Shape Optimization Using Density—based Topology Optimization

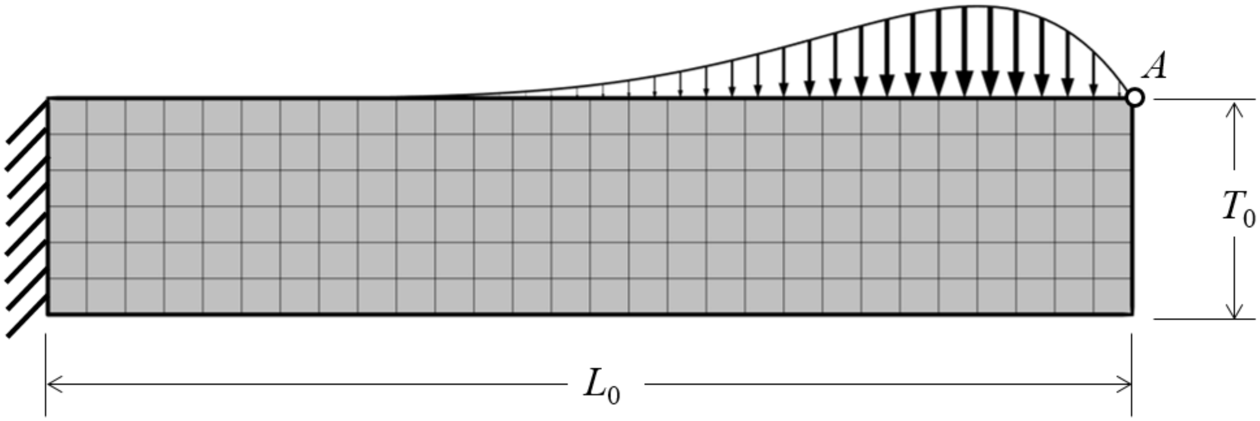
Tomas zegard[‡,1] ; Diego Salinas[1] ; Emilio Silva[2]

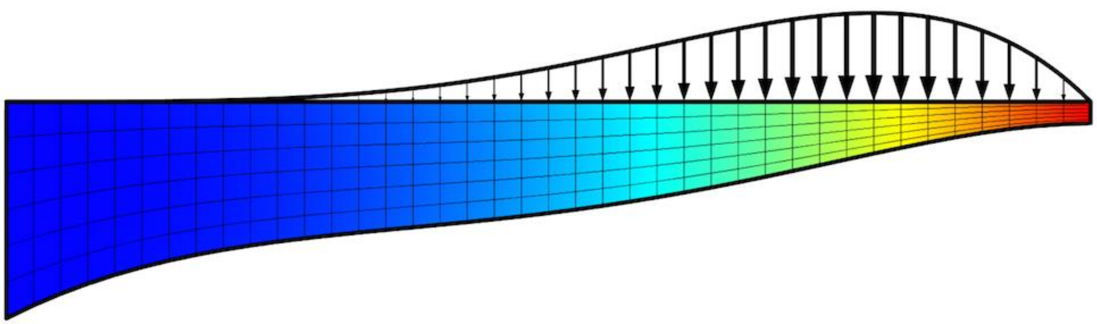
‡ — Presenting author
1 — PUC (Chile)
2 — USP (Brazil)

Universidade de São Paulo

USNCCM 15
Austin, TX

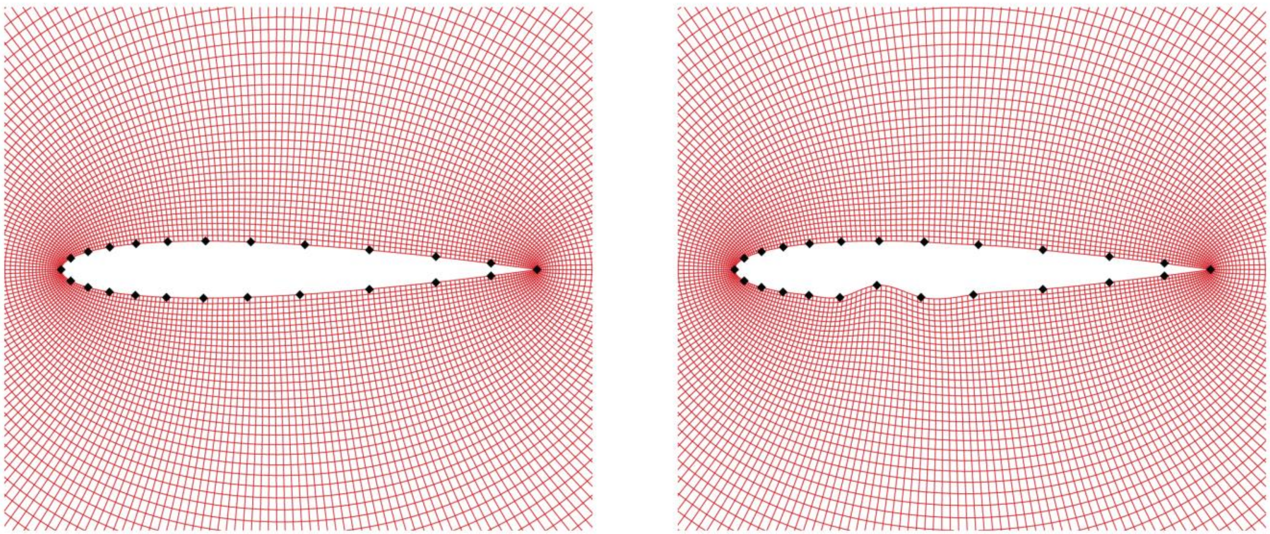
Motivation





COMSOL — optimization module
https://www.comsol.com/blogs/designing-new-structures-with-shape-optimization/

motivation



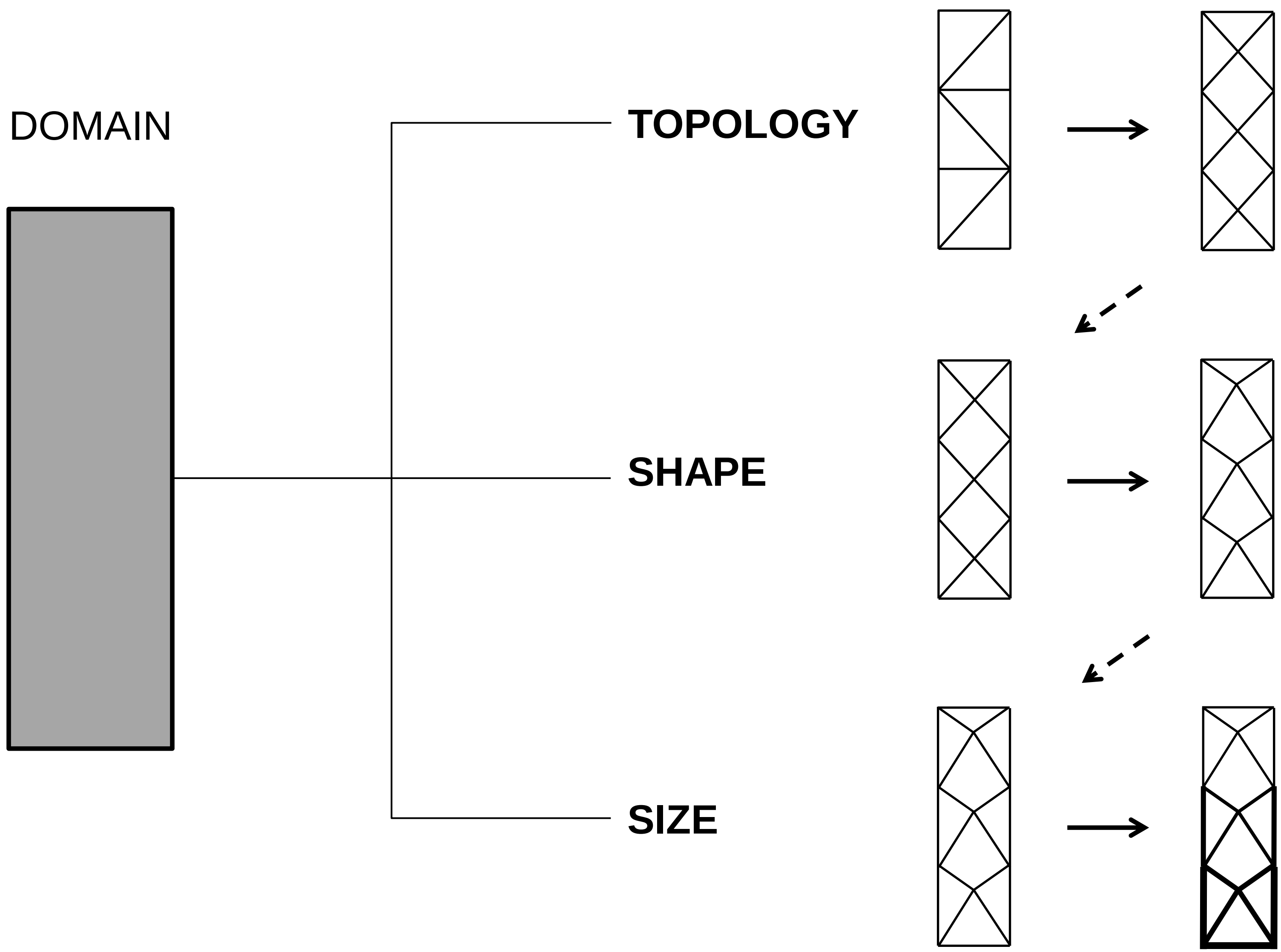
(a) Control points on surface

(b) Deformation of control points on surface

Figure 1: Example of control points on surface, showing how deformations affect volume and surface

Poole, Allen, & Rendall "Control Point-Based Aerodynamic Shape Optimization Applied to AIAA ADODG Test Cases", AIAA proceedings 53rd aerospace meeting, florida (2015)

Motivation

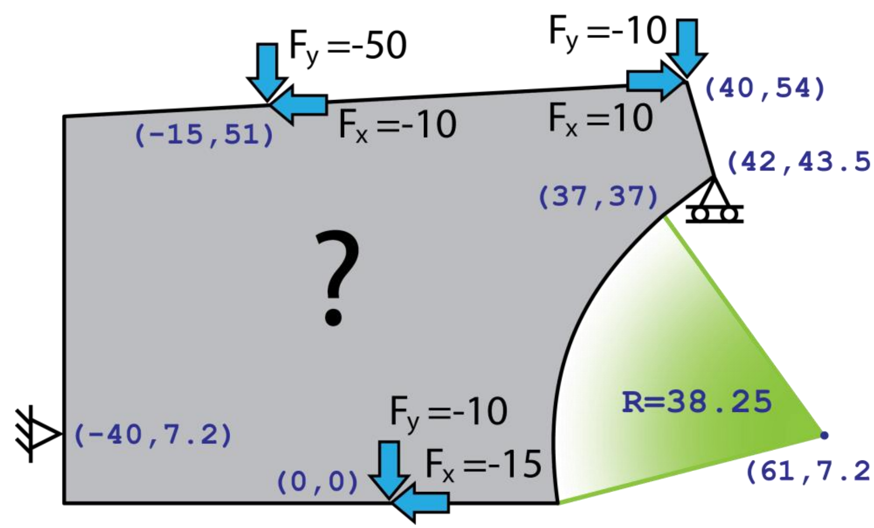


motivation

- density-based topology optimization

http://www.cannondale.com

Cannondale capo
(Urban commuter bikE)

