How to set the parameters of the structure and solve the topology optimization problem using the MATLAB routines developed by Gomes and Senne

We are going to show how to set the parameters of a structure and how to save them for solving the topology optimization problem. To do this, we take two examples: a cantilever beam and a force inverter.

1. Cantilever beam

Consider a cantilever beam, whose domain is showed in Figure 1.

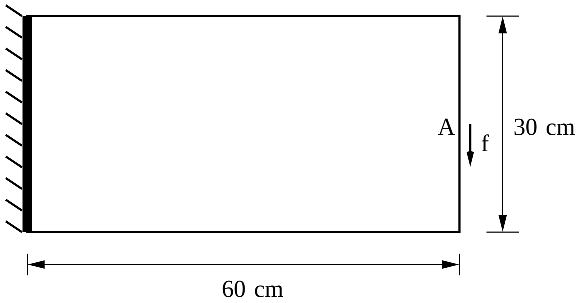

Figure 1: Design domain for a cantilever beam

.

In this example, a force $f = 1\,N$ is applied downwards in the center of the right edge of the beam. We suppose that the thickness of the structure is $e = 1\,cm$, the Poisson coefficient is $\sigma = 0.3$ and the Young modulus is $E = 1\,N/cm^2$. The domain is discretized into 1800 square elements with $1\,cm^2$ each, and the volume of the optimal structure is limited by $40\,\%$ of the design domain.

To set the parameters of the structures, call the routine `enter_structure_data.m`. We show how it appears on the MATLAB Command Window. After input each parameter, press Enter.

```
>> cantilever_beam_60x30 = enter_structure_data
Basis: 60
Height: 30
Number of elements in horizontal: 60
Number of elements in vertical: 30
Thickness: 1
Poisson coefficient: 0.3
Young modulus: 1
```

The next step is setting the data about the applied force on the structure. Again, after input each parameter, press Enter.

```
Load 1

x-component of load: 0
y-component of load: -1
Choose the load distribution (1 = concentrated, 2 = horizontal, 3 = vertical): 1
x-coordinate of load application: 60
y-coordinate of load application: 15
```

After giving the y-coordinate of load application, the routine will ask the user to give a second load, as showed below. In our example, we have only one load applied. Then, in this case, ignore the x-component of Load 2, only by pressing Enter. So, the routine will ask the user to give the informations about the first set of nodal supports. After each parameter given, press Enter.

```
Load 2

x-component of load:
Support 1

x-coordinate of support: 0
y-coordinate of support: 0
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 1
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 0
Is the support repeated in vertical (1 = yes, 0 = no)? 1
Support length (negative -> the support is below the initial point;
                positive -> the support is above the initial point): 30
```

In this case, we have 31 nodal supports on the left edge of the rectangular domain. After setting the support length, the routine will ask the user to give the 32^{nd} nodal support. In this example, we have only one set of nodal supports. Then, we ignore the second set of nodal supports by pressing Enter, and the MATLAB struct with the problem data will appear on the command window, as showed below.

```
Support 32

x-coordinate of support:


cantilever_beam_60x30 =

        b: 60
        h: 30
     nelx: 60
     nely: 30
        e: 1
       nu: 0.3000
        E: 1
      be2: 0.5000
      he2: 0.5000
       nx: 61
       ny: 31
     nnos: 1891
    nelem: 1800
    loads: [1x1 struct]
    force: [1x1 struct]
     supp: [1x1 struct]
```

We can save these MATLAB struct in the paste named “Tests” using the `save` command, to use it later for solving the topology optimization problem.

```
>> save cantilever_beam_60x30
```

To access the Young modulus, for example, we set `cantilever_beam60x30.E`. To access the informations about the nodal forces, we set `cantilever_beam60x30.force`, and so on.

```
>> cantilever_beam_60x30.E

ans =

     1
```

```
>> cantilever_beam_60x30.force

ans =

      Fx: 0
      Fy: -1
    node: 1876
```

Finally, to solve the topology optimization problem associated to the cantilever beam, we call the routine `test_cantilever_beam_60x30`, placed in the paste "Tests". If the user wants, he can redefine the input parameters of the main program (for example: initial point, filter radius, etc.) inside this routine.

