13.472J/1.128J/2.158J/16.940J
COMPUTATIONAL GEOMETRY

Lecture 23

Dr. W. Cho
Prof. N. M. Patrikalakis

Contents

Lecture 23

F.E. and B.E. Meshing Algorithms

23.1 General

23.1.1 Finite Element Method (FEM)

FEM solves numerically complex *continuous* systems for which it is not possible to construct any *analytic* solution, see Figure 23.1.

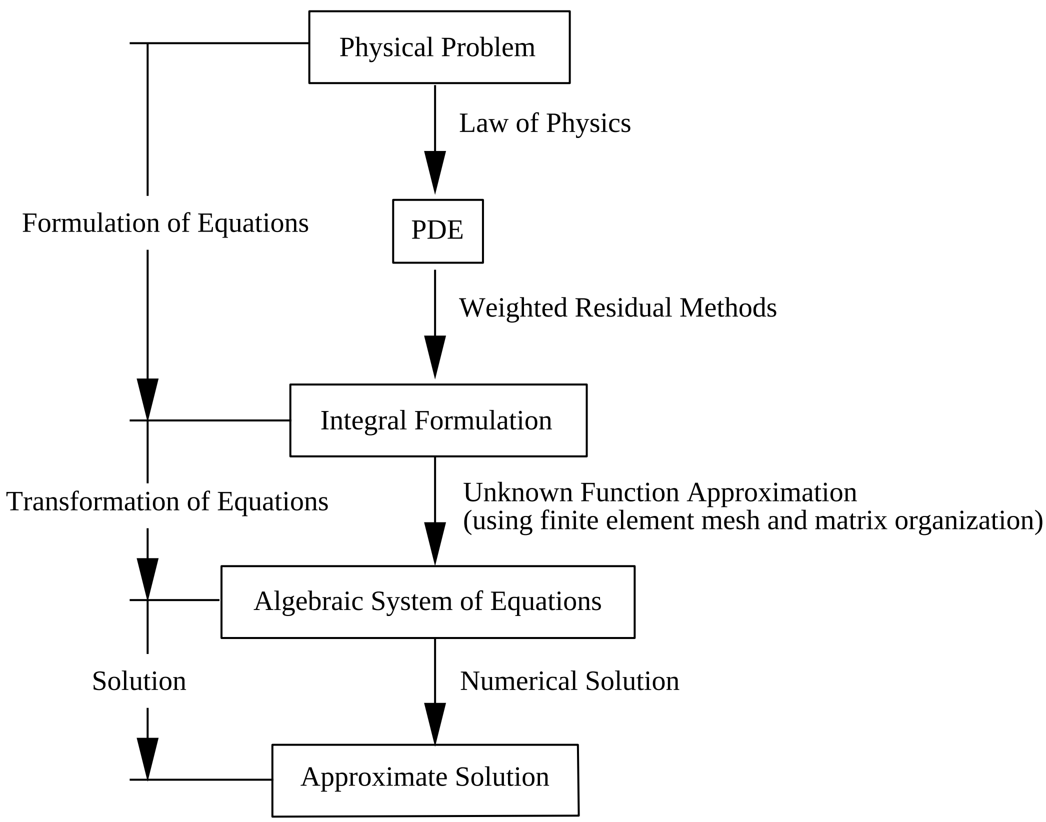

Figure 23.1: FEM Procedure

23.1.2 Mesh

Mesh is the complex of elements discretizing the simulation domain, eg. triangular or quadrilateral mesh in 2D, tetrahedral or hexahedral mesh in 3D.

Why Use Mesh ?

To construct a *discrete* version of the original PDE problems.

Types of Mesh

See Figure 23.2.

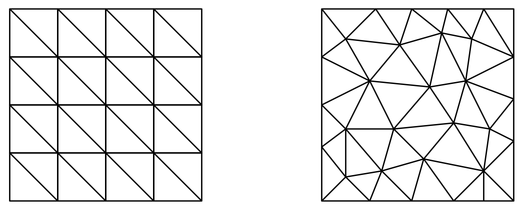

Figure 23.2: Structured ($n = 6$) and unstructured ($n \neq constant$) mesh

- Structured mesh
  - The number of elements n surrounding an internal node is *constant*.
  - The connectivity of the grid can be *calculated* rather than *explicitly stored*.

