AE 6310 Final Project

or

A Dark Outcome for Solar Powered Flight

Madeleine Graham Macy

4/29/2024

Abstract

This OpenMDAO-powered case study investigates the feasibility of a solar-powered aircraft using low-fidelity structural, aerodynamic, and propulsion models. Wing planform area, aspect ratio, taper ratio, cruise velocity, and trim angle of attack are used as design variables in an optimization problem that minimizes the weight of the payload subject to structural and aerodynamic constraints. This study finds that the feasability of such an aircraft is heavily constraint by the current solar panel technology. A trade-off study shows a direct correlation between solar efficiency and maximum payload, with the current technology being in the unfeasible regime. Furthermore, lightweight materials such as carbon fiber are a must when constructing this type of aircraft, but the weight of solar panels is still such that lift generated would be insufficient for any powered steady, level flight.

Introduction

Solar power is an enticing idea, especially in aviation. If the only obstacle to free and infinite power is shadow or clouds, why not make the best use of humankind's aviation technology fly above them? However, the only way to achieve the thrust required to fly an aircraft is to place solar panels on a surface area large enough to overcome the drag that that very surface area creates. This introduces a sizing dilemma for solar powered aircraft: how can one build an ultralight, solar-powered aircraft that has a large enough surface area to produce enough solar power for steady, level flight above the clouds? This study sets up a design space wherein wing planform area, aspect ratio, and taper ratio are considered as design variables and then subjected to calculations for the conditions of cruise condition at steady, level flight. An optimizer is used to find a maximum payload for this condition subject to feasibility constraints in terms of structures and aerodynamics. The structural feasibility constraints include weight of the wings as a function of number of ribs and spars, thickness and density of the material chosen, and a maximum stress multiplied by a safety factor. The feasibliliy constraints on the aerodynamics side include a 2-D coefficient of lift constraint that limits the angle of attack so as to prevent stall occurring on the wing. Other feasibility constraints include physical limits, such as a positive payload, a maximum flow condition of Mach = 0.4 to account for the fact that flow is incompressible in my model, and an altitude of 10,000 ft for cruise condition.

Variables

Variable	Value	Description	Units
S		wing planform area	ft^2
AR		wing aspect ratio	None
TR		wing taper ratio	None
V		velocity	ft/s
α		trim angle of attack	degrees
ρ		air density	$slugs/ft^3$
T		thrust	lbf
W_{wing}		wing weight	lbf

G	68.52176	solar radiation	lbf*ft/s/ft^2
η_s	0.2	solar panel efficiency	None
η_p	1	power chain efficienct	None
P_T		propulsive power	lbf*ft/s
$W_{fuselage}$		fuselage weight	
W_{motor}		motor weight	
g	32.17	gravitational constant	ft/s^2
m_{mat}		material mass	slugs
t_{spar}		thickness of spar	ft
t_{rib}		thickness of rib	ft
t_{sheet}		thickness of sheet of material covering wing	ft
C_f		coefficient of friction	
Re		Reynold's number	

Multidisciplinary Analysis and Optimization Problem

maximize	$W_{payload}$
with respect to	x = [S, AR, TR], V, α
subject to	Aerodynamics
	Structures
	Propulsion

