TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Wang, J., Liu, Y., Wu, J., & Westermann, R. (2026). Topology-aware stress analysis in shell structures. *Computer Methods in Applied Mechanics and Engineering*, *452*, Article 118770. https://doi.org/10.1016/j.cma.2026.118770

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Computer Methods in Applied Mechanics and Engineering 452 (2026) 118770

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Topology-aware stress analysis in shell structures

Junpeng Wang [a,*], Yingjian Liu [b,*], Jun Wu [c], Rüdiger Westermann [a]

[a] Technical University of Munich, Boltzmannstr. 3, Garching, 85748, Germany
[b] University of Georgia, Athens, 30602, USA
[c] Delft University of Technology, Landbergstraat 15, Delft, 2628 CE, The Netherlands

ARTICLE INFO

Keywords:
Shell structures
Principal stress lines
Topology analysis
Streamline seeding

ABSTRACT

We present a stable and accurate algorithm for tracing principal stress lines (PSLs) in shell structures, applicable to both first- and second-order triangular and quadrilateral elements. The algorithm operates directly in the isoparametric space of the elements, leveraging their inherent shape functions to account for curved geometry without resorting to artificial subdivision. This approach enables, for the first time, a consistent stress topology analysis for shell elements, including a rigorous treatment of stress degeneracies. Our PSL seeding strategy integrates stress topology with the curved shell surface, ensuring a uniform and consistent PSL distribution. We evaluate the algorithm's performance through a series of numerical experiments, demonstrating its utility for advanced stress analysis. Furthermore, we demonstrate the generation of a globally consistent, space-filling PSL structure, which is directly applicable to downstream tasks such as lightweight structural design. To support practical use, we provide a publicly available MATLAB implementation. The implementation features a unified file interface that supports diverse mesh types and is compatible with standard finite element method (FEM) output, offering a versatile tool for stress investigation and design evaluation in computational mechanics. The code is available at https://github.com/PSLer/PSLshell.

1. Introduction

The principal stress lines (PSLs) offer an intuitive and physically meaningful representation of how stresses propagate through a structure. By following the trajectories of principal stresses, PSLs reveal the underlying load paths that govern structural behavior, providing insights beyond what scalar stress measures can capture. For shell structures, PSLs are of particular importance: they visualize stress flow over curved surfaces and thus directly support engineering tasks such as aligning fibers in composite laminates, placing stiffening ribs along efficient load paths, and identifying critical regions that dominate the stress-carrying capacity of thin-walled components [1–5]. High-quality PSL generation and distribution, therefore, play a significant role in both the interpretation of structural mechanics and the guidance of practical design decisions.

As a trajectory-based representation of stress fields, PSLs trace the principal stress directions across a shell surface and are often visualized using color to convey stress magnitude. They reveal the tensile and compressive directions that shape the shell's underlying load-bearing behavior and show how these directions adapt to geometric curvature, boundary conditions, and applied loads. Patterns in the compressive PSLs can help identify mechanically sensitive regions: areas where these trajectories converge or exhibit pronounced curvature often coincide with high compressive stress and may correlate with increased susceptibility to buckling

* Corresponding authors.
E-mail addresses: junpeng.wang@tum.de (J. Wang), yingjian.liu@uga.edu (Y. Liu), j.wu-1@tudelft.nl (J. Wu), westermann@tum.de (R. Westermann).

https://doi.org/10.1016/j.cma.2026.118770
Received 19 November 2025; Received in revised form 14 January 2026; Accepted 19 January 2026
Available online 23 January 2026
0045-7825/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

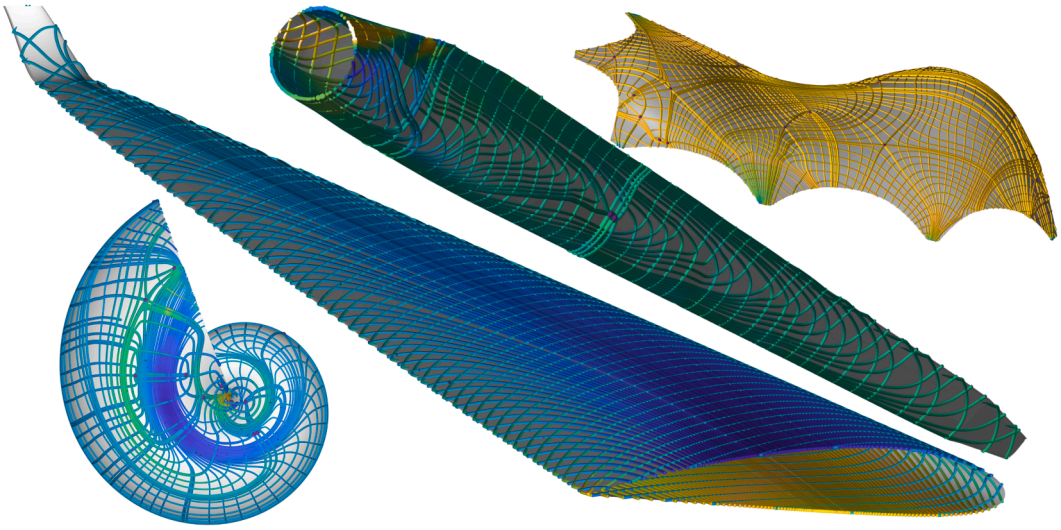

Fig. 1. The generated PSLs by the proposed framework, where the trajectories of the major and minor PSLs are separately color-coded with major and minor principal stresses along the corresponding trajectories. From yellow to blue, the corresponding principal stress values vary from the maximum to minimum. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

phenomena [6,7]. Tensile PSLs that vary smoothly across the surface indicate regions where the direction of tensile stress changes gradually, which can be a qualitative sign of efficient tension distribution and overall structural efficiency. At a broader structural scale, localized variations in the direction or curvature of PSLs may highlight geometric features or stress gradients that influence how loads are redistributed toward boundaries and supports [8,9]. In this sense, PSLs offer a physically grounded visualization of load flow, providing interpretable insight into how shells transmit and redistribute stresses [4].

Beyond the analytical values, PSLs also offer a valuable perspective for lightweight structural design, where stress-aligned material layouts are essential for achieving structural efficiency. This relevance traces back to Michell's Theory [10], which establishes that a stiffness-optimal structure avoids shear stresses and aligns material distribution with principal stress directions, a principle echoed in modern topology-optimization frameworks [11–13]. At the same time, stringent design requirements, such as tolerance against local damage, robustness to variations in loading direction, and resilience against buckling, often motivate structural layouts with increasingly fine-grained details. Capturing these features within a conventional topology optimization pipeline requires simulations at very high resolution, which quickly becomes computationally prohibitive [14–17]. This motivates interest in lower-cost design strategies that require only a single stress analysis rather than repeated high-fidelity optimization cycles.

Within this context, PSLs can be exploited as a heuristic material-alignment strategy when strict optimality is not required [18–20], as a diagnostic tool for interpreting topology optimization outcomes [21], or as guiding fields within hybrid methods that combine trajectory-based material alignment with numerical optimization [22,23]. These approaches assume that the stress field of the initial structure remains sufficiently close to that of the PSL-guided design, despite the inherent mismatch introduced when material layout is updated. Several infill-design studies have shown this approximation within their design settings [12,22,23]. While stress-informed methods have been established for 2D and 3D solids, applying them to shell structures is less straightforward and requires both theoretical and algorithmic adaptations. For example, recent work [24] uses PSLs to design rib reinforcements for thin shells, improving stiffness and regularity. However, that approach requires significant regularization of nodal stress directions in quadrilateral elements to interface with generic streamline integration tools (i.e., Matplotlib's streamplot). In contrast, our tracing algorithm is tailored to the inherent features of stress fields, eliminating the need for such regularization.

Achieving high-fidelity PSLs is particularly critical for shell structures, since their curved geometries and thin-walled nature make design decisions highly sensitive to the accuracy of stress trajectory representation. However, tracing PSLs on shell structures is fundamentally more complex than in 2D or 3D solids, where the principal directions are defined in the global Cartesian frame and trajectories can be integrated directly. By contrast, the principal stresses are usually interpreted on the tangent plane of the shell elements, i.e., an element-wise local frame, while the PSLs still need to be displayed in the global Cartesian frame. Ensuring consistency between these representations is nontrivial, particularly as trajectories cross elements with varying orientations. The challenge is further compounded by geometric curvature, which requires PSLs to remain faithful to the surface embedding and to navigate features such as creases. Higher-order shell elements, often introduced to capture curved geometry more accurately, increase the difficulty even further, since nonlinear interpolation complicates both direction evaluation and trajectory integration. Singularities, characterized by stress degeneracies, also demand careful treatment for a faithful PSL distribution. In applications, incomplete PSL distribution risk missing dominant load paths and separatrices, overlooking localized stress concentrations that are critical for reinforcement placement. Comprehensive coverage of PSLs across the shell is therefore essential [4,5,19,25]. In current practice, however, such coverage is often pursued by first generating a dense set of PSLs and then clustering to extract members,

which inevitably amplifies noise, obscures behavior near degeneracies, and relies on heuristic post-selection. This is partly because the quality of individual trajectories cannot be guaranteed, and the topological separatrices of the principal stress field are overlooked.

This paper is motivated by the above and aims to generate curvature-faithful and analysis-consistent stress trajectories on shell structures. Due to the limitations of pre-smoothed direction fields, we adopt a surface-intrinsic, FEM-aligned strategy for PSLs generation. Conceptually, the tracing algorithm does not predefine a direction field. Instead, it evaluates the stress state in which the curve currently resides and determines the principal direction on the fly, so PSL trajectories respond to the actual mechanics rather than to a preprocessed proxy. This design keeps the method agnostic to element type, allowing it to operate seamlessly on a wide range of element types, as well as their hybrids. The curvature is preserved by querying geometry and stress through the elements' native interpolation. To address ambiguity where principal values coalesce and to distinguish genuine mechanical features from potential tensor-field artifacts, we introduce the stress topology analysis to preserve structural coherence in the resulting PSL distribution. Instead of generating a dense PSL distribution first and then conducting clustering operations to get a visually clean PSL distribution [5,25], we propose a surface-adapted spacing mechanism to produce a domain-filling and evenly spaced PSL distribution suitable for analysis and design. Fig. 1 showcases the generated results in various applications.

Methodologically, PSL generation can be regarded as part of stress tensor visualization. In general, stress tensor visualization techniques can be grouped into three complementary categories based on the chosen visual abstractions: trajectory-based, glyph-based, and topology-based [26]. Trajectory-based approaches integrate principal stress direction fields to produce curves such as PSLs, thereby revealing stress flow in an intuitive manner [27]. Seeding strategies are often incorporated into this process to control line density and coverage [28,29]. Texture-based extensions such as HyperLIC also fall within this category [30]. These methods are valued for their ability to convey directionality, continuity, and convergence or divergence of stress fields. Glyph-based approaches, in contrast, encode tensor magnitudes and orientations locally (e.g., ellipsoids or superquadrics), allowing dense and quantitative inspection without integration, though at the expense of global continuity and with the risk of visual clutter in complex datasets [31, 32]. Topology-based approaches analyze the structural properties of the tensor field by identifying and classifying degenerate points in 2D or degenerate curves in 3D and extracting the associated topological skeleton [33–35]. Recent advances have enriched this category with topological features such as mode faces, transition points, and core lines [36–38], providing a foundation for structurally consistent seeding and reliable organization of PSL patterns, though the physical interpretation of these features and the practical robustness of the corresponding extraction methods are still in their infancy. Comprehensive surveys have contextualized these categories in mechanics, highlighting their relevance for stress and strain analysis and summarizing best practices [39]. Specific to PSL generation on shell structures, the problem falls into the trajectory-based category. However, its practical realization requires a range of adaptations and extensions to planar or volumetric settings, as well as the integration of other categories.

For shell structures, trajectory-based PSL pipelines primarily aim to identify a set of trajectories that represent the principal stress directions and typically follow one of several implementation patterns. A prevalent choice is to precompute principal directions on a triangulated surface, often after down-converting higher-order shells to first-order triangles. This is followed by smoothing or orienting the field, and then integrating streamlines in the world space. This route is simple and fast, benefiting from mature tooling (e.g., Karamba3D [40]) and has recently been explored by [25,41]. However, this routine linearizes geometry, inherits noise from stress recovery, and requires heavy field smoothing or orientation. Especially when hybrid or high-order shells are present, the necessary down-conversion can degrade curvature fidelity and impair mesh refinement behavior. Besides, though the smoothing operations on the principal stress directions can lead to more regular PSL trajectories, the directional deviations between the generated PSL trajectories and the original principal stress directions can become large, especially when the field carries singularities [42]. A second technical routine flattens the surface via a conformal or quasi-conformal parameterization, transforms the tensor in 2D, conducts the tracing operations, and then maps trajectories back. While numerically clean in the plane and convenient for seeding and texture or LIC-style filtering, it introduces distortion and management of seam and atlas. Further, it is still challenging to maintain fidelity for curved and high-order geometry [43–45]. A third line integrates steps in ambient 3D with closest-point or level-set re-projection to the surface. This is straightforward and can be metric-aware if a signed-distance or closest-point representation is maintained. However, repeated projections incur tangency drift, thus, are sensitive to high curvature, and become intricate at element boundaries and creases [46–48]. Finally, some approaches construct optimized frame fields to enforce global consistency before integrating integral curves. These fields improve robustness to noise and orientation flips but depend on pre-processing and smoothing that may blur stress variation, and they do not remove ambiguities near degeneracies or guarantee domain-filling coverage [49–51]. Taken together, existing strategies trade speed and convenience for geometric linearization, relying on pre-smoothed direction fields, projection, and parameterization artifacts that incur particularly brittle behavior near creases on curved high-order elements and across mixed discretizations.

The remainder of the paper is organized as follows. Section 2 establishes the conceptual foundations and notation. Section 3 details the PSL tracing algorithm. Sections 4 and 5 present the stress-topology analysis and the topology-aware PSL seeding strategy, respectively. Numerical verifications and case studies are reported in Section 6. Section 7 concludes the paper and outlines directions for future work.

2. Shell elements

Before introducing our PSL tracing algorithm on shell structures composed of first- and second-order triangular and quadrilateral elements that are separately denoted as *T3*, *Q4*, *T6*, and *Q8*, we review the fundamental concepts underlying such representations. Unlike the finite elements of volumetric and planar structures that only involve the global Cartesian frame ($\mathcal{G}$) and the natural coordinate system ($\mathcal{N}$), the distinctive feature of shell elements is the presence of an additional local frame ($\mathcal{L}$) attached to the shell's

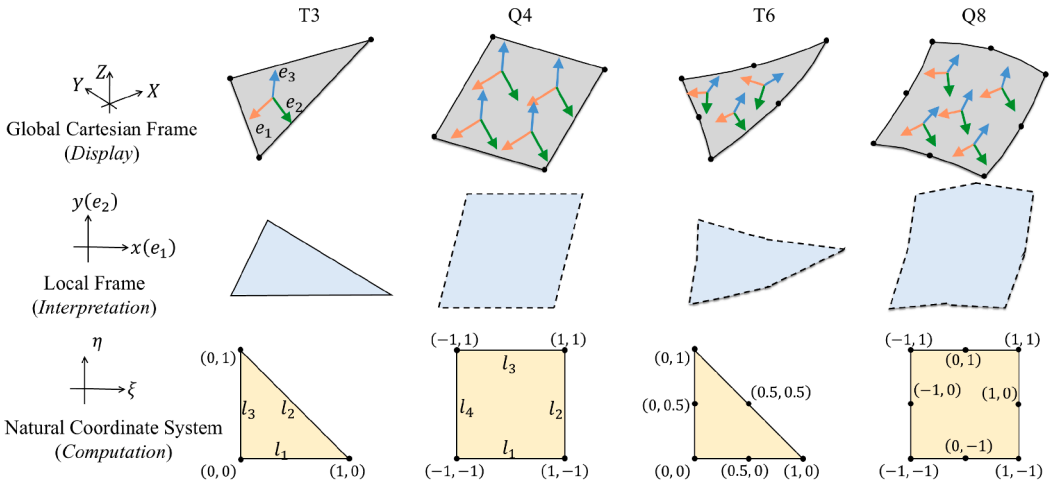

Fig. 2. Schematic diagrams of the shell elements. From top to bottom, each column sequentially shows a specific element type in the global Cartesian frame, local frame, and natural coordinate system. The edges of elements $Q4$, $T6$, and $Q8$ in the local frame are shown as dotted lines to indicate that such a projection can be a curved surface due to the spatially varying normals.

mid-surface. This local frame is introduced because the stress and strain measures for a thin shell are naturally defined in the tangent plane of the mid-surface, with the normal direction representing the shell's thickness axis. In other words, the constitutive law, the in-plane stress tensor, and the principal stress directions are most meaningfully expressed in a basis aligned with the shell surface rather than in the arbitrary global Cartesian frame.

