

Bending of a beam fixed on two sides

Marcin Górecki

June 15, 2017

Abstract

Aim of this project is to explore OpenFOAM capabilities in stress analysis. Results from OpenFOAM was checked with comercial program and analytics results to verified.

1 Introduction

OpenFOAM - open source Field Operation And Manipulation - is a C++ toolbox for the development of customized numerical solvers. It is utilize to pre-/post-processing for the solution of continuum mechanics problems, most often for computational fluid dynamics (CFD).

2 Bending analysis

2.1 Problem definition

Subject of analysis is beam fixed from both sides. Beam is bending by a constant pressure along the length. Please see Figure 1.

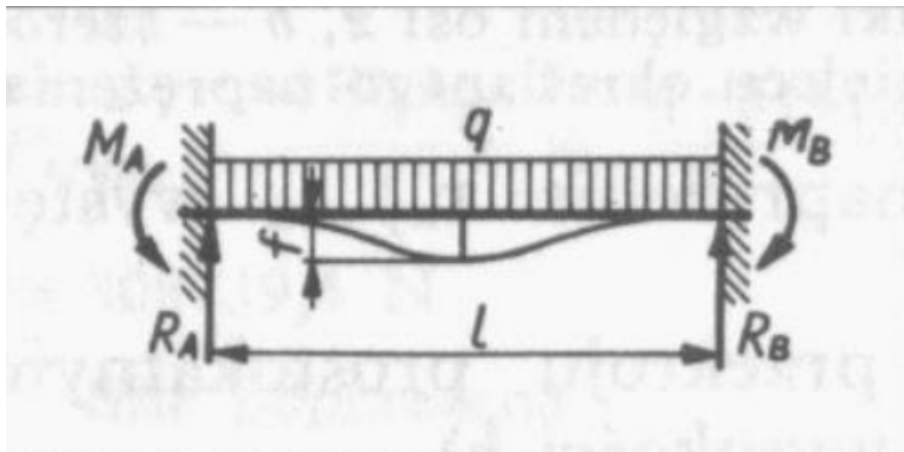


Figure 1: Problem definition.

Dimensions:

$l = 10$ [m]; $a = 1$ [m], when a is a side of a beam cross section

Loads:

$q = 0,1$ [MN/m] or in 3D case $p = 1$ [MPa];

2.2 OpenFOAM results

2.2.1 Mesh generation

Prismatic beam is really simple shape, so only one block was used. To generate mesh paste code bellow in file BlockMeshDict and then use comand *blockMesh*.

```

1  /*-----* C++ *-----*\
2  / ===== /
3  / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4  / \ \ / O p e r a t i o n / Version: 4.1 /
5  / \ \ / A n d / Web: www.OpenFOAM.org /
6  / \ \ / M a n i p u l a t i o n /
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        blockMeshDict;
14 }
15 // *****
16
17 convertToMeters 1;
18
19 vertices
20 (
21     (0 0 0)
22     (10 0 0)
23     (0 1 0)
24     (10 1 0)
25     (0 0 1)
26     (10 0 1)
27     (0 1 1)
28     (10 1 1)
29 );
30
31 blocks
32 (
33     hex (0 1 3 2 4 5 7 6) (50 5 5) simpleGrading (1 1 1)
34 );
35
36 boundary
37 (
38     left
39     {
40         type patch;
41         faces
42         (
43             (0 2 6 4)
44         );
45     }
46     right
47     {
48         type patch;
49         faces
50         (
51             (1 3 7 5)
52         );
53     }
54     down
55     {
56         type patch;
57         faces
58         (
59             (0 1 5 4)
60         );
61     }
62     up
63     {
64         type patch;
65         faces
66         (
67             (2 3 7 6)
68         );
69     }
70     frontAndBack
71     {
72         type empty;
73         faces

```

```

74 |         (
75 |         (0 1 3 2)
76 |         (4 5 7 6)
77 |         );
78 |     }
79 | );
80 |
81 | mergePatchPairs
82 | (
83 | );
84 |
85 | // *****

```

In *vertices* section we have specified corners positions. In *blocks* section beam and division to finite elements were defined. In *boundary* faces were defined. Mesh can be viewed in ParaView, using command *paraFoam* &

