Shaping Tension Structures with Actively Bent Linear Elements

by

Tom Van Mele, Lars De Laet, Diederik Veenendaal1, Marijke Mollaert and Philippe Block

Reprinted from

INTERNATIONAL JOURNAL OF
SPACE STRUCTURES
Volume 28 · Number 3 & 4 · 2013

MULTI-SCIENCE PUBLISHING CO. LTD.
5 Wates Way, Brentwood, Essex CM15 9TB, United Kingdom

Shaping Tension Structures with Actively Bent Linear Elements

Tom Van Mele[1,*], Lars De Laet[2], Diederik Veenendaal[1], Marijke Mollaert[2] and Philippe Block[1]

[1]BLOCK Research Group, Institute of Technology in Architecture, ETH Zurich. Wolfgang-Pauli-Strasse 15, 8093 Zurich, Switzerland. veenendaal@arch.ethz.ch, block@arch.ethz.ch
[2]Lightweight Structures Lab, Department of Architectural Engineering, Vrije Universiteit Brussel. Pleinlaan 2, 1050 Brussels, Belgium.
lars.de.laet@vub.ac.be, mmollaert@vub.ac.be

(Submitted on 19/02/2013, Reception of revised paper 02/07/2013, Accepted on 20/09/2013)

ABSTRACT: To achieve sufficient anticlastic (negative) curvature, membrane structures are tensioned between high and low anchor points, attached to the ground, buildings or poles. By integrating flexible bending elements in the membrane surface, an internal support and shape-defining system is created that provides more freedom in design and allows reducing the amount of external supports compared to traditional membrane structures.

This paper presents a computational framework for form finding of tension structures with integrated, elastically bent, linear elements, based on three-dimensional bending moment vectors and a mixed force density formulation. With an implementation of this framework in CAD modelling software, users can control form and forces by prescribing any combination of force densities, forces, stiffness or lengths to the spline and cable-net elements. Sparse matrix operations are used to compute the resulting equilibrium shapes.

The shape-defining possibilities of integrating 'bending-active' elements in tension structures are demonstrated through a series of design studies with various boundary conditions and spline configurations. The presented framework and implementation provide a straightforward method for the design of this hybrid structural system, and, therefore, facilitate its further exploration.

1. INTRODUCTION

Bending-active beam or surface structures derive their geometry from the elastic deformation of initially straight or planar elements [1]. The elastically bent elements, also called 'spline' elements, interact with other connecting elements that prevent them from taking their original shape, and thereby create a stable (prestressed) structural system. The advantage of using straight or planar elements is that they can be easily produced and transported in a compact, flat-packed configuration.

In a tension structure, flexible bending elements can be used both as support structure and as a shape-defining system for the membrane surface or cable-net. Thanks to the stabilising effect of the tension components, very slender elements can be used. The shape of such a 'bending-active' tension structure is entirely dependent on the interaction between its bent and tensioned components. Therefore, it has to be determined through a structural form-finding process.

Special-purpose tools for form finding of membrane or cable-net structures, such as EASY [2], typically only allow axial force members to be included in the form finding process. Although it is theoretically possible to combine these elements into a

*Corresponding author e-mail: van.mele@arch.ethz.ch

bending-stiff supporting structure (for example, by creating a spatial truss) the process is extremely tedious and becomes almost impossible when the number of bending elements and complexity of their configuration increases. With traditional FEM software, the form-finding process of a (bending-active tension) structure can be simulated by iteratively updating the structure's geometry based on the results of consecutive (static) analyses. However, this procedure is complicated and time-consuming, and successful completion relies heavily on the chosen starting geometry.

With tools such as Kangaroo [3], combined form finding of bending and axial force members is possible. Bending-active elements require the input of a 'bending strength', and a cable-net can be modelled as a network of springs governed by a 'stiffness' and 'rest length'. Since these are not traditional parameters in structural analysis to describe an elastically deformable structure, controlling and interpreting the final equilibrium shape in a structurally meaningful way is not straightforward.

In this paper, we present a computational framework for integrated form finding of bending-active tension structures based on three-dimensional bending moment vectors and a mixed force density formulation. We discuss its implementation in CAD modelling software (Rhinoceros [4]) and, using the implemented, interactive tool, demonstrate some of the design possibilities of integrating bending-active elements in tension structures through a series of examples.

2. PREVIOUS WORK

Based on the dynamic relaxation method, introduced by Day [5], Barnes [6] developed a method for the computation of the shape of equilibrium of spline elements, elastically bent from an initially straight, unstressed state. In this seminal paper, he furthermore described the use of spline elements in gridshells and as support for membrane structures.