To ease expression, we use (X, Y, Z), (x, y), and (ξ, η) to represent the coordinates in the global Cartesian frame, local frame, and natural coordinate system, respectively. I.e., $(X, Y, Z) \in \mathcal{G}$, $(x, y) \in \mathcal{L}$, and $(\xi, \eta) \in \mathcal{N}$. Fig. 2 shows the schematics of the shell elements in different frames.

2.1. Construction of the local frame

The local frame is an orthonormal basis attached to the shell's mid-surface at a given point, aligned with its tangent plane and normal direction. It is constructed so that in-plane quantities, such as principal stress directions, can be expressed in a more physically meaningful reference system before being mapped to or from the global Cartesian frame.

For a shell element (e) in 3D, any position $(\boldsymbol{P})$ on its mid-surface is given by the isoparametric mapping, or so-called element interpolation:

$$\boldsymbol{P}(\xi, \eta) = \sum_{i=1}^{M_e} N_i(\xi, \eta) \boldsymbol{P}^{[i]} \tag{1}$$

Here, M_e is the number of element nodes, (ξ, η) are the natural coordinates corresponding to $\boldsymbol{P}$, $\boldsymbol{P}^{[i]}$ are the node coordinates in the global frame, and N_i are the shape functions.

The tangent plane at $\boldsymbol{P}(\xi, \eta)$ is spanned by the two tangent vectors

$$\boldsymbol{t}_1 = \frac{\partial \boldsymbol{P}(\xi, \eta)}{\partial \xi}; \quad \boldsymbol{t}_2 = \frac{\partial \boldsymbol{P}(\xi, \eta)}{\partial \eta} \tag{2}$$

From these, the unit normal vector $\boldsymbol{n}$ at $P(\xi, \eta)$ is

$$\boldsymbol{n} = \frac{\boldsymbol{t}_1 \times \boldsymbol{t}_2}{\|\boldsymbol{t}_1 \times \boldsymbol{t}_2\|} \tag{3}$$

To resolve the in-plane rotation ambiguity, a reference vector $\boldsymbol{r}$ is projected onto the tangent plane to define the first local axis (i.e., x-axis in Fig. 2):

$$\boldsymbol{e}_1 = \frac{\boldsymbol{r} - (\boldsymbol{r} \cdot \boldsymbol{n})\boldsymbol{n}}{\|\boldsymbol{r} - (\boldsymbol{r} \cdot \boldsymbol{n})\boldsymbol{n}\|} \tag{4}$$

The second local axis (y-axis in Fig. 2) is obtained by the right-handed rule:

$$\boldsymbol{e}_2 = \boldsymbol{n} \times \boldsymbol{e}_1 \tag{5}$$

The local orthonormal frame $(\boldsymbol{R})$ at $P(\xi, \eta)$ is then

$$\boldsymbol{R} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3], \ \boldsymbol{e}_3 = \boldsymbol{n} \tag{6}$$

Where $\boldsymbol{e}_i$ $(i = 1, 2, 3)$ are the column vectors of $\boldsymbol{R}$.

Given that the first local axis in $\mathcal{L}$ aligns with $\boldsymbol{e}_1$ and, thus, is determined by the reference vector $\boldsymbol{r}$, the setting of $\boldsymbol{r}$ directly relates to the interpretation of the principal stress directions in $\mathcal{L}$. This paper primarily utilizes ABAQUS to generate stress datasets, which are node-wise stress fields represented in the local frame. By default, ABAQUS initializes $\boldsymbol{r}$ for all elements with $[1, 0, 0]$, but in case $\boldsymbol{n}$ is parallel to the X-axis in $\mathcal{G}$, $[0, 0, 1]$ is used as a fallback [52].

In the first-order triangular element ($T3$), the shape functions are affine in $\mathcal{L}$, so the tangent vectors $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ remain constant throughout the element. For the quadrilateral element ($Q4$), this constancy holds only when the element is a perfect parallelogram without bilinear distortion. In the general case, although $Q4$ is still classified as a first-order element, $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ can vary across the element. In higher-order elements ($T6$, $Q8$), the presence of mid-side nodes introduces curvature to the mid-surface, causing $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ (and consequently the normal $\boldsymbol{n}$ and local frame $\boldsymbol{R}$) to depend explicitly on (ξ, η). Therefore, the local frame must be evaluated at each point of interest (e.g., Gaussian integration points), and projections to or from the local coordinate system are performed point-wise.

2.2. Transformation between global and local frames

To make this paper self-contained and avoid ambiguity in later derivations, we define here the transformations between the global Cartesian frame $\mathcal{G}$ and the local frame $\mathcal{L}$ attached to each shell element. These transformations will be used repeatedly in the following sections for both position and direction vectors.

With the local frame $\boldsymbol{R}$ defined in Eq. (6), and the selected origin $\boldsymbol{O}$ in $\mathcal{G}$, which is usually set as the mass center of an element, the global-to-local transformations for positions and directions are achieved as

$$\begin{bmatrix} \boldsymbol{p} \\ 0 \end{bmatrix} = \boldsymbol{R}^{\mathrm{T}}(\boldsymbol{P} - \boldsymbol{O}), \tag{7}$$

$$\begin{bmatrix} \boldsymbol{v} \\ 0 \end{bmatrix} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{V}. \tag{8}$$

Here, $\boldsymbol{P}$ and $\boldsymbol{V}$ respectively denote the position and vector in $\mathcal{G}$, and $\boldsymbol{p}$ and $\boldsymbol{v}$ are the corresponding projections of $\boldsymbol{P}$ and $\boldsymbol{V}$ in $\mathcal{L}$, their 3rd entries are 0 in theory and shall be excluded in the use. By leveraging $\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} = \boldsymbol{I}$ ($\boldsymbol{I}$ is the identity matrix), the corresponding local-to-global transformations can be obtained similarly.

Considering that $\boldsymbol{R}$ is constant across $T3$, one can easily get the planar projection of such a shell element in $\mathcal{L}$ by applying Eq. (7) to the element vertices. For $Q4$, $T6$, and $Q8$ such a projection does not exist exactly, since $\boldsymbol{R}$ is not necessarily consistent across the element, refer to the middle row of Fig. 2.

2.3. Stress representation in shell elements

In finite elements of shells, the stress tensor ($\boldsymbol{T}_{\mathcal{L}}$) is directly computed in the local frame ($\mathcal{L}$) to reveal the in-plane properties, and the corresponding stress tensor ($\boldsymbol{T}_{\mathcal{G}}$) in $\mathcal{G}$ of $\boldsymbol{T}_{\mathcal{L}}$ can be obtained by the projection

$$\underbrace{\begin{bmatrix} \sigma_{XX} & \sigma_{XY} & \sigma_{XZ} \\ \sigma_{XY} & \sigma_{YY} & \sigma_{YZ} \\ \sigma_{XZ} & \sigma_{YZ} & \sigma_{ZZ} \end{bmatrix}}_{\boldsymbol{T}_{\mathcal{G}}} = \boldsymbol{R}\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_{yy} & 0 \\ 0 & 0 & 0 \end{bmatrix}\boldsymbol{R}^{\mathrm{T}}, \tag{9}$$

where, $\begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$ forms $\boldsymbol{T}_{\mathcal{L}}$.

Commercial FEM software typically provides these quantities after analysis in formats tailored to shell theory. For example, membrane stresses and bending stresses are often reported separately, either as stress resultants or as surface stresses evaluated at the top and bottom shell surfaces. Depending on the solver and element formulation (e.g., Mindlin-Reissner or Kirchhoff-Love theory), transverse shear stresses may also be available. These outputs already assume a local element frame, even if the results are stored in global coordinates for post-processing. Note that, while different shell formulations influence the stress quantities delivered by an FEM solver, the present work treats these stresses as given input and assumes a symmetric in-plane stress tensor field defined on the shell surface and expressed in the local shell frame. The proposed PSL tracing and stress topology analysis are therefore independent of the specific shell kinematic model used to generate the stress results.

This distinction is critical for PSL tracing, i.e., numerical integration along the principal stress directions, although different types of elements follow the same paradigm: the final result is displayed in $\mathcal{G}$, and the interpolation computation is conducted in $\mathcal{N}$. For a shell element, the principal directions are interpreted in $\mathcal{L}$ instead of $\mathcal{G}$ like in planar or volumetric elements. For intuitiveness, one can also consider $\mathcal{G}$ the *Display* frame, $\mathcal{L}$ the *Interpretation* frame, and $\mathcal{N}$ the *Computation* frame. By distinguishing the roles of $\mathcal{G}$, $\mathcal{L}$, and $\mathcal{N}$, PSL tracing in shell elements can be formulated in a way that is both mathematically consistent and directly compatible with the stress outputs of commercial FEM codes.

It is worth emphasizing that the stress interpolation within an element needs to be performed in a consistent frame, e.g., in $\mathcal{G}$ for convenience. Thus, if the stress field is given in $\mathcal{L}$, we first need to convert the corresponding stress tensors at nodes to $\mathcal{G}$ for element

interpolation, then project the interpolated stress tensor in $\mathcal{G}$ back to $\mathcal{L}$ for principal stress interpretation.

$$\boldsymbol{T}_{\mathcal{L}}(\xi,\eta)=\boldsymbol{R}^{\mathrm{T}}(\xi,\eta)\left(\sum_{i=1}^{M_e}N_i(\xi,\eta)\underbrace{\left(\boldsymbol{R}_{[i]}\boldsymbol{T}_{\mathcal{L}}^{[i]}\boldsymbol{R}_{[i]}^{\mathrm{T}}\right)}_{\boldsymbol{T}_{\mathcal{G}}^{[i]}}\right)\boldsymbol{R}(\xi,\eta) \tag{10}$$

Here, $\boldsymbol{R}_{[i]}$ denotes the local orthonormal frame at node-i of the considered element, $\boldsymbol{T}_{\mathcal{G}}^{[i]}$ refers to the stress tensors at node-i in $\mathcal{G}$. For $T3$, since $\boldsymbol{R}_{[i]},\ i=1:M_e$ are constant over the element, this formula can be simplified to directly conduct interpolation in $\mathcal{L}$. Since $\boldsymbol{R}_{[i]}$ can vary spatially over $Q4$, $T6$ and $Q8$, it is required to precisely follow Eq. (10) for higher precision. Note that the formulation above relies solely on the isoparametric description of geometry and the interpolation of nodal stress values via shape functions, whereas PSL tracing evaluates the stress field at arbitrary parametric locations and does not depend on element-specific integration schemes or Gauss points.

3. PSL tracing

Principal stress lines are trajectories that start from a seed point and follow the principal stress directions. These are the directions where the shear stress components vanish, and are computed as the eigenvectors of the symmetric second-order stress tensor ($\boldsymbol{T}_{\mathcal{L}}$), see, for instance, Wang et al. [29] for a more detailed discussion. The tracing algorithm propagates element-by-element along the selected stress direction. For efficiency, elements are stored in a topological data structure that records element types, adjacency relationships, and edge types (interior vs. boundary). When a PSL reaches an edge, this structure allows immediate retrieval of the adjacent element or termination of the trace if the edge indicates a boundary or hole.

In the following, we begin with the first-order Runge-Kutta method (Euler method) as the numerical integrator for computing principal stress lines, as it offers the simplest formulation for illustrating the core idea. Later, we employ the second-order Runge-Kutta method (RK2) to achieve higher accuracy in the actual tracing. The Euler update for PSL tracing can be written as

$$\boldsymbol{s}_{n+1}=\boldsymbol{s}_n+\delta\boldsymbol{d}_n \tag{11}$$

Here, δ is the prescribed step size for integration, and $\boldsymbol{d}_n$ is the selected principal stress direction at position $\boldsymbol{s}_n$, which is a unit directional vector. $\boldsymbol{s}_{n+1}$ is the newly updated point.

3.1. Point advancing strategy

In problems where the global Cartesian frame and the local frame coincide, Eq. (11) can be used directly to advance points along principal stress directions in $\mathcal{G}$. In shell elements, however, the stress tensor is defined in the local element frame $\mathcal{L}$, which typically differs from $\mathcal{G}$ and requires explicit consideration during point advancement.

There are two seemingly straightforward ways to tackle this situation, yet both face problems in practice. One approach is to project the stress tensor field from $\mathcal{L}$ to $\mathcal{G}$, and then conduct PSL tracing in the transformed field. However, this global representation mixes in-plane and out-of-plane components, obscuring the in-surface stress directions and requiring repeated projection operations. Furthermore, it becomes cumbersome to precisely locate $\boldsymbol{s}_{n+1}$ in $\mathcal{N}$ for interpolation purposes. The alternative approach is to directly conduct PSL tracing in $\mathcal{L}$ and then project the updated positions to $\mathcal{G}$. This is viable for $T3$ since its planar projection in $\mathcal{L}$ is always available. From this, one can easily detect whether $\boldsymbol{s}_{n+1}$ is within the current element or not. However, this approach becomes infeasible in $Q4$, $T6$, and $Q8$, since they do not necessarily obey a consistent projection in $\mathcal{L}$. Besides, this approach involves locating $\boldsymbol{s}_{n+1}$ in $\mathcal{N}$, which, except for $T3$, makes it similarly cumbersome as the first way, i.e., repeatedly solving a small nonlinear system using the Newton-Raphson method.

To overcome these limitations, we propose a new tracing algorithm. In particular, we solve Eq. (11) in $\mathcal{N}$, i.e., $\boldsymbol{s}_n$ and $\boldsymbol{s}_{n+1}$ are the natural coordinates (ξ_n,η_n) and (ξ_{n+1},η_{n+1}). To avoid that $\boldsymbol{d}_n$ is not defined in $\mathcal{N}$, we arrange the term $\delta\boldsymbol{d}_n$ as a mapping of the physically meaningful update in $\mathcal{L}$. This facilitates locating the updated integration point in the mesh by comparing $\boldsymbol{s}_{n+1}$ with the standard isoparametric element in $\mathcal{N}$, and also circumvents repeatedly calling the Newton-Raphson method since $\boldsymbol{s}_{n+1}$ is already given in $\mathcal{N}$. I.e., one only needs to conduct an element interpolation to get the physical coordinate of $\boldsymbol{s}_{n+1}$ in $\mathcal{G}$ to form the PSL.

