Open-Source Framework for Modeling the Evolution of Fiber Orientation

Bruno Ramoa, Ricardo Costa, Francisco Chinesta, João Miguel Nóbrega

OpenFOAM® Journal

Volume 5 [Full Papers], Pages 17–37
ISSN: 2753-8168

An Open-Source Framework for Modeling the Evolution of Fiber Orientation

B. Ramoa[1,*], R. Costa[1], F. Chinesta[2], and J.M. Nóbrega[1]

[1]Institute for Polymers and Composites, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
Email address: bruno.ramoa@dep.uminho.pt

[2]ESI Group Chair, PIMM Institute at ENSAM ParisTech, 155 Boulevard de l'Hôpital, F-75013 Paris, France

DOI: https://doi.org/10.51560/ofj.v5.131
Results with version(s): OpenFOAM®v2212
Repository: https://github.com/Computational-Rheology/paper_OFJ_FiberOrientation/tree/main

Abstract. Modeling fiber orientation plays a crucial role in predicting the behavior of fiber reinforced thermoplastic materials. The equation that governs the evolution of the fiber orientation is hyperbolic in nature and requires handling of fourth-order tensors, which are currently unavailable in OpenFOAM®. The current work explores the possibility of using OpenFOAM® and open-source symbolic computation for modeling the evolution of fiber orientation. For this purpose, a `functionObject` was programmed to work as a plug-in for any OpenFOAM® incompressible flow solver and compute the evolution of fiber orientation in a decoupled manner. Several fiber orientation models and closure relations available in the literature were implemented in the tool, which were verified by comparing their predictions with independent results for a single material point obtained by numerically integrating the associated governing equations.

1. Introduction

Thermoplastic materials are tied to our daily lives, encompassing a wide range of applications, from household products to advanced technical appliances. To meet the permanent demand for enhanced products, fiber-reinforced thermoplastic materials (FRTM) are often employed to achieve specific properties (property weighted by the density) that cannot be obtained with conventional materials. Industries such as automotive heavily rely on this technology to achieve lightweight designs, essential for optimizing fuel efficiency [1].

The final properties of thermoplastic parts depend on the microstructure developed during processing. For FRTM, the effects of fiber length, concentration, and orientation play a significant role in the performance of the final part. Particularly, fiber orientation has been extensively studied throughout the last decades [2–6], leading to the development of several phenomenological models [7–13] based on the seminal work of Jeffery [14] to quantify the state of orientation in FRTM parts. During the product development phase, knowledge of this final state of orientation is crucial for structural analysis and design, as it directly affects the parts' mechanical performance [15–17].

In the context of OpenFOAM® only a few works have been dedicated to fiber orientation modeling, where only the Folgar-Tucker (FT) [7] and reduce strain closure (RSC) [8] models in tensor-based formulations have been reported [18–21]. Given the open-source character and outreach of OpenFOAM®, this work presents a novel strategy for modeling the fiber orientation evolution in incompressible fluid flows using the OpenFOAM®framework. The developed plug-in can be utilized for injection molding, compression molding, and any other solver that involves fluid flow. The objective is to incorporate the capability of predicting fiber orientation as an extension, without making modifications to an existing solver. The approach involves creating a `functionObject` to solve the three-dimensional orientation evolution equation with state-of-the-art models and closure approximations in a decoupled manner, using a one-way interaction, where the velocity field affects the fiber orientation evolution but the computed orientation will not influence the relation between the velocity gradient and the stress tensor ($\underline{\tau}$), i.e., the constitutive model, required to calculate the velocity field. Nevertheless, the strategy employed is

* Corresponding author
Received: 21 December 2023, Accepted: 21 June 2024, Published: 20 January 2025

flexible enough for the interested user to extend the developed classes to account for the effect of fiber orientation into the stress tensor calculation [22].

The remaining paper is organized as follows: Sections 2 and 3 introduce, respectively, the fiber orientation modeling approaches and the state-of-the-art models together with the closure approximations used in commercial applications. In Section 4, the proposed approach to implement fiber orientation modeling within the OpenFOAM®framework is presented. Sections 6 and 7 showcase and discuss case studies involving simple-shear and the flow in a center-gated disk, tested to validate the numerical implementations. Finally, Section 8 summarizes the main conclusions drawn from this work.

2. Fiber orientation modeling with macroscopic descriptors

The employed formulation considers rigid, axi-symmetric, straight fibers, such as glass fibers typically used to manufacture FRTM [9], whose orientation characterization can be carried out at different scales. At the micro-scale, the orientation of individual fibers can be used as a complete descriptor. This can be done through a pair of θ and ϕ angles or by a unit vector $\underline{p}$ that is aligned with the fiber axis, as illustrated in Fig. 1. Both descriptions are related by:

$$\underline{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}. \tag{1}$$

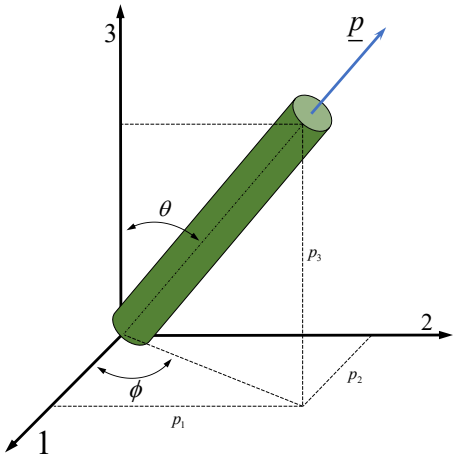


Figure 1. Schematic of the orientation of a single fiber.

The motion of a single, rigid, ellipsoidal fiber in a Newtonian fluid was formulated by Jeffery, assuming that the: (i) velocity field was not influenced by the fibers; (ii) inertial and body forces were negligible; (iii) the fluid velocity varied linearly with position and iv) the fluid where the fiber is suspended is incompressible [14,23–25]. Under these conditions and in the absence of additional forcings, the equation for describing the rate-of-change of the fiber orientation is given by [25]:

$$\dot{\underline{p}} = \underline{\underline{W}} \cdot \underline{p} + \xi \left(\underline{\underline{D}} \cdot \underline{p} - \underline{\underline{D}} : \underline{p}\,\underline{p}\,\underline{p} \right), \tag{2}$$

or, equivalently, in tensor notation as [25]:

$$\dot{p}_i = W_{ij} p_j + \xi \left(D_{ij} p_j - D_{jk} p_k p_j p_i \right), \tag{3}$$

where:

- $\dot{\underline{p}}$ is time rate-of-change of $\underline{p}$,
- $\underline{\underline{W}}$ is the vorticity tensor, given by: $\underline{\underline{W}} = \frac{1}{2}\left(\underline{\underline{L}} - \underline{\underline{L}}^{\mathrm{T}}\right)$,
- $\underline{\underline{D}}$ is the rate-of-deformation tensor, given by: $\underline{\underline{D}} = \frac{1}{2}\left(\underline{\underline{L}} + \underline{\underline{L}}^{\mathrm{T}}\right)$,
- $\underline{\underline{L}}$ is the velocity gradient tensor, given by $L_{ij} = \dfrac{\partial v_i}{\partial x_j}$,
- v_i are the components of the velocity vector, $\underline{v}$,
- ξ is the particle shape factor, given by $\xi = \dfrac{a_r^2 - 1}{a_r^2 + 1}$, and

- a_r is the fiber aspect-ratio

Equation (3) reflects the effect of the flow on the fiber motion. The first term on the right-hand side accounts for the influence of rigid-body rotation and the second (between parenthesis) for the effect of fluid deformation on fiber orientation [25]. The term $D_{jk}p_kp_jp_i$ will result in a vector that is parallel to p_i and scaled by $D_{jk}p_kp_j$. This subtracts from $D_{ij}p_j$ the component that is parallel to p_i, thus ensuring that $\dot{p}_i$ is always perpendicular to p_i and that the magnitude of p_i remains constant [25]. It is also known that Eqn. (3) is kinematically reversible. Meaning that the flow field might affect the fiber position and orientation, but if the flow is reversed the fiber will return to its initial position and orientation [26].

In FRTM, it is common to have several thousand fibers per cubic millimeter, consequently, keeping track of each orientation vector is a massive and mostly unnecessary task [25]. Thus, at the mesoscopic scale, statistical methods are used to describe the state of orientation for collections of fibers, through the use of a probability density function $\psi\left(\underline{p}, t\right)$, also known in literature as an orientation distribution function (ODF) [27]. The evolution of the ODF is governed by a Fokker-Planck equation [9, 27], reading:

$$\frac{\mathrm{D}\psi}{\mathrm{D}t} = -\nabla_s \cdot \left(\psi \underline{\dot{p}}\right), \tag{4}$$

where:

- $\dfrac{\mathrm{D}\psi}{\mathrm{D}t}$ is the material derivative of ψ in canonical space,
- ∇_s is the gradient operator evaluated at the unit sphere's surface, and
- $\psi\underline{\dot{p}}$ represents the probability flux.

Equation (4) has several properties, and the reader is referred to references [25, 28, 29] for a more complete description. Since $\underline{p}$ is a unit vector, all possible combinations of (θ, ϕ) or $\underline{p}$ are located on the surface of the unit sphere (conformational space). In this work, only the normalization condition of the ODF is reported. It states that the integral of ψ over the conformational space is unitary [25], mathematically:

$$\begin{aligned} &\oint \psi(\underline{p})\, \mathrm{d}\underline{p} = 1 \\ &\text{or} \\ &\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \psi(\theta, \phi) \sin(\theta)\, \mathrm{d}\theta\, \mathrm{d}\phi = 1. \end{aligned} \tag{5}$$

Collisions between fibers will result in a diffusion-like behavior [26]. In 1984, Folgar and Tucker [26] proposed a new model based on Jeffery's work to account for fiber interactions by adding a phenomenological diffusive flux $\underline{q}$, which reads [9]:

$$\frac{\mathrm{D}\psi}{\mathrm{D}t} = -\nabla_s \cdot \left(\psi \underline{\dot{p}} + \underline{q}\right), \tag{6}$$

with:

$$\underline{q} = -D_r \nabla_s \psi, \tag{7}$$

where D_r is a rotary diffusivity. The authors proposed a scalar rotary diffusivity, which has since been known in the literature as the isotropic rotary diffusion (IRD) model and is given by:

$$D_r = C_I \dot{\gamma}, \tag{8}$$

where:

- C_I is a phenomenological parameter modeling the randomizing effect of fiber-fiber interactions, and
- $\dot{\gamma}$ is the magnitude of the rate-of-deformation tensor, given by $\dot{\gamma} = \sqrt{2\underline{\underline{D}} : \underline{\underline{D}}}$.