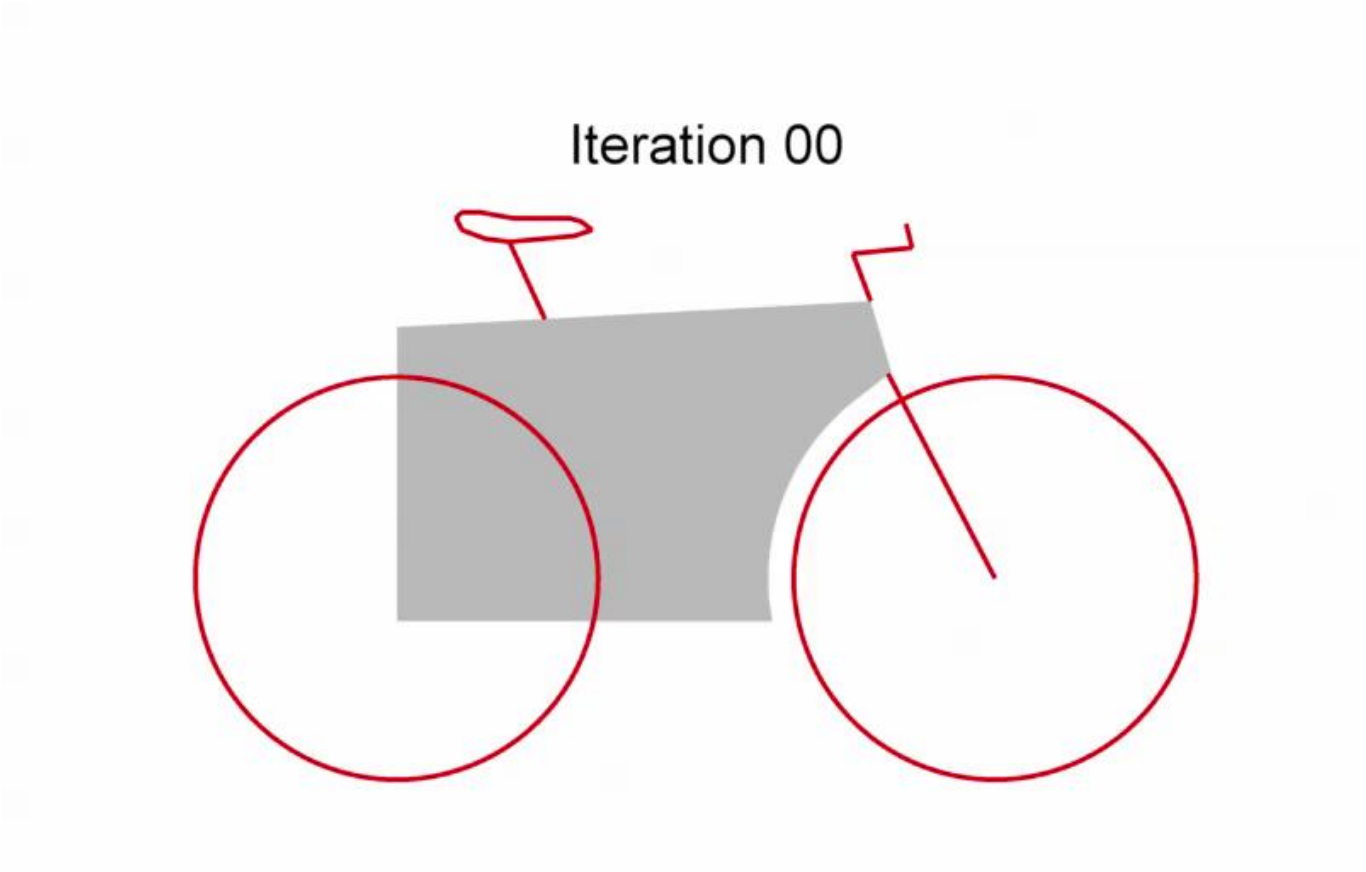


Bike Domain and loads

Zegard T, Paulino GH (2013) “Toward GPU accelerated topology optimization on unstructured meshes.” Structural and Multidisciplinary Optimization, 48(3):473-485

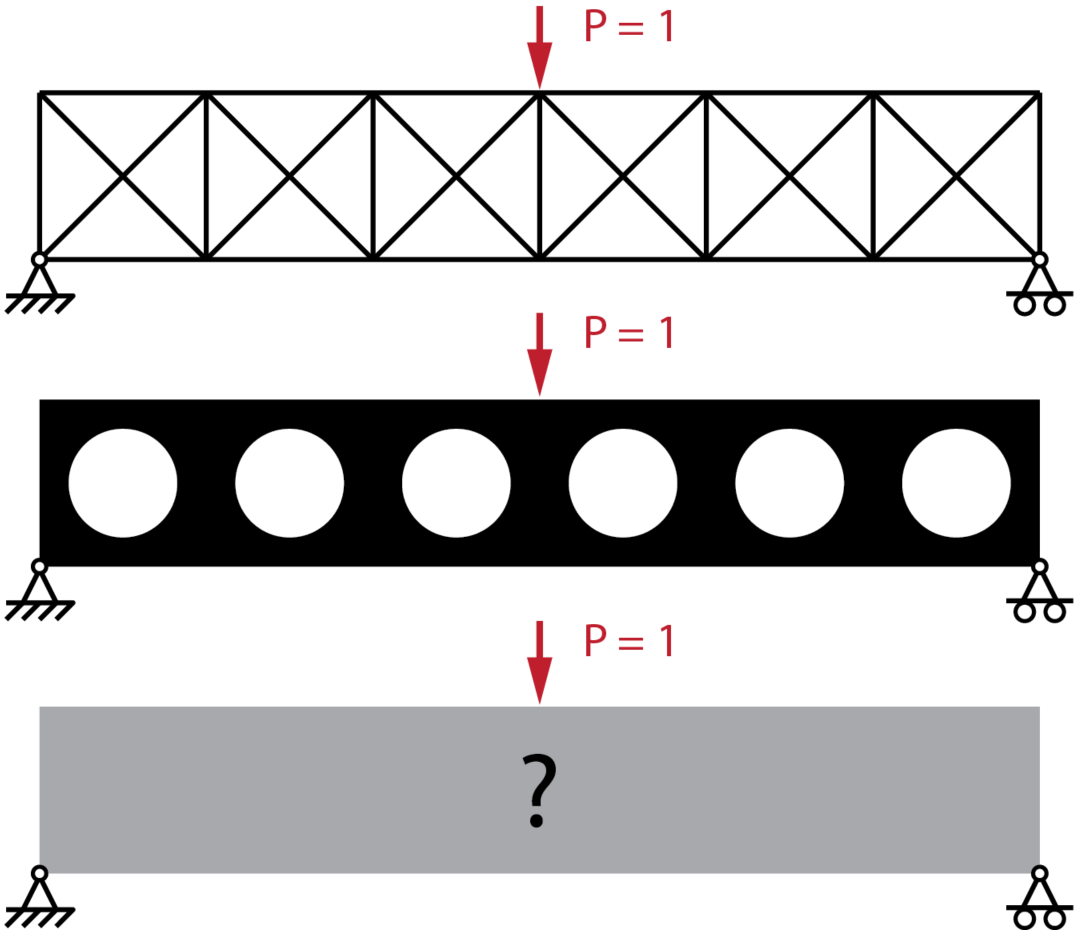
motivation

- density-based topology optimization



Zegard T, Paulino GH (2013) “Toward GPU accelerated topology optimization on unstructured meshes.” Structural and Multidisciplinary Optimization, 48(3):473-485

