

Reprogrammable, In-Materia Matrix-Vector Multiplication with Floppy Modes

Theophile Louvet, Parisa Omidvar, and Marc Serra-Garcia*

Matrix-vector multiplications are a fundamental building block of artificial intelligence; this essential role has motivated their implementation in a variety of physical substrates, from memristor crossbar arrays to photonic-integrated circuits. Yet their realization in soft-matter intelligent systems remains elusive. Herein, A reprogrammable elastic metamaterial that computes matrix-vector multiplications using floppy modes—deformations with near-zero stored elastic energy is experimentally demonstrated. Floppy modes allow to program complex deformations without being hindered by the natural stiffness of the material; but their practical application is challenging, as their existence depends on global topological properties of the system. To overcome this challenge, a continuously parameterized unit cell design with well-defined compatibility characteristics is introduced. This unit cell is then combined to form arbitrary matrix-vector multiplications that can even be reprogrammed after fabrication. The results demonstrate that floppy modes can act as key enablers for embodied intelligence, smart micro electro mechanical systems(MEMS) devices, and in-sensor edge computing.


Within the paradigm of physical computing, mechanics has gathered considerable excitement for a range of special-purpose but highly relevant applications.^[6] Mechanics offers the capability to directly process signals such as speech^[7] or gait^[8] without transduction into the electrical domain—facilitating zero-power, in-sensor information processing; it forms the basis for embodied intelligence in soft robotic structures—enabling such robots to mimic the agility and dexterity of living organisms,^[9–11] and, when scaled down to microscopic dimensions,^[12,13] can perform computations with ultra-low energy consumption^[14,15]—enabling intelligent devices that can operate under extreme power constraints. The overarching challenge in achieving mechanical computation lies in mapping relevant information-processing operations, such as matrix-vector multiplications, into the natural responses of mechanical systems.

1. Introduction

Linear transformations, encoded in matrix-vector products, are ubiquitous in modern computing; with applications spanning from deep learning to 3D graphics^[1] and data compression.^[2] In fact, universality theorems prove that any function can be approximated by a network of programmable linear transformations, interconnected by fixed nonlinear activation functions.^[3] This widespread applicability has motivated the development of specialized hardware to compute matrix-vector products, ranging from the well-established graphics processing unit (GPU) to emergent physical computing approaches such as photonic integrated circuits^[4] and memristor crossbar arrays.^[5]

In this work, we experimentally demonstrate a soft mechanical metamaterial that computes a matrix-vector multiplication using floppy modes (zero modes). The article is structured as follows: we start by decomposing the matrix-vector product into a network of elementary operations, each of which will be computed by a single metamaterial unit cell (**Figure 1**). Then, we implement this unit cell in a planar network of point masses subject to ideal constraints. The resulting unit cell presents two zero modes that branch and cross—as required by the matrix-vector multiplication, and preserves deformation compatibility when tiled. Afterward, we investigate, using finite element method simulations, how a real metamaterial geometry deviates from the ideal point-mass model. We use automatic differentiation^[16] to adjust the design and preserve accuracy in the presence of nonzero bending stiffness and finite compliance, and discuss the limitations of the metamaterial arising from such real-world effects. Finally, we experimentally realize the metamaterial using water-jet cut rubber, and characterize its displacement using optical flow. We experimentally characterize the metamaterial unit cell, a 2×2 matrix-vector product, and a tunable unit cell that can be reversibly switched between multiple matrix coefficients. We observe that the metamaterial can compute the desired matrix-vector product with remarkable accuracy, even in the presence of fabrication tolerances and material nonidealities. These results demonstrate that floppy modes can be harnessed to implement intelligent responses in elastic systems.

T. Louvet, P. Omidvar, M. Serra-Garcia
AMOLF
Amsterdam 1098XC, The Netherlands
E-mail: m.serragarcia@amolf.nl

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202500062>.

© 2025 The Author(s). Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202500062

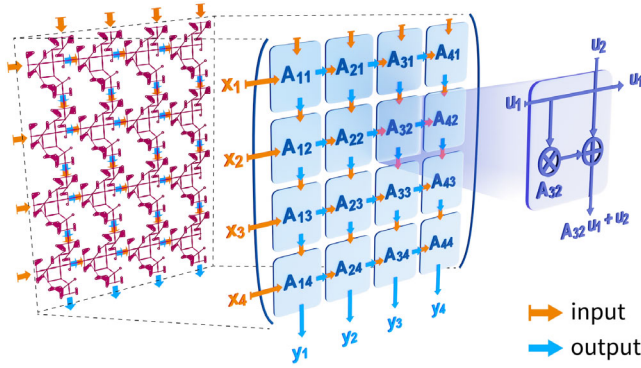


Figure 1. A metamaterial (red) computes a matrix-vector multiplication by decomposing it into elementary operations (blue), corresponding to individual matrix elements. Each elementary operation (purple) is implemented by a metamaterial unit cell. The input vector is presented as a displacement applied on the input control rods (orange, x_i) and the result is determined by measuring the displacement of the output rods (blue, y_i). Each internal output is connected to an input, ensuring that the resulting metamaterial is frustration-free irrespective of the particular matrix being implemented.

2. Design

Among the deformation responses of mechanical systems, floppy modes, or zero-modes, are unique in the fact that they are associated to a vanishingly low amount of stored elastic energy, that is, they require small forces to actuate. Thus, they are a prime candidate for computation in passive elastic systems where deformations must be driven by the energy available in the signal. Floppy modes originate in a rank deficiency of the rigidity matrix, caused by either an insufficient number of constraints,^[17–20] or by the topological properties of the system.^[21] Their global nature and topological connections not only point toward exciting research questions, but also pose a challenging design problem for applications that require a large number of on-demand floppy modes, such as matrix-vector multiplication (a matrix-vector product requires a number of modes equal to the dimension of the input space). Several approaches toward on-demand floppy modes have been introduced. Combinatorial methods draw from a small set of unit cells with well-defined compatibility characteristics.^[17,22,23] They can be highly efficient, but are limited by the small number of unit cells and large design space. Techniques based on iterative addition of constraints can accurately produce designs with a few desirable modes, but introduce additional unwanted modes as a byproduct.^[24] Finally, machine learning methods can identify the number of modes corresponding to a particular unit cell,^[25] but they do not yet possess the capability to solve inverse design problems. While progress has been made in the particular case of line modes,^[23,26–29] branches and crossings—necessary for the realization of mechanical information-processing circuits—still pose a challenge for these methods. We will tackle the challenge of designing a material with a large set of complex floppy modes by introducing a unit cell with well-defined compatibility characteristics, that can be combined to achieve the desired response.