    → *simpler* and *less* computer memory intensive
  - *Lack* of geometric flexibility
- Unstructured mesh
  - The number of elements surrounding an internal node can be *arbitrary*.
  - *Greater* geometric flexibility

    → crucial when dealing with domains of *complex* geometry or when the mesh has to be *adapted* to complicated features of computational domain or to complicated features of the solution (eg. boundary layers, internal layers, shocks, etc).
  - *Expensive* in time and memory requirements

23.1.3 Some Criteria for a Good Meshing

- Shape of elements [2, 18]
  - meshing should avoid both very *sharp* and *flat* angles, see Figure 23.3.
  - may cause serious *numerical* problems in both finite element mesh generation and analysis.

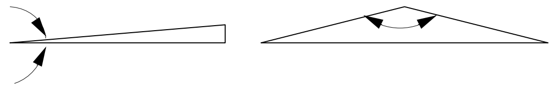

Figure 23.3: Elements with sharp and flat angles

- Number of elements
  - should be moderate.
  - related to *efficiency* of finite element analysis.
- Topological consistency *(homeomorphism)* between the *exact* input domain and its *mesh*.
  - related to *robustness* of finite element analysis [8, 9].
- Automation, adaptability, etc.

23.1.4 Finite Element Analysis in a CAD Environment

See Figure 23.4.

In Figure 23.4, "input data preparation" includes the preparations of boundary conditions, loads, and integration constants Δt etc.; "post-processing of results" includes scientific visualization, etc.

23.1.5 Background Information

Delaunay Triangulation [4, 19]

Given a set of points P_i, the Voronoi region, V_i is a set of points *closer* to a site P_i than any other site P_j (See Figure 23.5). Delaunay triangulation is constructed by connecting those points whose *Voronoi regions* have a common edge (2D) or face (3D). Some properties of Delaunay triangulation are:

- it maximizes the minimum angle (globally).
- circumcircle criterion: every circle passing through the 3 vertices of a triangle does not contain any other vertices.
- it covers the convex hull of all sites.

Mesh Conversion

Performed if a mesh generator produces only one type of element and another type is required.

- Quadrilaterals (Hexahedra) → Triangles (Tetrahedra) [12]: easy and well-shaped mesh, see Figure 23.6.
- Triangles (Tetrahedra) → Quadrilaterals (Hexahedra)

