

# FEALPy 偏微分方程数值解程序设计与实现： 网格数据结构

魏华祎 李成新

[weihuayi@xtu.edu.cn](mailto:weihuayi@xtu.edu.cn)

湘潭大学 • 数学与计算科学学院

July 15, 2020

# Outline

- 1 FEALPy 中的网格模块
- 2 FEALPy 中的网格生成示例

# Outline

- 1 FEALPy 中的网格模块
- 2 FEALPy 中的网格生成示例

## FEALPy 中的网格模块

在偏微分方程数值计算程序设计中, 网格是最核心的数据结构, 是下一步实现数值离散方法的基础。FEALPy 中核心网格数据结构是用**数组**表示。

- 三角形、四边形、四面体和六面体等网格, 因为每个单元顶点的个数固定, 因此可以用**节点坐标数组 node** 和**单元拓扑数组 cell** 来表示, 这是一种以**单元为中心的数据结构**。
- 其它的如**边数组 edge**、**面数组 face** 都可由 cell 生成。
- FEALPy 中把 node、edge、face 和 cell 统称为网格中的实体 entity。
- 在二维情形下, FEALPy 中的 edge 和 face 意义是相同的。
- FEALPy 中还有一种以**边中心的网格数据结构**, 称为**半边数据结构 (Half-Edge data structure)**, 它具有更灵活和强大的网格表达能力。

## FEALPy 网格模块中的网格对象

IntervalMesh	区间网格
TriangleMesh	三角形网格
QuadrangleMesh	四边形网格
TetrahedronMesh	四面体网格
HexahedronMesh	六面体网格
PolygonMesh	多边形网格
PolyhedronMesh	多面体网格
StructureQuadMesh	结构四边形网格
StructureHexMesh	结构六面体网格
Tritree	三角形树结构网格
Quadtree	四叉树
OCTree	八叉树
HalfEdgeMesh2d	二维半边网格
HalfEdgeMesh3d	三维半边网格

Table: FEALPy 中的网格类。

## FEALPy 中网格示例:三角形

---

### Python code 1 创建一个三角形网格。

---

```
1 import numpy as np
2 from fealpy.mesh import TriangleMesh
3
4 node = np.array([(0.0, 0.0), (1.0, 0.0), (1.0, 1.0), (0.0, 1.0)], dtype=np.float)
5 cell = np.array([(1, 2, 0), (3, 0, 2)], dtype=np.int)
6
7 mesh = TriangleMesh(node, cell)
8 mesh.uniform_refine(n=3)
9
10 node = mesh.entity('node')
11 edge = mesh.entity('edge')
12 cell = mesh.entity('cell')
13
14 fig = plt.figure()
15 axes = fig.gca()
16 mesh.add_plot(axes)
17 mesh.find_node(axes, showindex=True)
18 mesh.find_cell(axes, showindex=True)
19 plt.show()
```

---

## FEALPy 中网格示例:三角形

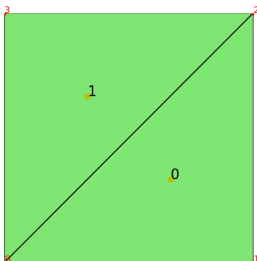


Figure: 初始三角形网格。

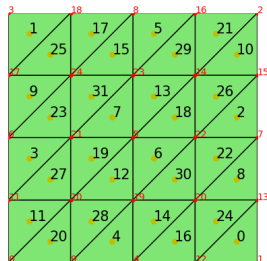


Figure: 一致加密 2 次三角形网格。

## FEALPy 中网格示例: 四边形

---

### Python code 3 创建一个四边形网格。

---

```
1 import numpy as np
2 from fealpy.mesh import QuadrangleMesh
3
4 node = np.array([(0,0), (1,0), (1,1), (0,1)], dtype=np.float)
5 cell = np.array([(0,1,2,3)], dtype = np.int)
6
7 mesh = QuadrangleMesh(node, cell)
8 mesh.uniform_refine(2)
9
10 node = mesh.entity('node')
11 edge = mesh.entity('edge')
12 cell = mesh.entity('cell')
13
14 fig = plt.figure()
15 axes = fig.gca()
16 mesh.add_plot(axes)
17 mesh.find_node(axes, showindex=True)
18 mesh.find_cell(axes, showindex=True)
19 plt.show()
```

