Aerospace Computational Engineering — GrETA

Assignment 4

Elastodynamics

David Beurnio

Biel Riera

Polithecnic University of Catalonia

January 16, 2023

Contents

1 Introduction

This assignment is a continuation of the previous one. The aim of the assignment 4 is to compute and study the behaviour of the beam in each of the first 25 modes of vibration on both cases (with the load and with the torque). For both cases the mesh 4 will be used. In this case, the applied load will suddenly be released and it will result in the addition of inertial forces to keep the equilibrium of forces.

2 Computation of the mass matrix

The first step is to compute the mass matrix. The equation that need to be solved is the differential equation that govern the free damped vibration of a given structure:

$$M_{ll}\ddot{d}_l(t) + D_{ll}\dot{d}_l + K_{ll}d_l(t) = 0 \tag{1}$$

where **D** is the damping matrix and **M** is the mass matrix. The nodal mass matrix of a structure is given by:

$$M = \int_{\Omega} \rho N^T N d\Omega \tag{2}$$

Knowing this, the matrix *ComputeK.m* from the previous assignment (and that will be used in ths one too) will be used to write the code for the mass matrix, as its form is very much the same with few changes. The code can be seen in Appendix A1.

3 Calculation of the natural frequencies and modes

To compute the natural frequencies and modes is necessary to modify the main script of the previous assignment. By doing this, the following data is saved in an independent data file: Stiffness matrix K, Mass matrix M, unrestricted degrees of freedom, coordinate matrix COOR, connectivity matrix CN, the type element, the position of the Gauss points posgp and the name of mesh generated by GiD. With this data, the frequencies and modes are computed with the function *UndampedFREQ* already given and can be seen in Appendix A2. Other functions given are also used to post process the GiD information and generate the corresponding files. After post processing the modes in GiD, the following representations of each mode are generated:

