Phys Enph 479/879 Assignment #3: Space Discretized Leapfrog and the Schrödinger Equation

Michael Jafs[1, *]

[1] *Department of Physics, Engineering Physics and Astronomy,*
Queen's University, Kingston, ON K7L 3N6, Canada

(Dated: February 23, 2022)

The TDSE is solved numerically using spacial discretization and the Leapfrog algorithm. We compare the run times for our algorithm when using array slicing, a sparse matrix solver, and a full matrix solver to assemble our ODE's and show how array slicing is the fastest. We examine the wavefunction behaviour found using our solver for the case of a free-space, harmonic oscillator, and double well potential. We show that the Virial Theorem is indeed satisfied for the harmonic oscillator potential. Finally, we find that the wavefunction demonstrates a considerable amount of quantum tunnelling effects depending on our choice of parameters for the double well potential.

I. INTRODUCTION

The Time-Dependent Schröinger equation (TDSE) provides a simple way of investigating a variety of numerical methods for solving ordinary differential equations (ODE's). This is because the TDSE is a second order linear partial differential equation that does not usually possess an analytic solution. Depending on the potential considered, we require a numerical algorithm to provide us with the solution. This report focuses on a basic way of dividing up the range of infinite space and time values that a solution could take so that a numerical approach can be used to solve the TDSE. The latter half of the report then considers run times of three different approaches to the algorithm used and looks into various wavepacket behaviours for three potentials of interest. The sections can be stated as follows: In Section II, we outline the theory behind the numerical approach used to solve the TDSE, Section III considers a simple free-space potential and examines the run times of three different routines used in our algorithm. Section IV provides a glimpse at the effects of a wavepacket in a harmonic oscillator potential and Section V considers a more exotic "double well" potential. Finally, Section VI gives concluding remarks.

II. SPACE DISCRETIZED LEAPFROG

In this section we outline a method of numerically solving the Schröinger equation thus allowing for a quick way of analyzing wavefunction behaviour for a given potential. We consider a method called space discretization which motivates the use of the Leapfrog (Velocity Verlet) algorithm to numerically obtain a solution to the resulting set of coupled ordinary equations. The example we consider is the one dimensional Time-Dependent Schröinger equation (TDSE) in full generality:

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

However, we use atomic units (a.u.) to simplify our calculations, which is equivalent to setting $\hbar = m_e = e = 1$. Therefore, the TDSE takes the form of Eq. 2.

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

We implement a spacial discretization so that $x_j = jh$ where $j = 0, 1, ..., N$. Here, h is not to be confused with Plancks constant and is rather, the step size as we move along an array of space values. The discretization therefore yields $\psi_j(t) = \psi(jh, t)$ and $V_j(t) = V(jh, t)$. Inserting ψ_j and V_j into 2, and using the definition of the spacial derivative as a limit on h, we arrive at an expression which contains $N + 1$ coupled ODE's:

$$\frac{d\psi_j(t)}{dt}=\frac{i}{2h^2}(\psi_{j+1}(t)-2\psi_j(t)+\psi_j-1(t))-iV_j(t)\psi_j(t), \quad (3)$$

Which is much easier to solve numerically than 2. More concretely, since we demand that the wavefunction's probability amplitude remains equal to one, we have a conserved quantity. Additionally, since $\psi_j(t) = R_j(t)+iI_j(t)$, we have a system of coupled ODE's:

$$\frac{dR_j}{dt}=-\frac{1}{2h^2}I_{j-1}+\left(\frac{1}{h^2}+V_j\right)I_j-\frac{1}{2h^2}I_{j+1} \quad (4a)$$

$$\frac{dI_j}{dt}=\frac{1}{2h^2}R_{j-1}-\left(\frac{1}{h^2}+V_j\right)R_j+\frac{1}{2h^2}R_{j+1}, \quad (4b)$$

that possess the form necessary for the use of a symplectic integrator. Defining $a_j = \frac{1}{h^2}+V_j$, $b = -\frac{1}{2h^2}$ and introducing the vector identities:

$$\mathbf{R}=\begin{bmatrix}R_0\\R_1\\\vdots\\R_N\end{bmatrix},\ \mathbf{I}=\begin{bmatrix}I_0\\I_1\\\vdots\\I_N\end{bmatrix},\ A=\begin{bmatrix}a_0 & b & & \\ b & a_1 & b & \\ & \ddots & \ddots & \ddots \\ & & b & a_N\end{bmatrix}, \quad (5)$$

converts our problem into a system of equations for "generalized position" and "generalized velocity":

$$\frac{d\mathbf{R}}{dt}=A\mathbf{I},\quad \frac{d\mathbf{I}}{dt}=-A\mathbf{R}. \quad (6)$$

