Assignment in
Structural Optimization, 2019

Division of Solid Mechanics

The task is to implement and analyze different optimization methods. The theory related to the methods and the results should be presented in a well structured report. The methods should be implemented in Matlab where use can be made of suitable subroutines included in the CALFEM toolbox. The developed computer code should be attached as an appendix in the report.

Problem description

Topology optimization of two simple geometries should be performed. The two geometries and associated boundary conditions are illustrated in Figs. 1 and 2. The material of both structures is homogeneous, isotropic and linear elastic. Symmetry is used such that only the right half of Fig. 1 and the upper half of Fig. 2 are analyzed.

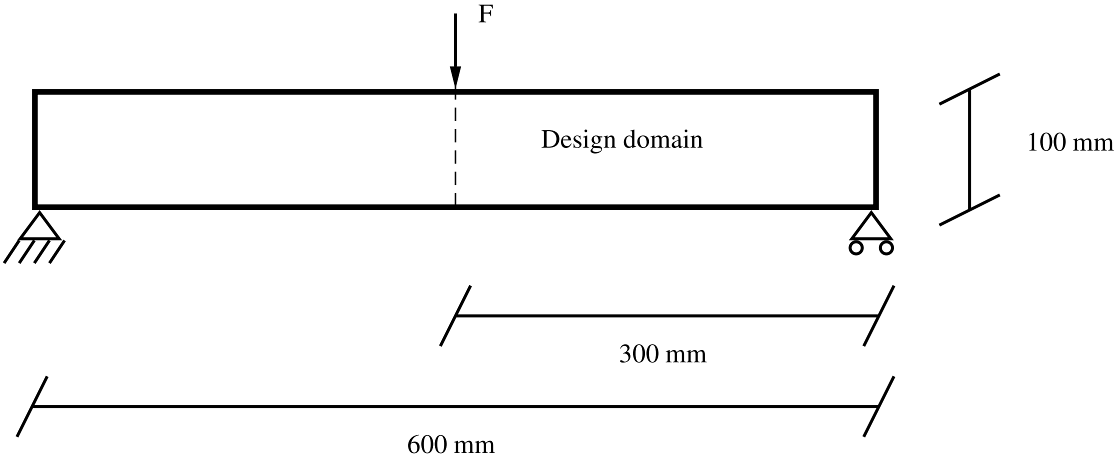

Figure 1: Illustration of geometry and boundary conditions.

Problem formulation

The optimization should be solved by the finite element method. To generate the finite element discretization, the function `genMesh.m` (available on the course homepage) should be used. Start by solving the optimization problems with a coarse mesh, e.g. 30×10 elements. When you are confident that your implementation works well, solve the same problems for a finer discretization. For a description of `genMesh.m`, see the Appendix.

In task a) the compliance of the beam in Fig. 1 should be minimized, and in task b) a filtering technique should be implemented to solve the optimization problem in a). In these first two tasks you should implement the OC method for updating the design.

