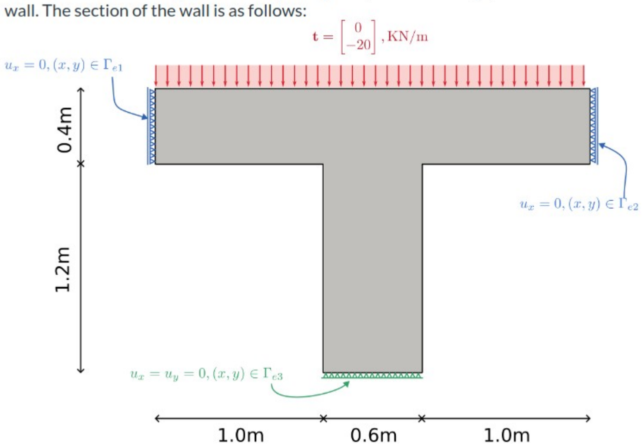
Midterm # 1 2023 Solution

Midterm Exam # 1

Instructions

This exam has a total of 220pt, with practical and theoretical questions. The conversion from points to final grade will be relative to the performance of the class, so solve as much as you can in the available time.

In the design of a critical-infrastructure building it is important to check design assumptions with high-precision modeling. To assess whether the steel reinforcement detailing was correctly setup in a particular design, we wish to model a wall-slab interface to check the transfer of shear forces from the slab to the wall. The section of the wall is as follows:



You can assume that the structure has infinite out-of-plane width and is made out of concrete with a Young's modulus of 35 GPA, a Poisson's ratio of 0.2 and a mass density of 2500 kg/m³. The detailed modeling of the intersection can be reduced by assuming the shown boundary conditions.

You will be modeling this situation using what you've learned in previous homeworks and reading. You can use the supplied FE implementation provided in-class.

1 File Upload 10 points Modeling the geometry

Model the situation using gmsh. Provide physical groups for all relevant components (two different types of essential BCs and one natural BC, as well as the concrete). Upload the .geo file.

```

SetFactory("OpenCASCADE");
Rectangle(1) = {-0.3, 0., 0, 0.6, 1.2, 0};
Rectangle(2) = {-1.3, 1.2, 0, 2.6, 0.4, 0};
Coherence;
Physical Curve("BC_e_12", 10) = {9, 7};
Physical Curve("BC_e_3", 11) = {1};
Physical Curve("BC_n", 12) = {8};
Physical Surface("Concrete", 13) = {2, 1};
```

