



Universiteit  
Leiden  
The Netherlands

## **Flexel ecosystem: simulating mechanical systems from entities with arbitrarily complex mechanical responses**

Ducarme, P.; Weber, B.; Hecke, M.L. van; Overvelde, J.T.B.

### **Citation**

Ducarme, P., Weber, B., Hecke, M. L. van, & Overvelde, J. T. B. (2025). Flexel ecosystem: simulating mechanical systems from entities with arbitrarily complex mechanical responses. *Arxiv*. doi:10.48550/arXiv.2510.19741

Version: Publisher's Version  
License: [Creative Commons CC BY-SA 4.0 license](https://creativecommons.org/licenses/by-sa/4.0/)  
Downloaded from: <https://hdl.handle.net/1887/4296512>

**Note:** To cite this publication please use the final published version (if applicable).

# Flexel ecosystem: simulating mechanical systems from entities with arbitrarily complex mechanical responses

Paul Ducarme<sup>a,b</sup>, Bart Weber<sup>b,c</sup>, Martin van Hecke<sup>a,d</sup>, Johannes T.B. Overvelde<sup>a,e</sup>

<sup>a</sup>AMOLF, Amsterdam, The Netherlands

<sup>b</sup>Advanced Research Center for Nanolithography, Amsterdam, The Netherlands

<sup>c</sup>Van der Waals–Zeeman Institute, Institute of Physics, Universiteit van Amsterdam, Amsterdam, The Netherlands

<sup>d</sup>Huygens-Kamerlingh Onnes Lab, Leiden Institute of Physics, Universiteit Leiden, The Netherlands

<sup>e</sup>Institute for Complex Molecular Systems and Department of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands

October 23, 2025

## Abstract

Nonlinearities and instabilities in mechanical structures have shown great promise for embedding advanced functionalities. However, simulating structures subject to nonlinearities can be challenging due to the complexity of their behavior, such as large shape changes, effect of pre-tension, negative stiffness and instabilities. While traditional finite element analysis is capable of simulating a specific nonlinear structure quantitatively, it can be costly and cumbersome to use due to the high number of degrees of freedom involved. We propose a framework to facilitate the exploration of highly nonlinear structures under quasistatic conditions. In our framework, models are simplified by introducing ‘flexels’, elements capable of intrinsically representing the complex mechanical responses of compound structures. By extending the concept of nonlinear springs, flexels can be characterized by multi-valued response curves, and model various mechanical deformations, interactions and stimuli, e.g., stretching, bending, contact, pneumatic actuation, and cable-driven actuation. We demonstrate that the versatility of the formulation allows to model and simulate, with just a few elements, complex mechanical systems such as pre-stressed tensegrities, tape spring mechanisms, interaction of buckled beams and pneumatic soft gripper actuated using a metafluid. With the implementation of the framework in an easy-to-use Python library, we believe that the flexel formulation will provide a useful modeling approach for understanding and designing nonlinear mechanical structures.

## 1 Introduction

These last years have garnered a significant interest in understanding how to leverage large deformation, nonlinearities and instabilities to design structures capable of complex yet functional mechanical responses [1, 2]. For example, compliant mechanisms leverage flexibility

to mimic conventional mechanism usually composed of many stiff components connected by joints [3]. Soft robots utilize shape changes to be intrinsically more robust, adaptable and autonomous, often by harnessing mechanical instabilities to embed sequenced [4], asymmetric [5], fast [6] or amplified [7] actuation. Recent progress in mechanical metamaterials have exploited large geometric changes and nonlinearities to achieve increasingly complex deformation pathways, enabling multistability [8] or high energy dissipation [9]. While the advancements driven by the exploitation of nonlinearities increase the capabilities of these fields, they also introduce new challenges for simulation and design.

Whereas numerically studying structures in their small-deformation regime follows established approaches given their linear responses, studying a structure subject to large geometric changes is more challenging due to the nonlinearities involved. Nonlinear finite element analyses suffer from limitations that make them less applicable for conceptual design, and do not always contribute to the understanding of systems that undergo large deformation and instabilities. For example, due to the high number of degrees of freedom involved, they can have prohibitive computational costs, fail upon facing mechanical instabilities or distorted meshes, and act as black boxes that are challenging to gain insight from.

To cope with these difficulties, reduced-order models that are characterized by fewer degrees of freedom offer a more practical alternative to finite element analyses. They are easier to define, solve, and interpret by promoting qualitative understanding over quantitative accuracy. They usually employ fewer elements which together are still able to capture the qualitative phenomenology. For example, the pseudo-rigid body model uses a combination of rigid links and torsional springs to represent flexible, slender parts [3]. Trusses of bars can be used to model and even inverse design multistable compliant structures [10]. Cosserat rods have shown great promise for modeling soft robotic [11] or soft living [12] systems. Simple spring or beam models that can be expressed analytically have been valuable in gaining insight into a wide variety of geometrically nonlinear mechanical structures [13, 14, 15, 16, 17, 18, 19].

Essentially, such existing approaches aim for a higher-level description of the mechanical system. Instead of describing the system as a mesh of many material elements, combining fewer abstract components allows to construct models that are cheaper to solve and easier to understand. However, strongly nonlinear mechanical behaviors such as nonmonotonic or multi-valued force-deformation paths are still either modeled from the bottom-up, by combining multiple more simple components, or by using overly abstract models such as ‘hysterons’, which require complex modeling to connect them to physical systems and moreover can lead to ill-defined systems [20, 21, 22, 23].

Here, we introduce an easy-to-use framework that defines components at a level of abstraction that allows them to single-handedly and intrinsically capture highly nonlinear static mechanical responses that are traditionally achieved using multiple components. Our formulation is based on energy potentials tuned to be stationary on prescribed, arbitrarily complex, possibly multi-valued generalized force-displacement curves. We show that this energy-based formulation can generate a broad ecosystem of mechanical components, with a wide range of geometries and intrinsic complexities. We also show that the framework allows for a relatively straightforward approach to build aggregate models where the various inter-

actions between simple and complex elements can be simulated and explored. We conclude this work by demonstrating that our framework allows to model, simplify, and simulate a wide variety of mechanical systems studied previously that are subject to, e.g., geometric nonlinearities, pre-stress, hysteresis, buckling, snapping, contact, cable-driven or pneumatic actuation.

## 2 Flexels

Before introducing the full ecosystem, let us stress that structures composed of linear springs can exhibit nonlinear force-displacement relations resulting from geometric nonlinearities. A classic example is the Von-Mises truss, which, despite being composed of three linear springs, produces a non-monotonic (Fig. 1a) or even a multi-valued force-displacement curve (Fig. 1b), depending on the relative stiffness of the springs. The foundation of our approach is to represent such structures by single entities, which we call ‘flexels’. As shown in Fig. 1c,d, the intrinsic nonlinear behavior of a flexel can be tuned to capture, and be equivalent to, the geometric nonlinear behavior of compound structures, such as the Von-Mises trusses shown in Fig. 1a,b (Movie S1).

More formally, a flexel is defined as a deformable element whose elastic energy directly depends on its geometric measure  $\alpha$ , a scalar quantity determined from the coordinates of the nodes composing the element, such as its length. The intrinsic nonlinear behavior of a flexel is defined by a generalized force-displacement curve, which relates the derivative of its elastic potential with respect to the geometric measure (that is, the generalized force  $f$ ) to the change in geometric measure (that is, the generalized displacement  $u = \Delta\alpha$ ). This relation encapsulates the nonlinear response and is used to construct an energy potential from which the ingredients needed for numerical simulations can be derived (SI section 1.1).

Flexels can be assembled to investigate how interactions between individual nonlinear elements give rise to more complex collective responses (SI section 1.2). For instance, coupling two flexels each characterized by a non-monotonic and multi-valued force-displacement curve in series reveals an equilibrium path with multiple turning points, indicating the presence of a snapping sequence upon loading and unloading (Fig. 1e, Movie S1). By assembling these flexels in other configurations, the interplay between intrinsic and geometric nonlinearities can be explored. For example, the same pair of flexels, assembled now at an angle and loaded from their connection point, produces a different snapping sequence, due to the additional geometric nonlinearities (Fig. 1f, Movie S1). The more complex force-displacement responses exhibited by the assemblies shown in Fig. 1e,f can in turn be replicated by a single flexel (Fig. 1g,h). Our formulation enables simulation at a higher level, eliminating the need of simulating the individual components of larger structures that flexels are intended to mimic. Still, it should be noted that while flexels provide a powerful approach to reduce the number of degrees of freedom while maintaining nonlinear behavior, the abstraction subsumes the deformation of the internal degrees of freedom into the flexel behavior, thereby hiding them from the surrounding and precluding direct coupling between internal nodes or non-actuated nodal loading direction with other flexels. For example, the deformation of the top node in Fig. 1b is absorbed within its flexel abstraction (Fig. 1d) and cannot be coupled to other flexels. Similarly, the deformation of the top node of the assembly shown in Fig. 1f along the

horizontal direction cannot be coupled to other elements when using its flexel abstraction (Fig. 1h). Information about these internal deformations is however not lost; it could be retrieved by mapping the flexel state back to the assembly that it mimics.

Simulating assemblies of flexels under quasi-static loading involves solving a system of parametrized nonlinear equations, which our toolkit achieves by implementing the arc-length method [24] (SI section 2). This numerical continuation scheme retrieves an entire succession of deformed states at equilibrium even if those form a path with turning points, allowing for simulations of structures subject to snapping instabilities at either constant force or displacement. Note that if the equilibrium path splits into multiple branches at a pitchfork bifurcation, only one branch will be continued by the arclength scheme. Moreover, the arc-length scheme is unable to retrieve equilibria disconnected from the initial path. Alternative methods could be implemented to cope with these two limitations [25].

### 3 Flexel ecosystem

The most basic flexel can be derived from the generalization of a nonlinear spring. For a nonlinear spring, the axial force is computed in two steps: the length is determined from the coordinates of its end nodes, then passed to an energy potential whose derivative yields the force. More general, a flexel extends this idea in two ways. First, the notion of length is broadened into a geometric measure (noted  $\alpha$ ), meaning any scalar quantity computed from a list of nodes' coordinates (noted  $z_i$ ), such as angle, area, total path length or the distance between a point and a line (Fig. 2a). Second, the class of energy potentials is widened by supporting tunable, potentially multi-valued generalized force-displacement curves (Fig. 2b). They can have an arbitrary number of turning points and intersections, allowing flexels to encode information about their loading history [21], or capture snapping or countersnapping phenomena [26], for example. This is achieved by defining a custom energy potential  $v(\alpha, t)$  that produces equilibria on the prescribed multi-valued path (SI section 4.2). The only restrictions are that the curve must remain continuous and must not form loops in which the tangent vector crosses the *vertical upward* direction, which prevents some structures to be abstracted into single flexels.

Thanks to the decoupling between geometry and intrinsic behavior, we can generate a whole flexel ecosystem by independently defining geometric measures on the one hand and generalized force-displacement curves on the other. By passing a geometric measure  $\alpha$  (computed from the nodes' coordinates  $z_i$ ) to an energy potential  $v$ , we pair a measure to a curve and produces a flexel that can be directly used in simulations to model complex mechanical entities (Fig. 2c, SI section 1, Movie S2). Let us give a few examples. Pairing an angle to a multi-valued curve gives a flexel that models a flexure capable of snapping at constant angular displacement (Fig. 3a). Combining an area to a softening curve yields a flexel that mimics the behavior of a compressible fluid and can be used to model pneumatic actuation (Fig. 3b). The total length of a polygonal chain coupled to a stiffening bilinear curve gives birth to a flexel that models a rope that is yet to be taut and can be used in cable-driven systems (Fig. 3c). Pairing the point-line distance to a curve yielding a nonzero repulsion force only under a certain threshold gives a flexel that models contact (Fig. 3d).

## 4 Workflow

In this section we illustrate how our toolkit can be integrated to the workflow to design a structure with complex deformation pathways (Fig. 4, Movie S3). To illustrate this, we consider two physical structures fabricated in silicone rubber (Smooth-On, Smooth-Sil 945) [26], shown in Fig. 4a. We set the goal of predicting the experimental tensile response of the system formed by coupling them in series.

First, a tensile test is carried out on each structure during which the force is measured while increasing and decreasing the extension. This reveals nonmonotonic force-displacement curves (Fig. 4b). Second, for each experimental test, a Bezier curve of degree 4 is fitted to the experimental data by adjusting the positions of control points (Fig. 4c-left). Fitting such a curve consists of finding the positions of control points, which form a control polygon that shapes the curve. Third, each Bezier curve is represented by a string of text listing the coordinates of the control points, which can then be used to define the generalized force-displacement curve of a longitudinal flexel, modeling each structure as a nonlinear spring (Fig. 4c-right). Fourth, we model the assembly of the two structures by connecting the two flexels in series (Fig. 4d-right).

In practical terms, this model is written in an input file that describes the node positions, the boundary conditions, the flexels, and the loading steps (Fig. 4d-left). Note that before loading node 2, the structure is preloaded by applying a load on node 1 to account for the weight of the assembly. More details on how to interpret or compose such input file are provided in SI section 6.

Finally, we find that the simulated force-displacement curve is complex and characterized by many turning points (Fig. 4e). We performed the physical experiment by connecting both structures in series and conducting a vertical tensile test, to validate the approach. Good agreement has been obtained both quantitatively and qualitatively (Fig. 4e). If multiple physical building blocks are available and characterized, this approach allows to build a catalog of achievable nonlinear behaviors that can be combined to quickly simulate assemblies. By scanning different nonlinear behavior combinations, the approach can help identify which blocks should be assembled within a certain network to get a specific desired response [26].

## 5 Use cases

Using this workflow, we next demonstrate the versatility of the ecosystem to model and handle complex use cases (Fig. 5, Movie S4). As a first example, we show that stress-free or prestressed tensegrity trusses [27] can be effectively modeled by connecting longitudinal flexels with compatible or incompatible rest lengths (SI section 7.4.1). We observe that the stress-free truss behaves as a mechanism in the small deformation regime, i.e., with zero initial stiffness, while the prestressed truss is stabilized by acquiring a nonzero initial stiffness (Fig. 5a). This confirms that this truss is a pure tensegrity structure [27]. Beyond small deformations, our approach is able to capture the nonlinearities originating from the rotation of the longitudinal flexels, allowing to study the behavior of such structures under larger loads.

In the second example, we show that a tape-spring system [28] can be modeled using a

multi-valued angular flexel and a stiff path flexel (SI section 7.4.2). By loading the system in two steps, buckling and kink formation followed by zero-stiffness deformation can be simulated (Fig. 5b), reproducing the soft deformation mode used for gripping [28]. The current implementation only allows to load along nodal coordinates, but the framework can naturally be extended to directly actuate the rest geometric measure of a flexel, allowing for different loading modes, e.g., angular loading, to model the various actuation modes of the mechanism [28].

Third, we show that a distance flexel can be used to model the contact between interacting buckled beams (SI section 7.4.3), reproducing the effect of the separating distance on the global buckling direction of pairs of “bumping beams” [29] (Fig. 5c). Unlike previous work studying similar systems using dynamic simulations [29, 30], our static approach allows to retrieve stable and unstable branches, giving insight into the various equilibrium configurations. We note that as the number of beams increases, our approach becomes costly as many stable and unstable configurations will have to be computed.

Fourth, we show that a pneumatic gripper actuated using a metafluid [31] (medium composed of collapsible capsules surrounded by a fluid) can be modeled and simulated using a single multi-valued area flexel for the metafluid, linear angular and longitudinal flexels for the gripper, and distance flexel for the gripping contact (SI section 7.4.4). The simulation reveals that the metafluid modulates the gripping force, allowing for delicate grasping as previously demonstrated experimentally in [31]. Instead of modeling the metafluid by adding degrees of freedom for each capsules, the curve itself is directly assigned to an area flexel, drastically reducing the number of degrees of freedom. By contrast to the modeling approach shown before [31], adding extra capsules does not cost any additional degrees of freedom. The price to pay for that improvement is that the curve should be determined in advance, via a traditional modeling approach or via an experiment [31]. But once it is known, it can be used to model systems where that fluid is used at a much cheaper cost, while still capturing the complexity.

Together, these examples show the versatility of our approach. Importantly, each example only required a few lines of code to be simulated (SI section 7).

## 6 Conclusion

We introduced the concept of flexels, entities able to single-handedly capture the strongly nonlinear responses of compound systems. This facilitates the simulation of various complex mechanical systems, with a focus on nonlinearities and instabilities. To promote accessibility and adoption, the formulation is implemented in an open-source and user-friendly Python library [33].

Key to developing this framework was the generalization of the concept of nonlinear spring along two independent axes, geometry and intrinsic mechanical behavior, which naturally generates a broad ecosystem of easy-to-combine elements. This ecosystem can be extended by defining different geometric measures or families of curves, which opens route to define custom flexels that are suitable to tackle specific mechanical problems, while still being compatible with the previously defined ones. Because our formulation is energy-based, it enables the use of well-established algorithms as well as the traditional stability analysis tools.

Furthermore, even though we focused on static simulations in this work, the modularity of the implementation allows for the extension of the formulation in dynamic contexts [34, 35], where the module to compute the nonlinear elastic forces can be re-used.

We close by listing future challenges and perspectives. Even though each flexel can capture complexity, its response is only governed by a single geometric measure, which can be an oversimplification of reality. Being able to define a flexel with force-displacement curves along more than one dimension could help build more abstract and reduced models. This can in principle be done in the current formulation by defining energy potentials with additional dimensions. For example, flexels whose energy depends on two deformation measures could help construct 2d metamaterial models where each unit cell is a flexel [36]. Flexels with a force-displacement curve that can be tuned via an external parameter could help model the interplay between active materials and geometric nonlinearities [37]. We believe that embracing the flexel formulation will simplify simulations, help gain insight into a large variety of nonlinear problems and strengthen the use of nonlinearities and instabilities as design paradigm for compliant mechanism, soft robots, mechanical metamaterials and nonlinear structures.

## Data availability

The ‘flexel’ ecosystem is available on PyPI: <https://pypi.org/project/springable/1.0.0/> (see SI section 5 for installation instructions), source code: <https://github.com/ducarme/springable/releases/tag/v1.0.0>.

## Acknowledgment

We thank Michel Habets, Micha Steur, and Pavel Antonov for the useful discussions. This work is part of the Dutch Research Council (NWO) and was conducted at the AMOLF institute and the Advanced Research Center for Nanolithography. The Advanced Research Center for Nanolithography is a public–private partnership between the University of Amsterdam, Vrije Universiteit Amsterdam, Rijksuniversiteit Groningen, The Netherlands Organization for Scientific Research (NWO), and the semiconductor-equipment manufacturer Advanced Semiconductor Materials Lithography (ASML).

## References

- [1] Pedro M. Reis. A Perspective on the Revival of Structural (In)Stability With Novel Opportunities for Function: From Buckliphobia to Buckliphilia. *Journal of Applied Mechanics*, 82(111001), September 2015.
- [2] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin van Hecke. Flexible mechanical metamaterials. *Nature Reviews Materials*, 2(11):17066, October 2017. Publisher: Nature Publishing Group.

- [3] Larry L. Howell. Compliant Mechanisms. In *Encyclopedia of Nanotechnology*, pages 457–463. Springer, Dordrecht, 2012.
- [4] Benjamin Gorissen, Edoardo Milana, Arne Baeyens, Eva Broeders, Jeroen Christiaens, Klaas Collin, Dominiek Reynaerts, and Michael De Volder. Hardware Sequencing of Inflatable Nonlinear Actuators for Autonomous Soft Robots. *Advanced Materials*, 31(3):1804598, 2019. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201804598>.
- [5] Amit Nagarkar, Won-Kyu Lee, Daniel J. Preston, Markus P. Nimitz, Nan-Nan Deng, George M. Whitesides, and L. Mahadevan. Elastic-instability-enabled locomotion. *Proceedings of the National Academy of Sciences*, 118(8):e2013801118, February 2021. Publisher: Proceedings of the National Academy of Sciences.
- [6] Richard Baumgartner, Alexander Kogler, Josef M. Stadlbauer, Choon Chiang Foo, Rainer Kaltseis, Melanie Baumgartner, Guoyong Mao, Christoph Keplinger, Soo Jin Adrian Koh, Nikita Arnold, Zhigang Suo, Martin Kaltenbrunner, and Siegfried Bauer. A Lesson from Plants: High-Speed Soft Robotic Actuators. *Advanced Science*, 7(5):1903391, 2020. \_eprint: <https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/advs.201903391>.
- [7] Yichao Tang, Yinding Chi, Jiefeng Sun, Tzu-Hao Huang, Omid H. Maghsoudi, Andrew Spence, Jianguo Zhao, Hao Su, and Jie Yin. Leveraging elastic instabilities for amplified performance: Spine-inspired high-speed and high-force soft robots. *Science Advances*, 6(19):eaaz6912, May 2020. Publisher: American Association for the Advancement of Science.
- [8] Zuolin Liu, Hongbin Fang, Jian Xu, and Kon-Well Wang. Cellular Automata Inspired Multistable Origami Metamaterials for Mechanical Learning. *Advanced Science*, 10(34):2305146, 2023. \_eprint: <https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/advs.202305146>.
- [9] Sen Yan, Wenlong Liu, Xiaojun Tan, Zhiqiang Meng, Weijia Luo, Hang Jin, Yongzheng Wen, Jingbo Sun, Lingling Wu, and Ji Zhou. Bio-inspired mechanical metamaterial with ultrahigh load-bearing capacity for energy dissipation. *Materials Today*, 77:11–18, August 2024.
- [10] Ran Zhang, Thomas Auzinger, and Bernd Bickel. Computational Design of Planar Multistable Compliant Structures. *ACM Trans. Graph.*, 40(5):186:1–186:16, October 2021.
- [11] Brandon Caasenbrood, Alexander Pogromsky, and Henk Nijmeijer. Energy-Shaping Controllers for Soft Robot Manipulators Through Port-Hamiltonian Cosserat Models. *SN Computer Science*, 3(6):494, September 2022.
- [12] Xiaotian Zhang, Fan Kiat Chan, Tejaswin Parthasarathy, and Mattia Gazzola. Modeling and simulation of complex dynamic musculoskeletal architectures. *Nature Communications*, 10(1):4825, October 2019. Publisher: Nature Publishing Group.

- [13] T. Schioler and S. Pellegrino. Space Frames with Multiple Stable Configurations. *AIAA Journal*, 45(7), 2007.
- [14] Ahmad Rafsanjani, Abdolhamid Akbarzadeh, and Damiano Pasini. Snapping Mechanical Metamaterials under Tension. *Advanced Materials*, 27(39):5931–5935, 2015. [\\_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201502809](https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201502809).
- [15] David Dykstra, Joris Busink, Bernard Ennis, and Corentin Coulais. Viscoelastic Snapping Metamaterials. *Journal of Applied Mechanics*, 86:1, June 2019.
- [16] Emma Steinhardt, Nak-seung P. Hyun, Je-sung Koh, Gregory Freeburn, Michelle H. Rosen, Fatma Zeynep Temel, S. N. Patek, and Robert J. Wood. A physical model of mantis shrimp for exploring the dynamics of ultrafast systems. *Proceedings of the National Academy of Sciences*, 118(33):e2026833118, August 2021. Publisher: Proceedings of the National Academy of Sciences.
- [17] Zhiqiang Meng, Weitong Chen, Tie Mei, Yuchen Lai, Yixiao Li, and C. Q. Chen. Bistability-based foldable origami mechanical logic gates. *Extreme Mechanics Letters*, 43:101180, February 2021.
- [18] Adam Bekele, M. Ahmer Wadee, and Andrew T. M. Phillips. Enhancing energy absorption through sequential instabilities in mechanical metamaterials. *Royal Society Open Science*, 10(8):230762, August 2023. Publisher: Royal Society.
- [19] Malte A. ten Wolde and Davood Farhadi. A single-input state-switching building block harnessing internal instabilities. *Mechanism and Machine Theory*, 196:105626, June 2024.
- [20] Martin van Hecke. Profusion of transition pathways for interacting hysterons. *Physical Review E*, 104(5):054608, November 2021. Publisher: American Physical Society.
- [21] Jingran Liu, Margot Teunisse, George Korovin, Ivo R. Vermaire, Lishuai Jin, Hadrien Bense, and Martin van Hecke. Controlled pathways and sequential information processing in serially coupled mechanical hysterons. *Proceedings of the National Academy of Sciences*, 121(22):e2308414121, May 2024. Publisher: Proceedings of the National Academy of Sciences.
- [22] Dor Shohat and Martin van Hecke. Geometric Control and Memory in Networks of Hysteretic Elements. *Physical Review Letters*, 134(18):188201, May 2025. Publisher: American Physical Society.
- [23] Dor Shohat, Paul Baconnier, Itamar Procaccia, Martin van Hecke, and Yoav Lahini. Aging of amorphous materials under cyclic strain, June 2025. arXiv:2506.08779 [cond-mat].
- [24] E. Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 15(7):529–551, January 1979.

- [25] Patrick E. Farrell, Casper H. L. Beentjes, and Ásgeir Birkiisson. The computation of disconnected bifurcation diagrams, 2016.
- [26] Paul Ducarme, Bart Weber, Martin van Hecke, and Johannes T. B. Overvelde. Exotic mechanical properties enabled by countersnapping instabilities. *Proceedings of the National Academy of Sciences*, 122(16):e2423301122, April 2025. Publisher: Proceedings of the National Academy of Sciences.
- [27] Paulina Obara, Joanna Kłosowska, and Wojciech Gilewski. Truth and Myths about 2D Tensegrity Trusses. *Applied Sciences*, 9(1):179, January 2019. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.
- [28] Gengzhi He, Curtis Sparks, and Nick Gravish. Grasping and rolling in-plane manipulation using deployable tape spring appendages. *Science Advances*, 11(15):eadt5905, April 2025. Publisher: American Association for the Advancement of Science.
- [29] Lennard J. Kwakernaak, Arman Guerra, Douglas P. Holmes, and Martin van Hecke. The collective snapping of a pair of bumping buckled beams. *Extreme Mechanics Letters*, 69:102160, June 2024.
- [30] Arman Guerra, Anja C. Slim, Douglas P. Holmes, and Ousmane Kodio. Self-ordering of buckling, bending, and bumping beams. *Phys. Rev. Lett.*, 130:148201, Apr 2023.
- [31] Adel Djellouli, Bert Van Raemdonck, Yang Wang, Yi Yang, Anthony Caillaud, David Weitz, Shmuel Rubinstein, Benjamin Gorissen, and Katia Bertoldi. Shell buckling for programmable metafluids. *Nature*, 628(8008):545–550, April 2024. Publisher: Nature Publishing Group.
- [32] Lennard J. Kwakernaak and Martin van Hecke. Counting and sequential information processing in mechanical metamaterials. *Phys. Rev. Lett.*, 130:268204, Jun 2023.
- [33] Paul Ducarme. springable. available on PyPI: <https://pypi.org/project/springable/1.0.0/>, source code: <https://github.com/ducarme/springable>, 2025. Accessed: 2025-08-27.
- [34] Lishuai Jin, Romik Khajehtourian, Jochen Mueller, Ahmad Rafsanjani, Vincent Tournat, Katia Bertoldi, and Dennis M. Kochmann. Guided transition waves in multistable mechanical metamaterials. *Proceedings of the National Academy of Sciences*, 117(5):2319–2325, 2020.
- [35] Junghwan Byun, Aniket Pal, Jongkuk Ko, and Metin Sitti. Integrated mechanical computing for autonomous soft machines. *Nature Communications*, 15(1), April 2024.
- [36] Katia Bertoldi, Pedro M. Reis, Stephen Willshaw, and Tom Mullin. Negative Poisson’s ratio behavior induced by an elastic instability. *Advanced Materials (Deerfield Beach, Fla.)*, 22(3):361–366, January 2010.

