Quick guide to *Mathematica*

To get help, place the cursor on the function and press F1
To run all commands in one cell, place the cursor there and press Shift + Enter or Enter on the numerical keyboard.
Output of many commands is suppressed by ; at the end of the command. If you want to see the output, delete ;
Argument of functions must be in square brackets [...], braces {...} are for arrays, ranges etc., use double brackets [[...]] to access elements of arrays
expr /. {$x \to a$} means substitute a for x in expr, expr // function means apply function to expr, i.e. it is equivalent to function[expr].
D[f[x],x] = derivative of $f[x]$ with respect to x.
To define a function of x one can use $f[x\_] := 1 + x^2$, notice the underscore and the colon, or # nad & like in $f := 1 + \#^2$ &
N[expr] evaluates expr with machine precision, if you use the decimal point in the expression it will be also evaluated with machine precision
Special characters can be inserted by pressing Esc ... Esc, e.g. Esc p Esc gives π
Clear[...] is used to unset any variables which could have been assigned previously.
Some other useful commands: Simplify, Expand, Factor; Integrate, Series, Sum

Clear all symbols from previous evaluations to avoid problems

In[1]:=
```
Clear["Global`*"]
```

FFT method and quantum time evolution

Fast Fourier Transform algorithm from Numerical Recipes:

In[2]:=
```
myFFT[data_, isign_] :=
  Module[
   {nn, n, m, i, j, istep, mmax, theta, tmp, wtemp, wpr, wpi, tempr, tempi, wr, wi, out},
   nn = Length[data] / 2;
   out = data;
   (* reordering using bit-reversal algorithm *)
   n = BitShiftLeft[nn, 1];
   j = 1;
   Do[
    If[j > i,
     tmp = out[[j]];
     out[[j]] = out[[i]];
     out[[i]] = tmp;
     tmp = out[[j + 1]];
     out[[j + 1]] = out[[i + 1]];
     out[[i + 1]] = tmp;
```

```
 ];
 m = nn;
 While[m ≥ 2 && j > m,
  j = j - m;
  m = BitShiftRight[m, 1];
 ];
 j = j + m,
 {i, 1, n - 1, 2}
];
(* Danielson-Lanczos algorithm -
 multiplying values by factors and combining them *)
mmax = 2;
While[n > mmax,
 istep = BitShiftLeft[mmax, 1];
 theta = isign * 2.0 * π / mmax;
 wtemp = Sin[0.5 * theta];
 wpr = -2.0 * wtemp * wtemp;
 wpi = Sin[theta];
 wr = 1.0;
 wi = 0.0;
 Do[
  Do[
   j = i + mmax;
   tempr = wr * out[[j]] - wi * out[[j + 1]];
   tempi = wr * out[[j + 1]] + wi * out[[j]];
   out[[j]] = out[[i]] - tempr;
   out[[j + 1]] = out[[i + 1]] - tempi;
   out[[i]] = out[[i]] + tempr;
   out[[i + 1]] = out[[i + 1]] + tempi,
   {i, m, n, istep}
  ];
  (* avoiding evaluation of Sin[] in each step,
  instead trigonometric recurence is used *)
  wtemp = wr;
  wr = wr * wpr - wi * wpi + wr;
  wi = wi * wpr + wtemp * wpi + wi,
  {m, 1, mmax - 1, 2}
 ];
 mmax = istep;
];
If[isign == 1,
 Return[out],
 Return[out / nn]
];
];
```

One step of time evolution of the 1D quantum system using the VTV split-operator method.
Psi is supposed to be a complex array containing $\psi(x)$ evaluated on the grid,
V an array of potential *V(x)* also on the grid, μ is the mass and dx and dt are the space and time step.