The ODF offers keen insights as an orientation descriptor, however, its usage for computer-assisted engineering (CAE) in three-dimensional simulations is computationally expensive. To minimize this difficulty, a macroscopic orientation descriptor was proposed by Advani and Tucker [7] in 1987, where the information contained in the ODF was condensed into a tensor-based formulation, by operating on the moments of the ODF. The most commonly used macro-descriptor is defined as [9, 27]:

$$\underline{\underline{A}} = \oint \underline{p}\,\underline{p}\,\psi\left(\underline{p}\right)\,\mathrm{d}\underline{p}$$
$$\text{or} \tag{9}$$
$$A_{ij} = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} p_i p_j\,\psi(\theta,\phi)\sin\left(\theta\right)\mathrm{d}\theta\,\mathrm{d}\phi.$$

This second-order tensor is symmetric and has a unitary trace. The tensor-based formulation of the model proposed by Folgar and Tucker (FT), Eqn. (6), reads [9]:

$$\dot{\underline{\underline{A}}} = \dot{\underline{\underline{A}}}^{\mathrm{H}} + \dot{\underline{\underline{A}}}^{\mathrm{IRD}}, \tag{10}$$

with:

$$\dot{\underline{\underline{A}}}^{\mathrm{H}} = \underline{\underline{W}}\cdot\underline{\underline{A}} - \underline{\underline{A}}\cdot\underline{\underline{W}} + \xi\left(\underline{\underline{D}}\cdot\underline{\underline{A}} + \underline{\underline{A}}\cdot\underline{\underline{D}} - 2\mathbb{A}:\underline{\underline{D}}\right), \tag{11}$$
$$\dot{\underline{\underline{A}}}^{\mathrm{IRD}} = 2C_I\dot{\gamma}\left(\underline{\underline{I}} - 3\underline{\underline{A}}\right), \tag{12}$$

where:

- $\dot{\underline{\underline{A}}}$ is the material derivative of $\underline{\underline{A}}$,
- $\dot{\underline{\underline{A}}}^{\mathrm{H}}$ is the hydrodynamic component of motion that results from Jeffery's model – the first term of the right-hand side of Eqn. (6),
- $\dot{\underline{\underline{A}}}^{\mathrm{IRD}}$ is the isotropic rotary diffusion – the second term of the right-hand side of Eqn. (6),
- $\mathbb{A}$ is a fourth-order orientation tensor, and
- $\underline{\underline{I}}$ is the second-order identity tensor.

Although computationally more efficient than the ODF, the use of orientation tensors have some drawbacks:

- The reconstruction of the orientation distribution function from $\underline{\underline{A}}$ is ambiguous, as $\underline{\underline{A}}$ might not represent a unique orientation state [28]. Figure 2 illustrates different fiber configurations that result in the same orientation tensor.
- In the derivation of the transport equation for $\underline{\underline{A}}$, the next even-order tensor appears. To close the system, a closure relationship is needed, and several authors have proposed solutions to write $A_{ijkl} = f\left(A_{ij}\right)$. In Section 3, the most relevant closure relations used in CAE will be presented.
- When an experimentally determined C_I is unavailable, some authors use semi-empirical expressions [21, 30] based on the product of the fiber volume fraction and the fiber aspect-ratio.

Nevertheless, this descriptor retains the generality of the ODF and provides a compact, computationally efficient and convenient representation of the fiber orientation state, being the state-of-the-art descriptor used in CAE applications.

Over the last four decades several modifications have been proposed to the FT model. Different studies have shown that the orientation kinetics predicted by the FT model evolve much faster than the ones observed experimentally [9]. Accordingly, aiming to achieve a better agreement with the transient experimental observations, Wang et al. [8] proposed a new objective model (independent of the considered coordinate system) designated reduced-strain closure (RSC). This model decomposes $\dot{\underline{\underline{A}}}$ into rate equations for both the eigenvalues (λ_i) and eigenvectors ($\underline{e}_i$). The approach consists in slowing down the evolution for the eigenvalues by a phenomenological scalar parameter ($\kappa \leq 1$), while leaving the rate equations for the eigenvectors unchanged [25]. The RSC model reads:

$$\dot{\underline{\underline{A}}} = \dot{\underline{\underline{A}}}^{\mathrm{RSC}} + \kappa\dot{\underline{\underline{A}}}^{\mathrm{IRD}}, \tag{13}$$

with:

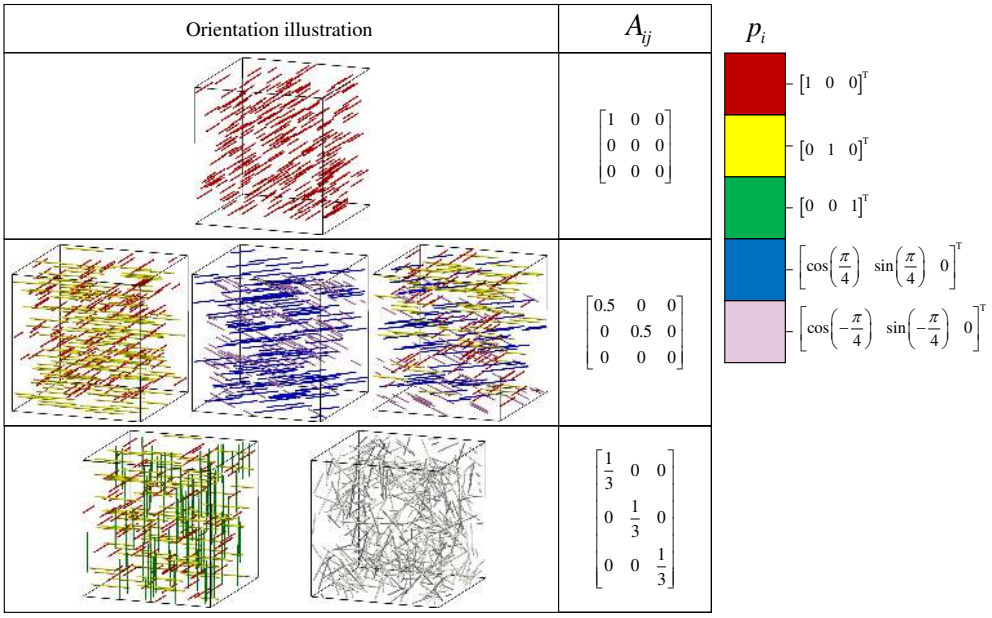


Figure 2. Illustration of fiber orientation states and the corresponding fiber orientation tensor, $\underline{\underline{A}}$.

$$\dot{\underline{\underline{A}}}^{\mathrm{RSC}} = \underline{\underline{W}} \cdot \underline{\underline{A}} - \underline{\underline{A}} \cdot \underline{\underline{W}} + \xi \left(\underline{\underline{D}} \cdot \underline{\underline{A}} + \underline{\underline{A}} \cdot \underline{\underline{D}} - 2 \left[\mathbb{A} + (1-\kappa) \left(\mathbb{L} - \mathbb{M} : \mathbb{A} \right) \right] : \underline{\underline{D}} \right), \tag{14}$$

$$\underline{\underline{A}} = \sum_{i=1}^{3} \lambda_i \left(\underline{e}_i \, \underline{e}_i \right), \tag{15}$$

$$\mathbb{L} = \sum_{i=1}^{3} \lambda_i \left(\underline{e}_i \, \underline{e}_i \, \underline{e}_i \, \underline{e}_i \right), \tag{16}$$

$$\mathbb{M} = \sum_{i=1}^{3} \left(\underline{e}_i \, \underline{e}_i \, \underline{e}_i \, \underline{e}_i \right). \tag{17}$$

The FT and RSC-IRD models work reasonably well for short FRTM [8]. However, when dealing with long FRTM, their accuracy reduces, as longer fibers tend to align less in the flow direction [9]. To better match experimental data, several authors [9–12] proposed the use of a tensorial rotary diffusion term, to make it direction-dependent, $\underline{\underline{C}}$. The general framework, known as anisotropic rotary diffusion (ARD), was proposed by Phelps and Tucker (PT) [9], and reads:

$$\dot{\underline{\underline{A}}} = \dot{\underline{\underline{A}}}^{\mathrm{H}} + \dot{\underline{\underline{A}}}^{\mathrm{ARD}}, \tag{18}$$

with:

$$\dot{\underline{\underline{A}}}^{\mathrm{ARD}} = \dot{\gamma} \left[2\underline{\underline{C}} - 2\mathrm{tr}\left(\underline{\underline{C}}\right) \underline{\underline{A}} - 5 \left(\underline{\underline{C}} \cdot \underline{\underline{A}} + \underline{\underline{A}} \cdot \underline{\underline{C}} \right) + 10\mathbb{A} : \underline{\underline{C}} \right]. \tag{19}$$

Phelps and Tucker [9] proposed $\underline{\underline{C}} = f\left(\underline{\underline{A}}, \underline{\underline{D}}\right)$ following a second-order polynomial as:

$$\underline{\underline{C}} = b_1 \underline{\underline{I}} + b_2 \underline{\underline{A}} + b_3 \underline{\underline{A}}^2 + \frac{b_4}{\dot{\gamma}} \underline{\underline{D}} + \frac{b_5}{\dot{\gamma}^2} \underline{\underline{D}}^2, \tag{20}$$

where the dimensionless parameters, b_i, $i = 1, \ldots, 5$, are obtained by fitting experimental data. However, it is known that certain values of b_i lead to unstable and non-physical solutions [25]. Additionally, this model simplifies to the IRD counterpart if b_1 is equal to C_I, and $b_{2\ldots5}$ to zero. To encompass the need of

matching transient results, the authors linked the ARD framework with the RSC methodology, creating the ARD-RSC model, which is presently available in the proprietary software AutoDesk MoldFlow® [31].

Due to numerical stability problems and fitting dependency requirements of the rotary diffusion tensor proposed by Phelps and Tucker, Tseng and co-workers [10, 32] proposed an alternative model, the improved ARD (iARD), which the authors combined with a new formulation for slowing down the orientation kinetics, known as the retarding principal rate (RPR). Although with a different approach than the one used in the RSC, in essence, the RPR also aims at slowing down the evolution of the eigenvalues without modifying the eigenvectors rate-of-change. Their fiber orientation model reads:

$$\dot{\underline{\underline{A}}} = \dot{\underline{\underline{A}}}^{\mathrm{H}} + \dot{\underline{\underline{A}}}^{\mathrm{ARD}} + \dot{\underline{\underline{A}}}^{\mathrm{RPR}}, \tag{21}$$

with:

$$\underline{\underline{C}}^{\mathrm{iARD}} = C_I^{\mathrm{iARD}} \left(\underline{\underline{I}} - C_M \tilde{\underline{\underline{L}}} \right), \tag{22}$$

$$\tilde{\underline{\underline{L}}} = \frac{\underline{\underline{D}}^2}{\left\| \underline{\underline{D}}^2 \right\|}, \tag{23}$$

$$\dot{\underline{\underline{A}}}^{\mathrm{RPR}} = -\underline{\underline{R}} \cdot \dot{\underline{\underline{\Lambda}}}^{\mathrm{IOK}} \cdot \underline{\underline{R}}^{\mathrm{T}}, \tag{24}$$

$$\dot{\underline{\underline{\Lambda}}}_{ii}^{\mathrm{IOK}} = \alpha \left[\dot{\lambda}_i - \beta \left(\dot{\lambda}_i^2 + 2 \dot{\lambda}_j \dot{\lambda}_k \right) \right], \qquad i,j,k \in \{1,2,3\}, \tag{25}$$

where:

- C_M is a scalar parameter representing the fiber-matrix interaction,
- $\left\| \underline{\underline{D}}^2 \right\|$ is the magnitude of $\underline{\underline{D}}^2$, given by $\left\| \underline{\underline{D}}^2 \right\| = \sqrt{\frac{1}{2} \underline{\underline{D}}^2 : \underline{\underline{D}}^2}$,
- $\dot{\underline{\underline{\Lambda}}}^{\mathrm{IOK}}$ is the material derivative of a diagonal tensor calculated under the intrinsic orientation kinetics (IOK) assumption [32],
- λ_i is the i^{th} eigenvalue of $\underline{\underline{A}}$, ordered as: $\lambda_1 \geq \lambda_2 \geq \lambda_3$,
- $\underline{\underline{R}}$ is an orthogonal rotation matrix whose columns are the eigenvectors of $\underline{\underline{A}}$ ($\underline{\underline{R}} = [\underline{e}_1, \underline{e}_3, \underline{e}_3]$),
- $\underline{\underline{R}}^{\mathrm{T}}$ is the transpose of $\underline{\underline{R}}$, and
- α and β are phenomenological scaling factors used to slow-down the eigenvalues evolution.

The proposed rotary diffusion tensor can be viewed as a special case of Eqn. (20) where only the first and last terms are used [25]. The iARD-RPR model reduces to the FT model by specifying $\alpha = 0$ and $C_M = 0$. Additionally, the RSC and RPR models are identical by setting $\alpha = 1 - \kappa$ and $\beta = 0$ [10]. This model is currently available in the proprietary injection molding simulation software Moldex3D® [33].

