Computational Physics - Exam: Quantum Harmonic Oscillator

Torben Steegmann

Introduction

The harmonic oscillator is a key system in both classical and quantum mechanics, used to describe the motion of a wide range of phenomena. In classical mechanics, it models scenarios such as the vibration of strings and the behavior of electronic circuits. In quantum mechanics, it extends to describing, for example, the movement of atoms in solids or the behavior of light.

Despite the quantum harmonic oscillator being one of the few quantum mechanical problems with an exact analytical solution, numerical methods are crucial for understanding and simulating more complex systems where analytical solutions are not feasible. In this exercise, we focus on the numerical solution of the quantum harmonic oscillator using the product formula approach. We will write a program to simulate the oscillation and calculate the variance for a set of given initial conditions.

Additionally, we will derive the analytical solution for the variance and compare it to our numerical results to validate the accuracy of our simulations. This approach not only reinforces our understanding of the quantum harmonic oscillator but also demonstrates the importance and application of numerical methods in solving quantum mechanical problems.

Simulation Model and Method

Background

While classical physics deals with the macroscopic world and can describe it deterministically, it cannot be used to describe microscopic phenomena like the movement of single atoms. For these microscopic phenomena, quantum mechanics is required, as it reveals that particles have both wave-like and particle-like properties. Quantum mechanics describes particles as a superposition of states in a wave function until it is measured, at which point the wave function collapses and the particle is forced into a specific state.

For a 1D harmonic oscillator in quantum space, this means that its position at any point in time is not described by a deterministic value, but by a probability

density, which can be derived from the Time-Dependent Schrödinger Equation (TDSE). Throughout this chapter, we will keep all variables dimensionless, thus we set $\hbar = m = 1$. With this, the TDSE for a 1D harmonic oscillator is

$$i\frac{\partial}{\partial t}\Phi(x,t) = \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{\Omega^2}{2}x^2\right)\Phi(x,t). \tag{1}$$

Physically, we can interpret the probability of finding the particle at an infinitesimal interval x and a time t from the wave function $\Phi(x,t)$ with $\Phi^*\Phi dx$. From this, we can derive the probability density as

$$\Phi(x,t) = \Phi^*(x,t)\Phi(x,t) = |\Phi(x,t)|^2. \tag{2}$$

With this, the expectation values are given as

$$\begin{aligned} \langle x(t)\rangle &= \int_{-\infty}^{+\infty} x\ |\Phi(x,t)|^2\ dx \\ \langle x^2(t)\rangle &= \int_{-\infty}^{+\infty} x^2\ |\Phi(x,t)|^2\ dx, \end{aligned} \tag{3}$$

and the variance as

$$V(x,t) = \langle x^2(t)\rangle - \langle x(t)\rangle^2. \tag{4}$$

In the following, we are going to derive a numerical solution for the wave function using the product formula approach. To compare our results, we will also derive the analytical solution of the expectation values and the variance.

Analytical Solution

The motion of a classical 1D harmonic oscillator is described by $x(t) = x_0\cos(\omega t)$. Because the initial wave packet

$$\Phi(x, t=0) = \left(\pi\sigma^2\right)^{-\frac{1}{4}} e^{-\frac{(x-x_0)^2}{2\sigma^2}} \tag{5}$$

is Gaussian and centered in x_0, it is easy to see that we can replace ω with its quantum equivalent Ω and derive

$$\langle x(t)\rangle = x_0\cos(\Omega t), \tag{6}$$

for dimensionless variables. The derivation for $\langle x^2\rangle$ is not as straight forward though. Through a combination of internet research and trial-and-error, we were able to come up with the following **suspected** solution:

$$\langle x^2(t)\rangle = x_0^2\cos^2(\Omega t) + \frac{\sigma^2}{2}\cos^2(\Omega t) + \frac{\sin^2(\Omega t)}{2\sigma^2\Omega^2}, \tag{7}$$

leading to

$$\langle x^2(t)\rangle - \langle x(t)\rangle^2 = \frac{\sigma^2}{2}\cos^2(\Omega t) + \frac{\sin^2(\Omega t)}{2\sigma^2\Omega^2}. \tag{8}$$

We were unfortunately not able to derive this analytically, but as this gives the desired results for all test cases, we will use this to compare the numerical results to.

Numerical Solution

The formal solution of the wave function can be written as $\Phi(x,t) = e^{-itH}\Phi(x,t=0)$, where H is the Hamiltonian. We can approximate $\Phi(x,t)$ using the product formula approach. For this, we first discretize the wave function with a spatial resolution of Δ and a grid size L, and express the spatial derivative with the finite difference approximation

$$\frac{\partial^2\Phi(x,t)}{\partial x^2} \approx \frac{\Phi(x+\Delta,t) - 2\Phi(x,t) + \Phi(x-\Delta,t)}{\Delta^2}. \tag{9}$$

The wave function itself is discretized as

$$\Phi(t) = \begin{pmatrix} \Phi_1(t) \\ \Phi_2(t) \\ \vdots \\ \Phi_{L(t)} \end{pmatrix}, \tag{10}$$

where $\Phi_i(t)$ is the wave function at position i.

We can then express the Schrödinger equation in matrix form:

$$i\frac{\partial}{\partial t}\begin{pmatrix} \Phi_1(t) \\ \Phi_2(t) \\ \Phi_3(t) \\ \vdots \\ \Phi_L(t) \end{pmatrix} = \underbrace{\Delta^{-2}\begin{pmatrix} 1+\Delta^2V_1 & -1/2 & 0 & & & 0 \\ -1/2 & 1+\Delta^2V_2 & -1/2 & & & \\ 0 & -1/2 & 1+\Delta^2V_3 & & & \\ & & & \ddots & & 0 \\ & & & & 1+\Delta^2V_{L-1} & -1/2 \\ 0 & & & 0 & -1/2 & 1+\Delta^2V_L \end{pmatrix}}_{H}\begin{pmatrix} \Phi_1(t) \\ \Phi_2(t) \\ \Phi_3(t) \\ \vdots \\ \vdots \\ \Phi_L(t) \end{pmatrix}$$

and decompose H into three block-diagonal matrices:

$$H = V + K_1 + K_2 = \Delta^{-2}\begin{pmatrix} 1+\Delta^2 V_1 & 0 & 0 & & & 0 \\ 0 & 1+\Delta^2 V_2 & 0 & & & \\ 0 & 0 & 1+\Delta^2 V_3 & & & \\ & & & \ddots & & 0 \\ & & & & 1+\Delta^2 V_{L-1} & 0 \\ 0 & & & 0 & 0 & 1+\Delta^2 V_L \end{pmatrix}$$

$$+\Delta^{-2}\begin{pmatrix} 0 & -1/2 & 0 & & & 0 \\ -1/2 & 0 & 0 & & & \\ 0 & 0 & 0 & & & \\ & & & \ddots & -1/2 & 0 \\ & & & -1/2 & 0 & 0 \\ 0 & & & 0 & 0 & 0 \end{pmatrix} + \Delta^{-2}\begin{pmatrix} 0 & 0 & 0 & & & 0 \\ 0 & 0 & -1/2 & & & \\ 0 & -1/2 & 0 & & & \\ & & & \ddots & & 0 \\ & & & & 0 & -1/2 \\ 0 & & & 0 & -1/2 & 0 \end{pmatrix}$$

