Reduction of 3D-FE models to 1D beam models for AI-based simulations

Moritz Treschau[1,*], Christian Witzgall[1], Sandro Wartzack[1]

[1] Engineering Design (KTmfk), Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

** Corresponding author:*
Moritz Treschau
Chair of Engineering Design, Friedrich-Alexander-Universität Erlangen-Nürnberg
Martenstraße 9
91058 Erlangen
☎ *+49 (0) 9131/8523217*
🖂 *treschau@mfk.fau.de*

Abstract

Especially in early stages of product development, it is necessary to evaluate designs using fast and automated simulations. While conventional finite element analysis is computationally expensive, physics-informed neural networks enable real-time prediction. However, they require problem-tailored architectures and therefore physically consistent training data. To address this issue, we present a procedure for automatically reducing 3D-FE models to physically consistent, backward-compatible 1D beam models. This method builds on geometric skeletonisation by mapping mechanical state variables onto the medial axis to preserve geometry, topology and physical information. The resulting models can be used to generate synthetic training data for machine learning models.

Keywords

Skeletonization, Feature Projection, Simulation, Physics-informed neural networks

 | DOI: 10.35199/dfx2026.2

1. Motivation

In the early stages of technical product development, rapid generation and evaluation of multiple design variants is crucial [1]. Conventional approaches for geometry generation, such as topology optimization, and subsequent evaluation using *finite element analysis* (FEA) are often limited by the high computational cost associated with repeatedly assembling and solving large systems of equations. In contrast, AI-based simulation approaches offer considerable potential for real-time prediction [2]. Among these, *physics-informed neural networks* (PINNs) are particularly promising, as they incorporate governing physical laws directly into the learning process, enabling improved generalization while requiring less training data than purely data-driven approaches [3]. These advantages, however, strongly depend on a problem-specific network architecture and physically consistent training data [4].

Although FEA is computationally too expensive for real-time evaluation, its high accuracy and automation level makes it well suited for generating synthetic training data. To develop a robust AI model, it is advantageous to reduce complex structural problems, which are typically represented by three-dimensional solid finite element meshes, to simple representations whose physical behavior can be described analytically [5]. One-dimensional structural elements, such as bars and beams, provide an appropriate starting point, since they constitute one of the most fundamental modeling abstractions in structural mechanics [6]. Although numerous approaches exist for reducing geometry and topology through geometric skeletonization, the automated and physically consistent transfer of mechanical properties to the resulting 1D model remains a significant challenge.

The goal of this work is therefore to investigate an automated approach for reducing three-dimensional solid finite element models to physically consistent one-dimensional beam models.

2. Structural model reduction

The following chapter provides an overview of the current state of the art in structural model reduction. Existing research does not yet offer a comprehensive methodology for reducing structural mechanical models, but instead focuses on individual aspects of the overall process. These can be broadly classified into skeletonization and feature mapping.

2.1. Skeletonization

Skeletonization aims to derive a compact one-dimensional representation of a geometric object while preserving its essential topological characteristics [7]. The theoretical foundation was established by Blum through the *Medial Axis Transform* (MAT), which defines the skeleton as the set of centers of all maximally inscribed spheres inside an object [8]. Subsequent to this, skeletons have become an established shape representation due to the fact that they provide a compact description of geometry, topology, symmetry, and local thickness in a computationally efficient way [7].

Owing to these properties, skeletonization has found applications in numerous disciplines, including computer graphics, medical image analysis, robotics, navigation and metrology [7,9]. Within the domain of engineering design, skeletons enable the reduction of complex three-dimensional components to curve or surface representations that preserve the dominant load paths and connectivity. This compact parameterization facilitates model reduction, geometry editing, design optimization, and the automated reconstruction of feature-based CAD models from topology optimization results. [10,11]

A plethora of skeletonization algorithms have been proposed, which can broadly be classified into contraction-, Voronoi-, and thinning-based approaches, as illustrated in Figure 1 [7]. **Contraction-based methods** determine the skeleton by shrinking the geometry toward its medial structure. This process is commonly formulated as an optimization problem in which

