Physics 3926 Project 4

Katie Brown

Part 1: Function Documentation

sch_eqn

```
sch_eqn(nspace, ntime, tau, method='ftcs', length=200, potential = [], wparam = [10, 0, 0.5])
```

The `sch_eqn` function numerically solves the one-dimensional, time-dependent Schrodinger equation using either the explicit forward-time-centre-step (FTCS) or Crank-Nicholson scheme (as specified by the user).

Input Parameters

Parameter	Type	Default	Description
nspace	int	none	Number of spatial grid points
ntime	int	none	Number of time steps
tau	float	none	Time step
method	string	'ftcs'	Method of numerical integration: -'ftsc': explicit forward time centre step scheme -'crank': Crank-Nicholson scheme
length	float	'200'	Size of spatial grid (grid extends from -L/2 to L/2)
potential	1-D array	[ ]	Spatial index values at which the potential equals 1
wparam	3-element list	[10,0,0.5]	Parameters for initial condition: [sigma0, x0, k0]: -sigma0 = packet width -x0 = centre of wave packet -k0 = average wavenumber

Returns

Variable	Type	Description
psi_sol	2-D Array	The amplitude of the wavefunction at each point in the spatial and time grids
x	1-D Array	The spatial grid values at which the solution was evaluated
t	1-D Array	The time grid values at which the solution was evaluated
probs	1-D Array	The total probability computed at each timestep

sch_plot

```
sch_plot(psi_sol, x, t, t_plot, plot_choice, save_plot=False)
```

The `sch_plot` function plots the solution to the Schrodinger equation, either as the amplitude of the wave-packet or probability distribution. Both are plotted against x (a spatial grid) and at a time specified

in the function call. Note that this function is designed to be used on the output of the `sch_eqn` function - this will ensure that the psi_sol, x and t input values are of the correct type and shape.

Parameter	**Type**	**Description**
psi_sol	2-D Array	The amplitude of the wavefunction at each point in the spatial and time grids
x	1-D Array	The spatial grid values over which to plot the solutions
t	1-D Array	The time grid values at which the solution was evaluated
t_plot	Int or Float	The specific time at which to plot the solution -Must be a time in the t array
plot_choice	String	Specifies which solution is to be plotted -'psi' prompts the wavefunction to be plotted -'prob' prompts the probability distribution to be plotted
save_plot	Boolean	Specifies whether to save the plot produced -If plot_choice is 'psi', figure is saved as WavefunctionReal.png -If plot_choice is 'prob', figure is saved as ProbabilityPlot.png

Physics 3926 Project 4

Katie Brown

Part 2: Report

Introduction

The objective of this project was to develop a program that would numerically solve the one-dimensional time-dependent Schrodinger equation. This was performed using either the Forward Time Centre Step (FTCS) or Crank-Nicholson scheme. This solution was passed to a plotting function that produced a plot of either the amplitude of the waveform or the probability distribution at a specified time. This program also computed the total probability at each time-step of the solution to ensure that this quantity was conserved. This process was tested for various initial conditions, spatial and time parameters, and potential distributions to confirm that it computes realistic solutions in all cases.

Program

The main function in this program, which numerically solves the Schrodinger equation using either the FTCS or Crank-Nicholson schemes was `sch_eqn`. This function first defines the Hamiltonian matrix as well as the transformation matrix corresponding to the scheme chosen. If FTCS was the specified method, a stability analysis would be performed by asserting that the spectral radius (the absolute value of the largest eigenvalue of the FTCS matrix) was no greater than one; if the solution was found to be unstable, the simulation would not proceed. The Gaussian wave packet was then initialized based on the given initial conditions, and this wave function was assigned to the first row of the solution matrix. The simulation then began, running through the specified number of time steps. At each iteration, the wave function from the previous step was multiplied with the transformation matrix and the result was appended to the 2-D solution array. Finally, the probability distribution at each time step was calculated by taking the square of the absolute value of the wave function solution. At each time step, the probabilities at every spacial grid point were summed to calculate the *total* probability at each time step. This function returns the solution of the wave function, the spatial and time grids on which the solutions were calculated, and the total probability at each time step.

The second function, `sch_plot` uses the outputs of the previous function to plot the real part of the wave function and the probability distribution corresponding to the calculated solution at a specified time.