- [37] Aniket Pal and Metin Sitti. Programmable mechanical devices through magnetically tunable bistable elements. *Proceedings of the National Academy of Sciences*, 120(15):e2212489120, 2023.

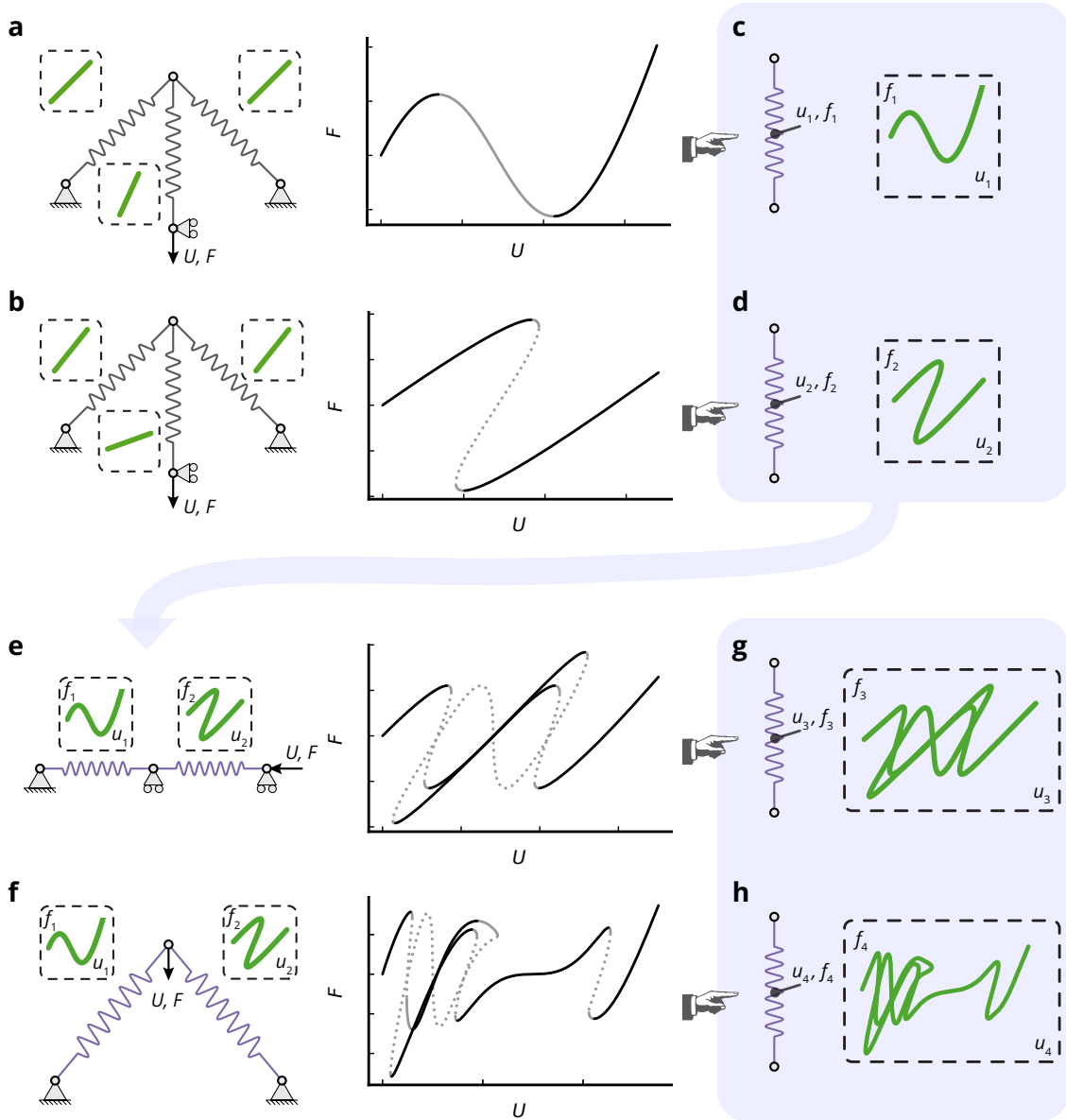


Figure 1: Construction of flexels. (a-b) Von-Mises truss composed of a pair of inclined linear springs (non-dimensional stiffness of 0.6 for (a), 1.0 for (b)) driven from their connection point through a third, vertical linear spring (stiffness of 20 for (a), 0.33 for (b)). The force-displacement response of the system is either (a) non-monotonic or (b) multi-valued. (c-d) Flexel equivalents of the Von-Mises trusses shown in (a) and (b), whose mechanical behavior has been tuned to mimic their force-displacement response. (e-f) Assemblies composed of nonmonotonic and multi-valued flexels as shown in (c) and (d) loaded in series (e) or at an angle (f), exhibiting complex force-displacement curves. (g-h) Flexel equivalents of the systems shown in (e) and (f). Black (gray) lines refer to states stable (unstable) under force-controlled conditions. Solid (dashed) lines refer to states stable (unstable) under displacement-driven conditions. The full descriptions of the models are provided in SI section 7.1.

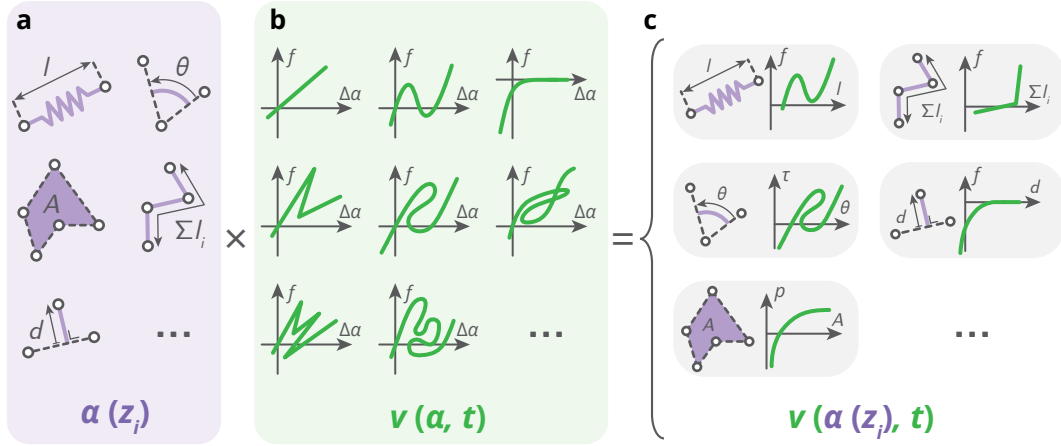


Figure 2: Flexel ecosystem. (a) Various geometric measures  $\alpha$  computed from a list of node coordinates  $z_i$ : length, angle, area, total length of a polygonal chain, distance point-line. (b) Various generalized force-displacement curves (linear, nonlinear, multi-valued) defining intrinsic mechanical behaviors via energy potentials  $v(\alpha, t)$ . (c) Examples of pairs of geometric measure and behavior forming the flexel ecosystem. The energy of a flexel is given by  $v(\alpha(z_i), t)$ . More details on the definition of the geometric measures  $\alpha$  and energy potentials  $v$  are provided in SI section 3 and 4.

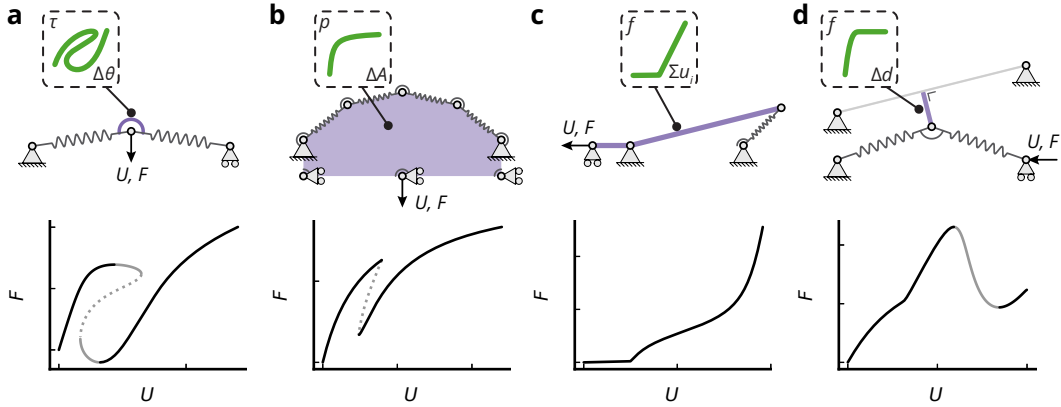


Figure 3: Examples of assemblies of flexels (top) that collectively produce force-displacement responses  $F(U)$  (bottom). Black (gray) lines refer to states stable (unstable) under force-controlled conditions. Solid (dashed) lines refer to states stable (unstable) under displacement-driven conditions. Gray flexels are characterized by a linear generalized force-displacement curve. (a) An angular flexel with a multi-valued torque-angular displacement curve modeling a snapping flexure. (b) An area flexel with a softening pressure-areal displacement curve modeling pneumatic actuation. (c) A path flexel with a bilinear stiffening force-displacement curve, modeling cable-driven actuation. (d) A distance flexel with a force-displacement curve yielding a high nonzero force only for small distance values, modeling contact between a point and a line. The full descriptions of the models are provided in SI section 7.2.

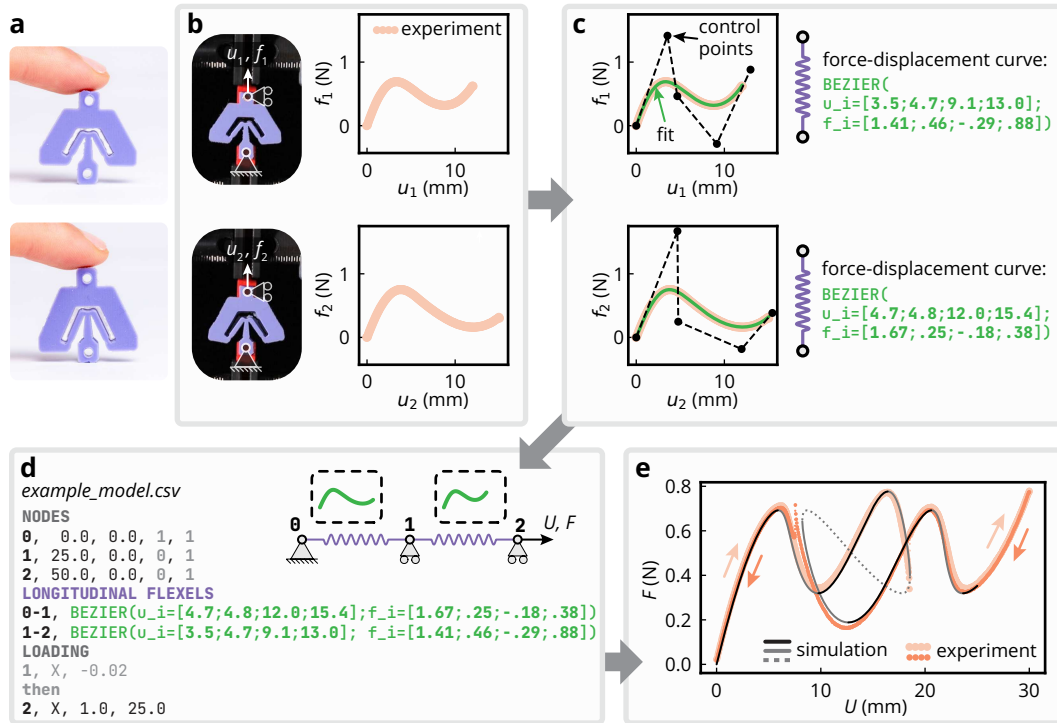


Figure 4: Workflow to simulate an experimental assembly of nonlinear building blocks [26]. (a) Two building blocks fabricated in silicone rubber. (b) Experimental force-displacement curves obtained by performing a tensile test on the building blocks. (c) Left: Bezier curves (solid green lines) fitting the experimental force-displacement curves (orange lines). The control polygons and the control points are depicted by the black dashed lines and the black dots. Right: Equivalent flexels with tensile behaviors defined by Bezier curves, specified by a string of text listing the coordinates of the control points (green text). (d) Flexel model describing the assembly of the serially-coupled building blocks (Right) and the input text file representing the model (Left). (e) Force-displacement curve of the serially-coupled assembly. Experimental data obtained during the loading (unloading) phase is depicted by light (bright) orange dots. Simulated data is depicted by black and gray curves, where black (gray) refers to states stable (unstable) under force-controlled conditions, and solid (dashed) curves refer to states stable (unstable) under displacement-driven conditions. Force and displacement are measured with respect to the preloaded configuration. The full descriptions of the models are provided in SI section 7.3.

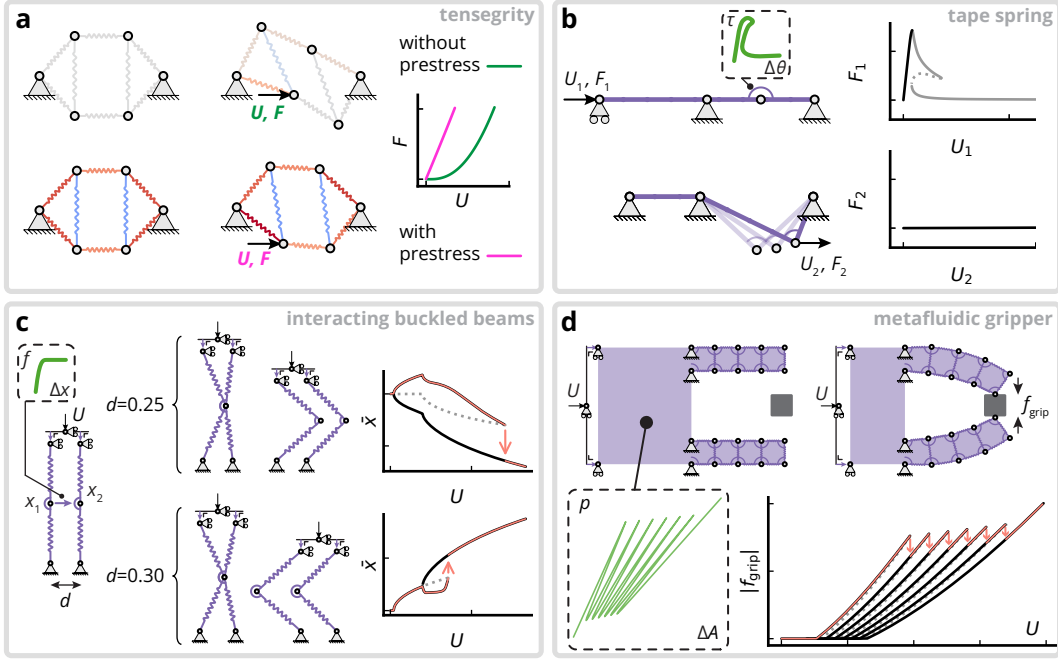


Figure 5: Examples of use cases. (a) Left: Tensegrity structures without prestress (top) and with prestress (bottom) subject to an external loading. Flexels in compression, at rest and in tension are colored in blue, gray and red respectively. Right: Force-displacement curves of the tensegrity structures with and without prestress. (b) Left: Model of a tape-spring gripper [28] loaded in two steps. The first one applies compression to the tape spring, eventually triggering buckling and creating a kink (top). The second deforms the buckled tape spring by driving the kink (bottom). Right: Force-displacement curves corresponding to the first (top) and second (bottom) loadsteps. (c) Left: model of a pair of buckling beams separated by a distance  $d$ , loaded in compression and interacting through the contact of their middle node [32, 30, 29]. Center: deformation sequence of the beams when  $d = 0.25$  (top) and  $d = 0.30$  (bottom). Right: Deformation paths when  $d = 0.25$  (top) and  $d = 0.30$  (bottom).  $\bar{x} := (x_1 + x_2)/2$ . (d) Top: Model of a gripper actuated using a metafluid [31] to grasp an object. Bottom: Intrinsic behavior of the area flexel modeling the metafluid (left) and deformation path of the system (right), shown as the gripping force  $f_{\text{grip}}$  as a function of the applied displacement  $U$ . The full descriptions of the models are provided in SI section 7.4.

Supplementary information for preprint  
"Flexel ecosystem: simulating mechanical systems made entities with  
arbitrarily complex mechanical responses"

Paul Ducarme, Bart Weber, Martin van Hecke, Johannes T.B. Overvelde

October 23, 2025

## Contents

<b>1</b>	<b>Flexel formulation</b>	<b>3</b>
1.1	Individual flexel . . . . .	3
1.2	Assembly of flexels . . . . .	5
<b>2</b>	<b>Path-following algorithm</b>	<b>6</b>
2.1	Model descriptions . . . . .	6
2.2	Algorithm . . . . .	7
<b>3</b>	<b>Geometric measures, gradients and Hessians</b>	<b>11</b>
3.1	Length (longitudinal flexel) . . . . .	11
3.2	Angle (angular flexel) . . . . .	12
3.3	Signed area . . . . .	13
3.4	Area (area flexel) . . . . .	14
3.5	Signed $x$ -distance and $y$ -distance ( $x$ -distance flexel and $y$ -distance flexel) . . . . .	15
3.6	Compound measure . . . . .	15
3.7	Signed point-line distance (distance flexel) . . . . .	16
3.8	Path length (path flexel) . . . . .	17
3.9	Area with holes (area flexel) . . . . .	18
<b>4</b>	<b>Intrinsic nonlinear mechanical behaviors</b>	<b>18</b>
4.1	Univariate behaviors . . . . .	18
4.2	Bivariate behavior . . . . .	26
<b>5</b>	<b>Installation and simulation</b>	<b>33</b>
5.1	Installation . . . . .	33
5.2	Simulating a model . . . . .	33
<b>6</b>	<b>Model file specifications</b>	<b>34</b>
6.1	The NODES section . . . . .	34
6.2	The FLEXELS sections . . . . .	35
6.3	The LOADING section . . . . .	40
6.4	The PARAMETERS section . . . . .	41

<b>7</b>	<b>Model descriptions</b>	<b>43</b>
7.1	Figure 1 . . . . .	43
7.2	Figure 3 . . . . .	46
7.3	Figure 4 . . . . .	50
7.4	Figure 5 . . . . .	51

# 1 Flexel formulation

To simulate a mechanical structure quasi-statically, the strategy consists of minimizing the total potential energy at varying load levels, as more formally described in section 2. Essentially, this is done by iteratively finding configurations wherein internal elastic forces balance external ones. The process requires computing both the elastic forces and the way they vary with structural deformations. The flexel formulation that we present in this section is divided in two parts. First, we derive the equations for computing the elastic force and its variation for an individual flexel. Then, we show how to combine these individual contributions to yield the global force and variation expressions for an assembly of flexels.

## 1.1 Individual flexel

The elastic force of a flexel and how this force varies can be derived from its potential energy  $e$ . For a flexel, the potential energy is the stored elastic energy, which depends on its the geometric measure  $\alpha$ , that is, its length if it is a longitudinal flexel, its angle if an angular flexel, its area if an area flexel, etc. The geometric measure  $\alpha$  is computed from the spatial coordinates  $\mathbf{z}$  of the nodes coupled by the flexel, using the equations derived in section 3.

More formally, the energy  $e$  of a flexel is described using an energy potential  $v$  which is either a univariate  $v(\cdot)$  or bivariate  $v(\cdot, \cdot)$  function:

$$e(\mathbf{q}) = \begin{cases} v(\alpha(\mathbf{z})) & \text{if } v \text{ is univariate,} \\ v(\alpha(\mathbf{z}), t) & \text{if } v \text{ is bivariate,} \end{cases} \quad (1)$$

where  $\mathbf{q}$  is the subset of the structural coordinates  $\mathbf{Q}$  (that describe the state of an entire assembly) used by the flexel and  $\alpha$  is the geometric measure computed from the spatial coordinates  $\mathbf{z} \subseteq \mathbf{q}$ . For a flexel based on a univariate potential, its energy is solely determined by its geometric measure and its coordinates  $\mathbf{q}$  are all spatial:  $\mathbf{q} = \mathbf{z}$ . For a flexel based on a bivariate potential, the energy depends on the geometric measure  $\alpha$  and an additional (non-spatial) coordinate  $t$ :  $\mathbf{q} = [\mathbf{z}^\top, t]^\top$ .

This extra coordinate  $t$  describes a degree of freedom internal to the flexel that has no direct physical interpretation. It is introduced to define flexels with multi-valued generalized force-displacement curves: equilibrium curves such that to a given geometric measure  $\alpha$  (length, angle, area, etc) can correspond multiple generalized force values (force, torque, 2d-pressure, or etc). This extra coordinate  $t$  disambiguates the state of the flexel, as the value of the geometric measure  $\alpha$  alone would not be enough to determine what the generalized force in the flexel is. At equilibrium, the coordinate  $t$  can be seen as the *curve parameter* of the parametric description of the multi-valued force-displacement curve (Eq. (119)). More details on bivariate potentials and how to construct them from multi-valued force-displacement curves are provided in subsection 4.2.

Different types of univariate and bivariate energy potentials are implemented in `springable`. They are defined in subsection 4.1 and 4.2 respectively.

**Energy gradient (elastic force vector)** The elastic force is described by the gradient of the energy  $e$  with respect to the coordinates  $\mathbf{q}$ . We will denote this vector as  $\mathbf{f}$ , and its expression is given by

$$\begin{aligned} \mathbf{f} &:= \frac{\partial e}{\partial \mathbf{q}} = [f_i] = \begin{bmatrix} \frac{\partial e}{\partial q_i} \end{bmatrix} = \frac{\partial v}{\partial \alpha} \begin{bmatrix} \frac{\partial \alpha}{\partial q_i} \end{bmatrix} + \frac{\partial v}{\partial t} \begin{bmatrix} \frac{\partial t}{\partial q_i} \end{bmatrix} \\ &= \frac{\partial v}{\partial \alpha} \begin{bmatrix} \frac{\partial \alpha}{\partial z_0} \\ \vdots \\ \frac{\partial \alpha}{\partial z_{n-2}} \\ 0 \end{bmatrix} + \frac{\partial v}{\partial t} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\ &= \frac{\partial v}{\partial \alpha} \begin{bmatrix} \frac{\partial \alpha}{\partial \mathbf{z}} \\ 0 \end{bmatrix} + \frac{\partial v}{\partial t} \begin{bmatrix} \mathbf{0}^{(n-1) \times 1} \\ 1 \end{bmatrix} \end{aligned} \quad (2)$$

where  $v$  is a bivariate potential (where the  $n$  coordinates used by the flexel are the spatial ones and the internal one, i.e.:  $\mathbf{q} = [\mathbf{z}^\top, t]^\top$ ). In case of univariate potential (where  $\partial v/\partial t = 0$  and all the  $n$  coordinates used by the flexel are spatial, i.e.:  $\mathbf{q} = \mathbf{z}$ ), Eq. (2) simplifies to

$$\mathbf{f} = \frac{\partial v}{\partial \alpha} \left[ \frac{\partial \alpha}{\partial z_i} \right] = \frac{\partial v}{\partial \alpha} \frac{\partial \alpha}{\partial \mathbf{z}} \quad (3)$$

To compute the energy gradient of an individual flexel, we therefore need to separately compute two types of quantities:

- the measure  $\alpha$  and its gradient  $\partial \alpha / \partial \mathbf{z} = [\partial \alpha / \partial z_i]$ , purely geometric quantities which reflects how the *shape* of the flexel (longitudinal, angular, area, etc) couples to the nodes of the structure, and whose equations are derived in section 3;
- the gradient of the potential  $v$ , i.e.  $\partial v / \partial \alpha$  and  $\partial v / \partial t$ , quantities which reflects the *intrinsic nonlinear mechanical behavior* of the flexel, and whose equations are derived in section 4.

