Mesh Application Development System

Zhiguo Zhang

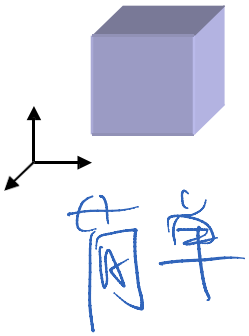
Mesh

- 1D,2D,3D
- Different Elements Selection:
  - 2D**: Tri**angle, **quad**rilateral
  - 3D: **Tet**rahedron, **hex**ahedron, **Pyramid**, **Wedge**
- structural, unstructured
- Partitioned or not

Geometry 几何形状

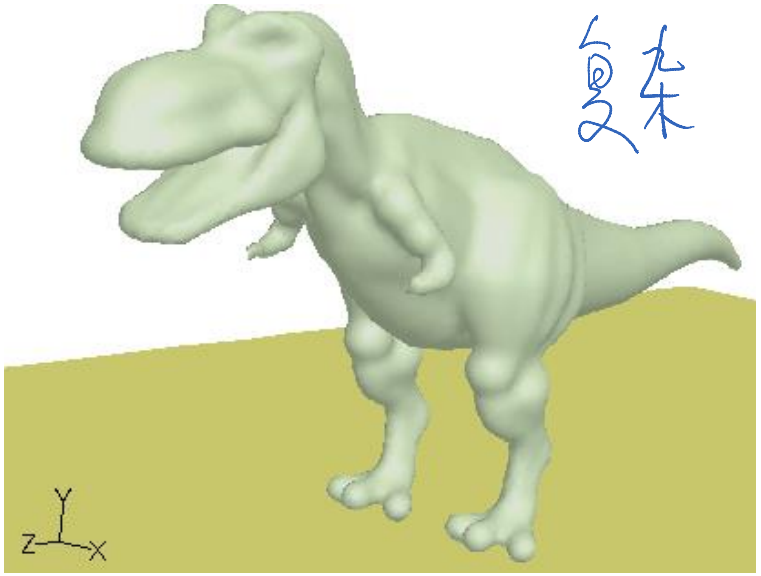
- The starting point for all problems is a "geometry."
- The geometry describes the shape of the problem to be analyzed.
- Can consist of volumes, faces (surfaces), edges (curves) and vertices (points).

Geometry can be very simple...



geometry for a "cube"

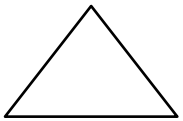
… or more complex



Typical cell shapes

离散化时的单元(Cell)形状

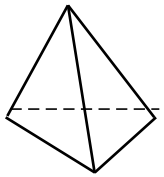
- Many different cell/element and grid types are available. Choice depends on the problem and the solver capabilities.
- Cell or element types:
  - 2D:



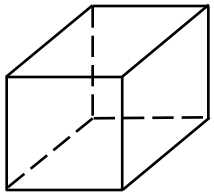
triangle (**"tri"**) 三角

2D prism (**quadrilateral** or **"quad"**) 四边形

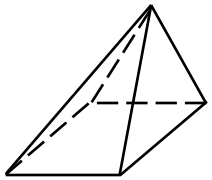
  - 3D:



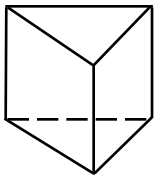
tetrahedron (**"tet"**) 四面体



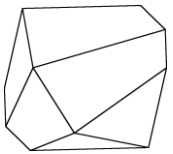
prism with quadrilateral base (**hexahedron** or **"hex"**) 六面体



pyramid 金字塔形



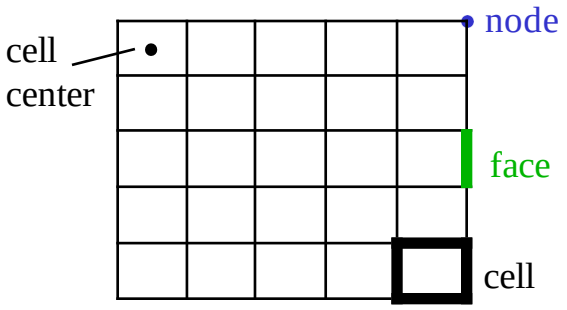
prism with triangular base (**wedge**) 三角形的棱柱体