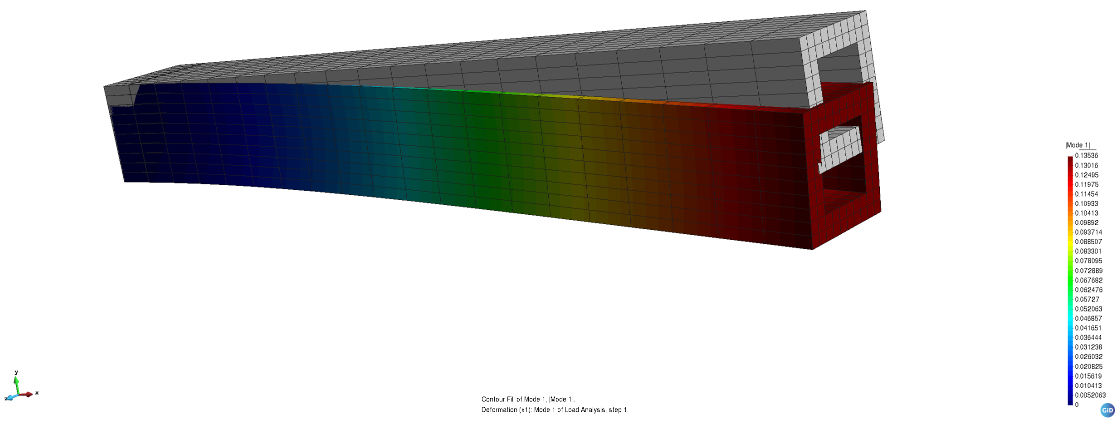

Figure 1: Mode 1

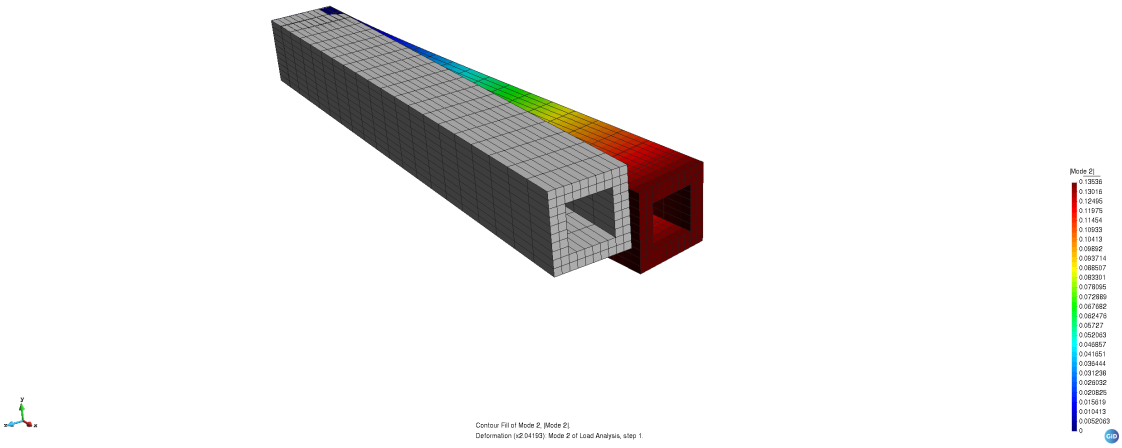

Figure 2: Mode 2

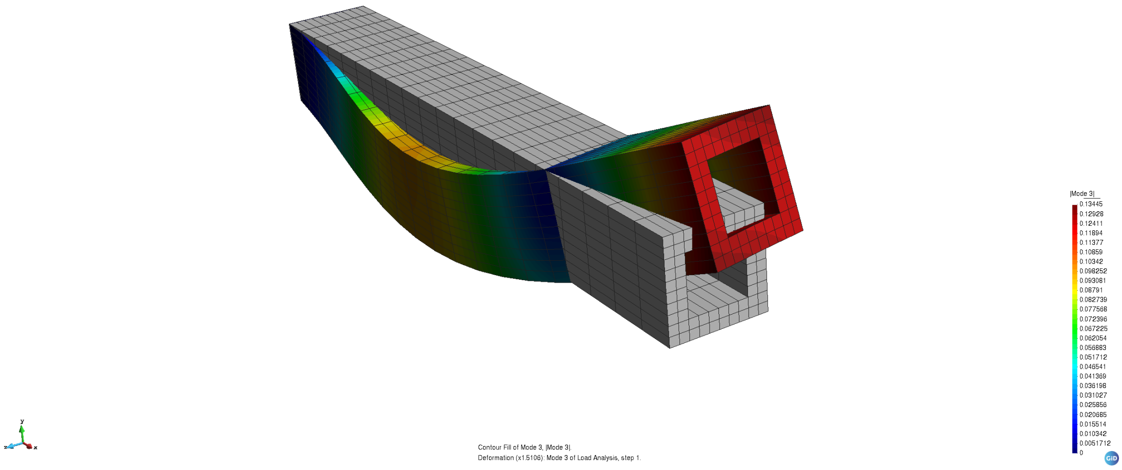

Figure 3: Mode 3

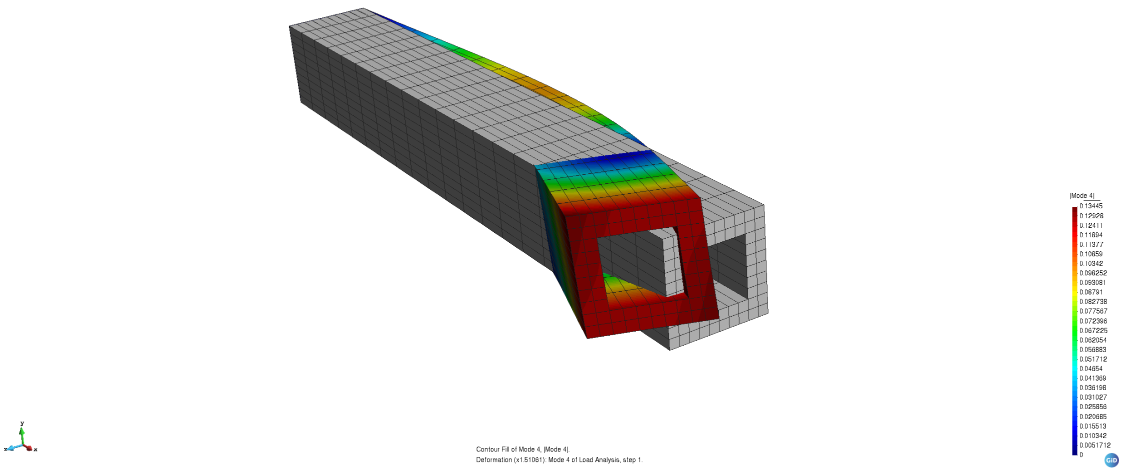

Figure 4: Mode 4

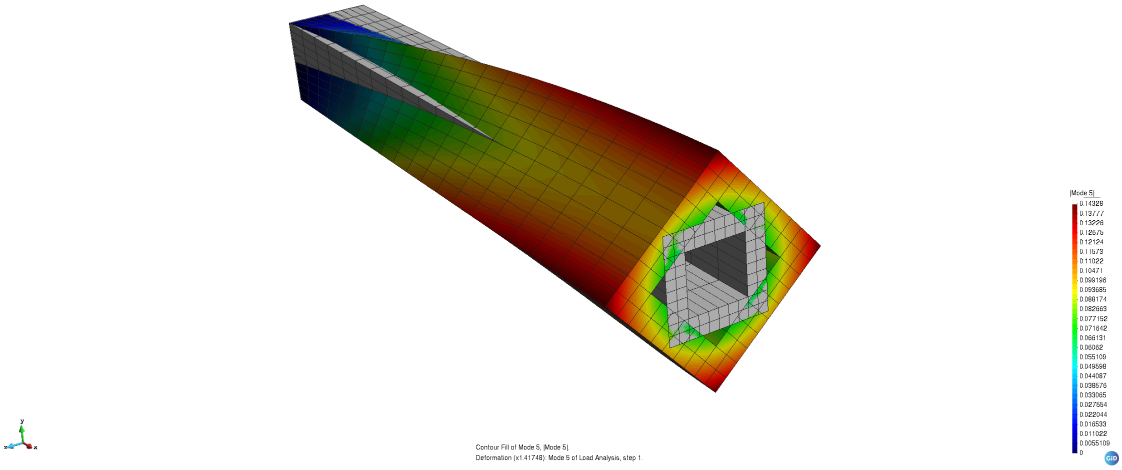

Figure 5: Mode 5