In this section, we introduce a metamaterial design that performs a matrix-vector multiplication $\vec{y} = A\vec{x}$. The input vector \vec{x} is presented to the system by prescribing the displacement of a set of input degrees of freedom, and the output result \vec{y} is read out by measuring the displacement of a different set of degrees of freedom. Since the system should be able to respond to any (small) input vector with no resistance, the space of floppy modes should span all possible input displacement combinations. At the same time, we require the output displacement to be entirely prescribed by the input vector, meaning that the system should not present additional floppy modes. For large input spaces, this involves designing a large number of modes—beyond the limit of direct optimization methods. We can overcome this limitation by decomposing the matrix-vector multiplication as a planar network of computation tiles (Figure 1), whose inputs u_1 and u_2 are related to the outputs v_1, v_2 as $v_1 = A_{ij}u_1 + u_2$ and $v_2 = u_1$. The coefficient A_{ij} is controlled by the angles such that $A_{ij} = \frac{1}{2}(\tan(\theta_3)\tan^{-1}(\theta_6))$. In this idealized case, the coefficients can take any arbitrary value, positive or negative. Each tile can be understood as individually implementing a small matrix-vector multiplication, with the form

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} A_{ij} & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (1)$$

Each individual computational tile is physically implemented by a metamaterial unit cell. Since unit cells are only required to perform a 2×2 matrix-vector multiplication, they require only two floppy modes. This simplified requirement can be met with existing floppy mode design methods—here, we solve it by hand-designing the unit cell.

By construction, the tiling in Figure 1 exhibits the minimal number of zero-modes: there should be at least as many zero-modes as inputs in order for the transformation to span the entirety of the input space. Each individual row added to the system will add $2n_{\text{col}}$ (number of columns of the matrix) floppy modes, but will also incorporate $2n_{\text{col}} - 1$ constraints (n_{col} constraining the input u_2 with the output v_1 of the upper row, and $n_{\text{col}} - 1$ intra-row constraints). This will increase the number of floppy modes by one, reflecting the fact that the extended system contains an extra input. Adding an additional column will add $2n_{\text{row}}$ (number of rows of the matrix) floppy modes, but also $2n_{\text{row}}$ constraints (n_{row} constraining the input u_1 of each site with the output v_2 of the previous column, $n_{\text{row}} - 1$ intra-column constraints plus an extra constraint setting the input u_2 of the top-most site to zero, as shown in Figure 1). Since the constraints always connect an output (by definition prescribed) with an input (by definition free), and the directed computation graph presents no cycles, the constraints will neither be redundant nor introduce frustration. Such designer compatibility relation is reminiscent of traditional combinatorial design approaches for floppy mode systems.^[22] However, here, the site is continuously parameterized by A_{ij} , thus, the same design can encode a variety of matrices by designing the individual building blocks. This is a key to our ability to design and reprogram the metamaterial.

The unit cell design consists of nine sites (Figure 2a,b) that can move on a plane. Before introducing any constraints, the sites are free to move along both x and y axes, and thus the constraint-free unit cell has 18 floppy modes. As a consequence

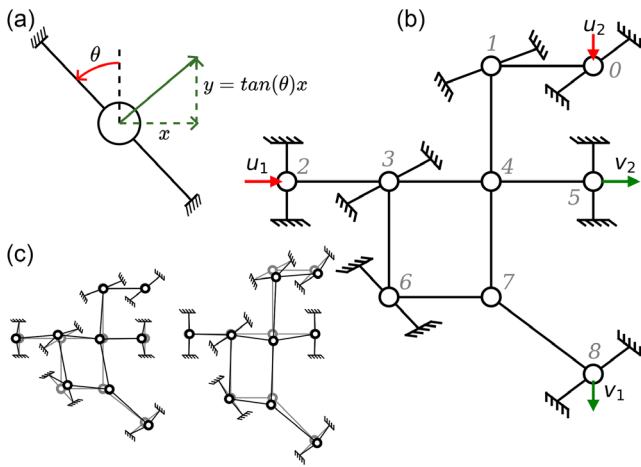


Figure 2. a) Single degree of freedom, the angle of the support beams control the coupling between x and y , b) the unit made of 9 degree of freedoms(DOFs). The angle at sites 3 and 6 (θ_3 and θ_6) controls the coupling between u_1 (u_2) and v_1 . c) The 2 floppy modes of the system.

of the Maxwell–Calladine theorem, restricting the unit cell to the two prescriptive modes requires incorporating 16 linearly independent constraints. To produce an initial guess for the metamaterial design, we start by modeling the constraints as ideal rods, for which bending stiffness is zero; and we consider the displacements infinitesimal, so as not to alter the length of the rods. We incorporate two classes of constraints: local constraints (Figure 2a)—represented by a pair of rods connected to a fixed boundary conditions, which restrict the site motion to a specific line along the x and y axes—introducing a rank-1 condition of the form $(\cos \theta, \sin \theta)^T(x, y) = 0$; and pairwise constraints, which are realized by rods connecting two sites, resulting in a rank-1 condition of the form $(\cos \theta, \sin \theta)^T [(x_1, y_1) - (x_2, y_2)] = 0$ between the displacements of the two sites.

