STAG: Smart Tools and Applications in Graphics (2022)
D. Cabiddu, T. Schneider, G. Cherchi, and R. Scateni (Editors)

A computational tool for the analysis of 3D bending-active structures based on the dynamic relaxation method

I. Manolas[†1,2], F. Laccone[1], G. Cherchi[3], L. Malomo[1], P. Cignoni[1]

[1]ISTI-CNR, Pisa, Italy, [2] University of Pisa, Italy, [3]University of Cagliari, Italy

Abstract

The use of elastic deformation of straight or flat structural components for achieving complex 3D shapes has acquired attention from recent computational design works, particularly in architectural geometry. The so-called bending-active structures are built by deforming and restraining the components mutually to form a stable configuration. While the manufacturing of components from flat raw material and their assembly are simple and inexpensive, the complexity lies in the design phase, in which computational tools are required to predict the deformation and forces under a prescribed form-finding load or displacement. Currently, there is a scarcity of open and efficient tools that hinder the design of bending-active structures. This paper proposes and validates an open-source computational tool for predicting the static equilibrium of general bending-active structures in the form of a network of elements using the dynamic relaxation method. We apply our tool to various real-world examples and compare the results to a commercial FEM solver. The proposed tool shows accuracy and good time performance, making it a significant addition to the available open-source structural engineering toolkit.

1. Introduction

Art, architecture, and fabrication research efforts are spent on automating design and processes. This transition is driven by algorithms that support the designer in making choices and exploring several options within the design domain, and communicating manufacturing instructions to computer numerical controlled machines.

Classic workflows, in which bespoke and sequential steps are applied to a specific case study or product, have been replaced by automated workflows based on generative design and *optioneering*, in which the singularity of the product and its customization are materialized through fabrication-aware techniques and generic methods of production. In these latter, the fabrication drastically influences the design and opens the way for new possibilities and aesthetic expression. At the same time, it raises novel needs for tools that can manage material usage and run physically plausible simulations.

To produce complex 3D objects and surfaces, the active-bending technique that has been known since ancient times has been revisited in light of the current trends. Elements and plates, which can be cut from flat sheets of material or printed on a plate, are elastically bent and then assembled to form a self-stable shape.

In this context, all the components undergo large deformations. Simulations are fundamental to check whether this deformation is feasible and to assess the accuracy of the assembly of reaching a target shape. The problem can be solved through sophisticated and accurate commercial FEM software packages; however, they are not available to the general user. Moreover, they are hard to use and customize, and are computationally intensive. Indeed, such software is used for detailed analyses and safety checks. Whereas, algorithms routines need efficient and customized tools.

The main contribution of this work is the implementation of an open-source, fast and accurate simulation tool that uses the Dynamic Relaxation Method (DRM) to solve large-deformation equilibrium problems. This tool targets networks of elements, which can form bending active structures of any shape. For speeding up the required computations we have reformulated the math in [SOA20]. Our implementation in C++ along with our code optimization discussed in section 4.4 is orders of magnitude faster, compared to the Matlab implementation of Sakai et al. of [SOA20]. After a brief research review, we introduce the theoretical background of the method. Then, we provide implementation details and optimization strategies. We use our tool to showcase several results obtained on real-world examples which we compare to a commercial FEM solver.

2. Related work

2.1. Bending-active structures

Realizing complex curved shapes is a challenging task in manufacturing processes. Several techniques can be used, such as molding, heat- or cold-bending, plastic deformation, etc. They are expensive

† Corresponding author

DOI: 10.2312/stag.20221250

since they require templates and formworks, which are not fully reusable, especially if the curvature varies punctually.

One of the most interesting and cheapest strategies to tackle these issues is represented by the bending-active technique [LAGK13]. Hence, the structure is segmented into components that are induced in a deformed state by bending and mutually combined to achieve a stable configuration. However, this process poses several limitations because the shape of these form-active structures depends on the amount of stress the elements can attain, which in turn depends on their geometry and on their material properties [KDH*13]. The most diffused bending-active structures are the post-formed grid shells, namely flat grids of crossing elements that are pushed from the boundaries to form a shell shape [DKZ*15; PKI*19].

Incredibly-lightweight and geometrically-complex objects can be realized by exploring various materials [GHB13; Lie14; Cro18]. By combining this forming technique with novel digital fabrication means, the spectrum of possible shapes and components can be further increased [NT13; LK15; TBM*20], so that nowadays the bending-active technique is increasingly used in challenging projects [La 17; MPI*18; LMP*20].

2.2. Dynamic Relaxation Method (DRM) as a form-finding and analysis tool

Form-finding describes the process of finding a shape of a structure for which forces are in equilibrium with respect to some applied stress [VB12]. Form-finding is primarily used in both passive and form-active structures such as grid shells, cable-nets, or tensegrities. Different numerical methods have been proposed for finding such an equilibrium state.

The DRM is an explicit integration method for pseudo-dynamically solving static problems of structures. The main idea behind it is to use the displacements of the nodes for tracking how the structure deforms. The motion of the structure is artificially damped in order to reach a static equilibrium state.

The method was initially proposed and used on membrane structures [Day65; Bar99] and later on elastic grid shells. Then, different frameworks were proposed which use a varying number of degrees of freedom (dof) for the nodes of the structure. Nodes of 3 translational dof were proposed in [AB01] for modelling tensegrity structures. The authors in [DZK16] use a 3-dof element model which is able to handle anisotropic cross section in bending active structures. The authors in [LTDC17] use a 4-dof node for descretizing the Kirchoff element theory with which they describe the centerline position of the elements and the rotation of the cross section around the centerline. A line of work uses the co-rotational element formulation which models the element elements using its two endpoints [Kre09; LK11; SP15]. The endpoints have 6 dof which are used to describe the deformed shape of a element.

