An efficient 146-line 3D sensitivity analysis code of stress-based topology optimization written in MATLAB

Hao Deng¹ · Praveen S. Vulimiri¹ · Albert C. To¹

Received: 23 April 2021 / Revised: 13 July 2021 / Accepted: 10 August 2021 / Published online: 26 August 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
This paper presents an efficient and compact MATLAB code for three-dimensional stress-based sensitivity analysis. The 146 lines code includes the finite element analysis and p-norm stress sensitivity analysis based on the adjoint method. The 3D sensitivity analysis for p-norm global stress measure is derived and explained in detail accompanied by corresponding MATLAB code. The correctness of the analytical sensitivity is verified by comparison with finite difference approximation. The nonlinear optimization solver is chosen as the Method of moving asymptotes (MMA). Three typical volume-constrained stress minimization problems are presented to verify the effectiveness of sensitivity analysis code. The MATLAB code presented in this paper can be extended to resolve different stress related 3D topology optimization problems. The complete program for sensitivity analysis is given in the Appendix and is intended for educational purposes. MATLAB code is additionally provided in electronic supplementary material for a simple cantilever beam optimization.

Keywords Topology optimization · Stress sensitivity · SIMP · Educational code

1 Introduction

As computational resources and advanced manufacturing techniques have improved, structural optimization tools such as topology optimization (TO) now contribute a greater role in the design process. TO, first introduced by Bendsoe and Kikuchi 1988, determines the ideal values for design variables representing the
material distribution in a geometry for an objective, typically for stiffness. TO
designs optimized for stiffness can produce complex, organic, and lightweight
designs attractive in industrial applications. However, these designs are prone to
introduce stress concentrations which could cause the design to fail in use. It is
therefore critical to include stress constraints in TO for industrial applications,
ensuring the optimal design satisfies all engineering requirements without major
modifications deviating from the original optimized result. As noted by Le et al.
2010, there are three challenges when implementing stress constraints which have
impeded development: the “singularity” phenomenon (Duysinx and Bendsøe 1998),
stress as a local result, and the non-linearity of stress.

As discussed in many papers (Guo et al. 2011; Paris et al. 2010; Cheng and Guo
1997; Kirsch 1990; Luo et al. 2013; Xia et al. 2012, 2018; Zhang et al. 2013; Fan
et al. 2019; Deng et al. 2020a, 2019, 2020b; Deng and To 2020), the singularity
phenomenon occurs in design variables with low densities still exhibiting a strain
value, resulting in an artificially large stress values and complicating the
optimization process. Different approaches have been developed to overcome this
challenge. Cheng and Guo 1997 use a strain relaxation approach to alleviate the
strain in these low-density elements. Bruggi 2012, Le et al. 2010, and Holmberg
2013 penalize the stress based on the density variable. All three of these works use
the Solid Isotropic Material with Penalization (SIMP) method which represents each
density variable as a continuous value between 1 and 0. Xia et al. 2018 avoids the
singularity problem by using the Bi-Directional Evolutionary Structural Optimiza-
tion (BESO) method. The BESO method represents each variable discretely as 1 or
0 to represent solid or void material, removing intermediate values and the
singularity the intermediate values present. It is also noted the level set approach,
another method for TO, has solved stress constrained TO problems as well (Guo
et al. 2011; Xia et al. 2012, 2014; Zhang et al. 2013; Wang and Li 2013; Allaire and
Jouve 2008; Suresh and Takaloozadeh 2013; James et al. 2012; Miegroet and
Duysinx 2007; Allaire et al. 2014; Picelli et al. 2018a, 2018b; Emmendoerfer and
Fancell 2016; Kambampati et al. 2020; Song et al. 2020; Yang et al. 2018). The
level set method also represents material discretely as found in BESO and avoids the
singularity phenomenon. Additional advanced methods for stress related topology
optimization can be found in Ref (Zhang et al. 2017, 2016, 2018; Cai and Zhang
2015). This paper, however, will focus on implementing the SIMP method for
stress-based topology optimization. The second challenge lies in the fact stress is
calculated locally at each element. As noted by Duysinx and Bendsoe (Duysinx and
Bendsøe 1998), this then requires a constraint for each element and increases the
necessary computational resources required to compute the problem. Alternatively,
Duysinx and Sigmund (Duysinx and Sigmund 1998) introduced a global stress
constraint, grouping all local stresses into one measure. However, the global
constraint does not provide great control of the local stress behavior. Instead,
regional approaches such as those introduced by Paris et al. 2009, Le et al. 2010,
and Holmberg et al. 2013 are preferred. These approaches group elemental
constraints based on nearby elements or based on the sorted stress values. It was
found the regional approaches are a satisfactory compromise, limiting the number of
constraints but still providing good local stress control. The third challenge is
presented in the nonlinear behavior of stress. Elemental stress measures are greatly affected by the densities in the local neighborhood of the element. As such, a density variable filter (Le et al. 2010) is used to smooth the density variables and sensitivities by taking a weighted average of neighboring nodes. This removes the checkerboard effect present in earlier topology optimization algorithms and produces smooth designs and better convergence to the global minimum. Most recently, Senhora et al. 2020 presents a consistent topology optimization formulation for mass minimization with local stress constraints by means of the augmented Lagrangian method, named as aggregation-free approach.

In recent years, several computational programs for topology optimization have been published for educational purposes. These codes are helpful for students and engineers to understand the basic mathematical formulation of topology optimization. The detailed reviews for these programs of TO can be found in Ref (Zhu 2020). Table 1 summarizes the codes published in recent years, including the three major methods: SIMP (Bendsøe and Sigmund 1999), level set (Wang et al. 2003), and