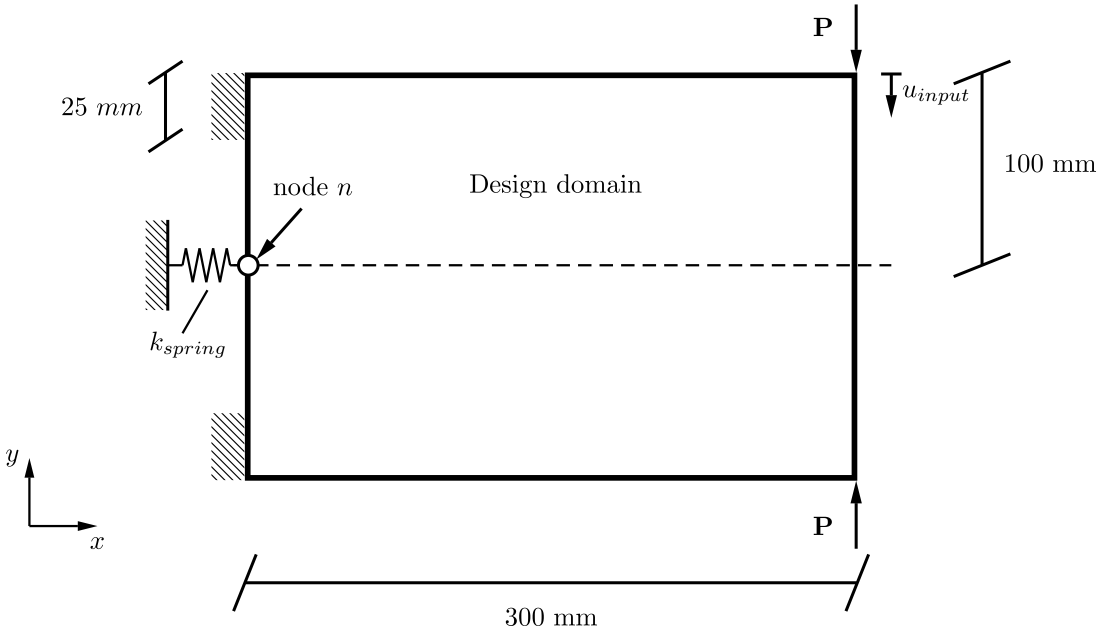

Figure 2: Illustration of geometry and boundary conditions. k_{spring} is the spring constant and u_{input} is the displacement in the direction of the external load.

In task c), a compliant mechanism should be optimized. This optimization problem includes two constraints. You are allowed (and recommended) to use the function `mma_solver` (also available on the course homepage) for updating the design. See the Appendix for a description of the function. The density filter implemented in task b) should also be included in this problem. Finally, the optional task d) is to implement a Heaviside thresholding technique to solve the previous tasks.

Plane stress conditions should be used in all optimization problems. The volume of the design domain is V_{box}, defined by Figs. 1 and 2 together with the thickness $t = 10\ mm$. Use the material parameters presented in [1], i.e. $E = 3\ GPa$ and $\nu = 0.4$.

Compliance minimization

a) Use the SIMP algorithm to derive a design optimized for maximum stiffness. You should investigate at least two different discretizations and different initial material distributions. The maximum allowed volume of the structure is $V_{max} = 0.4V_{box}$.

Filtering

b) Include a *density filter* (with the cone-shaped linearly decaying weighting function) in the SIMP algorithm developed in a). The theory describing density filters can be found in e.g. [2]. Examine the influence of the length scale parameter, r.

Design of compliant mechanism

c) Next, a compliant mechanism should be optimized such that the compression of an elastic spring (with spring constant k_{spring}) is maximized, i.e. the objective is to maximize the displacement of node n (see Fig. 2) in the negative x-direction which can be formulated as

$$\text{maximize } \boldsymbol{I}^T\boldsymbol{u} \quad \Longleftrightarrow \quad \text{minimize } -\boldsymbol{I}^T\boldsymbol{u}. \tag{1}$$

The vector $\boldsymbol{I}$ is constant and takes the value -1 in the x-degree of freedom of node n, and is zero elsewhere.

The spring constant is $k_{spring} = 10^7\ N/m$ and the magnitude of the applied load $P = 50\ N$. The displacement of the top right node of the structure in the direction of the load, u_{input}, is not allowed to exceed $u_{max}\ mm$, i.e. $u_{input}/u_{max} - 1 \leq 0$.

Investigate the effect of different values of u_{max}. Start with $u_{max} = 0.5\ mm$. Solve the optimization problem for $V_{max} = 0.4V_{box}$ and $V_{max} = 0.25V_{box}$.

Heaviside Thresholding (optional)

d) The filtering procedure generates designs with a diffuse boundary between solid and void material. To obtain a sharp boundary, i.e. a design that approaches $0-1$ values of the design, a Heaviside thresholding technique can be used. Implement this technique. The theory of the thresholding is described in e.g., [3]. The thresholding is applied by relating the filtered design variables $\tilde{\boldsymbol{\rho}}$ to the actual physical volume fractions $\hat{\boldsymbol{\rho}}$ through a smoothed Heaviside step function H according to

$$\hat{\rho}_e = H(\tilde{\rho}_e) = \frac{\tanh(\beta_H\omega) + \tanh(\beta_H(\tilde{\rho} - \omega))}{\tanh(\beta_H\omega) + \tanh(\beta_H(1 - \omega))}, \quad e = 1, \ldots, N \tag{2}$$

where ω and β_H are parameters controlling the thresholding. When you implement this technique, it is advised to keep ω constant, e.g. $\omega = 0.5$, and to start with a very small value of β_H, e.g. $\beta_H = 10^{-10}$. This choice results in that the thresholding has no effect since $H(x) \approx x$ as $\beta_H \rightarrow 0$, and thus you should obtain the same results as with only the filter implemented.