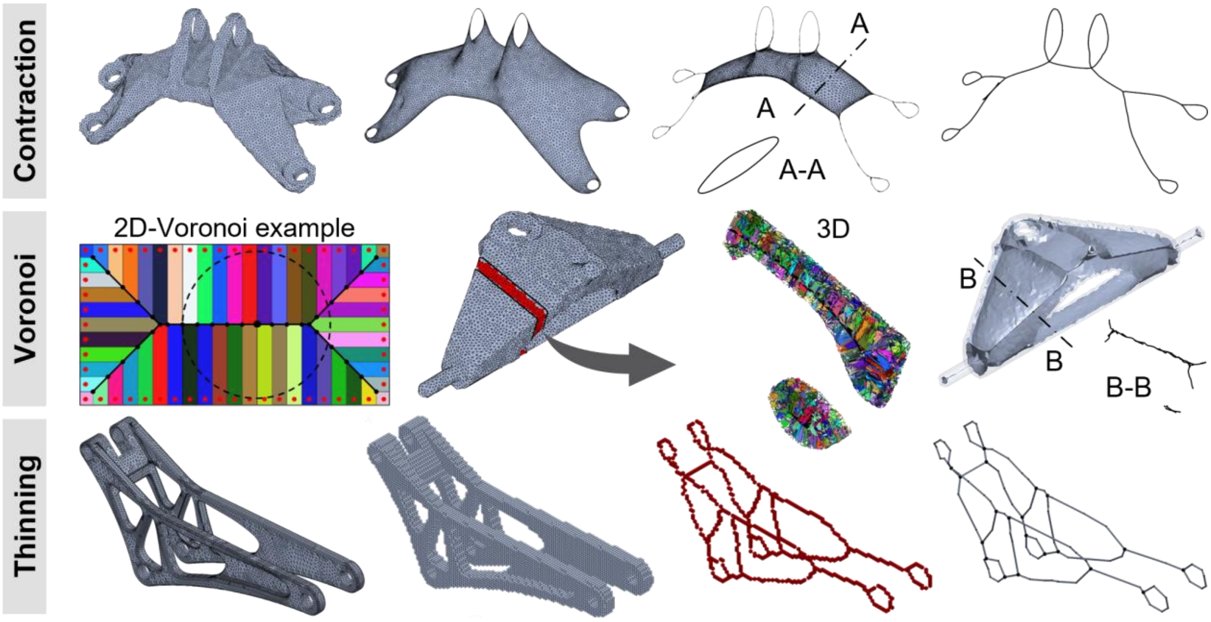

Figure 1: Comparison of the three main skeletonization methods [11–13]

the surface is iteratively smoothed and contracted, for example by Laplacian-based mesh deformation, while maintaining a shell-like crossection until only a one-dimensional structure remains [14]. A contraction-based curve skeleton extraction of a bracket is shown in Figure 1. These methods naturally generate smooth skeletons and are comparatively robust to noisy surfaces. However, their performance strongly depends on mesh quality, they may encounter difficulties in junction regions, and the resulting skeleton may extend beyond the original geometry. [7,13]

Voronoi-based methods derive the skeleton from the Voronoi diagram of the boundary samples. This principle can initially be understood in two dimensions, where sampled boundary points divide the surrounding area into Voronoi cells. The vertices and edges shared by adjacent cells are equidistant from multiple boundary points and therefore approximate the medial axis (see Figure 1). This concept naturally extends to three-dimensional geometries, where the resulting Voronoi diagram initially forms a branched medial surface which is typically smoothed and can subsequently be reduced to a curve skeleton [11]. The initial medial surface extraction based on a voronoi diagram is illustrated on a skateboard truck in Figure 1. Although these methods provide computationally efficient, well-centred skeletons, they encounter issues regarding segmentation within joint regions and are sensitive to irregular boundary sampling and surface noise. This often necessitates additional smoothing or post-processing [7].

Thinning methods follow the opposite strategy of contraction by iteratively removing boundary voxels while preserving the object's topology. Starting from a binary raster representation of the respective geometry, deletion rules based on neighborhood relationships and the evaluation of the Euler characteristic identify surface voxels as so-called *simple points*, whose removal neither disconnects the object nor alters its topology. The process continues until no more voxels can be deleted, resulting in a one-voxel-wide binary skeleton, which can subsequently be converted into a curve skeleton [15]. An example of a thinning-based skeleton is shown in Figure 1. Compared with contraction methods, thinning provides a deterministic and convergent algorithm and is well suited for fully automated processing pipelines. Additionally, thinning is generally less sensitive to surface noise than Voronoi-based approaches. However, the resulting skeleton quality depends on the voxel resolution and may not always remain perfectly centered. [7,16]

Once extracted, skeletons provide a compact structural representation that can be enriched with additional geometric information, such as local radii or cross-sectional properties, and subsequently converted into various parametric CAD representations including meta balls [17] constructive solid geometry (CSG) [10], convolutional surfaces [17], or freeform surfaces [18]. While skeletons are primarily employed to represent the geometry and topology of a structure, it could be beneficial to have an additional representation of application-specific information associated with the underlying geometry in many cases.

2.2. Feature-Mapping

Spatially distributed scalar, vector, and tensor fields are encountered in numerous scientific and engineering disciplines, including computational fluid dynamics [19], climate modeling [20], and structural mechanics [21]. Efficiently representing such high-dimensional data has therefore become an active field of research, particularly in the context of feature extraction, dimensionality reduction, and reduced-order modeling [22,23].

