Untitled

April 20, 2020

[65]:
```
!python3 --version
```

```
Python 3.6.9
```

[0]:
```
from numba import jit, prange
import numpy as np
import time
```

0.0.1 The function to use (I modified a bit the homework one)

[67]:
```
@jit(nopython=True, fastmath=True)
def f(x: float, y: float, z: float) -> float:
  return np.sin(np.sqrt(x * x + y * y)) + z

f(1, 2, 3)
```

[67]:
```
3.786749131547214
```

[68]:
```
from random import uniform
print(uniform(1.2, 1.25))
```

```
1.2253200088533978
```

0.0.2 The parell version

[0]:
```
@jit(nopython=True, parallel=True)
def integrate(fun, N: int, a: float, b: float, c: float, d: float, e: float, f:␣
 ↪float) -> float:
  total = 0.0
  for i in prange(N):
    rand_x = uniform(a, b)
    rand_y = uniform(c, d)
    rand_z = uniform(e, f)
    total += fun(rand_x, rand_y, rand_z)

  return (b - a) * (d - c) * (f - e) * (total / N)
```

0.0.3 Normal python version (+ fast fun)

[0]:
```
def integrate_slow(fun, N: int, a: float, b: float, c: float, d: float, e:␣
↪float, f: float) -> float:
 total = 0.0
 for i in range(N):
   rand_x = uniform(a, b)
   rand_y = uniform(c, d)
   rand_z = uniform(e, f)
   total += fun(rand_x, rand_y, rand_z)

 return (b - a) * (d - c) * (f - e) * (total / N)
```

0.1 The function to work

$$\int_0^1 \int_0^2 \int_{-1}^1 sin(sqrt(x * x + y * y)) + z \;\; dx\, dy\, dz$$

[71]:
```
# Example
integrate(f, 200000, 0, 1, 0, 2, -1, 1)
```

[71]:
```
3.251086176076951
```

[78]:
```
import time
start = time.time()
integrate(f, 2_000_000, 0, 1, 0, 2, -1, 1)
end = time.time()
print(f"Elapsed (parallel) = {end - start}")
```

```
Elapsed (parallel) = 0.08287668228149414
```

[79]:
```
import time
start = time.time()
integrate_slow(f, 2_000_000, 0, 1, 0, 2, -1, 1)
end = time.time()
print(f"Elapsed (normal) = {end - start}")
```

```
Elapsed (normal) = 1.9377877712249756
```