# Structural optimization, project 2019: Q & A

Division of Solid Mechanics, Lund University

March 7, 2019

- **Q: I don't know where to start. What CALFEM functions should be used?**
  A: The structure of your Matlab program should be similar to the one used for solving the size optimization computer exercise. You should solve the optimization problems using four-node isoparametric elements. For instance, use `plani4e` to calculate the element stiffness matrix.
- **Q: How can I debug in the Matlab environment?**
  A: One option is to type "keyboard" at the place in the code where you want to debug. To continue the debugging type "dbcont" and to quit type "dbquit".
- **Q: I want to know what happens in a CALFEM routine. How can I do that?**
  A: Open the m-file or enter "type filename" in the Matlab command window.
- **Q: How many Gauss points/what integration rule should I use?**
  A: $2 \times 2$ Gauss points is OK.
- **Q: How can I plot the optimized design?**
  A: You can use the Matlab function `fill`. For instance, the command `fill(Ex',Ey',[RHO RHO RHO RHO]')` will plot the topology described by the design variables `RHO`.
- **Q: What value for $\alpha$ in the OC approximation should I use?**
  A: Any value $\alpha \geq 1$. Try different values.
- **Q: What is $a$ in the expression for $b_e^k$ on page 184 in the course book?**
  A: This is a typo. In Equation 9.4, $a$ should be replaced by $\alpha$.
- **Q: What value for the penalization factor $q$ in the SIMP formulation should I use?**
  A: Try different values. $q > 1$.
- **Q: How do I check that my sensitivities are correct?**
  A: Compare them to numerical sensitivities. See pages 97-98 in the course book. See the last page of this document.
- **Q: My program is very slow, especially for running task $c$). Have I done something wrong?**
  A: Not necessarily. Depending on how many elements you use and if you recalculate the constant part of the stiffness matrix or not, the computational time required can vary a lot. For task c, the program is generally slower because the MMA solver needs to handle two constraints. Start with a coarse mesh and then try with a finer one.
- **Q: When solving $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F}$, Matlab warns me for the stiffness matrix being close to singular. What is wrong?**
  A: Check the value of $\rho_{min}$, i.e. the minimum value of the design variable, together with your penalization factor. Hint: what is $\rho_{min}^q$ if $\rho_{min}$ is too small and e.g. $q = 3$?
- **Q: I get the error message "Error using fzero (line 274) The function values at the interval endpoints must differ in sign.", what should I do?**
  A: Start with an initial design that satisfies or is very close to satisfying the volume constraint.

- **Q: How are $\rho$, $\boldsymbol{x}$ and the thickness $t$ related?**
  A: As described in the course book in Chapter 9, $\rho$ is a continuous "thickness function" and $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$ is a discrete design variable vector that represent the thickness for each finite element. The minimum value of each $x_e$, where $e$ is the element number, is $\underline{\rho}$ and the maximum value is $\bar{\rho} = t$.

- **Q: What is `RHO`?**
  A: In the SIMP-method, one uses variables that represent the design with upper limit equal to 1. This makes it convenient to use a dimensionless design variable $\rho_e$ instead of $x_e$, with $0 < \underline{\rho} \leq \rho_e \leq 1$. In the answer to Question 5 in this document, `RHO` is this design variable vector $\boldsymbol{\rho} = \begin{bmatrix} \rho_1 & \rho_2 & \dots \rho_n \end{bmatrix}^T$. Using $\boldsymbol{\rho}$ as design variable vector, the volume of the structure in task a) is $\sum_{e=1}^{n} \rho_e a_e t$, where $a_e$ is the area of element $e$ and $n$ is the number of elements.

- **Q: Some Matlab-functions on the course homepage does not seem to work. Why?**
  A: You've probably accessed some functions related to previous years project. The only two Matlab-functions that should be downloaded from the course homepage and used in this project are `genMesh.m` and `mma_solver.p`.

- **Q: What values of the MMA parameters $s_{init}$, $s_{slower}$ and $s_{faster}$ should be used?**
  A: You are recommended to use $s_{init} = 0.5$, $s_{slower} = 0.7$ and $s_{faster} = 1.2$.

**Comparing numerical and analytical sensitivities**
To calculate the numerical sensitivity of the function $g_i$ with respect to design variable $x_e$, use forward or central differences. As a demonstration, here is some pseudo-code to calculate the numerical sensitivity of element number 89 by forward difference:

```
% 1) Use the initial guess
  RHO = 0.4*ones(nelm,1);  % This is the design variable vector
  RHO(89) = RHO(89) + h    % Add a small perturbation, h.

% Now run you program once (no optimization),
  ...
% Evaluate the function g_i and save that value
  gisave1 = ...

% 2) Now, start over and run your program again with the initial guess
  RHO = 0.4*ones(nelm,1);
  ...
% Evaluate the function g_i again and save that value
  gisave2 = ...
  ...
% Also: calculate the the analytical sensitivity
  analyticSens = dgidx(89)
  ...

% 3) Compute numerical sensitivity:
  numSens =    (gisave1 - gisave2)/h

% 4) Check if dgidx is accurate by comparing the fraction
  shouldBeCloseToOne = numSens/analyticSens
```

In the pseudo-code above, `dgidx` is the vector containting the analytical sensitivity $\dfrac{\partial g_i}{\partial \boldsymbol{x}}$ that you want to verify is correct. Note that you can use other initial guesses than `RHO = 0.4*ones(nelm,1)` and check other sensitivities than for element number 89 aswell.

Last updated: March 7, 2019, NI.