

```
In[289]:= Clear["Global`*"];
```

Problem

Solve numerically the differential equation

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2} \quad (1)$$

with the following initial condition

$$u(x, 0) = e^{-x^2} \quad (2)$$

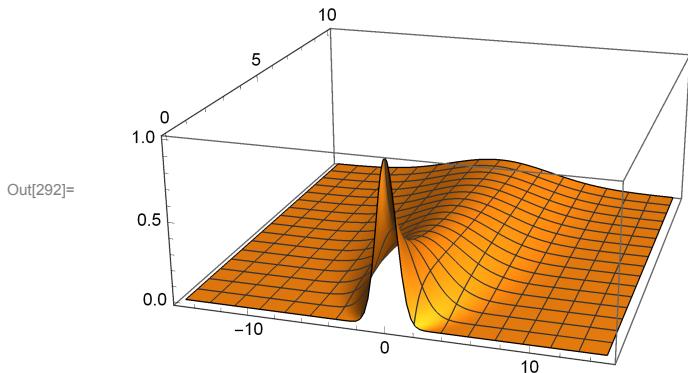
and Dirichlet boundary conditions

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

Exact solution

```
In[290]:= u0[x_, t_] = Exp[-x^2 / (1 + 4 t)] / Sqrt[1 + 4 t];  
sol = DSolve[{D[u[x, t], t] == D[u[x, t], x, x], u[x, 0] == u0[x, 0]}, u, {x, t}]  
Plot3D[u[x, t] /. sol, {x, -15, 15}, {t, 0, 10}, PlotRange -> All, PlotPoints -> 100]
```

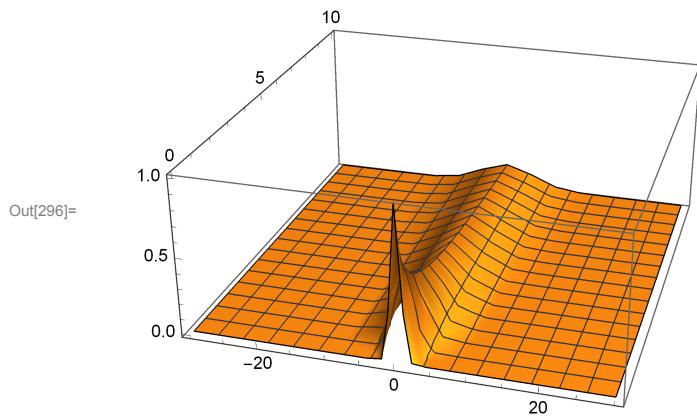
```
Out[291]= \{ \{ u \rightarrow Function[\{x, t\}, \frac{e^{-\frac{x^2}{1+4 t}}}{\sqrt{1+4 t}}] \} \}
```



Numerical solution using built-in function

```
In[293]:= {xmin, xmax} = {-30, 30};  
{tmin, tmax} = {0, 10};  
numsol =  
  NDSolve[{D[u[x, t], t] == D[u[x, t], x, x], u[x, 0] == u0[x, 0], u[xmin, t] == u0[xmin, 0],  
    u[xmax, t] == u0[xmax, 0]}, u, {x, xmin, xmax}, {t, tmin, tmax}]  
Plot3D[Evaluate[u[x, t] /. numsol], {x, xmin, xmax}, {t, tmin, tmax}, PlotRange -> All]
```

Out[295]= $\left\{ \begin{array}{l} \text{u} \rightarrow \text{InterpolatingFunction} \left[\begin{array}{c} \text{+} \quad \text{N} \\ \text{Domain: } \{\{-30., 30.\}, \{0., 10.\}\} \\ \text{Output: scalar} \end{array} \right] \end{array} \right\}$



