

Research Paper

Instant geodesic draping for optimization and path-planning and its application to incremental composite sheet forming

Doran Nettig^{ID}*, Jan-Erik Rath^{ID}, Johann Kipping^{ID}, Thorsten Schüppstuhl^{ID}

Hamburg University of Technology, Institute of Aircraft Production Technology, Denickestr. 17, 21073 Hamburg, Germany

ARTICLE INFO

Keywords:

Fiber reinforced plastic
Incremental Sheet Forming
Simulation
Optimization
Tool path
Computer aided manufacturing

ABSTRACT

Technical textiles and their use in composites are crucial for many high-performance applications. Utilizing them in applications dominated by iteration and individualization requires flexible manufacturing processes. One such process is Incremental Sheet Forming (ISF), the 3D-Printing of forming processes, which is currently being adapted to composites. Enabling ISF, and other dieless processes, and leveraging their flexibility requires optimization and path-planning tools. Such tools must consider material behavior, which in turn necessitates the use of simulation. While highly sophisticated methods for increasingly accurate modeling have been developed, the speed requirements of CAM workflows and optimizations have kept simple yet efficient approximations relevant, the most prominent being kinematic draping algorithms.

This work presents a novel heat-method based alternative that globally computes fabric draping under an assumption of uniform tow distances. This assumption makes it especially well-suited for the subsequently implemented drape-aware ISF paths. Additionally, the developed algorithm is highly computationally efficient, which is harnessed for visualization and optimization. While these are only explored computationally, the path-planning is employed to produce and compare a demonstrator part in a double sided ISF process. The initial results for the draping algorithm show an up to 80x speed increase compared to kinematic draping. This allows for real-time draping visualization and sub-minute optimization. The demonstrator part proves that the derived paths can counteract wrinkling, an important failure mode observed with conventional strategies. These results demonstrate the great potential of geodesic draping for ISF in particular and flexible manufacturing, optimization and visualization tasks in general.

1. Introduction

Lightweight design is an important means of enabling high performance, energy efficiency, and enhanced functionality. As one of the prime lightweight materials with a high strength-to-weight ratio, continuous fiber-reinforced plastics (CFRP) are increasingly relevant in the transportation, energy, sports, and medical sectors [1]. Due to the influence of both the highly anisotropic behavior of the fibers and the embedding matrix, the material behavior of CFRP is extremely complex. Simulations to predict this behavior are employed not only for the design of the final part, but also for its manufacturing processes [2]. Consequently, a large number of studies in the past have focused on developing numerical methods to describe the CFRP behavior. A wide range of modeling complexities can be considered, depending on how accurately the material properties need to be represented and how finely the various constituents need to be resolved. Lim and Ramakrishna [3] roughly differentiate between mapping approaches,

relying purely on geometric data, and mechanics approaches, requiring constitutive material models for, i.e., FE-based simulations. These can be viscous, elasticity, viscoelastic, and/or plasticity models of microscopic (single fiber and matrix), mesoscopic (fiber-bundle/yarn and matrix), and macroscopic (textile/composite) scales [2,3]. While a prediction that is as realistic as possible requires extensive (at least mesoscopic) mechanical models, even the simplest mapping-based approximation methods remain relevant. One area of application where approximation methods are particularly necessary is the optimization of textile placement in manufacturing processes [4] such as hand-layup, (thermo)forming/stamping, or innovative flexible manufacturing processes such as Incremental Sheet Forming (ISF). In ISF, simple standard tools, such as hemispherical tooltips, are moved along pre-programmed paths to locally shape a sheet material without part-specific molding tools. An additional benefit is the higher formability of many sheet materials in ISF compared to standard stamping processes [5]. Formability can be increased even further by the successive forming of intermediate

* Corresponding author.

E-mail address: doran.nettig@tuhh.de (D. Nettig).

<https://doi.org/10.1016/j.cad.2026.104155>

Received 8 April 2026; Received in revised form 26 June 2026; Accepted 16 August 2026

Available online 1 September 2026

0010-4485/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

shapes with increasing depth and wall angle [6]. While originally developed for metal sheets [7], research into the ISF of CFRP has gained increasing interest in recent years [8]. Most research in this field has focused on Single Point Incremental Forming, which uses only one tool to form the desired target geometry [9–11]. However, recent publications also address Double-Sided Incremental Forming (DSIF), which uses an additional back-side tool to improve the process's flexibility and quality by localizing the deformation and compressing the layup [12–14]. In this context, a DSIF setup for the forming of continuous fiber reinforced thermoplastics (CFRTP) was developed in [15]. It employs two synchronized industrial robots, one on each side of the clamped sheet layup. While the main deformation is introduced by the position-controlled guiding robot, the second supporting robot applies a defined support force [15]. While previous CFRP ISF approaches usually adopted path strategies from metal ISF [9,10,12], (hard-coded) fiber-following paths were employed in [11], tailored to the draping of the employed woven fabric. The concept is to move the tool along the fiber directions, starting with the paths that require the least or no shear. Moving from the center outwards, paths are followed in parallel lines in an order of ascending shear angle, gradually introducing shear [16]. Although not algorithmically explored, the first investigation in SPIF of woven FRTP has yielded promising results, leading to higher geometric accuracy and a reduction in wrinkling, an important failure mode in CFRP forming [11].

The draping process, bringing a 2D textile into a 3D shape, is one of the most challenging and essential steps in CFRP production, and in most cases requires in-plane shear, out-of-plane bending, and intra-ply shear to accommodate doubly-curved shapes [17]. A key determining factor in the draping of a textile is the selection of a fixed point on the fabric and the fiber orientations at that point [18]. In practice, this point represents, for example, the point initially fixed during hand-layup [19]; in pressing processes, the point of first contact with the mold is the starting point of the draping driven by the pressing motion [20]. The starting point and orientation of a textile significantly influence the maximum shear angle that develops between the warp and weft yarns of a woven fabric or between the fibers of a 0/90 layup during draping, and thus the drapability [4]. At the same time, the fiber directions naturally also influence the properties of the component under load. Algorithms used in the past to optimize textile placement include, for example, genetic algorithms [21] and particle swarm optimization approaches [22]. While these approaches are well-suited for determining globally optimal results, they typically require a large number of calculations. Complex computations, which can take several hours even for dry textiles [23], result in computation times that are only acceptable for very high production volumes of the final component. For medium and low production volumes, however, the time spent on preparation increasingly impacts the lead time, thereby also increasing the demands on the computation time of upstream processes.

To enable the extremely short computation times required for both optimization and in the context of flexible manufacturing processes, the aforementioned mapping approaches are commonly used to model the textile behavior [4,24]. These approximation methods are designed to be as independent as possible of the specific material properties and rely on highly simplified assumptions based on the general type of reinforcement [3]. Due to their high tensile stiffness and low bending stiffness, the reinforcement fibers are generally assumed to be inextensible and freely bendable. For woven fabrics and unidirectional (UD) layers, two subtly different assumptions are made [3]: The behavior of dry woven fabrics is defined by the assumption of fixed node spacings, which implies the absence of fiber slippage. Combined with the aforementioned simplifications, this assumption results in a continuum mechanical description based on pure shear. Algorithmically, the resulting draping pattern under this assumption is computed by modeling the fabric as a pin-jointed net of nodes connected by uniform-length line segments, mechanically a grid of inextensible trusses connected via

ball joints [25]. This type of algorithm is often referred to as kinematic draping in contemporary and commercial implementations [4,26]. The computation of the grid's draping pattern is carried out in three main steps:

1. An origin node and two perpendicular initial fiber directions are defined.
2. Generators, i.e., two initial fiber paths, are traced as approximated geodesic lines from the origin.
3. The rest of the nodes are placed constrained by the generators and the unit length assumption of the segments.

The main differences among existing kinematic algorithms lie in the specific determination of the generators and nodes. An open-source implementation is presented by Krogh et al. [4]. Here, the approximated geodesic generators are determined via energy minimization. Krogh et al.'s efficiency-oriented implementation calculates the nodes via the intersection of two spheres placed on the adjacent nodes. However, an actual finite element analysis using truss elements is also possible [23].

For UD layers, the assumption is not that of constant node spacing, but rather that of constant spacing between adjacent fiber tows [3]. From this, a shear behavior is derived that is continuum-mechanically equivalent to simple shear. Even though the aforementioned differences in the theoretical description of fabrics and UD layers exist, in practice, kinematic draping simulation is also used for modeling UD layups, assuming constant node spacing [27]. Sun et al. compared kinematic draping to more extensive simulations [23]. As expected, kinematic algorithms show a significantly lower level of modeling accuracy. However, they also require a significantly reduced computation time. The kinematic approximations can be executed in a matter of seconds even on desktop hardware, whereas more extensive simulations can take minutes, hours, or even days, even on high-performance clusters. It is likely this factor that has kept kinematic approximation a viable method for optimization applications [22], even with the dramatic increase in computing power over the past decades.

The draping algorithm developed in this work does not rely on the local construction of kinematic grids, it instead builds on the work of Keenan Crane and his colleagues, who developed several algorithms based on the so-called heat method [28–30]. The heat method was developed for the efficient and robust calculation of distances on triangular meshes, amidst other domains. It computes the so-called geodesic distance as the shortest distance between two points on an arbitrarily shaped subdomain of a Euclidean space, constrained to this subdomain. To give an intuitive example, it would be the actual shortest hike between two points in a mountain range, rather than their straight-line distance. The heat method as originally introduced by Crane, Weischedel and Wardetzky [28] computes the geodesic distance from a set of points on the domain to all other points of the domain. It does this by leveraging the fact that the gradient of short-time heat diffusion becomes (virtually) parallel to the gradient of the geodesic distance as the diffusion time approaches zero. A suitable intuition might be the fact that after a very short time, the only relevant contributions from heat diffusing along random walks on the surface are those that travel along shortest geodesics.

Building on the original heat method and the vector heat method introduced by Sharp, Soliman and Crane [29], Feng and Crane recently introduced the heat method for generalized signed distances [30]. Relevant to this work is this method's ability to compute the signed geodesic distance field from curves on a curved surface. While the original heat method could approximate the geodesic distance from curves by treating their vertices as a multi-source, the signed heat method adds three important properties to the resulting fields:

- The fields encode a notion of directionality, i.e., to which side of the source curve they are calculated.
- The fields, by construction, possess smooth gradients.

- The fields show an advantageous boundary behavior.

The importance of these properties will be elaborated when discussing the draping algorithm implementation in Section 2.1. In the original heat method, a heat gradient is calculated as the gradient of a scalar diffusion. The signed heat method instead obtains the signed heat gradient H_{sgn} by a vector diffusion from a source vector field H_0 along the source curve. This source vector field H_0 is computed such that it points normal to the source curve at every point of the source curve, to a consistent side, and along the surface. H_0 is 0 everywhere else. Once defined, the vector diffusion step of the signed heat method is based on a vector version of the heat equation, where the time derivative of a vector field H_{sgn} is equivalent to applying the Connection Laplacian Δ^∇ :

$$\frac{\partial}{\partial t} H_{sgn} = \Delta^\nabla H_{sgn}. \quad (1)$$

Accordingly, H_{sgn} for a short time step t is obtained via short time heat diffusion of H_0 by solving:

$$(id - t\Delta^\nabla)H_{sgn} = H_0, \quad (2)$$

where id is the identity. The resulting field H_{sgn} is normalized to obtain the normalized signed heat gradient $\hat{H}_{sgn}$:

$$\hat{H}_{sgn} = \frac{H_{sgn}}{\|H_{sgn}\|}. \quad (3)$$

$\hat{H}_{sgn}$ approximates the gradient of signed geodesic distance $\nabla\phi_{sgn}$. Thus, ϕ_{sgn} is found by solving a Poisson equation:

$$\Delta\phi_{sgn} = \nabla \cdot \hat{H}_{sgn}. \quad (4)$$

The heat methods basically break down geodesic distance computation into two (sparse) linear solves and an intermediate normalization step. This allows them to leverage mature linear solvers, making them incredibly efficient. Repeated computations on the same underlying domain can reuse an initially computed prefactorization, further accelerating those solves.

Our work uses the heat method for generalized signed distance to develop a global computing method based on the assumption of constant fiber spacing, which is also applicable to multiple fiber orientations. It can therefore also be used to predict shear in, for example, orthogonal layups or woven fabrics. Since the condition of constant crossing point distances is explicitly not satisfied, its mapping accuracy is likely higher for layups and woven fabrics with a stronger tendency towards fiber slippage than for fabrics that exhibit strictly kinematic behavior. While an empirical comparison of the accuracy of the kinematic approximation and the approach developed here for various materials goes beyond the scope of this work, we do compare the results of the two approaches both qualitatively and quantitatively. In addition to the differences in their predictions, the computational efficiency of the two approaches is also compared. The efficiency-oriented implementation by Krogh et al. [4] in their Python implementation serves as the reference implementation in this work, as this allows for a direct comparison of the two approaches under identical conditions.

The algorithm's extreme computational efficiency enables fast visualization of the shear angle distribution. For this purpose, a basic software tool is presented in this work. For the same reason, the algorithm is naturally particularly well-suited for optimization tasks. Here, this is demonstrated using the example of a basic optimization algorithm, which can find a shear angle-optimizing starting position for draping processes in just a few seconds. As part of this work, we perform an initial validation of this optimization step using geometries for which the optimum is trivially known, in order to compare the optimization result against a clear ground truth. As a mapping approach the potential of the novel draping algorithm, which we develop and examine in this work, lies in both engineering and manufacturing technology. While it is generally not limited to this field, we focus

in particular on its potential for incremental CFRP forming. In this context, both the quick determination of the draping starting point and the derivation of forming toolpaths are key applications. In addition to incorporating the behavior of the reinforcement textile, the geodesic approach ensures forming paths that are equidistant, thereby achieving good surface quality. To leverage these properties, a path planning pipeline for the ISF of CFRP is developed that uses the newly developed draping algorithm as the basis for fiber-following path planning. This path planning pipeline is validated by manufacturing a demonstrator part on a double-sided incremental forming setup and comparing it with a conventional path strategy.

