

Introduction

In this example, we express the area moment of inertia (I) of a hollow cylindrical section as a functional of its physical dimensions to formulate the bending section design problem in a GP form.

```
import numpy as np
from gpkits import Variable, Model, units
from gpkits.constraints.tight import Tight
import matplotlib.pyplot as plt

Tight.reltol = 1e-2

class Beam(Model):
    def __init__(self):
```

Hollow Cylinder Bending Inertia Equations

The inertia of a beam describes the stiffness of the beam in bending. If we desire to create a beam bending model in GP, it is a critical quantity. Since we are also interested in structural efficiency, we want to minimize the cross-sectional area of the section, which describes the weight of the beam along with the density of the beam and its length.

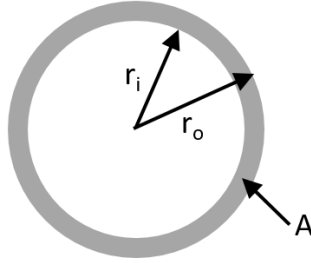


Figure 1: The cross-sectional parameters of a hollow cylindrical section

The equations for the inertia and cross-sectional area of a hollow cylindrical section are shown in Figure 1 are:

$$A = \pi(r_o^2 - r_i^2) \quad (1)$$

$$I = \frac{1}{4}\pi(r_o^4 - r_i^4) \quad (2)$$

where r_o is the outer radius, r_i is the inner radius, A is the cross-sectional area, and I is the moment of inertia of the section. The form as-is is not GP compatible by the following reasoning.

If we say that greater I is good and greater A is bad, then the right bounding inequalities for the expressions above are:

$$A \geq \pi(r_o^2 - r_i^2) \quad (3)$$

$$I \leq \frac{1}{4}\pi(r_o^4 - r_i^4) \quad (4)$$

since we are trying to maximize I and minimize A . Equation 4 is GP-compatible, since we can add $\frac{1}{4}\pi r_i^4$ to both sides of the equation and have a valid posynomial. However, no such transformation is possible for Equation 3. And if we changed the sign of the inequality in Equation 3, then A would go off to 0 ($10^{-\infty}$), a non-physical solution.

GP Formulation

We begin by rearranging Equation 3 to get the following:

$$\frac{A}{\pi} = r_o^2 - r_i^2 \quad (5)$$

We then split Equation 4 as shown in Equation 6. Note that we limit I to be less than the right hand side (RHS) of the equation, for later conversion into posynomial form. This allows us to use the cross-sectional area relation from Equation 5 to describe the inertia, yielding Equation 7.

$$I \leq \frac{1}{4}\pi(r_o^2 - r_i^2)(r_o^2 + r_i^2) \quad (6)$$

$$I \leq \frac{1}{4}A(r_o^2 + r_i^2) \quad (7)$$

We use the geometric mean to make an approximation for $(r_o^2 + r_i^2)$.

$$2r_o r_i \leq r_o^2 + r_i^2 \quad (8)$$

Substituting Equation 8 into Equation 7, we can get a conservative bound on the moment of inertia that is GP-compatible.

$$I \leq \frac{Ar_o r_i}{2} \quad (9)$$

Rearranging the area relation in Equation 3:

$$r_i^2 + \frac{A}{\pi} \leq r_o^2 \quad (10)$$

We substitute for r_i from Equation 11 in Equation 12:

$$r_i^2 \geq \frac{4I^2}{A^2 r_o^2} \quad (11)$$

$$\frac{4I^2}{A^2 r_o^2} + \frac{A}{\pi} \leq r_o^2 \quad (12)$$

The final formulation is as follows:

$$\bar{W} \geq \rho A \quad (13)$$

$$\frac{4I^2}{A^2 r_o^2} + \frac{A}{\pi} \leq r_o^2 \quad (14)$$

```

I      = Variable("I", "m^4", "Moment of inertia")
A      = Variable("A", "m^2", "Cross-sectional area")
ro     = Variable("r_o", "m", "Outer radius")
romax  = Variable("r_o_{max}", "m", "Maximum outer radius")

constraints = [Tight([4*I**2/(A**2*ro**2) + A/np.pi <= ro**2]),
               ro <= romax]

# Minimizing cross-sectional area => min(weight)
objective = A

Model.__init__(self,objective,constraints)

```

This formulation is GP-compatible, because the equations are of monomial and posynomial forms, and we have the right bounds on the two parameters we care about. Greater moment of inertia I is good, but it is upper bounded by the outer radius. And area A is bounded because it is directly proportional to the weight of the structure in question, which we want to minimize. Using Equations 13 and 14, we can give conservative approximations for the area moments of inertia of hollow cylindrical beams, allowing us to calculate their deflections under bending loads.

However, it is important to note that the geometric mean has been used in Equation 8 to give a conservative bound on the moment of inertia, which means that the solution is not exact. Furthermore, it doesn't seem absolutely clear that the constraint in Equation 14 will be always tight, but results from problems that use this models show empirically that it is always tight. The engineering intuition holds!

```
if __name__ == "__main__":
    M = Beam()
    M.substitutions.update({"I": 10**-4})
    M.substitutions.update({"r_o_{max}": 0.2})
    sol = M.sweep({"I": np.logspace(-10, -3, 100)}, skipsweepfailures=True)
    print sol.table()

    I, A, ro = sol("I"), sol("A"), sol("r_o")
    plt.semilogx(I, (4*I**2/(A**2*ro**2) + A/np.pi - ro**2)/(ro**2))
    plt.xlabel("Moment of inertia [m^-4]")
    plt.ylabel("Tightness of constraint")
    plt.ylim([-2.5e-5, 1e-5])
    plt.title("Checking constraint tightness (normalized by ro**2)")
    plt.savefig("tightness.png")
    # plt.show()
```

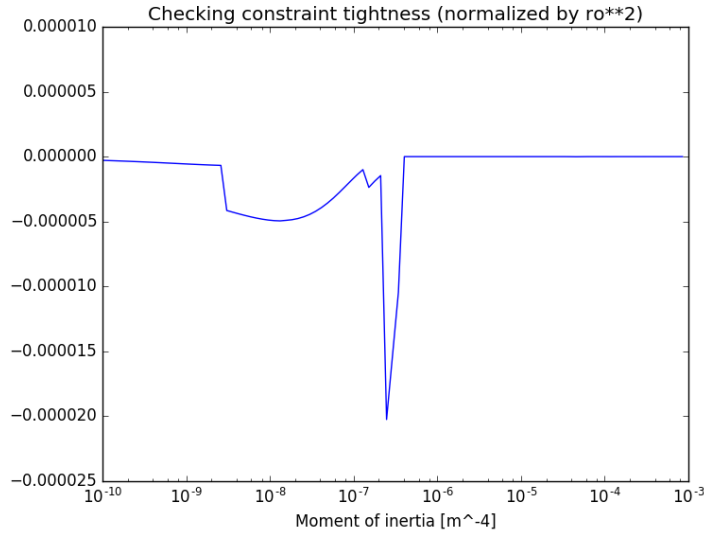


Figure 2: Verification that Equation 14 stays tight over a range of I values