Existing approaches can be broadly classified into linear and nonlinear methods. Classical linear techniques, such as principal component analysis (PCA), project the original data onto a reduced basis. Nonlinear methods, including kernel PCA, isomap, locally linear embedding (LLE), Laplacian eigenmaps, and other manifold-learning algorithms preserve nonlinear relationships within the data [23]. More recently, deep-learning approaches based on autoencoders have emerged as an effective alternative for highly nonlinear problems [21,24,25]. These methods are primarily evaluated based on their ability to accurately reconstruct the data and retain its essential characteristics [22].

Besides data-driven approaches, analytical field functions have also been employed for dimensionality reduction. Within skeletonization, generalized potential and force fields have been used to extract curve skeletons instead of relying solely on distance fields, demonstrating that skeletal representations can be derived from arbitrary scalar or vector fields [9].

3. Research need

Current skeletonization methods primarily focus on the geometric and topological representation of objects. Application-specific physical information is generally not considered during skeleton extraction or representation [7]. In contrast, existing dimensionality reduction approaches efficiently compress high-dimensional simulation data but typically generate abstract latent representations without preserving an explicit geometric reference to the underlying structure [22,23,25]. Consequently, both research directions have largely evolved independently.

At the same time, the development of machine learning models for the rapid evaluation of design proposals is becoming increasingly difficult with increasing dimensionality and complexity of technical problems. Reducing such models to compact curve representations enriched with physically relevant information could provide suitable training data for a wide range of machine learning models, particularly for PINNs. This motivates the development of a methodology that combines skeletonization with the projection of finite element field data to generate compact, physically meaningful training data for machine learning models and leads to the following research question: *How can 3D-FE models of solids be automatically converted into backward-compatible 1D beam models while retaining physically relevant properties?*

4. Workflow for a reduction of 3D-FE models to 1D beam models

The methodology presented in this paper addresses a problem in structural mechanics, but is formulated generically, so it can be applied to a broad range of physics-based model reduction tasks. In structural mechanics, components undergo boundary conditions and external loads, which result in stress, strain, and displacement fields depending on both geometry and material properties.

Furthermore, the method is designed for three-dimensional problems, which are predominant in practical engineering applications. For clarity of presentation, however, several processing steps are illustrated using two-dimensional examples.

As shown in Figure 2, our proposed workflow for generating a reduced one-dimensional finite element model consists of two main stages: geometric skeletonization and feature projection. The first stage involves extracting the medial axis to generate a curve skeleton with local cross-sectional information, largely following the approach of Denk et al. [26]. The second stage projects nodal structural quantities, such as stresses, strains, and displacements, onto the generated skeleton while preserving as much physical information as possible.

