16.888 Assignment 3, Part B

(b1) Simulation Completion

The simulation had been completed by the end of Assignment 2. It incorporated 4 different models for the wing, which were:

- Structural models (returns tip deflection and wing root stress)
- Weight models (returns wing weight)
- Aerodynamic models (returns lift and lift-to-drag ratio)
- Fuel volume models (returns total fuel volume of wing)

The 9 design variables are as follows:

- Normalized span
- Normalized root chord
- Taper ratio
- Geometric angle of attack (defined at 6 points along the wing)

The 4 constraints are imposed on:

- Wing tip deflection at maximum load factor
- Wing root stress (not active, and removed from model)
- Wing fuel volume
- Wing lift at cruise

The 2 objectives are:

- Minimizing wing weight
- Maximizing lift-to-drag.

The remaining open issues with the models implemented for the project are the actual fidelity of the models. There is a disparity between how well each model is able to evaluate the performance of a given wing.

- The structural models implement a very basic tapered beam-bending model. In the future, it might be better to implement some kind of finite-element model.
- The weight models are straightforward but accurate, using approximations for the densities and weights of composite spar-and-skin wings.
- The fuel volume model is very effective, being able to calculate the volume of any continuous 3D shape with linearly interpolated 2D sections.
- The aerodynamic model is also accurate for high aspect ratio, incompressible wings as the one we are designing.

(b2) Heuristic Optimization

(i) We implemented a simulated annealing algorithm as an initial optimization method for our system with 9 design variables, 4 constraints, and 2 objectives.
The reasoning for the choice of heuristic algorithm was as follows:

- Since we are limited computationally, the ability to set the initial temperature and cooling schedule enables us to effectively control the number of iterations that the algorithm will take to terminate.
- It allows for the random exploration of the design space while the temperature is high, while having improved local exploration as the system cools down.
- Furthermore, since our function evaluations are extremely expensive (~3s for every design), reducing the number of iterations required was key. This made the use of GAs impractical.
- The choice between particle swarms and simulated annealing was less clear. Having had less experience with simulated annealing, we chose to use as the initial method for the exploration of the design space. Furthermore, we believed that a simulated annealing scheme would take a fewer number of function calls compared to a particle swarm algorithm.

(ii) In the problem description, we were advised to optimize over a single objective for A3. But instead, we chose to implement a weighted sum objective function over both the 2 objectives and the 4 constraints, for an important reason. In an SA algorithm, constraint satisfaction is usually implemented in the perturbation function. However, to be able to evaluate some of our constraints, we would have to run computationally expensive evaluation functions that evaluate both the objectives and the constraints. This would have doubled the number of function evaluations we would have had to run. As a result, we implemented a weighted-sum objective function as follows:

$$cost = -LoD + weightCost + deltaCost + fuelCost + liftCost;$$

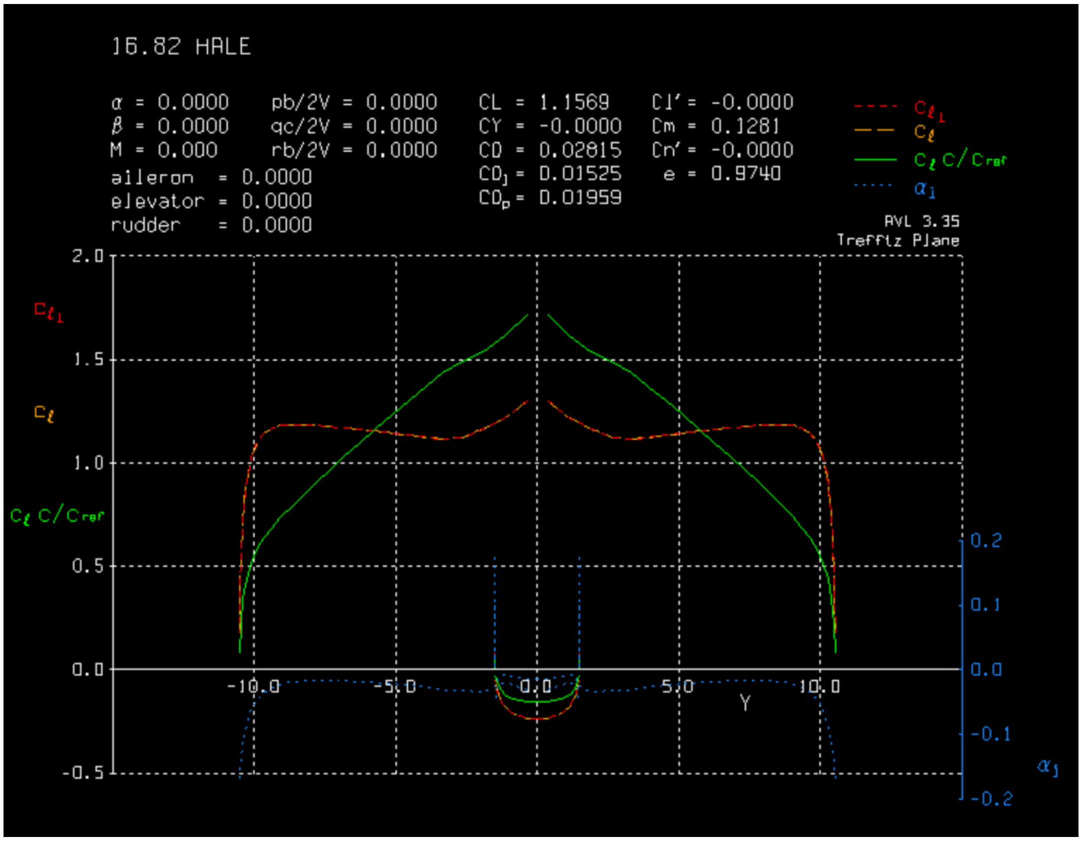
LoD and $weight$ are the two objectives, and they are acting in opposite directions for the cost function, which is to be minimized. The $deltaCost$, $fuelCost$, $liftCost$ are the components of the cost function that help satisfy the fuel volume, tip deflection, and cruise lift constraints on the wing. It was found that this cost scheme was useful in helping satisfy constraints without having to do more function calls.

This optimization scheme was implemented, with varying results. Unfortunately, it was difficult for us to refine the SA parameters since every run of the optimizer took over 15 minutes even for low initial temperatures and quick cooling schedules.

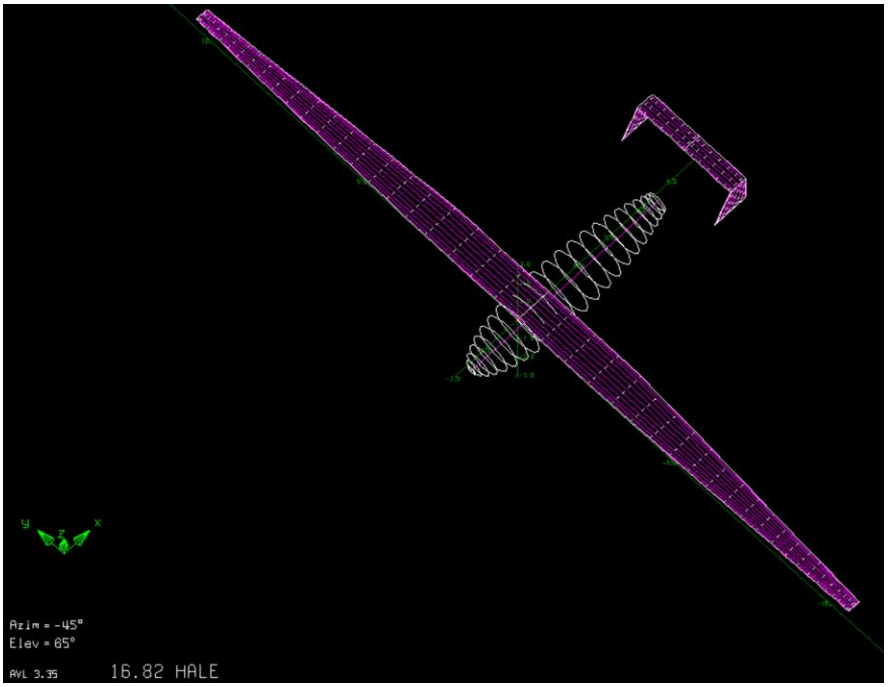
Our initial design (x0) had the following values for the objective:

- LoD: 41.0
- Wing weight: 33.5 N

Its lift distribution is given below.

