A Full-Space Method with Matrix Aggregates for Stress-Constrained Structural Optimization

Graeme J. Kennedy*

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia

Stress-constrained structural optimization is challenging due to the nature of the local stress constraints that must be imposed point-wise everywhere within the structural domain. These constraints lead to optimization problems for which classical reduced-space methods scale poorly with increasing number of local stress constraints. To address this issue, a full-space barrier method is developed which is designed for problems with large numbers of constraints. Within this method, local stress constraints are formulated as matrix inequalities using convex optimization methods. Matrix aggregates are introduced which aggregate local stress constraints formulated as linear matrix inequalities. The advantage of the linear matrix inequality formulation is that the resulting stress constraints are convex in both the design and state variables, where non-convexity enters only through the governing equations. The proposed method is demonstrated on a plane stress problem.

I. Introduction

Stress-constrained structural optimization problems impose a bound on the stress within a structure that is subject to a series of loading conditions. Stress-constrained design problems are more challenging to solve than compliance-based design problems due to the local nature of the stress constraints that must be imposed everywhere within the structure [1, 23, 16]. Imposing local stress constraints at many points within the structure leads to a large number of design- and state-dependent constraints whose quantity scales with problem size. Within the context of reduced-space methods, where the state variables are treated as implicit functions of the design variables [2, 3], the computational cost of evaluating the constraint-gradients dominates the solution time, making the solution of large-scale problems computationally prohibitive. The computational cost of gradient evaluation can be addressed using specialized reduced-space methods that efficiently compute Jacobian-vector products [15, 7]. However, these methods have not reached the same maturity as classical reduced-space techniques.

Discrete constraint aggregation techniques can be used to alleviate the computational cost of constraint gradient evaluation by aggregating local stress constraints into a single equivalent constraint [1, 23, 17]. The most common discrete constraint aggregation technique is the discrete Kreisselmeier–Steinhauser (KS) function, originally developed for control systems design [14], and subsequently adapted to a wide range of structural and multidisciplinary design optimization problems [24, 29, 1, 23, 16, 8, 21, 12, 17, 18, 13]. While discrete constraint aggregation reduces the computational cost of gradient evaluation, these methods also increase the nonlinearity in the design problem. Essentially, aggregation methods combine information from individual local constraints into a single function. This leads to an overall increase in the number of iterations required to solve the optimization problem since local gradient information is no longer available. Another issue with discrete aggregation techniques is that they impose a bound on the stress only at a discrete number of points within the structure. In the context of PDE-constrained optimization, the local bound on stress is infinite-dimensional in the sense that the constraint must be imposed everywhere within the domain. To remove this infinite-dimensionality, continuous aggregation methods can be used to rigorously enforce a bound on the stress [1, 11, 10]. Nevertheless, while continuous aggregation methods offer a more-attractive mathematical rationale, they still suffer from the same issue of nonlinearity as discrete aggregates, also producing design optimization problems that are more difficult to solve [10].

In this paper, we present methods to address the computational cost of stress-constrained structural optimization by using a full-space approach in which the design and state variables are treated equally. Full-space methods scale well with increasing dimensionality of the design space, number of state variables and number of constraints. We focus on convex stress-constraint formulations in order to control the nonlinearity of the stress constraints. In Section II we pose stress constraints as linear matrix inequalities that can be applied point-wise within the structure. Since these constraints are point-wise constraints, they suffer from the same difficulties as discrete aggregation methods. In order to overcome these issues, in Section III we propose *matrix aggregates* which aggregate matrix inequalities. In Section IV we propose a full-space barrier method to solve the resulting stress-constrained optimization problem

*Assistant Professor, AIAA Member

efficiently. Finally, in Section V, we present the results of a stress-constrained mass minimization studies using the full-space barrier method.

A. Notation

Before proceeding further, we describe the notation used within this paper. Bold face lower case letters denote vectors, such as $\mathbf{u} \in \mathbb{R}^m$, while bold face upper case letters denote matrices, such as $\mathbf{A} \in \mathbb{R}^{n \times n}$. When necessary the components of vectors, $\mathbf{u} \in \mathbb{R}^m$, are written as u_i. We denote the set of real symmetric $n \times n$ matrices as $\mathbb{S}^n$, the set of positive semi-definite matrices as $\mathbb{S}^n_+$, and the set of positive definite matrices as $\mathbb{S}^n_{++}$, respectively. We use the inequality symbols $\succ$ and $\succeq$ to denote a partial ordering on $\mathbb{S}^n$. In particular, if $\mathbf{A}, \mathbf{B} \in \mathbb{S}^n$, then $\mathbf{A} \succeq \mathbf{B}$ implies $\mathbf{A} - \mathbf{B} \in \mathbb{S}^n_+$ and $\mathbf{A} \succ \mathbf{B}$ implies that $\mathbf{A} - \mathbf{B} \in \mathbb{S}^n_{++}$. Furthermore, the conventional inequality symbols $>$ and $\geq$ denote component-wise inequalities on vectors and matrices.

II. Problem formulation

In this paper, we develop methods for stress-constrained mass minimization problems which take the following form:

$$
\begin{aligned}
\min_{\mathbf{x}, \mathbf{u}_1, \ldots, \mathbf{u}_K} \quad & m(\mathbf{x}) & & \\
\text{governed by} \quad & \mathbf{K}(\mathbf{x}) \mathbf{u}_k = \mathbf{f}_k & & k = 1, \ldots, K \\
\text{such that} \quad & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u & & \\
& \sigma(\xi, \mathbf{x}, \mathbf{u}_k) \leq \sigma_{allow} & & \xi \in \Omega
\end{aligned}
\tag{1}
$$

Here $m(\mathbf{x})$ is the mass, $\mathbf{x} \in \mathbb{R}^m$ are the design variables, and $\mathbf{u}_k \in \mathbb{R}^n$ are the state variables for the load cases $k = 1, \ldots, K$ associated with the load vectors $\mathbf{f}_k$. The finite-element stiffness matrix $\mathbf{K}(\mathbf{x})$ is a function of the design variables and the design variables are bounded by the lower and upper values, $\mathbf{x}_l$, and $\mathbf{x}_u$, respectively. The stress constraint, $\sigma(\xi, \mathbf{x}, \mathbf{u}_k) \leq \sigma_{allow}$, is imposed for each load case at all points ξ in the domain Ω, such that $\xi \in \Omega$. The precise formulation of this constraint is presented in detail below.

In the following, we use either a direct parametrization, where the design variables, $\mathbf{x}$, represent member thicknesses, $\mathbf{t} \in \mathbb{R}^m$, such that $x_i = t_i$, or an inverse parametrization, where the thicknesses are inversely proportional to the design variables, $x_i = 1/t_i$. In either case the stiffness matrix is a linear combination of the member thicknesses as follows:

$$\mathbf{K}(\mathbf{x}) = \sum_{i=1}^{m} t_i \mathbf{K}_i.$$

Each matrix $\mathbf{K}_i$, satisfies $\mathbf{K}_i \succeq 0$, while the stiffness matrix satisfies $\mathbf{K}(\mathbf{x}) \succ 0$ at a feasible point. The structural mass can likewise be written as a function of the member thicknesses as follows:

$$m(\mathbf{x}) = \sum_{i=1}^{m} t_i m_i = \mathbf{t}^T \mathbf{m},$$

where $m(\mathbf{x})$ is the mass.