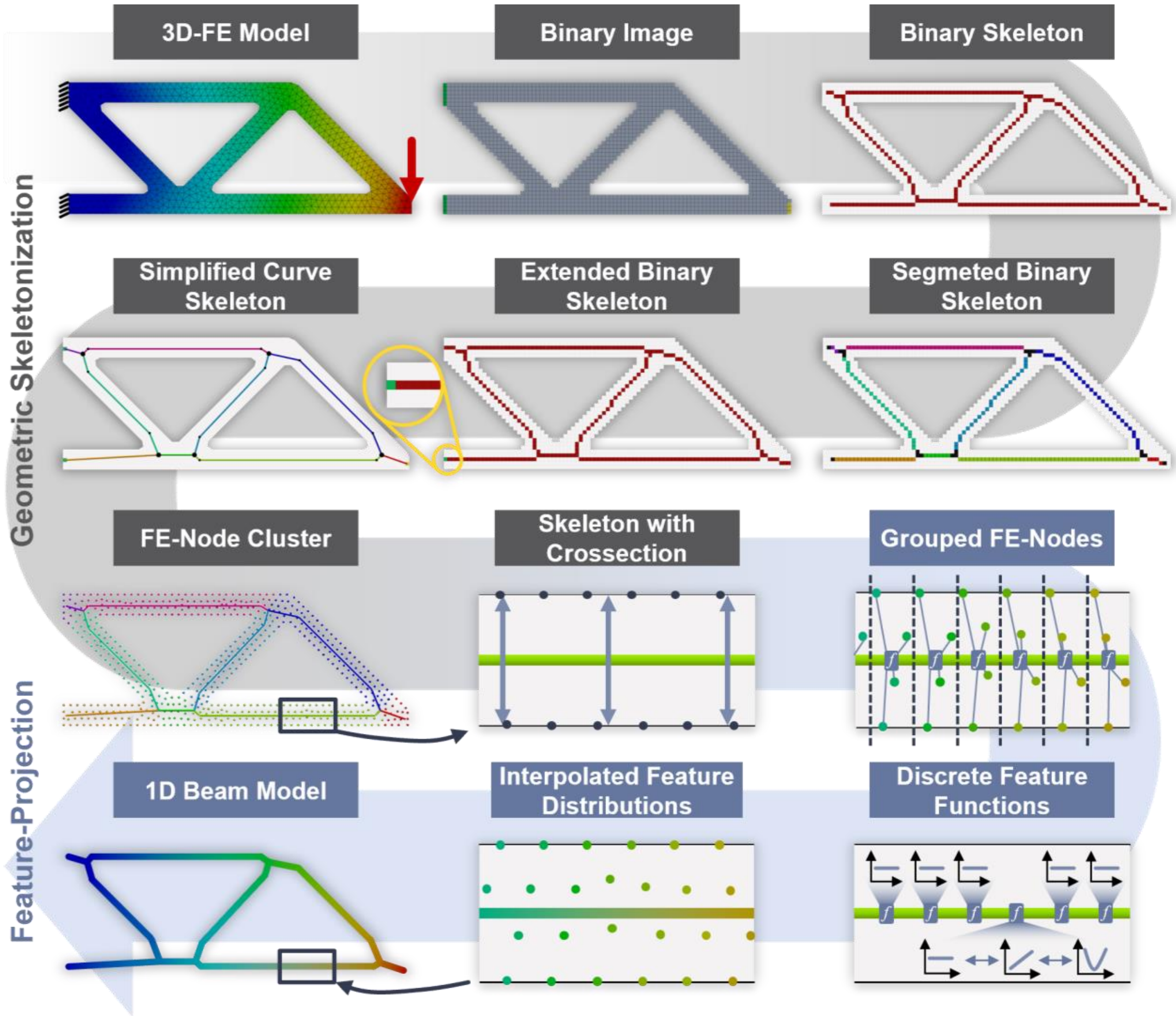

Figure 2: General Workflow for generating 1D beam representations out of 3D-FE simulation models

4.1. Geometric skeletonization

The entire workflow starts from a 3D-FE model, which in our case is obtained through an static mechanical simulation using Ansys Workbench software. Based on this, we generate a volume-based output of the FE mesh and export a point cloud of all FE nodes with node-by-node assignment of boundary conditions, loads, and information on the resulting structural mechanical behavior values, such as node stress, displacement, and strain.

Among the three primary methods of medial axis derivation as outlined in section 2.2, we opted for thinning. Due to its properties as a deterministically converging and fully automated algorithm, it appears to be the most suitable one for the intended application of generating datasets for machine learning (ML) models. However, it should be noted that this method necessitates the preliminary generation of a binary raster image. The thinning algorithm then generates a binary skeleton by evaluating the Euler characteristics and checking the connectivity of the neighborhood. [15]

The binary format of the resulting skeleton allows the subsequent segmentation by evaluating the remaining neighborhoods to identify endpoints and joint points. Each of these separates individual beam segments. At this point, pruning is usually applied to remove unnecessary branches from the skeleton [16]. However, since this rarely occurs with the beam-like structures primarily considered here, this intermediate step can usually be omitted in our case. Due to Blum's MAT definition, the classical thinning algorithm always results in a gap between the skeleton endpoint and the beam ends. Depending on the shape of a beam ending, this is not advantageous – on the one hand, for the correct representation of the boundary conditions, and on the other hand, for subsequent feature projection [26]. To address this, we introduce three different cases that determine whether the binary skeleton must be extended in the beam direction (see Figure 3). The distinction between these cases is based on the distribution of the distances from the skeleton endpoint to the surface of the beam ending.

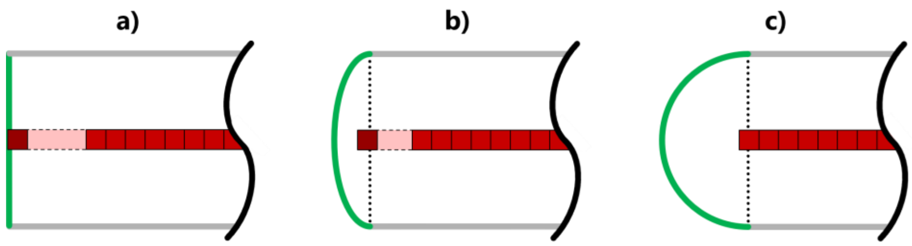

Figure 3: Skeleton endpoint extension for different beam-end configurations; a) angular b) mixed c) circular

In order to convert the extended and segmented binary skeleton into a suitable format for subsequent geometric evaluation and for the projection of physical behavior, it is converted into a curve skeleton in the next step. To this end, a method proposed by Denk et al. is used to approximate the path of the binary skeleton using polylines based on a collapse metric ϵ [17]. This ensures that a point $\boldsymbol{v}_i^k$ and the following point $\boldsymbol{v}_i^{t+k}$ along the polyline i is in a certain quadratic distance ϵ.

$$\left|\left|\boldsymbol{v}_i^{t+k} - \boldsymbol{v}_i^k\right|\right| \leq \epsilon$$

The result of this simplification is subdivision of each previously identified segments into multiple polylines. In the subsequent step, all FE nodes contained within the point cloud are assigned to the polyline that is spatially nearest. From the next step onward, each following operation in the workflow is executed independently for each individual beam line. The determination of the beam cross-section via the Euclidean distance from the medial axis to the edge points now provides a smooth transition to feature projection.

