Project Report

Project -2: Wing-Spar design

Unique ID: 4306

1 Executive summary

The objective of this work is to develop tools to design and optimize the shape of a spar of a wing, which will be used in building a telecommunication airplane. Specifically, the aim is to design the annulus circular beam with a minimum weight that can withstand the load of 2.5g during the maneuver and meet the manufacturing constraints. The length of the beam has been determined to be 7.5 m by aerodynamics analysis, whereas the thickness of the beam should remain greater than 0.0025 m throughout the length due to manufacturing constraints. The material is assumed to have constant density and bear the load linearly.

The numerical computational optimization method has been leveraged here to implement the analysis as I need to find the optimal radii out of myriad radii that are time-consuming for analytical or experimental methods. The radii are a function of distance from the tip of the wing and must satisfy the load/stress constraints. The bar is modeled using Euler-Bernoulli beam theory with some assumptions on planer symmetry, normality, strain energy, linearization, material, and cross-sectional variation. The beam geometry was discretized and the differential equations were solved using the finite element method satisfying the boundary conditions. After setting up the problem, I used the "active-set" algorithm in the "*fmincon*" function in Matlab® where the gradients computed for the objective and the nonlinear constraints using a complex step method were provided to *"fmincon"* for determining the design variables i.e. the radii.

The nominal spar mass was found to be 13.257 Kg and the optimal spar mass was found to be ~ 4.877 Kg which is 63.208% lower than the nominal mass. The optimal mass was seen to decrease gradually with each iteration during the optimization as expected for convergence. The performance was found to be increasing with an increase in the number of finite elements which means the accuracy of the optimization method improves with an increase in discretization. This was a required behavior to check if the program and model were behaving as it is supposed to. Note that, the gradients computed with the complex step method were verified with the "*fmincon*" gradient check, which validated the code, before using them for optimization.

The end product of this project is a computer code that can predict an optimal mass and configuration/geometry for a wing spar given only a few constraints (minimal thickness, geometrical dimensions, etc.). The design optimization method facilitated a comparison of different configuration and select the optimal one.

Contents

2 Optimization Problem statement

The objective is to minimize the spar weight of the given wing subjected to the maneuver and geometric constraints only by changing the shape of the annulus beam. The length of the spar, material to be used and the maneuvering load have been determined using aerodynamics. Thus, the inner and outer radii at each point of the annulus spar are the design variables. Following constraints must be satisfied by the optimal spar.

Constraints:

The spar can take any shape and mass, provided

- The thickness is greater than 0.0025m throughout the spar;
- The maximum outer radii should be less than or equal to 0.05m, and
- The minimum inner radii should be greater than or equal to 0.01m.

3 Analysis Method

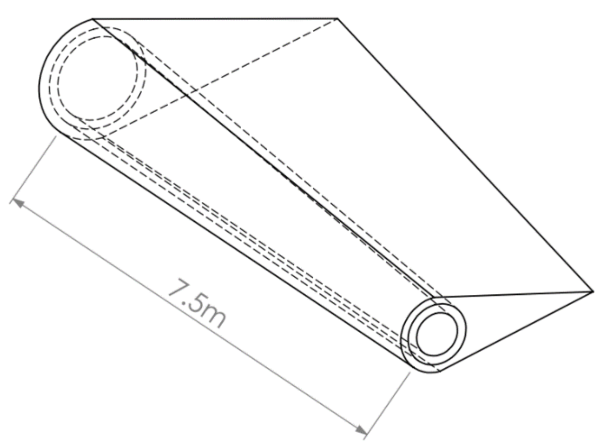


Figure 1: Illustration of Geometry of one of the wing-spar inside the wing.

The following design parameters and specifications of the wing have been predetermined by the aerodynamics and manufacturing specialist and they remain constant throughout the analysis:

- wing semi-span: 7.5 m, which is also the length of the spar.
- spar cross-section shape: circular annulus.
- material: Carbon fiber composite, with a density of 1600 kg/m^3, Young's modulus of 70 GPa, and ultimate tensile/compressive strength of 600 MPa.
- manufacturing constraints: The inner and outer radii of the annulus cannot be less than 2:5 mm apart, the inner radius cannot be smaller than 1 cm, and the outer radius cannot be larger than 5 cm.
- aircraft operational weight: The total mass of the aircraft will be 500 kg, including the spar.
- loading: At a 2:5 g maneuver (i.e. a maneuver at which the total force on the spar is 2.5 times the weight of the aircraft), the force distribution in the span wise direction will have an approximately linear distribution, with a maximum load at the root and zero loads at the tip.

