Materials & Design 261 (2026) 115281

Contents lists available at ScienceDirect

Materials & Design

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

Programming tension in 3D-printed networks inspired by spiderwebs

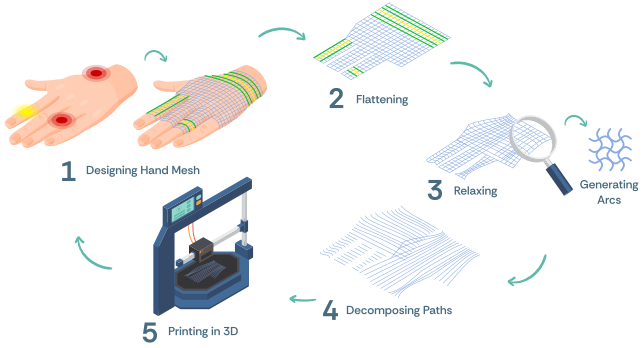
T.C.P. Masmeijer[a], C.C. Swain[b], J.R. Hill[b], E. Habtour[a,*]

[a] Aeronautics & Astronautics, The University of Washington, Seattle, WA 98195, USA
[b] Department of Mechanical Engineering, Brigham Young University, Provo, UT 84604, USA

HIGHLIGHTS

- A methodology that enables the direct 3D printing of flat, curved, and 3D structural networks with self-programmed tension gradients.
- The method accounts for the mechanistic interplay between geometric and material properties throughout fabrication and assembly.
- The required shape, stability, and compliance design objectives of a network can be realized and maintained automatically upon assembly, eliminating the need for undesirable force calibration of each structural element.
- A self-programmed tension gradient methodology for direct 3D printing of flat, curved, or 3D structural network.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Tensioned structures
Tensegrity
Programmed tension
Medical braces
Bioinspired

ABSTRACT

Each element in tensioned structural networks—e.g., tensegrity or architectural fabrics—requires a specific tension to achieve the desired shape and stability. These structures are challenging to manufacture, 3D print, or assemble because flattening them during fabrication introduces multiplicative inaccuracies in the final tension gradients. We overcome this challenge by offering an algorithm for direct 3D printing of such networks with programmed tension gradients, analogous to the spinning of spiderwebs.

The algorithm: (i) defines the desired network and prescribes its tension gradients; (ii) converts it into an unstretched counterpart by optimizing element lengths and converting straight elements into arcs; and (iii) decomposes the network into printable toolpaths; with the option to: (iv) flatten curved 2/3D networks to ensure printing compatibility; and (v) automatically resolve unwanted crossings introduced by flattening. Experimental validation is achieved using 2D unit cells with <1.0 % strain error in the tension gradients. The method remains effective for networks with a minimum element length of 5.8 mm and a maximum stress of 7.3 MPa. Fabricating complex cases is demonstrated for flat spiderweb, curved mesh, and tensegrity networks. The method represents a stepping-stone toward developing compact, integrated cable networks and orthotic devices with programmable moment-exerting structures.

* Corresponding author.
Email address: habtour@uw.edu (E. Habtour).

https://doi.org/10.1016/j.matdes.2025.115281
Received 10 September 2025; Received in revised form 3 December 2025; Accepted 4 December 2025
Available online 11 December 2025

1. Introduction

In any tensioned membrane structural system, precise tension levels between individual elements are critical to achieving and maintaining the desired stability, compliance, and geometry. While historically applied in civil engineering contexts such as lightweight catenary systems and tensioned fabric roofs [1–3], these systems now demonstrate significant potential across emerging domains. For example, recent advances include assistive wearables for medical rehabilitation [4], enhanced actuation in soft robots [5–7], tensegrity networks for lightweight robots [8–11], structural health monitoring [12], and adaptive systems for space debris removal [13–16]. In all applications, cable tension distribution is fundamental, not only for maintaining structural integrity but also for defining dynamic behaviors, such as natural frequencies and deflection modes [17,18]. Nevertheless, achieving a precise tension distribution in such structures remains a persistent challenge due to two main engineering limitations: (1) tension often only emerges after the application of external loads, rather than being inherent in the initial fabrication, and (2) conventional fabrication methods are inefficient and may fail to produce consistent or optimal prestress states [19–21]. These limitations are especially pronounced in miniaturized or thin systems, where minor deviations in prestress can cause unintended substantial variations in the mechanical performance. Inspired by the tensioning process in spiderwebs, the authors developed and experimentally validated an algorithm to engineer self-tensioning elements within 3D-printed membrane/mesh networks that overcome these limitations.

1.1. Challenges and recent advancements

Despite advances in optimization techniques for defining the tension of structural elements, realizing the intended tension state during manufacturing remains technically difficult [23]. This is because construction often requires jigs, multiple people, or time-consuming assembly of individual components. Assembly solutions can limit passive cable control or require active cable control [24–26]. Additive manufacturing has offered a promising route to fabricating small, prestressed structures and has been used to produce tensegrity lattices with tunable band gaps and dynamic properties [27,28]. Parallel developments in 3D knitting have further enabled localized tailoring of mechanical responses by varying material composition and patterning [29,30]. While notable progress has been made in 3D printing and 3D knitting, both approaches exhibit inherent limitations. In current fully automated 3D printed tensegrities, designed tension states emerge only after an external load is applied [27,28,31]. These automated methods aim to overcome the labor-intensive nature of piecewise fabrication, where cables and bars are assembled separately. However, the as-printed structures are typically unstressed. In contrast, 3D knitting allows fully automated fabrication of complex, deformable structures, but requires more costly equipment than widely available 3D printers. Several efforts have explored 3D printing prestressed structures using Fused Filament Fabrication (FFF), including strategies that involve post-processing to remove sacrificial molds [17,31,32]. Still, a scalable method to directly manufacture tension-programmed networks remains a persistent challenge.

1.2. Why spiderwebs-inspiration?

Spiderwebs are natural networks designed for tunable tension gradients. For instance, the typical tension ratio in an orb web between its anchor threads, frame threads, and radii is approximately 10:7:1 [33], see Fig. 2. Spiders build these structures in a matter of hours by sequentially laying down continuous elements such as bridging threads, primary/secondary frames, radii, sticky spirals, and a central hub [34,35]. This remarkable fabrication approach of a tensioned structure is created seamlessly, with the spider constantly maintaining the desired tensions as it builds. While spiderwebs are built strand by strand under tension, the networks we propose are fabricated in an unstressed state and only take on their designed tensioned shape upon application. Nonetheless, they are similarly constructed as a continuous path—making them well-suited for extrusion-based 3D printing.

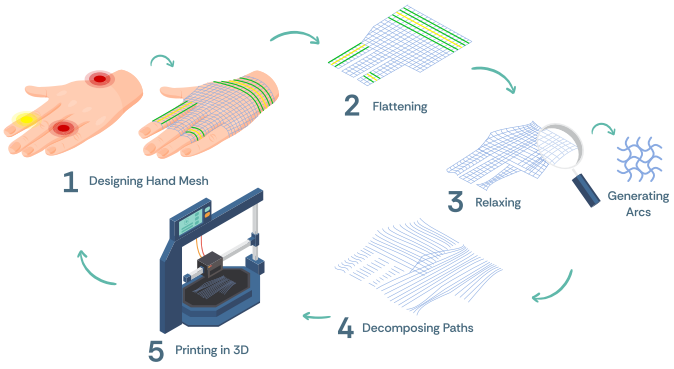


Fig. 1. Conceptual illustration of how a 3D printed tensioned network with programmable tension gradients could be applied to a custom, patient-specific compression cast. The steps to automatically manufacture them consist of 1) designing a network with a tension gradient; 2) flattening it; 3) finding the network's unstretched counterpart that will result in the target design and tension gradient when the patient wears the compression cast; 4) decomposing the cast into continuous paths; and 5) 3D printing the structure. The full implementation is openly available on GitHub [22].

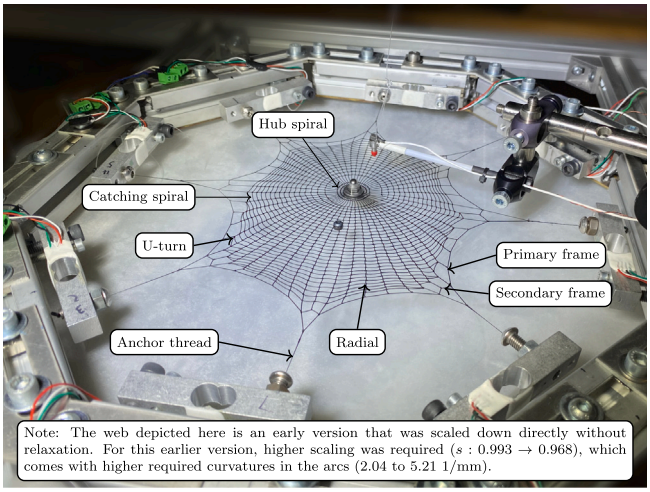


Fig. 2. A 3D-printed spiderweb-like structure suspended in a frame. The web incorporates design features commonly observed in spider orb webs. These features are annotated in the figure according to the nomenclature reported by Zschokke et al. [36].

1.3. Approach

In this study, we presented a method for manufacturing network structures with programmable tension gradients using accessible 3D printing, thereby overcoming the limitations mentioned above. The approach is also explained graphically in Fig. 1. Desired tension networks, for example, were designed using the Force Density Method (FDM). Although 3D printing a network in its equilibrium shape produced the correct geometry, it would not achieve the intended tension distribution. Therefore, the network was converted into an unstressed counterpart that attained the desired shape and tension after assembly. To manufacture systems similar to orb spider webs, the network was decomposed into continuous printable paths. The approach was not limited to planar networks such as spider webs; procedures were also provided to flatten curved 2D (2.5D) or fully 3D structures. The novelty of our method is

reported in this study, which is fundamentally different from 3D knitting, as programmable deformation was achieved by tuning the tension in individual edges rather than by adjusting local material properties.

