PHYS 222 - Homework 3: Monte Carlo

Fahed Abu Shaer

December 2023

1 Density Matrix for the Quantum Harmonic Oscillator

The one-dimensional harmonic oscillator is given by the following potential

$$V(x) = \frac{1}{2}kx^2$$

The Hamiltonian of the system is the Hamiltonian of a free particle plus the potential,

$$H = H_{free} + V(x)$$
$$H = \frac{p^2}{2m} + \frac{1}{2}kx^2$$

The general form of the density is given by

$$\rho_n(x, x', \beta) = \sum e^{-\beta E_n}\psi^*(x)\psi(x')$$

Similar to the Hamiltonian, the density can be split to be the product of the exponential of energies, Trotter formula,

$$\rho(x, x', \beta) \approx e^{-\frac{\beta V(x)}{2}} e^{-\beta H_f} e^{-\frac{\beta V(x')}{2}}$$

Another property for the density matrix is convolution where

$$\rho(x, x'', \beta_1 + \beta_2) = \int \rho(x, x', \beta_1)\rho(x', x'', \beta_2)dx'$$

When $\beta_1 = \beta_2$, we can start with a high temperature and convolve it with itself n times to get the behaviour at low temperature. For the harmonic oscillator we have the following densities,

$$\rho_{free} = \frac{1}{\sqrt{2\pi\beta}} e^{-\frac{(x-x')^2}{2\beta}} \text{ taking } \frac{h}{c} = 1$$
$$\rho_{harmonic} = e^{-\frac{\beta k x}{2}}$$

We applied the convolution and trotter property to find the density matrix for the harmonic oscillator. First, we defined the densities,

```
def density(x1,x2,beta):
    return 1/(np.sqrt(2*np.pi*beta)) * np.exp(-((x1-x2.reshape(-1,1))**2)/(2*beta))

def harmonic(x,beta,k):
    return np.exp(-(1/2)*beta*k*x) * density(x,x,beta) * np.exp(-(1/2)*beta*k*x)
```

Then, we applied it to an array of points,

```
x = np.linspace(0,1,1000)
beta = 0.01
k = .005
dx = x[1]-x[0]
h = harmonic(x,beta,k)
```

We start squaring it and summing it, discrete integral, and producing different diagrams for different powers.

```
for i in range(5):
    plt.imshow(h)
    plt.show()
    h = dx * (h@h)
plt.imshow(h)
plt.show()
```