[*] 17msj2@queensu.ca

The form of 6 is identical to that of the system of coupled ODE's obtained when studying the classical harmonic oscillator. Therefore the traditional Leapfrog algorithm will be effective for use with 6. For one time step, the algorithm takes the form:

$$\mathbf{R}_{1/2} = \mathbf{R}_0 + A\mathbf{I}_0\frac{\Delta t}{2} \tag{7}$$

$$\mathbf{I}_1 = \mathbf{I}_0 - A\mathbf{R}_{1/2}\Delta t \tag{8}$$

$$\mathbf{R}1 = \mathbf{R}_{1/2} + A\mathbf{I}_1\frac{\Delta t}{2}, \tag{9}$$

which allows us to step forward in time and solve sequentially, each of the $N+1$ coupled equations denoted by 6. In the following section, we summarize the results of our integrator for the simple case of a particle in a free potential.

III. PARTICLE IN A FREE-SPACE POTENTIAL

We initially considered the TDSE in 1-D for the case of $V(x) = 0$, the case of a "free particle". Our initial ψ was given a simple wavepacket condition when $t = 0$ so that:

$$\psi(x, 0) = (\sigma\sqrt{\pi})^{-1/2} \exp\left[-\frac{(x - x_0)^2}{2\sigma^2} + ik_0x\right]. \tag{10}$$

Where σ is the width, x_0 is the center of the Gaussian, and k_0 is the initial average momentum of the particle. Our initial parameters are taken to be: $x_0 = -5, \sigma = 0.5, k_0 = 5$. Since our integrator will step through an array of values defined by 5, we also needed to tell the algorithm what to do when it encounters a non-existent value. We invoke periodic boundary conditions on ψ such that: $\psi_{N+1} = \psi_0$ and $\psi_{-1} = \psi_N$, so the solution should not interact with the boundary of the array but rather, just the given $V(x)$.

We exploited the tridiagonal form of 5 to speed up computational time when solving our ODE's. We chose to consider three ways of defining the set of coupled ODE's in 4. These were array slicing, writing A in 5 as a sparse matrix (using a routine from SciPy), and leaving A as a full matrix. Comparing run times of each method for a spacial grid size of $N = 1000$, ranging from -10 to $+10$ a.u. and 15000 time steps, where $dt = 0.5h^2$ is the time step size, we found that the slicing method was much faster than the other derivative construction methods. Using these parameters and grid spacing, we obtained the run times seen in Table II.

Derivative Method	Run times (s)
Slicing	∼0.710
Sparse Matrix	∼5.293
Full Matrix	∼217.22

Table I. Method of derivative construction and run times for a grid size of $N = 1000$ and 15000 time steps of dt.

The times in Table II are written as estimates, however the solve time seemed to be consistent with earlier times to within roughly 0.04 seconds. The method of array slicing was still faster by a large margin than the other methods and it is clear that some matrix manipulation routine prior to the matrix multiplication carried out in Eq. 4 produced significantly faster results. We even campared the run times between derivative approaches for different initial grid spacing parameters and total time steps. These comparisons are outlined in Tables I and III. We found that a smaller grid size and a smaller number of time steps each produced faster results, with the slicing method coming out as the fastest derivative method in all cases. We found that increasing the grid size by a factor of 2, while only producing slightly slower run times for the sliced and sparse approaches, seemed to significantly increase the run times when using a full matrix.

Derivative Method	Run times (s)
Slicing	∼0.0464
Sparse Matrix	∼0.347
Full Matrix	∼13.62

Table II. Method of derivative construction and run times for a grid size of $N = 1000$ and 1000 time steps of dt.