motivation



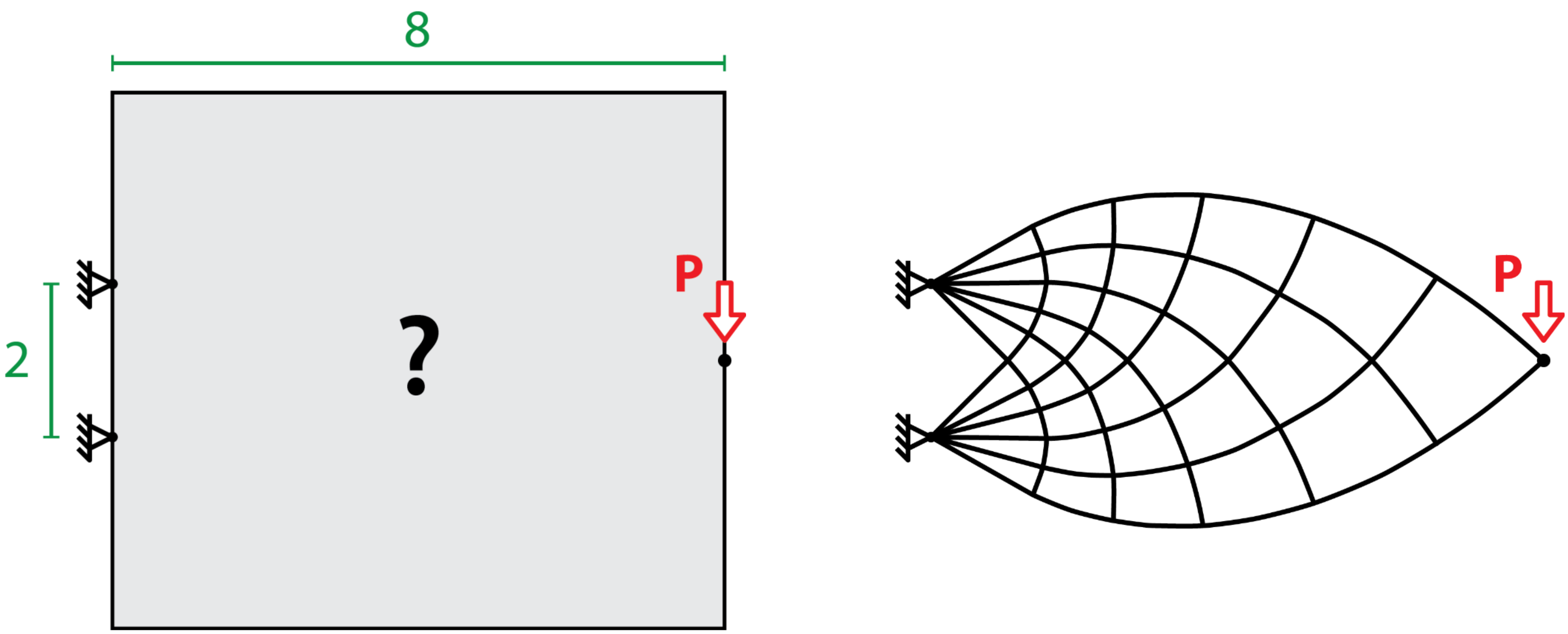
Sizing optimization

shape optimization

topology optimization

motivation

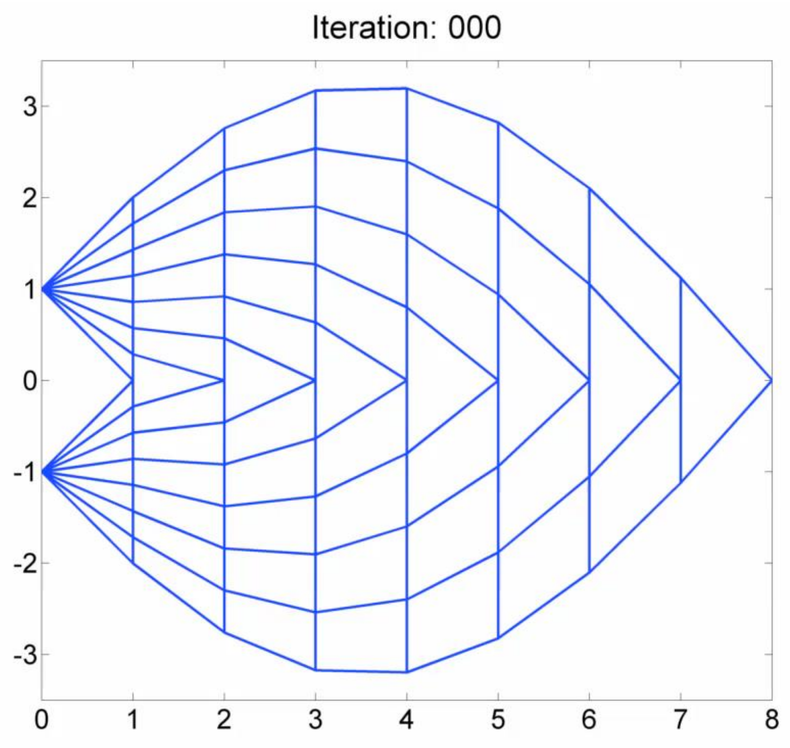
- Geometrical optimization is non-linear

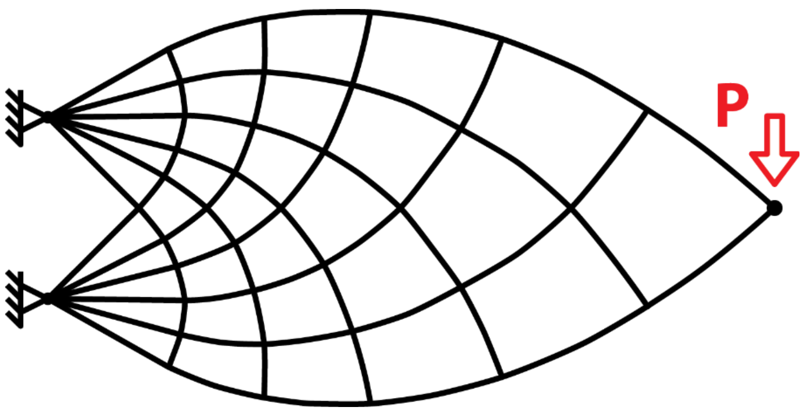


Mazurek a (2012) – “Geometrical aspects of optimum truss like structures for three-force problem”, struct and multidisciplinary opt 45(1), pp 21-32

motivation

- Geometrical optimization is non-linear

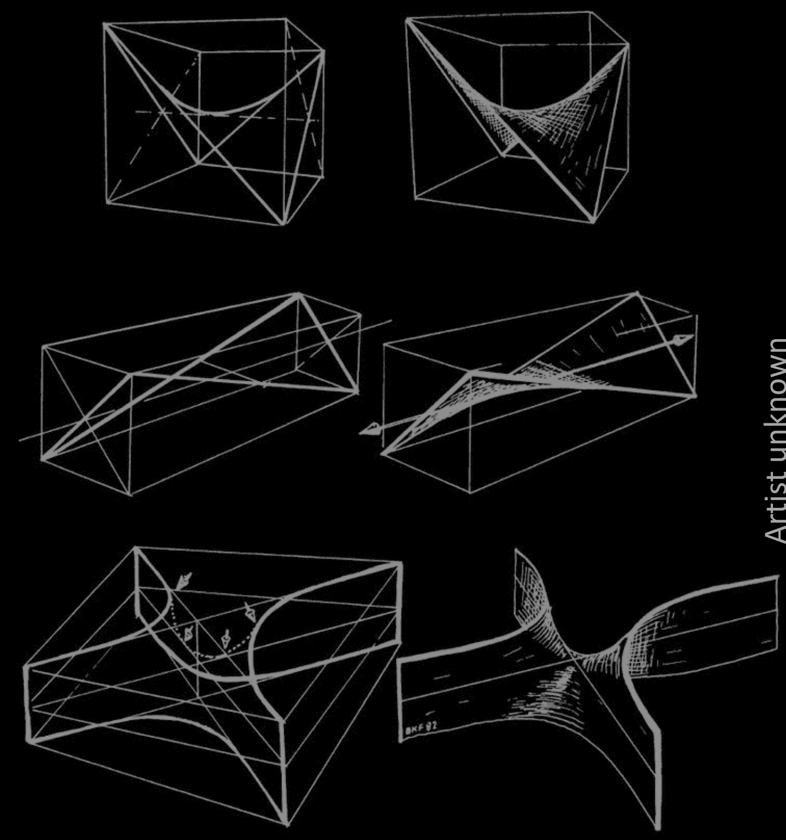




Shape optimization

=?

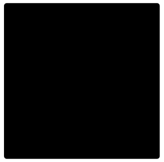
Topology-constrained topology optimization



Artist unknown

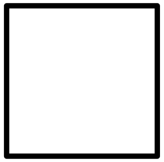
Topology constraints

- Wishlist
  - No creation/destruction of holes
  - No creation/destruction of members
  - Volume is constant
  - Eulerian method: no re-meshing!



SOLID

THIS IS A VERY DIFFICULT PROBLEM



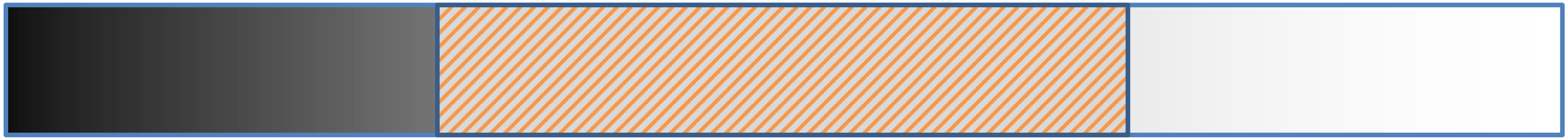
VOID

Topology constraints

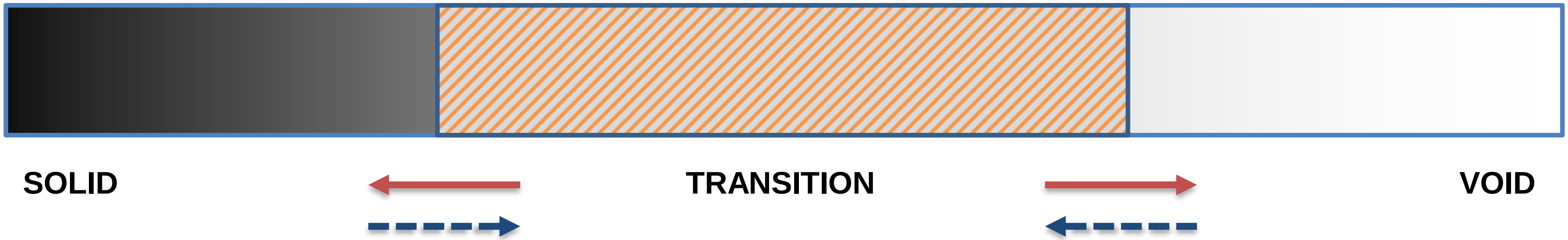
- Wishlist
  - No creation/destruction of holes
  - No creation/destruction of members
  - Volume is constant
  - Eulerian method: no re-meshing!

Topology constraints

- Wishlist
  - No creation/destruction of holes
  - No creation/destruction of members
  - Volume is constant
  - Eulerian method: no re-meshing!



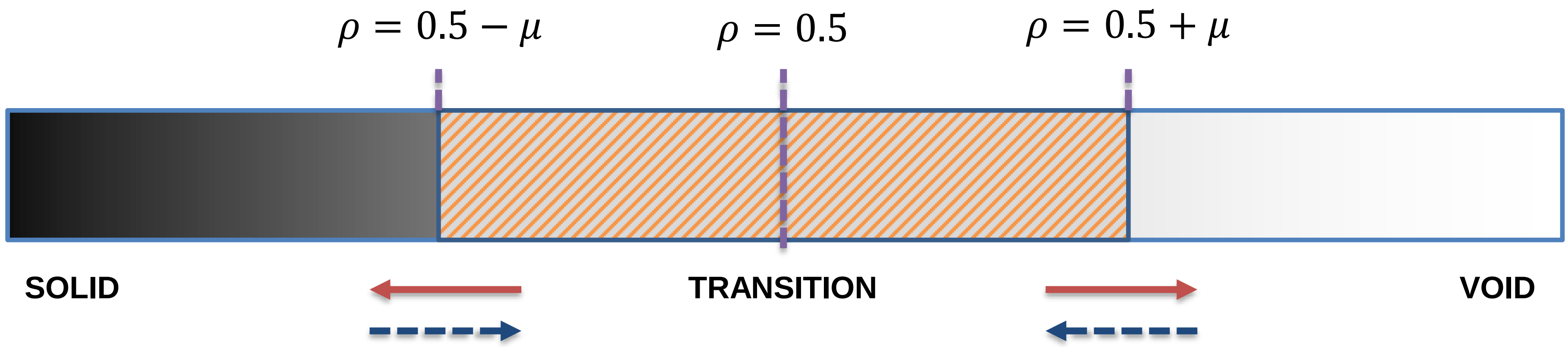
Topology constraints



- Transition elements
  - Can move to solid or void

- solid elements
  - if in boundary: can move to transition
- void elements
  - If in boundary: can move to transition

Topology constraints



- Transition elements
  - Can move to solid or void

- solid elements
  - if in boundary: can move to transition
- void elements
  - If in boundary: can move to transition

Topology constraints

- Tracking the boundary using phase
  - Find all three phases

$$\{\rho_i \in \text{Solid} \mid \rho_i \geq 0.5 + \mu\}$$
$$\{\rho_i \in \text{Void} \mid \rho_i \leq 0.5 - \mu\}$$
$$\{\rho_i \in \text{Trans} \mid 0.5 - \mu < \rho_i < 0.5 + \mu\}$$