Another recent work that uses the co-rotational approach is [SOA20]. Building on [Kre09] the authors use 3 dof for describing the displacements of the nodes and additionally 3 dof for describing rotations at elements' endpoints using the normal vector of the underlying surface. The computational tool presented in this paper is based on [SOA20].

3. Overview

The proposed method is based on explicitly integrating the equations of motion after an actuation load is applied to an initially flat-laying structure for reaching the static-equilibrium state. Since intermediate states are not of interest, the goal is to maximize convergence speed, and thus the equations are solved in a pseudo-dynamic manner. This means that quantities such as the mass of the structure are not realistic and are solely set for aiding convergence. Another essential fictitious component of this method is the so called *kinetic damping* which removes kinetic energy from the structure whenever a local maxima is detected. This approach damps oscillations of the structure around the equilibrium point enabling that way convergence. A high-level overview of the DRM-based algorithm implemented in this work is presented in Algorithm 11.

We begin by initializing the state of the structure (line 1). During this step, we initialize quantities such as initial positions, node velocities, kinetic energy, etc. Then, the DRM algorithm enters a loop until a static equilibrium state is reached (lines 2-11). This loop is composed of 3 steps which also form the essence of the algorithm. In line 3, we execute a DRM step that displaces the structure based on the current residual forces after a small time-step D_t. We go into more details regarding what the DRM step involves in section 4.2.

In order to speed-up convergence, we use the so-called kinetic damping. Kinetic damping is a method that removes energy from the structure every time the maximum in kinetic energy has been reached. The core idea behind kinetic damping is that since the structure oscillates around the static equilibrium, it will have its maximum kinetic energy at the equilibrium state. Since our goal is uniquely to reach the static equilibrium, without caring about the intermediate states, whenever the structure reaches a maximum in kinetic energy, the velocities of all its nodes are set to zero. We discuss more details regarding our kinetic damping approach in section 4.3. Before the end of each iteration (line 10) we decrease the current time-step using a constant $\xi < 1.0$. The purpose is to avoid overshooting the static equilibrium after the next iteration.

Algorithm 1: The implemented DRM algorithm.

```
   Input: Structure, Simulation scenario
   Output: Static equilibrium of the structure under the
           defined simulation scenario
1  Initialization;
2  while Static equilibrium of structure not reached do
3  |  Compute a DRM step;
4  |  if Structure's kinetic energy is at a local maxima then
5  |  |  Apply kinetic damping;
6  |  |  if Residual forces or kinetic energy norm of the
   |  |    structure is smaller than threshold then
7  |  |  |  terminate ;      /* Static equilibrium
   |  |  |    was reached */
8  |  |  end
9  |  end
10 |  Dt = ξ × Dt ;        /* Decreases timestep */
11 end
```

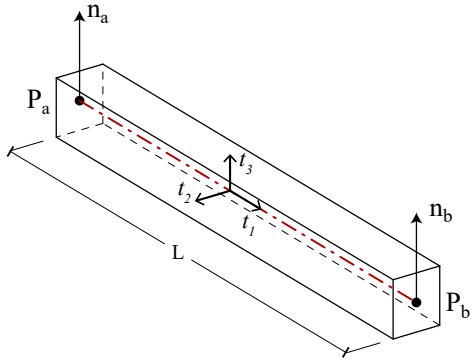

Figure 1: *Basic beam element and nomenclature.*

4. Theoretical background of the method

4.1. Structure representation

Our tool requires as input the structure and a simulation scenario to be applied. A simulation scenario is intended as a set of boundary conditions, namely restraints to nodes and a set of loads. We represent the structure as a set of vertices or nodes v_j, which are connected to each other using beam elements.

Each node j is identified by its coordinates P_j in the 3D space and a normal vector n_j. These quantities define the node variables in the global reference system during the simulation, i.e. 3 translational and 3 rotational dof. We symbolize the displacements for node j as $X_{j,x}$, $X_{j,y}$, $X_{j,z}$. The change in rotation is quantified with respect to the normal vector n_j as done in [SOA20]. We use the symbols $n_{j,x}$, $n_{j,y}$ to refer to the global x and y components of n_j and X_{j,n_x}, X_{j,n_y} to denote the displacement of these components. Lastly, we use $n_{j,r}$ to refer to the rotation around n_j. During the dynamic relaxation algorithm, we keep track of each node's 6D displacement, velocity, acceleration, forces, and its mass.

Each element has length L^k and connects two vertices (a,b). The edge (a,b) represents the axis of the beam, which is equipped with mechanical properties derived from its material and cross-section. In our examples, we assume that the whole structure has a uniform cross-section and material properties; however, diverse properties may be input per edge. We provide material characteristics through the Young's and shear modulus, E and G, respectively. The cross-section type and size is defined with reference to the local frame $< t_1, t_2, t_3 >$ in fig. 1. Accordingly, we determine the area A, the torsion constant J (i.e. for rotations around t_1), and the bending moments of inertia, I_2 and I_3 (for t_2 and t_3, respectively).

We initialize the above quantities for the given input shape and material at step 1 of algorithm 11. Conventionally, t_3 is assumed to be close to the normal of the surface.

4.2. DRM step

Each DRM step can be conceptually sketched as in algorithm 4. Starting from the current state of the structure (input in algorithm 2), each DRM step computes its displacement after a time-step D_t (output in algorithm 2). As long as the structure is not in static-equilibrium, the applied external loads are not balanced out by the internal forces, thus defining a surplus of forces within the structure, namely the residual forces (line 1). These residual forces, in turn, are able to displace the nodes (line 3), altering the kinetic energy of the structure (line 4).

