

# FEM Solver with Matlab

John Lian, Originally written in April 2015

February 11, 2016

## 1 The exercise

This document was part of a homework assignment originally submitted for MECH 546 - Finite Element Methods in Solid Mechanics at McGill University. The aim of the exercise was to build the stiffness matrix of the system depicted in Figure 1. The mesh was to be generated using FreeFem++ and imported into Matlab using the linear triangular T3 finite elements and assuming plane strains and a standard steel. The external force stems from (1) the vertical gravity with  $g = 1000 \text{ m/s}^2$  and (2) a constant pressure  $P = 10^4 \text{ N/m}^2$ .

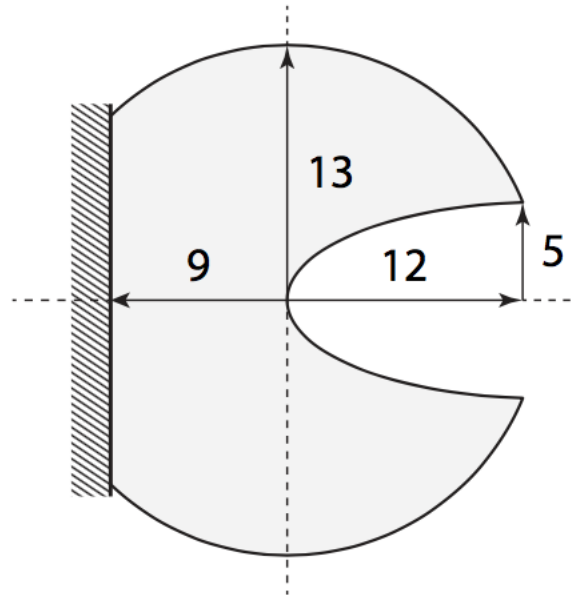


Figure 1: System of interest. Distances are expressed in meters

## 2 The solver and the methods

The system depicted was meshed in FreeFem++ with, essentially, a difference between a circle and an ellipse with a cut-off on the right. The circle has a radius of  $R = 13$  and the ellipse is given by

$$\left(\frac{x-12}{12}\right)^2 + \frac{y^2}{5^2} = 1, \quad (1)$$

thus, the ellipse is described with  $a = 12$  and  $b = 5$ . The mesh was implemented in FreeFem++ using

```
border aa(t=0,2*sqrt(R^2-9^2)){x=-9;y=ya-t;label=1;};
border bb(t=0,21){x=-9+t;y=-1*sqrt(R^2-x^2);label=2;};
border cc(t=0,12){x=12-t;y=-b*sqrt(1-(t/a)^2);label=3;};
border dd(t=0,12){x=t;y=b*sqrt(1-((t-12)/a)^2);label=4;};
border ee(t=0,21){x=12-t;y=sqrt(R^2-x^2);label=5;};
```

A FEM solver was then implemented in MATLAB. In order to make the implementation more efficient (fewer for loops), some *tricks* were employed. To begin, let  $[T^e]$  be the triangle geometric matrix defined as

$$[T^e] = \begin{bmatrix} 1 & 1 & 1 \\ x_1^e & x_2^e & x_3^e \\ y_1^e & y_2^e & y_3^e \end{bmatrix}. \quad (2)$$

Thus, the area of the triangle is given as  $|T^e|$ . With that, the elemental stiffness matrix is given as

$$[K^e] = \frac{1}{4|T^e|} [B]^T [D] [B] \quad (3)$$

for a T3 element. It can be computed simultaneously for all indices using the fact that

$$\begin{bmatrix} \frac{\partial N_1^4 Q}{\partial x} & \frac{\partial N_1^4 Q}{\partial y} \\ \frac{\partial N_2^4 Q}{\partial x} & \frac{\partial N_2^4 Q}{\partial y} \\ \frac{\partial N_3^4 Q}{\partial x} & \frac{\partial N_3^4 Q}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1^e & x_2^e & x_3^e \\ y_1^e & y_2^e & y_3^e \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{2|T^e|} \begin{bmatrix} y_2^e - y_3^e & x_3^e - x_2^e \\ y_3^e - y_1^e & x_1^e - x_3^e \\ y_1^e - y_2^e & x_2^e - x_1^e \end{bmatrix} \quad (4)$$

Then, the assembly operation for the global stiffness matrix is given by

$$[K_{ij}] = \sum_{T \in \Omega} [K_{ij}^e], \quad (5)$$

where  $T$  denotes T3 elements.

Using similar logic, the force vector is given by

$$(F_i) = \sum_{T \in \Omega} (F_{\Omega i}^e) + \sum_{E \in \partial \Omega} (F_{\partial \Omega i}^e), \quad (6)$$

where  $E$  are edges of the domain.

Assuming body forces  $f_\Omega = (f_1, f_2)$  are given at the mesh nodes, the integral can be approximated by

$$\int_{\Omega} [N^e]^T f_\Omega d\Omega = \frac{1}{6} \begin{vmatrix} x_2^e - x_1^e & x_3^e - x_1^e \\ y_2^e - y_1^e & y_3^e - y_1^e \end{vmatrix} f_i(x_c, y_c), \quad j = \text{mod}(i - 1, 2) + 1 \quad (7)$$

where  $(x_c, y_c)$  is the centre of mass of the triangle  $T$ . With the assumption on  $f$ ,

$$f_j(x_s, y_s) = (f_j(x_1, y_1) + f_j(x_2, y_2) + f_j(x_3, y_3))/3, \quad (8)$$

we can compute the body force using vectorization techniques given in [Koko, 2007].