3.1.1 Euler-Bernoulli Beam Theory:

Euler–Bernoulli beam theory is a simplification of the linear theory of elasticity which provides a means of calculating the load-carrying and deflection characteristics of beams. The Euler–Bernoulli equation[1] describes the relationship between the beam's deflection and the applied load as below:

$$\frac{d^2}{dx^2}\left(EI_{yy}\frac{d^2w}{dx^2}\right) = q, \qquad \forall x \epsilon [0, \mathrm{L}]$$

Where,

W ➔ is the vertical displacement in the z direction (in m)

q(x) ➔ is the applied distributed load, in other words, a force per unit length. (in N/m)

E ➔ is the elastic or Young's modulus (in Pascal) and

I_{yy}➔ is the second moment of area with respect to the y-axis (m^4), it must be calculated with respect to the axis which is perpendicular to the applied loading and which passes through the centroid of the cross-section.

Explicitly for an annulus circular beam, the second moment of area is given by:

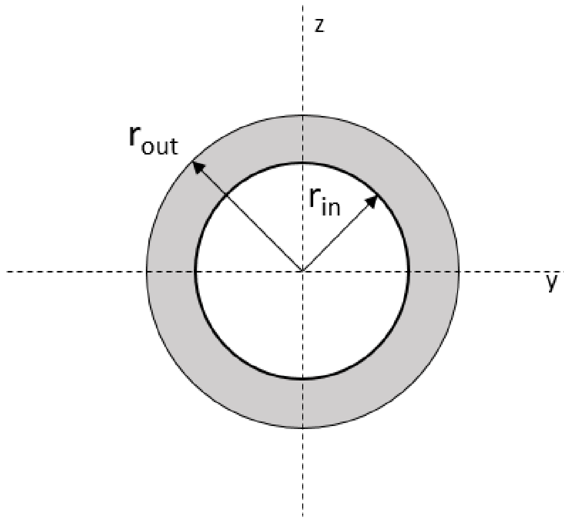


Figure 2: The annulus cross-section of the spar

$$I_{yy} = \iint z^2 dzdy$$

$$I_{yy} = \frac{1}{4}\pi({r_{out}}^2 - {r_{in}}^2)$$

The following assumptions were made for applying the Euler-Bernoulli beam theory:

- planar symmetry: longitudinal axis is straight, and the cross section of the beam has a longitudinal plane of symmetry
- cross-section variation: cross-section varies smoothly
- normality: plan sections that are normal to the longitudinal plane before bending and remain normal after bending
- strain energy: internal strain energy accounts only for bending moment deformations
- linearization: deformations are small enough that nonlinear effects are negligible
- material: the material is assumed to be elastic and isotropic

Applied loads may be represented either through boundary conditions or through the function *q(x)* which represents a distributed load.

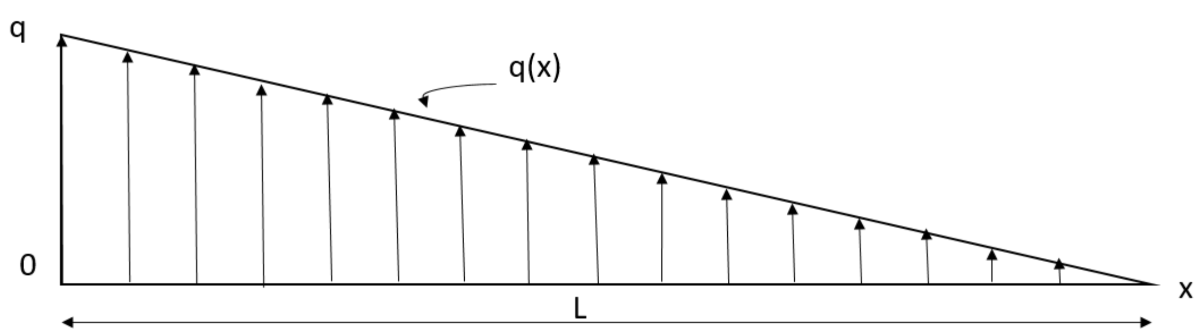


Figure 3: Linear loading of the beam

Area under the loading condition:

$$\int_0^L q(x) = 2.5 * 250 * 9.8 = W0$$

In this case, the q(x) can be expressed mathematically as:

$$q(x(i)) = \frac{W0}{L} * (L - x(i))$$

Where W0 is loading at the root of the spar.