Download link: https://www.dropbox.com/s/0gw53mihlgiyph3/midterm1.geo?dl=1

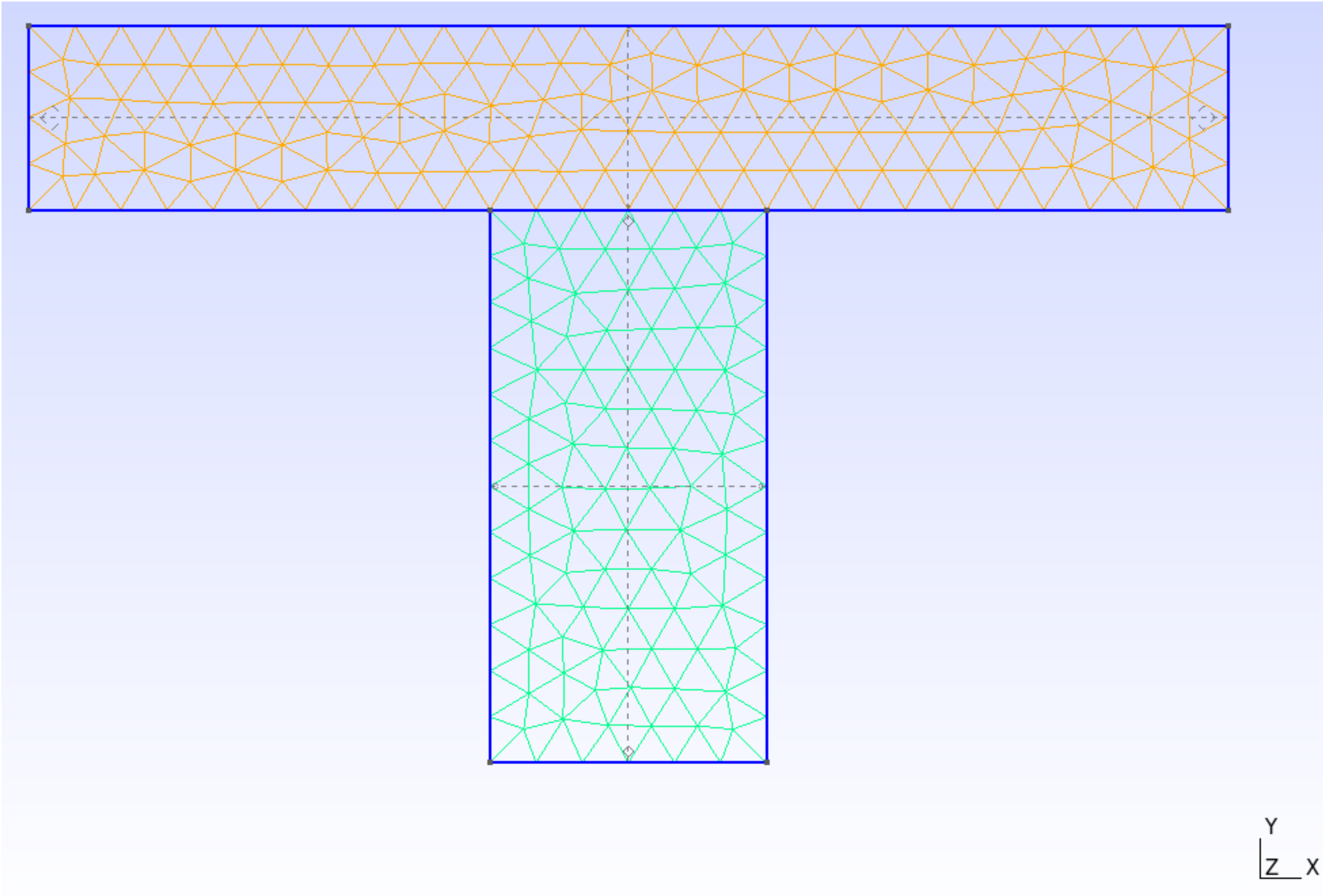
2 File Upload 5 points Create a mesh file for the problem

Export a triangular mesh file for the problem. Approximately aim to have elements of size 0.1m.

Download link: https://www.dropbox.com/s/x9d8le8cdu7fp4i/midterm1.msh?dl=1

3 File Upload 5 points Image of triangular mesh

Upload an image of the mesh you created.



4 Essay 10 points How can you know that a mesh is appropriate?

Explain: how can you know if you've chosen correctly the element sizes for a given problem? Answer in spanish or english.

The solution in terms of displacements and/or stresses (depending on quantity of interest) should converge with refinement of the mesh. To check if the mesh size is correct, build several meshes (coarse, medium and fine), choose the coarsest one that shows stabilized results. That is, the one with the biggest elements upon which a refinement would not yield significantly better results to warrant the increase in computational cost.

5 File Upload 30 points Assembly of stiffness equations

Upload a python file that only assembles the element stiffness equations. You do not need to apply boundary conditions or solve the problem in this file. The output of this file should be the maximum and minimum diagonal values of the stiffness matrix as well as the maximum and minimum global values of the stiffness matrix. The script should do nothing else other than reporting these values on the screen.

Download link: https://www.dropbox.com/s/d9vlmdl94i7woyd/midterm1_problem5_sol.py?dl=1

6 Essay 5 points Maximum/minimum stiffness values

Paste the screen output of your previous question python file here.

```
Info    : Reading 'midterm1.msh'...
Info    : 19 entities
Info    : 264 nodes
Info    : 482 elements
Info    : Done reading 'midterm1.msh'
Max k diagonal : 108376420.54427652
Max k diagonal : 21177722.915684383
Max k global   : 108376420.54427652
Max k global   : -43627908.137522705
[Finished in 217ms]
```