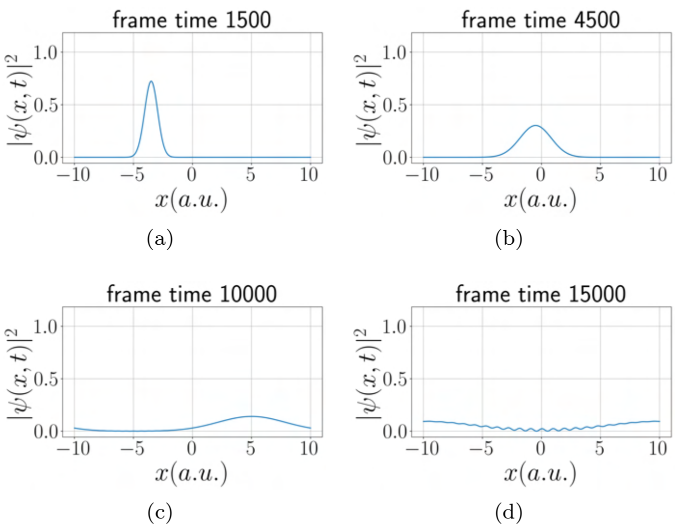
Derivative Method	Run times (s)
Slicing	∼0.0593
Sparse Matrix	∼0.397
Full Matrix	∼47.347

Table III. Method of derivative construction and run times for a grid size of $N = 2000$ and 1000 time steps of dt.

We also wanted to examine the behaviour of the wavepacket in both time and space to test the validity of our solver and to verify that our periodic boundary conditions were in fact being carried. Therefore, we simulated the wavepacket behaviour for the initial run time parameters (N=1000, number of timesteps=15000, grid from -10 to +10 a.u.) and produced an animation of which some representative snapshots can be seen in Figure 1. Using the method of array slicing, produced snapshots at 4 time intervals showing the wavepacket moving through space as we step through time and "spreading" out. The wavepacket also seems to travel from the right side of the graph through to the left as per our periodic boundary conditions. We ran the same simulation for both the sparse solver and the full matrix solution with the same results. Since our subroutine changed slightly between each method, it is worthwhile to compare final time frames for each approach to verify that the periodic boundary conditions are still satisfied. This comparison can be seen in Figure 2 which shows the final time frame for the simulation of the wavepacket using both the sparse and full matrix solver. Both plots are identical and it is clear that our periodic boundary conditions are satisfied. We were able to produce the same simulation result independent of the method used, albeit with different run times.

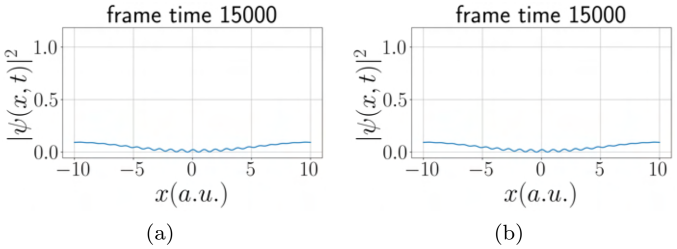
IV. HARMONIC OSCILLATOR POTENTIAL

We found that the fastest method for constructing our derivatives was the slicing method by a large margin, so we chose to use this method when solving the ODE's for



(a) (b) (c) (d)

Figure 1. Wavepacket behaviour at selected time frames within animation using the slicing approach. (a) - (d) only differ in slected timeframe. The simulation uses N=1000 and time steps=15000 with periodic boundary conditions.



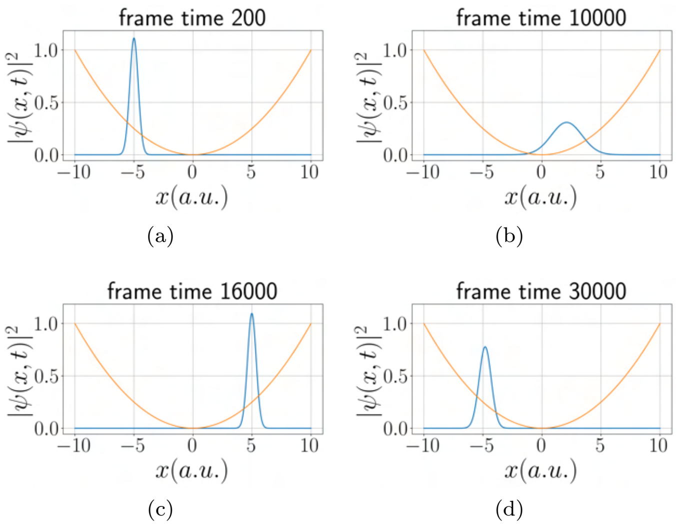
(a) (b)

