Validation of a nonlinear force method for large deformations in shape-morphing structures

Cosima du Pasquier¹ · Kristina Shea¹

Received: 31 August 2021 / Revised: 21 January 2022 / Accepted: 26 January 2022 / Published online: 12 February 2022
 © The Author(s) 2022

Abstract

Reducing energy and material consumption is a priority for the construction, aerospace, and automotive industries. Shape morphing addresses these concerns by broadening the band of functionality of a structure by adapting its shape to an external stimulus, such as pressure, or an internal stimulus, such as embedded actuation. This work outlines the development of an actuator placement optimization method for overdeterminate lattice structures with the objective of achieving predetermined large shape changes accurately. The deformation is modeled with both a linear and a nonlinear force method to determine their validity for large-shape change and their usefulness for the field of shape morphing. The linear and nonlinear methods are compared in four benchmark problems and two case studies relevant to the field of shape morphing. The nonlinear method is shown to achieve a level of accuracy 10^2 to 10^4 higher compared to FEM simulation, while using 23% fewer actuators and up to 77.3% less elongation of actuators, which makes it more favorable for shape-morphing applications. Two case studies for applications in aerospace and construction show that the nonlinear force method is better equipped for large shape change in overdeterminate meshed freeform target shapes and doubly curved surfaces with a high variable density. However, the nonlinear force method is less computationally efficient than the linear force method, as expected. A judicious choice of constraints can help reduce the run time.

Keywords Shape morphing · Nonlinear force method · Stochastic optimization · Sensitivity analysis

List of symbols

A	Equilibrium matrix
A_{red}	Reduced equilibrium matrix
C	Compatibility matrix
dim	Number of dimensions used
eff_i^u	Displacement efficiency for element i
f	Internal force vector
G	Element flexibility matrix
lb	Lower boundary in optimization
lb^{act}	Lower boundary reduced to active actuator positions
n^{act}	Number of actuators
n^{D}	Number of free degrees of freedom
n^{E}	Number of elements

n^{cDOFs}	Number of controlled degrees of freedom
n^{fDOFs}	Number of fixed degrees of freedom
n^{N}	Number of nodes
p	External load vector
Q	Probability of contribution to target deformation distribution vector
Q_i	Tanimoto index for the unitary elongation of element i
Q_j	Tanimoto index for the actuator combination at step j
S^F	Force sensitivity matrix
S^u	Displacement sensitivity matrix
S_j^{u cDOFs}	Displacement sensitivity matrix at controlled DOFs and at element j
T_k	Temperature at step k of the SA optimization
U	Left singular vectors
ub	Upper boundary in optimization
ub^{act}	Upper boundary reduced to active actuator positions
u_c	Controlled nodal displacement vector
u_{target}	Target nodal displacement vector
V	Singular values matrix

Responsible Editor: Makoto Ohsaki

✉ Kristina Shea
 kshea@ethz.ch

¹ Engineering, Design, and Computing Laboratory
 Department of Mechanical Engineering, ETH Zurich,
 Zurich, Switzerland

W	Right singular vectors
x	Actuator elongation vector
x^{act}	Actuator elongation vector reduced to active actuator positions
$\Delta\mathbf{L}$	Element elongation vector
ϵ	Element strain vector
λ	Penalty for the SA

1 Introduction

A global effort to reduce energy consumption and waste production is pushing engineers to rethink how to design and build structures. From the construction industry (Senatore et al. 2019), to aerospace (Jenett et al. 2017), automotive (Daynes and Weaver 2013), and robotics (Chen et al. 2018), many have converged to a similar conceptual solution: shape morphing. A shape-morphing structure is understood to have the capacity to broaden its band of functionality by adapting its structure to an external stimulus through a shape change. There are five main criteria to evaluate the design of a morphing structure: weight, load-carrying capacity, energy consumption, shape accuracy, and range and number of deformation modes (Wicks and Hutchinson 2004).

To tackle the criteria of weight and load-carrying capacity, much research has focused on the use of lattice structures. Their advantageous weight to stiffness ratio has motivated their implementation in airplane wings (Jenett et al. 2017), bridges (Senatore et al. 2019), and buildings (Reksowardojo et al. 2020). Pellegrino (1993) and Pellegrino and Calladine (1986) showed that lattice structures are also particularly well suited for controlled deformation, as their kinematic behavior can be entirely expressed through the equilibrium matrix of the morphing structure. Shape change can be induced by inserting the appropriate number of actuators at the appropriate locations in a lattice structure (dos Santos e Lucato and Evans 2006; dos Santos e Lucato et al. 2004; Reksowardojo et al. 2019; Tian et al. 2021). Several examples of the computational approach used to place actuators in a structure for shape morphing are given in Table 1.

They address some of the remaining criteria of shape morphing, but not entirely. Methods that assume small deformations (strain $\leq 1\%$) significantly impair their accuracy for large shape change, or reduce the range and number of deformation modes. The method presented by Reksowardojo et al. (2020) addresses all criteria the most completely but limits its application to five lattice cells and 26 variables (see Table 1). The use of a linear or nonlinear method impacts all the criteria of shape morphing but have not yet been compared directly.

The first contribution of this work is the strain-based nonlinear force method, which models how a strain input affects a structure as a whole in the optimization of the placement of linear actuators in a structure to achieve a target shape. The method is adapted from Reksowardojo et al. (2020), whose primary contribution is to counteract the effect of external loads through shape change. The second contribution is the inclusion of the optimization of the number and placement of actuators in both methods. The third contribution is the quantitative comparison of both methods to show that the proposed nonlinear force method is better suited for determining the necessary actuators and their movement for a structure to morph into several target shapes. Most work in the literature has focused on using linear FEM because it requires less computational effort and can be implemented directly in an optimization method. However, it limits the deformation range of a structure and loses accuracy in structures with a large number of states of self-stress.

Both methods presented are benchmarked for four main deformation modes and a similar number of cells and variables against the examples given in Table 1 with respect to the shape-morphing criteria: energy consumption (or number of actuators required), shape accuracy, and range and number of deformation modes achieved.

Finally, both methods are applied to two case studies ubiquitous in the field of shape morphing: a 2D NACA wing profile and a 3D morphing panel. These case studies demonstrate the contributions of this work by extending the field of application to overdeterminate meshed freeform shapes and doubly curved surfaces with a high variable density.

Table 1 Examples of shape-morphing lattice structures from the literature

	# nodes, # variables	Deformation modes and dimension	$\epsilon_{\max}(\%)$	Structural analysis method	Actuator # optimization
dos Santos e Lucato et al. (2004)	83, 96	Hinging and twisting, 2.5D	1.00	Linear combination of elongations (linear)	No
Wang et al. (2007)	83, 96	Twisting, 2.5D	1.00	Linear combination of elongations (linear)	No
Senatore et al. (2019)	12, 26	Concave up and concave down, 2D	0.34	IFM (linear)	Yes
Reksowardojo et al. (2020)	12, 26	Adapting to 5 load cases, 2D	67.2	NFM (nonlinear)	Yes
Tian et al. (2021)	8, 4	Bending, 2D	0.33	Kinematic (nonlinear)	No

The structure of this paper is as follows: first, the theory behind the linear and nonlinear force methods to determine the displacements resulting from the inclusion of actuators is explained in detail. The implementation of the actuator placement optimization using each method is detailed. Then, the benchmark problems, the case studies, and their evaluation parameters are introduced. The benchmark deformations are compared to FEM simulations conducted with Abaqus CAE. The linear and nonlinear force methods are evaluated first with respect to the benchmark problems and then with respect to the case studies. Finally, the results are discussed first in the context of this work and then in the greater context of shape morphing.

2 Background

2.1 Actuator placement optimization

The search for the appropriate number and location of actuators has been addressed using several numerical approaches. The main parameters of five examples are given in Table 1. dos Santos e Lucato et al. (2004), dos Santos Lucato et al. (2005), du Pasquier and Shea (2021) use a statically and kinematically determinate Kagome lattice structure to produce a hinging and twisting plate. Static and kinematic determinacy allow to model the effect of a set of actuator elongations as the linear combination of single-actuator elongations. This method is extremely quick once the effect of each actuator has been registered in a matrix, but it is limited to small deformations and determinate structures.