1.4. Impact

Our tension programmable method for structural networks opens the door to impactful, real-world applications. One of the clearest opportunities lies in the design of compression casts or splints. Traditional materials, such as plaster and fiberglass, are rigid and non-adjustable, often resulting in excessive pressure that requires valving [37] or complete recasting [38]. In contrast, our approach enables compression to be actively and locally tuned by embedding programmable tension gradients during fabrication. When combined with a ratcheting system or control cable, it becomes possible to design splints that adapt to the patient over time—improving comfort and reducing the need for medical intervention. Because our method is compatible with low-cost 3D printers, these devices can be manufactured locally, in homes or community spaces, enabling affordable and patient-specific care.

A second application area is wearable robotics. Devices like cable-driven exosuits (e.g., CAREX [39] and CRUX [4]) depend on tensioned elements to transmit forces and assist movement. Our method directly encodes these elements with programmable stiffness and directional force transfer, thereby reducing complexity and enabling more compact, lightweight designs. The ability to print such structures without bulky knitting machines further expands access to research, prototyping, and personalized solutions in rehabilitation and assistive technologies.

Stress relaxation is a known limitation in 3D-printed polymers and tensioned elements, leading over time to gradual force decay and loss of prestress. While our method cannot eliminate stress relaxation, this effect is inevitable for most polymers, and its rate and magnitude can be managed through design—by over-tensioning, embedding reinforcement, or scheduling periodic re-tensioning, as is standard practice in both medical and robotic applications [40]. For most short-term applications (e.g., wearable orthoses, temporary supports, rapid prototyping), stress relaxation does not undermine initial utility, and long-term utility can often be maintained with simple manual adjustments. Similarly, under cyclic loading, 3D printed TPU quickly reaches a steady-state stress–strain response after initial degradation. While this work used pristine material properties, future designs could adopt these steady-state values to better represent long-term behavior [41].

Additionally, our method can be applied to deployable space systems [42] in which tensioned cables are used to support the deployment process itself, but are not required to maintain the structure under operational loads. Examples include deployable booms, reflectors, or antennas where cable tension ensures reliable transformation to the final shape, after which the main structure is rigidized or locked, and the cables may relax or even be removed. This capability reduces mass and complexity compared to traditional assembly, broadening the utility of programmable tension networks in aerospace and field-deployable systems.

Finally, this methodology is particularly well-suited for rapid prototyping of tensioned cable systems. Even when a different manufacturing or assembly method is ultimately chosen for the final product, the ability to quickly and accurately realize programmable prestress and directional force transfer in a single printed network enables rapid iteration on routing, stiffness gradients, and deployment strategies before committing to costlier processes. This reduces design risk and accelerates translation for applications ranging from medical wraps to deployable mechanisms, where early-stage verification of tension fields and load paths is a primary bottleneck.

This article is structured as follows. In Section 2, the applicability of the method to three cases is detailed; namely, a 2D spiderweb-inspired network (Section 2.1), a 2.5D moment-exerting mesh (Section 2.2), and a 3D tensegrity system (Section 2.3). In Section 3, the method's steps for designing, processing, and manufacturing network structures are described. A Python code accompanying this paper is publicly available to reproduce the results and test its applicability to other networks [22]. The method's experimental validation and limitations are described in Section 4. The results and conclusions are provided in Section 5.

2. Results

In this section, we demonstrate three cases of structural networks manufactured using the proposed method. Case-1 was an orb spider web to illustrate nature-tensioning in a 2D network. Case-2 was a moment-exerting mesh for a medical arm compression, which was a 2.5D network. Finally, Case-3 was a 3D tensegrity network to demonstrate the applicability of our method to a complex structural system.

2.1. Case-1: spider web

A spider web was the source of our inspiration due to its ingenious tension-gradient approach to realizing its remarkable, highly specialized structural network. Using the proposed method, spiderweb-like structures were fabricated with controlled tension distributions that closely mimic those of real webs. The structure presented here follows average design variability as measured by Vollrath et al. and Rhisiart et al. [43,44], with modeled tension gradients based on measurements by Wirth et al. [33]. The designed orb web with tension gradient and its unstretched printable counterpart are depicted in Fig. 3.

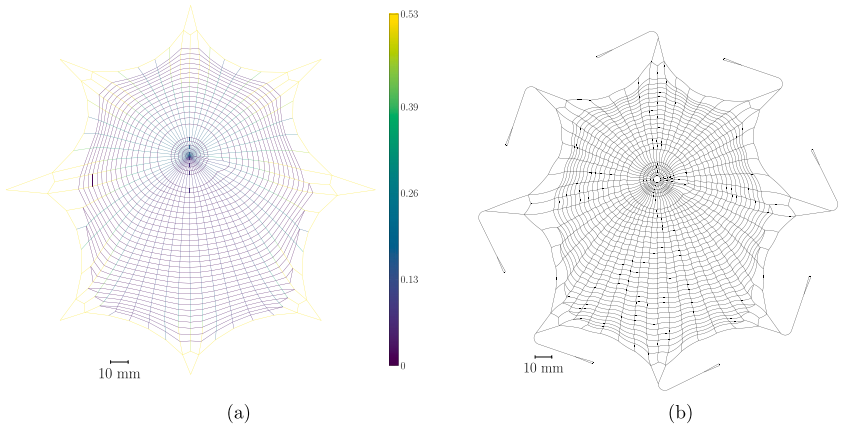


Fig. 3. a) A spiderweb-like network is designed using the Force Density Method. The color of the edges indicates the tension in Newton. b) An unstretched counterpart of the web with arcs before 3D printing. Post-processing involves adding eight connection loops and manually modifying a hole at the center to prevent material aggregation.

The target geometry was initially generated using the force density method, detailed in Section 3.1. Converting the target geometry into an unstretched counterpart involved three steps: relaxing the vertices, scaling down the web, and converting edges into arcs with target arc lengths. Details are provided in Section 3. The final network was decomposed into continuous paths that mimic the building steps of real spider webs: a central hub spiral, a catching spiral with a U-turn, a primary frame, eight secondary frame sections, and eight anchor threads. Descriptions of the web sections are illustrated in Fig. 2. Methods for decomposing networks into continuous paths are detailed in Section 3.3.1. The 3D-printed result, suspended in a customized frame, is shown in Fig. 2. Optimization convergence was achieved after 1852 iterations with a final error, ϵ_{1852}, of approximately 0.153 (using a damping parameter $\beta = 0.1$ and an optimization convergence tolerance $\tau = 10^{-6}$). The web was scaled by a factor $s = 0.993$; the underlying rationale is described in Section 3.2. The web was fabricated in 14 min.

The successful printing of this web-like structure demonstrated that complex, highly connected networks could be constructed directly and automatically using our method. The capability to integrate tension gradients into spider web-like structures distinguished our approach from experimental studies on artificial spider webs [45,46], which focused on geometry and material properties but did not account for appropriate tensions. Quantitative validation was performed using simple unit cells (Section 4).

2.2. Case-2: moment-exerting mesh

Tension gradients in a network can be designed to deform and exert loads in desirable ways. To demonstrate this, a moment-exerting mesh of a thin-walled cylinder was presented. The surface of the thin-walled cylinder under bending experiences stress, which can be described by

$$\sigma_{\text{bending}} = \frac{M R \cos\theta}{I}, \tag{1}$$

where M is the bending moment, R the radius, θ the circumferential angle, and I is the second moment of area.

To replicate this stress distribution, the cylinder was unwrapped into a flat sheet and discretized using $n_x = 35$ vertical and $n_y = 32$ horizontal members, as shown in Fig. 4, ensuring that all edge lengths exceed those used in the validation study. In a post-processing step, the left and right sides were connected using a nylon wire that alternately passed through the gaps on both sides and was secured with a knot. The force density of the vertical edges was varied as a function of θ using

$$q(\theta) = q^0 + \Delta q \sin(\theta), \qquad q_i = q(\theta_i), \qquad \theta_i = \frac{2\pi i}{n_x}, \quad i = 0, \dots, n_x - 1. \tag{2}$$

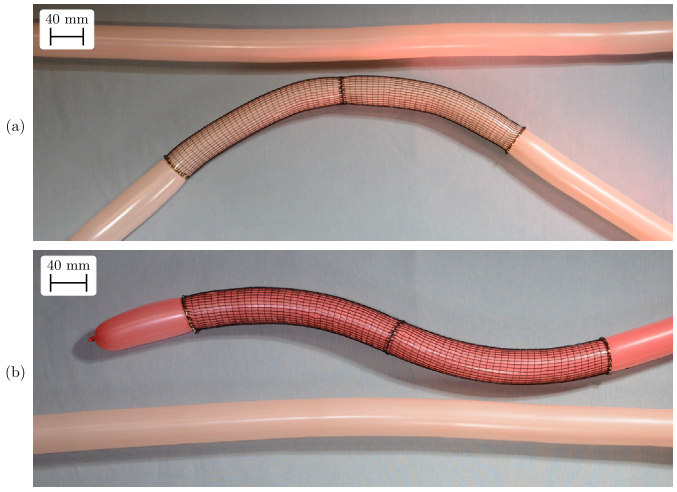


Fig. 5. Moment exerting wraps on balloons next to unloaded balloons. (a) Two wraps in the same orientation, resulting in a U-shape. (b) Two wraps in opposing directions, resulting in an S-shape.

with $q^0 = 0.16$ N/mm, and $\Delta q = 0.076$ N/mm. Horizontal elements were assigned a constant force density of 0.035 N/mm. Boundary vertices were fixed. The resulting network is shown in Fig. 4(a). To ensure sufficient stiffness, vertical members were printed with three layers of filament. The designed mesh was relaxed, scaled down, and the edges were turned into arcs, as detailed in Section 3.2. The optimized shape is shown in Fig. 4(c). The mesh was relaxed using a damping $\beta = 0.7$ and convergence tolerance $\tau = 10^{-6}$. After 1703 iterations, the shape converged with an error of $\epsilon_{1703} = 0.017$, and the mesh was scaled by a factor of $s = 0.996$. Fabrication of the mesh was completed in 26 min.

To enable physical testing, loops were added at the top and bottom of each vertical strand, allowing a metal ring to pass through. When applied to a balloon, the wrap produced a directional moment: two wraps in the same direction generated a U-shape, while opposing directions yielded an S-shape (see Fig. 5).