In the following, we formulate the stress constraints using linear matrix inequalities in a manner that ensures that the point-wise stress constraints are convex. The mass objective, using either direct or inverse parametrizations, is also a convex function of the design variables and the bound constraints are also trivially convex. However, non-convexity enters the optimization problem (1) through the governing equations which are bi-linear in the displacement vectors $\mathbf{u}_k$ and the thickness variables $\mathbf{t}$. Note that this constraint could be linearized about a point $\mathbf{x}$, $\mathbf{u}_k$ to obtain the following:

$$\mathbf{K}(\mathbf{x}) \Delta \mathbf{u}_k + \mathbf{A}(\mathbf{u}_k) \Delta \mathbf{x} = \mathbf{f}_k - \mathbf{K}(\mathbf{x}) \mathbf{u}_k,$$

where the linearization $\mathbf{A}(\mathbf{u}_k)$ is defined as follows:

$$\mathbf{A}(\mathbf{u}_k) = \frac{\partial \mathbf{K}(\mathbf{x}) \mathbf{u}_k}{\partial \mathbf{x}}.$$

This linearization would eliminate the bi-linearity in the governing equations and would make the optimization problem (1) convex locally. It would then be possible to solve a sequence of convex problems to obtain the solution to (1). However, we do not employ such an approach here and instead solve the non-convex mass-minimization problem directly.

In the following sections, we describe how we formulate the stress constraint as a linear matrix inequality. Throughout this development, we omit the dependence of the displacement solution on the load case number and simply write the displacements as $\mathbf{u}$.

A. Stress constraints as matrix inequalities

As a first approximation to the stress-constrained mass minimization problem (1), we impose the stress constraints at a set of predefined points $\{\xi_i\}_{i=1}^N \in \Omega$. While this is a common approach for formulating stress constraints, this method does not impose a bound on the stress anywhere except at $\{\xi_i\}_{i=1}^N$.

In this work, we implement stress constraints as matrix inequalities. Conventional stress constraints that can be modified to fit within this matrix-inequality framework can be written as a quadratic function of the components of stress, or strain, as follows:

$$1 - \mathbf{s}^T\mathbf{h} - \mathbf{s}^T\mathbf{G}\mathbf{s} \geq 0, \tag{2}$$

where $\mathbf{s}$ are scaled components of stress, and $\mathbf{h}$ and $\mathbf{G} = \mathbf{G}^T \succ 0$ are material- and failure-criteria-dependent constants. Note that the constraint on positive definite $\mathbf{G}$ is usually not restrictive since it is a requirement to obtain a closed failure surface.

Throughout the following, we focus on plane stress problems. However, the reformulation techniques presented here can also be extended to 3D problems using analogous methods. For plane stress problems, the scaled stress components, $\mathbf{s} \in \mathbb{R}^3$, can be written as follows:

$$\mathbf{s} = \begin{bmatrix} s_x \\ s_y \\ s_{xy} \end{bmatrix} = \frac{1}{\sigma_{scale}} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix}.$$

As a result, the failure criteria coefficients are a 3-vector $\mathbf{h} \in \mathbb{R}^3$ and a positive definite matrix $\mathbf{G} \in \mathbb{S}^3_{++}$.

The scaled stress components are evaluated within the finite-element model based on the nodal displacements, $\mathbf{u}$, and the location within the structural domain, $\xi \in \Omega$,

$$\mathbf{s}(\xi, \mathbf{u}) = \frac{1}{\sigma_{scale}} \mathbf{D}\mathbf{B}(\xi)\mathbf{u},$$

where $\mathbf{B}(\xi)$ computes the local strains as a function of the displacements and $\mathbf{D}$ is the constitutive tensor. The material tensor $\mathbf{D}$ could also depend on the position within the domain without affecting the analysis that follows.

Since the coefficient matrix $\mathbf{G}$ is positive definite, $\mathbf{G}^{-1}$ exists and is also positive definite. Therefore, the constraint (2), can be reformulated using a Schur-complement technique such that

$$\mathbf{W}(\xi, \mathbf{u}) = \begin{bmatrix} \mathbf{G}^{-1} & \mathbf{s} \\ \mathbf{s}^T & 1 - \mathbf{h}^T\mathbf{s} \end{bmatrix} \succeq 0. \tag{3}$$

The resulting matrix, $\mathbf{W}(\xi, \mathbf{u})$ is a linear function of the displacement variables $\mathbf{u}$. The resulting matrix can be scaled using a similarity transform that does not modify its eigenvalues. Therefore the form of the matrix inequality is not unique. In the following section, we demonstrate how both the von Mises stress criteria and the Tsai–Wu failure criteria can be reformulated as linear matrix inequalities.

1. Von Mises stress criterion

The von Mises stress constraint can be written as follows:

$$\sigma_{allow}^2 \geq \sigma_x^2 + \sigma_y^2 + 3\sigma_{xy}^2 - \sigma_x\sigma_y.$$

Using the scaling $\sigma_{scale} = \sigma_{allow}$ and introducing the scaled stress results in the following equivalent constraint:

$$1 - s_x^2 - s_y^2 - 3s_{xy}^2 + s_x s_y \geq 0. \tag{4}$$

Based on this expression, we can identify the corresponding $\mathbf{G}$ and $\mathbf{h}$ coefficient matrices

$$\mathbf{G} = \begin{bmatrix} 1 & -\frac{1}{2} & \\ -\frac{1}{2} & 1 & \\ & & 3 \end{bmatrix}, \qquad \mathbf{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Reformulating the expression using (3) gives the following linear matrix inequality:

$$\frac{1}{3}\begin{bmatrix} 4 & 2 & & 3s_x \\ 2 & 4 & & 3s_y \\ & & 1 & 3s_{xy} \\ 3s_x & 3s_y & 3s_{xy} & 3 \end{bmatrix} \succeq 0.$$

This matrix can be re-scaled to the equivalent matrix inequality:

$$\mathbf{W}_{VM}(\xi,\mathbf{u}) = \begin{bmatrix} 4 & 2 & & s_x \\ 2 & 4 & & s_y \\ & & 1 & s_{xy} \\ s_x & s_y & s_{xy} & \frac{1}{3} \end{bmatrix} \succeq 0.$$

2. Tsai–Wu criterion

The Tsai–Wu failure criterion [9] can also be formulated as a linear matrix inequality. In the analysis presented here, we assume that the orthotropic axis is aligned with the *x*-*y* coordinate frame. This assumption is not required to reformulate the Tsai–Wu criterion as a linear matrix inequality as long as the angle to the orthotropic material axis is not design-dependent.

The Tsai–Wu criterion can be written as follows:

$$1 - F_1\sigma_x - F_2\sigma_y - F_{11}\sigma_x^2 - F_{22}\sigma_y^2 - 2F_{12}\sigma_x\sigma_y - F_{66}\sigma_{xy}^2 \geq 0.$$

Within this context, the coefficients **h**, and **G** can be written as follows:

$$\mathbf{G} = \begin{bmatrix} F_{11} & F_{12} & \\ F_{12} & F_{22} & \\ & & F_{66} \end{bmatrix}, \qquad \mathbf{h} = \begin{bmatrix} F_1 \\ F_2 \\ 0 \end{bmatrix}.$$

The Tsai–Wu coefficient stability criterion [9, 22] for the coefficients requires $F_{11} > 0$, $F_{22} > 0$, $F_{66} > 0$, and F_{12} must satisfy

$$F_{12}^2 < F_{11}F_{22}.$$

As a result, the matrix **G** is positive definite and $\mathbf{G}^{-1}$ can be obtained as follows:

$$\mathbf{G}^{-1} = \frac{1}{F_{11}F_{22} - F_{12}^2} \begin{bmatrix} F_{22} & -F_{12} & \\ -F_{12} & F_{11} & \\ & & F_{66}^{-1}(F_{11}F_{22} - F_{12}^2). \end{bmatrix}.$$

Using the Schur-complement reformulation (3) gives the following linear matrix inequality for the unscaled components of stress:

$$\mathbf{W}_{TW}(\xi,\mathbf{u}) = \begin{bmatrix} F^{-1}F_{22} & -F^{-1}F_{12} & & \sigma_x \\ -F^{-1}F_{12} & F^{-1}F_{11} & & \sigma_y \\ & & F_{66}^{-1} & \sigma_{xy} \\ \sigma_x & \sigma_y & \sigma_{xy} & 1 - F_1\sigma_x - F_2\sigma_y \end{bmatrix} \succeq 0,$$

where $F = F_{11}F_{22} - F_{12}^2$.