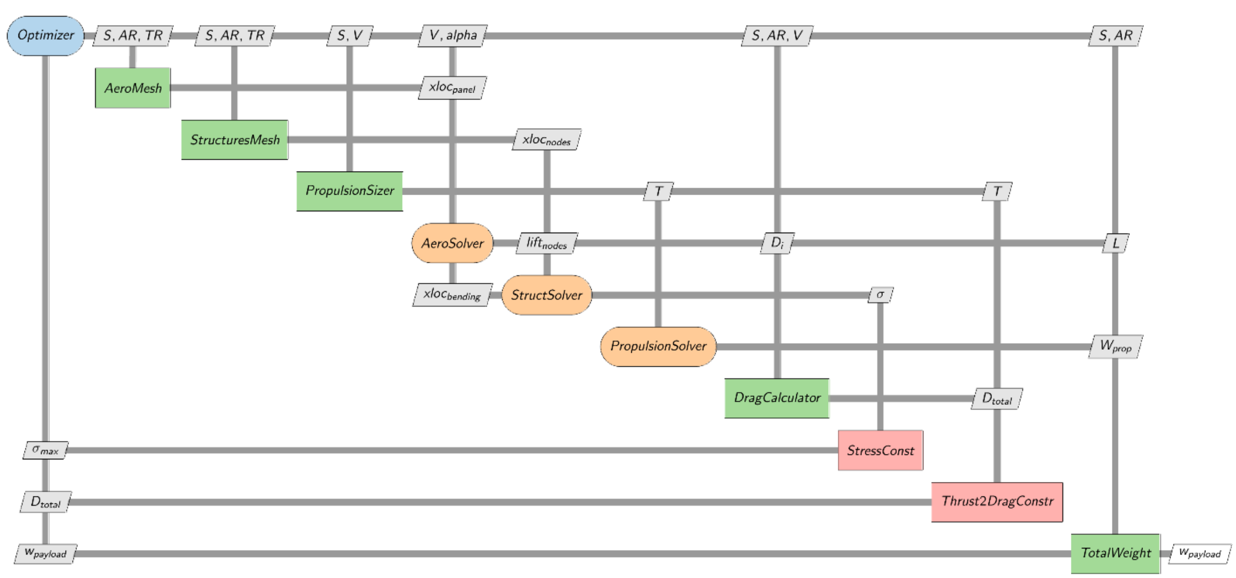

Figure 1

Figure 1 shows an XDSM diagram for this particular design problem. The optimizer selects the aircraft design variables, S, AR, and TR, which feed into the Aero mesh and the Structures Mesh. The Aero and Structures Mesh translate the design parameters into locations of panels and nodes, respectively. The location of the panels feeds into the "AeroSolver," which implements the Vortex Lattice Method analysis to find the lift and drag forces acting on each panel.

The panel lift is fed into the "StructSolver" code, which uses Finite Element Analysis in order to find the displacements on the wing caused by the lifting forces. Once the bending is found, it is then plugged back into the AeroSolver in order to adjust the panel lift forces, which will now be different due to the bending. This cycle is iterated upon until convergence.

After the Aero-Structures cycle converges, the structural solver also finds the maximum stress on each panel and connects that to the stress constraint on our problem. The aerodynamics solver also sums the forces to be used in the calculation for "Total Weight" and the "Drag Calculator" further downstream.

The "PropulsionSizer" code takes in V and S and outputs Thrust using the following equation:

$$T = \frac{\eta_s \eta_p G S}{V}$$

(Eq 1)

The thrust can then be used to fulfill the thrust-to-drag equality constraint. Thrust is also used in the "PropulsionSolver" equation, wherein the propulsion motor weight is found thuswise:

$$W_{motor} = a P_T^2$$

(Eq 2)

Where "a" is some small coefficient and P_T is the thrust of the motor multiplied by the velocity.

The Drag Calculator uses the following equation to calculate drag:

$$D = q\epsilon 0 + qS2kC_f + D_i$$

(Eq 3)

In calculating drag, a laminar boundary layer was assumed, as opposed to the model given in the proposal, which was a fully turbulent boundary layer. This changed the equation for skin friction drag, C_f to the following:

$$C_f = \frac{0.074}{Re^{0.5}}$$

(Eq 4)

The objective function is the total weight function, given by the following equation:

$$W_{total} = L = W_{payload} + W_{motor} + W_{wing} + W_{fuselage}$$

(Eq 5)

The above equation is rearranged for $W_{payload}$, and W_{total} is replaced by Lift generated by the Aero Solver code in order to allow the optimizer to maximize $W_{payload.}$ The wing weight is given by the following equation:

$$W_{wing} = 2 * (St_{sheet} m_{mat} g + 8 * t_{rib} m_{mat} gc + 4 * \frac{t_{spar} b}{2} m_{mat} g)$$

(Eq 6)

Where, m_{mat} is the mass of the chosen material, t_{sheet}, t_{rib}, and t_{spar} are chosen thicknesses (assuming a very thin airfoil), g is the gravitational constant, c is the mean aerodynamic chord, and b is the span of the wing. The rib and spar numbers are multiplied by 8 and 4 respectively, beecause that is how many ribs and spars were part of the design choice.

