```
function [gamma] = MD_etran (coordi, coordj, webdir)
% Code developed by Mrunmayi Mungekar and Devasmit Dutta
%
% MD_etran.m computes the element transformation matrix for a given element
%

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  Functions Called
%              none
%
% Dictionary of variables
% Input information
                % coordi  = 1st coordinate of given element in [x ;y ;z] form
                % coordj  = 2nd coordinate of given element in [x ;y ;z] form
                % webdir  = direction of the web for given element in vector form
%
% Output information
                % gamma = final transformation matrix in 12 x 12 form
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define global coordinate directions

global_x = [1 ;0 ;0];
global_y = [0 ;1 ;0];
global_z = [0 ;0 ;1];

% Define local coordinate directions

local_x = coordj - coordi;
local_y = webdir;

% Define the angles between local and global axes

phixx = acos(dot(local_x,global_x)/(norm(local_x)*norm(global_x)));
phixy = acos(dot(local_x,global_y)/(norm(local_x)*norm(global_y)));
phixz = acos(dot(local_x,global_z)/(norm(local_x)*norm(global_z)));
phiyx = acos(dot(local_y,global_x)/(norm(local_y)*norm(global_x)));
phiyy = acos(dot(local_y,global_y)/(norm(local_y)*norm(global_y)));
phiyz = acos(dot(local_y,global_z)/(norm(local_y)*norm(global_z)));

% Evaluating the direction cosines

mat = [cos(phixx) cos(phixy) cos(phixz);
       cos(phiyx) cos(phiyy) cos(phiyz)];

G = zeros(3,3);
G(1:2,:) = mat;

% Getting the direction cosine for the local Z direction

G(3,:) = cross(mat(1,:),mat(2,:));


% Consolidating the final gamma as 12 x 12 transformation matrix

gamma = [   G                  zeros(3,9);
```

```
zeros(3,3)      G     zeros(3,6);
zeros(3,6)      G     zeros(3,3);
zeros(3,9)            G];
```