Project 2 – MANE 6710 Numerical Design Optimization

Unique ID: 5284

Table of Contents

List of Figures

Introduction

The main objective of this project is to design the cross-sectional profile of a spar of an aircraft wing. The idea is to arrive at a design which leads to the least weight possible for the structure given certain design parameters. The optimal design that comes as a result should not fail under the loading conditions and in fact it must be manufacturable. Additionally, the loading conditions are not deterministic, and it is perturbed by uncertainties.

Choice of Design Variable

With the cross section fixed to be a circular annulus, the most obvious choice of design variables are the Radii of the annulus along the length of the spar. The parameterization of design variable chosen in this project is as follows: $R = \{R_{out}^1\ R_{in}^1\ R_{out}^2\ R_{in}^2 \ldots . R_{out}^n\ R_{in}^n\}' - \{R_{out}^i, R_{in}^i\}\ i{:}\ 1(1)n$

The length of the spar can be discretized into n elements (N_{elem}) and with a pair of design variables $\{R_{out}^i, R_{in}^i\}$ at each node as shown in Figure 1.

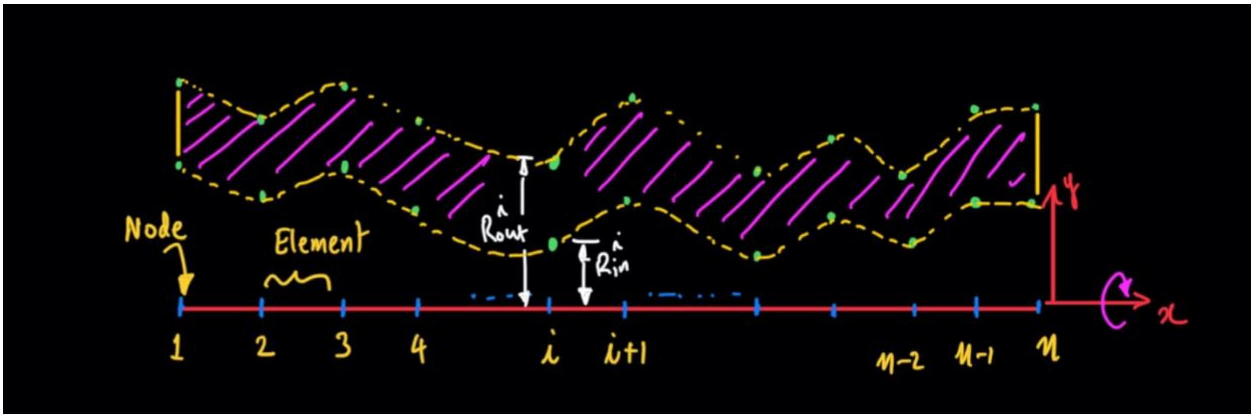

Figure 1. Cross- sectional Profile of the spar

Objective Function and Constraints

The objective of this project is to optimize the design of the spar's cross-section that has the least weight possible.
This cross-sectional profile needs to adhere to certain design parameters such as:

1. Semi Span Length – 7.5 m
2. Shape of Cross section – Circular Annulus
3. Material – Carbon Fiber composite – {Density $\rho = 1600\ kg/m^3$;Young's modulus $E = 70\ GPa$; Yield Compressive/Tensile stress $\sigma_{max} = 600\ MPa$}
4. Aircraft operational weight – 500 kg
5. Nominal Loading – 2.5 g with force distribution linearly varying along the length (maximum load at the root and no load at the tip)
6. Loading with uncertainty – Uncertain loading is modelled and described in section **"Loading with uncertainty"**.

With these parameters, the objective function will be the Mass or the Weight of the spar which can be calculated as $m = \rho V\ kg; V - volume\ (m^3), \rho - density\ (kg/m^3)$. Hence this design problem can now be formulated as:

$$\min_R \rho V(R) \qquad \text{Equation 1}$$

This minimization problem is subject to constraints that stem from the manufacturing side of the spar, and they are:

1. $c_1(R) = R_{out} \le 5cm$ along the entire length of the spar
2. $c_2(R) = R_{in} \ge 1cm$ along the entire length of the spar
3. $c_3(R) = R_{out} - R_{in} \ge 2.5mm$ along the entire length of the spar
4. $c_4(R) = E(\sigma) + 6\sqrt{\sigma^2 - E(\sigma)^2} \le \sigma_{max}(yield\ stress)$ – this condition arises

MATLAB's in-built optimizer $fmincon(obj, a_0, A, b, Aeq, beq, lb, ub, Nonlinear, options)$ is used to perform the optimization of spar cross-section.