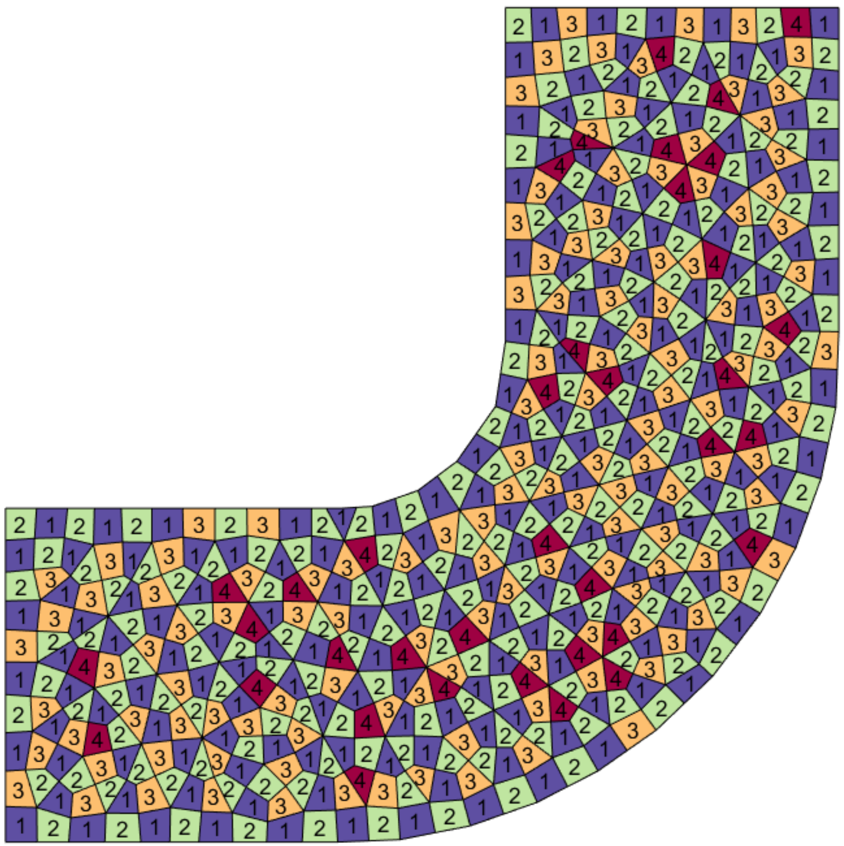
  - Boundary is defined as:
    - Having 1+ neighbor in solid

      AND
    - Having 1+ neighbor in VOID

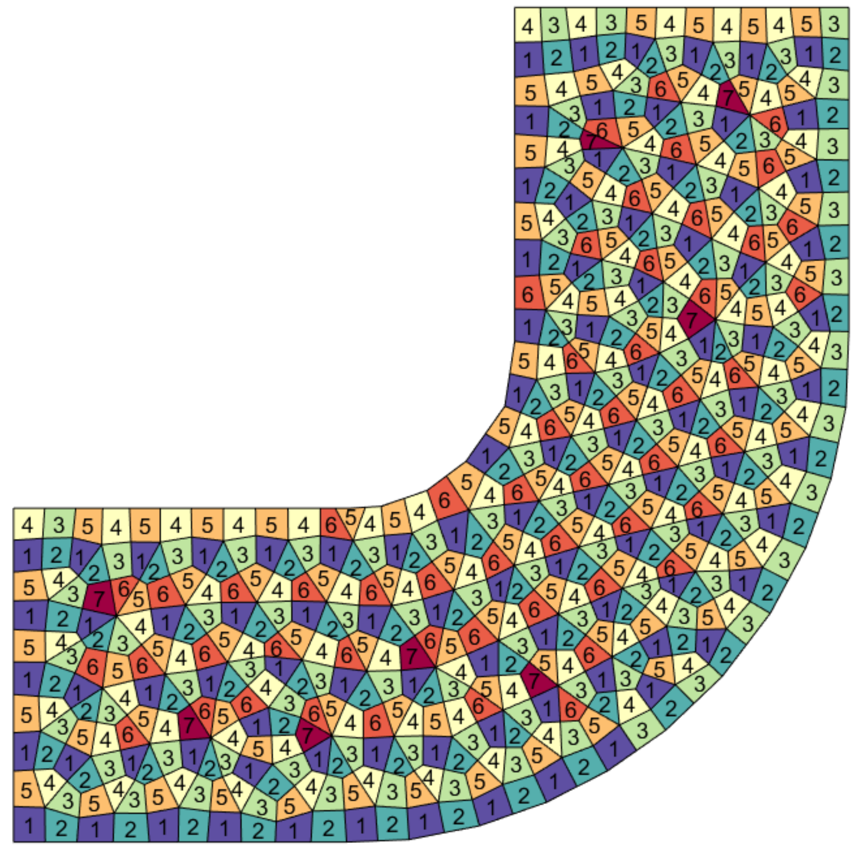
  - What is a neighbor?

Topology constraints

- Connectivity (neighbors)



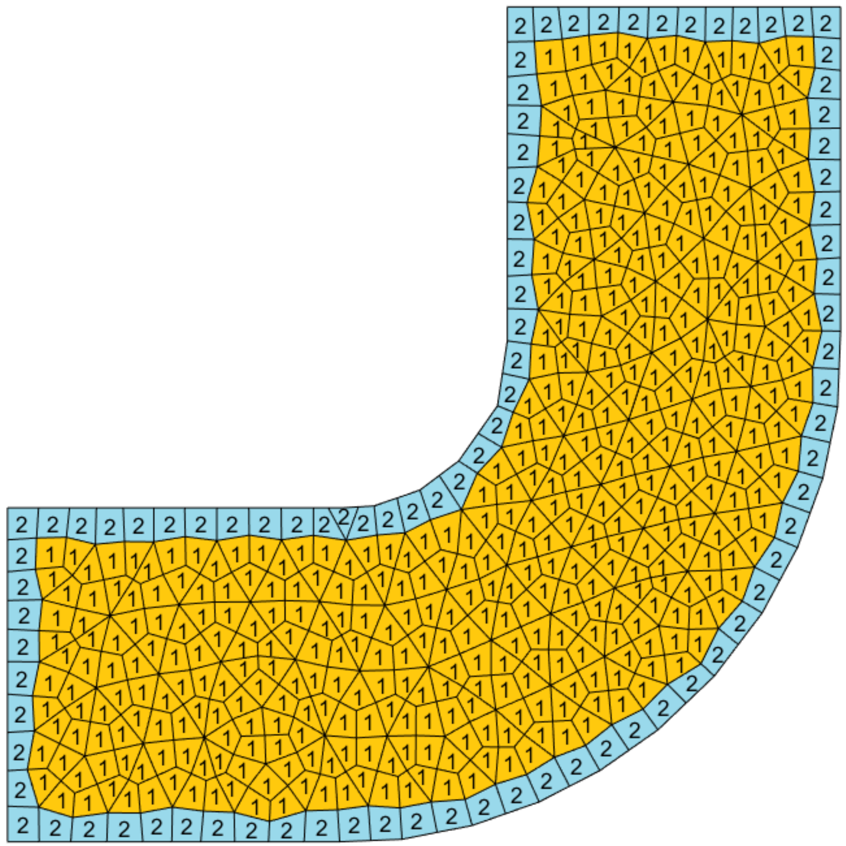
Edge-based



node-based

Topology constraints

- Domain boundary



Element who doesn't share an edge

Putting the pieces together

Vitruvius Teutsch (1548)

Proposed method: ingredients

- Neighboring

$$L_{ij} = \begin{cases} 1 & \text{if} \quad \text{element } i \text{ and } j \text{ are neighbors} \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$

- Boundary

$$\text{FlagBorder}_i = \begin{cases} 1 & \text{if} \quad \text{element } i \text{ belongs to domain boundary} \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$

- Material phase

$$\text{FlagSolid}_i = \begin{cases} 1 & \text{if} \quad \rho_i \geq 0.5 + \mu \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$

$$\text{FlagVoid}_i = \begin{cases} 1 & \text{if} \quad \rho_i \leq 0.5 - \mu \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$

$$\text{FlagTrans}_i = \begin{cases} 1 & \text{if} \quad !\,(\text{FlagSolid}_i \mid \text{FlagVoid}_i) \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$

$$\text{FlagBound} = (\mathbf{L} \cdot \text{FlagSolid})\ \&\ (\mathbf{L} \cdot \text{FlagVoid})$$

Proposed method: O.C. Update

- Standard O.C.

$$x_i^{\text{new}} = \begin{cases} \text{UB} & \text{if} \quad x_i^{\text{old}} \cdot B_i^{\eta} > \text{UB} \\ \text{LB} & \text{if} \quad x_i^{\text{old}} \cdot B_i^{\eta} < \text{LB} \\ x_i^{\text{old}} \cdot B_i^{\eta} & \text{if} \quad \text{otherwise} \end{cases}$$

$$\text{UB} = \min \begin{cases} x_i^{\text{old}} + \text{move} \\ 1 \end{cases}$$

$$\text{LB} = \min \begin{cases} x_i^{\text{old}} - \text{move} \\ 0 \end{cases}$$

Proposed method: O.C. Update

- Shape optimization — O.C.

$$x_i^{\text{new}} = \begin{cases} \text{UB} & \text{if} \quad x_i^{\text{old}} \cdot B_i^{\eta} > \text{UB} \\ \text{LB} & \text{if} \quad x_i^{\text{old}} \cdot B_i^{\eta} < \text{LB} \\ x_i^{\text{old}} \cdot B_i^{\eta} & \text{if} \quad \text{otherwise} \end{cases}$$

$$\text{UB} = \min\begin{cases} x_i^{\text{old}} + \text{move} \\ 1 \end{cases} \qquad \text{LB} = \min\begin{cases} x_i^{\text{old}} - \text{move} \\ 0 \end{cases}$$

boundary

$(\mathbf{L} \cdot \text{FlagSolid})\ \&\ (\mathbf{L} \cdot \text{FlagVoid})$

$$\text{UB} = \min\begin{cases} x_i^{\text{old}} + \text{move} \\ 1 \end{cases} \qquad \text{LB} = \min\begin{cases} x_i^{\text{old}} - \text{move} \\ 0.5 + \mu \end{cases}$$

Pure-solid

$!\,(\mathbf{L} \cdot \text{FlagVoid})$

$$\text{UB} = \min\begin{cases} x_i^{\text{old}} + \text{move} \\ 0.5 - \mu \end{cases} \qquad \text{LB} = \min\begin{cases} x_i^{\text{old}} - \text{move} \\ 0 \end{cases}$$

Pure-void

$!\,(\mathbf{L} \cdot \text{FlagSolid})$

Proposed method: O.C. Update

- Shape optimization — O.C.