2. Algorithms

2.1. Geodesic draping algorithm

Currently the method of choice for approximating fabric draping is kinematic draping, both for woven fabrics and for UD-layups. However, certain properties make it less well suited for applications like path-planning. In the context of path-planning for manufacturing processes two properties are desirable, which current kinematic approaches are lacking: Many manufacturing processes benefit from or even require (close to) equidistant path spacing, which is often referred to as constant scallop width. Furthermore, path-planning usually requires carrying out multiple successive algorithmic operations. These concurrent steps benefit from a shared mesh geometry that is ideally identical to the target geometry provided by a CAD-system. This shared structure avoids the need for error prone sampling and interpolation operations to transfer information between meshes. In this context, we present a new alternative for thinking about fabric draping approximation. Instead of a local cell-based approach, we consider the problem globally. Instead of generators constraining an iteratively constructed equilateral grid, they constrain a global pattern of crossing distance fields. Instead of assuming a pure shear behavior with constant line segment lengths, our method assumes equidistant shear with constant geodesic fiber tow distances. Although the assumption of constant fiber tow distances originates from the modeling of UD layers, the approach developed here is also suitable for determining the shear of intersecting fiber directions. As described in the introduction, UD layers with constant fiber tow distances can be modeled by simple shear. However, this local approach cannot be transferred to a net-based method with two fiber directions without violating the condition of fiber inextensibility. If a net is sheared while the lengths of the line segments between crossing points remain constant, the distance between the segments decreases. Conversely, if segment distances remain constant, the segments themselves must elongate. Only the UD formulation of simple shear with a single equidistant fiber family does not lead to a contradiction here. But simple shear too is not capable of representing the shear of two intersecting equidistant fiber families. While the elongation occurs in only one direction in simple shear, it must take place in both fiber families to keep both families equidistant, which we refer to as equidistant shear in the following. Fig. 1 illustrates the difference between the three cases of pure, simple, and equidistant shear. In the original unsheared case two equally spaced fiber families of five fibers each cross each other at right angles. The resulting grid consists of 40 equal length line segments with a total length L . In the case of pure shear, the individual and total length of the line segments between the crossing points remain constant. Consequently, the distance between neighboring fibers is reduced and the height of the pure shear fabric is lower than that of the unsheared fabric. In the case of simple-shear the length of the horizontal fiber family stays constant as well. However, the distance between the fibers of this family is kept constant as well. This is achieved by elongating the segments of the vertical fiber family by a total length of e . In the case of unidirectional fabrics, this vertical family does not physically exist. Thus the inextensibility constraint only applies to the fiber segments

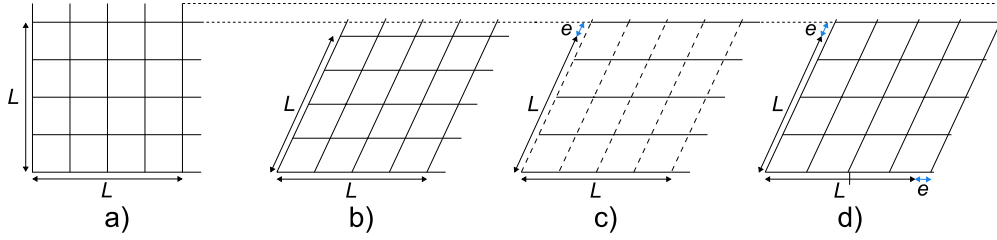


Fig. 1. Comparison of different shear types: (a) original unsheared state (b) pure shear (c) simple shear (d) equidistant shear.

of the horizontal family. In contrast, the equidistant shear assumption elongate the segments of both fiber families by a total length of e and keeps the distance between all neighboring fibers constant.

Locally this elongation of line segments violates the inextensibility assumption of the fibers. To illustrate why segment elongation represents only an apparent contradiction to the assumption of constant fiber lengths, Fig. 1 depicts each fiber's reach beyond its last crossing point. While the segment lengths change, the lengths of the depicted fibers stay the same under all assumptions. In real textiles, including woven fabrics, the distances between intersection points can and do change as well. However, this is not due to an elongation or shortening of the fiber bundles themselves. Rather, it is caused by fiber slippage. While segment lengths provide a local description of the relative positions of the intersecting fiber bundles, the immutability of fiber lengths is a global property. Furthermore, the property of fiber inextensibility should not be confused with the uniformity of all fiber lengths, which is valid only for a square-cut blank. Therefore, in theory, a curved surface can be perfectly covered with a textile with intersecting fiber directions, in which the distances between adjacent bundles in both fiber families remain constant everywhere, without violating the condition of fiber inextensibility. Thus, from a global perspective, a draping approximation can result in the desired equidistant fiber positions, even with two crossing fiber families. To realize such an approximation the presented algorithm is executed according to the same basic procedure as kinematic draping:

1. define the origin and initial fiber directions
2. compute the geodesic generators
3. compute the tow locations and shear constrained by the generators

with the requirements of resulting in an equidistant shear approximation that works as much on the input mesh geometry as possible. The first step of the computation starts with the mesh $\mathcal{M}$, the seed point $p_{in} \in \mathbb{R}^2$, and the initial fiber direction $d_{in} \in \mathbb{R}^2$ as inputs. The origin $p_{origin} \in \mathcal{M}$ is placed by projecting p_{in} vertically onto $\mathcal{M}$. The generator direction tuple D is created from the seed direction, its rotation by 90 degrees $d_{\perp}$, and the inverses of both directions.

$$D = (d_{in}, -d_{in}, d_{\perp}, -d_{\perp}) \quad (5)$$

While this initial step is quite unremarkable, we found two highly synergetic methods to realize the following steps of generator creation and shear computation: The generator creation uses straightest geodesic tracing in barycentric coordinates by tracing straight lines during iterative triangle unfolding and is shown in Fig. 2. They are traced from the origin outward along the generator directions as straightest-geodesics, following the definitions of Polthier and Schmies [31]. They describe this tracing process as the discrete initial value problem of geodesics: For a polyhedral surface S , a polyhedral surface point $p \in S$ and a polyhedral tangent vector $v \in T_p S$, there exists a unique straightest geodesic G that fulfills the initial conditions

$$G(0) = p \quad (6)$$

$$G'(0) = v, \quad (7)$$

and extends to the boundary of S . Locally, within a face f of S , the geodesic curve G reduces to a straight line. G can be extended into an adjacent face by proceeding in the direction for which the left and right curve angles are equal.

In this work, a custom version of the implementation of this concept in the potpourri3D library, which wraps the C++ library geometrycentral [32], is used. On the piecewise-flat triangulated surface mesh $\mathcal{M}$, straight line segments are traced iteratively across the mesh by isometrically unfolding each adjacent face into the tangent plane of the current face along their shared edge and extending the straight line into the unfolded face. The line segments are mapped back onto $\mathcal{M}$. Differing from the original potpourri3D implementation, we modified the bindings to return the points at which the lines cross from one face into the next as a tuple of barycentric coordinates, such that each point p_i consists of a reference to a face $f_j \in \mathcal{M}$ and the distances to that face's vertices t_A, t_B, t_C .

$$G = (p_1, p_2, \dots, p_n), \quad p_i = (j, (t_A, t_B, t_C)).$$

The four generators resulting from the directions D are paired into two orthogonal generators G_x and G_y . We pair this method of generator tracing with the heat method for generalized signed distance described in the introduction. This allows us to use the generators as constraints for two crossing signed distance fields, which result in the global definition of equidistant shear discussed before. Importantly, the method we adopted for generator creation allows us to incorporate them as the heat method constraints without changing the underlying mesh geometry. The conceptual progression of the method is shown in Fig. 3. The mesh $\mathcal{M}$ is used to initialize a signed distance solver. Because none of the algorithm's steps changes the underlying mesh geometry, it can serve the important purpose of maintaining the initial pre-factorization, effectively accelerating repeated solves. With the solver the signed distance field from each combined generator is calculated using the heat method for generalized signed distance as described in the introduction based on the original authors' implementation in potpourri3D. In two consecutive computations, the source vector field H_0 is computed from G_x and G_y , respectively, as the source curve. Then a discretized version of the short-time heat diffusion of H_0 (Eq. (2)) is solved. The diffused vector field H_{sgn} is normalized (Eq. (3)) and the signed distance field ϕ_{G_i} is obtained by solving the Poisson problem (Eq. (4)). The two resulting signed distance scalar fields, ϕ_{G_x} and ϕ_{G_y} , essentially encode an approximation of the warp and weft tows as their isocontours. It should be noted that although they are termed G_x, G_y , the initial fiber directions, defined by d_{in} and Eq. (5), need not be chosen as the x and y unit vectors. Even the orthogonality of the initial directions is not required by the method. Preshear could therefore be introduced by adjusting Eq. (5) accordingly. The tow locations can be derived trivially from the distance fields by just extracting their isocontours at the desired contour spacing. The two families of isolines then form our methods' equivalent of the kinematic grid. The gradients of the distance fields are orthogonal to the isolines of the respective fiber family and therefore to their fiber directions. Thus, the shear angle can be derived directly from these gradients of the computed distance fields. Assuming an orthogonal reference this is achieved by computing:

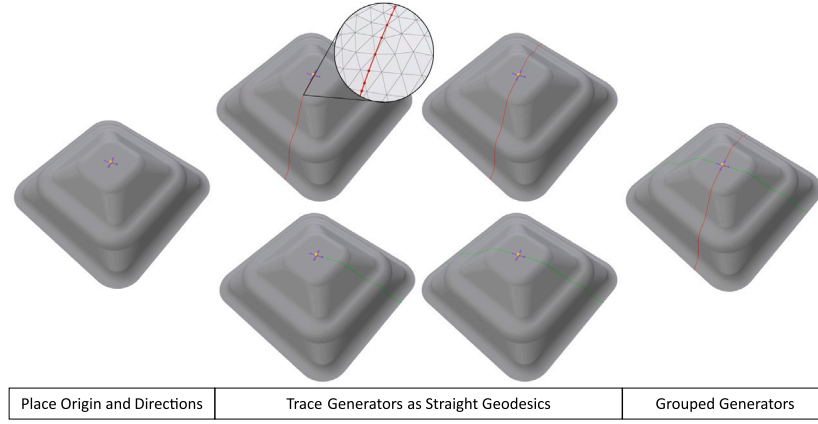


Fig. 2. The first two steps of the geodesic draping algorithm, the origin and initial directions are placed and generators are traced as straight geodesics in barycentric coordinates.

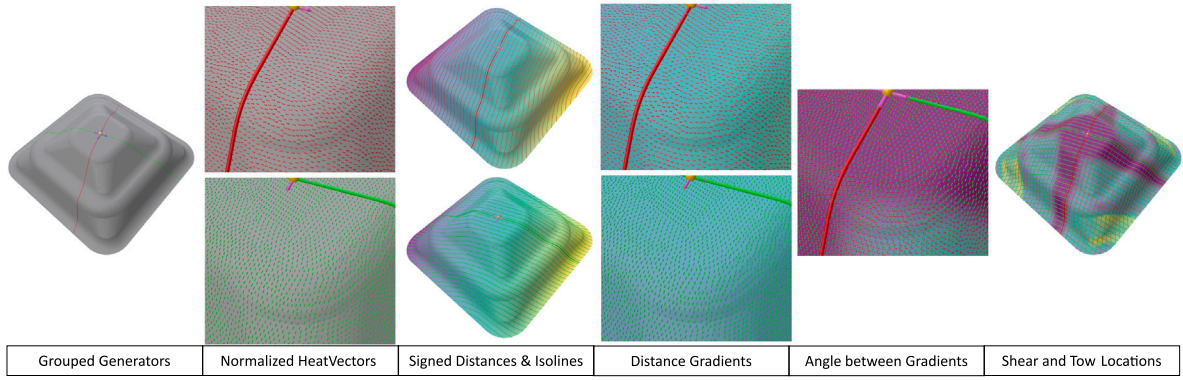


Fig. 3. The conceptual progression of the last step in the geodesic draping algorithm: Shear and fiber tow locations are computed constrained by the generators.

$$\alpha = \left| \arccos \left(\frac{\nabla \phi_{G_x} \cdot \nabla \phi_{G_y}}{\|\nabla \phi_{G_x}\| \|\nabla \phi_{G_y}\|} \right) - \frac{\pi}{2} \right| \quad (8)$$

Some applications might benefit from a non-orthogonal reference for the shear-angle computation. One such case might be to highlight the deviation from introduced pre-shear. In this case Eq. (8) can simply be adjusted by subtracting the relevant reference instead of $\pi/2$. The general algorithm implementing these steps is shown in Algorithm 1. The solver object is omitted as an implementation detail. Because the algorithm is built on the computation of straight geodesics and geodesic distance fields, we term the overall approach *geodesic draping*.

Although the presented algorithm is best suited to understand the conceptual progression of the presented method, we found that a significantly more efficient version of Algorithm 1 can be implemented, which is conceptually shown in Fig. 4. The vector fields $\hat{H}_{sgn}$ and $\nabla \phi$ are not necessarily exactly identical due to the approximative nature of the Poisson problem. Nevertheless, it may be worthwhile to accept this small additional inaccuracy. This allows us to take a short cut within our method: Instead of solving the Poisson problem and computing the gradients ($\nabla \phi_{G_x}$ and $\nabla \phi_{G_y}$) the shear angle can be calculated directly from the $\hat{H}_{sgn}$ fields. We refer to this algorithm as *fast geodesic draping* in the following. This alternative is especially useful, if the global fiber paths are not required and only α is needed, as is often the case in optimization.