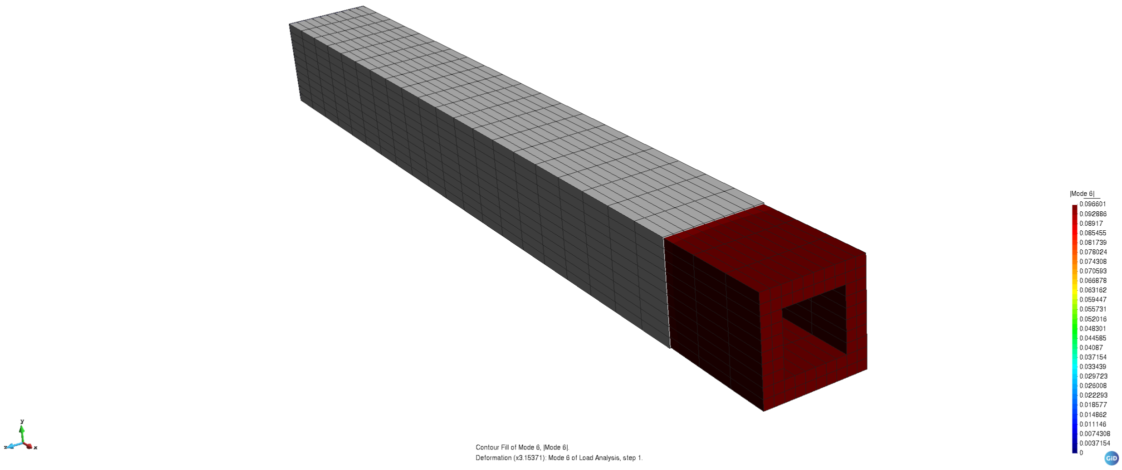

Figure 6: Mode 6

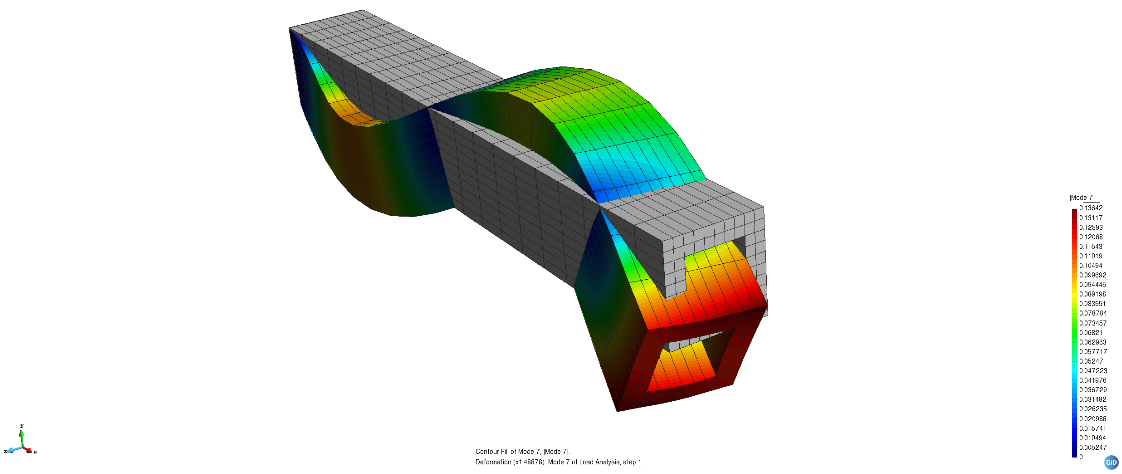

Figure 7: Mode 7

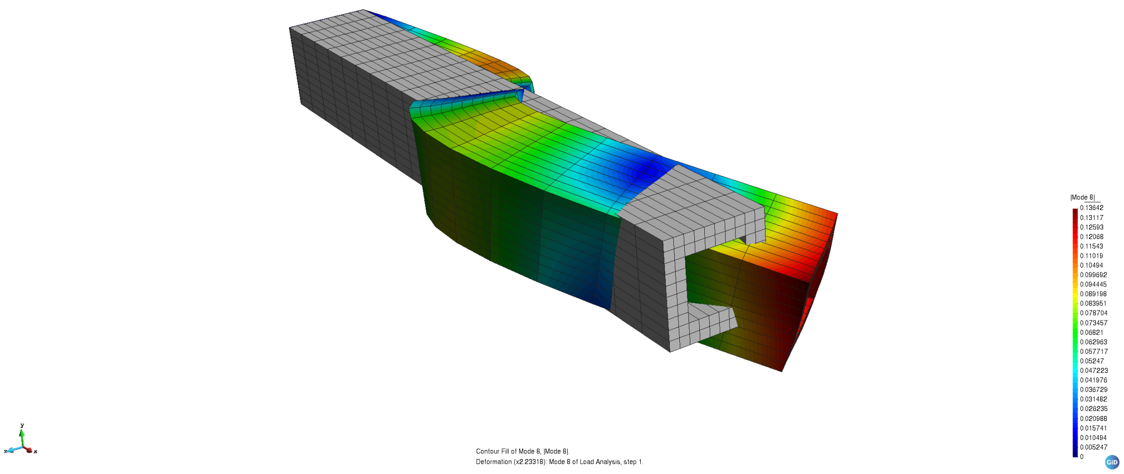

Figure 8: Mode 8

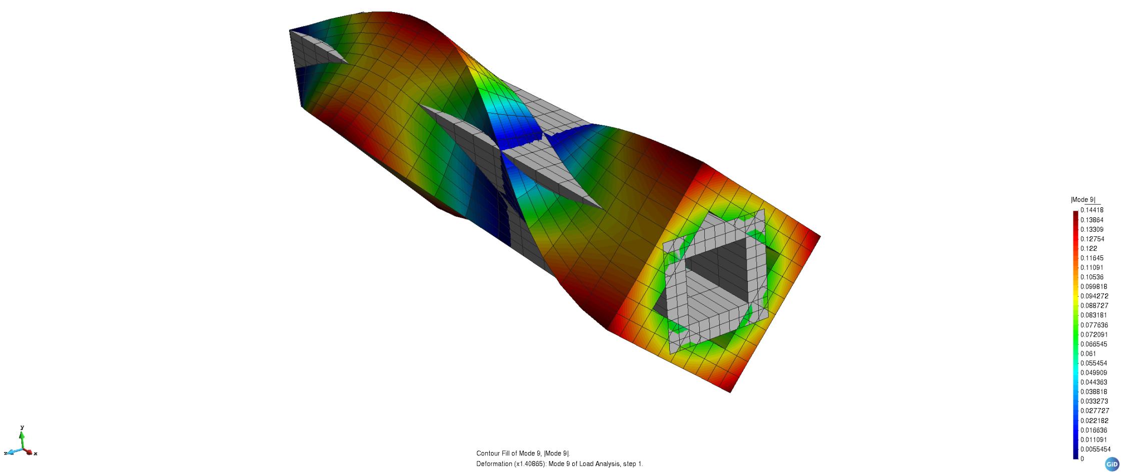

Figure 9: Mode 9

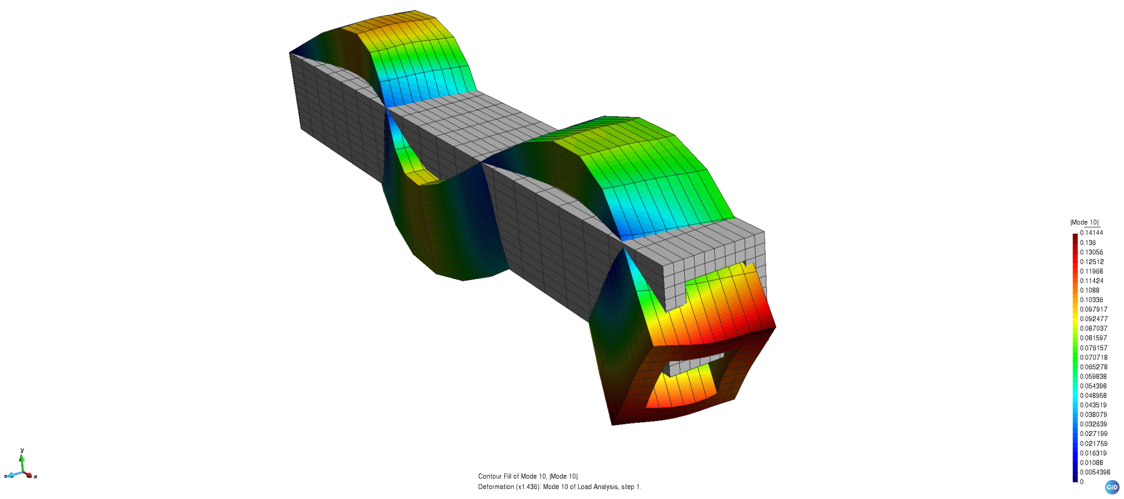

Figure 10: Mode 10

