lab08-error

March 24, 2019

In [1]:
```
from sympy import *
from calc import Symbol
init_printing()
```

In [2]:
```
R1 = Symbol.from_mean('R_1', 28.55, 30.45) / 200
R2 = Symbol.from_mean('R_2', 29.25, 29.9) / 200
d = Symbol('d', val=15.95, err=0.1) / 100
N = 130
mu = 4 * Pow(10, -7) * pi
p = Rational(3, 2)
D = N * mu / 2 * (R1**2 / ((d/2)**2 + R1**2)**p + R2**2 / ((d/2)**2 + R2**2)**p)
```

In [3]:
```
D.simp()
D.display(name="D")
```

Numeric:

$$D = 0.000753566902559207 \pm 0.0000393646598980527$$

Symbolic:

$$\text{Value: } D = \frac{0.0052\pi R_1^2}{\left(R_1^2 + d^2\right)^{\frac{3}{2}}} + \frac{0.0052\pi R_2^2}{\left(R_2^2 + d^2\right)^{\frac{3}{2}}}$$

$$= \frac{0.0052\pi 29.5^2}{\left(15.95^2 + 29.5^2\right)^{\frac{3}{2}}} + \frac{0.0052\pi 29.575^2}{\left(15.95^2 + 29.575^2\right)^{\frac{3}{2}}}$$

$$\text{Error: } \sigma_D = 104.0\pi \sqrt{\frac{R_1^2 \left(R_2^2 + d^2\right)^5 \left(2.25 \cdot 10^{-8} R_1^2 \left(R_1^2 \sigma_{R1}^2 + d^2 \sigma_d^2\right) + 1.0 \cdot 10^{-8} \sigma_{R1}^2 \left(R_1^2 + d^2\right)^2\right) + R_2^2 \left(R_1^2 + d^2\right)^5 \left(2.25 \cdot 10^{-8} R_2^2 \left(R_2^2 \sigma_{R2}^2 + d^2 \sigma_d^2\right) + 1.0 \cdot 10^{-8} \sigma_{R2}^2 \left(R_2^2 + d^2\right)^2\right)}{\left(R_1^2 + d^2\right)^5 \left(R_2^2 + d^2\right)^5}}$$

$$= 104.0\pi \sqrt{\frac{29.5^2 \left(15.95^2 + 29.575^2\right)^5 \left(1.0 \cdot 10^{-8} \cdot 0.94999999999999992^2 \left(15.95^2 + 29.5^2\right)^2 + 2.25 \cdot 10^{-8} \cdot 29.5^2 \left(0.1^2 \cdot 15.95^2 + 0.94999999999999992^2 \cdot 29.5^2\right)\right) + 29.575^2 \left(15.95^2 + 29.5^2\right)^5 \left(1.0 \cdot 10^{-8} \cdot 0.32499999999999923^2 \left(15.95^2 + 29.575^2\right)^2 + 2.25 \cdot 10^{-8} \cdot 29.575^2 \left(0.1^2 \cdot 15.95^2 + 0.32499999999999923^2 \cdot 29.575^2\right)\right)}{\left(15.95^2 + 29.5^2\right)^5 \left(15.95^2 + 29.575^2\right)^5}}$$

In [4]:
```
# slope = 0.0874
# q = 1.6*10**(-19)
# m = 9.10938356 * 10**(-31)
# v_predicted = sqrt(2*q*400 / m)
# r = 3 / 100
# v_actual = q*D*r/m * 2.18
# v_predicted, v_actual
# (v_actual - v_predicted) / v_actual * 100
```