| Authors and Reference | Programming language | Method |
|-----------------------|----------------------|--------|
| (2001) O. Sigmund     | MATLAB               | SIMP   |
| (2005) Liu Z, Korvink J G et al. | FEMLAB          | Level set |
| (2010) Challis V J    | MATLAB               | Level set |
| (2010) Suresh, Krishnan | MATLAB            | SIMP (Pareto) |
| (2010) Huang X, Xie Y M. | MATLAB           | BESO   |
| (2011) Andreassen, Erik, et al. | MATLAB         | SIMP   |
| (2012) Talischi, Cameron, et al. | MATLAB     | PolyTop |
| (2014) Zegard T, Paulino G H | MATLAB  | Ground Structure |
| (2015) Aage, Niels, et al. | PETSc            | SIMP   |
| (2015) Otomori, Masaki, et al. | MATLAB         | Level set |
| (2015) Xia L, Breitkopf P | MATLAB         | SIMP   |
| (2016) Pereira, Anderson, et al. | MATLAB    | PolyTop |
| (2018) Wei, Peng, et al. | MATLAB        | Level set |
| (2018) Loyola, Rubén Ansola, et al. | MATLAB    | SERA   |
| (2018) Laurain, Antoine. | FEniCS       | Level set |
| (2018) Sanders, Emily D., et al. | MATLAB   | PolyMat |
| (2018) Dapogny, Charles, et al. | FreeFem + + | Shape variation |
| (2019) Chen Q, Zhang X, Zhu B. | MATLAB, APDL | SIMP   |
| (2019) Gao, Jie, et al. | MATLAB       | SIMP   |
| (2020) Liang Y, Cheng G. | MATLAB       | Integer programming |
| (2020) Smith H, Norato J A. | MATLAB     | Geometry projection |
| (2020) Picelli R, et al. | MATLAB     | TOBS   |
| (2020) Lin H, Xu A, Misra A, et al. | APDL        | DER-BESO |
| (2005) Ferrari F, Sigmund O. | MATLAB     | SIMP   |
This paper aims to educate the reader to implement stress sensitivity analysis of TO. The code provided in this paper is designed to be easy-to-understand and will discuss the implementation for stress-related problems. As stress related TO is a nonlinear optimization, the problems presented are solved using the Method of Moving Asymptotes (MMA) (Svanberg 1987).

The remainder of the paper is organized as follows. Section 2 presents the formulation of the p-norm global stress measure. Section 3 describes the three-dimensional finite element method (FEM) formulation and numerical implementation for eight-node hexahedron elements. Section 4 describes the sensitivity analysis and MATLAB implementation. Section 5 formulates in detail the volume-constrained stress minimization problem. Section 6 demonstrates three typical stress related topology optimization problems to verify the effectiveness of sensitivity analysis code, followed by conclusions in Sect. 7.

2 The stress-based topology optimization problem

2.1 Stress and stiffness penalization

The design domain is discretized with the eight-node hexahedron elements, and each element is assigned with a density variable. The design variable can be written as $x = (x_1, x_2, \ldots, x_{nele})$, where $nele$ is total number of elements. The design variable $x$ for the SIMP method is constrained within $[0, 1]$, where 0 corresponds to void material and 1 to solid material. To obtain a black-and-white design, a penalization function is introduced to penalize the intermediate density. For stress-based topology optimization problem, the penalization of stress and stiffness for intermediate design variable values is described in Ref (Le et al. 2010; Holmberg et al. 2013). The element stiffness can be expressed based on SIMP penalization function as follows,

$$D_i = x_i^{pl} D_0$$

where $D_0$ denotes the stiffness of solid material. $pl$ is a penalization factor, which is set to $pl = 3$ in general (Andreassen et al. 2011). The effective stress vector (artificial measurement) is given as follows,

$$\sigma_i = D_0 B_i u_i$$

where $u_i$ denotes the displacement vector at nodes of the $i$th element, and $B_i$ is strain matrix of $i$th element. Note that the stress vector $\sigma_i$ in Voigt notation is given as,

$$\sigma_i = (\sigma_{ix}, \sigma_{iy}, \sigma_{iz}, \sigma_{iyy}, \sigma_{iyz}, \sigma_{izx})^T$$

The penalized or relaxed stress measure $\tilde{\sigma}_i(x_i)$ is expressed as,

$$\tilde{\sigma}_i(x_i) = \eta(x_i) \sigma_i$$
Several different penalization schemes have been proposed in recent years (Le et al. 2010). A general stress penalization scheme is given by,

$$\eta(x_i) = (x_i)^q$$

(5)

where \(q\) is a non-negative stress relaxation parameter. As mentioned by Holmberg et al. 2013, this non-physical penalization scheme is such that the \(\tilde{\sigma}_i\) equals \(\sigma_i\) for solid material and,

$$\lim_{x_i \to 0} \tilde{\sigma}_i(x_i) = 0$$

(6)

Following this scheme alleviates the singularity phenomenon discussed in the introduction (Kočvara and Stingl 2012). The definition of von Mises stress can be written as follows,

$$\sigma_{vm,i} = \left( \sigma_{ix}^2 + \sigma_{iy}^2 + \sigma_{iz}^2 - \sigma_{ix} \sigma_{iy} - \sigma_{ix} \sigma_{iz} - \sigma_{iy} \sigma_{iz} + 3\tau_{ixy}^2 + 3\tau_{iyz}^2 + 3\tau_{izx}^2 \right)^{1/2}$$

(7)

### 2.2 Global p-norm stress measure

The standard p-norm global stress measure is applied for approximating the maximum stress as follows,

$$\sigma_{PN} = \left( \sum_{i=1}^{nele} \sigma_{vm,i}^p \right)^{1/p}$$

(8)

where \(\sigma_{vm,i}\) denotes the von Mises stress at the centroid of the \(i\)th element, and \(p\) is the p-norm aggregation parameter. It is worth to note that the p-norm value approaches the maximum value of \(\sigma_{vm}\) when \(p \to \infty\), and that

$$\max \tilde{\sigma}_{vm} \leq \left( \sum_{i=1}^{nele} \sigma_{vm,i}^p \right)^{\frac{1}{p}}$$

(9)

In general, a greater value of \(p\) can provide a more accurate approximation of the maximum von-Mises stress. However, the problem may become ill-conditioned and cause severe oscillations in the optimization process if the value of \(p\) is too great (\(p > 30\)). Thus, an appropriate p-norm value should be selected such that the convergence history is smooth and the maximum stress approximation is sufficient.