7 Essay 5 points Code explanation

In the python script provided as template for homework solution (here: https://www.dropbox.com/s/l9zqhhoq4s0vt2z/HW2-for-midterm.zip?dl=1), explain briefly what this part of the code does:

```
65  properties = {}
66  properties["b"] = b
67  properties["E"] = E_σ
68  properties["t"] = 3*_mm
69
70  eleTags, nodeTags, element_name, element_nnodes = get_elements_and_nodes_in_physical_group("
        thickness_3mm", gmsh.model)
71
72  for tag, nodes in zip(eleTags, nodeTags):
73      print(f"{tag=} {nodes=}")
74      xy_e = xyz[np.array(nodes,dtype=int)-1,:]
75
76      ke, fe = cst(xy_e, properties)
77
78      dofs = [2*(nodes[0]-1), 2*(nodes[0]-1)+1,
79      2*(nodes[1]-1), 2*(nodes[1]-1)+1,
80      2*(nodes[2]-1), 2*(nodes[2]-1)+1]
81
82
83      for i in range(6):
84          p = dofs[i]
85          for j in range(6):
86              q = dofs[j]
87              K[p,q] += ke[i,j]
88          f[p] += fe[i]
```

You can answer in english or spanish.

That part of the code is in charge of doing stiffness assembly. In line 70 we get the elements numbers and nodes that belong to the physical group “thickness_3mm”. Then at line 72 forward we iterate over element tags and nodes. Line 74 extracts the nodal coordinates for the current element nodes, then line 76 builds the stiffness matrix for the current element. Line 78 has the DOF mapping into the global DOF numbering. Then lines 83-88 are the stiffness and force vector assembly.

8 Essay 10 points Code explanation

In the python script provided as template for homework solution (here: https://www.dropbox.com/s/l9zqhhoq4s0vt2z/HW2-for-midterm.zip?dl=1), explain briefly what this part of the code does:

```
142 eleTags, nodeTags, element_name, element_nnodes = get_elements_and_nodes_in_physical_group("
        BC_essential", gmsh.model)
143
144 fixed_nodes = []
145
146 for n in nodeTags:
147     print(n)
148     fixed_nodes += n
149
150 fixed_nodes = np.unique(fixed_nodes)
151
152 fixed_dofs = []
153 for n in fixed_nodes:
154     fixed_dofs.append(2*(n-1))
155     fixed_dofs.append(2*(n-1)+1)
156
157 fixed_dofs = np.array(fixed_dofs, dtype=np.int32)
158 free_dofs = np.arange(NDOFS, dtype=np.int32)
159 free_dofs = np.setdiff1d(free_dofs, fixed_dofs)
```

You can answer in english or spanish.

This section of the code is in charge of building a list of DOF numbers that are fixed and free to do the partitioning. In line 142 the element number and nodes (connectivity) belonging to physical group “BC_essential” are obtained. The nodes of each element are placed in a list “fixed_nodes” in lines 146-148. At line 150 the list is made unique, that is, duplicate node numbers are removed. For each fixed node, another list with the DOF numbers is built. Both x and y displacements are fixed hence for each node number, two dofs are added to the list.

Line 157 converts that list into a python array. Lines 158 and 159 build a list of “free_dofs” with unconstrained DOFS by subtracting the fixed DOFS from the list of all dofs.

9 File Upload 60 points Analysis of the model

Upload a python script that analyzes the situation at hand. The script should provide comments clearly indicating where

- Essential and boundary conditions are applied
- Self weight is applied
- Element properties are specified
- Stiffness assembly is performed
- System of equations partitioning is performed
- Solution is performed
- Stresses are post-processed

The script should output to the console the following information

- Vertical displacement at the right side
- Vertical displacement at the left side
- Maximum and minimum global vertical displacements
- Maximum and minimum components of the stress tensor

