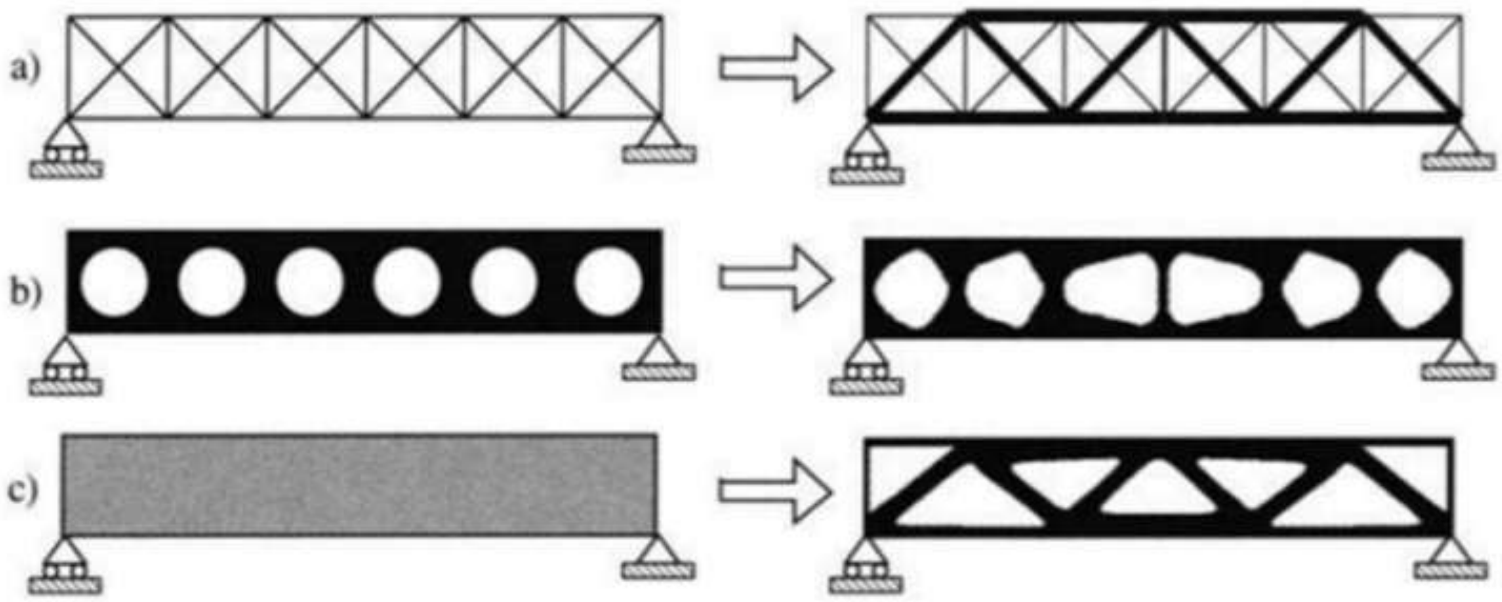
Topology optimization of a stress constrained and minimal compliance problems using MATLAB

Skoltech

Alexander Shumilov, IST

Categories of structural optimization



a) Sizing optimization of a truss structure, b) shape optimization, and c) topology optimization. The initial problems are shown at the left hand side and the optimal solutions are shown at the right [2].

Skoltech

Stress constrained problem

$$\begin{cases} \min \sum_{i}^{N} \rho_i v_i \\ such\ that \begin{cases} Ku = f \\ \sigma_i \leq \sigma_l\ if\ \rho > 0 \\ 0 \leq \rho_{min} \leq \rho_i \leq \rho_{max} \leq 1 \end{cases} \end{cases}$$

N is the number of elements, ***ρ*** is the density (and also the design variable), ***ρ_i*** is the elemental density, ***v_i*** is the elemental area/volume, ***K*** is the stiffness, ***u*** is the displacement, ***f*** is the external force, ***σ_i*** is the elemental stress measure, ***σ_l*** is the stress limit [1]

Skoltech

Algorithm for stress constrained problem

- Setup FE and stress analyses and filtering
- Until convergence
  - Perform FE and stress analyses
  - Check stop criteria, break if satisfied
  - Run optimization algorithm
    - Determine TM
      - Distribute RM
        - If stress limit is exceeded, TM = CM + MM
        - Else, TM = CM - MM
    - Set RM = TM
    - Until RM is small enough
      - Distribute RM to elements proportionally to their stress values
      - Apply filter
      - Apply density limits
      - Calculate AM
      - Update RM = TM – AM
    - Update density

TM is the target material amount

CM is the current material amount

MM is the material move amount

RM is the remaining material amount

AM is the actual material amount.