Material choices

Initially, I had chosen to use aluminum because it is a lightweight metal widely used in aviation, and therefore there was a lot of information about its usage online.

However, I quickly ran into issues with my design being unable to withstand the stress placed on the wings. I therefore switched to carbon fiber. I created a strong, lightweight wing using the following data I was able to find online.

Table I: Properties of Carbon Fiber

Property	Value	Units
Elastic Modulus	725.18	Mpsi
Yield Stress	72.518	ksi
Density	3.8	slugs/ft^3

Aerodynamics and Structures Coupled Problem Formulation

The Aerodynamic solver finds the forces acting on each "panel." These panels are then summed in order to find the Aerodynamic forces for the entire wing. In order to solve for the corresponding displacements, a method of transferring forces from the panels to the elements required to perform a finite element analysis. The elements correspond with physical spars and ribs that exist as the wing structure.

The following method was used in order to convert panel-centric forces to element forces:

Since the panel forces act towards the top of the panel (towards the leading edge), the forces were distributed towards amongst the elements on the elements that are at the spar closest to the top of the panel. The panel force is divided evenly.

The structures code calculates the displacements based on those forces and then the displacements are fed back into a bending equation superimposed on the aerodynamic forces solver.

When converting back to the aerodynamic solver, it was again necessary to transfer the calculations of displacement from the elements used in the finite element analysis to the mesh used for the panel method. The mesh is then reshaped to accommodate for the bending. The bending lowers the projected area of the wing, which changes the panel forces.

A converged model resulted in the code printout in Appendix I.

Baseline Model

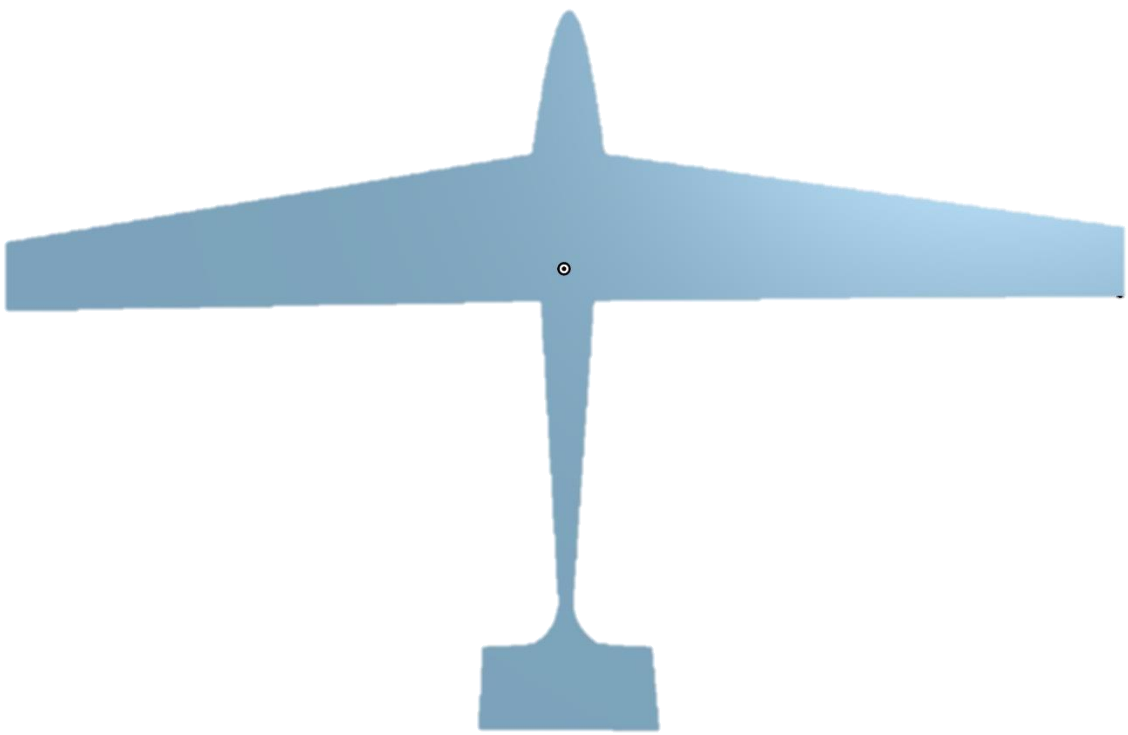

Figure 2

Baseline model characteristics:

Variable	Value	Units
S	350	ft^2
AR	10	None
TR	0.15	None
V	100	ft/s
ρ	0.0017	$slugs/ft^3$
α	0.07	degrees
max stress	8648	psi

T	95.9	lbf
Payload weight	13	lbf
η_s	0.4	None
η_p	1	None
$W_{fuselage}$	67.4426	lbf

Assumptions that were made for the baseline model were the following: solar power efficiency is twice the level of the current technology. The fuselage is 10 times lighter than what was initially presented in the problem. The model is flying at 10,000 ft (therefore, a lower density than at sea level). The baseline model also has solar panels that are not factored into the weight of the aircraft whatsoever, thus assuming a future ultra light-weight solar panel technology. Lastly, an ideal powerchain situation is assumed.

Gradient Evaluation of Functions of Interest

The two most important functions in the code are the structures solver and the aerodynamic solver, which find the lift, drag, and structures. In order to conduct a sensitivity analysis, it was necessary to use a method called automatic differentiation. Since the relevant code exists in python, a python-compatible library called JAX was implemented. However, due to the limited time available for this study, JAX was never run successfully. Python is notorious for slow runtimes, and JAX was no exception – one bug each accounted for at least 30 minutes to an hour. Unfortunately, this did not allow for automatic differentiation to be implemented as a part of the optimization of the model. Though it is not exact, I used the finite difference method in order to implement the gradient for each of my functions.