Remember that the formal solution of the Schrödinger Equation is $\Phi(x,t) = e^{-itH}\Phi(x,t=0)$. With this, we can approximate $e^{-i\tau H}$ as

$$e^{-i\tau H} \approx e^{-i\tau\frac{K_1}{2}} e^{-i\tau\frac{K_2}{2}} e^{-i\tau V} e^{-i\tau\frac{K_2}{2}} e^{-i\tau\frac{K_1}{2}}. \tag{11}$$

Since V is a diagonal matrix, $e^{-i\tau V}$ is a diagonal matrix where all diagonal elements V_i are of the form $e^{-i\tau V_i}$. Remember that V(x) is given by Equation 1 as $V(x) = \frac{\Omega^2}{2}x^2$. It can be shown that the exponentials of the block diagonal matrices K_1 and K_2 are

$$e^{-i\tau K_1/2} = \begin{pmatrix} c & is & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ is & c & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & c & is & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & is & c & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & c & is & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & is & c & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & \ddots & is & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & is & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{pmatrix} \quad e^{-i\tau K_2/2} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & c & is & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & is & c & 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & c & is & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & is & c & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 & c & is & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & is & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \ddots & is \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & is & c \end{pmatrix}$$

Here, $c = \cos\left(\frac{\tau}{4\Delta^2}\right)$ and $is = i\sin\left(\frac{\tau}{4\Delta^2}\right)$. For each time-step t, we can now obtain the result of $\Phi(t+\tau)$ as

$$\Phi(t+\tau) = e^{-i\tau\frac{K_1}{2}} e^{-i\tau\frac{K_2}{2}} e^{-i\tau V} e^{-i\tau\frac{K_2}{2}} e^{-i\tau\frac{K_1}{2}} \Phi(t). \tag{12}$$

As we have discretized $\Phi(x,t)$, the expectation values are also discretized as

$$\begin{aligned}\langle x(t)\rangle &= \sum_{i=0}^{L}\left(i - \left\lfloor\frac{L}{2}\right\rfloor\right)\ |\Phi(x,t)|^2 \\ \langle x^2(t)\rangle &= \sum_{i=0}^{L}\left(i - \left\lfloor\frac{L}{2}\right\rfloor\right)\ |\Phi(x,t)|^2\ .\end{aligned} \tag{13}$$

Implementation

The implementation follows the described model closely and can be found in the Appendix. We define a spatial grid from $-15 \leq x \leq 15$ by choosing a spatial resolution of $\Delta = 0.025$ and a grid size of $L = 1201$. We simulate the wave function for times $t = 0$ to $t = 10$ by selecting a time resolution of $\tau = 0.00025$ and a time-step amount of $m = 40000$. The starting configuration is controlled by adjusting the corresponding variables for Ω, σ, and x_0.

The main() function calls the driving function TSDE(), which begins by calling the function Phi_0() to calculate the initial wave packet. This was given as

$$\Phi(x, t=0) = \left(\pi\sigma^2\right)^{-\frac{1}{4}} e^{-\frac{(x-x_0)^2}{2\sigma^2}} \tag{14}$$

We store the result in a vector phi, after which we calculate the variance for $t = 0$. Note that we also initialize a matrix phi_t that will store phi at predefined points in time. We pre-compute values that stay constant throughout all time-steps, notably $V(x)$, c, and is. After this, the main for-loop begins, where the product formula is applied to phi, and the variance of phi is calculated for m time-steps numerically and analytically. As a last step, we calculate the probability distribution of phi for the times $t = 0$, $t = 2$, $t = 4$, $t = 6$, $t = 8$, $t = 10$, and plot the results.

Simulation Results

Running the described program yields the results depicted in the graphs in Figure 1 to Figure 12. For each of the given initial conditions, we calculate the probability distributions of the position of the wave, its expectation value $\langle x\rangle$ and the variance $\langle x^2\rangle - \langle x\rangle^2$ as described earlier.

The first set of initial conditions, $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 0$ result in the probability distribution of Figure 1 and the variance depicted in Figure 3. In classical mechanics, the particle would have no chance of being anywhere other than the center $x_0 = 0$. However, in quantum mechanics, there is a non-zero probability that the particle can be found not at the center, but besides it. Zoomed out, it looks like this probability does not change over time; however, when we

zoom in on the center, like in Figure 2, we can see that the peak probability is slightly oscillating. This is a result of the numerical approximation of the product formula approach. As we will see in the variance, increasing the temporal resolution decreases this inaccuracy. Figure 3 depicts the variance of the system. We can see that the expectation value $\langle x \rangle$ is 0. This is because the initial wave starts in the center in its equilibrium and is thus not expected to oscillate. For the same reason, the variance stays almost constant at $\langle x^2 \rangle - \langle x \rangle^2 = 0.5$. Again, the small oscillations are because of numerical inaccuracies caused by the chosen temporal resolution. Increasing the temporal resolution by a factor of 10 results in Figure 4, where we can observe a result that is even closer to the analytical result of 0.5 (Figure 13). Because these results are only slightly more accurate but come at the cost of a lot of computational speed, we choose to remain at the temporal resolution discussed earlier.