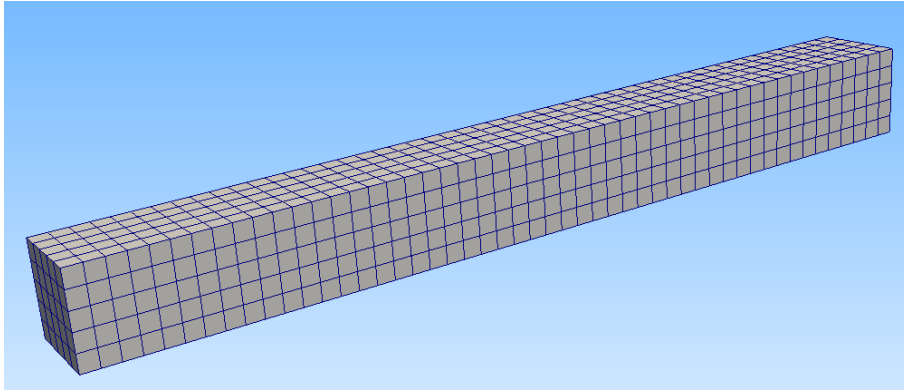


Figure 2: Mesh viewed in ParaView.

2.2.2 Boundary and initial conditions

For a stress analysis case without thermal stresses, only displacement D needs to be set. The 0/D is as follows:

```

1 | /*----- C++ -----*/
2 | / ===== /
3 | / \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4 | / \ / O p e r a t i o n / Version: 4.1 /
5 | / \ / A n d / Web: www.OpenFOAM.org /
6 | / \ / M a n i p u l a t i o n / /
7 | \*-----*/
8 | FoamFile
9 | {
10 |     version      2.0;
11 |     format        ascii;
12 |     class         volVectorField;
13 |     object        D;
14 | }
15 | // *****
16 |
17 | dimensions      [0 1 0 0 0 0 0];
18 |
19 | internalField    uniform (0 0 0);
20 |
21 | boundaryField
22 | {
23 |     left
24 |     {
25 |         type      fixedValue;
26 |         value      uniform (0 0 0);
27 |     }
28 |     up
29 |     {
30 |         type      tractionDisplacement;
31 |         traction   uniform (0 -1000000 0);

```

```

32         pressure      uniform 0;
33         value          uniform (0 0 0);
34     }
35     down
36     {
37         type            tractionDisplacement;
38         traction        uniform (0 0 0);
39         pressure        uniform 0;
40         value           uniform (0 0 0);
41     }
42     right
43     {
44         type            fixedValue;
45         value           uniform (0 0 0);
46     }
47     frontAndBack
48     {
49         type            empty;
50     }
51 }
52
53 // *****

```

In this case on left and right side (patches) we have fixed support handled by *fixedValue* condition. Patch *frontAndBack* is declared as empty, because don't participate in 2D bounding. Patches up and down are using *tractionDisplacement* condition and to face up 1 MPa pressure is applied.

2.2.3 Mechanical properties

The physical properties for the case are set in the *mechanicalProperties* dictionary in the constant directory. Code:

```

1  /*-----* C++ *-----*/
2  / ===== /
3  / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4  / \ \ / O p e r a t i o n / Version: 4.1 /
5  / \ \ / A n d / Web: www.OpenFOAM.org /
6  / \ \ / M a n i p u l a t i o n / /
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object        mechanicalProperties;
15 }
16 // *****
17
18 rho
19 {
20     type          uniform;
21     value         7850;
22 }
23
24 nu
25 {
26     type          uniform;
27     value         0.3;
28 }
29
30 E
31 {
32     type          uniform;
33     value         2e+11;
34 }
35
36 planeStress     yes;
37
38
39 // *****

```

2.2.4 Thermal properties

In this case we do not want to solve for the thermal equation. Therefore we must set the *thermalStress* keyword entry to no in the *thermalProperties* dictionary.

```
1  /*----- C++ -----*/
2  / ===== /
3  / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4  / \ \ / O p e r a t i o n / Version: 4.1 /
5  / \ \ / A n d / Web: www.OpenFOAM.org /
6  / \ \ / M a n i p u l a t i o n / /
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object        thermalProperties;
15 }
16 // ***** //
17
18 C
19 {
20     type          uniform;
21     value         434;
22 }
23
24 k
25 {
26     type          uniform;
27     value         60.5;
28 }
29
30 alpha
31 {
32     type          uniform;
33     value         1.1e-05;
34 }
35
36 thermalStress    no;
37
38
39 // ***** //
```