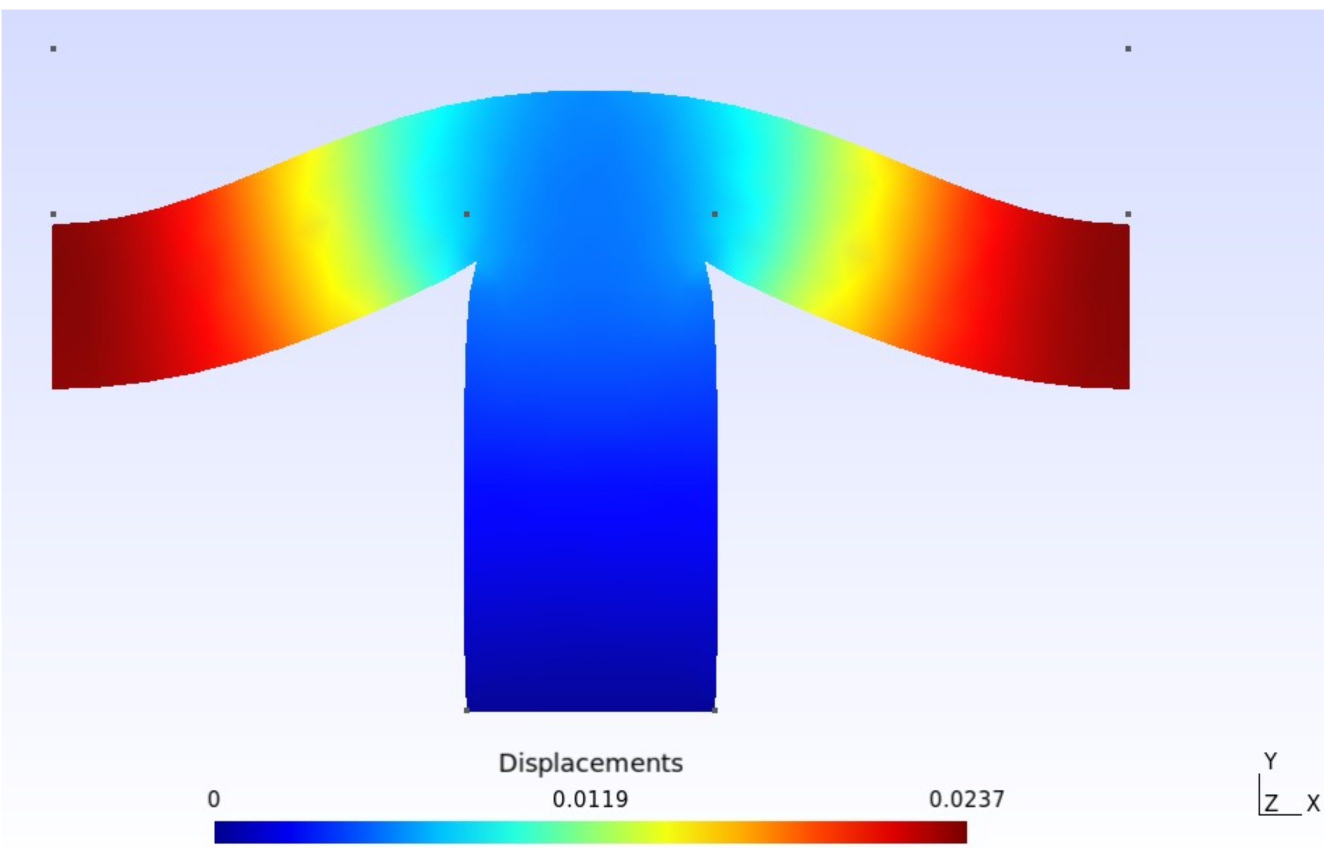
Additionally, the following gmsh post processing views should be generated:

- Displaced shape of the mesh
- Shear stress field

Download link: https://www.dropbox.com/s/zakjxs3vp2jh7ye/midterm1_problem9_sol.py?dl=1