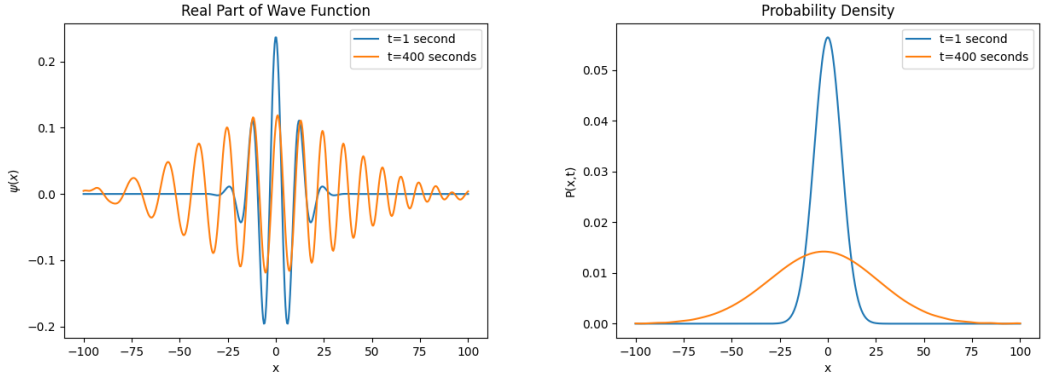
Testing

The default calling sequence used for all test cases was:

```
sch_eqn(500, 400, 1, method='crank', length=200, potential = [], wparam = [10, 0, 0.5])
```

The code was tested by varying each parameter, one at a time, while maintaining the rest constant. In each test case described, any parameters that are not explicitly mentioned can be assumed to have been set to the values above. For testing purposes, the program was slightly modified in order to include the solutions at multiple times on a single plot. In order to conform to project specifications, these changes were not included in the submitted code.

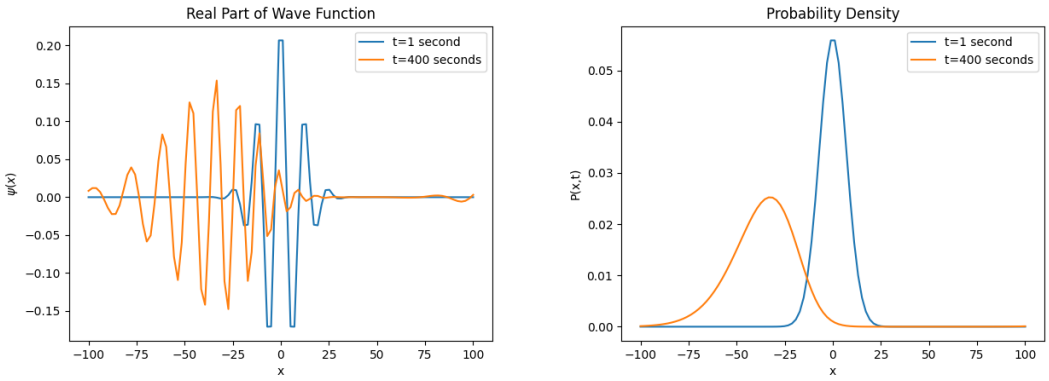
For each test run, the probability and wavefunction plots were produced at both the second and last time in the time grid (for the default parameters, this corresponds to times of 1 second and 400 second). This allowed for the observation of the system's evolution. The total probability at each time step was also printed, and it was confirmed that for each test case the probabilities were conserved over time. Note that with these default parameters, the FTCS method is unstable, and therefore no plots are produced. The plots were first produced with all default parameters (see Figure 1), which were consistent with the basic form of the wavefunction and probability density of the Schrodinger equation. As expected, the wave packet spread out spatially and decreased in amplitude over time.



(a) Evolution of wave packet (b) Evolution of probability density

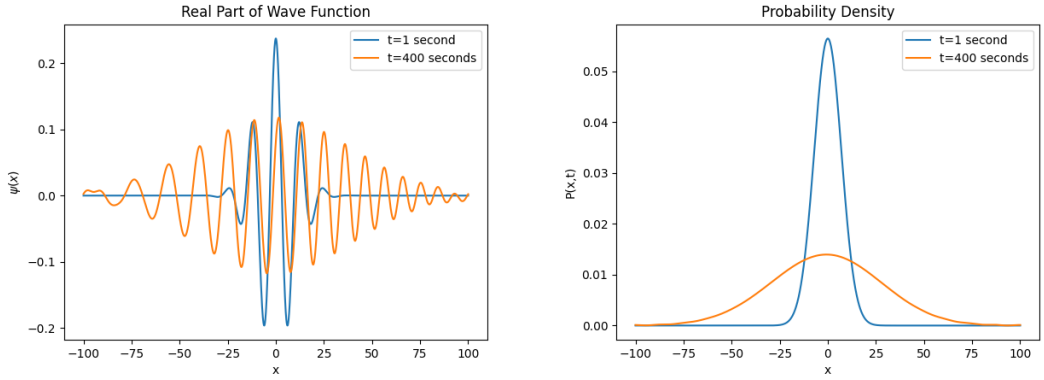
Figure 1: Default calling sequence.

