SANDIA REPORT

SAND2021-13231
Printed October 2021

Sandia
National
Laboratories

Optimization-based Design for Manufacturing

Joshua Robbins, Miguel Aguilo, Ryan Alberdi, Brett Clark, Bradley Jared, Scott Jensen, Kyle Johnson, Ryan Viertel

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods

ABSTRACT

This report provides detailed documentation of the algorithms that where developed and implemented in the Plato software [53] over the course of the Optimization-based Design for Manufacturing LDRD project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SHAPE AND TOPOLOGY OPTIMIZATION

Abstract: The typical topology optimization workflow uses a design domain that does not change during the optimization process. Consequently, features of the design domain, such as the location of loads and constraints, must be determined in advance and are not optimizable. A method is proposed herein that allows the design domain to be optimized along with the topology. This approach uses topology and shape derivatives to guide nested optimizers to the optimal topology and design domain. The details of the method are discussed, and examples are provided that demonstrate the utility of this approach.

1.1. Introduction

Topology optimization aims to distribute material within a given design domain such that a desired performance objective is extremized while meeting prescribed constraints. Tracing back to the work of Bendsøe and Kikuchi [12] and spurred by the maturation of additive manufacturing techniques, the field of topology optimization has developed into a highly active research area while increasingly becoming adopted into engineering practice [34]. Because of the freeform nature of topology optimization, it is often used as a preliminary design tool when the overall form is not clear due to an unexplored design space or convoluted set of requirements.

Shape optimization is another field of computational design that has been a focus of research for decades [27] and found adoption into engineering workflows. This approach is more restrictive than topology optimization as it focuses on improving the size and shape of structural boundaries, and is typically used at a later stage of the design process after the overall form is already decided. As such, topology optimization (TO) and shape optimization (SO) are often seen as separate stages which can be combined sequentially in a design workflow [60, 70].

Early work in shape optimization focused on using the coordinates of nodes within a finite element model as design variables [72], and later considered spline-based paremeterizations of the boundary [41, 16], or direct optimization of CAD parameters [71]. Advances in node-based shape optimization such as the traction method [7, 6] can lead to smooth designs with exact boundary representation and mesh-independence, but this approach is still limited to relatively small design changes. On the other hand, topology optimization allows for exceptional geometric freedom as the interface between material and void can be changed throughout the design process [13]. The most commonly employed topology optimization approach uses a density-based parameterization (see [55] for a review of different approaches), wherein the problem of finding a discrete characteristic function representing the presence (1) or absence (0) of material at every point in the design domain is relaxed to that of finding a continuous density-like field. This field is represented on a fixed background mesh which is also used for evaluation of the physics problem.

Relaxation to a density-like field allows for the use of nonlinear programming techniques for performing optimization while retaining the design freedom of the characteristic function. However, it also leads to mesh-dependent results and introduces non-physical intermediate density values which represent neither material nor void and result in a blurred interface. The mesh-dependency issue has been addressed through the development of filtering techniques which introduce a minimum length scale [17, 15, 38]. To reduce the presence of intermediate density values, penalization techniques such as the ubiquitous SIMP approach [52, 11] are used to degrade the effectiveness of material within regions of intermediate densities by penalizing the physics model. Additionally, projection techniques [25, 65] can be employed to further drive densities to discrete values and obtain a more crisp interface.

Aspects of both shape and topology optimization are present in level-set approaches which utilize the iso-contours of an implicit level-set function to represent the material interface [64]. Shape derivatives are used to drive this interface in a similar manner to node-based shape optimization, but the implicit representation allows for merging, splitting, and disappearance of holes [4, 66]. As the level-set function is commonly represented on a fixed background mesh, the implicit interface needs to be mapped to a material domain for carrying out the physics computations. A common way to do this is by using the level-set to define a density distribution [4, 66], which in turn results in a blurred interface representation. Using a density distribution allows for nucleation of new holes through e.g. topological derivatives [3, 18], making the approach similar to density-based topology optimization with the associated advantages and drawbacks. Alternatively, a crisp boundary for evaluation of the physics model can be obtained by generating a conforming discretization of the evolving implicit interface [26, 2]. As with node-based shape optimization, mesh deformation during the interface update and an increased computational cost associated with a remeshing process are downsides of this approach. Immersed boundary techniques have also been utilized within level-set approaches to achieve a crisp boundary representation for the physics model while retaining a fixed background mesh. The extended finite element method (XFEM), wherein additional shape functions are utilized to enrich the solution field near the interface, is most commonly used in these approaches [10, 37, 54].

Other work has focused on combining shape and topology optimization to retain the flexibility of density-based topology optimization but with an explicit representation of and control over the material interface. In [23] the design domain is parameterized by B-splines to allow for shape optimization, and topological changes are enabled by a “bubble" method for inserting holes after the shape has been optimized. A staggered approach wherein density-based topology optimization and spline-based shape optimization are alternatingly used was presented in [42, 43, 14]. In this work a number of cycles are carried out wherein density-based topology optimization is first performed, the resulting iso-surface is used to generate a spline parameterization for shape optimization, and a conforming mesh is generated on the parameterized domain for use in the next cycle. In the method presented in [20, 39], a deformable simplical complex (DSC) is used to explicitly conform to the material interface and the boundary is evolved by optimizing the position of nodes using shape sensitivities. The use of a DSC naturally allows for topological changes through merging of holes and new holes can be introduced using topological derivatives [20] or other methods [39]. An extension wherein density-based topology optimization is performed as a first step and the DSC is generated from the density distribution for shape optimization in a subsequent step is presented in [47]. In [50], an approach is presented that uses node-based shape optimization with an evolutionary-type element removal strategy for topological changes. Recently, a combined approach involving XFEM-based immersed boundary level-set and a density-based parameterization for topology optimization has been proposed [8]. This provides an implicit representation of the boundary while also allowing for topological freedom by nucleating and merging holes smoothly with the density field. Along a similar vein, a density-based parameterization was combined with an embedded domain approach to shape optimization in [56].

In the above studies a design domain is specified prior to carrying out optimization, restricting the available design space a priori. A handful of works have instead considered using shape and topology optimization to include the design domain as part of the optimization problem. In [35, 32] this is done by allowing enlargement of the design domain through the addition of pixels (new design elements) to the boundary. An inner loop optimizes the topology and once converged, an outer loop optimizes the design domain through the addition of pixels, subsequently increasing the number of design variables in the inner loop for the next iteration. In order to avoid discontinuous jumps in sensitivities, the new pixels are added such that they have values of 0 and the design remains the same as before, but as a subset of the new design space. In [33], this approach was expanded by performing mesh refinement on the adapted design space. Related work considers the combined shape and topology optimization of plate structures, wherein the in-plane topology is optimized along with the out-of plane shape through parameterization by B-splines [5] or NURBS [30]. More recently, a similar approach to [5, 30] was proposed in [67] wherein NURBS are used to parameterize the design domain boundary. A staggered solution scheme is employed wherein the topology of the design domain is first optimized, followed by the shape of the design domain, and this process is iteratively carried out. This approach was subsequently applied to full 3D design domains as well as to 2.5D design domains where planar cross-sections are optimized and out-of-plane response is modeled by lower dimension elements [69, 68]. In [68], the staggered solution scheme is compared to a monolithic one where both the topology and shape design variables are updated concurrently. Finally, in [63], a surrogate-based multiobjective approach for parameterized shape optimization is combined with level-set topology optimization and compared to the case where just topology optimization and just shape optimization are used. In the combined approach, a Gaussian process surrogate model is first constructed by sampling the space of CAD parameter design variables and a multi-objective genetic algorithm is used to minimize both mass and maximum stress. After each iteration, the new design domain is used for carrying out level-set topology optimization. For the three benchmark design examples considered, the combined shape and topology optimization approach led to a greater reduction in mass than just topology optimization or shape optimization alone, while also meeting stress requirements.

In this work, density-based topology optimization is combined with parametric shape optimization to simultaneously optimize the topology and design domain. Such an approach enlarges the design space by not restricting the topology to a static design domain, allowing for more optimal designs to be found. Moreover, by using CAD parameters as the variables for shape optimization of the design domain, many of the important design decisions that are made before density-based topology optimization is carried out, such as the shape of design domain and the location of loads and constraints, are removed. Such a parameterization also allows designers to specify exactly which features of the design domain should be allowed to vary, and which should

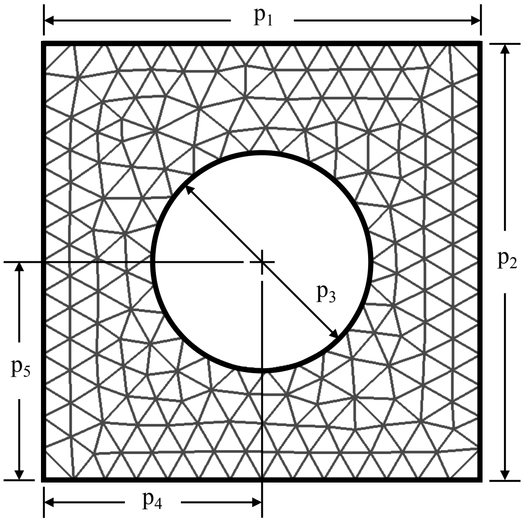


Figure 1-1. Example set of shape parameters, p. The topology field, ϕ, is defined at the mesh nodes.

be fixed. Thus, the domain can still be restricted to conform to certain specifications needed in assemblies, etc.

1.2. Approach

Let $p \in \mathbb{R}^{n_p}$ be the set of shape parameters that defines the design domain, Ω_d, and which will be used for shape optimization (see Figure 1-1). This design domain is discretized into Ω_d^h upon which a nodal field of design variables, $\phi \in \mathbb{R}^{n_n}$, is defined for topology optimization. The optimization problem written in these terms takes the form

$$\min_{(p,\phi)} : f(p,\phi) = F(p,\phi,U(p,\phi)) \tag{1.1}$$

$$s.t. : g(p,\phi,U(p,\phi)) = 0 \tag{1.2}$$

$$: h(p,\phi) \leq 0 \tag{1.3}$$

where

$$f : \mathbb{R}^{n_p} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R} \tag{1.4}$$

$$g : \mathbb{R}^{n_p} \times \mathbb{R}^{n_n} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_d} \tag{1.5}$$

$$h : \mathbb{R}^{n_p} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R} \tag{1.6}$$

and $U \in \mathbb{R}^{n_d}$ is the state field that satisfies the PDE constraint in Equation (1.2) (the governing physics model) for a given p and ϕ. The number of shape parameters is n_p, n_n is the number of nodes in the finite element discretization, and n_d is the total number of degrees of freedom in the state.

Gradient-based optimization is used to find the optimal shape and topology in Equation (2.3), so total derivatives of f and h with respect to the shape parameters, p, and topology design variables, ϕ, are required.

Topology Sensitivities: Total derivatives of f are found via the adjoint variable method (AVM). For this purpose, the Lagrangian function is given by:

$$\hat{f} = f + \lambda^T g \tag{1.7}$$

where $\lambda \in \mathbb{R}^{n_d}$ is the adjoint vector. Since g is zero for all p and ϕ, $\hat{f} \equiv f$ for any value of the adjoint vector when U satisfies Equation (1.2). In this case,

$$\frac{d\hat{f}}{d\phi} \equiv \frac{df}{d\phi} = \frac{dF}{d\phi} + \lambda^T \frac{dg}{d\phi} \tag{1.8}$$

Using the chain rule to expand the total derivatives in Equation (1.8) leads to

$$\frac{df}{d\phi} = \frac{\partial F}{\partial \phi} + \frac{\partial F}{\partial U}\frac{\partial U}{\partial \phi} + \lambda^T \left(\frac{\partial g}{\partial \phi} + \frac{\partial g}{\partial U}\frac{\partial U}{\partial \phi} \right) \tag{1.9}$$

for constant p. Collecting terms that involve derivatives of the state with respect to the topology yields:

$$\frac{df}{d\phi} = \frac{\partial F}{\partial \phi} + \lambda^T \frac{\partial g}{\partial \phi} + \left(\frac{\partial F}{\partial U} + \lambda^T \frac{\partial g}{\partial U} \right) \frac{\partial U}{\partial \phi} \tag{1.10}$$

The last term in Equation (1.10) can be eliminated by finding the adjoint vector that satisfies:

$$\frac{\partial F}{\partial U} + \lambda^T \frac{\partial g}{\partial U} = 0. \tag{1.11}$$

To summarize, the objective and gradient of the objective are computed by the following process:

1. Given p and ϕ, find U that satisfies Equation (1.2). Newton iteration is used for this purpose, i.e., starting with $k = 0$ and $U^k = 0$, solve

$$\left. \frac{\partial g}{\partial U} \right|_{U=U^k} \Delta U^k = -g(U^k) \tag{1.12}$$

$$U^{k+1} = U^k + \Delta U^k \tag{1.13}$$

$$k \leftarrow k + 1. \tag{1.14}$$

   until $\left|\Delta U^k\right|$ or $\left|g\left(U^k\right)\right|$ is smaller than a specified tolerance, ε_u or ε_g, respectively.

At this equilibrium state, U:

2. Compute the objective value, F, in Equation (2.3).

3. Find λ that satisfies

$$\frac{\partial g}{\partial U}^T \lambda = -\frac{\partial F}{\partial U} \tag{1.15}$$

4. Compute the total derivative of the objective:

$$\frac{df}{d\phi} = \frac{\partial F}{\partial \phi} + \lambda^T \frac{\partial g}{\partial \phi} \tag{1.16}$$

Shape Sensitivities: The adjoint variable method can also be used to find derivatives of the objective with respect to the shape parameters, p. This would require computing the partial derivatives of the objective and PDE constraint, $\frac{\partial F}{\partial p}$ and $\frac{\partial g}{\partial p}$. In the current work, an alternative approach is proposed wherein the shape sensitivity is computed by

$$\frac{dF}{dp} = \frac{dF}{dX}\frac{dX}{dp} \tag{1.17}$$

etc., where $X \in \mathbb{R}^{n_x}$ is the global vector of nodal coordinates, and n_x is the total number mesh nodes times the number of space dimensions.

The dependence of the objective and constraint functions upon the nodal coordinates is not shown explicitly in Equations (2.3)-(2.5). However, since these functions are computed on the discrete domain, Ω_d^h, which is composed of finite elements whose node locations are in turn a function of the shape parameters, the form can be expressed as follows:

$$\min_{(p,\phi)} : f(p,\phi) = F(X(p),\phi,U(X(p),\phi)) \tag{1.18}$$

$$s.t. : g(X(p),\phi,U(X(p),\phi)) = 0 \tag{1.19}$$

$$: h(X(p),\phi) \leq 0 \tag{1.20}$$

The AVM is used to compute total derivatives with respect to X, and derivatives of the nodal coordinates with respect to the shape parameters are computed using the Engineering Sketch Pad (ESP) [21, 28].

ESP is built on top of OpenCSM [29], EGADS [29], and OpenCASCADE [48] and provides i) a web-based user interface for creating parameterized CAD models and ii) a library that computes the sensitivity of these models with respect to the defined parameters.

1.3. Implementation

The approach described above has been implemented in the Plato software [53] developed at Sandia National Laboratories. Plato is designed to simplify the implementation of physics (PDE constraints) and criteria (objectives and constraints) for gradient-based shape and topology optimization. This is accomplished by defining templated abstractions for physics residual

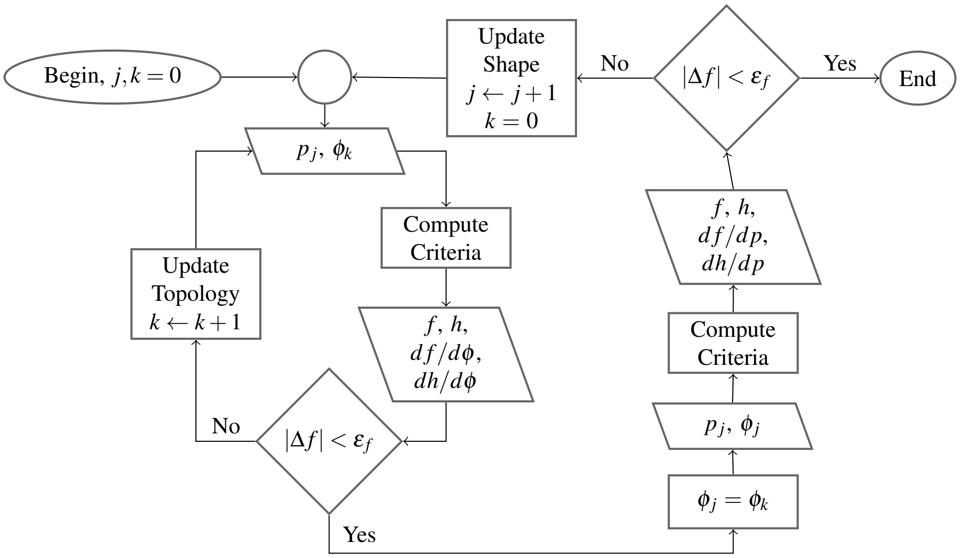


Figure 1-2. Flowchart for the nested optimization approach used for concurrent shape and topology optimization.

functions (e.g., g) and criterion functions (e.g., F and h) that permit the use of automatic differentiation (AD) [49] for computing all relevant partial derivatives, i.e.,

$$\frac{\partial g}{\partial U}, \frac{\partial g}{\partial \phi}, \frac{\partial g}{\partial X}, \frac{\partial F}{\partial U}, \frac{\partial F}{\partial \phi}, \frac{\partial F}{\partial X}, \frac{\partial h}{\partial \phi}, \text{ and } \frac{\partial h}{\partial X}.$$

Developers are able to implement functions for residuals and criteria, and derivatives are computed automatically. The procedure described in Section (1.2) is implemented for the standard classes of PDEs (elliptic, parabolic, and hyperbolic), based on these abstractions.

A nested optimization procedure is utilized, as shown in Figure (1-2). The method of moving asymptotes (MMA) [58] is used for both the inner and outer loops. Indices j and k indicate the shape and topology iterations, respectively. Beginning with $j,k = 0$, the initial design domain is used as the basis for an inner topology optimization loop in k. Once the inner loop converges, either by stagnation in the objective value or by exceeding a maximum number of iterations, the resulting topology, $\phi_j = \phi_k$, is used to compute the criterion derivatives, e.g.,

$$f_{,X} = \left.\frac{df}{dX}\right|_{X=X(p_j),\phi=\phi_j} \tag{1.21}$$

$$X_{,p} = \left.\frac{dX}{dp}\right|_{p=p_j} \tag{1.22}$$

$$\frac{df}{dp} = f_{,X} X_{,p} \tag{1.23}$$

where $\square_{,X}$ denotes differentiation with respect to X, etc. As described in Section (1.2), the AVM is used to compute $f_{,X}$, and ESP is used to compute $X_{,p}$. The current values for f and h and their derivatives are used to update the shape parameters, and the finite element mesh is regenerated. This nested optimization process is repeated until either the objective value stagnates or a maximum number of outer iterations is exceeded.

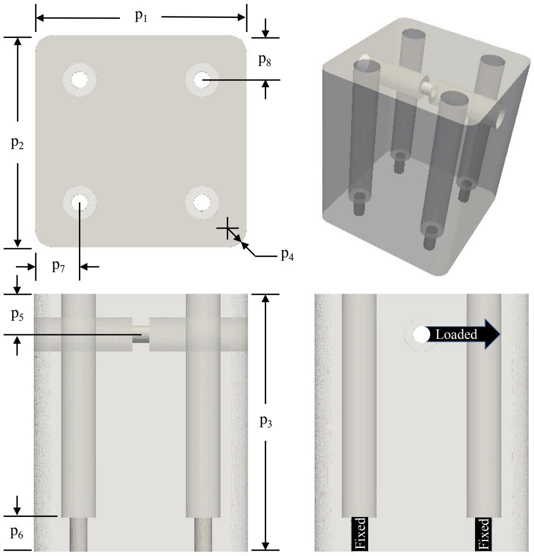


Figure 1-3. Shape parameter definitions and load scenario for compliance minimization problem. The coordinate axes, X, Y, and Z are aligned with shape parameters, p_1, p_2, and p_3, respectively.

Verification: Shape derivatives were verified on the example problem shown in Figure (1-3). The four holes at the base of the design domain are given a fixed zero displacement, and a mechanical load is applied to the hole at the top as shown. The topology field is one everywhere, and the objective is mechanical compliance.

The shape parameters are normalized to vary between zero and one:

$$P_i = P_i^{min} + p_i \left(P_i^{max} - P_i^{min} \right) \tag{1.24}$$

where P_i^{min} and P_i^{max} are the minimum and maximum values of P_i. Derivatives of the objective with respect to parameters p_1 and p_2 are used as a step direction for 20 steps of magnitude 0.05 while all other parameters are held fixed. The predicted values, f_p, are computed by adding the gradient multiplied by the step, s,

$$f_p^{k+1} = f(p^k) + \left. \frac{df}{dp} \right|_{p=p^k} s \tag{1.25}$$

and the realized values, f_r, are computed by updating the shape parameters with the step and evaluating the objective at the new values.

$$p^{k+1} = p^k + s \tag{1.26}$$

$$f_r^{k+1} = f(p^{k+1}) \tag{1.27}$$

$$k \leftarrow k+1 \tag{1.28}$$

The predicted versus realized objective values are shown in Figure (1-4). The average error in the shape derivative is 0.074 which is not particularly accurate, however it will be shown that this is accurate enough for the shape optimization problems considered.

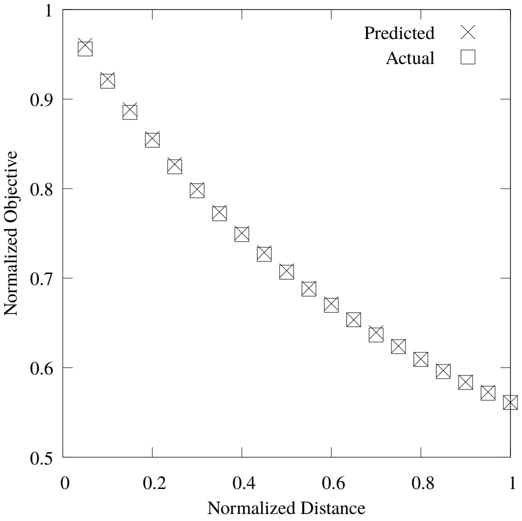


Figure 1-4. Predicted and actual normalized objective value versus distance in descent direction.

To verify that the implentation yields the correct optimum, the objective surface in p_1 and p_2 was determined by computing the forward solutions, objective, and constraint values at each grid point, (p_1, p_2) in $[0,1] \times [0,1]$ at intervals of 0.1 (See Figure 1-5). The design volume, $V(p,\phi)$, was constrained to be constant,

$$h = V(p,\phi^0) - V(p^0,\phi^0) = 0. \tag{1.29}$$

The topology field is fixed as one everywhere, $\phi^0 = \mathbf{1}$, and p^0 are the initial parameter values. The iso-contour of the constraint was then computed, and the location of the minimum objective value on this curve was determined to be 0.7414 at $p_1 = 0.077482$ and $p_2 = 1.0$. The minimum objective value found by optimizing with Plato is 0.7400 and occurs at $p_1 = 0.077478$ and $p_2 = 1.0$.

Topology derivatives were also verified by stepping in the descent direction (Figure (1-6)). Topology derivatives are computed much more accurately than shape derivatives because the topology can be updated without regenerating the mesh. Accuracy as a function of step size is given in Table (1-1).

1.4. Results

The following examples include two load cases that are weighted by a parameter, α:

$$\begin{aligned}
\text{(load case 1):}\quad & g_1\left(p,\phi,U_1\left(p,\phi\right)\right) = 0 & (1.30)\\
\text{(load case 2):}\quad & g_2\left(p,\phi,U_2\left(p,\phi\right)\right) = 0 & (1.31)\\
\text{(objective):}\quad & f\left(p,\phi\right) = \alpha F_1\left(p,\phi,U_1\right) & \\
& \quad + \left(1-\alpha\right) F_2\left(p,\phi,U_2\right) & (1.32)
\end{aligned}$$

where F_1 and F_2 are the quantities of interest for the respective load cases.

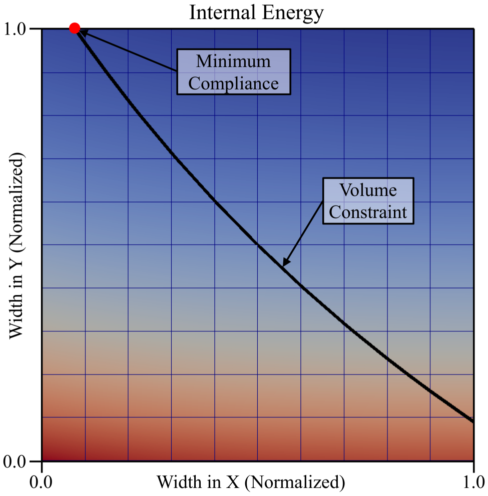


Figure 1-5. Paint plot of the objective surface in p_1 (Width in X) and p_2 (Width in Y). The curve is the volume constraint, and the minimum objective value is shown in red.

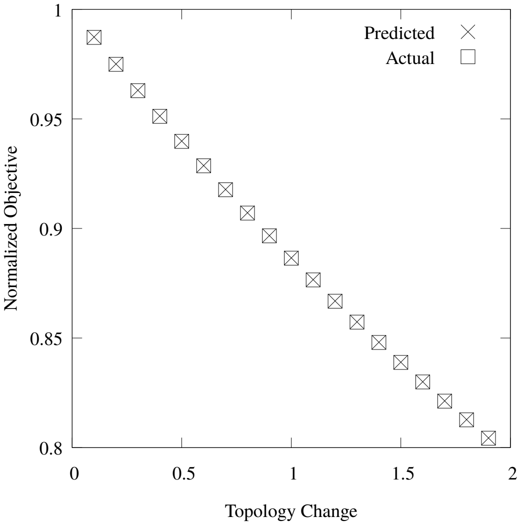


Figure 1-6. Predicted and actual normalized objective value versus distance in the descent direction for a step size of 0.1.

Table 1-1. Accuracy of topology derivatives versus step size.

Step Size	Error
1×10^{-1}	1.1274×10^{-2}
1×10^{-2}	1.1268×10^{-3}
1×10^{-3}	1.2855×10^{-4}
1×10^{-4}	1.2873×10^{-5}
1×10^{-5}	1.2875×10^{-6}
1×10^{-6}	7.2156×10^{-7}
1×10^{-7}	6.4192×10^{-6}

Table 1-2. Optimal shape parameters for varying priority, α.