Skoltech

Minimal compliance problem

$$
\begin{cases}
\min C = u^T K u \\
\text{such that} \begin{cases}
Ku = f \\
\sum_{i}^{N} \rho_i v_i = M \\
0 \leq \rho_{min} \leq \rho_i \leq \rho_{max} \leq 1
\end{cases}
\end{cases}
$$

N is the number of elements, ***ρ*** is the density (and also the design variable), $\boldsymbol{\rho_i}$ is the elemental density, $\boldsymbol{v_i}$ is the elemental area/volume, ***K*** is the stiffness, ***u*** is the displacement, ***f*** is the external force, $\boldsymbol{\sigma_i}$ is the elemental stress measure, $\boldsymbol{\sigma_l}$ is the stress limit, ***C*** is the compliance and ***M*** is the total mass [1]

Algorithm for minimal compliance problem

- Setup FE and compliance analyses and filtering
- Determine TM
- Until convergence
  - Perform FE and compliance analyses
  - Check stop criteria, break if satisfied
  - Run optimization algorithm
    - Set RM = TM
    - Until RM is small enough
      - Distribute RM to elements proportionally to their compliance values
      - Apply filter
      - Apply density limits
      - Calculate AM
      - Update RM = TM – AM
    - Update density

TM is the target material amount

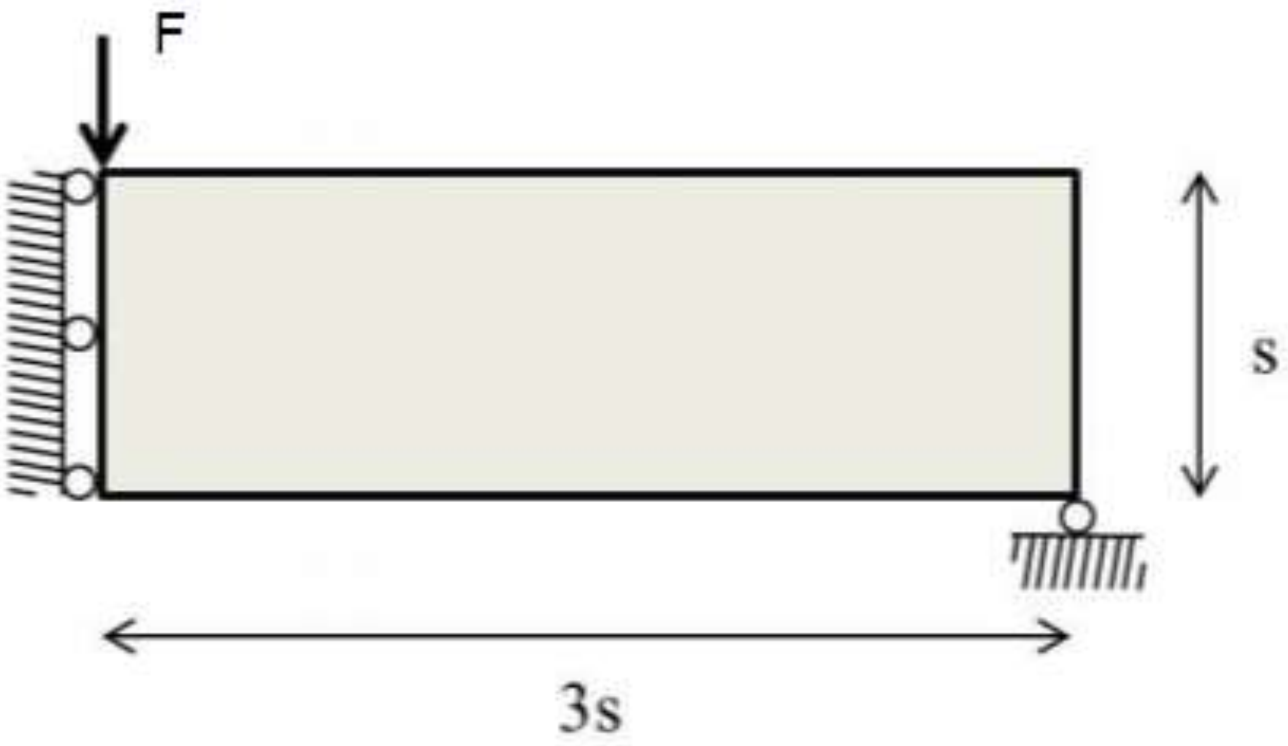
CM is the current material amount

MM is the material move amount

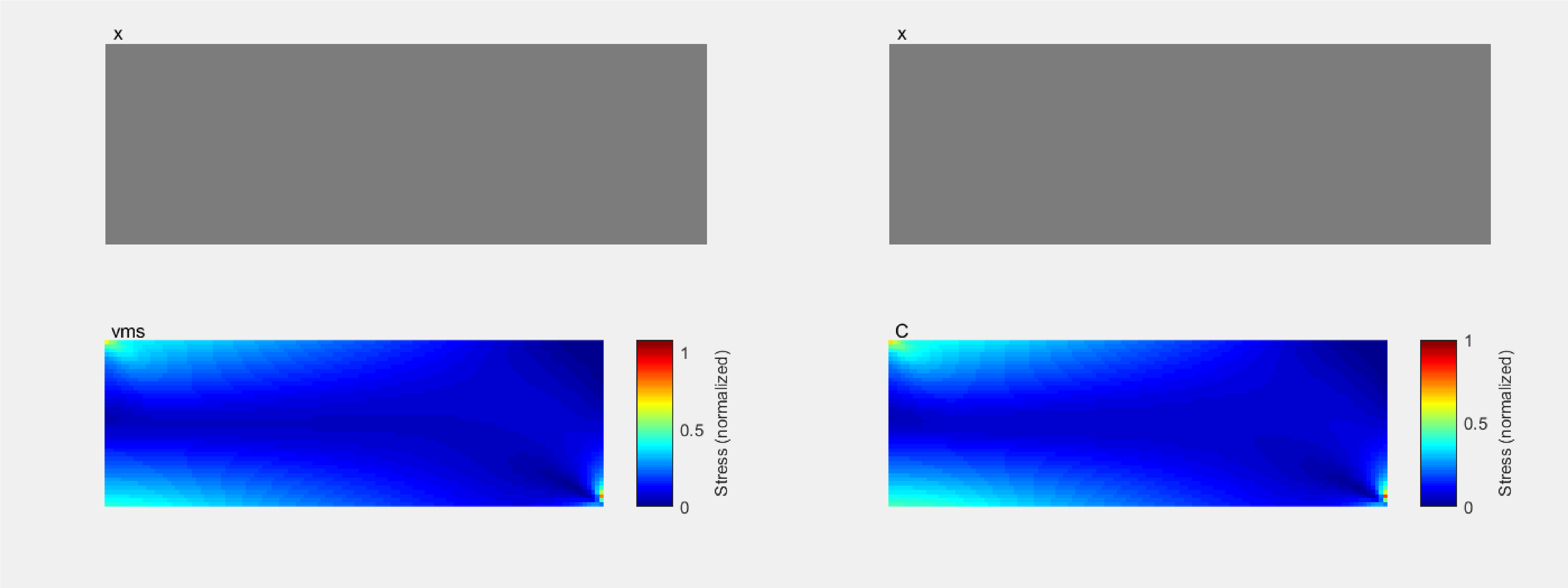
RM is the remaining material amount

AM is the actual material amount.

