二维矩形网格的表示方法

# 总览

使用一个mesh结构来记录整个矩形网格。这个网格要求每条边都平行于轴或轴。mesh中应该包含下列信息：

1. 格点的坐标；
2. 哪些格点对之间有边相连；
3. 哪些边围成一个矩形区域；
4. 哪些边在边界上。

mesh结构的成员有：

mesh.Nnodes, mesh.Nedges, mesh.Nsurfaces: 整数，各种元素的个数

mesh.nodes: cell数组，记录每个节点的详细信息

mesh.edges: cell数组，记录每条边的详细信息

mesh.surfaces: cell数组，记录每个矩形区域的详细信息

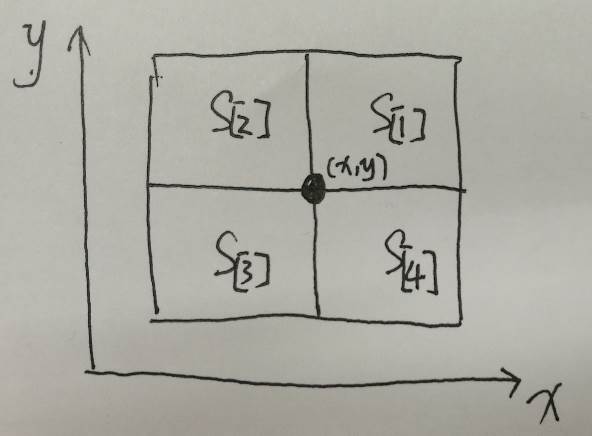
# 节点信息

节点信息储存在mesh.nodes中，mesh.nodes{i}记录编号为i的节点的信息。总结点数为mesh.Nnodes。下面是mesh.nodes{i}的成员：

x, y: 节点的坐标

s[1:4]: 与节点相邻的四个矩形的编号，0表示该位置没有矩形区域(在求解域外)。以当前节点为原点，s[1]~s[4]分别对应第一~四象限位置的矩形。x,y轴正方向的定义与全局一致。

onBoundary: 0/1，表示节点是否在边界上。



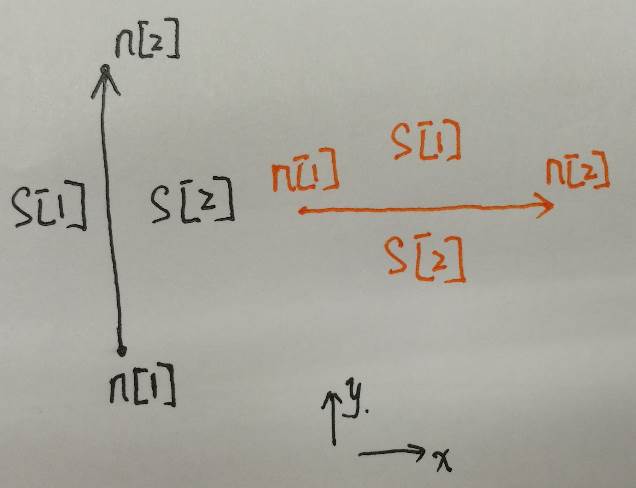
# 边信息

边的信息存储在mesh.edges中，mesh.edges{i}记录编号为i的边的信息。总边数为mesh.Nedges。下面是mesh.edges{i}的成员：

n[1:2]: 边的两个端点的节点编号，这里的边总是平行于坐标轴的，约定节点n[1]的横纵坐标总是不大于n[2]的。

s[1:2]: 边两侧的矩形区域的编号，以n[1]🡪n[2]方向看，左手边的是s[1]，右手边的是s[2]。0表示该方向已经出了求解域。

onBoundary: 0/1，表示边是否在边界上。



# 面信息

面信息存储在mesh.surfaces中，mesh.surfaces{i}记录编号为i的面的信息。总面数为mesh.Nsurfaces。下面是mesh.surfaces{i}的成员：

n[1:4]: 面的四个顶点编号（这里的面仅限于矩形区域），n[1]~n[4]分别是第一~四象限方向的顶点。

e[1:4]: 面的四条边的编号，e[1]~e[4]分别连接n[1]🡪 n[2]🡪 n[3]🡪 n[4]🡪n[1]。

x[1:2], y[1:2]：矩形区域的范围，x[1]<x[2], y[1]<y[2]

hx, hy: hx=x[2]-x[1], hy=y[2]-y[1]