For small- to medium-span structures, the splines can be very slender and flexible, because they are stabilised by the action of the membrane. Adriaenssens and Barnes [7] proposed a tensegrity bracing system for splines for larger-scale membrane structures, where the action of the membrane alone is insufficient. Adriaenssens [8] furthermore investigated the feasibility of using spliced, spline beams in medium-span membrane systems. These spliced elements are discontinuous and do not have any torsional stiffness. Also building upon the work of Barnes and the dynamic relaxation method, Douthe *et al*. [9] describe the form finding of gridshells that result from bending slender tubes in composite materials.

A different approach to the form finding of membrane structures with an actively bent support structure can be found in the work of Lienhard and Knippers [10]. They describe the calculation of a funnel-shaped membrane roof for which they used an iterative approach using large-displacement finite element modelling.

Flexible bending elements can also be applied in a membrane structure to enlarge the covering area by including the elements in the membrane surface, connecting opposite boundary cables, as presented by Off [11]. Other case studies, investigated by Alpermann and Gengnagel [12], describe the structural combination of the bending and membrane elements on the level of separate building components, such as membrane-restrained arches and columns.

An extensive review of spatial structures where bending is used in a shape-defining way is given by Lienhard *et al*. [13].

3. COMPUTATIONAL FRAMEWORK

The bending-active tension structures investigated in this paper consist of cables and cable-nets supported by elastically bent, linear splines. Multiple splines can be included and connected to each other. Spline-cable-net and spline-spline connections are such that no eccentric forces are applied to the splines. Clamped support connections are only kinematically restrained (see Section 4.1). The splines are initially straight and have round (tubular) sections. Therefore, the splines are only subjected to shear and axial forces, even when bent out of plane.

3.1. Dynamic relaxation

Form finding starts from an initial, arbitrary configuration, which is relaxed to a stable shape using a dynamic relaxation algorithm [6, 9]. Essentially, this is an iterative procedure in which the nodes of the system move under influence of residual force vectors until equilibrium (or the maximum number of iterations) is reached. The residual forces, $\mathbf{F_r}$, are the result of an imbalance at the free nodes between the loads, $\mathbf{P}$, forces due to bending in the splines, $\mathbf{F_b}$, and the axial forces, $\mathbf{F_a}$, in all other elements of the structure:

$$\mathbf{F_r} = \mathbf{P} + \mathbf{F_b} - \mathbf{F_a}. \quad (1)$$

In the following two sections, we will discuss how the forces due to bending and the axial forces are

calculated at each iteration. In order to be able to perform these calculations using matrix operations, rather than using a node-per-node approach, we use branch-node matrices to represent the connectivity of the structure. The connectivity of the entire network is represented by a typical $[m \times n]$ branch-node matrix $\mathbf{C}$ [14–15], with m the number of branches in the network (including the branches that belong to the splines), and n the total number of vertices. For each spline, an $[m_{spl} \times n]$ branch-node matrix $\mathbf{C}_{spl}$ identifies the spline branches, with m_{spl} the number of branches in the spline. As each $\mathbf{C}_{spl}$ has n columns, corresponding to the n nodes of the entire system, compatibility between the network and spline matrices is guaranteed. All branch-node matrices are partitioned column-wise such that the first n_i columns correspond to the free nodes and the remaining columns to the fixed/support nodes:

$$\mathbf{C} = [\mathbf{C}_i \,|\, \mathbf{C}_f] \tag{2}$$

$$\mathbf{C}_{spl} = \left[\mathbf{C}_{spl,i} \,|\, \mathbf{C}_{spl,f}\right] \quad \text{with} \quad spl = 1 \rightarrow s. \tag{3}$$

3.2. Nodal forces due to bending

Nodal forces due to bending in (1) are calculated based on three-dimensional bending moment vectors. At any point along its length, a bent spline is subjected to a bending moment around an axis perpendicular to the local osculating plane.

In a discretised spline, assuming i, j and k to be consecutive points of the discretisation, as seen in Figure 1, the curvature vector at node j is

$$\mathbf{r}_{spl,j} = \frac{\left(\|\mathbf{a}\|^2 \mathbf{b} - \|\mathbf{b}\|^2 \mathbf{a}\right) \times (\mathbf{a} \times \mathbf{b})}{2\|\mathbf{a} \times \mathbf{b}\|^2}, \tag{4}$$

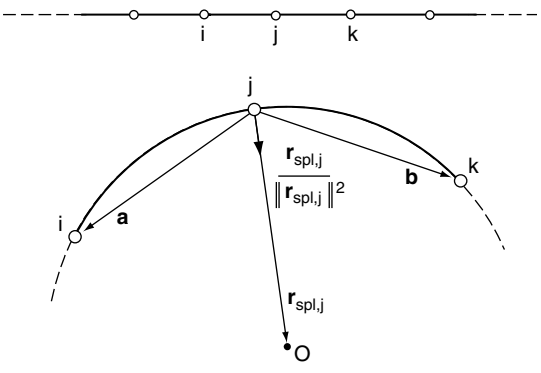

Figure 1. The curvature vector $\mathbf{r}_{spl,j}$ at node j of a discretised spline.

