Harnessing mechanical instabilities for functional structures using nonlinear building blocks

Paul Ducarme

PhD thesis, Universiteit van Amsterdam, 2026.
Harnessing mechanical instabilities for functional structures using nonlinear building blocks
Paul Ducarme

UNIVERSITEIT VAN AMSTERDAM

ASML

The work described in this thesis was performed at AMOLF (Science Park 104, 1098XG Amsterdam, The Netherlands) and the Advanced Research Center for Nanolithography, ARCNL (Science Park 106, 1098XG Amsterdam, The Netherlands). AMOLF is an institute of the Foundation for Dutch Scientific Research Institutes (NWO-I). ARCNL is a public-private partnership between the Universiteit van Amsterdam (UvA), the Vrije Universiteit Amsterdam (VU), the Rijksuniversiteit Groningen (RUG), the Netherlands Organization for Scientific Research (NWO), and the semiconductor-equipment manufacturer ASML.

Printing: Ridderprint, `www.ridderprint.nl`
Cover design: Paul Ducarme
An electronic copy of this thesis is available at
`https://dare.uva.nl`, `https://ir.amolf.nl` and `https://ir.arcnl.nl`.
ISBN 978-94-6537-430-7

Harnessing mechanical instabilities for functional structures using nonlinear building blocks

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P. P. C. C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op maandag 4 mei 2026, te 16:00 uur

door

Paul Ambroise E. Ducarme

geboren te Liège

Promotiecommissie

Promotores:	dr. ir. J.T.B. Overvelde	AMOLF
	dr. ir. C.J.M. Coulais	Universiteit van Amsterdam
Co-promotores:	prof. dr. M.L. van Hecke	Universiteit Leiden
	dr. B.A. Weber	Universiteit van Amsterdam
Overige leden:	prof. dr. D. Bonn	Universiteit van Amsterdam
	dr. J.A. Dijksman	Universiteit van Amsterdam
	dr. A. Deblais	Universiteit van Amsterdam
	dr. A.O. Krushynska	Rijksuniversiteit Groningen
	dr. S. Ramananarivo	Institut Polytechnique de Paris

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

To my parents, for their unconditional support.

Contents

1

Introduction

Shapes govern how objects deform. For example, a long rubber band stretches more easily than a small one. The corrugated segment of a straw extends progressively in steps, while the straight segment hardly elongates. The corrugated part bends and remains bent, while the tube bends and springs back. A metallic wire shaped into a coil forms a spring that can be compressed or stretched to store energy. Yarn can be knit in different patterns to achieve different textile flexibility. These examples highlight how geometry can be harnessed to change the mechanical behavior of a system and make it more functional. But, what kind of mechanical responses can we envision to achieve from this interplay between shape and material? Can we encode any deformation in a structure? Can a piece of rubber be shaped to embed advanced functionalities that are typically achieved using electronic components? Have we explored the full potential of mechanics[†]?

Before we can address these questions, we will take a moment to put into context the paradigm shift that has appeared in recent years consisting of seeking richer mechanical behaviors in regimes of large structural deformations. We will show that the mechanical behaviors of simple structures can be non-trivial when undergoing large deformations as a result of the intricate link between geometry, deformation and forces (Section 1.1). This coupling yields *geometric nonlin-*

[†]In this thesis, we will focus on a special branch of mechanics, called *statics*. Statics is the study of mechanical systems that are in a static equilibrium with the environment; wherein the deformations are assumed to occur slowly enough so that inertial and damping forces do not play a significant role.

earities that can potentially lead to *instabilities*. We will present examples where embracing the behavior of structures *beyond* the onset of those instabilities has allowed natural and artificial systems to gain functionalities. A tool that we will use extensively throughout this thesis to examine these systems is the concept of equilibrium paths, which offer useful insights when seen as *force-displacement curves*. In particular, we will show that that the profile of a force-displacement curve encodes information to understand the behavior of structures subject to instabilities. On this basis, we will draft a definition of "complexity" in quasistatic systems (Section 1.2).

Next, we will introduce the idea that additional levels of complexity can be reached when nonlinear structures are assembled together (Section 1.3). Harnessing this collective complexity is one of the core principles that has led to the development of *metamaterials*; structures made from patterns of unit cells whose global properties surpass the local ones. When operating in the regime of large deformations, we will show that they can display exotic and sometimes surprising mechanical responses, often characterized by intricate force-displacement curves.

Drawing from those insights, we wonder: *what kind of force-displacement curves and underlying mechanical behaviors can we envision to achieve purely mechanically by combining nonlinear building blocks?* From that question, we will define three specific research objectives (Section 1.4), each addressed in a separate chapter. By the end of this thesis, I hope the reader will have gained valuable insights and a set of practical tools to explore and tame the world of complex mechanics built from nonlinear building blocks, from both an experimental and computational standpoint.

1.1 Geometric nonlinearity and instability enable functionality

It might seem that mechanics is a field where our intuition can be safely trusted and where nothing is left to be explored. Everyday, we interact with structures and materials and experience how they deform. A more quantitative understanding has developed since people started to design and build structures, with most of the effort directed toward understanding how to avoid failure and ensure sturdiness, as those are important requirements in many conventional applications [1]. In that context, theory of structures has traditionally focused

on regimes where applied forces cause small and proportional deformations, and where stability must be maintained. The analytical approach presented by Lagrange in 1788 made it possible to formally determine the stability of structures [2]. It provided the back-cloth to research on elastic stability conducted during the 20th century, which was driven by the increasing use of thinner structures (more prone to instabilities) in aerospace, naval, and civil engineering [3]. Instability was viewed as a synonym of failure, as illustrated by Thompson in 1973 in his book *A general theory of elastic stability* [3]:

> *"Thus elastic instability is now a subject of prime concern to all structural engineers, and as designs become even more efficient the engineer will be faced with ever more severe and complex instabilities demanding the sophisticated treatments presented in this book. This is because an optimum design is by its very nature prone to instability. Thus if a small perturbation carries an optimum structure into an adjacent, and therefore non-optimal, configuration the structure will be unable to sustain its load and failure will occur."*

However, recent years have seen an increasing research interest in embracing mechanical instabilities as a way to embed functionalities into structures [4]. This approach exposes the richness of mechanics, allowing for surprising phenomena, defying our intuition [5]. By adopting this paradigm, structures can fulfill multiple functions, allowing for more integrated, passive and robust systems. A door has been opened to a world of unexplored and potentially useful mechanical structures. Before diving deeper into this exciting research direction and its prospect for applications, we will take a moment to describe what we mean by instabilities and how they come about in mechanical systems.

1.1.1 Instabilities as a result of geometric nonlinearities

Let us consider a symmetric arch, comprising two linear springs* of stiffness k, and loaded by a vertical force F, as illustrated in Fig. 1.1a. At equilibrium, the applied force F is counter-balanced by the vertical components of the springs' axial

*In this context, 'linear' means that the axial force f in the spring is proportional to the change u in its length, i.e. $f = ku$. The proportionality constant is the stiffness k, also known as *spring constant*.

forces. If the force causes a small deformation of the arch, the relation between the force F and the vertical displacement U of the central node is approximately linear: $F \approx (2k \sin^2 \theta)U$, with θ the inclination of the arch. From this relation, we see that the stiffness* of the arch depends on its geometry through the angle θ. Therefore, when the force is larger, the arch deforms in a way that gradually affects its stiffness, which, in turn, affects its geometry, then again its stiffness, etc. This intricate interplay between the change in geometry and the stiffness of a structure is known as *geometric nonlinearity*. Geometric nonlinearities give rise to nonlinear force-displacement relations. Those relations can be *multivalued*, meaning that multiple states can exist for a single value of applied force (Fig. 1.1c), as illustrated by the arch (Fig. 1.1a); or for a single value of applied displacement (Fig. 1.1d), as illustrated by an arch loaded through a softer spring (Fig. 1.1b).

In this thesis, we focus on the force-displacement curves of elastic† mechanical systems. The force-displacement curve of such a system is found by determining the stationary points of the *total potential energy* Π. This is done by finding the points or *equilibrium states* $\boldsymbol{Q}_{\text{eq}}$ that satisfy

$$\frac{\partial \Pi}{\partial Q_i}(\boldsymbol{Q}_{\text{eq}}, F) = 0 \quad \forall Q_i \in \boldsymbol{Q}, \tag{1.1}$$

where

$$\Pi(\boldsymbol{Q}, F) = E(\boldsymbol{Q}) - FU(\boldsymbol{Q}), \tag{1.2}$$

where $\boldsymbol{Q}$ is the array of general coordinates of the system, E is the elastic energy, and U the displacement along the direction of the external force‡ F. The

*We define the stiffness of a structure as the derivative of the force with respect to the displacement of the point where the force is applied, i.e. stiffness $= \partial F / \partial U$.

†An "elastic" system is a system that is reversible and time-independent. In other words, the system can always be reset to its initial state without producing permanent deformation or damage (in contrast to elasto-plastic systems), and behaves independently of the loading speed (in contrast to visco-elastic systems). Consequently, the structural forces at play are conservative (they derive from an energy potential), and are called "elastic" forces.

‡For the purpose of this introduction, we limit ourselves to cases where only one force acts on the system. A more general formulation that includes dead preloading forces combined with multiple forces changing together is presented in Chapter 3. Force-displacement curves can still be uniquely defined in those more advanced cases, as long as the forces evolve proportionally (Chapters 3 and 4).

set of all equilibrium points $\{\boldsymbol{Q}_{\text{eq}}, F\}$ found for various levels of force F forms the *equilibrium path* of the mechanical system. The force-displacement curve is then obtained by projecting the equilibrium path onto the F-U plane. Geometric nonlinearities cause the system of equations represented by Eq. (1.1) to be nonlinear*, thereby allowing for multiple solutions (meaning multiple possible equilibrium states) at a given load. Note that, in this thesis, statements regarding force–displacement curves can equivalently be interpreted as statements on any curve linking a conjugate pair of variables whose product defines work, such as a pressure-volume, temperature-entropy curve or a voltage-charge curve.

The stability of an equilibrium point $\boldsymbol{Q}_{\text{eq}}$ depends on the loading conditions [6]. If the system is force-driven (F is prescribed, U is measured), the stability is determined by the number of negative eigenvalues of the full hessian matrix $\boldsymbol{K}$ evaluated at that equilibrium point, i.e.

$$\boldsymbol{K}(\boldsymbol{Q}_{\text{eq}}) := \frac{\partial^2 \Pi}{\partial \boldsymbol{Q} \partial \boldsymbol{Q}^\top}(\boldsymbol{Q}_{\text{eq}}). \tag{1.3}$$

Instead, if the system is displacement-driven (U is prescribed, F is the measured reaction force), the stability is determined from the reduced hessian matrix $\tilde{\boldsymbol{K}}$, which, in case† the displacement U is the change in the coordinate Q_k, is the full matrix $\boldsymbol{K}$ from which the k-th row and column have been removed. An equilibrium point is stable if and only if all the eigenvalues are positive‡. When a stable point becomes unstable, one or more eigenvalues become zero, and the equilibrium point is said to be *critical*. The continuous portions of an equilibrium path (or force-displacement curve) delimited by critical points are called *branches*. Assessing the stability of an equilibrium point is important as it will inform us about

*Geometry is not the only source of nonlinearities and instabilities. Material nonlinearities, failure, damage, plasticity or loss of contact are other examples that bring about nonlinearities and instabilities in quasistatic mechanical systems.

†With a change of variables, it is most often possible to redefine the mechanical problem in such a way that the displacement U is itself the change in a single generalized coordinate.

‡Consequently, equilibrium points that are stable under force-driven conditions are always stable under displacement-driven conditions, and equilibrium points that are unstable under displacement-driven conditions are always unstable under force-driven conditions. The reciprocal statements are not true in general; equilibrium points can be stable under displacement-driven conditions and unstable under force-driven conditions (e.g. states represented by the solid gray line in Fig. 1.1c).

how nonlinear structures deform when an equilibrium point becomes unstable, that is, when the structure experiences an instability. When this happens, the structure faces two possible scenarios, snapping or buckling [7].

Snapping instability When the force-displacement curve is nonmonotonic (Fig. 1.1c) or "curves back" (Fig. 1.1d), the profile of the curve folds so that the force or the displacement reaches an extremum (known as a *limit point*) as one moves along the curve. If the force (resp. the displacement) is driven to a value beyond that extremum, the displacement (resp. force) changes discontinously when it reaches the critical point, causing a sudden reconfiguration known as a *snapping instability*. When it snaps, the arch is temporarily no longer in a static equilibrium and its behavior is governed by dynamical equations [8]. However, from the (static) force-displacement curve, we are still able to see in which state (called *restabilization* state) the structure will jump to and settle after the dynamic snapping transition eventually dampens out (Fig. 1.1c,d). In the case of the arch, the restabilization state corresponds to the inverted configuration, in which the arch forms a cup instead of a dome (Fig. 1.1e,f).

Buckling instability Let us now consider a pair of linear springs hinging through a linear angular spring [*], and loaded by a vertical force F, as shown in Fig. 1.2a. For any value of force F, the structure is at equilibrium in the perfectly vertical configuration, as the springs' axial force can counter-balance the applied force. However, our experience with squeezing coffee stirrers tells us that for a sufficiently high compressive force, such a system can bend on one side or the other, suggesting the presence of additional equilibrium states. Those can be visualized when plotting the set of equilibrium points of the system (Fig. 1.2b,c). Those form a path with a *bifurcation*, that is, a critical point where the path splits into multiple branches. A bifurcation point indicates that an initial mode of deformation (often symmetric) becomes unstable as the load increases, and that the structure will now follow one of the bifurcated or *buckled* branches that corresponds to a different mode of deformation (often asymmetric). This change in mode of deformation is known as a *buckling instability*. The actual buckled branch

[*] A linear 'angular' spring is a mechanical component whose internal torque τ is proportional to the change $\Delta\theta$ with respect to its rest angle, i.e. $\tau = k_r \Delta\theta$.

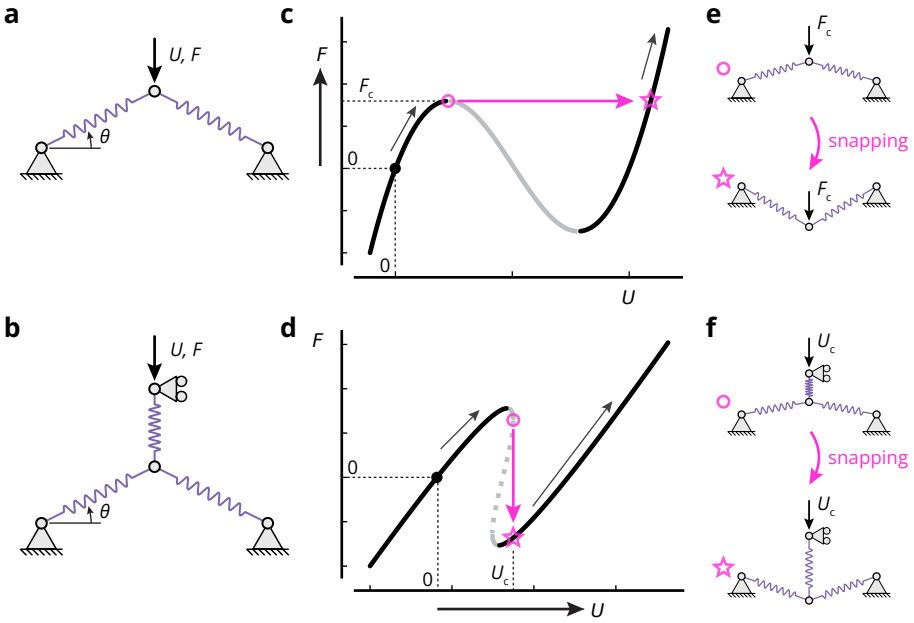


Figure 1.1: Example of snapping instabilities occurring under increasing controlled force (top) or increasing controlled displacement (bottom). **(a-b)** Archs composed of two linear springs, loaded vertically directly from the center in force (a) or through a softer spring in displacement (b). **(c-d)** Force F - displacement U curves corresponding to the mechanical systems shown in (a) and (b). Black (gray) lines denote stable (unstable) states under force-driven conditions. Solid (dashed) lines denote stable (unstable) states under displacement-driven conditions. The black dot represents the initial, unloaded state. The pink circle represents the critical state. The pink star represents the restabilization state. Arrows indicate the direction of loading and deformation. **(e-f)** Critical (top) and restabilization (bottom) states corresponding to the mechanical systems shown in (a) and (b).

that the system will follow can only be determined if imperfections are taken into account.

Unlike snapping, buckling does not necessarily bring about a fast dynamic transition in general. However, a buckling instability can coexist with a snapping instability if the bifurcated branch is also unstable [9, 10]. In such a case, a fast transition will take place.

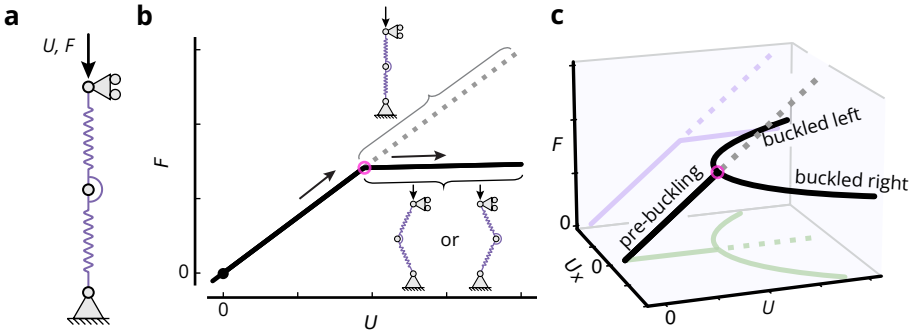


Figure 1.2: Example of buckling instability. **(a)** Beam composed of two springs hinging through an angular spring, loaded vertically from the top. **(b)** Force F - displacement U curve. Solid black (dashed gray) lines denote stable (unstable) branches. The black dot represents the initial, unloaded state. The pink circle represents the critical state. Arrows indicate the direction of loading and deformation. **(c)** Equilibrium path in the U, U_x, F space. U_x is the horizontal displacement of the central node. Light purple (green) lines represent the projection of the equilibrium path onto the F-U plane (U_x-U plane).

1.1.2 Snapping and buckling enable functionality

As shown in Figs. 1.1 and 1.2, mechanical structures display rich mechanical behaviors beyond their critical point. This complexity is leveraged in natural systems, but also, these last decades, in artificial ones. Part of this trend is attributed to the recent advances in nonlinear mathematics and geometry combined with the increased availability of soft materials and 3d-printing techniques which allowed researchers and engineers to better harness these post-instability regimes [11], by avoiding irreversible damage or embracing plasticity [12].

Nonmonotonicity in force-displacement curves causes a mismatch between the work put in by external loads and the change in potential elastic energy during the snapping transition, leading to a sudden energy release. Therefore, snapping structures are able to store and rapidly release energy, which is used as a power amplification strategy by click beetles to jump [13, 14], and mantis shrimps [15–17] and humming birds [18] to catch preys, for example (Fig. 1.3a,b). Sudden energy releases upon snapping have also been used for rapid actuation in soft robotic systems, enabling fast grasping [19–21], jumping [22–25], swimming [26, 27] or locomotion [28] (Fig. 1.3c,d).

Snapping also allows to switch between two states, which have qualitatively different geometries as they belong to stable branches separated by at least one

unstable branch, thereby amplifying small loads into large displacements [29]. This is used to make mechanical switches, akin transistors, to store and process information [30–32]. When the nonmonotonic force-displacement curve is composed of multiple stable branches intersecting the $(F = 0)$-axis, systems will maintain their new qualitative configuration after the force is released. In those cases, temporary force inputs allow to program and lock structures in different stable shapes [33]. By coupling the applied load to the geometry, nonmonotonic force-displacement curves can be harnessed to convert a steady energy input into a functional oscillating output [22, 34, 35].

Buckling, on the other hand, does not necessarily bring about quick reconfigurations. However, it can be harnessed to generate patterns or out-of-plane deformation from initially symmetric and flat configurations. This can be used in various applications, such as thin-film nanopatterning [36], deployable microstructures [37] and foldable structures [38, 39].

In this thesis, we will focus on functionalities enabled by snapping instabilities. We already see that the profile of the force-displacement curve is linked to potential functionalities. Are more complex force-displacement curves associated to more functionalities? To explore this, let us next define what we mean by "complexity" and have a look at systems combining multiple nonlinear structures.

1.2 Complexity of a quasistatic mechanical system

Complexity is often used as a vague concept to describe a surprising or seemingly unpredictable phenomenon [40]. Because definitions based on such subjective human judgments can lead to misunderstandings and ambiguity, in this section, we will give a more objective definition of complexity. Definitions of complexity have been proposed in the context of biological or dynamical systems [40, 41]. In this thesis, as we focus on quasistatic elastic mechanical systems, we devise our own definition of complexity as follows.

> *The complexity of a quasistatic mechanical system is the minimum amount of information that one would need to gather or measure in order to determine the external state of the system, and predict how that state will change upon additional loading.*

An external state can be any quantity that gives a sufficiently good description

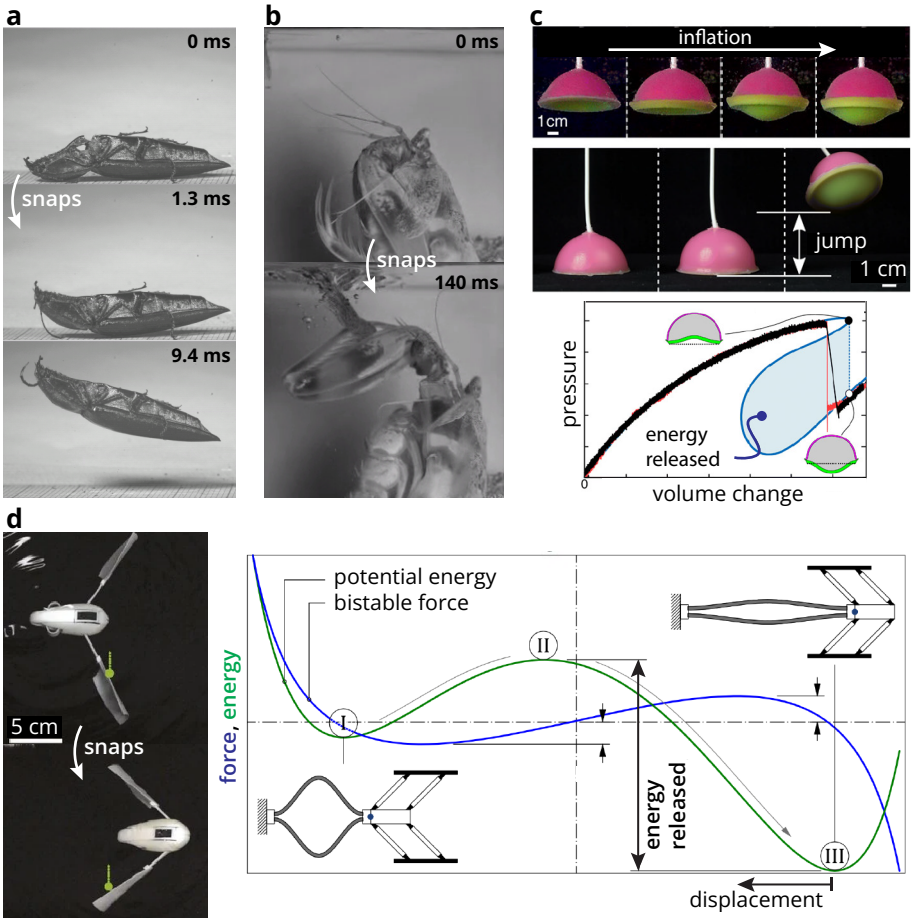


Figure 1.3: Functionalities enabled by snapping instabilities. **(a)** A click beetle snapping between two body configurations to jump. From Ref. [14], used under CC-BY license. **(b)** A zebra mantis shrimp snapping its spearing appendages to capture a prey. Reproduced from Ref. [15] with permission. **(c)** An inflatable shell able to snap and jump upon inflation. The energy released is defined by the pressure-volume curve (blue area). Reproduced from Ref. [24] with permission from AAAS. **(d)** A soft robot able to swim via snapping. The force is generated by a shape memory polymer deforming through temperature changes. The energy released can be determined from the force-displacement curve. Reproduced from Ref. [26] with permission.

of the state of the system to an external observer in the context of a given application. For example, for most applications, only knowing the elongation and the force of a coil spring is enough; there is no need to describe the state of every bit of material that constitutes the spring. So, the pair displacement U and force F is

a sufficiently good description of the state of the spring, and the external state to consider is therefore (U, F).

To illustrate the definition of complexity, we will use displacement and force, (U, F), as the external state, and consider three different systems: a linear rubber band, the arch in Fig. 1.1a and the arch in Fig. 1.1b. To evaluate the complexity of those three systems, we ask: what is the minimum amount of information that I need to measure to identify the external state and predict how that external state will change upon loading?

For a linear rubber band, we can start by measuring its stiffness k. To know its external state in a given scenario, measuring the force is enough to know the displacement, $U = F/k$, or vice versa $F = kU$; and its next state can be predicted from the stiffness. Overall, we had to measure stiffness and displacement to determine the external state and predict how that state will change upon additional loading.

For the arch in Fig. 1.1a, we first perform a tensile test* to determine the entire nonmonotonic force-displacement curve. To know the external state in a given scenario, measuring the force is not enough information to know the displacement, as multiple displacement values can correspond to a single force value (Fig. 1.1c). So, additional information about the past force history is necessary to disambiguate the external state. Alternatively, we could measure the displacement instead and map it unambiguously to the force. How the state will change is then predicted by the force-displacement curve previously measured. Overall, we had to conduct a tensile test, and measure the current and past forces (or the current displacement) to determine the external state and predict how that state will change upon additional loading.

For the arch in Fig. 1.1b, performing a tensile test where the displacement is simply increased is not enough; the displacement must be increased then decreased to retrieve the two stable branches in full. To know the external state in a given scenario, measuring either the force or the displacement is not enough in-

*A tensile test consists of measuring the reaction force of a structure under a prescribed displacement signal. Typically, both ends of the structure are attached to a pair of clamps that are slowly moved apart in a linear stage. The displacement is measured as the change in distance between the clamps while the reaction force is measured with a load cell. Note that, in this thesis, we use the term "tensile test" in the general sense of measuring a reaction force when varying a displacement, which could also be of a compressive or angular nature for example.

formation to determine the external state, as the curve is multivalued. Both force and displacement must be measured simultaneously to determine the external state unambiguously (or alternatively, keeping track of the load history). Once the external state is identified, the previously measured stable branches can be used to predict how the state will change. Overall, we had to conduct a tensile test upon both loading and unloading, and measure both the displacement and the force simultaneously to determine the external state and predict how that state will change upon additional loading.

Using our definition of complexity, we can say that the rubber band is less complex than the arch in Fig. 1.1a, which itself is less complex than the arch in Fig. 1.1b. In some applications, it is useful to also know the elastic energy E currently stored in the system (in order to evaluate the amount of energy released during snapping for example), and the external state that we consider is then (U, F, E). To determine the elastic energy, we need to know the force-displacement curve of the system, and where the state currently lies on this curve. However, measuring the entire force-displacement curve is not possible with a tensile test when the curve folds and curves back, since the unstable branch is not accessible (Fig. 1.1d). A solution is to decompose the structure into sub-structures or *building blocks* whose individual force-displacement curves can be measured entirely with a tensile test. The current energy E of the full structure can then be determined in a given situation by measuring the displacements u_j of each building block within the structure, mapping each displacement to the corresponding individual energy e_j, which are finally added together ($E = \sum e_j$). We consequently need more measurements to describe the external state.

These examples illustrate that the profile of the force-displacement curve gives insights into the complexity of a structure. The number of folds (limit points) and where the stable and unstable branches lie inform us about the multivaluedness of the system, which is closely connected to our definition of complexity. Characterizing the behavior and measuring the state of complex systems can be facilitated by considering those structures as assemblies of nonlinear building blocks. Therefore, combining building blocks could help design complex structures with behaviors that are easier to measure and understand.

1.3 Collective complexity arises from the interplay of nonlinear behaviors

Complex behaviors resulting from the interplay of less complex elements is ubiquitous across physics. Mechanical metamaterials can be seen as a platform to investigate and leverage this collective complexity in the quasistatic mechanical domain. The term "mechanical metamaterial" is often used to describe an artificial structure made from a repeating pattern of engineered unit cells, with global effective properties, following from the unit cells' local properties, that go beyond what is observed in conventional materials. Such properties are typically achieved through the rational design of the internal structure, which operates at a scale that allows geometric tuning, in contrast to other materials whose structure is not so easily accessible and tunable [42].

Research on mechanical metamaterials initially focused on achieving "extreme" elastic moduli in the linear regime, such as negative Poisson ratio [43], high specific stiffness [44] and quasi-zero shear modulus [45]. More recent trends utilize larger elastic deformations leading to nonlinearities and instabilities [46]. Slender structures have been used to make bistable unit cells and form metamaterials that maintain a high-elastic-energy state after an impact, allowing to trap elastic energy and cushion shocks [47], or that can be configured into many different rest states [33, 48, 49]. Metamaterials based on nonmonotonic unit cells exhibit more complex force-displacements curves, characterized by many folds leading to sequences of snapping instabilities [50] (Fig. 1.4a,b). Impact protection properties can be embedded in metamaterials based on those snapping unit cells, as they can absorb energy while keeping the force to a plateauing value, alike metals when deformed plastically [51] (Fig. 1.4b).

Sequencing of snapping instabilities is also harnessed in pneumatic systems. Upon pressurization, serially-coupled inflatables with designed nonmonotonic pressure-volume curves inflate sequentially [29] (Fig. 1.4c). This principle can be used to turn a simple input into locomotion gaits for soft robots [54], targeted deformation pathways [55] or a desired sequence of actuation [56]. In all these examples, the complex behavior is achieved through a modular design, comprising building blocks whose individual behaviors can be geometrically tuned. By controlling the peaks and valleys of the individual pressure-volume curves, the snapping sequence can be programmed [56]. As the pressure increases, inflata-

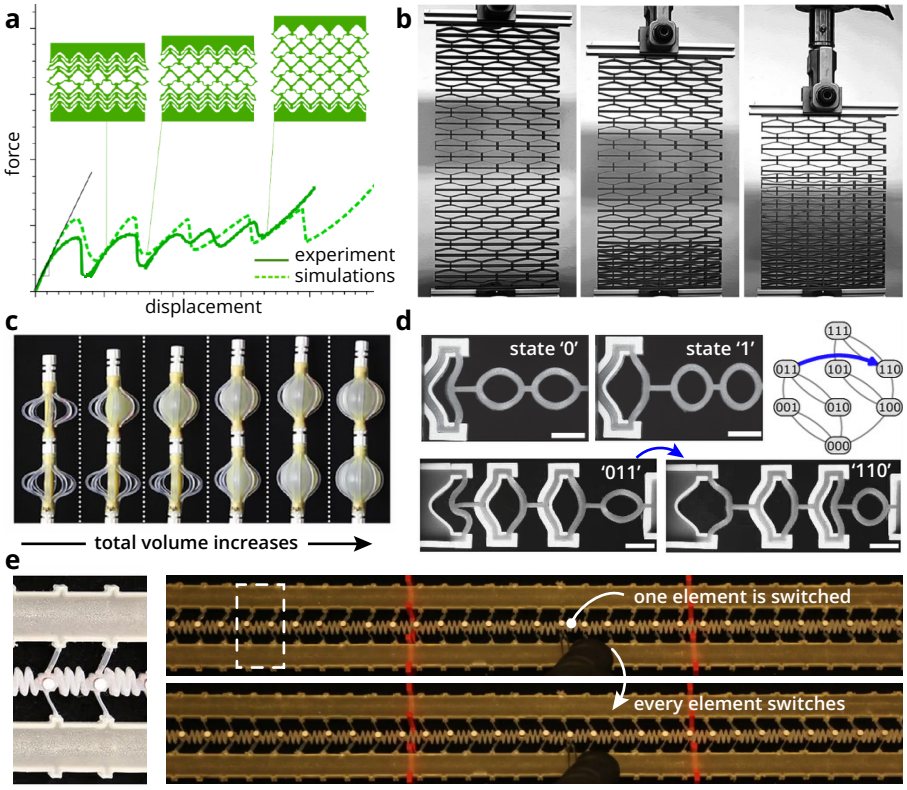


Figure 1.4: Collective complexity arising from the interplay of snapping elements. **(a)** Tensile force-displacement response of a mechanical metamaterial made from a pattern of nonmonotonic unit cells. Each peak represents a snapping event. Reproduced from Ref. [50] with permission. **(b)** Mechanical metamaterial based on nonmonotonic unit cells sequentially snapping under a compressive load, exhibiting energy absorption properties reminiscent of plastic deformation in steel. Reproduced from Ref. [51] with permission. **(c)** Serially-coupled inflatables sequentially inflating as the total volume is steadily increased. Each inflatable displays a similar nonmonotonic pressure-volume curve originating from the ballooning effect [29]. Reproduced from Ref. [29] with permission. **(d)** Avalanche occurring in a chain of three serially-coupled snapping units steadily loaded in displacement, during which the outer-most units switch branches simultaneously. The phenomenology of such a system can be described by a transition graph. Each unit is characterized by a multivalued force-displacement curve, similar to the one shown in Fig. 1.1d. Reproduced from Ref. [52], used under CC-BY license. **(e)** Nonmonotonic elements coupled by springs and a wave guide. When one element is switched, a wave propagates and makes every element switch together. Reproduced from Ref. [53], used with permission.

bles with the lowest peaks snap first; as the pressure decreases, those with the highest valleys snap back first.

Different couplings among the building blocks and loading conditions lead to more complex snapping pathways [57]. For example, serially-coupled snapping units driven in displacement can be subject to avalanches [52], where multiple building blocks can snap and snap back together (Fig. 1.4d). These systems can be viewed as finite state machines described by transition graphs, where each state represents the combination of stable branches that the individual building blocks occupy [58]. For example, if each block can be in one of two branches, labeled '0' for the initial branch and '1' for the second branch, then a system with three blocks can have states such as '000', '001', '010' , and so on. The transitions between these states correspond to snapping reconfigurations of the blocks. Systems with rich transitions provide a promising avenue to embed computation in matter, without electronics. Many blocks can switch during a single transition. For example, when snapping instabilities are coupled through a wave guide [53] or a lattice of rotating squares [59], the whole system switches during a single dynamic event, giving rise to a propagating wave, which can be used to help transmit and process information in purely mechanical ways (Fig. 1.4e).

1.4 Research objectives

In the previous sections, we introduced the role of geometric nonlinearities in enabling mechanical behaviors that cannot be achieved in the linear regime. In particular, when the nonlinearities make a force-displacement curve fold, mechanical structures break the one-to-one correspondence between load and shape, and undergo snapping instabilities that replace slow continuous deformations with fast discontinuous ones. Such systems can be viewed as programmable sets of configurations, where snapping constitutes the mechanism to switch from one to another, thereby exhibiting more functional attributes than traditional mechanical structures. The profile of the force-displacement curve is crucial as it dictates the functionalities the structure can realize. Different curves could therefore unlock new functionalities. Combining snapping unit cells in metamaterial lattices and coupling snapping modules in series has shown promise in realizing complex curves. Inspired by these observations, in this thesis, we wonder:

What kind of force-displacement curves and mechanical behaviors can we possibly achieve using assemblies of nonlinear building blocks? What functionalities could be unlocked? How can we explore and understand the interplay between individual nonlinear behaviors?

A common attribute of nonmonotonic or multivalued force-displacement curves is the fact that the second stable branch lies to the right of the initial branch. Therefore, snapping instabilities typically produce a sudden increase in output displacement under an increasing input force. But does it need to be that way [60, 61]? Can we harness the complexity arising from an assembly of nonlinear building blocks to design a structure where the second stable branch lies on the left of the initial branch? Is it physically possible? Upon loading, will the snapping instability produce a decrease in displacement? If so, what functionalities would such an exotic mechanical behavior enable? This is our first research objective.

Objective 1

Develop a countersnapping structure, that is a structure with a force-displacement curve composed of two stable branches that intersect, using a construction of nonlinear building blocks; and demonstrate the properties and functionalities enabled.

Effective modeling of nonlinear structures is crucial to guide their design and gain understanding into their mechanical behavior. Conventional nonlinear finite element analyses are often impractical for conceptual design and for studying systems with large deformations or instabilities, as they can be computationally expensive, fail under distorted meshes or mechanical instabilities, and act as black boxes that provide little insight. Oftentimes, reduced-order models are preferred, as they provide a higher-level description of a system, making them easier to define and solve. How can we build reduced-order models for assemblies of nonlinear building blocks? How can we simulate the geometric interplay between elements of various shapes, characterized by multivalued force-displacement curves and subject to different forms of loading? This is our second objective.

Objective 2

Develop a theoretical framework for the efficient modeling and simulation of assemblies constructed from building blocks with diverse shapes and intrinsic nonlinear, potentially multivalued, behaviors.

Understanding the deformation pathways from the interaction of snapping and countersnapping elements is essential in shape-morphing applications. In particular, investigating how initially warped objects can be transformed into their flat configuration upon contact with highly nonlinear pillars (i.e. geometric rectification) could help address positioning challenges posed by wafer clamping in the nanolithography industry. How can we effectively model the geometric rectification process of warped objects? How to assess how snapping and countersnapping pillars would perform? How to determine deformation pathways of highly nonlinear systems from their force-displacement curves? This is our third objective.

Objective 3

Develop tools to determine the deformation pathways of systems comprising highly nonlinear elements, and showcase them in the context of the geometric rectification of warped objects. Assess the potential of snapping and countersnapping pillars in flattening warped wafers.

1.5 Thesis contributions

Objective 1 is addressed in Chapter 2. There, we develop a structure with a force–displacement curve that self-intersects in the stable domain, constructed from five geometrically nonlinear building blocks. The design strategy relies on the development of a catalog of potential building blocks, enabling the scanning and identification of promising combinations through a hybrid experimental–computational method. The unique profile of the force–displacement curve gives rise to *countersnapping instabilities*, where the structure is able to become suddenly *shorter* upon increasing tensile force, or display a *jump* in reaction force upon increasing tensile displacement, contrary to regular snapping instabilities. We experimentally demonstrate that countersnapping enables

behaviors unattainable via regular snapping, including unidirectional stick–slip motion, programmable stiffness without altering the external state, and passive resonance avoidance. We also show that combining multiple countersnapping elements creates force–displacement curves with multiple self-intersections, which, when serially coupled, lead to a collective snapping chain reaction.

Objective 2 is addressed in Chapter 3. There, we develop a framework to build reduced-order models of highly nonlinear mechanical systems and simulate them under quasistatic conditions. We introduce the concept of "flexels", elements that intrinsically capture the complex mechanical responses of compound structures while reducing the number of degrees of freedom drastically. A flexel is constructed from a given force-displacement curve that can potentially fold, curve back and self-intersect multiple times, which is then converted into a bivariate elastic energy potential. That energy potential can then be coupled to any geometric measure computed from a set of nodes, allowing the flexels to represent a wide range of mechanical deformations, interactions, and stimuli, including stretching, bending, contact, pneumatic actuation, and cable-driven actuation. We demonstrate through many examples that assembling flexels enables the modeling and simulation of complex mechanical systems using only a small number of elements, such as pre-stressed tensegrities, tape-spring mechanisms, interacting buckled beams, and a soft gripper actuated through a metafluid. To promote usability, the framework is implemented in an open-access Python library.

Objective 3 is addressed in Chapter 4. There, we present a method to determine the deformation pathways of systems composed of highly nonlinear elements, such as snapping and countersnapping elements, and showcase it in the context of geometric rectification of initially curved beams. To understand such complex systems, we develop a method to construct transition graphs from force-displacement curves computed from the arclength method. This allows us to find that countersnapping structures can achieve levels of rectification unattainable by linear pillars of comparable stiffness, effectively mimicking much stiffer structures.

Finally, we conclude this thesis by discussing our findings and reflecting on using nonlinear building blocks as a design paradigm for increasing the complexity of mechanical systems, in the light of what is established throughout this thesis.

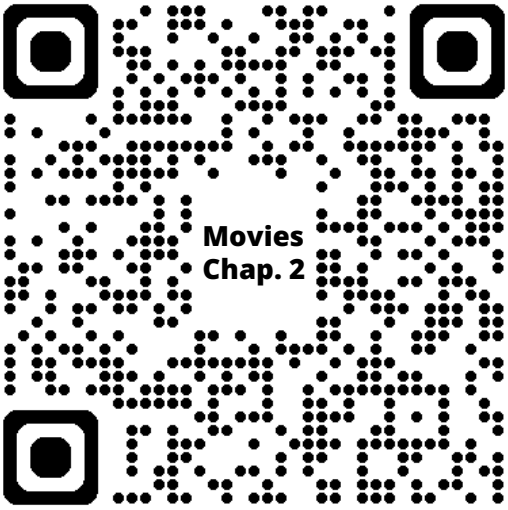
We discuss the implications of self-intersecting force-displacement curves for future research, limitations and possible extensions of the flexel framework, and strategies to broaden adoption of nonlinearities in applications.

2

Exotic mechanical properties enabled by countersnapping instabilities

Based on publication: Ducarme *et al.*, PNAS, 122(16), e2423301122 [62]

Mechanical snapping instabilities *are leveraged by natural systems, metamaterials and devices for rapid sensing, actuation and shape changes, as well as to absorb impact. In all current forms of snapping, shapes deform in the same direction as the exerted forces, even though there is no physical law that dictates this. Here, we realize countersnapping mechanical structures that respond in the opposite way. In contrast to regular snapping, countersnapping manifests itself in a sudden shortening transition under increasing tension, or a sudden increase in tensile force under increasing extension. We design these structures by combining basic flexible building blocks that leverage geometric nonlinearities. We demonstrate experimentally that countersnapping can be employed to obtain new exotic properties, such as unidirectional stick-slip motion, switchable stiffness that does not otherwise affect the state of the system, and passive resonance avoidance. Moreover, we demonstrate that combining multiple countersnapping elements allows sequential stiffness switching for elements coupled in parallel, or instantaneous collective switching for elements in series. By expanding the repertoire of realizable elastic instabilities, our work opens routes to new principles for mechanical sensing, computation and actuation.*



Link to the Movies S2.1 to S2.8

2.1 Introduction

Historically, structural stability has been a strong requirement for the design of mechanical systems. In recent years, however, a paradigm shift toward embracing and harnessing instabilities has enabled materials and structures that embody complex functionalities. Examples can be found in fields as diverse as mechanical metamaterials [5,47,63,64], signal propagation [53,59,65], shape-morphing structures [66–68], deployable structures [55], surface patterning [36, 69], mechanical computation [52, 70, 71], and soft robotics [23, 24, 26, 72–74]. Snapping instabilities, the sudden transition between two distinct configurations (Fig. 2.1a-b), have demonstrated to be a fundamental building block for these applications.

Snapping is often associated with a nonmonotonic force-displacement curve, which leads to discontinuous changes in displacement triggered by an incremental change in force (Fig. 2.1a). Additionally, a multivalued force-displacement curve can lead to discontinuous changes in force under incremental displacements (Fig. 2.1b). Based on these characteristics, snapping instabilities can be leveraged to amplify small input loads into large deformations [26, 29], to endow soft robots with fast actuation and jumping capabilities [24,28,74], to realize metamaterials with unprecedented shock-absorbing properties [47], to passively convert a steady stream of energy into functional pulsatile outputs [23,75] or to accurately morph structures into various stable shapes [33,66–68,76]. Structures built from multiple snapping elements display rich deformation pathways characterized by a succession of snapping events [57,77–80], which has been utilized to generate locomotion gaits for pneumatic soft robots [34] and deterministic deployment sequences [55, 81]. Snapping transitions and multistability have also proven to be pivotal ingredients to store and process information in mechanical platforms [52,70,71].

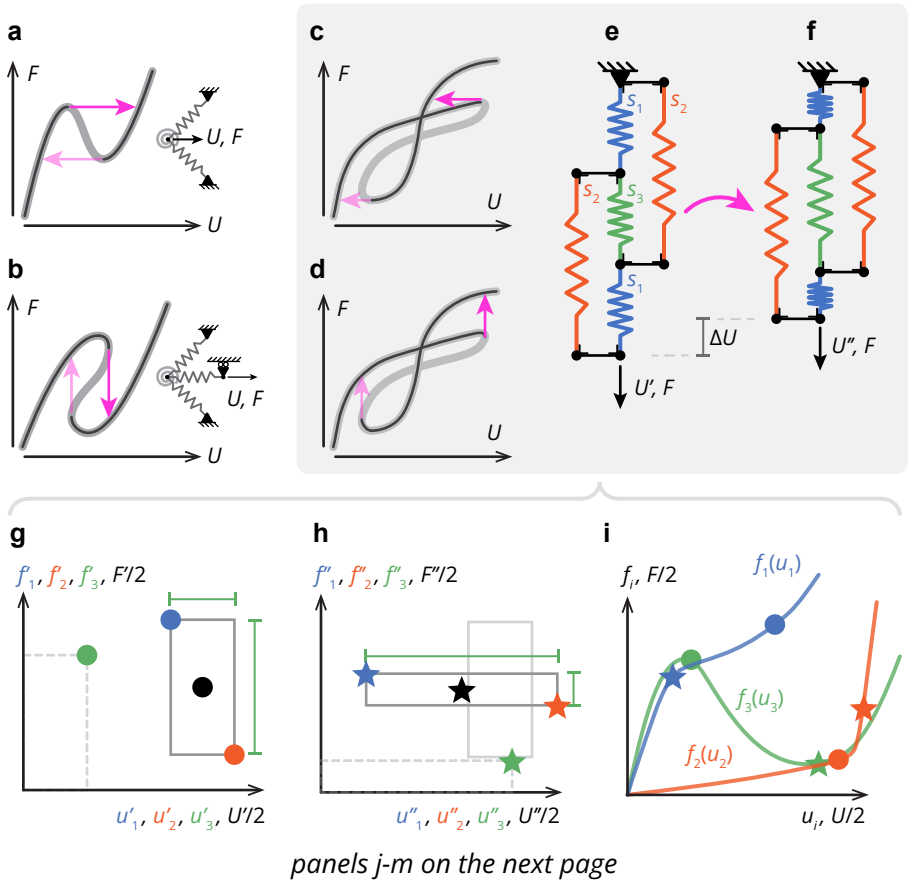
Still, all these exotic properties arise from snapping events that have one aspect in common: snapping always occurs in the same direction as the incremental load. For example, under controlled force, pulling leads to a sudden extension, and under controlled deformation, stretching leads to a sudden force drop (Fig. 2.1a-b). While this link between the direction of forces and deformations seems natural, there is no physical law that requires it. Theoretical work has shown that it should be possible to observe what we refer to as *countersnapping*, where pulling leads to a sudden contraction and incremental stretching leads to

a sudden and sharp increase in force [61]. Yet, it is an open question how to design and experimentally realize a (meta)structure that exhibits this behavior, with only limited experimental work hinting toward — but not demonstrating — the existence of such counter-intuitive snapping response [82], and no reported occurrence in natural systems. Importantly, it remains an open question what kind of mechanical exotic — yet useful — behavior can be realized by leveraging such countersnapping instabilities.

Here, we design, fabricate, and investigate structures that materialize such a countersnapping behavior. Our structure employs three different building blocks, each with a different type of nonlinear geometric behavior, and combines these in a small network that collectively realizes a *self-intersecting* force-displacement relation. Moving along such an equilibrium path by either driving the force or displacement produces countersnapping (Fig. 2.1c-d). Using this physical implementation, we show how countersnapping manifests itself under various loading conditions and experimentally demonstrate unique properties originating from the self-intersection observed in the force-displacement curve. We show unidirectional stick-slip motion under cyclic driving, programmable stiffness that does not modify the externally observed state of the structure, self-switching stiffness for resonance avoidance, and imperfection-insensitive collective snapping sequences at constant deformation.

2.2 Materializing countersnapping instabilities by coupling nonlinear mechanical building blocks

Countersnapping instabilities can be seen as a mechanical counterpart of Braess' paradox [61]. In its original formulation, this paradox describes how closing a road can improve traffic flow [83]. Similarly, one can construct a set of springs connected by strings, such that under a constant tensile load, the cutting of a taut string leads to a contraction of the whole assembly [60, 84]. The essence of this construction is multistability, where an initial pre-tensioned "serial" configuration of springs transforms into a "parallel" configuration, and where the trigger for this transition is the removal of a single link. More generally, such a behavior can occur in systems with sufficiently complex energy potentials, as was shown theoretically using a set of particles interacting through strongly nonlinear potentials, and also using a set of linear springs that exhibit geometric nonlinearities



panels j-m on the next page

Figure 2.1: Combining nonlinear mechanical building blocks to realize countersnapping instabilities. Regular snapping instabilities observed in a nonmonotonic force (F) - displacement (U) relation realized by a pair of springs forming a left buckled beam and driven from the connection point **(a)**, and a multivalued force-displacement relation achieved by serially coupling a nonmonotonic element to a spring **(b)**. **(c-d)** Self-intersecting force-displacement relation that leads to countersnapping: a sudden decrease in U when F is increased (c), or a sudden increase in F when U is increased (d). **(e-f)** Realization of countersnapping elements with a network of three different weakly nonlinear springs ($s_1 - s_3$). **(g-i)** Geometric construction of the force-displacement curves needed to achieve the countersnapping behavior. The color of the markers refer to the type of spring. Black markers refers to the global state. In (g) and (h), the dark gray rectangles represent the state before (g) and after (h) switching. The width and height of each rectangle are used to create the green marker which represents the state of spring s_3. In (i), markers from both (g) and (h) are plotted together, which allows to draw the three local force-displacement curves. **(continuing on the next page)**

panels a-i on the previous page

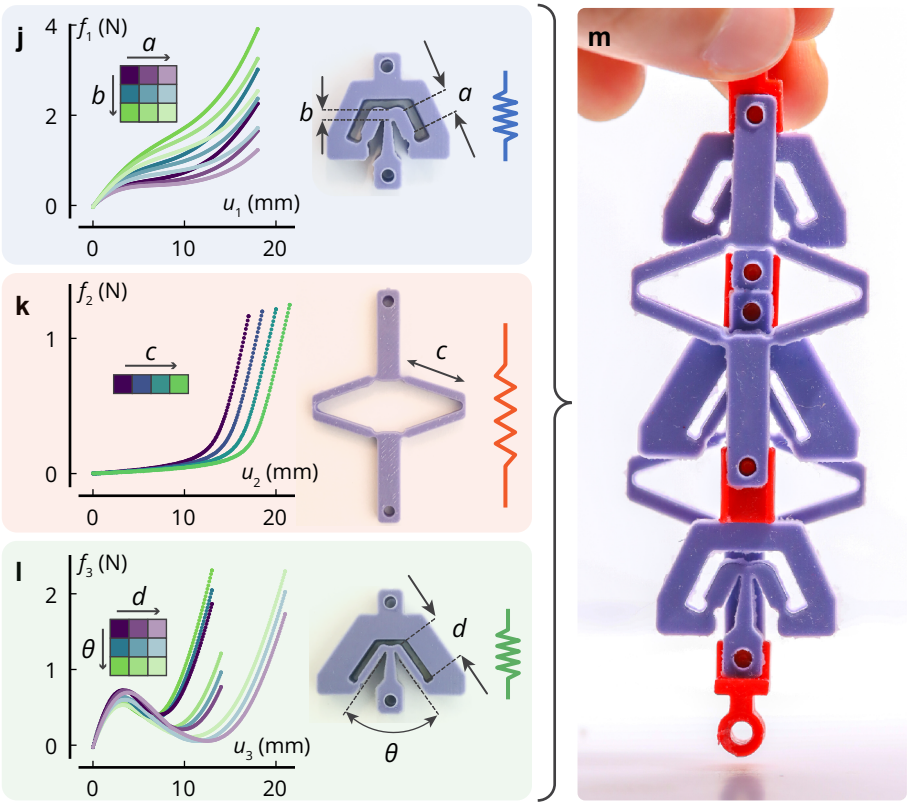


Figure 2.1 (continued): (j-l) Experimentally evaluated nonlinear building blocks for a range of parameters. (j) Softening building block with $a = [4, 5, 6]$ mm, $b = [1.0, 1.5, 2.0]$ mm. (k) Stiffening building block with $c = [12, 13, 14, 15]$ mm. (l) Non-monotonic building block with $d = [5, 7, 9]$ mm, $\theta = [50°, 60°, 70°]$. **(m)** Assembly of the building blocks that lead to countersnapping.

under tension, originating from changes in orientation [61]. However, the former design requires steep potentials (e.g. Lennard-Jones potential), whereas the latter design requires 21 perfectly hinging springs with required stiffnesses that vary over six orders of magnitude. Experimentally, overcoming these design requirements has proven prohibitively hard, with a recent attempt only hinting toward the existence of countersnapping as the experiments were dominated by viscoelastic behavior [82]. Hence, while inspiring, it remains an open question how to robustly realize countersnapping instabilities experimentally.

2.2.1 Construction of local force-displacement curves

To realize reversible countersnapping behavior, we focus on tensile deformation and modify the spring-network geometry that exhibits the mechanical pendant of Braess' paradox [61]. Note that this spring network is not proven to be the simplest countersnapping mechanical system, yet it constitutes a useful starting point for the rational design that we are presenting here.