Calculation of Volume

Volume is generally calculated by the integral $\int_0^L A(x)\,dx$ where, $A(x)$ is the cross-section along the length of the spar. With very fine discretization of the length of spar into N_{elem}, this integral can be approximated as:

$$\sum_{i=1}^{N_{elem}} \frac{1}{2}\left\{\pi\left({R_{out}^{i}}^2 - {R_{in}^{i}}^2\right) + \pi\left({R_{out}^{i+1}}^2 - {R_{in}^{i+1}}^2\right)\right\}\delta x_i \qquad \text{Equation 2}$$

This calculates the average area of the cross-section on two adjacent node points and multiplies it with the distance between these two node points δx_i. This is a variational form of trapezoidal integration technique.

Calculation of Constraints

Setting up Linear Inequality constraints

Constraints $c_1(R), c_2(R), c_3(R)$ are set up as Linear Inequality constraints of the format $fmincon$ requires it in - $AR \le b$ where A is the matrix that contains the linear mapping between the design variables R and constraint values b.

Setting up Nonlinear constraints

This is the most important aspect of this project as the computation of normal stress at each cross-section requires the use of numerical methods like Finite Element Methods to calculate it. The Area moment of Inertia for each cross-section along the length is,

$$I_{yy}^{i} = \frac{1}{4}\pi({R_{out}^{i}}^{4} - {R_{in}^{i}}^{4}) \qquad \text{Equation 3}$$

Then, the nominal force which varies linearly along the length of the wing is calculated at each Node point as

$$\int_{0}^{L} f\, dx = \frac{2.5 * Operational\ Load(N)}{2}\ ; F_{nom}^{i} = f(1 - x^{i}/L)\ \ N \qquad \text{Equation 3}$$

The operational load considered in this problem is Operational Weight - $500kg\ \times 9.81\ m/s^2$

Loading with Uncertainty

Loading in this project has uncertainty of the form given by the following Equation 4.

$$f(x,\xi) = F_{nom}(x) + \delta_f(x,\xi)$$
$$\delta_f(x,\xi) = \sum_{n=1}^{n=4} \xi_n \cos\left(\frac{(2n-1)\pi x}{2L}\right) \qquad \text{Equation 4}$$

$$\xi_n \sim \mathcal{N}\left(0, \frac{F_{nom}(0)}{10n}\right)$$

This perturbed vector of forces are inputs to computing the normal stress at each nodal location and hence it is extremely important to capture the entire statistics of the constraint rather than just its behavior under nominal loading conditions. These perturbations could lead to both over loading and under loading conditions, and the optimized design should be able to handle this range of loading conditions that arise as a result of perturbations. The statistics of primary interest are the Estimated value of normal stress and its standard deviation. They can be computed using Equations 5,6, and 7.

$$\vec{\mu}_s(\vec{f},\xi_1,\xi_2,\xi_3,\xi_4) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(\vec{f},\xi_1,\xi_2,\xi_3,\xi_4) P_1(\xi_1;\vec{f},\sigma_1) d\xi_1\ P_2(\xi_2;\vec{f},\sigma_2)\ d\xi_2\ \ P_3(\xi_3;\vec{f},\sigma_3)\ d\xi_3\ \ P_4(\xi_4;\vec{f},\sigma_4) d\xi_4$$

$$s(f,\xi) - stress$$
$$\vec{f} - nominal\ force$$
$$\xi - perturbations$$
$$P(\xi;x,\sigma) - probability\ model\ for\ perturbations \qquad \text{Equation 5}$$
$$\sigma - standard\ deviation\ of\ perturbations$$

$$\vec{\mu}_{s^2}\left(\vec{f},\xi_1,\xi_2,\xi_3,\xi_4\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[s\left(\vec{f},\xi_1,\xi_2,\xi_3,\xi_4\right)\right]^2 P_1\left(\xi_1;\vec{f},\sigma_1\right)d\xi_1\,P_2\left(\xi_2;\vec{f},\sigma_2\right)d\xi_2\,P_3\left(\xi_3;\vec{f},\sigma_3\right)d\xi_3\,P_4\left(\xi_4;\vec{f},\sigma_4\right)d\xi_4$$

Equation 6

$$\vec{\sigma}_s = \sqrt{\left(\vec{\mu}_{s^2}\left(\vec{f},\xi_1,\xi_2,\xi_3,\xi_4\right) - \vec{\mu}^2\left(\vec{f},\xi_1,\xi_2,\xi_3,\xi_4\right)\right)}$$