Senatore et al. (2019) use a combination of force and displacement sensitivity matrices and load path redirection to first determine the most efficient actuator locations, and then the internal force method to optimize their lengths. This method assumes small deformations: its cases studies achieve a maximum element elongation of 0.34%.

Reksowardojo et al. (2020) use simulated annealing (SA) and the Tanimoto index, a shape similarity criterion, to determine actuator locations. The actuator elongations are determined using force and displacement sensitivity matrices and optimized with a Newton–Raphson scheme. Force and displacement commands are determined using a nonlinear method, which allows for large element elongations, 67.2% at the most.

Wang et al. (2020) and Tian et al. (2021) establish control sequences based on a nonlinear kinematic method using constant curvature, segmentation method, and beam theory. They achieve an exact curvature change in a 2D morphing panel in three steps using two lattice cells by elongating the outer edges using linear slide modules, thus severely, restricting their achievable design space.

2.2 Modeling of strain-induced deformation using the FM

Most work in structural analysis focuses on the displacement based finite-element method (FEM) (Luo and Lu 2006). However, the works of Pellegrino and Calladine (1986), Pellegrino (1993), and Patnaik (1973) are the basis for formulations of the force method (FM) that have inherent advantages in modeling adaptive lattice structures. The main building blocks for the FM are the equilibrium equations and the compatibility equations. The equilibrium matrix $\mathbf{A}$ and its subspaces explicitly link the structural and kinematic properties of a lattice structure. Actuator elongations can be formulated as imposed strain distributions so internal forces and nodal displacements can be calculated directly. However, the FM requires a first empirical step of eliminating redundant lattice members and replacing them with external loads. Also, the equilibrium matrix is rectangular and potentially asymmetric for overdeterminate systems and is, thus, not sufficient to solve for internal forces directly. Pellegrino (1993) and Patnaik (1973) developed two formulations to avoid these issues.

The approach of Pellegrino (1993), building on earlier work by Pellegrino and Calladine (1986), is based on the singular value decomposition (SVD) of $\mathbf{A}$ and the interpretation of its subspaces. Starting with the equilibrium equations and the compatibility equations, he formulates a method to derive static and kinematic determinacy of a structure, s and m . Subdividing the SVD of $\mathbf{A}$ based on s and m , he derives the nodal displacements and element forces resulting from imposed strain distributions and external load. For a statically and kinematically determinate structure ($s = 0$ and $m = 0$), the solution is direct and exact. This formulation is only valid for small linear displacements and is extended by Luo and Lu (2006) to the nonlinear force method (NFM) by using a Newton–Raphson scheme. This formulation of the FM has recently been used by Reksowardojo et al. (2020) and Kim et al. (2016), focusing on keeping structures within serviceability limits, compensating for vibrations, or detecting damages.

The approach of Patnaik (1973), the internal force method (IFM), avoids the expensive computation time of SVD by compensating for the rank deficiency of $\mathbf{A}$. It builds a new matrix $\mathbf{S} = [\mathbf{A}^{\text{red}} \mid \mathbf{C} \cdot \mathbf{G}]^T$, where the equilibrium matrix is reduced to free degrees of freedom (DoF) $\mathbf{A}^{\text{red}}$ and is completed with the multiplication of the compatibility matrix $\mathbf{C}$ with the local flexibility matrix $\mathbf{G}$. The additional equations ensure that the desired element elongation is compatible with an imposed strain distribution.

The concept of sensitivity analysis in adaptive lattice structures is introduced in Senatore et al. (2019), Senatore and Reksowardojo (2020), and Yuan et al. (2016). The force and displacement actuation sensitivity matrices $\mathbf{S}^F$ and $\mathbf{S}^u$

contain the element forces and nodal displacements triggered by a unitary length change for each of the structure's elements in turn. Element forces and nodal displacements caused by a given element elongation vector $\mathbf{x}$ are given directly by $\mathbf{S}^F \cdot \mathbf{x}$ and $\mathbf{S}^u \cdot \mathbf{x}$ for small displacements. $\mathbf{S}^F$ and $\mathbf{S}^u$ can be calculated using both the FM and the IFM, but their calculation is a lot more efficient with the FM formulation. The importance of the sensitivity matrices in the optimization of actuator placement in lattice structures motivates the use of the FM formulation of Pellegrino (1993) and the NFM formulation of Luo and Lu (2006) in this work.

3 Method

This section describes the theory and implementation for the actuator placement optimization assuming first that the effect of elongating elements is linear and then that it is nonlinear. Both the linear and nonlinear methods are based on previous work by Pellegrino and Calladine (1986) and Pellegrino (1993) and their definition of kinematic and static determinacy.

3.1 The equilibrium matrix

The equilibrium matrix $\mathbf{A}$ of a lattice structure contains all information pertaining to its static and kinematic behavior. It relates external loads to internal loads directly. The SVD of this matrix separates the information into three matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{V}\mathbf{W}^T = [\mathbf{U}_r \ \mathbf{U}_m] \begin{bmatrix} \mathbf{V}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{W}_r \ \mathbf{W}_s]^T, \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{n^D \times n^E}$, $\mathbf{U} \in \mathbb{R}^{n^D \times n^D}$, $\mathbf{W} \in \mathbb{R}^{n^E \times n^E}$, and $\mathbf{V} \in \mathbb{R}^{n^E \times n^E}$. $[\mathbf{U}_r \ \mathbf{U}_m]$, $[\mathbf{W}_r \ \mathbf{W}_s]$, and $\mathbf{V}_r$ are the left singular vectors, right singular vectors, and singular values of matrix $\mathbf{A}$, respectively. $\mathbf{U}$ contains information on the kinematic behavior of $\mathbf{A}$, and $\mathbf{W}$ on the static behavior. $\mathbf{U}_r \in \mathbb{R}^{n^D \times (n^D - m)}$ contains the load components in equilibrium with the stress components in $\mathbf{W}_r \in \mathbb{R}^{n^E \times (n^E - s)}$ multiplied with the corresponding singular value in $\mathbf{V}$, so it is the basis of the row space of $\mathbf{A}$, $\mathcal{R}(\mathbf{A})$. Since the components of $\mathbf{U}_r$ are in equilibrium, they do not cause any displacements. $\mathbf{W}_s$ is the basis of the null space of $\mathbf{A}$, $\mathcal{N}(\mathbf{A})$. $\mathbf{U}_m \in \mathbb{R}^{n^D \times m}$ contains the load components that are not in equilibrium and, thus, cause displacements and form the basis of the inextensional mechanisms of the structure. Similarly, $\mathbf{W}_s \in \mathbb{R}^{n^E \times s}$ contains the s linearly independent states of self-stress of $\mathbf{A}$. $\mathbf{W}_s$ can also be interpreted as the incompatible strain basis.

A structure is statically determinate if it contains no states of self-stress, or $s = 0$, and is kinematically determinate if

it contains no mechanisms, or $m = 0$. The values of m and s are calculated directly from $\mathbf{A}$:

$$m = n^D - r, \quad (2)$$

and

$$s = n^E - r, \quad (3)$$

where n^E is the number of elements in the structure, or the number of rows of $\mathbf{A}$; $n^D = \dim \cdot n^N - n^k$ is the number of nodes multiplied by the dimension of the structure minus the number of kinematic constraints. r is the rank of $\mathbf{A}$. If $\mathbf{A}$ is kinematically determinate, $\mathbf{U}_m$ is empty. If $\mathbf{A}$ is statically determinate, $\mathbf{W}_s$ is empty.

