```
Clear["Global`*"];
```

Problem

Solve numerically the differential equation

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

with the following initial condition

$$u(x,y,0) = e^{-(x^2+y^2)} \tag{2}$$

and Dirichlet boundary conditions

$u(x,y,t) = 0$ for at least one of x and y going to infinity. (3)

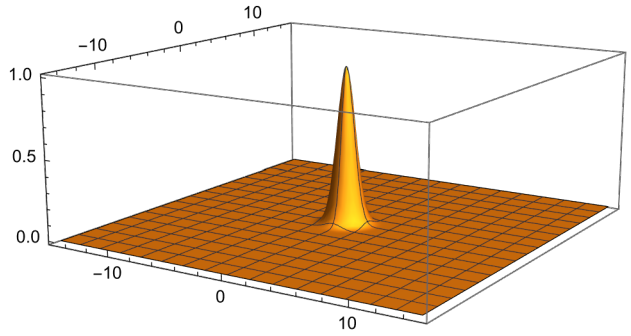
Exact solution

In[1]:=
```
u0[x_, y_, t_] = Exp[-(x^2 + y^2) / (1 + 4 t)] / (1 + 4 t);
sol = DSolve[{D[u[x, y, t], t] == D[u[x, y, t], x, x] + D[u[x, y, t], y, y],
    u[x, y, 0] == u0[x, y, 0]}, u, {x, y, t}]
```

Out[2]=
$$\left\{\left\{u \to \text{Function}\left[\{x, y, t\}, \frac{e^{-\frac{x^2+y^2}{1+4t}}}{1+4t} \text{ if } \text{Re}[t] > -\frac{1}{4}\right]\right\}\right\}$$

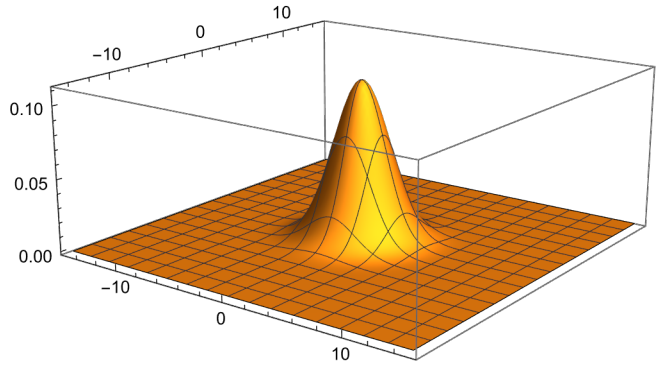
In[3]:=
```
Plot3D[u0[x, y, 0], {x, -15, 15}, {y, -15, 15}, PlotRange → All, PlotPoints → 100]
```

Out[3]=



In[4]:=
```
Plot3D[u0[x, y, 2], {x, -15, 15}, {y, -15, 15}, PlotRange → All, PlotPoints → 100]
```

Out[4]=



Numerical solution using built-in function

In[5]:=
```
{xmin, xmax} = {-15, 15};
{ymin, ymax} = {-15, 15};
{tmin, tmax} = {0, 10};
numsol = NDSolve[{D[u[x, y, t], t] == D[u[x, y, t], x, x] + D[u[x, y, t], y, y],
    u[x, y, 0] == u0[x, y, 0], u[xmin, y, t] == u0[xmin, y, 0], u[xmax, y, t] == u0[xmax, y, 0],
    u[x, ymin, t] == u0[x, ymin, 0], u[x, ymax, t] == u0[x, ymax, 0]},
  u, {x, xmin, xmax}, {y, ymin, ymax}, {t, tmin, tmax}]
```

NDSolve: Using maximum number of grid points 100 allowed by the MaxPoints or MinStepSize options for independent variable x.

NDSolve: Using maximum number of grid points 100 allowed by the MaxPoints or MinStepSize options for independent variable y.

NDSolve: Using maximum number of grid points 100 allowed by the MaxPoints or MinStepSize options for independent variable x.

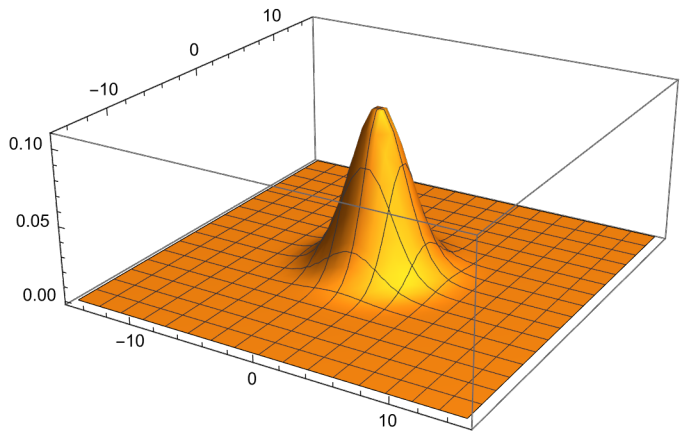
General: Further output of NDSolve::mxsst will be suppressed during this calculation.

