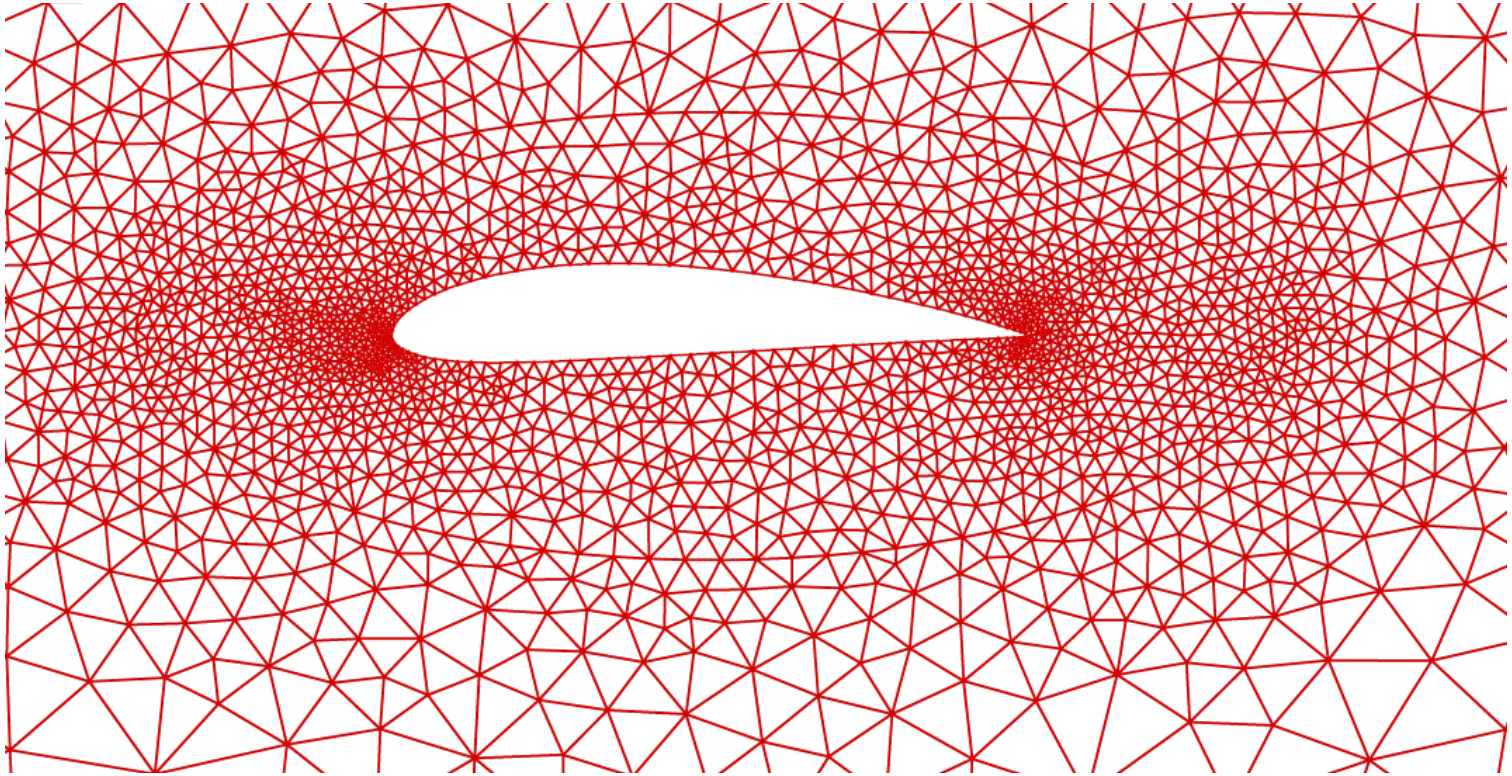
NATIONAL TECHNICAL UNIVERSITY OF ATHENS
School of Mechanical Engineering
Lab. Of Thermal Turbomachines
Parallel CFD & Optimization Unit (PCOpt/NTUA)

Unstructured Grids

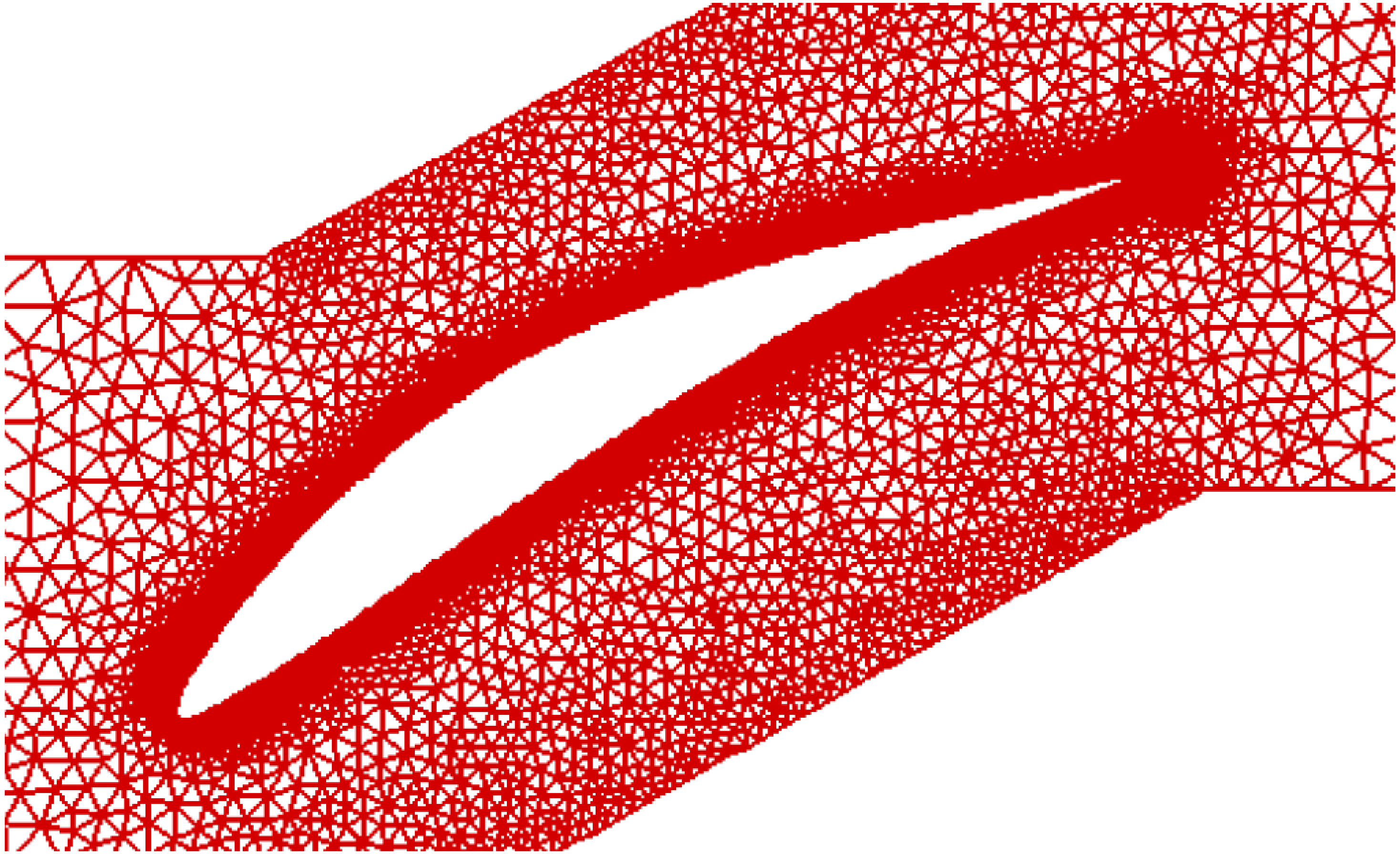
Kyriakos C. GIANNAKOGLOU, Professor NTUA
kgianna@mail.ntua.gr
http://velos0.ltt.mech.ntua.gr/research