2. Force inverter

Consider a force inverter, whose domain is showed in Figure 2.

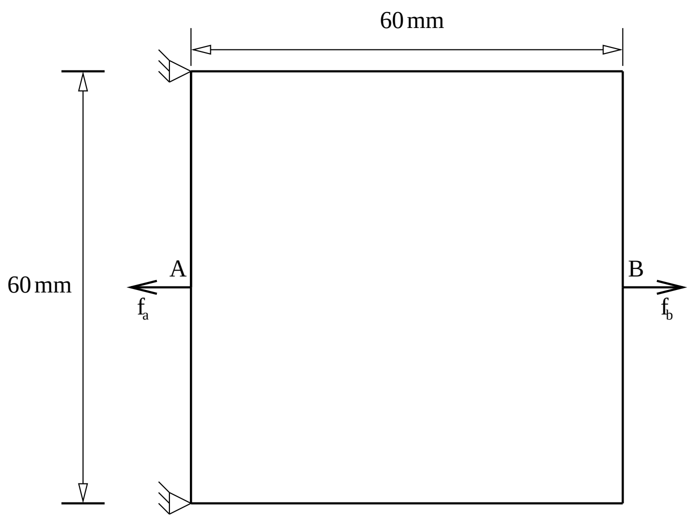

Figure 2: Design domain for a force inverter

.

In this example, an input force $f_a = 1\,N$ is applied to the center of the left side of the domain and the mechanism should generate an output force $f_b = 1\,N$ on the right side of the structure. We suppose that the thickness of the mechanism is $e = 1\,mm$, the Poisson coefficient is $\sigma = 0.3$ and the Young modulus is $E = 210000\,N/mm^2$. The domain is discretized into 3600 square elements with $1\,mm^2$ each, and the volume of the optimal structure is limited by 20 % of the design domain. But, due to the symmetry of the mechanism, only the upper half of the structure will be obtained.

Here, we follow the Nishiwaki *et al.* formulation (Nishiwaki, S.; Frecker, M. I.; Seungjae, M.; Kikuchi, N.; *Topology optimization of compliant mechanisms using the homogenization method.* International Journal for Numerical Methods in Engineering, 42, 1998, p. 535-559). In this work, two load cases are considered, with distinct boundary conditions: one related to the applied traction and to the dummy loads, and other associated to the reaction force. Having this in mind, we need to generate three MATLAB structs (one for each kind of load) to set the problem data, by calling the routine `enter_structure_data.m`. The procedure is analogous to the previous example, and we can save the three MATLAB structs to use it later to solve the related topology optimization problem.

Firstly, we print the MATLAB command window where the user sets the parameters related to the applied traction.

```
>> force_inverter_60x30_1 = enter_structure_data
Basis: 60
Height: 30
Number of elements in horizontal: 60
Number of elements in vertical: 30
Thickness: 1
Poisson coefficient: 0.3
Young modulus: 210000

Load 1

x-component of load: -1
y-component of load: 0
Choose the load distribution (1 = concentrated, 2 = horizontal, 3 = vertical): 1
x-coordinate of load application: 0
y-coordinate of load application: 0

Load 2

x-component of load:
Support 1

x-coordinate of support: 0
y-coordinate of support: 30
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 1
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 0
```

```
Is the support repeated in vertical (1 = yes, 0 = no)? 0

Support 2

x-coordinate of support: 0
y-coordinate of support: 0
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 0
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 1
Support length (negative -> the support is on the left of initial point;
                positive -> the support is on the right of initial point): 60
Is the support repeated in vertical (1 = yes, 0 = no)? 0

Support 63

x-coordinate of support:


force_inverter_60x30_1 =

        b: 60
        h: 30
     nelx: 60
     nely: 30
        e: 1
       nu: 0.3000
        E: 210000
      be2: 0.5000
      he2: 0.5000
       nx: 61
       ny: 31
     nnos: 1891
    nelem: 1800
    loads: [1x1 struct]
    force: [1x1 struct]
     supp: [1x1 struct]

>> save force_inverter_60x30_1
```