### 3 Finite element analysis and MATLAB implementation

The 3D finite element formulation in this paper is based on the code provided by Liu et al. 2014, where more details regarding efficient numerical implementation are provided. For 3D problems, the constitutive matrix \(D_0\) with unit elastic modulus is given by
where \( \nu \) is the Poisson’s ratio of the isotropic material. Based on finite element method, the stiffness of a linear elastic element for solid material can be formulated as follows,

\[
D_0 = \frac{1}{(1+\nu)(1-2\nu)} \begin{bmatrix}
1 - \nu & \nu & \nu & 0 & 0 & 0 \\
\nu & 1 - \nu & \nu & 0 & 0 & 0 \\
\nu & \nu & 1 - \nu & 0 & 0 & 0 \\
0 & 0 & 0 & (1-2\nu)/2 & 0 & 0 \\
0 & 0 & 0 & 0 & (1-2\nu)/2 & 0 \\
0 & 0 & 0 & 0 & 0 & (1-2\nu)/2
\end{bmatrix}
\]

(10)

where \( \nu \) is the Poisson’s ratio of the isotropic material. Based on finite element method, the stiffness of a linear elastic element for solid material can be formulated as follows,

\[
K_0 = \int \int \int B^T D_0 B d\xi_1 d\xi_2 d\xi_3
\]

(11)

where \( d\xi_i \) denotes the natural coordinates. Note that the Jacobian matrix is ignored here due to the unit element length. A detailed description can be found in Liu et al. 2014. Note that \( B \) is strain–displacement matrix, where detailed mathematical formulation and MATLAB implementation for an eight-node hexahedral element can be found in Gao et al. 2019. In this paper, the MATLAB implementation of the element stiffness matrix \( K_0 \) and the strain–displacement matrix \( B \) can be done by lines 74–145. The global stiffness \( K \) can be obtained by assembling the element stiffness as follows,

\[
K = \sum_{i=1}^{nele} K_i(x_i) = \sum_{i=1}^{nele} E(x_i) K_0
\]

(12)

where \( nele \) is total number of elements, and the interpolation coefficient \( E(x_i) \) is defined as,

\[
E(x_i) = \left( E_m + x_i^p l (E_0 - E_m) \right)
\]

(13)

where \( pl \) is a penalty parameter, and \( E_0 \) and \( E_m \) are chosen as: \( E_0 = 1, E_m = 1 \times 10^9 \). Finally, the nodal displacement from the FEM can be computed through solving the equilibrium equation

\[
KU = F
\]

(14)

where \( U \) is the nodal displacement vector, and \( F \) is the vector of external loading, which is independent from the design variable. The node ID system is the same as Liu et al. 2014, as shown in Fig. 1.

The connectivity matrix edofMat is formulated in lines 20–27, which is used for assembling the global stiffness \( K \). The element stiffness matrix and strain–displacement matrix are obtained by line 3. The global stiffness matrix is assembled in lines 28–31 using vectorized MATLAB code. The solution of the nodal displacement vector \( U \) is obtained by solving the sparse linear system in line 67. Note that the freedofs and fixeddofs denote the unconstrained and constrained DOFs, respectively. The code example below is for a cantilever beam problem as
shown in Fig. 2, where the left side is fixed and an evenly distributed force is applied at the bottom right line. For such boundary conditions, the freedofs and fixeddofs are defined in lines 13, 14 and 19. The user can change the boundary and loading conditions by changing the corresponding node IDs.

4 Sensitivity analysis and MATLAB implementation

4.1 Sensitivity analysis

To perform the stress-based topology optimization, the sensitivity of the global stress measure with respect to elemental density needs to be provided, a core component of this paper. Note that the total number of elements and nodes are nele
and \( ndof \), respectively. The derivative of \( \sigma_{PN} \) with respect to \( j \)th design variable \( x_j \) is as follows,

\[
\frac{\partial \sigma_{PN}}{\partial x_j} = \sum_{i=1}^{nele} \left( \frac{\partial \sigma_{PN}}{\partial \hat{\sigma}_{vm,i}} \right) \left( \frac{\partial \hat{\sigma}_{vm,i}}{\partial \hat{\sigma}_i} \right)^T \frac{\partial \eta(x_i)}{\partial \hat{\sigma}_i} \frac{\partial \hat{\sigma}_i}{\partial x_j} + \frac{\partial \sigma_{PN}}{\partial \hat{\sigma}_{vm,i}} \left( \frac{\partial \eta(x_i)}{\partial \hat{\sigma}_i} \right) \frac{\partial \hat{\sigma}_i}{\partial x_j}
\]

The equations above can be rewritten in the following form,

\[
\frac{\partial \sigma_{PN}}{\partial x_j} = T_1 + T_2
\]

\[
T_1 = \sum_{i=1}^{nele} \left( \frac{\partial \sigma_{PN}}{\partial \hat{\sigma}_{vm,i}} \right) \left( \frac{\partial \hat{\sigma}_{vm,i}}{\partial \hat{\sigma}_i} \right)^T \frac{\partial \eta(x_i)}{\partial \hat{\sigma}_i} \frac{\partial \hat{\sigma}_i}{\partial x_j}
\]

\[
T_2 = \sum_{i=1}^{nele} \frac{\partial \sigma_{PN}}{\partial \hat{\sigma}_{vm,i}} \left( \frac{\partial \eta(x_i)}{\partial \hat{\sigma}_i} \right) \frac{\partial \hat{\sigma}_i}{\partial x_j}
\]

The term \( \frac{\partial \sigma_{PN}}{\partial \hat{\sigma}_{vm,i}} \) can be expressed as follows,

\[
\frac{\partial \sigma_{PN}}{\partial \hat{\sigma}_{vm,i}} = \left( \sum_{i=1}^{nele} (\hat{\sigma}_{vm,i})^p \right)^{1-p} (\hat{\sigma}_{vm,i})^{p-1}
\]

Based on the von Mises definition defined in Eq. (7), the derivative of the local element von Mises stress \( \hat{\sigma}_{vm,i} \) with respect to the stress vector \( \hat{\sigma}_i \) can be written as follows,

\[
\frac{\partial \hat{\sigma}_{vm,i}}{\partial \hat{\sigma}_i} = \begin{cases} 
\frac{1}{2\hat{\sigma}_{vm,i}} (2\hat{\sigma}_{ix} - \hat{\sigma}_{iy} - \hat{\sigma}_{iz}) \\
\frac{1}{2\hat{\sigma}_{vm,i}} (2\hat{\sigma}_{iy} - \hat{\sigma}_{ix} - \hat{\sigma}_{iz}) \\
\frac{1}{2\hat{\sigma}_{vm,i}} (2\hat{\sigma}_{iz} - \hat{\sigma}_{ix} - \hat{\sigma}_{iy}) \\
\frac{3}{\hat{\sigma}_{vm,i}} \hat{\tau}_{ixy} \\
\frac{3}{\hat{\sigma}_{vm,i}} \hat{\tau}_{ixy} \\
\frac{3}{\hat{\sigma}_{vm,i}} \hat{\tau}_{ixy} \\
\frac{3}{\hat{\sigma}_{vm,i}} \hat{\tau}_{ixy}
\end{cases}
\]

