Section3

Elliott Ashby

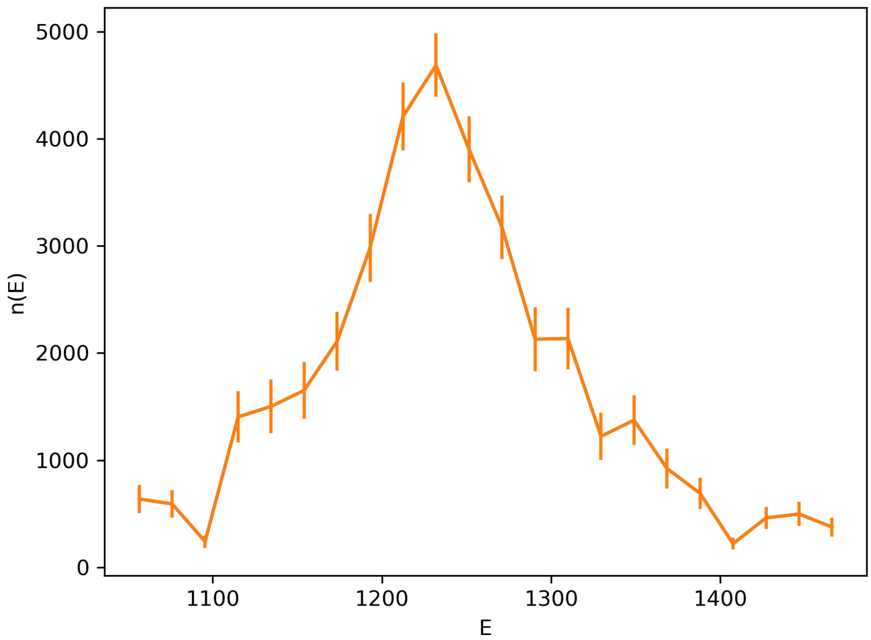
October 26, 2023

1 q1

```
from numpy import linspace
from numpy import arange
from scipy.optimize import curve_fit
from numpy import array
import matplotlib.pyplot as plt
from minimise import gmin
four = __import__('2_4')


def theory(e, w):
    th = [(1.4767e7 / (w ** 2 / 4 + (i - 1232) ** 2)) for i in e]
    return th


def parse_data(file: str):
    f = open(file, 'r')
    ea = []
    na = []
    dn = []
    for line in f:
        estr, nstr, errstr = line.split()
        ea.append(float(estr))
        na.append(float(nstr))
        dn.append(float(errstr))
    ea = array(ea)
    na = array(na)
    dn = array(dn)
    f.close()
    return [ea, na, dn]


def one():
    ea, na, dn = parse_data('./data1.txt')

    def discrep(w):
        r = [(na - theory(ea, w))]
```

```
        return four.discrepancy(r, dn)

    print(gmin(discrep, 80, 120, tol=3.0e-8))
    w = 111.8916236

    # plt.figure(figsize=(3, 3))
    plt.plot(ea, na)
    plt.errorbar(ea, na, dn)
    plt.xlabel('E')
    plt.ylabel('n(E)')
    plt.savefig('2_1.png', dpi=300)

    plt.plot(ea, theory(ea, w))
    plt.savefig('2_3th.png', dpi=300)
    plt.clf()


def minimum_discrepancy(ea, na, dn):
    r = []
    minimum = []
    ranger = arange(80, 120, 0.01)

    for i in ranger:
        w = i
        r = [(na - theory(ea, w))]
        minimum.append(four.discrepancy(r, dn))

    mindiscrep = min(minimum)
    w = ranger[minimum.index(min(minimum))]
    print('minimum discrepancy =', mindiscrep)
    print('which is when w =', w)
    return (mindiscrep, w)


def one_theory():
    ea, na, dn = parse_data('./data1.txt')

    def discrep(w):
        r = [(na - theory(ea, w))]
        return four.discrepancy(r, dn)

    plt.plot(ea, na)
    plt.errorbar(ea, na, dn)
    plt.xlabel('E')
    plt.ylabel('n(E)')
    plt.plot(ea, theory(ea, gmin(discrep, 80, 120, tol=3.0e-8)[0]))
    plt.savefig('2_5.png', dpi=300)

    x = linspace(start=ea[0], stop=ea[-1], num=len(theory(ea, gmin(discrep, 80,
    popt, pcov = curve_fit(theory, x, theory(ea, gmin(discrep, 80, 120, tol=3.0
```