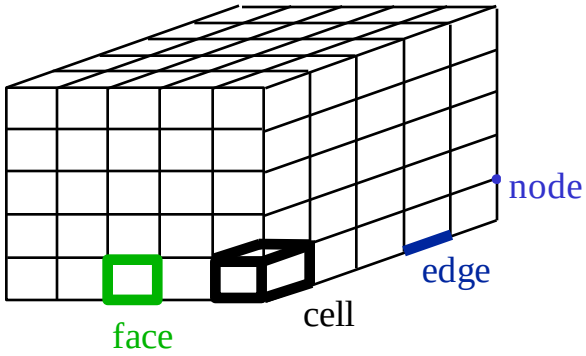
arbitrary polyhedron 任意多面体

Terminology 术语 (有关Mesh)

- **Cell** = control volume into which domain is broken up.
- **Node** = grid point.
- **Cell center** = center of a cell.
- **Edge** = boundary of a face.
- **Face** = boundary of a cell.
- **Zone** = grouping of nodes, faces, and cells:
  - Wall boundary zone.
  - Fluid cell zone.
- **Domain** = group of node, face and cell zones.



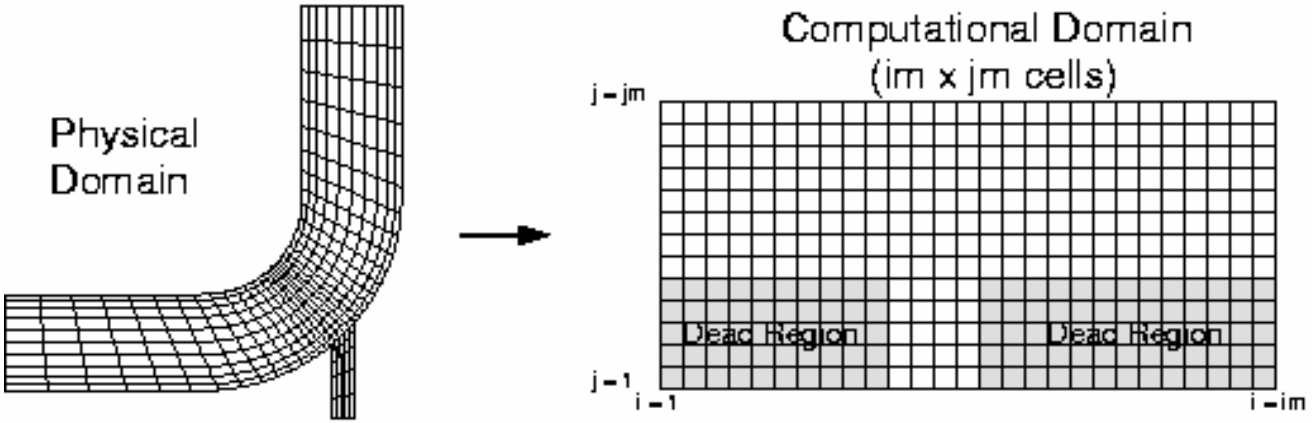
2D computational grid



3D computational grid

Grid types: structured grid 结构化网格

- Single-block, structured grid.
  - i,j,k indexing to locate neighboring cells. 描述为数组，读下标
  - Grid lines must pass all through domain.
- Obviously can't be used for very complicated geometries.