α	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
0.1	0.99	0.33	0.0	0.51	1.0	1.0	0.02	0.47
0.3	0.94	0.58	0.0	0.51	1.0	1.0	0.13	0.31
0.5	0.77	0.76	0.0	0.51	1.0	1.0	0.25	0.24
0.7	0.60	0.91	0.0	0.51	1.0	1.0	0.32	0.14
0.9	0.36	1.0	0.0	0.50	1.0	1.0	0.51	0.03

1.4.1. Mechanical Compliance

In this example, the load cases g_1 and g_2 correspond to mechanical loads in Y and X, respectively, on the design domain shown in Figure (1-3). The quantities of interest, F_1 and F_2, are the internal energy (compliance) of each load case, and the volume is constrained to $1/4$ of the initial design domain. That is:

$$g_K = \mathscr{K}(p,\phi)\,U_K - \mathscr{F}_K(p) = 0 \tag{1.33}$$

$$F_K = \frac{1}{2}\mathscr{F}_K^T U_K = \frac{1}{2}U_K^T \mathscr{K} U_K \;\; \text{(no sum on } K\text{)} \tag{1.34}$$

for $K = 1,2$. The global stiffness matrix is given by

$$\mathscr{K}(p,\phi) = \underset{e\in\Omega_d^h(p)}{\mathbb{A}} k^e \tag{1.35}$$

where k_e is the local stiffness matrix for element e, and $\mathbb{A}$ represents the standard finite element assembly operation (see, e.g., [31]) over elements in the current mesh, Ω_d^h. The dependence of the mesh on the shape parameters, p, is shown explicitly to emphasize that the discretization is updated any time the shape parameters change. For number of dimensions, n_D, and number of nodes per element, n_n^e, the nodal forces, r_{iI}^e, $(i = 1,...,n_D$ and $I = 1,...,n_n^e)$ for element e are a function of the topology, position, and state of the nodes that comprise the element:

$$\phi^e = \{\phi_I \mid I = 1,...,n_n^e\} \tag{1.36}$$

$$x^e = \{x_{iI} \mid i = 1,...,n_D,\ I = 1,...,n_n^e\} \tag{1.37}$$

$$u^e = \{u_{iI} \mid i = 1,...,n_D,\ I = 1,...,n_n^e\} \tag{1.38}$$

$$r_{iI}^e = \int_{\Omega^e} \rho(\phi^e) B_{Ij}^e(x^e)\sigma_{ij}(x^e,u^e)d\Omega \tag{1.39}$$

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl} \tag{1.40}$$

$$\varepsilon_{kl} = (B_{Jk}^e u_{lJ} + B_{Jl}^e u_{kJ})/2 \tag{1.41}$$

$$c_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{1.42}$$

$$\rho\left(\phi^e\right) = (N_I\phi_I)^3 \tag{1.43}$$

In the preceding, B_{Ii}^e is the gradient matrix, N_I are the basis functions, c_{ijkl} is the fourth rank elastic stiffness tensor, and λ and μ are material constants. For more information on the finite element procedure or elastostatics, consult any introductory text, e.g. [31]. The element stiffness matrix is the derivative of the nodal forces with respect to the nodal displacements,

$$k_{iIjJ}^e = r_{iI,u_{jJ}}^e \tag{1.44}$$

Because automatic differentiation is used to compute derivatives with respect to state (in this case, displacements), nodal position, and topology, closed-form expressions for these quantities are not needed.

The global forcing vector is given by

$$\mathscr{F}(p) = \underset{e\in\Gamma_t^h(p)}{\mathbb{A}} f^e \tag{1.45}$$

$$f_{iI}^e = \int_{\Gamma^e} N_I\tau_i d\Gamma \tag{1.46}$$

for the surface, Γ_t^h, on which the load, τ, is applied.

Figure (1-7) shows a comparison of performance for varying priority between the vertical (g1) and lateral (g2) load cases for designs that are topology optimized and designs that are shape and topology optimized. Note that the compliance decreases by roughly 75 percent by allowing the mount locations to move during optimization. The degree of improvement achieved by concurrently optimizing the design domain with the topology depends on the initial values of the shape parameters. For example, decreasing p_3 would clearly improve the stiffness in all cases; however, in general cases, changes to the design domain that improve performance may not be obvious.

Designs with optimal shape and topology are shown in Figure (1-8) and the optimal shape parameters are given in Table (1-2). As might be expected, the fillet radius, p_4, does not appear to have a strong effect on the stiffness, so this parameter does not change significantly. Parameters p_3 and p_5 determine the distance of the loaded surface from the fixed surfaces, so these values are consistently 0.0 and 1.0, respectively, which minimizes the distance. Parameter p_6 controls the

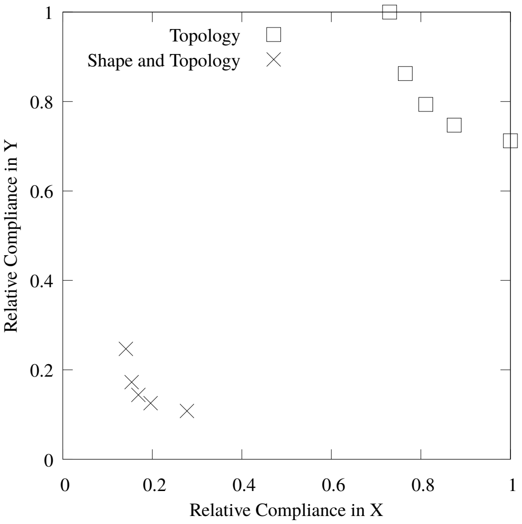
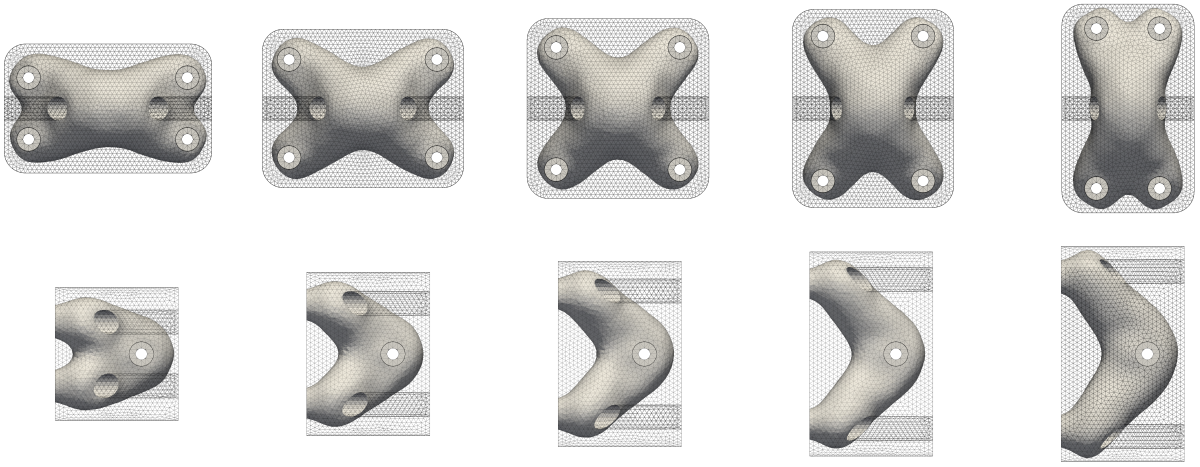


Figure 1-7. Compliance in Y versus compliance in X for varying priority, α.



(f) $\alpha = 0.1$ (g) $\alpha = 0.3$ (h) $\alpha = 0.5$ (i) $\alpha = 0.7$ (j) $\alpha = 0.9$

Figure 1-8. Results for concurrent shape and topology optimization for varying priority, α, assigned to load case 1 (load in the Y direction).

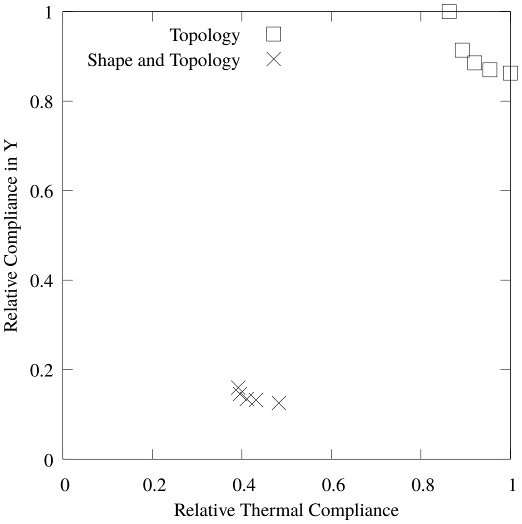


Figure 1-9. Compliance in Y versus thermal compliance for varying priority, α.

depth of the mounting bolt holes and is consistently the maximum value of 1.0. The bolt hole spacing in the X direction is determined by parameters p_1 and p_7, and the Y spacing is determined by parameters p_2 and p_8. Note that these parameters vary oppositely with respect to load priority, α, due to the near symmetry of the design domain. That is, the designs for $\alpha = 0.1$ and $\alpha = 0.9$ are equivalent given a 90 degree rotation, and similarly for $\alpha = 0.3$ and 0.7.

1.4.2. Thermal and Mechanical Compliance

In this example, the load cases are i) a mechanical load in the Y direction and ii) a thermal flux again for the design domain shown in Figure (1-3). The quantities of interest are the internal elastic energy and internal thermal energy, respectively. The PDE constraint, g_1, and objective, F_1, for the mechanical load case are as defined in Section (1.4.2). The thermal load case is:

$$g_2 = \mathscr{S}(p,\phi)T - \mathscr{Q}(p) = 0 \tag{1.47}$$

$$F_2 = \frac{1}{2}\mathscr{Q}^T T = \frac{1}{2}T^T \mathscr{S} T \tag{1.48}$$

The global stiffnes matrix is defined as:

$$\mathscr{S}(p,\phi) = \underset{e \in \Omega_d^h(p)}{\mathsf{A}} s^e_{I,T_J} \tag{1.49}$$

$$T^e = \{T_I \mid I = 1,...,n_n^e\} \tag{1.50}$$

$$s^e_I = \int_{\Omega^e} \rho(\phi^e) B^e_{Ii}(x^e) q_i(x^e, T^e) d\Omega \tag{1.51}$$

$$q_i = \kappa_{ij} B^e_{Jj} T_J \tag{1.52}$$

$$\kappa_{ij} = \kappa \delta_{ij} \tag{1.53}$$

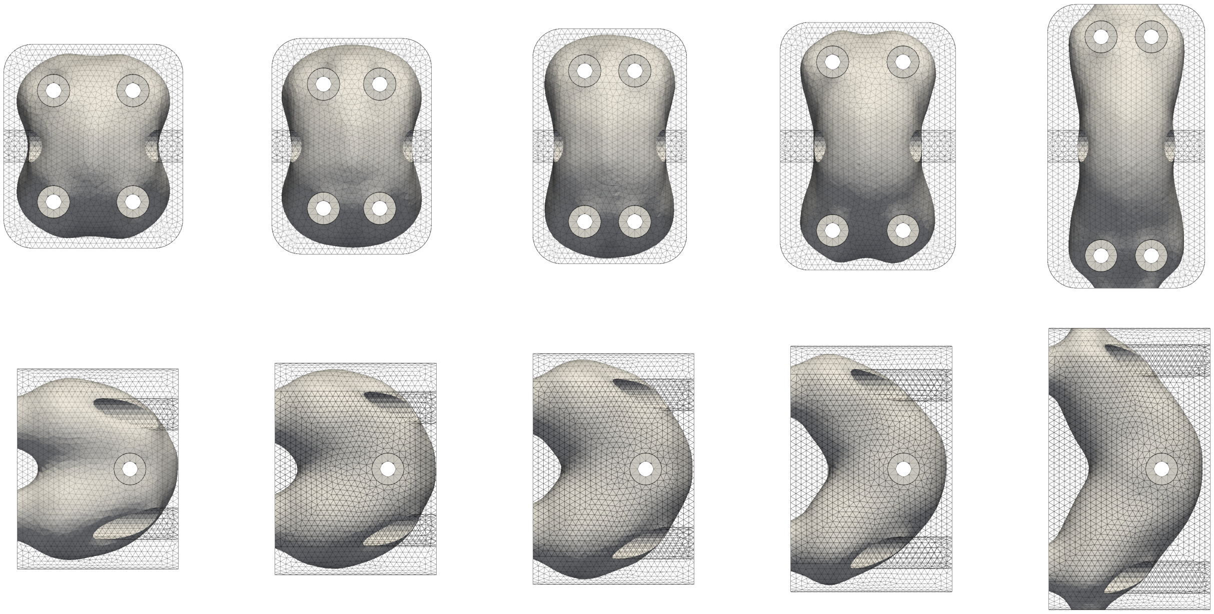

(f) $\alpha = 0.1$ (g) $\alpha = 0.3$ (h) $\alpha = 0.5$ (i) $\alpha = 0.7$ (j) $\alpha = 0.9$

Figure 1-10. Results for concurrent shape and topology optimization for varying priority, α, assigned to load case 1 (load in the Y direction).

where κ_{ij} is the isotropic thermal conduction tensor, and κ is a material constant. The nodal thermal flux, s_I^e is differiated using automatic differentiation, so no closed-form expression for s_{I,T_J}^e is needed. The global forcing vector is given by:

$$\mathscr{Q}(p) = \underset{e \in \Gamma_t^h(p)}{\mathbb{A}} \int_{\Gamma^e} N_I \gamma d\Gamma \tag{1.54}$$

for the surface, Γ_t^h, on which the thermal flux, γ, is applied.

Figures (1-9) and (2-11) show the performance and design, respectively, for varying values of priority, α. Similar effects are observed in this example. The performance in both load cases improves significantly when the design domain is optimized concurrently with the topology. At a prioritization of $\alpha = 0.9$ the mechanical and thermal compliances decrease by 85 percent and 52 percent, respectively, with concurrent shape and topology optimization. As the priority of the thermal performance is increased, the mechanical compliance increased by only 4 percent, while the thermal performance improved by 9 percent. In the initial design domain, an improvement in thermal performance of 14 percent was accompanied by a decrease in mechanical performance of 16 percent. By allowing the design volume to change during optimization, the degree to which the mechanical and thermal performance are in competition was reduced.

2. PROCESS-AWARE DESIGN OPTIMIZATION

Part distortion and residual stress are critical factors for metal additive manufacturing (AM) because they can lead to high failure rates during both manufacturing and service. We present a topology optimization approach that incorporates a fast AM process simulation at each design iteration to provide predictions of manufacturing outcomes (i.e., residual stress, distortion, residual elastic energy) that can be optimized or constrained. The details of the approach and implementation are discussed, and an example design is presented that illustrates the efficacy of the method.

2.1. Introduction

Additive manufacturing, or 3D printing, of metals involves the controlled deposition of molten metal in a point-wise fashion to gradually build the final part from a bulk input usually in powder form [57]. Metal 3D printing is seeing rapid adoption in a range of industries, including aerospace, automotive, and medical [44]. The most commonly used form of metal printing is powder bed fusion (PBF) [36] in which the part is constructed one layer at a time in a bed of metal powder that is selectively melted with a moving point energy source. This process involves repeated heating and cooling as the layers are deposited which results in excessive and highly anisotropic residual stresses in the final part [9] that can lead to fatique failure during service and/or process failure during manufacturing [22]. These issues can often be overcome by trial and error modification of the manufacturing process parameters in order to print the design "as-is". Alternatively, the design process can be modified to include information about the specific manufacturing technique, leading to parts that have optimal or at least desirable manufacturability [61, 34].

Various simulation methods are available for predicting manufacturing process outcomes, i.e., residual stresses, distortion, etc., ranging from high-fidelity, high-cost approaches [24] to coarser, low-cost approaches such as the inherent strain method [40]. The motivation for the present work is to tightly integrate process simulation into a design optimization workflow so that process outcomes can be affected by controlling the part geometry. For this to be practical, the simulations must be fast enough to run in a few minutes and light-weight enough to run on a laptop or desktop computer. The sequential inherent strain method is a reasonable balance between accuracy and speed. In this method, the deposition process is coarsened into a sequential depostion of "super-layers" of material that consist of many actual deposition layers. The deposited material has an initial strain (i.e., an inherent strain) that approximates the thermal strains in the actual process. Mechanical equilibrium between the part and the new layer is computed after each step, and the part is removed from the baseplate after the final layer is placed.

In recent work by Takezawa et al. [59], the deposition process was simulated at each optimization iteration using the sequential inherent strain method. Their formulation uses a general objective that is a function of the displacements at each step in the inherent strain deposition sequence. The sensitivity of this objective to changes in the design was computed using the adjoint variable method, and gradient-based optimization was used to minimize distortion of a predefined part by varying the cellular density within the part domain.

Misium et al. [45] use a similar approach in that a sequential inherent strain simulation is performed at each iteration during the optimization process, however the resulting quantities of interest were included as constraints for a compliance minimization problem rather than quantity to be minimized.

We pursue an approach similar to Takezawa [59] and Misium [45] by directly incorporating sequential inherent strain simulations in the optimization loop. In contrast to those prior works, we incorporate the process predictions as another objective that is assigned a weight relative to the performance objectives, e.g., mechanical compliance. This results in a trade space between manufacturability and performance in which designers can make informed trade-offs. Further, our formulation expresses the process objective as a function of displacements *and internal states*. Herein, we present only results for linear process simulations, but our formulation extends naturally to non-linear cases. While previous authors used measures of the displacement, we use *internal residual energy*, which is the internal elastic energy associated with the deposition process, as an objective. Finally, the formulation has been implemented for both topology and shape optimization.

2.2. Formulation

The design domain, Ω_d, is discretized into Ω_d^h upon which a nodal field of design variables, $\boldsymbol{\rho} \in \mathbb{R}^{n_n}$, is defined, where n_n is the number of nodes in the finite element mesh. The mesh node locations are expressed as a global vector, $\boldsymbol{X} \in \mathbb{R}^{n_x}$, and n_x is the number of mesh nodes times the number of space dimensions, n_D, i.e., $n_x = n_n \times n_D$.

A set of n_l boundary value problems (BVP) are defined on the design domain that represent the load cases and corresponding responses for which the design is to be optimized. These are solved using the finite element method. The discrete form for problem I is expressed as

$$\text{(discrete BVP):}\quad \boldsymbol{g}_I\left(\boldsymbol{\rho}, \boldsymbol{X}, \boldsymbol{S}_I\right) = \boldsymbol{0} \tag{2.1}$$

$$\text{(response):}\quad F_I\left(\boldsymbol{\rho}, \boldsymbol{X}, \boldsymbol{S}_I\left(\boldsymbol{\rho}, \boldsymbol{X}\right)\right), \tag{2.2}$$

The equilibrium state, $\boldsymbol{S}_I$, is implicitly defined by Equation (2.1), and the response, $F_I \in \mathbb{R}$, is a scalar value that is to be minimized.

The statement of the optimization problem is as follows:

$$\min_{\boldsymbol{\rho}} : f(\boldsymbol{\rho},\boldsymbol{X}) = \sum_{I=1}^{n_l} \alpha_I F_I(\boldsymbol{\rho},\boldsymbol{X},\boldsymbol{S}_I(\boldsymbol{\rho},\boldsymbol{X})) \tag{2.3}$$

$$s.t. : \boldsymbol{g}_I(\boldsymbol{\rho},\boldsymbol{X},\boldsymbol{S}_I) = \mathbf{0} \text{ for } I = 1,...,n_l \tag{2.4}$$

$$: h(\boldsymbol{\rho},\boldsymbol{X}) \leq 0 \tag{2.5}$$

where Equation (2.5) ($h : \mathbb{R}^{n_n} \times \mathbb{R}^{n_x} \to \mathbb{R}$) is a constraint on the design, and the objective weights, α_I, sum to one.

The first load case is a sequential inherent strain simulation of the deposition process. The design domain is broken into super-layers using masks, $\boldsymbol{M}_N^k \in \mathbb{B}^{n_n}$ and $\boldsymbol{M}_E^k \in \mathbb{B}^{n_e}$ (n_e is the number of mesh elements, and $\mathbb{B}$ is the boolean domain $\{0,1\}$), that indicate whether a mesh node or element, respectively, is active (1) or inactive (0) at each sequence step, k. The deposition sequence consists of a mechanical equilibration of the nodal forces of all active elements,

$$\boldsymbol{R}^k\left(\boldsymbol{\rho},\boldsymbol{X},\boldsymbol{U}^k,\boldsymbol{c}^{k-1}\right) = \mathop{\mathrm{A}}_{e\in m_E^k} \boldsymbol{f}^{ek} = 0 \tag{2.6}$$

and state update,

$$\boldsymbol{H}^k\left(\boldsymbol{\rho},\boldsymbol{X},\boldsymbol{U}^k,\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right) = \tag{2.7}$$

$$\boldsymbol{c}^k - \boldsymbol{c}^{k-1} - \boldsymbol{M}_e^k\left(\boldsymbol{B}\left(\boldsymbol{U}^k\right)\right) = 0, \tag{2.8}$$

at each process step, $k = 1,...,n_s$. The finite element assembly operation is denoted by $\mathrm{A}_{e\in m}$ where m is the set of elements to be assembled. The set of active elements at step k is m_E^k, and the nodal forces, $\boldsymbol{f}^e = f_{iI}^e$, ($i = 1,...,n_D$ and $I = 1,...,n_n^e$) for element e are a function of the topology, position, and state of the nodes comprised by the element:

$$\boldsymbol{f}^{ek} = f_{iI}^{ek} = \int_{\Omega^e} p(\boldsymbol{\rho}^e) B_{Ij}^e(\boldsymbol{x}^e) \sigma_{ij}^k(\boldsymbol{x}^e,\boldsymbol{u}^{ek},\boldsymbol{c}^{ek-1}) d\Omega \tag{2.9}$$

$$\boldsymbol{\rho}^e = \{\rho_I \mid I = 1,...,n_n^e\} \tag{2.10}$$

$$\boldsymbol{x}^e = \{x_{iI} \mid i = 1,...,n_D,\ I = 1,...,n_n^e\} \tag{2.11}$$

$$\boldsymbol{u}^{ek} = \left\{u_{iI}^k \mid i = 1,...,n_D,\ I = 1,...,n_n^e\right\} \tag{2.12}$$

$$\sigma_{ij}^k = c_{ijkl}\left(\varepsilon_{kl}^k + c_{kl}^{e\ k-1} - \gamma_{kl}\right) \tag{2.13}$$

$$\varepsilon_{kl}^k = \left(B_{Jk}^e u_{lJ}^k + B_{Jl}^e u_{kJ}^k\right)/2 \tag{2.14}$$

$$c_{ijkl} = \lambda \delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{2.15}$$

$$p(\boldsymbol{\rho}^e) = (N_I \rho_I)^3 \tag{2.16}$$

for number of dimensions, n_D, and number of nodes per element, n_n^e. In the preceding, B_{Ii}^e is the gradient matrix for element e, N_I are the basis functions, c_{ijkl} is the fourth rank elastic stiffness tensor, δ_{ij} is the Kronecker delta, and λ and μ are material constants. The inherent strain, $\boldsymbol{\gamma} = \gamma_{jk}$, approximates the residual strain state due to the deposition process. The function, p, is the standard SIMP penalty function [11, 52].

Equation (2.8) is the residual expression for the update of the reference strain, $\boldsymbol{c}^k$, at step k given the reference strain, $\boldsymbol{c}^{k-1}$ at $k-1$ and the displacement increment, $\boldsymbol{U}^k$, and $\boldsymbol{B}(*)$ is a global symmetric gradient operator. The application of the element mask to the global symmetric gradient operator zeros out all rows that are associated with an inactive element.

The BVP constraint includes the equilibration and state update from each step, i.e.,

$$\boldsymbol{g}_1^k = \left\{ \begin{array}{c} \boldsymbol{R}^k \\ \boldsymbol{H}^k \end{array} \right\} \tag{2.17}$$

$$\boldsymbol{g}_1 = \left\{ \boldsymbol{g}_1^k \mid k = 1, ..., n_s \right\} \tag{2.18}$$

and the state consists of the nodal displacement increment and element reference strain at each step:

$$\boldsymbol{S}_1^k = \left\{ \begin{array}{c} \boldsymbol{U}^k \\ \boldsymbol{c}^k \end{array} \right\} \tag{2.19}$$

$$\boldsymbol{S}_1 = \left\{ \boldsymbol{S}_1^k \mid k = 1, ..., n_s \right\} \tag{2.20}$$

The remaining load cases, $I = 2, ..., n_l$, vary by application but will generally consist of scenarios that approximate the service loads for which the part is being optimized.

2.2.1. Forward Problem

The sequential inherent strain simulation of the deposition process proceeds as follows: Given the previous reference state, $\boldsymbol{c}^{k-1}$, the displacement increment, $\boldsymbol{U}^k$, that produces a zero residual is determined using Newton iteration:

$$\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^{k,N}} \Delta \boldsymbol{U}^{k,N} = -\boldsymbol{R}^{k,N} \tag{2.21}$$

$$\boldsymbol{U}^{k,N+1} = \boldsymbol{U}^{k,N} + \Delta \boldsymbol{U}^{k,N} \tag{2.22}$$

$$N \leftarrow N + 1 \tag{2.23}$$

Once the equilibrium displacement is known, the updated reference strain satisfying Equation (2.8) is computed, i.e.:

$$\boldsymbol{c}^k = \boldsymbol{c}^{k-1} + \boldsymbol{B}\left(\boldsymbol{U}^k\right) \tag{2.24}$$

The process for computing the forward solution is shown in Algorithm (1).

