PHYS 240: HW 1

A quantum particle moving freely in space is described by the wavefunction $\psi(x,t)$, which satisfies the time-dependent Schrödinger equation

$$i\hbar\frac{\partial\psi}{\partial t} = \hat{H}\psi = -\frac{\hbar^2}{2m}\frac{\partial\psi}{\partial x^2}.$$

Here, there is no external potential, and the Hamiltonian $\hat{H} = \hat{p}^2/2m$ consists of a kinetic term only. A complete set of solutions is provided by plane waves of the form $e^{i(kx-E_kt/\hbar)}$. The energy in each mode, $E_k = \hbar^2k^2/2m$, is a quadratic function of the wavevector k.

An arbitrary wavefunction can be constructed from a linear superpositions of such states,

$$\psi(x,t) = \int_{-\infty}^{\infty} dk\, w_k e^{i(kx-E_kt/\hbar)}.$$

At $t = 0$, the equation above is just a Fourier transform. Hence, a weight $w_k \sim e^{-\sigma_0^2(k-k_0)^2-ikx_0}$ leads to an initial Gaussian wave packet centered on x_0:

$$\psi(x,0) = \frac{1}{(\sqrt{2\pi}\sigma_0)^{1/2}}\exp\Big(-\frac{(x-x_0)^2}{4\sigma_0^2} + ik_0x\Big).$$

At all subsequent times, the probability density has the form

$$|\psi(x,t)|^2 = \frac{1}{\sqrt{2\pi}\sigma(t)}\exp\Big(-\frac{(x-x_0-\hbar k_0t)^2}{2\sigma(t)^2}\Big),$$

where $\sigma(t)^2 = \sigma_0^2 + t^2/4\sigma_0^2$. In other words, the packet center moves uniformly with group velocity $v_0 = (1/\hbar)\partial E_k/\partial k|_{k_0} = \hbar k_0/m$, while the packet of width $\sigma(t)$ spreads.

In this special Gaussian case, the solution can be derived analytically. We can also solve it numerically, computing the eigenvalue problem $E\psi = \hat{H}\psi$. We proceed by discretizing the spatial derivative:

$$E\psi(x) = \hat{H}\psi(x) = \frac{\hbar^2}{2m}\frac{\psi(x+\Delta x)+\psi(x-\Delta x)-2\psi(x)}{(\Delta x)^2}.$$

Then the matrix formulation of the problem on some finite line segment is

$$\frac{\hbar^2}{2m(\Delta x)^2}\begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix}\begin{pmatrix}\psi_1\\ \psi_2\\ \vdots\\ \psi_N\end{pmatrix} = E\begin{pmatrix}\psi_1\\ \psi_2\\ \vdots\\ \psi_N\end{pmatrix}.$$

Numerical libraries will give us the eigenvectors $\{\phi_i^{(\alpha)}\}$ and their corresponding energies $\{E^{(\alpha)}\}$. The complete packet evolution is given by

$$\psi(\Delta x\cdot i,t) = \psi_i(t) = \sum_{\alpha=1}^{N} c^{(\alpha)}\phi_i^{(\alpha)}e^{-iE^{(\alpha)}t},$$

where

$$c^{(\alpha)} = \sum \phi_i^{(\alpha)^*}\psi_i(x,0)$$

For convenience, we work in simplified units where $\hbar = m = \Delta x = 1$.

1. Download the files in the HW1 directory on Canvas. The `make` command will generate an executable `packet` that simulates a Gaussian wave packet in one and two spatial dimensions. The geometry is controlled by the variables `Lx` and `Ly`, which are set by the user through a command-line flag.

2. Run the program with the `evolution` command line argument, and view the resulting time evolution with the provided `gnuplot` script:

   ```
   $ ./packet -L=400 evolution -k=0.1,0 -w=15

   $ gnuplot movie.gp
   Press return to start
   ```

   The `gnuplot` animation shows the packet's motion alongside the exact analytical result. Try running the program with different values of the width. What happens when σ_0 is comparable to or smaller than the discretization length $\Delta x = 1$? Try various values of the wavevector and in particular `-k=1.57,0` and `-k=3.14,0`. What happens to the group velocity? How does the motion differ from that of the continuum packet?

