



On the co-rotational method for geometrically nonlinear topology optimization

Peter D. Dunning¹

Received: 18 October 2019 / Revised: 18 March 2020 / Accepted: 7 April 2020 / Published online: 9 May 2020
© The Author(s) 2020

Abstract

This paper investigates the application of the co-rotational method to solve geometrically nonlinear topology optimization problems. The main benefit of this approach is that the tangent stiffness matrix is naturally positive definite, which avoids some numerical issues encountered when using other approaches. Three different methods for constructing the tangent stiffness matrix are investigated: a simplified method, where the linear elastic stiffness matrix is simply rotated; the consistent method, where the tangent stiffness is derived by differentiating residual forces by displacements; and a symmetrized method, where the consistent tangent stiffness is approximated by a symmetric matrix. The co-rotational method is implemented for 2D plane quadrilateral elements and 3-node shell elements. Matlab code is given in the appendix to modify an existing, freely available, density-based topology optimization code so it can solve 2D problems with geometric nonlinear analysis using the co-rotational method. The approach is used to solve four benchmark problems from the literature, including optimizing for stiffness, compliant mechanism design, and a plate problem. The solutions are comparable with those obtained with other methods, demonstrating the potential of the co-rotational method as an alternative approach for geometrically nonlinear topology optimization. However, there are differences between the methods in terms of implementation effort, computational cost, final design, and objective value. In summary, schemes involving the symmetrized tangent stiffness did not outperform the other schemes. For problems where the optimal design has relatively small displacements, then the simplified method is suitable. Otherwise, it is recommended to use the consistent method, as it is the most accurate.

Keywords Nonlinear geometry · Topology optimization · Co-rotational method · Compliant mechanism

1 Introduction

Research into structural topology optimization considering nonlinear geometry goes back to Jog (1996), who included its effect when optimizing for thermoelastic properties. Since then, there has been increasing interest in geometrically nonlinear topology optimization, due to the observation that, for some problems, the nonlinear and linear responses of a structure are significantly different. Thus, the optimal design can also be significantly different if nonlinear modeling is used instead of linear modeling.

Several authors have investigated the effect of nonlinear geometry on the optimal design of stiff structures, for example Buhl et al. (2000), Bruns and Tortorelli (2001), and Gea and Luo (2001). This research showed that when nonlinear geometric effects are considered, solutions can be dependent on the applied load magnitude and direction (e.g., up versus down). This was particularly noticeable for structures where the solution using linear geometry could exhibit snap-through behavior when analyzed using nonlinear geometry. Noting this, topology optimization methods have been proposed to design structures that have a specific snap-through behavior (for applications such as bi-stable switches) which obviously require nonlinear geometric modeling (Bruns et al. 2002; Bruns and Sigmund 2004).

Compliant mechanism design is another key motivation for considering nonlinear geometry, as the aim is often to achieve some relatively large motion at specific points, such as maximizing the output displacement. Several

Responsible Editor: Zhen Luo

✉ Peter D. Dunning
peter.dunning@abdn.ac.uk

¹ School of Engineering, University of Aberdeen, Aberdeen, AB24 3UE, UK

authors have shown that designs obtained using geometric nonlinearity are again different from those obtained using linear assumptions and that when linear solutions are analyzed using nonlinear geometry, their performance can be significantly worse than predicted by linear geometric modeling (Bruns and Tortorelli 2001; Pedersen et al. 2001).

Other applications of nonlinear geometric topology optimization include accurately modeling instabilities (instead of using a linear buckling analysis) (Kemmler et al. 2005; Lindgaard and Dahl 2013), maximizing the natural frequency of geometrically nonlinear structures (Yoon 2010), nonlinear material stiffness in a multi-scale optimization problem (Nakshatrala et al. 2013), maximizing the energy absorption of cellular materials (Carstensen et al. 2016), and the design of a bi-stable cardiovascular stent (James and Waisman 2016). Thus, there are a wide range of applications for nonlinear geometric topology optimization.

Most methods that consider nonlinear geometry in topology optimization (including all those mentioned above) use either an updated or total Lagrangian approach with nonlinear strain measures. Some approaches are only suitable for small strain problems (but with potentially large displacements or rotations), such as using Green's strain, e.g., Buhl et al. (2000), Kemmler et al. (2005), Luo and Tong (2008), and Lindgaard and Dahl (2013). Other methods employ hyperelastic material models for problems with potentially large elastic strains, e.g., Lahuerta et al. (2013), Wang et al. (2014), and Gomes and Senne (2014). However, it has been observed that approaches based on nonlinear strain measures may fail to converge to the nonlinear equilibrium because the tangent stiffness can become indefinite or negative definite due to large strains in the void (low-density) region (Jog 1996; Buhl et al. 2000; Bruns and Tortorelli 2001). This is primarily an issue for density-based topology optimization methods, although it affects any method that employs an ersatz material approach with low-density void elements, such as the parameterized level-set method (Luo and Tong 2008).

Several techniques have been proposed to address the convergence issue caused by low-density elements. Buhl et al. (2000) suggest removing nodes completely surrounded by void elements from the equilibrium convergence criterion, which was effective for their problems. A simple idea is to just remove void elements from the analysis. However, in density-based methods, this prevents void elements from becoming solid again. Thus, Bruns and Tortorelli (2003) created a strategy to remove and reintroduce elements. Alternatively, Jung and Gea (2004) exclude void elements from the analysis, but enable their reintroduction using a sensitivity filter. A similar strategy is employed by Cho and Kwak (2006) when using a mesh-free analysis method and a density point parameterization. Yoon and Kim (2005) used an

element connectivity parameterization scheme to avoid issues with low-density elements, where all elements remain solid and the variables are stiffness values of zero length links that connect elements. When optimizing plate structures, Boroomand and Barekatein (2009) used a modified stiffness interpolation function when the density is below a threshold, essentially stiffening void and near void elements. Kawamoto (2009) investigated using the Levenberg-Marquardt method to solve for nonlinear equilibrium, instead of the more commonly used Newton-Raphson method. It was shown that this can avoid convergence issues with low-density elements, although its success depends on the choice and update of a heuristic parameter. A strain energy-based convergence criterion was used by He et al. (2014), instead of the usual displacement, or force-based criteria, which avoided convergence issues in their problems. Lahuerta et al. (2013) used a polyconvex hyperelastic material model and relaxed the element density variable update, depending on a measure of element distortion. Also, when using a hyperelastic material model, Wang et al. (2014) suggested using an energy interpolation scheme to penalize intermediate densities, without causing convergence problems in low-density elements.

All the above methods use nonlinear strain measures and an updated, or total Lagrangian, formulation for nonlinear geometric analysis. In contrast, there is far fewer research using the co-rotational method, which is the main alternative for modeling small strain geometric nonlinearity (i.e., potentially large displacement or rotation). The co-rotational method splits the displacements computed in the global coordinate system into rigid body and straining parts (Belytschko and Hsieh 1973; Rankin and Nour-Omid 1988). The internal forces are then computed using only the straining part. Pajot and Maute (2006) presented sensitivity analysis for a consistent co-rotational method and applied it to several problems, including topology optimization of a compliant mechanism. One motivation for using the co-rotational method is that it can reduce implementation effort, as existing code for linear elements can be reused and implementation of kinematically nonlinear element formulations is not required (Pajot and Maute 2006). Another potential benefit is that convergence issues with low-density elements are naturally avoided because internal forces are computed using a linear, small strain assumption. However, despite these benefits, there have been few applications of the co-rotational method for geometrically nonlinear topology optimization problems. The author could only find one other example of it being applied to the thickness optimization of an aeroelastic flapping shell (Stanford and Beran 2013).

Perhaps one reason for the lack of co-rotational topology optimization methods is that the implementation for the consistent method is not straightforward. The consistent

co-rotational method defines the tangent stiffness matrix as the variation of internal forces with respect to the displacement and rotational degrees of freedom in the global coordinate system. In the method presented by Pajot and Maute (2006), the consistent tangent stiffness matrix is the sum of up to five matrices, some of which are unsymmetric. An unsymmetric tangent stiffness matrix increases storage requirements and solution time compared with a symmetric matrix. However, Pajot and Maute (2006) suggest using a symmetrized matrix to avoid implementing schemes for unsymmetric matrices, as the approximate (symmetric) tangent stiffness does not prevent the method from reaching equilibrium. Furthermore, part of the consistent tangent stiffness matrix is dependent on the element type and must be derived for each new element.

To avoid some of the difficulties in implementing a consistent co-rotational method, a simplified method can be used, where element tangent stiffness matrices are defined from the element rigid body rotation and linear elastic matrix. Thus, once element rigid body motions are computed, the tangent stiffness matrix is easily obtained and is naturally symmetric. As mentioned above, the use of an approximate (symmetric) tangent stiffness does not necessarily prevent the co-rotational method from reaching equilibrium. However, as sensitivities with respect to design variables often require the tangent stiffness (i.e., when solving the adjoint), Pajot and Maute (2006) recommend at least using a consistent matrix for sensitivity computation to obtain accurate values.

The aim of this paper is to investigate the use of the co-rotational method for density-based topology optimization. As discussed above, there are three methods for constructing the tangent stiffness matrix: the simplified method (simple), consistent, and symmetrized-consistent method (symmetric). This gives rise to six potential co-rotational schemes for topology optimization, which are summarized in Table 1.

The simple method requires least implementation effort and each evaluation of the tangent stiffness is cheaper (compared with the symmetric and consistent methods).

Table 1 Summary of investigated co-rotational schemes

No.	Method name	Tangent stiffness	
		Equilibrium	Adjoint
1	Simple	Simple	Simple
2	Simp-Sym	Simple	Symmetric
3	Simp-Con	Simple	Consistent
4	Symmetric	Symmetric	Symmetric
5	Sym-Con	Symmetric	Consistent
6	Consistent	Consistent	Consistent

However, the sensitivities are only approximate, which could lead to more optimization iterations, convergence issues, or poor solutions for some problems. This may be improved by using a more accurate tangent stiffness for the adjoint solve only (e.g., the Simp-Sym or Simp-Con methods) at increased implementation effort and higher computational cost when constructing the tangent stiffness for the adjoint solve. The consistent matrix is unsymmetric, which leads to more computational cost in solving linear equations, compared with the simple and symmetrized methods, which have symmetric matrices. Thus, these six schemes are different in terms of implementation effort, computational cost, and potentially the quality of the solution (i.e., objective function value). These aspects are investigated in this paper using benchmark problems from the literature that are chosen because it has been demonstrated that geometric nonlinearity significantly influences the optimal design (compared with linear modeling).

