In[1]:=

```
<< NDSolve`FEM`
```

Solution of the Stokes equation.

Defining geometry of the domain

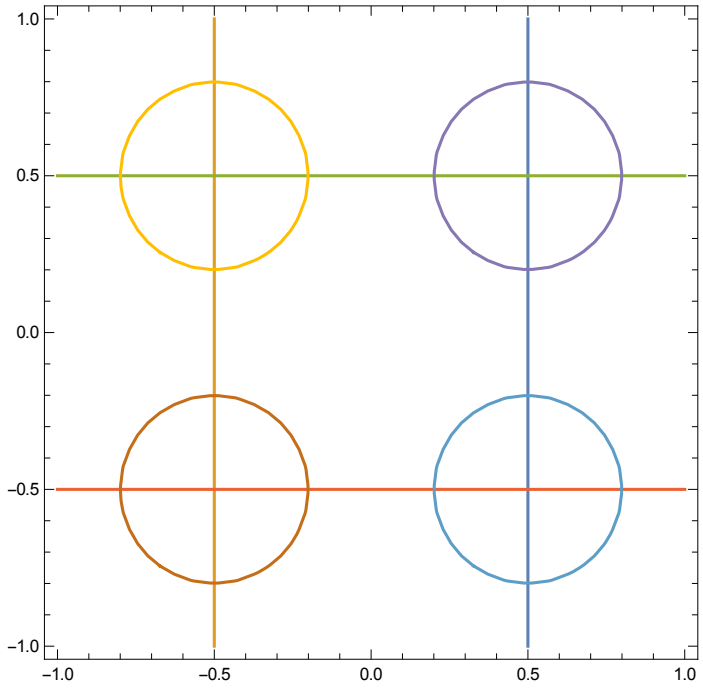
In[2]:=

$$b = \left\{x-\frac{1}{2}, x+\frac{1}{2}, y-\frac{1}{2}, y+\frac{1}{2}, \left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2-0.3^2,\right.$$
$$\left.\left(x+\frac{1}{2}\right)^2+\left(y+\frac{1}{2}\right)^2-0.3^2, \left(x-\frac{1}{2}\right)^2+\left(y+\frac{1}{2}\right)^2-0.3^2, \left(x+\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2-0.3^2\right\};$$

In[3]:=

```
ContourPlot[Evaluate[(# == 0 & /@ b)], {x, -1, 1}, {y, -1, 1}, Contours → {0}]
```

Out[3]=

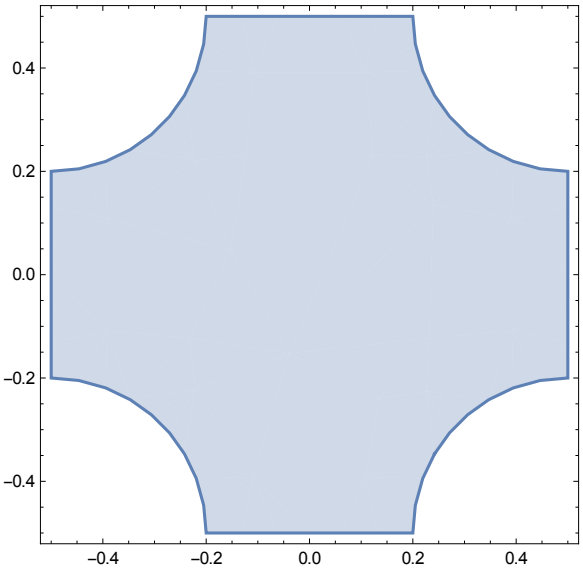


In[4]:=

$$\Omega = \text{ImplicitRegion}\left[!\left(\left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2 \le 0.3^2 \,||\, \left(x+\frac{1}{2}\right)^2+\left(y+\frac{1}{2}\right)^2 \le 0.3^2 \,||\right.\right.$$
$$\left.\left.\left(x-\frac{1}{2}\right)^2+\left(y+\frac{1}{2}\right)^2 \le 0.3^2 \,||\, \left(x+\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2 \le 0.3^2\right), \left\{\left\{x, -\frac{1}{2}, \frac{1}{2}\right\}, \left\{y, -\frac{1}{2}, \frac{1}{2}\right\}\right\}\right];$$