with $\mathbf{a}$ the vector defined from j to i and $\mathbf{b}$ the vector from j to k. The bending moment vector at j is

$$\mathbf{m}_{spl,j} = EI \frac{\mathbf{r}_{spl,j}}{\left\|\mathbf{r}_{spl,j}\right\|^2}. \tag{5}$$

The $[m_{spl} \times 3]$ matrix of bending moment difference vectors of the branches of the entire spline is then

$$\Delta\mathbf{M}_{spl} = \mathbf{C}_{spl} \cdot \mathbf{M}_{spl}, \tag{6}$$

with $\mathbf{C}_{spl}$ the $[m_{spl} \times n]$ branch-node matrix of the spline as in (3), and $\mathbf{M}_{spl}$ the $[n \times 3]$ matrix of bending moment vectors of the spline. The $[m_{spl} \times 3]$ matrix of shear forces in the branches of the spline is therefore

$$\mathbf{S}_{spl} = \mathbf{L}_{spl}^{-1} \cdot \Delta\mathbf{M}_{spl}, \tag{7}$$

with $\mathbf{L}_{spl}$ an $[m_{spl} \times m_{spl}]$ diagonal matrix containing the spline branch lengths on its diagonal.

Using (6) and (7), and combining the branch forces at the nodes using the spline's branch-node matrix, the $[n_i \times 3]$ matrix of nodal forces due to bending in the current spline is

$$\mathbf{F}_{b,spl} = \mathbf{C}_{spl,i}^{T} \cdot \mathbf{L}_{spl}^{-1} \cdot \mathbf{C}_{spl} \cdot \mathbf{M}_{spl}. \tag{8}$$

Adding up all $\mathbf{F}_{b,spl}$ then provides us with the $[n_i \times 3]$ matrix $\mathbf{F}_b$ of nodal forces due to bending in the system:

$$\mathbf{F}_b = \sum_{spl} \mathbf{F}_{b,spl}. \tag{9}$$

3.3. Nodal forces due to axial forces

Nodal forces due to the axial forces in (1) are calculated based on a mixed force density formulation. The vector $\mathbf{q}$ of force densities is

$$\begin{aligned} \mathbf{q} &= \mathbf{q}_{pre} + \mathbf{q}_{l,pre} + \mathbf{q}_{f,pre} + \mathbf{q}_{EA} \\ &= \mathbf{q}_{pre} + \frac{\mathbf{f}}{\mathbf{l}_{pre}} + \frac{\mathbf{f}_{pre}}{\mathbf{l}} + \frac{\mathbf{EA}}{\mathbf{l}} \frac{\mathbf{l} - \mathbf{l_0}}{\mathbf{l_0}}, \end{aligned} \tag{10}$$

where $\mathbf{l}$ and $\mathbf{f}$ are the current length and force vectors, $\mathbf{l}_0$ is the vector of initial lengths, and $\mathbf{q}_{pre}$, $\mathbf{l}_{pre}$, $\mathbf{f}_{pre}$ and $\mathbf{EA}$ the vectors of user-prescribed force densities, lengths, forces and axial stiffnesses, respectively.

Using (10), the nodal forces due to the branch forces in the system are therefore

$$\mathbf{F_a} = \mathbf{C_i^T} \cdot \mathbf{Q} \cdot \mathbf{C} \cdot \mathbf{V}, \qquad (11)$$

where $\mathbf{Q}$ is an $[m \times m]$ diagonal matrix with on its diagonal the force densities $\mathbf{q}$, and $\mathbf{V}$ is the $[n \times 3]$ matrix of vertex coordinates.

4. IMPLEMENTATION

In this section, we discuss the implementation of the computational framework in Rhinoceros [4], using the Python scripting language [16]. With this implementation, users can interactively steer the design of a bending-active tension structure by controlling its boundary conditions, spline geometry and stiffness, cable and cable-net force densities, and solver settings.

4.1. Input

The only required input is a network of connected lines from which the topology of the structure can be determined. The topological information is stored in a halfedge datastructure [17–18], which is an edge-centred datastructure capable of maintaining incidence information on vertices, edges and faces. This provides a robust basis for keeping track of all vertex and edge properties, and allows vertices and edges to be added or removed, and splines to be (re-)defined without reprocessing the input. Furthermore, the branch-node matrices required for structural calculations can be easily and efficiently derived from this data structure. By default, the geometry of the connected lines is used as starting geometry, but these starting values can be overwritten by the user.