Further alternatives to model $\underline{\underline{C}}$ were made in 2018 by Tseng et al. [11] and Bakharev et al. [12], who proposed a similar fiber orientation model, where the rotary diffusion tensor was considered to be coaxial with $\underline{\underline{A}}$. Tseng et al. [11] proposed the principal ARD (pARD), in which the rotary diffusion tensor is defined as [27]:

$$\underline{\underline{C}}^{\mathrm{pARD}} = C_I^{\mathrm{pARD}} \left(\underline{e}_1 \, \underline{e}_1 + \Omega \, \underline{e}_2 \, \underline{e}_2 + (1 - \Omega) \, \underline{e}_3 \, \underline{e}_3 \right), \tag{26}$$

where $0.5 \leq \Omega \leq 1$ is a fitted scalar parameter. Bakharev et al. [12] proposed the Moldflow rotational diffusion model (MRD), where $\underline{\underline{C}}$ is defined as [27]:

$$\underline{\underline{C}}^{\mathrm{MRD}} = C_I^{\mathrm{MRD}} \left(D_1 \, \underline{e}_1 \, \underline{e}_1 + D_2 \, \underline{e}_2 \, \underline{e}_2 + D_3 \, \underline{e}_3 \, \underline{e}_3 \right), \tag{27}$$

where D_1, D_2 and D_3 are non-negative scalar quantities.

Both $\underline{\underline{C}}^{\mathrm{pARD}}$ and $\underline{\underline{C}}^{\mathrm{MRD}}$ have a similar formulation. However, $\underline{\underline{C}}^{\mathrm{pARD}}$ is less flexible and cannot be reduced to the FT model whilst the $\underline{\underline{C}}^{\mathrm{MRD}}$ formulation recovers the FT model by setting all parameters equal to unity. The report of Bakharev et al. [12] provides suitable choices for the model parameters to achieve the best correlation with experimental data.

Other fiber orientation models were proposed in the literature, such as Wang's two constant model [25], Kugler et al. [13] flow dependent model, MRD/pARD with RSC formulation for slowing fiber orientation kinetics from Kugler et al. [34] and Kech et al. [35], and even data-driven approaches [36, 37]. The interested reader is referred to [25, 27, 29] for a comprehensive literature review of the most relevant associated topics.

3. Closure models

As pointed out in Section 2, one of the drawbacks of using the tensor-based approach as a macro-descriptor of fiber orientation is the need to calculate the next even-ordered orientation tensor, which results from the derivation of the evolution equation of the lower order tensor. Breuer et al. [28] identified in literature 17 closures to describe A_{ijkl} and 4 to identify a sixth-order tensor. However, in commercial applications, only the hybrid, the invariant-based optimal fitting (IBOF), or an orthotropic fitted closure are typically used [33, 38].

The hybrid closure was referred in the review of Advani and Tucker [39] as being a blend between a quadratic and a linear closures. For a 3D case, the model reads:

$$A_{ijkl}^{\text{hyb}} = f A_{ijkl}^{\text{quad}} + (1-f)\, A_{ijkl}^{\text{lin}}, \tag{28}$$

where:

- f is the blending factor, which is usually calculated as $f = 1 - 27 \det(A_{ij})$ [39],
- $A_{ijkl}^{\text{quad}} = A_{ij}A_{kl}$, and
- $A_{ijkl}^{\text{lin}} = -\frac{1}{35}(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \frac{1}{7}(A_{ij}\delta_{kl} + A_{ik}\delta_{jl} + A_{il}\delta_{jk} + A_{jl}\delta_{ik} + A_{jk}\delta_{il} + A_{kl}\delta_{ij})$, with δ_{ij} being the second-order identity tensor.

The quadratic closure component is known to be exact when the fibers are perfectly aligned in a single direction, while the linear closure component is known to be exact for isotropic orientation [39]. This blended closure is used due to its small computational cost, when compared to other alternatives. However, it is known to over-predict flow-induced fiber orientation [40–42].

The eigenvalue-based version of the natural closure (ORE) [43] is an orthotropic eigenvalue-based fitted closure. The second-order tensor is decomposed into its eigenvalues and corresponding eigenvectors, with λ_i sorted in descending order, $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Due to the normalization condition, Eqn. (5), the summation of the eigenvalues of $\underline{\underline{A}}$ should equal unity, consequently, only two eigenvalues are independent, typically λ_1 and λ_2.

Consider $\hat{\mathbb{A}}$ to be a fourth-order tensor defined in the principal axis of $\underline{\underline{A}}$. Due to the orthotropic character, this tensor can be written with contracted notation having 12 independent parameters as [40]:

$$\hat{\mathbb{A}} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} & \hat{A}_{13} & 0 & 0 & 0 \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{23} & 0 & 0 & 0 \\ \hat{A}_{31} & \hat{A}_{32} & \hat{A}_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{A}_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & \hat{A}_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{A}_{66} \end{bmatrix}. \tag{29}$$

Assuming a complete symmetry of the fourth-order tensor, there are six independent variables as:

$$\begin{aligned} \hat{A}_{23} &= \hat{A}_{32} = \hat{A}_{44}, \\ \hat{A}_{13} &= \hat{A}_{32} = \hat{A}_{55}, \\ \hat{A}_{12} &= \hat{A}_{21} = \hat{A}_{66}. \end{aligned} \tag{30}$$

Moreover, taking into account that the higher order tensor, A_{ijkl}, includes information from the lower order tensor, and since $A_{ijkk} = A_{ij}$ [25, 29], the number of independent components can be further reduced to 3, by:

$$\begin{aligned} \hat{A}_{11} + \hat{A}_{12} + \hat{A}_{13} &= \lambda_1, \\ \hat{A}_{21} + \hat{A}_{22} + \hat{A}_{23} &= \lambda_2, \\ \hat{A}_{31} + \hat{A}_{32} + \hat{A}_{33} &= \lambda_3, \end{aligned} \tag{31}$$

and

$$\begin{aligned} \hat{A}_{11} &= \mathrm{f}_1(\lambda_1, \lambda_2), \\ \hat{A}_{22} &= \mathrm{f}_2(\lambda_1, \lambda_2), \\ \hat{A}_{33} &= \mathrm{f}_3(\lambda_1, \lambda_2), \end{aligned} \tag{32}$$

where f_1, f_2, and f_3 are functions of the independent eigenvalues of $\underline{\underline{A}}$. For the ORE closure, these are defined as:

$$\begin{aligned}\mathrm{f}_i(\lambda_1,\lambda_2) =& C_{i_1} + C_{i_2}\lambda_1 + C_{i_3}\lambda_2 + C_{i_4}\lambda_1\lambda_2 + C_{i_5}\lambda_1^2 + C_{i_6}\lambda_2^2 \\ &+ C_{i_7}\lambda_1^2\lambda_2 + C_{i_8}\lambda_1\lambda_2^2 + C_{i_9}\lambda_1^3 + C_{i_{10}}\lambda_2^3 + C_{i_{11}}\lambda_1^2\lambda_2^2 \qquad i \in \{1,2,3\}. \\ &+ C_{i_{12}}\lambda_1^3\lambda_2 + C_{i_{13}}\lambda_1\lambda_2^3 + C_{i_{14}}\lambda_1^4 + C_{i_{15}}\lambda_2^4,\end{aligned} \tag{33}$$

The coefficients $C_{i1},\ldots,C_{i15}$ are reported in Verweyst's PhD thesis [43], but are also available in [44].

Considering the normalization condition, the entries for $\hat{A}_{44}$, $\hat{A}_{55}$ and $\hat{A}_{66}$ can be computed as [43]:

$$\begin{aligned}\hat{A}_{44} &= \frac{1}{2}\left(1 - 2\lambda_1 + \hat{A}_{11} - \hat{A}_{22} - \hat{A}_{33}\right), \\ \hat{A}_{55} &= \frac{1}{2}\left(1 - 2\lambda_2 - \hat{A}_{11} + \hat{A}_{22} - \hat{A}_{33}\right), \\ \hat{A}_{66} &= \frac{1}{2}\left(-1 + 2\lambda_1 + 2\lambda_2 - \hat{A}_{11} - \hat{A}_{22} + \hat{A}_{33}\right).\end{aligned} \tag{34}$$

Once the components of $\hat{\mathbb{A}}$ have been calculated, the tensor is rotated to the case study reference frame by:

$$A_{ijkl} = R_{im}R_{jn}R_{ko}R_{lp}\hat{A}_{mnop}. \tag{35}$$

Another alternative is the IBOF closure approximation, proposed by Chung and Kown [41]. This closure does not require the eigendecomposition of $\underline{\underline{A}}$ and, instead, resorts to the invariants of the second-order orientation tensor. The IBOF closure reads [41]:

$$\begin{aligned}A_{ijkl} =& \beta_1 \mathrm{S}\left(\delta_{ij}\delta_{kl}\right) + \beta_2 \mathrm{S}\left(\delta_{ij}A_{kl}\right) + \beta_3 \mathrm{S}\left(A_{ij}A_{kl}\right) + \\ & \beta_4 \mathrm{S}\left(\delta_{ij}A_{km}A_{ml}\right) + \beta_5 \mathrm{S}\left(A_{ij}A_{km}A_{ml}\right) + \beta_6 \mathrm{S}\left(A_{im}A_{mj}A_{kn}A_{nl}\right),\end{aligned} \tag{36}$$

where the function S gives the fully symmetric part of its fourth-order tensor argument:

$$\begin{aligned}\mathrm{S}\left(T_{ijkl}\right) = \frac{1}{24}[\ & T_{ijkl} + T_{ijlk} + T_{ikjl} + T_{iklj} + T_{iljk} + T_{ilkj} \\ & + T_{jikl} + T_{jilk} + T_{jkil} + T_{jkli} + T_{jlik} + T_{jlki} \\ & + T_{kijl} + T_{kilj} + T_{kjil} + T_{kjli} + T_{klij} + T_{klji} \\ & + T_{lijk} + T_{likj} + T_{ljik} + T_{ljki} + T_{lkij} + T_{lkji}]\,.\end{aligned} \tag{37}$$

The six β_i coefficients are functions of the second (I_2) and third (I_3) invariants of $\underline{\underline{A}}$, given by:

$$I_2 = \frac{1}{2}\left[\mathrm{tr}\left(\underline{\underline{A}}\right)^2 - \mathrm{tr}\left(\underline{\underline{A}}^2\right)\right], \tag{38}$$

$$I_3 = \det\left(\underline{\underline{A}}\right). \tag{39}$$

Additionally, only three coefficients are independent, β_3, β_4 and β_6, whose values are given by:

$$\begin{aligned}\beta_i =& a_{i1} + a_{i2}I_2 + a_{i3}I_2^2 + a_{i4}I_3 + a_{i5}I_3^2 + a_{i6}I_2I_3 + a_{i7}I_2^2I_3 \\ &+ a_{i8}I_2I_3^2 + a_{i9}I_2^3 + a_{i10}I_3^3 + a_{i11}I_2^3I_3 + a_{i12}I_2^2I_3^2 \\ &+ a_{i13}I_2I_3^3 + a_{i14}I_2^4 + a_{i15}I_3^4 + a_{i16}I_2^4I_3 + a_{i17}I_2^3I_3^2 \qquad i \in \{3,4,6\}. \\ &+ a_{i18}I_2^2I_3^3 + a_{i19}I_2I_3^4 + a_{i20}I_2^5 + a_{i21}I_3^5,\end{aligned} \tag{40}$$

where the fitting coefficients $a_{i1},\ldots,a_{i21}$ are provided in the Appendix of [41]. The remaining coefficients are formulated as [41]:

$$\begin{aligned}\beta_1 =& \frac{3}{5}\left[-\frac{1}{7} + \frac{1}{5}\beta_3\left(\frac{1}{7} + \frac{4}{7}I_2 + \frac{8}{3}I_3\right) - \beta_4\left(\frac{1}{5} - \frac{8}{15}I_2 - \frac{14}{15}I_3\right)\right. \\ & \left. -\beta_6\left(\frac{1}{35} - \frac{24}{105}I_3 - \frac{4}{35}I_2 + \frac{16}{15}I_2I_3 + \frac{8}{35}I_2^2\right)\right],\end{aligned} \tag{41}$$

$$\beta_2 = \frac{6}{7}\left[1 - \frac{1}{5}\beta_3\left(1 + 4I_2\right) + \frac{7}{5}\beta_4\left(\frac{1}{6} - I_2\right) - \beta_6\left(-\frac{1}{5} + \frac{2}{3}I_3 + \frac{4}{5}I_2 - \frac{8}{5}I_2^2\right)\right], \tag{42}$$

$$\beta_5 = -\frac{4}{5}\beta_3 - \frac{7}{5}\beta_4 - \frac{6}{5}\beta_6\left(1 - \frac{4}{3}I_2\right). \tag{43}$$

Due to the reduced number of operations compared to eigen-fitted closures, the IBOF closure is considered as computationally more efficient, while maintaining a similar level of accuracy [41].

