Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Low-cost formulation-free sensitivity analysis for path-dependent topology optimization using fixed-point automatic differentiation

Shun Ogawa

ogawa@struct.t.u-tokyo.ac.jp

The University of Tokyo: Tokyo Daigaku https://orcid.org/0009-0006-0480-1309

Katsuyuki Suzuki

The University of Tokyo

Research Article

Keywords: Topology optimization, Automatic differentiation, Multiphysics, Path-dependent sensitivity analysis, Thermo-elastoplasticity, Multi-material

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11670167/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Low-cost formulation-free sensitivity analysis for path-dependent topology optimization using fixed-point automatic differentiation

Shun Ogawa[1*] and Katsuyuki Suzuki[1]

[1*] Department of Systems Innovation, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, 113-8656, Tokyo, Japan .

*Corresponding author(s). E-mail(s): ogawa@struct.t.u-tokyo.ac.jp;

Abstract

A low-cost formulation-free sensitivity analysis method based on fixed-point automatic differentiation (AD) is presented for the topology optimization of multiphysics and path-dependent problems. With the conventional adjoint variable method, any change to the constitutive law, material interpolation, objective function, or coupling with another physical field requires the adjoint recursion and partial derivatives with respect to the design variables to be derived again by symbolic expansion, and the formulation and its verification must be redone. With the proposed method, the adjoint recursion and partial derivatives are obtained automatically so long as the forward analysis can be evaluated as the subject of AD. Thus, no formulation specific to a physical model is needed. To avoid the increasing memory consumption from AD recording every operation of the forward analysis, the proposed method fixes a consistent tangent at the converged point of each load step. It uses reverse-mode AD to record only once a map consisting of the evaluation of the residual, the update of the solution, and the update of the internal variables. Finally, it propagates the sensitivity of the internal variables through a backward recursion between the steps. Thus, the tape held at any one time is that of a single analysis step and is independent of the number of load steps. The proposed method was applied to the topology optimization of a thermo-elastoplastic multi-material problem based on a von Mises elastoplastic model coupled with steady-state heat conduction. In numerical examples, the sensitivities obtained by the proposed method agreed with those of the analytical adjoint variable method and reverse-mode AD recording every operation to within the roundoff error, and the computation time and peak memory of one gradient evaluation grew with the number of elements and number of load steps at the same rate as the analytical adjoint variable method while remaining

lower than those of reverse-mode AD. Two-dimensional optimization yielded the same design regardless of the sensitivity analysis method, and three-dimensional optimization showed the practicality of the proposed method.

Keywords: Topology optimization, Automatic differentiation, Multiphysics, Path-dependent sensitivity analysis, Thermo-elastoplasticity, Multi-material

1 Introduction

Gradient-based topology optimization involves searching along gradients of the design variables to determine the optimal solution that still satisfies the objective and constraint functions that define the optimization problem, which is also known as a sensitivity analysis. Conventionally, sensitivity analyses have been conducted by using the adjoint variable method (hereafter referred to as the analytical adjoint variable method). However, the analytical adjoint variable method must be formulated according to the functions treated in the optimization problem and the definition of the physical model, and the formulation must be repeated whenever the problem changes. This in turn requires the optimization program to be rewritten and verified again, which imposes a practical burden. For structures undergoing plastic deformation, the response is materially nonlinear, so the equilibrium equations must be solved iteratively. In addition, the response is not determined by the current displacement alone as the plastic strain and hardening variables at each integration point depend on the preceding load history. These factors increase the complexity of the physical model and the number of dependent variables, which makes the analytical adjoint variable method increasingly impractical and any optimization program difficult to verify[1]. Therefore, a new topology optimization method is needed that solves problems with arbitrary functions and physical models without requiring the optimization program to be rewritten for each case.

The analytical adjoint variable method has been applied to the topology optimization of elastoplastic structures, whose formulation is complicated by internal variables and path dependence[2]. It has since been extended to finite-strain plasticity[3], stress constraints based on the plastic strain[4], elastoplastic composites[5], and periodic microstructures that maximize plastic dissipation[6]. Alberdi et al.[7] established a unified framework for sensitivity analysis of nonlinear path-dependent problems, which Abueidda et al.[8] applied to the optimization of three-dimensional elastoplastic structures with periodic boundary conditions. Granlund and Wallin[9] developed an adjoint recursion that mirrors the state-update equations and demonstrated its application to three-dimensional problems involving millions of degrees of freedom. These studies demonstrate the applicability of the analytical adjoint method to large-scale three-dimensional problems. However, changes to the constitutive law, material interpolation, objective function, boundary conditions, or coupling with other physical fields may require re-derivation of the adjoint recursion and the partial derivatives with respect to the design variables, as well as corresponding modifications and re-verification of the optimization code.

Automatic differentiation (AD)[10] eliminates the need to derive complicated derivative expressions analytically. It evaluates derivatives of a target function by applying the chain rule to the derivatives of individual operations in its computational graph. Because sensitivity analysis involves differentiating the objective and constraint functions with respect to the design variables, AD enables sensitivity analysis without requiring manual derivation of problem-specific sensitivity expressions. In topology optimization, the number of design variables is typically proportional to the number of elements. Therefore, computing gradients with respect to all design variables requires an approach whose computational cost is comparable to that of a single forward analysis. Reverse-mode AD is well suited to this setting. By recording the operations performed during the forward analysis and propagating derivatives backward through the computational graph, reverse-mode AD can automatically compute sensitivities following changes to the constitutive law or coupling with other physical fields, potentially addressing limitations of the analytical adjoint method. Nørgaard et al.[11] applied AD to topology optimization of unsteady flow problems, comparing gradients obtained using CoDiPack and Tapenade with those derived manually using an adjoint method. Chandrasekhar et al.[12] presented AuTO, a JAX-based framework for topology optimization of linear elastic problems. Yin et al.[13] unrolled the time integration of elastoplasticity into a computational graph using JAX, enabling topology optimization without manually deriving the sensitivities. However, recording every operation retains the computational history of all load steps, including global Newton iterations and local return-mapping iterations. Consequently, the memory required to store the tape (i.e., the sequence of operations recorded by reverse-mode AD[10]) scales with the number of load steps, the number of iterations per load step, and the number of elements. The authors[14] previously combined reverse-mode AD with checkpointing[15] to reduce memory requirements in elastoplastic multi-material topology optimization. However, checkpointing reduces the memory required to store the computational history at any given time by recomputing parts of the forward analysis during the backward pass, thereby trading computational time for memory. Other researchers have combined AD with parts of the analytical adjoint derivation. Jia and Zhang[16] performed three-dimensional multi-material topology optimization of finite-strain elastoplasticity using FEniCSx. They derived the adjoint recursion for each internal variable and used AD to compute the individual partial derivatives. Although this approach reduces the need for manual derivative calculations, deriving the adjoint equations remains the most complex step and depends on the underlying physical model. Thus, this approach does not fully eliminate the formulation-specific derivation required by the analytical adjoint method.

With recent developments in machine learning, the forward analysis itself can be included in the scope of AD, and some researchers have focused on making the finite element framework differentiable. Because a forward analysis involves a large amount of computation, AD must find the derivatives while minimizing its memory consumption. Dolfin-adjoint[17, 18] derives the discrete adjoint automatically from a forward analysis written in variational form. Because the solution to the nonlinear equations is differentiated by the implicit function theorem, the Newton iterations are not recorded. Xue et al.[19] developed JAX-FEM, which uses AD of the element

residual to compute the element tangent matrix, enabling the construction of forward analyses without manually deriving the tangent matrix, even for constitutive models such as J2 plasticity. For inverse problems, the discretized governing equations are treated as a single constraint, and the sensitivities of the solution are computed using the implicit function theorem. However, Xue et al. considered only inverse problems such as full-field inference for a linear Poisson equation and compliance minimization of linear elastic structures subject to volume, eigenfrequency, and stress constraints. These problems involve no load history and require only a single solve. In contrast, an elastoplastic forward analysis involves sequential load steps, with the solution at each step depending on the internal variables from the preceding step, such as stress and plastic strain. These internal variables also depend on the design variables. The adjoint formulation developed by Xue et al. addresses a single system of governing equations but does not explicitly formulate the backward propagation of sensitivities across load steps through the history-dependent internal variables. To realize this propagation, one approach is to embed the solve of each load step in the computational graph as a single operation differentiated by the implicit function theorem and the chain of all load steps traversed backward by reverse-mode AD. Such a construction can be realized with the custom differentiation rules provided by frameworks such as JAX[20]. However, to the best of the authors' knowledge, it has not been applied to the topology optimization of rate-independent elastoplasticity with a yield surface, and neither the formulation of the sensitivity analysis, the condition under which a single differentiation at the converged solution coincides with the differentiation by the implicit function theorem, a verification of the gradients, nor a quantitative comparison of the computational cost has been reported for it. A second approach to propagating the sensitivity back to the previous step involves constructing a separate adjoint recursion between steps. Sharma et al.[21] applied this approach to material parameter identification by implementing the constitutive update as a custom differentiation rule based on the implicit function theorem and computing sensitivities using a discrete adjoint that propagates backward through the load history. The finite element library torch-fem[22] similarly implements a procedure that solves the adjoint equation at the converged Newton solution of each load increment and reevaluates the residual to propagate sensitivities to the state variables of the preceding increment and the design parameters. However, to the best of the authors' knowledge, the computational cost of sensitivity analysis using these frameworks has not been quantitatively compared with that of the analytical adjoint variable method for elastoplastic topology optimization.

The concept of differentiating the result of a converged iteration without recording the course of the iteration has long been part of AD. Christianson[23] showed, for a computation containing a convergent iteration, that the gradient can be obtained in reverse mode by reusing the computational graph of a single iteration at the fixed point. Griewank and Faure[24] developed this into a method that obtains reduced gradients and Hessians from the fixed-point iteration of a state equation, and a systematic treatment is given by Griewank and Walther[10]. Fixed-point AD has been used for practical applications such as aerodynamic shape design in the discrete adjoint of the flow solver SU2[25] and the AD library CoDiPack[26]. In the field of machine learning,

Blondel et al.[27] established a framework for fixed-point AD that gives any solver routine a derivative based on the implicit function theorem. Other researchers have discussed the validity of an approximation that differentiates only the last iteration of an iterative algorithm[28, 29]. However, these studies treated a single and stationary fixed point, and fixed-point AD has not been applied to elastoplastic problems in which a fixed point appears at every load step and its solution is handed to the next step through internal variables, nor has it been used in topology optimization.

In this study, a low-cost formulation-free sensitivity analysis method based on fixed-point AD was developed for multiphysics and path-dependent problems. In the proposed method, the adjoint recursion and partial derivatives with respect to the design variables are obtained by AD, so they need not be derived by symbolic expansion and handled individually for each physical model. So long as the forward analysis can be evaluated as the subject of AD and the tangent at the converged point is consistent, the procedure need not be changed when the constitutive law or coupling is changed. To reduce the enormous computational cost of AD, Mehmood and Ochs'[30] approach was adopted for fixed-point AD, where the map is only recorded once at the point at which the iteration has converged and is differentiated by reverse-mode AD. Because the tape of a single analysis step is discarded after the backward propagation, the amount of tape held at one time is independent of the number of load steps, and the procedure does not depend on the features of a particular AD library. To demonstrate the proposed method, it is applied to the topology optimization of a thermo-elastoplastic multi-material problem involving a von Mises elastoplastic model coupled with steady-state heat conduction. The remainder of the paper is organized as follows. First, we present the formulation of fixed-point AD for path-dependent finite element elastoplastic analysis, including the conditions under which a single recording at the converged solution yields sensitivities equivalent to those obtained by differentiating the discrete solution via the implicit function theorem. We also describe how sensitivities are propagated through the internal variables and the heat conduction analysis. Second, we demonstrate that the computed sensitivities agree with those obtained using the analytical adjoint variable method and reverse-mode AD that records every operation for the same discrete problem, up to roundoff error. Furthermore, the computation time and the peak memory of one gradient evaluation are shown to stay at a computational cost comparable to the analytical adjoint variable method and vastly lower than that of reverse-mode AD recording every operation and checkpointing. Third, two- and three-dimensional optimizations are presented to demonstrate how the proposed method reduces the high computational cost of conventional AD recording every operation and can solve practical problems.

2 Governing equations

The topology optimization of a quasi-static thermo-elastoplastic model involves multiphysics, history dependence, and material nonlinearity. The governing equations of the heat conduction problem and elastoplastic problem associated with this model are given below.

2.1 Heat conduction problem

The temperature field T obeys the following steady-state heat conduction equation without heat generation:

$$\begin{cases} -\nabla \cdot (k \nabla T) = 0 & \text{in } \Omega, \\ T = \bar{T} & \text{on } \Gamma_T, \\ -k \nabla T \cdot \boldsymbol{n} = \bar{q} & \text{on } \Gamma_q, \end{cases} \tag{1}$$

where k is the thermal conductivity, $\boldsymbol{n}$ is the outward unit normal vector, $\bar{q}$ is the prescribed heat flux, Γ_q denotes the heat-flux boundary, and Γ_T denotes the boundary on which the temperature $\bar{T}$ is prescribed. The remaining boundary is adiabatic. The weak form of Eq. (1) is given by

$$\int_\Omega k \, \nabla T \cdot \nabla \delta T \, \mathrm{d}\Omega = \int_{\Gamma_q} \bar{q} \delta T \, \mathrm{d}\Gamma, \tag{2}$$

where δT is the virtual temperature. Discretizing with finite elements gives the following equation for the nodal temperature vector $\boldsymbol{T}$:

$$\begin{cases} \boldsymbol{K}_T \boldsymbol{T} = \boldsymbol{Q} \\ \boldsymbol{K}_T = \displaystyle\int_\Omega \boldsymbol{B}_T^{\mathsf{T}} \boldsymbol{k} \, \boldsymbol{B}_T \, \mathrm{d}\Omega \\ \boldsymbol{Q} = \displaystyle\int_{\Gamma_q} \boldsymbol{N}_T^{\mathsf{T}} \bar{q} \, \mathrm{d}\Gamma, \end{cases} \tag{3}$$

where $\boldsymbol{N}_T$ denotes the temperature shape functions, $\boldsymbol{B}_T$ is the temperature gradient operator, $\boldsymbol{K}_T$ is the thermal conductivity matrix, and $\boldsymbol{Q}$ is the prescribed heat-flux vector.

In general, heat conduction is transient. However, under time-invariant thermal boundary conditions, the transient temperature field approaches the steady-state solution, provided a steady state exists. If deformation occurs sufficiently slowly relative to thermal diffusion, the temperature field can be approximated by this steady-state solution. Rate-independent elastoplasticity has no intrinsic material time scale; the load steps represent pseudo-time increments used to discretize the loading path in a quasi-static analysis. Accordingly, the present formulation assumes that the structure reaches a steady-state temperature field before being subjected to quasi-static loading.

2.2 Elastoplastic problem

The elastoplastic problem can be formulated as a boundary value problem in which the equilibrium equation and the constitutive law are coupled. Specifically, the Cauchy stress $\boldsymbol{\sigma}$ in the equilibrium equation is not an independent variable but depends on the deformation history through the constitutive law and the evolution of internal variables. Consequently, the equilibrium problem cannot be formulated independently of the constitutive law, and both must be solved as a coupled system. This section

presents the equilibrium equation, boundary conditions, their discretization, and the numerical solution procedure. The constitutive definition of $\boldsymbol{\sigma}$ is provided in Section 3.

The equilibrium equation of the quasi-static problem is as follows:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} & \text{in } \Omega, \\ \boldsymbol{u} = \bar{\boldsymbol{u}} & \text{on } \Gamma_u, \\ \boldsymbol{\sigma}\boldsymbol{n} = \bar{\boldsymbol{t}} & \text{on } \Gamma_t, \end{cases} \tag{4}$$

where $\boldsymbol{u}$ is the displacement vector and Ω is the domain occupied by the solid. Body forces are neglected for simplicity. The prescribed displacement $\bar{\boldsymbol{u}}$ is imposed on the Dirichlet boundary Γ_u, while the prescribed traction vector $\bar{\boldsymbol{t}}$ is applied on the Neumann boundary Γ_t. The weak form of Eq. (4) is given by

$$\int_\Omega \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} \, \mathrm{d}\Omega = \int_{\Gamma_t} \bar{\boldsymbol{t}} \cdot \delta\boldsymbol{u} \, \mathrm{d}\Gamma. \tag{5}$$

The infinitesimal strain is given by $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^{\mathsf{T}})$. Discretizing the space by the finite element method gives the following discrete equations:

$$\begin{cases} \boldsymbol{F}^{\mathrm{int}}(\boldsymbol{U}) = \boldsymbol{F}^{\mathrm{ext}} \\ \boldsymbol{F}^{\mathrm{int}} = \displaystyle\int_\Omega \boldsymbol{B}_U^{\mathsf{T}} \boldsymbol{\sigma} \, \mathrm{d}\Omega \\ \boldsymbol{F}^{\mathrm{ext}} = \displaystyle\int_{\Gamma_t} \boldsymbol{N}_U^{\mathsf{T}} \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma, \end{cases} \tag{6}$$

where $\boldsymbol{F}^{\mathrm{int}}$ and $\boldsymbol{F}^{\mathrm{ext}}$ denote the internal and external force vectors, respectively, $\boldsymbol{U}$ is the global nodal displacement vector, $\boldsymbol{N}_U$ is the shape function matrix, and $\boldsymbol{B}_U$ is the strain–displacement matrix.