This approach demonstrates how stress gradients can be embedded into a 3D printed network to control out-of-plane deformation. Parallel developments in 3D knitting have shown that local patterning can tune the mechanical response in soft robots and assistive gloves [29,30]. Our method achieves a similar outcome, but through force-based optimization and filament deposition, offering a distinct

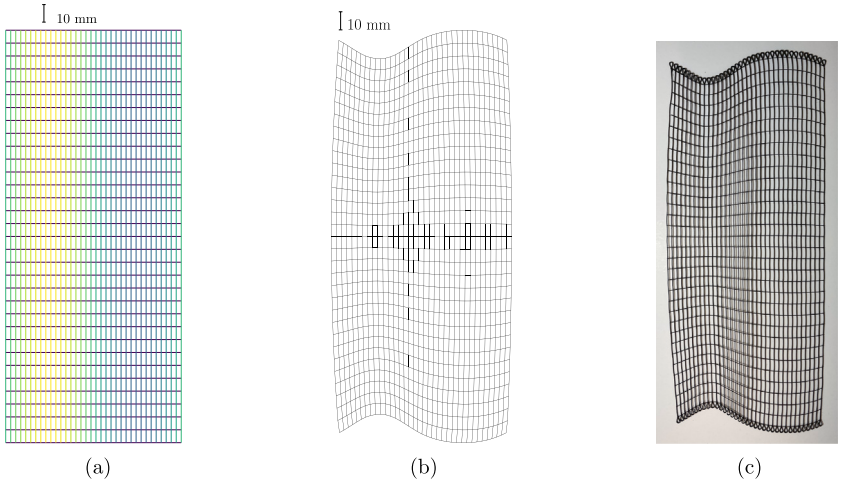


Fig. 4. Moment exerting wrap. a) The network tension gradient of an unwrapped cylinder is used to generate, b) a design file of the network after optimizing the edge lengths, then to 3D print c) the network of the cylinder mesh using 3Dn nScrypt.

Fig. 6. a) Graphical image of the expandable octahedron (red: struts, blue: cables) and b) the assembled tensegrity.

working mechanism. Potential applications include customizable compression casts or splints with locally adjustable tension [37,38].

2.3. Case-3: tensegrity system

Printing 3D cable networks suspended in the air is typically not feasible using planar 3D printing techniques, such as FFF. Instead, a network must first be flattened while preserving its topology, which is particularly challenging when the network is spatially complex and densely connected, as is the case in tensegrity structures. Flattening can introduce internal crossings between edges that were not originally intersecting in 3D space. These crossings must be identified and resolved to preserve the intended topology (see Section 3.4 for details). Note that the flattening and crossing resolution steps are automated to ensure the method can be scaled to more complex networks.

To showcase this challenge, a classic tensegrity structure is manufactured: the expandable octahedron [47]. This structure includes an intricate 3D cable network, making it a good candidate for validating the flattening and intersection-resolution methods. It consists of three parallel pairs of struts connected by 24 cables, forming a symmetric, force-balanced configuration.

The force density method is a useful tool for designing tensegrity structures, but not every set of force densities results in a stable configuration. Previous work has shown that the force density ratio $-q_{\text{strut}} = 1.5q_{\text{cable}}$ did not result in a stable system where each cable and each strut has equal lengths, and the length ratio between struts and cables was 1.632 [48–50]. The designed structure is depicted in Fig. 6(a) (an animation is available in the supplementary material). For a strut length of 80 mm and a cable stress of 7.5 MPa (~ % strain), the resulting force densities were -0.018 for struts and 0.012 for cables.

The tensegrity was fabricated in two steps: struts were printed using PETG, and the cable network was made from TPU. Due to the significantly higher stiffness of PETG and the larger cross-sectional area of the struts compared to the cables, the strut deformation was neglected: $l^0 = l^1$.

The cable network was flattened using a polar coordinate transformation, and intersecting edges were resolved before optimization. The designed and fabricated tensegrity networks are shown in Fig. 6. Flattening the cable network had the side effect of introducing additional crossings. These additional crossings were automatically resolved using the methods described in Section 3.4.

Optimization was performed with parameters $\beta = 0.1$, $\tau = 10^{-6}$, yielding a final error $\epsilon_{1907} = 1.7 \times 10^{-4}$. The network was scaled down by a factor of $s = 0.9999$. The tensegrity was manufactured in 1 minute and 7 s.

Programmable tensegrity structures such as this could support applications in cable-driven exosuits, such as CAREX [39] or CRUX [4], where tunable stiffness and force transmission are critical.

3. Methods

This section discusses methods for manufacturing 3D-printed networks with programmable tension. We detail the methods for (i) defining the goal network using the force density method; (ii) converting this goal network into an unstretched counterpart; (iii) transforming the network into a 3D printing compatible structure; (iv) flattening 2.5D/3D networks; and (v) accounting for unwanted crossings. Steps (i) and (ii) are also described graphically in Fig. 7, step (iii) in Fig. 8, and steps (iv) and (v) in Fig. 9.

3.1. Form-finding

Tensioned structural networks are typically defined using a form-finding technique: the force density method [51]. This technique was developed to design tensioned networks with equilibrated structures for given boundaries and topologies [52], such as tensioned roofs (e.g., the Munich Olympic Stadium roof (1972) [53] and fabric formwork systems for the construction of curved concrete shells [54]). The force density method linearizes the static force equilibrium equations, enabling the fast form-finding of networks with desired shapes and tension gradients. First, the topology of a network containing M edges and N vertices is described by an $M \times N$ connectivity matrix C_s:

$$C_s(i,j) = \begin{cases} +1 & \text{if edge i starts at vertex j} \\ -1 & \text{if edge i ends at vertex j} \\ 0 & \text{in the other cases.} \end{cases} \tag{3}$$

Next, a force density q_i is assigned to each i-th edge in the network, where q_i represents the ratio of tension F_i to stretched length l_i of the edge, as $q_i = F_i/l_i$. Vertices are divided into free $\mathbf{x}$ and fixed $\mathbf{x}_f$ vectors. Optionally, an external load vector $\mathbf{p}$ can be applied at $\mathbf{x}$. The free vertex coordinates $\mathbf{x}$ can be found by solving

$$\begin{aligned} &\mathbf{x} = D^{-1}\left(\mathbf{p} - D_f\mathbf{x}_f\right), \text{with} \\ &D = C^T Q C, \text{and} \\ &D_f = C^T Q C_f. \end{aligned} \tag{4}$$

where Q represents the diagonal matrix of force densities q_i. The full connectivity matrix is divided into free and fixed columns according to $C_s = [C \quad C_f]$. The tension in each edge can be determined using $F_i = l_i q_i$. Optionally, a user can update the force densities in a network until the desired shape and tension gradient are achieved. If required, a nonlinear force density method can be used to incorporate constraints on the network's node locations, tensions, edge lengths, or any desired combination of these. A network that meets these constraints is found by employing numerical optimization [53]. The force density method does not require a constitutive model for form-finding. The unstretched lengths of the edges, l^0, are determined only after form finding, based on the engineering strain relationship and the general stress definition. These quantities are computed independently for each edge:

$$l_i^0 = \frac{l_i}{1+\epsilon(\sigma)}, \quad \text{where} \quad \sigma_i = \frac{F_i}{A_i}. \tag{5}$$

This calculation requires a constitutive model, $\epsilon(\sigma) := f(\sigma)$, and the cross-sectional areas A of the edges. An Ogden hyperelastic material model is constructed from tensile tests, and the cross-sectional area is determined by printing a 36 m continuous fiber and measuring the amount of filament used (more info in Appendix A).

3.2. Network optimization, scaling, and arc generation

Printing the equilibrated form directly would result in the correct geometry, but without tensions. To ensure the designed tension gradient, each edge in the network must be manufactured with the unstretched lengths l^0. The first step toward achieving this is by numerically optimizing the vertex locations.

The optimization problem is highly coupled since adjusting the coordinates of a vertex to satisfy one length constraint inevitably affects the lengths of neighboring edges. This coupling makes the optimization procedure challenging, as a coordinate update in one iteration can

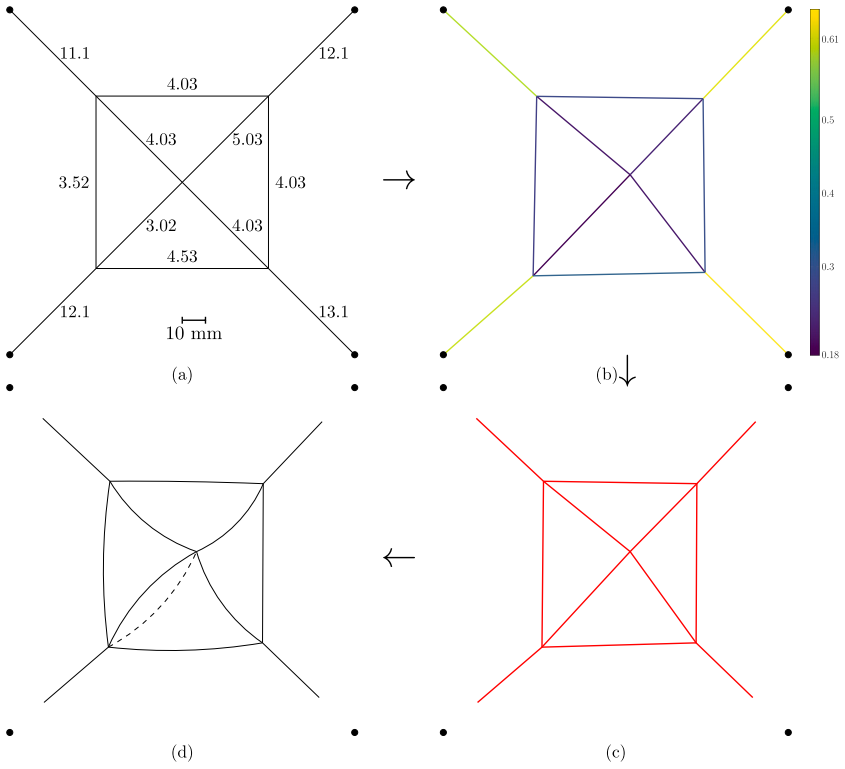


Fig. 7. a) A unit cell topology with user-described force densities q times a thousand (N/mm) labeled at each edge. The corner circles indicate fixed points $\mathbf{x}_f$. b) The stretched shape as found with the force density method, see Section 3.1. The color represents the tension in the edge in Newton. c) The shape of the network after optimization and scaling. d) Each edge is converted to an arc with arc length l^0. Note the dashed arc: Edges can be turned into arcs in two directions. Directions should be flipped to ensure no overlapping features.

