered by experimentally determining the fiber length distribution in the plaque. A peak at about 500 µm and maximum lengths of about
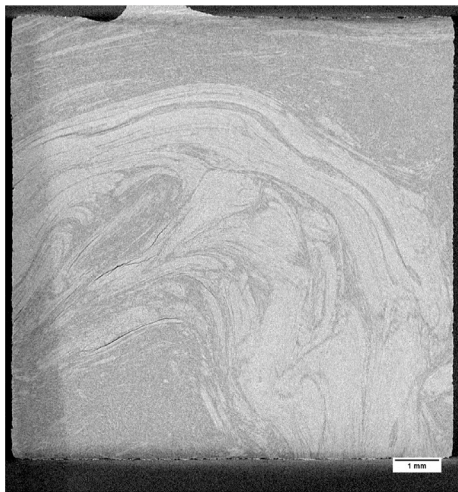
Fig. 5. 2D image taken from the central area (concerning the thickness and thus the z-coordinate) of the CT scan of the UR specimen.

Table 3
Components of the nine measured tensors of the set $\mathcal{T}_m$.

$$UL = \begin{bmatrix} 0.039019 & 0.000565 & 0.001542 \\ 0.000565 & 0.949269 & 0.002303 \\ 0.001542 & 0.002303 & 0.011712 \end{bmatrix} \quad UM = \begin{bmatrix} 0.105635 & -0.093760 & 0.008141 \\ -0.093760 & 0.859305 & -0.104421 \\ 0.008141 & -0.104421 & 0.035060 \end{bmatrix}$$

$$UR = \begin{bmatrix} 0.164933 & 0.140757 & -0.005911 \\ 0.140757 & 0.758623 & -0.119834 \\ -0.005911 & -0.119834 & 0.076445 \end{bmatrix} \quad ML = \begin{bmatrix} 0.091657 & -0.012547 & 0.002165 \\ -0.012547 & 0.892490 & 0.015488 \\ 0.002165 & 0.015488 & 0.015853 \end{bmatrix}$$

$$MM = \begin{bmatrix} 0.657207 & -0.138266 & 0.014728 \\ -0.138266 & 0.278636 & -0.007545 \\ 0.014728 & -0.007545 & 0.064157 \end{bmatrix} \quad MR = \begin{bmatrix} 0.248761 & 0.196508 & -0.028304 \\ 0.196508 & 0.673319 & -0.099772 \\ -0.028304 & -0.099772 & 0.077920 \end{bmatrix}$$

$$LL = \begin{bmatrix} 0.073140 & -0.026562 & 0.002841 \\ -0.026562 & 0.908407 & 0.028146 \\ 0.002841 & 0.028146 & 0.018454 \end{bmatrix} \quad LM = \begin{bmatrix} 0.812019 & -0.053106 & -0.003032 \\ -0.053106 & 0.147797 & -0.022115 \\ -0.003032 & -0.022115 & 0.040184 \end{bmatrix}$$

$$LR = \begin{bmatrix} 0.398303 & 0.118596 & 0.006867 \\ 0.118596 & 0.521223 & -0.015272 \\ 0.006867 & -0.015272 & 0.080475 \end{bmatrix}$$

15 mm resulted in a weight-averaged length of 1.66 mm and a value-averaged length of 0.4 mm. The sample area of 10×10 mm^2 could thus be well justified, since the fewest fibers would be cut off.

6. Results

6.1. Fiber orientation tensors

First, µCT scans of the nine samples were made as described above. An example 2D image from the center of such a scan (in this case of the sample UR) is shown in Fig. 5.

Then, one FOT per sample was determined from these scans using the structure tensor method described above. The result of the structure tensor based evaluation of the µCT scans resulted in the following FOT for the set $\mathcal{T}_m = \{\mathbf{UL}, \mathbf{UM}, \mathbf{UR}, \mathbf{ML}, \mathbf{MM}, \mathbf{MR}, \mathbf{LL}, \mathbf{LM}, \mathbf{LR}\}$:

The nine tensors, which are displayed in Table 3, are therefore the basis of all further calculations. They are only one possible example of FOD of a single plate, but they represent a rather extreme case of fiber orientation in such a plate, which is why the different behavior of the individual interpolation methods can be shown relatively well. In Fig. 6, these tensors can be seen represented as tensor glyphs at their respective positions in the plate.

The tensors in the areas in between (marked with a question mark) are to be determined from the nine given by means of the different methods.

It should be mentioned that the measured tensors were determined over the complete sample volume and were thus also averaged over

Fig. 6. Visualization of the nine measured (blue) fiber orientation tensors via tensor glyphs pictured at the specific position of the underlying sample.

the thickness. However, especially over the thickness, the tensor components still change strongly even in such a small volume. In Fig. 7, where the principal components of all fiber orientations tensors are plotted over the thickness, one can see strongly heterogeneous courses.

In particular, the change of orientation in the edge layers can be detected in all tensors. In the left and upper tensors, the a_{22} component is extremely dominant across the thickness, but it decreases rapidly towards the first and last slices. The tensors show there an increase either in the value of the a_{11} or the a_{33} component. One can see inverted behavior for the $\mathbf{MM}$ and $\mathbf{LM}$ tensor, where the a_{11} component is dominant throughout the sample with an increase of the a_{33} component at the borders.

This edge phenomenon, known in compression molding as the "shell-core effect", results from the rapid freezing of the hot fiber-polymer mass (about 230 °C) at the "cold" (about 80 °C –90 °C) mold wall, while the central region longer remains fluid. This central layer flows forward in a so-called swelling flow from the center into the not yet filled mold area.

Fig. 8 explains, why at the position of $\mathbf{ML}$, the a_{33} component does not increase at the edges of the height of the plate. The tensor $\mathbf{ML}$ was determined from a specimen located in the area where the plastificate is inserted (cf. Fig. 4). In the µCT scan of the plastificate in Fig. 8, the upper and lower region show pronounced orientation in x direction (hence "11-direction"), which is transferred onto the plate as the mass freezes directly with the closing mold in these areas (the plastificate is much thicker than the final plaque). Therefore, for $\mathbf{ML}$, the a_{11} component increases at the borders (cf. Fig. 7.

On the contrary, one can see the increase in the a_{33} component for $\mathbf{MM}$ and $\mathbf{MR}$ in Fig. 7, which results from the mass flowing upwards and downwards at the edges of the height (Fig. 8). For the $\mathbf{MR}$ position, this occurs especially pronounced as the mold ends and the movement of the mass in both the x- and the y-direction stops, resulting in increased flow towards both positive and negative z-direction from the center of the mold.

