Computational Structural Mechanics and Dynamics
Assignment 8 - Shells

Federico Valencia Otálvaro

Master's in Numerical Methods in Engineering
Universitat Politècnica de Catalunya

April 20^{th}, 2020

Contents

List of Figures

1 Problem Description

The problem consists of analyzing the behavior of a hyperbolic concrete shell under self-weight load. The shell has a thickness t = 0.1 and the following geometry:

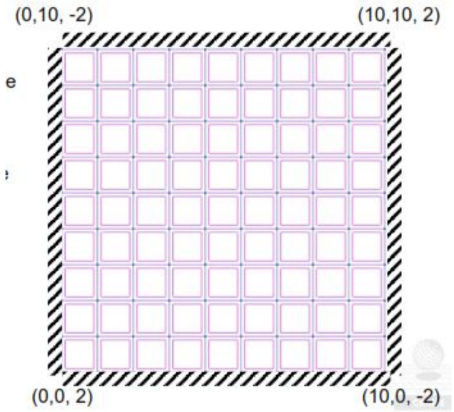


Figure 1: Problem Geometry

2 Solution

2.1 Modelling

The analysis was performed using the available MatFEM program for Reissner-Mindlin shells `Shell_T_RM_v1_1`. Since the shell is supposed to be made of concrete, the material properties used were (working with units consistent with N and m) $E = 3e10$, $\eta = 0.2$, and $\rho = 24e3$.

The structure was discretized in a mesh of 200 triangular elements and 121 nodes, all elements being isosceles triangles of sides size 1. Since the Z coordinate of the corners are known, the Z coordinates of all other nodes were interpolated using linear shape functions for a rectangular element, obtaining the following hyperbolic geometry:

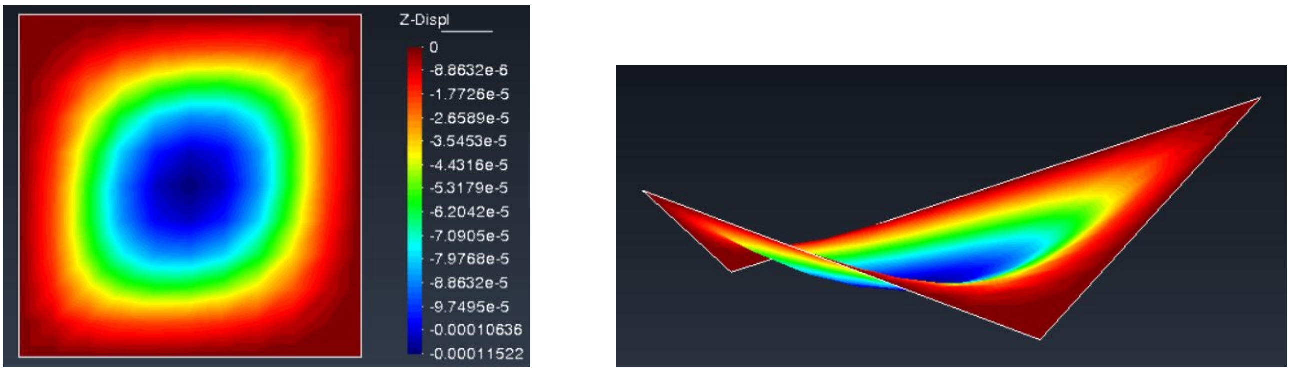
Figure 2: Model Geometry

For boundary conditions, all 5 degrees of freedom of all nodes at the boundary were restrained.

2.2 Results

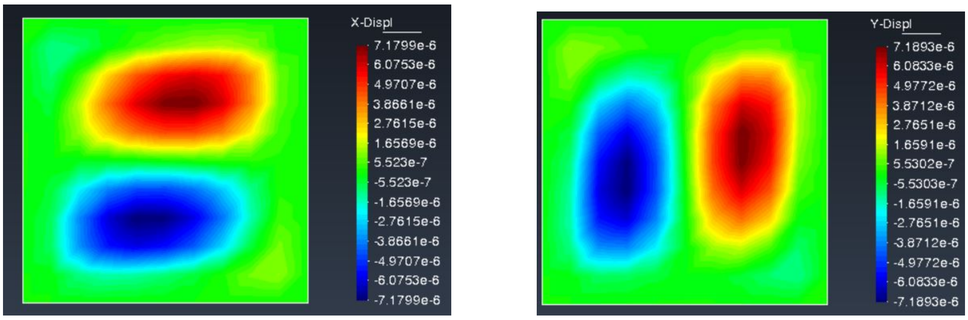
2.2.1 Displacement

The vertical displacements suffered by the shell under its self-weight are very low (largest displacement is around 0.11 mm). Due to the structure geometry, gravity loads do not act perpendicularly to the shell plane and therefore, a fraction of the load is applied as in-plane load on the membrane, improving its structural response.