**Energy hessian (stiffness matrix)** How the elastic force varies with respect to the state of the flexel is described by the hessian of the energy  $e$  with respect to the coordinates  $\mathbf{q}$  (that is, the gradient of the gradient, or the matrix of the second-order derivatives). This matrix is commonly known as the *stiffness matrix*, and we will denote it as  $\mathbf{k}$ . Its expression is given by

$$\begin{aligned} \mathbf{k} &= \frac{\partial^2 e}{\partial \mathbf{q} \partial \mathbf{q}^\top} = [k_{ij}] = \left[ \frac{\partial^2 e}{\partial q_i \partial q_j} \right] = \left[ \frac{\partial f_i}{\partial q_j} \right] \\ &= \frac{\partial^2 v}{\partial \alpha^2} \left[ \frac{\partial \alpha}{\partial q_i} \frac{\partial \alpha}{\partial q_j} \right] + \frac{\partial^2 v}{\partial \alpha \partial t} \left[ \frac{\partial \alpha}{\partial q_i} \frac{\partial t}{\partial q_j} + \frac{\partial t}{\partial q_i} \frac{\partial \alpha}{\partial q_j} \right] + \frac{\partial v}{\partial \alpha} \left[ \frac{\partial^2 \alpha}{\partial q_i \partial q_j} \right] \\ &\quad + \frac{\partial^2 v}{\partial t^2} \left[ \frac{\partial t}{\partial q_i} \frac{\partial t}{\partial q_j} \right] + \frac{\partial v}{\partial t} \left[ \frac{\partial^2 t}{\partial q_i \partial q_j} \right] \\ &= \frac{\partial^2 v}{\partial \alpha^2} \begin{bmatrix} \frac{\partial \alpha}{\partial z_0} \frac{\partial \alpha}{\partial z_0} & \dots & \frac{\partial \alpha}{\partial z_0} \frac{\partial \alpha}{\partial z_{n-2}} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial \alpha}{\partial z_{n-2}} \frac{\partial \alpha}{\partial z_0} & \dots & \frac{\partial \alpha}{\partial z_{n-2}} \frac{\partial \alpha}{\partial z_{n-2}} & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix} + \frac{\partial v}{\partial \alpha} \begin{bmatrix} \frac{\partial^2 \alpha}{\partial z_0 \partial z_0} & \dots & \frac{\partial^2 \alpha}{\partial z_0 \partial z_{n-2}} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial^2 \alpha}{\partial z_{n-2} \partial z_0} & \dots & \frac{\partial^2 \alpha}{\partial z_{n-2} \partial z_{n-2}} & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix} \\ &\quad + \frac{\partial^2 v}{\partial \alpha \partial t} \begin{bmatrix} 0 & \dots & 0 & \frac{\partial \alpha}{\partial z_0} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \frac{\partial \alpha}{\partial z_{n-2}} \\ \frac{\partial \alpha}{\partial z_0} & \dots & \frac{\partial \alpha}{\partial z_{n-2}} & 0 \end{bmatrix} + \frac{\partial^2 v}{\partial t^2} \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \\ &= \frac{\partial^2 v}{\partial \alpha^2} \begin{bmatrix} \frac{\partial \alpha}{\partial \mathbf{z}} \frac{\partial \alpha}{\partial \mathbf{z}^\top} & \mathbf{0}^{(n-1) \times 1} \\ \mathbf{0}^{1 \times (n-1)} & 0 \end{bmatrix} + \frac{\partial v}{\partial \alpha} \begin{bmatrix} \frac{\partial^2 \alpha}{\partial \mathbf{z} \partial \mathbf{z}^\top} & \mathbf{0}^{(n-1) \times 1} \\ \mathbf{0}^{1 \times (n-1)} & 0 \end{bmatrix} \\ &\quad + \frac{\partial^2 v}{\partial \alpha \partial t} \begin{bmatrix} \mathbf{0}^{(n-1) \times (n-1)} & \frac{\partial \alpha}{\partial \mathbf{z}} \\ \frac{\partial \alpha}{\partial \mathbf{z}^\top} & 0 \end{bmatrix} + \frac{\partial^2 v}{\partial t^2} \begin{bmatrix} \mathbf{0}^{(n-1) \times (n-1)} & \mathbf{0}^{(n-1) \times 1} \\ \mathbf{0}^{1 \times (n-1)} & 1 \end{bmatrix} \quad (4) \end{aligned}$$

where  $v$  is a bivariate potential (where the  $n$  coordinates used by the flexel are the spatial ones and the internal one, i.e.:  $\mathbf{q} = [\mathbf{z}^\top, t]^\top$ ). In case of a univariate potential (where  $\partial^2 v / \partial \alpha \partial t = \partial v^2 / \partial t^2 = 0$  and all the  $n$  coordinates used by the flexel are spatial, i.e.:  $\mathbf{q} = \mathbf{z}$ ), Eq. (4) simplifies to

$$\mathbf{k} = \frac{\partial^2 v}{\partial \alpha^2} \begin{bmatrix} \partial \alpha & \partial \alpha \\ \partial z_i & \partial z_j \end{bmatrix} + \frac{\partial v}{\partial \alpha} \begin{bmatrix} \partial^2 \alpha \\ \partial z_i \partial z_j \end{bmatrix} = \frac{\partial^2 v}{\partial \alpha^2} \frac{\partial \alpha}{\partial \mathbf{z}} \frac{\partial \alpha}{\partial \mathbf{z}^\top} + \frac{\partial v}{\partial \alpha} \frac{\partial^2 \alpha}{\partial \mathbf{z} \partial \mathbf{z}^\top}. \quad (5)$$

Similarly to the energy gradient, to compute the energy hessian, we need to separately compute quantities describing the geometry and quantities describing the intrinsic nonlinear behavior:

- the geometric measure, its gradient and its hessian,  $\alpha$ ,  $\partial \alpha / \partial \mathbf{z} = [\partial \alpha / \partial z_i]$  and  $\partial^2 \alpha / \partial \mathbf{z} \partial \mathbf{z}^\top = [\partial^2 \alpha / \partial z_i \partial z_j]$ , purely geometric quantities which reflects how the *shape* of the flexel (longitudinal, angular, area, etc) couples to the rest of the structure, and whose equations are derived in section 3;
- the gradient and hessian of the potential,  $\partial v / \partial \alpha$ ,  $\partial^2 v / \partial \alpha^2$ ,  $\partial^2 v / \partial \alpha \partial t$  and  $\partial^2 v / \partial t^2$ , which reflects the *intrinsic nonlinear mechanical behavior* of the flexel, and whose equations are derived in section 4.

## 1.2 Assembly of flexels

The elastic force of an assembly of flexels and how this force varies can be derived from its elastic energy  $E$ . The total elastic energy  $E$  stored in the structure is the sum of each flexel's elastic energy:

$$E(\mathbf{Q}) = \sum_{s=0}^{N_f-1} e_s(\mathbf{q}^s), \quad (6)$$

where  $\mathbf{Q}$  is the set of all the structural coordinates, which describe the state of the assembly,  $N_f$  is the number of flexels in the assembly,  $e_s$  is the energy of flexel  $s$  (Eq. (1)) and  $\mathbf{q}^s \subseteq \mathbf{Q}$  is the subset of structural coordinates used by flexel  $s$ .

**Energy gradient (elastic force vector)** The elastic force is described by the gradient of the energy  $E$  with respect to the coordinates  $\mathbf{Q}$ . We will denote this vector as  $\mathbf{F}$ , and its expression is given by

$$\mathbf{F} = \frac{\partial E}{\partial \mathbf{Q}} = \left[ \frac{\partial E}{\partial Q_k} \right] = \sum_{s=0}^{N_f-1} \left[ \frac{\partial e_s(\mathbf{q}^s)}{\partial Q_k} \right] = \sum_{s=0}^{N_f-1} \left( \sum_{i=0}^{n_s-1} \frac{\partial e_s}{\partial q_i^s} \left[ \frac{\partial q_i^s}{\partial Q_k} \right] \right) = \sum_{s=0}^{N_f-1} \left( \sum_{i=0}^{n_s-1} \frac{\partial e_s}{\partial q_i^s} [L_{ik}^s] \right), \quad (7)$$

where  $n_s$  is the number of coordinates in  $\mathbf{q}^s$ ,  $\partial e_s / \partial q_i^s$  are the components of the energy gradient  $\mathbf{f}^s$  of flexel  $s$  (Eq. (2)), and where

$$L_{ik}^s = \begin{cases} 1 & \text{if } q_i^s = Q_k, \\ 0 & \text{else.} \end{cases} \quad (8)$$

In practical terms, Eq. (7) means that the gradient  $\mathbf{F} = [F_k]$  is constructed by adding together the components of each individual flexel's energy gradient  $\mathbf{f}^s = [f_i^s] = [\partial e_s / \partial q_i^s]$  that correspond to the same coordinate  $Q_k$ .

**Energy hessian (stiffness matrix)** How the elastic force varies with respect to the state of the assembly is described by the hessian of the energy  $E$  with respect to the coordinates  $\mathbf{Q}$  (that is, the gradient of the gradient, or the matrix of the second-order derivatives). This matrix is commonly known as the *stiffness matrix*, and we will denote it as  $\mathbf{K}$ . Its expression is given by:

$$\begin{aligned} \mathbf{K} &= \frac{\partial^2 E}{\partial \mathbf{Q} \partial \mathbf{Q}^\top} = [K_{kl}] = \left[ \frac{\partial^2 E}{\partial Q_k \partial Q_l} \right] = \left[ \frac{\partial F_k}{\partial Q_l} \right] = \sum_{s=0}^{N_f-1} \left[ \frac{\partial^2 e_s(\mathbf{q}^s)}{\partial Q_k \partial Q_l} \right] \\ &= \sum_{s=0}^{N_f-1} \left( \sum_{i=0}^{n_s-1} \sum_{j=0}^{n_s-1} \frac{\partial^2 e_s}{\partial q_i^s \partial q_j^s} \left[ \frac{\partial q_i^s}{\partial Q_k} \frac{\partial q_j^s}{\partial Q_l} \right] \right) = \sum_{s=0}^{N_f-1} \left( \sum_{i=0}^{n_s-1} \sum_{j=0}^{n_s-1} \frac{\partial^2 e_s}{\partial q_i^s \partial q_j^s} [L_{ik}^s L_{jl}^s] \right), \end{aligned} \quad (9)$$

where  $\partial^2 e_s / \partial q_i^s \partial q_j^s$  are the components of the energy hessian of flexel  $s$  (Eq. (4)), and where  $[L_{ik} L_{jl}]$  is determined using Eq. (8). In practical terms, Eq. (9) means that the energy hessian  $\mathbf{K} = [K_{kl}]$  is constructed by adding together the components of each individual flexel's energy hessian  $\mathbf{k}^s = [k_{ij}^s] = [\partial e_s / \partial q_i^s \partial q_j^s]$  that correspond to the same pair of coordinates  $(Q_k, Q_l)$ .

## 2 Path-following algorithm

First, we provide a general mathematical description of the mechanical systems that we aim to model and simulate, and express the governing equations that need to be solved by the algorithm to simulate such systems. Using that formalism, we then describe the algorithm itself.

### 2.1 Model descriptions

Let us consider a mechanical system (or structure), defined as a set of nodes coupled by flexels. The state of the system is defined by  $N$  coordinates gathered in the array  $\mathbf{Q}$ . Among these  $N$  coordinates,  $M$  are free, while the other are fixed. The free coordinates are denoted by  $\tilde{\mathbf{Q}}$ , which is the array  $\mathbf{Q}$  without the fixed coordinates. The objective of the simulation is to calculate the equilibrium states of the structure from a given initial load up to a final prescribed load, acting on the free coordinates  $\tilde{\mathbf{Q}}$ . To this end, we first consider the total potential energy  $\Pi$  of the system:

$$\Pi = E - \tilde{\mathbf{F}}^{\text{ext}} \cdot \tilde{\mathbf{U}} = E - \sum_{k=0}^{M-1} \tilde{F}_k^{\text{ext}} \tilde{U}_k, \quad (10)$$

where  $E$  is the elastic energy stored in the system and  $\tilde{\mathbf{F}}^{\text{ext}} \cdot \tilde{\mathbf{U}}$  is the combined work done by each external force  $\tilde{F}_k^{\text{ext}}$  applied along the displacement  $\tilde{U}_k = \Delta \tilde{Q}_k$  of the free coordinate  $\tilde{Q}_k$ . Since equilibrium states under the external load  $\tilde{\mathbf{F}}^{\text{ext}}$  are stationary points of the total potential energy with respect to the free coordinates  $\tilde{\mathbf{Q}}$ , they are solutions of the following system of  $M$  equations

$$\begin{aligned} \frac{\partial E}{\partial \tilde{\mathbf{Q}}}(\mathbf{Q}) - \tilde{\mathbf{F}}^{\text{ext}} &= 0 & (11) \\ \Leftrightarrow & \\ \left\{ \begin{array}{l} \frac{\partial E}{\partial \tilde{Q}_0}(\mathbf{Q}) - \tilde{F}_0^{\text{ext}} = 0 \\ \vdots \\ \frac{\partial E}{\partial \tilde{Q}_k}(\mathbf{Q}) - \tilde{F}_k^{\text{ext}} = 0 \\ \vdots \\ \frac{\partial E}{\partial \tilde{Q}_{M-1}}(\mathbf{Q}) - \tilde{F}_{M-1}^{\text{ext}} = 0. \end{array} \right. & (12) \end{aligned}$$

Each equation in this system represents the balance between the internal elastic forces and the external forces on coordinate  $\tilde{Q}_k$  that must be achieved by a configuration to be in a static equilibrium under the external forces  $\tilde{\mathbf{F}}^{\text{ext}}$ . Since we are interested in solving this system for various levels of loading from the initial given load to the final prescribed load (that we will now denote as  $\tilde{\mathbf{F}}^{\text{ext},0}$  and  $\tilde{\mathbf{F}}^{\text{ext}, \text{final}}$ ), the system

can be rewritten in the following parametrized form

$$\begin{aligned} \frac{\partial E}{\partial \tilde{\mathbf{Q}}}(\mathbf{Q}) - \tilde{\mathbf{F}}^{\text{ext}} &= \frac{\partial E}{\partial \tilde{\mathbf{Q}}}(\mathbf{Q}) - (\tilde{\mathbf{F}}^{\text{ext},0} + \lambda \tilde{\mathbf{F}}^{\text{dir}}) = 0 & (13) \\ \Leftrightarrow & \\ \left\{ \begin{array}{l} \frac{\partial E}{\partial \tilde{Q}_0}(\mathbf{Q}) - (\tilde{F}_0^{\text{ext},0} + \lambda \tilde{F}_0^{\text{dir}}) = 0 \\ \vdots \\ \frac{\partial E}{\partial \tilde{Q}_k}(\mathbf{Q}) - (\tilde{F}_k^{\text{ext},0} + \lambda \tilde{F}_k^{\text{dir}}) = 0 \\ \vdots \\ \frac{\partial E}{\partial \tilde{Q}_{M-1}}(\mathbf{Q}) - (\tilde{F}_{M-1}^{\text{ext},0} + \lambda \tilde{F}_{M-1}^{\text{dir}}) = 0. \end{array} \right. & \end{aligned}$$

where

$$\tilde{\mathbf{F}}^{\text{dir}} := \frac{\tilde{\mathbf{F}}^{\text{step}}}{\|\tilde{\mathbf{F}}^{\text{step}}\|} \quad (14)$$

is the direction of the step load vector  $\tilde{\mathbf{F}}^{\text{step}} := \tilde{\mathbf{F}}^{\text{ext, final}} - \tilde{\mathbf{F}}^{\text{ext},0}$  (which represents the additional load needed to reach the final prescribed load from the initial load), and where  $\lambda$  is the load parameter, which is initially 0 and reaches  $\|\tilde{\mathbf{F}}^{\text{step}}\|$  when the load step is completed. Eq. (13) represents a system of parametrized nonlinear equations, which can be solved using numerical continuation algorithms.

## 2.2 Algorithm

In **springable**, we solve Eq. (13) using the arc-length continuation scheme introduced in [1, 2], as it is able to reliably calculate the entire succession of equilibrium states even if those form a branch with turning points, unlike more naive approaches that consists of iteratively increasing the parameter  $\lambda$  and applying Newton's method. The algorithm consists of computing successive increments, each performing one prediction, followed by multiple corrections to converge towards an equilibrium state. The algorithm proceeds as follows. Note:  $x \leftarrow y$  means that  $y$  is assigned to the variable  $x$ .

1. The inputs of the problem are provided: initial coordinates  $\mathbf{Q}^0$ , initial external forces  $\tilde{\mathbf{F}}^{\text{ext},0}$ , step load vector  $\tilde{\mathbf{F}}^{\text{step}}$  (defined as  $\tilde{\mathbf{F}}^{\text{ext, final}} - \tilde{\mathbf{F}}^{\text{ext},0}$ ).
2. The settings of the solver are provided: initial radius  $r_0$ , convergence tolerance  $\varepsilon$ .
3. The direction of of the step load vector is computed:  $\tilde{\mathbf{F}}^{\text{dir}} := \tilde{\mathbf{F}}^{\text{step}} / \|\tilde{\mathbf{F}}^{\text{step}}\|$ .
4. The variables that will be used and updated by the algorithm are initialized:
  - current radius  $r \leftarrow r_0$ ,
  - load parameter  $\lambda \leftarrow 0$ ,
  - current coordinates  $\mathbf{Q} \leftarrow \mathbf{Q}^0$ ,
  - current external force vector  $\tilde{\mathbf{F}}^{\text{ext}} \leftarrow \tilde{\mathbf{F}}^{\text{ext},0}$ ,
  - list of previously computed equilibrium states  $\mathcal{Q}^{\text{eq}} \leftarrow [\mathbf{Q}^0]$ ,
  - list of external force vectors corresponding to the previously computed equilibrium states  $\tilde{\mathcal{F}}^{\text{eq}} \leftarrow [\tilde{\mathbf{F}}^{\text{ext},0}]$ ,
  - the current hessian matrix  $\mathbf{K}$  is computed based on the coordinates  $\mathbf{Q}^0$  (Eq. (9)), the current reduced hessian matrix  $\tilde{\mathbf{K}}$  is formed by removing the rows and columns corresponding to the fixed coordinates.

5. A new increment is started.

(a) The prediction step is started.

- i. The increment is initialized:  $\Delta \mathbf{U}^{\text{inc}} = \mathbf{0}$ ,  $\Delta \lambda^{\text{inc}} = 0$ .
- ii. Eq. (15) is solved for  $\Delta \hat{\mathbf{U}}$ :

$$\tilde{\mathbf{K}} \Delta \hat{\mathbf{U}} = \tilde{\mathbf{F}}^{\text{dir}}. \quad (15)$$

iii. The quantity  $\Delta \lambda^{\text{ite}}$  is computed [3]:

$$\Delta \lambda^{\text{ite}} = s \frac{r}{\sqrt{\Delta \hat{\mathbf{U}} \cdot \Delta \hat{\mathbf{U}}}}, \quad (16)$$

with

$$s = \begin{cases} 1 & \text{if first increment, or if } \Delta \hat{\mathbf{U}} \cdot \Delta \mathbf{U}^{\text{prev. inc}} \geq 0 \\ -1 & \text{else.} \end{cases} \quad (17)$$

iv. The following quantities are updated:  $\Delta \lambda^{\text{inc}} \leftarrow \Delta \lambda^{\text{inc}} + \Delta \lambda^{\text{ite}}$ , and  $\Delta \mathbf{U}^{\text{inc}} \leftarrow \Delta \mathbf{U}^{\text{inc}} + \Delta \mathbf{U}^{\text{ite}}$ , where

$$\Delta \mathbf{U}^{\text{ite}} = \Delta \lambda^{\text{ite}} \Delta \hat{\mathbf{U}}. \quad (18)$$

- v. The external force vector is updated:  $\tilde{\mathbf{F}}^{\text{ext}} \leftarrow \tilde{\mathbf{F}}^{\text{ext}} + \Delta \lambda^{\text{ite}} \tilde{\mathbf{F}}^{\text{dir}}$ .
- vi. The free coordinates in  $\mathbf{Q}$  are updated:  $\tilde{\mathbf{Q}} \leftarrow \tilde{\mathbf{Q}} + \Delta \mathbf{U}^{\text{ite}}$ .
- vii. The residual  $\tilde{\mathbf{R}}$  of the force balance based on the updated coordinates is computed:

$$\tilde{\mathbf{R}} = \tilde{\mathbf{F}} - \tilde{\mathbf{F}}^{\text{ext}}, \quad (19)$$

where  $\tilde{\mathbf{F}}$  is the array  $\mathbf{F} := \partial E / \partial \mathbf{Q}$  (Eq. (7)) from which the components corresponding to fixed coordinates have been removed. The array  $\mathbf{F}$  represents the internal net force resulting on each coordinate due the elastic forces of the flexels,

viii. The next action is decided based on the convergence criterion:

$$\begin{cases} \text{start a correction step} & \text{if } \frac{\|\tilde{\mathbf{R}}\|}{\|\tilde{\mathbf{F}}^{\text{step}}\|} > \varepsilon, \\ \text{prepare for the next increment} & \text{else.} \end{cases}$$

(b) A correction step is started (if no convergence):

- i. The current hessian matrix  $\mathbf{K}$  is computed based on the updated coordinates  $\mathbf{Q}$  (see Eq. 9), and the current reduced hessian matrix  $\tilde{\mathbf{K}}$  is formed by removing the rows and columns corresponding to the fixed coordinates.
- ii. Eqs. (20) and (21) are solved for  $\Delta \hat{\mathbf{U}}$  and  $\Delta \bar{\mathbf{U}}$  respectively:

$$\tilde{\mathbf{K}} \Delta \hat{\mathbf{U}} = \tilde{\mathbf{F}}^{\text{dir}}, \quad (20)$$

$$\tilde{\mathbf{K}} \Delta \bar{\mathbf{U}} = -\tilde{\mathbf{R}}. \quad (21)$$

iii. The following quantities are computed [3]:

- $a_0 = \Delta \hat{\mathbf{U}} \cdot \Delta \hat{\mathbf{U}}$ ,
- $b_0 = 2 \Delta \mathbf{U}^{\text{inc}} \cdot \Delta \hat{\mathbf{U}}$ ,
- $b_1 = 2 \Delta \bar{\mathbf{U}} \cdot \Delta \hat{\mathbf{U}}$ ,
- $c_0 = \Delta \mathbf{U}^{\text{inc}} \cdot \Delta \mathbf{U}^{\text{inc}} - r^2$ ,
- $c_1 = 2 \Delta \mathbf{U}^{\text{inc}} \cdot \Delta \bar{\mathbf{U}}$ ,

- $c_2 = \Delta\bar{\mathbf{U}} \cdot \Delta\bar{\mathbf{U}}$ ,
- $a = a_0$ ,
- $b = b_0 + b_1$ ,
- $c = c_0 + c_1 + c_2$ ,
- $\rho = b^2 - 4ac$ .

iv. The correction feasibility is assessed, and the next action is decided:

$$\begin{cases} \text{continue} & \text{if } \rho > 0, \\ \text{prepare for increment restart} & \text{else.} \end{cases}$$

v. The quantity  $\Delta\lambda^{\text{ite}}$  is computed:

$$\Delta\lambda^{\text{ite}} = \begin{cases} \frac{-b + \sqrt{\rho}}{2a} & \text{if } b_0 > 0, \\ \frac{-b - \sqrt{\rho}}{2a} & \text{else.} \end{cases} \quad (22)$$

vi. The following quantities are updated:  $\Delta\lambda^{\text{inc}} \leftarrow \Delta\lambda^{\text{inc}} + \Delta\lambda^{\text{ite}}$ , and  $\Delta\mathbf{U}^{\text{inc}} \leftarrow \Delta\mathbf{U}^{\text{inc}} + \Delta\mathbf{U}^{\text{ite}}$ , where

$$\Delta\mathbf{U}^{\text{ite}} \leftarrow \Delta\bar{\mathbf{U}} + \Delta\lambda^{\text{ite}} \Delta\hat{\mathbf{U}}. \quad (23)$$

vii. The external force vector is updated:  $\tilde{\mathbf{F}}^{\text{ext}} \leftarrow \tilde{\mathbf{F}}^{\text{ext}} + \Delta\lambda^{\text{ite}} \tilde{\mathbf{F}}^{\text{dir}}$ .

viii. The free coordinates in  $\mathbf{Q}$  are updated:  $\tilde{\mathbf{Q}} \leftarrow \tilde{\mathbf{Q}} + \Delta\mathbf{U}^{\text{ite}}$ .

ix. The residual  $\tilde{\mathbf{R}}$  of the force balance based on the updated coordinates is computed:

$$\tilde{\mathbf{R}} = \tilde{\mathbf{F}} - \tilde{\mathbf{F}}^{\text{ext}}, \quad (24)$$

where  $\tilde{\mathbf{F}}$  is the array  $\mathbf{F} := \partial E / \partial \mathbf{Q}$  (Eq. (7)) from which the components corresponding to the fixed coordinates have been removed.

x. The next action is decided based on the convergence criterion:

$$\begin{cases} \text{start an additional correction step} & \text{if } \|\tilde{\mathbf{R}}\| > \varepsilon, \\ \text{prepare for the next increment} & \text{else.} \end{cases}$$

(c) The next increment is prepared.

- i. The current hessian matrix  $\mathbf{K}$  is computed based on the coordinates  $\mathbf{Q}$  (Eq. (9)), the current reduced hessian matrix  $\tilde{\mathbf{K}}$  is formed by removing the rows and columns corresponding to the fixed coordinates.
- ii. The following quantities are updated:  $\Delta\mathbf{U}^{\text{prev. inc}} \leftarrow \Delta\mathbf{U}^{\text{inc}}$ , and  $r \leftarrow \min(r_0, 2r)$ .
- iii. The load parameter is updated:  $\lambda \leftarrow \lambda + \Delta\lambda_{\text{inc}}$ .
- iv. The new equilibrium state  $\mathbf{Q}$  is appended to the list  $\mathcal{Q}^{\text{eq}}$  of previously computed equilibrium states:  $\mathcal{Q}^{\text{eq}} \leftarrow \mathcal{Q}^{\text{eq}} + [\mathbf{Q}]$ .
- v. The new external force vector  $\tilde{\mathbf{F}}^{\text{ext}}$  is appended to the list  $\tilde{\mathcal{F}}^{\text{eq}}$  of external force vectors corresponding to previously computed equilibrium states:  $\tilde{\mathcal{F}}^{\text{eq}} \leftarrow \tilde{\mathcal{F}}^{\text{eq}} + [\tilde{\mathbf{F}}^{\text{ext}}]$ .
- vi. The radius is updated:  $r \leftarrow \min(2r, r_0)$ .

vii. The next action is decided based on whether the final load has been reached:

$$\begin{cases} \text{stop (load step has successfully finished)} & \text{if } \lambda \geq \|\tilde{\mathbf{F}}^{\text{step}}\|, \\ \text{start new increment} & \text{else.} \end{cases}$$

(d) An increment restart is prepared (only after the correction phase fails).

- i. The array of coordinates  $\mathbf{Q}$  is reset to the last item in the list  $\mathcal{Q}^{\text{eq}}$  of previously computed equilibrium states:  $\mathbf{Q} \leftarrow \mathcal{Q}^{\text{eq}}[\text{last}]$ .
- ii. The external force vector is reset the last item in the list  $\tilde{\mathbf{F}}^{\text{ext}}$  of external force vectors corresponding to the previously computed equilibrium states :  $\tilde{\mathbf{F}}^{\text{ext}} \leftarrow \tilde{\mathcal{F}}^{\text{eq}}[\text{last}]$ .
- iii. The current hessian matrix  $\mathbf{K}$  is computed based on the coordinates  $\mathbf{Q}$  (Eq. (9)), the current reduced hessian matrix  $\tilde{\mathbf{K}}$  is formed by removing the rows and columns corresponding to the fixed coordinates.
- iv. The current radius is reduced:  $r \leftarrow r/2$ .
- v. A new increment is started.

### 3 Geometric measures, gradients and Hessians

When computing the gradient and hessian of the elastic energy, necessary for the arc-length method, the measure of the flexel (length, angle, area, etc) and its derivatives (gradient, hessian) with respect to the spatial coordinates  $z_i$  must be computed. In this section, the equations used to calculate those quantities are derived. An overview of the different geometric measure that can be used in `springable` is shown in Fig. S1.

