Kyle Fairfield

5587

Executive Summary

The purpose of this project was to design the spar of a UAV. While minimizing the weight of the spar, the structure may not exceed the ultimate strength of the carbon fiber, which is the material of the spar, during a 2.5 g maneuver.

To determine the design of the spar, a parameterization of the weight was made. Since the spar length is constant, and the cross section is a circular annulus, the weight can be determined through the volume of the spar, which can be calculated through the inner and outer radii. The spar was divided into multiple sections, with nodes of inner and outer radii. To ensure that any point on the spar does not exceed its ultimate strength, a representation of the moment of area should be made with the radii at each node. Then the displacement and stress at each node can be found.

The function fmincon() was utilized to find the radii at which the weight is at a minimum. It was determined that the minimum weight of the spar to support a 2.5 g maneuver would be 9.8568 kg.

Assumptions and limitations

- The material is carbon fiber composite, with density 1600kg/m^3, Young's modulus of70 GPa, and ultimate strength of 600 MPa, it is assumed to be uniform with no defects
- The total mass of the aircraft will be 500 kg, including the spar
- The spar cross-section shape is a circular annulus
- At a 2.5 g maneuver the force distribution in the spanwise direction will have an linear distribution, with maximum load at the root and zero load at the tip
- 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.

Analysis Method

The analysis is achieved by first modeling the force distribution. Since it is assumed to have a linear distribution, the load on each wing would be equal to half of the load at the base multiplied by the length of the wing. With this, the force at each node can be stored in an array. Using the inner and outer radii at each node, the moment of area for each node can be computed. This is important because one of the constraints involves the structure not exceeding its ultimate strength. The moment of area allows the calculations of displacement and stress at each node. Finally, the method to minimize the weight is to first calculate the volume of each section between nodes, then go through optimization with given constraints.

Parameterization

The system is parameterized by dividing the spar into multiple sections and storing the inner and outer radii at each node. This would also decide the volume of each section of the spar.

Optimization

The objective of the optimization is to minimize the weight of the spar. The weight can be computed by multiplying the volume by material density. Therefore, the objective function can simply be a function of the volume of the spar. To make use of this function, it must be able to be represented by r, the array that contains the inner and out radii at each node. The volume is calculated by adding the volumes of each section. The volumes of each section are calculated by representing the sections as multiple cylinders of the outer radii and inner radii. Subtracting the sum of the small cylinders from the sum of the large cylinders give the approximate volume of each section. Adding each section gives the approximate volume of the spar.

The function fmincon() in MATLAB was utilized. Given the parameter inequalities, this function finds the minimum value of the objective function and its parameters through many iterations. For this problem, the used function is shown below:

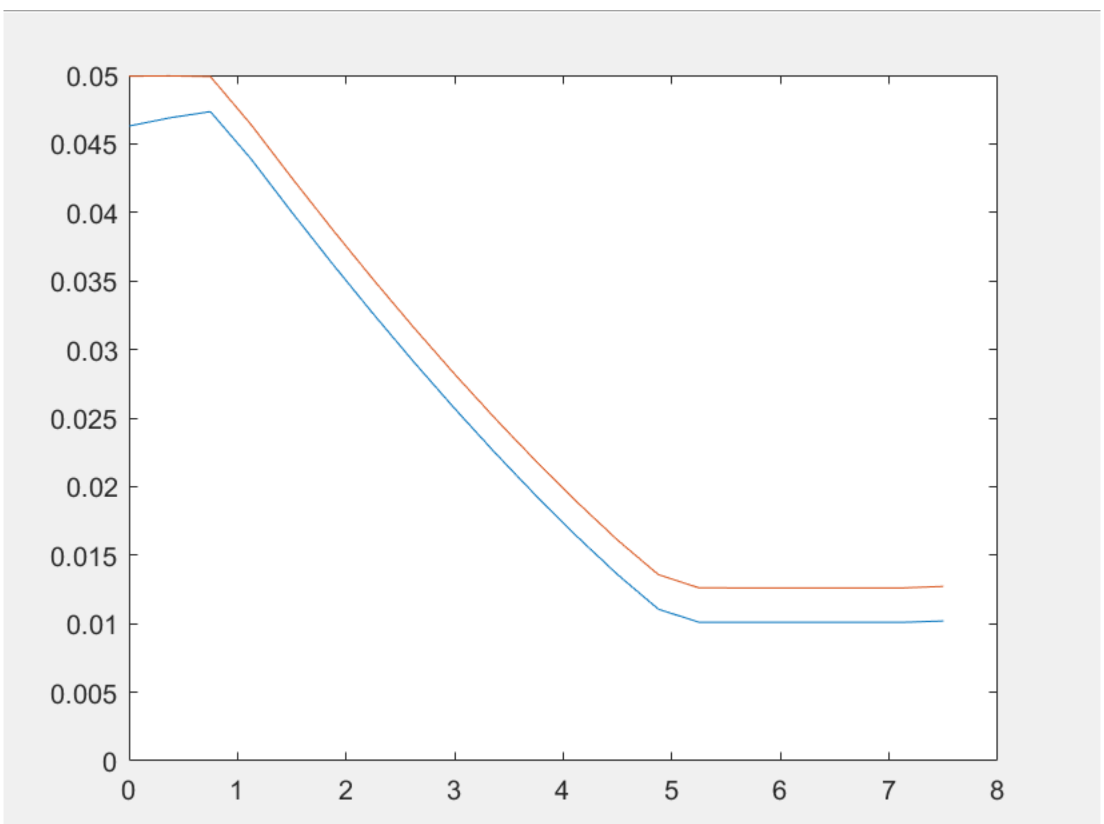
```
[r,fval]= fmincon(fun,r0,A,b,[],[],lb,ub,nonlcon,options);
```