The beam equation contains a fourth-order derivative in x. To find a unique solution *w(x)* we need four boundary conditions. The support or displacement boundary conditions are used to fix values of displacement *w* and rotations *dw/dx* on the boundary. Such boundary conditions are also called Dirichlet boundary conditions. Here, at the root end of the spar, there cannot be any displacement or rotation of the beam. This means that at the left end both deflection and slope are zero. Since no external bending moment is applied at the free end of the beam, the bending moment at that location is zero. In addition, if there is no external force applied to the beam, the shear force at the free end is also zero. Mathematically these boundary conditions are expressed below:

$$w(x = 0) = 0, \qquad \frac{d^2}{dx^2}(x = L) = 0$$

$$\frac{dw}{dx}(x = 0) = 0, \quad \frac{d^3}{dx^3}(x = L) = 0$$

$$\sigma_{xx}(x) = -ZmaxE\frac{d^2w}{dx^2}$$

Where Z_{max} is the maximum height of the cross-section (in this case the outer radius of the circle)

Here, I considered the spar for only one of the two wings. Both wings will support the total loading on the whole aircraft, so with symmetric loading, it can be assumed that one spar will support half the total loading. So I will use 250 kg in the calculation.

3.1.2 Complex step method for gradient:

Complex step differentiation is a technique that employs complex arithmetic to obtain the numerical value of the first derivative of a real valued analytic function of a real variable, avoiding the loss of precision inherent in traditional finite differences. The catch is that it involves complex arithmetic. For small *h,* it is expressed as below.

$$\frac{\partial f}{\partial x} = Imag(\frac{f(x + ih)}{h})$$

Where f is the function to be differentiated.

3.1.3 Finite Element Discretization:

The FEM was used for solving the above differential equations with the boundary value problems. To solve a problem, the FEM subdivides a large system into smaller, simpler parts that are called finite elements. I will discretize the Euler-Bernoulli beam equation using the finite-element method.

- solution is represented using Hermit-cubic shape functions
- finite-element equations result from the minimization of the potential energy functional

As the shape of the bar varies for each iteration, we need to find the mass of each iteration for which the cross-section of each finite element was integrated using the "*trapz.m*" Matlab function, which computes the approximate integral via the trapezoidal method, to obtain the volume which was multiplied by density to get the mass.

3.2 Design variables

In this analysis, the heights or radii from the center to the outer surface and inner surface of the annulus beam are the design variables. The design variables are subject to some geometric constraints that the optimal design must satisfy. One of the geometric constraints is on the thickness of the bar; the distance between the outer radii and inner radii must remain greater than the minimum thickness i.e. 0.0025 m at all the nodal points of the beam.

$$\text{i.e. } r_{out} - r_{in} \geq 0.0025 \text{ m}$$

Moreover, the outer radii must be greater than or equal 0.0125 m and smaller than or equal 0.05 m and the inner radii must be greater than or equal 0.01m and smaller than or equal 0.0475m.

$$\text{i.e. } 0.0125 \leq r_{out,\, i} \leq 0.05 \text{ m}$$

$$0.01 \leq r_{in,\, i} \leq 0.0475 \text{ m}$$

4 Results

The code was validated with the nominal values, where the nominal mass and configuration was obtained for a cylindrical pipe as show below:

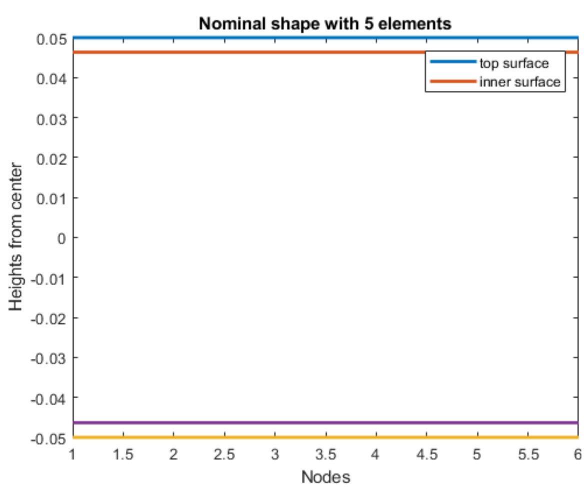


Figure 4: Nominal shape of the spar. Note: the constant radius at each of the nodal points.

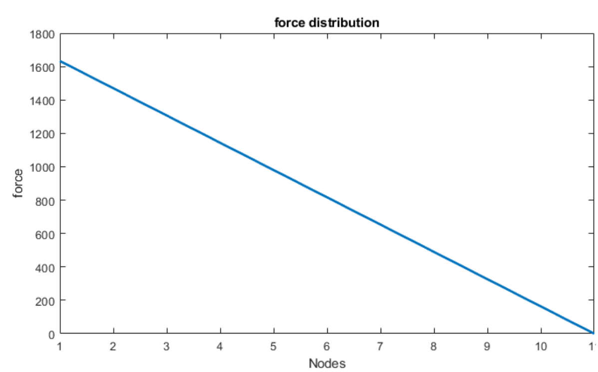


Figure 5: force distribution from root to the tip of the spar

The iteration history for 10 elements case is shown below to see how the variables changes during optimization. The nominal mass of the spar was found with a cylindrical shape bar with outer radii of 0.05 and inner radii of 0.01m. The nominal mass was found to be 13.259 Kg. The spar mass starts from the nominal mass i.e. 13.259 Kg and reaches the optimal weight i.e. 4.877 Kg. In other words, the optimal weight was found to be 63.208 % of the nominal mass.

Iterations	F-count	Spar mass (kg)	Max constraint	Line search step length	Directional derivative	First-order optimality Procedure
0	15	13.2579	0			
1	32	8.01196	0	0.5	-100	285
2	49	6.38439	0	0.5	-66.3	158
3	66	5.69308	0	0.5	-61.5	111
4	82	4.87599	0.02008	1	-72	25.4
5	98	4.87768	0.0004188	1	4.1	6.84
6	114	4.87784	1.01E-07	1	29.9	0.085

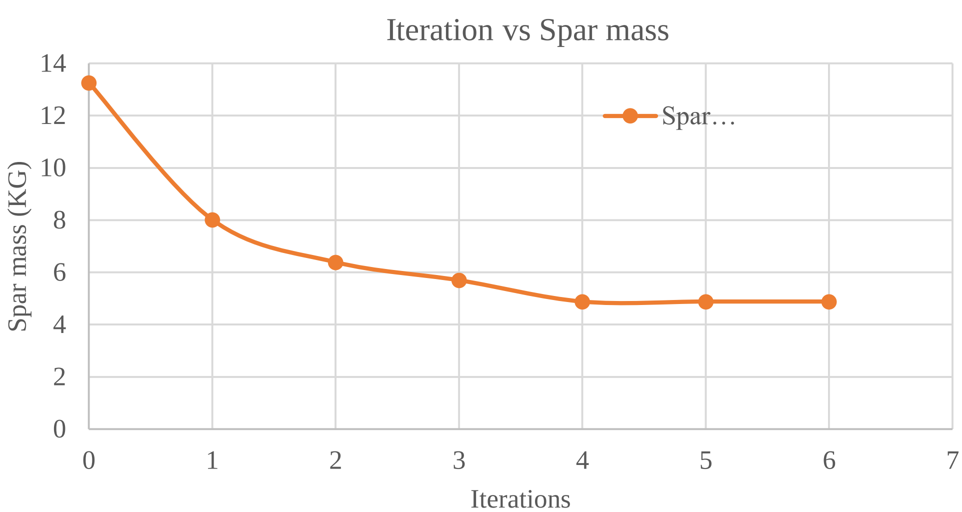


Figure 6: Minimization of spar mass with iteration.