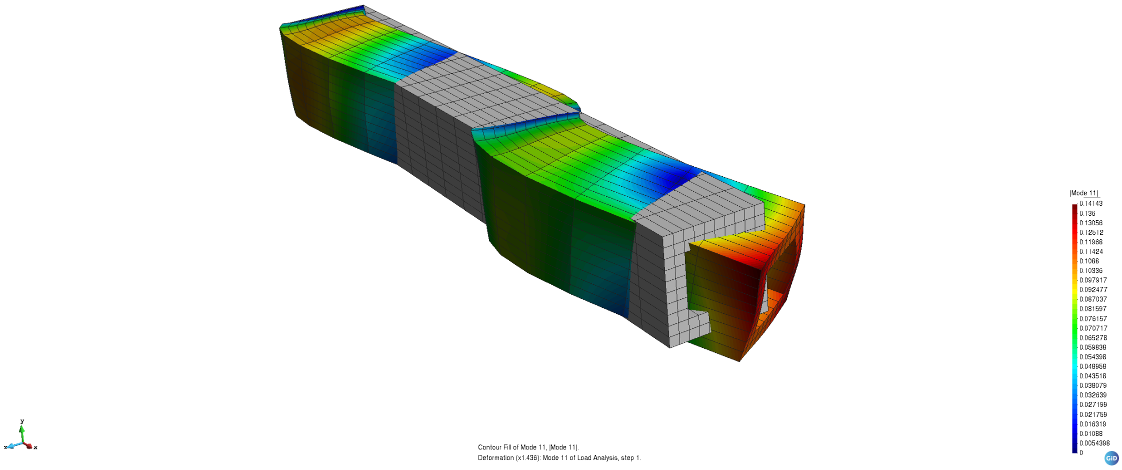

Figure 11: Mode 11

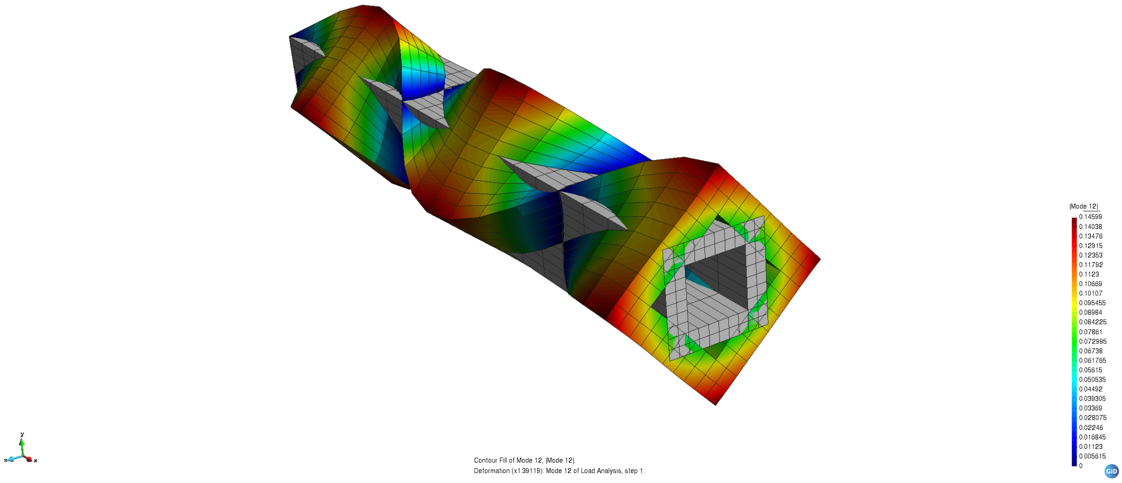

Figure 12: Mode 12

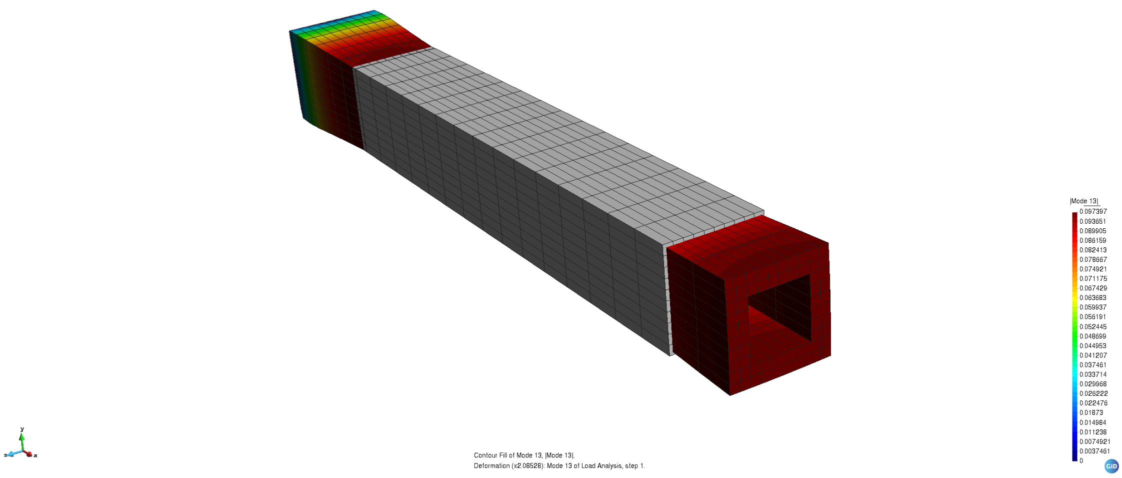

Figure 13: Mode 13

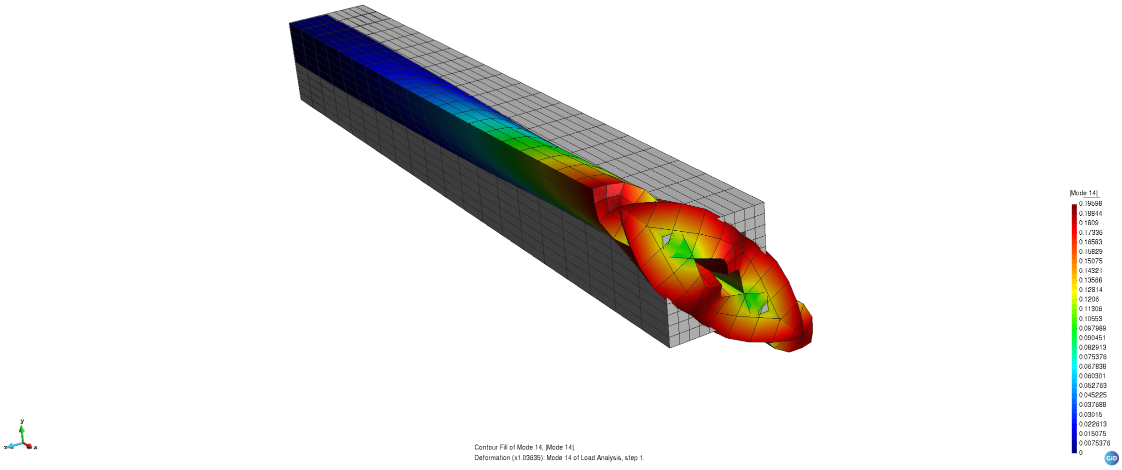

Figure 14: Mode 14

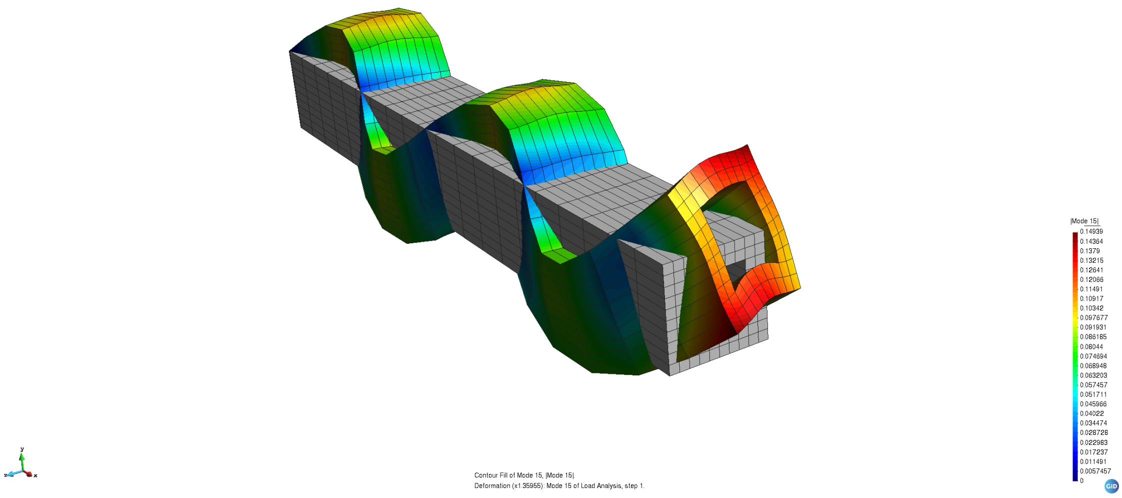

Figure 15: Mode 15