Figure 2 and Figure 3 depict typical input networks and their corresponding equilibrium shapes. In both cases, the splines and anchor points have already been identified and are marked in blue. The first network is a cable-net supported by a spline arch. The spline has a circular starting geometry. Note that this is not a requirement, though; the user can simply draw any geometry and specify the initial lengths of the spline segments numerically.

The second configuration consists of a cable-net attached to a cantilever spline. The clamped support is modelled by applying a kinematic constraint to two consecutive nodes at one end of the spline, as seen in Figure 3. Note that this type of connection still allows the spline to rotate around its length axis while keeping its inclination (with respect to the ground) fixed. One can think of this as a rod inserted in a fixed tube in the ground; the rod is inserted such that it cannot be pulled out, but it is free to rotate around its axis.

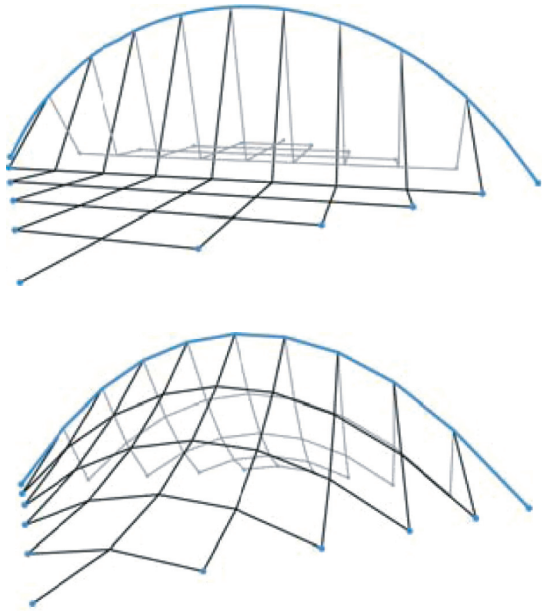

Figure 2. An arch-supported, bending-active tension structure: the input network and identification of the spline branches and anchor points (top), and shape of equilibrium (bottom).

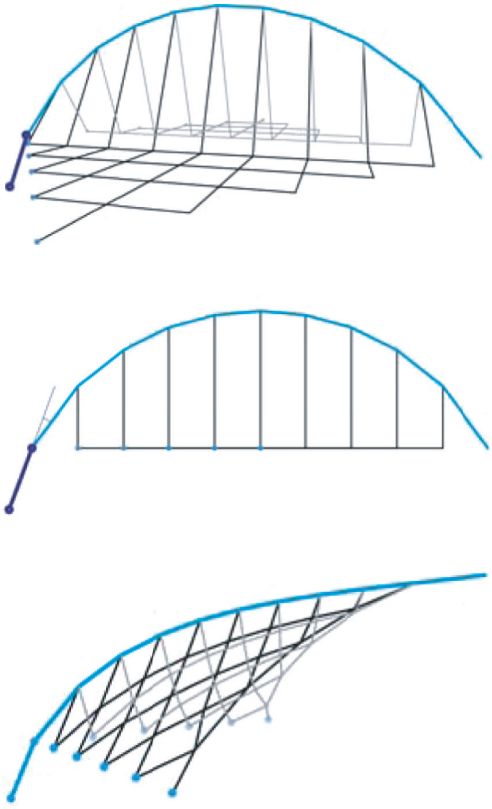

Figure 3. A cable-net attached to a cantilevering bending element (bottom): the fixed connection at the base of the spline is modelled by applying kinematic constraints to two consecutive points (top and middle).

4.2. Solving

The dynamic relaxation algorithm is implemented using an explicit Runge-Kutta (RK) formulation for which the user can choose the order. Time step, dt, damping (kinetic or viscous), maximum number of iterations, k_{max}, and convergence tolerances, ε, are also user-defined parameters. The user can choose to evaluate convergence based on the norm of residual forces or on the norm of nodal displacements, or on both. Separate tolerances for the norm of residual forces can be specified for the splines (ε_{spl}) and the rest of the structure (ε_{net}). The default values are summarised in Table 1.

Since the branch-node matrices are extremely sparse, especially those of the splines, all matrix calculations are performed using the sparse algorithms of SciPy [19]. This improves solving times and allows larger networks to be handled more efficiently. Since the sparse algorithms of SciPy are not available in IronPython [20] (i.e. the .Net implementation of Python used in Rhinoceros), the equilibrium calculations are run in a separate CPython [16] process using the subprocess module.

4.3. Output

During the iterative procedure, the geometry in Rhino is updated in real time, while the evolution of the convergence criteria is plotted live in a separate graph window. When the procedure is complete, all element forces, reaction forces and residual forces can be visualised (Figure 4). This information allows the user to evaluate the generated shape both geometrically and structurally, change attributes if so desired and continue the form exploration.

4.4. User interaction

The user can stop the solving procedure at any time; for example, if the shape of the structure is not evolving in the way that was expected from the chosen parameters. All element properties, solver settings, geometrical data, spline definitions, nodal restraints and point loads can be changed by the user through a custom toolbar in the Rhinoceros interface.