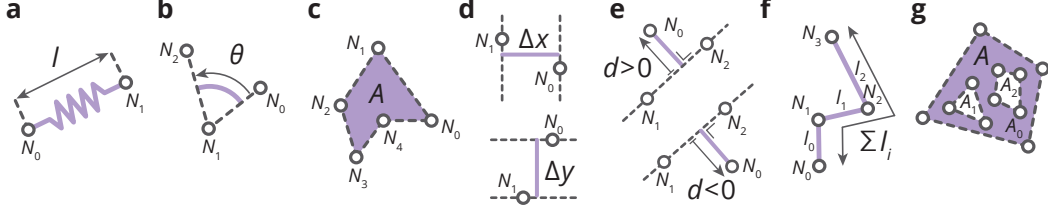


Figure S1: Geometric measures. (a) length, (b) angle, (c) area, (d) signed  $x$ - (top) and signed  $y$ - (bottom) distances, (e) signed point-line distance, which can be positive (top) or negative (bottom), (f) path length, (g) area with holes.

Notation:

- $\alpha$ : the geometric measure,
- $x_k$ : horizontal coordinate of node  $k$ ,
- $y_k$ : vertical coordinate of node  $k$ ,
- $N_k$ : node  $k$ ,
- $\overrightarrow{N_k N_l}$ : vector going from node  $N_k$  to node  $N_l$ ,
- $\mathbf{z}$ : array of spatial nodal coordinates,
- $z_i$ :  $(i + 1)$ th component in the array  $\mathbf{z}$  (so,  $z_0$  is the first component).

#### 3.1 Length (longitudinal flexel)

A longitudinal flexel is defined by two nodes, node  $N_0$  and node  $N_1$  (Fig. S1a). The measure  $\alpha$  of a longitudinal flexel is the length  $l$  of the segment connecting the two nodes. In 2d, this length depends on the four coordinates  $\mathbf{z} = [x_0, y_0, x_1, y_1]$ .

Measure  $\alpha$

$$\alpha = l(\mathbf{z}) = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \quad . \quad (25)$$

Gradient  $\partial\alpha/\partial\mathbf{z}$

$$\frac{\partial\alpha}{\partial\mathbf{z}} = \left[ \frac{\partial\alpha}{\partial z_i} \right] = \left[ \frac{\partial l}{\partial t_0} \quad \frac{\partial l}{\partial y_0} \quad \frac{\partial l}{\partial t_1} \quad \frac{\partial l}{\partial y_1} \right]^\top = \frac{1}{l} \begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \end{bmatrix} \quad . \quad (26)$$

**Hessian**  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$

$$\begin{aligned}
\frac{\partial^2\alpha}{\partial\mathbf{z}\partial\mathbf{z}^\top} &= \left[ \frac{\partial^2\alpha}{\partial z_i \partial z_j} \right] \\
&= \frac{1}{l} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} - \frac{1}{l^2} \frac{1}{l} \begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \end{bmatrix} \begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \end{bmatrix}^\top \\
&= \frac{1}{l} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} - \frac{1}{l^3} \begin{bmatrix} \Delta x \Delta x & \Delta x \Delta y & -\Delta x \Delta x & -\Delta x \Delta y \\ \Delta x \Delta y & \Delta y \Delta y & -\Delta x \Delta y & -\Delta y \Delta y \\ -\Delta x \Delta x & -\Delta x \Delta y & \Delta x \Delta x & \Delta x \Delta y \\ -\Delta x \Delta y & -\Delta y \Delta y & \Delta x \Delta y & \Delta y \Delta y \end{bmatrix}, \tag{27}
\end{aligned}$$

where  $\Delta x := x_0 - x_1$  and  $\Delta y := y_0 - y_1$ .

### 3.2 Angle (angular flexel)

An angular flexel is defined by three nodes: node  $N_0$ , node  $N_1$  and node  $N_2$  (Fig. S1b). The measure  $\alpha$  of an angular flexel is the angle  $\theta$  defined by the three nodes, where node  $N_1$  is the vertex. More precisely, the angle  $\theta \in [0, 2\pi[$  is the angle by which the vector  $\overrightarrow{N_1 N_0}$  must rotate counter-clockwise about  $N_1$  to align with the vector  $\overrightarrow{N_1 N_2}$ . This angle depends on the six coordinates  $\mathbf{z} = [x_0, y_0, x_1, y_1, x_2, y_2]^\top$ .

**Measure**  $\alpha$

$$\alpha = \theta(\mathbf{z}) = \text{atan2}(Y(\mathbf{z}), X(\mathbf{z})) \pmod{2\pi}, \tag{28}$$

where

$$Y(\mathbf{z}) = (x_1 - x_0)(y_1 - y_2) - (y_1 - y_0)(x_1 - x_2), \tag{29}$$

and

$$X(\mathbf{z}) = (x_1 - x_0)(x_1 - x_2) + (y_1 - y_0)(y_1 - y_2). \tag{30}$$

**Gradient**  $\partial\alpha/\partial\mathbf{z}$

$$\frac{\partial\alpha}{\partial\mathbf{z}} = \left[ \frac{\partial\alpha}{\partial z_i} \right] = \left[ \frac{\partial\theta}{\partial z_i} \right] = \left[ \frac{\partial\theta}{\partial Y} \frac{\partial Y}{\partial z_i} + \frac{\partial\theta}{\partial X} \frac{\partial X}{\partial z_i} \right] = \frac{\partial\theta}{\partial Y} \left[ \frac{\partial Y}{\partial z_i} \right] + \frac{\partial\theta}{\partial X} \left[ \frac{\partial X}{\partial z_i} \right], \tag{31}$$

where

$$\left[ \frac{\partial Y}{\partial z_i} \right] = \begin{bmatrix} y_2 - y_1 \\ x_1 - x_2 \\ y_0 - y_2 \\ x_2 - x_0 \\ y_1 - y_0 \\ x_0 - x_1 \end{bmatrix}, \tag{32}$$

$$\left[ \frac{\partial X}{\partial z_i} \right] = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ 2x_1 - x_0 - x_2 \\ 2y_1 - y_0 - y_2 \\ x_0 - x_1 \\ y_0 - y_1 \end{bmatrix}, \tag{33}$$

$$\frac{\partial\theta}{\partial Y} = \frac{X}{X^2 + Y^2}, \tag{34}$$

and

$$\frac{\partial\theta}{\partial X} = -\frac{Y}{X^2 + Y^2}. \tag{35}$$

**Hessian**  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$

$$\begin{aligned}\frac{\partial^2\alpha}{\partial\mathbf{z}\partial\mathbf{z}^\top} &= \left[ \frac{\partial^2\alpha}{\partial z_i\partial z_j} \right] \\ &= \frac{\partial\theta}{\partial Y} \left[ \frac{\partial^2 Y}{\partial z_i\partial z_j} \right] + \frac{\partial\theta}{\partial X} \left[ \frac{\partial^2 X}{\partial z_i\partial z_j} \right] \\ &\quad + \frac{\partial^2\theta}{\partial Y^2} \left[ \frac{\partial Y}{\partial z_i} \frac{\partial Y}{\partial z_j} \right] + \frac{\partial^2\theta}{\partial Y\partial X} \left[ \frac{\partial Y}{\partial z_i} \frac{\partial X}{\partial z_j} + \frac{\partial X}{\partial z_i} \frac{\partial Y}{\partial z_j} \right] + \frac{\partial^2\theta}{\partial X^2} \left[ \frac{\partial X}{\partial z_i} \frac{\partial X}{\partial z_j} \right],\end{aligned}\quad (36)$$

where

$$\left[ \frac{\partial^2 Y}{\partial z_i\partial z_j} \right] = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}, \quad (37)$$

$$\left[ \frac{\partial^2 X}{\partial z_i\partial z_j} \right] = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}, \quad (38)$$

$$\frac{\partial^2\theta}{\partial Y^2} = -\frac{2XY}{(X^2 + Y^2)^2}, \quad (39)$$

$$\frac{\partial^2\theta}{\partial Y\partial X} = \frac{Y^2 - X^2}{(X^2 + Y^2)^2}, \quad (40)$$

and

$$\frac{\partial^2\theta}{\partial X^2} = \frac{2XY}{(X^2 + Y^2)^2}. \quad (41)$$

### 3.3 Signed area

Currently, the measure ‘signed area’ is not directly used as the measure of a flexel. Yet it is used indirectly to define area flexels and ‘signed distance’ flexels (see subsection 3.4 and 3.7). A signed area  $A_s$  is defined by  $n \geq 3$  nodes:  $N_k$ ,  $k = 0, 1, \dots, n-1$ , which are the vertices forming the boundary of a simple polygon. If the sequence  $N_k$  are the vertices listed sequentially counter-clockwise, the ‘signed area’ is the (positive) area of the polygon. Conversely, if the sequence  $N_k$  are the vertices listed sequentially clockwise, the ‘signed area’ is the negative of the (positive) area of the polygon. The polygon is assumed to be not self-intersecting. The signed area depends on  $2n$  coordinates  $\mathbf{z} = [x_0, y_0, \dots, x_{n-1}, y_{n-1}]^\top$

**Measure**  $\alpha$

$$\alpha = A_s(\mathbf{z}) = \frac{1}{2} \sum_{k=0}^{n-1} (x_k y_{k^+} - y_k x_{k^+}) \quad (42)$$

where  $k^+ = (k+1) \bmod n$ .

### Gradient $\partial\alpha/\partial\mathbf{z}$

$$\frac{\partial\alpha}{\partial\mathbf{z}} = \left[ \frac{\partial\alpha}{\partial z_i} \right] = \left[ \frac{\partial A_s}{\partial z_i} \right] = \frac{1}{2} [S_i], \quad (43)$$

where

$$S_i = \begin{cases} y_{k^+} - y_{k^-} & \text{if } i \text{ is even } \Leftrightarrow z_i \text{ is a } x\text{-coordinate} \\ \text{with } k = i/2 & \\ x_{k^-} - x_{k^+} & \text{if } i \text{ is odd } \Leftrightarrow z_i \text{ is a } y\text{-coordinate} \\ \text{with } k = i/2 - 1/2, & \end{cases} \quad (44)$$

with  $k^- = (k - 1) \bmod n$ . Put differently,

$$[S_i] = \begin{bmatrix} y_1 - y_{n-1} \\ x_{n-1} - x_1 \\ y_2 - y_0 \\ x_0 - x_2 \\ \vdots \\ y_0 - y_{n-2} \\ x_{n-2} - x_0 \end{bmatrix}. \quad (45)$$

### Hessian $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$

$$\frac{\partial^2\alpha}{\partial\mathbf{z}\partial\mathbf{z}^\top} = \left[ \frac{\partial^2\alpha}{\partial z_i \partial z_j} \right] = \frac{1}{2} [T_{ij}], \quad (46)$$

where

$$T_{ij} = \begin{cases} +1 & \text{if } i \text{ is even and } j = (i + 3) \bmod 2n \\ -1 & \text{if } i \text{ is even and } j = (i - 1) \bmod 2n \\ -1 & \text{if } i \text{ is odd and } j = (i + 1) \bmod 2n \\ +1 & \text{if } i \text{ is odd and } j = (i - 3) \bmod 2n \\ 0 & \text{else.} \end{cases} \quad (47)$$

Matrices  $[T_{ij}]$  for  $n \in [3, 11]$  are shown in Fig. S2.

### 3.4 Area (area flexel)

An area flexel is defined by  $n \geq 3$  nodes:  $N_0, N_1, \dots, N_{n-1}$  (Fig. S1c). The measure  $\alpha$  of an area flexel is the area  $A \geq 0$  of the polygon whose vertices is the sequence of nodes  $N_k$ . The area  $A$  is always positive regardless of whether the vertices are sequentially listed clockwise or counter-clockwise. The area depends on  $2n$  coordinates  $\mathbf{z} = [x_0, y_0, \dots, x_{n-1}, y_{n-1}]$ .

#### Measure $\alpha$

$$\alpha = A(\mathbf{z}) = |A_s(\mathbf{z})|, \quad (48)$$

where  $A_s$  is the ‘signed area’ measure, defined in Eq. (42).

#### Gradient $\partial\alpha/\partial\mathbf{z}$

$$\frac{\partial\alpha}{\partial\mathbf{z}} = \left[ \frac{\partial\alpha}{\partial z_i} \right] = \text{sgn}(A_s) \left[ \frac{\partial A_s}{\partial z_i} \right] = \frac{1}{2} \text{sgn}(A_s) [S_i], \quad (49)$$

where  $[S_i]$  is the array defined in Eq. (44).

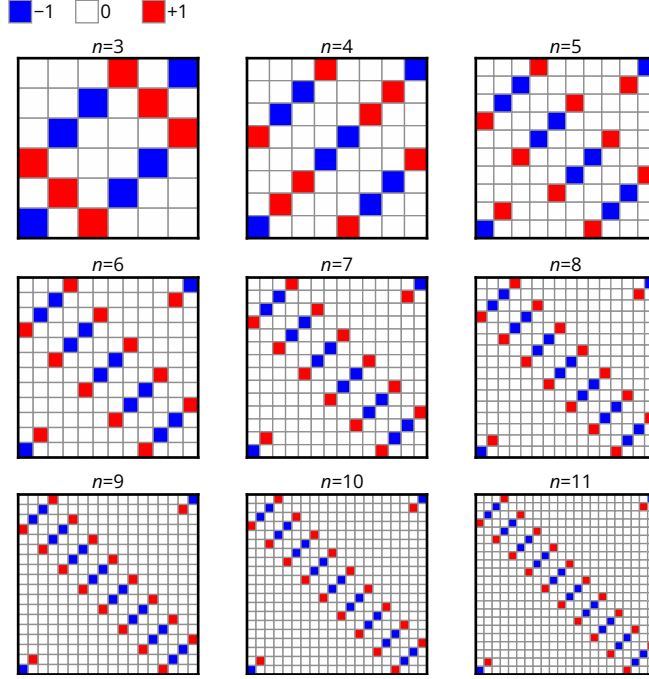


Figure S2:  $[T_{ij}]$  matrices for  $n = 3$  to 11.

**Hessian**  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$

$$\frac{\partial^2\alpha}{\partial\mathbf{z}\partial\mathbf{z}^\top} = \left[ \frac{\partial^2\alpha}{\partial z_i \partial z_j} \right] = \text{sgn}(A_s) \left[ \frac{\partial^2 A_s}{\partial z_i \partial z_j} \right] = \frac{1}{2} \text{sgn}(A_s) [T_{ij}], \quad (50)$$

where  $[T_{ij}]$  is the matrix defined in Eq. (47).

### 3.5 Signed $x$ -distance and $y$ -distance ( $x$ -distance flexel and $y$ -distance flexel)

An  $x$  (resp.  $y$ ) -distance flexel is defined by two nodes  $N_0$  and  $N_1$  (Fig. S1d). The measure of such flexel is the signed horizontal (resp. vertical) distance between both nodes. If  $N_0$  is on the right of (resp. above)  $N_1$ , the signed distance is positive, else negative. The signed distance depends on four coordinates  $\mathbf{z} = [x_0, y_0, x_1, y_1]^\top$ .

**Measure**  $\alpha$

$$\alpha = \Delta x(\mathbf{z}) = x_0 - x_1 \quad (\text{resp.} \quad \alpha = \Delta y(\mathbf{z}) = y_0 - y_1). \quad (51)$$

**Gradient**  $\partial\alpha/\partial\mathbf{z}$

$$\frac{\partial\alpha}{\partial\mathbf{z}} = [1, 0, -1, 0]^\top \quad (\text{resp.} \quad \frac{\partial\alpha}{\partial\mathbf{z}} = [0, 1, 0, -1]^\top). \quad (52)$$

**Hessian**  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$

$$\frac{\partial^2\alpha}{\partial\mathbf{z}\partial\mathbf{z}^\top} = \mathbf{0}^{4 \times 4}. \quad (53)$$

### 3.6 Compound measure

A flexel's measure  $\alpha$  can be expressed as a function  $C$  of  $p$  other measures  $\alpha_k$ ,  $k = 0, 1, \dots, p-1$ . Each measure  $\alpha_k$  itself depends on  $m_k$  coordinates  $\mathbf{z}^k$ . In general, it is possible that measures  $\alpha_k$  share some

of their coordinates; therefore the number  $m$  of coordinates that the measure  $\alpha$  depends on is less than  $\sum_{k=0}^{p-1} m_k$ . The array  $\mathbf{z}$  gathers the set of unique coordinates collected across each  $\mathbf{z}^k$ .

**Measure  $\alpha$**

$$\alpha = C(\alpha_0, \alpha_1, \dots, \alpha_{p-1}). \quad (54)$$

**Gradient  $\partial\alpha/\partial\mathbf{z}$**

$$\begin{aligned} \frac{\partial\alpha}{\partial\mathbf{z}} &= \left[ \frac{\partial\alpha}{\partial z_i} \right] = \sum_{k=0}^{p-1} \frac{\partial C}{\partial\alpha_k} \left[ \frac{\partial\alpha_k(\mathbf{z}^k)}{\partial z_i} \right] = \sum_{k=0}^{p-1} \sum_{r=0}^{m_k-1} \frac{\partial C}{\partial\alpha_k} \frac{\partial\alpha_k}{\partial z_r^k} \left[ \frac{\partial z_r^k}{\partial z_i} \right] \\ &= \sum_{k=0}^{p-1} \sum_{r=0}^{m_k-1} \frac{\partial C}{\partial\alpha_k} \frac{\partial\alpha_k}{\partial z_r^k} \left[ L_{ir}^k \right], \end{aligned} \quad (55)$$

where  $[\partial\alpha_k/\partial z_r^k] = \partial\alpha_k/\partial\mathbf{z}^k$  is the gradient of the measure  $\alpha_k$  in its ‘own coordinates’  $\mathbf{z}^k$ , and

$$L_{ir}^k = \begin{cases} 1 & \text{if } z_i = z_r^k \\ 0 & \text{else.} \end{cases} \quad (56)$$

In practical terms, Eq. (55) means that the gradient  $\partial\alpha/\partial\mathbf{z} = [\partial\alpha/\partial z_i]$  of a compound measure is constructed by summing together the components of each individual measure’s gradient  $\partial\alpha_k/\partial\mathbf{z}^k$  that correspond to the same coordinate  $z_i$ , weighted by  $\partial C/\partial\alpha_k$ .

**Hessian  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$**

$$\begin{aligned} \frac{\partial^2\alpha}{\partial\mathbf{z}\partial\mathbf{z}^\top} &= \left[ \frac{\partial^2\alpha}{\partial z_i \partial z_j} \right] = \sum_{k=0}^{p-1} \sum_{l=0}^{p-1} \frac{\partial^2 C}{\partial\alpha_k \partial\alpha_l} \left[ \frac{\partial\alpha_k(\mathbf{z}^k)}{\partial z_i} \frac{\partial\alpha_l(\mathbf{z}^l)}{\partial z_j} \right] + \sum_{k=0}^{p-1} \frac{\partial C}{\partial\alpha_k} \frac{\partial^2\alpha_k(\mathbf{z}^k)}{\partial z_i \partial z_j} \\ &= \sum_{k=0}^{p-1} \sum_{l=0}^{p-1} \sum_{r=0}^{m_k-1} \sum_{s=0}^{m_l-1} \frac{\partial^2 C}{\partial\alpha_k \partial\alpha_l} \frac{\partial\alpha_k}{\partial z_r^k} \frac{\partial\alpha_l}{\partial z_s^l} \left[ L_{ir}^k L_{js}^l \right] + \sum_{k=0}^{p-1} \sum_{r=0}^{m_k-1} \sum_{s=0}^{m_k-1} \frac{\partial C}{\partial\alpha_k} \frac{\partial^2\alpha_k}{\partial z_r^k \partial z_s^k} \left[ L_{ir}^k L_{js}^k \right], \end{aligned} \quad (57)$$

where  $[\partial\alpha_k/\partial z_r^k] = \partial\alpha_k/\partial\mathbf{z}^k$  and  $[\partial\alpha_l/\partial z_s^l] = \partial\alpha_l/\partial\mathbf{z}^l$  are the gradients of the individual measures in their own coordinates;  $[\partial^2\alpha_k/\partial z_r^k \partial z_s^k] = \partial^2\alpha_k/\partial\mathbf{z}^k \partial\mathbf{z}^{k\top}$  are the hessian matrices of the individual measures in their own coordinates; and where  $[L_{ir}^k L_{js}^l]$  and  $[L_{ir}^k L_{js}^k]$  are computed using Eq. (56). In practical terms, Eq. (57) means that the hessian matrix  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$  is constructed by summing together the components

- of the outer-product matrices  $\partial\alpha_k/\partial\mathbf{z}^k \otimes \partial\alpha_l/\partial\mathbf{z}^l$  for every possible pair of individual gradients (weighted by  $\partial^2 C/\partial\alpha_k \partial\alpha_l$ ); and
- of the individual hessian matrices  $\partial^2\alpha_k/\partial\mathbf{z}^k \partial\mathbf{z}^{k\top}$  (weighted by  $\partial C/\partial\alpha_k$ )

that correspond to the same pair of coordinates  $(z_i, z_j)$ .

### 3.7 Signed point-line distance (distance flexel)

A distance flexel is defined by 3 nodes:  $N_0$ ,  $N_1$  and  $N_2$  (Fig. S1e). The measure  $\alpha$  of a distance flexel is the ‘signed distance’  $d$  between  $N_0$  and the infinite line passing through  $N_1$  and  $N_2$ . If  $N_0$  is on the left of that line (whose direction is defined by the vector  $\overrightarrow{N_1 N_2}$ ), the measure is the (positive) distance. If  $N_0$  is on the right of the line, the measure is the negative of the (positive) distance. The ‘signed distance’ depends on the six coordinates  $\mathbf{z} = [x_0, y_0, x_1, y_1, x_2, y_2]^\top$ .

### Measure $\alpha$

The ‘signed distance’ measure  $\alpha$  can be regarded as a compound measure: it is twice the ‘signed area’ of the triangle  $N_0N_1N_2$  ( $\alpha_0$ ) divided by the length of the segment  $N_1N_2$  ( $\alpha_1$ ):

$$\alpha = C(\alpha_0, \alpha_1) = \frac{2\alpha_0}{\alpha_1} = \frac{2A_s(\mathbf{z}^0)}{l(\mathbf{z}^1)} = d(\mathbf{z}), \quad (58)$$

where  $A_s(\mathbf{z}^0)$  is the ‘signed area’ of the triangle  $N_0N_1N_2$  (Eq. (42)), with  $\mathbf{z}^0 = [x_0, y_0, x_1, y_1, x_2, y_2]^\top$ ; and  $l(\mathbf{z}^1)$  is the length of the segment  $N_1N_2$  (Eq. (25)), with  $\mathbf{z}^1 = [x_1, y_1, x_2, y_2]^\top$ .

**Gradient  $\partial\alpha/\partial\mathbf{z}$**  As the signed distance measure is a compound measure, its gradient  $\partial\alpha/\partial\mathbf{z}$  is computed using Eq. (55), with  $\partial C/\partial\alpha_0 = 2/\alpha_1$  and  $\partial C/\partial\alpha_1 = -2\alpha_0/\alpha_1^2$ , and where the individual gradients  $\partial\alpha_0/\partial\mathbf{z}^0$  and  $\partial\alpha_1/\partial\mathbf{z}^1$  are computed using Eq. (43) and Eq. (26), respectively.

**Hessian  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$**  As the signed distance measure is a compound measure, its hessian matrix  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$  is computed using Eq. (57), with  $\partial C/\partial\alpha_0 = 2/\alpha_1$ ,  $\partial C/\partial\alpha_1 = -2\alpha_0/\alpha_1^2$ ,  $\partial^2 C/\partial\alpha_0^2 = 0$ ,  $\partial^2 C/\partial\alpha_0\partial\alpha_1 = \partial^2 C/\partial\alpha_1\partial\alpha_0 = -2/\alpha_1^2$  and  $\partial^2 C/\partial\alpha_1^2 = 4\alpha_0/\alpha_1^3$ , where the individual gradients and hessian matrices  $\partial\alpha_0/\partial\mathbf{z}^0$ ,  $\partial\alpha_1/\partial\mathbf{z}^1$ ,  $\partial^2\alpha_0/\partial\mathbf{z}^0\partial\mathbf{z}^0{}^\top$ ,  $\partial^2\alpha_1/\partial\mathbf{z}^1\partial\mathbf{z}^1{}^\top$  are computed using Eq. (43), Eq. (26), Eq. (46) and Eq. (27), respectively.

### 3.8 Path length (path flexel)

A path flexel is defined by an ordered sequence of  $p+1$  nodes  $\{N^k\}$  selected from a set of  $n \geq 2$  distinct nodes  $N_0, N_1, \dots, N_{n-1}$  (Fig. S1f). The measure of  $\alpha$  of a path flexel is the length of the polygonal chain whose vertices are defined by the sequence of nodes  $\{N^k\}$ . The path length depends on  $2n$  coordinates  $\mathbf{z} = [x_0, y_0, \dots, x_{n-1}, y_{n-1}]^\top$ .

#### Measure $\alpha$

The path length measure  $\alpha$  can be regarded as a compound measure: it is the sum of the lengths of the  $p$  segments  $N^kN^{k+1}$ :

$$\alpha = C(\alpha_0, \alpha_1, \dots, \alpha_{p-1}) = \alpha_0 + \alpha_1 + \dots + \alpha_{p-1} = \sum_{k=0}^{p-1} l_k(\mathbf{z}^k), \quad (59)$$

where  $\alpha_k = l_k$  is the length of the segment  $N^kN^{k+1}$  (Eq. (25)) and  $\mathbf{z}^k = [x_k, y_k, x_{k+1}, y_{k+1}]^\top$ .

**Gradient  $\partial\alpha/\partial\mathbf{z}$**  As the path length measure is a compound measure, its gradient  $\partial\alpha/\partial\mathbf{z}$  is computed using Eq. (55), with  $\partial C/\partial\alpha_k = 1$ , and where the individual gradients  $\partial\alpha_k/\partial\mathbf{z}^k$  are computed using Eq. (26). In case that the sequence is composed of distinct nodes ( $N^k = N_k$ ), the gradient  $\partial\alpha/\partial\mathbf{z}$  takes the following form:

$$\frac{\partial\alpha}{\partial\mathbf{z}} = \frac{1}{l_0} \begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \frac{1}{l_1} \begin{bmatrix} 0 \\ 0 \\ x_1 - x_2 \\ y_1 - y_2 \\ x_2 - x_1 \\ y_2 - y_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \dots + \frac{1}{l_{n-2}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_{n-2} - x_{n-1} \\ y_{n-2} - y_{n-1} \\ x_{n-1} - x_{n-2} \\ y_{n-1} - y_{n-2} \end{bmatrix}. \quad (60)$$

**Hessian**  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$  As the path length measure is a compound measure, its hessian matrix  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$  is computed using Eq. (57), with  $\partial C/\partial\alpha_k = 1$  and  $\partial^2 C/\partial\alpha_k\partial\alpha_l = 0$ , where the individual gradients and hessian matrices  $\partial\alpha_k/\partial\mathbf{z}^k$  and  $\partial^2\alpha_k/\partial\mathbf{z}^k\partial\mathbf{z}^{k\top}$  are computed using Eq. (26) and Eq. (27), respectively.

