Scientific Visualization

Hai Lin

12 Sept 2024

Lecture 2
Data type and data operation

Data type and data operation

- Data presentation and data acquisition technique
  - Regular and irregular data, CT,MRI etc.
- Data operation
  - Sampling , interpolating , filtering

1

Data type

- Data source
- Sources of error
- Data representation
- Domain
- Data structures

- Data source
- Sources of error
- Data representation
- Domain
- Data structures

Data source

- Real-world measurements

• Medical Imaging (MRI, CT, PET) • Geographical information systems (GIS) • Electron microscopy	*MB*
• Meteorology and environmental sciences (satellites) • Seismic data • Crystallography	*GB*
• High energy physics • Astronomy (e.g. Hubble Space Telescope 100MB/day) • Defense	*TB*

Data source

- Theoretical world
- Computer simulations
  - Sciences
    - Molecular dynamics
    - Quantum chemistry
    - Mathematics

    MB

    - Molecular modeling
    - Computational physics
    - Meteorology
    - Computational fluid mechanics (CFD)

    GB

  - Engineering
    - Architectural walk-throughs

    MB

    - Structural mechanics
    - Car body design

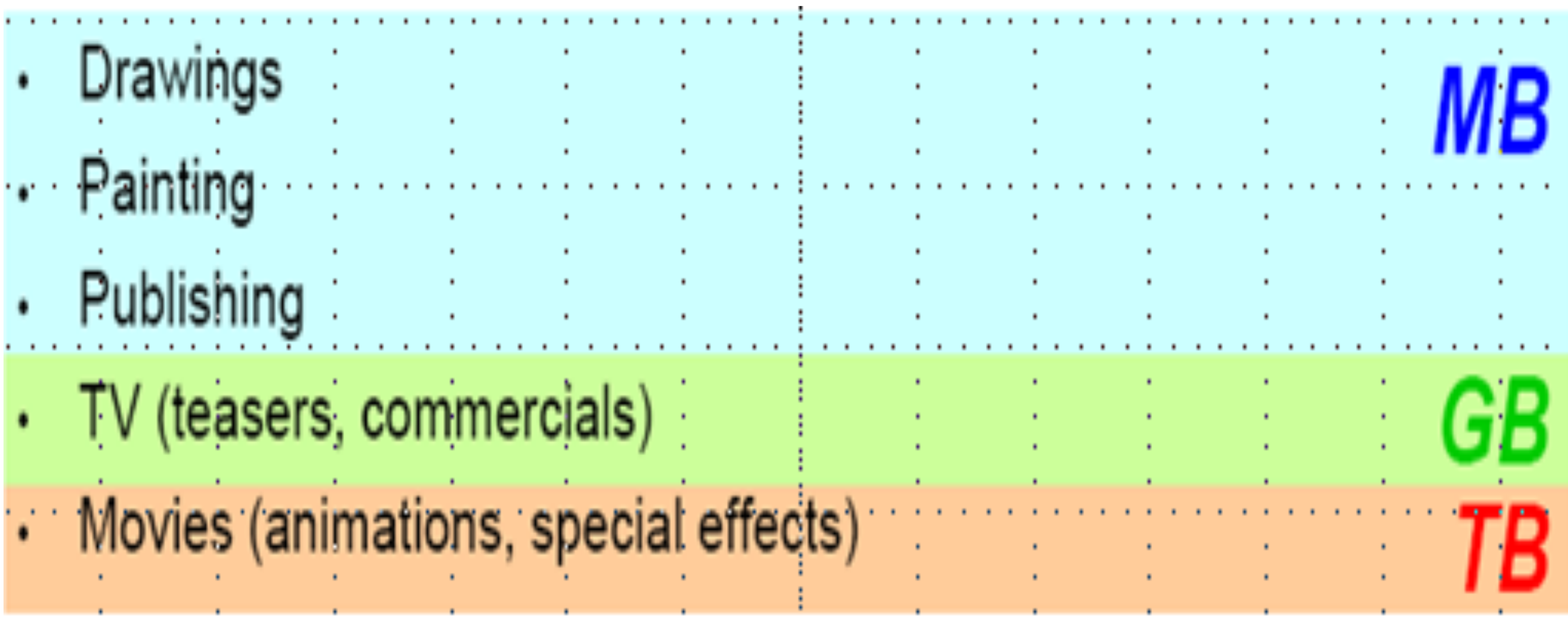
    GB

Data source

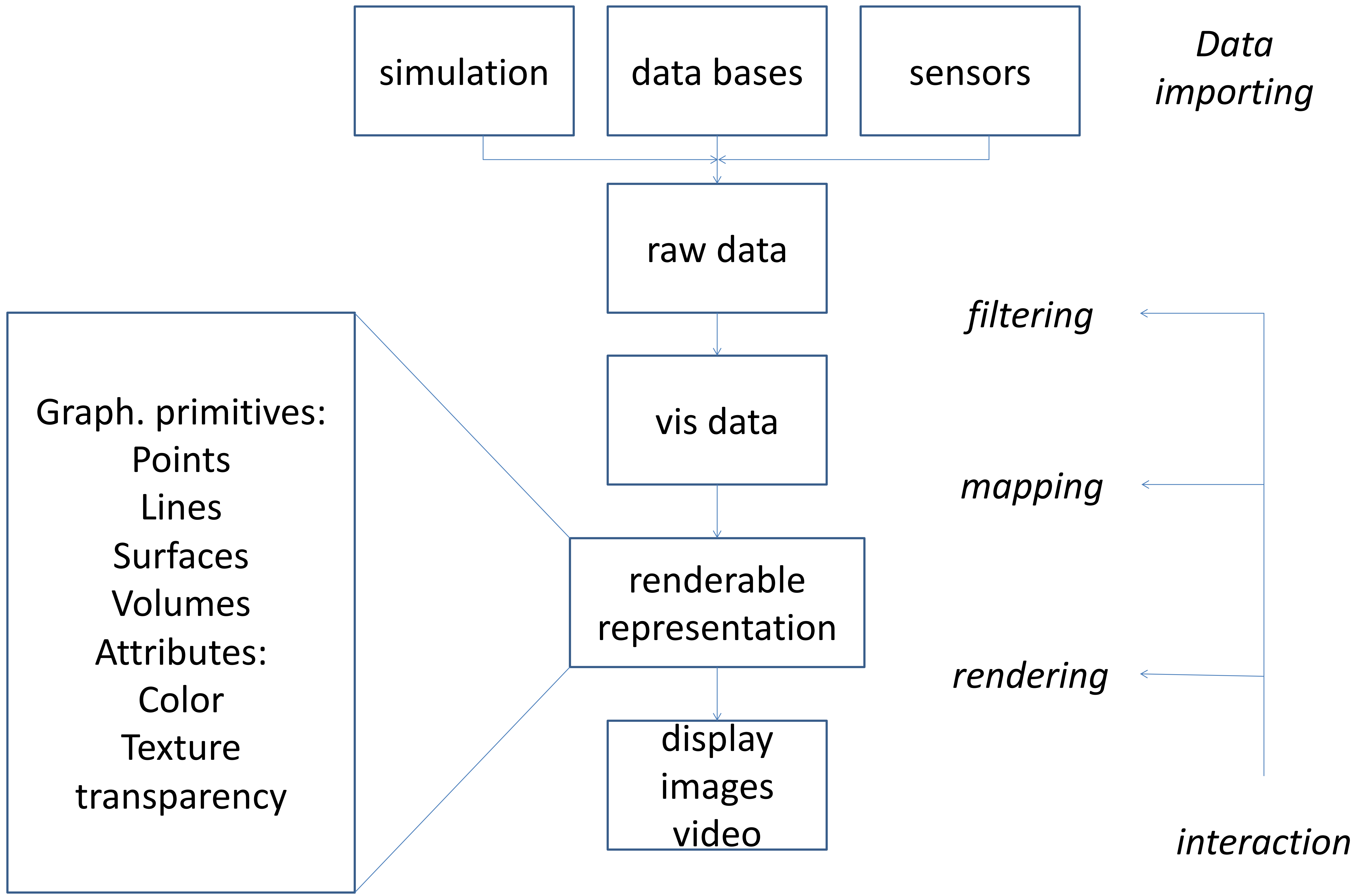
- Theoretical world
- Computer simulations
  - Commercial
    - Business graphics — MB
    - Economic models
    - Financial modeling — GB
- Information systems
  - Stock market (300 Mio. transactions per day in NY)
  - Market and sales analysis
  - World Wide Web !!! — TB