In the context of path planning, the distance fields remain important, and further useful fields can be derived. The most important in this work is the element-wise minimum distance field ϕ_{min} .

$$\phi_{min} = \min(|\phi_{G_x}|, |\phi_{G_y}|) \quad (9)$$

Algorithm 1 The geodesic draping algorithm

Require: $\mathcal{M}, p_{in}, d_{in} \in \mathbb{R}^2$
 $p_{origin} \leftarrow \text{PROJECT}(p_{in}, \mathcal{M})$
 $D \leftarrow (d_{in}, -d_{in}, d_{\perp}, -d_{\perp})$
for d_i in D **do**
 $G_i \leftarrow \text{TRACE}(p_{origin}, d_i, \mathcal{M})$
end for
 $G_x \leftarrow \text{CONCATENATE}(G_0, G_1)$
 $G_y \leftarrow \text{CONCATENATE}(G_2, G_3)$
for G_i in (G_x, G_y) **do**
 $H_{sgn} \leftarrow \text{DIFFUSE}(G_i, \mathcal{M})$
 $\hat{H}_{sgn} \leftarrow \text{NORMALIZE}(H_{sgn})$
 $\phi_{G_i} \leftarrow \text{SOLVEPOISSON}(\Delta \phi_{sgn} = \nabla \cdot \hat{H}_{sgn})$ $\triangleright$ omitted in fast version
 $g_i \leftarrow \text{GRAD}(\phi_{G_i})$ $\triangleright g_i \leftarrow \hat{H}_{sgn}$ in fast version
end for
 $\alpha \leftarrow \text{SHEAR}(g_x, g_y)$ $\triangleright$ fast version only returns α
 $\phi_{min} \leftarrow \text{MIN}(\phi_{G_i})$

The distance fields have the additional useful property of providing a direct mapping from the target geometry to the planar initial state of the textile. Therefore, each point $p_m \in \mathcal{M}$ on the curved surface domain can be mapped to an original point $p_o \in \mathbb{R}^2$ in the textile's initial planar state. From the initial direction vector d_{in} , we can trivially extract the rotation relative to the plane's x -axis $R_{x \rightarrow in}$. Together with the initial seed point p_{in} , the mapping between curved and planar state can be calculated according to the procedure described in Algorithm 2.

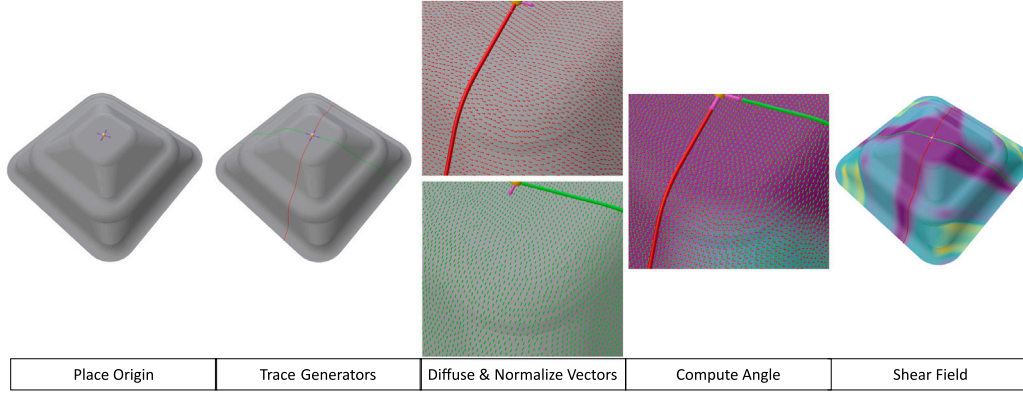


Fig. 4. The main steps of the fast geodesic draping algorithm computing shear directly from the normalized vector diffusion.

Algorithm 2 Transforming a submesh to the initial sheet domain

Require: $\mathcal{P}, \mathcal{F}, \phi_{G_x}, \phi_{G_y}, p_{in}, d_{in}$
 $\mathcal{M}_s \leftarrow (\mathcal{P}_s \subseteq \mathcal{P}, \mathcal{F}_s \subseteq \mathcal{F})$
 $\mathcal{P}_o \leftarrow \{T_o(p) \mid p \in \mathcal{P}_s\}$
 $\mathcal{M}_o \leftarrow (\mathcal{P}_o, \mathcal{F}_s)$
function $T_o(p)$
 $R_{x \rightarrow in} \leftarrow \text{rot}(\mathbf{e}_x, d_{in})$
 $p_o \leftarrow R_{x \rightarrow in} \cdot \begin{bmatrix} \phi_{G_x}(p) \\ \phi_{G_y}(p) \end{bmatrix} + p_{in}$
return p_o
end function

One application of this mapping is the computation of a minimum viable cutout from the flat textile that still covers the entire part. This can be achieved by simply choosing the part's boundary as the submesh $\mathcal{M}_s$ and mapping it back to the initial flat state. In reality, the true shear behavior will neither comply completely with the pure shear nor the equidistant shear assumption. As stated in the introduction, which description is better suited will likely depend on the reinforcement type and, in the case of woven reinforcements, on the importance of tow slippage. It remains to be investigated which materials, under which conditions, are better modeled by which set of assumptions. However, considering the inherently coarse approximative nature of a material-independent algorithm, this might not be as relevant in practice. More importantly the method we presented here allows us to effectively address the requirements mentioned in the beginning and therefore ends up with the following advantages:

- It allows derived toolpaths to be both fiber-following and of constant scallop width.
- It works directly on a possibly anisotropic, triangulated input-mesh.
- It can leverage shared pre-factorization to accelerate repeated solves.
- It allows for a straightforward computation of further useful quantities on the mesh geometry.

Important libraries used to implement the presented draping algorithm and the optimization, visualization, and path-planning algorithms presented in the following are numpy [33], vtk [34], and scipy [35] for basic mathematical operations and optimization algorithms, networkx [36] for graph operations, pymeshlab [37] and pyvista [38] for triangle mesh related operations, and matplotlib [39] and polscope [40] for visualization.

2.2. Optimization

The high computational efficiency of the developed draping algorithm predestines it for use in optimizations. To showcase this potential, this work implements a basic textile placement optimization. The scipy implementation of differential evolution, as outlined in [41], is utilized to find an initial position that minimizes the maximum shear in the fabric. The differential evolution solver uses a rand-to-best-1-bin strategy with a population size of 3. For mutation, the evolution uses dithering with values between 0.6 and 1, and the recombination constant is set to 0.8. The optimization terminates with an absolute tolerance of 0.001. The minimization of the 99.99th percentile of the shear field is chosen as the optimization target. The parametrization of the optimizer was chosen based on a preliminary parameter study across different input geometries, excluding the test geometries described later. After the differential evolution has converged, the result is used as input to a gradient-based minimization to further refine it. This work uses the scipy implementation of the modified Powell algorithm for this purpose [42,43]. The query bounds are automatically set to lie within the largest rectangle that fits the planar projection of the part, with a 5 mm margin to the boundary.

To assess the viability of the optimization, the input geometries used in this work are chosen such that the optimal fabric placement is (mostly) trivial for a human operator. Namely, the chosen geometries are a rotationally symmetric cone frustum, a half dome shape, and a variable-angle cone frustum with an off-center tip. This choice of test geometries allows an initial comparison of optimizer recommendations with the intuitively optimal placement, without conducting real-world experiments. For the cones, the optimal placement is expected to be found in the center of their top surface. For the dome, it is expected at its peak. Fiber orientation can be trivially included in the optimization. Yet it is omitted here as it is arbitrary for the rotationally symmetric parts and non-trivial for the off-center part.

Times, number of iterations, final shear, and the deviation from the trivial best point are recorded for analysis. Because differential evolution depends on the initial condition, each optimization is repeated five times with random initial conditions, and results are averaged across the five runs.

2.3. Visualization

An alternative to a fully automated optimization is the use of visualization to aid a human operator's decision. This offers the advantage of improving the process understanding of the operator. Furthermore, it allows more complex targets to be achieved by incorporating human knowledge and intuition. For example, while the optimization will usually target a simple metric such as maximum shear, the human operator might benefit from intuitions about where, in what shape, and in what size high shear areas occur.

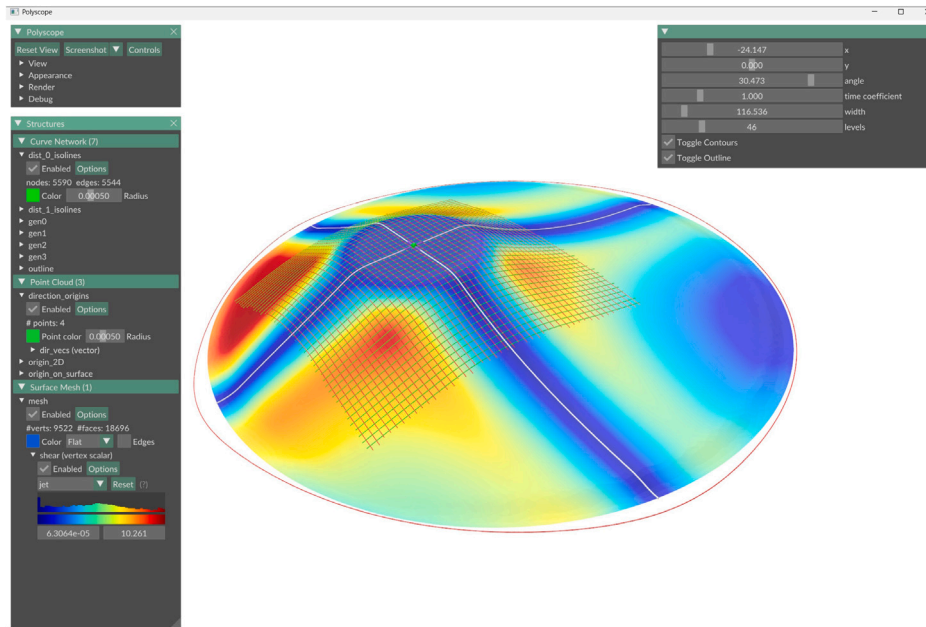


Fig. 5. The developed interactive shear simulation implemented using polyscope.

To enable this alternative route, we leverage the speed of the geodesic draping algorithm to build a simple interactive visualization tool. A screenshot of the visualization is shown in Fig. 5. The tool is realized in polyscope and allows the operator to change a number of parameters, enabling the observation of their impact on the draping of the textile, updated in real-time. The main parameters are the starting position and orientation. By default, the visualization uses the fast geodesic draping algorithm to approximate the shear field and display it as a color range on the mesh. To gain further insight, the operator can also toggle a visualization of the textile made up of the isocontours of the distance fields. If this option is toggled, the approximation uses the full geodesic draping approximation internally. The contours are cropped to a length specified by the width parameter and sampled at a distance derived from the levels parameter. When the width exceeds the part's size, the contours remain constrained to the part. One last option is to activate the display of an outline. This adds a red line to the visualization, which represents the minimal cut-out from a flat textile required to completely cover the part geometry, given the current choice of origin parameters.

Together, these options allow an operator to adjust parameters and immediately observe how they affect the results. Drawing conclusions from the observed behavior, the operator can then make an informed decision on which parameters to choose for subsequent process steps.

2.4. Incremental forming paths

The newly developed geodesic draping algorithm can be leveraged for the efficient and geometry-independent generation of draping-aware toolpaths for the incremental forming of CFRP. As explained in chapter 1, the concept of fiber-following paths has been identified as a promising approach for the suppression of wrinkling defects in the ISF of woven CFRP. Initial forming trials with hard-coded paths showed improved results compared to standard elevation-contour paths known from metal ISF [11]. To automate this approach a novel draping-aware path planning strategy specifically tailored to the shear- and bending-dominated deformation properties of CFRP is implemented using the geodesic draping algorithm. For comparison, and to enable efficient processing, we furthermore implement a constant-elevation spiral strategy commonly used in ISF. This chapter describes both path-planning algorithms, along with the necessary pre- and post-processing.

The path planning implementation is structured as a directed acyclic graph (DAG). All operations are therefore built up from interconnected nodes that process inputs from previous nodes to generate outputs, which are, in turn, processed by subsequent nodes. Interpreting these nodes as filters and their connecting edges as pipes, the path planning is, therefore, literally a pipeline. To guide the description in the following sections, the entire DAG will be subdivided into several consecutive sub-graphs. The relevant libraries used for the implementation of the pipeline are the same as mentioned in Section 2.1.

The input mesh geometry is first run through a preprocessing pipeline (see Fig. 6(a)). It allows for the application of basic transformations and remeshing to increase robustness to input variations. If the input geometry is correctly positioned and oriented and has a sufficiently fine and homogeneous mesh resolution, these steps can be skipped. Aside from these operations, the main task of the preprocessing pipeline is to generate intermediate shapes to enable multi-stage forming strategies. In the implementation presented here, the target mesh is warped into the desired intermediate shapes according to user-defined depth separations/layer distances l_d . First, the elevation field ϵ , i.e., the z-coordinates of each point p of the mesh, is extracted. It is then used to linearly interpolate the mesh point coordinates to the new coordinates p_{warp} according to the desired layer distances l_d along the z-direction n_z such that the maximum height of the warped mesh is equal to d_l .

$$p_{\text{warp}} = p - k \epsilon n_z, \quad k = 1 - \frac{l_d}{\max(\epsilon)} \quad (10)$$

This work uses the newly developed draping algorithm to automatically generate fiber-following toolpaths. First, the generators G_x and G_y , as well as the distance fields ϕ_{G_x} , ϕ_{G_y} , and $\phi_{\min}$ are obtained according to the procedure discussed in Section 2.1. The pipeline for turning the geodesic draping approximation into fiber-following paths is shown in Fig. 7. The isolines of the $\phi_{\min}$ field are computed according to the scallop width defined by the user. The generators G_x and G_y are included as the zero-contours. To efficiently sort the isolines, the arctan of their (ϕ_{G_x}, ϕ_{G_y}) position is calculated, which can then be used to order the individual lines. Together with $\phi_{\min}$, the set of lines is subsequently sorted by isovalue and quadrant. This ordering and sorting step determines several properties of the final path strategy. Isolines could be traversed clockwise, counter-clockwise, or alternatingly; consecutive isolines could increase or decrease in geodesic distance

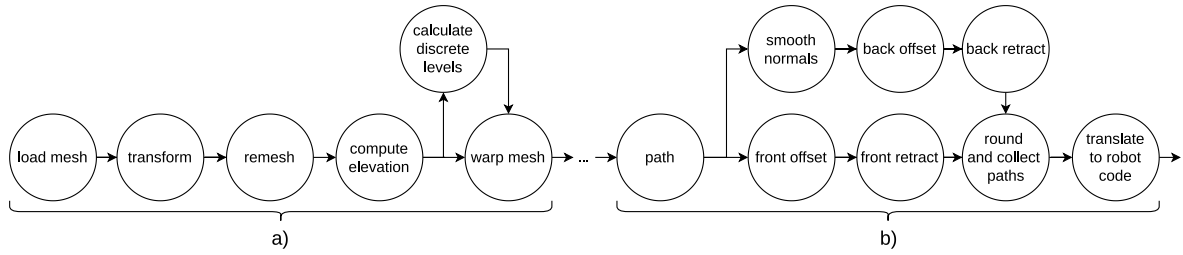


Fig. 6. (a) Pre- and (b) post-processing pipelines of the developed path-planning procedure.

