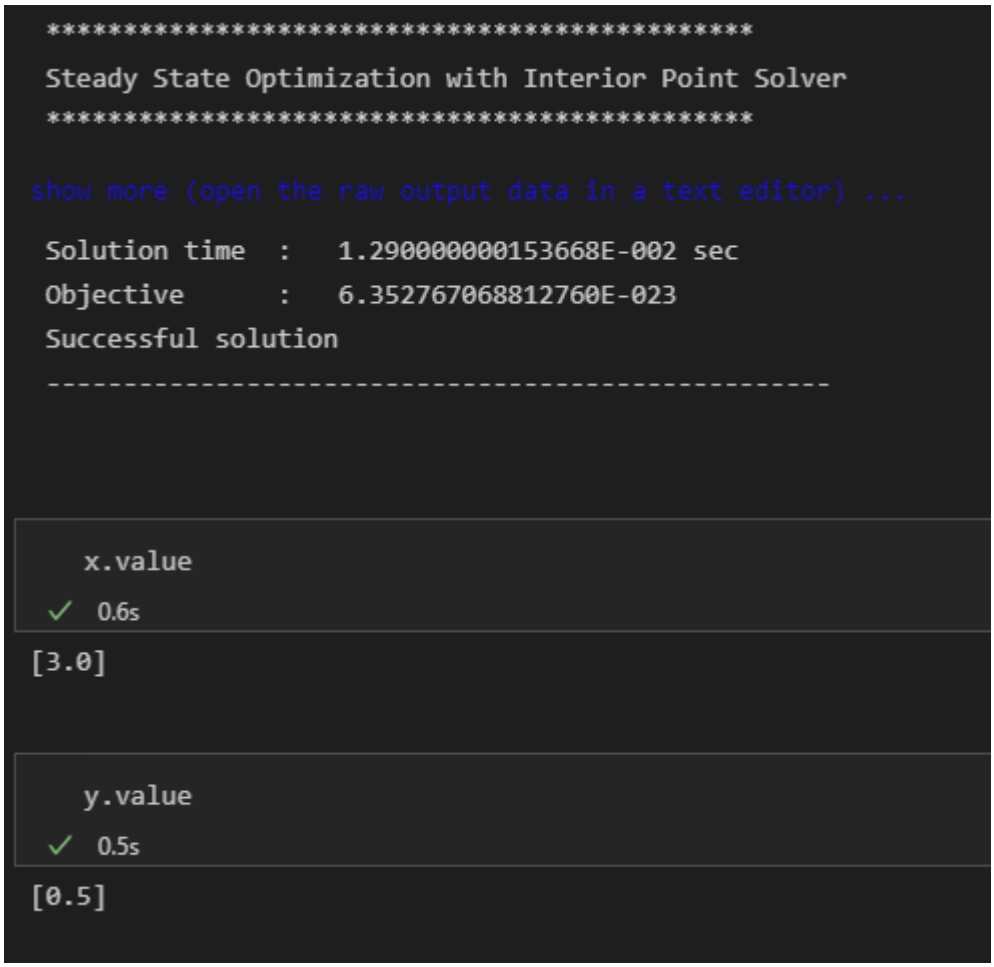
Testing optimization tools

Task 1:

Unconstrained functions:

For the unconstrained problems the Gekko library for python were used.

