Calculate the Fourier coefficient v_2 from two-particle correlations

In [3]:

```
import numpy as np
from itertools import combinations
import math as m
```

Read data

In [4]:

```
event, phi = np.loadtxt("dndphi_events.csv", delimiter=',', skiprows=1,
unpack=True)
```

Define function that calculates v_2 for a given event

One can use `itertools.combinations` (https://docs.python.org/3/library/itertools.html#itertools.combinations) to get all pairs for a given 1d array.

The function we use here is

$$c_n\{2\} \equiv \left\langle\left\langle e^{in(\varphi_1-\varphi_2)}\right\rangle\right\rangle = \left\langle v_n^2\right\rangle$$

whereas the ´v2´ will average over all pairs from φs and the main function will average over all evets Note that only the real part is kept, which $\exp(i2\varphi)$ is now $\cos(2\varphi)$

In [5]:

```
def v2(phi_vals):
    temp=combinations(phi_vals,2)
    all=0
    num=0
    for i in temp:
        all+=m.cos(2*(i[0]-i[1]))
        num+=1
    return all/num
```

Loop over all events and determine v_2 averaged over all events

In [20]:

```
v2val = np.array([]) # array with v2 values for each event, can use numpy.append() to append a value

nevt = 100
for i in range(nevt):
    phi_vals = phi[event == i]
    v2val=np.append(v2val,v2(phi_vals))
print('v2:',m.sqrt(sum(v2val)/nevt))
```

```
v2: 0.1462940675740694
```

In [0]:

```
```