Data source

- Artificial world

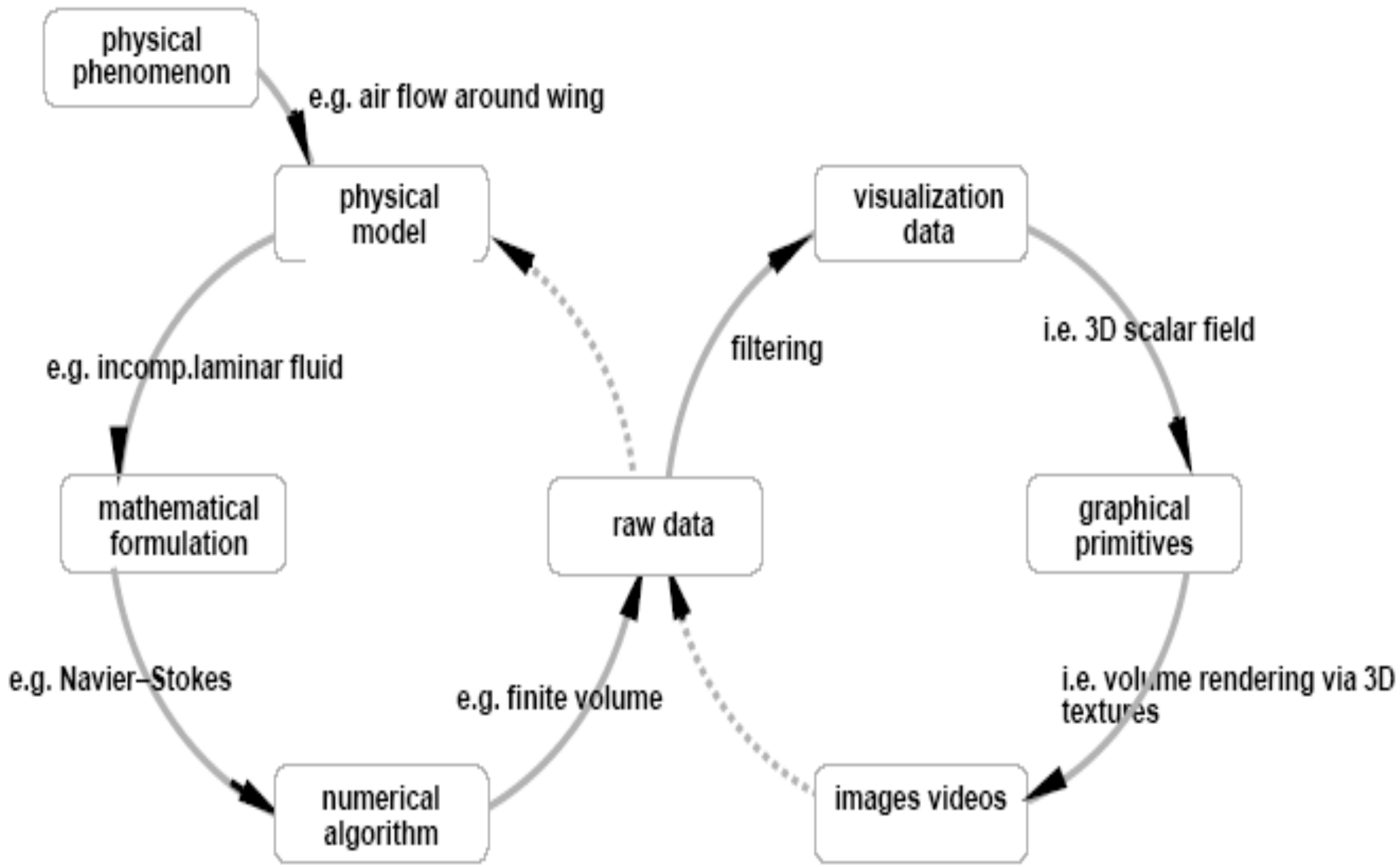


Visualization pipeline



Visualization pipeline

- Example: simulation of the flow within a fluid around a wing



- Data source
- Sources of error
- Data representation
- Domain
- Data structures

Sources of error

- Data acquisition
  - Sampling density
  - Sampling quantization
- filtering
  - Whether features of interest are preserved
- Selecting the “right” variable

