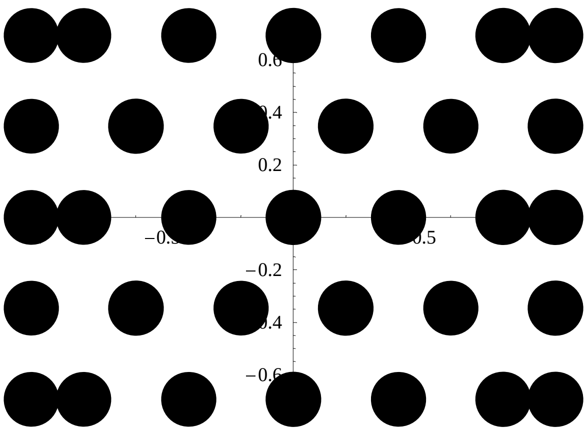
■ Here is some code for a nice "Honey Comb" mesh composed of equilateral triangles:

Calculate the x, y coordinates of the nodes :

In[1]:=
```
nx = 7; ny = 5; dx = 1.0 / (nx − 2);
nodes = Flatten[Table[Table[Chop[{Min[1, Max[−1, x]], dx (j − 0.5 (ny + 1)) / Tan[π / 6]}],
        {x, −(1 + dx Mod[j, 2]), 1 + dx, 2 dx}], {j, 1, ny}], 1]
```

Out[2]= {{−1, −0.69282}, {−0.8, −0.69282}, {−0.4, −0.69282}, {0, −0.69282}, {0.4, −0.69282}, {0.8, −0.69282}, {1, −0.69282}, {−1, −0.34641}, {−0.6, −0.34641}, {−0.2, −0.34641}, {0.2, −0.34641}, {0.6, −0.34641}, {1., −0.34641}, {−1, 0}, {−0.8, 0}, {−0.4, 0}, {0, 0}, {0.4, 0}, {0.8, 0}, {1, 0}, {−1, 0.34641}, {−0.6, 0.34641}, {−0.2, 0.34641}, {0.2, 0.34641}, {0.6, 0.34641}, {1., 0.34641}, {−1, 0.69282}, {−0.8, 0.69282}, {−0.4, 0.69282}, {0, 0.69282}, {0.4, 0.69282}, {0.8, 0.69282}, {1, 0.69282}}

```
$DefaultFont = {"Times−Roman", 9};
ListPlot[nodes, PlotStyle → {PointSize[dx / 2]}, PlotRange → All, AspectRatio → Automatic];
```



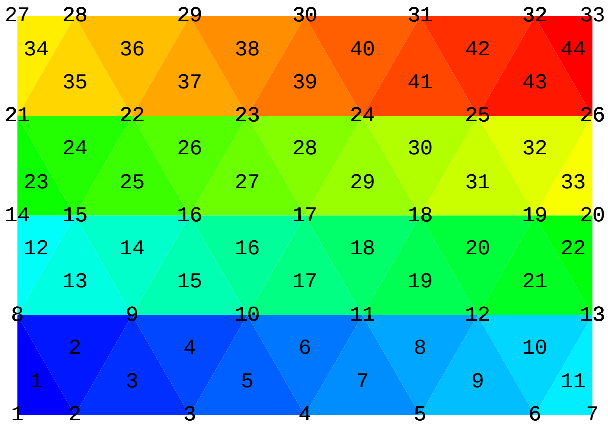
Calculate the indices of nodes of the triangle elements :

In[3]:=
```
elements = Flatten[Table[Table[i + (j − 1) nx − Floor[j / 2] + Join[If[Mod[j, 2] == 1 || i > 1, {{0, 1, nx}}, {}],
              If[Mod[j, 2] == 0 || i < nx − 1, {{1, nx + 1, nx}}, {}]], {i, 1, nx − 1}], {j, 1, ny − 1}], 2]
```