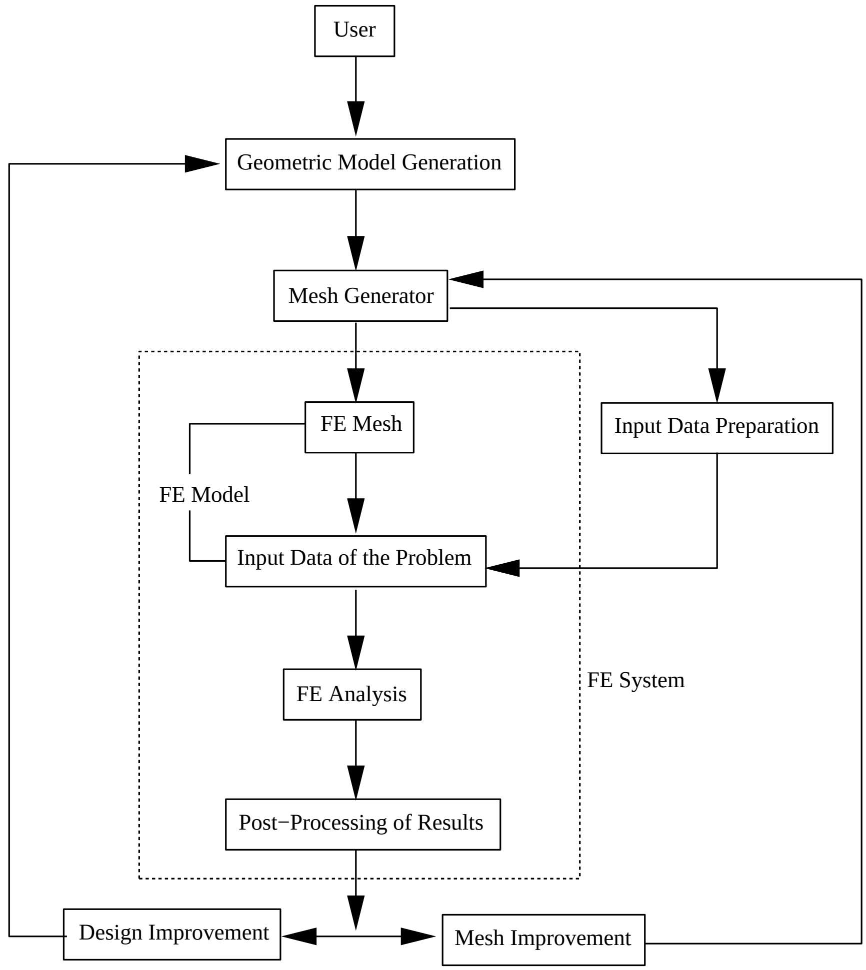

Figure 23.4: FE analysis in a CAD environment

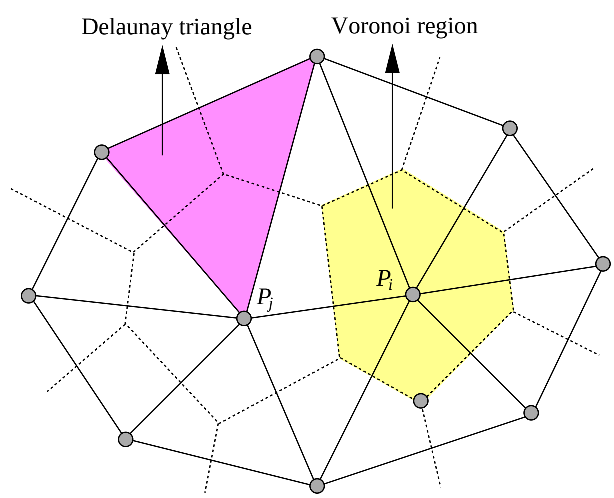

Figure 23.5: 2D Delaunay triangulation

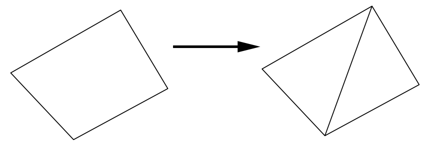

Figure 23.6: Mesh conversion : quadrilateral → triangles

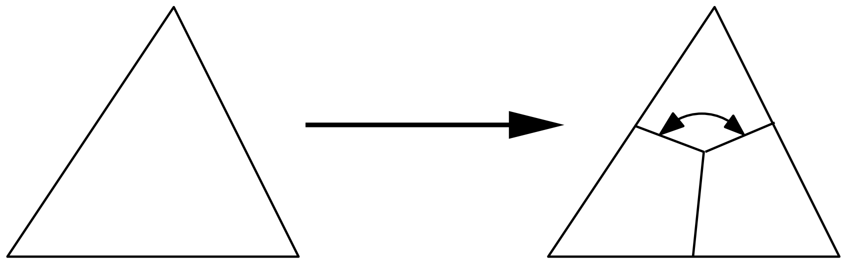

Figure 23.7: New node insertion

- New node insertion [12]: not well-shaped mesh → flat angle, see Figure 23.7.
- Pairing of adjacent triangles (tetrahedra) [14]: pairing test should be performed to generate *convex* quadrilaterals. → pairing ΔABC and ΔCDA is unsuitable, see

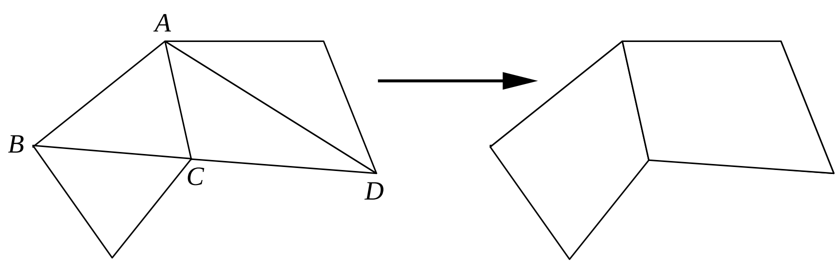

Figure 23.8: Pairing of triangles

Figure 23.8.