6.2. Component averaging method

The set of measured orientation tensors $\mathcal{T}_m$ via CT scan and subsequent calculation via structure tensor is represented by the blue tensor
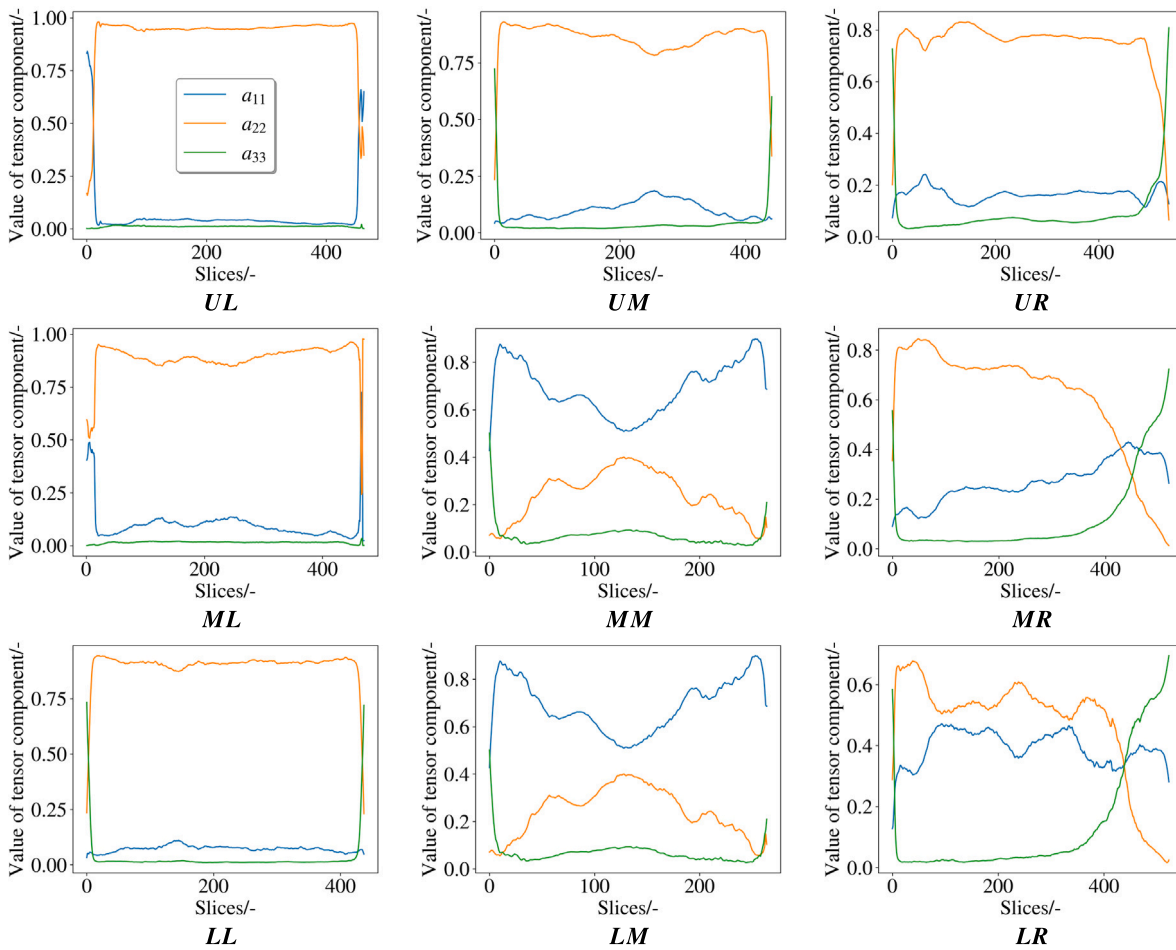
Fig. 7. Visualization of the main diagonal components of the tensor set $\mathcal{T}_m$ over the thickness of the specimens.
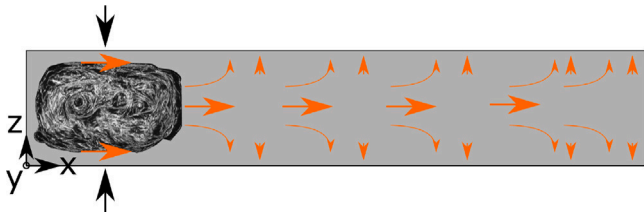

Fig. 8. Schematic x-z section at about $y = 200$ mm (where **ML** is located) through the plate with sketched swelling flow explaining the behavior of the a_{33} component over the thickness at the different positions in Fig. 7. A real µCT image of one of the plastificates used in the compression molding process can be seen at the left.

glyphs in Fig. 9, the set of interpolated tensors $\mathcal{T}_i = \{T_{xy} \ \forall \ x \in 1, \dots, 13 \cap y \in 1, \dots, 13\}$ by the orange tensor glyphs.

The origin of the global coordinate system is located in the lower left corner of the plaque. The original LFT charge covered almost the entire left side of the 400×400 mm^2 mold with a width of about $x = 90$ mm (to the right), a length of about $y = 350$ mm (up) and a height of about $z = 60$ mm. Thus, when the press closes, one would expect a quasi 1D flow to the right.

However, in Fig. 9 a clear progression can be seen in the measured (blue) fiber orientation.

After a clear preferred direction in the left region resulting from the plastificate, i.e., from the last extrusion step in the LFT-D process, a turn to a rather dominant orientation in the x-axis seems to happen towards the middle of the plaque (apart from the top region). At the right side another turn to an again more y-direction-dominant

orientation happens (apart from the lowest tensor glyph, which in general seems to be more isotropic than the other tensors). Considering these measured tensors, the interpolated tensors should follow some kind of curve. In fact, the interpolation does not seem to cover this behavior smoothly but instead rather accomplished the orientation changes through rounding the tensors. Following the literature, this behavior was expected (cf. swelling effect in the State of Art (chapter 2) and can be confirmed.

In order to be able to approach quantitative error analyses and to better assess the interpolation behavior, one measured tensor of $\mathcal{T}_m$ was omitted in each case and instead also determined with the interpolation method. The visualization results are shown in the nine lower pictures in Fig. 10.

To obtain a quantitative error value, the Frobenius norm of the measured tensors and their respective interpolated substitutes was formed. The result of the difference between the Frobenius norm of the interpolated and the original tensor can be seen as an error map in Fig. 11. The method seems to predict the **MM**, **LM** and **LL** tensors the worst. It is difficult to judge whether the Frobenius norm is suitable as a quantitative assessment, but it will be discussed further in the discussion.

Therefore, a third possibility of error analysis is considered, namely the direct component comparison between interpolated and measured tensors. Fig. 12 shows this for the component averaging method.

The nine graphs in Fig. 12 correlate to the nine different tensor components of a 3×3 tensor. Each graph shows the component value of the measured tensors in blue and of the interpolated tensors in orange on the y-axis for each of the nine measured tensors (depicted on the x-axis, starting with **UL**). While the differences are largest for
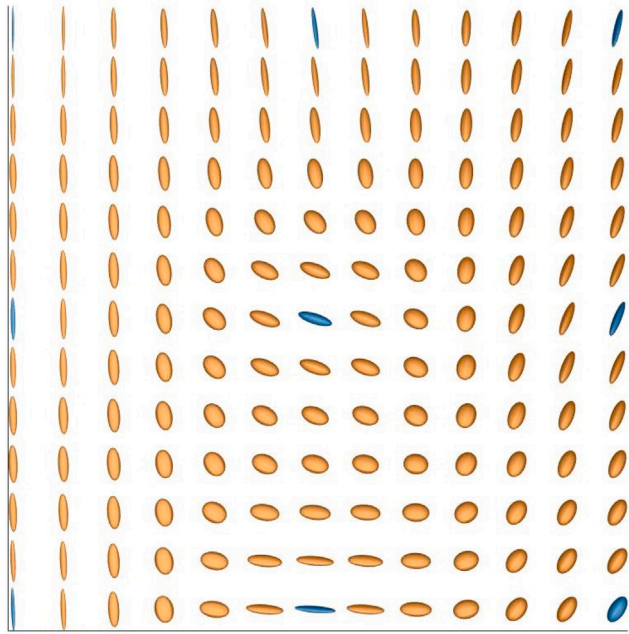
Fig. 9. Visualization of interpolated (orange) and measured (blue) tensors when using the component averaging interpolation method described in this paper.