These are the results we got,

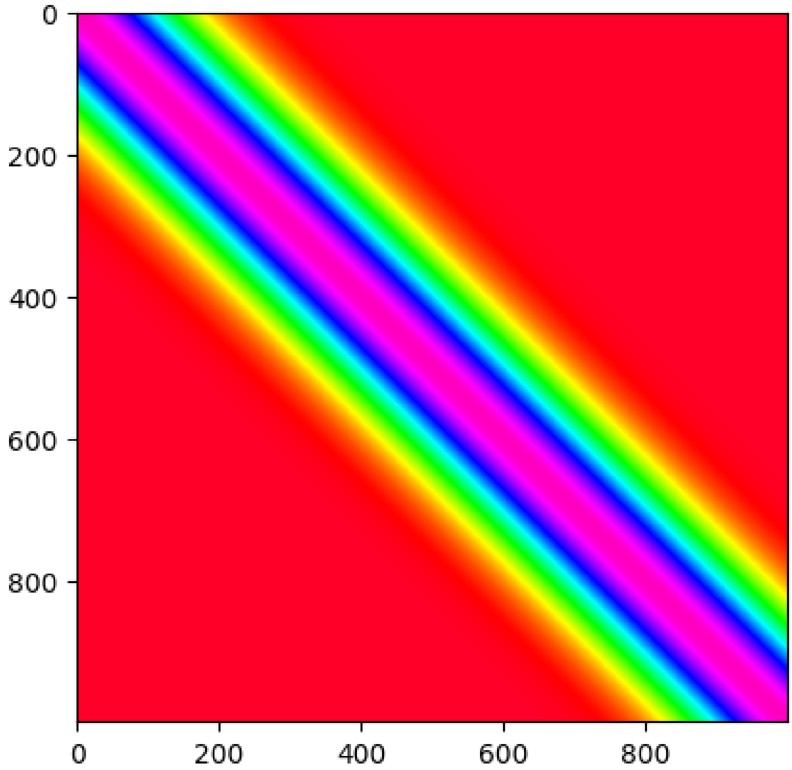

Figure 1: Density Matrix at n = 1

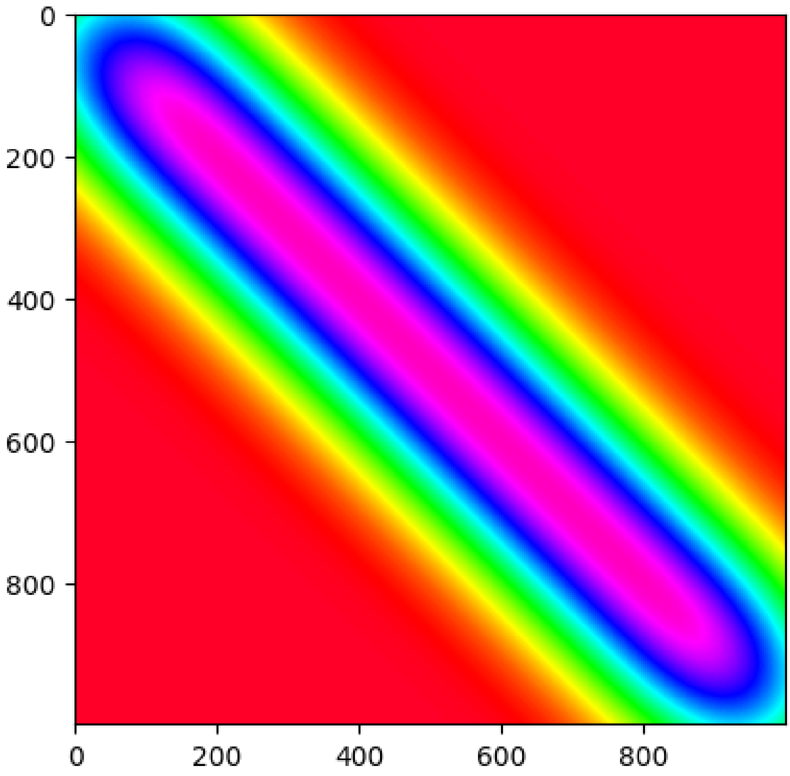

Figure 2: Density Matrix at n = 2

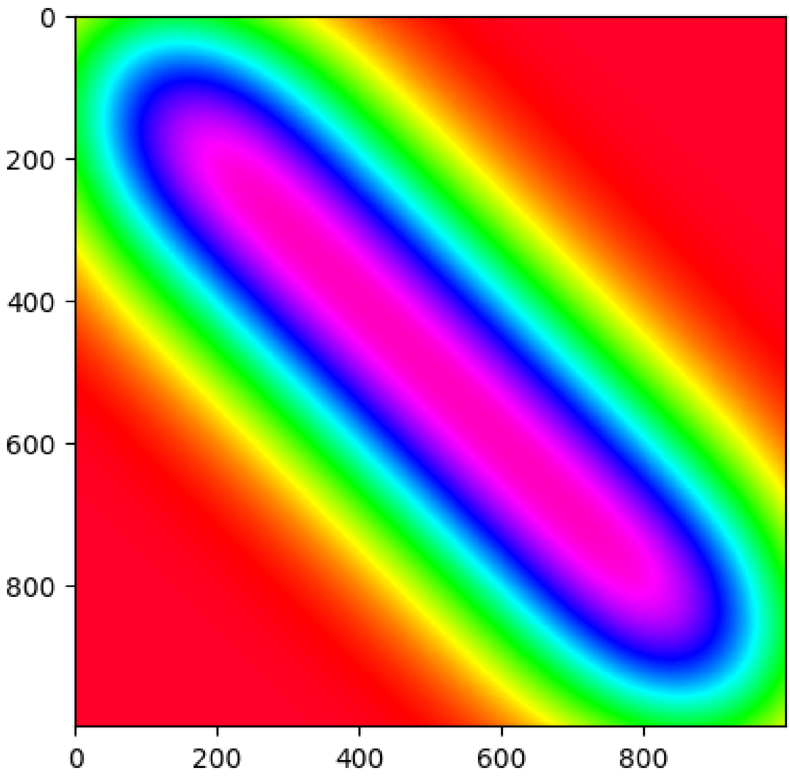

Figure 3: Density Matrix at n = 4

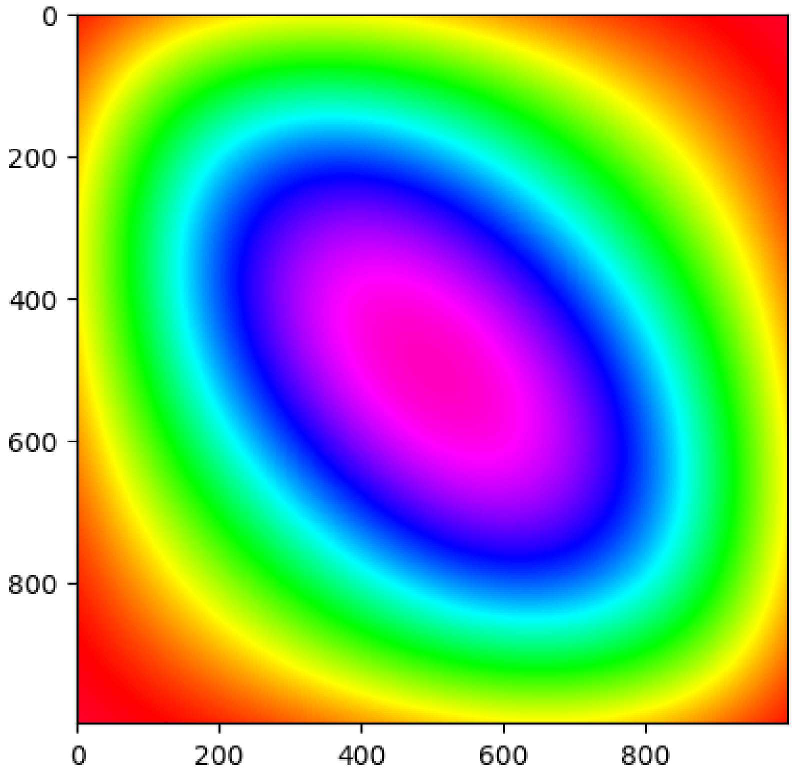