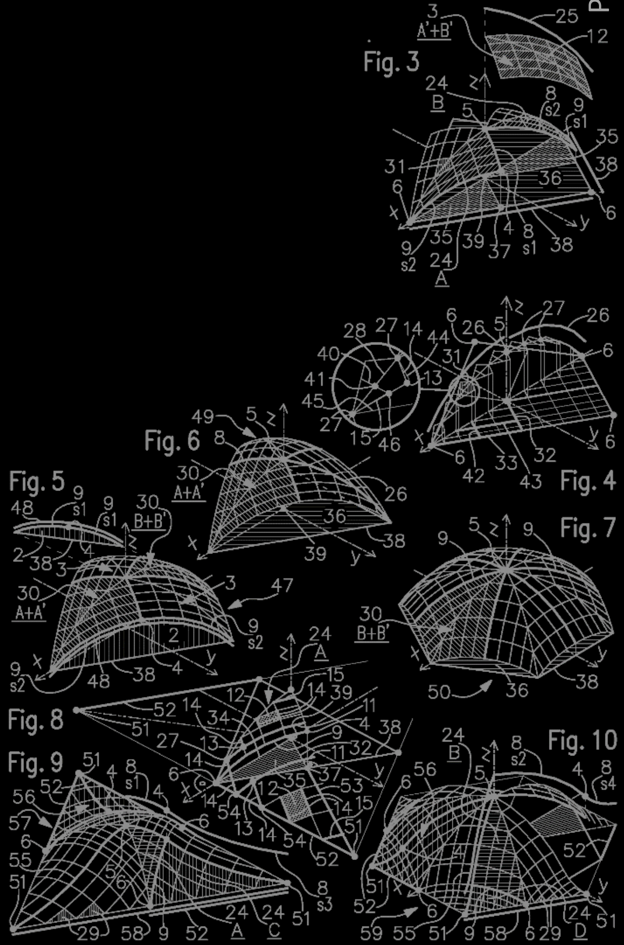
$$x_i^{\text{new}} = \begin{cases} \text{UB} & \text{if} \quad x_i^{\text{old}} \cdot B_i^{\eta} > \text{UB} \\ \text{LB} & \text{if} \quad x_i^{\text{old}} \cdot B_i^{\eta} < \text{LB} \\ x_i^{\text{old}} \cdot B_i^{\eta} & \text{if} \quad \text{otherwise} \end{cases}$$

$$\text{UB} = \min \begin{cases} x_i^{\text{old}} + \text{move} \\ \max \begin{cases} 1 - \text{PureVoid} \\ \text{PureVoid} \cdot (0.5 + \mu) \end{cases} \end{cases}$$

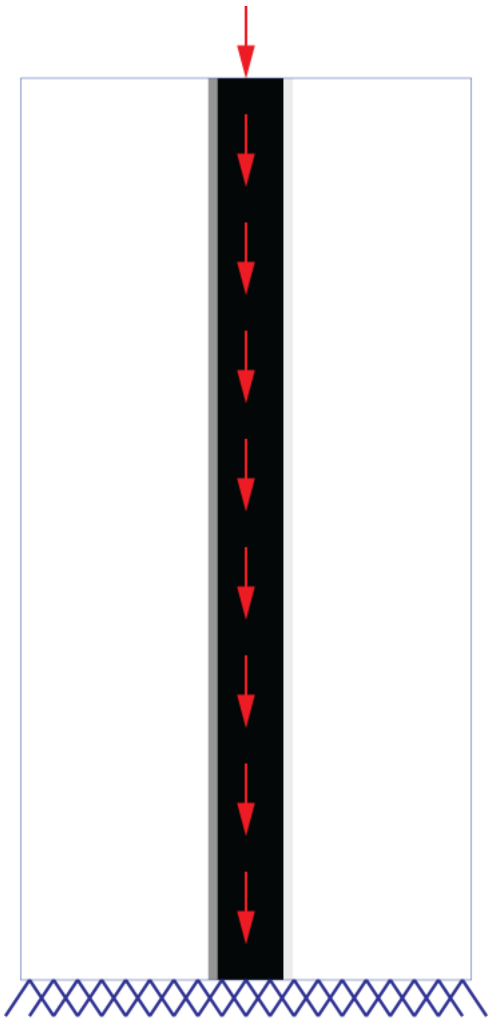
$$\text{LB} = \min \begin{cases} x_i^{\text{old}} - \text{move} \\ \text{PureSolid} \cdot (0.5 + \mu) \end{cases}$$