Mesh Smoothing

Make elements better-shaped by *iteratively* repositioning an internal node P_i, [12].

- $P_i = \frac{P_i + \sum_{j=1}^{N} P_j}{N+1}$, where N is the number of elements connected by P_i.

  → P_i converges to the centroid of the polygon formed by its connected neighbors.

- Example: After 5 iterations, P_i converges from (7,4) to (4.5,3), see Figure 23.9.

23.1.6 Mesh Conformity

- If adjacent elements share a common vertex or whole edge or a whole face, see Figure 23.10, we have conforming mesh. Otherwise, mesh is non-conforming.

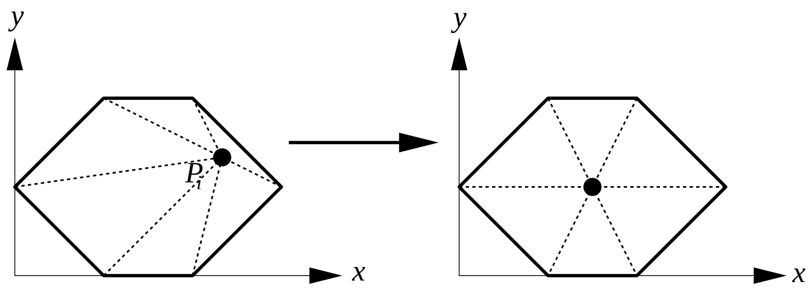

Figure 23.9: Mesh smoothing

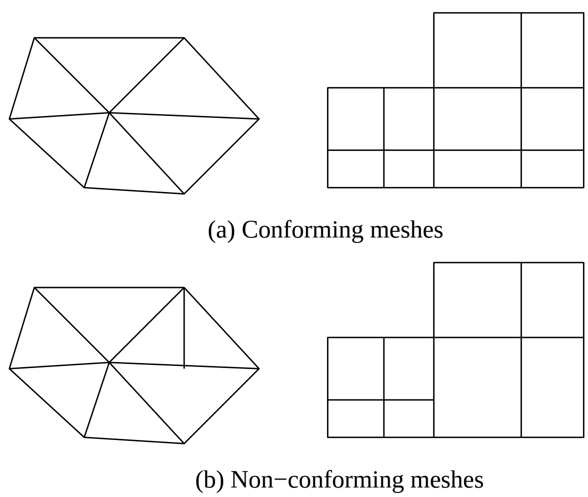

Figure 23.10: Conforming and non-conforming mesh

- In case of non-conforming mesh, multiple *constraints* need to be applied to the midside nodes to make the mesh analyzable [1], see Figures 23.11 and 23.12.

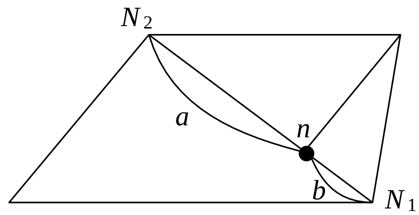

Figure 23.11: Constrained node, n

$\rightarrow$ (eg) $U_n = \frac{b}{a+b} U_{N_2} + \frac{a}{a+b} U_{N_1}$

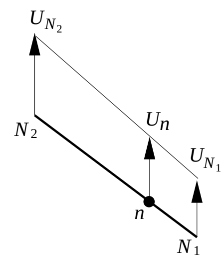

Figure 23.12: Displacement

23.1.7 Mesh Refinement

Increase mesh density from region to region.

Conforming mesh refinement algorithm [16]

Let τ_0 be any conforming initial triangulation having N_0 triangles. Then a new triangulation τ_1 is defined as follows:

1. Bisect triangle T by the *longest* side, $\forall T \in \tau_0$, see Figure 23.13.

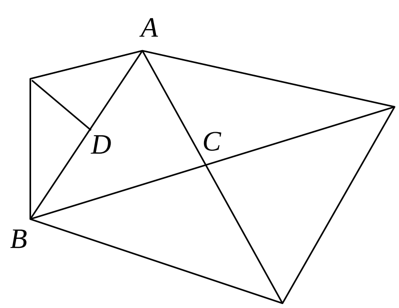

Figure 23.13: Step 1

2. Find the set S_1 of triangles generated in **step 1** and such that the midpoint of one of its sides is a *non-conforming* node, i.e. node D and the corresponding $S_1 = \{\Delta ABC\}$, see Figure 23.13.

