Large-Scale Multimaterial Topology Optimization for Additive Manufacturing

Graeme J. Kennedy*
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia

Advanced manufacturing technologies, such as additive manufacturing, give the engineer greater design freedom by removing geometric and processing constraints that are required by conventional manufacturing methods. Additive manufacturing, therefore, has the potential to enable the design and production of low-weight high-performance structures. However, optimization of such additively-manufactured structures using conventional optimization techniques, such as topology optimization, is challenging due to the demanding mesh requirements and large size of the design problem. In this paper, we address these difficulties by proposing a scalable approach for analysis and design of large-scale topology and multimaterial optimization problems. This approach includes a multigrid-preconditioned Krylov method for solving large structural finite-element problems, and a parallel interior-point optimization technique for solving large-scale constrained optimization problems. We demonstrate the proposed methods on a large-scale mass-constrained compliance minimization problem for a structure discretized using a $64 \times 64 \times 256$ element mesh, resulting in 3.26 million structural degrees of freedom, 5.24 million design variables and 1.05 million linear constraints.

I. Introduction

Aerospace vehicles utilize slender, high-aspect ratio structures with small material volume fractions to achieve low structural weight while satisfying structural performance requirements. To meet demanding weight reduction goals, aerospace engineers are increasingly turning to highly-tailored structures fabricated using novel manufacturing processes. New manufacturing technologies, such as additive manufacturing, give the engineer greater design freedom by removing geometric restrictions imposed by conventional manufacturing processes. However, the best way to parametrize such structures for design optimization remains an open question, with various authors suggesting either CAD-based design methods, or topology optimization approaches.

Topology optimization techniques offer an attractive solution to the problem of geometry parametrization for additive manufacturing, since these techniques can be used to optimize structures without constraints on structural shape or layout. However, the low volume fraction and high aspect ratios exhibited by many aerospace structures place demanding computational requirements on topology optimization methods. For instance, raster or voxel-based topology methods require near-uniform mesh spacing to resolve detailed structural design features, thereby imposing onerous meshing requirements for three-dimensional problems. Topology optimization techniques based on level-set methods can alleviate this requirement to some degree, since there is only an indirect linkage between the design parametrization and structural discretization. However, high-accuracy structural solutions for refined structural designs still require large structural models. While topology optimization of moderate size problems is now routine, large-scale topology optimization is challenging due to the difficulty of both efficiently analyzing large structural models and performing efficient large-scale optimization. These difficulties arise together, since in topology optimization the size of the design problem and analysis problems are proportional.

Many authors have developed methods for large-scale topology optimization. Borrvall and Petersson [2] developed techniques for large-scale topology optimization of 3D elastic continua using a regularized intermediate density control. They used parallel computing techniques to solve topology problems with up to 663 000 degrees of freedom. Kim et al. [18] proposed methods to handle large-scale topology optimization problems for eigenvalue-related design problems. Within their framework, the sensitivity analysis and design variable updates were performed with minimal communication amongst subdomains to achieve excellent parallel scalability. Evgrafov et al. [9] discuss the issues that arise when using a finite-element tearing and interconnecting dual-primal (FETI-DP) substructuring iterative solution method for large-scale topology optimization. In particular, the optimal domain decomposition for their solution method is design-dependent, where the best performance is obtained by splitting the domains along solid-void boundaries. More recently, Amir et al. [1] used an algebraic multigrid preconditioner to solve large 3D compliance topology optimization problems involving 400 000 degrees of freedom.

*Assistant Professor, AIAA Member

In this work, we use multimaterial parametrization techniques that require the use of sparse linear and nonlinear constraints. While these techniques have been demonstrated on moderate-scale two-dimensional problems [13, 16] using sophisticated optimization algorithms [12, 30], these design parametrization methods have not been used on large-scale topology optimization problems with millions of degrees of freedom.

In this paper, we develop methods to address the issue of scalability in topology and multimaterial optimization. In particular, we address the solution and optimization scalability as follows:

- We utilize iterative Krylov subspace methods, preconditioned using a geometric multigrid method, to solve the large linear systems that arise from the finite-element analysis and adjoint-based derivative evaluation method. This multigrid preconditioner scales well with increasing numbers of processors, and also exhibits good algorithmic scaling with increasing finite-element mesh size.
- We utilize a parallel optimization method that achieves good parallel efficiency for large-scale multimaterial optimization problems. In particular, our optimization method can handle the millions of sparse linear constraints that arise due to the multimaterial structural parametrizations.

The remainder of this paper is structured as follows: In Section II, we describe techniques for combined multimaterial and topology optimization that we will use in this work. The number of sparse constraints used by these methods scales with the size of the design and analysis problems. In Section III, we present a parallel multigrid method for analysis of large-scale structural topology and multimaterial optimization problems. In Section V, we describe some preliminary results obtained using the methods presented in this paper. In Section VI, we present preliminary conclusions from our work and expected results for the final paper.

II. Multimaterial parametrization techniques

In this section, we review discrete material optimization methods and describe the design parametrizations that we will utilize in this work.

A. Discrete material optimization

Discrete material optimization (DMO) methods, first proposed by Stegmann and Lund [28], are a generalization of penalization-based topology optimization methods for both multimaterial and lamination sequence design problems in which the lamination angles are restricted to a discrete set of allowable values. In the DMO method, the point-wise stiffness and density of the material are expressed as a sum of the product of the N candidate material stiffness and design-dependent weights, as follows:

$$\mathbf{C}(\mathbf{x}) = \sum_{j=1}^{N} w_j(\mathbf{x})\mathbf{C}_j,$$

$$\rho(\mathbf{x}) = \sum_{j=1}^{N} w_j^{\rho}(\mathbf{x})\rho_j.$$

In each component there are N design variables $\mathbf{x} \in [\varepsilon, 1]^N$, where ε is a small but finite lower bound. The goal of the DMO approach is to formulate the design-dependent weights on the stiffness, $w_j(\mathbf{x})$, and density, $w_j^{\rho}(\mathbf{x})$, such that the final solution is discrete, $x_j \in \{0,1\}$, and the design-dependent weights satisfy the properties $w_j = 1$ if $x_j = 1$ and $w_j \approx 0$ if $x_j = 0$. The weights are designed to impose a stiffness penalization that reduces the stiffness-to-mass ratio of intermediate designs, making the stiffness-to-mass ratios of the discrete designs more favorable. The precise relationship between the design variables and the material weighting factors can have a significant impact on the performance of the method, and for this reason several different schemes have been proposed [28, 14, 13, 13, 26, 27, 16].

Various authors have developed enhancements of the DMO approach. Hvejsel et al. [14] developed a technique for laminate parametrization that, in a similar manner to DMO, uses a weighted sum of contributions to the stiffness. In a departure from the original DMO approach, they employed an exact, quadratic concave penalty constraint function, first used by Borrvall and Petersson [3], to force the design towards a discrete solution. They demonstrated their approach on a series of compliance minimization problems. Taking a different approach, Hvejsel and Lund [13] extended the SIMP interpolation scheme to multimaterial selection problems, including ply-angle selection. In a departure from the original DMO method, a large number of sparse linear constraints were added to enforce a partition of unity constraint on the design variables. Kennedy and Martins [16] developed a discrete ply selection parametrization that can be used as an alternative, or in conjunction with, existing DMO methods. The technique utilizes an exact ℓ_1 penalty

function to drive intermediate designs to discrete 0-1 solutions. Recently, an extension of DMO has been developed for both material and thickness selection problems [26].