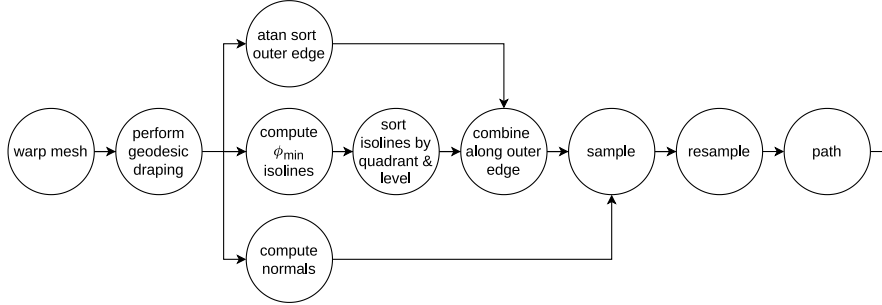


Fig. 7. Computation pipeline of the developed fiber-following path-planning algorithm.

to the generators. They could be formed levelwise or quadrantwise. Combinations of these could also be used. Once an ordered list is generated, the consecutive isolines are combined with the atan sorted outer contour of the mesh. In this step, the isolines are connected via shortest paths along the outer contour to form one continuous path. To prepare them for the final postprocessing steps, the normals of the underlying mesh are sampled onto the path, and it is resampled to a constant point spacing. When progressing from low to high shear angle areas, the fiber-following paths limit the maximum relative depth of an intermediate shape because of their inside-out character. It was reported in [11] that a high count of intermediate shapes led to the failure of the employed metal dummy sheets, which are required for clamping and protecting the CFRP.

To circumvent an excessive amount of intermediate shapes, only a limited number of intermediate shapes can be formed using fiber-following paths, while the remaining intermediate shapes are formed with a conventional outside-in spiral approach, which allows higher relative depth. The pipeline for the generation of these elevation contour based spiral paths is shown in Fig. 8. First, the scalar elevation field is extracted from the preprocessed surface mesh. The next node extracts the isolines of this scalar field as a set of contours, which is sorted into a list of clockwise-ordered contours with increasing isovalue. Starting from the first point of the first contour as the reference point, the contours are aligned by iteratively computing the squared distance from all points of the next contour to the reference point and then shifting the point indices to start with the point closest to the reference. To spiralize the isolines, the contours are pairwise resampled to a constant point spacing at an identical point count, which is chosen as the maximum of the point counts of both contours. The spiral segment is then created by linearly interpolating between the points of the contours and moving each successive point closer to the next contour by the same fraction of the total distance to the closest point on the next contour. This is iteratively repeated for each contour pair, and the resulting spiral segments are concatenated to form the entire spiral path. The resulting polyline is combined with the first and last contour to form the complete list of path points. This list is then resampled in its entirety to a user-defined equal point spacing. The interpolation and resampling are performed in Cartesian space and might therefore not lie exactly on the surface domain. For this reason, the path points are projected back onto the mesh by finding the closest cell and the closest

point within this cell for each path point. To complete the necessary path information, the surface normals of the mesh are computed and sampled at the path points.

The final path postprocessing step, shown in Fig. 6(b), is identical for both path planning pipelines. The path points are offset along their associated normals to form the path of the guiding robot, and approach and retract motions are appended. The same is done for the support robot, although its normals are smoothed with a Gaussian kernel w_k with a radius of nine points

$$\tilde{f}_i = \sum_{k=-9}^9 w_k f_{i+k}, \quad w_k = \frac{\exp\left(-\frac{k^2}{18}\right)}{\sum_{j=-9}^9 \exp\left(-\frac{j^2}{18}\right)}, \quad k = -9, \dots, 9. \quad (11)$$

This is done as a smooth path, and therefore, a smoother force is deemed important enough that a slight deviation from the calculated position is acceptable. With this, the positional path description is completed, and the path points can be translated into robot instructions via a custom parser. An elevation-based spiral path, as well as a fiber-following path, is shown in Fig. 9. To focus the visualization on the different path types, they are displayed on the final shape, and the paths for the support robot are omitted. Additionally, the contour distance is intentionally increased to an unusually high value of 8 mm of scallop width for the fiber-following paths and 4 mm of step depth for the elevation based spiral.

3. Forming experiments

To practically assess the path planning algorithms, incremental forming experiments were conducted on a hot DSIF setup specifically developed for CFRTP forming [15]. The setup is depicted in Fig. 10. It employs two KUKA KR 300 industrial robots positioned on opposite sides of a vertical clamping frame, which is able to clamp sheet layups of $350 \times 350 \text{ mm}^2$. Due to the near-inextensible nature of the fibers, the employed Ensinger TECATEC SAN CF47 CFRTP sheet cannot be clamped directly. Instead, it is quasi-floatingly interposed between a DC04 steel sheet on the front and an EN AW-1050A H111 aluminum sheet on the back side. Polytetrafluoroethylene (PTFE) separator sheets are introduced between CFRTP and metal sheets to prevent adhesion. The sheet material details are shown in Table 1.

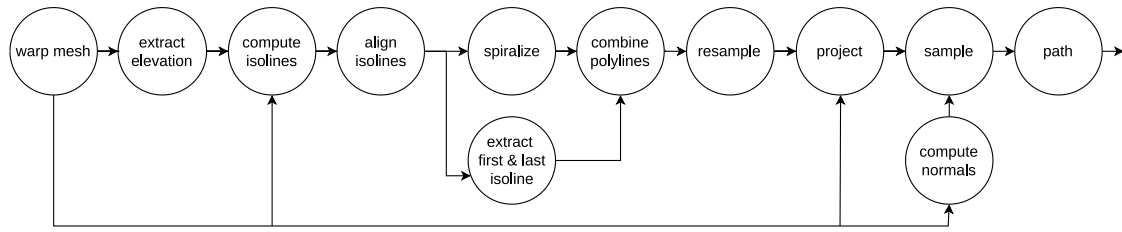


Fig. 8. Computation pipeline of the conventional elevation spiral algorithm.

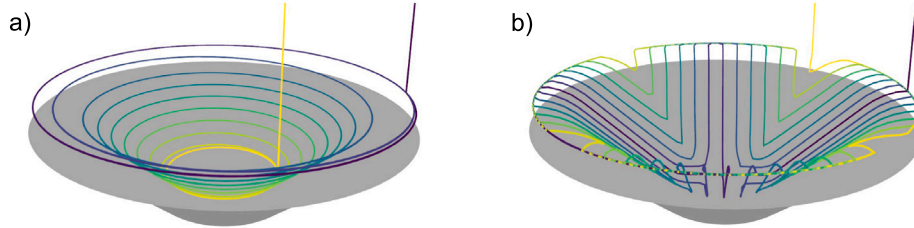


Fig. 9. Visual comparison of incremental forming path strategies: (a) elevation based spiral (b) fiber-following paths (path progress blue to yellow). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

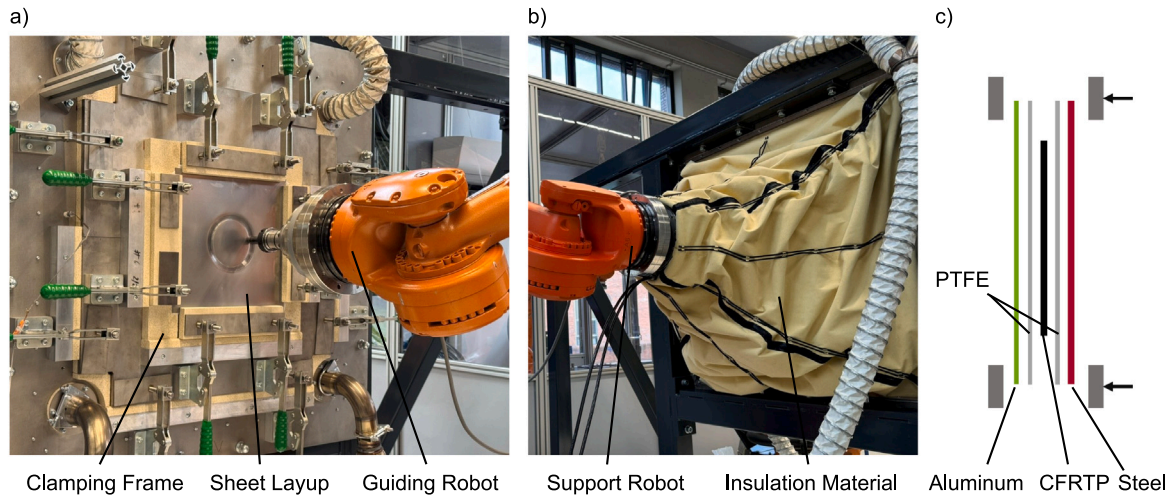


Fig. 10. Setup for the hot DSIF of CFRTP, (a) front view with guiding robot, (b) back view with support robot, (c) schematic sectional view of the layup.

During layup preparation on a separate horizontal layup table, care must be taken to centrally place the CFRTP on the underlying aluminum and PTFE sheet, with the fiber directions aligned with the sheet edges. The corners of the composite sheet were attached to the PTFE foil with temperature-resistant tape to prevent any dislocation during transfer to the clamping frame. A flexible insulation material, suspended between the rear side of the clamping frame and the back side robot's flange, and the back aluminum sheet of the layup constitute an oven chamber. A recirculation blower, a heater, and air deflection plates are employed for a homogeneous heating of the layup. After securing the layup in the clamping frame, the oven chamber is heated to a temperature of 180 °C. Once the target temperature is reached, the final clamping force is applied, and the forming process can start. To introduce local deformations into the sheet layup both robots are equipped with hardened steel tools with hemispherical tips of 20 mm diameter. The position controlled front robot, on the cold side of the layup, performs the main forming operation and is thus regarded as the guiding robot. The TCP of the support robot on the hot back side of the layup aids in localizing the deformation, supporting the

layup and preventing undesired global deformation, by applying a support force of 150 N. To this end, the supporting robot is not only synchronized with the guiding robot using the KUKA RoboTeam package, but also force-controlled leveraging the KUKA ForceTorqueControl package. Measurement data is provided by a six-axis force-torque measurement cell (Schunk FTN-Omega 160) mounted on the robot flange, decoupled from the hot area of the tool via a water-cooled adapter. Further details of the setup and the control architecture are reported in [13,15]. Following paths created by the pipeline described in Section 2.4, two cone frusta were formed. They have a final maximum depth of 40 mm and are each divided into four intermediate shapes. For both cones, three equally spaced intermediate shapes with a relative depth of approximately 12.3 mm are formed using the elevation spiral approach. The first strategy follows these by a fourth elevation spiral shape to the final depth. This strategy is thus referred to as E12-E12-E12-E3. The second cone instead uses fiber-following paths for the final shape and is referred to as E12-E12-E12-D3. The height distance between subsequent spiral passes is set to 1 mm, and the geodesic contour distance of the fiber-following paths is set to 2 mm. The target

Table 1
Details of the sheet layup constituents.

DC04 steel sheet	Size	350 × 350 mm ²
	Thickness	0.7 mm
EN AW-1050A H111 aluminum sheet	Size	350 × 350 mm ²
	Thickness	1 mm
SAN CF47 CFRTF sheet	Size	200 × 200 mm ²
	Thickness	0.6 mm
	Fiber Volume Content	47%
	Fiber Type	Carbon HT 3k
	Fiber Architecture	2/2 twill weave
	Textile Basis Weight	245 g/m ²
PTFE sheet	Size	350 × 350 mm ²
	Thickness	0.025 mm

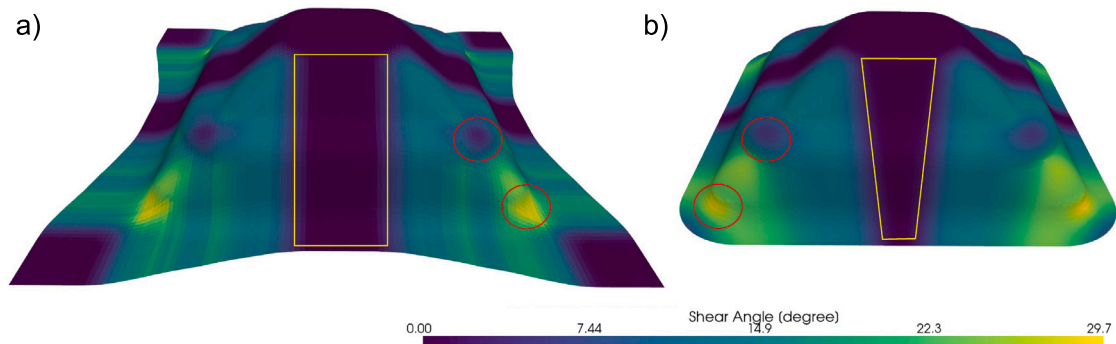


Fig. 11. Qualitative comparison of the (a) kinematic and (b) geodesic draping algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

TCP velocity is set to 20 mm/s for the guiding robot and to a relative speed of 5 mm/s for the support robot. After forming is finished, both tools retract to their home positions, and the heating is disabled. Once the layup has cooled down to a safe handling temperature, it is removed from the clamping frame, and the formed CFRTF is separated from the auxiliary sheets.

