OEMP 3 Part 2

```
clc
clear
format shortG
% minimum beam dimension is 1/4 in
% multiple for loop, iterating from 1/4 in to 1 ft on each dimension
rangeSquare = linspace((0.25/12),1,50); % [ft]
% rangeMoment = linspace(0,27.25,100); % [ft]
% rangeSquare = 0.5;
% rangeSquare2 = 0.25;
rho = 276.48; % [lb/ft^3] titanium
L = 27.25;
w = 2001;

% momentTest = zeros(1,1);
% for k = rangeMoment
%     w = 2001;
%     L = 27.25;
%     A = 0.13485;
%     momentTest(end+1) = (1/6) * (((-w / L) * k^3) + (3 * w * k^2) -
 (3 * w * k * L) + (w * L^2)) - ((rho * A * (L - k)^2)/2);
% end
% momentTest
% max(momentTest)

base = zeros(1,1);
height = zeros(1,1);
area = zeros(1,1);
moment = zeros(1,1);
inertia = zeros(1,1);
dist = zeros(1,1);

centerWidthI = zeros(1,1);
flangeWidthI = zeros(1,1);
flangeHeightI = zeros(1,1);
centerHeightI = zeros(1,1);
areaI = zeros(1,1);
areaM = zeros(1,1);
areaF = zeros(1,1);
momentI = zeros(1,1);
inertiaI = zeros(1,1);
inertiaF = zeros(1,1);
distI = zeros(1,1);
dy = zeros(1,1);
heightI = zeros(1,1);
ShearForceI = zeros(1,1);
ShearForceS = zeros(1,1);
ShearStressS = zeros(1,1);
ShearStressI = zeros(1,1);
totalHeightI = zeros(1,1);
costI = zeros(1,1);
```

```
costS = zeros(1,1);
FSShearI = zeros(1,1);
FSShearS = zeros(1,1);


YS_vec = [5040000 10080000 5040000 16560000 17280000]; % [lb/ft^2]
rho_vec = [169.344 489.024 525.312 490.752 276.48]; % [lb/ft^3]
cost_vec = [8.03 8.07 53.78 29.63 115.36]; % [$/lb]

% yield strength of titanium in lb/ft^2
titanium_YS = 17280000; % [lb/ft^2]
steel4130_YS = 10080000; % [lb/ft^2]

bestSquare = zeros(1,11);
squareVec = zeros(1,11);
bestIBeam = zeros(1,13);

for c = 1:length(YS_vec)
    for centerWidth = linspace((0.25/12),0.5,50)
        for flangeWidth = linspace((0.5/12),1,50)
            for flangeHeight = linspace((0.25/12),0.25,50)
                for centerHeight = linspace((0.25/12),1-
(2.*flangeHeight),50)
                    centerWidthI(end+1) = centerWidth;
                    flangeWidthI(end+1) = flangeWidth;
                    flangeHeightI(end+1) = flangeHeight;
                    centerHeightI(end+1) = centerHeight;
                    areaI(end+1) = (centerWidth * centerHeight) + (2 *
 (flangeHeight * flangeWidth));
                    areaM(end+1) = centerWidth * centerHeight;
                    areaF(end+1) = flangeHeight * flangeWidth;
                    heightI(end+1) = (centerHeight + (2 *
 flangeHeight)) / 2;
                    totalHeightI(end+1) = centerHeight + (2 *
 flangeHeight);
                    dy(end+1) = (centerHeight / 2) + (flangeHeight /
 2);
                    momentI(end+1) = (1/6) * (((-w / L) * (0)^3) +
 (3 * w * (0)^2) - (3 * w * (0) * L) + (w * L^2)) - ((rho_vec(1,c) *
 areaI(end) * (L - (0))^2)/2);
                    inertiaF(end+1) = (flangeWidth *
 flangeHeight^3)/12;
                    inertiaI(end+1) = 2 * (inertiaF(end) + areaF(end)
 * dy(end)^2);

                    ShearForceI(end+1) = (w * L)/2 - ((rho_vec(1,c) *
 areaI(end)) * L);
                    ShearStressI(end+1) = (ShearForceI(end)/(8 *
 inertiaI(end) * centerWidth)) * ((flangeWidth * totalHeightI(end)^3)
 - (flangeWidth * centerHeight^3) + (centerWidth * centerHeight^3));
                    costI(end+1) = (areaI(end) * L * rho_vec(1,c)) *
 cost_vec(1,c);
                end
            end
```

```
        end
    end
    FSShearI = YS_vec(1,c)./abs(ShearStressI);
    bendStressI = (-momentI .* heightI) ./ inertiaI;
    FSI = YS_vec(1,c)./abs(bendStressI);
    matrixI = [centerWidthI' centerHeightI' flangeWidthI'
flangeHeightI' areaI' momentI' inertiaI' heightI' bendStressI' FSI'
FSShearI' costI' ShearStressI'];
    matrixI(1,:) = [];
    holderMI = matrixI((matrixI(:,10) >= 1.5 & matrixI(:,11) >=
1.5),:);
    % potentialI = holderMI((holderMI(:,10) <= 1.53),:); % all
combinations in the FS range
    lowestAI = min(holderMI(:,12));
    [~,ind] = min(abs(holderMI(:,12)-lowestAI));
    bestIBeam(end+1,:) = holderMI(ind,:);

    for i = rangeSquare
        for j = rangeSquare
            area(end+1) = i * j; % rectangular cross section
            base(end+1) = i; % arbitrary choice but okay as long as
consistent
            height(end+1) = j;