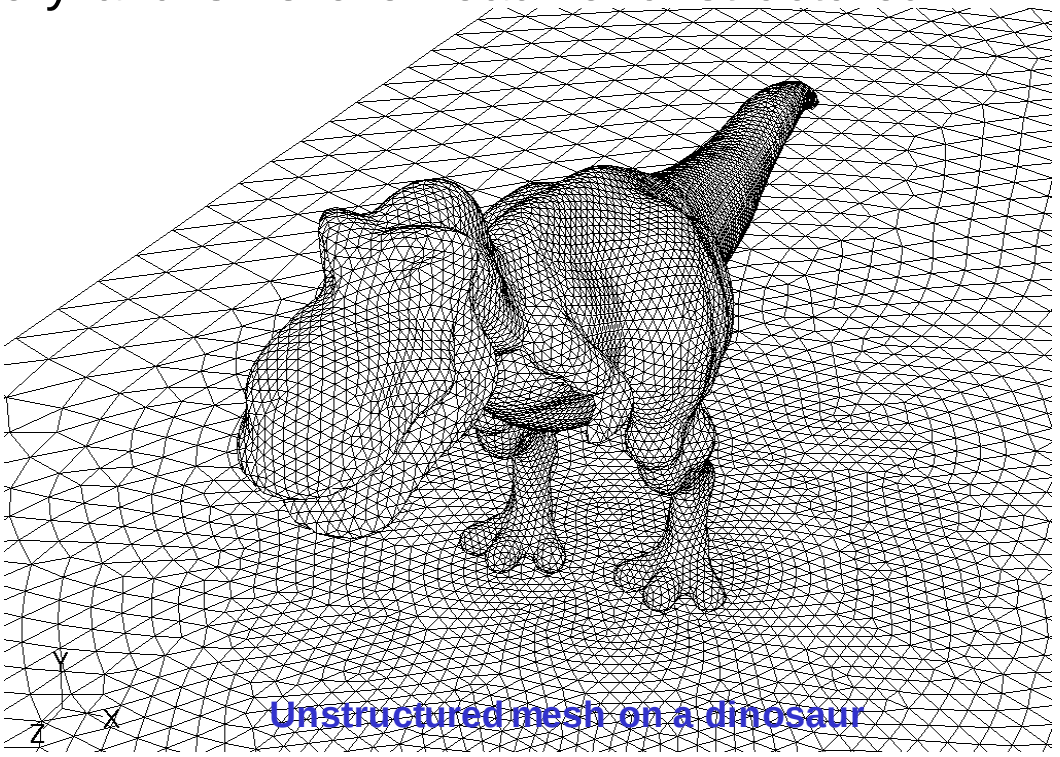
Grid types: unstructured

非结构化网格

- Unstructured grid.
  - The cells are arranged in an arbitrary fashion.
  - No i,j,k grid index, no constraints on cell layout.

单元之间无顺序关系
不能用 i,j,k 索引

- There is some memory and CPU overhead for unstructured referencing.



Unstructured mesh on a dinosaur

Mesh naming conventions - topology

- Structured mesh: the mesh follows a structured i,j,k convention.
- Unstructured mesh: no regularity to the mesh.
- Multiblock: the mesh consists of multiple blocks, each of which can be either structured or unstructured.