Unstructured Meshes/Grids

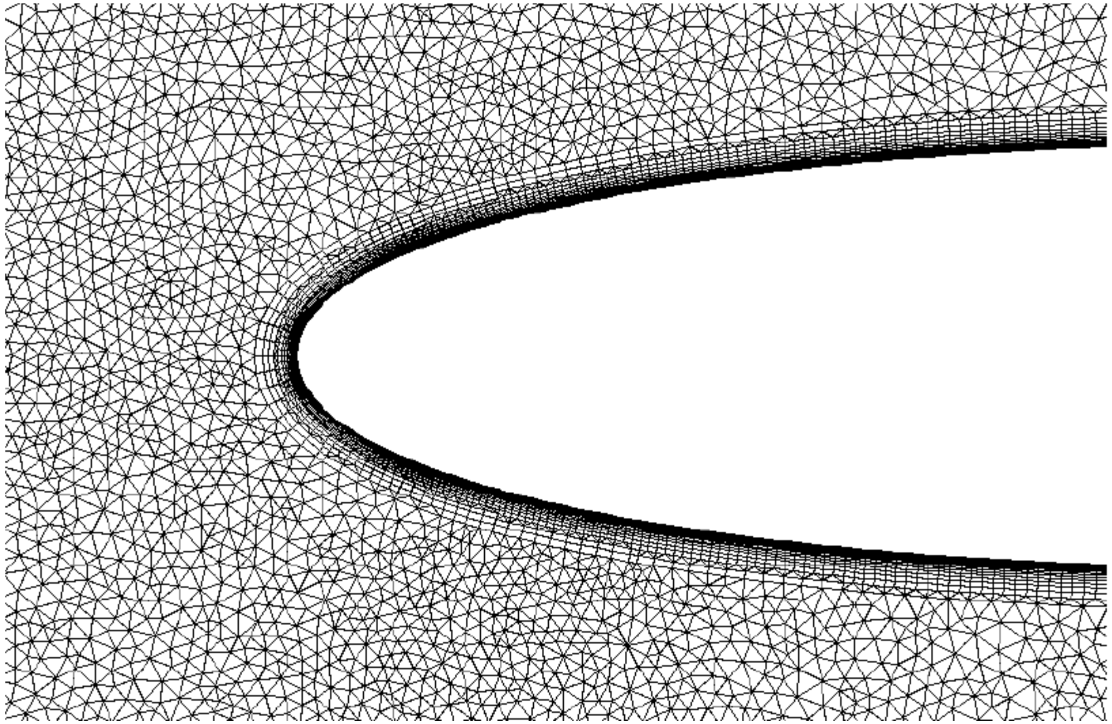
Flexibility vs. Accuracy

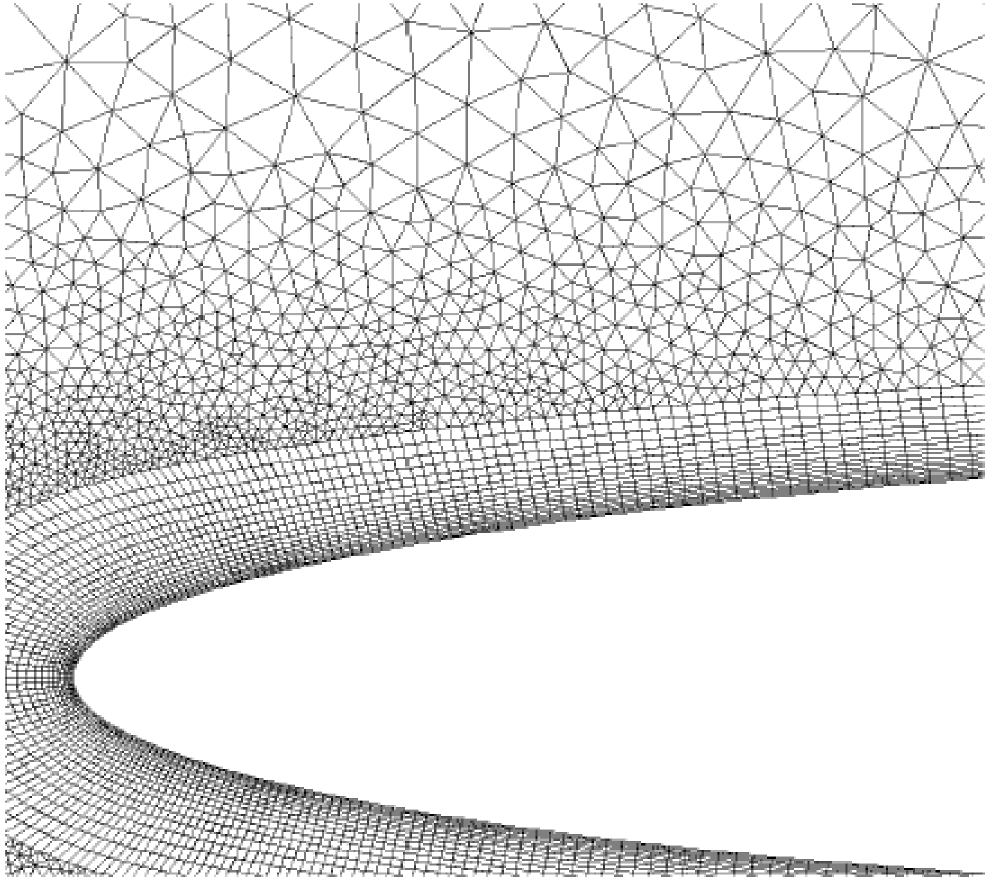


Unstructured Meshes/Grids

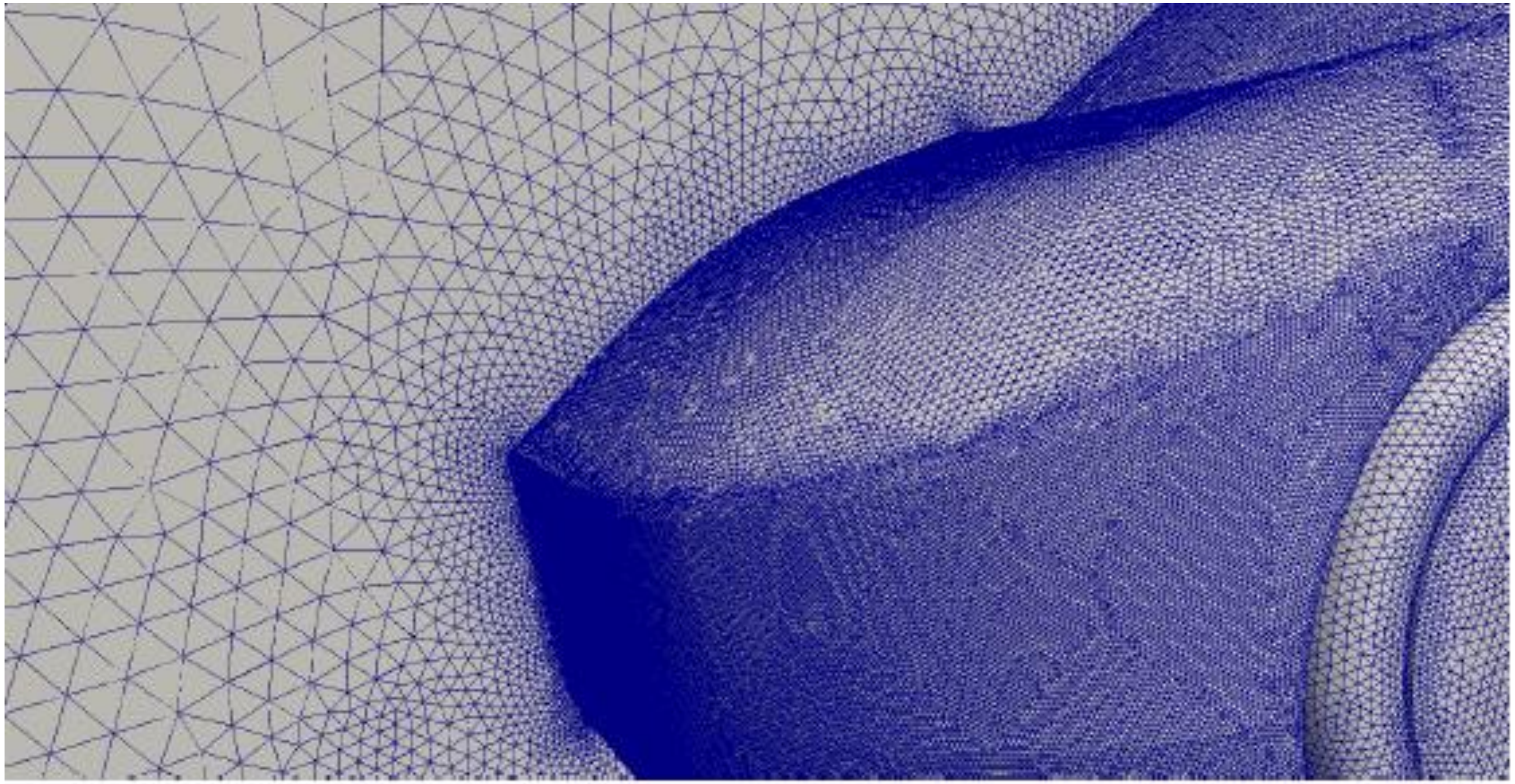


Unstructured Meshes/Grids





Unstructured Meshes/Grids



Structured → Unstructured Grids

A 2D **structured** grid (of quadrilaterals) consists of a set of coordinates and connectivities that directly map onto the elements of a matrix. Thus, neighboring points in the mesh (in the physical space) correspond to the neighboring elements in the grid matrix. Indices (i,j) naturally identify the two families of grid lines.

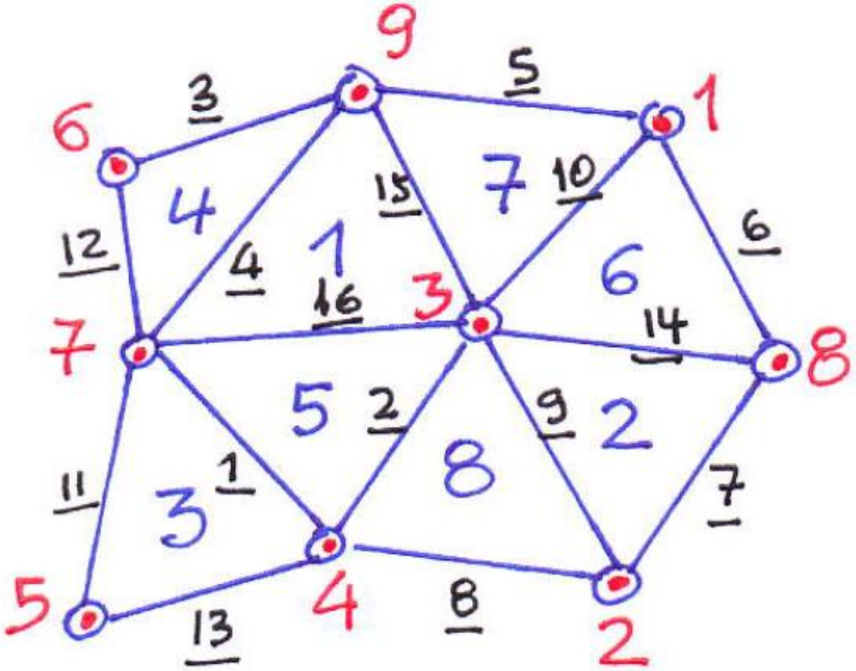
For a 2D/3D **unstructured** grid, additional information has to be provided. For each grid point, the connection with other grid points must explicitly be defined; this information is included in the connectivity matrix.

The organization of data structure in a code depends on several factors. Among them: (a) the type of operations we wish to perform on the data and (b) the available computer memory. Many data structure types can be found in the literature.

Data Structures

(2D) Triangular Meshes/Grids:

Triangle→Edge→Node



8 (triangles)

$$\left.\begin{array}{ccc} 7 & 3 & 9 \\ 3 & 2 & 8 \\ 5 & 4 & 7 \\ \ldots & & \end{array}\right\} 8 \text{ "lines"}$$

9 (nodes)

$$\left.\begin{array}{cc} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \\ x_9 & y_9 \end{array}\right\} 9 \text{ "lines"}$$

Data Structures (& Connectivities)

(2D) Triangular Meshes/Grids:

Each Triangle: IDs of its Nodes, IDs of its Edges, IDs of surrounding Triangles

Each Edge: IDs of its defining Nodes, number & ID list of adjacent Triangles

Each node: number & ID list of Edges emanating from this Node, number & ID list of Nodes connected with this Node by an Edge, number & ID list of Triangles sharing this Node

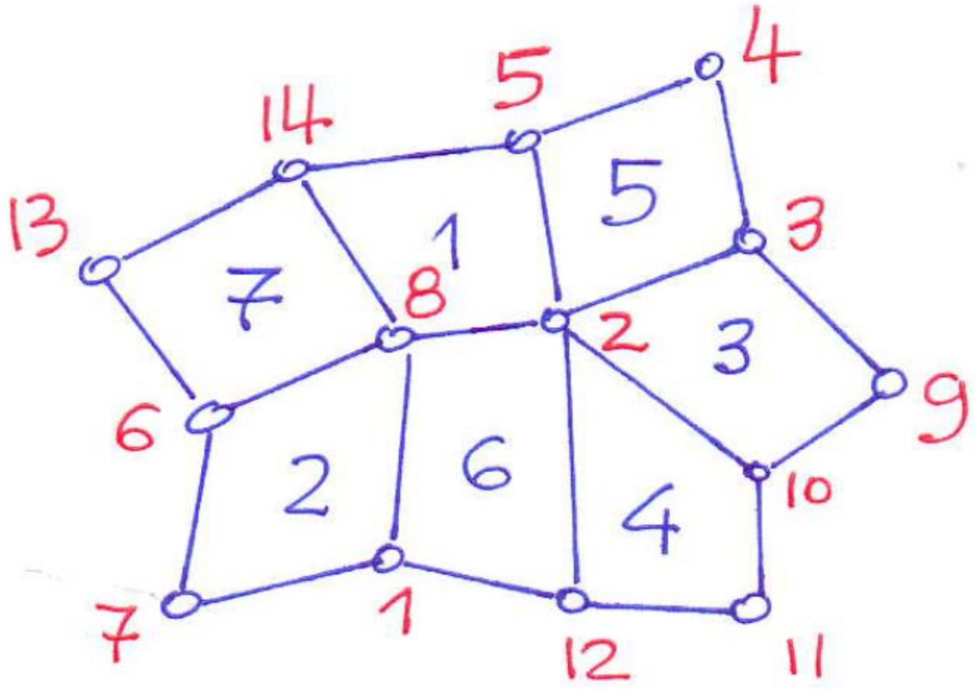
Plus: Geometrical Quantities (Areas of Triangles of Finite Volumes), Lengths of Edges, Normal etc along the Boundary Edges….

Question: Retrieval of the contents of records?

Data Structures

(2D) Quadrilateral Meshes/Grids:

Quad→Edge→Node



7 (quads)

14	8	2	5
6	7	1	8
2	10	9	3
...			

7 "lines"

14 (nodes)