examples

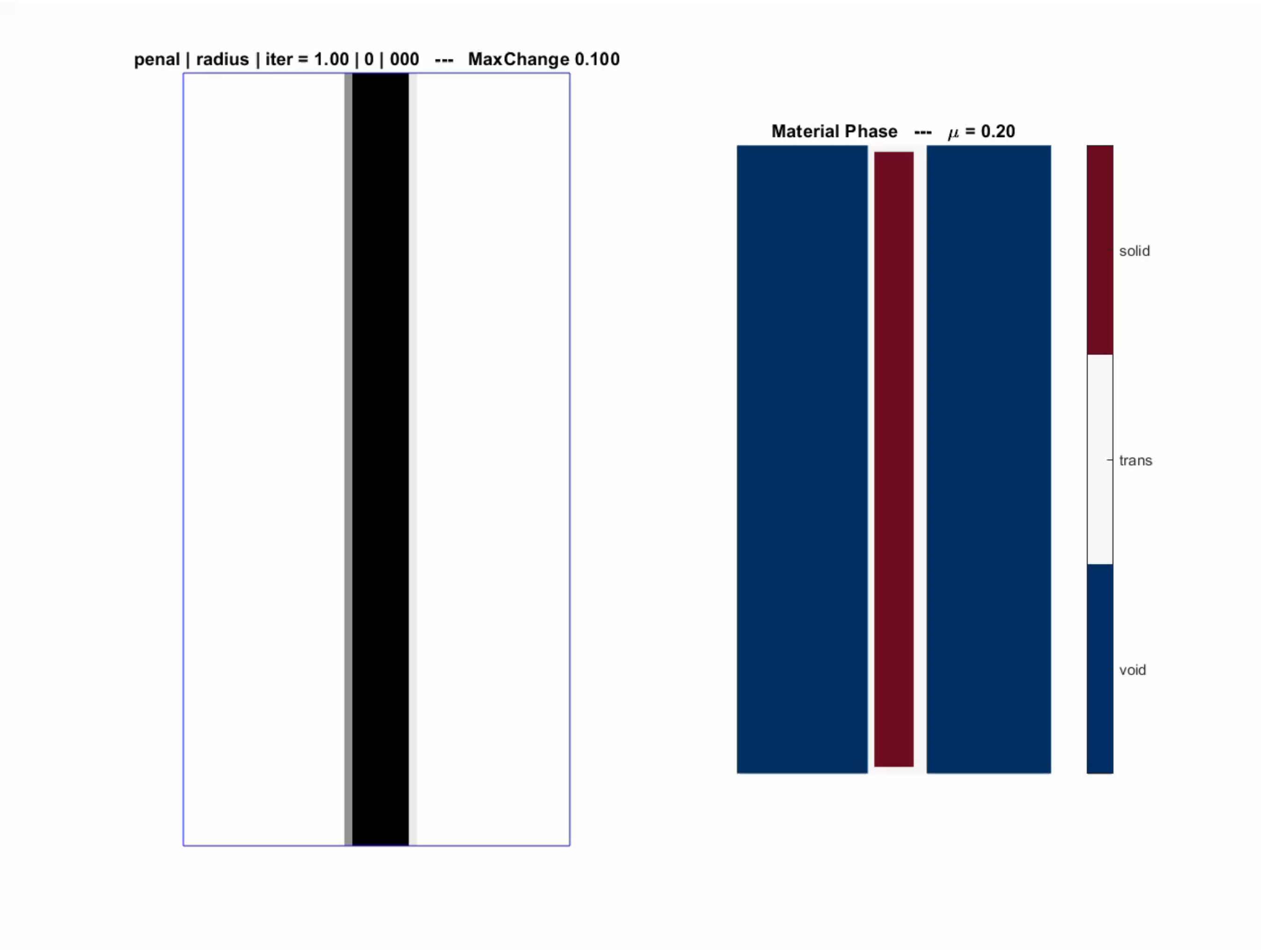
Patent US7591108



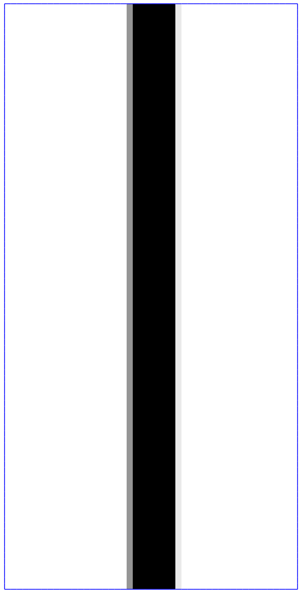
Ex1 : column + distrib. force

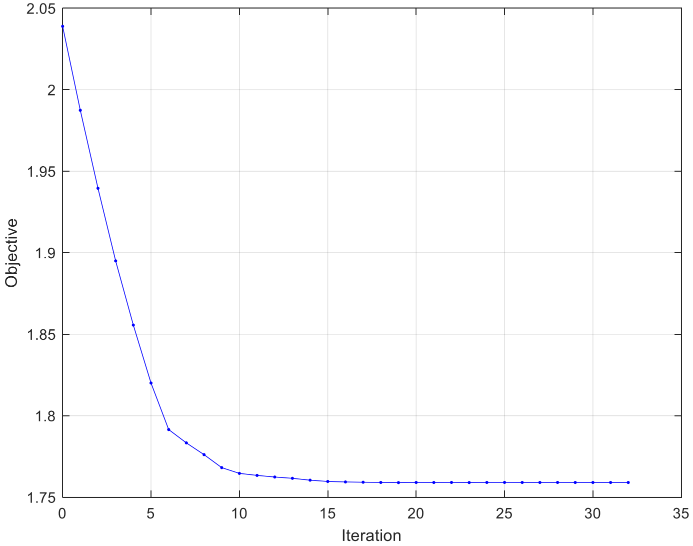


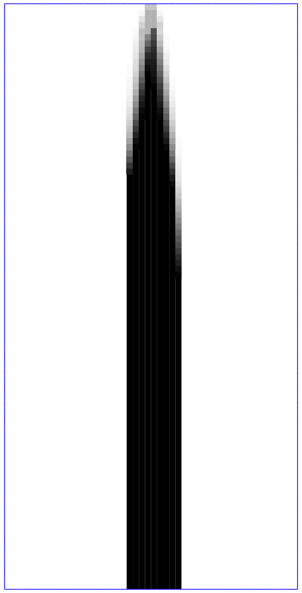
Ex1 : column + distrib. force



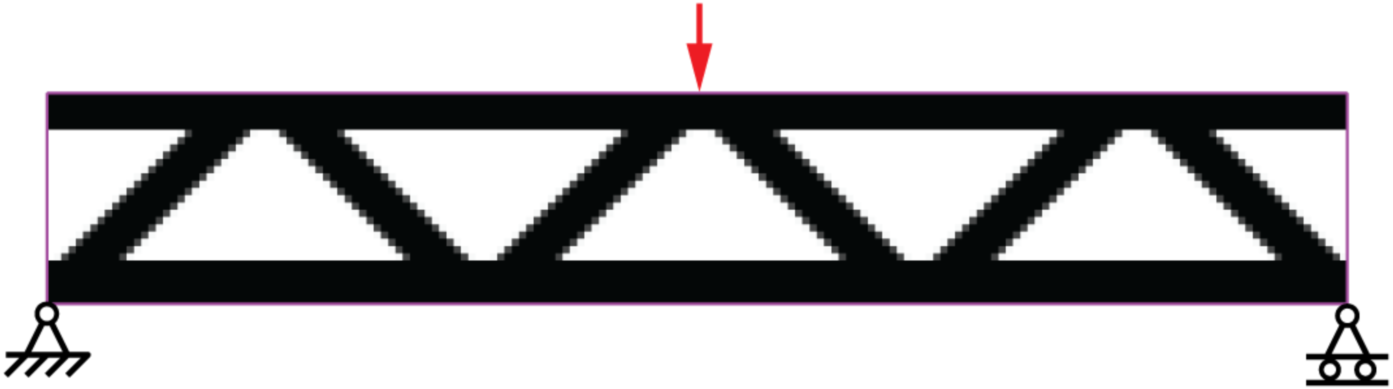
Ex1 : column + distrib. force



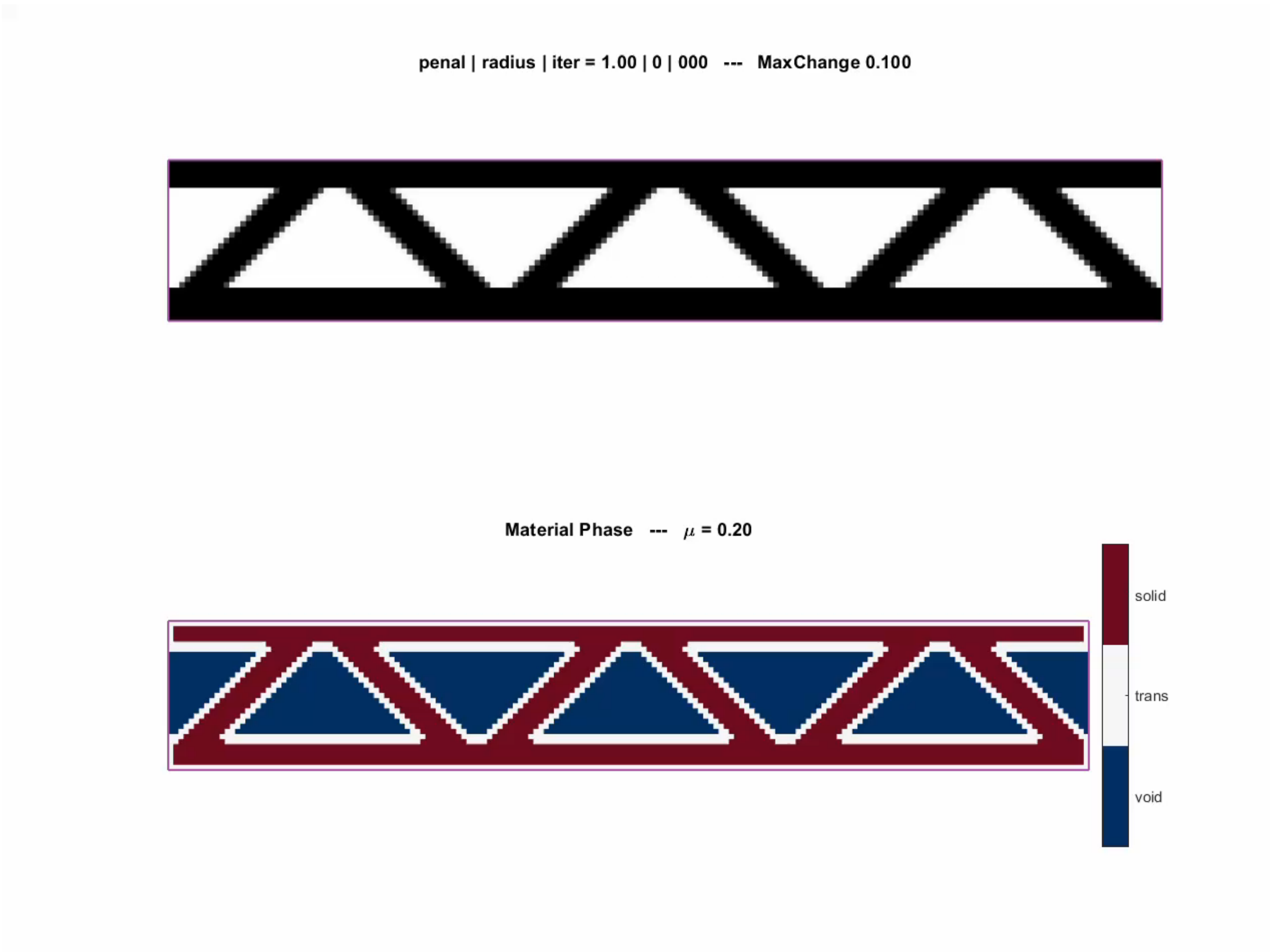




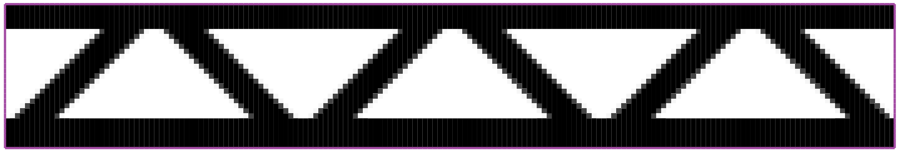
Ex2 : S.S. beam

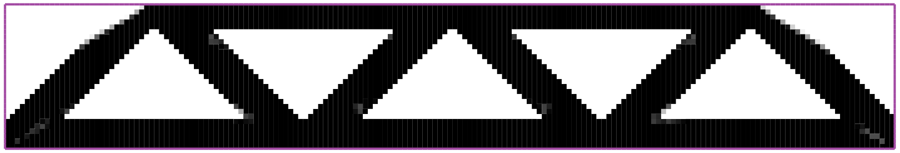


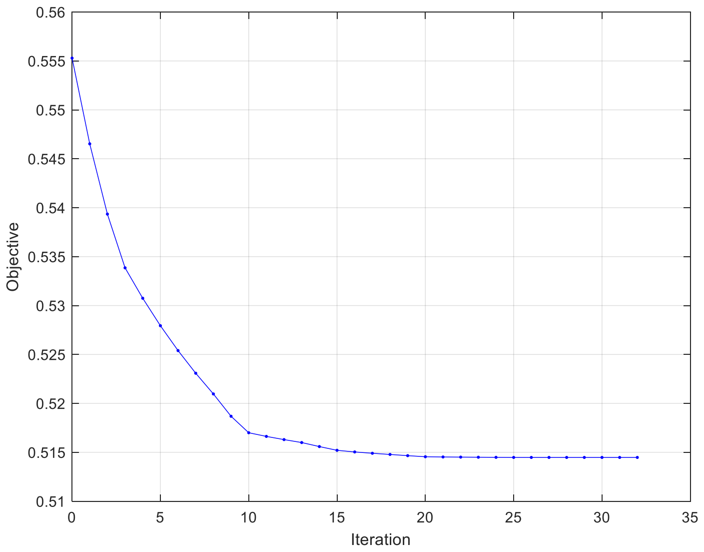
Ex2 : s.s. beam


penal | radius | iter = 1.00 | 0 | 000 --- MaxChange 0.100
Material Phase --- μ = 0.20
solid
trans
void