Out[8]=
```
{{u → InterpolatingFunction[ ]}}
```
Domain: {{−15., 15.}, {−15., 15.}, {0., 10.}}
Output: scalar

Data not in notebook. Store now

In[9]:=
```
Plot3D[Evaluate[(u[x, y, t] /. numsol) /. {t → 2}],
 {x, xmin, xmax}, {y, ymin, ymax}, PlotRange → All, PlotPoints → 40]
```

Out[9]=



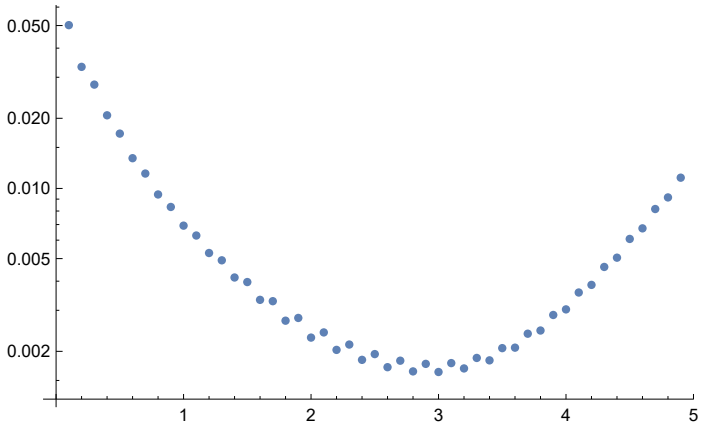
Numerical solution using basic explicit methods

In[10]:=
```
(* grid initialization *)
{xmin, xmax} = {-15, 15};
{ymin, ymax} = {-15, 15};
{tmin, tmax} = {0, 5};
{nx, ny, nt} = {51, 51, 51};
dx = (xmax - xmin) / (nx - 1);
dy = (ymax - ymin) / (ny - 1);
dt = (tmax - tmin) / (nt - 1);
Print["λ = ", λ = N[dt / dx^2]]
X = N[Range[xmin, xmax, dx]];
Y = N[Range[ymin, ymax, dy]];
T = N[Range[tmin, tmax, dt]];

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

(* Initial state *)
Do[v[[i, j, 1]] = u0[X[[i]], Y[[j]], tmin], {j, 2, ny - 1}, {i, 2, nx - 1}];
Do[v[[i, j, 2]] = u0[X[[i]], Y[[j]], tmin + dt], {j, 2, ny - 1}, {i, 2, nx - 1}];

method = 1;
Which[
  method == 1,
  Print["Euler explicit method - order 1, stable for λ = k/h^2 < 1/4"];
  Do[v[[i, j, n + 1]] =
    v[[i, j, n]] + λ (v[[i + 1, j, n]] + v[[i, j + 1, n]] - 4 v[[i, j, n]] + v[[i - 1, j, n]] + v[[i, j - 1, n]]),
   {n, 1, nt - 1}, {j, 2, ny - 1}, {i, 2, nx - 1}]
 ];
Do[
  error[[n]] = 0.0;
  Do[
   error[[n]] = Max[error[[n]], Abs[v[[i, j, n]] - u0[X[[i]], Y[[j]], T[[n]]]]],
   {j, 2, ny - 1}, {i, 2, nx - 1}
  ],
  {n, 1, nt - 1}
 ];
```

```
λ = 0.277777777778

Euler explicit method - order 1, stable for λ = k/h^2 < 1/4
```

In[28]:=
```
(* show maximal errors *)
ListLogPlot[{Table[{T[[n]], error[[n]]}, {n, 1, nt}]},
 PlotRange → All]
Print["Maximal error: ", Max[error]]
```

Out[28]=




```
Maximal error: 0.0502009075401
```

In[30]:=
```
ListPlot3D[
 Flatten[Table[{X[[i]], Y[[j]], v[[i, j, 50]]}, {j, 1, ny}, {i, 1, nx}], 1], PlotRange → All]
```

Out[30]=