Fig. 10. The graphic shows the measured (blue) and interpolated (orange) fiber orientation tensor glyphs when leaving one measured tensor out of the calculation and interpolating it instead with the component averaging method respectively.

the components with the largest values and fluctuations (mainly a_{11} and a_{22}), the character of the component averaging method is also clearly evident in this type of error analysis. Thus, the components of the interpolated tensors almost resemble a moving average of the components of the measured tensors.
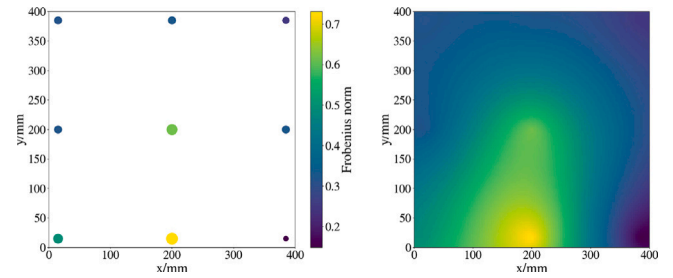
Fig. 11. Visualization of the discontinuous (left image) and continuous (right image) error of the component averaging method across the plaque of the interpolated tensor in comparison to the measured one when leaving this specific tensor out of the computation. Value determined via Frobenius norm.
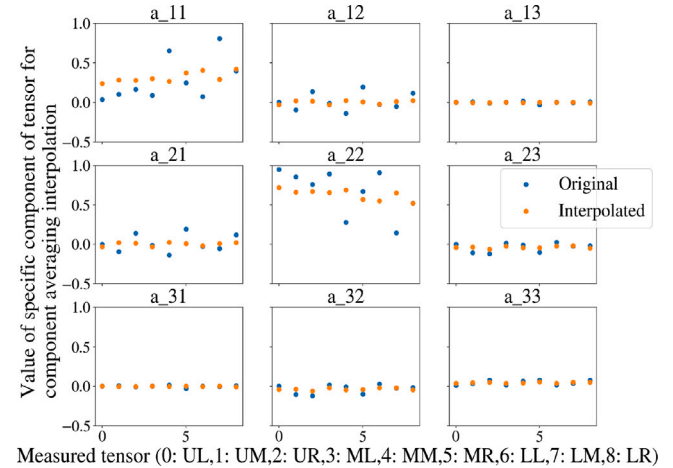
Fig. 12. Comparison of each of the nine components of each of the nine measured tensors with the corresponding resulting component of the interpolated tensors for the component averaging method.

6.3. Decomposition methods

Quaternion-based

In Fig. 13, the results of the interpolation with the quaternion-based decomposition approach can be seen.

The before-mentioned progress of orientations can be visually traced as a clear curve. As for the interpolation method as such, the visual results are for the most part very appealing. Interpolation between the individual measured FOT is good and the transition between two adjacent tensors also appears reasonable. The anisotropy is not basically lost between two differently oriented tensors by "rounding the tensor". The rotation of two adjacent tensors occurs with small angles and therefore smoothly. The only exception to this can be seen at the upper right edge: The interpolated tensor $T_{10,13}$ in the middle of UM and UR behaves somewhat strangely as far as the behavior of the row is concerned. Instead of closing the estimated angle of 20° between UM and UR by a piece wise change of 10°, the tensor $T_{10,13}$ is oriented in an angle deviating by around 80° compared to the measured ones. However, the tensor MM is for example also taken into account for the calculation of this tensor (just like all the other measured ones of the set $\mathcal{T}_m$), even if weighted less strongly than UM and UR, which favors the big rotation of the tensors in the uppermost row considering the global orientation behavior. Furthermore, the behavior in this column looks much better than could be expected if the tensor had rotated in mathematically negative (clockwise) direction around the z-axis than the chosen positive (anti-clockwise) direction. When leaving measured tensors out of the "ground truth" and interpolating them instead, there are definite changes in the orientation course, which can be seen in Fig. 14.
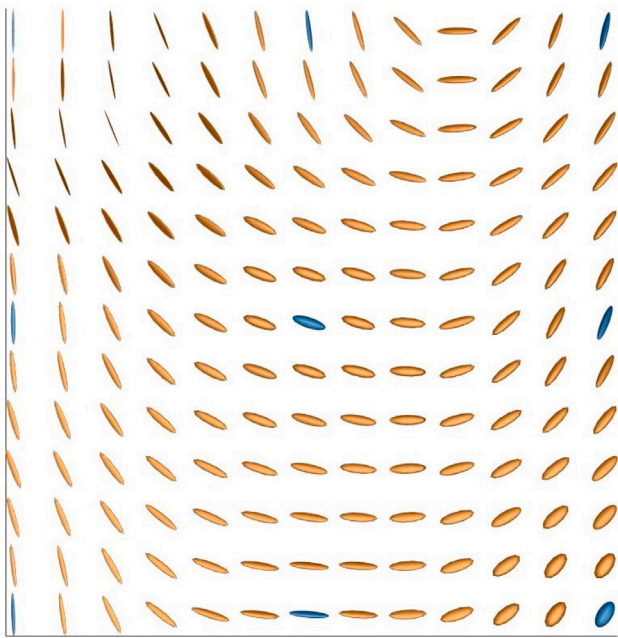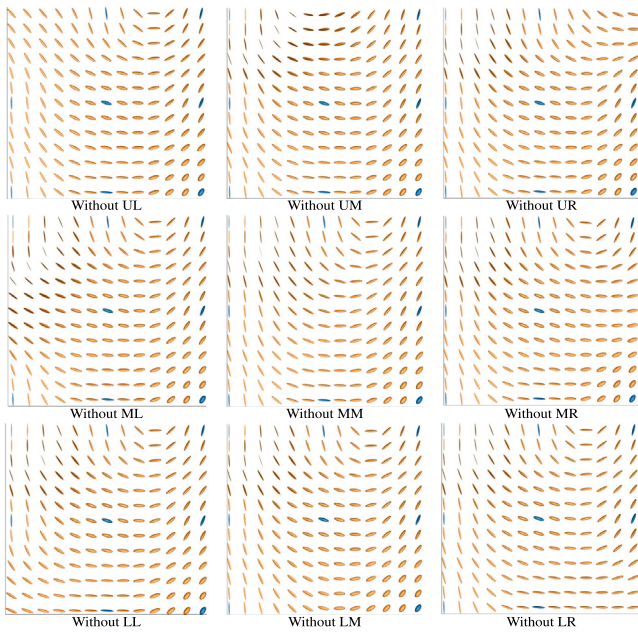
Fig. 13. Visualization of interpolated (orange) and measured (blue) tensors when using the decomposition-based interpolation method described in this paper.

Fig. 15. Visualization of the discontinuous (left image) and continuous (right image) error of the decomposition method across the plaque of the interpolated tensor in comparison to the measured one when leaving this specific tensor out of the computation. Value determined via Frobenius norm.

Fig. 16. Comparison of each of the nine components of each of the nine measured tensors with the corresponding resulting component of the interpolated tensors for the decomposition method.

Fig. 14. The graphic shows the measured (blue) and interpolated (orange) fiber orientation tensor glyphs when leaving one measured tensor out of the calculation and interpolating it instead with the decomposition method respectively.

For example, the behavior of the afore-mentioned $T_{10,13}$ changes significantly when UM or UR are omitted. In general, however, it must be stated that the orientation courses react very agilely and sensibly to the changes when individual tensors are omitted when using the decomposition method.

The quantitative evaluation based on the Frobenius norm is visualized in Fig. 15. The rather poorer interpolation at the left and upper edges and the relatively good performance in the middle of the plaque (and lower right) are noticeable.

