

Scientific Computing WS 2017/2018

Lecture 13

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

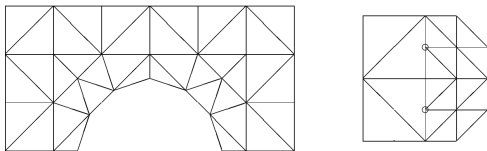
Meshes

- ▶ Regard boundary value problems for PDEs in a finite domain $\Omega \subset \mathbb{R}^n$
- ▶ Assume the domain is polygonal, its boundary $\partial\Omega$ is the union of a finite number of subsets of hyperplanes in \mathbb{R}^n (line segments for $d = 2$, planar polygons for $d = 3$)
- ▶ A mesh (grid) is a subdivision Ω into a finite number of elementary closed (polygonal) subsets $T_1 \dots T_M$.
- ▶ Mostly, the elementary shapes are triangles or quadrilaterals ($d = 2$) or tetrahedra or cuboids ($d = 3$)
- ▶ During this course: focus on $d = 2$, triangles
⇒ mesh = grid = triangulation

(FEM)-Admissible meshes

Definition: A grid is FEM-admissible if

- (i) $\bar{\Omega} = \cup_{m=1}^M T_m$
- (ii) If $T_m \cap T_n$ consists of exactly one point, then this point is a common vertex of T_m and T_n .
- (iii) If for $m \neq n$, $T_m \cap T_n$ consists of more than one point, then $T_m \cap T_n$ is a common edge (or a common facet for $d = 3$) of T_m and T_n .



Source: Braess, FEM

Left: admissible mesh. Right: mesh with hanging nodes

Acute + weakly acute triangulations

Definition A triangulation of a domain Ω is

- ▶ acute, if all interior angles of all triangles are less than $\frac{\pi}{2}$,
- ▶ weakly acute, if all interior angles of all triangles are less than or equal to $\frac{\pi}{2}$.

Triangulation methods

- ▶ Geometrically most flexible
- ▶ Basis als for more general methods of subdivision into quadrilaterals
- ▶ Problem seems to be simple only at the first glance . . .
- ▶ Here, we will discuss Delaunay triangulations, which have a number of interesting properties when it comes to PDE discretizations

Voronoi diagrams

Definition Let $\mathbf{p}, \mathbf{q} \in \mathbb{R}^d$. The set of points $H_{\mathbf{p}\mathbf{q}} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \mathbf{p}\| \leq \|\mathbf{x} - \mathbf{q}\|\}$ is the *half space* of points \mathbf{x} closer to \mathbf{p} than to \mathbf{q} .

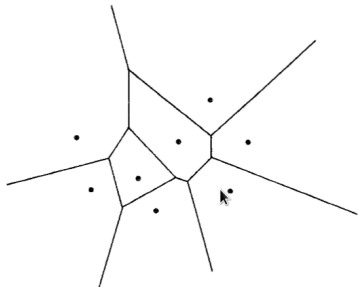
Definition Given a finite set of points $S \subset \mathbb{R}^d$, the *Voronoi region* (*Voronoi cell*) of a point $\mathbf{p} \in S$ is the set of points \mathbf{x} closer to \mathbf{p} than to any other point $\mathbf{q} \in S$:

$$V_{\mathbf{p}} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \mathbf{p}\| \leq \|\mathbf{x} - \mathbf{q}\| \forall \mathbf{q} \in S\}$$

The *Voronoi diagram* of S is the collection of the Voronoi regions of the points of S .