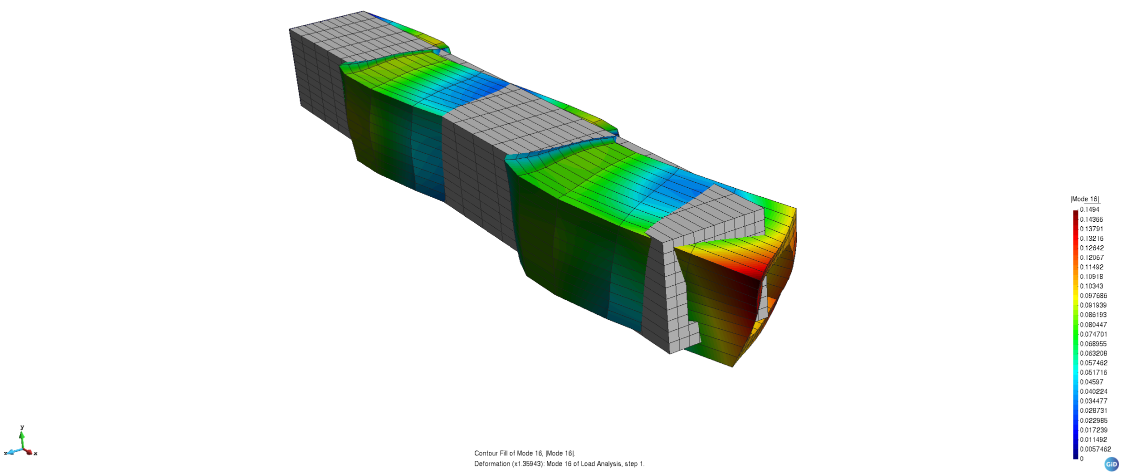

Figure 16: Mode 16

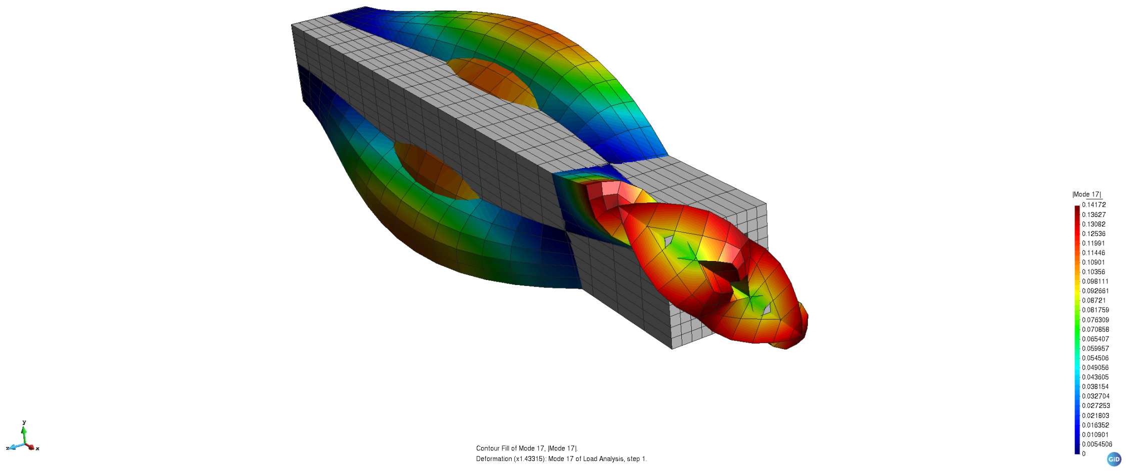

Figure 17: Mode 17

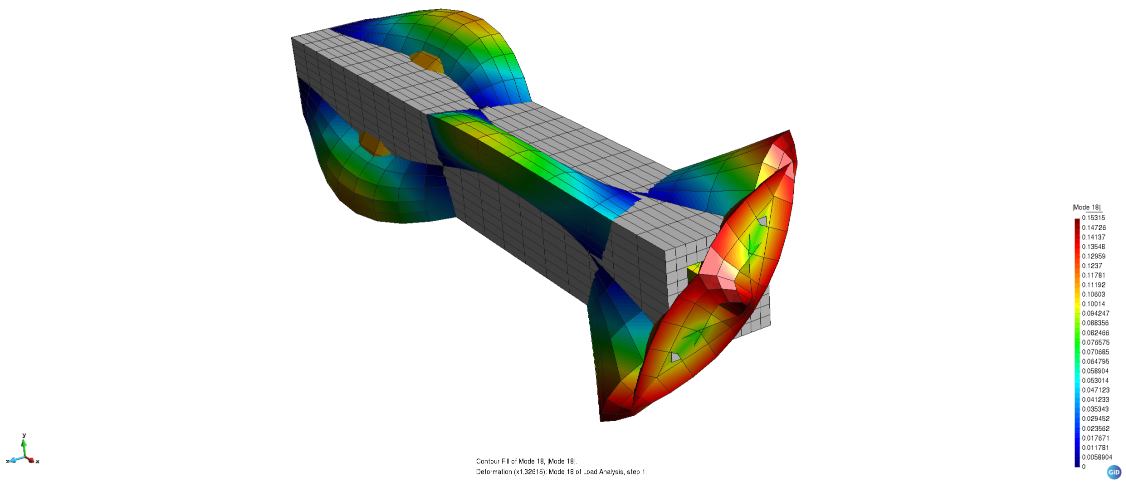

Figure 18: Mode 18

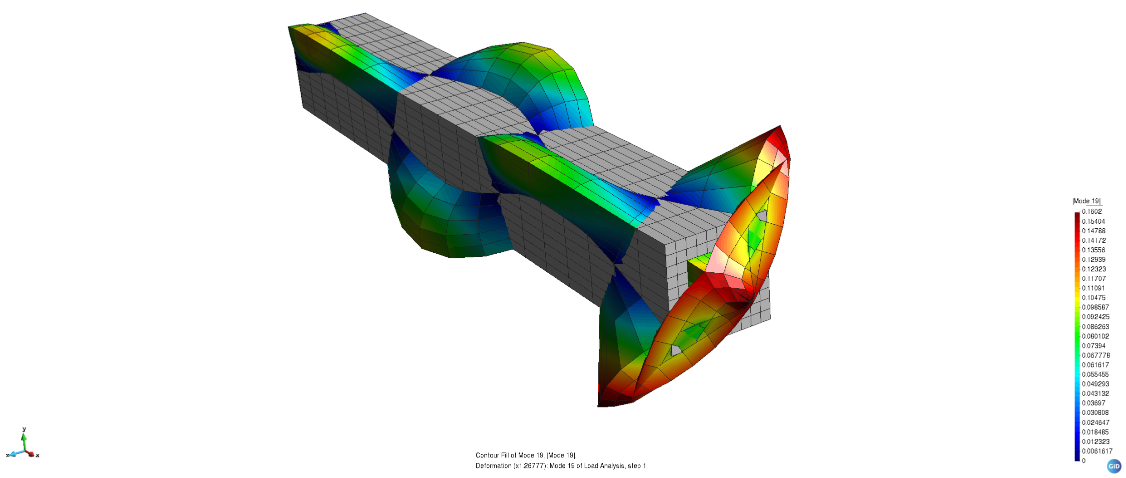

Figure 19: Mode 19

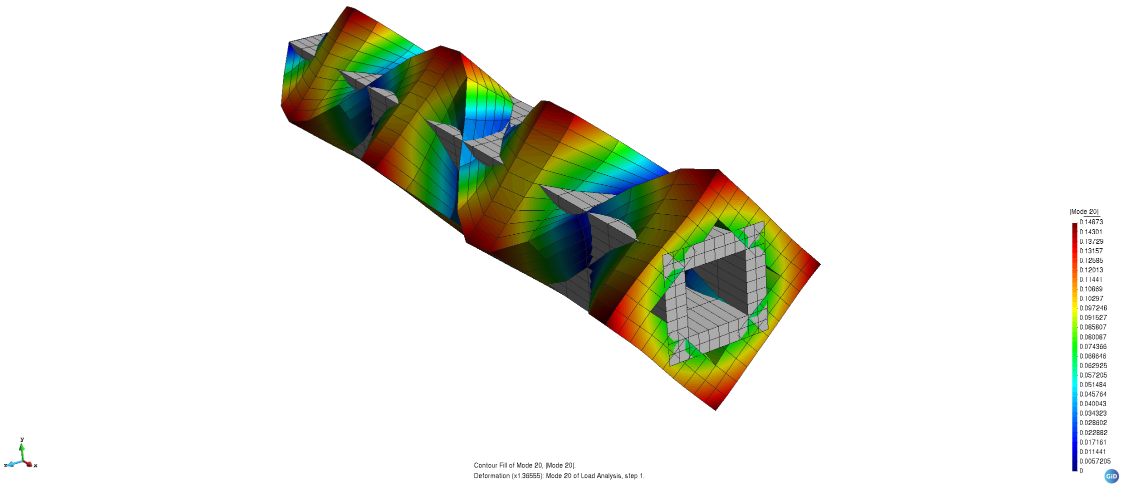

Figure 20: Mode 20