The derivative \( \eta(x_i) \) with respect to design variable \( x_j \) can be written as,
\[
\frac{\partial \eta(x_j)}{\partial x_j} = q x_j^{q-1} \tag{19}
\]

For simplicity of computation, the term \( \beta \) is defined as,

\[
\beta = \left( \sigma_{vm,i} \right)^{p-1} \left( \frac{\partial \sigma_{vm,i}}{\partial \sigma_i} \right)^T \frac{\partial \eta(x_j)}{\partial x_j} \sigma_i \tag{20}
\]

The analytical form of term \( \frac{\partial \sigma_i}{\partial x_j} \) can be expressed as,

\[
\frac{\partial \sigma_i}{\partial x_j} = D_0B_i \frac{\partial u_i}{\partial x_j} = D_0B_i L_i \frac{\partial U}{\partial x_j} \tag{21}
\]

where \( [L_i]_{24 \times ndof} \) is a 0–1 sparse matrix to extract the nodal displacement of the \( i \)th element from the global displacement \( [U]_{ndof \times 1} \) \( (u_i = L_i U) \). Note that material stiffness matrix \( [D_0]_{6 \times 6} \) and the elemental strain matrix \( [B_i]_{6 \times 24} \) are independent of the design variable \( x_j \). Inserting Eq. (21) into the term \( T_2 \) in Eq. (16), we can find \( T_2 \) can be rewritten as,

\[
T_2 = \sum_{i=1}^{n_e l e} \left( \eta(x_i) \frac{\partial \sigma_{PN}}{\partial \sigma_{vm,i}} \left( \frac{\partial \sigma_{vm,i}}{\partial \sigma_i} \right)^T \frac{\partial \sigma_i}{\partial x_j} \right) = \sum_{i=1}^{n_e l e} \left( \eta(x_i) \frac{\partial \sigma_{PN}}{\partial \sigma_{vm,i}} \left( \frac{\partial \sigma_{vm,i}}{\partial \sigma_i} \right)^T D_0B_i L_i \frac{\partial U}{\partial x_j} \right) \tag{22}
\]

The adjoint method is applied here to resolve above equation. The term \( \frac{\partial U}{\partial x_j} \) can be obtained through differentiating both sides of the equilibrium Eq. (12) as follows,

\[
\frac{\partial K}{\partial x_j} U + K \frac{\partial U}{\partial x_j} = 0 \tag{23}
\]

Therefore, the term \( T_2 \) can be further written as,

\[
T_2 = \sum_{i=1}^{n_e l e} \left( -\eta(x_i) \frac{\partial \sigma_{PN}}{\partial \sigma_{vm,i}} \left( \frac{\partial \sigma_{vm,i}}{\partial \sigma_i} \right)^T D_0B_i L_i \right) K^{-1} \frac{\partial U}{\partial x_j} \tag{24}
\]

An adjoint variable \( [\lambda]_{ndof \times 1} \) is now defined as,

\[
\lambda^T = \left( \sum_{i=1}^{n_e l e} \eta(x_i) \frac{\partial \sigma_{PN}}{\partial \sigma_{vm,i}} \left( \frac{\partial \sigma_{vm,i}}{\partial \sigma_i} \right)^T D_0B_i L_i \right) K^{-1} \tag{25}
\]

Therefore, adjoint variable \( \lambda \) can be calculated from the adjoint equation as follows,

\[
K \lambda = \left( \sum_{i=1}^{n_e l e} \eta(x_i) \frac{\partial \sigma_{PN}}{\partial \sigma_{vm,i}} \left( D_0B_i L_i \right)^T \frac{\partial \sigma_{vm,i}}{\partial \sigma_i} \right) \tag{26}
\]
We define the vector $\gamma$ as,

$$\gamma = \left( \sum_{i=1}^{nele} \eta(x_i) \frac{\partial \sigma_{PN}}{\partial \sigma_{vm,i}} (D_0 B_i L_i) \frac{\partial \sigma_{vm,i}}{\partial \sigma_i} \right)$$  \hspace{1cm} (27)$$

Thus, the term $T_2$ can be further simplified to

$$T_2 = -\lambda^T \frac{\partial K}{\partial x_j} U$$  \hspace{1cm} (28)$$

The sensitivity of the global stiffness matrix $K$ with respect to a design variable $x_j$ equals,

$$\frac{\partial K}{\partial x_j} = \sum_{i=1}^{nele} L_i^T \frac{\partial k_i}{\partial x_j} L_i = L_j^T \left( \frac{\partial k_j}{\partial x_j} \right) L_j = L_j^T \left( p \cdot x_j^{p-1} k_j^{(s)} \right) L_j$$  \hspace{1cm} (29)$$

where $[k_j]_{24 \times 24}$ is the $j$th elemental stiffness matrix, and $k_j^{(s)}$ is the $j$th element stiffness with solid material. The detailed expression of element stiffness can be formulated as follows.

Fig. 3  Implementation flow of p-norm stress sensitivity
where \( \Omega_j \) is the \( j \)th element domain. The implementation flow of p-norm stress sensitivity is drawn in Fig. 3.

4.2 MATLAB implementation

The MATLAB code for 3D stress sensitivity analysis is executed with the following command:

\[
\text{where } \text{nelx, nely and nelz denote the number of elements along } x, y \text{ and } z \text{ directions. } [x]_{\text{nelx} \times 1} \text{ is the elemental density variable constrained in the range of } [0, 1]. \ p_l \text{ is the penalization parameter, } q \text{ is the stress relaxation parameter, and } p \text{ is the p-norm aggregation parameter. The output of the MATLAB function } \text{Stress}_3\text{D}_-\text{Sensitivity} \text{ is a } \text{nele} \times 1 \text{ vector, which is the sensitivity of p-norm stress with respect to the elemental density. The numerical implementation of von Mises stress calculation can be done by lines 33–40 as the following:}
\]

\[
\text{Stress}_3\text{D}_-\text{Sensitivity}(x, \text{nelx, nely, nelz, pl, q, p})
\]

where \( S \) is a \( \text{nele} \times 6 \) relaxed stress matrix, where each row of matrix \( S \) denotes a relaxed stress vector \( \bar{\sigma}_i \) corresponding to the \( i \)th element. MISES is a \( \text{nele} \times 1 \) vector, which stores values of von Mises stress. The objective function (Eq. 8) is formulated in line 46. The numerical implementation of computing \( T_1 \) can be done as following,