In[3]:=
```
myOneStepVTV[Psi_, V_, mu_, dx_, dt_] :=
  Module[{n, expV, psicmpl, psix, psip, tmp, sinT, cosT},
   n = Length[Psi];
   expV = Exp[-i * V * dt / 2];
   (* applying exp(-i V dt/2) to ψ(x) *)
   psicmpl = expV * Psi;
   (* FFT to p-representation *)
   psix = Flatten[Transpose[{Re[psicmpl], Im[psicmpl]}]];
   psip = myFFT[psix, -1];
   (* applying exp(-i T dt) to ψ(p) *)
   Do[
    (* positive momenta *)
    tmp = -((2 π (i - 1)) / (n * dx))^2 * dt / (2 * mu); (* -T dt *)
    sinT = Sin[tmp]; cosT = Cos[tmp];
    j = 2 * i;
    tmp = psip[[j - 1]];
    psip[[j - 1]] = cosT * tmp - sinT * psip[[j]];
    psip[[j]] = sinT * tmp + cosT * psip[[j]];
    (* negative momenta *)
    tmp = -((2 π (i - 1 - n / 2)) / (n * dx))^2 * dt / (2 * mu); (* -T dt *)
    sinT = Sin[tmp]; cosT = Cos[tmp];
    j = n + 2 * i;
    tmp = psip[[j - 1]];
    psip[[j - 1]] = cosT * tmp - sinT * psip[[j]];
    psip[[j]] = sinT * tmp + cosT * psip[[j]],
    {i, 1, n / 2}
   ];
   (* FFT to x-representation *)
   psix = myFFT[psip, 1];
   (* applying exp(-i V dt/2) to ψ(x) *)
   psicmpl = psix[[1 ;; 2 * n ;; 2]] + i * psix[[2 ;; 2 * n ;; 2]];
   psicmpl = expV * psicmpl;
   Return[psicmpl]
  ];
```

Time evolution of the free wave packet - exact evolution

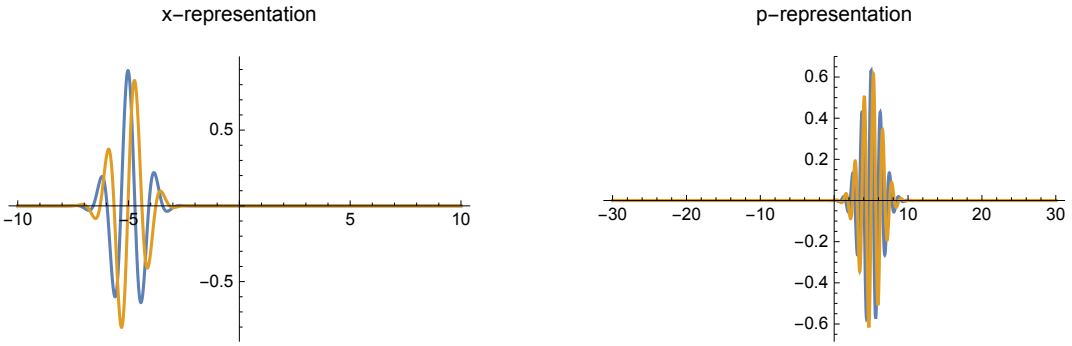
The initial Gaussian wave packet:

In[4]:=
```
σ = 1 / 2; x0 = -5; p0 = 5;
ψ[x_] = Exp[-(x - x0)^2 / (4 σ^2) + i x p0] / (2 π σ^2)^(1 / 4)
ψp[p_] = InverseFourierTransform[ψ[x], x, p]
GraphicsRow[{Plot[{Re[ψ[x]], Im[ψ[x]]},
    {x, -10, 10}, PlotRange → All, PlotLabel → "x-representation"],
  Plot[{Re[ψp[x]], Im[ψp[x]]}, {x, -30, 30}, PlotRange → All,
   PlotLabel → "p-representation"]}]
```