The proposed unit cell design is introduced in Figure 2b. The two floppy modes (Figure 2c), spanning the two-dimensional input space, can be understood as follows: when the input u_1 is displaced, the rank-1 conditions introduced by the couplings between sites 2-3, 3-4, and 4-5 cause the output degree of freedom v_2 (horizontal displacement of site 5) to follow the input degree of freedom (satisfying the condition $v_2 = u_1$ required by Equation (1)). At the same time, the condition introduced by the local constraints in site 3 couple its vertical displacement to its horizontal displacement, with a proportionality factor given by $\tan \theta_3$. This vertical displacement is then transferred to site 6, and converted to a horizontal displacement, scaled by the factor $\tan^{-1} \theta_6$, and transferred to the horizontal displacement of site 7. Similarly, the displacement u_2 is scaled by $\tan^{-1} \theta_0$ and $\tan \theta_1$, and then transferred through sites 1-4 and 4-7 to the vertical displacement of site 7. The diagonal coupling between sites 7 and 8 ensures that the displacement along the diagonal direction of site 8 matches the diagonal displacement of site 7. This diagonal displacement is the projection on $1/\sqrt{2} (dx_7, dy_7)$, where dx_7 is the scaled displacement of input u_1 , and dy_7 is the scaled displacement of input u_2 . This projection amounts to adding

the horizontal and vertical displacements together. A local constraint on site 8 prevents the output site to move in the tangent direction to the 7-8 couplings. The output vertical displacement amounts to the projection of the diagonal displacement into the y -axis, introducing a total scaling of $1/2$. By setting the angles θ_0 and θ_1 to compensate for the factor of $1/2$, and setting the angles θ_3 and θ_6 to implement a coefficient $2A_{ij}$, the resulting system satisfies the required input–output characteristics.

The final system retains two floppy modes after the addition of the 16 constraints (7 local constraints plus 9 coupling constraints). These two resulting floppy modes are plotted in Figure 2c. Since any prescribed input displacement can be expressed as a combination of these modes, in the ideal rod approximation, this (idealized) matrix-vector product requires no energy to be actuated. The floppy modes in our system are reminiscent to line-modes^[23,26–29] observed in two-dimensional soft matter systems. A remarkable novelty of our work is that we are able to design line modes that can branch and cross.

3. Numerical Validation

Until now, we have considered a design composed of ideal rods (zero bending stiffness) under infinitesimal displacements. However, these assumptions do not accurately describe the dynamics of experimental samples. To capture the effect of the finite bending stiffness of the rods, we construct a three-dimensional finite element simulation of the metamaterial (see Appendix A). To compensate for some of these errors, we fine-tune the geometry by performing a gradient descent optimization. We define a loss $\eta(\vec{\theta})$ as the norm between the simulated and desired transformation, and then we use automatic differentiation to compute its gradient with regards to the design parameters $\vec{\theta}$. At every iteration i , we update the parameters such that $\vec{\theta}_{i+1} = \vec{\theta}_i - \lambda \frac{\partial \eta(\vec{\theta}_i)}{\partial \theta_i}$, where λ is the learning rate.

Attempting to compensate for real-beam effects by adjusting the beam angles illustrates the limitations of the system (Figure 3a): we see that it is possible to fine-tune the geometry, however, when the aspect ratio (length over width) of the beams is below a given threshold, we see saturation of the loss function. This indicates that the system cannot encode the desired transformation. The origin of this error can be found in the relation between beam angle θ_3 and matrix coefficients (Figure 3c). While systems composed of ideal rods (infinite aspect ratio/zero bending stiffness) can approximate any matrix coefficient, rods with finite aspect ratio have a limit on the maximum absolute value the matrix coefficient A_{ij} can reach. This limitation arises from the ratio between longitudinal and bending stiffness in finite aspect ratio rods: Increasing rod angles has the effect of stiffening the input of the unit cell. Beyond a specific angle, the decrease in input displacement due to the finite longitudinal stiffness of the preceding rods exceeds the increase in output displacement due to the increase in angle. At this point, the matrix coefficient has reached saturation. We quantify this effect by determining the maximum matrix size that can be represented using a given aspect ratio. To do so, we consider 5 random matrices of each size, and perform an optimization to determine the minimum aspect ratio at which it is possible to achieve arbitrary

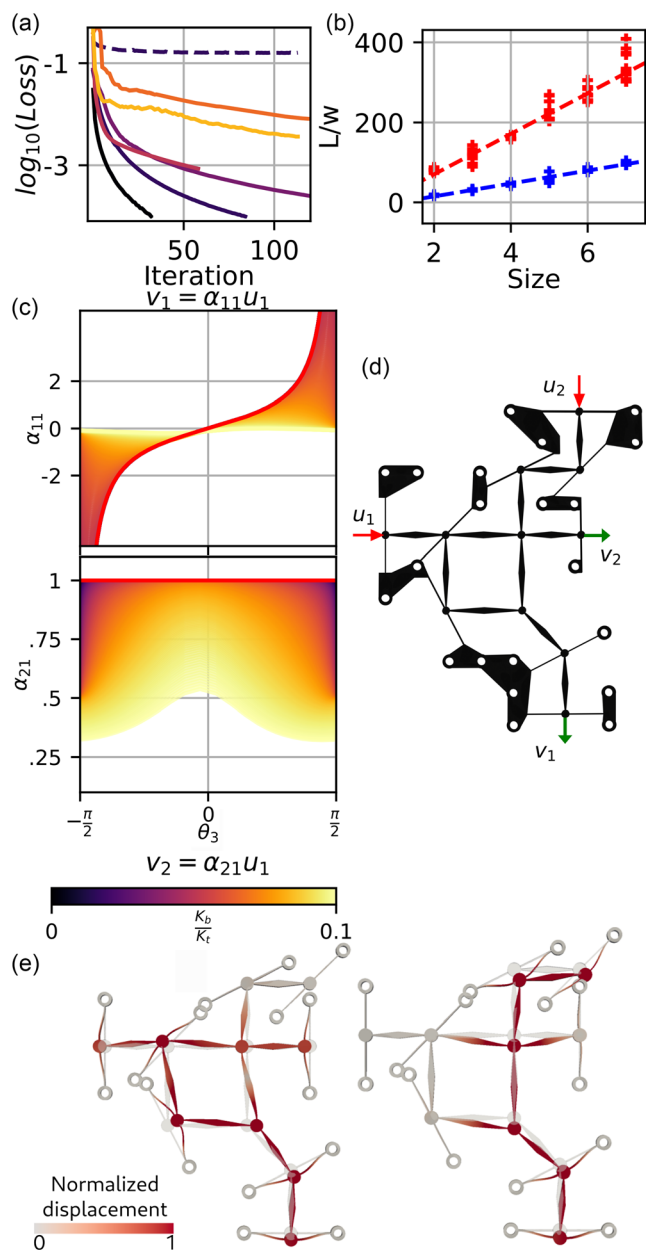


Figure 3. a) Evolution of the loss function during the optimization of the design from 1×1 lattices (black) to 6×6 (yellow). Optimization of the 2×2 matrix is also run with a larger beam width (see dashed line), and we observe saturation of the loss: the system cannot encode the target transformation. b) Minimum aspect ratio required to accurately approximate a set of unitary random matrices. In the red curve, only θ_3 is varied, while for the blue curve, 5 angles are allowed to vary. c) Input/output relationships between u_1 and v_1/v_2 , as a function of the angle of site 3, depending on the ratio of bending K_b to tensile K_t stiffness. The red curve is the response of the idealized model. We see that as the aspect ratio is reduced, both the response at each given angle and the maximum attainable matrix coefficient are lowered. d) 3D model of a unit cell, in experiments the white holes are clamped with screws, and there are no other clamps. e) The first two normalized eigenmodes of the system.