Optimization of the Model

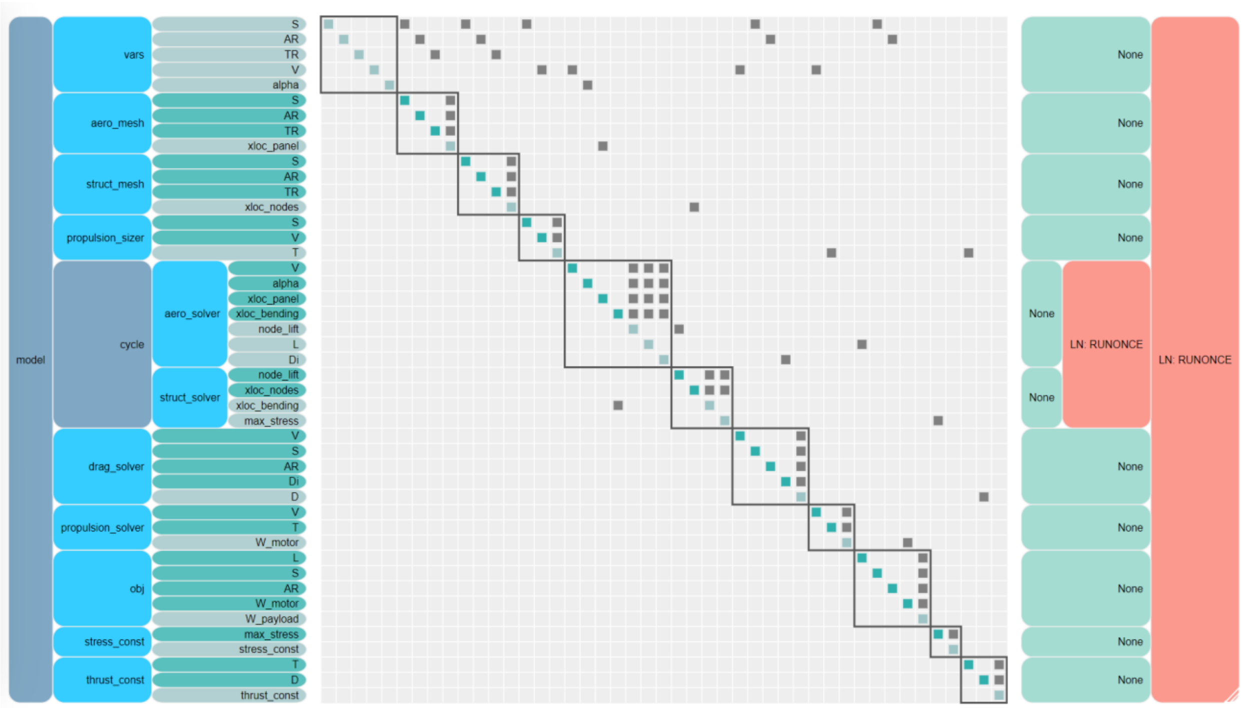

Figure 3

Optimization was performed using OpenMDAO, a Python-based optimizer. Setting up payload weight as a minimum and using the XDSM diagram from my model, I inputted reasonable bounds into the optimizer and let it use finite difference as its gradient method. Figure 3 shows the OpenMDAO N2 diagram of the code structure.

Table II: Design Variables for Optimization and Bounds

Design Variable	Lower bound	Upper Bound
S	50	350
AR	10	15
TR	0.1	0.9
V	50	100
α	0	18

The constraints imposed on the optimization problem were a maximum stress constraint, which was the yield stress of carbon fiber (Table I). The C_{lmax} constraint was met by imposing a restriction on the angle of attack to satisfy a lift-curve slope for a 2-D symmetric airfoil:

$$C_l = 2\pi\alpha$$
$$C_{lmax} = 2.0$$

A symmetric airfoil was assumed because the vortex lattice method code that was given produced no lift at an angle of attack of 0.0 degrees.

An equality constraint of thrust to drag was imposed as well.

The behavior of the optimizer seemed to either favor a very high wing area or an extremely small wing area, and almost nothing in between. The optimizer tended to favour an extremely small wing planform area, AR, and TR in order to maximize payload weight. This behavior tended to result in physically impossible situation: a negative payload weight with an extremely low thrust value. This was only overcome when I increased the bounds for wing planform area.

Because I did not start with a feasible design point in my model, my optimized point is the “Baseline model” seen above.

Trade-Off Study

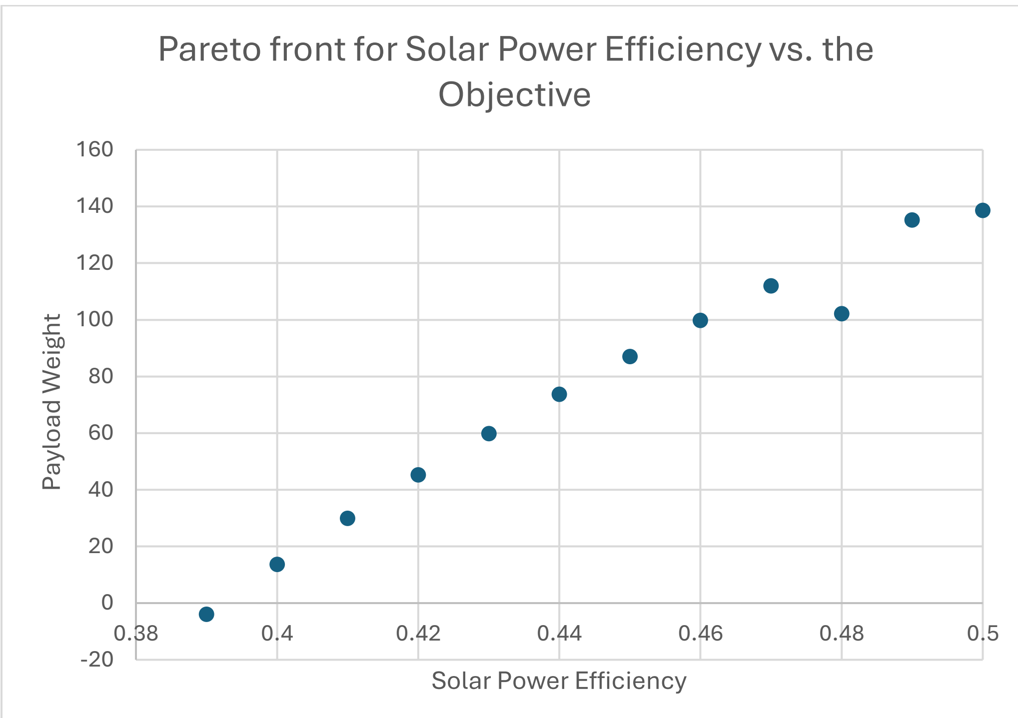

Figure 4

The Pareto front in figure 4 shows that as solar power efficiency grows, the optimized payload weight increases (with different conditions attached). There is a slight dip at an assumed solar efficiency of 0.48 due to the optimizer finding slightly different flight conditions (V, α), but the trend is obvious.