Often, the Tsai–Wu criterion coefficients, F_*, will be small as a result of the units in which they are expressed. Instead of using conventional units, the constraint itself can be re-scaled to avoid poor scaling. The scaled Tsai–Wu criterion can be written as follows:

$$1 - \tilde{F}_1 s_x - \tilde{F}_2 s_y - \tilde{F}_{11}s_x^2 - \tilde{F}_{22}s_y^2 - 2\tilde{F}_{12}s_x s_y - \tilde{F}_{66}s_{xy}^2 \geq 0,$$

where $\sigma_{scale} = X$, where X is the tensile failure load along the 1-direction. As a result of this scaling, the coefficients of the failure criterion become $\tilde{F}_1 = \sigma_s F_1$, $\tilde{F}_2 = \sigma_s F_2$, $\tilde{F}_{11} = \sigma_s^2 F_{11}$, $\tilde{F}_{22} = \sigma_s^2 F_{22}$, $\tilde{F}_{66} = \sigma_s^2 F_{66}$, and $\tilde{F}_{12} = \sigma_s^2 F_{12}$. The Tsai–Wu criterion can now be written as:

$$\mathbf{W}_{TW}(\xi,\mathbf{u}) = \begin{bmatrix} \tilde{F}^{-1}\tilde{F}_{22} & -\tilde{F}^{-1}\tilde{F}_{12} & & s_x \\ -\tilde{F}^{-1}\tilde{F}_{12} & \tilde{F}^{-1}\tilde{F}_{11} & & s_y \\ & & \tilde{F}_{66}^{-1} & s_{xy} \\ s_x & s_y & s_{xy} & 1 - \tilde{F}_1 s_x - \tilde{F}_2 s_y \end{bmatrix} \succeq 0,$$

where $\tilde{F} = \tilde{F}_{11}\tilde{F}_{22} - \tilde{F}_{12}^2$.

B. ε-relaxation formulations

The stress singularity problem arises when considering simultaneous sizing and topology problems where the lower bound thickness approaches zero, producing design spaces with isolated optimum points. To remove this degeneracy, we propose an ε-relaxation approach based on the work of Cheng [5] and Cheng and Guo [6]. The essential idea of the ε-relaxation approach is that as the thickness of the member approaches zero, the stress should be permitted to exceed the allowable stress. In this context, an ε-relaxation approach can be written as follows:

$$1 + \frac{\varepsilon}{t} - \mathbf{h}^T\mathbf{s} - \mathbf{s}^T\mathbf{G}\mathbf{s} \geq 0,$$

where ε is a small parameter, and t is a local thickness variable constrained such that $t \geq \varepsilon^2$. As a result, as $t \to \varepsilon^2$, the constraint becomes:

$$1 + \frac{1}{\varepsilon} - \mathbf{h}^T\mathbf{s} - \mathbf{s}^T\mathbf{G}\mathbf{s} \geq 0.$$

This constraint will be satisfied for all members with thickness at the lower bound for ε sufficiently small. Using inverse design variables written as $x = 1/t$, we arrive at the following constraint:

$$1 + \varepsilon x - \mathbf{h}^T\mathbf{s} - \mathbf{s}^T\mathbf{G}\mathbf{s} \geq 0,$$

where the lower bound on $t \geq \varepsilon^2$ is replaced with an upper bound on $x \leq \varepsilon^{-2}$. As a result of this transformation, the inequality constraints can now be written as follows:

$$\mathbf{W}(\xi, \mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{G}^{-1} & \mathbf{s} \\ \mathbf{s}^T & 1 + \varepsilon x - \mathbf{h}^T\mathbf{s} \end{bmatrix} \succeq 0.$$

Note that this ε-relaxation formulation must be used in conjunction with the inverse thickness parametrization. However, this formulation leads to a linear, rather than nonlinear, matrix inequality.

III. Matrix aggregates

In this section, we introduce aggregation methods that operate on positive semi-definite matrices. We call these aggregation methods *matrix aggregates*. The advantage of these matrix aggregation methods is that they maintain convexity of the linear matrix inequality constraints formulated above, despite their nonlinear form.

We formulate two types of related matrix aggregates: discrete and continuous. The discrete aggregates are designed to enforce the stress constraints over a series of discrete points within the domain, $\mathscr{P} = \{\xi_i\}_{i=1}^N$, such that $\sigma(\xi_i) \leq \sigma_{allow} \;\; \forall \xi_i \in \mathscr{P}$, whereas the continuous aggregates are designed to enforce the stress constraint everywhere in the aggregation domain Ω_A, such that $\sigma(\xi) \leq \sigma_{allow} \;\; \forall \xi \in \Omega_A$. Both of these aggregation methods take a similar form to Kreisselmeier–Steinhauser (KS) aggregation techniques [14, 25, 11, 10].

In the following, we make extensive use of the matrix exponential and matrix logarithm on symmetric matrices. Given $\mathbf{A} \in \mathbb{S}^n$, with the eigenvalue decomposition $\mathbf{A} = \mathbf{T}\Lambda\mathbf{T}^T$, where $\mathbf{T} \in \mathbb{R}^{n\times n}$ is a real, orthonormal matrix such that $\mathbf{T}\mathbf{T}^T = \mathbf{I}$ and $\Lambda \in \mathbb{R}^{n\times n}$ are the eigenvalues such that $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$, the matrix exponential, $\exp : \mathbb{S}^n \to \mathbb{S}^n_+$, is defined as follows:

$$\exp \mathbf{A} = \mathbf{T}(\exp \Lambda)\mathbf{T}^T$$

where $\exp \Lambda = \text{diag}\{e^{\lambda_1}, e^{\lambda_2}, \dots, e^{\lambda_n}\}$. Likewise, the matrix logarithm, $\ln : \mathbb{S}^n_+ \to \mathbb{S}^n$, is defined in an analogous manner as follows:

$$\ln \mathbf{A} = \mathbf{T}(\ln \Lambda)\mathbf{T}^T,$$

where $\ln \Lambda = \text{diag}\{\ln \lambda_1, \ln \lambda_2, \dots, \ln \lambda_n\}$.

A. Discrete matrix aggregates

The discrete matrix aggregate takes the following form:

$$\mathbf{W}_A(\mathbf{x}, \mathbf{u}; \rho) = -\frac{1}{\rho} \ln \left[\sum_{i=1}^{N} \exp\left(-\rho \mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u})\right) \right] \succeq 0, \qquad (5)$$

where $\mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u})$ are the linear matrix inequality constraints evaluated at a number of points, ξ_i within the structural domain. The parameter ρ serves an equivalent role as the KS parameter for classical aggregation methods.