accuracy. The results are plotted in Figure 3b—highlighting a linear relation between aspect ratio and maximum matrix size. Given that MEMS resonators with aspect ratios up to $1000:1$ ^[30] are possible, we anticipate that our results can lead to matrix-vector products with dimensions above 64×64 . After optimization, we observe that the first 2 eigenmodes of the geometry (see Figure 3e) have a similar shape to the ideal model (Figure 2c)

4. Experimental Section

The experiments (see Figures 4 and 5 as well as Appendix B) on a single unit cell reveal that, in the linear regime (where input displacements remain below a few mm), the displacement transfer matrix coefficients between input and output are within 20% of the target value from the ideal unit. Beyond the linear regime

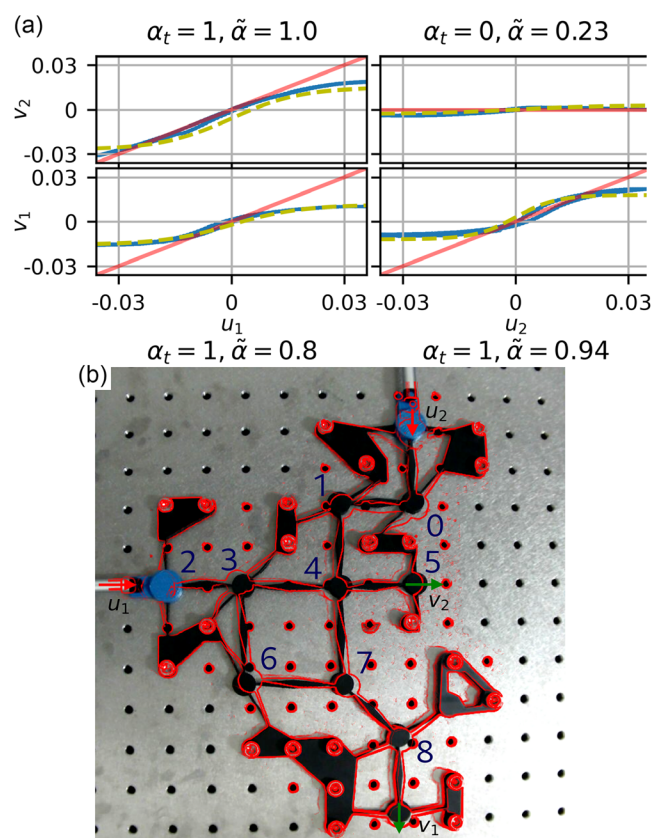


Figure 4. a) Complete experimental input/output transfer relation for a single unit cell, the measurements, in blue, are gathered from an optical flow algorithm. The displacement is plotted as the strain with regards to the width of the sample (250×275 mm). The fitted linear response (solid red lines) is within 20% of the target matrix value. The dashed olive line corresponds to a sigmoidal fit of the output displacement. b) Picture of the experimental setup during characterization. The blue components are the output of the linear actuators. The red lines represent the deformation of the unit, it is computed using an edge detection algorithm and some edited to remove the edges of holes around the sample.

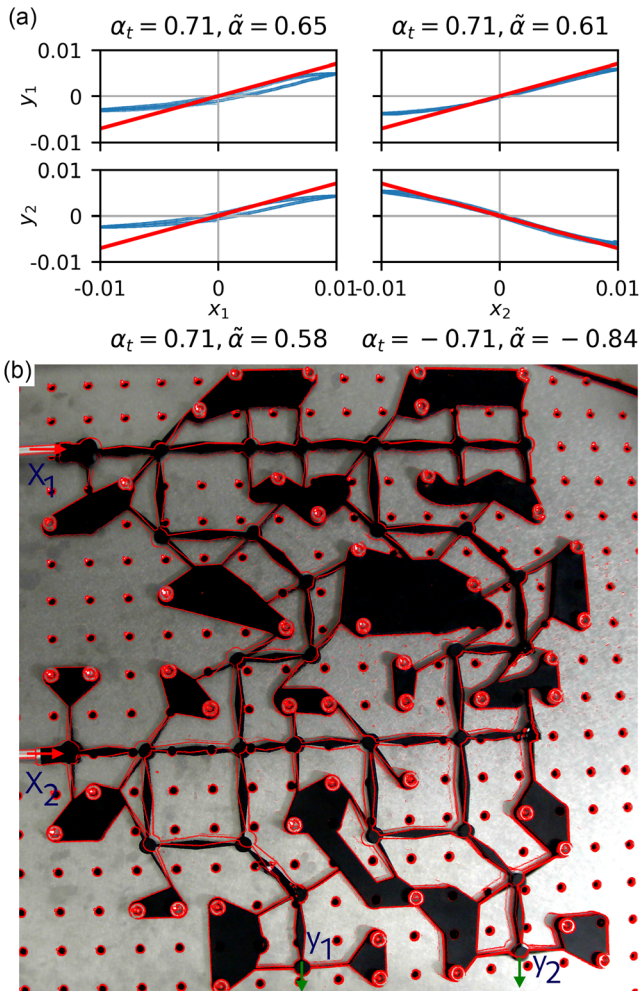


Figure 5. a) Input/output relationship of a 2×2 lattice. We plot the strains with regard to the width of the system (325×400 mm). The matrix of the transformation is within 15% of the target matrix. b) Picture of the 2×2 lattice with the deformation plotted in red. The first output is zero because the components cancel each other out, while they add for the second output. Videos of the experiment are provided as Supporting Information.