Equation 7

$$\vec{\sigma}_s - standard\ deviation\ of\ stress$$

In this problem, the perturbations are modeled as Gaussian white noise given in Equation 4. On top of that, an oscillatory behavior is added to these perturbations using a cosine function along the length of the spar and it varies in amplitude scaled by these perturbations. These integrals are not trivial integrals, and it requires special integration techniques to carry them out. Perturbations to forces can be added and mean stress can be computed using Monte-Carlo sampling technique to keep the mathematics simple, but it drives the cost of the constraint up to a large extent. Hence stochastic collocation technique is preferred to perform the integration effectively, and as accurately as possible. The integrals reduce to this form for this 4-dimensional system:

$$\vec{\mu}_s(f) = \frac{1}{\sqrt{\pi}\sqrt{\pi}\sqrt{\pi}\sqrt{\pi}}\sum_{k_1=1}^{m} w^{(k_1)}\sum_{k_2=1}^{m} w^{(k_2)}\sum_{k_3=1}^{m} w^{(k_3)}\sum_{k_3=1}^{m} w^{(k_4)}\, s\left(\left\{\sqrt{2}\sigma_1\xi_1^{(k_1)}\cos\frac{\pi\vec{x}}{2L} + \sqrt{2}\,\sigma_2\xi_2^{k_2}\cos\frac{3\pi\vec{x}}{2L} + \sqrt{2}\sigma_3\xi_3^{k_3}\cos\frac{5\pi\vec{x}}{2L} + \sqrt{2}\sigma_4\xi_4^{k_4}\cos\frac{7\pi\vec{x}}{2L}\right\} + \vec{f}_{nom}\right)$$

Equation 8

$$\vec{\mu}_{s^2}(f) = \frac{1}{\sqrt{\pi}\sqrt{\pi}\sqrt{\pi}\sqrt{\pi}}\sum_{k_1=1}^{m} w^{(k_1)}\sum_{k_2=1}^{m} w^{(k_2)}\sum_{k_3=1}^{m} w^{(k_3)}\sum_{k_3=1}^{m} w^{(k_4)}\, s^2\left(\left\{\sqrt{2}\sigma_1\xi_1^{(k_1)}\cos\frac{\pi\vec{x}}{2L} + \sqrt{2}\,\sigma_2\xi_2^{k_2}\cos\frac{3\pi\vec{x}}{2L} + \sqrt{2}\sigma_3\xi_3^{k_3}\cos\frac{5\pi\vec{x}}{2L} + \sqrt{2}\sigma_4\xi_4^{k_4}\cos\frac{7\pi\vec{x}}{2L}\right\} + \vec{f}_{nom}\right)$$

Equation 9

$$\left(\xi_n^{k_i}, w^{k_i}\right) \equiv \left(Quadrature\ points\ of\ random\ variable\ \xi, Quadrature\ integration\ weights\right)$$

The nonlinear constraint is now computed as $c_4(R) = \vec{\mu}_s + 6\vec{\sigma}_s \le \sigma_{max}(yield\ stress)$. Incorporating these statistics into the constraint calculations results in a robust design which can handle uncertain loading conditions. These conditions are more likely to occur in practical scenarios.

Setting up Optimization Problem

With the objective and the constraints set up, it is now important to calculate accurate gradients of the objective and the constraints for successful optimization of the design. Complex step gradient method is used to compute the gradients of the objective and the inequality constraints. With a very small complex step $h = 10^{-60}$,the round off errors caused by the central difference approximation can be avoided.
Two functions, $Calc\_objGrad(X)$ and $Calc\_consJac(X, N_{nodes}, L, E, F, \sigma_{max})$ are used to compute the gradients of the objective function and the nonlinear constraint function.

Number of elements - 30
Initial point chosen - $[0.05 \quad 0.0415 \quad 0.05 \quad 0.0415 \quad 0.05 \cdots\cdots\cdots 0.05 \quad 0.0415]'_{62\times 1}$

Options chosen for $fmincon$ –

i. Choice of Algorithm – ‘Sequential Quadratic Programming (SQP)’
ii. Gradients – Provided and computed using Complex Step Algorithm