Next, each parameter was systematically changed to test the solution at various conditions. Firstly, the plots were reproduced using different numbers of spatial grid points (achieved by varying the nspace parameter). It is evident that the plots get less smooth as nspace is decreased, although there is little increase in quality from nspace=500 to nspace=2000.



(a) Evolution of wave packet (b) Evolution of probability density

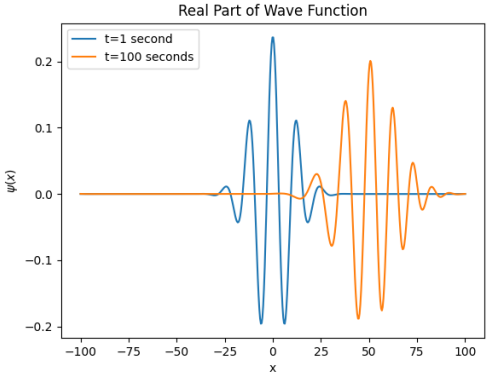
Figure 2: Less spatial grid points (nspace=100)



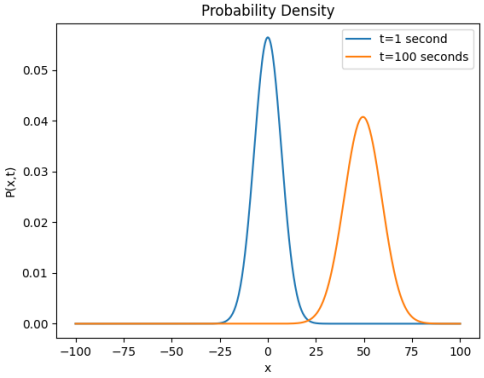
(a) Evolution of wave packet (b) Evolution of probability density

Figure 3: More spatial grid points (nspace=2000)

Similarly, the number of time points was then changed (by varying the ntime parameter) to plot the evolution of the system over 100 seconds and 1000 seconds. As would be expected, when plotted over 100 seconds, the wave packet demonstrated less change in its shape and amplitude, and traveled a smaller distance. In contrast, when plotted over 1000 seconds, the wave packet evolved significantly in shape and amplitude, as demonstrated in Figure 5.