Note that this study uses a linear elastic material model. For problems where the effect of large or nonlinear strains (e.g., hyperelastic or elastic-plastic material models) is important, then these models can be combined with the co-rotational method by simply evaluating strains after rigid body motion has been eliminated from the element displacement vector (Cook et al. 2002).

The paper is organized as follows. Section 2 introduces the co-rotational method for two types of element: a 4-node 2D element and a 3-node 3D shell element. Section 3 details the density-based topology optimization method used, optimization problems considered, and sensitivity analysis. Section 4 presents the results for four benchmark problems from the literature, followed by the conclusions in Section 5. In Appendix A, 116 lines of Matlab code are presented that can be used to modify the 88-line density-based topology optimization code, developed by Andreassen et al. (2011), so that it can solve geometrically nonlinear problems using the co-rotational method.

2 Co-rotational formulation

The principle of the co-rotational method is to split displacements computed in the global coordinate system into rigid body and straining parts. This can be done at the element, or integration point, level. In this paper, a co-rotational method is implemented at the element level for two types of element: a 2D quadrilateral (without rotational dof) and a three-node shell (with rotational dof). In both cases, the rigid body movement and rotation are determined by assigning a local coordinate system to each element, which moves with the element during deformation, as shown in Fig. 1. The rotation of the local coordinate system

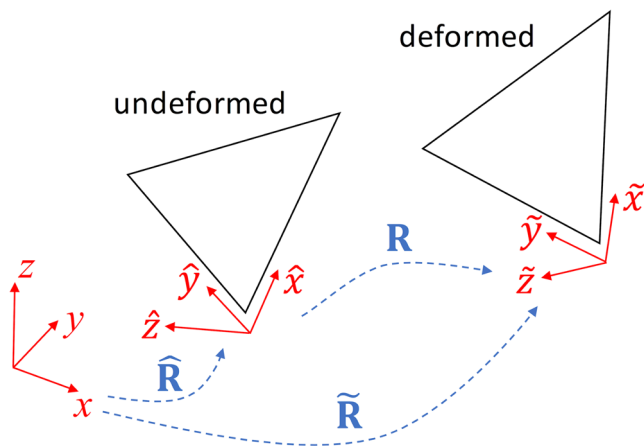


Fig. 1 Overview of co-rotational method coordinate systems

from undeformed to deformed state is described using a rotation matrix, \mathbf{R}_e . Once the rotation matrix for an element is determined, the simplified element tangent stiffness matrix is computed by rotating the element stiffness matrix computed in the global coordinate system:

$$\mathbf{K}_{T,e} = \mathbf{R}_e^T \mathbf{K}_e \mathbf{R}_e \quad (1)$$

where \mathbf{K}_e is the element stiffness matrix in the global coordinate system, and $\mathbf{K}_{T,e}$ is the element tangent stiffness matrix which is assembled into the global matrix in the usual manner for finite element methods. Note that this matrix is not consistent, as it is not obtained by differentiating the internal forces with respect to global displacements. The derivation of the consistent tangent stiffness is element dependent and explained below.

It is also useful to introduce notation for the different coordinate systems used in the formulation: x refers to the global coordinate system, \hat{x} refers to the element local coordinate system in the undeformed state, and \tilde{x} refers to the element local coordinate system in the deformed state (see Fig. 1).

2.1 2D quadrilateral elements

The rotation matrix for 2D quadrilateral elements is constructed using the bisector method (Izzuddin 2005), which computes axis vectors for the local undeformed coordinate system from the nodal coordinates in the global system as:

$$\begin{aligned} \mathbf{e}_{31} &= (x_3 - x_1, y_3 - y_1), \quad \bar{\mathbf{e}}_{31} = \mathbf{e}_{31} / \|\mathbf{e}_{31}\|_2 \\ \mathbf{e}_{42} &= (x_4 - x_2, y_4 - y_2), \quad \bar{\mathbf{e}}_{42} = \mathbf{e}_{42} / \|\mathbf{e}_{42}\|_2 \\ \mathbf{e}_x &= \bar{\mathbf{e}}_{31} - \bar{\mathbf{e}}_{42}, \quad \bar{\mathbf{e}}_x = \mathbf{e}_x / \|\mathbf{e}_x\|_2 \\ \mathbf{e}_y &= \bar{\mathbf{e}}_{31} + \bar{\mathbf{e}}_{42}, \quad \bar{\mathbf{e}}_y = \mathbf{e}_y / \|\mathbf{e}_y\|_2 \end{aligned} \quad (2)$$

where (x_i, y_i) are undeformed nodal coordinates in the global system. The vectors, $\bar{\mathbf{e}}_x$ and $\bar{\mathbf{e}}_y$, are then rows in

the 4×4 rotation matrix, $\hat{\mathbf{R}}_e$, that transforms from the global to the undeformed local system. To compute the rotation matrix, $\tilde{\mathbf{R}}_e$, that transforms from global to deformed element coordinate system, the same equations are used (2), but the undeformed nodal coordinates are replaced with the deformed ones (in the global system). The rotation matrix required in (1) transforms from the local element undeformed system to the local deformed system, which can be computed by first rotating from element undeformed to global and then from the global to element deformed system:

$$\mathbf{R}_e = \tilde{\mathbf{R}}_e \hat{\mathbf{R}}_e^T \quad (3)$$

This matrix is then expanded to a 16×16 block diagonal matrix to complete the matrix multiplication required by (1). Note that if the undeformed coordinate system is the same as the global system, then: $\hat{\mathbf{R}}_e = \mathbf{I}$ and $\mathbf{R}_e = \tilde{\mathbf{R}}_e$. This simplification is used in the Matlab implementation included in Appendix A, as the analysis mesh is a regular grid aligned to the global coordinate system.

To solve the nonlinear equilibrium equation, the internal element forces are also required, so that the global residual force vector can be computed. In the co-rotational method, internal element forces are computed from the straining part of the displacement vector in the deformed local coordinate system. These forces are then transformed back to the global system using the already computed rotation matrix:

$$\mathbf{f}_{int,e} = \mathbf{R}_e^T \mathbf{K}_e \tilde{\mathbf{u}}_e \quad (4)$$

where $\mathbf{f}_{int,e}$ is the element internal force vector in the global coordinate system and $\tilde{\mathbf{u}}_e$ is a vector of the straining part of the local element displacement vector:

$$\tilde{\mathbf{u}}_e = \mathbf{R}_e \tilde{\mathbf{x}}_e - \mathbf{x}_e \quad (5)$$

where $\tilde{\mathbf{x}}_e$ and \mathbf{x}_e are vectors of the nodal coordinates in the global system for the deformed and undeformed states, respectively. The global internal force vector is assembled by summing the elemental vectors in the normal manner. Thus, we can see that the simplified tangent stiffness matrix, defined in (1), is not consistent with the internal force defined in (4) because it ignores the variation of the rotation matrix \mathbf{R}_e with respect to displacements in the global system \mathbf{u} , although this does not necessarily prevent the method from obtaining equilibrium (Pajot and Maute 2006). The consistent tangent stiffness matrix is defined as:

$$\mathbf{K}_{T,e} = \frac{d\mathbf{f}_{int,e}}{d\mathbf{u}_e} = \left[\frac{d\mathbf{R}_e}{d\mathbf{u}_e} \right]^T \mathbf{K}_e \tilde{\mathbf{u}}_e + \mathbf{R}_e^T \mathbf{K}_e \frac{d\tilde{\mathbf{u}}_e}{d\mathbf{u}_e} \quad (6)$$

Now, using (5) and assuming \mathbf{K}_e is symmetric, after some manipulation, this can be written as:

$$\mathbf{K}_{T,e} = \mathbf{R}_e^T \mathbf{K}_e \mathbf{R}_e + [\mathbf{M}_e + \mathbf{M}_e^T] \tilde{\mathbf{x}}_e - \mathbf{N}_e \mathbf{x}_e \quad (7)$$

where the matrices \mathbf{M}_e and \mathbf{N}_e are defined as:

$$\mathbf{N}_e = \left[\frac{d\mathbf{R}_e}{d\mathbf{u}_e} \right]^T \mathbf{K}_e, \mathbf{M}_e = \mathbf{N}_e \mathbf{R}_e \tag{8}$$

The derivative $\frac{d\mathbf{R}_e}{d\mathbf{u}_e}$ can be efficiently obtained using directional derivatives. The full equations are complex and omitted here for brevity. However, they are included in the Matlab code in Appendix A. Note that the first term of the consistent tangent stiffness in (7) is the same as the simplified tangent stiffness in (1). Thus, the last two terms in (7) can be viewed as modifying the simplified version, by including the dependence of the rotation matrix on the global displacement vector.

Note that when computing nodal coordinates in (5), it is good practice to center the coordinate system somewhere within the element. This helps avoid rounding error issues, especially when the structural dimensions are much larger than displacements. For the Matlab implementation included in Appendix A, the coordinate system is always centered on the first node.

2.2 Three-node shell elements

The procedure used for computing the rotation matrix for a three-node shell element starts by defining the local x -axis vector (in the global system) as pointing from node 1 to node 2:

$$\mathbf{e}_x = (x_2 - x_1, y_2 - y_1, z_2 - z_1), \bar{\mathbf{e}}_x = \mathbf{e}_x / \|\mathbf{e}_x\|_2 \tag{9}$$

Then, the local z -axis vector is defined as being normal to the plane and the y -axis vector is perpendicular to the already computed local x - and z -axes. The details are omitted for brevity, although care must be taken to ensure the sign convention of the axis vectors is consistent; otherwise, the element normal could erroneously change sign, resulting in errors. The axis vectors are then rows in the 3×3 matrix, $\hat{\mathbf{R}}_e$, that transforms from the global to the undeformed local system. The rotation matrix, $\tilde{\mathbf{R}}_e$, that transforms from global to deformed local element coordinate system is obtained in the same way, except the deformed nodal coordinates are used instead of the undeformed ones (in the global system).