Ex2 : s.s. beam



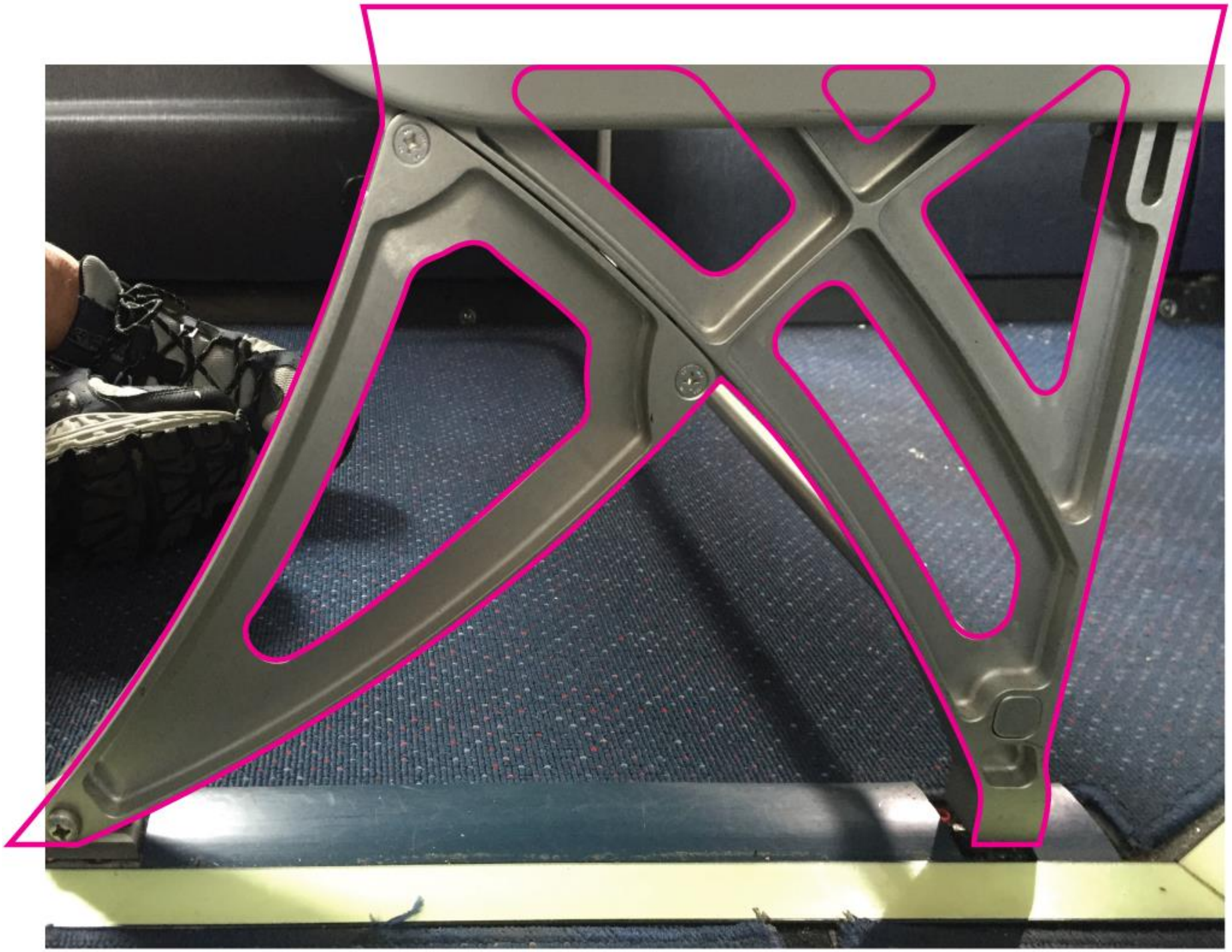




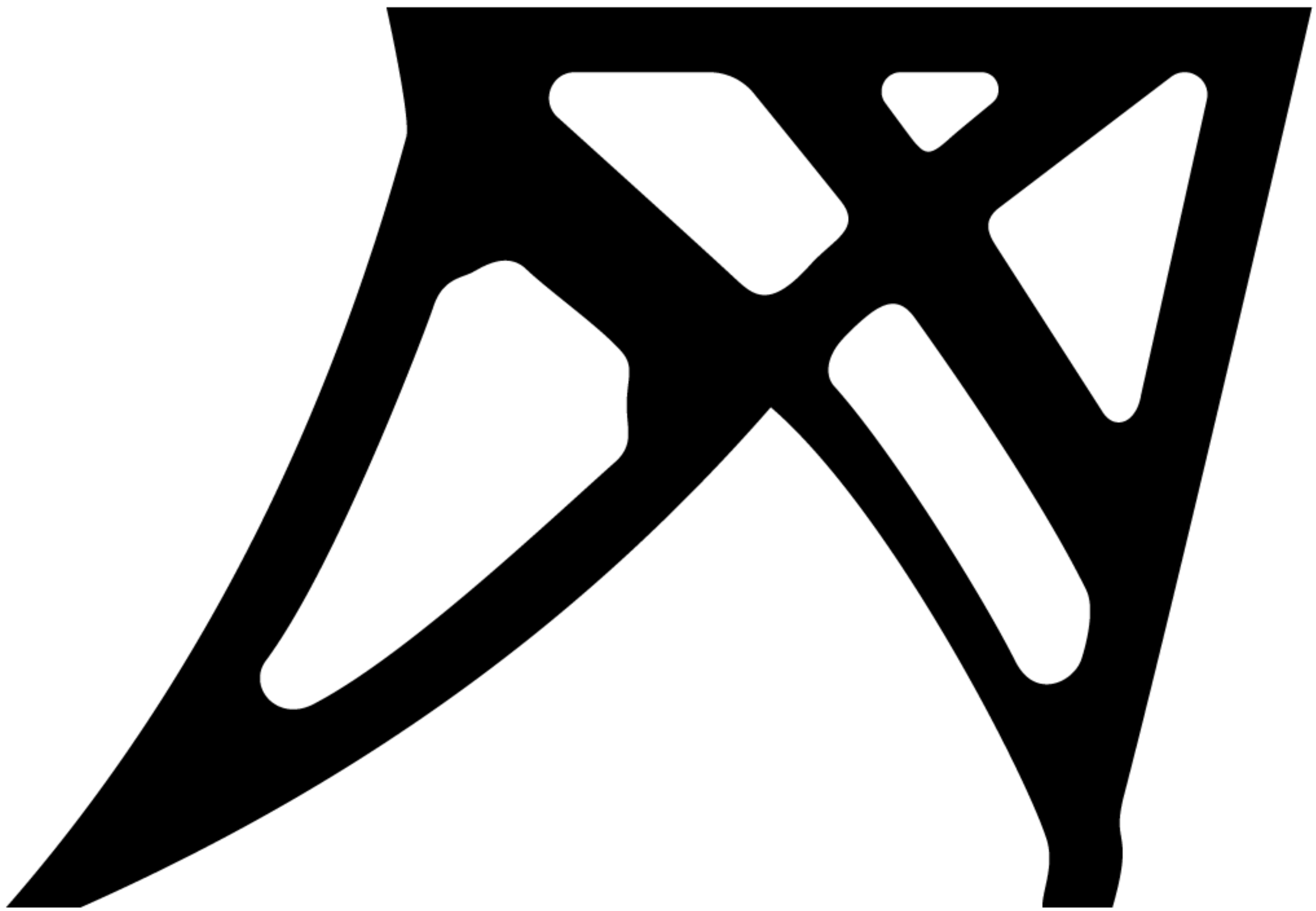
Objective
Iteration
0.56
0.555
0.55
0.545
0.54
0.535
0.53
0.525
0.52
0.515
0.51
0
5
10
15
20
25
30
35