This takes in the objective function f=Vout-Vin, the initial parameters, the inequality matrix A
, the vector matrix b, the lower bound, upper bound, and nonlinear constraints.

The inequalities ensure that the inner and outer radii are at least 2.5mm apart, as per the limitations, the upper and lower bounds ensure that the outer radius is at most 5cm and the inner radius is at least 1cm. These combined satisfy the parameter constraints. The non linear constraint is a function array $c(i)=\sigma_i/\sigma_{ult}-1$. This would satisfy the constraint of the structure not reaching its ultimate strength.

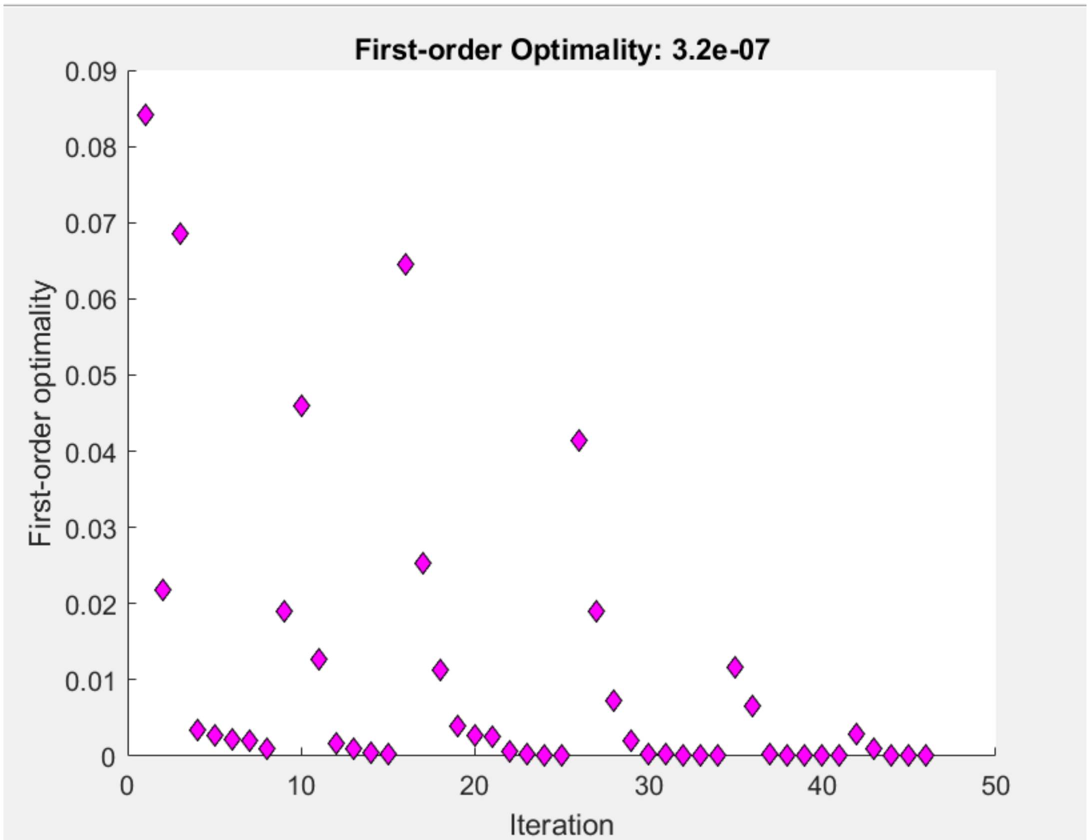
This function returns the optimal values of the radii and the minimum volume that is within the constraints.