x_1 y_1

x_2 y_2

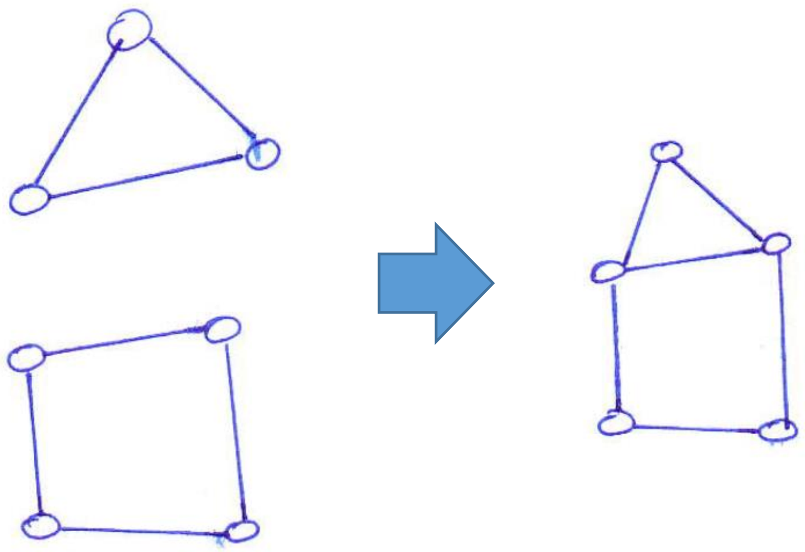
⋮

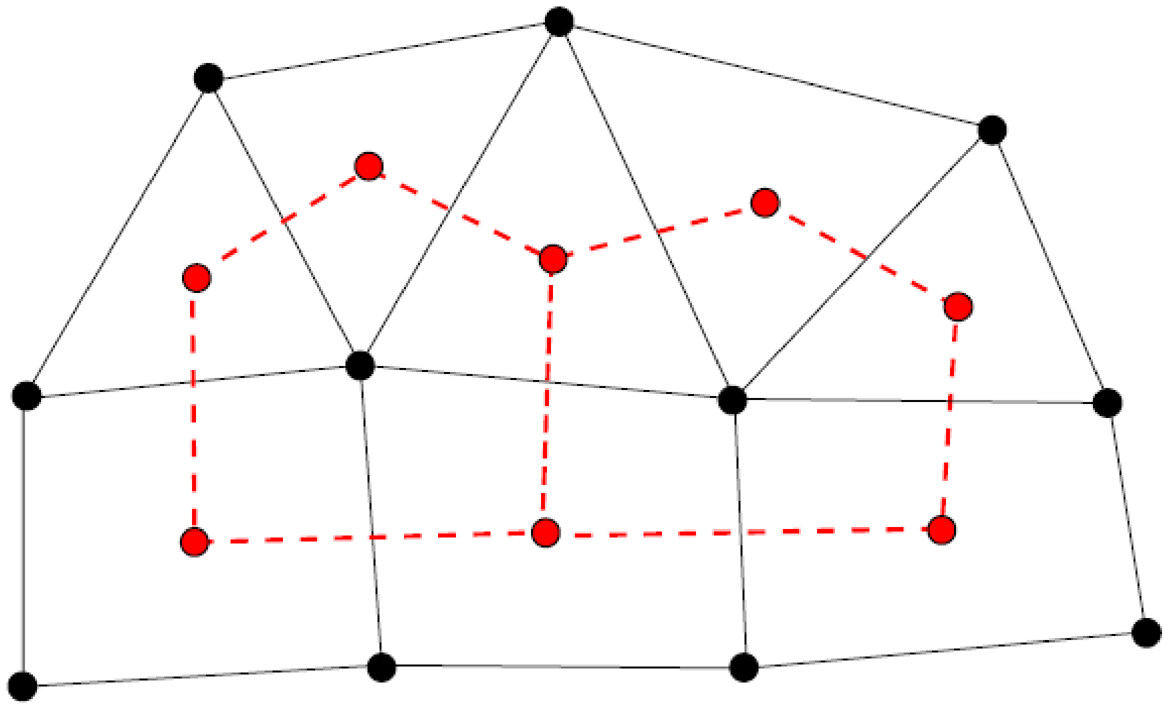
x_{14} y_{14}

14 "lines"

Unstructured Meshes/Grids

Hybrid 2D Meshes/Grids (triangular & Quadrilateral Elements):

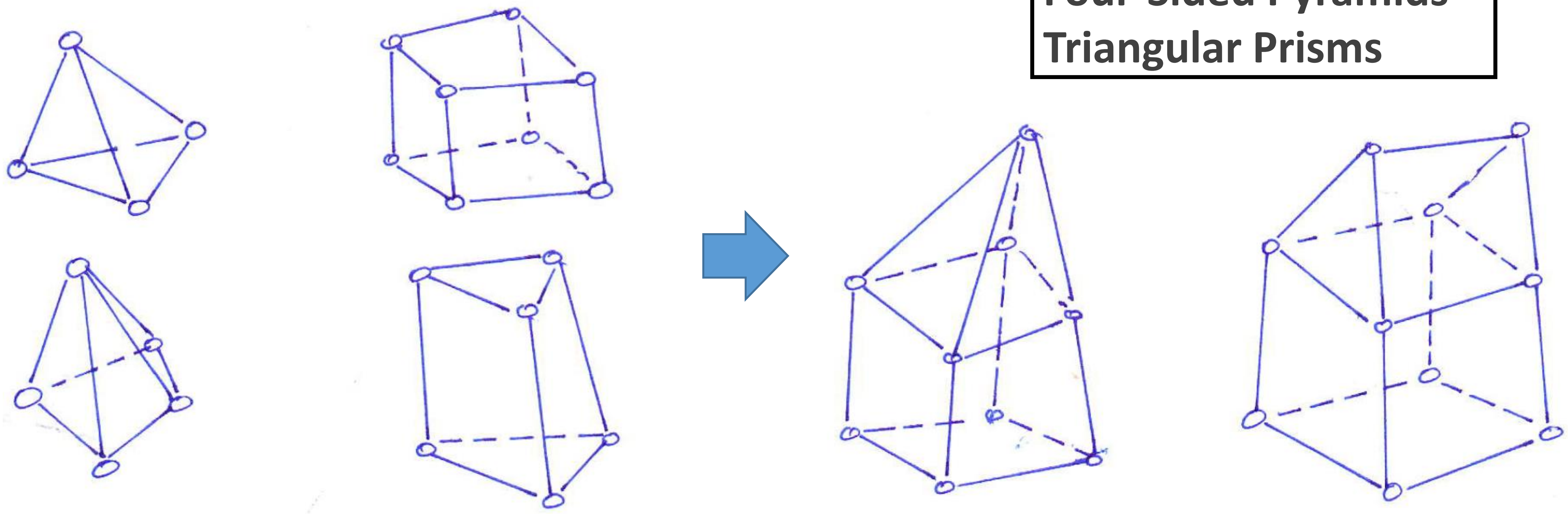




Unstructured Meshes/Grids

Hybrid 3D Meshes/Grids

Tetrahedra
Hexahedra
Four-Sided Pyramids
Triangular Prisms

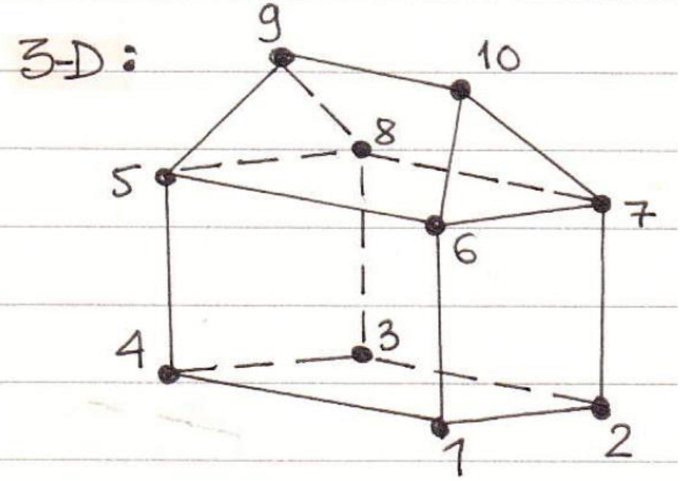


Graphs & Meshes

Euler's Formula for a SINGLE Polyhedron:

n_e = Number of Edges
n_v = Number of Vertices
n_f = Number of Faces

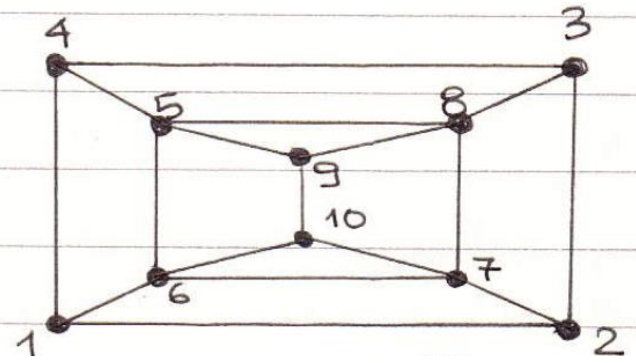
$$\boxed{n_f = n_e - n_v + 2}$$



$n_v = 10$, $n_e = 17$, $n_f = 9$

$9 = 17 - 10 + 2$ ✓

Development on a plane



Includes the External Face=Infinity

Graphs & Meshes

Euler's Formula … in 2D (planar meshes):

Eliminate infinite face → Introduce boundary segments

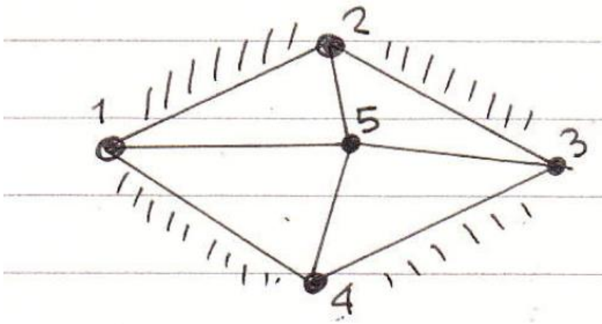
n_e = Number of Edges

n_v = Number of Vertices

n_f = Number of Faces

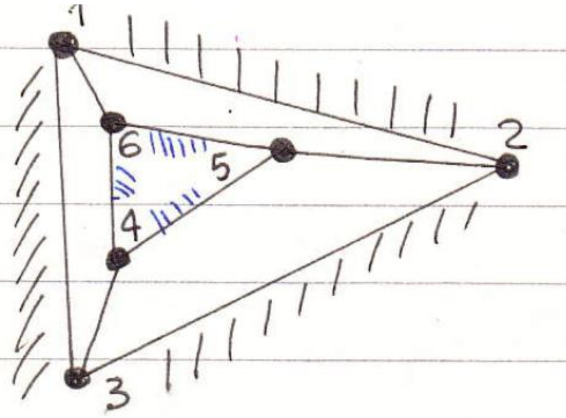
n_h = Number of Internal Closed Loops ("holes")

$$\boxed{n_f + n_v = n_e + 1 - n_h}$$



$n_f = 4,\ n_e = 8,\ n_v = 5,\ n_h = \emptyset$

$4 + 5 = 8 + 1 + \emptyset$ ✓



$n_f = 3,\ n_e = 9,\ n_v = 6,\ n_h = 1$

$3 + 6 = 9 + 1 - 1$ ✓

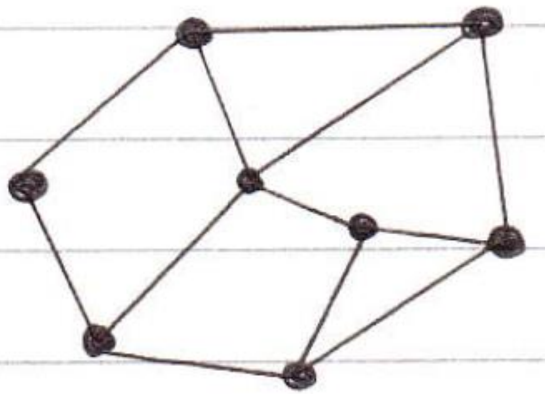
Graphs & Meshes

Euler's Formula … in 2D (planar meshes):

n_{ie} = Number of Interior Edges
n_{be} = Number of Boundary Edges
$n_f^{(i)}$ = Number of Faces with i Edges

$$\boxed{2n_{ie} + n_{be} = \sum_{\substack{\forall i \\ i \geq 3}} i\, n_f^{(i)}}$$

$$n_{ie} + n_{be} = n_e$$



$n_{ie} = 6$ | $n_f^3 = 2$ **Triangles**
$n_{be} = 6$ | $n_f^4 = 3$ **Quadrilaterals**

$12 + 6 = 3 \times 2 + 4 \times 3$ ✓

Graphs & Meshes

Euler's Formula … in 2D (planar meshes):

In case ONLY a 2D Mesh with Triangular Elements ($n_f = n_f^{(3)}$)

1/ $2n_{ie} + n_{be} = 2n_e - n_{be} = 3n_f$

2/ $n_e = n_f + n_\vartheta - 1 + n_h$

$$\boxed{n_f = 2n_\vartheta - n_{be} - 2 + 2n_h}$$

or

$n_e = \frac{3}{2} n_f + \frac{1}{2} n_{be}$

$$\boxed{n_e = 3n_\vartheta - n_{be} - 3 + 3n_h}$$

Graphs & Meshes

Euler's Formula … in 2D (planar meshes):

Practically:

MAXNOD=….
MAXFAC=2*MAXNOD
MAXSEG=3*MAXNOD

Graphs & Meshes

R^d Generalization of the Euler Theorem:

A polytope P in R^d contains k-faces for k in {-1,0,1,…,d}

(-1-face) → Null Set Face (Convention), $N_{-1}(P)=1$

(d-face) → by definition $N_d(P)=1$

Examples:

2Δ Grid: $N_0 \equiv n_v$, $N_1 \equiv n_e$, $N_2 \equiv n_f$

3Δ Grid: $N_0 \equiv n_v$, $N_1 \equiv n_e$, $N_2 \equiv n_f$, $N_3 \equiv n_\phi$ (volumes)

Euler's Formula in R^d:

$$\boxed{\sum_{k=-1}^{d} (-1)^k N_k(P) = 0}$$

$$\left(-N_{-1} + N_0 - N_1 + N_2 - N_3 + N_4 \cdots = 0\right)$$

Graphs & Meshes

R^d Generalization of the Euler Theorem:

d=3 or 2D Mesh, R^3 :

