Additive manufacturing technologies

- Greatly reduce or remove geometric restrictions
- Multimaterial manufacturing methods enable additional structural tailoring
- Metallic additive manufacturing for primary structures?

Polymer-based single material

Graeme Kennedy, Georgia Tech

Polymer-based multimaterial

Chris Williams, Virginia Tech
Topology optimization: The essential ideas

Problem domain → Discretization → Optimize
Penalization: Achieving a nearly discrete design

\[ E = x^P E_0 \quad \rho = x \rho_0 \quad x \in [0, 1] \]
Example: Stolpe–Svanberg truss

- Mass-constrained compliance minimization

\[
\min_x \quad \frac{1}{2} f^T K(x)^{-1} f
\]

such that \( m(x) \leq m_{\text{fixed}} \)

- No adjoint required: \( \psi = u \)

\[
\frac{df}{dx_i} = -u^T \frac{\partial K}{\partial x_i} u
\]
Discrete material optimization (DMO)

- Combined discrete material and topology optimization problem proposed by Sørensen and Lund [8]
- $\mathbf{x} \in \mathbb{R}^{N+1}$: 1 topology variable, $N$ material selection variables
- RAMP-based penalization scheme for stiffness/density interpolation:

\[
C(\mathbf{x}) = \frac{x_1}{1 + q(1 - x_1)} \underbrace{\sum_{j=1}^{N} \frac{x_{j+1}}{1 + q(1 - x_{j+1})} C_j}_{\text{material selection}}
\]

\[
\rho(\mathbf{x}) = x_1 \sum_{j=1}^{N} x_{j+1} \rho_j
\]

- One catch: Partition of unity constraint for each group of variables:

\[
\sum_{j=1}^{N} x_{j+1} = 1
\]
The need for large-scale topology optimization

- Aerospace structures are high-aspect ratio/low volume fraction
- Very large analysis and design optimization problems

Example: Common Research Model

- $n_{\text{depth}} = 32$ elements through-thickness
- $(t/c)_{\text{average}} = 0.14$ and $AR = 9$
- Number of elements $\approx 15\text{ million}$
Compliance minimization vs. stress-based design

Compliance min:

Stress-constrained:

- Compliance minimization and stress-constrained mass minimization lead to different shapes and topologies (Kennedy [3])
- Compliance minimization followed by fixed-topology stress-based shape design leads to different answers
Tailoring optimization algorithms for compliance vs. stress

- Compliance and stress-constrained problems have different mathematical properties
- *Different optimization methods required*

Key elements:

- Multigrid preconditioning: scales well with problem size
- Scalable optimization algorithm: Optimization algorithm distributed and parallel
- Scalable gradient evaluation: gradient computation cannot be a bottleneck
A geometric multigrid preconditioner

- Difficulty in topology problems: large changes in stiffness between adjacent regions: $\mathcal{O}(10^6)$ difference
- We evaluate the stiffness on the coarse mesh by computing an average stiffness over the fine mesh:

$$C_c(x) = \frac{1}{V_f} \int_{V_f} C(x) \, dV = \frac{1}{V_f} \sum_i V_i C_i(x)$$

- We combine multigrid with flexible GMRES to solve the governing equations
- Relatively robust: Solution time roughly doubles between initial (well-scaled) and final (poorly-scaled) designs
Scalable direct solve on the coarse mesh

- Key to robustness: Direct solve on the coarsest mesh
- Schur-complement based factorization scales well with number of processors, but poorly with number of degrees of freedom
  - But the size of the coarse mesh is much smaller
Fast, scalable gradient evaluation

- Gradient evaluation can be a bottleneck [6, 9]
- Special care is required for multimaterial problems with large numbers of design variables per element
  - 4 material problem with 33 filter variables: 165 variables per element
- For compliance, element-by-element gradient evaluation:

\[
\nabla_x f = - \sum_e u_e^T \frac{\partial K_e}{\partial x} u_e = - \sum_e \sum_q [B_e u_e]^T \frac{\partial C}{\partial x} [B_e u_e]
\]