Voronoi diagrams II

- ▶ the Voronoi diagram subdivides the whole space into “nearest neighbor” regions
- ▶ Being intersections of half planes, the Voronoi regions are convex sets

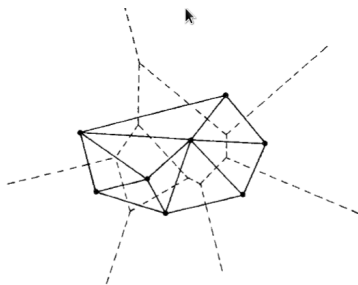


Voronoi diagram of 8 points in the plane

(H. Si)

Delaunay triangulation

- ▶ Assume that the points of S are in *general position*, i.e. no $d + 2$ points of S are on one sphere (in 2D: no 4 points on one circle)
- ▶ Connect each pair of points whose Voronoi regions share a common edge with a line
- ▶ \Rightarrow *Delaunay triangulation* of the convex hull of S



Delaunay triangulation of the convex hull of 8 points in the plane

(H. Si)

Delaunay triangulation II

- ▶ The circumsphere (circumcircle in 2D) of a d -dimensional simplex is the unique sphere containing all vertices of the simplex
- ▶ The circumball (circumdisc in 2D) of a simplex is the unique (open) ball which has the circumsphere of the simplex as boundary

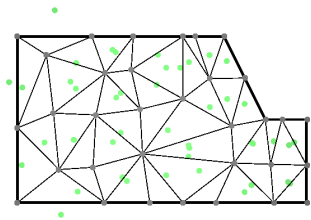
Definition A triangulation of the convex hull of a point set S has the *Delaunay property* if each simplex (triangle) of the triangulation is Delaunay, i.e. it is empty wrt. S , i.e. it does not contain any points of S .

- ▶ The Delaunay triangulation of a point set S , where all points are in general position is unique
- ▶ Otherwise there is an ambiguity - if e.g. 4 points are one circle, there are two ways to connect them resulting in Delaunay triangles

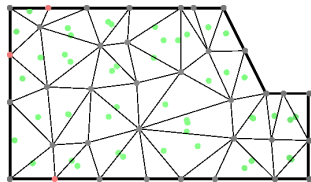
Boundary conforming Delaunay triangulations

Definition: An admissible triangulation of a polygonal Domain $\Omega \subset \mathbb{R}^d$ has the boundary conforming Delaunay property if

- (i) All simplices are Delaunay
- (ii) All boundary simplices (edges in 2D, facets in 3d) have the Gabriel property, i.e. their minimal circumdisks are empty
 - ▶ Equivalent definition in 2D: sum of angles opposite to interior edges $\leq \pi$, angle opposite to boundary edge $\leq \frac{\pi}{2}$
 - ▶ Creation of boundary conforming Delaunay triangulation description may involve insertion of Steiner points at the boundary



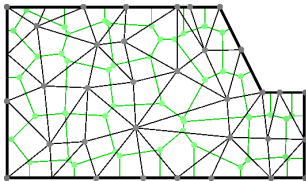
Delaunay grid of Ω



Boundary conforming Delaunay grid of Ω

Domain blendend Voronoi cells

- ▶ For Boundary conforming Delaunay triangulations, the intersection of the Voronoi diagram with the domain yields a well defined dual subdivision which can be used for finite volume discretizations



Boundary conforming Delaunay triangulations II

- ▶ Weakly acute triangulations are boundary conforming Delaunay, but not vice versa!
- ▶ Working with weakly acute triangulations for general polygonal domains is unrealistic, especially in 3D
- ▶ For boundary conforming Delaunay triangulations of polygonal domains there are algorithms with mathematical termination proofs valid in many relevant cases
 - ▶ Software in 2D: Triangle by J.R.Shewchuk
<https://www.cs.cmu.edu/~quake/triangle.html>
 - ▶ Software in 3D: TetGen by H. Si <http://tetgen.org>
 - ▶ polygonal geometry description
 - ▶ automatic insertion of points according to given mesh size criteria
 - ▶ accounting for interior boundaries
 - ▶ local mesh size control for a priori refinement
 - ▶ quality control
 - ▶ standalone executable & library

The Triangle mesh generator

- ▶ Free for non-commercial use
- ▶ By J.R.Shewchuk, Berkeley
- ▶ Distributed with numcxx

```
$ triangle --help
```