MBB-beam



Results (MATLAB)



Stress constrained problem

Minimal compliance problem

Reference list

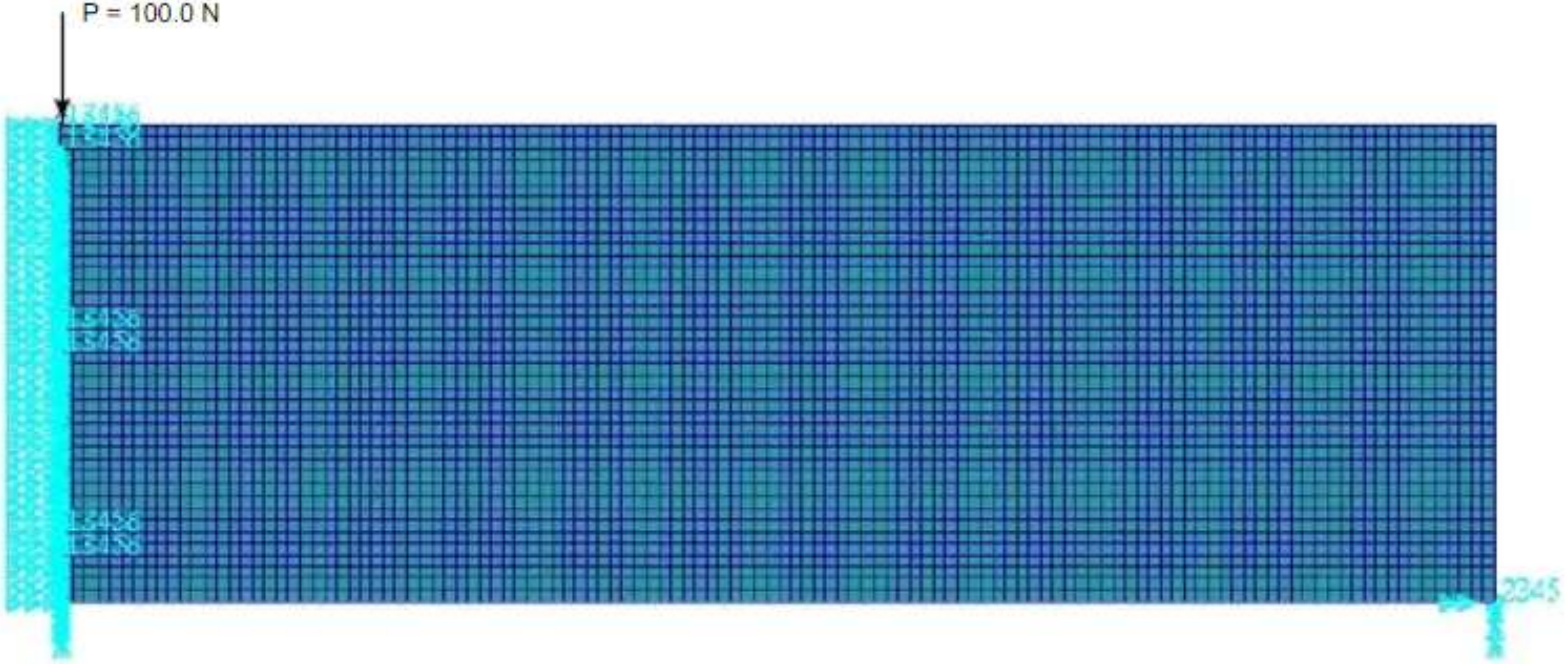
1. Biyikli, Emre & To, Albert. (2014). Proportional Topology Optimization: A new non-gradient method for solving stress constrained and minimum compliance problems and its implementation in MATLAB. PLOS ONE. 10. 10.1371/journal.pone.0145041.
2. Grinde, Seth. (2018). TOPOLOGY OPTIMIZATION FOR ADDITIVE MANUFACTURING USING SIMP METHOD.
3. Challis, V. J. (2009). A discrete level-set topology optimization code written in Matlab. Structural and Multidisciplinary Optimization, 41(3), 453–464
4. París, José & Navarrina, F & Colominas, Ignasi & Casteleiro, M. (2008). Advances in the statement of stress constraints in structural topology optimization.