To demonstrate that the constraint (5) is equivalent to the point-wise constraints

$$\mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u}) \succeq 0, \qquad i = 1, \dots, N,$$

we show that the minimum eigenvalue from all the point-wise constraints is bounded from below by the minimum eigenvalue of the constraint aggregate, such that

$$\min_i \min_j \lambda_j(\mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u}) \geq \min_i \lambda_i(\mathbf{W}_A). \tag{6}$$

As a result, enforcing $\mathbf{W}_A(\mathbf{x}, \mathbf{u}; \rho) \succeq 0$, imposes $\mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u}) \succeq 0$ for all $i = 1, \dots, N$.

To illustrate this property, first consider the eigenvalue decomposition of the individual matrices as follows:

$$\mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u}) = \mathbf{T}_i \Lambda_i \mathbf{T}_i^T.$$

Using this decomposition, and denoting the individual eigenvalues as $\lambda_{ij} = \lambda_j(\mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u}))$, the matrix exponential for each term in the summation in the matrix aggregate (5) can be written as follows:

$$\exp\left(-\rho \mathbf{W}(\xi_i, \mathbf{x}, \mathbf{u})\right) = \sum_{j=1}^{n} e^{-\rho \lambda_{ij}} \, \mathbf{t}_{ij} \mathbf{t}_{ij}^T,$$

where $\mathbf{t}_{ij}$ is the j-th column, or eigenvector, of the matrix $\mathbf{T}_i$ and λ_{ij} is its corresponding eigenvalue. As a result, the matrix aggregate can now be written as:

$$\mathbf{W}_A(\mathbf{x}, \mathbf{u}; \rho) = -\frac{1}{\rho} \ln \left[\sum_{i=1}^{N} \sum_{j=1}^{n} e^{-\rho \lambda_{ij}} \, \mathbf{t}_{ij} \mathbf{t}_{ij}^T \right].$$

Next, consider the set of index pairs, (i, j), such that $\lambda_{ij} = \lambda_{\min} = \min_i \min_j \lambda_{ij}$, and denote this set $\mathscr{I}$, defined as follows:

$$\mathscr{I} = \{ \, (i, j) \mid \lambda_{ij} = \lambda_{\min} \, \}.$$

Furthermore, define the vector $\mathbf{v}$ as the sum of the eigenvectors corresponding to all $(i, j) \in \mathscr{I}$ such that

$$\mathbf{v} = \sum_{i,j \in \mathscr{I}} \mathbf{t}_{ij},$$

where $\mathbf{v}^T \mathbf{v} \geq 1$ due to the orthonormal nature of each of the eigenvectors $\mathbf{t}_{ij}$.

Using these definitions, the summation in the matrix aggregate (5), can be divided into the sum over all eigenvalues that attain the minimum value and the remaining terms:

$$\begin{aligned}
\mathbf{W}_A(\mathbf{x}, \mathbf{u}; \rho) &= -\frac{1}{\rho} \ln \left[\sum_{i,j \in \mathscr{I}} e^{-\rho \lambda_{ij}} \, \mathbf{t}_{ij} \mathbf{t}_{ij}^T + \sum_{i,j \notin \mathscr{I}} e^{-\rho \lambda_{ij}} \, \mathbf{t}_{ij} \mathbf{t}_{ij}^T \right], \\
&= -\frac{1}{\rho} \ln \left[e^{-\rho \lambda_{\min}} \mathbf{v} \mathbf{v}^T + \sum_{i,j \notin \mathscr{I}} e^{-\rho \lambda_{ij}} \, \mathbf{t}_{ij} \mathbf{t}_{ij}^T \right], \\
&= \lambda_{\min} \mathbf{I} - \frac{1}{\rho} \ln \left[\mathbf{v} \mathbf{v}^T + \sum_{i,j \notin \mathscr{I}} e^{-\rho (\lambda_{ij} - \lambda_{\min})} \, \mathbf{t}_{ij} \mathbf{t}_{ij}^T \right], \\
&= \lambda_{\min} \mathbf{I} - \frac{1}{\rho} \ln \left[\mathbf{v} \mathbf{v}^T + \mathbf{R} \right],
\end{aligned}$$

where in the last step we have applied a definition for $\mathbf{R}$. The eigenvalues of the matrix aggregate can now be written as follows:

$$\lambda_j(\mathbf{W}_A(\mathbf{x}, \mathbf{u}; \rho)) = \lambda_{\min} + \frac{1}{\rho} \ln \lambda\left(\mathbf{v} \mathbf{v}^T + \mathbf{R}\right). \tag{7}$$

To obtain a bound on these eigenvalues, first note that in the limit $\lim_{\rho \to \infty} \mathbf{R} = 0$. Therefore, at some large but finite ρ, the matrix $\mathbf{R}$ will satisfy $||\mathbf{R}|| < \varepsilon \ll 1$. The eigenvalues of the matrix $[\mathbf{v} \mathbf{v}^T + \mathbf{R}]$ can be approximated based on this observation. First consider the vector $\mathbf{v}$ such that

$$\left[\mathbf{v} \mathbf{v}^T + \mathbf{R} \right] \mathbf{v} \approx \lambda \mathbf{v}.$$

This implies that $\lambda \approx \mathbf{v}^T\mathbf{v}$ is an approximate eigenvalue. Furthermore, the remaining eigenvectors, $\mathbf{w}$, must satisfy $\mathbf{w} \in \mathbf{v}^{\perp}$ with $\mathbf{w}^T\mathbf{w} = 1$. Then

$$\left[\mathbf{v}\mathbf{v}^T + \mathbf{B}\right]\mathbf{w} = \mathbf{B}\mathbf{w} = \lambda\mathbf{w}$$

therefore $\lambda = \mathbf{w}^T\mathbf{R}\mathbf{w}$, but $\mathbf{w}^T\mathbf{R}\mathbf{w} \leq \varepsilon \ll 1$. As a result, the eigenvalues of the matrix $\mathbf{v}\mathbf{v}^T + \mathbf{R}$ will consist of one eigenvalue that is approximately $\mathbf{v}^T\mathbf{v} \geq 1$, and, for sufficiently large ρ, the remaining eigenvalues will satisfy $\lambda \leq \varepsilon$. As a result, the eigenvalues of the aggregation matrix can be written as follows:

$$\min_i \lambda_i(\mathbf{W}_A(\mathbf{x},\mathbf{u};\rho)) \approx \lambda_{\min} - \frac{1}{\rho}\ln \mathbf{v}^T\mathbf{v} \leq \lambda_{\min},$$

and all remaining eigenvalues will satisfy

$$\lambda(\mathbf{W}_A(\mathbf{x},\mathbf{u};\rho)) > \lambda_{\min} - \frac{1}{\rho}\ln\varepsilon \geq \lambda_{\min}.$$

Therefore, we have established that

$$\lambda_{\min} = \min_i \min_j \lambda_j(\mathbf{W}(\xi_i,\mathbf{x},\mathbf{u}) \geq \min_i \lambda_i(\mathbf{W}_A(\mathbf{x},\mathbf{u};\rho))$$

which verifies (6).

B. Continuous matrix aggregates

A continuous variant of the discrete matrix aggregate (5), can be obtained by replacing the discrete sum over a set of points with an integral over a subset of the structural domain $\Omega_A \subset \Omega$. The continuous matrix aggregate takes the form:

$$\mathbf{W}_A(\mathbf{x},\mathbf{u};\rho) = -\frac{1}{\rho}\ln\left[-\frac{1}{\alpha}\int_{\Omega_A}\exp\left(\rho\mathbf{W}(\xi,\mathbf{x},\mathbf{u})\right)d\Omega\right]. \tag{8}$$

In this work, we have not used continuous matrix aggregates, but they are analogous to continuous aggregation techniques for scalar functions [11, 10].