Algorithm 1: Forward solution

```
Output: Solution $\boldsymbol{U}^k$, $\boldsymbol{c}^k$ for $k = 1...n$
Initialize $k = 1$, $\boldsymbol{c}^1 = 0$
while $k \leq n$ do
    Set $\boldsymbol{U}^k = 0$
    while $N < N_{max}$ and $\mathtt{mag}(\boldsymbol{R}^k) > R_{tol}$ do
        Solve Equation (2.21) for $\Delta\boldsymbol{U}^{k,N}$
        Update state: $\boldsymbol{U}^{k,N+1} = \boldsymbol{U}^{k,N} + \Delta\boldsymbol{U}^{k,N}$
        $N \leftarrow N + 1$
    end
    Solve Equation (2.24) for $\boldsymbol{c}^N$
    $k \leftarrow k + 1$
end
```

2.2.2. Objective and Gradient

Calculation of the sensitivity follows the approach described by Alberdi [1]. The objective is a function of the displacement increment and reference strain at all steps,

$$\begin{aligned} F = & F\left(\boldsymbol{\rho}, \boldsymbol{X},\right. \\ & \boldsymbol{U}^1\left(\boldsymbol{\rho}, \boldsymbol{X}\right), ..., \boldsymbol{U}^n\left(\boldsymbol{\rho}, \boldsymbol{X}\right), \\ & \left.\boldsymbol{c}^1\left(\boldsymbol{\rho}, \boldsymbol{X}\right), ..., \boldsymbol{c}^n\left(\boldsymbol{\rho}, \boldsymbol{X}\right)\right). \end{aligned} \tag{2.25}$$

The objective function used for this work is the *residual internal energy*:

$$F = \sum_{k=1}^{n_s} \left(\sum_{e \in m_E^k} \int_{\Omega^e} \frac{1}{2} \sigma_{ij}^k \left(\varepsilon_{ij}^k + c_{ij}^{e\,k-1} \right) d\Omega \right) \tag{2.26}$$

The Lagrangian function is:

$$\hat{F} = F + \sum_{k=1}^{n} \boldsymbol{\lambda}^{k^T} \boldsymbol{R}^k + \sum_{k=1}^{n} \boldsymbol{\mu}^{k^T} \boldsymbol{H}^k \tag{2.27}$$

The gradient of the Lagrangian function with respect to the topology field, $\boldsymbol{\rho}$, is found by using the chain rule:

$$
\begin{aligned}
\frac{d\hat{F}}{d\boldsymbol{\rho}} = \frac{\partial F}{\partial \boldsymbol{\rho}} &+ \sum_{k=1}^{n} \left(\frac{\partial F}{\partial \boldsymbol{U}^k} \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\rho}} + \frac{\partial F}{\partial \boldsymbol{c}^k} \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\rho}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\lambda}^{kT} \left(\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\rho}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\rho}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^{k-1}} \frac{\partial \boldsymbol{c}^{k-1}}{\partial \boldsymbol{\rho}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\mu}^{kT} \left(\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\rho}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\rho}} \right. \\
&\left. + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\rho}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^{k-1}} \frac{\partial \boldsymbol{c}^{k-1}}{\partial \boldsymbol{\rho}} \right)
\end{aligned}
\tag{2.28}
$$

Collecting terms that involve gradients of state with respect to the topology field results in the following form:

$$
\begin{aligned}
\frac{d\hat{F}}{d\boldsymbol{\rho}} = \frac{\partial F}{\partial \boldsymbol{\rho}} &+ \sum_{k=1}^{n} \left(\boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\rho}} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\rho}} \right) \\
&+ \left(\frac{\partial F}{\partial \boldsymbol{c}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n} \right) \frac{\partial \boldsymbol{c}^n}{\partial \boldsymbol{\rho}} \\
&+ \sum_{k=1}^{n} \left(\frac{\partial F}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} \right) \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\rho}} \\
&+ \sum_{k=1}^{n-1} \left(\frac{\partial F}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} \right. \\
&\left. + \boldsymbol{\mu}^{k+1^T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{k+1^T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^k} \right) \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\rho}}
\end{aligned}
\tag{2.29}
$$

The dependence on state gradients can be eliminated by choosing values for the Lagrange multipliers that solve the sequence of adjoint equations:

(nth step)

$$\frac{\partial F}{\partial \boldsymbol{c}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n} = 0 \tag{2.30}$$

$$\frac{\partial F}{\partial \boldsymbol{U}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n} = 0 \tag{2.31}$$

(kth step), $k = n-1, ..., 2, 1$

$$\frac{\partial F}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{k+1^T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{k+1^T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^k} = 0 \tag{2.32}$$

$$\frac{\partial F}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} = 0 \tag{2.33}$$

resulting in a total derivative given by:

$$\frac{d\hat{F}}{d\boldsymbol{\rho}} = \frac{\partial F}{\partial \boldsymbol{\rho}} + \sum_{k=1}^{n} \left(\boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\rho}} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\rho}} \right) \tag{2.34}$$

In the following, derivatives are expressed as $\partial a / \partial b = a_{,b}$. The adjoint equations can be simplified somewhat since $\boldsymbol{H}^k_{,c^k} = \boldsymbol{I}$ and $\boldsymbol{H}^k_{,c^{k-1}} = -\boldsymbol{I}$ to:

(nth step)

$$\boldsymbol{\mu}^n = -F_{,\boldsymbol{c}^n} \tag{2.35}$$

$$\boldsymbol{R}^{n\,T}_{,\boldsymbol{U}^n} \boldsymbol{\lambda}^n = -\left(F_{,U^n} + \boldsymbol{H}^{n\,T}_{,\boldsymbol{U}^n} \boldsymbol{\mu}^n \right) \tag{2.36}$$

(kth step), $k = n-1, ..., 2, 1$

$$\boldsymbol{\mu}^k = \boldsymbol{\mu}^{k+1} - F_{,\boldsymbol{c}^k} - \boldsymbol{R}^{k+1\,T}_{,\boldsymbol{c}^k} \boldsymbol{\lambda}^{k+1} \tag{2.37}$$

$$\boldsymbol{R}^{k\,T}_{,\boldsymbol{U}^k} \boldsymbol{\lambda}^k = -\left(F_{,\boldsymbol{U}^k} + \boldsymbol{H}^{k\,T}_{,\boldsymbol{U}^k} \boldsymbol{\mu}^k \right) \tag{2.38}$$

The process for computing the objective gradient is shown in Algorithm (2).

The above derivation was done in terms of the nodal topology field, $\boldsymbol{\rho}$, but the same process applies to gradients with respect to the global vector of nodal positions, $\boldsymbol{X}$. With $dF/d\boldsymbol{X}$ computed, shape sensitivities can be found by

$$\frac{dF}{d\boldsymbol{p}} = \frac{dF}{d\boldsymbol{X}} \frac{d\boldsymbol{X}}{d\boldsymbol{p}} \tag{2.39}$$

where $\boldsymbol{p} \in \mathbb{R}^{n_p}$ is the set of n_p shape parameters that define the part geometry. The mesh sensitivities, $d\boldsymbol{X}/d\boldsymbol{p}$, are computed using the Engineering Sketch Pad [21, 28] which is built on top of OpenCSM [29], EGADS [29], and OpenCASCADE [48]. Further detail can be found in [51].

2.3. Implementation

This approach has been implemented in the Plato software (github.com/platoenine) developed at Sandia National Laboratories. Plato is designed to simplify the implementation of physics (PDE constraints) and criteria (objectives and constraints) for gradient-based shape and topology optimization. This is accomplished by defining templated abstractions for physics residual functions (e.g., $\boldsymbol{R}$ and $\boldsymbol{H}$) and criterion functions (e.g., F and $\boldsymbol{h}$) that permit the use of automatic

Algorithm 2: Objective gradient

```
Input: Solution $\boldsymbol{U}^k$, $\boldsymbol{c}^k$ for $k = 1...n$
Output: Adjoint solution $\boldsymbol{\lambda}^k$, $\boldsymbol{\mu}^k$ for $k = 1...n$, and objective gradient, $dF/d\boldsymbol{\rho}$
Compute $\boldsymbol{\mu}^n$
Compute ${\boldsymbol{R}^n_{,\boldsymbol{U}^n}}^T$, $F_{,U^n}$, ${\boldsymbol{H}^n_{,\boldsymbol{U}^n}}^T$
Solve Equation (2.36) for $\boldsymbol{\lambda}^n$
Initialize $k = n - 1$
while $k > 0$ do
    Compute $\boldsymbol{\mu}^k$
    Compute ${\boldsymbol{R}^k_{,\boldsymbol{U}^k}}^T$, $F_{,U^k}$, ${\boldsymbol{H}^k_{,\boldsymbol{U}^k}}^T$
    Solve Equation (2.38) for $\boldsymbol{\lambda}^k$
    $k \leftarrow k - 1$
end
Compute $F_{,\rho}$, $\boldsymbol{R}^k_{,\rho}$, and $\boldsymbol{H}^k_{,\rho}$
Compute $dF/d\boldsymbol{\rho}$
```

differentiation (AD) [49] for computing all relevant partial derivatives, i.e.,

$$
\begin{aligned}
&\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\rho}}, \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{X}}, \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k}, \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^{k-1}}, \\
&\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\rho}}, \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{X}}, \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k}, \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{c}^k}, \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^{k-1}}, \\
&\frac{\partial F}{\partial \boldsymbol{\rho}}, \frac{\partial F}{\partial \boldsymbol{X}}, \frac{\partial F}{\partial \boldsymbol{U}^k}, \frac{\partial F}{\partial \boldsymbol{c}^k}, \frac{\partial F}{\partial \boldsymbol{U}^k}, \\
&\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\rho}}, \text{ and } \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{X}}.
\end{aligned}
$$

Developers are able to implement functions for residuals and criteria, and derivatives are computed automatically. The procedure described in Section (2.2) is implemented based on these abstractions.

Verification The implementation was verified by comparing computed results against analytical solutions.

2.3.1. Normal Strain Problem

Figure (2-1) shows a comparison for a basic verification problem in which a single layer (thickness $l = 0.125$ m) is deposited onto a substrate (thickness $l = 0.125$ m). The deposited layer has an inherent strain in the build direction (z) only of $\gamma_{zz} = 0.01$. The bottom and top surfaces are

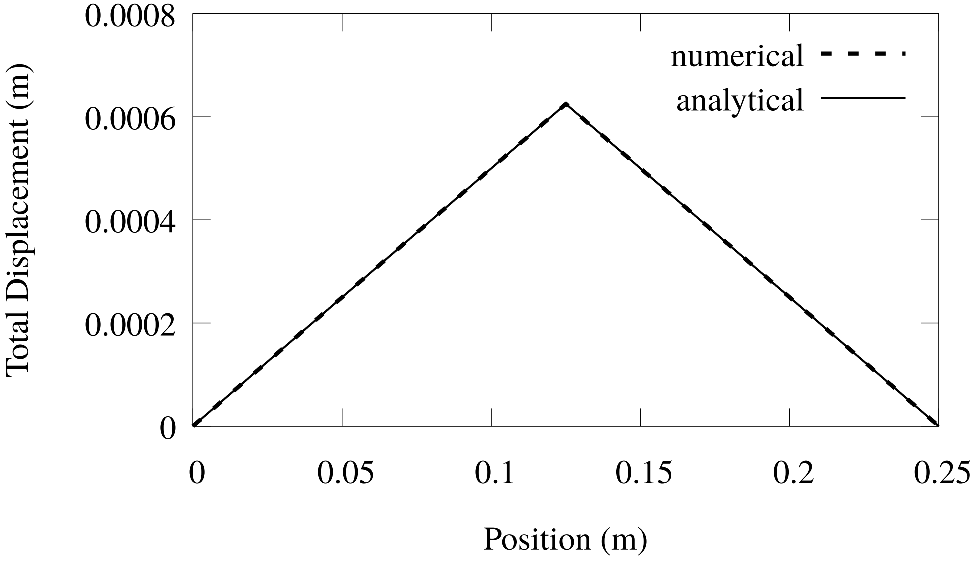


Figure 2-1. Comparison between numerical and analytical solutions for a simplified two-layer deposition problem.

fully constrained, and Poisson's ratio is zero, so the problem is effectively one-dimensional. The displacement of the interface between the substrate and deposited layer is given by

$$u_c = \frac{l^2}{2l}, \tag{2.40}$$

and the displacement as a function of the distance from the bottom surface is:

$$u(z) = \frac{u_c}{l} \begin{cases} z & 0 < z < l \\ (2l - z) & l \leq z < 2l \end{cases} \tag{2.41}$$

The analytical solution has an L2 norm error of 5.77×10^{-10}.

2.3.2. *Shear Strain Problem*

The second verification problem involves elastic bilayer bending: A single layer (thickness $l = 0.05$ m) is deposited onto a substrate (thickness $l = 0.05$ m). The deposited layer has an inherent strain in the build plane (x-y) of $\gamma_{xx} = -0.01$. The negative x face is fully constrained, and Poisson's ratio is zero, so the problem is effectively one-dimensional. The curvature of the interface between the substrate and deposited layer is given by [62]:

$$\kappa = \frac{6(\varepsilon_2 - \varepsilon_1)(1+m)^2}{t\,(3(1+m)^2 + (1+mn)(m^2 + 1/(mn))} \tag{2.42}$$

where $m = 1$ is the ratio of the top layer thickness to bottom layer thickness, $n = 1$ is the ratio of top layer stiffness to bottom layer stiffness, and $t = 0.1$ is the total bilayer thickness. $\varepsilon_1 = \gamma_{xx}$ and

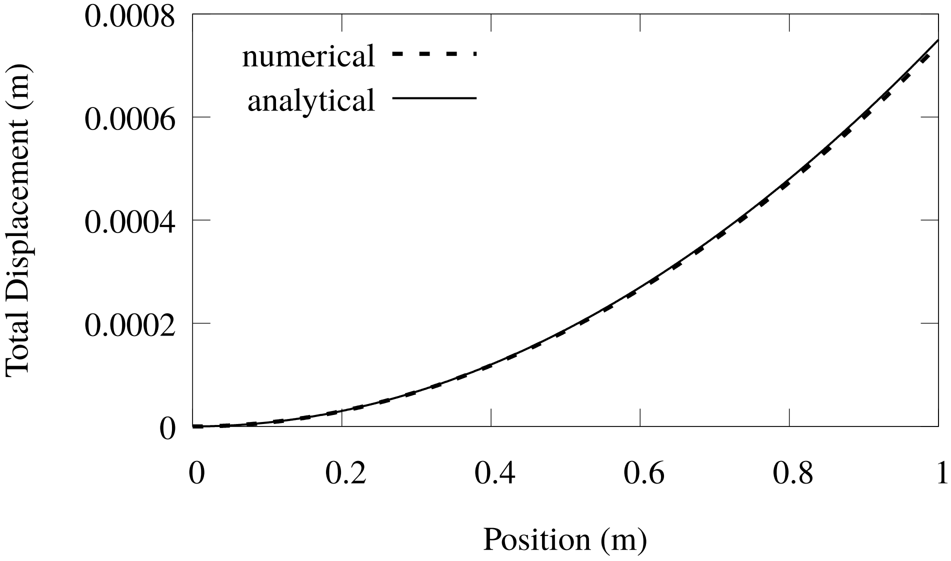


Figure 2-2. Comparison between numerical and analytical solutions for a simplified two-layer deposition problem with lateral inherent strains.

$\varepsilon_2 = 0.0$ are the in-plane strains of the top and bottom layers, respectively. The displacement as a function of the distance from the negative x surface is:

$$u(x) = \frac{1}{\kappa}(1 - \cos(x\kappa)) \tag{2.43}$$

A comparison between numerical and analytical results is shown in Figure (2-2), and the L2 norm error is 1.4×10^{-4}.

2.3.3. *Four Layer Problem*

The third and final verification problem consists of multiple deposition layers: Three layers (thickness $l = 0.25$ m) are successively deposited onto a substrate (thickness $l = 0.25$ m). The deposited layers have an inherent strain in the build direction (z) only of $\gamma_{zz} = 0.01$. The bottom of the substrate and top of the final layer are fully constrained, and Poisson's ratio is zero, so the problem is effectively one-dimensional. The displacement along the build direction is:

$$u(z) = \begin{cases} -l\gamma_{zz}z & 0 < z \leq l \\ \gamma_{zz}((1-l)z - l) & l < z \leq 2l \\ \gamma_{zz}((1-l)z - 2l) & 2l < z \leq 3l \\ \gamma_{zz}((1-l)z - 3l) & 3l < z \leq 4l \end{cases} \tag{2.44}$$

The comparison between numerical and analytical results (see Figure (2-3)) shows an L2 norm error of 7.76×10^{-3}.

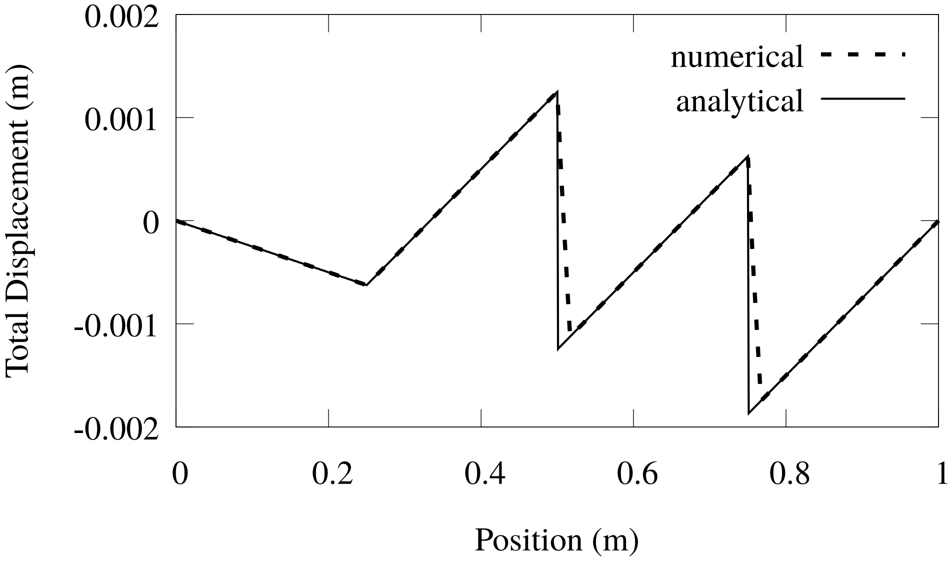


Figure 2-3. Comparison between numerical and analytical solutions for a simplified two-layer deposition problem.

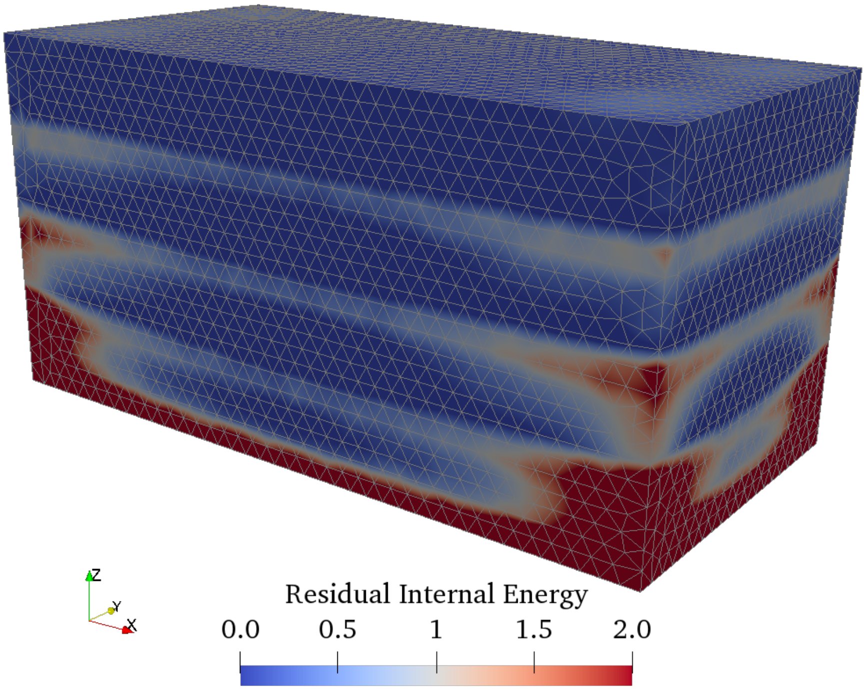


Figure 2-4. Paint plot of the residual internal energy for the derivative verification problem. The bottom surface is fully constrained and the deposition consists of four steps (layers).

Table 2-1. Error in the topology gradient versus finite difference step size, ε.

ε	dF_{AD} $(\times 10^{-3})$	dF_{FD} $(\times 10^{-3})$	error
10^{-1}	1.03614525	1.03614526	5.2×10^{-12}
10^{-2}	1.03614525	1.03614532	6.6×10^{-11}
10^{-3}	1.03614525	1.03614492	3.3×10^{-10}
10^{-4}	1.03614525	1.03614160	3.7×10^{-9}
10^{-5}	1.03614525	1.03566338	4.6×10^{-9}

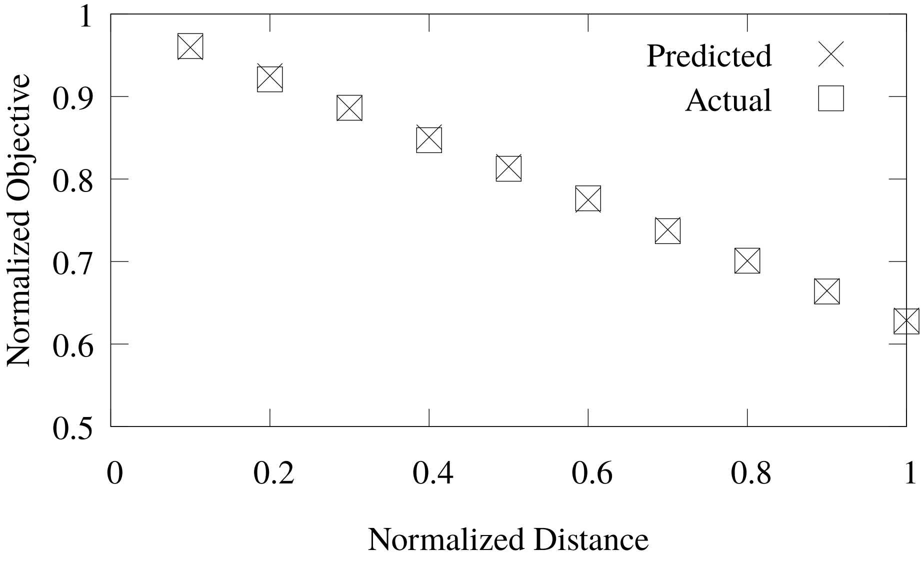


Figure 2-5. Comparison between predicted and actual normalized objective values.

2.3.4. Derivative Verification

The derivative of the residual internal energy (see Figure (2-4)) with respect to the topology was verified using a five-point finite differencing stencil

$$dF_{FD} = \left(-F\left(\boldsymbol{\rho}_o + 2\varepsilon\boldsymbol{\delta}\right) + 8F\left(\boldsymbol{\rho}_o + \varepsilon\boldsymbol{\delta}\right)\right. \tag{2.45}$$

$$\left. -8F\left(\boldsymbol{\rho}_o - \varepsilon\boldsymbol{\delta}\right) + F\left(\boldsymbol{\rho}_o - 2\varepsilon\boldsymbol{\delta}\right)\right)/12\varepsilon \tag{2.46}$$

$$dF_{AD} = \frac{dF}{d\boldsymbol{\rho}}\boldsymbol{\delta} \tag{2.47}$$

$$\text{error} = |dF_{AD} - dF_{FD}| \tag{2.48}$$

where $\boldsymbol{\delta}$ is a randomly generated step in the range $(-\delta, \delta)$. The error versus step size, ε, in direction $\boldsymbol{\delta}$ is shown in Table (2-1)

Derivatives with respect to shape parameters are verified on the part shown in Figure (2-6). The

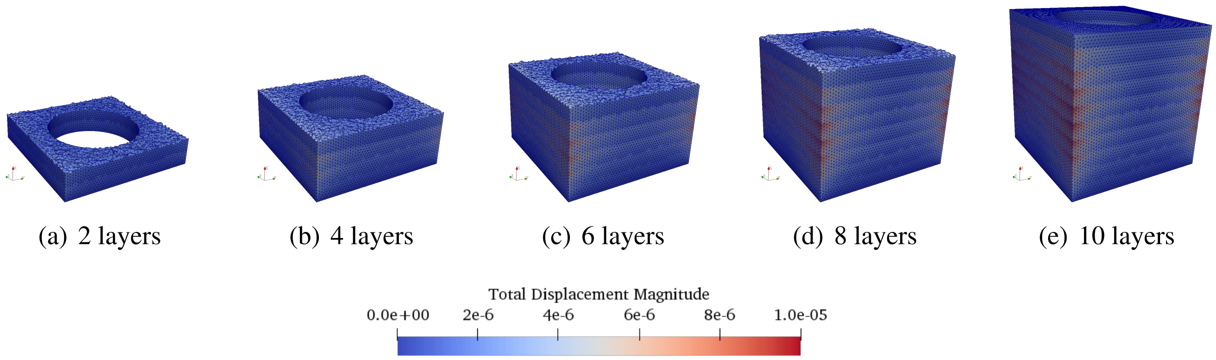

(a) 2 layers (b) 4 layers (c) 6 layers (d) 8 layers (e) 10 layers

Figure 2-6. Various deposition steps in the shape derivative verification problem.

predicted change, dF_P, is computed as the gradient multiplied by the step, $\boldsymbol{s}$,

$$\boldsymbol{d} = \left.\frac{dF}{d\boldsymbol{p}}\right|_{\boldsymbol{p}=\boldsymbol{p}^j} \tag{2.49}$$

$$\boldsymbol{s} = s\frac{\boldsymbol{d}}{|\boldsymbol{d}|} \tag{2.50}$$

$$dF_P^{j+1} = \boldsymbol{d}^T\boldsymbol{s} \tag{2.51}$$

for step size $s = 0.01$, and the actual change, dF_A, is computed by updating the shape parameters with the step and evaluating the objective at the new values.

$$\boldsymbol{p}^{j+1} = \boldsymbol{p}^j + \boldsymbol{s} \tag{2.52}$$

$$dF_A^{j+1} = F(\boldsymbol{p}^{j+1}) - F(\boldsymbol{p}^j) \tag{2.53}$$

$$j \leftarrow j+1 \tag{2.54}$$

The predicted versus realized objective values are shown in Figure (2-5) and the part geometry for various steps in the deposition sequence is shown in Figure (2-6). At each step in the verification sequence, the mesh is updated to reflect the new shape parameters (see Figure (2-6)). The average relative error

$$\text{error} = \frac{1}{10}\sum_{j=1}^{10}\left(2\left|\frac{dF_P^j - dF_A^j}{dF_P^j + dF_A^j}\right|\right) \tag{2.55}$$

in the predicted change in objective is 5.5×10^{-2}.