(around 2 mm—for a unit of 250×275 mm), the output displacement saturates. Interestingly, the saturation function resembles a sigmoid function. In neural networks, sigmoid functions are one of the most commonly used nonlinear activation functions—thus, the saturation characteristics of the material present an exciting prospect for the realization of full machine learning models in elastic systems. In addition to the sigmoidal-like saturation, the experimental results present two deviations from the ideal response: first, we observe that the response presents some hysteresis—which we attribute to the viscoelastic response of the polymer material and other nonidealities (Appendix B). Second, the linear region is asymmetric—saturation occurs earlier under compressive loads than under tensile loads. Experiments on a metamaterial consisting of two unit cells reveal a similar response—with 12% error between

the objective matrix and the measured linear transformation, and similar hysteresis and saturation characteristics.

We now proceed to demonstrate a programmable metamaterial that can be reconfigured to implement different matrix-vector products into a single device, that is, a metamaterial where it is possible to switch between transformations, while still preserving zero modes. Reconfigurable floppy modes are a timely area of research, with existing results focusing on controlling the number of modes.^[31,32] The requirements for a programmable matrix-vector product are different, as the number of modes shall remain constant while the mode shape is altered. Our solution to allow for multiple possible values for the transformation coefficients is to incorporate additional rods in the site 3, the site that defines A_{ij} , the coordinate transformation. Increasing the number of constraining rods causes the system to be overconstrained—thus eliminating the zero modes. To achieve programmability, only a subset of the rods is active in each configuration, which is equivalent to only having two-rods connecting to the site, just like in the original design. Given the different angles of the sets of rods, the transformation will be different depending on which set is active. We accomplish this by incorporating bistable elements into the metamaterial.^[33] In particular, we use a variable stiffness-compliant mechanism,^[34] connected to the base of each of the support rods (Figure 6a). When the compliant mechanism is in the low-stiffness configuration, the rods are free to move, and the constraint does not act into the

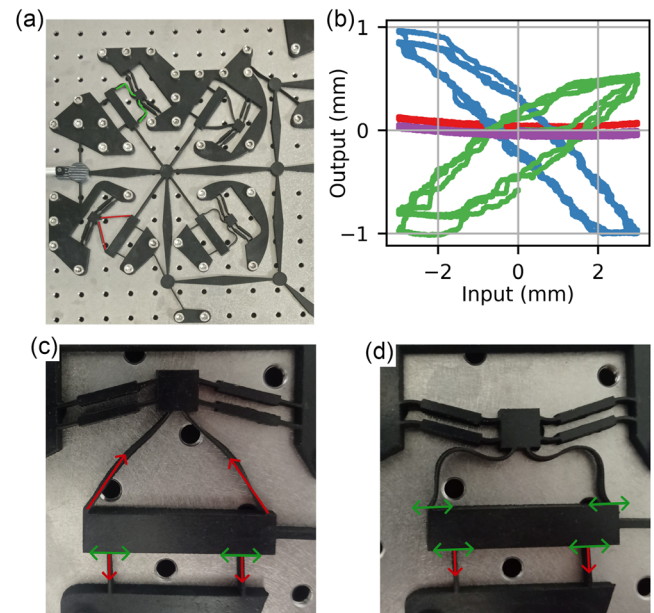


Figure 6. a) Picture of the sample during characterization with the lines showing a compliant mechanism in the low-stiffness (green) and high-stiffness (red) configurations. In the low-stiffness configuration, the beams are buckled, while in the the high-stiffness configuration, the beams are straight. b) Experimental results demonstrating the switching between three different matrix coefficients, $A_{ij} = +.25$ (green), $A_{ij} = 0$ (purple), $A_{ij} = -.33$ (blue). Switching between green and blue curves does not alter the number of floppy modes. c,d) Variable stiffness mechanism in the clamped position (c), where the tension in the upper beams holds the shaft in place, and (d) in the ‘free’ position the buckling beams allow the shaft to move.

system. By switching the compliant mechanisms between different configurations, the unit cell coefficient A_{ij} can be switched between three different values (Figure 6b). When only one set of rods is activated, we see that the system exhibits two different transformations. In the case where both are activated, the unit is overconstrained, while when both are in the ‘free-state’, it is underconstrained. In both these cases, the input and output are decoupled. The programmable unit cell shows an increased hysteresis compared to previous samples due to slack in the fabricated compliant mechanism. The unit tested here only shows 2 possible transformations, but there are several avenues to increase the resolution in the matrix coefficients, including using more than four switchable support rods, using this method for multiple sites or cascading multiple unit cells by cascading programmable units.

5. Discussion

The present work demonstrated a mechanical metamaterial capable of reprogrammable matrix-vector multiplication.^[35] The results can be interpreted in the light of the in-materia computing paradigm, where nontraditional material responses—here, floppy modes—are harnessed to perform computations—in this case, matrix-vector products. The key factor limiting the matrix size is the aspect ratio of the supporting beams, with matrix sizes above 64×64 attainable within present technology. Since, relevant machine learning such as phoneme recognition can be efficiently encoded in classifiers acting on a as little as 39 features,^[36,37] the proposed passive matrix-vector multiplier can have an enabling role toward passive speech-recognizing MEMS.^[7] Additionally, we expect more precise fabrication methods to improve the accuracy of the experiment.^[38–55] We have also shown a reprogrammable unit cell where every weight A_{ij} can be adjusted from a pair of values while preserving the number of zero modes. This level of programmability—that can be expanded using more complex designs—has the potential to be applied to relevant problems, as machine learning models are known to be resilient to extreme weight quantization, down to 3 values per weight.

Appendix A: Simulation and Optimization

We simulate and optimize the metamaterial using the finite element package FEniCSx.^[56–58] To do so, we partition the structure into subcomponents 7. Each subcomponent is obtained by extruding a two-dimensional geometry, composed of lines and circle arcs. This results in a 3D geometry, while there are no out-of-plane effects in the system, the choice of doing 3D simulations comes from compatibility issues with existing tools. The lines and arcs are generated using Python functions, from a set of high-level parameters, such as centers, angles, radiuses, and thicknesses. We use JAX to map the high-level parameters into the elementary features (arcs and lines) to allow for automatic differentiation. Automatic differentiation is used to assemble the geometry from a graph of interconnected components. The graph specifies the topological characteristics of the system—that is, which component is connected to which. Then, by computing the Jacobian of the distances between

component boundary contact point with respect to the high-level parameters, J_{contact} , a distance minimization algorithm is capable of automatically constructing the full metamaterial geometry—enforcing continuity between components.