IV. Full space barrier methods

In this section, we present full space barrier methods for stress-constrained mass minimization problems which take the following form:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & m(\mathbf{x}) \\ \text{governed by} \quad & \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f} \\ \text{such that} \quad & \mathbf{x} \geq 0 \\ & \mathbf{W}_i(\mathbf{x},\mathbf{u}) \succeq 0 \qquad i = 1,\ldots,N \end{aligned} \tag{9}$$

This problem is simplified from (1) by considering a single load case, and imposing only non-negative bounds on the design variables. The extension of the methods presented in this section to the optimization problem (1) is straightforward.

The principal difference between this problem statement and many conventional methods is the presence of the matrix inequalities, $\mathbf{W}_i(\mathbf{x},\mathbf{u}) \succeq 0$. These inequalities may be either the point-wise inequalities (3), or matrix aggregates, either (5) or (8). To solve the optimization problem (9), we use a barrier method where the governing equations are treated as equality constraints, and the non-negativity bounds and matrix inequalities are handled using barrier functions. In particular, we use the self-concordant barriers $\ln x$ for the bound constraints, and $\ln\det\mathbf{A}$ for the matrix inequalities [4, 19, 28]. This results in the following barrier problem

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & m(\mathbf{x}) - \mu\sum_{i=1}^{n}\ln x_i - \mu\sum_{i=1}^{N}\ln\det\mathbf{W}_i \\ \text{governed by} \quad & \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f} \end{aligned} \tag{10}$$

where μ is the barrier parameter. Note that for ease of presentation, we discard the dependence of the matrix inequalities $\mathbf{W}_i$ on the design variables, $\mathbf{x}$, and the displacements $\mathbf{u}$.

As $\mu \to 0$, the solution of the barrier problem (10) approaches the solution of the original problem (9). In the barrier method, an initial value of μ is selected and the resulting barrier problem is solved to a specified tolerance at which point the barrier parameter is reduced by a constant fraction and the sequence is repeated until μ is sufficiently small. The barrier method requires that all iterates must remain strictly feasible throughout the optimization such that $\mathbf{x} > 0$ and $\mathbf{W}_i \succ 0$. A feasible starting point satisfying these conditions can be obtained by selecting the initial design point with sufficiently large thicknesses, and solving the governing equations to obtain a pair $\mathbf{x}$, $\mathbf{u}$ such that the stress constraints are satisfied everywhere within the domain. In the barrier method, the feasibility of all subsequent iterates with respect to the inequality constraints is guaranteed through the line search procedure described below.

The Lagrangian of the barrier problem (10) is

$$\mathscr{L}(\mathbf{x},\mathbf{u},\psi) = m(\mathbf{x}) - \mu \sum_{i=1}^{n} \ln x_i - \mu \sum_{i=1}^{N} \ln\det \mathbf{W}_i + \psi^T (\mathbf{K}(\mathbf{x})\mathbf{u} - \mathbf{f}), \tag{11}$$

where the Lagrange multipliers ψ have been introduced to enforce the governing equations. The KKT conditions for the barrier problem (10) can be obtained by differentiating the Lagrangian with respect to the design variables, $\mathbf{x}$, displacement vector, $\mathbf{u}$, and Lagrange multipliers, ψ, which yields the following system of equations:

$$\begin{aligned} \mathbf{r}_{\mathbf{x}} &= \nabla_{\mathbf{x}} m - \mu \mathbf{X}^{-1}\mathbf{e} - \mu \mathbf{t}_{\mathbf{x}} + \mathbf{A}(\mathbf{u})^T \psi, \\ \mathbf{r}_{\mathbf{u}} &= \mathbf{K}(\mathbf{x})\psi - \mu \mathbf{t}_{\mathbf{u}}, \\ \mathbf{r}_{\psi} &= \mathbf{K}(\mathbf{x})\mathbf{u} - \mathbf{f}. \end{aligned} \tag{12}$$

Note that $\mathbf{X} = \mathrm{diag}\{\mathbf{x}\}$ is a diagonal matrix with the design variable value entries along its diagonal, and $\mathbf{e}$ is a vector of all unit entries. The terms $\mathbf{t}_{\mathbf{x}}$ and $\mathbf{t}_{\mathbf{u}}$ are obtained by differentiating the barrier for the matrix inequalities, $\sum_{i=1}^{N} \ln\det \mathbf{W}_i$. Note that the derivative of the expression $\ln\det \mathbf{A}(\mathbf{x})$ is given by

$$\frac{\partial}{\partial x_i} (\ln\det \mathbf{A}(\mathbf{x})) = \mathrm{tr}\left(\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x_i} \right),$$

where $\mathrm{tr}(\cdot)$ is the trace operator. As a result, $\mathbf{t}_{\mathbf{x}}$ and $\mathbf{t}_{\mathbf{u}}$ can be obtained as follows:

$$\begin{aligned} [\mathbf{t}_{\mathbf{u}}]_j &= \sum_{i=1}^{N} \mathrm{tr}\left(\mathbf{W}_i^{-1} \frac{\partial \mathbf{W}_i}{\partial u_j} \right), \\ [\mathbf{t}_{\mathbf{x}}]_j &= \sum_{i=1}^{N} \mathrm{tr}\left(\mathbf{W}_i^{-1} \frac{\partial \mathbf{W}_i}{\partial x_j} \right). \end{aligned}$$

While this expression requires the computation of $\mathbf{W}_i^{-1}$, each matrix $\mathbf{W}_i$ is small and positive definite such that $\mathbf{W}_i \in \mathbb{S}^4_{++}$ due to the strict feasibility requirement of the barrier method. Therefore, computing these derivatives is not computationally expensive compared with other operations required during the optimization.

At each iteration of the full space barrier method, a Newton step is computed by solving a linearization of the KKT conditions for the barrier problem (12). This linearization results in the following system of equations:

$$\begin{bmatrix} \nabla_{\mathbf{xx}} m + \mu \mathbf{X}^{-2} + \mu \mathbf{C}_{\mathbf{xx}} & \mathbf{A}(\psi)^T + \mu \mathbf{C}_{\mathbf{ux}}^T & \mathbf{A}(\mathbf{u})^T \\ \mathbf{A}(\psi) + \mu \mathbf{C}_{\mathbf{ux}} & \mu \mathbf{C}_{\mathbf{uu}} & \mathbf{K}(\mathbf{x}) \\ \mathbf{A}(\mathbf{u}) & \mathbf{K}(\mathbf{x}) & \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{u} \\ \Delta \psi \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathbf{x}} \\ \mathbf{r}_{\mathbf{u}} \\ \mathbf{r}_{\psi} \end{bmatrix}, \tag{13}$$

where $\Delta\mathbf{x}$, $\Delta\mathbf{u}$ and $\Delta\psi$ are the steps in the design variables, displacements, and Lagrange multiplier variables, respectively. Within the linear system (13), the matrix $\mathbf{A}(\mathbf{u})$ is obtained through a linearization of the governing equations with respect to the design variables as follows:

$$\mathbf{A}(\mathbf{u}) = \frac{\partial \mathbf{K}(\mathbf{x})\mathbf{u}}{\partial \mathbf{x}},$$

where $\mathbf{A}(\psi)$ is obtained by substitution of ψ for $\mathbf{u}$. Note also that $\psi^T \mathbf{A}(\mathbf{u}) = \mathbf{u}^T \mathbf{A}(\psi)$. The terms $\mathbf{C}_{\mathbf{xx}}$, $\mathbf{C}_{\mathbf{ux}}$, and $\mathbf{C}_{\mathbf{uu}}$ are obtained from the second derivatives of the barrier term for the matrix inequality. Note that the second derivative of $\ln\det \mathbf{A}(\mathbf{x})$ can be written as follows:

$$\frac{\partial^2}{\partial x_i \partial x_j} (\ln\det \mathbf{A}(\mathbf{x})) = \mathrm{tr}\left(\mathbf{A}^{-1} \frac{\partial^2 \mathbf{A}}{\partial x_i \partial x_j} \right) - \mathrm{tr}\left(\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x_i} \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x_j} \right).$$

