Untitled

March 31, 2025

[1]:
```
import numpy as np

from numpy.random import default_rng
rng = default_rng()
```

[2]:
```
def f(x):
    return x**2
```

[3]:
```
def get_random_points(a,b,N):
    return a+(b-a)*rng.random(N)
```

[20]:
```
def approx_method_1(f,a,b,mu,N):
    xs = get_random_points(a,b,N)
    ys = get_random_points(0,mu,N)

    M=len(list(filter(lambda z: z[1] <= f(z[0]),zip(xs,ys))))

    return (b-a)*mu*M/N
```

[24]:
```
approx_method_1(lambda x:x,0,1,1,1000)
```

[24]:
```
0.504
```

[25]:
```
def approx_method_2(f,a,b,N):
    xs = get_random_points(a,b,N)
    return (b-a)/N * sum(map(f,xs))
```

[28]:
```
approx_method_2(lambda x:x,0,1,1000)
```

[28]:
```
np.float64(0.5074600461174628)
```

[29]:
```
import pandas as pd
```

[33]:
```
trial = pd.DataFrame([ approx_method_1(f,0,1,1,1000) for t in range(20)])
trial
```

[33]:

	0
0	0.341

```
1   0.329
2   0.353
3   0.338
4   0.348
5   0.342
6   0.325
7   0.317
8   0.306
9   0.331
10  0.342
11  0.314
12  0.332
13  0.351
14  0.351
15  0.362
16  0.321
17  0.309
18  0.352
19  0.342
```

[35]:
```
[trial.mean(), trial.std() ]
```

[35]:
```
[0    0.3353
 dtype: float64,
 0    0.015981
 dtype: float64]
```

[36]:
```
print(trial.std())
```

```
0    0.015981
dtype: float64
```

[ ]:
```

```