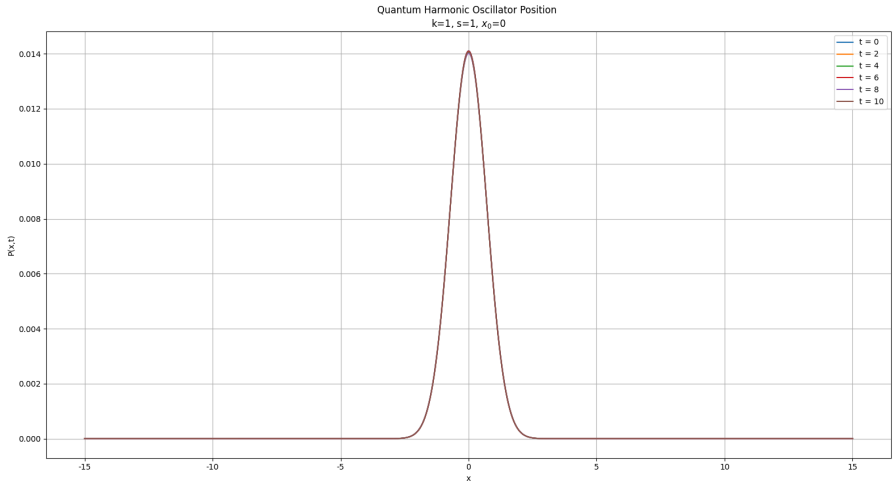

Figure 1: This graph illustrates the probability distribution according to the numerical solution of the TDSE for a 1D-harmonic oscillator. The x-axis is the position of the particle. The y-axis is the probability density of finding the particle at position x at time t. Different snapshots were taken at different times t, ranging from $t = 0$ (blue) to $t = 10$ (brown). The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1, x_0 = 0$.

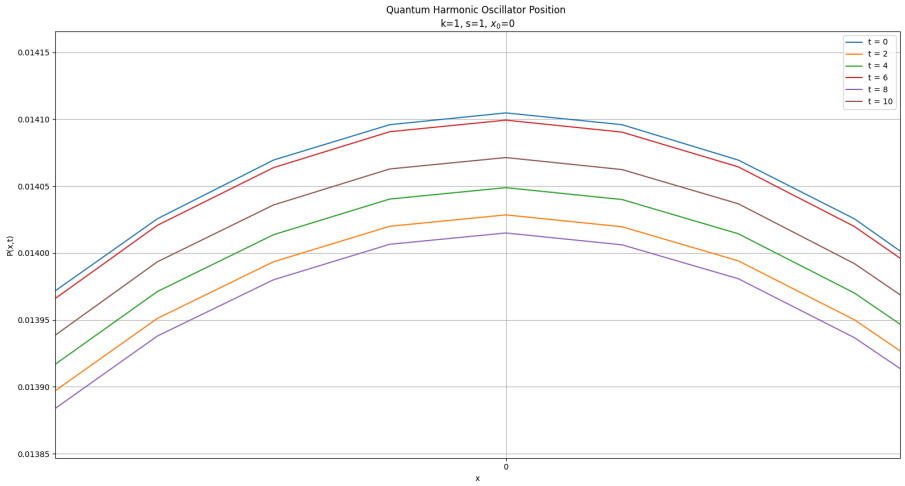

Figure 2: A close-up view of the maximum of the probability density distribution of Figure 1, where the initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 0$. We see small numerical inaccuracies.

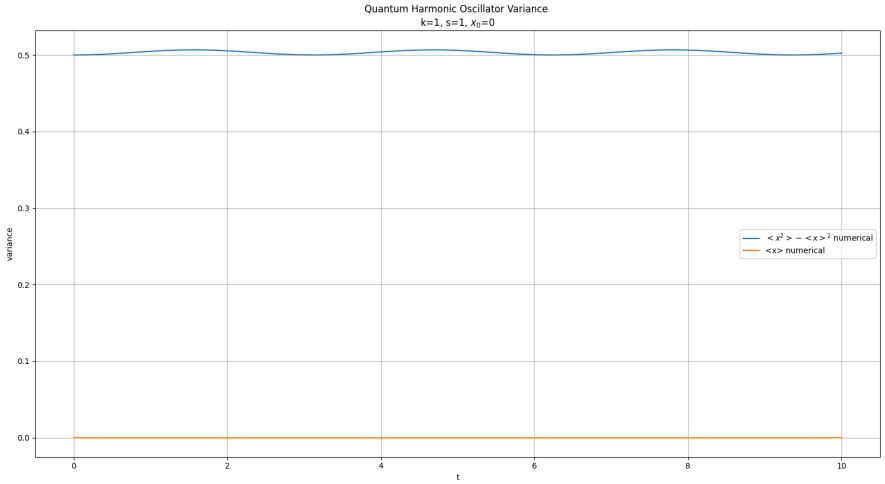

Figure 3: The expectation value $\langle x \rangle$ (orange) and variance $\langle x^2 \rangle - \langle x \rangle^2$ (blue) of x for Figure 1. The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 0$. The temporal resolution is $\tau = 0.00025$.

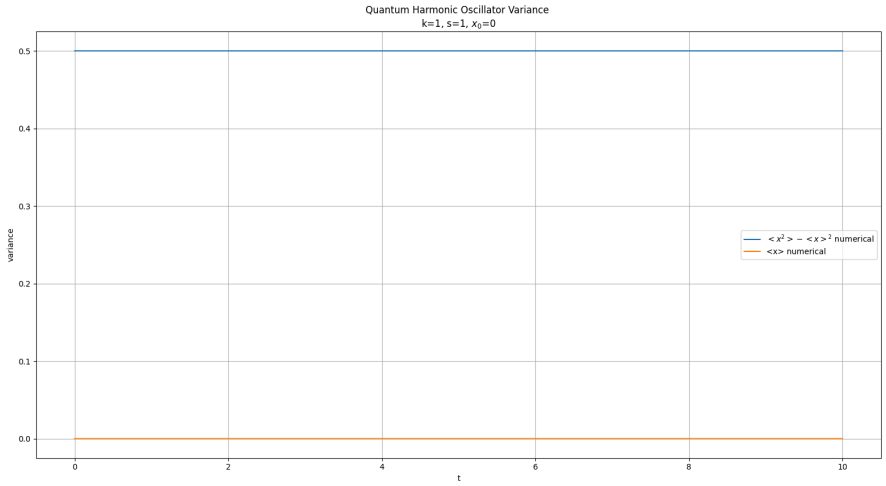

Figure 4: The expectation value $\langle x \rangle$ (orange) and variance $\langle x^2 \rangle - \langle x \rangle^2$ (blue) of x for Figure 1. The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 0$. The temporal resolution is $\tau = 0.000025$, meaning 10 times higher than in Figure 3. This results in a slightly more accurate plot.

The next set of initial conditions is $\Omega = 1$, $\sigma = 1$, $x_0 = 1$, meaning instead of initializing the center of the wave at equilibrium, we introduce an offset, resulting in the positional oscillations around the center $x = 0$, observed in Figure 5. As the angular velocity is still set to $\Omega = 1$ and the width of the wave is also still $\sigma = 1$, the variance still remains constant, up to numerical errors, at 0.5. As the wave is oscillating between 1 and -1 the expectation value $\langle x \rangle$ is oscillating with it.