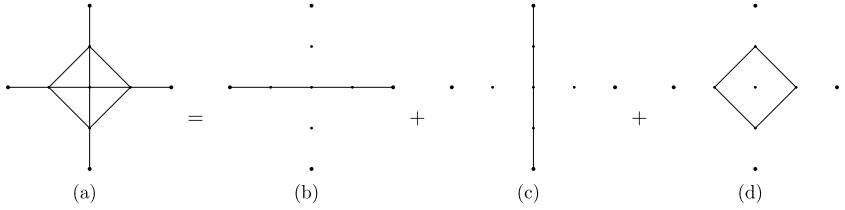


Fig. 8. A unit cell network (a) is used for validating the method. The unit cell can be decomposed into parts b, c, and d.

propagate errors into subsequent ones, leading to oscillations or slow convergence. Common state-of-the-art optimization techniques, such as L-BFGS-B [55], update all coordinates simultaneously based on global gradient and Hessian approximations, which can result in conflicting updates across coupled regions. These methods often produce exact but computationally intensive steps, making them relatively inefficient for problems with strong local dependencies. In contrast, a Gauss-Seidel optimization algorithm—often referred to as a relaxation method—updates a single vertex pair at a time and immediately incorporates each change into subsequent corrections, promoting more stable and efficient convergence. Moreover, because these updates are localized, the algorithm scales linearly with network size in sparse systems, making it well-suited for large-scale form-finding tasks [56].

An overview of the Gauss-Seidel optimization algorithm is summarized as follows:

(i) Use the form retrieved with the force density method as the initial guess $\mathbf{x}_0$. For curved or 3D networks, the initial guess first requires flattening and possibly intersection resolution, as detailed in Section 3.4.

(ii) Iterate over each vertex pair in the network and update their coordinates $\mathbf{x}^a$ and $\mathbf{x}^b$. The update procedure for a generic edge is

$$\mathbf{x}^a_{k+1} = \mathbf{x}^a_k + \beta(l^0 - l^1_k)\frac{\mathbf{x}^a_k - \mathbf{x}^b_k}{2l^1_k},$$
$$\mathbf{x}^b_{k+1} = \mathbf{x}^b_k - \beta(l^0 - l^1_k)\frac{\mathbf{x}^a_k - \mathbf{x}^b_k}{2l^1_k}. \tag{6}$$

Here $l^1_k = \|\mathbf{x}^a_k - \mathbf{x}^b_k\|$ is the current edge length in iteration k, and $\beta \in (0, 1]$ is a numerical damping factor to improve convergence (e.g., $\beta = 0.1$).

(iii) Repeat step (ii) until convergence or until a maximum number of steps is reached (10,000 steps unless stated otherwise). Convergence is reached when the difference between the current total error ϵ_k and the previous step's error ϵ_{k-1} is smaller than a predefined tolerance τ, e.g., 10^{-6}.

In many optimization problems, care is taken to avoid convergence to local minima in favor of finding a global minimum. This is typically

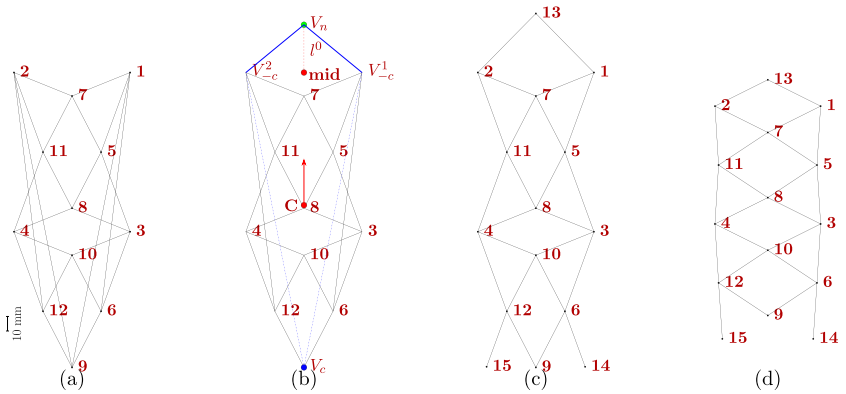


Fig. 9. Network transformation on the tensegrity network in Section 2.3. (a) The cable network of the tensegrity is flattened. $\mathbf{c} = (0, 0, 0)$, and no rotations are applied before the Cartesian-to-polar transformation, i.e., $\mathbf{t} = [0; 0; 1]$. b) Steps to remove crossings: Identify the vertex with the most crossings, $V_c = 9$, and duplicate it to create $V_n = 13$. The new vertex V_n is placed along the direction from the network center C toward the midpoint of the complementary vertices $V_{-c} = \{1, 2\}$. Its distance from mid is set to the mean target length l^0 of the edges in E_c. At the intersecting edges, the old vertex V_c is replaced with V_n, as indicated by the dashed blue edges changing to solid blue edges. c) Repeat the process until no crossings are left. d) The network after relaxation.

addressed by exploring a variety of initial conditions or applying regularization techniques. However, in our case, the nearest local minimum is preferred. While a lower-error configuration could, in theory, be found by allowing a vertex to move beyond its neighboring vertices, such a result would introduce edge crossings and distort the initial geometry. This would lead to forms that are difficult or impossible to manufacture. To preserve manufacturability, the optimization is constrained to maintain the original geometric layout. Therefore, a low damping factor, β, is recommended to ensure that vertex updates are conservative and do not cause significant, destabilizing changes in geometry. The importance of the initial form $\mathbf{x}^0$, which influences the local minimum reached during optimization, is further emphasized by the process of flattening 2.5D and 3D networks, as detailed in Section 3.4.

Some residual errors may remain after optimizing the vertices. To resolve these residual errors, the network is scaled down with a scalar $s = \min\left(\mathbf{l}^1/\mathbf{l}^0\right)$, such that all edge lengths in the network are shorter than or equal to their unstretched counterparts ($s\mathbf{l}^1 \leq \mathbf{l}^0$).

The final step is to turn each edge into an arc with a curve length l^0. Each arc is defined by a radius R and an angle α, which can be determined for a generic edge by solving

$$l^0 = R\alpha \qquad \text{(Arc length equation)} \tag{7}$$

$$\frac{sl^1}{2} = R\sin\left(\frac{\alpha}{2}\right) \qquad \text{(Trigonometric relation)} \tag{8}$$

Rewriting Eqs. (8) and (7) yields an expression for the length ratio as a function of the arc angle α:

$$\frac{sl^1}{l^0} = \frac{2}{\alpha}\sin\left(\frac{\alpha}{2}\right). \tag{9}$$

A cubic interpolation function of Eq. (9) is set up to avoid solving a transcendental equation every time an arc angle needs to be determined from a length ratio. Further details can be found in Appendix C.3. One final post-processing step is considered: leaf edges. Consider an edge of which one vertex is connected to only one other vertex. After scaling down the network, the leaf edges can be made to the exact desired length without penalty. Leaf edges are accounted for by moving their free vertex in the direction of the edge's long axis such that the edge has a length of l^0.

3.3. Network fabrication

The steps to convert the flattened, relaxed, and scaled networks into machine code compatible with FFF are as follows. Decompose the network into continuous paths (Section 3.3.1). Account for intersections between the printed path and previously printed paths (Section 3.3.2).

3.3.1. Path decomposition

The FFF process is most unstable at the beginning and end of a print path, primarily due to the viscoelastic behavior of the melted polymers. This viscoelasticity introduces a delay between the intended and actual start/stop locations, which may compromise printing precision. Therefore, minimizing the number of start–stop events by printing structures in as few continuous paths as possible is highly desirable. In this work, structures are manually decomposed into printable paths. For example, a unit cell structure, later used for validation, can be divided into three sections: a horizontal path, a vertical path, and a loop, as shown in Fig. 8(a).

Automatic decomposition of networks into continuous edge-covering paths can be achieved using optimization-based methods [57]. However, computing exact solutions is an NP-hard problem, which limits the feasibility for large graphs. To address this, developments such as the Hybrid Lagrangian Relaxation and Particle Swarm Optimization (LaPSO) approach offer relatively scalable approximations that yield practical solutions for complex networks [58]. These methods provide a promising foundation for automating and generalizing the decomposition process, especially when manual design becomes impractical.

3.3.2. Intersections

After decomposing the network into printable paths, the intersections between them must be identified. For each printable path, a set of intersecting vertices is computed by finding the intersection between its vertices and the union of the vertices from all previously printed paths. When the print path approaches an intersection, the print nozzle raises to avoid collision. This raising follows a linear trajectory, beginning 0.6 mm before the intersection point (1.5 times the nozzle diameter) and lifting by one layer height. The transition is designed with a slight overlap to ensure good adhesion between layers. A parameter study confirmed that this approach results in reliable adhesion with minimal damage to previously printed paths. In practice, permanent deformation of the meshes was observed to occur before debonding at intersections.

3.4. Flattening and accounting for crossing edges

To print curved (2.5D) or 3D networks on a flat 2D printer bed, networks must first be flattened, i.e., projected onto a planar surface. It is vital that the original geometric layout of the network is preserved as much as possible to ensure reliable manufacturability. The proposed flattening procedure is a coordinate transformation from Cartesian space to a polar or spherical coordinate system defined with respect to a user-defined unit vector $\mathbf{t}$ and center point $\mathbf{c}$. The final flattening procedure

consists of disregarding vertex-specific radii and using a mean radius for all vertices. The procedure is set out in more detail in Appendix C.2.