- Generic implementation: Compute the inner product with \( \psi \) and \( \phi \):

\[
d \leftarrow d + \alpha \psi^T \frac{\partial C}{\partial x} \phi
\]

- Best performance for few functions of interest, many design variables per element
Parallel interior-point method for large-scale design

- Optimization cost is small for moderate-scale problems; much more significant for large-scale problems
- To achieve good parallel scalability for large-scale design problems, we implement parallel optimization methods
- Interior-point methods: More complex parallel implementation compared to the method of moving asymptotes (MMA), but better for MDO applications
- Algorithm details:
  - Limited-memory BFGS update scheme for the approximate Hessian of the Lagrangian with damped updates
  - Line-search interior-point method with standard techniques for the Fiacco–McCormick barrier update
Optimization problem structure

- Our interior-point method is based on the following problem structure:

  \[
  \begin{align*}
  \text{minimize} & \quad f(x, u(x)) \\
  \text{with respect to} & \quad x \\
  \text{such that} & \quad x_l \leq x \leq x_u \\
  & \quad c(x, u(x)) \geq 0 \\
  & \quad c_s(x) \geq 0 \\
  \text{governed by} & \quad K(x)u = f
  \end{align*}
  \]

- The constraints are divided into two sets:
  - Dense constraints: \( c(x, u(x)) \); \( A = \nabla_x c(x) \) fully populated
  - Sparse constraints \( c_s(x) \); \( A_s = \nabla_x c_s(x) \) is sparse

- Dense constraints store on all procs; sparse constraints distributed

- \( A_s \) has a special structure such the product is block diagonal for any diagonal matrix \( D \):

  \[
  A_s D A_s^T = G = \text{diag}\{G_{11}, G_{22}, \ldots, G_{qq}\}
  \]

- DMO constraints fit this structure
Implementing the approximate Newton step in parallel

- At each step in the interior-point method, we compute a search direction by solving the linearized optimality conditions
- After manipulation, this system takes the following form:

\[
\begin{bmatrix}
F & E^T \\
E & A_sD^{-1}A_s^T + G
\end{bmatrix}
\begin{bmatrix}
p_z \\
p_s
\end{bmatrix} =
\begin{bmatrix}
r_c \\
r_s
\end{bmatrix}
\]

- \(F\) is small, dense, and stored on all procs
- \(E\) is stored as a series of distributed vectors
- The lower block-diagonal \(A_sD^{-1}A_s^T + G\) can be very large, but is distributed and block-diagonal
- We can compute an update using a Schur-complement method - more details in the paper

\[
\left[F - E^T \left(A_sD^{-1}A_s^T + G\right)^{-1} E\right] p_z = r
\]

- Sparse constraint structure enables efficient parallel solution
Compliance minimization problem

- Problem discretization: $64 \times 64 \times 256$ elements
- DMO parametrization
- 5.24 million design variables, 1.05 million constraints

\[
P = 10^3 \text{ kN}
\]

\[
E_1 = 210 \text{ GPa} \quad \rho_1 = 3\rho_0
\]
\[
E_2 = 140 \text{ GPa} \quad \rho_2 = 2\rho_0
\]
\[
E_3 = 70 \text{ GPa} \quad \rho_3 = \rho_0
\]
\[
E_4 = 35 \text{ GPa} \quad \rho_4 = \frac{1}{2}\rho_0
\]

\[
L = 20
\]

\[
P = 10^3 \text{ kN}
\]

\[
L/4 = 5
\]
Optimization required 9 hours and 52 minutes of wall time on 32 processors

- 4.2 iterations a minute, or roughly an iteration every 15 seconds
Multimaterial compliance minimization

Compliance vs. Major iterations graph.
Post-processing and build

- Generated an .stl file from the 0.5-level set of the function:

\[ x_1 \max_{j=1,\ldots,N} x_{j+1} \]

- No post-smoothing applied

- Thanks to Justin Gray (NASA) for printing the structure
Stress-constrained design optimization

- Minimize the structural mass
- Bound a stress measure $\sigma(\xi) \leq \sigma_y$ everywhere within the domain, $\xi \in \Omega$