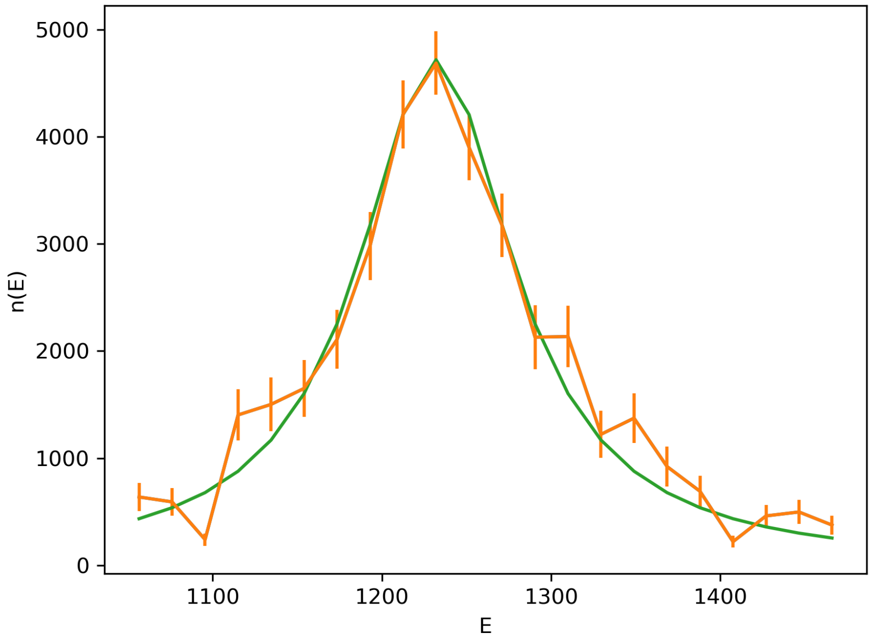
```
    print(popt)
    print(pcov)
    plt.clf()


if __name__ == "__main__":
    one()
    one_theory()
```



2 q2

As E increases from 1000 n(E) increases at an increasing rate until around 1200 where the gradient change decreases to a peak at around 1250. This shape is mirrored on the other side ending around 1500.

3 q3



$$w = \frac{600\sqrt{163}}{\sqrt{4687}} \quad (1)$$

Changing w effects the maximum value of the peak of the theoretical curve without changing the minimum values at roughly $n(E) = 500$. Increasing w decreases the maximum (since w^2 is divided by in $n(E)$). Whereas decreasing w increases the maximum of $n(E)$.

4 q4

```
from numpy import sum as sm


def discrepancy(r, dn):
    '''where r is the residuals,
    and dn is the error on the residuals'''
    return sm([(r[i] / dn[i]) ** 2 for i in range(len(r))])
```

This function takes in the values of the residuals and their errors and returns a sum over each residual divided by their error squared.

5 q5

Manually using trial and error to reduce the discrepancy results in a minimum discrepancy of 86.76 which is when $w = 113.15$. The method used is just iterating on the value of w and returning the value of the discrepancy each time.

6 q6

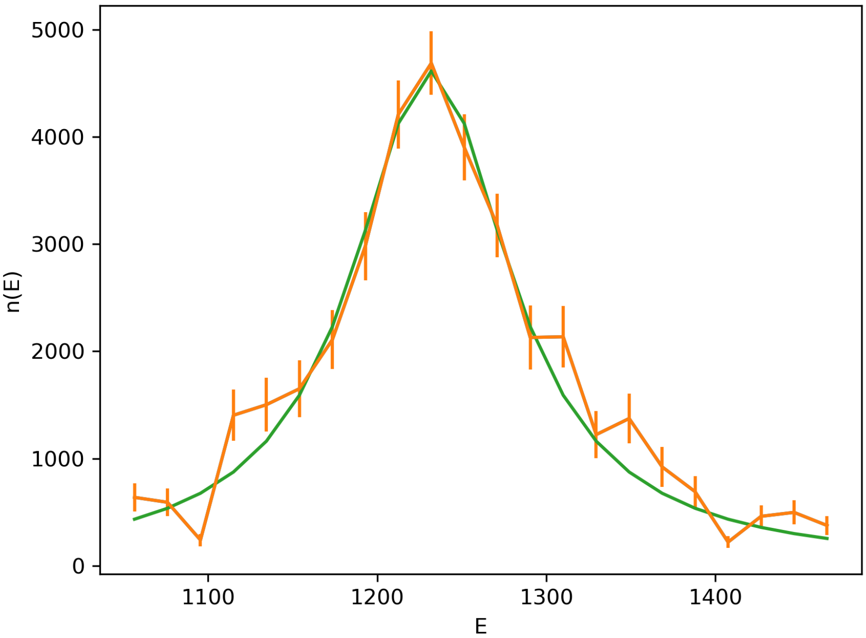
Using the gmin function like:

```
from minimise import gmin
from numpy import exp


def f(x):
    return exp(x) + 1 / x


print(gmin(f, 0, 5, tol=3.0e-8))
```

This returns the only minimum in the curve since in the negatives, f(x) is decreasing to a asymptote.
This returns:
$\rightarrow$ (0.7034674247387416, 3.4422772944949744)
which is the (x, y) coords of the minimum.

7 q7

Using gmin() on the discrepancy function reveals a minimum w of 113.14847521305097 where the discrepancy is 86.76529920076851. using these values in a plot yields:



8 q8

```
x = linspace(start=ea[0], stop=ea[-1], \
        num=len(theory(ea, gmin(discrep, \
        80, 120, tol=3.0e-8)[0])))
popt, pcov = curve_fit(theory, x, \
        theory(ea, gmin(discrep, \
        80, 120, tol=3.0e-8)[0]), \
        sigma=dn, absolute_sigma=True)
print(popt)
print(pcov)
```

This outputs a value of w of 113.14847521 with an uncertainty of 4.84317108