Conclusion

In conducting my investigation into the feasibility of this type of aircraft, it became clear that the largest bottleneck was producing enough power from the solar panels to overcome the drag of the vehicle. Another difficulty was choosing the right material so as not to exceed the stress limit on the wing. Since a high wing area and low aspect ratio (due to the laminar boundary layer assumption) is necessary to produce enough thrust, the bending stress is severe without a material that is stiff enough. Since my area of expertise falls more into the aerodynamics catergory, I suggest further investigation into the arrangement of ribs and spars in the wings and materials for wing construction.

It was necessary to lessen the weight of the fuselage in order to carry any payload, as well as assume a weightless solar panel. This assumption is not a good assumption, but it might inform a design/usage direction of a future solar aircraft to be unpiloted, so that a human's weight need not be considered as payload weight.

Lastly, the optimization problem may have been improperly set up. If I had to perform this design task again, I would set up the optimization problem in the following way:

maximize	Range
with respect to	x = [S, AR, TR], V, α
subject to	Aerodynamics
	Structures
	Propulsion
	Phyisical constraints

In this case, I would be able to set the constraint that payload be a positive number, and the optimizer would be looking for a higher thrust such that range would be maximized. With this set up, I think I could avoid areas in the problem space where the optimizer is shrinking the aircraft size as much as possible.

References

https://bluetunadocs.com/downloads/sheetmetalintro.pdf

https://www.machinedesign.com/materials/article/21831769/basics-of-aerospace-materials-aluminum-and-composites

https://www.theworldmaterial.com/weight-density-of-aluminum/

https://material-properties.org/carbon-fiber-density-strength-melting-point/

Appendix I

=====

cycle

=====

NL: NLBGS Converged in 2 iterations

Optimization terminated successfully   (Exit mode 0)

Current function value: -146.89380087081423

Iterations: 17

Function evaluations: 24

Gradient evaluations: 16

Optimization Complete

----

minimum found at

S = [350.]

AR = [10.]

TR = [0.15221181]

V = [71.31359516]

alpha = [0.21945062]

minumum objective

[-146.89380087]

max_stress

[30179.5314667]

stress constraint

[-42339.3185333]

thrust

[134.5191808]

drag

[134.519155]

motor weight

[1.87916456]

Bending from Aero Solver

```
[-2.84976445e-14  1.26420822e-07  4.56089388e-07  9.24913830e-07
  1.48231232e-06  2.09050832e-06  2.72340346e-06  3.36507456e-06
 -3.40325664e-12  1.31785952e-07  4.78212999e-07  9.64790273e-07
  1.53487727e-06  2.14787469e-06  2.77759421e-06  3.41117043e-06
 -2.68931487e-12  1.37863266e-07  5.00221552e-07  1.00442846e-06
  1.58734045e-06  2.20519197e-06  2.83182251e-06  3.45722241e-06
  9.98345118e-13  1.44288584e-07  5.22270039e-07  1.04393354e-06
  1.63975831e-06  2.26248050e-06  2.88610148e-06  3.50321818e-06]
```

Bending from Structural Solver

```
[-2.84976445e-14  1.26420822e-07  4.56089388e-07  9.24913830e-07
  1.48231232e-06  2.09050832e-06  2.72340346e-06  3.36507456e-06
 -3.40325664e-12  1.31785952e-07  4.78212999e-07  9.64790273e-07
  1.53487727e-06  2.14787469e-06  2.77759421e-06  3.41117043e-06
 -2.68931487e-12  1.37863266e-07  5.00221552e-07  1.00442846e-06
  1.58734045e-06  2.20519197e-06  2.83182251e-06  3.45722241e-06
  9.98345118e-13  1.44288584e-07  5.22270039e-07  1.04393354e-06
  1.63975831e-06  2.26248050e-06  2.88610148e-06  3.50321818e-06]
```

difference between the two

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0.]
```

Lift from Aero Solver