4. Modeling fiber orientation within the OpenFOAM framework

Tensor-based fiber orientation modeling is a pure advective problem, hence the formulations presented above are better suited for a Lagrangian approach [25]. To calculate the fiber orientation evolution equation in an Eulerian framework, the following definition of material derivative can be employed:

$$\dot{\underline{\underline{A}}} = \frac{\mathrm{D}\underline{\underline{A}}}{\mathrm{D}t} = \frac{\partial \underline{\underline{A}}}{\partial t} + \underline{v} \cdot \nabla \underline{\underline{A}} = \frac{\partial \underline{\underline{A}}}{\partial t} + \nabla \cdot \left(\underline{v}\, \underline{\underline{A}}\right) - \left(\nabla \cdot \underline{v}\right) \underline{\underline{A}}. \tag{44}$$

Due to the incompressibility assumption, $(\nabla \cdot \underline{v})\, \underline{\underline{A}}$ is null. The equation that governs the evolution of $\underline{\underline{A}}$ is a component-wise passive scalar equation, coupled by a source term given by the right-hand side of the fiber orientation model. In order to introduce fiber orientation modeling into OpenFOAM®, some considerations have to be made. In Section 2, the velocity gradient was defined as $L_{ij} = \dfrac{\partial v_i}{\partial x_j}$, however in OpenFOAM®, $\nabla v_{ij} = L_{ji} = \dfrac{\partial v_j}{\partial x_i}$, which requires the adaptation of the the vorticity tensor, $\underline{\underline{W}}$, whose definition in the OpenFOAM®framework is the anti-symmetric of the one defined in Section 2 (Eqn. (3)).

Currently, OpenFOAM®does not provide a fourth-order tensor class, and only the *extend* branch offers a simplified fourth-order tensor class, with nine independent components and a limited set of operations. However, to be able to calculate the evolution of fiber orientation, it is necessary to know the product of $D_{kl}A_{ijkl}$. Therefore, A_{ijkl} must be computed based on a closure relation, as described in Section 3.One potential solution for addressing this issue is to implement a comprehensive class to handle the operations. Alternatively, recognizing that $D_{kl}A_{ijkl}$ is a second-order tensor, it can be introduced as a source term in the evolution equation, and symbolic computation can be used to infer the required operations and generate the necessary code. In this work the latter approach was employed.

The framework for fiber orientation modeling was developed considering two abstract classes, `fiberOrientationModel` and `closureModel`. These were developed following the structure of the viscosity models available in `$FOAM_SRC/transportModels/incompressible/viscosityModels`, with run-time selection. For each class the required minimum attributes and member functions were added to model fiber orientation. Accordingly, the `fiberOrientationModel` class contains a dictionary for collecting input data from the user, fields that are common to all models ($\underline{\underline{A}}$, $\underline{\underline{L}}$, $\underline{\underline{D}}$, $\underline{\underline{W}}$, $\dot{\gamma}$, and $\mathbb{A} : \underline{\underline{D}}$), a pure virtual `solve` method for assembling and solving the system of linear equations and some other minor methods to assure its functionality. The `closureModel` class includes one pure virtual function named `computeClosure` to compute the product $D_{kl}A_{ijkl}$ and one virtual function named `computeRSCClosure` to compute $\left[A_{ijkl} + (1-\kappa)\left(L_{ijkl} - M_{ijmn}A_{mnkl}\right)\right] D_{kl}$, required by the RSC model (Eqn. (14)), together with additional minor functionalities.

Based on the above structure, each fiber orientation model that derives from the `fiberOrientationModel` class should: (i) specify a sub-dictionary to collect model-specific information and any required additional fields (e.g. $\underline{\underline{C}}$), and (ii) implement the corresponding `solve` method. Additionally, each closure model (hybrid, ORE, IBOF, etc.), which derives from the `closureModel` class, should implement the `computeClosure` method. If one decides to make the closure also available for the RSC approach, the method `computeRSCClosure` should also be defined.

The framework implemented in OpenFOAM®is depicted in Fig. 3. Currently, these classes are wrapped in a `functionObject` (`FO`) to work as a plug-in for any incompressible flow solver. At the end of each specified time step, upon calling the `execute` method from the `FO`, the `solve` function from a specific `fiberOrientationModel` will be executed, assembling and solving the corresponding system of linear equations. On the user's side, inside the `functions` section in the `controlDict` dictionary, in addition to the usual `FO` parameters, it is required to define a `dictionary` named `fiberOrientationProperties` that indicates the fiber model (parameter `model`), fiber parameters (`subdictionary <fiberModel>Properties`), and closure model (parameter `closureModel`).

Following the OpenFOAM®modeling approach, the final equation syntax is similar to the mathematical notation. As an example, the final form of the FT model, Eqn. (10), which is included in `solve()` method, is given in Listing 1. The variable `D_doubleDot_A4_` contains the calculation of $D_{kl}A_{ijkl}$, whose calculation is described in Section 5.

The resultant framework for the implemented fiber orientation modeling should be easily extendable to other fiber orientation and closure models. Currently, due to the validity of the implemented formulations, the available closure models are only valid for 3D cases.

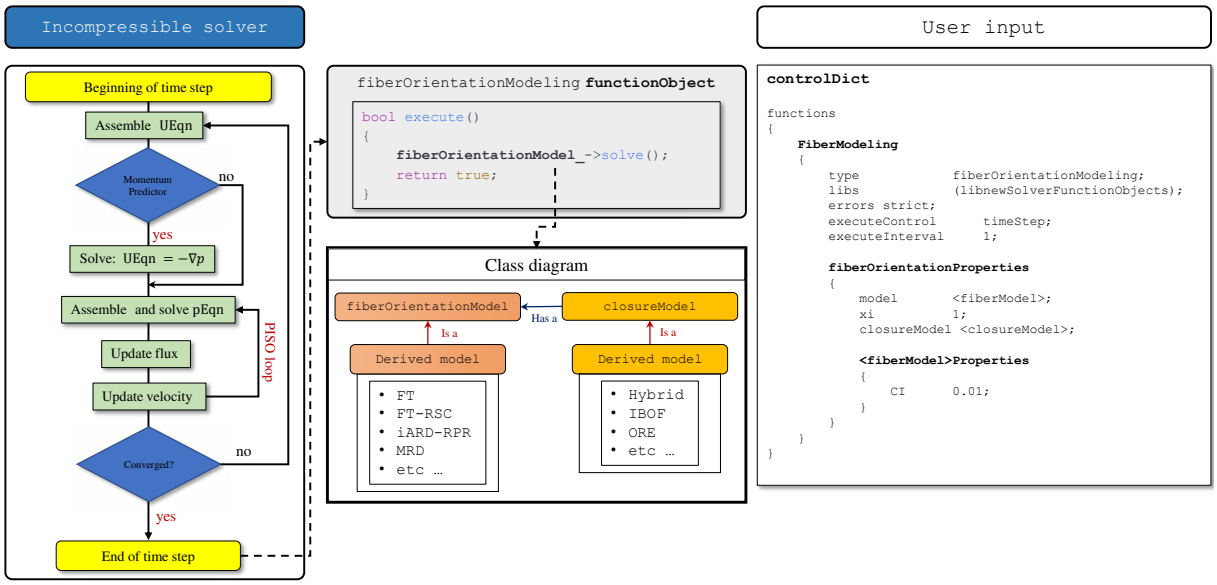


Figure 3. Concept for the `functionObject` for modeling fiber orientation.

```
1  fvSymmTensorMatrix dA2dtEqn
2  (
3      fvm::ddt(A2_) + fvm::div(phi_, A2_, divScheme)
4      ==
5      symm( (W_ & A2_) - (A2_ & W_)
6      +xi_*(
7              (D_ & A2_) + (A2_ & D_) - (2.0*D_doubleDot_A4_)
8           ))
9      + 2.0*CI_*shrRate_*(symmTensor::I-3.0*A2_)
10 );
```

Listing 1. Folgar-Tucker fiber orientation model.

5. Closure model with symbolic computation

The open-source library *Sympy* v1.11 [45] was used to perform the operations that OpenFOAM®does not have, namely the contractions between fourth- and second-order tensors. This rich-featured library for symbolic computation is written in *Python* language and provides functionalities to convert the symbolic code into *C*-style code, which can be used in OpenFOAM® [46].

In the developed *Python* scripts provided in the repository folder `fiberClosureSymbolic`, a vector is treated as a $[3 \times 1]$ matrix, a second-order tensor as a $[3 \times 3]$ matrix, and a fourth-order tensor as a dense multidimensional array with shape $[3 \times 3 \times 3 \times 3]$.

The *Python* script required to generate the *C*-code that calculates the closure method (`computeClosure`), for the hybrid closure model in the `hybrid` class, is illustrated in Listing 2. In Lines 1–3 of Listing 2, the *Sympy* module is imported with the alias `sym` and the functions `symm`, `fourthOrderIndexPermutation` and `generateCCode` from the module `fiberOrientationModelling_Tools`. These functions are responsible for generating a symmetric second-order tensor, performing index permutations within the fourth-order tensor and generating *C*-code, respectively.

Listing 2 comprises two functions, `computeHybridClosure` and `main`. The former computes the hybrid closure symbolically, while the latter, creates the symbolic variables, performs the tensor contraction and generates the associated *C*-code.

In the `computeHybridClosure` function, a matrix symbol is created representing a second-order tensor (Line 7), symmetry conditions are applied (Line 9), and a second-order identity tensor is defined (Line 11). Finally, in Lines 13–36, the hybrid closure is computed following Eqn. (28), to obtain A_{ijkl}^{hyb}.