As a result, the second derivatives of the term $-\sum_{i=1}^{N} \ln\det \mathbf{W}_i$ can be written as follows:

$$[\mathbf{C_{xx}}]_{jk} = \sum_{i=1}^{N}\left[\mathrm{tr}\left(W_i^{-1}\frac{\partial \mathbf{W}_i}{\partial x_j}W_i^{-1}\frac{\partial \mathbf{W}_i}{\partial x_k}\right) - \mathrm{tr}\left(\mathbf{W}_i^{-1}\frac{\partial^2 \mathbf{W}_i}{\partial x_j \partial x_k}\right)\right],$$
$$[\mathbf{C_{ux}}]_{jk} = \sum_{i=1}^{N}\left[\mathrm{tr}\left(W_i^{-1}\frac{\partial \mathbf{W}_i}{\partial u_j}W_i^{-1}\frac{\partial \mathbf{W}_i}{\partial x_k}\right) - \mathrm{tr}\left(\mathbf{W}_i^{-1}\frac{\partial^2 \mathbf{W}_i}{\partial u_j \partial x_k}\right)\right],$$
$$[\mathbf{C_{uu}}]_{jk} = \sum_{i=1}^{N}\left[\mathrm{tr}\left(W_i^{-1}\frac{\partial \mathbf{W}_i}{\partial u_j}W_i^{-1}\frac{\partial \mathbf{W}_i}{\partial u_k}\right) - \mathrm{tr}\left(\mathbf{W}_i^{-1}\frac{\partial^2 \mathbf{W}_i}{\partial u_j \partial u_k}\right)\right].$$

What may not be clear from these expressions is that if the matrix inequality $\mathbf{W}_i(\mathbf{x},\mathbf{u})$ is defined over each finite-element in the mesh, then $\mathbf{C_{uu}}$ will have the same sparsity pattern as the stiffness matrix $\mathbf{K}(\mathbf{x})$. Furthermore, when the matrix inequalities are linear the second-derivatives in the above expressions vanish.

A. Solving the Newton system

Following Biros and Ghattas [2, 3] and Hicken [7], we use a Krylov subspace method to solve the Newton system (13) at each optimization iteration. Since the linearization of the KKT system is indefinite, a Krylov method that is capable of handling indefiniteness is required. For this purpose, we use GMRES [27], and when required, flexible GMRES [26].

We have implemented three preconditioners for the linearized KKT system based on block factorizations of the KKT system obtained by discarding certain blocks. The first two preconditioners are similar to the $\mathbf{P}_2$ and $\tilde{\mathbf{P}}_2$ preconditioners described by Biros and Ghattas [2]. The key difference between the preconditioners proposed here and those presented by Biros and Ghattas [2] is that we do not use a quasi-Newton approximation for the diagonal block associated with the design variables. The third preconditioner is designed to exploit the capability to compute exact second derivatives of the matrix inequality barrier term.

The first preconditioner, $\mathbf{P}_2$ is obtained by discarding blocks from (13) and retaining an exact linearization of $\mathbf{K}(\mathbf{x})$ as follows:

$$\mathbf{P}_2 = \begin{bmatrix} \nabla_{\mathbf{xx}} m + \mu \mathbf{X}^{-2} & & \mathbf{A}(\mathbf{u})^T \\ & & \mathbf{K}(\mathbf{x}) \\ \mathbf{A}(\mathbf{u}) & \mathbf{K}(\mathbf{x}) & \end{bmatrix}.$$

As a result, the application of $\mathbf{P}_2$ on a right-hand-side, $\mathbf{b} = \begin{bmatrix}\mathbf{b_x} & \mathbf{b_u} & \mathbf{b}_\psi\end{bmatrix}^T$, can be obtained in the following steps. First, find the update for the Lagrange multipliers, $\Delta\psi$, by solving

$$\mathbf{K}(\mathbf{x})\Delta\psi = \mathbf{b_u},$$

then obtain the update for the design variables, $\Delta\mathbf{x}$, by solving

$$\left[\nabla_{\mathbf{xx}} m + \mu \mathbf{X}^{-2}\right]\Delta\mathbf{x} = \mathbf{b_x} - \mathbf{A}(\mathbf{u})^T\Delta\psi,$$

and finally obtain the update for the displacements, $\Delta\mathbf{u}$, by solving

$$\mathbf{K}(\mathbf{x})\Delta\mathbf{u} = \mathbf{b}_\psi - \mathbf{A}(\mathbf{u})\Delta\mathbf{x}.$$

Each step of the application of $\mathbf{P}_2$ requires an exact solution of a linear system. To reduce computational costs, these exact solutions can be replaced by the application of a preconditioner for $\mathbf{K}(\mathbf{x})$. This modification results in the second preconditioner, $\tilde{\mathbf{P}}_2$, which can be written as follows:

$$\tilde{\mathbf{P}}_2 = \begin{bmatrix} \nabla_{\mathbf{xx}} m + \mu \mathbf{X}^{-2} & & \mathbf{A}(\mathbf{u})^T \\ & & \tilde{\mathbf{K}} \\ \mathbf{A}(\mathbf{u}) & \tilde{\mathbf{K}} & \end{bmatrix},$$

where $\tilde{\mathbf{K}}$ is a preconditioner for the matrix $\mathbf{K}(\mathbf{x})$. The application of $\tilde{\mathbf{P}}_2$ requires the same sequence of steps as $\mathbf{P}_2$ where each solution with the matrix $\mathbf{K}(\mathbf{x})$ is replaced by the action of $\tilde{\mathbf{K}}^{-1}$ on the given right-hand-side.

The last preconditioner is obtained based on the following block structure:

$$\tilde{\mathbf{P}}_3 = \begin{bmatrix} \nabla_{\mathbf{xx}} m + \mu \mathbf{X}^{-2} + \mu \mathbf{C_{xx}} & & \\ \mathbf{A}(\psi) + \mu \mathbf{C_{ux}} & \mu \mathbf{C_{uu}} & \tilde{\mathbf{K}} \\ \mathbf{A}(\mathbf{u}) & \tilde{\mathbf{K}} & \end{bmatrix},$$

where $\tilde{\mathbf{K}}$ is again a preconditioner for $\mathbf{K}(\mathbf{x})$. In this case, the action of the preconditioner on a right-hand-side, $\mathbf{b}$, can be obtained in the following steps. First, find the update for the design variables, $\Delta\mathbf{x}$, by solving

$$\left[\nabla_{\mathbf{xx}} m + \mu \mathbf{X}^{-2} + \mu \mathbf{C}_{\mathbf{xx}}\right] \Delta\mathbf{x}, = \mathbf{b}_{\mathbf{x}}$$

then obtain the update for the displacements, $\Delta\mathbf{u}$, by solving

$$\tilde{\mathbf{K}}\Delta\mathbf{u} = \mathbf{b}_{\psi} - \mathbf{A}(\mathbf{u})\Delta\mathbf{x},$$

and finally obtain the update for the Lagrange multipliers, $\Delta\psi$, by solving

$$\tilde{\mathbf{K}}\Delta\psi = \mathbf{b}_{\mathbf{u}} - \mu \mathbf{C}_{\mathbf{uu}}\Delta\mathbf{u} - \left[\mathbf{A}(\psi) + \mu \mathbf{C}_{\mathbf{ux}}\right]\Delta\mathbf{x}.$$

The first step of the application of $\tilde{\mathbf{P}}_3$ requires the solution of an equation with the matrix $\left[\nabla_{\mathbf{xx}} m + \mu \mathbf{X}^{-2} + \mu \mathbf{C}_{\mathbf{xx}}\right]$. This matrix is sparse and may be diagonal depending on the manner in which the matrix inequalities $\mathbf{W}_i(\mathbf{x},\mathbf{u})$ depend on the design variables.