Other authors have proposed alternatives to the DMO approach for both lamination sequence design and multimaterial selection problems. Bruyneel [5] proposed the SFP parametrization technique for discrete ply-angle selection problems. The SFP approach uses bilinear shape functions to select amongst the available ply-angle options. An important advantage of the SFP approach is that it reduces the number of design variables required per discrete selection. Bruyneel et al. [6] extended the SFP approach to material selection amongst different numbers of plies by using higher-dimensional interpolation functions. More recently, Gao et al. [11] developed a bi-value coding (BCP) method based on the SFP approach that is well suited for problems with large numbers of candidate materials.

In this work, we use a recent extension of the DMO technique for discrete material and topology optimization employing a RAMP-based penalization scheme [29, 13, 26, 27], which is defined as follows:

$$\begin{aligned} \mathbf{C}(\mathbf{x}) &= \frac{x_1}{1+q(1-x_1)} \sum_{j=1}^{N} \frac{x_{j+1}}{1+q(1-x_{j+1})} \mathbf{C}_j, \\ \rho(\mathbf{x}) &= \sum_{j=1}^{N} x_1 x_{j+1} \rho_j, \\ 1 &= \sum_{j=1}^{N} x_{j+1}, \end{aligned} \tag{1}$$

where $\mathbf{x} \in [\varepsilon, 1]^{N+1}$ and q is the RAMP penalization factor, here we use $\varepsilon = 10^{-3}$ and $q = 5$. Note that the first variable represents a topology variable that has the effect of removing the stiffness contribution of the component as $x_1 \to 0$. The linear constraint in the stiffness parametrization (1) is designed to ensure that the mass and stiffness properties are interpolated reasonably at intermediate values of the design variables. An extensive discussion of this method, and similar DMO techniques, can be found in the literature; see for instance [29, 13, 16, 26, 27].

B. Discrete stiffness optimization

As an alternative to the DMO penalization technique described above, we also present a multimaterial and topology optimization approach analogous to the discrete thickness optimization approach presented by Kennedy [15]. This technique is limited to select amongst materials in which the stiffness and density are proportional. As a result, we can write the stiffness of the material as a linear function of the material density, ρ, as follows:

$$\mathbf{C}(\rho) = \rho \mathbf{C}_0 \qquad \rho \in \mathscr{D}, \tag{2}$$

where $\mathscr{D}$ is a discrete, ordered set of values such that $\mathscr{D} = \{\rho_1, \rho_2, \ldots, \rho_N\}$ and $\rho_1 < \rho_2 < \ldots < \rho_N$. In the proposed parametrization technique, we use a continuous relaxation of the original discrete problem (2), using $\rho \in [\rho_1, \rho_N]$ in place of $\rho \in \mathscr{D}$, and force the design towards a discrete solution by imposing a stiffness-to-weight penalty on intermediate designs through a series of constraints and an exact penalty function.

To construct the discrete stiffness parametrization, we utilize a continuous density variable and add an interpolation variable $\eta \in [0, 1]$. As we will show, the trace of the feasible domain cannot be parametrized by a single continuous function, so we cannot write ρ directly as a function of η. Instead, we add both ρ and η as design variables. To ensure that intermediate designs are unfavorable, we construct a series of constraints designed to impose lower stiffness-to-mass ratios than discrete designs. These constraints consist of a linear constraint and a series of $N-1$ nonlinear constraints. Together, these constraints can be written as follows:

$$\begin{aligned} &\rho \le \rho_1 + (\rho_N - \rho_1)\eta \\ &\rho \le \rho_j^c(\eta; q), \qquad\qquad j = 1, \ldots, N-1, \end{aligned} \tag{3}$$

The linear constraint ensures that the variable, ρ, remains below the interpolation line $\rho_1 + (\rho_N - \rho_1)\eta$, while the nonlinear constraints impose a penalty on the stiffness of intermediate designs that do not satisfy $\rho \in \mathscr{D}$.

The combination of the penalization constraints and the exact penalty function are illustrated in Figure 1 for the discrete densities $\mathscr{D} = \{1, 2, 4\}$ where the feasible region of the (η, ρ) space is shaded. Also illustrated is the exact penalty function which is designed to push the solution towards the constraint boundary. Note that the trace of the domain boundary of the feasible region is non-differentiable at the discrete design points. The domain boundary resembles the classical SIMP penalization between adjacent discrete values of ρ, but cannot be parametrized directly as a single differentiable function.

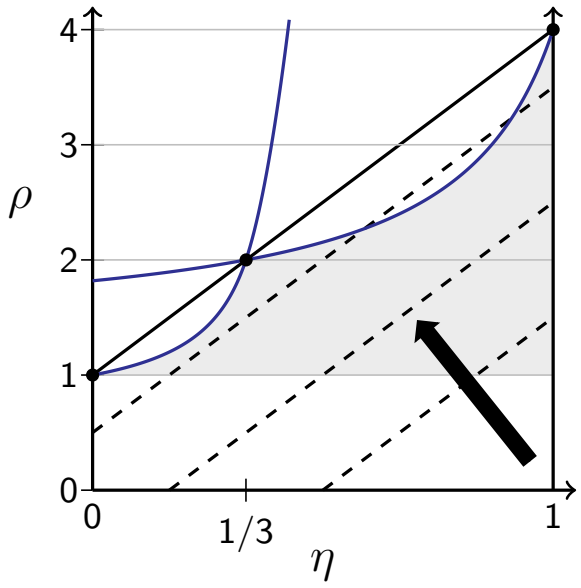

Figure 1: Illustration of the proposed penalization technique for $\mathscr{D} = 1, 2, 4$. The feasible region is shown in the gray shaded region and the contours of the exact penalization are shown by the dashed lines.

While various penalization constraint functions can be formulated, here we use the piecewise Rational Approximation of Material Properties (pRAMP) penalization technique presented by Kennedy [15]. This penalization approach is based on RAMP penalization first developed by Stolpe and Svanberg [29] as an alternative penalization to SIMP methods. A key advantage of the RAMP function is that it maintains a strictly positive derivative over its domain. The pRAMP interpolation can be written as follows:

$$\rho_j^c(\eta) = \begin{cases} \rho_j + (\rho_{j+1} - \rho_j)\dfrac{u}{1+q(1-u)} & \eta \leq \eta_{j+1} \\ \rho_{j+1} + (\rho_{j+1} - \rho_j)((q+1)(u-1) + q(q+1)(u-1)^2) & \eta > \eta_{j+1} \end{cases} \tag{4}$$

where u is defined locally as follows $u = (\eta - \eta_j)/(\eta_{j+1} - \eta_j)$. An extensive discussion of this approach in the context of discrete thickness selection can be found in [15].

C. Filter methods for multimaterial optimization

The proposed multimaterial design parametrizations exhibit mesh-dependence and checkerboard instabilities that are commonly encountered in topology optimization. To mitigate these issues, we employ a spatial filter that smears the stiffness contribution from one element spatially across several adjacent elements. This spatial filter is based on the work of Bruns and Tortorelli [4] and can be written as follows:

$$x^{(i)} = \frac{\sum_{r_{ij}<r_0}(r_0 - r_{ij})\tilde{x}^{(j)}}{\sum_{r_{ij}<r_0}(r_0 - r_{ij})} \tag{5}$$

where $\tilde{x}^{(j)}$ is the unfiltered variable of element j, and $x^{(i)}$ is the filtered variable. The filter radius is given by the parameter r_0, and r_{ij} is the spatial distance between the centroids of the elements i and j.