In the `main` function, in Lines 41–43, a matrix representing the second-order tensor $\underline{D}$ is created. In Line 45 the `computeHybridClosure` function is executed, returning the computation of the hybrid

```
1  import sympy as sym
2
3  from fiberOrientationModelling_Tools import symm, fourthOrderIndexPermutation,
       ↪ generateCCode
4
5  def computeHybridClosure():
6
7      a = sym.MatrixSymbol('A',3,3)
8
9      A2 = symm(sym.Matrix(a))
10
11     I = sym.eye(3)
12
13     A4_linear = (
14                  sym.Rational(-1,35)*
15                  (
16                       fourthOrderIndexPermutation('ijkl', sym.tensorproduct(I,I))
17                     + fourthOrderIndexPermutation('ikjl', sym.tensorproduct(I,I))
18                     + fourthOrderIndexPermutation('iljk', sym.tensorproduct(I,I))
19                  )
20
21                  + sym.Rational(1,7)*
22                  (
23                       fourthOrderIndexPermutation('ijkl', sym.tensorproduct(A2,I))
24                     + fourthOrderIndexPermutation('ikjl', sym.tensorproduct(A2,I))
25                     + fourthOrderIndexPermutation('iljk', sym.tensorproduct(A2,I))
26                     + fourthOrderIndexPermutation('klij', sym.tensorproduct(A2,I))
27                     + fourthOrderIndexPermutation('jlik', sym.tensorproduct(A2,I))
28                     + fourthOrderIndexPermutation('jkil', sym.tensorproduct(A2,I))
29                  )
30                )
31
32     A4_quadratic =  sym.tensorproduct(A2, A2)
33
34     f = 1.0 - 27*A2.det()
35
36     A4_hybrid = f*A4_quadratic + (1.0 - f)*A4_linear
37
38     return A4_hybrid
39
40 def main():
41     d = sym.MatrixSymbol('D',3,3)
42
43     D = symm(sym.Matrix(d))
44
45     A4 = computeHybridClosure()
46
47     D_doubleDot_A4 = sym.tensorcontraction(sym.tensorproduct(D, A4),(0,4),(1,5))
48
49     generateCCode(sym.Matrix(D_doubleDot_A4))
50
51 if __name__ == "__main__":
52     main()
```

Listing 2. Symbolic computation of the hybrid closure model.

closure, and, in Line 47, the contraction $D_{kl}A_{ijkl}$ is performed. To conclude this function, in Line 49, C-code is generated from the symbolic computations.

The derivation of the C-code to implement the `computeRSCClosure` method in the `Hybrid` class is illustrated in Listings 3 and 4. To compute the RSC closure, a script with a general framework was developed through the `RSC` function. This method takes a closure function as an input argument and performs the required symbolic computations. In Lines 2–8 of Listing 3, the slow-down symbolic variable,

κ, is created as well as variables that will hold the eigenvalues (λ_i) and eigenvectors (e_i) of $\underline{\underline{A}}$. Although not required by the RSC approach, some orientation models and associated closures require the eigenvalues to be sorted in descending order and to have the eigenvectors as column vectors. Within the OpenFOAM®framework, the eigenvalues of a symmetric second-order tensor are computed and sorted in ascending order and the corresponding eigenvectors stored in a row-wise manner. The order of computations within OpenFOAM®was taken into account in the symbolic computations (see Lines 10–21).

In Line 23 the closure relation (the argument of the `RSC` function) is executed to obtain A_{ijkl}. Place holders for tensors $\mathbb{M}$ (Line 25) and $\mathbb{L}$ (Line 26) are created and the respective expressions are formulated (Lines 28–35) to get the expression between square brackets of the RSC model, Eqn. (14).

```
1  def RSC(closureFunc):
2      k = sym.Symbol('k')
3
4      e = sym.MatrixSymbol('eigenVector',3,3)
5      eigVec = sym.Matrix(e)
6
7      w = sym.MatrixSymbol('eigenValue',3,1)
8      eigVal = sym.Matrix(w)
9
10     # Invert order to be consistent with OpenFOAM
11     tmpEigVal = eigVal.copy()
12
13     # sort eigenValues
14     eigVal[0] = tmpEigVal[2]
15     eigVal[2] = tmpEigVal[0]
16
17     # sort eigenVectors
18     eigVec = eigVec.T
19     tmpEigVec = eigVec.copy()
20     eigVec[:, 0] = tmpEigVec[:, 2]
21     eigVec[:, 2] = tmpEigVec[:, 0]
22
23     A4 = closureFunc()
24
25     M = sym.MutableDenseNDimArray.zeros(3,3,3,3)
26     L = sym.MutableDenseNDimArray.zeros(3,3,3,3)
27
28     for i in range(3):
29         e_vector =  sym.Array(sym.flatten(eigVec[:, i]))
30         M += sym.tensorproduct(e_vector, e_vector, e_vector, e_vector)
31         L += eigVal[i]*sym.tensorproduct(e_vector, e_vector, e_vector, e_vector)
32
33     tmp1 = sym.tensorcontraction( sym.tensorproduct(M, A4), (2,4), (3,5) )
34
35     tensorCombination = A4 + (1-k)*(L- tmp1)
36
37     return tensorCombination
```

Listing 3. Symbolic computation of the RSC closure model.

The hybrid closure calculation used in the `computeRSCClosure` method is shown in Listing 4, which comprises only the `main` function. Similarly to the one shown in Listing 2, upon creating A_{ij} (Lines 7–9), and $[\mathbb{A} + (1-\kappa)(\mathbb{L} - \mathbb{M} : \mathbb{A})]$ (Line 11), following the RSC framework, the contraction with the rate-of-deformation tensor is performed (Line 13). Finally, in Line 15, the associated C-code is generated.

6. Case studies

This section presents the analysis performed to verify the implementation of the fiber orientation plug-in, implemented for modeling the evolution of fiber orientation, which was undertaken with two case studies: simple-shear flow and the flow in a center-gated disk. For these, an adaptation of the solver `pimpleFoam` was used. In both cases, because an analytic solution for the velocity field is available, the `pimpleFoam` solver was modified to remove all computations of primitive variables. For assessment

```
import sympy as sym
from fiberOrientationModelling_symbolicComputationTools import symm, generateCCode
from hybricClosure import computeHybridClosure
from rsc import RSC

def main():
    d = sym.MatrixSymbol('D',3,3)

    D = symm( sym.Matrix(d) )

    A4 = RSC(computeHybridClosure)

    D_doubleDot_A4 = sym.tensorcontraction(sym.tensorproduct(D, A4),(0,4),(1,5))

    generateCCode(sym.Matrix(D_doubleDot_A4))

if __name__ == "__main__":
    main()
```

Listing 4. Symbolic computation of the hybrid closure model with RSC.

purposes, the fiber orientation evolution results obtained in OpenFOAM®were compared with reference data obtained from the *Python* scripts provided in the repository folder `fiberOrientationPython`, which integrate the same fiber orientation evolution equations assuming homogeneous time evolving flows. For that purpose, a Runge-Kutta5(4) time integration scheme [47] was used, with an absolute and relative tolerances of 10^{-12} for both quantities.

6.1. **Simple-shear case study.** The velocity field for a simple-shear flow, with velocity along x, and a null velocity at $y = 0$ plane, is given by:

$$\underline{v} = \begin{bmatrix} \dot{\gamma} y \\ 0 \\ 0 \end{bmatrix}, \tag{45}$$

which, in accordance with the definition of gradient in OpenFOAM®, will result in the velocity gradient:

$$\nabla \underline{v} = \begin{bmatrix} 0 & 0 & 0 \\ \dot{\gamma} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{46}$$

To verify the accuracy of the numerical implementation, a mesh refinement study using 7 progressively refined grids was performed on the three-dimensional geometry shown in Fig. 4. The coarser mesh has 32 cells along the x- and y-directions and one cell in the z- direction. The number of cells in the x-direction were doubled for each successively finer grid.

The initial velocity field and boundary conditions were defined with the utilities `setExprFields` and `setExprBoundaryFields` by applying the exact solution in Eqn. (45). The shear-rate and fiber shape factor were assumed to be unitary, $\dot{\gamma} = 1\, s^{-1}$ and $\xi = 1$. The initial and Dirichlet boundary conditions for A_{ij} on the `left` patch is given by:

$$A_{ij} = \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{bmatrix}. \tag{47}$$

For the "(`right` | `top` | `bottom`)" patches a `zeroGradient` condition was assigned and for the "(`front` | `back`)" patches a `symmetryPlane` boundary condition was used.

For this case study, three fiber orientation models were tested, namely FT, iARD-RPR and MRD. The IBOF closure relation was employed with all models. For the FT model, two additional closure relations were tested, Hybrid and ORE. The relevant parameters for these models and closure relations are reported in Tab. 1 and were obtained from Favaloro and Tucker [27].

The discretization schemes and solution requirements employed are reported in Tabs. 2 and 3, respectively. The orientation evolution equation was integrated from 0 s until 65 s, with the time step size being controlled by the CFL condition for a maximum Courant number of 0.5.

Table 1. Fiber orientation model parameters for the simple-shear case study.

	Model		
Parameter	FT	iARD-RPR	MRD
C_I	0.0311	0.0562	0.0198
C_M	—	0.9977	—
α	—	0	—
D_i	—	—	$D_2 = 0.4796$ $D_3 = 0.0120$
Closure	Hybrid IBOF ORE	IBOF	IBOF

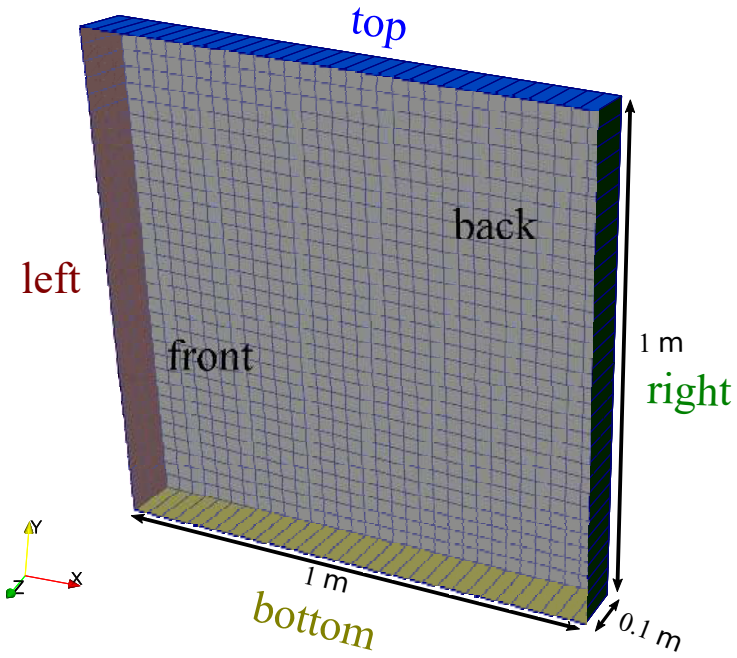


Figure 4. Geometry, mesh and patches for the simple-shear case study.

Table 2. Temporal and spatial discretization schemes for the simple-shear case study.

Term	Scheme
ddtSchemes	Euler
gradSchemes	grad(U) Gauss linear
divSchemes	div(phi,A2) Gauss upwind
laplacianSchemes	none
interpolationSchemes	linear
snGradSchemes	none

Table 3. Solution methods for the simple-shear case study.

Parameter	`p`
solver	PBiCGStab
smoother	DILU
tolerance	10^{-10}
relTol	0

6.2. **Center-gated disk case study.** The center-gated disk is a representative benchmark case of a typical geometry in the injection-molding (IM) process. Due to the usual small thickness of parts produced by IM, the flow is typically dominated by shear, but extensional flow can also occur [48]. In this case study, assuming constant viscosity (Newtonian behavior) is not realistic for IM of thermoplastic materials. However, it was used for convenience in verifying the implemented fiber orientation add-on, as an analytical solution is available.

The geometry and patches considered for the present case study are illustrated in Fig. 5. Due to the axisymmetric flow conditions just a small angular region of the disk was employed for verification purposes. Accordingly, the geometry in this case study is a 5° slice from a typical center-gated disk with a feeding system, with a radius $r \in [0.01, 0.12]$ m and a thickness of 0.003 m.