Note that each of these preconditioners are also compatible with multiple load cases, where each load case can be solved concurrently at each step of the preconditioner. This provides both an opportunity to exploit parallelism as well as an opportunity to use block-based methods to improve computational performance [12].

B. Line search globalization

We globalize the Newton method for the barrier problem using a line search. The line search is based on an exact ℓ_2 merit function. To define the merit function, we first introduce the function ϕ as follows:

$$\phi(\mathbf{x},\mathbf{u};\gamma) = m(\mathbf{x}) - \mu \sum_{i=1}^{n} \ln x_i - \mu \sum_{i=1}^{N} \ln\det \mathbf{W}_i + \gamma ||\mathbf{K}(\mathbf{x})\mathbf{u} - \mathbf{f}||_2 \qquad (14)$$

which is a function of the design variables, $\mathbf{x}$, and the displacements, $\mathbf{u}$, but not the Lagrange multipliers. The parameter γ is a penalty parameter that is selected at each iteration to ensure a sufficiently negative directional derivative along the search direction [20]. The merit function itself is defined using (14) as follows:

$$\varphi(\alpha;\gamma) = \phi(\mathbf{x} + \alpha\Delta\mathbf{x}, \mathbf{u} + \alpha\Delta\mathbf{u}; \gamma), \qquad (15)$$

where $\Delta\mathbf{x}$ and $\Delta\mathbf{u}$ are the steps computed from the linearized KKT system (13).

A back-tracking line search is used that seeks a point that satisfies the first Wolfe condition

$$\varphi(\alpha;\gamma) < \varphi(0;\gamma) + \alpha c_1 \varphi'(0;\gamma),$$

where we typically choose $c_1 = 10^{-4}$. To prevent the iterate from entering an infeasible region of the design space, we limit the initial step size based on a fixed fraction to the feasible boundary. For the bound constraints, this step length is computed as

$$\alpha_{\max} = \max\left\{\alpha \in (0,1] \mid \mathbf{x} + \alpha\Delta\mathbf{x} \geq (1-\tau)\mathbf{x}\right\},$$

while for the matrix inequalities, this step length is computed as

$$\alpha_{\max} = \max\left\{\alpha \in (0,1] \mid \det\mathbf{W}_i(\mathbf{x} + \alpha\Delta\mathbf{x}, \mathbf{u} + \alpha\Delta\mathbf{u}) \geq (1-\tau)\det\mathbf{W}_i(\mathbf{x},\mathbf{u}),\ i = 1,\dots,N\right\}.$$

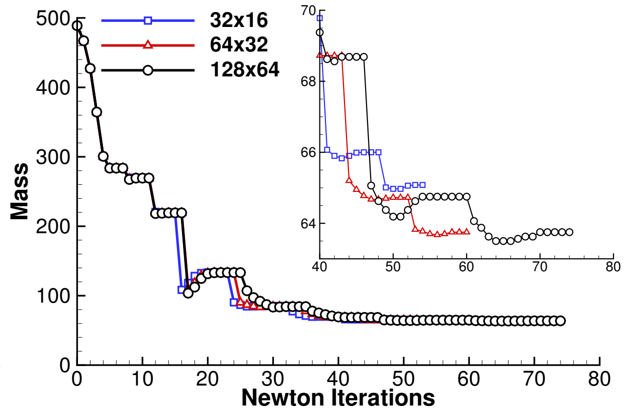
The minimum of the two step lengths is set as the initial maximum step length for the back-tracking line search method.

V. Results

In this section, we present results using the full-space barrier method described above. To demonstrate the proposed method, we solve a minimum mass plane-stress design optimization problem subject to a von Mises stress criterion. The problem domain, dimensions, and loading conditions are shown in Figure 1a. Within the problem, we use normalized material properties and set the Young's modulus to a value of $E = 70000$, the Poisson ratio to a value of $\nu = 0.3$, and the allowable stress to a value of $\sigma_{allow} = 100$. The problem domain is discretized using either 32×16, 64×32 or 128×64 bilinear finite-elements, with a uniform thickness over each element.

Within the full-space method we use a multigrid preconditioner for $\mathbf{K}(\mathbf{x})$, in which we apply a single V-cycle of multigrid over 6 mesh levels. We use FGMRES(250) preconditioned with $\tilde{\mathbf{P}}_2$ to solve the Newton update (13). We

(a) Design domain for the plane stress variable thickness sheet test problem.

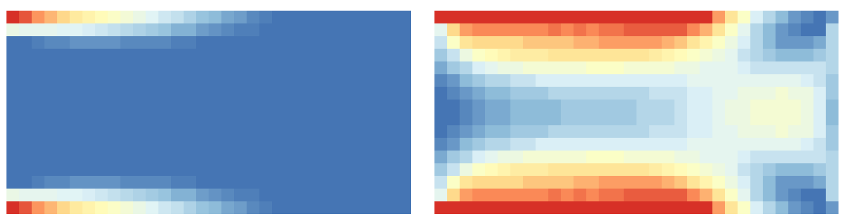


(b) Mass vs. optimization iteration for the 32×16, 64×32 and 128×64 meshes.

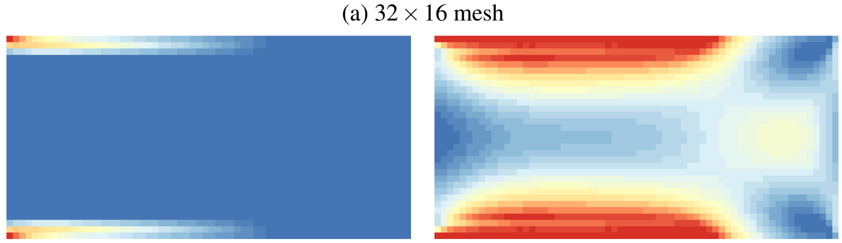
Figure 1: Design domain and mass history for the variable thickness sheet design problem.

note that similar performance results were obtained with the preconditioner $\mathbf{P}_3$, while $\mathbf{P}_2$ required the same number of iterations but took more computational time due to the linear solutions required for each preconditioner application.

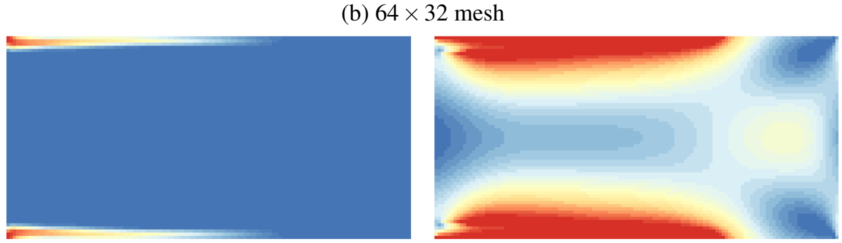
Figure 1b shows the mass for each design over the course of the optimization. The mass of all designs follow similar trends with small optimized mass for the structures with more-refined meshes. Figure 2 shows the distribution of the thicknesses and the von Mises stress over the structural domain for the optimized design. As can be seen from Figure 2, the stress constraint is active near the upper and lower corners of the clamped edge.



