Numerical optimization techniques for structural optimization

Four classes of techniques…

ME 260 at the Indian Institute of Science, Bengaluru

Structural Optimization: Size, Shape, and Topology

G. K. Ananthasuresh

Professor, Mechanical Engineering, Indian Institute of Science, Bengaluru

suresh@iisc.ac.in

If you have an analysis program, you are only a small step away from writing a topology optimization program.

What to solve and what to solve for in topology optimization problems?

What to solve: Partial Differential Equations (PDEs)

- Governing equation(s)
- Adjoint equation(s)
- Design equation(s)

What to solve for:

- State variables
- Adjoint variables
- Lagrange multipliers
- Design variables

What to ensure:

- Constraint(s) should be satisfied.

Four methods for solving topology optimization problems

Optimality Criteria (OC) method

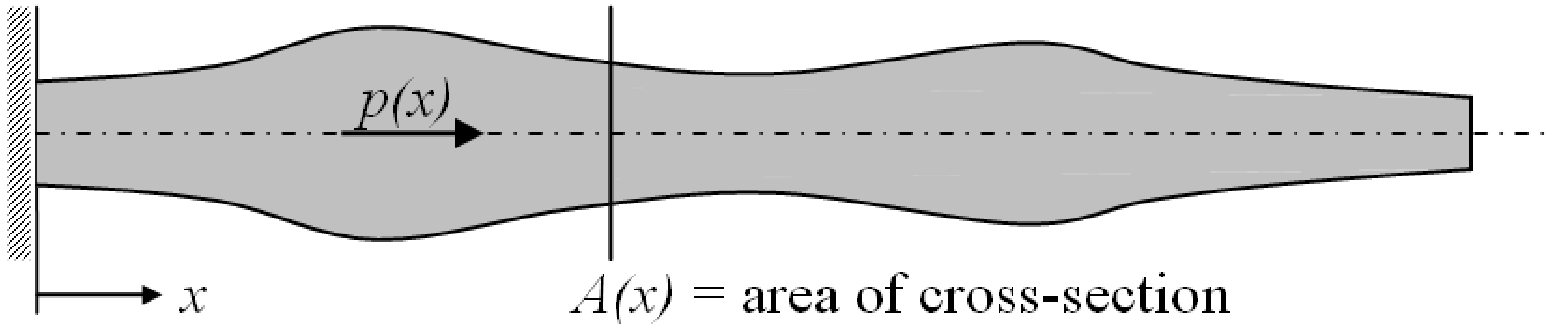
Mathematical Programming (MP) methods
- Black-box approach (mostly)

Convex Approximation methods
- ConLin
- MMA

Topology-derivative-based level-set method

Optimality Criteria (OC) method

Simplest structural optimization problem



Maximize the stiffness for given volume of material of a bar.

General structure of a structural optimization problem

A simple problem

$$\underset{A(x)}{Min}\ MC = \int_0^L p\,u\,dx$$

Subject to

$$\lambda(x): \qquad (EAu')' + p = 0$$

$$\Lambda: \qquad \int_0^L A\,dx - V^* \le 0$$

$$\mu_u(x): \qquad A - A_u \le 0$$

$$\mu_l(x): \qquad A_l - A \le 0$$

$$Data: L, p(x), E, V^*, A_l, A_u$$

Optimize w.r.t. design variables — Objective function

(It depends on design variables and state variables.)

Subject to — Governing equation(s)

(They govern state variables.)

They create conflict in optimizing the objective function. — Resource constraint(s), Performance constraints

Data *(This should be properly chosen although the nature of the solution is not decided by the data.)*

Steps in the solution procedure

Step 1: Write the Lagrangian.

Step 2: Take variation of the Lagrangian w.r.t. the design variable and equate to zero to get the design equation.

Step 3: Take variation of the Lagrangian w.r.t. state variable(s) and equate to zero to get the adjoint equation.

Step 4: Collect all the equations, including the governing equation(s), complementarity condition(s), resource constraints, etc.

