hw06

November 14, 2018

```
In [1]: import numpy as np
        from numba import jit

In [2]: to_group_e_min = 0.4
        to_group_e_max = 0.5

In [3]: from_group_e_min = 0.6
        from_group_e_max = 0.7

In [4]: alpha = ((12. - 1.) / (12. + 1.))**2

In [5]: from_es = np.linspace(from_group_e_min, from_group_e_max)
        to_es = np.linspace(to_group_e_min, to_group_e_max)

In [6]: @jit
        def numba_my_function(ef, et):
            scale = np.log(from_group_e_max) - np.log(from_group_e_min)
            def flux(e):
                return 1 / e / scale
            def sigma(e_from, e_to, sig=5):
                if e_to < alpha * e_from:
                    return 0
                elif e_to > e_from:
                    return 0
                else:
                    return sig
            return sigma(ef, et, 5) / ef * flux(ef) / (1 - alpha)

In [7]: contributions = []

        num_bins = 5000

        from_energies = np.linspace(from_group_e_min, from_group_e_max, num_bins)
        to_energies = np.linspace(to_group_e_min, to_group_e_max, num_bins)

        delta_e_from = from_energies[1] - from_energies[0]
        delta_e_to = to_energies[1] - to_energies[0]
```

```
ef = from_group_e_min

answer = 0

for ef in from_energies:
    for et in to_energies:
        c = 0.5 * (numba_my_function(ef, et) + numba_my_function(ef+delta_e_from, et+del
        c *= delta_e_from * delta_e_to
        answer += c

print(answer)
```

```
0.9919603072947006
```