Clearly, the key idea of this method is to establish a mapping relation between $\mathcal{L}$ and $\mathcal{N}$, ensuring that the position update in $\mathcal{L}$ can be precisely correlated with the update in $\mathcal{N}$. This is laid on the basis that either an increment in $\mathcal{L}$ $(\Delta x,\Delta y)$ or $\mathcal{N}$ $(\Delta\xi,\Delta\eta)$ corresponds to an increment in $\mathcal{G}$ $(\Delta X,\Delta Y,\Delta Z)$. Given the increment is essentially a vector in the corresponding frames, we can get $(\Delta X,\Delta Y,\Delta Z)$ from $(\Delta x,\Delta y)$ by Eq. (8), i.e.,

$$\begin{bmatrix}\Delta X\\ \Delta Y\\ \Delta Z\end{bmatrix}=\boldsymbol{R}\begin{bmatrix}\Delta x\\ \Delta y\\ 0\end{bmatrix} \tag{12}$$

Analogously, an increment in the natural coordinates can also be mapped to the global space through the surface Jacobian

$$\begin{bmatrix}\Delta X\\ \Delta Y\\ \Delta Z\end{bmatrix}=\begin{bmatrix}\frac{\partial\boldsymbol{P}(\xi,\eta)}{\partial\xi} & \frac{\partial\boldsymbol{P}(\xi,\eta)}{\partial\eta}\end{bmatrix}_{[3\times2]}\begin{bmatrix}\Delta\xi\\ \Delta\eta\end{bmatrix}=\begin{bmatrix}\boldsymbol{t}_1 & \boldsymbol{t}_2\end{bmatrix}\begin{bmatrix}\Delta\xi\\ \Delta\eta\end{bmatrix} \tag{13}$$

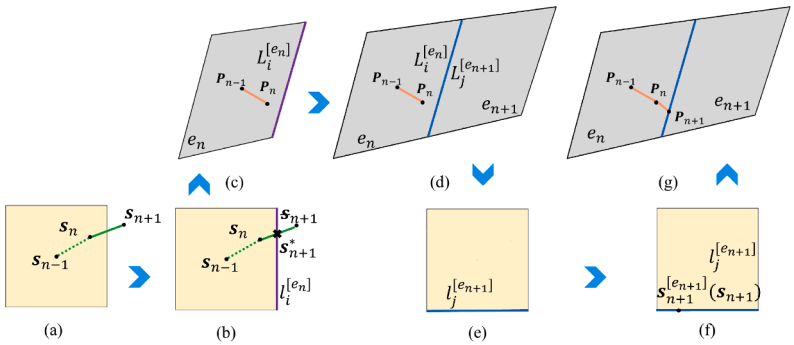

Fig. 3. Schematic diagram of advancing PSL when the directly computed new point ($s_{n=1}$) is outside of the current element. Grey patches show the elements in $\mathcal{G}$, and light yellow ones are the corresponding isoparametric elements in $\mathcal{N}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Let Eq. (12) be equal to Eq. (13) and left-multiply by $\boldsymbol{R}^{\mathrm{T}}$, we obtain

$$\begin{bmatrix}\Delta x\\ \Delta y\\ 0\end{bmatrix} = \boldsymbol{R}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{t}_1 & \boldsymbol{t}_2\end{bmatrix}\begin{bmatrix}\Delta\xi\\ \Delta\eta\end{bmatrix} \tag{14}$$

Substituting Eq. (6) into Eq. (14) and omitting the zero items corresponding to the third dimension, we can establish the correlation between $(\Delta x, \Delta y)$ and $(\Delta\xi, \Delta\eta)$ as

$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{e}_1\cdot\boldsymbol{t}_1 & \boldsymbol{e}_1\cdot\boldsymbol{t}_2\\ \boldsymbol{e}_2\cdot\boldsymbol{t}_1 & \boldsymbol{e}_2\cdot\boldsymbol{t}_2\end{bmatrix}}_{\boldsymbol{J}}\begin{bmatrix}\Delta\xi\\ \Delta\eta\end{bmatrix} \tag{15}$$

Here, $\boldsymbol{J}$ represents the Jacobian matrix that transforms parametric increments in $\mathcal{N}$ into $\mathcal{L}$. In our context, the increment in the local frame is directly updated by the principal stress direction $\boldsymbol{d}_n$, i.e., $(\Delta x, \Delta y) = \delta\boldsymbol{d}_n$. Thereby, the corresponding increment $\Delta\boldsymbol{s} = (\Delta\xi, \Delta\eta)$ in $\mathcal{N}$ is given by $\delta\boldsymbol{J}_n^{-1}\boldsymbol{d}_n$, and thus,

$$\boldsymbol{s}_{n+1} = \boldsymbol{s}_n + \underbrace{\delta\boldsymbol{J}_n^{-1}\boldsymbol{d}_n}_{\Delta\boldsymbol{s}} \tag{16}$$

Thereby, the point is advanced in $\mathcal{N}$ under the guidance of the actual principal stress directions in $\mathcal{L}$.

3.2. Point update

Eq. (16) conceptually paves the way for advancing a point along a PSL, yet some aspects concerning the implementation of PSL tracing remain critical. In the case where $\boldsymbol{s}_n$ and $\boldsymbol{s}_{n+1}$ are within the same element e_n, we can determine the position $\boldsymbol{P}_{n+1}$ of $\boldsymbol{s}_{n+1}$ in $\mathcal{G}$, and thus the PSL direction, by element interpolation (Eq. (1)). The stress tensor ($\boldsymbol{T}_{\mathcal{L}}^{[n+1]}$) corresponding to $\boldsymbol{s}_{n+1}$ can be computed accordingly using Eq. (10). For the selected type of principal direction ($\boldsymbol{v}_1^{[n+1]}$), either major or minor, it needs to be determined into which direction to proceed, either $+\boldsymbol{v}_1^{[n+1]}$ or $-\boldsymbol{v}_1^{[n+1]}$. To obtain a consistent continuation of a PSL, we introduce a direction regularization process as follows

$$\boldsymbol{d}_{n+1} = \measuredangle(\boldsymbol{D}_n, \boldsymbol{V}_1^{[n+1]}) < \measuredangle(\boldsymbol{D}_n, -\boldsymbol{V}_1^{[n+1]})\ ?\ \boldsymbol{v}_1^{[n+1]}\ :\ -\boldsymbol{v}_1^{[n+1]} \tag{17}$$

where, $\boldsymbol{D}_n$ and $\boldsymbol{V}_1^{[n+1]}$ are the projections of $\boldsymbol{d}_n$ and $\boldsymbol{d}_1^{[n+1]}$ through Eq. (8). With the updated $\boldsymbol{d}_{n+1}$ and $\boldsymbol{s}_{n+1}$ one can proceed to the next position $\boldsymbol{s}_{n+2}$. Note that the direction comparison is performed in $\mathcal{G}$ for directional consistency.

When $\boldsymbol{s}_{n+1}$ falls outside e_n (Fig. 3a), we must determine the next element e_{n+1} that actually contains this point and then perform element interpolation there to obtain $\boldsymbol{P}_{n+1}$ and $\boldsymbol{T}_{\mathcal{L}}^{[n+1]}$. This transition is non-trivial because $\boldsymbol{s}_{n+1}$ is expressed in $\mathcal{N}$ of e_n, but the interpolation on e_{n+1} requires $\boldsymbol{s}_{n+1}$ to be expressed in $\mathcal{N}$ of e_{n+1}. However, $\mathcal{N}$ carries no information about mesh connectivity, and there is no direct mapping between the natural coordinate systems of two adjacent elements. Moreover, if the surface normal of e_n and e_{n+1} differ substantially, as is typical when the two elements lie on different faces meeting at a crease, naively using the traced point $\boldsymbol{s}_{n+1}$ in e_{n+1} as the updated PSL point in e_{n+1} leads to geometric inconsistencies.

To handle this situation, we perform element interpolation in four steps: Firstly, we compute the intersection ($\boldsymbol{s}_{n+1}^*$) of the line segment $\overline{\boldsymbol{s}_n\boldsymbol{s}_{n+1}}$ and the vertex edge (l_i) of the corresponding isoparametric element of e_n (Fig. 3b) (the ordering of l_i is according to Fig. 2). Secondly, from $l_i^{[e_n]}$ we relate to the corresponding vertex edge ($L_i^{[e_n]}$) of e_n in $\mathcal{G}$ (Fig. 3c). $L_i^{[e_n]}$ must be shared by a different element e_{n+1} if it's not the boundary edge for a 2-manifold shell mesh, saying $L_j^{[e_{n+1}]}$ (Fig. 3d). Thirdly, we relate $L_j^{[e_{n+1}]}$ to the corresponding edge $l_j^{e_{n+1}}$ in the isoparametric element of e_{n+1} (Fig. 3e). Fourthly, according to the positions of $l_i^{[e_n]}$ and $l_j^{[e_{n+1}]}$ in their corresponding isoparametric elements, the intersection ($\boldsymbol{s}_{n+1}^*$) in e_n can be equivalently transferred to $\boldsymbol{s}_{n+1}^{[e_{n+1}]}$, which is taken as

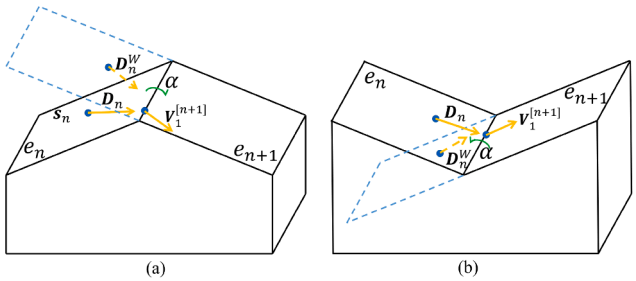

Fig. 4. Schematic diagrams of how Rodrigues rotation works for a convex ridge (a) and a concave valley (b).

the updated position $\boldsymbol{s}_{n+1}$ but in element e_{n+1} (Fig. 3f). In this way, the tracing process can be advanced smoothly, i.e., $\boldsymbol{P}_{n+1}$ and $\boldsymbol{T}_{\mathcal{L}}^{[n+1]}$ can be correctly computed by conducting element interpolation with $\boldsymbol{s}_{n+1}$ in e_{n+1}, see Fig. 3g. If $L_i^{e_n}$ is only shared by e_n, which indicates that the tracing process is approaching the boundary, then $\boldsymbol{s}_{n+1}^*$ in e_n is considered as the updated position $\boldsymbol{s}_{n+1}$ to compute the end point of a PSL and terminate the tracing process. The efficient computation of edge indices when transitioning from e_n to e_{n+1} is supported by a pre-computed element-connectivity map. Since e_n and e_{n+1} always share a common edge by nature, i.e., they are adjacent to each other, PSL tracing proceeds strictly element-by-element and cannot skip intermediate elements.

It is worth noting that Eq. (17) needs to be adapted carefully when advancing to a different element and encountering the scenario where e_n and e_{n+1} form a crease, which can either be a convex ridge or a concave valley, as shown in Fig. 4. To counteract this issue, we resort to the Rodrigues rotation. Specifically, element e_n, including $\boldsymbol{D}_n$ on it, is rotated by a certain angle α around the edge tangent ($\boldsymbol{h}$) that is shared by e_{n+1} to make these two elements coplanar. Thus, $\boldsymbol{D}_n^W$, i.e., the rotated $\boldsymbol{D}_n$, becomes coplanar with $\boldsymbol{V}_1^{[n+1]}$, which is given by

$$\boldsymbol{D}_n^W = \boldsymbol{W}(\boldsymbol{h}, \alpha)\boldsymbol{D}_n \tag{18}$$

with

$$\boldsymbol{W}(\boldsymbol{h}, \alpha) = \boldsymbol{I} + \sin\alpha\boldsymbol{H} + (1 - \cos\alpha)\boldsymbol{H}^2 \tag{19}$$

Here, $\boldsymbol{W}(\boldsymbol{h}, \alpha)$ is the Rodrigues matrix. $\boldsymbol{I}$ is the identity matrix. $\boldsymbol{H}$ is the 3×3 skew-symmetric cross-product matrix of $\boldsymbol{h} = (h_x, h_y, h_z)^{\mathrm{T}}$ and defined by

$$\boldsymbol{H} = \begin{bmatrix} 0 & -h_z & h_y \\ h_z & 0 & -h_x \\ -h_y & h_x & 0 \end{bmatrix} \tag{20}$$

α is the signed dihedral angle that ranges in $[-\pi, \pi]$ and given by

$$\begin{aligned} &\alpha = \operatorname{atan2}\left(\boldsymbol{h} \cdot (\boldsymbol{n}_n \times \boldsymbol{n}_{n+1}), \boldsymbol{n}_n \cdot \boldsymbol{n}_{n+1}\right) \\ &\alpha > 0 \Rightarrow \text{convex ridge} \\ &\alpha < 0 \Rightarrow \text{concave valley} \\ &\alpha \approx 0 \Rightarrow \text{nearly coplanar} \\ &\alpha \approx \pi \Rightarrow \text{flipped} \end{aligned} \tag{21}$$

Where $\boldsymbol{n}_n$ and $\boldsymbol{n}_{n+1}$ are the outward normals of e_n and e_{n+1}. For $Q4$, $T6$, and $Q8$ elements that do not necessarily have a consistent normal vector, the normal at position $\boldsymbol{s}_n$ and $\boldsymbol{s}_{n+1}$ are taken. Replacing $\boldsymbol{D}_n$ with $\boldsymbol{D}_n^W$, we can decide the tracing direction in e_{n+1} by adapting Eq. (17) as following

$$\boldsymbol{d}_{n+1} = \measuredangle(\boldsymbol{D}_n^W, \boldsymbol{V}_1^{[n+1]}) < \measuredangle(\boldsymbol{D}_n^W, -\boldsymbol{V}_1^{[n+1]}) \; ? \; \boldsymbol{v}_1^{[n+1]} \; : \; -\boldsymbol{v}_1^{[n+1]} \tag{22}$$

Clearly, $\boldsymbol{D}_n^W$ becomes $\boldsymbol{D}_n$ when e_n and e_{n+1} are coplanar. And note that in implementation, the direction of the edge tangent must respect the orientation of the edge in e_n, i.e., it is consistently used in determining $\boldsymbol{n}_n$.

3.3. PSL creation

Fig. 5 summarizes the PSL tracing process from the given position s_0 in e_0 and the principal stress direction $\boldsymbol{d}_0$. Note that for computing the complete PSL going through this position, also $-\boldsymbol{d}_0$ needs to be considered in a separate run.

Given that the point update is essentially driven by an increment in $\mathcal{L}$, rather than by absolute positions in $\mathcal{G}$, thus, the algorithm does not rely on a consistent projection of the physical element onto $\mathcal{L}$. This makes the algorithm applicable to $T3$, $Q4$, $T6$, and $Q8$, as well as for hybrid meshes that include both triangular and quadrilateral elements.

For $T3$ elements, the local frame $\boldsymbol{R}$ and the tangent plane spanned by $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ remain constant within each element. In contrast, for $Q4$, $T6$, and $Q8$ elements, $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ must be evaluated at each tracing point $\boldsymbol{s}_n$. Within a single element, we assume that $\boldsymbol{s}_n$ and $\boldsymbol{s}_{n+1}$ lie in the same local tangent plane, i.e., the tangent plane (and therefore the frame $\boldsymbol{R}$) is effectively constant over one integration step (δ).

