

The Correlation Amplitude

Aarhus University

Background

This exercise will introduce you to time-dependence in Composer, but it will also illustrate a very important concept in physics: the correlation amplitude. Basically, we will explore how a displaced Gaussian evolves in a harmonic oscillator potential and focus on the interesting behavior that results.

The ground state of the harmonic oscillator we explored in the last section is given by a Gaussian function (setting \hbar and the particle mass m to one)

$$\phi_0(x) = Ae^{-\omega x^2/2}. \quad (1)$$

where A is some normalization constant. We can displace this ground state by some amount x_c , and this displaced ground state will be the initial state of our system

$$\phi_D(x, x_c) = Ae^{-\omega(x-x_c)^2/2}. \quad (2)$$

The flowfile

As before, you can set up this flowscene yourself, but if you get stuck, we have provided a flowfile that shows you how to do it. However, we will describe here how to set the problem up yourself to help you out a little bit. (Note that we assume that you have done the Spectrum exercise in what follows.)

To begin with, set up a harmonic oscillator potential using the *Spatial Dimension*, *Potential*, and *Hamiltonian* nodes just like you did in Problem 1 of the Spectrum exercise. The default potential in the *Potential* node is a harmonic oscillator, but remember to set the value of a properly!

We can set up the initial state $\phi_D(x, x_c)$ in two different ways. Try them both before moving on.

- **Analytic method:** First, we could put in the initial state analytically. Create an *Analytic Wavefunction* node and enter the expression for $\phi_D(x, x_c)$ as given in Eq. (2). To get an exponential in Composer, use the command `exp()`, and imaginary numbers are just represented by `i`. What about normalization? If you tick the *Normalize output* box, the output wavefunction is already normalized for you.
- **Numeric method:** In order to numerically create the initial state, you can create a second harmonic potential that is centered on x_c , that is $V(x, x_c) = \omega^2(x - x_c)^2/2$. The ground state of this potential is exactly $\phi_D(x, x_c)$. In order to do this in Composer, you have to create a new set of *Potential*, *Hamiltonian*, and *Spectrum* nodes. Enter the displaced potential into the new *Potential* node and connect everything up as in the Spectrum exercise. In order to get the ground state out of the *Spectrum* node, insert a *Linear Combination* node and wire it to the output of the spectrum node. The default behavior of the *Linear Combination* node is to give the ground state of the system, which in this case is exactly what we need.

You should also verify that these two approaches indeed give the same result. To do so, insert a *Fidelity* node into Composer and wire the two versions of $\phi_D(x, x_c)$ to the *Fidelity* node. Given two states ψ and ϕ , the *Fidelity* node calculates $|\langle\phi|\psi\rangle|^2$, which should be 1 if you did everything properly in Composer.

Now we are ready for time evolution. The *Time Evolution* node takes a single step (of length Δt) in time, but we want to take many such steps. To do this, we can create a *For Loop* scope in Composer (found under the *Misc* category). Then, we can put the *Time Evolution* node inside the *For Loop* scope. We can

connect the *Hamiltonian*, *Potential*, and wave function nodes to the scope by dragging them to the scope. You will see *boundary nodes* created as you drag wires from those nodes into the scope. From there, you can hook up the *Time Evolution* node and anything else placed inside the scope.

To set up time evolution, you need to create three *Scalar* nodes and connect them to the *For Loop* scope. These three scalars define the loop conditions as in other programming languages; they specify the starting time value (usually zero), the ending time value, and the increment dt . In order to get the *Time Evolution* node to work properly, you will need to create a *boundary node* that connects the increment dt through the *For Loop* scope to the *Time Evolution* node.

Now you should have everything you need to connect the *Time Evolution* node properly. In this way, we can step through from some initial time to a final time in steps of dt and explore the quantum dynamics that happen in this time. In particular, we will explore how the wavefunction $\phi(x, t)$ evolves in time from its initial state $\phi(x, 0) = \phi_D(x, x_c)$.

Problem 1

We will analyze this problem in two ways. First, in order to see what is going on, create a *Position Plot* and place it in the *For Loop* scope. If you attach the output of the *Potential* boundary node and the *Time Evolution* node to this plot, you can view what is going on in real time. Press the *Play* button and observe the wavefunction dynamics on the *Position Plot* node. Describe qualitatively the motion of the wavefunction.

Problem 2

This is very qualitative, though, so to be more quantitative, create a *Fidelity* node and place it in the scope. By wiring the initial state $\phi_D(x, x_c)$ from its boundary node to one input of the *Fidelity* node and wiring the wavefunction output from the *Time Evolution* node, we can calculate a quantity known as the **correlation amplitude** that measures the orthogonality of a time-evolved quantum state with respect to its initial state. For more information, see Chapter 2 of the second edition of Sakurai and Napolitano's quantum mechanics book.

In order to visualize this quantity, create a *Scalar Time Trace Plot* and put it inside the *For Loop* scope. Attach the output of the *For Loop* (labelled by i) and the output of the *Fidelity* node to the *Scalar Time Trace Plot*. Now, when you press play, you should see how the correlation amplitude changes as a function of time.

You should see a periodic structure in time. Argue why this happens.

Now vary the amplitude of the motion by changing x_c as well as the harmonic frequency ω . In particular, describe what parameters in the problem you need to change to change this structure from being more sinusoidal to being more comb-like. What parameters control the frequency of the periodic structure?

Problem 3

Set up the system from Problem 2 so that the structure of the correlation amplitude plot is comb-like. In these cases, one can describe two time scales:

- The **decoherence time** where the correlation amplitude has dropped significantly from unity and
- the **revival time** defining the amount of time it takes the correlation amplitude to return to unity.

How do these timescales depend on x_c and ω ? Note that the initial state $\phi(x, 0) = \phi_D(x, x_c)$ is a linear combination of N energy eigenstates of the harmonic oscillator. As we increase x_c , we are actually increasing the amount of energy that we put into the harmonic oscillator, so the number of eigenstates N that we need to accurately describe $\phi(x, 0)$ increases. Can you say anything about the how the two timescales above vary with N ?

Problem 4

If you change the potential to be anharmonic (for example, the quartic ax^4 potential), do you still see the same behavior? Why or why not?