3.2 The force method for imposed strain

The force method used in this work is based on the work of Pellegrino (1993), Senatore et al. (2019), Senatore and Rekswardojo (2020) and is referred to as FM. The FM is based on two sets of equations: the equilibrium conditions and the compatibility conditions. The force equilibrium conditions at a structure's nodes are

$$\mathbf{A}\mathbf{f} = \mathbf{p}, \quad (4)$$

where $\mathbf{f}$ is the internal load vector and $\mathbf{p}$ is the external load vector. The compatibility conditions as defined for the SVD method by Pellegrino (1993) ensure that compatible strain should vanish through orthogonality to the incompatible strain basis $\mathbf{W}_s$:

$$\mathbf{W}_s^T \boldsymbol{\epsilon} = \mathbf{0}, \quad (5)$$

or

$$\mathbf{W}_s^T (\mathbf{G}\mathbf{f} + \Delta\mathbf{L} \cdot \mathbf{L}^{-1}) = \mathbf{0}, \quad (6)$$

where $\mathbf{G} \in \mathbb{R}^{n^E \times n^E}$ is the element flexibility matrix and $\Delta\mathbf{L} \in \mathbb{R}^{n^E \times 1}$ is the element elongation vector.

Through the principle of Virtual Work, the compatibility conditions can be rewritten for statically and kinematically determinate systems as Pellegrino and Calladine (1986):

$$\mathbf{A}^T \mathbf{u} = \mathbf{G}\mathbf{f} + \Delta\mathbf{L} \cdot \mathbf{L}^{-1}, \quad (7)$$

so

$$\Delta\mathbf{L} = \mathbf{L} \cdot (\mathbf{A}^T \mathbf{u} - \mathbf{G}\mathbf{f}), \quad (8)$$

where $\mathbf{u}$ are nodal displacements. The FM, thus, allows the calculation of element elongations directly from the nodal displacements of a target deformation for kinematically and

statically determinate structures, assuming that the effects of the elongations are linear.

3.3 Sensitivity matrices

For indeterminate structures and assuming small deformations, element elongations can be calculated through the force and displacement sensitivity matrices $\mathbf{S}^F \in \mathbb{R}^{n^E \times n^E}$ and $\mathbf{S}^u \in \mathbb{R}^{n^D \times n^E}$. $\mathbf{S}^F$ relates the structure's flexibility and states of self-stress to determine reactive forces. $\mathbf{S}^u$ relates the structure's flexibility, force sensitivity matrix, and the pseudo-inverse equilibrium matrix to determine displacements (Senatore et al. (2019)). $\mathbf{S}_i^F$ and $\mathbf{S}_i^u$ contain the effect of the unitary length change of the i th element of $\Delta \mathbf{L}$, excluding the external load $\mathbf{p}$. Another advantage of the FM formulation is that the two matrices can be calculated directly rather than with a loop (Yuan et al. 2016; Senatore and Reksowardojo 2020):

$$\mathbf{S}^F = -\mathbf{W}_s(-\mathbf{W}_s^T \mathbf{G} \mathbf{W}_s)^{-1} \mathbf{W}_s^T. \quad (9)$$

$$\mathbf{S}^u = (\mathbf{A}^+)^T (\mathbf{G} \mathbf{S}^F + \mathbf{I}), \quad (10)$$

where $(\mathbf{A}^+)^T$ is the Moore–Penrose inverse of $\mathbf{A}$, $(\mathbf{A}^+)^T = \mathbf{W}_r \mathbf{V}_r^{-1} \mathbf{U}_r^T$ and $\mathbf{I}$ is an identity matrix of dimensions $n^E \times n^E$.

3.4 Optimization objectives

The objective of this work is to find the actuator placement and elongations in a lattice structure that best approximate a given target shape, assuming that all elements can be actuators. For indeterminate structures, this problem translates into the following optimization:

$$\begin{cases} \min \|\mathbf{S}^u \mathbf{x} - \mathbf{u}_{\text{target}}\|^2 \\ \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \end{cases}, \quad (11)$$

where $\mathbf{lb}$ and $\mathbf{ub}$ are the maximum contraction and elongation for each element, and $\mathbf{x}$ is a vector of length $n^E \times 1$. x_i is the value of the elongation of the i th element, so $x_i = 0$ if the element is not an actuator. The quality of a solution $\mathbf{x}$ is evaluated with the Tanimoto index, a shape similarity criterion used in machine learning (Tanimoto 1958; Reksowardojo et al. 2020):

$$Q_i = \frac{(\mathbf{u}_c)^T \mathbf{u}_{\text{target}}}{(\mathbf{u}_c)^T \mathbf{u}_c + (\mathbf{u}_{\text{target}})^T \mathbf{u}_{\text{target}} - (\mathbf{u}_c)^T \mathbf{u}_{\text{target}}}, \quad (12)$$

$0 \leq Q_i \leq 1$

where Q_i is the Tanimoto index for the unitary elongation of element i , $\mathbf{u}_c$ is the controlled deformation, and $\mathbf{u}_{\text{target}}$ is the target deformation. The closer the Q is to one, the better

the shape approximation. In general, the lowest objective function value corresponds to the highest Tanimoto index value but not always since the latter is both a shape and a node similarity criterion, as opposed to the objective function which is only a node position similarity criterion.

3.5 Linear method

The linear method assumes small deformations of the elements and large nodal displacements. Elements are allowed to elongate and contract up to 100% of their initial length to compare the validity of the model for large-scale shape morphing to the nonlinear method. Figure 1a shows the optimization framework used for the linear method: after calculating the sensitivity matrices as explained above, the actuator placement is optimized using element displacement efficacy assuming that the number of actuators is predetermined, and then the actuator lengths are optimized to achieve the desired target shape $\mathbf{u}_{\text{target}}$.

3.5.1 Actuator placement

The placement of the actuators within the structure is determined using the concept of placement efficacy from Senatore et al. (2019). The efficacy is a measure of the contribution of each individual element to the desired displacement. The closer the efficacy of an element is to 1, the more it contributes to the desired displacement. It is evaluated as follows:

$$\text{eff}_i^u = \frac{\sum_{j=1}^{n^{\text{cDOFs}}} (\mathbf{S}_j^u |^{\text{cDOFs}} \cdot \Delta \mathbf{L}_i) \circ \frac{1}{\mathbf{u}_{\text{target}}^{\text{cDOFs}}}}{n^{\text{cDOFs}}} \quad (13)$$

$$0 \leq \text{eff}_i^u \leq 1$$

$\mathbf{S}_j^u |^{\text{cDOFs}}$ is built by extracting the rows from $\mathbf{S}^u$ corresponding to the j th controlled DOF. $\Delta \mathbf{L}_i$ is the unitary elongation vector of the i th element. $\mathbf{u}_{\text{target}}^{\text{cDOFs}}$ is the target vector reduced to the controlled DOFs.

The necessary number of actuators is determined as the number of actuators needed to solve Eq. 14 exactly. For the linear method, the actuator positions are chosen as the n^{act} elements with the highest efficacy.

3.5.2 Actuator length optimization

Assuming small displacements and no external loads, Senatore and Reksowardojo (2020) show that element elongations to achieve a target shape can be directly calculated with

$$\mathbf{x} = \mathbf{W}_r (\mathbf{S}^u \mathbf{W}_r)^+ \mathbf{u}_{\text{target}} \quad (14)$$

where the $+$ indicates the Moore–Penrose inverse. However, the quality of the results depends on how close $\mathbf{W}_r$ and $\mathbf{S}^u$ are to singularity. If the norm of the difference between $\mathbf{u}_{\text{target}}$

Fig. 1 Optimization method for the linear method (left) and nonlinear method (right); the NLE solver is described in the Replication of results section

and u_0 , the nodal displacement caused by x , is smaller than the convergence criterion e_{Lin} , then x is already the optimal

solution, so the number of objective function calls is 0 and the Tanimoto index $Q = 1$. If the norm of the difference

between $\mathbf{u}_{\text{target}}$ and $\mathbf{u}_0$ is larger than e_{Lin} , the element elongations are optimized using Eq. 11 reduced to the controlled DOFs and the known actuator positions:

$$\begin{cases} \min \| \mathbf{S}_{\text{act}}^{\text{u|cDOFs}} \mathbf{x}^{\text{act}} - \mathbf{u}_{\text{target}}^{\text{cDOFs}} \|^2 \\ \mathbf{l}^{\text{act}} \leq \mathbf{x}^{\text{act}} \leq \mathbf{u}^{\text{act}} \end{cases} \quad (15)$$

The starting point $\mathbf{x}_0$ is previously calculated with Eq. 14. The quality of $\mathbf{x}_{\text{opt}}$ is determined by the resulting Tanimoto index.