Secondly, we print the MATLAB command window where the user sets the parameters related to the dummy load.

```
>> force_inverter_60x30_2 = enter_structure_data
Basis: 60
Height: 30
Number of elements in horizontal: 60
Number of elements in vertical: 30
Thickness: 1
Poisson coefficient: 0.3
Young modulus: 210000

Load 1

x-component of load: 1
y-component of load: 0
Choose the load distribution (1 = concentrated, 2 = horizontal, 3 = vertical): 1
x-coordinate of load application: 60
y-coordinate of load application: 0

Load 2

x-component of load:
Support 1

x-coordinate of support: 0
y-coordinate of support: 30
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 1
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 0
Is the support repeated in vertical (1 = yes, 0 = no)? 0

Support 2

x-coordinate of support: 0
y-coordinate of support: 0
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 0
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 1
Support length (negative -> the support is on the left of initial point;
                positive -> the support is on the right of initial point): 60
Is the support repeated in vertical (1 = yes, 0 = no)? 0

Support 63
```

```
x-coordinate of support:


force_inverter_60x30_2 =

        b: 60
        h: 30
     nelx: 60
     nely: 30
        e: 1
       nu: 0.3000
        E: 210000
      be2: 0.5000
      he2: 0.5000
       nx: 61
       ny: 31
     nnos: 1891
    nelem: 1800
    loads: [1x1 struct]
    force: [1x1 struct]
     supp: [1x1 struct]

>> save force_inverter_60x30_2
```

Thirdly, we print the MATLAB command window where the user sets the parameters related to the reaction force.

```
>> force_inverter_60x30_3 = enter_structure_data
Basis: 60
Height: 30
Number of elements in horizontal: 60
Number of elements in vertical: 30
Thickness: 1
Poisson coefficient: 0.3
Young modulus: 210000

Load 1

x-component of load: -1
y-component of load: 0
Choose the load distribution (1 = concentrated, 2 = horizontal, 3 = vertical): 1
```

```
x-coordinate of load application: 60
y-coordinate of load application: 0

Load 2

x-component of load:
Support 1

x-coordinate of support: 0
y-coordinate of support: 30
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 1
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 0
Is the support repeated in vertical (1 = yes, 0 = no)? 0

Support 2

x-coordinate of support: 0
y-coordinate of support: 0
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 1
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 0
Is the support repeated in vertical (1 = yes, 0 = no)? 0

Support 3

x-coordinate of support: 1
y-coordinate of support: 0
Does the support prevent the movement in horizontal (1 = yes, 0 = no)? 0
Does the support prevent the movement in vertical (1 = yes, 0 = no)? 1
Is the support repeated in horizontal (1 = yes, 0 = no)? 1
Support length (negative -> the support is on the left of initial point;
                positive -> the support is on the right of initial point): 59
Is the support repeated in vertical (1 = yes, 0 = no)? 0

Support 63

x-coordinate of support:


force_inverter_60x30_3 =

        b: 60
```

```
        h: 30
     nelx: 60
     nely: 30
        e: 1
       nu: 0.3000
        E: 210000
      be2: 0.5000
      he2: 0.5000
       nx: 61
       ny: 31
     nnos: 1891
    nelem: 1800
    loads: [1x1 struct]
    force: [1x1 struct]
     supp: [1x1 struct]

>> save force_inverter_60x30_3
```

Finally, to solve the topology optimization problem associated to the force inverter, we call the routine `test_force_inverter_60x30`, placed in the paste "Tests". If the user wants, he can redefine the input parameters of the main program (for example: initial point, filter radius, etc.) inside this routine.