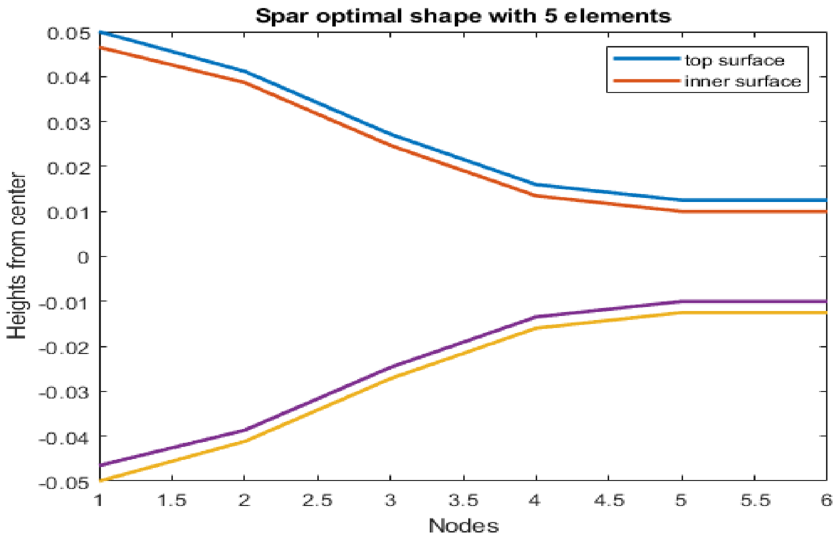


Figure 7: Optimal shape of the spar with 5 discrete elements.

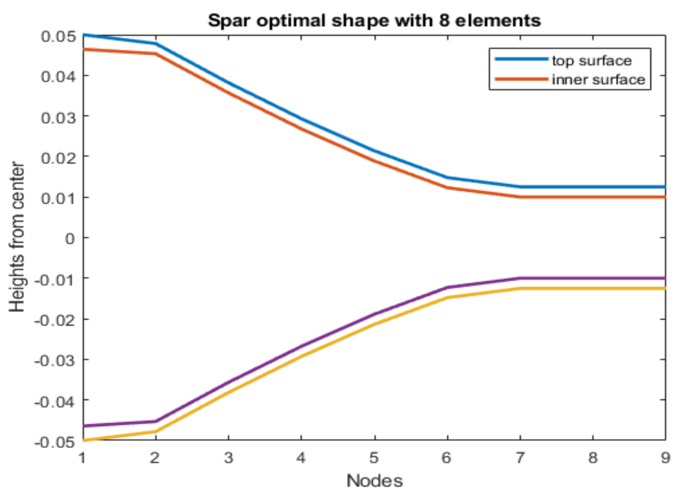
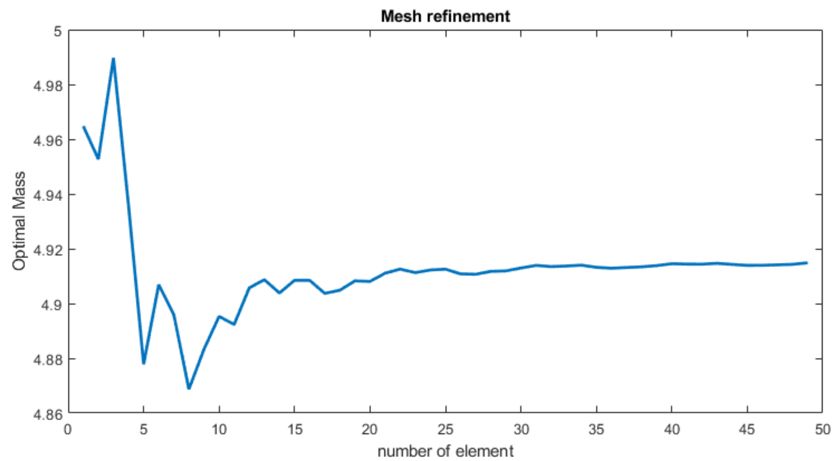


Figure 8: Optimal shape of the spar with 8 discrete elements.

Through optimization, the minimizer (i.e. coefficients) was obtained and the optimal shape was determined. Figure 7 shows the heights/radii at different nodal points along the bar. The largest radii i.e. 0.05m is found to be at the root of the spar and the smallest radii is found to be 0.01m at the tip.