In the context of the DMO method, we apply the spatial filter to the design variables and compute the stiffness using (1). Note that we still apply the constraints and calculate the mass using the unfiltered variables. In the context of the discrete stiffness optimization method, we apply the spatial filter to the density variable, ρ, but still apply the constraints (3) on an element-by-element basis.

III. Parallel solution of large-scale linear systems

In this work, we solve the large-scale linear systems that arise from the structural analysis problem using geometric multigrid-preconditioned Krylov subspace methods. The advantage of using the multigrid method as a preconditioner

is that it offers the potential of both algorithmic scalability and good parallel efficiency. We utilize geometric rather than algebraic multigrid for its ease of implementation on the simple domains that we examine in this paper. While algebraic multigrid methods have been successfully applied to large-scale topology optimization problems, for instance [1], we focus on a geometric approach that is simpler to implement within our existing finite-element analysis and design optimization framework [17]. Specifically, the geometric approach does not require parallel matrix-matrix products to form the coarse grid approximations. One of the primary benefits of the algebraic approach, within the context of topology optimization problems, is that the large stiffness gradients within the structure are interpolated automatically, leading to higher-quality coarse grid approximations. However, our preliminary results suggest that the algebraic method proposed here is relatively insensitive to these large changes in stiffness in converged designs.

The geometric multigrid method is a recursive solution technique that operates on a hierarchy of meshes of decreasing dimension that discretize the same underlying problem [31, 24]. We refer to each mesh in the hierarchy as a mesh level. The multigrid algorithm requires four main components: matrix-vector products on each mesh level, an iterative method that acts to smooth the solution on each mesh level, interpolation and restriction operators to transfer the solution and residual between mesh levels, and a direct solution method on the coarsest mesh. In this work, we write the family of linear systems on each mesh level $i = 1, \ldots, N$, as follows:

$$\mathbf{A}_i \mathbf{u}_i = \mathbf{f}_i, \tag{6}$$

where $\mathbf{A}_i$ are the finite-element matrices, $\mathbf{u}_i$ is the solution vector at the current iteration and $\mathbf{f}_i$ is the right-hand-side. Here $i = 1$ is the finest mesh, and $i = N$ is the coarsest mesh. The smoothing operator on each mesh level is represented here as:

$$\mathbf{u}_i \leftarrow \mathbf{G}_i(\mathbf{r}_i), \tag{7}$$

where we use successive, symmetric over-relaxation [24]. Finally, we write the restriction and interpolation operators on all but the coarsest mesh as follows:

$$\begin{aligned} \mathbf{r}_{i+1} &= \mathbf{R}_i \mathbf{r}_i, \\ \mathbf{u}_i &= \mathbf{I}_i \mathbf{u}_{i+1}, \end{aligned} \tag{8}$$

for $i = 1, \ldots, N-1$, where $\mathbf{R}_i$ and $\mathbf{I}_i$ are the restriction and interpolation operators, respectively. The last requirement for multigrid is the solution on the coarsest mesh. Here we write this direct solution as $\mathbf{u}_N = \mathbf{A}_N^{-1} \mathbf{r}_N$. A complete multgrid V-cycle algorithm is illustrated in Algorithm 1.

Algorithm 1 Multigrid V-cycle

```
function MULTIGRID(r_i)
    if i = N then
        return A_N^{-1} r_N                          ▷ Direct solve on coarsest mesh
    end if
    u_i ← G_i(r_i)                                   ▷ Pre-smooth
    r_i ← r_i − A_i u_i                              ▷ Compute the residual at the current level
    r_{i+1} ← R_i r_i                                ▷ Restrict the residual to the next coarsest mesh
    u_{i+1} ← MULTIGRID(r_{i+1})                     ▷ Apply multigrid at the next level
    u_i ← u_i + I_i u_{i+1}                          ▷ Interpolate the solution from the coarser mesh
    u_i ← G_i(r_i)                                   ▷ Post-smooth
    return u_i
end function
```

In this work, we construct the interpolation and restriction operators using a full weighting scheme [31]. We note that our parallel implementation takes no shortcuts and applies the same interpolation and restriction operators regardless of the parallel domain decomposition. A key performance advantage of our multigrid implementation is that it utilizes an efficient parallel direct solution method on the coarsest mesh presented in Kennedy and Martins [17]. This solution method uses a direct Schur-complement based factorization that scales well with the number of processors, but poorly with increasing problem size. Within the multigrid method, however, the problem size of the coarsest mesh is significantly smaller than the finest mesh and is therefore within the range that can be solved efficiently using the direct matrix factorization.

Finally, we note that it is possible to use multigrid directly as the solution algorithm, however, we have found that it is more effective to use a single multigrid V-cycle as a preconditioner. Within this work, we solve the sparse linear systems using FGMRES(80) [25, 23].

A. Construction of the coarse approximations

One of the key aspects of the geometric multigrid algorithm presented above is the construction of the coarse grid approximations. In this work, we construct these approximations by coarsening the mesh uniformly and constructing equivalent volume-weighted element constitutive properties for each element in the coarse mesh. Given the design and spatially-dependent stiffness matrix, $\mathbf{C}(\mathbf{x})$, on the fine mesh, where $\mathbf{x}$ are the design variables, we compute the element stiffness on the coarse mesh as follows:

$$\mathbf{C}_r(\mathbf{x}) = \frac{1}{V_r}\int_{V_r} \mathbf{C}(\mathbf{x})\,\mathrm{d}V, \tag{9}$$

where V_r is the volume occupied by the coarse element. In this work, we use design parametrizations with uniform element properties. As a result, the integration (9) can be simplified as follows:

$$\mathbf{C}_r(\mathbf{x}) = \frac{1}{V_r}\sum_i V_i \mathbf{C}(\mathbf{x}) \tag{10}$$

where V_i are the volumes of the elements in the fine mesh within the coarse mesh element. For the case of isotropic materials with the same Poisson's ratio, Equation (10) can be simplified further to a volume-weighted average of the Young's modulus.

IV. Large-scale parallel optimization method

In the following section, we describe the large-scale parallel interior-point method that we use to solve the multi-material optimization problems. The proposed interior-point method uses well-established techniques from the literature, see for instance [30, 8, 22], but is implemented so that the design and gradient vectors, and constraint Jacobian matrices are stored in a distributed manner. In particular, the underlying multimaterial design parametrizations uses element-based design variables that are well-aligned with the element-based domain decomposition for the finite-element problem [17]. This element-based scheme allows distributed storage and parallel computation, leading to excellent parallel performance for very large design problems. The parallel performance of the optimization method is critical since the multimaterial design problems we examine contain millions of design variables and millions of sparse constraints. Furthermore, we take advantage of the sparsity structure of the constraint Jacobians for both the DMO parametrization (1) and discrete stiffness parametrization (3). Without this structure, the interior-point method would require operations on matrices with more general sparsity patterns that would be significantly more challenging to implement efficiently in parallel.