Algorithm 2: A DRM step.

Input: A structure state
Output: The advanced structure state after a time step
1 Compute the residual forces at each node;
2 Update the nodal masses;
3 Compute the acceleration, velocity and displacement for each node;
4 Compute the total kinetic energy of the system;

With reference to a generic beam (fig. 1), we adopt subscript a and b to refer to the start and end node of the element. Additionally, we use upperscript and lowerscript for element and node quantities respectively. For referring to quantities in the undeformed state e.g. initial length we use an overline $\overline{L^k}$ For each DRM step the residual force $F_{R,j}$ of the j^{th} node of the structure is defined as the difference of external $F_{E,j}$ and internal forces $F_{I,j}$:

$$F_{R,j} = F_{E,j} - F_{I,j} \tag{1}$$

Since the external loads are defined for a simulation scenario, the main task is to compute the internal forces. We derive the internal forces which arise at its endpoints by differentiating the potential energy with respect to the endpoints' dof. The potential energy of an element k is formulated with respect to the normal vectors n_a and n_b at its endpoints a and b and to the local frame. As can be seen in Fig. 1, the element's local frame $< t_1^k, t_2^k, t_3^k >$ is orthonormal and consists of t_1^k the unit vector connecting the two endpoints, t_2^k the normalized cross product of t_1 and the normal vector of the element n^k, and the normalized cross product of $t_3^k = t_1^k \times t_2^k$.

$$n^k = \frac{n_a + n_b}{\|n_a + n_b\|}; t_1 = \frac{P_b - P_a}{\|P_b - P_a\|};$$
$$t_2 = \frac{n^k \times t_1^k}{\|n^k \times t_1^k\|}; t_3 = \frac{t_1^k \times t_2^k}{\|t_1^k \times t_2^k\|}$$

Next, we formulate the expressions for computing the internal potential energy of an element of the structure.

Whenever an element k is deformed, its internal potential energy U_I^k increases to resist the applied deformation. The total internal potential energy of the structure is simply the sum of the internal energies of the structure's elements:

$$U_I = \sum_k U_I^k \tag{2}$$

Element k in our framework can resist an applied deformation in 4 main ways, which generates 4 potential internal energies, namely elongation or shrinking along the beam axis U_A^k, torsion U_T^k and

bending on two planes $U^k_{B_2}$ and $U^k_{B_3}$. The sum of these four terms yields:

$$U^k_I = U^k_A + U^k_T + U^k_{B_2} + U^k_{_3} \tag{3}$$

In order to formulate the internal energies we first need to define the element's stiffness in the 4 main ways of resisting external loads:

$$\kappa_A = \frac{EA}{\overline{L^k}},\ \kappa_T = \frac{GJ}{\overline{L^k}},\ \kappa_{B_2} = \frac{EI_2}{\overline{L^k}},\ \kappa_{B_3} = \frac{EI_3}{\overline{L^k}} \tag{4}$$

We formulate the internal potential energies component-wise. The axial potential energy increases whenever the element's length L^k is changed with respect to its initial length $\overline{L^k}$:

$$U^k_A = \frac{\kappa_A}{2}(L^k - \overline{L^k})^2, \tag{5}$$

For formulating the torsion component we use the rotation angles around the frame vector t^k_1 at the two endpoints of the element (fig. 2). Under the assumption of small angles, $t^k_3 \times n_j$ can be considered as the local rotation angle at the endpoint node j. From [Kre09] the rotation angles around t^k_1 with respect to the frame of element k are:

$$\theta^k_{1,j} = t^k_1 \cdot (t^k_3 \times n_j) = n_j \cdot (t^k_1 \times t^k_3) = -t^k_2 \cdot n_j \tag{6}$$

So that the torsion energy can be formulated as:

$$U^k_T = \frac{\kappa_T}{2}(\theta^k_{1,b} - \theta^k_{1,a})^2 \tag{7}$$

Similarly the rotation angles around the frame axis t^k_2 (fig. 3) are:

$$\theta^k_{2,j} = t^k_2 \cdot (t^k_3 \times n_j) = n_j \cdot (t^k_2 \times t^k_3) = t^k_1 \cdot n_j \tag{8}$$

and thus the first bending energy is defined as:

$$U_{B_2} = \frac{\kappa_{B_2}}{2}(4(\theta^k_{2,a})^2 + 4\theta^k_{2,a}\theta^k_{2,b} + 4(\theta^k_{2,b})^2) \tag{9}$$

For computing the second bending energy we use the rotation angles $\theta^k_{3,a}$ and $\theta^k_{3,b}$ around the normal vectors n_a and n_b at the endpoints. As such, similar to the first bending energy the second bending energy is defined as:

$$U^k_{B_3} = \frac{\kappa_{B_3}}{2}(4(\theta^k_{3,a})^2 + 4\theta^k_{3,a}\theta^k_{3,b} + 4(\theta^k_{3,b})^2) \tag{10}$$

We model the intersections of multiple elements as rigid joints. Accordingly, the third rotation angle at node a with respect to the k^{th} element $\theta^k_{3,a} = X_{j,n_r}$ needs to be enforced to all incident elements (fig. 4). For doing that we follow the approach of [SOA20].