Because the stress in an elastoplastic problem depends on the deformation history, the loading path is discretized into N load steps, and the equilibrium solution is determined at each step n. Defining the residual at step n as $\boldsymbol{R}_n(\boldsymbol{U}) = \boldsymbol{F}^{\mathrm{int}}(\boldsymbol{U}) - \boldsymbol{F}_n^{\mathrm{ext}}$, the Newton–Raphson method obtains the solution by solving the following linearized equation:

$$\left.\frac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{U}}\right|_{\boldsymbol{U}^{(k)}} \Delta\boldsymbol{U} = -\boldsymbol{R}_n(\boldsymbol{U}^{(k)}), \tag{7}$$

where k denotes the Newton–Raphson iteration number. The displacement is updated according to $\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} + \Delta\boldsymbol{U}$. The stress $\boldsymbol{\sigma}$ appearing in the equilibrium equation above is determined by the constitutive law and the evolution of internal variables. Specifically, at each load step, the stress is implicitly determined through a return-mapping algorithm that enforces consistency with the yield surface by incorporating the yield condition, flow rule, and hardening law. Consequently, global equilibrium in Eq. (6) and local constitutive consistency are coupled and must be satisfied together. The outer Newton–Raphson iteration enforces global equilibrium, while the return-mapping algorithm performs local iterative calculations at each integration point to

enforce constitutive consistency. Accordingly, the equilibrium solution and internal-variable updates at load step n are organized as follows:

$$\begin{cases} \boldsymbol{R}_n(\boldsymbol{u}_n;\, \boldsymbol{h}_{n-1}, \boldsymbol{T}) = \boldsymbol{0} \\ \boldsymbol{h}_n = \boldsymbol{H}_n(\boldsymbol{u}_n;\, \boldsymbol{h}_{n-1}, \boldsymbol{T}), \end{cases} \tag{8}$$

where $\boldsymbol{u}_n$ denotes the unknown nodal displacement vector at load step n, $\boldsymbol{h}_n$ denotes the internal variables at all integration points, and $\boldsymbol{T}$ denotes the temperature field described in Section 2.1. The semicolon separates the unknowns to be determined at the current step (left) from the prescribed inputs (right); this notation is used throughout the paper. $\boldsymbol{H}_n$ denotes the set of return-mapping conditions at all integration points.

3 Thermo-elastoplastic model

The thermo-elastoplastic model couples the equilibrium equation, Eq. (4), with the constitutive law, yield condition, flow rule, hardening law, and evolution equations of internal variables. The relations below are not a separate system of equations that holds independently of the equilibrium equation; together with the equilibrium equation, they constitute the coupled system of the elastoplastic problem.

3.1 Constitutive law

An isotropic von Mises elastoplastic model with isotropic hardening was adopted[31]. The total strain $\boldsymbol{\varepsilon}$ is additively decomposed into elastic strain $\boldsymbol{\varepsilon}^e$, plastic strain $\boldsymbol{\varepsilon}^p$, and thermal strain $\boldsymbol{\varepsilon}^{\mathsf{th}}$ as follows:

$$\begin{aligned} \boldsymbol{\varepsilon} &= \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p + \boldsymbol{\varepsilon}^{\mathsf{th}} \\ \boldsymbol{\varepsilon}^{\mathsf{th}} &= \alpha\,(T - T_0)\,\boldsymbol{I}, \end{aligned} \tag{9}$$

where α is the coefficient of linear thermal expansion, T_0 is the reference temperature, and $\boldsymbol{I}$ is the second-order identity tensor. The temperature T is obtained by solving the heat conduction equation (3). The stress is determined from the elastic strain using Hooke's law:

$$\begin{aligned} \boldsymbol{\sigma} &= \mathbb{C} : \boldsymbol{\varepsilon}^e \\ &= \mathbb{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) - \alpha\,(T - T_0)\,\mathbb{C} : \boldsymbol{I}, \end{aligned} \tag{10}$$

where $\mathbb{C}$ is the fourth-order isotropic elasticity tensor. Factoring out Young's modulus E obtains

$$\begin{aligned} \mathbb{C} &= E\,\bar{\mathbb{C}} \\ \bar{\mathbb{C}} &= \frac{1}{3(1-2\nu)}\,\boldsymbol{I} \otimes \boldsymbol{I} + \frac{1}{1+\nu}\,\mathbb{I}^{\mathrm{dev}}, \end{aligned} \tag{11}$$

where ν is Poisson's ratio and $\mathbb{I}^{\mathrm{dev}}$ is the fourth-order tensor that extracts the deviatoric part. The tensor $\bar{\mathbb{C}}$ depends only on Poisson's ratio. Introducing the thermal stress coefficient $\Xi = E\alpha$[32] gives $\alpha\,\mathbb{C} = \Xi\,\bar{\mathbb{C}}$, and Eq. (10) becomes

$$\begin{aligned}\boldsymbol{\sigma} &= \mathbb{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) - \Xi\,(T - T_0)\,\bar{\mathbb{C}} : \boldsymbol{I} \\ &= \mathbb{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) - \frac{\Xi\,(T - T_0)}{1 - 2\nu}\,\boldsymbol{I}.\end{aligned} \tag{12}$$

The thermal stress affects only the hydrostatic component of the stress. In this study, Young's modulus E_i and the thermal stress coefficient $\Xi_i = E_i\alpha_i$ for each material are interpolated separately, as described in Section 4.1. The constitutive law is then given by Eq. (12). For problems without thermo-elastoplastic effects, $\Xi = 0$. The yield function f is defined below based on the von Mises yield criterion with isotropic hardening.

$$\begin{cases} f(\boldsymbol{\sigma}, \bar{\alpha}) = q(\boldsymbol{\sigma}) - \sigma_y(\bar{\alpha}) \\ q(\boldsymbol{\sigma}) = \sqrt{\dfrac{3}{2}}\,\|\boldsymbol{s}\| \\ \boldsymbol{s} = \mathbb{I}^{\mathrm{dev}} : \boldsymbol{\sigma}, \end{cases} \tag{13}$$

where q is the von Mises equivalent stress, $\boldsymbol{s}$ is the deviatoric stress, σ_y is the yield stress, and $\bar{\alpha}$ is the equivalent plastic strain. Introducing the plastic multiplier γ, the associated flow rule defines the plastic strain rate and the equivalent plastic strain rate as follows:

$$\begin{cases} \dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma}\,\dfrac{\partial f}{\partial \boldsymbol{\sigma}} = \dot{\gamma}\sqrt{\dfrac{3}{2}}\,\dfrac{\boldsymbol{s}}{\|\boldsymbol{s}\|} \\ \dot{\bar{\alpha}} = \dot{\gamma}. \end{cases} \tag{14}$$

The yield function and plastic multiplier satisfy the following Karush–Kuhn–Tucker conditions to determine whether plastic flow occurs:

$$\begin{cases} f \le 0 \\ \dot{\gamma} \ge 0 \\ f\,\dot{\gamma} = 0, \end{cases} \tag{15}$$

3.2 Nonlinear hardening law

The yield stress of isotropic hardening is given by the Voce law[33]:

$$\sigma_y(\bar{\alpha}) = \sigma_{y0} + (\sigma_\infty - \sigma_{y0})\left(1 - e^{-b\bar{\alpha}}\right), \tag{16}$$

where σ_{y0} is the initial yield stress, σ_∞ is the saturation stress, and b controls the rate of saturation. With the Voce hardening law, the plastic multiplier cannot generally be obtained in closed form during the return-mapping procedure described in Section 3.3; therefore, a local Newton–Raphson iteration is required.

3.3 Return mapping

Return mapping is used to determine the stress at each integration point of the finite element analysis[31]. The elastic trial state is defined according to the total strain $\boldsymbol{\varepsilon}_{n+1}$ of step $n+1$ and the internal variables $(\boldsymbol{\varepsilon}_n^p, \bar{\alpha}_n)$ of the previous step:

$$\begin{cases} \boldsymbol{\sigma}_{n+1}^{\mathrm{tr}} = \mathbb{C} : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n^p) - \dfrac{\Xi\,(T - T_0)}{1-2\nu}\,\boldsymbol{I} \\ q_{n+1}^{\mathrm{tr}} = \sqrt{\dfrac{3}{2}}\,\|\boldsymbol{s}_{n+1}^{\mathrm{tr}}\|. \end{cases} \tag{17}$$

If $f(\boldsymbol{\sigma}_{n+1}^{\mathrm{tr}}, \bar{\alpha}_n) \le 0$, the step is elastic, and the trial state is accepted as the solution. Otherwise, the discretized governing equations are solved:

$$\begin{cases} \boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p + \Delta\gamma\sqrt{\dfrac{3}{2}}\,\boldsymbol{\nu}_{n+1} \\ \bar{\alpha}_{n+1} = \bar{\alpha}_n + \Delta\gamma \\ \boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{\mathrm{tr}} - 2\mu\Delta\gamma\sqrt{\dfrac{3}{2}}\,\boldsymbol{\nu}_{n+1} \\ q(\boldsymbol{\sigma}_{n+1}) - \sigma_y(\bar{\alpha}_{n+1}) = 0, \end{cases} \tag{18}$$

where $\Delta\gamma$ is the discretized plastic multiplier, $\boldsymbol{\nu}_{n+1} = \boldsymbol{s}_{n+1}^{\mathrm{tr}}/\|\boldsymbol{s}_{n+1}^{\mathrm{tr}}\|$ is the flow direction, and $\mu = E/[2(1+\nu)]$ is the shear modulus. For the von Mises model, the flow direction is deviatoric, so the deviatoric trial stress retains its direction during the return mapping. Consequently, $q(\boldsymbol{\sigma}_{n+1}) = q_{n+1}^{\mathrm{tr}} - 3\mu\Delta\gamma$, and Eq. (18) reduces to the scalar equation

$$\tilde{\Phi}(\Delta\gamma) = q_{n+1}^{\mathrm{tr}} - 3\mu\Delta\gamma - \sigma_y(\bar{\alpha}_n + \Delta\gamma) = 0, \tag{19}$$

where $\Delta\gamma$ is the only unknown. Owing to the Voce hardening term in Eq. (16), Eq. (19) is solved using a local Newton–Raphson method. Once $\Delta\gamma$ is determined, the stress and internal variables are updated using Eq. (18). The consistent tangent used in Eq. (7) is given in the literature[31]:

$$\begin{cases} \mathbb{C}^{\mathrm{ep}} = \kappa\,\boldsymbol{I}\otimes\boldsymbol{I} + 2\mu\theta\,\mathbb{I}^{\mathrm{dev}} - 2\mu\bar{\theta}\,\boldsymbol{\nu}_{n+1}\otimes\boldsymbol{\nu}_{n+1} \\ \theta = 1 - \dfrac{3\mu\Delta\gamma}{q_{n+1}^{\mathrm{tr}}} \\ \bar{\theta} = \dfrac{1}{1 + \sigma_y'(\bar{\alpha}_{n+1})/(3\mu)} - (1-\theta), \end{cases} \tag{20}$$

where $\kappa = E/[3(1-2\nu)]$ is the bulk modulus. Because the thermal stress is independent of the displacement, it does not contribute to the consistent tangent. The internal variables at all integration points are collected into $\boldsymbol{h}_n$ in Eq. (8), while applying Eq. (18) at all integration points defines the map $\boldsymbol{H}_n$.

4 Topology optimization

4.1 Multi-material topology optimization

Density-based topology optimization reformulates the problem of finding an optimal structure as one of determining the optimal material distribution within a fixed design domain. To accommodate multiple materials, fictitious densities representing the presence of each material at spatial position $\boldsymbol{X}$ are defined as follows:

$$0 \leq x_i(\boldsymbol{X}) \leq 1 \qquad (i = 1, 2, \ldots, N_m), \tag{21}$$

where N_m is the number of candidate materials, excluding the void phase. A value of $x_i(\boldsymbol{X}) = 1$ indicates that position $\boldsymbol{X}$ is fully occupied by material i. The design domain is discretized into N_e elements, with element-wise design variables x_{ei}, where $e = 1, 2, \ldots, N_e$. Thus, the total number of design variables is $N_m \times N_e$.

To mitigate checkerboard patterns and mesh dependence, a density filter[34, 35] is applied:

$$\begin{cases} \tilde{x}_{ei} = \dfrac{\sum_{j \in \mathcal{N}_e} w_{ej} V_j \, x_{ji}}{\sum_{j \in \mathcal{N}_e} w_{ej} V_j} \\ w_{ej} = r_{\text{fil}} - \|\boldsymbol{X}_e - \boldsymbol{X}_j\|, \end{cases} \tag{22}$$

where $\tilde{x}_{ei}$ is the filtered design variable, $\boldsymbol{X}_e$ is the centroid of element e, V_j is the volume of element j, and $\mathcal{N}_e$ is the set of elements whose centroids lie within a distance r_{fil} of $\boldsymbol{X}_e$. Because density filtering tends to produce intermediate density values, the Heaviside projection[36] is applied to obtain the physical density $\tilde{\tilde{x}}_{ei}$ and promote a black-and-white design:

$$\tilde{\tilde{x}}_{ei} = \frac{\tanh(\beta\eta) + \tanh[\beta(\tilde{x}_{ei} - \eta)]}{\tanh(\beta\eta) + \tanh[\beta(1 - \eta)]}, \tag{23}$$

where η and β are parameters that represent the threshold and sharpness, respectively, of the projection.

The discrete material optimization (DMO) method[37] is used for material interpolation in multi-material topology optimization. In the elastoplastic model, the parameters of the hardening law are interpolated with the design variables in addition to Young's modulus E. The DMO method interpolates the material properties $\phi \in \{E, \sigma_{y0}, \sigma_\infty, b, \Xi, k\}$ as follows:

$$\phi = \phi_0 + \sum_{i=1}^{N_m} \left[\tilde{\tilde{x}}_{ei}^{\,p} \prod_{\substack{j=1 \\ j \neq i}}^{N_m} \left(1 - \tilde{\tilde{x}}_{ej}^{\,p}\right) \right] (\phi_i - \phi_0), \tag{24}$$

where ϕ_0 and ϕ_i denote the properties of the void phase and material i, respectively, and p is the penalization exponent that controls intermediate density values. The Young's modulus of the void phase, E_0, is assigned a small positive value for numerical regularization. A separate penalization exponent p is specified for each property.

4.2 Optimization problem

To maximize the total energy received by the structure along the loading history, the optimization problem is defined as follows:

$$\max_{\boldsymbol{x}} \quad f(\boldsymbol{x}) = \sum_{n=1}^{N} \int_{\Omega} \boldsymbol{\sigma}_n : (\boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_{n-1}) \, \mathrm{d}\Omega \tag{25}$$

$$\text{s.t.} \quad g_i(\boldsymbol{x}) = \frac{1}{v_d} \int_{\Omega} \tilde{\bar{x}}_i \, \mathrm{d}\Omega - \bar{v}_i \leq 0 \qquad (i = 1, \ldots, N_m), \tag{26}$$

$$\boldsymbol{K}_T(\boldsymbol{x}) \boldsymbol{T} = \boldsymbol{Q}, \tag{27}$$

$$\boldsymbol{R}_n(\boldsymbol{u}_n;\, \boldsymbol{x}, \boldsymbol{h}_{n-1}, \boldsymbol{T}) = \boldsymbol{0}, \tag{28}$$

$$\boldsymbol{h}_n = \boldsymbol{H}_n(\boldsymbol{u}_n;\, \boldsymbol{x}, \boldsymbol{h}_{n-1}, \boldsymbol{T}), \tag{29}$$

$$\boldsymbol{0} < \boldsymbol{x}_{\min} \leq \boldsymbol{x}_i \leq \boldsymbol{1}, \tag{30}$$

The objective function f is defined as the sum, over all load steps, of the stress–strain increment product integrated over the entire domain. Under displacement control, this quantity corresponds to the work done by the reaction force through the prescribed displacement and is represented by the area under the load–displacement curve. Maximizing f yields a structure that maintains resistance to the prescribed deformation even after substantial plastic deformation. The constraint g_i limits the volume of each material i, where v_d is the volume of the design domain and $\bar{v}_i$ is the upper bound on the volume fraction of material i. Eqs. (27), (28), and (29) represent the governing equations for heat conduction (Eq. (3)), elastoplastic equilibrium (Eq. (6)), and internal-variable updates (Eq. (8)), respectively. Through the material interpolation in Eq. (24), these equations depend on the design variables $\boldsymbol{x}$ and are expressed as $\boldsymbol{K}_T(\boldsymbol{x})$, $\boldsymbol{R}_n(\boldsymbol{u}_n; \boldsymbol{x}, \boldsymbol{h}_{n-1}, \boldsymbol{T})$, and $\boldsymbol{H}_n(\boldsymbol{u}_n; \boldsymbol{x}, \boldsymbol{h}_{n-1}, \boldsymbol{T})$, respectively. Their solutions are the temperature $\boldsymbol{T}$, displacement $\boldsymbol{u}_n$, and internal variables $\boldsymbol{h}_n$, respectively, each of which can also be regarded as a function of $\boldsymbol{x}$. The lower bound $\boldsymbol{x}_{\min}$ is imposed on the design variables to ensure that finite element analysis remains feasible in regions representing void.

5 Proposed method

The proposed method solves each governing equation and applies AD at the converged point where the solution has been obtained. Thus, the sensitivity analysis is independent of the objective function and physical model without requiring the formulation of the analytical adjoint variable method. For multiphysics problems with path dependence and internal variables such as elastoplastic models, the derivation of the adjoint equations[1, 7–9, 16] for each physical model and the writing of a program to solve them become unnecessary. Consequently, topology optimization can be performed merely by defining the optimization problem to be solved without needing to consider the sensitivity analysis. Mathematically, the proposed method is equivalent

to the discrete adjoint variable method, but because it is derived by AD, no formulation specific to each optimization problem is needed. The computations shown below are carried out automatically by the AD library, and no analytical formulation by symbolic expansion is required for the sensitivity analysis. Nevertheless, their internal computation is explained to clarify the workings of the proposed method. The general form is given first in Sections 5.1 to 5.3, the computational flow is given in Section 5.4, and its application to the thermo-elastoplastic model is given in Section 5.5.