Out[3]= {{1, 2, 8}, {2, 9, 8}, {2, 3, 9}, {3, 10, 9}, {3, 4, 10}, {4, 11, 10}, {4, 5, 11}, {5, 12, 11}, {5, 6, 12}, {6, 13, 12}, {6, 7, 13}, {8, 15, 14}, {8, 9, 15}, {9, 16, 15}, {9, 10, 16}, {10, 17, 16}, {10, 11, 17}, {11, 18, 17}, {11, 12, 18}, {12, 19, 18}, {12, 13, 19}, {13, 20, 19}, {14, 15, 21}, {15, 22, 21}, {15, 16, 22}, {16, 23, 22}, {16, 17, 23}, {17, 24, 23}, {17, 18, 24}, {18, 25, 24}, {18, 19, 25}, {19, 26, 25}, {19, 20, 26}, {21, 28, 27}, {21, 22, 28}, {22, 29, 28}, {22, 23, 29}, {23, 30, 29}, {23, 24, 30}, {24, 31, 30}, {24, 25, 31}, {25, 32, 31}, {25, 26, 32}, {26, 33, 32}}

Here is a picture to visualize the triangle elements :

In[4]:=
```
jmax = Length[elements]; PlotColor[hue_] := Hue[2 (1 − hue) / 3];
Show[Graphics[Table[plist = Map[nodes[[#]] &, elements[[j]]];
        {PlotColor[(j − 1) / (jmax − 1)], Polygon[plist], RGBColor[0, 0, 0], Text[j, Plus @@ plist / 3],
          Table[Text[elements[[j, i]], plist[[i]], {i, 1, 3}]}, {j, 1, jmax, 1}]], AspectRatio → Automatic, PlotRange → All];
```



If you want to use this mesh in Fortran or some other program, you can use this code to export the mesh :

In[6]:=
```
Export["C:/nodes.txt", Chop[nodes], "Table"]; Export["C:/elements.txt", Chop[elements], "Table"];
```

Calculate the local stiffness matrices : $K^j_{mn} = \frac{1}{\Delta x_2\,\Delta y_3 - \Delta x_3\,\Delta y_2}\begin{pmatrix} \lambda_x + \lambda_y - 2\,\lambda_{xy} & \lambda_{xy} - \lambda_y & \lambda_{xy} - \lambda_x \\ \lambda_{xy} - \lambda_y & \lambda_y & -\lambda_{xy} \\ \lambda_{xy} - \lambda_x & -\lambda_{xy} & \lambda_x \end{pmatrix}$

$\Delta x_2 \equiv x_2 - x_1,\ \Delta y_2 \equiv y_2 - y_1,\ \Delta x_3 \equiv x_3 - x_1,\ \Delta y_3 \equiv y_3 - y_1,\ \lambda_x \equiv \Delta x_2^2 + \Delta x_3^2,\ \lambda_{xy} \equiv \Delta x_2\,\Delta y_2 + \Delta x_3\,\Delta y_3,\ \lambda_y \equiv \Delta y_2^2 + \Delta y_3^2$

In[7]:=
```
Klocal =
  Map[Module[{}, Do[{Δx_i, Δy_i} = nodes[[#[[i]]]] − nodes[[#[[1]]]], {i, 2, 3}]; λx = Δx2^2 + Δx3^2; λxy = Δx2 Δy2 + Δx3 Δy3;
      λy = Δy2^2 + Δy3^2; Chop[1/(Δx2 Δy3 − Δx3 Δy2) {{λx + λy − 2 λxy, λxy − λy, λxy − λx}, {λxy − λy, λy, −λxy}, {λxy − λx, −λxy, λx}}]] &, elements];

Map[
  MatrixForm,
  Klocal]
```

General::spell1 : Possible spelling error: new symbol name "Δy" is similar to existing symbol "Δx". More…