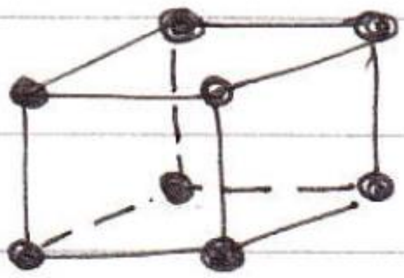
$$\boxed{-1+n_v-n_e+n_f-1=0 \Rightarrow n_f+n_v=n_e+2}$$

d=4 or 3D Mesh, R^4 :

The 3D Euler formula does not account for Boundary Effects

$$-1+n_v-n_e+n_f-n_\phi+1=0 \Rightarrow \boxed{n_f+n_v=n_e+n_\phi}$$

3D →



$$\left.\begin{array}{l} n_v=8 \\ n_e=12 \\ n_f=6 \end{array}\right\} \begin{array}{l} n_\phi=1+1 \ (\text{IN+OUT})=2 \ !!! \\ 8+6=12+2 \ \checkmark \end{array}$$

Graphs & Meshes

Exact Euler's Formula for 3D Tetrahedral Meshes:

$$n(e) = n(\phi) + n(v) + \frac{3}{4} n(f)_{bound} - \frac{1}{2} n(v)_{bound}$$

$$n_f = 2n_\phi + \frac{1}{2} n_{f,bound}$$

$$\boxed{n_e + n_\phi = n_f + n_v + \frac{1}{4} n_{f,bound} - \frac{1}{2} n_{v,bound}}$$

Duality

Median Dual:

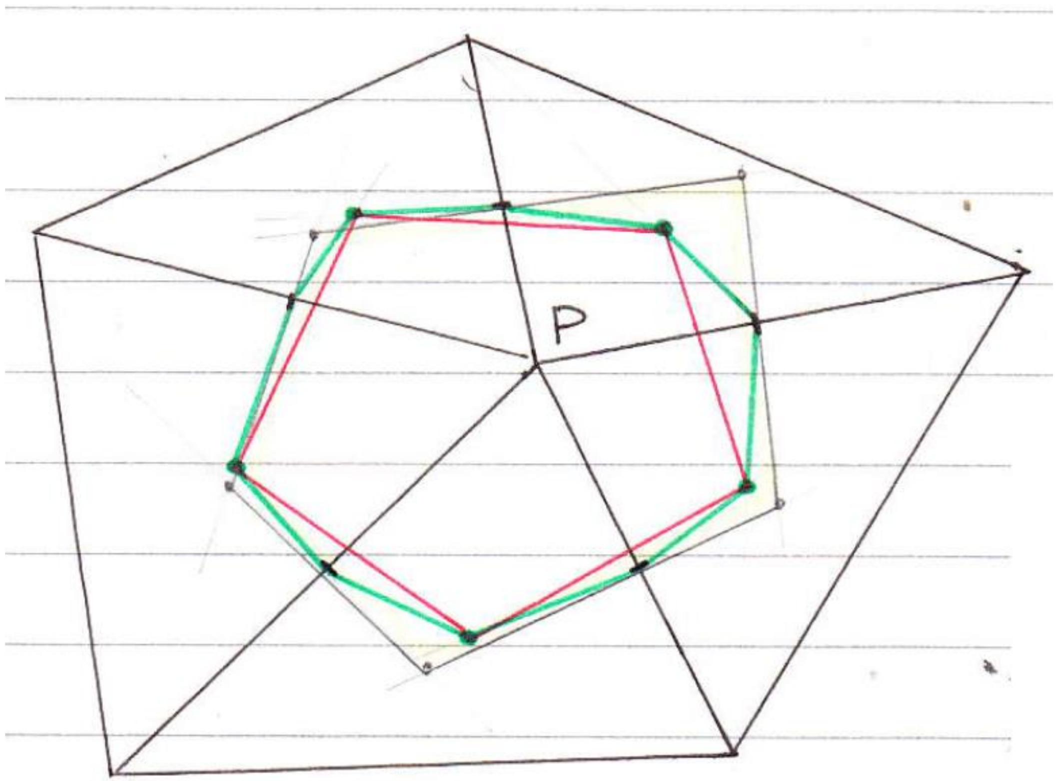
Barycenter → Mid-Center →
Barycenter → Mid-Center → …

Centroid Dual:

Barycenter → Barycenter → …

Dirichlet Region:

Perpendicular bisectors of edges…



Cell-Centered vs. Vertex-Centered Finite Volumes

1) Cèll-centrered 3Δ → Control Cell = Tetrahedron.

Th. Gauss: Fluxes through faces, Balance.

$$\text{Work}_{CC} \propto n_f = 2n_\phi + \frac{1}{2} n_{f,\text{bound}}.$$

2) Vertex-centered 3Δ → Control Cell = Mesh Dual

Th Gauss: Fluxes "along" edges, Balance.

$$\text{Work}_{VC} \propto n_e = n_\phi + n_v + \frac{1}{4} n_{f,\text{bound}} - \frac{1}{2} n_{v,\text{bound}}$$

Cell-Centered vs. Vertex-Centered Finite Volumes

Assume : $n_\phi = \beta n_\vartheta$ ($\beta = 5 \div 7$, Tetrahedral Meshes)

Ignore Boundaries:

$$\frac{Work_{CC}}{Work_{VC}} = \frac{2n_\phi}{n_\phi + n_\vartheta} = \frac{2\beta}{1+\beta} \,, \quad \overset{\text{"}\beta=6\text{"}}{\downarrow} \ \therefore \ = \frac{12}{7} \sim 2\,!$$

Solution Accuracy?

Unstructured Grid Generation Methods

- **Delaunay Triangulation**
- **Advancing Front Method**

Voronoi Diagram & Delaunay Triangulation

Dirichlet Tessellation in a plane pattern of convex regions

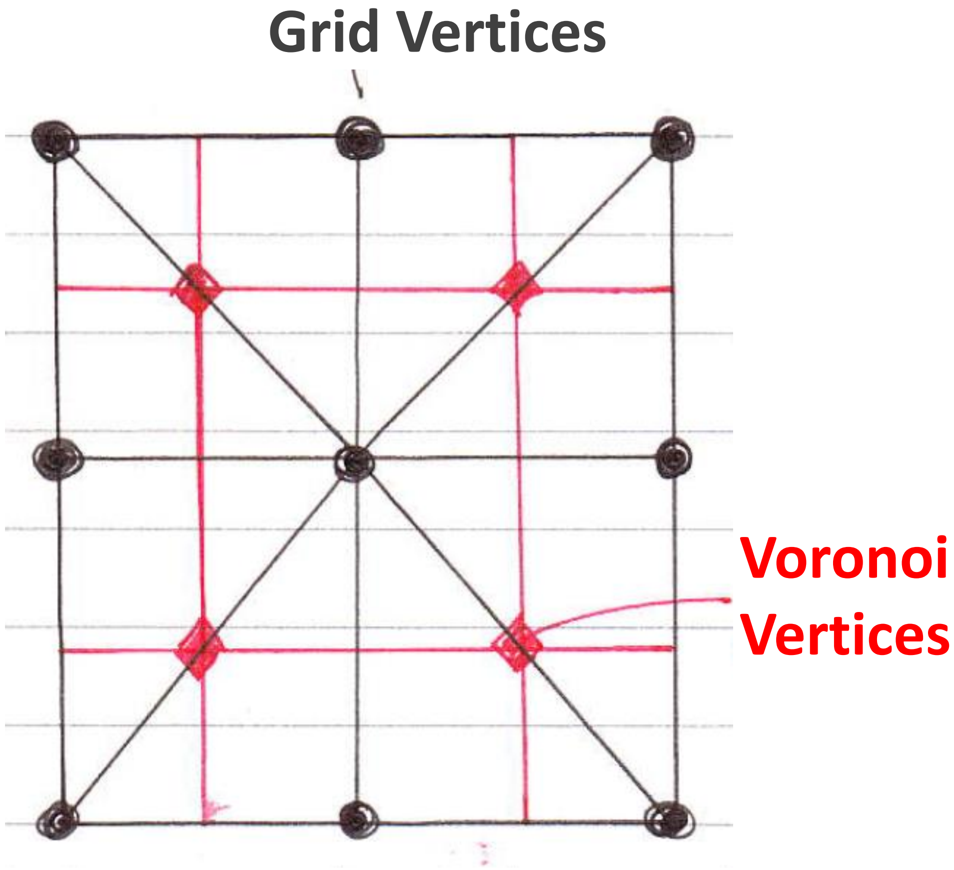
Region closer to each point P

Dirichlet regions=Voronoi regions

Definition: The Delaunay triangulation of a point set S is defined as the dual to the Voronoi Diagram

Delaunay triangulation → Two grid vertices are connected if their Voronoi regions have a common border segment

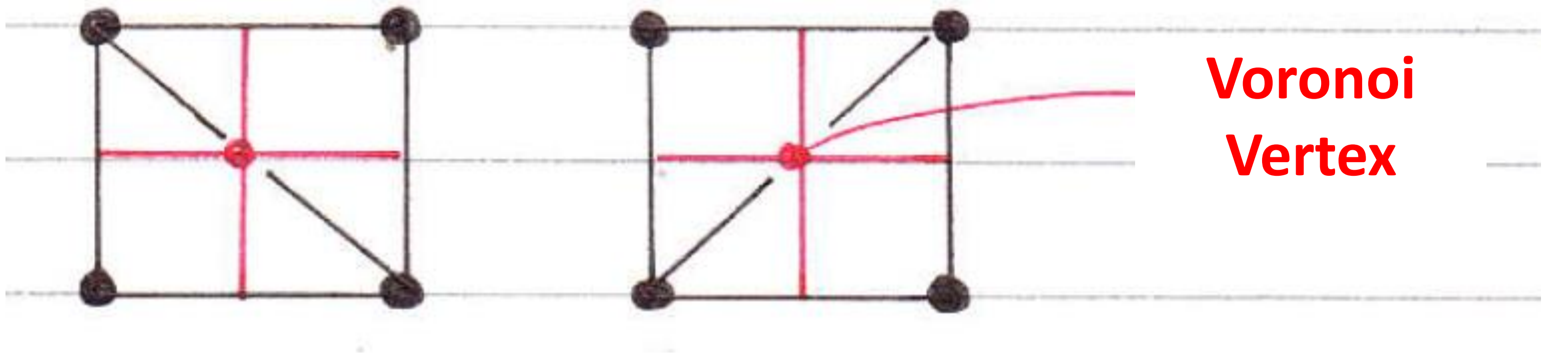
Voronoi vertices <-> Circumcenters of triangles