Sources of error

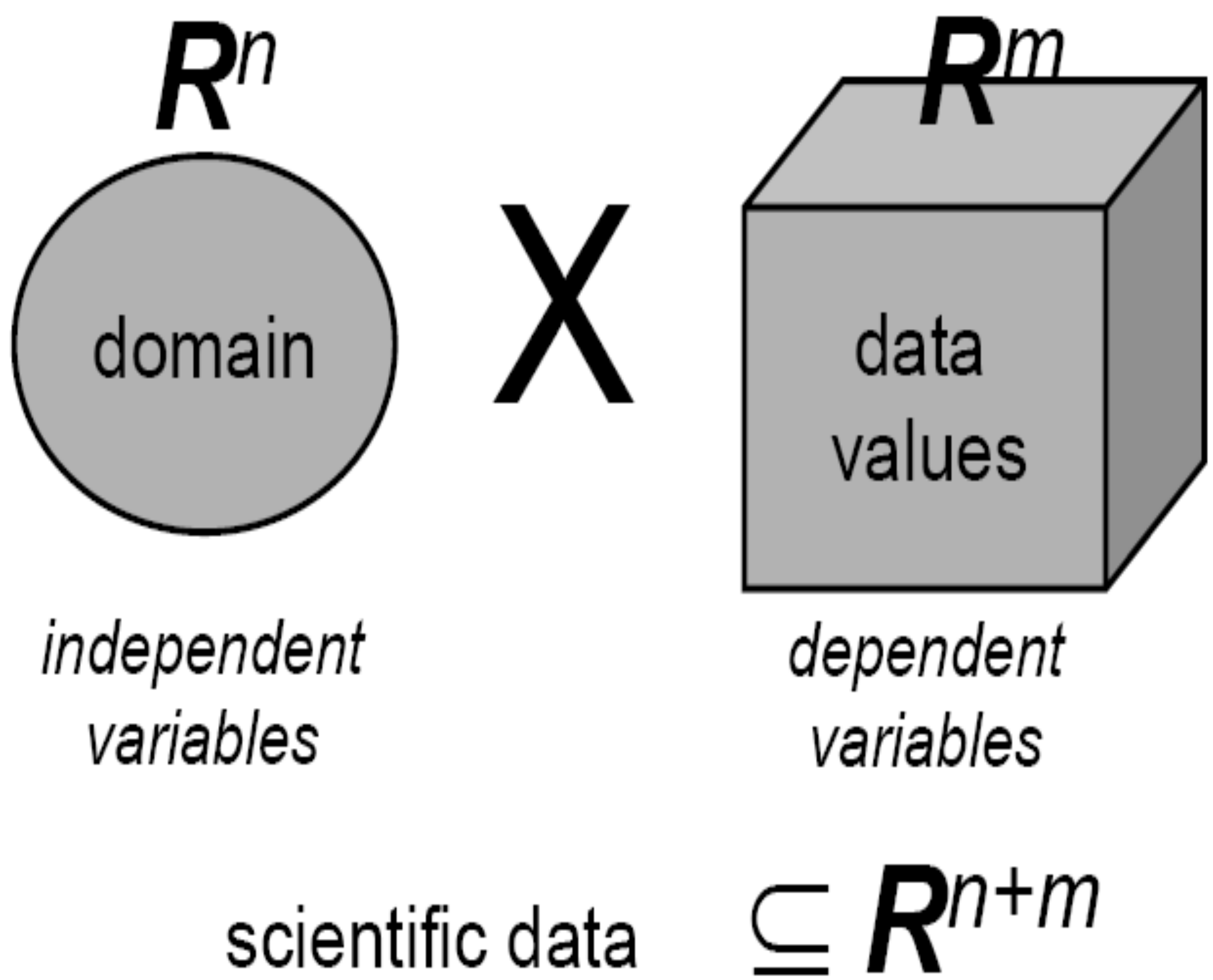
- function model for resampling
- mapping
  - Are we choosing the graphical primitives appropriately in order to depict the kind of information we want to get out of the data?
- rendering
  - Need for interactive rendering often determines the chosen abstraction level
  - Consider limitations of the underlying display technology
  - Carefully add “realism”

- Data source
- Sources of error
- Data representation
- Domain
- Data structures

Data Representation

- Dimension of data:0,1,2,3
- Data types
  - Divergence、vector、tensor、multi-variable
- variable range
- Structure of the data