3.6 Nonlinear method

The nonlinear method presented in this work can be used for large deformations ($\epsilon \gg 1\%$). The nodal displacements caused by element elongations are computed using a nonlinear solver based on the work of Pellegrino (1990), Luo and Lu (2006). ϕ maps the initial structure and given elongations to the resulting nodal displacements, element forces, and Tanimoto index Q :

$$\phi : (\mathbf{x}, \mathbf{D}, \mathbf{E}) \mapsto (\mathbf{u}_c, \mathbf{f}, Q) \quad (16)$$

where $\mathbf{D}$ and $\mathbf{E}$ are the cross-section vector and elastic modulus vector for all elements. Elements can elongate and contract up to 100% of their initial length to assess applicability of the method to large-shape changes.

Figure 1b shows the optimization method used for the nonlinear method introduced in this work. A shape-morphing lattice structure should contain as few actuators as possible to not compromise its structural soundness. Reksowardojo et al. (2020) define that

$$\begin{cases} \mathbf{S}^{\text{u}} \mathbf{x} = \mathbf{u}_{\text{target}} & \text{for } n^{\text{act}} = n^{\text{E}} \\ \mathbf{S}^{\text{u}} \mathbf{x} \simeq \mathbf{u}_{\text{target}} & \text{for } n^{\text{cDOFs}} + s < n^{\text{act}} < n^{\text{E}} \\ \mathbf{S}^{\text{u}} \mathbf{x} \sim \mathbf{u}_{\text{target}} & \text{for } n^{\text{act}} < n^{\text{cDOFs}} + s \end{cases} \quad (17)$$

The first optimization shown in black in Figure 1b starts at $n_0^{\text{act}} = n^{\text{cDOFs}} + s$. It is implemented to calculate the necessary number of actuators, first using simulated annealing (SA) to determine the actuator positions given unitary elongations, and using IPOPT to optimize the actuator elongations given the positions. SA is chosen due to the combinatorial nature of the actuator placement problem. In the first step ($i = 1$), the placement and lengths are optimized r times. In the second step ($i = 2$), the best result from $i = 1$ Q_{best} is saved as Q_{old} , and actuators are added so $n^{\text{act}} = n^{\text{act}} + \frac{n^{\text{E}}}{100}$.

The placement and lengths are optimized again for the new n^{act} . Convergence is achieved if the new best solution Q_{best} is worse than Q_{old} , or $\frac{Q_{\text{best}}}{Q_{\text{old}}} < 1$. The first optimization method is shown in Algorithm 1.

Algorithm 1 Actuator number optimization

```

1: Initialize  $n^{\text{act}}$ ,  $\mathbf{x}_0$ ,  $Q_0$ ,  $Q_{\text{best}}$ ,  $Q_{\text{old}}$ ,  $ratio$ 
2: while  $ratio < 1$  do
3:   for  $k = 1 : r$  do
4:     Evaluate  $\mathbf{x}_k$  with SA
5:     Optimize actuator lengths with Alg. 2
6:     Save  $\mathbf{x}_{\text{best}}$  and  $Q_{\text{best}}$ 
7:      $k = k + 1$ 
8:   end for
9:    $ratio = \frac{Q_{\text{best}}}{Q_{\text{old}}}$ 
10:   $n^{\text{act}} = n^{\text{act}} + \text{round}(\frac{n^{\text{E}}}{100})$ 
11:   $Q_{\text{old}} = Q_{\text{best}}$ 
12: end while

```

The second optimization method calculates the best actuator placement and elongations, given $n_{\text{opt}}^{\text{act}}$. It loops through the same SA and IPOPT algorithms as the first optimization and converges either when the Tanimoto index Q_{best} exceeds the convergence criterion Q_{conv} or when the maximum number of iterations is achieved. For each step, the Tanimoto index, the value of the objective function, and the iteration number are stored in a matrix if they exceed the previous best result.

This work modifies the method reported by Reksowardojo et al. (2020) by optimizing an imposed internal strain ϵ to achieve a target shape deformation rather than minimizing any deformations caused by an external load $\mathbf{p}$. It further enhances the method by introducing loops within both optimizations. This meta-heuristic approach enables both optimization methods to avoid local optima in this nonlinear problem.

The linear method is used to determine the nodal displacements given the element elongations $\mathbf{x}$ within the objective function of the IPOPT step, given in Eq. 15, for computational efficiency. The results are then adjusted using a nonlinear method before calculating the Tanimoto index and checking for convergence. The details of the implementation of the actuator placement using SA and the elongation optimization using IPOPT are given below.

3.6.1 Actuator placement using SA

The nonlinear nature of the system nullifies the assumption that the n^{act} most efficient elements constitute the best actuator placement solution, since it takes into account only the efficacy of a single actuator at a time rather than actuator combinations. Instead, the efficacy is evaluated using the Tanimoto index for the unitary elongation of each element. The number of possible actuator combinations increases exponentially with the number of elements, so an exhaustive search is impractical. Since the Tanimoto index still only describes the efficacy of single elements but not their combination, the vector $\mathbf{Q}$ containing the Tanimoto indices for each element elongation is assumed to represent a probability distribution for elements to contribute to $\mathbf{u}_{\text{target}}$.

$$\mathbf{Q} = \begin{pmatrix} Q_1 \\ \vdots \\ Q_n \end{pmatrix} \quad (18)$$

The distribution and its pseudo-inverse, $\mathbf{Q}_{\text{pinv}}$, are used in a SA stochastic search to determine the best actuator placement given an initial n^{act} and assuming unitary elongations. The problem size is reduced by setting the probability distributions to zero for elements that link two pinned nodes, as they cannot elongate. The method, which mimics the cooling schedule of molten metals through annealing, was first formulated by Kirkpatrick et al. (1983). In the beginning, as the temperature T is high, the algorithm is likely to accept inferior neighboring solutions and avoids local optima. As T reduces, the algorithm is likely to reject inferior neighboring solutions, and the search is increasingly focused towards an optimum.

The goal is to maximize the Tanimoto index for a combination of actuators. The objective function is

$$\min(1 - Q_j + \lambda), \quad (19)$$

where λ is a penalty. The starting point $\mathbf{x}_0$ is chosen as the n^{act} elements with the highest Q_j . Neighboring solutions $\mathbf{x}_j$ are generated by drawing without replacement from the $n^{\text{E}} - n^{\text{act}}$ remaining elements using the probability distribution $\mathbf{Q}$ and replacing certain actuator elements using the pseudo-inverse distribution $\mathbf{Q}_{\text{pinv}}$. In each iteration, the number of elements replaced in $\mathbf{x}_j$ is chosen at random, and the Tanimoto index of the combined unitary elongation of the new actuator set is evaluated. The acceptance probability A_Q for a new solution is dependent on T :

$$A_Q(\mathbf{x}_j, \mathbf{x}_{j+1}) = \exp\left(-\frac{Q_j - Q_{j+1}}{T}\right) \quad (20)$$

λ is draw at random from a uniform distribution $\mathcal{U}(0, 1)$. The acceptance probability A_λ for a new penalty is also dependent on T :

$$A_\lambda(\mathbf{x}_j, \mathbf{x}_{j+1}) = \begin{cases} \exp(-\frac{\lambda_j - \lambda_{j+1}}{T}) & \text{if } \lambda_j \geq \lambda_{j+1} \\ 1 & \text{if } \lambda_j < \lambda_{j+1} \end{cases} \quad (21)$$

As T sinks, so does the acceptance probability of a lower penalty value, and so the search is increasingly focused.

T is updated n_{temp} times, and the search is repeated n^{rep} times for each temperature. T_k , the temperature at step k , is defined as follows:

$$T_k = -\frac{1}{\log(1 - \frac{k}{n^{\text{temp}}})}. \quad (22)$$

The best solution found so far is stored. Convergence is achieved when the solution no longer improves after several iterations. The resulting solution is the actuator placement that scored the highest Tanimoto index. Further implementation details on this method can be found in Wah and Wang (1999) and Reksowardojo et al. (2020) and in the Replication of results section of this work.

