# 1 Introduction

In a previous project this semester we have solved the time-independent Schrödinger equation or energy eigenvalue equation by transforming it from a continuous second-order differential equation in space to a dicscrete eigenvalue problem. For this problem, though, we were only interested in obtaining bound state solutions and entirely neglected the scattering states. Since the time of Rutherford, results from scattering theory and experiments have provided us essential information about the structure and contents of our universe, and therefore, these scattering states are important in their own right.

For this work, we would like to explore scattering numerically in one dimension. Even though from the outset this seems quite restrictive, developing the scattering formalism in one dimension will be quite instructive, and in addition, many three dimensional problems, namely those related to central potentials, can be reduced to one dimensional problems.

# 2 Mathematical framework

The time-independent Schrödinger equation is given by[1]

$$-\frac{1}{2m}\frac{\mathrm{d}^2\psi_E(x)}{\mathrm{d}x^2} + V(x)\psi_E(x) = E\psi_E(x), \tag{1}$$

where $E$ is the energy eigenvalue of this equation and $\psi_E(x)$ is the corresponding eigenfunction. When $E < \lim_{|x|\to\infty} V(x)$, we have bound states, usually discrete levels, which are normalizable since they fall off exponentially as $|x| \to \infty$. On the other hand, when $E$ is larger than the potential in one of these asymptotic regions, we have a scattering state, and have a continuum of solutions. In the asymptotic regions, however, our solutions are of the generic form $e^{\pm ikx}$ and are therefore not normalizable and hence not physical. To solve definite problems, we must work with physical solutions in the context of wavepackets of the form

$$\psi(x) = \int \frac{\mathrm{d}k}{\sqrt{2\pi}}\widetilde{\psi}(k)e^{ikx}, \tag{2}$$

where the time-dependence can be generated by the time-evolution operator. Taking motivation from this, we simulate scattering, which inherently requires dynamics, in two ways.

## 2.1 Finite differences

For our first numerical method, we solve the time-dependent Schrödinger equation

$$i\frac{\partial\Psi(x,t)}{\partial t} = -\frac{1}{2m}\frac{\partial^2\Psi(x,t)}{\partial x^2} + V(x)\Psi(x,t). \tag{3}$$

[1] Note that we work in natural units where $\hbar = c = 1$.

directly by implementing a finite difference scheme. Using a forward difference formula for the time derivative, the three-point formula for the second derivative, and rearranging, we obtain

$$\Psi_n^{(m+1)} = \Psi_n^{(m)} + i\Delta t\left(\frac{\Psi_{n-1}^{(m)} - 2\Psi_n^{(m)} + \Psi_{n+1}^{(m)}}{2m\Delta x^2} - V_n\Psi_n^{(m)}\right), \tag{4}$$

where $\Psi_n^{(m)} = \Psi(x_n, t_m)$, $x_n = x_0 + n\Delta x$, and $t_m = m\Delta t$. Thus, we have a sequence defined by $\Psi_n^{(m)} = U^m\Psi_n^{(0)}$, where

$$U_{kn} = \delta_{kn} - i\Delta t H_{kn}, \tag{5}$$

where the discretized Hamiltonian

$$H_{kn} = -\frac{\delta_{k-1,n} - 2\delta_{kn} + \delta_{k+1,n}}{2m\Delta x^2} + V_k\delta_{kn}. \tag{6}$$

Such an equation reminds us of the unitary time evolution, which can be explored by direct matrix multiplication:

$$\begin{aligned} U_{kn}^\dagger U_{nk'} &= (\delta_{kn} + i\Delta t H_{kn}^\dagger)(\delta_{nk'} - i\Delta t H_{nk'}) = \delta_{kk'} + i\Delta t(H_{kk'}^\dagger - H_{kk'}) + \Delta t^2 H_{kn}^\dagger H_{nk'} \\ &= \delta_{kk'} + \mathcal{O}(\Delta t^2). \end{aligned} \tag{7}$$

Herein lies a primary difficulty of implementing this method: the eigenvalues of $U$ essentially scale the eigenvector components of our initial wavevector. Hence, for stability, we would like to select $\Delta t$ and $\Delta x$ such that all of these eigenvalues are less than one. Unfortunately, this is not possible since the eigenvalues of a unitary matrix have unit norm, prohibiting us from performing our time evolution in this manner.