The method follows the same procedure as detailed for the 2D quadrilateral elements above. The rotation matrix \mathbf{R} is computed using (3), simplified tangent stiffness matrix by (1), and internal forces by (4). However, for elements with rotational dof, an additional procedure is required to compute the straining part for the rotational dof. This is not trivial, as compound rotations in 3D are non-communicative (i.e., the result depends on the order that rotations are

applied). A common tool to deal with this issue is to use the incremental rotation matrix (Argyris 1982):

$$\mathbf{Q}(\beta) = \mathbf{I} + \frac{\sin(\beta)}{\beta} \mathbf{S}(\beta) + \frac{1}{2} \frac{\sin(\beta/2)^2}{(\beta/2)^2} \mathbf{S}(\beta)^2 \tag{10}$$

where β is a vector of rotational dof for a single node and β and $\mathbf{S}(\beta)$ are defined by:

$$\beta = (\beta_x^2 + \beta_y^2 + \beta_z^2)^{0.5} \tag{11}$$

$$\mathbf{S}(\beta) = \begin{bmatrix} 0 & -\beta_z & \beta_y \\ \beta_z & 0 & -\beta_x \\ -\beta_y & \beta_x & 0 \end{bmatrix} \tag{12}$$

where β_x is the rotation about the x -axis, etc.

The problem is to find nodal rotational dof, $\tilde{\beta}$, that removes the rigid body rotation identified by the element rotation matrix, \mathbf{R}_e . This can be found by solving:

$$\tilde{\mathbf{Q}}(\tilde{\beta}) = \mathbf{R}_e^T \mathbf{Q}(\beta) \tag{13}$$

where β is a vector of rotational dof at a node in the global coordinate system and $\tilde{\beta}$ is the local rotational dof for the node with respect to the rigid body rotation of element e . This is solved using quaternions and an algorithm presented by Spurrier (1978). The method to compute $\tilde{\beta}$ from $\tilde{\mathbf{Q}}$ follows that of Crisfield (1990) and is detailed in Appendix B.

Once $\tilde{\beta}$ is obtained for each node in the element, and the straining parts of the translational dof are computed using (5), then the element residual force vector is computed using (4) (which will contain both forces and moments).

The consistent tangent stiffness matrix for the shell element is more complex than the 2D quadrilateral due to the rotational degrees of freedom. It can be obtained analytically, e.g., by using the projector method (Pajot and Maute 2006), which is the preferred method for computational efficiency, but requires more implementation effort. In this paper, a simpler method (in terms of implementation) is used, where the consistent tangent stiffness is built using finite differences on the element internal force vector.

2.3 Nonlinear FEA

The co-rotational method introduces nonlinearity in the finite element (FE) equilibrium equation because the stiffness depends nonlinearly on the deformed state (1). In this paper, the nonlinear equilibrium equation is solved using the Newton-Raphson method:

$$\mathbf{K}_T(\mathbf{u}^n) \Delta \mathbf{u}^n = \mathbf{f}_{\text{res}}(\mathbf{u}^n) = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}(\mathbf{u}^n) \tag{14}$$

where $\mathbf{K}_T(\mathbf{u}^n)$ is the global tangent stiffness matrix, which depends on the displacement vector \mathbf{u}^n (where n is the current Newton-Raphson iteration), $\mathbf{f}_{\text{res}}(\mathbf{u}^n)$, \mathbf{f}_{ext} , and $\mathbf{f}_{\text{int}}(\mathbf{u}^n)$ are the residual, external, and internal force vectors, respectively. Note that in this paper the external force is assumed independent of the displacements. The displacement increment, $\Delta\mathbf{u}^n$, is obtained by solving (14) and then used to update the guess for the displacement vector (in the global coordinate system):

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta\mathbf{u}^n \quad (15)$$

The displacement vector is iteratively updated using (14) and (15) until a convergence criterion is met. Several convergence criteria can be used, such as force-based, displacement-based, and energy-based. In the Matlab implementation included in Appendix A, a simple force-based criterion is used, where the maximum residual force must be much smaller than the external applied forces:

$$f_{\text{err}}(\mathbf{u}) = \|\mathbf{f}_{\text{res}}(\mathbf{u})\|_{\infty} / \|\mathbf{f}_{\text{ext}}\|_2 < \epsilon \quad (16)$$

where ϵ is a small value (5×10^{-6} is used in this paper, unless otherwise stated).

The Newton-Raphson method may sometimes start to diverge. Thus, two schemes are used to help stabilize convergence. The first scheme is an under-relaxation, where, if the measure used for the convergence criterion increases, $f_{\text{err}}(\mathbf{u}^{n+1}) > f_{\text{err}}(\mathbf{u}^n)$, then the displacement increment is reduced:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \gamma \Delta\mathbf{u}^n \quad (17)$$

where $0 < \gamma < 1$ is a relaxation factor. Initially, $\gamma = 0.25$, but if f_{err} is still increasing, then γ is further reduced by a factor of 0.25. This continues until either $f_{\text{err}}(\mathbf{u}^{n+1}) < f_{\text{err}}(\mathbf{u})$, or $\gamma < 0.01$. If the second condition is true, then a second scheme is activated, where the load increment is reduced by a factor of 0.25 and an attempt is made to converge at a lower load level. Note that the initial load increment is set to 1. If the load increment is reduced to less than 0.01, then it is assumed a local limit point has been reached because the maximum residual force increases with a small change in the displacement (in the direction of the tangent stiffness). When this occurs, then the load is increased to the maximum level and one final attempt is made to converge (so the optimizer cannot take advantage of a design with a lower applied load). If this fails, then no special treatment is used in this paper and the Newton-Raphson iterations are simply stopped. Objective function, constraint values, and sensitivities are then computed using the displacement vector that corresponds to the lowest force residual (i.e., the displacement vector from the final Newton-Raphson iteration). This is not generally a robust

way to deal with limit points, as the optimizer may take advantage of the equilibrium equation not being satisfied. However, limit points do not occur for most problems studied in this paper and for all problems, the final solution is feasible, i.e., the nonlinear equilibrium equation is satisfied. The full Newton-Raphson algorithm is detailed in the Matlab code in Appendix A.

3 Topology optimization

This section describes the topology optimization formulation, problems studied, and sensitivity analysis when using the co-rotational method.

3.1 Modified SIMP and filtering

The modified SIMP (Simple Isotropic Material with Penalization) method is used, which assigns a constant pseudo-density to each element: $\bar{\chi}_e \in [0, 1]$. Young's modulus for an element, $E_e(\bar{\chi}_e)$, is then:

$$E_e(\bar{\chi}_e) = E_{\min} + \bar{\chi}_e^p (E_0 - E_{\min}) \quad (18)$$

where E_0 is Young's modulus of the material, E_{\min} is a small value to prevent the stiffness matrix becoming singular ($E_{\min} = 10^{-9}E_0$), and p is the penalization factor, where the typical value of $p = 3$ is used in this paper. Note that the co-rotational method does not use nonlinear strain measures, so the linear, small strain formulation is used to create element stiffness matrices.

A density filter (Bruns and Tortorelli 2001; Bourdin 2001) is used to avoid the well-known problems of checkerboard patterns and mesh-dependent solutions. The idea is to introduce a second set of pseudo-density values for each element, $\chi_e \in [0, 1]$, that are filtered using a linear hat filter to obtain the physical densities used in (18):

$$\bar{\chi}_e = \frac{\sum_{i \in N_{i,e}} H_{i,e} \chi_i}{\sum_{i \in N_{i,e}} H_{i,e}} \quad (19)$$

where $N_{i,e}$ is the set of elements i where the distance between the center of element i and center of element e , $d(e, i)$, is less than the filter radius r_{\min} . $H_{i,e}$ is a weight factor, defined as:

$$H_{i,e} = \max(0, r_{\min} - d(e, i)) \quad (20)$$

The second set of pseudo-density values χ are then the design variables. However, as χ has no physical meaning, solutions are plotted for the filtered, physical densities, $\bar{\chi}$.

3.2 Optimization problems

Three problem are considered in this paper. The first is the classic minimization of compliance problem, subject to an upper limit on material volume:

$$\begin{aligned} \min_{\chi} C &= \mathbf{f}_{\text{ext}}^T \cdot \mathbf{u} \\ \text{s.t. } V(\chi) &\leq V^* \\ \mathbf{f}_{\text{int}}(\chi, \mathbf{u}) &= \mathbf{f}_{\text{ext}} \\ \mathbf{0} &\leq \chi \leq \mathbf{1} \end{aligned} \tag{21}$$

where $V(\chi)$ is the volume ratio, defined as:

$$V(\chi) = \frac{\sum_{N_e} \bar{\chi}_e(\chi) V_e}{\sum_{N_e} V_e} \tag{22}$$

where N_e is the set of elements in the design domain and V_e is the volume of element e . For geometrically nonlinear analysis, problem (21) is also called the minimization of end compliance, as it considers only the stiffness when the full load is applied.

For nonlinear problems, the minimization of complimentary work is often used as the objective, as it considers the stiffness of the structure over the full range of loading. Complimentary work is the integral of work as load is increased from zero to full. It is usually evaluated by discretizing over n equal load steps $\Delta \mathbf{f}_{\text{ext}}$:

$$\begin{aligned} \min_{\chi} CW &\approx \Delta \mathbf{f}_{\text{ext}}^T \left[\sum_{i=1}^{n-1} \mathbf{u}_i(\mathbf{f}_{\text{ext},i}) + \frac{1}{2} \mathbf{u}_n(\mathbf{f}_{\text{ext},n}) \right] \\ \text{s.t. } V(\chi) &\leq V^* \\ \mathbf{f}_{\text{int},i}(\chi, \mathbf{u}_i) &= \mathbf{f}_{\text{ext},i} \\ \mathbf{0} &\leq \chi \leq \mathbf{1} \end{aligned} \tag{23}$$

where $\mathbf{f}_{\text{ext},i} = i \times \Delta \mathbf{f}_{\text{ext}}$. In this paper, $n = 10$ is used; so, $\Delta \mathbf{f}_{\text{ext}} = 0.1 \mathbf{f}_{\text{ext}}$.