### 3.9 Area with holes (area flexel)

In subsection 3.4, the geometric measure of an area flexel was defined as the area of a simple polygon. An area flexel can also be defined by a polygon with holes, that is, a polygon with an outer boundary and one or multiple inner boundaries (Fig. S1g). The geometric measure  $\alpha$  of such polygon is the area delimited by the outer boundary  $A_0$  to which is subtracted the areas delimited by the inner boundaries  $A_1, A_2, \dots, A_{p-1}$ . Each area  $A_k$  is defined by a set of coordinates, noted  $\mathbf{z}^k$ . The array of unique coordinates  $\mathbf{z}$  is constructed by sequentially appending each coordinate from  $\mathbf{z}^0, \dots, \mathbf{z}^{p-1}$  that has not yet been encountered.

**Measure**  $\alpha$  The area-with-holes measure  $\alpha$  can be regarded as a compound measure: it is outer area minus the sum of the inner areas:

$$\alpha = C(\alpha_0, \alpha_1, \dots, \alpha_{p-1}) = A_0 - \sum_{k=1}^{p-1} A_k(\mathbf{z}^k) = A(\mathbf{z}). \quad (61)$$

where  $\alpha_k = A_k(\mathbf{z}^k)$  is computed using Eq. (48).

**Gradient**  $\partial\alpha/\partial\mathbf{z}$  As the area-with-holes measure is a compound measure, its gradient  $\partial\alpha/\partial\mathbf{z}$  is computed using Eq. (55), with  $\partial C/\partial\alpha_0 = 1$  and  $\partial C/\partial\alpha_k = -1$  ( $\forall k \neq 0$ ), and where the individual gradients  $\partial\alpha_k/\partial\mathbf{z}^k$  are computed using Eq. (49).

**Hessian**  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$  As the area-with-holes measure is a compound measure, its hessian matrix  $\partial^2\alpha/\partial\mathbf{z}\partial\mathbf{z}^\top$  is computed using Eq. (57), with  $\partial C/\partial\alpha_0 = 1$ ,  $\partial C/\partial\alpha_k = -1$  ( $\forall k \neq 0$ ) and  $\partial^2 C/\partial\alpha_k\partial\alpha_l = 0$ , where the individual gradients and hessian matrices  $\partial\alpha_k/\partial\mathbf{z}^k$  and  $\partial^2\alpha_k/\partial\mathbf{z}^k\partial\mathbf{z}^{k\top}$  are computed using Eq. (49) and Eq. (50), respectively.

## 4 Intrinsic nonlinear mechanical behaviors

When computing the gradient and hessian of the elastic energy, necessary for the arc-length method, the gradient and hessian of the intrinsic elastic potential  $v$  of the flexel with respect to the measure  $\alpha$  (and the internal degree of freedom  $x$  for bivariate potentials) must be computed. In this section, the equations used to calculate those quantities are derived for various kinds of potentials. In **springable**, potentials are specified by providing the generalized force-displacement curve, where the generalized force  $f$  is defined as  $\partial v/\partial\alpha$ , and the generalized displacement  $u$  as  $\alpha - \alpha_0$  ( $\alpha_0$  is a constant representing the natural measure). An overview of the different types of nonlinear generalized force-displacement curves implemented in **springable** is shown in Fig. S3. In the following, we use apostrophes,  $g'$  and  $g''$ , to denote the first and second derivatives of functions  $g$  of one single argument.

### 4.1 Univariate behaviors

A univariate behavior is characterized by a univariate energy potential, which is constructed from a generalized force-displacement curve that can be described as function:  $f(u)$ .

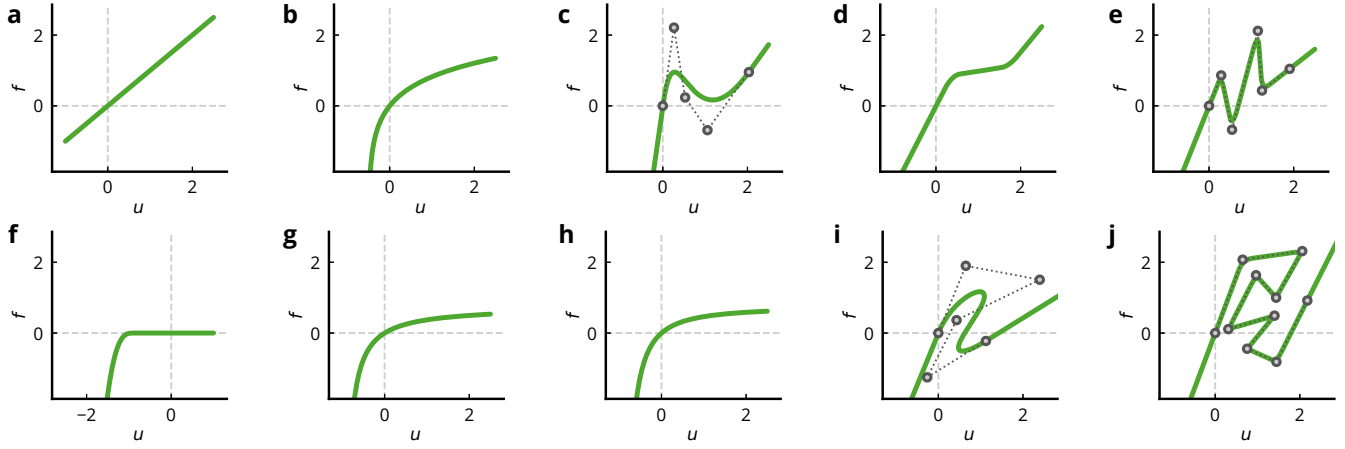


Figure S3: Nonlinear generalized force ( $f$ )-displacement ( $u$ ) curves used to construct nonlinear behaviors. (a) linear, (b) logarithmic, (c) Bezier (univariate) (d)  $\mathcal{C}_1$ -piecewise linear, (e) zigzag (univariate), (f) contact, (g) isothermal, (h) isentropic, (i) Bezier (bivariate), (j) zigzag (bivariate). Control points and control polygons are depicted by gray markers and dotted lines.

**Linear behavior** A linear behavior is defined by a linear generalized force-displacement curve (Fig. S4a),

$$f(u) = ku, \quad (62)$$

where  $k$  is the spring constant, that is, the slope of the generalized force-displacement curve, whose unit is the unit of the generalized force  $f$  divided by the unit of the generalized displacement  $u$ . From the curve, the potential  $v$ , its gradient  $\partial v/\partial\alpha$  and hessian  $\partial^2 v/\partial\alpha^2$  can be constructed as follows:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)du = \frac{1}{2}k(\alpha - \alpha_0)^2, \quad (63)$$

$$\frac{\partial v}{\partial\alpha}(\alpha) = f(\alpha - \alpha_0) = k(\alpha - \alpha_0), \quad (64)$$

$$\frac{\partial^2 v}{\partial\alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = k. \quad (65)$$

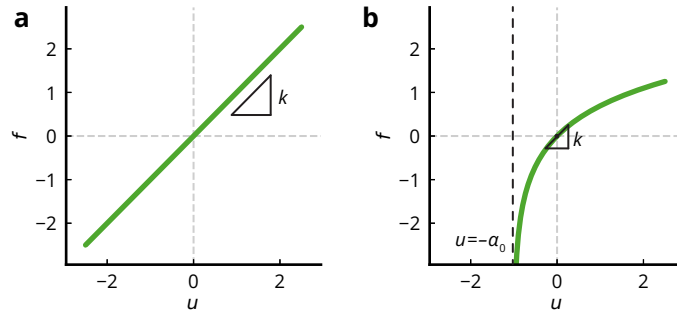


Figure S4: Linear (a) and logarithmic (b) force-displacement curves.

**Logarithmic behavior** A logarithmic behavior is defined by a generalized force-displacement curve given by

$$f(u) = k\alpha_0 \ln\left(\frac{u + \alpha_0}{\alpha_0}\right), \quad (66)$$

where  $k$  is a parameter that represents the slope of the generalized force-displacement curve at  $u = 0$ , whose unit is the unit of the generalized force  $f$  divided by the unit of the generalized displacement  $u$ . Unlike the linear behavior, a logarithmic behavior can prevent flexels from reaching a zero measure ( $\alpha = 0$ ), as the force tends to infinity when the flexel becomes ‘fully compressed’ (Fig. S4b). The behavior is identical to the linear behavior at first order around the natural measure  $\alpha_0$ . From the curve, the potential  $v$ , its gradient  $\partial v/\partial\alpha$  and hessian  $\partial^2 v/\partial\alpha^2$  can be constructed as follows:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)du = k\alpha\alpha_0 (\ln(\alpha/\alpha_0) - 1), \quad (67)$$

$$\frac{\partial v}{\partial\alpha}(\alpha) = f(\alpha - \alpha_0) = k\alpha_0 \ln(\alpha/\alpha_0), \quad (68)$$

$$\frac{\partial^2 v}{\partial\alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = k\alpha_0/\alpha. \quad (69)$$

**Bezier behavior** A Bezier behavior is described by a Bezier curve of degree  $n$ , with control points positioned at  $(u_0 = 0, f_0 = 0)$ ,  $(u_1, f_1), \dots, (u_n, f_n)$ , with linear extrapolation beyond the first and last control points that preserves  $\mathcal{C}_1$  continuity. The nonlinear behavior defined by the Bezier curve can either describe the tensile behavior (a.k.a ‘tensile’ mode, Fig. S5a), the compressive behavior (a.k.a ‘compressive’ mode, Fig. S5b) or both simultaneously (a.k.a ‘symmetric’ mode, Fig. S5c).

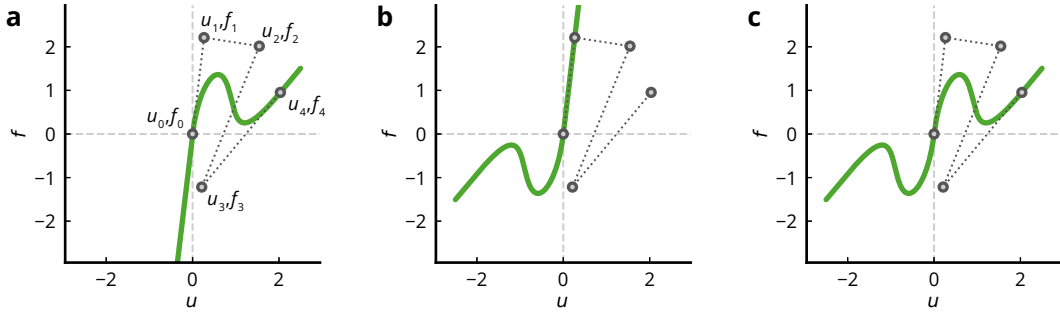


Figure S5: Bezier force-displacement curves for the tensile (a), compressive (b) and symmetric (c) modes.

For convenience, let us define the functional  $\mathcal{F}$  as

$$\mathcal{F}(g)(x) = m(x)g(m(x)x), \quad (70)$$

where  $g$  is a function and

$$m(x) = \begin{cases} +1 & \text{if ‘tensile’ mode,} \\ -1 & \text{if ‘compressive’ mode,} \\ \text{sgn}(x) & \text{if ‘symmetric’ mode.} \end{cases} \quad (71)$$

Note that

$$\frac{\partial}{\partial x}\mathcal{F}(g)(x) = g'(m(x)x) := \mathcal{F}'(g)(x), \quad (72)$$

$$\frac{\partial^2}{\partial x^2}\mathcal{F}(g)(x) = m(x)g''(m(x)x) := \mathcal{F}''(g)(x). \quad (73)$$

Mathematically speaking, the force displacement curve  $f(u)$  is defined as

$$f(u) = \mathcal{F}(\bar{f})(u), \quad (74)$$

where

$$\bar{f}(u) = \begin{cases} (f_1/u_1)u & \text{if } u \leq 0, \\ b(a^{-1}(u)) & \text{if } 0 < u \leq u_n \\ f_n + \frac{f_n - f_{n-1}}{u_n - u_{n-1}}(u - u_n) & \text{if } u > u_n, \end{cases} \quad (75)$$

where the functions  $a(\cdot), b(\cdot)$  describe the parametric Bezier curve:

$$\begin{cases} a(x) = \sum_{i=0}^n u_i B_{i,n}(x) \\ b(x) = \sum_{i=0}^n f_i B_{i,n}(x), \end{cases} \quad (76)$$

where  $x$  is the curve parameter that runs from 0 to 1 and  $B_{i,n}$  are the Bernstein polynomials of degree  $n$ . Note that the generalized force-displacement curve is only well defined if  $a(x)$  is monotonic on the interval  $[0, 1]$ . The stiffness  $k(u)$  along the curve is given by the derivative of  $f(u)$ , i.e.:

$$k(u) = \frac{df}{du} = \mathcal{F}'(\bar{f})(u), \quad (77)$$

where  $\mathcal{F}'$  is defined in Eq. (72) and

$$\frac{d\bar{f}}{du} = \begin{cases} f_1/u_1 & \text{if } u \leq 0, \\ \frac{b'(a^{-1}(u))}{a'(a^{-1}(u))} & \text{if } 0 < u \leq u_n \\ \frac{f_n - f_{n-1}}{u_n - u_{n-1}} & \text{if } u > u_n, \end{cases} \quad (78)$$

where  $a', b'$  are the derivatives of  $a, b$ :

$$a'(x) = n \sum_{i=0}^{n-1} (u_{i+1} - u_i) B_{i,n-1}(x) \quad \text{and} \quad b'(x) = n \sum_{i=0}^{n-1} (f_{i+1} - f_i) B_{i,n-1}(x). \quad (79)$$

From the curve, the potential  $v$ , its gradient  $\partial v / \partial \alpha$  and hessian  $\partial^2 v / \partial \alpha^2$  can be constructed as follows:

$$v(\alpha) = \int_0^{\alpha - \alpha_0} f(u) du, \quad (80)$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0), \quad (81)$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = k(\alpha - \alpha_0), \quad (82)$$

where  $f$  and  $k$  are described in Eqs. (74, 77).

**$\mathcal{C}_1$ -piecewise behavior** A  $\mathcal{C}_1$ -piecewise behavior is defined by a piecewise linear function whose ‘corners’ have been smoothed by quadratic functions, ensuring  $\mathcal{C}_1$  continuity. Bilinear and trilinear piecewise linear force-displacement curves have been used to study softening, stiffening or bistable interactions in spring chains [4, 5, 6]. The piecewise curve presented here generalizes the concept to  $n$  segments with control on the smoothness of the transitions. The curve is described by  $n$  slopes  $k_0, \dots, k_{n-1}$ ,  $n - 1$  displacement values  $u_0, \dots, u_{n-2}$  at which the non-smoothed piecewise curve transitions from slope  $k_i$  to  $k_{i+1}$ , and a

smoothing parameter  $u_s$  that controls the size of the region around the transition displacement  $u_i$  where the curve is given by the quadratic function instead of the piecewise linear function. Similarly to the Bezier behavior, the nonlinear behavior defined by the smoothed piecewise linear curve can either describe the tensile behavior (a.k.a ‘tensile’ mode, Fig. S6a), the compressive behavior (a.k.a ‘compressive’ mode, Fig. S6b) or both simultaneously (a.k.a ‘symmetric’ mode, Fig. S6c).

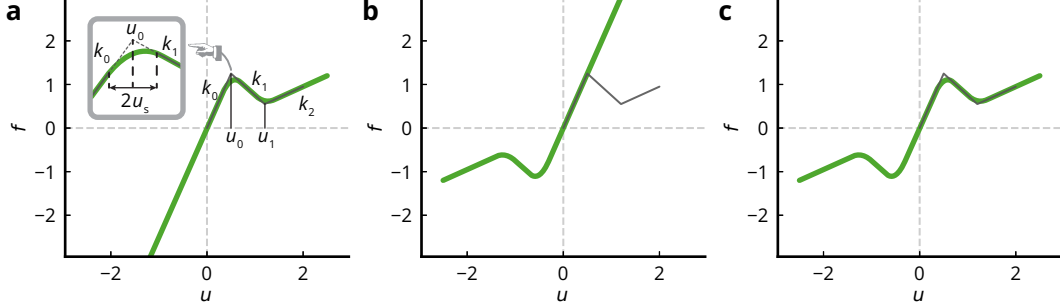


Figure S6:  $\mathcal{C}_1$ -piecewise linear force-displacement curves for the tensile (a), compressive (b) and symmetric (c) modes.

Mathematically speaking,

$$f(u) = \mathcal{F}(\bar{f})(u), \quad (83)$$

where  $\mathcal{F}$  is defined in Eq. (70) and

$$\bar{f}(u) = P(u; \mathbf{s} = [k_0, k_1, \dots, k_{n-1}]; \boldsymbol{\eta} = [u_0, u_1, \dots, u_{n-2}]; \eta_s = u_s), \quad (84)$$

where

$$P(\eta; \mathbf{s}; \boldsymbol{\eta}; \eta_s) = \begin{cases} s_0\eta & \text{if } \eta \leq \eta_0 - \eta_s \\ a_0\eta^2 + b_0\eta + c_0 & \text{if } \eta_0 - \eta_s < \eta < \eta_0 + \eta_s \\ s_1\eta + p_1 & \text{if } \eta_0 + \eta_s \leq \eta \leq \eta_1 - \eta_s \\ \vdots & \vdots \\ a_i\eta^2 + b_i\eta + c_i & \text{if } \eta_i - \eta_s < \eta < \eta_i + \eta_s \\ s_i\eta + p_i & \text{if } \eta_{i-1} + \eta_s \leq \eta \leq \eta_i - \eta_s \\ \vdots & \vdots \\ a_{n-2}\eta^2 + b_{n-2}\eta + c_{n-2} & \text{if } \eta_{n-2} - \eta_s < \eta < \eta_{n-2} + \eta_s \\ s_{n-1}\eta + p_{n-1} & \text{if } \eta_{n-2} + \eta_s \leq \eta < +\infty, \end{cases} \quad (85)$$

with

$$p_i = p_{i-1} + \eta_{i-1} (s_{i-1} - s_i); \quad p_0 = 0, \quad (86)$$

$$a_i = \frac{s_{i+1} - s_i}{4\eta_s}, \quad (87)$$

$$b_i = \frac{s_i(\eta_i + \eta_s) - s_{i+1}(\eta_i - \eta_s)}{2\eta_s}, \quad (88)$$

$$c_i = \frac{(s_{i+1} - s_i)(\eta_i - \eta_s)^2}{4\eta_s} + p_i. \quad (89)$$

Note that for  $P$  is well-defined only if  $2\eta_s < \min\{2\eta_0, \eta_1 - \eta_0, \dots, \eta_{n-2} - \eta_{n-3}\}$ . The stiffness  $k(u)$  along the curve is given by the derivative of  $f(u)$ , i.e.:

$$k(u) = \frac{df}{du} = \mathcal{F}'(\bar{f})(u), \quad (90)$$

where  $\mathcal{F}'$  is defined in Eq. (72) and

$$\frac{d\bar{f}}{du} = \frac{\partial P}{\partial \eta}(u; [k_0, \dots, k_{n-1}]; [u_0, \dots, u_{n-2}]; u_s), \quad (91)$$

where

$$\frac{\partial P}{\partial \eta}(\eta, \mathbf{s}, \boldsymbol{\eta}, \eta_s) = \begin{cases} s_0 & \text{if } \eta \leq \eta_0 - \eta_s \\ 2a_0\eta + b_0 & \text{if } \eta_0 - \eta_s < \eta < \eta_0 + \eta_s \\ s_1 & \text{if } \eta_0 + \eta_s \leq \eta \leq \eta_1 - \eta_s \\ \vdots & \vdots \\ 2a_i\eta + b_i & \text{if } \eta_i - \eta_s < \eta < \eta_i + \eta_s \\ s_i & \text{if } \eta_{i-1} + \eta_s \leq \eta \leq \eta_i - \eta_s \\ \vdots & \vdots \\ 2a_{n-2}\eta + b_{n-2} & \text{if } \eta_{n-2} - \eta_s < \eta < \eta_{n-2} + \eta_s \\ s_{n-1} & \text{if } \eta_{n-2} + \eta_s \leq \eta < +\infty. \end{cases} \quad (92)$$

From the curve, the potential  $v$ , its gradient  $\partial v/\partial \alpha$  and hessian  $\partial^2 v/\partial \alpha^2$  can be constructed as follows:

$$v(\alpha) = \int_0^{\alpha - \alpha_0} f(u) du, \quad (93)$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0), \quad (94)$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = k(\alpha - \alpha_0), \quad (95)$$

where  $f$  and  $k$  are described in Eqs. (83, 90).

**Zigzag behavior** A zigzag behavior is described by a ‘smooth zigzag’, that is, a polygonal chain with vertices positioned at  $(u_0 = 0, f_0 = 0), (u_1, f_1), \dots, (u_n, f_n)$ , whose ‘corners’ have been smoothed. The amount of smoothing is governed by the parameter  $\epsilon \in ]0, 1[$ . The zigzag behavior serves a similar purpose as the  $\mathcal{C}_1$ -piecewise linear behavior, except that it can be generalized to a multi-valued curve as shown in subsection 4.2. Similarly to the Bezier behavior, the smooth zigzag can either describe the tensile behavior (Fig. S7a), the compressive behavior (Fig. S7b) or both simultaneously (Fig. S7c).

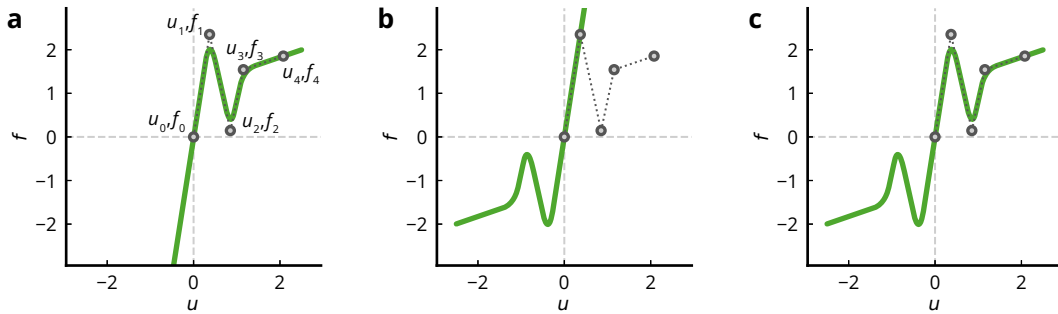


Figure S7: Zigzag force-displacement curves for the tensile (a), compressive (b) and symmetric (c) modes.

Mathematically speaking,

$$f(u) = \mathcal{F}(\bar{f})(u) \quad (96)$$

where  $\mathcal{F}$  is defined in Eq. (70) and

$$\bar{f}(u) = b(a^{-1}(u)), \quad (97)$$

where the functions  $a(\cdot)$  and  $b(\cdot)$  are defined by  $\mathcal{C}_1$ -piecewise linear functions:

$$a(x) = P(x; \mathbf{s}^a; \boldsymbol{\eta}^a; \eta_s^a) \quad (98)$$

$$b(x) = P(x; \mathbf{s}^b; \boldsymbol{\eta}^b; \eta_s^b), \quad (99)$$

with  $P$  defined in Eq. (85),  $s_i^a = (n-1)(u_{i+1} - u_i)$ ,  $s_i^b = (n-1)(f_{i+1} - f_i)$ ,  $\eta_i^a = \eta_i^b = i/(n-1)$ , and  $\eta_s^a = \eta_s^b = \epsilon/(2n-2)$ . Note that the generalized force-displacement curve is only well defined if  $a(x)$  is monotonic on the interval  $[0, +\infty[$ , which is valid only if  $0 < u_1 < \dots < u_{n-2}$ . The stiffness  $k(u)$  along the curve is given by the derivative of  $f(u)$ , i.e.:

$$k(u) = \frac{df}{du}(u) = \mathcal{F}'(\bar{f})(u), \quad (100)$$

where  $\mathcal{F}'$  is defined in Eq. (72) and

$$\frac{d\bar{f}}{du}(u) = b'(a^{-1}(u))/a'(a^{-1}(u)), \quad (101)$$

where  $a' = \partial P/\partial \eta(x, \mathbf{s}^a, \boldsymbol{\eta}^a, \eta_s^a)$  and  $b' = \partial P/\partial \eta(x, \mathbf{s}^b, \boldsymbol{\eta}^b, \eta_s^b)$  are the derivatives of  $a$  and  $b$  with respect to  $x$ , with  $\partial P/\partial \eta$  given in Eq. (92). From the curve, the potential  $v$ , its gradient  $\partial v/\partial \alpha$  and hessian  $\partial^2 v/\partial \alpha^2$  can be constructed as follows:

$$v(\alpha) = \int_0^{\alpha - \alpha_0} f(u) du, \quad (102)$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0), \quad (103)$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = k(\alpha - \alpha_0), \quad (104)$$

where  $f$  and  $k$  are described in Eqs. (96, 100).