The mixed force-density formulation allows the user to steer the equilibrium exploration in a very flexible and versatile manner. For example, in a single structure, the equilibrium of sets of branches intended as boundary cables can be force-controlled ($\mathbf{q} = \mathbf{q}_{f,pre}$); external cables tying, for example, the end of a cantilevering spline to the ground, can be length-controlled ($\mathbf{q} = \mathbf{q}_{l,pre}$); and, branches belonging to a spline can be stiffness-controlled ($\mathbf{q} = \mathbf{q}_{EA}$); while the branches of the network representing the inner part of the cable-net can be simply force density-controlled ($\mathbf{q} = \mathbf{q}_{pre}$).

Table 1. Default solving parameters

RK	dt	damping	k_{max}	ε_{spl}	ε_{net}	ε_{disp}
1	1	viscous	100000	0.01	0.0001	0.00001

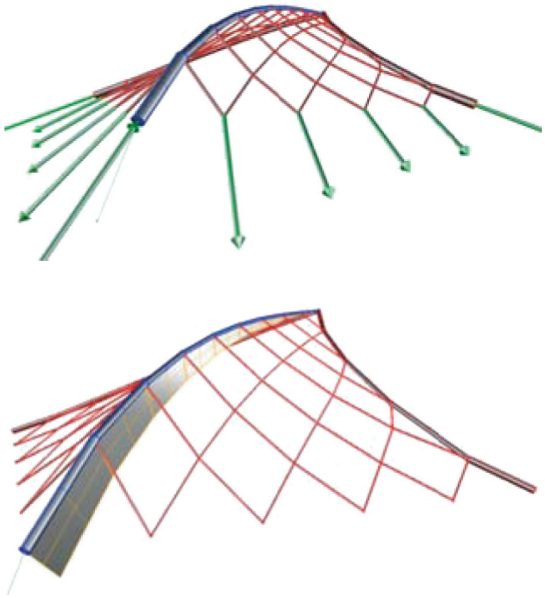

Figure 4. Graphical representation of axial forces (red tension, blue compression), reaction forces (top), and 3D bending moments (bottom).

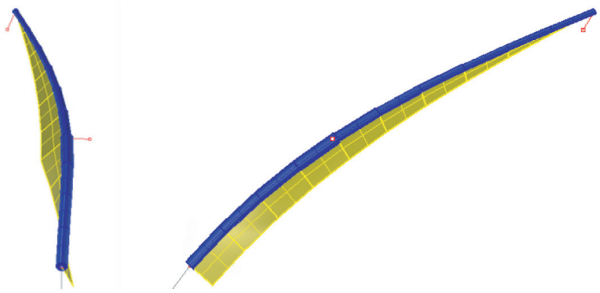

Figure 5. A cantilevered element is bent in 3D by pulling its midpoint and tip in different directions: front and side view. The 3D bending moment is visualised graphically (yellow).

The authors found the use of length-controlled, external cables the most direct and straightforward way of influencing the final equilibrium shape. The spatially curved cantilever and the nerve in Figure 5 and Figure 10, respectively, were generated using length-controlled external cables.

5. VERIFICATION

To verify the proposed three-dimensional bending moment vector approach, the results of a spatially curved cantilever, generated with the Rhino implementation,

were compared with the outcome of calculations conducted with the FE analysis software ANSYS [21].

In both tools, the spline element was fixed to the ground by applying kinematic restraints to two consecutive nodes, as described in Section 4.1 and illustrated in Figure 3. The spline element has a cross section with a radius of 0.01 m and stiffness 30 GPa. In the Rhino implementation, the bending element was pulled in different directions using cable elements, attached to its midpoint and tip (Figure 5). In ANSYS, the spline was modelled with the PIPE16 element, and the curved shape was generated by applying external forces identical to the action of the cables at the same locations.

Figure 6 shows a comparison of the equilibrium shapes and bending moments generated with both tools. The x-axes of both graphs contain the numbering of the nodes (0 being the fixed node and 10 the tip). Node 4 is the node in the middle at which the spline is pulled to the right. Node 10 is pulled down and to the left.

