SUPPLEMENATY MATERIAL

Python Code used for the Monte Carlo simulations

```
import numpy
import scipy.stats
import matplotlib.pyplot as plt
plt.style.use("bmh")
import math
from math import pi
#
RPM=1200
kW=500
T=9549*kW/RPM
print(T)
outerdia=0.05557082
innerdia=0.044456656
C=outerdia/2
J=(pi / 32) * (outerdia**4 - innerdia**4)
print(T*C/(J*1e6))
#
def simulated_stress():
#	outerdia=scipy.stats.norm(0.05557082, 0.05557082*0.6*0.01).rvs()
#	innerdia=scipy.stats.norm(0.044456656, 0.044456656*0.6*0.01).rvs()
#	J=(pi / 32) * (outerdia**4 - innerdia**4)
#	C=outerdia/2
	Tvar=scipy.stats.norm(T, T*12*0.01).rvs()
	return Tvar*C/(J*1e6)
#
def simulated_yield_strength():
	return scipy.stats.norm(300, 300*12*0.01).rvs()
#
N = 100000
failures = 0
stresses = numpy.empty(N)
yield_strengths = numpy.empty(N)
for i in range(N):
	stresses[i] = simulated_stress()
	yield_strengths[i] = simulated_yield_strength()
	if stresses[i] > yield_strengths[i]:
		failures += 1
#
plt.hist(stresses, bins=100, alpha=0.5, label="Allowable Stress")
plt.hist(yield_strengths, bins = 100, alpha=0.5, label="Strength")
plt.xlabel("Stress (MPa)")
plt.ylabel("f(A), f(Y)")
```

```
plt.xlim([100, 400])
plt.ylim([0, 4000])
plt.legend(loc="upper right");
plt.show();
#
# the probability of failure and frugal reliability
print('FAILURE PROBABILITY',f"{failures / N:.6}")
#print(f"{failures / N:.6}")
#print(failures / N)
#print(1-failures / N)
#print(f"{1-(failures / N):.6}")
print('FRUGAL RELIABILITY',f"{1-(failures / N):.6}")
```