Figure 4: Density Matrix at n = 8

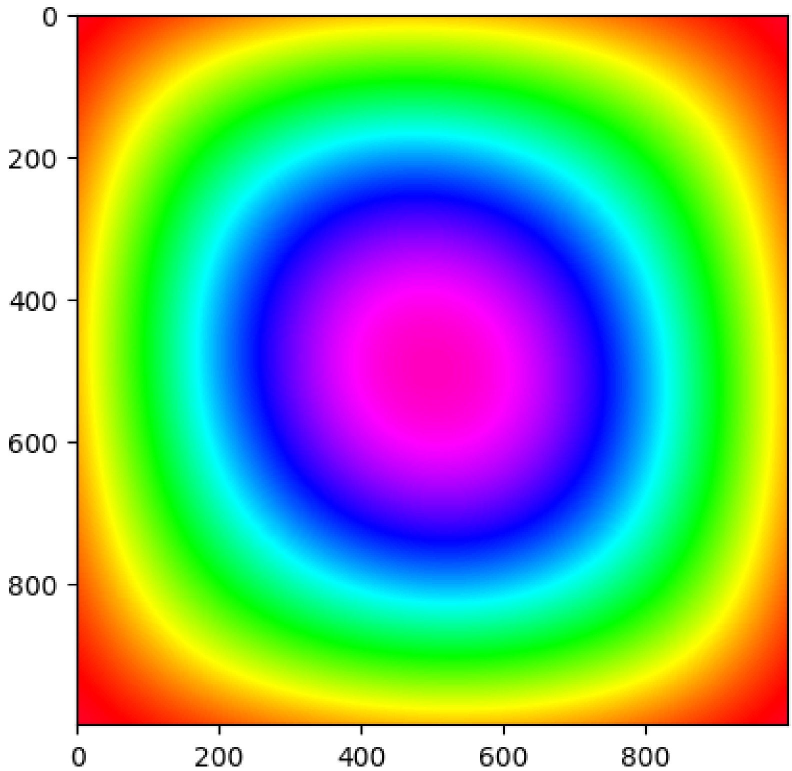

Figure 5: Density Matrix at n = 10

We can see as we covolve the density matrix with itself, we go from a high temperature regime, when the density is concentrated on the diagonal, to a low temperature regime where the density is more disperse.