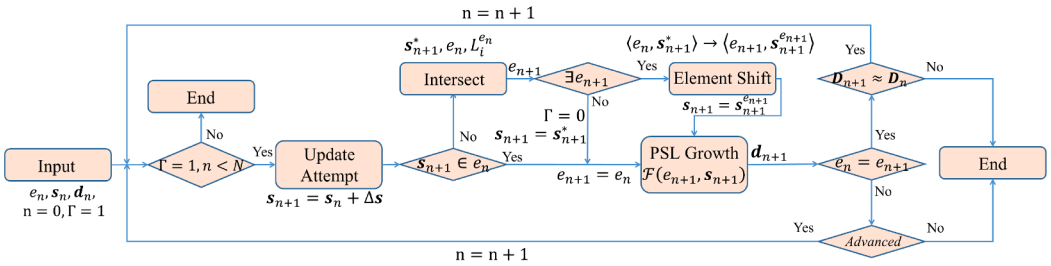

Fig. 5. Flow chart of tracing a PSL. The symbol $\exists e_{n+1}$ refers to checking whether element e_{n+1} exists. $\mathcal{F}(e_{n+1}, s_{n+1})$ is for computing the information of a PSL at the position s_{n+1} in the element e_{n+1}, including determining the tracing direction for updating the next point. $\Gamma = 0$ indicates the tracing process reaches the boundary edge. N is the maximum number of integration steps. $\approx$ is a generalized comparison operator to evaluate whether encountering excessive deviation in direction between two consecutive points. "*Advanced*" refers to a more sophisticated determining condition when stepping into a new element.

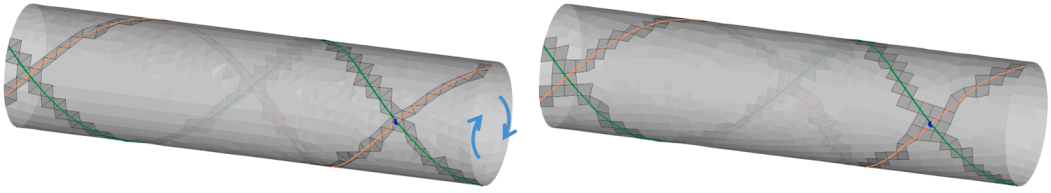

Fig. 6. Demonstration of PSLs generated on $T6$ (a) and $Q8$ (b) from the same starting position (blue $*$). Orange and green trajectories represent the major and minor PSLs, respectively. The left end of the cylinder is clamped, and an axial torque is applied to the right end (blue arrows). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

When working with high-curvature elements, this assumption may break down if δ is chosen too large. In that case, s_n and s_{n+1} may no longer be sufficiently co-planar, leading to inconsistencies between the PSL trajectory and the underlying curved geometry. To mitigate this issue, we propose an element-adaptive strategy to determine δ, i.e.,

$$\delta_{e_n} = \frac{C_{e_n}}{K^c_{e_n} K^u} \tag{23}$$

Here, C_{e_n} is the length of the shortest edge of element e_n, $K^c_{e_n}$ is a scaling factor reflecting the curvature of this element, which is selected to range in $[1,\ 5]$ from the flat element to the highly curved one. K^u is a user-defined scaling factor to control the granularity of the PSL, and also serves as a safety factor. We let $K^u = 5$ in this paper. With this strategy, the deviation between the tangent planes at s_n and s_{n+1} does not exceed 10^{-4} radian in all of the involved examples in this paper. Fig. 6 shows the generated PSLs by this method, where the cylinder object is separately discretized by the $T6$ and $Q8$ elements.

4. Stress topology analysis

Principal stress directions are the fundamental ingredient for PSL tracing, yet these directions are not always uniquely defined. At certain points, known as *Degenerate Points*, the two in-surface principal stresses become equal ($\sigma_1 = \sigma_2$) and the associated principal directions collapse. In the context of stress tensor analysis, such degenerate points correspond to the bifurcation or branching locations of PSLs, where principal directions are undefined. To correctly trace PSLs in the vicinity of such points, it is necessary to analyze the local topology of the direction field. This gives rise to the *topological skeleton* [33], a set of special PSLs that originate at degenerate points, with their initial directions determined by the local behavior of the tensor field rather than a single eigenvector. The topological skeleton is not only essential for continuing PSLs across singularities, but also serves as the natural separatrices of the stress field: within each subdomain bounded by separatrices, PSLs exhibit similar behavior and never cross the topological skeleton.

4.1. Degenerate point

A degenerate point does not necessarily coincide with mesh nodes, and because the FE stress field is only guaranteed to be continuous within individual elements, degenerate points must be identified on a per-element basis. Similarly to Section 3, the stress state in $\mathcal{L}$ is considered, i.e., the degenerate point is defined in $\mathcal{L}$, and element interpolation in Eq. (10) is used to represent the stress state within an element.

To locate the degenerate points, we introduce the deviatoric stress vector

$$\boldsymbol{g} = (f_1, f_2) = \left(\frac{1}{2}(\sigma_{xx} - \sigma_{yy}), \sigma_{xy}.\right) \tag{24}$$

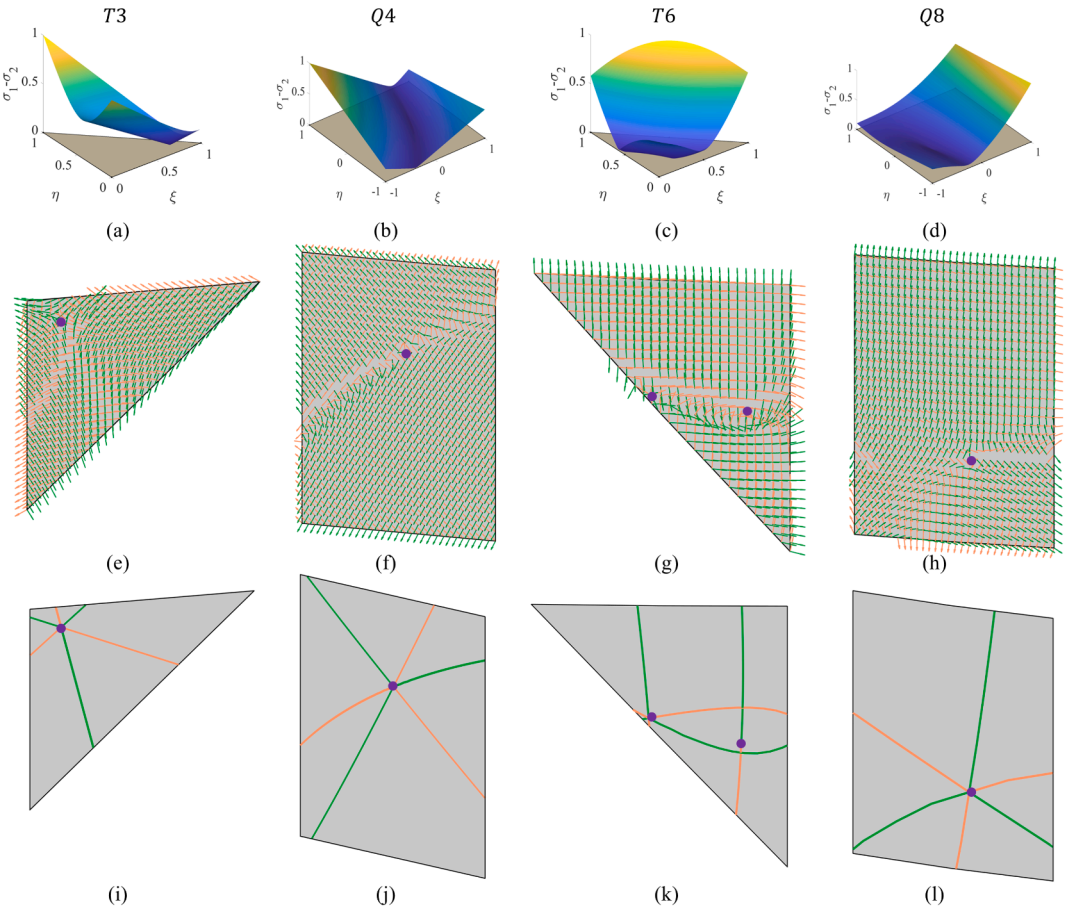

Fig. 7. Demonstration of principal stress behaviors within an element containing the degenerate point. (a)-(d) use curved color maps sequentially show the normalized $\sigma_1 - \sigma_2$ on $T3$, $Q4$, $T6$, and $Q8$ elements in $\mathcal{L}$. (e)-(f) show the corresponding principal stress directions in $\mathcal{G}$. (i)-(l) show the corresponding topological skeleton in $\mathcal{G}$. The selected $T6$ element contains 2 degenerate points, including a single wedge degenerate point. The purple points indicate the positions of the degenerate points in the elements. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The magnitude of $\boldsymbol{g}$ is the Mohr-circle radius, and its argument ($\tan(2\theta) = \frac{f_2}{f_1} = \frac{2\sigma_{xy}}{\sigma_{xx}-\sigma_{yy}}$) gives the double of the principal direction angle (θ). Degeneracy corresponds to zero radius. Fig. 7a–d show the possible distributions of $\sigma_1 - \sigma_2$ over the selected element that contains the degenerate point.

Clearly, at a degenerate point (x^*, y^*), $f_1(x^*, y^*) = f_2(x^*, y^*) = 0$. The algebraic structure of Eq. (24) is determined by the element shape functions N_i, $i = 1 : M_e$. For $T3$, N_i are affine, hence f_1 and f_2 are affine and their zero sets are straight lines. Consequently, Eq. (24) reduces to a 2×2 linear system, yielding either no intersection or a single intersection in the reference triangle. In contrast, for $Q4$, the fields are bilinear, and for $T6$ and $Q8$, they are quadratic on the reference domains, as shown in Fig. 2. Algebraically, two bilinear or two quadratic level sets may intersect in multiple isolated points, which is generically up to two for bilinear pairs and four for quadratic pairs, with the multiplicity being counted. Thus, in principle, an element of $Q4$, $T6$, and $Q8$ may contain more than one degenerate point, though this expectation is at odds with stress-analysis practice, where one typically observes at most one degenerate point per element. This is because the stress state produced by linear elasticity on regular shell meshes typically varies smoothly and only mildly across the domain. Therefore, (f_1, f_2) generally behaves as a gently curved map, even when higher-order interpolation or stress recovery is used. Nevertheless, the possibility of having multiple degenerate points within a single element cannot be simply excluded.

The number of potential degenerate points within an element directly decides the design of the root-searching strategy for Eq. (24). If it's known that there is at most one degenerate point in the targeted element, then one can apply a damped Newton-Raphson method to solve Eq. (24). Since the intended solution is unique, this process can converge rapidly. Otherwise, one may need to subdivide the element into smaller patches to isolate different potential degenerate points.

In principle, one can evaluate the number of potential degenerate points in an element by introducing a boundary-based winding-number (w) for the vector field (f_1, f_2). This is typically achieved by mapping the element boundary through (f_1, f_2) and tracking the resulting closed curve in the tailored plane $f_1 - f_2$. If this curve winds once around the origin $(0, 0)$ of this plane ($|w| = 1$), there

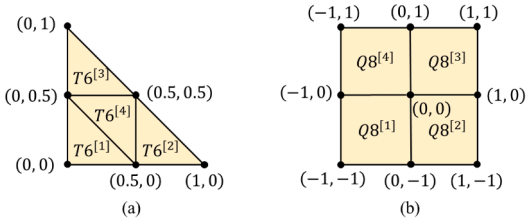

Fig. 8. The subdivision strategies for $T6$ (a) and $Q8$ (b), that is used to counteract the issues where an element contains multiple degenerate points.

is exactly one interior point where $(f_1, f_2) = (0, 0)$. However, this method can only precisely predict the situation where only one degenerate point is within the element. For cases where the degenerate point is located at the element edge or multiple isolated degenerate points exist in an element, one still needs to subdivide the element for a deeper search, since these two situations may lead to $w = 0$, which also corresponds to the situation where the element does not contain a degenerate point.

To maintain a simple and robust procedure, we always subdivide the $T6$ and $Q8$ into four smaller subdomains. Within each subdomain, we assume at most one degenerate point and locate it, if present, by a standalone Newton iteration. In practice, the subdomains are small, and the deviatoric stress field is nearly affine within them, making multiple roots within a single subdomain highly unlikely. Even if it occurs in rare cases, we report a single representative. Note here that the deviatoric stress vectors in these subdomains are still interpolated by the parent element shape function, i.e., we only separate them by the definition domains of the involved natural coordinates. If the degenerate point is located precisely at the shared edge of different subdomains, a merging operation is performed. Fig. 8 shows the subdivision strategy, where $T6^{[i]}$ and $Q8^{[i]}$ with $i = 1 : 4$ refer to the partitioned subdomains. Fig. 7e–h show the located degenerate point(s) and the behaviors of the corresponding principal stress directions in an element containing the degenerate point, where it is worth mentioning that the selected T_6 element has two degenerate points (Fig. 7g, k).

4.2. Topological skeleton

As mentioned above, the topological skeleton is a set of special PSLs that originate at degenerate points. To tackle the problem of indeterminable principal stress directions (i.e. eigenvectors) at the degenerate point, we follow the procedure introduced by Delmarcelle et al. [33] and resolve the indeterminacy by considering the local behavior of the tensor field. In this method, the deviatoric part is expanded into a first-order Taylor series around the degenerate point (x^*, y^*). I.e., for a small displacement $\Delta = (\Delta x, \Delta y)^{\mathrm{T}}$

$$g((x^*, y^*) + \Delta) = g(x^*, y^*) + \boldsymbol{M}(x^*, y^*)\Delta + O(\|\Delta^2\|) \tag{25}$$

Since (x^*, y^*) is a degenerate point, i.e., $g(x^*, y^*) = \mathbf{0}$, the linear part can be approximated by

$$\boldsymbol{g}((x^*, y^*) + \Delta) \approx \boldsymbol{M}\boldsymbol{\Delta} \tag{26}$$

Here, $\boldsymbol{M}$ is matrix of first partial derivatives of $\boldsymbol{g}$ in Eq. (24)

$$\boldsymbol{M} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{27}$$

This implies that the pair (f_1, f_2) transforms under a rotation of the spatial coordinates by an angle θ as a 2D vector rotated by 2θ. This is also reflected in Mohr's circle where principal directions rotate at twice the coordinate angle. Consequently, the determinant of $\boldsymbol{M}$ denoted by $\delta = ad - bc$ is invariant under such rotations. According to [33], this invariant δ at a degenerate point is used to determine both the index and the sector structure

$$\text{index } I = \frac{1}{2}\delta, \quad \delta < 0 \Rightarrow \text{ trisector}, \quad \delta > 0 \Rightarrow \text{ wedge} \tag{28}$$

Equivalently, a trisector has three hyperbolic sectors (Fig. 7i–l), and a wedge has one hyperbolic sector (Fig. 7k). These are the only two types of stable degenerate points in a planar stress field. Intuitively, the process above, which is essentially a mapping from $(\Delta x, \Delta y$ to f_1, f_2 by $\boldsymbol{M}(\Delta x, \Delta y))$, can be understood as how the tensor lifts off isotropy in different directions, and δ being negative or positive flips the local winding of eigen-directions, which is what the tensor index counts.