Skoltech

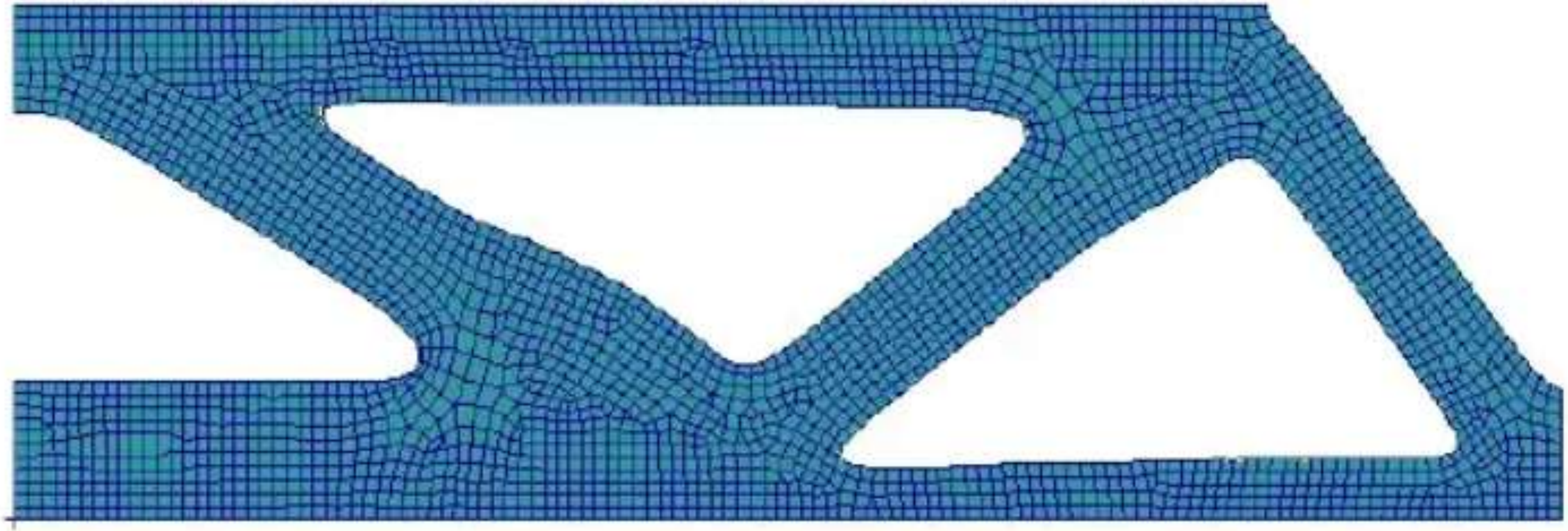
thx.

