

Numerical Simulation of the TDSE

Background

The Time Dependent Schrödinger Equation (TDSE) in one dimension in free space (where the potential term is zero) is:

$$i\hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial x^2} \quad (1)$$

If we compare this expression with the standard differential equation for a one dimensional wave, namely:

$$\frac{\partial^2 z}{\partial t^2} = -c^2 \frac{\partial^2 z}{\partial x^2} \quad (2)$$

we note two important differences between this expression and the TDSE.

First, ψ is only differentiated *once* with respect to t ; second, this differential is multiplied by i . The reason why the TDSE is a wave equation is that, while differentiating \sin twice gives $-\sin$, differentiating \sin once and then rotating by 90° in the complex plane also results in $-\sin$.

Numerical approximations

For the purposes of a simulation the TDSE can be reduced to

$$\frac{\partial \psi}{\partial t} = i k \frac{\partial^2 \psi}{\partial x^2} \quad (3)$$

where k is a (real) constant which determines the speed and wavelength of the wave.

The $\frac{\partial^2 \psi}{\partial x^2}$ term at ψ_x can be approximated by the expression

$$\psi_{x-1} - 2\psi_x + \psi_{x+1} \quad (4)$$

(where δx is assumed equal to one). Of course ψ_{x-1} , ψ_x and ψ_{x+1} are all complex numbers and must be added and subtracted as such.

Using equation (3) we now have

$$\frac{d\psi}{dt} \approx ik(\psi_{x-1} - 2\psi_x + \psi_{x+1}) \quad (5)$$

At each time step, the next value of ψ_x can be calculated by adding this quantity to the current value of ψ_x but a much better approximation can be achieved by adding twice this value to the *previous* value of ψ_x . In other words, we should use

$$\psi_{x,t+1} \approx \psi_{x,t-1} + 2ik(\psi_{x-1} - 2\psi_x + \psi_{x+1}) \quad (6)$$

rather than

$$\psi_{x,t+1} \approx \psi_{x,t} + ik(\psi_{x-1} - 2\psi_x + \psi_{x+1}) \quad (7)$$

Initial conditions

A good starting point is to set the amplitude and phase of ψ to a bell-shaped curve with a constant phase difference. This can be done using a function of the form

$$\psi_x = e^{-ax^2} e^{ipx} \quad (8)$$

where x runs from $-n$ to $+n$, a is a constant which determines the width of the pulse and p is a constant which determines its wavelength and hence speed.

The two endpoints should be set to zero and not updated. This will effectively confine the pulse to an infinite potential well.

Running the simulation

Having set the initial values of ψ we are almost ready to start the simulation. There is just one problem. We have set the *current* values of ψ , but our equation (6) also requires us to know the *previous* values of ψ . What we must do is to use the less accurate first order approximation for the first iteration (equation (7)), then then use equation (6) for all the subsequent iterations.

There is one last consideration. What value of k should we use?

In general terms, the larger the value of k the faster the simulation will run but the less accurate it will be. But with the TDSE there is another point to consider. It turns out that for any value of k greater than 0.25, the simulation is unstable and rapidly goes berserk – typically in only a few iterations. The reason for this is basically that the time step must be smaller than the time it takes for the simulation to propagate across each spatial step.

Displaying the results

A standard method of displaying the results is to plot the modulus of ψ on the Y axis and use colour to indicate the phase. Another method is to use a quasi-3D representation. These methods are illustrated below.