To simulate the system, we extract the stiffness matrix for every component using second-order Lagrange elements and a linear elastic energy function. Since we are only interested in the degrees of freedom located on boundaries; we use the Guyan reduction.^[59] This computation allows us to determine the exact, statically condensed component stiffness matrix, that contains only a few hundred rows and columns. The determination of the statically condensed stiffness matrix, requiring a matrix inversion, is performed on the GPU—we found out that, given the relatively small size of each component (a few thousands of degrees of freedom), GPU-based dense linear algebra achieved the highest performance. The matrices for each component can be calculated fully in parallel.

To be able to assemble a full system matrix from the reduced component matrices, we make sure that all connecting boundaries are meshed with identical, regular grids. Then, we assemble the statically condensed stiffness matrix from every component into a global stiffness matrix. This global stiffness matrix can also be decomposed in inner degrees of freedom (corresponding to components that are connected to each other (not accessible from outside the metamaterial), and external boundaries (that correspond to the system inputs and outputs). Since we are interested in the response of the metamaterial to displacements in the external boundaries, we apply a second Guyan reduction. This second Guyan reduction can be very expensive for large systems. Thus, it is performed recursively, assembling intermediate structures with a larger and larger number of components until the full system is assembled.

Once the global stiffness matrix has been statically condensed to remove internal boundaries, we divide its degrees of freedom into input and output displacements

$$K_{\text{global}}(\vec{\theta}) = \begin{pmatrix} K_{\text{in,in}}(\vec{\theta}) & K_{\text{in,out}}(\vec{\theta}) \\ K_{\text{in,out}}^T(\vec{\theta}) & K_{\text{out,out}}(\vec{\theta}) \end{pmatrix} \quad (\text{A1})$$

where $\vec{\theta}$ is the vector of high-level geometric parameters.

The transfer function between input and output displacements can be calculated as $T(\vec{\theta}) = -K_{\text{out,out}}^{-1}(\vec{\theta})K_{\text{in,out}}^T(\vec{\theta})$. Once this transfer matrix $T(\vec{\theta})$ has been calculated, we prescribe the inputs to move along a specified direction—given by the displacement direction of the actuators, and project the resulting output into the direction of measurement. The projection $A(\vec{\theta}) = \hat{P}_{\text{out}}T(\vec{\theta})P_{\text{in}}$, where P_{in} and P_{out} are the input and output projectors, provides the effective implemented matrix A . In our case, A should match the objective matrix of the design, T . The agreement between the implemented and objective matrices can be quantified by defining a loss function of the form

$$\eta(\theta) = \sum_{i=0}^{N_{\text{in}}} \sum_{j=0}^{N_{\text{out}}} (T - A(\theta))^2 \quad (\text{A2})$$

The entire evaluation tree, from the component stiffness matrices to the loss function $\eta(\theta)$ can be automatically

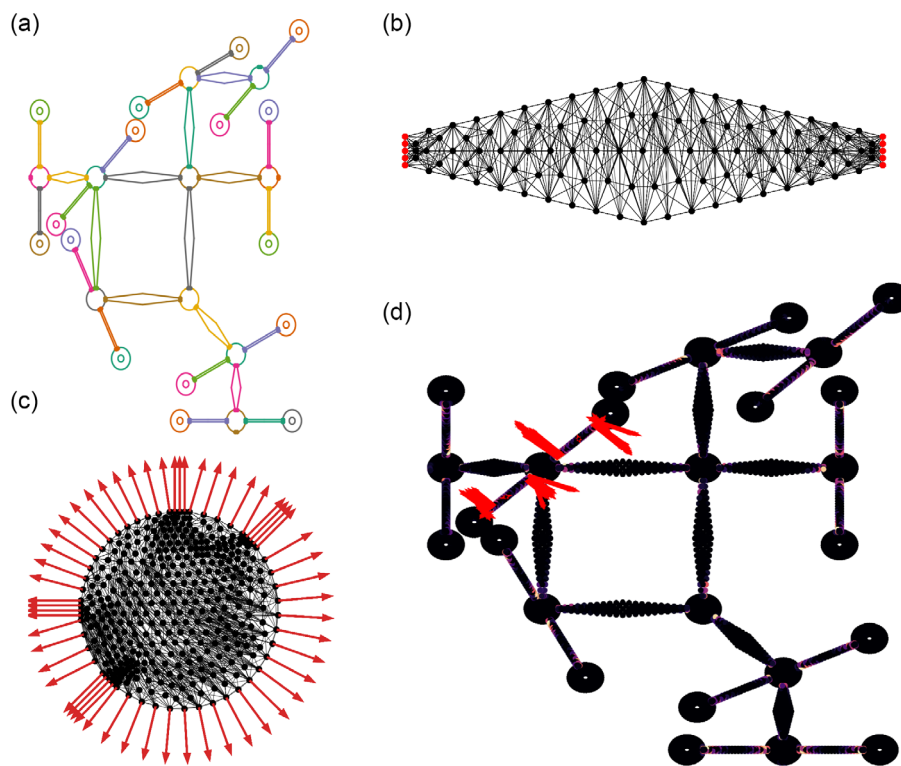


Figure A1. a) Division of the planar geometry into subcomponents (each component is plotted in a different color). The solid dots represent the contact points—that are automatically matched between each component to ensure compatibility. b) finite element meshing of a single component, red dots represent boundary degrees of freedom, which match between components, which are deterministically positioned and thus guaranteed to match between components. c) Gradient of the external mesh node coordinates with respect to a high-level parameter (in this case, the radius). d) Gradient of the misfit η with respect to the mesh node coordinates.

differentiated using JAX. Once the change in misfit with respect to the component stiffness matrices has been determined, the gradient with respect to the finite element mesh can be calculated in FEniCSx using the built-in automatic differentiation capabilities in the UFL library.^[60] This produces a gradient of the misfit with respect to the mesh coordinates (Figure A1d), indicating how the mesh should be deformed to improve the matrix-vector multiplication accuracy. This mesh gradient can be projected into a gradient of the boundary coordinates with respect to the high-level parameters (Figure A1d), producing the gradient of the misfit with respect to the high-level parameters. It should be noted that not all parameter updates are valid; some changes to the geometry will result in disconnected components. To prevent this, we project the gradient into the nullspace of the contact point distances J_{contact} with respect to the high-level parameters, and perform an additional boundary distance minimization optimization to preserve the well-connectedness of the metamaterial.