- ▶ Accompanied by showme program to visualize grids
- ▶ Triangle as a standalone program is controlled by certain flags, reads input from disk, writes output to disk
- ▶ Triangle as a library is controlled by the same flags, but takes input as double* and int* arrays, and creates output in the same form

Convex hulls of point sets with Triangle

```
$ triangle --help
...
-c Creates segments on the convex hull of the triangulation.  If you
  are triangulating a vertex set, this switch causes a .poly file to
  be written, containing all edges of the convex hull.
...
-v Outputs the Voronoi diagram associated with the triangulation.
...
-V Verbose: Gives detailed information about what Triangle is doing.
...
$ triangle -Vvc hello.node
$ ls hello.*
hello.node      # input
hello.1.node    # output, nodes of the Delaunay triangulation
hello.1.ele     # output, triangles of the Delaunay triangulation
hello.1.poly    # output, Edges of the convex hull of the point set
hello.1.v.node  # output, nodes of Voronoi diagram
hello.1.v.edge  # output, edges of Voronoi diagram
```

- ▶ .node files contain lists of points (possibly with attributes)
- ▶ .poly files contain lists of polygons (possibly with attributes)
- ▶ .edge files contain lists of edges (possibly with attributes)
- ▶ .ele files contain lists of triangles (possibly with attributes)

Discretization of a domain with Triangle

```
$ triangle --help
-p Reads a Planar Straight Line Graph (.poly file), which can specify
  vertices, segments, holes, regional attributes, and regional area
  constraints. Generates a constrained Delaunay triangulation (CDT)
  fitting the input; or, if -s, -q, -a, or -u is used, a conforming
  constrained Delaunay triangulation (CCDT).
...
-a Imposes a maximum triangle area. If a number follows the 'a', no
  triangle is generated whose area is larger than that number.
...
-q Quality mesh generation by Delaunay refinement. Adds vertices
  to the mesh to ensure that all angles are between 20 and 140 degrees.
  An alternative bound on the minimum angle, replacing 20 degrees, may
  be specified after the 'q'.
...
-D Conforming Delaunay triangulation: use this switch if you want to
  ensure that all the triangles in the mesh are Delaunay, and not
  merely constrained Delaunay; or if you want to ensure that all the
  Voronoi vertices lie within the triangulation. (Some finite volume
  methods have this requirement.)
...
$ triangle -Vp hello.poly
$ ls hello.*
hello.1.node      # output, nodes of the triangulation
hello.1.ele      # output, triangles of the triangulation
hello.1.poly      # output, boundary edges
```

Triangle in numcxx

- ▶ In addition to nodes, triangles and boundary edges we need a region attribute for each triangle (for different data in different regions) and a boundary region attribute for each boundary edge (for different boundary conditions)
- ▶ Triangle handles these attributes
- ▶ `class numcxx::Geometry` → `class numcxx::SimpleGrid`
- ▶ `class numcxx::SimpleGrid` is ready for use in finite element and finite volume methods
 - ▶ points
 - ▶ cells
 - ▶ boundary faces
 - ▶ cell and boundary regions

Conjugate gradients in numxx

- ▶ Header `netlib/netlib.hxx`, `netlib/cg.h`
- ▶ Example `examples/10-numcxx-basics/17-numcxx-cg.cxx`
- ▶ This is a typical "duck typing" API: the implementation is parametrized by a number of classes which are expected to have certain methods:
 - ▶ Matrix: matrix-vector multiplication $A*x$
 - ▶ Vector: constructor $v(n)$, linear algebra, dot product $\text{dot}(u,v)$, norm $\text{norm}(u)$
 - ▶ Preconditioner: Solution of preconditioning system $M.\text{solve}(u,f)$
- ▶ "When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck."
– James Whitcomb Riley
- ▶ Similar approach has been used for vectors in `vtkfig`:
 - ▶ API works with `std::vector`
 - ▶ API works with `numcxx::darray`