Out[5]=
$$e^{5 i x-(5+x)^2} \left(\frac{2}{\pi}\right)^{1/4}$$

Out[6]=
$$\frac{e^{-25-\frac{1}{4}((-5-10 i)+p)^2}}{(2\pi)^{1/4}}$$

Out[7]=

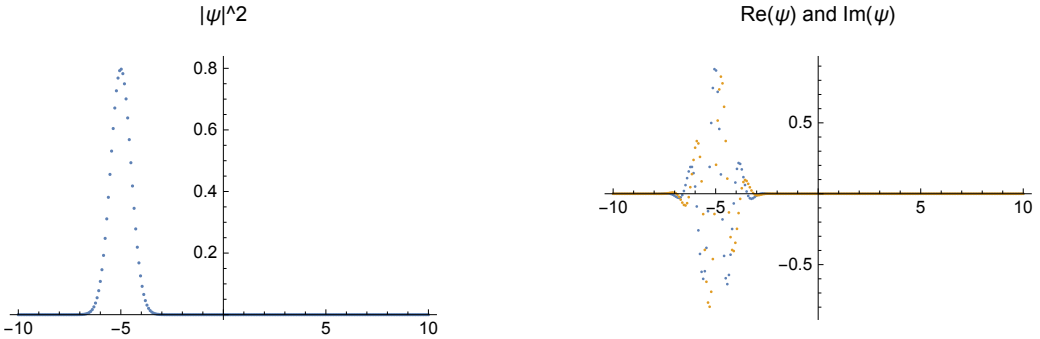


Initial wave packet on the grid:

In[8]:=
```
n = 2^8; Print["Number of points: ", n];
(* equadistatnt grid with the step h *)
xmin = -10; xmax = 10;
dx = (xmax - xmin) / (n - 1);
xpoints = Table[N[xmin + dx (i - 1)], {i, 1, n}];
PsiIni = Table[N[ψ[xpoints[[i]]]], {i, 1, n}];
GraphicsRow[{ListPlot[
    Transpose[{xpoints, Abs[PsiIni]^2}], PlotRange → All, PlotLabel → "|ψ|^2"],
  ListPlot[{Transpose[{xpoints, Re[PsiIni]}], Transpose[{xpoints, Im[PsiIni]}]},
   PlotRange → All, PlotLabel → "Re(ψ) and Im(ψ)"]}]
```

```
Number of points: 256
```

Out[13]=

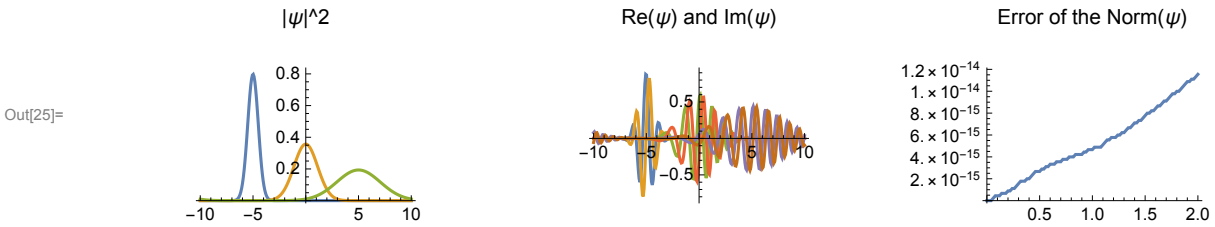


The time evolution up to time t = T using Nt steps:

In[14]:=
```
(* parameters *)
μ = 1;
T = 2; Nt = 100;
dt = N[T / Nt];
(* potential *)
V[x_] := 0;
Vx = Table[N[V[xpoints[[i]]]], {i, 1, n}];
(* arrays for the wave packet evolution and its norm *)
Psi = ConstantArray[0.0, {Nt + 1, Length[PsiIni]}];
normPsi = ConstantArray[0.0, Nt + 1];
(* initialization *)
Psi[[1]] = PsiIni;
normPsi[[1]] = Sqrt[dx] * Norm[PsiIni];
(* evolution *)
Do[
  Psi[[i]] = myOneStepVTV[Psi[[i - 1]], Vx, μ, dx, dt];
  normPsi[[i]] = Sqrt[dx] * Norm[Psi[[i]]],
  {i, 2, Nt + 1}
 ];
```

In[24]:=
```
(* time indices for which to plot wave packet *)
timesToPlot = {1, 51, 101};
GraphicsRow[{ListPlot[Table[Transpose[{xpoints, Abs[Psi[[t]]^2}], {t, timesToPlot}],
    PlotRange → All, Joined → True, PlotLabel → "|ψ|^2"],
  ListPlot[Flatten[Table[{Transpose[{xpoints, Re[Psi[[t]]}],
      Transpose[{xpoints, Im[Psi[[t]]}]}, {t, timesToPlot}], 1],
   PlotRange → All, Joined → True, PlotLabel → "Re(ψ) and Im(ψ)"],
  ListPlot[{Transpose[{Table[dt * i, {i, 0, Nt}], Abs[normPsi - 1]}]},
   PlotRange → All, Joined → True, PlotLabel → "Error of the Norm(ψ)"]}]
```