2.2.5 Control

I take this part from OpenFoam user guide. Code:

```
1  /*----- C++ -----*/
2  / ===== /
3  / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4  / \ \ / O p e r a t i o n / Version: 4.1 /
5  / \ \ / A n d / Web: www.OpenFOAM.org /
6  / \ \ / M a n i p u l a t i o n / /
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        controlDict;
15 }
16 // ***** //
17
18 application      solidDisplacementFoam;
19
20 startFrom        startTime;
21
22 startTime        0;
23
24 stopAt           endTime;
```

```

25
26 endTime          100;
27
28 deltaT            1;
29
30 writeControl       timeStep;
31
32 writeInterval      20;
33
34 purgeWrite         0;
35
36 writeFormat        ascii;
37
38 writePrecision     6;
39
40 writeCompression   off;
41
42 timeFormat         general;
43
44 timePrecision      6;
45
46 graphFormat        raw;
47
48 runTimeModifiable true;
49
50
51 // *****

```

2.2.6 Discretisation schemes and linear-solver control

I take this part from OpenFoam user guide. *fvSchemes* code:

```

1  /*-----* C++ *-----*/
2  / ===== /
3  / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4  / \ \ / O p e r a t i o n / Version: 4.1 /
5  / \ \ / A n d / Web: www.OpenFOAM.org /
6  / \ \ / M a n i p u l a t i o n / /
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        fvSchemes;
15 }
16 // *****
17
18 d2dt2Schemes
19 {
20     default       steadyState;
21 }
22
23 ddtSchemes
24 {
25     default       Euler;
26 }
27
28 gradSchemes
29 {
30     default       leastSquares;
31     grad(D)       leastSquares;
32     grad(T)       leastSquares;
33 }
34
35 divSchemes
36 {
37     default       none;
38     div(sigmaD)   Gauss linear;
39 }
40
41 laplacianSchemes

```

```

42 | {
43 |     default          none;
44 |     laplacian(DD,D) Gauss linear corrected;
45 |     laplacian(DT,T) Gauss linear corrected;
46 | }
47 |
48 | interpolationSchemes
49 | {
50 |     default          linear;
51 | }
52 |
53 | snGradSchemes
54 | {
55 |     default          none;
56 | }
57 |
58 | // *****

```

fvSolution code:

```

1 | /*----- C++ -----*/
2 | / ===== /
3 | / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4 | / \ \ / O p e r a t i o n / Version: 4.1 /
5 | / \ \ / A n d / Web: www.OpenFOAM.org /
6 | / \ \ / M a n i p u l a t i o n / /
7 | /*-----*/
8 | FoamFile
9 | {
10 |     version          2.0;
11 |     format            ascii;
12 |     class             dictionary;
13 |     location          "system";
14 |     object            fvSolution;
15 | }
16 | // *****
17 |
18 | solvers
19 | {
20 |     "(D|T)"
21 |     {
22 |         solver          GAMG;
23 |         tolerance       1e-08;
24 |         relTol          0.9;
25 |         smoother        GaussSeidel;
26 |         nCellsInCoarsestLevel 20;
27 |     }
28 | }
29 |
30 | stressAnalysis
31 | {
32 |     compactNormalStress yes;
33 |     nCorrectors        2;
34 |     D                  1e-10;
35 | }
36 |
37 |
38 | // *****

```

Here is one change in comparison to file from use guide. I have changed *nCorrectors* value from 1 to 2. It have positive influence on deformation result.

2.2.7 Running the code

To solve the case following lines should be executed in project directory:

solidDisplacementFoam > log &.

2.2.8 Post-processing

For post-processing individual scalar field components, Sigma(xx), Sigma(xy) etc. , can be generated by running the postProcess:

postProcess -func "components(sigma)"

2.2.9 Results

Results are present by paraView. Deformation of beam can be shown by using *Wrap by vector* command.