Step 5: Obtain the optimality criterion by substituting adjoint and equilibrium equations into the design equation, when it is possible.

Step 6: Identify all boundary conditions.

Step 7: Solve the equations analytically as much as possible.

Step 8: Use the optimality criteria method to solve the equations numerically.

Step 1

Step 1: Write the Lagrangian

$$\underset{A(x)}{Min}\ MC = \int_0^L p\,u\,dx$$

Subject to

$$\lambda(x): \qquad (EAu')' + p = 0$$

$$\Lambda: \qquad \int_0^L A\,dx - V^* \le 0$$

$$\mu_u(x): \qquad A - A_u \le 0$$

$$\mu_l(x): \qquad A_l - A \le 0$$

$$Data: L, p(x), E, V^*, A_l, A_u$$

Lagrangian

$$L = \int_0^L p\,u\,dx + \int_0^L \lambda\left\{(EAu')' + p\right\}dx + \Lambda\left(\int_0^L A\,dx - V^*\right)$$

$$+\int_0^L \mu_u\left(A - A_u\right)dx + \int_0^L \mu_l\left(A_l - A\right)dx$$

Integrand in the Lagrangian functional

$$\hat{L} = p\,u + \lambda\left\{(EAu')' + p\right\} + \Lambda A + \mu_u\left(A - A_u\right) + \mu_l\left(A_l - A\right)$$

Step 2

Step 2: Take variation of the Lagrangian w.r.t. the design variable and equate to zero to get the design equation.

$$\hat{L} = pu + \lambda\left\{\left(EAu'\right)' + p\right\} + \Lambda A + \mu_u\left(A - A_u\right) + \mu_l\left(A_l - A\right)$$

$$= pu + E\lambda A'u' + E\lambda Au'' + \Lambda A + \mu_u\left(A - A_u\right) + \mu_l\left(A_l - A\right)$$

$$\delta_A L = 0 \Rightarrow \frac{\partial \hat{L}}{\partial A} - \left(\frac{\partial \hat{L}}{\partial A'}\right)' = 0$$

$$\Rightarrow E\lambda u'' + \Lambda + \mu_u - \mu_l - \left(E\lambda u'\right)' = 0$$

$$\Rightarrow \Lambda + \mu_u - \mu_l - E\lambda'u' = 0$$

Design equation

$$\{\Lambda\} + \{\mu_u\} - \{\mu_l\} + \{-E\lambda'u'\} = 0$$

What multiplies the Lagrange multiplier is the sensitivity of the corresponding constraint. Here, it is positive or negative unity for all three.

Sensitivity (derivative) of the objective function

Step 3

Step 3: Take variation of the Lagrangian w.r.t. state variable(s) and equate to zero to get the adjoint equation.

$$\hat{L} = pu + E\lambda A'u' + E\lambda Au'' + \Lambda A + \mu_u\left(A - A_u\right) + \mu_l\left(A_l - A\right)$$

$$\delta_u L = 0 \Rightarrow \frac{\partial \hat{L}}{\partial u} - \left(\frac{\partial \hat{L}}{\partial u'}\right)' + \left(\frac{\partial \hat{L}}{\partial u''}\right) = 0$$

Adjoint (equilibrium) equation

$$p + \left(EA\lambda'\right)' = 0$$

$$\Rightarrow p - \left(E\lambda A'\right)' + \left(E\lambda A\right)'' = 0$$

$$\Rightarrow p + \left(EA\lambda'\right)' = 0$$

Adjoint load; here it is equal to the actual load because the objective function is mean compliance.

No change here as compared to Problem 1.