The third problem considered is the design of a compliant mechanism that maximizes the displacement for a specific degree of freedom, u_{out} for given external input force, and constraints on compliance (or end compliance) and volume.

$$\begin{aligned} \max_{\chi} u_{\text{out}} &= \mathbf{z}^T \cdot \mathbf{u} \\ \text{s.t. } V(\chi) &\leq V^* \\ \mathbf{f}_{\text{ext}}^T \cdot \mathbf{u} &\leq C^* \\ \mathbf{f}_{\text{int}}(\chi, \mathbf{u}) &= \mathbf{f}_{\text{ext}} \\ \mathbf{0} &\leq \chi \leq \mathbf{1} \end{aligned} \tag{24}$$

where \mathbf{z} is a vector with only one nonzero value indicating the degree of freedom and direction (sign) for which the displacement should be maximized.

If geometrically nonlinear analysis is used, then the equilibrium condition in problems (21), (23), and (24) is solved using the co-rotational method with the Newton-Raphson method, as detailed in Section 2. For linear

problems, equilibrium is obtained directly by solving the following linear equation:

$$\mathbf{f}_{\text{int}}(\chi, \mathbf{u}) = \mathbf{K}(\chi) \mathbf{u} = \mathbf{f}_{\text{ext}} \tag{25}$$

where \mathbf{K} is the global linear elastic stiffness matrix.

3.3 Sensitivity analysis

Derivatives are required to use a gradient-based optimizer to efficiently solve the optimization problems introduced in the previous section. The derivatives when linear geometric analysis is used are well-known and omitted for brevity, but can be found in various publications, e.g., Sigmund (2007) and Andreassen et al. (2011). Thus, this section presents the sensitivity analysis for the end compliance, complimentary work, and output displacement functions, when nonlinear geometry is modeled using the co-rotational method.

Derivatives of end compliance, with respect to physical element constant pseudo-density values, $\tilde{\chi}_i$ are obtained using the adjoint method:

$$L = \mathbf{f}_{\text{ext}}^T \cdot \mathbf{u} - \lambda^T [\mathbf{f}_{\text{int}}(\chi, \mathbf{u}) - \mathbf{f}_{\text{ext}}] \tag{26}$$

where λ are the adjoint variables. Differentiating with respect to a physical pseudo-density values gives:

$$\frac{dL}{d\tilde{\chi}_e} = \mathbf{f}_{\text{ext}}^T \cdot \frac{\partial \mathbf{u}}{\partial \tilde{\chi}_e} - \lambda^T \left[\frac{\partial \mathbf{f}_{\text{int}}}{\partial \tilde{\chi}_e} + \frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \tilde{\chi}_e} \right] \tag{27}$$

The resulting adjoint equation is then:

$$\left[\frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{u}} \right]^T \lambda = \mathbf{K}_T^T(\mathbf{u}) \lambda = \mathbf{f}_{\text{ext}} \tag{28}$$

where the definition of the tangent stiffness matrix, $\mathbf{K}_T(\mathbf{u}) = [\partial \mathbf{f}_{\text{int}} / \partial \mathbf{u}]$, has been used, and $\mathbf{K}_T(\mathbf{u})$ is conveniently obtained at the end of the Newton-Raphson equilibrium iterations.

Note that the tangent stiffness used in the simplified method, (1), is not consistent with the internal forces. Thus, using the simplified tangent stiffness matrix in (28) results in some error in the solution of adjoint variables, leading to inaccurate sensitivities. This is also true if the symmetrized consistent tangent stiffness matrix is used to solve the adjoint, although perhaps the error is lower, compared with the simplified method. This is investigated using the numerical examples in Section 4.

After solving (28) for λ , the derivative of the end compliance is then:

$$\frac{dC}{d\tilde{\chi}_e} = -\lambda^T \cdot \frac{\partial \mathbf{f}_{\text{int}}}{\partial \tilde{\chi}_e} \tag{29}$$

When using the co-rotational method for geometrically nonlinear analysis, the derivative of the internal forces with respect to the element physical pseudo-density is found by differentiating (4):

$$\frac{\partial \mathbf{f}_{\text{int}}}{\partial \tilde{\chi}_e} = \mathbf{R}_e^T \frac{\partial \mathbf{K}_e}{\partial \tilde{\chi}_e} \tilde{\mathbf{u}}_e \quad (30)$$

From the modified SIMP (18), this can be written as:

$$\frac{\partial \mathbf{f}_{\text{int}}}{\partial \tilde{\chi}_e} = p \tilde{\chi}_e^{p-1} \left(1 - \frac{E_{\text{min}}}{E_0} \right) \mathbf{f}_{\text{int},e}(\mathbf{u}) \quad (31)$$

The same method is used to obtain derivatives for output displacement for the compliant mechanism problem (24):

$$\frac{du_{\text{out}}}{d\tilde{\chi}_e} = -\gamma^T \cdot \frac{\partial \mathbf{f}_{\text{int}}}{\partial \tilde{\chi}_e} \quad (32)$$

where $\partial \mathbf{f}_{\text{int}}/\partial \tilde{\chi}_e$ is obtained from (31) and γ are the adjoint variables, obtained by solving the following adjoint equation:

$$\mathbf{K}_T^T(\mathbf{u})\gamma = \mathbf{z} \quad (33)$$

Complimentary work (23) can be formulated as the weighted sum of end compliance values at different load levels. Thus, the sensitivity is obtained by simple extension of (30) to be the weighted sum of sensitivity for different load levels. Note that a separate adjoint is required for each load level, as the tangent stiffness matrix depends on the deformed state.

Finally, the chain rule is used to obtain derivatives with respect to the design variables χ . For example:

$$\frac{dC}{d\chi_e} = \sum_{i \in N_{i,e}} \left(\frac{dC}{d\tilde{\chi}_i} \frac{d\tilde{\chi}_i}{d\chi_e} \right) \quad (34)$$

where the $d\tilde{\chi}_i/d\chi_e$ term is obtained by differentiating the filtering function (19). For more details, see Andreassen et al. (2011).

4 Numerical examples

All optimization problems are solved using the Method of Moving Asymptotes (MMA) (Svanberg 2002), as implemented in the package NLopt (Johnson 2018). The convergence criterion used for all examples is the relative change in objective function to be less than 10^{-5} .

It was found that, in general, better solutions (in terms of objective function) are obtained when the problem is scaled before sending information to the optimizer. Compliance, complimentary work, and displacement functions are scaled such that the mean of the absolute sensitivity values in

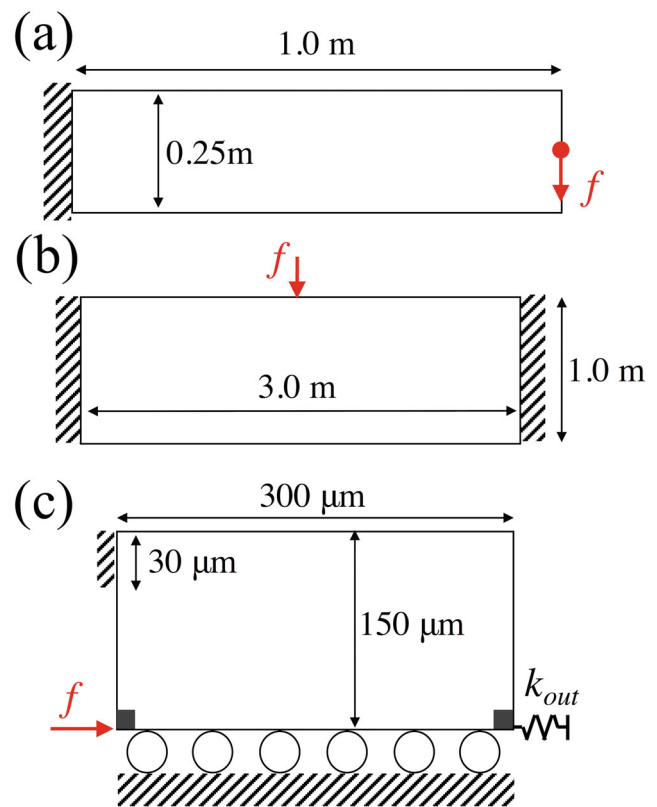


Fig. 2 2D examples: **a** cantilever beam, **b** double-clamped beam (with possible snap-through behavior), **c** compliant inverter mechanism (top half)

the first optimization iteration is 1. The volume fraction function is scaled by multiplying by the number of elements.

Figure 2 shows the 2D problems used in this study, which were all previously studied by others in the context of geometrically nonlinear topology optimization (Buhl et al. 2000; Pedersen et al. 2001).

4.1 2D cantilever

The first example is a long cantilever beam (aspect ratio 4) with a tip load applied at the center of the right edge (Fig. 2a), as previously studied by Buhl et al. (2000) and others. The material has a Young's modulus of 3 GPa, Poisson's ratio 0.4, and the domain thickness is 0.1 m. The end compliance of the beam is minimized, subject to a volume constraint equal to 50% of the design domain (21) and the force is 144 kN. The domain is discretized with 120×30 square plane stress elements. The filter radius in (20) is set to 2 element edge lengths.