3.6.2 Actuator elongation optimization using IPOPT

Once the actuator placement has been determined assuming unitary element elongations using SA, the element elongations are optimized using IPOPT. Since the objective function optimization given in Eq. 15 is estimated using the linear model for computational efficiency, the process is iterative. First, Eq. 15 is solved using IPOPT. Then, the objective function value and Tanimoto index are calculated for the resulting element elongation $\mathbf{x}_j$ using the nonlinear model. Convergence is achieved either if the objective function value is smaller than e_{NL} or if the Tanimoto index is lower than in the previous step. If convergence is not achieved, the initial actuator elongations of the next step are set to $\mathbf{x}_j$ and the coordinates, equilibrium matrix, and sensitivity matrices are updated to reflect the element elongations of $\mathbf{x}_j$. The boundary conditions are also updated so the total element elongation boundary remains the same. The element elongation optimization is shown in detail in Algorithm 2.

Table 2 Benchmark nodal displacements

	Target deformation
Linear	$y = 0.111 \cdot x$
Concave up	$y = 10^{-4} \cdot (5.21x^2 - 24.5x + 252)$
Concave down	$y = -10^{-4} \cdot (5.21x^2 - 24.5x + 252)$
Sine	$y = 3.15 \cdot \sin(\frac{45}{\pi}x)$

Table 3 Benchmark method parameters for the optimization methods and the simulation

Element length:	$L = 10 \text{ mm}$
Cross-section area:	$A = 3 \text{ mm}^2$
Young's modulus:	$E = 200 \text{ GPa}$
Density:	$\rho = 1.4 \times 10^{-6} \text{ t} \cdot \text{mm}^{-3}$

Algorithm 2 Actuator elongation optimization

```

1: Initialize  $\mathbf{x}_0$ ,  $\mathbf{coord}_0$ ,  $\mathbf{A}_0$ ,  $\mathbf{S}_0^u$ ,  $\mathbf{ub}$ ,  $\mathbf{lb}$ 
2:  $j = 0$ 
3: while true do
4:   Solve Eq. 15 for  $\mathbf{x}_j$ 
5:   Find resulting nodal displacements  $\mathbf{u}_j$  and Tanimoto
     index  $Q_j$  with NLE solver
6:   if  $\|\mathbf{u}_j - \mathbf{u}_{\text{target}}\|^2 < e_{NL}$  then
7:     Break
8:   else if  $Q_j < Q_{j-1}$  then
9:     Break
10:  end if
11:  Update  $\mathbf{x}_{j+1}$ ,  $\mathbf{coord}_{j+1}$ ,  $\mathbf{A}_{j+1}$ ,  $\mathbf{S}_{j+1}^u$ 
12:  Update dynamic  $\mathbf{ub}$  and  $\mathbf{lb}$ 
13:   $j = j + 1$ 
14: end while

```

3.7 Benchmark problems

Both methods are tested on four 2D target deformations given in Table 2 that are benchmarked against results using Abaqus CAE 6.14: 1. linear deformation, 2. concave up, 3. concave down, and 4. sine, the transition between the two. The method parameters used in the optimization method as well as the simulation are given in Table 3. All problems have static determinacy $s = 1$ and kinematic determinacy $m = 0$. The quality of both methods is assessed using the

Tanimoto index of the optimized results, the average error of their nodal displacements with respect to the simulation, the number of actuators needed for each target deformation, the absolute largest strain, the number of iterations, and the number of objective function evaluations.

3.8 Case studies

The methods are tested on two case studies: in 3D, for complex doubly curved deformations of a tetragonal lattice structure plate, and in 2D, on a NACA 2640 airfoil, using an Abaqus CAE generated tetragonal mesh.

3.9 Implementation details

The work presented above is implemented in MATLAB R2020a and verified with Abaqus CAE 6.14-1 using a static general step and NLGEOM. This problem is solved using an interior point optimization (IPOPT) made available by (Wachter and Biegler, 2004) with default settings and maximum iteration number set to 250. The optimization is run on the ETH Euler cluster using ten cores with 64-core AMD EPYC 7H12 processors.

4 Results

This section first details the results for the benchmark problems of both methods and then the results from the case studies.

For the linear method, the convergence criterion e_{Lin} is set to 10^{-3} and the maximum number of iterations is 100.

For the nonlinear method, the convergence criterion Q_{conv} is set to 0.999 and the maximum number of iterations is 100. Preliminary studies of the stored results matrix showed that setting the maximum number of iterations higher did not lead to a better result for the structures analyzed in this work. The number of loops r in the actuator number optimization is set to five. Preliminary studies showed that increasing the number of loops to ten or twenty only lead to better results in less than 10% of the cases; the number of loops is, thus, set to five for computational efficiency.

Preliminary studies where results were calculated exhaustively for $n^{\text{cDOFs}} + s \leq n^{\text{act}} \leq n^{\text{E}}$ showed that, for problems with values of $n^{\text{E}} > 100$, the ratio $\frac{Q_{\text{best}}}{\frac{Q_{\text{opt}}}{n^{\text{E}}}}$ increased every $\frac{n^{\text{E}}}{100}$ steps. The size of a problem with $\frac{Q_{\text{opt}}}{n^{\text{E}}}$ also motivated the reduction of the number of steps. The convergence criterion for the actuator length optimization e_{NL} is set to 10^{-6} .

Fig. 2 Benchmark results for both methods (**a** and **b**) and four deformations (1 to 4) with static determinacy $s = 1$; for each deformation type: upper left—actuator placement, blue is elongation, red is contraction; lower left—deformed structure from the method, the red

circles indicate the target positions for moving nodes, the black circles indicate fixed nodes; right: axial stress for the deformed structure from the simulation; the stress scales on the right of each column apply for all simulation results

Table 4 Benchmark results

Target deformation	# Elements	# Nodes	Tanimoto Index	Error	# Actuators	$ \epsilon_{\max} $ (%)	# Objective function evaluations	Run time
Linear	39	21	LE 1	1.41 mm	27	91.0	0	35 s
			NLE 0.998	2.39×10^{-3} mm	16	20.9	10,835	131 s
Concave up	39	21	LE 1	1.30 mm	24	43.7	475	24 s
			NLE 0.998	1.43×10^{-4} mm	17	16.7	10,167	93 s
Concave down	39	21	LE 1	1.47 mm	26	43.7	532	48 s
			NLE 0.998	2.98×10^{-3} mm	17	22.3	12,500	89 s
Sine	39	21	LE 1	2.75 mm	16	27.1	0	20 s
			NLE 0.999	3.56×10^{-2} mm	17	26.6	10,598	182 s

4.1 Benchmark

The benchmark results for the two methods are detailed in Fig. 2 and Table 4. Comparing the LE and the NLE method, the Tanimoto indices of all results vary by less than 0.2% from the target shape, but the error with respect to the simulation are 10^2 to 10^4 orders of magnitude larger for the LE. This is also visible when comparing the left and the right columns of Fig. 2a and b, especially in the fourth benchmark problem. The LE method does not take into account internal stresses caused by other element elongations, which in fact

cause the structure to warp significantly. Further tests show that the LE method can only achieve similar levels of accuracy if limited to less than 10% element elongation, which in turns limits the achievable macroscopic deformations and drastically increases the number of actuators.

The NLE method needs on average 23% less actuators than the LE method. Its actuators elongate between 1.8% and 77.3% less. The actuator placements are analogous, but better calibrated for the NLE method. Comparing for example the sine deformation in Fig. 2.4, the contracting and elongating elements are placed similarly for the

Fig. 3 Convergence results of the benchmark problems for the actuator number optimization; the best solution determined with the convergence criteria is indicated in red; the values for one actuator

have been cut off to clarify the convergence process, their average is $fval_{linear,1} = 33.6$, $fval_{concave\ up,1} = 10.9$, $fval_{concave\ down,1} = 8.4$, and $fval_{sine,1} = 26.3$

concave down and up portions of the structure, but the NLE method makes more efficient use of the placement, using two rather than eight actuators in both portions, which can lead to a more structurally sound and energy efficient structure. The LE method has the advantage of converging faster, needing roughly 200 times less objective function evaluations and on average 4.64 times less run time. The run time for the LE is under one minute whereas the NLE method needs up to three minutes. The average run time per objective function evaluation is 7×10^{-2} s for the LE method and 2.3×10^{-2} s for the NLE method, so the NLE method actually iterates faster.