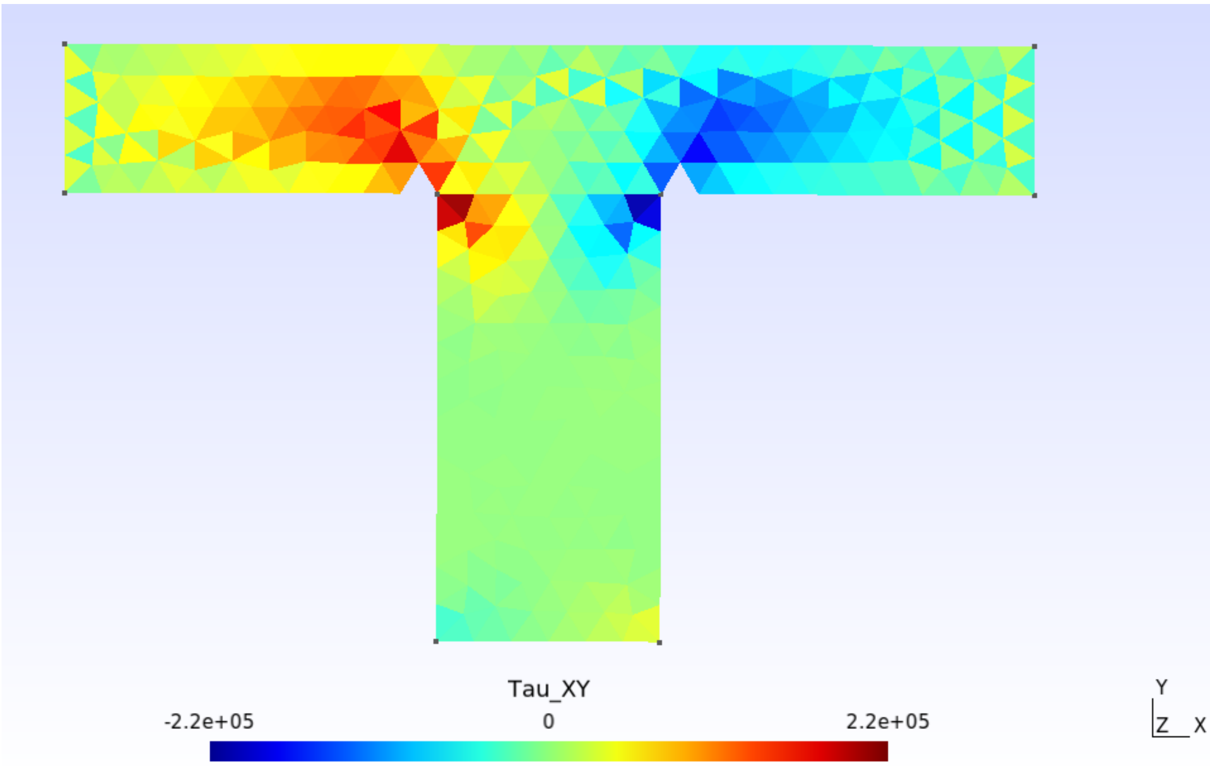
10 File Upload 5 points Displaced shape

Upload an image of the displaced shape of the problem. Choose an appropriate scaling factor so that the displacementents might be properly observed.



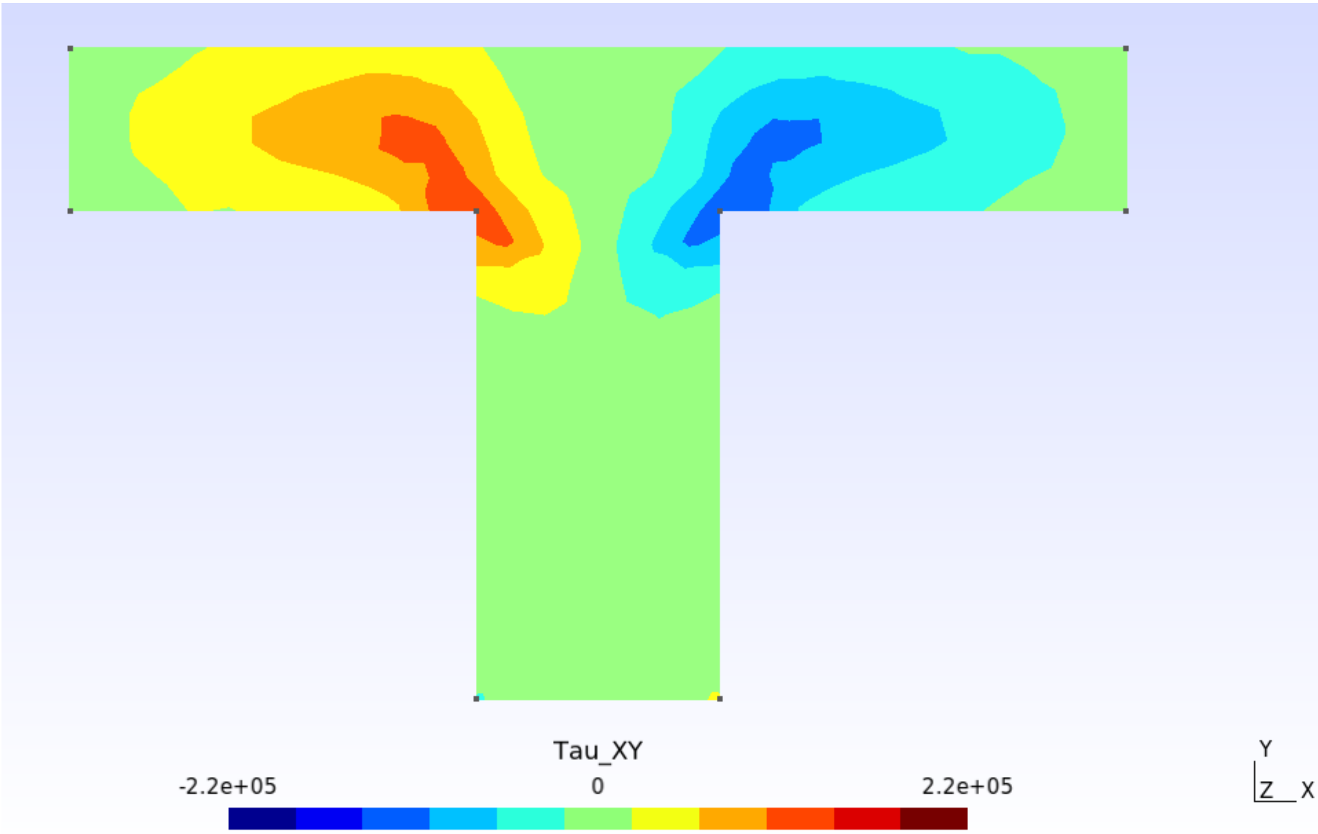
11 File Upload 5 points Shear stress field