We assume that its five elastic springs are nonlinear and of three distinct types s_1, s_2 and s_3. Through symmetry, springs of type s_1 and s_2 occur twice in the network, yet we assume that they are in the same state (Fig. 2.1e-f). At any point during the loading process, a spring s_i has a certain elongation noted u_i and a certain tensile force noted f_i.

Our design target is to construct local force-displacement $f_i(u_i)$ curves for $s_1 - s_3$, such that, for a given global force F, the network has two stable configurations with global extensions U' and U''. Essential is that the system switches from U' to U'' at an incremental increase in force, with $U'' < U'$. To construct the local curves $f_i(u_i)$, we develop a method that relies on a geometric interpretation of the constraints and force balance equations that govern the network shown in Fig. 2.1e. From geometry and force balance we indeed find that the global displacement is the sum of the extensions of springs s_1 and s_2 ($U = u_1 + u_2$), and the global force is the sum of the tensions in springs s_1 and s_2 ($F = f_1 + f_2$). Additionally, the extension of spring s_3 is the difference in extension between springs s_2 and s_1 ($u_3 = u_2 - u_1$), and its tension is the difference in tension between springs s_1 ad s_2 ($f_3 = f_1 - f_2$). By using these relations, the state of the structure can be geometrically represented by drawing a rectangle in a force-displacement plane (Fig. 2.1g-h). The top-left corner of the rectangle coordinates represent the

stretch u_1 and tension f_1 in spring s_1, while the bottom-right corner coordinates represent those in s_2. Since $u_3 = u_2 - u_1$ and $f_3 = f_1 - f_2$, the width and height of that rectangle are the stretch u_3 and tension f_3 in spring s_3. Moreover, since $U/2 = (u_1+u_2)/2$ and $F/2 = (f_1+f_2)/2$, the coordinates of the rectangle's center point represent half the global displacement $U/2$ and half the global force $F/2$.

In Fig. 2.1g, a first rectangle is drawn to represent the state before switching (Fig. 2.1e), while in Fig. 2.1h, a second rectangle is drawn to represent the state just after switching (Fig. 2.1f). Crucially, the center of the second rectangle must be shifted horizontally to the left compared to the center of the first rectangle (as $U'' < U'$ and the force F is conserved). Next, in Fig. 2.1i, we plot together the top and bottom corners of both rectangles, along with two extra markers whose coordinates are the width and height of both rectangles (green dot, green star). Finally, for each of the three springs, a force-displacement curve is drawn through the origin and corresponding markers. To ensure that the first state becomes unstable for an infinitesimal increase in F, the curve $f_3(u_3)$ must meet additional conditions detailed in Chapter S2 (Section S2.2). To maintain monotonic force-displacement curves for s_1 and s_2, we can retrospectively notice that the second rectangle (Fig. 2.1h) must be wider and flatter than the first one, indicating that s_3 has a higher stretch and a lower tension after the switch, similar to the cut spring in the Braess' paradox analogue [60, 84].

Most importantly, this geometric construction reveals that the force-displacement response of s_1 needs to be softening, s_2 stiffening, and s_3 nonmonotonic (Fig. 2.1i). Note that the same geometric construction can also be applied to identify the individual curves leading to countersnapping during *unloading*, that is a sudden increase in elongation as the tensile force is slowly reduced (Chapter S2, Section S2.3, Fig. S2.4). A key insight unveiled by this approach is that a countersnapping instability is possible by using only three different *weakly* nonlinear potentials, which can be materialized using relatively basic elastic building blocks as we will demonstrate next.

2.2.2 Materializing nonlinear building blocks

Our approach to realize the countersnapping behavior in experiments requires designing and fabricating several monolithic and elastic structures, each materializing one of the three types of nonlinear springs. These monolithic structures

will then be assembled together in the arrangement shown in Fig. 2.1e. However, designing manufacturable structures with exact force-extension curves is a difficult inverse design problem [85, 86]. Instead, we focus on building an experimental library of building blocks that demonstrate the general profile of the desired softening, stiffening and nonmonotonic force-extension behavior, and then evaluating assemblies of these nonlinear springs numerically. We focus on using geometric nonlinearities, but other approaches where material nonlinearities or contact are harnessed could potentially be used. For both the softening and nonmonotonic building block designs (Fig. 2.1j,l), we use a v-shaped pair of inclined beams, wedged within a relatively stiffer structure [33, 66]. The stiffening building block design (Fig. 2.1k) is based on a diamond-shaped set of beams connected by flexures that, upon tension, transition from a soft-and-bending to a stiff-and-stretching mode of deformation (Chapter S2, Section S2.1.1, Fig. S2.1a-c for detailed drawings).

Next, we fabricated several centimeter-scale samples of the three different building blocks, each spanning a parameter range (Fig. 2.1j-l). The structures are fabricated by casting a silicone (Smooth-Sil 945, Smooth-On) in a 3d-printed mold (PolyJet Eden260VS, Stratasys) (Chapter S2, Section S2.1.1, Fig. S2.1). We then measured the force-extension response (Instron model 5965, 100-N load cell) of all the structures individually. In total, our catalog of building blocks with different nonlinear behaviors consists of nine units for the softening, four units for the stiffening and nine units for the nonmonotonic behavior. We then developed a numerical algorithm to simulate the behavior of assemblies composed of these building blocks, and assess whether they display a countersnapping instability (Chapter S2, Section S2.1.1). By running all possible combinations in our catalog (324 simulations), we found eight potentially countersnapping assemblies (Fig. S2.3e). This low percentage suggests that the countersnapping effect is relatively sensitive to the geometric parameters. A more detailed study of the influence of the geometric parameters reveals that the countersnapping effect exists in a relatively narrow yet feasible window of geometric parameters (Chapter S2, Section S2.4, Fig. S2.5).

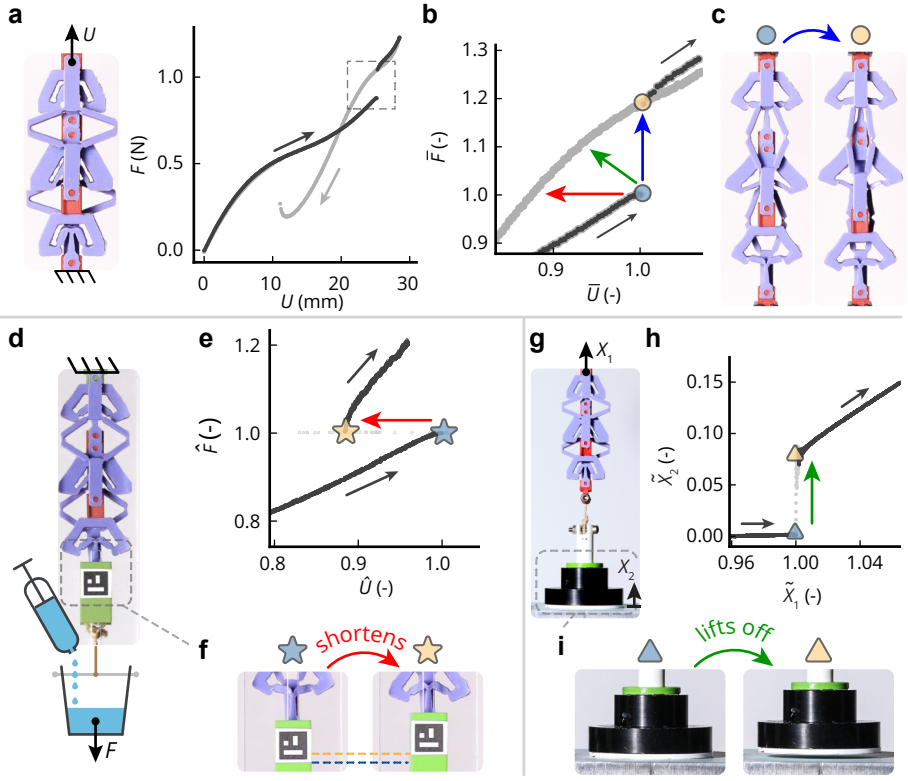


Figure 2.2: Experimental observation of countersnapping behavior. **(a)** Force-displacement curve for a countersnapping assembly obtained by increasing (dark gray) then decreasing (light gray) the applied displacement U and measuring the reaction force F. **(b)** Close-up on the critical point indicated in (a). The axes have been normalized $\bar{U} = U/U_{\mathrm{c}}$, $\bar{F} = F/F_{\mathrm{c}}$, where the subscript c indicates the critical point under controlled displacement. $U_{\mathrm{c}'} = 25.2$ mm, $F_{\mathrm{c}'} = 0.88$ N. Colored arrows indicate the jumps under controlled displacement (blue), controlled force (red), and mixed conditions (green). **(c)** Snapshots of the configuration just before (left) and just after (right) snapping during loading (displacement-controlled). **(d)** Set-up for the force-controlled experiment: the structure is attached from the top and a cup, suspended at the bottom, is slowly filled with water. **(e)** Force-displacement curve obtained by increasing the applied force F and measuring the displacement U. The axes have been normalized $\hat{U} = U/U_{\mathrm{c}'}$, $\hat{F} = F/F_{\mathrm{c}'}$, where the subscript c' indicates the critical point under controlled force. $U_{\mathrm{c}'} = 23.9$ mm, $F_{\mathrm{c}'} = 0.81$ N. **(f)** Snapshots just before (left) and just after (right) snapping. **(g)** Set-up for loading under mixed conditions, measuring the elevation X_2 of the weight. The structure is pulled upwards from the top by an increasing displacement X_1 while the weight attached to the bottom is initially sitting on a flat platform. **(h)** Weight elevation as a function of the applied displacement. The axes have been normalized $\tilde{X}_1 = X_1/X_{1\mathrm{c}''}$, $\tilde{X}_2 = X_2/X_{1\mathrm{c}''}$, where the subscript c'' indicates the critical point under mixed conditions. $X_{1\mathrm{c}''} = 23.5$ mm. **(i)** Snapshots just before (left) and just after (right) the sudden liftoff. Notice the higher weight elevation in the right snapshot.

2.2.3 Experimental observation of a countersnapping instability

To observe the countersnapping behavior experimentally, we physically assemble the five building blocks of one of the eight potential assemblies (Fig. 2.1m, Fig. S2.3f, Fig. S2.2), and perform three mechanical tests (Fig. 2.2). First, we characterize the force-extension behavior of the nonlinear spring network under controlled extension using a tensile tester. We observe that when the extension U reaches a critical value, the element snaps and the reaction force suddenly jumps to a higher value — a first and clear signature of countersnapping behavior (Fig. 2.2a-c, Movie S2.1). Note that during unloading we observe an ordinary snapping event, where this asymmetry is the result of the intersection in the force-displacement behavior. Based on energy considerations, we argue that an elastic system that would countersnap upon both loading and unloading should be possible but would necessitate a more complex force-displacement curve with two points where stable branches intersect (Chapter S2, Section S2.5, Fig. S2.6).

Second, we study countersnapping under constant load. To do so, we suspend the structure and load it by slowly filling a cup attached to one side of the structure with water, while measuring the force and extension using a load cell and video tracking. Once the critical tensile load is reached, we observe that the structure suddenly reduces its elongation by about 12%, leading to an upward jump of the suspended cup — a second hallmark of countersnapping (Fig. 2.2d-f, Movie S2.2).

Third, we demonstrate that countersnapping also can lead to a simultaneous discontinuous contraction and force increase, by slowly lifting an object that initially is resting on a surface (Fig. 2.2g). For an appropriate weight that lies between the forces observed before and after snapping in the controlled displacement experiment, increasing the lifting force triggers the countersnapping instability and leads to a sudden lifting of the suspended weight (Fig. 2.2g-i, Movie S2.3). Hence, depending on the loading conditions, the countersnapping spring network exhibits a combination of a sudden increase in force and contraction (Fig. 2.2b), which originates from the switch from a predominantly series to a predominantly parallel distribution of the load (Chapter S2, Section S2.6, Fig. S2.7).

2.3 Exotic mechanical properties

Having demonstrated experimentally the existence of the countersnapping instability, we next explore the functionalities of a single countersnapping element. First, we harness its unusual combination of snapping directions under cyclic loading; then we exploit the functionalities originating from the self-intersection of the force-displacement curve.

2.3.1 Unidirectional stick-slip actuation

Stick-slip actuation is an important mechanism for precision manipulation and positioning applications [87, 88]. It relies on a combination of an external actuator and a frictionally coupled slider, so that alternating slow and fast actuation produces sticking and incremental slipping between slider and actuator. Snapping could in principle be used to generate the fast phase of actuation, reducing the need for intricate control systems. However, as ordinary snapping occurs in the direction of the applied force (Fig. 2.1a), cyclic actuation drives snapping events in opposite directions with little net effect. By contrast, our countersnapping element snaps in the same direction during loading and unloading, resulting in two successive contractions per cycle (Fig. 2.1c), which enables unidirectional stick-slip actuation under cyclic loading.

To demonstrate this, we study the behavior of elastic structures in the following set-up. We horizontally suspend the structure, clamp its right side, and rest its left side on a 5cm-long block of foam placed on a rigid surface. We cyclically load its left side through a much softer elastic rubber band attached to a robotic arm. We compare the resulting motion of the foam block for an ordinary snapping spring (Fig. 2.3a) and our countersnapping structure (Fig. 2.3b). Note that the friction forces between the left side, foam and surface are such that under smooth motion of the total structure, the foam block smoothly slides over the surface, and there is no sliding between the foam block and the elastic structure. However, when snapping instabilities are triggered, the structure and the foam exhibit relative sliding, as the inertial force caused by the large acceleration exceeds the static friction limit. As expected, for ordinary snapping, these events are of opposite sign, such that the total motion of the foam after one cycle is negligible (Fig. 2.3c, Movie S2.4). Instead, for our countersnapping element, the

sliding motion is in the same direction during snapping and unsnapping, leading to a unidirectional motion of the foam block (Fig. 2.3d-e, Movie S2.4). Hence, cyclic loading of a countersnapping element can be leveraged for unidirectional, incremental stick-slip motion, with potential applications in sensors that count numbers of loading cycles [89], stick-slip actuators [88] and soft robotic locomotion.

2.3.2 Programmable stiffness

Our countersnapping element features an unusual point where the force-displacement curve self-intersects (Fig. 2.4a). Corresponding to this point are two distinct configurations characterized by the same extension and the same tensile force, yet by different stiffnesses (Fig. 2.4b). By measuring the slope of each branch at the intersection in the force-displacement plane, we characterized the stiffness of state 0 to be 0.03 N/mm (soft state), and of state 1 to be 0.09 N/mm (stiff state).

We now demonstrate that switching from one stiffness to the other can be achieved by means of snapping transitions, and that this can be used to manipulate the oscillation frequency of a suspended mass, without affecting the elongation at equilibrium (Fig. 2.4c)).

First, we load the countersnapping structure to the intersection point by suspending a cup of water, which applies a tensile dead load of around 0.7 N, resulting in an elongation of around 20 mm (Fig. 2.4d). To characterize the natural frequency, we excite the structure with a slight pull and release, while making sure that our element remains in its serial configuration (Fig. 2.4b-left), and observe that the mass' frequency equals 3.7 Hz (Fig. 2.4e-top, Movie S2.5). Once the oscillations have damped out, we again pull on the suspended weight, but this time, up to the point where the structure snaps and switches to its parallel configuration (Fig. 2.4b-right). By gently releasing the mass, the structure reaches the same equilibrium elongation, yet remains in the parallel configuration. When we now excite the structure by slightly pulling and releasing the mass, we observe a significantly larger frequency of 6.4 Hz (Fig. 2.4e-bottom, Movie S2.5). This illustrates that countersnapping structures can be used to program the natural frequency of a mass-spring system — without changing the equilibrium load or extension.

This programmable stiffness allows the realization of passive self-switching of

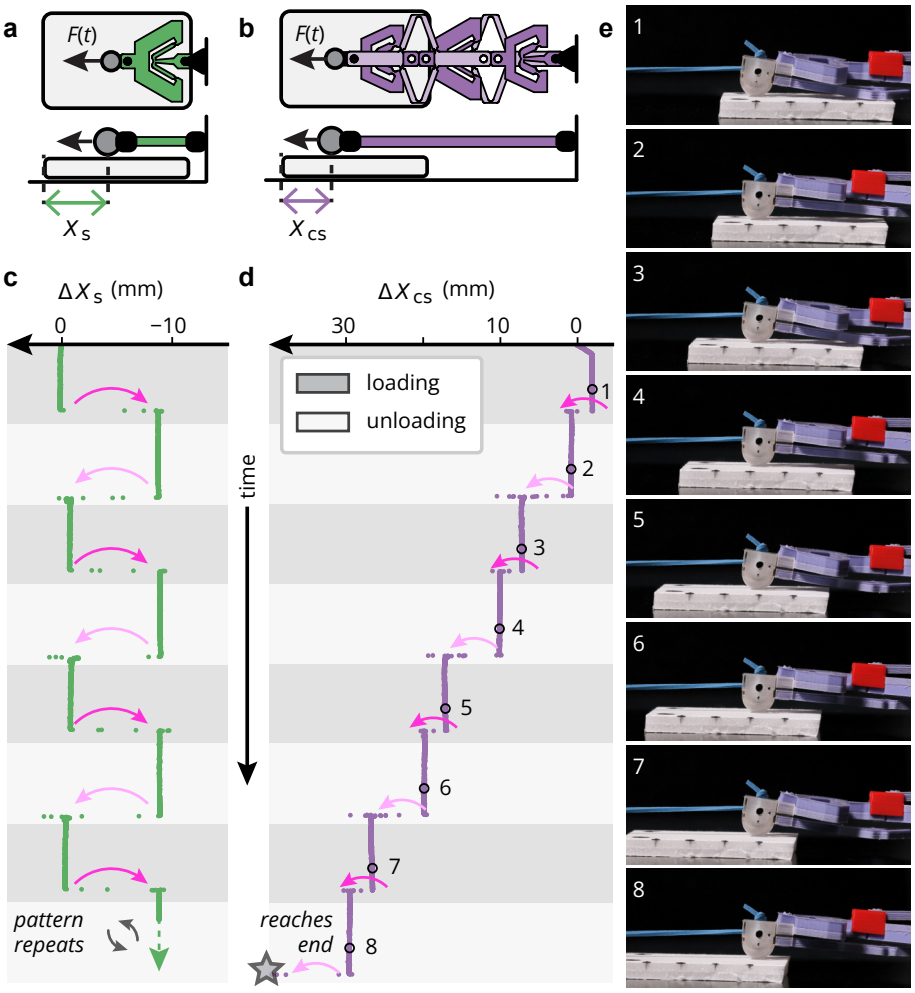
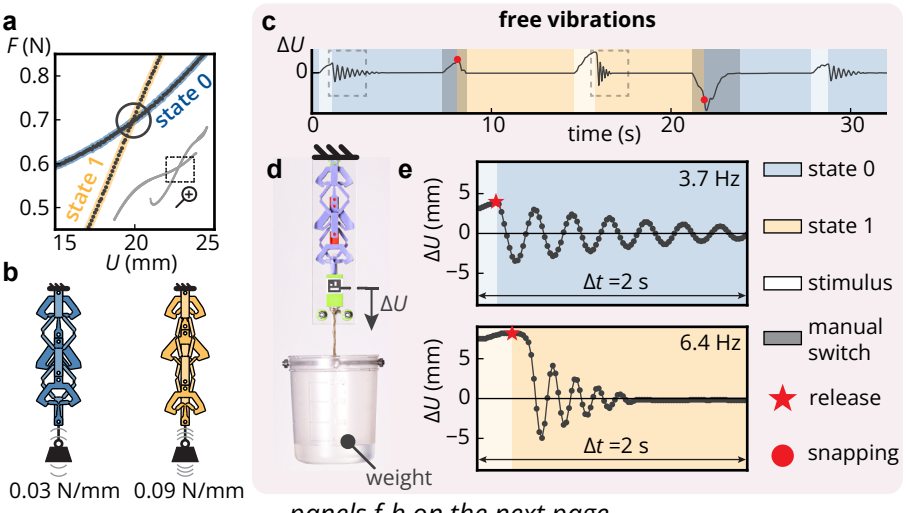


Figure 2.3: Experimental observation of unidirectional stick-slip behavior. **(a-b)** Sliding experiment set-up used for the snapping (a) and countersnapping (b) structures. The block of foam is depicted by the gray rounded rectangles. **(c-d)** Change in distance between the contact point and the piece of foam over time, for the (c) snapping and (d) countersnapping structures. The gray star indicates the instant at which the countersnapping structure completely slides off the piece of foam. Each loading and unloading sequence lasts 9.38 s (± 1.9%) in (c) and 16.87 s (± 0.4%) in (d). **(e)** Snapshots of the sliding experiment using the countersnapping structure (side view) taken at fixed time intervals.



panels f-h on the next page

Figure 2.4: Experiments to demonstrate programmable and self-switching stiffness. **(a)** Close-up on the self-intersection of the force-displacement curve from Fig. 2.2a. **(b)** The two states, soft (0, blue) and stiff (1, yellow) that correspond to the intersection point in (a) have the same elongation under the same applied force, yet different stiffesses. **(c)** Change in elongation ΔU over time during three episode of free oscillations. Each episode is triggered by a stimulus (slight pull and release) indicated by the whitened backgrounds. Between each episode, the state is manually switched by pulling or pushing on the weight, indicated by the darkened background. The red circles indicate the instant at which the state changes. The amplitude ΔU required to switch from 0 to 1 (resp. 1 to 0) is 5.0 mm (resp. -11.5 mm). **(d)** Experimental set-up to study the free vibrations. **(e)** Close-up on the first two episodes of free oscillations highlighted by the boxes in (c). Each episode is triggered by a stimulus consisting of releasing the weight from a slightly stretched configuration compared to the intersection state, indicated by the red stars. The amplitude of that stretch is respectively 3.9 mm (top) and 8.2 mm (bottom). After releasing the weight when in state 0 (resp. state 1), the system oscillates freely at a frequency of 3.7 Hz (resp. 6.4 Hz). **(continuing on the next page)**

panels a-e on the previous page

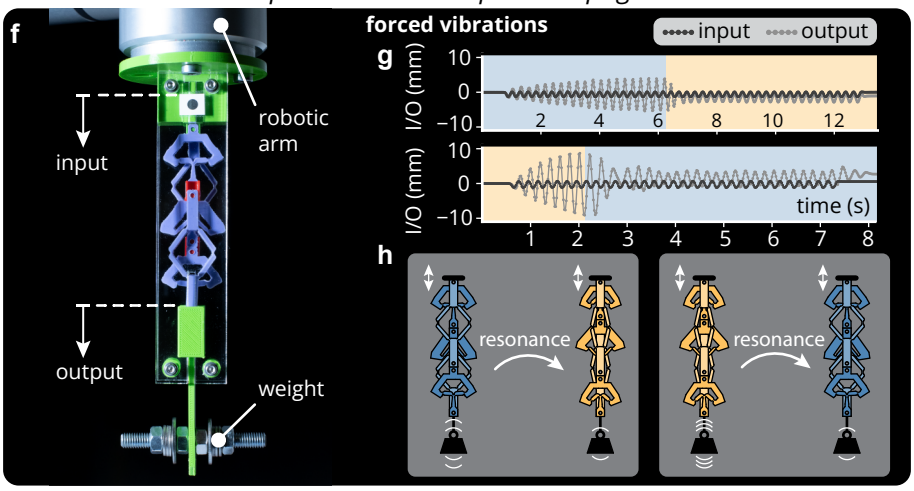


Figure 2.4 (continued): (f) Experimental set-up to study the forced vibrations. The sample is shaken vertically from the top using a robotic arm (input), while the vertical position of the suspended weight is measured (output). **(g)** Evolution of the input and output displacements during forced vibrations, leading to resonance and a switch from soft to stiff ($0 \rightarrow 1$, top) and stiff to soft ($1 \rightarrow 0$, bottom). **(h)** Schematic representation of the resonance switches for the soft-to-stiff ($0 \rightarrow 1$, left) and stiff-to-soft ($1 \rightarrow 0$, bottom) transitions.

natural frequency for an externally-driven mass-spring system. To demonstrate this, we suspend the countersnapping element from a robotic arm that allows vertical excitation at a specified frequency, and attach a mass that stretches the countersnapping structure to its intersection point (Fig. 2.4f). Starting from the serial state, we vertically drive the robotic arm close to the resonance frequency. The effective vibrational live load results in oscillations that eventually trigger the countersnapping instability, causing the element to switch to its stiffer parallel state, which subsequently reduces the amplitude of the oscillations (Fig. 2.4g-top, h-left, Movie S2.6). Similarly, when the countersnapping element is initially in the parallel state and we set the driving frequency close to the corresponding natural frequency, the system eventually switches to the softer serial state, again leading to diminishing of the vibrations (Fig. 2.4g-bottom, h-right, Movie S2.6). Passively switching the natural frequency has minimum impact on the equilibrium deformation under the dead load, which stands in stark contrast with regular snapping structures actuated by resonance [90]. Hence, countersnapping can be used to passively protect a system from specific vibrational resonances.

2.4 Collective behavior of countersnapping metamaterials

We next show that mechanical metamaterials consisting of interacting countersnapping elements can switch between multiple internal configurations and stiffnesses. To explore the collective properties of such assemblies, we consider the two simplest configurations: parallel and serially coupled pairs of nearly identical countersnapping elements (Fig. 2.5). Each of the units can be in two configurations that we refer to as '0' (initial branch) and '1' (snapped branch), leading to collective configurations such as $\{00\}, \{10\}, \ldots$ (Fig. 2.5b, i) as observed in hysteron metamaterials [57, 80]. To describe the transitions that can occur in our countersnapping metamaterials, we denote the critical extensions of countersnapping element j as u_j^+ for the $0 \longrightarrow 1$ transition, and u_j^- for the $1 \longrightarrow 0$ transition.

We first consider the mechanical response of two countersnapping elements, A and B, placed in parallel, which effectively limits the interactions between both elements as their extension is directly controlled (Fig. 2.5a). Note that even though both elements show similar behavior, imperfections during manufacturing cause the value of u_A^+ to be about 0.7 mm *lower* than u_B^+, and u_A^- is about 0.6 mm *lower* than u_B^-. Hence, when the extension is increased or decreased, the two units have

a definite switching order. To demonstrate this, we design a specific driving protocol (Fig. 2.5c) that allows to access three collective states and all their respective transitions (Fig. 2.5d-e, Movie S2.7). We note that the number of states that are accessible through our driving protocol with N countersnapping elements ranges from $N+1$ to 2^N, depending on the relative order of the critical extensions for the $0 \longrightarrow 1$ and $1 \longrightarrow 0$ transitions of the individual snapping elements [91]. For example, a design with $u_{\mathrm{A}}^{+} < u_{\mathrm{B}}^{+}$ and $u_{\mathrm{A}}^{-} > u_{\mathrm{B}}^{-}$ would allow to access all four states [57, 92]. For nearly identical countersnapping elements, the branches of the three configurations that are accessible, $\{00\}$, $\{10\}$ and $\{11\}$, intersect at a single point in the force-extension plane (Fig. 2.5d,f-g). The stiffness at this point only depends on the number of elements in state '0', and for assemblies of $N = 2$ and $N = 3$ elements (Fig. S2.8), we show that all $(N+1)$ possible stiffnesses are accessible. Hence, parallel assemblies of countersnapping elements allow flexibility in the ability to program the stiffness without influencing the observed external force and displacement.

When two elements are coupled in series, they experience global interactions as the snapping of one unit influences all other units [52, 93]. Different from ordinary snapping elements, serially coupled countersnapping elements experience an avalanche transition, directly switching from the $\{00\}$ to the $\{11\}$ during extension. Specifically, once one element switches phase $0 \longrightarrow 1$, the concomitant increase in force triggers the instability of the other element in phase '0' (Fig. 2.6a, Movie S2.8). By contrast, for serially coupled ordinary snapping structures the first snapping event results in a force drop, which stabilizes the other elements in phase '0' [52]. We observed this simultaneous snapping also for three countersnapping elements placed in series (Fig. S2.9), and believe that this phenomenon persists in longer chains as long as the level of imperfections is limited [93]. When the load is sufficiently large, such an avalanche can also be triggered by manipulating one of the countersnapping elements so that it switches from phase '0' to '1' (Fig. 2.6b, Movie S2.8). Upon unloading, we observe the elements to switch back to phase '0' in steps, instead of collectively (Fig. 2.5k,l, Fig. S2.9d,e, Movie S2.7). This asymmetric response extends the range of possibilities for non-reciprocal devices, e.g., unidirectional stick-slip actuators. We note that, similarly to parallel assemblies, N serially connected countersnapping elements allow for $N+1$ collective stiffnesses (Fig. 2.5k,m-n, Fig. S2.9 , Movie S2.7).

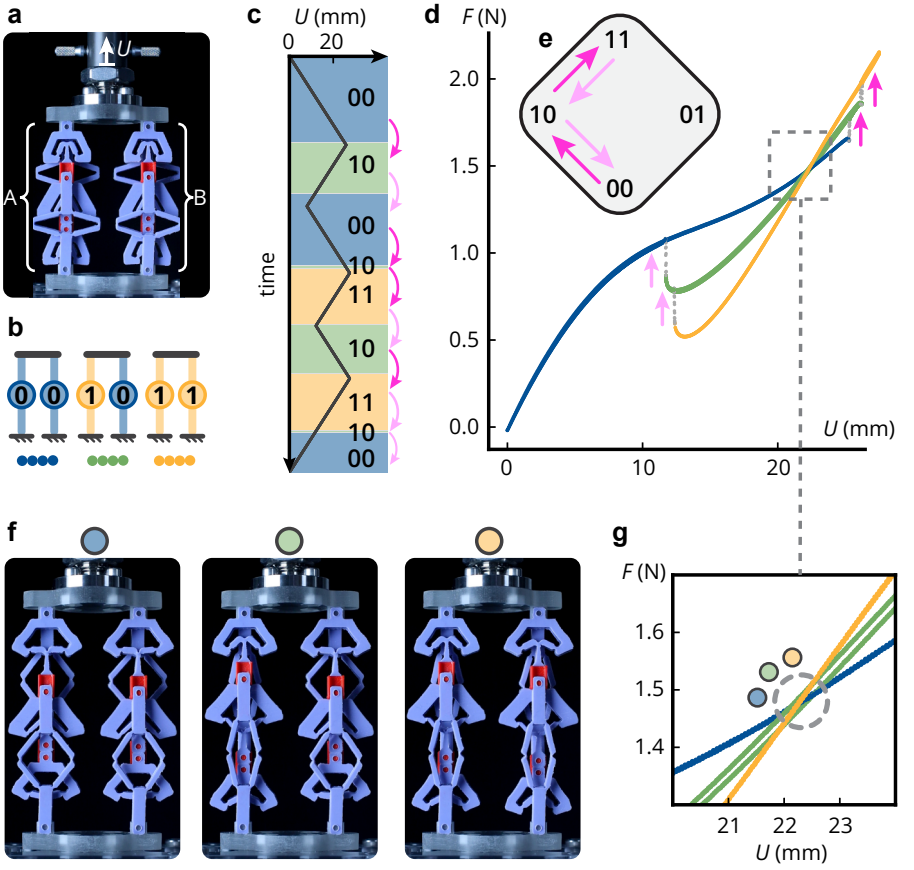
These observations suggest that the self-intersection of the force-displacement curve is preserved in meta-structures formed from countersnapping elements. For any assembly of countersnapping elements where one element is loaded to its intersection point, switching that element to the other state will affect the stiffness but not the external state of the assembly since the tension and elongation in that element is unchanged. These new phenomena therefore highlight that countersnapping opens up a new toolbox for mechanical metamaterials.

2.5 Conclusion and outlook

We demonstrated that mechanical countersnapping instabilities can be realized experimentally, and that their unique self-intersecting force-displacement behavior allows new forms of mechanical intelligence, with potential applications in soft robots or computing metamaterials. Key to achieving countersnapping was a modular approach, where we translated the design of a highly nonlinear, self-intersecting force-displacement response to the design of a few weakly nonlinear building blocks which are then assembled to realize the targeted behavior. Combining multiple countersnapping units then leads to hierarchical designs with controllable snapping sequences and rapid collective snapping events, revealing another layer of complexity. As inversely designing matter with arbitrarily equilibrium curves has remained a challenging problem, we suggest that embracing this hierarchical paradigm opens up a route to achieving other kinds of complex, yet useful and targeted, responses.

Our work suggests that studying and leveraging the dynamic behaviors of snapping structures could lead to a more versatile level of programmability. Extending our observations of resonance avoidance through self-switching to more complex multistate assemblies could allow rapid switching between many more configurations and a targeted response to vibrations. Additionally, dynamical effects could be explored to promote or modify the rapid domain-wall propagation that occurs when countersnapping instabilities are coupled, leading to rich switching properties with potential applications in signal propagation and sensing [53, 94].

We close by listing key challenges. First, even though we were able to demonstrate countersnapping reliably, its implementation in larger assemblies and applications will likely require more compact designs with improved robustness.



panels h-n on the next page

Figure 2.5: Experimentally observed collective behavior of countersnapping metamaterials. **(a)** Two parallel countersnapping elements controlled by displacement U. **(b)** Possible collective states of the system. **(c)** The binary state of the system (background color) as a function of U, where bright (light) pink arrows indicate snapping events during loading (unloading). **(d-e)** Corresponding force-displacement curve (colors indicate states) and transition graph. **(f)** Snapshots of the $\{00\}$, $\{10\}$ and $\{11\}$ states at $U = 22.3$ mm near the intersection point of the force-displacement curves. **(g)** Close-up on the intersection point. **(continuing on the next page)**

panels a-g on the previous page

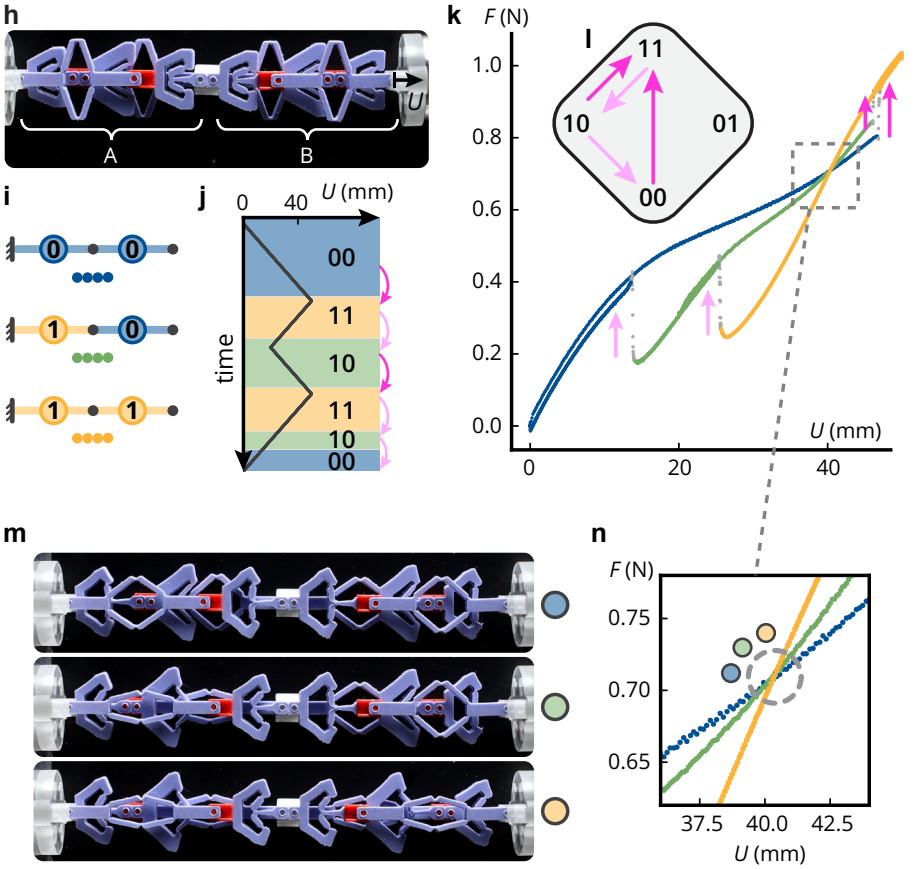


Figure 2.5 (continued): **(h)** Two serially coupled countersnapping elements. **(i)** Possible collective states of the system. **(j)** State as a function of U; note the avalanche transition $\{00\} \rightarrow \{11\}$ where both countersnapping elements flip simultaneously. **(k-l)** Corresponding force-displacement curve and transition graph. **(m)** Snapshots of the $\{00\}$, $\{10\}$ and $\{11\}$ states at $U = 40.3$ mm near the intersection point. **(n)** Close-up on the intersection point.

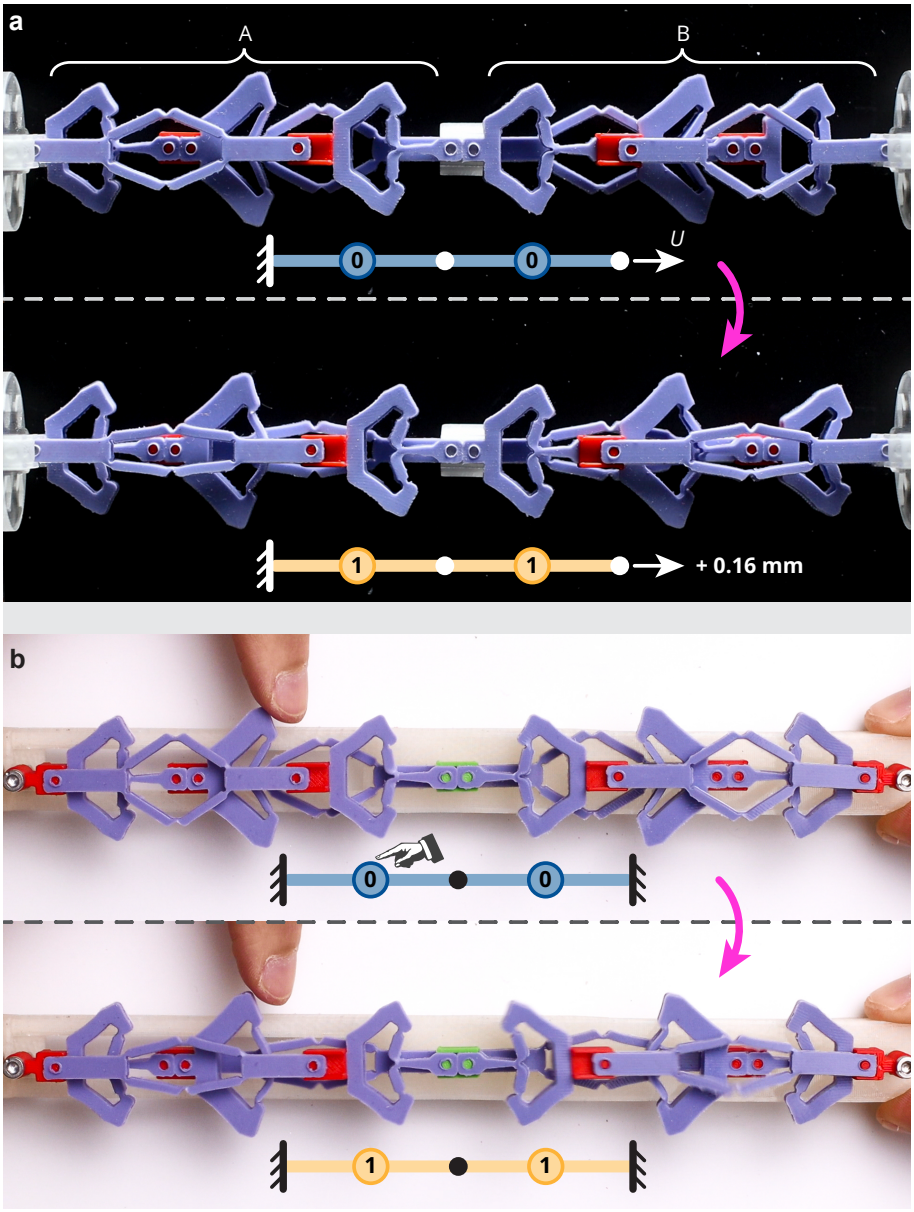


Figure 2.6: Avalanche transitions in serially coupled countersnapping elements. **(a)** Snapshots of two serially-coupled countersnapping elements, steadily stretched under displacement control, before and after the sudden avalanche where both elements switch. **(b)** Snapshots of two countersnapping elements forming a chain whose total length is constant. Poking the left element triggers a transition in both elements. Note that, despite the pair of countersnapping elements being oriented differently in panels (a) and (b), both arrangements correspond to the same (serial) coupling.

Second, further studies may enable countersnapping in other physical systems, such as fluidics and electronics, by engineering self-intersecting equilibrium curves associated with other pairs of conjugated variables such as pressure-volume, pressure-flow or current-voltage. Third, the use of active materials could enable alternative ways to induce countersnapping, e.g. through heat [95], light [96], wetting [23] or magnetism [97], or enable reprogrammable responses that allow switching between counter- and regular snapping. We hope that our mechanical implementation of countersnapping is just the beginning and believe that, in time, countersnapping will achieve an impact comparable to that of conventional snapping instabilities.

S2

Supplementary information for Chapter 2

S2.1 Materials and methods

S2.1.1 Manufacturing

The building blocks were manufactured by injecting a silicone elastomer prepolymer mixture (Smooth-Sil 945, Smooth-On) into a mold. Each mold is composed of three parts that were screwed tight together, thereby encasing a hollow cavity with the desired shape. Two metallic pins were inserted into the mold assembly to create the connecting features of the building block. Each mold was designed with one inlet and one outlet, allowing the prepolymers to be injected into the mold while letting air escape. After curing, the mold was disassembled carefully, which left the sample wedged in the central mold part. The sample was separated from the silicone cured in the inlet and outlet using a narrow blade, before using a stiff PLA block (with the same outline as the building block) to push and dislodge the sample out of the mold gently. More about the building block dimensions, mold design and fabrication process is shown in Fig. S2.1.

Mold fabrication and preparation The mold parts were 3d-printed using photocuring resin VeroClear for the model and SUP705 for support, using a PolyJet Eden260VS printer (Stratasys). After printing, the molds parts were placed in a KOH solution (5%) for 48 hours to dissolve the support material, then cleaned fully using a high-pressure waterjet. After drying each mold part using com-

pressed air, the surfaces of the mold parts that are intended to be in contact with the curing silicone were treated by applying two layers of InhibitX (Smooth-On) using a soft brush. This is paramount to ensure a proper curing at the interface between the silicone and the mold surface. Once the InhibitX dried and before assembling the mold, the same surfaces were sprayed with mold release Ease Release 200 (Mann). When a mold was used again, it was cleaned using a high-pressure waterjet, then prepared following the same method: drying, InhibitX, mold release.

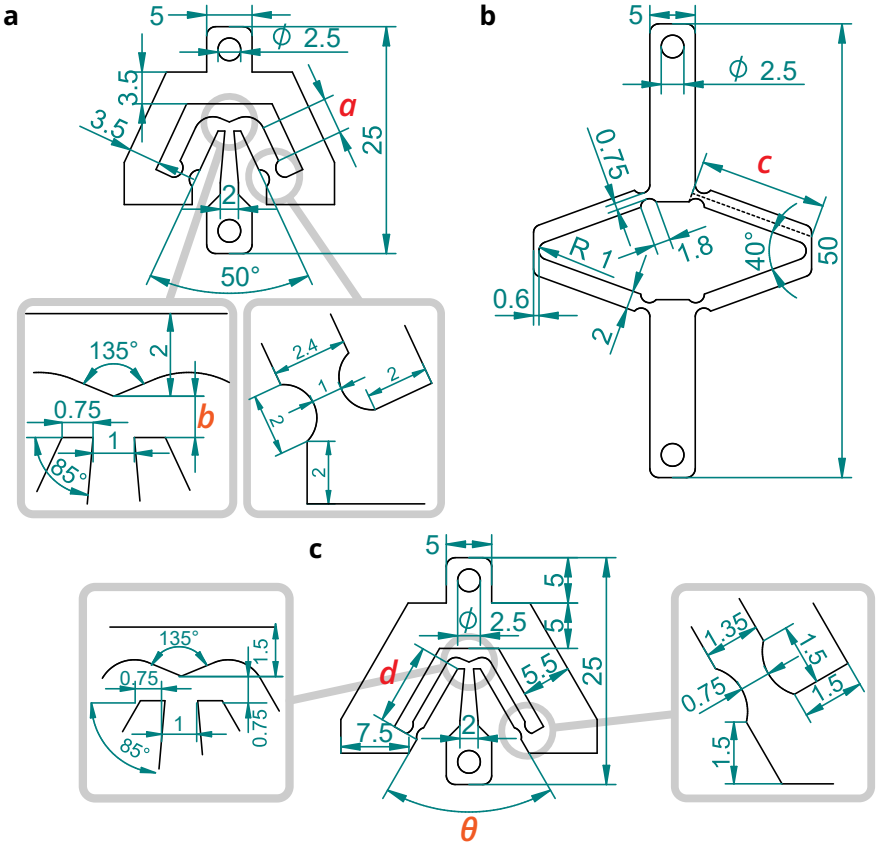
Material and cartridge preparation The prepolymer mixture that cures inside the mold was obtained as follows. First, the prepolymers part A and B were stirred for about 30 s separately. To every 100 g of part A 4 g of accelerant (Plat-Cat, Smooth-On) was added. The mix of part A and accelerant is mixed thourougly using a planetary mixer (Dispermill KK250) for 90 s, then poured into one compartment of a two-compartment cartridge (MIXPAC AF 400-01-10-01, Sulzer), while part B is poured into the other one. Before closing and sealing the compartements, the cartridge is placed vertically in a vacuum chamber for degassing (pressure of about -0.8 bar) for one hour.

Injection procedure To mix and inject the prepolymers into the mold, the cartridge was placed in a pneumatic dispenser gun (MIXPAC DP400-85, Sulzer) and a static mixer (QUADRO MFQ 05-24L, Sulzer) was mounted to its end and secured using a retaining nut (MIXPAC UM10-PP, Sulzer). To the Luer end of the static mixer was attached a female Luer (FTLL035), which was in turn connected to a silicone tube (5x8 mm) about 15 cm long, which was in turn connected to a male Luer (MLRL035) with a rotating ring lock (FSLLR) to which a dispensing conical needle (EFD7005009, 0.58 mm, Nordson) was screwed. Each assembled mold was secured vertically with the inlet hole facing downward, into which the conical needle was inserted. Each mold was slowly filled (about 1 min) until silicone started pouring from the outlet. Each mold was placed horizontally for about 12 hours at room temperature for curing.

Post-processing After removing each sample from their mold, all samples were post-cured by leaving them at 80°C in a fan-assisted oven (UF30,

Memmert). The samples were not stretched before the post-processsing was complete.

Assembly The building blocks were manually assembled together using 3d-printed PLA connectors, as shown in Fig. S2.2 and Fig. 2.1m.



panels d-n on the next page

Figure S2.1: Fabrication process of the building blocks. **(a-c)** Drawings of the softening (a), stiffening (b) and nonmonotonic (c) building blocks. Values for the design parameters a, b, c, d and θ are available in Fig. S2.3f. **(continuing on the next page)**

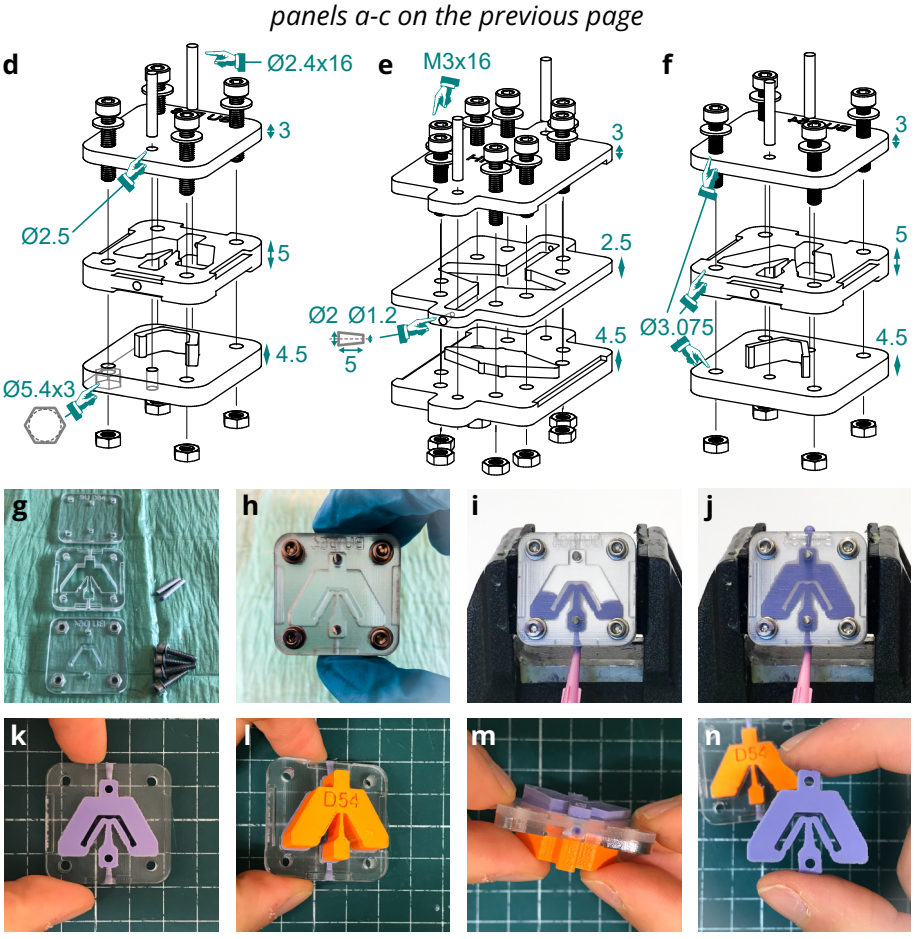


Figure S2.1 (continued): (d-f) Mold design and assembly for the softening (d), stiffening (e) and nonmonotonic (f) building blocks. **(g-n)** Mold preparation (g-h), injection procedure (i-j) and demolding process (k-n) for a nonmonotonic building block.

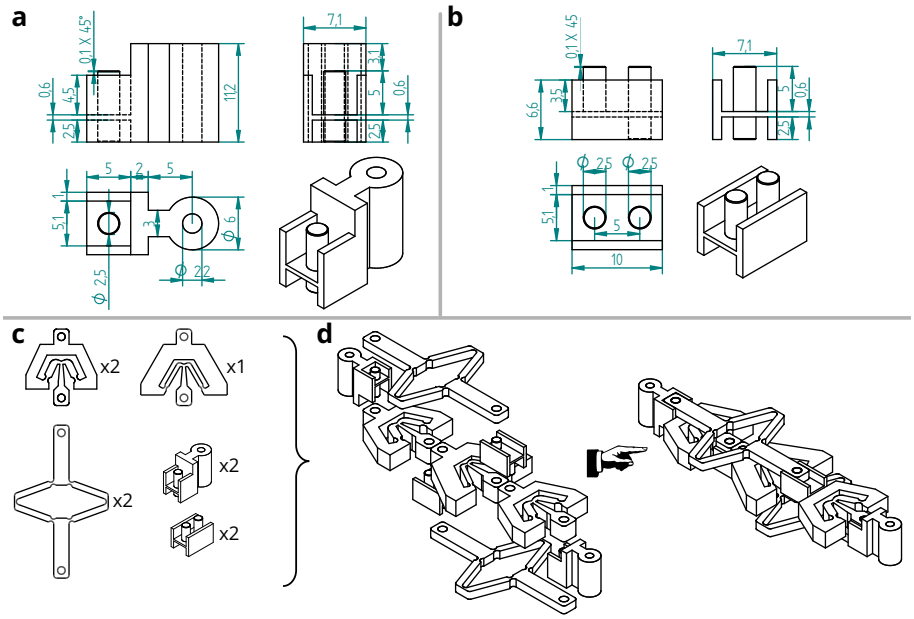


Figure S2.2: Assembly of the building blocks into a countersnapping structure. **(a)** End connector drawing. **(b)** Internal connector drawing. **(c)** List of mechanical entities required for the assembly. **(d)** Assembly process.

S2.1.2 Method to numerically predict the behavior of assemblies of nonlinear building blocks

To identify which building blocks should be assembled together to form a structure that countersnaps, we proceeded as follows.

- Each building block was characterized by measuring its force-displacement tensile curve (Fig. S2.3a-c).
- For each building block, a cubic Bezier curve was fitted to the measured force-displacement curve. A numerical nonlinear longitudinal spring was associated to each building block, with the tensile response of the spring defined by the cubic Bezier curve (Fig. S2.3a-c).
- The arclength method was implemented to compute the equilibrium path of a structure consisting of five longitudinal nonlinear springs assembled in the network topology shown in Fig. S2.3d.
- For each combination of building blocks, the equilibrium path was computed and the direction of the snapping instability was determined (Fig. S2.3e).
- One numerical simulation predicting countersnapping was selected, then validated against the experimental tensile response (Fig. S2.3f-g).