Initial weight of the spar was $29.320484235953565\ \boldsymbol{kg}$

Initial Design choice is shown in Figure 2

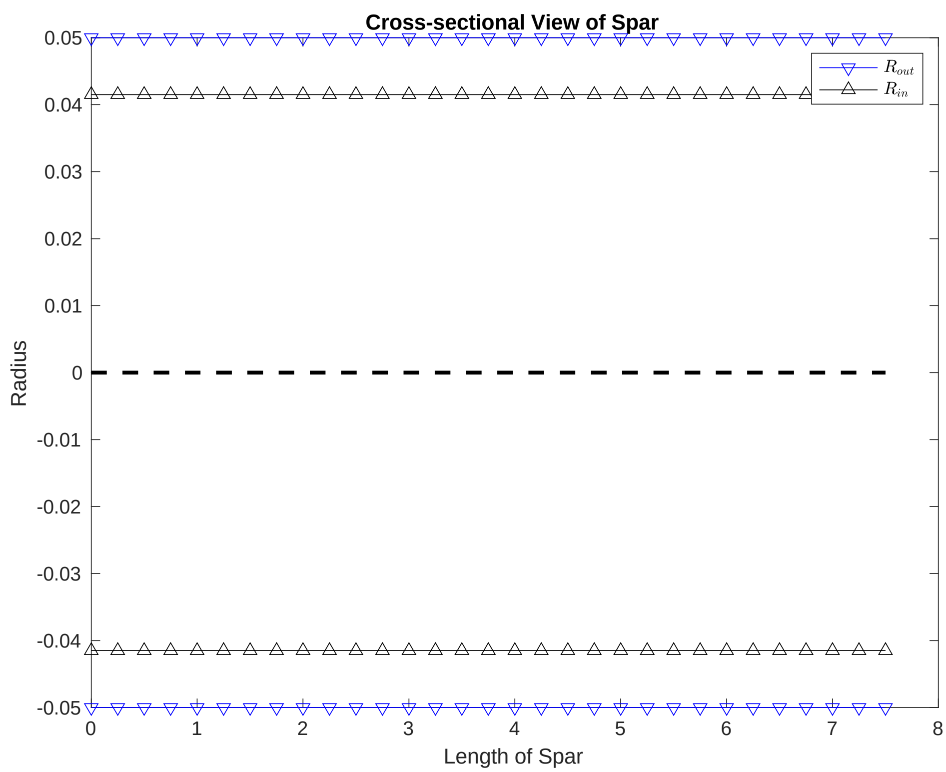

Figure 2. Initial Design Parameterization

After optimization is performed, the optimal profile of the cross-section is shown in Figure 3

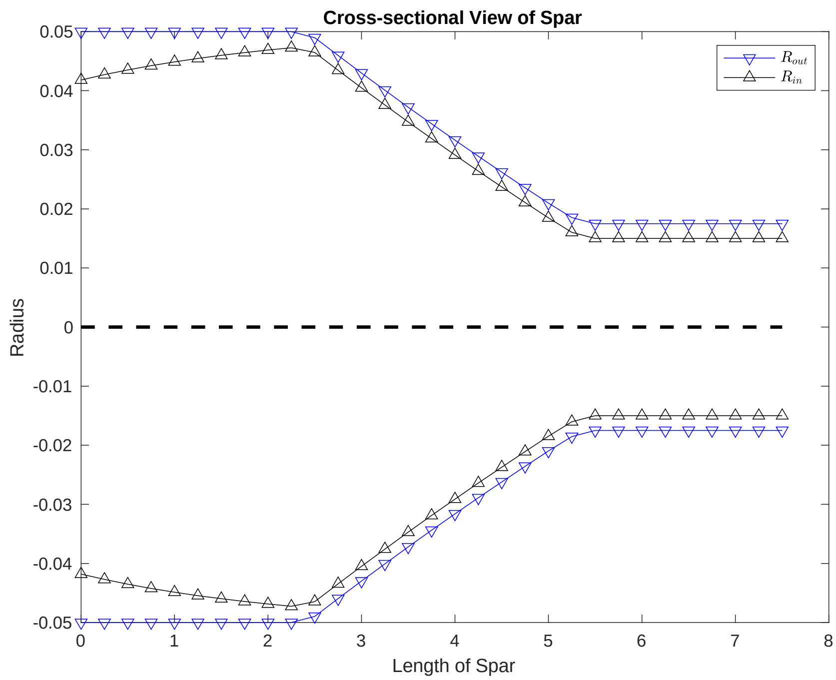

Figure 3. Optimal design of Spar Cross-section

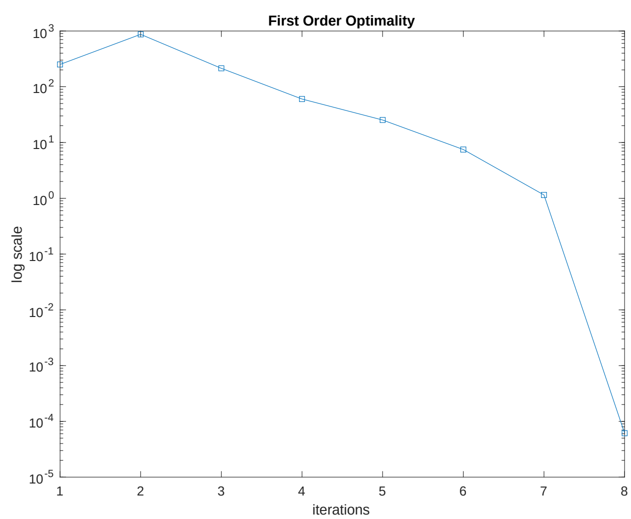

Figure 4. First Order Optimality condition

Figure 4 shows the first order optimality condition to decrease and this conveys the fact that the optimizer has converged to a local minimizer which also satisfies the constraints.

The final weight of this optimal design is $\mathbf{8.837485790505804}\ \boldsymbol{kg}$**.** This is a $\mathbf{69.86}\%$ reduction of weight from the initial design.