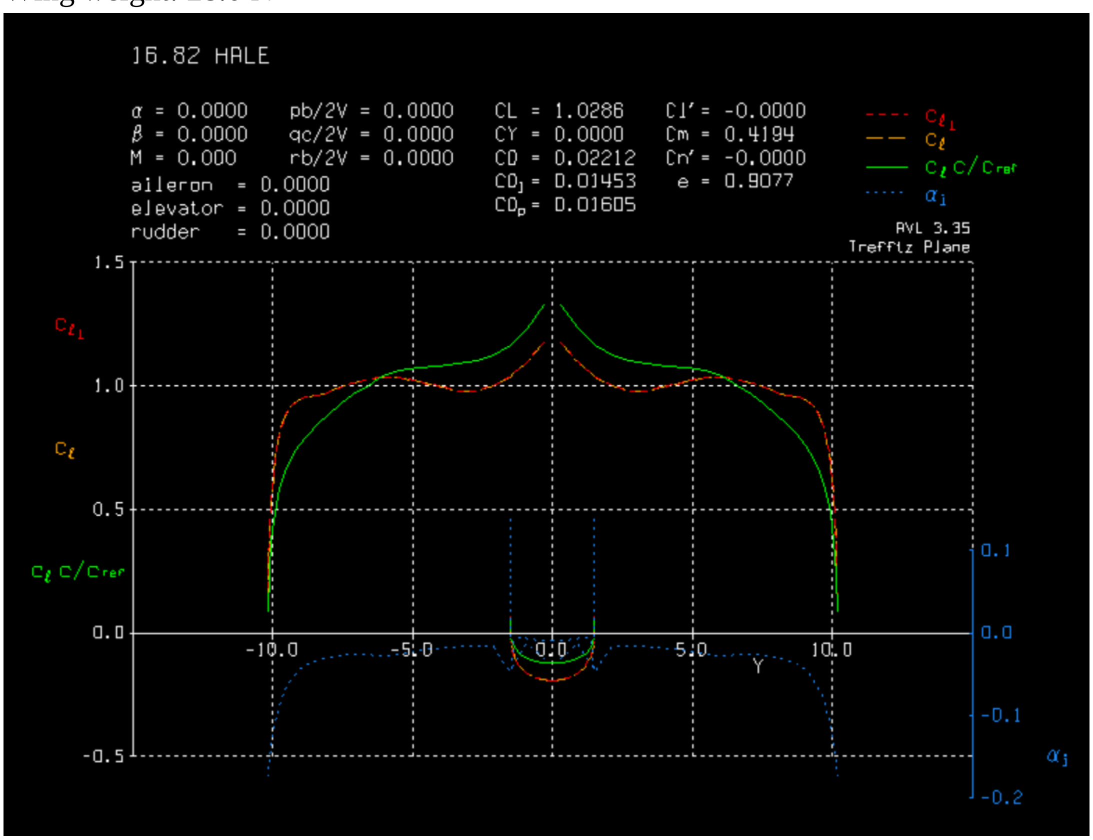


This wing has a suboptimal lift distribution (shown in green) because it doesn't approach elliptical lift, which is the theoretical limit on the best LoD for a given wing. It is also important to note that the spike in lift near the center of the wing is due to fuselage effects, which are considered within the aerodynamic model. Furthermore, this wing does not have enough volume to satisfy the fuel volume constraint of 0.0265 m^3. As a reference, here is the planform of the initial design, which has a span of 21 ft.