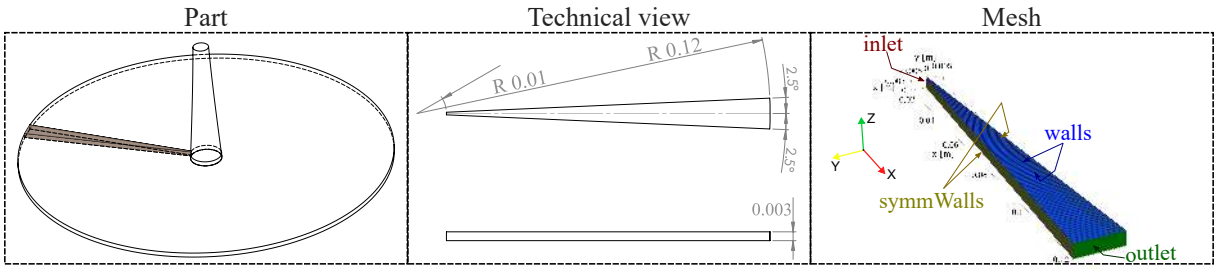


Figure 5. Part and mesh representation with the corresponding patches for the center-gated disk case study (dimensions in m).

The velocity field, in cylindrical coordinates (r, θ and z) for a Newtonian fluid in a center-gated disk is given by [41]:

$$\underline{v} = \begin{bmatrix} v_r \\ v_\theta \\ v_z \end{bmatrix} = \begin{bmatrix} \dfrac{3Q}{8\pi r b}\left(1 - \dfrac{z^2}{b^2}\right) \\ 0 \\ 0 \end{bmatrix}, \tag{48}$$

where:

- Q is the volumetric flow-rate,
- r is the radial position,
- b is the half-thickness of the disk, and
- z is the position along the thickness, where $z = 0$ corresponds to the mid-plane.

Taking into account the gradient definition in OpenFOAM®, the velocity gradient tensor is given by [41]:

$$\nabla \underline{v} = \frac{3Q}{8\pi r b} \begin{bmatrix} -\dfrac{1}{r}\left(1 - \dfrac{z^2}{b^2}\right) & 0 & 0 \\ 0 & \dfrac{1}{r}\left(1 - \dfrac{z^2}{b^2}\right) & 0 \\ -\dfrac{2}{b}\left(\dfrac{z}{b}\right) & 0 & 0 \end{bmatrix}. \tag{49}$$

Given the cylindrical symmetry of the problem and the Cartesian gradient implemented in OpenFOAM®, to ensure that Eqn. (49) was taken into account, the problem geometry was defined so that the x-direction aligns with the radial direction, $y = 0$, and z is perpendicular to the flow direction. For verification purposes a mesh refinement study was carried out, using five progressively finer meshes. The coarser mesh has 128 cells along the radial and thickness directions and one along the angular direction (θ in the cylindrical coordinates). The cells in the radial direction were doubled for each successively finer grid to increase the accuracy of the fiber orientation evolution.

For this case study, the flux and velocity fields were computed assuming a total flow rate of $Q = 134.774\ \mathrm{cm^3 s^{-1}}$. Given that only a $\alpha = 5°$ slice is being considered, the case study flow rate was adjusted to $Q\alpha/360$.

For this case, the flux and the gradient at the cell center were prescribed through a coded `functionObject` following Eqns. (48) and (49), respectively. For the second-order orientation tensor, the initial condition and Dirichlet boundary condition for the inlet were defined as isotropic state, Eqn. (47). The

`zeroGradient` boundary condition was applied to the `walls` and `outlet`, and `symmetry` to the `symmWalls` patch. The discretization and solution settings were the same as those considered considered for the simple-shear case study, Tabs. 2 and 3. Moreover, the fiber orientation parameters are described in Tab. 4.

A `coded functionObject` was used to define a steady-state criteria. The simulation was stopped when the initial maximum residual of A_{ij} was below 10^{-6}, and during 15 consecutive time steps the absolute difference between the current and previously computed A_{ij} was below 10^{-6}, in all computational cells.

Table 4. Fiber orientation model parameters for the center-gated disk case study.

	Model
Parameter	FT
ξ	1
C_I	0.001
closure	IBOF

7. Results and discussion

7.1. **Simple-shear case study.** In the simple-shear case study, taking into account the velocity field given by Eqn. (45), the A_{xz} and A_{yz} components of the fiber orientation tensor are null, whereas the A_{zz} component will have a similar dynamic to the A_{xx} component [39]. Thus, for verification purposes, only the A_{xx} and A_{xy} components will be analyzed. The A_{xx} component indicates the alignment in the x-direction and the off-diagonal component the direction of orientation in the $x-y$ plane [39]. To compare the results from the *Python* script with the ones computed in OpenFOAM®, and given that all streamlines are parallel to the x-axis, a pseudo-time ($t^{'}$) was calculated based on the cell velocity and the domain length by:

$$t^{'} = \frac{L_x}{u}, \tag{50}$$

where L_x is the length of the domain along the x-direction, with $L_x = 0$ at the inlet, and u is the x-direction velocity component.

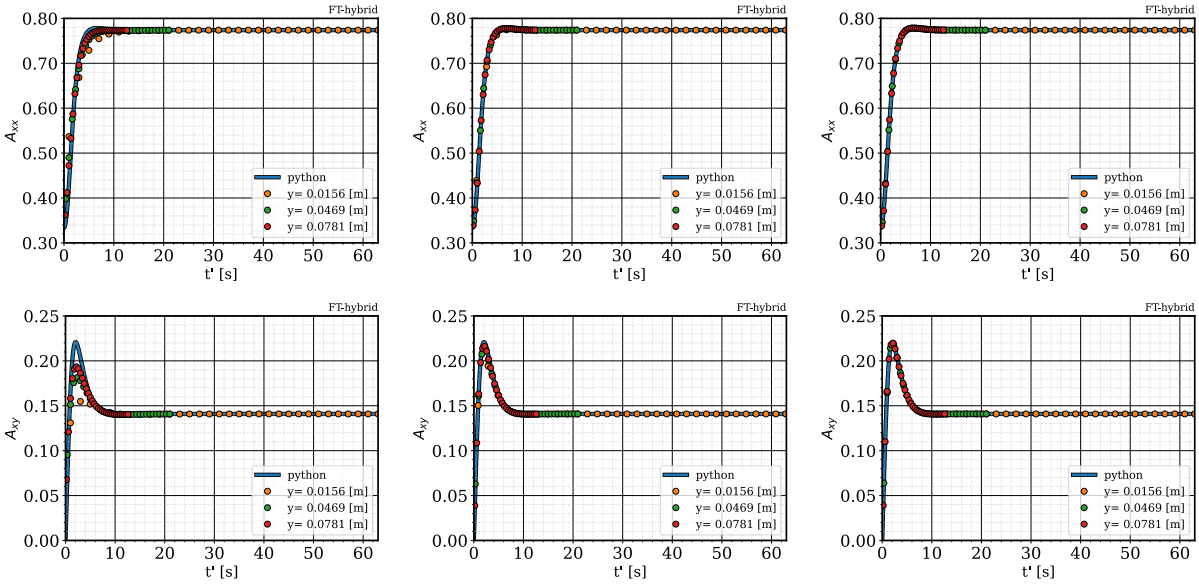
Given that only the velocity component in the x-direction is non-null, each line of cells in the x-direction is expected to match a section in the fiber orientation curve. Figures 6, 7 and 8 depict the comparison between the A_{xx} and A_{xy} components of the fiber orientation tensor with progressive mesh refinement for the FT model using the different closure models tested. The scatter data displayed corresponds to the line of cells at the y-position with the most significant numerical difference compared with the *Python* reference data. Figure 9 shows the evolution of A_{xx} when using the iARD and MRD models with the IBOF closure.

The parameters defined for the models were set to obtain roughly the same value at a steady-state ($A_{xx} \approx 0.65$) [27]. Comparing the different closures used with the FT model it is clear that the hybrid model over-predicts the value of orientation for this simple-flow. This is a known issue in literature [39,41], however, due to its computational simplicity it is employed in some proprietary software. The IBOF and ORE present very similar transient results. This has also been addressed in literature [41], where the major practical difference is the computational effort. The ORE requires the eigendecomposition of the tensor A_{ij} whilst the IBOF works with its invariants. Comparing the FT model with the iARD and MRD, although the solution at steady-state is the same, the transient behavior differs substantially. The FT model flattens out after ≈ 10 s while the iARD and MRD model present a pronounced peak. This shows the ability of different phenomenological orientation models to capture different orientation behaviors, whose choice is not trivial.

The results also show that with progressive mesh refinement, the simulated data computed in OpenFoam® approximates the reference data generated with the *Python* script.

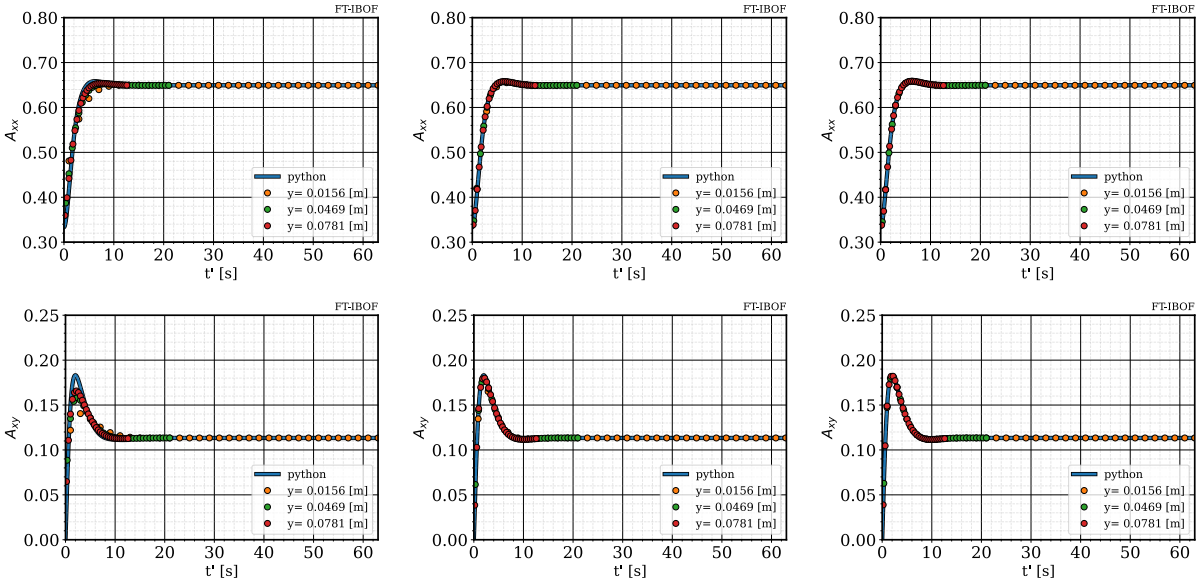
7.2. **Center-gated disk case study.** The parabolic flow profile given by Eqn. (48) represents a non-homogeneous radial diverging flow field where the velocity magnitude decreases along the radial direction, and, accordingly, the shear ($\dot{\gamma}$) and stretch ($\dot{\epsilon}$) rates ratio are functions of the thickness and radial position, as illustrated in Fig. 10.

Figure 10 depicts the ratio between the shear and extensional components on half-thickness of the disk. For these conditions, null shear and elongation will occur at the mid-plane and wall, respectively.



(a) Mesh 1 **(b)** Mesh 4 **(c)** Mesh 7

Figure 6. Mesh refinement study with the Folgar-Tucker model and the Hybrid closure for the simple-shear case study.