## 2.2 Fourier transform

If we cannot perform time evolution directly, perhaps we can take inspiration from a different place. We know from quantum mechanics that our unitary time evolution operator $U(t, t')$ acts as $\Psi(x, t) = U(t, t')\Psi(x, t')$ and therefore satisfies the differential equation

$$i\frac{\partial U(t, t')}{\partial t} = HU(t, t'), \tag{8}$$

which has solution

$$U(t, t') = e^{-iH(t-t')} \tag{9}$$

if $H$ is independent of time, which will be the case in our scattering problems. Let us write $H = H_0 + V$, where $H_0$ is the free-particle Hamiltonian and $V$ contains the potential terms. If we select our time step $\Delta t$ between time slices $t_m$ and $t_{m+1}$, then using the Baker-Campbell-Hausdorff formula, we can write

$$U(\Delta t) = e^{-iH\Delta t} \approx e^{-iH_0\Delta t}e^{-iV\Delta t}\Big(1 + \mathcal{O}(\Delta t^2)\Big). \tag{10}$$

Since we are working in the context of wave-packets, we can act directly on the state at time $t_m$ as follows

$$\begin{aligned}\Psi(x,t_{m+1}) = U(\Delta t)\Psi(x,t_m) &= e^{-iV(x)\Delta t}\int \frac{\mathrm{d}k}{\sqrt{2\pi}}\widetilde{\Psi}(k,t_m)e^{-iH_0\Delta t}e^{ikx} \\ &= e^{-iV(x)\Delta t}\int \frac{\mathrm{d}k}{\sqrt{2\pi}}\widetilde{\Psi}(k,t_m)e^{-ik^2\Delta t/(2m)}e^{ikx}. \end{aligned} \tag{11}$$

Thus, for small time steps, our potential changes the phase of the wave-function locally and the free part of the Hamiltonian propagates our modes according to their velocity $k/m$.

Because the above equation is set up on the continuous spatial domain we could set up our wave-packets as functions numerically and implement a numerical quadrature to take the Fourier transform at each $x$, but this could become quite computationally expensive, resource and time-wise. Another approach utilizes the fast-fourier transform, which requires us to discretize our spatial domain. Doing so, however, also requires conjugate momentum domain to be discretized. The details for computing the Fourier transform above can be found in App. B.

# A Stability of the explicit and implicit finite difference methods

In this work, the operators we deal with are typically either hermitian or unitary, both of which are special cases of the broader class of normal matrices $A$ which commute with their hermitian conjugate $A^\dagger$. There is a theorem of equivalence statements in linear algebra that defines a normal matrix and states that any normal matrix $A$ is diagonalizable via a unitary similarity transformation and that there exists a set of eigenvectors which forms an complete orthonormal basis for the $n$-dimensional vector space. Let us denote the spectrum of $A$ by the set $\{a_n\}$ and the corresponding set of eigenvectors by $\{x^{(n)}\}$ Thus, any vector $\Psi$, which is analogous to our discretized wave-function, we can write

$$\Psi = \sum_n c_n x^{(n)}, \tag{12}$$

where $c_n = \Psi^\dagger x^{(n)}$. It follows then, that repeated action of $A$ on $\Psi$ yields

$$\Psi^{(k)} = A^k\Psi = \sum_n c_n a_n^k x^{(n)}. \tag{13}$$

If we now consider $A$ to be nearly unitary, as in the finite difference methods above for time-evolution, then we see our fundamental issue with implementing the finite difference methods above. Indeed, because of the discretization effects, our eigenvalues $a_n$ have approximately unit norm but not quite. Thus, for large times (i.e. in the limit $k \to \infty$), we have

$$A^k\Psi \to c_{n_{\max}} a_{n_{\max}}^k x^{(n_{\max})}, \tag{14}$$

where $n_{\max} = \text{argmax}(\{a_n\})$. Hence, if $a_{n_{\max}} > 1$, then our solution explodes, and if $a_{\max} < 1$, then our solution vanishes at large times. Of course then, even the slightest bit of unitarity violation in our matrix evolution is a fatal flaw for a finite difference method propagating the entire wave-function, both its real and imaginary parts, at equal times.

# B Relating the continuous and discrete Fourier transforms

The expressions we use for the continuous fourier and inverse fourier transforms of some function $f(x)$ are as follows:

$$f(x) = \int \frac{\mathrm{d}k}{\sqrt{2\pi}} \tilde{f}(k) e^{ikx} \tag{15}$$

$$\tilde{f}(k) = \int \frac{\mathrm{d}x}{\sqrt{2\pi}} f(x) e^{-ikx}. \tag{16}$$

On the other hand, for a collection of discrete data points $f_n$, their discrete Fourier and inverse Fourier transforms are as follows (consistent with the *numpy* conventions):

$$f_n = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} F_m e^{-2\pi i \frac{mn}{N}} \tag{17}$$

$$F_m = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f_n e^{-2\pi i \frac{mn}{N}}. \tag{18}$$

We can write the grid points in $x$-space as $x_n = x_0 + n\Delta x$ and those in $k$-space as $k_m = m\Delta k$ with $n, m \in \{0, 1, \ldots, N-1\}$, where

$$\Delta x \Delta k = \frac{2\pi}{N}. \tag{19}$$