In[5]:=
```
RegionPlot[Ω, AspectRatio→ Automatic , ImageSize →300]
```

Out[5]=



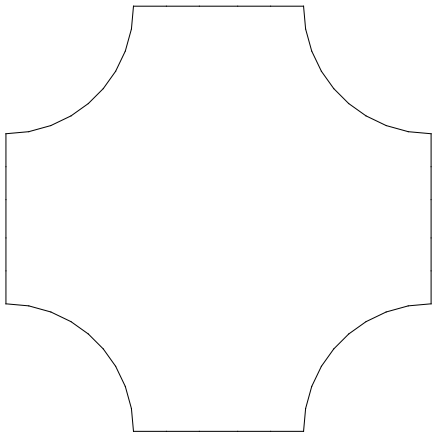
Creating finite element mesh

In[6]:=
```
bM = ToBoundaryMesh[Ω, "MaxBoundaryCellMeasure" → 0.1]
```

Out[6]= `ElementMesh [{{-0.5, 0.5}, {-0.5, 0.5}}, Automatic ]`

In[7]:=
```
ElementMeshWireframe  [bM]
```

Out[7]=



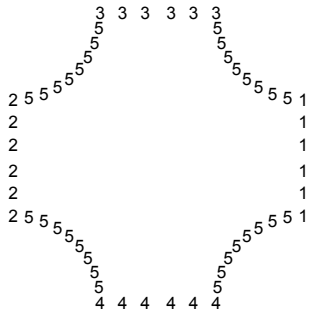
In[8]:=
```
pointMarkerFunction=
    Compile [{{coords, _Real, 2}}, (Block[{x = #1[[1]], y = #1[[2]], epsilon}, epsilon= 1/10^5.;
            Which[Abs[x - 1/2] ≤ epsilon, 1, Abs[x + 1/2] ≤ epsilon, 2, Abs[y - 1/2] ≤ epsilon,
              3, Abs[y + 1/2] ≤ epsilon, 4, ((x - 1/2)^2 + (y - 1/2)^2 - 0.3^2 ≤ epsilon ||
                (x + 1/2)^2 + (y + 1/2)^2 - 0.3^2 ≤ epsilon || (x - 1/2)^2 + (y + 1/2)^2 - 0.3^2 ≤ epsilon ||
                (x + 1/2)^2 + (y - 1/2)^2 - 0.3^2 ≤ epsilon), 5, True, 0]] &) /@coords];
```

In[9]:=
```
pm  = pointMarkerFunction[bM["Coordinates"]]
```

Out[9]=
```
{2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
  5, 3, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5}
```

In[10]:=
```
Graphics[{PointSize[0.002], Point[{4 / Sqrt[5], 2 / Sqrt[5]}],
    MapThread[Text[ToString[#1], #2] &, {pm , bM["Coordinates"]}]}]
```

Out[10]=



In[11]:=
```
boundaryMarkerFunction= Compile [{{boundaryElementCoords , _Real, 3},
      {boundaryElementPointMarkres , _Integer, 2}},
    Which[Union[#] == {1}, 1, Union[#] == {2}, 2, Union[#] == {3}, 3,
      Union[#] == {4}, 4, True, 5] &/@boundaryElementPointMarkres ];
```

In[12]:=
```
bM = ToBoundaryMesh[Ω, "PointMarkerFunction" → pointMarkerFunction,
    "BoundaryMarkerFunction" → boundaryMarkerFunction,
    "MaxBoundaryCellMeasure" → 0.1,
    "BoundaryMeshGenerator" → "RegionPlot", "RegionHoles" → None]
```

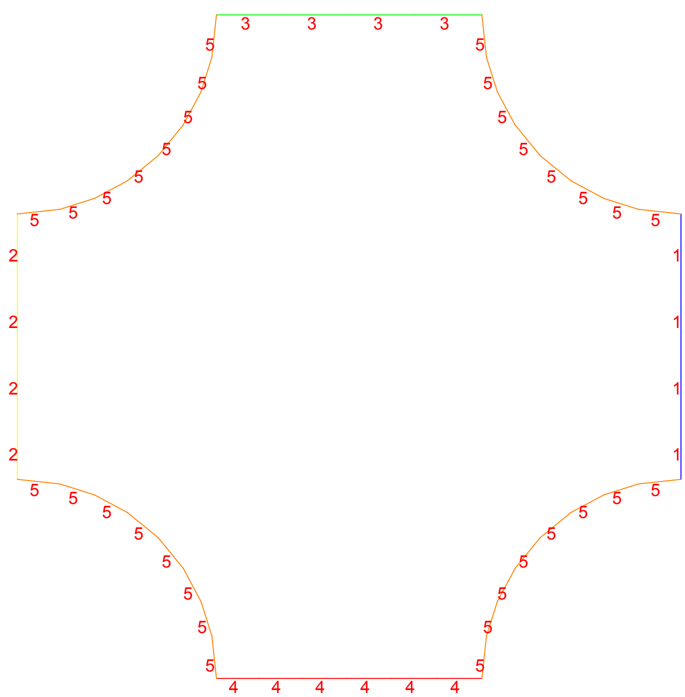
Out[12]=
```
ElementMesh [{{-0.5, 0.5}, {-0.5, 0.5}}, Automatic ]
```