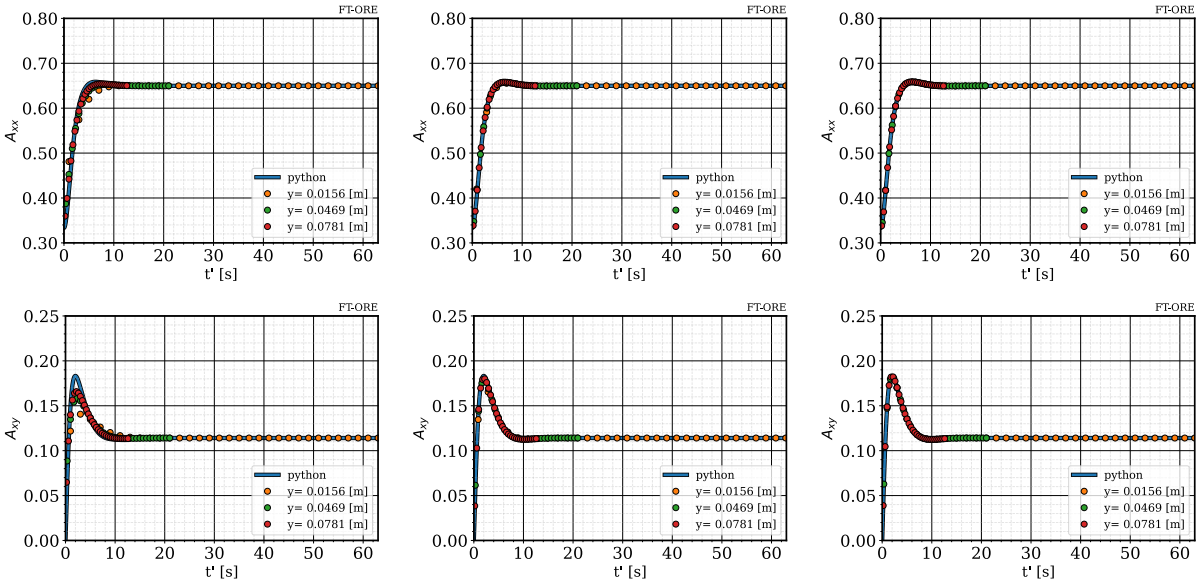
(a) Mesh 1 **(b)** Mesh 4 **(c)** Mesh 7

Figure 7. Mesh refinement study with the Folgar-Tucker model and the IBOF closure for the simple-shear case study.

At the mid-plane, where $\frac{\partial u}{\partial x} < 0$ with null shear-rate, the fibers are compelled to orient along the cross-flow direction. As the magnitude of z increases, the shear-rate becomes more pronounced, reaching its maximum at the wall. Consequently, the fibers tend to orient increasingly along the flow direction.

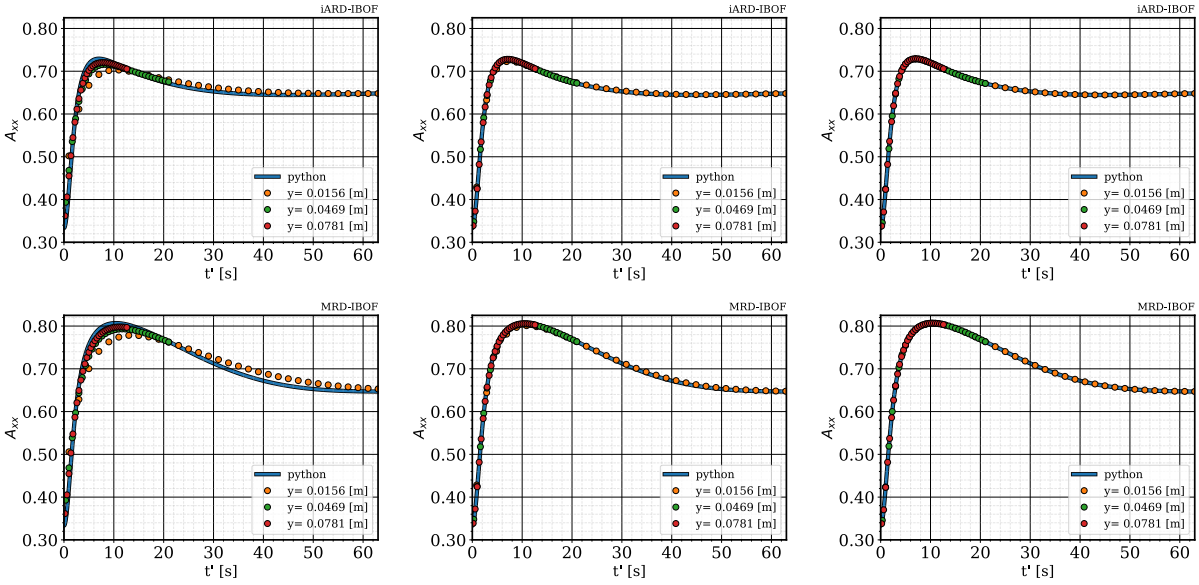
Similarly to the simple-shear case study, the A_{xy} and A_{yz} components of the fiber orientation tensor are always zero, whereas the A_{zz} component will have a similar dynamic to the A_{xx} (since the trace must be unitary). Thus, for verification purposes, only the A_{xx} and A_{xz} components will be presented.

Referring to the results shown in Figs. 10 and 11, near mid-plane, at $z/b = 0.102$, the A_{xx} component initially tends towards zero due to the presence of a negative elongational rate and an insufficient shear-rate to keep the alignment of the fibers in the flow direction. However, as the fluid advances along the



(a) Mesh 1 **(b)** Mesh 4 **(c)** Mesh 7

Figure 8. Mesh refinement study with the Folgar-Tucker model and the ORE closure for the simple-shear case study.



(a) Mesh 1 **(b)** Mesh 4 **(c)** Mesh 7

Figure 9. Mesh refinement study with the iARD and MRD models for the A_{xx} component for the simple-shear case study

radial direction, the small but non-null shear-rate gradually, albeit slowly, aligns the fibers along the flow direction. This effect intensifies with z/b, i.e., while approaching the wall.

To compare the results obtained in OpenFOAM®, the FT model was integrated using 3000 linearly spaced points along the radius, and 128 points along the thickness corresponding the z-coordinate of the mesh. The orientation tensor was sampled at the cell centers with the `sampling functionObject` utility of OpenFOAM®. The mesh refinement results for the closest ten evenly spaced z/b ratios are displayed in Fig. 11. The mesh refinement study shows that with progressive refinement, the results obtained with OpenFOAM®approach the reference data generated from the *Python* script.

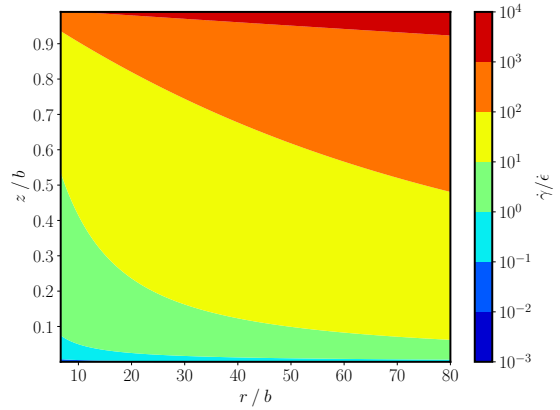
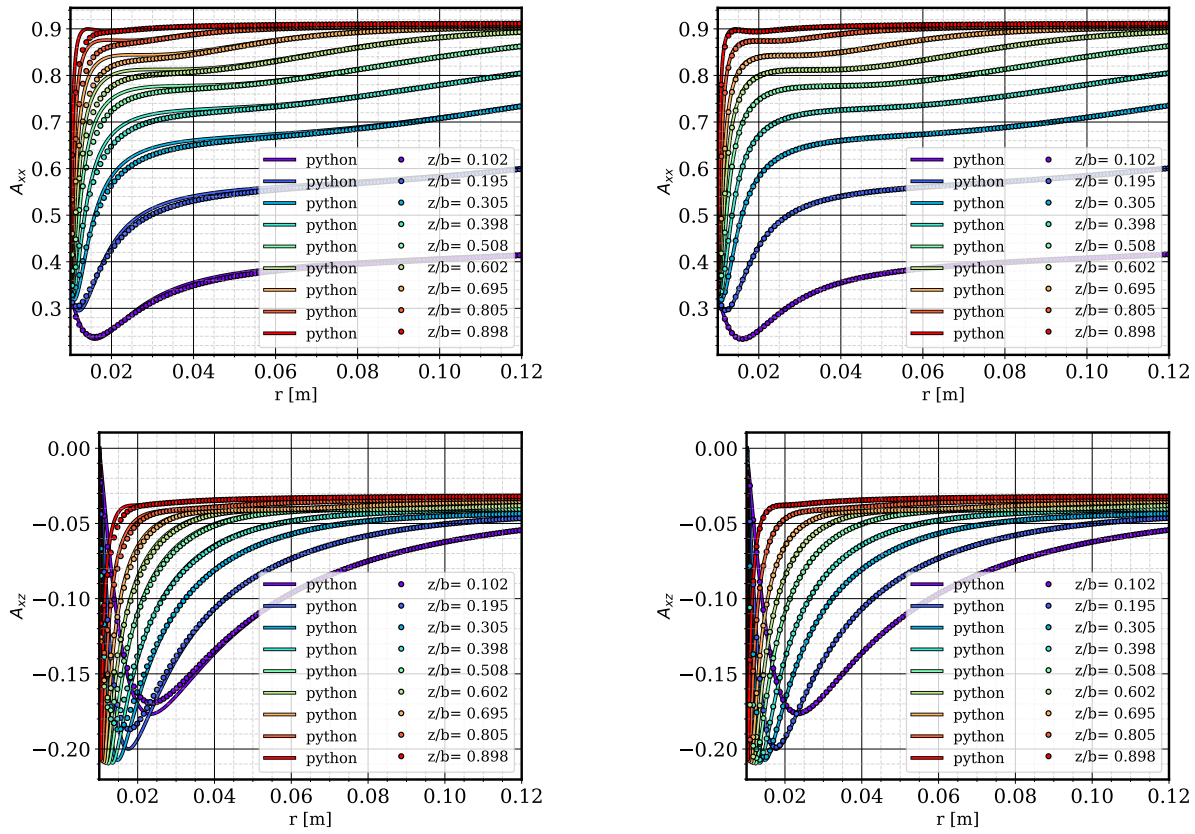


Figure 10. Map of shear/stretch ratio for the center gated disk.



(a) Mesh 1 **(b)** Mesh 5

Figure 11. Mesh refinement study with the Folgar-Tucker model and the IBOF closure for the center-gated disk case study.

8. **Conclusions**

In this work, a `functionObject` that works as a plug-in tool for any OpenFOAM®incompressible flow solver was programmed to calculate the evolution of fiber orientation in fiber reinforced thermoplastic materials, following a decoupled approach. Several fiber orientation models and closure relations available in the literature were implemented in the tool, which was verified by comparing the solver predictions with independent results obtained by numerically integrating the associated governing equations.

Due to OpenFOAM®'s limitations in handling higher-order tensors, the developed methodology for modeling fiber orientation, which requires fourth-order tensors, utilizes open-source symbolic computation to perform the necessary calculations. This symbolic approach retains only the non-trivial operations,

thus enhancing efficiency. However, this comes with the cost of code readability, as the functions resulting from the symbolic computations tend to be almost impossible to interpret.

State-of-the-art models and closure relationships currently employed in commercial solutions have been described and implemented in OpenFOAM®. Benchmark test cases, including a simple-shear flow and the flow in a center-gated disk, have been used to verify the numerical implementation, through the comparison with results obtained with a `Python` routine that integrates the equations for a single material point. A mesh refinement study was carried out for each type of flow. The results show that with progressive mesh refinement the OpenFOAM® results approach the `Python` counterpart.

The developed code, scripts, and case studies are open to the community, and the authors welcome suggestions and improvements.

Acknowledgments

This work is funded by FEDER funds through the COMPETE 2020 Programme and National Funds through FCT (Portuguese Foundation for Science and Technology) under the PhD Grant 2020.07424.BD and projects UID-B/05256/ 2020, UID-P/05256/2020. The authors acknowledge M. Oliveira for the fruitful discussions related to the numerical implementation of the orientation models.