5.1 Generalization of the forward analysis

Forward analysis solves a problem by evaluating a sequence of governing equations. Each system is solved in a separate analysis step, indexed by $k = 1, 2, \ldots, N_{\mathsf{a}}$. At analysis step k, the unknown $\boldsymbol{y}_k$ is determined by solving

$$\begin{cases} \boldsymbol{R}_k(\boldsymbol{y}_k;\, \boldsymbol{z}_k) = \boldsymbol{0} \\ \boldsymbol{z}_k = \big(\boldsymbol{x},\ \{\boldsymbol{w}_j\}_{j\in\mathcal{P}(k)}\big), \end{cases} \tag{31}$$

where $\boldsymbol{R}_k$ is the residual, and $\boldsymbol{z}_k$ is the input of analysis step k comprising the design variables $\boldsymbol{x}$ and outputs of the set $\mathcal{P}(k)$ of preceding analysis steps. The output of analysis step k is given by

$$\begin{cases} \boldsymbol{w}_k = \boldsymbol{G}_k(\boldsymbol{z}_k) = \big(\boldsymbol{y}_k,\ \boldsymbol{h}_k,\ a_k\big) \\ \boldsymbol{h}_k = \boldsymbol{H}_k(\boldsymbol{y}_k;\, \boldsymbol{z}_k) \\ a_k = \varphi_k(\boldsymbol{y}_k;\, \boldsymbol{z}_k) + \displaystyle\sum_{j\in\mathcal{P}(k)} a_j, \end{cases} \tag{32}$$

where $\boldsymbol{h}_k$ denotes the set of internal variables constructed from the converged solution. If no internal variables are defined, as in the heat conduction equation, $\boldsymbol{h}_k$ is an empty vector. The quantity φ_k is the contribution of analysis step k to the objective function, and a_k is the cumulative contribution through that step. The objective function is therefore given by the accumulated value at the final analysis step, $f = a_{N_{\mathsf{a}}}$. In the following formulation, $\boldsymbol{R}_k$, $\boldsymbol{H}_k$, and φ_k are evaluated as functions of $\boldsymbol{y}_k$ and $\boldsymbol{z}_k$. Weakly coupled multiphysics problems, in which multiple governing equations are solved independently, fit naturally within this formulation. For strongly coupled multiphysics problems, in which multiple physical fields are solved simultaneously as a single system, the coupled system is treated as one analysis step. Similarly, transient or time-dependent governing equations are accommodated by treating each time step as one analysis step. Thus, Eqs. (31) and (32) retain their generality.

5.2 Map at the converged point and implicit differentiation

Let $\boldsymbol{y}_k^*$ denote the converged solution at analysis step k, and let $\boldsymbol{K}_k^* = \partial\boldsymbol{R}_k/\partial\boldsymbol{y}_k(\boldsymbol{y}_k^*, \boldsymbol{z}_k)$ denote the tangent matrix evaluated at that solution. Define $\boldsymbol{M}_k = (\boldsymbol{K}_k^*)^{-1}$, which is fixed and independent of $\boldsymbol{y}$ and $\boldsymbol{z}_k$. The following map can then be

constructed for analysis step k:

$$\boldsymbol{S}_k(\boldsymbol{y};\, \boldsymbol{z}_k) = \boldsymbol{y} - \boldsymbol{M}_k \boldsymbol{R}_k(\boldsymbol{y};\, \boldsymbol{z}_k), \tag{33}$$

Because $\boldsymbol{R}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k) = \boldsymbol{0}$, then $\boldsymbol{S}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k) = \boldsymbol{y}_k^*$. At the converged point, the following holds:

$$\begin{cases} \dfrac{\partial \boldsymbol{S}_k}{\partial \boldsymbol{y}} = \boldsymbol{I} - \boldsymbol{M}_k \dfrac{\partial \boldsymbol{R}_k}{\partial \boldsymbol{y}_k} = \boldsymbol{0} \\ \dfrac{\partial \boldsymbol{S}_k}{\partial \boldsymbol{z}_k} = -\boldsymbol{M}_k \dfrac{\partial \boldsymbol{R}_k}{\partial \boldsymbol{z}_k} \end{cases}. \tag{34}$$

This map corresponds to the fixed-point formulation of the convergent iteration $\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{y}, \boldsymbol{u})$ described by Christianson[23]. When Newton's method is used with the consistent tangent, $\partial \boldsymbol{S}_k / \partial \boldsymbol{y} = \boldsymbol{0}$ at the fixed point, so the adjoint iteration converges in a single step. Even when $\boldsymbol{y}_k^*$ is treated as constant with respect to the input and $\boldsymbol{S}_k$ is differentiated only once, the resulting partial derivative of the output agrees with the derivative of the discrete solution, as follows from the implicit function theorem:

$$\frac{\mathrm{d}\boldsymbol{y}_k^*}{\mathrm{d}\boldsymbol{z}_k} = -(\boldsymbol{K}_k^*)^{-1} \frac{\partial \boldsymbol{R}_k}{\partial \boldsymbol{z}_k} = \frac{\partial \boldsymbol{S}_k}{\partial \boldsymbol{z}_k}. \tag{35}$$

Neglecting the derivative of $\boldsymbol{M}_k$ with respect to $\boldsymbol{z}_k$ does not introduce a gradient error at the converged solution because $\boldsymbol{R}_k = \boldsymbol{0}$ there. In practice, the forward analysis leaves a residual $\boldsymbol{R}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k)$ within the convergence tolerance. The resulting gradient error is of the same order as the residual[23], so the gradient accuracy is comparable to that of the forward solution. The output of an analysis step includes not only $\boldsymbol{y}_k$, but also the internal variables $\boldsymbol{h}_k = \boldsymbol{H}_k(\boldsymbol{y}_k^+;\, \boldsymbol{z}_k)$ and the objective contribution $\varphi_k(\boldsymbol{y}_k^+;\, \boldsymbol{z}_k)$, both evaluated at the corrected state $\boldsymbol{y}_k^+ = \boldsymbol{S}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k)$. Although $\boldsymbol{y}_k^+$ equals $\boldsymbol{y}_k^*$ at convergence, its derivative is given by Eq. (35). If $\boldsymbol{h}_k$ is instead evaluated using the fixed value $\boldsymbol{y}_k^*$, the term $(\partial \boldsymbol{H}_k / \partial \boldsymbol{y}_k)(\mathrm{d}\boldsymbol{y}_k / \mathrm{d}\boldsymbol{z}_k)$ is omitted, and the gradient contribution propagated through the history variables is lost. Thus, the complete map $\boldsymbol{w}_k = \boldsymbol{G}_k(\boldsymbol{z}_k)$ for analysis step k is represented by evaluating $\boldsymbol{S}_k$, $\boldsymbol{H}_k$, and φ_k once at the converged point.

The above result holds under two conditions. First, $\boldsymbol{R}_k$ must be continuously differentiable with respect to $\boldsymbol{y}_k$ and $\boldsymbol{z}_k$ in a neighborhood of the converged point. At nonsmooth points, such as when an integration point lies exactly on the yield surface, the gradient corresponds to the branch selected at convergence. Care is also required when verifying the gradient using finite differences. These considerations are the same as those encountered with the analytical adjoint variable method. Second, $\boldsymbol{K}_k^*$ must be the exact Jacobian evaluated at the converged point and must be nonsingular. If the tangent is inexact and $\partial \boldsymbol{S}_k / \partial \boldsymbol{y} \neq \boldsymbol{0}$, a single differentiation is generally insufficient to obtain the exact gradient, resulting in an error. However, an exact tangent stiffness (i.e., the consistent tangent) is typically used in nonlinear finite element analysis to achieve efficient convergence with the Newton–Raphson method. Thus, this condition is generally satisfied in standard forward analyses.

5.3 Backward propagation

Let $\bar{\boldsymbol{w}}_k = \partial f / \partial \boldsymbol{w}_k$ be the adjoint of the output of each analysis step. Traversing the analysis steps in reverse order gives

$$\begin{cases} \bar{\boldsymbol{w}}_j = \displaystyle\sum_{k\,:\,j \in \mathcal{P}(k)} \left(\frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{w}_j} \right)^{\mathsf{T}} \bar{\boldsymbol{w}}_k \\ \dfrac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \displaystyle\sum_{k=1}^{N_{\mathrm{a}}} \left(\frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{x}} \right)^{\mathsf{T}} \bar{\boldsymbol{w}}_k. \end{cases} \tag{36}$$

The initial value sets the $a_{N_{\mathrm{a}}}$ component of $\bar{\boldsymbol{w}}_{N_{\mathrm{a}}}$ to 1 and all other components to 0. For analysis step k, define the adjoint of the output as $\bar{\boldsymbol{w}}_k = (\bar{\boldsymbol{y}}_k, \bar{\boldsymbol{h}}_k, \bar{a}_k)$. Tracing the computations in Eqs. (32) and (33) backward, the adjoint of $\boldsymbol{y}_k^+$ is first accumulated as follows:

$$\hat{\boldsymbol{y}}_k = \bar{\boldsymbol{y}}_k + \left(\frac{\partial \boldsymbol{H}_k}{\partial \boldsymbol{y}_k} \right)^{\mathsf{T}} \bar{\boldsymbol{h}}_k + \bar{a}_k \left(\frac{\partial \varphi_k}{\partial \boldsymbol{y}_k} \right)^{\mathsf{T}}. \tag{37}$$

As the solution update in $\boldsymbol{S}_k$ is traced backward, the following adjoint equation is solved:

$$\boldsymbol{K}_k^{*\mathsf{T}} \boldsymbol{\zeta}_k = \hat{\boldsymbol{y}}_k. \tag{38}$$

The adjoint of the input is given by

$$\bar{\boldsymbol{z}}_k = \left(\frac{\partial \boldsymbol{H}_k}{\partial \boldsymbol{z}_k} \right)^{\mathsf{T}} \bar{\boldsymbol{h}}_k + \bar{a}_k \left(\frac{\partial \varphi_k}{\partial \boldsymbol{z}_k} \right)^{\mathsf{T}} - \left(\frac{\partial \boldsymbol{R}_k}{\partial \boldsymbol{z}_k} \right)^{\mathsf{T}} \boldsymbol{\zeta}_k. \tag{39}$$

Splitting this expression into contributions from the design variables and the outputs of the preceding analysis steps yields the terms in Eq. (36). For the accumulated objective function, $\bar{a}_j = \bar{a}_k$ is added.

Eq. (38) has the same form as the discrete adjoint equation for each analysis step, with $\boldsymbol{\zeta}_k$ corresponding to the adjoint variable of analysis step k. However, this equation is not derived analytically through symbolic expansion; instead, it is obtained automatically by tracing the solution update in Eq. (33) backward. In the analytical adjoint variable method[1, 7–9, 16], this equation and each term in Eq. (39) are derived analytically for each physical model. In the proposed method, this derivation is automated. Moreover, when reverse-mode AD is applied to the evaluation of $\boldsymbol{G}_k$, the transposed products in Eq. (39) are computed without explicitly forming the Jacobian matrices. All partial derivatives in Eqs. (36) to (39) are obtained by evaluating $\boldsymbol{R}_k$, $\boldsymbol{H}_k$, and φ_k. Consequently, the adjoint equations need not be rederived when the constitutive law, physical coupling, or objective function changes. Exact sensitivities are obtained automatically for any objective function and physical model whose forward analysis can be evaluated.

Another advantage of the proposed method is that it computes sensitivities at a computational cost comparable to that of the analytical adjoint variable method, without requiring the adjoint equations to be derived separately for each physical model.

In contrast, applying reverse-mode AD directly to every operation in the forward analysis[13] requires recording the entire computational process, including the load steps, Newton–Raphson iterations for convergence, and tangent stiffness matrix assembly, as a computational graph. Storing this graph and its intermediate data can require an enormous amount of memory, while the additional computational overhead can make this approach slower than the analytical adjoint variable method. Checkpointing[14, 15] reduces memory consumption but increases computational cost because the forward analysis must be recomputed. Because the recording for a single analysis step includes all its Newton iterations, applying checkpointing to nonlinear, history-dependent three-dimensional problems remains challenging. In contrast, the proposed method begins with the converged state obtained by solving the governing equations, allowing the forward analysis, including the Newton iterations, to be excluded from the scope of AD. Furthermore, because a tape is recorded for each analysis step in Eq. (31) and discarded after being traced backward, only the tape for a single analysis step needs to be stored at any given time. Thus, the memory required to record the computational graph does not depend on the number of analysis steps. Only the storage required for the converged states scales with the number of analysis steps, and this storage is also required by the analytical adjoint variable method. Consequently, the proposed method substantially reduces the computational cost compared with recording every operation of the forward analysis using AD, while achieving sensitivities at a computational cost comparable to that of the analytical adjoint variable method without requiring model-specific adjoint formulations.

5.4 Computational flow

The computational workflow of the proposed sensitivity analysis is shown in Algorithm 1. Hereafter, the sequence of operations recorded by reverse-mode AD is referred to as the tape[10]. The procedure described in this section is independent of any particular AD library and can be implemented using any library that supports reverse-mode AD.

In the forward analysis, each analysis step is solved using the standard solution procedure, and all converged solutions are stored. This stage is not subject to AD. Storing the converged solutions is also required in the conventional analytical adjoint variable method. In contrast, AD applied to every operation of the forward analysis also records this stage, potentially requiring an enormous amount of memory. The proposed method avoids this overhead.

During sensitivity computation, the analysis steps are traversed in reverse order. For each step, the following operations are performed. First, the tangent stiffness matrix is assembled at the stored converged solution $\boldsymbol{y}_k^*$:

$$\boldsymbol{K}_k^* = \frac{\partial \boldsymbol{R}_k}{\partial \boldsymbol{y}_k}(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k). \tag{40}$$

Here, $\boldsymbol{K}_k^*$ is excluded from AD and treated as a constant input. With $\boldsymbol{y}_k^*$ held constant and the analysis-step input $\boldsymbol{z}_k$ designated as the AD input, the following computation

is executed once and recorded on the tape:

$$\begin{cases} \boldsymbol{r}_k = \boldsymbol{R}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k) \\ \boldsymbol{K}_k^*\, \boldsymbol{v}_k = \boldsymbol{r}_k \\ \boldsymbol{y}_k^+ = \boldsymbol{y}_k^* - \boldsymbol{v}_k \\ \boldsymbol{h}_k = \boldsymbol{H}_k(\boldsymbol{y}_k^+;\, \boldsymbol{z}_k) \\ a_k = \varphi_k(\boldsymbol{y}_k^+;\, \boldsymbol{z}_k) + \displaystyle\sum_{j \in \mathcal{P}(k)} a_j. \end{cases} \tag{41}$$

The first three equations define the map $\boldsymbol{S}_k$ in Eq. (33), while the fourth and fifth equations define the output in Eq. (32). The recorded operations comprise assembly of the residual $\boldsymbol{R}_k$, a single solution update using the fixed tangent $\boldsymbol{K}_k^*$ (i.e., the second and third equations of Eq. (41)), and re-evaluation of the residual at $\boldsymbol{y}_k^+$. The internal variables $\boldsymbol{H}_k$ and the objective contribution φ_k are evaluated together with the stress at each integration point during residual evaluation. Thus, they are obtained simultaneously when the residual is re-evaluated at $\boldsymbol{y}_k^+$.

The solution of the linear system in the second equation of Eq. (41) is recorded on the tape as a single operation with its own differentiation rule. Because $\boldsymbol{K}_k^*$ is constant, the rule mapping the adjoint $\bar{\boldsymbol{v}}_k$ of the output $\boldsymbol{v}_k$ to the adjoint $\bar{\boldsymbol{r}}_k$ of the input $\boldsymbol{r}_k$ is given by

$$\boldsymbol{K}_k^{*\mathsf{T}}\, \bar{\boldsymbol{r}}_k = \bar{\boldsymbol{v}}_k. \tag{42}$$

The linear system is differentiated using this rule rather than by recording the operations performed inside the solver. This is a standard differentiation rule for linear systems, not an adjoint equation specific to a physical model. From the third equation of Eq. (41), $\bar{\boldsymbol{v}}_k = -\hat{\boldsymbol{y}}_k$. Therefore, Eq. (42) is equivalent to Eq. (38), and $\bar{\boldsymbol{r}}_k = -\boldsymbol{\zeta}_k$.

Seeding the recorded tape with the output adjoint $\bar{\boldsymbol{w}}_k = (\bar{\boldsymbol{y}}_k, \bar{\boldsymbol{h}}_k, \bar{a}_k)$ and tracing it backward yields $\bar{\boldsymbol{z}}_k$ in Eq. (39), the adjoint of the input $\boldsymbol{z}_k = (\boldsymbol{x}, \{\boldsymbol{w}_j\}_{j \in \mathcal{P}(k)})$. This adjoint is then split into its components and accumulated:

$$\begin{cases} \dfrac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} \leftarrow \dfrac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} + \left[\bar{\boldsymbol{z}}_k\right]_{\boldsymbol{x}} \\ \bar{\boldsymbol{w}}_j \leftarrow \bar{\boldsymbol{w}}_j + \left[\bar{\boldsymbol{z}}_k\right]_{\boldsymbol{w}_j} \qquad j \in \mathcal{P}(k), \end{cases} \tag{43}$$

where $[\,\cdot\,]_{\boldsymbol{x}}$ and $[\,\cdot\,]_{\boldsymbol{w}_j}$ denote the components corresponding to the design variables and the output of the preceding analysis step j, respectively. Because the tape for each analysis step is discarded after being traced backward, only the tape for a single analysis step needs to be stored at any given time. The outputs of preceding analysis steps are passed to subsequent steps while remaining subject to AD. In the thermo-elastoplastic problem, these outputs include the temperature field, allowing sensitivities to propagate along this path through the dependence of the temperature field on the design variables.