\begin{verbatim}
33 MISES = zeros(nele,1);
34 S = zeros(nele,6);
35 for i = 1:nele
36 temp = x(i)^q * (D * B * U(edofMat(i,:)))';
37 S(i,:) = temp;
38 MISES(i) = sqrt(0.5 * ((temp(1) - temp(2))^2 + (temp(1) - temp(3))^2... + (temp(2) - temp(3))^2 + 6 * sum(temp(4:6).^2)));
40 end
\end{verbatim}
DvmDs is a $\text{nele} \times 6$ matrix, where each row represents the derivative of $\sigma_{vm,i}$ with respect to $\hat{\sigma}_i$ corresponding to the $ith$ element. $\beta$ is a $\text{nele} \times 1$ vector, which represents the term $\beta$ in Eq. (20). $\text{dpn}_{dvms}$ is a scaler and given in line 44. It is worth to note that $[U]_{ndof \times 1}$ is global displacement vector and $[u]_{24 \times 1}$ is element displacement vector. The term $T_2$, is computed with the following MATLAB lines: where $\gamma$ is a $\text{ndof} \times 1$ vector defined in Eq. (27). The vector $\gamma$ is computed through lines 62–65 based on the formulation in Eq. (27). The adjoint vector $\lambda$ is computed through lines 66–67, and the term $T_2$ is obtained by lines 68–72. Finally, the sensitivity of p-norm global stress measure with respect to density is obtained in line 73.

5 Stress minimization problem formulation

5.1 Stress minimization

For simplicity, we focus on the volume-constrained stress minimization problem in this paper, expressed as follows,
\[
\begin{aligned}
\min \sigma_{PN} \\
\text{subject to } \int_{\Omega} d\Omega \leq V 
\end{aligned}
\]  

where $V$ is the upper limit for the volume constraint, and $\Omega$ is the design domain. Note that stress sensitivity analysis and corresponding MATLAB code in this paper can be applied to different stress-related problems, such as stress-constrained compliance minimization.

### 5.2 Optimization algorithms

For topology optimization, several advanced optimization algorithms have been proposed in recent years. The Optimality Criteria (OC) (Sigmund 2001) is a classical approach for structural optimization problems, followed by recent methods such as Sequential Linear Programming (SLP) and Sequential Quadratic programming (SQP) (Liu and Tovar 2014). Compared to these methods, the Method of moving asymptotes (MMA) proposed by Svanberg 1987 is more popular in topology optimization. MMA optimizer approximates the nonlinear optimization problem with the following sub programming problem, where $L_i^{(k)}$ and $U_i^{(k)}$ are lower and upper asymptotes and updated in every iteration. The detailed description of updating scheme for $L_i^{(k)}$ and $U_i^{(k)}$ can be found in Svanberg 1987. $x^{(k)}$ is current design point, and $v$ denotes the volume of element. The MATLAB program (mmasub.) implementing the MMA algorithm may be obtained through contacting Prof. Krister Svanberg from KTH. The implementation of the MMA algorithm for the volume-constrained stress minimization problem is straightforward. A sample MATLAB code for computing objective, constraint and their sensitivities for MMA optimizer are as follow,

\[
\begin{aligned}
\text{find } \vec{x} \\
\text{minimize } - \sum_{i=1}^{n} \left( \frac{x_i^{(k)} - L_i^{(k)}}{\bar{x}_i - L_i^{(k)}} \right) \frac{\partial \sigma_{PN}}{\partial x_i} (x^{(k)}) \\
\text{subject to } \vec{x}^T v - \bar{v} \leq 0 \\
\text{where } 0.1x_i^{(k)} + 0.9L_i^{(k)} \leq \bar{x}_i \leq 0.9U_i^{(k)} + 0.1x_i^{(k)} \ (i = 1, 2, \ldots n)
\end{aligned}
\]
Note that $H_s$ and $H$ are filter coefficient vectors; the MATLAB program for computing $H_s$ and $H$ can be found in Liu et al. 2014). volfrac corresponds to the desired volume fraction. $f_0val$ is the objective function (p-norm stress), and vector $df_0dx$ is the sensitivity of objective with respect to design variables. Scaler $fval$ is the volume constraint value, and the vector $dfdx$ is the sensitivity of volume constraint with respect to design variables. The filter technique is applied in lines 4, 7 and 8. For density-based topology optimization, filters are critical to avoid numerical instabilities, such as mesh-dependency and checkerboard patterns (Bendsoe and Sigmund 2013). Several different filters have been proposed in recent years, and each filter may yield different topology solutions. A review of filter techniques for density-based topology optimization methods can be found in Sigmund 2007.

6 Numerical example

6.1 Sensitivity verification

To verify the correctness of p-norm stress sensitivity, the analytical sensitivity is compared with the sensitivity obtained by the finite difference method. The forward finite difference method for approximating derivatives of a p-norm stress based on a truncated Taylor series expansion can be expressed as,

$$
\left[ \frac{D\sigma_{PN}(x)}{Dx_i} \right]_f = \frac{\sigma_{PN}(x + \varepsilon e_i) - \sigma_{PN}(x)}{\varepsilon}
$$

where $e_i = [0, 0, \ldots, 1, \ldots, 0, 0]^T$ is a unit vector of component $i$ and $\varepsilon$ is the perturbation. In general, a smaller $\varepsilon$ can get a better approximation for analytical sensitivity, while numerical round-off error will erode the accuracy of the approximation if the value of $\varepsilon$ is too small. To quantify the difference between the
forward finite difference with the analytical sensitivity, the relative percentage error is introduced as follows,

\[ e_f = \left| \frac{D\sigma_{ps}(x)}{DX_i} - f \frac{D\sigma_{ps}(x)}{DX_i} \right| \]

where the operator \(|\cdot|\) denotes the absolute value. The MATLAB code for sensitivity verification is as follows,

```matlab
1 clear
2 clc
3 nelx = 40;
4 nely = 20;
5 nelz = 1;
6 x = 0.3 * ones(nely, nelx, nelz);
7 x = x(:);
8 d = 1e - 4;
9 e = zeros(nelx * nely, 1);
10 fd = zeros(nelx * nely, 1);
11 pl = 3;
12 q = 0.5;
13 p = 8;
14 for i = 1: nelx * nely
15 x0 = x;
16 x0(i) = x0(i) + d;
17 [pnorm, pnorm Sen] = Stress_3D_Sensitivity(x, nelx, nely, nelz, pl, q, p);
18 [pnorm f, pnorm Sen] = Stress_3D_Sensitivity(x0, nelx, nely, nelz, pl, q, p);
19 fd(i) = (pnorm f - pnorm Sen)/d;
20 e(i) = abs((fd(i) - pnorm Sen(i))/pnorm Sen(i));
21 end
```