The solutions obtained using all six methods are shown in Fig. 3, with objective function values, number of optimization iterations, and number of linear solves shown in Table 2. Firstly, Fig. 3 shows that all methods, except the

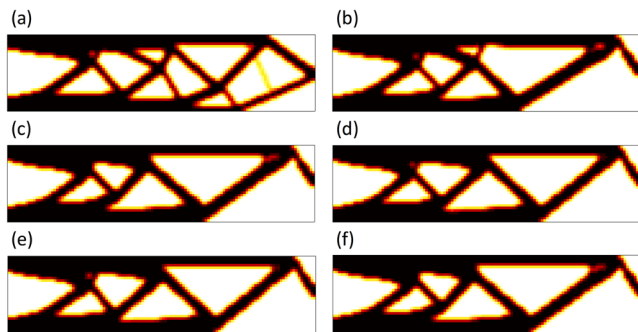


Fig. 3 Cantilever solutions, obtained using **a** simple, **b** Simp-Sym, **c** Simp-Con, **d** symmetric, **e** Sym-Con, and **f** consistent methods

simple method, obtain very similar solutions, which are also very similar to the solution obtained by Buhl et al. (2000). The objective values for these five solutions are within about 1% (Table 2). The solution obtained using the simplified method does not have the characteristic degenerate strut connected to the loading point. There is also a thin diagonal member made from intermediate material near the right-hand side. On investigation, the lower right-hand member (attached to the thin member) is close to buckling, and when the optimizer tries to remove the thin member, the end compliance increases as the attached member starts to buckle. The increase in objective value activates a line search built into the optimizer, resulting in the thin member being retained in the solution. Thus, it does not appear to be a problem with the simplified method itself, but caused by a sudden change in stiffness due to an instability. For example, if the same problem is solved with a slightly different setup, e.g., using a larger filter radius, or employing a continuation scheme, then a solution can be obtained without the thin member.

Convergence for the first 100 iterations is shown in Fig. 4 for all six methods, after which there is little change in objective value and convergence is generally smooth. However, there are some oscillations near the start, particularly for the Simp-Sym and Simp-Con methods. This is caused by the optimizer updating to a design that either contains a limit point or where equilibrium is near

Table 2 Cantilever data

Method	Iterations	C (kNm)	No. of solves
Simple	90	27.37	3165
Simp-Sym	292	27.08	67,926
Simp-Con	263	26.94	69,132
Symmetric	312	26.77	8793
Sym-Con	385	26.78	10,853
Consistent	221	26.81	6584

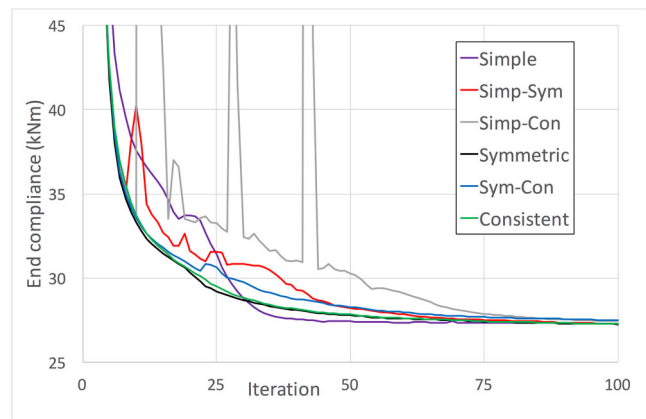


Fig. 4 Cantilever convergence

a limit point and convergence to equilibrium fails. Further investigation suggests that the simplified tangent stiffness sometimes has difficulty obtaining equilibrium near a limit point. Notice that the symmetric and Sym-Con methods do not have such oscillations. However, in all cases, the chosen optimizer is able to recover from a failed equilibrium convergence, helped by the in-built line search feature, although this may not be the case if the failed convergence resulted in an artificially improved objective value.

To compare computational cost, the number of linear solves is used as a proxy in Table 2 because exact timing is influenced by many factors, such as hardware, implementation, efficiency of library functions, and compiler. However, the simplified tangent stiffness requires fewer computations to assemble, compared with the consistent method, and solution of the linear equation using an unsymmetric matrix takes longer than if a symmetric matrix is used. Thus, the relative computational cost of each method should also be compared. For the Matlab implementation shown in Appendix A, the times for one linear solve are approximately 0.75 s, 4.45 s, and 4.51 s when using the simple, symmetrized, and consistent tangent stiffness matrices, respectively (Matlab R2018a, running on a 2.4 GHz Intel Core 2 Duo). This includes the time to evaluate the residual, assemble the global tangent stiffness matrix, and solve the resulting linear equation. Note that, for this example, the solve time is dominated by the assembly of the tangent stiffness matrix. There was some effort at optimizing this function, but further efficiency gains may be possible. For example, the blockdiag function is called several times, but could be replaced by a more efficient implementation. However, for the current implementation, the simple method is approximately 6 times faster per solve than the other two methods.

From Table 2, by far the most efficient method is the simple method, as it requires the least number of

solves and the tangent stiffness is quickest to assemble. However, the use of the simplified tangent stiffness when solving the adjoint leads to some error in the sensitivities and a notably different solution, compared with the other methods. Despite inaccurate sensitivities, a reasonable solution is still obtained with smooth convergence and only a 2% penalty in objective function. Considering the other methods, Table 2 shows that the Simp-Sym and Simp-Con methods use approximately 10 times the number of solves compared with the consistent method, which is partly due to the extra optimization iterations required to reach the solution. Thus, the next best method, in terms of computational cost (for this example and implementation), is the consistent method, which also obtains an objective value within 0.2% of the best solution.

In summary, for this example (and implementation), the consistent method produces the best overall result, as it found a good solution (close to the best in terms of objective) with lowest computational cost (compared to the other methods that found similar solutions). However, the simple method is also an option, as it is the most efficient and easiest to implement, but still finds a reasonable solution, with only a 2% penalty in objective value (compared with the best solution). Methods involving the symmetrized consistent matrix did not provide any significant advantage over using the actual consistent matrix, as the computational cost is dominated by the assembly of the global tangent stiffness matrix, which is the same for both methods.

4.2 Double-clamped beam

The next example is a double-clamped beam, with a point load applied at the center of the top edge (Fig. 2b), as previously studied by Buhl et al. (2000). The material has a Young's modulus of 3 GPa, Poisson's ratio 0.4, and the domain thickness is 0.1 m. The end compliance is minimized, subject to a volume constraint equal to 10% of the design domain, (21). The domain is discretized with 240×80 square plane stress elements and the filter radius is equal to 2 times the element edge length. Previous studies of this problem have shown that it usually progresses slowly toward the optimum. Thus, the convergence tolerance is reduced to 10^{-10} , or a maximum of 1500 iterations.

The solutions using all six methods are very similar (Fig. 5), with only some minor differences in geometry, leading to a 4% variance in objective value (Table 3). The solutions are also very similar to that found by Buhl et al. (2000), with a difference in the two small struts at the clamped edges. The end compliance is higher than that found by Buhl et al. (2000). However, in this study, a density filter is used that results in inefficient

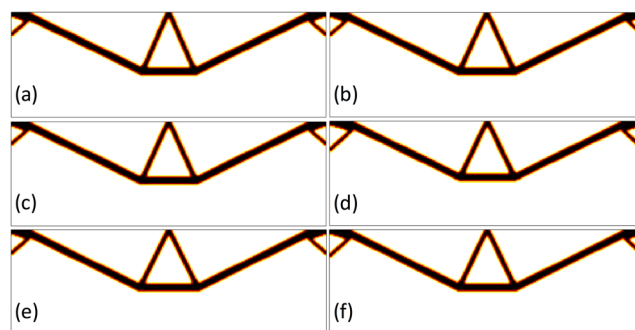


Fig. 5 Double-clamped beam solutions, obtained using **a** simple, **b** Simp-Sym, **c** Simp-Con, **d** symmetric, **e** Sym-Con, and **f** consistent methods

intermediate densities around the boundary, whereas the previous study used a sensitivity filter that resulted in almost no intermediate densities in the solution. If the co-rotational schemes presented in this paper are used with a topology optimization method that produces solutions with almost no intermediate densities, then a similar end compliance value to Buhl et al. (2000) is obtained and the topology is almost identical to that shown in Fig. 5.

This example shows that the simplified method can obtain a similar solution as the consistent method. This can be explained, as the displacement of the optimum design is not large, so the error in the tangent stiffness and sensitivities when using the simplified method is small. The effect of nonlinear geometry in this example is that it helps the optimizer avoid a solution with snap-through instability, which is obtained if linear modeling is used (Buhl et al. 2000).

Figure 6 shows that all methods converge smoothly and monotonically for this example. The simple, Simp-Sym, and Simp-Con methods have almost identical convergence curves and the convergence rate is initially quicker than the other three methods.

The computational times for one linear solve are approximately 6.7 s, 26.9 s, and 27.9 s when using the simple, symmetrized, and consistent tangent stiffness matrices, respectively. So, again, computational time is dominated by the assembly of the tangent stiffness matrix.

Table 3 Double-clamped beam data

Method	Iterations	C (kNm)	No. of solves
Simple	1195	15.28	18,879
Simp-Sym	1500	15.22	24,178
Simp-Con	1383	15.23	22,124
Symmetric	1500	15.83	8007
Sym-Con	1444	15.42	6763
Consistent	1500	15.50	7388

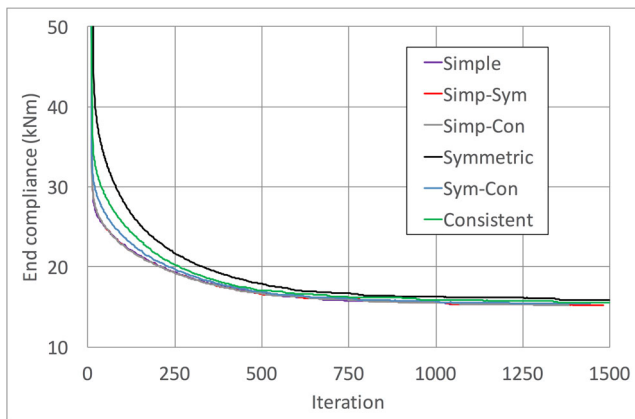


Fig. 6 Double-clamped beam convergence

From Table 3, the simple method requires approximately 2.5 to 2.7 times the number of linear solves, compared with the symmetric, Sym-Con, and consistent methods. However, the cost per solve is about 4 times less. So, the most efficient method is the simplified method, which also produces good optimal solutions and is perhaps the best option for this example.

4.3 Compliant mechanism

Another motivation for including nonlinear geometric effects in topology optimization is the design of compliant mechanisms, as the objective is often to achieve large displacement. The inverter mechanism problem (Fig. 2c) is solved to maximize displacement at the output node in the opposite direction to the applied force, (24). This problem was also studied by Pedersen et al. (2001) using a total Lagrangian approach.