Upload an image of the shear stress field. The field should not be smoothed (no inter-element averaging applied) and the color scale should be chosen to be symmetrical such that the color corresponding to zero stress occurs at the middle of the colorbar (green color by default).



12 File Upload 5 points Smoothed shear stress field

Upload an image of the shear stress field with smoothing (nodal averaging) applied. The color scale should be the same as before, except this time use filled iso-values with an odd-number of intervals to visualize the stress gradients.



13 Essay 5 points Solution symmetry?

Are your results (displacements, stresses) symmetric? Why?

The displacements at the left and right hand sides are not equal:

```
uy_left=-0.02994620494786868
uy_right=-0.029932071756003997
```

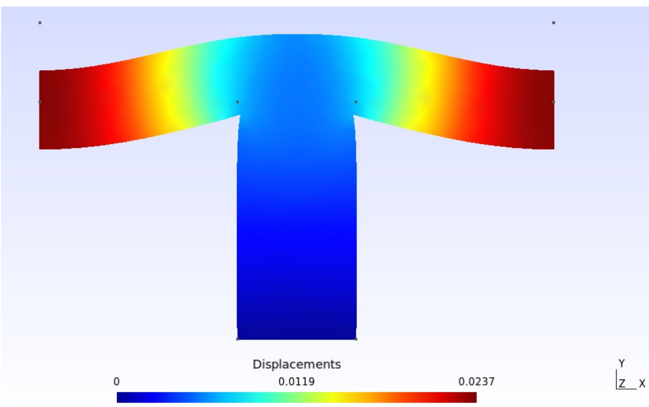
therefore the results are not symmetric, despite the problem being symmetric. The only source of asymmetry in this case is the mesh, which is not symmetric. It is expected that with a symmetric mesh, results would be symmetric up to the round-off error.

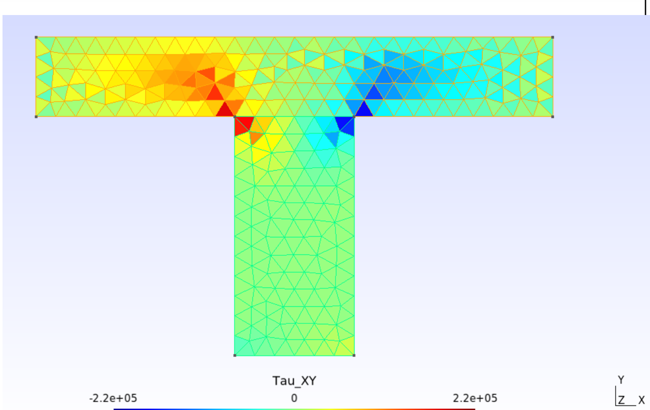
14 Essay 20 points Influence of self-weight on results

In this question write about the influence that considering self-weight has on your results. How sensitive are the results to the self weight of the structure? Is self-weight more relevant than the applied distributed load at the top of the slab? Explain what you had to do to answer these questions. Provide images showing your results as well as quantitative data (values of shear stresses, displacements, etc.) (you can use the rich-text editor in canvas to upload the images within this question). Answer in spanish or english.