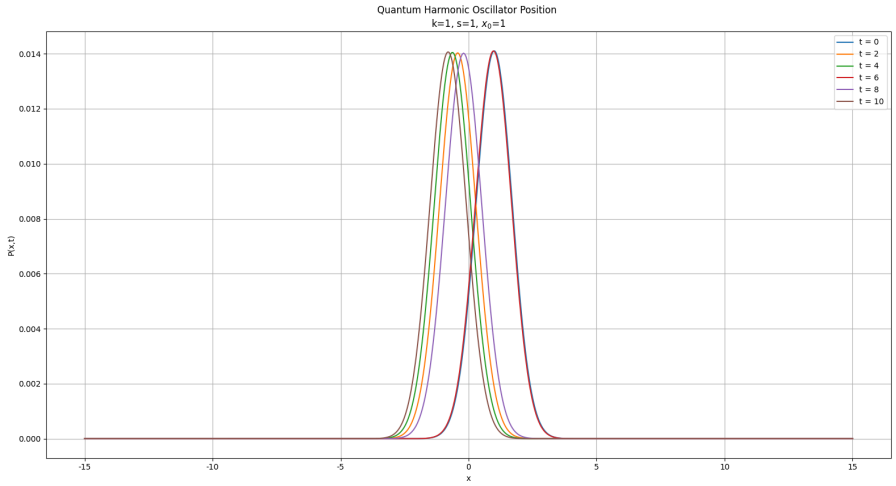

Figure 5: This graph illustrates the probability distribution according to the numerical solution of the TDSE for a 1D-harmonic oscillator. The x-axis is the position of the particle. The y-axis is the probability density of finding the particle at position x at time t. Different snapshots were taken at different times t, ranging from $t = 0$ (blue) to $t = 10$ (brown). The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 1$.

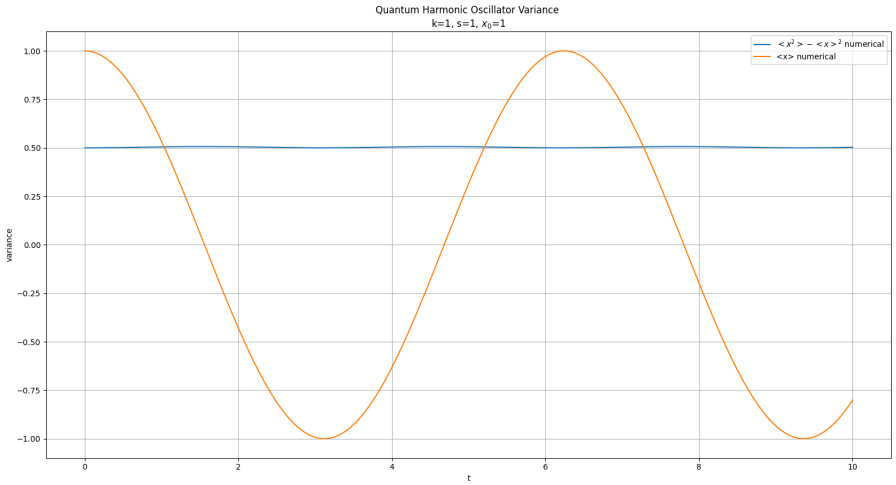

Figure 6: The expectation value $\langle x \rangle$ (orange) and variance $\langle x^2 \rangle - \langle x \rangle^2$ (blue) of x for Figure 1. The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 1$.

For the next simulation, we initialize the wave with $\Omega = 1$, $\sigma = 2$, $x_0 = 0$. This means we set the center of the wave at ground state back to $x = 0$, but increase its width to $\sigma = 2$. Figure 7 demonstrates that this results in the oscillation of uncertainty, but not position. This is further emphasized in Figure 8, where we can observe a constant expectation value $\langle x \rangle$ but an oscillating variance $\langle x^2 \rangle - \langle x \rangle^2$ between 2 and approximately 0.2, independent of x.

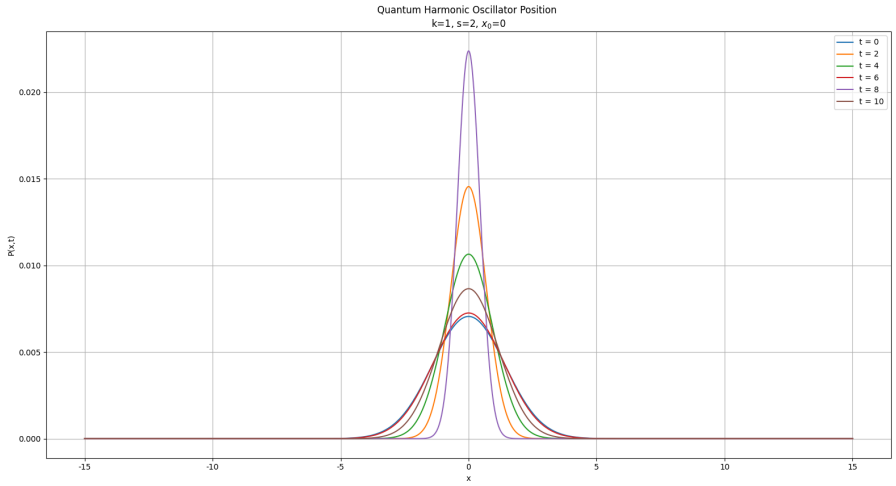

Figure 7: This graph illustrates the probability distribution according to the numerical solution of the TDSE for a 1D-harmonic oscillator. The x-axis is the position of the particle. The y-axis is the probability density of finding the particle at position x at time t. Different snapshots were taken at different times t, ranging from $t = 0$ (blue) to $t = 10$ (brown). The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 2$, $x_0 = 0$.

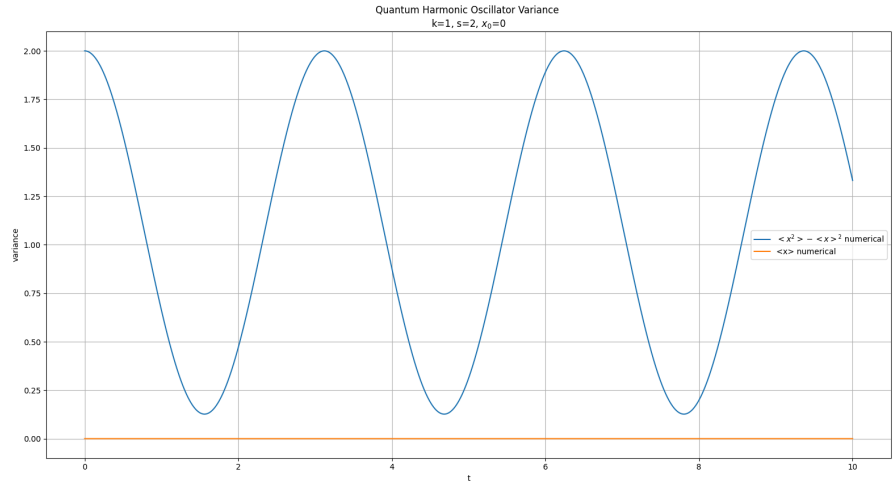

Figure 8: The expectation value $\langle x \rangle$ (orange) and variance $\langle x^2 \rangle - \langle x \rangle^2$ (blue) of x for Figure 1. The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 2$, $x_0 = 0$.