Step 4

Step 4: Collect all the equations, including the governing equation(s), complementarity condition(s), resource constraints, etc.

$$\Lambda + \mu_u - \mu_l - E\lambda' u' = 0$$

Design equation

$$p + \left(EA\lambda'\right)' = 0$$

Adjoint (equilibrium) equation

$$p + \left(EAu'\right)' = 0$$

Equilibrium (governing) equation

Complementarity conditions

$$\Lambda\left(\int_0^L A\,dx - V^*\right) = 0, \Lambda \geq 0; \quad \mu_u\left(A - A_u\right) = 0, \mu_u \geq 0; \quad \mu_l\left(A_l - A\right) = 0, \mu_l \geq 0;$$

Feasibility conditions

$$\int_0^L A\,dx - V^* \leq 0; \; A - A_u \leq 0; \; \left(A_l - A\right) \leq 0$$

Step 5

Step 5: Obtain the optimality criterion by substituting adjoint and equilibrium equations into the design equation, when it is possible.

$$\Lambda + \mu_u - \mu_l - E\lambda' u' = 0$$

$$\left.\begin{aligned} p + \left(EA\lambda'\right)' &= 0 \\ p + \left(EAu'\right)' &= 0 \end{aligned}\right\} \lambda = u$$

Optimality criterion

$$\Lambda + \mu_u - \mu_l - Eu'^2 = 0$$

These are functions of x; so, the strain energy density is not necessarily constant throughout.

Step 6

Step 6: Identify all boundary conditions.

$$\hat{L} = pu + E\lambda A'u' + E\lambda Au'' + \Lambda A + \mu_u\left(A - A_u\right) + \mu_l\left(A_l - A\right)$$

$$\left(\frac{\partial \hat{L}}{\partial A'}\right)\delta A\Bigg|_0^L = 0 \Rightarrow \left(E\lambda u' + \mu_u - \mu_l\right)\delta A\Big|_0^L = 0 \quad \text{BC1}$$

$$\left\{\frac{\partial \hat{L}}{\partial u'} - \left(\frac{\partial \hat{L}}{\partial u''}\right)'\right\}\delta u\Bigg|_0^L = 0 \Rightarrow \left\{\left(E\lambda A'\right) - \left(E\lambda A\right)'\right\}\delta u\Big|_0^L = 0 \Rightarrow \left(EA\lambda'\right)\delta u\Big|_0^L = 0 \quad \text{BC2}$$

$$\left(\frac{\partial \hat{L}}{\partial u''}\right)\delta u'\Bigg|_0^L = 0 \Rightarrow \left(EA\lambda\right)\delta u'\Big|_0^L = 0 \quad \text{BC3}$$

Step 7

Step 7: Solve the equations analytically as much as possible.

Optimality criterion

$$\Lambda + \mu_u - \mu_l - Eu'^2 = 0$$

We have three cases now.

Case 1: $\mu_u > 0, \mu_l = 0 \Rightarrow A = A_u$ Area of cross-section is thus known.