Out[8]=

$$\left\{\begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},\right.$$

$$\begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix},$$

$$\begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},$$

$$\begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 0.886751 & -0.732051 & -0.154701 \\ -0.732051 & 1.73205 & -1. \\ -0.154701 & -1. & 1.1547 \end{pmatrix},$$

$$\begin{pmatrix} 2.04145 & -2.4641 & 0.42265 \\ -2.4641 & 3.4641 & -1. \\ 0.42265 & -1. & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},$$

$$\begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix},$$

$$\begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},$$

$$\begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 6.04145 & -4.4641 & -1.57735 \\ -4.4641 & 3.4641 & 1. \\ -1.57735 & 1. & 0.57735 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},$$

$$\begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},$$

$$\begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix},$$

$$\begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},$$

$$\begin{pmatrix} 0.886751 & -0.732051 & -0.154701 \\ -0.732051 & 1.73205 & -1. \\ -0.154701 & -1. & 1.1547 \end{pmatrix}, \begin{pmatrix} 2.04145 & -2.4641 & 0.42265 \\ -2.4641 & 3.4641 & -1. \\ 0.42265 & -1. & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix},$$

$$\begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix},$$

$$\begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix},$$

$$\begin{pmatrix} 2.3094 & -1.73205 & -0.57735 \\ -1.73205 & 1.73205 & 0 \\ -0.57735 & 0 & 0.57735 \end{pmatrix}, \begin{pmatrix} 1.3094 & -0.366025 & -0.943376 \\ -0.366025 & 0.866025 & -0.5 \\ -0.943376 & -0.5 & 1.44338 \end{pmatrix}, \begin{pmatrix} 6.04145 & -4.4641 & -1.57735 \\ -4.4641 & 3.4641 & 1. \\ -1.57735 & 1. & 0.57735 \end{pmatrix}\}$$

Assemble global stiffness matrix :

In[9]:=
```
nxy = Length[nodes]; Kglobal = Table[0, {nxy}, {nxy}];
Do[element = elements⟦j⟧;
  Do[Kglobal⟦element⟦m⟧, element⟦n⟧⟧ += Klocal⟦j, m, n⟧, {m, 1, 3}, {n, 1, 3}], {j, 1, jmax}];
Kglobal // MatrixForm
```

General::spell : Possible spelling error: new symbol name "element" is similar to existing symbols {Element, elements}. More…

Out[11]//MatrixForm=

2.3094	−1.73205	0	0	0	0	0	−0.57735	0	0
−1.73205	5.35085	−0.366025	0	0	0	0	−0.57735	−2.67543	0
0	−0.366025	4.48483	−0.366025	0	0	0	0	−1.07735	−2.675
0	0	−0.366025	4.48483	−0.366025	0	0	0	0	−1.077
0	0	0	−0.366025	4.48483	−0.366025	0	0	0	0
0	0	0	0	−0.366025	4.06218	−0.732051	0	0	0
0	0	0	0	0	−0.732051	1.73205	0	0	0
−0.57735	−0.57735	0	0	0	0	0	4.50555	−0.366025	0
0	−2.67543	−1.07735	0	0	0	0	−0.366025	8.2376	−0.366
0	0	−2.67543	−1.07735	0	0	0	0	−0.366025	8.2376
0	0	0	−2.67543	−1.07735	0	0	0	0	−0.366
0	0	0	0	−2.67543	−1.07735	0	0	0	0
0	0	0	0	0	−1.88675	−1.	0	0	0
0	0	0	0	0	0	0	0.42265	0	0
0	0	0	0	0	0	0	−3.40748	−1.07735	0
0	0	0	0	0	0	0	0	−2.67543	−1.077
0	0	0	0	0	0	0	0	0	−2.675
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Calculate the load vector :

In[12]:=
```
Q = Table[1, {nxy}];
```

Modify the global stiffness matrix and load vector to include boundary conditions :

In[13]:=
```
Do[{x, y} = nodes[[i]]; If[Abs[x] == 1 || Abs[y] == 0.5 dx (ny − 1) / Tan[π / 6],
    Kglobal[[i]] = Table[If[j == i, 1, 0], {j, 1, nxy}]; Q[[i]] = 0], {i, 1, nxy}];
Kglobal //
  MatrixForm
```