The resulting flat 2D network can be seen as the initial guess $\mathbf{x}_0$ for the Gauss-Seidel relaxation procedure (Section 3.2). However, the following issue needs to be addressed first. The flattened network can have new internal crossings, thereby altering its inherent topology. The first step in accounting for these new crossings is to identify them all. This is achieved by checking all possible edge pairs for crossings, which scales with $O(M^2)$, where M is the number of edges. Despite this quadratic scaling, the process remains efficient for moderately sized networks. For instance, a network with 1000 edges is expected to process in under 1 s, assuming an edge-edge check takes approximately 0.1 µs. The resolution of a crossing is explained in Fig. 9. The center of all vertices is defined as C. For each vertex involved in crossing edges, the number of crossings it participates in is counted. The vertex with the most crossings is denoted $V_c = 9$, and its crossing edges are $E_c = [1,9]$ and $[2,9]$. The complementary vertices of these edges are $V_{-c} = \{1,2\}$. A new vertex V_n is introduced along the direction from C toward the midpoint of V_{-c}. Its distance from C is set to the mean target length l^0 of the edges in E_c. The vertices V_c in edges E_c are then reassigned to V_n. This duplication and reassignment process is repeated until all crossings are resolved.

3.5. Limitations

The manufacturability of a network depends strongly on its topology, the tension gradients it is designed to achieve, and the extent to which it deviates from a flat geometry, making it challenging to quantify general limitations. In networks with steep tension gradients or in 2.5D/3D configurations where radii vary widely after coordinate transformation, the Gauss-Seidel optimization may cause a vertex to move past a neighbor. This can introduce unintended crossings and alter the network's inherent form. A method to resolve such crossings is discussed in Section 3.4, although it introduces additional assembly effort.

Despite the difficulty in quantifying manufacturability in general terms, a theoretical lower bound can be established based on geometric constraints. Take an overdefined network, that is, no vertex position exists that satisfies all desired edge lengths. To overcome this, the methods allow for global scaling and printing of edges as arcs. However, the minimum achievable arc-to-chord length ratio is 0.6366 (see Eq. 9 at $\alpha = \pi$), establishing a hard lower bound on the manufacturability of designs.

The force density method is a form-finding technique for structures composed of elements that bear loads only in the axial direction, such as bars and cables. In this work, the network edges were made from a flexible polymer, TPU, using a high-precision 3Dn nScrypt printer. The edges were assumed to behave like cables. This assumption is only valid as long as their lengths are significantly greater than their widths: $l^0 \gg w$. It is expected that, as networks are printed with shorter edges, the method's accuracy decreases. This limitation is tested, and the results are presented in Section 4.1.

Similarly, cases with short lengths and large radii need to be considered. In such cases, the edge length will vary across its thickness. The inner length will be shorter, and the outer length will be longer than the designed length l^0. This limitation has also been tested, and the results are presented in Section 4.1.

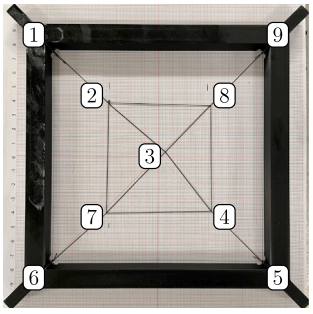
The flattening technique provided in Section 3.4 is limited to basic 3D designs. For more complex 3D structures, the authors recommend exploring sophisticated unwrapping algorithms designed to handle complex topologies, such as Lévy et al. [59] in future studies.

4. Validation

The methods for manufacturing networks with programmable tensions are validated using a unit cell structure suspended within a frame. The design process is described in Fig. 7, and the printed unit cell is shown in Fig. 10. The unit cell is photographed orthogonally on top of printed paper with a 1 mm–spaced grid, allowing the vertex locations $\mathbf{x}^m$ to be measured. Exact vertex positions can be interpolated between the grid lines in the photograph by counting pixels, achieving a measurement tolerance of 0.1 mm. The frame ensures the unit cell is in contact with the grid paper, minimizing image distortion and perspective errors. The agreement between the measured and designed edge lengths is quantified using

$$\text{error} = \frac{1}{M}\sum_{i=1}^{M}\frac{|l_i^{1,m} - l_i^1|}{l_i^1} \cdot 100\ \% \tag{10}$$

where the distance is normalized with the target edge lengths to make the error scale-invariant. Since the vertex coordinates and tensions are



Vertex No.	x	$\mathbf{x}^m$
1	(-74.2, 74.2)	-
2	(-34.1, 37.0)	(-34.0, 38.8)
3	(6.16, 3.25)	(6.7, 4.3)
4	(38.4, -38.8)	(39.6, -38.3)
5	(74.2, -74.2)	-
6	(-74.2, -74.2)	-
7	(-35.6, -40.3)	(-34.4, -39.5)
8	(37.6, 36.0)	(38.2, 37.2)
9	(74.2, 74.2)	-

Fig. 10. The unit cell corresponding to 7.3 MPa in Fig. 11(a) suspended in the frame. The tables shows the designed $\mathbf{x}$ and measured $\mathbf{x}^m$ vertex locations, enabling evaluating Eq. (10) to determine the mean edge length error to be 0.52 %.

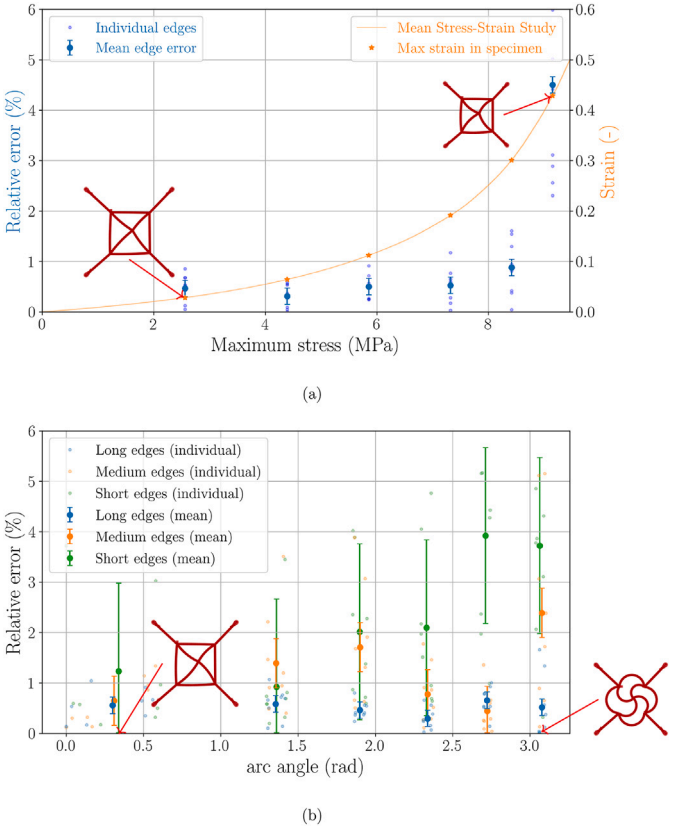


Fig. 11. a) Blue axis (left): Relative edge length error versus maximum specimen stress. Orange axis (right): Average measured stress–strain relationship of TPU. High accuracy is observed for stresses up to 7.3 MPa. b) Relative edge length error versus arc angle for Long (≈62 mm), Medium (≈21 mm), and Short (≈5.8 mm) edges.

linked through equilibrium equations (e.g., see Eq. 4), validating the vertex positions implicitly validates the tensions. The validation structures are intentionally asymmetrical, ensuring that any deviations in the tension gradient would produce noticeable distortions, which cannot cancel out. The mean edge length error of the validation structure in Fig. 10 is 0.52 %.

4.1. Testing limitations

Expected limitations of the proposed methods are discussed in Section 3.5. The performance of these methods can be quantified using Eq. (10). In Fig. 11(a), the edge length error is plotted as a function of the edge stress to identify the method's limits. The results show that the methods remain accurate for stresses up to 7.3 MPa, with errors below 1 %. Reduced accuracy at higher stresses is attributed to the sensitivity of the stress–strain curve in this range, where small strain offsets produce larger errors.

Next, the edge length error is plotted against the edge arc radius for Long (≈62 mm), Medium (≈21 mm), and Short (≈5.8 mm) edges, as shown in Fig. 11(b). The data indicate that the methods maintain accuracy even at very high arc angles. Still, two sources of increased error are observed: short edges alone cause higher errors, and the combination of short edges with high arc angles leads to even larger errors. Both effects are consistent with the limitations discussed in Section 3.5. Notably, accuracy remains within 2.1 %, even for short edges, provided that edges are printed with an arc length below 2.4 rad. Environmental conditions during printing and testing were consistent (Humidity: $\mu = 25.3$ %, $\sigma^2 = 0.51$ %; Temperature: $\mu = 23.0\,^{\circ}\mathrm{C}$, $\sigma^2 = 0.26\,^{\circ}\mathrm{C}^2$).

5. Conclusion

The presented work demonstrated a scalable, accessible approach to fabricating network structures with programmable tension gradients using standard FFF techniques. By introducing a design algorithm that transforms tensioned 2D, 2.5D, and 3D cable networks into flat, relaxed layouts, the method enabled the direct 3D printing of entire cable networks as single, continuous pieces. This innovation addressed the longstanding challenge of achieving precise, pre-programmed cable tensions in miniaturized tensegrity and network structures—an essential factor for their mechanical performance and functional adaptability. The approach further streamlined the manufacturing process by minimizing assembly steps and errors, and it leveraged numerical optimization and geometric transformations to ensure that printed networks, upon suspension, realized their intended tension distributions and structural forms.

Our method achieved a favorable balance of speed, functionality, and manufacturability compared to current methods, such as the manual modular approach [54], direct 3D printing of struts and cables [17], or multi-material 3D printing with molded cables [32].

Experimental validations, including the fabrication of spider web-inspired networks, moment-exerting meshes, and a classic tensegrity structure, highlighted the method's versatility in reproducing complex tension gradients. Validation steps confirmed that the printed networks closely matched their designed geometries and mechanical properties, with average edge strain errors remaining low: the methods remained accurate within 1 % for tensile stresses up to 7.3 MPa, and within 2.1 % for edges as short as 5.8 mm, provided the arc angle of short edges does not exceed 2.4 rad. While some limitations persisted—such as restrictions on manufacturable tension gradients and geometric configurations due to material and process constraints—the method's compatibility with widely available 3D printers democratizes access to programmable tensegrity fabrication. This opens new avenues for customizable, lightweight, and adaptive devices in fields ranging from medical orthotics to wearable robotics, paving the way for a broader adoption of tension-programmed structures in both research and practical applications.