Representation of each building block as a longitudinal nonlinear spring For the purpose of the numerical simulation, each building block is reduced to a longitudinal nonlinear spring, whose tensile behavior is represented by a cubic Bezier curve. A cubic Bezier curve is represented by the following parametric equations:

$$x_{[x_0,x_1,x_2,x_3]}(t) = x_0(1-t)^3 + x_1(1-t)^2 t + x_2(1-t)t^2 + x_3 t^3 \tag{S2.1}$$

$$y_{[y_0,y_1,y_2,y_3]}(t) = y_0(1-t)^3 + y_1(1-t)^2 t + y_2(1-t)t^2 + y_3 t^3, \tag{S2.2}$$

where t is a parameter that runs from 0 to 1 to draw the Bezier curve, while x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3 are constant parameters that shape the curve. Upon choosing suitable values for x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, the Bezier curve $(x(t), y(t))$ can

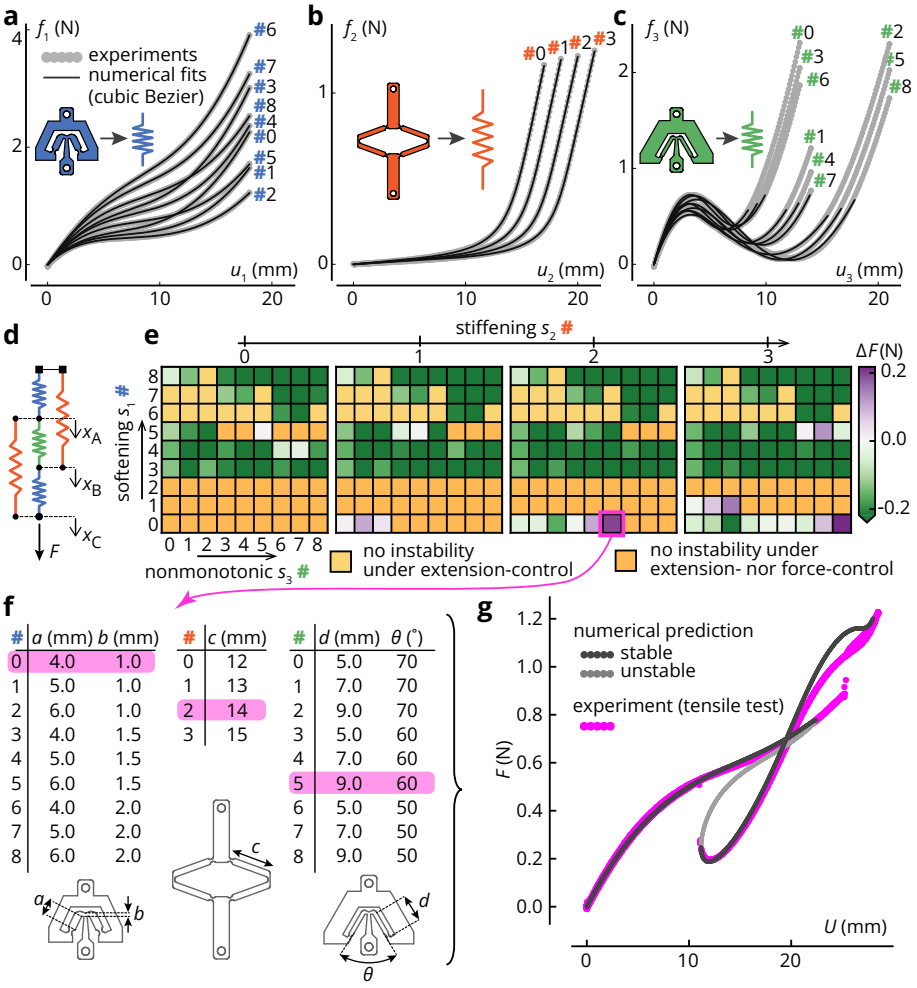


Figure S2.3: Numerical approach used to guide the design of the countersnapping structure. **(a-c)** Experimental force-displacement curves and numerical fits of the softening (a), stiffening (b) and nonmonotonic (c) building blocks. **(d)** Nonlinear spring network. **(e)** Results of the brute-force search. Each of the 324 squares corresponds to one combination of softening, stiffening and nonmonotonic building blocks. Their color indicates the sudden change in force due to the first (displacement-driven) snapping instability encountered. **(f)** Tables mapping the design id number to the set of design parameters for each type of building block. The combination of building blocks used to physically realize the countersnapping structure is highlighted in magenta. **(g)** Comparison between the force-displacement curves predicted by the numerical simulation and obtained through the tensile test experiment.

be tuned to fit the experimental data (u_i, f_i). To ensure that the nonlinear spring yields no force when the extension is zero, x_0 and y_0 were set to 0. The 6 remaining fitting parameters $\boldsymbol{p} = [x_1, x_2, x_3, y_1, y_2, y_3]$ were found by minimizing the mismatch between the Bezier curve and the experimental curve. By enforcing $0 < x_1 < x_2 < x_3$, $x_{[0,x_1,x_2,x_3]}(t)$ is a monotonic function of t and therefore has an inverse, $t = x^{-1}_{[0,x_1,x_2,x_3]}(x)$. The Bezier curve can then be expressed as a function $\bar{y}_{\boldsymbol{p}}$ of x

$$y = y_{[0,y_1,y_2,y_3]}(x^{-1}_{[0,x_1,x_2,x_3]}(x)) = \bar{y}_{\boldsymbol{p}}(x), \tag{S2.3}$$

and the minimization problem takes the following form

$$\min_{\boldsymbol{p}} \quad \sum_{i=1}^{n} \left(f_i - \bar{y}_{\boldsymbol{p}}(u_i)\right)^2, \tag{S2.4}$$

$$\text{s.t.} \quad 0 < x_1 < x_2 < x_3, \tag{S2.5}$$

where n is the number of experimental data points. For each building block, the experimental force-extension curve (u_i, f_i) considered is the one obtained during the second loading sequence. For the nonmonotonic building blocks, this experimental curve was truncated so that (u_n, f_n) is the state at which the nonmonotonic building block would be just after a snap-through instability. This helps obtain a better fit to the experimental data in the regime where the snapping building block will be used. For convenience, it is easier to express the force f_s in the spring s directly as a function of the extension u_s. To do this, we used $t = x^{-1}(x)$ (which is well-defined since we choose $0 < x_1 < x_2 < x_3$, forcing $u(t)$ to be a monotonic function of t), which leads to

$$f_s(u_s) = y(t) = y(x^{-1}(x)) = \bar{y}(x) = \bar{y}(u_s). \tag{S2.6}$$

To compute the tangent stiffness k_s for spring s given its extension u_s, we used

$$k_s(u_s) = k(t) = k(x^{-1}(x)) = \bar{k}(x) = \bar{k}(u_s), \tag{S2.7}$$

where $k(t) = y'(t)/x'(t)$ is the slope of the Bezier curve at the point $(x(t), y(t))$. The experimental force-extension curves along with their respective fitting Bezier curves are represented in Fig. S2.3a-c.

Simulating the nonlinear spring network We used the arclength method [98, 99] to compute the equilibrium states of the assembled structure under tension. This method requires to compute the external nodal force vector f_i^{ext}, the internal nodal force vector f_i^{int} and the stiffness matrix K_{ij} of the spring structure depicted in Fig. S2.3d. The external nodal force vector $\boldsymbol{F}^{\text{ext}}$ is the set of external forces applied on each of the three degrees of freedom $\boldsymbol{Q} = [x_A, x_B, x_C]$, and is therefore given by

$$\boldsymbol{F}^{\text{ext}} = \begin{pmatrix} 0 \\ 0 \\ F \end{pmatrix}, \tag{S2.8}$$

where F is the tensile force applied on the bottom node of the structure, corresponding to the degree of freedom x_C (Fig. S2.3d) To compute the internal nodal force vector F_i^{int}, we start by expressing the elastic energy U of the structure as a function of the degrees of freedom $\boldsymbol{Q} = [x_A, x_B, x_C]$

$$E(\boldsymbol{Q}) = e_1(x_A) + e_3(x_B - x_A) + e_2(x_B) + e_1(x_C - x_B) + e_2(x_C - x_A), \tag{S2.9}$$

where $e_i(.)$ is the elastic energy stored in a nonlinear spring of type i. The internal nodal force vector F_i^{int} is then given by

$$F_i^{\text{int}}(\boldsymbol{Q}) = \frac{\partial E}{\partial Q_i}, \tag{S2.10}$$

or in a more developed form

$$F_1^{\text{int}} = f_1(x_A) - f_3(x_B - x_A) - f_2(x_C - x_A) \tag{S2.11}$$

$$F_2^{\text{int}} = f_2(x_B) + f_3(x_B - x_A) - f_1(x_C - x_B) \tag{S2.12}$$

$$F_3^{\text{int}} = f_2(x_C - x_A) + f_1(x_C - x_B), \tag{S2.13}$$

where $f_i(.) := e_i'(.)$ is the derivative of the elastic energy $e_i(.)$. The stiffness matrix $K_{ij}(\boldsymbol{Q})$ is given by

$$K_{ij}(\boldsymbol{Q}) = \frac{\partial^2 E}{\partial Q_i \partial Q_j} = \frac{\partial F_i^{\text{int}}(\boldsymbol{Q})}{\partial Q_j}, \tag{S2.14}$$

or in a more developed form

$$K_{11} = k_1(x_A) + k_3(x_B - x_A) + k_2(x_C - x_A) \tag{S2.15}$$

$$K_{12} = -k_3(x_B - x_A) = K_{21} \tag{S2.16}$$

$$K_{13} = -k_2(x_C - x_A) = K_{31} \tag{S2.17}$$

$$K_{22} = k_1(x_C - x_B) + k_3(x_B - x_A) + k_2(x_B) \tag{S2.18}$$

$$K_{23} = -k_1(x_C - x_B) = K_{32} \tag{S2.19}$$

$$K_{33} = k_1(x_C - x_B) + k_2(x_C - x_A), \tag{S2.20}$$

where $k_i(.) := f_i'(.) = e_i''(.)$ is the derivative of the tensile force $f_i(.)$. Given these three ingredients, the arclength method allowed us to retrieve the entire equilibrium path connected to the initial configuration. This path is composed of equilibrium points $(\boldsymbol{Q}_{eq}, F)$ whose stability can be assessed by evaluating the stiffness matrix of the entire structure, $\boldsymbol{K}(\boldsymbol{Q}_{eq})$. The stability of an equilibrium point depends on the loading mode. If one drives the force, an equilibrium point is stable if and only if all the eigenvalues of the stiffness matrix $\boldsymbol{K}(\boldsymbol{Q}_{eq})$ are positive. If one drives the extension, an equilibrium point is stable if and only if the eigenvalues of the reduced stiffness matrix $\tilde{\boldsymbol{K}}(\boldsymbol{Q}_{eq})$ are positive. The reduced stiffness matrix is defined as the stiffness matrix $\boldsymbol{K}$ from which the row and column corresponding to the degree of freedom controlled have been removed. Note that when an equilibrium point is stable under force control, it will necessarily be stable under extension control, but the opposite is generally not true.

COMPUTATIONAL DETAILS We scanned all the combinations of nonlinear springs and simulated every single assembly. In total, we performed 324 simulations (9 softening building blocks × 4 stiffening building blocks × 9 snapping building blocks). The simulations were run on a desktop PC with the following specifications: Windows 10 Enterprise, Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, 16GB RAM. The total time needed to simulate all the assemblies was 7 min 26 s. The average time per assembly was 1.38 s (standard deviation of 0.39 s). The median time was 1.27 s. The arclength method involves evaluating the residual of the system, evaluating the jacobian of the system and solving linear systems iteratively. Together, these three tasks accounted for about 75% of the computation time. Cumulatively, over all assembly simulations, system

residuals and jacobians were evaluated 338582 times each and 561291 linear systems (3×3) were solved.

Overview of the scanning results, identification of promising combinations and experimental validation For each computed equilibrium path, we extracted the force-displacement curve which allowed us to determine the direction of the first snapping instability ΔF (if any). The direction of the snapping instability is defined as the sudden change in reaction force F caused by a snapping instability under displacement-driven conditions. If negative, the instability causes the reaction force to drop (regular snapping); if positive, the instability causes the reaction force to jump (countersnapping). For many combinations, the structure formed is not subject to any instability under displacement-driven conditions (the force-displacement curve does not "curve back"), or even under force-driven conditions (the force-displacement curve is monotonic). In these two latter cases, we cannot define any sudden force change.

The results of the simulations are shown in Fig. S2.3e. As show in Fig. S2.3e, a few building block assemblies are predicted to show a countersnapping instability (purple-colored squares). We selected a combination with a relatively pronounced predicted force jump which corresponds to a specific set of design parameters (as highlighted in magenta in Fig. S2.3e-f). Once we manufactured extra softening and stiffening building blocks, the entire structure was built, tested physically, and compared to the numerical prediction. As shown in Fig. S2.3g, the spring model correctly predicts the countersnapping instability. The slight discrepancy is mainly caused by the small difference in behaviors between samples with the same design, and the fact that the building blocks are not aligned along exactly one line as it is the case in the simulation.

S2.1.3 Testing methods

All extension-driven tensile tests were performed using a vertical uniaxial testing machine (model 5965, Instron) equipped with a 100-N load cell, except for the tensile tests involving countersnapping elements combined in series that were performed using a horizontal uniaxial testing machine (custom model, controller type 5900, Instron) equipped with a 10-N load cell.

Tensile test of the individual building blocks (displacement-controlled), Fig. 2.1j-l Each sample was attached to two PLA connectors (similar to the one shown in Fig. S2.2a), which were then clamped by the testing machine clamps. Each sample was subjected to a cyclic loading (30 s starting hold at 0 mm, 4 loading-unloading cycles at 1 mm/s). Acquisition rate: 10 Hz.

Tensile test of the countersnapping structure (displacement-controlled), Fig. 2.2a-c The countersnapping structure was attached to the testing machine by clamping its PLA end connectors (similar to the one shown in Fig. S2.2a). Each sample was subjected to a cyclic loading (30 s starting hold at 0 mm, 4 loading-unloading cycles at 1 mm/s). Acquisition rate: 10 Hz.

Force-driven tensile test of the countersnapping structure, Fig. 2.2d-f The countersnapping structure was inserted into a custom-made sliding mechanism allowing it to deform only uniaxially (preventing bending and swinging motion). The mechanism consists of a slider able to move up and down in the slots of two parallel acrylic plates. The slider featured a similar connecting design to the one shown in Fig. S2.2a. We attached the bottom of the countersnapping structure to the slider. The top of the countersnapping structure was attached to the fixed connector, located in between both plates. The sliding mechanism was attached vertically from the top of the 100-N load cell. We attached a plastic cup to the slider using an thin inextensible rope. The force was measured over time using the load cell (acquisition rate = 100 Hz). The extension was monitored by tracking an ArUco marker, sticked to the slider, and recording its position using a high-resolution camera (EOS 850D, Canon) at 50 frames per second. Tracking was performed using the blob detection algorithm of the OpenCV Python library. Water was manually poured into the cup using a syringe. Note that we assumed that the applied force value (weight of the cup and water) was the same as the force measured using the load cell, except around the countersnapping instability which caused the system to enter a temporary dynamic regime for about 0.5 s. The force values measured by the load cell during that short dynamic period were discarded and replaced by values deduced from a linear regression (time-force) based on the force evolution data collected a few second before and after the dynamic phase.

Lift-weight experiment, Fig. 2.2g-i The weight was a custom-made 3d-printed PLA block to which flat cylindrical calibrated weights can be added. The total weight was about 0.83 N. The weight was placed on a horizontal platform right underneath the top clamp of the vertical tensile machine. The platform was covered by a sheet of paper to mitigate the effect of electrostatic forces between the platform and the PLA weight. The weight was attached to the bottom of the countersnapping structure using a thin, inextensible rope. The top of the countersnapping structure was clamped via the top end connector to the movable head of the uniaxial tensile machine. The top clamp was raised by 26mm at 0.05 mm/s, then lowered back to its initial level at 2.00 mm/s. This cycle was repeated 4 times. The top displacement was measured using the uniaxial vertical tensile machine (acquisition rate: 100 Hz), while the weight elevation was measured by tracking two black dots marked on the weight using a high-resolution camera (EOS 850D, Canon) at 50 frames per second.

Stick-slip experiments, Fig. 2.3 The mechanical structures were cyclically loaded using the linear motion of a programmed robotic arm (UR5e, Universal Robots), pulling on the end connector via a soft elastic rubber band (measured stiffness: 0.003 N/mm). The loaded end connector was 3d-printed in VeroClear (Stratasys). The foam block was cut from a white foam board. It is 51 mm long, 34 mm wide, 5 mm thick and weighed 0.7 g. One side was made rougher by scraping it against sand paper. The other side was kept smooth and intact, and was in contact with the bottom surface (anodized aluminum sheet). The desired amount of roughness (and friction) was tuned by incrementally scraping against sand paper up to the point where stick behavior was observed during slow motion (regular loading) and slip behavior during fast motion (snapping). The same foam block with the same level of roughness was used for all experiments. The regular snapping structure was a nonmonotonic building block with parameters $d = 9$ mm, $\theta = 60^{\circ}$. The change in drift, defined as the change of distance between the foam block and the contact point (end connector/foam block) was computed by tracking both the foam block position and contact point position using high-resolution cameras (EOS 850D, Canon), recording simultaneously at 50 fps. The foam block position was monitored by tracking black markers marked on the top of the foam block (top view camera), while the contact point

position was computed by tracking three holes in the end connector (side view camera).

Free oscillation test, Fig. 2.4c-e The set-up was the same as for the force-driven tensile test. The amount of water added to the cup was tuned so that the extension of the countersnapping structure is the same for both stable configurations. This was ensured by comparing photographs of both configurations taken from the same camera set-up. The change in extension ΔU compared to the preloaded configuration was monitored by tracking the aruco marker and recording its position using a high-resolution camera (EOS 850D, Canon) at 50 frames per second.

Forced oscillation test, Fig. 2.4f-h The set-up was the same as for the force-driven tensile test, except for two aspects. First, the weight consists of a dumbbell composed of a PLA slider to which is attached a threaded rod carrying washers and nuts to tune the load to the desired level. Second, the system was attached vertically to the end effector of a 6-axis robotic arm (UR5e, Universal Robots). The oscillating displacement input and output were monitored by sticking circular markers to the top and bottom of the structure, and recording them using a high-resolution camera (EOS 850D, Canon) at 50 frames per second. The input and output displacements were measured by detecting the vertical position of the markers on every frame using the blob-detection algorithms of the OpenCv library.

Parallel-coupled countersnapping elements tensile test, Fig. 2.5a-g, Fig. S2.8 Countersnapping elements were combined using custom-made connectors, 3d-printed in VeroClear (Stratasys), that were attached to the uniaxial testing machine. Force and displacement values were measured at 100 Hz. Loading speed was 1 mm/s.

Serially-coupled countersnapping element tensile tests, Fig. 2.5h-n, Fig. 2.6a, Fig. S2.9 To combine countersnapping elements in series, connectors were 3d-printed in VeroClear using a PolyJet Eden260VS printer (Stratasys). Those connectors were designed in such a way that they connected the stiffening (and softening) building blocks of adjacent countersnapping elements at the

same level. Those connectors were designed with long feet that allowed them to slide on the (horizontal) surface (anodized aluminum sheet) without letting the countersnapping elements themselves touch the surface. To mitigate the effect of friction when three countersnapping elements were connected in series, the connectors were sliding on glass microscopy slices brushed with a thin layer of oil (Morlina S2 BL, Shell). The force and displacement were measured at 100 Hz. For the tests with two countersnapping elements, the loading speed was 2 mm/s. For the tests with three countersnapping elements, the loading speed was 3 mm/s.

Serially-coupled countersnapping elements, constant length tests, Fig. 2.6b The two countersnapping elements were connected in series using a 3d-printed PLA connector in such a way that the stiffening (and softening) building blocks of each countersnapping element were connected at the same level. The chain that they formed was prestretched in a custom-made, 3d-printed PLA linear stage that allowed for tuning the desired total length, then keeping it fixed for the duration of the test. The prestretch applied was about 47 mm. The behavior of the structure was recorded using a high-resolution camera (EOS 850D, Canon) at 25 fps.

S2.2 Additional conditions to obtain countersnapping

The force-displacement curves of the individual nonlinear springs must satisfy some conditions in order to obtain a countersnapping instability. To derive these conditions, we first determine the equilibrium points of the system, then study the stability of these equilibria. To determine to equilibrium points of the system, we first express the total potential energy $\Pi(.,.,.)$ of the system described in Fig. S2.3d as function of the three degrees of freedom (DOF) x_A, x_B, x_C:

$$\Pi(x_A, x_B, x_C) = e_1(x_A) + e_2(x_B) + e_3(x_B - x_A) + e_1(x_C - x_B) + e_2(x_C - x_A) - x_C F, \tag{S2.21}$$

where $e_i(.)$ is the elastic energy stored in a nonlinear spring of type i, F is the force applied on the bottom node. Equilibrium points are stationary points of

the total potential energy Π, so they must satisfy the following conditions

$$0 = \frac{\partial \Pi}{\partial x_A} = f_1(x_A) - f_3(x_B - x_A) - f_2(x_C - x_A) \tag{S2.22}$$

$$0 = \frac{\partial \Pi}{\partial x_B} = f_2(x_B) + f_3(x_B - x_A) - f_1(x_C - x_B) \tag{S2.23}$$

$$0 = \frac{\partial \Pi}{\partial x_C} = f_1(x_C - x_B) + f_2(x_C - x_A) - F, \tag{S2.24}$$

where the tension force $f_i(.) := e_i'(.)$ is the derivative of the elastic energy e_i. From Eqs. (S2.22, S2.23), we have

$$f_1(x_A) - f_1(x_C - x_B) = f_2(x_C - x_A) - f_2(x_B) \tag{S2.25}$$

In the following, we show that by assuming that $f_1(.)$ and $f_2(.)$ are monotonically increasing functions, $x_A = x_C - x_B$ is necessarily true, which is a useful result that will simplify the stability analysis we will conduct next. For the sake of the proof, let's assume $x_A > x_C - x_B$. In that case, given the monotonicity of $f_1(.)$ and $f_2(.)$, $x_C - x_A > x_B$ in order to maintain Eq. (S2.25) balanced, which contradicts our initial assumption. Therefore, $x_A = x_C - x_B$. This result means that our system has effectively only two DOF. We can choose u_1 and u_2 (stretch of springs s_1 and s_2) to represent those. The total potential energy $\Omega(.,.)$ of the system can be then expressed as follows:

$$\Omega(u_1, u_2) = 2e_1(u_1) + 2e_2(u_2) + e_3(u_2 - u_1) - (u_1 + u_2)F, \tag{S2.26}$$

and the equilibrium conditions take the following form

$$0 = \frac{\partial \Omega}{\partial u_1} = 2f_1(u_1) - f_3(u_2 - u_1) - F \tag{S2.27}$$

$$0 = \frac{\partial \Omega}{\partial u_2} = 2f_2(u_2) + f_3(u_2 - u_1) - F. \tag{S2.28}$$

When the above equations are rewritten as follows:

$$(f_1(u_1) + f_2(u_2))/2 = F/2 \tag{S2.29}$$

$$f_1(u_1) - f_2(u_2) = f_3(u_2 - u_1), \tag{S2.30}$$

it is clear that half the global force F is the average of f_1 and f_2, while f_3 is the difference between f_1 and f_2, thereby justifying the geometric construction employing rectangles introduced in Chapter 2 to represent equilibrium configurations of the system.

To study the stability of the equilibrium points, we construct the hessian matrix $\boldsymbol{K}$ of the system, that is the matrix of the second derivatives of Ω with respect to the two degrees of freedom $\boldsymbol{Q} = [u_1, u_2]$,

$$\boldsymbol{K} = \frac{\partial^2 \Omega}{\partial \boldsymbol{Q} \partial \boldsymbol{Q}^\top} = \begin{bmatrix} 2k_1(u_1) + k_3(u_3) & -k_3(u_3) \\ -k_3(u_3) & 2k_2(u_2) + k_3(u_3) \end{bmatrix}, \tag{S2.31}$$

where the stiffness $k_i(.) := f_i'(.) = e_i''(.)$ is the derivative of the tension force f_i, and $u_3 := u_2 - u_1$. By applying Sylvester's criterion, we derive that the system is unstable if and only if

$$k_3(u_3) < -2\frac{k_1(u_1)k_2(u_1)}{k_1(u_1) + k_2(u_2)}, \tag{S2.32}$$

where the right-hand side is always negative given the monotonicity of f_1 and f_2.

This means that two additional conditions must be met when drawing the force-displacement curves during the third step of the geometric construction detailed in Chapter 2 (Fig. 2.1i):

- for the system to be on the verge of becoming unstable, the slope of $f_3(u_3)$ must equal the (negative) right-hand side of Ineq. (S2.32),
- for the system to be stable after the sudden reconfiguration, the slope of $f_3(u_3)$ must be greater than the (negative) right-hand side of Ineq. (S2.32).

It is important to note that the slope of $f_3(u_3)$ must approach the prescribed negative critical value by decreasing as the system is loaded. This implies that the stiffness k_1 (slope of f_1) must be greater than the stiffness k_2 (slope of f_2) when reaching the critical equilibrium point. Otherwise, the spring s_3 will see its elongation decrease when approaching the critical point, thereby making the desired critical point actually unreachable by simply loading the system from the starting configuration.

S2.3 Geometric construction of local force-displacement curves leading to countersnapping during unloading

The geometric construction of local force-displacement curves introduced in Chapter 2 can also be used to identify individual force-displacement curves leading to countersnapping during unloading, instead of loading.

We consider the same nonlinear spring network as illustrated in Fig. S2.4a. As shown in Fig. S2.4b, we first draw a rectangle whose top-left and bottom-right corners correspond to the states of springs s_1 and s_2, just before the instability. Next, because we want the restabilized state to be more elongated (and at the same level of critical force), we draw a second rectangle whose center is shifted, horizontally to the *right* compared to the first rectangle. Again, the top-left and bottom-right corners correspond to the state of s_1 and s_2, just after the instability. Since the widths and heights of both rectangles correspond to the states of the spring s_3 before and after the instability, two additional markers can be positioned in the force-displacement plane (green dot and green star). The three individual force-displacement curves are then plotted through their corresponding markers. The same two conditions as detailed in the previous section must be met by the curve $f_3(u_3)$ to ensure that the first rectangle describes a critical equilibrium point. Importantly, the slope of $f_3(u_3)$ must approach the prescribed negative critical value by decreasing as the system is unloaded. This implies that the stiffness k_1 (slope of f_1) must be *smaller* than the stiffness k_2 (slope of f_2) when reaching the critical equilibrium point. Otherwise, the spring s_3 will see its elongation decrease when approaching the critical point, thereby making the desired critical point actually unreachable by simply unloading the system from the stretched configuration.

To confirm that such a set of curves leads to countersnapping, we numerically simulate the nonlinear spring network (using the arclength method, detailed in Chapter S2, Section S2.1.2) using the exact curves drawn in Fig. S2.4b, and compute the force-displacement curve of the system. As shown in Fig. S2.4c, if we start unloading the system from the preloaded configuration (marked by the triangle) by reducing the tensile force, the elongation slowly decreases. Upon reaching a critical force level (marked by the circle), the system becomes unstable and reconfigures suddenly in a more elongated state (marked by a star). Less importantly, we can notice that the system will later undergo a second snap-through

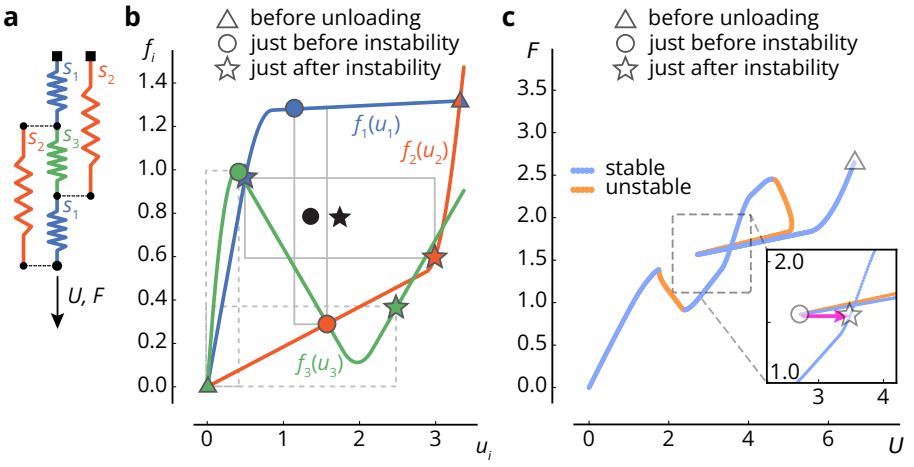


Figure S2.4: Geometric construction of local force-displacement curves leading to countersnapping during unloading. **(a)** Nonlinear spring network. **(b)** Individual nonlinear force-displacement curves of each spring leading to countersnapping during unloading. **(c)** Numerically simulated force-displacement curve of the spring assembly shown in (a). By unloading the structure from the equilibrium point marked by a triangle (high tension, high stretch), the first encountered instability leads to a sudden elongation as the force decreases.

instability (this time leading to a typical decrease in elongation) if the force is reduced further. This particular system shares similar properties as the countersnapping structure used in Chapter 2: self-intersecting force-displacement curve, force drop if one unloads under displacement-driven conditions, etc.

The individual curves used for the simulation are piecewise linear curves whose corners have been rounded off using a quadratic function. They can be expressed mathematically as follows:

$$f(u) = \sum_{i=0}^{n-1} L_i(u)(k_i u + p_i) + Q_i(u)(a_i u^2 + b_i u + c_i), \tag{S2.33}$$

where f is the force, u is the displacement, k_i is the slope of the ith segment and $n \geq 2$ is the number of segments composing the (not-smoothened) piecewise

linear curve. The quantities $L_i(u)$, $Q_i(u)$, p_i, a_i, b_i and c_i are defined as follows:

$$L_i(u) = \begin{cases} \begin{cases} 1 & u < u_i - \epsilon \\ 0 & \text{else} \end{cases} & i = 0 \\ \begin{cases} 1 & u_{i-1} + \epsilon \leq u < u_i - \epsilon \\ 0 & \text{else} \end{cases} & 1 \leq i \leq n-2 \\ \begin{cases} 1 & u_{i-1} + \epsilon \leq u \\ 0 & \text{else} \end{cases} & i = n-1 \end{cases} \tag{S2.34}$$

$$Q_i(u) = \begin{cases} \begin{cases} 1 & u_i - \epsilon \leq u < u_i + \epsilon \\ 0 & \text{else} \end{cases} & 0 \leq i \leq n-2 \\ 0 & i = n-1 \end{cases} \tag{S2.35}$$

$$p_i = \begin{cases} 0 & i = 0 \\ p_{i-1} + u_{i-1}\left(k_{i-1} - k_i\right) & 1 \leq i \leq n-1 \end{cases} \tag{S2.36}$$

$$a_i = \begin{cases} \frac{k_{i+1} - k_i}{4\epsilon} & 0 \leq i \leq n-2 \\ 0 & i = n-1 \end{cases} \tag{S2.37}$$

$$b_i = \begin{cases} \frac{k_i(u_i+\epsilon) - k_{i+1}(u_i-\epsilon)}{2\epsilon} & 0 \leq i \leq n-2 \\ 0 & i = n-1 \end{cases} \tag{S2.38}$$

$$c_i = \begin{cases} \frac{(k_{i+1} - k_i)(u_i-\epsilon)^2}{4\epsilon} + p_i & 0 \leq i \leq n-2 \\ 0 & i = n-1 \end{cases}, \tag{S2.39}$$

where u_i is the displacement at which the (not-smoothened) piecewise linear curve goes from slope k_i to k_{i+1} and ϵ is a parameter to tune the smoothing which must satisfy the following constraint,

$$\epsilon < \min\left\{ \min_{1 \leq i \leq n-2} \left\{ \frac{u_i - u_{i-1}}{2} \right\}, u_0 \right\} \tag{S2.40}$$

The force-displacement curves of springs s_1, s_2 and s_3 are defined using the following parameters,

- spring s_1: $\boldsymbol{k} = [1.9438; 0.01723; 3.0603]$, $\boldsymbol{u} = [0.6542; 3.9302]$, $\epsilon = 0.2$

- spring s_2: $\boldsymbol{k} = [0.18401; 3.0934]$, $\boldsymbol{u} = [3.0766]$, $\epsilon = 0.2$
- spring s_3: $\boldsymbol{k} = [4.7619; -0.6042; 0.61]$, $\boldsymbol{u} = [0.2315; 1.9711]$, $\epsilon = 0.2$.

S2.4 Sensitivity to the geometric parameters

In this section, we conduct a two-step analysis to investigate the interplay between geometric parameters and the countersnapping effect.

For the first step, we investigated how the force-displacement curves of the individual building blocks are influenced by the geometric parameters (Fig. S2.5a-e), to map the geometric parameters of each building block to the shape of their force-displacement curve. This mapping was established by analyzing how the control points' coordinates of the Bezier curves that fit the experimental tensile test data of the individual building blocks are modified as one geometric parameter is varied.

For any value of a geometric parameter, we estimate the force-displacement curve of the concerned building block by constructing a cubic Bezier curve whose control points' coordinates are determined by the interpolation or extrapolation of the discrete set of measured samples.

For example, for the parameter a (that is, the length of the inclined beam of the softening unit), we identified the three softening building blocks for which a was varied ($a = [4, 5, 6]$ mm) while keeping the other parameter constant, namely $b = 1$ mm. For each of the three softening building blocks (respectively characterized by $(a, b) = [(4, 1), (5, 1), (6, 1)]$ mm), we determined the control points' coordinates of the cubic Bezier curve that fits their force-displacement tensile curve. To estimate the force-displacement curve for a new value of a, let us say $a = 4.5$ mm, we constructed a new cubic Bezier curve for which each of its control points' coordinates is estimated by interpolating between the corresponding known coordinates when $a = 4$ mm and $a = 5$ mm. The estimated force-displacement curves obtained for values of a ranging from 3 to 7 mm, with $b = 1$ mm are shown in Fig. S2.5a. The same process is repeated for parameters b, c, d and θ, for which results are shown in Fig. S2.5b, c, d and e, respectively. Note that for the softening and nonmonotonic building blocks, the value of the parameter that is kept constant (while the other one is varied) is the value used for the current countersnapping design structure with $a = 4$ mm, $b = 1$ mm, $d = 9$ mm

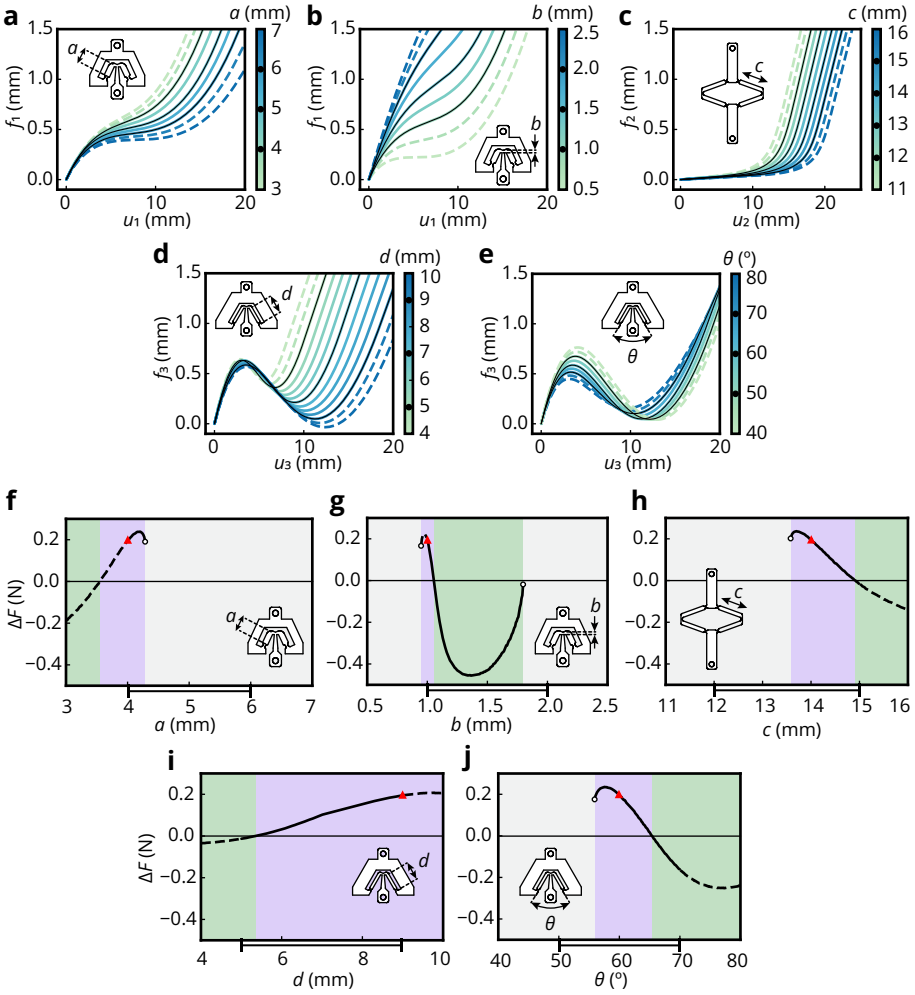


Figure S2.5: Influence of the geometric parameters. **(a-e)** Estimated force-displacement curves of the individual building blocks as the geometric parameters a (a), b (b), c (c), d (d), θ (e) are varied. Black lines correspond to the Bezier curves fitting the experimental data, for the geometric parameter values indicated in the color bar by circular black markers. Solid (dashed) lines denote estimated force-displacement curves obtained by interpolation (extrapolation). **(f-j)** Countersnapping magnitude ΔF as a function of the geometric parameters a (f), b (g), c (h), d (i), θ (j). Purple (green) backgrounds refer to regions where countersnapping, $\Delta F > 0$, (regular snapping, $\Delta F < 0$) is predicted. Gray backgrounds refer to regions where no instability occurs, ΔF is undefined. Triangular red markers indicate where the current design lies. Black solid (dashed) lines denote estimation from interpolated (extrapolated) building block force-displacement curves. Each interpolation range is depicted by a segment on the horizontal axes.

and $\theta = 60^{o}$.

For the second step, we use the mapping established in the first step to investigate the magnitude of the countersnapping effect under displacement-driven conditions as a function of the geometric parameters (Fig. S2.5f-j). We define this magnitude as the sudden change in force ΔF due to the instability. If the force jumps due to a countersnapping instability, the magnitude is positive ($\Delta F > 0$). If the force drops due to a regular instability, the magnitude is negative ($\Delta F < 0$). If there is no instability, we cannot define a magnitude (ΔF is undefined). We studied each parameter by independently varying each parameter with respect to the reference spring network (Fig. 2.1e), and show the magnitude ΔF as a function of each parameter in Fig. S2.5f-j. Note that the bar under the x-axis indicates the range for which data fall within tested parameters, where results outside this range represent less accurate extrapolated results.

From the results in Fig. S2.5 we can extract how sensitive the countersnapping effect is with respect to the geometric parameters. Importantly, changing the value of a parameter can continuously decrease the sudden change in force ΔF from a positive (countersnapping, purple regions) to a negative (regular snapping, green regions) value. Interestingly, tuning a geometric parameter can also remove any instability from the system (no instability, gray regions).

For example, the thickness of the central flexure of the softening unit, b, has a strong impact on the countersnapping effect. If b is slightly decreased, the system is not able to undergo an instability. If b is slightly increased, the force jumps quickly becomes a force drop (Fig. S2.5g). This influence makes sense intuitively. If b is large, the softening unit becomes a very stiff element (Fig. S2.5b), thereby leading the system to destabilize and reconfigure at a lower global displacement, at which the stiffening units are not yet in their stiffening regime. Inversely, if b is small, the softening unit becomes a very soft element (Fig. S2.5b), thereby requiring a larger global displacement to approach the instability, which eventually becomes unreachable as soon as the stiffening units stiffen. By contrast, the length of the beams of the nonmonotonic unit, d, has a significantly lower influence on the countersnapping effect. The parameter d mostly affects the valleys of the nonmonotonic curves, whereas it almost does not alter the peaks (Fig. S2.5d), which are crucial as they govern the onset of instability.

S2.5 Countersnapping upon both loading and unloading

In this section we discuss in detail the constraints on potential force displacement curves that realize countersnapping upon both loading and unloading (Fig. S2.6).

A naive approach to construct countersnapping upon loading and unloading is based on two stable branches, with the secondary branch shifted upwards and to the left compared to the primary one in the force-displacement plane. Assuming both stable branches are connected by an unstable branch, one might be tempted to draw an unstable path like in Fig. S2.6a,b. However, we can immediately exclude the existence of such force-displacement curves based on the existing folding rules [100]. This theory tells us that each time the curve folds clockwise (by reaching a force extremum, or a displacement extremum), the number of unstable modes (under force-, or displacement-driven conditions) is increased by one. Inversely, each time the curve folds counter-clockwise, the number of unstable modes under the respective loading conditions is decreased by one. Since the primary branch is stable, the curve must fold clockwise and counter-clockwise the same number of times before reaching the second stable branch, thereby making the curve drawn in Fig. S2.6a not realizable by a physical system. Also, the first fold must necessarily be clockwise since the number of unstable modes cannot be lower than zero, which also excludes the curve drawn in Fig. S2.6b.

The two stable branches could still potentially be connected by drawing a curve that folds first clockwise then counter-clockwise (Fig. S2.6c). However, while respecting these folding rules is necessary, it is not sufficient to guarantee that such a curve represents a physical elastic system: snapping instabilities at constant displacement should also cause a transition from a high elastic energy state to a lower elastic energy state. For the curve depicted in Fig. S2.6c, satisfying these conditions for both the loading and unloading snapping transitions is impossible, as we will demonstrate next.

First, we calculate the elastic energy $E_{\text{elastic}}(X)$ stored in a state X along the equilibrium curve using

$$E_{\text{elastic}}(X) = \int_0^X F(t)\frac{dU(t)}{dt}dt, \tag{S2.41}$$

where t is the curve parameter, which monotonically increases as one moves along the parametric curve $(U(t), F(t))$. Eq. (S2.41) defines an integral quantity

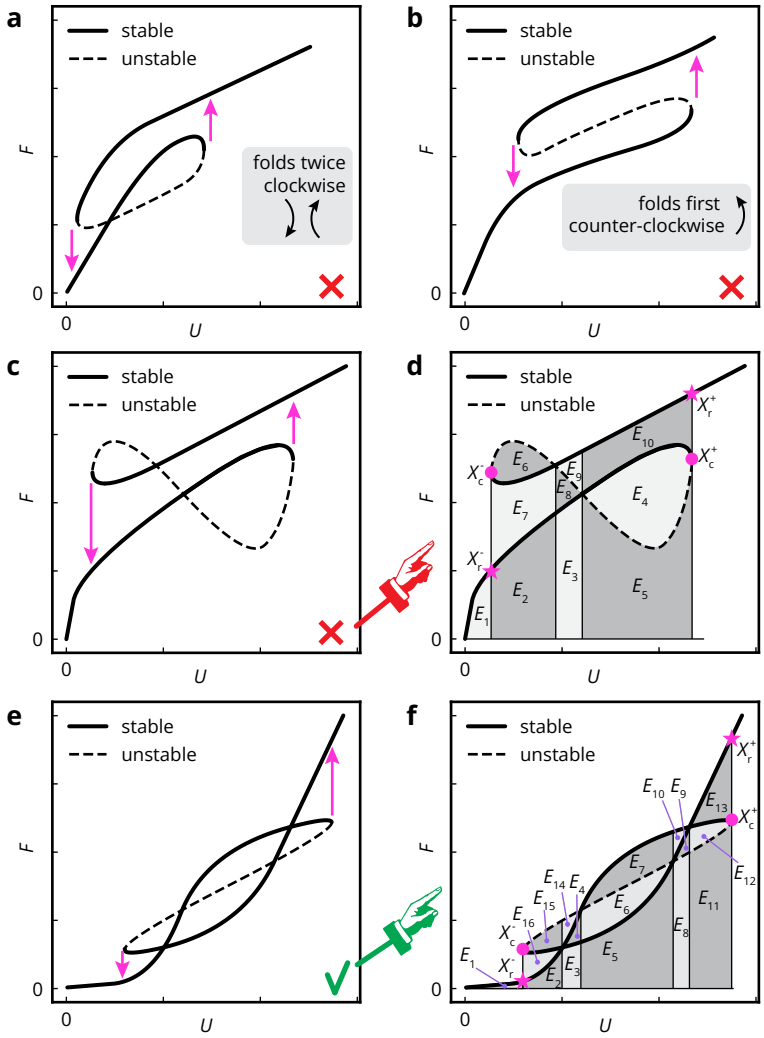


Figure S2.6: Force-displacement curves that seem to lead to countersnapping upon both loading and unloading. Each curve is composed of three branches. Solid (dashed) lines denote stable (unstable) branches. Pink arrows show the snapping transitions that would occur under displacement-driven conditions. **(a)** Impossible force-displacement curve, because the two stable branches are connected by a path that folds clockwise twice. **(b)** Impossible force-displacement curve, because the two stable branches are connected by a path that starts by folding counter-clockwise. **(c)** Impossible force-displacement curve, because both snapping transitions cannot be associated to an energy release. **(d)** Decomposition of the elastic energy associated to the force-displacement curve in (c) into areas. Pink dots (stars) refer to critical (restabilization) points. **(e)** Force-displacement curve that respects the "folding rules" and both energy release conditions (S2.44) and (S2.45). **(f)** Decomposition of the elastic energy associated to the force-displacement curve in (e) into areas. Pink dots (stars) refer to critical (restabilization) points.

that can be decomposed as a combination of areas E_i (Fig. S2.6d). For the point X_c^+, the critical point reached during loading, the elastic energy is

$$E_{\text{elastic}}(X_c^+) = E_1 + E_2 + E_3 + E_4 + E_5.$$

For the point X_r^+, the restabilization point reached just after the loading snapping transition, the elastic energy is

$$E_{\text{elastic}}(X_r^+) = E_{\text{elastic}}(X_c^+) - E_6 + E_9 + E_{10} + E_4.$$

Therefore, for the snapping transition upon loading to release elastic energy, we must have

$$0 < E_{\text{elastic}}(X_c^+) - E_{\text{elastic}}(X_r^+) \Leftrightarrow E_6 > E_4 + E_9 + E_{10}. \tag{S2.42}$$

Similarly, for the point X_c^-, the critical point reached during unloading, the elastic energy is

$$E_{\text{elastic}}(X_c^-) = E_1 + E_4 - E_8 - E_6 - E_7$$

For the point X_r^-, the restabilization point reached just after the unloading snapping transition, the elastic energy is

$$E_{\text{elastic}}(X_r^-) = E_1.$$

Therefore, for the snapping transition upon unloading to release elastic energy, we must have

$$0 < E_{\text{elastic}}(X_c^-) - E_{\text{elastic}}(X_r^-) \Leftrightarrow E_4 > E_6 + E_7 + E_8. \tag{S2.43}$$

Since $E_i > 0$, condition (S2.43) is incompatible with the energy release condition for the snapping transition upon loading (S2.42). This contradiction leads us to believe that a force-displacement curve whose secondary stable branch is shifted upwards and to the left with respect to the primary branch (as in Fig. S2.6a-c) cannot represent a physical system.

Nevertheless, countersnapping upon both loading and unloading could be achieved by allowing intersections between the two stable branches (Fig. S2.6e).

Such a curve respects the folding rules and ensures that both countersnapping instabilities at constant displacement are associated to an energy release, under some conditions, as we will demonstrate next.

Again, we can determine the elastic energy stored in the system just before and after the countersnapping instabilities, by decomposing the integral quantity defined in Eq. (S2.41) as a combination of areas E_i (Fig. S2.6f). We get the following conditions

$$0 < E_{\text{elastic}}(X_{\text{c}}^{+}) - E_{\text{elastic}}(X_{\text{r}}^{+}) \Leftrightarrow E_6 + E_4 + E_{14} + E_{15} > E_9 + E_{12} + E_{13}, \quad \text{(S2.44)}$$

and

$$0 < E_{\text{elastic}}(X_{\text{c}}^{-}) - E_{\text{elastic}}(X_{\text{r}}^{-}) \Leftrightarrow E_7 + E_9 + E_{10} + E_{12} > E_{14} + E_{15} + E_{16}. \quad \text{(S2.45)}$$

Conditions (S2.44) and (S2.45) can be satisfied simultaneously (like it is the case for the curve shown in Fig. S2.6e).

In conclusion, these results suggest that countersnapping upon both loading and unloading might be physically possible in a purely-elastic mechanical system. Such a system would necessarily be characterized by a force-displacement curve composed of two stable branches intersecting twice; similar to the curve shown in Fig. S2.6e. How to design and experimentally demonstrate a mechanical system characterized by this more complex curve is an open question that we leave for future research, as it requires additional constraints compared to a system that only countersnaps upon loading.

S2.6 Comparison of the countersnapping structure with the purely-series and purely-parallel assemblies

To demonstrate that the countersnapping behavior originates from a change from a series to parallel configuration, we compare the behavior of the full assembly with the behavior of (i) the same assembly minus the stiffening building blocks, that is the purely series configuration, and (ii) the same assembly minus the snapping building block only, that is the purely parallel configuration.

As it can be seen in Fig. S2.7, the behavior of the purely series assembly (up to its own critical point) approximates the primary stable branch of the full as-

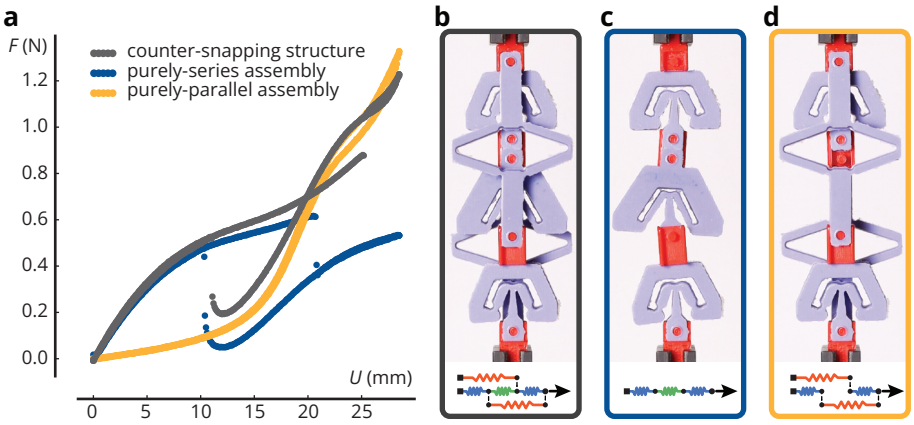


Figure S2.7: Comparison of the countersnapping structure with its purely-series and purely-parallel counterparts. **(a)** Force-displacement curves of the three building-block assemblies. **(b)** Countersnapping structure. **(c)** Purely-series assembly. **(d)** Purely-parallel assembly.

sembly, whereas the behavior of the purely parallel structure approximates the secondary stable branch. This observation supports the pivotal idea that the two stable configurations the structure can be in, and switch to, correspond to configurations predominantly behaving like series and parallel spring arrangements respectively.

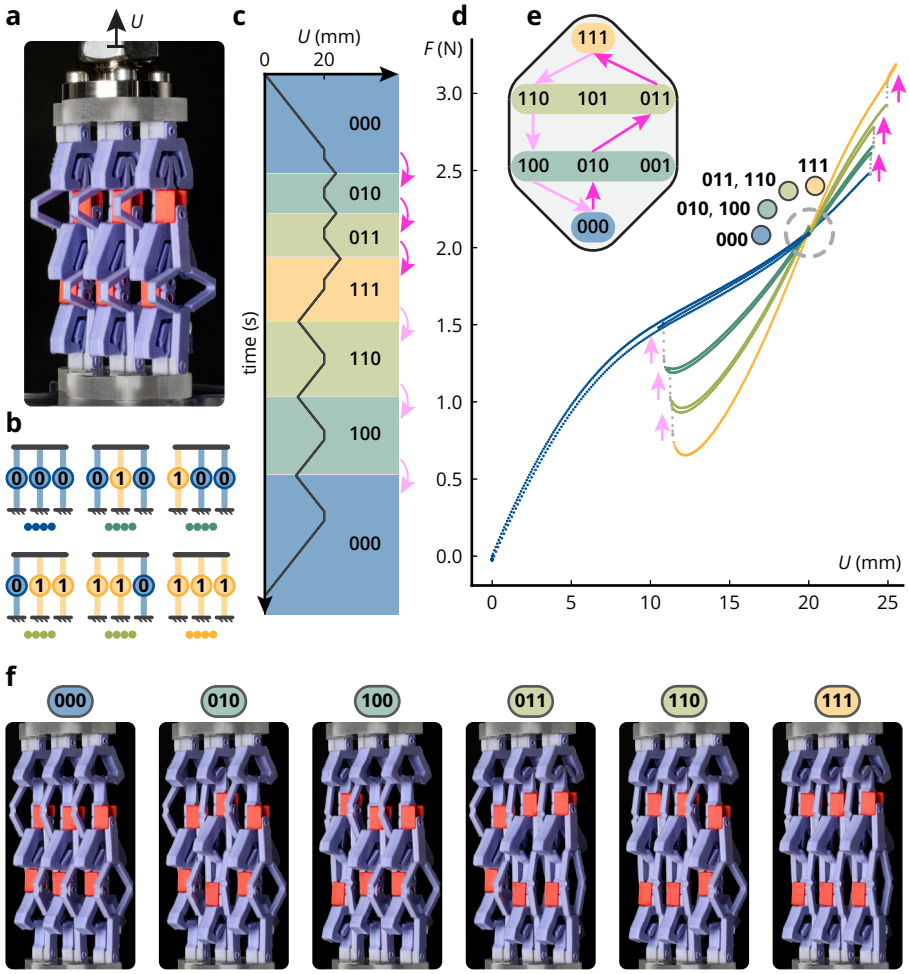


Figure S2.8: Experimentally observed behavior of three countersnapping elements combined in parallel. **(a)** Three parallel countersnapping elements controlled by displacement U. **(b)** Possible collective states of the system. **(c)** The binary state of the system (background color) as a function of U, where bright (light) pink arrows indicate snapping events during loading (unloading). **(d)** Corresponding force-displacement curve (colors indicate states). **(e)** Transition graph. **(f)** Snapshots of the states near the intersection point of the force-displacement curves.