Results

A convergence analysis was performed to find if the mass estimate computed in this project improves as the number of elements (N_{elem}) are increased. As more cross-sections are added along the length elements, approximate volume calculated using the trapezoidal integral method improves as δx_i decreases and a small increase in the mass computed can be noticed as shown in Figure 5.

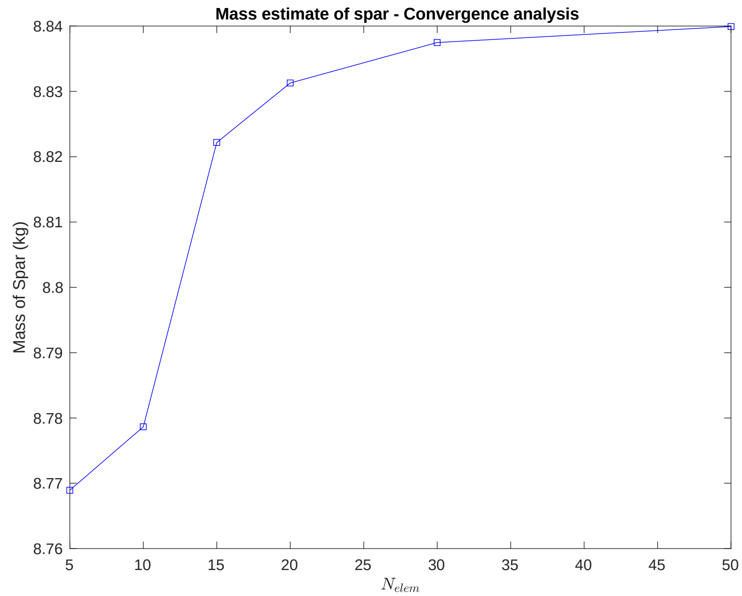

Figure 5. Mass estimates (v) number of elements

Optimized profile undergoes the following mean displacement and mean normal stress displayed in Figure 6 and Figure 7 due to uncertain loading conditions.

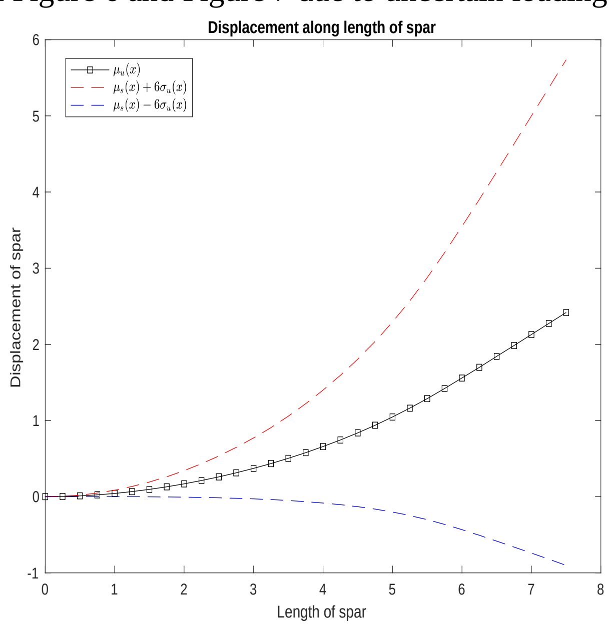

Figure 6. Displacement under uncertain loading

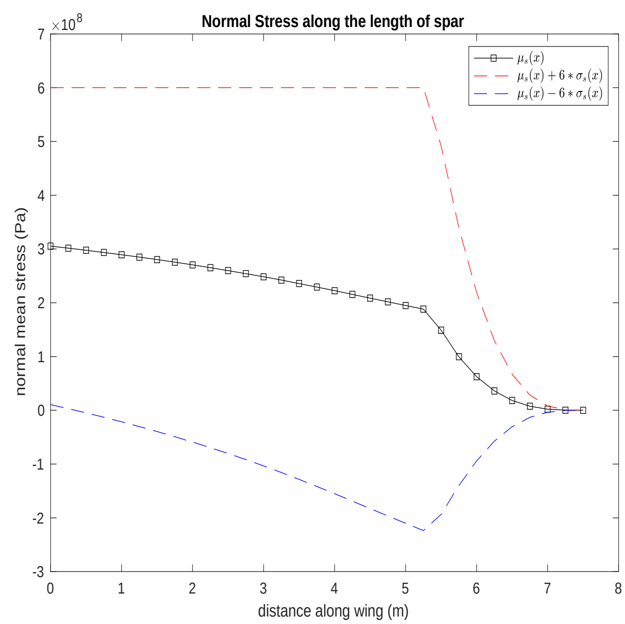

Figure 7. Stress under uncertain loading

It is important to notice the fact that, if the optimization was performed without this uncertainty, the mass of the final spar was approximately 38% lesser than the mass of spar obtained when considering the uncertain loading conditions. The performance of the optimal design achieved using deterministic loading conditions were exactly like the plots of $(\mu + 6\sigma)_{\{u,stress\}}$ shown in Figure 6 and Figure 7. Another important aspect to note is that the increase in mass has particularly happened near the root of the spar where there is maximum load. Since there is a possibility of maximum uncertainty at the root, the addition of more mass at the root makes intuitive sense as well.