CRediT authorship contribution statement

T.C.P. Masmeijer: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **C.C. Swain:** Writing – review & editing, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation. **J.R. Hill:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Conceptualization. **E. Habtour:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

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.

Acknowledgment

The authors are grateful for the support provided by the National Science Foundation Award No. 2341950: Can Irregular Structural Patterns Beat Perfect Lattices? The authors also acknowledge Victoria Leigh's and Keira Morrissey's extensive efforts to assist with the method validation. The authors acknowledge using the GitHub repository GcodeGenerator, by Tibor Barši, which was used to generate the 3D printer machine code for the test cases in this manuscript. The authors thank Reagan Rogers and Myah Derderian for their assistance with the mechanical testing of the TPU. The authors are grateful to Christy Graves for designing the graphical abstract to facilitate approachable scientific communication.

Appendix A. Constitutive model

To determine the unstretched lengths (l^0) of the edges in a structural network, an accurate constitutive model is required. In this study, stress–strain tests were conducted to model the nonlinear behavior of the Overture TPU, as shown in Fig. A1. Following ASTM Standard D882-18, eight specimens with $5 \times 0.2 \times 100$ mm^3 dimensions were tested at a displacement rate of 10 mm/min. The maximum material strain was 50 %. An Instron 3300 with a 500 N load cell was utilized to conduct uniaxial stress–strain tests.

It is challenging to determine the area of a sheet of TPU that is only one 3D print layer thick. However, calculating the stress using the theoretical cross-sectional area from the CAD model is insufficient to achieve a reasonably accurate stress–strain curve. Therefore, the cross-sectional area of a single edge A in an FFF network is determined by printing a single edge of length $L_e = 36.22$ m, with negligible tolerances. After printing, the length of the used filament was $L_f = 1.62 \pm 0.02$ m. The cross-sectional area of the filament A_f was specified in the technical datasheet $A_f = 1.75$ mm$^2 \pm 0.02$. Using conservation of volume, the cross-sectional area of a single edge was determined according to:

$$A = \left(A_f \cdot L_f\right) / L_e = 0.0783 \pm 0.002 \text{ mm}^2 \tag{A.1}$$

Typically, a constitutive model for a nonlinear elastic material is based on the strain energy density function S. Thus, the stress can be obtained simply by taking the derivative of S with respect to strain. Characterizing the properties of rubbery materials is often based on the stretch ratio rather than strain; i.e., $\lambda = l/l^o = 1 + \varepsilon$. Because the deformation in TPU is three-dimensional, the strain can be related to the principal stretch ratios [60]:

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \qquad I_2 = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2, \quad \text{and} \quad I_1 = \lambda_1^2\lambda_2^2\lambda_3^2$$

where λ_i denotes the principal stretches in the 1-2-3 (or xyz) directions. The Ogden model was used in this study, as it provided a good fit to the

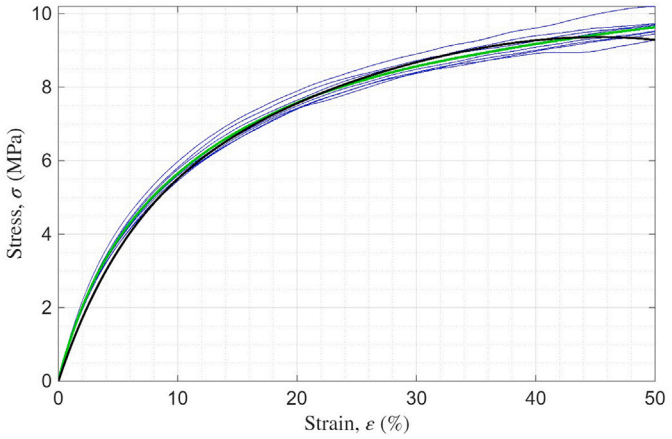


Fig. A1. Stress-strain curve of Overture TPU. The blue curves are experimental data for eight specimens. The green curve is the average of the eight specimens. The black curve is the Ogden Material Model.

experimental data in Fig. A1. The strain energy function for this model is [61]:

$$S = \sum_i \frac{\mu_i}{\alpha_i}\left(\lambda_1^{\alpha i} + \lambda_2^{\alpha i} + \lambda_3^{\alpha i} - 3\right) \tag{A.2}$$

The α_i and μ_i are material constants estimated numerically using Hyper-Data, a Matlab-based optimization [61] based on the uniaxial test data. These constants were $\alpha_i = (0.0024, 7.04, -13.6)$ and $\mu_i = (81634, -5.64, -6.26)$. The material response was assumed to be incompressible and isothermal; hence $\lambda_1\lambda_2\lambda_3 = 1$. This assumption was used to obtain an expression for λ_2 and λ_3 using only uniaxial stress–strain tests, as follows [26]:

$$\lambda = \lambda_1 = \frac{l}{l_o}, \lambda_2 = \lambda_3 = \sqrt{\frac{l_o}{l}} \tag{A.3}$$

Finally, the uniaxial nominal stress was obtained by differentiating S with respect to λ instead of ε.

$$\sigma = \sum_i \mu_i \left(\lambda_1^{\alpha_i - 1} - \lambda_1^{\frac{-\alpha_1}{2} - 1}\right) \tag{A.4}$$

Appendix B. Dynamic loading behavior

For control scenarios in future applications, it is desirable to predict how TPU cables will behave under dynamic loading conditions. Hysteresis testing was performed to measure differences in the stress–strain curves for loading vs. unloading. Cyclic testing was performed to determine how the stress–strain curve would change from cycle to cycle. For these tests, each specimen was loaded to 20 % strain and, without pausing, returned to zero strain, at which point it was allowed to recover until stress stabilized. This process was performed 3 times. After the 3rd cycle recovery, the process was repeated 17 times for a total of 20 cycles, except that the specimen was not allowed to recover between cycles. Loading and unloading were performed at a rate of 10 mm/min. An exponential decay rate was observed (see Fig. B1). A curve fit of the exponential decay of the stress at 20 % strain yields ~ 26 cycles to settle (see Fig. B2).

It was also desirable to ascertain the time constant for TPU recovery. Two 1 × 1 tensegrity arrays [10] were tested by applying loads from 0 to 650 grams (0 to 6.37 Newtons) in increments of 50 grams to the control string. The position of Node 2 was then measured. Each array was allowed to rest for 24 hours before retesting. The process was then repeated with a two- and three-day rest. The results are shown in Fig. B3. It was determined that TPU requires ~ 2 days to recover fully.

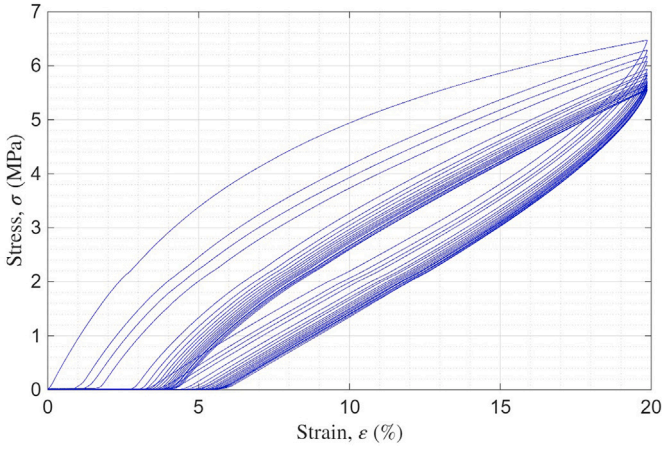


Fig. B1. Average stress-strain curve for three specimens of Overture TPU.

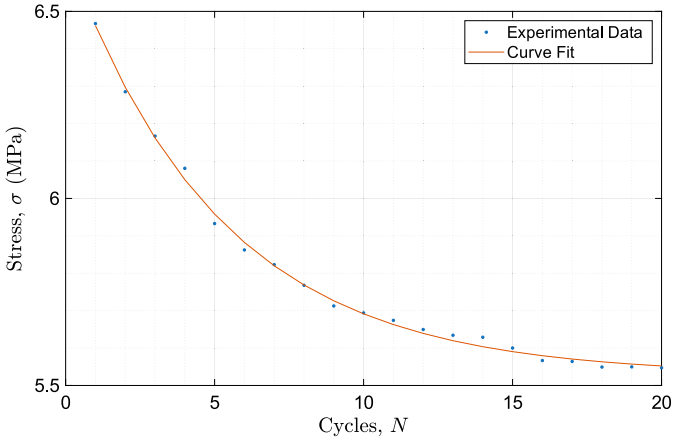


Fig. B2. Exponential decay of the stress at 20 % strain as a function of the number of cycles.

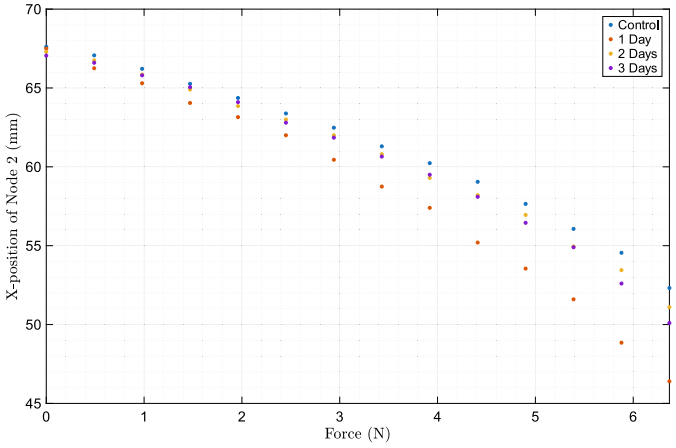


Fig. B3. Average position vs mass plot of 2 arrays when allowed to rest for 1 day, 2 days, and 3 days.

Strain-rate dependency is another important characteristic of polymers to consider. In this work, we did not consider the loading rate. Instead, we assumed that the test frames were loaded sufficiently slowly to be considered quasi-static.

Appendix C. Additional methods

Additional methods are detailed here.