2.4. Results

The method described above is demonstrated on the problem shown in Figure (2-8). Two load cases are included: i) a process simulation that introduces manufacturability into the optimization,

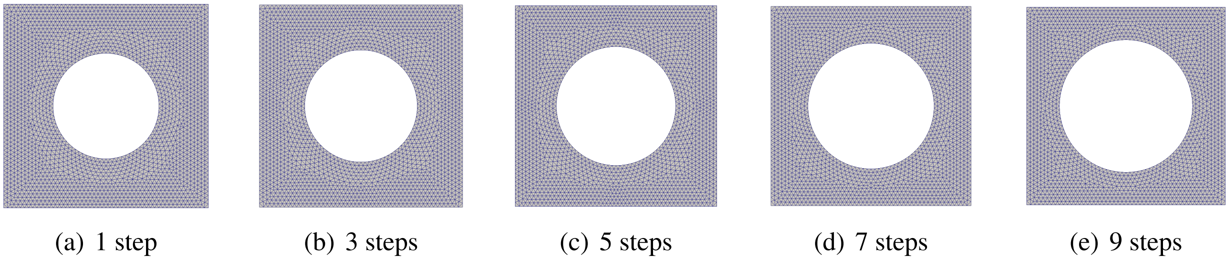


Figure 2-7. Part cross-section at various steps in the shape derivative verification problem.

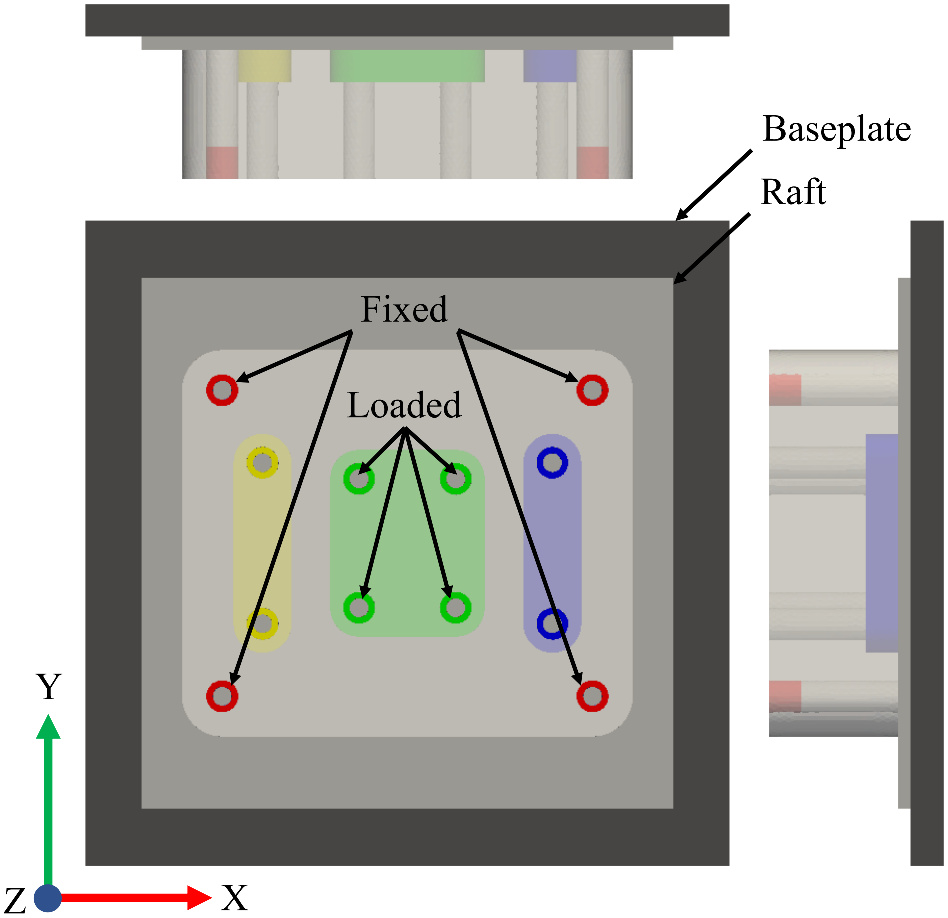


Figure 2-8. Layout of the process-aware topology optimization problem.

Table 2-2. Trade-off between normalized mechanical compliance and normalized residual internal energy for varying weighting, α.

α	Compliance	Change (%)	Residual Energy	Change (%)	Volume (%)
0.01	0.274	0.0	1.0	0.0	100
0.1	0.274	0.06	0.959	-4.09	100
0.2	0.279	1.93	0.895	-10.5	100
0.3	0.284	3.49	0.841	-15.9	100
0.4	0.299	9.15	0.755	-24.5	100
0.5	0.400	46.1	0.562	-43.8	83.9
0.6	0.499	82.2	0.502	-49.8	78.4
0.7	0.691	152	0.389	-61.1	65.3
0.8	1.0	265	0.342	-65.8	57.4

and ii) a mechanical load case that represents the performance of the part in service. The multi-objective is

$$f(\boldsymbol{\rho},\boldsymbol{X}) = \alpha F_0(\boldsymbol{\rho},\boldsymbol{X},\boldsymbol{S}_0) \tag{2.56}$$

$$+(1-\alpha)F_1(\boldsymbol{\rho},\boldsymbol{X},\boldsymbol{U}_1), \tag{2.57}$$

and the parameter, α, determines the weight between the manufacturability and performance of the part. The colored regions in the design volume are not optimized, i.e., these regions have solid density throughout the optimization.

In the manufacturability load case, the $-Z$ face of the baseplate is fully constrained, and the part is deposited in ten super-layers on a "raft" of low density support structure. After all layers are deposited, the part is removed from the raft/baseplate. The objective is the internal residual energy defined in Equation (2.26).

In the performance load case, the four bolt holes shown in red are fixed at zero displacement in X, Y, and Z, and traction is applied in the Z direction on the four bolt holes shown in green. The objective in this case is the mechanical compliance.

The optimization includes a sequential inherent strain simulation of the manufacturing process at each design iteration. Figure (2-10) shows the design for $\alpha = 0.5$ at a few of the deposition steps and at the final step after removal from the raft/baseplate. Each process simulation takes approximately two minutes on an Nvidia Quadro RTX 5000 mobile GPU for a total optimization run time of 75 minutes (25 iterations).

The trade-off between mechanical compliance (performance) and residual internal energy (manufacturability) is shown in Figure (2-9) and Table (2-2). As the manufacturability priority increases from $\alpha = 0.0$ to $\alpha = 0.4$, a decrease in residual internal energy of 24.5 percent is achieved and accompanied by a 9.15 percent increase in mechanical compliance. At higher levels of α, the volume of the final part begins to decrease below the volume inequality constraint, and the two lateral non-optimized regions become detached from the part.

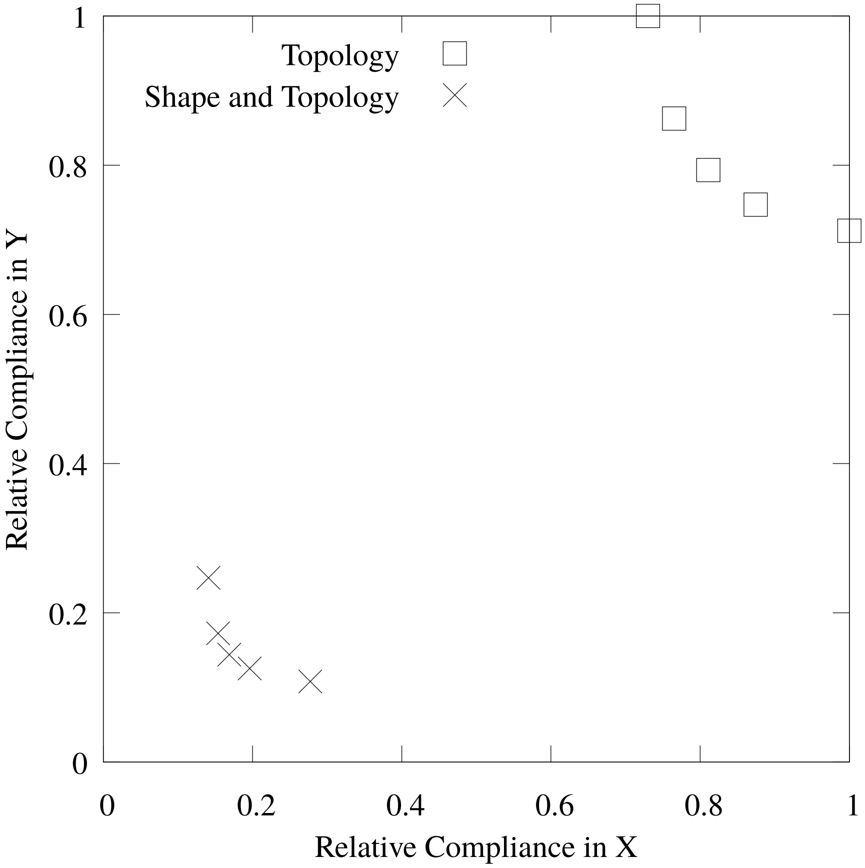
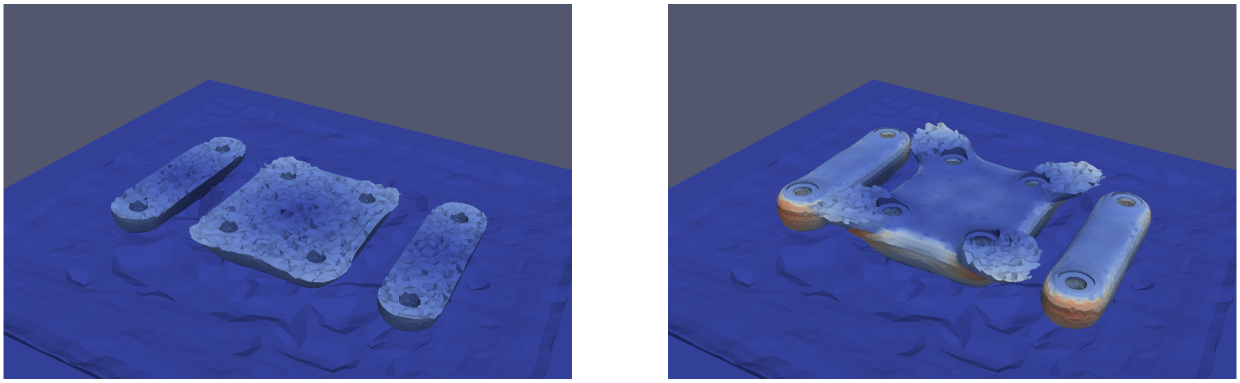
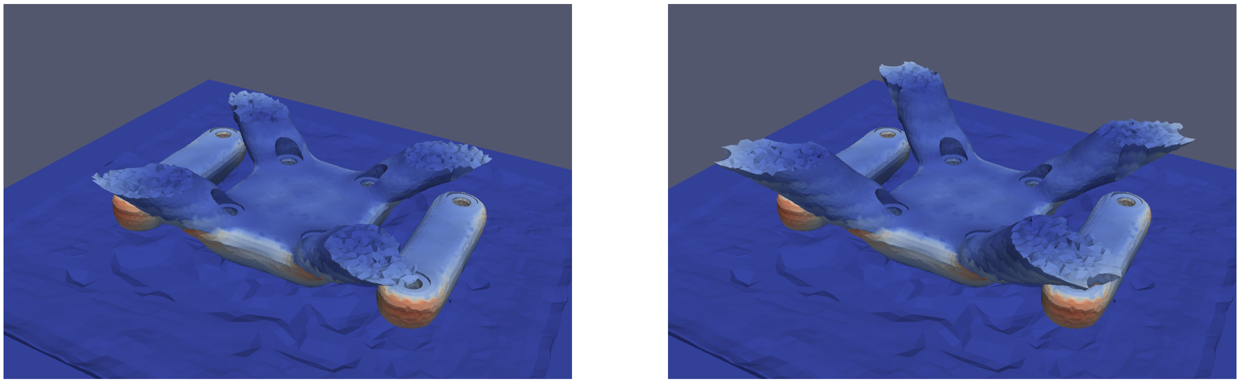


Figure 2-9. Trade-off between mechanical compliance and residual internal energy for various priority, α.



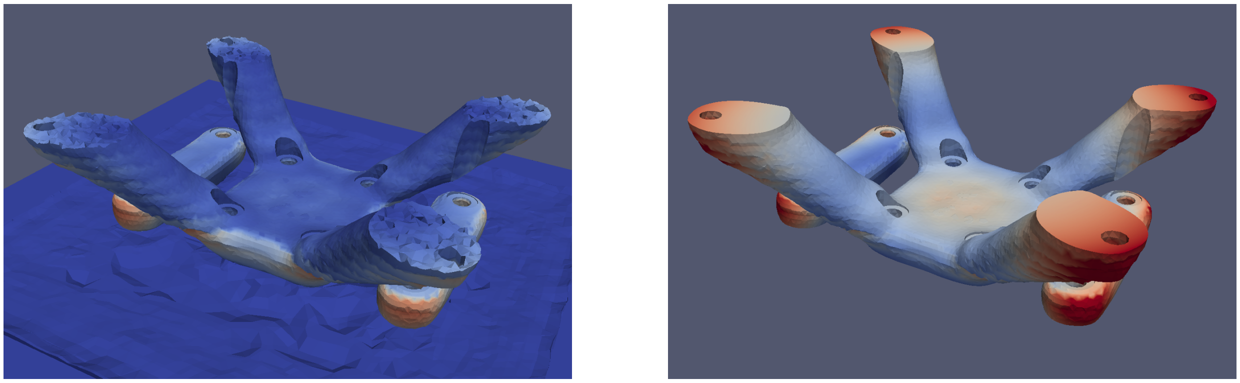
(a) 1 layer

(b) 3 layers



(c) 5 layers

(d) 7 layers



(e) 9 layers

(f) Removed

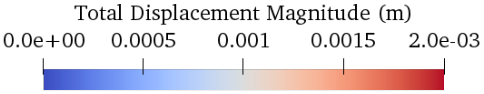
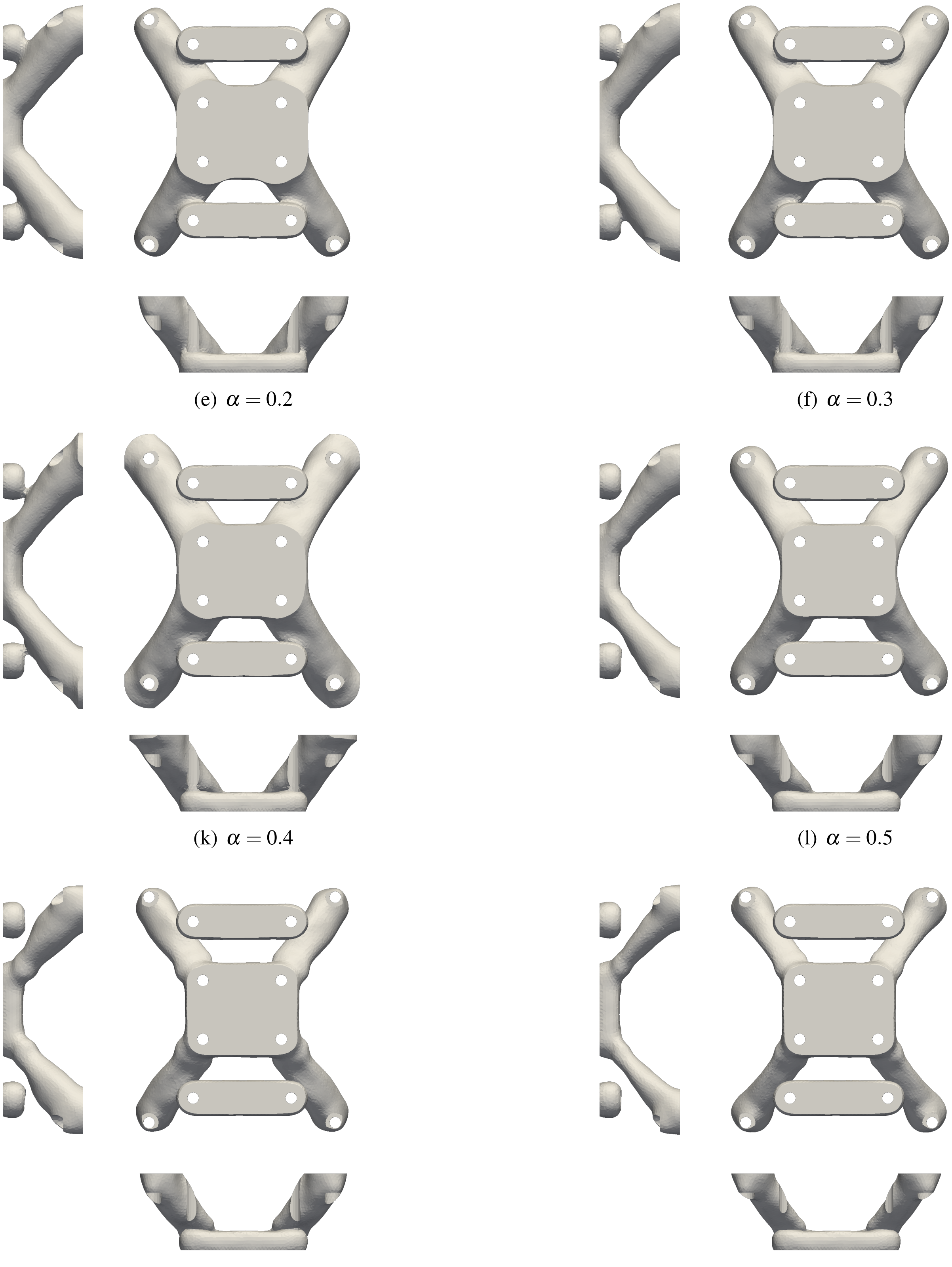


Figure 2-10. Various stages of the sequential inherent strain simulation of the deposition process. Run time is approximately two minutes on an Nvidia Quadro RTX 5000 mobile GPU. The mesh consists of 1.1×10^6 elements, and the simulation includes 10 deposition super-layers and an equilibration step after removal of the part from the baseplate.



(e) $\alpha = 0.2$

(f) $\alpha = 0.3$

(k) $\alpha = 0.4$

(l) $\alpha = 0.5$

(q) $\alpha = 0.6$

(r) $\alpha = 0.7$

Figure 2-11. Results for varying priority, α, between manufacturability and performance.

3. FORMULATION AND IMPLEMENTATION

3.1. Fast heat equation

Consider the heat equation [31] in domain Ω with fixed temperature on surface Γ^T and thermal flux on surface Γ^f, $\partial\Omega = \Gamma^T \cup \Gamma^f$, $\Gamma^T \cap \Gamma^f = \emptyset$:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot k \nabla T = \dot{q}_v \;\text{ in } \Omega \in R^{n_d} \tag{3.1}$$

$$T \equiv T(\mathbf{x},t) \;:\; \text{temperature} \tag{3.2}$$

$$\dot{q}_v \equiv \dot{q}_v(\mathbf{x},t) \;:\; \text{volumetric heat source} \tag{3.3}$$

with boundary conditions:

$$T(\mathbf{x},t) = \tilde{T}(t) \;\text{ on } \Gamma^T \;\forall\, t \in \left(t_o, t_f\right) \tag{3.4}$$

$$-k\nabla T \cdot \hat{\mathbf{n}} = \tilde{f}(t) \;\text{ on } \Gamma^f \;\forall\, t \in \left(t_o, t_f\right) \tag{3.5}$$

$$T(\mathbf{x},0) = T_o(\mathbf{x}) \;\;\forall\, \mathbf{x} \in \Omega \tag{3.6}$$

where $\tilde{T}(t)$, $\tilde{f}(t)$, and $T_o(\mathbf{x})$ are given, ρ, C_p, and k are material parameters, and $\hat{\mathbf{n}}$ is the outward normal to Γ^f.

3.1.1. Weak Statement

The weak statement of the heat equation is

$$\int_\Omega \delta\phi \left(\rho C_p \dot{T} - \nabla \cdot k \nabla T - \dot{q}_v\right) d\Omega = 0 \;\;\forall \delta\phi \tag{3.7}$$

Expanding terms yields:

$$\int_\Omega \delta\phi \;\rho C_p \dot{T} d\Omega - \int_\Omega \delta\phi \;\nabla \cdot k \nabla T d\Omega - \int_\Omega \delta\phi \;\dot{q}_v d\Omega = 0. \tag{3.8}$$

Integration by parts,

(product rule)

$$\nabla \cdot (\delta\phi\, k \nabla T) = \nabla \delta\phi \cdot k \nabla T + \delta\phi \nabla \cdot k \nabla T \tag{3.9}$$

(integrated over domain)

$$-\int_\Omega \nabla \cdot (\delta\phi\, k \nabla T)\, d\Omega + \int_\Omega \nabla \delta\phi \cdot k \nabla T \, d\Omega = -\int_\Omega \delta\phi \nabla \cdot k \nabla T \, d\Omega \tag{3.10}$$

Divergence theorem:

$$\int_{\Omega} \nabla \cdot (\delta\phi k \nabla T)\, d\Omega = \int_{\partial\Omega} \delta\phi\, k \nabla T \cdot \hat{\mathbf{n}}\, d\Gamma \tag{3.11}$$

Combine Equation (3.11) and Equation (3.10),

$$-\int_{\partial\Omega} \delta\phi\, k \nabla T \cdot \hat{\mathbf{n}}\, d\Gamma + \int_{\Omega} \nabla\delta\phi \cdot k \nabla T\, d\Omega = -\int_{\Omega} \delta\phi\, \nabla \cdot k \nabla T\, d\Omega, \tag{3.12}$$

and substitute Equation (3.12) into Equation (3.8):

$$\int_{\Omega} \delta\phi\, \rho C_p \dot{T}\, d\Omega + \int_{\Omega} \nabla\delta\phi \cdot k \nabla T\, d\Omega - \int_{\partial\Gamma} \delta\phi\, k \nabla T \cdot \hat{\mathbf{n}}\, d\Gamma - \int_{\Omega} \delta\phi\, \dot{q}_v\, d\Omega = 0 \tag{3.13}$$

Substitute the flux boundary condition into Equation (3.13) to arrive at the weak form:

$$\int_{\Omega} \delta\phi\, \rho C_p \dot{T}\, d\Omega + \int_{\Omega} \nabla\delta\phi \cdot k \nabla T\, d\Omega + \int_{\partial\Omega} \delta\phi \tilde{f}(t)\, d\Gamma - \int_{\Omega} \delta\phi\, \dot{q}_v\, d\Omega = 0 \tag{3.14}$$

3.1.2. Semi-discrete form

The weak form in Equation (3.14) is then discretized in space to create the semi-discrete form which is a system of ordinary differential equations that can be integrated in time. The spatial disretization starts with a set of nodal basis functions that create a partition of unity over the domain:

$$\sum_{\alpha \in I_n} N_\alpha(\mathbf{x}) = 1 \;\; \forall\, \mathbf{x} \in \Omega \tag{3.15}$$

where I_n is the set of node indices in the finite element mesh. The basis functions are used to approximate relevant fields in physical coordinates in terms of discrete nodal values. For example, the temperature field can be expressed in terms of the nodal temperature values, $T_\alpha(t)$:

$$T(\mathbf{x},t) \approx \sum_{\alpha \in I_n} N_\alpha(\mathbf{x}) T_\alpha(t), \tag{3.16}$$

The nodal basis functions are more conveniently represented in a coordinate system, $\boldsymbol{\xi}$, that is local to each element:

$$T(\boldsymbol{\xi},t) \approx \sum_{\alpha \in I_e} N_\alpha(\boldsymbol{\xi}) T_\alpha(t) \;\; \boldsymbol{\xi} \in \Omega_e \tag{3.17}$$

where I_e is the set of indices of connected nodes and Ω_e is the domain of element e. Similarly for other fields:

$$\delta\phi(\boldsymbol{\xi}) \approx \sum_{\alpha \in I_e} N_\alpha(\boldsymbol{\xi}) \delta\phi_\alpha(t) \tag{3.18}$$

$$\boldsymbol{x}(\boldsymbol{\xi}) \approx \sum_{\alpha \in I_e} N_\alpha(\boldsymbol{\xi}) \boldsymbol{x}_\alpha(t) \tag{3.19}$$

Substitute the discrete approximations into Equation (3.14) and change the integration variable from the global coordinates to the local element coordinates. The integral over the entire domain, Ω, is conducted over the individual element domains, Ω_e, and summed:

(first term)

$$\int_{\Omega} \delta\phi\, \rho C_p \dot{T}\, d\Omega = \sum_{e\in I_e} \delta\phi_I \int_{\Omega_e} N_I(\boldsymbol{\xi})\, \rho C_p N_J(\boldsymbol{\xi})\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Omega_e\, \dot{T}_J(t) \tag{3.20}$$

(second term)

$$\int_{\Omega} \nabla\delta\phi \cdot k\nabla T\, d\Omega = \sum_{e\in I_e} \delta\phi_I \int_{\Omega_e} B_{iI}\, k_{ij}\, B_{jJ}\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Omega_e\, T_J(t) \tag{3.21}$$

(third term)

$$\int_{\partial\Omega} \delta\phi \tilde{f}(t)\, d\Gamma = \sum_{e\in I_e} \delta\phi_I \int_{\partial\Omega} N_I(\boldsymbol{\xi})\, \tilde{f}(t)\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Gamma \tag{3.22}$$

(fourth term)

$$\int_{\Omega} \delta\phi\, \dot{q}_v\, d\Omega = \sum_{e\in I_e} \delta\phi_I \int_{\partial\Omega} N_I(\boldsymbol{\xi})\, \dot{q}_v\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Omega \tag{3.23}$$

where det$J_{\boldsymbol{x\xi}}$ is the determinant of the Jacobian of the mapping (Equation (3.19)) from local to physical coordinates. Note that the discrete nodal values are not a function of position, so they have been taken out of the integrals. The individual element contributions

$$m^e_{IJ} = \int_{\Omega_e} N_I(\boldsymbol{\xi})\, \rho C_p N_J(\boldsymbol{\xi})\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Omega_{\xi} \tag{3.24}$$

$$k^e_{IJ} = \int_{\Omega_e} B_{iI}(\boldsymbol{\xi})\, k_{ij}\, B_{jJ}(\boldsymbol{\xi})\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Omega_{\xi} \tag{3.25}$$

$$f^e_J = \int_{\partial\Omega} N_I(\boldsymbol{\xi})\, \tilde{f}(t)\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Gamma_{\xi} \tag{3.26}$$

$$g^e_J = \int_{\partial\Omega} N_I(\boldsymbol{\xi})\, \dot{q}_v\, \mathrm{det}J_{\boldsymbol{x\xi}}\, d\Gamma_{\xi} \tag{3.27}$$

are assembled into the global system,

$$\boldsymbol{M} = \mathop{\Large\mathsf{A}}_{e} \boldsymbol{m}_e, \text{ etc.}, \tag{3.28}$$

resulting in a global system of coupled ordinary differential equations that can be integrated in time, t:

$$\boldsymbol{M}\dot{\boldsymbol{T}}(t) + \boldsymbol{K}\boldsymbol{T}(t) + \boldsymbol{F} - \boldsymbol{Q} = 0. \tag{3.29}$$

The size of the system in Equation (3.29) is equal to the number of nodes in the finite element mesh, and integrating the problem forward in time will generally be computationally expensive. In the next section, an approach based on [46] is described that leverages anticipated characteristics of the solutions to substantially reduce the problem size and associated computational cost.