(a)

Figure 9: variation of optimal mass of the spar with respect to the number of discrete elements. NOTE: the radii for each of the experiments were initialized with the same values to maintain consistency.

The optimal weight decreased significantly after four elements and the minimum weight was found for 8 number of elements. However, the shape of the spar with 8 elements (see figure 7) has some sharp cut which produces stress concentration making it weak and such shape increase the complexity in manufacturing. Nevertheless, the mass was relatively the same for more than 5 elements with a maximum difference of 0.02 kg. Thus the shape with 5 elements should be preferred.

The nodal displacement for the inner and outer radii was computed which is shown below in figure 10. Note the displacement on the tip of the spar is comparable to the length of the spar.

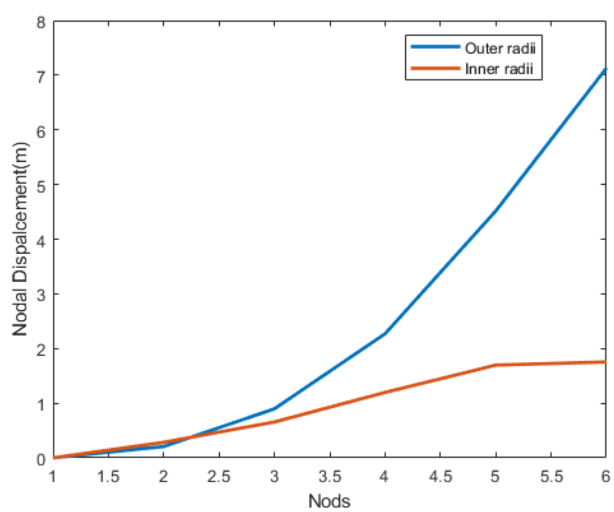


Figure 10: Nodal displacement for 5 element

5 Discussion and Conclusion

I first validated the program and model by comparing the output with the cylindrical bar. The nominal spar mass was found to be 13.259 kg (see figure 4 and 6). Then the optimization was performed to minimize the weight of the spar. The design variables were initialized with the nominal values. The mass was seen decreasing with each iteration and the optimal mass was found to be i.e. 4.877 kg (see figure 6). The optimal mass was 63.208% less than the nominal mass. Note that the optimal shape of the spar is like a truncated annulus cone with larger radii at the root of the bar which aligns with the load-bearing capacity required due to the linear loading. Comparing the shape for 5 and 8 (see figure 7 and 8) elements helped to select the spar with marginally higher mass but with less shape complexity to help manufacturing.

Based on the Euler-Bernoulli model, the optimization analysis shows that the optimal mass determined using finite element equations corresponds to the truncated cone-like shape. The shape of the spar doesn't have any sharp edges or wiggle shape which could bring manufacturing difficulties and thus the shape can be manufactured without needing any sophisticated machine. The complex step method helped to eliminate all the limitation of the widely used difference method like the "step-size dilemma", truncation error and complex. In contrasts, the complex step method allowed simple and fast implementation.

I performed a sensitivity analysis in which I compared the effect of the number of elements on the optimal mass. The number of elements was increased from 1 to 50 and the analysis was performed with the same initial design variable values (see figure 9(a)). The mass decreased rapidly with an increase in the number of elements up to 5 and then it plateaus. This indicates a fairly good number of elements for achieving the desired mass with less computational expenses. Beyond 5 number of element, the mass increased slightly but the difference is not significant. Notably, the displacement at the tip of the spar was comparable to the spar length (see figure 10) which could be due the low stiffness of the carbon fiber and probably due to higher load.

Limitations:

A compressive analysis is required to get an understanding of the choice of the algorithm i.e. sqp or the active-set. Further, the marginal increase in mass after 5 numbers (see figure 9) of elements needs to be analyzed in more detail. Due to the several assumptions made in the Euler-Bernoulli's beam theory during analysis, the final model needs to be experimentally tested.

6 Appendix

NOTE: The code that were provided by Prof.Hicken, which were not modified at all, are not shown here.