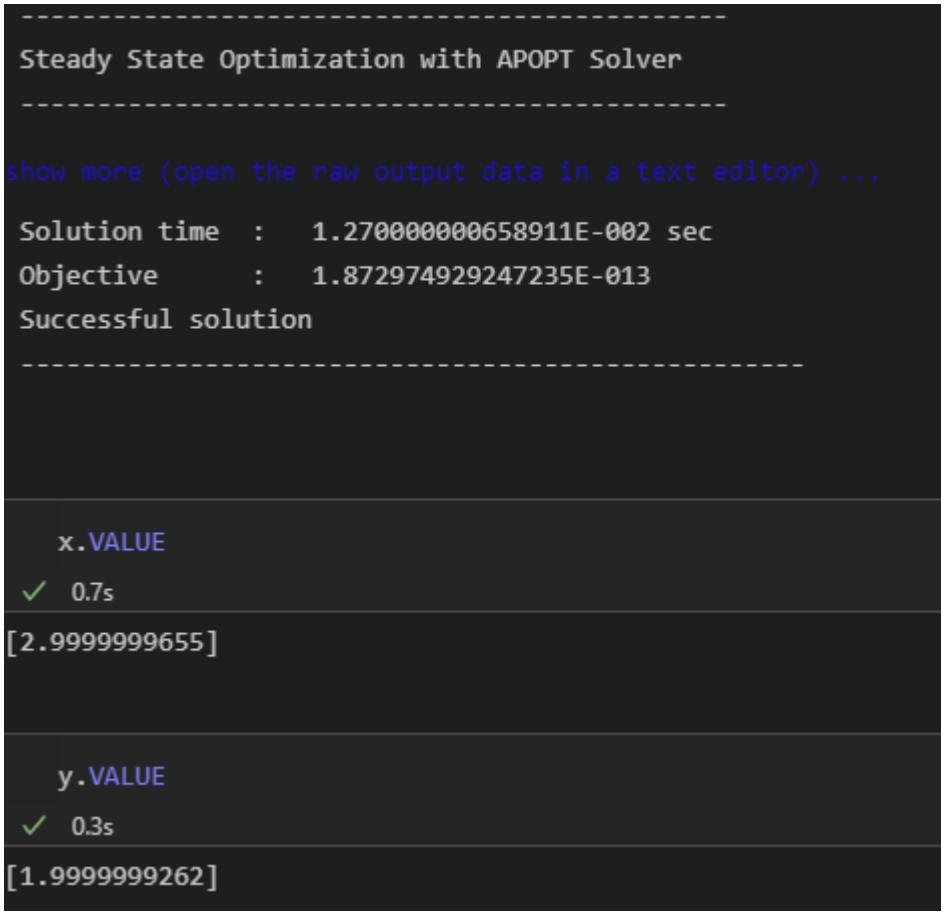
Beal function



Correct! Solved with Steadu State Optimization with Interior Point Solver. f(3, 0.5) = 0 .

The solution (Objective in the solver) shows a small, this is due to a rounding errors in code, it can be rounded down to 0.

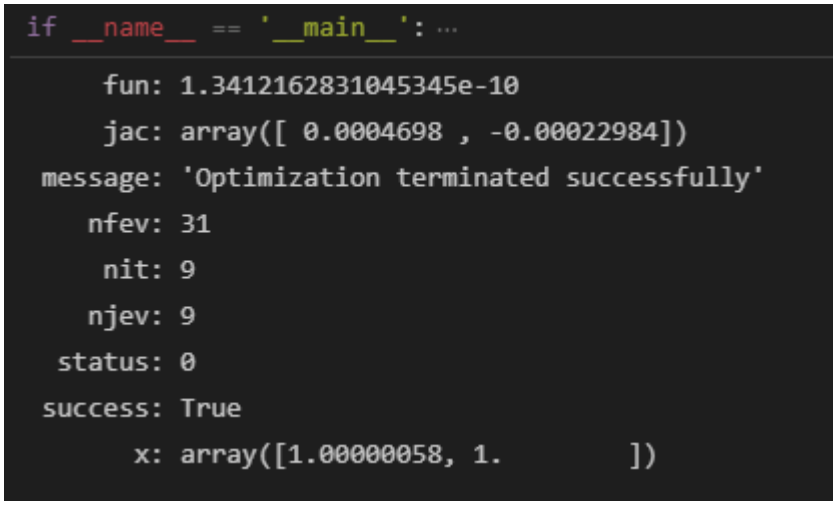
Himmelbaus



Correct! The solver found the solution f(3, 2) = 0. This is one of 4 solutions to the problem. The solver did not reckognize this. The program had the same rounding error as the previous task

Constrained Optimization:

Rosenbrock, cubic and line constrained



Correct answer! The SciPy libraryr in python is used in this problem. It uses Sequential Least Squares Programming to solve the problem. f(1,1) = 0

