1 Stationary limit of the underdamped Fokker-Planck equation

The Fokker-Planck equation for $p(x,p,t)$ reads

$$\dot{p}(x,p,t) = \left(-\frac{p}{m}\partial_x + kx\partial_p + \frac{\xi}{m} + \frac{\xi}{m}p\partial_p + D_p\partial_p^2\right) p(x,p,t)\,. \tag{1}$$

Note that here D_p is a diffusion constant for the momentum p, not for the position x.

1. The stationary solution $p_s(x,p)$ is a product of two Gaussians in x and p. Write the ansatz

$$p_s(x,p) \propto e^{-\lambda_1 x^2} e^{-\lambda_2 p^2}. \tag{2}$$

Insert in the above equation and obtain

$$0 = \left(\frac{p}{m}2\lambda_1 x - kx2p\lambda_2 + \frac{\xi}{m} - \frac{\xi}{m}2p^2\lambda_2 + D_p(-2\lambda_2 + 4p^2\lambda_2^2)\right) e^{-\lambda_1 x^2} e^{-\lambda_2 p^2}\,, \tag{3}$$

$$0 = \left(\frac{p}{m}\lambda_1 x - kxp\lambda_2 + \frac{\xi}{2m} - \frac{\xi}{m}p^2\lambda_2 + D_p(-\lambda_2 + 2p^2\lambda_2^2)\right)\,. \tag{4}$$

We notice that the constant term has to vanish independentely of the others: $\frac{\xi}{2m} - D_p\lambda_2 = 0$. From this, we can deduce

$$\lambda_2 = \frac{\xi}{2mD_p}\,. \tag{5}$$

This is consistent with the condition that the terms proportional to p^2 have to vanish too. It follows that λ_1 is given by

$$\lambda_1 = mk\lambda_2 = \frac{k\xi}{2D_p}\,. \tag{6}$$

And thus

$$p_s(x,p) \propto e^{-\frac{k\xi x^2}{2D_p}} e^{-\frac{\xi p^2}{2mD_p}}. \tag{7}$$

2. Comparing the Boltzmann distribution $p_s(x,p) \propto e^{-(T(p)+U(x))/(k_BT)}$, where $T(p) = p^2/(2m)$ is the kinetic energy term, we can identify the following relations

$$D_p = \xi k_B T\,, \tag{8}$$

$$\frac{U(x)}{k_BT} = \frac{k\xi x^2}{2D_p}\,. \tag{9}$$

3. Calculate the first and second moments of the stationary distribution. Explain the physical meaning of your results.

2 Ideal Gas - Momentum in the microcanonical ensemble

Add this after tutorial!

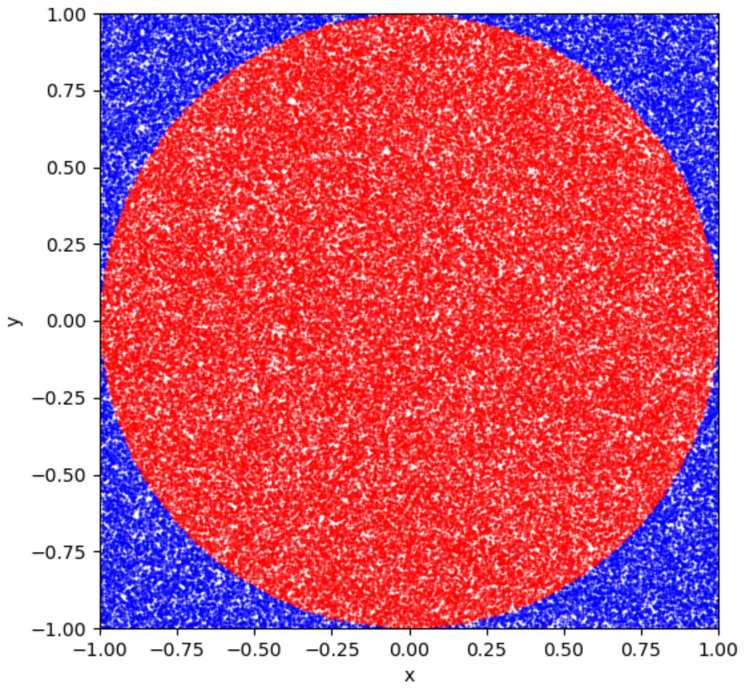
3 Computer exercise - Monte Carlo integration

The algorithm can be implemented using the following python code snippet:

```python
import numpy as np

A = 0

N = 100000
for _ in range(N):
    x = np.random.uniform(-1, 1)
    y = np.random.uniform(-1, 1)
    if x**2 + y**2 <= 1:
        A += 1

z = 4 * A / N
print(z)
```

With this, a value for π can be estimated, running the above code returns 3.14312. The ratio of A to N is directly proportional to another ratio, namely that of a the area of a unit circle to the area of a square with side length 2. One could thus in principle calculate π by e.g. throwing darts onto a square surface with a circle drawn on it. This is because the number of "hits" can be assumed to be proportional to the hit probability, which again is directly proportional to the respective area. This method can only work if the distribution of random numbers is uniform and N is large enough.

Visualization:



1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
y
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
x