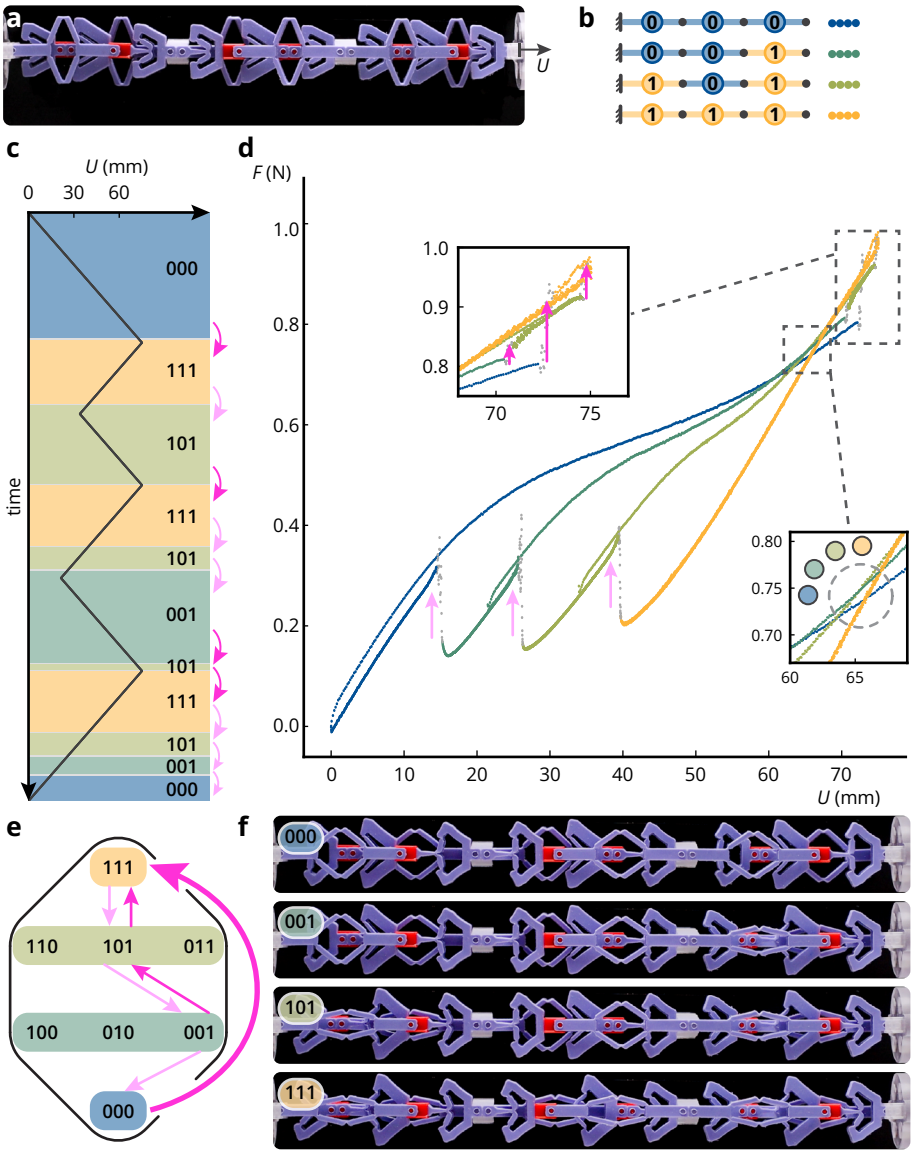
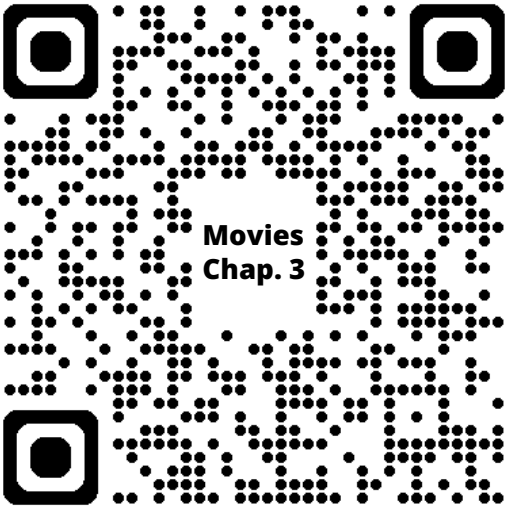


Figure S2.9: Experimentally observed behavior of three countersnapping elements combined in series. **(a)** Three serially-coupled countersnapping elements controlled by displacement U. **(b)** Possible collective states of the system. **(c)** The binary state of the system (background color) as a function of U, where bright (light) pink arrows indicate snapping events during loading (unloading). **(d)** Corresponding force-displacement curve (colors indicate states). **(e)** Transition graph. **(f)** Snapshots of the states near the intersection point of the force-displacement curves.

3

Simulating mechanical systems made from entities with arbitrarily complex responses

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 large 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 multivalued 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 a 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.*



Link to the Movies S3.1 to S3.4

3.1 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 [4, 46]. For example, compliant mechanisms leverage flexibility to mimic conventional mechanisms usually composed of many stiff components connected by joints [101]. Soft robots utilize shape changes to be intrinsically more robust, adaptable and autonomous, often by harnessing mechanical instabilities to embed sequenced [54], asymmetric [73], fast [21] or amplified [28] actuation. Recent mechanical metamaterials have exploited large geometric changes and nonlinearities to achieve increasingly complex deformation pathways, enabling multistability [102] or high energy dissipation [103]. 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 large 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 [101]. Trusses of bars can be used to model and even inverse design multistable compliant structures [104]. Cosserat rods have shown great promise for modeling soft robotic [105] or soft living [106] 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 [17, 32, 33, 50, 107–109].

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 multivalued 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 [52, 57, 110, 111].

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 multivalued 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 interactions 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.

3.2 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 nonmonotonic (Fig. 3.1a) or even a multivalued force-displacement curve (Fig. 3.1b), 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. 3.1c,d, the intrinsic nonlinear behavior of a flexel can be tuned to capture, and be equiva-

lent to, the geometric nonlinear behavior of compound structures, such as the Von-Mises trusses shown in Fig. 3.1a,b (Movie S3.1).

More formally, a flexel is defined as a deformable element whose elastic energy directly depends on its geometric measure α, 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 (Chapter S3, Section S3.1.2).

Flexels can be assembled to investigate how interactions between individual nonlinear elements give rise to more complex collective responses (Chapter S3, Section S3.1.2). For instance, coupling two flexels each characterized by a nonmonotonic and multivalued 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. 3.1e, Movie S3.1). 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. 3.1f, Movie S3.1). The more complex force-displacement responses exhibited by the assemblies shown in Fig. 3.1e,f can in turn be replicated by a single flexel (Fig. 3.1g,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. 3.1b is absorbed within its flexel abstraction (Fig. 3.1d) and cannot be coupled to other flexels. Similarly, the deformation of

the top node of the assembly shown in Fig. 3.1f along the horizontal direction cannot be coupled to other elements when using its flexel abstraction (Fig. 3.1h). 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 quasistatic loading involves solving a system of parametrized nonlinear equations, which our toolkit achieves by implementing the arclength method [112] (Chapter S3, Section S3.2). 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 arclength scheme is unable to retrieve equilibria disconnected from the initial path. Alternative methods could be implemented to cope with these two limitations [113].

3.3 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 α), 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. 3.2a). Second, the class of energy potentials is widened by supporting tunable, potentially multivalued generalized force-displacement curves (Fig. 3.2b). They can have an arbitrary number of turning points and intersections, allowing flexels to encode information about their loading history [52], or capture snapping or countersnapping phenomena [62], for example. This is achieved by defining a custom energy potential $v(\alpha, t)$ that produces equilibria on the prescribed multivalued force-displacement curve:

$$v(\alpha,t) = \frac{1}{2}k(t)\,(\alpha - \alpha_0 - a(t))^2 + b(t)(\alpha - \alpha_0 - a(t)) + \int_0^t b(\tilde{t})a'(\tilde{t})\mathrm{d}\tilde{t}, \quad (3.1)$$

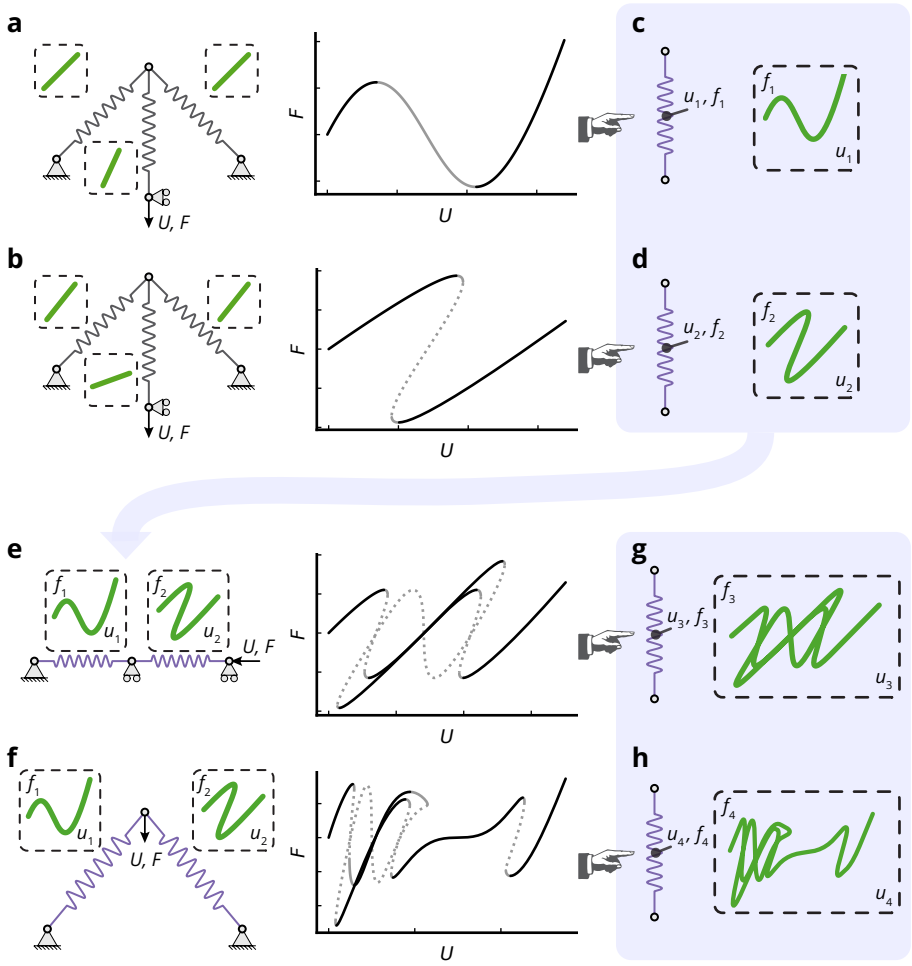


Figure 3.1: Construction of flexels. **(a-b)** Von-Mises truss composed of a pair of inclined linear springs (non-dimensional stiffness of 0.6 for (a), 1.0 for (b)) driven from their connection point through a third, vertical linear spring (stiffness of 20 for (a), 0.33 for (b)). The force-displacement response of the system is either nonmonotonic (a) or multivalued (b). **(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 multivalued 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 Chapter S3, Section S3.7.1.

where α_0 is the geometric measure at rest, $a(.)$ and $b(.)$ are the functions describing the (potentially multivalued) parametric force (b)-displacement (a) curve, and $k(.)$ is a function constructed from a and b (see Chapter S3, Section S3.4.2 for additional information, including the mathematical proof and stability considerations). 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 α (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. 3.2c, Chapter S3, Section S3.1, Movie S3.2). Let us give a few examples. Pairing an angle to a multivalued curve gives a flexel that models a flexure capable of snapping at constant angular displacement (Fig. 3.3a). 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. 3.3b). 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. 3.3c). 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. 3.3d).

3.4 Workflow

In this section we illustrate how our toolkit can be integrated to the workflow to design a structure with complex deformation pathways (Fig. 3.4, Movie S3.3). To illustrate this, we consider two physical structures fabricated in silicone rubber (Smooth-On, Smooth-Sil 945) [62], shown in Fig. 3.4a. 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. 3.4b). Second, for each experimental test, a Bezier curve of degree 4 is fitted to the experimental data by adjusting the po-

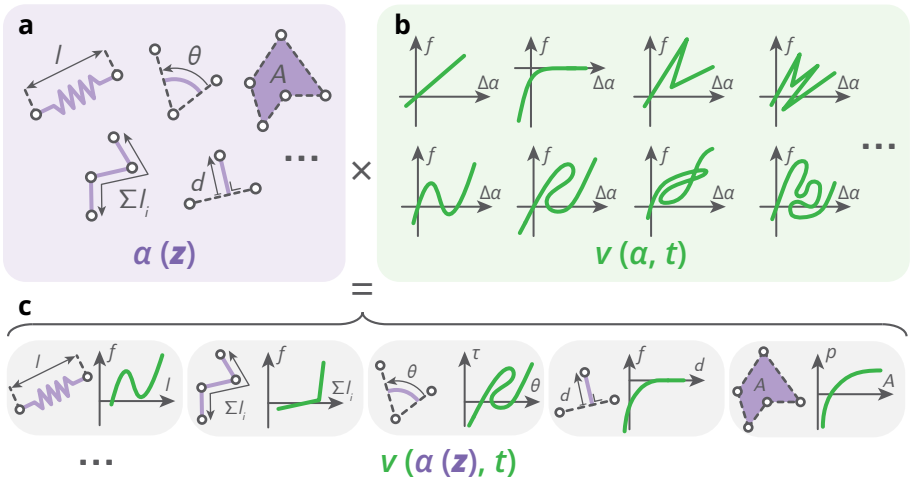


Figure 3.2: Flexel ecosystem. **(a)** Various geometric measures α computed from a list of node coordinates z_i: length, angle, area, total length of a polygonal chain, distance point-line. **(b)** Various generalized force-displacement curves (linear, nonlinear, multivalued) defining intrinsic mechanical 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 $v(\alpha(\boldsymbol{z}), t)$. More details on the definition of the geometric measures α and energy potentials v are provided in Chapter S3, Sections S3.3 and S3.4.

sitions of control points (Fig. 3.4c-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. 3.4c-right). Fourth, we model the assembly of the two structures by connecting the two flexels in series (Fig. 3.4d-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. 3.4d-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 files are provided in Chapter S3, Section S3.6.

Finally, we find that the simulated force-displacement curve is complex and characterized by many turning points (Fig. 3.4e). 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 quantita-

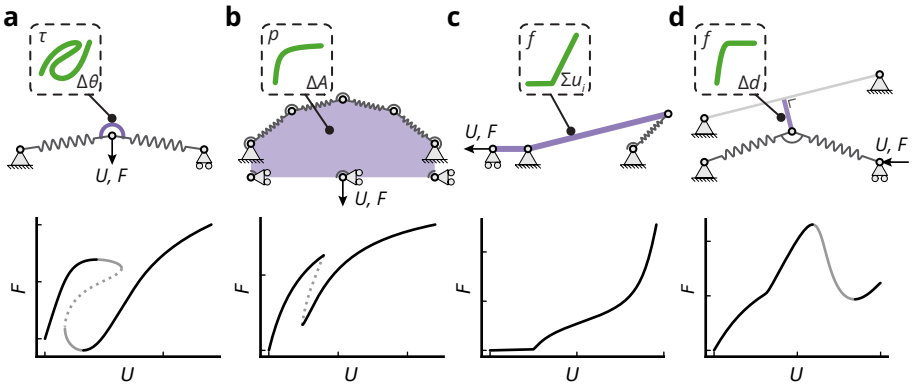


Figure 3.3: Examples of assemblies of flexels (top) that collectively produce force-displacement responses $F(U)$ (bottom). Black (gray) lines refer to states stable (unstable) under force-controlled conditions. Solid (dashed) lines refer to states stable (unstable) under displacement-driven conditions. Gray flexels are characterized by a linear generalized force-displacement curve. **(a)** An angular flexel with a multivalued 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 Chapter S3, Section S3.7.2.

tively and qualitatively (Fig. 3.4e). 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 [62].

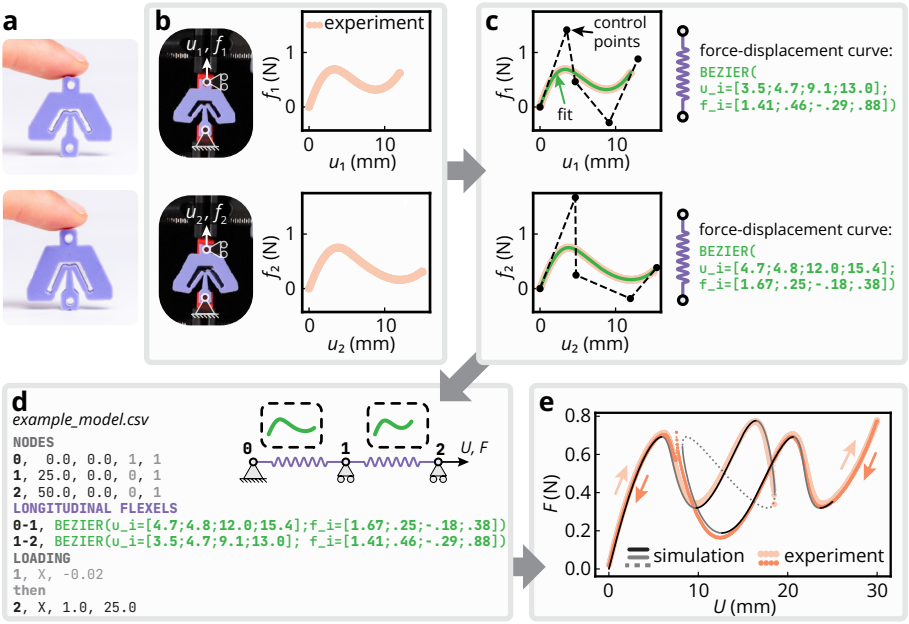


Figure 3.4: Workflow to simulate an experimental assembly of nonlinear building blocks [62]. **(a)** Two building blocks fabricated in silicone rubber. **(b)** Experimental force-displacement curves obtained by performing a tensile test on the building blocks. **(c)** Left: Bezier curves (solid green 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 Chapter S3, Section S3.7.3.

3.5 Use cases

Using this workflow, we next demonstrate the versatility of the ecosystem to model and handle complex use cases (Fig. 3.5, Movie S3.4). A full descriptions of all the models along with the script to reproduce the results is provided in Chapter S3, Section S3.7.4. As a first example, we show that stress-free or prestressed tensegrity trusses [114] can be effectively modeled by connecting longitudinal flexels with compatible or incompatible rest lengths. 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. 3.5a). This confirms that this truss is a pure tensegrity structure [114]. 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 [115] can be modeled using a multivalued angular flexel and a stiff path flexel. By loading the system in two steps, buckling and kink formation followed by zero-stiffness deformation can be simulated (Fig. 3.5b), reproducing the soft deformation mode used for gripping [115]. 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 [115].

Third, we show that a distance flexel can be used to model the contact between interacting buckled beams, reproducing the effect of the separating distance on the global buckling direction of pairs of "bumping beams" [116] (Fig. 3.5c). Unlike previous work studying similar systems using dynamic simulations [116, 117], 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 [118] (medium composed of collapsable capsules surrounded by a fluid) can be modeled and simulated using a single multivalued area flexel for the metafluid, linear angular and longitudinal flexels for the gripper, and distance flexel for the gripping contact. The simulation reveals that the metafluid modulates the gripping

force, allowing for delicate grasping as previously demonstrated experimentally in [118]. 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 [118], 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 [118]. 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.

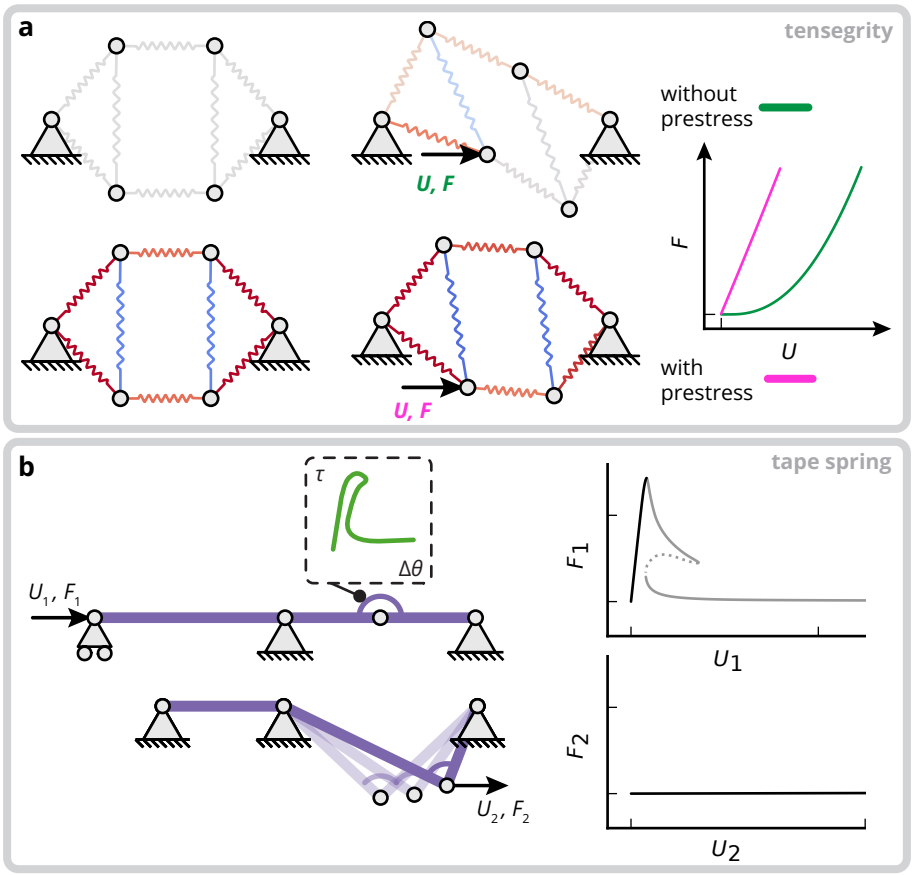
Together, these examples show the versatility of our approach. Importantly, each example only required a few lines of code to be simulated (Chapter S3, Section S3.7.4).

3.6 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 [119].

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 a 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 [59, 120], 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



panels c-d on the next page

Figure 3.5: 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 [115] 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) load steps. **(continuing on the next page)**

panels a-b on the previous page

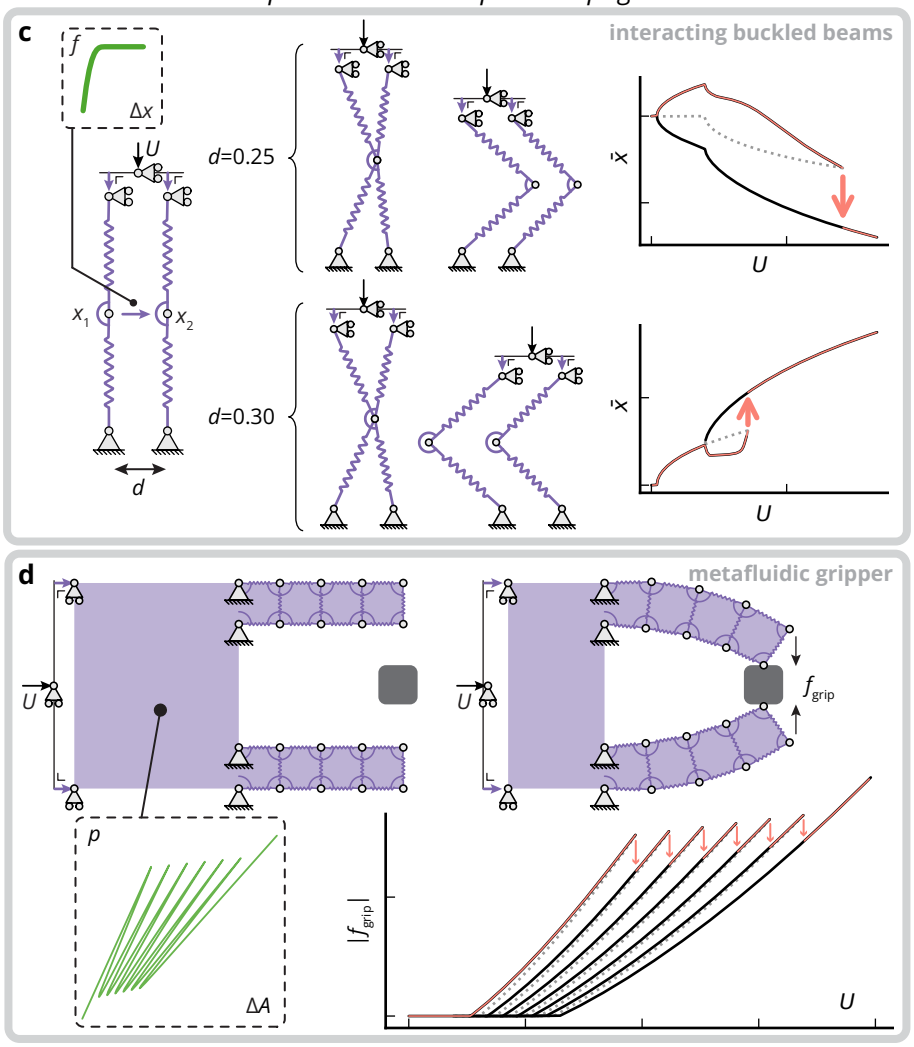


Figure 3.5 (continued): **(c)** Left: model of a pair of buckling beams separated by a distance d, loaded in compression and interacting through the contact of their middle node [89, 116, 117]. Center: deformation sequence of the beams when $d = 0.25$ (top) and $d = 0.30$ (bottom). Right: Deformation 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 [118] 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_{grip} as a function of the applied displacement U. The full descriptions of the models are provided in Chapter S3, Section S3.7.4.

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 [5]. 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 [94]. 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 a design paradigm for compliant mechanism, soft robots, mechanical metamaterials and nonlinear structures.

S3

Supplementary information for Chapter 3

S3.1 Flexel formulation

To simulate a mechanical structure quasistatically, the strategy consists of minimizing the total potential energy at varying load levels, as more formally described in Section S3.2.1. 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.

S3.1.1 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 geometric measure α, 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 α is computed from the spatial coordinates $\boldsymbol{z}$ of the nodes coupled by the flexel, using the equations derived in Section S3.3.

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:

$$e(\boldsymbol{q}) = \begin{cases} v(\alpha(\boldsymbol{z})) & \text{if } v \text{ is univariate,} \\ v(\alpha(\boldsymbol{z}), t) & \text{if } v \text{ is bivariate,} \end{cases} \tag{S3.1}$$

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

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 multivalued generalized force-displacement curves: equilibrium curves such that to a given geometric measure α (length, angle, area, etc) can correspond multiple generalized force values (force, torque, 2d-pressure, etc). This extra coordinate t disambiguates the state of the flexel, as the value of the geometric measure α 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 multivalued force-displacement curve (Eq. (S3.119)). More details on bivariate potentials and how to construct them from multivalued force-displacement curves are provided in Section S3.4.2.

Different types of univariate and bivariate energy potentials are implemented in `springable`. They are defined in Section S3.4.1 and S3.4.2 respectively.

Energy gradient (elastic force vector) The elastic force is described by the gradient of the energy e with respect to the coordinates $\boldsymbol{q}$. We will denote this

vector as $\boldsymbol{f}$, and its expression is given by

$$\begin{aligned}\boldsymbol{f} := \frac{\partial e}{\partial \boldsymbol{q}} = [f_i] &= \left[\frac{\partial e}{\partial q_i}\right] = \frac{\partial v}{\partial \alpha}\left[\frac{\partial \alpha}{\partial q_i}\right] + \frac{\partial v}{\partial t}\left[\frac{\partial t}{\partial q_i}\right] \\ &= \frac{\partial v}{\partial \alpha}\begin{bmatrix}\partial\alpha/\partial z_0 \\ \vdots \\ \partial\alpha/\partial z_{n-2} \\ 0\end{bmatrix} + \frac{\partial v}{\partial t}\begin{bmatrix}0 \\ \vdots \\ 0 \\ 1\end{bmatrix} \\ &= \frac{\partial v}{\partial \alpha}\begin{bmatrix}\partial\alpha/\partial \boldsymbol{z} \\ 0\end{bmatrix} + \frac{\partial v}{\partial t}\begin{bmatrix}\boldsymbol{0}^{(n-1)\times 1} \\ 1\end{bmatrix}\end{aligned} \tag{S3.2}$$

where v is a bivariate potential (where the n coordinates used by the flexel are the spatial ones and the internal one, i.e. $\boldsymbol{q} = [\boldsymbol{z}^\top, t]^\top$). In case of univariate potential (where $\partial v/\partial t = 0$ and all the n coordinates used by the flexel are spatial, i.e. $\boldsymbol{q} = \boldsymbol{z}$), Eq. (S3.2) simplifies to

$$\boldsymbol{f} = \frac{\partial v}{\partial \alpha}\left[\frac{\partial \alpha}{\partial z_i}\right] = \frac{\partial v}{\partial \alpha}\frac{\partial \alpha}{\partial \boldsymbol{z}}. \tag{S3.3}$$

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

- the measure α and its gradient $\partial\alpha/\partial\boldsymbol{z} = [\partial\alpha/\partial z_i]$, 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 S3.3;
- 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 S3.4.

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 $\boldsymbol{q}$ (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 $\boldsymbol{k}$. Its expression is given by

$$
\begin{aligned}
\boldsymbol{k} =& \frac{\partial^2 e}{\partial \boldsymbol{q} \partial \boldsymbol{q}^\top} = \left[k_{ij}\right] = \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] \\
&+ \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} & \cdots & \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} & \cdots & \frac{\partial \alpha}{\partial z_{n-2}}\frac{\partial \alpha}{\partial z_{n-2}} & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{\partial^2 v}{\partial \alpha \partial t}\begin{bmatrix} 0 & \cdots & 0 & \frac{\partial \alpha}{\partial z_0} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \frac{\partial \alpha}{\partial z_{n-2}} \\ \frac{\partial \alpha}{\partial z_0} & \cdots & \frac{\partial \alpha}{\partial z_{n-2}} & 0 \end{bmatrix} \\
&+ \frac{\partial v}{\partial \alpha}\begin{bmatrix} \frac{\partial^2 \alpha}{\partial z_0 \partial z_0} & \cdots & \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} & \cdots & \frac{\partial^2 \alpha}{\partial z_{n-2} \partial z_{n-2}} & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{\partial^2 v}{\partial t^2}\begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \\
=& \frac{\partial^2 v}{\partial \alpha^2}\begin{bmatrix} \frac{\partial \alpha}{\partial \boldsymbol{z}}\frac{\partial \alpha}{\partial \boldsymbol{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 \boldsymbol{z} \partial \boldsymbol{z}^\top} & \mathbf{0}^{(n-1)\times 1} \\ \mathbf{0}^{1\times(n-1)} & 0 \end{bmatrix} \\
&+ \frac{\partial^2 v}{\partial \alpha \partial t}\begin{bmatrix} \mathbf{0}^{(n-1)\times(n-1)} & \frac{\partial \alpha}{\partial \boldsymbol{z}} \\ \frac{\partial \alpha}{\partial \boldsymbol{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}
\end{aligned}
\tag{S3.4}
$$

where v is a bivariate potential (where the n coordinates used by the flexel are the spatial ones and the internal one, i.e. $\boldsymbol{q} = [\boldsymbol{z}^\top, t]^\top$). 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. $\boldsymbol{q} = \boldsymbol{z}$), Eq. (S3.4) simplifies to

$$\boldsymbol{k} = \frac{\partial^2 v}{\partial \alpha^2}\left[\frac{\partial \alpha}{\partial z_i}\frac{\partial \alpha}{\partial z_j}\right] + \frac{\partial v}{\partial \alpha}\left[\frac{\partial^2 \alpha}{\partial z_i \partial z_j}\right] = \frac{\partial^2 v}{\partial \alpha^2}\frac{\partial \alpha}{\partial \boldsymbol{z}}\frac{\partial \alpha}{\partial \boldsymbol{z}^\top} + \frac{\partial v}{\partial \alpha}\frac{\partial^2 \alpha}{\partial \boldsymbol{z}\partial \boldsymbol{z}^\top}. \tag{S3.5}$$

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, α, $\partial \alpha / \partial \boldsymbol{z} = [\partial \alpha / \partial z_i]$ and $\partial^2 \alpha / \partial \boldsymbol{z} \partial \boldsymbol{z}^\top = [\partial^2 \alpha / \partial z_i \partial z_j]$, 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 S3.3;
- 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 S3.4.

S3.1.2 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:

$$E(\boldsymbol{Q}) = \sum_{s=0}^{N_\mathrm{f}-1} e_s(\boldsymbol{q}^s), \tag{S3.6}$$

where $\boldsymbol{Q}$ 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. (S3.1)) and $\boldsymbol{q}^s \subseteq \boldsymbol{Q}$ 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 $\boldsymbol{Q}$. We will denote this

vector as $\boldsymbol{F}$, and its expression is given by

$$\boldsymbol{F} = \frac{\partial E}{\partial \boldsymbol{Q}} = \left[\frac{\partial E}{\partial Q_k}\right] = \sum_{s=0}^{N_\mathrm{f}-1}\left[\frac{\partial e_s(\boldsymbol{q}^s)}{\partial Q_k}\right] = \sum_{s=0}^{N_\mathrm{f}-1}\left(\sum_{i=0}^{n_s-1}\frac{\partial e_s}{\partial q_i^s}\left[\frac{\partial q_i^s}{\partial Q_k}\right]\right)$$
$$= \sum_{s=0}^{N_\mathrm{f}-1}\left(\sum_{i=0}^{n_s-1}\frac{\partial e_s}{\partial q_i^s}\left[L_{ik}^s\right]\right), \tag{S3.7}$$

where n_s is the number of coordinates in $\boldsymbol{q}^s$, $\partial e_s/\partial q_i^s$ are the components of the energy gradient $\boldsymbol{f}^s$ of flexel s (Eq. (S3.2)), and where

$$L_{ik}^s = \begin{cases} 1 & \text{if } q_i^s = Q_k, \\ 0 & \text{else.} \end{cases} \tag{S3.8}$$

In practical terms, Eq. (S3.7) means that the gradient $\boldsymbol{F} = [F_k]$ is constructed by adding together the components of each individual flexel's energy gradient $\boldsymbol{f}^s = [f_i^s] = \left[\partial e_s/\partial q_i^s\right]$ 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 $\boldsymbol{Q}$ (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 $\boldsymbol{K}$. Its expression is given by:

$$\boldsymbol{K} = \frac{\partial^2 E}{\partial \boldsymbol{Q}\partial \boldsymbol{Q}^\top} = [K_{kl}] = \left[\frac{\partial^2 E}{\partial Q_k \partial Q_l}\right] = \left[\frac{\partial F_k}{\partial Q_l}\right] = \sum_{s=0}^{N_\mathrm{f}-1}\left[\frac{\partial^2 e_s(\boldsymbol{q}^s)}{\partial Q_k \partial Q_l}\right]$$
$$= \sum_{s=0}^{N_\mathrm{f}-1}\left(\sum_{i=0}^{n_s-1}\sum_{j=0}^{n_s-1}\frac{\partial^2 e_s}{\partial q_i^s \partial q_j^s}\left[\frac{\partial q_i^s}{\partial Q_k}\frac{\partial q_j^s}{\partial Q_l}\right]\right) = \sum_{s=0}^{N_\mathrm{f}-1}\left(\sum_{i=0}^{n_s-1}\sum_{j=0}^{n_s-1}\frac{\partial^2 e_s}{\partial q_i^s \partial q_j^s}\left[L_{ik}^s L_{jl}^s\right]\right), \tag{S3.9}$$

where $\partial^2 e_s/\partial q_i^s \partial q_j^s$ are the components of the energy hessian of flexel s (Eq. (S3.4)), and where $[L_{ik}L_{jl}]$ is determined using Eq. (S3.8). In practical terms, Eq. (S3.9) means that the energy hessian $\boldsymbol{K} = [K_{kl}]$ is constructed by adding together the components of each individual flexel's energy hessian $\boldsymbol{k}^s = [k_{ij}^s] = [\partial e_s/\partial q_i^s \partial q_j^s]$ that correspond to the same pair of coordinates (Q_k, Q_l).

S3.2 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.

S3.2.1 Problem formulation

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 $\boldsymbol{Q}$. Among these N coordinates, M are free, while the others are fixed. The free coordinates are denoted by $\tilde{\boldsymbol{Q}}$, which is the array $\boldsymbol{Q}$ 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 $\tilde{\boldsymbol{Q}}$. To this end, we first consider the total potential energy Π of the system [2]:

$$\Pi = E - \tilde{\boldsymbol{F}}^{\text{ext}} \cdot \tilde{\boldsymbol{U}} = E - \sum_{k=0}^{M-1} \tilde{F}_k^{\text{ext}} \tilde{U}_k, \tag{S3.10}$$

where E is the elastic energy stored in the system and $\tilde{\boldsymbol{F}}^{\text{ext}} \cdot \tilde{\boldsymbol{U}}$ 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 $\tilde{\boldsymbol{F}}^{\text{ext}}$ are stationary points of the total potential energy with respect to the free coordi-

nates $\tilde{\boldsymbol{Q}}$ [2], they are solutions of the following system of M equations

$$\frac{\partial E}{\partial \tilde{\boldsymbol{Q}}}(\boldsymbol{Q}) - \tilde{\boldsymbol{F}}^{\text{ext}} = 0 \tag{S3.11}$$

$$\Leftrightarrow$$

$$\begin{cases} \dfrac{\partial E}{\partial \tilde{Q}_0}(\boldsymbol{Q}) - \tilde{F}_0^{\text{ext}} = 0 \\ \vdots \\ \dfrac{\partial E}{\partial \tilde{Q}_k}(\boldsymbol{Q}) - \tilde{F}_k^{\text{ext}} = 0 \\ \vdots \\ \dfrac{\partial E}{\partial \tilde{Q}_{M-1}}(\boldsymbol{Q}) - \tilde{F}_{M-1}^{\text{ext}} = 0. \end{cases} \tag{S3.12}$$

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 $\tilde{\boldsymbol{F}}^{\text{ext}}$. 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 $\tilde{\boldsymbol{F}}^{\text{ext,0}}$ and $\tilde{\boldsymbol{F}}^{\text{ext, final}}$), the system can be rewritten in the following parametrized form

$$\frac{\partial E}{\partial \tilde{\boldsymbol{Q}}}(\boldsymbol{Q}) - \tilde{\boldsymbol{F}}^{\text{ext}} = \frac{\partial E}{\partial \tilde{\boldsymbol{Q}}}(\boldsymbol{Q}) - (\tilde{\boldsymbol{F}}^{\text{ext,0}} + \lambda \tilde{\boldsymbol{F}}^{\text{dir}}) = 0 \tag{S3.13}$$

$$\Leftrightarrow$$

$$\begin{cases} \dfrac{\partial E}{\partial \tilde{Q}_0}(\boldsymbol{Q}) - (\tilde{F}_0^{\text{ext,0}} + \lambda \tilde{F}_0^{\text{dir}}) = 0 \\ \vdots \\ \dfrac{\partial E}{\partial \tilde{Q}_k}(\boldsymbol{Q}) - (\tilde{F}_k^{\text{ext,0}} + \lambda \tilde{F}_k^{\text{dir}}) = 0 \\ \vdots \\ \dfrac{\partial E}{\partial \tilde{Q}_{M-1}}(\boldsymbol{Q}) - (\tilde{F}_{M-1}^{\text{ext,0}} + \lambda \tilde{F}_{M-1}^{\text{dir}}) = 0. \end{cases}$$

where

$$\tilde{\boldsymbol{F}}^{\text{dir}} := \frac{\tilde{\boldsymbol{F}}^{\text{step}}}{||\tilde{\boldsymbol{F}}^{\text{step}}||} \tag{S3.14}$$

is the direction of the step load vector $\tilde{\boldsymbol{F}}^{\text{step}} := \tilde{\boldsymbol{F}}^{\text{ext, final}} - \tilde{\boldsymbol{F}}^{\text{ext,0}}$ (which represents the additional load needed to reach the final prescribed load from the initial load), and where λ is the load parameter, which is initially 0 and reaches $||\tilde{\boldsymbol{F}}^{\text{step}}||$ when the load step is completed. Eq. (S3.13) represents a system of parametrized nonlinear equations, which can be solved using numerical continuation algorithms.

S3.2.2 Algorithm

In `springable`, we solve Eq. (S3.13) using the arclength continuation scheme introduced in [98,112], 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 λ and applying Newton's method. The algorithm consists of computing successive increments, each performing one prediction, followed by multiple corrections to converge toward 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 $\boldsymbol{Q}^0$, initial external forces $\tilde{\boldsymbol{F}}^{\text{ext,0}}$, step load vector $\tilde{\boldsymbol{F}}^{\text{step}}$ (defined as $\tilde{\boldsymbol{F}}^{\text{ext, final}} - \tilde{\boldsymbol{F}}^{\text{ext,0}}$).

2. The settings of the solver are provided: initial radius r_0, convergence tolerance ε.

3. The direction of the step load vector is computed: $\tilde{\boldsymbol{F}}^{\text{dir}} := \tilde{\boldsymbol{F}}^{\text{step}}/||\tilde{\boldsymbol{F}}^{\text{step}}||$.

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 $\boldsymbol{Q} \leftarrow \boldsymbol{Q}^0$,
   - current external force vector $\tilde{\boldsymbol{F}}^{\text{ext}} \leftarrow \tilde{\boldsymbol{F}}^{\text{ext,0}}$,
   - list of previously computed equilibrium states $\boldsymbol{Q}^{\text{eq}} \leftarrow [\boldsymbol{Q}^0]$,
   - list of external force vectors corresponding to the previously computed equilibrium states $\tilde{\mathcal{F}}^{\text{eq}} \leftarrow [\tilde{\boldsymbol{F}}^{\text{ext,0}}]$,
   - the current hessian matrix $\boldsymbol{K}$ is computed based on the coordinates $\boldsymbol{Q}^0$ (Eq. (S3.9)), the current reduced hessian matrix $\tilde{\boldsymbol{K}}$ 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: $\Delta \boldsymbol{U}^{\text{inc}} = \boldsymbol{0}$, $\Delta\lambda^{\text{inc}} = 0$.

      ii. Eq. (S3.15) is solved for $\Delta\hat{\boldsymbol{U}}$:

$$\tilde{\boldsymbol{K}}\Delta\hat{\boldsymbol{U}} = \tilde{\boldsymbol{F}}^{\text{dir}}. \tag{S3.15}$$

      iii. The quantity $\Delta\lambda^{\text{ite}}$ is computed [99]:

$$\Delta\lambda^{\text{ite}} = s\frac{r}{\sqrt{\Delta\hat{\boldsymbol{U}} \cdot \Delta\hat{\boldsymbol{U}}}}, \tag{S3.16}$$

      with

$$s = \begin{cases} 1 & \text{if first increment, or if } \Delta\hat{\boldsymbol{U}} \cdot \Delta\boldsymbol{U}^{\text{prev. inc}} \geq 0 \\ -1 & \text{else.} \end{cases} \tag{S3.17}$$

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

$$\Delta\boldsymbol{U}^{\text{ite}} = \Delta\lambda^{\text{ite}}\Delta\hat{\boldsymbol{U}}. \tag{S3.18}$$

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

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

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

$$\tilde{\boldsymbol{R}} = \tilde{\boldsymbol{F}} - \tilde{\boldsymbol{F}}^{\text{ext}}, \tag{S3.19}$$

      where $\tilde{\boldsymbol{F}}$ is the array $\boldsymbol{F} := \partial E/\partial \boldsymbol{Q}$ (Eq. (S3.7)) from which the components corresponding to fixed coordinates have been removed. The array $\boldsymbol{F}$ represents the internal net force resulting on each coordinate due to the elastic forces of the flexels,

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

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

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

i. The current hessian matrix $\boldsymbol{K}$ is computed based on the updated coordinates $\boldsymbol{Q}$ (see Eq. (S3.9), and the current reduced hessian matrix $\tilde{\boldsymbol{K}}$ is formed by removing the rows and columns corresponding to the fixed coordinates.

ii. Eqs. (S3.20) and (S3.21) are solved for $\Delta\hat{\boldsymbol{U}}$ and $\Delta\bar{\boldsymbol{U}}$ respectively:

$$\tilde{\boldsymbol{K}}\Delta\hat{\boldsymbol{U}} = \tilde{\boldsymbol{F}}^{\text{dir}}, \tag{S3.20}$$

$$\tilde{\boldsymbol{K}}\Delta\bar{\boldsymbol{U}} = -\tilde{\boldsymbol{R}}. \tag{S3.21}$$

iii. The following quantities are computed [99]:

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

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:

$$\Delta\lambda^{\text{ite}} = \begin{cases} \dfrac{-b+\sqrt{\rho}}{2a} & \text{if } b_0 > 0, \\ \dfrac{-b-\sqrt{\rho}}{2a} & \text{else.} \end{cases} \tag{S3.22}$$

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

$$\Delta\boldsymbol{U}^{\text{ite}} \leftarrow \Delta\bar{\boldsymbol{U}} + \Delta\lambda^{\text{ite}}\Delta\hat{\boldsymbol{U}}. \tag{S3.23}$$

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

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

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

$$\tilde{\boldsymbol{R}} = \tilde{\boldsymbol{F}} - \tilde{\boldsymbol{F}}^{\text{ext}}, \tag{S3.24}$$

where $\tilde{\boldsymbol{F}}$ is the array $\boldsymbol{F} := \partial E/\partial \boldsymbol{Q}$ (Eq. (S3.7)) from which the components corresponding to the fixed coordinates have been removed.

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

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

(c) The next increment is prepared.

i. The current hessian matrix $\boldsymbol{K}$ is computed based on the coordinates $\boldsymbol{Q}$ (Eq. (S3.9)), the current reduced hessian matrix $\tilde{\boldsymbol{K}}$ is formed by removing the rows and columns corresponding to the fixed coordinates.

ii. The following quantities are updated: $\Delta\boldsymbol{U}^{\text{prev. inc}} \leftarrow \Delta\boldsymbol{U}^{\text{inc}}$, 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 $\boldsymbol{Q}$ is appended to the list $\boldsymbol{Q}^{\text{eq}}$ of previously computed equilibrium states: $\boldsymbol{Q}^{\text{eq}} \leftarrow \boldsymbol{Q}^{\text{eq}} + [\boldsymbol{Q}]$.

v. The new external force vector $\tilde{\boldsymbol{F}}^{\text{ext}}$ is appended to the list $\tilde{\mathcal{F}}^{\text{eq}}$ of external force vectors corresponding to previously computed equilibrium states: $\tilde{\mathcal{F}}^{\text{eq}} \leftarrow \tilde{\mathcal{F}}^{\text{eq}} + [\tilde{\boldsymbol{F}}^{\text{ext}}]$.

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:

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

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

i. The array of coordinates $\boldsymbol{Q}$ is reset to the last item in the list $\boldsymbol{Q}^{\text{eq}}$ of previously computed equilibrium states: $\boldsymbol{Q} \leftarrow \boldsymbol{Q}^{\text{eq}}[\text{last}]$.

ii. The external force vector is reset to the last item in the list $\tilde{\boldsymbol{F}}^{\text{ext}}$ of external force vectors corresponding to the previously computed equilibrium states : $\tilde{\boldsymbol{F}}^{\text{ext}} \leftarrow \tilde{\mathcal{F}}^{\text{eq}}[\text{last}]$.

iii. The current hessian matrix $\boldsymbol{K}$ is computed based on the coordinates $\boldsymbol{Q}$ (Eq. (S3.9)), the current reduced hessian matrix $\tilde{\boldsymbol{K}}$ 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.

S3.3 Geometric measures, gradients and hessians

When computing the gradient and hessian of the elastic energy, necessary for the arclength 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. S3.1.

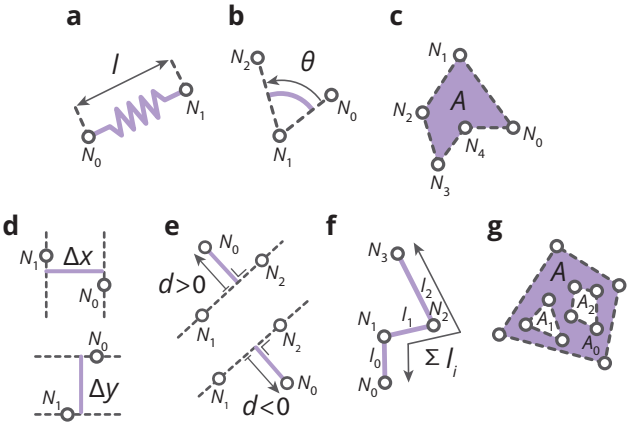


Figure S3.1: Geometric measures. **(a)** Length. **(b)** Angle. **(c)** Area. **(d)** Signed x- (top) and signed y- (bottom) distances. **(e)** Signed point-line distance, which can be positive (top) or negative (bottom). **(f)** Path length. **(g)** Area with holes.

Notation:

- α: 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,
- $\boldsymbol{z}$: array of spatial nodal coordinates,
- z_i: $(i+1)$-th component in the array $\boldsymbol{z}$ (so, z_0 is the first component).

S3.3.1 Length (longitudinal flexel)

A longitudinal flexel is defined by two nodes, node N_0 and node N_1 (Fig. S3.1a). The measure α of a longitudinal flexel is the length l of the segment connecting the two nodes. In 2d, this length depends on the four coordinates $\boldsymbol{z} = [x_0, y_0, x_1, y_1]$.

MEASURE α

$$\alpha = l(\boldsymbol{z}) = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \quad . \tag{S3.25}$$

GRADIENT $\partial\alpha/\partial\boldsymbol{z}$

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = \left[\frac{\partial \alpha}{\partial z_i}\right] = \begin{bmatrix} \frac{\partial l}{\partial t_0} & \frac{\partial l}{\partial y_0} & \frac{\partial l}{\partial t_1} & \frac{\partial l}{\partial y_1} \end{bmatrix}^\top = \frac{1}{l}\begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \end{bmatrix}. \tag{S3.26}$$

HESSIAN $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$

$$\begin{aligned} \frac{\partial^2 \alpha}{\partial \boldsymbol{z} \partial \boldsymbol{z}^\top} &= \left[\frac{\partial^2 \alpha}{\partial z_i \partial z_j}\right] \\ &= \frac{1}{l}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} - \frac{1}{l^2}\frac{1}{l}\begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \end{bmatrix}\begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \end{bmatrix}^\top \\ &= \frac{1}{l}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} - \frac{1}{l^3}\begin{bmatrix} \Delta x \Delta x & \Delta x \Delta y & -\Delta x \Delta x & -\Delta x \Delta y \\ \Delta x \Delta y & \Delta y \Delta y & -\Delta x \Delta y & -\Delta y \Delta y \\ -\Delta x \Delta x & -\Delta x \Delta y & \Delta x \Delta x & \Delta x \Delta y \\ -\Delta x \Delta y & -\Delta y \Delta y & \Delta x \Delta y & \Delta y \Delta y \end{bmatrix}, \end{aligned} \tag{S3.27}$$

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

S3.3.2 Angle (angular flexel)

An angular flexel is defined by three nodes: node N_0, node N_1 and node N_2 (Fig. S3.1b). The measure α of an angular flexel is the angle θ 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 $\overrightarrow{N_1N_0}$ must rotate counter-clockwise about N_1 to align with the vector $\overrightarrow{N_1N_2}$. This angle depends on the six coordinates $\boldsymbol{z} = [x_0, y_0, x_1, y_1, x_2, y_2]^\top$.