4. Results

4.1. Algorithm comparison

4.1.1. Angle prediction

Fig. 11 shows a qualitative comparison of the results of a kinematic draping algorithm as implemented by Krogh et al. [4] and the geodesic draping algorithm developed in this work. The geometry chosen for the comparison of the algorithms is a square pyramid with three different wall angles: A high angle section at the base (60°), lower angle section towards the tip (40°) and a small transition region (10°). As shown, both algorithms produce qualitatively similar results for this more complex geometry. Local peaks and valleys in the shear field appear at the same locations on the surface (examples marked in red). As expected, a region of relatively low shear forms immediately around the generators. While this is true for both methods, the most significant qualitative difference between them lies in the shape of this region: whereas the width of this region remains nearly constant in kinematic draping, it tapers in geodesic draping as the distance from the origin increases (marked in yellow). An obvious difference between the two approaches is that geodesic draping is calculated on the target geometry, whereas the kinematic draping algorithm always simulates a square fabric. It is also possible to calculate geodesic draping on a geometry with a boundary surface and extract a square or rectangle of any size; however, this is not done in this case.

Fig. 12 shows a quantitative comparison of the mean and maximum shear values computed by both algorithms on the same test geometry

that is used for the qualitative comparison. For this analysis, the starting point is fixed at the center of the component, and the angle of the initial generator direction to the x-axis is then increased in 150 steps from 0° to 90°. For initial angles of below about 25° and over about 65°, the mean and maximum shear angle prediction of both approximations remains very close. The maximum shear values differ within a band of about $\pm 2^\circ$ and are even closer looking at the 99.9th percentile of shear, which eliminates the influence of possible outlier nodes. This can also be seen in the absolute predictions, where small oscillations are effectively smoothed by excluding the extreme-valued nodes. The difference in the maximum predictions in the aforementioned angle section varies around a mean deviation of zero. However, the mean shear prediction of the geodesic draping algorithm is consistently about 1.3° larger than that of the kinematic algorithm. This is likely due to the narrowing band of low shear around the generators discussed previously. The deviation in the mean shear prediction rises only slightly in the section between 25° and 65°. In comparison, the mean on the maximum prediction shifts more significantly. After a short transition region the difference between the maximum predictions of both algorithms oscillates around about 5° , with a lower prediction in the geodesic algorithm. This implies that the algorithms react differently when the curvature in the area of the generators is higher. However, the difference is still quite low, and the overall progression shows a similar qualitative behavior, with a minimum of maximum shear near 45°. Due to the very small differences between geodesic draping and fast geodesic draping, the predictions of the latter algorithm are omitted here.

4.1.2. Computation speed

For the empirical computation speed assessment both geodesic draping algorithms (complete and fast) are used to compute shear for random starting points on the square pyramid geometry with varying wall angles. 200 points in a 30×30 grid are generated and used as

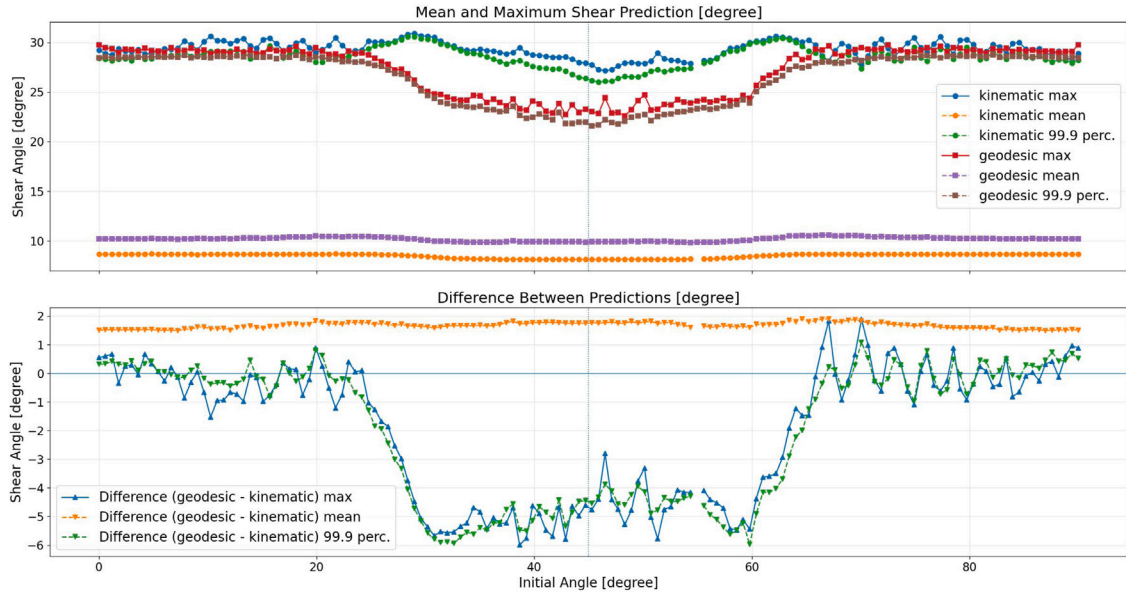


Fig. 12. Comparison of the mean and maximum shear prediction of the kinematic and geodesic draping algorithms for different initial fiber directions (angle around e_z relative to the e_x).

input. The input geometry is isotropically remeshed to a node count of 13749. They are furthermore compared to the kinematic draping algorithm using a grid with 13689 nodes. All three algorithms were run in direct succession on the same Lenovo ThinkPad P14s using an AMD Ryzen Pro 7840U 3.3 GHz processor and 64 GB of DDR4 RAM.

To assess which steps of the geodesic algorithms contribute most to the overall computation time, the times for each of the steps (origin placement, generator tracing, heat computation, and field derivation) are recorded and their contributions are plotted in Fig. 13. These contributions represent the mean over all 200 computations. In the complete algorithm, the origin placement makes up about 6.8% of the overall computation time with a mean step time of 8.3 ms. Generator placement takes only about 3.6 ms and makes up about 3% of the overall computation time. The largest contributor to computation time in both implementations is the core heat computation, which takes about 73.22 ms and accounts for about 60% of the total time. Another large contribution is the subsequent field derivation with 36.5 ms or about 30%. Both the heat computation and field derivation are significantly reduced by the fast geodesic algorithm, which omits one solve from the heat step and the gradient computation from the fields step. This results in just over 50 ms for the heat computation, which becomes the only relevant contributor, accounting for over 75% of the computation time. The field derivation is reduced to just under 4 ms and becomes the least significant contributor.

The comparison of the kinematic and geodesic draping approaches is first pointed towards their computational complexity. The main step of the kinematic draping approach is the iterative placement of nodes, each constrained by two adjacent neighbors. The placement method differs between implementations, but is generally carried out sequentially for each of the nodes. This results in linear scaling with the number of placed nodes in the resulting quad-mesh. The initial step of origin placement and generator tracing contributes only little to computation time based on the empirical analysis. They are also of low complexity, with the origin placement being $O(1)$ and the iterative generator tracing being $O(m < n)$ for m unfolded triangles. As seen in Fig. 13 the main steps in the geodesic methods are the heat and angle field computations. The angle field computation according to Eq. (8) consists only of operations with linear scaling and therefore has an overall complexity of $O(n)$. The main contributors to the complexity of the heat computations are the two sparse linear solves of the two subsequent heat method executions and especially their prefactorisation. For an initial

computation the prefactorization step dominates the overall complexity of the method. The current implementation uses the Eigen solver with a complexity of about $O(n^{1.5})$. The back-substitution for the solves of the heat diffusion and Poisson problem then only contributes with a complexity of $O(n \log n)$. For sufficiently small and regular meshes the complexity of these solves will share an approximately linear $O(n)$ complexity with the remaining operations of the matrix assembly and the normalization step. Because of this combination of complexities it is crucial for our method to preserve mesh geometry between solves, which allows the sharing of one prefactorization. With a shared prefactorisation its $O(n^{1.5})$ complexity gets amortized with repeated solves and for common CAD mesh geometries the scaling of the geodesic draping algorithm can achieve approximately linear $O(n)$ scaling. This behavior was confirmed by the empirical scaling investigation, which is elaborated in Section 4.2 and showed $n^{1.42}$ scaling for an initial draping computation and $n^{1.07}$ for a subsequent run on isotropic mesh geometries with 8-60,000 points.

While the scaling of the geodesic approach is slightly worse than that of the kinematic approach, one clear advantage of the geodesic algorithm is that it works directly on the triangulated mesh and can in theory leverage intrinsic triangulations. The kinematic grid on the other hand is a uniform rediscritization by construction. In practice this means that the actual number of vertices required to represent the underlying geometry with the same accuracy can be lower with our method. While the kinematic draping scales linearly with the number of grid vertices that is required to sufficiently approximate the underlying mesh, the geodesic approach scales approximately linearly with the number of nodes in the input mesh itself. This means the geodesic draping algorithm will show better scaling for parts that only require locally high mesh resolution, especially if the mesh is still not vastly irregular and sufficiently small. The kinematic approach, on the other hand, scales better for very large meshes that require a high overall mesh resolution anyways, especially if only single or few solves are required. In general, though, the computational scaling behavior of the approaches is similar and well-behaved. Because of their (approximately) linear scaling, neither algorithm experiences combinatorial explosions, and both are suited even for large node counts in the tens of thousands and more. So especially for repeated solves the difference in their computational efficiency is dominated rather by their constant cost.

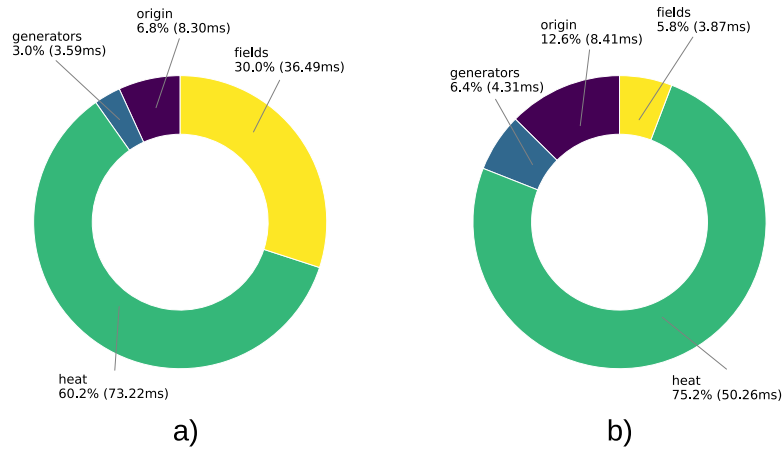


Fig. 13. Contributions of the individual computation steps to the overall computation time of the complete (a) and fast (b) geodesic draping algorithm.

The constant cost will always be a combination of algorithmic formulation and implementation efficiency. In our method the main contributor to computation time is the heat diffusion, a prefactorized sparse linear solve with a subsequent vector normalization. The complete method adds a second solve for the Poisson problem and the gradient computation. The advantage here is that solving the sparse linear systems can leverage decades of development in the underlying solvers. Still, we can at this point not prove whether there is a potentially more efficient implementation of the kinematic draping algorithm that could have a significantly lower constant cost as well. For now, we rely on the empirical investigation of computation time to demonstrate the potential of our method.

To assess and compare the actual computation speed, both algorithms are again used to compute shear for random starting points on the square pyramid geometry with varying wall angles. 200 points in a 30x30 grid are generated and used as input. The input geometry is isotropically remeshed, and the grid size and resolution are chosen such that both algorithms use a similar node count close to 13750 nodes (13689 for kinematic and 13749 for geodesic draping). In addition to the efficient kinematic draping algorithm and the complete geodesic draping algorithm, the fast geodesic draping algorithm is employed as well. All algorithms are run in direct succession on the same Lenovo ThinkPad P14s using an AMD Ryzen Pro 7840U 3.3 GHz processor and 64 GB of DDR4 RAM. The time of each iteration and for each step of the algorithms is recorded. Fig. 14 shows the quotient of the cumulative sum of the iteration times of the kinematic algorithm and those of the geodesic algorithms. It can be seen that their computational speed differs significantly. Fig. 14 clearly shows the speed gain from prefactorization in the geodesic algorithms, which was theoretically discussed before. While the initial solve is about fifteen times faster than the kinematic algorithm, the overall speed advantage continues to improve with each subsequent solve. After about 100 solves, the overall computation time is entirely dominated by the prefactored solves, and a mostly constant factor is established. While the kinematic draping algorithm takes about 18 min to compute all 200 shear angle fields, the geodesic draping algorithm performs the same number of computations in just over 24 s, making it about 44 times faster. The fast geodesic algorithm again increases the computation speed by almost a factor of two, resulting in a computation time of just over 13 s and an over 80-fold increase in computation speed compared to the kinematic implementation.

These results indicate that our current method and implementation clearly surpass the kinematic state of the art in terms of computation time and will likely do so for most practically relevant meshes. While very large and irregular meshes could at some point lead to a narrowing between the methods due to the slightly superlinear scaling of our approach, utilization of more efficient solvers like cholmod could bring

down the current constant cost of our method by another order of magnitude and applications with only few repeated solves could benefit greatly from GPU parallelized factorization. However, the kinematic draping algorithm most likely has further potential for acceleration as well. Future research into both approaches will have to show which one will finally end up in the most efficient implementation.

4.2. Robustness and scaling

The four comparatively simple geometries with similar edge lengths and point counts in Sections 4.1 and 4.3 give limited insight into the robustness and scaling behavior of the proposed draping algorithm. Therefore, the draping algorithm was also applied to complex freeform surfaces in the form of topographical models of Iceland and Mount Everest. The model of Mount Everest was made up of 166,149 points and the model of Iceland was made up of 32,000 points. The draping algorithm was able to successfully predict the shear angle distribution and fiber tow locations for both of these complex geometries, as shown in Figs. 15(a) and 16(a). Still, these tests also revealed two remaining limitations of the approach. In the heat method the time duration of the diffusion step is calculated from the mean node distance h and a constant k_{time} according to $t = k_{time} * h^2$. Usually good results can be reached with the constant being left at one. However, we observed that for very fine meshes and at large distances, the heat gradient becomes so small that the method becomes numerically unstable and a reliable shear computation is no longer possible. This behavior can be seen in Fig. 15(b), where the shear prediction breaks down in the mesh corners, at which the geodesic distance to at least one of the generators surpasses a certain threshold. This problem can be mitigated by choosing a slightly larger k_{time} . As shown in Fig. 15(a), a k_{time} of 2.5 was sufficient for this mitigation and only led to minor differences in the overall prediction.