As prescribed, the principal stress angle θ of an eigenvector satisfies $\tan(2\theta) = \frac{f_2}{f_1}$, and a PSL is a curve whose tangent aligns with an eigenvector. Near a degenerate point (x^*, y^*), a separatrix, i.e., a skeleton branch, is a PSL that leaves (x^*, y^*) along a ray φ for which the direction is self-consistent at leading order. By parameterizing the ray via $(\Delta x, \Delta y) = r \cdot (\cos\varphi, \sin\varphi)$, and using the linear model above, we can have

$$f_1 \approx r(a\cos\varphi + b\sin\varphi), \quad f_2 \approx r(c\cos\varphi + d\sin\varphi) \tag{29}$$

According to the self-consistency condition, i.e., the tangent of the PSL equals the eigenvector direction,

$$\theta \equiv \varphi \,(\mathrm{mod}\,\pi) \quad \Leftrightarrow \quad \tan(2\varphi) = \frac{f_2}{f_1} = \frac{c\cos\varphi + d\sin\varphi}{a\cos\varphi + b\sin\varphi} \tag{30}$$

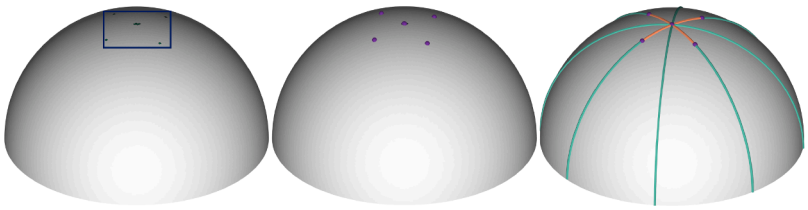

Fig. 9. Pipeline for conducting stress topology analysis. (1). Pre-selecting the candidate elements. (2) Locating degenerate points. (3) Computing the topological skeleton. The hemispherical shell is discretized by $Q8$, where the bottom is fixed and a downward distributed force is applied at the top. The topological skeleton aligning with the major and minor principal stress directions is distinguished by colors, red for major and green for minor. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Let $t = \tan\varphi$ and use $\tan(2\varphi) = \frac{2t}{1-t^2}$. By dividing the numerator and denominator by $\cos\varphi$, we obtain the cubic equation firstly presented in [33] and used in [21]

$$dt^3 + (c + 2b)t^2 + (2a - d)t - c = 0 \tag{31}$$

Solving this cubic equation, its real roots give the admissible initial directions for the separatrices of the topological skeleton. This is because tracing a PSL can be considered solving an ordinary differential equation that fulfills $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\theta(x, y)$. Along a straight line $y = tx$, the limiting slope must equal the field direction. The cubic encodes exactly this fixed-point condition for the linearized field, so each real root is a direction where a PSL can approach or leave the degenerate point without immediate turning, i.e., a separatrix. The cubic has at most three real roots. In the classical 2D theory, trisectors and wedges, respectively, have three and two separatrices, since in the later case the parabolic sector collapses to one, as shown in Fig. 7i–l.

To determine the coefficients a, b, c, and d in Eq. (31), we use the mapping ($\boldsymbol{J}$) defined in Eq. (15) to convert the explicitly expressed $\frac{\partial f_1}{\partial \xi}$, $\frac{\partial f_1}{\partial \eta}$, $\frac{\partial f_2}{\partial \xi}$, and $\frac{\partial f_2}{\partial \eta}$ to $\frac{\partial f_1}{\partial x}$, $\frac{\partial f_1}{\partial y}$, $\frac{\partial f_2}{\partial x}$, and $\frac{\partial f_2}{\partial y}$. Specifically, for any scalar field on the surface, the chain rule gives

$$\begin{bmatrix} a \\ b \end{bmatrix} = \boldsymbol{J}^{-\mathrm{T}} \begin{bmatrix} \frac{\partial f_1}{\partial \xi} \\ \frac{\partial f_1}{\partial \eta} \end{bmatrix}, \quad \begin{bmatrix} c \\ d \end{bmatrix} = \boldsymbol{J}^{-\mathrm{T}} \begin{bmatrix} \frac{\partial f_2}{\partial \xi} \\ \frac{\partial f_2}{\partial \eta} \end{bmatrix} \tag{32}$$

Although different shell elements interpolate stresses with different orders, the topological skeleton near a degenerate point is governed solely by the leading-order departure from isotropy, i.e., by the first derivatives of the in-plane stress tensor in the local tangent frame. Linearizing the deviatoric representation $\boldsymbol{g}$ at the degenerate point yields $\boldsymbol{g} \approx \boldsymbol{M}\Delta$. This first-order model alone determines both the rotation-invariant classifier $\delta = \det(\boldsymbol{M})$ and the separatrix directions obtained as the real roots of the standard cubic equation derived from the self-consistency condition between the PSL tangent and the principal directions. Because these constructions depend only on $\boldsymbol{M}$ at the point and are invariant under smooth reparameterization ($\mathcal{N} \to \mathcal{G} \to \mathcal{L}$) and uniform scalings of (f_1, f_2), they are independent of the underlying element formulation. In other words, the same analysis applies uniformly to all shell element types considered here, including $T3$, $Q4$, $T6$, and $Q8$ elements.

4.3. Pre-selection of candidate elements for degenerate points

Degenerate points are critical for a robust PSL tracing algorithm, yet they occur under very restrictive conditions: the two in-plane principal stresses must coincide, the stress state must approach pure isotropy, and the deviatoric stress component must vanish. Consequently, large parts of the shell structure are fundamentally incapable of containing such points. For instance, regions dominated by strong tensile-compressive contrast or by uniaxial or shear-dominated stress states. This observation motivates a computationally efficient preselection strategy, i.e., before applying the more elaborate degenerate point search, one can exclude with certainty those elements that cannot satisfy the necessary isotropy condition. By culling these non-candidate elements in a cheap and robust manner, the overall detection procedure becomes significantly more tractable without sacrificing accuracy.

In earlier work by Wang et al. [21], a simple sign-comparison strategy was utilized to achieve such a pre-selection, where they concluded that if the signs of f_1 or f_2 remain consistent at all four nodes of a quadrilateral element in 2D, that element can be excluded from further degenerate point searching. This is effective for first-order elements $T3$ and $Q4$, and even some higher-order elements like $T6$ with non-negative shape functions over the isoparametric triangle. However, it does not naturally extend to $Q8$, where stress variations along edges and within interiors may introduce zeros that are invisible at nodal locations. To achieve a reliable preselection across both first- and second-order shell elements, a more general framework is required.

We therefore adopt *Invariant Bernstein-Bézier Bounding*. The central idea is to represent the scalar fields f_1 and f_2 (Eq. (24)) in the Bernstein-Bézier basis of the isoparametric element. This representation expresses the polynomial as a convex combination of its Bernstein coefficients, which serve as "control values". A fundamental consequence is the convex-hull property, i.e., the range of the polynomial over the element lies entirely within the minimum and maximum of these coefficients. Hence, once f_1 and f_2 are cast in Bernstein-Bézier form, we immediately obtain rigorous element-wise intervals $[f_1^{[\min]}, f_1^{[\max]}]$ and $[f_2^{[\min]}, f_2^{[\max]}]$. If either interval lies strictly above or strictly below zero, then the corresponding function cannot vanish within the element, meaning no degenerate point can exist there, and the element can be safely excluded. The derivation of these intervals for $Q8$ is given in Appendix A.

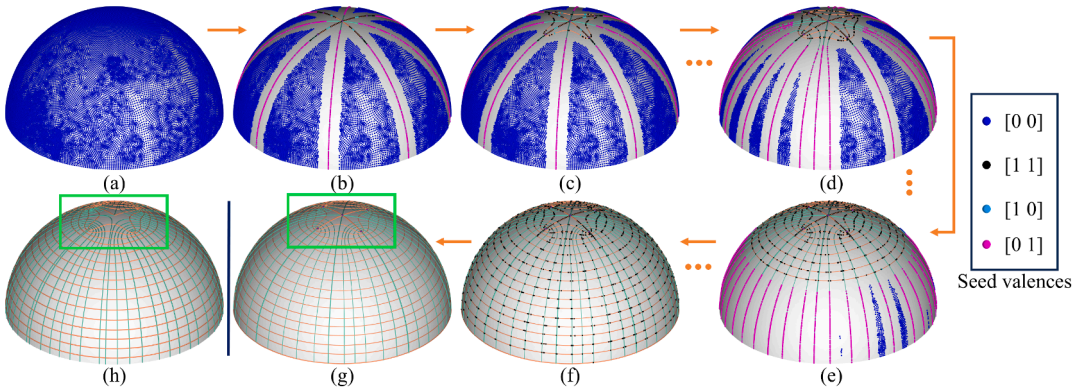

Fig. 10. Procedure of the topology-aware PSLs seeding. (a) Initial seed points. (b) Seed points that are embedded with the topological skeleton. (c)-(e) The intermediate results of the seeding process, where the seeds with different valence values are distinguished by colors.

It is worth noting that one could construct more precise exclusion rules by directly bounding the isotropy measure $2\sqrt{f_1^2 + f_2^2}$ using vector-valued Bernstein control points. While such refinements may exclude additional elements, their increased complexity offers little practical benefit, as the interval-based criterion already eliminates the vast majority of non-candidate elements at negligible cost. This criterion is computationally efficient, invariant, and order-agnostic. For $T3$, $Q4$, and $T6$, the Bernstein coefficients coincide with the nodal values, reducing the method to the original sign-checking rule. In contrast, for $Q8$, mid-edge and interior control values are naturally incorporated, ensuring that interior stress variations are correctly captured. More generally, the Bernstein-Bézier representation extends seamlessly to arbitrary orders, providing a unified and mathematically certified tool for preselecting candidate elements in degenerate point searching.

Fig. 9 shows the pipeline for conducting the stress topology analysis. It begins with the pre-selection strategy mentioned to distill the candidate elements that may contain degenerate points. Then, precisely locating the positions of the degenerate points on the candidate elements. Finally, from the located degenerate points, the corresponding topological skeleton is computed.

5. Topology-aware PSL seeding

The placement of PSLs is a decisive factor in how effectively a stress field can be conveyed. In general, the objective of PSL seeding is to produce a set of trajectories that are domain-filling, evenly spaced, and visually balanced, thereby enabling a global understanding of the directional structure of the stress field. Due to the strong convergence and divergence of principal directions, a naive seeding strategy often results in PSLs that cluster in specific regions, while leaving other areas sparsely populated or completely empty. Moreover, the presence of degenerate points and separatrices further complicates the distribution, potentially producing large void regions that obscure essential field characteristics. Although such patterns reflect intrinsic properties of the stress field to a certain degree, an uneven or incomplete set of PSLs can significantly hinder both interpretation and subsequent engineering use. From a visualization perspective, a balanced PSL distribution ensures consistent coverage, reduces clutter, and provides a faithful representation of the stress field. From a design perspective, regularly distributed PSLs serve as valuable primitives for applications such as rib placement and fiber-reinforcement paths, where global completeness and uniformity are indispensable. For these reasons, a topology-aware PSL seeding strategy is required to explicitly account for degenerate structures, ensure spatial balance, and preserve the meaningful features of the stress field, while enabling its effective use in both analysis and design.

From a methodological perspective, the central task of PSL seeding is to control the spacing between adjacent PSL trajectories. For shell structures, this can be regarded as treating the major and minor principal stress fields as two independent pseudo-vector fields and applying streamline seeding techniques developed in flow visualization. Since major and minor PSLs typically intersect, handling them separately does not increase geometric complexity. Nevertheless, concurrent consideration of both major and minor principal stress directions ensures that the resulting PSL set is space-filling and topologically consistent. This problem has been systematically discussed by Wang et al. [29] in their work for 3D stress fields. The key idea there is to distribute a set of domain-filling seed points and assign each seed a binary valence indicator $[0, 0, 0]$ that records whether the local region is traversed by major, medium, or minor PSLs. When a PSL passes through a region, the associated seeds are merged with that trajectory, and their corresponding valence components are turned to 1. In this way, seeds effectively "share" trajectories, and a merging threshold provides direct control over spacing and sparsity. Our PSL seeding strategy extends this concept to shell structures by introducing adaptations specifically tailored to surface-based stress fields.

The first adaptation for shell structures involves replacing the Euclidean distance metric with the geodesic distance along the shell surface when the shell structure has high curvature, as seen in the wing example reported in Fig. 1. In the volumetric case considered by Wang et al. [29], Euclidean distances between seeds and PSL trajectories provide a natural and consistent measure of spacing. However, for shell structures, PSLs are intrinsically confined to the curved surface, and Euclidean distances may underestimate proximity in regions with high curvature. By adopting a geodesic distance metric, spacing is evaluated in the intrinsic geometry of

the surface, thereby ensuring that the merging of seeds with PSL trajectories reflects the true physical neighborhood on the shell. This adaptation enables the spacing control mechanism to remain consistent with the shell's geometry, preventing distortions that would arise from using Cartesian distances. In our implementation, we use an approximate method to compute geodesic distances by counting the lengths along the mesh edges.

The second adaptation involves incorporating stress topology analysis as a pre-processing step for PSL seeding. While the overall strategy still relies on a set of domain-filling seed points (Fig. 10a), regions associated with the topological skeleton are treated with priority. Specifically, merging operations are first carried out along the topological skeleton to ensure that these critical structures are faithfully represented (Fig. 10b). The remaining PSLs are then traced to cover the remaining seeds until all points are traversed. This hierarchical procedure integrates the intrinsic topology of the stress field into the PSL distribution, guaranteeing that degenerate structures are explicitly captured while maintaining a balanced, space-filling coverage of the entire shell domain.

The entire seeding process is governed by the binary valence indicator assigned to each seed point, as shown in Fig. 10a–g. Since the stress field of a shell structure involves only major and minor PSLs, the valence is initialized as a bi-value vector. Its possible states are $[0,0]$ (no PSLs passed), $[1,0]$ (passed by a major PSL), $[0,1]$ (passed by a minor PSL), and $[1,1]$ (passed by both), refer to the remarks in Fig. 10. During seeding, seed points with non-filled valence are iteratively selected (Fig. 10c–e), and the process terminates once all seed points reach the state $[1,1]$ (Fig. 10f). Provided that the initial seeds cover the domain (Fig. 10a), this mechanism ensures that the resulting PSLs are evenly spaced and collectively span the entire surface (Fig. 10g). Moreover, if one is interested only in either the major or the minor principal stress direction field, the method can be specialized by initializing the valence to $[1,0]$ or $[0,1]$, respectively. This flexible formulation enables the generation of PSL distributions that are both complete and balanced, while remaining easily adaptable to single-direction analyses, thus making the method broadly applicable to visualization and design tasks. Additionally, we show in Fig. 10h the PSL distribution without considering topology analysis, i.e., directly conducting seeding from the initial seed points. By comparing it to Fig. 10g, it can be clearly seen that the proposed strategy achieves a more consistent structure around the degenerate points.

In principle, seed points may be initialized arbitrarily as long as they lie on the mesh. In this work, we simply take the element centers as seeds, which guarantees domain coverage without additional pre-processing. To prescribe the PSL spacing in a dimensionless and size-independent manner, we use the minimal edge length $D_{\min}$ of the object's bounding box as a reference and define the spacing threshold as $D_{\min}/\omega$, where ω is a user-specified control parameter. Larger values of ω reduce the spacing and therefore produce a denser PSL distribution.

6. Results

In this section, we demonstrate the capabilities of the proposed framework, entitled *PSLshell*, through a series of representative experiments. We begin by employing a planar benchmark to highlight the core ideas of *PSLshell* and to contrast it with existing pipelines. A cylindrical shell under torque is used to showcase the framework's support for multiple discretization formats, followed by a polyhedral structure with sharp edges to evaluate robustness across creases and element-orientation variations. To further assess robustness on highly complex shell topologies, we also apply *PSLshell* to triply periodic minimal surface (TPMS) shells [53], which feature strong curvature variations and multiple holes. Finally, several examples from engineering, architecture, and nature demonstrate the broad practical applicability of the framework. In all results, unless otherwise stated, major and minor PSLs are respectively rendered in red and green. The topological skeleton adopts the same color convention, but with thicker trajectories. Degenerate points are depicted as purple spheres.