The input force is 0.5 mN and the end compliance constraint is $C^* = 2.5 \times 10^{-9}$ Nm (this can also be stated as a constraint on the input displacement of $5 \mu\text{m}$, as the input force is prescribed) and the volume constraint is equal to 10% of the design domain. The Young's modulus is 180 GPa, Poisson's ratio 0.22, output stiffness $k_{out} = 2$ N/m, and domain thickness $7 \mu\text{m}$. The domain is discretized with 90×45 square plane stress elements and the filter radius is equal to 2 element edge lengths. Note that only the top half of the inverter is modeled.

For this example, it is found that a continuation method is required to obtain reasonable solutions with a clear, near solid-void, topology. The problem is initially solved with $p = 3$ in (18), then p is increased by 0.25 and the problem solved again, using the current solution as the initial guess. This is repeated until $p = 6$. When $p < 6$, the maximum number of optimization iterations is set to 50 and less strict

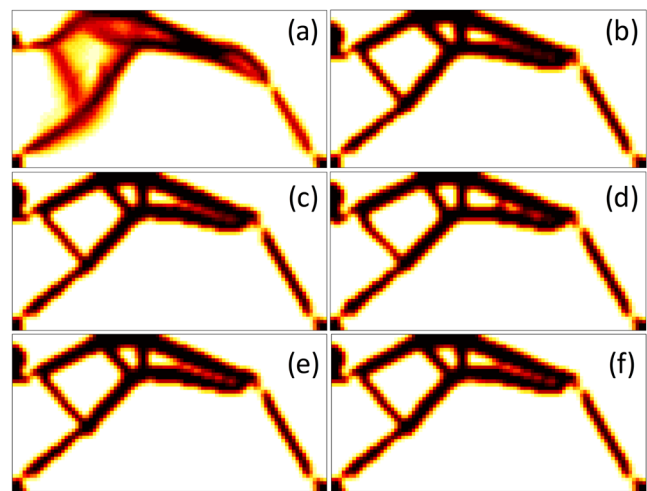


Fig. 7 Inverter mechanism solutions, obtained using **a** simple, **b** Simp-Sym, **c** Simp-Con, **d** symmetric, **e** Sym-Con, and **f** consistent methods

convergence tolerance of 10^{-3} is used. In addition, to speed up convergence to equilibrium, the converged displacement vector from the previous optimization iteration is used as the initial guess, which was also used by Pajot and Maute (2006) when solving an inverter problem. This method was also tried for the previous examples in this paper, with some success in reducing computational cost, but also some negative effects, such as converging to a worse local minima. This should be investigated further, as reusing previous data can significantly improve efficiency, but it does not seem to always work well.

The inverter problem is solved using all six co-rotational methods for nonlinear geometry (Fig. 7), and also with linear modeling (Fig. 8). A summary of optimization iterations, objective values, and number of linear solves is shown in Table 4.

Figure 7 shows that all methods obtain similar solutions, except for the simple method, which contains a significant amount of intermediate “gray” material. The inverter problem is more challenging, as it has a nonlinear constraint that requires the solution of an adjoint to obtain sensitivities. Thus, using the simplified tangent stiffness results in inaccurate sensitivities for both the objective function and compliance constraint. For this example, the optimizer cannot cope with the inaccurate sensitivities, resulting in a poor solution. This is not remedied by changing the problem setup and similar poor solutions are obtained when a different continuation scheme, or filter radius, is used.

It should be noted that the designs obtained have very thin, almost single-node, hinges that make the designs difficult to manufacture directly. This is a known

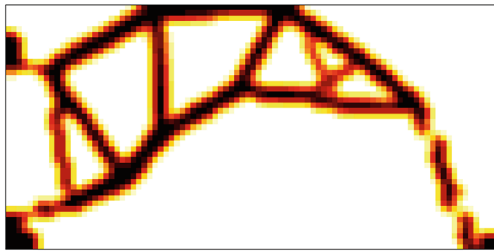


Fig. 8 Inverter mechanism, solution obtained with linear modeling

problem when using density-based topology optimization for compliant mechanism design and various techniques have been proposed to avoid these features, for example the robust formulation (Wang et al. 2011).

When analyzed using nonlinear geometry, the output displacement for the linear solution (Fig. 8) is $10.35 \mu\text{m}$, less than half the value obtained by the nonlinear solutions. Further investigation revealed that the linear solution undergoes stress stiffening when the member connected to the output node becomes more vertical, resulting in a locking of the mechanism, which was also observed by Pedersen et al. (2001). The same members in the nonlinear solution are initially more angled, so the stress stiffening effect is less severe and the mechanism does not lock under large displacement.

The convergence behavior is very similar for all methods (except the simple method). Thus, only the convergence for the consistent method is shown in Fig. 9. The convergence over the first 50 iterations contains several sawtooth-like oscillations. These are caused by the optimizer aggressively trying to improve the objective (maximize output displacement) in the presence of a nonlinear constraint (end compliance, or input displacement), which results in several constraint violations that are remedied by the built-in line search feature of the optimizer

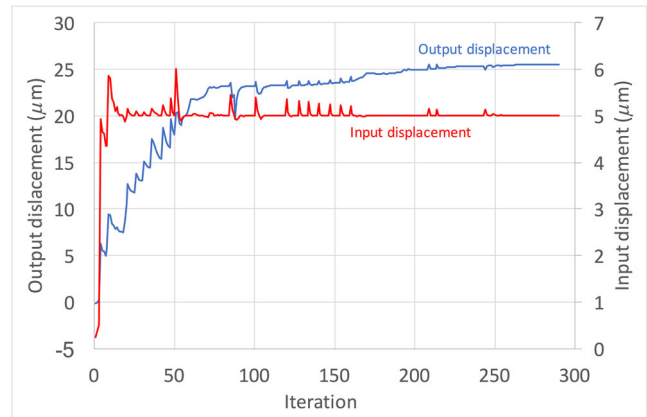


Fig. 9 Inverter mechanism convergence using the consistent method

(hence, the sawtooth-like oscillations). After the first 50 iterations, convergence is smoother and mostly monotonic, except for when the penalization power is increased during the continuation method. Increasing the penalization power leads to an immediate constraint violation, as the mechanism becomes less stiff, although the optimizer is able to find a feasible solution within a few iterations.

For this example (and implementation), the computational times for one linear solve are approximately 0.95 s, 4.95 s, and 5.15 s when using the simple, symmetrized, and consistent tangent stiffness matrices, respectively, again showing that computational time is dominated by the assembly of the tangent stiffness matrix. From Table 4, the number of linear solves for the Simp-Sym and Simp-Con methods is approximately 5 times more than the symmetric, Sym-Con, and consistent methods, whereas the cost per solve is approximately 5 times smaller. Thus, the overall cost for each method (excluding the simple method) is approximately the same.

In summary, for this example and implementation, the simple method is not suitable, as the solution contains a significant amount of intermediate material, caused by inaccurate sensitivities. The other five methods all perform similarly, obtaining similar solutions in approximately the same amount of time.

Table 4 Inverter mechanism data

Method	Iterations	$u_{\text{out}} (\mu\text{m})$	No. of solves
Simple	586	22.66	2928
Simp-Sym	281	25.46	3780
Simp-Con	303	25.48	3592
Symmetric	316	25.51	759
Sym-Con	297	25.50	744
Consistent	290	25.49	728

4.4 Simply supported square plate

The final example is a square plate, simply supported on all edges, with a central point-load acting in the normal direction (Boroomand and Barekatein 2009). The full plate is $2 \times 2 \text{ m}$ square, 3-mm thick, and made from a material with Young's modulus 200 GPa and Poisson's ratio $1/3$. The force is 800 N and the complimentary work is minimized subject to a 30% volume constraint, (23).

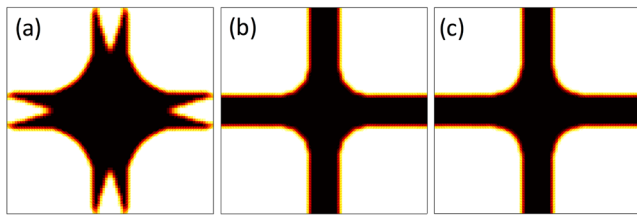


Fig. 10 Simply supported plate: **a** linear solution and nonlinear solutions obtained with **b** simple and **c** consistent methods

Due to symmetry, only one quarter of the plate is modeled, which is discretized using 3200 equal-sized thin, triangular shell elements. The shell elements are composed of a discrete Kirchhoff triangle plate element for bending stiffness and a linear strain triangle (with corner drilling dof, instead of mid-side nodes) for membrane stiffness (Cook et al. 2002). This results in a 3-node element with 6 dof per node. The filter radius is 50 mm, which is twice the shortest element edge length.

The solutions obtained using all six methods are almost identical, so Fig. 10 only shows solutions obtained using the simple and consistent methods, alongside the solution obtained using linear modeling. These solutions are similar to those obtained by Boroomand and Barekatein (2009). Objective values for the nonlinear solutions are within 0.25%, with slightly better values for the methods not using a simplified tangent stiffness matrix, at the cost of more optimization iterations (Table 5), although this is mainly a result of the convergence tolerance used. A more detailed analysis of the computational cost is not performed for this example, as the implementation of the consistent tangent stiffness is not efficient, as it uses finite differences.