In the initial configuration, for each node we chose a reference element k among all n elements incident to a joint, and compute the angles $\phi^{k,n}_j$. Then, the third rotation angle $\theta^n_{3,j}$ at node j for element k is obtained as follows. First, we project the frame vector t^k_1 onto the plane perpendicular to the normal vector n_j in order to compute the vector f_1 (eq. 11). We then compute the deformed state of element n on the projected plane by rotating it around the normal vector with an angle $\phi^{k,n}_j + \theta^k_{3,j}$ (eq. 13). Lastly, for computing the angle $\theta^n_{3,j}$ we compute the cross product of t^n_1 and f_2 and project it onto the normal vector (eq. 14).

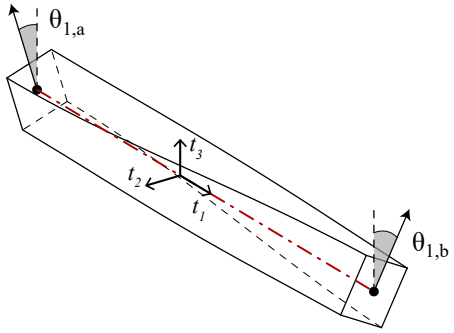

Figure 2: *Beam torsion defined from the angles* θ_1 *(about* t_1*) at the endpoints.*

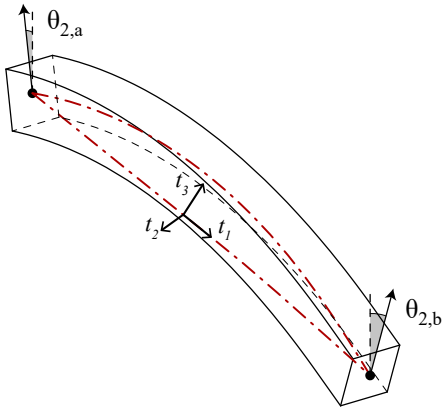

Figure 3: *Beam bending 2 defined from the angles* θ_2 *(about* t_2*) at the endpoints.*

$$f_1 = t^k_1 - (t^k_1 \cdot n_j)n_j \tag{11}$$

$$\hat{f}_1 = \frac{f_1}{\|f_1\|} \tag{12}$$

$$f_2 = \hat{f}_1 cos(\phi^{k,n}_j + \theta^k_{3,j}) + (n_j \times \hat{f}_1) sin(\phi^{k,n}_j + \theta^k_{3,j}) \tag{13}$$

$$\theta^n_{3,j} = (t^n_1 \times f_2) \cdot n_j \tag{14}$$

For nodes which have all their dof constrained we compute the third rotation angle $\theta^k_{3,j}$ as:

$$\theta^k_{3,j} = n_j \cdot (t^k_1 \times \overline{t^k_1}) \tag{15}$$

At this point, the total internal potential energy of the structure is entirely defined from the frames of its elements (eq. 2). In order to compute the internal force $F^k_{I,j,i}$ on a node j with respect to its i_{th} dof imposed from the deformation of an element k we differentiate

eq. 3 with respect to the node's i_{th} dof:

$$F^k_{I,j,i} = \frac{\partial U^k_{I,j}}{\partial u_{j,i}} \tag{16}$$

where $u_{j,i}$ represents the i_{th} dof of node j. To speed up the computation of the internal forces we provide the analytical derivatives with respect to each dof in the supplementary material of this work.

Lastly, the nodes' mass (line 2) is required in order to compute the per-node 6D-vector of residual forces. Since we are only interested in the static-equilibrium state and not any intermediate state, the node masses do not need to make physical sense but are solely used to aid convergence. For this purpose, we use the artificial mass formulation of [LTDC17]. The translational and rotational masses m^k_t and m^k_r of each element are initialized as:

$$m^k_t = 2\gamma\Delta t^2_{ini}\frac{EA}{L^k} \qquad m^k_r = 8\gamma\Delta t^2_{ini}\frac{E(I_2+I_3)}{(L^k)^3} \tag{17}$$

where $\gamma = 0.8$. Note here that the masses do not remain constant but change as the length of the element changes. The corresponding node mass M_j of node j is simply obtained from adding the masses of the n incident elements:

$$M_j = \sum_n m^k \tag{18}$$

where m^k is the mass of element $k \in n$. Having computed the residual force $F_{R,j}$ we compute the acceleration α_j, the velocity υ_j and finally the displacement vector X_j of node j (step 3 in algorithm 4) as

$$\alpha_j = \frac{F_{res,j,i}}{M_j} \tag{19}$$

$$\upsilon_j = \upsilon_j + \alpha_j\Delta t \tag{20}$$

$$X_j = X_j + \upsilon_j\Delta t \tag{21}$$

At this point we also update the system's kinetic energy (step 4 in algorithm 4) which is used to detect local maxima in the kinetic energy of the structure (see sec.4.3). Finally, we update for each node the coordinates P_j, its normal vector n_j and its rotation angle around its normal $n_{r,j}$ using the displacement vector $X_j =$ as:

$$P_j = \overline{P_j} + X_{j,xyz} \tag{22}$$

$$n_{j,x} = \overline{n_{j,x}} + X_{j,n_x} \tag{23}$$

$$n_{j,y} = \overline{n_{j,y}} + X_{j,n_y} \tag{24}$$

$$n_{j,r} = X_{j,n_r} \tag{25}$$

where $\overline{P_j}$ and $\overline{n_j}$ denote the initial position and normal vector of node j.

4.3. Kinetic damping

An essential component of the DRM is the removal of energy in order to reach converge. We use the so-called kinetic damping as a way to artificially remove energy from the structure and reach convergence. We bring the structure to its static equilibrium by halting the movement of the nodes whenever a peak in the kinetic energy of the structure is detected. A local maximum in the kinetic energy of the structure corresponds to local minima in the system's potential energy and, thus, a potential static equilibrium state for the structure.