2 Solution of the Schrödinger Equation using Random Walk

We tried to solve the Schrödinger equation using the density matrix and changing it randomly to settle to a steady solution. First, we pick a random index which defines the position I'm looking at. Then, calculate the new probabilities and define the new acceptance probabilities. After that, we generate a random number between zero and one, and if it is less than $\frac{\pi_b}{\pi_a}$, then we accept it. We do this iteratively until we think we reached a steady solution. Then, we plot a histogram of the probability of x, and this histogram is the modulus of the wave function.

```
def rho(x0,x1,beta):
    return 1/(np.sqrt(2*np.pi*beta)) * np.exp(-((x0-x1) ** 2)/(2*beta))

def randomeWalk(x,beta,d=0.1):
    N = len(x)
    dt = beta/N
    k = np.random.randint(0, N-2)
    k1 = k+1
    k2 = k-1
    if k2 == -1:
        k2 = N-1
    xk1 = x[k] + rand.uniform(-d, d)
    pia = rho(x[k2], x[k], dt) * rho(x[k], x[k1], dt) * np.exp(-0.5 * dt * x[k] **2)
    pib = rho(x[k2], xk1, dt) * rho(xk1, x[k1], dt) * np.exp(-0.5 * dt * xk1 **2)
    if rand.uniform(0,1) < pib/pia:
        x[k] = xk1
    return x

x = np.linspace(0,1,1000)
beta = 4
d = 0.1
for i in tqdm(range(1000000)):
    x = randomeWalk(x,beta,d)

plt.hist(x,bins=80, density=True)
plt.show()
```

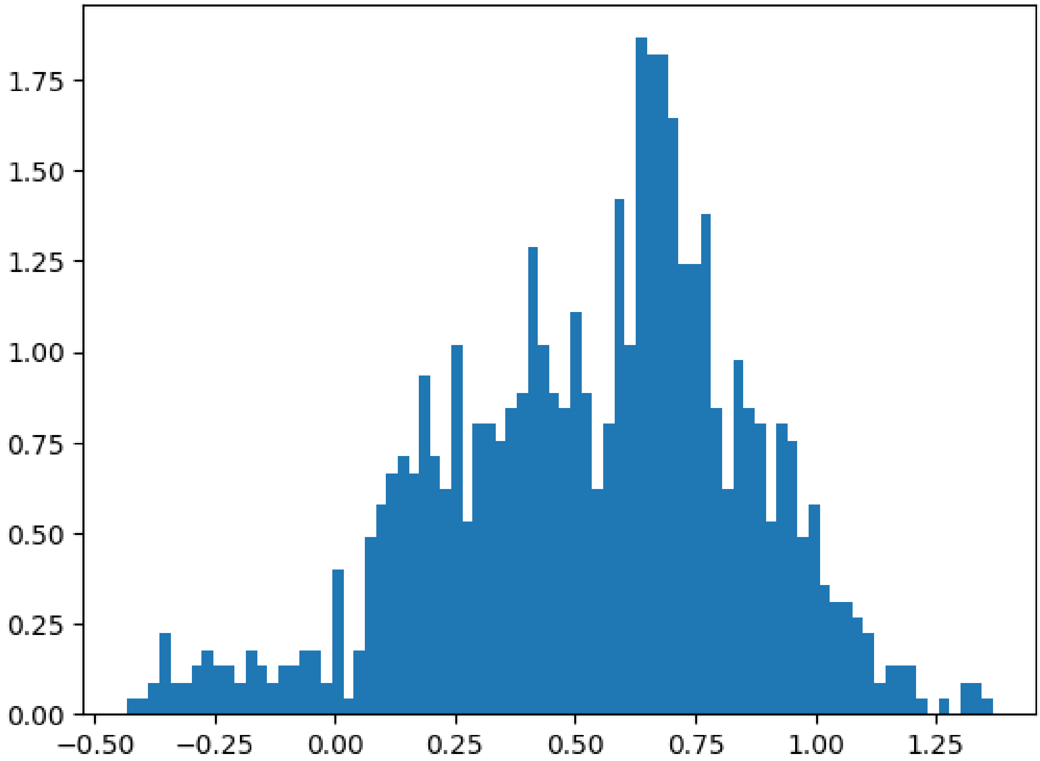

Figure 6: Histogram of x

As expected, the histogram looks like the modulus of the wave function.