<table>
<tr><th>Results without self-weight</th><th>Results with self-weight</th></tr>
<tr><td>

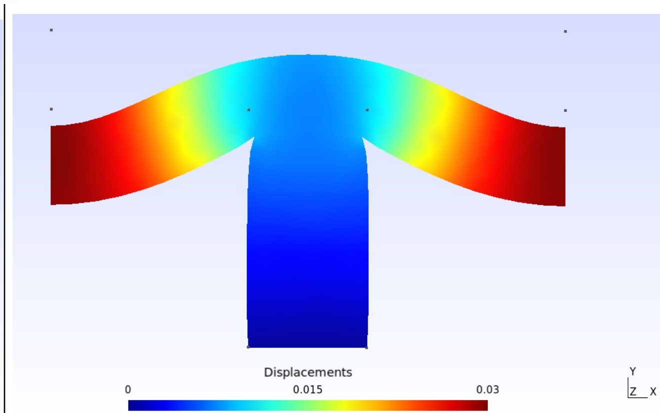
```
uy_left=-0.023707146507980833
uy_right=-0.023695376338765613
global_uy_max=0.0041116315082323045
global_uy_min=-0.023745396615961693
sigma_xx.max() = 333778.6081963076
sigma_xx.min() = -469264.79424786265
sigma_yy.max() = 10496.869060925208
sigma_yy.min() = -451437.6267204726
tau_xy.max() = 178682.4713148908
tau_xy.min() = -179799.5710739153
```

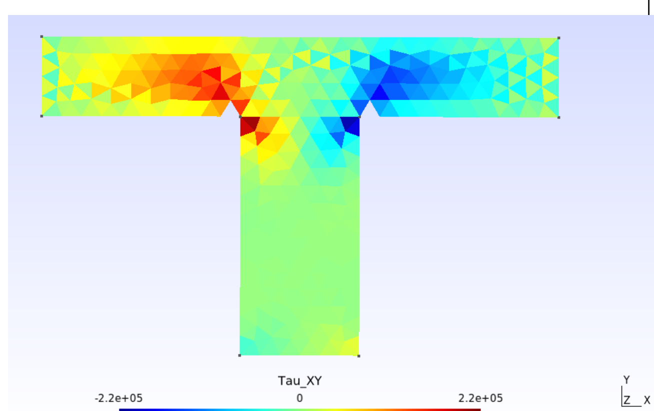




</td><td>

```
uy_left=-0.02994620494786868
uy_right=-0.029932071756003997
global_uy_max=0.005119443831799953
global_uy_min=-0.030022626463503687
sigma_xx.max() = 417654.2689936936
sigma_xx.min() = -584229.1184402688
sigma_yy.max() = 13992.992487775293
sigma_yy.min() = -563945.1985085158
tau_xy.max() = 222582.78252221076
tau_xy.min() = -223984.5137669389
```





</td></tr>
</table>

To obtain these results, just set b = 0 for the case without self weight.

It can be seen that the maximum shear stress without self weight is 178682 N/m² whereas with self weight it is 222582 N/m². This is a 25% increase in shear stress if self-weight is considered. In terms of maximum displacements it is -2.37cm vs. 2.99cm, which corresponds to a 26% difference.

Therefore self-weight in this case is not negligible.

15 Essay 20 points Model assmumptions

List as many modeling assumptions made in this model as you can think of. In view of these assumptions criticize the model. What improvements are possible and how relevant might they be? Answer in spanish or english.

Some assumptions (more are possible)

- Plane-strain condition (2D model)
- Isotropic-elastic material
- Linear stress-strain relationship
- Un-cracked concrete
- Linear strain-displacement relationship (small deformations and rotations)
- Perfectly rigid boundary conditions
- Not modeling any reinforcement.
- Assumed concrete material properties (as opposed to measured)
- Use of the CST element
- Uniform gravity field
- Converged mesh (discretization)

Improvements

- Use a 3D model
- Model the rest of the structure
- Consider flexible boundary conditions (soil or the rest of the structure)
- Use better elements (LST, etc.)
- Consider using cracked concrete properties or a non-linear stress-strain relationship
- Improve the mesh
- Model reinforcement explicitly

16 Essay 5 points Plane-stress vs plane-strain

What is the difference between a plane-stress and plane-strain assumption in 2-D modeling of solids? Give examples of situations that might be modeled in plain-stress and others in plain strain. Answer in spanish or english.