3. For each triangle $T \in S_1$, join its non-conforming node with the *opposite* vertex, i.e. join D and C, see Figure 23.14.

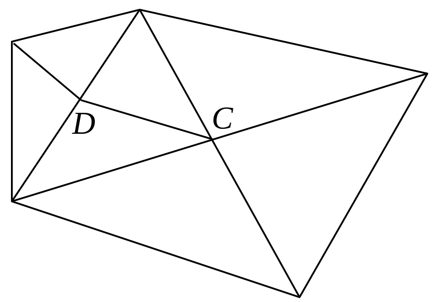

Figure 23.14: Step 3, New conforming triangulation

23.2 Mesh Generation Methods

23.2.1 Mapped Mesh Generation [23, 10, 11, 5]

This is the earliest method for mesh generation which appeared in 1970's.

Mapped Element Approach

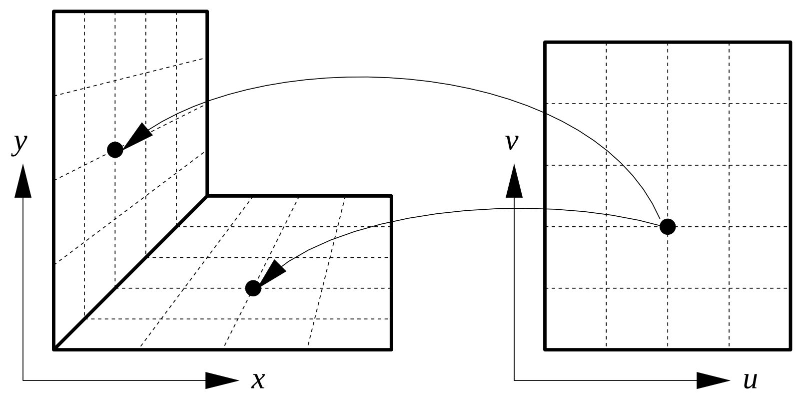

Figure 23.15: Mapped element approach

1. Subdivide the *physical domain* into simple regions, e.g., quadrilaterals, triangles.
2. Generate mesh in *computational space* (u, v), see Figure 23.15.
3. Determine nodes in the physical domain using shape functions.

- Example 1: Isoparametric curvilinear mapping of quadrilaterals
  $\rightarrow$ a mapped mesh generation method for surfaces.

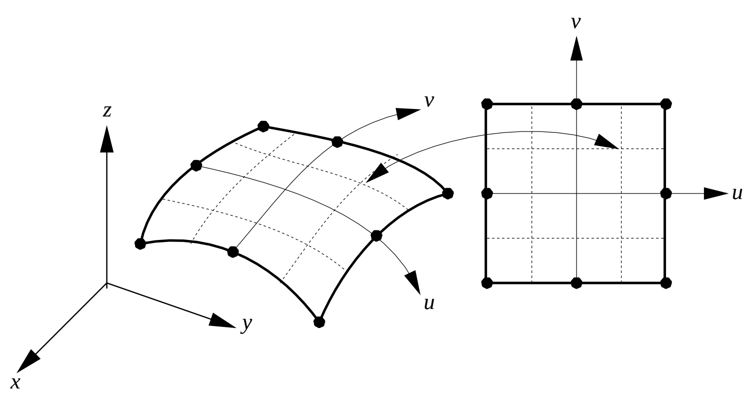

Figure 23.16: Isoparametric curvilinear mapping

$\rightarrow \vec{X} = \sum_{i=1}^{8} N_i(u, v)\vec{X}_i$,
where $\vec{X}_i = (x_i, y_i, z_i)$: coordinates of each node in the physical domain $\vec{X} = (x, y, z)$ and $N_i(u, v)$ is the shape function associated with each node, see Figure 23.16.

- Example 2: Isoparametic mapping for 3D objects, see Figure 23.17.