C.1. Validation methods

Results of validation were set out in Section 4, where unit cells with long (≈62 mm), medium (≈21 mm), and short (≈5.8 mm) edges were suspended in a frame. The frame was 3D printed from PETG. The unit cells were printed with a 9 mm long loop, allowing them to be suspended in the frames. The hooks on the frame were designed such that the fixed points $\mathbf{x}_f$ are located 210 mm apart for the unit cells with long edges, or 70 mm apart for the unit cells with medium and short edges. The average edge length error was calculated using equation Eq. (10). Notably, the edges used for suspending the unit cells were excluded from the analysis. The motivation is that any offset between the grid line paper and the PETG frame would cause additional error only in the connecting edges, not in the other edges. In Fig. C1 a technical drawing of the frame is depicted.

C.2. Flattening with cylindrical or spherical coordinates

To print curved 2D or 3D networks, the network must be flattened (see Section 3.4). The detailed approach to flattening using a coordinate transformation is as follows:

(i) **Translation:** Shift $\mathbf{x}$ such that the vertices have the center point $\mathbf{c}$ as the origin:

$$\mathbf{p}_j = \mathbf{x}_j - \mathbf{c}$$

(ii) **Rotation(s):** Rotate the points such that the user-defined unit vector $\mathbf{t}$ aligns with the z-axis.

(iii) **Coordinate transformation:** Apply a coordinate transformation:

Cylindrical coordinates	Spherical coordinates
$\theta_j = \text{atan2}(p_{j,y}, p_{j,x})$	$\theta_j = \text{atan2}(p_{j,y}, p_{j,x})$
$r_j = \sqrt{p_{j,x}^2 + p_{j,y}^2}$	$\phi_j = \text{atan2}(p_{j,z}, \sqrt{p_{j,x}^2 + p_{j,y}^2})$
$z_j = p_{j,z}$	$r_j = \sqrt{p_{j,x}^2 + p_{j,y}^2 + p_{j,z}^2}$

(iv) **Dimensional reduction:** The radial coordinates $\mathbf{r}$ are disregarded, and the angular coordinates are scaled with the mean radius $\bar{r}$, such that for cylindrical coordinates

$$x_j = \theta_j \cdot \bar{r}, \quad y_j = z_j,$$

and for spherical coordinates:

$$x_j = \theta_j \cdot \bar{r}, \quad y_j = \phi_j \cdot \bar{r}.$$

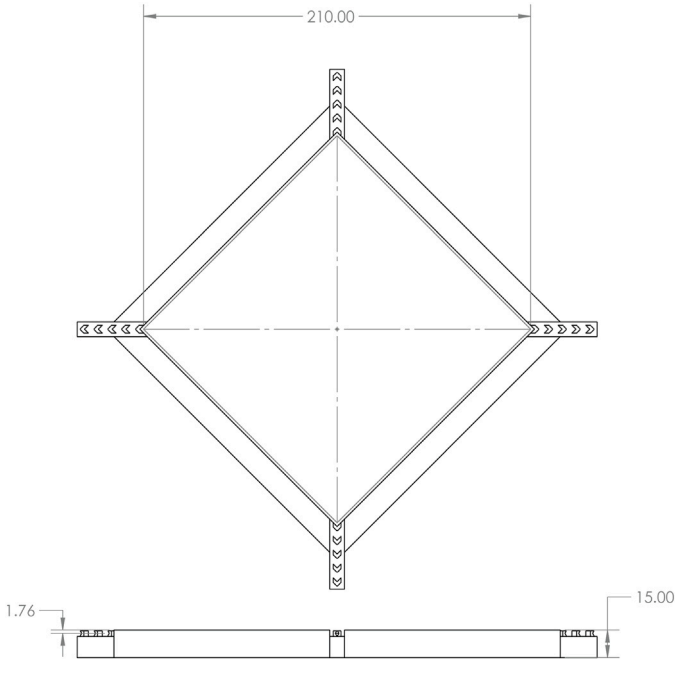


Fig. C1. Technical drawing of the bottom and side view of the frame used for validating the long-edged unit cells. A similar frame was printed for unit cells with Short and Medium edges, but instead of 210 mm, the diagonal length between fixed points was 70 mm.

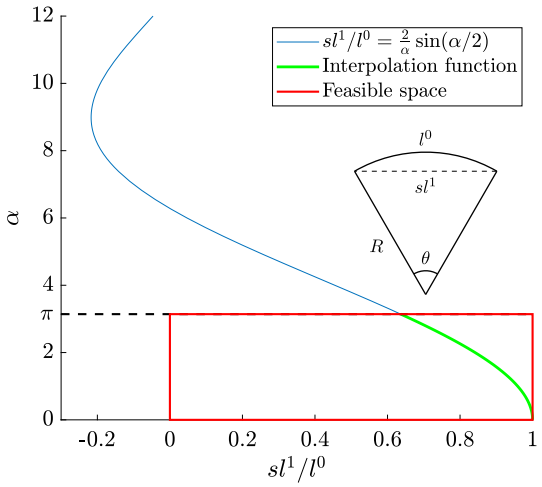


Fig. C2. Determining arc parameters. α can be determined fast by evaluating a cubic interpolation function in the feasible space.

C.3. Determining arc parameters

The interpolation function is only set up for a feasible space. Lengths are positive and an arc length is always longer than its chord length, i.e. $0 \leq l^1/l^0 \leq 1$. The limits for α become $0 \leq \alpha < \pi$ when only allowing minor arcs and positive angles. The arc radius is found by evaluating $R = l^0/\alpha$ (Fig. C2).

Appendix A. Supplementary material

Supplementary material to this article can be found online at doi:10.1016/j.matdes.2025.115281.

Data availability

The authors provided GitHub link to all code and data to download freely.

References

[1] N. Pollini, Gradient-based prestress and size optimization for the design of cable domes, Int. J. Solids Struct. 222–223 (Jul. 2021) 111028. ISSN: 00207683, https://doi.org/10.1016/j.ijsolstr.2021.03.015

[2] J. Kłosowska, P. Obara, W. Gilewski, Self-stress control of real civil engineering tensegrity structures, AIP Conf. Proc. (2018) 150004, https://doi.org/10.1063/1.5019157

[3] W. Gilewski, J. Kłosowska, P. Obara, The influence of self-stress on the behavior of tensegrity-like real structure, in: MATEC Web of Conferences, vol. 117, Jul. 2017, https://doi.org/10.1051/matecconf/201711700079, p. 00079. ISSN: 2261-236X.

[4] S. Lessard, et al., CRUX: a compliant robotic upper-extremity exosuit for lightweight, portable, multi-joint muscular augmentation, in: 2017 International Conference on Rehabilitation Robotics (ICORR), IEEE, Jul. 2017, pp. 1633–1638, https://doi.org/10.1109/ICORR.2017.8009482. ISBN: 978-1-5386-2296-4.

[5] Y. Chen, et al., Towards human-like walking with biomechanical and neuro-muscular control features: personalized attachment point optimization method of cable-driven exoskeleton, Front. Aging Neurosci. 16 (Feb. 2024) 1663–4365, https://doi.org/10.3389/fnagi.2024.1327397

[6] F. Renda, et al., Dynamic model of a multibending soft robot arm driven by cables, IEEE Trans. Robot. 30(5) (Oct. 2014) 1109–1122. ISSN: 1552-3098, https://doi.org/10.1109/TRO.2014.2325992

[7] N. Willemstein, et al., Pellet-based 3D printing of soft thermoplastic elastomeric membranes for soft robotic applications, Mater. Des. 258 (Oct. 2025) 114589. ISSN: 02641275, https://doi.org/10.1016/j.matdes.2025.114589

[8] B.D. Layer, H. Denning, J.R. Hill, Control and locomotion of tensegrity robots through manipulation of the center of mass, Robotica 42 (9) (Sept. 2024) 2885–2907. ISSN: 0263-5747, https://doi.org/10.1017/S0263574724001139

[9] D. Zappetti, et al., Variable-stiffness tensegrity spine, Smart Mater. Struct. 29 (7) (Jul. 2020) 075013. ISSN: 0964-1726, https://doi.org/10.1088/1361-665X/ab87e0

[10] A.R. Brown, et al., Development of a continuous cable tensegrity, in: American Society of Mechanical Engineers, Sept. 2024, https://doi.org/10.1115/SMASIS2024-140185. ISBN: 978-0-7918-8832-2.

[11] B. Shekastehband, N. Pourmand, Effects of self-stress distributions on stability of tensegrity structures, Int. J. Struct. Stab. Dyn. 17 (3) (Apr. 2017) 1750029. ISSN: 0219-4554, https://doi.org/10.1142/S0219455417500298

[12] T. Masmeijer, et al., Directional DIC method with automatic feature selection, Mech. Syst. Signal Process. 224 (2025) 112080.

[13] H. Furuya, Concept of deployable tensegrity structures in space application, Int. J. Space Struct. 7 (2) (Jun. 1992) 143–151. ISSN: 0956-0599, https://doi.org/10.1177/026635119200700207

[14] S.D. Shinde, S.H. Upadhyay, The novel design concept for the tensioning system of an inflatable planar membrane reflector, Arch. Appl. Mech. 91 (4) (Apr. 2021) 1233–1246. ISSN: 0939-1533. https://doi.org/10.1007/s00419-020-01841-w

[15] B. Wang, et al., Space deployable mechanics: a review of structures and smart driving, Mater. Des. 237 (Jan. 2024) 112557. ISSN: 02641275, https://doi.org/10.1016/j.matdes.2023.112557

[16] E. Shang, et al., Adaptive deployable structure enabled by actively controlled tensegrity for space debris removal, Adv. Sci. 12 (14) (Apr. 2025) 2198–3844, https://doi.org/10.1002/advs.202408617

[17] K. Pajunen, et al., Design and impact response of 3D-printable tensegrity-inspired structures, Mater. Des. 182 (Nov. 2019) 107966. ISSN: 02641275, https://doi.org/10.1016/j.matdes.2019.107966

[18] P. Zhang, J. Feng, Initial prestress design and optimization of tensegrity systems based on symmetry and stiffness, Int. J. Solids Struct. 106–107 (Feb. 2017) 68–90. ISSN: 00207683, https://doi.org/10.1016/j.ijsolstr.2016.11.030

[19] Y. Chen, J. Feng, Initial prestress distribution and natural vibration analysis of tensegrity structures based on group theory, Int. J. Struct. Stab. Dyn. 12 (2) (Mar. 2012) 213–231. ISSN: 0219-4554, https://doi.org/10.1142/S0219455412500010

[20] B.H. Kim, T. Park, Estimation of cable tension force using the frequency-based system identification method, J. Sound Vib. 304 (3–5) (Jul. 2007) 660–676. ISSN: 0022460X, https://doi.org/10.1016/j.jsv.2007.03.012

[21] A.A. Sabouni-Zawadzka, et al., Design and fabrication of modular tensegrity-like lattices with auxetic properties, Mater. Des. 258 (Oct. 2025) 114513. ISSN: 02641275, https://doi.org/10.1016/j.matdes.2025.114513

[22] T. Masmeijer, et al., ThijsIllimited/printing-tensioned-structures: code release accompanying Masmeijer et al., 2025. Version v1.0, Nov. 2025, https://doi.org/10.5281/zenodo.17555002

[23] H. Denning, A. Bullinger, J.R. Hill, Tuning tensegrity: simplified construction and cable tension interactions, in: American Society of Mechanical Engineers, Sept. 2024, https://doi.org/10.1115/SMASIS2024-140381. ISBN: 978-0-7918-8832-2.