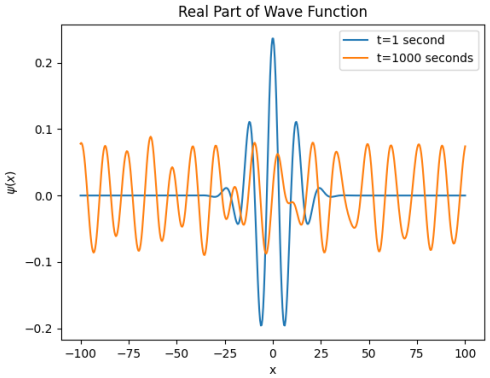


(a) Evolution of wave packet

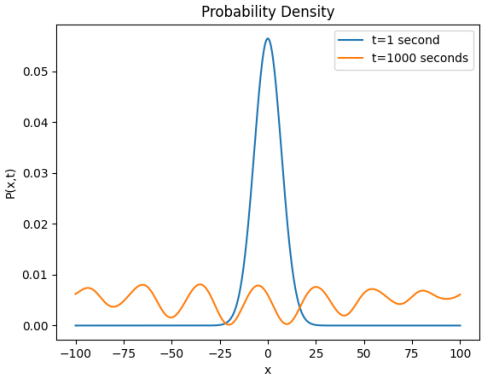


(b) Evolution of probability density

Figure 4: Evolution plotted over 100 seconds



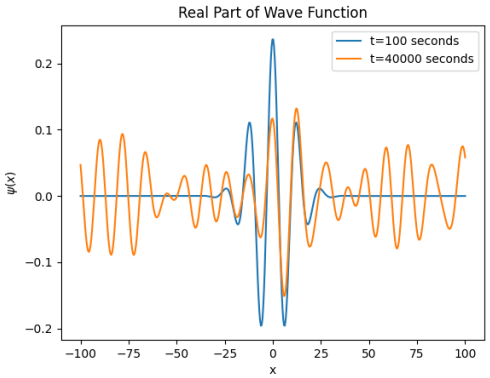
(a) Evolution of wave packet



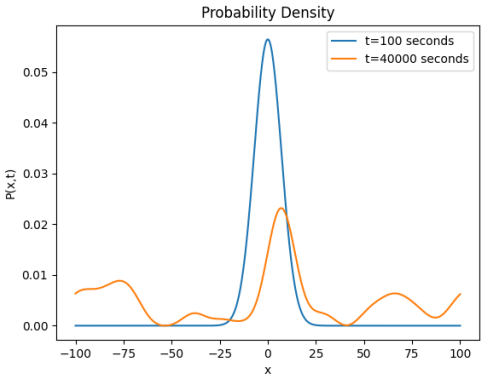
(b) Evolution of probability density

Figure 5: Evolution plotted over 1000 seconds

Next, the time-step used was varied (by altering the parameter tau) while maintaining the *number* of time-steps constant. When using a time-step of 0.01 seconds, the evolution of the system was plotted between 0.01 and 4 seconds. Both plots are thus very similar in form and position to those generated from the default calling sequence; these plots were not included. In contrast, when tau was increased to 100, the system was evaluated at 100 seconds and 40,000 seconds. Although the wave-packet was allowed to evolve over a very long time period, due to the number of time-steps remaining constant, it actually evolved a relatively small *number* of times during this period. This resulted in the somewhat dispersed yet disorganized appearance of Figures 6a and 6b.



(a) Evolution of wave packet

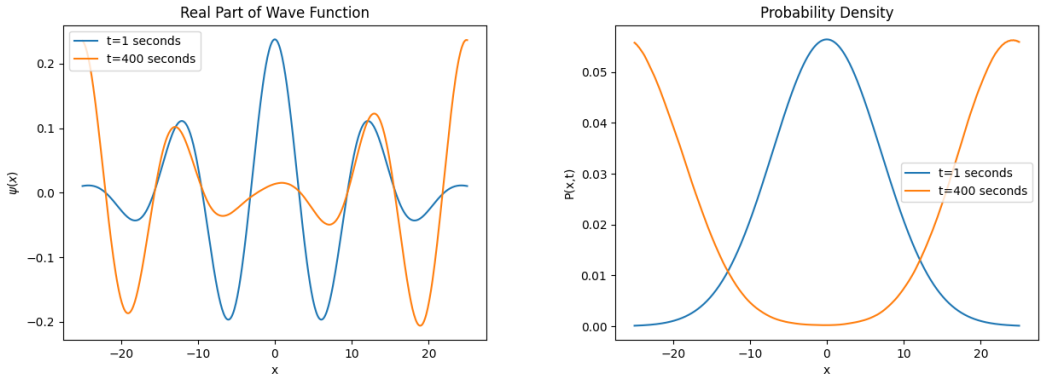


(b) Evolution of probability density

Figure 6: Evolution plotted over 40,000 seconds

The next parameter to be varied was the length of the spatial grid. With a length of 50 (and therefore a grid extending from -25 to 25), the wave-packet - which has an initial spatial standard deviation of 10 - occupies most of

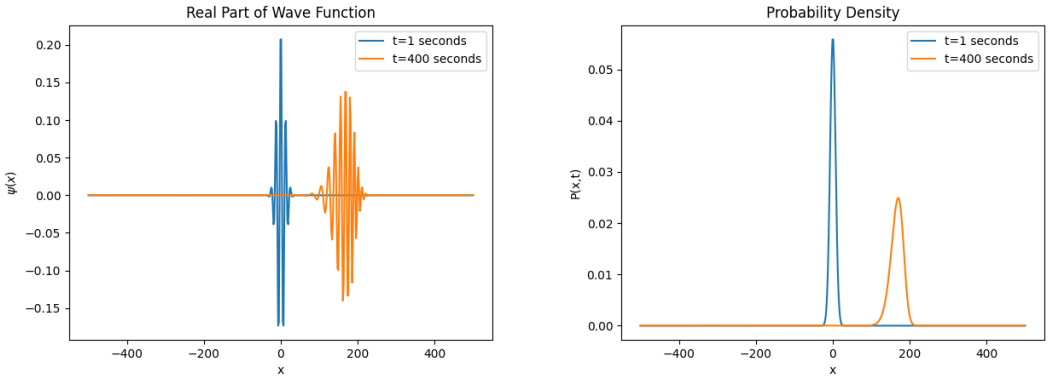
the spatial grid. This is a good opportunity to confirm that the system obeys the requirement of periodic boundaries; observe in Figure how the packet has a discontinuity at the right side of the plot and continues on the left. When plotted over a spatial grid length of 1000 (see Figure), the wave-packet maintains its form and amplitude, it is simply compressed in the plot.