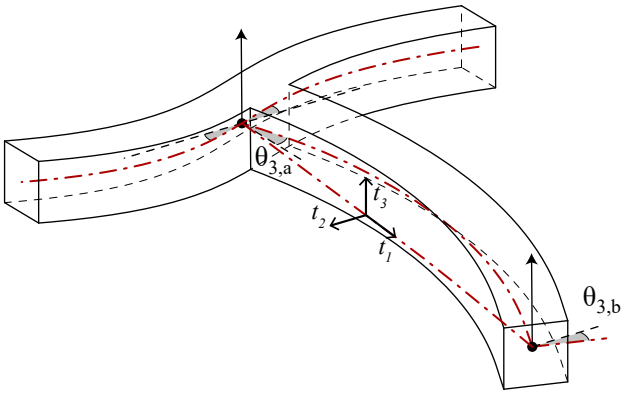

Figure 4: *Beam bending 3 defined from the angles θ_3 (about t_3) at the endpoints. We applied rigid nodes formulation, so multiple beams that are incident in a node preserve locally the same mutual angle.*

In fig.10, we plot the kinetic energy with respect to the executed iterations of the tripod test case presented in sec.5. The orange points in the plot correspond to iterations in which kinetic damping occurs for removing energy from the system. The colors on the structure demonstrate how the kinetic energy norm is distributed over the nodes of the structure. Initially the kinetic energy rises as a result of applying the defined boundary conditions and a temporal external force. The structure then gradually converges to its equilibrium state and the kinetic energy minimums consistently decrease.

4.4. Implementation details

We have implemented the algorithm presented in this paper in C++ and provided it as an open-source library available on GitHub. In this section, we give details on the code structure and on the strategies we adopted to accelerate our implementation.

Along with the simulation library core, we provide a demo application with which the user can replicate the results presented in sec.5. We incorporate [Sha*19] for visualizing the structures, the applied simulation scenarios, and results.

Minimizing computation times is essential to any simulation tool, and as such, our implementation was done with that goal in mind. Since the computation of the energy of each element is independent of the others in each DRM step, we compute the residual forces of each element in parallel allowing significant speedup, especially on more complex structures. In order to minimize processing times and memory consumption, we cache and precompute the derivatives required for computing eq.1, which is the most computation-intensive task in each DRM step. Since the internal force expressions vary depending on the dof to be computed, we have reformulated the required equations for minimizing the computational cost for each dof. Additionally, we observed that eq.16 is non-zero only for the dof of the endpoints of element k and the dof of the non-common endpoint of the reference element of node j. Therefore, to increase the performance restrict the partial derivative

calculations the non-zero dof. The reformulated expressions can be found in the additional material document provided with this paper.

5. Experimental results

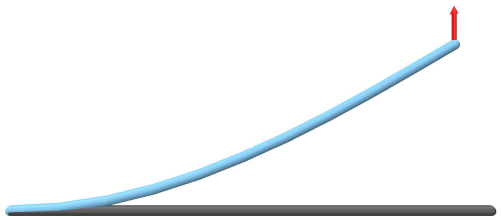

Figure 5: *Cantilever test case in which we apply a vertical force in the right end of the beam which is composed of 10 elements leading to a maximum displacement of 0.35m.*

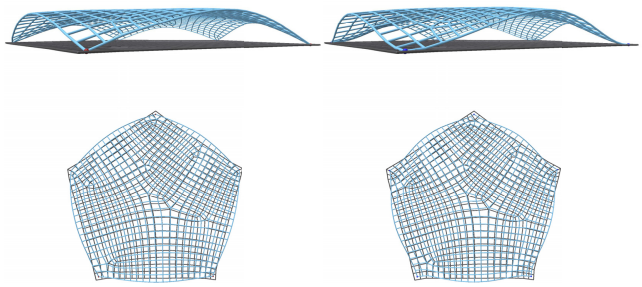

Figure 6: *PentaShell side view(bottom row) and top view(bottom row). We displace the corners of the pentagon towards its center leaving the rotational dof free (upper left and bottom left) and constraining them (upper and bottom right).*

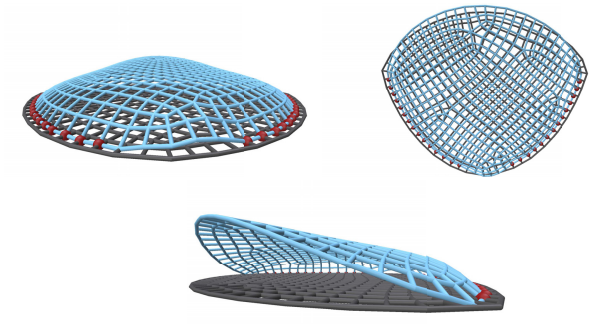

Figure 7: *Roof test case in which some border nodes are displaced towards the interior of the structure. The deformed shape reaches a maximum displacement of 5.7m.*

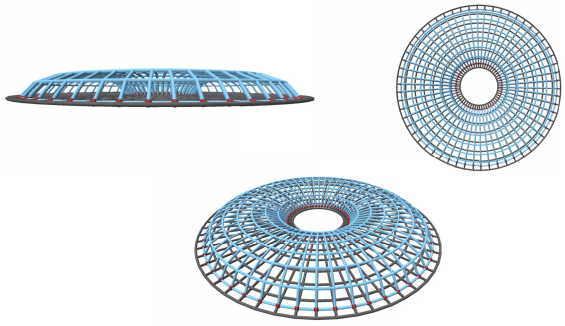

Figure 8: *Torus test case in which the outer border vertices are squeezed towards the interior resulting in a maximum displacement of 2.4m.*

