Table of Contents

OEMP 3 Group 2 Part 2

Group 11, 10:40 lab

Code by Ian Faber and Daniel Mascarenas

```
clear; clc; close all;
```

Constants

```
w0 = 2001/12; % lb/in
L = 27.25*12; % in

maxDim = 1*12; % in

%Titanium
rho = 0.16; % lb/in^3
sigmaYield = 120000; % psi
```

Equations

```
moment = @(A, x) (w0/2)*(((-x^3)/(3*L)) + x^2 - L*x + (L^2)/2) - (rho*
 A * (L-x)^2)/2;

inertiaCircle = @(r) (pi*r^4)/4;
inertiaRectangle = @(b, h) (1/12)*b*h^3;

bendingStress = @(M, y, I) (M*y)/I;

factorOfSafety = @(sigmaYield, sigmaApplied) sigmaYield/sigmaApplied;

maxX = @(A) -L*(2*A*rho - w0)/w0;

factors.shape = [];
factors.area = [];
factors.factorOfSafety = [];
factors.efficiency = [];
```

Circle

```
for k = 2:8*(maxDim/2) % 1/4 in to 6 in diameter

    r = k/8;

    area = pi*r^2;
    inertia = inertiaCircle(r);

    x = maxX(area);

    sigmaBend = bendingStress(moment(area, x), r, inertia);
    factorCircle = factorOfSafety(sigmaYield, sigmaBend);

    if(factorCircle >= 1.5 && factorCircle <= 1.53)
        description = sprintf("circle, r = %.3f",r);
        factors.shape = [factors.shape; description];
        factors.area = [factors.area; area];
        factors.factorOfSafety = [factors.factorOfSafety;
 factorCircle];
        factors.efficiency = [factors.efficiency; factorCircle/area];
    end

end

factors.shape = [factors.shape; ""];
factors.area = [factors.area; 0];
factors.factorOfSafety = [factors.factorOfSafety; 0];
factors.efficiency = [factors.efficiency; 0];
```

Rectangle

```
for k = 1:4*maxDim % 1/4 in to 12 in length
    for l = 1:4*maxDim % 1/4 in to 12 in height
        b = k/4;
        h = l/4;

        area = b*h;
        inertia = inertiaRectangle(b,h);

        x = maxX(area);

        sigmaBend = bendingStress(moment(area, 0), h/2, inertia);
        factorRectangle = factorOfSafety(sigmaYield, sigmaBend);

        if(factorRectangle >= 1.5 && factorRectangle <= 1.53)
            description = sprintf("rectangle, b = %.3f, h = %.3f", b,
 h);
            factors.shape = [factors.shape; description];
            factors.area = [factors.area; area];
            factors.factorOfSafety = [factors.factorOfSafety;
 factorRectangle];
```

```
            factors.efficiency = [factors.efficiency; factorRectangle/
area];
        end
    end
end

factors.shape = [factors.shape; ""];
factors.area = [factors.area; 0];
factors.factorOfSafety = [factors.factorOfSafety; 0];
factors.efficiency = [factors.efficiency; 0];
```

Hollow Square

```
for k = 1:4*maxDim % 1/4 in to 12 in length
    for l = 1:4*maxDim % 1/4 in to 12 in height
        for m = 1:4*(maxDim/2) % 1/4 in to 6 in wall thickness
            b = k/4;
            h = l/4;
            t = m/4;

            if(t >= b/2 || t >= h/2)
                break;
            end

            b1 = t;
            b2 = b - (2*t);
            h1 = h;
            h2 = t;

            area1 = b1*h1;
            area2 = b2*h2;
            area = 2*area1 + 2*area2;

            inertia = ((h1^4)/12)-(((h1-2*h2)^4)/12);

            x = maxX(area);

            sigmaBend = bendingStress(moment(area, 0), h/2, inertia);
            factorHollow = factorOfSafety(sigmaYield, sigmaBend);

            if(factorHollow >= 1.5 && factorHollow <= 1.53)
                description = sprintf("hollow square, b = %.3f, h =
 %.3f, t = %0.3f", b, h, t);
                factors.shape = [factors.shape; description];
                factors.area = [factors.area; area];
                factors.factorOfSafety = [factors.factorOfSafety;
 factorHollow];
                factors.efficiency = [factors.efficiency;
 factorHollow/area];
            end
        end
    end
end
```

```
factors.shape = [factors.shape; ""];
factors.area = [factors.area; 0];
factors.factorOfSafety = [factors.factorOfSafety; 0];
factors.efficiency = [factors.efficiency; 0];
```

I-Beam

```
for k = 1:4*maxDim % 1/4 in to 12 in length
    for l = 1:4*maxDim % 1/4 in to 12 in height
        for m = 1:4*maxDim % 1/4 in to 12 in middle thickness
            for n = 1:4*(maxDim/2) % 1/4 in to 6 in flange thickness
                b = k/4;
                h = l/4;
                d = m/4;
                t = n/4;

                if(t >= b/2 || t >= h/2 || d >= b)
                    break;
                end

                b1 = d;
                b2 = b;
                h1 = h - (2*t);
                h2 = t;

                area1 = b1*h1;
                area2 = b2*h2;
                area = area1 + 2*area2;

                inertia = ((h2*h1^3)/12)+((b1*(h1+2*h2)^3-h1)/12);

                x = maxX(area);

                sigmaBend = bendingStress(moment(area, 0), h/2,
 inertia);
                factorIBeam = factorOfSafety(sigmaYield, sigmaBend);

                if(factorIBeam >= 1.5 && factorIBeam <= 1.53)
                    description = sprintf("I-beam, b = %.3f, h = %.3f,
 t = %0.3f, d = %0.3f", b, h, t, d);
                    factors.shape = [factors.shape; description];
                    factors.area = [factors.area; area];
                    factors.factorOfSafety = [factors.factorOfSafety;
 factorIBeam];
                    factors.efficiency = [factors.efficiency;
 factorIBeam/area];
                end
            end
        end
    end
end
```

```
factors.shape = [factors.shape; ""];
factors.area = [factors.area; 0];
factors.factorOfSafety = [factors.factorOfSafety; 0];
factors.efficiency = [factors.efficiency; 0];
```