Results and discussion



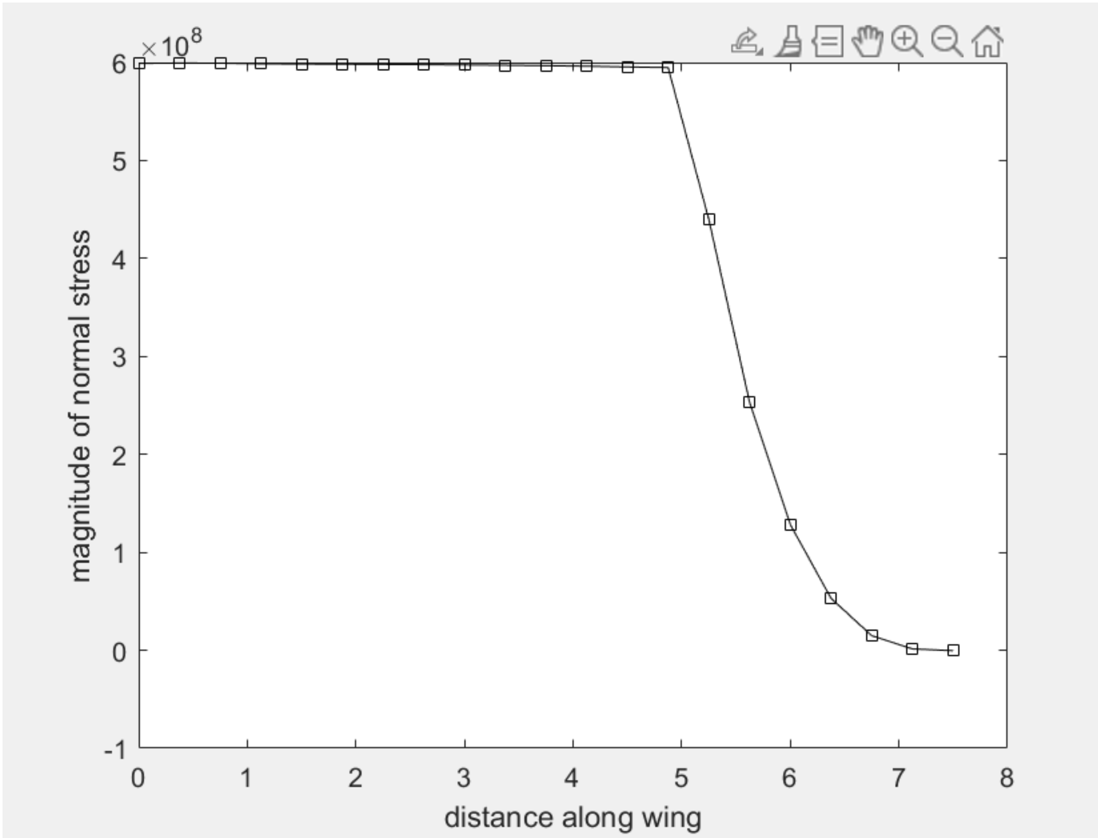
Graph 1. The radius of the spar along one side in meters

At Nelem=20, the resulting volume from the fmincon() function was 0.0031m^3. With the material density being 1600 kg/m^3, the weight on one side would be 4.9284kg, so the total weight is 9.8568kg.