Another limitation is one which goes back to assuming the generators as straight geodesics. The assumption of straight geodesic generators is one that is reasonable in many cases. However, for certain combinations of geometry, origin and generator direction the straight geodesic can wrap around local hills in the elevation profile. An extreme case of this is shown in Fig. 16(b), where the generator got wrapped around three such hills, which led to an obviously unrealistic generator placement. Such critical combinations can usually be avoided by slight perturbations to either origin position or direction. However, to make the method more robust against such problems, and therefore more realistic in modeling these edge cases, the straight geodesic assumption would need to be adjusted. One possible avenue would be to combine straight geodesic tracing with information on the geodesic distance from the origin. For example the angle preservation from the straight geodesic trace could be combined with a penalty for

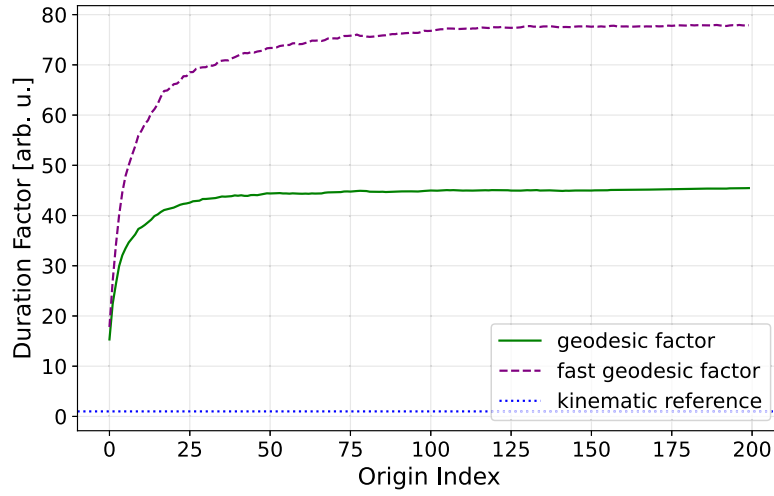


Fig. 14. Duration factor computed as the quotient of the cumulative computation time of the kinematic draping algorithm divided by that of both geodesic draping variants over 200 random starting points.

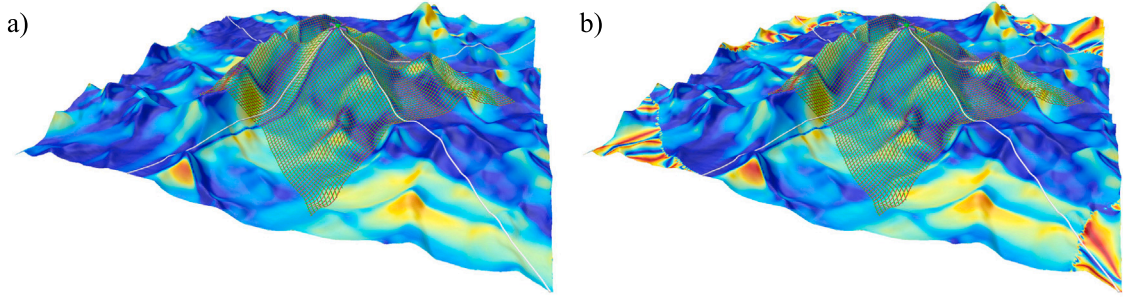


Fig. 15. Geodesic draping performed on a topographic map of Mount Everest: (a) successful prediction with increased time coefficient of 2.5 (b) breakdown of shear prediction for large distances from generators with standard coefficient of 1.

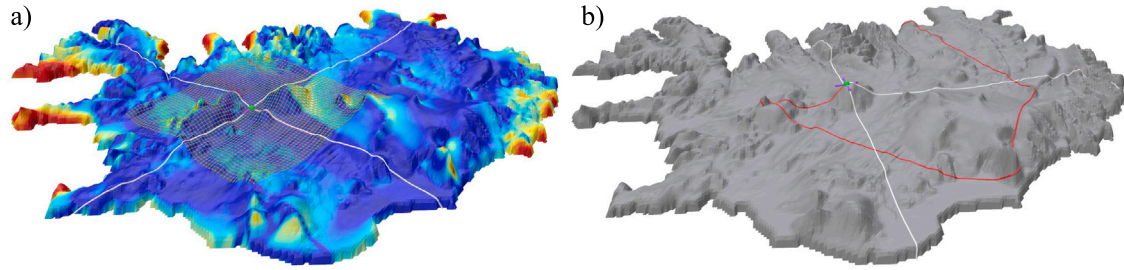


Fig. 16. Geodesic draping performed on a topographic map of Iceland: (a) successful shear angle prediction (b) one of the generator segments wrapping around hills in the topography.

large divergences from the geodesic distance or elevation gradient. The extent of such a penalty might also be a possible tuning parameter to tailor the method to different material types.

In addition to these initial robustness tests, the draping algorithm was applied to several variants of a cone geometry with varying edge lengths to assess the scaling behavior of the method. The computational scaling behavior was benchmarked across 20 mesh densities, with mesh sizes between roughly 5000 and 60,000 points. The draping prediction for each of these meshes was computed five times. Both the complete and the fast geodesic draping algorithms were tested and the initial computation (cold start) was compared to a second prefactorized computation (warm start). Both computation times as well as the shear angle predictions were recorded for all 100 computations and are shown in Fig. 17. The results confirm a super-linear

scaling for the initial execution of both algorithms. The initial computation scales approximately as $n^{1.37}$ and $n^{1.42}$ for the complete and fast algorithm respectively with the mesh point count n , which is close to the theoretical complexity of $n^{1.5}$. The approximately linear scaling after prefactorization discussed in Section 2.1 is confirmed by the empirically observed scaling of $n^{1.07}$ and $n^{1.06}$ respectively. The theoretical $\log n$ component therefore seems to be negligible at the investigated point counts. Overall the fast solver offers a consistent reduction in initial computation time relative to the complete solver across all mesh sizes and for both algorithm variants. The total time of the prefactorized computations is greatly reduced compared to the initial one and remains well below 1 s even at 60,000 points. The computed shear angle results show that both mean and maximum shear predictions remain mostly stable across the full range of mesh

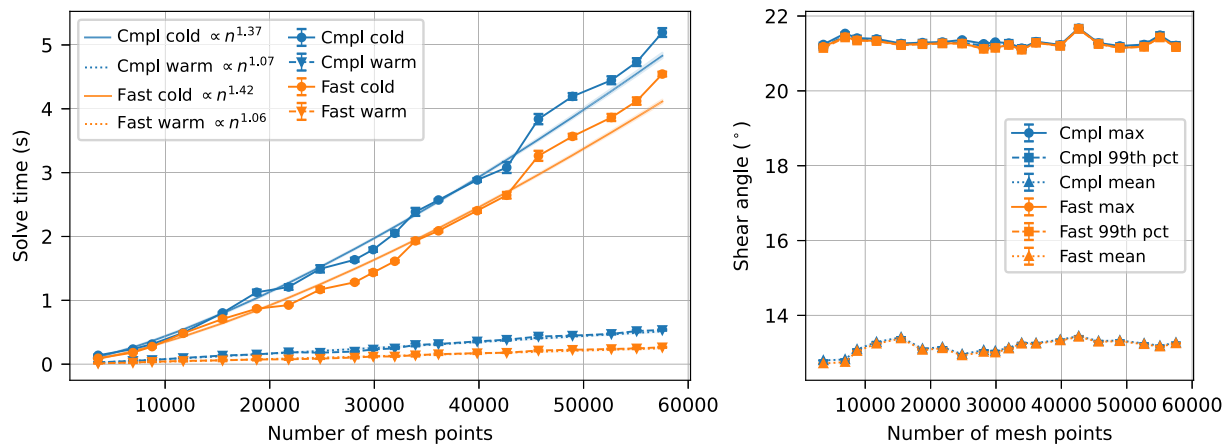


Fig. 17. Scaling behavior investigation with a cone test geometry: (l) solve time scaling of complete and fast geodesic draping (r) stability of the shear angle prediction over different mesh densities.

densities and for both algorithm variants. This stability suggests that the algorithms are robust with respect to the mesh discretization itself. The only larger effect seems to be the lower mean shear prediction for low mesh densities, which was likely a result of the loss of geometric accuracy as the discretization dropped below the original geometrys point count of 8322.

4.3. Optimization results

The optimization results for the three test geometries, cone, dome, and eccentric cone, are shown in Fig. 18. Additionally Fig. 19 exemplifies the progression of the evolution for the evolutionary algorithm, using the complete geodesic draping algorithm and the cone geometry. Note that it records the query points and function values per function evaluation and not per generation. Both visual representations already show that the optimization correctly converged towards the intuitively correct locations. Fig. 19 additionally illustrates that the evolutionary optimizer seems to behave well, at least for the given geometry, as it shows an initial wide cast exploration and a clear convergence plateau towards the end. However, significantly different geometries might require different parameter sets to achieve the intended behavior. To further assess the optimizations, the number of iterations, time, final shear, final position, and its deviation from the intuitively optimal position are listed in Table 2. All results are averages over five computations with random initial values that are shared between the methods, but differed between geometries. Using the complete geodesic draping algorithm, the initial evolutionary algorithm performs about 70–80 iterations in about 6.5 s to converge for all three geometries. The subsequent modified Powell algorithm takes longer, with an average of 135.6–248.6 iterations depending on the geometry. This number of iterations results in an average computation time of 12–22.2 s. Therefore, the final optimization result is obtained for all geometries in under 30 s total. Considering how closely the optimization converges to the expected solution, there are some differences between the geometries. For cone and dome, the total distance to the expected solution is already close to or even below 1 mm after the initial evolutionary optimization. The additional gradient descent effectively achieves the goal of refining the result. This is shown by a reduced average distance to the expected solution of 0.05 mm for the cone and 0.134 mm for the dome. For the eccentric part, the Y component of the resulting position also converges very closely to the expected value of 0. However, the X component shows a consistent offset of 1.938 mm. The solution is consistent over all five trials up to 3 decimal places. Looking at the results from the fast geodesic draping algorithm, a slightly larger distance from the expected solution is observed for the cone and the dome, although the final distance remains about 0.2 mm for both

geometries. The final shear prediction is approximately equal to that of the complete implementation, with a 0.146° increase for the cone, a 0.039° increase for the eccentric part, and a 0.056° decrease for the dome. As expected, the computation time is significantly reduced using the fast implementation. Using this variant of the algorithm, the combined computation time of both optimization routines remains below a total of 10 s for all three test geometries.

4.4. Forming results

Both incremental forming strategies could be executed without exceeding process or material limits. The four resulting cone geometries were scanned with a GOM ATOS 5M structured light 3D scanner. In the corresponding ZEISS Inspect 2023 software, geometric deviations from the target geometry were assessed by performing a least-squares alignment and plotting the point-wise distance between the as-is and target geometries. The flat outer sheet was included in the target geometry, to better assess wrinkling and the overall fabric deformation. Images of two of the parts as well as scans of all four experiments are shown in Fig. 20 including the color coded geometric deviation of the scans. The minimum and maximum geometric errors, as well as the standard deviation of the errors, were extracted and are shown in the respective scan images as well.

Qualitatively, there are two main differences between the parts using pure spiral tool-paths and those that ended on a draping shape. The first difference is the manifestation of wrinkling in the fabric: While the tip region of all cones is wrinkle free, the bases of the spiral parts show severe wrinkling along their entire circumference. Wrinkles in the spiral parts manifest as sharp and mostly straight ridges with a height of up to 3.5–4 mm. They reach into the base of the cone, before they transition into the general overforming at the base. In contrast, there is significantly less wrinkling in the draping parts. Wrinkles are on average significantly smaller in length, width and elevation. They manifest as elliptical hills that flatten out before reaching the base area of the cone and are therefore constrained to the area of the flat sheet, which was not reached by any tool-paths. It can thus be concluded that the addition of a final draping path did achieve its goal of mitigating wrinkles in the final formed part. The second main difference between the two strategies is the shape of the geometric deviation. Both strategies resulted in overforming at the base of the cone and underforming in the tip region. The location and extent of the deviations are similar, but their general shape differs significantly. While the spiral parts show radially symmetric deviations, the fiber following pass introduced a cross-shaped deviation, which is most pronounced in the area from the tip's transition radius down the sides, which coincide with the generator directions. The overforming at the

Table 2

Optimization results for three test geometries using the complete and fast variants of geodesic draping separated into the initial differential evolution and subsequent Powell optimization step. All values represent the average over five runs with random starting positions. Starting positions are shared between methods, but not between the geometries.

Part	Method	Iterations	Time [s]	Shear [°]	X	Y	Error X	Error Y	Error Norm
Cone	Evolution	74.40	6.759	21.578	0.165	0.153	0.710	0.427	0.879
Cone	Powell	248.60	22.151	21.159	-0.028	-0.040	0.028	0.040	0.050
Cone	Fast Evo	70.80	2.843	21.792	0.042	0.909	0.777	0.910	1.312
Cone	Fast Pow	145.40	5.632	21.305	0.206	0.038	0.206	0.038	0.211
Dome	Evolution	70.80	6.467	29.608	-0.307	0.269	0.321	0.386	0.612
Dome	Powell	135.60	12.039	29.117	0.129	0.034	0.129	0.034	0.134
Dome	Fast Evo	69.60	2.799	29.693	0.352	0.493	0.412	1.103	1.340
Dome	Fast Pow	166.80	6.601	29.061	0.093	-0.025	0.152	0.078	0.185
Eccentric	Evolution	78.00	6.470	9.581	-26.585	0.112	1.585	1.072	2.089
Eccentric	Powell	150.00	12.407	9.282	-26.938	0.013	1.938	0.013	1.938
Eccentric	Fast Evo	87.60	3.317	9.566	-26.054	0.359	1.994	0.463	2.076
Eccentric	Fast Pow	177.80	6.508	9.321	-26.879	-0.232	1.879	0.232	1.893

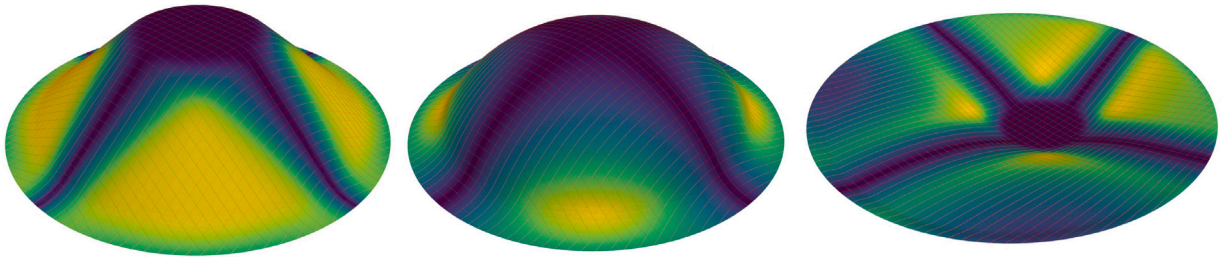


Fig. 18. Qualitative shear distribution after starting point optimization for the test shapes cone, dome, and eccentric cone (l-r) with lighter color indicating higher shear angle.