---



## FEALPy 中网格示例: 四边形

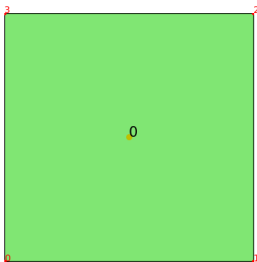


Figure: 初始四边形网格

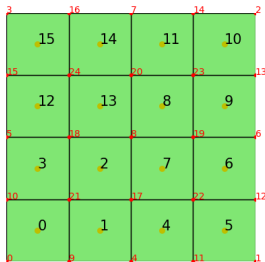


Figure: 一致加密 2 次四边形网格

## FEALPy 中网格示例: 四边形

---

### Python code 5 创建一个半边数据结构网格。

---

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from fealpy.mesh import PolygonMesh, HalfEdgeMesh2d
4
5 node = np.array([
6     (0.0, 0.0), (0.0, 1.0), (0.0, 2.0),
7     (1.0, 0.0), (1.0, 1.0), (1.0, 2.0),
8     (2.0, 0.0), (2.0, 1.0), (2.0, 2.0)], dtype=np.float)
9 cell = np.array([0, 3, 4, 4, 1, 0,
10     1, 4, 5, 2, 3, 6, 7, 4, 4, 7, 8, 5], dtype=np.int)
11 cellLocation = np.array([0, 3, 6, 10, 14, 18], dtype=np.int)
12
13 mesh = PolygonMesh(node, cell, cellLocation)
14 mesh = HalfEdgeMesh2d.from_mesh(mesh)
15
16 fig = plt.figure()
17 axes = fig.gca()
18 mesh.add_plot(axes)
19 mesh.find_cell(axes, showindex=True)
20 mesh.find_node(axes, showindex=True)
21
22 mesh.uniform_refine(n=3)
23
24 node = mesh.entity('node')
25 edge = mesh.entity('edge')
26 cell, cellLocation = mesh.entity('cell')
27 halfedge = mesh.entity('halfedge')
```

## FEALPy 中网格示例: 四边形

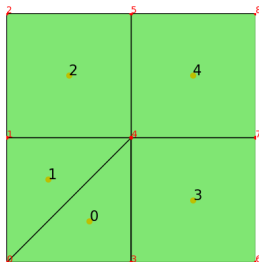


Figure: 初始多边形网格。

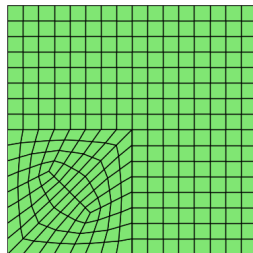


Figure: 一致加密 3 次多边形网格。

## FEALPy 中网格对象的命名与接口约定

变量名	含义
NN	节点的个数
NC	单元的个数
NE	边的个数
NF	面的个数
GD	空间维数
TD	拓扑维数
node	节点数组, 形状为 (NN, GD)
cell	单元数组, 形状为 (NC, NCV)
edge	边数组, 形状为 (NE, 2)
face	面数组, 形状为 (NF, NFV)
ds	网格的拓扑数据结构对象, 所有的拓扑关系数据都由其管理和获取

Table: FEALPy 中网格对象数据成员(属性)的命名约定。

## FEALPy 中网格对象的命名与接口约定

成员函数名	功能
mesh.geo_dimension()	获得网格的几何维数
mesh.top_dimension()	获得网格的拓扑维数
mesh.number_of_nodes()	获得网格的节点个数
mesh.number_of_cells()	获得网格的单元个数
mesh.number_of_edges()	获得网格的边个数
mesh.number_of_faces()	获得网格的面的个数
mesh.number_of_entities(etype)	获得 etype 类型实体的个数
mesh.entity(etype)	获得 etype 类型的实体
mesh.entity_measure(etype)	获得 etype 类型的实体的测度
mesh.entity_barycenter(etype)	获得 etype 类型的实体的重心