In plane stress we assume that the out-of-plane stress (σ_z) is zero, hence out-of-plane strains are different from zero due to Poisson's effect. Conversely, in plane-strain conditions the out-of-plane strain (ε_z) is zero and the out-of-plane stress is different from zero.

A thin plate subject to in-plane bending might be adequately modeled using plane stress theory.

A long dam (crest length is much larger than height/width) might be adequately modeled using plane-strain theory.

17 Essay 10 points The CST element

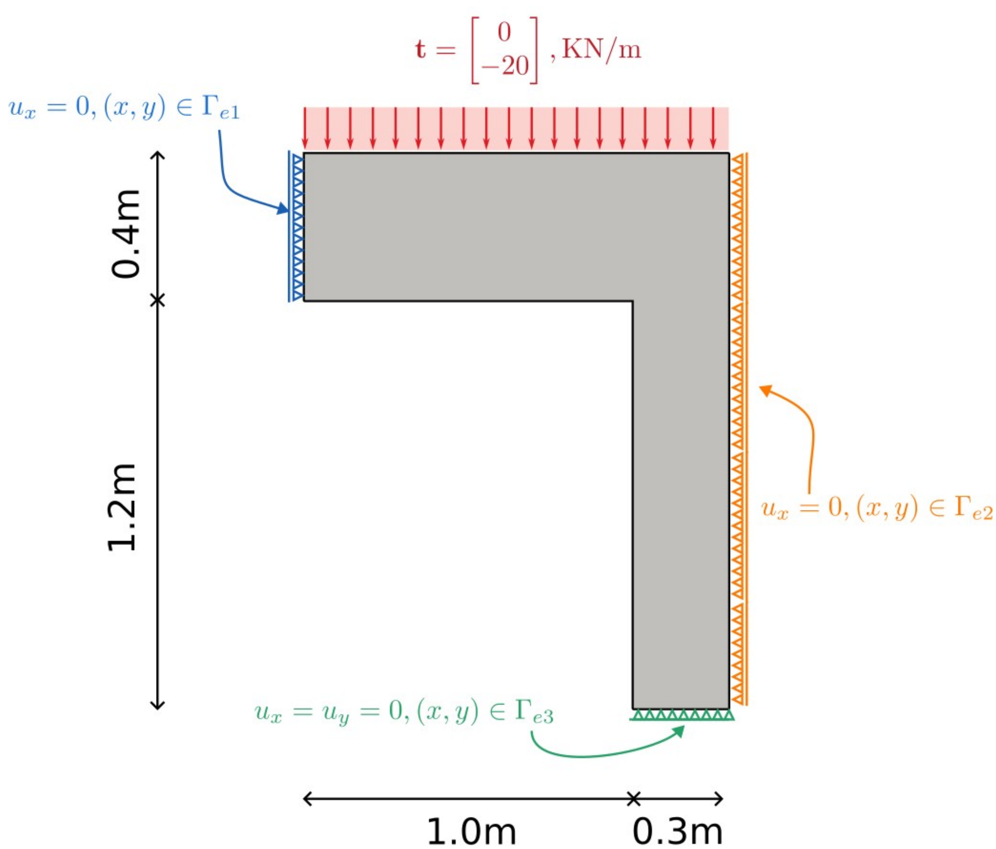
Explain what the acronym CST means? Why does this triangle element receive this name? Explain what the B matrix in the CST element formulation does and how it relates to the name of the element. In what cases might the use of the CST element be adequate or inadequate?

CST means constant-stress or constant-strain triangle. This element assumes that the strain/stress field is constant in the element interior. The strain-displacement matrix B, gives the strain field inside the element for a given nodal displacement vector ($\varepsilon = B\ u$) and it is constant inside the element (does not depend on the point inside the element).

18 Essay 5 points Exploiting symmetry

Explain how the symmetry of the proposed problem can be exploited to reduce the problem size. How does the domain change? How do the boundary conditions change? Other than reducing computational cost, are there any other advantages in using the problem symmetry?

You can divide the domain into half, like so:



This results in a reduction of number of nodes to half as before, resulting in half the memory usage. But since solution of the stiffness equations is of $O(N^3)$ then the reduction in compute time is about x8. Also, the memory use is reduced.

Other than that, the enforcement of the symmetry means that results are automatically symmetric independent of mesh. Nodes located on the midline on the complete mesh would not have a zero horizontal displacement in general, whereas in this setup it would.