Considering the component-wise deviations displayed in Fig. 16, it is striking that they are considerably high in this specific case for this
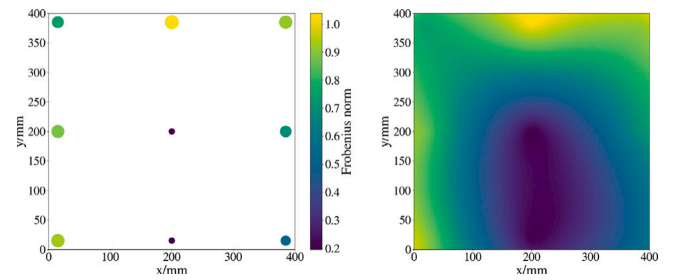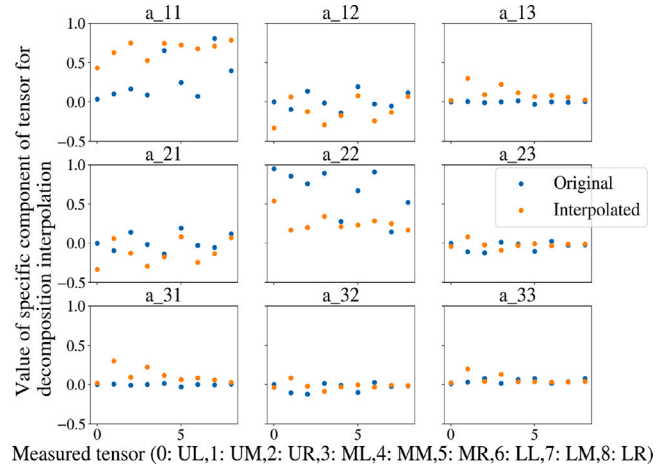
coordinate system. However, it is noticeable that major trends between the different tensors are mostly preserved by this method (cf., e.g., a_{11}).

Projector-based

For the sake of completeness, projector-based interpolation has been applied to the experimental data. The according results are displayed in Fig. 17.

Analogously to the quaternion-based approach, tensor shape is interpolated monotonously. Thus, the interpolation does not yield results with larger isotropy than included in the data set. However, significant deviations can be observed regarding tensor orientation. While the row-wise "flipping" behavior appears to be resolved (compare first and last row of 16 and 17), a singularity right next to the domain's midpoint occurs. As the implementation of the projector-based interpolation is based upon iterative methods, it cannot be vectorized. Therefore, the authors decided to neglect this method and concentrate on the quaternion-based approach for all further investigations in this work.

6.4. ANN method

The results of the tensor field when interpolating with the neural network can be seen in Fig. 18.

It strikes that when training this network with the measured tensors, it is able to produce both very anisotropic and very isotropic tensors at the points with missing tensors, compared to the other two main methods that rather dispensed one or the other. While some areas look smooth like, e.g., the upper and right area, there is non-monotonous interpolation behavior concerning, e.g., MM and LL where the surrounding tensors are much more isotropic than the measured one and
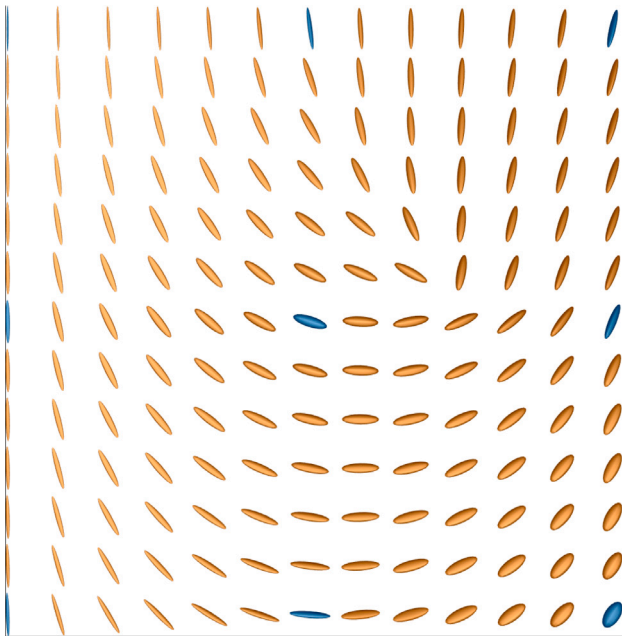
Fig. 17. Visualization of interpolated (orange) and measured (blue) tensors when using the projector-based interpolation method described by Krauß and Kärger [25].
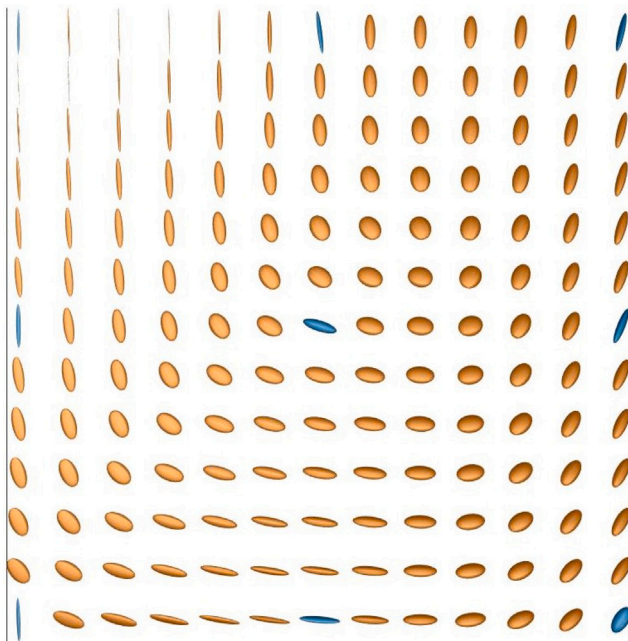

Fig. 18. Visualization of interpolated (orange) and measured (blue) tensors when using the ANN-based interpolation method described in this paper.

also quite differently oriented (looking specifically at the tensor LL). When training the network multiple times with the same input data, the results look very much alike, speaking for the robustness of the method.

However, looking at the plots of the tensor fields when leaving measured tensors out of the input data (Fig. 19), some of the non-monotonous behavior shows again (cf., e.g., the fields without MM and without MR).

Consequently, the difference of the Frobenius norm for these two cases is significantly large and so is the error for the interpolation of the UM tensor, which is all depicted in Fig. 20.
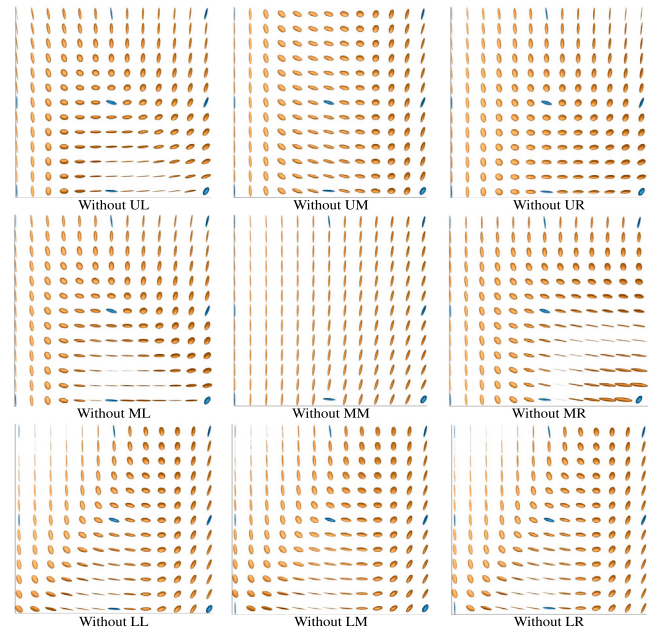

Fig. 19. The graphic shows the measured (blue) and interpolated (orange) fiber orientation tensor glyphs when leaving one measured tensor out of the calculation and interpolating it instead with the neural network respectively.
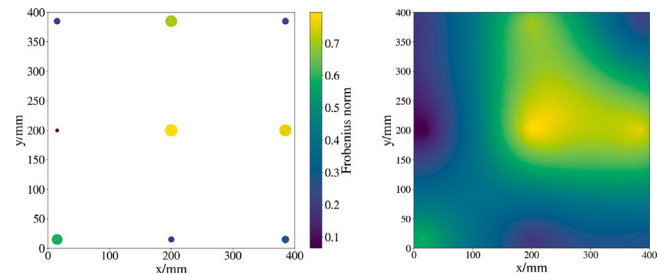

Fig. 20. Visualization of the discontinuous (left image) and continuous (right image) error of the ANN method across the plaque of the interpolated tensor in comparison to the measured one when leaving this specific tensor out of the computation. Value determined via Frobenius norm.