在分割网格时的单元类型

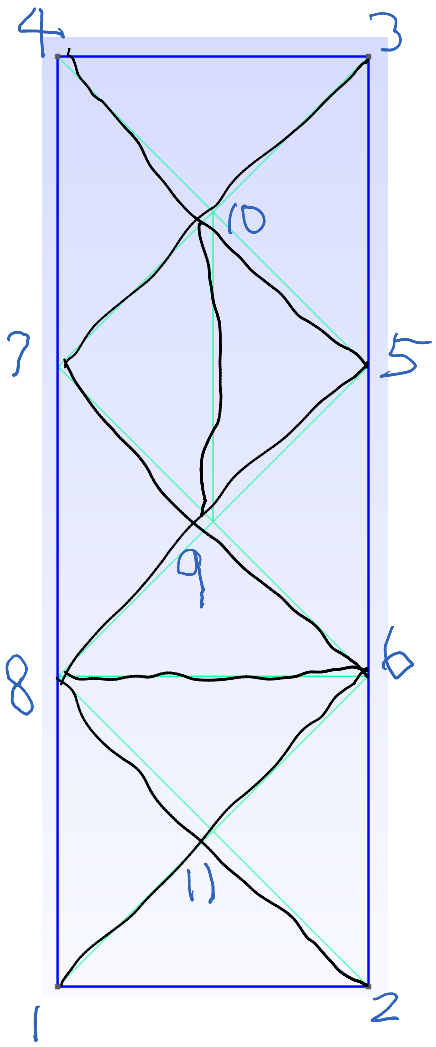
Mesh naming conventions – cell type

- **Tri mesh**: mesh consisting entirely of triangular elements.
- **Quad mesh**: consists entirely of quadrilateral elements.
- **Hex mesh**: consists entirely of hexahedral elements.
- **Tet me**sh: mesh with only tetrahedral elements.
- **Hybrid mesh**: mesh with one of the following:
  - Triangles and quadrilaterals in 2D.
  - Any combination of tetrahedra, prisms, pyramids in 3D.
  - Boundary layer mesh: prizms at walls and tetrahedra everywhere else.
  - Hexcore: hexahedra in center and other cell types at walls.
- **Polyhedral mesh**: consists of arbitrary polyhedra.
- **Nonconformal mesh**: mesh in which grid nodes do not match up along an interface.

Mesh Storage

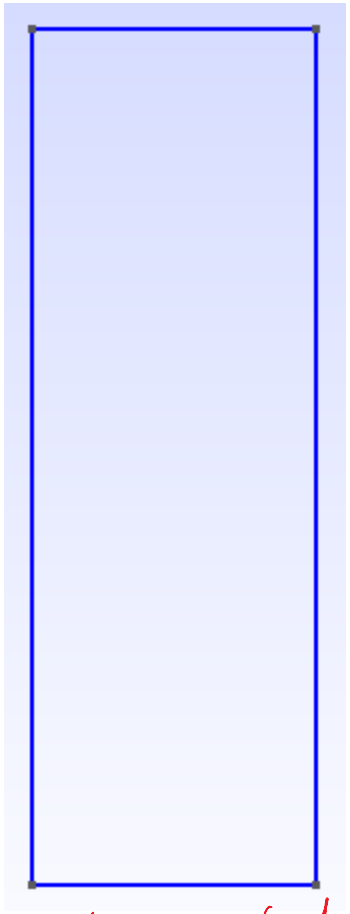
- Storage in Text, binary File ,
- Single file or multiple files
- plain file or structural file like hdf5
- storage in Memory
- Different File Storage formats
- Different memory storage formats

Mesh File Formats: An Example



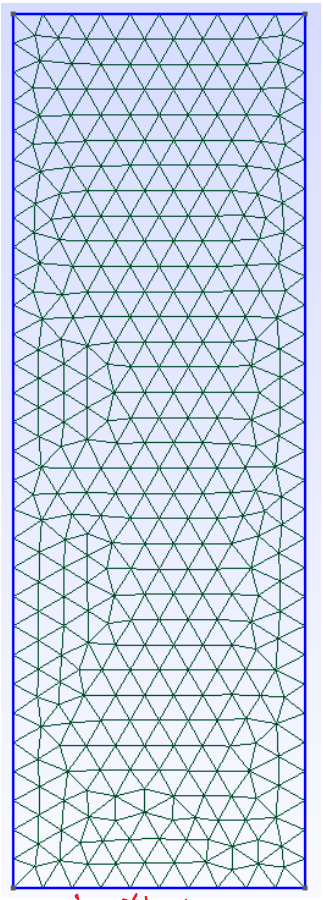
- See “Mesh File Formats.docx” file on Baidu disk.