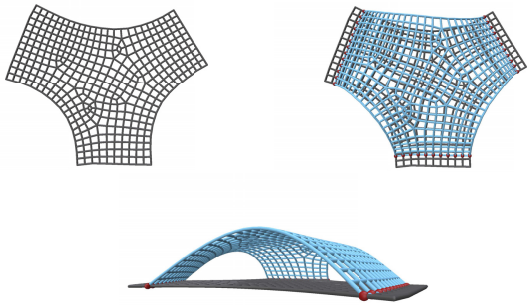

Figure 9: *Tripod test case in which we displace towards the interior some vertices of the structure's border reaching a maximum displacement of 7m.*

We test our tool against a classic benchmark problem of a cantilever beam (fig. 5) and several bending-active structures (figs. 6-9). For all scenarios, we used a Poisson's ratio of 0.3 (that is required to compute the G) and a rectangular cross-section of uniform dimensions over the whole structure. As a convergence criteria, we used a threshold of 1e-3 Newtons or a Kinetic energy smaller than 1e-15. Note that for all scenarios besides the cantilever for coping with bifurcation buckling, we apply an initial vertical force of 1 kN for the first 1000 DRM steps in order to move the structure out-of-plane. After this initial distortion, we apply the displacement fields of those scenarios. The characteristics of all the examples, along with the required DRM iterations, execution time and error with respect to commercial FEM software [GD 05] results are included in tab. 1.

In fig.5-9 we use dark gray and cyan in order to depict the initial and deformed configurations, respectively. We denote nodes on which we apply a specific displacement with red spheres.

In fig. 5 we use a unit length cantilever beam composed of 10 elements with a cross-section of 0.015x0.015m^2, 1 GPa Young's modulus and apply a vertical force on the right end of 5N.

In fig.6 we use a pentagonal surface of size 32x36m^2, a Young's modulus of 10 GPa and beam cross sections of 0.1x0.1m^2. We dis-

Table 1: *Statistics and results on different examples: number of elements |k|, number of nodes |j|, cross section size (m^2), Young's modulus E, number of iterations to convergence, computation time, maximum displacement X_{max}(m), max and mean error(m) computed by comparing the results to an FEM simulation.*

Name	|k|	|j|	Cross section(m^2)	BBox (m^2)	E (GPa)	Iterations	Time(sec)	X_{max}(m)	Max Error(m)	Mean Error(m)
Cantilever	10	11	0.015x0.015	1	1	6171	0.23	0.35	0.00036	0.00019
PentaShell pinned	1146	597	0.1x0.1	32x36	10	39620	35	3.7	0.019	0.0039
PentaShell fixed	1146	597	0.1x0.1	32x36	10	23682	23	4	0.024	0.016
Roof	848	408	0.2x0.2	37x37	200	43380	29	5.7	0.17	0.056
Torus	870	464	0.1x0.1	37x37	10	8952	5	2.4	0.14	0.059
Tripod	828	439	0.25x0.2	37x35	50	21874	14	7	0.021	0.0068