We formulate the large-scale multimaterial design problems using a reduced-space method where the optimization problem can be written as follows:

$$\begin{aligned}
\text{minimize} \quad & f(\mathbf{x},\mathbf{u}(\mathbf{x})) \\
\text{with respect to} \quad & \mathbf{x} \\
\text{such that} \quad & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \\
& \mathbf{c}(\mathbf{x},\mathbf{u}(\mathbf{x})) \geq 0 \\
& \mathbf{c}_s(\mathbf{x}) \geq 0 \\
\text{governed by} \quad & \mathbf{R}(\mathbf{x},\mathbf{u}(\mathbf{x})) = 0
\end{aligned} \tag{11}$$

where $f(\mathbf{x},\mathbf{u}(\mathbf{x}))$ is the objective, $\mathbf{x}$ are the design variables, $\mathbf{u}$ are the state variables, $\mathbf{R}(\mathbf{x},\mathbf{u}) = 0$ are the residuals of the governing equations, and $\mathbf{x}_l$ and $\mathbf{x}_u$ are the upper and lower bounds on the design variables, respectively. In the reduced space method, the governing equations, $\mathbf{R}(\mathbf{x},\mathbf{u}) = 0$, are used to recast the state variables $\mathbf{u}$ as implicit functions of $\mathbf{x}$.

To obtain the best computational efficiency on large-scale problems, it is important to exploit the sparsity structure of the constraint Jacobians in the optimization problem (11). In particular, the constraints are divided into two distinct sets: constraints that depend on both the design variables and the state variables, $\mathbf{c}(\mathbf{x},\mathbf{u}(\mathbf{x}))$, and constraints that depend only on design variables, $\mathbf{c}_s(\mathbf{x})$. Writing the gradient, or total derivative [19], of a function with respect to the design variables $\mathbf{x}$ as $\nabla_{\mathbf{x}}\cdot$, we define the constraint Jacobians as follows:

$$\begin{aligned}
\mathbf{A} &= \nabla_{\mathbf{x}}\mathbf{c}(\mathbf{x}), \\
\mathbf{A}_s &= \nabla_{\mathbf{x}}\mathbf{c}_s(\mathbf{x}).
\end{aligned}$$

Note that the constraint Jacobian, $\mathbf{A}$, is a dense matrix due to the implicit dependence of $\mathbf{c}$ on the state variables, requiring the use of a derivative implementation such as the adjoint method [19]. However, the second constraint

Jacobian, $\mathbf{A}_s$, is very sparse and has a special structure with groups of orthogonal rows, such that whenever the matrix $\mathbf{D}$ is diagonal, the product:

$$\mathbf{A}_s\mathbf{D}\mathbf{A}_s^T = \mathbf{G} = \text{diag}\{\mathbf{G}_{11}, \mathbf{G}_{22}, \ldots, \mathbf{G}_{qq}\}, \tag{12}$$

is block-diagonal where the dimensions of the block matrices $\mathbf{G}_{11}, \mathbf{G}_{22}, \ldots, \mathbf{G}_{qq}$ are small matrices with sizes less than or equal to the number of discrete material options. This property of the constraint Jacobian enables an efficient distributed parallel implementation, even for cases with millions of sparse constraints.

A. The perturbed Karush–Kuhn–Tucker conditions

The interior-point method is based on the perturbed Karush–Kuhn–Tucker conditions that can be written as follows:

$$\begin{aligned} \mathbf{g} - \mathbf{A}^T\mathbf{z} - \mathbf{A}_s^T\mathbf{z}_s - \mathbf{z}_l + \mathbf{z}_u = 0 \quad & \quad \mathbf{S}\mathbf{z} = \mu\mathbf{e} \\ \mathbf{c} - \mathbf{s} = 0 \quad & \quad \mathbf{S}_s\mathbf{z}_s = \mu\mathbf{e} \\ \mathbf{c}_s - \mathbf{s}_s = 0 \quad & \quad (\mathbf{X} - \mathbf{X}_l)\mathbf{z}_l = \mu\mathbf{e} \\ & \quad (\mathbf{X}_u - \mathbf{X})\mathbf{z}_u = \mu\mathbf{e} \end{aligned} \tag{13}$$

where $\mathbf{g}^T = \nabla_{\mathbf{x}} f(\mathbf{x})$ is the gradient, μ is the barrier parameter, $\mathbf{z}$, $\mathbf{z}_s$, $\mathbf{z}_l$, and $\mathbf{z}_u$ are Lagrange multipliers for the dense, sparse, lower-bounds and upper-bounds, respectively, and $\mathbf{s}$ and $\mathbf{s}_s$ are slack variables for the dense and sparse constraints, respectively. Here, we use the convention that the upper case letters of vectors, e.g. $\mathbf{X}$, $\mathbf{X}_l$, $\mathbf{X}_u$, represent diagonal matrices formed from the elements of the corresponding vector, e.g. $\mathbf{S} = \text{diag}\{s_1, s_2, \ldots\}$. In addition, the vector $\mathbf{e}$ is used to denote a generic vector of all unit entries, whose length can be inferred from the context. Note that, as in all interior point methods, we explicitly enforce strict positivity for the slack variables and Lagrange multipliers. In addition, we enforce that the design variables $\mathbf{x}$ satisfy $\mathbf{x}_l < \mathbf{x} < \mathbf{x}_u$, for all iterations.

We approximately solve the equations (13) for a sequence of barrier parameters μ_k such that $\mu_k \to 0$ for $k \to \infty$. Here, we use a monotone approach Fiacco and McCormick [10], and determine whether to reduce the barrier parameter based on the stopping criterion:

$$\begin{aligned} \max\{\ & ||\mathbf{g} + \mathbf{A}^T\mathbf{z} + \mathbf{A}_s^T\mathbf{z}_s - \mathbf{z}_l + \mathbf{z}_u||_\infty, ||\mathbf{c} - \mathbf{s}||_\infty,\ ||\mathbf{c}_s - \mathbf{s}_s||_\infty, \\ & ||\mathbf{S}\mathbf{z} - \mu_k\mathbf{e}||_\infty,\ ||\mathbf{S}_s\mathbf{z}_s - \mu_k\mathbf{e}||_\infty,\ ||(\mathbf{X} - \mathbf{X}_l)\mathbf{z}_l - \mu_k\mathbf{e}||_\infty,\ ||(\mathbf{X}_u - \mathbf{X})\mathbf{z}_u - \mu_k\mathbf{e}||_\infty,\} \leq 10\mu_k. \end{aligned} \tag{14}$$

After the stopping criterion is satisfied, we adjust the barrier parameter using the expression $\mu_{k+1} \leftarrow \min\{\theta\mu_k, \mu_k^\beta\}$ for $\theta \in (0,1]$, $\beta \in (1,2]$. In the present study, we use the parameters $\theta = 0.25$, and $\beta = 1.1$.