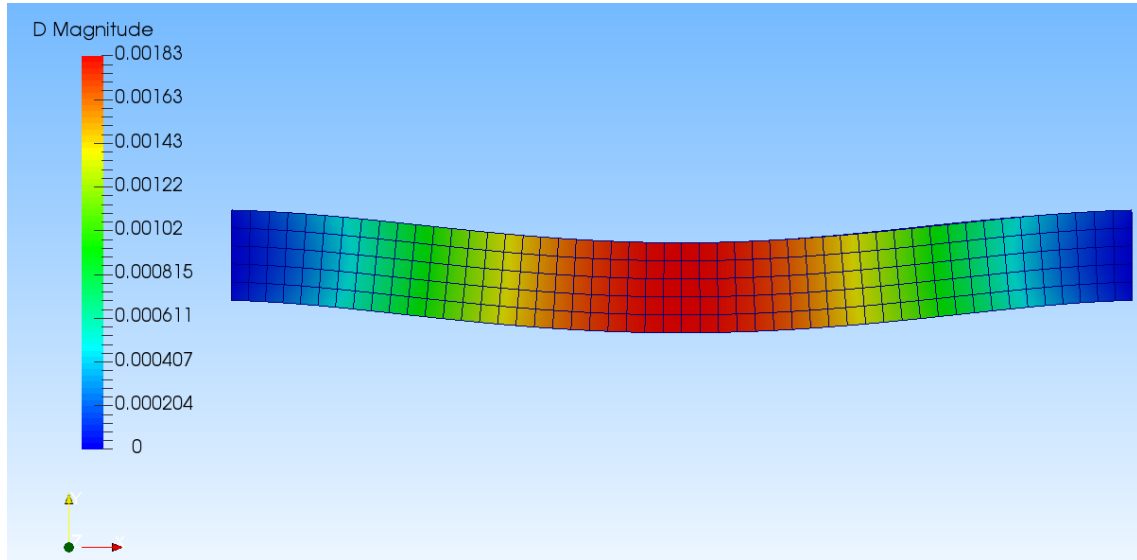


Figure 3: Deformation of a beam.

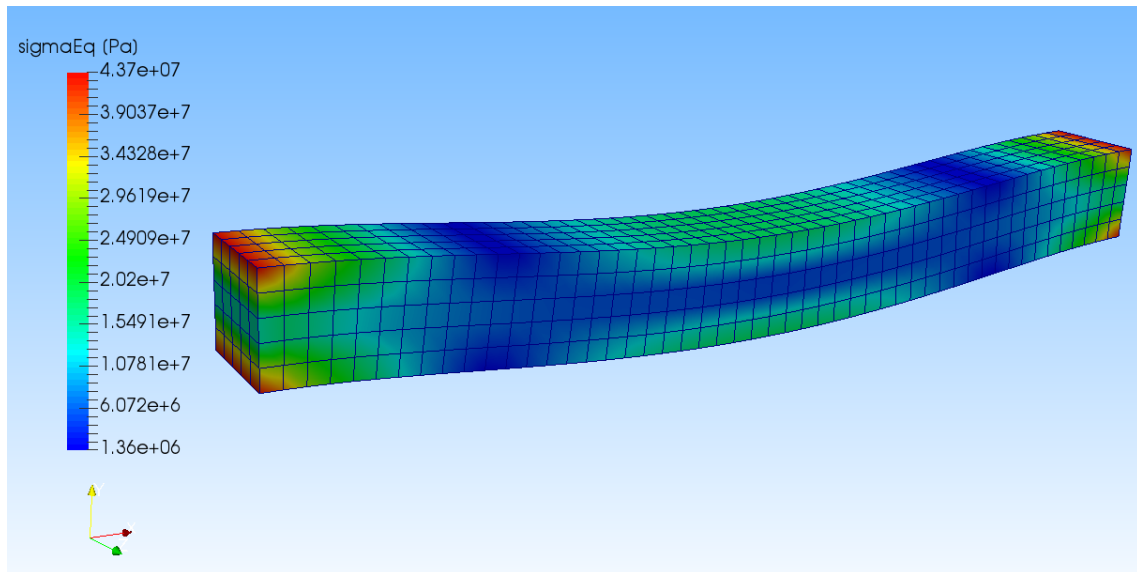


Figure 4: Equivalent stress.

2.3 Results from Ansys

Ansys in version 16.0 was used.