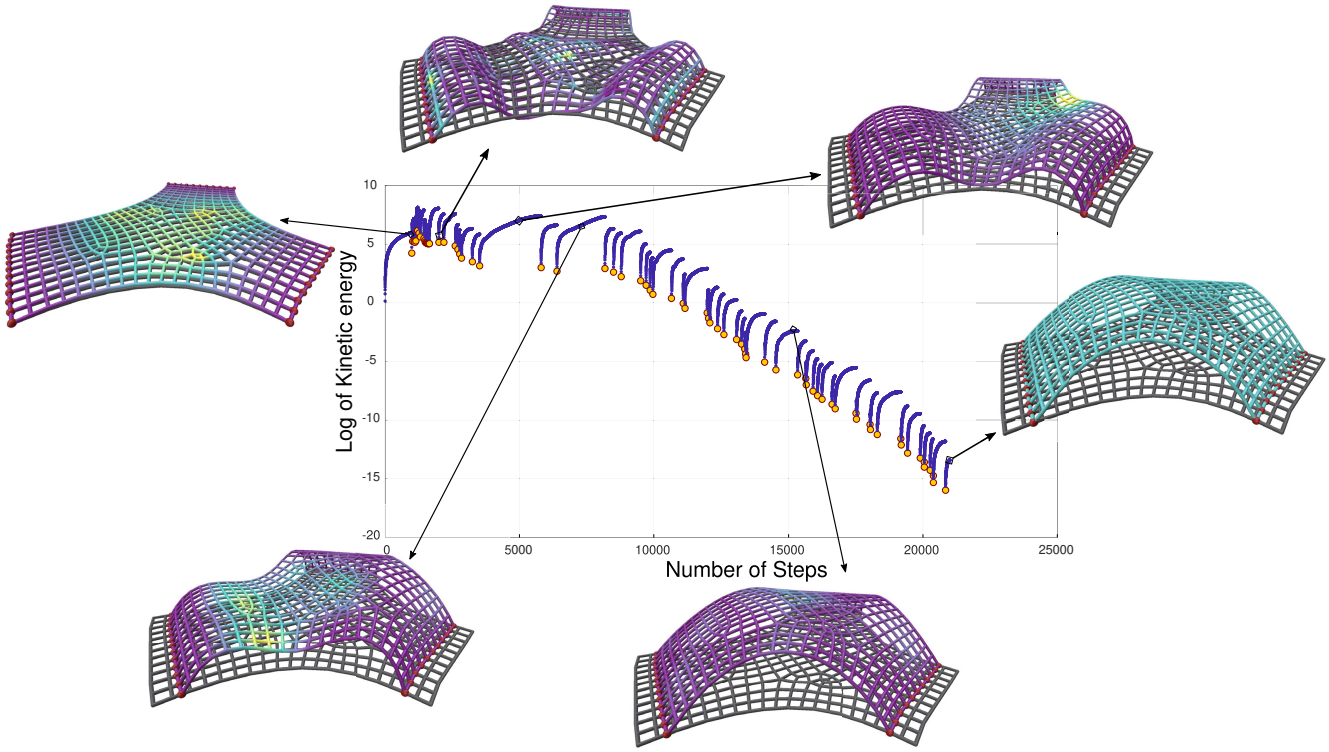

Figure 10: *A plot of the logarithm of the kinetic energy of the scenario in fig.9 on the Tripod structure. The orange points depict the iterations on which kinetic damping occurs. The color-map of the structure depicts the kinetic energy norm throughout the structure. From step 0 until 1000 a vertical out-of-plane force is applied for coping with bifurcation buckling effects. From step 1000 until 2000 we gradually displace the boundary nodes towards the interior of the structure. After step 7000 we see a steady decrease of the kinetic energy while the structure is moving towards its static equilibrium shape.*

place the vertices of the pentagon towards its center with a displacement of 0.7 m resulting in a maximum displacement in the deformed state of 3.7 m. On the scenario depicted on the top and bottom right images of fig.6, we additionally constrain the rotational dof of the displaced vertices resulting in a maximum displacement of 4m. In fig.7 we present a simulation scenario applied to a roof-like structure of bounding box size 37x37m^2, a Young's modulus of 200 GPa, a cross-section of 0.2x0.2m^2. We apply a displace of some border nodes towards the interior with a displacement of 3% of their distance from the center of the structure. The maximum displacement we get as a result is 5.7m.

In fig.8 we deform a circular surface with a hole in its center of size 32x36m^2, with elements of cross-section 0.1x0.1m^2 and a Young's modulus of 10 GPa by squeezing the outer border vertices towards the interior with a displacement of close to 3% of the surface's diameter in the initial state, reaching a maximum displacement in the deformed state of 2.4m. In fig.9 we deform a tripod surface of size 37x35m^2, elements of cross-section 0.25x0.2m^2,

where 0.25 corresponds to the in-plane dimension and a Young's modulus of 50GPa. We displace the border vertices laying on the "non-curved sides" towards the interior, reaching a maximum displacement in the deformed state of 7m.

The timings for the computation of the FEM results are 3 (cantilever), 27 (PentaShell pinned), 66 (PentaShell fixed), 39 (Roof), 18 (Torus) and 18 (Tripod) seconds as computed on a Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz machine. The DRM results presented in this section were computed using an Intel(R) Core(TM) i7-9750H @ 2.6GHz CPU. Even though we did not use a high-end processor for performing the tests in this work, the performance is significant, with an average processing time for the performed cases, excluding the cantilever, of less than 1.5 microseconds per iteration and per node allowing the use of our tool on personal computers.

6. Conclusion

In this paper, we presented a simulation model based on the dynamic relaxation method for analyzing bending-active structures. The presented model is based on the computation of internal forces of the structure using the per-vertex global normals for defining per-element local frames. We discussed and provided a novel open-source implementation of our tool along with the theoretical background. We have tested the accuracy and performance of our computational tool on several real-world architectural structures composed of beam elements with various material and cross-section properties targeting a wide range of displacements. We assessed the accuracy of our tool, comparing the results to an accurate ground truth FEM model. Our tool demonstrates both accuracy and high performance with low computation times, even on low-end personal computers.

Although we demonstrated that our tool can be used on bending-active structures for solving static-equilibrium problems, the DRM has several parameters which determine both its stability and convergence time making its use less intuitive to the non-expert user. Additionally, since the discussed simulation model does not consider shear forces in the elements' cross-section of the elements, its accuracy diminishes when these forces become relevant.

Acknowledgments

The authors would like to thank Yusuke Sakai for readily explaining parts of his work. This research was partially funded by the EU H2020 Programme EVOCATION: Advanced Visual and Geometric Computing for 3D Capture, Display, and Fabrication (grant no. 813170). Gianmarco Cherchi gratefully acknowledges the support to his research by PON R&I 2014-2020 AIM1895943-1.

References

[AB01] ADRIAENSSENS, S.M.L and BARNES, M.R. "Tensegrity spline beam and grid shell structures". *Engineering Structures* 23.1 (2001), 29–36. ISSN: 0141-0296. DOI: https://doi.org/10.1016/S0141-0296(00)00019-5. URL: https://www.sciencedirect.com/science/article/pii/S0141029600000195 2.

[Bar99] BARNES, MICHAEL R. "Form finding and analysis of tension structures by dynamic relaxation". *International journal of space structures* 14.2 (1999), 89–104 2.

[Cro18] CROLLA, KRISTOF. "Bending Bamboo Rules: Beyond Century-Old Typologies". *Journal of Architectural Education* 72.1 (2018), 135–145. DOI: 10.1080/10464883.2018.1410669 2.

[Day65] DAY, AS. "An introduction to dynamic relaxation(Dynamic relaxation method for structural analysis, using computer to calculate internal forces following development from initially unloaded state)". *The engineer* 219 (1965), 218–221 2.

[DKZ*15] D'AMICO, B, KERMANI, A, ZHANG, H, et al. "Timber gridshells: Numerical simulation, design and construction of a full scale structure". *Structures* 3 (2015), 227–235 2.