Numerical solution using basic explicit methods

```
In[378]:= (* grid initialization *)
{nx, nt} = {201, 301};
dx = (xmax - xmin) / (nx - 1);
dt = (tmax - tmin) / (nt - 1);
Print["λ = ", λ = N[dt/dx^2]];
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];

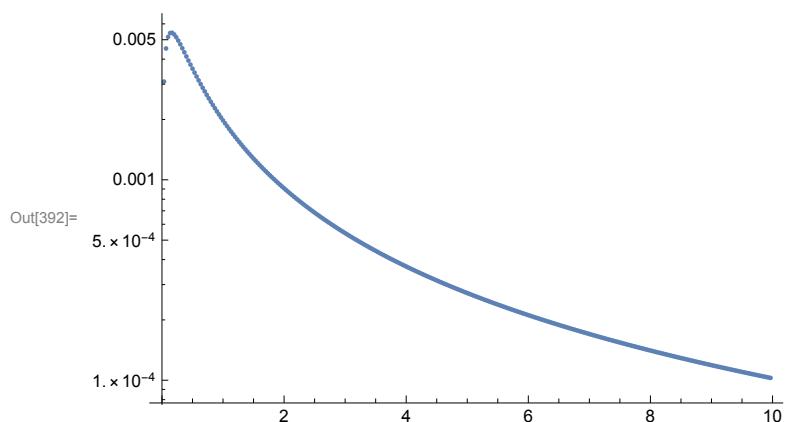
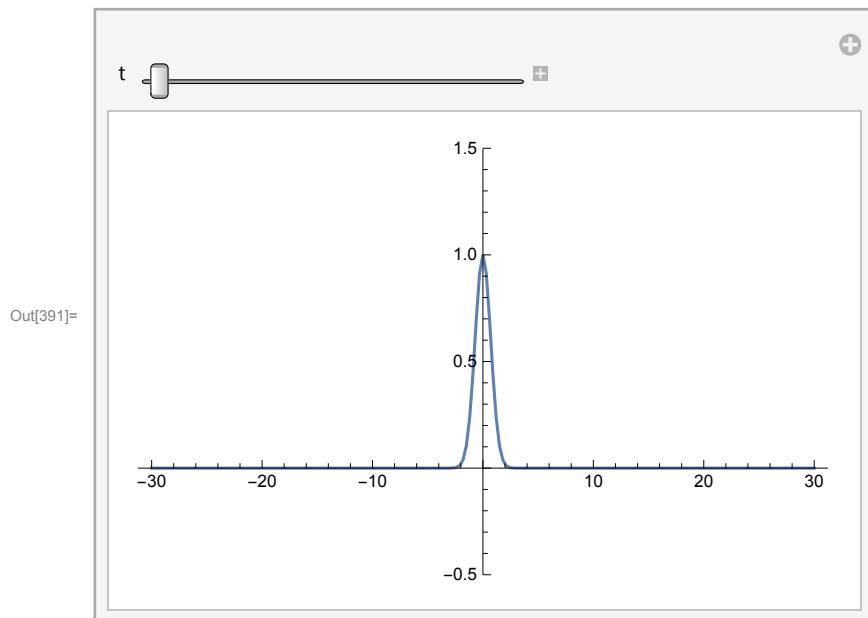
(* Initial state *)
Do[v[[i, 1]] = u0[X[[i]], 0], {i, 2, nx - 1}];
Do[v[[i, 2]] = u0[X[[i]], 0], {i, 2, nx - 1}];

method = 1;
Which[
  method == 1,
  Print["Euler explicit method - order 1, stable for λ = k/h^2 < 1/2"];
  Do[v[[j, n + 1]] = v[[j, n]] + λ (v[[j + 1, n]] - 2 v[[j, n]] + v[[j - 1, n]]),
    {n, 1, nt - 1}, {j, 2, nx - 1}],
  (* Leap-frog method - unstable *)
  method == 2,
  Print["Leap-frog method - unstable"];
  Do[v[[j, n + 1]] = v[[j, n - 1]] + 2 λ (v[[j + 1, n]] - 2 v[[j, n]] + v[[j - 1, n]]),
    {n, 2, nt - 1}, {j, 2, nx - 1}]
];
Do[
  error[[n]] = 0.0;
  Do[
    error[[n]] = Max[error[[n]], Abs[v[[j, n]] - u0[X[[j]], T[[n]]]]],
    {j, 2, nx - 1}
  ],
  {n, 1, nt - 1}
];

(* fancy plotting *)
Manipulate[
  n = Round[t/dt] + 1;
  ListLinePlot[Table[{X[[j]], v[[j, n]]}, {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 = 0.3703703704
Euler explicit method - order 1, stable for λ = k/h^2 < 1/2
```



Maximal error: 0.005415445012

In[313]:=