Discretilizing Mesh



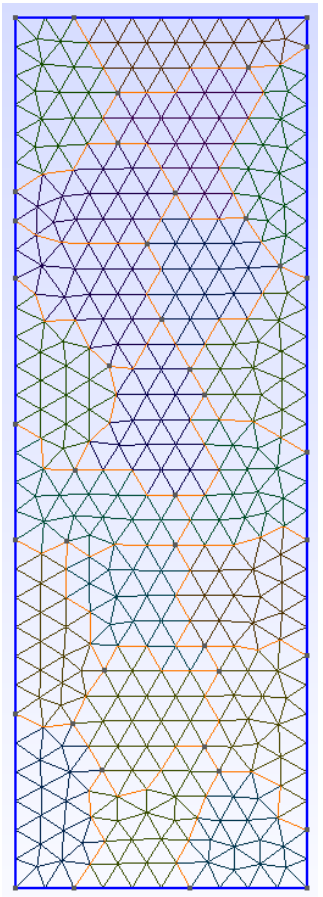
一个平面(二维)

Geometry



离散化

2D Mesh

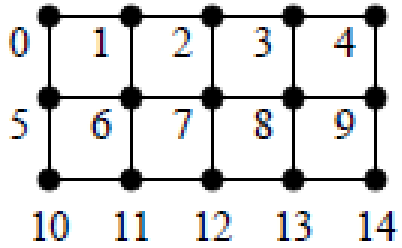


划分成约束(颜色块)

Partitioned Mesh

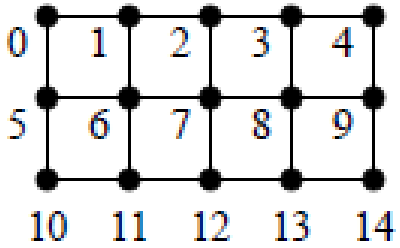
Mesh Can be Understood as Graph

4	9	14
3	8	13
2	7	12
1	6	11
0	5	10



Serial CSR Format

邻接矩阵内存存储格式



xadj $[n+1]$, vwgt $[n]$, adjncy $[2m]$, and adjwgt $[2m]$

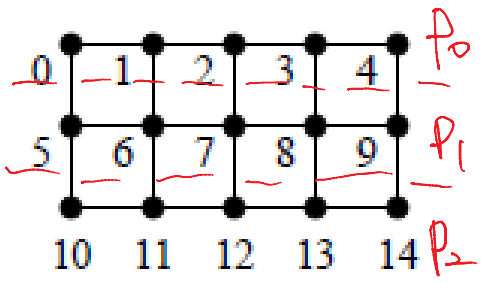
0 12 .....

xadj	0	2	5	8	11	13	16	20	24	28	31	33	36	39	42	44
adjncy	1 5	0	2	6	1	3	7	2	4	8	3	9	0	6	10	1 5 7 11 2 6 8 12 3 7 9 13 4 8 14 5 11 6 10 12 7 11 13 8 12 14 9 13

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

稀疏矩阵的存储方法.

Distributed CSR Format (内存)



Processor 0:	xadj	0 2 5 8 11 13
	adjncy	1 5 0 2 6 1 3 7 2 4 8 3 9
	vtxdist	0 5 10 15
Processor 1:	xadj	0 3 7 11 15 18
	adjncy	0 6 10 1 5 7 11 2 6 8 12 3 7 9 13 4 8 14
	vtxdist	0 5 10 15
Processor 2:	xadj	0 2 5 8 11 13
	adjncy	5 11 6 10 12 7 11 13 8 12 14 9 13
	vtxdist	0 5 10 15

Mesh Application

- It is a C program.
- LoadMesh()
- Define kernel functions
- Define algorithm in terms of kernel functions and looping over mesh.

Mesh Application

```
int save_soln(…) {
    …
}

int main(int argc, char **argv) {
  loadMesh(file);
  partition("PTSCOTCH"); // partitioning and halo creation routines
  for (int iter = 1; iter <= 10000; iter++) {  // main time-marching loop
      par_loop(save_soln, "save_soln",…);
      ....
  }
```

提示：可以把此程序理解为Actors.

Mesh processing

- sequential or parallel processing
- Editing
- Load in and Output
- Coarsening
- sketching
- partitioning
- Viewing (text)
- Visualization

Computing Over Mesh

- Kernel functions on mesh elements
- Algorithm over all mesh
- sequential or parallel Computing
- Partition Topology matches with architectural topology

Development Teams

- Development Environment: Coding (Implementation)
- UML Modeling:
- Cmake
- Git