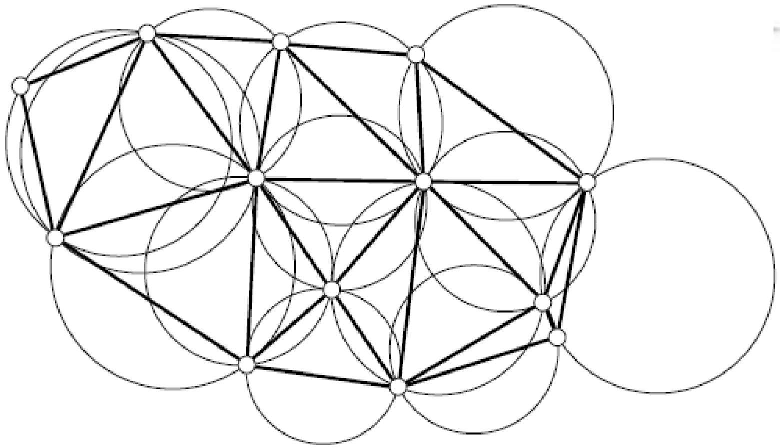


(2D) Delaunay Triangulation

Every point S has a Unique Delaunay Triangulation

Exception four Co-circular vertices

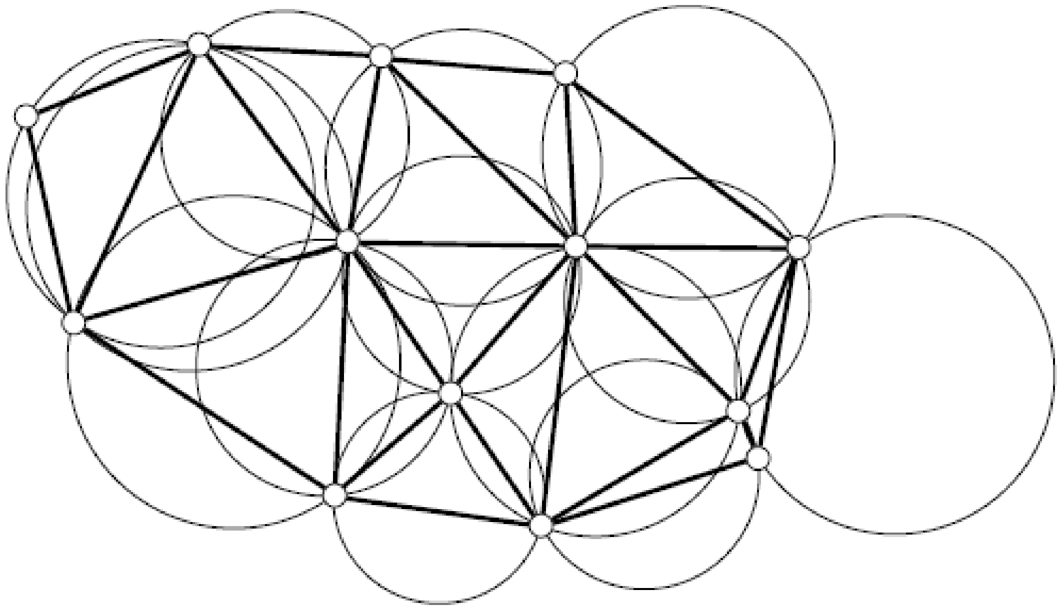


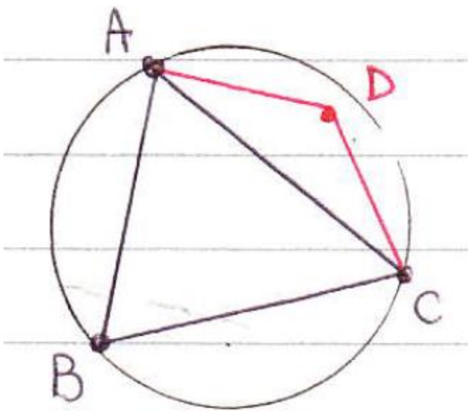


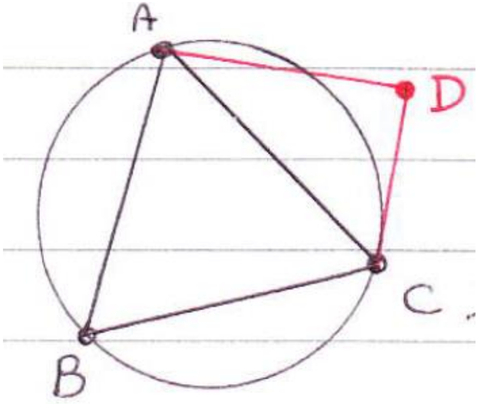
(2D) Delaunay Triangulation

The Empty Circumdisk Property

The open circumdisk (the interior of the circle passing through the three vertices) of every triangle contains no points from the point set.



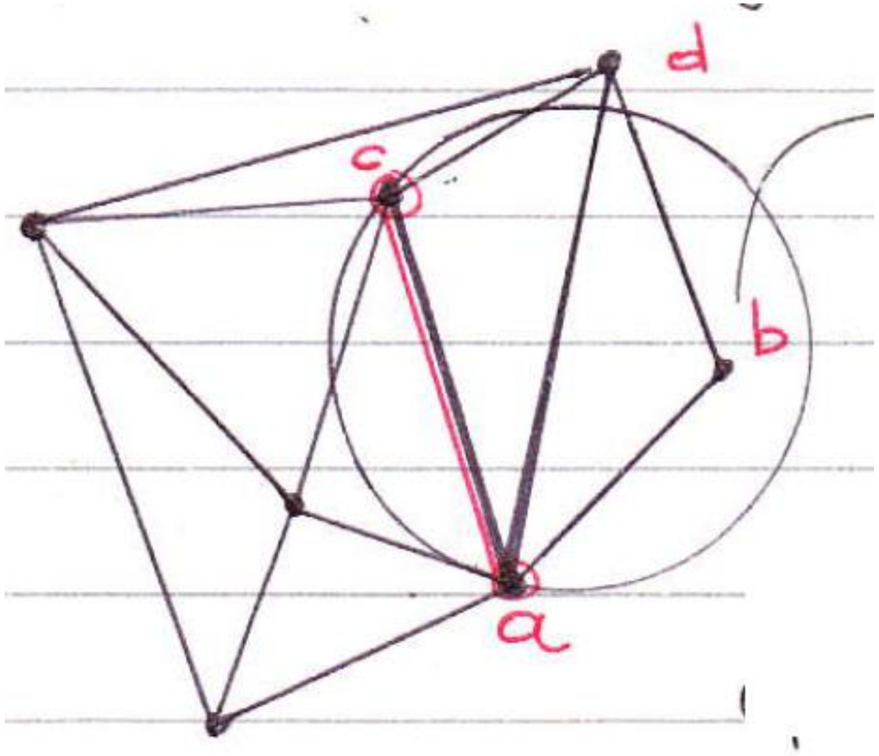




(2D) Delaunay Triangulation

Edge Circle Property

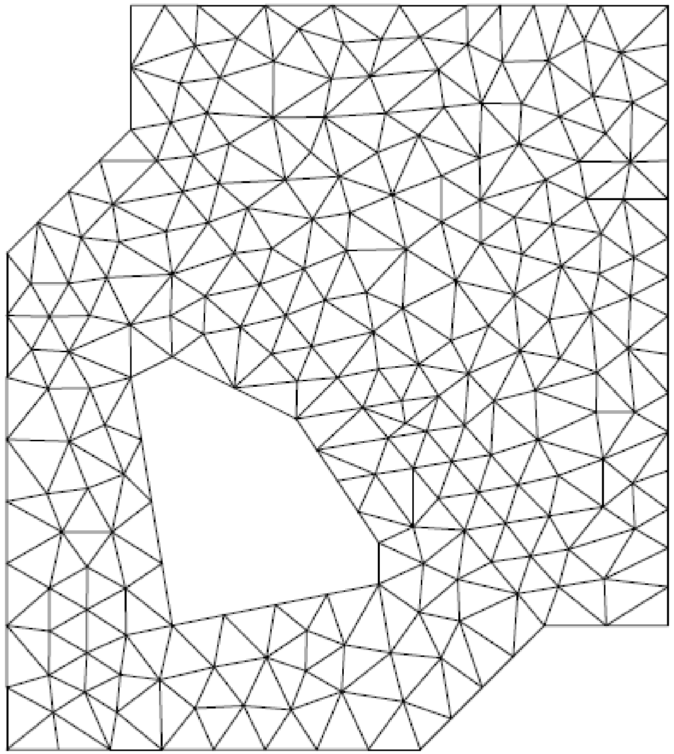
For each and every edge, there exists an empty circle passing through its end-pointswhich contains no points from the point set.



(2D) Delaunay Triangulation

Equiangularity Property

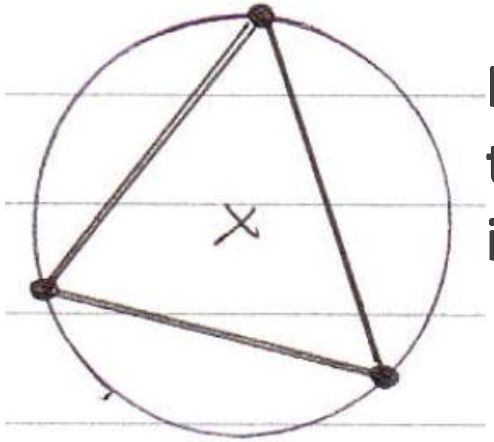
Maximizes the minimum angle of the triangulation



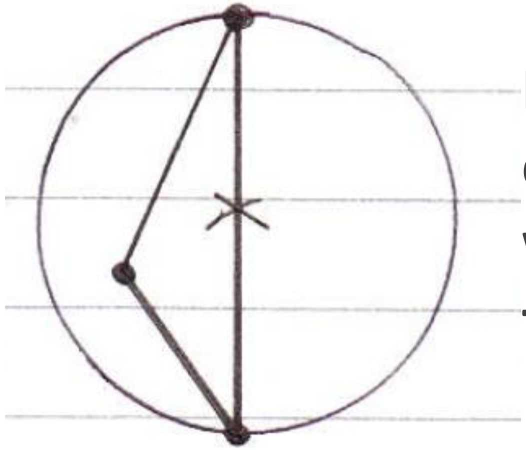
(2D) Delaunay Triangulation

Minimum Containment Circle

Minimizes the maximum containment circle over the entire triangulation (Containment circle is the smallest circle enclosing the 3 vertices of a triangle)



For an acute triangle, the containment circle is just its circumcircle



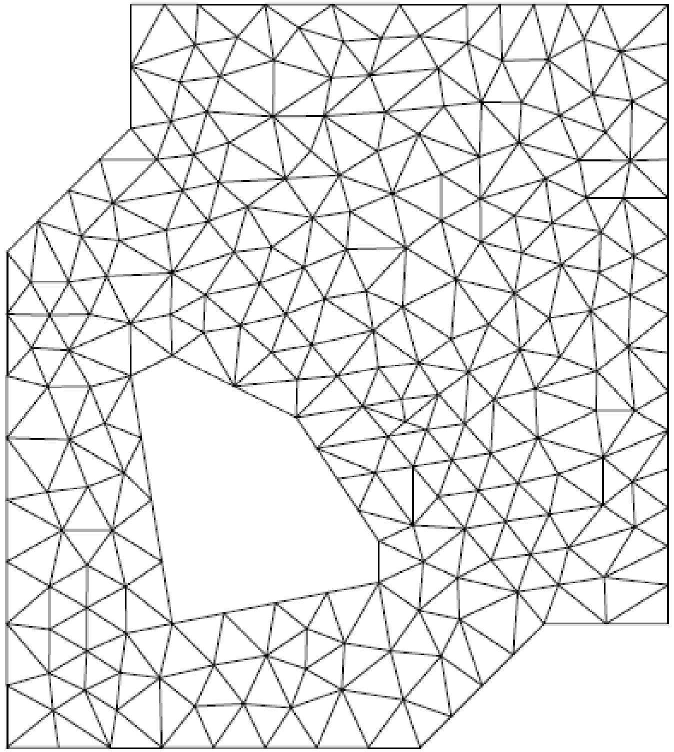
For an obtuse triangle, the containment circle is the circle with the longest side of the triangle as diameter

(2D) Delaunay Triangulation

Minimal Roughness Property

Minimizes the roughness triangulation of arbitrary sets of scattered data - The Sobolev Norm

$$\sum_{TRIANGLES}\left[\left(\frac{\partial F}{\partial x}\right)^2+\left(\frac{\partial F}{\partial y}\right)^2\right]Area_{\Delta}$$



Quality of Triangular Mesh

Measured by:

The deviation of any quality metric from the corresponding value of an equilateral triangle

Area of an equilateral triangle = $$E^* = \frac{\sqrt{3}a^2}{4}$$

Circumcircle radius = $$R = \frac{abc}{4E}$$

Incircle radius = $$r = \frac{2E}{a+b+c}$$

Quality of Triangular Mesh - Quality Metrics:

1) $$Q_1 = \frac{R}{\max(a,\beta,\gamma)} \rightarrow \frac{1}{\sqrt{3}}$$

2) $$Q_2 = \frac{R}{r} \rightarrow 2$$

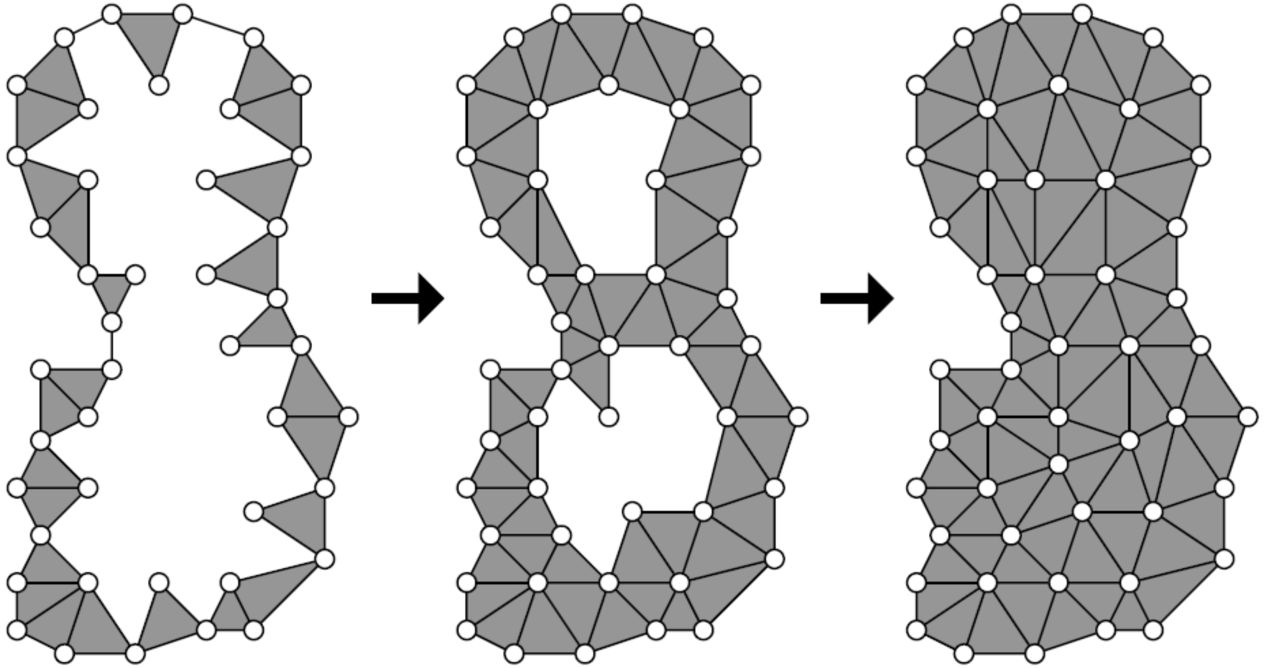
3) $$Q_3 = \frac{\max(a,\beta,\gamma)}{2} \rightarrow 2\sqrt{3}$$

4) $$Q_4 = \frac{\left(\frac{a+\beta+\gamma}{3}\right)^2}{E} \rightarrow \frac{4\sqrt{3}}{3}$$

Ideal values; those of an equilateral triangle

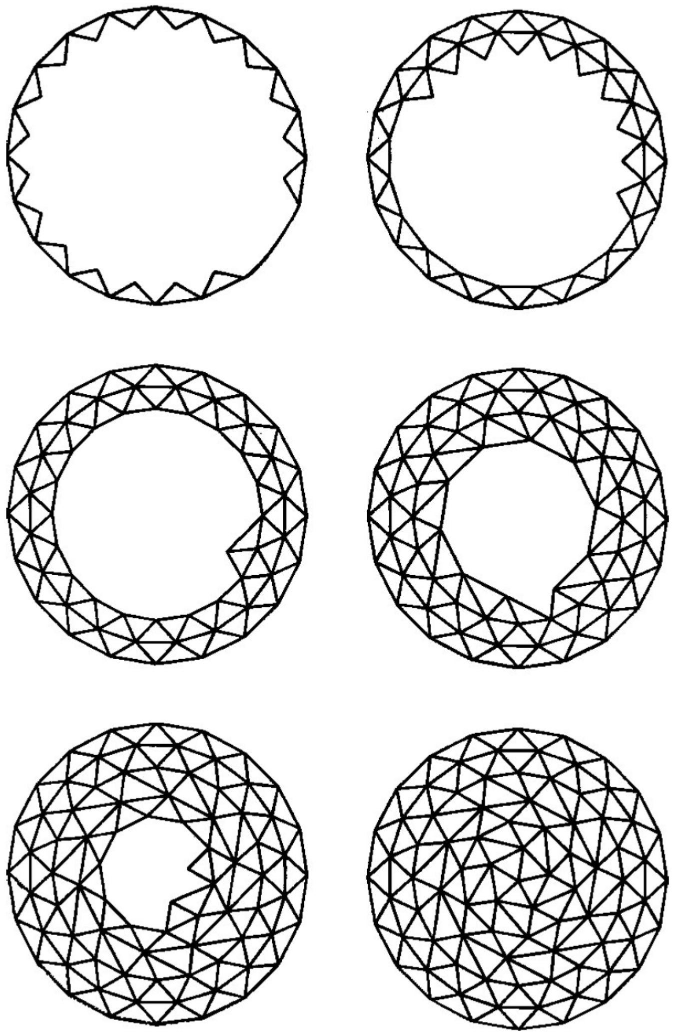
Advancing Front (AF) Technique(s)

The Concept



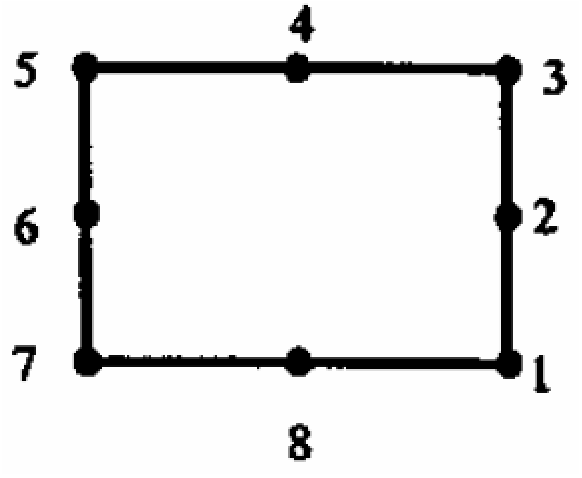
Advancing Front (AF) Technique(s)

The Concept

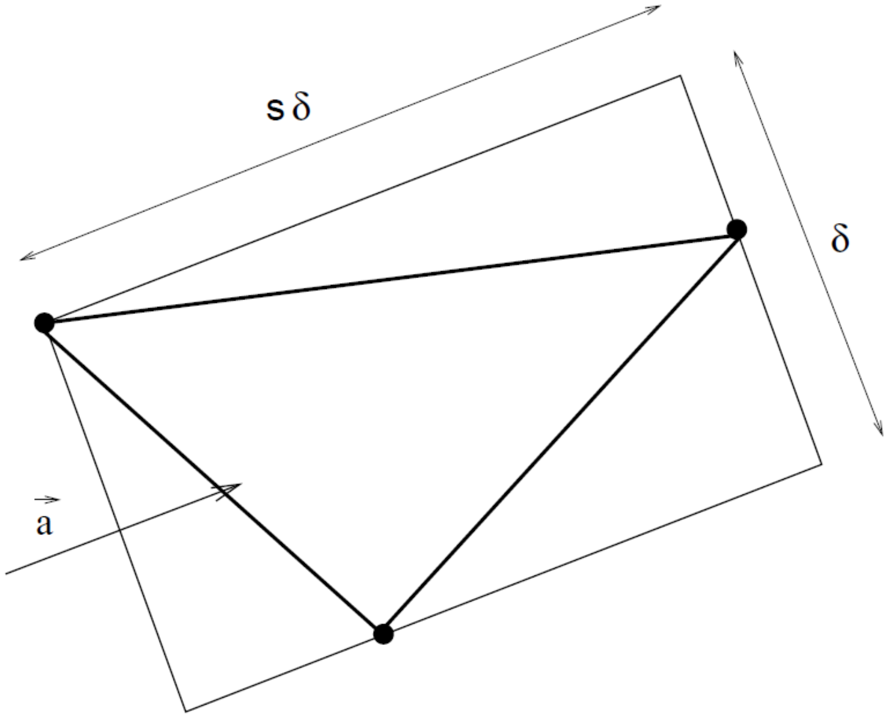


Advancing Front (AF) Technique(s)

Starting point & User Defined Data



$\delta, \; s, \; \vec{a}$



Starting Edge

1	1	2	3	4	5	6	7	8
	2	3	4	5	6	7	8	1

Advancing Front (AF) Technique(s)

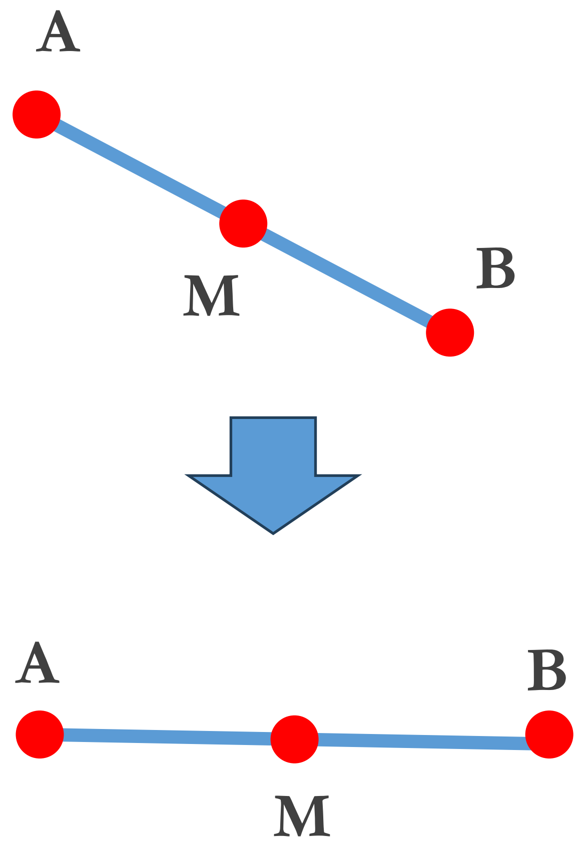
Step 1:

An active edge of the front is chosen (criteria?). In case there are noticeable variations in the values of the auxiliary quantity δ, among the various parts of the domain to be meshed, usually the edge with the smallest length is selected.

Advancing Front (AF) Technique(s)

Step 2:

Let A & B be the endpoints of the selected line segment. The values of δ, s and vector(α) at the mid-point M of AB is selected. All these computations are performed on the background grid (this step is omitted if all three quantities take on fixed values. A transformation (rotation) is performed so that vector (a_M) coincides with the x-axis, while the x-coordinates are multiplied with s_M. In the new transformed space an as much regular as possible tringle should be generated.

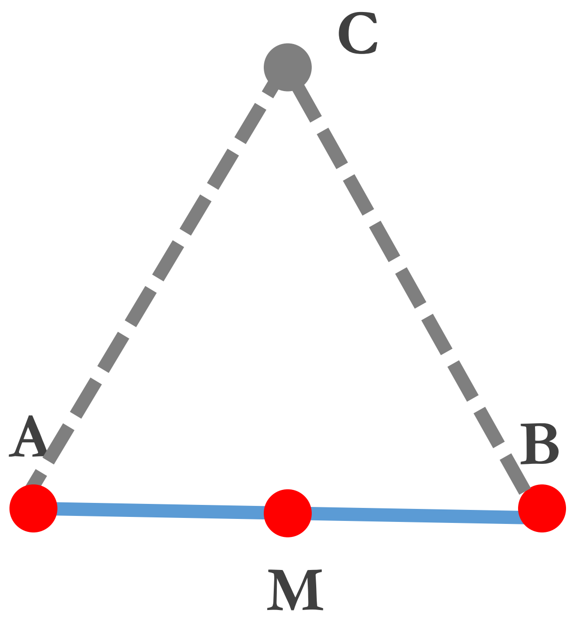


Advancing Front (AF) Technique(s)

Step 3:

Computed distance δ_1 as follows:

$$\delta_1 = \begin{cases} 0{,}55AB & \text{if} \quad \delta_M < 0{,}55AB \\ \delta_M & \text{if} \quad 0{,}55AB < \delta_M < 2AB \\ 2AB & \text{if} \quad 2AB < \delta_M \end{cases}$$



Define C as the point at distance δ_1 from both A and B.