4.2. Feature projection

To the best of our knowledge, feature projection, as presented in the following, hast not been reported in this context and therefore represents the key novel aspect of the proposed method. The initial state is comosed of the skeleton represented by polylines and containing information on beam cross-sections. In order to map physical properties, the points assigned to each polyline via FE node clustering are first grouped. Therefore, we developed three different variants (see Figure 4).

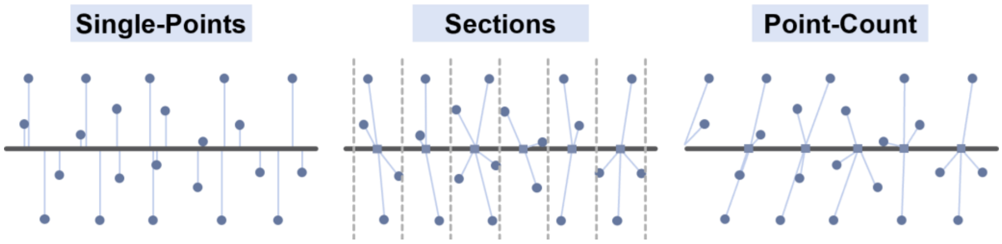

Figure 4: FE-node grouping methods

The first one intends a direct projection of the physical properties of all FE nodes separatly as single points onto the curve skeleton. This is the simplest method, which does not require any additional functions afterward. However, this direct projection of the property values onto the polyline can result in significant discontinuities in the property profiles. For this reason, this method is only suitable for properties that do not exhibit large gradients across the beam cross-section. The second variant involves grouping FE nodes that lie within defined sections. This ensures that a consistent assignment can be made across the entire beam span, regardless of the resolution of the FE mesh. In contrast, the third variant specifies a defined number of FE nodes for the point count, which is used to consistently form groups with the same number of FE nodes along the entire length of the beam. While this method can reduce the field data with higher resolution during FE mesh refinements in specific beam regions, it has a significant disadvantage compared to grouping by sections due to the uneven distribution of points within the cross-section.

To address the challenge of mapping physical field data that is unevenly distributed over the cross-section, the property fields must be transfered into feature functions for each group when using sections or point counts. This step is first illustrated in Figure 5 for the two-dimensional case with properties distributed uniformly (u_y) and non-uniformly (u_x) over the cross-section, and is also demonstrated in Figure 6 for an identical three-dimensional case. Of central importance here is the linear mapping of the point distribution from an auxiliary beam coordinate system K_b to a coordinate system K_F specifically designed for 2D or 3D feature functions. The resulting feature functions provide discrete distribution information for the desired physical field, as demonstrated in Figure 4 by the quadratic objects for "Sections" and "Point Count." In order to ultimately obtain continuous distribution fields – just as in the calculation of 2D- or 3D-FE models – these feature functions, which are discretely distributed along the polyline, must be interpolated. The combination of the continuous function curves for each beam segment into a feature-enriched polyline with cross-sectional information is the final step in the process that ultimately results in the desired 1D model.