The complimentary work for the linear solution is 2.727, when using geometrically nonlinear modeling. Complimentary work values obtained using linear modeling are 51.49, 65.65, and 64.29, for the linear, simple, and consistent solutions, respectively. Thus, complimentary work values are an order of magnitude smaller when the

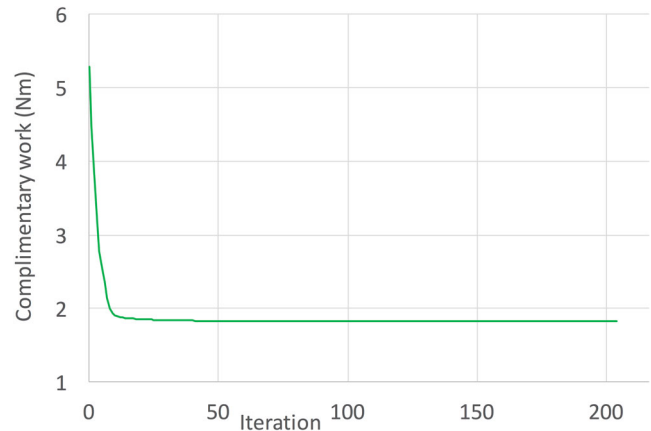


Fig. 11 Plate convergence using the consistent method

solutions are analyzed with nonlinear geometry, compared with linear modeling. This is because linear modeling does not take account of the membrane stiffness, as the plate is initially flat, whereas this is accounted for when including nonlinear geometric effects. Note that, even if out-of-plane displacements are small (compared with the size of the plate), the membrane stiffness effect can be significant. This is why the simple method performed well in this example, as actual displacements are small, so the error in ignoring the variation of the rotation matrix with respect to global displacements is also small. However, it is clear that, even if displacements are small, the geometrically nonlinear membrane effect has a significant affect on the optimal design. In this case, the nonlinear solution is composed of four, almost constant thickness, members that connect the central load to the clamped edges, as this maximizes the benefit of membrane stiffness effect. However, the solution obtained using linear modeling places more material near the load, with less at the boundary, to maximize bending stiffness.

Convergence for all methods is almost identical and the convergence for the consistent method is shown in Fig. 11, which is smooth and monotonic.

Table 5 Plate data

Method	Iterations	CW (Nm)	No. of solves
Simple	91	1.8285	17,230
Simp-Sym	93	1.8284	17,744
Simp-Con	96	1.8284	18,402
Symmetric	205	1.8240	9745
Sym-Con	205	1.8240	9744
Consistent	205	1.8240	8571

5 Conclusions

The co-rotational method is investigated for density-based, geometrically nonlinear topology optimization. Three methods for constructing the tangent stiffness matrix are considered: a simplified definition of the tangent stiffness matrix, which is the assembly of linear stiffness matrices rotated according to the rigid body motion; a consistent definition, which is derived by differentiating the residual force

vector by the global displacement vector; and finally a symmetrized consistent matrix. This leads to six co-rotational schemes for topology optimization, as different tangent stiffness matrices can be used for the solution of nonlinear equilibrium (via the Newton-Raphson method) and adjoint problem (required to obtain sensitivities).

One benefit of the co-rotational method for density-based topology optimization is that the tangent stiffness matrix remains positive definite. Previous studies that used a total Lagrangian approach could suffer from non-convergence of the equilibrium iterations, due to distortion in low-density elements, leading to an indefinite or negative definite tangent stiffness matrix. This issue is naturally avoided when using the co-rotational method for small strain problems.

The six co-rotational schemes are tested using four benchmark examples from the literature, where it has been shown that geometric nonlinearity significantly affects the optimal design. These include 2D stiffness maximization problems, a compliant mechanism, and a plate problem (modeled using 3-node thin shell elements). In all cases, the co-rotational method is able to obtain solutions that are comparable with those obtained in the literature using a total Lagrangian approach, although there are differences in the performance of the six considered schemes, in terms of final design, objective value, and computational cost.

In all examples, the methods involving the symmetrized tangent stiffness did not outperform any of the other methods. This is because the computational cost is dominated by assembling the tangent stiffness matrix, for the implementation and examples in this paper. When compared against the consistent method, the symmetrized method does not provide significant reduction in computational cost, while also being less accurate. Comparing the simplified method with the consistent method, the results show that the best choice is problem dependent. If the problem has large displacements (e.g., the cantilever and inverter mechanism), then it is recommended to use the consistent method, as the accurate tangent stiffness allows faster convergence of the nonlinear equilibrium and the error in sensitivities caused by the simplified method can lead to a poor design (especially for compliant mechanisms). On the other hand, if the optimal design has relatively small displacements, then the error in the simplified method is small and designs comparable with those obtained using the consistent method are obtained with less computational effort. Note that, even if displacements are small, then geometric nonlinearity can still significantly affect the optimal solution, for example, by avoiding instability or accounting for membrane stiffness in shell problems.

Acknowledgments The author would like to thank the Numerical Analysis Group at the Rutherford Appleton Laboratory for their FORTRAN HSL packages (HSL, a collection of Fortran codes for large-scale scientific computation; see <http://www.hsl.rl.ac.uk/>).

Compliance with ethical standards

Conflict of interest The author declares no conflict of interest.

Replication of results The results for the 2D problems in Sections 4.1 to 4.3 can be replicated by combining the Matlab code in Appendix A with the 88-line code developed by Andreassen et al. (2011) and the NLOpt package (Johnson 2018) (as detailed in Appendix A). The results for the plate problem in Section 4.4 were generated by a more complex home-grown code, although all necessary details are included in the paper and solution files can be obtained by contacting the author.

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/>.

Appendix A: Matlab code for simplified co-rotational method

The following 116 lines of Matlab code can be used to modify the 88-line topology optimization code created by Andreassen et al. (2011) to solve 2D geometrically nonlinear problems using the co-rotational method detailed in Section 2. The first 74 lines replace lines 53 to 60 in the 88-line code and the remaining lines (from 75 onwards) make a function that can be added to the end of the code. This makes the whole code 196 lines long. The code can use any of the six co-rotational schemes discussed in this paper by changing the “method” in line 2 (see Table 1). The code also requires another input for the element edge length, h . Note that the results in this paper are obtained by linking this code with the optimization package NLOpt and not by using the optimality criteria method in the original 88-line code. Also, it is not claimed that the code is efficient, in terms of computational time or number of lines.

```

%% FE-ANALYSIS
method = 6; % select co-rotational method (see Table 1)
if method < 4, exact = 0; else exact = 1; end
U = zeros(2*(nely+1)*(nelx+1),1);
Fint = zeros(nely*nelx,8); % data storage for internal element forces
load_fac = 0; load_inc = 1; % initial load increment controls
% Newton-Raphson loop
while load_fac < 1
    load_fac = min(load_fac + load_inc,1); % increase load factor (max = 1)
    if load_inc < 0.01, load_fac = 1; load_inc = 0; end % force load_fac = 1
    if load_fac < 1, tol = 5e-4; else, tol = 5e-6; end % set convergence tol
    U_trial = U; % starting disp for this increment
    nl_res = 1; nl_res_old = 1;
    relax = 1; % relaxation factor
    nl_loop = 0;
while 1
    nl_loop = nl_loop + 1;
    Fres = F*load_fac; % initialise residual force vector
    for i=1:(nelx*nely)
        xdef = U_trial(edofMat(i,:)) + [0;0;h;0;h;h;0;h]; % deformed coords
        % Assemble element tangent stiffness into global
        [KT,R] = Ktangent(h,xdef,KE,exact);
        sK(((i-1)*64)+1:i*64) = (Emin+xPhys(i)'.^penal*(E0-Emin))*reshape(KT,64,1);
        % Compute and assemble internal force vector
        Fint(i,:) = R'*KE*(R*xdef - [0;0;h;0;h;h;0;h]);
        Fres(edofMat(i,:)) = Fres(edofMat(i,:))-(Emin+xPhys(i)'.^penal*(E0-Emin))*Fint(i,:)';
    end
    % check convergence, or divergence
    nl_res = norm(Fres(freedofs),Inf) / (norm(F)*load_fac);
    if nl_res < tol
        U = U_trial;
        % Switch for Simple-Symmetric & Simple-Consistent methods
        if(method == 2 || method == 3)
            if load_fac < 1, break; % only switch when load_fac = 1
            elseif exact==1, break; % already switched
            else, exact=1; % switch to consistent KT for adjoint
            end
        else
            break;
        end
    end
    % Check for divergence
    elseif(nl_res_old < nl_res && nl_loop > 2)
        relax = 0.25*relax;
        % reduce load increment
        if relax < 0.01
            load_fac = load_fac - load_inc; % reset
            load_inc = load_inc*0.25;
            if load_inc < 1e-6, U = U_trial; end % if non-conv at load_fac=1
            break;
        end
        % Otherwise, continue line search - under relaxation
        U_trial(freedofs) = U_trial(freedofs) - 0.75*delU;
        delU = 0.25*delU;
    % if not converged and not diverging, then solve for delU
    else
        K = sparse(iK,jK,sK);
        if method ~= 6, K = (K+K')/2; end % make K symmetric
        delU = K(freedofs, freedofs)\Fres(freedofs);
        U_trial(freedofs) = U_trial(freedofs) + delU; % displacement increment
        nl_res_old = nl_res;
        relax = 1;
    end
end
end
end