Figure 2. Final wavepacket behaviour showing satisfied periodic boundary conditions. (a) Uses a sparse matrix. (b) Uses the full matrix solution. The simulation uses N=1000 and time steps=15000 with periodic boundary conditions.

the rest of our work. We chose to further explore the capabilities of our solving routine by considering a traditional harmonic oscillator potential:

$$V(x) = \frac{1}{2}x^2. \tag{11}$$

We then simulated the behaviour of the gaussian wavepacket from the previous section but used $x_0 = -5, \sigma = 0.5, k_0 = 0$. Again, using a grid size of 1000, ranging from a minimum position value of -10 to a maximum of +10 (a.u.), we simulated the wavepacket for two nominal periods. Figure 3 shows four snapshots of the simulation which take place during the first period of oscillation. The wave packet remained confined to the potential and did not seem to decrease in amplitude by any visual measurable margin during the two period length of each simulation. We show the probability density distribution as a contour plot in Figure 4. Since the wavepacket has a higher velocity at the bottom of the potential and a minimum velocity at the edges, we expect there to be a larger probability of finding the particle at the edges of the potential, where it is travelling the slowest. We also expect the wave packet to obey the uncertainty relation meaning the particle should appear more "smeared out" when in the center of the potential where the speed is the largest. Figure 4 shows the particle having a higher probability density at the edges of the potential (where $|x|$ is the largest) which is precisely what we expect from this configuration.



(a) (b) (c) (d)

Figure 3. Wavepacket behaviour in the harmonic oscillator potential. (a) - (d) demonstrate the solution moving forward in time with the timestep per graph labeled on top of each plot. The simulation uses N=1000 and the number of time steps equal to nT/dt where n is the number of periods, T is the period (2π), and dt is the timestep size. The moving line (blue) is $|\psi(x,t)|^2$, while the fixed line (orange) is the normalized potential). We use periodic boundary conditions.

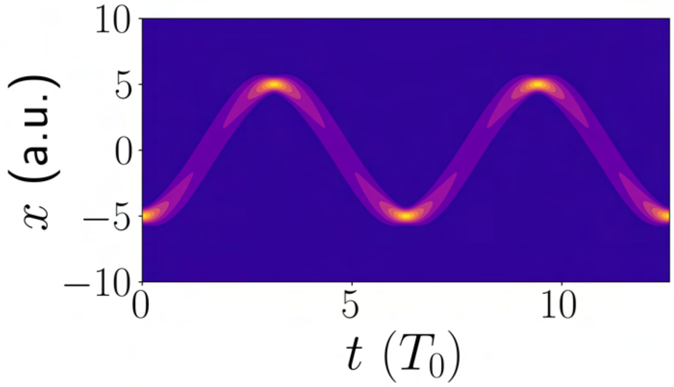


Figure 4. Probability density distribution corresponding to the simulation seen in Figure 3: wavepacket in the harmonic oscillator potential for two nominal periods of 2π.