This profile for the spar has not just decreased its mass but has also adhered to all the manufacturing constraints specified for the design of the spar. This is the final optimized profile achieved using this parameterization for the choice of design variables $\{R_{out}^{i}, R_{in}^{i}\}\ i: 1(1)n$. This is a robust design as opposed to the optimal design achieved without the consideration of uncertain loading conditions.

Appendix

The MATLAB code written for this project has been attached in this section.

Contents


```
clc
clear all
```

Inital point of design variable

```
Nelem = 30 ; % number of elements along spar
r_out = 5e-2; % m - Outer radius
r_in = 4.15e-2; % m - Inner radius
L = 7.5; %m - Length of spar
x = (0:L/Nelem:L)'; % discretization of length of spar
slope1 = 0;
slope2 = 0;
X0 = ones((2*(Nelem+1)),1); % Inital design variable
k = 1;
for i=1:2:(2*Nelem+1)
    X0(i) = r_out+x(k)*slope1;
    X0(i+1) = r_in+x(k)*slope2;
    k = k+1;
end
```

Setting up Linear Inequality constraint

```
Nnodes = Nelem + 1;
% rin > 1cm ---> -rin < -1cm
A1 = zeros(Nnodes,2*Nnodes);
k=2;
for i=1:((Nnodes))
    A1(i,k) = -1;
    k = k+2;
end
b1 = -1e-2*ones(Nnodes,1);

% rout - rin > 2.5mm ---->  -rout + rin < -2.5mm
A2 = zeros((Nnodes),2*(Nnodes));
k = 1;
for i=1:((Nnodes))
    A2(i,k) = -1;
    A2(i,k+1) = 1;
    k = k+2;
end
b2 = -2.5e-3*ones(Nnodes,1);

% rout < 5cm
A3 = zeros((Nnodes),2*(Nnodes));
k=1;
for i=1:((Nnodes))
    A3(i,k) = 1;
    k = k+2;
```

```
end
b3 = 5e-2*ones(Nnodes,1);

% -rout + rin < 0 ---> rout > rin
A4 = zeros(Nnodes,2*Nnodes);
k=1;
for i=1:Nnodes
    A4(i,k) = -1;
    A4(i,k+1) = 1;
    k = k+2;
end
b4 = zeros(Nnodes,1);
A = [A1;A2;A3;A4];
b = [b1;b2;b3;b4];

lb = ones(2*Nnodes,1);
ub = lb;
lb(2:2:end) = 0.01;
lb(1:2:end) = 0.0175;
ub(2:2:end) = 0.0475;
ub(1:2:end) = 0.05;

W_ini = obj_func(X0); % initial weight of spar
```

Running Optimization

```
options = optimoptions('fmincon','Display','iter-detailed','Algorithm','sqp',...
    'SpecifyObjectiveGradient',true,'SpecifyConstraintGradient',true);

[X_opt,fvalue,~,op,~,grad]=fmincon(@obj_func,X0,A,b,[],[],lb,ub,@NonLnCons,options);
```

Plotting section

```
[Msigma,SDsigma, Mu,SDu] = GetStresses(X_opt);

figure
plot(x,Msigma,'ks-')
hold on;
grid on;
plot(x,Msigma + 6* SDsigma,'r--');
plot(x,Msigma - 6* SDsigma,'b--');
legend('$\mu_{s}(x)$', '$\mu_{s}(x) + 6* \sigma_{s}(x)$',...
    '$\mu_{s}(x) - 6* \sigma_{s}(x)$','Interpreter','latex');
xlabel('distance along wing (m)')
ylabel('normal mean stress (Pa)')
title('Normal Stress along the length of spar')


figure
plot(x,X_opt(1:2:end),'bv-');
hold on;
plot(x,X_opt(2:2:end),'k^-');
plot(x,-X_opt(1:2:end),'bv-');
plot(x,-X_opt(2:2:end),'k^-');
plot(x,0*X_opt(1:2:end),'k--','lineWidth',2);
xlabel('Length of Spar');
ylabel('Radius');
legend('$R_{out}$','$R_{in}$','Interpreter','latex');
```

```
title ('Cross-sectional View of Spar')

figure
plot(x,X0(1:2:end),'bv-');
hold on;
plot(x,X0(2:2:end),'k^-');
plot(x,-X0(1:2:end),'b^-');
plot(x,-X0(2:2:end),'kv-');
plot(x,0*X0(1:2:end),'k--','lineWidth',2);
xlabel('Length of Spar');
ylabel('Radius');
legend('$R_{out}$','$R_{in}$','Interpreter','latex');
title ('Cross-sectional View of Spar')

figure
plot(x,Mu(1:2:end),'ks-');
hold on
grid on
plot(x,Mu(1:2:end) + 6* SDu(1:2:end),'r--');
plot(x,Mu(1:2:end) - 6* SDu(1:2:end),'b--');
xlabel('Length of spar')
ylabel('Displacement of spar')
legend('$\mu_{u}(x)$','$\mu_{s}(x) + 6\sigma_{u}(x)$',...
    '$\mu_{s}(x) - 6\sigma_{u}(x)$','Interpreter','latex');
title('Displacement along length of spar')
```