Considering the component-wise errors in Fig. 21, the results are still comparatively good. The largest deviations occur especially for the three mentioned tensors before. Just as was the case for the other two methods, the biggest deviations appear mostly for the a_{11} and a_{22} components. The comparison between the three methods regarding the differences between the measured and interpolated tensor components will be taken up again in subchapter 7.2.

The quality of the neural network must also be considered with respect to the progression of an error measure over the number of epochs. As an error measure, the already mentioned MAE was used and the course over the epochs can be seen in Fig. 22. As expected with the small number of training data, the network does not behave optimally. However, the graphs show both overfitting (a_{23}, a_{33}, a_{31}) and underfitting (a_{11}) trends. Thus, it is difficult to draw a general conclusion. Overfitting rather argues for using more training data or stopping at a lower number of epochs, apart from solutions that require more specific methods which are very dependent on the model. Underfitting can be combated by different measures depending on the cause of the underfitting; either more epochs (unlikely here) or more parameters of the model can help, or a change to a completely different model.
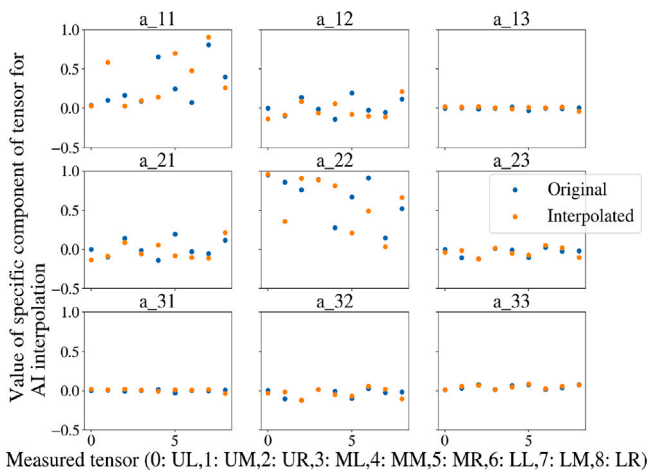
Fig. 21. Comparison of each of the nine components of each of the nine measured tensors with the corresponding resulting component of the interpolated tensors for the ANN method.
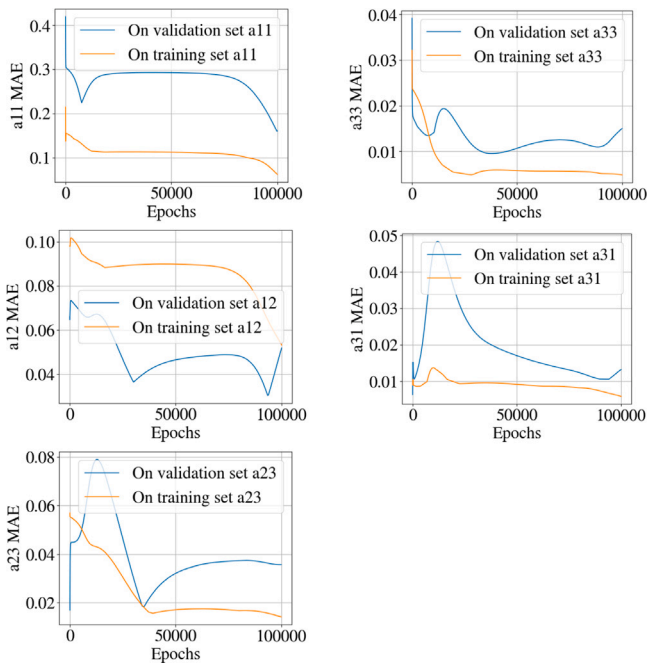

Fig. 22. MAE over training epochs of the ANN per tensor component.

However, more training data can also help with underfitting, which is most likely in the case considered.

7. Discussion

7.1. Curve in fiber orientation

Although this work only shows the fiber orientation field of one plaque, scans and also tensile tests in different angles of further plates hint towards a slight drift of the mass to the top too. Uneven height or dimensions of the plastificate and uneven temperatures in the tool could explain the curve in the orientation tensor field. The authors distinctively measured slight variations in both temperature of the mold and geometry of the plastificate. Furthermore, the initial charge is inserted in the mold manually, resulting in a possibly skewed/angular position of the latter when the press closes. However, the authors assume the main reason for the drift to be the ejection of the initial
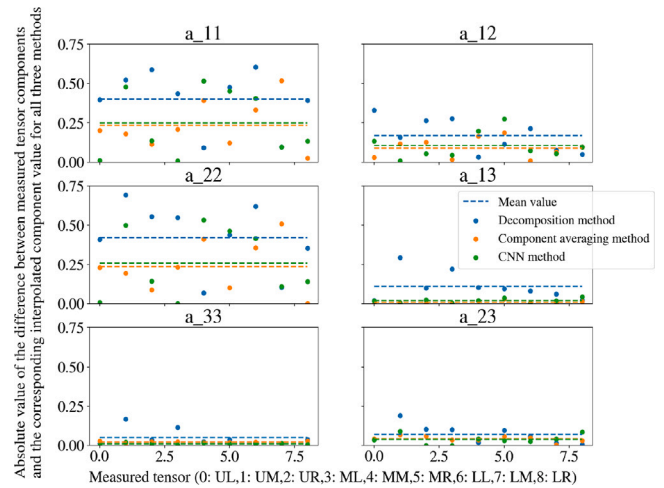

Fig. 23. Comparison of all methods concerning the absolute difference between the same component of the measured and corresponding interpolated tensor.

charge from the extruder. As this process takes a few seconds, the "old", first ejected plastificate has been cooling down for longer than the part that leaves the extruder at the end. The colder part is located in the upper part of the plaque (higher y value in the coordinate system given in this work). As a result, the lower part of the fiber–matrix-mass should be less viscous and better flowable, presumably causing the flow front to advance faster there and hence eventuating in the curve.

7.2. Comparison of the three methods

Looking at the deviations of the measured and calculated components for all three methods in Fig. 23, it is noticeable that the component averaging method actually performs best for this particular case.

However, the question remains whether an averaging can physically correspond to the same idea as that of an interpolation. After all, the search is not for the "mean" value, but for the "middle" of two given tensors. The already mentioned difficulties of this method concerning the shape of the interpolated orientation tensors rather pleads against this method, but the frequency of use and the simplicity of implementation and calculation are of course advantages.

The neural network is also very close to the component averaging method in terms of absolute deviations. The neural network is explicitly trained on the components of the measured tensors and tries to predict them, so this is not surprising.

The decomposition method performs worst with respect to the value deviations of the components.

Table 4 sums up the mean average absolute error of the different methods.

However, it has to be stated that the retroactive computation of originally given tensors does not correspond exactly to the task of the determination of tensors between given ones. The former works with an input data set reduced by 1/9. In particular, when leaving out one of the four corner nodes, rather the extrapolation instead of the interpolation behavior of the respective method is demanded and judged as the calculated point is located outside of the support point grid. In fact, when calculating the average absolute error of the different methods without including the prediction of the corner tensors, the mean average error gets significantly smaller as can be seen in Table 5.

The deviation to the quantitative error when involving all nine tensors is the highest for the decomposition method. This could lead to the assumption that the decomposition method performs so poorly quantitatively because this error assessment is not appropriate for the

Table 4
Deviation of calculated value of a specific component for a specific method averaged over all nine tensors.