The stress datasets used in our experiments are primarily generated with ABAQUS, which provides node-wise stress fields in the local frame. To further validate the generality of *PSLshell*, we also employ an open-source FEM package (*MiniFEM*[1]) to produce stress fields represented in the global Cartesian frame. For stress simulation, the per-element local frame is defined using the default reference vector $[1,0,0]$ in ABAQUS. All components of *PSLshell* are implemented in a single MATLAB script ("*PSLshell.m*"). A lightweight ASCII format (".TSV") is introduced to interface stress data from common CAE packages. Additional implementation details, usage instructions, and datasets are provided in the public code repository. In *PSLshell*, the stress field is interpreted as an in-plane tensor field defined on the shell surface, and the framework analyzes the corresponding in-plane stress behavior. The specific stress quantity to be visualized (e.g., membrane stresses, bending-related stresses, or stress resultants) is selected by the user during stress extraction and represented as a symmetric in-plane stress tensor in the local shell frame.

The computational cost of *PSLshell* is modest and is governed primarily by the number of seeds and the number of PSLs to be generated. The dominant expense arises from computing distance maps between seeds and existing trajectories. For thin-featured geometries such as wings or blades, the geodesic distance metric is required for reliable spacing control, which is slower than the Euclidean alternative. Across all examples, PSL generation completes within approximately 2 s (the cylinder case below, where trajectories are traced from prescribed positions without spacing control) to about 2 min (the wing case, where the full *PSLshell* pipeline is employed), adding only a negligible overhead to the overall structural analysis process.

Planar Plate. In this example, we consider a planar rectangular domain discretized with $T3$ elements. The four corner vertices are fixed, and a uniform vertical load is applied across the top surface, inducing a bending deformation (Fig. 11a). The corresponding von Mises stress distribution is shown in Fig. 11b for reference. The PSL distribution generated by *PSLshell* is shown in Fig. 11c and

[1] https://github.com/PSLer/MiniFEM

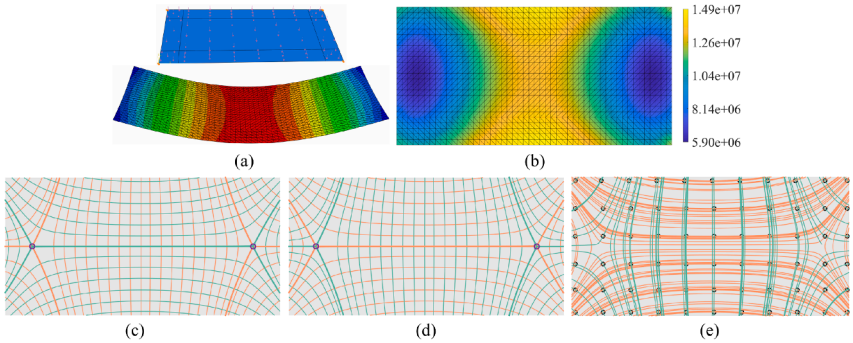

Fig. 11. (a) Problem description and deformation pattern of the planar plate. (b) Von Mises stress for reference. (c) and (d) PSLs that are generated by the proposed framework, separately from the stress fields on the top surface and the bottom one. (e) PSLs distribution generated by Karamba3D, with the same stress field in (d). The major and minor PSLs are distinguished by orange and green. The thicker PSLs in (c) and (d) are the topological skeleton, and circles in purple indicate the degenerate points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

d, clearly illustrating the key features of our framework: a domain-filling, evenly spaced PSL distribution built upon a high-fidelity tracing algorithm and a systematic treatment of stress singularities.

The difference between Fig. 11c and d lies in the stress field used: (c) corresponds to the top surface, while (d) corresponds to the bottom. As expected from Fig. 11a, the top surface is primarily under horizontal compression, whereas the bottom surface is dominated by horizontal tension. This difference is reflected in the PSL orientation, i.e., in Fig. 11c, the minor PSLs (green) align with the horizontal direction under compression, while in Fig. 11d, it is the major PSLs (orange) that align horizontally in response to tension.

For comparison, Fig. 11e shows the result generated using a widely adopted approach based on pre-computed principal stress directions. Since that method does not systematically address stress singularities or PSL spacing, it exhibits noticeable inconsistencies near degenerate points and suffers from visual clutter. From the perspective of PSL-guided lightweight structural design [5,25,41], where clustering of dense PSLs is typically required to extract a structured, manufacturable layout, our method directly yields a well-organized PSL distribution without post-processing or trajectory distortion.

Cylinder. In this experiment, we revisit the cylindrical shell shown in Fig. 6 and discretize it using six different mesh types: $T3$, $Q4$, hybrid $T3$-$Q4$, $T6$, $Q8$, and hybrid $T6$-$Q8$ (Fig. 12, top). This setup allows us to evaluate the adaptivity of *PSLshell* across diverse discretization formats. The experiment is designed with two complementary objectives. First, to enable a fair comparison of the PSL tracing algorithm, the proposed PSL-seeding strategy is disabled, and PSLs are traced from an identical set of prescribed initial positions for all discretizations. This allows the influence of mesh type on the tracing procedure itself to be isolated. As shown in the second row of Fig. 12, all discretizations produce coherent and physically meaningful PSL trajectories, confirming that the element-transition strategy operates reliably across different mesh types. Second, the proposed PSL-seeding strategy is enabled on the same set of discretizations. The resulting PSL distributions, shown in the third row of Fig. 12, exhibit consistent global patterns with similar density and regularity, demonstrating that the seeding algorithm remains robust with respect to variations in mesh type and discretization.

A closer comparison of the PSL trajectories in Fig. 12 further shows that, under identical boundary conditions, all discretizations yield the same stress trajectory patterns. This provides strong evidence for the robustness of *PSLshell* across supported mesh types. It is essential to note that this consistency does not diminish the value of high-order shell elements in practice, particularly when high geometric fidelity is required. First-order elements approximate the geometry only in a piecewise linear manner, which may introduce visible gaps between PSL trajectories and the intended curved surface. Such geometric discrepancies are substantially reduced when higher-order elements are used. Finally, although this example focuses on structured meshes for clarity of comparison, *PSLshell* handles unstructured meshes equally well (see Fig. 6).

Polyhedra-like structure. This example illustrates the capability of *PSLshell* to handle shell structures featuring sharp creases. We consider a polyhedra-like shell discretized with $T6$ elements (Fig. 13b), subject to the boundary conditions shown in Fig. 13a. The geometry features multiple creases, including both convex ridges and concave valleys, creating a challenging environment for PSL propagation. As shown in Fig. 13c, our strategy based on the signed dihedral angle and Rodrigues rotation performs robustly when PSLs traverse these creases, regardless of whether the transition is across a ridge or a valley. Fig. 13d presents the same result from an alternative viewpoint to better highlight the PSL geometry and the associated stress topology. In Fig. 13e, we suppress the background surface and color-code each PSL according to its corresponding principal stress component. A close-up view of a selected crease further confirms the correctness of *PSLshell* in handling PSL propagation across sharp geometric transitions, where the arrows, matching the color of the nearby PSLs, indicate the local principal stress directions.

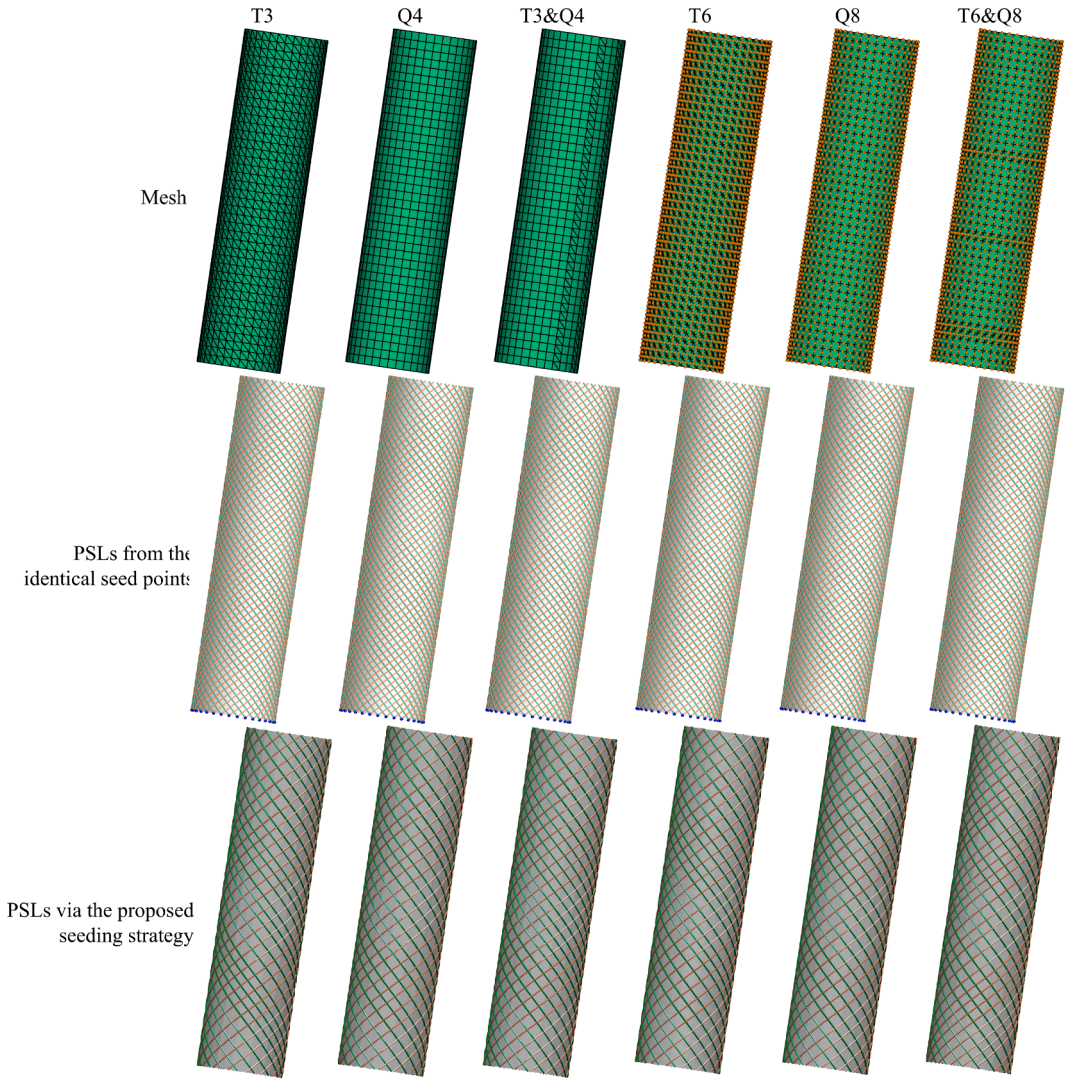

Fig. 12. Top row: Different discretizations of the cylinder model. Middle row: the corresponding PSLs traced from an identical set of prescribed initial positions. Bottom row: the corresponding PSLs generated by the proposed PSL-seeding strategy. The yellow spheres in the top row indicate the mid-nodes of the second-order elements, and the orange and green trajectories separately represent the major and minor PSLs that are traced from the same positions denoted in blue spheres. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

TPMS. This experiment examines the robustness of *PSLshell* on shell structures with highly complex geometry and topology, characterized by strong curvature variations and multiple holes. We consider four representative TPMS shells: the Schwarz-P, Diamond, Neovius, and Gyroid surfaces, each arranged in a $2 \times 2 \times 2$ periodic configuration. All shells are discretized using $T3$ elements, fixed at the bottom edges, and subjected to a downward load applied on the top edges. The top row of Fig. 14 shows the corresponding surface meshes, while the second row presents the traced PSLs. Despite the intricate geometry, dense perforations, and frequent proximity to degenerate regions, *PSLshell* produces coherent and smoothly varying PSL patterns across all cases. These results demonstrate that the proposed framework remains stable on multiply connected shell domains with complex boundaries, and further highlight its applicability to complex shell structures encountered in metamaterial design and additive manufacturing.

Wing and wind turbine blade. In this experiment, we first examine a wing model discretized with $T6$ elements and subjected to boundary conditions that approximate in-situ aerodynamic loading (Fig. 15a). The resulting PSL distribution is shown in Fig. 15b. In addition to the regular spacing of the PSLs, a clear physical pattern emerges: the minor PSLs on the upper surface (Fig. 15f) converge toward the wing root, where the structure is fixed, while the major PSLs on the lower surface (Fig. 15d) exhibit a similar convergence. This behavior aligns with the actual stress state of aircraft wings in flight, i.e., lift causes the wing to bend upward, producing dominant compressive stresses on the upper surface and tensile stresses on the lower surface. Fig. 15d also provides a

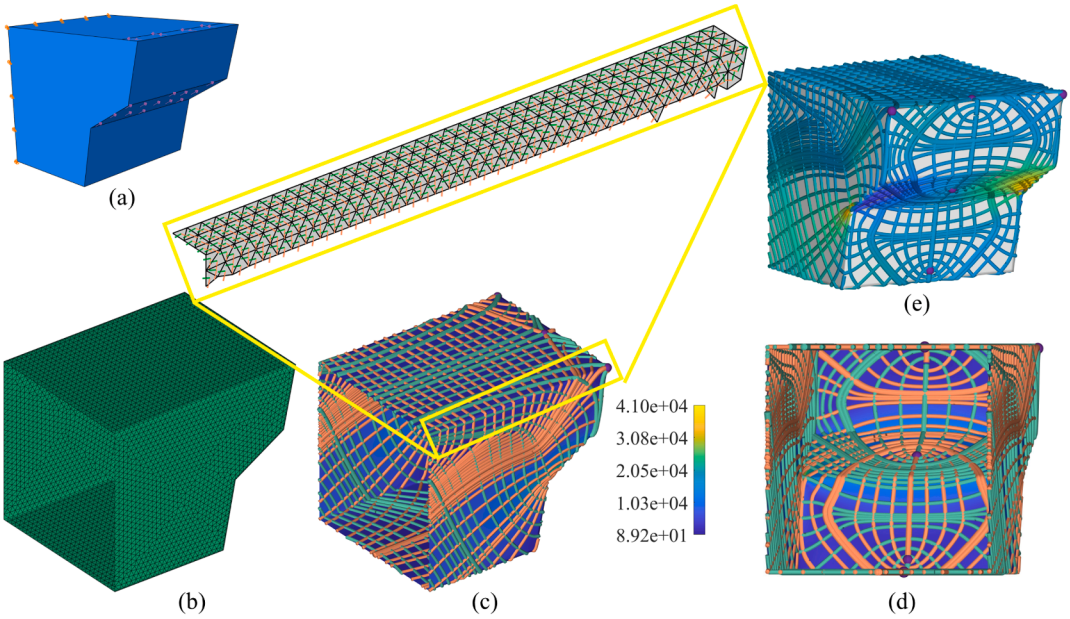

Fig. 13. PSLs on the polyhedron. (a) shows the used $T6$ mesh, where the four edges on the left are fixed, and two distributed forces are separately applied at the top surface and the slope surface on the right. (b) shows the generated PSLs that are drawn on the color map of the von Mises stress field. The close-up view shows the principal stress directions in the corresponding region. (c) shows the same result as (b) but in a different viewing angle. (d) shows the same PSLs before, but both the major and minor PSLs are color-coded with the shear stress component (σ_{xy}) along the PSL trajectories. In (b) and (c), the orange and green tubes correspond to the major and minor PSLs. The topological skeleton and degenerate points are indicated by the thicker tubes and purple spheres, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