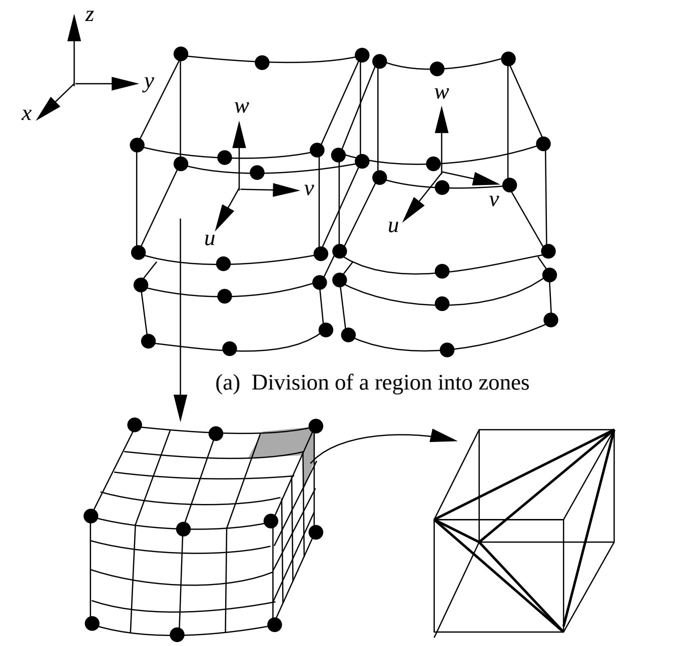

Figure 23.17: Isoparametric mapping for 3D objects

Decomposition of 3D objects requires

- user interaction,
- skeleton transform.

Conformal Mapping Approach [5]

Conformal mapping deals with 2D simply-connected regions having more than 4 sides, see Figure 23.18.

- **Algorithm**

  1. Model a 2D simply-connected region to be meshed with an N-gon P in the complex w-plane.
  2. Define an N-gon Q in the complex z-plane.
  3. Find the *Schwarz-Christoffel* transformation G that maps from the upper half u-plane onto the interior of Q, and the transformation F from the upper half plane onto the interior of P.
  4. Map the mesh from Q onto P using the composite mapping $w = F(G^{-1}(z))$.

- **Note:** G^{-1} is not always found.

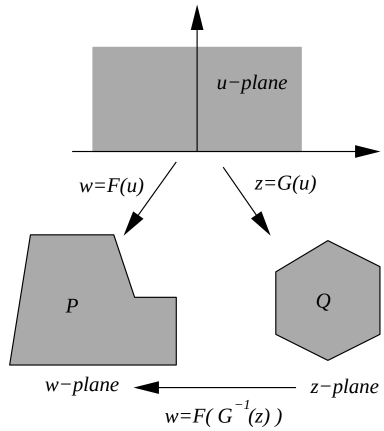

Figure 23.18: Conformal mapping

23.2.2 Topology Decomposition Approach [20, 3]

Subtract a simplex from an object repeatedly until the object is reduced to a single simplex. Simplex in 2D is triangle, while simplex in 3D is tetrahedron.

- **(2D Cases)** Cut off triangles whose vertices are the object vertices until only 3 vertices are left, see Figure 23.19.

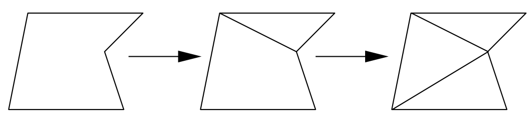

Figure 23.19: 2D topology decomposition

- **(3D Cases)**
  - Operator OP_1 : constructs a tetrahedron by cutting out a corner vertex which has exactly 3 adjacent edges, see Figure 23.20.

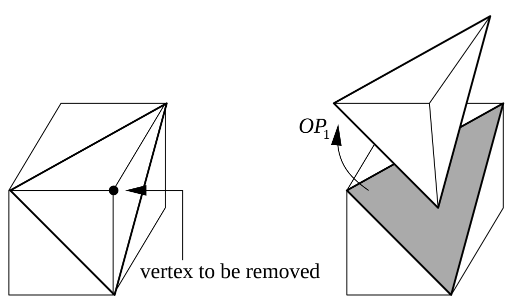

Figure 23.20: Operator OP_1

  - Operator OP_2 : digs out a tetrahedron from a convex edge, see Figure 23.21.

