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

Problem

Solve numerically the differential equation (in atomic units $\hbar = 1$, $m_e = 1$)

$$i \frac{\partial \psi(x, t)}{\partial t} = -\frac{1}{2\mu} \frac{\partial^2 \psi(x, t)}{\partial x^2} + V(x)\, \psi(x, t) \tag{1}$$

with the following initial condition

$$\psi(x, 0) = \frac{1}{(2\pi\sigma^2)^{1/4}} e^{-(x-x_0)^2/(4\sigma^2) + i p x} \tag{2}$$

and Dirichlet boundary conditions

$$\begin{aligned} u(-\infty, t) &= 0 \\ u(+\infty, t) &= 0 \end{aligned} \tag{3}$$

Exact solution for V(x) = 0

Initial normalized Gaussian packet :

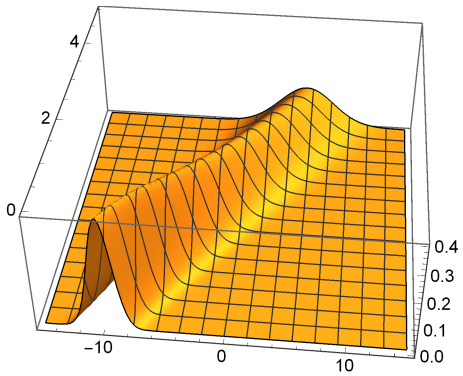
In[2]:=
```
u0[x_, x0_, σ_, p_] = 1 / (2 π σ^2) ^ (1 / 4) Exp[(- (x - x0) ^2 / (4 σ^2) + ⅈ p x)];
Assuming[σ > 0, Integrate[u0[x, x0, σ, p] * u0[x, x0, σ, -p], {x, -Infinity, Infinity}]]
```

Out[3]= 1

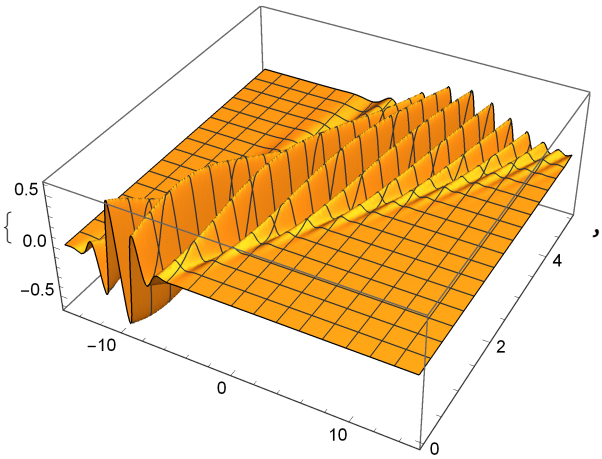
In[4]:=
```
sol = DSolve[
  {ⅈ D[u[x, t], t] == -D[u[x, t], x, x] / (2 μ), u[x, 0] == u0[x, x0, σ, p]}, u, {x, t}]
pini = 3; mass = 1; xini = -10; σini = 1;
xmin = -15;  xmax = 15;  tmax = 5;
Plot3D[Abs[u[x, t] /. sol /. {σ → σini, x0 → xini, p → pini, μ → mass}]^2,
 {x, xmin, xmax}, {t, 0, tmax}, PlotRange → All, PlotPoints → 100]
{Plot3D[Re[u[x, t] /. sol /. {σ → σini, x0 → xini, p → pini, μ → mass}],
  {x, xmin, xmax}, {t, 0, tmax}, PlotRange → All, PlotPoints → 100],
 Plot3D[Im[u[x, t] /. sol /. {σ → σini, x0 → xini, p → pini, μ → mass}],
  {x, xmin, xmax}, {t, 0, tmax}, PlotRange → All, PlotPoints → 100]}
```