See load and supports on [Figure 5](#)

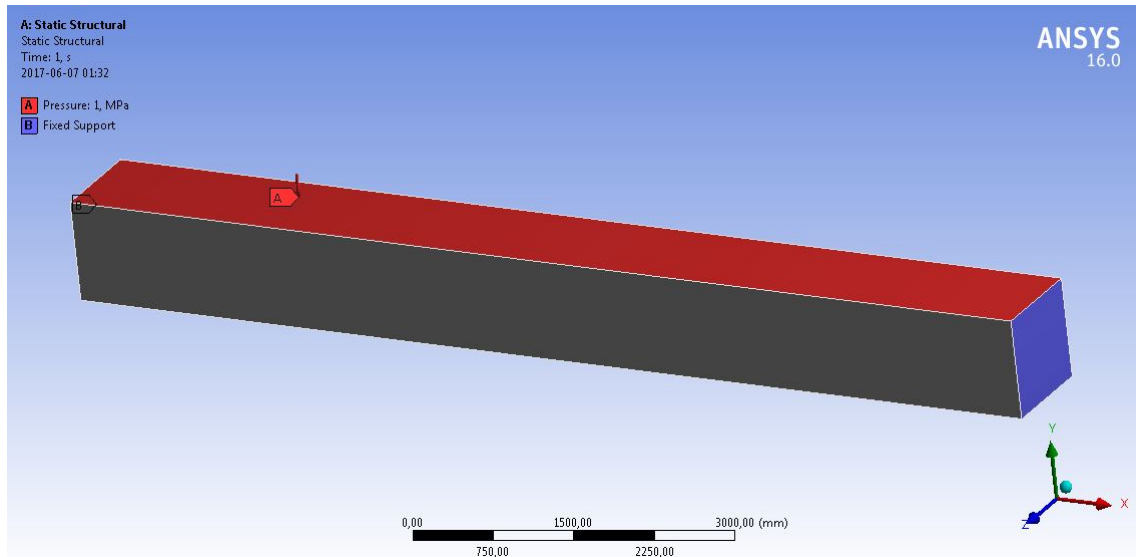


Figure 5: load and supports.

Mesh of beam is shown on Figure 6

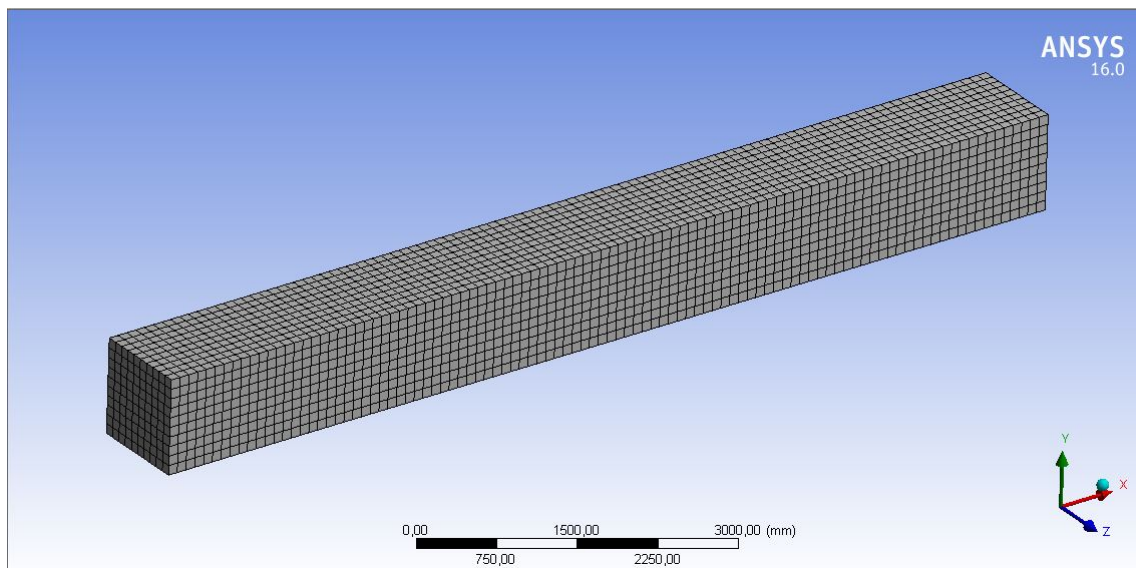


Figure 6: Division into finite elements.