The next set of initial conditions is $\Omega = 2$, $\sigma = 1$, $x_0 = 1$. Here we increase the angular velocity Ω and reintroduce the positional offset of Figure 5. In Figure 9, we can observe the expected oscillations from $x = 1$ to $x = -1$, and small oscillations of uncertainty. As we increased the angular velocity, the positional oscillations are faster than in Figure 5, which can be derived from the faster oscillation of $\langle x \rangle$. Again, in Figure 10 we observe the expectation value $\langle x \rangle$ oscillating

with the center of the wave, and the variance oscillating with the uncertainty. The variance is smallest whenever $x = 0$.

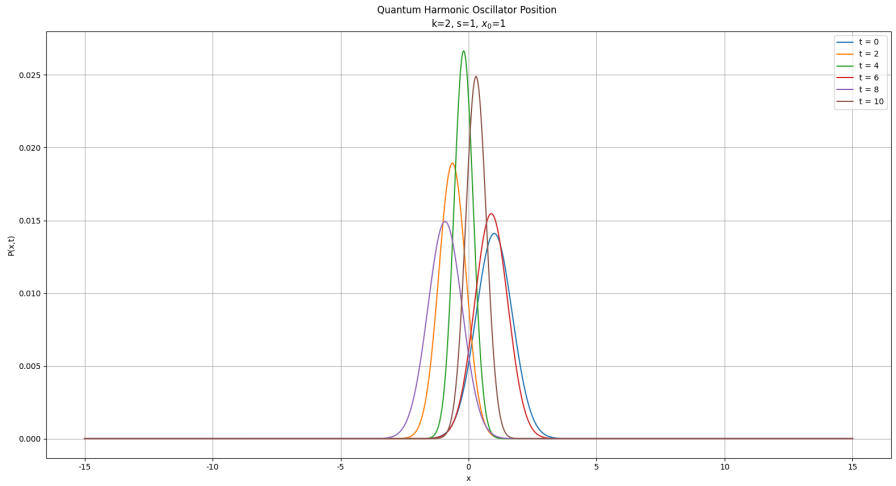

Figure 9: This graph illustrates the probability distribution according to the numerical solution of the TDSE for a 1D-harmonic oscillator. The x-axis is the position of the particle. The y-axis is the probability density of finding the particle at position x at time t. Different snapshots were taken at different times t, ranging from $t = 0$ (blue) to $t = 10$ (brown). The initial conditions are set to $\Omega(k) = 2$, $\sigma(s) = 1$, $x_0 = 1$.

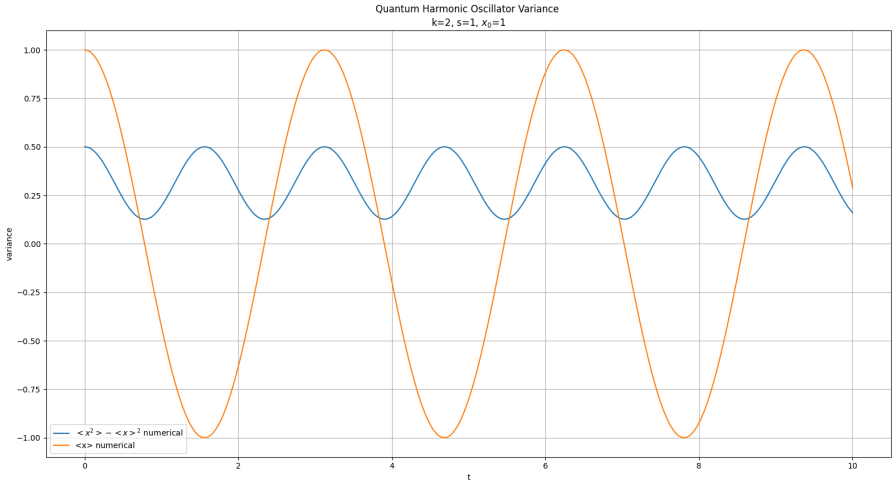

Figure 10: The expectation value $\langle x \rangle$ (orange) and variance $\langle x^2 \rangle - \langle x \rangle^2$ (blue) of x for Figure 1. The initial conditions are set to $\Omega(k) = 2$, $\sigma(s) = 1$, $x_0 = 1$.

In the last simulation, we combine all observations, as we initialize it with $\Omega = 2$, $\sigma = 2$, $x_0 = 2$. In Figure 11 and Figure 12 we observe strong oscillations in

uncertainty, minimized whenever $x = 0$. We observe the position oscillating between $x = 2$ and $x = -2$.

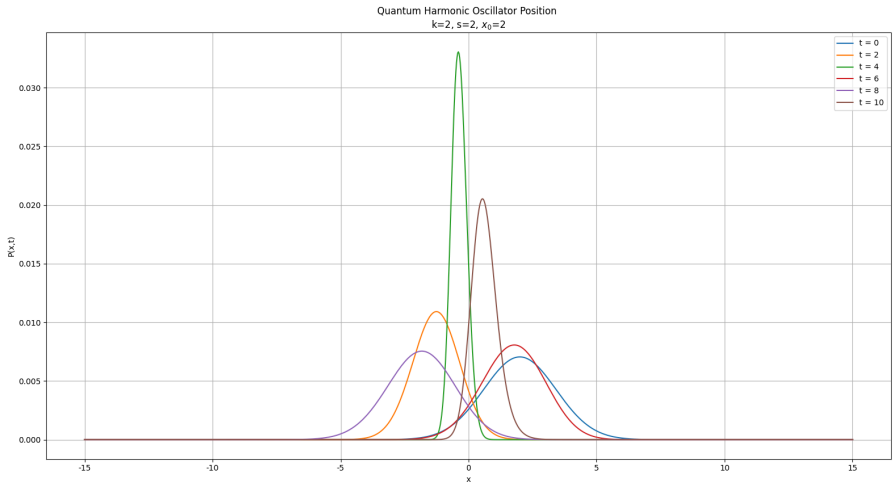

Figure 11: This graph illustrates the probability distribution according to the numerical solution of the TDSE for a 1D-harmonic oscillator. The x-axis is the position of the particle. The y-axis is the probability density of finding the particle at position x at time t. Different snapshots were taken at different times t, ranging from $t = 0$ (blue) to $t = 10$ (brown). The initial conditions are set to $\Omega(k) = 2, \sigma(s) = 2, x_0 = 2$.