Skoltech

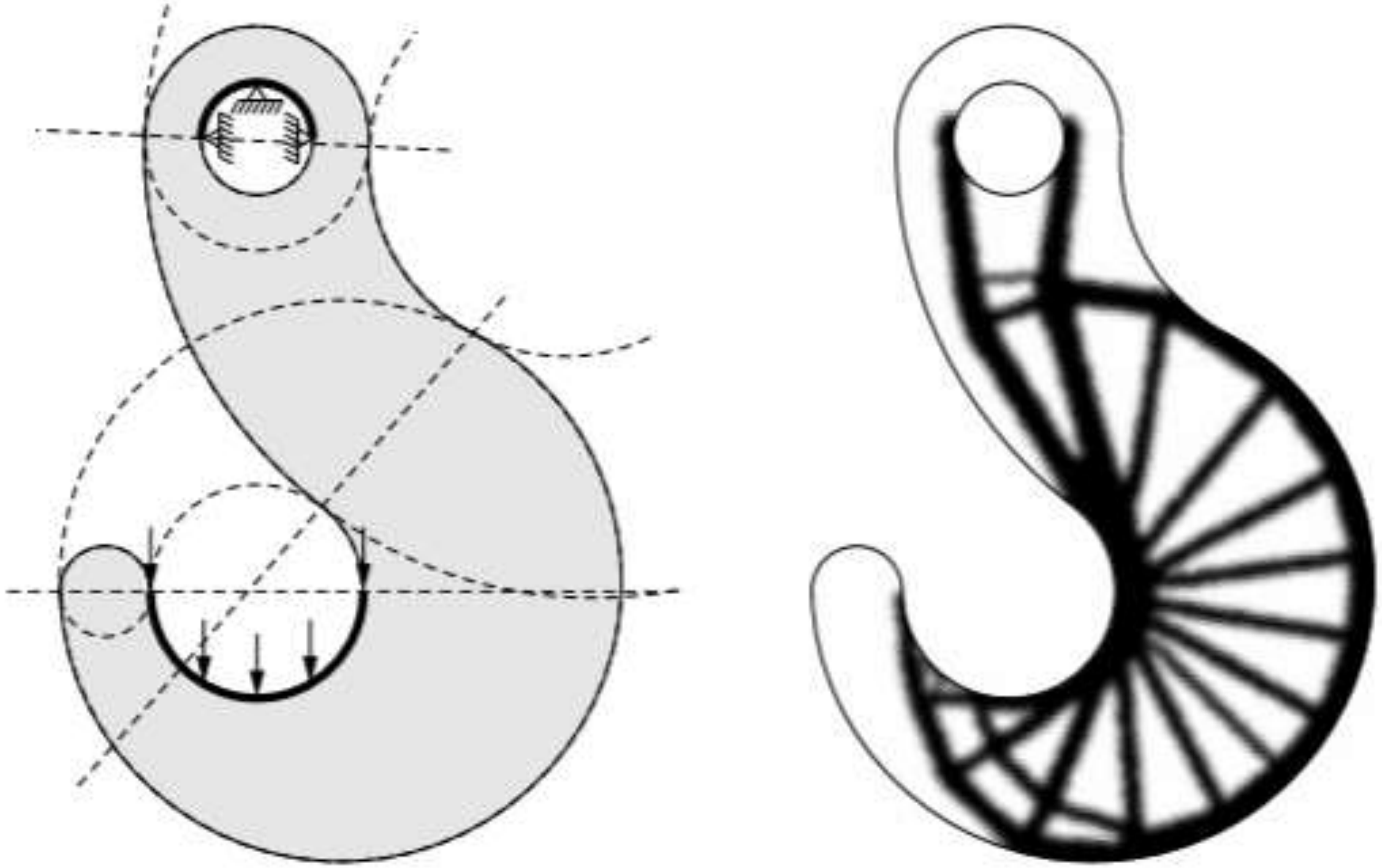
Appendix



Skoltech



Skoltech



(a) Hook geometry. (b) Hook final topology.

$$\rho_i = \frac{\sum w_{ij} d_j}{\sum w_{ij}} \ where \ w_{ij} = \begin{cases} \dfrac{r_0 - r_{ij}}{r_0} & for \ r_{ij} < r_0 \\ 0 & for \ r_{ij} \geq r_0 \end{cases}$$

