

```

function memberlocalFEF = MD_computeMemberFEFs_2ndnode_MyMz_release(w, L)
% Code developed by Mrunmayi Mungekar and Devasmit Dutta
%
% MD_computeMemberFEFs.m computes the element stiffness matrix for a given element
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Functions Called
%         none
%
% Dictionary of Variables
% Input information
%         % w = distributed load
%         % L = length of the member
%
% Output information
%         % memberlocalFEF = fixed end forces in the local element directions
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Take the load components along the local x', y', z' directions

    wx = w(1);
    wy = w(2);
    wz = w(3);

% Calculate the corresponding fixed end forces due to load in each local x', y', z'
directions

    FEF_X = [-wx*L/2;0;0;0;0;0; -wx*L/2;0;0;0;0;0];

    FEF_Y = [0;-wy*L/2;0;0;0;-wy*L^2/12; 0;-wy*L/2;0;0;0;wy*L^2/12];
    Mb = FEF_Y(12);
    FEF_Y(2) = FEF_Y(2) - (3/(2*L))*Mb;
    FEF_Y(8) = FEF_Y(8) + (3/(2*L))*Mb;
    FEF_Y(6) = FEF_Y(6) - (1/2)*Mb;
    FEF_Y(12) = 0;

    FEF_Z = [0;0;-wz*L/2;0;wz*L^2/12;0; 0;0;-wz*L/2;0;-wz*L^2/12;0];
    Mb = FEF_Z(11);
    FEF_Z(3) = FEF_Z(3) - (3/(2*L))*Mb;
    FEF_Z(9) = FEF_Z(9) + (3/(2*L))*Mb;
    FEF_Z(5) = FEF_Z(5) - (1/2)*Mb;
    FEF_Z(11) = 0;

% Sum up to get the total fixed end forces

    FEF = FEF_X + FEF_Y + FEF_Z;
    memberlocalFEF = FEF;

end

```