```

```

%% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
c = U'*F;
% Solve adjoint problem
K = sparse(iK,jK,sK);
if(method==1 || method==2 || method==4), K = (K+K')/2; end % make K symmetric
P = zeros(2*(nely+1)*(nelx+1),1);
P(freedofs) = K(freedofs,freedofs)\F(freedofs);
% Compute sensitivities (adjoint x internal forces)
ce = reshape(sum(Fint.*P(edofMat),2),nely,nelx);
dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
%% ROTATION MATRIX & TANGENT STIFFNESS MATRIX
function [KT,R] = Ktangent(h,xdef,KE,exact)
% useful values to simplify equations (note, xdef is scaled by h for R)
x51 = (xdef(5)-xdef(1))/h; x62 = (xdef(6)-xdef(2))/h; l31 = (x51^2 + x62^2)^(1/2);
x73 = (xdef(7)-xdef(3))/h; x84 = (xdef(8)-xdef(4))/h; l42 = (x73^2 + x84^2)^(1/2);
% Compute rotation matrix using bisector coordinate axes
e31 = [x51,x62]/l31; e42 = [x73,x84]/l42;
ex = e31 - e42; ex = ex/(ex(1)^2+ex(2)^2)^(1/2);
ey = e31 + e42; ey = ey/(ey(1)^2+ey(2)^2)^(1/2);
R1 = [ex;ey]; R = blkdiag(R1,R1,R1,R1); % expand to 8 x 8 rotation matrix
% Simplified part of tangent stiffness matrix
KT = R'*KE*R;
if(exact)
% dR/du is a third order tensor, assemble using directional derivatives
dRdu = zeros(2,2,8);
denom = ((x51/l31 - x73/l42)^2 + (x62/l31 - x84/l42)^2)^(3/2);
dn12 = l31^4*l42^2*denom; dn34 = l31^2*l42^4*denom;
sn = sign(R1(1,1)*R1(2,2));
% dR/du5 & dR/du1
dx51 = [-(l42*x62 - l31*x84)*(x84*l31^2 - l42*x62*l31 - x84*x51^2 + x62*x73*x51), ...
((l42*x51 - l31*x73)*(x84*l31^2 - l42*x62*l31 - x84*x51^2 + x62*x73*x51))]/dn12;
dRdu(:, :, 5) = [dx51(1),dx51(2);-sn*dx51(2),sn*dx51(1)]; dRdu(:, :, 1) = -dRdu(:, :, 5);
% dR/du6 & dR/du2
dx62 = [((l42*x62 - l31*x84)*(x73*l31^2 - l42*x51*l31 - x73*x62^2 + x51*x84*x62)), ...
-(l42*x51 - l31*x73)*(x73*l31^2 - l42*x51*l31 - x73*x62^2 + x51*x84*x62)]/dn12;
dRdu(:, :, 6) = [dx62(1),dx62(2);-sn*dx62(2),sn*dx62(1)]; dRdu(:, :, 2) = -dRdu(:, :, 6);
% dR/du7 & dR/du3
dx73 = [-(l42*x62 - l31*x84)*(x62*l42^2 - l31*x84*l42 - x62*x73^2 + x51*x84*x73), ...
((l42*x51 - l31*x73)*(x62*l42^2 - l31*x84*l42 - x62*x73^2 + x51*x84*x73))]/dn34;
dRdu(:, :, 7) = [dx73(1),dx73(2);-sn*dx73(2),sn*dx73(1)]; dRdu(:, :, 3) = -dRdu(:, :, 7);
% dR/du8 & dR/du4
dx84 = [((l42*x62 - l31*x84)*(x51*l42^2 - l31*x73*l42 - x51*x84^2 + x62*x73*x84)), ...
-(l42*x51 - l31*x73)*(x51*l42^2 - l31*x73*l42 - x51*x84^2 + x62*x73*x84)]/dn34;
dRdu(:, :, 8) = [dx84(1),dx84(2);-sn*dx84(2),sn*dx84(1)]; dRdu(:, :, 4) = -dRdu(:, :, 8);
% Tangent stiffness (part due to dR/du)
for i=1:8
dR = blkdiag(dRdu(:, :, i),dRdu(:, :, i),dRdu(:, :, i),dRdu(:, :, i));
N = dR'*KE; M = N*R;
KT(:, i) = KT(:, i) + (1/h)*(M + M')*xdef - N*[0;0;1;0;1;0;1;1];
end
end
end

```

Appendix B: Algorithm for extracting rotations from a rotation matrix

Given a 3D rotation matrix \mathbf{R} , the following algorithm is used to extract the unknown rotations required to produce that matrix (Crisfield 1990). The algorithm involves the

eigenvalues of \mathbf{R} to extract the quaternions, which are then scaled to obtain the actual rotations.

$$a = \max(\text{Tr}(\mathbf{R}), R_{11}, R_{22}, R_{33}) \quad (35)$$

If $a = \text{Tr}(\mathbf{R})$, then:

$$\bar{q} = 0.5(1 + a)^{\frac{1}{2}} \quad (36)$$

$$q_i = 0.25(R_{kj} - R_{jk})/\bar{q} \quad (37)$$

with i, j, k the cyclic permutation of 1, 2, 3.

Otherwise, if $a = R_{ii}$, then:

$$q_i = [0.5a + 0.25(1 - \text{Tr}(\mathbf{R}))]^{1/2} \quad (38)$$

$$\bar{q} = 0.25(R_{kj} - R_{jk})/q_i \quad (39)$$

$$q_l = 0.25(R_{li} - R_{il})/q_i, \text{ for } l = j, k \quad (40)$$

For rotations less than 180° , the actual rotations, θ_i , are obtained by:

$$\bar{\theta} = \arccos \bar{q} \quad (41)$$

$$\theta_i = [(2\bar{\theta})/(\bar{q} \tan \bar{\theta})]q_i \quad (42)$$

References

- Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in Matlab using 88 lines of code. *Struct Multidiscip Optim* 43(1):1–16
- Argyris J (1982) An excursion into large rotations. *Comput Methods Appl Mech Eng* 32(1–3):85–155
- Belytschko T, Hsieh B (1973) Non-linear transient finite element analysis with convected co-ordinates. *Int J Numer Methods Eng* 7(3):255–271
- Boroomand B, Barekatein A (2009) On topology optimization of linear and nonlinear plate problems. *Struct Multidiscip Optim* 39(1):17–27
- Bourdin B (2001) Filters in topology optimization. *Int J Numer Methods Eng* 50(9):2143–2158
- Bruns T, Sigmund O (2004) Toward the topology design of mechanisms that exhibit snap-through behavior. *Comput Methods Appl Mech Eng* 193(36–38):3973–4000
- Bruns TE, Tortorelli DA (2001) Topology optimization of non-linear elastic structures and compliant mechanisms. *Comput Methods Appl Mech Eng* 190(26–27):3443–3459
- Bruns TE, Tortorelli DA (2003) An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. *Int J Numer Methods Eng* 57(10):1413–1430
- Bruns T, Sigmund O, Tortorelli DA (2002) Numerical methods for the topology optimization of structures that exhibit snap-through. *Int J Numer Methods Eng* 55(10):1215–1237
- Buhl T, Pedersen CB, Sigmund O (2000) Stiffness design of geometrically nonlinear structures using topology optimization. *Struct Multidiscip Optim* 19(2):93–104
- Carstensen JV, Lotfi R, Guest JK, Chen W, Schroers J (2016) Topology optimization of cellular materials with maximized energy absorption. In: ASME 2015 International design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers Digital Collection
- Cho S, Kwak J (2006) Topology design optimization of geometrically non-linear structures using meshfree method. *Comput Methods Appl Mech Eng* 195(44–47):5909–5925
- Cook RD, Malkus DS, Plesha ME, Witt RJ (2002) Concepts and applications of finite element analysis, 4th edn. Wiley
- Crisfield MA (1990) A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. *Comput Methods in Appl Mech Eng* 81(2):131–150
- Gea HC, Luo J (2001) Topology optimization of structures with geometrical nonlinearities. *Comput Struct* 79(20–21):1977–1985
- Gomes FA, Senne TA (2014) An algorithm for the topology optimization of geometrically nonlinear structures. *Int J Numer Methods Eng* 99(6):391–409
- He Q, Kang Z, Wang Y (2014) A topology optimization method for geometrically nonlinear structures with meshless analysis and independent density field interpolation. *Comput Mech* 54(3):629–644
- Izzuddin B (2005) An enhanced co-rotational approach for large displacement analysis of plates. *Int J Numer Methods Eng* 64(10):1350–1374
- James KA, Waisman H (2016) Layout design of a bi-stable cardiovascular stent using topology optimization. *Comput Methods Appl Mech Eng* 305:869–890
- Jog C (1996) Distributed-parameter optimization and topology design for non-linear thermoelasticity. *Comput Methods Appl Mech Eng* 132(1–2):117–134
- Johnson SG (2018) The NLOpt nonlinear optimization package. Version 2.5.0. <http://github.com/stevengj/nlopt>
- Jung D, Gea HC (2004) Topology optimization of nonlinear structures. *Finite Elem Anal Des* 40(11):1417–1427
- Kawamoto A (2009) Stabilization of geometrically nonlinear topology optimization by the levenberg–marquardt method. *Struct Multidiscip Optim* 37(4):429–433
- Kemmler R, Lipka A, Ramm E (2005) Large deformations and stability in topology optimization. *Struct Multidiscip Optim* 30(6):459–476
- Lahuerta RD, Simões ET, Campello EM, Pimenta PM, Silva EC (2013) Towards the stabilization of the low density elements in topology optimization with large deformation. *Comput Mech* 52(4):779–797
- Lindgaard E, Dahl J (2013) On compliance and buckling objective functions in topology optimization of snap-through problems. *Struct Multidiscip Optim* 47(3):409–421
- Luo Z, Tong L (2008) A level set method for shape and topology optimization of large-displacement compliant mechanisms. *Int J Numer Methods Eng* 76(6):862–892
- Nakshatrala PB, Tortorelli DA, Nakshatrala K (2013) Nonlinear structural design using multiscale topology optimization. Part i: static formulation. *Comput Methods Appl Mech Eng* 261:167–176
- Pajot JM, Maute K (2006) Analytical sensitivity analysis of geometrically nonlinear structures based on the co-rotational finite element method. *Finite Elem Anal Des* 42(10):900–913
- Pedersen CB, Buhl T, Sigmund O (2001) Topology synthesis of large-displacement compliant mechanisms. *Int J Numer Methods Eng* 50(12):2683–2705
- Rankin C, Nour-Omid B (1988) The use of projectors to improve finite element performance. *Comput Struct* 30(1–2):257–267
- Sigmund O (2007) Morphology-based black and white filters for topology optimization. *Struct Multidiscip Optim* 33(4–5):401–424
- Spurrer RA (1978) Comment on "singularity-free extraction of a quaternion from a direction-cosine matrix". *J Spacecraft Rockets* 15(4):255–255
- Stanford B, Beran P (2013) Optimal thickness distributions of aeroelastic flapping shells. *Aerospace Sci Technol* 24(1):116–127

- Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J Optim* 12(2):555–573
- Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. *Struct Multidiscip Optim* 43(6):767–784
- Wang F, Lazarov BS, Sigmund O, Jensen JS (2014) Interpolation scheme for fictitious domain techniques and topology optimization of finite strain elastic problems. *Comput Methods Appl Mech Eng* 276:453–472
- Yoon GH (2010) Maximizing the fundamental eigenfrequency of geometrically nonlinear structures by topology optimization based on element connectivity parameterization. *Comput Struct* 88(1–2):120–133
- Yoon GH, Kim YY (2005) Element connectivity parameterization for topology optimization of geometrically nonlinear structures. *Int J Solids Struct* 42(7):1983–2009

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