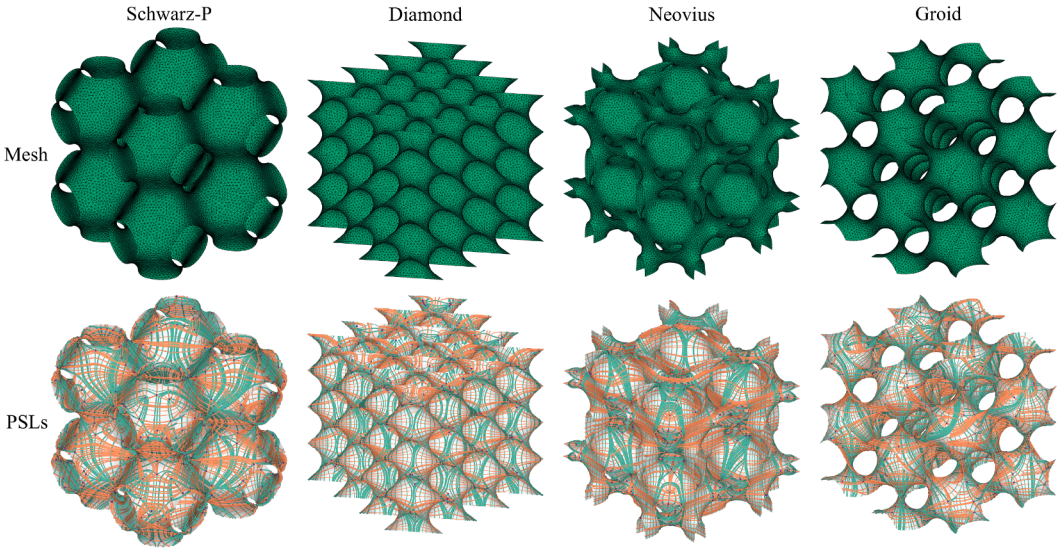

Fig. 14. Top: The $T3$ meshes of the considered TPMS shells. Bottom: the corresponding PSLs.

close-up view of the trailing edge, confirming the robustness of our crease-crossing strategy in PSL propagation. A different viewing angle in Fig. 15c further highlights the orthogonality of the major and minor PSLs. In Fig. 15e, both major and minor PSL groups are color-coded according to the von Mises stress along each trajectory, clearly illustrating the expected stress intensification toward the wing root.

We next consider a wind turbine blade [54], discretized with $T3$ elements (Fig. 16a). The corresponding stress topology, consisting of the topological skeleton and degenerate points, is shown in Fig. 16b. Fig. 16c and d present the complete PSL distributions. In Fig. 16c, all PSLs are color-coded with the von Mises stress along their trajectories, whereas in Fig. 16d, major and minor PSLs are

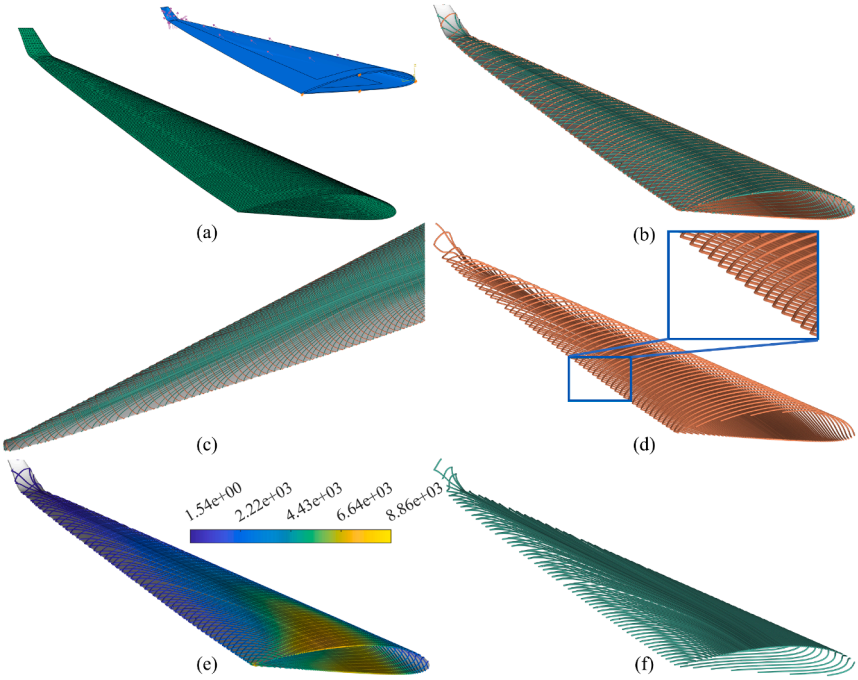

Fig. 15. (a) Problem description and mesh of the wing example. (b) The major (orange) and minor (green) PSLs. (c) The same PSLs but in a different viewing angle to show the orthogonality of the major and minor PSLs. (d) and (f) the major and minor PSLs shown separately. (e) The PSLs that are color-coded with von Mises stress. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

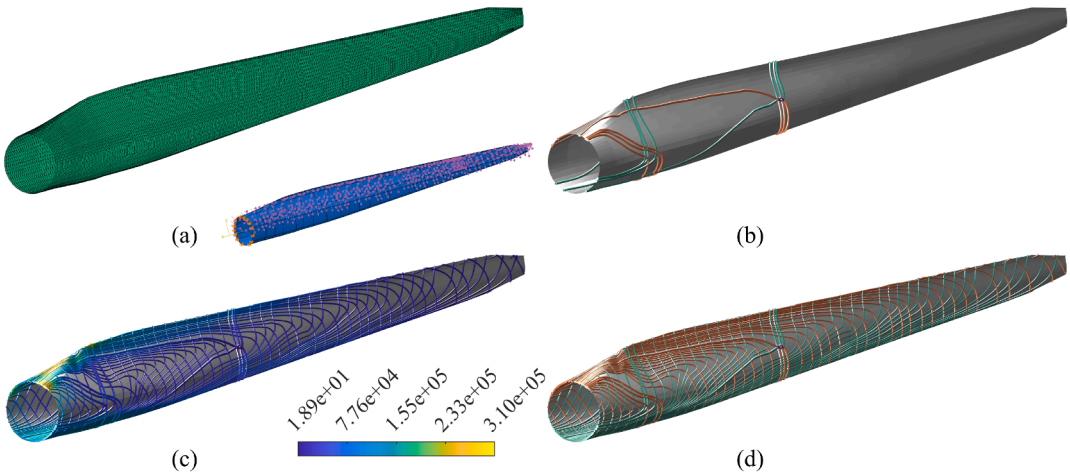

Fig. 16. (a) Problem description and mesh of the blade example. (b) The degenerate points and topological skeleton. (c) and (d) The PSLs that are color-coded differently, in (c), the color map shows the shear stress components along the PSL trajectories.

distinguished to highlight their structural roles. These results illustrate the applicability of *PSLshell* to complex freeform geometries commonly encountered in wind energy engineering.

Doom. In this experiment, we consider a dome-like structure inspired by [5,25], discretized with $Q8$ elements. The short edges along the base (P. 1-9 in Fig. 17a) are fixed by constraining the translational DOFs in the Y- and Z-directions and the rotational DOF around the Z-axis. The stress field is induced by the self-weight of the structure. Fig. 17b shows the resulting major and minor PSLs, while Fig. 17c presents a top view in which both major and minor PSL groups are color-coded by their corresponding principal stress magnitudes.

Beyond exhibiting a regular and coherent PSL layout, the visualization also reveals a strong correlation between PSL convergence and elevated stress levels near the constrained boundary. This provides a clear indication of structurally critical or “weak” regions and

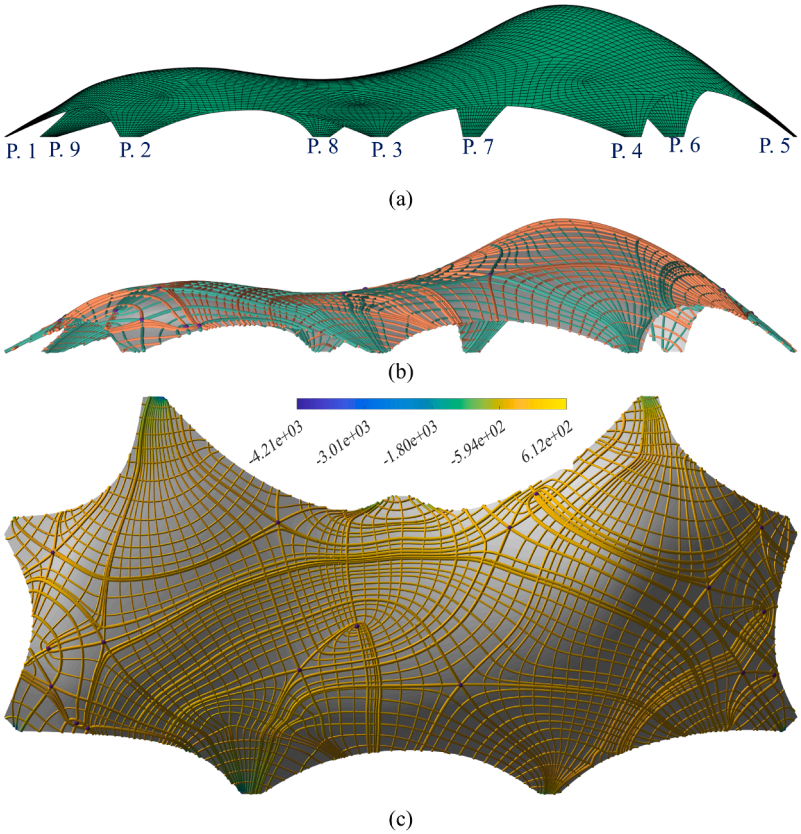

Fig. 17. (a) Dome model, P. 1–9 indicate the fixed positions. (b) The major and minor PSLs. (c) The PSLs color-coded with the corresponding principal stress values and viewed from the top.

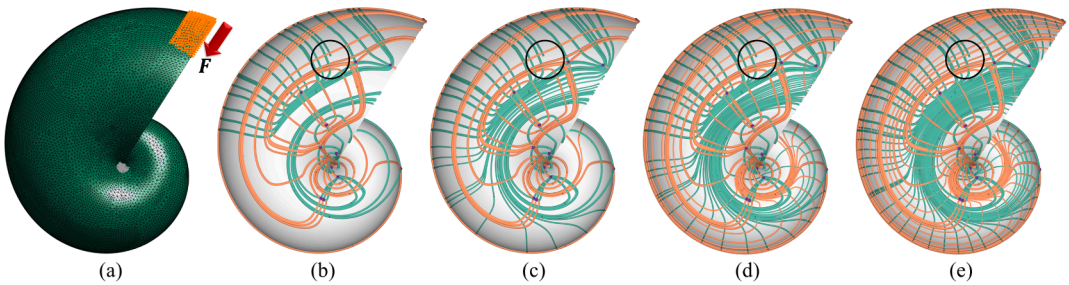

Fig. 18. (a) Problem description of the snail shell, where the orange and grey regions indicate the loaded and fixed nodes, respectively. (b) The topology of the stress field. (c) to (e) The resulting PSLs under the density control parameter $\omega = 6,\ 10,\ 14$ sequentially.

offers intuitive guidance on where stiffeners or reinforcement should be placed. These observations underline the utility of *PSLshell* in supporting early-stage conceptual and performance-driven architectural design.

Snail shell. We conclude the experimental section with a nature-inspired example, a snail shell, chosen to illustrate a current limitation of *PSLshell*. The structure is discretized with $T3$ elements, and the boundary conditions are indicated in Fig. 18a. Fig. 18b–e first display the stress field topology and then show PSL distributions of progressively increasing density, controlled by the parameter ω introduced in Section 5. In this case, several minor PSLs belonging to the topological skeleton terminate prematurely, as highlighted by the black circles in Fig. 18b-e. This phenomenon is concurrently caused by the spiral geometry and the intricate stress topology, resulting in trajectories that wind indefinitely rather than converging outward. As a result, the tracing process becomes constrained by the maximum number of integration steps, leading to visually discontinuous trajectories. This behavior represents an inherent limitation of the current tracing algorithm when applied to geometries with strong spiral features and nontrivial topology.

7. Conclusion and future work

In this paper, we have introduced a novel framework that systematically addresses the critical challenges associated with generating principal stress lines (PSLs) on shell structures. The proposed approach comprehensively accommodates the practical conditions in shell analysis, where higher-order elements are frequently employed to represent complex curvatures. It consistently supports both first- and second-order triangular and quadrilateral elements by rigorously adhering to the element shape functions, thereby avoiding the geometric distortion and artificial discontinuities introduced by piecewise linear subdivision. Furthermore, the framework enables seamless PSL propagation across sharp creases, ensuring continuity of stress trajectories on non-smooth surfaces. The inherent singularities in shell stress fields are systematically analyzed and resolved through an extended stress topology formulation, which guarantees the topological consistency of PSLs in degenerate regions. By integrating stress topology with surface-based spatial computations, the framework achieves domain-filling and evenly spaced PSL distributions over curved shells while maintaining local coherence near singularities.

All components of the methodology are implemented in a unified and openly available MATLAB package, providing a robust and user-friendly platform for stress field investigation, design evaluation, and the development of PSL-guided lightweight structural design strategies. Beyond serving as an advanced tool for stress investigation, the framework establishes a general foundation for diverse domain-oriented applications. The structured PSL patterns provide direct insights into load transfer mechanisms and offer quantitative guidance for the placement of stiffeners, the orientation of fibers in composite laminates, and the design of lightweight shell structures that follow principal stress trajectories. Through a series of examples, the framework has been demonstrated to capture the stress characteristics across a variety of shell geometries. These examples collectively illustrate the framework's potential to bridge stress analysis and structural design, providing a unifying basis for future research and applications that couple mechanical insight with geometry-driven design principles.

Although the proposed framework provides a comprehensive solution for generating domain-filling and topology-consistent PSLs on shell structures, several extensions are envisioned. The current implementation is limited to 2-manifold shell meshes, and its adaptation to non-manifold configurations such as intersecting webs would broaden its applicability. Incorporating physically informed adaptive seeding strategies could further enhance the quality and efficiency of PSL distribution. Moreover, coupling PSLs with geometric parametrization techniques presents an opportunity for field-aligned modeling and PSLs-guided structural optimization on curved shells. These directions will further strengthen the framework's integration with computational design and geometry processing.

CRediT authorship contribution statement

Junpeng Wang: Conceptualization, Investigation, Methodology, Software, Writing – original draft; **Yingjian Liu:** Methodology, Software, Writing – review & editing; **Jun Wu:** Conceptualization, Methodology, Writing – review & editing, Supervision; **Rüdiger Westermann:** Conceptualization, Methodology, Writing – review & editing, Supervision, Funding acquisition.

Data availability

A public link to the code and data is provided in the abstract of this paper.

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.

Acknowledgments

This work was supported in part by a grant from the German Research Foundation (DFG) under grant number WE 2754/10-1. We acknowledge Dr. Dennis Bukenberger at the Technical University of Munich for contributing to the dome dataset. We also acknowledge the creators of the open-source CAD models used in the snail shell and wing examples, which were obtained from the *Craftsmanspace* and *GrabCAD* online repositories, respectively.

Appendix A. Derivation of Bernstein-Bézier interval bounds for quadratic elements

In this appendix, we illustrate how the element-wise intervals $[f_1^{[\min]}, f_1^{[\max]}]$ and $[f_2^{[\min]}, f_2^{[\max]}]$ are obtained using Bernstein-Bézier representations for quadratic elements $Q8$.

