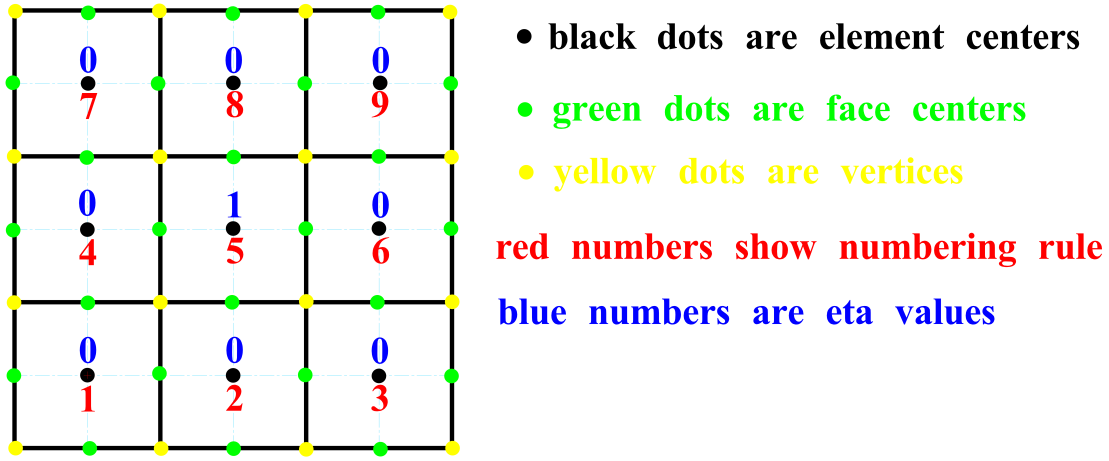


Figure 1: fipy mesh

I generate a mesh by

```
import fipy as fp
dx = 1.
dy = dx
nx = 3 #number of elements in x direction
ny = nx #number of elements in y direction
fvm_mesh = fp.Grid2D(dx=dx,dy=dy,nx=nx,ny=ny)
```

The mesh is shown in Fig1. Then I define a scalar field $\eta$ by

```
eta = fp.CellVariable(mesh=fvm_mesh)
```

The $\eta$ values(default 0) can only be defined at the element(cell) centers. Then I change the $\eta$ value at the mesh middle by

```
fvm_x = fvm_mesh.cellCenters[0]
fvm_y = fvm_mesh.cellCenters[1]
eta.setValue(1., where= (fvm_x > 0.5*nx*dx-0.6*dx) & (fvm_x < 0.5*nx*dx+0.6*dx)
    & (fvm_y > 0.5*ny*dy-0.6*dy) & (fvm_y < 0.5*ny*dy+0.6*dy))
```

We therefore get blue numbers in Fig1. No matter the coordinates array fvm_x(fvm_y) or $\eta$ values array, both are stored according to numbering rule. So, $\eta$ values now are actually array([0,0,0,0,1,0,0,0,0]).
In fipy(or FVM), we do not care about the $\eta$ values at vertices. However, In FEM, calculation is performed in vertices. So, I have to manually assign values to vertices, in order to be usable in FEM. This is done by

```
eta_value = np.zeros((nx*ny*4))
for i in range(nx*ny):
    eta_value[i*4] = eta.value[i]
    eta_value[i*4+1] = eta.value[i]
    eta_value[i*4+2] = eta.value[i]
    eta_value[i*4+3] = eta.value[i]
```

Now, eta_value stores the vertices values. There are nx∗ny elements, each element has 4 vertices. So, the shape of eta_value is nx∗ny∗4. Now, a non-constant(which means not all

elements have the same prestress) prestress field can be defined by

```
stress00 = np.array([30.03881026, 30.03881026, 43.6856391])
eta_value.shape = (nx*ny*4,1,)
prestress_value = eta_value*stress00
```

However, I do not know how to fit this prestress_value into sfepy framework.