MEASURE α

$$\alpha = \theta(\boldsymbol{z}) = \operatorname{atan2}\left(Y(\boldsymbol{z}), X(\boldsymbol{z})\right) \mod 2\pi, \tag{S3.28}$$

where

$$Y(\boldsymbol{z}) = (x_1 - x_0)(y_1 - y_2) - (y_1 - y_0)(x_1 - x_2), \tag{S3.29}$$

and

$$X(\boldsymbol{z}) = (x_1 - x_0)(x_1 - x_2) + (y_1 - y_0)(y_1 - y_2). \tag{S3.30}$$

GRADIENT $\partial\alpha/\partial\boldsymbol{z}$

$$\frac{\partial\alpha}{\partial\boldsymbol{z}} = \left[\frac{\partial\alpha}{\partial z_i}\right] = \left[\frac{\partial\theta}{\partial z_i}\right] = \left[\frac{\partial\theta}{\partial Y}\frac{\partial Y}{\partial z_i} + \frac{\partial\theta}{\partial X}\frac{\partial X}{\partial z_i}\right] = \frac{\partial\theta}{\partial Y}\left[\frac{\partial Y}{\partial z_i}\right] + \frac{\partial\theta}{\partial X}\left[\frac{\partial X}{\partial z_i}\right], \tag{S3.31}$$

where

$$\left[\frac{\partial Y}{\partial z_i}\right] = \begin{bmatrix} y_2 - y_1 \\ x_1 - x_2 \\ y_0 - y_2 \\ x_2 - x_0 \\ y_1 - y_0 \\ x_0 - x_1 \end{bmatrix}, \tag{S3.32}$$

$$\left[\frac{\partial X}{\partial z_i}\right] = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ 2x_1 - x_0 - x_2 \\ 2y_1 - y_0 - y_2 \\ x_0 - x_1 \\ y_0 - y_1 \end{bmatrix}, \tag{S3.33}$$

$$\frac{\partial\theta}{\partial Y} = \frac{X}{X^2 + Y^2} \quad , \tag{S3.34}$$

and

$$\frac{\partial\theta}{\partial X} = -\frac{Y}{X^2 + Y^2} \quad . \tag{S3.35}$$

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

$$\begin{aligned}\frac{\partial^2\alpha}{\partial\boldsymbol{z}\partial\boldsymbol{z}^\top} &= \left[\frac{\partial^2\alpha}{\partial z_i\partial z_j}\right] \\ &= \frac{\partial\theta}{\partial Y}\left[\frac{\partial^2 Y}{\partial z_i\partial z_j}\right] + \frac{\partial\theta}{\partial X}\left[\frac{\partial^2 X}{\partial z_i\partial z_j}\right] \\ &\quad + \frac{\partial^2\theta}{\partial Y^2}\left[\frac{\partial Y}{\partial z_i}\frac{\partial Y}{\partial z_j}\right] + \frac{\partial^2\theta}{\partial Y\partial X}\left[\frac{\partial Y}{\partial z_i}\frac{\partial X}{\partial z_j} + \frac{\partial X}{\partial z_i}\frac{\partial Y}{\partial z_j}\right] + \frac{\partial^2\theta}{\partial X^2}\left[\frac{\partial X}{\partial z_i}\frac{\partial X}{\partial z_j}\right],\end{aligned} \tag{S3.36}$$

where

$$\left[\frac{\partial^2 Y}{\partial z_i\partial z_j}\right] = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}, \tag{S3.37}$$

$$\left[\frac{\partial^2 X}{\partial z_i\partial z_j}\right] = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}, \tag{S3.38}$$

$$\frac{\partial^2\theta}{\partial Y^2} = -\frac{2XY}{(X^2+Y^2)^2}\quad, \tag{S3.39}$$

$$\frac{\partial^2\theta}{\partial Y\partial X} = \frac{Y^2-X^2}{(X^2+Y^2)^2}\quad, \tag{S3.40}$$

and

$$\frac{\partial^2\theta}{\partial X^2} = \frac{2XY}{(X^2+Y^2)^2}\quad. \tag{S3.41}$$

S3.3.3 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

Section S3.3.4 and S3.3.7). A signed area A_s is defined by $n \geq 3$ nodes: N_k, $k = 0, 1, \ldots, n-1$, 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 $\boldsymbol{z} = [x_0, y_0, \ldots, x_{n-1}, y_{n-1}]^\top$

Measure α

$$\alpha = A_s(\boldsymbol{z}) = \frac{1}{2} \sum_{k=0}^{n-1} (x_k y_{k^+} - y_k x_{k^+}) \tag{S3.42}$$

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

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

$$\frac{\partial \alpha}{\partial \boldsymbol{z}} = \left[\frac{\partial \alpha}{\partial z_i}\right] = \left[\frac{\partial A_s}{\partial z_i}\right] = \frac{1}{2}\left[S_i\right], \tag{S3.43}$$

where

$$S_i = \begin{cases} y_{k^+} - y_{k^-} & \text{if } i \text{ is even} \Leftrightarrow z_i \text{ is a } x\text{-coordinate} \\ \text{with } k = i/2 & \\ & \\ x_{k^-} - x_{k^+} & \text{if } i \text{ is odd} \Leftrightarrow z_i \text{ is a } y\text{-coordinate} \\ \text{with } k = i/2 - 1/2, & \end{cases} \tag{S3.44}$$

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

$$[S_i] = \begin{bmatrix} y_1 - y_{n-1} \\ x_{n-1} - x_1 \\ y_2 - y_0 \\ x_0 - x_2 \\ \vdots \\ y_0 - y_{n-2} \\ x_{n-2} - x_0 \end{bmatrix}. \tag{S3.45}$$

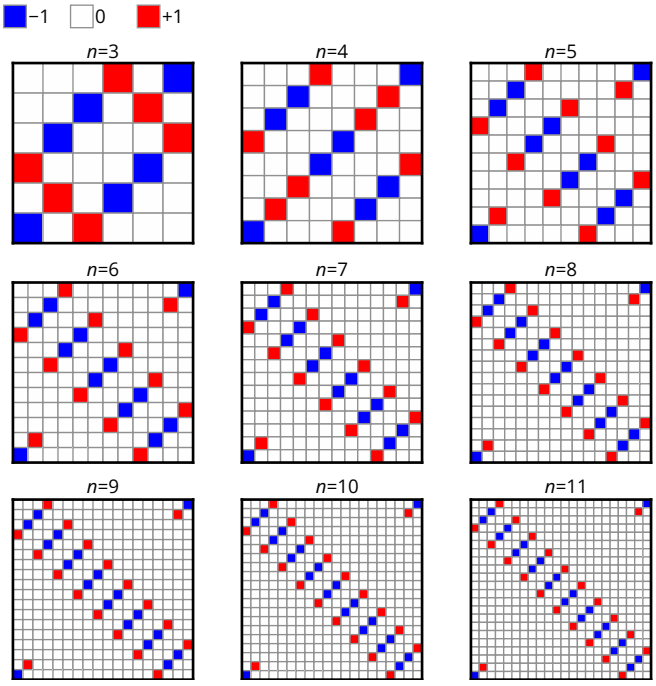


Figure S3.2: $\left[T_{ij}\right]$ matrices for $n = 3$ to 11.

HESSIAN $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$

$$\frac{\partial^2\alpha}{\partial\boldsymbol{z}\partial\boldsymbol{z}^\top} = \left[\frac{\partial^2\alpha}{\partial z_i \partial z_j}\right] = \frac{1}{2}\left[T_{ij}\right], \tag{S3.46}$$

where

$$T_{ij} = \begin{cases} +1 & \text{if } i \text{ is even and } j = (i+3) \mod 2n \\ -1 & \text{if } i \text{ is even and } j = (i-1) \mod 2n \\ -1 & \text{if } i \text{ is odd and } j = (i+1) \mod 2n \\ +1 & \text{if } i \text{ is odd and } j = (i-3) \mod 2n \\ 0 & \text{else.} \end{cases} \tag{S3.47}$$

Matrices $\left[T_{ij}\right]$ for $n \in [3, 11]$ are shown in Fig. S3.2.

S3.3.4 Area (area flexel)

An area flexel is defined by $n \geq 3$ nodes: $N_0, N_1, \ldots, N_{n-1}$ (Fig. S3.1c). The measure α of an area flexel is the area $A \geq 0$ of the polygon whose vertices are 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 $\boldsymbol{z} = [x_0, y_0, \ldots, x_{n-1}, y_{n-1}]$.

MEASURE α

$$\alpha = A(\boldsymbol{z}) = |A_\mathrm{s}(\boldsymbol{z})|\,, \tag{S3.48}$$

where A_s is the signed-area measure, defined in Eq. (S3.42).

GRADIENT $\partial\alpha/\partial\boldsymbol{z}$

$$\frac{\partial\alpha}{\partial\boldsymbol{z}} = \left[\frac{\partial\alpha}{\partial z_i}\right] = \mathrm{sgn}(A_\mathrm{s})\left[\frac{\partial A_\mathrm{s}}{\partial z_i}\right] = \frac{1}{2}\,\mathrm{sgn}(A_\mathrm{s})\,[S_i]\,, \tag{S3.49}$$

where $[S_i]$ is the array defined in Eq. (S3.44).

HESSIAN $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$

$$\frac{\partial^2\alpha}{\partial\boldsymbol{z}\partial\boldsymbol{z}^\top} = \left[\frac{\partial^2\alpha}{\partial z_i\partial z_j}\right] = \mathrm{sgn}(A_\mathrm{s})\left[\frac{\partial^2 A_\mathrm{s}}{\partial z_i\partial z_j}\right] = \frac{1}{2}\,\mathrm{sgn}(A_\mathrm{s})\left[T_{ij}\right], \tag{S3.50}$$

where $\left[T_{ij}\right]$ is the matrix defined in Eq. (S3.47).

S3.3.5 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. S3.1d). The measure of such a 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 $\boldsymbol{z} = [x_0, y_0, x_1, y_1]^\top$.

Measure α

$$\alpha = \Delta x(\boldsymbol{z}) = x_0 - x_1 \quad \text{(resp.} \quad \alpha = \Delta y(\boldsymbol{z}) = y_0 - y_1\text{).} \tag{S3.51}$$

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

$$\frac{\partial\alpha}{\partial\boldsymbol{z}} = [1, 0, -1, 0]^\top \quad \text{(resp.} \quad \frac{\partial\alpha}{\partial\boldsymbol{z}} = [0, 1, 0, -1]^\top\text{).} \tag{S3.52}$$

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

$$\frac{\partial^2\alpha}{\partial\boldsymbol{z}\partial\boldsymbol{z}^\top} = \mathbf{0}^{4\times4}. \tag{S3.53}$$

S3.3.6 Compound measure

A flexel's measure α can be expressed as a function C of p other measures α_k, $k = 0, 1, \ldots, p-1$. Each measure α_k itself depends on m_k coordinates $\boldsymbol{z}^k$. In general, it is possible that measures α_k share some of their coordinates; therefore the number m of coordinates that the measure α depends on is less than $\sum_{k=0}^{p-1} m_k$. The array $\boldsymbol{z}$ gathers the set of unique coordinates collected across each $\boldsymbol{z}^k$.

Measure α

$$\alpha = C(\alpha_0, \alpha_1, \ldots, \alpha_{p-1}). \tag{S3.54}$$

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

$$\begin{aligned}\frac{\partial\alpha}{\partial\boldsymbol{z}} &= \left[\frac{\partial\alpha}{\partial z_i}\right] = \sum_{k=0}^{p-1} \frac{\partial C}{\partial\alpha_k}\left[\frac{\partial\alpha_k(\boldsymbol{z}^k)}{\partial z_i}\right] = \sum_{k=0}^{p-1}\sum_{r=0}^{m_k-1} \frac{\partial C}{\partial\alpha_k}\frac{\partial\alpha_k}{\partial z_r^k}\left[\frac{\partial z_r^k}{\partial z_i}\right] \\ &= \sum_{k=0}^{p-1}\sum_{r=0}^{m_k-1} \frac{\partial C}{\partial\alpha_k}\frac{\partial\alpha_k}{\partial z_r^k}\left[L_{ir}^k\right],\end{aligned} \tag{S3.55}$$

where $\left[\partial\alpha_k/\partial z_r^k\right] = \partial\alpha_k/\partial\boldsymbol{z}^k$ is the gradient of the measure α_k in its "own" coordinates $\boldsymbol{z}^k$, and

$$L_{ir}^k = \begin{cases} 1 & \text{if } z_i = z_r^k \\ 0 & \text{else.} \end{cases} \tag{S3.56}$$

In practical terms, Eq. (S3.55) means that the gradient $\partial\alpha/\partial\boldsymbol{z} = [\partial\alpha/\partial z_i]$ of a compound measure is constructed by summing together the components of each individual measure's gradient $\partial\alpha_k/\partial\boldsymbol{z}^k$ that correspond to the same coordinate z_i, weighted by $\partial C_k/\partial\alpha_k$.

HESSIAN $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$

$$\begin{aligned}\frac{\partial^2\alpha}{\partial\boldsymbol{z}\partial\boldsymbol{z}^\top} &= \left[\frac{\partial^2\alpha}{\partial z_i\partial z_j}\right] = \sum_{k=0}^{p-1}\sum_{l=0}^{p-1}\frac{\partial^2 C}{\partial\alpha_k\partial\alpha_l}\left[\frac{\partial\alpha_k(\boldsymbol{z}^k)}{\partial z_i}\frac{\partial\alpha_l(\boldsymbol{z}^l)}{\partial z_j}\right] + \sum_{k=0}^{p-1}\frac{\partial C}{\partial\alpha_k}\frac{\partial^2\alpha_k(\boldsymbol{z}^k)}{\partial z_i\partial z_j} \\ &= \sum_{k=0}^{p-1}\sum_{l=0}^{p-1}\sum_{r=0}^{m_k-1}\sum_{s=0}^{m_l-1}\frac{\partial^2 C}{\partial\alpha_k\partial\alpha_l}\frac{\partial\alpha_k}{\partial z_r^k}\frac{\partial\alpha_l}{\partial z_s^l}\left[L_{ir}^k L_{js}^l\right] \\ &\quad + \sum_{k=0}^{p-1}\sum_{r=0}^{m_k-1}\sum_{s=0}^{m_k-1}\frac{\partial C}{\partial\alpha_k}\frac{\partial^2\alpha_k}{\partial z_r^k\partial z_s^k}\left[L_{ir}^k L_{js}^k\right], \end{aligned} \tag{S3.57}$$

where $\left[\partial\alpha_k/\partial z_r^k\right] = \partial\alpha_k/\partial\boldsymbol{z}^k$ and $\left[\partial\alpha_l/\partial q_s^l\right] = \partial\alpha_l/\partial\boldsymbol{z}^l$ are the gradients of the individual measures in their own coordinates; $\left[\partial^2\alpha_k/\partial z_r^k\partial z_s^k\right] = \partial^2\alpha_k/\partial\boldsymbol{z}^k\partial\boldsymbol{z}^{k\top}$ are the hessian matrices of the individual measures in their own coordinates; and where $\left[L_{ir}^k L_{js}^l\right]$ and $\left[L_{ir}^k L_{js}^k\right]$ are computed using Eq. (S3.56). In practical terms, Eq. (S3.57) means that the hessian matrix $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$ is constructed by summing together the components

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

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

S3.3.7 Signed point-line distance (distance flexel)

A distance flexel is defined by 3 nodes: N_0, N_1 and N_2 (Fig. S3.1e). The measure α 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 $\overrightarrow{N_1N_2}$), 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 $\boldsymbol{z} = [x_0, y_0, x_1, y_1, x_2, y_2]^\top$.

Measure α The signed distance measure α can be regarded as a compound measure. It is twice the signed area of the triangle $N_0N_1N_2$ (α_0) divided by the length of the segment N_1N_2 (α_1):

$$\alpha = C(\alpha_0, \alpha_1) = \frac{2\alpha_0}{\alpha_1} = \frac{2A_s(\boldsymbol{z}^0)}{l(\boldsymbol{z}^1)} = d(\boldsymbol{z}), \tag{S3.58}$$

where $A_s(\boldsymbol{z}^0)$ is the signed area of the triangle $N_0N_1N_2$ (Eq. (S3.42)), with $\boldsymbol{z}^0 = [x_0, y_0, x_1, y_1, x_2, y_2]^\top$; and $l(\boldsymbol{z}^1)$ is the length of the segment N_1N_2 (Eq. (S3.25)), with $\boldsymbol{z}^1 = [x_1, y_1, x_2, y_2]^\top$.

Gradient $\partial\alpha/\partial\boldsymbol{z}$ As the signed distance measure is a compound measure, its gradient $\partial\alpha/\partial\boldsymbol{z}$ is computed using Eq. (S3.55), 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 $\partial\alpha_0/\partial\boldsymbol{z}^0$ and $\partial\alpha_1/\partial\boldsymbol{z}^1$ are computed using Eq. (S3.43) and Eq. (S3.26), respectively.

Hessian $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$ As the signed distance measure is a compound measure, its hessian matrix $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$ is computed using Eq. (S3.57), 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 $\partial\alpha_0/\partial\boldsymbol{z}^0$, $\partial\alpha_1/\partial\boldsymbol{z}^1$, $\partial^2\alpha_0/\partial\boldsymbol{z}^0\partial\boldsymbol{z}^{0\top}$, $\partial^2\alpha_1/\partial\boldsymbol{z}^1\partial\boldsymbol{z}^{1\top}$ are computed using Eq. (S3.43), Eq. (S3.26), Eq. (S3.46) and Eq. (S3.27), respectively.

S3.3.8 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 $N_0, N_1, \ldots, N_{n-1}$ (Fig. S3.1f). The measure of α 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 $\boldsymbol{z} = [x_0, y_0, \ldots, x_{n-1}, y_{n-1}]^\top$.

Measure α The path length measure α can be regarded as a compound measure. It is the sum of the lengths of the p segments N^kN^{k+1}:

$$\alpha = C(\alpha_0, \alpha_1, \ldots, \alpha_{p-1}) = \alpha_0 + \alpha_1 + \cdots + \alpha_{p-1} = \sum_{k=0}^{p-1} l_k(\boldsymbol{z}^k), \tag{S3.59}$$

where $\alpha_k = l_k$ is the length of the segment $N^k N^{k+1}$ (Eq. (S3.25)) and $\boldsymbol{z}^k = [x_k, y_k, x_{k+1}, y_{k+1}]^\top$.

GRADIENT $\partial\alpha/\partial\boldsymbol{z}$ As the path length measure is a compound measure, its gradient $\partial\alpha/\partial\boldsymbol{z}$ is computed using Eq. (S3.55), with $\partial C/\partial\alpha_k = 1$, and where the individual gradients $\partial\alpha_k/\partial\boldsymbol{z}^k$ are computed using Eq. (S3.26). In case the sequence is composed of distinct nodes ($N^k = N_k$), the gradient $\partial\alpha/\partial\boldsymbol{z}$ takes the following form:

$$\frac{\partial\alpha}{\partial\boldsymbol{z}} = \frac{1}{l_0}\begin{bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ x_1 - x_0 \\ y_1 - y_0 \\ 0 \\ \vdots \\ \\ \\ \\ \\ 0 \end{bmatrix} + \frac{1}{l_1}\begin{bmatrix} 0 \\ 0 \\ x_1 - x_2 \\ y_1 - y_2 \\ x_2 - x_1 \\ y_2 - y_1 \\ 0 \\ \vdots \\ \\ \\ 0 \end{bmatrix} + \cdots + \frac{1}{l_{n-2}}\begin{bmatrix} 0 \\ \vdots \\ \\ \\ \\ \\ 0 \\ x_{n-2} - x_{n-1} \\ y_{n-2} - y_{n-1} \\ x_{n-1} - x_{n-2} \\ y_{n-1} - y_{n-2} \end{bmatrix}. \tag{S3.60}$$

HESSIAN $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$ As the path length measure is a compound measure, its hessian matrix $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$ is computed using Eq. (S3.57), 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 $\partial\alpha_k/\partial\boldsymbol{z}^k$ and $\partial^2\alpha_k/\partial\boldsymbol{z}^k\partial{\boldsymbol{z}^k}^\top$ are computed using Eq. (S3.26) and Eq. (S3.27), respectively.

S3.3.9 Area with holes (area flexel)

In Section S3.3.4, 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. S3.1g). The geometric measure α of such a polygon is the area delimited by the outer boundary A_0 to which is subtracted the areas delimited by the inner boundaries $A_1, A_2, \ldots, A_{p-1}$. Each area A_k is defined by a set of coordinates, noted $\boldsymbol{z}^k$. The array of unique coordinates $\boldsymbol{z}$ is constructed by sequentially appending

each coordinate from $\boldsymbol{z}^0, \ldots, \boldsymbol{z}^{p-1}$ that has not yet been encountered.

Measure α The area-with-holes measure α can be regarded as a compound measure. It is outer area minus the sum of the inner areas:

$$\alpha = C(\alpha_0, \alpha_1, \ldots, \alpha_{p-1}) = A_0 - \sum_{k=1}^{p-1} A_k(\boldsymbol{z}^k) = A(\boldsymbol{z}). \tag{S3.61}$$

where $\alpha_k = A_k(\boldsymbol{z}^k)$ is computed using Eq. (S3.48).

Gradient $\partial\alpha/\partial\boldsymbol{z}$ As the area-with-holes measure is a compound measure, its gradient $\partial\alpha/\partial\boldsymbol{z}$ is computed using Eq. (S3.55), with $\partial C/\partial\alpha_0 = 1$ and $\partial C/\partial\alpha_k = -1 \quad (\forall k \neq 0)$, and where the individual gradients $\partial\alpha_k/\partial\boldsymbol{z}^k$ are computed using Eq. (S3.49).

Hessian $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$ As the area-with-holes measure is a compound measure, its hessian matrix $\partial^2\alpha/\partial\boldsymbol{z}\partial\boldsymbol{z}^\top$ is computed using Eq. (S3.57), with $\partial C/\partial\alpha_0 = 1$, $\partial C/\partial\alpha_k = -1 \quad (\forall k \neq 0)$ and $\partial^2 C/\partial\alpha_k\partial\alpha_l = 0$, where the individual gradients and hessian matrices $\partial\alpha_k/\partial\boldsymbol{z}^k$ and $\partial^2\alpha_k/\partial\boldsymbol{z}^k\partial{\boldsymbol{z}^k}^\top$ are computed using Eq. (S3.49) and Eq. (S3.50), respectively.

S3.4 Intrinsic nonlinear mechanical behaviors

When computing the gradient and hessian of the elastic energy, necessary for the arclength method, the gradient and hessian of the intrinsic elastic potential v of the flexel with respect to the measure α (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 $\alpha - \alpha_0$ (α_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. S3.3. In the following, we use apostrophes, g' and g'', to denote the first and second derivatives of functions g of one single argument.

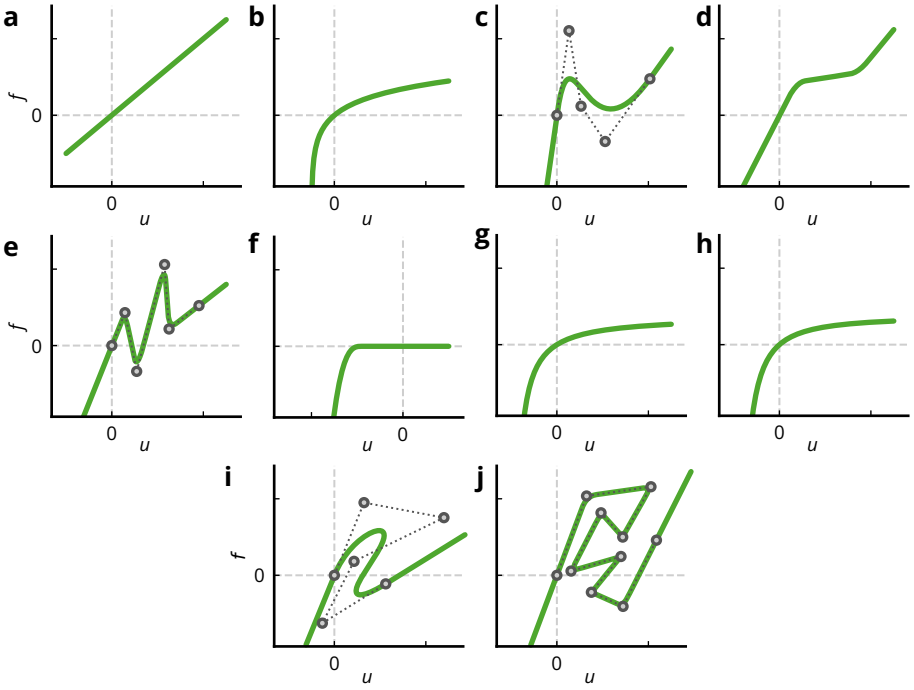


Figure S3.3: Nonlinear generalized force (f)-displacement (u) curves used to construct nonlinear behaviors. **(a)** Linear. **(b)** Logarithmic. **(c)** Bezier (univariate). **(d)** C_1-piecewise linear. **(e)** Zigzag (univariate). **(f)** Contact. **(g)** Isothermal. **(h)** Isentropic. **(i)** Bezier (bivariate). **(j)** Zigzag (bivariate). Control points and control polygons are depicted by gray markers and dotted lines.

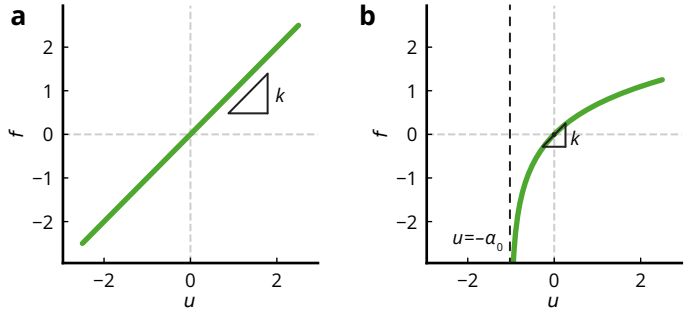


Figure S3.4: Linear **(a)** and logarithmic **(b)** force-displacement curves.

S3.4.1 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 a function $f(u)$.

Linear behavior A linear behavior is defined by a linear generalized force-displacement curve (Fig. S3.4a),

$$f(u) = ku, \tag{S3.62}$$

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:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)\mathrm{d}u = \frac{1}{2}k(\alpha-\alpha_0)^2, \tag{S3.63}$$

$$\frac{\partial v}{\partial\alpha}(\alpha) = f(\alpha-\alpha_0) = k(\alpha-\alpha_0), \tag{S3.64}$$

$$\frac{\partial^2 v}{\partial\alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha-\alpha_0) = k. \tag{S3.65}$$

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

$$f(u) = k\alpha_0 \ln\left(\frac{u+\alpha_0}{\alpha_0}\right), \tag{S3.66}$$

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. S3.4b). The behavior is identical to the linear behavior at first order around the natural measure α_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:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)\mathrm{d}u = k\alpha\alpha_0\left(\ln\left(\alpha/\alpha_0\right) - 1\right), \tag{S3.67}$$

$$\frac{\partial v}{\partial\alpha}(\alpha) = f(\alpha-\alpha_0) = k\alpha_0 \ln\left(\alpha/\alpha_0\right), \tag{S3.68}$$

$$\frac{\partial^2 v}{\partial\alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha-\alpha_0) = k\alpha_0/\alpha. \tag{S3.69}$$

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),..., (u_n, f_n), with linear extrapolation beyond the first and last control points that preserves C_1 continuity. The nonlinear behavior defined by the Bezier curve can either describe the tensile behavior (a.k.a 'tensile' mode, Fig. S3.5a), the compressive behavior (a.k.a 'compressive' mode, Fig. S3.5b) or both simultaneously (a.k.a 'symmetric' mode, Fig. S3.5c).

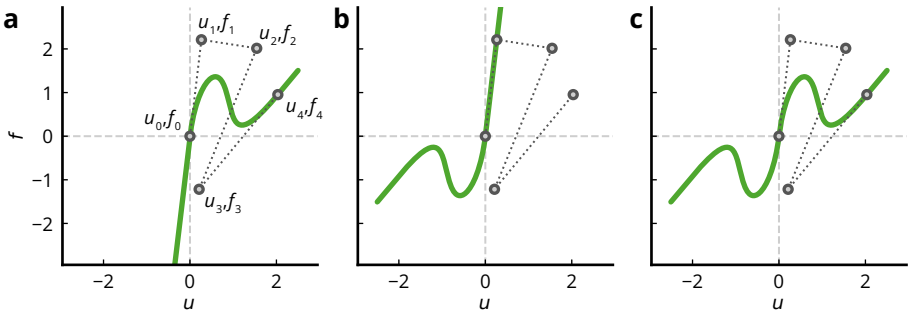


Figure S3.5: Bezier force-displacement curves for the tensile **(a)**, compressive **(b)** and symmetric **(c)** modes.

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

$$\mathcal{F}(g)(x) = m(x)g(m(x)x), \tag{S3.70}$$

where g is a function and

$$m(x) = \begin{cases} +1 & \text{if 'tensile' mode,} \\ -1 & \text{if 'compressive' mode,} \\ \operatorname{sgn}(x) & \text{if 'symmetric' mode.} \end{cases} \tag{S3.71}$$

Note that

$$\frac{\partial}{\partial x}\mathcal{F}(g)(x) = g'(m(x)x) := \mathcal{F}'(g)(x), \tag{S3.72}$$

$$\frac{\partial^2}{\partial x^2}\mathcal{F}(g)(x) = m(x)g''(m(x)x) := \mathcal{F}''(g)(x). \tag{S3.73}$$

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

$$f(u) = \mathcal{F}(\bar{f})(u), \tag{S3.74}$$

where

$$\bar{f}(u) = \begin{cases} (f_1/u_1)u & \text{if } u \le 0, \\ b(a^{-1}(u)) & \text{if } 0 < u \le u_n \\ f_n + \dfrac{f_n - f_{n-1}}{u_n - u_{n-1}}(u - u_n) & \text{if } u > u_n, \end{cases} \tag{S3.75}$$

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

$$\begin{cases} a(x) &= \sum_{i=0}^{n} u_i B_{i,n}(x) \\ b(x) &= \sum_{i=0}^{n} f_i B_{i,n}(x), \end{cases} \tag{S3.76}$$

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.

$$k(u) = \frac{\mathrm{d}f}{\mathrm{d}u} = \mathcal{F}'(\bar{f})(u), \tag{S3.77}$$

where $\mathcal{F}'$ is defined in Eq. (S3.72) and

$$\frac{\mathrm{d}\bar{f}}{\mathrm{d}u} = \begin{cases} f_1/u_1 & \text{if } u \le 0, \\ \dfrac{b'(a^{-1}(u))}{a'(a^{-1}(u))} & \text{if } 0 < u \le u_n \\ \dfrac{f_n - f_{n-1}}{u_n - u_{n-1}} & \text{if } u > u_n, \end{cases} \tag{S3.78}$$

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

$$a'(x) = n\sum_{i=0}^{n-1}(u_{i+1} - u_i)B_{i,n-1}(x) \quad \text{and} \quad b'(x) = n\sum_{i=0}^{n-1}(f_{i+1} - f_i)B_{i,n-1}(x). \tag{S3.79}$$

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:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)\mathrm{d}u, \tag{S3.80}$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0), \tag{S3.81}$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha - \alpha_0) = k(\alpha - \alpha_0), \tag{S3.82}$$

where f and k are described in Eqs. (S3.74, S3.77).

C_1-PIECEWISE BEHAVIOR A C_1-piecewise behavior is defined by a piecewise linear function whose 'corners' have been smoothed by quadratic functions, ensuring C_1 continuity. Bilinear and trilinear piecewise linear force-displacement curves have been used to study softening, stiffening or bistable interactions in spring chains [121–123]. 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 $k_0, \ldots, k_{n-1}$, $n-1$ displacement values $u_0, \ldots, u_{n-2}$ 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. S3.6a), the compressive behavior (a.k.a 'compressive' mode, Fig. S3.6b) or both simultaneously (a.k.a 'symmetric' mode, Fig. S3.6c).

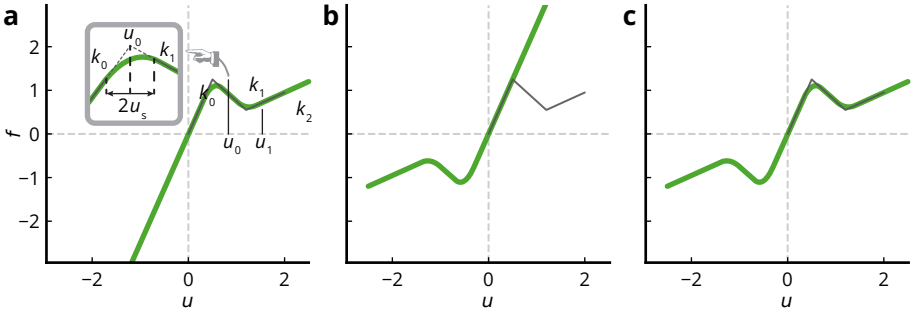


Figure S3.6: C_1-piecewise linear force-displacement curves for the tensile **(a)**, compressive **(b)** and symmetric **(c)** modes.

Mathematically speaking,

$$f(u) = \mathcal{F}(\bar{f})(u), \tag{S3.83}$$

where $\mathcal{F}$ is defined in Eq. (S3.70) and

$$\bar{f}(u) = P(u; \boldsymbol{s} = [k_0, k_1, \ldots, k_{n-1}]; \boldsymbol{\eta} = [u_0, u_1, \ldots, u_{n-2}]; \eta_s = u_s), \tag{S3.84}$$

where

$$P(\eta; \boldsymbol{s}; \boldsymbol{\eta}; \eta_s) = \begin{cases} s_0\eta & \text{if } \eta \le \eta_0 - \eta_s \\ a_0\eta^2 + b_0\eta + c_0 & \text{if } \eta_0 - \eta_s < \eta < \eta_0 + \eta_s \\ s_1\eta + p_1 & \text{if } \eta_0 + \eta_s \le \eta \le \eta_1 - \eta_s \\ \vdots & \vdots \\ a_i\eta^2 + b_i\eta + c_i & \text{if } \eta_i - \eta_s < \eta < \eta_i + \eta_s \\ s_i\eta + p_i & \text{if } \eta_{i-1} + \eta_s \le \eta \le \eta_i - \eta_s \\ \vdots & \vdots \\ a_{n-2}\eta^2 + b_{n-2}\eta + c_{n-2} & \text{if } \eta_{n-2} - \eta_s < \eta < \eta_{n-2} + \eta_s \\ s_{n-1}\eta + p_{n-1} & \text{if } \eta_{n-2} + \eta_s \le \eta < +\infty, \end{cases} \tag{S3.85}$$

with

$$p_i = p_{i-1} + \eta_{i-1}\,(s_{i-1} - s_i)\,; \quad p_0 = 0, \tag{S3.86}$$

$$a_i = \frac{s_{i+1} - s_i}{4\eta_s}, \tag{S3.87}$$

$$b_i = \frac{s_i(\eta_i + \eta_s) - s_{i+1}(\eta_i - \eta_s)}{2\eta_s}, \tag{S3.88}$$

$$c_i = \frac{(s_{i+1} - s_i)(\eta_i - \eta_s)^2}{4\eta_s} + p_i. \tag{S3.89}$$

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

$$k(u) = \frac{\mathrm{d}f}{\mathrm{d}u} = \mathcal{F}'(\bar{f})(u), \tag{S3.90}$$

where $\mathcal{F}'$ is defined in Eq. (S3.72) and

$$\frac{\mathrm{d}\bar{f}}{\mathrm{d}u} = \frac{\partial P}{\partial \eta}(u; [k_0, \ldots, k_{n-1}]; [u_0, \ldots, u_{n-2}]; u_s), \tag{S3.91}$$

where

$$\frac{\partial P}{\partial \eta}(\eta, \boldsymbol{s}, \boldsymbol{\eta}, \eta_s) = \begin{cases} s_0 & \text{if } \eta \leq \eta_0 - \eta_s \\ 2a_0\eta + b_0 & \text{if } \eta_0 - \eta_s < \eta < \eta_0 + \eta_s \\ s_1 & \text{if } \eta_0 + \eta_s \leq \eta \leq \eta_1 - \eta_s \\ \vdots & \vdots \\ 2a_i\eta + b_i & \text{if } \eta_i - \eta_s < \eta < \eta_i + \eta_s \\ s_i & \text{if } \eta_{i-1} + \eta_s \leq \eta \leq \eta_i - \eta_s \\ \vdots & \vdots \\ 2a_{n-2}\eta + b_{n-2} & \text{if } \eta_{n-2} - \eta_s < \eta < \eta_{n-2} + \eta_s \\ s_{n-1} & \text{if } \eta_{n-2} + \eta_s \leq \eta < +\infty. \end{cases} \tag{S3.92}$$

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:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)\mathrm{d}u, \tag{S3.93}$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0), \tag{S3.94}$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha - \alpha_0) = k(\alpha - \alpha_0), \tag{S3.95}$$

where f and k are described in Eqs. (S3.83, S3.90).

Zigzag behavior A zigzag behavior is described by a "smooth zigzag", that is, a polygonal chain with vertices positioned at $(u_0 = 0, f_0 = 0), (u_1, f_1), \ldots, (u_n, f_n)$, 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 C_1-piecewise linear behavior, except that it can be generalized to a multivalued curve as shown in subsection S3.4.2. Similarly to the Bezier behavior, the smooth zigzag can either describe the tensile behavior (Fig. S3.7a), the compressive behavior (Fig. S3.7b) or both simultaneously (Fig. S3.7c).

Mathematically speaking,

$$f(u) = \mathcal{F}(\bar{f})(u) \tag{S3.96}$$

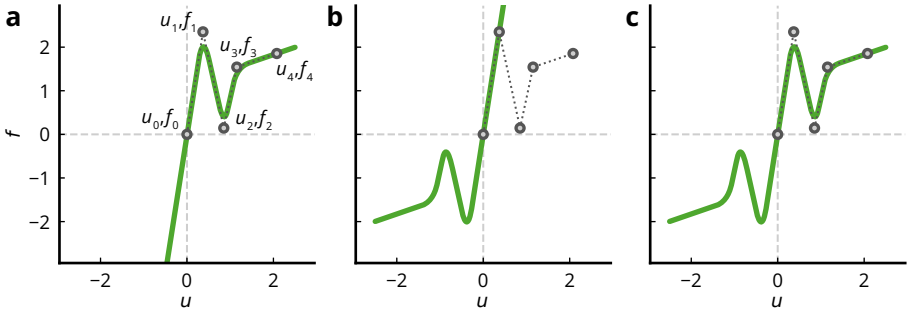


Figure S3.7: Zigzag force-displacement curves for the tensile **(a)**, compressive **(b)** and symmetric **(c)** modes.

where $\mathcal{F}$ is defined in Eq. (S3.70) and

$$\bar{f}(u) = b(a^{-1}(u)), \tag{S3.97}$$

where the functions $a(.)$ and $b(.)$ are defined by C_1-piecewise linear functions:

$$a(x) = P(x; \boldsymbol{s}^a; \boldsymbol{\eta}^a; \eta_{\mathrm{s}}^a) \tag{S3.98}$$

$$b(x) = P(x; \boldsymbol{s}^b; \boldsymbol{\eta}^b; \eta_{\mathrm{s}}^b), \tag{S3.99}$$

with P defined in Eq. (S3.85), $s_i^a = (n-1)(u_{i+1} - u_i)$, $s_i^b = (n-1)(f_{i+1} - f_i)$, $\eta_i^a = \eta_i^b = i/(n-1)$, and $\eta_{\mathrm{s}}^a = \eta_{\mathrm{s}}^b = \epsilon/(2n-2)$. 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 $0 < u_1 < \cdots < u_{n-2}$. The stiffness $k(u)$ along the curve is given by the derivative of $f(u)$, i.e.

$$k(u) = \frac{\mathrm{d}f}{\mathrm{d}u}(u) = \mathcal{F}'(\bar{f})(u), \tag{S3.100}$$

where $\mathcal{F}'$ is defined in Eq. (S3.72) and

$$\frac{\mathrm{d}\bar{f}}{\mathrm{d}u}(u) = b'(a^{-1}(u))/a'(a^{-1}(u)), \tag{S3.101}$$

where $a' = \partial P/\partial\eta(x, \boldsymbol{s}^a, \boldsymbol{\eta}^a, \eta_{\mathrm{s}}^a)$ and $b' = \partial P/\partial\eta(x, \boldsymbol{s}^b, \boldsymbol{\eta}^b, \eta_{\mathrm{s}}^b)$ are the derivatives of a and b with respect to x, with $\partial P/\partial\eta$ given in Eq. (S3.92). From the curve, the

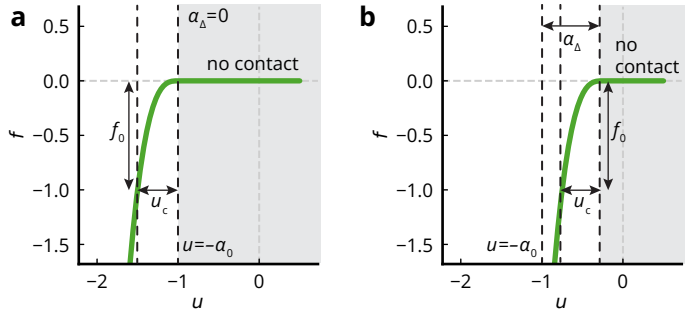


Figure S3.8: "Contact" force-displacement curves with $\alpha_0 = 1$ when $\alpha_\Delta = 0$ **(a)** and $\alpha_\Delta = 0.75$ **(b)**.

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

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)\mathrm{d}u, \tag{S3.102}$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0), \tag{S3.103}$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha - \alpha_0) = k(\alpha - \alpha_0), \tag{S3.104}$$

where f and k are described in Eqs. (S3.96, S3.100).

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 α_Δ (Fig. S3.8):

$$f(u) = \begin{cases} 0 & \text{if } u \geq \alpha_\Delta - \alpha_0 \\ -f_0 \left(\dfrac{\alpha_\Delta - \alpha_0 - u}{u_c} \right)^3 & \text{if } u < \alpha_\Delta - \alpha_0, \end{cases} \tag{S3.105}$$

where $f_0 > 0$ is the magnitude of the repulsive force when the measure is decreased by $u_c > 0$ from α_Δ. The curve is C_2 continuous. 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:

$$v(\alpha) = \int_{\alpha_\Delta}^{\alpha} f(u)\mathrm{d}u = \begin{cases} 0 & \text{if } \alpha \geq \alpha_\Delta \\ \dfrac{f_0 u_c}{4}\left(\dfrac{\alpha_\Delta - \alpha}{u_c}\right)^4 & \text{if } \alpha < \alpha_\Delta, \end{cases} \tag{S3.106}$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0) = \begin{cases} 0 & \text{if } \alpha \geq \alpha_\Delta \\ -f_0\left(\dfrac{\alpha_\Delta - \alpha}{u_c}\right)^3 & \text{if } \alpha < \alpha_\Delta, \end{cases} \tag{S3.107}$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha - \alpha_0) = \begin{cases} 0 & \text{if } \alpha \geq \alpha_\Delta \\ \dfrac{3f_0}{u_c}\left(\dfrac{\alpha_\Delta - \alpha}{u_c}\right)^2 & \text{if } \alpha < \alpha_\Delta. \end{cases} \tag{S3.108}$$

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):

$$p - p_0 = nRT_0(1/V - 1/V_0), \tag{S3.109}$$

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. (S3.109) 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. S3.9a):

$$f(u) = \frac{nRT_0}{\alpha_0}\frac{u}{u + \alpha_0}. \tag{S3.110}$$

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 con-

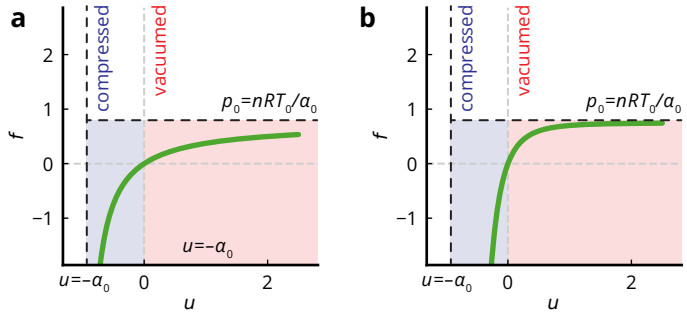


Figure S3.9: Force-displacement curves mimicking the ideal gas law. **(a)** Isothermal with $n = 0.03, R = 0.083, T_0 = 300$. **(b)** Isentropic with $n = 0.03, R = 0.083, T_0 = 300, \gamma = 4$.

structed as follows:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)\mathrm{d}u = nRT_0(\alpha/\alpha_0 - 1 - \ln(\alpha/\alpha_0)), \tag{S3.111}$$

$$\frac{\partial v}{\partial \alpha}(\alpha) = f(\alpha - \alpha_0) = nRT_0\frac{\alpha - \alpha_0}{\alpha\alpha_0}, \tag{S3.112}$$

$$\frac{\partial^2 v}{\partial \alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha - \alpha_0) = \frac{nRT_0}{\alpha^2}. \tag{S3.113}$$

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):

$$p - p_0 = nRT_0\left(\frac{1}{V}\left(\frac{V_0}{V}\right)^{\gamma-1} - \frac{1}{V_0}\right), \tag{S3.114}$$

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 γ the heat capacity ratio. The generalized-force displacement curve is defined from Eq. (S3.114) 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. S3.9b):

$$f(u) = nRT_0\left(\frac{1}{\alpha_0} - \frac{1}{u + \alpha_0}\left(\frac{\alpha_0}{u + \alpha_0}\right)^{\gamma-1}\right), \tag{S3.115}$$

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:

$$v(\alpha) = \int_0^{\alpha-\alpha_0} f(u)\mathrm{d}u = nRT_0\left(\frac{\alpha}{\alpha_0} - 1\right) + \frac{nRT_0}{\gamma-1}\left(\left(\frac{\alpha_0}{\alpha}\right)^{\gamma-1} - 1\right), \quad \text{(S3.116)}$$

$$\frac{\partial v}{\partial\alpha}(\alpha) = f(\alpha-\alpha_0) = nRT_0\left(\frac{1}{\alpha_0} - \frac{1}{\alpha}\left(\frac{\alpha_0}{\alpha}\right)^{\gamma-1}\right), \quad \text{(S3.117)}$$

$$\frac{\partial^2 v}{\partial\alpha^2}(\alpha) = \frac{\mathrm{d}f}{\mathrm{d}u}(\alpha-\alpha_0) = \frac{nRT_0}{\alpha^2}\left(\frac{\alpha_0}{\alpha}\right)^{\gamma-1}. \quad \text{(S3.118)}$$

S3.4.2 Bivariate behavior

A bivariate behavior is characterized by a bivariate energy potential, which is constructed from a generalized force-displacement that can possibly be multivalued, 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:

$$\begin{cases} u = a(x) \\ f = b(x), \end{cases} \quad \text{(S3.119)}$$

where $a(.)$ and $b(.)$ 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 α and an internal coordinate t, is constructed

$$v(\alpha,t) = \frac{1}{2}k(t)\,(\alpha-\alpha_0-a(t))^2 + b(t)(\alpha-\alpha_0-a(t)) + \int_0^t b(\tilde{t})a'(\tilde{t})\mathrm{d}\tilde{t}. \quad \text{(S3.120)}$$

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. (S3.126, S3.127, S3.128, S3.129) (see colored box below). The function $k(.)$ can be defined so that the potential v has the desired stability properties, as we will demonstrate afterwards. Its expression is given in Eq. (S3.146). At equilibrium, the internal coordinate t plays the role of the curve parameter. From the potential, we can compute its gradient

and hessian components:

$$\frac{\partial v}{\partial \alpha} = k(t)(\alpha - \alpha_0 - a(t)) + b(t), \tag{S3.121}$$

$$\frac{\partial v}{\partial t} = (\alpha - \alpha_0 - a(t))(b'(t) - k(t)a'(t)) + \frac{k'(t)}{2}(\alpha - \alpha_0 - a(t))^2, \tag{S3.122}$$

$$\frac{\partial^2 v}{\partial \alpha^2} = k(t), \tag{S3.123}$$

$$\frac{\partial^2 v}{\partial \alpha \partial t} = b'(t) - k(t)a'(t) + k'(t)(\alpha - \alpha_0 - a(t)), \tag{S3.124}$$

$$\begin{aligned}\frac{\partial^2 v}{\partial t^2} = &(\alpha - \alpha_0 - a(t))(b''(t) - k(t)a''(t)) + a'(t)(k(t)a'(t) - b'(t)) \\ &+ (\alpha - \alpha_0 - a(t))\left(\frac{k''(t)}{2}(\alpha - \alpha_0 - a(t)) - 2k'(t)a'(t)\right).\end{aligned} \tag{S3.125}$$

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. (S3.152, S3.153)) or a smooth zigzag curve (Eqs. (S3.164, S3.165)) (see paragraphs below).

LEMMA: POINTS ON THE PARAMETRIC CURVE $(a(t), b(t))$ CORRESPOND TO STATIC EQUILIBRIA

PROOF States (α, t) are equilibrium points if they are stationary points of the total potential energy $\pi(\alpha, t) := v(\alpha, t) - f\alpha$ [2], meaning that they have to satisfy the two following equations,

$$\begin{aligned} 0 &= \frac{\partial \pi}{\partial \alpha} = \frac{\partial v}{\partial \alpha} - f \\ &= k(t)(\alpha - \alpha_0 - a(t)) + b(t) - f, \end{aligned} \tag{S3.126}$$

$$\begin{aligned} 0 &= \frac{\partial \pi}{\partial t} = \frac{\partial v}{\partial t} \\ &= (\alpha - \alpha_0 - a(t)) \left(b'(t) - k(t)a'(t) + \frac{k'(t)}{2}(\alpha - \alpha_0 - a(t)) \right). \end{aligned} \tag{S3.127}$$

From (S3.127), we notice that

$$\alpha = \alpha_0 + a(t) \Leftrightarrow u = a(t) \tag{S3.128}$$

is a solution, which, when plugged back into Eq. (S3.126), yields

$$f = b(t). \tag{S3.129}$$

From Eqs. (S3.128) and (S3.129), it can be noticed that each point on the force-displacement curve given in Eq. (S3.119) is a stationary point of the total potential energy, which retrospectively explains why we define the elastic energy of the flexel using Eq. (S3.120).

DEFINING $k(.)$ TO SATISFY STABILITY CONDITIONS On the one hand, stability of equilibria are determined by the hessian matrix $\boldsymbol{h}$ of the total potential energy

$$\boldsymbol{h}(\alpha, t) = \begin{bmatrix} \partial^2 \pi / \partial \alpha^2 & \partial^2 \pi / \partial \alpha \partial t \\ \partial^2 \pi / \partial \alpha \partial t & \partial^2 \pi / \partial t^2 \end{bmatrix}. \tag{S3.130}$$

More precisely, stability of points under force-controlled conditions is governed by the number μ_f of negative eigenvalues in the matrix $\boldsymbol{h}$, each indicating an un-

stable 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 μ_α of negative eigenvalues in $\boldsymbol{h}$ from which the row and column corresponding to the variable α 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 extremum) should dictate the stability of equilibrium points along the $u - f$ curve [100]. 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 μ_f or μ_α, under the respective loading condition should increase (decrease) by one (Fig. S3.10a).

To ensure consistency between the folding rules and the numbers of negative eigenvalues μ_f and μ_α, the following conditions must be satisfied:

$$\text{if } (\nearrow) \equiv a'(t) > 0, b'(t) > 0 \Leftrightarrow \mu_f = 0 \text{ and } \mu_\alpha = 0; \tag{S3.131}$$

$$\text{if } (\searrow) \equiv a'(t) > 0, b'(t) < 0 \Leftrightarrow \mu_f = 1 \text{ and } \mu_\alpha = 0; \tag{S3.132}$$

$$\text{if } (\swarrow) \equiv a'(t) < 0, b'(t) < 0 \Leftrightarrow \mu_f = 1 \text{ and } \mu_\alpha = 1; \tag{S3.133}$$

$$\text{if } (\nwarrow) \equiv a'(t) < 0, b'(t) > 0 \Leftrightarrow \mu_f = 2 \text{ and } \mu_\alpha = 1. \tag{S3.134}$$

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

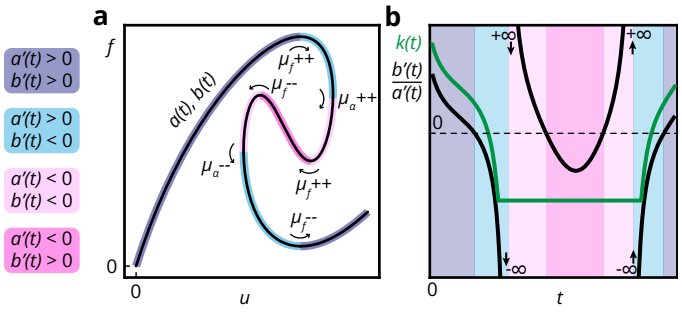


Figure S3.10: Example of multivalued force-displacement curve and its slope. **(a)** Force-displacement curve described by parametric equations $u = a(t), f = b(t)$. **(b)** Slope of the tangent of the curve as a function of the curve parameter t (black) and the associated $k(t)$ function (Eq. (S3.146)).

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 $1 \geq \mu_\alpha(\nearrow) = 2$, both being impossible. An example of generalized force-displacement showing valid folds is shown in Fig. S3.10a.