The return mapping (i.e., the local iteration at each integration point) is executed within the tape as in the forward analysis, with each iteration recorded separately.

Algorithm 1 Sensitivity analysis of the proposed method

Require: design variables $\boldsymbol{x}$; $\boldsymbol{R}_k$, $\boldsymbol{H}_k$, φ_k, $\mathcal{P}(k)$ of the analysis steps $k = 1, \dots, N_{\mathsf{a}}$
Ensure: $\mathrm{d}f/\mathrm{d}\boldsymbol{x}$

Forward analysis

for $k = 1, \dots, N_{\mathsf{a}}$ **do**
  $\boldsymbol{z}_k \leftarrow (\boldsymbol{x}, \{\boldsymbol{w}_j\}_{j\in\mathcal{P}(k)})$
  solve $\boldsymbol{R}_k(\boldsymbol{y}_k;\, \boldsymbol{z}_k) = \boldsymbol{0}$ to obtain the converged solution $\boldsymbol{y}_k^*$
  $\boldsymbol{h}_k \leftarrow \boldsymbol{H}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k)$, $a_k \leftarrow \varphi_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k) + \sum_{j\in\mathcal{P}(k)} a_j$
  store $\boldsymbol{w}_k \leftarrow (\boldsymbol{y}_k^*, \boldsymbol{h}_k, a_k)$
end for

Backward propagation

$\mathrm{d}f/\mathrm{d}\boldsymbol{x} \leftarrow \boldsymbol{0}$, $\bar{\boldsymbol{w}}_k \leftarrow \boldsymbol{0}$ $(k = 1, \dots, N_{\mathsf{a}})$, $\bar{a}_{N_{\mathsf{a}}} \leftarrow 1$
for $k = N_{\mathsf{a}}, \dots, 1$ **do**
  $\boldsymbol{K}_k^* \leftarrow \partial\boldsymbol{R}_k/\partial\boldsymbol{y}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k)$ ▷ fixed as a constant
  start recording the tape with $\boldsymbol{z}_k$ as the subject of automatic differentiation
  $\boldsymbol{r}_k \leftarrow \boldsymbol{R}_k(\boldsymbol{y}_k^*;\, \boldsymbol{z}_k)$, solve $\boldsymbol{K}_k^*\boldsymbol{v}_k = \boldsymbol{r}_k$, $\boldsymbol{y}_k^+ \leftarrow \boldsymbol{y}_k^* - \boldsymbol{v}_k$
  $\boldsymbol{h}_k \leftarrow \boldsymbol{H}_k(\boldsymbol{y}_k^+;\, \boldsymbol{z}_k)$, $a_k \leftarrow \varphi_k(\boldsymbol{y}_k^+;\, \boldsymbol{z}_k) + \sum_{j\in\mathcal{P}(k)} a_j$
  stop recording the tape
  give the adjoint of the output $\bar{\boldsymbol{w}}_k$ and trace the tape backward to obtain the adjoint of the input $\bar{\boldsymbol{z}}_k$
  $\mathrm{d}f/\mathrm{d}\boldsymbol{x} \leftarrow \mathrm{d}f/\mathrm{d}\boldsymbol{x} + [\bar{\boldsymbol{z}}_k]_{\boldsymbol{x}}$
  $\bar{\boldsymbol{w}}_j \leftarrow \bar{\boldsymbol{w}}_j + [\bar{\boldsymbol{z}}_k]_{\boldsymbol{w}_j}$ $(j \in \mathcal{P}(k))$
  discard the tape
end for

Because the local Newton method uses the exact slope, the derivative of the converged solution coincides with the derivative of $\Delta\gamma$ obtained using the implicit function theorem[24]. Because the local equation is scalar and requires few iterations, its contribution to the tape is small.

5.5 Example

The application of the proposed method to the thermo-elastoplastic model is presented here as an example. In practice, the formulation below is carried out by AD, so the designer does not have to formulate the sensitivities. Analysis step $k = 1$ corresponds to heat conduction analysis, and analysis step $k = n+1$ corresponds to the elastoplastic analysis of load step $n = 1, \dots, N$, so $N_{\mathsf{a}} = N + 1$. The heat conduction analysis uses Eq. (3):

$$\begin{cases} \boldsymbol{y}_1 = \boldsymbol{T} \\ \boldsymbol{R}_1(\boldsymbol{y}_1;\, \boldsymbol{z}_1) = \boldsymbol{K}_T(\boldsymbol{x})\boldsymbol{T} - \boldsymbol{Q}(\boldsymbol{x}) \\ \varphi_1 = 0 \\ \mathcal{P}(1) = \emptyset. \end{cases} \tag{44}$$

The heat conduction analysis has no internal variables. In analysis step $k = n+1$, the elastoplastic analysis for load step n uses the equilibrium and update of the internal

variables of Eq. (8) and the n-th term of Eq. (25):

$$\begin{cases} \boldsymbol{y}_k = \boldsymbol{u}_n \\ \boldsymbol{R}_k(\boldsymbol{y}_k;\, \boldsymbol{z}_k) = \boldsymbol{R}_n(\boldsymbol{u}_n;\, \boldsymbol{x}, \boldsymbol{h}_{n-1}, \boldsymbol{T}) \\ \boldsymbol{H}_k(\boldsymbol{y}_k;\, \boldsymbol{z}_k) = \boldsymbol{H}_n(\boldsymbol{u}_n;\, \boldsymbol{x}, \boldsymbol{h}_{n-1}, \boldsymbol{T}) \\ \varphi_k = \displaystyle\int_\Omega \boldsymbol{\sigma}_n : (\boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_{n-1})\, \mathrm{d}\Omega \\ \mathcal{P}(k) = \{k-1,\, 1\}. \end{cases} \tag{45}$$

On the right-hand side, $\boldsymbol{R}_n$ and $\boldsymbol{H}_n$ are defined in Eq. (8). They take as inputs the output $\boldsymbol{w}_{k-1}$ of the preceding load step and the output $\boldsymbol{w}_1$ of the heat conduction analysis. The objective function is $f = a_{N+1}$. For problems without thermo-elastoplastic coupling, no heat conduction analysis is required. In this case, load step n corresponds to analysis step $k = n$, with $\mathcal{P}(n) = \{n-1\}$. The dependence on the design variables in $\boldsymbol{z}_k$ is captured by the filter in Eq. (22), the projection in Eq. (23), and the material interpolation in Eq. (24).

The two conditions stated in Section 5.2 are satisfied as follows. The differentiability condition requires the elastic or plastic state at each integration point to remain unchanged in a neighborhood of the converged solution. If an integration point lies exactly on the yield surface, the derivative corresponds to the branch selected at convergence. Regarding the tangent condition, the heat conduction analysis is linear, so $\boldsymbol{K}_1^* = \boldsymbol{K}_T$ is the exact tangent. The elastoplastic analysis uses the consistent tangent defined in Eq. (20).

The backward propagation traverses the elastoplastic analysis in reverse load-step order, $n = N, N-1, \ldots, 1$. At each step, the adjoint $\bar{\boldsymbol{w}}_n$ is propagated to the preceding load step, and the adjoint contributions to the temperature field are accumulated as follows:

$$\bar{\boldsymbol{T}} = \sum_{n=1}^{N} \left(\frac{\partial \boldsymbol{G}_{n+1}}{\partial \boldsymbol{T}} \right)^{\mathsf{T}} \bar{\boldsymbol{w}}_{n+1}, \tag{46}$$

where $\partial \boldsymbol{G}_{n+1}/\partial \boldsymbol{T}$ represents the dependence arising from the thermal stress defined in Eq. (17). Finally, the heat conduction analysis is traced backward. Because this analysis step has neither internal variables nor an objective function contribution, Eqs. (38) and (39) reduce to

$$\begin{cases} \boldsymbol{K}_T^{\mathsf{T}} \boldsymbol{\lambda}_T = \bar{\boldsymbol{T}} \\ \left(\dfrac{\partial \boldsymbol{G}_1}{\partial \boldsymbol{x}} \right)^{\mathsf{T}} \bar{\boldsymbol{w}}_1 = \left(\dfrac{\partial \boldsymbol{Q}}{\partial \boldsymbol{x}} \right)^{\mathsf{T}} \boldsymbol{\lambda}_T - \left[\dfrac{\partial (\boldsymbol{K}_T(\boldsymbol{x}) \boldsymbol{T})}{\partial \boldsymbol{x}} \right]^{\mathsf{T}} \boldsymbol{\lambda}_T, \end{cases} \tag{47}$$

where $\boldsymbol{\lambda}_T = \boldsymbol{\zeta}_1$. The first equation of Eq. (47) follows from the general formulation in Eq. (33) and reduces to the adjoint equation for heat conduction. Thus, no separate thermo-structural coupled adjoint equation needs to be derived. The dependence of the thermal conductivity on the design variables is captured by $\partial (\boldsymbol{K}_T(\boldsymbol{x}) \boldsymbol{T})/\partial \boldsymbol{x}$.

Table 1: Material properties

Material	E [MPa]	ν	σ_{y0} [MPa]	σ_∞ [MPa]	b	Ξ [MPa/K]	k [W/(m K)]
1	70,000	0.3	500	700	50	1.61	130
2	190,000	0.3	255	2,000	5	3.23	15

6 Verification of the sensitivities

6.1 Analysis model and conditions

Fig. 1 shows the two-dimensional shear problem used to verify the sensitivities. A 1000 mm × 2000 mm plane-strain domain was divided into four-node quadrilateral elements, where the bottom edge is fully fixed, a horizontal prescribed displacement $\bar{u}_x$ is applied to the top edge, and the vertical displacement of the top edge is constrained to 0. The prescribed displacement follows the one-cycle history shown in Fig. 1(c), which comprises $N = 80$ steps. The prescribed displacement has a maximum value of $\bar{u}_{\max} = 125\,\mathrm{mm}$. Two cases were examined: elastoplastic and thermo-elastoplastic. In the thermo-elastoplastic case, the bottom edge was at the reference temperature $T_0 = 0$, the top edge was at the temperature T, and the left and right edges were adiabatic, where $T = 1000\,\mathrm{K}$. To keep the computational cost reasonable, the analysis mesh was set to $5 \times 10 = 50$ elements. Two design variables were assigned to each element, so there were $50 \times 2 = 100$ design variables in total, which were given for each component by uniform random numbers. Fig. 2 shows the distribution of design variables for each design field. The same design was used in both cases.

Table 1 lists the material constants. The thermal stress coefficient Ξ corresponds to Eq. (12), and the values in the table are given directly. The void phase was set to a value of 1.0×10^{-3} for Young's modulus, the initial yield stress, the parameters of the Voce hardening law, and the thermal conductivity and to a value of 0 for the thermal stress coefficient. Based on preliminary studies, the DMO penalization exponent was set to $p = 3$ for Young's modulus E, the thermal stress coefficient Ξ, and the thermal conductivity k, which govern the stiffness, and to $p = 1$ for σ_{y0}, σ_∞, and b, which govern yielding and hardening. Based on preliminary studies, the projection parameters were set to $\beta = 5$ and $\eta = 0.5$. The same penalization exponents and projection parameters are used in subsequent numerical examples.

The gradients $\boldsymbol{g}_{\mathrm{full}}$ of full AD, which records every operation of all load steps on a single tape, and of central differences (Appendix A) were adopted as benchmarks. If $\boldsymbol{g}$ is the gradient obtained by each method, then the following relative L^2 error can be used to assess the validity:

$$\varepsilon_{\mathrm{rel}} = \frac{\|\boldsymbol{g} - \boldsymbol{g}_{\mathrm{full}}\|_2}{\|\boldsymbol{g}_{\mathrm{full}}\|_2}. \tag{48}$$

Computations were carried out on an Apple M3 Ultra with 256 GB of memory using the Apple clang 21.0.0 compiler. CoDiPack v3.1.2[26] was used as the AD library. Comparisons of the computational cost were run on a single thread with each method executed in turn in a separate process.

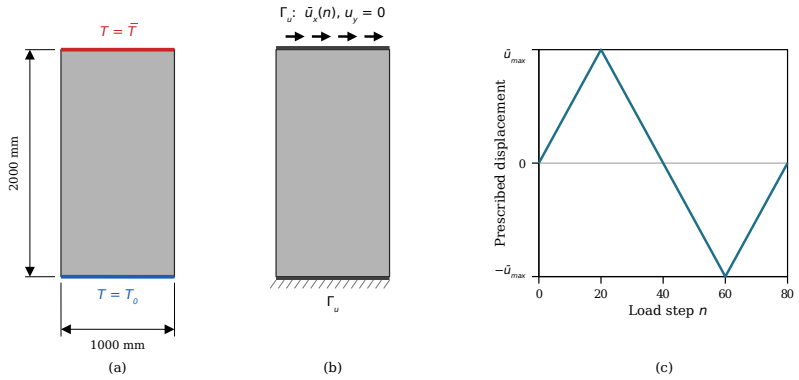


Fig. 1: Analysis model and boundary conditions of the two-dimensional shear problem: (a) thermal boundary conditions, (b) structural boundary conditions, and (c) history of the prescribed displacement. The thermal boundary conditions are imposed only in the thermo-elastoplastic case: $T = \bar{T}$ on the top edge, $T = T_0$ on the bottom edge, and adiabatic left and right edges. As structural boundary conditions, the bottom edge Γ_u is fully fixed, and the horizontal prescribed displacement $\bar{u}_x$ is applied to the top edge Γ_u.

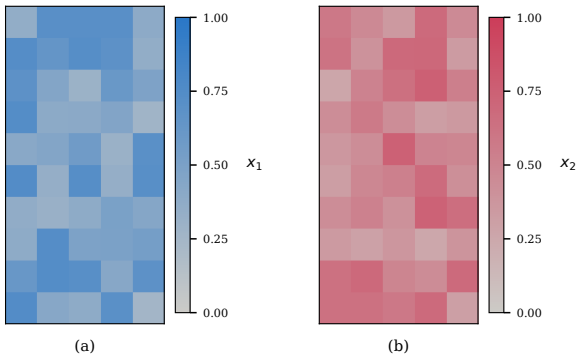


Fig. 2: Distribution of design variables used for verification of the sensitivities: (a) x_1 of design field 1 and (b) x_2 of design field 2.

6.2 Verification results

Table 2 presents the verification results. The proposed method using fixed-point AD agreed with full AD to within the roundoff error for both cases and agreed with the analytical adjoint variable method (Appendix B) for the elastoplastic case. The thermo-elastoplastic case was outside the scope of the comparison because the analytical adjoint variable method could not be used to consider the thermal stress. Fig. 3 shows the sensitivities of each design field for the elastoplastic case, and Fig. 4

Table 2: Relative L^2 error of the gradients with respect to full AD

Case	Proposed method	Analytical adjoint variable method
Elastoplastic	2.4×10^{-15}	2.9×10^{-14}
Thermo-elastoplastic	1.2×10^{-14}	-

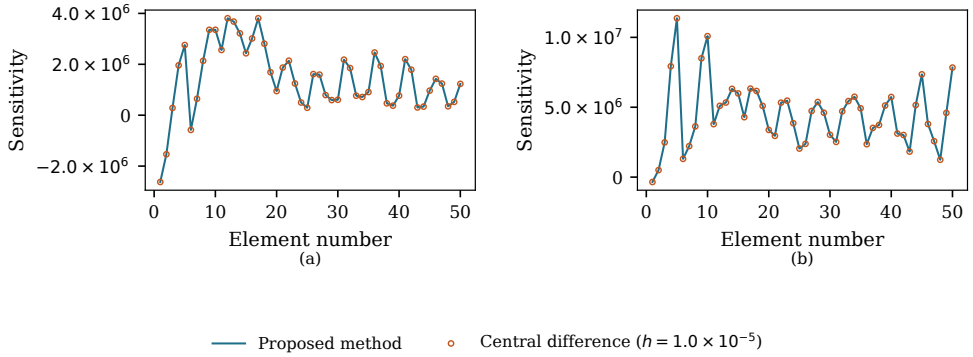


Fig. 3: Sensitivity distributions of the two-dimensional elastoplastic case: (a) design variable x_1 and (b) design variable x_2.

shows those for the thermo-elastoplastic case by superimposing the results of the proposed method and central differences. In both cases, the distributions of the proposed method and central differences coincided, which indicates that the proposed method gives the correct sensitivity distribution. Fig. 5 shows the equivalent plastic strain at the final load step of the design. The contour shows the average over the integration points of each element. In both cases, the plastic zone spread over almost the entire design domain, so the sensitivities were compared in a state that included path dependence. Thus, the topology optimization results obtained with the proposed method were verified.

7 Computational cost

The computational cost of the proposed method was evaluated on an elastoplastic problem whose sensitivities were formulated by the analytical adjoint variable method (Appendix B). Fig. 6 shows the analysis model and boundary conditions of a three-dimensional shear problem, where the 1000 mm × 2000 mm × 1000 mm domain was obtained by extruding the two-dimensional domain in Section 6.1 in the thickness direction. The model was divided into eight-node hexahedral elements. The bottom face was fully fixed. A horizontal prescribed displacement with the same history as shown in Fig. 1(c) was applied to the top face. The other two components on the top face were constrained to 0. The design was uniform with all design variables set to $\boldsymbol{x}_i = \mathbf{0.5}$. The material properties and computing environment were the same as those described in Section 6.1. The growth rate of the computational cost with respect

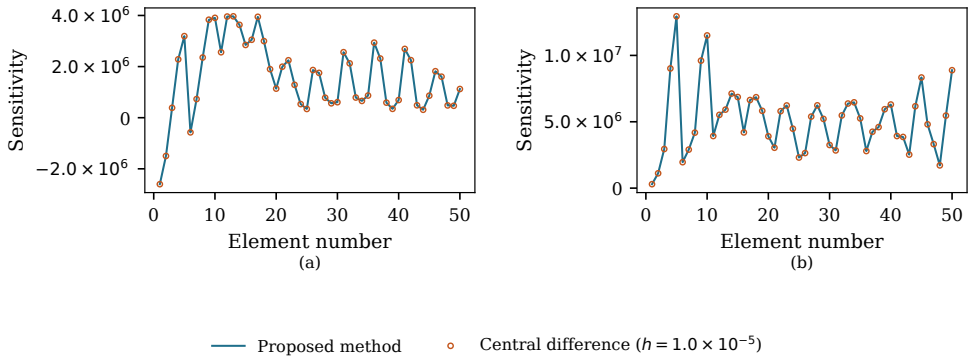


Fig. 4: Sensitivity distributions of the two-dimensional thermo-elastoplastic case: (a) design variable x_1 and (b) design variable x_2.

