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.
Shortcuts: $\hbar$ = esc hb esc; ω = esc w esc.

```
ℏ = 1.054571628251774`*^-27;
ω = 10.^12;
protonMass = 1.672621637`*^-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.

```
L = ...;
Np = ...;
dx = ...;
x = Table[n dx - L / 2, {n, 0, Np - 1}];
```

Now we define the initial wavefunction ψ[0] on our grid. We'll store the maximum ψMax to set the scale for the graphics:

```
ψ[0] = ...;
ψMax = Max[Abs[ψ[0]]];
```

It's slightly complicated to make plots of arrays in *Mathematica*, since ListPlot takes {{x0,y0},{x1,y1}, ...} = Transpose[{x,y}]. We'll fix the vertical scale for use in later animations.

```
Plotψ[t_] := ListPlot[{Transpose[{x, Re[ψ[t]]}], Transpose[{x, Im[ψ[t]]}]},
  Joined -> True, PlotRange -> {- 1.1 ψMax, 1.1 ψMax}]
Plotψ2[t_] := ListPlot[Transpose[{x, Abs[ψ[t]]^2}],
  Joined -> True, PlotRange -> {0, ψMax^2}]
```

Check that the original state is roughly of width a0

```
Show[Plotψ2[0], PlotRange → {{-2 a0, 2 a0}, All}]
```

If ψ(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 to Fourier space using a FFT (Fast Fourier Transform), which returns ψ[k] at points separated by dk = 2π/L:

k= {0, dk, 2 dk, ..., (Np/2) dk, -(Np/2 -1) dk, ..., -2 dk, -dk}

Notice that k grows until halfway down the list, and then jumps to negative values and shrinks. Let's define k and k^2

```
dk = ...;
k = Join[Table[n dk, {n, 0, ...}], Table[n dk, {n, - ..., -1}]];
k2 = k^2;
ListPlot[k2]
```

We now define the time evolution operator Ukin in Fourier space in terms of k2

```
UkinTilde[t_] := Exp[-(I ...)];
```

We evolve the free particle until a time t.

```
ψ[t_] := InverseFourier[ ... * Fourier[ ψ[0]]]
P = ...;

Plotψ[P / 4]
Plotψ[ ...]
Plotψ[ ...]
```

The packet develops real and imaginary parts, and delocalizes (spreads out) through the box. We can use Animate to view this dynamics...

```
Animate[Plotψ[t], {t, 0, 3 P}]
```

We can understand this spread as a consequence of the uncertainty principle. We can calculate the wavepacket width Sqrt[<x^2(t)>] by the discrete approximation to the integral <x^2(t)> ~ sum(x^2 |ψ[t]|^2 dx.

```
width[t_] := Sqrt[Sum[ ... [[n]]^2 Abs[ ... [t][[n]]^2] ..., {n, 1, ...}]]
```

Our initial wavepacket has a RMS width
Δx = a0

```
{width[0], a0}
```

$\{1.77551\times10^{-8}, 1.77551\times10^{-8}\}$

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 v t with v given by the momentum uncertainty. We plot the comparison...

```
Δp = ...;
v = ...;
Plot[{..., ...}, {t, 0, 2 P}]
```