We employ a line search method and obtain the search direction by solving an approximate linearization of the perturbed KKT conditions (13). The linearization is exact excluding the Hessian of the Lagrangian which we approximate using a quasi-Newton method. To achieve an efficient parallel implementation, we use a Hessian approximation based on the compact limited-memory BFGS representation [7]. In this approach, the design variable updates and gradient changes for the last m iterations are stored in a matrix $\bar{\mathbf{Y}} \in \mathbb{R}^{n \times 2m}$, where $m \ll n$. An approximate Hessian, $\mathbf{B}$, is formed from the BFGS update scheme [20] as follows:

$$\mathbf{B} = \gamma\mathbf{I} + \bar{\mathbf{Y}}\mathbf{M}\bar{\mathbf{Y}}^T \tag{15}$$

where γ is a scalar and $\mathbf{M} \in \mathbb{R}^{2m \times 2m}$ is a small matrix. The matrix $\mathbf{B}$ is a Hessian approximation that approximates the true Hessian of the Lagrangian such that:

$$\mathbf{B} \approx \nabla_{\mathbf{x}} \left(\mathbf{g} + \mathbf{A}^T\mathbf{z} + \mathbf{A}_s^T\mathbf{z}_s\right),$$

with the added condition that $\mathbf{B}$ remain positive definite so that the search direction is a descent direction of the merit function. To ensure that the Hessian approximation remains positive definite, we use a damped update scheme that ensures that the curvature condition is satisfied [21]. In this work, we typically set m to a value of $m = 20$. This choice balances additional computational costs associated with each optimization iteration against potentially fewer optimization iterations as a result of a more-accurate Hessian approximation.

B. Parallel solution of the KKT system

The single most computationally expensive operation at each iteration of the optimization algorithm is the solution of the linearized KKT system obtained from the perturbed KKT conditions (13). We now present an overview of how we

solve this linear system in a computationally efficient manner in parallel. To derive the proposed solution procedure, we first express the linearized KKT matrix as a combination of two matrices which take the form:

$$\mathbf{J}\mathbf{p} = \left[\mathbf{J}_0 + \mathbf{Y}\mathbf{M}\mathbf{Y}^T\right]\mathbf{p} = -\mathbf{r} \tag{16}$$

where the matrix $\mathbf{Y}$ can be written as:

$$\mathbf{Y} = \begin{bmatrix}\bar{\mathbf{Y}}^T & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T,$$

and the matrices $\bar{\mathbf{Y}}$ and $\mathbf{M}$ are from the compact BFGS representation (15). Note that the terms in $\mathbf{J}_0$ represent the diagonal term γ from the compact BFGS representation and all other first-order terms from the linearized KKT system.

We can obtain an exact solution to this linear system (16), using the Sherman–Morrison–Woodbury formula. This formula leads to the following expression for the update $\mathbf{p}$:

$$\mathbf{p} = \mathbf{J}_0^{-1}\mathbf{Y}\mathbf{C}^{-1}\mathbf{Y}^T\mathbf{J}_0^{-1}\mathbf{r} - \mathbf{J}_0^{-1}\mathbf{r}$$

where the matrix $\mathbf{C} \in \mathbb{R}^{2m\times 2m}$ is given as follows:

$$\mathbf{C} = \mathbf{Y}^T\mathbf{J}_0^{-1}\mathbf{Y} - \mathbf{M}.$$

Note that we can now obtain a solution of the linear system (16) from the solution of $2m+1$ linear systems of the form $\mathbf{J}_0\mathbf{y} = \mathbf{b}$. Furthermore, if we store the matrix $\bar{\mathbf{Y}}$ as a series of column vectors, then the operations required to compute the solution consist of operations with small matrices of size $\mathcal{O}(m)$, parallel vector-vector products, and the application of $\mathbf{J}_0^{-1}$. Since vector-vector operations parallelize efficiently for distributed vectors, and the small matrix operations normally constitute a small contribution to the overall computational time, we concentrate on the parallel solution of systems of the form $\mathbf{J}_0\mathbf{y} = \mathbf{b}$. Note that we refer to the linear system $\mathbf{J}_0\mathbf{y} = \mathbf{b}$ as the diagonal KKT matrix since the Hessian term in the matrix $\mathbf{J}_0$ is replaced by a diagonal matrix, $\mathbf{B} = \gamma\mathbf{I}$.

C. Parallel solution of the diagonal KKT system

We solve the diagonal KKT system $\mathbf{J}_0\mathbf{y} = \mathbf{b}$ in parallel through a series of variable eliminations. In general, this type of method suffers from numerical cancellation, however, our experience has been that this method produces remarkably accurate steps, even for very small values of the barrier parameter. We attribute this observed behavior to the structure of the constraint Jacobians and the form of the multimaterial parametrization constraints. In our solution method, we use a series of variable eliminations that result in a linear system for the Lagrange multipliers of the dense constraints. These eliminations produce a series of Schur-complement matrices that can be computed in sequence. Each of these computations require vector-vector operations and can therefore be implemented efficiently in parallel. Since each individual operation in the solution procedure can be performed in parallel, apart from a limited number of matrix operations on matrices of size $\mathcal{O}(m)$, the entire solution procedure scales efficiently.

The solution procedure can be derived by first linearizng the final four equations of the perturbed KKT equations, and obtaining the solution for the slack variables and lower and upper Lagrange multiplier bound variables as follows:

$$\begin{aligned}
\mathbf{y}_s &= \mathbf{Z}^{-1}\mathbf{b}_s - \mathbf{Z}^{-1}\mathbf{S}\mathbf{y}_z,\\
\mathbf{y}_{ss} &= \mathbf{Z}_s^{-1}\mathbf{b}_{ss} - \mathbf{Z}_s^{-1}\mathbf{S}_s\mathbf{y}_{zs},\\
\mathbf{y}_{zl} &= (\mathbf{X} - \mathbf{X}_l)^{-1}(\mathbf{b}_{zl} - \mathbf{Z}_l\mathbf{y}_x),\\
\mathbf{y}_{zu} &= (\mathbf{X}_u - \mathbf{X})^{-1}(\mathbf{b}_{zu} + \mathbf{Z}_u\mathbf{y}_x).
\end{aligned} \tag{17}$$

Next, we linearize the first three KKT equations, and replace the Hessian with the diagonal term $\mathbf{B} = \gamma\mathbf{I}$, to obtain the following expressions:

$$\begin{aligned}
\gamma\mathbf{y}_x - \mathbf{A}^T\mathbf{y}_z - \mathbf{A}_s^T\mathbf{y}_{zs} - \mathbf{y}_{zl} + \mathbf{y}_{zu} &= \mathbf{b}_x,\\
\mathbf{A}\mathbf{y}_x - \mathbf{y}_s &= \mathbf{b}_c,\\
\mathbf{A}_s\mathbf{y}_x - \mathbf{y}_{ss} &= \mathbf{b}_{cs}.
\end{aligned} \tag{18}$$

Substituting the expressions for the slack and Lagrange multiplier updates (17) into the expression for the first three linearized KKT conditions (18) yields the following expressions:

$$\begin{aligned}
\mathbf{D}\mathbf{y}_x - \mathbf{A}^T\mathbf{y}_z - \mathbf{A}_s^T\mathbf{y}_{zs} &= \mathbf{b}_x + (\mathbf{X} - \mathbf{X}_l)^{-1}\mathbf{b}_{zl} - (\mathbf{X}_u - \mathbf{X})^{-1}\mathbf{b}_{zu},\\
\mathbf{A}\mathbf{y}_x + \mathbf{Z}^{-1}\mathbf{S}\mathbf{y}_z &= \mathbf{b}_c + \mathbf{Z}^{-1}\mathbf{b}_s,\\
\mathbf{A}_s\mathbf{y}_x + \mathbf{Z}_s^{-1}\mathbf{S}_s\mathbf{y}_{zs} &= \mathbf{b}_{cs} + \mathbf{Z}_s^{-1}\mathbf{b}_{ss},
\end{aligned} \tag{19}$$

where the diagonal matrix $\mathbf{D}$ is defined as follows:

$$\mathbf{D} = \left[\gamma\mathbf{I} + (\mathbf{X} - \mathbf{X}_l)^{-1}\mathbf{Z}_l + (\mathbf{X}_u - \mathbf{X})^{-1}\mathbf{Z}_u\right].$$