Out[4]=
$$\left\{\left\{u \to \text{Function}\left[\{x, t\}, \left(e^{\frac{-(x-x0)^2 \mu - 2 i p^2 t \sigma^2 + p (-2 t x0 + 4 i x \mu \sigma^2)}{2 i t + 4 \mu \sigma^2}} \left(\frac{2}{\pi}\right)^{1/4} \sqrt{\frac{1}{\sigma^2}} \left(\sigma^2\right)^{3/4}\right) \Big/ \left(\sqrt{\frac{i t}{\mu} + 2 \sigma^2}\right)\right]\right\}\right\}$$

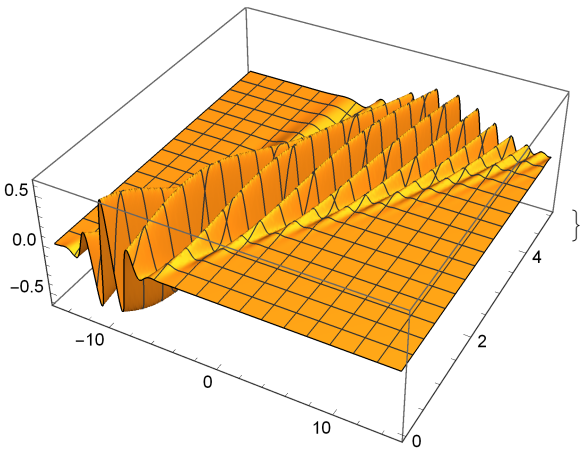
Out[7]=



Out[8]= {



,



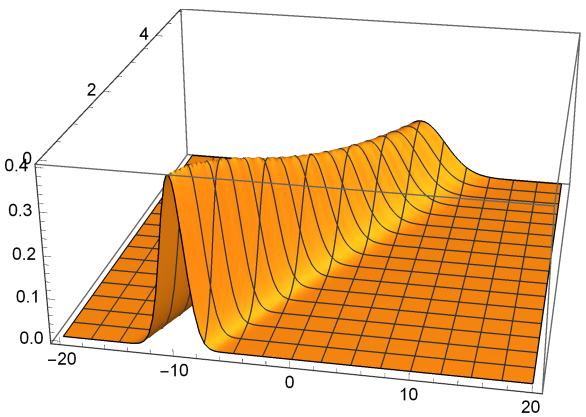
}

Numerical solution using built-in function

In[9]:=
```
{xmin, xmax} = {-20, 20};
{tmin, tmax} = {0, 5};
numsol = NDSolve[
  {ⅈ D[u[x, t], t] == -D[u[x, t], x, x] / (2 mass), u[x, tmin] == u0[x, xini, σini, pini],
   u[xmin, t] == 0, u[xmax, t] == 0}, u, {x, xmin, xmax}, {t, tmin, tmax}]
Plot3D[Evaluate[Abs[u[x, t] /. numsol]^2], {x, xmin, xmax},
 {t, tmin, tmax}, PlotRange → All, PlotPoints → 100]
```

Out[11]= {{u → InterpolatingFunction[Domain: {{-20., 20.}, {0., 5.}} Output: scalar]}}