Calculate Mass of Spar

```
function [Spar_mass,g] = obj_func(X)
% Input - X - Design Variable
% Output - Spar Mass, gradient of objective

Nnodes = (length(X)/2); % No. of nodes
L = 7.5; %m Semi-Length of spar
rho = 1600; % kg/m^3 - density

% Calculate Volume
Volume = Calc_vol(X,L,Nnodes-1);

% Compute Mass
Spar_mass = rho*Volume;

% Compute gradient
g = Calc_objGrad(X);
end
```

Calculate Volume of beam

```
function [Volume] = Calc_vol(R,L,Nelem)
% Input - Radius profile, Lengthm # elements;
% Output - Volume

% R - vector containing the radius [Rout1 Rin1 Rout2 Rin2 ... RoutN RinN]'
Volume = 0;
k = 1;
for i=1:Nelem
    r = R(k:k+3);
    % Average CS area between 2 adjacent nodes
    avgCS = 0.5*(pi*(r(1)^2-r(2)^2)+pi*(r(3)^2-r(4)^2));
    dV = avgCS*(L/Nelem);
    k = k+2;
    Volume = Volume + dV;
end
end
```

Calculate Gradient of the Objective

```
function g = Calc_objGrad(X)
% Input - X - Design Variable
% Output - Gradient of the objective

h = 10^-60; % complex step size

e = zeros(length(X),1);
Nnodes = (length(X)/2); % No. of elements
L = 7.5; %m - Semi-Length of spar
rho = 1600; % kg/m^3 - density of material

for j=1:length(X)
    e(j) = 1;
    x_cmplx = X + (h*e)*i;
    vol_cmplx = Calc_vol(x_cmplx,L,Nnodes-1);
    f_cmplx = vol_cmplx*rho;
    g(j,1) = imag(f_cmplx)/h;
    e(j) = 0;
end
end
```

Calculate the Nonlinear Inequality constraint

```
function [c,ceq,J,Jeq] = NonLnCons(X)
% Input - X - Design Variable;
% Output - Nonlinear inequality {constraint, gradient}
%        - Nonlinear equality {constraint, gradient}

Mass = 500; % total operational mass of aircraft
Nnodes = length(X)/2; % Number of nodes
L = 7.5; %m -  Semi Length of spar
x = (0:L/(Nnodes-1):L)'; % discretize the length
E = 70e9; % 70 GPa Young's modulus
Max_Tensile_Strength = 600e6; % Tensile Strength

% Calculate Iyy
Iyy = Calc_Iyy(X,Nnodes);

zmax = X(1:2:end);
force_nominal = Calc_force(x,Mass,L);
[msig_u,stdDev_sig_u] = uncertainity(zmax,force_nominal,Iyy,E,L,Nnodes-1);


% Compute Nonlinear constraint and its gradient
c = (msig_u + 6* stdDev_sig_u)/Max_Tensile_Strength-1;
J = Calc_consJac(X,Nnodes,L,E,force_nominal,Max_Tensile_Strength);

ceq = [];
Jeq = [];
end
```

Calculate Iyy

```
function Iyy = Calc_Iyy(R,Nnodes)
% R - radii ordered as [Rout1 Rin1 Rout2 Rin2.... RoutN RinN]'

Iyy = zeros(Nnodes,1);
k=1;
for i=1:Nnodes
     Iyy(i) = pi*(R(k)^4 - R(k+1)^4)/4;
     if(Iyy(i)<=1e-12)
        Iyy(i) = 1e-12;
     end
    k = k+2;
end
end
```

Calculate force distribution along length of spar

```
function force = Calc_force(x,Mass,L)
% Input - x (length discretization), Mass, Length of spar;
% Output - force at each node

% force is linear in nature - cX
% integral(cx dx)=2.5*Weight/2 - c = 2.5*Weight/L^2

g = 9.81; % m/s^2
c = 2.5*Mass*g/L^2;
force = c*x;
force = flip(force);
end
```