In[13]:=
```
bM["BoundaryElements "]
```

Out[13]=
```
{LineElement [{{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 7}, {7, 8}, {8, 9}, {9, 10}, {10, 11},
    {11, 12}, {12, 13}, {13, 14}, {14, 15}, {15, 16}, {16, 17}, {17, 18}, {18, 19},
    {19, 20}, {20, 21}, {21, 22}, {22, 23}, {23, 24}, {24, 25}, {25, 26}, {26, 27},
    {27, 28}, {28, 29}, {29, 30}, {30, 31}, {31, 32}, {32, 33}, {33, 34}, {34, 35},
    {35, 36}, {36, 37}, {37, 38}, {38, 39}, {39, 40}, {40, 41}, {41, 42}, {42, 43},
    {43, 44}, {44, 45}, {45, 46}, {46, 47}, {47, 48}, {48, 49}, {49, 50}, {50, 1}},
  {4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 2, 2, 2, 2, 5, 5, 5, 5, 5, 5, 5, 5, 3, 3, 3, 3,
    5, 5, 5, 5, 5, 5, 5, 5, 1, 1, 1, 1, 5, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4}]}
```

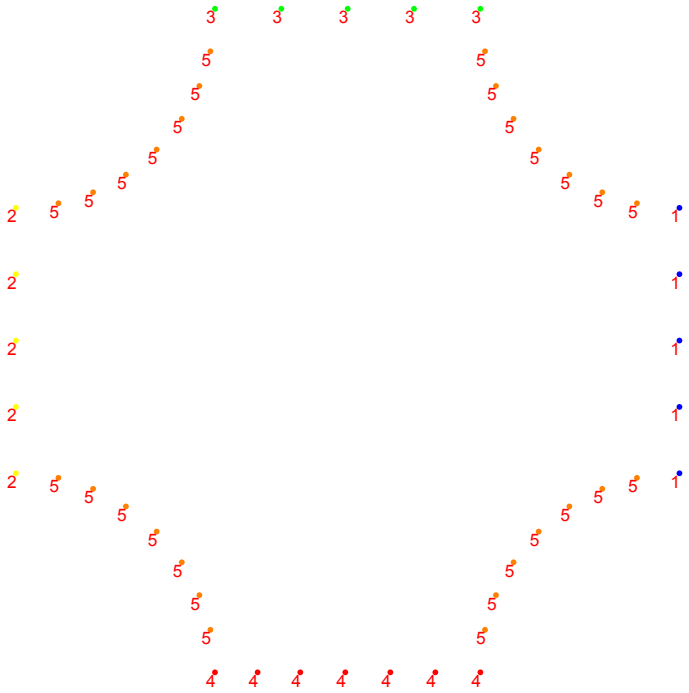
In[14]:=
```
bM["Wireframe "["MeshElementMarkerStyle " → Red,
   "MeshElementStyle " → {Blue, Yellow, Green, Red, Orange}]]
```

Out[14]=



In[15]:=
```
bM["Wireframe "["MeshElement "→"PointElements ", "MeshElementMarkerStyle "→Red,
    "MeshElementStyle "→{Blue, Yellow, Green, Red, Orange}]]
```