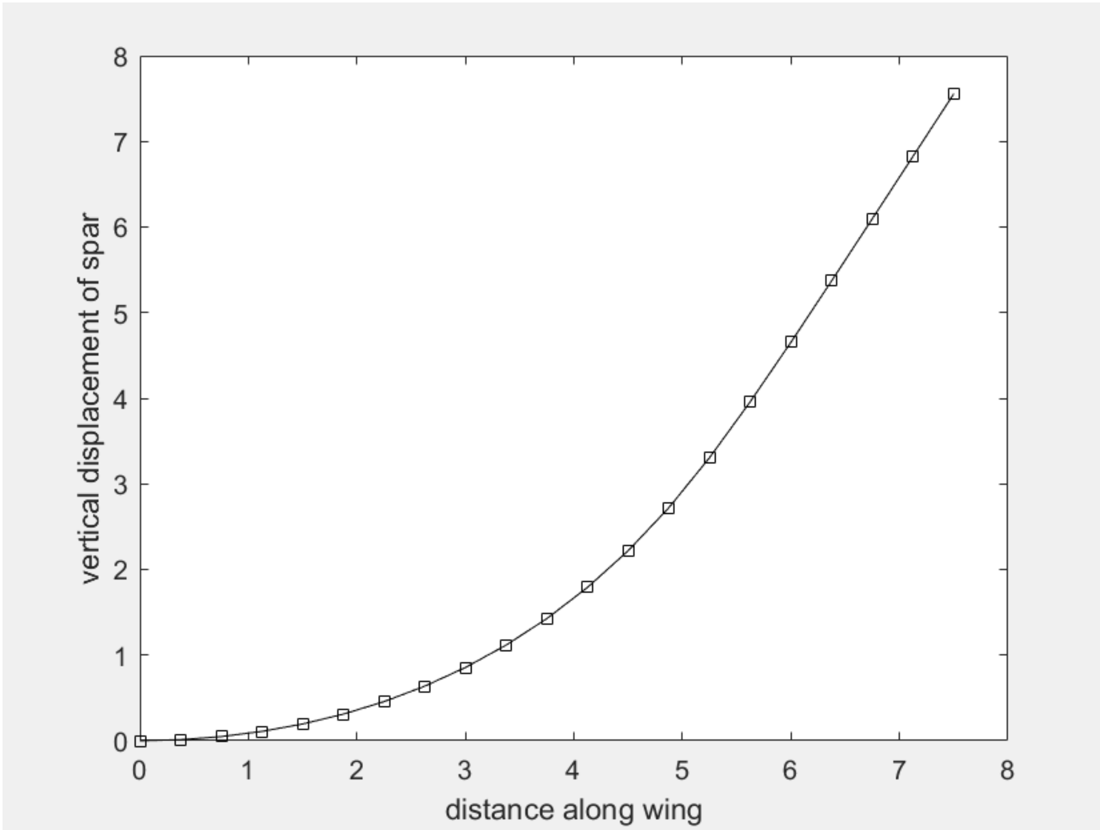
Graph 2. First order optimality of each iteration

The resulting radii start at high values, and linearly decrease with a rate of about 0.009m/m until they reach their constraint values. This makes sense because the force follows a linear distribution as well. Closer to the root of the wing, a higher radius is required to support the load without exceeding the ultimate strength of 600 MPa.



Graph 3. Normal stress (Pa) at nodes along the wing (m)

The stress remains close to the ultimate strength of the material from its root to roughly 5m away. This aligns with the approximate positions where the radii are linearly decreasing. After 5m, the stress rapidly approaches zero.



Graph 4. Vertical displacement (m) along the wing (m)