Out[25]=



Time evolution of the wave packet in the harmonic potential

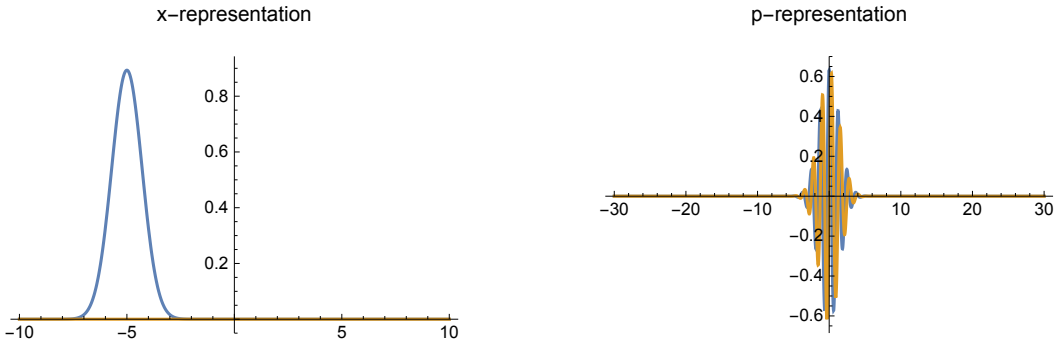
The initial Gaussian wave packet:

In[26]:=
```
σ = 1 / 2; x0 = -5; p0 = 0;
ψ[x_] = Exp[-(x - x0)^2 / (4 σ^2) + i x p0] / (2 π σ^2)^(1 / 4)
ψp[p_] = InverseFourierTransform[ψ[x], x, p]
GraphicsRow[{Plot[{Re[ψ[x]], Im[ψ[x]]},
    {x, -10, 10}, PlotRange → All, PlotLabel → "x-representation"],
  Plot[{Re[ψp[x]], Im[ψp[x]]}, {x, -30, 30}, PlotRange → All,
   PlotLabel → "p-representation"]}]
```

Out[27]=
$$e^{-(5+x)^2}\left(\frac{2}{\pi}\right)^{1/4}$$

Out[28]=
$$\frac{e^{-\frac{1}{4}p\,(-20\,i+p)}}{(2\pi)^{1/4}}$$

Out[29]=

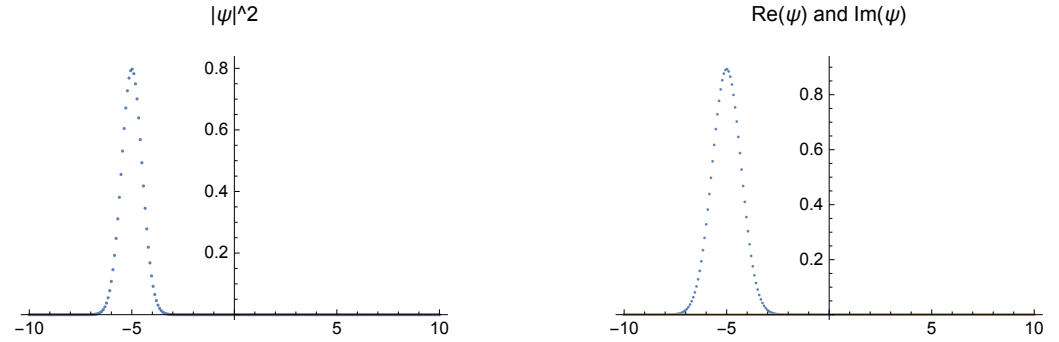


Initial wave packet on the grid:

In[30]:=
```
n = 2^8; Print["Number of points: ", n];
(* equadistatnt grid with the step dx *)
xmin = -10; xmax = 10;
dx = (xmax - xmin) / (n - 1);
xpoints = Table[N[xmin + dx (i - 1)], {i, 1, n}];
PsiIni = Table[N[ψ[xpoints[[i]]]], {i, 1, n}];
GraphicsRow[{ListPlot[
    Transpose[{xpoints, Abs[PsiIni]^2}], PlotRange → All, PlotLabel → "|ψ|^2"],
  ListPlot[{Transpose[{xpoints, Re[PsiIni]}], Transpose[{xpoints, Im[PsiIni]}]},
   PlotRange → All, PlotLabel → "Re(ψ) and Im(ψ)"]}]
```

```
Number of points: 256
```

Out[35]=

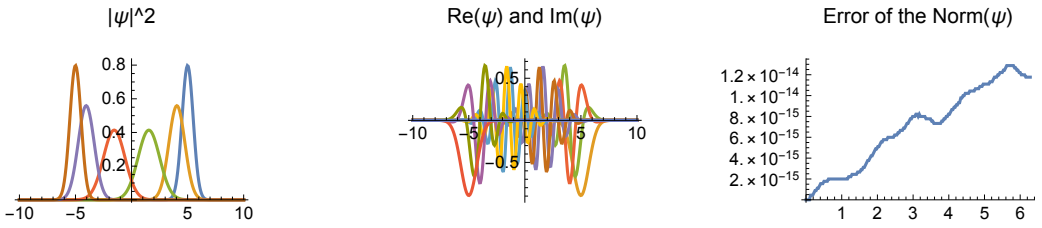


The time evolution up to time t = T using Nt steps:

In[36]:=
```
(* parameters *)
ω = 1; μ = 1;
T = 2 * π; Nt = 200;
dt = N[T / Nt];
(* potential *)
V[x_] := 1 / 2 * μ * (ω * x) ^ 2;
Vx = Table[N[V[xpoints[[i]]]], {i, 1, n}];
(* arrays for the wave packet evolution and its norm *)
Psi = ConstantArray[0.0, {Nt + 1, Length[PsiIni]}];
normPsi = ConstantArray[0.0, Nt + 1];
(* initialization *)
Psi[[1]] = PsiIni;
normPsi[[1]] = Sqrt[dx] * Norm[PsiIni];
(* evolution *)
Do[
  Psi[[i]] = myOneStepVTV[Psi[[i - 1]], Vx, μ, dx, dt];
  normPsi[[i]] = Sqrt[dx] * Norm[Psi[[i]]],
  {i, 2, Nt + 1}
 ];
```

In[46]:=
```
(* time indices for which to plot wave packet *)
timesToPlot = Table[1 + i * 20, {i, 5, 10}];
GraphicsRow[{ListPlot[Table[Transpose[{xpoints, Abs[Psi[[t]]^2}], {t, timesToPlot}],
   PlotRange → All, Joined → True, PlotLabel → "|ψ|^2"],
  ListPlot[Flatten[Table[{Transpose[{xpoints, Re[Psi[[t]]}],
      Transpose[{xpoints, Im[Psi[[t]]}]}, {t, timesToPlot}], 1],
   PlotRange → All, Joined → True, PlotLabel → "Re(ψ) and Im(ψ)"],
  ListPlot[{Transpose[{Table[dt * i, {i, 0, Nt}], Abs[normPsi - 1]}]},
   PlotRange → All, Joined → True, PlotLabel → "Error of the Norm(ψ)"]}]
```

Out[47]=



Scattering of the wave packet off the potential barrier

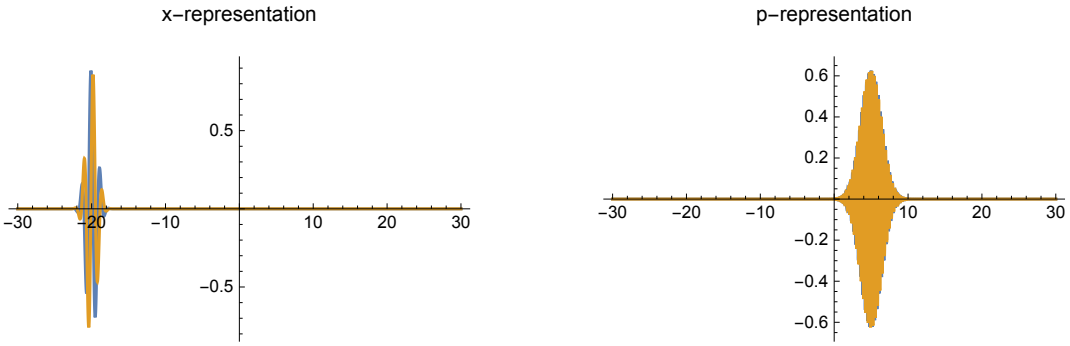
The initial Gaussian wave packet:

In[48]:=
```
σ = 1 / 2; x0 = -20; p0 = 5;
ψ[x_] = Exp[- (x - x0) ^ 2 / (4 σ^2) + i x p0] / (2 π σ^2) ^ (1 / 4)
ψp[p_] = InverseFourierTransform[ψ[x], x, p]
GraphicsRow[{Plot[{Re[ψ[x]], Im[ψ[x]]},
    {x, -30, 30}, PlotRange → All, PlotLabel → "x-representation"],
  Plot[{Re[ψp[x]], Im[ψp[x]]}, {x, -30, 30}, PlotRange → All,
   PlotLabel → "p-representation"]}]
```

Out[49]=
$$e^{5 i x-(20+x)^2} \left(\frac{2}{\pi}\right)^{1/4}$$

Out[50]=
$$\frac{e^{-400-\frac{1}{4}((-5-40 i)+p)^2}}{(2\pi)^{1/4}}$$

Out[51]=



Initial wave packet on the grid:

In[52]:=
```
n = 2^9; Print["Number of points: ", n];
(* equadistatnt grid with the step dx *)
xmin = -30; xmax = 30;
dx = (xmax - xmin) / (n - 1);
xpoints = Table[N[xmin + dx (i - 1)], {i, 1, n}];
PsiIni = Table[N[ψ[xpoints[[i]]]], {i, 1, n}];
GraphicsRow[{ListPlot[
    Transpose[{xpoints, Abs[PsiIni]^2}], PlotRange → All, PlotLabel → "|ψ|^2"],
  ListPlot[{Transpose[{xpoints, Re[PsiIni]}], Transpose[{xpoints, Im[PsiIni]}]},
    PlotRange → All, PlotLabel → "Re(ψ) and Im(ψ)"]}]
```

```
Number of points: 512
```

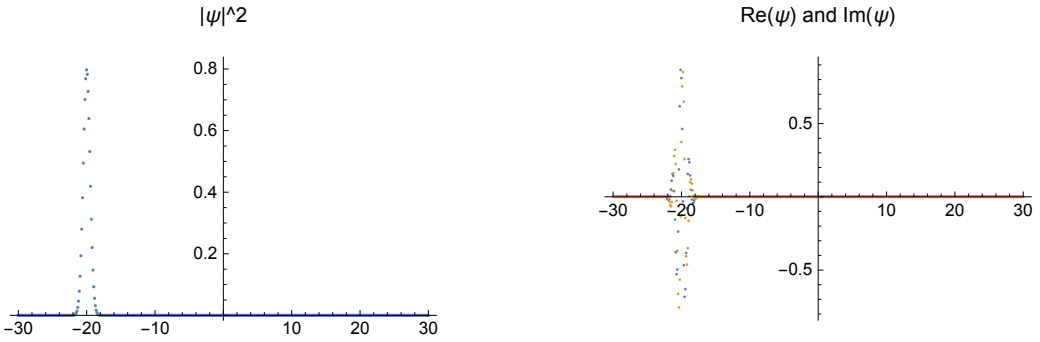
General: Exp[-709.3726663 + 33.1702544 ī] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[-715.6410247 + 33.75733855 ī] is too small to represent as a normalized machine number; precision may be lost.

General: Exp[-721.9369564 + 34.3444227 ī] is too small to represent as a normalized machine number; precision may be lost.

General: Further output of General::munfl will be suppressed during this calculation.