Results:

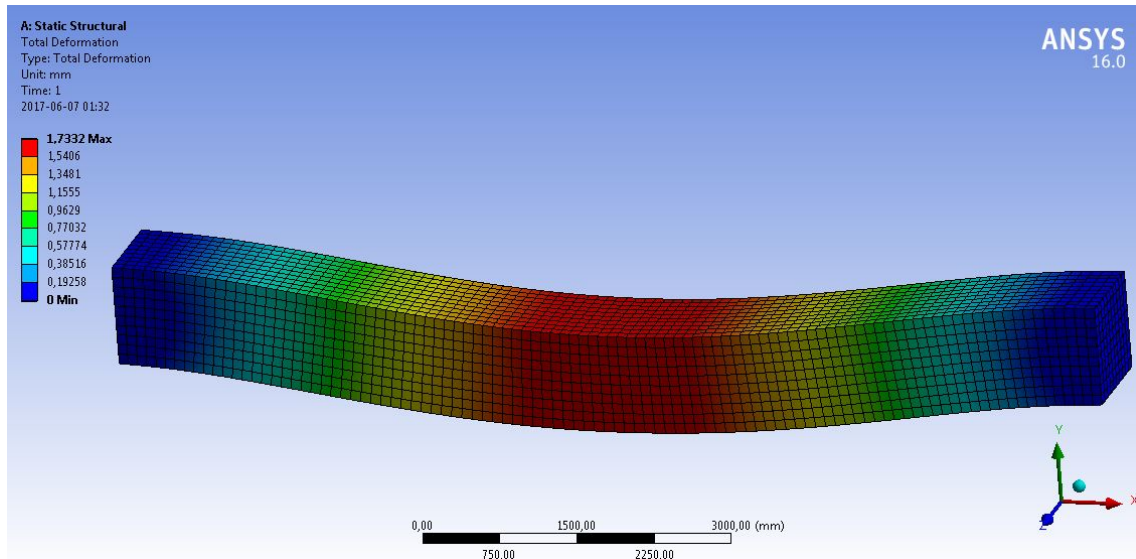


Figure 7: Deformation of a beam.

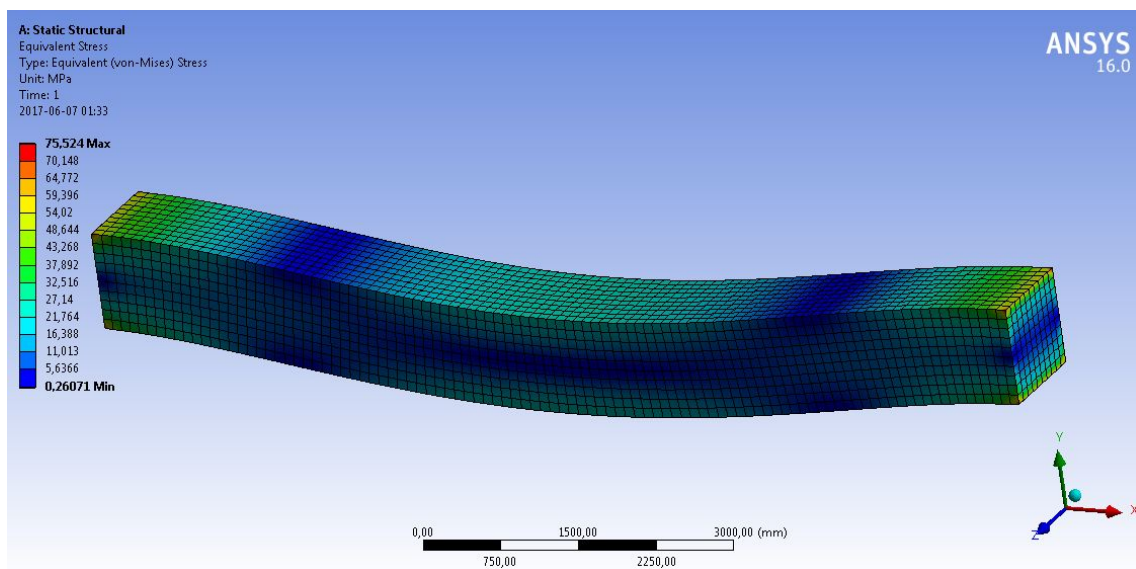


Figure 8: Equivalent stress.

3 Conclusions

Differences between results in OpenFOAM and commercial program are really small. OpenFOAM can be used to stress analysis, but each case have to be checked with lab test or anlitic calculations.