Codes:

Main.m

```
% Main function to do the experiments.
clear all; clc; close all; format long;
% constants
rho     = 1600;                         % kg/m^3
E       = 70*10^9;                      % pascal
U       = 600*10^6;                     % pascal
Weight_plane = 500;                     % Kg
L       = 7.5;                          % spar length (m)
Nelem   = 6;                           % number of elements
Nx      = Nelem+1;                      % number of nodes or x
x       = linspace(0,7.5,Nx)';
p       = 1;
h0      = zeros(2*(Nelem+1),1);
for i=1:(Nelem+1)                       % initilization
    h0(p)    = 0.05;                    % height of outer heights
    h0(p+1)  = 0.04635;                 % height of inner heights
```

```
    p       = p+2;
end
% computes nominal
force    = forceDist(x);                % compute nominal force displacements
figure()
plot(force,'linewidth',2);
title('force distribution')
xlabel('Nodes')
ylabel('force')

[Nominal_Spar_weight, grad_nominal]=  obj(h0,x);

fprintf('\n nominal spar weight: %f',Nominal_Spar_weight);

Iyy = AreaMomentIyy(h0);
%%
type = 'nominal'; algo = 'active-set';

[h_coef_minimizer,fval] = optimize(x, h0, L, E, Nx, Nelem, U, algo,type);

optimal_mass    = CalcSparWeight(rho,x,h_coef_minimizer);

percent         = (Nominal_Spar_weight- optimal_mass)/Nominal_Spar_weight*100;
fprintf('\n The optimal mass is %f ', optimal_mass);
fprintf('\n The optimal mass is %f %% of Nominal', percent);

% refinement effect
% mesh_refine(L, E,U, algo,rho);

h = reshape(h_coef_minimizer,2,[])';
optimal_thickness = h(:,1)-h(:,2);
for i = 1:numel(optimal_thickness)  % check if the thickness is satisfied.
    if optimal_thickness(i) < 0.0025
        fprintf('Thickness is above the minimum thickness');
    end
end

figure()
plot(h(:,1),'linewidth',2 ) ; hold on
plot(h(:,2), 'linewidth',2) ; hold on
plot(-h(:,1),'linewidth',2) ; hold on
plot(-h(:,2),'linewidth',2)
legend('top surface','inner surface')
xlabel('Nodes')
ylabel('Heights from center')
ttl = sprintf('Spar optimal shape with %d elements',Nelem);
title(ttl)
```

CalcSparWeight.m

```
function [mass] = CalcSparWeight(rho,x,h)
% Computes weight of the spar member
% Inputs:
%   x - nodes along the x-axis to compute the lenght of the each elements
%   h - height at each of the nodal element
%   Nelem - number of finite elements to use
% Outputs:
%   mass - total mass of one spar
%
% Assumes the beam is symmetric about the y axis
%---------------------------------------------------------------------------
```

```
Nelem = size(h,1)/2-1;                   % number of elements

H = reshape(h,[2,Nelem+1])';             % 1st col is outer heights, 2nd col is inner heights.
nodal_area = pi*(H(:,1).^2-H(:,2).^2);

mass = rho*trapz(x,nodal_area);

end
```

AreaMomentIyy.m

```
function Iyy = AreaMomentIyy(h)
% returns the Iyy of the pipe
% Inputs
%     h: height of the outer and inner surface of the spar
% Outputs:
%     Iyy : Area moment of Inertia of the spar

n   = size(h,1)/2;
p   = 1;
Iyy = zeros(n,1);
lowerbound = 0.00000000000001;            % set this value to the negative Iyy(i)
for i=1:n
    Iyy(i) = pi/4*(h(p)^4 - h(p+1)^4);
    p      = p+2;
    if Iyy(i) < 0                         % to remove the negative area moments.
        Iyy(i) = lowerbound;
    end
end

end
```

forceDist.m

```
function F = forceDist(x)
% returns for distribution of the spar
% Inputs
%     x: nodal points
% Outputs
%     F: force distribution

W0 = 1633.33;            % weight/length
L = 7.5;                 % m (length of spar)
n = size(x,1);
F = zeros(n,1);          % force
for i=1:n
    F(i) = W0/L*(L-x(i));
end

end
```