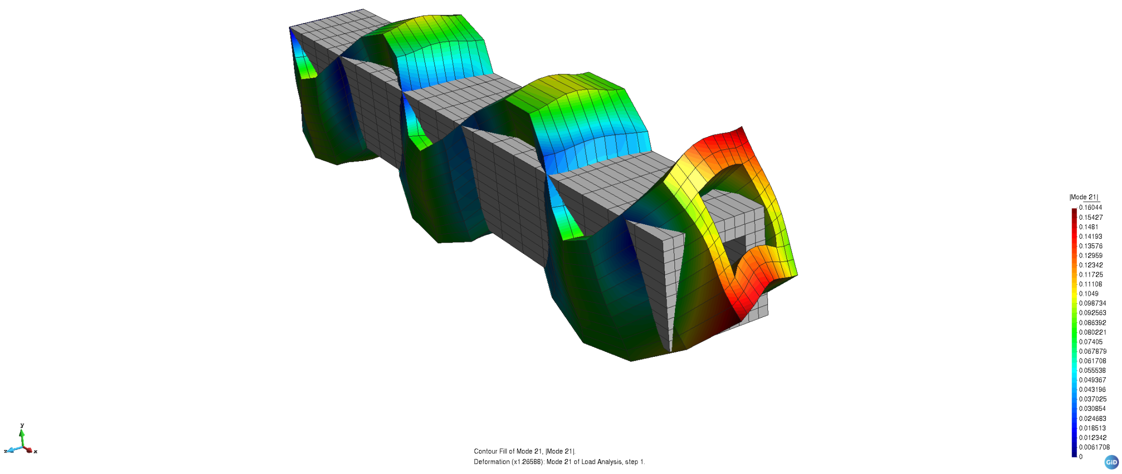

Figure 21: Mode 21

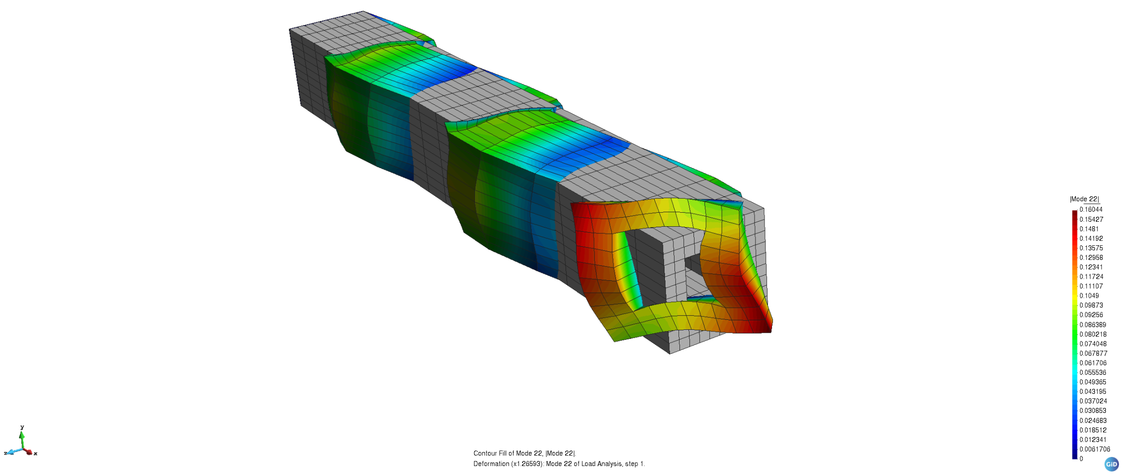

Figure 22: Mode 22

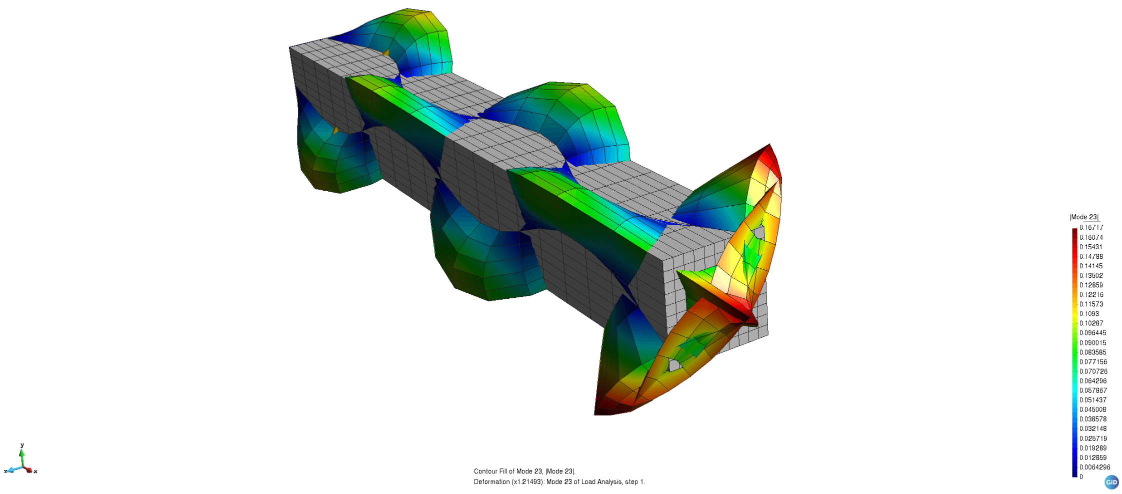

Figure 23: Mode 23

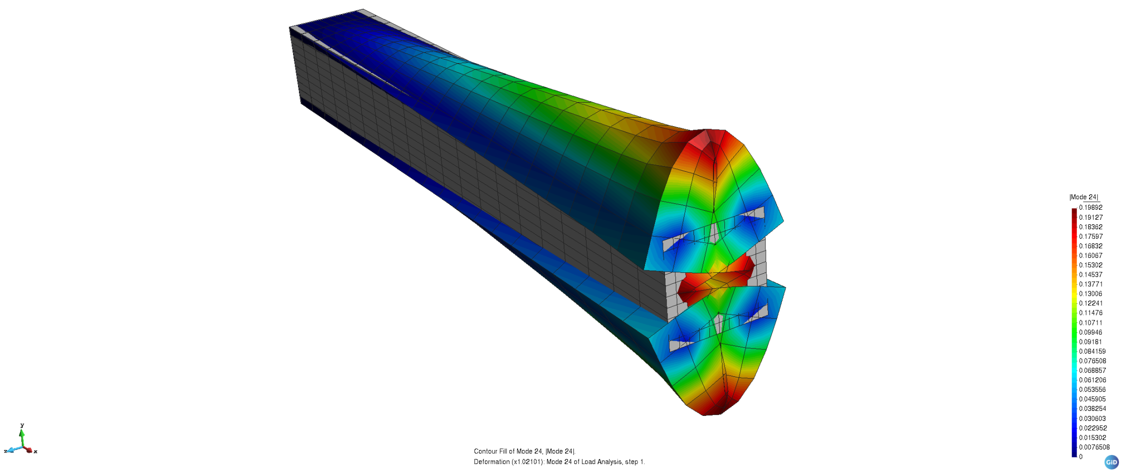

Figure 24: Mode 24

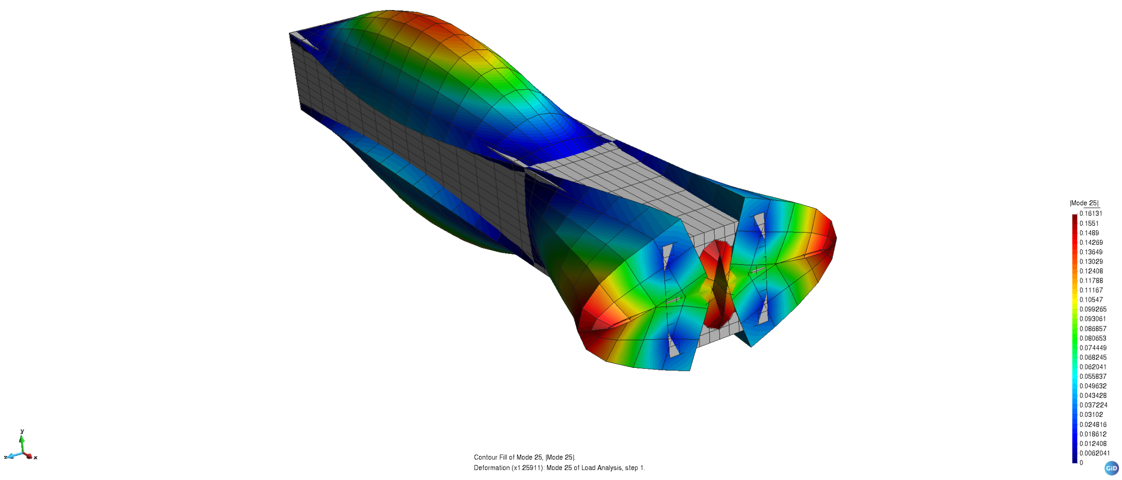

Figure 25: Mode 25

4 Dominant natural modes

To examine which are the natural modes that are more dominant in each case, the initial amplitude of the temporal response for each mode has been plotted. This way, the more dominant modes can be easily identified. To calculate the amplitude of each node the following equation has been used:

$$q^0 = \Phi^T M d^0 \tag{3}$$

4.1 Case for the distributed load on the surface $y = y_{max}$

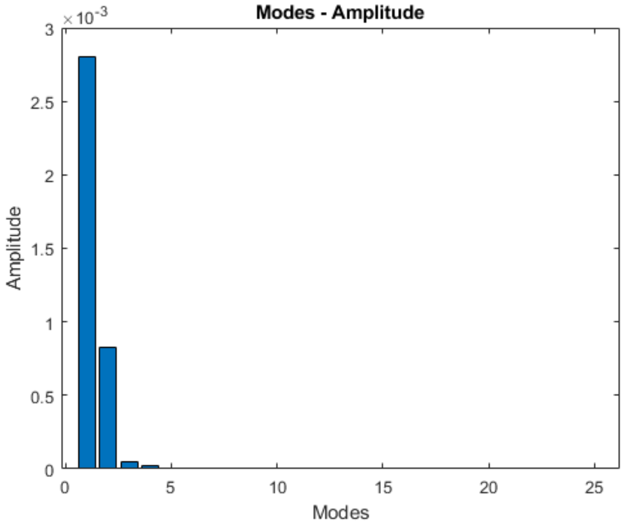

Figure 26: Modes VS Amplitude

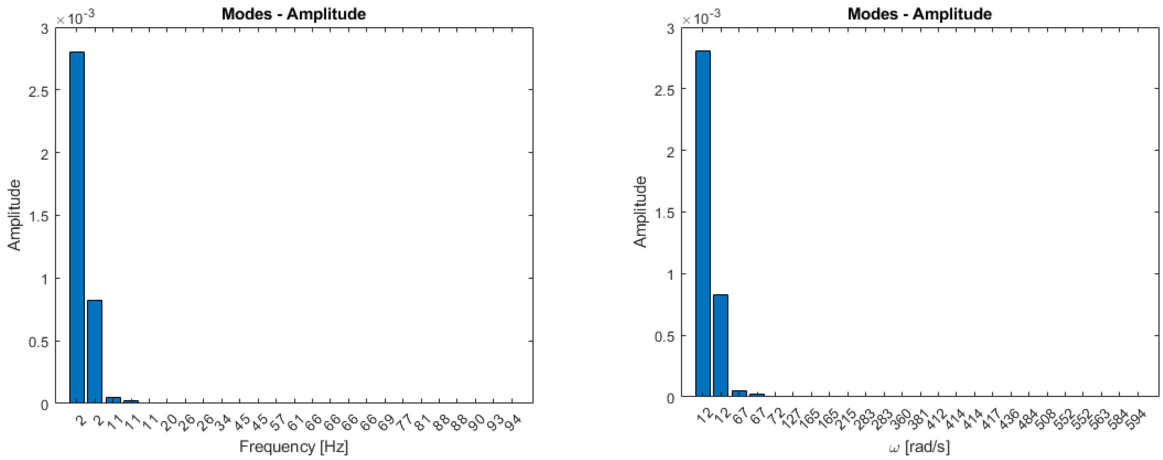

Figure 27: Frequency of each node VS Amplitude of each node

4.2 Case for the torque applied on the free end of the cantilever

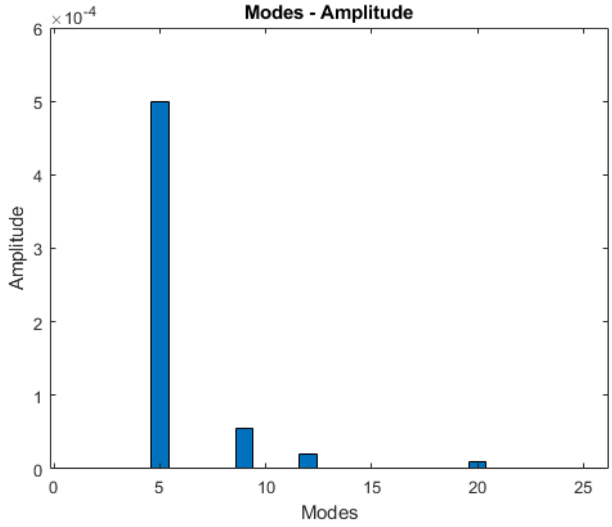

Figure 28: Modes VS Amplitude

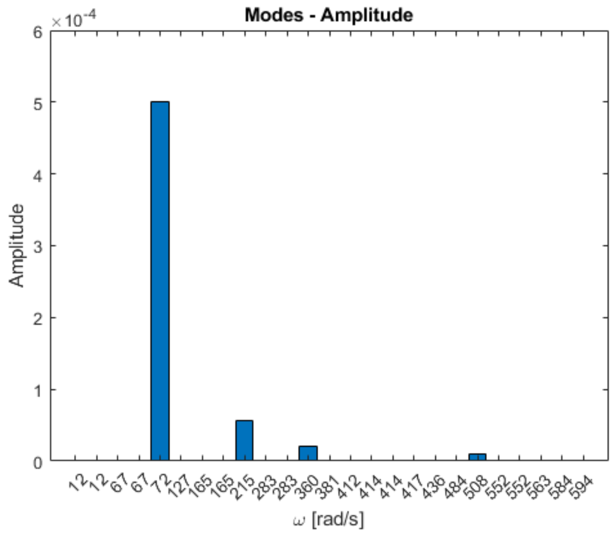

Figure 29: Frequency of each node (Hz) VS Amplitude of each node

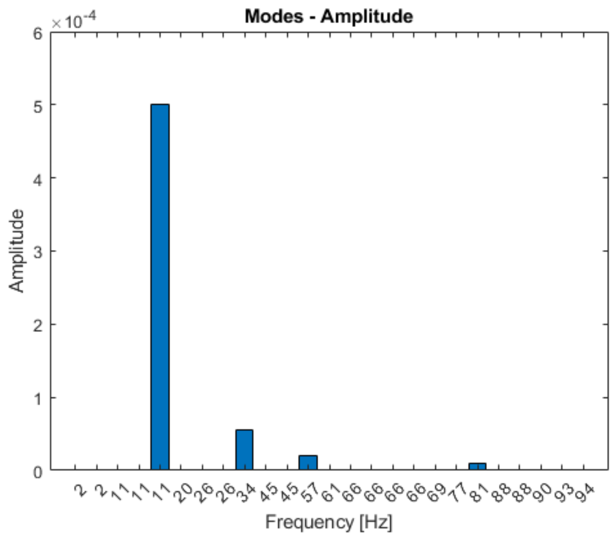

Figure 30: Frequency of each node (rad/s) VS Amplitude of each node

5 Generation of the elastodynamics oscillation videos

The videos generated show the oscillation for the two cases. The interval of time used for the videos is m times (where m=40) the first dominant frequency of each case (frequency 1 for the distributed load and frequency 5 for the torque). A total of 500 steps are used in the discretization of the time interval. The damping factor ξ is 0.01 as stated in the guidelines.
To see the videos generated, click **here**.

6 Conclusions

To conclude this report, it is clearly visible that there are four dominant modes for each case. In the distributed load case, the dominant modes are the first four modes (as seen in figure 26) whereas in the case where the torque is applied the dominant modes are 5, 9, 12 and 20, as you can see in figure 28. This is a fair representation of reality: in the first case, when a distributed load is applied in a cantilever beam its tendency is to deflect downwards, as seen in the first mode (figure 1); in the second case, the dominant mode is the fifth mode (figure 5) and its a right representation because the torque applied on the free end of the beam makes it rotate.

7 Bibliografia

[1] Hernández, J., *Chapter 4: Classical Linear Elastodynamics*, Lecture notes (Aeroespace Computational Engineering), 2022.

Appendix A

Codes

1 Mass matrix

```
function M = ComputeM(COOR,CN,TypeElement, densglo);

if nargin == 0
    load('tmp1.mat')
end
nnode = size(COOR,1); ndim = size(COOR,2); nelem = size(CN,1); nnodeE = size(CN,2) ;
% nstrain = size(celasglo,1) ;
% Shape function routines (for calculating shape functions and derivatives)
TypeIntegrand = 'K';
[weig,posgp,shapef,dershapef] = ComputeElementShapeFun(TypeElement,nnodeE,TypeIntegrand) ;
% Assembly of matrix K
% ----------------
M = sparse([],[],[],nnode*ndim,nnode*ndim,nnodeE*ndim*nelem) ;
for e = 1:nelem
    dens = densglo(e) ;  % Stiffness matrix of element "e"
    CNloc = CN(e,:) ;   % Coordinates of the nodes of element "e"
    Xe = COOR(CNloc,:)' ;     % Computation of elemental stiffness matrix
    Me = ComputeMeMatrix(dens,weig,dershapef,shapef,Xe) ;

    for an=1:nnodeE
        a=Nod2DOF(an,ndim);
        An=CN(e,an);
        A=Nod2DOF(An,ndim);
        for bn=1:nnodeE
            b=Nod2DOF(bn,ndim);
            Bn=CN(e,bn);
            B=Nod2DOF(Bn,ndim);
            M(A,B)=M(A,B)+Me(a,b);

        end
    end

    if mod(e,10)==0  % To display on the screen the number of element being assembled
        disp(['e=',num2str(e)])
    end

end
```

```
function Me = ComputeMeMatrix(dens,weig,dershapef,shapef,Xe)
ndim = size(Xe,1) ; ngaus = length(weig) ; nnodeE = size(Xe,2) ;
Me = zeros(nnodeE*ndim,nnodeE*ndim) ;
for  g = 1:ngaus
    BeXi = dershapef(:,:,g) ;
    NeSCL=shapef(g,:);
    Ne = StransfN(NeSCL,ndim) ;
    Je = Xe*BeXi' ;
    detJe = det(Je) ;
    Me = Me + weig(g)*(detJe*Ne'*dens*Ne) ;
end
end
```

2 Computation of frequencies and modes

```
 function [MODES FREQ] = UndampedFREQ(M,K, neig)
 % neig = Number of modes to be calculated
%  [MODES EIGENVAL] = eigs(M,K, neig) ;
%  EIGENVAL = diag(EIGENVAL) ;
%  FREQ = sqrt(1./EIGENVAL) ;
if nargin == 0
    load('tmp.mat')

end


% We turn M and K symmetric, because otherwise
% the modes are not orthogonalized with respect to M
% It should be noted that the lack of symmetry is caused by finine machine precision:  theoretically
    speaking,
% M and K are symmetric by construction
M = (M+M')/2;
K = (K+K')/2;


  [MODES EIGENVAL] = eigs(K,M, neig, 'sm') ;
  EIGENVAL = diag(EIGENVAL) ;
  FREQ = sqrt(EIGENVAL) ;

  [FREQ, imodes] = sort(FREQ) ;
  MODES= MODES(:, imodes);

 end
```