**Contact behavior** A contact behavior mimics contact using a generalized force-displacement curve that produces a nonzero force only for measures  $\alpha = u + \alpha_0$  below a certain threshold  $\alpha_\Delta$  (Fig. S8):

$$f(u) = \begin{cases} 0 & \text{if } u \geq \alpha_\Delta - \alpha_0 \\ -f_0 \left( \frac{\alpha_\Delta - \alpha_0 - u}{u_c} \right)^3 & \text{if } u < \alpha_\Delta - \alpha_0, \end{cases} \quad (105)$$

where  $f_0 > 0$  is the magnitude of the repulsive force when the measure is decreased by  $u_c > 0$  from  $\alpha_\Delta$ . The curve is  $\mathcal{C}_2$  continuous. From the curve the potential  $v$ , its gradient  $\partial v/\partial \alpha$  and hessian  $\partial^2 v/\partial \alpha^2$  can

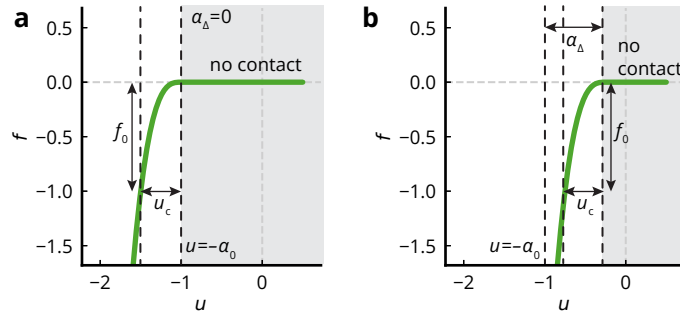


Figure S8: ‘Contact’ force-displacement curves with  $\alpha_0 = 1$  when  $\alpha_\Delta = 0$  (a) and  $\alpha_\Delta = 0.75$ .

be constructed as follows:

$$v(\alpha) = \int_{\alpha_\Delta}^{\alpha} f(u) du = \begin{cases} 0 & \text{if } \alpha \geq \alpha_\Delta \\ \frac{f_0 u_c}{4} \left( \frac{\alpha_\Delta - \alpha}{u_c} \right)^4 & \text{if } \alpha < \alpha_\Delta, \end{cases} \quad (106)$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0) = \begin{cases} 0 & \text{if } \alpha \geq \alpha_\Delta \\ -f_0 \left( \frac{\alpha_\Delta - \alpha}{u_c} \right)^3 & \text{if } \alpha < \alpha_\Delta, \end{cases} \quad (107)$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = \begin{cases} 0 & \text{if } \alpha \geq \alpha_\Delta \\ \frac{3f_0}{u_c} \left( \frac{\alpha_\Delta - \alpha}{u_c} \right)^2 & \text{if } \alpha < \alpha_\Delta. \end{cases} \quad (108)$$

**Isothermal behavior** An isothermal behavior is described by a generalized force-displacement curve that mimics the ideal gas law during an isothermal process (process where the temperature remains constant):

$$p - p_0 = nRT_0(1/V - 1/V_0), \quad (109)$$

where  $p_0$  is the ambient pressure,  $V_0$  is the volume of the gas at ambient pressure,  $T_0$  is the temperature of the gas,  $p$  the pressure,  $V$  the volume,  $n$  is the amount of substance,  $R$  the gas constant. The generalized-force displacement curve is defined from Eq. (110) by letting  $f$  play the role of the pressure difference  $p_0 - p$  and  $u$  the role of volume change  $V - V_0$ , with  $\alpha_0 = V_0$  (Fig. S9a):

$$f(u) = \frac{nRT_0}{\alpha_0} \frac{u}{u + \alpha_0}. \quad (110)$$

Note that a negative generalized force  $f$  corresponds to a compressed state (the pressure  $p$  is greater than the ambient pressure  $p_0$ ), while a positive  $f$  corresponds to a ‘vacuumed’ state (the pressure  $p$  is less than the ambient pressure  $p_0$ ). From the curve, the potential  $v$ , its gradient  $\partial v/\partial \alpha$  and hessian  $\partial^2 v/\partial \alpha^2$  can be constructed as follows:

$$v(\alpha) = \int_0^{\alpha - \alpha_0} f(u) du = nRT_0(\alpha/\alpha_0 - 1 - \ln(\alpha/\alpha_0)), \quad (111)$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0) = nRT_0 \frac{\alpha - \alpha_0}{\alpha \alpha_0}, \quad (112)$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = \frac{nRT_0}{\alpha^2}. \quad (113)$$

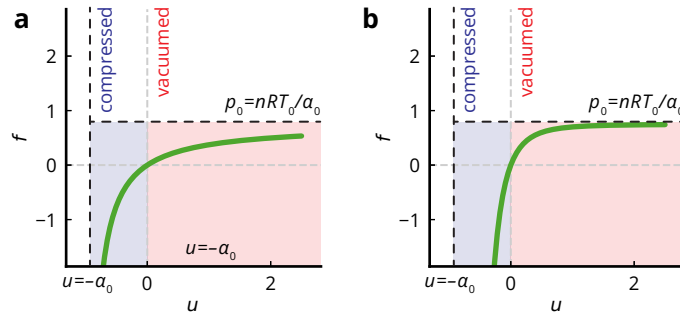


Figure S9: Force-displacement curves mimicking the ideal gas law. (a) Isothermal with  $n = 0.03$ ,  $R = 0.083$ ,  $T_0 = 300$ . (b) Isentropic with  $n = 0.03$ ,  $R = 0.083$ ,  $T_0 = 300$ ,  $\gamma = 4$ .

**Isentropic behavior** An isentropic behavior is described by a generalized force-displacement curve that mimics the ideal gas law during an isentropic process (process where the entropy remains constant):

$$p - p_0 = nRT_0 \left( \frac{1}{V} \left( \frac{V_0}{V} \right)^{\gamma-1} - \frac{1}{V_0} \right), \quad (114)$$

where  $p_0$  is the ambient pressure,  $V_0$  the volume of the gas at ambient pressure,  $T_0$  is the initial temperature,  $p$  the pressure,  $V$  the volume,  $n$  the amount of substance,  $R$  the gas constant and  $\gamma$  the heat capacity ratio. The generalized-force displacement curve is defined from Eq. by letting  $f$  play the role of the pressure difference  $p_0 - p$  and  $u$  the role of volume change  $V - V_0$ , with  $\alpha_0 = V_0$  (Fig. S9b):

$$f(u) = nRT_0 \left( \frac{1}{\alpha_0} - \frac{1}{u + \alpha_0} \left( \frac{\alpha_0}{u + \alpha_0} \right)^{\gamma-1} \right), \quad (115)$$

From the curve, the potential  $v$ , its gradient  $\partial v / \partial \alpha$  and hessian  $\partial^2 v / \partial \alpha^2$  can be constructed as follows:

$$v(\alpha) = \int_0^{\alpha - \alpha_0} f(u) du = nRT_0 \left( \frac{\alpha}{\alpha_0} - 1 \right) + \frac{nRT_0}{\gamma - 1} \left( \left( \frac{\alpha_0}{\alpha} \right)^{\gamma-1} - 1 \right), \quad (116)$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0) = nRT_0 \left( \frac{1}{\alpha_0} - \frac{1}{\alpha} \left( \frac{\alpha_0}{\alpha} \right)^{\gamma-1} \right), \quad (117)$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{df}{du}(\alpha - \alpha_0) = \frac{nRT_0}{\alpha^2} \left( \frac{\alpha_0}{\alpha} \right)^{\gamma-1}. \quad (118)$$

## 4.2 Bivariate behavior

A bivariate behavior is characterized by a bivariate energy potential, which is constructed from a generalized force-displacement that can possibly be multi-valued, meaning that for a given generalized displacement  $u$  can correspond to multiple generalized force values  $f$ . Such a curve is defined by the following parametric equations:

$$\begin{cases} u = a(x) \\ f = b(x), \end{cases} \quad (119)$$

where  $a(\cdot)$  and  $b(\cdot)$  are two continuous functions, and  $x$  is the curve parameter that monotonically increases as one moves along the curve. From the parametric curve  $(u, f) = (a(x), b(x))$ , the following bivariate potential, which takes two inputs, the measure  $\alpha$  and an internal coordinate  $t$ , is constructed

$$v(\alpha, t) = \frac{1}{2}k(t) (\alpha - \alpha_0 - a(t))^2 + b(t)(\alpha - \alpha_0 - a(t)) + \int_0^t b(\tilde{t})a'(\tilde{t})d\tilde{t}. \quad (120)$$

Each point on the force-displacement curve  $(a(x), b(x))$  corresponds to a state  $(\alpha, t) = (\alpha_0 + a(x), x)$  that is a static equilibrium state of the potential  $v$  under force  $b(x)$ , as we will demonstrate through Eqs. (126, 127, 128, 129) (see colored box below). The function  $k(\cdot)$  can be defined so that the potential  $v$  has the desired stability properties, as we will demonstrate afterwards. Its expression is given in Eq. (146). At equilibrium, the internal coordinate  $t$  plays the role of the curve parameter. From the potential, we can

compute its gradient and hessian components:

$$\frac{\partial v}{\partial \alpha} = k(t)(\alpha - \alpha_0 - a(t)) + b(t), \quad (121)$$

$$\frac{\partial v}{\partial t} = (\alpha - \alpha_0 - a(t))(b'(t) - k(t)a'(t)) + \frac{k'(t)}{2}(\alpha - \alpha_0 - a(t))^2, \quad (122)$$

$$\frac{\partial^2 v}{\partial \alpha^2} = k(t), \quad (123)$$

$$\frac{\partial^2 v}{\partial \alpha \partial t} = b'(t) - k(t)a'(t) + k'(t)(\alpha - \alpha_0 - a(t)), \quad (124)$$

$$\begin{aligned} \frac{\partial^2 v}{\partial t^2} &= (\alpha - \alpha_0 - a(t))(b''(t) - k(t)a''(t)) + a'(t)(k(t)a'(t) - b'(t)) \\ &\quad + (\alpha - \alpha_0 - a(t)) \left( \frac{k''(t)}{2}(\alpha - \alpha_0 - a(t)) - 2k'(t)a'(t) \right). \end{aligned} \quad (125)$$

To define a bivariate behavior, it therefore suffices to provide the  $a$  and  $b$  functions (and their first and second derivatives) that describe the parametric generalized force-displacement curve. In **springable**, the parametric curve can either be described as a Bezier curve (Eqs. (152, 153)) or a smooth zigzag curve (Eqs. (164, 165)) (see paragraphs below).

**Proposition:** "Points on the curve  $(a(t), b(t))$  correspond to static equilibria."

**Proof** States  $(\alpha, t)$  are equilibrium points if they are stationary points of the total potential energy  $\pi(\alpha, t) := v(\alpha, t) - f\alpha$ , meaning that they have to satisfy the two following equations,

$$\begin{aligned} 0 &= \frac{\partial \pi}{\partial \alpha} = \frac{\partial v}{\partial \alpha} - f \\ &= k(t)(\alpha - \alpha_0 - a(t)) + b(t) - f, \end{aligned} \quad (126)$$

$$\begin{aligned} 0 &= \frac{\partial \pi}{\partial t} = \frac{\partial v}{\partial t} \\ &= (\alpha - \alpha_0 - a(t)) \left( b'(t) - k(t)a'(t) + \frac{k'(t)}{2}(\alpha - \alpha_0 - a(t)) \right). \end{aligned} \quad (127)$$

From (127), we notice that

$$\alpha = \alpha_0 + a(t) \Leftrightarrow u = a(t) \quad (128)$$

is a solution, which, when plugged back into Eq. (126), yields

$$f = b(t). \quad (129)$$

From Eqs. (128) and (129), it can be noticed that each point on the force-displacement curve given in Eq. (119) is a stationary point of the total potential energy, which retrospectively explain why we define the elastic energy of the flexel using Eq. (120).

**Defining  $k(\cdot)$  to satisfy stability conditions** On the one hand, stability of equilibria are determined by the hessian matrix  $\mathbf{h}$  of the total potential energy

$$\mathbf{h}(\alpha, t) = \begin{bmatrix} \partial^2 \pi / \partial \alpha^2 & \partial^2 \pi / \partial \alpha \partial t \\ \partial^2 \pi / \partial \alpha \partial t & \partial^2 \pi / \partial t^2 \end{bmatrix}. \quad (130)$$

More precisely, stability of points under force-controlled conditions is governed by the number  $\mu_f$  of negative eigenvalues in the matrix  $\mathbf{h}$ , each indicating an unstable deformation mode. Stable points under

force-controlled conditions are therefore characterized by a hessian matrix with no negative eigenvalues ( $\mu_f = 0$ ). Similarly, stability under displacement-controlled conditions is governed by the number  $\mu_\alpha$  of negative eigenvalues in  $\mathbf{h}$  from which the row and column corresponding to the variable  $\alpha$  have been removed, that is, just the single component  $\partial^2\pi/\partial t^2$ . Note that  $\mu_f \geq \mu_\alpha$ , with  $2 \geq \mu_f \geq 0$  and  $1 \geq \mu_\alpha \geq 0$ , since the hessian matrix is 2-by-2.

On the other hand, the way that the curve folds (by reaching force and displacement extrema) should dictate the stability of equilibrium points along the  $u - f$  curve [7]. This theory tells us that each time the curve folds clockwise (counter-clockwise) by reaching a force or displacement extrema, the number of negative unstable modes, indicated by  $\mu_f$  or  $\mu_\alpha$ , under the respective loading condition should increase (decreased) by one (Fig. S10a).

To ensure consistency between the folding rules and the numbers of negative eigenvalues,  $\mu_f$  and  $\mu_\alpha$ , the following conditions must be satisfied:

$$\text{if } (\nearrow) \equiv a'(t) > 0, b'(t) > 0 \Leftrightarrow \mu_f = 0 \text{ and } \mu_\alpha = 0; \quad (131)$$

$$\text{if } (\searrow) \equiv a'(t) > 0, b'(t) < 0 \Leftrightarrow \mu_f = 1 \text{ and } \mu_\alpha = 0; \quad (132)$$

$$\text{if } (\swarrow) \equiv a'(t) < 0, b'(t) < 0 \Leftrightarrow \mu_f = 1 \text{ and } \mu_\alpha = 1; \quad (133)$$

$$\text{if } (\nwarrow) \equiv a'(t) < 0, b'(t) > 0 \Leftrightarrow \mu_f = 2 \text{ and } \mu_\alpha = 1. \quad (134)$$

Those conditions have been derived by first assuming that points along ( $\nearrow$ )-branches (that is, where  $a'(t), b'(t) > 0$ ) are stable under both force- and displacement-controlled conditions. Therefore,

- when folding clockwise from a ( $\nearrow$ )-branch to a ( $\searrow$ )-branch (or inversely counter-clockwise),  $\mu_f$  must be incremented (decremented) by one and  $\mu_\alpha$  remains 0;
- when folding clockwise from a ( $\searrow$ )-branch to a ( $\swarrow$ )-branch (or inversely counter-clockwise),  $\mu_f$  remains 1 and  $\mu_\alpha$  is incremented (decremented) by one.
- when folding clockwise from a ( $\swarrow$ )-branch to a ( $\nwarrow$ )-branch (or inversely counter-clockwise),  $\mu_f$  is incremented (decremented) by one and  $\mu_\alpha$  remains 1.

Note that other folds such as ( $\nearrow$ ) – to – ( $\nwarrow$ ) or its inverse ( $\nwarrow$ ) – to – ( $\nearrow$ ) are not permitted, since the former would necessitate  $0 \leq \mu_\alpha(\nwarrow) = -1$  and the later would necessitate  $1 \geq \mu_\alpha(\nearrow) = 2$ , which both are impossible. An example of generalized force-displacement showing valid folds is shown in Fig. S10a.

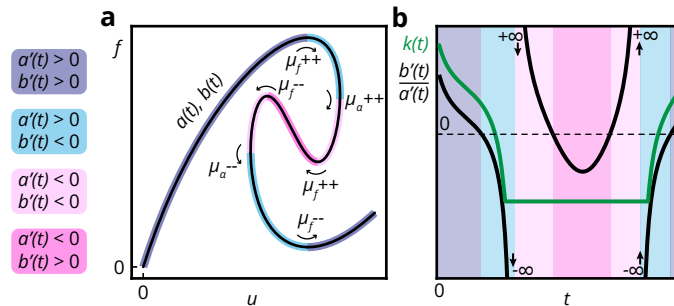


Figure S10: Example of multi-valued force-displacement curve and its slope. (a) Force-displacement curve described by parametric equations  $u = a(t)$ ,  $f = b(t)$ . (b) Slope of the tangent of the curve as a function of the curve parameter  $t$  (black) and the associated  $k(t)$  function (Eq. (146)).

To derive the conditions to satisfy the desired stability conditions (131),(132), (133) and (134) let us first express the hessian matrix when evaluated at an equilibrium point  $(t, \alpha) = (t, \alpha_0 + a(t))$ :

$$\mathbf{h}(\alpha_0 + a(t), t) = \begin{bmatrix} k(t) & b'(t) - k(t)a'(t) \\ b'(t) - k(t)a'(t) & a'(t)(k(t)a'(t) - b'(t)) \end{bmatrix}. \quad (135)$$

From conditions (131),(132), (133) and (134), notice that  $\mu_\alpha = 0$  if  $a' > 0$  else 1, meaning that

$$\begin{cases} a'(ka' - b') > 0 & \text{if } a' > 0 \\ a'(ka' - b') < 0 & \text{if } a' < 0 \end{cases} \Leftrightarrow \begin{cases} k > b'/a' & \text{if } a' > 0 \\ k < b'/a' & \text{if } a' < 0. \end{cases} \quad (136)$$

To satisfy  $\mu_f = 0$  in condition (131), the hessian matrix (Eq. (135)) must be positive-definite on ( $\nearrow$ )-branches, which, by applying Sylvester criterion, yields

$$k > 0 \quad \text{and} \quad \det \mathbf{h} = b'(ka' - b') > 0 \quad \text{when } a', b' > 0 \quad (137)$$

$\Leftrightarrow$

$$k > b'/a' \quad \text{when } a', b' > 0. \quad (138)$$

To satisfy  $\mu_f = 1$  in condition (132), the hessian matrix must be indefinite on ( $\searrow$ )-branches, which, for a 2-by-2 matrix, yields

$$\det \mathbf{h} = b'(ka' - b') < 0 \quad \text{when } a' > 0, b' < 0 \quad (139)$$

$\Leftrightarrow$

$$k > b'/a' \quad \text{when } a' > 0, b' < 0. \quad (140)$$

To satisfy  $\mu_f = 1$  in condition (133), the hessian matrix must be indefinite on ( $\swarrow$ )-branches as well:

$$\det \mathbf{h} = b'(ka' - b') < 0 \quad \text{when } a', b' < 0 \quad (141)$$

$\Leftrightarrow$

$$k < b'/a' \quad \text{when } a', b' < 0. \quad (142)$$

To satisfy  $\mu_f = 2$  in condition (134), the hessian matrix must be negative-definite on ( $\nwarrow$ )-branches, which, by applying Sylvester criterion, yields

$$k < 0 \quad \text{and} \quad \det \mathbf{h} = b'(ka' - b') > 0 \quad \text{when } a' < 0, b' > 0 \quad (143)$$

$\Leftrightarrow$

$$k < b'/a' \quad \text{when } a' < 0, b' > 0. \quad (144)$$

Inequalities (136), (138), (140), (142) and (144) are consistent and can be equivalently summarized as follows

$$\begin{cases} k(t) > b'(t)/a'(t) & \text{if } a'(t) > 0 \\ k(t) < b'(t)/a'(t) & \text{if } a'(t) < 0, \end{cases} \quad (145)$$

or, in other words, the  $k$  function must be higher than the  $b'/a'$  function when  $a' > 0$ , else lower. For generalized force-displacement curves with valid folds, the  $k$  function can always be constructed to satisfy those requirements, as illustrated in Fig. S10b. In `springable`, the  $k$  function is constructed as follows

$$k(t) = \begin{cases} k^* & \text{if } k_{\min} - k_{\max} > 2\delta \\ \hat{k}(t) & \text{else,} \end{cases} \quad (146)$$

where

$$k_{\max} = \max_{t:a'(t)>0} b'(t)/a'(t) \quad (147)$$

$$k_{\min} = \begin{cases} \min_{t:a'(t)<0} b'(t)/a'(t) & \text{if } \exists t : a'(t) < 0 \\ +\infty & \text{else,} \end{cases} \quad (148)$$

$$\delta = k_{\max}/20, \quad (149)$$

$$k^* = \min(k_{\min} - \delta, k_{\max} + \delta), \quad (150)$$

$$\hat{k}(t) = \begin{cases} \max\{b'(t)/a'(t) + \delta, k^*\} & \forall t : a'(t) > 0 \\ k^* & \forall t : a'(t) < 0. \end{cases} \quad (151)$$

**Bivariate Bezier behavior** A bivariate Bezier behavior is a Bezier curve of degree  $n$  with control points at  $(u_0 = 0, f_0 = 0), (u_1, f_1) \dots (u_n, f_n)$ , with linear extrapolation beyond the first and last control points that preserves  $\mathcal{C}_1$ -continuity. Unlike its univariate counterpart (subsection 4.1, paragraph ‘Bezier behavior’), its curve can be multi-valued. The nonlinear behavior defined by the Bezier curve can either describe the tensile behavior (a.k.a ‘tensile’ mode, Fig. S11a), the compressive behavior (a.k.a ‘compressive’ mode, Fig. S11b) or both simultaneously (a.k.a ‘symmetric’ mode, Fig. S11c). Mathematically speaking, the force-displacement curve is defined by

$$u = a(t) = \mathcal{F}(\bar{a})(t) \quad (152)$$

$$f = b(t) = \mathcal{F}(\bar{b})(t) \quad (153)$$

where the functional  $\mathcal{F}$  is defined in Eq. (70) and

$$\bar{a}(t) = \begin{cases} nu_1 t/t_{\max} & \text{if } t \leq 0 \\ \sum_{i=0}^n u_i B_{i,n}(t/t_{\max}) & \text{if } 0 < t \leq t_{\max} \\ u_n + n(u_n - u_{n-1})(t/t_{\max} - 1) & \text{if } t > t_{\max}, \end{cases} \quad (154)$$

$$\bar{b}(t) = \begin{cases} nf_1 t/t_{\max} & \text{if } t \leq 0 \\ \sum_{i=0}^n f_i B_{i,n}(t/t_{\max}) & \text{if } 0 < t \leq t_{\max} \\ f_n + n(f_n - f_{n-1})(t/t_{\max} - 1) & \text{if } t > t_{\max}, \end{cases} \quad (155)$$

where  $t_{\max} = \sum_{i=1}^n |u_i - u_{i-1}|$  and  $B_{i,n}$  are the Bernstein polynomials of degree  $n$ .

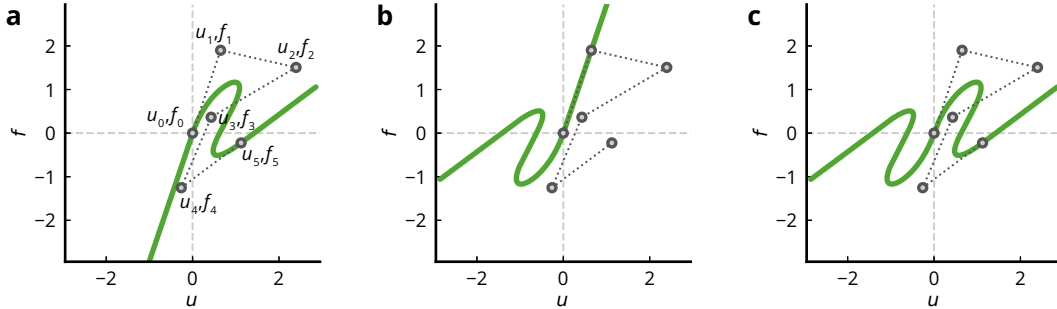


Figure S11: Multi-valued Bezier force-displacement curves for the tensile (a), compressive (b) and symmetric (c) modes.

Functions  $a$  and  $b$  are used to construct an energy potential (Eq. (120)), and compute its gradient (Eqs. (121, 122)) and hessian (Eqs. (123, 124, 125)). To calculate those quantities, the first and second derivatives of  $a$  and  $b$  must also be provided:

$$a'(t) = \mathcal{F}'(\bar{a})(t), \quad (156)$$

$$b'(t) = \mathcal{F}'(\bar{b})(t), \quad (157)$$

$$a''(t) = \mathcal{F}''(\bar{a})(t), \quad (158)$$

$$b''(t) = \mathcal{F}''(\bar{b})(t), \quad (159)$$

where the functionals  $\mathcal{F}'$  and  $\mathcal{F}''$  are defined in Eqs. (72, 73) and

$$\bar{a}'(t) = \begin{cases} nu_1/t_{\max} & \text{if } t \leq 0 \\ n \sum_{i=0}^{n-1} (u_{i+1} - u_i) B_{i,n-1}(t/t_{\max})/t_{\max} & \text{if } 0 < t \leq t_{\max} \\ n(u_n - u_{n-1})/t_{\max} & \text{if } t > t_{\max}, \end{cases} \quad (160)$$

$$\bar{b}'(t) = \begin{cases} nf_1/t_{\max} & \text{if } t \leq 0 \\ n \sum_{i=0}^{n-1} (f_{i+1} - f_i) B_{i,n-1}(t/t_{\max})/t_{\max} & \text{if } 0 < t \leq t_{\max} \\ n(f_n - f_{n-1})/t_{\max} & \text{if } t > t_{\max}, \end{cases} \quad (161)$$

$$\bar{a}''(t) = \begin{cases} n(n-1) \sum_{i=0}^{n-2} (u_{i+2} - 2u_{i+1} + u_i) B_{i,n-2}(t/t_{\max})/t_{\max}^2 & \text{if } 0 < t \leq t_{\max} \\ 0 & \text{else,} \end{cases} \quad (162)$$

$$\bar{b}''(t) = \begin{cases} n(n-1) \sum_{i=0}^{n-2} (f_{i+2} - 2f_{i+1} + f_i) B_{i,n-2}(t/t_{\max})/t_{\max}^2 & \text{if } 0 < t \leq t_{\max} \\ 0 & \text{else.} \end{cases} \quad (163)$$

**Bivariate zigzag behavior** A bivariate zigzag behavior is described by ‘smoothed zigzag’, similar to its univariate counterpart (subsection 4.1, paragraph ‘zigzag behavior’). However, this bivariate version can defined multi-valued curves. The nonlinear behavior defined by the Bezier curve can either describe the tensile behavior (a.k.a ‘tensile’ mode, Fig. S12a), the compressive behavior (a.k.a ‘compressive’ mode, Fig. S12b) or both simultaneously (a.k.a ‘symmetric’ mode, Fig. S12c). Mathematically speaking, the force-displacement curve is defined by

$$u = a(t) = \mathcal{F}(\bar{a})(t) \quad (164)$$

$$f = b(t) = \mathcal{F}(\bar{b})(t) \quad (165)$$

where the functional  $\mathcal{F}$  is defined in Eq. (70) and

$$\bar{a}(t) = P(t/t_{\max}, \mathbf{s}^a; \boldsymbol{\eta}^a; \eta_s^a), \quad (166)$$

$$\bar{b}(t) = P(t/t_{\max}, \mathbf{s}^b; \boldsymbol{\eta}^b; \eta_s^b), \quad (167)$$

with  $P$  defined in Eq. (85),  $t_{\max} = \sum_{i=1}^n |u_i - u_{i-1}|$ ,  $s_i^a = (n-1)(u_{i+1} - u_i)$ ,  $s_i^b = (n-1)(f_{i+1} - f_i)$ ,  $\eta_i^a = \eta_i^b = i/(n-1)$ , and  $\eta_s^a = \eta_s^b = \epsilon/(2n-2)$ .

