Table of Contents


```
clc
clear
close all

E = 70E9;
v = 0.3;
quad_coord = [0,0;
              2,0;
              2,1;
              0,2];

ref_coord = [-1/sqrt(3), -1/sqrt(3);
              1/sqrt(3), -1/sqrt(3);
              1/sqrt(3),  1/sqrt(3);
             -1/sqrt(3),  1/sqrt(3)];

surf_coord = [1, 1/sqrt(3);
              1, -1/sqrt(3)];

f = [0;1E6]; %Pa
T = [1E6;0]; %Pa
```

Question 1

```
[N,dNdS,J] = element(ref_coord(2,:),quad_coord);
```

Question 2

```
bodyForceTerm = zeros(8,1);
for i = 1:size(ref_coord,1)
    [out] = bodyForceQuad(f,ref_coord(i,:),quad_coord);
    bodyForceTerm = bodyForceTerm + out;
end
```

Question 3

```
surfForceTerm = zeros(8,1);
for i = 1:size(surf_coord,1)
    [out] = tractionQuad(T,surf_coord(i,:),quad_coord);
```

```
    surfForceTerm = surfForceTerm + out;
end
```

Question 4

```
K = zeros(8,8);
for i = 1:size(ref_coord,1)
    [out] = stiffMatrix(E,v,ref_coord(i,:),quad_coord);
    K = K + out;
end

K = K/1E9;
% Open a file for writing
fid = fopen('matrix.tex', 'w');

% Write the matrix header
fprintf(fid, '\\begin{bmatrix}\n');

% Write the matrix contents with limited decimal places
fprintf(fid, '\\tfrac{');
fprintf(fid, '%.3f & ', K(1:end-1,1:end-1));
fprintf(fid, '%.3f \\\\[2ex]\n', K(end,1:end-1));
fprintf(fid, '\\hdotsfor{3}\n');
fprintf(fid, '\\tfrac{');
fprintf(fid, '%.3f & ', K(1:end-1,end));
fprintf(fid, '%.3f \\\\ \n', K(end,end));
fprintf(fid, '\\end{bmatrix}\n');

% Close the file
fclose(fid);
```

Functions Declared

```
function [N,dNdS,J] = element(ref,quad_coord)
    %Input:
    %       xi [1x1]: reference coordinate x
    %       nu [1x1]: reference coordinate y
    %       quad_coord [4x2]: global coordinates

    syms xi_sym nu_sym
    N_1 = (nu_sym - 1).*(xi_sym - 1)./4;
    N_2 = -(xi_sym + 1).*(nu_sym - 1)./4;
    N_3 = (xi_sym + 1).*(nu_sym + 1)./4;
    N_4 = -(xi_sym - 1).*(nu_sym + 1)./4;

    N = [N_1,N_2,N_3,N_4];

    dNdS = [diff(N,xi_sym); diff(N,nu_sym)];

    N = subs(N,[xi_sym,nu_sym],[ref(1),ref(2)]);
    N = double(N);

    dNdS = subs(dNdS,[xi_sym,nu_sym],[ref(1),ref(2)]);
```

```
    dNdS = double(dNdS);

    J = dNdS*quad_coord;
end

function [out] = bodyForceQuad(f,ref,quad_coord)
    syms xi_sym nu_sym
    N_1 = (nu_sym - 1).*(xi_sym - 1)./4;
    N_2 = -(xi_sym + 1).*(nu_sym - 1)./4;
    N_3 = (xi_sym + 1).*(nu_sym + 1)./4;
    N_4 = -(xi_sym - 1).*(nu_sym + 1)./4;

    N = [N_1, 0, N_2, 0, N_3, 0, N_4, 0;
         0, N_1, 0, N_2, 0, N_3, 0, N_4];

    N = subs(N, [xi_sym, nu_sym], [ref(1), ref(2)]);
    N = double(N);
    [~,~,J] = element(ref,quad_coord);

    out = N'*f.*det(J);
end

function [out] = tractionQuad(T,ref,quad_coord)
    syms xi_sym nu_sym
    N_1 = (nu_sym - 1).*(xi_sym - 1)./4;
    N_2 = -(xi_sym + 1).*(nu_sym - 1)./4;
    N_3 = (xi_sym + 1).*(nu_sym + 1)./4;
    N_4 = -(xi_sym - 1).*(nu_sym + 1)./4;

    N = [N_1, 0, N_2, 0, N_3, 0, N_4, 0;
         0, N_1, 0, N_2, 0, N_3, 0, N_4];

    N = subs(N, [xi_sym, nu_sym], [ref(1), ref(2)]);
    N = double(N);

    [~,~,J] = element(ref,quad_coord);

    out = N'*T.*det(J);
end

function [out] = stiffMatrix(E,v,ref,quad_coord)
    D = E./(1-v.^2).*[1, v, 0;
                      v, 1, 0;
                      0, 0, (1-v)./2];
    syms xi_sym nu_sym
    N_1 = (nu_sym - 1).*(xi_sym - 1)./4;
    N_2 = -(xi_sym + 1).*(nu_sym - 1)./4;
    N_3 = (xi_sym + 1).*(nu_sym + 1)./4;
    N_4 = -(xi_sym - 1).*(nu_sym + 1)./4;

    B_1 = [diff(N_1,xi_sym), 0, diff(N_2,xi_sym), 0, diff(N_3, xi_sym), 0,
 diff(N_4,xi_sym),0];
    B_2 = [0, diff(N_1,nu_sym), 0, diff(N_2,nu_sym), 0, diff(N_3, nu_sym), 0,
 diff(N_4,nu_sym)];
```

```
    B_3 = [diff(N_1,nu_sym), diff(N_1,xi_sym), diff(N_2,nu_sym),
 diff(N_2,xi_sym), diff(N_3, nu_sym), diff(N_3, xi_sym), diff(N_4,nu_sym),
 diff(N_4,xi_sym)];

    B = [B_1;B_2;B_3];
    B = subs(B, [xi_sym, nu_sym],[ref(1),ref(2)]);
    B = double(B);
    [~,~,J] = element(ref,quad_coord);

    out = B'*D*B.*det(J);

end
```

Published with MATLAB® R2022b