To derive the conditions to satisfy the desired stability conditions (S3.131),(S3.132), (S3.133) and (S3.134), let us first express the hessian matrix when evaluated at an equilibrium point $(t, \alpha) = (t, \alpha_0 + a(t))$:

$$\boldsymbol{h}(\alpha_0 + a(t), t) = \begin{bmatrix} k(t) & b'(t) - k(t)a'(t) \\ b'(t) - k(t)a'(t) & a'(t)(k(t)a'(t) - b'(t)) \end{bmatrix}. \tag{S3.135}$$

From conditions (S3.131), (S3.132), (S3.133) and (S3.134), notice that $\mu_\alpha = 0$ if $a' > 0$ else 1, meaning that

$$\begin{cases} a'(ka' - b') > 0 & \text{if } a' > 0 \\ a'(ka' - b') < 0 & \text{if } a' < 0 \end{cases} \Leftrightarrow \begin{cases} k > b'/a' & \text{if } a' > 0 \\ k < b'/a' & \text{if } a' < 0. \end{cases} \tag{S3.136}$$

To satisfy $\mu_f = 0$ in condition (S3.131), the hessian matrix (Eq. (S3.135)) must be positive-definite on ($\nearrow$)-branches, which, by applying Sylvester's criterion,

yields

$$k > 0 \quad \text{and} \quad \det \boldsymbol{h} = b'(ka' - b')) > 0 \quad \text{when } a', b' > 0 \tag{S3.137}$$

$$\Leftrightarrow$$

$$k > b'/a' \quad \text{when } a', b' > 0. \tag{S3.138}$$

To satisfy $\mu_f = 1$ in condition (S3.132), the hessian matrix must be indefinite on ($\searrow$)-branches, which, for a 2-by-2 matrix, yields

$$\det \boldsymbol{h} = b'(ka' - b') < 0 \quad \text{when } a' > 0, b' < 0 \tag{S3.139}$$

$$\Leftrightarrow$$

$$k > b'/a' \quad \text{when } a' > 0, b' < 0. \tag{S3.140}$$

To satisfy $\mu_f = 1$ in condition (S3.133), the hessian matrix must be indefinite on ($\swarrow$)-branches as well:

$$\det \boldsymbol{h} = b'(ka' - b') < 0 \quad \text{when } a', b' < 0 \tag{S3.141}$$

$$\Leftrightarrow$$

$$k < b'/a' \quad \text{when } a', b' < 0. \tag{S3.142}$$

To satisfy $\mu_f = 2$ in condition (S3.134), the hessian matrix must be negative-definite on ($\nwarrow$)-branches, which, by applying Sylvester's criterion, yields

$$k < 0 \quad \text{and} \quad \det \boldsymbol{h} = b'(ka' - b')) > 0 \quad \text{when } a' < 0, b' > 0 \tag{S3.143}$$

$$\Leftrightarrow$$

$$k < b'/a' \quad \text{when } a' < 0, b' > 0. \tag{S3.144}$$

Inequalities (S3.136), (S3.138), (S3.140), (S3.142) and (S3.144) are consistent and can be equivalently summarized as follows

$$\begin{cases} k(t) > b'(t)/a'(t) & \text{if } a'(t) > 0 \\ k(t) < b'(t)/a'(t) & \text{if } a'(t) < 0, \end{cases} \tag{S3.145}$$

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. S3.10b. In `springable`, the k function is constructed as follows

$$k(t) = \begin{cases} k^\star & \text{if } k_{\min} - k_{\max} > 2\delta \\ \hat{k}(t) & \text{else,} \end{cases} \tag{S3.146}$$

where

$$k_{\max} = \max_{t:a'(t)>0} b'(t)/a'(t) \tag{S3.147}$$

$$k_{\min} = \begin{cases} \min_{t:a'(t)<0} b'(t)/a'(t) & \text{if } \exists t : a'(t) < 0 \\ +\infty & \text{else,} \end{cases} \tag{S3.148}$$

$$\delta = k_{\max}/20, \tag{S3.149}$$

$$k^\star = \min(k_{\min} - \delta, k_{\max} + \delta), \tag{S3.150}$$

$$\hat{k}(t) = \begin{cases} \max\{b'(t)/a'(t) + \delta, k^\star\} & \forall t : a'(t) > 0 \\ k^\star & \forall t : a'(t) < 0. \end{cases} \tag{S3.151}$$

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 C_1-continuity. Unlike its univariate counterpart (Section S3.4.1, paragraph 'Bezier behavior'), its curve can be multivalued. The nonlinear behavior defined by the Bezier curve can either describe the tensile behavior (a.k.a 'tensile' mode, Fig. S3.11a), the compressive behavior (a.k.a 'compressive' mode, Fig. S3.11b) or both simultaneously (a.k.a 'symmetric' mode, Fig. S3.11c). Mathematically speaking, the force-displacement curve is defined by

$$u = a(t) = \mathcal{F}(\bar{a})(t) \tag{S3.152}$$

$$f = b(t) = \mathcal{F}(\bar{b})(t) \tag{S3.153}$$

where the functional $\mathcal{F}$ is defined in Eq. (S3.70) and

$$\bar{a}(t) = \begin{cases} nu_1 t/t_{\max} & \text{if } t \leq 0 \\ \sum_{i=0}^{n} u_i B_{i,n}(t/t_{\max}) & \text{if } 0 < t \leq t_{\max} \\ u_n + n(u_n - u_{n-1})(t/t_{\max} - 1) & \text{if } t > t_{\max}, \end{cases} \tag{S3.154}$$

$$\bar{b}(t) = \begin{cases} nf_1 t/t_{\max} & \text{if } t \leq 0 \\ \sum_{i=0}^{n} f_i B_{i,n}(t/t_{\max}) & \text{if } 0 < t \leq t_{\max} \\ f_n + n(f_n - f_{n-1})(t/t_{\max} - 1) & \text{if } t > t_{\max}, \end{cases} \tag{S3.155}$$

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

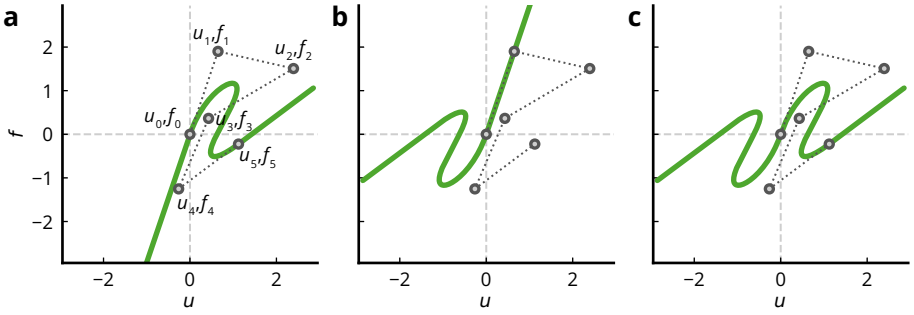


Figure S3.11: Multivalued Bezier force-displacement curves for the tensile **(a)**, compressive **(b)** and symmetric **(c)** modes.

Functions a and b are used to construct an energy potential (Eq. (S3.120)), and compute its gradient (Eqs. (S3.121, S3.122)) and hessian (Eqs. (S3.123, S3.124, S3.125)). To calculate those quantities, the first and second derivatives of a and b must also be provided:

$$a'(t) = \mathcal{F}'(\bar{a})(t), \tag{S3.156}$$

$$b'(t) = \mathcal{F}'(\bar{b})(t), \tag{S3.157}$$

$$a''(t) = \mathcal{F}''(\bar{a})(t), \tag{S3.158}$$

$$b''(t) = \mathcal{F}''(\bar{b})(t), \tag{S3.159}$$

where the functionals $\mathcal{F}'$ and $\mathcal{F}''$ are defined in Eqs. (S3.72, S3.73) and

$$\bar{a}'(t) = \begin{cases} nu_1/t_{\max} & \text{if } t \leq 0 \\ n\sum_{i=0}^{n-1}(u_{i+1}-u_i)B_{i,n-1}(t/t_{\max})/t_{\max} & \text{if } 0 < t \leq t_{\max} \\ n(u_n - u_{n-1})/t_{\max} & \text{if } t > t_{\max}, \end{cases} \tag{S3.160}$$

$$\bar{b}'(t) = \begin{cases} nf_1/t_{\max} & \text{if } t \leq 0 \\ n\sum_{i=0}^{n-1}(f_{i+1}-f_i)B_{i,n-1}(t/t_{\max})/t_{\max} & \text{if } 0 < t \leq t_{\max} \\ n(f_n - f_{n-1})/t_{\max} & \text{if } t > t_{\max}, \end{cases} \tag{S3.161}$$

$$\bar{a}''(t) = \begin{cases} n(n-1)\sum_{i=0}^{n-2}(u_{i+2}-2u_{i+1}+u_i)B_{i,n-2}(t/t_{\max})/t_{\max}^2 & \text{if } 0 < t \leq t_{\max} \\ 0 & \text{else,} \end{cases} \tag{S3.162}$$

$$\bar{b}''(t) = \begin{cases} n(n-1)\sum_{i=0}^{n-2}(f_{i+2}-2f_{i+1}+f_i)B_{i,n-2}(t/t_{\max})/t_{\max}^2 & \text{if } 0 < t \leq t_{\max} \\ 0 & \text{else.} \end{cases} \tag{S3.163}$$

BIVARIATE ZIGZAG BEHAVIOR A bivariate zigzag behavior is described by a smoothed zigzag, similar to its univariate counterpart (Section S3.4.1, paragraph "zigzag behavior"). However, this bivariate version can be defined multivalued curves. The nonlinear behavior defined by the Bezier curve can either describe the tensile behavior (a.k.a 'tensile' mode, Fig. S3.12a), the compressive behavior (a.k.a 'compressive' mode, Fig. S3.12b) or both simultaneously (a.k.a 'symmetric' mode, Fig. S3.12c). Mathematically speaking, the force-displacement curve is defined by

$$u = a(t) = \mathcal{F}(\bar{a})(t) \tag{S3.164}$$

$$f = b(t) = \mathcal{F}(\bar{b})(t) \tag{S3.165}$$

where the functional $\mathcal{F}$ is defined in Eq. (S3.70) and

$$\bar{a}(t) = P(t/t_{\max}, \boldsymbol{s}^a; \boldsymbol{\eta}^a; \eta_s^a), \tag{S3.166}$$

$$\bar{b}(t) = P(t/t_{\max}, \boldsymbol{s}^b; \boldsymbol{\eta}^b; \eta_s^b), \tag{S3.167}$$

with P defined in Eq. (S3.85), $t_{\max} = \sum_{i=1}^{n} |u_i - u_{i-1}|$, $s_i^a = (n-1)(u_{i+1} - u_i)$, $s_i^b = (n-1)(f_{i+1} - f_i)$, $\eta_i^a = \eta_i^b = i/(n-1)$, and $\eta_s^a = \eta_s^b = \epsilon/(2n-2)$.

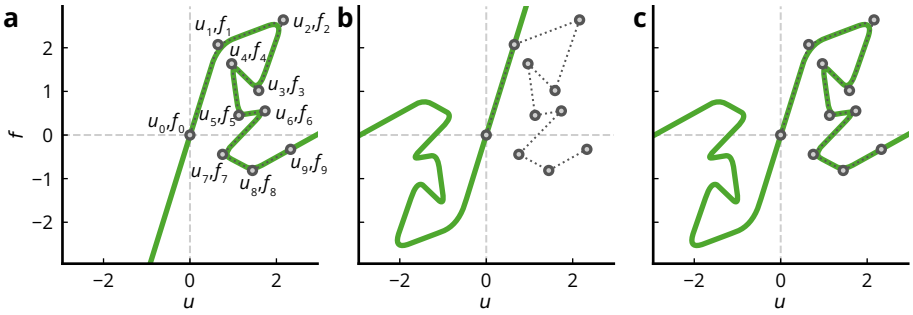


Figure S3.12: Multivalued zigzag force-displacement curves for the tensile **(a)**, compressive **(b)** and symmetric **(c)** modes.

Functions a and b are used to construct an energy potential (Eq. (S3.120)), and compute its gradient (Eqs. (S3.121, S3.122)) and hessian (Eqs. (S3.123, S3.124, S3.125)). To calculate those quantities, the first and second derivatives of a and b must also be provided:

$$a'(t) = \mathcal{F}'(\bar{a})(t), \tag{S3.168}$$

$$b'(t) = \mathcal{F}'(\bar{b})(t), \tag{S3.169}$$

$$a''(t) = \mathcal{F}''(\bar{a})(t), \tag{S3.170}$$

$$b''(t) = \mathcal{F}''(\bar{b})(t), \tag{S3.171}$$

where the functionals $\mathcal{F}'$ and $\mathcal{F}''$ are defined in Eqs. (S3.72, S3.73) and

$$\bar{a}'(t) = \frac{1}{t_{\max}} \frac{\partial P}{\partial \eta}(t/t_{\max}, \boldsymbol{s}^a; \boldsymbol{\eta}^a; \eta_s^a), \tag{S3.172}$$

$$\bar{b}'(t) = \frac{1}{t_{\max}} \frac{\partial P}{\partial \eta}(t/t_{\max}, \boldsymbol{s}^b; \boldsymbol{\eta}^b; \eta_s^b), \tag{S3.173}$$

$$\bar{a}''(t) = \frac{1}{t_{\max}^2} \frac{\partial^2 P}{\partial \eta^2}(t/t_{\max}, \boldsymbol{s}^a; \boldsymbol{\eta}^a; \eta_s^a), \tag{S3.174}$$

$$\bar{b}''(t) = \frac{1}{t_{\max}^2} \frac{\partial^2 P}{\partial \eta^2}(t/t_{\max}, \boldsymbol{s}^b; \boldsymbol{\eta}^b; \eta_s^b), \tag{S3.175}$$

with $\partial P/\partial\eta$ defined in Eq. (S3.92), and

$$\frac{\partial^2 P}{\partial \eta^2}(\eta, \boldsymbol{s}, \boldsymbol{\eta}, \eta_\mathrm{s}) = \begin{cases} 0 & \text{if } \eta \leq \eta_0 - \eta_\mathrm{s} \\ 2a_0 & \text{if } \eta_0 - \eta_\mathrm{s} < \eta < \eta_0 + \eta_\mathrm{s} \\ 0 & \text{if } \eta_0 + \eta_\mathrm{s} \leq \eta \leq \eta_1 - \eta_\mathrm{s} \\ \vdots & \vdots \\ 2a_i & \text{if } \eta_i - \eta_\mathrm{s} < \eta < \eta_i + \eta_\mathrm{s} \\ 0 & \text{if } \eta_{i-1} + \eta_\mathrm{s} \leq \eta \leq \eta_i - \eta_\mathrm{s} \\ \vdots & \vdots \\ 2a_{n-2} & \text{if } \eta_{n-2} - \eta_\mathrm{s} < \eta < \eta_{n-2} + \eta_\mathrm{s} \\ 0 & \text{if } \eta_{n-2} + \eta_\mathrm{s} \leq \eta < +\infty, \end{cases} \tag{S3.176}$$

with a_i defined in Eq. (S3.87).

S3.5 Installation and simulation

S3.5.1 Installation

Simulations shown in this work have been produced with the version 1.0.0 of the `springable` toolkit. It is available on PyPI and can then easily be installed by running

```
python -m pip install springable==1.0.0
```

on Windows, or

```
python3 -m pip install springable==1.0.0
```

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

S3.5.2 Simulating a model

To simulate a model, you can execute the Python script shown in Code S3.1,

```python
from springable.simulation import simulate_model

simulate_model('my_model.csv')
```

Code S3.1: Python script `run_simulation.py` used to start a simulation.

which can be done 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 S3.6. The CSV model files used in Chapter 3 are provided in the Section S3.7.

S3.6 Model file format specification

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 flexel: `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 S3.7. For instructions on how to install the `springable` package, you can refer to Section S3.5.1.

S3.6.1 The `NODES` section

The `NODES` section starts 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.

S3.6.2 The FLEXELS sections

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

- a length (Section S3.3.1, Fig. S3.1a) are grouped under the section named `LONGITUDINAL FLEXELS`;
- an angle (Section S3.3.2, Fig. S3.1b) under the section named `ANGULAR FLEXELS`;

- an area (Sections S3.3.4, S3.3.9, Fig. S3.1c,g) under the section named `AREA FLEXELS`;
- a signed x-distance (Section S3.3.5, Fig. S3.1d) under the section named `X DISTANCE FLEXELS`;
- a signed y-distance (Section S3.3.5, Fig. S3.1d) under the section `Y DISTANCE FLEXELS`;
- a signed point-line distance (Section S3.3.7, Fig. S3.1e) under the section named `DISTANCE FLEXELS`;
- a path length (Section S3.3.8, Fig. S3.1f) under the section named `PATH FLEXELS`.

Under each section, a flexel is specified as follows:

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

where

- `<list of node indices>` specifies the nodes coupled by the flexel (see paragraph S3.6.2 for more details);
- `<nonlinear behavior>` (together with `<natural measure>`) specifies the intrinsic nonlinear behavior of the flexel (see paragraph S3.6.2 for more details);
- `<natural measure>` describes α_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 geometric measure computed from the initial positions of the nodes provided in the `NODES` section.

EXAMPLE Two longitudinal flexels couple 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 π 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
```

Specifying the flexel's nodes

- For a longitudinal flexel (see Section S3.3.1), use

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

- For an angular flexel (see Section S3.3.2), use

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

- For an area flexel (see Section S3.3.4), use

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

- For an x- (or a y-) distance flexel (see Section S3.3.5), use

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

- For a distance flexel (see Section S3.3.7), use

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

- For a path flexel (see Section S3.3.8), use

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

- For an area-with-holes flexel (see Section S3.3.9), use

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

  Example:

  ```
  (0-1-2-3-4)-(5-6-7)-(8-9-10-11)
  ```

  Note that flexels whose geometric measures are areas or an areas 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.

Specifying the flexel's intrinsic nonlinear behavior

- For a linear behavior (see Eq. (S3.62)), use

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

  where <k> is the spring constant.

- For a logarithmic behavior (see Eq. (S3.66)), use

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

- For a univariate Bezier behavior (see Eq. (S3.74)), use

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

  <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 C_1-piecewise beahvior (see Eq. (S3.83)), use

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

  <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 univariate zigzag behavior (see Eq. (S3.96)), use

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

  <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 contact behavior (see Eq. (S3.105)), use

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

- For an isothermal behavior (see Eq. (S3.110)), use

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

- For an isentropic behavior (see Eq. (S3.115)), use

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

- For a bivariate Bezier behavior (see Eqs. (S3.152, S3.153)), use

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

  <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 (see Eqs. (S3.164, S3.165)), use

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

  <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'.

Nonlinear behaviors can be interactively tuned and created using the 'behavior creation' graphical interface (Fig. S3.13), which can be started by running the following Python script

```python
from springable.behavior_creation import start

start()
```

Code S3.2: Python script to start the graphical interface and create custom force-displacement curves, as shown in Fig. S3.13.

Note 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:

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

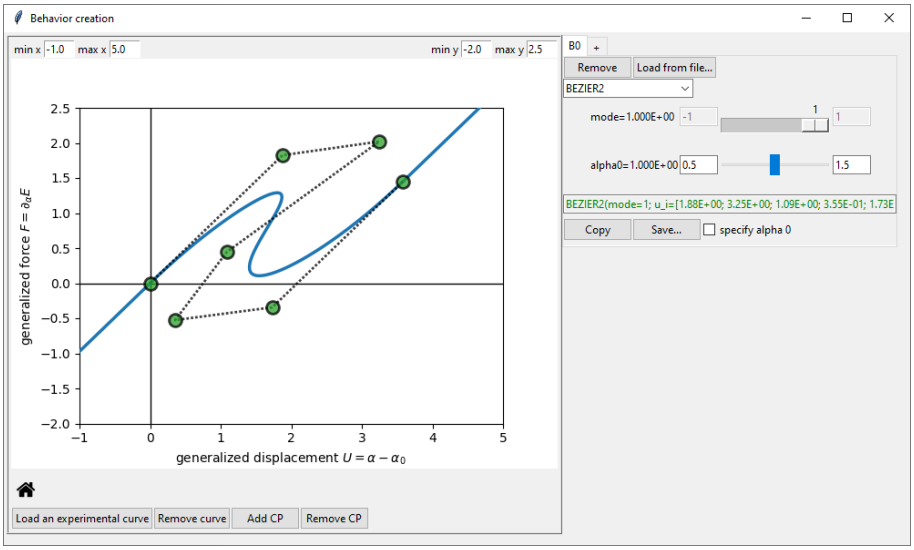


Figure S3.13: Screenshot of the behavior creation graphical interface, which can be used to design custom nonlinear behaviors and generate the corresponding code to be pasted or imported in the model CSV file. The interface is started by running the Python script given in Code S3.2.

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 to the CSV model file instead (for example `relative/path/to/behavior.csv`), 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')
```

S3.6.3 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 $\tilde{\boldsymbol{F}}^{\text{dir}}$ that remains constant during the entire load step (Eq. (S3.13)). 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. (S3.6)) 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 of 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 blocked 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.

S3.6.4 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 π.
- 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).

S3.7 Model descriptions

The model CSV files used in Chapter 3 are provided in this section. To simulate all those models, the Python script given in Code S3.3 can be run.

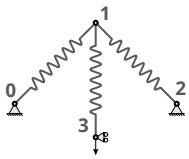
```
from springable.discover import run_simulations_from_article

run_simulations_from_article()
```

Code S3.3: Python script to run all the simulations of the models shown in Chapter 3 at once.

To simulate the model of each individual figure, you can refer to the Python scripts specified at the end of each Section S3.7.X (see herein below). For instructions on how to install the `springable` package, you can refer to Section S3.5.1.

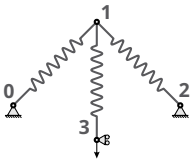
S3.7.1 Models and behaviors from Fig. 3.1

```
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)
```



Code S3.4: Nonmonotonic Von-Mises truss, model CSV file from Fig. 3.1a (`nonmonotonic_vonmises_truss.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=1.0)
1-2, LINEAR(k=1.0)
1-3, LINEAR(k=0.33)
LOADING
3, Y, -0.5, -1.2*2*l*SIN(alpha/180*PI)
```

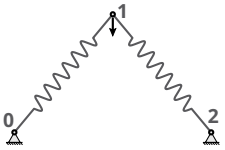


Code S3.5: Multivalued Von-Mises truss, model CSV file from Fig. 3.1b (`multivalued_vonmises_truss.csv`).

```
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('nm_vonmises_behavior.csv')
2-1, FROMFILE('mv_vonmises_behavior.csv')
LOADING
2, X, -0.325, -3.5
```

Code S3.6: Nonlinear flexels in series, model CSV file from Fig. 3.1e (`nl_flexels_series.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('nm_vonmises.csv')
1-2, FROMFILE('mv_vonmises.csv')
LOADING
1, Y, -2.0, -5.5
```



Code S3.7: Arch of nonlinear flexels, model CSV file from Fig. 3.1f (`nl_flexels_arch.csv`).

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

Code S3.8: Nonlinear behavior from Fig. 3.1c (`nm_vonmises_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)
```

Code S3.9: Nonlinear behavior from Fig. 3.1d (`mv_vonmises_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)
```

Code S3.10: Nonlinear behavior from Fig. 3.1g (`nl_series_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; -1.356
    E-01; -5.714E-02; 1.830E-01]; epsilon=7.500E-01; mode=-1)
```

Code S3.11: Nonlinear behavior from Fig. 3.1h (`nl_arch_behavior.csv`).

Simulation script

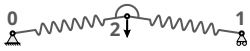
To simulate the models related to Fig. 3.1, the script shown in Code S3.12 can be run. The CSV behavior files must live in the same (working) directory as the CSV model files.

```python
from springable.simulation import simulate_model

simulate_model('nonmonotonic_vonmises_truss.csv')
simulate_model('multivalued_vonmises_truss.csv')
simulate_model('nl_flexels_series.csv')
simulate_model('nl_flexels_arch.csv')
```

Code S3.12: Python script to run to simulate the four models from Fig. 3.1.

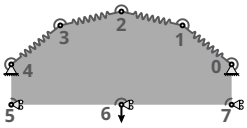
S3.7.2 Models and behaviors from Fig. 3.3

```
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('tape_spring_behavior.csv')
LONGITUDINAL FLEXELS
0-2, LINEAR(k=1.0)
2-1, LINEAR(k=1.0)
LOADING
2, Y, -5.0
```



Code S3.13: Example of nonlinear angular flexel, model CSV file from Fig. 3.3a (`nl_angular_flexel.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=kl)
1-2, LINEAR(k=kl)
2-3, LINEAR(k=kl)
3-4, LINEAR(k=kl)
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
```

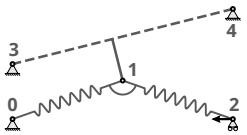


Code S3.14: Example of nonlinear area flexel, model CSV file from Fig. 3.3b (`nl_area_flexel.csv`).

```
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
```

Code S3.15: Example of nonlinear path flexel, model CSV file from Fig. 3.3c (`nl_path_flexel.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
```



Code S3.16: Example of nonlinear distance flexel to model contact, model CSV file from Fig. 3.3d (`nl_distance_flexel.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])
```

Code S3.17: Nonlinear behavior used by the angular flexel from Fig. 3.3a. It describes a multivalued torque-angular displacement curve (`tape_spring_behavior.csv`).

Simulation script

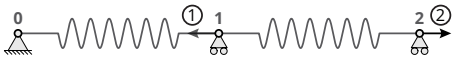
To simulate the models related to Fig. 3.3, the script shown in Code S3.18 can be run. The CSV behavior file must live in the same (working) directory as the CSV

model files.

```
from springable.simulation import simulate_model

simulate_model('nl_angular_flexel.csv')
simulate_model('nl_area_flexel.csv')
simulate_model('nl_path_flexel.csv')
simulate_model('nl_distance_flexel.csv')
```

Code S3.18: Python script to run to simulate the four models from Fig. 3.3.

S3.7.3 Model and behaviors from Fig. 3.4




```
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('bottom_block_behavior.csv')
1-2, FROMFILE('top_block_behavior.csv')
LOADING
1, X, -2.0*9.81/1000
then
2, X, 1.0, 25
```

Code S3.19: Serially-coupled nonlinear building blocks, model CSV file from Fig. 3.4 (`nl_blocks_in_series.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])
```

Code S3.20: Nonlinear behavior of the bottom block from Fig. 3.4 (`bottom_block_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])
```

Code S3.21: Nonlinear behavior of the top block from Fig. 3.4 (`top_block_behavior.csv`).

Simulation script

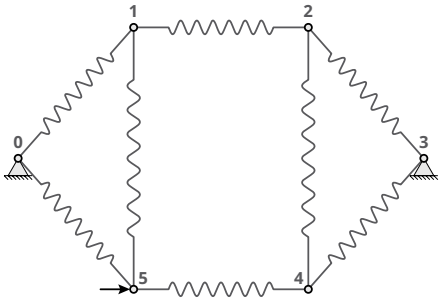
To simulate the model related to Fig. 3.4, the script shown in Code S3.22 can be run. The CSV behavior files must live in the same (working) directory as the CSV model files.

```
from springable.simulation import simulate_model

simulate_model('nl_blocks_in_series.csv')
```

Code S3.22: Python script to run to simulate the model from Fig. 3.3.

S3.7.4 Models and behaviors from Fig. 3.5

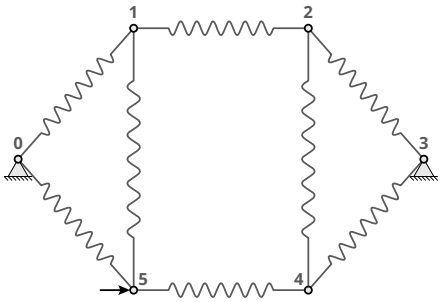
Tensegrities (Fig. 3.5a)




```
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
```

Code S3.23: Tensegrity without pre-stress, model CSV file from Fig. 3.5a-top (`tensegrity_wo_prestress.csv`).



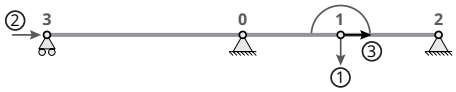

```
PARAMETERS
k, 0.5
a, 1.0
b, 1.0
c, 1.5
prestress, 0.3
cable_behavior, 'tensegrity_cable_behavior.csv'
NODES
0, 0.0, 0.0, 1, 1
1, X0+COS(ARCSIN(c/2/a))*a, Y0+c/2, 0, 0
2, X1+b, Y1, 0, 0
3, X2+COS(ARCSIN(c/2/a))*a, Y0, 1, 1
4, X2, -Y2, 0, 0
5, X1, -Y1, 0, 0
LONGITUDINAL FLEXELS
# cables
0-1, FROMFILE(cable_behavior),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

Code S3.24: Tensegrity with pre-stress, model CSV file from Fig. 3.5a-bottom (`tensegrity_w_prestress.csv`).

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

Code S3.25: Nonlinear behavior used in models of Fig. 3.5a (`tensegrity_cable_behavior.csv`).

Tape spring (Fig. 3.5b)




```
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
```

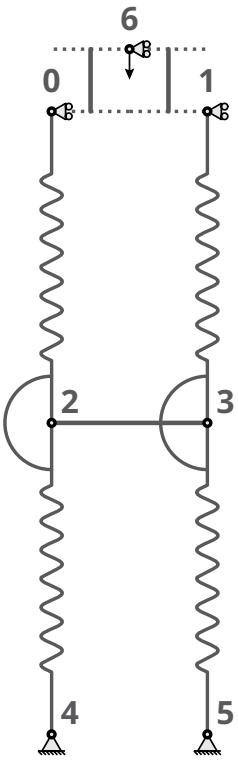
Code S3.26: Tape spring, model CSV file from Fig. 3.5b (`tape_spring.csv`).

```
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)
```

Code S3.27: Nonlinear behavior used in the model of Fig. 3.5b (`tape_behavior.csv`).

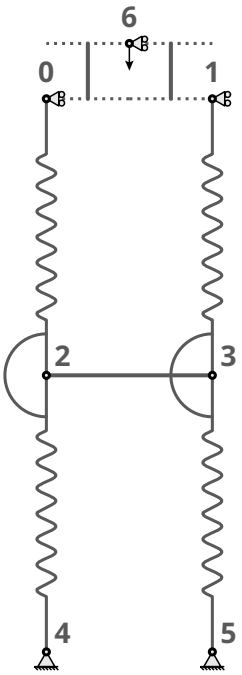
Interacting buckled beams (Fig. 3.5c)

```
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
```



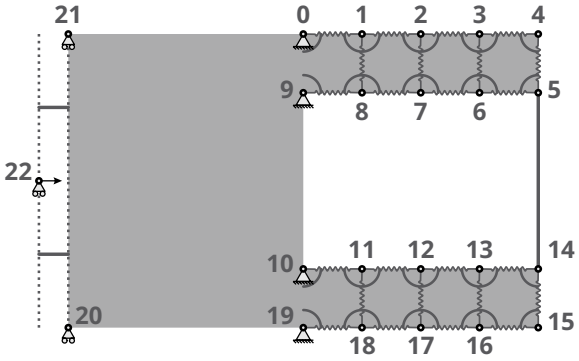
Code S3.28: Interacting beams buckling to the right $d = 0.25$, model CSV file from Fig. 3.5c-top (`buckling_right.csv`).

```
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
```



Code S3.29: Interacting beams buckling to the left $d = 0.30$, model CSV file from Fig. 3.5c-bottom (`buckling_left.csv`).

Metafluidic gripper (Fig. 3.5d)



```
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)
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
```

Code S3.30: Soft gripper atuated via a metafluid and gripping an object, model CSV file from Fig. 3.5d (`metafluidic_gripper.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)
```

Code S3.31: Nonlinear behavior of the metafluid, used in the model of Fig. 3.5d (`metafluid_behavior.csv`).