(a) Z disp. (b) Deformed shape

Figure 3: Vertical Displacements

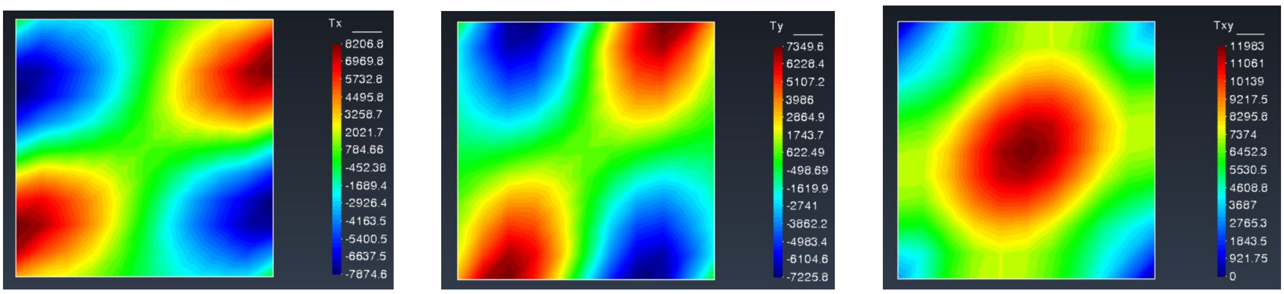


(a) X disp. (b) Y disp.

Figure 4: Horizontal Displacements

2.2.2 Membrane Stress

It may be observed how the lower points of the shell are subjected to compressive stress, while the upper parts are under traction and the axial stress at the center of the sell in both directions X and Y is close to zero. A flat shell with the same in-plane dimensions would not present compressive membrane stress and the largest traction stresses would be located at the center of the shell. We may therefore conclude that since concrete behaves better under compression, the modified geometry presents an improved structural performance with respect to a flat shell.



(a) X stress (b) Y stress (c) XY stress

Figure 5: Membrane Stresses

A Appendices

A.1 Input file for *Shell_T_RM_v1_1* MatFEM code

```
%==========================================================================
% MAT-fem_Shells 1.0 - MAT-fem is a learning tool for undestanding
%                       the Finite Element Method with MATLAB and GiD
%==========================================================================
% PROBLEM TITLE = Hyperbolic Paraboloid Shell - CSMD Assignment 8
%
%  Material Properties
%
  young  =   3e10 ;
  poiss  =  0.2 ;
  denss  =   24e3 ;
  thick   =   0.1 ;
%
% Coordinates
%
global coordinates
X = linspace(-5,5,11);
b = X(end);        % width
h = b;               % length(square shell)
z0 = [2 -2 2 -2];         % z at corners

coordinates = [];

cont = 1;
for i = 1:length(X)      % y
    for j = 1:length(X) % x
        x = X(j);
        y = X(i);

        N1 = (b-x)*(h-y)/(4*b*h);
        N2 = (b+x)*(h-y)/(4*b*h);
        N3 = (b+x)*(h+y)/(4*b*h);
        N4 = (b-x)*(h+y)/(4*b*h);

        z = N1*z0(1) + N2*z0(2) + N3*z0(3) + N4*z0(4);

        coordinates(cont,1) = x;
        coordinates(cont,2) = y;
        coordinates(cont,3) = z;

        cont = cont +1;
    end
end

%
% Elements
%
global elements

elements = [];

cont = 1;
n = length(X);       % number of nodes in each direction
```

```
for i = 1:n-1      % y
    for j = 1:n-1 % x

        elements(cont,1) = n*(i-1) + j + 1;
        elements(cont,2) = n*(i) + j + 1;
        elements(cont,3) = n*(i-1) + j;

        cont = cont +1;

        elements(cont,1) = n*(i) + j;
        elements(cont,2) = elements(cont-1,3);
        elements(cont,3) = elements(cont-1,2);

        cont = cont +1;
    end
end

%
% Fixed Nodes
%
bound = [1 2 3 4 5 6 7 8 9 10 11 12 22 23 33 34 44 45 55 56 66 67 77 78 ...
    88 89 99 100 110 111 112 113 114 115 116 117 118 119 120 121];

fixnodes = [];
cont = 1;

for i = 1:length(bound)
    for j = 1:5
        fixnodes(cont,1) = bound(i);
        fixnodes(cont,2) = j;
        fixnodes(cont,3) = 0;

        cont = cont +1;

    end
end

%
% Point loads
%
pointload = [ ] ;
%
% Side loadsss
%
uniload = sparse ( size(elements,1) , 1 );
```