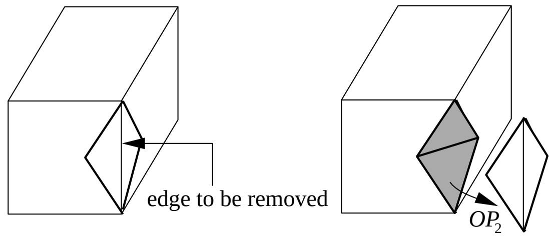

Figure 23.21: Operator OP_2

- **Note:** In either case, the removed tetrahedron must be a part of the object in its entirety.
- **Algorithm**
  1. Start with a simply connected region.
  2. Apply OP_1 to all appropriate vertices.
  3. If all the remaining vertices have more than 3 adjacent edges, apply OP_2 to dig out a tetrahedron.
  4. Go back to **step 2** and continue this process until the object is reduced to a single tetrahedron.
- **Note:** The topology decomposition approach generates gross elements and the mesh must be *refined* later to satisfy the required mesh density distribution.

23.2.3 Geometry Decomposition Approach [6, 17, 13]

Recursive Subdivision

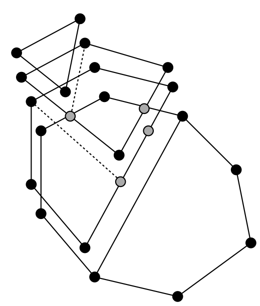

Figure 23.22: Recursive subdivision

1. Start with a convex object, see Figure 23.22.
2. Insert nodes into the boundary of the object to satisfy mesh density distribution.
3. Divide the object roughly in the middle of its longest axis.
4. Insert nodes into the split line according to the mesh density distribution.
5. Recursively divide the two halves until they become triangles

Iterative Removal of Elements

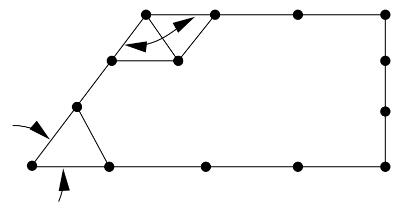

Figure 23.23: Iterative removal of elements

1. Start with a simply connected region, see Figure 23.23.
2. Insert nodes into the boundary of the object to satisfy mesh density distribution.
3. Remove all vertex angles of the polygon which are less than 90° by forming triangular elements.
4. Start at some vertex with an angle of less than 180° but above 90°, and form two triangles with the adjacent points to the vertex.
5. Repeat **step 3**, **step 4** until the last three points form a triangular element.

Iterative Removal of a Boundary Layer [13]

Iteratively remove a boundary layer at a time followed by triangulation, see Figure 23.24.

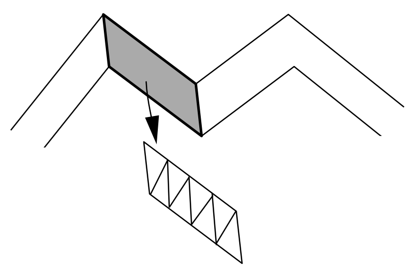

Figure 23.24: Iterative removal of a boundary layer

- Advancing front methods or paving technique
- **Note:** Triangulation near *boundary* is especially well-shaped.

23.2.4 Volume Triangulation – Delaunay Based [7, 15]

2D Case

See page 4.

3D Case

1. Insert nodes on each cross-sections, see Figure 23.25.

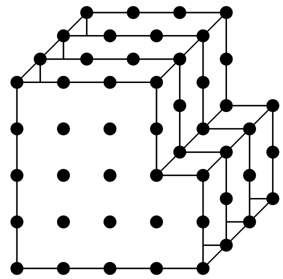

Figure 23.25: Inserted nodes on cross sections

2. Perform Delaunay triangulation, see Figure 23.26.

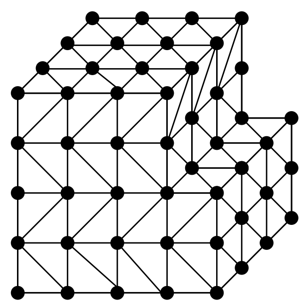

Figure 23.26: Delaunay triangulation of points in Figure 23.25

23.2.5 Spatial Enumeration Methods [21, 22]

Based on *quadtree (octree)* decomposition in 2D (3D), respectively.