Obj.m

```
function [Spar_weight,grad_objective] = obj(h,x)
```

```
% Returns the sparweigth and gradients
%Input
%   h: design variables
%   x: nodal points
% Outputs
%     Spar_weight: computed spar weight
%     grad_objective: Gradient of objective w.r.t desing variable using complex step

rho              = 1600;
Spar_weight      = CalcSparWeight(rho,x,h);
dh               = 1e-60;
grad_objective   = zeros(numel(h),1);

for i=1:numel(h)
    hc                  = h;
    hc(i)               = hc(i)+ complex(0, dh);
    grad_objective(i)   = -imag(CalcSparWeight(rho,x,hc))/dh;                   % gradient of
objective
end

end
```

Optimize.m

```
function [h_coef_minimizer,fval] = optimize(x, h0,L,E, Nx,Nelem,U,algo, type)
%Runs the optimization algorithm using complex step method
% Inputs:
%     x: points along the horizontal
%     h0: intial a design variable coefficients % should be in form:
%     [r_out_1, r_in_1,r_out_2, r_in_2] to match the Aineq coefficient form
%     (need to check again?)
%     algo: type of algorithm to use for the optimization.
%     Nx: number of nodes(or x)
% Outputs:
%     h_coef_minimizer: nodal height; (design variable)
%       fval = Objective value

Aineq        = zeros(Nx,2*Nx);
bineq        = zeros(Nx,1);
lb           = zeros(2*Nx,1);
ub           = zeros(2*Nx,1);
k            = 1;

for i = 1:Nx
  Aineq(i,k)  = -1;      % this coefficient corresponds to variable h_out
  lb(k)       = 0.0125;       % r_out lb
  ub(k)       = 0.05;         % r_out up
  k           = k+1;
  lb(k)       = 0.01;         % r_in lb
  ub(k)       = 0.0475;       % r_in up

  Aineq(i,k)  = 1;       % this coefficient corresponds to variable a_in
  k           = k+1;
  bineq(i)    = -0.0025;   % negative to accomodate fmincon
end

force = forceDist(x);   %Compute force distribution

%%
objective = @(h)obj(h,x);                                        % creates the objective function
to minimize

stress_distribution = stress_dist(L,E,Nelem,h0,U,force);
if strcmp(type, 'Not_refine')
    figure()
    plot(stress_distribution,'linewidth',2)
    title('Initial stress distribution')
    xlabel('Nodes')
```

```
    ylabel('stress (N/m)')
end

nonlcon = @(h) stress_dist(L,E,Nelem,h,U,force);                  % impose non-linear constraints.

options = optimoptions('fmincon','SpecifyObjectiveGradient',
true,'Display','iter','Algorithm',algo, 'CheckGradients', true );  %'Display','final-detailed',
set the option for chaning the algorithm
[h_coef_minimizer,fval] = fmincon(objective,h0,Aineq,bineq,[],[],lb,ub,nonlcon,options); %nonlcon

%plot
% if isequal(type,'Nominal') || isequal(type,'Optimal')
%     plot_(h,x,type,h_coef_minimizer)
% end
End
```

mesh_refine.m

```
% Runs the mesh_refine algorithm by increaseing the number of elements

function [] = mesh_refine(L, E,U, algo,rho)

num_mesh    = 50;       % number of mesh; +2
OptimalMass = zeros(num_mesh-1,1);
type        = 'refine';

for i =2:num_mesh       % Loop for mesh refinment
    %clc;
    fprintf('Running refinment for %d elements',i);
    Nelem   = i;
    h0      = zeros(2*(Nelem+1),1);
    Nx      = Nelem+1;
    x       = linspace(0,7.5,Nx)';
    p       =1;
    for j=1:(Nelem+1)               % initilization
        h0(p)   = 0.05;             % height of outer heights
        h0(p+1) = 0.04635;          % height of inner heights
        p       = p+2;
    end
    [h_coef_minimizer,~] = optimize(x, h0, L, E, Nx, Nelem, U, algo, type);
    OptimalMass(i-1)       = CalcSparWeight(rho,x,h_coef_minimizer);
end

figure()
plot(OptimalMass,'linewidth',2)
title('Mesh refinement')
xlabel('number of element')
ylabel('Optimal Mass')
end
```