Appendix

Runopt.m

```
%optimization
% conditions
E= 70000000000;
L=7.5;
m=500;
Nelem=20;
uts=600000000;
p=1600;

%force at each node
force=forcev(L,m,Nelem);

%Aineq for fmincon to get rin-rout<=-0.0025
A=zeros(Nelem+1,2*(Nelem+1));
for i=1:Nelem+1
    A(i,i)=1;
    A(i,Nelem+1+i)=-1;
end
%set r0 where inner radius is 0.03 and outer is 0.04
r0=[];
for i=1:(Nelem+1)
    r0(i)=0.03;
end

for i=1:Nelem +1
    r0(Nelem+1+i)=0.04;
end

b=[];
for i=1:(Nelem+1)
    b(i)=-0.0025;
end



fun = @(r) obj(r,Nelem,L,E,force);

%sets the upper and lower bounds of radii
ub=[];
for i=1:(Nelem+1)
    ub(i)=0.0475;
end

for i=1:Nelem +1
    ub(Nelem+1+i)=0.05;
end

lb=[];
for i=1:(Nelem+1)
    lb(i)=0.01;
end
```

```
for i=1:Nelem +1
    lb(Nelem+1+i)=0.0125;
end



options=optimset('Display','iter','plotfcns','optimplotfirstorderopt');

%the stress constraint
nonlcon=@(r) stresscon(r, L, E, Nelem, force, uts);

%fmincon
[r,fval]= fmincon(fun,r0,A,b,[],[],lb,ub,nonlcon,options);

hold on

figure(1);

%calculate I, zmax, u, and sigma for personal testing, does not affect
%optimization
I=calcI(r,Nelem);

zmax=transpose(r(Nelem+2:2*(Nelem+1)));

u = CalcBeamDisplacement(L, E, I, force, Nelem);
sigma=CalcBeamStress(L, E, zmax, u, Nelem);

figure(2);
% plot the vertical displacement
x = [0:L/Nelem:L];
plot(x,u(1:2:2*(Nelem+1)),'ks-');
xlabel('distance along wing')
ylabel('vertical displacement of spar')

% plot the stresses
figure(3);

[sigma] = CalcBeamStress(L, E, zmax, u, Nelem);
plot(x,sigma,'ks-')
xlabel('distance along wing')
ylabel('magnitude of normal stress')

%calculates the total weight
weight= 2*fval*p;

figure(4);
%plots the radii to give an idea of the shape
plot(x,r(1:Nelem+1),x,r(Nelem+2:2*(Nelem+1)),x,0)
```

Obj.m

```
%objective function
function [f]=obj(r,Nelem,L,E,force)
```

```
%set up Volume of shape from outer radius
Vout=0;
for i = 1:Nelem

    %divide each section into 50 subsections
    %dV=pi*routx^2*dx is the volume of each subsection,

    for j= 1:50

        x=L*j/(Nelem*50);
        dx=L/(Nelem*50);
        %the slope of each section to find r at each subsection node
        slope=((r(i+1+Nelem+1)-r(i+1+Nelem))/(L/Nelem));
        %r at each subsection node
        routx=slope*x+r(i+1+Nelem);

        %routxsq=slope^2*x^2+2*slope*x*r(i+1+Nelem)+(r(i+1+Nelem))^2;
        %experimental code

        Vout=Vout+pi*routx^2*dx;

        %Vout=Vout+pi*(1/3*slope^2*x^3+slope*x^2*r(i+1+Nelem)+x*(r(i+1+Nelem))^2);
        %experimental code
    end
end

%similar to Vout
Vin=0;
for i = 1:Nelem
    %dV=pi*rinx^2*dx
    for j= 1:50

        x=L*j/(Nelem*50);
        dx=L/(Nelem*50);
        slope=((r(i+1)-r(i))/(L/Nelem));
        rinx=slope*x+r(i);
        %rinxsq=slope^2*x^2+2*slope*x*r(i)+(r(i))^2;
        Vin=Vin+pi*rinx^2*dx;
        %Vin=Vin+pi*(1/3*slope^2*x^3+slope*x^2*r(i)+x*(r(i))^2);
    end
end


%gives the volume of each half of the spar
f=Vout-Vin;
```

calcI.m

```
%funtion to calculate the moment of area

function Iyy= calcI(r, Nelem)
%takes in r and gives the I at each node
Iyy=[];
```

```
for i=1:(1+Nelem)
    Iyy(i,1)= pi/4*(r(i+1+Nelem)^4-r(i)^4);
    if Iyy(i,1)<0
        Iyy(i,1)=-Iyy(i,1);
    end

end
```

stresscon.m

```
%nonlincon for fmincon
%stores the stress in sigma then outputs c(i) to represent
%sigma(i)/uts-1<=0
function [c,ceq]= stresscon(r, L, E, Nelem, force, uts)

Iyy=calcI(r, Nelem);

zmax=transpose(r(Nelem+2:2*(Nelem+1)));



u = CalcBeamDisplacement(L, E, Iyy, force, Nelem);
sigma=CalcBeamStress(L, E, zmax, u, Nelem);

for i=1:Nelem+1

    c(i)=sigma(i)/uts-1;
end
ceq=[];
```

forcev.m

```
%storage of force on each node
function [q]=forcev(L, m, Nelem)
q=[];

dx=L/Nelem;


q0=2.5*9.8*m/L;

for i =1:Nelem+1
    x=(i-1)*dx;
    q(i,1)=q0-q0*x/L;
end
```