3.1.3. Semi-discrete form using super elements

The temperature profile of 3D printed parts during printing will be approximated as spatially one-dimensional in the build direction. To form a coarse discrete approximation from a higher resolution mesh, partion the domain, Ω, into non-overlapping sub-domains, Ω_i $i \in I_{se}$, such that

$$\bigcup_{i \in I_{se}} \Omega_i = \Omega \tag{3.30}$$

and $\partial\Omega_j$ is the boundary between domains $j-1$ and j, and I_{se} are the set of indices of the sub-domains. Define a corresponding set of global basis functions:

$$\begin{aligned} \mathcal{N} = \{\mathcal{N}_i \mid \nabla \cdot \nabla \mathcal{N}_i &= 0 \; \forall \, \boldsymbol{x} \in \Omega_{i-1} \cup \Omega_i, & (3.31)\\ \mathcal{N}_i(\boldsymbol{x}) &= 0 \; \forall \, \boldsymbol{x} \in \partial\Omega_{i+1}, & (3.32)\\ \mathcal{N}_i(\boldsymbol{x}) &= 1 \; \forall \, \boldsymbol{x} \in \partial\Omega_{i}, & (3.33)\\ \mathcal{N}_i(\boldsymbol{x}) &= 0 \; \forall \, \boldsymbol{x} \in \partial\Omega_{i-1}, \; i \in I_{sn} \, \} & (3.34) \end{aligned}$$

that forms a partition of unity:

$$\sum_{\alpha \in I_{sn}} \mathcal{N}_\alpha(\boldsymbol{x}) = 1 \; \forall \, \boldsymbol{x} \in \Omega \tag{3.35}$$

where I_{sn} is the set of indices of the super nodes. In global coordinates the temperature field is approximated in terms of the discrete temperatures, $T_\alpha(t)$ $\alpha \in I_{sn}$, at the super nodes:

$$T(\boldsymbol{x},t) = \sum_{\alpha \in I_{sn}} \mathcal{N}_\alpha(\boldsymbol{x}) T_\alpha(t) \tag{3.36}$$

In super element *se*:

$$T(\boldsymbol{x},t) = \sum_{\alpha \in I_{se}} \mathcal{N}_\alpha(\boldsymbol{x}) T_\alpha(t) \tag{3.37}$$

where I_{se} is the set of indices of super nodes comprised by *se*. Evaluate the first integral in Equation (3.14) over the super elements:

$$\int_{\Omega_{se}} \delta\phi \, \rho C_p \, \dot{T} \, d\Omega = \int_{\Omega_{se}} \mathcal{N}_\alpha(\boldsymbol{x}) \, \delta\phi_\alpha \, \rho C_p \, \mathcal{N}_\beta(\boldsymbol{x}) \, \dot{T}_\beta \, d\Omega \tag{3.38}$$

to determine the super element's capacitance matrix:

$$m^{se}_{\alpha\beta} = \int_{\Omega_{se}} \mathcal{N}_\alpha(\boldsymbol{x}) \, \rho C_p \, \mathcal{N}_\beta(\boldsymbol{x}) \, d\Omega \tag{3.39}$$

The integral in Equation (3.39) is more conveniently computed on the sub-elements that compose the super-element:

$$m^{se}_{\alpha\beta} = \sum_{e \in I_{se}} \int_{\Omega_{se} \cap \Omega_e} \mathcal{N}_\alpha(\boldsymbol{x}(\boldsymbol{\xi})) \, \rho C_p \, \mathcal{N}_\beta(\boldsymbol{x}(\boldsymbol{\xi})) \, \det \boldsymbol{J}_{\boldsymbol{x}\boldsymbol{\xi}} \, d\Omega_\xi \tag{3.40}$$

Quadrature is used to integrate Equation (3.40):

$$m^{se}_{\alpha\beta} \approx \sum_{e\in I_{se}} \sum_{i\in g_e} w^e_i \mathcal{N}_\alpha(\boldsymbol{x}(\boldsymbol{\xi}^e_i))\, \rho C_p\, \mathcal{N}_\beta(\boldsymbol{x}(\boldsymbol{\xi}^e_i))\, \det J_{\boldsymbol{x}\boldsymbol{\xi}} \tag{3.41}$$

where w^e_i and $\boldsymbol{\xi}^e_i$ are the quadrature weights and points for element e. Note that if the boundaries of the super elements conform to the high resolution mesh then no special considerations are required to evaluate the integral in Equation (3.39) since there is a set of elements, J_{se}, that cover the domain of the super element: $\cup_{e\in J_{se}} \Omega_e = \Omega_{se}$

The global basis is evaluated on the high resolution mesh in element coordinates in terms of the solutions, $\mathcal{N}_\alpha$, of Equation (3.31),

$$\mathcal{N}_\alpha\,(\boldsymbol{x}\,(\boldsymbol{\xi})) = \sum_{K\in I_e} N_K(\boldsymbol{\xi})\mathcal{N}_{\alpha K}, \tag{3.42}$$

so Equation (3.41) becomes:

$$m^{se}_{\alpha\beta} \approx \sum_{e\in I_{se}} \sum_{i\in g_e} w^e_i \left(\sum_{K\in I_e} N_K(\boldsymbol{\xi})\mathcal{N}_{\alpha K} \right) \rho C_p \left(\sum_{L\in I_e} N_L(\boldsymbol{\xi})\mathcal{N}_{\beta L} \right) \det J_{\boldsymbol{x}\boldsymbol{\xi}} \tag{3.43}$$

which is the capacitance matrix for super element, *se*.

Evaluate the second integral in Equation (3.14) over the domain of the super element:

$$\int_\Omega \nabla\delta\phi \cdot k\nabla T\, d\Omega = \int_{\Omega_{se}} \frac{\partial \mathcal{N}_\alpha(\boldsymbol{x})}{\partial x_i}\, \delta\phi_\alpha\, k_{ij} \frac{\partial \mathcal{N}_\beta(\boldsymbol{x})}{\partial x_j}\, T_\beta(t)\, d\Omega \tag{3.44}$$

to determine the conduction matrix of the super element:

$$k^{se}_{\alpha\beta} = \int_{\Omega_{se}} \frac{\partial \mathcal{N}_\alpha(\boldsymbol{x})}{\partial x_i}\, k_{ij} \frac{\partial \mathcal{N}_\beta(\boldsymbol{x})}{\partial x_j}\, d\Omega \tag{3.45}$$

or, as a sum over the local sub-elements:

$$k^{se}_{\alpha\beta} = \sum_{e\in I_{se}} \int_{\Omega_{se}\cap\Omega_e} \frac{\partial \mathcal{N}_\alpha(\boldsymbol{x}(\boldsymbol{\xi}))}{\partial x_i}\, k_{ij} \frac{\partial \mathcal{N}_\beta(\boldsymbol{x}(\boldsymbol{\xi}))}{\partial x_j}\, \det J_{\boldsymbol{x}\boldsymbol{\xi}}\, d\Omega_\xi \tag{3.46}$$

The gradient operation,

$$\frac{\partial T(\boldsymbol{x},t)}{\partial x_i} = \sum_{\alpha\in I_{sn}} \frac{\partial \mathcal{N}_\alpha(\boldsymbol{x})}{\partial x_i} T_\alpha(t) \tag{3.47}$$

can be evaluated on the local elements in element coordinates:

$$\frac{\partial \mathcal{N}_\alpha(\boldsymbol{x}(\boldsymbol{\xi}))}{\partial x_i} = \sum_{k=1}^{n_d} \frac{\partial \mathcal{N}_\alpha(\boldsymbol{x}(\boldsymbol{\xi}))}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \tag{3.48}$$

Differentiating Equation (3.42),

$$\frac{\partial \mathscr{N}_\alpha(\boldsymbol{x}(\boldsymbol{\xi}))}{\partial \xi_k} = \sum_{K\in I_e} \frac{\partial N_K(\boldsymbol{\xi})}{\partial \xi_k} \mathscr{N}_{\alpha K} \tag{3.49}$$

and substituting into Equation (3.48) yields,

$$\frac{\partial \mathscr{N}_\alpha(\boldsymbol{x}(\boldsymbol{\xi}))}{\partial x_i} = \sum_{K\in I_e} \sum_{k=1}^{n_d} \frac{\partial N_K(\boldsymbol{\xi})}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \mathscr{N}_{\alpha K} = \sum_{K\in I_e} B_{iK}(\boldsymbol{\xi}) \mathscr{N}_{\alpha K} \tag{3.50}$$

which defines the gradient of the global basis in terms of the local element operator, B. Substituting into Equation (3.46) and approximating the integrals by quadrature yields the thermal conduction matrix for super element, *se*:

$$k^{se}_{\alpha\beta} \approx \sum_{e\in I_{se}} \sum_{g\in I_g} w^e_g\, B_{iL}(\boldsymbol{\xi}^e_g)\, \mathscr{N}_{\alpha L}\, K_{ij}\, B_{jM}(\boldsymbol{\xi}^e_g)\, \mathscr{N}_{\beta M}\, \det J_{\boldsymbol{x}\boldsymbol{\xi}} \tag{3.51}$$

Element quantities, Equations (3.43) and (3.51), are assembled into the coarsened global capacitance and conduction matrices, $\mathscr{M}$ and $\mathscr{K}$, respectively. The system of ODEs,

$$\mathscr{M}\, \dot{\boldsymbol{T}}(t) + \mathscr{K}\, \boldsymbol{T}(t) + \mathscr{F} - \mathscr{Q} = 0, \tag{3.52}$$

will generally be $\mathscr{O}(10^2)$ in size and can be solved much faster that the high resolution counterpart (typically $\mathscr{O}(10^4)$ to $\mathscr{O}(10^6)$ in size). Once the time histories, $\boldsymbol{T}(t)$, are computed, the approximated high resolution temperature history can be determined in element coordinates using Equations (3.37) and (3.42):

$$T(\boldsymbol{x}(\boldsymbol{\xi}),t) = \sum_{\alpha\in I_{se}} \sum_{K\in I_e} N_K(\boldsymbol{\xi}) \mathscr{N}_{\alpha K} T_\alpha(t). \tag{3.53}$$

3.1.4. Discrete form

Equation (3.52) is discrete spatially, but continuous temporally. A backward Euler method,

$$\dot{\boldsymbol{T}}^k \approx \frac{\boldsymbol{T}^k - \boldsymbol{T}^{k-1}}{\Delta t}, \quad \boldsymbol{T}^k = \boldsymbol{T}(t_k) \tag{3.54}$$

is substituted into Equation (3.52) at step k $(t = t_k)$ to produce a fully discrete form for the heat equation residual,

$$\boldsymbol{Q}^k\left(\boldsymbol{T}^k, \boldsymbol{T}^{k-1}\right) = (\mathscr{M} + \Delta t \mathscr{K})\, \boldsymbol{T}^k - \mathscr{M} \boldsymbol{T}^{k-1} + \Delta t \boldsymbol{F}^k = 0, \tag{3.55}$$

that is first order accurate in Δt. Alternatively, the Trapezoid rule can be used which produces the residual expression:

$$\boldsymbol{Q}^k\left(\boldsymbol{T}^k, \boldsymbol{T}^{k-1}\right) = \left(\mathscr{M} + \frac{\Delta t}{2}\mathscr{K}\right) \boldsymbol{T}^k - \left(\mathscr{M} - \frac{\Delta t}{2}\mathscr{K}\right) \boldsymbol{T}^{k-1} - \frac{\Delta t}{2}\left(\boldsymbol{F}^k + \boldsymbol{F}^{k-1}\right) = 0 \tag{3.56}$$

which is second order accurate in Δt.

3.2. Thermo-elastoplasticity

3.2.1. Strong Statement

Consider the equation for static mechanical equilibrium in domain Ω with fixed displacement on surface Γ^d and loads on surface Γ^n, $\partial\Omega = \Gamma^d \cup \Gamma^n$, $\Gamma^d \cap \Gamma^n = \emptyset$:

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} &= 0 \ \text{ in } \Omega \in R^{n_d} \qquad (3.57) \\ \boldsymbol{\sigma} \equiv \boldsymbol{\sigma}(\boldsymbol{\varepsilon}, T) &\ : \ \text{Cauchy stress} \\ \boldsymbol{b} \equiv \boldsymbol{b}(\boldsymbol{x}, t) &\ : \ \text{body forces} \end{aligned}$$

with boundary conditions:

$$\boldsymbol{u}(\boldsymbol{x}, t) = \tilde{\boldsymbol{u}}(t) \ \text{ on } \Gamma^d \ \forall\, t \in \left(t_o, t_f\right) \tag{3.58}$$

$$\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}} = \tilde{\boldsymbol{\tau}}(t) \ \text{ on } \Gamma^n \ \forall\, t \in \left(t_o, t_f\right) \tag{3.59}$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_o(\boldsymbol{x}) \ \ \forall\, \boldsymbol{x} \in \Omega \tag{3.60}$$

where $\tilde{\boldsymbol{u}}(t)$, $\tilde{\boldsymbol{\tau}}(t)$, and $\boldsymbol{u}_o(\mathbf{x})$ are given, and $\hat{\mathbf{n}}$ is the outward normal to Γ^n.

3.2.2. Weak Statement

$$\int_\Omega \delta\boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b})\, d\Omega = 0 \ \ \forall\, \delta\boldsymbol{u} \tag{3.61}$$

$$\tag{3.62}$$

Expanding terms,

$$\int_\Omega \delta\boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\sigma})\, d\Omega + \int_\Omega \delta\boldsymbol{u} \cdot \boldsymbol{b}\, d\Omega = 0 \ \ \forall\, \delta\boldsymbol{u}, \tag{3.63}$$

and integrating by parts,

(product rule)

$$\nabla \cdot (\delta\boldsymbol{u} \cdot \boldsymbol{\sigma}) = \nabla\delta\boldsymbol{u} : \boldsymbol{\sigma} + \delta\boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\sigma}) \tag{3.64}$$

(integrated over domain)

$$\int_\Omega \delta\boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\sigma})\, d\Omega = \int_\Omega \nabla \cdot (\delta\boldsymbol{u} \cdot \boldsymbol{\sigma})\, d\Omega - \int_\Omega \nabla\delta\boldsymbol{u} : \boldsymbol{\sigma}\, d\Omega \tag{3.65}$$

(divergence theorem)

$$\int_\Omega \nabla \cdot (\delta\boldsymbol{u} \cdot \boldsymbol{\sigma})\, d\Omega = \int_{\partial\Omega} (\delta\boldsymbol{u} \cdot \boldsymbol{\sigma}) \cdot \hat{\boldsymbol{n}}\, d\Gamma \tag{3.66}$$

yields the weak form:

$$\int_\Omega \nabla\delta\boldsymbol{u} : \boldsymbol{\sigma}\, d\Omega + \int_\Omega \delta\boldsymbol{u} \cdot \boldsymbol{b}\, d\Omega - \int_{\partial\Omega} \delta\boldsymbol{u} \cdot \hat{\boldsymbol{\tau}}\, d\Gamma = 0 \tag{3.67}$$

3.2.3. Discrete Form

Essential fields are approximated using nodal basis functions:

$$u_i(\boldsymbol{x}) = \sum_{\alpha \in I_n} N_\alpha(\boldsymbol{x}) u_{\alpha i}. \tag{3.68}$$

Or, in element coordinates, $\boldsymbol{\xi}$:

$$u_i(\boldsymbol{\xi}) = \sum_{\alpha \in I_e} N_\alpha(\boldsymbol{\xi}) u_{\alpha i} \;\; \boldsymbol{\xi} \in \Omega_e. \tag{3.69}$$

The map from element coordinates to global is:

$$x_i(\boldsymbol{\xi}) = \sum_{\alpha \in I_e} N_\alpha(\boldsymbol{\xi}) x_{\alpha i} \;\; \boldsymbol{\xi} \in \Omega_e. \tag{3.70}$$

The gradient of $\boldsymbol{u}$ is:

$$\frac{\partial u_i(\boldsymbol{\xi})}{\partial \xi_j} = \sum_{\alpha \in I_e} \frac{\partial N_\alpha(\boldsymbol{\xi})}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_j} u_{\alpha i} = B_{\alpha j}(\boldsymbol{\xi}) u_{\alpha i} \tag{3.71}$$

where

$$\frac{\partial \xi_k}{\partial x_j} = \left(\frac{\partial x_k}{\partial \xi_j} \right)^{-1} \tag{3.72}$$

$$\frac{\partial x_j}{\partial \xi_k} = \sum_{\alpha \in I_e} \frac{\partial N_\alpha(\boldsymbol{\xi})}{\partial \xi_k} x_{\alpha j} \tag{3.73}$$

Equation (3.71) defines the gradient operator, $B_{\alpha j}(\boldsymbol{\xi})$. The discrete approximations are substituted into the weak form in Equation (3.67) and the integral over the domain is evaluated over the individual mesh elements and summed:

(first term)

$$\int_\Omega \delta u_{i,j} \sigma_{ij} \, d\Omega = \sum_{e \in I_e} \delta u_{\alpha i} \int_{\Omega_e} B_{\alpha j}(\boldsymbol{\xi}) \, \sigma_{ij}(\boldsymbol{\xi}) \det J_{\boldsymbol{x}\boldsymbol{\xi}} \, d\Omega_\xi \tag{3.74}$$

(second term)

$$\int_\Omega \delta u_i b_i \, d\Omega = \sum_{e \in I_e} \delta u_{\alpha i} \int_{\Omega_e} N_\alpha(\boldsymbol{\xi}) b_i(\boldsymbol{\xi}) \, d\Omega_\xi \tag{3.75}$$

(third term)

$$\int_{\partial\Omega} \delta u_i \hat{\tau}_i \, d\Gamma = \sum_{e \in I_e} \delta u_{\alpha i} \int_{\partial\Omega_e} N_\alpha(\boldsymbol{\xi}) \, \hat{\tau}_i \, d\Gamma^n_\xi \tag{3.76}$$

The individual element terms are

$$f^e_{\alpha i} = \int_{\Omega_e} B_{\alpha j}(\boldsymbol{\xi}) \, \sigma_{ij}(\boldsymbol{\xi}) \det J_{\boldsymbol{x}\boldsymbol{\xi}} \, d\Omega_\xi \tag{3.77}$$

$$\tau^e_{\alpha i} = \int_{\Omega_e} N_\alpha(\boldsymbol{\xi}) b_i(\boldsymbol{\xi}) \det J_{\boldsymbol{x}\boldsymbol{\xi}} \, d\Omega_\xi \tag{3.78}$$

$$b^e_{\alpha i} = \int_{\partial\Omega_e} N_\alpha(\boldsymbol{\xi}) \hat{\tau}_i(\boldsymbol{\xi}) \det J_{\boldsymbol{x}\boldsymbol{\xi}} \, d\Gamma^n_\xi \tag{3.79}$$

which are approximated by quadrature:

$$f^e_{\alpha i} \approx \sum_{g \in I_g} w_g B_{\alpha j}(\boldsymbol{\xi_g})\, \sigma_{ij}(\boldsymbol{\xi_g})\, \det J_{\boldsymbol{x}\boldsymbol{\xi}_g} \tag{3.80}$$

$$\tau^e_{\alpha i} \approx \sum_{g \in I_g} w_g N_\alpha(\boldsymbol{\xi_g}) b_i(\boldsymbol{\xi_g})\, \det J_{\boldsymbol{x}\boldsymbol{\xi}_g} \tag{3.81}$$

$$b^e_{\alpha i} \approx \sum_{g \in I_g} w_g N_\alpha(\boldsymbol{\xi_g}) \hat{\tau}_i(\boldsymbol{\xi_g})\, \det J_{\boldsymbol{x}\boldsymbol{\xi}_g} \tag{3.82}$$

The global force vectors are:

$$\boldsymbol{F}_{int} = \mathop{\Large\mathsf{A}}_{e \in I_e} \boldsymbol{f}^e \tag{3.83}$$

$$\boldsymbol{F}_{ext} = \mathop{\Large\mathsf{A}}_{e \in I_e} (\boldsymbol{b}^e - \boldsymbol{\tau}^e) \tag{3.84}$$

and the global residual is:

$$\boldsymbol{R}(\boldsymbol{U}) = \boldsymbol{F}_{int}(\boldsymbol{U}) + \boldsymbol{F}_{ext} = 0 \tag{3.85}$$

where $\boldsymbol{U}$ is the global vector of nodal displacements.

The internal force vector, $\boldsymbol{F}_{int}$, depends on the stress value at each gauss point. The stress is a function of the local strain state ε_{ij}, temperature change ΔT, and reference strain ε^o_{kj}. In the context of elastoplasticity, the reference strain is the accumulated plastic strain, $\varepsilon^o_{ij} \equiv \varepsilon^p_{ij}$, which varies with deformation as a non-linear function, $\boldsymbol{H}$, of internal state variables, $\boldsymbol{c}$. To accurately compute the non-linear response, the external loads are applied over a sequence of load increments, and the residual, Equation (3.85), is solved at each step, k. The problem statement becomes:

$$\boldsymbol{R}^k \left(\boldsymbol{U}^k, \boldsymbol{U}^{k-1}, \boldsymbol{T}^k, \boldsymbol{T}^{k-1}, \boldsymbol{c}^k, \boldsymbol{c}^{k-1} \right) = 0 \tag{3.86}$$

$$\boldsymbol{H}^k \left(\boldsymbol{U}^k, \boldsymbol{U}^{k-1}, \boldsymbol{T}^k, \boldsymbol{T}^{k-1}, \boldsymbol{c}^k, \boldsymbol{c}^{k-1} \right) = 0 \tag{3.87}$$

$$k = 1, 2, ..., n \tag{3.88}$$

where the dependence of the residuals at k on the previous displacement solution, temperature, and internal state, ($\boldsymbol{U}^{k-1}$, $\boldsymbol{T}^{k-1}$, and $\boldsymbol{c}^{k-1}$), is shown explicitly. The state residual function, $\boldsymbol{H}$, can be solved locally at each gauss point but is shown as a global function for simplicity.

3.3. Stabilized Simplex Formulation

This section describes a formulation proposed by Chiumenta [19] that can be used to accurately model incompressible or nearly incompressible deformation behavior using simplex elements. A two-field approach is used where the independent fields are displacement, $\boldsymbol{u}$, and pressure, p. The stress is given as a sum of the volumetric and deviatoric parts,

$$\boldsymbol{\sigma}(p, \boldsymbol{u}) = p\boldsymbol{I} + \boldsymbol{s}(\boldsymbol{u}), \tag{3.89}$$

and the pressure is related to the displacements by

$$p(\boldsymbol{u}) = K\nabla \cdot \boldsymbol{u}. \tag{3.90}$$

The condition for mechanical equilibrium, Equation (3.57), is combined with Equations (3.89) and (3.90) to produce a two-field system:

$$\begin{aligned}
\nabla \cdot \boldsymbol{s}(\boldsymbol{u}) + \nabla p + f &= 0 \ \text{ in } \ \Omega \qquad (3.91)\\
\nabla \cdot \boldsymbol{u} - \frac{1}{K} p &= 0 \ \text{ in } \ \Omega \qquad (3.92)\\
\boldsymbol{u}(\boldsymbol{x},t) &= \boldsymbol{u}_d(t) \ \text{ on } \Gamma^d \ \forall\, t \in \left(t_o, t_f\right)\\
\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}} &= \boldsymbol{\tau}_n(t) \ \text{ on } \Gamma^n \ \forall\, t \in \left(t_o, t_f\right)\\
\boldsymbol{u}(\boldsymbol{x},0) &= \boldsymbol{u}_o(\boldsymbol{x}) \ \ \forall\, \boldsymbol{x} \in \Omega
\end{aligned}$$

Defining $\langle \cdot, \cdot \rangle$ to be an inner product in $L^2(\Omega)$, the weak form of the two-field system can be expressed as

$$\begin{aligned}
\langle \nabla^s \boldsymbol{v}, \boldsymbol{s} \rangle + \langle \nabla \cdot \boldsymbol{v}, p \rangle - \langle \boldsymbol{v}, \boldsymbol{f} \rangle - \langle \boldsymbol{v}, \boldsymbol{\tau} \rangle_{\partial\Omega} &= 0 \qquad (3.93)\\
\left\langle q, \left(\nabla \cdot \boldsymbol{u} - \frac{1}{K} p \right) \right\rangle &= 0 \qquad (3.94)
\end{aligned}$$

where integration by parts has been used to eliminate gradients on stress and pressure and introduce the Neumann boundary condition, and $\boldsymbol{v}$ and q are the variations on displacement and pressure. The system can be expressed compactly as

$$\begin{aligned}
\mathscr{R}(\boldsymbol{U}, \boldsymbol{V}) &= 0 \qquad (3.95)\\
\boldsymbol{U} &= [\boldsymbol{u}, p]^T \qquad (3.96)\\
\boldsymbol{V} &= [\boldsymbol{v}, q]^T. \qquad (3.97)
\end{aligned}$$

Discrete forms of Equation (3.95) may or may not be stable depending on the space of functions used for the independent and variational variables. For example, a P2P0 interpolation of $(\boldsymbol{u}, p)$ can be shown to be stable, whereas a P1P1 or P1P0 interpolation cannot.