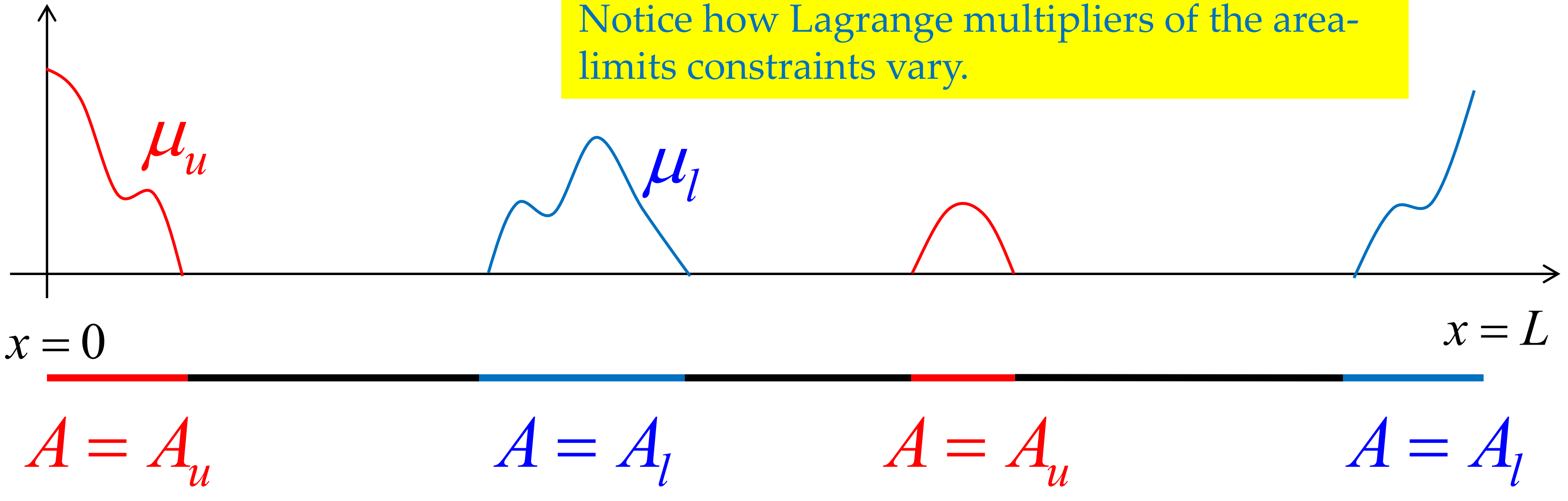
Case 2: $\mu_l > 0, \mu_u = 0 \Rightarrow A = A_l$ Area of cross-section is thus known.

Case 3: $\mu_l = 0, \mu_u = 0 \Rightarrow$ We have dealt with this in Problem 1.

We need to partition the domain (0,L) into these three cases. We do this numerically using the optimality criteria method.

Partitioning the domain into three types

Notice how Lagrange multipliers of the area-limits constraints vary.



$$A_{k+1}(x) = \left\{ \left(\frac{Eu'^2}{\Lambda} \right)^{\beta} \right\}_k A_k(x)$$

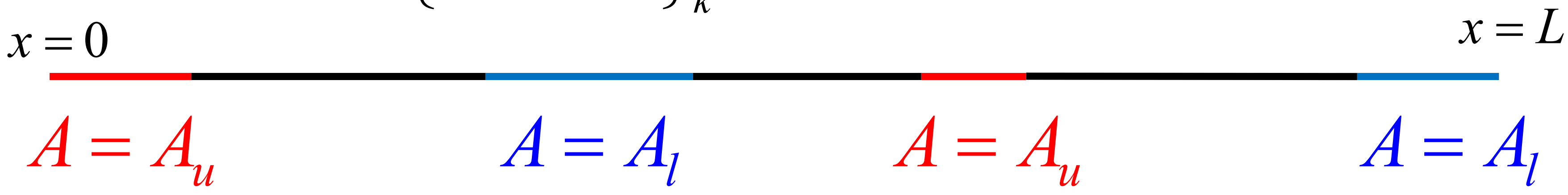
This is how we update when $A(x)$ is in between the bounds.

As per the optimality criterion

While we partition the domain, we should ensure that the volume constraint is satisfied.

Calculate the Lagrange multiplier

Substitute $A_{k+1}(x) = \left\{ \left(\dfrac{Eu'^2}{\Lambda} \right)^{\beta} \right\}_k A_k(x)$ into $\displaystyle\int_0^L A\,dx - V^* = 0$



$$\int_{\Omega_u} A_u\,dx + \int_{\Omega} A\,dx + \int_{\Omega_l} A_l\,dx = V^* \Rightarrow \int_{\Omega_u} A_u\,dx + \int_{\Omega} \left\{ \left(\frac{Eu'^2}{\Lambda} \right)^{\beta} \right\}_k A_k(x)dx + \int_{\Omega_l} A_l\,dx = V^*$$

$$\Rightarrow \int_{\Omega} \left\{ \left(\frac{Eu'^2}{\Lambda} \right)^{\beta} \right\}_k A_k(x)dx = V^* - \int_{\Omega_u} A_u\,dx - \int_{\Omega_l} A_l\,dx \Rightarrow \Lambda = \frac{\int_{\Omega} \left\{ \left(Eu'^2 \right)^{\beta} \right\}_k A_k(x)dx}{V^* - \int_{\Omega_u} A_u\,dx - \int_{\Omega_l} A_l\,dx}$$

Step 8

Step 8: Use the **optimality criteria** method to solve the equations numerically.