Due to computational cost, the number of elements in each direction are chosen as nelx = 40, nely = 20, nelz = 1. The perturbation is chosen as \(\varepsilon = 1e - 4\) in line 8, and the relative percentage error is computed in line 20. The relative percentage error is plotted in Fig. 4 for all design variables \(x\). It is worth noting a smaller error can be achieved through adapting the value of perturbation \(\varepsilon\). The comparison of sensitivity distribution of analytical and forward finite difference approximation is plotted in Fig. 5, confirming good consistency is achieved. Note that the relative sensitive errors are caused by finite difference step size \(\varepsilon\). Adapting step size \(\varepsilon\) can obtain a better agreement. We can also verify the sensitivity consistency for any other random density distribution in the range of \([0.1,1]\) by changing the code in line 6 to the following: \(x = 0.9*rand(nely, nelx, nelz) + 0.1\).
6.2 Stress minimization for the cantilever beam

The first stress minimization example shown here is a 2D cantilever beam, where $n_{elx} = 200$, $n_{ely} = 60$, $n_{elz} = 1$ as shown in Fig. 5. The loading is applied on the right-bottom corner, and left side is fully fixed. The volume fraction constraint $V$ is chosen as 0.3. The filter radius is $r = 2.5$. The other parameters are selected as: $pl = 3$; $q = 0.5$; $p = 10$ (Fig. 6). The moving limit of MMA optimizer is set as 0.1. the MMA solver converges after 100 iterations and optimized result is plotted in Fig. 7a. The von Mises stress distribution is shown in Fig. 7b. The convergence history is plotted in Fig. 8. The p-norm stress value decreases from initial 33.17 to 3.31 after optimization. The completed MATLAB program is provided in electric supplementary material for educational purposes only.
6.3 Stress minimization for L-bracket

In this section, a well-known 2D L-bracket benchmark is illustrated in Fig. 9 with the characteristic dimensions. The design domain is discretized by \( n_{elx} = 200, n_{ely} = 200, n_{elz} = 1 \). The passive elements technique is applied as
shown in Fig. 9, and the detailed description for passive element implementation can be found in Liu et al. 2014. As shown in Fig. 9, the design domain of the L-bracket contains an internal corner which causes a stress singularity. The optimization parameters are chosen as: \( p_l = 3; q = 0.5; p = 10 \). The filter radius is set to be 2.5. The volume fraction constraint is 0.3. It is worth noting the loading is evenly distributed on three elements at the right corner. Because the loading is uniformed applied on elements, there is no singularity phenomenon nearby loading points. The MATLAB code for boundary and loading can be found and explained in Sect. 6.4. The optimized result is demonstrated in Fig. 10a, and von Mises distribution is plotted in Fig. 10b. The p-norm stress decreases from the initial 56.23 to 5.51 after 120 iterations.
6.4 Stress minimization for 3D L-bracket

In this section, we extended the previous 2D L-bracket example to 3D design. The loading and boundary condition are shown in Fig. 11. The design domain is described by \( n_{ex} = 100, n_{ey} = 100, \) \( n_{ez} = 30. \) The passive element technique is applied as following MATLAB code, where \( x \) is elemental density vector. The design parameters for 3D L-bracket case is chosen as: \( p_l = 3; q = 0.5; p = 10. \) The filter radius is 2.5.

\[
x_t = \text{reshape}(x, n_{ey}, n_{ex}, n_{ez});
x_t(1: n_{ex}/2, n_{ey}/2: \text{end}, :) = 1e^{-4};
x = x_t(:,);
\]

As shown in Fig. 11, the upper side of L-bracket is fully fixed. The corresponding MATLAB code for such boundary condition is as follows,

```matlab
fixed_node = 1: (n_{ey} + 1): ((n_{ey} + 1) * (n_{ex} + 1)) - n_{ey};
for j = 1: (n_{ez} + 1)
  fixed_node = [fixed_node, 1: (n_{ey} + 1): ((n_{ey} + 1) * (n_{ex} + 1)) - n_{ey} + j * (n_{ey} + 1) * (n_{ex} + 1)];
end
fixeddof = [3 * fixed_node - 2,3 * fixed_node - 1,3 * fixed_node];
```

The vertical loading is applied at the right-upper side, which is uniformly distributed on \( 3 \times n_{ez} \) elements. The corresponding MATLAB code is shown as following,

Fig. 11 3D L-bracket example
For the 3D L-bracket design problem, the direct solver will take a long time to resolve the large-scale linear systems. The preconditioned conjugate solver built in MATLAB as the function pcg replaces the direct solver. The MATLAB implementation procedure is similar to Ref (Liu and Tovar 2014) as follows,

\begin{verbatim}
[il, jl, kl] = meshgrid(nelx, 0, 0:nelz);
loadnid = kl * (nelx + 1) * (nely + 1) + il * (nely + 1) + (nely + 1 - jl);
loaddf1 = (3 * loadnid(:, :) - 1) - nelx/2 * 3;
loaddf2 = (3 * loadnid(:, :) - 1) - nelx/2 * 3 + 3;
loaddf3 = (3 * loadnid(:, :) - 1) - nelx/2 * 3 + 6;
loaddf = union(union(loaddf1, loaddf2), loaddf3);
\end{verbatim}

where matrix M works as a preconditioner. For computing adjoint vector \( \lambda \), we replace the line 67 in MATLAB function Stress_3D_Sensitivity with following code,

\begin{verbatim}
tolit = 1e-8;
maxit = 5000;
M = diag(diag(K(freedofs, freedofs)));
U = pcg(K(freedofs, freedofs), F(freedofs), tolit, maxit, M);
\end{verbatim}

\textbf{Fig. 12} Optimized material layout \textbf{a} front view \textbf{b} rear view
The optimized material layout for 3D L-bracket example is demonstrated in Fig. 12. Note that the initial sharp corner is replaced by a round shape to avoid a local stress concentration. The p-norm stress decreases significantly from the initial 46.12 to 4.72 after 60 iterations. The von Mises distribution for the optimized result is plotted in Fig. 13. The MATLAB command for plotting von Mises stress is as follows,

```
plot_von_Mises(density, von_Mises)
```

where the density and von_Mises are both input matrices with dimensions \( nely \times nelx \times nelz \). The MATLAB code of the function `plot_von_Mises` can be found in Appendix B. It is worth mentioning the 2D L-bracket example is a special case of the 3D example (\( nelz = 1 \)).