Appendix B: Experiments

We fabricate the design on a rubber sheet with a thickness of 6 mm, using water-jet cutting. The input displacement is prescribed using stepper motors and the output displacement is measured with a camera and processed using an optical flow algorithm.

We ran experiments at different excitation periods to find the influence of the loading rate on the hysteresis in the response of the system (Figure A2). For each period, the system is loaded and

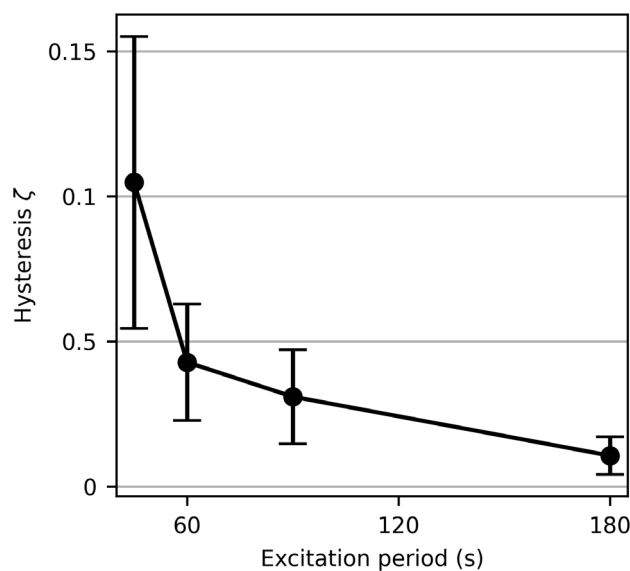


Figure A2. Evolution of the hysteresis as a function of the loading period.

unloaded three times. We quantify the hysteresis ζ as $\zeta = \frac{E_{\text{hys}}}{E_0}$, where E_{hys} is the energy present in the hysteresis region of the curve, and E_0 is the energy in the mean curve.^[61] We see the amount of hysteresis goes down, and hint at disappearing completely if we keep increasing the period. Similarly, the deviation of the hysteresis is also reduced. This confirms our hypothesis that hysteresis is due to viscoelasticity effect.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Theophile Louvet: conceptualization (lead); investigation (lead); visualization (lead); writing—original draft (lead); writing—review & editing (supporting). **Parisa Omidvar:** conceptualization (supporting); data curation (supporting); software (supporting); writing—review & editing (supporting). **Marc Serra-Garcia:** conceptualization (equal); investigation (equal); supervision (lead); writing—review & editing (equal).

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

floppy modes, mechanical metamaterials, physical computing, programmable matter, soft matter

Received: February 11, 2025
Published online:

- [1] M. Alexa, *ACM Trans. Graphics* **2002**, 21, 380.
- [2] N. Kambhatla, T. K. Leen, *Neural Comput.* **1997**, 9, 1493.
- [3] K. Hornik, M. Stinchcombe, H. White, *Neural Networks* **1989**, 2, 359.
- [4] F. Ashtiani, A. J. Geers, F. Aflatouni, *Nature* **2022**, 606, 501.
- [5] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, H. Qian, *Nature* **2020**, 577, 641.
- [6] H. Yasuda, P. R. Buskohl, A. Gillman, T. D. Murphey, S. Stepney, R. A. Vaia, J. R. Raney, *Nature* **2021**, 598, 39.
- [7] T. Dubček, D. Moreno-Garcia, T. Haag, P. Omidvar, H. R. Thomsen, T. S. Becker, L. Gebraad, C. Bärolocher, F. Andersson, S. D. Huber, D.-J. van Manen, L. G. Villanueva, J. O. Robertsson, M. Serra-Garcia, *Adv. Funct. Mater.* **2024**, 34, 2311877.
- [8] G. Dion, A. Tessier-Poirier, L. Chiasson-Poirier, J.-F. Morissette, G. Brassard, A. Haman, K. Turcot, J. Sylvestre, *Commun. Eng.* **2024**, 3, 48.
- [9] A. Rafsanjani, Y. Zhang, B. Liu, S. M. Rubinstein, K. Bertoldi, *Sci. Rob.* **2018**, 3, eaar7555.
- [10] W. Hu, G. Z. Lum, M. Mastrangeli, M. Sitti, *Nature* **2018**, 554, 81.
- [11] H. Yasuda, T. Tachi, M. Lee, J. Yang, *Nat. Commun.* **2017**, 8, 962.