Simulation script

To simulate the model related to Fig. 3.5, the script shown in Code S3.32 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('tensegrity_wo_prestress.csv', solver_settings=settings)
simulate_model('tensegrity_w_prestress.csv', solver_settings=settings)
simulate_model('tape_spring.csv', solver_settings=settings)
simulate_model('buckling_right.csv', solver_settings=settings)
simulate_model('buckling_left.csv', solver_settings=settings)
simulate_model('metafluidic_gripper.csv', solver_settings=settings)
```

Code S3.32: Python script to run to simulate the six models from Fig. 3.5.

4

Exploring the interaction of highly nonlinear building blocks for geometric rectification

4.1 Introduction

In Chapter 2, we developed structures with new intrinsic mechanical behaviors, allowing for deformation pathways and properties not attainable in other mechanical systems. Using the framework developed in Chapter 3, we can now treat these highly nonlinear structures directly as building blocks (instead of compound assemblies of lower-level components) and simulate the interplay of many of those blocks with more standard elements. We are therefore now equipped with effective tools to build reduced-order models of highly nonlinear mechanical systems, explore their complex behaviors and facilitate understanding.

In this chapter, we will focus on a particularly useful attribute of nonlinear structures: their shape-morphing properties [124]. By carefully designing the interplay between geometry and elastic energy, the response to external forces can translate into non-trivial changes of curvature [125,126] or desired, non-uniform surface patterns for example [127, 128]. When their energy landscape is complex enough to be characterized by multiple minima, nonlinear structures can even switch among a finite set of stable configurations upon loading [68, 129–131]. Designing those complex energy landscapes through architecture has enabled structures in an initial compact or "ground" state to self-lock [132], reconfigure [48]

or rapidly deploy [133] into a new targeted shape. Research efforts have mainly focused on transforming a geometrically simple baseline configuration into a desired, known, complex and more functional one. Thus, the reverse problem naturally arises: how can unknown, complex, undesired shapes be transformed into a baseline state? We will define this transformation as *the geometric rectification process*.

Even though that question has comparatively received little attention, it is of crucial importance in alignment and positioning applications. In particular, nanolithography processes require to flatten potentially warped wafers to ensure overlay precision. Current solutions often rely on pushing such undesirably-shaped plates on an array of stiff pillars that sustain minimal deformation. While straightforward, this approach does not permit to rectify the plate fully and suffers from wear induced by sliding friction at the interface between the object and the pillars, which affects the repeatability of the process in the long run. These limitations can potentially be addressed by strategies involving more compliant nonlinear components with shape-morphing properties.

However, unlike other more traditional shape-morphing problems, the rectification problem deals with initially warped objects that are unknown and therefore cannot be designed. Drawing inspiration from the current technological solution used in nanolithography, a potential strategy is to instead design for the reaction forces of a nonlinear support structure against which the object would be pushed. The resulting external loads felt by the initially warped object would then rectify it to the desired baseline configuration.

For the purpose of this exploratory investigation into geometric rectification, we will focus on the case of an initially curved beam and examine the rectification process when uniformly pushed against a set of highly nonlinear pillars. How flat can the beam become? What are the deformation pathways toward a rectified state? Could nonlinearities brought by soft pillars improve the rectification compared to stiff ones?

To address those questions, we start with a simplified model consisting of two pillars from which equations describing the rectification performance can be derived. Those equations form a basis to interpret numerical simulations that we will conduct next, where we examine the deformation pathways toward possible rectification for different types of snapping and countersnapping pillars. Finally,

by simulating the rectification of various initially warped beams, we can evaluate the performance of the soft-pillar approach at rectifying initially unknown configurations.

Beyond the specifics of the rectification problem, this chapter showcases how mechanical systems subject to complex reconfiguration pathways can be simulated using the flexel framework and analyzed through transition graphs that can be constructed using a robust and reliable scheme.

4.2 Rigid bar pushed against two pillars

We first consider a rigid bar of length l initially inclined by an angle θ_0 with respect to the horizontal direction, which can be changed if energy is provided to deform an attached angular spring (Fig. 4.1).

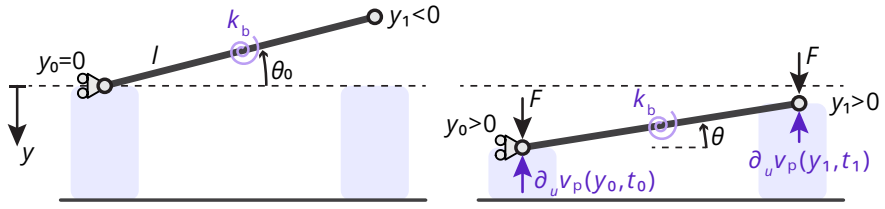


Figure 4.1: Model of the inclined rigid bar pushed against two pillars, used for the analytical study of the rectification. Left: initial configuration; before the force load F is applied. Right: configuration when the bar is in contact with both pillars, at equilibrium with the applied load F.

This system acts as a simplified model for the segment of a curved beam. Understanding how such a system can be rectified will provide useful insight into the case of the whole curved beam.

4.2.1 Analytical study

To study this system analytically, let us first derive the governing equations. Those are obtained by first expressing the total potential energy Π of the system

$$\Pi(y_0, y_1, t_0, t_1) = v_p(u_0(y_0), t_0) + v_p(u_1(y_1), t_1) + \frac{1}{2} k_b \left(\theta(y_0, y_1) - \theta_0\right)^2 - F(y_0 + y_1), \tag{4.1}$$

where y_i are the vertical coordinates of the bar ends, u_i and t_i are the compression displacement and the internal coordinate that unambiguously describes the

state of pillar i, whose energy is governed by the potential $v_p(u,t)$, k_b is the spring constant of the angular spring, θ is the inclination of the bar with the horizontal and F is the magnitude of each downward force acting on the bar ends. The compression displacements u_i are functions of the bar ends' vertical coordinates

$$u_0(y_0) = \max(0, y_0), \tag{4.2}$$

$$u_1(y_1) = \max(0, y_1). \tag{4.3}$$

To find the equilibrium points of the system, we need to find the stationary points of the total potential energy Π. When both pillars are in contact (i.e. $y_i > 0 \Leftrightarrow u_i = y_i$), such points must necessarily satisfy

$$0 = \frac{\partial \Pi}{\partial y_0} = \frac{\partial v_p}{\partial u}(y_0, t_0) + k_b(\theta - \theta_0)\frac{\partial \theta}{\partial y_0} - F, \tag{4.4}$$

$$0 = \frac{\partial \Pi}{\partial y_1} = \frac{\partial v_p}{\partial u}(y_1, t_1) + k_b(\theta - \theta_0)\frac{\partial \theta}{\partial y_1} - F. \tag{4.5}$$

By subtracting Eq. (4.5) to Eq. (4.4), we obtain

$$\frac{\partial v_p}{\partial u}(y_0, t_0) - \frac{\partial v_p}{\partial u}(y_1, t_1) + \frac{2k_b}{l\cos\theta}(\theta - \theta_0) = 0, \tag{4.6}$$

given that $\sin\theta = (y_0 - y_1)/l \Rightarrow \partial\theta/\partial y_i = (-1)^i/(l\cos\theta)$.

From Eq. (4.6), two important insights can be gained, as detailed in the two following paragraphs.

Linear pillars First, if the pillars are linear with a spring constant k, $v_p(u,t) = v_p(u) = ku^2/2$, Eq. (4.6) yields

$$\begin{aligned} 0 &= k(y_0 - y_1) + \frac{2k_b}{l\cos\theta}(\theta - \theta_0) \\ &= kl\sin\theta + \frac{2k_b}{l\cos\theta}(\theta - \theta_0) \\ &= \mu\sin\theta\cos\theta + 2(\theta - \theta_0), \end{aligned} \tag{4.7}$$

where $\mu := kl^2/k_b$. From Eq. (4.7), we notice that the inclination angle θ is solely determined by the problem parameters k_b, k, l, θ_0 and is independent of the load F as soon as both pillars are in contact. Eq. (4.7) defines a relation for θ as a function

of μ and θ_0, which is represented in Fig. 4.2.

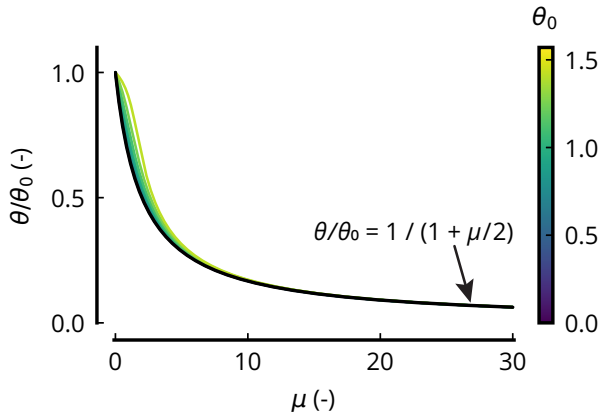


Figure 4.2: Relative inclination angle θ/θ_0 as a function of the rectification compliance μ for different values of initial inclination θ_0, when both pillars are linear. The black line represents the analytical solution obtained under the small-angle assumption ($\theta \ll 1$).

As it can be seen in Fig. 4.2, the dimensionless quantity μ describes the capability of the system to rectify the bar, and will be therefore named the *rectification compliance*. If $\mu \gg 1$, the inclination angle is greatly reduced on linear pillars. Conversely, if $\mu \ll 1$, the inclination angle remains close to its initial value θ_0. If we assume that the angle θ is small, we have $\sin\theta \sim \theta$ and $\cos\theta \sim 1$, and Eq. (4.7) simplifies to

$$\mu\theta + 2(\theta - \theta_0) = 0 \Leftrightarrow \theta/\theta_0 = \frac{1}{1+\mu/2}. \tag{4.8}$$

As shown in Fig. 4.2, this relation gives a good approximation of the relative angle θ/θ_0, especially when $\mu \gg 1$ or when $\theta_0 \ll 1$, conditions leading to a small angle θ. Importantly, reaching a perfectly flat configuration is in general impossible on linear pillars, as it would require infinitely stiff pillars ($\mu = +\infty$).

Improved rectification on multivalued pillars Second, from Eq. (4.6), we notice that a complete rectification, i.e. $\theta = 0 \Leftrightarrow y_0 = y_1 = y^\star$, is possible only if

$$\frac{\partial v_\mathrm{p}}{\partial u}(y^\star, t_0) - \frac{\partial v_\mathrm{p}}{\partial u}(y^\star, t_1) = \frac{2k_\mathrm{b}}{l}\theta_0. \tag{4.9}$$

In other words, both pillars (which are identical) must provide a different force $\partial v_\mathrm{p}/\partial u$ despite being compressed by the same amount $y^\star$. This is only possible if they are different, or characterized by a multivalued force-displacement curve, with the state of each pillar being on separate stable branches. Along a stable branch X, the force is unambiguously described by

$$\partial_u v(u,t) = f_\mathrm{X}(u), \tag{4.10}$$

where f_X is the force at equilibrium when the pillar is compressed by u and on stable branch X. The rectification on such pillars could in principle be enhanced. Let's assume that the left and right pillars are compressed by $y_0 = y^\star + (1-\gamma)\Delta y$ and $y_1 = y^\star - \gamma\Delta y$ (with $\Delta y = y_0 - y_1 = l\sin\theta$) while being on branches A and B, respectively. Then, the equilibrium equation becomes

$$\begin{aligned}&\left(f_\mathrm{A}(y^\star) + \sum_{k=1}^{\infty} \frac{\partial^k f_\mathrm{A}}{\partial u^k}(y^\star)(1-\gamma)^k l^k \sin^k\theta/k!\right)\\ &-\left(f_\mathrm{B}(y^\star) + \sum_{k=1}^{\infty}(-1)^k \frac{\partial^k f_\mathrm{B}}{\partial u^k}(y^\star)\gamma^k l^k \sin^k\theta/k!\right) = \frac{2k_\mathrm{b}}{l\cos\theta}(\theta_0-\theta)\end{aligned} \tag{4.11}$$

using Taylor series to represent the force-displacement relation of each branch. In case there is no strong nonlinearity along each stable branch (i.e. $\partial_u^2 f_\mathrm{X}(u) l^2 \sin^2\theta \ll f_\mathrm{X}(u)$ on stable equilibria), we can reasonably truncate the Taylor series and obtain the following approximation:

$$f_\mathrm{A}(y^\star) - f_\mathrm{B}(y^\star) + (1-\gamma) l\sin\theta \frac{\partial f_\mathrm{A}}{\partial u}(y^\star) + \gamma l\sin\theta\frac{\partial f_\mathrm{B}}{\partial u}(y^\star) = \frac{2k_\mathrm{b}}{l\cos\theta}(\theta_0-\theta)$$

$\Leftrightarrow$

$$\mu_\gamma(y^\star)\sin\theta\cos\theta + \nu(y^\star)\cos\theta + 2(\theta-\theta_0) = 0, \tag{4.12}$$

where $\mu_\gamma(y^\star) := k_\gamma l^2/k_\mathrm{b}$ and $\nu(y^\star) := \Delta f(y^\star) l/k_\mathrm{b}$, with $k_\gamma := (1-\gamma)\partial_u f_\mathrm{A}(y^\star) + \gamma\partial_u f_\mathrm{B}(y^\star)$ and $\Delta f(y^\star) := f_\mathrm{A}(y^\star) - f_\mathrm{B}(y^\star)$. Under the small angle assumption, it simplifies to

$$\theta/\theta_0 = \frac{1}{1+\mu_\gamma(y^\star)/2}\left(1 - \frac{\nu(y^\star)}{2\theta_0}\right) \tag{4.13}$$

From Eq. (4.13), we see that there exist states leading to improved rectification compared to the linear case when the left pillar is on a branch providing more force than the force provided by the branch of the right pillar, at a given compression displacement ($\gamma > 0$).

Note that if both pillars are on the same branch, that is if A = B, then $\mu_\gamma = \partial_u f_\mathrm{A}(y^\star) l^2 / k_\mathrm{b}$ and $\Delta f = 0$, and we notice that the relative angle is solely determined by the tangent stiffness along that branch, reducing this case to the linear case (Eq. (4.7)), locally.

4.2.2 Numerical study

Whereas we proved the existence of states with improved rectification thanks to the analytical study, it is still an open question to assess whether such states are reachable. What are the conditions that the force-displacement curve of the pillars must satisfy for a state with improved rectification to be reached by simply controlling the load F?

Determining snapping transitions To address this question, it is important to understand how a mechanical system reconfigures as it is loaded and unloaded. A highly nonlinear system can in general be multistable under a given load. Those stable states correspond to minima of the energy landscape. Upon slow loading and unloading, this energy landscape is continuously shaped, causing minima to move, appear and vanish. Under quasistatic conditions, we assume that the state of the system is always located at one of the minima of the energy landscape. If that minimum continuously moves in the state space, the state of the system will follow it under quasistatic conditions. However, if that minimum vanishes, the system needs to find another minimum to recover a new stable configuration. The minimum that will be the home of the next stable state cannot be determined from the static equilibrium equations alone, as this reconfiguration process is by nature a dynamic process influenced by inertia and damping. We can however append a rule to the static equations to determine the next state.

The hysteron model provides a framework to establish such a rule, and has been successful in determining the state reconfigurations of a wide variety of physical systems, subject to non-trivial transitions such as avalanches [52, 134, 135]. However, because this approach relies on simplified abstract entities and

interaction terms, some physical systems can be difficult or even impossible to model using hysteron models [110]. Moreover, it can face ill-defined scenarios where no single valid transition can be easily determined [57, 58, 136].

Here, we leverage the information computed by the arclength algorithm capable of not only finding stable branches but also unstable ones. The method essentially tracks a stationary point in the energy landscape, which yields a continuous path in the space state. Importantly, moving along this path continuously deforms the structure without discontinuities. The distance run along this path therefore provides a way to measure "how far" configurations are from each other. Going from one stable branch to another is done continuously by running along unstable branches, which provides a way to order stable branches. We leverage these properties to establish the rules that allow us to find the load-driven deformation path given the arclength deformation path.

Determining the load-driven deformation path from the arclength path

Given an initial stable state on a stable branch, the next state reached upon loading (unloading) is

- the next (previous) equilibrium along the branch as long as no unstable state is encountered;
- the next (previous) equilibrium along the arclength path that
  - is stable, *and*
  - is characterized by the same load, *and*
  - leads to a transition where the change in elastic energy stored in the structure is less than the work done by the external forces during the transition,

  when a critical point is encountered.

This rule is formalized mathematically in Section S4.1, and can be used directly to determine transitions in mechanical systems model using the flexel formulation introduced in Chapter 3.

To obtain the arclength deformation path needed to determine the transitions

and the load-driven deformation pathways, we first need to build the two-pillar model using flexels, as shown in Fig. 4.3. Each pillar can be modeled as a y-distance flexel with a given, possibly multivalued, force-distance curve. Contact between the bar ends and the pillars is modeled with y-distance flexel with contact behaviors, the rigid bar with a stiff linear longitudinal flexel (that is, a stiff spring) and the bending stiffness using a linear angular flexel.

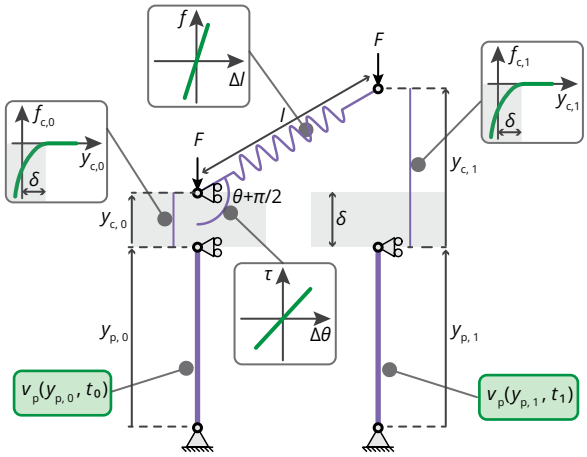


Figure 4.3: Flexel model used to studying the two-pillar bar rectification numerically.

We consider three cases: linear, regular snapping and countersnapping.

LINEAR CASE For the linear case, we assign a linear behavior to the pillars characterized by a stiffness k. We study the inclination of the plate for different values of k, for a starting angle $\theta_0 = 10^o$, $k_b = 0.1$, $l = 1$.

In Fig. 4.4a, we see that the angle decreases then remains constant as the force increases. This transition corresponds to the instant where the bar impacts the second pillar, which leads to equilibria where the angle is independent of the force, as expected from the analytical solution obtained in the previous section in the case of linear pillars (Eq. (4.7)).

As shown in Fig. 4.4b, the relation between the final angle and rectification compliance μ shows good agreement with the analytical solution (Eq. (4.8)). This confirms that perfect rectification cannot be obtained with linear pillars.

For applications where low rectification angles are desired, restricting the de-

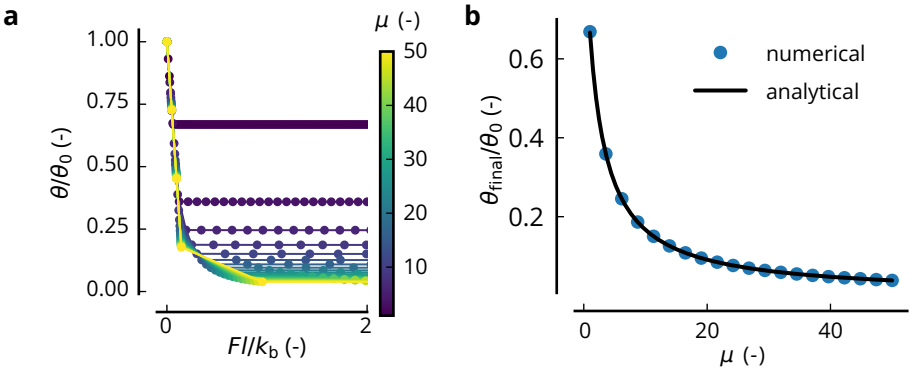


Figure 4.4: Rectification on linear pillars. **(a)** Relative angle θ/θ_0 as a function of the dimensionless force Fl/k_b. **(b)** Final relative angle θ/θ_0 as a function of the rectification compliance.

sign to linear pillars might be constraining as it might require extremely stiff pillars. As we will show next, allowing for pillars with multivalued displacement curves relaxes the constraints on the stiffness of the pillars allowing for greater rectification on softer pillars.

Regular snapping pillars For the case of regular snapping pillars, we assign a multivalued curve characterized by two branches labeled '0' and '1', characterized by transitions at $(u^-, u^+) \approx (0.5, 1.0)$ with corresponding force offsets of Δf^+ and Δf^-, with respective stiffness k_0 and $k_1 = k_0 + (\Delta f^+ - \Delta f^-)/(u^+ - u^-)$, as shown in Fig. 4.5a. We study the effect of Δf^+ and Δf^- on the deformation pathways. Snapping transitions are determined according to the rules detailed in Section S4.1 (Chapter S4).

We observed four qualitatively different transition graphs (Fig. 4.5b-c). Interestingly, sequential hysteresis (difference between the loading and unloading sequence) is only observed using pillars where $k_0 \neq k_1$. More important to the context of rectification, none of the transition graphs leads to state '01', the state related to the rectified state as described analytically in the previous section. The lowest relative angle θ/θ_0 is always achieved on one of the symmetric states ('00' or '11'), as shown in Fig. 4.5d. In those symmetric states, pillars are on the same branch and the rectification is identical to the linear case ($\nu = 0$, Eq. (4.13)). The asymmetric state '10' worsens rectification as the left pillar is on a branch yielding *less* force than the right pillar, leading to $\nu < 0$ (Eq. (4.13)), while improved

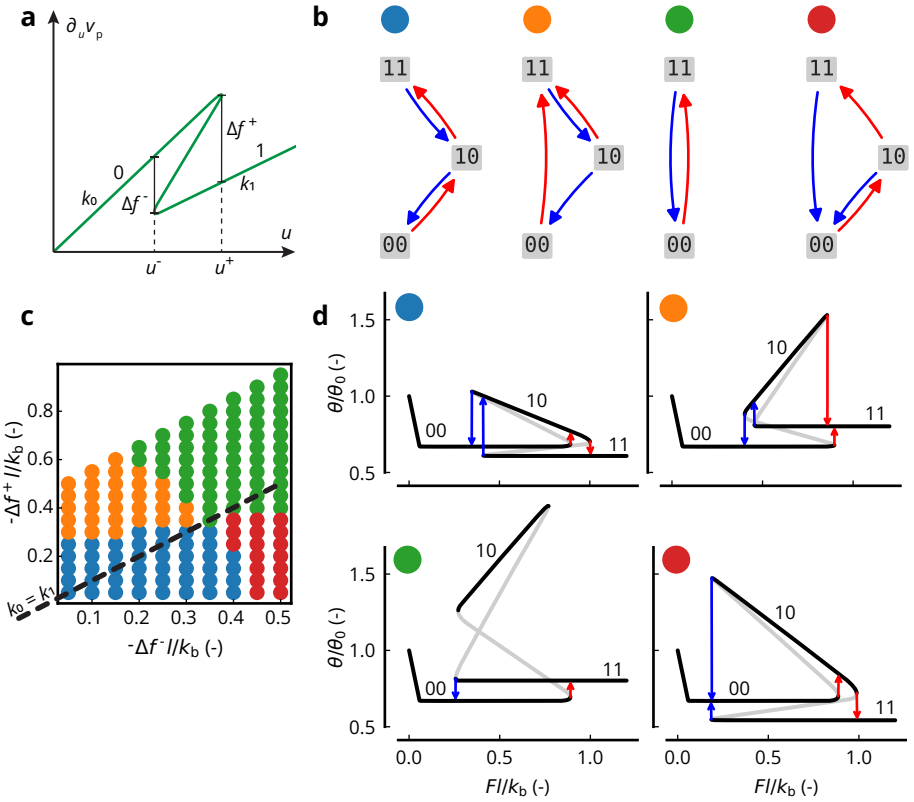


Figure 4.5: Rectification on snapping pillars. **(a)** Multivalued force-displacement characterizing the behavior of the pillars. It is represented as a zigzag2 bivariate behavior (see Chapter 3, Section S3.4.2). **(b)** Transition graphs obtained when scanning Δf^+ and Δf^-. **(c)** Map of transition graphs in the $(\Delta f^- l/k_\mathrm{b}, \Delta f^+ l/k_\mathrm{b})$ space. **(d)** Relative angle θ/θ_0 as a function of the dimensionless force Fl/k_b for $(\Delta f^- l/k_\mathrm{b}, \Delta f^+ l/k_\mathrm{b}) = \{(0.2, 0.05); (0.1, 0.35); (0.3, 0.55); (0.45, 0.1)\}$, from top to bottom, from left to right.

rectification would necessitate $\nu > 0$, only achievable in the '01' state, that is not reachable. In summary, regular snapping pillars do not allow to reach a state leading to improved rectification.

Countersnapping pillars In the previous paragraph, we observed that the left pillar always transitions from the '0'-branch to the '1'-branch during the first transition upon loading. Because the '1'-branch did not provide more force than the '0'-branch, it did not lead to improved rectification. It is therefore natural to study the rectification on pillars where the '1'-branch can generate more force than the '0'-branch. This is possible in principle with pillars characterized by a self-intersecting force-displacement curve, that is, a *countersnapping behavior* (Chapter 2). We assign to the pillars a curve from which the stiffness k_1 can be tuned in such a way that sufficiently high values lead to a countersnapping behavior (Fig. 4.6a). Note that too large k_1 values lead to unphysical force-displacement curves, as the snapping transition of an individual pillar would be associated to negative energy release (see Chapter S4, Fig. S4.1d-e).

We study the effect of k_1 on the deformation pathways and the rectification of the bar. We observed two different transitions graph (Fig. 4.6b-c) as k_1 is gradually increased. In the range of k_1 values leading to a countersnapping behavior, only one transition graph is observed (Fig. 4.6c). Importantly, we observe that countersnapping pillars allow to achieve improved rectification with the asymmetric branch '10' compared to the symmetric branches '00' and '11', standing in stark contrast with regular snapping pillars (Fig. 4.6d). The portion of the '10'-branch leading to improved rectification corresponds to the range of pillar compression where the '1'-branch leads to a higher force than the '0'-branch, that is, the regime after the self-intersection ($\nu > 0$, Eq. (4.13)), whereas the other portion of the '10'-branch corresponds to the pre-intersection regime ($\nu < 0$). When the angular stiffness of the bar is reduced, we even observe that *perfect* rectification can be achieved using the same countersnapping pillars (Fig. 4.6e), as the force offset is now large enough to provide the necessary torque to flatten the bar (Eq. (4.9)).

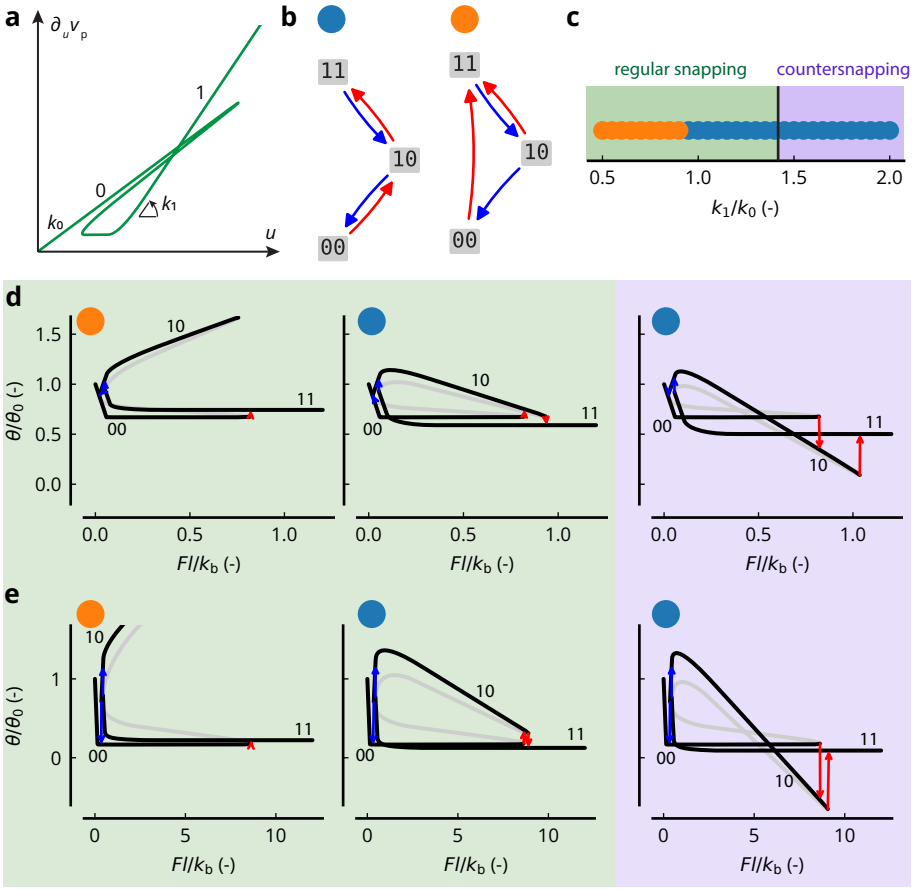


Figure 4.6: Rectification on (counter)snapping pillars. **(a)** Multivalued force-displacement characterizing the behavior of the pillars. It is represented as a zigzag2 bivariate behavior (see Chapter 3, Section S3.4.2). **(b)** Transition graphs obtained when scanning k_1. **(c)** Map of transition graphs in the k_1 space. **(d-e)** Relative angle θ/θ_0 as a function of the dimensionless force Fl/k_b for $k_1/k_0 = \{0.7; 1.4; 2.0\}$, from left to right, when $k_b = 1.0$ (d) and $k_b = 0.1$ (e).

4.3 Rectification on N countersnapping pillars

In the previous section, we demonstrated that countersnapping pillars enabled perfect rectification of the inclined bar. As the inclined bar is a simplified model of the segment of a beam, it is natural to wonder how an initially warped beam, modeled as a chain of multiple segments hinging through angular springs, would behave when pushed against countersnapping pillars. Can improved rectification be achieved with countersnapping pillars compared to linear pillars?

4.3.1 Space of initially curved beams

To address this question, we simulate the rectification process of beams discretized into $N - 1$ segments when pushed against a set of N countersnapping pillars. For each value of N, we generate a family of initially curved beams by assigning to each of the N nodes either an 'up' or 'down' initial vertical position. The initial state of a N-beam can therefore be described by a binary string, where '0' means 'down' and '1' means 'up'.

This binary approach allows to cover a rich and varied space of initial configurations, while keeping the search space relatively small. In principle, the family of N-beams consists of 2^N different configurations. However, some configurations are initially characterized by rigid-body modes, meaning that they would need to be rotated first before forces can actually deform the beam. For the purpose of this exploratory investigation, we will simply discard those initial configurations to simplify the analysis. We will therefore discard configurations characterized by less than two 'down' nodes or when the center of the beam is not in between the left-most and right-most 'down' nodes, as those configurations necessitate a rigid-body rotation before the pillars can be compressed.

Furthermore, we will not simulate initial configurations if the mirrored version has been encountered and simulated already. For example, '0001' and '1000' are equivalent, and there is no need to simulate both cases. Also, the '0000' configuration is not simulated as it is already rectified.

4.3.2 Model

Each N-beam is generated according to the scheme detailed in the previous section. Under each of the N nodes, we place a pillar with a countersnapping behav-

ior (Fig. 4.7a).

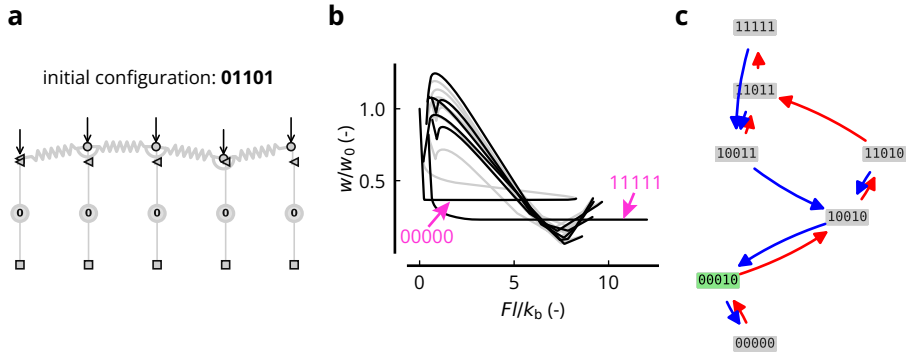


Figure 4.7: Workflow to simulate the rectification process of a curved beam. **(a)** A warped beam is generated from a binary string, and countersnapping pillars are placed under each beam node. **(b)** The relative warping w/w_0 is extracted from the arclength deformation path. **(c)** The transition graph is extracted and the branch realizing the minimum warping is determined (green state).

Each countersnapping behavior is perfectly identical across the entire array of pillars, except when an initial beam configuration is symmetric (meaning that its binary string representation is equal to its reversed version). In those cases, we need to break the symmetry to force the arclength continuation scheme to follow bifurcated branches instead of the possible "unbifurcated" branches corresponding to unstable symmetric states. We break the symmetry by perturbing the countersnapping curves of the pillars in such a way that a pillar has a lower (resp. higher) snapping threshold under both force- and displacement-driven conditions upon loading (resp. unloading) than a pillar on its right.

The countersnapping curve assigned to the pillars is characterized by a post-stiffness k_1 twice as large as the initial stiffness k_0.

4.3.3 Results and analysis

For each rectification process, we can extract the relative warping of the beam (Fig. 4.7b). At any point S during the rectification process, the warping w can be computed using

$$w(S) = \frac{1}{N-2} \sum_{k=1}^{N-2} |\theta(S) - \pi|, \tag{4.14}$$

where θ is the (obtuse) angle formed by two consecutive beam segments. The warping w is the average of the mismatch of each angle compared to the flat angle. It is zero if and only if each angle along the beam is perfectly flat. For each beam, we define the initial warping as $w_0 := w(0)$.

For each simulation, we observed that once the system reaches a state where all the pillars belong to the same linear portion of a branch, the warping value does not evolve with the load, similar to the two-pillar case (Fig. 4.7c). We also observe that the warping value in those uniform states is the same as the warping value we obtain when rectifying the same curved beam against a set of linear pillars with the same stiffness. To compare the countersnapping and linear case, it is therefore interesting to compare the minimum warping value $w_{\text{min}}^{\text{cs}}$ obtained with countersnapping pillars with the minimum warping value $w_{\text{min}}^{\text{lin}}$ obtained with linear pillars characterized by a stiffness equal to the largest (stable) tangent stiffness along the countersnapping curve (that is, k_1).

To compute $w_{\text{min}}^{\text{cs}}$, we first compute the transition graph using the rule established in Section S4.1. From that transition graph, we can then extract the list of all equilibria that are accessible from the initial configuration. Across this list, we compute the minimum warping value using Eq. (4.14). We observe that for every simulation of all N-beams families the rectification performance is improved using countersnapping pillars. The median minimum warping value obtained with countersnapping pillars is at most 30% of the warping achievable using linear pillars (Fig. 4.8a). The number of snapping transitions upon loading to reach the branch leading to the best rectification tends to increase with N (Fig. 4.8b). Interestingly, very few cases necessitate to unload to reach the target branch (Fig. 4.8b).

4.4 Discussion and outlook

First, we have shown that highly nonlinear pillars cause rich deformation pathways in rectification problems, which, in the case of countersnapping pillars, allow to reach states more rectified than what linear pillars with the same stiffness would permit. Matching the flatness achieved by countersnapping pillars would necessitate linear pillars with a stiffness several orders of magnitude larger, or even an infinite stiffness, which is physically impossible. In the context of geometric rectification, countersnapping structures are able to mimic much stiffer or infinitely stiff linear structures. This highlights that strong nonlinearities en-

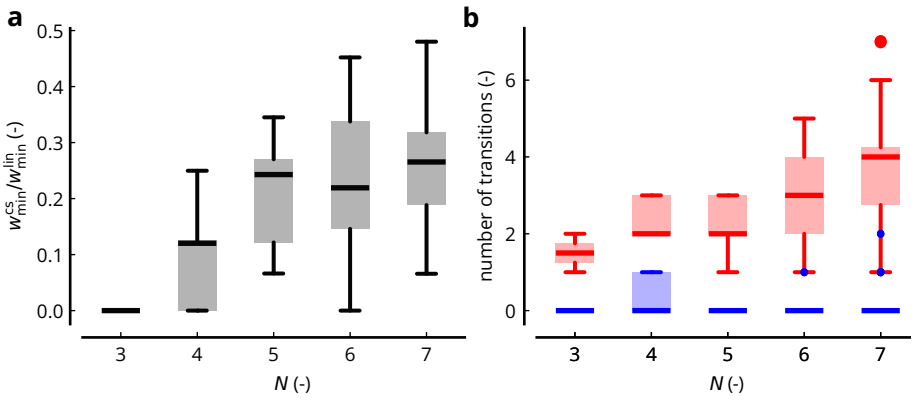


Figure 4.8: Rectification statistics for the N-beams families for increasing N. **(a)** Box plots describing the distribution of the minimum warping values attainable using countersnapping pillars relative to the minimum warping value achievable using linear pillars. **(b)** Box plots describing distribution of the number of transitions upon loading (red) and unloading (blue) to reach the state where the minimum warping value is achieved.

able effective mechanical properties that might seem out of reach or physically impossible at first glance.

Second, unlike rectification on linear pillars, reaching the flattest configuration on countersnapping pillars requires to control the force signal precisely. Moreover, without knowing the exact shape of the warped object, the transition graph is a priori unknown, which brings additional challenges. A possible strategy to find a rectified state in practical scenarios could be to drive the force and detect snapping transitions to evaluate the state of the pillars, while monitoring the flatness of the object. Each time a new state is reached, the force is increased or decreased in order to find another un-visited state, and the process is repeated until the flatness requirement is met. Snapping transitions could be detected by monitoring reaction force spikes at the base of the pillar array, sudden changes in shape or changes in the overall stiffness of the array for example.

Third, in addition to pathways toward a rectified state, we show that many other states associated with different beam shapes can be reached. This suggests that arrays of (counter)snapping pillars have a large and rich configuration space accessible by a simple global and uniform load signal. The ability to reconfigure the state of such arrays could be relevant in other shape-morphing applications

aiming at switching between various surface orientations and curvatures, thereby paving the way toward functional mechanical meta-surfaces.

Finally, beyond the specifics of geometric rectification, we demonstrated that the flexel framework introduced in the previous chapter coupled to the transition rules allows not only to simulate how a nonlinear structure deforms upon loading, but also to build a map of all possible configurations along with the deformation pathways connecting them. Using this approach in the context of other multistable mechanical systems has the potential to facilitate the exploration of the different states that the system can reach. It could also help design deformation pathways toward targeted configurations or avoiding undesirable ones.

S4

Supplementary information for Chapter 4

S4.1 Computing transitions from an arclength deformation path

The arclength path describes the deformation of a structure from an initial state characterized by coordinates $\boldsymbol{Q}_0$, displacements $\boldsymbol{U} = \boldsymbol{0}$, external forces $\boldsymbol{F}^{\text{ext},0}$ and elastic energy E_0 subject to a load step that either drives the displacements or an additional force vector along a subset of coordinates $\tilde{\boldsymbol{Q}}$. The state of the structure changes continuously when one moves along the arclength deformation path. At any point along the path, the structure is characterized by coordinates $\boldsymbol{Q}(S)$, displacements $\boldsymbol{U}(S)$ (with respect to the configuration at the start of the load step), external forces $\boldsymbol{F}^{\text{ext}} = \boldsymbol{F}^{\text{ext},0} + \boldsymbol{F}(S)$ and elastic energy $E(S)$, where S is a scalar that describes the arclength of the path from 0 to the current point.

Under load-driven conditions, the state of the structure runs along the arclength deformation path until it becomes unstable at some *critical* state noted C, i.e. the state where the (reduced) stiffness matrix becomes singular (see Chapter 3). From C, the structure will jump to a different state, the *restabilization* state, noted R. To determine the feasibility of a transition $C \rightarrow R$, we need to be able to evaluate the change in elastic energy (Section S4.1.1), the external work (Section S4.1.2) and the energy release (Section S4.1.3).

S4.1.1 Computing the change in elastic energy

The change in elastic energy at a state S (compared to the initial state) is the work done by the external forces along the continuous arclength path, i.e.

$$\begin{aligned} E(S) &= E_0 + \int_0^S \boldsymbol{F}^{\text{ext}}(\hat{S}) \cdot \frac{\mathrm{d}\boldsymbol{U}(\hat{S})}{\mathrm{d}S}\mathrm{d}\hat{S} \\ &= E_0 + \int_0^S \left(\boldsymbol{F}^{\text{ext},0} + \boldsymbol{F}(\hat{S})\right) \cdot \frac{\mathrm{d}}{\mathrm{d}S}\boldsymbol{U}(\hat{S})\mathrm{d}\hat{S} \\ &= E_0 + \boldsymbol{F}^{\text{ext},0} \cdot \boldsymbol{U}(S) + \int_0^S \tilde{\boldsymbol{F}}(\hat{S}) \cdot \frac{\mathrm{d}}{\mathrm{d}S}\tilde{\boldsymbol{U}}(\hat{S})\mathrm{d}\hat{S}, \end{aligned} \tag{S4.1}$$

where the tilde '~' denotes the subvector with entries corresponding to the coordinates loaded during the load step.

The change in elastic energy $\Delta E_{C\to R}$ between a critical state C and restabilization state R is therefore

$$\begin{aligned} \Delta E_{C\to R} &= E(R) - E(C) \\ &= \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) + \int_C^R \tilde{\boldsymbol{F}}(\hat{S}) \cdot \frac{\mathrm{d}\tilde{\boldsymbol{U}}(\hat{S})}{\mathrm{d}S}\mathrm{d}\hat{S}. \end{aligned} \tag{S4.2}$$

S4.1.2 Computing the work

During a snapping transition, the work done by the external forces is different from the change in elastic energy as the actual deformation path is different from the arclength path. Part of that work is used to change the elastic energy of the system, while the rest is released in the form of kinetic energy.

Under force-driven transition, the work $W^{\text{ext}}_{C\to R}$ done by the external forces during the snapping transition is

$$\begin{aligned} W^{\text{ext}}_{C\to R} &= \int_{\text{force-driven path}} \boldsymbol{F}^{\text{ext}} \cdot \mathrm{d}\boldsymbol{U} \\ &= \int_{\text{force-driven path}} \left(\boldsymbol{F}^{\text{ext},0} + \boldsymbol{F}\right) \cdot \mathrm{d}\boldsymbol{U} \\ &= \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) + \int_{\text{force-driven path}} \tilde{\boldsymbol{F}} \cdot \mathrm{d}\tilde{\boldsymbol{U}} \\ &= \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) + \tilde{\boldsymbol{F}}(C) \cdot \left(\tilde{\boldsymbol{U}}(R) - \tilde{\boldsymbol{U}}(C)\right), \end{aligned} \tag{S4.3}$$

as the additional forces on the loaded coordinates $\tilde{\boldsymbol{F}} = \tilde{\boldsymbol{F}}(C)$ remain constant during the transition.

Under displacement-driven transition, the work $W^{\text{ext}}_{C\to R}$ done by the external forces during the snapping transition is

$$\begin{aligned} W^{\text{ext}}_{C\to R} &= \int_{\text{disp.-driven path}} \boldsymbol{F}^{\text{ext}} \cdot \mathrm{d}\boldsymbol{U} \\ &= \int_{\text{disp.-driven path}} \left(\boldsymbol{F}^{\text{ext},0} + \boldsymbol{F}\right) \cdot \mathrm{d}\boldsymbol{U} \\ &= \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) + \int_{\text{disp.-driven path}} \tilde{\boldsymbol{F}} \cdot \mathrm{d}\tilde{\boldsymbol{U}} \\ &= \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) \,, \end{aligned} \tag{S4.4}$$

since $\tilde{\boldsymbol{U}}$ remains constant during the transition.

S4.1.3 Computing the energy released

The energy released $K^{\text{released}}_{C\to R}$ is then given by

$$\begin{aligned} K^{\text{released}}_{C\to R} &= W^{\text{ext}}_{C\to R} - \Delta E_{C\to R} \\ &= \begin{cases} \tilde{\boldsymbol{F}}(C) \cdot (\tilde{\boldsymbol{U}}(R) - \tilde{\boldsymbol{U}}(C)) - \displaystyle\int_C^R \tilde{\boldsymbol{F}}(\hat{S}) \cdot \frac{\mathrm{d}\tilde{\boldsymbol{U}}(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S} & \text{if force-driven,} \\ \\ -\displaystyle\int_C^R \tilde{\boldsymbol{F}}(\hat{S}) \cdot \frac{\mathrm{d}\tilde{\boldsymbol{U}}(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S} \\ \text{if displacement-driven,} \end{cases} \\ &= \begin{cases} +\displaystyle\int_C^R \tilde{\boldsymbol{U}}(\hat{S}) \cdot \frac{\mathrm{d}\tilde{\boldsymbol{F}}(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S} & \text{if force-driven,} \\ \\ -\displaystyle\int_C^R \tilde{\boldsymbol{F}}(\hat{S}) \cdot \frac{\mathrm{d}\tilde{\boldsymbol{U}}(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S} & \text{if displacement-driven,} \end{cases} \end{aligned} \tag{S4.5}$$

using $\int ab' = ab - \int a'b$ and $\tilde{\boldsymbol{F}}(C) = \tilde{\boldsymbol{F}}(R)$.

A physical snapping transition should necessarily be characterized by a positive energy released $K^{\text{released}}_{C\to R}$ as the change in elastic energy cannot be larger than the amount of energy put in through work. Note that the energy released $K^{\text{released}}_{C\to R}$

does not depend on the preloading force $\tilde{\boldsymbol{F}}^{\text{ext},0}$.

ELASTIC ENERGY, WORK AND ENERGY RELEASED DURING SNAPPING IN PROPORTIONALLY LOADED STRUCTURES In case of proportional loading, the direction $\tilde{\boldsymbol{n}}$ of the load (that is the direction of $\tilde{\boldsymbol{F}}$ if force-driven or $\tilde{\boldsymbol{U}}$ if displacement-driven) remains constant. For convenience, we define $U(S) := \tilde{\boldsymbol{U}}(S) \cdot \tilde{\boldsymbol{n}}$ and $F(S) := \tilde{\boldsymbol{U}}(S) \cdot \tilde{\boldsymbol{n}}$ (regardless of whether $\tilde{\boldsymbol{n}}$ describes the force-driven or displacement-driven loading direction). By exploiting the fact that $\tilde{\boldsymbol{n}}$ remains constant along the path, the energy equations simplify to

$$E(S) = E_0 + \boldsymbol{F}^{\text{ext},0} \cdot \boldsymbol{U}(S) + \int_0^S F(\hat{S}) \frac{\mathrm{d}U(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S}; \tag{S4.6}$$

$$\Delta E_{C\to R} = \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) + \int_C^R F(\hat{S}) \frac{\mathrm{d}U(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S}; \tag{S4.7}$$

$$W^{\text{ext}}_{C\to R} = \begin{cases} \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) + F(C)\,(U(R) - U(C)) \\ \text{if force-driven,} \\ \\ \boldsymbol{F}^{\text{ext},0} \cdot (\boldsymbol{U}(R) - \boldsymbol{U}(C)) \\ \text{if displacement-driven;} \end{cases} \tag{S4.8}$$

$$K^{\text{released}}_{C\to R} = \begin{cases} + \displaystyle\int_C^R U(S) \frac{\mathrm{d}F(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S} & \text{if force-driven,} \\ \\ - \displaystyle\int_C^R F(\hat{S}) \frac{\mathrm{d}U(\hat{S})}{\mathrm{d}S} \mathrm{d}\hat{S} & \text{if displacement-driven.} \end{cases} \tag{S4.9}$$

It is interesting to point out that the energy released can be best visualized by looking at the arclength deformation path projected in the $F - U$ plane, where F and U are the change in external forces $\boldsymbol{F}^{\text{ext}}$ and the displacements $\boldsymbol{U}$ (with respect to the start of the load step) projected onto the constant loading direction $\tilde{\boldsymbol{n}}$. In this specific projection, Eq. (S4.9) says that the kinetic energy released during a force-driven snapping transition is the area enclosed between the curve and the horizontal line $F = F(C)$, whereas for a displacement-driven snapping transition, it is the area enclosed by the curve and the vertical line $U = U(C)$. This is shown in Fig. S4.1 and Fig. S4.2.

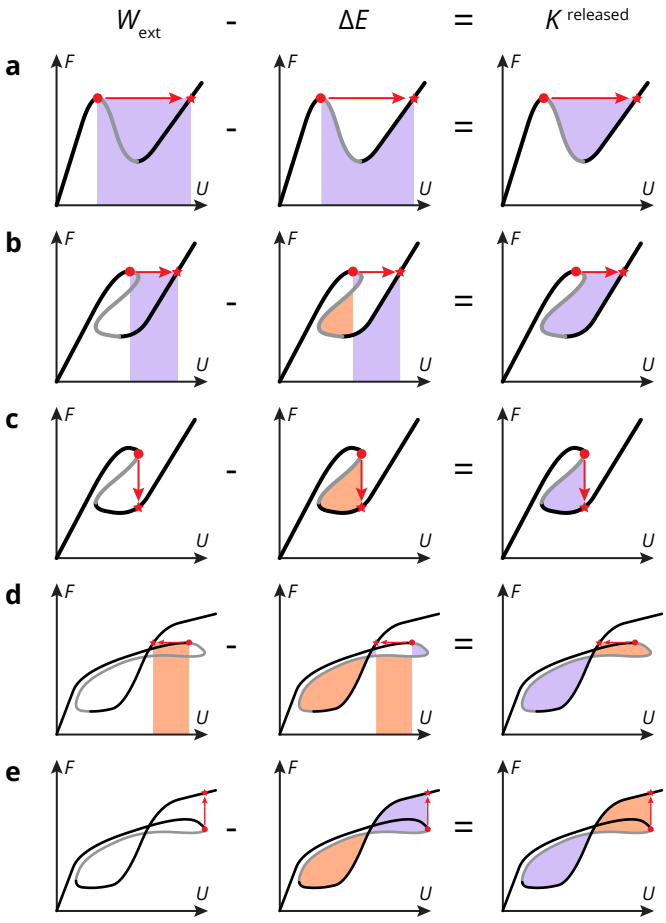


Figure S4.1: Energy balance of snapping transitions upon loading for various force-displacement curves under force and displacement driven conditions. Each energy quantity is represented by a purple (orange) area if positive (negative). Each snapping transition is represented by a red arrow going from the critical point (red dot) to the restabilization point (red star). The first column represents the work done by the external forces W^{ext} (assuming no preloading force) during the snapping transition. The second column represents the change in elastic energy ΔE (assuming no preloading force). The third column represents the amount of energy released K^{released}. **(a)** Nonmonotonic, force-driven. **(b)** Multivalued, force-driven. **(c)** Multivalued, displacement-driven. **(d)** Self-intersecting, force-driven. **(e)** Self-intersecting, displacement-driven.

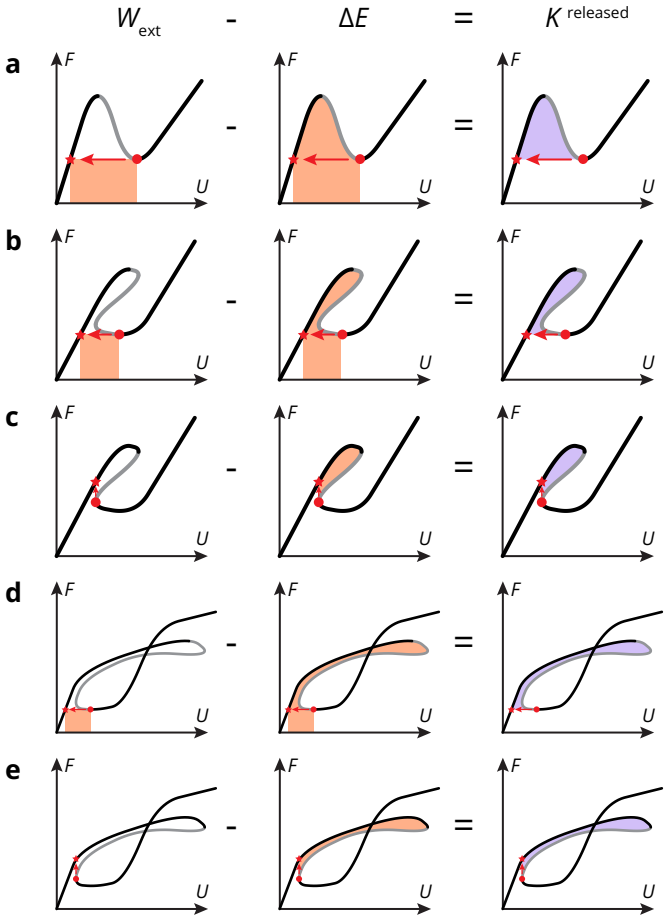


Figure S4.2: Energy balance of snapping transitions upon unloading for various force-displacement curves under force and displacement driven conditions. Each energy quantity is represented by a purple (orange) area if positive (negative). Each snapping transition is represented by a red arrow going from the critical point (red dot) to the restabilization point (red star). The first column represents the work done by the external forces W^{ext} (assuming no preloading force) during the snapping transition. The second column represents the change in elastic energy ΔE (assuming no preloading force). The third column represents the amount of energy released K^{released}. **(a)** Nonmonotonic, force-driven. (b) Multivalued, force-driven. (c) Multivalued, displacement-driven. (d) Self-intersecting, force-driven. (e) Self-intersecting, displacement-driven.

For more complex force-displacement curves, it is harder to visualize the amount of energy released, but it can still be perfectly computed from Eq. (S4.9). For example, for a self-intersecting force-displacement curve, hallmark of countersnapping structures, the energy released can be visualized as a sum of positive and negative areas (Fig. S4.1d-e).

Interestingly, the fact that the energy released must necessarily be positive introduces restrictions on the shape of force-displacement curves. For systems where the load is applied along a single coordinate, such as flexels where the load can only be applied along its geometric measure by definition, the arclength deformation path obtained under force and displacement proportional loading is the same; such a path must therefore satisfy one energy release condition per force-driven and displacement-driven critical point. Those conditions supplemented by the folding rules from Maddocks [100] help assign physical force-displacement curves to flexels.

Finally, it is worth pointing out that a self-intersecting force-displacement curve yields force-driven snapping transitions characterized by *negative work* ($W^{\text{ext}}_{C\to R} < 0$) upon both loading (Fig. S4.1d) and unloading (Fig. S4.2d), unlike the other nonlinear curves. Negative work should be interpreted as the ability of the system to convert some of its elastic energy into positive work *on* the environment during the transition, such as the capability of lifting an increasingly-heavier cup against gravity, or the ability to unidirectionally move an object subject to friction upon cyclic loading (Chapter 2).

Determining the load-driven deformation path Going back to the initial objective of determining the load-driven deformation path given an arclength continuous path, we can now describe the rule more formally. Given an arclength deformation path $(\boldsymbol{Q}(S), \lambda(S))$, where $\lambda(S)$ is the projection of the load along $\tilde{\boldsymbol{n}}$ (that is, $F(S)$ if proportionally loaded in force or $U(S)$ if proportionally loaded in displacement), we can define an ordered list of stable branches, where each stable branch i is a continuous sequence of equilibrium points that are stable under the loading conditions considered. For each branch i, the load is monotonically increasing with S between its lower and upper bounds $C_i^{\pm}$, and we can therefore

define the inverse

$$\lambda = \lambda_i(S) \Leftrightarrow S = \lambda_i^{-1}(\lambda) \quad \forall S \in [C_i^-, C_i^+], \forall \lambda \in [\lambda(C_i^-), \lambda(C_i^+)], \tag{S4.10}$$

and establish the complete rule to construct the load-driven deformation path.

DETERMINING THE LOAD-DRIVEN DEFORMATION PATH FROM THE ARCLENGTH PATH
Given a load signal $\lambda^{\text{driven}}(T)$, the load-driven deformation path $\boldsymbol{Q}^{\text{driven}}(T)$ is constructed as follows:

$$\boldsymbol{Q}^{\text{driven}}(T) = \boldsymbol{Q}(\lambda_i^{-1}(\lambda^{\text{driven}}(T))) \tag{S4.11}$$

subject to the transitions

$$i \xrightarrow{\text{becomes}} \text{next}(i) \qquad \text{when } \lambda^{\text{driven}} \text{ reaches } \lambda_i(C_i^+) \tag{S4.12}$$

$$i \xrightarrow{\text{becomes}} \text{previous}(i) \qquad \text{when } \lambda^{\text{driven}} \text{ reaches } \lambda_i(C_i^-) \tag{S4.13}$$

where

$$\begin{aligned} \text{next}(i) := \min_{j>i}\{j : \exists R \in [C_j^-, C_j^+] \\ \text{such that } \lambda_j(R) = \lambda_i(C_i^+) \text{ and } K^{\text{released}}_{C_i^+ \to R} > 0\} \end{aligned} \tag{S4.14}$$

$$\begin{aligned} \text{previous}(i) := \max_{j<i}\{j : \exists R \in [C_j^-, C_j^+] \\ \text{such that } \lambda_j(R) = \lambda_i(C_i^-) \text{ and } K^{\text{released}}_{C_i^- \to R} > 0\} \end{aligned} \tag{S4.15}$$

5 Conclusion and outlook

The work presented in this thesis is devoted to designing and understanding highly nonlinear mechanical behaviors in elastic structures. We developed an experimental, computational and theoretical framework that leverages the interplay of nonlinear building blocks to build and explore structures with promising properties, derived from their complex force-displacement curve. In this chapter, we review the main contributions of this thesis in the light of the research objectives presented in the introduction and discuss their implications for future research and applications.

5.1 Self-intersecting force-displacement curve

The point of departure of this thesis is the experimental demonstration that self-intersecting force-displacement curves are physically attainable. The profile of the curve translates into mechanical behaviors that are shown to contrast qualitatively with what typical multivalued force-displacement curves enable.

Countersnapping First, the curve swaps the positions of the critical and restabilization points, giving rise to a peculiar subset of snapping instabilities that we call "countersnapping instabilities". Countersnapping reverses the meaning typically associated with snapping. Snapping under increasing tension no longer implies that a system is suddenly pulled apart or that the reaction force drops. Countersnapping not only reinforces the idea that instabilities are not synonyms of failure [4], but demonstrates that they can even be the antonyms, by making

structures suddenly stiffer and able to carry higher loads.

The reversed alignment between the driving force and the displacement also means that countersnapping is able to do work *against* the driving force, as opposed to regular snapping where the driving force provides work to the structure. In general, snapping instabilities are able to do work on the environment by releasing elastic energy. However, they typically do work by generating displacements against *non-driving* forces. For example, the body of the click beetle or a jumping inflatable snaps in response to increased muscular forces or pressure forces, and releases elastic energy that is converted into work by displacing the system against another force, the gravitational force [14, 24]. By contrast, countersnapping provides a mechanism for doing mechanical work along directions that are typically inaccessible.

This property is demonstrated experimentally in Chapter 2. In a chain of countersnapping elements, an initial countersnapping event does work on neighboring elements, and each event in turn does work and triggers the same response in the next element, resulting in a self-propagating wave that travels through the chain as a collective snapping avalanche. Upon cyclic loading, a countersnapping structure exhibits a form of non-reciprocity originating from the change in alignment between the snapping and loading directions across the cycle, which permits unidirectional actuation through friction.

These examples motivate further exploration on how countersnapping or new profiles of force-displacement curves could be harnessed to channel elastic energy release into new forms of actuation and snapping patterns. Could we unlock functional behaviors by providing new ways to write and propagate information [53, 57, 93, 94, 137], generate complex actuation patterns [29, 54], or store energy [47] in mechanical media?

Self-intersection and dynamics Second, the intersection between the two stable branches allows for an increased level of complexity compared to typical multivalued force-displacement curves. Measuring both force and displacement is not sufficient to determine on which branch the state currently lies, as two states are equivalent to an external observer. This form of multivaluedness maps the pair (U, F) to two stiffness values. Snapping instabilities can therefore be harnessed to switch between both branches and achieve programmable natural

frequency and passive resonance avoidance, while maintaining the same external state (U, F) at equilibrium.

Our work only scratches the surface of the potential of nonlinear structures in the dynamic regime. There is little doubt that the dynamic phenomenology of countersnapping structures remains mostly untapped, as more simple statically nonlinear structures can already exhibit complex behaviors, captured by multivalued amplitude-frequency curves [138]. Our work suggests that the self-intersection plays a central role in the dynamic behavior, as it allows for multiple natural frequencies to coexist and influences the amount of kinetic energy released upon snapping. Developing the effective tools to explore the dynamic realm of such structures is a nascent and exciting challenge [139], with promising potential in applications such as vibration isolation [139, 140], MEMS resonators [141] and energy harvesters [142].

5.2 Nonlinearities as building blocks

In Chapter 2, we experimentally demonstrate that complexity can be constructed hierarchically, with weakly nonlinear building blocks assembling into strongly nonlinear structures, which in turn form even more complex nonlinear metastructures. Chapter 3 introduces a computational framework for building high-level descriptions of such mechanical systems.

The framework fits within the common practice across physics and engineering of adopting the right level of abstraction to facilitate modeling, understanding, simulation, and exploration of physical systems, and keep complexity tractable. For example, to study the influence of geometry on the mechanics of a softening building block, a truss of linear springs and angular springs puts the focus on the geometric nonlinearities and abstracts away the material behavior into spring constants. To design a countersnapping structure, focusing on the interplay of the individual nonlinear behaviors is essential (Chapter 2); hence an assembly of springs with nonlinear force-displacement curves that abstract away the deformations internal to the building blocks is more suitable. To study the interaction of multiple countersnapping structures, the local interplay between the softening, stiffening and nonmonotonic building blocks is no longer necessary to simulate. A system comprising elements directly characterized by self-intersecting force-displacement curves provides a higher

and more convenient level of description, as shown in Chapter 4.

The framework we introduce in Chapter 3 unites all these different levels of description into a single element formulation, allowing to treat nonlinearities as building blocks. Key to this unification is a method to construct an elastic energy potential that can be tuned to produce equilibrium points along any given continuous force-displacement curve, even if multivalued. At first glance, it might seem that this method allows to model the behavior of any elastic structure. However, the method has two crucial yet insightful limitations.

Winding limitation First, if we want the energy potential to not only describe a force-displacement behavior via equilibria but also the *stability* of those equilibria, constraints are introduced. Those originate from the link between the stability of an equilibrium and the way the curve folds, which we referred to as the "folding rules", formalized in previous research [100]. Essentially, since stability is governed by the number of turns accumulated by the curve, and a two-dimensional energy potential can only remember a certain number of turns, such a potential is inherently limited at representing stability. New, higher-dimensional (trivariate, quadrivariate, ...) potentials would be necessary in order to describe force-displacement behaviors and their stability based on force-displacement curves that winds more and more, as shown in Fig. 5.1. We leave future researchers go-

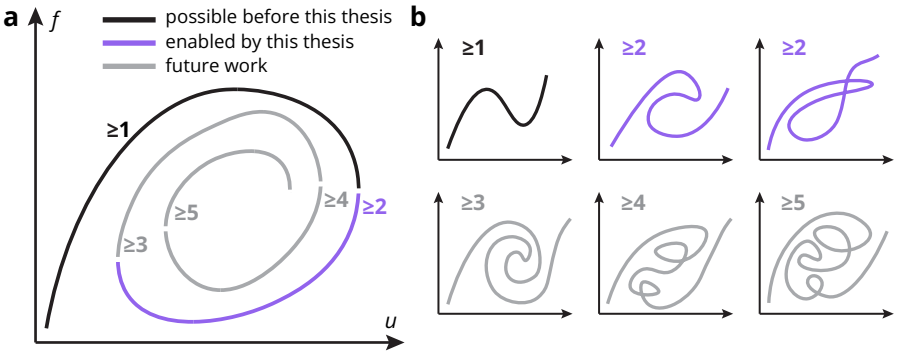


Figure 5.1: Modeling feasibility of winding force-displacement curves. Each number indicates the minimum dimension that an energy potential must have to capture the corresponding behavior and exhibit stability properties consistent with the folding rules. **(a)** Branches of a hypothetical force-displacement curve with an increasing level of winding. **(b)** Examples of force-displacement curves that can (top) and cannot (bottom) be modeled using the framework introduced in this thesis.

ing down that spiral. For now, we will simply point out that the winding of a force-displacement curve seems to place a fundamental restriction on how reduced a physical model can be, if one cares about stability.

Single-measure limitation The second limitation is that flexels can only model structures that have only one relevant measure of deformation for the level of abstraction considered. For example, a countersnapping structure can be abstracted into a single flexel with length as geometric measure. However, if we want to account for the effect of the lateral confinement* on the axial countersnapping behavior, the flexel approach presented in this thesis is inadequate. It lacks the ability to make the intrinsic force-displacement curve *tunable* by some forms of interaction with the rest of the system. The flexel formulation could however be extended in future work by generalizing the formulation of energy potentials and making them dependent on *multiple* measures of deformation. This would open a route toward reduced-order models of metamaterials for example, wherein each unit cell in the lattice is a single flexel capturing horizontal, vertical and shear interactions with its neighbors.

That same extended framework could also be used to model responsive structures, whose force-displacement curves and geometric measures at rest depend on some external field. For example, a piece of liquid crystalline elastomer nonlinearly deforms when exposed to heat, light or electric voltage, as a result of molecular rotations [143]. Wood veneer or elastomeric sheets embedded with pneumatic channels bend in response to humidity or pressure [126, 144]. A magnetic field can tune the peaks and valleys of the nonmonotonic force-displacement curves of magnetic archs [94]. A natural approach would be to represent the external field with an additional degree of freedom T (such as temperature) that influences the energy e of the flexel by tuning the parameters $\boldsymbol{p}$ of its elastic potential v (such as the positions of the curve's control points, geometric measure at rest, etc), i.e.

$$e(\boldsymbol{q}) = v(\alpha(\boldsymbol{z}), t, \boldsymbol{p}(T)). \tag{5.1}$$

Gradients and hessians would then be obtained by taking the derivatives of Eq. (5.1) with respect to $\boldsymbol{q} = [\boldsymbol{z}^\top, t, T]^\top$. That approach could help build reduced-order model of mechanical systems comprising responsive nonlinear

*which would restrict the deformation perpendicular to the axial direction

elements. Solving them statically could reveal a map of deformation pathways that is insightful for general understanding and designing more responsive and adaptable behaviors in nonlinear structures [145].

5.3 Toward broader adoption and applications

This thesis further illustrates that nonlinearities and instabilities offer promising properties for a wide range of potential applications. They enable more adaptable behaviors through passive means by embedding complexity and functionalities within the mechanics, rather than relying on electronic sensors, processors and motors. Yet, broader adoption of nonlinear structures remains challenging; likely due to the relative difficulty of their design and analysis [101], and perhaps also because they face more mature, established, intuitive engineering design traditions [4]. This thesis attempts at addressing those challenges by inspiring engineers and researchers through new tools and demonstrative examples.

Breaking problems into building blocks Chapter 2 demonstrates that using building blocks enables a form of inverse design where a complex target mechanical response is decomposed into less complex entities. Those entities can be designed independently of each other with connectivity compatibility as the sole constraint. This compartmentalization is reminiscent of software architecture, where each module can be implemented freely as long as interfaces are satisfied, enabling the construction of large, complex yet tractable programs. This suggests that inverse design of highly nonlinear structures could be facilitated by methods based on building blocks.

For example, structural optimization could be made more tractable, with fewer design variables, by first searching for a set of weakly nonlinear curves (parametrized by just a few control points, for instance) to assign to blocks forming a stronger nonlinear system, and then, in a second step, separately optimizing the shapes of the blocks to realize those curves. This process can be repeated hierarchically, using flexels as abstractions (Chapter 3), to generate even more complex mechanical responses. This could facilitate topology optimization of structures with targeted multivalued force-displacement curves [146].