T-Beam

```
for k = 1:4*maxDim % 1/4 in to 12 in length
    for l = 1:4*maxDim % 1/4 in to 12 in height
        for m = 1:4*maxDim % 1/4 in to 12 in middle thickness
            for n = 1:4*maxDim % 1/4 in to 12 in flange thickness
                b = k/4;
                h = l/4;
                d = m/4;
                t = n/4;

                if(d >= b || t >= h)
                    break;
                end

                b1 = d;
                b2 = b;
                h1 = h-t;
                h2 = t;

                area1 = b1*h1;
                area2 = b2*h2;
                area = area1 + area2;

                centroid = (b1*b2*h1 +(1/2)*(b1*b2^2+b2*h1*h1^2))/
(b1*b2+h1*b2);

                inertia = (1/3)*(b2*centroid^3+b1*(h1+b1-centroid)^3-
(h1-b2)*(h1-centroid)^3);

                x = maxX(area);

                sigmaBend = bendingStress(moment(area, 0), centroid,
 inertia);
                factorTBeam = factorOfSafety(sigmaYield, sigmaBend);

                if(factorTBeam >= 1.5 && factorTBeam <= 1.53)
                    description = sprintf("T-beam, b = %.3f, h = %.3f,
 t = %0.3f, d = %0.3f", b, h, t, d);
                    factors.shape = [factors.shape; description];
                    factors.area = [factors.area; area];
                    factors.factorOfSafety = [factors.factorOfSafety;
 factorTBeam];
                    factors.efficiency = [factors.efficiency;
 factorTBeam/area];
                end

            end
```

```
        end
    end
end

%allFactors = struct2table(factors);

shapeChange = find(factors.shape == "");

circle = struct('shape',factors.shape(1:
(shapeChange(1)-1)),'area',factors.area(1:
(shapeChange(1)-1)),'FoS',factors.factorOfSafety(1:
(shapeChange(1)-1)),'efficiency',factors.efficiency(1:
(shapeChange(1)-1)));
rectangle = struct('shape',factors.shape((shapeChange(1)+1):
(shapeChange(2)-1)),'area',factors.area((shapeChange(1)+1):
(shapeChange(2)-1)),'FoS',factors.factorOfSafety((shapeChange(1)+1):
(shapeChange(2)-1)),'efficiency',factors.efficiency((shapeChange(1)+1):
(shapeChange(2)-1)));
hollow = struct('shape',factors.shape((shapeChange(2)+1):
(shapeChange(3)-1)),'area',factors.area((shapeChange(2)+1):
(shapeChange(3)-1)),'FoS',factors.factorOfSafety((shapeChange(2)+1):
(shapeChange(3)-1)),'efficiency',factors.efficiency((shapeChange(2)+1):
(shapeChange(3)-1)));
IBeam = struct('shape',factors.shape((shapeChange(3)+1):
(shapeChange(4)-1)),'area',factors.area((shapeChange(3)+1):
(shapeChange(4)-1)),'FoS',factors.factorOfSafety((shapeChange(3)+1):
(shapeChange(4)-1)),'efficiency',factors.efficiency((shapeChange(3)+1):
(shapeChange(4)-1)));
TBeam =
 struct('shape',factors.shape((shapeChange(4)+1):end),'area',factors.area((shapeCh

circleBeamResult = struct2table(circle);
rectangleBeamResult = struct2table(rectangle);
hollowBeamResult = struct2table(hollow);
IBeamResult = struct2table(IBeam);
TBeamResult = struct2table(TBeam);

circleBest = find(circle.area == min(circle.area));
rectangleBest = find(rectangle.area == min(rectangle.area));
hollowBest = find(hollow.area == min(hollow.area));
IBeamBest = find(IBeam.area == min(IBeam.area));
TBeamBest = find(TBeam.area == min(TBeam.area));

circleFinal = circleBeamResult(circleBest,:)
rectangleFinal = rectangleBeamResult(rectangleBest,:)
hollowFinal = hollowBeamResult(hollowBest,:)
IBeamFinal = IBeamResult(IBeamBest,:)
TBeamFinal = TBeamResult(TBeamBest,:)


circleFinal =

  1×4 table
```

shape	area	FoS	efficiency
"circle, r = 2.875"	25.967	1.5075	0.058054

rectangleFinal =

1×4 table

shape	area	FoS	efficiency
"rectangle, b = 2.500, h = 11.250"	28.125	1.5006	0.053355

hollowFinal =

1×4 table

shape	area	FoS	efficiency
"hollow square, b = 1.750, h = 8.500, t = 0.750"	13.125	1.5265	0.1163

IBeamFinal =

1×4 table

shape	area	FoS	efficiency
"I-beam, b = 2.000, h = 12.000, t = 0.750, d = 1.750"	21.375	1.5134	0.070804

TBeamFinal =

1×4 table

shape	area	FoS	efficiency

```
    "T-beam, b = 1.250, h = 4.750, t = 0.250, d = 0.250"    1.4375
  1.5206      1.0578
```

Published with MATLAB® R2021a