The convergence diagrams for the nonlinear actuator number optimization of the benchmark problems are shown in Fig. 3, where the exact results given in Table 4 are indicated in red. The aim of this work is to obtain the best compromise of shape similarity to number of actuators. As shown in the figure, adding actuators beyond the convergence criteria given in Sect. 3.6 and at the beginning of this section offer little to no improvement.

4.2 Case studies

4.2.1 3D panel

The results for the 3D panel for the two methods are detailed in Fig. 4 and in Table 5.

The NLE doubly curved deformation requires nearly four times as many actuators as the LE method. The actuator pattern in Fig. 4a shows that the LE method only deforms locally to accommodate the target shape, whereas the NLE method presents a wider actuation response. The pattern of contracting and elongating elements is, however, apparent in both cases.

The corner lift deformation differs from the doubly curved deformation significantly. For the first and only time, the maximal strain of the actuators for the LE method is significantly lower than for the NLE method. However, it requires nearly three times as many actuators overall, so the

Fig. 4 3D panel deformation for the a. LE and b. NLE methods; the red circles indicate the target positions for moving nodes, the black circles indicate pinned nodes; the undeformed structures in the upper

left corner of each deformation show the actuator placement; the actuators in blue are elongating, the actuators in red are contracting, and the elements in gray are fixed

Table 5 3D panel deformation results

Target deformation	# Nodes	# Elements	Overdeterminacy	Tanimoto Index	# Actuators	$ \epsilon_{\max} $ (%)	# Objective function evaluations	Run time
Doubly curved	346	1,535	$s = 509$	LE 1	159	81.6	0	7594 s
			$m = 0$	NLE 0.999	606	98.5	322	27,880 s
Corner lift	346	1,535	$s = 504$	LE 1	1505	9.8	92	5108 s
			$m = 0$	NLE 1	532	53.5	332	46,465 s

lower strain is compensated by a full body response. The NLE method requires significantly more objective function calls and on average 6.38 times more run time, but its

resulting Tanimoto index is perfect. Setting lower convergence limits could cut down the number of function evaluations, although it would affect the accuracy. The highest run

Fig. 5 NACA airfoil “tail up” and “tail down” deformation for the a. LE and b. NLE methods; the red circles indicate the target positions for moving nodes, the black circles indicate pinned nodes; the unde-

formed structures in the upper left corner of each deformation show the actuator placement; the actuators in blue are elongating, the actuators in red are contracting, and the elements in gray are fixed

Table 6 NACA Results

Target deformation	# Nodes	# Elements	Overdeterminacy	Tanimoto Index	# Actuators	$ \epsilon_{\max} $ (%)	# Objective function evaluations	Run time
Tail up	203	520	$s = 207$	LE	302	67.8	841	317 s
			$m = 0$	NLE	428	36.0	1820	70,923 s
Tail down	203	520	$s = 207$	LE	344	92.4	843	284 s
			$m = 0$	NLE	363	41.8	1771	42,726 s

time of the NLE optimization is 46,465 s, or roughly 12 h, compared to 2.1 h for the LE optimization. The average run time per objective function evaluation is 55.5 s for the LE method and 113 s for the NLE method.

4.2.2 NACA airfoil

The results of the meshed NACA airfoil for the two methods are detailed in Fig. 5 and in Table 6. This case study has five times more nodes than the benchmark problems

and twenty-five times more elements. The increase in the elements to nodes ratio considerably increases the numbers of states of self-stress in the structure which affects the results.

As opposed to the benchmark problems, the LE method achieves a Tanimoto index of only 0.793, compared to a nearly perfect 0.993 for the NLE method. This is clearly visible in Fig. 5a for both deformations, where the method results overshoot the desired deformation. However, the increase in shape accuracy comes with a cost, since the NLE method needs 41.7% and 5.5% more actuators for

“tail up” and “tail down,” respectively. The additional actuators are offset by a lower maximal strain (48.5% and 62.0% less, respectively). The ratio of objective function evaluations for the NLE and LE methods is roughly 10^2 s, similar to the benchmark. The average run time per objective function evaluation is 3.60×10^{-1} s for the LE method and 31.6 s for the NLE method.

5 Discussion

This section first reviews the results presented and shows the superiority of the NLE method quantitatively. It then compares qualitatively to examples of shape-morphing lattice structures from literature and discusses the relevance of this work in the broader field of shape morphing.

The results from the benchmark problems clearly show the advantages of the NLE method. With respect to the benchmark problems, the solutions for the NLE method are more accurate and need on average less actuators, which is an advantage for a structure’s soundness. The doubly curved 3D case study is the only instance where the NLE method needs more actuators at a slightly higher maximum strain. Using the actuator location combinations efficacy calculated with the SA rather than single-actuator efficacy combined with the NLE method consistently achieves better results with less actuators and less elongation. The only benchmark problem where the NLE method needs more actuators than the LE method is the sine deformation, which is also where the LE method deviates the most from simulation. However, the accuracy comes at the cost of computational efficiency, with calculation times roughly four times longer than the LE method. The LE method does not take into account internal stresses caused by other element elongations, which in fact cause the structure to warp significantly. Further tests show that the LE method can only achieve similar levels of accuracy if limited to less than 10% element elongation, which in turn limits the achievable macroscopic deformations and drastically increases the number of actuators.

These differences are made more obvious with an increase in number of variables and structural complexity. Adding elements, and thus, states of self-stress, impacts the computational effort: multiplying the number of variables by 25 results in a calculation time roughly 100 times longer.

The effect on the run time per iteration is different for the LE and NLE optimizations. In the NLE optimization, increasing the degree of overdeterminacy from 1 to 207 results in a multiplication of the run time per iteration by a factor of 1,374, but a further increase to $s = 509$ in the 3D panel case study only increases the run time per iteration by a factor of 3.60. In the LE optimization, the same increases correspond to a factor of 5.14 for $s = 1$ to $s = 207$ and a factor of 154 for $s = 207$ to $s = 509$. So, although the LE method is generally more efficient, it loses its advantage with an increasing degree of overdeterminacy. The NLE methods is thus more appropriate for very large complex structures.

When comparing the results from the two case studies, the 3D panel deformation requires up to 24 times less calculation effort than the NACA profile for the LE method, whereas it requires 2.5 times more calculation effort for the NLE method. The LE method, where the impact of the states of self-stress is less significant, is impacted exponentially by the number of elements and the deformation complexity. However, the NLE method needs more objective function evaluations in the NACA case where there are less variables, but a higher variables-to-states of self-stress ratio. In the NACA case study, the constraints are often on both ends of elements, which creates inherent self-stress, and further increases the number of objective function evaluations needed to solve for the target deformation, which shows that the position of constraints impacts the efficiency of the methods greatly.

However, the case studies differ from the benchmark in the number of actuators needed for both methods. Since the LE method does not take into account the internal stresses caused by element elongations, it underestimates how many actuators it will need to achieve the target deformation, which in turn affects the Tanimoto index. This problem is particularly apparent in the NACA case study, where several elements are self-stressed by the constraints. The NLE method deforms incrementally and includes the induced states of self-stress, which increases the number of necessary actuators significantly while conserving the accuracy of the model. On the other hand, the LE method overestimates the number of actuators needed for the panel “corner lift” case by a factor of four, and delivers a structure that would be significantly harder to build than the NLE method. The case studies also differ from the benchmark in that they need 10 to 100 times fewer objective function evaluations. This

is explained by two factors. First, the heuristic for Optimization 1 takes steps of $n^{\text{act}} = \frac{n^{\text{E}}}{100}$, which is 1 for the benchmark, 15 for the 3D panel, and 5 for the NACA profile. This means that, for example, the Corner lift deformation of the 3D panel only iterates twice before n^{act} converges. Second, the ratio of deformation to structure size is much smaller in the case studies, so they require a smaller actuator density than the benchmark to achieve the target shapes.