On the isoparametric element of $Q8$, defined by natural coordinate $(\xi, \eta) \in [-1, 1]^2$, we introduce affine coordinates

$$u = \frac{\xi + 1}{2}, \quad v = \frac{\eta + 1}{2}, \quad (u, v) \in [0, 1]]^2 \tag{A.1}$$

Each stress-derived scalar field $f \in \{f_1, f_2\}$ is biquadratic in (u, v). We evaluate it on the 3×3 grid

$$(u, v) \in \{0, \frac{1}{2}, 1\} \times \{0, \frac{1}{2}, 1\} \tag{A.2}$$

which corresponds to the eight nodes of $Q8$ and the element center. Collecting the values of f on (u, v), we can form a 3×3 matrix $\boldsymbol{S}$ with each entry s_{ij} being given by $f(u_i, v_j)$.

As known in 1D, the quadratic Lagrange values $(s_0, s_{\frac{1}{2}}, s_1)$ relate to the Bernstein coefficients (b_0, b_1, b_2) through

$$b_0 = s_0, \quad b_2 = s_1, \quad b_1 = 2s_{\frac{1}{2}} - \frac{1}{2}(s_0 + s_1) \tag{A.3}$$

This defines a fixed 3×3 linear transformation $\boldsymbol{C}$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 2 & -\frac{1}{2} \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{C}} \begin{bmatrix} s_0 \\ s_{\frac{1}{2}} \\ s_1 \end{bmatrix} \tag{A.4}$$

The biquadratic Bernstein coefficients are obtained by applying this transformation to $\boldsymbol{S}$

$$\boldsymbol{B} = \boldsymbol{C}\boldsymbol{S}\boldsymbol{C}^{\mathrm{T}} \tag{A.5}$$

The convex-hull property ensures

$$\min \boldsymbol{B} \leq f(u, v) \leq \max \boldsymbol{B} \tag{A.6}$$

and thus the element-wise intervals are

$$[f^{[\min]}, f^{\max}] = [\min \boldsymbol{B}, \max \boldsymbol{B}] \tag{A.7}$$

References

[1] K.-M.M. Tam, C.T. Mueller, J.R. Coleman, N.W. Fine, Stress line additive manufacturing (SLAM) for 2.5-D shells, J. Int. Assoc. Shell Spatial Struct. 57 (4) (2016) 249–259. https://doi.org/10.20898/j.iass.2016.190.856

[2] W. Li, A. Zheng, L. You, X. Yang, J. Zhang, L. Liu, Rib-reinforced shell structure, in: Computer Graphics Forum, 36, Wiley Online Library, 2017, pp. 15–27. https://doi.org/10.1111/cgf.13268

[3] G. Liu, W. Huang, Y. Wang, H. Ren, G. Zhang, L. Zhou, Y. Xiong, Stress field-aware infill toolpath generation for additive manufacturing of continuous fiber reinforced polymer composites, Mater. Des. 239 (2024) 112756. https://doi.org/10.1016/j.matdes.2024.112756

[4] Y. Wang, L. Jin, Y. Zhang, P. Hao, B. Wang, CAD-integrated stiffener sizing-topology design via force flow members (FFM), Comput. Methods Appl. Mech. Eng. 415 (2023) 116201. https://doi.org/10.1016/j.cma.2023.116201

[5] J. Zhu, J. Xu, X. Zhou, L. Gao, J. Gao, Principal stress lines (PSLs)-guided discrete topology designs for self-similar porous infills in shell structures with fractal geometry, Thin-Walled Struct. 210 (2025) 112915. https://doi.org/10.1016/j.tws.2025.112915

[6] F. Thamm, The role of the stress trajectories as an aid in the choice of the suitable shape of load-bearing structural elements of engines and structure, Period. Polytech. Mech. Eng. 44 (1) (2000) 171–183.

[7] J.W. Hutchinson, Buckling of spherical shells revisited, Proc. R. Soc. A: Math. Phys. Eng. Sci. 472 (2195) (2016) 20160577. https://doi.org/10.1098/rspa.2016.0577

[8] D.W. Kelly, M.W. Tosh, Interpreting load paths and stress trajectories in elasticity, Eng. Comput. 17 (2) (2000) 117–135. https://doi.org/10.1108/02644400010313084

[9] A.Y. Tamijani, J. Hurley, K. Gharibi, Determination of load paths in plates and shells, Thin-Walled Struct. 127 (2018) 646–653. https://doi.org/10.1016/j.tws.2018.02.031

[10] A.G.M. Michell, LVIII. The limits of economy of material in frame-structures, London, Edinburgh, Dublin Philosophical Mag. J. Sci. 8 (47) (1904) 589–597.

[11] O. Sigmund, K. Maute, Topology optimization approaches, Struct. Multidiscip. Optim. 48 (6) (2013) 1031–1055. https://doi.org/10.1007/s00158-013-0978-6

[12] J. Wu, N. Aage, R. Westermann, O. Sigmund, Infill optimization for additive manufacturing – approaching bone-like porous structures, IEEE Trans. Vis. Comput. Graph. 24 (2) (2018) 1127–1140. https://doi.org/10.1109/TVCG.2017.2655523

[13] E.A. Träff, O. Sigmund, N. Aage, Topology optimization of ultra high resolution shell structures, Thin-Walled Struct. 160 (2021) 107349. https://doi.org/10.1016/j.tws.2020.107349

[14] J. Wu, C. Dick, R. Westermann, A system for high-resolution topology optimization, IEEE Trans. Vis. Comput. Graph. 22 (3) (2015) 1195–1208. https://doi.org/10.1109/TVCG.2015.2502588

[15] N. Aage, E. Andreassen, B.S. Lazarov, O. Sigmund, Giga-voxel computational morphogenesis for structural design, Nature 550 (7674) (2017) 84–86. https://doi.org/10.1038/nature23911

[16] H. Liu, Y. Hu, B. Zhu, W. Matusik, E. Sifakis, Narrow-band topology optimization on a sparsely populated grid, ACM Trans. Graph. (TOG) 37 (6) (2018) 1–14. https://doi.org/10.1145/3272127.3275012

[17] E.A. Träff, A. Rydahl, S. Karlsson, O. Sigmund, N. Aage, Simple and efficient GPU accelerated topology optimisation: codes and applications, Comput. Methods Appl. Mech. Eng. 410 (2023) 116043. https://doi.org/10.1016/j.cma.2023.116043

[18] A. Kratz, M. Schoeneich, V. Zobel, B. Burgeth, G. Scheuermann, I. Hotz, M. Stommel, Tensor visualization driven mechanical component design, in: 2014 IEEE Pacific Visualization Symposium, IEEE, 2014, pp. 145–152. https://doi.org/10.1109/PacificVis.2014.51

[19] T.-H. Kwok, Y. Li, Y. Chen, A structural topology design method based on principal stress line, Comput.-Aided Des. 80 (2016) 19–31. https://doi.org/10.1016/j.cad.2016.07.005

[20] D.R. Bukenberger, J. Wang, J. Wu, R. Westermann, Stress-aligned hexahedral lattice structures, in: Computer Graphics Forum, 44, 2025, p. e15265. https://doi.org/10.1111/cgf.15265

[21] J. Wang, J. Wu, R. Westermann, Stress topology analysis for porous infill optimization, Struct. Multidiscip. Optim. 65 (3) (2022) 1–13. https://doi.org/10.1007/s00158-022-03186-0

[22] J. Wang, J. Wu, R. Westermann, Stress trajectory guided structural design and topology optimization, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 1, American Society of Mechanical Engineers, 2022, p. 1. https://doi.org/10.1115/DETC2022-89030

[23] H. Dong, D. Zhang, X. Zhai, J.-N. Xiao, X.-M. Fu, Porous structure optimization via non-uniform thermal diffusion, Comput. Methods Appl. Mech. Eng. 428 (2024) 117112. https://doi.org/10.1016/j.cma.2024.117112

[24] F. Vergara, T. Zegard, Reconstruction of principal stress lines using FEA applied to Nervi-type shell design, Struct. Multidiscip. Optim. 67 (11) (2024) 193. https://doi.org/10.1007/s00158-024-03913-9

[25] H. Dai, Z. Yu, A new design method for the openings of shell structures under the dual control of principal stress lines and stress contours, Adv. Eng. Software 183 (2023) 103479. https://doi.org/10.1016/j.advengsoft.2023.103479

[26] J. Wang, J. Wu, R. Westermann, A globally conforming lattice structure for 2D stress tensor visualization, Comput. Graphics Forum 39 (3) (2020) 417–427. https://doi.org/10.1111/cgf.13991

[27] T. Delmarcelle, L. Hesselink, Visualizing second-order tensor fields with hyperstreamlines, IEEE Comput. Graph. Appl. 13 (4) (1993) 25–33. https://doi.org/10.1109/38.219447
[28] B. Jobard, W. Lefer, Creating evenly-spaced streamlines of arbitrary density, in: Visualization in Scientific Computing 1997, Springer, 1997, pp. 43–55. https://doi.org/10.1007/978-3-7091-6876-9_5
[29] J. Wang, C. Neuhauser, J. Wu, X. Gao, R. Westermann, 3D-TSV: the 3D trajectory-based stress visualizer, Adv. Eng. Software 170 (2022) 103144. https://doi.org/10.1016/j.advengsoft.2022.103144
[30] X. Zheng, A. Pang, HyperLIC, in: IEEE Visualization, 2003. VIS 2003., IEEE, 2003, pp. 249–256. https://doi.org/10.1109/VISUAL.2003.1250379
[31] Y.M.A. Hashash, J.I.-C. Yao, D.C. Wotring, Glyph and hyperstreamline representation of stress and strain tensors and material constitutive response, Int. J. Numer. Anal. Methods Geomech. 27 (7) (2003) 603–626. https://doi.org/10.1002/nag.288
[32] G. Kindlmann, Superquadric tensor glyphs, in: Proceedings of the Sixth Joint Eurographics-IEEE TCVG Conference on Visualization, 2004, pp. 147–154. https://doi.org/10.5555/2384225.2384248
[33] T. Delmarcelle, L. Hesselink, The topology of symmetric, second-order tensor fields, in: Proceedings Visualization'94, IEEE, 1994, pp. 140–147. https://doi.org/10.1109/VISUAL.1994.346326
[34] L. Hesselink, Y. Levy, Y. Lavin, The topology of symmetric, second-order 3D tensor fields, IEEE Trans. Vis. Comput. Graph. 3 (1) (1997) 1–11. https://doi.org/10.1109/2945.582332
[35] X. Zheng, B. Parlett, A. Pang, Topological structures of 3D tensor fields, in: VIS 05. IEEE Visualization, 2005., IEEE, 2005, pp. 551–558. https://doi.org/10.1109/VISUAL.2005.1532841
[36] J. Palacios, H. Yeh, W. Wang, Y. Zhang, R.S. Laramee, R. Sharma, T. Schultz, E. Zhang, Feature surfaces in symmetric tensor fields based on eigenvalue manifold, IEEE Trans. Vis. Comput. Graph. 22 (3) (2015) 1248–1260. https://doi.org/10.1109/TVCG.2015.2484343
[37] T. Oster, C. Rössl, H. Theisel, Core lines in 3D second-order tensor fields, Comput. Graphics Forum 37 (3) (2018) 327–337. https://doi.org/10.1111/cgf.13423
[38] B. Qu, L. Roy, Y. Zhang, E. Zhang, Mode surfaces of symmetric tensor fields: topological analysis and seamless extraction, IEEE Trans. Vis. Comput. Graph. 27 (2) (2020) 583–592. https://doi.org/10.1109/TVCG.2020.3030431
[39] C. Hergl, C. Blecha, V. Kretzschmar, F. Raith, F. Günther, M. Stommel, J. Jankowai, I. Hotz, T. Nagel, G. Scheuermann, Visualization of tensor fields in mechanics, in: Computer Graphics Forum, Wiley Online Library, 2021. https://doi.org/10.1111/cgf.14209
[40] C. Preisinger, Linking structure and parametric geometry, Archit. Des. 83 (2) (2013) 110–113. https://doi.org/10.1002/ad.1564
[41] L. Liu, X. Wu, J. Huang, L. Cao, X. Liu, C. Tu, L. Lu, Principal stress field-guided optimization for rib structure generation, Comput.-Aided Des. (2025) 103955. https://doi.org/10.1016/j.cad.2025.103955
[42] R. Arora, A. Jacobson, T.R. Langlois, Y. Huang, C. Mueller, W. Matusik, A. Shamir, K. Singh, D.I.W. Levin, Volumetric michell trusses for parametric design & fabrication, in: Proceedings of the ACM Symposium on Computational Fabrication, ACM, 2019, pp. 1–13. https://doi.org/10.1145/3328939.3328999
[43] A. Sheffer, B. Lévy, M. Mogilnitsky, A. Bogomyakov, ABF++: fast and robust angle based flattening, ACM Trans. Graph. (TOG) 24 (2) (2005) 311–330. https://doi.org/10.1145/1061347.1061354
[44] G.-S. Li, X. Tricoche, D. Weiskopf, C.D. Hansen, Flow charts: visualization of vector fields on arbitrary surfaces, IEEE Trans. Vis. Comput. Graph. 14 (5) (2008) 1067–1080. https://doi.org/10.1109/TVCG.2008.58
[45] B. Lévy, S. Petitjean, N. Ray, J. Maillot, Least squares conformal maps for automatic texture atlas generation, in: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, ACM, 2023, pp. 193–202. https://doi.org/10.1145/3596711.3596734
[46] S.J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, J. Comput. Phys. 227 (3) (2008) 1943–1961. https://doi.org/10.1016/j.jcp.2007.10.009
[47] T. Marz, C.B. Macdonald, Calculus on surfaces with general closest point functions, SIAM J. Numer. Anal. 50 (6) (2012) 3303–3328. https://doi.org/10.1137/120865537
[48] C. Kublik, R. Tsai, Integration over curves and surfaces defined by the closest point mapping, Res. Math. Sci. 3 (1) (2016) 3. https://doi.org/10.1186/s40687-016-0053-1
[49] A. Vaxman, M. Campen, O. Diamanti, D. Panozzo, D. Bommes, K. Hildebrandt, M. Ben-Chen, Directional field synthesis, design, and processing, in: Computer Graphics Forum, 35, Wiley Online Library, 2016, pp. 545–572. https://doi.org/10.1111/cgf.12864
[50] D. Panozzo, Y. Lipman, E. Puppo, D. Zorin, Fields on symmetric surfaces, ACM Trans. Graph. (TOG) 31 (4) (2012) 1–12. https://doi.org/10.1145/2185520.2185607
[51] W. Jakob, M. Tarini, D. Panozzo, O. Sorkine-Hornung, et al., Instant field-aligned meshes, ACM Trans. Graph. 34 (6) (2015) 1–15. https://doi.org/10.1145/2816795.2818078
[52] Dassault Systèmes, ABAQUS Version 6.5: Getting Started with ABAQUS - Shell Elements, Section 5.3, Dassault Systèmes, 2009. Chapter 5.3, https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.5/books/gss/ch05s03.html.
[53] I. Maskery, A.O. Aremu, L. Parry, R.D. Wildman, C.J. Tuck, I.A. Ashcroft, Effective design and simulation of surface-based lattice structures featuring volume fraction and cell type grading, Mater. Des. 155 (2018) 220–232. https://doi.org/10.1016/j.matdes.2018.05.058
[54] B.R. Resor, Definition of a 5MW/61.5 m Wind Turbine Blade Reference Model, Technical Report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2013. https://doi.org/10.2172/1095962