Out[57]=



The time evolution up to time t = T using Nt steps:

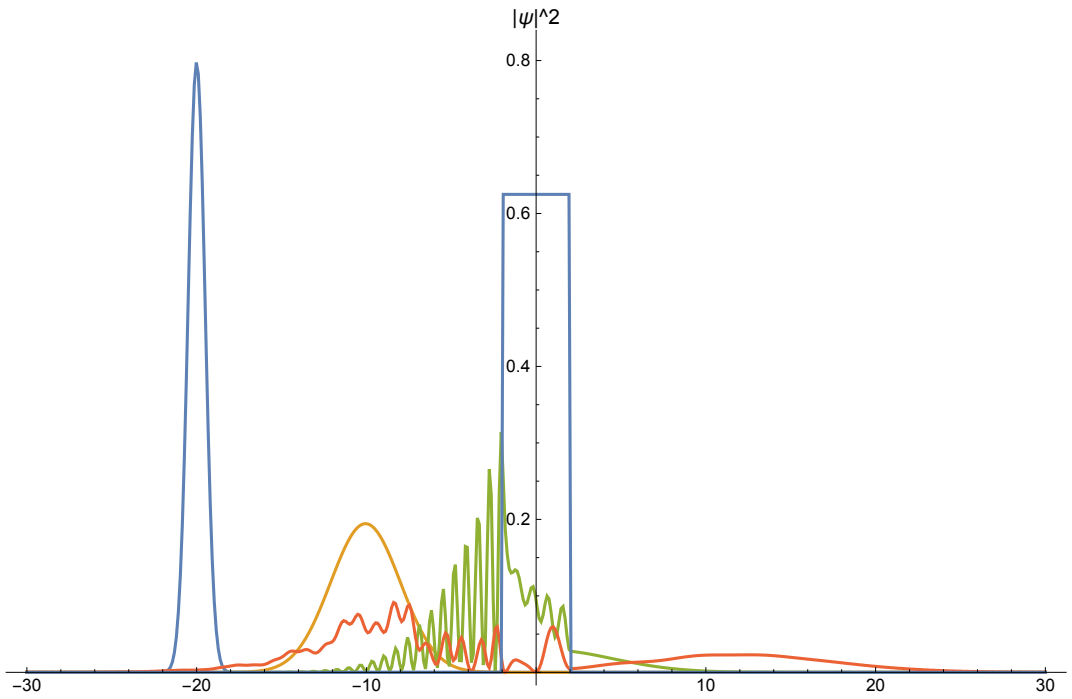
In[58]:=
```
(* parameters *)
ω = 1; μ = 1;
T = 6; Nt = 600;
dt = N[T / Nt];
(* potential *)
a = 2; V0 = 12.5; (* width and height of the barrier *)
V[x_] := If[Abs[x] ≤ a, V0, 0];
Vx = Table[N[V[xpoints[[i]]]], {i, 1, n}];
(* arrays for the wave packet evolution and its norm *)
Psi = ConstantArray[0.0, {Nt + 1, Length[PsiIni]}];
normPsi = ConstantArray[0.0, Nt + 1];
(* initialization *)
Psi[[1]] = PsiIni;
normPsi[[1]] = Sqrt[dx] * Norm[PsiIni];
(* evolution *)
Do[
  Psi[[i]] = myOneStepVTV[Psi[[i - 1]], Vx, μ, dx, dt];
  normPsi[[i]] = Sqrt[dx] * Norm[Psi[[i]]],
  {i, 2, Nt + 1}
 ];
```

In[69]:=
```
(* time indices for which to plot wave packet *)
timesToPlot = {1, 200, 400, 600};
Print["Barrier height V0 = ", N[V0]];
Print["Mean energy of the incoming particle E = ", N[p0^2 / (2 μ)]];
Show[{ListPlot[Table[Transpose[{xpoints, Abs[Psi[[t]]]^2}], {t, timesToPlot}],
    PlotRange → All, Joined → True, PlotLabel → "|ψ|^2"],
  ListPlot[Transpose[{xpoints, Vx / 20}], PlotRange → All, Joined → True]}]
ListPlot[{Transpose[{Table[dt * i, {i, 0, Nt}], Abs[normPsi - 1]}]},
 PlotRange → All, Joined → True, PlotLabel → "Error of the Norm(ψ)"]
```

```
Barrier height V0 = 12.5

Mean energy of the incoming particle E = 12.5
```

Out[72]=



Out[73]=