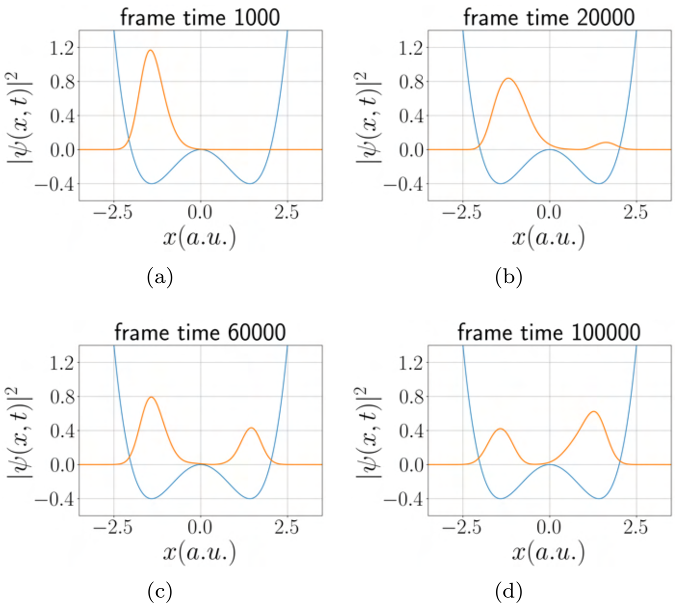
Our simulation allowed for the calculation of several common observables including the expectation of kinetic and potential energy. Using the definitions for the time-averaged expectation of kinetic and potential energy in x-representation and in a.u.:

$$\langle V \rangle = \int_{-\infty}^{\infty} \psi^*(x)V(x)\psi(x)dx \tag{12}$$

$$\langle T \rangle = \frac{1}{2}\int_{-\infty}^{\infty} \left|\frac{\partial\psi(x)}{\partial x}\right|^2 dx, \tag{13}$$

we checked that the Virial Theorem was satisfied to within a respectable margin. Running our simulation, we found that $\frac{1}{2}\langle T\rangle = \langle V\rangle$ to within 0.03% for each run.

V. PARTICLE IN A DOUBLE WELL POTENTIAL



(a) (b) (c) (d)

Figure 5. Wavepacket behaviour in the double well potential with a=1,b=4. (a) - (d) demonstrate the solution moving forward in time with the timestep per graph labeled on top of each plot. The simulation uses N=1000 and the number of time steps equal to nT/dt where n=4 is the number of periods, T is the period (2π), and dt is the timestep size. The moving line (orange) is $|\psi(x,t)|^2$, while the fixed line (blue) is the potential scaled by a factor of 10 to make the wavefunction behaviour more obvious.

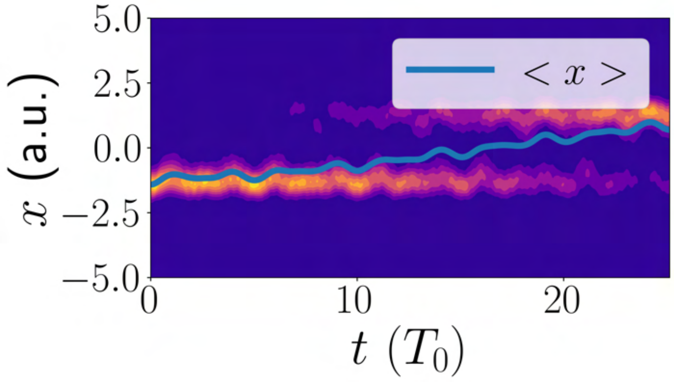
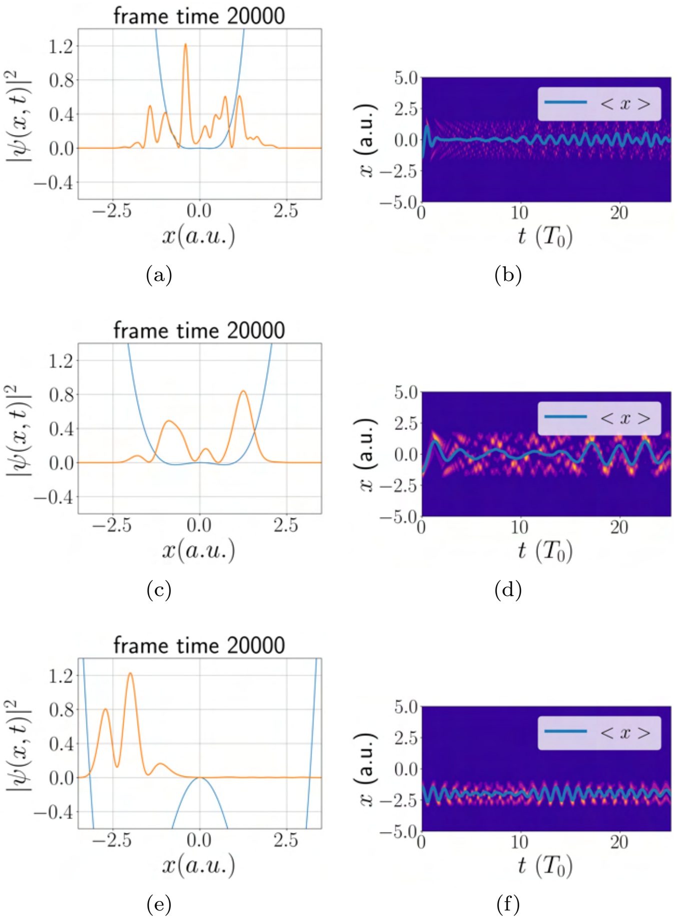