[24] T. Siegmund, et al., Manufacture and mechanics of topologically interlocked material assemblies, Appl. Mech. Rev. 68 (4) (2016) 040803.

[25] T. Rhodes, C. Gotberg, V. Vikas, Compact shape morphing tensegrity robots capable of locomotion, Front. Robot. AI 6 (2019) 2296–9144, https://doi.org/10.3389/frobt.2019.00111

[26] P. Wiersinga, et al., Hybrid compliant musculoskeletal system for fast actuation in robots, Micromachines 13 (10) (2022) 1783.

[27] K. Pajunen, P. Celli, C. Daraio, Prestrain-induced bandgap tuning in 3D-printed tensegrity-inspired lattice structures, Extrem. Mech. Lett. 44 (Apr. 2021) 101236. ISSN: 2352-4316, https://doi.org/10.1016/J.EML.2021.101236

[28] H. Lee, et al., 3D-printed programmable tensegrity for soft robotics, Sci. Robot. 5 (Aug. 2020). ISSN: 2470-9476, https://doi.org/10.1126/scirobotics.aay9024

[29] Y. Luo, et al., Digital fabrication of pneumatic actuators with integrated sensing by machine knitting, ACM, Apr. 2022, pp. 1–13, https://doi.org/10.1145/3491102.3517577. ISBN: 9781450391573.

[30] V. Sanchez, et al., 3D knitting for pneumatic soft robotics, Adv. Funct. Mater. 33 (26) (Jun. 2023). ISSN: 1616-301X, https://doi.org/10.1002/adfm.202212541

[31] A.A. Sabouni-Zawadzka, et al., Stability of tensegrity-inspired structures fabricated through additive manufacturing, Composite Struct. 345 (Oct. 2024) 118377. ISSN: 02638223, https://doi.org/10.1016/j.compstruct.2024.118377

[32] H. Lee, et al., 3D-printed programmable tensegrity for soft robotics, Sci. Robot. 5 (Aug. 2020). ISSN: 2470-9476, https://doi.org/10.1126/scirobotics.aay9024

[33] E. Wirth, G.B. Friedrich, Forces in the spider orb web, J. Comp. Physiol. A 171 (3) (Oct. 1992) 359–371. ISSN: 0340-7594, https://doi.org/10.1007/BF00223966

[34] S. Zschokke, Early stages of orb web construction in Araneus diadematus Clerck, Rev. Suisse Zool. Ann. Soc. Zool. Suisse Mus. Hist. Nat. Genève (Jan. 1996) 709–720.

[35] R. Ramousse, F. Davis, Web-building time in a spider: preliminary applications of ultrasonic detection, Physiol. Behav. 17 (6) (Dec. 1976) 997–1000. ISSN: 00319384, https://doi.org/10.1016/0031-9384(76)90020-2

[36] S. Zschokke, Nomenclature of the orb-web, J. Arachnol. 27 (2) (1999) 542–546.

[37] C.V. Crickard, et al., Analysis and comparison of the biomechanical properties of univalved and bivalved cast models, J. Pediatr. Orthop. 31 (1) (Jan. 2011) 39–43. ISSN: 0271-6798, https://doi.org/10.1097/BPO.0b013e318202c446

[38] M. Halanski, J.N. Kenneth, Cast and splint immobilization: complications, J. Am. Acad. Orthop. Surg. 16 (1) (Jan. 2008) 30–40. ISSN: 1067-151X, https://doi.org/10.5435/00124635-200801000-00005

[39] Y. Mao, S.K. Agrawal, Design of a Cable-Driven Arm Exoskeleton (CAREX) for neural rehabilitation, IEEE Trans. Robot. 28 (4) (Aug. 2012) 922–931. ISSN: 1552-3098, https://doi.org/10.1109/TRO.2012.2189496

[40] A. Moffat, et al., Relaxation behavior of cerclage cables and its effect on bone clamping force, Bioengineering 11 (12) (2024) 2306–5354, https://doi.org/10.3390/bioengineering11121289

[41] G. Scetta, et al., Cyclic fatigue failure of TPU using a crack propagation approach, Polym. Test. 97 (May 2021) 107140, https://doi.org/10.1016/j.polymertesting.2021.107140

[42] A.G. Tibert, S. Pellegrino, Deployable tensegrity reflectors for small satellites, J. Spacecr. Rockets 39 (5) (Sept. 2002) 701–709. ISSN: 0022-4650, https://doi.org/10.2514/2.3867

[43] A. ap Rhisiart, F. Vollrath, Design features of the orb web of the spider, Araneus diadematus, Behav. Ecol. 5 (3) (1994) 280–287. ISSN: 1045-2249, https://doi.org/10.1093/beheco/5.3.280

[44] F. Vollrath, M. Downes, S. Krackow, Design variability in web geometry of an orb-weaving spider, Physiol. Behav. 62 (4) (Oct. 1997) 735–743. ISSN: 00319384, https://doi.org/10.1016/S0031-9384(97)00186-8

[45] S. Isabelle, et al., In situ three-dimensional spider web construction and mechanics, Proc. Natl. Acad. Sci., 118(33) (Aug. 2021), ISSN: 0027-8424, https://doi.org/10.1073/pnas.2101296118.

[46] E.L. Buehler, S. Isabelle, M.J. Buehler, WebNet: a biomateriomic three-dimensional spider web neural net, Ext. Mech. Lett. 42 (Jan. 2021) 101034. ISSN: 23524316, https://doi.org/10.1016/j.eml.2020.101034

[47] A. Pugh, Introduction to Tensegrity, University of California Press, en. Berkeley, CA, Nov. 1976.

[48] A.G. Tibert, S. Pellegrino, Review of form-finding methods for tensegrity structures, Int. J. Space Struct. 18 (4) (Dec. 2003) 209–223. ISSN: 0956-0599, https://doi.org/10.1260/026635103322987940

[49] K. Koohestani, Form-finding of tensegrity structures via genetic algorithm, Int. J. Solids Struct. 49 (5) (Mar. 2012) 739–747. ISSN: 00207683, https://doi.org/10.1016/j.ijsolstr.2011.11.015

[50] H.C. Tran, J. Lee, Advanced form-finding of tensegrity structures, Comput. Struct. 88 (3–4 Feb. 2010) 237–246. ISSN: 00457949, https://doi.org/10.1016/j.compstruc.2009.10.006

[51] T.C.P. Masmeijer, et al., Effect of eccentricity on sensing in spider web inspired cable nets, J. Phys. Conf. Ser. 2647(19) (2024) 192012. IOP Publishing.

[52] R.M.O. Pauletti, F.L. Fernandes, An outline of the natural force density method and its extension to quadrilateral elements, Int. J. Solids Struct. 185 (2020) 423–438.

[53] H.-J. Schek, The force density method for form finding and computation of general networks, Comput. Methods Appl. Mech. Eng. 3 (1) (Jan. 1974) 115–134. ISSN: 00457825, https://doi.org/10.1016/0045-7825(74)90045-0

[54] T. Méndez Echenagucia, et al., A cable-net and fabric formwork system for the construction of concrete shells: design, fabrication and construction of a full scale prototype, Structures 18 (Apr. 2019) 72–82. ISSN: 23520124, https://doi.org/10.1016/j.istruc.2018.10.004

[55] C. Zhu, et al., Algorithm 778: L-BFGS-B, ACM Trans. Math. Softw. 23 (4) (Dec. 1997) 550–560. ISSN: 0098-3500, https://doi.org/10.1145/279232.279236

[56] A. Legarra, I. Misztal, Technical note: computing strategies in genome-wide selection, J. Dairy Sci. 91 (1) (Jan. 2008) 360–366. ISSN: 00220302, https://doi.org/10.3168/jds.2007-0403

[57] F. Botler, R. Cano, M. Sambinelli, On computing the path number of a graph, Electr. Notes Theor. Comput. Sci. 346 (Aug. 2019) 185–197. ISSN: 15710661, https://doi.org/10.1016/j.entcs.2019.08.017

[58] J. Weiner, et al., Solving the maximum edge disjoint path problem using a modified Lagrangian particle swarm optimisation hybrid, Eur. J. Oper. Res. 293 (3) (Sept. 2021) 847–862. ISSN: 03772217, https://doi.org/10.1016/j.ejor.2021.01.009

[59] B. Lévy, et al., Least squares conformal maps for automatic texture atlas generation, ACM Trans. Graph. 21 (3) (Jul. 2002) 362–371. ISSN: 0730-0301, https://doi.org/10.1145/566654.566590

[60] H. Dal, O. Gültekin, K. Açkgöz, An extended eight-chain model for hyperelastic and finite viscoelastic response of rubberlike materials: theory, experiments and numerical aspects, J. Mech. Phys. Solids 145 (2020) 104159.

[61] R. Durna, et al., Hyper-Data: a Matlab based optimization software for data-driven hyperelasticity, SoftwareX 26 (2024) 101642.