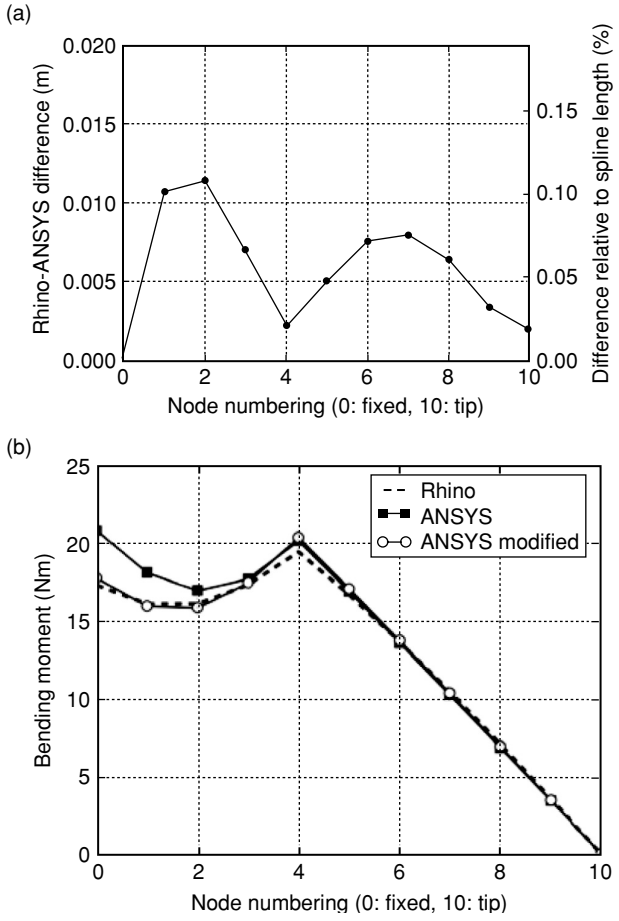

Figure 6. Comparison of the equilibrium shapes (top) and bending moments (bottom) generated with the Rhino implementation and ANSYS.

The y-axis in Figure 6.a shows the distance between the corresponding nodes of both equilibrium shapes. The graph shows that the largest difference between the two shapes measures 0.011 m, or, relative to the spline's length (10.54 m), only 0.1%, approximately.

Figure 6.b shows a discrepancy of approximately 3.5 Nm between the bending moments at the fixed support, with ANSYS generating a higher bending moment (16.8%) than our implementation. Despite identical kinematic constraints and similar equilibrium shapes, the difference in bending moments at the supports is significant. On closer inspection, it turns out that ANSYS, despite not calculating translations at the kinematically restrained nodes, does calculate rotations along the straight PIPE16 element between them. The rotations vary linearly along the element, being zero at one third of its length. If we extend the bottom element downwards by approximately 50%, such that the point of zero rotation occurs at the location of the bottom fixed node in our own implementation, agreement between our results and the ANSYS results improves considerably (2.5% at support, 4.5% at the node 4) as seen in Figure 6.

6. DESIGN EXAMPLES

In this section, we demonstrate some of the design possibilities with bending-active tension structures by means of four examples.

All structures have only kinematic restraints at the supports. All spline elements were initially straight and have round (tubular) sections with a diameter of 0.02 m. A material with Young's modulus 30 GPa was used. Unless mentioned otherwise, edges of the inner cable-nets have a prescribed force density of $q_{pre} = 1$; and the edges forming boundary cables have a prescribed value of $q_{pre} = 10$.

The first structure consists of three cantilevering bending elements, positioned alternatingly at opposite sides of the structure (Figure 7). The resulting equilibrium shape looks like a concatenation of modular four-point hyperbolic paraboloids (hypars), but only requiring anchors at ground level.

The second case is shown in Figure 8. The bending elements are oriented towards the middle of the structure and the tension elements wrapped around them. Opposite bending elements have the same stiffness, and one pair is three times stiffer than the other. The boundary cables have a prescribed force density of 5.

Figure 9 illustrates the combination of two elastically bent arches with two suspended bending

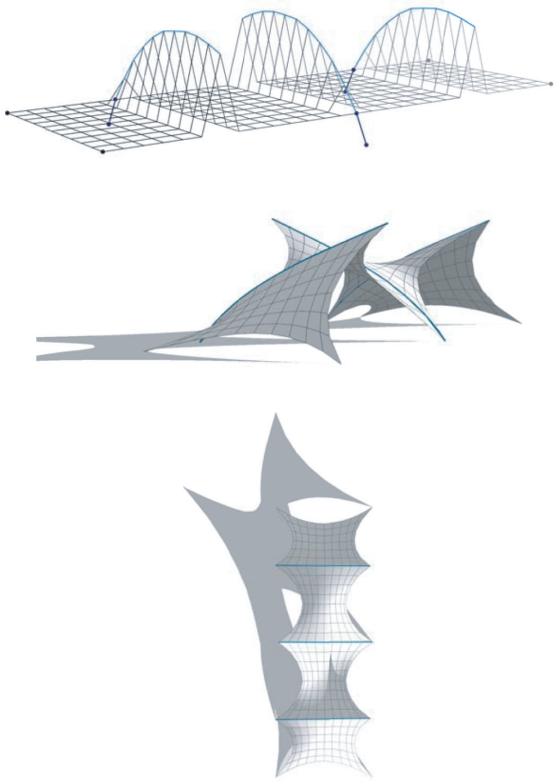

Figure 7. Multiple cantilevering bending elements, positioned alternatingly at the opposite side of the structure, generate a concatenation of four-point-hypar-like modules.