Figure 6. Probability density distribution for the double well potential using a=1 and b=4. The expectation value for the position over time is also plotted as a solid blue line.

Lastly, we considered the the case of a more exotic potential given by:

$$V(x) = ax^4 - bx^2, \tag{14}$$

where we let a and b take an integer value. This is a double well potential for which the Schrödinger equation does not usually have an analytic solution. We set up the problem similar to the sections above, but considered a smaller grid size (N=500, ranging from -5 to +5 (a.u.)) and provided the gaussian wavepacket with the following initial parameters: $x0 = -\sqrt{2}, \sigma = 0.5, k_0 = 0$. As an initial example, we set a=1 and b=4 and observed the wavepacket behaviour for four nominal periods. Figure 5 provides four snapshots from across the four periods and displays some of the unusual behaviour exhibited from the wavepacket inside the double well.



(a) (b) (c) (d) (e) (f)

Figure 7. Wavepacket behaviour (a),(c),(e) and corresponding probability density (b),(d),(f) for various double well parameters a and b. (a) a=6,b=1,(c)a=1,b=1,(d) a=1,b=10. The simulation uses N=1000 and the number of time steps equal to nT/dt where n=4 is the number of periods, T is the period (2π), and dt is the timestep size. The orange line is $|\psi(x,t)|^2$, while the blue line is the potential scaled by a factor of 10.

Unique to this potential is the fact that the wavefunction experiences a small quantum tunnelling effect. This effect is further exemplified is Figure 6 which shows the corresponding probability density distribution for the potential from Figure 5. Figure 6 displays the expected value of position over time overlayed on the probability distribution. Since these two features do not coincide at all times, we can see that the expected position of the particle is not always in the same location as the particle itself. We begin with a traditional Gaussian wavepacket, but our simulations show that there is a non-zero probability that the particle can be found in either of the sides of the potential at some later time t. To investigate the effects of the parameters a and b in the double well potential, we simulated the wavepacket behaviour for the case that $a/b > 1$, $a \sim b$ and $a/b < 1$. These three cases, along with the probability distribution can be seen in Figure 7. We found that for the case of $a/b > 1$ we see more tunnelling while for $a/b < 1$ the opposite is observed and the wavepacket is more confined to one of the wells of the potential. The two extremes are outlined in Figure 7 where for $a/b = 6$, we have a large amount of tunnelling (the wavefunction seems almost dissociated from the potential), whereas for $a/b = 1/10$, the wavefunction is mostly confined to one of the wells.

VI. CONCLUSIONS

We developed an approach to solving the TDSE using a Leapfrog algorithm by means of discretizing a grid of space points. This allowed us to also discretize an array of time values and step forward in time to numerically find a solution describing the wavepacket behaviour for a free-space, harmonic oscillator, and double well potential. We showed how the method of manual array slicing proved to be the fastest method of assembling derivative definitions when writing our algorithm, independent of how many grid points or time steps we chose for our simulation. We chose to use the method of array slicing then to simulate the wavepacket behaviour for the rest of our simulations. We found that in the harmonic oscillator potential, the wavepacket was confined to the potential and did not decrease in amplitude over a total time of two periods. We then considered the case of a double well potential with a variable depth. We found that as the depth of the well decreased, the wavefunction was much less confined to the potential and our numerical results demonstrated a significant amount of quantum tunnelling. Conversely, as the depth increased, The wavefunction was mainly fixed to the side of origination in the double well.