Nonlinearities in other physical domains Chapter 4 illustrates that countersnapping instabilities may improve the performance of systems beyond what linear or weakly nonlinear components are capable of. While promising, those highly nonlinear systems may remain too impractical to use in some applications. When subject to real-world loading conditions, their inherent compliance make them more prone to undesired modes of deformation compared to stiff conventional structures. Implementing the desired complexity purely mechanically can also lead to designs that are too bulky for applications where space is limited.

A potential solution is to separate the implementation of the complex behavior from the interface with the environment. This separation is difficult in purely mechanical structures, but could in principle be achieved by fluidic or electromechanical systems. For example, in Chapter 4, the array of pillars with countersnapping behaviors might be difficult to achieve due to the limited space between each pillar. Instead, each pillar could be replaced by a hydraulic piston, where compressive loads from the warped plate are converted into pressure signals and passed to a separate system that implements the countersnapping behavior hydraulically, away from the array. A similar approach could also be implemented electro-mechanically, using piezoelectric components to convert loads into voltage signals to interact at a distance with countersnapping systems. Dome-shaped inflatables or memcapacitors are potential candidates to induce nonmonotonicity in those domains [147, 148].

An invitation to build and explore Mechanics remains however a more accessible and convenient playground to experiment with building blocks, feel forces and instabilities and gain insight into the interplay of nonlinearities. Hopefully, the ideas presented in this thesis appear as an invitation to further explore the possibilities of nonlinear mechanics, through tools such as the catalog of nonlinear building blocks presented in Chapter 2 or the open-access Python library that implements the computational framework developed in Chapter 3 [119]. Some of these concepts have already sparked interest beyond academia [149], and inspired other people to reproduce countersnapping in other systems (Fig. 5.2), 3d printed [150] or even using LEGO [151].

We end this thesis with a quote that Richard Feynman once wrote on his blackboard:

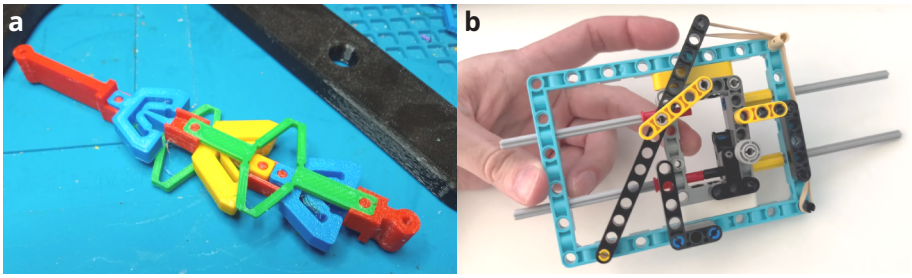


Figure 5.2: Countersnapping structures inspired by the work presented in Chapter 2. **(a)** Same design as presented in this thesis, using blocks 3d-printed in TPU (photo by AxMn, Printables.com, used under CC BY-NC 4.0 [150]). **(b)** Design comprising LEGO blocks and a rubber band (snapshot of Youtube video by Dmytry Lavrov (dizekat), Youtube.com, used with permission [151]).

> "What I cannot create, I do not understand."

We hope that nonlinear building blocks will continue to help us create and understand more of the still fascinating and exciting world of complex mechanics, where things buckle, snap and countersnap.

Bibliography

[1] S. Timoshenko, *History of strength of materials : with a brief account of the history of theory of elasticity and theory of structures*. McGraw-Hill, 1953.

[2] J. L. Lagrange, *Mécanique analytique*. Courcier, 1788.

[3] J. M. T. Thompson, *A general theory of elastic stability*. Wiley, 1973.

[4] P. M. Reis, "A Perspective on the Revival of Structural (In)Stability With Novel Opportunities for Function: From Buckliphobia to Buckliphilia," *Journal of Applied Mechanics*, vol. 82, no. 111001, 2015.

[5] K. Bertoldi, P. M. Reis, S. Willshaw, and T. Mullin, "Negative Poisson's ratio behavior induced by an elastic instability," *Advanced Materials (Deerfield Beach, Fla.)*, vol. 22, no. 3, pp. 361–366, 2010.

[6] J. M. T. Thompson, "Stability of Elastic Structures and Their Loading Devices," *Journal of Mechanical Engineering Science*, vol. 3, no. 2, pp. 153–162, 1961.

[7] ——, "Basic principles in the general theory of elastic stability," *Journal of the Mechanics and Physics of Solids*, vol. 11, no. 1, pp. 13–20, 1963.

[8] M. Gomez, D. E. Moulton, and D. Vella, "Dynamics of viscoelastic snap-through," *Journal of the Mechanics and Physics of Solids*, vol. 124, pp. 781–813, 2019.

[9] J. M. T. Thompson, "The elastic instability of a complete spherical shell," *Aeronautical Quarterly*, vol. 13, pp. 189–201, 1962.

[10] C. Coulais, J. T. Overvelde, L. A. Lubbers, K. Bertoldi, and M. van Hecke, "Discontinuous Buckling of Wide Beams and Metabeams," *Physical Review Letters*, vol. 115, no. 4, p. 044301, 2015.

[11] K. Krieger, "Extreme mechanics: Buckling down," *Nature*, vol. 488, no. 7410, pp. 146–147, 2012.

[12] W. Liu, S. Janbaz, D. Dykstra, B. Ennis, and C. Coulais, "Harnessing plasticity in sequential metamaterials for ideal shock absorption," *Nature*, vol. 634, no. 8035, pp. 842–847, 2024.

[13] G. Ribak and D. Weihs, "Jumping without Using Legs: The Jump of the Click-Beetles (Elateridae) Is Morphologically Constrained," *PLOS ONE*, vol. 6, no. 6, p. e20871, 2011.

[14] Y. Ruan, M. Zhang, R. Kundrata, L. Qiu, S. Ge, X. Yang, X. Chen, S. Jiang, Y. Ruan, M. Zhang, R. Kundrata, L. Qiu, S. Ge, X. Yang, X. Chen, and S. Jiang, "Functional Morphology of the Thorax of the Click Beetle Campsosternus auratus (Coleoptera, Elateridae), with an Emphasis on Its Jumping Mechanism," *Insects*, vol. 13, no. 3, 2022.

[15] M. S. deVries, E. A. K. Murphy, and S. N. Patek, "Strike mechanics of an ambush predator: the spearing mantis shrimp," *Journal of Experimental Biology*, vol. 215, no. 24, pp. 4374–4384, 2012.

[16] S. N. Patek, "The Power of Mantis Shrimp Strikes: Interdisciplinary Impacts of an Extreme Cascade of Energy Release," *Integrative and Comparative Biology*, vol. 59, no. 6, pp. 1573–1585, 2019.

[17] E. Steinhardt, N.-s. P. Hyun, J.-s. Koh, G. Freeburn, M. H. Rosen, F. Z. Temel, S. N. Patek, and R. J. Wood, "A physical model of mantis shrimp for exploring the dynamics of ultrafast systems," *Proceedings of the National Academy of Sciences*, vol. 118, no. 33, p. e2026833118, 2021.

[18] M. L. Smith, G. M. Yanega, and A. Ruina, "Elastic instability model of rapid beak closure in hummingbirds," *Journal of Theoretical Biology*, vol. 282, no. 1, pp. 41–51, 2011.

[19] T. G. Thuruthel, S. Haider Abidi, M. Cianchetti, C. Laschi, and E. Falotico, "A bistable soft gripper with mechanically embedded sensing and actuation for fast grasping," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 1049–1054.

[20] Y. Wang, U. Gupta, N. Parulekar, and J. Zhu, "A soft gripper of fast speed and low energy consumption," *Science China Technological Sciences*, vol. 62, no. 1, pp. 31–38, 2019.

[21] R. Baumgartner, A. Kogler, J. M. Stadlbauer, C. C. Foo, R. Kaltseis, M. Baumgartner, G. Mao, C. Keplinger, S. J. A. Koh, N. Arnold, Z. Suo, M. Kaltenbrunner, and S. Bauer, "A Lesson from Plants: High-Speed Soft Robotic Actuators," *Advanced Science*, vol. 7, no. 5, p. 1903391, 2020.

[22] Y. Kim, J. van den Berg, and A. J. Crosby, "Autonomous snapping and jumping polymer gels," *Nature Materials*, vol. 20, no. 12, pp. 1695–1701, 2021.

[23] H. Lee, C. Xia, and N. X. Fang, "First jump of microgel; actuation speed enhancement by elastic instability," *Soft Matter*, vol. 6, no. 18, pp. 4342–4345, 2010.

[24] B. Gorissen, D. Melancon, N. Vasios, M. Torbati, and K. Bertoldi, "Inflatable soft jumper inspired by shell snapping," *Science Robotics*, vol. 5, no. 42, p. eabb1967, 2020.

[25] Y. Yang and Y. Wang, "Snapping for 4D-Printed Insect-Scale Metal-Jumper," *Advanced Science*, vol. 11, no. 3, p. 2307088, 2024.

[26] T. Chen, O. R. Bilal, K. Shea, and C. Daraio, "Harnessing bistability for directional propulsion of soft, untethered robots," *Proceedings of the National Academy of Sciences*, vol. 115, no. 22, pp. 5698–5702, 2018.

[27] Y. Chi, Y. Hong, Y. Zhao, Y. Li, and J. Yin, "Snapping for high-speed and high-efficient butterfly stroke–like soft swimmer," *Science Advances*, vol. 8, no. 46, p. eadd3788, 2022.

[28] Y. Tang, Y. Chi, J. Sun, T.-H. Huang, O. H. Maghsoudi, A. Spence, J. Zhao, H. Su, and J. Yin, "Leveraging elastic instabilities for amplified performance: Spine-inspired high-speed and high-force soft robots," *Science Advances*, vol. 6, no. 19, p. eaaz6912, 2020.

[29] J. T. B. Overvelde, T. Kloek, J. J. A. D'haen, and K. Bertoldi, "Amplifying the response of soft actuators by harnessing snap-through instabilities," *Proceedings of the National Academy of Sciences*, vol. 112, no. 35, pp. 10 863–10 868, 2015.

[30] J. S. Go, Y.-H. Cho, B. M. Kwak, and K. Park, "Snapping microswitches with adjustable acceleration threshold," *Sensors and Actuators A: Physical*, vol. 54, no. 1, pp. 579–583, 1996.

[31] N. Masters and L. Howell, "A self-retracting fully compliant bistable micromechanism," *Journal of Microelectromechanical Systems*, vol. 12, no. 3, pp. 273–280, 2003.

[32] M. A. ten Wolde and D. Farhadi, "A single-input state-switching building block harnessing internal instabilities," *Mechanism and Machine Theory*, vol. 196, p. 105626, 2024.

[33] T. Schioler and S. Pellegrino, "Space Frames with Multiple Stable Configurations," *AIAA Journal*, vol. 45, no. 7, pp. 1740–1747, 2007.

[34] L. C. van Laake, J. de Vries, S. Malek Kani, and J. T. B. Overvelde, "A fluidic relaxation oscillator for reprogrammable sequential actuation in soft robots," *Matter*, vol. 5, no. 9, pp. 2898–2917, 2022.

[35] A. Comoretto, H. A. H. Schomaker, and J. T. B. Overvelde, "Physical synchronization of soft self-oscillating limbs for fast and autonomous locomotion," *Science*, vol. 388, no. 6747, pp. 610–615, 2025.

[36] J.-H. Lee, H. W. Ro, R. Huang, P. Lemaillet, T. A. Germer, C. L. Soles, and C. M. Stafford, "Anisotropic, Hierarchical Surface Patterns via Surface Wrinkling of Nanopatterned Polymer Films," *Nano Letters*, vol. 12, no. 11, pp. 5995–5999, 2012.

[37] S. Xu, Z. Yan, K.-I. Jang, W. Huang, H. Fu, J. Kim, Z. Wei, M. Flavin, J. McCracken, R. Wang, A. Badea, Y. Liu, D. Xiao, G. Zhou, J. Lee, H. U. Chung, H. Cheng, W. Ren, A. Banks, X. Li, U. Paik, R. G. Nuzzo, Y. Huang, Y. Zhang, and J. A. Rogers, "Assembly of micro/nanomaterials into complex, three-dimensional architectures by compressive buckling," *Science*, vol. 347, no. 6218, pp. 154–159, 2015.

[38] J. T. B. Overvelde, S. Shan, and K. Bertoldi, "Compaction Through Buckling in 2D Periodic, Soft and Porous Structures: Effect of Pore Shape," *Advanced Materials*, vol. 24, no. 17, pp. 2337–2342, 2012.

[39] J. Shim, C. Perdigou, E. R. Chen, K. Bertoldi, and P. M. Reis, "Buckling-induced encapsulation of structured elastic shells under pressure," *Proceedings of the National Academy of Sciences*, vol. 109, no. 16, pp. 5978–5983, 2012.

[40] S. M. Carroll and A. Parola, "What Emergence Can Possibly Mean," 2024.

[41] C. Adami, "What is complexity?" *BioEssays*, vol. 24, no. 12, pp. 1085–1094, 2002.

[42] C. Lu, M. Hsieh, Z. Huang, C. Zhang, Y. Lin, Q. Shen, F. Chen, and L. Zhang, "Architectural Design and Additive Manufacturing of Mechanical Metamaterials: A Review," *Engineering*, vol. 17, pp. 44–63, 2022.

[43] R. Lakes, "Foam Structures with a Negative Poisson's Ratio," *Science*, vol. 235, no. 4792, pp. 1038–1040, 1987.

[44] J. B. Berger, H. N. G. Wadley, and R. M. McMeeking, "Mechanical metamaterials at the theoretical limit of isotropic elastic stiffness," *Nature*, vol. 543, no. 7646, pp. 533–537, 2017.

[45] M. Kadic, T. Bückmann, R. Schittny, and M. Wegener, "On anisotropic versions of three-dimensional pentamode metamaterials," *New Journal of Physics*, vol. 15, no. 2, p. 023029, 2013.

[46] K. Bertoldi, V. Vitelli, J. Christensen, and M. van Hecke, "Flexible mechanical metamaterials," *Nature Reviews Materials*, vol. 2, no. 11, p. 17066, 2017.

[47] S. Shan, S. H. Kang, J. R. Raney, P. Wang, L. Fang, F. Candido, J. A. Lewis, and K. Bertoldi, "Multistable Architected Materials for Trapping Elastic Strain Energy," *Advanced Materials*, vol. 27, no. 29, pp. 4296–4301, 2015.

[48] A. Iniguez-Rabago, Y. Li, and J. T. B. Overvelde, "Exploring multistability in prismatic metamaterials through local actuation," *Nature Communications*, vol. 10, no. 1, p. 5577, 2019.

[49] E. Bernardes and S. Viollet, "Design of an Origami Bendy Straw for Robotic Multistable Structures," *Journal of Mechanical Design*, vol. 144, no. 033301, 2021.

[50] A. Rafsanjani, A. Akbarzadeh, and D. Pasini, "Snapping Mechanical Metamaterials under Tension," *Advanced Materials*, vol. 27, no. 39, pp. 5931–5935, 2015.

[51] D. Restrepo, N. D. Mankame, and P. D. Zavattieri, "Phase transforming cellular materials," *Extreme Mechanics Letters*, vol. 4, pp. 52–60, 2015.

[52] J. Liu, M. Teunisse, G. Korovin, I. R. Vermaire, L. Jin, H. Bense, and M. Van Hecke, "Controlled pathways and sequential information processing in serially coupled mechanical hysterons," *Proceedings of the National Academy of Sciences*, vol. 121, no. 22, p. e2308414121, 2024.

[53] J. R. Raney, N. Nadkarni, C. Daraio, D. M. Kochmann, J. A. Lewis, and K. Bertoldi, "Stable propagation of mechanical signals in soft media using stored elastic energy," *Proceedings of the National Academy of Sciences*, vol. 113, no. 35, pp. 9722–9727, 2016.

[54] B. Gorissen, E. Milana, A. Baeyens, E. Broeders, J. Christiaens, K. Collin, D. Reynaerts, and M. De Volder, "Hardware Sequencing of Inflatable Nonlinear Actuators for Autonomous Soft Robots," *Advanced Materials*, vol. 31, no. 3, p. 1804598, 2019.

[55] D. Melancon, A. E. Forte, L. M. Kamp, B. Gorissen, and K. Bertoldi, "Inflatable Origami: Multimodal Deformation via Multistability," *Advanced Functional Materials*, vol. 32, no. 35, p. 2201891, 2022.

[56] B. Van Raemdonck, E. Milana, M. De Volder, D. Reynaerts, and B. Gorissen, "Nonlinear Inflatable Actuators for Distributed Control in Soft Robots," *Advanced Materials*, vol. 35, no. 35, p. 2301487, 2023.

[57] M. van Hecke, "Profusion of transition pathways for interacting hysterons," *Physical Review E*, vol. 104, no. 5, p. 054608, 2021.

[58] M. Teunisse and M. van Hecke, "Transition graphs of interacting hysterons: Structure, design, organization and statistics," *Royal Society Open Science*, vol. 12, no. 9, p. 250753, 2025.

[59] L. Jin, R. Khajehtourian, J. Mueller, A. Rafsanjani, V. Tournat, K. Bertoldi, and D. M. Kochmann, "Guided transition waves in multistable mechanical metamaterials," *Proceedings of the National Academy of Sciences*, vol. 117, no. 5, pp. 2319–2325, 2020.

[60] C. M. Penchina and L. J. Penchina, "The Braess paradox in mechanical, traffic, and other networks," *American Journal of Physics*, vol. 71, no. 5, pp. 479–482, 2003.

[61] Z. G. Nicolaou and A. E. Motter, "Mechanical metamaterials with negative compressibility transitions," *Nature Materials*, vol. 11, no. 7, pp. 608–613, 2012.

[62] P. Ducarme, B. Weber, M. van Hecke, and J. T. B. Overvelde, "Exotic mechanical properties enabled by countersnapping instabilities," *Proceedings of the National Academy of Sciences*, vol. 122, no. 16, p. e2423301122, 2025.

[63] H. Yang and L. Ma, "Multi-stable mechanical metamaterials by elastic buckling instability," *Journal of Materials Science*, vol. 54, no. 4, pp. 3509–3526, 2019.

[64] L. Wu and D. Pasini, "In Situ Activation of Snap-Through Instability in Multi-Response Metamaterials through Multistable Topological Transformation," *Advanced Materials*, vol. 35, no. 36, p. 2301109, 2023.

[65] N. Nadkarni, A. F. Arrieta, C. Chong, D. M. Kochmann, and C. Daraio, "Unidirectional Transition Waves in Bistable Lattices," *Physical Review Letters*, vol. 116, no. 24, p. 244501, 2016.

[66] B. Haghpanah, L. Salari-Sharif, P. Pourrajab, J. Hopkins, and L. Valdevit, "Multistable Shape-Reconfigurable Architected Materials," *Advanced Materials*, vol. 28, no. 36, pp. 7915–7920, 2016.

[67] Y. Liu, F. Pan, B. Ding, Y. Zhu, K. Yang, and Y. Chen, "Multistable shape-reconfigurable metawire in 3D space," *Extreme Mechanics Letters*, vol. 50, p. 101535, 2021.

[68] A. S. Meeussen and M. van Hecke, "Multistable sheets with rewritable patterns for switchable shape-morphing," *Nature*, vol. 621, no. 7979, pp. 516–520, 2023.

[69] C. M. Stafford, C. Harrison, K. L. Beers, A. Karim, E. J. Amis, M. R. VanLandingham, H.-C. Kim, W. Volksen, R. D. Miller, and E. E. Simonyi, "A buckling-based metrology for measuring the elastic moduli of polymeric thin films," *Nature Materials*, vol. 3, no. 8, pp. 545–550, 2004.

[70] T. Mei and C. Q. Chen, "In-memory mechanical computing," *Nature Communications*, vol. 14, no. 1, p. 5204, 2023.

[71] L. P. Hyatt and R. L. Harne, "Programming metastable transition sequences in digital mechanical materials," *Extreme Mechanics Letters*, vol. 59, p. 101975, 2023.

[72] D. Yang, B. Mosadegh, A. Ainla, B. C. G. Lee, F. Khashai, Z. Suo, K. Bertoldi, and G. M. Whitesides, "Buckling of Elastomeric Beams Enables Actuation of Soft Machines," *Adv. Mater.*, 2015.

[73] A. Nagarkar, W.-K. Lee, D. J. Preston, M. P. Nemitz, N.-N. Deng, G. M. Whitesides, and L. Mahadevan, "Elastic-instability–enabled locomotion," *Proceedings of the National Academy of Sciences*, vol. 118, no. 8, p. e2013801118, 2021.

[74] X. Cai and B. Tang, "Mechanically controlled robotic gripper with bistability for fast and adaptive grasping," *Bioinspiration and Biomimetics*, vol. 18, p. 014001, 2023.

[75] L. C. Van Laake, A. Comoretto, and J. T. Overvelde, "On the coexistence of pressure regulation and oscillation modes in soft hysteretic valves," *Journal of Fluids and Structures*, vol. 126, p. 104090, 2024.

[76] D. P. Holmes and A. J. Crosby, "Snapping Surfaces," *Advanced Materials*, vol. 19, no. 21, pp. 3589–3593, 2007.

[77] G. Puglisi and L. Truskinovsky, "Mechanics of a discrete chain with bi-stable elements," *Journal of the Mechanics and Physics of Solids*, vol. 48, no. 1, pp. 1–27, 2000.

[78] E. Ben-Haim, L. Salem, Y. Or, and A. D. Gat, "Single-Input Control of Multiple Fluid-Driven Elastic Actuators via Interaction Between Bistability and Viscosity," *Soft Robotics*, vol. 7, no. 2, pp. 259–265, 2020.

[79] H. Mofatteh, B. Shahryari, A. Mirabolghasemi, A. Seyedkanani, R. Shirzadkhani, G. Desharnais, and A. Akbarzadeh, "Programming Multistable Metamaterials to Discover Latent Functionalities," *Advanced Science*, vol. 9, no. 33, p. 2202883, 2022.

[80] J. Ding and M. van Hecke, "Sequential snapping and pathways in a mechanical metamaterial," *The Journal of Chemical Physics*, vol. 156, no. 20, p. 204902, 2022.

[81] L. S. Novelino, Q. Ze, S. Wu, G. H. Paulino, and R. Zhao, "Untethered control of functional origami microrobots with distributed actuation," *Proceedings of the National Academy of Sciences*, vol. 117, no. 39, pp. 24 096–24 101, 2020.

[82] J. Zha and Z. Zhang, "Reversible negative compressibility metamaterials inspired by braess's paradox," *Smart Materials and Structures*, vol. 33, no. 7, p. 075036, 2024.

[83] D. Braess, "Über ein Paradoxon aus der Verkehrsplanung," *Unternehmensforschung*, vol. 12, no. 1, pp. 258–268, 1968.

[84] J. E. Cohen and P. Horowitz, "Paradoxical behaviour of mechanical and electrical networks," *Nature*, vol. 352, no. 6337, pp. 699–701, 1991.

[85] T. E. Bruns, O. Sigmund, and D. A. Tortorelli, "Numerical methods for the topology optimization of structures that exhibit snap-through," *International Journal for Numerical Methods in Engineering*, vol. 55, no. 10, pp. 1215–1237, 2002.

[86] M. Wallin, N. Ivarsson, and D. Tortorelli, "Stiffness optimization of non-linear elastic structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 330, pp. 292–307, 2018.

[87] D. W. Pohl, "Dynamic piezoelectric translation devices," *Review of Scientific Instruments*, vol. 58, no. 1, pp. 54–57, 1987.

[88] S. Mohith, A. R. Upadhya, K. P. Navin, S. M. Kulkarni, and M. Rao, "Recent trends in piezoelectric actuators for precision motion and their applications: a review," *Smart Materials and Structures*, vol. 30, no. 1, p. 013002, 2020.

[89] L. J. Kwakernaak and M. van Hecke, "Counting and sequential information processing in mechanical metamaterials," *Phys. Rev. Lett.*, vol. 130, p. 268204, 2023.

[90] J. Casals-Terre, A. Fargas-Marques, and A. M. Shkel, "Snap-Action Bistable Micromechanisms Actuated by Nonlinear Resonance," *Journal of Microelectromechanical Systems*, vol. 17, no. 5, pp. 1082–1093, 2008.

[91] M. Mungan and M. M. Terzi, "The structure of state transition graphs in systems with return point memory: I. general theory," *Annales Henri Poincaré*, vol. 20, no. 8, pp. 2819–2872, 2019.

[92] M. M. Terzi and M. Mungan, "State transition graph of the Preisach model and the role of return-point memory," *Physical Review E*, vol. 102, no. 1, p. 012122, 2020.

[93] D. Shohat and M. van Hecke, "Geometric control and memory in networks of bistable elements," 2024.

[94] A. Pal and M. Sitti, "Programmable mechanical devices through magnetically tunable bistable elements," *Proceedings of the National Academy of Sciences*, vol. 120, no. 15, p. e2212489120, 2023.

[95] L. M. Korpas, R. Yin, H. Yasuda, and J. R. Raney, "Temperature-Responsive Multistable Metamaterials," *ACS Applied Materials & Interfaces*, vol. 13, no. 26, pp. 31 163–31 170, 2021.

[96] M. R. Shankar, M. L. Smith, V. P. Tondiglia, K. M. Lee, M. E. McConney, D. H. Wang, L.-S. Tan, and T. J. White, "Contactless, photoinitiated snap-through in azobenzene-functionalized polymers," *Proceedings of the National Academy of Sciences*, vol. 110, no. 47, pp. 18 792–18 797, 2013.

[97] A. Abbasi, T. Chen, B. F. Aymon, and P. M. Reis, "Leveraging the Snap Buckling of Bistable Magnetic Shells to Design a Refreshable Braille Dot," *Advanced Materials Technologies*, vol. 9, no. 3, p. 2301344, 2024.

[98] M. A. Crisfield, "A fast incremental/iterative solution procedure that handles "snap-through"," *Computers & Structures*, vol. 13, no. 1, pp. 55–62, 1981.

[99] M. Ritto-Corrêa and D. Camotim, "On the arc-length and other quadratic control methods: Established, less known and new implementation procedures," *Computers & Structures*, vol. 86, no. 11-12, pp. 1353–1368, 2008.

[100] J. H. Maddocks, "Stability and folds," *Archive for Rational Mechanics and Analysis*, vol. 99, no. 4, pp. 301–328, 1987.

[101] L. L. Howell, *Compliant Mechanisms*. Wiley, 2001.

[102] Z. Liu, H. Fang, J. Xu, and K.-W. Wang, "Cellular Automata Inspired Multistable Origami Metamaterials for Mechanical Learning," *Advanced Science*, vol. 10, no. 34, p. 2305146, 2023.

[103] S. Yan, W. Liu, X. Tan, Z. Meng, W. Luo, H. Jin, Y. Wen, J. Sun, L. Wu, and J. Zhou, "Bio-inspired mechanical metamaterial with ultrahigh load-bearing capacity for energy dissipation," *Materials Today*, vol. 77, pp. 11–18, 2024.

[104] R. Zhang, T. Auzinger, and B. Bickel, "Computational Design of Planar Multistable Compliant Structures," *ACM Trans. Graph.*, vol. 40, no. 5, pp. 186:1–186:16, 2021.

[105] B. Caasenbrood, A. Pogromsky, and H. Nijmeijer, "Energy-Shaping Controllers for Soft Robot Manipulators Through Port-Hamiltonian Cosserat Models," *SN Computer Science*, vol. 3, no. 6, p. 494, 2022.

[106] X. Zhang, F. K. Chan, T. Parthasarathy, and M. Gazzola, "Modeling and simulation of complex dynamic musculoskeletal architectures," *Nature Communications*, vol. 10, no. 1, p. 4825, 2019.

[107] D. Dykstra, J. Busink, B. Ennis, and C. Coulais, "Viscoelastic Snapping Metamaterials," *Journal of Applied Mechanics*, vol. 86, p. 1, 2019.

[108] Z. Meng, W. Chen, T. Mei, Y. Lai, Y. Li, and C. Q. Chen, "Bistability-based foldable origami mechanical logic gates," *Extreme Mechanics Letters*, vol. 43, p. 101180, 2021.

[109] A. Bekele, M. A. Wadee, and A. T. M. Phillips, "Enhancing energy absorption through sequential instabilities in mechanical metamaterials," *Royal Society Open Science*, vol. 10, no. 8, p. 230762, 2023.

[110] D. Shohat and M. van Hecke, "Geometric Control and Memory in Networks of Hysteretic Elements," *Physical Review Letters*, vol. 134, no. 18, p. 188201, 2025.

[111] D. Shohat, P. Baconnier, I. Procaccia, M. v. Hecke, and Y. Lahini, "Aging of amorphous materials under cyclic strain," 2025.

[112] E. Riks, "An incremental approach to the solution of snapping and buckling problems," *International Journal of Solids and Structures*, vol. 15, no. 7, pp. 529–551, 1979.

[113] P. E. Farrell, C. H. L. Beentjes, and Ásgeir Birkisson, "The computation of disconnected bifurcation diagrams," 2016.

[114] P. Obara, J. Kłosowska, and W. Gilewski, "Truth and Myths about 2D Tensegrity Trusses," *Applied Sciences*, vol. 9, no. 1, p. 179, 2019.

[115] G. He, C. Sparks, and N. Gravish, "Grasping and rolling in-plane manipulation using deployable tape spring appendages," *Science Advances*, vol. 11, no. 15, p. eadt5905, 2025.

[116] L. J. Kwakernaak, A. Guerra, D. P. Holmes, and M. van Hecke, "The collective snapping of a pair of bumping buckled beams," *Extreme Mechanics Letters*, vol. 69, p. 102160, 2024.

[117] A. Guerra, A. C. Slim, D. P. Holmes, and O. Kodio, "Self-ordering of buckling, bending, and bumping beams," *Phys. Rev. Lett.*, vol. 130, p. 148201, 2023.

[118] A. Djellouli, B. Van Raemdonck, Y. Wang, Y. Yang, A. Caillaud, D. Weitz, S. Rubinstein, B. Gorissen, and K. Bertoldi, "Shell buckling for programmable metafluids," *Nature*, vol. 628, no. 8008, pp. 545–550, 2024.

[119] P. Ducarme, "springable," available on PyPI: https://pypi.org/project/springable/1.0.0/, source code: https://github.com/ducarme/springable, 2025.

[120] J. Byun, A. Pal, J. Ko, and M. Sitti, "Integrated mechanical computing for autonomous soft machines," *Nature Communications*, vol. 15, no. 1, 2024.

[121] E. Fermi, P. Pasta, S. Ulam, and M. Tsingou, "Studies of the nonlinear problems," Los Alamos National Laboratory, Tech. Rep., 1955.

[122] 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*, vol. 64, pp. 426–439, 2014.

[123] A. Vainchtein and L. Truskinovsky, "Solitary Waves and Kinks in FPU Lattices with Soft–Hard–Soft Trilinear Interactions," *Journal of Nonlinear Science*, vol. 34, no. 6, p. 113, 2024.

[124] K. K. Dudek, M. Kadic, C. Coulais, and K. Bertoldi, "Shape Morphing Metamaterials," 2025.

[125] M. J. Mirzaali, A. Ghorbani, K. Nakatani, M. Nouri-Goushki, N. Tümer, S. J. P. Callens, S. Janbaz, A. Accardo, J. Bico, M. Habibi, and A. A. Zadpoor, "Curvature Induced by Deflection in Thick Meta-Plates," *Advanced Materials*, vol. 33, no. 30, p. 2008082, 2021.

[126] E. Siéfert, E. Reyssat, J. Bico, and B. Roman, "Bio-inspired pneumatic shape-morphing elastomers," *Nature Materials*, vol. 18, no. 1, pp. 24–28, 2019.

[127] C. Coulais, E. Teomy, K. de Reus, Y. Shokef, and M. van Hecke, "Combinatorial design of textured mechanical metamaterials," *Nature*, vol. 535, no. 7613, pp. 529–532, 2016.

[128] M. Czajkowski, C. Coulais, M. van Hecke, and D. Z. Rocklin, "Conformal elasticity of mechanism-based metamaterials," *Nature Communications*, vol. 13, no. 1, p. 211, 2022.

[129] K. Mahadevan, M. C. Yuen, V. Sanchez, R. J. Wood, and K. Bertoldi, "Knitting Multistability," 2024.

[130] J. A. Faber, J. P. Udani, K. S. Riley, A. R. Studart, and A. F. Arrieta, "Dome-Patterned Metamaterial Sheets," *Advanced Science*, vol. 7, no. 22, p. 2001955, 2020.

[131] L. Wu, S. Chen, A. E. Acha, and D. Pasini, "Remorphable Architectures: Reprogramming Global Bistability through Locally Bistable Metamaterials," *Advanced Materials*, vol. n/a, no. n/a, p. e09940, 2025.

[132] D. Melancon, B. Gorissen, C. J. García-Mora, C. Hoberman, and K. Bertoldi, "Multistable inflatable origami structures at the metre scale," *Nature*, vol. 592, no. 7855, pp. 545–550, 2021.

[133] A. Zareei, B. Deng, and K. Bertoldi, "Harnessing transition waves to realize deployable structures," *Proceedings of the National Academy of Sciences*, vol. 117, no. 8, pp. 4015–4020, 2020.

[134] T. Jules, A. Reid, K. E. Daniels, M. Mungan, and F. Lechenault, "Delicate memory structure of origami switches," *Physical Review Research*, vol. 4, no. 1, p. 013128, 2022.

[135] G. Muhaxheri and C. D. Santangelo, "Bifurcations of inflating balloons and interacting hysterons," *Physical Review E*, vol. 110, no. 2, p. 024209, 2024.

[136] P. Baconnier, M. H. Teunisse, and M. v. Hecke, "Dynamic self-loops in networks of passive and active binary elements," 2025.

[137] H. Yasuda, P. R. Buskohl, A. Gillman, T. D. Murphey, S. Stepney, R. A. Vaia, and J. R. Raney, "Mechanical computing," *Nature*, vol. 598, no. 7879, pp. 39–48, 2021.

[138] N. Mallon, R. Fey, and H. Nijmeijer, "Dynamic Stability of a Base-Excited Thin Beam with Top Mass," *Brain Research*, 2006.

[139] S. R. Patro, H. Sharma, S. Yang, and J. Yang, "Passive Vibration Isolation Characteristics of Negative Extensibility Metamaterials," 2025, arXiv:2507.00396 [nlin]. [Online]. Available: http://arxiv.org/abs/2507.00396

[140] A. Carrella, M. J. Brennan, and T. P. Waters, "Optimization of a quasi-zero-stiffness isolator," *Journal of Mechanical Science and Technology*, vol. 21, no. 6, pp. 946–949, 2007.

[141] R. M. C. Mestrom, R. H. B. Fey, J. T. M. van Beek, K. L. Phan, and H. Nijmeijer, "Modelling the dynamics of a MEMS resonator: Simulations and experiments," *Sensors and Actuators A: Physical*, vol. 142, no. 1, pp. 306–315, 2008.

[142] R. L. Harne and K. W. Wang, "Prospects for Nonlinear Energy Harvesting Systems Designed Near the Elastic Stability Limit When Driven by Colored Noise," *Journal of Vibration and Acoustics*, vol. 136, no. 021009, 2013.

[143] K. M. Herbert, H. E. Fowler, J. M. McCracken, K. R. Schlafmann, J. A. Koch, and T. J. White, "Synthesis and alignment of liquid crystalline elastomers," *Nature Reviews Materials*, vol. 7, no. 1, pp. 23–38, 2022.

[144] D. Luo, A. Maheshwari, A. Danielescu, J. Li, Y. Yang, Y. Tao, L. Sun, D. K. Patel, G. Wang, S. Yang, T. Zhang, and L. Yao, "Autonomous self-burying seed carriers for aerial seeding," *Nature*, vol. 614, no. 7948, pp. 463–470, 2023.

[145] E. Kurt, Y. M. Golestani, E. Barmpoutsis, S. Picella, R. L. B. Selinger, J. T. B. Overvelde, and D. Liu, "Regulating Airflow Using Hybrid LCN for Soft Pneumatic Circuits," *Advanced Intelligent Systems*, vol. 7, no. 6, p. 2401069, 2025.

[146] H. P. Meijer, "Topology optimisation of mechanical metamaterials for prescribed snapback behaviour," Master's thesis, Eindhoven University of Technology, 2024.

[147] A. Comoretto, S. Koppen, T. Mandke, and J. T. B. Overvelde, "Embodying mechano-fluidic memory in soft machines to program behaviors upon interactions," *Device*, vol. 3, no. 10, 2025.

[148] R. D. Yamaletdinov, O. V. Ivakhnenko, O. V. Sedelnikova, S. N. Shevchenko, and Y. V. Pershin, "Snap-through transition of buckled graphene membranes for memcapacitor applications," *Scientific Reports*, vol. 8, no. 1, p. 3566, 2018.

[149] Veritasium, "This mechanism shrinks when pulled," YouTube video, 2025. [Online]. Available: https://www.youtube.com/watch?v=-QTkPfq7w1A

[150] AxMn, "Photo of 3d-printed countersnapping assembly," Printables.com, 2025, used under CC BY-NC-SA 4.0. [Online]. Available: https://www.printables.com/model/1343085-amolf-countersnapping-models-3mf-format-injection

[151] D. Lavrov, "This lego mechanism shrinks when pulled," YouTube video, 2025. [Online]. Available: https://www.youtube.com/watch?v=sHAn2J2k2mI

Summary

Instabilities in mechanical structures, such as buckling or snapping, have long been considered signs of failure, and engineers have traditionally learned to avoid them. In recent years, however, new design approaches have emerged wherein instabilities are instead embraced and harnessed. This paradigm shift has opened new ways to think about mechanics and embed functionalities within structures, such as sensing, computation, actuation, passive adaptability and programmability. Yet, these instabilities lead to more intricate systems, where the interplay between geometry and elasticity gives rise to exotic behaviors that often challenge intuition, thereby complicating analysis and the exploration of the design space. Here, we ask how to tame this complexity to uncover new mechanical behaviors. Through this experimental and computational study, we investigate the interplay of nonlinear building blocks to construct rich mechanical structures, demonstrate their properties and functionalities, and facilitate understanding, design and exploration.

We start this journey by developing a structure capable of exhibiting *countersnapping instabilities*, experimentally demonstrating that a mechanical system can suddenly become *shorter* in response to an increasing tensile force, or provide a sudden increase in tension upon snapping. We show that countersnapping arises from a force-displacement curve that self-intersects in the stable domain. We materialize the curve by shaping the interplay of five geometrically nonlinear building blocks assembled together, carefully designed and selected via an experimental and computational approach. We demonstrate that the unique profile of the force-displacement curve enables properties and functionalities unattainable by other mechanical systems, including unidirectional stick-slip actuation, programmable stiffness without altering the externally observed state, and passive resonance avoidance. We further show that coupling multiple countersnapping structures together leads to systems with greater stiffness programmability

or that are capable of surprising snapping chain reaction contrasting with the typically observed sequenced deformation pathways. Those results highlight the potential of constructing complex mechanical behaviors hierachically.

Next, we develop a computational framework to better simulate and explore the interaction between highly nonlinear components (such as countersnapping structures for example) through reduced-order models. The framework is built around *flexels*, elements that we introduced to single-handedly capture the complex mechanical responses of compound structures. A flexel is constructed from a given force-displacement curve that can be linear, nonlinear, nonmonotonic or even multivalued with multiple turns and intersections, extending the idea of a nonlinear spring. That curve is then converted into a bivariate energy potential that can subsequently be coupled to any geometric measure, allowing to build an ecosystem of building blocks with various intrinsic behaviors and modes of deformation, able to model springs, flexures, contact, pneumatic, hydraulic or cable-driven components. By combining flexels, we demonstrate that we can build reduced-order models of a wide range of nonlinear systems, including prestressed tensegrities, tape-spring mechanisms, interacting buckled beams and a soft gripper actuated through a metafluid. The framework is implemented and documented in the form of an open-access Python library to promote adoption among academics, engineers and researchers.

Finally, we bring together the concepts of countersnapping and flexel to study the interaction between a warped plate pushed against an array of nonlinear pillars, a scenario with relevant implications for the nanolithography industry. In particular, we numerically examine how the nonlinearities of the pillars could enhance the flattening (referred to as *geometric rectification*) of warped plates, where each pillar is a flexel with some highly nonlinear intrinsic behavior. We find that countersnapping pillars enable levels of rectification unattainable by linear or regular snapping pillars of similar stiffness. Beyond the specifics of that particular system, we develop and showcase a method to better understand and simulate the deformations of highly nonlinear systems through transition graphs, by building on the tools and concepts previously presented in the thesis.

Altogether, the findings presented in this thesis highlight the potential of nonlinear building blocks for understanding complex nonlinear mechanics and instabilities, as well as for constructing structures with advanced properties and

functionalities relevant for various applications. Through the ideas, tools and demonstrative examples presented throughout this thesis, we hope to inspire engineers, researchers and enthusiasts to further explore and design instabilities in mechanical structures.

Nederlandse samenvatting

Instabiliteiten in mechanische structuren, zoals knikken of doorslaan, werden lange tijd gezien als iets dat je moet vermijden. Ingenieurs leerden vooral hoe ze dit soort gedrag konden voorkomen. De laatste jaren is daar verandering in gekomen: er zijn nieuwe ontwerpmethoden ontwikkeld waarin instabiliteiten juist bewust worden gebruikt. Dit heeft geleid tot een andere manier van kijken naar mechanica, waarbij structuren extra functies kunnen krijgen, zoals meten, rekenen, actueren, zich passief aanpassen of programmeerbaar gedrag vertonen. Tegelijkertijd maken deze instabiliteiten systemen ook ingewikkelder. De combinatie van vorm en elasticiteit kan leiden tot onverwacht en soms moeilijk te begrijpen gedrag, wat het analyseren en ontwerpen lastiger maakt. In dit werk onderzoeken we hoe we met die complexiteit om kunnen gaan en hoe we daarmee nieuwe mechanische effecten kunnen ontdekken. Met experimenten en simulaties bestuderen we hoe niet-lineaire bouwstenen samen kunnen worden ingezet om rijke mechanische structuren te maken, hun eigenschappen te laten zien en het ontwerp en begrip ervan te verbeteren.

We beginnen met het ontwikkelen van een structuur die zogenoemde countersnapping-instabiliteiten kan vertonen. We laten experimenteel zien dat een mechanisch systeem plotseling korter kan worden wanneer je er harder aan trekt, of juist ineens een hogere trekkracht kan leveren wanneer het doorslaat. Dit gedrag ontstaat doordat de kracht-verplaatsingsrelatie van het systeem zichzelf kruist binnen een stabiel gebied. Die bijzondere curve realiseren we door vijf geometrisch niet-lineaire bouwstenen zorgvuldig te ontwerpen en samen te voegen, op basis van experimenten en berekeningen. We laten zien dat deze specifieke kracht-verplaatsingscurve eigenschappen mogelijk maakt die bij andere mechanische systemen niet voorkomen, zoals eenrichtings stick-slip-actuatie, programmeerbare stijfheid zonder dat het de krachten en verplaatsingen op de aanknopingspunten van het systeem verandert, en het

vermijden van resonantie zonder actieve regeling. Daarnaast tonen we aan dat het koppelen van meerdere countersnapping-structuren kan leiden tot systemen met meer instelbare stijfheid, of tot verrassende kettingreacties van doorslaan, in plaats van de gebruikelijke stap-voor-stap vervorming. Dit onderstreept hoe je door structuren hiërarchisch op te bouwen complexe mechanische gedragingen kunt creëren.

Daarna introduceren we een computationeel raamwerk om de interactie tussen sterk niet-lineaire componenten beter te kunnen simuleren en verkennen met vereenvoudigde modellen. Dit raamwerk is gebaseerd op het concept van flexels: elementen die in staat zijn om het complexe mechanische gedrag van samengestelde structuren in één bouwsteen te vangen. Een flexel begint met een kracht-verplaatsingscurve, die lineair of niet-lineair kan zijn, maar ook niet-monotoon of zelfs meerdere stabiele toestanden en kruisingen kan bevatten. Daarmee gaat het concept verder dan dat van een klassieke niet-lineaire veer. Deze curve wordt omgezet in een energiepotentiaal die gekoppeld kan worden aan verschillende geometrische elementen. Zo ontstaat een verzameling bouwstenen met uiteenlopend gedrag en verschillende manieren van vervormen, geschikt om onder andere veren, flexures, contact, pneumatische en hydraulische elementen of kabelsystemen te modelleren. Door flexels te combineren laten we zien dat we vereenvoudigde modellen kunnen maken van veel verschillende niet-lineaire systemen, zoals voorgespannen tensegrity-structuren, tape-spring-mechanismen, samenwerkende geknikte balken en een zachte grijper die wordt aangedreven door een metafluid. Het raamwerk is uitgewerkt in een open-access Python-bibliotheek, zodat anderen het eenvoudig kunnen gebruiken en uitbreiden.

Tot slot combineren we de ideeën van countersnapping en flexel om de interactie te bestuderen tussen een vervormde plaat en een rooster van niet-lineaire pilaren, een situatie die relevant is voor toepassingen in de nanolithografie. Met numerieke simulaties onderzoeken we hoe het niet-lineaire gedrag van de pilaren kan bijdragen aan het afvlakken van de plaat, een effect dat we geometrische rectificatie noemen. Elke pilaar wordt hierbij gemodelleerd als een flexel met sterk niet-lineair gedrag. We laten zien dat countersnapping-pilaren een veel sterkere rectificatie mogelijk maken dan lineaire of conventionele snapping-pilaren met een vergelijkbare stijfheid. Los van dit specifieke voorbeeld presenteren we ook

een methode om het gedrag van sterk niet-lineaire systemen beter te begrijpen en te simuleren met behulp van overgangsgrafen, voortbouwend op de concepten die eerder in het proefschrift zijn geïntroduceerd.

Alles bij elkaar laten de resultaten in dit proefschrift zien dat niet-lineaire bouwstenen veel potentie hebben voor het begrijpen en ontwerpen van complexe mechanische systemen met instabiliteiten. Ze bieden nieuwe mogelijkheden om structuren te maken met geavanceerde en nuttige eigenschappen voor uiteenlopende toepassingen. Met de ideeën, hulpmiddelen en voorbeelden in dit proefschrift hopen we ingenieurs, onderzoekers en andere geïnteresseerden te stimuleren om instabiliteiten niet langer te vermijden, maar juist actief te verkennen en te benutten in mechanisch ontwerp.

Publications

Journal publications

Ducarme, Weber, van Hecke, Overvelde, "Exotic mechanical properties enabled by countersnapping instabilities," *PNAS*, vol. 122, no. 16, p. e2423301122, 2025.

Ducarme, Weber, van Hecke, Overvelde, "Flexel ecosystem: simulating mechanical systems made from entities with arbitrarily complex mechanical responses," under review (https://arxiv.org/abs/2510.19741), 2025.

Author contributions

Chapter 2 of this thesis is based on a publication, as referenced on the first page. P.D., B.W., M.v.H., and J.T.B.O. designed research; P.D. performed research; P.D. contributed new analytic tools; P.D., B.W., M.v.H., and J.T.B.O. analyzed data; and P.D., B.W., M.v.H., and J.T.B.O. wrote the paper.

Acknowledgments

Looking back on these past years, I have come to realize how lucky I am to end up in a position where I could conduct the research presented in this thesis, within a positive, warm and encouraging environment. I am truly grateful for that opportunity and would like to address my thanks to all the people who helped me get there, directly and indirectly.

First and foremost, I would like to thank you, Bas. You trusted me in joining your group without knowing me much, and allowed me to bifurcate to a completely different and exciting career path, rich on both professional and personal levels. Over these past years, you have taught me a lot and have been a great source of inspiration. In particular, you have taught me to dare more in the lab; that trying things out *quick and dirty* can be more valuable and insightful than a bunch of numerical simulations. Your strong sense for nonlinearities has been eye-opening to me: intuition is a formidable asset in research, which has encouraged me to try to develop my own. Under your guidance, I have learned that conducting research is not simply about applying technical skills, but also about weaving together visions, research questions, discussions, brainstorms, intuition, and experimental and numerical creativity. Being around you has taught me to truly appreciate that everything can be questioned. Thank you also for trusting me and giving me the freedom to pursue my own research ideas. I truly admire you for that, and will try to adopt that style in my future career, as I believe that it is key to ultimately producing interesting and innovative research. Last but not least, thank you also for your kindness, positivity and constant availability throughout this journey.

I would like to also thank you, Martin. It has been a pleasure to have you as a co-advisor. I have really enjoyed discussing deep ideas of mechanics and physics with you. I have learned to appreciate thinking of physical systems more as a physicist, rather than only from an engineer's viewpoint. You made my PhD years

a truly richer experience. Thank you for your clear, precise and direct feedback throughout these past years; often helping me get unstuck and providing an uplifting perspective. Thank you for your support, care, passion and availability.

I would like to also thank you, Bart. I feel very lucky to have you as a co-advisor. You have been constantly available to answer all my questions and provide more big-picture review and feedback of my work. The link you enabled with ARCNL allowed me to learn more about mechanics and physics at smaller scales, which I found very enriching. Thank you for helping me navigate and set up the collaboration with ASML. Thank you for your calm and composed positivity, kindness and support during these past few years.

Thank you, Corentin, for accepting to be my promotor. You were also my first point of contact with the Dutch academic community, and you helped me connect with Bas, for which I am very grateful. Thank you for your kindness and the support you provided every time I asked you for advice.

Thank you, Wim and Marjan, for making me feel welcome at ARCNL even though I was predominantly working at AMOLF. Your enthusiasm toward my research and allowing me to present my work on multiple occasions (including to ASML's CTO) really lifted my spirit and boosted my confidence and motivation. I am truly grateful for your approachability, friendliness and support.

Thank you, Michel, for being my point of contact at ASML. Thank you for the interesting discussions, sharing your thoughts on my work from an application perspective, and for the opportunity to present my results to other ASML engineers. Thank you also for your help and support during these past few years. Also, thank you, Micha and Pavel, for dedicating time to this project, discussing ideas and guiding me in the first years of the PhD.

This PhD journey had its ups and downs. At times, it was challenging to keep my spirits up when things failed, progress was slow or reaching a dead end. I feel incredibly lucky to have been part of the Soft Robotic Matter group, which made this bumpy road easier to navigate. Our daily three-coffee-break routine was more than simple pauses; it was the opportunity to share ideas, exchange (funny?) jokes, laugh, vent frustrations, brainstorm crazy (and sometimes realistic) ideas, celebrate successes, share ideas to overcome obstacles, and genuinely look out for each other. Thank you to the whole group for creating this safe and

warm environment, where we supported each other, inside and outside the lab, at AMOLF, conferences, drinks, dinners, outings, or elsewhere. I am truly grateful for being part of this group, and cannot overstate how important and essential it was in order to reach the end of this PhD.

Thank you, Sergio, for being my colleague, friend and paranymph! You were always there to encourage me, support me, give honest feedback, and keep things in perspective throughout this journey. I will always remember you helping me shoot a last-minute video for APS, well after working hours. But that is just one example of the many occasions where you offered your time to help me (or anyone else around you). I also really enjoyed chatting with you, brainstorming ideas and being your *rubber duck*. Thank you for your generosity, incredible kindness and moral support. I really hope we can continue to collaborate together in the future.

Thank you, Shibo, for being such a kind, honest and caring office mate and friend. Sharing the office with you for almost four years was a true pleasure. Having someone so kind, caring and generous to whom I can talk at any moment of the day was a blessing. I really enjoyed our dinners, fetching plants for the office and playing music together. Thank you also for your moral support, your guidance, and showing me your beautiful country (Shanghai, Hangzhou, Suzhou, Beijing). I miss you already, and wish you the best in Belfast!

Thank you, Mannus, for being such a positive colleague and friend. It has been a real pleasure to spend time with you during these past few years, always with funny or unbelievable stories to tell and ready to laugh at my (stupid?) jokes. You have taught me to not worry too much, and keep a positive mindset. Thank you for that! I found both your research ideas and free spirit in life very inspiring. You genuinely made those past years a richer experience, thank you! I wish you all the best and hope we will cross paths again in the future, if you go some day outside the Amsterdam ring (*lol*).

Thank you, Alberto, for being such a passionate and nice colleague and friend. Your drive for developing new research ideas has been truly inspiring and contagious. I genuinely admire your attention to scientific aesthetics through presentations, videos, posters, figures, and photography. It has left a strong impression on me, inspiring me to incorporate a similar care into my own work. Thank you for your moral support, cheerful presence and sharing your passion for science.

I wish you the best of luck in Leuven, and I am looking forward to seeing what inspiring works and ideas you will publish next!

Thank you, Niels, for being such a dedicated, helpful and skilled technician. This work would not have been possible without your expertise. By keeping the equipment up and running, and a well-organized lab, you made my PhD a much smoother journey. I am very grateful for that. It has been a true pleasure to discuss 3d-printing techniques with you, share countless coffee breaks together, go bouldering a couple of times, and having you, there, ready to help anytime I needed support or advice. Thank you for your kindness, enthusiasm and availability during these past few years.

Thank you, Maddy, for being such an enthusiastic, honest and caring office mate and colleague. You directly made me feel welcomed in the group on the very first day of my PhD, and continued to be a kind, supportive and cheerful friend with whom I can speak of everything to this day. I feel very lucky to have met you, and wish you the best in the final stint of your PhD journey.

Thank you, Stijn, for your friendliness, openness, positivity and being such a *why* person. I really enjoyed our philosophical discussions and deeply technical sessions, which, at times, intertwined. It has been a true pleasure to spend time with you, and hope we will keep in touch as friends and collaborators in the future.

Thank you, Elif, for your enthusiasm, positivity and friendliness. I really enjoyed collaborating and chatting with you. You made me realize my units and *springable* might be potentially useful, and have learned a lot regarding LCE. It has been really fun and interesting to work together on your project. Thanks for everything. I wish you all the best for the future, and I hope we will stay in touch, even if you go back to Turkey.

Thank you, Bob, for being such an outspoken, direct and passionate person. It has been really fun to chat about your uncompromising takes on AI and politics. Thank you also for being such a fun and caring person (I still remember the birthday card you created and made other people sign). Thanks also for your precise and acute feedback. I wish you the best, and hope to cross paths again in the future.

Thank you, Ze, for being such a cheerful and sunny colleague and friend. Thank you for your moral support, kindness, friendliness and guidance. I will

always remember our time spent in China and on the Great Wall. I wish you all the best in your career in the industry.

Thank you, *dr.* Maziar, for your calm and grounded presence. I found your career path as well as your research truly inspiring. I admire your intuition and expertise for design, which inspires me for my future career. Thanks also for trying out *springable*, your kindness, humor and positivity. All the best in the future!

Thank you, Katrien, for your enthusiasm, true passion for design, and constant stream of new inventions, which amaze and inspire me. Thank you for caring about the cohesion of the group; always down for coffee breaks or after-work dinners. It has been a pleasure to work together with you, and I hope we will continue to work on *popping straws* and other projects in the future. All the best for the rest of your (already impressive) PhD adventure.

Thank you, Nienke, for your friendliness, easy-going nature, and always willing to laugh at my jokes (right?). I enjoyed our coffee and snack breaks, as well as, shooting funny movies for our colleagues. I wish you all the best for the rest of your PhD, already full of trophies and awards, and am looking forward to seeing your heart breaking the one-million-beat mark!

Thank you, Sumit, for your friendliness, enthusiasm, outspokenness and sharing your honest views on academia and the world. I also enjoyed discussing with you the physics and mechanics at smaller scales. All the best for your career at TNO; I hope to cross your path again in the future.

Thank you, Luuk, for your positivity and always being there to lighten up the mood. You made me laugh a lot, and it was really fun and enriching to hear wise or entertaining stories from an *old* man like you. Congrats on becoming professor. I wish you all the best.

Thank you, Charlotte, for your friendliness, passion for design, and being a fun office mate. It was inspiring to witness your creativity, sense of aesthetics and fabrication skills. It was fun to go to the London Design Biennale with you, and connect with the Automorph community together. All the best in the future.

Thank you, Arthur and Zak, for being my interns. I am truly proud and honored to have been your mentor, and I learned a lot with you. Thank you for the good work and drive to collaborate on my projects. I wish you both all the best for the future.

Thank you, Lio, for your positivity and for always willing to join activities, such as going on a bike ride all the way to the seaside for example. Your enthusiasm and friendliness were really uplifting during this last year of my PhD. I am sure your future will be bright, whether it is in research, design or engineering (given your impressive *portfo-Lio*).

Thank you, Jack, for being such a entertaining and caring person. I really enjoyed our long morning coffee breaks, chats and listening to your funny stories, told with your ~~British~~ Scottish accent. I wish you all the best for the final stint of your PhD.

Thank you, Yorick, for sharing your passion for mathematics and science. I really enjoyed chatting with you, and was inspired by your fearless adventures. I am sure a bright future awaits you.

Thank you, Ellen, my fellow Belgian, for visiting the group with your cheerful and sunny personality. I enjoyed working with you, learning about self-healing materials, showing you *springable*, and going to GRC together. I wish you all the best at the VUB, and hope we can resume some collaboration some day.

Thank you, Dennis, Laura, Astrid, Olivier, Erwin, Louise, Tanaya, Oskar, Manon, Tess B., Kun, Luca, Milo and Jelle. Your stints in the *Soroma* group contributed to a more enriching and enjoyable PhD journey, for which I am truly grateful. I wish you all the best.

This work was also enabled by the great collaborative and positive environment at AMOLF. I am truly grateful to all the support departments, who were always ready to provide help and guidance throughout this entire PhD. There are many people I would like to thank, and while I cannot name everyone, I am especially grateful to Remco from the purchase department, Remco from ICT, Hinco and Dion from the technicians' office, Petra from communications, and Max from mechanical engineering.

I would also like to thank the many people I have met through AMOLF directly or indirectly, who made this PhD adventure a more enjoyable and richer experience.

Thank you, Dhawal, for your friendliness, openness and passion. You have been a great captain, coach, leader, founder and player of the AMOLF futsal team. I have really enjoyed spending time with you during this past few years. Thank

you for everything, and being such a good friend. Also, thank you, Lisha, for your positivity and sunny personality. I feel very honored to have been invited to your many home parties and your wedding. Thank you both for making me discover your beautiful country.

Thank you, Bernat, for your positivity and friendliness. It has been a real pleasure to chat with you about mechanics, and exchanging research and design ideas. Thanks also for scoring so many goals for the AMOLF team! I hope we will stay in touch as friends and collaborators in the future.

Thank you, Cyrian, Steve F., Thibault L., Felix C., Junxiao, Dor, Paul B., Margot, Colin, Parisa, Finn, Theophile, Sima, Saeed, Wenjie, Hujie, Sujeeka, Tess H., Mees, Agustin O. A., Katrien S., Nachi, Mohammad H. Z., Daan H., Wenfeng, Giorgio, Lars P., Agustin I. R., Benjamin G., Gregor C., and many more I am probably forgetting at the moment of writing these lines (sorry in advance), for making this PhD experience more pleasant and enriching.

Finally, I would like to thank my friends and family for paving the way toward the start of my PhD, as well as the moral support throughout.

First and foremost, thank you, Loïs, for being such a great friend, partner and paranymph. You have provided continued and unconditional emotional support during these past few years. You have pushed me to celebrate successes, comforted me in more difficult times, provided feedback on many of the works I carried out during my PhD, rooted for me, reassured me, had my back and stayed on my side even when I had no time left to dedicate to anything else than the PhD. I will forever be grateful for your help and sacrifice. Thank you so much. I love you a lot. Also, thank you, Toby, my dear cat. Probably without realizing it, you have provided a great deal of moral support over these past years. I adore you very much, despite you destroying the couch, the curtains, many glasses, some bowls, my headphone cables and leaving a dead pigeon lying on the apartment floor.

Thank you, Benjamin, Louis and Estel, for always being there for me and being such great friends, even though I am not often available. I am truly grateful for the moral support and the good times we have had together for so many years!

Thank you, Mathilde, for always being there for me, always positive, easy-going and ready for new and crazy adventures. Your journeys and approach to

life are truly inspiring to me; you inspired me to travel the world and open my horizons, which ultimately led me to end up in Amsterdam. I am truly grateful. Thank you for everything!

Thank you, Mattias, Trystan, Vincent, Audrey and Adri, for your positivity, enthusiasm and good mood. It is always a pleasure to catch up, which I always find very uplifting. Thank you for your moral support.

Thank you, Philipp, for your calm, positivity and friendliness; always ready to hang out and play a game of chess, even though you know you will lose. Thank you for being such a good friend these last years.

Thank you, Tamar, for your friendliness and openness. It is always a pleasure, and *gezellig* to have you at home and chat about life!

Thank you, Alphons, Dorien, Lara, Rogier, Sophie and Mats, for always being so supportive, positive and generous. I am truly blessed and lucky to have you all close to me. Since I moved here, you have made my time in Amsterdam a great experience. Thank you very much.

Thank you to all my teachers and professors from Vinalmont, Collège Saint-Quirin, University of Liège and University of Ottawa, who ultimately made it possible for me to begin my PhD here in Amsterdam.

Last but not least, thank you Maman, Papa, Dimitri, Justine, Céline, Julien, Bonne-Maman d'Havré, Papy, Bonne-Maman d'Horion, Bebeth and Ginette, without whom I would not have been able to be where I am right now. Thank you for your unconditional support. I will forever be grateful. I love you very much.