RosenBrock, disk constraines

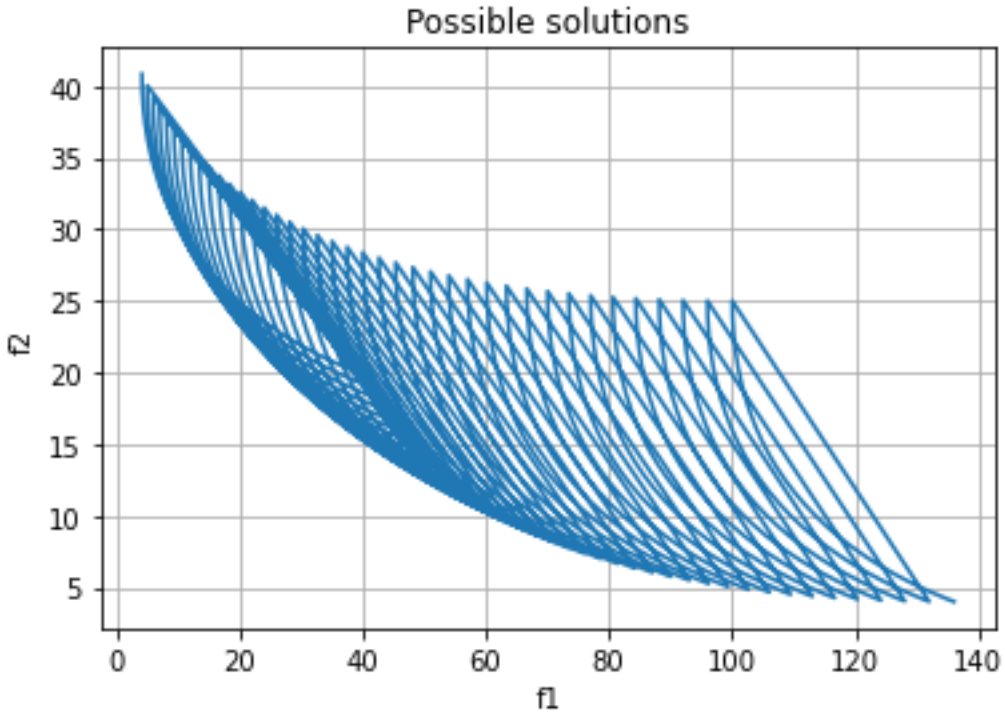
```
if __name__ == '__main__': ...
     fun: 3.608907291128452e-08
     jac: array([ 0.00229613, -0.0013219 ])
 message: 'Optimization terminated successfully'
    nfev: 71
     nit: 22
    njev: 22
  status: 0
 success: True
       x: array([0.99982193, 0.99963726])
```

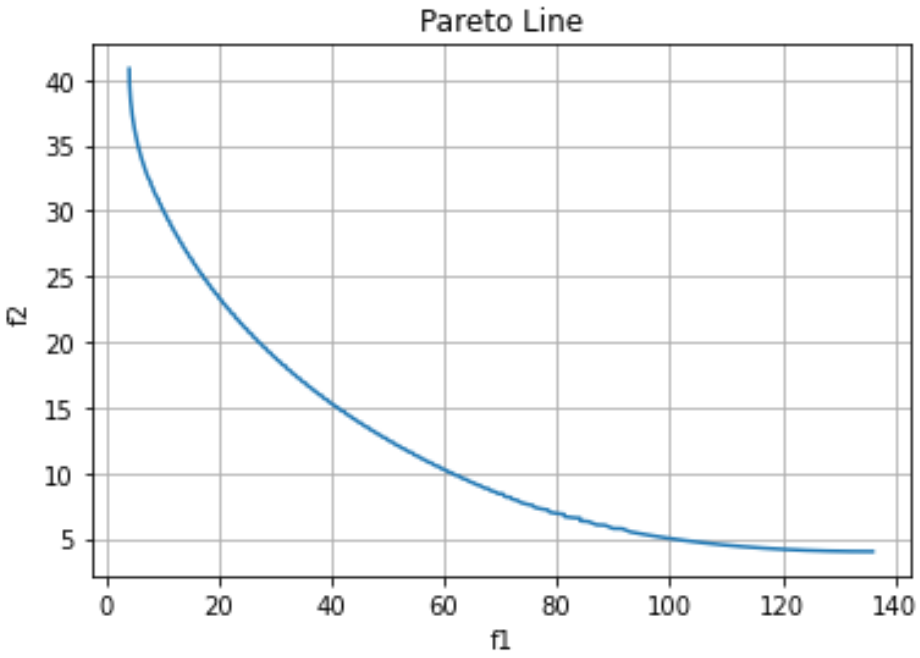
The same method is used in this problem as the previous one. It finds the right answer, altough with a rounding error. f(0,0) = 1

Task 2:

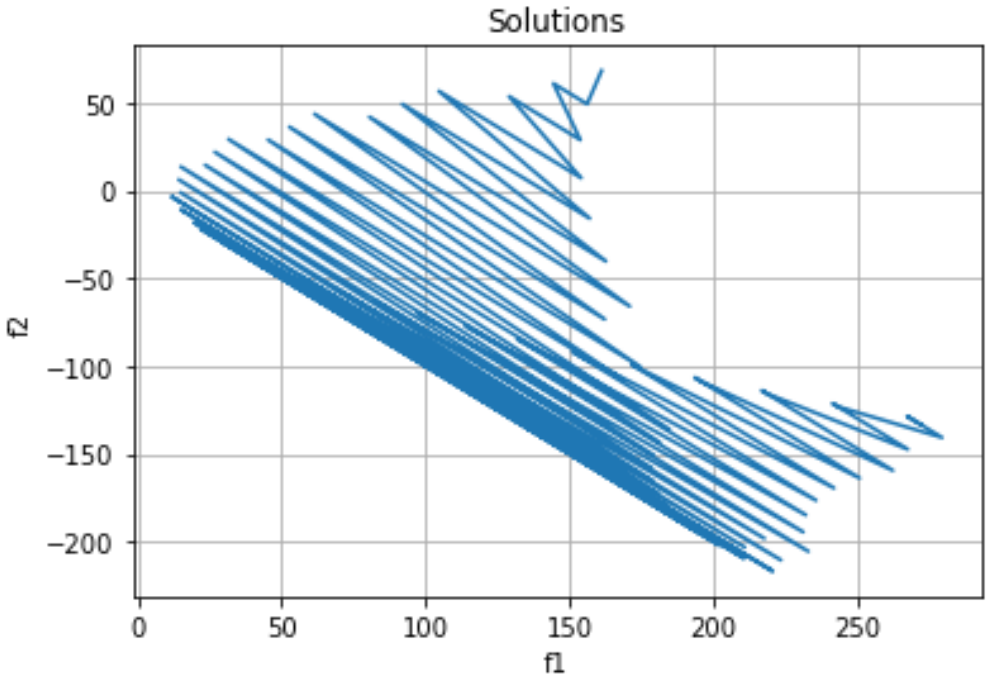
With the possible solutions plotted, its clear where the pareto line should go. A python function for generations a pareto line werre found at https://code.activestate.com/recipes/578230-pareto-front/

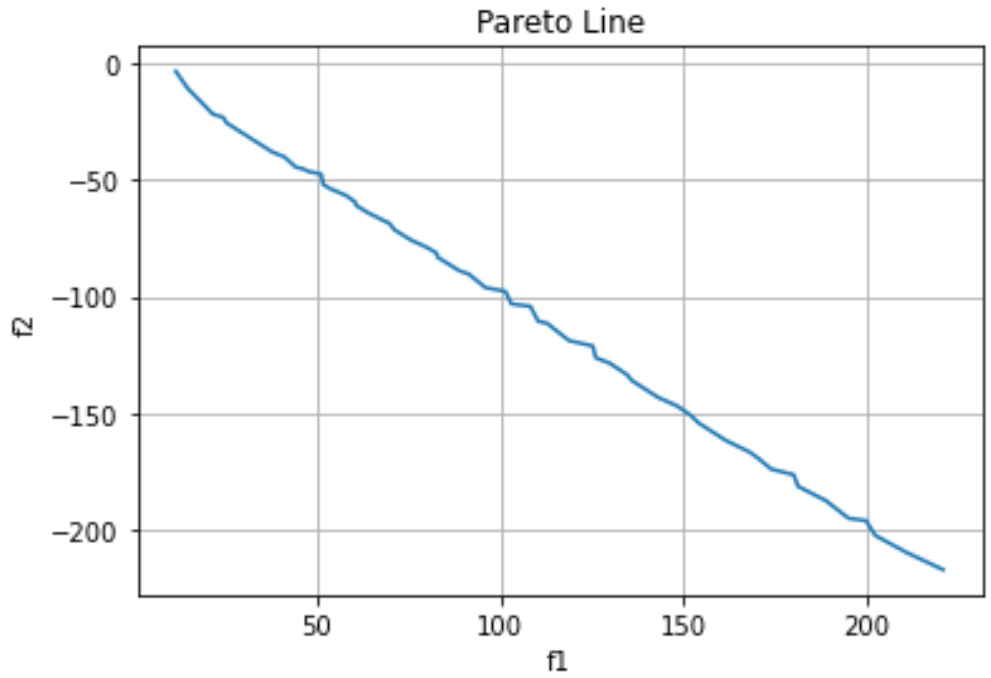
Binh and Korn function





Chankong and Haimes:





For both of the problems we can easily see that the correct pareto line is plotted. However, the solution is not perfect. Due to errors in the calcutalion at the start. The first 5% of the data had to be excluded-