[DZK16] D'AMICO, B., ZHANG, H., and KERMANI, A. "A finite-difference formulation of elastic rod for the design of actively bent structures". en. *Engineering Structures* 117 (June 2016), 518–527. ISSN: 0141-0296. DOI: 10.1016/j.engstruct.2016.03.034. URL: https://www.sciencedirect.com/science/article/pii/S0141029616300694 (visited on 07/19/2022) 2.

[GD 05] G+D COMPUTING. *Straus7*. 2005 6.

[GHB13] GENGNAGEL, CHRISTOPH, HERNÁNDEZ, ELISA LAFUENTE, and BÄUMER, RALF. "Natural-fibre-reinforced plastics in actively bent structures". *Proceedings of the Institution of Civil Engineers-Construction Materials* 166.6 (2013), 365–377. DOI: 10.1680/coma.12.00026 2.

[KDH*13] KOTELNIKOVA-WEILER, NATALIA, DOUTHE, CYRIL, HERNANDEZ, E LAFUENTE, et al. "Materials for actively-bent structures". *International Journal of Space Structures* 28.3-4 (2013), 229–240. DOI: 10.1260/0266-3511.28.3-4.229 2.

[Kre09] KRENK, STEEN. *Non-linear modeling and analysis of solids and structures*. Cambridge University Press, 2009 2, 4.

[La 17] LA MAGNA, RICCARDO. "Bending-active plates: strategies for the induction of curvature through the means of elastic bending of plate-based structures". 43. Institut für Tragkonstruktionen und Konstruktives Entwerfen (ITKE), Universität Stuttgart, 2017 2.

[LAGK13] LIENHARD, JULIAN, ALPERMANN, HOLGER, GENGNAGEL, CHRISTOPH, and KNIPPERS, JAN. "Active bending, a review on structures where bending is used as a self-formation process". *International Journal of Space Structures* 28.3-4 (2013), 187–196. DOI: 10.1260/0266-3511.28.3-4.187 2.

[Lie14] LIENHARD, JULIAN. "Bending-active structures: form-finding strategies using elastic deformation in static and kinetic systems and the structural potentials therein". 36. Institut für Tragkonstruktionen und Konstruktives Entwerfen (ITKE), Universität Stuttgart, 2014 2.

[LK11] LI, JIAN-MIN and KNIPPERS, JAN. "Form-finding of grid shells with continuous elastic rods". (2011) 2.

[LK15] LIENHARD, JULIAN and KNIPPERS, JAN. "Bending-active textile hybrids". *Journal of the International Association for Shell and Spatial Structures* 56.1 (2015), 37–48 2.

[LMP*20] LACCONE, FRANCESCO, MALOMO, LUIGI, PÉREZ, JESÚS, et al. "A bending-active twisted-arch plywood structure: computational design and fabrication of the FlexMaps Pavilion". *SN Applied Sciences* 2.9 (2020), 1–9. DOI: 10.1007/s42452-020-03305-w 2.

[LTDC17] LEFEVRE, BAPTISTE, TAYEB, FRÉDÉRIC, DU PELOUX, LIONEL, and CARON, JEAN-FRANÇOIS. "A 4-degree-of-freedom Kirchhoff beam model for the modeling of bending–torsion couplings in active-bending structures". *International Journal of Space Structures* 32.2 (2017), 69–83 2, 5.

[MPI*18] MALOMO, LUIGI, PÉREZ, JESÚS, IARUSSI, EMMANUEL, et al. "FlexMaps: computational design of flat flexible shells for shaping 3D objects". *ACM Trans. Graph.* (2018), 241. DOI: 10.1145/3272127.3275076 2.

[NT13] NICHOLAS, PAUL and TAMKE, MARTIN. "Computational strategies for the architectural design of bending active structures". *International Journal of Space Structures* 28.3-4 (2013), 215–228. DOI: 10.1260/0266-3511.28.3-4.215 2.

[PKI*19] PANETTA, J., KONAKOVIĆ-LUKOVIĆ, M., ISVORANU, F., et al. "X-Shells: A New Class of Deployable Beam Structures". *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3323040. URL: https://doi.org/10.1145/3306346.3323040 2.

[Sha*19] SHARP, NICHOLAS et al. *Polyscope*. www.polyscope.run. 2019 5.

[SOA20] SAKAI, YUSUKE, OHSAKI, MAKOTO, and ADRIAENSSENS, SIGRID. "A 3-dimensional elastic beam model for form-finding of bending-active gridshells". *International Journal of Solids and Structures* 193 (2020), 328–337. DOI: 10.1016/j.ijsolstr.2020.02.034 1–4.

[SP15] SENATORE, GENNARO and PIKER, DANIEL. "Interactive real-time physics: an intuitive approach to form-finding and structural analysis for design and education". *Computer-Aided Design* 61 (2015), 32–41 2.

[TBM*20] TAMKE, MARTIN, BARANOVSKAYA, YULIYA SINKE, MONTEIRO, FILIPA, et al. "Computational knit – design and fabrication systems for textile structures with customised and graded CNC knitted fabrics". *Architectural Engineering and Design Management* 0.0 (2020), 1–21. DOI: 10.1080/17452007.2020.1747386. eprint: https://doi.org/10.1080/17452007.2020.1747386. URL: https://doi.org/10.1080/17452007.2020.1747386 2.

[VB12] VEENENDAAL, DIEDERIK and BLOCK, PHILIPPE. "An overview and comparison of structural form finding methods for general networks". *International Journal of Solids and Structures* 49.26 (2012), 3741–3753 2.