Solving Schrodinger's equation: Free particles and uncertainty

We'll be exploring the evolution of a Gaussian packet, corresponding to the ground state of Hydrogen atom in a harmonic oscillator with frequency one Terahertz. We start by defining some constants.

In [ ]:
```
hbar = 1.054571628251774e-27;
omega = 1.e12;
protonMass = 1.672621637e-24;
m = protonMass;
a0 = sqrt(...); # RMS width of Gaussian
```

We define ψ on a lattice of Np points x, from -L/2 to L/2.

In [ ]:
```
L = ...;
Np = ...;
dx = ...;
x = linspace(-L/2,L/2-dx,Np)
```

Now we define the initial wavefunction $\psi(0)$ on our grid.

In [ ]:
```
psi0 = (...)**(1./4.) * exp(-...)
```

In [ ]:
```
plot(x,abs(psi0)**2)
xlim(-2*a0,2*a0);
```

If $\psi(k)$ is expressed in Fourier space, applying the kinetic energy is pointwise multiplication. $p = -i\hbar k$, so $p^2/2m = -\hbar^2 k^2/2m$. We convert from real space into Fourier space using a FFT (Fast Fourier Transform), which returns $\tilde{\psi}(k)$ at Np points separated by $dk = 2\pi/L$:
$k = [0, dk, 2dk, \ldots, (Np/2)dk, -(Np/2-1)dk, \ldots, -2dk, -dk]$
Notice that k grows until halfway down the list, and then jumps to negative values and shrinks. We define k and $k2 = k^2$ as arrays

In [ ]:
```
dk = ...;
k = zeros(Np);
for j in range(0,...):
    k[j] = ...*dk
for j in range(...,Np):
    k[j] = (j-Np) * dk
k2 = k**2
```

In [ ]:
```
plot(k2)
```

We now define the time evolution operator $\widetilde{U}_{kin}(t)$ in Fourier space. Note that the square root of minus one is 1j in Python.

```
In [ ]: def UkinTilde(t):
            return exp(-1j * ...)
```

and we define $\psi(t)$ using Fourier transforms.

```
In [ ]: from scipy import fft, ifft
        def psi(t):
            return ifft(...*fft(psi0))
```

```
In [ ]: P = ...
        plot(x, psi(P/4).real, x, psi(P/4).imag)
        figure()
        plot(x, ..., x, ...)
        figure()
        plot(x, ..., x, ...)
```

We can understand this spread as a consequence of the uncertainty principle. We can calculate the wavepacket width $\sqrt{\langle x^2(t)\rangle}$ by the discrete approximation to the integral $\langle x^2(t)\rangle \sim \sum x^2 |\psi(x)|^2 dx$

```
In [ ]: def width(t):
            return sqrt(sum(... * abs(...)**2 * ...))
```

Our initial wavepacket has a RMS width $\Delta x = a0$

```
In [ ]: width(0), a0
```

We know that a minimum uncertainty wavepacket like ours has $\Delta x \Delta p = \hbar/2$, so we expect the packet width to grow like vt with v given by the momentum uncertainty. We plot the comparison...

```
In [ ]: deltaP = ...;
        v = ...;
        times = linspace(0,2*P,100);
        widths = [width(t) for t in times];
        plot(times, widths, times, ...)
```