```
[[ 0.          0.          11.45485093  0.          0.          0.        ]
 [ 0.          0.          11.45485093  0.          0.          0.        ]
 [ 0.          0.          11.45485093  0.          0.          0.        ]
 [ 0.          0.          10.87876531  0.          0.          0.        ]
 [ 0.          0.          10.87876531  0.          0.          0.        ]
```

```
[ 0.          0.          10.12939456  0.          0.          0.        ]
[ 0.          0.          10.12939456  0.          0.          0.        ]
[ 0.          0.          9.24849013  0.          0.          0.        ]
[ 0.          0.          9.24849013  0.          0.          0.        ]
[ 0.          0.          8.23618716  0.          0.          0.        ]
[ 0.          0.          8.23618716  0.          0.          0.        ]
[ 0.          0.          7.03153665  0.          0.          0.        ]
[ 0.          0.          7.03153665  0.          0.          0.        ]
[ 0.          0.          5.34274222  0.          0.          0.        ]
[ 0.          0.          5.34274222  0.          0.          0.        ]
[ 0.          0.          2.95317484  0.          0.          0.        ]
[ 0.          0.          2.95317484  0.          0.          0.        ]
[ 0.          0.          2.95317484  0.          0.          0.        ]
[ 0.          0.          4.25457158  0.          0.          0.        ]
[ 0.          0.          4.25457158  0.          0.          0.        ]
[ 0.          0.          3.98159122  0.          0.          0.        ]
[ 0.          0.          3.98159122  0.          0.          0.        ]
[ 0.          0.          3.64130507  0.          0.          0.        ]
[ 0.          0.          3.64130507  0.          0.          0.        ]
[ 0.          0.          3.23691914  0.          0.          0.        ]
[ 0.          0.          3.23691914  0.          0.          0.        ]
[ 0.          0.          2.73727652  0.          0.          0.        ]
[ 0.          0.          2.73727652  0.          0.          0.        ]
[ 0.          0.          1.98655773  0.          0.          0.        ]
[ 0.          0.          1.98655773  0.          0.          0.        ]
[ 0.          0.          0.711365    0.          0.          0.        ]
[ 0.          0.          0.711365    0.          0.          0.        ]
```

```
[ 0.         0.         0.711365   0.         0.         0.        ]
[ 0.         0.         1.0375805  0.         0.         0.        ]
[ 0.         0.         1.0375805  0.         0.         0.        ]
[ 0.         0.         0.97582641 0.         0.         0.        ]
[ 0.         0.         0.97582641 0.         0.         0.        ]
[ 0.         0.         0.8939744  0.         0.         0.        ]
[ 0.         0.         0.8939744  0.         0.         0.        ]
[ 0.         0.         0.79377289 0.         0.         0.        ]
[ 0.         0.         0.79377289 0.         0.         0.        ]
[ 0.         0.         0.66656807 0.         0.         0.        ]
[ 0.         0.         0.         0.         0.         0.        ]
[ 0.         0.         0.46996183 0.         0.         0.        ]
[ 0.         0.         0.46996183 0.         0.         0.        ]
[ 0.         0.         0.711365   0.         0.         0.        ]
[ 0.         0.         0.711365   0.         0.         0.        ]
[ 0.         0.         0.711365   0.         0.         0.        ]
[ 0.         0.         1.0375805  0.         0.         0.        ]
[ 0.         0.         1.0375805  0.         0.         0.        ]
[ 0.         0.         0.97582641 0.         0.         0.        ]
[ 0.         0.         0.97582641 0.         0.         0.        ]
[ 0.         0.         0.8939744  0.         0.         0.        ]
[ 0.         0.         0.8939744  0.         0.         0.        ]
[ 0.         0.         0.79377289 0.         0.         0.        ]
[ 0.         0.         0.79377289 0.         0.         0.        ]
[ 0.         0.         0.66656807 0.         0.         0.        ]
[ 0.         0.         0.66656807 0.         0.         0.        ]
[ 0.         0.         0.46996183 0.         0.         0.        ]
```