This is the step-wise procedure of the optimality criteria method.

(i) Choose $A_0(x)$ as the initial guess.

(ii) $$A_{k+1}(x) = \left\{ \left(\frac{Eu'^2}{\Lambda} \right)^{\beta} \right\}_k A_k(x)$$

(iii) Evaluate Λ using $\int_0^L A\,dx - V^* = 0$

(iv) Ensure that everywhere $A \leq A_u;\quad A_l \leq A$

This to is iterative; we call it the inner iterative loop. This is where we partition the domain into three types.

(v) Update $A_{k+1}(x)$ until convergence, i.e., until $A_{k+1}(x) - A_k(x) \leq$ tolerance.

Outer and inner loop iterations

Begin the outer loop with $A_0(x)$

Compute $$\Lambda = \frac{\int_\Omega \left\{ \left(E u'^2 \right)^\beta \right\}_k A_k(x)\,dx}{V^*}$$ $A_k(x)$

Update $$A_{k+1}(x) = \left\{ \left(\frac{E u'^2}{\Lambda} \right)^\beta \right\}_k A_k(x)$$

Inner loop begins here.

Check against bounds: $If \;\; A_{k+1}(x) > A_u, \;\; A_{k+1}(x) = A_u$ Add that x to Ω_u

$If \;\; A_{k+1}(x) < A_l, \;\; A_{k+1}(x) = A_l$ Add that x to Ω_l

Now, recalculate Λ As show in the next slide.

Inner loop (contd.)

$$\Lambda = \frac{\int_{\Omega} \left\{ \left(E u'^2 \right)^{\beta} \right\}_k A_k(x)\, dx}{V^* - \int_{\Omega_u} A_u\, dx - \int_{\Omega_l} A_l\, dx}$$

Update again $$A_{k+1}(x) = \left\{ \left(\frac{E u'^2}{\Lambda} \right)^{\beta} \right\}_k A_k(x)$$

Check against bounds again

$$\mathit{If} \quad A_{k+1}(x) > A_u, \quad A_{k+1}(x) = A_u \qquad \text{update } \Omega_u$$

$$\mathit{If} \quad A_{k+1}(x) < A_l, \quad A_{k+1}(x) = A_l \qquad \text{update } \Omega_l$$

Now, recalculate Λ

Repeat this until partitioning does not change.

Optimality criteria method

Updating area of cross-section is the outer loop using the optimality criterion.

Updating Lambda and partitioning of the domain is the inner loop.
Note that area of cross-section gets update in the inner loop also.
This ends when partitioning does not change anymore.

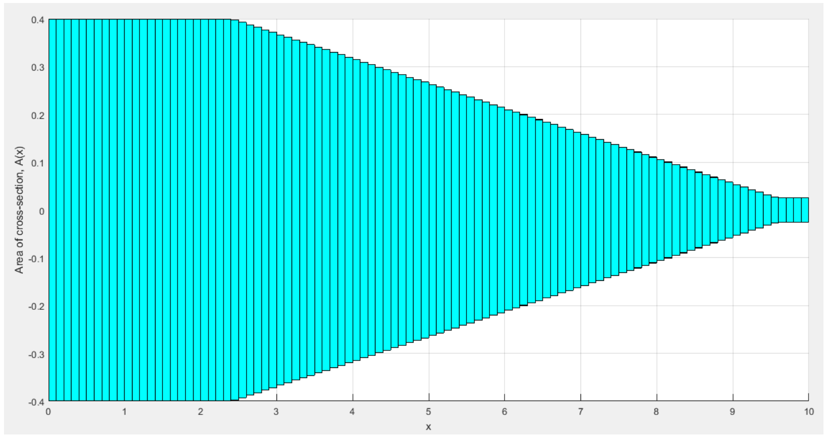
The outer loop ends when the design variable over the entire domain does not change anymore.