Data Representation



Data Representation

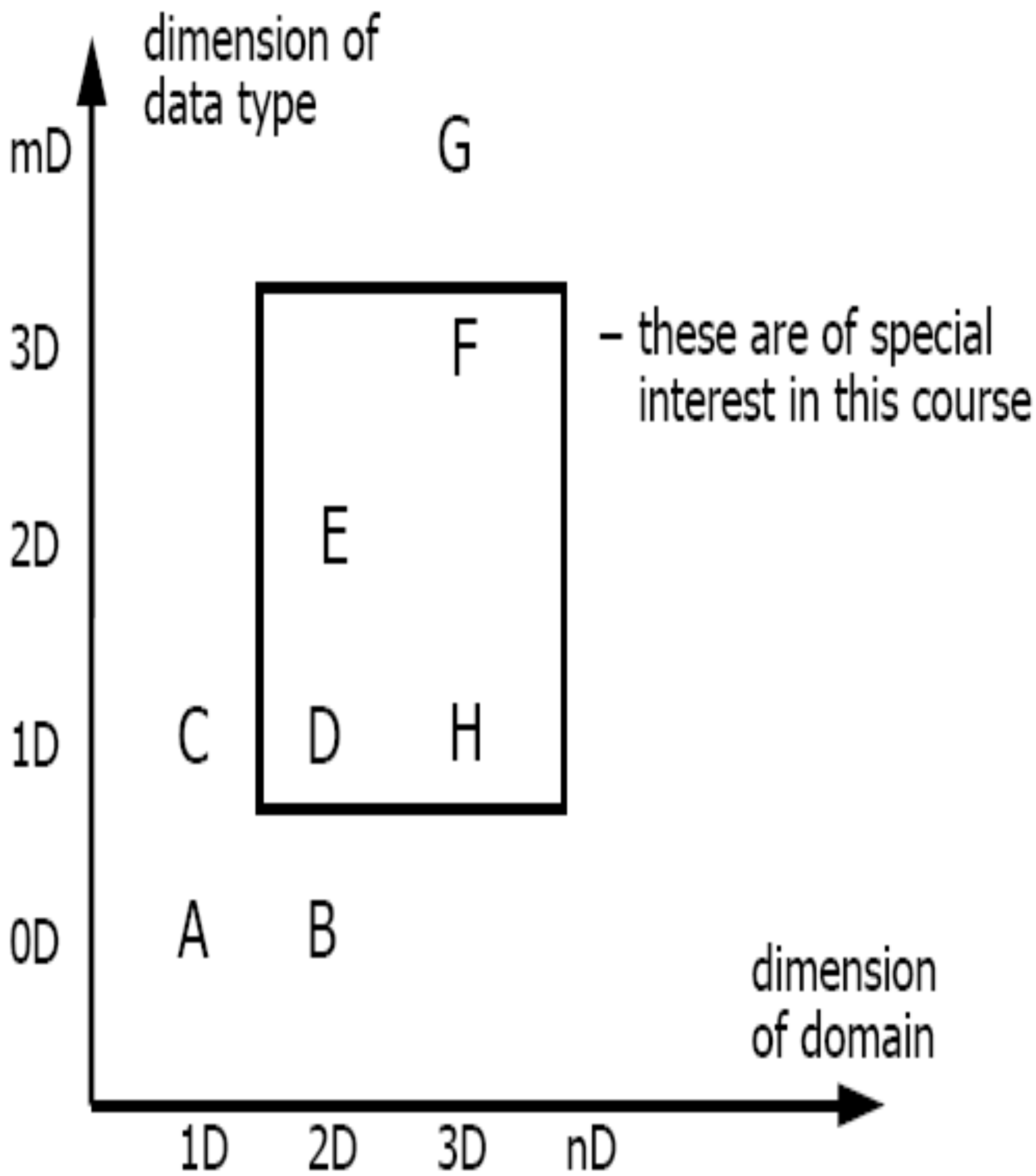
Discrete representations

- The objects we want to visualize are often ‘continuous’
- the visualization data is given only at discrete locations in space and/or time
- Discrete structures consist of samples, from which grids/meshes consisting of cells are generated

- Primitives in multi dimensions

dimension	cell	mesh
0D	points	
1D	lines (edges)	polyline(–gon)
2D	triangles, quadrilaterals (rectangles)	2D mesh
3D	tetrahedra, prisms, hexahedra	3D mesh