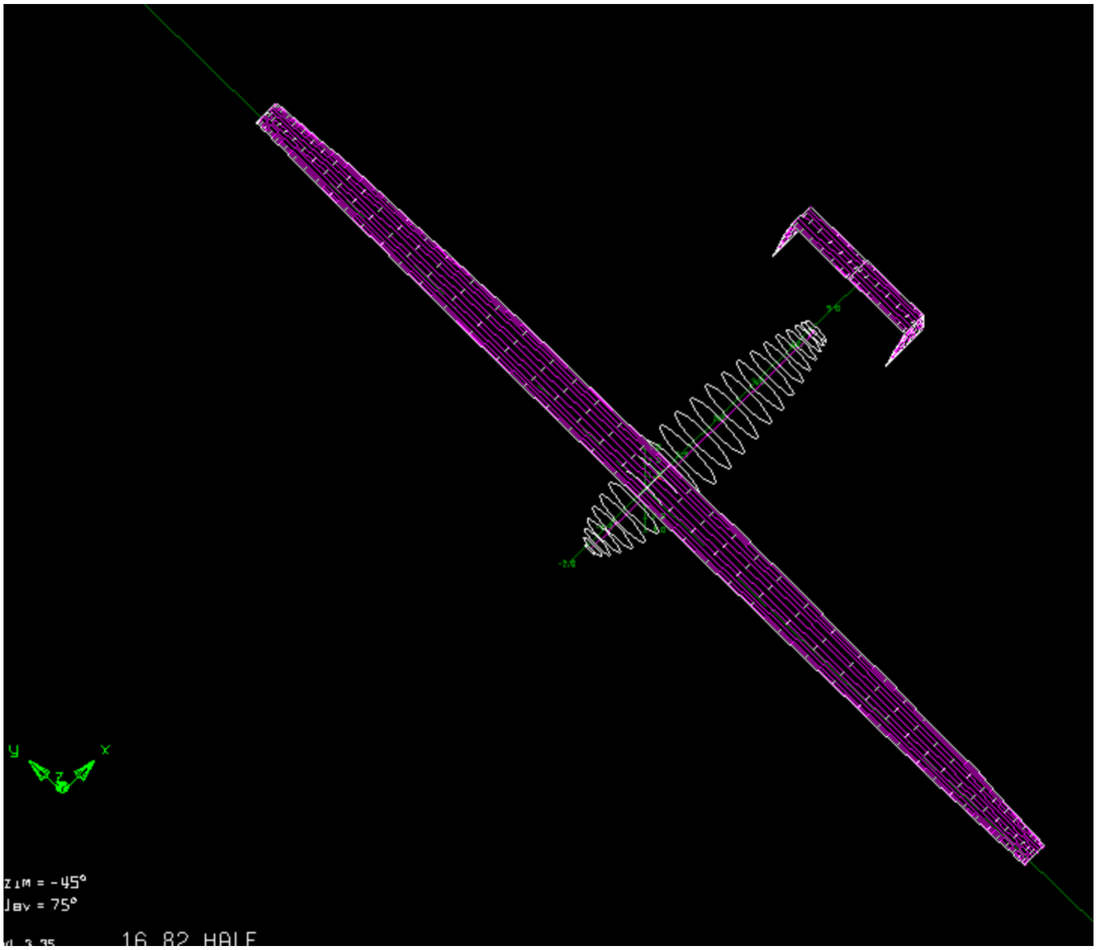


Result of a random sampling SA solution (T0 = 7800K, Tn+1/Tn = 0.9), over 1400 iterations (76 minute runtime):

- LoD: 46.0
- Wing weight: 28.0 N



This wing was found to satisfy all constraints, and give significant improvements in both LoD and weight! The lift distribution is approaching elliptical, which is predicted to be the optimal for L/D ratio from basic aerodynamics. The planform is given below.



The new wing has a lower taper ratio, presumably to be able to fulfill the fuel volume constraint. It has 96.6% the span, and 89.0% of the chord of the initial design, helping save weight and reducing the tip deflection of the wing.

The algorithm was run with a variety of initial temperatures, cooling rates, and perturbation functions. A large majority of these runs resulted in infeasible designs (not fulfilling the onstraints). As a result, it was difficult for the investigators to make a recommendation on which schedule was the best. However, an initial temperature of 7800K with an exponential cooling schedule with decay rate of 0.9 was found to be effective.

It was also observed that it is very important to pick effective perturbation functions for this problem. A number of different functions with and without memory were implemented, of which a large majority caused non-convergence in solution. This had to do with the complex dynamics of the weighted sum objective function. This would also assert that a gradient-based method would be more effective in designing this aircraft, since previous evaluations would be able inform the most logical search direction through finite differencing.

(iii) We are not confident that we have found a global optimum, but we are decently confident that we have found a solution that satisfies the constraints, and gains significant improvements over the initial design, while approaching the theoretical maximum efficiencies as defined by aerodynamic theory.

An interesting investigation would be to take the optimal aircraft from (ii), and a run a local SA algorithm to see whether it is truly a local optimum.

(b3) Gradient-based Optimization

(i) We expected our problem to be amenable to gradient-based optimization, so we chose the *Quasi-Newton* method. As we have a very large set of variables, an objective function costly to evaluate (in time) but do not have any information on analytical derivatives, this was estimated to be the most efficient way to do gradient-based optimization. Furthermore, the *Quasi-Newton* method is also implemented in MATLAB through *fmincon* and *fminunc*, speeding up the process. The gradient was not estimated through any provided function, but is numerically computed by the quasi-Newton method. Both *forward difference* and *central difference* were tested, not showing any large discrepancies.