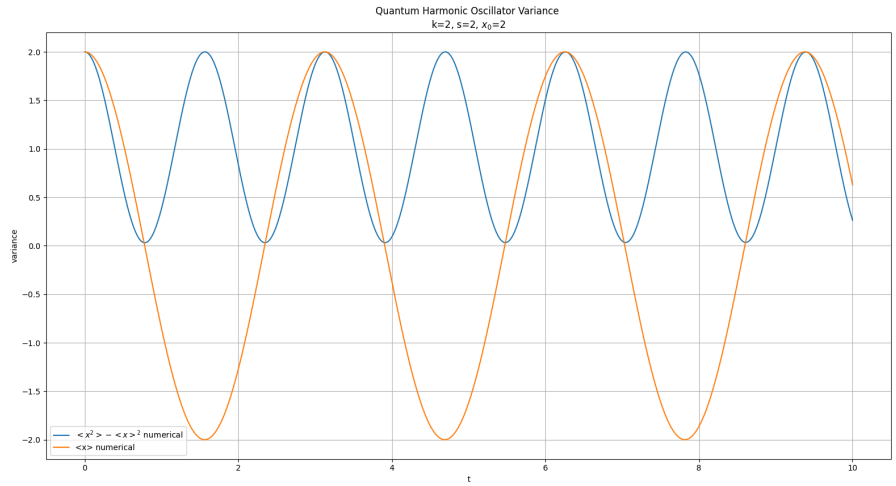

Figure 12: The expectation value $\langle x \rangle$ (orange) and variance $\langle x^2 \rangle - \langle x \rangle^2$ (blue) of x for Figure 1. The initial conditions are set to $\Omega(k) = 2, \sigma(s) = 2, x_0 = 2$.

In Figure 13 to Figure 17 we compare the numerical results for the expectation values and the variance with the analytical solutions. For all simulations, we observe that the numerical results of the expectation value $\langle x \rangle$ and the variance

$\langle x^2 \rangle - \langle x \rangle^2$ are close to the analytical results, demonstrating the effectiveness of the numerical solution using the product formula approach.

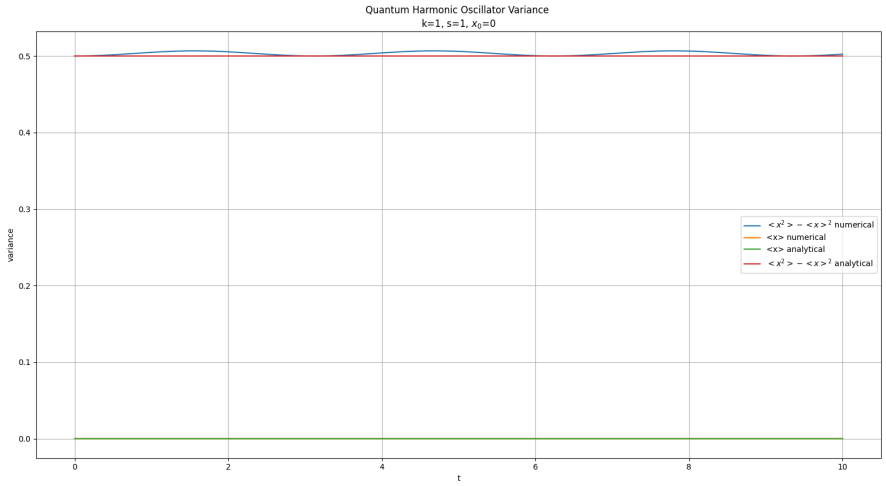

Figure 13: The expectation value $\langle x \rangle$ and variance $\langle x^2 \rangle - \langle x \rangle^2$ of x for Figure 1. The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 0$. The numerical solutions approximate the analytical solutions closely.

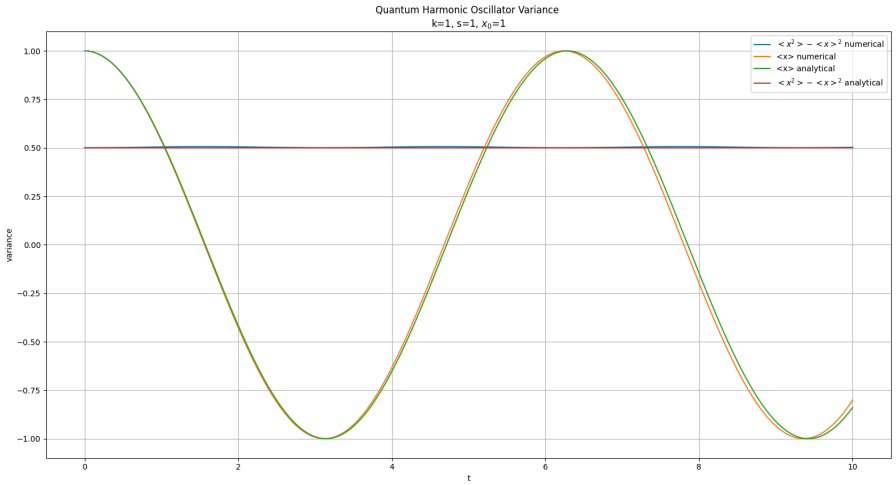

Figure 14: The expectation value $\langle x \rangle$ and variance $\langle x^2 \rangle - \langle x \rangle^2$ of x for Figure 1. The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 1$, $x_0 = 1$. The numerical solutions approximate the analytical solutions closely.