Data Representation



Examples:

A: gas station along a road
B: map of cholera in London
C: temperature along a rod
D: height field of a continent
E: 2D air flow
F: 3D air flow in the atmosphere
G: stress tensor in a mechanical part
H: ozon concentration in the atmosphere

Data Representation

- The (geometric) shape of the domain is determined by the positions of sample points

- Domain is characterized by
  - dimension
  - influence
  - structure

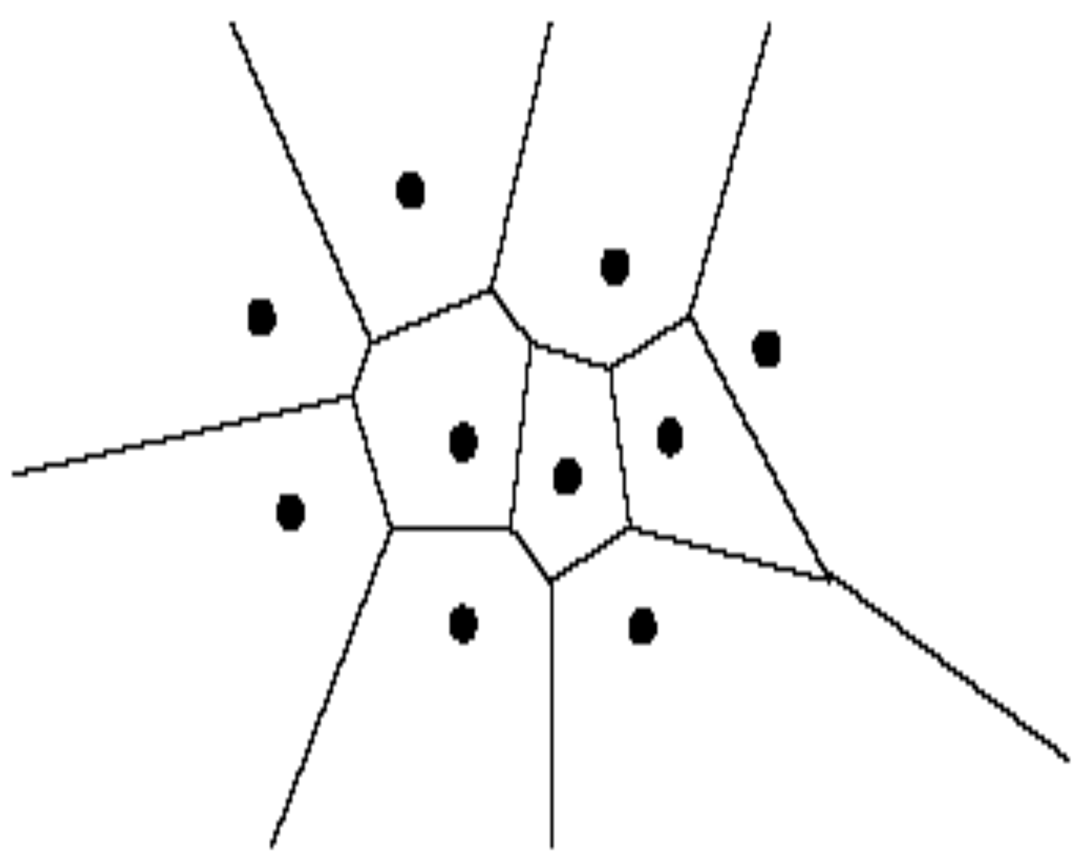
- Data source
- Sources of error
- Data representation
- Domain
- Data structures

Domain

- Influence of data points
  - Values at sample points influence the data distribution in a certain region around these samples
  - To reconstruct the data at arbitrary points within the domain, the distribution of all samples has to be calculated
- Point influence
  - Only influence on point itself
- Local influence
  - Only within a certain region
    - Voronoi-diagram
    - Cell-wise interpolation(see later in course)
- Global influence
  - Each sample might influence any other point within the domain
    - Material properties for whole object
    - Scattered data interpolation

domain

- Voronoi-diagram
  - Construct a region around each sample point that are closer to that sample than to every other sample
  - Each point within a certain region gets assigned the value of the sample point



- Data source
- Sources of error
- Data representation
- Domain
- Data structures