3. Take a look at the code listing for `packet.cpp`. Read through the subroutines `build_Hamiltonian` and `eigensolve`, both of which are called in the first few lines of `main`. Be sure you understand how the Hamiltonian matrix is organized in "packed" storage format and sent to LAPACK's `DSEPVD` routine for diagonalization. (Notice that we have left all the elements on the main diagonal empty. This just amounts to an energy shift of 1 in the one-dimensional case and 2 in the two-dimensional case.)

4. Observe the two-dimensional problem:
   ./packet -L=40,40 evolution -k=1.57,1.57 -w=3

```
$ gnuplot movie.gp
```

What difference do you notice when you run it again with `k=1.57,0`?

5. Run the program with the `dos` command line argument. This will cause the program to dump a histogram of the density of states

$$g(E) = \sum_{\alpha} \delta(E - E^{(\alpha)})$$

to a file named `dos.dat`. In one dimension, the exact result is

$$\int_{-\infty}^{\infty} \frac{dk}{2\pi} \delta(E - E_k) = \int_0^{\infty} \frac{dk}{\pi} \delta(E + 1 - k^2/2) = \frac{1}{\pi\sqrt{2(1+E)}}.$$

Try making the comparison in `gnuplot`.

```
$ ./packet -L=400 dos
$ gnuplot
> plot "dos.dat" with lines, 1/sqrt(2*(1+x))/pi
```

Increase the mesh size up from 400 until the plot looks smooth. Use `gnuplot` to convince yourself that what you're actually generating is

$$\frac{1}{\pi\sqrt{(1+E)(1-E)}},$$

which is symmetric about $E = 0$. What's going on here?

Be sure to include your plots, one of "dos.dat" with lines and 1/sqrt(2*(1+x))/pi, and one of "dos.dat" with lines and 1/sqrt((1+x)*(1-x))/pi.

6. The exact density of states in two dimensions is a constant, independent of E. Try systematically changing the aspect ratio from `Lx = 400`, `Ly = 1` to `Lx = 20`, `Ly = 20` (or larger).

```
$ ./packet -L=400,1 dos
$ ./packet -L=200,2 dos
$ ./packet -L=100,4 dos
$ ./packet -L=80,5 dos
.
.
$ ./packet -L=20,20 dos
```

Does the density of states flatten out?

7. The `trajectory` mode shows how the semi-classical trajectory $(\langle x(t)\rangle - x_0, \langle y(t)\rangle - y_0)$ changes with the choice of initial wavevectors.

```
./packet -L=400 trajectory -w=5
A linear mesh of 400 points
Diagonalizing the Hamiltonian ...
Computing k=0
Computing k=0.261799
Computing k=0.523599
Computing k=0.785398
Computing k=1.0472
Computing k=1.309
Computing k=1.5708
Computing k=1.8326
Computing k=2.0944
Computing k=2.35619
Computing k=2.61799
Computing k=2.87979
Computing k=3.14159
$ gnuplot
> plot "traj.dat" u 2:3 w l
> plot "traj.dat" i 1 u 2:3, 0.261799*x
> plot "traj.dat" i 2 u 2:3, 0.523599*x
> f(x) = a*x
> fit f(x) "traj.dat" i 1 u 2:3 via a
> print a
0.257528127931512
> fit f(x) "traj.dat" i 2 u 2:3 via a
> print a
0.497505902926639
```

Extract the slopes of these curves and plot them versus their wavevector. The continuum result is $\partial E_k/\partial k = k$. Convince yourself that the numerical result is actually $\partial E_k/\partial k = \sin k$ (and with the units, $\partial E_k/\partial k = \sin(k\Delta x)/\Delta x$), i.e. plot "group_vel.dat" and $\sin x$ together.