mesh.integrator(i)	获得该网格上的第 i 个积分公式

**Table:** 网格对象的常用方法成员(属性)列表。表格中 etype 值可以是 0, 1, 2, 3 或者字符串 ‘cell’, ‘node’, ‘edge’, ‘face’。对于二维网格, etype 的值取 ‘face’ 和 ‘edge’ 是等价的, 但不能取 3。

## FEALPy 中网格对象的命名与接口约定

成员函数名	功能
cell2cell = mesh.ds.cell_to_cell(...)	单元与单元的邻接关系
cell2face = mesh.ds.cell_to_face(...)	单元与面的邻接关系
cell2edge = mesh.ds.cell_to_edge(...)	单元与边的邻接关系
cell2node = mesh.ds.cell_to_node(...)	单元与节点的邻接关系
face2cell = mesh.ds.face_to_cell(...)	面与单元的邻接关系
face2face = mesh.ds.face_to_face(...)	面与面的邻接关系
face2edge = mesh.ds.face_to_edge(...)	面与边的邻接关系
face2node = mesh.ds.face_to_node(...)	面与节点的邻接关系
edge2cell = mesh.ds.edge_to_cell(...)	边与单元的邻接关系
edge2face = mesh.ds.edge_to_face(...)	边与面的邻接关系
edge2edge = mesh.ds.edge_to_edge(...)	边与边的邻接关系
edge2node = mesh.ds.edge_to_node(...)	边与节点的邻接关系
node2cell = mesh.ds.node_to_cell(...)	节点与单元的邻接关系
node2face = mesh.ds.node_to_face(...)	节点与面的邻接关系
node2edge = mesh.ds.node_to_edge(...)	节点与边的邻接关系
node2node = mesh.ds.node_to_node(...)	节点与节点的邻接关系

Table: 网格拓扑数据成员 ds 的方法成员。

## FEALPy 中网格对象的命名与接口约定

成员函数名	功能
<code>isBdNode = mesh.ds.boundary_node_flag()</code>	一维逻辑数组, 标记边界节点
<code>isBdEdge = mesh.ds.boundary_edge_flag()</code>	一维逻辑数组, 标记边界边
<code>isBdFace = mesh.ds.boundary_face_flag()</code>	一维逻辑数组, 标记边界面
<code>isBdCell = mesh.ds.boundary_cell_flag()</code>	一维逻辑数组, 标记边界单元
<code>bdNodeIdx = mesh.ds.boundary_node_index()</code>	一维整数数组, 边界节点全局编号
<code>bdEdgeIdx = mesh.ds.boundary_edge_index()</code>	一维整数数组, 边界边全局编号
<code>bdFaceIdx = mesh.ds.boundary_face_index()</code>	一维整数数组, 边界面全局编号
<code>bdCellIdx = mesh.ds.boundary_cell_index()</code>	一维整数数组, 边界单元全局编号

Table: 网格拓扑数据成员 `ds` 的方法成员。

# Outline

- 1 FEALPy 中的网格模块
- 2 FEALPy 中的网格生成示例



## MeshFactory 模块

FEALPy 中提供了一个 MeshFactory 模块, 用于生成各种类型的常见网格, 方便用户进行 FEALPy 的学习和使用。

---

**Python code 7** 创建一个半边数据结构网格。

---

```
1 from fealpy.mesh import MeshFactory
2
3 mf = MeshFactory()
```

---