Out[15]=



In[16]:=
```
mesh = ToElementMesh [bM, "MaxCellMeasure"→0.1]
```

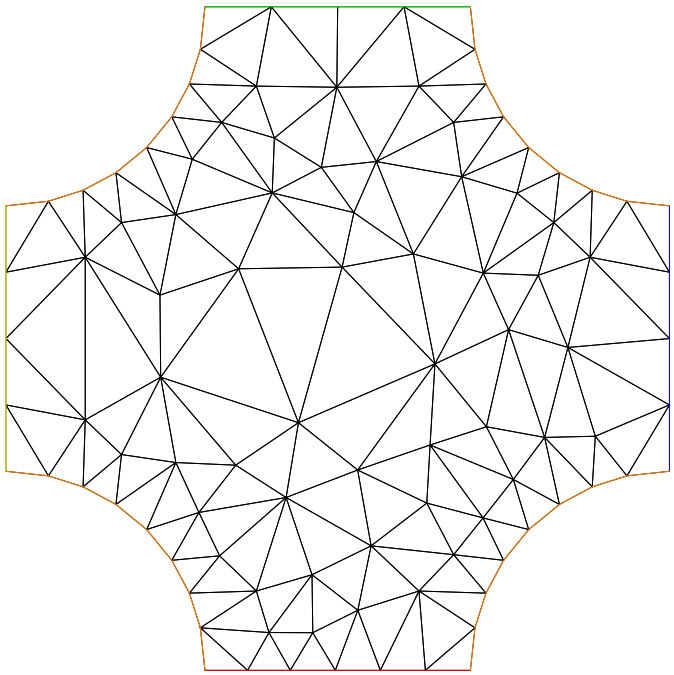
Out[16]=
```
ElementMesh [{{-0.5, 0.5}, {-0.5, 0.5}}, {TriangleElement [<152>]}]
```

In[17]:=
```
Union[Join@@ElementMarkers [mesh ["BoundaryElements "]]]
```

Out[17]=
```
{1, 2, 3, 4, 5}
```

In[18]:=
```
Show[mesh ["Wireframe "], mesh ["Wireframe "["MeshElement " → "BoundaryElements ",
    "MeshElementStyle " → {Blue, Yellow, Green, Red, Orange}]]]
```

Out[18]=

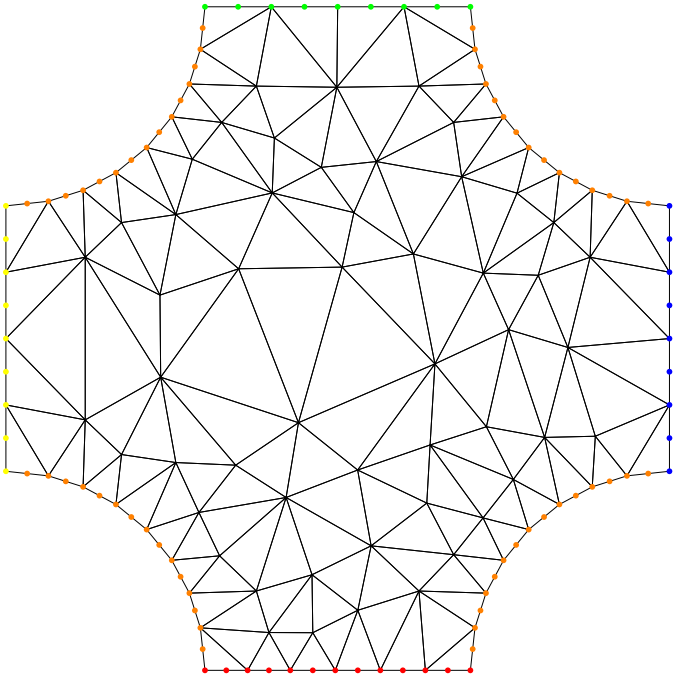


In[19]:=
```
Union[Join@@ElementMarkers [mesh ["PointElements "]]]
```

Out[19]= {1, 2, 3, 4, 5}

In[20]:=
```
Show[mesh ["Wireframe "], mesh ["Wireframe "["MeshElement " → "PointElements ",
    "MeshElementStyle " → {Blue, Yellow, Green, Red, Orange}]]]
```