Out[14]//MatrixForm=

1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	−2.67543	−1.07735	0	0	0	0	−0.366025	8.2376	−0.366025	0	0
0	0	−2.67543	−1.07735	0	0	0	0	−0.366025	8.2376	−0.366025	0
0	0	0	−2.67543	−1.07735	0	0	0	0	−0.366025	8.2376	−0.36
0	0	0	0	−2.67543	−1.07735	0	0	0	0	−0.366025	8.237
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	−3.40748	−1.07735	0	0	0
0	0	0	0	0	0	0	0	−2.67543	−1.07735	0	0
0	0	0	0	0	0	0	0	0	−2.67543	−1.07735	0
0	0	0	0	0	0	0	0	0	0	−2.67543	−1.07
0	0	0	0	0	0	0	0	0	0	0	−2.67
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Solve for Prandtl stress function :

In[15]:=
```
q = Chop[LinearSolve[Kglobal, Q]]
```

Out[15]=
```
{0, 0, 0, 0, 0, 0, 0, 0, 0.312802, 0.374756, 0.370526, 0.296165, 0, 0, 0.203826, 0.455993,
  0.502991, 0.472738, 0.297062, 0, 0, 0.263393, 0.36375, 0.377434, 0.330553, 0, 0, 0, 0, 0, 0, 0}
```

Here is the same code again, this time using a finer mesh :

In[16]:=
```
nx = 50; ny = Round[2 (nx - 2) Tan[π / 6] + 1]; dx = 1.0 / (nx - 2);
nodes = Flatten[Table[Table[{Min[1, Max[-1, x]], dx (j - 0.5 (ny + 1)) / Tan[π / 6]},
      {x, -(1 + dx Mod[j, 2]), 1 + dx, 2 dx}], {j, 1, ny}], 1];
elements = Flatten[Table[Table[i + (j - 1) nx - Floor[j / 2] +
      Join[If[Mod[j, 2] == 1 || i > 1, {{0, 1, nx}}, {}], If[Mod[j, 2] == 0 || i < nx - 1, {{1, nx + 1, nx}}, {}]],
    {i, 1, nx - 1}], {j, 1, ny - 1}], 2]; jmax = Length[elements];
```

Klocal = Map[Module[{}, Do[{Δx_i, Δy_i} = nodes[[#[[i]]]] − nodes[[#[[1]]]], {i, 2, 3}]; $\lambda_x = \Delta x_2^2 + \Delta x_3^2$;

$\lambda_{xy} = \Delta x_2\,\Delta y_2 + \Delta x_3\,\Delta y_3$; $\lambda_y = \Delta y_2^2 + \Delta y_3^2$; $\frac{1}{\Delta x_2\,\Delta y_3 - \Delta x_3\,\Delta y_2}\begin{pmatrix} \lambda_x + \lambda_y - 2\lambda_{xy} & \lambda_{xy} - \lambda_y & \lambda_{xy} - \lambda_x \\ \lambda_{xy} - \lambda_y & \lambda_y & -\lambda_{xy} \\ \lambda_{xy} - \lambda_x & -\lambda_{xy} & \lambda_x \end{pmatrix}$] &, elements];

```
nxy = Length[nodes]; Kglobal = Table[0, {nxy}, {nxy}];
Do[element = elements[[j]]; Do[Kglobal[[element[[m]], element[[n]]]] += Klocal[[j, m, n]], {m, 1, 3}, {n, 1, 3}], {j, 1, jmax}];
Q = Table[1, {nxy}];
Do[{x, y} = nodes[[i]]; If[Abs[x] == 1 || Abs[y] == 0.5 dx (ny - 1) / Tan[π / 6],
    Kglobal[[i]] = Table[If[j == i, 1, 0], {j, 1, nxy}]; Q[[i]] = 0], {i, 1, nxy}];
q = LinearSolve[Kglobal, Q]; q1 = Min[q]; q2 = Max[q]; PlotColor[hue_] := Hue[2 (1 - hue) / 3];
Show[Graphics[{PointSize[1.5 dx], Table[{PlotColor[(q[[i]] - q1) / (q2 - q1)], Point[nodes[[i]]]}, {i, 1, nxy}]},
    AspectRatio → Automatic, Background → RGBColor[0, 0, 0]]];
```