

## SimPleAC

In this example, we modify the simple wing model proposed by Prof. Warren Hoburg in his thesis to create a simple aircraft model incorporating signomial programming (SP). Firstly, we initialize the SimPleAC model, and define all of its constants and free variables.

```
from gpkit import Model, Variable, SignomialsEnabled, VarKey, units
import numpy as np
import matplotlib.pyplot as plt

class SimPleAC(Model):
    def setup(self):
        # Env. constants
        g          = Variable("g", 9.81, "m/s^2", "gravitational acceleration")
        mu         = Variable("\\mu", 1.775e-5, "kg/m/s", "viscosity of air", pr=4.)
        rho        = Variable("\\rho", 1.23, "kg/m^3", "density of air", pr=5.)
        rho_f      = Variable("\\rho_f", 817, "kg/m^3", "density of fuel")

        # Non-dimensional constants
        C_Lmax     = Variable("C_{L,max}", 1.6, "-", "stall CL", pr=5.)
        e          = Variable("e", 0.92, "-", "Oswald efficiency factor", pr=3.)
        k          = Variable("k", 1.17, "-", "form factor", pr=10.)
        N_ult      = Variable("N_{ult}", 3.3, "-", "ultimate load factor", pr=15.)
        S_wetratio = Variable("\\frac{S}{S_{wet}}", 2.075, "-",
            "wetted area ratio", pr=3.)
        tau        = Variable("\\tau", 0.12, "-",
            "airfoil thickness to chord ratio", pr=10.)
        W_W_coeff1 = Variable("W_{W_{coeff1}}", 2e-5, "1/m",
            "wing weight coefficient 1", pr= 30.)
        W_W_coeff2 = Variable("W_{W_{coeff2}}", 60., "Pa",
            "wing weight coefficient 2", pr=10.)
        p_labor    = Variable('p_{labor}', 1., '1/min', 'cost of labor', pr = 20.)

        # Dimensional constants
        Range      = Variable("Range", 3000, "km", "aircraft range")
        TSFC       = Variable("TSFC", 0.6, "1/hr",
            "thrust specific fuel consumption")
        V_min      = Variable("V_{min}", 25, "m/s", "takeoff speed", pr=20.)
        W_0        = Variable("W_0", 6250, "N",
            "aircraft weight excluding wing", pr=20.)

        # Free Variables
        LoD        = Variable('L/D', '-', 'lift-to-drag ratio')
        D          = Variable("D", "N", "total drag force")
        V          = Variable("V", "m/s", "cruising speed")
```

```

W          = Variable("W", "N", "total aircraft weight")
Re         = Variable("Re", "-", "Reynold's number")
CDA0       = Variable("(CDA0)", "m^2", "fuselage drag area")
C_D        = Variable("C_D", "-", "drag coefficient")
C_L        = Variable("C_L", "-", "lift coefficient of wing")
C_f        = Variable("C_f", "-", "skin friction coefficient")
W_f        = Variable("W_f", "N", "fuel weight")
V_f        = Variable("V_f", "m^3", "fuel volume")
V_f_avail  = Variable("V_{f_{avail}}", "m^3", "fuel volume available")
T_flight   = Variable("T_{flight}", "hr", "flight time")

# Free variables (fixed for performance eval.)
A          = Variable("A", "-", "aspect ratio", fix = True)
S          = Variable("S", "m^2", "total wing area", fix = True)
W_w        = Variable("W_w", "N", "wing weight", fix = True)
W_w_strc   = Variable('W_w_strc', 'N', 'wing structural weight', fix = True)
W_w_surf   = Variable('W_w_surf', 'N', 'wing skin weight', fix = True)
V_f_wing   = Variable("V_f_wing", 'm^3', 'fuel volume in the wing', fix = True)
V_f_fuse   = Variable('V_f_fuse', 'm^3', 'fuel volume in the fuselage', fix = True)
constraints = []

```

## Weight and Lift model

The simple aircraft borrows a majority of its aerodynamic model from Prof. Warren Hoburg's thesis, with some minor adjustments to make the model a SP. The aircraft is assumed to be in steady-level flight so that thrust equals drag, and the lift equals the weight.

The total weight of the aircraft is the sum of the payload weight  $W_0$ , the wing weight  $W_w$ , and the fuel weight  $W_f$ .

$$W \geq W_0 + W_w + W_f \quad (1)$$

We use the naive  $W \leq L$  model below for steady state flight:

$$W_0 + W_w + 0.5W_f \leq \frac{1}{2}\rho SC_L V^2 \quad (2)$$

where the lift of the aircraft is equal to weight of the aircraft with half-fuel. We would also like the fully-fueled aircraft to be able to fly at a minimum speed of  $V_{min}$  without stalling, so we add the following constraint:

$$W \leq \frac{1}{2}\rho V_{min}^2 SC_{L_{max}} \quad (3)$$

The time of flight of the aircraft, which is a useful metric of performance, is simply the range over velocity:

$$T_{flight} \geq \frac{Range}{V} \quad (4)$$

The lift-to-drag ratio is also defined:

$$L/D = \frac{C_L}{C_D} \quad (5)$$

```
# Weight and lift model
constraints += [W >= W_0 + W_w + W_f,
                W_0 + W_w + 0.5 * W_f <=
                    0.5 * rho * S * C_L * V ** 2,
                W <= 0.5 * rho * S * C_Lmax * V_min ** 2,
                T_flight >= Range / V,
                LoD == C_L/C_D]
```