Since the constraints are calculated at the same time as the objective function, it was hard to use *fmincon*'s built in constraint solver. As a result, the same penalty function as mentioned above was used for the objective function.

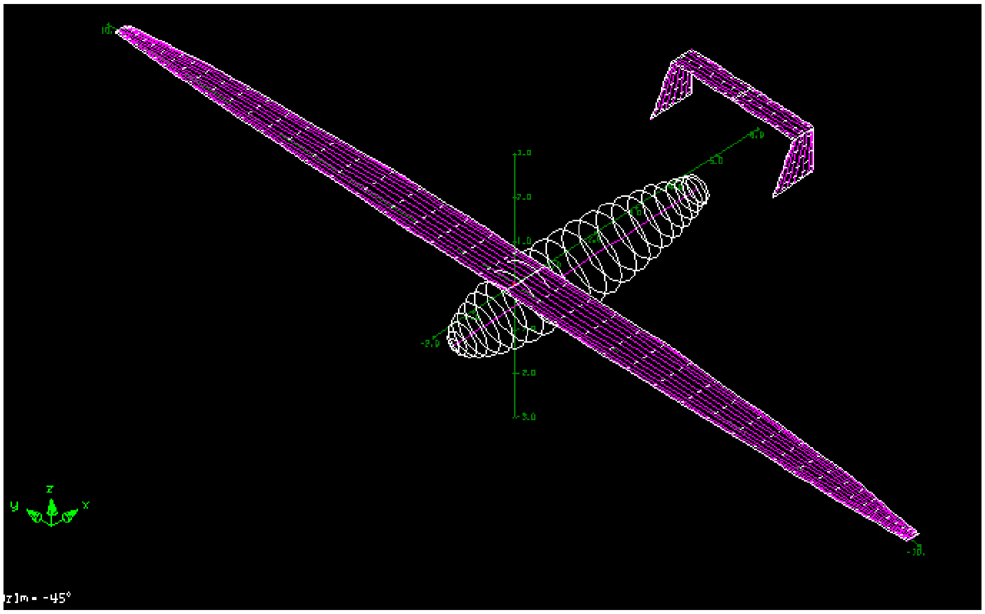
(ii) Unexpectedly results did not seem to converge (even after over a hundred iterations), but rather cycle around the same value. The results also appear to be highly dependent on initial conditions. The following tables show different series for different initial conditions:

Iter	b	Chord	λ	α_1	α_2	α_3	α_4	α_5	α_6	J
1	1.0000	1.0000	0.5000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	9.6259
11	1.0000	1.0000	0.5000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	9.6259
21	0.9547	0.9546	0.3756	-0.9934	-0.9934	-0.9934	-0.9934	-0.9934	-0.9934	-30.2886
31	0.9348	0.9322	0.3190	-0.9933	-0.9933	-0.9933	-0.9933	-0.9933	-0.9933	-35.8314
41	0.9447	0.9434	0.3473	-0.9933	-0.9933	-0.9933	-0.9933	-0.9933	-0.9933	-30.4481
51	0.9438	0.9502	0.3462	-0.9778	-0.9778	-0.9778	-0.9778	-0.9778	-0.9778	-37.0328
61	0.9355	0.9653	0.3231	-0.9757	-0.9757	-0.9757	-0.9757	-0.9757	-0.9757	-37.4318
71	**0.9342**	**0.9674**	**0.3193**	**-0.9684**	**-0.9684**	**-0.9684**	**-0.9684**	**-0.9684**	**-0.9684**	**-38.4173**