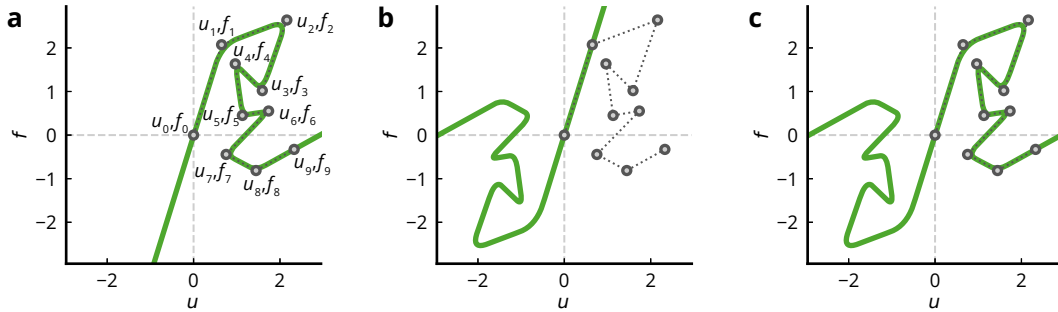


Figure S12: Multi-valued zigzag force-displacement curves for the tensile (a), compressive (b) and symmetric (c) modes.

Functions  $a$  and  $b$  are used to construct an energy potential (Eq. (120)), and compute its gradient (Eqs. (121, 122)) and hessian (Eqs. (123, 124, 125)). To calculate those quantities, the first and second

derivatives of  $a$  and  $b$  must also be provided:

$$a'(t) = \mathcal{F}'(\bar{a})(t), \quad (168)$$

$$b'(t) = \mathcal{F}'(\bar{b})(t), \quad (169)$$

$$a''(t) = \mathcal{F}''(\bar{a})(t), \quad (170)$$

$$b''(t) = \mathcal{F}''(\bar{b})(t), \quad (171)$$

where the functionals  $\mathcal{F}'$  and  $\mathcal{F}''$  are defined in Eqs. (72, 73) and

$$\bar{a}'(t) = \frac{1}{t_{\max}} \frac{\partial P}{\partial \eta}(t/t_{\max}, \mathbf{s}^a; \boldsymbol{\eta}^a; \eta_s^a), \quad (172)$$

$$\bar{b}'(t) = \frac{1}{t_{\max}} \frac{\partial P}{\partial \eta}(t/t_{\max}, \mathbf{s}^b; \boldsymbol{\eta}^b; \eta_s^b), \quad (173)$$

$$\bar{a}''(t) = \frac{1}{t_{\max}^2} \frac{\partial^2 P}{\partial \eta^2}(t/t_{\max}, \mathbf{s}^a; \boldsymbol{\eta}^a; \eta_s^a), \quad (174)$$

$$\bar{b}''(t) = \frac{1}{t_{\max}^2} \frac{\partial^2 P}{\partial \eta^2}(t/t_{\max}, \mathbf{s}^b; \boldsymbol{\eta}^b; \eta_s^b), \quad (175)$$

with  $\partial P/\partial \eta$  defined in Eq. (92), and

$$\frac{\partial^2 P}{\partial \eta^2}(\eta, \mathbf{s}, \boldsymbol{\eta}, \eta_s) = \begin{cases} 0 & \text{if } \eta \leq \eta_0 - \eta_s \\ 2a_0 & \text{if } \eta_0 - \eta_s < \eta < \eta_0 + \eta_s \\ 0 & \text{if } \eta_0 + \eta_s \leq \eta \leq \eta_1 - \eta_s \\ \vdots & \vdots \\ 2a_i & \text{if } \eta_i - \eta_s < \eta < \eta_i + \eta_s \\ 0 & \text{if } \eta_{i-1} + \eta_s \leq \eta \leq \eta_i - \eta_s \\ \vdots & \vdots \\ 2a_{n-2} & \text{if } \eta_{n-2} - \eta_s < \eta < \eta_{n-2} + \eta_s \\ 0 & \text{if } \eta_{n-2} + \eta_s \leq \eta < +\infty, \end{cases} \quad (176)$$

with  $a_i$  defined in Eq. (87).

## 5 Installation and simulation

### 5.1 Installation

Simulations shown in this article have been produced with the version 1.0.1 of this toolkit. It is available on PyPI and can easily be installed by running

```
python -m pip install springable==1.0.1
```

on Windows, or

```
python3 -m pip install springable==1.0.1
```

on MacOS and Linux. It is supported on Python versions  $\geq 3.10$ .

### 5.2 Simulating a model

To simulate a model, you can execute the following Python script

```
run_simulation.py

from springable.simulation import simulate_model

simulate_model('my_model.csv')
```

by running

```
python run_simulation.py
```

on Windows, or

```
python3 run_simulation.py
```

on MacOS and Linux. The CSV file `my_model.csv` describes the model (a flexel assembly subject to loading conditions). All the details on how to write such a file is described in section 6. The CSV model files used in the main article are provided in the section 7.

## 6 Model file specifications

A model is described in a CSV file (text file saved with extension `.csv`). The CSV file is composed of three necessary parts.

- The first part specifies the initial positions of the nodes and their boundary conditions (whether they are able to move horizontally and vertically). This is described in the **NODES** section in the CSV file.
- The second part specifies how the nodes are coupled by flexels. This is described in multiple sections, one per type of flexels: **LONGITUDINAL FLEXELS**, **ANGULAR FLEXELS**, **AREA FLEXELS**, **DISTANCE FLEXELS**, **X DISTANCE FLEXELS**, **Y DISTANCE FLEXELS** and **PATH FLEXELS**.
- The third part specifies how the structure is loaded, by defining one or multiple load steps wherein nodes are loaded along specific directions. This is described in the **LOADING** section.

Many examples with the accompanying model depictions are provided in section 7. For instructions on how to install the `springable`, you can refer to section 5.

### 6.1 The NODES section

The **NODES** section start with a line named **NODES** followed by multiple lines, one per node, with the following specification:

```
<node index>, <x>, <y>, <fixed along x>, <fixed along y>
```

where

- `<node index>` is the index of the node (its label), it must be a natural number (0, 1, 2, ...);
- `<x>` is the initial  $x$ -coordinate of the node (initial horizontal position), it can be any real number;
- `<y>` is the initial  $y$ -coordinate of the node (initial vertical position), it can be any real number;
- `<fixed along x>` is either '0' if the node is *not* fixed along  $x$  (that is, free to move horizontally), or '1' if the node is fixed along  $x$ ;
- `<fixed along y>` is either '0' if the node is *not* fixed along  $y$  (that is, free to move vertically), or '1' if the node is fixed along  $y$ .

**Example** Three nodes labeled '0', '1' and '2' are defined at initial positions (0,0), (3,2), (6,0). Nodes 0 and 2 are fixed both horizontally and vertically. Node 1 is free to move vertically, but not horizontally.

```
NODES
0, 0.0, 0.0, 1, 1
1, 3.0, 2.0, 1, 0
2, 6.0, 0.0, 1, 1
```

## Notes

- Nodes can be defined in any order, as long as no node index is missing at the end of the section. If there are 6 nodes, node indices '0', '1', '2', '3', '4' and '5' must be used, but the order does not matter.
- Nodes that are not intended to be coupled by flexels can still be defined. They must be fixed along  $x$  and  $y$ , as otherwise, zero modes (a.k.a. rigid body modes) will be present.
- When a node is defined, its initial coordinates can be used in math expressions, in order to, for example, facilitate the positioning of subsequent nodes. For example, once node '2' is defined, 'X2' and 'Y2' can be used as variables in the rest of the CSV file.

## 6.2 The FLEXELS sections

Flexels sharing the same type of geometric measure are grouped in separate sections. Flexels whose geometric measure is

- a length (subsection 3.1, Fig. S1a) are grouped under section named `LONGITUDINAL FLEXELS`;
- an angle (subsection 3.2, Fig. S1b) under section named `ANGULAR FLEXELS`;
- an area (subsections 3.4, 3.9, Fig. S1c,g) under section named `AREA FLEXELS`;
- a signed  $x$ -distance (subsection 3.5, Fig. S1d) under section named `X DISTANCE FLEXELS`;
- a signed  $y$ -distance (subsection 3.5, Fig. S1d) under section named `Y DISTANCE FLEXELS`;
- a signed point-line distance (subsection 3.7, Fig. S1e) under section named `DISTANCE FLEXELS`;
- a path length (subsection 3.8, Fig. S1f) under section named `PATH FLEXELS`.

Under each section, a flexel is specified as follows:

```
<list of node indices>, <nonlinear behavior>, <natural measure>
```

where

- `<list of nodes indices>` specifies the nodes coupled by the flexel (see paragraph 6.2.1 for more details);
- `<nonlinear behavior>` (together with `<natural measure>`) specifies the intrinsic nonlinear of the flexel (see paragraph 6.2.2 for more details);
- `<natural measure>` describes  $\alpha_0$ : the natural measure of the flexel, that is its geometric measure at rest. It can be any real number. It is used with the `<nonlinear behavior>` to construct the intrinsic nonlinear behavior of the flexel. It is an optional parameter. If not provided, the natural measure defaults to the the geometric measure computed from the initial positions of the nodes provided in the `NODES` section.

**Example** Two longitudinal flexels couples nodes ‘0’ and ‘1’ and nodes ‘1’ and ‘2’ respectively. One angular flexel is defined with an associated angle defined by nodes ‘0’, ‘1’ and ‘2’ (with the vertex on node ‘1’). The longitudinal flexels are characterized by intrinsic Bezier behavior in compression. Their natural length (rest length) is the distance computed from the initial positions of their nodes. The angular flexel is characterized by a linear behavior (with spring constant 1.0) and a rest angle of  $\pi$  radians (180°).

```

NODES
0, 0.0, 0.0, 1, 1
1, 3.0, 2.0, 1, 0
2, 6.0, 0.0, 1, 1

LONGITUDINAL FLEXELS
0-1, BEZIER(u_i=[0.83; 0.74; 2.0]; f_i=[0.48; -0.83; 0.52]; mode=-1)
1-2, BEZIER2(u_i=[2.9; -2.3; 2.8]; f_i=[0.73; -1.0; 0.38]; mode=-1)

ANGULAR FLEXELS
0-1-2, LINEAR(k=1.0), PI

```

### 6.2.1 Specifying the flexel’s nodes

- For a longitudinal flexel, use

```
<node N0 index>-<node N1 index>
```

(see subsection 3.1).

- For an angular flexel, use

```
<node N0 index>-<node N1 index>-<node N2 index>
```

(see subsection 3.2).

- For an area flexel, use

```
<node N0 index>-<node N1 index>-...-<node N n-1 index>
```

(see subsection 3.4).

- For an  $x$ - (or a  $y$ -) distance flexel, use

```
<node N0 index>-<node N1 index>
```

(see subsection 3.5).

- For a distance flexel, use

```
<node N0 index>-<node N1 index>-<node N2 index>
```

(see subsection 3.7).

- For a path flexel, use

```
<node N0 index>-<node N1 index>-...-<node Np index>
```

(see subsection 3.8).

- For an area-with-holes flexel, use

```
(<area0 node indices>)-(<area1 node indices>)-...-(<area p-1 node indices>)
```

(see subsection 3.9). Example: (0-1-2-3-4)-(5-6-7)-(8-9-10-11). Note that flexels whose geometric measure is an area or an area with holes are grouped under the same section **AREA FLEXELS**. Their actual geometric type will be determined from the way that their nodes are specified.

## 6.2.2 Specifying the flexel's intrinsic nonlinear behavior

- For a linear behavior, use

```
LINEAR(k=<k>)
```

where  $\langle k \rangle$  is the spring constant (see Eq. (62)).

- For a logarithmic behavior, use

```
LOGARITHMIC(k=<k>)
```

(see Eq. (66)).

- For a univariate Bezier behavior, use

```
BEZIER(u_i=[<u1>; <u2>; ... ; <un>]; f_i=[<f1>; <f2>; ... ; <fn>]; mode=<mode>)
```

(see Eq. (74)).  $\langle \text{mode} \rangle$  is either '1' for 'tensile' mode, '-1' for 'compressive mode' or '0' for 'symmetric mode'. It is an optional parameter; if not provided, it defaults to '0'.

- For a  $\mathcal{C}_1$ -piecewise behavior, use

```
PIECEWISE(k_i=[<k0>; ... ; <k n-1>]; u_i=[<u0>; ... ; <u n-2>]; us=<us>; mode=<mode>)
```

(see Eq. (83)).  $\langle \text{mode} \rangle$  is either '1' for 'tensile' mode, '-1' for 'compressive mode' or '0' for 'symmetric mode'. It is an optional parameter; if not provided, it defaults to '0'.

- For a univariate zigzag behavior, use

```
ZIGZAG(u_i=[<u1>; ... ; <un>]; f_i=[<f1>; ... ; <fn>]; epsilon=<epsilon>; mode=<mode>)
```

(see Eq. (96)).  $\langle \text{mode} \rangle$  is either '1' for 'tensile' mode, '-1' for 'compressive mode' or '0' for 'symmetric mode'. It is an optional parameter; if not provided, it defaults to '0'.

- For a contact behavior, use

```
CONTACT(f0=<f0>; uc=<uc>; delta=<alpha_delta>)
```

(see Eq. (105)).

- For an isothermal behavior, use

```
ISOTHERMAL(n=<n>; R=<R>; T0=<T0>)
```

(see Eq. (110)).

- For an isentropic behavior, use

```
ISENTROPIC(n=<n>; R=<R>; T0=<T0>; gamma=<gamma>)
```

(see Eq. (115)).

- For a bivariate Bezier behavior, use

```
BEZIER2(u_i=[<u1>; <u2>; ... ; <un>]; f_i=[<f1>; <f2>; ... ; <fn>]; mode=<mode>)
```

(see Eqs. (152, 153)). `<mode>` is either '1' for 'tensile' mode, '-1' for 'compressive mode' or '0' for 'symmetric mode'. It is an optional parameter; if not provided, it defaults to '0'.

- For a bivariate zigzag behavior, use

```
ZIGZAG2(u_i=[<u1>; ... ; <un>]; f_i=[<f1>; ... ; <fn>]; epsilon=<epsilon>; mode=<mode>)
```

(see Eqs. (164, 165)). `<mode>` is either '1' for 'tensile' mode, '-1' for 'compressive mode' or '0' for 'symmetric mode'. It is an optional parameter; if not provided, it defaults to '0'.

## Notes

- A nonlinear behavior can be saved in a separate CSV file and used in a model file using

```
FROMFILE(<nonlinear behavior csv file>)
```

Example:

```
NODES
0, 0.0, 0.0, 1, 1
1, 3.0, 2.0, 1, 0
2, 6.0, 0.0, 1, 1

LONGITUDINAL FLEXELS
0-1, BEZIER(u_i=[0.83; 0.74; 2.0]; f_i=[0.48; -0.83; 0.52]; mode=-1)
1-2, BEZIER2(u_i=[2.9; -2.3; 2.8]; f_i=[0.73; -1.0; 0.38]; mode=-1)

ANGULAR FLEXELS
0-1-2, FROMFILE('custom_nonlinear_behavior.csv'), PI
```

where the `custom_nonlinear_behavior.csv` is for example:

```
custom_nonlinear_behavior.csv
```

```
BEZIER2(u_i=[0.21; -0.1; 3.14]; f_i=[1.0; -2.0; +3.0]; mode=0)
```

The file path to the behavior is relative to the working directory, that is, the directory from where the script is run. If the CSV behavior file lives in a subdirectory `path/to/behavior.csv` relative to the working directory, then we would use

```
FROMFILE('path'; 'to'; 'behavior.csv')
```

To specify a CSV behavior file that would live in a subdirectory `relative/path/to/behavior.csv` relative to the CSV model file instead, we can use the keyword `HERE` that encodes the directory where the CSV model file lives (relative to the working directory) as follows:

```
FROMFILE(HERE; 'relative'; 'path'; 'to'; 'behavior.csv')
```

- Nonlinear behaviors can be interactively tuned and created using the ‘behavior creation’ graphical interface (Fig. S13), which can be started by running the following Python script

```
from springable.behavior_creation import start
start()
```

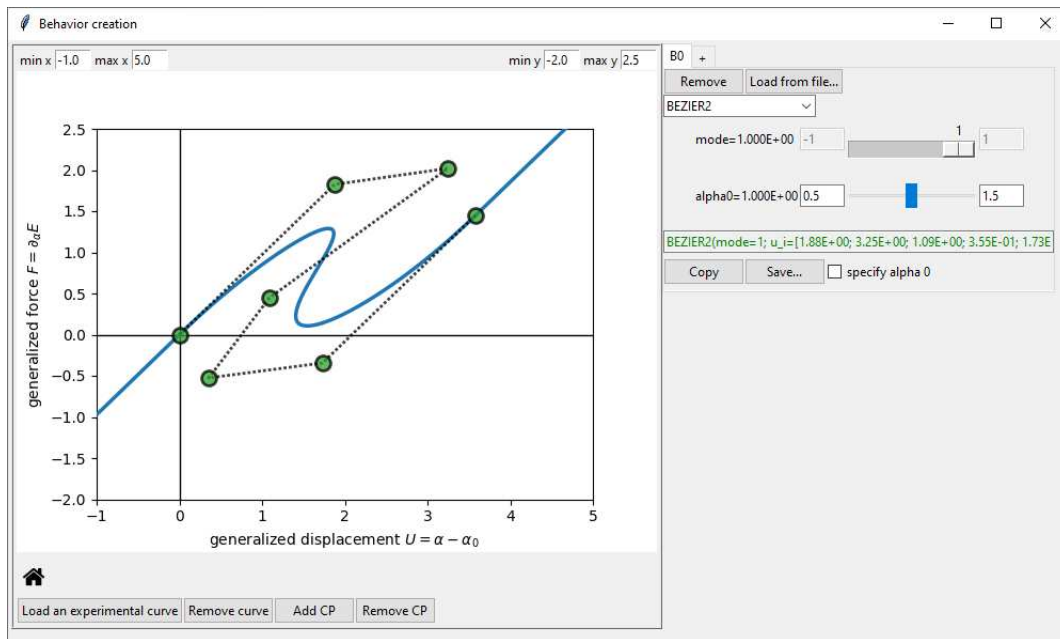


Figure S13: Screenshot of the behavior creation graphical interface, which can be used to make custom nonlinear behavior and generate the corresponding code to be pasted or imported in the model CSV file.

### 6.3 The LOADING section

The **LOADING** section is composed of one or multiple load steps. Each load step describes an additional force load acting on the nodes, relative to the end state reached at the previous load step. When multiple nodes are loaded within the same load step, each additional force component will evolve in proportion to a single parameter. In other words, each load step is characterized by a step load vector direction  $\tilde{\mathbf{F}}^{\text{dir}}$  that remains constant during the entire load step (Eq. (13)). Before each load step, some nodes that were free to move during the previous load step can be blocked (along  $x$ ,  $y$  or both), so that they remain fixed (along  $x$ ,  $y$  or both) in the state reached at the end of the previous load step. The starting state of the first load step starts in the state reached by minimizing the total elastic energy  $E$  (Eq. (6)) using the configuration described in the **NODES** section as initial guess. The **LOADING** section is structured as follows:

```
<load step>
then
<load step>
then
...
```

where each `<load step>` is composed of one of multiple lines, one per loaded coordinate, as follows:

```
<node index>, <direction>, <force>, <max displacement>
```

with

- `<node index>` is the index of the node on which the force is applied,
- `<direction>` is either ‘X’ or ‘Y’ depending on whether the force is applied along the  $x$  or  $y$  coordinate of the node,
- `<force>` is the signed magnitude of the additional force applied on node `<node index>` along `<direction>` (negative or positive real number),
- `<max displacement>` is the signed magnitude of the additional displacement of node `<node index>` along `<direction>` beyond which the load step is considered completed (positive or negative real number). It is an optional parameter; if not provided, the load step ends when the applied force is reached.

In addition to these ‘nodal load lines’, the `<load step>` can optionally start with a list of coordinates that were free that need to be blocked before applying the load step, using the following specification:

```
block
<node index>, <direction>
<node index>, <direction>
...
```

where each `<load index>` is the index of the node that will be blocked along the direction specified in `<direction>`.

**Example** A load of  $-3.0$  is first applied along the vertical direction of node 3 (so, a downward force of magnitude 3.0). When the prescribed force load is reached or when node 3 is moved downward by more than 4.0 (whichever comes first), the load step is completed. Then, node 3 is blocked vertically and node 1 is block horizontally, followed by the application of a vertical force of magnitude 6.0 on node 1.

```

NODES
0, 0.0, 0.0, 1, 1
1, 3.0, 2.0, 0, 0
2, 6.0, 0.0, 1, 1
3, 0.0, -1.0, 1, 0

LONGITUDINAL FLEXELS
0-1, BEZIER(u_i=[0.83; 0.74; 2.0]; f_i=[0.48; -0.83; 0.52]; mode=-1)
1-2, BEZIER(u_i=[0.81; 0.72; 2.0]; f_i=[0.45; -0.81; 0.52]; mode=-1)
1-3, LINEAR(k=2.0)

ANGULAR FLEXELS
0-1-2, LINEAR(k=1.0), PI

LOADING
3, Y, -3.0, -4.0
then
block
3, Y
1, X
1, Y, 6.0

```

## Notes

- When multiple loads are applied within the same load step, they will evolve proportionally in *force*, not *displacement*.
- Each load is always relative to the state reached at the previous load step. So, when multiple load steps are specified, each load (force and max displacement) is *not absolute*, but relative.
- Within a single load step, the same node can be loaded multiple times. If the direction is the same, the loads will be simply added together to make an equivalent load. If the directions are different, the node will be effectively loaded in a specific direction that is not purely horizontal or vertical.

## 6.4 The PARAMETERS section

Optionally, parameters can be defined at the start of the CSV model file, under a separate section starting with **PARAMETERS** using the following specification for each parameter:

```
<parameter name>, <parameter value>
```

where

- `<parameter name>` is the name of the parameter (which should only contains letters, numbers and underscores '\_', no space or other special characters), it is also recommended to use lowercase only, as uppercase are reserved for keywords;
- `<parameter value>` is the value of the parameter, it can be any real number, or a string of text between single quotes (which can be useful if the parameter is the name of the custom nonlinear behavior file).

**Example** Multiple parameters are defined: `width=6`, `height=3`, `angle=45`, `nonlinear_behavior0='curve0.csv'`, `nonlinear_behavior1='curve1.csv'`, `stiffness=1`, `f=3`.

```
PARAMETERS
width, 6.0
height, 3.0
angle, 45
nonlinear_behavior0, 'curve0.csv'
nonlinear_behavior1, 'curve1.csv'
stiffness, 1.0
f, 3.0

NODES
0, 0.0, 0.0, 1, 1
1, width/2, width/2*TAN(angle/180*PI), 1, 0
2, width, 0.0, 1, 1
3, 0.0, -width/3, 1, 0

LONGITUDINAL FLEXELS
0-1, FROMFILE(nonlinear_behavior0)
1-2, FROMFILE(nonlinear_behavior0)
1-3, FROMFILE(nonlinear_behavior1)

ANGULAR FLEXELS
0-1-2, LINEAR(k=stiffness), PI

LOADING
3, Y, -f, -4.0
```

## Notes

- Numerical parameters can be combined in mathematical expressions using operators `+`, `-`, `*`, `/`, `**` (exponentiation) and functions `SIN()`, `COS()`, `TAN()`, `ARCSIN()`, `ARCCOS()`, `ARCTAN()`, `SQRT()`, anywhere in the model file *outside the parameter section*, where a real number is expected (so numerical parameters cannot be used to describe a node index).
- A parameter cannot be defined as a function of other parameters.
- The parameter `PI` can be used directly without having to define it to represent the number  $\pi$ .
- Once a node is defined (with, let's say, node index = 4), parameters `X4` and `Y4` are automatically available to describe the initial  $x$  and  $y$  positions of node 4 (as described in the `NODES` section).

## 7 Model descriptions

The model CSV files used in the main manuscript are provided in this section. To simulate all those models, the following Python script can be run.

```
from springable.discover import run_simulations_from_article

run_simulations_from_article()
```

To simulate the model of each individual figure, you can refer to the Python scripts specified at the end of each subsection 7.x (see herein below).

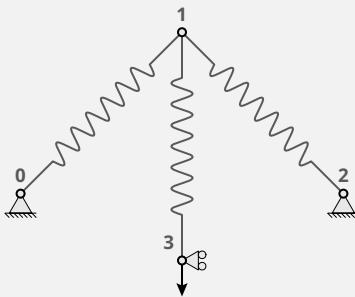
For instructions on how to install the `springable`, you can refer to section 5.