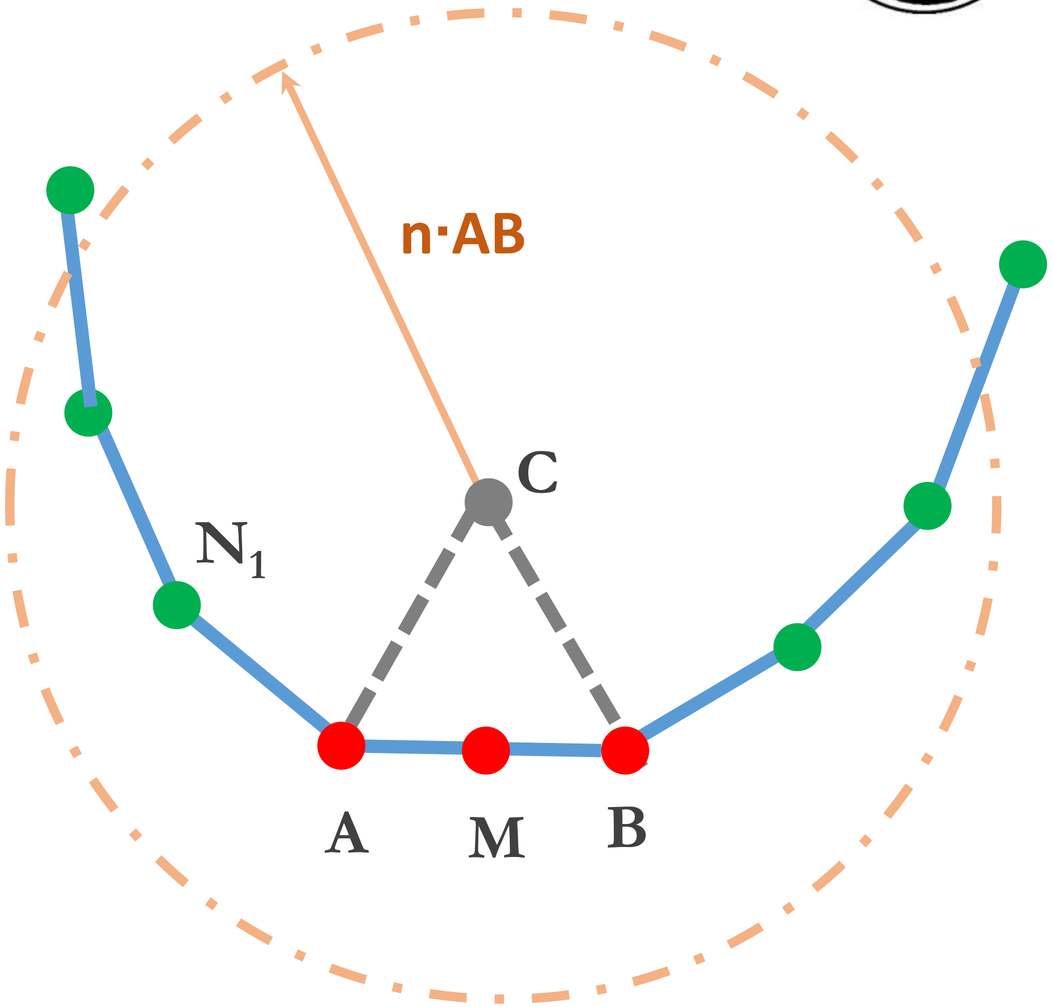
$\boldsymbol{\delta_M \rightarrow \delta_1}$

Advancing Front (AF) Technique(s)

Step 4:

All active nodes of the front which lie in a circle centered at C and its radius equals (n·AB). Recommended values of n between 4 and 5.

The identified nodes are rank sorted according to their distance from C; first entry in this list is the node having min. distance from C. Let us assume that this process identified and recorded p nodes as follows; $N_1, N_2, \ldots, N_p$.



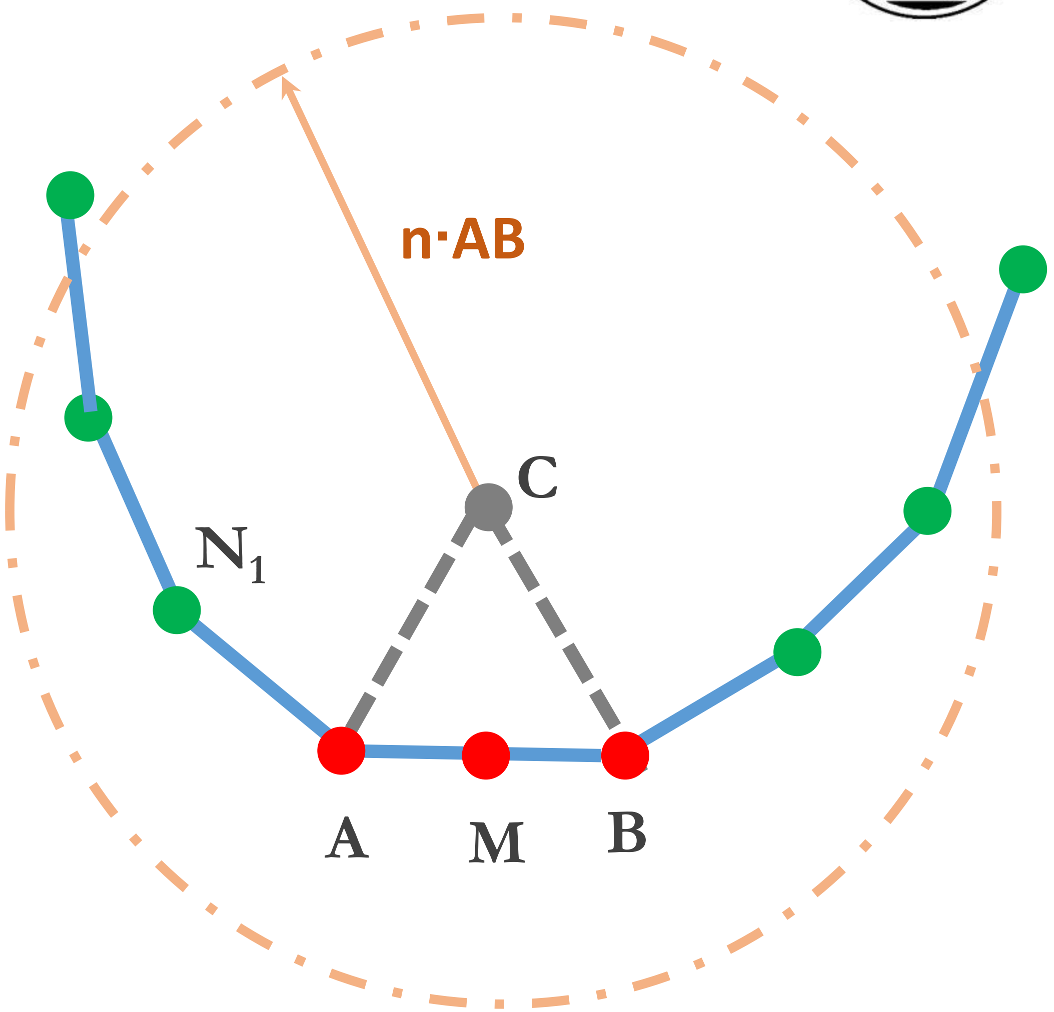
Advancing Front (AF) Technique(s)

Step 5:

Put node C at the first entry of this list (push all nodes down by one position), unless the following two inequalities

$$AN_1<1.5\delta_1 \text{ and } BN_1<1.5\delta_1$$

are both met (SEE NEXT SLIDE). Otherwise, retain the list created in Step 4.



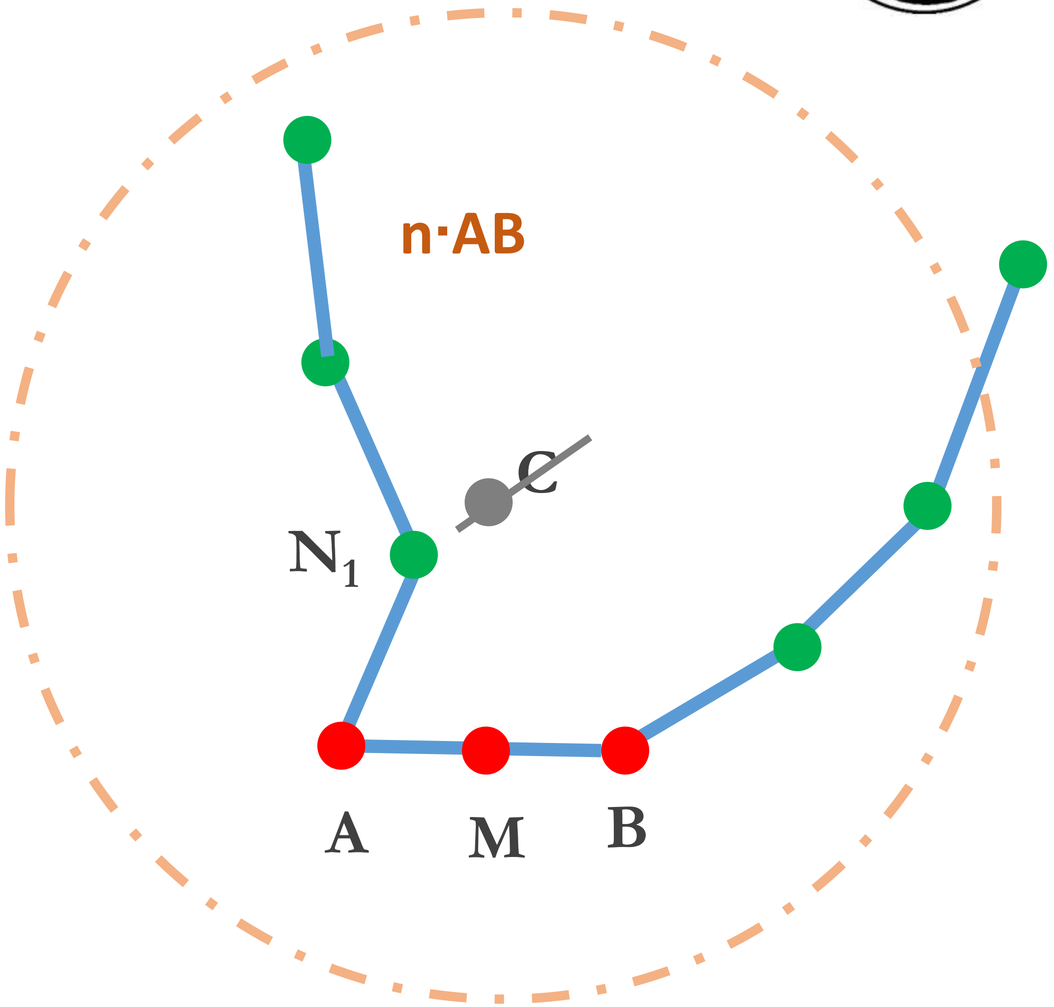
Advancing Front (AF) Technique(s)

Step 5:

Put node C at the first entry of this list (push all nodes down by one position), unless the following two inequalities

$AN_1<1.5\delta_1$ and $BN_1<1.5\delta_1$

are both met. Otherwise, retain the list created in Step 4.

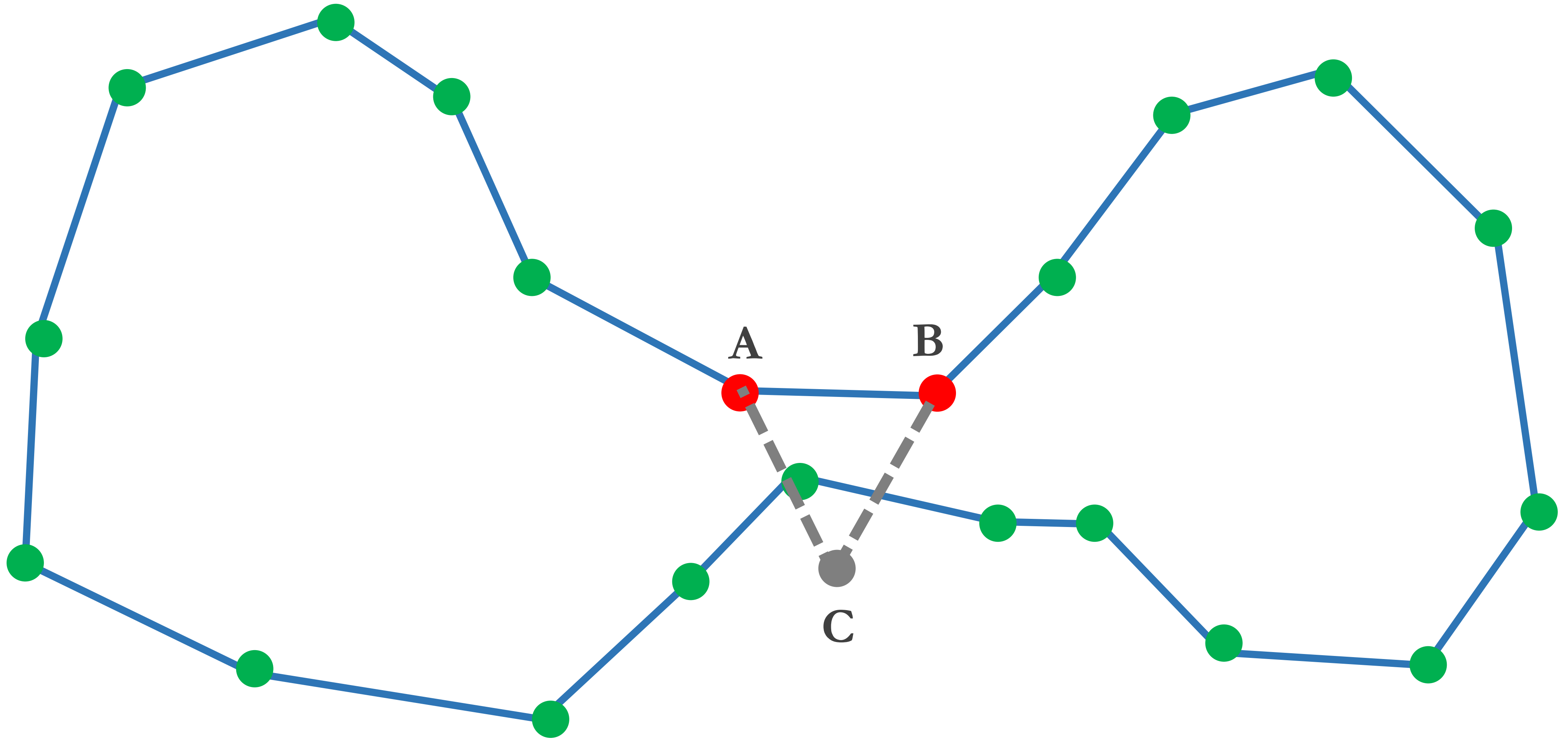


Advancing Front (AF) Technique(s)