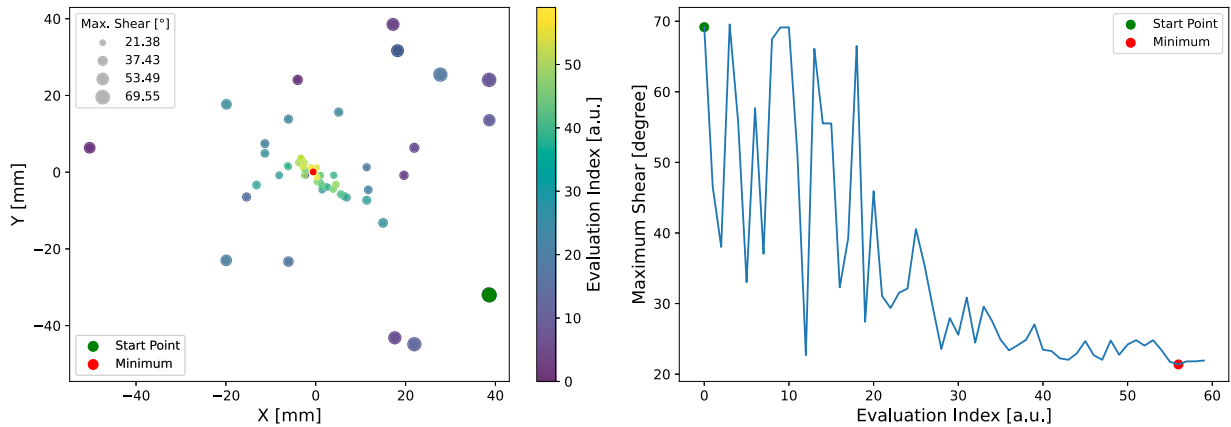


Fig. 19. Progress of the evolutionary optimization using the complete geodesic draping algorithm for the cone geometry with a spacial plot (l) of the query points and computed shear angles and a convergence plot (r) to further illustrate the change of the function value over the course of the optimization.

base transition radius is also biased towards these directions. The extent of the forming deviations is similar for both strategies. Overforming at the base reached 2–2.3 mm for the spiral strategy and was similar at the generator cross for the draping strategy. Underforming at the tip was about 2.3 mm for the spiral strategy and even larger in the corners of the sheet. For the draping parts, significant underforming only occurred in the cross-shaped tip region, but was even slightly higher in the maximum (2.4–2.7 mm). Although it should be noted that the best fit alignment will distort the absolute values of these two effects to some extent.

In summary, the draping paths enabled the formed region of the sheet to be free from wrinkle defects and therefore achieved their intended primary goal. However, the geometric accuracy still remained similar to that of the spiral parts, because the cross-shaped deformation impairs the quality of the draping parts. Similar cross-shaped deformations were observed in [11], where they were attributed to the limited

robot stiffness. To investigate this possibility the forming forces during the final pass of each strategy (fiber-following vs. spiral) were recorded and are shown in Fig. 21. Independent of the path strategy, both wall angle and depth increment increase with the forming depth, due to the method of intermediate shape creation. The spiral path increases them gradually and on rotational paths. On the other hand, the fiber-following path covers the largest increase on its initial passes and in a non-rotational manner. This difference is also reflected in the force plots. The force of the spiral increases gradually towards 1000 N. The fiber-following path instead shows a series of spikes and drops with diminishing amplitude. The highest forces of over 1500–1750 N are reached during the initial passes towards the tip. The highest forces in the draping pass therefore coincide with the location of the cross-shaped deformation along the generator directions. The amplitude of the forces makes a significant deflection likely, as previous research reported deflections exceeding 3 mm for forming forces around 2000

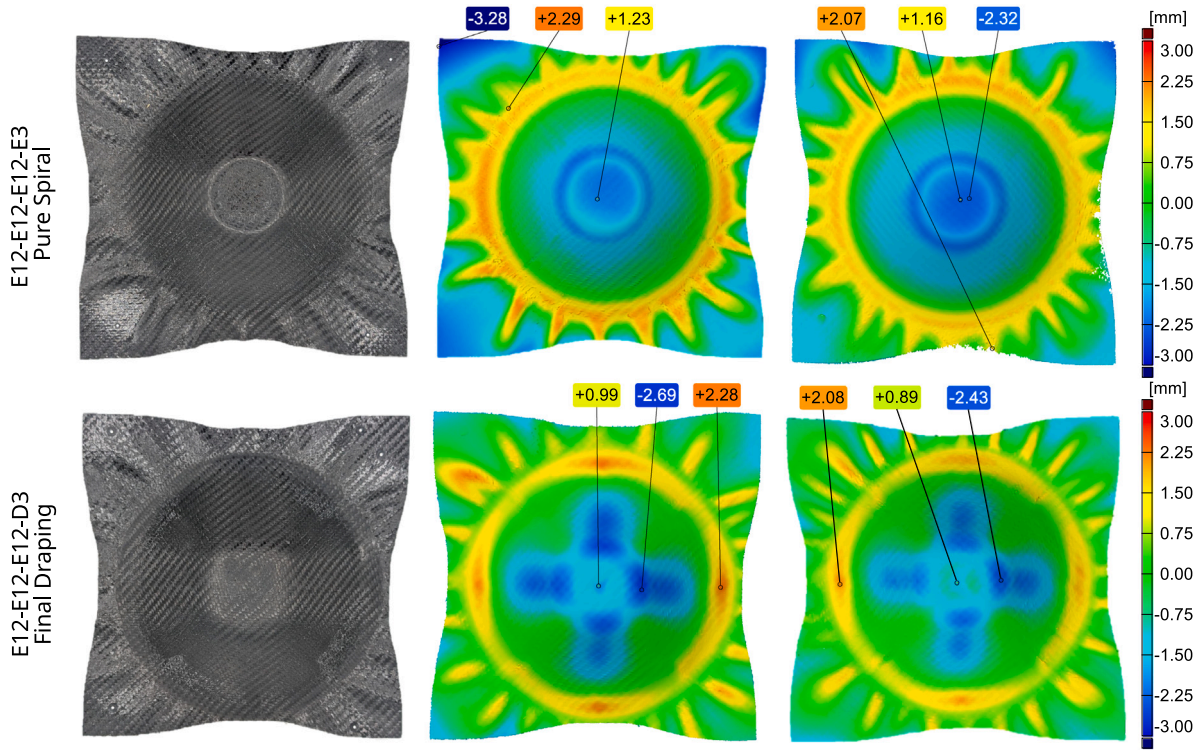


Fig. 20. Incremental forming results comparing a cone formed with solely elevation-based spiral toolpaths (E12-E12-E12-E3) and a cone formed with a fiber-following draping toolpath in the final shape (E12-E12-E12-D3) with identical intermediate shape succession. Surface scans and the best-fit surface deviations are included for all four experiments including repeats. Flags indicate maximum, minimum and standard deviation of the geometric errors. The camera images correspond to the scans on the left.

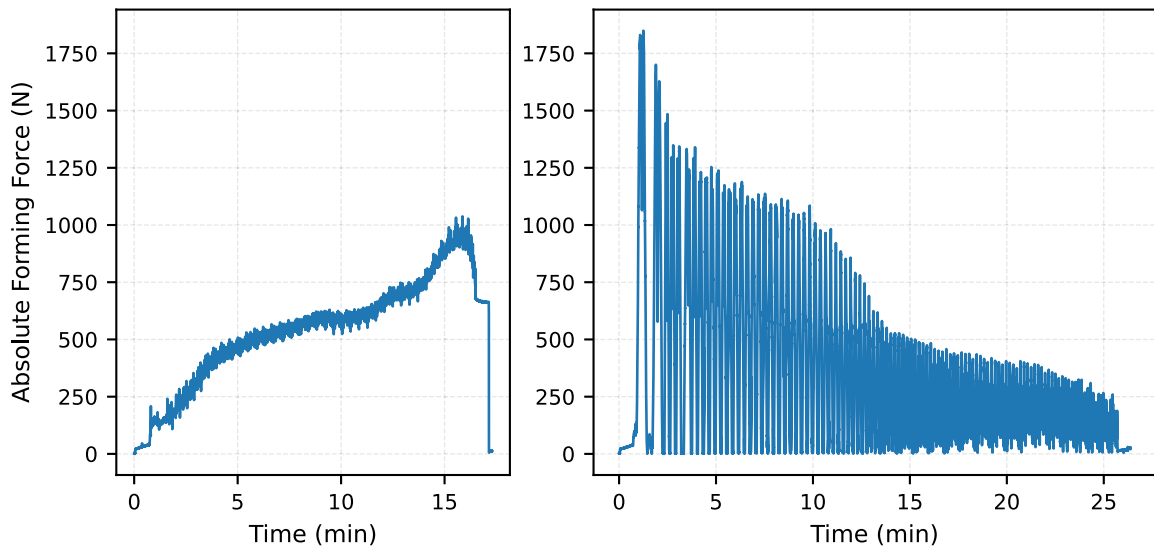


Fig. 21. Comparison of the absolute forming force as measured at the guiding robot during the final intermediate shape of the pure spiral strategy (l) and the strategy with a final fiber-following path (r).

N during incremental forming with KUKA KR340 robots [44], which are similar to the KUKA KR300 used in this work.

5. Discussion

The results of this work show that the shear angle prediction of the newly developed geodesic draping algorithm is similar to that of a state-of-the-art kinematic draping simulation. This is especially the case when the curvature transverse to the generators is small. As generator

placement gets closer towards ridges, the approximations seem to diverge more significantly. Further investigations are needed to assess which approach better approximates real textile behavior in these cases and for different materials. In any case, some divergence in the results is to be expected given the different assumptions underlying the two algorithms. Still, the results imply that, just as kinematic simulations, although developed for woven fabrics, have been used to model UD-Layups, the geodesic approach is likely applicable to woven fabrics as well, although its assumptions are usually associated with UD-plyies. An

interesting aspect to investigate is the influence of the diffusion time on the draping prediction. The robustness investigation already revealed that very large and fine meshes will necessitate an increase of the diffusion time, but it would also be interesting to investigate whether the diffusion time can be used to tailor the predictions to different materials. To this end, the use of scalar and vector fields to model the fabric also opens up further possibilities to incorporate constraints and modifications. For example the direction fields could be smoothed at local shear angle spikes or modified by slight oscillations to incorporate fiber waviness. However, such modifications would move the solution away from an equal tow spacing and thus would be more fitting in a modeling context, and less for the proposed path-planning method. The generator placement modification by incorporating geodesic distance or elevation information into the angle constraint, which was already discussed in Section 4.2, would be a modification that would be applicable in both contexts. However, independent of the specific application and possible modifications, geodesic draping, like kinematic draping, should still be viewed as a basic approximation scheme. The material independence might be an advantage, as it does not require specific knowledge of the material system to be applied. Yet this also means that both approaches incorporate only very basic material-specific behavior and can therefore only provide a rough approximation. Additionally, they are incapable of modeling wrinkling. Consequently, they will produce inaccurate results when applied to shapes that exceed the locking angle of the fabric.

The question of modeling accuracy is less crucial for the use case of path planning for incremental forming. The equidistant nature of the approximation inherently combines two important properties of draping-aware incremental forming: paths are at least approximately fiber-following and have constant scallop width. This makes geodesic draping a particularly well-suited approximation, even for textiles whose real behavior more closely matches a kinematic approximation or requires more complex modeling. Other domains in which this property might be useful include 3D printing and automated fiber placement (AFP), where the “textile” is not draped but created on a curved surface.

The optimizations carried out in the scope of this work demonstrate that the computational efficiency of the geodesic approach can be leveraged to optimize draping starting points within seconds. There is still room for improvement in the meta-optimization of the optimization routines, which might reduce convergence times even further. In many applications, the submillimeter accuracies achieved with just differential evolution might be sufficient. In these cases, optimization times are already as low as 2.8 s. However, it should be noted that run times will increase with larger meshes and more nodes. Furthermore, the performance should be studied more extensively using more complex geometries and real-world experiments. Still, the initial results show that for simple geometries, the optimization converges extremely fast to the global optimum. The only meaningful offset from the expected location is observed with the eccentric cone. However, the consistency of the offset makes it very unlikely that the optimization is getting stuck in a local minimum. Instead, it is more likely that the true optimal solution actually shows a slight shift, as the non-trivial shape of the connecting surface between the tip and base makes the intuitive optimal location less certain than for the dome and cone.