The thresholding should be implemented via a continuation strategy. This means that the parameter β_H should be increased in increments during the optimization procedure to its final value β_H^{max}. Use e.g. $\beta_H^{max} = 10$ and solve the problems in b) and c) again, and compare the results.

Some hints

- Finish the size optimization computer lab BEFORE you start with this assignment.
- Use SI-units.
- It is preferable to scale the objective function and the constraints. Normalize the constraints. For task c), the objective function is preferably scaled by a factor of u_{max}, i.e. $g_0 = -\boldsymbol{I}^T\boldsymbol{u}/u_{max}$.
- It is crucial that the sensitivities of the objective and constraint functions are correctly computed. One way to verify that the sensitivities are correctly computed is to compare them to numerical sensitivities (see Section 6.1 of the course book).

Procedure

The analysis is to be performed in CALFEM. A well structured concise report of your findings should be returned to the Div. of Solid Mechanics no later than **2019-03-13 16.00**. The results should be presented in the form of illustrative graphs and tables. Note that it should be possible to generate the results from the information provided in the report, i.e. numerical parameters used should be clearly stated in the report. MATLAB/CALFEM files (appendix) should be well structured and carefully commented. The reader of the report is assumed to have the same knowledge level as the author before taking the structural optimization course. It is possible to obtain up to 30 points. The task should be solved in groups of *two* (or *individually*). Keep the report as concise as possible. It is strongly recommended that you keep the report well below 10 pages excluding the appendix containing the code.

Submission

You should submit your report in **PDF** to FHLN01@solid.lth.se. In addition to your report you should also attach your m-files. Moreover, a **paper** version should be handed in to the Division of Solid Mechanics.

References

[1] Ole Sigmund. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines*, 25(4):493–524, 1997. doi: 10.1080/08905459708945415. URL `https://doi.org/10.1080/08905459708945415`.

[2] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4):401–424, 2007.

[3] Fengwen Wang, Boyan Stefanov Lazarov, Ole Sigmund, and Jakob Sondergaard Jensen. Interpolation scheme for fictitious domain techniques and topology optimization of finite strain elastic problems. *Computer Methods in Applied Mechanics and Engineering*, 276:453 – 472, 2014. ISSN 0045-7825.

Appendix

The function `genMesh.m` should be used to generate the finite element discretizations. The function does not include the spring element.

- The input arguments are the width `w` and height of the box `h`, the number of elements in the horizontal and vertical directions, `nx` and `ny`, respectively. Note that only quadratic elements are allowed.
- The output arguments are the degrees of freedom connectivity matrix `Edof`, the nodal connectivity matrix `Enod`, coordinate matrix `coord`, degrees of freedom `Dof`, the external load vector `F`, the identification vector `I` and the (fixed and symmetry) boundary conditions `bc`.
- The function is applicable for both geometries illustrated in Figs. 1 and 2; however, **the user must manually comment/uncomment parts of the code depending on which geometry is wanted. Please open the function and read the instructions and comments there.**
- A simple example of how to use the code:

```
clc; close all; clear all
[Edof,Enod,coord,Dof,F,I,bc]=genMesh(0.3,0.1,6,2);
[ex,ey]=coordxtr(Edof,coord,Dof,4);
figure(1); eldraw2(ex,ey,[1 2 2],Edof(:,1));
save myGeom.mat
```

The solver `mma_solver` should be called via the command
`[xmma,low,upp] = mma_solver(iOpt,RHO,xold1,xold2,df0dx,fval,dfdx,low,upp,paramVec)`,

where the input arguments are the current optimization iteration iOpt, the design variables RHO, the design variables one iteration ago xold1, the design variables two iterations ago xold2, the sensitivity of the objective function df0dx, the values of the constraints fval (size: 2×1), the sensitivity of the constraints (size: $nelm \times 2$), the lower and upper asymptotes low and upp, and the parameter vector paramVec $= \begin{bmatrix} nelm & s_{init} & s_{faster} & s_{slower} & \delta_o \end{bmatrix}^T$, where $nelm$ and δ_o is the number of elements and the lower bound of the design variables, respectively. The output arguments are the updated design variables xmma, and the updated lower and upper asymptotes low and upp. In the first optimization iteration, the asymptotes are initiated inside the function mma_solver based on s_{init} and RHO as described on p. 68 in the course book. This means that you do not need to specify initial values for low and upp.