$\sigma(\xi) \leq \sigma_y$ is an infinite-dimensional constraint; it must be imposed everywhere in $\Omega$

We cheat: Impose it at lots of points
Problem formulation: Thickness parametrization

▶ Start with continuous thickness problems, topology later
▶ Direct or inverse parametrization for the structural thicknesses

Direct: \( t_i = x_i \)
Inverse: \( t_i = \frac{1}{x_i} \)

▶ Mass and stiffness are linear functions of the thickness

\[
m(x) = \mathbf{t}^T \mathbf{m} \\
K(x) = \sum_i t_i K_i
\]

▶ Use of a thickness-filter does not alter these expressions
Problem formulation: Stress constraint formulations

- We study stress constraints that take the form:
  \[ g(\xi, u) = 1 - s^T h - s^T G s \geq 0 \]

- Scaled stress, \( s(\xi) \), evaluated at a point in the structure \( \xi \in \Omega \)
- \( s(\xi) = DBu \) depends on \( u \) but not \( x \) directly

- Many failure criteria take this form, e.g.
  Von Mises:
  \[ s = \frac{1}{\sigma_s} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix}, \quad G = \begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix}, \quad h = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

- Note: \( G \) is positive definite, so \( g(\xi, u) \) is concave!
Optimization using reduced-space methods

Full-space problem:

\[
\begin{align*}
\text{minimize} & \quad m(x) \\
\text{with respect to} & \quad x \geq 0, \ u \\
\text{such that} & \quad g(\xi_i, u) \geq 0 \\
& \quad K(x)u = f
\end{align*}
\]

- If direct parametrization used, non-convexity only enters through the governing equation
Optimization using reduced-space methods

Reduced-space problem:

minimize \( m(x) \)

with respect to \( x \geq 0 \)

such that \( g(\xi_i, u(x)) \geq 0 \)

governed by \( K(x)u = f \)

- Convexity properties destroyed due to \( u(x) \)
Optimization using reduced-space methods

Aggregation-constrained reduced space problem:

\[
\text{minimize } \quad m(x) \\
\text{with respect to } \quad x \geq 0 \\
\text{such that } \quad c(g(\xi_i, u), \rho) \geq 0 \\
governed by \quad K(x)u = f
\]

Aggregation functionals:

\[
c_{\text{KS}}(g, \rho) = -\frac{1}{\rho} \ln \left[ \sum e^{-\rho g_i} \right]
\]

\[
c_{\text{IE}}(g, \rho) = \frac{\sum ge^{-\rho g}}{\sum e^{-\rho g}}
\]
Issues with constraint aggregation

- Aggregation constraints are very nonlinear
- Adaptive aggregation methods can be used, but pose issues for very large design problems

Kennedy and Hicken [6, 5, 4]
Full-space barrier methods

\[ \min \quad m(x) \]

with respect to \( x \geq 0, \ u \)

such that \( g(\xi_i, u) \geq 0 \quad K(x)u = f \)

- Impose the governing equation as an equality constraint
- Move the stress-constraints into the object through a barrier term

\[ \min_{x,u} \quad m(x) - \mu \sum_{i=1}^{n} \ln x_i - \mu \sum_{i=1}^{N} \ln g(\xi_i, u) \]

such that \( K(x)u = f \)

- Minimize for a sequence of decreasing \( \mu \)
- Issue: Problem becomes poorly scaled for small \( \mu \)
Full-space barrier methods

- The Lagrangian for the problem is:

\[
\mathcal{L}(x, u, \psi) = m(x) - \mu \sum_{i=1}^{n} \ln x_i - \mu \sum_{i=1}^{N} \ln g(\xi_i, u) + \psi^T (K(x)u - f)
\]

- The KKT conditions for the barrier problem take the form:

\[
\begin{align*}
    r_x &= \nabla_x m - \mu X^{-1} e + A(x, u)^T \psi = 0 \\
    r_u &= K(x) \psi - \mu g = 0 \\
    r_\psi &= K(x)u - f = 0
\end{align*}
\]

- Derivatives of the governing equation with respect to the design variables:

\[
A(x, u) = \frac{\partial K(x)u}{\partial x}
\]