Component	Average absolute error of different methods		
	CA	D	AI
a_{11}	0.233	0.401	0.248
a_{12}	0.09	0.169	0.105
a_{13}	0.009	0.112	0.02
a_{22}	0.236	0.421	0.258
a_{23}	0.043	0.071	0.04
a_{33}	0.021	0.05	0.011
MAE	0.105	0.204	0.114

Table 5
Deviation of calculated value of a specific component for a specific method averaged over the five tensors of $\mathscr{T}_m$ without the corner nodes (UL, UR, LL, LR).

Component	Average absolute error of different methods		
	CA	D	AI
a_{11}	0.158	0.181	0.172
a_{12}	0.061	0.073	0.065
a_{13}	0.007	0.087	0.01
a_{22}	0.161	0.206	0.179
a_{23}	0.022	0.047	0.023
a_{33}	0.01	0.039	0.008
MAE	0.07	0.106	0.076
Deviation to error with all tensors	0.035	0.098	0.038

task. The visual impression, previous literature [25] as well as mathematical accuracy clearly speak for the superiority of the decomposition approach. This is supported by the fact that the decomposition method performs significantly better without the extrapolation task (omitting corner points). Additionally, the reduction of the input tensors is a change of the initial situation and only slightly resembles the actual interpolation task.

So, judging the methods in different ways is necessary. While concerning the component deviations the decomposition method performed the worst out of the three methods, the visually agile interpolation behavior in the tensor field has already been mentioned. The authors conducted a study, where the measured tensors were expressed in different coordinate systems (rotated by different angles). For these new tensor components, it was remarkable how well the decomposition method interpolation adapted itself. Furthermore, the monotonous behavior concerning anisotropy is preferable compared to the other methods. In addition, the authors highly assume that the decomposition method might perform better when the tensors are located further in the center of the plate/part, since the decomposition method interpolated worse at the edges. This will be taken up again in the outlook.

The two different versions of the decomposition method differ in the use of projectors instead of invariants. The former have the advantage that they are unique for a given tensor. While this prevents the effect observed at the upper right edge of the plaque of the invariant decomposition method, it shows a singularity close to the domain's midpoint. The visual smoothness of rotation depends on whether one considers the row or the column course of rotation. The question remains whether the measured tensors of this specific plate are generally too inhomogeneous to represent the original fiber flow optimally. After all, the sample sizes of 10×10 mm are small and can also map very local effects. Again, this will be addressed in the Outlook.

Considering adaptability, the ANN provides the significant disadvantage of being trained for a specific grid of measured tensors. So the network trained on those will not perform well for a different plaque. However, this work was precisely about the quick characterization of a single plate or part. For this rather unusual application for a neural network, the possibilities to achieve meaningful results relatively fast with little data should be discussed. With run times of less than one hour for a network with two hidden layers as well as for one with three (which has also been tested by the authors), one can definitely speak of a time saving in comparison to a one and a half hour scan and additional computation time for the generation of a tensor via µCT scan. As far as the component comparison between measured and calculated tensors is concerned, the neural network behaves similarly well as the component averaging method. However, both decomposition method and component averaging are both significantly faster as they generate results near-instantly. The neural network is more suitable for a similar application: Just as in [34], the fiber orientation field can be predicted for one specific, stable process by evaluating multiple plaques via µCT scan and subsequent FOT determination and training the network with a high number of FOT. This provides for good results of a general fiber orientation of a stable process.

In general, the problem of the issue addressed in this paper lies in the combination of the mathematical implementation of orientation and rotation in three-dimensional space and the physical effects of the flow of material. In particular, the assessment of the extent to which an interpolation is visually or quantitatively convincing is also inevitably related to the question of how many mathematical bounds one may give so that the tensor interpolation becomes meaningful, but one does not artificially interpolate in such a way that it "looks better".

While neural networks usually ignore any implementation of physical effects (except for physics-informed neural networks (PINNs)) and learn only from previous data, the other two analytical methods also do not consider boundary conditions (such as the location of the initial charge in the die in the press), although this would be theoretically possible. On the contrary, these are also purely mathematical considerations of tensor interpolation, since a large intervention in the interpolation ought to be avoided and the main focus was on the given data. The aim was to find a relatively easy to implement and relatively well-performing method for fast interpolation of given orientation tensors of small number to avoid costly scan times.

In fact, Brannon even argues in chapter 14 of her book "Rotation, Reflection, and Frame Changes - Orthogonal tensors in computational engineering mechanics" [43] that methods for mixtures of rotations must be selected appropriate to the physical application. The motivation for this paper therefore lies in finding a reliable tensor interpolation method for this specific application case. For any other application than fiber orientation tensors, this benchmarking study might have to be repeated (even though the decomposition method seems to perform quite well for both diffusion and orientation tensors). In addition to the application, the amount of data or grid points also might influence the functionality of the specific method. Many original algebra-derived methods might work for two given tensors with limited tensors to be interpolated in-between, but perform way worse for many given tensors or many tensor to be calculated. Conversely, many given tensors could certainly improve the result of the methods presented here.

8. Conclusion

In summary, the authors applied four different tensor interpolation methods to fiber orientation tensors. The implemented methods provide for good macroscopic interpolation results for fiber orientation tensor fields interpolating measured FOT determined from microscopic X-ray computed tomography scans. The authors were able to generate orientation fields of plaques in significantly less time than scans for all of these positions would have taken. A particularly interesting aspect is the use of only nine input tensors, showing that the presented approaches work for scarce data. These methods could therefore replace high amounts of resource-consuming μCT scans. Each method has its advantages and disadvantages. The observed monotonous behavior of the decomposition method is particularly satisfactory. When calculating originally measured tensors, the performance of the other two methods were more convincing. The mean absolute error when calculating measured tensors ranges from 0.105 for the component averaging over 0.114 for the neural network to 0.204 for the decomposition method using invariants. However, as this assessment includes an extrapolation task and a changed initial situation, this cannot necessarily be attributed to the quality of the interpolation procedure itself.

9. Outlook

There are multiple possible adjustments of the methods shown in this work. The interpolation behavior of the invariant decomposition method at the top right could be artificially prevented by restricting the possible angle. This could be implemented by comparing the results of the scalar products of the specific quaternions involved and choosing the combination providing the maximum scalar product and hence the smallest possible angle. However, the question arises whether this is not an intervention, since the influence on the interpolation of the tensors in the immediate vicinity would then be significantly higher than given in this work. The authors tried to implement as little artificial restrictions as possible. Furthermore, q_{ges} is not normalized in the invariant decomposition method. And if several unit quaternions are added, the result is no longer a unit quaternion. Analogously, this is true for rotation tensors. Thus, the rotation interpolation as implemented in this work also changes a bit the shape of the tensors. By a normalization a further adjustment could take place here.

However, the most important and perhaps most obvious next step according to the authors – after benchmarking different methods for this interpolation application in this work – is a sensitivity study regarding plaques, sample geometry and sample location. The performance of the methods should definitely be studied for different orientation fields, i.e., different plaques. In a small way this has already been done, but the focus of this paper was rather on the difference between the methods for a specific case. Furthermore, it would be important to take more and also less samples from a plate and evaluate which amount is too little to result in a meaningful orientation field and vice versa from which quantity the error no longer becomes significantly smaller. Analogously, a sensitivity analysis should be performed regarding the sample position in the plate, i.e., which samples at which positions make the result particularly error-prone if they are omitted. Especially the idea of having less of the specimens at the borders of the plaque and more in a central area could be promising, particularly for the invariant decomposition method as the quantitative error seemed to be higher towards the edges of the plaque.

Another approach would be to deal with fourth-order tensors. In this work, only second level tensors were used, which already require a closure, which in principle means a reduction of information. The development or comparison of different methods for the interpolation of fourth level tensors would be highly interesting, but mathematically more complicated.