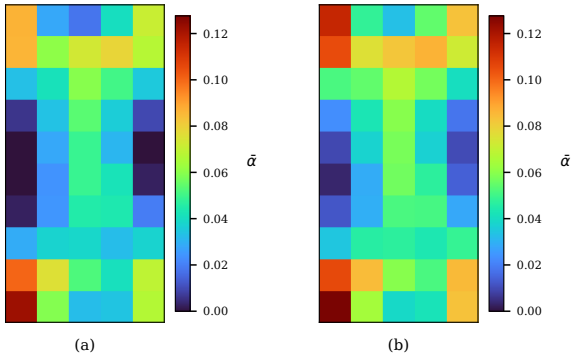


Fig. 5: Equivalent plastic strain $\bar{\alpha}$ at the final load step: (a) elastoplastic case and (b) thermo-elastoplastic case.

to the number of elements was examined by varying the total number of elements over $N_e \in \{16, 128, 1{,}024, 8{,}192, 31{,}250\}$ to obtain the computation time and peak memory. The growth rate of the computational cost with respect to the number of load steps was examined by fixing the total number of elements and varying the number of load steps.

7.1 Comparison with other sensitivity analysis methods

Table 3 compares the proposed method with other sensitivity analysis methods. For the analytical adjoint variable method, the derivation of the adjoint equations and the explicit partial derivatives are all given in closed form, and the sensitivity analysis (Appendix B) does not consider heat conduction or the effect of thermal stress (i.e., $T = 0$). Hybrid adjoint variable methods[19, 21, 22], which derive the adjoint equations

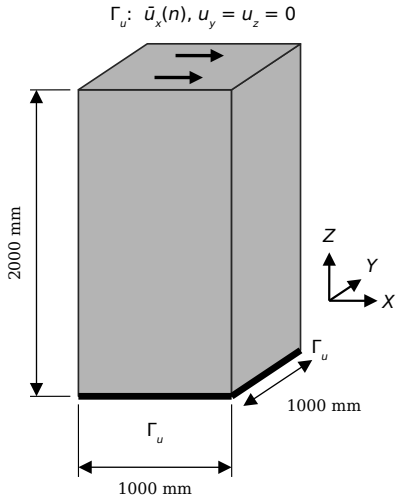


Fig. 6: Analysis model and boundary conditions of the three-dimensional shear problem. The bottom face Γ_u is fully fixed, and the horizontal prescribed displacement $\bar{u}_x$ is applied to the top face Γ_u. The history of the prescribed displacement is the same as shown in Fig. 1(c).

analytically and build only the partial derivatives by AD, were not included in the comparison because their backward recursion of the adjoint depends on the physical model.

Full AD corresponds to reverse-mode AD where the operations of all load steps, which includes the global Newton iterations and the local iterations of the return mapping, are recorded on a single tape, and the gradient is obtained in a single reverse sweep. Checkpointing[15] stores the state every two steps in the forward analysis; in the reverse sweep, the required steps are recomputed from the stored states while all operations of one step were recorded and replayed backward. For an 80-step problem, this means that 40 states are stored. The tape held at one time is reduced to that of a single step, but that tape contains all Newton iterations, and the computational cost increases owing to the recomputation. The proposed method obtains the sensitivities at a computational cost comparable to that of the analytical adjoint variable method while keeping the generality of AD, which requires no formulation.

7.2 Growth rate according to the number of elements

Table 4 summarizes the computation time and peak memory required for sensitivity analysis according to the total number of elements. Fig. 7 graphs the results on log–log axes. Full AD and checkpointing are shown up to the size that could be executed with the present implementation. Every method is almost a straight line on the log–log plot, and the computation time and peak memory increase approximately linearly with the number of elements. In other words, the computational costs of the sensitivity

Table 3: Comparison of sensitivity analysis methods

Method	Derivation of the sensitivities	Tape held at one time
Analytical adjoint variable method	Analytical derivation of Appendix B	-
Full AD	All operations	All N steps
Checkpointing	All operations including recomputation	One step
Proposed method	Map at the converged point only	One analysis step

Table 4: Computation time and peak memory required by different sensitivity analysis methods according to the number of elements: (a) analytical adjoint variable method, (b) proposed method, (c) checkpointing, and (d) full AD

N_e	Computation time [s]				Peak memory [MiB]			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
16	0.24	0.31	1.20	1.06	22	43	108	4,184
128	1.86	2.23	9.77	8.80	65	99	847	34,341
1,024	17.3	19.9	88.8	-	419	636	7,236	-
8,192	193	211	867	-	2,995	4,648	53,582	-
31,250	1,228	1,299	-	-	11,792	17,946	-	-

analysis methods differed by constant factors. For the proposed method, the computation time and peak memory showed the same growth rate as the analytical adjoint variable method. At the maximum number of 31,250 elements, the computation time was 1.06 times and the peak memory was 1.5 times those of the analytical adjoint variable method. Compared with checkpointing, the proposed method requires about 1/4 of the computation time; for 1024 elements and more, it requires about 1/11 of the peak memory. Full AD required an enormous amount of memory and could be executed only up to a very small number of elements. Thus, the proposed method greatly reduced the computational cost compared with full AD and checkpointing while obtaining sensitivities at a computational cost comparable to that the analytical adjoint variable method. Consequently, the proposed method can replace the analytical adjoint variable method in terms of computational cost without requiring the derivation of adjoint equations for each physical model.

7.3 Growth rate according to the number of load steps

The proposed method records and discards the tape for each analysis step, so the amount of tape held at one time does not depend on the number of load steps. To confirm this, the total number of elements was fixed at 128, and the one-cycle loading

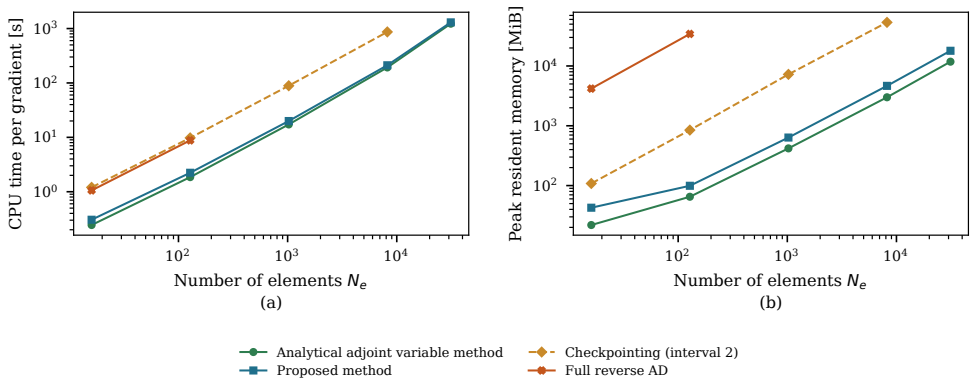


Fig. 7: Log–log plots of the (a) computation time and (b) peak memory required by different sensitivity analysis methods according to the number of elements.

path shown in Fig. 1(c) was divided into $N \in \{20, 40, 80, 160, 320, 640\}$ load steps to measure the computation time, peak memory, and largest amount of tape held at one time. Because the loading path is the same and only its division changes, increasing N decreases the load increment per step, which affects the number of Newton iterations. The computation times and tape sizes below therefore include this effect as well.

Table 5 summarizes the computation time, peak memory, and tape size for different sensitivity analysis methods. For each method, the computation time was almost proportional to N. The computation time of the proposed method was about 1.2 times that of the analytical adjoint variable method, and this ratio did not depend on N. In contrast, checkpointing required more than four times the computation time of the proposed method, because it re-solves each step from the stored states. Because topology optimization requires a sensitivity analysis to be performed at every iteration, this difference in computation time translates directly into a difference in the total optimization time, and a constant-factor difference has a large impact.

Fig. 8 shows (a) the tape size and (b) the peak memory on log–log axes. The peak memory can be approximated by the sum of the tape held at one time for backward propagation and the converged states stored in the forward analysis (i.e., the state variables and internal variables). The proposed method had a constant tape size of about 30 MiB regardless of the number of load steps N. This is because the tape was recorded for each analysis step and discarded after it had been traced backward. In contrast, the tape size for full AD grew in proportion to N because the operations of all load steps were recorded on a single tape. It reached about 33 GiB at $N = 80$, and full AD could not be executed at $N = 160$ on a computer with 256 GB of memory. For checkpointing, the tape size was for a single step and therefore did not depend on N. However, it contained all of the Newton iterations of that step, so the tape size was about 20 times that of the proposed method. This difference in tape size is reflected in the peak memory. For the proposed method and analytical adjoint variable method, the peak memory increased with N because the state variables and internal variables of the converged state of every load step were stored at once for all analysis steps.

Table 5: Computation time, largest tape held at one time, and peak memory of different sensitivity analysis methods according to the number of load steps N (three-dimensional, 128 elements): (a) analytical adjoint variable method, (b) proposed method, (c) checkpointing, and (d) full AD

N	Computation time [s]				Tape size [MiB]			Peak memory [MiB]			
	(a)	(b)	(c)	(d)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
20	0.51	0.60	2.78	2.46	30	721	9,545	44	77	921	9,670
40	0.96	1.14	5.16	4.60	30	680	17,931	51	82	876	18,146
80	1.86	2.24	9.82	8.80	30	627	33,979	66	99	838	34,340
160	3.58	4.32	19.0	-	30	627	-	94	126	874	-
320	7.11	8.57	37.7	-	30	584	-	152	186	885	-
640	14.3	17.2	74.5	-	29	579	-	266	298	912	-

Thus, the proposed method and analytical adjoint variable method had comparable computational costs. For checkpointing, the peak memory was almost constant at about 1000 MiB regardless of N. The stored converged states amounted to $N/2$ steps, and the peak memory increased with N, but this increase was not conspicuous because the tape of a single step dominated the peak memory. For full AD, the peak memory was almost the tape size itself, so an enormous amount of memory was required.

Thus, the computational cost of the proposed method is comparable to that of the analytical adjoint variable method because the tape is limited to a single analysis step. Checkpointing similarly limits the tape to a single step, but that tape contains all of the Newton iterations, which increases the tape size. In addition, the computation time is more than four times longer because of the recomputation. When the number of load steps N is large and the storage of converged states dominates the peak memory, both the proposed method and checkpointing need to thin out the storage of the converged states and recompute the steps without stored states during backward propagation. However, the computation time increases with recomputation, so there is a tradeoff between memory and computation time. This is the same for the analytical adjoint variable method. Within the range of this study, the proposed method had a peak memory below 300 MiB even at $N = 640$, so it could be used as it is. The computational cost of the proposed method showed the same growth rate as the analytical adjoint variable method with respect to the number of load steps N, and the difference between the two was constant regardless of N. In path-dependent problems, the load steps must sometimes be refined to follow the stress and strain histories accurately. Even in such cases, the computational cost of the proposed method stayed comparable to that of the analytical adjoint variable method, and the tape did not grow with the number of steps as in the case of full AD.

8 Numerical example of a two-dimensional problem

8.1 Analysis model and conditions

To demonstrate that the proposed method can solve topology optimization problems of a practical size, steady-state heat conduction was added to the two-dimensional

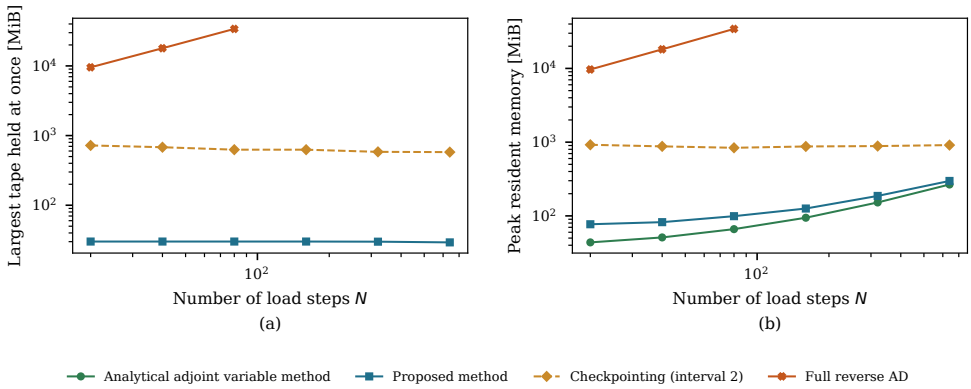


Fig. 8: Log–log plots of (a) the largest tape held at one time and (b) peak memory against the number of load steps N for different sensitivity analysis methods applied to the 128-element three-dimensional model.

multi-material shear problem shown in Fig. 1 to obtain a thermo-elastoplastic optimization problem. The domain was divided into $40 \times 80 = 3{,}200$ four-node quadrilateral elements, and the number of design variables was set to 6400. The boundary conditions and history of the prescribed displacement were the same as described in Section 6.1. As shown in Fig. 1(b), the bottom edge was fully fixed, and a prescribed displacement with a maximum of $\bar{u}_{\max} = 125\,\mathrm{mm}$ was applied to the top edge over one cycle of $N = 80$ steps following the profile shown in Fig. 1(c). The thermal boundary conditions were also the same as described in Section 6.1: the bottom edge was at the reference temperature $T_0 = 0$, the top edge was at the temperature T, and the left and right edges were adiabatic. The top edge with the prescribed displacement was hot while the fixed bottom edge was cold, so heat flowed from the top edge to the bottom edge. Two temperature cases were considered: $T = 0$ (no heat) and $1000\,\mathrm{K}$. The $T = 0$ case retained the heat conduction analysis while setting the thermal stress to 0, which reduced the problem to elastoplastic optimization without heat.

The materials corresponded to the two materials listed in Table 1 and the void phase. The DMO penalization exponent was 3 for E, Ξ, and k and 1 for σ_{y0}, σ_∞, and b. The initial design was $x_{ei} = 0.5$ for all elements and both design fields, and the lower bound of the design variables was $x_{\min} = 0.001$. The upper bounds of the volume were $\bar{v}_1 = \bar{v}_2 = 0.2$. The projection parameters were $\beta = 5$ and $\eta = 0.5$. The method of moving asymptotes (MMA)[38] was used as the optimization algorithm, which updates the design variables based on the sensitivity distribution obtained from the sensitivity analysis.

8.2 Results

Fig. 9 shows the topology optimization results, the temperature field in the resulting structure, and the equivalent plastic strain after one cycle of loading. Gray denotes the void phase, blue denotes material 1, and red denotes material 2. For $T = 0$, material 1

crossed at the center to resist the applied shear deformation, and material 2 occupied the tips of the arms extending to the four corners and center of the crossing. The result was symmetric top to bottom. For $T = 1000\,\mathrm{K}$, the overall crossing structure resisting the shear deformation was the same, but more of material 2 was placed on the cold lower side, and more of material 1 was on the hot upper side.

The structure changed with the thermal load owing to the difference in the thermal stress coefficient Ξ between the materials. Table 1 indicates that the thermal stress coefficient Ξ of material 2 was about twice that of material 1, so a large thermal stress was more likely to arise. In a problem where expansion is constrained at the top and bottom edges, a material placed at a hotter location receives a compressive thermal stress proportional to Ξ and reaches the plastic range more easily. Therefore, placing material 2 with its larger thermal stress coefficient Ξ near the cold lower part suppresses plastic behavior and allows the structure to resist the prescribed displacement more strongly.

Fig. 10 shows the iteration histories of the objective function and volume constraints. In both cases, the increase in the objective function became gradual after about 30 iterations, and thereafter the volume constraints remained satisfied. In the initial stage of optimization, the value of the objective function decreased considerably. This is because the optimization started with the design variables set at an initial value of 0.5, which violated the volume constraint of 0.2. Thus, the optimization algorithm first updated the design to satisfy the volume constraints. Eventually, however, the value of the objective function became higher than that of the initial structure while the volume constraints were maintained, and the optimization converged, so the topology optimization was carried out properly.

For the $T = 0$ case, optimization was performed with different sensitivity analysis methods, and the computational costs were compared. The computations were run on a single thread, with each method executed in turn in a separate process. Table 6 compares the computation time and peak memory of 100 optimization iterations and the relative L^2 error of the obtained design variables with respect to the analytical adjoint variable method. The proposed method had a computation time and peak memory that were 1.11 times and 1.10 times, respectively, those of the analytical adjoint variable method. In contrast, checkpointing had a computation time and peak memory that were 3.1 times and 2.0 times, respectively, those of the analytical adjoint variable method. The designs obtained by the three methods agreed to within the roundoff error. Fig. 11 shows the iteration histories of the objective function and volume constraints obtained with the proposed method and analytical adjoint variable method. The two curves overlapped at every iteration, which indicates that they followed the same optimization path. Hence, the proposed method can replace the analytical adjoint variable method as it gives the same optimization result at a comparable computational cost while not requiring complicated formulations for multiphysics and path-dependent problems.