Check out the BarOpt Matlab code.

BarOpt has four files:
baropt.m >> implements optimality criteria method.
fembar.m >> finite element code for bar elements.
matcut.m and veccut.m >> These are used by fembar.m

baropt.m may be modified by you to change the data such as the length of the bar, loading, displacement boundary conditions, the number of elements in the bar, total number of iterations, the tolerance to stop the iterative process, etc.

A representative result of optimized area of cross-section.



What to solve and what to solve for in topology optimization problems?

What to solve: Partial Differential Equations (PDEs)

- Governing equation(s)
- Adjoint equation(s)
- Design equation(s)

What to solve for:

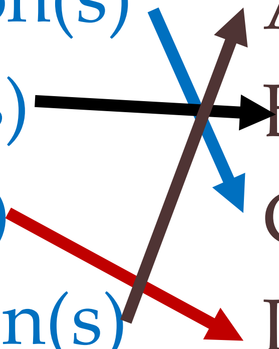
- State variables
- Adjoint variables
- Lagrange multipliers
- Design variables

What to ensure:

- Constraint(s) should be satisfied.

Now, match the following!

1. Governing equation(s)
2. Adjoint equation(s)
3. Design equation(s)
4. Constraint equation(s)



A. Lagrange multiplier(s)
B. Adjoint variable(s)
C. State variable(s)
D. Design variable(s)

Answer (correct matching)

1. Governing equation(s)
2. Adjoint equation(s)
3. Design equation(s)
4. Constraint equation(s)

A. State variable(s)
B. Adjoint variable(s)
C. Design variable(s)
D. Lagrange multiplier(s)

How do you solve?

1.	Governing equation(s)	A.	FEA, CFD, etc.
2.	Adjoint equation(s)	B.	FEA, CFD, etc.
3.	Design equation(s)	C.	Outer loop
4.	Constraint equation(s)	D.	Inner loop

Mathematical Programming (MP) methods

Mathematical programming algorithms

Unconstrained minimization

- Zeroth order methods
- First order methods
- Second order methods

Constrained minimization

- Sequential unconstrained methods
- Sequential approximation methods

Unconstrained minimization

- Zeroth order methods
- First order methods
- Second order methods

Zeroth order methods

- Univariate method
- Pattern search methods
- Simplex method
- Nelder-Mead method

Pattern search methods

- Hooke and Jeeves method
- Powell's conjugate method
- Rosenbrock's method

Unconstrained minimization

- Zeroth order methods
- First order methods
- Second order methods

First order methods

- Steepest descent method
- Fletcher-Reeves method

Unconstrained minimization

- Zeroth order methods
- First order methods
- Second order methods

Second order methods

- Newton method
- Variable metric methods

Variable metric methods

- DFP (Davidon-Fletcher-Powell) method
- BFGS (Broydon-Fletcher-Goldfarb-Shanno) method

Constrained minimization

- Sequential unconstrained methods
- Sequential approximation methods

Sequential unconstrained minimization methods

- EPF (Exterior Penalty Function) method
- IPF (Interior Penalty Function) method
- Extended IPF (EIPF)
- Augmented Lagrangian method

Constrained minimization

- Sequential unconstrained methods
- Sequential approximation methods

Sequential approximation methods

- SLP (Sequential Linear Programming)
- SQP (Sequential Quadratic Programming)
- Method of feasible directions
- Generalized Reduced Gradient (GRG) method
- Trust region method
- Convex approximation methods

Convex linearization (CONLIN)
Method of Moving Asymptotes (MMA)
Generalized Convex Approximation (GCA)

The end note

Numerical optimization methods

- General statement of the problem revisited
- Four classes of techniques
- Optimality criteria method
- Mathematical programming methods
- Convex linearization and level-set methods

Thanks