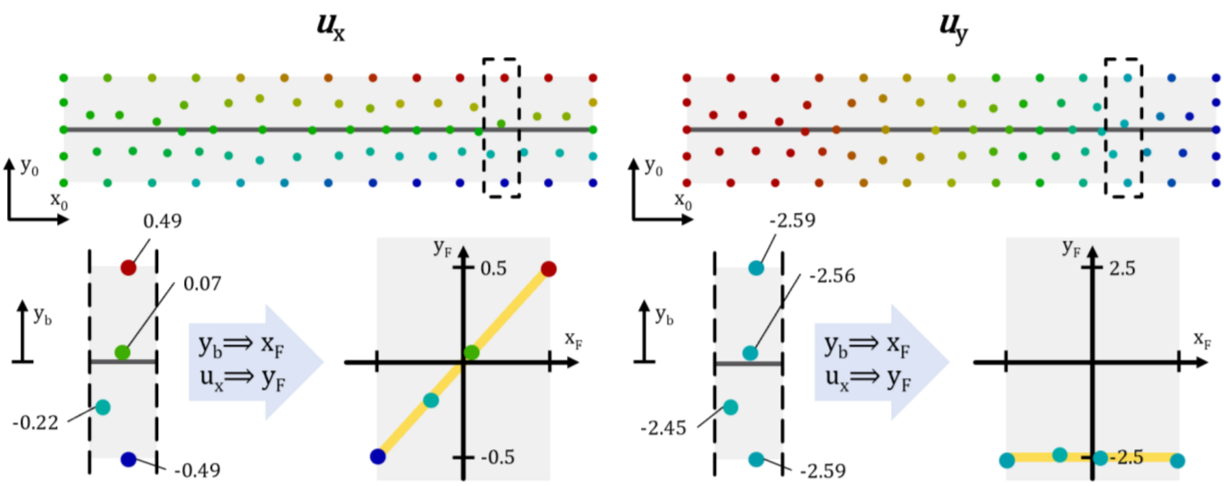

Figure 5: Generation and analysis of feature functions for 2D-FE-point clouds over the beam cross-section

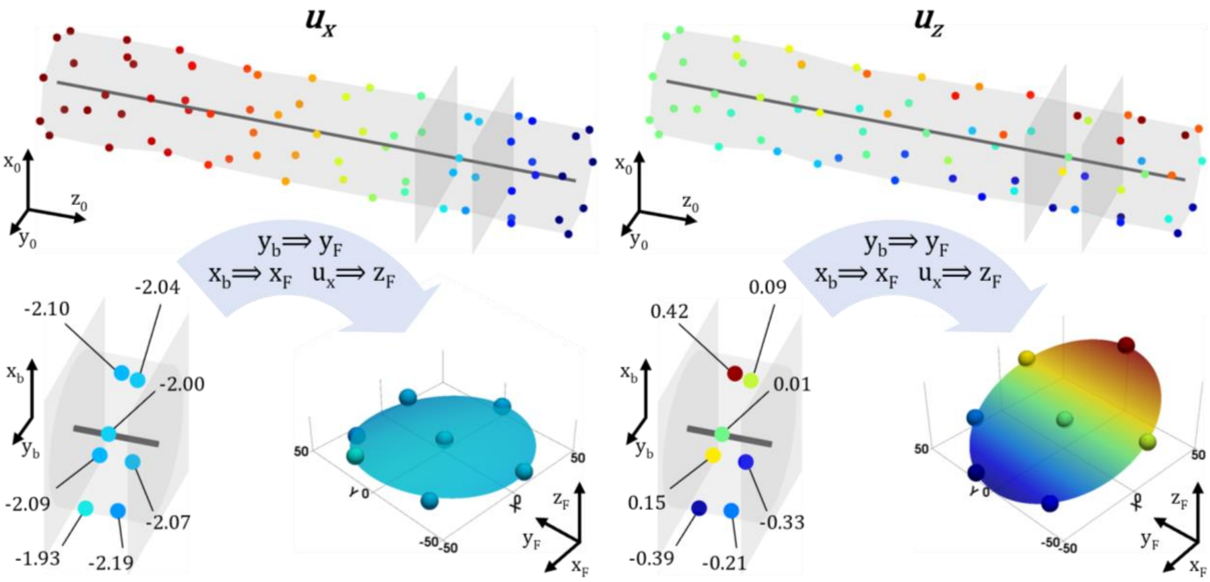

Figure 6: Generation and analysis of feature functions for 3D-FE-point clouds over the beam cross-section

5. Evaluation of the reduction approach

For qualitative evalutation, we tested our workflow using several 2D and 3D datasets. Figure 7 shows a section highlighting particularly relevant steps in the reduction of a 3D bicycle frame to a 1D beam representation with a maximum model length of 300 voxels and a collapse metric of 2.0. A visual comparison of the generated 1D model with the original 3D model looks promising and demonstrates, in particular, the applicability of the method for 3D models. Even when applying the method to various other problems, it is evident that our method generally creates meaningful 1D models.

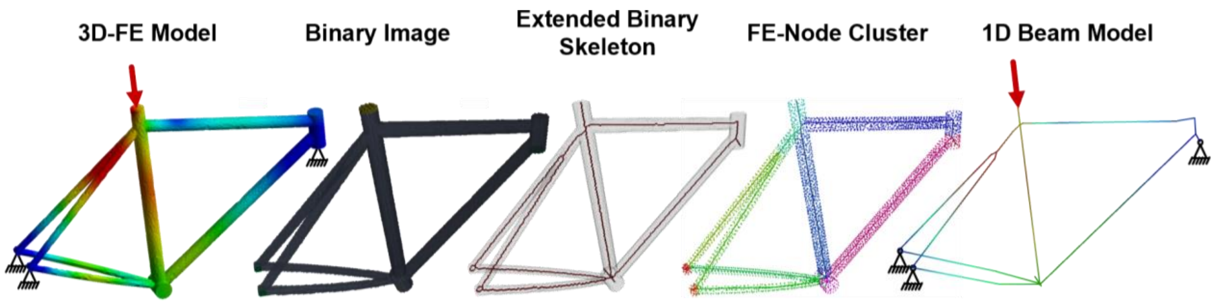

Figure 7: Excerpt of the 3D-FE reduction of a bicycle frame to a 1D representation, based on the total deformation