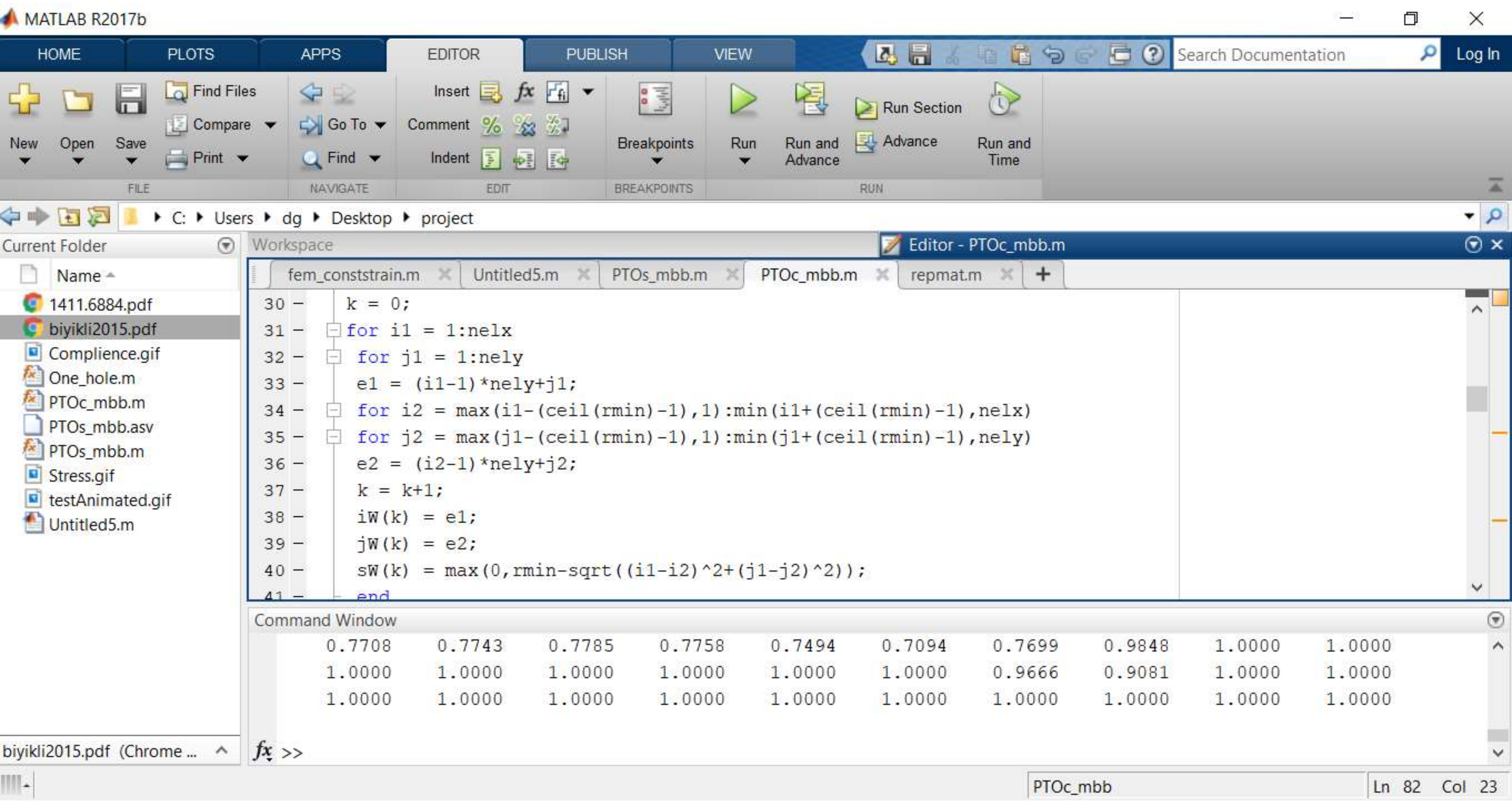
ρi is the filtered density of element i, wij is the filtering weight of elements i and j, dj is the non-filtered density of element j, rij is the distance between elements i and j, and r0 is the filter radius. The weight is inversely proportional to the distance between the element and its neighbors. In this sense, the cone density filtering is actually nothing but local averaging

Filtering is endorsed to be advantageous for many reasons:

Small scale features such as jagged edges, narrow members, and sharp interfaces are prevented.

As a result of smoothing, a blurred region around the structural members is obtained.

The algorithm is saved from getting stuck in local minima.



MATLAB R2017b
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Log In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Users ▸ dg ▸ Desktop ▸ project
Current Folder
Name
1411.6884.pdf
biyikli2015.pdf
Complience.gif
One_hole.m
PTOc_mbb.m
PTOs_mbb.asv
PTOs_mbb.m
Stress.gif
testAnimated.gif
Untitled5.m
biyikli2015.pdf (Chrome ...
Workspace
Editor - PTOc_mbb.m
fem_conststrain.m
Untitled5.m
PTOs_mbb.m
PTOc_mbb.m
repmat.m
30 k = 0;
31 for i1 = 1:nelx
32 for j1 = 1:nely
33 e1 = (i1-1)*nely+j1;
34 for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
35 for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
36 e2 = (i2-1)*nely+j2;
37 k = k+1;
38 iW(k) = e1;
39 jW(k) = e2;
40 sW(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
41 end
Command Window
0.7708 0.7743 0.7785 0.7758 0.7494 0.7094 0.7699 0.9848 1.0000 1.0000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.9666 0.9081 1.0000 1.0000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
fx >>
PTOc_mbb
Ln 82 Col 23

$$\sigma_{vM} = \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x\sigma_y + 3\sigma_{xy}^2}$$

$$\sigma = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} \qquad \sigma = DBu$$

$$D = \frac{E}{1 - v^2}\begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & (1-v)/2 \end{bmatrix}$$

$$B = \frac{1}{2L}\begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix}$$