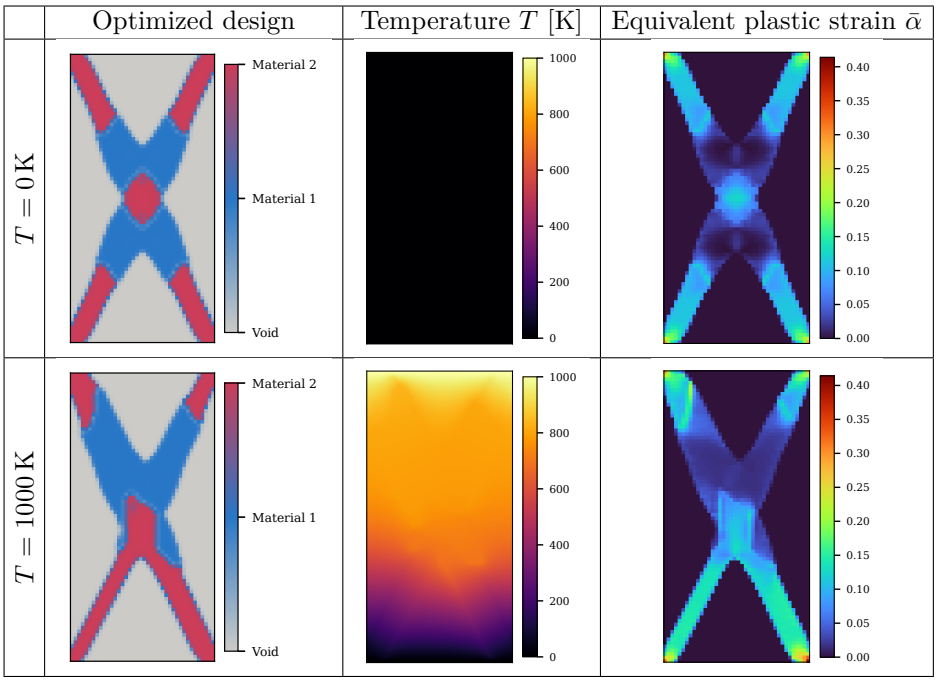


Fig. 9: Topology optimization results for the two-dimensional thermo-elastoplastic problem. The rows correspond to the temperature T. The columns correspond to the material layout of the optimized design, the temperature field, and the equivalent plastic strain after one cycle. The material layout is drawn element by element as the material number (0 for the void phase, 1 for material 1, and 2 for material 2) interpolated with the weights of Eq. (24) ($p = 1$).

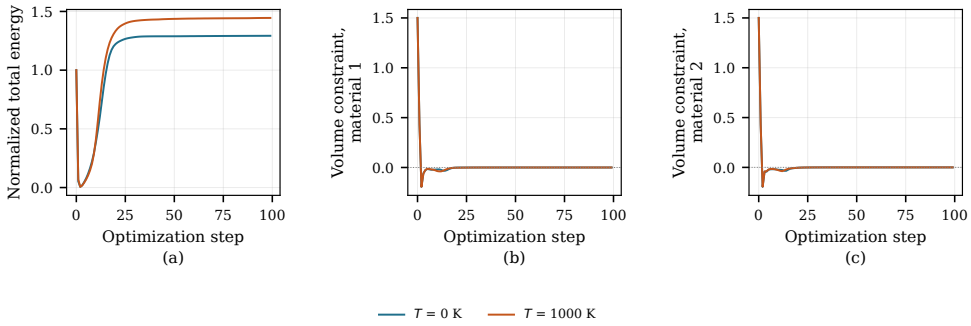


Fig. 10: Iteration histories for optimization of the two-dimensional thermo-elastoplastic problem: (a) total energy normalized by the value of the initial design, (b) volume constraint of material 1, and (c) volume constraint of material 2.

Table 6: Computational cost of the two-dimensional elastoplastic optimization (3,200 elements, $N = 80$, 100 iterations) for each sensitivity analysis method.

Method	Computation time [min]	Peak memory [MiB]	Relative L^2 difference of the design variables
Analytical adjoint variable method	18.2	947	-
Proposed method	20.1	1,037	4.2×10^{-13}
Checkpointing	57.2	1,938	5.8×10^{-13}

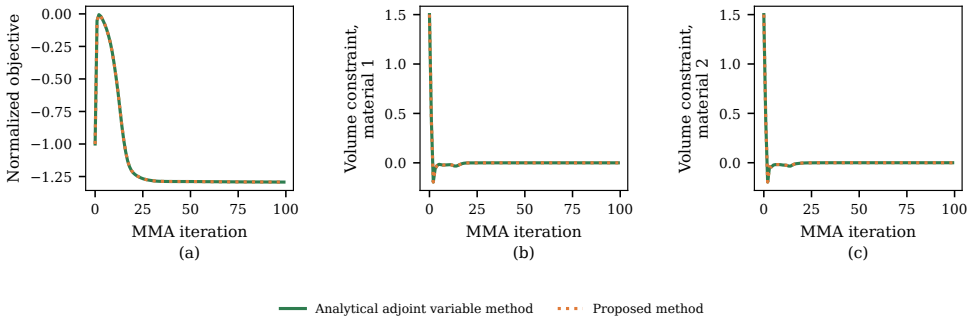


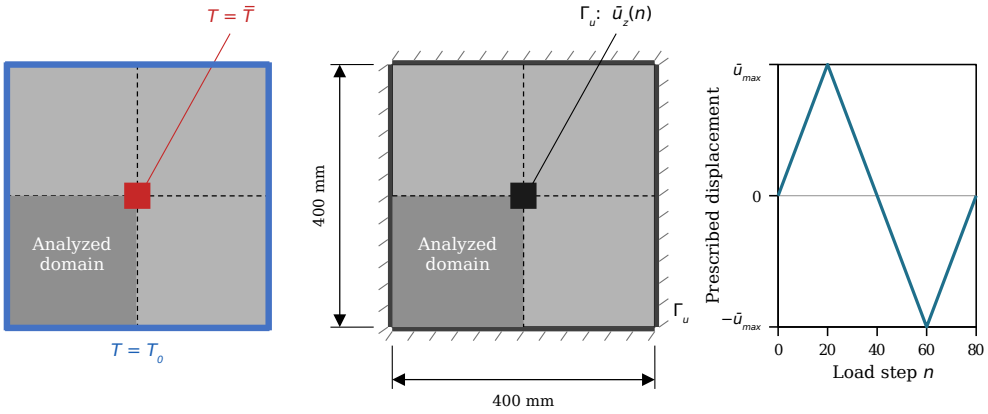
Fig. 11: Iteration histories with different sensitivity analysis methods: (a) normalized objective function, (b) volume constraint of material 1, and (c) volume constraint of material 2.

9 Numerical example of a three-dimensional problem

9.1 Analysis model and conditions

The proposed method was also applied to a three-dimensional problem with a correspondingly larger computational cost to demonstrate its effectiveness. Fig. 12 shows the analysis model and boundary conditions. The design domain was set to 400 mm $\times$ 400 mm $\times$ 100 mm. Owing to the symmetry of the analysis model, a quarter of the domain with the symmetry planes $X = 200\,\mathrm{mm}$ and $Y = 200\,\mathrm{mm}$ was analyzed. The quarter domain was divided into $40 \times 40 \times 20$ eight-node hexahedral elements, which corresponded to 32,000 elements and 64,000 design variables.

For the structural boundary conditions, the side faces of the plate were fully fixed, and only the normal displacement was constrained on the symmetry planes. As shown in Fig. 12(c), a prescribed displacement was applied in the $-Z$ direction with a maximum of $\bar{u}_{\max} = 10\,\mathrm{mm}$ to the 40 mm $\times$ 40 mm area at the center of the top face over one cycle divided into $N = 80$ steps. For the thermal boundary conditions, three temperature cases were applied at the location of the prescribed displacement: $T = 0\,\mathrm{K}$, $T = 500\,\mathrm{K}$, and $T = 1000\,\mathrm{K}$. The fully fixed outer boundary was at the reference temperature $T_0 = 0$, and the other faces were adiabatic. For the $T = 0$ case, no temperature distribution arose in the body, and no thermal stress occurred, which reduced the thermo-elastoplastic problem to the elastoplastic problem. The materials



(a) (b) (c)

Fig. 12: Plan view of the analysis model and boundary conditions of a three-dimensional plate: (a) thermal boundary conditions, (b) structural boundary conditions, and (c) history of the prescribed displacement. The plate thickness is 100 mm. Thermally, the side faces are at $T = T_0$, the 40×40 mm area at the center of the top face is at $T = \bar{T}$, and the other faces are adiabatic. Structurally, the side faces Γ_u are fully fixed, and the prescribed displacement $\bar{u}_z$ is applied to the same area at the center of the top face. The dark part is the analyzed quarter domain, and the dashed lines are the symmetry planes.

corresponded to the two materials listed in Table 1 and the void phase. The DMO penalization exponent was 3 for E, Ξ, and k and 1 for σ_{y0}, σ_∞, and b. For the initial design, all elements and both design fields were set to $x_{ei} = 0.5$. The lower bound of the design variables was $x_{\min} = 10^{-3}$, and the upper bound of the volume for each material was $\bar{v}_1 = \bar{v}_2 = 0.1$. Based on preliminary studies, the projection parameters were $\beta = 5$ and $\eta = 0.5$. MMA[38] was used as the optimization algorithm with a maximum of 100 iterations. The design variables were updated based on the sensitivities obtained by the proposed method. The three cases were run simultaneously as separate processes with eight threads each. Therefore, the computation times described below are for wall-clock times with eight threads and are not equivalent to the computation times described in Section 7.

9.2 Results

Fig. 13 shows the material layouts on the top face, at mid-thickness, and on the bottom face. Fig. 14 shows the vertical section through the center. Fig. 15 shows the three-dimensional view with the void phase removed. All show the results of the quarter domain mirrored back to the full domain. In all three cases, an arch-like structure rising in the thickness direction from the fully fixed side faced the prescribed displacement at the center of the top face. The upper chord of the arch was occupied by material 2, and the diagonals around the fixed ends were occupied by material 1.

When $T \neq 0$ and the effect of thermal stress was added, the structural form changed. While the $T = 0$ case had a horizontal member of material 1 directly below the loading point at mid-thickness, it became shorter for the $T = 500\,\mathrm{K}$ case, and it disappeared and was replaced by material 2 for the $T = 1000\,\mathrm{K}$ case. Fig. 16 shows the temperature field and equivalent plastic strain at mid-thickness. Because the fully fixed side faces were cooled, the temperature decreased from the prescribed displacement toward the outer boundary, and the average temperature of the structure stayed at about 25% of T. The total energy increased with T, and the fraction of plastic integration points was almost the same in the three cases. Fig. 17 shows the iteration histories. The volume constraints were satisfied, and the objective function was almost constant after 60 iterations. This indicates that the optimization converged properly. Similar to Section 8.2, the objective function deteriorated considerably in the initial steps of the optimization because the initial structure violated the volume constraints, so the design variables were first updated to ensure that the constraints were satisfied.

These results show that the proposed method can be applied to the topology optimization of multiphysics and path-dependent three-dimensional problems of a practical size. The proposed method resolves the high computational cost of conventional AD without requiring the complicated formulations of the analytical adjoint variable method.

10 Conclusions

The proposed method used fixed-point AD for the topology optimization of path-dependent problems at a reasonable computational cost. Rather than evaluating every operation of the forward analysis by AD at once, the proposed method fixes the consistent tangent at the converged point of each analysis step, records the fixed-point map consisting of the evaluation of the residual vector only once by reverse-mode AD, updates the solution with the fixed consistent tangent, updates the internal variables, and computes the sensitivity of the internal variables through a backward recursion between analysis steps. This avoids the complicated formulations of the analytical adjoint variable method while computing the sensitivities required for topology optimization at a comparable computational cost. The main findings and remaining issues are discussed below.

- Problems in which several governing equations are solved in sequence can be generalized as a sequence of analysis steps, and a general formulation independent of the physical model is presented where recording the fixed-point map once at the converged point of each analysis step yields the derivative of the discrete solution by the implicit function theorem, which includes the sensitivity that goes back across steps through the internal variables. This formulation extends fixed-point AD to path-dependent problems with internal variables and is the main contribution of the proposed method. The complicated adjoint equations need not be rederived when the coupled physical model, constitutive law, objective function, or interpolation of the material properties by the design variables is changed.
- The proposed method was verified for elastoplastic and thermo-elastoplastic problems, which have a yield condition and branching. The gradient of the proposed

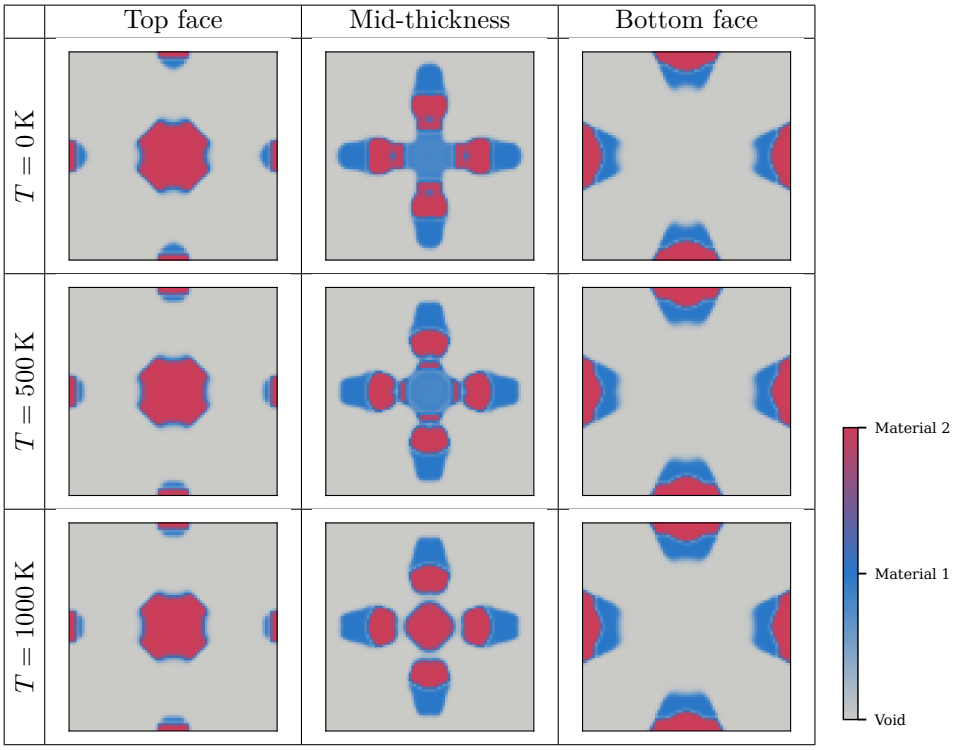


Fig. 13: Material layouts of three-dimensional designs after thermo-elastoplastic optimization. The rows correspond to the temperature T. The columns correspond to the top face, mid-thickness, and bottom face. The interpolation of the material number is the same as in Fig. 9.

method agreed with the analytical adjoint variable method, reverse-mode AD, and central differences, which shows that it gives the correct sensitivities.

- The computation time and peak memory of the proposed method showed the same growth rate as the analytical adjoint variable method with respect to the number of elements. They were much smaller than those of checkpointing, which allowed the proposed method to solve problems of a size that cannot be handled by reverse-mode AD. These results quantitatively demonstrate that the proposed method can be used to conduct a sensitivity analysis requiring no formulation at a computational cost comparable to that of the analytical adjoint variable method.
- The amount of tape held at one time by the proposed method is constant regardless of the number of load steps. The peak memory increases in proportion to the number of analysis steps because the converged states of all steps are stored, but it is kept small compared with full AD. The proposed method was confirmed to handle three-dimensional optimization problems without difficulty. This property is shared with the analytical adjoint variable method, and the computational cost can be reduced further if necessary by thinning the converged states as in checkpointing.

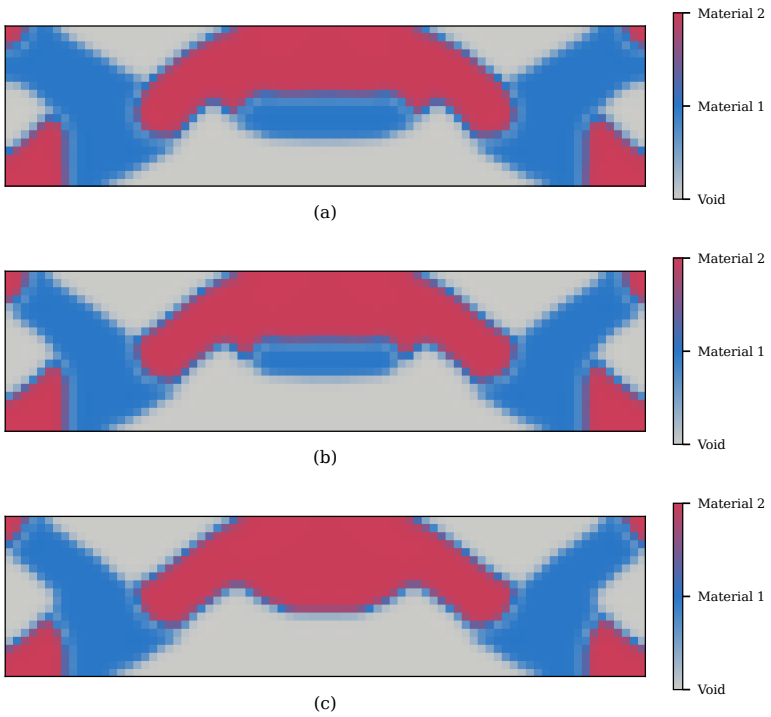


Fig. 14: Vertical section ($Y = 200\,\mathrm{mm}$) of the three-dimensional designs after thermo-elastoplastic optimization: (a) $T = 0\,\mathrm{K}$, (b) $500\,\mathrm{K}$, and (c) $1000\,\mathrm{K}$.