## Thrust and drag model

We assume a constant thrust specific fuel consumption (TSFC) for the ‘engine’ of the aircraft, which is assumed to provide as much thrust as needed. Since  $T \geq D$ :

$$W_f \geq TSFC \times T_{flight} \times D \quad (6)$$

the fuel weight required is the product of the TSFC, time of flight, and the total drag on the aircraft. The drag of the aircraft is the product of dynamic pressure ( $\frac{1}{2}\rho V^2$ ), the planform area  $S$ , and the coefficient of drag of the aircraft:

$$D \geq \frac{1}{2}\rho V^2 S C_D \quad (7)$$

The drag coefficient of the aircraft is assumed to be the sum of the fuselage drag, the wing profile drag, and the wing induced drag coefficients:

$$C_D \geq C_{D_{fuse}} + C_{D_{wpar}} + C_{D_{ind}} \quad (8)$$

The fuselage drag is a function of its drag area  $CDA_0$  and the planform area of the wing:

$$C_{D_{fuse}} = \frac{CDA_0}{S} \quad (9)$$

where the  $CDA_0$  is linearly proportional to the volume of fuel in the fuselage:

$$V_{fuse} \leq CDA_0 \times 10 \text{ [meters]} \quad (10)$$

Note that we correct the dimensionality of the volume here, since GPkit automatically checks units.

The wing profile drag is the product of the form factor, the friction drag coefficient, and the wetted area ratio of the wing,

$$C_{D_{wpar}} = kC_f S_{wetratio} \quad (11)$$

The Reynolds number of the aircraft wing is approximated

$$Re \leq \frac{\rho}{\mu} V \sqrt{\frac{S}{AR}} \quad (12)$$

and used to find the friction drag coefficient of wing. We approximate the  $C_f$  by assuming a turbulent Blasius flow over a flat plate as below:

$$C_f \geq \frac{0.074}{Re^{0.2}} \quad (13)$$

The induced drag of the wing is calculated with a span efficiency factor  $e$ , and is a function of the  $C_L$  and aspect ratio  $AR$  of the wing.

$$C_{D_{induced}} = \frac{C_L^2}{\pi AR e} \quad (14)$$

```
# Thrust and drag model
C_D_fuse = CDA0 / S
C_D_wpar = k * C_f * S_wetratio
C_D_ind = C_L ** 2 / (np.pi * A * e)
constraints += [W_f >= TSFC * T_flight * D,
                D >= 0.5 * rho * S * C_D * V ** 2,
                C_D >= C_D_fuse + C_D_wpar + C_D_ind,
                V_f_fuse <= 10*units('m')*CDA0,
                Re <= (rho / mu) * V * (S / A) ** 0.5,
                C_f >= 0.074 / Re ** 0.2]
```

## Fuel volume model

The fuel volume model is the main difference between simpleWing and SimpleAC, and introduces the only signomial constraints in the model. Firstly we define the required fuel volume using fuel density  $\rho_f$ .

$$V_f = \frac{W_f}{\rho_f g} \quad (15)$$

We consider wing fuel tanks and fuselage fuel tanks. The fuselage fuel was defined in the aerodynamic model, where the fuel volume contributes to drag.

$$V_{f_{wing}}^2 \leq 0.0009 \frac{S \tau^2}{AR} \quad (16)$$

The signomial constraint is introduced here, where the available fuel volume is upper-bounded by the sum of the fuselage and wing fuel volumes.

$$V_{f_{avail}} \leq V_{f_{wing}} + V_{f_{fuse}} \quad (17)$$

We constrain the total fuel volume to be less than the available fuel volume.

$$V_{f_{avail}} \geq V_f \quad (18)$$

```
# Fuel volume model
with SignomialsEnabled():
    constraints += [V_f == W_f / g / rho_f,
                    V_f_wing**2 <= 0.0009*S**3/A*tau**2,
                    # linear with b and tau, quadratic with chord
                    V_f_avail <= V_f_wing + V_f_fuse, #[SP]
                    V_f_avail >= V_f]
```

## Wing Weight Build-Up

The wing surface weight is a function of the planform area of the wing.

$$W_{w_{surf}} \geq W_{w_{coeff2}} S \quad (19)$$

The wing structural weight is a complex posynomial expression that takes into account the root bending moment and shear relief due to presence of fuel in the wings.

$$W_{w_{strc}}^2 \geq \frac{W_{w_{coeff1}}^2}{\tau^2} (N_{ult}^2 AR^3 ((W_0 + \rho_f g V_{f_{fuse}}) W S)) \quad (20)$$

The total wing weight is lower-bounded.

$$W_w \geq W_{w_{surf}} + W_{w_{strc}} \quad (21)$$

```

# Wing weight model
constraints += [W_w_surf >= W_W_coeff2 * S,
                W_w_strc**2. >= W_W_coeff1**2./ tau**2. *
                (N_ult**2. * A ** 3. * ((W_0+V_f_fuse*g*rho_f) * W * S)),
                W_w >= W_w_surf + W_w_strc]

return constraints

```

## Running the model

The main method within `SimpleAC.py` allows the model to be run directly from a terminal with the command `python SimpleAC.py`.

```

if __name__ == "__main__":
    m = SimpleAC()
    m.cost = m['W_f']
    sol = m.localsolve(verbosity = 4)
    print sol.table()

```

## Valid objective functions

We have tested a variety of potential objectives for the `SimpleAC` model, some of which are as follows:

- $W_f$ : Fuel weight, the default objective.
- $W$ : Total aircraft weight. Like fuel weight, but also adding extra cost for airframe weight.
- $D$ : Drag.
- $\frac{W_f}{T_{flight}}$ : Product of the fuel weight and the inverse of the time of flight.
- $W_f + c \times T_{flight}$ : A linear combination of fuel weight and time of flight. This can simulate recurring costs (fuel and labor), and yield interesting results.