### 7.1 Figure 1

#### 7.1.1 Nonmonotonic Von-Mises truss (Fig. 1a)

Model CSV file for Fig. 1a: `fig1a_model.csv`

```
PARAMETERS
alpha, 45.0
l, 1.0
NODES
0, 0.0, 0.0, 1, 1
1, l*COS(alpha/180*PI), l*SIN(alpha/180*PI), 0, 0
2, 2*l*COS(alpha/180*PI), 0.0, 1, 1
3, l*COS(alpha/180*PI), Y1-l, 1, 0
LONGITUDINAL FLEXELS
0-1, LINEAR(k=0.6)
1-2, LINEAR(k=0.6)
1-3, LINEAR(k=20.0)
LOADING
3, Y, -0.5, -1.2*2*l*SIN(alpha/180*PI)
```



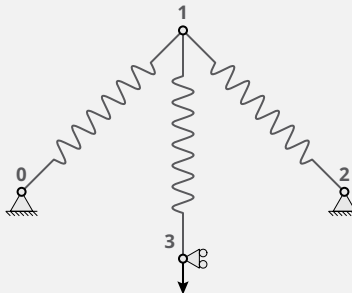
### 7.1.2 Multi-valued Von-Mises truss (Fig. 1b)

Model CSV file for Fig. 1b: fig1b\_model.csv

```

PARAMETERS
alpha, 45.0
1, 1.0
NODES
0, 0.0, 0.0, 1, 1
1, 1*COS(alpha/180*PI), 1*SIN(alpha/180*PI), 0, 0
2, 2*1*COS(alpha/180*PI), 0.0, 1, 1
3, 1*COS(alpha/180*PI), Y1-1, 1, 0
LONGITUDINAL FLEXELS
0-1, LINEAR(k=1.0)
1-2, LINEAR(k=1.0)
1-3, LINEAR(k=0.33)
LOADING
3, Y, -0.5, -1.2*2*1*SIN(alpha/180*PI)

```



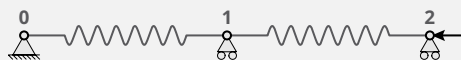
### 7.1.3 Nonlinear flexels in series (Fig. 1e)

Model CSV file for Fig. 1e: fig1e\_model.csv

```

NODES
0, 0.0, 0.0, 1, 1
1, 4.0, 0.0, 0, 1
2, 8.0, 0.0, 0, 1
LONGITUDINAL FLEXELS
0-1, FROMFILE('fig1c_behavior.csv')
2-1, FROMFILE('fig1d_behavior.csv')
LOADING
2, X, -0.325, -3.5

```



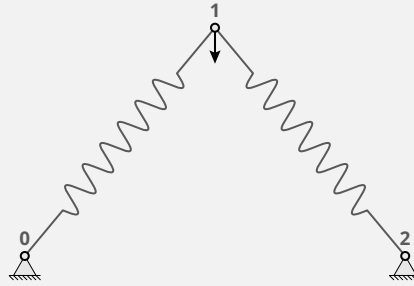
### 7.1.4 Nonlinear flexels connected at an angle (Fig. 1f)

Model CSV file for Fig. 1f: fig1f\_model.csv

```

NODES
0, 0.0, 0.0, 1, 1
1, 2.5, 3.0, 0, 0
2, 5.0, 0.0, 1, 1
LONGITUDINAL FLEXELS
0-1, FROMFILE('fig1c_behavior.csv')
1-2, FROMFILE('fig1d_behavior.csv')
LOADING
1, Y, -2.0, -5.5

```



### 7.1.5 Nonlinear behaviors (Fig. 1c,d,g,h)

Nonlinear behavior of Fig. 1c: fig1c\_behavior.csv

```
BEZIER(u_i=[8.323E-01; 7.419E-01; 2.019E+00]; f_i=[4.784E-01; -8.377E-01; 5.216E-01]; mode=-1)
```

Nonlinear behavior of Fig. 1d: fig1d\_behavior.csv

```
BEZIER2(u_i=[2.931E+00; -2.323E+00; 2.841E+00]; f_i=[7.294E-01; -1.045E+00; 3.831E-01]; mode=-1)
```

Nonlinear behavior of Fig. 1g: fig1g\_behavior.csv

```
ZIGZAG2(u_i=[9.355E-01; 4.065E-01; 2.845E+00; 1.490E+00; 1.335E+00; -1.484E-01; 2.381E+00; 1.839E+00; 3.503E+00]; f_i=[1.395E-01; -1.633E-01; 2.386E-01; -1.652E-01; 1.605E-01; -2.490E-01; 1.624E-01; -1.576E-01; 1.243E-01]; epsilon=7.500E-01; mode=-1)
```

Nonlinear behavior of Fig. 1h: fig1h\_behavior.csv

```
ZIGZAG2(u_i=[6.742E-01; 3.806E-01; 1.649E+00; 2.477E+00; 1.765E+00; 1.032E+00; 9.484E-01; -1.086E-01; 1.589E+00; 1.996E+00; 1.303E+00; 2.206E+00; 3.682E+00; 4.690E+00; 4.028E+00; 4.735E+00; 5.533E+00]; f_i=[2.085E-01; -2.071E-01; 1.530E-01; 9.885E-02; 5.267E-02; -1.911E-01; 2.460E-01; -3.543E-01; 1.190E-01; 1.017E-01; -1.465E-01; -1.371E-02; 1.227E-02; 1.553E-01; -1.356E-01; -5.714E-02; 1.830E-01]; epsilon=7.500E-01; mode=-1)
```

### 7.1.6 Simulation script

To simulate the models related to Fig. 1, the following script can be run. The CSV behavior files must live in the same (working) directory as the CSV model files.

```
from springable.simulation import simulate_model

simulate_model('fig1a_model.csv')
simulate_model('fig1b_model.csv')
simulate_model('fig1e_model.csv')
simulate_model('fig1f_model.csv')
```

## 7.2 Figure 3

### 7.2.1 Example of nonlinear angular flexel (Fig. 3a)

Model CSV file of Fig. 3a: fig3a\_model.csv

```
NODES
0, -1.0, 0.0, 1, 1
1, 1.0, 0.0, 0, 1
2, 0.0, 0.15, 0, 0
ANGULAR FLEXELS
1-2-0, FROMFILE('fig3a_behavior.csv')
LONGITUDINAL FLEXELS
0-2, LINEAR(k=1.0)
2-1, LINEAR(k=1.0)
LOADING
2, Y, -5.0
```



Nonlinear behavior used in model of Fig. 3a: fig3a\_behavior.csv

```
BEZIER2(u_i=[2.0E-01; 10E-01; 10E-01; -13.33E-02; -2E-01; 3.3E-01; 6.774E-01]; f_i=[2.749E+00; 3.297E+00; 1.515E-01; 1.623E+00; 1.190E+00; -2.648E+00; 1.364E+00])
```

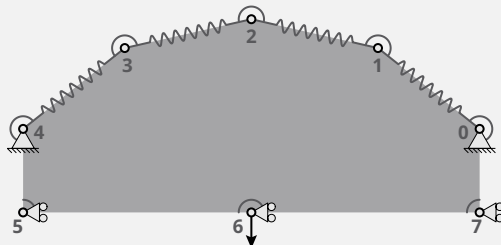
## 7.2.2 Example of nonlinear area flexel (Fig. 3b)

Model CSV file of Fig. 3b: fig3b\_model.csv

```

PARAMETERS
diameter, 10.0
alpha0, 0.675
kl, 2.0
kr, 0.075
NODES
0, diameter/2 * COS(alpha0 + (PI-2*alpha0) / 4 * 0), diameter/2 * SIN(alpha0 + (PI-2*alpha0) / 4 * 0), 1, 1
1, diameter/2 * COS(alpha0 + (PI-2*alpha0) / 4 * 1), diameter/2 * SIN(alpha0 + (PI-2*alpha0) / 4 * 1), 0, 0
2, diameter/2 * COS(alpha0 + (PI-2*alpha0) / 4 * 2), diameter/2 * SIN(alpha0 + (PI-2*alpha0) / 4 * 2), 0, 0
3, diameter/2 * COS(alpha0 + (PI-2*alpha0) / 4 * 3), diameter/2 * SIN(alpha0 + (PI-2*alpha0) / 4 * 3), 0, 0
4, diameter/2 * COS(alpha0 + (PI-2*alpha0) / 4 * 4), diameter/2 * SIN(alpha0 + (PI-2*alpha0) / 4 * 4), 1, 1
5, X4, Y4-diameter/7, 1, 0
6, (X0+X4)/2, Y4-diameter/7, 1, 0
7, X0, Y4-diameter/7, 1, 0
LONGITUDINAL FLEXELS
0-1, LINEAR(k=k1)
1-2, LINEAR(k=k1)
2-3, LINEAR(k=k1)
3-4, LINEAR(k=k1)
ANGULAR FLEXELS
7-0-1, LINEAR(k=0.99*kr)
0-1-2, LINEAR(k=0.98*kr)
1-2-3, LINEAR(k=1.01*kr)
2-3-4, LINEAR(k=1.00*kr)
3-4-5, LINEAR(k=1.02*kr)
6-5-4, LINEAR(k=40.0)
7-6-5, LINEAR(k=40.0)
0-7-6, LINEAR(k=40.0)
AREA FLEXELS
0-1-2-3-4-5-6-7, ISOTHERMAL(n=0.14; R=1.0; T0=4.0)
LOADING
6, Y, -0.25, -diameter*1.2

```



### 7.2.3 Example of nonlinear path flexel (Fig. 3c)

Model CSV file of Fig. 3c: fig3c\_model.csv

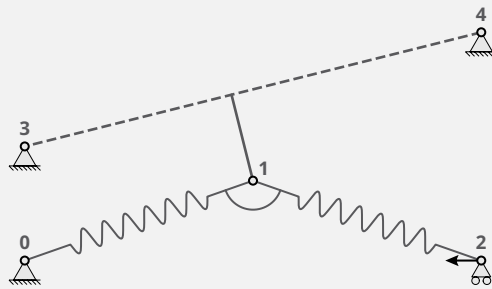
```
NODES
0, 0.0, 0.0, 1, 1
1, 0.5, 0.5, 0, 0
2, -1.5, 0.0, 1, 1
3, -2.0, 0.0, 0, 1
LONGITUDINAL FLEXELS
0-1, LINEAR(k=20.0)
ANGULAR FLEXELS
1-0-2, LINEAR(k=0.2)
PATH FLEXELS
1-2-3, PIECEWISE(k_i=[0.02;10.0];u_i=[0.5];us=0.01)
LOADING
3, X, -1.25
```



## 7.2.4 Example of nonlinear distance flexel (Fig. 3d)

Model CSV file of Fig. 3d: fig3d\_model.csv

```
NODES
0, 0.0, 0.0, 1, 1
1, 1.0, 0.35, 0, 0
2, 2.0, 0.0, 0, 1
3, 0.0, 0.5, 1, 1
4, 2.0, 1.0, 1, 1
LONGITUDINAL FLEXELS
0-1, LINEAR(k=20.0)
1-2, LINEAR(k=20.0)
ANGULAR FLEXELS
0-1-2, LINEAR(k=2.0)
DISTANCE FLEXELS
1-4-3, CONTACT(f0=3.0;uc=0.05;delta=0.0)
LOADING
2, X, -10.0
```



## 7.2.5 Simulation script

To simulate the models related to Fig. 3, the following script can be run. The CSV behavior files must live in the same (working) directory as the CSV model files.

```
from springable.simulation import simulate_model

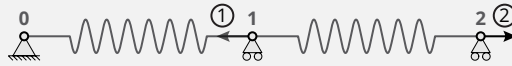
simulate_model('fig3a_model.csv')
simulate_model('fig3b_model.csv')
simulate_model('fig3c_model.csv')
simulate_model('fig3d_model.csv')
```

## 7.3 Figure 4

### 7.3.1 Model of the two nonlinear building blocks in series (Fig. 4d)

Model CSV file of Fig. 4d: fig4d\_model.csv

```
NODES
0, 0.0, 0.0, 1, 1
1, 25.0, 0.0, 0, 1
2, 50.0, 0.0, 0, 1
LONGITUDINAL FLEXELS
0-1, FROMFILE('block2_fig4c_behavior.csv')
1-2, FROMFILE('block1_fig4c_behavior.csv')
LOADING
1, X, -2.0*9.81/1000
then
2, X, 1.0, 25
```



Nonlinear behavior of Fig. 4c-top: block1\_fig4c\_behavior.csv

```
BEZIER(u_i=[3.5435E+00; 4.6615E+00; 9.1484E+00; 1.2964E+01]; f_i=[1.4147E+00; 4.6408E-01; -2.8637E-01; 8.8110E-01])
```

Nonlinear behavior of Fig. 4c-bottom: block2\_fig4c\_behavior.csv

```
BEZIER(u_i=[4.6731E+00; 4.7645E+00; 1.1955E+01; 1.5443E+01]; f_i=[1.6709E+00; 2.4704E-01; -1.8301E-01; 3.8442E-01])
```

### 7.3.2 Simulation script

To simulate the model related to Fig. 4, the following script can be run. The CSV behavior files must live in the same (working) directory as the CSV model file.

```
from springable.simulation import simulate_model

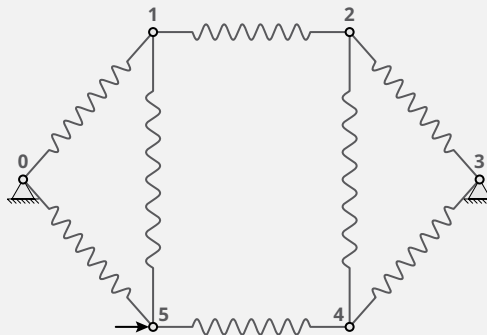
simulate_model('fig4d_model.csv')
```

## 7.4 Figure 5

### 7.4.1 Tensegrity (Fig. 5a)

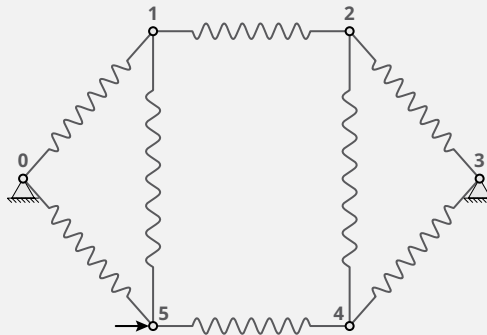
Model CSV file of Fig. 5a (no prestress): fig5atop\_model.csv

```
PARAMETERS
k, 0.5
a, 1.0
b, 1.0
c, 1.5
prestress, 1e-5
cable_behavior, 'tensegrity_cable_behavior.csv'
NODES
0, 0.0, 0.0, 1, 1
1, X0+COS(ARCSIN(c/2/a))*a, Y0+c/2, 0, 0
2, X1+b, Y1, 0, 0
3, X2+COS(ARCSIN(c/2/a))*a, Y0, 1, 1
4, X2, -Y2, 0, 0
5, X1, -Y1, 0, 0
LONGITUDINAL FLEXELS
# cables
0-1, FROMFILE(cable_behavior), a * (1-prestress)
1-2, FROMFILE(cable_behavior), b * (1-prestress)
2-3, FROMFILE(cable_behavior), a * (1-prestress)
3-4, FROMFILE(cable_behavior), a * (1-prestress)
4-5, FROMFILE(cable_behavior), b * (1-prestress)
5-0, FROMFILE(cable_behavior), a * (1-prestress)
# struts
1-5, LINEAR(k=k), c * (1+prestress)
2-4, LINEAR(k=k), c * (1+prestress)
LOADING
5, X, +0.1, -0.001
```



Model CSV file of Fig. 5a (with prestress): fig5abottom\_model.csv

```
PARAMETERS
k, 0.5
a, 1.0
b, 1.0
c, 1.5
prestress, 0.3
cable_behavior, 'tensegrity_cable_behavior.csv'
NODES
0, 0.0, 0.0, 1, 1
1, X0+COS(ARCSIN(c/2/a))*a, Y0+c/2, 0, 0
2, X1+b, Y1, 0, 0
3, X2+COS(ARCSIN(c/2/a))*a, Y0, 1, 1
4, X2, -Y2, 0, 0
5, X1, -Y1, 0, 0
LONGITUDINAL FLEXELS
# cables
0-1, FROMFILE(cable_behavior), a * (1-prestress)
1-2, FROMFILE(cable_behavior), b * (1-prestress)
2-3, FROMFILE(cable_behavior), a * (1-prestress)
3-4, FROMFILE(cable_behavior), a * (1-prestress)
4-5, FROMFILE(cable_behavior), b * (1-prestress)
5-0, FROMFILE(cable_behavior), a * (1-prestress)
# struts
1-5, LINEAR(k=k), c * (1+prestress)
2-4, LINEAR(k=k), c * (1+prestress)
LOADING
5, X, +0.1, -0.001
```



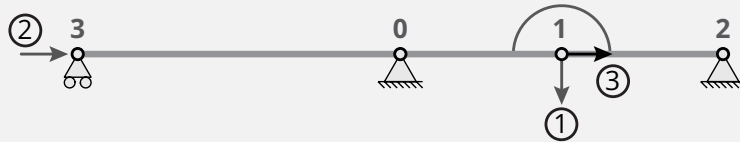
Nonlinear behavior used in model of Fig. 5a: tensegrity\_cable\_behavior.csv

```
PIECEWISE(k_i=[1e-6; 1.0];u_i=[1e-3]; us=0.99e-3;mode=1)
```

### 7.4.2 Tape spring (Fig. 5b)

Model CSV file of Fig. 5b: fig5b\_model.csv

```
NODES
0, 0.0, 0.0, 1, 1
1, 1.0, 0.0, 0, 0
2, 2.0, 0.0, 1, 1
3, -2.0, 0.0, 0, 1
PATH FLEXELS
3-0-1-2, LINEAR(k=1000.0)
ANGULAR FLEXELS
2-1-0, FROMFILE('tape_behavior.csv'), PI
LOADING
1, Y, -1e-3
then
3, X, +50.0, +0.75
then
block
3, X
1, X, 0.5, 1.0
```



Nonlinear behavior used in model of Fig. 5b: tape\_behavior.csv

```
ZIGZAG2(u_i=[1.553E-01; 3.548E-01; 5.613E-01; 3.419E-01; 1.418E-01; 1.524E+00]; f_i=[1.227E+00; 1.448E+00; 1.292E+00; 1.123E+00; 1.234E-01; 1.893E-01]; epsilon=9.000E-01)
```

### 7.4.3 Bumping buckled beams (Fig. 5c)

Model CSV file of Fig. 5c ( $d = 0.25$ ): fig5cright\_model.csv

PARAMETERS

kr0, 1.0  
 kr1, 27.0  
 d, 0.25  
 h, 1.0  
 eps, 1e-4

NODES

0, 0.0, h, 1, 0  
 1, X0+d, Y0, 1, 0  
 2, X0+eps, Y0 - h/2, 0, 0  
 3, X1-eps, Y1 - h/2, 0, 0  
 4, X0, Y0 - h, 1, 1  
 5, X1, Y1 - h, 1, 1  
 6, (X0+X1)/2, (Y0+Y1)/2+0.1, 1, 0

LONGITUDINAL FLEXELS

0-2, LINEAR(k=1e3)  
 2-4, LINEAR(k=1e3)  
 1-3, LINEAR(k=3e3)  
 3-5, LINEAR(k=3e3)

ANGULAR FLEXELS

0-2-4, LINEAR(k=kr0)  
 1-3-5, LINEAR(k=kr1)

X DISTANCE FLEXELS

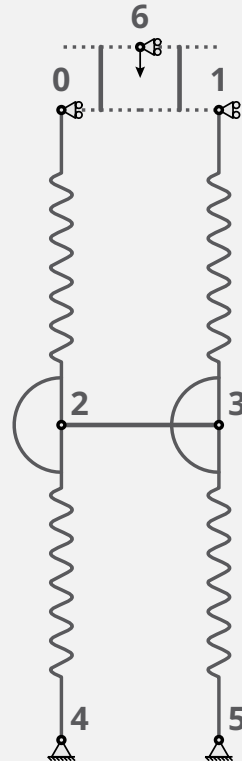
3-2, CONTACT(f0=5.0;uc=0.001;delta=0.0)

Y DISTANCE FLEXELS

6-0, LINEAR(k=1e5)  
 6-1, LINEAR(k=1e5)

LOADING

6, Y, -500.0, -h/3.0



Model CSV file of Fig. 5c ( $d = 0.30$ ): fig5cleft\_model.csv

PARAMETERS

kr0, 1.0  
 kr1, 27.0  
 d, 0.3  
 h, 1.0  
 eps, 1e-4

NODES

0, 0.0, h, 1, 0  
 1, X0+d, Y0, 1, 0  
 2, X0+eps, Y0 - h/2, 0, 0  
 3, X1-eps, Y1 - h/2, 0, 0  
 4, X0, Y0 - h, 1, 1  
 5, X1, Y1 - h, 1, 1  
 6, (X0+X1)/2, (Y0+Y1)/2+0.1, 1, 0

LONGITUDINAL FLEXELS

0-2, LINEAR(k=1e3)  
 2-4, LINEAR(k=1e3)  
 1-3, LINEAR(k=3e3)  
 3-5, LINEAR(k=3e3)

ANGULAR FLEXELS

0-2-4, LINEAR(k=kr0)  
 1-3-5, LINEAR(k=kr1)

X DISTANCE FLEXELS

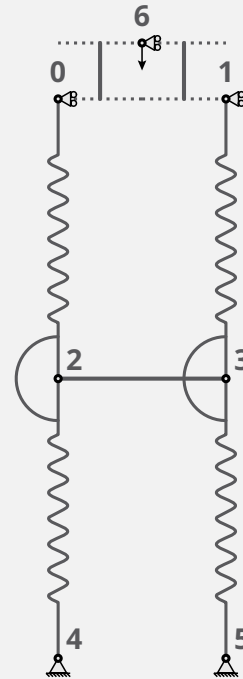
3-2, CONTACT(f0=5.0;uc=0.001;delta=0.0)

Y DISTANCE FLEXELS

6-0, LINEAR(k=1e5)  
 6-1, LINEAR(k=1e5)

LOADING

6, Y, -500.0, -h/3.0



### 7.4.4 Metafluidic gripper (Fig. 5d)

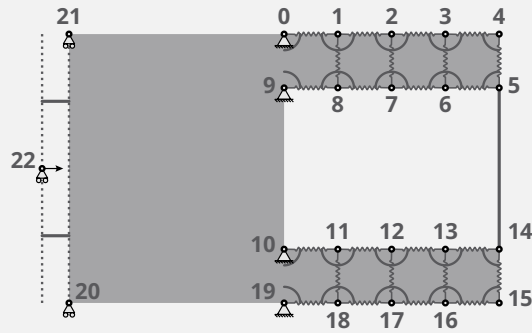
Model CSV file of Fig. 5d: fig5d\_model.csv

```
PARAMETERS
dx, 0.1
k1, 40
k0, 0.6
kv, 2.8
kr, 0.002
delta, 0.1
NODES
0 , 0*dx, 3*dx, 1, 1
1 , 1*dx, 3*dx, 0, 0
2 , 2*dx, 3*dx, 0, 0
3 , 3*dx, 3*dx, 0, 0
4 , 4*dx, 3*dx, 0, 0
5 , 4*dx, 2*dx, 0, 0
6 , 3*dx, 2*dx, 0, 0
7 , 2*dx, 2*dx, 0, 0
8 , 1*dx, 2*dx, 0, 0
9 , 0*dx, 2*dx, 1, 1
10, 0*dx, -1*dx, 1, 1
11, 1*dx, -1*dx, 0, 0
12, 2*dx, -1*dx, 0, 0
13, 3*dx, -1*dx, 0, 0
14, 4*dx, -1*dx, 0, 0
15, 4*dx, -2*dx, 0, 0
16, 3*dx, -2*dx, 0, 0
17, 2*dx, -2*dx, 0, 0
18, 1*dx, -2*dx, 0, 0
19, 0*dx, -2*dx, 1, 1
20, -4*dx, -2*dx, 0, 1
21, -4*dx, 3*dx, 0, 1
22, -4.5*dx, 0.5*dx, 0, 1
LONGITUDINAL FLEXELS
0-1, LINEAR(k=k0)
1-2, LINEAR(k=k0)
2-3, LINEAR(k=k0)
3-4, LINEAR(k=k0)
5-6, LINEAR(k=k1)
6-7, LINEAR(k=k1)
7-8, LINEAR(k=k1)
8-9, LINEAR(k=k1)
1-8, LINEAR(k=kv)
2-7, LINEAR(k=kv)
3-6, LINEAR(k=kv)
4-5, LINEAR(k=kv)
```

*(to be continued on the rightside)*

*(continued)*

```
10-11, LINEAR(k=k1)
11-12, LINEAR(k=k1)
12-13, LINEAR(k=k1)
13-14, LINEAR(k=k1)
15-16, LINEAR(k=k0)
16-17, LINEAR(k=k0)
17-18, LINEAR(k=k0)
18-19, LINEAR(k=k0)
11-18, LINEAR(k=kv)
12-17, LINEAR(k=kv)
13-16, LINEAR(k=kv)
14-15, LINEAR(k=kv)
ANGULAR FLEXELS
0-1-2, LINEAR(k=kr)
1-2-3, LINEAR(k=kr)
2-3-4, LINEAR(k=kr)
3-4-5, LINEAR(k=kr)
4-5-6, LINEAR(k=kr)
5-6-7, LINEAR(k=kr)
6-7-8, LINEAR(k=kr)
7-8-9, LINEAR(k=kr)
8-9-0, LINEAR(k=kr)
9-0-1, LINEAR(k=kr)
10-11-12, LINEAR(k=kr)
11-12-13, LINEAR(k=kr)
12-13-14, LINEAR(k=kr)
13-14-15, LINEAR(k=kr)
14-15-16, LINEAR(k=kr)
15-16-17, LINEAR(k=kr)
16-17-18, LINEAR(k=kr)
17-18-19, LINEAR(k=kr)
18-19-10, LINEAR(k=kr)
19-10-11, LINEAR(k=kr)
X DISTANCE FLEXELS
21-22, LINEAR(k=100.0)
20-22, LINEAR(k=100.0)
Y DISTANCE FLEXELS
5-14, CONTACT(f0=5.0; uc=0.0025; delta=delta)
AREA FLEXELS
0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21,
FROMFILE('metafluid_behavior.csv')
LOADING
22, X, 1.0, 3.95*dx
```



## Nonlinear behavior used in model of Fig. 5d: metafluid\_behavior.csv

```
ZIGZAG2(u_i=[6.000E-02; 5.000E-03; 7.500E-02; 1.000E-02; 9.000E-02; 1.500E-02; 1.050E-01; 2.000E-02; 1.200E-01; 2.500E-02; 1.350E-01; 3.000E-02; 1.500E-01]; f_i=[1.000E+00; 0.000E+00; 1.010E+00; 1.000E-02; 1.020E+00; 2.000E-02; 1.030E+00; 3.000E-02; 1.040E+00; 4.000E-02; 1.050E+00; 5.000E-02; 1.060E+00]; epsilon=5.000E-01)
```

### 7.4.5 Simulation script

To simulate the models related to Fig. 5, the following script can be run. The CSV behavior files must live in the same (working) directory as the CSV model files.

```
from springable.simulation import simulate_model

settings = {'radius': 0.005, 'convergence_value': 1e-8, 'detect_mechanism': False}

simulate_model('fig5abottom_model.csv', solver_settings=solver_settings)
simulate_model('fig5atop_model.csv', solver_settings=settings)
simulate_model('fig5b_model.csv', solver_settings=settings)
simulate_model('fig5cleft_model.csv', solver_settings=settings)
simulate_model('fig5cright_model.csv', solver_settings=settings)
simulate_model('fig5d_model.csv', solver_settings=settings)
```

## References

- [1] E. Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 15(7):529–551, January 1979.
- [2] M. A. Crisfield. A fast incremental/iterative solution procedure that handles “snap-through”. In AHMED K. Noor and HARVEY G. McCOMB, editors, *Computational Methods in Nonlinear Structural and Solid Mechanics*, pages 55–62. Pergamon, January 1981.
- [3] Manuel Ritto-Corrêa and Dinar Camotim. On the arc-length and other quadratic control methods: Established, less known and new implementation procedures. *Computers & Structures*, 86(11-12):1353–1368, June 2008.
- [4] E. Fermi, P Pasta, S Ulam, and M Tsingou. Studies of the nonlinear problems. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 05 1955.
- [5] T. Cohen and S. Givli. Dynamics of a discrete chain of bi-stable elements: A biomimetic shock absorbing mechanism. *Journal of the Mechanics and Physics of Solids*, 64:426–439, March 2014.
- [6] Anna Vainchtein and Lev Truskinovsky. Solitary Waves and Kinks in FPU Lattices with Soft–Hard–Soft Trilinear Interactions. *Journal of Nonlinear Science*, 34(6):113, October 2024.
- [7] J. H. Maddocks. Stability and folds. *Archive for Rational Mechanics and Analysis*, 99(4):301–328, December 1987.