The formulation proposed by Chiumenta [19] uses a sub-grid approach to stabilize the P1P1 interpolation, so one can in fact use linear simplex elements to model incompressible and nearly-incompressible materials. To that end, the solution and variations are approximated as the combination of coarse and fine scales:

$$\begin{aligned}
\boldsymbol{U} &= \boldsymbol{U}_h + \tilde{\boldsymbol{U}} \qquad (3.98)\\
\boldsymbol{U}_h &= [\boldsymbol{u}_h, p_h]^T \ \ : \text{ coarse}\\
\tilde{\boldsymbol{U}} &= [\tilde{\boldsymbol{u}}, 0]^T \quad \ : \text{ fine}
\end{aligned}$$

The coarse scale variables, $\boldsymbol{u}_h$ and p_h, are supported on a linear tetrahedron (3D) or triangle (2D) mesh, and the fine scale displacement, $\tilde{\boldsymbol{u}}$, will be computed in a sub-space that is orthogonal to the finite element space. Note that there is not a pressure defined at the fine scale. With this partitioning, the system becomes

$$\mathscr{R}\left(\boldsymbol{U}_h+\tilde{\boldsymbol{U}},\boldsymbol{V}_h\right)=0 \tag{3.99}$$
$$\mathscr{R}\left(\boldsymbol{U}_h+\tilde{\boldsymbol{U}},\tilde{\boldsymbol{V}}\right)=0 \tag{3.100}$$

which is solved to determine the fine scale displacement, $\tilde{\boldsymbol{u}}$, and the coarse scale displacement and pressure, $\boldsymbol{u}_h$ and p_h. In more concrete terms, Equation (3.99) is

$$\langle\nabla^s\boldsymbol{v}_h,\boldsymbol{s}(\boldsymbol{u}_h+\tilde{\boldsymbol{u}})\rangle+\langle\nabla\cdot\boldsymbol{v}_h,p_h\rangle-\langle\boldsymbol{v}_h,\boldsymbol{f}\rangle-\langle\boldsymbol{v}_h,\boldsymbol{\tau}\rangle_{\partial\Omega}=0 \tag{3.101}$$
$$\left\langle q_h,\left(\nabla\cdot(\boldsymbol{u}_h+\tilde{\boldsymbol{u}})-\frac{1}{K}p_h\right)\right\rangle=0, \tag{3.102}$$

or, by moving terms involving fine scale quantities to the right hand side,

$$\mathscr{R}(\boldsymbol{U}_h,\boldsymbol{V}_h)=-\mathscr{R}\left(\tilde{\boldsymbol{U}},\boldsymbol{V}_h\right) \tag{3.103}$$
$$\mathscr{R}(\boldsymbol{U}_h,\boldsymbol{V}_h)=\left[\begin{array}{l}\langle\nabla^s\boldsymbol{v}_h,\boldsymbol{s}(\boldsymbol{u}_h)\rangle+\langle\nabla\cdot\boldsymbol{v}_h,p_h\rangle-\langle\boldsymbol{v}_h,\boldsymbol{f}\rangle-\langle\boldsymbol{v}_h,\boldsymbol{\tau}\rangle_{\partial\Omega}\\ \langle q_h,\nabla\cdot\boldsymbol{u}_h\rangle-\left\langle q_h,\frac{1}{K}p_h\right\rangle\end{array}\right] \tag{3.104}$$
$$\mathscr{R}\left(\tilde{\boldsymbol{U}},\boldsymbol{V}_h\right)=\left[\begin{array}{l}\langle\nabla^s\boldsymbol{v}_h,\tilde{\boldsymbol{s}}(\tilde{\boldsymbol{u}})\rangle\\ \langle q_h,\nabla\cdot\tilde{\boldsymbol{u}}\rangle\end{array}\right]. \tag{3.105}$$

where it has been assumed that the total deviatoric stress is the sum of the coarse scale and fine scale parts, i.e.,

$$\boldsymbol{s}(\boldsymbol{u}_h+\tilde{\boldsymbol{u}})=\boldsymbol{s}(\boldsymbol{u}_h)+\tilde{\boldsymbol{s}}(\tilde{\boldsymbol{u}}). \tag{3.106}$$

To simplify the forcing term in Equation (3.105) it is further assumed that the fine scale displacements do not produce plastic deformation, i.e., $\tilde{\boldsymbol{s}}(\tilde{\boldsymbol{u}})=2G\,\mathrm{dev}(\nabla^s\cdot\tilde{\boldsymbol{u}})$, then integration by parts is used to arrive at

$$\mathscr{R}\left(\tilde{\boldsymbol{U}},\boldsymbol{V}_h\right)=\left[\begin{array}{l}\langle\tilde{\boldsymbol{u}},\nabla\cdot(2G\,\mathrm{dev}\,(\nabla^s\boldsymbol{v}_h))\rangle\\ \langle\tilde{\boldsymbol{u}},\nabla q_h\rangle\end{array}\right]. \tag{3.107}$$

Given fine scale displacements, $\tilde{\boldsymbol{u}}$, Equations (3.103), (3.104), and (3.107) can be solved to find the coarse scale displacement and pressure.

The fine scale displacement is provided by Equation (3.100) which concretely is

$$\mathscr{R}\left(\boldsymbol{U}_h,\tilde{\boldsymbol{V}}\right)=-\mathscr{R}\left(\tilde{\boldsymbol{U}},\tilde{\boldsymbol{V}}\right) \tag{3.108}$$
$$\mathscr{R}\left(\boldsymbol{U}_h,\tilde{\boldsymbol{V}}\right)=\left[\begin{array}{l}\langle\tilde{\boldsymbol{v}},\nabla\cdot\boldsymbol{s}_h(\boldsymbol{u}_h)\rangle+\langle\tilde{\boldsymbol{v}},\nabla p_h\rangle+\langle\tilde{\boldsymbol{v}},\boldsymbol{f}\rangle\\ 0\end{array}\right] \tag{3.109}$$
$$\mathscr{R}\left(\tilde{\boldsymbol{U}},\tilde{\boldsymbol{V}}\right)=\left[\begin{array}{l}\langle\tilde{\boldsymbol{v}},\nabla\cdot\tilde{\boldsymbol{s}}(\tilde{\boldsymbol{u}})\rangle\\ 0\end{array}\right]. \tag{3.110}$$

Note that the weak form above has not been integrated by parts. Recalling that fine scale displacements do not lead to plastic deformation, the non-zero term in Equation (3.110) is simplified to

$$\langle\tilde{\boldsymbol{v}},\nabla\cdot\tilde{\boldsymbol{s}}(\tilde{\boldsymbol{u}})\rangle=\langle\tilde{\boldsymbol{v}},\nabla\cdot(2G\,\mathrm{dev}\,(\nabla^s\tilde{\boldsymbol{u}}))\rangle \tag{3.111}$$

The fine scale displacement, in order to stabilize the pressure gradient (a piecewise continuous field), is approximated within each element as

$$\langle \tilde{\boldsymbol{v}}, \nabla \cdot (2G\,\mathrm{dev}\,(\nabla^s \tilde{\boldsymbol{u}})) \rangle \approx \sum_{e=1}^{N_{el}} \left\langle \frac{\tilde{\boldsymbol{v}}}{c_1 h_e}, \frac{\tilde{\boldsymbol{u}}_e}{c_1 h_e} \right\rangle_{\Omega_e} = \sum_{e=1}^{N_{el}} \tau_e^{-1} \langle \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{u}}_e \rangle_{\Omega_e} \tag{3.112}$$

where $\tau_e = c_1 c_2 h_e^2/(2G)$, and $\tilde{\boldsymbol{u}}_e$ is the fine scale displacement for element e. The non-zero term in Equation (3.109) can be stated as

$$\langle \tilde{\boldsymbol{v}}, \nabla \cdot \boldsymbol{s}_h(\boldsymbol{u}_h) \rangle + \langle \tilde{\boldsymbol{v}}, \nabla p_h \rangle + \langle \tilde{\boldsymbol{v}}, \boldsymbol{f} \rangle = \langle \tilde{\boldsymbol{v}}, \boldsymbol{r}_h \rangle \tag{3.113}$$

$$\boldsymbol{r}_h = \nabla \cdot \boldsymbol{s}_h(\boldsymbol{u}_h) + \nabla p_h + \boldsymbol{f}. \tag{3.114}$$

The fine scale displacement will be computed in a space that is orthogonal to the finite element space of the coarse scale. That is, Equation (3.108) combined with Equations (3.112) and (3.114) yields

$$\sum_{e=1}^{N_{el}} \langle \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{u}}_e \rangle_{\Omega_e} = \sum_{e=1}^{N_{el}} \tau_e \left\langle \tilde{\boldsymbol{v}}, P_h^{\perp}(\boldsymbol{r}_h) \right\rangle_{\Omega_e} \tag{3.115}$$

where $P_h^{\perp}(\cdot)$ is the projection of $(\cdot)$ into the orthogonal space, i.e., $P_h^{\perp}(\cdot) = (\cdot) - P_h(\cdot)$. The fine scale displacement within each element is then

$$\tilde{\boldsymbol{u}}_e = \tau_e \left(\boldsymbol{r}_h - P_h(\boldsymbol{r}_h)\right) \tag{3.116}$$

Because $\boldsymbol{f}$ is in the finite element space (so projections into the orthogonal space are zero), and $\boldsymbol{u}$ is linear (so second derivatives are zero), $\boldsymbol{r}_h = \nabla p_h$, and

$$\tilde{\boldsymbol{u}}_e = \tau_e \left(\nabla p_h - P_h(\nabla p_h)\right) \tag{3.117}$$

The projection of the pressure gradient, ∇p_h, into the finite element space is

$$\langle \boldsymbol{\Pi}_h, \boldsymbol{\eta}_h \rangle = \langle \nabla p_h, \boldsymbol{\eta}_h \rangle \tag{3.118}$$

where $\boldsymbol{\Pi}$ is the projected nodal pressure gradient, so the fine scale displacement is finally

$$\tilde{\boldsymbol{u}}_e = \tau_e \left(\nabla p_h - \boldsymbol{\Pi}_h\right). \tag{3.119}$$

The stabilization term, Equation (3.107), is then

$$\mathscr{R}\left(\tilde{\boldsymbol{U}}, \boldsymbol{V}_h\right) = \begin{bmatrix} 0 \\ \sum_{e=1}^{N_{el}} \tau_e \left\langle (\nabla p_h - \boldsymbol{\Pi}_h), \nabla q_h \right\rangle_{\Omega_e} \end{bmatrix}. \tag{3.120}$$

and the stabilized system is

$$\langle \nabla^s \boldsymbol{v}_h, \boldsymbol{s}(\boldsymbol{u}_h) \rangle + \langle \nabla \cdot \boldsymbol{v}_h, p_h \rangle - \langle \boldsymbol{v}_h, \boldsymbol{f} \rangle - \langle \boldsymbol{v}_h, \boldsymbol{\tau} \rangle_{\partial\Omega} = 0 \tag{3.121}$$

$$\langle q_h, \nabla \cdot \boldsymbol{u}_h \rangle - \left\langle q_h, \frac{1}{K} p_h \right\rangle - \sum_{e=1}^{N_{el}} \tau_e \left\langle (\nabla p_h - \boldsymbol{\Pi}_h), \nabla q_h \right\rangle_{\Omega_e} = 0 \tag{3.122}$$

$$\langle \boldsymbol{\Pi}_h, \boldsymbol{\eta}_h \rangle = \langle \nabla p_h, \boldsymbol{\eta}_h \rangle = 0, \tag{3.123}$$

which can be put in the form

$$\boldsymbol{R}^k\left(\boldsymbol{\phi},\boldsymbol{u}^k,\boldsymbol{u}^{k-1},\boldsymbol{p}^k,\boldsymbol{p}^{k-1},\boldsymbol{\Pi}^k,\boldsymbol{\Pi}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)=0 \tag{3.124}$$

$$\boldsymbol{H}^k\left(\boldsymbol{\phi},\boldsymbol{u}^k,\boldsymbol{u}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)=0 \tag{3.125}$$

$$k=1,2,...,n \tag{3.126}$$

where $\boldsymbol{\phi}$ is a nodal field that describes the material topology. Alternatively, as proposed by Chiumenti [19], the projected pressure gradient at time step $k-1$, i.e. $\boldsymbol{\Pi}^{k-1}$, is used to solve the stabilized global residual at time step k, i.e.

$$\boldsymbol{R}^k\left(\boldsymbol{\phi},\boldsymbol{u}^k,\boldsymbol{u}^{k-1},\boldsymbol{p}^k,\boldsymbol{p}^{k-1},\boldsymbol{\Pi}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)=0. \tag{3.127}$$

Therefore, the stabilized system of equations is defined as

$$\boldsymbol{R}^k\left(\boldsymbol{\phi},\boldsymbol{u}^k,\boldsymbol{u}^{k-1},\boldsymbol{p}^k,\boldsymbol{p}^{k-1},\boldsymbol{\Pi}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)=0 \tag{3.128}$$

$$\boldsymbol{H}^k\left(\boldsymbol{\phi},\boldsymbol{u}^k,\boldsymbol{u}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)=0 \tag{3.129}$$

$$\boldsymbol{P}^k\left(\boldsymbol{\Pi}^k,\boldsymbol{p}^k\right)=0 \tag{3.130}$$

$$k=1,2,...,n \tag{3.131}$$

or in its compact form

$$\begin{aligned}\boldsymbol{R}^k\left(\boldsymbol{\phi},\boldsymbol{U}^k,\boldsymbol{U}^{k-1},\boldsymbol{\Pi}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)&=0\\ \boldsymbol{H}^k\left(\boldsymbol{\phi},\boldsymbol{U}^k,\boldsymbol{U}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)&=0\\ \boldsymbol{P}^k\left(\boldsymbol{\Pi}^k,\boldsymbol{p}^k\right)&=0\\ k&=1,2,...,n\end{aligned} \tag{3.132}$$

where $\boldsymbol{U}=[\boldsymbol{u},\boldsymbol{p}]^T$.

3.4. Elastoplastic Topology Optimization

Consider a general optimization problem that consists of solving the stabilized, non-linear elastoplasticity equations:

$$\begin{aligned}\boldsymbol{R}^k\left(\boldsymbol{\phi},\boldsymbol{u}^k,\boldsymbol{u}^{k-1},\boldsymbol{p}^k,\boldsymbol{p}^{k-1},\boldsymbol{\Pi}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)&=0\\ \boldsymbol{H}^k\left(\boldsymbol{\phi},\boldsymbol{u}^k,\boldsymbol{u}^{k-1},\boldsymbol{c}^k,\boldsymbol{c}^{k-1}\right)&=0\\ \boldsymbol{P}^k(\boldsymbol{\Pi}^k,\boldsymbol{p}^k)&=0\\ k&=1,2,...,n\end{aligned} \tag{3.133}$$

where $\boldsymbol{u}^0$, $\boldsymbol{p}^0$, and $\boldsymbol{c}^0$ are the initial conditions. The residual functions are now considered dependent on a set of discrete values, $\boldsymbol{\phi}$, that define the current design. The optimization problem is stated as:

$$\boldsymbol{\phi}^* = \arg \min_{\boldsymbol{\phi} \in \mathbb{R}^m} f(\boldsymbol{\phi}) \tag{3.134}$$

subject to the set of equality constraint Equations (3.133). The objective function in Equation (3.134) is in the form:

$$f(\boldsymbol{\phi}) = F\left(\boldsymbol{\phi}, \boldsymbol{u}^k(\boldsymbol{\phi}), \boldsymbol{c}^k(\boldsymbol{\phi})\right) \quad k = 1, 2, ..., n \tag{3.135}$$

Following Alberdi [1], define a Lagrangian functional:

$$\hat{f} = f(\boldsymbol{\phi}) + \sum_{k=1}^{n} \boldsymbol{\lambda}^{n^T} \boldsymbol{R}^k + \sum_{k=1}^{n} \boldsymbol{\mu}^{n^T} \boldsymbol{H}^k + \sum_{k=1}^{n} \boldsymbol{\gamma}^{k^T} \boldsymbol{P}^k, \tag{3.136}$$

where

$$\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{\boldsymbol{u}}, \boldsymbol{\lambda}_{\boldsymbol{p}}]^T.$$

Next, compute the total derivative of $\hat{f}$ with respect to $\boldsymbol{\phi}$ using the chain rule:

$$\begin{aligned}
\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} &+ \sum_{k=1}^{n} \left(\frac{\partial F}{\partial \boldsymbol{U}^k} \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} + \frac{\partial F}{\partial \boldsymbol{c}^k} \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} + \frac{\partial F}{\partial \boldsymbol{\Pi}^k} \frac{\partial \boldsymbol{\Pi}^k}{\partial \boldsymbol{\phi}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\lambda}^{k^T} \left(\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^{k-1}} \frac{\partial \boldsymbol{U}^{k-1}}{\partial \boldsymbol{\phi}} \right. \\
&+ \left. \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^k} \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^{k-1}} \frac{\partial \boldsymbol{c}^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\Pi}^k} \frac{\partial \boldsymbol{\Pi}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\Pi}^{k-1}} \frac{\partial \boldsymbol{\Pi}^{k-1}}{\partial \boldsymbol{\phi}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\mu}^{k^T} \left(\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^{k-1}} \frac{\partial \boldsymbol{U}^{k-1}}{\partial \boldsymbol{\phi}} \right. \\
&+ \left. \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^{k-1}} \frac{\partial \boldsymbol{c}^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\Pi}^k} \frac{\partial \boldsymbol{\Pi}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\Pi}^{k-1}} \frac{\partial \boldsymbol{\Pi}^{k-1}}{\partial \boldsymbol{\phi}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\gamma}^{k^T} \left(\frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{U}^k} \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{U}^{k-1}} \frac{\partial \boldsymbol{U}^{k-1}}{\partial \boldsymbol{\phi}} \right. \\
&+ \left. \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{c}^k} \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{c}^{k-1}} \frac{\partial \boldsymbol{c}^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{\Pi}^k} \frac{\partial \boldsymbol{\Pi}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{\Pi}^{k-1}} \frac{\partial \boldsymbol{\Pi}^{k-1}}{\partial \boldsymbol{\phi}} \right)
\end{aligned} \tag{3.137}$$

Rearranging terms yields:

$$
\begin{aligned}
\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} + \sum_{k=1}^{n} \left(\boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\gamma}^{kT} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{\phi}} \right) \\
+ \left(\frac{\partial F}{\partial \boldsymbol{U}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n} + \boldsymbol{\gamma}^{nT} \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{U}^n} \right) \frac{\partial \boldsymbol{U}^n}{\partial \boldsymbol{\phi}} \\
+ \left(\frac{\partial F}{\partial \boldsymbol{c}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{c}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n} + \boldsymbol{\gamma}^{nT} \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{c}^n} \right) \frac{\partial \boldsymbol{c}^n}{\partial \boldsymbol{\phi}} \\
+ \left(\frac{\partial F}{\partial \boldsymbol{\Pi}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\Pi}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{\Pi}^n} + \boldsymbol{\gamma}^{nT} \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{\Pi}^n} \right) \frac{\partial \boldsymbol{\Pi}^n}{\partial \boldsymbol{\phi}} \\
+ \sum_{k=1}^{n-1} \left(\frac{\partial F}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{k+1T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{k+1T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\gamma}^{k+1T} \frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{U}^k} \right. \\
\left. + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\gamma}^{kT} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{U}^k} \right) \frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} \\
+ \sum_{k=1}^{n-1} \left(\frac{\partial F}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{k+1T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{k+1T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\gamma}^{k+1T} \frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{c}^k} \right. \\
\left. + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} + \boldsymbol{\gamma}^{kT} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{c}^k} \right) \frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} \\
+ \sum_{k=1}^{n-1} \left(\frac{\partial F}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\lambda}^{k+1T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\mu}^{k+1T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\gamma}^{k+1T} \frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{\Pi}^k} \right. \\
\left. + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\gamma}^{kT} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{\Pi}^k} \right) \frac{\partial \boldsymbol{\Pi}^k}{\partial \boldsymbol{\phi}}
\end{aligned}
\tag{3.138}
$$

which defines the adjoint problems:

Final step:

$$
\begin{aligned}
\frac{\partial F}{\partial \boldsymbol{U}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n} + \boldsymbol{\gamma}^{nT} \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{U}^n} = 0 \\
\frac{\partial F}{\partial \boldsymbol{c}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{c}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n} + \boldsymbol{\gamma}^{nT} \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{c}^n} = 0 \\
\frac{\partial F}{\partial \boldsymbol{\Pi}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\Pi}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{\Pi}^n} + \boldsymbol{\gamma}^{nT} \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{\Pi}^n} = 0
\end{aligned}
\tag{3.139}
$$

k^{th} step:

$$\begin{aligned}
&\frac{\partial F}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{k+1\,T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{k+1\,T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\gamma}^{k+1\,T} \frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{U}^k} \\
&\quad + \boldsymbol{\lambda}^{k\,T} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\gamma}^{k\,T} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{U}^k} = 0 \\
&\frac{\partial F}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{k+1\,T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{k+1\,T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\gamma}^{k+1\,T} \frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{c}^k} \\
&\quad + \boldsymbol{\lambda}^{k\,T} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} + \boldsymbol{\gamma}^{k\,T} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{c}^k} = 0 \\
&\frac{\partial F}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\lambda}^{k+1\,T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\mu}^{k+1\,T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\gamma}^{k+1\,T} \frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{\Pi}^k} \\
&\quad + \boldsymbol{\lambda}^{k\,T} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\Pi}^k} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{T}^k} + \boldsymbol{\gamma}^{k\,T} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{\Pi}^k} = 0 \\
&k = n-1, ..., 1, 2
\end{aligned} \tag{3.140}$$

The final step adjoint system of equations is given by:

$$\begin{pmatrix} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n}^T & \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n}^T & \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{U}^n}^T \\ \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{c}^n}^T & \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n}^T & \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{c}^n}^T \\ \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\Pi}^n}^T & \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{\Pi}^n}^T & \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{\Pi}^n}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}^n \\ \boldsymbol{\mu}^n \\ \boldsymbol{\gamma}^n \end{pmatrix} = - \begin{pmatrix} \frac{\partial F}{\partial \boldsymbol{U}^n} \\ \frac{\partial F}{\partial \boldsymbol{c}^n} \\ \frac{\partial F}{\partial \boldsymbol{\Pi}^n} \end{pmatrix} \tag{3.141}$$

The explicit expressions for $\boldsymbol{\lambda}^n$, $\boldsymbol{\mu}^n$, and $\boldsymbol{\gamma}^n$ $\forall\, k = 1, ..., n$ are derived from Equation (3.141). Indeed, solving for $\boldsymbol{\gamma}^n$ in Equation (3.141) yields the following expression for $\boldsymbol{\gamma}^n$:

$$\boldsymbol{\gamma}^n = -\left(\frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{\Pi}^n}\right)^{-T} \left(\frac{\partial F}{\partial \boldsymbol{\Pi}^n} + \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\Pi}^n}^T \boldsymbol{\lambda}^n + \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{\Pi}^n}^T \boldsymbol{\mu}^n\right) \tag{3.142}$$

However, the right hand side in Equation (3.142) is zero since $\frac{\partial F}{\partial \boldsymbol{\Pi}^n} = \boldsymbol{0}$ and Jacobians $\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{\Pi}^n}^T = \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{\Pi}^n}^T = \boldsymbol{0}$. Therefore, the adjoint variables $\boldsymbol{\gamma}^n = \boldsymbol{0}$. Next, $\boldsymbol{\mu}^n$ is derived by solving for $\boldsymbol{\mu}^n$ in Equation (3.141), which yields:

$$\boldsymbol{\mu}^n = -\left(\frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n}\right)^{-T} \left(\frac{\partial F}{\partial \boldsymbol{c}^n} + \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{c}^n}^T \boldsymbol{\lambda}^n + \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{c}^n}^T \boldsymbol{\gamma}^n\right) \tag{3.143}$$

Since $\boldsymbol{\gamma}^n = \boldsymbol{0}$, Equation (3.143) can be recast as:

$$\boldsymbol{\mu}^n = -\left(\frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n}\right)^{-T} \left(\frac{\partial F}{\partial \boldsymbol{c}^n} + \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{c}^n}^T \boldsymbol{\lambda}^n\right) \tag{3.144}$$

The explicit expression for $\boldsymbol{\lambda}^n$ is derived by solving for $\boldsymbol{\lambda}^n$ in Equation (3.141), which yields:

$$\left(\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n}^T \boldsymbol{\lambda}^n + \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n}^T \boldsymbol{\mu}^n\right) = -\left(\frac{\partial F}{\partial \boldsymbol{U}^n} + \frac{\partial \boldsymbol{P}^n}{\partial \boldsymbol{U}^n}^T \boldsymbol{\gamma}^n\right) \tag{3.145}$$

Recall that $\boldsymbol{\gamma}^n = \boldsymbol{0}$; hence, Equation (3.145) is recast as:

$$\left(\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n}^T \boldsymbol{\lambda}^n + \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n}^T \boldsymbol{\mu}^n\right) = -\left(\frac{\partial F}{\partial \boldsymbol{U}^n}\right) \tag{3.146}$$

Finally, substituting Equation (3.144) into Equation (3.146) yields:

$$\boldsymbol{\lambda}^{n} = \left({\frac{\partial \boldsymbol{R}^{n}}{\partial \boldsymbol{U}^{n}}}^{T} - \left({\frac{\partial \boldsymbol{H}^{n}}{\partial \boldsymbol{U}^{n}}}^{T} \left(\frac{\partial \boldsymbol{H}^{n}}{\partial \boldsymbol{c}^{n}} \right)^{-T} {\frac{\partial \boldsymbol{R}^{n}}{\partial \boldsymbol{c}^{n}}}^{T} \right) \right)^{-1} \boldsymbol{F}^{n} \tag{3.147}$$

where

$$\boldsymbol{F}^{n} = - \left(\frac{\partial F}{\partial \boldsymbol{U}^{n}} - {\frac{\partial \boldsymbol{H}^{n}}{\partial \boldsymbol{U}^{n}}}^{T} \left(\frac{\partial \boldsymbol{H}^{n}}{\partial \boldsymbol{c}^{n}} \right)^{-T} \frac{\partial F}{\partial \boldsymbol{c}^{n}} \right)$$

The k^{th} step adjoint system of equations is given by:

$$\begin{pmatrix} {\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} & {\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} & {\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} \\ {\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{c}^{k}}}^{T} & {\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{c}^{k}}}^{T} & {\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{c}^{k}}}^{T} \\ {\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{\Pi}^{k}}}^{T} & {\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{\Pi}^{k}}}^{T} & {\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}^{k} \\ \boldsymbol{\mu}^{k} \\ \boldsymbol{\gamma}^{k} \end{pmatrix} = - \begin{pmatrix} \frac{\partial F}{\partial \boldsymbol{U}^{k}} + {\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\lambda}^{k+1} + {\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\mu}^{k+1} + {\frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\gamma}^{k+1} \\ \frac{\partial F}{\partial \boldsymbol{c}^{k}} + {\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\lambda}^{k+1} + {\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\mu}^{k+1} + {\frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\gamma}^{k+1} \\ \frac{\partial F}{\partial \boldsymbol{\Pi}^{k}} + {\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\lambda}^{k+1} + {\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\mu}^{k+1} + {\frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\gamma}^{k+1} \end{pmatrix} \tag{3.148}$$