The NLE method supplemented with the stochastic actuator number optimization (optimization 1) and length optimization (optimization 2) addresses the five main criteria of shape morphing better than the LE method: weight, load-carrying capacity, energy consumption, shape accuracy, and range and number of deformation mode. As in comparable work given in Table 1, weight and load-carrying capacity are addressed in the use of lattice structures. By using actuators more efficiently both in number and maximal strain, the NLE method shows better load-carrying capacities and better energy consumption. The shape accuracy and deformation range are similar for both in simple problems with low levels of overdeterminacy, but the NLE method is better equipped for a range of complex target shapes and high levels of overdeterminacy.

Comparing back to the examples given in the Introduction, the NLE method addresses all criteria where many addressed only one or two. Comparing further to Reksowardojo et al. (2020), on whose work some of the method is based, the NLE method is shown to achieve a larger shape range and to work with a more complex design space.

The extension of the design space and target shape range to highly overdeterminate freeform meshes and variable dense 3D problems of the case studies show the promise of the optimization method proposed in this work for the field of shape morphing. The nonlinear method can be applied to problems with any arbitrary lattice layout, variable number, and target shape. However, an increase in a problem's complexity is mirrored by an increase in its computational effort. Further work should focus on finding an acceptable compromise between accuracy and computational effort with certain simplifications of the NLE method.

6 Conclusion

This work describes a linear and a nonlinear method to estimate the shape-morphing capacities of an overdeterminate lattice structure by optimizing the number and position of linear actuators within it. It compares the methods quantitatively in four benchmark problems and two case studies relevant to shape morphing. It shows that the nonlinear method is the most appropriate for large deformations of complex structures and leads to results of up to 10^4 higher accuracy than the linear method. By implementing a new

stochastic optimization approach to minimize the number of actuators in the nonlinear method, it delivers accurate results that are more structurally sound than the results from the linear method. The case studies validate the method for highly overdeterminate, meshed freeform target shapes and doubly curved surfaces with a high number of variables, and indicate the importance of the choice of constraints in these systems. The nonlinear method can, thus, be applied to problem sizes with a number of elements on the order of 10^3 and even larger, target shapes with varying curvature, and high degrees of overdeterminacy. However, the efficiency of the nonlinear method is affected by the number of variables, the constraints, and the degree of overdeterminacy. The case studies show the potential for the method to optimize the design of complex, active structures applied in aerospace and construction. Future work should further test the limits of this method in terms of number of variables versus achievable deformation, and the results should be verified experimentally.

Replication of results

Nonlinear method implementation

The NLE method employed in this work is based on Luo and Lu (2006). This appendix describes its implementation in Matlab R2020a in detail.

The process iterates in order to reduce the residual strain and the residual force, both of which are possible inputs. So, given the strain vector ϵ and/or the external force vector $\mathbf{F}$, the process outputs the elements strains and nodal displacements. In this work, the equilibrium matrix is extended to include support reactions, since element elongations affect the supports directly. The formulation of the extended equilibrium matrix $\mathbf{A}^{\text{ext}}$ is detailed in Senatore and Reksowardojo (2020). The matrix is built as follows:

$$\mathbf{A}^{\text{ext}} = [\mathbf{A}^{\text{el}} \mid \mathbf{A}^{\text{sr}}] \quad (23)$$

where $\mathbf{A}^{\text{el}}$ is the usual equilibrium matrix, as defined by Pellegrino and Calladine (1986), and $\mathbf{A}^{\text{sr}}$ contains the effect of the support reactions. If the support reactions are aligned with the main axes of the system, as they usually are, this matrix contains all zeros except for the constrained nodes, where they contain ones for each constrained direction. The flexibility matrix $\mathbf{G}$ is also necessary for the NLE method, and so also needs to be extended for the support reactions. Since the supports are considered to be infinitely stiff elements, $\mathbf{G}^{\text{ext}}$ contains n^{DOFs} additional rows and columns, all set to zero.

The NLE method goes through the following steps:

1. Set nodal displacements $\mathbf{u}_0 = 0$ and elements forces $\mathbf{f}_0 = 0$.
2. Calculate $\mathbf{A}_1^{\text{ext}}$ using nodal coordinates $\mathbf{xyz}_1 = \mathbf{xyz}_0 + \mathbf{u}_0$ and the transformation matrix $\mathbf{T}$.
3. Calculate the initial residual strains and residual forces:

$$\epsilon_{\text{res},i} = \epsilon - \mathbf{A}^{\text{ext}^T}_i \cdot \mathbf{u}_i \quad (24)$$

$$\mathbf{F}_{\text{res},i} = \mathbf{F} - \mathbf{A}^{\text{ext}^T}_i \cdot \mathbf{f}_i. \quad (25)$$

4. (Start the iterative process) Use Eq. 1 for the SVD of $\mathbf{A}_1^{\text{ext}}$, calculate the stiffness matrix $\mathbf{K}$, the flexibility matrix $\mathbf{G}^{\text{ext}}$, and the geometrical force matrix $\mathbf{Kg}$ based on $\mathbf{A}_1^{\text{ext}}$ and $\mathbf{T}_1$. The calculation of $\mathbf{Kg}$ is given in Pellegriano (1990).
5. Calculate the step solutions for the element forces:

$$\mathbf{f}_i^{\text{prime}} = \mathbf{W}_r \cdot \mathbf{V}_r^{-1} \cdot (\mathbf{U}_r \cdot \mathbf{F}_{\text{res},i}) \quad (26)$$

$$a = -(\mathbf{W}_s^T \cdot \mathbf{G}^{\text{ext}} \cdot \mathbf{W}_s)^{-1} \cdot (\mathbf{W}_s^T \cdot (\epsilon_{\text{res},i} + \mathbf{G}^{\text{ext}} \cdot \mathbf{f}_i^{\text{prime}})) \quad (27)$$

$$\mathbf{f}_i = \mathbf{f}_i^{\text{prime}} + \mathbf{f}_i^{\text{prime}} \cdot a \quad (28)$$

and for the nodal displacements:

$$\mathbf{u}_i^{\text{prime}} = \mathbf{U}_r \cdot \mathbf{V}_r^{-1} \cdot (\mathbf{W}_r \cdot (-\epsilon_{\text{res},i} + \mathbf{G}^{\text{ext}} \cdot \mathbf{f}_i)) \quad (29)$$

$$b = (\mathbf{Kg} \cdot \mathbf{U}_m)^{-1} \cdot (\mathbf{Kg} \cdot \mathbf{u}_i^{\text{prime}}) \quad (30)$$

$$\mathbf{u}_i = \mathbf{u}_i^{\text{prime}} + \mathbf{U}_m \cdot b. \quad (31)$$

6. Update $\mathbf{f}_{i+1}$, $\mathbf{u}_{i+1}$, $\mathbf{xyz}_{i+1}$.
7. Calculate $\mathbf{A}_{i+1}^{\text{ext}}$ and $\mathbf{T}_{i+1}$ (see step 2).
8. Calculate $\epsilon_{\text{res},i+1}$ and $\mathbf{F}_{\text{res},i+1}$.
9. Check for convergence with $\epsilon_{\text{res},i+1} < \text{tol}$ or $\mathbf{F}_{\text{res},i+1} < \text{tol}$; if converged, exit, if not, start again at step 4.

Simulated annealing implementation

In this work, a simulated annealing (SA) algorithm is implemented to optimize the actuator placement given n^{act} . Its main characteristics are given in Sect. 3.6.1, and the details of its implementation are given below.

n_{temp} and n_{rep} are the number of temperature steps and the number of iterations at every step respectively. They are set to 30 and to n^E . The initial actuator layout $\mathbf{x}_0$ is determined using the LE method. The initial penalty λ is chosen at random from $\mathcal{U}(0, 1)$. The objective function is defined in Eq. 19 and is defined here as J .