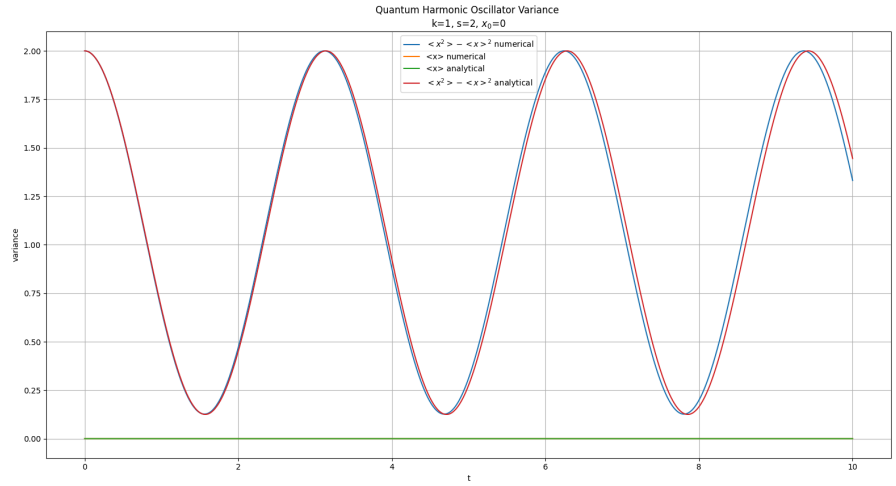

Figure 15: The expectation value $\langle x \rangle$ and variance $\langle x^2 \rangle - \langle x \rangle^2$ of x for Figure 1. The initial conditions are set to $\Omega(k) = 1$, $\sigma(s) = 2$, $x_0 = 0$. The numerical solutions approximate the analytical solutions closely.

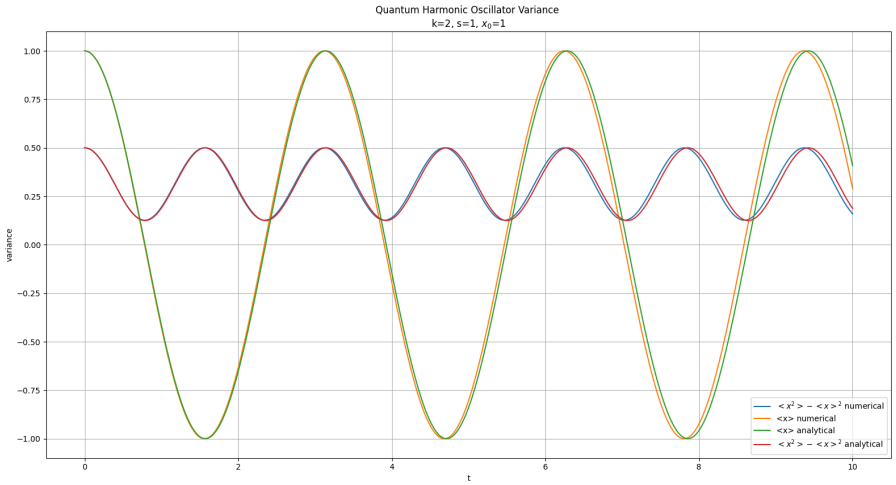

Figure 16: The expectation value $\langle x \rangle$ and variance $\langle x^2 \rangle - \langle x \rangle^2$ of x for Figure 1. The initial conditions are set to $\Omega(k) = 2$, $\sigma(s) = 1$, $x_0 = 1$. The numerical solutions approximate the analytical solutions closely.

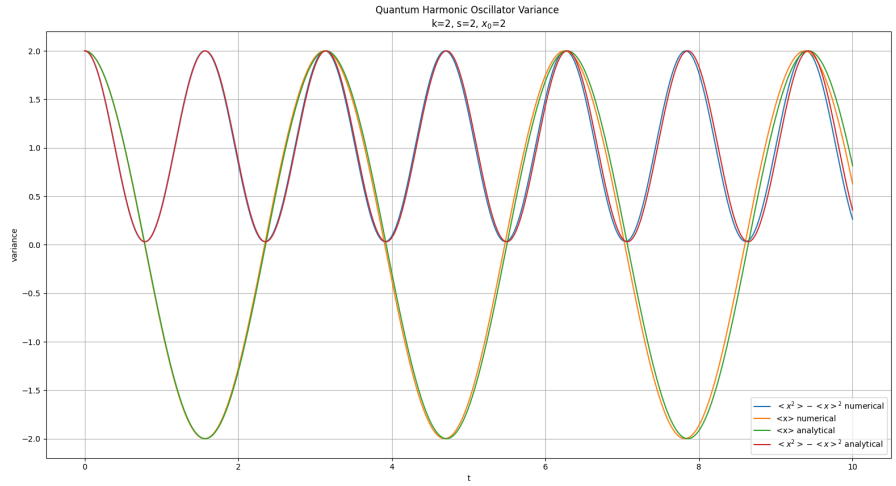

Figure 17: The expectation value $\langle x \rangle$ and variance $\langle x^2 \rangle - \langle x \rangle^2$ of x for Figure 1. The initial conditions are set to $\Omega(k) = 2$, $\sigma(s) = 2$, $x_0 = 2$. The numerical solutions approximate the analytical solutions closely.

Discussion

In this report, we explored the numerical and analytical solutions of the one-dimensional quantum harmonic oscillator. We began by discussing the importance of the harmonic oscillator in both classical and quantum mechanics, highlighting its wide range of applications from the behavior of electronic circuits to the movement of atoms in solids.

We implemented a numerical simulation to solve the time-dependent Schrödinger equation for the quantum harmonic oscillator. The simulation involved discretizing the problem in both space and time, initializing the wave packet, and evolving it using the second-order product formula. We computed and plotted the probability density and examined the expectation values and variances of the position over time. We compared our results with the analytical results and saw that the numerical results approximate the analytical results closely, given that the temporal resolution is high enough. With this, we demonstrated the effectiveness of the product formula approach for simulating quantum mechanics.