To conduct an initial quantitative evaluation of the results, we also performed a reduction of a rigidly fixed, single-loaded cantilever with a circular cross-section and compared the results of the 1D model at certain points with the node displacements in the 3D-FE model and the analytical solution. In doing so, we observed a relative deviation in vertical displacement at three selected positions of approximately 7.9% compared to the initial 3D model and 5.5% compared to the analytical solution. As initial quantitative results, these values can certainly be considered positive. However, for the generation of a meaningful and physically consistent training dataset for an ML model, even smaller deviations are definitely desirable.

In general, we observe that the quality of the results depends heavily on the assumptions we make regarding predefined parameters such as the resolution of the rasterization, the collapse metric value, and the chosen grouping method for feature projection. In doing so, we are constantly faced with a trade-off between the quality of the results on the one hand and memory consumption and computation time on the other. Hence, the reduction process inevitably results in a loss of information. A quantitative evaluation of this has not yet been performed. However, by preserving the cross-sectional information together with the spatial distribution of the physical properties, the proposed 1D representation format establishes the necessary prerequisites for such a backward transformation.

6. Conclusion and outlook

The workflow presented here enables the automatic derivation of 1D models using an algorithm comprising twelve steps, primarily out of frame- and beam-like 3D-FE models. This approach provides a solution to the research question. In the process, the well-established method of geometric skeletonization is extended by a novel process of feature projection for physical property fields onto the skeleton. The implementation of feature functions for the storage of physical property fields enables the mapping of field quantities and the application of this method to a variety of physical and scientific problems.

To finally generate training data with the presented method, a control mechanism for evaluating the quality of the model reduction is required to ensure physical consistency. This also entails with a suitable method for backward transformation on 3D representations in order to quantify the information loss. Furthermore, the computational cost and memory consumption must be reduced to enable higher resolution and improved result quality. Finally, future work will investigate skeletonization directly on arbitrary FE meshes to avoid rasterization and the partially non-centered skeletons introduced by binary voxel representations.

Acknowledgement

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 523871886. The authors gratefully acknowledge the financial support. The authors also thank Marie Schmid for her preliminary work on this topic.

References

[1] Tanveer, Mohad et al.: Generative Design for Engineering Applications: A State-of-the-Art Review. In: Archives of Computational Methods in Engineering vol. 33 (2026). DOI: 10.1007/s11831-025-10302-y.

[2] Shin, Seungyeon; Shin, Dongju; Kang, Namwoo: Topology optimization via machine learning and deep learning: a review. In: Journal of Computational Design and Engineering vol. 10 (2023), no. 4, pp. 1736–1766. DOI: 10.1093/jcde/qwad072.

[3] Raissi, Maziar; Perdikaris, Paris; Karniadakis, George Em: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. In: Journal of Computational Physics vol. 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[4] Karniadakis, George Em et al.: Physics-informed machine learning. In: Nature Reviews Physics vol. 3 (2021), pp. 422–440.

[5] Deckert, Felix et al.: Exploring the Potential of Physics-Informed Neural Networks for the Structural Analysis of 2D Frame Structures. In: Applied Mechanics vol. 6 (2025), no. 4, p. 84. DOI: 10.3390/applmech6040084.

[6] Cook, Robert D. et al. (eds.): Concepts and applications of finite element analysis. 4. ed. New York, NY: Wiley, 2002. ISBN: 978-0-471-35605-9.

[7] Tagliasacchi, Andrea et al.: 3D Skeletons: A State-of-the-Art Report. In: Computer Graphics Forum vol. 35 (2016), no. 2, pp. 573–597. DOI: 10.1111/cgf.12865.

[8] Blum, H.: A Transformation for Extracting New Descriptors of Shapev. In: Models for the Perception of Speech and Visual Form: M.I.T. Press, 1967.

[9] Cornea, Nicu D.; Silver, Deborah; Min, Patrick: Curve-Skeleton Properties, Applications, and Algorithms. In: IEEE Transactions on Visualization and Computer Graphics vol. 13 (2007), no. 3, pp. 530–548. DOI: 10.1109/TVCG.2007.1002.

[10] Nana, Alexandre; Cuillière, Jean-Christophe; Francois, Vincent: Automatic reconstruction of beam structures from 3D topology optimization results. In: Computers & Structures vol. 189 (2017), pp. 62–82. DOI: 10.1016/j.compstruc.2017.04.018.