(a) Evolution of wave packet

(b) Evolution of probability density

Figure 7: Spatial grid length of 50

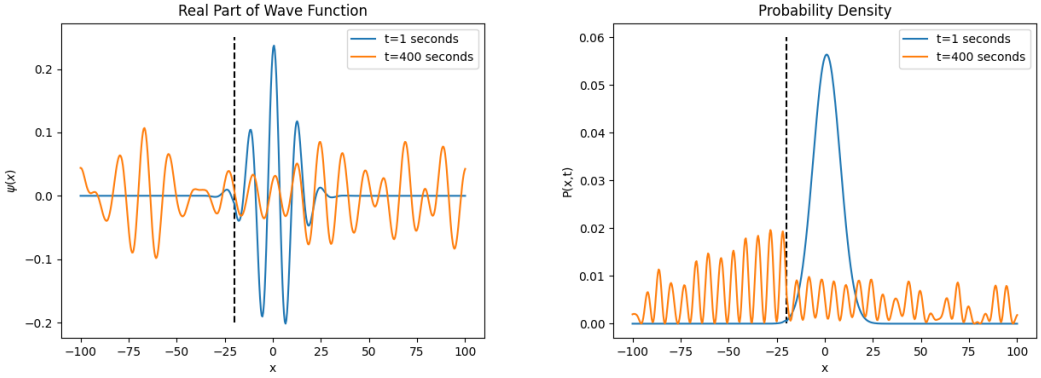


(a) Evolution of wave packet

(b) Evolution of probability density

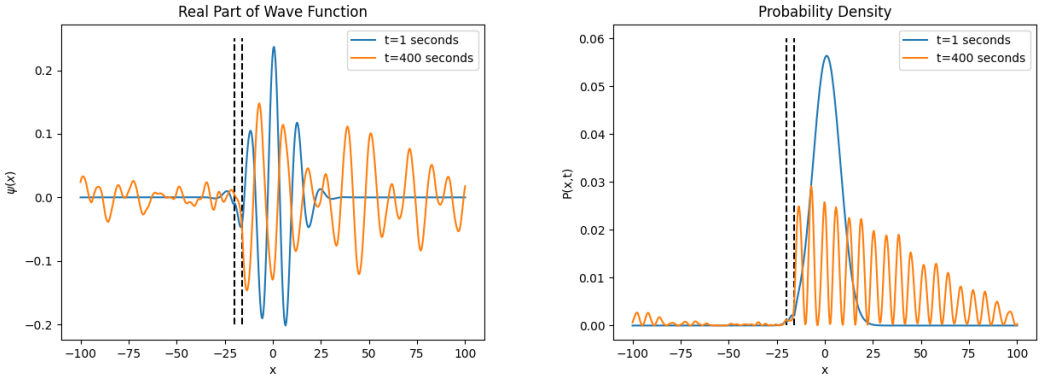
Figure 8: Spatial grid length of 1000

The next step was to test the effect of including non-zero potentials in the system. Firstly, potential was set to one at the spatial index of 200 (located at an x-position of -20 on the plot). The code was also run with potential set to one at all spatial indices from 200 to 210 (corresponding to x-positions from -16 to -20). The results are shown in figures 9 and 10. As one can observe in both cases, the amplitude of the wave function as well as its probability density falls to zero in the region of potential. In the case of potential at only one point, it can be seen that the probability density and wave function recover faster than in the case of a range of potential.