Calculate the mean statistics of stresses and displacement due to uncertainties

```
function [mStress,stdDevStress,m_u,stdDevu] = uncertainity(zmax,f_nom,Iyy,E,L,Nelem)
% Inputs - zmax   - R_out values at each node point - array of size [Nelem+1 ,1]
%          f_nom  - nominal force values at the node locations
%          Iyy    - Area Moment of Inertia at each node location
%          E      - Young's Modulus
%          L      - Length of the Spar
%          Nelem  - Num of elements
% Outputs- mStress- mean normal stress due to uncertain loading at each
%                   nodal location
%     stdDevStress- standard deviation of normal stress due to uncertain
%                   loading at nodal location
%             m_u - Mean Displacement due to uncertain loading
%          stdDevu- standard deviaiton of displacement due to uncertain
%                   loading

% using a 3 point Gaussian quadrature rule .
xi = [-1.22474487139; 0.0; 1.22474487139];
wt = [0.295408975151; 1.1816359006; 0.295408975151]./sqrt(pi);
% standard deviation of the perturbation variables .
sigma1 = f_nom(1)/10; sigma2 = f_nom(1)/20;
sigma3 = f_nom(1)/30; sigma4 = f_nom(1)/40;

mStress = 0;
m_sqStress = 0;
m_u = 0;
m_u2 = 0;
for i1 = 1:length(xi)
    pt1 = sqrt(2)*sigma1*xi(i1);
    for i2 = 1:length(xi)
        pt2 = sqrt(2)*sigma2*xi(i2);
        for i3 = 1:length(xi)
            pt3 = sqrt(2)*sigma3*xi(i3);
            for i4 = 1:length(xi)
                pt4 = sqrt(2)*sigma4*xi(i4);
                D = Delta(pt1,pt2,pt3,pt4,L,Nelem);
                f_u = f_nom + D;
                u_u = CalcBeamDisplacement(L,E,Iyy,f_u,Nelem);
                stress_u = CalcBeamStress(L,E,zmax,u_u,Nelem);
                % compute mean stress
                mStress = mStress + wt(i1)*wt(i2)*wt(i3)*wt(i4)*stress_u;
                % compute mean stress square
                m_sqStress = m_sqStress + wt(i1)*wt(i2)*wt(i3)*wt(i4).*stress_u.*stress_u;
                % compute mean displacement
                m_u = m_u + wt(i1)*wt(i2)*wt(i3)*wt(i4)*u_u;
                % compute mean displacement square
                m_u2 = m_u2 + wt(i1)*wt(i2)*wt(i3)*wt(i4)*u_u.*u_u;
            end
        end
    end
end
% calculate standard deviaiton of stress
stdDevStress = sqrt(m_sqStress - mStress.^2);

% calculate standard deviation of displacement
stdDevu = sqrt(m_u2 - m_u.^2);
end
```

Compute Jacobian of Nonlinear Inequality constraint

```
function gc = Calc_consJac(X,Nnodes,L,E,force,Max_Tensile_Strength)
% Inputs:
% X - Deisgn Variable;
% Nnodes - # of Nodes;
% L - Semi Length of spar;
% E - Young's Modulus;
% force - Force on each node;
% Max_Tensile_Strenth - Yield tensile/compressive stress;
% Output - Jacobian

h = 10^-60; % complex step size
gc = zeros(length(X),Nnodes);

for j=1:length(X)
    x_cmplx = X;
    x_cmplx(j) = x_cmplx(j) + complex(0,h);
    Iyy_cmplx = Calc_Iyy(x_cmplx,Nnodes);

    zmax_cmplx = x_cmplx(1:2:end);
    [msig_cmplx,sdSig_cmplx] = uncertainity(zmax_cmplx,force,Iyy_cmplx,E,L,Nnodes-1);

    c_cmplx = (msig_cmplx + 6* sdSig_cmplx)./Max_Tensile_Strength - 1;
    gc(j,:) = imag(c_cmplx)./h;
end

end
```

Evaluate statistics of stresses

```
function [msig_u,stdDev_sig_u,mu_u,stdDev_u_u] = GetStresses(X)
% Input  - X            - Design Variable;
% Output - mSig_u       - mean Normal stresses on nodal location
%          stdDev_sig_u - standard deviation of normal stresses on nodes
%          mu_u         - mean displacement on nodal location
%          stdDev_u_u   - standard deviation of displacement on nodes

Mass = 500; % total operational mass of aircraft
Nnodes = length(X)/2; % Number of nodes
L = 7.5; %m -  Semi Length of spar
x = (0:L/(Nnodes-1):L)'; % discretize the length
E = 70e9; % 70 GPa Young's modulus

% Calculate Iyy
Iyy = Calc_Iyy(X,Nnodes);

zmax = X(1:2:end);
force_nominal = Calc_force(x,Mass,L);
[msig_u,stdDev_sig_u,mu_u,stdDev_u_u] = uncertainity(zmax,force_nominal,Iyy,E,L,Nnodes-1);

end
```