```
 [ 0.          0.          0.46996183  0.          0.          0.        ]]
Lift from Structural Solver
[[ 0.          0.         11.45485093  0.          0.          0.        ]
 [ 0.          0.         11.45485093  0.          0.          0.        ]
 [ 0.          0.         11.45485093  0.          0.          0.        ]
 [ 0.          0.         10.87876531  0.          0.          0.        ]
 [ 0.          0.         10.87876531  0.          0.          0.        ]
 [ 0.          0.         10.12939456  0.          0.          0.        ]
 [ 0.          0.         10.12939456  0.          0.          0.        ]
 [ 0.          0.          9.24849013  0.          0.          0.        ]
 [ 0.          0.          9.24849013  0.          0.          0.        ]
 [ 0.          0.          8.23618716  0.          0.          0.        ]
 [ 0.          0.          8.23618716  0.          0.          0.        ]
 [ 0.          0.          7.03153665  0.          0.          0.        ]
 [ 0.          0.          7.03153665  0.          0.          0.        ]
 [ 0.          0.          5.34274222  0.          0.          0.        ]
 [ 0.          0.          5.34274222  0.          0.          0.        ]
 [ 0.          0.          2.95317484  0.          0.          0.        ]
 [ 0.          0.          2.95317484  0.          0.          0.        ]
 [ 0.          0.          2.95317484  0.          0.          0.        ]
 [ 0.          0.          4.25457158  0.          0.          0.        ]
 [ 0.          0.          4.25457158  0.          0.          0.        ]
 [ 0.          0.          3.98159122  0.          0.          0.        ]
 [ 0.          0.          3.98159122  0.          0.          0.        ]
 [ 0.          0.          3.64130507  0.          0.          0.        ]
 [ 0.          0.          3.64130507  0.          0.          0.        ]
 [ 0.          0.          3.23691914  0.          0.          0.        ]
```

```
[ 0.         0.         3.23691914  0.         0.         0.        ]
[ 0.         0.         2.73727652  0.         0.         0.        ]
[ 0.         0.         2.73727652  0.         0.         0.        ]
[ 0.         0.         1.98655773  0.         0.         0.        ]
[ 0.         0.         1.98655773  0.         0.         0.        ]
[ 0.         0.         0.711365    0.         0.         0.        ]
[ 0.         0.         0.711365    0.         0.         0.        ]
[ 0.         0.         0.711365    0.         0.         0.        ]
[ 0.         0.         1.0375805   0.         0.         0.        ]
[ 0.         0.         1.0375805   0.         0.         0.        ]
[ 0.         0.         0.97582641  0.         0.         0.        ]
[ 0.         0.         0.97582641  0.         0.         0.        ]
[ 0.         0.         0.8939744   0.         0.         0.        ]
[ 0.         0.         0.8939744   0.         0.         0.        ]
[ 0.         0.         0.79377289  0.         0.         0.        ]
[ 0.         0.         0.79377289  0.         0.         0.        ]
[ 0.         0.         0.66656807  0.         0.         0.        ]
[ 0.         0.         0.         0.         0.         0.        ]
[ 0.         0.         0.46996183  0.         0.         0.        ]
[ 0.         0.         0.46996183  0.         0.         0.        ]
[ 0.         0.         0.711365    0.         0.         0.        ]
[ 0.         0.         0.711365    0.         0.         0.        ]
[ 0.         0.         0.711365    0.         0.         0.        ]
[ 0.         0.         1.0375805   0.         0.         0.        ]
[ 0.         0.         1.0375805   0.         0.         0.        ]
[ 0.         0.         0.97582641  0.         0.         0.        ]
[ 0.         0.         0.97582641  0.         0.         0.        ]
```

```
 [ 0.         0.         0.8939744  0.         0.         0.        ]
 [ 0.         0.         0.8939744  0.         0.         0.        ]
 [ 0.         0.         0.79377289  0.         0.         0.        ]
 [ 0.         0.         0.79377289  0.         0.         0.        ]
 [ 0.         0.         0.66656807  0.         0.         0.        ]
 [ 0.         0.         0.66656807  0.         0.         0.        ]
 [ 0.         0.         0.46996183  0.         0.         0.        ]
 [ 0.         0.         0.46996183  0.         0.         0.        ]]
difference between the two
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
```

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

[0. 0. 0. 0. 0. 0.]

```
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0.]]
```