- Independent of \( x \) for the direct parametrization
Newton method for the full-space barrier problem

- Solve the Newton system inexactly with FGMRES for the step:

\[
\begin{bmatrix}
\nabla_{xx} \mathcal{L} & A(x, \psi)^T & A(x, u)^T \\
A(x, \psi) & \mu C & K(x) \\
A(x, u) & K(x) &
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta u \\
\Delta \psi
\end{bmatrix}
=
\begin{bmatrix}
-r_x \\
r_u \\
r_\psi
\end{bmatrix}
\]

- Hessian of the Lagrangian takes different forms depending on the parametrization
  - Direct parametrization:
    \[
    \nabla_{xx} \mathcal{L} = \mu X^{-2}
    \]
  - Inverse parametrization:
    \[
    \nabla_{xx} \mathcal{L} = \nabla_{xx} m + \mu X^{-2} + \nabla_{xx}(\psi^T K(x) u)
    \]

- $X^{-2}$ and $\nabla_{xx} m$ are positive definite
- $\nabla_{xx}(\psi^T K(x) u)$ may be indefinite
Results for direct thickness parametrization

- Discretization using either $32 \times 16$, $64 \times 32$, or $128 \times 64$ elements/design variables
Mass minimization results

Newton Iterations

Mass

Newton Iterations

32x16
64x32
128x64
Design and stress
Topology optimization: Unfortunate complications

1. Penalization (e.g. RAMP, SIMP) with \( t \in (0, 1] \)
   - Penalization only affects the governing equations

2. \( \epsilon \)-relaxation to avoid stress singularity (Allows large stresses in elements with small thicknesses)
   - \( \epsilon \)-relaxation requires adding terms to the stress-constraints
     \[
     g(\xi, u, x) = 1 + \frac{\epsilon}{t} - \epsilon - s^T h - s^T G s \geq 0
     \]
   - Pick small \( \epsilon > 0 \), as \( t \to 0 \), stress constraint is always satisfied
   - Direct parametrization: non-concave
     \[
     g(\xi, u, x) = 1 + \frac{\epsilon}{x} - \epsilon - s^T h - s^T G s \geq 0
     \]
   - Inverse parametrization: concave
     \[
     g(\xi, u, x) = 1 + \epsilon x - \epsilon - s^T h - s^T G s \geq 0
     \]
Newton system for the topology optimization problem

\[
\begin{bmatrix}
\nabla_{xx} \mathcal{L} & A(x, \psi)^T + \mu C_{ux}^T & A(x, u)^T \\
A(x, \psi) + \mu C_{ux} & \mu C & K(x) \\
A(x, u) & K(x) & \\
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta u \\
\Delta \psi \\
\end{bmatrix}
= 
\begin{bmatrix}
- r_x \\
r_u \\
r_\psi \\
\end{bmatrix}
\]

- \(K(x)\) is not a linear function of \(x\) due to penalization
  - Direct parametrization:
    \[
    \nabla_{xx} \mathcal{L} = \mu X^{-2} + \mu C_{xx} + \nabla_{xx}(\psi^T K(x) u)
    \]
  - Inverse parametrization:
    \[
    \nabla_{xx} \mathcal{L} = \nabla_{xx} m + \mu X^{-2} + \mu C_{xx} + \nabla_{xx}(\psi^T K(x) u)
    \]

- Direct parametrization: \(C_{xx}\) is indefinite since the constraint contains \(1/x\) term
- Inverse parametrization: \(C_{xx}\) is positive definite
  - Use inverse parametrization and discard \(\nabla_{xx}(\psi^T K(x) u)\)
Design and stress
Topology optimization results
Topology optimization results

Mass

Newton Iterations
Conclusions

1. Large-scale topology and multimaterial problems require efficient parallel analysis, parallel gradient evaluation, and parallel optimization methods

2. Full-space barrier methods enable tailoring for specific structural optimization problems
   - Control of positive definite/indefinite terms

3. Recommendations for stress-constrained design:
   - Use direct thickness variables when performing sizing
   - Use inverse variables for topology optimization