Out[12]=

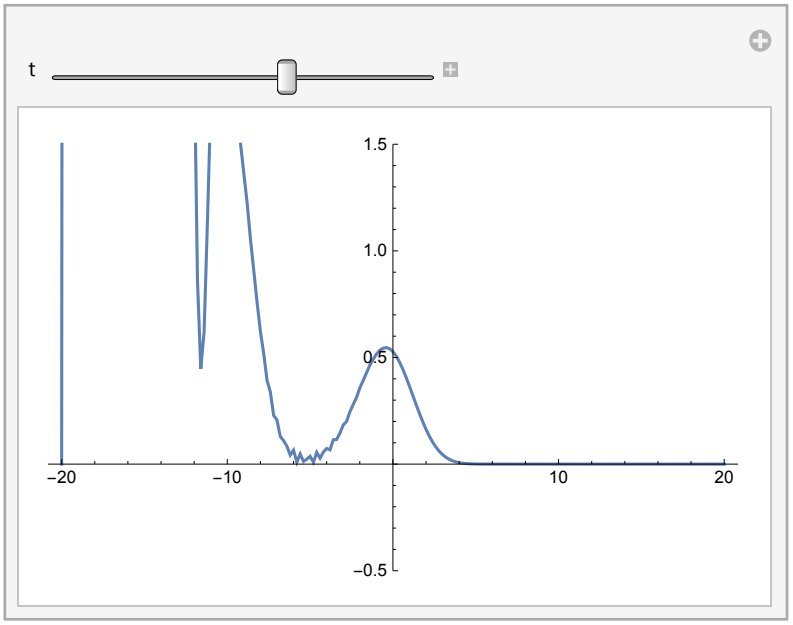


Numerical solution using Euler explicit method, unstable

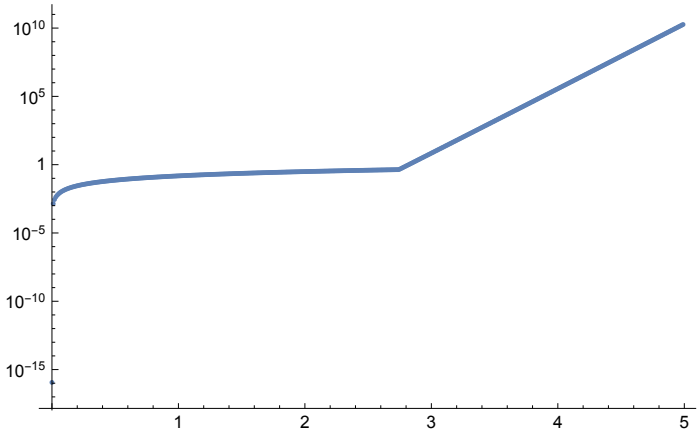
In[13]:=
```
(* grid initialization *)
{nx, nt} = {201, 501};
dx = (xmax - xmin) / (nx - 1);
dt = (tmax - tmin) / (nt - 1);
Print["λ/2m = ", λ = N[dt / dx^2 / (2 mass)]]
X = N[Range[xmin, xmax, dx]];
T = N[Range[tmin, tmax, dt]];

(* Initialization of the array with zeroes - Dirichlet's boundary conditions *)
v = ConstantArray[0.0, {nx, nt}];
error = ConstantArray[0.0, nt];

(* Initial state *)
Do[v[[i, 1]] = u0[X[[i]], xini, σini, pini], {i, 2, nx - 1}];

Print["Euler explicit method - order 1, unstable"];
Do[v[[j, n + 1]] = v[[j, n]] + λ ⅈ (v[[j + 1, n]] - 2 v[[j, n]] + v[[j - 1, n]]),
 {n, 1, nt - 1}, {j, 2, nx - 1}]

Do[
  error[[n]] = 0.0;
  Do[
   error[[n]] = Max[error[[n]], Abs[v[[j, n]] - u[x, t] /. sol /.
        {σ → σini, x0 → xini, p → pini, μ → mass, x → X[[j]], t → T[[n]]}]],
   {j, 2, nx - 1}
  ],
  {n, 1, nt - 1}
 ];

(* fancy plotting *)
Manipulate[
 n = Round[t / dt] + 1;
 ListLinePlot[Table[{X[[j]], Abs[v[[j, n]]^2]}, {j, 1, nx}],
  PlotRange → {-0.5, 1.5}],
 {t, tmin, tmax, dt}]

(* show maximal errors *)
ListLogPlot[{Table[{T[[n]], error[[n]]}, {n, 1, nt}]},
 PlotRange → All]
Print["Maximal error: ", Max[error]]
```

```
λ/2m = 0.125

Euler explicit method - order 1, unstable
```

Out[25]=



Out[26]=



Maximal error: $1.825659303 \times 10^{10}$

In[28]:=