Iter	b	Chord	λ	α_1	α_2	α_3	α_4	α_5	α_6	J
1	*0.80*	*0.8*	*0.5*	*0*	*0*	*0*	*0*	*0*	*0*	*2810.67*
11	0.98	0.98	0.5	0	0	0	0	0	0	43.08
21	0.98	0.985326	0.531094	0	0	0	0	0	0	45.39
31	0.98	0.980042	0.500243	0	0	0	0	0	0	43.06
41	0.98	0.979994	0.500291	0	0	0	0	0	0	43.05
51	0.89	0.930272	0.550012	0	0	0	0	0	0	-1.94
61	0.85	0.88217	0.598114	0	0	0	0	0	0	-15.41
71	0.87	0.906221	0.574063	0	0	0	0	0	0	1407.44
81	0.851021	0.88463	0.595655	0	0	0	0	0	0	-18.40
91	0.855561	0.896212	0.584072	0	0	0	0	0	0	-22.09
101	0.848611	0.90288	0.577404	0	0	0	0	0	0	-23.23
111	0.836519	0.917188	0.563096	0	0	0	0	0	0	-24.73
121	0.816403	0.937949	0.542335	0	0	0	0	0	0	-20.25
131	**0.836436**	**0.916779**	**0.563505**	**0**	**0**	**0**	**0**	**0**	**0**	**-24.86**

Iter	X	Chord	λ	α_1	α_2	α_3	α_4	α_5	α_6	J
1	1.2000	1.2000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	623.0163
11	1.0200	1.0200	0.5500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	132.1996
21	0.9200	0.9199	0.2764	0.9894	0.9894	0.9894	0.9894	0.9894	0.9894	27.53608
31	0.8971	0.8935	0.2076	0.9744	0.9744	0.9744	0.9744	0.9744	0.9744	3.666405
41	0.8781	0.8702	0.1572	0.9524	0.9524	0.9524	0.9524	0.9524	0.9524	-2.16214
51	0.8862	0.8815	0.1743	0.9326	0.9326	0.9326	0.9326	0.9326	0.9326	-8.83068
61	0.8806	0.8886	0.1593	0.8797	0.8797	0.8797	0.8797	0.8797	0.8797	-12.5787
71	0.8636	0.9150	0.1197	0.8344	0.8344	0.8344	0.8344	0.8344	0.8344	-13.9287
81	0.8454	0.9509	0.0761	0.8353	0.8353	0.8353	0.8353	0.8353	0.8353	-14.1532
91	0.8002	1.0310	0.0125	0.7895	0.7895	0.7895	0.7895	0.7895	0.7895	-11.0509
101	0.8152	1.0364	0.0253	0.8237	0.8237	0.8237	0.8237	0.8237	0.8237	-14.4188
111	0.8137	1.0308	0.0226	0.7697	0.7697	0.7697	0.7697	0.7697	0.7697	-18.9925
121	0.8135	1.0149	0.0224	0.7696	0.7696	0.7696	0.7696	0.7696	0.7696	-17.8177
131	0.8136	1.0268	0.0225	0.7697	0.7697	0.7697	0.7697	0.7697	0.7697	-19.9506
141	0.8136	1.0263	0.0225	0.7697	0.7697	0.7697	0.7697	0.7697	0.7697	-19.9957
151	0.8136	1.0266	0.0225	0.7697	0.7697	0.7697	0.7697	0.7697	0.7697	4474.45
161	**0.8136**	**1.0265**	**0.0225**	**0.7697**	**0.7697**	**0.7697**	**0.7697**	**0.7697**	**0.7697**	**-20.0182**

The geometry of the best configuration point found for the first initial condition case can be seen in the figure below.



(iii) Below is a sensitivity analysis at the "optimal" points found in (ii) for the first initial conditions. The calculation is carried out by holding all but one variable fixed, and changing this variable by Δx_i. Δx_i is decreased until a linear approximation about the point is good enough(less than 0.1 residual norm). Finally the gradient is easily computed as $\Delta f/2\Delta x_i$ as shown in the following table :

x_i	$2\Delta x_i$	Δf_i	$\frac{\partial f}{\partial x_i}$
b	0.02	-27.0356	-1351.78
Chord	0.02	-23.7262	-1186.31
λ	0.04	-11.5335	-288.34
$\boldsymbol{\alpha_1}$	0.2	-4.1672	-20.84
$\boldsymbol{\alpha_2}$	0.2	-8.4771	-42.39
$\boldsymbol{\alpha_3}$	0.2	-6.3534	-31.77
$\boldsymbol{\alpha_4}$	0.2	-3.2470	-16.24
$\boldsymbol{\alpha_5}$	0.2	-1.3462	-6.73
$\boldsymbol{\alpha_6}$	0.2	-0.3427	-1.71

As can be seen, the points are not truly optimal. Since the objective function is so expensive to evaluate a long run for the gradient method has still to be carried out, in the hope of potentially finding a better optimal point.