(a) 32×16 mesh



(b) 64×32 mesh



(c) 128×64 mesh

Figure 2: Distribution of the thickness and von Mises stress for the optimized design for the 128×64 element mesh.

In addition to the continuous thickness optimization results, the full-space barrier method was also applied to stress-constrained topology optimization. Figure 3 shows the optimization history for the design and stress as a function of optimization iteration for the 128×64 element mesh. The design converges to a well-defined truss within 130 optimization iterations.

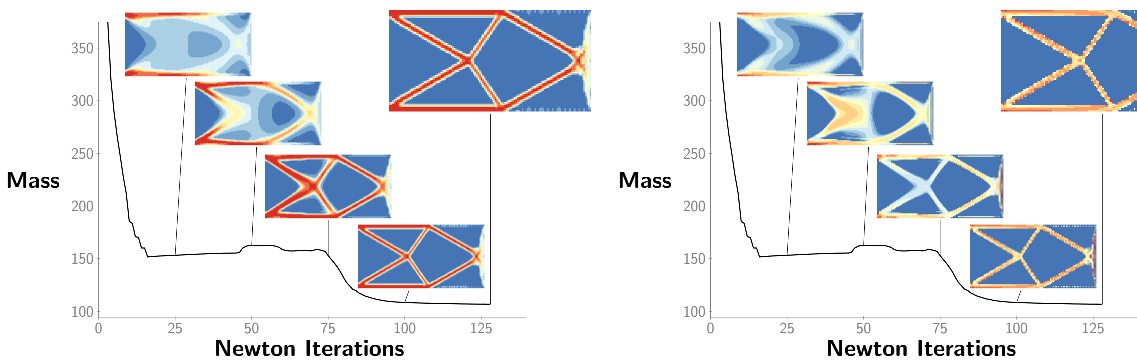


Figure 3: Design and von Mises stress histories for the topology optimization problem for the 128×64 element mesh.

VI. Conclusions

In this paper, we have presented a full-space barrier method that is designed to address the computational costs of stress-constrained mass minimization problems. We presented linear matrix inequality stress-constraint formulations for both the von Mises and Tsai–Wu failure criterion. In addition, we proposed matrix aggregates, which are analogous to classical aggregation methods, that can be used to aggregate matrix inequalities. Finally, we presented preliminary results of a stress-constrained mass minimization study using the proposed full-space barrier method.

Acknowledgment

The research reported in this paper was supported by the Georgia Institute of Technology.

References

[1] M. A. Akgun, R. T. Haftka, K. C. Wu, J. L. Walsh, and J. H. Garcelon. Efficient structural optimization for multiple load cases using adjoint sensitivities. *AIAA Journal*, 39(3):511–516, 2001. doi:10.2514/2.1336.

[2] G. Biros and O. Ghattas. Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part I: The Krylov–Schur solver. *SIAM Journal on Scientific Computing*, 27(2):687–713, 2005. doi:10.1137/S106482750241565X.

[3] G. Biros and O. Ghattas. Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part II: The Lagrange–Newton solver and its application to optimal control of steady viscous flows. *SIAM Journal on Scientific Computing*, 27(2):714–739, 2005. doi:10.1137/S1064827502415661.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 9780521833783.

[5] G. Cheng. Some aspects of truss topology optimization. *Structural and Multidisciplinary Optimization*, 10:173–179, 1995. ISSN 1615-147X. doi:10.1007/BF01742589.

[6] G. D. Cheng and X. Guo. ε-relaxed approach in structural topology optimization. *Structural and Multidisciplinary Optimization*, 13:258–266, 1997. ISSN 1615-147X. doi:10.1007/BF01197454.

[7] J. E. Hicken. Inexact Hessian-vector products in reduced-space differential-equation constrained optimization. *Optimization and Engineering*, pages 1–34, 2014. doi:10.1007/s11081-014-9258-6. URL `http://dx.doi.org/10.1007/s11081-014-9258-6`.

[8] K. A. James, J. S. Hansen, and J. R. Martins. Structural topology optimization for multiple load cases using a dynamic aggregation technique. *Engineering Optimization*, 41(12):1103–1118, 2009. doi:10.1080/03052150902926827.

[9] R. M. Jones. *Mechanics of Composite Materials*. Technomic Publishing Co., 1996.

[10] G. J. Kennedy. Strategies for adaptive optimization with aggregation constraints using interior-point methods. *Computers & Structures*, 153(0):217 – 229, 2015. ISSN 0045-7949. doi:10.1016/j.compstruc.2015.02.024.

[11] G. J. Kennedy and J. E. Hicken. Improved constraint-aggregation methods. *Computer Methods in Applied Mechanics and Engineering*, 289(0):332 – 354, 2015. ISSN 0045-7825. doi:10.1016/j.cma.2015.02.017.

[12] 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.

[13] G. J. Kennedy and J. R. R. A. Martins. A parallel aerostructural optimization framework for aircraft design studies. *Structural and Multidisciplinary Optimization*, 50(6):1079–1101, 2014. ISSN 1615-147X. doi:10.1007/s00158-014-1108-9.

[14] G. Kreisselmeier and R. Steinhauser. Systematic control design by optimizing a vector performance index. In *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems*, Zurich, Switzerland, 1979.

[15] A. B. Lambe, G. J. Kennedy, and J. R. R. A. Martins. Multidisciplinary design optimization of an aircraft wing via a matrix-free approach. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, 2014.

[16] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli. Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, 41:605–620, 2010. ISSN 1615-147X. doi:10.1007/s00158-009-0440-y.

[17] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, 2004. doi:10.2514/1.11478.

[18] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. A coupled–adjoint sensitivity analysis method for high–fidelity aero–structural design. *Optimization and Engineering*, 6:33–62, 2005. doi:10.1023/B:OPTE.0000048536.47956.62.

[19] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994. doi:10.1137/1.9781611970791.

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

[21] J. París, F. Navarrina, I. Colominas, and M. Casteleiro. Topology optimization of continuum structures with local and global stress constraints. *Structural and Multidisciplinary Optimization*, 39(4):419–437, 2009. ISSN 1615-147X. doi:10.1007/s00158-008-0336-2.

[22] R. B. Pipes and B. W. Cole. On the off-axis strength test for anisotropic materials. *Journal of Composite Materials*, 7(4): 246–256, 1973. doi:10.1177/002199837300700208.

[23] N. Poon and J. R. R. A. Martins. An adaptive approach to constraint aggregation using adjoint sensitivity analysis. *Structural and Multidisciplinary Optimization*, 34:61–73, 2007. ISSN 1615-147X. doi:10.1007/s00158-006-0061-7.

[24] J. Qin and D. Nguyen. Generalized exponential penalty function for nonlinear programming. *Computers & Structures*, 50(4): 509 – 513, 1994. ISSN 0045-7949. doi:10.1016/0045-7949(94)90021-3.

[25] C. Raspanti, J. Bandoni, and L. Biegler. New strategies for flexibility analysis and design under uncertainty. *Computers & Chemical Engineering*, 24(910):2193 – 2209, 2000. ISSN 0098-1354. doi:10.1016/S0098-1354(00)00591-3.

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

[27] 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.

[28] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996. doi:10.1137/1038003.

[29] R. Yang and C. Chen. Stress-based topology optimization. *Structural optimization*, 12(2-3):98–105, 1996. ISSN 0934-4373. doi:10.1007/BF01196941.