The ANN could be optimized in terms of the amount of layers, the depth of layers, the learning rate, the batch size, etc. However, as mentioned before, the authors strongly suggest an increase in training data to optimize the network for a specific process, in which similar fiber orientations are to be expected per plate/part instead of using it with this amount of data.

In the further course of the research project, the use of an interpolation method for three-dimensional components is envisaged. The ANN can be used for the 3D case without any changes, when inserting changing z values in the input file instead of constant ones. The component averaging method and the decomposition method can be used for 3D cases basically the same way as well. The only difference appears in the weighting. The inverse distance weighting method that was implemented for this paper, only reads in an x and y value. The procedure can be similarly applied with just one more coordinate value. The rest of the methods can stay the same as the tensors are read in as 3×3 matrices with the possibility to change in any axis. For a more complex tool and component geometry, the extension of the interpolation functionality when dealing with different z-components of the positions of the FOT can then be tested. Alternatively, slicing the FOT through the thickness could provide for more input data and varying z coordinates. This approach has been chosen by Sabiston et al. [34] before.

Another interesting problem would be to investigate the (on average) increased isotropy of the tensors interpolated by the ANN. While this relationship seems logical for the component averaging method, it is so far not explainable for the neural network. A sensitivity analysis of both the ANN parameters and the input files and their effects on the interpolated tensors would be of interest in this context. Under certain circumstances, the additional, irrelevant input of a constant z-value for the 2D plate might be detrimental to the result.

CRediT authorship contribution statement

J. Blarr: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **T. Sabiston:** Conceptualization, Software, Formal analysis, Writing – review & editing. **C. Krauß:** Conceptualization, Software, Validation, Writing – review & editing. **J.K. Bauer:** Consulting, Writing – review & editing. **W.V. Liebig:** Conceptualization, Resources, Writing – review & editing, Supervision. **K. Inal:** Conceptualization, Resources, Supervision. **K.A. Weidenmann:** Conceptualization, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The raw and processed data (μCT scans of all specimens and FOT of these specimens) as well as the Python scripts for all three methods required to reproduce the above findings have been published by the authors [44] and are available to download via the following DOI: https://doi.org/10.5445/IR/1000153725. This DOI includes all raw data acquired by the authors and all code written by the authors of this paper.

However, the up-to-date code for the component averaging and the decomposition method of second (and soon also fourth) order can also be found in the following github repository: https://github.com/jewelsbla/oriopy. The methods have also been processed into a Python package named *oriopy* which can be accessed, downloaded and locally installed: https://pypi.org/project/oriopy/.

The C++ code to generate the FOT from the μCT scans is available to download from https://sourceforge.net/p/composight/code/HEAD/

tree/trunk/SiOTo/StructureTensorOrientation/FibreOrientation/StructureTensorOrientationFilter.cxx#l186. For further explanation please consider the paper by Pinter et al. cited in this work [35].

The Matlab method for the visualization of the tensors via tensor glyphs is available to download from https://de.mathworks.com/matlabcentral/fileexchange/27462-diffusion-tensor-field-dti-visualization. For more information please consider the paper by Barmpoutis et al. cited in this paper [39].

Acknowledgments

References

[1] P. Eyerer, H. Schüle, Polymer Engineering 1, Springer Berlin Heidelberg, Berlin, Heidelberg, 2020, http://dx.doi.org/10.1007/978-3-662-59837-5.

[2] H. Schürmann, Konstruieren Mit Faser-Kunststoff-Verbunden: Mit 39 Tabellen, second ed., in: VDI-/Buch, Springer, Berlin, 2007.

[3] R. Stewart, Thermoplastic composites — recyclable and fast to process, Reinf. Plast. 55 (3) (2011) 22–28, http://dx.doi.org/10.1016/S0034-3617(11)70073-X.

[4] P.F. Chu, S. Iwasawa, P.L. Schell, C.-Y. Lin, Carbon fiber versus glass fiber reinforcements: A novel, true comparison in thermoplastics, Polym. Compos. 42 (11) (2021) 6173–6181, http://dx.doi.org/10.1002/pc.26294.

[5] A. Trauth, P. Pinter, K. Weidenmann, Investigation of quasi-static and dynamic material properties of a structural sheet molding compound combined with acoustic emission damage analysis, J. Compos. Sci. 1 (2) (2017) 18, http://dx.doi.org/10.3390/jcs1020018.

[6] J. Görthofer, N. Meyer, T.D. Pallicity, L. Schöttl, A. Trauth, M. Schemmann, M. Hohberg, P. Pinter, P. Elsner, F. Henning, A. Hrymak, T. Seelig, K. Weidenmann, L. Kärger, T. Böhlke, Virtual process chain of sheet molding compound: Development, validation and perspectives, Composites 169 (2019) 133–147, http://dx.doi.org/10.1016/j.compositesb.2019.04.001.

[7] S. Galkin, E. Kunze, L. Kärger, R. Böhm, M. Gude, Experimental and numerical determination of the local fiber volume content of unidirectional non-crimp fabrics with forming effects, J. Compos. Sci. 3 (1) (2019) 19, http://dx.doi.org/10.3390/jcs3010019.

[8] M. Teßmann, S. Mohr, S. Gayetskyy, U. Haßler, R. Hanke, G. Greiner, Automatic determination of fiber-length distribution in composite material using 3D CT data, EURASIP J. Adv. Signal Process. 2010 (1) (2010) http://dx.doi.org/10.1155/2010/545030.

[9] D. Salaberger, K.A. Kannappan, J. Kastner, J. Reussner, T. Auinger, Evaluation of computed tomography data from fibre reinforced polymers to determine fibre length distribution, Int. Polym. Process. 26 (3) (2011) 283–291, http://dx.doi.org/10.3139/217.2441.

[10] S.G. Advani, C.L. Tucker, The use of tensors to describe and predict fiber orientation in short fiber composites, J. Rheol. 31 (8) (1987) 751–784, http://dx.doi.org/10.1122/1.549945.

[11] K. Kanatani, Distribution of directional data and fabric tensors, Int. J. Eng. Sci. 22 (2) (1984) 149–164, http://dx.doi.org/10.1016/0020-7225(84)90090-9.

[12] J.K. Bauer, T. Böhlke, Variety of fiber orientation tensors, Math. Mech. Solids (2021) 108128652110576, http://dx.doi.org/10.1177/10812865211057602.

[13] L. Schreyer, J. Blarr, K. Höger, N. Meyer, L. Kärger, Generation of initial fiber orientation states for long fiber reinforced thermoplastic compression molding simulation, 2022, http://dx.doi.org/10.5445/IR/1000155599.

[14] L. Schöttl, K.A. Weidenmann, T. Sabiston, K. Inal, P. Elsner, Fiber bundle tracking method to analyze sheet molding compound microstructure based on computed tomography images, NDT & E Int. 117 (2021) 102370, http://dx.doi.org/10.1016/j.ndteint.2020.102370.

[15] M. Krause, J.M. Hausherr, B. Burgeth, C. Herrmann, W. Krenkel, Determination of the fibre orientation in composites using the structure tensor and local X-ray transform, J. Mater. Sci. 45 (4) (2010) 888–896, http://dx.doi.org/10.1007/s10853-009-4016-4.

[16] V. Kunc, B. Frame, B.N. Nguyen, C.L. Tucker III, G. Velez-Garcia (Eds.), Fiber Length Distribution Measurement for Long Glass and Carbon Fiber Reinforced Injection Molded Thermoplastics, 2007, URL https://experts.illinois.edu/en/publications/fiber-length-distribution-measurement-for-long-glass-and-carbon-f.

