```
for index in mesh.specialSets["MinJ_VertexSet"]:下边界
    temperatureField.data[index] = tempMax
for index in mesh.specialSets["MaxJ_VertexSet"]:上边界
    temperatureField.data[index] = tempMin
```

自由边界，无滑边界，stokes边界：

```
freeSlipBC = uw.conditions.DirichletCondition(velocityField, (IWalls,JWalls) )
# this will give free slip sides；猜测：上下边界vy的导数固定为0，即vx可以不为0？
 noSlipBC = uw.conditions.DirichletCondition(velocityField,
(IWalls+JWalls,IWalls+JWalls) )  # this will give no slip sides
stokesBC = uw.conditions.DirichletCondition( variable      = velocityField,
                                             indexSetsPerDof = (allWalls, jWalls) )
periodicBC = uw.conditions.DirichletCondition( variable      = velocityField,
                                               indexSetsPerDof = ( bottomWall, jWalls) )
or：periodicBC = uw.conditions.DirichletCondition( variable      = velocityField,
                                                   indexSetsPerDof = ( jWalls, jWalls) )
```

Create free-slip condition on the vertical boundaries, and a no-slip condition on the horizontal boundaries

```
# Construct node sets using the mesh specialSets
iWalls = mesh.specialSets["MinI_VertexSet"] + mesh.specialSets["MaxI_VertexSet"]
jWalls = mesh.specialSets["MinJ_VertexSet"] + mesh.specialSets["MaxJ_VertexSet"]
allWalls = iWalls + jWalls

# Prescribe degrees of freedom on each node to be considered Dirichlet conditions.
# In the x direction on allWalls flag as Dirichlet
# In the y direction on jWalls (horizontal) flag as Dirichlet
stokesBC = uw.conditions.DirichletCondition( variable      = velocityField,
                                             indexSetsPerDof = (allWalls, jWalls) )
```

```
# 2D velocity vector can have two Dirichlet conditions on each vertex,
# v_x is fixed on the iWalls (vertical), v_y is fixed on the jWalls (horizontal)
velBC  = uw.conditions.DirichletCondition( variable        = velocityField,
                                           indexSetsPerDof = (iWalls, jWalls) )
```