7 Conclusion

In recent years, stress has become the most important consideration for topology optimization. However, stress-based optimization is challenging due to local nature of computing the stress and highly nonlinear behavior. Local stress is highly dependent on the design, and extreme changes of local stress may occur due to density changes in neighboring regions, meaning stress gradients are highly sensitive to local density changes. Therefore, an accurate, analytical sensitivity is critical for stress related topology optimization problems. In this paper, we derive the sensitivity of the p-norm global stress measure based on a relaxed stress formulation (Le et al. 2010), and a corresponding MATLAB code is explained in detail. The finite element formulation in this paper is based on eight-node hexahedral elements, where the design domain is discretized by uniform hexahedral elements. The MMA optimizer is applied for volume-constrained stress minimization problems, where different loading and boundary conditions are explained by MATLAB code. The 146-line code for stress sensitivity analysis is provided for educational purposes, and the users can modify the code based on their

Fig. 13 Von Mises stress distribution
requirements. The sensitivity analysis code is written in a compact and vectorized way to enhance the computational efficiency. The complete code is given in the Appendix for users’ reference and the source code can be directly downloaded in supplementary material.

Appendix A: MATLAB Program Stress_3D_Sensitivity

The MATLAB code used in this work can be downloaded from: https://github.com/PittAMRL/StressTopOpt.

Appendix B: MATLAB Program Plot_von_Mises

```matlab
function []=plot_von_Mises(density,von_Mises)
fv=isosurface(density,0.5);
[F1,V1]=isosurface(density,0.5);
[F2,V2]=isocaps(density,0.5);
F3=[F1,F2+length(V1(:,1))];
V3=[V1;V2];
fv.vertices=V3;
fv.faces=F3;
p = patch(fv);
cdata = von_Mises;
isoscolors(cdata,p);
p.FaceColor = 'interp';
p.EdgeColor = 'none';
view(3);
axis equal;camlight headlight;lighting phong;
material([0.3,0.6,1,15,1.0]);axis off;colormap('jet');drawnow;
```

Acknowledgements The financial support for this work from National Science Foundation (CMMI-1634261) is gratefully acknowledged. The authors would like to thank Krister Svanberg for providing the MATLAB implementation of his Method of Moving Asymptotes, which was used in this work.