Note that the linear system (19) only involves the updates $\mathbf{y}_x$, $\mathbf{y}_z$ and $\mathbf{y}_{zs}$. We now proceed by eliminating $\mathbf{y}_x$ from the linear system (19), producing the following linear system:

$$\begin{bmatrix} \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T + \mathbf{Z}^{-1}\mathbf{S} & \mathbf{A}\mathbf{D}^{-1}\mathbf{A}_s^T \\ \mathbf{A}_s\mathbf{D}^{-1}\mathbf{A}^T & \mathbf{A}_s\mathbf{D}^{-1}\mathbf{A}_s^T + \mathbf{Z}_s^{-1}\mathbf{S}_s \end{bmatrix} \begin{bmatrix} \mathbf{y}_z \\ \mathbf{y}_{zs} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_c + \mathbf{Z}^{-1}\mathbf{b}_s - \mathbf{A}\mathbf{d} \\ \mathbf{b}_{cs} + \mathbf{Z}_s^{-1}\mathbf{b}_{ss} - \mathbf{A}_s\mathbf{d} \end{bmatrix}, \tag{20}$$

where the right-hand term is computed as follows:

$$\mathbf{d} = \mathbf{D}^{-1}\left[\mathbf{b}_x + (\mathbf{X} - \mathbf{X}_l)^{-1}\mathbf{b}_{zl} - (\mathbf{X}_u - \mathbf{X})^{-1}\mathbf{b}_{zu}\right].$$

The linear system (20) is only in terms of the updates to the Lagrange multipliers. We now use the sparsity property of the constraint Jacobian (12) and recognize that the matrix,

$$\mathbf{G} \equiv \mathbf{A}_s\mathbf{D}^{-1}\mathbf{A}_s^T + \mathbf{Z}_s^{-1}\mathbf{S}_s, \tag{21}$$

is block-diagonal, where each block matrix is at largest $N \times N$, where N is the number of candidate materials, and can be factored independently on each processor. Using the factorization of $\mathbf{G}$, we can compute the solution to the Lagrange multipliers for the dense constraints as follows:

$$\begin{aligned} \left[\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T + \mathbf{Z}^{-1}\mathbf{S} - \mathbf{A}\mathbf{D}^{-1}\mathbf{A}_s^T\mathbf{G}^{-1}\mathbf{A}_s\mathbf{D}^{-1}\mathbf{A}^T\right]\mathbf{y}_z = \\ \mathbf{b}_c + \mathbf{Z}^{-1}\mathbf{b}_s - \mathbf{A}\mathbf{d} - \mathbf{A}\mathbf{D}^{-1}\mathbf{A}_s^T\mathbf{G}^{-1}\left(\mathbf{b}_{cs} + \mathbf{Z}_s^{-1}\mathbf{b}_{ss} - \mathbf{A}_s\mathbf{d}\right). \end{aligned} \tag{22}$$

After obtaining a value for $\mathbf{y}_z$, we can obtain the values of the remaining variables using (19) and (17).

This solution procedure is invoked each time a solution of the form $\mathbf{J}_0\mathbf{y} = \mathbf{b}$ is required. However, entries such as the factorization of $\mathbf{G}$, and the matrix $\mathbf{D}$, can be computed once and stored for all subsequent solutions.

D. Merit function and line search

We globalize the interior-point method using a line search method that ensures a sufficient decrease of a merit function at each iteration. The line search is based on the following ℓ_2 merit function:

$$\varphi(\mathbf{d}) = f(\mathbf{x}) + \nu||\mathbf{c}(\mathbf{x}) - \mathbf{s}||_2 + \nu||\mathbf{c}_s(\mathbf{x}) - \mathbf{s}_s||_2 - \mu\left(\mathbf{e}^T\log\mathbf{s} + \mathbf{e}^T\log\mathbf{s}_s + \mathbf{e}^T\log(\mathbf{x} - \mathbf{x}_l) + \mathbf{e}^T\log(\mathbf{x}_u - \mathbf{x})\right), \tag{23}$$

where $\mathbf{d}$ is the vector of all design, slack and Lagrange multiplier variables, and we use the notation $\log\mathbf{s} = \mathbf{t}$ such that $t_i = \log s_i$. The penalty parameter ν is selected to ensure a sufficiently negative descent direction, such that $\mathbf{p}^T\nabla\varphi$ is sufficiently negative [21]. At each step we use a line search that seeks a point that satisfies the Armijo sufficient decrease condition:

$$\varphi(\mathbf{d} + \alpha\mathbf{p}) < \varphi(\mathbf{d}) + c_1\mathbf{p}^T\nabla\varphi,$$

where we typically choose $c_1 = 10^{-3}$. If a step is unsuccessful, we select the next step using a quadratic interpolation based on the initial point and slope of the merit function along the search direction, as well as the most recent merit function value. Since $\mathbf{p}^T\nabla\varphi$ is negative, and $\varphi(\mathbf{d} + \alpha\mathbf{p}) \geq \varphi(\mathbf{d}) + c_1\mathbf{p}^T\nabla\varphi$, this sequence of step lengths is decreasing.

To ensure that the design variables remain within bounds and that the Lagrange multipliers and slack variables remain sufficiently positive, we use a fraction-to-the-boundary rule, such as:

$$\begin{aligned} \alpha_x &= \max\left\{\alpha \in (0,1] \mid \mathbf{x} + \alpha\mathbf{p}_\mathbf{x} - \mathbf{x}_l \geq (1-\tau)(\mathbf{x} - \mathbf{x}_l)\right\}, \\ \alpha_z &= \max\left\{\alpha \in (0,1] \mid \mathbf{z} + \alpha\mathbf{p}_\mathbf{z} \geq (1-\tau)\mathbf{z}\right\}, \end{aligned}$$

with analogous expressions for the remaining components of the step length vector $\mathbf{p}$. Note that α_x is the step length for the design and slack variables, and α_z is the step length for the Lagrange multipliers. Following Wächter and Biegler [30], we set the parameter τ as follows:

$$\tau = \max(0.95, 1 - \mu).$$

To avoid situations in which there is a large discrepancy between the step lengths, we guard α_x and α_z such that if $\alpha_x \geq \alpha_z$, we set:

$$\alpha_x = \max(\min(\alpha_x, 100\alpha_z), \alpha_z/100),$$

otherwise if $\alpha_z > \alpha_x$, we set:

$$\alpha_z = \max(\min(\alpha_z, 100\alpha_x), \alpha_x/100).$$

Note that this modification only has an effect if the difference in step lengths exceeds 100. This modification does not interfere with the asymptotic convergence behavior of the algorithm and enables faster recovery from poor steps early in the optimization.

V. Results

In this section, we present the results from the implementation of the algorithms described above. To demonstrate the proposed framework, we perform mass-constrained compliance minimization of a structure subject to an applied load. The design domain and loading conditions are illustrated in Figure 2. The problem domain is a rectangular prism of length $5 \times 5 \times 20$ units along each coordinate direction, with an aspect ratio of 4. We utilize four candidate materials, with properties illustrated in Figure 2. We impose a mass constraint on the structure such that the mass cannot exceed $\rho_0 L^3/125$. The structure is discretized using a mesh of $64 \times 64 \times 256$ brick finite-elements, resulting in a uniform aspect ratio for all elements.