(a) Evolution of wave packet (b) Evolution of probability density

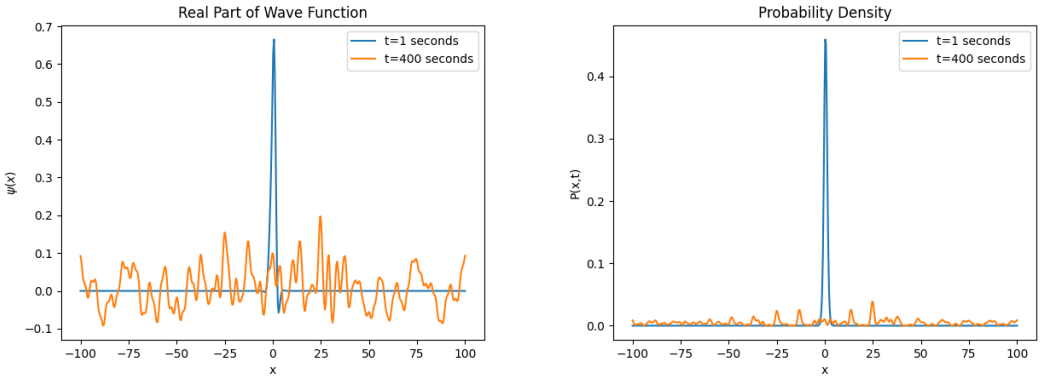
Figure 9: Potential set to 1 at x=-20 (denoted by the dashed black line)



(a) Evolution of wave packet (b) Evolution of probability density

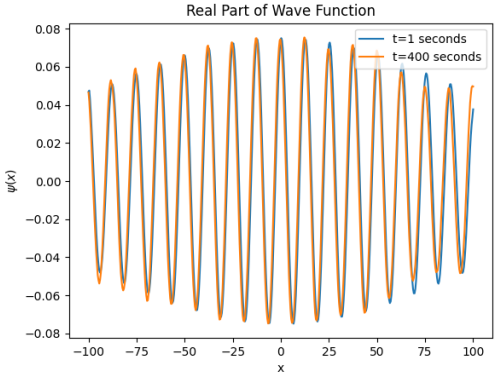
Figure 10: Potential set to 1 from x=-16 to x=-20 (denoted by the dashed black lines)

Next, the initial average spatial width of the wave-packet was varied by changing the sigma0 parameter in the wparam list. The initial width was decreased to 1 unit length and increased to 100 unit length, as illustrated in Figures 11 and 12, respectively. As can be seen in figure 12, with a large packet width the probability density appears to become somewhat chaotic. Upon further investigation it was determined that this was due to the fact that the spatial domain does not properly encapsulate the spatial extent of the wave packet. Upon increasing the length of the domain to 1000, it was found that the probability density became smooth.

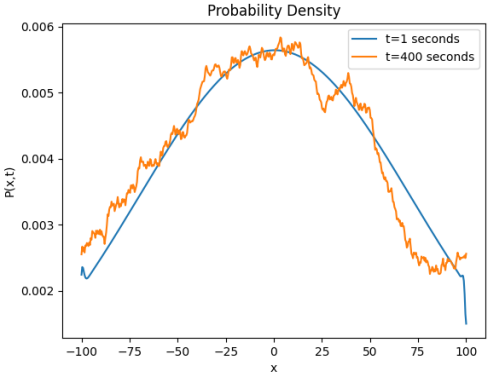


(a) Evolution of wave packet (b) Evolution of probability density

Figure 11: Initial average packet width of 1



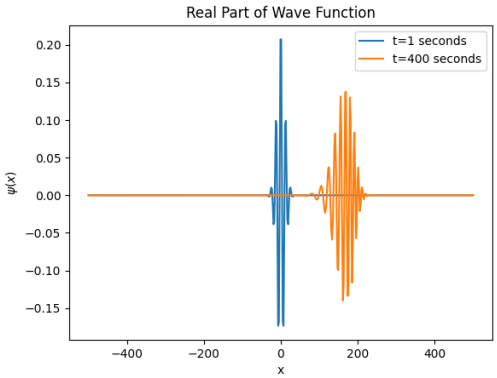
(a) Evolution of wave packet



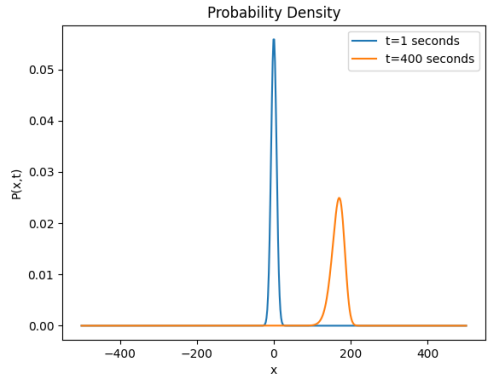
(b) Evolution of probability density

Figure 12: Initial average packet width of 100

The next parameter to be varied was the initial spatial location of the centre of the wave-packet (by changing the x0 parameter in the wparam list). When x0=50 was used, the form and behaviour of the wave-packet remained the same, it was simply shifted by 50 spatial grid points.