References

[1] C.-T. Huang and H.-C. Tseng, "Simulation prediction of the fiber breakage history in regular and barrier structure screws in injection molding," *Polymer Engineering & Science*, vol. 58, no. 4, pp. 452–459, 2018. [Online]. Available: https://4spepublications.onlinelibrary.wiley.com/doi/abs/10.1002/pen.24660

[2] F. W. J. Van Hattum and C. A. Bernardo, "A model to predict the strength of short fiber composites," *Polymer Composites*, vol. 20, no. 4, pp. 524–533, 1999. [Online]. Available: https://4spepublications.onlinelibrary.wiley.com/doi/abs/10.1002/pc.10376

[3] M. Sepehr, G. Ausias, and P. Carreau, "Rheological properties of short fiber filled polypropylene in transient shear flow," *Journal of Non-Newtonian Fluid Mechanics*, vol. 123, no. 1, pp. 19–32, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377025704001582

[4] S. Mortazavian and A. Fatemi, "Effects of fiber orientation and anisotropy on tensile strength and elastic modulus of short fiber reinforced polymer composites," *Composites Part B: Engineering*, vol. 72, pp. 116–129, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359836814005642

[5] D. Masato, J. Rathore, M. Sorgato, S. Carmignato, and G. Lucchetta, "Analysis of the shrinkage of injection-molded fiber-reinforced thin-wall parts," *Materials & Design*, vol. 132, pp. 496–504, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264127517307001

[6] Y. Mu, G. Zhao, A. Chen, Y. Liu, Y. Song, and C. Sun, "Numerical investigation of three-dimensional fiber suspension flow by using finite volume method," *Polymer Bulletin*, vol. 74, no. 11, pp. 4393–4414, 2017.

[7] S. G. Advani and I. Tucker, Charles L., "The Use of Tensors to Describe and Predict Fiber Orientation in Short Fiber Composites," *Journal of Rheology*, vol. 31, no. 8, pp. 751–784, 11 1987. [Online]. Available: https://doi.org/10.1122/1.549945

[8] J. Wang, J. F. O'Gara, and I. Tucker, Charles L., "An objective model for slow orientation kinetics in concentrated fiber suspensions: Theory and rheological evidence," *Journal of Rheology*, vol. 52, no. 5, pp. 1179–1200, 09 2008. [Online]. Available: https://doi.org/10.1122/1.2946437

[9] J. H. Phelps and C. L. Tucker, "An anisotropic rotary diffusion model for fiber orientation in short- and long-fiber thermoplastics," *Journal of Non-Newtonian Fluid Mechanics*, vol. 156, no. 3, pp. 165–176, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377025708001602

[10] H.-C. Tseng, R.-Y. Chang, and C.-H. Hsu, "An objective tensor to predict anisotropic fiber orientation in concentrated suspensions," *Journal of Rheology*, vol. 60, no. 2, pp. 215–224, 03 2016. [Online]. Available: https://doi.org/10.1122/1.4939098

[11] H.-C. Tseng, R.-Y. Chang, and C.-H. Hsu, "The use of principal spatial tensor to predict anisotropic fiber orientation in concentrated fiber suspensions," *Journal of Rheology*, vol. 62, no. 1, pp. 313–320, 01 2018. [Online]. Available: https://doi.org/10.1122/1.4998520

[12] A. Bakharev, H. Yu, S. Ray, R. Speight, and J. Wang, "Using new anisotropic rotational diffusion model to improve prediction of short fibers in thermoplastic injection molding," in *SPE ANTEC Conference*, 2018.

[13] S. K. Kugler, A. P. Dey, S. Saad, C. Cruz, A. Kech, and T. Osswald, "A flow-dependent fiber orientation model," *Journal of Composites Science*, vol. 4, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2504-477X/4/3/96

[14] G. B. Jeffery, "The motion of ellipsoidal particles immersed in a viscous fluid," *Proc. R. Soc. Lond. A*, vol. 102, pp. 161–179, 1922.

[15] M. Gupta and K. K. Wang, "Fiber orientation and mechanical properties of short-fiber-reinforced injection-molded composites: Simulated and experimental results," *Polymer Composites*, vol. 14, no. 5, pp. 367–382, 1993. [Online]. Available: https://4spepublications.onlinelibrary.wiley.com/doi/abs/10.1002/pc.750140503

[16] W. Ogierman and G. Kokot, "A study on fiber orientation influence on the mechanical response of a short fiber composite structure," *Acta Mechanica*, vol. 227, no. 1, pp. 173–183, 2016.

[17] H.-C. Tseng, R.-Y. Chang, and C.-H. Hsu, "Numerical prediction of fiber orientation and mechanical performance for short/long glass and carbon fiber-reinforced composites," *Composites Science and Technology*, vol. 144, pp. 51–56, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0266353816318802

[18] K. Heinen, "Mikrostrukturelle orientierungszustände strömender polymerlösungen und fasersuspensionen," Ph.D. dissertation, Universität Dortmund, 2007.

[19] F. Ospald, "Contributions to the Simulation and Optimization of the Manufacturing Process and the Mechanical Properties of Short Fiber-Reinforced Plastic Parts," Ph.D. dissertation, Technische Universität Chemnitz, 2017.
[20] F. Wittemann, R. Maertens, A. Bernath, M. Hohberg, L. Kärger, and F. Henning, "Simulation of reinforced reactive injection molding with the finite volume method," *Journal of Composites Science*, vol. 2, no. 1, 2018. [Online]. Available: https://www.mdpi.com/2504-477X/2/1/5
[21] F. Wittemann, R. Maertens, L. Kärger, and F. Henning, "Injection molding simulation of short fiber reinforced thermosets with anisotropic and non-Newtonian flow behavior," *Composites Part A: Applied Science and Manufacturing*, vol. 124, no. May, p. 105476, 2019. [Online]. Available: https://doi.org/10.1016/j.compositesa.2019.105476
[22] K. Chiba and F. Chinesta, "Numerical simulation of flow kinematics and fiber orientation for multi-disperse suspension," *Rheologica Acta*, vol. 45, no. 1, pp. 1–13, 2005.
[23] G. Lipscomb, M. Denn, D. Hur, and D. Boger, "The flow of fiber suspensions in complex geometries," *Journal of Non-Newtonian Fluid Mechanics*, vol. 26, no. 3, pp. 297–325, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377025788800235
[24] J. Wang, "Improved fiber orientation predictions for injection molded composites," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2007.
[25] C. Tucker, *Fundamentals of Fiber Orientation: Description, Measurement and Prediction*. Hanser Publishers, 2022.
[26] F. Folgar and I. Charles L. Tucker, "Orientation behavior of fibers in concentrated suspensions," *Journal of Reinforced Plastics and Composites*, vol. 3, no. 2, pp. 98–119, 1984.
[27] A. J. Favaloro and C. L. Tucker, "Analysis of anisotropic rotary diffusion models for fiber orientation," *Composites Part A: Applied Science and Manufacturing*, vol. 126, p. 105605, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359835X19303549
[28] K. Breuer, M. Stommel, and W. Korte, "Analysis and evaluation of fiber orientation reconstruction methods," *Journal of Composites Science*, vol. 3, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2504-477X/3/3/67
[29] S. K. Kugler, A. Kech, C. Cruz, and T. Osswald, "Fiber orientation predictions—a review of existing models," *Journal of Composites Science*, vol. 4, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2504-477X/4/2/69
[30] A. J. Pontes, N. M. Neves, and A. S. Pouzada, "The role of the interaction coefficient in the prediction of the fiber orientation in planar injection moldings," *Polymer Composites*, vol. 24, no. 3, pp. 358–366, 2003. [Online]. Available: https://4spepublications.onlinelibrary.wiley.com/doi/abs/10.1002/pc.10035
[31] "Moldflow ard-rsc," https://help.autodesk.com/view/MFIA/2018/ENU/?guid=GUID-98E7B581-D2E6-4E9A-98C5-B79CAF919035, accessed: 2024-02-27.
[32] H.-C. Tseng, R.-Y. Chang, and C.-H. Hsu, "Phenomenological improvements to predictive models of fiber orientation in concentrated suspensions," *Journal of Rheology*, vol. 57, no. 6, pp. 1597–1631, 11 2013. [Online]. Available: https://doi.org/10.1122/1.4821038
[33] "Fiber function overview," https://support.moldex3d.com/2023/sync/for-nx/en/2-2-5-1_fiberfunctionoverview.html, accessed: 2024-02-27.
[34] S. K. Kugler, G. M. Lambert, C. Cruz, A. Kech, T. A. Osswald, and D. G. Baird, "Efficient parameter identification for macroscopic fiber orientation models with experimental data and a mechanistic fiber simulation," *AIP Conference Proceedings*, vol. 2205, no. 1, p. 020050, 01 2020. [Online]. Available: https://doi.org/10.1063/1.5142965
[35] A. Kech, S. Kugler, and T. Osswald, "Significance of model parameter variations in the pard-rsc model," *Journal of Composites Science*, vol. 4, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2504-477X/4/3/109
[36] A. Scheuer, A. Ammar, E. Abisset-Chavanne, E. E. E. Cueto, A. Scheuer, A. Ammar, E. Abisset-Chavanne, E. E. E. Cueto, F. Chinesta, R. Keunings, and S. G. Advani, "Data-driven upscaling of orientation kinematics in suspensions of rigid fibres," *Computer Modeling in Engineering and Sciences*, vol. 117, no. 3, pp. 367–386, 2018.
[37] M. Yun, C. A. Martin, P. Giormini, F. Chinesta, and S. Advani, "Learning the macroscopic flow model of short fiber suspensions from fine-scale simulated data," *Entropy*, vol. 22, no. 1, p. 30, 2020.
[38] "Fiber orientation prediction theory references," https://knowledge.autodesk.com/support/moldflow-insight/learn-explore/caas/CloudHelp/cloudhelp/2014/ENU/MoldflowInsight/files/GUID-BD0CFAC8-D8A9-42E7-8838-7DD2E67829B1-htm.html, accessed: 2024-02-27.
[39] S. G. Advani and C. L. Tucker, "Closure approximations for three-dimensional structure tensors," *Journal of Rheology*, vol. 34, no. 3, pp. 367–386, 1990.
[40] J. Cintra, Joaquim S. and I. Tucker, Charles L., "Orthotropic closure approximations for flow-induced fiber orientation," *Journal of Rheology*, vol. 39, no. 6, pp. 1095–1122, 11 1995. [Online]. Available: https://doi.org/10.1122/1.550630
[41] D. H. Chung and T. H. Kwon, "Invariant-based optimal fitting closure approximation for the numerical prediction of flow-induced fiber orientation," *Journal of Rheology*, vol. 46, no. 1, pp. 169–194, 2002.
[42] D. Chung and T. Kwon, "Fiber orientation in the processing of polymer composites," *Korea-Australia Rheology Journal*, vol. 14, no. 4, pp. 175–188, 2002. [Online]. Available: http://ftp.cheric.org/research/tech/periodicals/view.php?seq=395367
[43] B. E. VerWeyst, "Numerical predictions of flow-induced fiber orientation in three-dimensional geometries," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1998.
[44] "Fiber orientation tools," https://github.com/charlestucker3/Fiber-Orientation-Tools, accessed: 2024-02-27.
[45] "*Sympy*," https://www.sympy.org/en/index.html, accessed: 22024-02-27.
[46] B. Ramoa, R. Costa, F. Chinesta, and J. M. Nóbrega, "A semi-automatic approach based on the method of manufactured solutions to assess the convergence order in openfoam," *OpenFOAM® Journal*, vol. 2, p. 148–165, Nov. 2022. [Online]. Available: https://journal.openfoam.com/index.php/ofj/article/view/75
[47] "Scipy user manual," https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html, accessed: 2024-02-27.
[48] R. Zheng, R. I. Tanner, and X.-J. Fan, *Injection Molding: Integration of Theory and Modeling Methods*. Springer-Verlag Berlin Heidelberg, 2011.