Ex2 : airplane seat bracket

Ex2 : airplane seat bracket

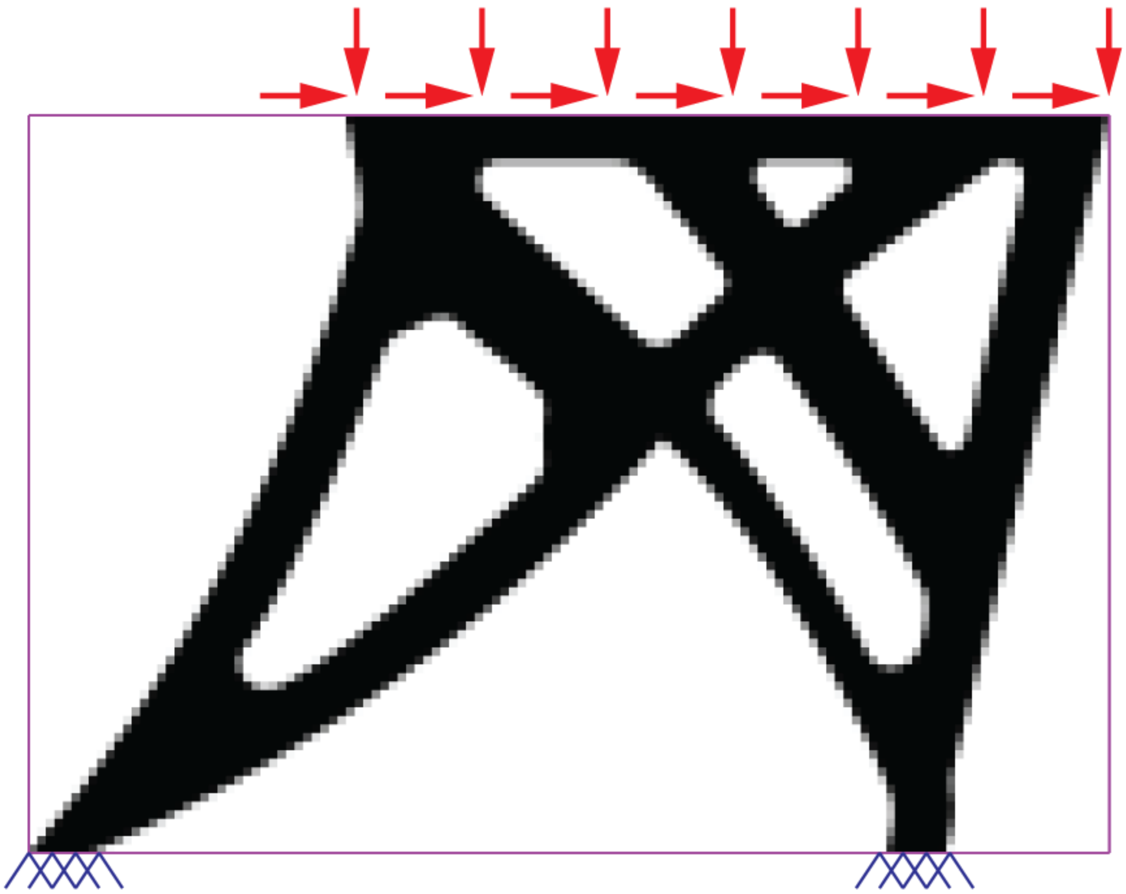


Ex2 : airplane seat bracket

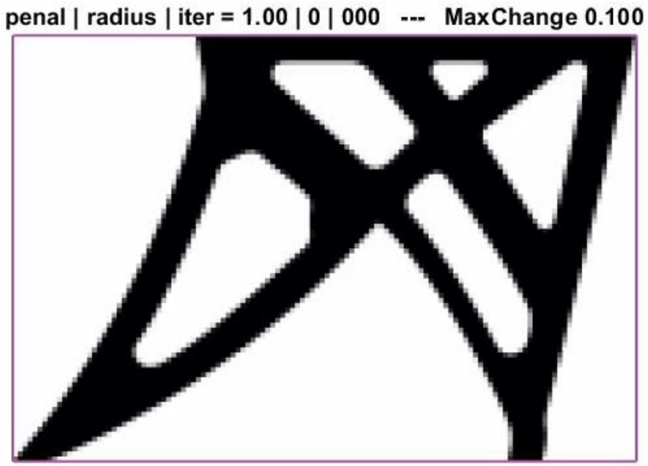


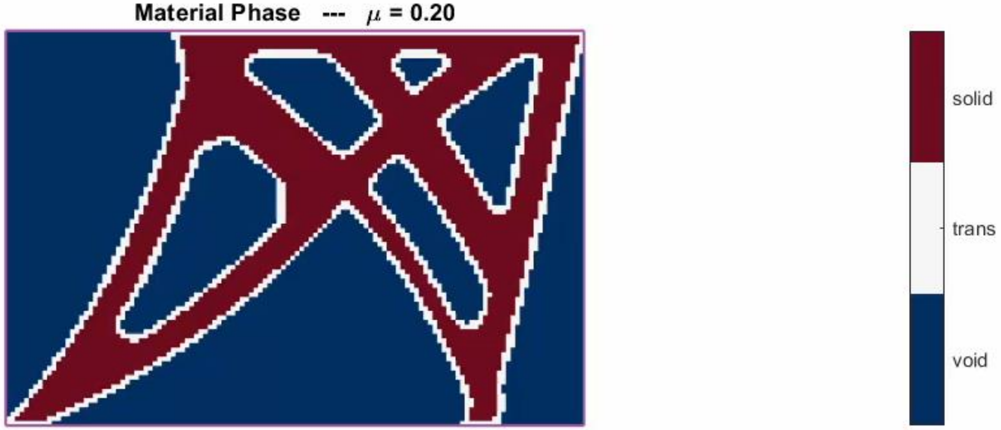
Ex2 : airplane seat bracket

- $F_v : F_h = 1 : 3$

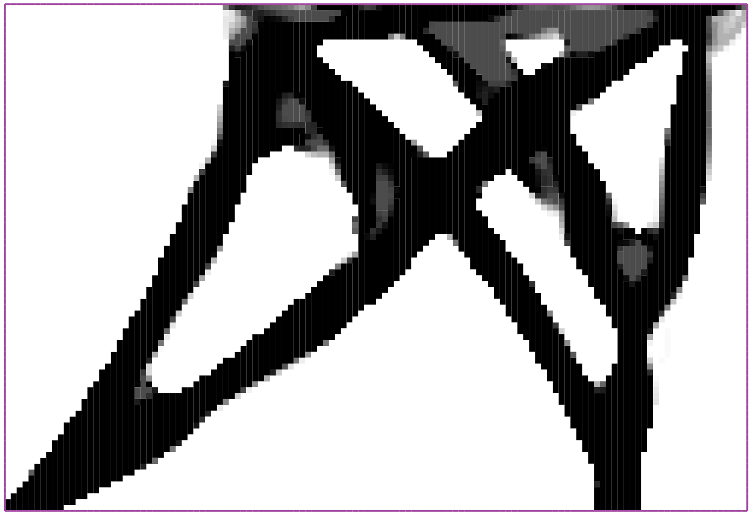


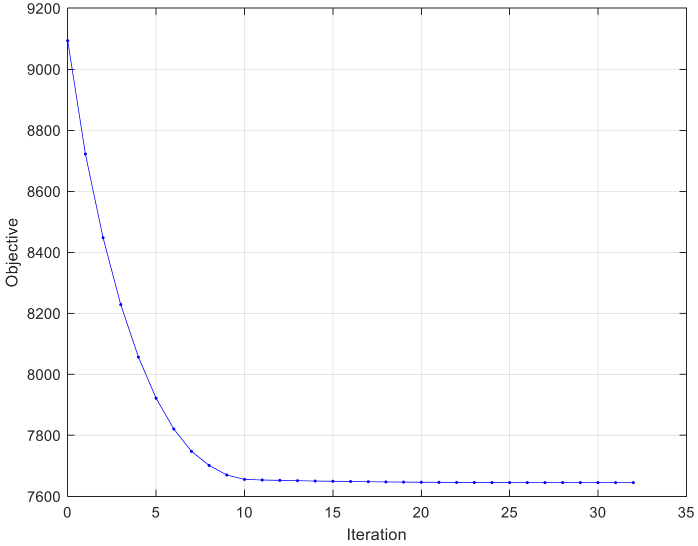
Ex2 : airplane seat bracket





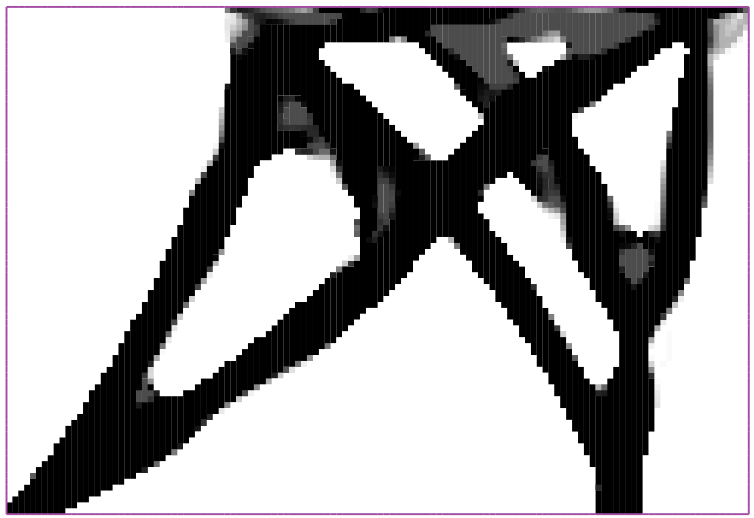
Ex2 : airplane seat bracket





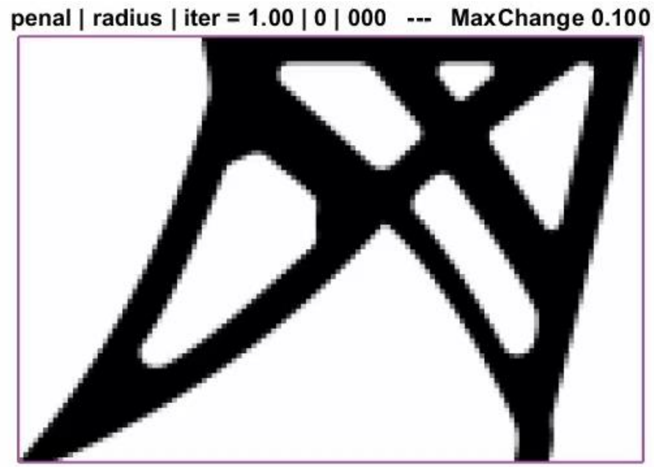
Ex2 : airplane seat bracket

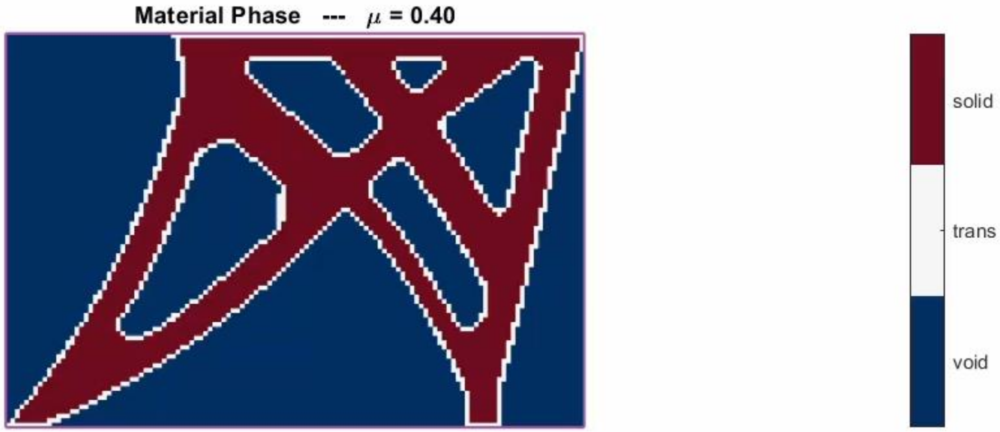
- Too much gray…



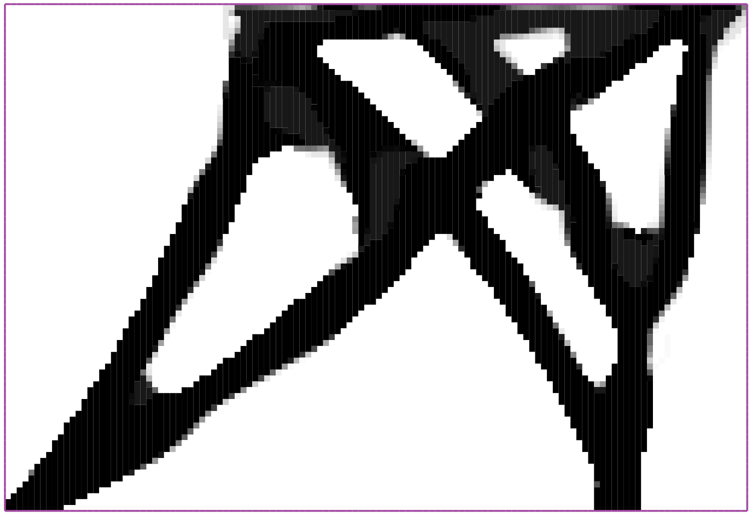
- Increase to $\mu = 0.40$

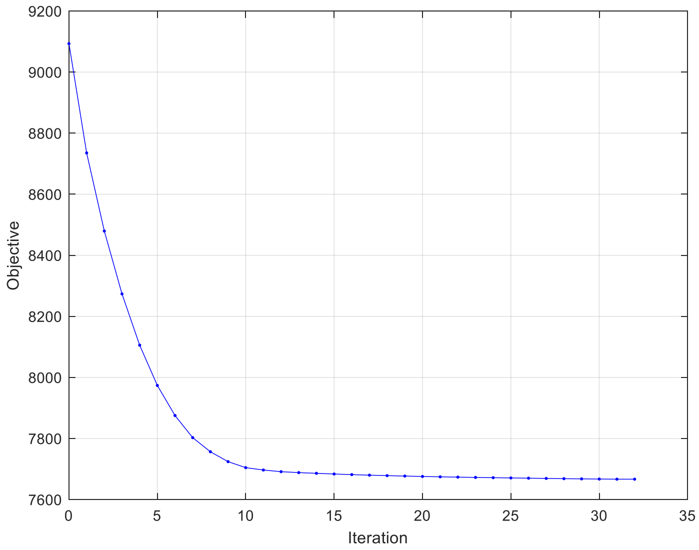
Ex2 : airplane seat bracket



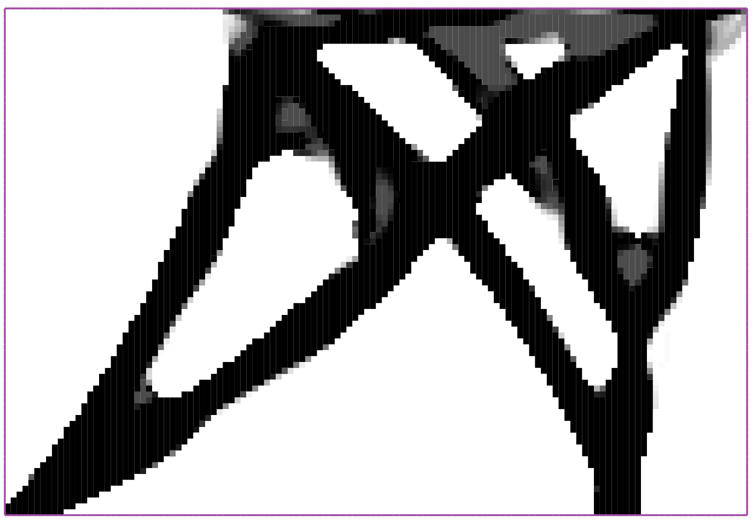


Ex2 : airplane seat bracket

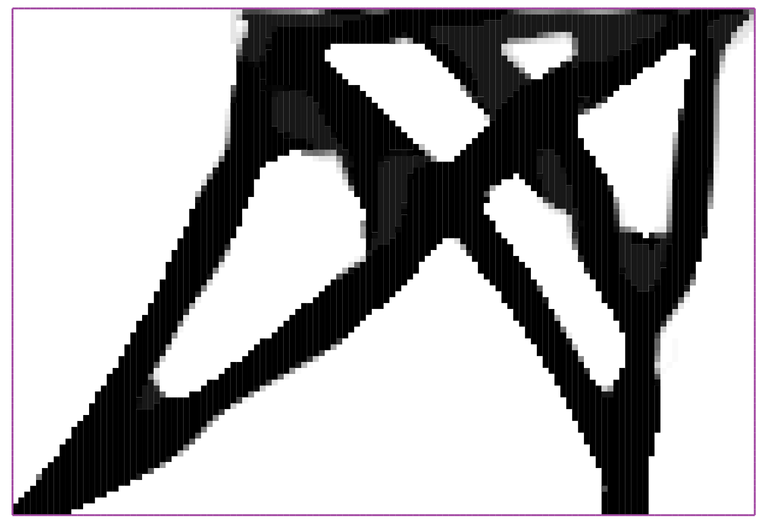




Ex2 : airplane seat bracket



μ = 0.20



μ = 0.40

summary

Proposed method

What it is…

- Simple
- Based on graph theory
- modified-OC
- any mesh
- Any objective (sensitivity must exist)
- Scales nicely (scaling of TOP)
- Continuation μ
- Extends to 3D!

What it isn't

- Topology optimization
- Fixes all and every problem
- Topology guarantee
  - «kissing» members
  - Void «droplet» problem

The end

?