- When applied to two-dimensional optimization, the proposed method obtained the same design as the analytical adjoint variable method at a comparable computational cost. The same sensitivity analysis was applied without modification to the three-dimensional thermo-elastoplastic problem to which the analytical adjoint variable method is inapplicable. Thus, the proposed method can replace the analytical adjoint variable method in terms of both computational cost and generality.
- The computational cost for the topology optimization of path-dependent problems primarily comes from the forward analysis. Large-scale parallel computation may be a solution, but its combination with the proposed method has not yet been examined and will be the subject of future work.

Declarations

Funding

The authors did not receive support from any organization for the submitted work.

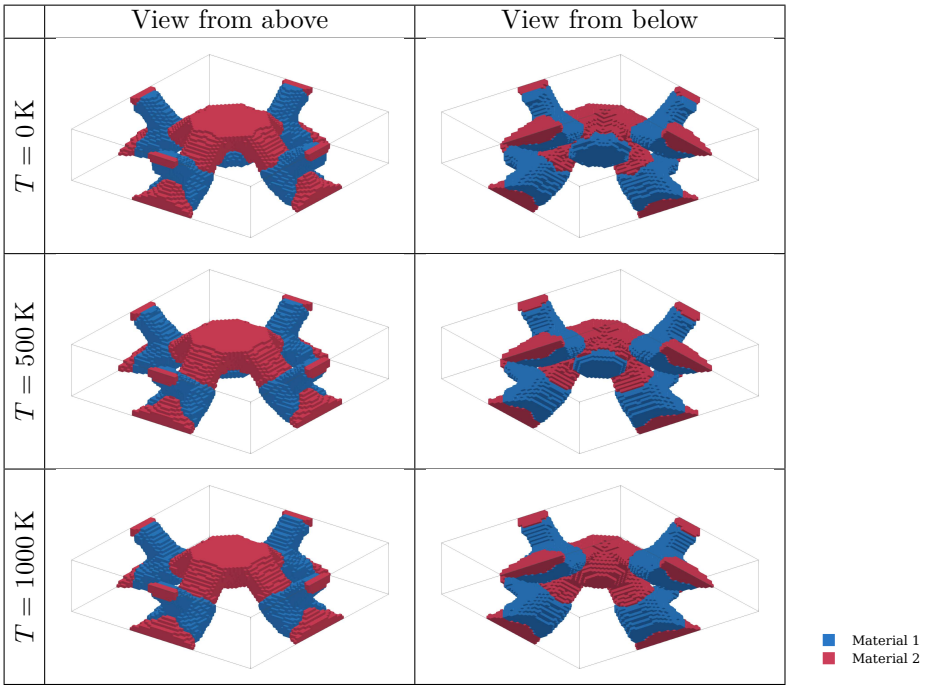


Fig. 15: Three-dimensional views of the three-dimensional designs after thermo-elastoplastic optimization. The rows correspond to the temperature T, and the columns correspond to the viewpoints. Each element is binarized to the phase with the largest weight w_0, w_1, or w_2 of Eq. (24) ($p = 1$). The elements of the void phase are hidden, and each material is drawn in a single color. The frame corresponds to the design domain (400 mm $\times$ 400 mm $\times$ 100 mm).

Author contributions

Shun Ogawa contributed to the study conception and design, developed the method, implemented the code, and performed the numerical computations. The first draft of the manuscript was written by Shun Ogawa, and Katsuyuki Suzuki reviewed and commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Conflict of interest

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

Ethics approval and Consent to participate

Not applicable.

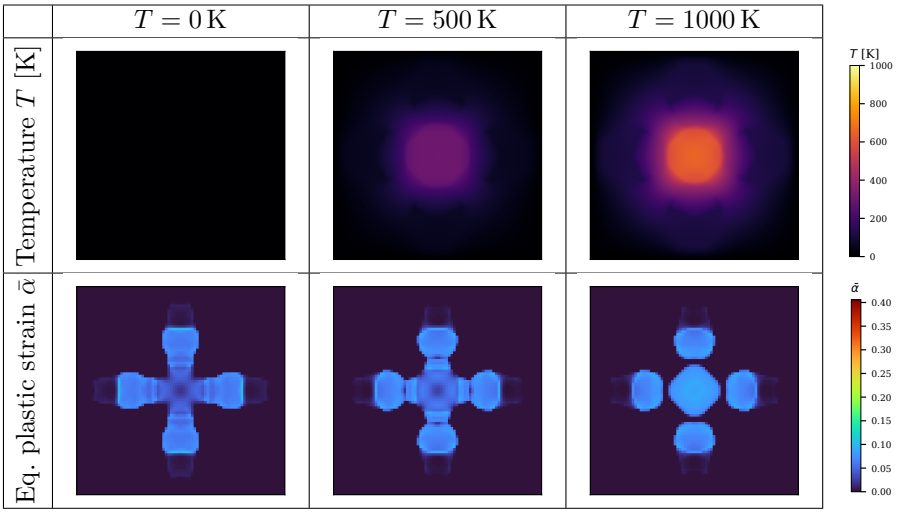


Fig. 16: Temperature field and equivalent plastic strain at mid-thickness of the three-dimensional designs after thermo-elastoplastic optimization. The upper row corresponds to the temperature field. The lower row corresponds to the equivalent plastic strain after one cycle. The columns correspond to the temperature T. The same color scale is used forr the three cases, and the color bar is shown to the right of each row.

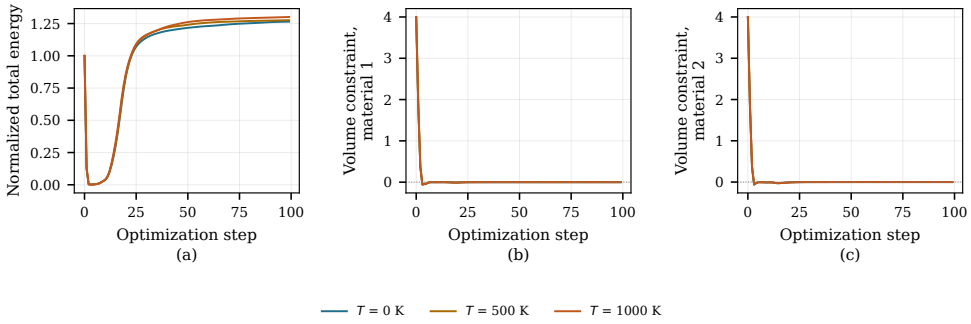


Fig. 17: Iteration histories of the three-dimensional thermo-elastoplastic optimization: (a) total energy normalized by the value of the initial design, (b) volume constraint of material 1, and (c) volume constraint of material 2.

Data availability

Data will be made available on request.

Appendix A Verification of the gradients by central differences

The central differences (Section 6) were computed for each component of the design variables as follows:

$$\frac{\partial f}{\partial x_{ei}} \approx \frac{f(\boldsymbol{x} + h\,\boldsymbol{e}_{ei}) - f(\boldsymbol{x} - h\,\boldsymbol{e}_{ei})}{2h}, \tag{A1}$$

where $\boldsymbol{e}_{ei}$ is the unit vector whose only nonzero component is x_{ei} and the filter, projection, and forward analysis are all recomputed for each perturbation. Because no AD is used, this check does not depend on the evaluation of the forward analysis shared by full AD and fixed-point AD. The step h was chosen by comparing the relative L^2 error with respect to full AD over the range from 1.0×10^{-4} to 1.0×10^{-7} and was unified to $h = 1.0 \times 10^{-5}$ for which the error was below 1.0×10^{-5} in every problem. Decreasing the step reduced the truncation error, and decreasing it further increased the error again by roundoff, so the minimum lay within this range.

Appendix B Sensitivity analysis by the analytical adjoint variable method

The analytical adjoint variable method was formulated by deriving the adjoint equation by hand for the discretized path-dependent problem in Eq. (8)[1, 7], which was adapted to DMO multi-material interpolation and Voce hardening. Every equation below has to be derived again whenever the constitutive law, material interpolation, or objective function was changed. In contrast, fixed-point AD uses none of these equations.

B.1 Adjoint equations and backward recursion

The contribution of step n to the objective function is written as

$$\varphi_n(\boldsymbol{u}_n;\, \boldsymbol{u}_{n-1}, \boldsymbol{h}_{n-1}, \tilde{\tilde{\boldsymbol{x}}}) = \int_\Omega \boldsymbol{\sigma}_n : (\boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_{n-1})\,\mathrm{d}\Omega. \tag{B2}$$

Because the stress $\boldsymbol{\sigma}_n$ is determined by the return mapping from $\boldsymbol{u}_n$, $\boldsymbol{h}_{n-1}$ and the physical densities $\tilde{\tilde{\boldsymbol{x}}}$, φ_n depends on these and on the displacement $\boldsymbol{u}_{n-1}$ of the previous step. With the equilibrium and the update of the internal variables of Eq. (8) as constraints, the Lagrangian is defined as

$$\mathcal{L} = \sum_{n=1}^{N} \Big[\varphi_n(\boldsymbol{u}_n;\, \boldsymbol{u}_{n-1}, \boldsymbol{h}_{n-1}, \tilde{\tilde{\boldsymbol{x}}}) - \boldsymbol{\lambda}_n^{\mathsf{T}} \boldsymbol{R}_n(\boldsymbol{u}_n;\, \boldsymbol{h}_{n-1}, \tilde{\tilde{\boldsymbol{x}}}) - \boldsymbol{\mu}_n^{\mathsf{T}} \big(\boldsymbol{h}_n - \boldsymbol{H}_n(\boldsymbol{u}_n;\, \boldsymbol{h}_{n-1}, \tilde{\tilde{\boldsymbol{x}}})\big)\Big], \tag{B3}$$

where $\boldsymbol{\lambda}_n$ is the adjoint variable for the equilibrium (placed on the nodal degrees of freedom excluding those with prescribed displacement) and $\boldsymbol{\mu}_n$ is the adjoint variable for updating the internal variables (placed at each integration point). The stationarity

condition with respect to $\boldsymbol{h}_n$ gives

$$\boldsymbol{\mu}_n = \frac{\partial \varphi_{n+1}}{\partial \boldsymbol{h}_n} - \left(\frac{\partial \boldsymbol{R}_{n+1}}{\partial \boldsymbol{h}_n}\right)^{\mathsf{T}} \boldsymbol{\lambda}_{n+1} + \left(\frac{\partial \boldsymbol{H}_{n+1}}{\partial \boldsymbol{h}_n}\right)^{\mathsf{T}} \boldsymbol{\mu}_{n+1}, \tag{B4}$$

and the stationarity condition with respect to $\boldsymbol{u}_n$ gives

$$\begin{cases} \boldsymbol{K}_n^{\mathsf{T}} \boldsymbol{\lambda}_n = \dfrac{\partial \varphi_n}{\partial \boldsymbol{u}_n} + \dfrac{\partial \varphi_{n+1}}{\partial \boldsymbol{u}_n} + \left(\dfrac{\partial \boldsymbol{H}_n}{\partial \boldsymbol{u}_n}\right)^{\mathsf{T}} \boldsymbol{\mu}_n \\ \boldsymbol{K}_n = \dfrac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{u}_n}. \end{cases} \tag{B5}$$

Because $\boldsymbol{h}_N$ and $\boldsymbol{u}_N$ do not appear in any later step, the terminal conditions are $\boldsymbol{\mu}_N = \boldsymbol{0}$, $\boldsymbol{\lambda}_{N+1} = \boldsymbol{0}$, and $\partial \varphi_{N+1} / \partial \boldsymbol{u}_N = \boldsymbol{0}$. The adjoint variables are obtained in the order of $n = N, N-1, \ldots, 1$ by first determining $\boldsymbol{\mu}_n$ from Eq. (B4) and then solving Eq. (B5) for $\boldsymbol{\lambda}_n$. The gradient with respect to the physical densities is accumulated as

$$\frac{\mathrm{d}f}{\mathrm{d}\tilde{\tilde{\boldsymbol{x}}}} = \sum_{n=1}^{N} \left[\frac{\partial \varphi_n}{\partial \tilde{\tilde{\boldsymbol{x}}}} - \left(\frac{\partial \boldsymbol{R}_n}{\partial \tilde{\tilde{\boldsymbol{x}}}}\right)^{\mathsf{T}} \boldsymbol{\lambda}_n + \left(\frac{\partial \boldsymbol{H}_n}{\partial \tilde{\tilde{\boldsymbol{x}}}}\right)^{\mathsf{T}} \boldsymbol{\mu}_n \right]. \tag{B6}$$

Backward recursion requires the converged states $(\boldsymbol{u}_n, \boldsymbol{h}_n)$ of all steps, which are stored in the forward analysis. For the von Mises model with the associated flow rule, $\boldsymbol{K}_n$ is symmetric, and the transposed solve of Eq. (B5) is needed only once per step.

B.2 Partial derivatives at the integration points

The partial derivatives of the residual $\boldsymbol{R}_n = \int_\Omega \boldsymbol{B}^{\mathsf{T}} \boldsymbol{\sigma}_n \, \mathrm{d}\Omega$ and the update of the internal variables $\boldsymbol{H}_n$ are obtained by assembling the partial derivatives of the stress and internal variables at each integration point:

$$\begin{cases} \dfrac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{u}_n} = \displaystyle\int_\Omega \boldsymbol{B}^{\mathsf{T}} \frac{\partial \boldsymbol{\sigma}_n}{\partial \boldsymbol{\varepsilon}_n} \boldsymbol{B} \, \mathrm{d}\Omega \\ \dfrac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{h}_{n-1}} = \displaystyle\int_\Omega \boldsymbol{B}^{\mathsf{T}} \frac{\partial \boldsymbol{\sigma}_n}{\partial \boldsymbol{h}_{n-1}} \, \mathrm{d}\Omega \\ \dfrac{\partial \boldsymbol{H}_n}{\partial \boldsymbol{u}_n} = \dfrac{\partial \boldsymbol{h}_n}{\partial \boldsymbol{\varepsilon}_n} \boldsymbol{B}. \end{cases} \tag{B7}$$

At integration points in the plastic state, the consistency condition of Eq. (19) is differentiated as an implicit function to obtain the partial derivatives of the plastic

multiplier. Because $\partial q^{\mathrm{tr}}/\partial \boldsymbol{\varepsilon}^{e,\mathrm{tr}} = 2\mu\sqrt{3/2}\,\boldsymbol{\nu}$,

$$\begin{cases} \dfrac{\partial \Delta\gamma}{\partial \boldsymbol{\varepsilon}_n} = -\dfrac{\partial \Delta\gamma}{\partial \boldsymbol{\varepsilon}^p_{n-1}} = \dfrac{2\mu\sqrt{3/2}}{3\mu + \sigma'_y}\,\boldsymbol{\nu} \\ \dfrac{\partial \Delta\gamma}{\partial \bar{\alpha}_{n-1}} = -\dfrac{\sigma'_y}{3\mu + \sigma'_y}, \end{cases} \tag{B8}$$

where $\sigma'_y = \sigma'_y(\bar{\alpha}_n)$. From Eq. (16), $\sigma'_y(\bar{\alpha}) = (\sigma_\infty - \sigma_{y0})\,b\,e^{-b\bar{\alpha}}$. The partial derivative of the flow direction is

$$\frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\varepsilon}_n} = \frac{2\mu}{\|\boldsymbol{s}^{\mathrm{tr}}\|}\left(\mathbb{I}^{\mathrm{dev}} - \boldsymbol{\nu}\otimes\boldsymbol{\nu}\right). \tag{B9}$$

Substituting these into Eq. (18) obtains the partial derivatives of the stress:

$$\begin{cases} \dfrac{\partial \boldsymbol{\sigma}_n}{\partial \boldsymbol{\varepsilon}_n} = -\dfrac{\partial \boldsymbol{\sigma}_n}{\partial \boldsymbol{\varepsilon}^p_{n-1}} = \mathbb{C}^{\mathrm{ep}} \\ \dfrac{\partial \boldsymbol{\sigma}_n}{\partial \bar{\alpha}_{n-1}} = \dfrac{2\mu\sqrt{3/2}\,\sigma'_y}{3\mu + \sigma'_y}\,\boldsymbol{\nu}, \end{cases} \tag{B10}$$

where the consistent tangent of Eq. (20) appears. The partial derivatives of the internal variables are

$$\begin{cases} \dfrac{\partial \boldsymbol{\varepsilon}^p_n}{\partial \boldsymbol{\varepsilon}_n} = \sqrt{\dfrac{3}{2}}\left(\boldsymbol{\nu}\otimes\dfrac{\partial \Delta\gamma}{\partial \boldsymbol{\varepsilon}_n} + \Delta\gamma\dfrac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\varepsilon}_n}\right) \\ \dfrac{\partial \boldsymbol{\varepsilon}^p_n}{\partial \boldsymbol{\varepsilon}^p_{n-1}} = \mathbb{I} - \dfrac{\partial \boldsymbol{\varepsilon}^p_n}{\partial \boldsymbol{\varepsilon}_n} \\ \dfrac{\partial \boldsymbol{\varepsilon}^p_n}{\partial \bar{\alpha}_{n-1}} = \sqrt{\dfrac{3}{2}}\dfrac{\partial \Delta\gamma}{\partial \bar{\alpha}_{n-1}}\,\boldsymbol{\nu} \\ \dfrac{\partial \bar{\alpha}_n}{\partial \boldsymbol{\varepsilon}_n} = \dfrac{\partial \Delta\gamma}{\partial \boldsymbol{\varepsilon}_n} \\ \dfrac{\partial \bar{\alpha}_n}{\partial \boldsymbol{\varepsilon}^p_{n-1}} = -\dfrac{\partial \Delta\gamma}{\partial \boldsymbol{\varepsilon}_n} \\ \dfrac{\partial \bar{\alpha}_n}{\partial \bar{\alpha}_{n-1}} = \dfrac{3\mu}{3\mu + \sigma'_y}. \end{cases} \tag{B11}$$

At integration points in the elastic state, $\Delta\gamma = 0$, $\partial\boldsymbol{\sigma}_n/\partial\boldsymbol{\varepsilon}_n = \mathbb{C}$, and $\partial\boldsymbol{\sigma}_n/\partial\boldsymbol{\varepsilon}^p_{n-1} = -\mathbb{C}$. The internal variables are carried over unchanged from the previous step.