Out[20]=

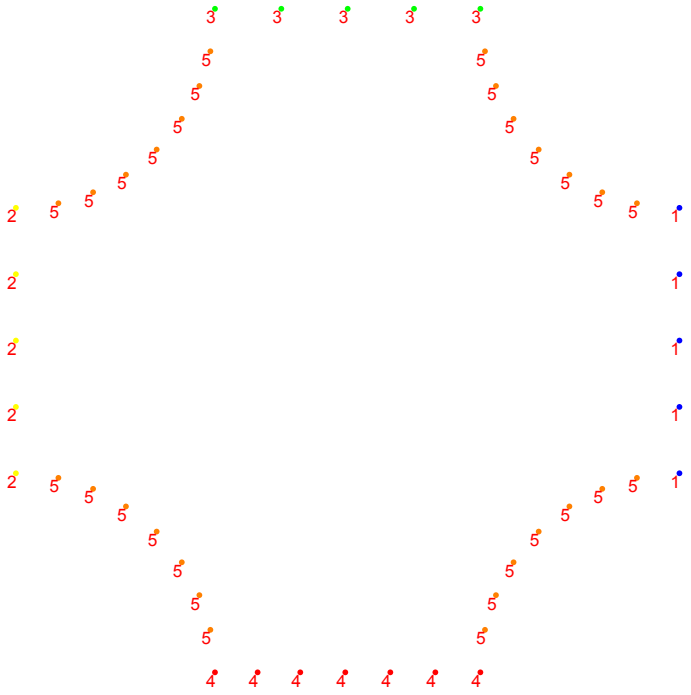


In[21]:=
```
ℐ = IdentityMatrix[2];
```

In[22]:=
```
stokesFlowOperator = {∇{x,y} · (ℐ . ∇{x,y} u[x, y]) + w^(1,0)[x, y],
    ∇{x,y} · (ℐ . ∇{x,y} v[x, y]) + w^(0,1)[x, y], v^(0,1)[x, y] + u^(1,0)[x, y]};
```

In[23]:=
```
bM["Wireframe "["MeshElement "→"PointElements ", "MeshElementMarkerStyle "→Red,
    "MeshElementStyle "→{Blue, Yellow, Green, Red, Orange}]]
```

Out[23]=



Defining boundary conditions

```
Γ_D = {DirichletCondition[{w[x, y] == 1}, ElementMarker == 1],
       DirichletCondition[{w[x, y] == 0}, ElementMarker == 2],
       DirichletCondition[{u[x, y] == 0., v[x, y] == 0.}, ElementMarker == 3],
       DirichletCondition[{u[x, y] == 0., v[x, y] == 0.}, ElementMarker == 4],
       DirichletCondition[{u[x, y] == 0., v[x, y] == 0.}, ElementMarker == 5]};
```

Solution of the equation

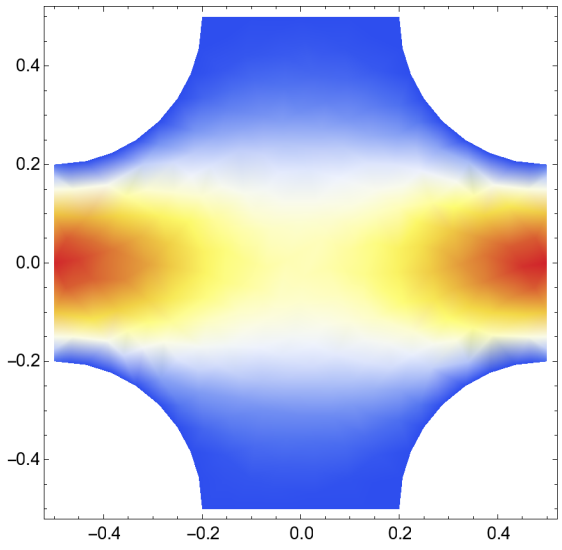
In[25]:=
```
{xVel, yVel, pressure} =
    NDSolveValue[{stokesFlowOperator == {0, 0, 0}, Γ_D}, {u, v, w}, {x, y} ∈ mesh ,
      Method→{"FiniteElement", "InterpolationOrder"→{u→2, v→2, w→2}}];
```

Plot the equation

In[28]:=

```
DensityPlot[Sqrt[xVel[x, y]^2 + yVel[x, y]^2], {x, y} ∈ mesh ,
  AspectRatio → Automatic , ColorFunction → "TemperatureMap "]
```

Out[28]=



In[29]:=

```
Show[BoundaryDiscretizeRegion[Ω],
  StreamPlot [{xVel[x, y], yVel[x, y]}, {x, y} ∈ mesh , AspectRatio → Automatic ]]
```

InterpolatingFunction::dmval : Input value {−0.5, −0.5} lies outside the range of data in the interpolating function. Extrapolation will be used. >>

Out[29]=