References

Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: an easy-to-use, fully parallel, open source topology optimization framework. Struct Multidiscip Optim 51(3):565–572
Allaire G, Jouve F (2008) Minimum stress optimal design with the level set method. Eng Anal Bound Elem 32(11):909–918
Allaire G, Dapogny C, Frey P (2014) Shape optimization with a level set based mesh evolution method. Comput Methods Appl Mech Eng 282:22–53
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
Bendsoe MP, Sigmund O (2013) Topology optimization: theory, methods, and applications. Springer Science & Business Media.
Bendsoe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71(2):197–224
Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9–10):635–654
Bruggi M, Duysinx P (2012) Topology optimization for minimum weight with compliance and stress constraints. Struct Multidiscip Optim 46(3):369–384
Cai S, Zhang W (2015) Stress constrained topology optimization with free-form design domains. Comput Methods Appl Mech Eng 289:267–290
Challis VJ (2010) A discrete level-set topology optimization code written in Matlab. Struct Multidiscip Optim 41(3):453–464
Chen Q, Zhang X, Zhu B (2019) A 213-line topology optimization code for geometrically nonlinear structures. Struct Multidiscip Optim 59(5):1863–1879
Cheng G, Guo X (1997) ε-relaxed approach in structural topology optimization. Struct Optim 13(4):258–266
Dapogny C, Frey P, Omnès F, Privat Y (2018) Geometrical shape optimization in fluid mechanics using FreeFem++. Struct Multidiscip Optim 58(6):2761–2788
Deng H, To AC (2020) Topology optimization based on deep representation learning (DRL) for compliance and stress-constrained design. Comput Mech 66:449–469
Deng H, Cheng L, To AC (2019) Distortion energy-based topology optimization design of hyperelastic materials. Struct Multidiscip Optim 59(6):1895–1913
Deng H, Hinnebusch S, To AC (2020a) Topology optimization design of stretchable metamaterials with Bézier skeleton explicit density (BSED) representation algorithm. Comput Methods Appl Mech Eng 366:113093
Deng H, Cheng L, Liang X, Hayduke D, To AC (2020b) Topology optimization for energy dissipation design of lattice structures through snap-through behavior. Comput Methods Appl Mech Eng 358:112641
Duysinx P, Sigmund O (1998) New developments in handling stress constraints in optimal material distribution. In 7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, p 4906.
Duysinx P, Bendsøe MP (1998) Topology optimization of continuum structures with local stress constraints. Int J Numer Meth Eng 43(8):1453–1478
Emmendoerfer H Jr, Fancello EA (2016) Topology optimization with local stress constraint based on level set evolution via reaction–diffusion. Comput Methods Appl Mech Eng 305:62–88
Fan Z, Xia L, Lai W, Xia Q, Shi T (2019) Evolutionary topology optimization of continuum structures with stress constraints. Struct Multidiscip Optim 59(2):647–658
Ferrari F, Sigmund O (2020) A new generation 99 line Matlab code for compliance Topology Optimization and its extension to 3D. arXiv preprint arXiv:2005.05436.
Gao J, Luo Z, Xia L, Gao L (2019) Concurrent topology optimization of multiscale composite structures in Matlab. Struct Multidiscip Optim 60(6):2621–2651
Guo X, Zhang WS, Wang MY, Wei P (2011) Stress-related topology optimization via level set approach. Comput Methods Appl Mech Eng 200(47–48):3439–3452
Holmberg E, Torstenfelt B, Klarbring A (2013) Stress constrained topology optimization. Struct Multidiscip Optim 48(1):33–47
Huang X, Xie Y-M (2010) A further review of ESO type methods for topology optimization. Struct Multidiscip Optim 41(5):671–683
James KA, Lee E, Martins JR (2012) Stress-based topology optimization using an isoparametric level set method. Finite Elem Anal Des 58:20–30
Kambampati S, Gray JS, Kim HA (2020) Level set topology optimization of structures under stress and temperature constraints. Comput Struct 235:106265
Kirsch U (1990) On singular topologies in optimum structural design. Struct Optim 2(3):133–142
Kočvara M, Stingl M (2012) Solving stress constrained problems in topology and material optimization. Struct Multidiscip Optim 46(1):1–15
Laurain A (2018) A level set-based structural optimization code using FEniCS. Struct Multidiscip Optim 58(3):1311–1334
Le C, Norato J, Bruns T, Ha C, Tortorelli D (2010) Stress-based topology optimization for continua. Struct Multidiscip Optim 41(4):605–620
Liang Y, Cheng G (2020) Further elaborations on topology optimization via sequential integer programming and Canonical relaxation algorithm and 128-line MATLAB code. Struct Multidiscip Optim 61(1):411–431
Lin H, Xu A, Misra A, Zhao R (2020) An ANSYS APDL code for topology optimization of structures with multi-constraints using the BESO method with dynamic evolution rate (DER-BESO). Struct Multidiscip Optim 62:1–26
Liu K, Tovar A (2014) An efficient 3D topology optimization code written in Matlab. Struct Multidiscip Optim 50(6):1175–1196
Liu Z, Korvink JG, Huang R (2005) Structure topology optimization: fully coupled level set method via FEMLAB. Struct Multidiscip Optim 29(6):407–417
Loyola RA, Querin OM, Jiménez AG, Gordoa CA (2018) A sequential element rejection and admission (SERA) topology optimization code written in Matlab. Struct Multidiscip Optim 58(3):1297–1310
Luo Y, Wang MY, Kang Z (2013) An enhanced aggregation method for topology optimization with local stress constraints. Comput Methods Appl Mech Eng 254:31–41
Otomori M, Yamada T, Izui K, Nishiwaki S (2015) Matlab code for a level set-based topology optimization method using a reaction diffusion equation. Struct Multidiscip Optim 51(5):1159–1172
París J, Navarrina F, Colominas I, Casteleiro M (2009) Topology optimization of continuum structures with local and global stress constraints. Struct Multidiscip Optim 39(4):419–437
París J, Navarrina F, Colominas I, Casteleiro M (2010) Stress constraints sensitivity analysis in structural topology optimization. Comput Methods Appl Mech Eng 199(33–36):2110–2122
Pereira A, Talischi C, Paulino GH, Menezes IF, Carvalho MS (2016) Fluid flow topology optimization in PolyTop: stability and computational implementation. Struct Multidiscip Optim 54(5):1345–1364
Picelli R, Townsend S, Brampton C, Norato J, Kim HA (2018a) Stress-based shape and topology optimization with the level set method. Comput Methods Appl Mech Eng 329:1–23
Picelli R, Townsend S, Kim HA (2018b) Stress and strain control via level set topology optimization. Struct Multidiscip Optim 58(5):2037–2051
Picelli R, Sivapuram R, Xie YM (2020) A 101-line MATLAB code for topology optimization using binary variables and integer programming. Struct Multidiscip Optim 63:1–20
Sanders ED, Pereira A, Aguiol MA, Paulino GH (2018) PolyMat: an efficient Matlab code for multi-material topology optimization. Struct Multidiscip Optim 58(6):2727–2759
Senhora FV, Giraldo-Londono O, Menezes IF, Paulino GH (2020) Topology optimization with local stress constraints: a stress aggregation-free approach. Struct Multidiscip Optim 62(4):1639–1668
Sigmund O (2001) A 99 line topology optimization code written in Matlab. Struct Multidiscip Optim 21(2):120–127
Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidiscip Optim 33(4–5):401–424
Smith H, Norato JA (2020) A MATLAB code for topology optimization using the geometry projection method. Struct Multidiscip Optim 62:1–16
Song Y, Ma Q, He Y, Zhou M, Wang MY (2020) Stress-based shape and topology optimization with cellular level set in B-splines. Struct Multidiscip Optim 62(5):2391–2407
Suresh K (2010) A 199-line Matlab code for Pareto-optimal tracing in topology optimization. Struct Multidiscip Optim 42(5):665–679
Suresh K, Takallozadeh M (2013) Stress-constrained topology optimization: a topological level-set approach. Struct Multidiscip Optim 48(2):295–309
Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Meth Eng 24(2):359–373
Talischi C, Paulino GH, Pereira A, Menezes IF (2012) PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. Struct Multidiscip Optim 45(3):329–357
Van Miegroet L, Duysinx P (2007) Stress concentration minimization of 2D filets using X-FEM and level set description. Struct Multidiscip Optim 33(4–5):425–438
Wang MY, Li L. (2013) Shape equilibrium constraint: a strategy for stress-constrained structural topology optimization. Struct Multidiscip Optim 47(3):335–352
Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1–2):227–246
Wei P, Li Z, Li X, Wang MY (2018) An 88-line MATLAB code for the parameterized level set method based topology optimization using radial basis functions. Struct Multidiscip Optim 58(2):831–849
Xia L, Breitkopf P (2015) Design of materials using topology optimization and energy-based homogenization approach in Matlab. Struct Multidiscip Optim 52(6):1229–1241
Xia Q, Shi T, Liu S, Wang MY (2012) A level set solution to the stress-based structural shape and topology optimization. Comput Struct 90:55–64
Xia Q, Wang MY, Shi T (2014) A level set method for shape and topology optimization of both structure and support of continuum structures. Comput Methods Appl Mech Eng 272:340–353
Xia L, Zhang L, Xia Q, Shi T (2018) Stress-based topology optimization using bi-directional evolutionary structural optimization method. Comput Methods Appl Mech Eng 333:356–370
Yang D, Liu H, Zhang W, Li S (2018) Stress-constrained topology optimization based on maximum stress measures. Comput Struct 198:23–39
Zegard T, Paulino GH (2014) GRAND—Ground structure based topology optimization for arbitrary 2D domains using MATLAB. Struct Multidiscip Optim 50(5):861–882
Zhang WS, Guo X, Wang MY, Wei P (2013) Optimal topology design of continuum structures with stress concentration alleviation via level set method. Int J Numer Meth Eng 93(9):942–959
Zhang S, Norato JA, Gain AL, Lyu N (2016) A geometry projection method for the topology optimization of plate structures. Struct Multidiscip Optim 54(5):1173–1190
Zhang S, Gain AL, Norato JA (2017) Stress-based topology optimization with discrete geometric components. Comput Methods Appl Mech Eng 325:1–21
Zhang W, Li D, Zhou J, Du Z, Li B, Guo X (2018) A moving morphable void (MMV)-based explicit approach for topology optimization considering stress constraints. Comput Methods Appl Mech Eng 334:381–413
Zhu B et al (2020) Design of compliant mechanisms using continuum topology optimization: a review. Mech Mach Theory 143:103622

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