            % maximum bending moment is at the origin
            moment(end+1) = (1/6) * (((-w / L) * (0)^3) + (3 * w *
(0)^2) - (3 * w * (0) * L) + (w * L^2)) - ((rho_vec(1,c) * area(end)
* (L - (0))^2)/2);
            inertia(end+1) = (i * j^3)/12;
            dist(end+1) = j/2; % distance is just half of height (from
centroid to edge)

            ShearForceS(end+1) = (w * L)/2 - ((rho_vec(1,c) *
areaI(end)) * L);
            ShearStressS(end+1) = (3/2) * (ShearForceS(end)/
area(end));
            costS(end+1) = (area(end) * L * rho_vec(1,c)) *
cost_vec(1,c);
        end
    end
    FSShearS = YS_vec(1,c)./abs(ShearStressS);
    bendStress = (-moment .* dist) ./ inertia;
    FS = YS_vec(1,c)./abs(bendStress);
    matrix = [base' height' area' moment' inertia' dist' bendStress'
FS' FSShearS' costS' ShearStressS'];
    matrix(1,:) = [];
    holderM = matrix((matrix(:,8) >= 1.5 & matrix(:,9) >= 1.5),:);
    potential = holderM((holderM(:,8) <= 1.53),:); % all combinations
in the FS range
    lowestArea = min(potential(:,3));
    [~,indAREA] = min(abs(potential(:,3)-lowestArea));
    squareVec(end+1,:) = potential(indAREA,:);
    lowestAS = min(holderM(:,10));
    [~,indS] = min(abs(holderM(:,10)-lowestAS));
    bestSquare(end+1,:) = holderM(indS,:);
```

```
end
bestIBeam
bestSquare
squareVec


bestIBeam =

  Columns 1 through 6

            0            0            0            0            0
       0
     0.020833      0.82738      0.98044      0.08631      0.18648
 2.3592e+05
     0.020833      0.92092            1     0.039541     0.098267
 2.4147e+05
     0.020833      0.82738      0.98044      0.08631      0.18648
 2.3592e+05
     0.020833      0.94898      0.92177      0.02551     0.066799
 2.4344e+05
     0.020833      0.93963      0.74575     0.030187     0.064599
 2.4358e+05

  Columns 7 through 12

            0            0            0            0            0
       0
     0.035427          0.5  -3.3296e+06       1.5137       2.5796
  6910.1
     0.018248          0.5  -6.6162e+06       1.5235       4.8607
  3641.3
     0.035427          0.5  -3.3296e+06       1.5137       2.5796
  6910.1
     0.011168          0.5    -1.09e+07       1.5193       7.5319
  2475.3
      0.01059          0.5  -1.1501e+07       1.5025       7.8347
  2393.8

  Column 13

            0
   1.9538e+06
   2.0738e+06
   1.9538e+06
   2.1986e+06
   2.2056e+06


bestSquare =

  Columns 1 through 6

            0            0            0            0            0
       0
```

```
     0.40051           1      0.40051   2.2246e+05     0.033376
    0.5
     0.22066     0.98002      0.21625   2.3405e+05     0.017308
0.49001
     0.40051           1      0.40051   2.2246e+05     0.033376
    0.5
     0.14073     0.98002      0.13792   2.3897e+05     0.011039
0.49001
     0.14073     0.96003      0.13511   2.3915e+05     0.010377
0.48002

  Columns 7 through 11

           0           0           0           0           0
 -3.3327e+06      1.5123      56.536       14841       89146
 -6.6261e+06      1.5213      61.053      8013.4    1.651e+05
 -3.3327e+06      1.5123      56.536       14841       89146
 -1.0608e+07      1.5611      63.969      5110.7   2.5888e+05
 -1.1063e+07       1.562      65.389      5006.4   2.6426e+05


squareVec =

  Columns 1 through 6

           0           0           0           0           0
     0
     0.40051           1      0.40051   2.2246e+05     0.033376
    0.5
     0.20068     0.98002      0.19667   2.1194e+05     0.015741
0.49001
     0.34056     0.98002      0.33376   1.8255e+05     0.026713
0.49001
     0.16071     0.90009      0.14466   2.3855e+05    0.0097662
0.45004
     0.16071      0.8801      0.14144   2.3875e+05      0.00913
0.44005

  Columns 7 through 11

           0           0           0           0           0
 -3.3327e+06      1.5123      56.536       14841       89146
 -6.5976e+06      1.5278      76.531       21150   1.3171e+05
 -3.3486e+06      1.5051      67.851   2.5694e+05       74280
 -1.0993e+07      1.5064      67.094      5360.3   2.4682e+05
 -1.1507e+07      1.5016      68.456      5241.3   2.5242e+05
```

Published with MATLAB® R2020a