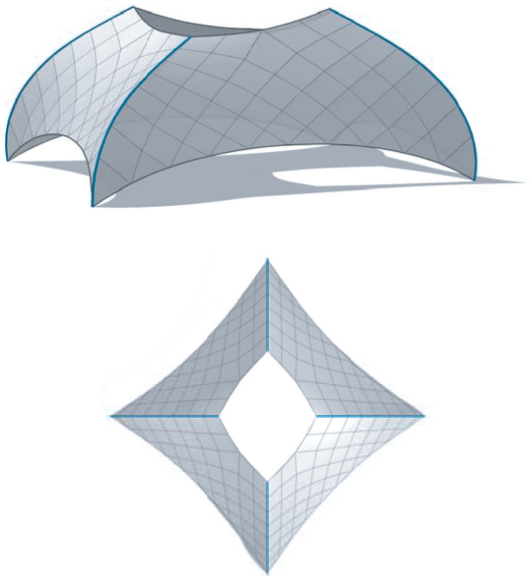

Figure 8. Bending the cantilevering, linear elements towards the middle and wrapping the tension elements around them generates a tent-like structure.

elements. The suspended elements create internal 'high' points and thus provide the structure with openings without any additional, external structural

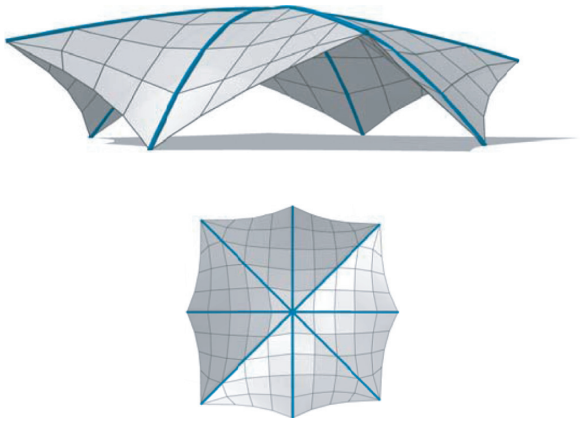

Figure 9. Combining two elastically bent arches with two integrated suspended bending elements creates openings without additional structural elements.

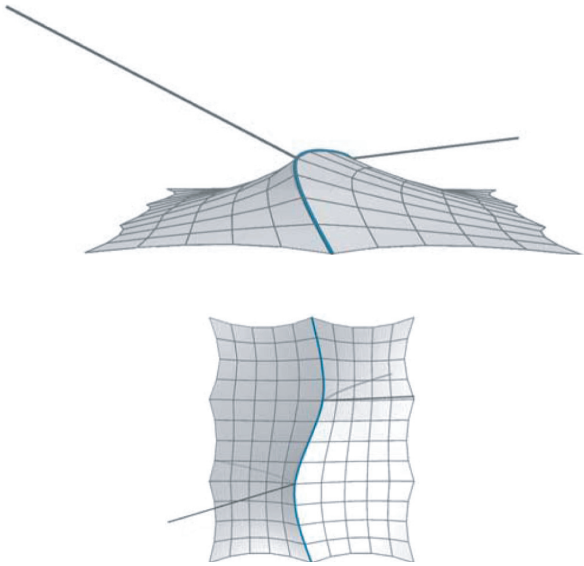

Figure 10. A bending element suspended by two cables connected, for example, to an adjacent building. The suspension cables pull the spline in opposite directions, creating a spatially curved arch.

element. The boundary cables have a prescribed force density of 5.

The last structure is a cable-net roof supported by a central arch. The arch is pulled in opposite directions by two suspension cables to generate the spatial curve of the arch. The cables on the left and right have a prescribed length of 7.8 m and 4.8 m, respectively. One can imagine these cables to be connected to an adjacent building.

Table 2 summarises the numbers of edges, m, nodes, n, and fixed nodes, n_f, and provides the required number of iterations and solving times for the two convergence strategies discussed in Section 4.2.

In all cases, convergence was significantly faster when using the displacement criterion. However,

Table 2. Overview design examples. m is the number of edges, n the number of vertices, n_f the number of fixed vertices, s the number of splines. The number of iterations and solving times are given for the two convergence strategies. Times are in seconds

ex.	*m*	*n*	n_f	*s*	k_{force}	t_{force}	k_{displ}	t_{displ}
1	621	322	7	3	103600	315	21600	63
2	420	216	4	4	158400	478	21000	62
3	230	117	4	4	309300	878	11200	32
4	274	145	12	1	314300	474	8100	12

with this criterion, all structures still had small residual forces at the nodes of the splines. Nevertheless, the geometry of the structure and the internal forces were virtually identical to the results produced with the force criterion. Furthermore, the residuals were very small compared to the internal forces.