[17] R. Maertens, A. Hees, L. Schöttl, W. Liebig, P. Elsner, K.A. Weidenmann, Fiber shortening during injection molding of glass fiber-reinforced phenolic molding compounds: fiber length measurement method development and validation, Polym.-Plastics Technol. Mater. 60 (8) (2021) 872–885, http://dx.doi.org/10.1080/25740881.2020.1867170.

[18] M. Terada, A. Yamanaka, D. Shimamoto, Y. Hotta, K. Shiraki, Y. Kimoto, M. Arai, Carbon fiber sampling method for determining the fiber length distribution, Adv. Compos. Mater. 30 (sup1) (2021) 59–76, http://dx.doi.org/10.1080/09243046.2020.1753635.

[19] K.A. Acton, S.C. Baxter, Characterization of random composite properties based on statistical volume element partitioning, J. Eng. Mech. 144 (2) (2018).

[20] X. Yin, W. Chen, A. To, C. McVeigh, W.K. Liu, Statistical volume element method for predicting microstructure–constitutive property relations, Comput. Methods Appl. Mech. Eng. 197 (43–44) (2008) 3516–3529, http://dx.doi.org/10.1016/j.cma.2008.01.008.

[21] T. Sabiston, K. Inal, P. Lee-Sullivan, Method to determine the required microstructure size to be represented by a second order fibre orientation tensor using X-ray micro computed tomography to evaluate compression moulded composites, Compos. Sci. Technol. 182 (2019) 107777, http://dx.doi.org/10.1016/j.compscitech.2019.107777.

[22] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Log-Euclidean metrics for fast and simple calculus on diffusion tensors, Magn. Reson. Med. 56 (2) (2006) 411–421, http://dx.doi.org/10.1002/mrm.20965.

[23] J.K. Gahm, N. Wisniewski, G. Kindlmann, G.L. Kung, W.S. Klug, A. Garfinkel, D.B. Ennis, Linear invariant tensor interpolation applied to cardiac diffusion tensor MRI, in: Medical Image Computing and Computer-Assisted Intervention: MICCAI (International Conference on Medical Image Computing and Computer-Assisted Intervention), Vol. 15, (Pt 2) 2012, pp. 494–501, http://dx.doi.org/10.1007/978-3-642-33418-4_61.

[24] D.B. Ennis, G. Kindlmann, Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images, Magn. Reson. Med. 55 (1) (2006) 136–146, http://dx.doi.org/10.1002/mrm.20741.

[25] C. Krauß, L. Kärger, Tensor interpolation in virtual manufacturing chains for fiber reinforced composites, Int. J. Mech. Sci. 226 (2022) 107378, http://dx.doi.org/10.1016/j.ijmecsci.2022.107378.

[26] F. Hiai, D. Petz, Riemannian metrics on positive definite matrices related to means, Linear Algebr. Appl. 430 (11–12) (2009) 3105–3130, http://dx.doi.org/10.1016/j.laa.2009.01.025.

[27] J. Blarr, N. Kresin, C. Krauß, K.A. Weidenmann, W.V. Liebig, P. Elsner (Eds.), Application of a tensor interpolation method on the determination of fiber orientation tensors from computed tomography images, in: Composites Meet Sustainability - Proceedings of the 20th European Conference on Composite Materials, Anastasios, Vassilopoulos, 2022, http://dx.doi.org/10.5445/IR/1000155486, URL https://publikationen.bibliothek.kit.edu/1000155486.

[28] S. Jung, A. Schwartzman, D. Groisser, Scaling-rotation distance and interpolation of symmetric positive-definite matrices, SIAM J. Matrix Anal. Appl. 36 (3) (2015) 1180–1201, http://dx.doi.org/10.1137/140967040.

[29] F.L. Markley, Y. Cheng, J.L. Crassidis, Y. Oshman, Averaging quaternions, J. Guid. Control Dyn. 30 (4) (2007) 1193–1197, http://dx.doi.org/10.2514/1.28949.

[30] R. Hartley, J. Trumpf, Y. Dai, H. Li, Rotation averaging, Int. J. Comput. Vis. 103 (3) (2013) 267–305, http://dx.doi.org/10.1007/s11263-012-0601-0.

[31] M. Moakher, Means and averaging in the group of rotations, SIAM J. Matrix Anal. Appl. 24 (1) (2002) 1–16, http://dx.doi.org/10.1137/S0895479801383877.

[32] S. Han, O.A. Bauchau, On the global interpolation of motion, Comput. Methods Appl. Mech. Eng. 337 (2018) 352–386, http://dx.doi.org/10.1016/j.cma.2018.04.002, URL https://www.sciencedirect.com/science/article/pii/S0045782518301725.

[33] J. Köbler, M. Schneider, F. Ospald, H. Andrä, R. Müller, Fiber orientation interpolation for the multiscale analysis of short fiber reinforced composite parts, Comput. Mech. 61 (6) (2018) 729–750, http://dx.doi.org/10.1007/s00466-017-1478-0.

[34] T. Sabiston, K. Inal, P. Lee-Sullivan, Application of Artificial Neural Networks to predict fibre orientation in long fibre compression moulded composite materials, Compos. Sci. Technol. 190 (2020) 108034, http://dx.doi.org/10.1016/j.compscitech.2020.108034.

[35] P. Pinter, S. Dietrich, B. Bertram, L. Kehrer, P. Elsner, K.A. Weidenmann, Comparison and error estimation of 3D fibre orientation analysis of computed tomography image data for fibre reinforced composites, NDT & E Int. 95 (2018) 26–35, http://dx.doi.org/10.1016/j.ndteint.2018.01.001.

[36] L.A. Feldkamp, L.C. Davis, J.W. Kress, Practical cone-beam algorithm, J. Opt. Soc. Am. A 1 (6) (1984) 612–619.

[37] G. Kindlmann (Ed.), Superquadric Tensor Glyphs, The Eurographics Association, 2004, http://dx.doi.org/10.2312/VISSYM/VISSYM04/147-154.

[38] T. Schultz, G.L. Kindlmann, Superquadric glyphs for symmetric second-order tensors, IEEE Trans. Vis. Comput. Graphics 16 (6) (2010) 1595–1604, http://dx.doi.org/10.1109/TVCG.2010.199.

[39] A. Barmpoutis, B.C. Vemuri, T.M. Shepherd, J.R. Forder, Tensor splines for interpolation and approximation of DT-MRI with applications to segmentation of isolated rat hippocampi, IEEE Trans. Med. Imaging 26 (11) (2007) 1537–1546, http://dx.doi.org/10.1109/TMI.2007.903195.

[40] J.K. Gahm, D. Ennis (Eds.), Dyadic Tensor-Based Interpolation of Tensor Orientation: Application to Cardiac DT-MRI, 2013, http://dx.doi.org/10.1007/978-3-642-54268-8_16.

[41] X. Glorot, Y. Bengio (Eds.), Understanding the Difficulty of Training Deep Feedforward Neural Networks, (9) JMLR: W&CP 9, 2010.

[42] W. Krause, F. Henning, S. Tröster, O. Geiger, P. Eyerer, LFT-D — A process technology for large scale production of fiber reinforced thermoplastic components, J. Thermoplast. Compos. Mater. 16 (4) (2003) 289–302, http://dx.doi.org/10.1177/0892705703016004001.

[43] R.M. Brannon, Rotation, Reflection, and Frame Changes, 2053-2563, IOP Publishing, 2018, http://dx.doi.org/10.1088/978-0-7503-1454-1.

[44] J. Blarr, FOT interpolation code and scans: 3D μCT images of specimens of carbon fiber reinforced polyamide 6 plaque, fiber orientation tensor data of these images, and three Python code files for two different algebraic and one machine learning based tensor interpolation algorithms, 2022, http://dx.doi.org/10.5445/IR/1000153725.