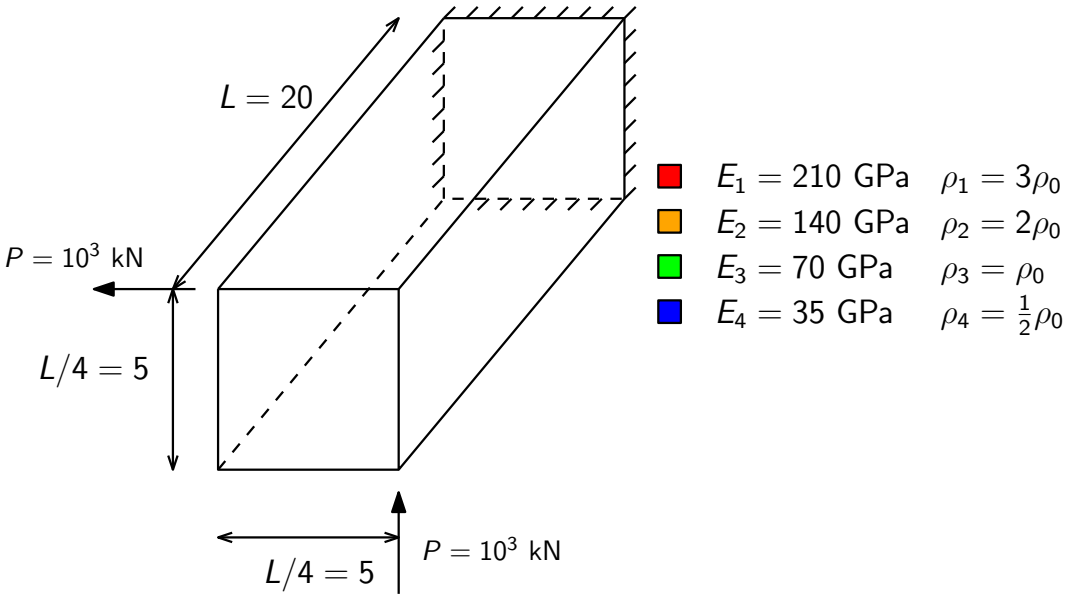

Figure 2: Domain geometry and loading conditions for the multimaterial optimization problem.

We solve the design problem illustrated in Figure 2 using the DMO parametrization presented above. This parametrization requires 5 design variables per element: one for the topology variable, and one for each of the four material selection variables. As a result, the design problem contains 5.24 million design variables. In addition, the DMO parametrization requires a single linear equality constraint for each group of element design variables, resulting in 1.05 million linear equality constraints. In conjunction with these element variables, we also use a filter (5) that includes the element itself and the 32 closest neighbors.

Figure 3 illustrates the compliance and design history for the DMO parametrization at the 500-th, 1000-th, 1500-th, 2000-th and 2450-th major iterations. Note that no post-solution smoothing is performed in the visualization and each element is distinctly visible. The jumps in the compliance correspond to a modification of the barrier parameter. The design at the 500-th iteration exhibits a similar overall topology to the final design, but with a significantly different material distribution. These differences would not be observed without fully converging the optimization problem. Over the course of the optimization, most of the softest (blue) material is removed and only small amounts of the second-softest (green) material remain. The majority of the material used in the structure is the stiffest (red) material. The final change in topology occurs slowly, as a slender member connects the main structure to the point of load application.

Overall the optimization problem required 9 hours and 52 minutes of wall time on 32 processors with just over 2500 major iterations. The optimization algorithm, therefore, averages approximately 4.2 iterations a minute, or roughly an iteration every 15 seconds. Each iteration consists of the solution of the finite-element governing equations, the evaluation of the constraint gradients, and the computation of the interior-point step. This rapid optimization iteration illustrates the effectiveness of the proposed parallel optimization algorithms.

VI. Conclusions

In this paper, we have presented techniques for large-scale topology and multimaterial optimization. First, we presented two multimaterial parametrization methods that have proven to be effective for moderate-scale problems. Second, we presented a geometric multigrid preconditioner that scales well with increasing numbers of processors

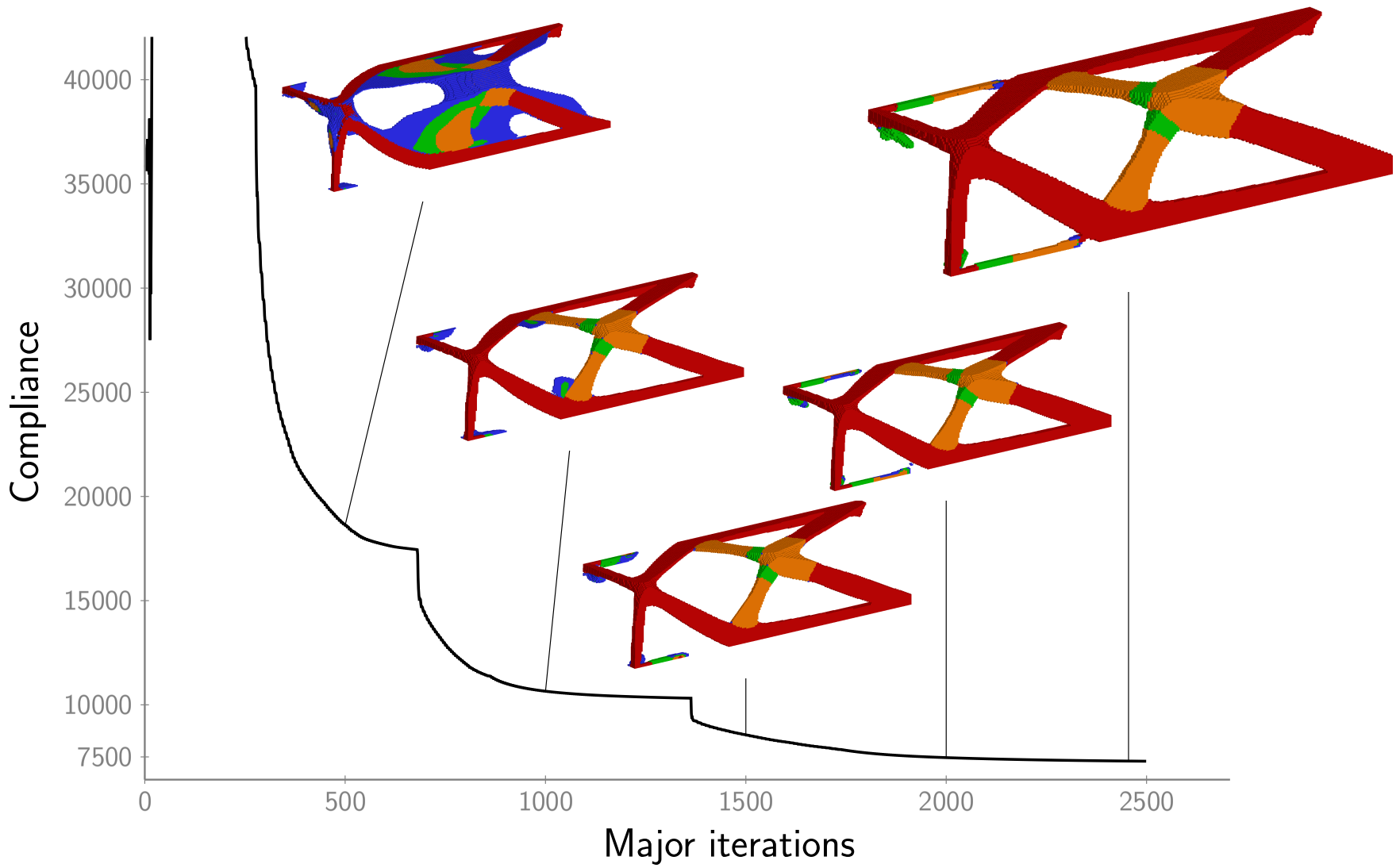

Figure 3: Optimization history for the mass-constrained compliance minimization problem for a $64 \times 64 \times 256$ mesh with 3.26 million degrees of freedom, 5.24 million design variables, and 1.05 million linear constraints. The red, orange, green and blue colors indicate the four material selections that are possible within the structure. Note that in each figure the loads are applied on the left hand side of the structure while the right-hand-side is fully clamped.