[11] Mayer, Johannes; Denk, Martin; Wartzack, Sandro: Reconstruction of Topology Optimized Geometry with Casting Constraints in a Feature-Based Approach. In: Proceedings of the Design Society vol. 3 (2023), pp. 3015–3024. DOI: 10.1017/pds.2023.302.

[12] Mayer, Johannes; Wartzack, Sandro: Ermittlung eines Skelettierungsverfahrens zur Konvertierung von Topologieoptimierungsergebnissen. In: Proceedings of the 31st Symposium Design for X (DFX2020): The Design Society, 2020, pp. 111–120. DOI: 10.35199/dfx2020.12.

[13] Mayer, Johannes; Wartzack, Sandro: Computational Geometry Reconstruction from 3D Topology Optimization Results: A New Parametric Approach by the Medial Axis. In: Computer-Aided Design and Applications (2023), pp. 960–975. DOI: 10.14733/cadaps.2023.960-975.

[14] Au, Oscar Kin-Chung et al.: Skeleton extraction by mesh contraction. In: ACM Transactions on Graphics vol. 27 (2008), no. 3, pp. 1–10. DOI: 10.1145/1360612.1360643.

[15] Lee, Ta-Chih; Kashyap, Kashyap; Chu, Chong-Nam: Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms. In: CVGIP: Graphical Models and Image Processing vol. 56 (1994), no. 6, pp. 462–478. DOI: 10.1006/cgip.1994.1042.

[16] Sobiecki, André; Jalba, Andrei; Telea, Alexandru: Comparison of curve and surface skeletonization methods for voxel shapes. In: Pattern Recognition Letters vol. 47 (2014), pp. 147–156. DOI: 10.1016/j.patrec.2014.01.012.

[17] Denk, M.; Rother, K.; Paetzold, K.: Truss Parametrization of Topology Optimization Results with Curve Skeletons and Meta Balls. In: Proceedings of the Design Society vol. 2 (2022), pp. 363–372. DOI: 10.1017/pds.2022.38.

[18] Amroune, Abdennour; Cuillière, Jean-Christophe; François, Vincent: Automated Lofting-Based Reconstruction of CAD Models from 3D Topology Optimization Results. In: Computer-Aided Design vol. 145 (2022), p. 103183. DOI: 10.1016/j.cad.2021.103183.

[19] Omata, Noriyasu; Shirayama, Susumu: A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder. In: AIP Advances vol. 9 (2019), no. 1, p. 015006. DOI: 10.1063/1.5067313.

[20] Falasca, Fabrizio; Perezhogin, Pavel; Zanna, Laure: Data-driven dimensionality reduction and causal inference for spatiotemporal climate fields. In: Physical Review E vol. 109 (2024), no. 4, p. 044202. DOI: 10.1103/PhysRevE.109.044202.

[21] Shinde, Krushna et al.: Dimensionality reduction through convolutional autoencoders for fracture patterns prediction. In: Applied Mathematical Modelling vol. 114 (2023), pp. 94–113. DOI: 10.1016/j.apm.2022.09.034.

[22] Hou, Chun Kit Jeffery; Behdinan, Kamran: Dimensionality Reduction in Surrogate Modeling: A Review of Combined Methods. In: Data Science and Engineering vol. 7 (2022), no. 4, pp. 402–427. DOI: 10.1007/s41019-022-00193-5.

[23] Jia, Weikuan et al.: Feature dimensionality reduction: a review. In: Complex & Intelligent Systems vol. 8 (2022), no. 3, pp. 2663–2693. DOI: 10.1007/s40747-021-00637-x.

[24] Csala, Hunor; Dawson, Scott T. M.; Arzani, Amirhossein: Comparing different nonlinear dimensionality reduction techniques for data-driven unsteady fluid flow modeling. In: Physics of Fluids vol. 34 (2022), no. 11, p. 117119. DOI: 10.1063/5.0127284.

[25] Niu, Hongqi et al.: Exploring unsupervised feature extraction algorithms: tackling high dimensionality in small datasets. In: Scientific Reports vol. 15 (2025), no. 1, p. 21973. DOI: 10.1038/s41598-025-07725-9.

[26] Denk, Martin; Rother, Klemens; Paetzold, Kristin: Fully Automated Subdivision Surface Parametrization for Topology Optimized Structures and Frame Structures Using Euclidean Distance Transformation and Homotopic Thinning. In: Pfingstl, Simon et al. (eds.): Proceedings of the Munich Symposium on Lightweight Design 2020. Berlin, Heidelberg: Springer, 2020, pp. 18–27. DOI: 10.1007/978-3-662-63143-0_2.