Algorithm 3 SA implementation

```

1: Initialize  $n_{\text{temp}}, n_{\text{rep}}, \mathbf{x}_0, \mathbf{Q}, \mathbf{Q}_{\text{pinv}}, Q_{\text{new}}, Q_{\text{old}}, Q_{\text{best}}, \lambda_{\text{new}},$ 
    $\lambda_{\text{old}}, J_{\text{new}}, J_{\text{old}}$ 
2: for  $k = 1 : n^{\text{temp}}$  do Update T with Eq. 22
3:   for  $l = 1 : n^{\text{rep}}$  do
4:     if  $\text{rand}(1, 1) < 0.5$  then
5:       Update  $Q_{\text{old}}$  and  $\lambda_{\text{old}}$ 
6:       Generate a random number of actuator posi-
         tions to update  $n^u$  from  $n^u = \text{rand}(1, n^{\text{act}})$ 
7:       Reduce  $\mathbf{Q}$  and  $\mathbf{Q}_{\text{pinv}}$  to current actuator posi-
         tions and non-actuator positions respectively
8:       Replace  $n^u$  actuator positions in  $\mathbf{x}_{\text{old}}$  to form
          $\mathbf{x}_{\text{new}}$ 
9:       Evaluate  $Q_{\text{new}}$  with the NLE solver
10:      Evaluate  $J_{\text{new}} = 1 - Q_{\text{new}} + \lambda_{\text{new}}$ 
11:      Always accept for  $Q_{\text{new}} < Q_{\text{old}}$ ; else accept
         with the probability given in Eq. 20
12:      Update  $J_{\text{best}}$  if  $J_{\text{new}} < J_{\text{old}}$ 
13:    else if then
14:      Calculate  $\lambda_{\text{new}}$  and update  $J_{\text{new}}$ 
15:      Always accept for  $\lambda_{\text{new}} > \lambda_{\text{old}}$ ; else accept
         with the probability given in Eq. 21
16:      Update  $J_{\text{best}}$  if  $J_{\text{new}} < J_{\text{old}}$ 
17:    end if
18:    Record  $J_{\text{best}}$  in matrix  $\mathbf{J}$ 
19:  end for
20: end for

```

In this work, the convergence criterion is based on the variability of the entries of $\mathbf{J}$. If they have to changed over four temperature increments, the algorithm outputs J_{best} .

Acknowledgements This work was partially supported by the Strategic Focus Area Advanced Manufacturing of the ETH Board through the project “Sustainable Design of 4D Printed Active Systems.”

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated

otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Chen T, Bilal OR, Shea K, Daraio C (2018) Harnessing bistability for directional propulsion of untethered, soft robots. *Proc Natl Acad Sci USA* 115(22):5698–5702. <https://doi.org/10.1073/pnas.1800386115>
- Daynes S, Weaver PM (2013) Review of shape-morphing automobile structures: concepts and outlook. *Proc Inst Mech Eng D* 227(11):1603–1622. <https://doi.org/10.1177/0954407013496557>
- dos Santos e Lucato SL, Evans AG (2006) The load capacity of a Kagome based high authority shape morphing structure. *J Appl Mech* 73(1):128–133. <https://doi.org/10.1115/1.2042482>
- dos Santos Lucato SL, McMeeking RM, Evans AG (2004) Design and demonstration of a high authority shape morphing structure. *Int J Solids Struct* 41:3521–3543
- dos Santos Lucato SL, McMeeking RM, Evans AG (2005) Actuator placement optimization in a Kagome based high authority shape morphing structure. *Smart Mater Struct* 14:869. <https://doi.org/10.1088/0964-1726/14/4/047>
- du Pasquier C, Shea K (2021) Actuator placement optimization in an active lattice structure using generalized pattern search and verification. *Smart Mater Struct* (in review)
- Jenett B, Calisch S, Cellucci D, Cramer N, Gershenfeld N, Swei S, Kenneth C (2017) Digital morphing wing: active wing shaping concept using composite lattice-based cellular structures. *Soft Robot* 4(1):33–48. <https://doi.org/10.1089/soro.2016.0032>
- Kim YW, Kim NI, Lee J (2016) Damage identification of truss structures based on force method and free vibration analysis. *Adv Struct Eng* 19(1):3–13. <https://doi.org/10.1177/1369433215618300>
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680. <https://doi.org/10.1126/science.220.4598.671>
- Luo Y, Lu J (2006) Geometrically non-linear force method for assemblies with infinitesimal mechanisms. *Comput Struct* 84(31–32):2194–2199. <https://doi.org/10.1016/j.compstruc.2006.08.063>
- Pellegrino S (1990) Analysis of prestressed mechanisms. *Int J Solids Struct* 26(12):1329–1350. [https://doi.org/10.1016/0020-7683\(90\)90082-7](https://doi.org/10.1016/0020-7683(90)90082-7)
- Pellegrino S (1993) Structural computations with the singular value decomposition of the equilibrium matrix. *Int J Solids Struct* 30(21):3025–3035. [https://doi.org/10.1016/0020-7683\(93\)90210-X](https://doi.org/10.1016/0020-7683(93)90210-X)
- Pellegrino S, Calladine CR (1986) Matrix analysis of statically and kinematically indeterminate frameworks. *Int J Solids Struct* 22(4):409–428. [https://doi.org/10.1016/0020-7683\(86\)90014-4](https://doi.org/10.1016/0020-7683(86)90014-4)
- Reksowardojo AP, Senatore G, Smith IFC (2019) Experimental testing of a small-scale truss beam that adapts to loads through large shape changes. *Front Built Environ* 5:7. <https://doi.org/10.3389/fbuil.2019.00093>
- Reksowardojo AP, Senatore G, Smith IFC (2020) Design of structures that adapt to loads through large shape changes. *J Struct Eng* 146(5):4020068. [https://doi.org/10.1061/\(asce\)st.1943-541x.0002604](https://doi.org/10.1061/(asce)st.1943-541x.0002604)
- Senatore G, Reksowardojo AP (2020) Force and shape control strategies for minimum energy adaptive structures. *Front Built Environ* 6:105. <https://doi.org/10.3389/fbuil.2020.00105>
- Senatore G, Duffour P, Winslow P (2019) Synthesis of minimum energy adaptive structures. *Struct Multidisc Optim* 60(3):849–877. <https://doi.org/10.1007/s00158-019-02224-8>
- Suryanarayan Patnaik (1973) An integrated force method for discrete analysis. *Int J Numer Methods Eng* 6(2):237–251. <https://doi.org/10.1002/nme.1620060209>
- Tanimoto T (1958) An elementary mathematical theory of classification and prediction. https://books.google.ch/books/about/An_Elementary_Mathematical_Theory_of_Cla.html?id=yp34HAAACAAJ&redir_esc=y
- Tian Y, Zhao Y, Li L, Yuan G, Xi F (2021) Design and analysis of a multisegment shape morphing mechanism. *J Mech Robot* 13(2):4. <https://doi.org/10.1115/1.4049443>
- Wah BW, Wang T (1999) Constrained simulated annealing with applications in nonlinear continuous constrained global optimization. In: *Proceedings of the international conference on tools with artificial intelligence*, pp 381–388. IEEE. <https://doi.org/10.1109/tai.1999.809827>
- Wang J, Nausieda A, dos Santos e Lucato SL, Evans AG (2007) Twisting of a high authority morphing structure. *Int J Solids Struct* 44(9):3076–3099. <https://doi.org/10.1016/j.ijsolstr.2006.09.008>
- Wang W, Xi F, Tian Y, Zhao Y, Li Y (2020) Modeling and analysis of a planar soft panel continuum mechanism. *J Mech Robot* 12:44. <https://doi.org/10.1115/1.4046029>
- Wicks N, Hutchinson JW (2004) Journal of, and undefined. Sandwich plates actuated by a Kagome planar truss. ... http://appliedmechanics.asmedigitalcollection.asme.org/data/journals/jamcav/26584/652_1.pdf
- Yuan X, Liang X, Li A (2016) Shape and force control of prestressed cable-strut structures based on nonlinear force method. *Adv Struct Eng* 19(12):1917–1926. <https://doi.org/10.1177/1369433216652411>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.