One of the reasons for the difference in solving time seems to be that when the residual forces become very small, they cause very little movement of the nodes. Therefore, little change occurs, even over a large number of iterations. Furthermore, the changes that do occur have very little influence on the final result. Therefore, the force criterion seems unnecessarily strict, especially for form finding purposes.

7. CONCLUSIONS

Integrating bending-active elements is an interesting and versatile way to support and shape tension structures. Various configurations and applications of this hybrid structural system are possible. To allow interactive exploration of the design possibilities, a computational framework for form finding of tension structures with integrated, linear, actively bent elements has been developed. The use of a halfedge datastructure provides a robust and flexible infrastructure for management and manipulation of network topology and spline definitions, and of all vertex, edge and spline properties.

The use of the mixed force density formulation allows the user to interactively steer the design by controlling form and forces through prescribed force densities, forces, stiffnesses and/or lengths, in addition to boundary conditions, spline geometry and stiffness and solver settings.

Although this paper focussed on form finding of a particular type of bending-active tension structures, the framework and implementation can also be used for gridshells or combinations of gridshells and membranes.

The presented tool generates equilibrium shapes based on nodal equilibrium, without taking into account allowable stresses and deformations, or other structural constraints. Future development of the tool will therefore focus on the integration of such constraints in the form finding process, and on the addition of a module for static analyses of the obtained shapes.

REFERENCES

[1] J. Knippers, J. Cremers, M. Gabler, and J. Lienhard, "Construction Manual for Polymers + Membranes," Munchen, 2011.

[2] Easy. [Online]. *http://www.technet-gmbh.com/index.php?id=63&L=1*

[3] Kangaroo. [Online]. *http://www.grasshopper3d.com/group/kangaroo*

[4] Rhinoceros. [Online]. *http://www.rhino3d.com*

[5] A.S. Day, "An introduction to Dynamic Relaxation," *The Engineer*, pp. 218–221, January 1965.

[6] M.R. Barnes, "Form Finding and Analysis of Tension Structures by Dynamic Relaxation," *International Journal of Space Structures*, pp. 14(2), 89–104, 1999.

[7] S.M.L. Adriaenssens and M.R. Barnes, "Tensegrity Spline Beam and Grid Shell Structures," *Engineering Structures*, pp. 23, 29–36, 2001.

[8] S. Adriaenssens, "Feasibility Study of Medium Span Spliced Spline Stressed Membranes," *International Journal of Space Structures*, pp. 23(4), 243–251, 2008.

[9] C. Douthe, O. Baverel, and J.-F. Caron, "Form-finding of a Grid Shell in Composite Materials," *Journal of the International Association of Shell and Spatial Structures*, pp. 47(1), 53–62, 2006.

[10] J. Lienhard and J. Knippers, "Permanent and Convertible Membrane Structures with Intricate Bending-active Support Systems," in *Proceedings of the IASS symposium 2012*, Seoul, Korea, 2012.

[11] R. Off, "New trends on membrane and shell structures - examples of bat sail and cushion-belt technologies," in *Structures and Architecture – Cruz (Ed.)*. London: Taylor & Francis Group London, 2010.

[12] H. Alpermann and C. Gengnagel, "Shaping Actively-bent Elements by Restraining Systems," in *Conference Proceedings IASS- APCS Symposium: From Spatial Structures to Space Structures*, Seoul, Korea, 2012.

[13] J. Lienhard, H. Alpermann, C. Gengnagel, and J. Knippers, "Active Bending, a Review on structures

where bending is used as a self-forming process," in *Proceedings of the IASS symposium 2012*, Seoul, Korea, 2012.

[14] H.-J. Schek, "The Force Density Method for Form Finding and Computation of General Networks," *Computer Methods in Applied Mechanics and Engineering*, pp. 3, 115–134, 1974.

[15] D. Veenendaal and P. Block, "An overview and Comparison of Structural Form Finding Methods for General Networks," *International Journal of Solids and Structures*, pp. 49(26), 3741–3753, 2012.

[16] Python. [Online]. *http://www.python.org*

[17] L. Kettner, "Using generic programming for designing a data structure for polyhedral surfaces," *Computational Geometry: Theory and Applications*, no. 13, pp. 65–90, 1999.

[18] C.M. Eastman and S.F. Weiss, "Tree structures for high dimensionality nearest neighbor searching," *Information Systems*, vol. 7, no. 2, pp. 115–122, 1982.

[19] SciPy.org. [Online]. *http://www.scipy.org*

[20] IronPython. [Online]. *http://ironpython.net*

[21] ANSYS. [Online]. *http://www.ansys.com*