A Quadtree Based 2D Meshing Algorithm

- Subdivide the object interior into quadrants whose size satisfy the mesh density distribution.
- Neighboring quadrants may differ by at most one level of subdivision.
- The quadrants on the *boundary* may have *cut corners*.
- Finally, each quadrant is broken up into triangles such that the resulting mesh is *conforming*.
- **Note:** Excellent *interior* elements but in general, not well-shaped *boundary* elements.

Bibliography

[1] *ABAQUS User's Manual*. Hibbitt, Karlsson, and Sorensen, Inc., 1984.

[2] I. Babuska and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, April 1976.

[3] B. Bödenweber. Finite element mesh generation. *Computer-Aided Design*, 16(5):285–291, 1984.

[4] A. Bowyer. Computing dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.

[5] P. R. Brown. A non-interactive method for the automatic generation of finite element meshes using the schwarz-christoffel transformation. *Computer Methods in Applied Mechanics and Engineering*, 25(1):101–126, January 1981.

[6] A. Bykat. Design of a recursive, shape controlling mesh generator. *International Journal for Numerical Methods in Engineering*, 19(9):1375–1390, September 1983.

[7] J. C. Cavendish, D. A. Field, and W. H. Frey. An approach to automatic three-dimensional finite element mesh generation. *International Journal of Numerical Methods in Engineering*, 21:329–347, 1985.

[8] W. Cho, T. Maekawa, N. M. Patrikalakis, and J. Peraire. Topologically reliable approximation of trimmed polynomial surface patches. Design Laboratory Memorandum 97-4, MIT, Department of Ocean Engineering, Cambridge, MA, November 1997.

[9] W. Cho, T. Maekawa, N. M. Patrikalakis, and J. Peraire. Robust tesselation of trimmed rational B-spline surface patches. In F.-E. Wolter and N. M. Patrikalakis, editors, *Proceedings of Computer Graphics International, CGI '98, June 1998*, pages 543–555, Los Alamitos, CA, 1998. IEEE Computer Society.

[10] W. A. Cook. Body oriented (natural) coordinates for generating three dimensional meshes. *International Journal for Numerical Methods in Engineering*, 8:27–43, 1974.

[11] W. J. Gordon. Construction of curvilinear coordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, 7:461–477, 1973.

[12] K. Ho-Le. Finite element mesh generation methods: A review and classification. *Computer-Aided Design*, 20(1):27–38, 1988.

[13] D. A. Lindholm. Automatic triangular mesh generation on surfaces of polyhedra. *IEEE Trans. Mag.*, 19(6):1539–1542, 1983.

[14] A. O. Moscardini, B. A. Lewis, and M. Cross. AGTHOM-Automatic generation of triangular and higher order meshes. *International Journal for Numerical Methods in Engineering*, 19:1331–1353, 1983.

[15] V. P. Nguyen. Automatic mesh generation with tetrahedron elements. *International Journal for Numerical Methods in Engineering*, 18:273–280, 1982.

[16] M. C. Rivara. Algorithm for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal for Numerical Methods in Engineering*, 20:745–756, 1984.

[17] E. Sadek. A scheme for the automatic generation of triangular finite elements. *International Journal for Numerical Methods in Engineering*, 15:1813–1822, 1980.

[18] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[19] D. F. Watson. Computing the n-dimensional delaunay tessellation with applications to voronoi polytopes. *The Computer Journal*, 24(2):167–172, May 1981.

[20] T. C. Woo. An algorithm for generating solid elements in objects with holes. *Computers and Structures*, 8(2):333–342, 1984.

[21] M. A. Yerry and M. S. Shephard. A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics and Applications*, (1-2):39–46, 1983.

[22] M. A. Yerry and M. S. Shephard. Automatic three-dimensional mesh generation by the modified octree technique. *International Journal of Numerical Methods in Engineering*, 20:1965–1990, 1984.

[23] O. C. Zienkiewicz and D. V. Phillips. An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates. *International Journal for Numerical Methods in Engineering*, 7:461–477, 1971.