Step 6:

Point N_j to be used to form the new triangle ABN_j is the first one in the list for which triangle ABN_j does not included any other node N_i (excluding C, should this be at the top of this list) and, at the same time, line segment MN_j does not intersect with any side of the current front. Having identified node N_j, triangle ABN_j is recorded and the front is updated (node coordinates should, first, be transformed to the physical space. Updating the front means that line segments used to form the new triangle will be eliminated or new triangle edges should be appended to the list. (SEE NEXT SLIDE)

Advancing Front (AF) Technique(s)

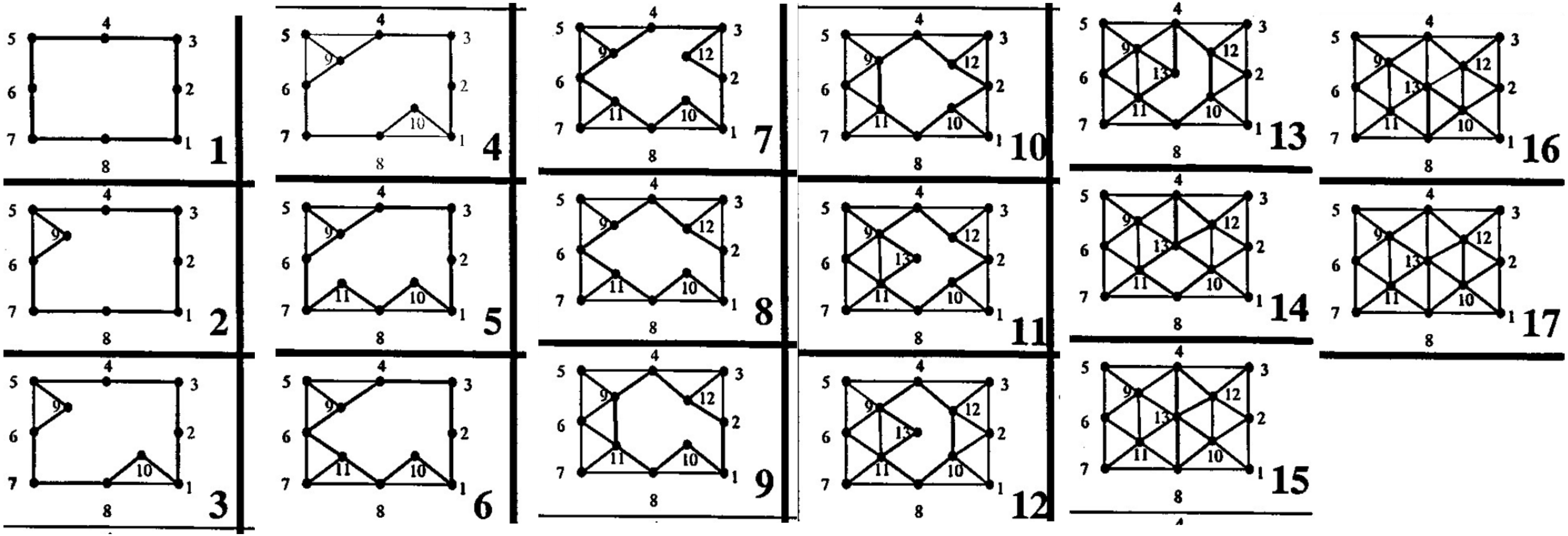


Advancing Front (AF) Technique(s)

Step 7:

Return to Step 1 or terminate if the list of active line segments in the front is empty.

Advancing Front (AF) Technique(s)



Active Edges

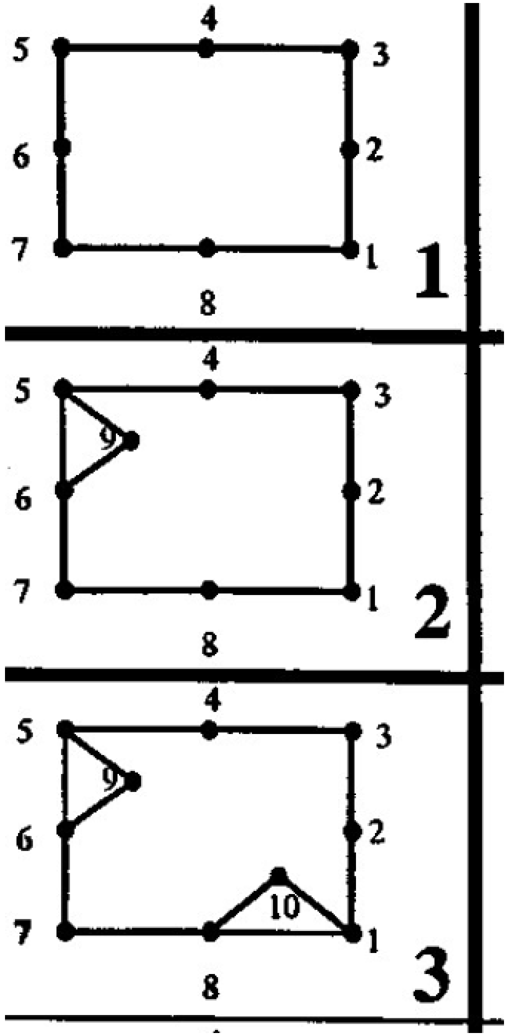
Inactive Edges

Advancing Front (AF) Technique(s)

BH MA										
1	1 2	2 3	3 4	4 5	5 6	6 7	7 8	8 1		
2	1 2	2 3	3 4	4 5	6 7	7 8	8 1	5 9	9 6	
3	1 2	2 3	3 4	4 5	6 7	7 8	5 9	9 6	10 1	8 10
4	1 2	2 3	3 4	6 7	7 8	9 6	10 1	8 10	4 9	
5	1 2	2 3	3 4	6 7	9 6	10 1	8 10	4 9	7 11	11 8
6	1 2	2 3	3 4	9 6	10 1	8 10	4 9	11 8	6 11	
7	1 2	3 4	9 6	10 1	8 10	4 9	11 8	6 11	2 12	12 3
8	1 2	9 6	10 1	8 10	4 9	11 8	6 11	2 12	12 4	
9	1 2	10 1	8 10	4 9	11 8	2 12	12 4	9 11		

10	8 10	4 9	11 8	2 12	12 4	9 11	10 2			
11	8 10	4 9	11 8	2 12	12 4	10 2	9 13	13 11		
12	8 10	4 9	11 8	12 4	9 13	13 11	10 12			
13	8 10	11 8	12 4	13 11	10 12	4 13				
14	8 10	11 8	12 4	13 11	4 13	10 13	13 12			
15	11 8	12 4	13 11	4 13	13 12	8 13				
16	12 4	4 13	13 12							
17	---									

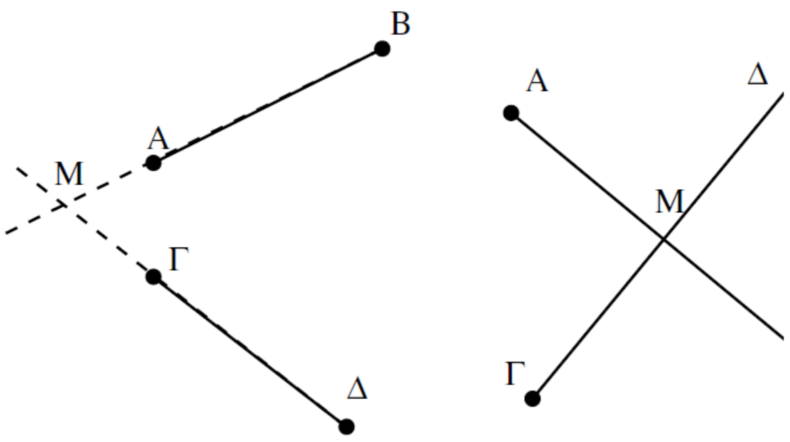
Advancing Front (AF) Technique(s)



BH MA									
1	1	2	3	4	5	6	7	8	
	2	3	4	5	6	7	8	1	
2	1	2	3	4	6	7	8	5	9
	2	3	4	5	7	8	1	9	6
3	1	2	3	4	6	7	5	9	10	8
	2	3	4	5	7	8	9	6	1	10

Advancing Front (AF) Technique(s)

Auxiliary Algorithms



Σχήμα ΠΜ.6: Εύρεση της τομής δύο ευθυγράμμων τμημάτων.

$$\vec{r}_M = \vec{r}_A + \beta_1(\vec{r}_B - \vec{r}_A)$$

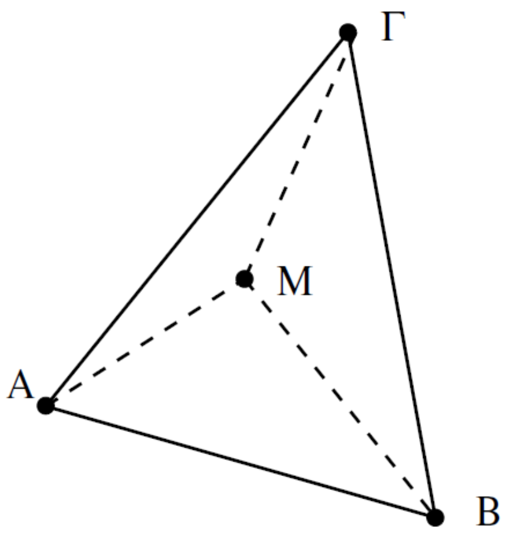
$$\vec{r}_M = \vec{r}_\Gamma + \beta_2(\vec{r}_\Delta - \vec{r}_\Gamma)$$

$$\begin{bmatrix} x_B - x_A & x_\Gamma - x_\Delta \\ y_B - y_A & y_\Gamma - y_\Delta \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} x_\Gamma - x_A \\ y_\Gamma - y_A \end{bmatrix}$$

$0 \leq \beta_1 \leq 1$ και $0 \leq \beta_2 \leq 1$

Advancing Front (AF) Technique(s)

Auxiliary Algorithms



Σχήμα ΠΜ.7: Τοποθέτηση σημείου Μ ως προς τρίγωνο (εσωτερικό ή εξωτερικό)

Natural Coordinates:

$$L_1 = \frac{(\mathrm{M B \Gamma})}{(\mathrm{A B \Gamma})} \qquad L_1 = \frac{(\mathrm{M A \Gamma})}{(\mathrm{A B \Gamma})} \qquad L_3 = \frac{(\mathrm{M A B})}{(\mathrm{A B \Gamma})}$$

$$L_1 + L_2 + L_3 = 1$$

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \frac{1}{2E} \begin{bmatrix} (x_{\mathrm{B}} y_{\Gamma} - x_{\Gamma} y_{\mathrm{B}}) & (y_{\mathrm{B}} - y_{\Gamma}) & (x_{\Gamma} - x_{\mathrm{B}}) \\ (x_{\Gamma} y_{\mathrm{A}} - x_{\mathrm{A}} y_{\Gamma}) & (y_{\Gamma} - y_{\mathrm{A}}) & (x_{\mathrm{A}} - x_{\Gamma}) \\ (x_{\mathrm{A}} y_{\mathrm{B}} - x_{\mathrm{B}} y_{\mathrm{A}}) & (y_{\mathrm{A}} - y_{\mathrm{B}}) & (x_{\mathrm{B}} - x_{\Gamma}) \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

$$0 \le L_1 \le 1 \qquad 0 \le L_2 \le 1, \qquad 0 \le L_3 \le 1$$