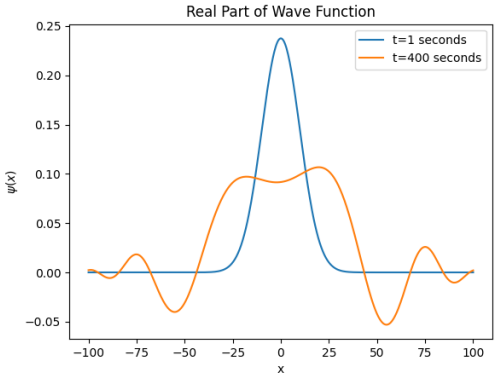
(a) Evolution of wave packet



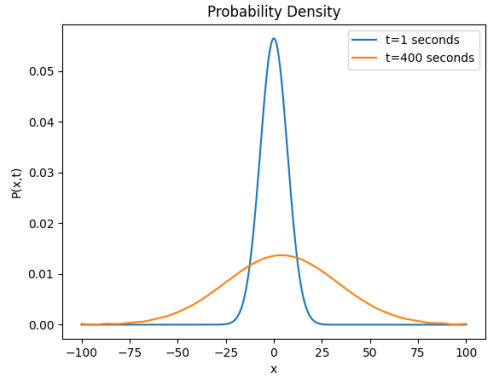
(b) Evolution of probability density

Figure 13: Spatial grid length of 1000

The final parameter to vary was the wavenumber k0. This parameter was decreased to 0.01 and increased to 50, as shown in Figures 14 and 15, respectively. As expected, this resulted in a change in the number of cycles (peaks and troughs of the wave) per unit distance.

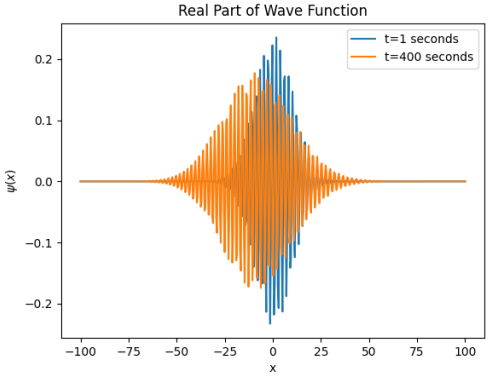


(a) Evolution of wave packet

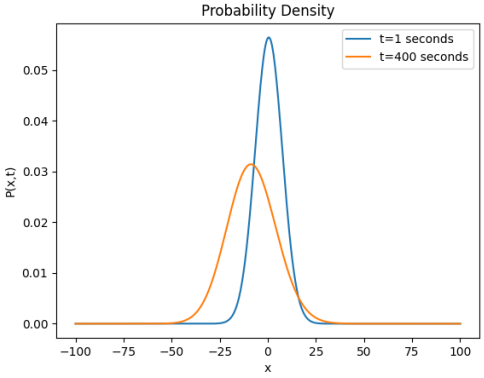


(b) Evolution of probability density

Figure 14: Initial wave number of 0.1



(a) Evolution of wave packet



(b) Evolution of probability density

Figure 15: Initial wave number of 50

Conclusions

The objective of this project was to produce a program to solve the one-dimensional time-dependent Schrodinger equation and plot the resulting wave function and probability distribution.Table 1 summarizes the different parameters that were used to test the functioning of this code in producing physically reasonable results.

Table 1: All key parameters used in the sch_eqn function and the values that were used for testing.

Parameter	**Default Value**	**Alternative Values Tested**
nspace	500	100, 2000
ntime	400	100, 1000
tau	1	0.01, 100
method	'crank'	'ftcs'
length	200	50, 1000
potential	[]	[200], list(range(200,211))
sigma0	10	1, 100
x0	0	50
k0	0.5	0.1, 50

In each test case used, the shape and evolution of the wave packet and probability distributions were consistent with expectations. Furthermore, total probability at each time-step was conserved in each case.