and increasing size of the finite-element mesh. Finally, we presented a parallel interior-point method for design optimization that exhibits excellent parallel scalability. We demonstrated the efficiency of the proposed analysis and optimization methods on a large-scale multimaterial optimization problem with 3.26 million structural degrees of freedom, 5.24 million design variables, and 1.05 million constraints. This problem required 9 hours and 52 minutes of wall time to complete the optimization on 32 processors. Overall the optimization required just over 2500 major iterations, or roughly a major iteration every 15 seconds of wall time. Future work will consider the application of these methods to stress-constrained multimaterial problems.

Acknowledgments

The author gratefully acknowledges the financial support of the Georgia Institute of Technology. The computations in this work were performed on the Stampede supercomputer administered under the Texas Advanced Compute Center (TACC). Access to this system was provided under the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575.

References

[1] O. Amir, N. Aage, and B. Lazarov. On multigrid-CG for efficient topology optimization. *Structural and Multidisciplinary Optimization*, 49(5):815–829, 2014. ISSN 1615-147X. doi:10.1007/s00158-013-1015-5.

[2] T. Borrvall and J. Petersson. Large-scale topology optimization in 3D using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190(4647):6201 – 6229, 2001. ISSN 0045-7825. doi:10.1016/S0045-7825(01)00216-X.

[3] T. Borrvall and J. Petersson. Topology optimization using regularized intermediate density control. *Computer Methods in Applied Mechanics and Engineering*, 190(37-38):4911 – 4928, 2001. ISSN 0045-7825. doi:10.1016/S0045-7825(00)00356-X.

[4] T. E. Bruns and D. A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(2627):3443 – 3459, 2001. ISSN 0045-7825. doi:10.1016/S0045-7825(00)00278-4.

[5] M. Bruyneel. SFP - A new parameterization based on shape functions for optimal material selection: application to conventional composite plies. *Structural and Multidisciplinary Optimization*, 43:17–27, 2011. ISSN 1615-147X. doi:10.1007/s00158-010-0548-0.

[6] M. Bruyneel, P. Duysinx, C. Fleury, and T. Gao. Extensions of the shape functions with penalization parametrization for composite-ply optimization. *AIAA Journal*, 49(10):2325–2329, October 2011. doi:10.2514/1.J051225.

[7] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156, 1994. ISSN 0025-5610. doi:10.1007/BF01582063.

[8] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000. ISSN 0025-5610. doi:10.1007/PL00011391.

[9] A. Evgrafov, C. J. Rupp, K. Maute, and M. L. Dunn. Large-scale parallel topology optimization using a dual-primal substructuring solver. *Structural and Multidisciplinary Optimization*, 36(4):329–345, 2008. ISSN 1615-147X. doi:10.1007/s00158-007-0190-7.

[10] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming*. Society for Industrial and Applied Mathematics, 1990. doi:10.1137/1.9781611971316.

[11] T. Gao, W. Zhang, and P. Duysinx. A bi-value coding parameterization scheme for the discrete optimal orientation design of the composite laminate. *International Journal for Numerical Methods in Engineering*, 91(1):98–114, 2012. ISSN 1097-0207. doi:10.1002/nme.4270.

[12] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):pp. 99–131, 2005. ISSN 00361445. doi:10.1137/S0036144504446096.

[13] C. Hvejsel and E. Lund. Material interpolation schemes for unified topology and multi-material optimization. *Structural and Multidisciplinary Optimization*, 43:811–825, 2011. ISSN 1615-147X. doi:10.1007/s00158-011-0625-z.

[14] C. Hvejsel, E. Lund, and M. Stolpe. Optimization strategies for discrete multi-material stiffness optimization. *Structural and Multidisciplinary Optimization*, 44:149–163, 2011. ISSN 1615-147X. doi:10.1007/s00158-011-0648-5.

[15] G. J. Kennedy. Discrete thickness optimization via piecewise constraint penalization. *Structural and Multidisciplinary Optimization*, 2014. doi:10.1007/s00158-014-1210-z. Accepted.

[16] G. J. Kennedy and J. R. R. A. Martins. A laminate parametrization technique for discrete ply-angle problems with manufacturing constraints. *Structural and Multidisciplinary Optimization*, pages 1–15, 2013. ISSN 1615-147X. doi:10.1007/s00158-013-0906-9.

[17] G. J. Kennedy and J. R. R. A. Martins. A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures. *Finite Elements in Analysis and Design*, 87(0):56 – 73, 2014. ISSN 0168-874X. doi:10.1016/j.finel.2014.04.011.

[18] T. S. Kim, J. E. Kim, and Y. Y. Kim. Parallelized structural topology optimization for eigenvalue problems. *International Journal of Solids and Structures*, 41(9-10):2623 – 2641, 2004. ISSN 0020-7683. doi:10.1016/j.ijsolstr.2003.11.027.

[19] J. R. R. A. Martins and J. T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA Journal*, 51(11):2582–2599, November 2013. doi:10.2514/1.J052184.

[20] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980. doi:10.1090/S0025-5718-1980-0572855-7.

[21] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[22] M. H. R.H. Byrd and J. Nocedal. An interior point method for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999. doi:10.1137/S1052623497325107.

[23] Y. Saad. A fexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993. doi:10.1137/0914028.

[24] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Pub. Co., 2nd edition, 2003.

[25] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986. doi:10.1137/0907058.

[26] S. N. Sørensen and E. Lund. Topology and thickness optimization of laminated composites including manufacturing constraints. *Structural and Multidisciplinary Optimization*, 48(2):249–265, 2013. ISSN 1615-147X. doi:10.1007/s00158-013-0904-y.

[27] S. N. Sørensen, R. Sørensen, and E. Lund. DMTO – a method for discrete material and thickness optimization of laminated composite structures. *Structural and Multidisciplinary Optimization*, 50(1):25–47, 2014. ISSN 1615-147X. doi:10.1007/s00158-014-1047-5.

[28] J. Stegmann and E. Lund. Discrete material optimization of general composite shell structures. *International Journal for Numerical Methods in Engineering*, pages 2009–2027, 2005. ISSN 1097-0207. doi:10.1002/nme.1259.

[29] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22:116–124, 2001. ISSN 1615-147X. doi:10.1007/s001580100129.

[30] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1):25–57, May 2006. ISSN 0025-5610. doi:10.1007/s10107-004-0559-y.

[31] P. Wesseling. *An introduction to multigrid methods*. John Wiley & Sons, 1992.