B.3 Design derivatives

The partial derivative with respect to the physical density $\tilde{\tilde{x}}_{ek}$, where the strain and internal variables of the previous step are fixed, is obtained by differentiating the

consistency condition of Eq. (19) with respect to $\tilde{\tilde{x}}_{ek}$:

$$\frac{\partial \Delta\gamma}{\partial \tilde{\tilde{x}}_{ek}} = \frac{\dfrac{\partial q^{\mathrm{tr}}}{\partial \tilde{\tilde{x}}_{ek}} - 3\dfrac{\partial \mu}{\partial \tilde{\tilde{x}}_{ek}}\Delta\gamma - \left.\dfrac{\partial \sigma_y}{\partial \tilde{\tilde{x}}_{ek}}\right|_{\bar{\alpha}_n}}{3\mu + \sigma_y'(\bar{\alpha}_n)}, \tag{B12}$$

where

$$\begin{cases} \dfrac{\partial q^{\mathrm{tr}}}{\partial \tilde{\tilde{x}}_{ek}} = \dfrac{q^{\mathrm{tr}}}{\mu}\dfrac{\partial \mu}{\partial \tilde{\tilde{x}}_{ek}} \\ \left.\dfrac{\partial \sigma_y}{\partial \tilde{\tilde{x}}_{ek}}\right|_{\bar{\alpha}} = \dfrac{\partial \sigma_{y0}}{\partial \tilde{\tilde{x}}_{ek}} e^{-b\bar{\alpha}} + \dfrac{\partial \sigma_\infty}{\partial \tilde{\tilde{x}}_{ek}}\left(1 - e^{-b\bar{\alpha}}\right) + \left(\sigma_\infty - \sigma_{y0}\right)\bar{\alpha} e^{-b\bar{\alpha}}\dfrac{\partial b}{\partial \tilde{\tilde{x}}_{ek}} \end{cases}.$$

Because the direction of the deviatoric part of the trial stress does not depend on the elastic constants, the design derivative of the flow direction is zero. Therefore the design derivatives of the stress and internal variables are

$$\begin{cases} \dfrac{\partial \boldsymbol{\sigma}_n}{\partial \tilde{\tilde{x}}_{ek}} = \dfrac{\partial \mathbb{C}}{\partial \tilde{\tilde{x}}_{ek}} : \boldsymbol{\varepsilon}_n^{e,\mathrm{tr}} - 2\sqrt{\dfrac{3}{2}}\left(\dfrac{\partial \mu}{\partial \tilde{\tilde{x}}_{ek}}\Delta\gamma + \mu\dfrac{\partial \Delta\gamma}{\partial \tilde{\tilde{x}}_{ek}}\right)\boldsymbol{\nu} \\ \dfrac{\partial \boldsymbol{\varepsilon}_n^p}{\partial \tilde{\tilde{x}}_{ek}} = \sqrt{\dfrac{3}{2}}\dfrac{\partial \Delta\gamma}{\partial \tilde{\tilde{x}}_{ek}}\boldsymbol{\nu} \\ \dfrac{\partial \bar{\alpha}_n}{\partial \tilde{\tilde{x}}_{ek}} = \dfrac{\partial \Delta\gamma}{\partial \tilde{\tilde{x}}_{ek}}. \end{cases} \tag{B13}$$

Because Poisson's ratio does not depend on the design, the design derivatives of the elastic constants are $\partial\mu/\partial\tilde{\tilde{x}}_{ek} = (\partial E/\partial\tilde{\tilde{x}}_{ek})/[2(1+\nu)]$ and $\partial\kappa/\partial\tilde{\tilde{x}}_{ek} = (\partial E/\partial\tilde{\tilde{x}}_{ek})/[3(1-2\nu)]$.

The design derivative of a property ϕ is obtained by writing the weight inside the square brackets of Eq. (24) as $w_i = \tilde{\tilde{x}}_{ei}^p \prod_{j\neq i}(1-\tilde{\tilde{x}}_{ej}^p)$. The closed-form derivative is given by

$$\begin{cases} \dfrac{\partial w_i}{\partial \tilde{\tilde{x}}_{ei}} = p\,\tilde{\tilde{x}}_{ei}^{p-1}\displaystyle\prod_{j\neq i}\left(1-\tilde{\tilde{x}}_{ej}^p\right) \\ \dfrac{\partial w_i}{\partial \tilde{\tilde{x}}_{ek}} = -p\,\tilde{\tilde{x}}_{ei}^p\,\tilde{\tilde{x}}_{ek}^{p-1}\displaystyle\prod_{j\neq i,k}\left(1-\tilde{\tilde{x}}_{ej}^p\right) \quad (k\neq i), \end{cases} \tag{B14}$$

and then as $\partial\phi/\partial\tilde{\tilde{x}}_{ek} = \sum_{i=1}^{N_m}(\partial w_i/\partial\tilde{\tilde{x}}_{ek})\,(\phi_i - \phi_0)$. The only difference from a single-material problem is this design derivative of the properties. Eqs. (B8)–(B13) and the backward recursion of the adjoint apply the same equations to each material field.

B.4 Partial derivatives of the objective function

Based on $\Delta\boldsymbol{\varepsilon}_n = \boldsymbol{B}(\boldsymbol{u}_n - \boldsymbol{u}_{n-1})$, the contribution of Eq. (B2) to the objective function has the partial derivatives

$$\begin{cases} \dfrac{\partial \varphi_n}{\partial \boldsymbol{u}_n} = \displaystyle\int_\Omega \boldsymbol{B}^\mathsf{T} \left(\mathbb{C}_n^{\mathrm{ep}} : \Delta\boldsymbol{\varepsilon}_n + \boldsymbol{\sigma}_n\right) \mathrm{d}\Omega \\ \dfrac{\partial \varphi_n}{\partial \boldsymbol{u}_{n-1}} = -\displaystyle\int_\Omega \boldsymbol{B}^\mathsf{T} \boldsymbol{\sigma}_n \,\mathrm{d}\Omega \\ \dfrac{\partial \varphi_n}{\partial \boldsymbol{h}_{n-1}} = \displaystyle\int_\Omega \left(\frac{\partial \boldsymbol{\sigma}_n}{\partial \boldsymbol{h}_{n-1}}\right)^\mathsf{T} : \Delta\boldsymbol{\varepsilon}_n \,\mathrm{d}\Omega \\ \dfrac{\partial \varphi_n}{\partial \tilde{\tilde{x}}_{ek}} = \displaystyle\int_{\Omega_e} \frac{\partial \boldsymbol{\sigma}_n}{\partial \tilde{\tilde{x}}_{ek}} : \Delta\boldsymbol{\varepsilon}_n \,\mathrm{d}\Omega. \end{cases} \tag{B15}$$

$\partial\varphi_n/\partial\boldsymbol{u}_{n-1}$ enters the right-hand side of the adjoint equation (B5) of the previous step, and $\partial\varphi_n/\partial\boldsymbol{h}_{n-1}$ enters the right-hand side of Eq. (B4).

B.5 Gradient through the filter and projection

Eq. (B6) is the gradient with respect to the physical densities. The gradient with respect to the design variables is obtained by tracing the projection of Eq. (23) and the filter of Eq. (22) in reverse order:

$$\begin{cases} \dfrac{\mathrm{d}f}{\mathrm{d}x_{ji}} = \displaystyle\sum_e \frac{\partial \tilde{x}_{ei}}{\partial x_{ji}} \frac{\partial \tilde{\tilde{x}}_{ei}}{\partial \tilde{x}_{ei}} \frac{\mathrm{d}f}{\mathrm{d}\tilde{\tilde{x}}_{ei}} \\ \dfrac{\partial \tilde{\tilde{x}}_{ei}}{\partial \tilde{x}_{ei}} = \dfrac{\beta\left\{1 - \tanh^2[\beta(\tilde{x}_{ei} - \eta)]\right\}}{\tanh(\beta\eta) + \tanh[\beta(1-\eta)]}. \end{cases} \tag{B16}$$

The gradients of the volume constraints are obtained by the same equation.

Of the equations above, Sections B.2 and B.3 are specific to the von Mises model, Voce hardening, and DMO interpolation, and Section B.4 is specific to the objective function. To treat thermal stress, another constitutive law, or another objective function, each has to be derived again. This is why the scope of the analytical adjoint variable method was limited to this configuration in this study.

Replication of results

The formulation and the algorithm of the proposed sensitivity analysis are described in Sections 2 to 5 and Algorithm 1, and the analysis models, the material properties, the boundary conditions, and the optimization settings are given in Sections 6 to 9, so that the results can be reproduced from the information in the manuscript. The numerical data of the figures and tables and further details of the implementation are available from the corresponding author on reasonable request.

References

[1] Michaleris, P., Tortorelli, D.A., Vidal, C.A.: Tangent operators and design sensitivity formulations for transient non-linear coupled problems with applications to elastoplasticity. International Journal for Numerical Methods in Engineering **37**(14), 2471–2499 (1994) https://doi.org/10.1002/nme.1620371408

[2] Maute, K., Schwarz, S., Ramm, E.: Adaptive topology optimization of elastoplastic structures. Structural Optimization **15**(2), 81–91 (1998) https://doi.org/10.1007/BF01278493

[3] Wallin, M., Jönsson, V., Wingren, E.: Topology optimization based on finite strain plasticity. Structural and Multidisciplinary Optimization **54**(4), 783–793 (2016) https://doi.org/10.1007/s00158-016-1435-0

[4] Amir, O.: Stress-constrained continuum topology optimization: a new approach based on elasto-plasticity. Structural and Multidisciplinary Optimization **55**(5), 1797–1818 (2017) https://doi.org/10.1007/s00158-016-1618-8

[5] Kato, J., Hoshiba, H., Takase, S., Terada, K., Kyoya, T.: Analytical sensitivity in topology optimization for elastoplastic composites. Structural and Multidisciplinary Optimization **52**(3), 507–526 (2015) https://doi.org/10.1007/s00158-015-1246-8

[6] Alberdi, R., Khandelwal, K.: Design of periodic elastoplastic energy dissipating microstructures. Structural and Multidisciplinary Optimization **59**, 461–483 (2019) https://doi.org/10.1007/s00158-018-2076-2

[7] Alberdi, R., Zhang, G., Li, L., Khandelwal, K.: A unified framework for nonlinear path-dependent sensitivity analysis in topology optimization. International Journal for Numerical Methods in Engineering **115**, 1–56 (2018) https://doi.org/10.1002/nme.5794

[8] Abueidda, D.W., Kang, Z., Koric, S., James, K.A., Jasiuk, I.M.: Topology optimization for three-dimensional elastoplastic architected materials using a path-dependent adjoint method. International Journal for Numerical Methods in Engineering **122**(8), 1889–1910 (2021) https://doi.org/10.1002/nme.6604

[9] Granlund, G., Wallin, M.: Large-scale elasto-plastic topology optimization. International Journal for Numerical Methods in Engineering **125**(23), 7583 (2024) https://doi.org/10.1002/nme.7583

[10] Griewank, A., Walther, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, 2nd edn. SIAM, Philadelphia (2008)

[11] Nørgaard, S.A., Sagebaum, M., Gauger, N.R., Lazarov, B.S.: Applications

of automatic differentiation in topology optimization. Structural and Multidisciplinary Optimization **56**(5), 1135–1146 (2017) https://doi.org/10.1007/s00158-017-1708-2

[12] Chandrasekhar, A., Sridhara, S., Suresh, K.: AuTO: a framework for automatic differentiation in topology optimization. Structural and Multidisciplinary Optimization **64**, 4355–4365 (2021) https://doi.org/10.1007/s00158-021-03025-8

[13] Yin, S., Ferreira, B.P., Kuś, G., Bessa, M.A.: Design of a specimen to train path-dependent deep learning material models from a single uniaxial test: Eliciting strain diversity via automatically differentiable elastoplastic topology optimization. Journal of the Mechanics and Physics of Solids **214**, 106631 (2026) https://doi.org/10.1016/j.jmps.2026.106631

[14] Ogawa, S., Ishida, H., Suzuki, K.: Using automatic differentiation with checkpointing for elastoplastic topology optimization. Computers & Structures **330**, 108353 (2026) https://doi.org/10.1016/j.compstruc.2026.108353

[15] Griewank, A., Walther, A.: Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Transactions on Mathematical Software **26**(1), 19–45 (2000) https://doi.org/10.1145/347837.347846

[16] Jia, Y., Zhang, X.S.: Multimaterial topology optimization for finite strain elastoplasticity: Theory, methods, and applications. Computer Methods in Applied Mechanics and Engineering **449**, 118445 (2026) https://doi.org/10.1016/j.cma.2025.118445

[17] Farrell, P.E., Ham, D.A., Funke, S.W., Rognes, M.E.: Automated derivation of the adjoint of high-level transient finite element programs. SIAM Journal on Scientific Computing **35**(4), 369–393 (2013) https://doi.org/10.1137/120873558

[18] Mitusch, S.K., Funke, S.W., Dokken, J.S.: dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. Journal of Open Source Software **4**(38), 1292 (2019) https://doi.org/10.21105/joss.01292

[19] Xue, T., Liao, S., Gan, Z., Park, C., Xie, X., Liu, W.K., Cao, J.: JAX-FEM: A differentiable GPU-accelerated 3D finite element solver for automatic inverse design and mechanistic data science. Computer Physics Communications **291**, 108802 (2023) https://doi.org/10.1016/j.cpc.2023.108802

[20] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q.: JAX: Composable transformations of Python+NumPy programs. http://github.com/jax-ml/jax (2018)

[21] Sharma, D., Salgado, I., Huang, L., Wang, H.-P., Cao, J.: JAX-FEM-ANISO:

Differentiable GPU-accelerated finite element framework for inverse identification of finite-strain anisotropic plasticity. arXiv:2606.17390 (2026)

[22] Meyer, N.: torch-fem: GPU accelerated differentiable finite elements for solid mechanics with PyTorch. https://github.com/meyer-nils/torch-fem. Software. Differentiation through load increments by an implicit adjoint was introduced in March 2026. Accessed 23 September 2026 (2026)

[23] Christianson, B.: Reverse accumulation and attractive fixed points. Optimization Methods and Software **3**(4), 311–326 (1994) https://doi.org/10.1080/10556789408805572

[24] Griewank, A., Faure, C.: Reduced functions, gradients and Hessians from fixed-point iterations for state equations. Numerical Algorithms **30**(2), 113–139 (2002) https://doi.org/10.1023/A:1016051717120

[25] Albring, T., Sagebaum, M., Gauger, N.R.: Efficient aerodynamic design using the discrete adjoint method in SU2. In: 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2016). https://doi.org/10.2514/6.2016-3518 . AIAA Paper 2016-3518

[26] Sagebaum, M., Albring, T., Gauger, N.R.: High-performance derivative computations using CoDiPack. ACM Transactions on Mathematical Software **45**(4), 38 (2019) https://doi.org/10.1145/3356900

[27] Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., Vert, J.-P.: Efficient and modular implicit differentiation. In: Advances in Neural Information Processing Systems, vol. 35, pp. 5230–5242 (2022)

[28] Bolte, J., Pauwels, E., Vaiter, S.: One-step differentiation of iterative algorithms. In: Advances in Neural Information Processing Systems, vol. 36, pp. 77089–77103 (2023)

[29] Wu Fung, S., Heaton, H., Li, Q., McKenzie, D., Osher, S., Yin, W.: JFB: Jacobian-free backpropagation for implicit networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 6648–6656 (2022). https://doi.org/10.1609/aaai.v36i6.20619

[30] Mehmood, S., Ochs, P.: Fixed-point automatic differentiation of forward–backward splitting algorithms for partly smooth functions. arXiv:2208.03107 (2022)

[31] Simo, J.C., Hughes, T.J.R.: Computational Inelasticity. Springer, New York (1998)

[32] Gao, T., Zhang, W.: Topology optimization involving thermo-elastic stress loads.

Structural and Multidisciplinary Optimization **42**, 725–738 (2010) https://doi.org/10.1007/s00158-010-0527-5

[33] Voce, E.: The relationship between stress and strain for homogeneous deformation. Journal of the Institute of Metals **74**, 537–562 (1948)

[34] Bruns, T.E., Tortorelli, D.A.: Topology optimization of non-linear elastic structures and compliant mechanisms. Computer Methods in Applied Mechanics and Engineering **190**(26–27), 3443–3459 (2001) https://doi.org/10.1016/S0045-7825(00)00278-4

[35] Bourdin, B.: Filters in topology optimization. International Journal for Numerical Methods in Engineering **50**(9), 2143–2158 (2001) https://doi.org/10.1002/nme.116

[36] Wang, F., Lazarov, B.S., Sigmund, O.: On projection methods, convergence and robust formulations in topology optimization. Structural and Multidisciplinary Optimization **43**(6), 767–784 (2011) https://doi.org/10.1007/s00158-010-0602-y

[37] Stegmann, J., Lund, E.: Discrete material optimization of general composite shell structures. International Journal for Numerical Methods in Engineering **62**(14), 2009–2027 (2005) https://doi.org/10.1002/nme.1259

[38] Svanberg, K.: The method of moving asymptotes—a new method for structural optimization. International Journal for Numerical Methods in Engineering **24**(2), 359–373 (1987) https://doi.org/10.1002/nme.1620240207