Appendix

```
#define _USE_MATH_DEFINES
#include <cmath>
#include <complex>
#include <iostream>
#include <vector>

#define WITHOUT_NUMPY
```

```cpp
#include "matplotlibcpp.h"

namespace exam
{
using namespace std::complex_literals;
namespace plt = matplotlibcpp;
typedef std::complex<double> complex;

double const Delta = 0.025;
int const L = 1201;
double const tau = 0.00025;
int const m = 40000;
double const pi = M_PI;

// controlls configurations
double const Omega = 1.;
double const sigma = 1.;
double const x_0 = 0.;

double const sigma_squared = sigma * sigma;

std::vector<complex> Phi_0() // Initial Wave Function
{
    std::vector<complex> phi_0(L);
    for (int i = 0; i < L; ++i)
    {
        complex x = (i - L / 2) * Delta;
        complex term_1 = std::pow(pi * sigma_squared, -0.25);
        complex term_2 = std::exp(-(((x - x_0) * (x - x_0)) / (2. * sigma_squared)));
        phi_0[i] = term_1 * term_2;
    };

    return phi_0;
}

std::vector<complex> V() // (Omega^2 / 2) * x^2
{
    std::vector<complex> v(L);
    for (int i = 0; i < L; ++i)
    {
        auto x = (i - L / 2) * Delta;
        v[i] = ((Omega * Omega) / 2) * (x * x);
    }
    return v;
}

void TDSE() // driving function, implementation of the product forumla approach
{
    std::vector<std::vector<complex>> phi_t; // matrix to store phi at different time steps for
plotting
    std::vector<complex> phi = Phi_0();      // initialize phi as described in exercise
    phi_t.emplace_back(phi);                 // add phi(t=0) to plot later
    std::vector<double> probabilities(L);    // for plotting
    std::vector<double> xv1(m + 1);          // <x>
    double xv2 = 0.;                         // <x^2>
    std::vector<double> variance_n(m + 1);   // variance vector numerical
    std::vector<double> expectation_a(m + 1); // variance vector analytical
    std::vector<double> variance_a(m + 1);    // variance vector analytical
    for (int i = 0; i < L; ++i)              // variance at t=0
    {
        double x = (i - L / 2) * Delta;
        double probability;
```

```
        probability = std::sqrt(std::pow(phi[i].real(), 2) + std::pow(phi[i].imag(), 2));
        probability *= std::sqrt(std::pow(phi[i].real(), 2) + std::pow(phi[i].imag(), 2));
        probability *= Delta;
        xv1[0] += probability * x;
        xv2 += (probability * (x * x));
    }
    variance_n[0] = (xv2 - (xv1[0] * xv1[0]));
    xv2 = 0;
    expectation_a[0] = (x_0)*std::cos(0);
    variance_a[0] = (0.5 * sigma_squared) * (std::cos(Omega * 0) * std::cos(Omega * 0))
                    + ((std::sin(Omega * 0) * std::sin(Omega * 0)) / (2 * sigma_squared * Omega
* Omega));

    // pre-computations
    complex const c = std::cos(tau / (4. * Delta * Delta));
    complex const is = 1i * std::sin(tau / (4. * Delta * Delta));
    auto v = V();
    // Product formula implementation
    for (int j = 1; j < m + 1; ++j)
    {
        double time = j * tau;
        complex p1;
        complex p2;
        for (int i = 0; i + 1 < L; i += 2) // K1/2
        {
            p1 = c * phi[i] + is * phi[i + 1];
            p2 = is * phi[i] + c * phi[i + 1];
            phi[i] = p1;
            phi[i + 1] = p2;
        }
        for (int i = 1; i < L; i += 2) // K2/2
        {
            p1 = c * phi[i] + is * phi[i + 1];
            p2 = is * phi[i] + c * phi[i + 1];
            phi[i] = p1;
            phi[i + 1] = p2;
        }
        for (int i = 0; i < L; ++i) // V
        {
            phi[i] *= std::exp(-(1i * tau * (1. + (Delta * Delta) * v[i] / (Delta * Delta))));
        }
        for (int i = 1; i < L; i += 2) // K2/2
        {
            p1 = c * phi[i] + is * phi[i + 1];
            p2 = is * phi[i] + c * phi[i + 1];
            phi[i] = p1;
            phi[i + 1] = p2;
        }
        for (int i = 0; i + 1 < L; i += 2) // K1/2
        {
            p1 = c * phi[i] + is * phi[i + 1];
            p2 = is * phi[i] + c * phi[i + 1];
            phi[i] = p1;
            phi[i + 1] = p2;
        }

        // Variance
        for (int i = 0; i < L; ++i)
        {
            double x = (i - L / 2) * Delta;
            double probability;
            probability = std::sqrt(std::pow(phi[i].real(), 2) + std::pow(phi[i].imag(), 2));
```

```
            probability *= std::sqrt(std::pow(phi[i].real(), 2) + std::pow(phi[i].imag(), 2));
            probability *= Delta;
            xv1[j] += probability * x;
            xv2 += probability * (x * x);
        }
        variance_n[j] = (xv2 - (xv1[j] * xv1[j]));
        xv2 = 0;
        expectation_a[j] = (x_0)*std::cos(Omega * time);
        variance_a[j] = (0.5 * sigma_squared) * (std::cos(Omega * time) * std::cos(Omega * time))
                        + ((std::sin(Omega * time) * std::sin(Omega * time)) / (2 * sigma_squared
* Omega * Omega));

        if (time != 2 && time != 4 && time != 6 && time != 8 && time != 10)
            continue;

        phi_t.emplace_back(phi);
    }

    // Probability calculations
    for (int i = 0; i < phi_t.size(); ++i)
    {
        for (int j = 0; j < L; ++j)
        {
             probabilities[j] = std::sqrt(std::pow(phi_t[i][j].real(), 2) + std::pow(phi_t[i]
[j].imag(), 2));
             probabilities[j] *= std::sqrt(std::pow(phi_t[i][j].real(), 2) + std::pow(phi_t[i]
[j].imag(), 2));
            probabilities[j] *= Delta;
        }
        plt::plot(probabilities, {{"label", "t = " + std::to_string(2 * i)}});
    }

    // Plotting
     plt::title("Quantum Harmonic Oscillator Position \nk=" + std::to_string((int)Omega) + ",
s=" + std::to_string((int)sigma)
                + ", $x_0$=" + std::to_string((int)x_0));
    plt::xlabel("x");
    plt::ylabel("P(x,t)");
    std::vector<double> xticks = {0, 200, 400, 600, 800, 1000, 1200};
    std::vector<std::string> labels = {"-15", "-10", "-5", "0", "5", "10", "15"};
    plt::xticks(xticks, labels); // Set custom ticks and labels
    plt::grid(true);
    plt::legend();
    plt::show();

     plt::title("Quantum Harmonic Oscillator Variance \nk=" + std::to_string((int)Omega) + ",
s=" + std::to_string((int)sigma)
                + ", $x_0$=" + std::to_string((int)x_0));
    plt::xlabel("t");
    plt::ylabel("variance");
    plt::plot(variance_n, {{"label", "$<x^2> - <x>^2$ numerical"}});
    plt::plot(xv1, {{"label", "<x> numerical"}});
    plt::plot(expectation_a, {{"label", "<x> analytical"}});
    plt::plot(variance_a, {{"label", "$<x^2> - <x>^2$ analytical"}});
    xticks = {0, 8000, 16000, 24000, 32000, 40000};
    labels = {"0", "2", "4", "6", "8", "10"};
    plt::xticks(xticks, labels);
    plt::grid(true);
    plt::legend();
    plt::show();
};
} // namespace exam
```

```cpp
int main()
{
    exam::TDSE();
    return 0;
}
```