The explicit expressions for $\boldsymbol{\lambda}^{k}$, $\boldsymbol{\mu}^{k}$, and $\boldsymbol{\gamma}^{k}$ $\forall\ k = 1, ..., n$ are derived from Equation (3.148). First, the explicit expression for $\boldsymbol{\gamma}^{k}$ is derived by solving for $\boldsymbol{\gamma}^{k}$ in Equation (3.148), which yields:

$$\boldsymbol{\gamma}^{k} = - \left(\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{\Pi}^{k}} \right)^{-T} \left(\frac{\partial F}{\partial \boldsymbol{\Pi}^{k}} + {\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\lambda}^{k+1} + {\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\mu}^{k+1} + {\frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\gamma}^{k+1} + {\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\lambda}^{k} + {\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\mu}^{k} \right) \tag{3.149}$$

Equation (3.149) can be further simplified since multiple terms in Equation (3.149) are zero. Therefore, the final expression for $\boldsymbol{\gamma}^{k}$ is given by:

$$\boldsymbol{\gamma}^{k} = - \left(\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{\Pi}^{k}} \right)^{-T} \left({\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{\Pi}^{k}}}^{T} \boldsymbol{\lambda}^{k+1} \right) \tag{3.150}$$

Next, an explicit expression for $\boldsymbol{\mu}^{k}$ is derived by solving for $\boldsymbol{\mu}^{k}$ in Equation (3.148), which yields:

$$\boldsymbol{\mu}^{k} = - \left(\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{c}^{k}} \right)^{-T} \left(\frac{\partial F}{\partial \boldsymbol{c}^{k}} + {\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\lambda}^{k+1} + {\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\mu}^{k+1} + {\frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\gamma}^{k+1} + {\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\lambda}^{k} + {\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\gamma}^{k} \right) \tag{3.151}$$

Once again, multiple terms in Equation (3.151) are zero. Therefore, the explicit expression for $\boldsymbol{\mu}^{k}$ is given by:

$$\boldsymbol{\mu}^{k} = - \left(\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{c}^{k}} \right)^{-T} \left(\frac{\partial F}{\partial \boldsymbol{c}^{k}} + {\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\mu}^{k+1} + {\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{c}^{k}}}^{T} \boldsymbol{\lambda}^{k} \right) \tag{3.152}$$

Finally, the explicit expression for $\boldsymbol{\lambda}^{k}$ is derived by solving for $\boldsymbol{\lambda}^{k}$ in Equation (3.148) as follows:

$$\left({\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\lambda}^{k} + {\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\mu}^{k} \right) = - \left(\frac{\partial F}{\partial \boldsymbol{U}^{k}} + {\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\lambda}^{k+1} + {\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\mu}^{k+1} + {\frac{\partial \boldsymbol{P}^{k+1}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\gamma}^{k+1} + {\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\gamma}^{k} \right) \tag{3.153}$$

Since multiple terms in Equation (3.153) are zero, the expression for $\boldsymbol{\lambda}^{k}$ can be recast as:

$$\left({\frac{\partial \boldsymbol{R}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\lambda}^{k} + {\frac{\partial \boldsymbol{H}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\mu}^{k} \right) = - \left(\frac{\partial F}{\partial \boldsymbol{U}^{k}} + {\frac{\partial \boldsymbol{P}^{k}}{\partial \boldsymbol{U}^{k}}}^{T} \boldsymbol{\gamma}^{k} \right) \tag{3.154}$$

Substituting Equation (3.152) into Equation (3.154) yields:

$$\boldsymbol{\lambda}^k = \left(\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k}^T - \left(\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k}^T \left(\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} \right)^{-T} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^k}^T \right) \right)^{-1} \boldsymbol{F}^k \tag{3.155}$$

where

$$\boldsymbol{F}^k = -\left(\frac{\partial F}{\partial \boldsymbol{U}^k} + \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{U}^k}^T \boldsymbol{\gamma}^k - \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k}^T \left(\left(\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} \right)^{-T} \left(\frac{\partial F}{\partial \boldsymbol{c}^k} + \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^k}^T \boldsymbol{\mu}^{k+1} \right) \right) \right)$$

The total derivative of the objective is given by:

$$\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} + \sum_{k=1}^{n} \left(\boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\gamma}^{kT} \frac{\partial \boldsymbol{P}^k}{\partial \boldsymbol{\phi}} \right) \tag{3.156}$$

3.5. Thermo-elastoplastic Topology Optimization

Consider a thermo-elastoplasticity problem that consists of solving the heat equation and non-linear elastoplasticity equations:

$$\boldsymbol{Q}^k \left(\boldsymbol{\phi}, \boldsymbol{T}^k, \boldsymbol{T}^{k-1} \right) = 0 \tag{3.157}$$

$$\boldsymbol{R}^k \left(\boldsymbol{\phi}, \boldsymbol{T}^k, \boldsymbol{T}^{k-1}, \boldsymbol{U}^k, \boldsymbol{U}^{k-1}, \boldsymbol{c}^k, \boldsymbol{c}^{k-1} \right) = 0 \tag{3.158}$$

$$\boldsymbol{H}^k \left(\boldsymbol{\phi}, \boldsymbol{T}^k, \boldsymbol{T}^{k-1}, \boldsymbol{U}^k, \boldsymbol{U}^{k-1}, \boldsymbol{c}^k, \boldsymbol{c}^{k-1} \right) = 0 \tag{3.159}$$

$$k = 1, 2, ..., n$$

where $\boldsymbol{T}^0$, $\boldsymbol{U}^0$, and $\boldsymbol{c}^0$ are given. The stabilization residual function, Equation (3.130), is excluded from this derivation for brevity. The residual functions are now considered dependent on a set of discrete values, $\boldsymbol{\phi}$, that define the current design. An optimization problem is stated as:

$$\min_{\boldsymbol{\phi}} f(\boldsymbol{\phi}) \tag{3.160}$$

subject to constraint Equations (3.157) through (3.159). The objective function is in the form:

$$f(\boldsymbol{\phi}) = F\left(\boldsymbol{\phi}, \boldsymbol{T}^1(\boldsymbol{\phi}), \boldsymbol{T}^2(\boldsymbol{\phi}), ..., \boldsymbol{T}^n(\boldsymbol{\phi}), \boldsymbol{U}^1(\boldsymbol{\phi}), \boldsymbol{U}^2(\boldsymbol{\phi}), ..., \boldsymbol{U}^n(\boldsymbol{\phi}), \boldsymbol{c}^1(\boldsymbol{\phi}), \boldsymbol{c}^2(\boldsymbol{\phi}), ..., \boldsymbol{c}^n(\boldsymbol{\phi}) \right) \tag{3.161}$$

Define a Lagrangian function:

$$\hat{f} = f(\boldsymbol{\phi}) + \sum_{k=1}^{n} \boldsymbol{\gamma}^{k^T} \boldsymbol{Q}^k + \sum_{k=1}^{n} \boldsymbol{\lambda}^{n^T} \boldsymbol{R}^k + \sum_{k=1}^{n} \boldsymbol{\mu}^{n^T} \boldsymbol{H}^k \tag{3.162}$$

and compute the total derivative of $\hat{f}$ using the chain rule:

$$
\begin{aligned}
\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} &+ \sum_{k=1}^{n} \left(\frac{\partial F}{\partial \boldsymbol{T}^k}\frac{\partial \boldsymbol{T}^k}{\partial \boldsymbol{\phi}} + \frac{\partial F}{\partial \boldsymbol{U}^k}\frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} + \frac{\partial F}{\partial \boldsymbol{c}^k}\frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\gamma}^{k^T} \left(\frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{T}^k}\frac{\partial \boldsymbol{T}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{T}^{k-1}}\frac{\partial \boldsymbol{T}^{k-1}}{\partial \boldsymbol{\phi}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\lambda}^{k^T} \left(\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{T}^k}\frac{\partial \boldsymbol{T}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{T}^{k-1}}\frac{\partial \boldsymbol{T}^{k-1}}{\partial \boldsymbol{\phi}} \right. \\
&\left. + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k}\frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^{k-1}}\frac{\partial \boldsymbol{U}^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^k}\frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^{k-1}}\frac{\partial \boldsymbol{c}^{k-1}}{\partial \boldsymbol{\phi}} \right) \\
&+ \sum_{k=1}^{n} \boldsymbol{\mu}^{k^T} \left(\frac{\partial \boldsymbol{H}_l^n}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{T}^k}\frac{\partial \boldsymbol{T}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{T}^{k-1}}\frac{\partial \boldsymbol{T}^{k-1}}{\partial \boldsymbol{\phi}} \right. \\
&\left. + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k}\frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^{k-1}}\frac{\partial \boldsymbol{U}^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k}\frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^{k-1}}\frac{\partial \boldsymbol{c}^{k-1}}{\partial \boldsymbol{\phi}} \right)
\end{aligned} \tag{3.163}
$$

Rearranging terms yields:

$$
\begin{aligned}
\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} &+ \sum_{k=1}^{n} \left(\boldsymbol{\gamma}^{kT}\frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\lambda}^{kT}\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\mu}^{kT}\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\phi}} \right) \\
&+ \left(\frac{\partial F}{\partial \boldsymbol{U}^n} + \boldsymbol{\lambda}^{nT}\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n} + \boldsymbol{\mu}^{nT}\frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n} \right)\frac{\partial \boldsymbol{U}^n}{\partial \boldsymbol{\phi}} \\
&+ \left(\frac{\partial F}{\partial \boldsymbol{c}^n} + \boldsymbol{\lambda}^{nT}\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{c}^n} + \boldsymbol{\mu}^{nT}\frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n} \right)\frac{\partial \boldsymbol{c}^n}{\partial \boldsymbol{\phi}} \\
&+ \left(\frac{\partial F}{\partial \boldsymbol{T}^n} + \boldsymbol{\gamma}^{nT}\frac{\partial \boldsymbol{Q}^n}{\partial \boldsymbol{T}^n} + \boldsymbol{\lambda}^{nT}\frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{T}^n} + \boldsymbol{\mu}^{nT}\frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{T}^n} \right)\frac{\partial \boldsymbol{T}^n}{\partial \boldsymbol{\phi}} \\
&+ \sum_{k=1}^{n-1} \left(\frac{\partial F}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{k+1^T}\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{k+1^T}\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{kT}\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{nT}\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} \right)\frac{\partial \boldsymbol{U}^k}{\partial \boldsymbol{\phi}} \\
&+ \sum_{k=1}^{n-1} \left(\frac{\partial F}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{k+1^T}\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{k+1^T}\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{kT}\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{nT}\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} \right)\frac{\partial \boldsymbol{c}^k}{\partial \boldsymbol{\phi}} \\
&+ \sum_{k=1}^{n-1} \left(\frac{\partial F}{\partial \boldsymbol{T}^k} + \boldsymbol{\gamma}^{k+1^T}\frac{\partial \boldsymbol{Q}^{k+1}}{\partial \boldsymbol{T}^k} + \boldsymbol{\lambda}^{k+1^T}\frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{T}^k} + \boldsymbol{\mu}^{k+1^T}\frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{T}^k} \right. \\
&\left. + \boldsymbol{\gamma}^{kT}\frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{T}^k} + \boldsymbol{\lambda}^{kT}\frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{T}^k} + \boldsymbol{\mu}^{nT}\frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{T}^k} \right)\frac{\partial \boldsymbol{T}^k}{\partial \boldsymbol{\phi}}
\end{aligned} \tag{3.164}
$$

which defines the adjoint problems:

Final step:

$$\frac{\partial F}{\partial \boldsymbol{U}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{U}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{U}^n} = 0 \tag{3.165}$$

$$\frac{\partial F}{\partial \boldsymbol{c}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{c}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{c}^n} = 0 \tag{3.166}$$

$$\frac{\partial F}{\partial \boldsymbol{T}^n} + \boldsymbol{\gamma}^{nT} \frac{\partial \boldsymbol{Q}^n}{\partial \boldsymbol{T}^n} + \boldsymbol{\lambda}^{nT} \frac{\partial \boldsymbol{R}^n}{\partial \boldsymbol{T}^n} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^n}{\partial \boldsymbol{T}^n} = 0 \tag{3.167}$$

k^{th} step:

$$\frac{\partial F}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{k+1^T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{k+1^T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{U}^k} + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{U}^k} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{U}^k} = 0 \tag{3.168}$$

$$\frac{\partial F}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{k+1^T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{k+1^T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{c}^k} + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{c}^k} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{c}^k} = 0 \tag{3.169}$$

$$\begin{aligned}&\frac{\partial F}{\partial \boldsymbol{T}^k} + \boldsymbol{\gamma}^{k+1^T} \frac{\partial \boldsymbol{Q}^{k+1}}{\partial \boldsymbol{T}^k} + \boldsymbol{\lambda}^{k+1^T} \frac{\partial \boldsymbol{R}^{k+1}}{\partial \boldsymbol{T}^k} + \boldsymbol{\mu}^{k+1^T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{T}^k} \\ &+ \boldsymbol{\gamma}^{kT} \frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{T}^k} + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{T}^k} + \boldsymbol{\mu}^{nT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{T}^k} = 0\end{aligned} \tag{3.170}$$

$$k = n-1, ..., 1, 2$$

The Lagrange multipliers, $\boldsymbol{\lambda}^k$, $\boldsymbol{\mu}^k$, and $\boldsymbol{\gamma}^k$ for $k = 1, ..., n$, are computed using Equations (3.165) to (3.170), then the total derivative of the objective is:

$$\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} + \sum_{k=1}^{n} \left(\boldsymbol{\gamma}^{kT} \frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\lambda}^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\mu}^{kT} \frac{\partial \boldsymbol{H}^k}{\partial \boldsymbol{\phi}} \right) \tag{3.171}$$

3.6. Topology optimization with process objectives

The powder-bed build process is approximated as successive layer depositions where the layer material is added instantaneously. The temperature value on the supernode(s) for each deposited layer is computed to match the total thermal energy in the added material. The residual equations are solved after each layer deposition:

$$\boldsymbol{Q}_j^k \left(\boldsymbol{\phi}, \boldsymbol{T}_j^k, \boldsymbol{T}_j^{k-1} \right) = 0 \tag{3.172}$$

$$\boldsymbol{R}_j^k \left(\boldsymbol{\phi}, \boldsymbol{T}_j^k, \boldsymbol{T}_j^{k-1}, \boldsymbol{U}_j^k, \boldsymbol{U}_j^{k-1}, \boldsymbol{c}_j^k, \boldsymbol{c}_j^{k-1} \right) = 0 \tag{3.173}$$

$$\boldsymbol{H}_j^k \left(\boldsymbol{\phi}, \boldsymbol{T}_j^k, \boldsymbol{T}_j^{k-1}, \boldsymbol{U}_j^k, \boldsymbol{U}_j^{k-1}, \boldsymbol{c}_j^k, \boldsymbol{c}_j^{k-1} \right) = 0 \tag{3.174}$$

$$k = 1, 2, ..., n \tag{3.175}$$

$$j = 1, 2, ..., l \tag{3.176}$$

$$\boldsymbol{T}_j^0 = \boldsymbol{T}_{j-1}^n \cup \tilde{\boldsymbol{T}}_j \tag{3.177}$$

where n is number of steps required to integrate between layer deposit times, t^{j-1} and t^{j}, l is the number of layers, and $\tilde{\boldsymbol{T}}_j$ is the temperature of the supernodes added during the deposition of layer j.

The objective function takes the form:

$$\begin{aligned}
f(\boldsymbol{\phi}) &= F\left(\boldsymbol{\phi}, \mathscr{T}(\boldsymbol{\phi}), \mathscr{U}(\boldsymbol{\phi}), \mathscr{C}(\boldsymbol{\phi})\right) && (3.178)\\
\mathscr{T}(\boldsymbol{\phi}) &= \left\{ \boldsymbol{T}_j^k(\boldsymbol{\phi}) \mid k = 1,2,...,n; \;\; j = 1,2,...,l \right\} && (3.179)\\
\mathscr{U}(\boldsymbol{\phi}) &= \left\{ \boldsymbol{U}_j^k(\boldsymbol{\phi}) \mid k = 1,2,...,n; \;\; j = 1,2,...,l \right\} && (3.180)\\
\mathscr{C}(\boldsymbol{\phi}) &= \left\{ \boldsymbol{c}_j^k(\boldsymbol{\phi}) \;\; \mid k = 1,2,...,n; \;\; j = 1,2,...,l \right\} && (3.181)
\end{aligned}$$

Define a Lagrangian function:

$$\hat{f} = f(\boldsymbol{\phi}) + \sum_{k=1}^{n} \sum_{j=1}^{l} \boldsymbol{\gamma}_j^{k^T} \boldsymbol{Q}_j^k + \sum_{k=1}^{n} \sum_{j=1}^{l} \boldsymbol{\lambda}_j^{k^T} \boldsymbol{R}_j^k + \sum_{k=1}^{n} \sum_{j=1}^{l} \boldsymbol{\mu}_j^{k^T} \boldsymbol{H}_j^k \tag{3.182}$$

and compute the total derivative of $\hat{f}$ using the chain rule:

$$\begin{aligned}
\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} &+ \sum_{k=1}^{n} \sum_{j=1}^{l} \left(\frac{\partial F}{\partial \boldsymbol{T}_j^k} \frac{\partial \boldsymbol{T}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial F}{\partial \boldsymbol{U}_j^k} \frac{\partial \boldsymbol{U}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial F}{\partial \boldsymbol{c}_j^k} \frac{\partial \boldsymbol{c}_j^k}{\partial \boldsymbol{\phi}} \right)\\
&+ \sum_{k=1}^{n} \sum_{j=1}^{l} \boldsymbol{\gamma}_j^{k^T} \left(\frac{\partial \boldsymbol{Q}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{Q}_j^k}{\partial \boldsymbol{T}_j^k} \frac{\partial \boldsymbol{T}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{Q}_j^k}{\partial \boldsymbol{T}_j^{k-1}} \frac{\partial \boldsymbol{T}_j^{k-1}}{\partial \boldsymbol{\phi}} \right)\\
&+ \sum_{k=1}^{n} \sum_{j=1}^{l} \boldsymbol{\lambda}_j^{k^T} \left(\frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{T}_j^k} \frac{\partial \boldsymbol{T}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{T}_j^{k-1}} \frac{\partial \boldsymbol{T}_j^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{U}_j^k} \frac{\partial \boldsymbol{U}_j^k}{\partial \boldsymbol{\phi}} \right.\\
&+ \left. \frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{U}_j^{k-1}} \frac{\partial \boldsymbol{U}_j^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{c}_j^k} \frac{\partial \boldsymbol{c}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{c}_j^{k-1}} \frac{\partial \boldsymbol{c}_j^{k-1}}{\partial \boldsymbol{\phi}} \right)\\
&+ \sum_{k=1}^{n} \sum_{j=1}^{l} \boldsymbol{\lambda}_j^{k^T} \left(\frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{T}_j^k} \frac{\partial \boldsymbol{T}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{T}_j^{k-1}} \frac{\partial \boldsymbol{T}_j^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{U}_j^k} \frac{\partial \boldsymbol{U}_j^k}{\partial \boldsymbol{\phi}} \right.\\
&+ \left. \frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{U}_j^{k-1}} \frac{\partial \boldsymbol{U}_j^{k-1}}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{c}_j^k} \frac{\partial \boldsymbol{c}_j^k}{\partial \boldsymbol{\phi}} + \frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{c}_j^{k-1}} \frac{\partial \boldsymbol{c}_j^{k-1}}{\partial \boldsymbol{\phi}} \right) && (3.183)
\end{aligned}$$

Rearranging terms yields:

$$
\begin{aligned}
\frac{d\hat{f}}{d\boldsymbol{\phi}} = {} & \frac{\partial F}{\partial \boldsymbol{\phi}} + \sum_{k=1}^{n}\sum_{j=1}^{l}\left(\boldsymbol{\gamma}_l^{kT}\frac{\partial \boldsymbol{Q}_l^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\lambda}_l^{kT}\frac{\partial \boldsymbol{R}_l^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\mu}_l^{kT}\frac{\partial \boldsymbol{H}_l^k}{\partial \boldsymbol{\phi}}\right) \\
& + \left(\frac{\partial F}{\partial \boldsymbol{U}_l^n} + \boldsymbol{\lambda}_l^{nT}\frac{\partial \boldsymbol{R}_l^n}{\partial \boldsymbol{U}_l^n} + \boldsymbol{\mu}_l^{nT}\frac{\partial \boldsymbol{H}_l^n}{\partial \boldsymbol{U}_l^n}\right)\frac{\partial \boldsymbol{U}_l^n}{\partial \boldsymbol{\phi}} \\
& + \left(\frac{\partial F}{\partial \boldsymbol{c}_l^n} + \boldsymbol{\lambda}_l^{nT}\frac{\partial \boldsymbol{R}_l^n}{\partial \boldsymbol{c}_l^n} + \boldsymbol{\mu}_l^{nT}\frac{\partial \boldsymbol{H}_l^n}{\partial \boldsymbol{c}_l^n}\right)\frac{\partial \boldsymbol{c}_l^n}{\partial \boldsymbol{\phi}} \\
& + \left(\frac{\partial F}{\partial \boldsymbol{T}_l^n} + \boldsymbol{\gamma}_l^{nT}\frac{\partial \boldsymbol{Q}_l^n}{\partial \boldsymbol{T}_l^n} + \boldsymbol{\lambda}_l^{nT}\frac{\partial \boldsymbol{R}_l^n}{\partial \boldsymbol{T}_l^n} + \boldsymbol{\mu}_l^{nT}\frac{\partial \boldsymbol{H}_l^n}{\partial \boldsymbol{T}_l^n}\right)\frac{\partial \boldsymbol{T}_l^n}{\partial \boldsymbol{\phi}} \\
& + \sum_{j=1}^{l-1}\left(\frac{\partial F}{\partial \boldsymbol{U}_j^n} + {\boldsymbol{\lambda}_{j+1}^{1}}^T\frac{\partial \boldsymbol{R}_{j+1}^1}{\partial \boldsymbol{U}_{j+1}^0} + {\boldsymbol{\mu}_{j+1}^{1}}^T\frac{\partial \boldsymbol{H}_{j+1}^1}{\partial \boldsymbol{U}_{j+1}^0} + \boldsymbol{\lambda}_j^{nT}\frac{\partial \boldsymbol{R}_j^n}{\partial \boldsymbol{U}_j^n} + \boldsymbol{\mu}_j^{nT}\frac{\partial \boldsymbol{H}_j^n}{\partial \boldsymbol{U}_j^n}\right)\frac{\partial \boldsymbol{U}_j^n}{\partial \boldsymbol{\phi}} \\
& + \sum_{k=1}^{n-1}\left(\frac{\partial F}{\partial \boldsymbol{c}_j^n} + {\boldsymbol{\lambda}_{j+1}^{1}}^T\frac{\partial \boldsymbol{R}_{j+1}^1}{\partial \boldsymbol{c}_{j+1}^0} + {\boldsymbol{\mu}_{j+1}^{1}}^T\frac{\partial \boldsymbol{H}_{j+1}^1}{\partial \boldsymbol{c}_{j+1}^0} + \boldsymbol{\lambda}_j^{nT}\frac{\partial \boldsymbol{R}_j^n}{\partial \boldsymbol{c}_j^n} + \boldsymbol{\mu}_j^{nT}\frac{\partial \boldsymbol{H}_j^n}{\partial \boldsymbol{c}^k}\right)\frac{\partial \boldsymbol{c}_j^n}{\partial \boldsymbol{\phi}} \\
& + \sum_{k=1}^{n-1}\left(\frac{\partial F}{\partial \boldsymbol{T}_j^n} + {\boldsymbol{\gamma}_{j+1}^{1}}^T\frac{\partial \boldsymbol{Q}_{j+1}^1}{\partial \boldsymbol{T}_{j+1}^0} + {\boldsymbol{\lambda}_{j+1}^{1}}^T\frac{\partial \boldsymbol{R}_{j+1}^1}{\partial \boldsymbol{T}_{j+1}^0} + {\boldsymbol{\mu}_{j+1}^{1}}^T\frac{\partial \boldsymbol{H}_{j+1}^1}{\partial \boldsymbol{T}_{j+1}^0}\right. \\
& \left. + \boldsymbol{\gamma}_j^{nT}\frac{\partial \boldsymbol{Q}_j^n}{\partial \boldsymbol{T}_j^n} + \boldsymbol{\lambda}_j^{nT}\frac{\partial \boldsymbol{R}_j^n}{\partial \boldsymbol{T}_j^n} + \boldsymbol{\mu}_j^{nT}\frac{\partial \boldsymbol{H}_j^n}{\partial \boldsymbol{T}_j^n}\right)\frac{\partial \boldsymbol{T}_j^n}{\partial \boldsymbol{\phi}} \\
& + \sum_{k=1}^{n-1}\sum_{j=1}^{l-1}\left(\frac{\partial F}{\partial \boldsymbol{U}_j^k} + \boldsymbol{\lambda}_j^{k+1T}\frac{\partial \boldsymbol{R}_j^{k+1}}{\partial \boldsymbol{U}_j^k} + \boldsymbol{\mu}_j^{k+1T}\frac{\partial \boldsymbol{H}_j^{k+1}}{\partial \boldsymbol{U}_j^k} + \boldsymbol{\lambda}_j^{kT}\frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{U}_j^k} + \boldsymbol{\mu}_j^{kT}\frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{U}_j^k}\right)\frac{\partial \boldsymbol{U}_j^k}{\partial \boldsymbol{\phi}} \\
& + \sum_{k=1}^{n-1}\sum_{j=1}^{l-1}\left(\frac{\partial F}{\partial \boldsymbol{c}_j^k} + \boldsymbol{\lambda}_j^{k+1T}\frac{\partial \boldsymbol{R}_j^{k+1}}{\partial \boldsymbol{c}_j^k} + \boldsymbol{\mu}_j^{k+1T}\frac{\partial \boldsymbol{H}_j^{k+1}}{\partial \boldsymbol{c}_j^k} + \boldsymbol{\lambda}_j^{kT}\frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{c}_j^k} + \boldsymbol{\mu}_j^{kT}\frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{c}_j^k}\right)\frac{\partial \boldsymbol{c}_j^k}{\partial \boldsymbol{\phi}} \\
& + \sum_{k=1}^{n-1}\sum_{j=1}^{l-1}\left(\frac{\partial F}{\partial \boldsymbol{T}_j^k} + \boldsymbol{\gamma}_j^{k+1T}\frac{\partial \boldsymbol{Q}_j^{k+1}}{\partial \boldsymbol{T}_j^k} + \boldsymbol{\lambda}_j^{k+1T}\frac{\partial \boldsymbol{R}_j^{k+1}}{\partial \boldsymbol{T}_j^k} + \boldsymbol{\mu}_j^{k+1T}\frac{\partial \boldsymbol{H}_j^{k+1}}{\partial \boldsymbol{T}_j^k}\right. \\
& \left. + \boldsymbol{\gamma}_j^{kT}\frac{\partial \boldsymbol{Q}_j^k}{\partial \boldsymbol{T}_j^k} + \boldsymbol{\lambda}_j^{kT}\frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{T}_j^k} + \boldsymbol{\mu}_j^{kT}\frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{T}_j^k}\right)\frac{\partial \boldsymbol{T}_j^k}{\partial \boldsymbol{\phi}} \qquad (3.184)
\end{aligned}
$$

which defines the adjoint problems:

Final layer:

$$\frac{\partial F}{\partial \boldsymbol{U}_l^n} + \boldsymbol{\lambda}_l^{nT} \frac{\partial \boldsymbol{R}_l^n}{\partial \boldsymbol{U}_l^n} + \boldsymbol{\mu}_l^{nT} \frac{\partial \boldsymbol{H}_l^n}{\partial \boldsymbol{U}_l^n} = 0 \tag{3.185}$$

$$\frac{\partial F}{\partial \boldsymbol{c}_l^n} + \boldsymbol{\lambda}_l^{nT} \frac{\partial \boldsymbol{R}_l^n}{\partial \boldsymbol{c}_l^n} + \boldsymbol{\mu}_l^{nT} \frac{\partial \boldsymbol{H}_l^n}{\partial \boldsymbol{c}_l^n} = 0 \tag{3.186}$$

$$\frac{\partial F}{\partial \boldsymbol{T}_l^n} + \boldsymbol{\gamma}_l^{nT} \frac{\partial \boldsymbol{Q}_l^n}{\partial \boldsymbol{T}_l^n} + \boldsymbol{\lambda}_l^{nT} \frac{\partial \boldsymbol{R}_l^n}{\partial \boldsymbol{T}_l^n} + \boldsymbol{\mu}_l^{nT} \frac{\partial \boldsymbol{H}_l^n}{\partial \boldsymbol{T}_l^n} = 0 \tag{3.187}$$

for $k = n-1, ..., 2, 1:$

$$\frac{\partial F}{\partial \boldsymbol{U}_l^k} + \boldsymbol{\lambda}_n^{k+1^T} \frac{\partial \boldsymbol{R}_n^{k+1}}{\partial \boldsymbol{U}_n^k} + \boldsymbol{\mu}_n^{k+1^T} \frac{\partial \boldsymbol{H}_n^{k+1}}{\partial \boldsymbol{U}_n^k} + \boldsymbol{\lambda}_n^{kT} \frac{\partial \boldsymbol{R}_n^k}{\partial \boldsymbol{U}_n^k} + \boldsymbol{\mu}_n^{kT} \frac{\partial \boldsymbol{H}_n^k}{\partial \boldsymbol{U}_n^k} = 0 \tag{3.188}$$

$$\frac{\partial F}{\partial \boldsymbol{c}_n^k} + \boldsymbol{\lambda}_n^{k+1^T} \frac{\partial \boldsymbol{R}_n^{k+1}}{\partial \boldsymbol{c}_n^k} + \boldsymbol{\mu}_n^{k+1^T} \frac{\partial \boldsymbol{H}_n^{k+1}}{\partial \boldsymbol{c}_n^k} + \boldsymbol{\lambda}_n^{kT} \frac{\partial \boldsymbol{R}_n^k}{\partial \boldsymbol{c}_n^k} + \boldsymbol{\mu}_n^{kT} \frac{\partial \boldsymbol{H}_n^k}{\partial \boldsymbol{c}_n^k} = 0 \tag{3.189}$$

$$\begin{aligned} &\frac{\partial F}{\partial \boldsymbol{T}_n^k} + \boldsymbol{\gamma}_n^{k+1^T} \frac{\partial \boldsymbol{Q}_n^{k+1}}{\partial \boldsymbol{T}_n^k} + \boldsymbol{\lambda}_n^{k+1^T} \frac{\partial \boldsymbol{R}_n^{k+1}}{\partial \boldsymbol{T}_n^k} + \boldsymbol{\mu}_n^{k+1^T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{T}_n^k} \\ &+ \boldsymbol{\gamma}_n^{kT} \frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{T}_n^k} + \boldsymbol{\lambda}_n^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{T}_n^k} + \boldsymbol{\mu}_n^{kT} \frac{\partial \boldsymbol{H}_n^k}{\partial \boldsymbol{T}_n^k} = 0 \end{aligned} \tag{3.190}$$

for $\boldsymbol{j = \; l - 1, ..., 2, 1:}$

$$\frac{\partial F}{\partial \boldsymbol{U}_j^n} + \boldsymbol{\lambda}_{j+1}^{1}{}^T \frac{\partial \boldsymbol{R}_{j+1}^1}{\partial \boldsymbol{U}_{j+1}^0} + \boldsymbol{\mu}_{j+1}^{1}{}^T \frac{\partial \boldsymbol{H}_{j+1}^1}{\partial \boldsymbol{U}_{j+1}^0} + \boldsymbol{\lambda}_j^{nT} \frac{\partial \boldsymbol{R}_j^n}{\partial \boldsymbol{U}_j^n} + \boldsymbol{\mu}_j^{nT} \frac{\partial \boldsymbol{H}_j^n}{\partial \boldsymbol{U}_j^n} = 0 \tag{3.191}$$

$$\frac{\partial F}{\partial \boldsymbol{c}_j^n} + \boldsymbol{\lambda}_{j+1}^{1}{}^T \frac{\partial \boldsymbol{R}_{j+1}^1}{\partial \boldsymbol{c}_{j+1}^0} + \boldsymbol{\mu}_{j+1}^{1}{}^T \frac{\partial \boldsymbol{H}_{j+1}^1}{\partial \boldsymbol{c}_{j+1}^0} + \boldsymbol{\lambda}_j^{nT} \frac{\partial \boldsymbol{R}_j^n}{\partial \boldsymbol{c}_j^n} + \boldsymbol{\mu}_j^{nT} \frac{\partial \boldsymbol{H}_j^n}{\partial \boldsymbol{c}^k} = 0 \tag{3.192}$$

$$\frac{\partial F}{\partial \boldsymbol{T}_j^n} + \boldsymbol{\gamma}_{j+1}^{1}{}^T \frac{\partial \boldsymbol{Q}_{j+1}^1}{\partial \boldsymbol{T}_{j+1}^0} + \boldsymbol{\lambda}_{j+1}^{1}{}^T \frac{\partial \boldsymbol{R}_{j+1}^1}{\partial \boldsymbol{T}_{j+1}^0} + \boldsymbol{\mu}_{j+1}^{1}{}^T \frac{\partial \boldsymbol{H}_{j+1}^1}{\partial \boldsymbol{T}_{j+1}^0} \tag{3.193}$$

$$+ \boldsymbol{\gamma}_j^{nT} \frac{\partial \boldsymbol{Q}_j^n}{\partial \boldsymbol{T}_j^n} + \boldsymbol{\lambda}_j^{nT} \frac{\partial \boldsymbol{R}_j^n}{\partial \boldsymbol{T}_j^n} + \boldsymbol{\mu}_j^{nT} \frac{\partial \boldsymbol{H}_j^n}{\partial \boldsymbol{T}_j^n} = 0 \tag{3.194}$$

for $k = n-1, ..., 2, 1:$

$$\frac{\partial F}{\partial \boldsymbol{U}_l^k} + \boldsymbol{\lambda}_l^{k+1^T} \frac{\partial \boldsymbol{R}_l^{k+1}}{\partial \boldsymbol{U}_l^k} + \boldsymbol{\mu}_l^{k+1^T} \frac{\partial \boldsymbol{H}_l^{k+1}}{\partial \boldsymbol{U}_l^k} + \boldsymbol{\lambda}_l^{kT} \frac{\partial \boldsymbol{R}_l^k}{\partial \boldsymbol{U}_l^k} + \boldsymbol{\mu}_l^{kT} \frac{\partial \boldsymbol{H}_l^k}{\partial \boldsymbol{U}_l^k} = 0 \tag{3.195}$$

$$\frac{\partial F}{\partial \boldsymbol{c}_l^k} + \boldsymbol{\lambda}_l^{k+1^T} \frac{\partial \boldsymbol{R}_l^{k+1}}{\partial \boldsymbol{c}_l^k} + \boldsymbol{\mu}_l^{k+1^T} \frac{\partial \boldsymbol{H}_l^{k+1}}{\partial \boldsymbol{c}_l^k} + \boldsymbol{\lambda}_l^{kT} \frac{\partial \boldsymbol{R}_l^k}{\partial \boldsymbol{c}_l^k} + \boldsymbol{\mu}_l^{kT} \frac{\partial \boldsymbol{H}_l^k}{\partial \boldsymbol{c}_l^k} = 0 \tag{3.196}$$

$$\begin{aligned} &\frac{\partial F}{\partial \boldsymbol{T}_l^k} + \boldsymbol{\gamma}_l^{k+1^T} \frac{\partial \boldsymbol{Q}_l^{k+1}}{\partial \boldsymbol{T}_l^k} + \boldsymbol{\lambda}_l^{k+1^T} \frac{\partial \boldsymbol{R}_l^{k+1}}{\partial \boldsymbol{T}_l^k} + \boldsymbol{\mu}_l^{k+1^T} \frac{\partial \boldsymbol{H}^{k+1}}{\partial \boldsymbol{T}_l^k} \\ &+ \boldsymbol{\gamma}_l^{kT} \frac{\partial \boldsymbol{Q}^k}{\partial \boldsymbol{T}_l^k} + \boldsymbol{\lambda}_l^{kT} \frac{\partial \boldsymbol{R}^k}{\partial \boldsymbol{T}_l^k} + \boldsymbol{\mu}_l^{kT} \frac{\partial \boldsymbol{H}_l^k}{\partial \boldsymbol{T}_l^k} = 0 \end{aligned} \tag{3.197}$$

The Lagrange multipliers, $\boldsymbol{\lambda}_j^k$, $\boldsymbol{\mu}_j^k$, and $\boldsymbol{\gamma}_j^k$ for $k = 1, ..., n;\ j = 1, ..., l$, are computed using Equations (3.185) to (3.197), then the total derivative of the objective is:

$$\frac{d\hat{f}}{d\boldsymbol{\phi}} = \frac{\partial F}{\partial \boldsymbol{\phi}} + \sum_{k=1}^{n} \sum_{j=1}^{l} \left(\boldsymbol{\gamma}_j^{k\,T} \frac{\partial \boldsymbol{Q}_j^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\lambda}_j^{k\,T} \frac{\partial \boldsymbol{R}_j^k}{\partial \boldsymbol{\phi}} + \boldsymbol{\mu}_j^{k\,T} \frac{\partial \boldsymbol{H}_j^k}{\partial \boldsymbol{\phi}} \right) \tag{3.198}$$

The optimization procedure is:

Topology Optimization with Process Objectives: Gradient Calculation

1. Forward Simulation: Given the design, $\boldsymbol{\phi}$:
   a) Compute $\boldsymbol{T}_j^k$, $\boldsymbol{U}_j^k$, and $\boldsymbol{c}_j^k$ for $k = 1, 2, ..., n$ and $j = 1, 2, ..., l$ using Equations (3.172) to (3.177).
2. Backward Simulation
   a) Final build layer (layer l)
      i. Compute $\boldsymbol{\lambda}_l^n$, $\boldsymbol{\mu}_l^n$, and $\boldsymbol{\gamma}_l^n$, using Equations (3.185) to (3.187).
      ii. Compute $\boldsymbol{\lambda}_l^k$, $\boldsymbol{\mu}_l^k$, and $\boldsymbol{\gamma}_l^k$, for $k = n-1, ..., 2, 1$ using Equations (3.188) to (3.190).
   b) For each previous build layer, $j = l-1, ..., 2, 1$
      i. Compute $\boldsymbol{\lambda}_j^n$, $\boldsymbol{\mu}_j^n$, and $\boldsymbol{\gamma}_j^n$, using Equations (3.191) to (3.194).
      ii. Compute $\boldsymbol{\lambda}_j^k$, $\boldsymbol{\mu}_j^k$, and $\boldsymbol{\gamma}_j^k$, for $k = n-1, ..., 2, 1$ using Equations (3.195) to (3.197).
   c) Compute the gradient using Equation (3.198)

REFERENCES

[1] Ryan Alberdi, Guodong Zhang, Lei Li, and Kapil Khandelwal. A unified framework for nonlinear path-dependent sensitivity analysis in topology optimization. *International Journal for Numerical Methods in Engineering*, 115(1), 2018.

[2] Grégoire Allaire, Charles Dapogny, and Pascal Frey. A mesh evolution algorithm based on the level set method for geometry and topology optimization. *Structural and Multidisciplinary Optimization*, 48(4):711–715, 2013.

[3] Grégoire Allaire, Frédéric De Gournay, François Jouve, and Anca-Maria Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control and cybernetics*, 34(1):59, 2005.

[4] Grégoire Allaire, François Jouve, and Anca-Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of computational physics*, 194(1):363–393, 2004.

[5] R Ansola, J Canales, JA Tarrago, and J Rasmussen. On simultaneous shape and material layout optimization of shell structures. *Structural and multidisciplinary optimization*, 24(3):175–184, 2002.

[6] Hideyuki Azegami and Kenzen Takeuchi. A smoothing method for shape optimization: traction method using the robin condition. *International journal of computational methods*, 3(01):21–33, 2006.

[7] Hideyuki Azegami and Zhi Chang Wu. Domain optimization analysis in linear elastic problems: approach using traction method. *JSME international journal. Ser. A, Mechanics and material engineering*, 39(2):272–278, 1996.

[8] Jorge L Barrera, Markus J Geiss, and Kurt Maute. Hole seeding in level set topology optimization via density fields. *Structural and Multidisciplinary Optimization*, 61(4):1319–1343, 2020.

[9] Jamison L. Bartlett and Xiaodong Li. An overview of residual stresses in metal powder bed fusion. *Additive Manufacturing*, 27:131–149, MAY 2019.

[10] T Belytschko, SP Xiao, and C Parimi. Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering*, 57(8):1177–1196, 2003.

[11] Martin P Bendsøe and Ole Sigmund. Material interpolation schemes in topology optimization. *Archive of applied mechanics*, 69(9):635–654, 1999.

[12] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*, 71(2):197–224, 1988.

[13] Martin Philip Bendsoe and Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.

[14] Kai-Uwe Bletzinger and Kurt Maute. Towards generalized shape and topology optimization. *Engineering Optimization*, 29(1-4):201–216, 1997.

[15] Blaise Bourdin. Filters in topology optimization. *International journal for numerical methods in engineering*, 50(9):2143–2158, 2001.

[16] Vincent Braibant and Claude Fleury. Shape optimal design using b-splines. *Computer methods in applied mechanics and engineering*, 44(3):247–267, 1984.

[17] Tyler E Bruns and Daniel A Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer methods in applied mechanics and engineering*, 190(26-27):3443–3459, 2001.

[18] Martin Burger, Benjamin Hackl, and Wolfgang Ring. Incorporating topological derivatives into level set methods. *Journal of computational physics*, 194(1):344–362, 2004.

[19] M Chiumenti, Q Valverde, C Agelet de Saracibar, and M Cervera. A stabilized formulation for incompressible plasticity using linear triangles and tetrahedra. *International Journal of Plasticity*, 20(8):1487–1504, 2004.

[20] Asger Nyman Christiansen, Morten Nobel-Jørgensen, Niels Aage, Ole Sigmund, and Jakob Andreas Bærentzen. Topology optimization using an explicit interface representation. *Structural and Multidisciplinary Optimization*, 49(3):387–399, 2014.

[21] John Dannenhoffer and Robert Haimes. *Design Sensitivity Calculations Directly on CAD-based Geometry*. 2015.

[22] A.J. Dunbar, E.R. Denlinger, J. Heigel, P. Michaleris, P. Guerrier, R. Martukanitz, and T.W. Simpson. Development of experimental method for in situ distortion and temperature measurements during the laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, 12:25–30, 2016.

[23] Hans A Eschenauer, Vladimir V Kobelev, and Axel Schumacher. Bubble method for topology and shape optimization of structures. *Structural optimization*, 8(1):42–51, 1994.

[24] M.M. Francois, A. Sun, W.E. King, N.J. Henson, D. Tourret, C.A. Bronkhorst, N.N. Carlson, C.K. Newman, T. Haut, J. Bakosi, J.W. Gibbs, V. Livescu, S.A. Vander Wiel, A.J. Clarke, M.W. Schraad, T. Blacker, H. Lim, T. Rodgers, S. Owen, F. Abdeljawad, J. Madison, A.T. Anderson, J-L. Fattebert, R.M. Ferencz, N.E. Hodge, S.A. Khairallah, and O. Walton. Modeling of additive manufacturing processes for metals: Challenges and opportunities. *Current Opinion in Solid State and Materials Science*, 21(4):198–206, 2017.

[25] James K Guest, Jean H Prévost, and Ted Belytschko. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International journal for numerical methods in engineering*, 61(2):238–254, 2004.

[26] Seung-Hyun Ha and Seonho Cho. Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. *Computers & structures*, 86(13-14):1447–1455, 2008.

[27] Raphael T Haftka and Ramana V Grandhi. Structural shape optimizationâĂŤa survey. *Computer methods in applied mechanics and engineering*, 57(1):91–106, 1986.

[28] Robert Haimes and John Dannenhoffer. *The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry*. 2013.

[29] Robert Haimes and Mark Drela. *On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design*. 2012.

[30] Behrooz Hassani, Seyed Mehdi Tavakkoli, and Hossein Ghasemnejad. Simultaneous shape and topology optimization of shell structures. *Structural and Multidisciplinary Optimization*, 48(1):221–233, 2013.

[31] Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.

[32] In Gwun Jang and Byung Man Kwak. Evolutionary topology optimization using design space adjustment based on fixed grid. *International journal for numerical methods in engineering*, 66(11):1817–1840, 2006.

[33] In Gwun Jang and Byung Man Kwak. Design space optimization using design space adjustment and refinement. *Structural and Multidisciplinary optimization*, 35(1):41–54, 2008.

[34] Bradley H. Jared, Miguel A. Aguilo, Lauren L. Beghini, Brad L. Boyce, Brett W. Clark, Adam Cook, Bryan J. Kaehr, and Joshua Robbins. Additive manufacturing: Toward holistic design. *Scripta Materialia*, 135:141–147, 2017.

[35] Il Yong Kim and Byung Man Kwak. Design space optimization using a numerical design continuation method. *International Journal for Numerical Methods in Engineering*, 53(8):1979–2002, 2002.

[36] W. E. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, S. A. Khairallah, and A. M. Rubenchik. Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges. *Applied Physics Reviews*, 2(4), 2015.

[37] Sebastian Kreissl and Kurt Maute. Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimization*, 46(3):311–326, 2012.

[38] Boyan Stefanov Lazarov and Ole Sigmund. Filters in topology optimization based on helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86(6):765–781, 2011.

[39] Haojie Lian, Asger N Christiansen, Daniel A Tortorelli, Ole Sigmund, and Niels Aage. Combined shape and topology optimization for minimization of maximal von mises stress. *Structural and Multidisciplinary Optimization*, 55(5):1541–1557, 2017.

[40] Xuan Liang, Lin Cheng, Qian Chen, Qingcheng Yang, and Albert C. To. A modified method for estimating inherent strains from detailed process simulation for fast residual distortion prediction of single-walled structures fabricated by directed energy deposition. *Additive Manufacturing*, 23:471–486, 2018.

[41] ML Luchi, A Poggialini, and F Persiani. An interactive optimization procedure applied to the design of gas turbine discs. *Computers & Structures*, 11(6):629–637, 1980.

[42] K Maute and E Ramm. Adaptive topology optimization. *Structural optimization*, 10(2):100–112, 1995.

[43] Kurt Maute, Stefan Schwarz, and Ekkehard Ramm. Adaptive topology optimization of elastoplastic structures. *Structural optimization*, 15(2):81–91, 1998.

[44] John O Milewski. *Additive Manufacturing of Metals*. Springer, 2017.

[45] Grzegorz Misiun, Emiel van de Ven, Matthijs Langelaar, Hubert Geijselaers, Fred van Keulen, Ton van den Boogaard, and Can Ayas. Topology optimization for additive manufacturing with distortion constraints. *Computer Methods in Applied Mechanics and Engineering*, 386, 2021.

[46] Stephen T. Montgomery and Patrick F. Chavez. Basic equations and solution method for the calculation of the transient electromechanical response of dielectric devices. Technical Report SAND1986-0755, Sandia National Laboratories, Albuquerque, New Mexico, June 1986.

[47] Tuan T Nguyen, J Andreas Bærentzen, Ole Sigmund, and Niels Aage. Efficient hybrid topology and shape optimization combining implicit and explicit design representations. *Structural and Multidisciplinary Optimization*, 62(3):1061–1069, 2020.

[48] Open CASCADE SAS. Opencascade.

[49] Eric Phipps and Roger Pawlowski. Efficient expression templates for operator overloading-based automatic differentiation. In Shaun Forth, Paul Hovland, Eric Phipps, Jean Utke, and Andrea Walther, editors, *Recent Advances in Algorithmic Differentiation*, pages 309–319, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[50] Stefan Riehl and Paul Steinmann. A staggered approach to shape and topology optimization using the traction method and an evolutionary-type advancing front algorithm. *Computer Methods in Applied Mechanics and Engineering*, 287:1–30, 2015.

[51] Joshua Robbins, Ryan Alberdi, and Brett W. Clark. Concurrent shape and topology optimization. Technical Report SAND2021-11984, Sandia National Laboratories, Albuquerque, New Mexico, September 2021.

[52] George IN Rozvany, Ming Zhou, and Torben Birker. Generalized shape optimization without homogenization. *Structural optimization*, 4(3-4):250–252, 1992.

[53] Sandia National laboratories. Plato.

[54] Ashesh Sharma and Kurt Maute. Stress-based topology optimization using spatial gradient stabilized xfem. *Structural and Multidisciplinary Optimization*, 57(1):17–38, 2018.

[55] Ole Sigmund and Kurt Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[56] Gabriel Stankiewicz, Chaitanya Dev, and Paul Steinmann. Coupled topology and shape optimization using an embedding domain discretization method. *Structural and Multidisciplinary Optimization*, pages 1–21, 2021.

[57] Ian Gibson; David Rosen; Brent Stucker. *Additive Manufacturing Technologies*. Springer, 2015.

[58] Krister Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, pages 555–573, 2002.

[59] Akihiro Takezawa, Albert C. To, Qian Chen, Xuan Liang, Florian Dugast, Xiaopeng Zhang, and Mitsuru Kitamura. Sensitivity analysis and lattice density optimization for sequential inherent strain method used in additive manufacturing process. *Computer Methods in Applied Mechanics and Engineering*, 370, 2020.

[60] Poh-Soong Tang and Kuang-Hua Chang. Integration of topology and shape optimization for design of structural components. *Structural and Multidisciplinary Optimization*, 22(1):65–82, 2001.

[61] Mary Kathryn Thompson, Giovanni Moroni, Tom Vaneker, Georges Fadel, R. Ian Campbell, Ian Gibson, Alain Bernard, Joachim Schulz, Patricia Graf, Bhrigu Ahuja, and Filomeno Martina. Design for additive manufacturing: Trends, opportunities, considerations, and constraints. *CIRP Annals*, 65(2):737–760, 2016.

[62] S. Timoshenko. Analysis of bi-metal thermostats. *J. Opt. Soc. Am.*, 11(3):233–255, Sep 1925.

[63] Evangelos Tyflopoulos and Martin Steinert. Topology and parametric optimization-based design processes for lightweight structures. *Applied Sciences*, 10(13):4496, 2020.

[64] Nico P van Dijk, Kurt Maute, Matthijs Langelaar, and Fred Van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, 2013.

[65] Fengwen Wang, Boyan Stefanov Lazarov, and Ole Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43(6):767–784, 2011.

[66] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology optimization. *Computer methods in applied mechanics and engineering*, 192(1-2):227–246, 2003.

[67] Zhijun Wang, Akke SJ Suiker, Hèrm Hofmeyer, Ivo Kalkman, and Bert Blocken. Sequentially coupled gradient-based topology and domain shape optimization. *Optimization and Engineering*, pages 1–34, 2020.

[68] Zhijun Wang, Akke SJ Suiker, Herm Hofmeyer, Twan van Hooff, and Bert Blocken. Optimization of thin-walled beam structures: monolithic versus staggered solution schemes. *Thin-Walled Structures*, 159:107182, 2021.

[69] Zhijun Wang, Akke SJ Suiker, Hèrm Hofmeyer, Twan van Hooff, and Bert Blocken. Sequentially coupled shape and topology optimization for 2.5 d and 3d beam models. *Acta Mechanica*, 232(4):1683–1708, 2021.

[70] Guilian Yi and Nam H Kim. Identifying boundaries of topology optimization results using basic parametric features. *Structural and Multidisciplinary Optimization*, 55(5):1641–1654, 2017.

[71] Wei-Hong Zhang, Pierre Beckers, and Claude Fleury. A unified parametric design approach to structural shape optimization. *International Journal for Numerical Methods in Engineering*, 38(13):2283–2292, 1995.

[72] O.C. Zienkiewicz and J.S. Campbell. Shape optimization and sequential linear programming. In R.H. Gallagher and O.C. Zienkiewicz, editors, *Optimum Structural Design*, pages 109–126. John Wiley & Sons, 1973.

DISTRIBUTION

Hardcopy—External

Number of Copies	Name(s)	Company Name and Company Mailing Address

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop

Email—Internal [redacted]

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Sandia
National
Laboratories

Sandia National Laboratories
is a multimission laboratory
managed and operated by
National Technology &
Engineering Solutions of
Sandia LLC, a wholly owned
subsidiary of Honeywell
International Inc., for the U.S.
Department of Energy's
National Nuclear Security
Administration under contract
DE-NA0003525.