Integrals involving Neumann conditions can be approximated using the value of  $f_{\partial\Omega}$  at the centre of the edge  $E$

$$\int_{\partial\Omega} [N^e]^T f_{\partial\Omega} d\Omega = \frac{1}{2} |E| g_j(x_c, y_c), \quad j = \text{mod}(i - 1, 2) + 1 \quad (9)$$

with  $f_{\partial\Omega} = (g_1, g_2)$ . Since the pressure force in this system acts normally on the edge of the domain, the external force could not be easily generalized in MATLAB. Each border was examined separately and the pressure force was applied more or less with *brute force*. Additionally, the mesh edge data needed to be extracted from the FreeFem++ .msh file deliberately. Please see the attached files.

Lastly, the nodal displacement can be solved by solving the linear system of equations like in Exercise 1. The stresses and the von Mises stresses can be obtained from the displacement solution via

$$\begin{bmatrix} \sigma_1^e \\ \sigma_2^e \\ \sigma_{12}^e \end{bmatrix} = [D][B^e](d^e) \quad (10)$$

and

$$\sigma_v = \sqrt{\sigma_1^2 - \sigma_1\sigma_2 + \sigma_2^2 + 3\sigma_{12}^2} \quad (11)$$

## 2.1 Number of elements needed to ensure convergence in stress

A convergence of stress was sought. After trial and error, it was determined that about 33053 elements are needed for the maximum von Mises stress value to converge. Interestingly, increasing the number of elements causes the maximum von Mises stress value to fluctuate semi-randomly, but the increased computational time for each cycle prevented further analysis to be performed. Please see Table 1 for details.

## 2.2 Highest von Mises stress

The highest von Mises stress is located at the bottom left corner of the system, as expected. Please see Figure 3. The point is highlighted on the figure. On the FreeFem++ result it is located at the same place. Please see Figure 2.

Number of Elements	FreeFem++ $\sigma_v$	MATLAB $\sigma_v$	FreeFem++ $\nu$	MATLAB $\nu$
220	$4.63664 \times 10^8$	$4.7143 \times 10^8$	-0.0464092	-0.0463
1393	$7.66613 \times 10^8$	$7.7227 \times 10^8$	-0.0505296	-0.0506
5592	$1.10681 \times 10^9$	$1.1077 \times 10^9$	-0.0516523	-0.0517
12541	$1.43700 \times 10^9$	$1.4367 \times 10^9$	-0.0520871	-0.0521
15018	$1.49578 \times 10^9$	$1.5015 \times 10^9$	-0.0521518	-0.0522
22486	$1.71684 \times 10^9$	$1.7169 \times 10^9$	-0.0521216	-0.0521
33053	$1.82630 \times 10^9$	$1.8262 \times 10^9$	-0.0523231	-0.0523

Table 1: Relationship between number of elements used and resulting stresses

Von Mises stress on deformed configuration

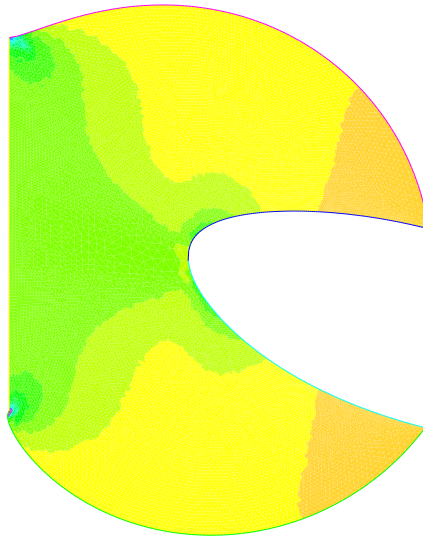


Figure 2: Converged von Mises stress field superimposed over deformation visualization from FreeFem++ for 33053 elements

### 2.3 Largest displacement

The largest displacement is located at the bottom right tip of the system, as expected. Please see Figure 3. The point is highlighted on the figure. On the FreeFem++ result it is located at the same place. Please see Figure 2.

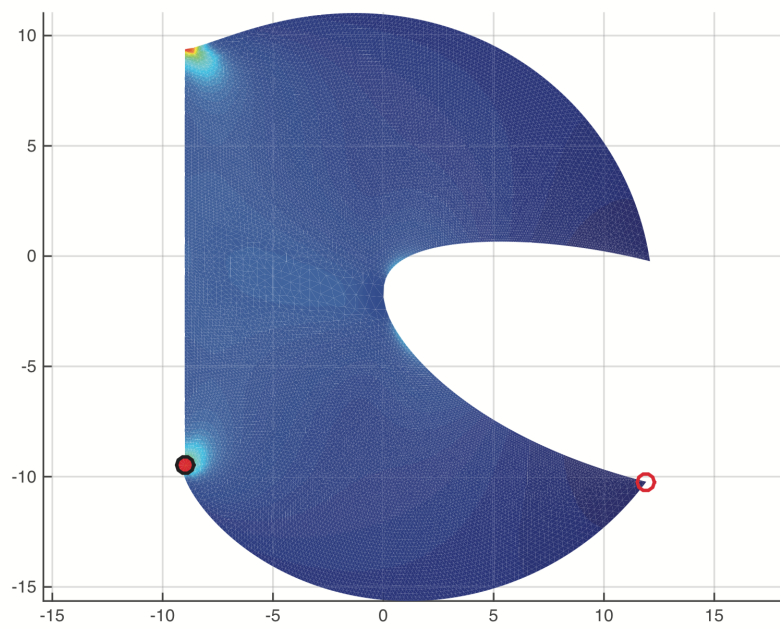


Figure 3: Converged von Mises stress field superimposed over deformation visualization from MATLAB for 33053 elements

## References

- [Koko, 2007] Koko, J. (2007). Vectorized matlab codes for linear two-dimensional elasticity. *Scientific Programming*, 15(3):157–172.