Aside from its general advantage for fast optimizations, the high computation speed also allows for the seamless integration of draping approximations into the path planning workflow for incremental forming in particular. If the starting point is non-trivial, an optimized starting position can be computed in as little as 3–30 s. Alternatively, the starting point can be chosen based on visualizations that are updated in real-time. The actual path computation is almost instantaneous, enabling fast parameter iterations and a smooth workflow. While this work chooses incremental forming as its primary field of application, the integration of a near real-time draping prediction into other existing workflows, for instance, in hand-layout, is likely to be

beneficial as well and should be studied in the future. Going further into the application of incremental forming, the results of this work show the practical benefit of the developed geodesic draping based path planning pipeline. The paths are not only executable, but actually achieve the goal of wrinkling reduction by introducing just one final draping pass. It seems likely that this could be combined with a general improvement of geometric accuracy once the cross-shaped deformations are addressed. The developed pipeline, therefore, successfully implements the idea of fiber-following paths [11] and shows that a combination of conventional and fiber-following paths can mitigate the need for a high number of intermediate shapes while still benefiting from the draping-aware paths. However, at this point, cross-shaped underforming remains an issue to be addressed. The results from the force analysis showed that the force induced deflection of the end effector is likely a major contributor to this defect. However, other contributions might be relevant as well, for example the changing internal stress state of the material during different forming strategies. A more in-depth study of different path planning strategies, investigating the influence of intermediate shapes and path planning sequences on part accuracy, is likely to further improve the quality of the forming results.

6. Conclusion

In this work, a novel geodesic draping algorithm was presented to offer an efficient alternative to kinematic draping that results in equidistant fiber bundles and works directly on the input mesh geometry. This algorithm was leveraged to build a fast start point optimization, a real-time visualization tool, and a path planning pipeline for the incremental forming of CFRP. All steps of this development were demonstrated in a number of numerical studies and real-world experiments. The draping algorithm was benchmarked against a state-of-the-art kinematic draping algorithm, which represents its closest alternative. The optimization was performed on geometries with known best starting points to verify the results against a solid ground truth. The path planning pipeline was utilized to generate paths for a simple cone frustum geometry, which was subsequently formed on a double-sided hot incremental forming setup. The resulting parts were compared to parts formed by a conventional elevation-based spiral strategy. The results of the computational analysis show that the geodesic draping algorithm produces shear distributions similar to those of the kinematic draping simulation, although some differences in the behavior were observed. However, the newly developed algorithm shows a significant speed advantage of about 40–80 times for repeated solves, depending on whether just shear or also fiber locations are required. Because of its high computational efficiency, the optimization converges quickly to the true solution, requiring 2.8 to 29 s, with an error margin of 0.05 to 2.1 mm, depending on the test geometry and the draping and optimization algorithms used. The high speed also allows for a visualization of shear distribution, fiber locations, and required flat cut-out that can be updated in real-time. The final formed parts show the benefit of using a drape-aware path planning pipeline based on geodesic draping. Introducing just a single fiber-following pass leads to parts without any wrinkles in the formed areas and reduces overall wrinkling compared to the conventional approach. However, cross-shaped deviations remain that warrant further investigations into more advanced strategy combinations.

Overall, the results of this work show that geodesic draping can enable fast and robust visualization and optimization tools. Furthermore, it is a valuable tool in the area of digital rapid manufacturing, where it can be a foundation for automated path planning. The path-planning pipeline derived in this work already proved itself useful for advancing moldless, automated manufacturing of CFRP parts by incremental forming. Likewise, the high computational efficiency, mesh-based approach and equidistant nature are promising for other applications as well. Consequently, future research should advance into multiple domains: On the simulation side, further investigations are required

to better understand to what extent and for which materials kinematic and geodesic draping are applicable, which is better suited in certain situations, and how their differences affect the final outcome of, e.g., an optimization pipeline. On the process side, the potential unlocked by the newly developed path planning algorithm needs to be further expanded by studying its combination with other path planning algorithms and intermediate shape sequences. Lastly, looking at further applications, it seems promising to develop similar path planning schemes for other applications, such as additive manufacturing and automated fiber placement, and to leverage the fast computational potential in other application domains, such as hand-layup or structural optimization.

CRediT authorship contribution statement

Doran Nettig: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jan-Erik Rath:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Investigation. **Johann Kipping:** Writing – review & editing, Methodology, Conceptualization. **Thorsten Schüppstuhl:** Supervision, Resources, Project administration, Funding acquisition.

Funding

This work was supported by the German Federal Ministry for Economic Affairs and Energy [grant number 20Q1917C].

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

Data will be made available on request.

References

- Breuer U, Mitschang P. Einführung. In: Neitzel M, Mitschang P, Breuer U, editors. *Handbuch verbundwerkstoffe*. München: Carl Hanser Verlag GmbH & Co. KG; 2014, p. 1–30. <http://dx.doi.org/10.3139/9783446436978.001>.
- Boisse P, Akkerman R, Carlone P, Kärger L, Lomov SV, Sherwood JA. Advances in composite forming through 25 years of ESAFORM. *Int J Mater Form* 2022;15(3). <http://dx.doi.org/10.1007/s12289-022-01682-8>.
- Lim T-C, Ramakrishna S. Modelling of composite sheet forming: a review. *Compos Part A: Appl Sci Manuf* 2002;33(4):515–37. [http://dx.doi.org/10.1016/S1359-835X\(01\)00138-5](http://dx.doi.org/10.1016/S1359-835X(01)00138-5).
- Krogh C, Bak BLV, Lindgaard E, Olesen AM, Hermansen SM, Broberg PH, Kepler JA, Lund E, Jakobsen J. A simple MATLAB draping code for fiber-reinforced composites with application to optimization of manufacturing process parameters. *Struct Multidiscip Optim* 2021;64(1):457–71. <http://dx.doi.org/10.1007/s00158-021-02925-z>.
- McAnulty T, Jeswiet J, Doolan M. Formability in single point incremental forming: A comparative analysis of the state of the art. *CIRP J Manuf Sci Technol* 2017;16:43–54. <http://dx.doi.org/10.1016/j.cirpj.2016.07.003>.
- Liu Z, Li Y, Meehan PA. Vertical wall formation and material flow control for incremental sheet forming by revisiting multistage deformation path strategies. *Material Manuf Process* 2013;28(5):562–71. <http://dx.doi.org/10.1080/10426914.2013.763964>.
- Jeswiet J, Micari F, Hirt G, Bramley A, Dufloy J, Allwood J. Asymmetric single point incremental forming of sheet metal. *CIRP Ann* 2005;54(2):88–114. [http://dx.doi.org/10.1016/S0007-8506\(07\)60021-3](http://dx.doi.org/10.1016/S0007-8506(07)60021-3).
- Nettig D, Rath J-E, Kipping J, Schüppstuhl T. Classifying parameters and target variables in incremental sheet forming of fiber reinforced polymers. *Solid State Phenom* 2026;389:11–25. <http://dx.doi.org/10.4028/p-8f1QUK>.
- Fiorotto M, Sorgente M, Luchetta G. Preliminary studies on single point incremental forming for composite materials. *Int J Mater Form* 2010;3(S1):951–4. <http://dx.doi.org/10.1007/s12289-010-0926-6>.
- Al-Obaidi A, Kunke A, Kräusel V. Hot single-point incremental forming of glass-fiber-reinforced polymer (PA6gf47) supported by hot air. *J Manuf Process* 2019;43:17–25. <http://dx.doi.org/10.1016/j.jmapro.2019.04.036>.
- Rath J-E, Schüppstuhl T. Tool path strategies for single point incremental forming of fiber-reinforced thermoplastic sheets. In: *Material forming: ESAFORM 2024. Materials research proceedings*, Materials Research Forum LLC; 2024, p. 641–50. <http://dx.doi.org/10.21741/9781644903131-71>.
- Xu P, Li X, Li X, Wang Y, Zhang W, Dong H, Zhi Q. Defects in double-sided incremental forming of woven fabric prepreg: experimental and numerical analysis. *Thin-Walled Struct* 2025;113229. <http://dx.doi.org/10.1016/j.tws.2025.113229>.
- Nettig D, Rath J-E, Tiedemann JO, Schüppstuhl T. Double-sided incremental forming of continuous fiber reinforced thermoplastics. *Compos Part A: Appl Sci Manuf* 2026;210:110096. <http://dx.doi.org/10.1016/j.compositesa.2026.110096>, URL <https://www.sciencedirect.com/science/article/pii/S1359835X26005439>.
- Rath J-E, Nettig D, Schüppstuhl T. Hot double-sided incremental forming of fiber-reinforced thermoplastics: Process capabilities and challenges. *Solid State Phenom* 2026;389:1–9. <http://dx.doi.org/10.4028/p-u5oc0K>.
- Nettig D, Rath J-E, Schüppstuhl T, Frank J, Riecken B, Keun C-A. Hot double sided incremental forming of continuous fiber reinforced thermoplastics: Process analysis and system design. In: *Material forming: ESAFORM 2025. Materials research proceedings*, Materials Research Forum LLC; 2025, p. 1267–76. <http://dx.doi.org/10.21741/9781644903599-138>.
- Rath J-E, Schwiager L-S, Schüppstuhl T. Robotic die-less forming strategy for fiber-reinforced plastic composites production. *Procedia CIRP* 2022;107:1281–6. <http://dx.doi.org/10.1016/j.procir.2022.05.145>.
- Long AC, Clifford MJ. Composite forming mechanisms and materials characterisation. In: *Composites forming technologies*. Elsevier; 2007, p. 1–21. <http://dx.doi.org/10.1533/9781845692537.1>.
- Wang JRP. The draping of woven fabric preforms and prepregs for production of polymer composite components. *Compos Part A: Appl Sci Manuf* 1999. [http://dx.doi.org/10.1016/S1359-835X\(98\)00187-0](http://dx.doi.org/10.1016/S1359-835X(98)00187-0).
- Potter K, Ward C. Draping processes for composites manufacture. In: *Advances in composites manufacturing and process design*. Elsevier; 2015, p. 93–109. <http://dx.doi.org/10.1016/B978-1-78242-307-2.00005-1>.
- Elkington MP, Mistry PJ, Johnson MS, Ou H. Hybrid vacuum-robotic forming of reinforced composite laminates. *J Reinf Plast Compos* 2022;0731684422113521. <http://dx.doi.org/10.1177/0731684422113521>.
- Skordos AA, Suitcliffe MPF, Klintworth JW, Adolfsson P. Multi-objective optimisation of woven composite draping using genetic algorithms genetic algorithms .doc. *Int Conf SAMPE EUROPE* 2006.
- Fitas R, Hesseler S, Wist S, Greb C. Kinematic draping simulation optimization of a composite B-pillar geometry using particle swarm optimization. *Heliyon* 2022;8(11):e11525. <http://dx.doi.org/10.1016/j.heliyon.2022.e11525>.
- Sun X, Belnoue JP-H, Thompson A, Said BE, Hallett SR. Dry textile forming simulations: A benchmarking exercise. *Front Mater* 2022;9. <http://dx.doi.org/10.3389/fmats.2022.831820>.
- Hancock SG, Potter KD. The use of kinematic drape modelling to inform the hand lay-up of complex composite components using woven reinforcements. *Compos Part A: Appl Sci Manuf* 2006;37(3):413–22. <http://dx.doi.org/10.1016/j.compositesa.2005.05.044>.
- Mack C, Taylor HM. 39—The fitting of woven cloth to surfaces. *J Text Inst Trans* 1956;47(9):T477–88. <http://dx.doi.org/10.1080/19447027.1956.10750433>.
- Altair HyperWorks. Kinematic draping theory. 2022, URL https://2022.help.altair.com/2022/hwdesktop/hwx/topics/user_interface/kinematic_draping_approach_r.htm.
- Fengler B, Kärger L, Henning F, Hrymak A. Multi-objective patch optimization with integrated kinematic draping simulation for continuous-discontinuous fiber-reinforced composite structures. *J Compos Sci* 2018;2(2):22. <http://dx.doi.org/10.3390/jcs2020022>.
- Crane K, Weischedel C, Wardetzky M. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans Graph* 2009. <http://dx.doi.org/10.1145/2516971.2516977>.
- Sharp N, Soliman Y, Crane K. The vector heat method. 2020, <http://dx.doi.org/10.1145/3243651>.
- Feng N, Crane K. A heat method for generalized signed distance. *ACM Trans Graph* 2024;43(4):1–19. <http://dx.doi.org/10.1145/3658220>.
- Polthier K, Schmies M. Straightest geodesics on polyhedral surfaces. 2006, p. 30–8. <http://dx.doi.org/10.1145/1185657.1185664>.
- Sharp N, Crane K, et al. *GeometryCentral: A modern c++ library of data structures and algorithms for geometry processing*. 2019.
- Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE. Array programming with NumPy. *Nature* 2020;585(7825):357–62. <http://dx.doi.org/10.1038/s41586-020-2649-2>.
- Schroeder W, Martin K, Lorensen B. *The visualization toolkit, 4th ed*. Kitware; 2006.

- [35] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 10 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 2020;17:261–72. <http://dx.doi.org/10.1038/s41592-019-0686-2>.
- [36] Hagberg A, Swart P, Chult D. Exploring network structure, dynamics, and function using NetworkX. 2008, <http://dx.doi.org/10.25080/TCWV9851>.
- [37] Muntoni A, Cignoni P. PyMeshLab. 2021, <http://dx.doi.org/10.5281/zenodo.4438750>.
- [38] Sullivan B, Kaszynski A. PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK). *J Open Source Softw* 2019;4(37):1450. <http://dx.doi.org/10.21105/joss.01450>.
- [39] Hunter JD. Matplotlib: A 2D graphics environment. *Comput Sci Eng* 2007;9(3):90–5. <http://dx.doi.org/10.1109/MCSE.2007.55>.
- [40] Sharp N, et al. Polyscope. 2019, www.polyscope.run.
- [41] Qiang J, Mitchell C. A unified differential evolution algorithm for global optimization. 2014.
- [42] Powell MJD. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J* 1964;7(2):155–62. <http://dx.doi.org/10.1093/comjnl/7.2.155>.
- [43] Press WH. Numerical recipes: The art of scientific computing. 3rd ed. Cambridge UK and New York: Cambridge University Press; 2007.
- [44] Möllensiep D, Gorlas T, Kulesa P, Kühlenkötter B. Real-time stiffness compensation and force control of cooperating robots in robot-based double sided incremental sheet forming. *Prod Eng* 2021;15(5):683–99. <http://dx.doi.org/10.1007/s11740-021-01052-4>.