- [12] A. H. Ghadimi, S. A. Fedorov, N. J. Engelsens, M. J. Beryhi, R. Schilling, D. J. Wilson, T. J. Kippenberg, *Science* **2018**, 360, 764.
- [13] A. Beccari, D. A. Visani, S. A. Fedorov, M. J. Beryhi, V. Boureau, N. J. Engelsens, T. J. Kippenberg, *Nat. Phys.* **2022**, 18, 436.
- [14] S. C. Masmanidis, R. B. Karabalin, I. D. Vlamincq, G. Borghs, M. R. Freeman, M. L. Roukes, *Science* **2007**, 317, 780.
- [15] M. Serra-Garcia, *Phys. Rev. E* **2019**, 100, 042202.
- [16] G. Bordiga, E. Medina, S. Jafarzadeh, C. Bösch, R. P. Adams, V. Tournat, K. Bertoldi, *Nat. Mater.* **2024**, 23, 1486.
- [17] Z. Hu, Z. Wei, K. Wang, Y. Chen, R. Zhu, G. Huang, G. Hu, *Nat. Commun.* **2023**, 14, 1266.
- [18] M. Czajkowski, D. Z. Rocklin, *Phys. Rev. Lett.* **2024**, 132, 068201.
- [19] J. C. Maxwell, *London, Edinburgh Dublin Philosop. Mag. J. Sci.* **1864**, 27, 294.
- [20] C. Calladine, *Int. J. Solids Struct.* **1978**, 14, 161.
- [21] C. L. Kane, T. C. Lubensky, *Nat. Phys.* **2014**, 10, 39.
- [22] C. Coulais, E. Teomy, K. de Reus, Y. Shokef, M. van Hecke, *Nature* **2016**, 535, 529.
- [23] A. Bossart, D. M. J. Dykstra, J. van der Laan, C. Coulais, *Proc. Natl. Acad. Sci.* **2021**, 118, e2018610118.
- [24] D. M. J. Dykstra, C. Coulais, *Phys. Rev. Applied* **2024**, 22, 064013.
- [25] R. van Mastrigt, M. Dijkstra, M. van Hecke, C. Coulais, *Phys. Rev. Lett.* **2022**, 129, 198003.
- [26] T. C. Lubensky, C. L. Kane, X. Mao, A. Souslov, K. Sun, *Rep. Prog. Phys.* **2015**, 78, 073901.
- [27] S. Guest, J. Hutchinson, *J. Mech. Phys. Solids* **2003**, 51, 383.
- [28] S. D. Papka, S. Kyriakides, *Int. J. Solids Struct.* **1998**, 35, 239.
- [29] S. Papka, S. Kyriakides, *Acta Mater.* **1998**, 46, 2765.
- [30] N. E. Glauvitz, R. A. Coutu, M. Kistler, I. R. Medvedev, D. T. Petkie, in *MEMS and Nanotechnology, Volume 5: Proc. of the 2013 Annual Conf. on Experimental and Applied Mechanics*, Springer, Lombard, IL, USA **2014**, June, pp. 73–79.
- [31] L. Wu, D. Pasini, *Phys. Rev. Appl.* **2023**, 19, L061001.
- [32] L. Wu, D. Pasini, *Nat. Commun.* **2024**, 15, 3087.
- [33] T. Chen, M. Pauly, P. M. Reis, *Nature* **2021**, 589, 386.
- [34] P. R. Kuppens, M. A. Bessa, J. L. Herder, J. B. Hopkins, *J. Mech. Rob.* **2021**, 13, 021010.
- [35] J. Salomon, Master of Science, School of Artificial Intelligence, Division of Informatics, University of Edinburgh **2001**.
- [36] B. Barazani, G. Dion, J.-F. Morissette, L. J. Sylvestre, *J. Microelectromech. Syst.* **2020**, 29, 338.
- [37] MEMS Optical Force Sensor Enhancement Via Compliant Mechanism, in *Int. Design Engineering Technical Conf. and Computers and Information in Engineering Conf., Vol. Volume 8: 31st Mechanisms and Robotics Conf., Parts A and B*, Las Vegas, NV, USA **2007**, September 2024, https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-pdf/IDETC-CIE2007/48094/127/4548420/127_1.pdf.
- [38] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, F. Wei (Preprint) arXiv:2402.17764, Submitted: 2024.
- [39] K. Bertoldi, V. Vitelli, J. Christensen, M. van Hecke, *Nat. Rev. Mater.* **2017**, 2, 17066.
- [40] M. Czajkowski, C. Coulais, M. van Hecke, D. Z. Rocklin, *Nat. Commun.* **2022**, 13, 211.
- [41] K. Bertoldi, P. M. Reis, S. Willshaw, T. Mullin, *Adv. Mater.* **2010**, 22, 361.
- [42] J. Paulose, B. G.-G. Chen, V. Vitelli, *Nat. Phys.* **2015**, 11, 153.
- [43] J. L. Silverberg, A. A. Evans, L. McLeod, R. C. Hayward, T. Hull, C. D. Santangelo, I. Cohen, *Science* **2014**, 345, 647.
- [44] M. Zhang, J. Yang, R. Zhu, *J. Appl. Mech.* **2021**, 88, 051009.
- [45] H. Fang, S.-C. A. Chu, Y. Xia, K.-W. Wang, *Adv. Mater.* **2018**, 30, 1706311.
- [46] C. Liu, X. Zhang, J. Chang, Y. Lyu, J. Zhao, S. Qiu, *Front. Mater.* **2024**, 11, <https://doi.org/10.3389/fmats.2024.1361408>.

- [47] R. H. Lee, E. A. B. Mulder, J. B. Hopkins, *Sci. Rob.* **2022**, 7, eabq7278.
- [48] M. Pishvar, R. L. Harne, *Adv. Sci.* **2020**, 7, 2001384.
- [49] R. Melik, E. Unal, N. K. Perkgoz, C. Puttlitz, H. V. Demir, *Appl. Phys. Lett.* **2009**, 95, 011106.
- [50] I. A. Baratta, J. P. Dean, J. S. Dokken, M. Habera, J. S. Hale, C. N. Richardson, M. E. Rognes, M. W. Scroggs, N. Sime, G. N. Wells (Preprint) Submitted: 2023.
- [51] A. S. Algamili, M. H. M. Khir, J. O. Dennis, A. Y. Ahmed, S. S. Alabsi, S. S. Ba Hashwan, M. M. Junaid, *Nanoscale Res. Lett.* **2021**, 16, 16.
- [52] C. Truesdell, W. Noll, C. Truesdell, W. Noll, *The Non-Linear Field Theories of Mechanics*, Springer, Heidelberg, Germany **2004**.
- [53] J. W. Rocks, N. Pashine, I. Bischofberger, C. P. Goodrich, A. J. Liu, S. R. Nagel, *Proc. Natl. Acad. Sci.* **2017**, 114, 2520.
- [54] O. R. Bilal, A. Foehr, C. Daraio, *Proc. Natl. Acad. Sci.* **2017**, 114, 4603.
- [55] S.-L. Lien, M. Shantz, V. Pratt, *ACM SIGGRAPH Comput. Graphics* **1987**, 21, 111.
- [56] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, G. N. Wells, *ACM Trans. Math. Software* **2014**, 40, 1.
- [57] M. W. Scroggs, I. A. Baratta, C. N. Richardson, G. N. Wells, *J. Open Source Software* **2022**, 7, 3982.
- [58] M. W. Scroggs, J. S. Dokken, C. N. Richardson, G. N. Wells, *ACM Trans. Math. Softw.* **2022**, 48, 1.
- [59] R. J. Guyan, *AIAA J.* **1965**, 3, 380.
- [60] D. A. Ham, L. Mitchell, A. Paganini, F. Wechsung, *Struct. Multidiscip. Optim.* **2019**, 60, 1813.
- [61] R. Penas, E. Balmes, A. Gaudin, *Mech. Syst. Signal Process.* **2022**, 170, 108793.