Data structure

Requirements

- Efficiency of accessing data
- Space efficiency
- Lossless vs. lossy
- Binary/Text

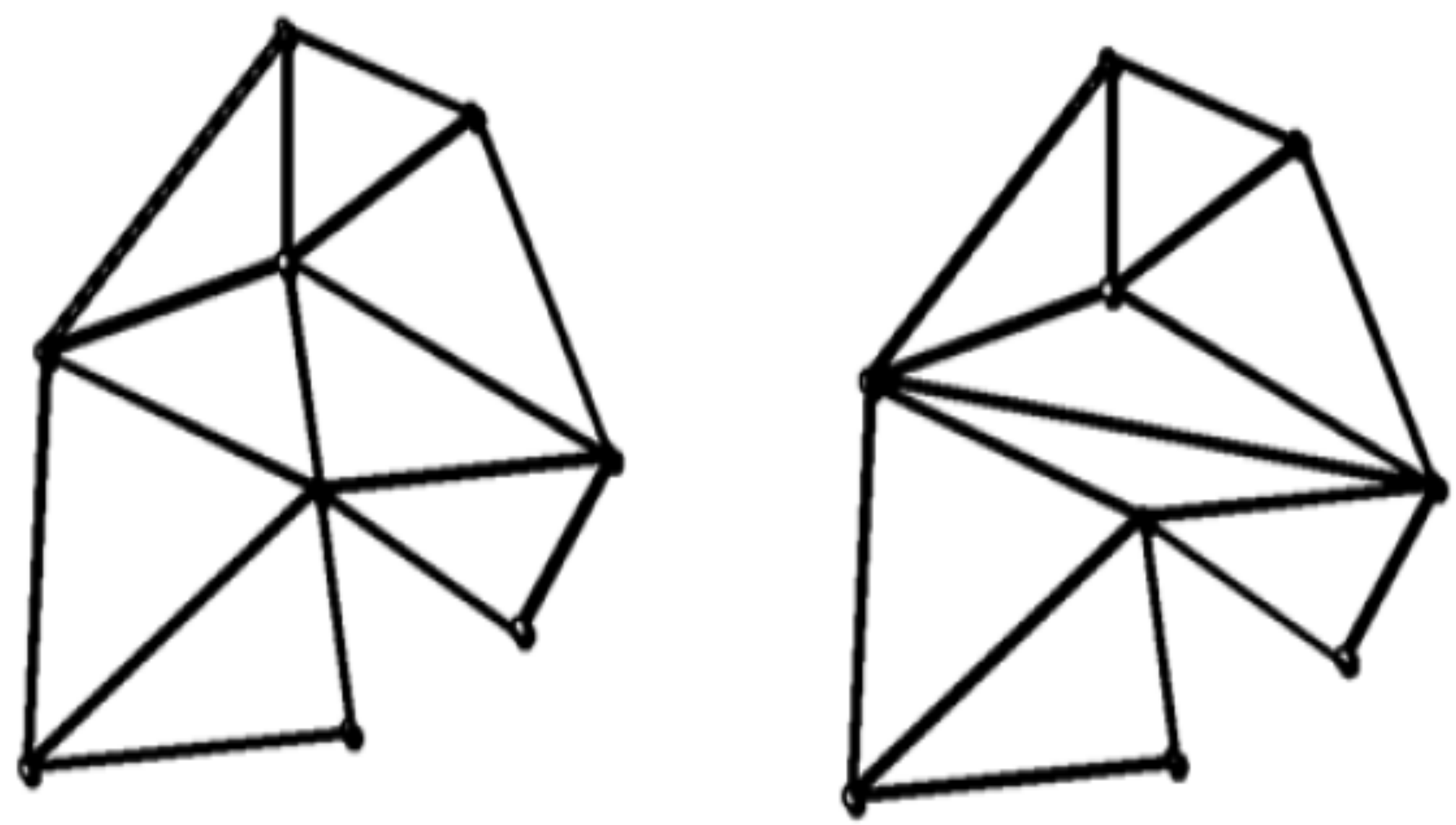
Definition

- Scattered: points are arbitrarily distributed and no connectivity exists between them
- cells
- Topology :the structure (connectivity) of the data
- Geometry： the position of the data

Data structure

■ Example

- ➢ Radial basis functions with increasing support

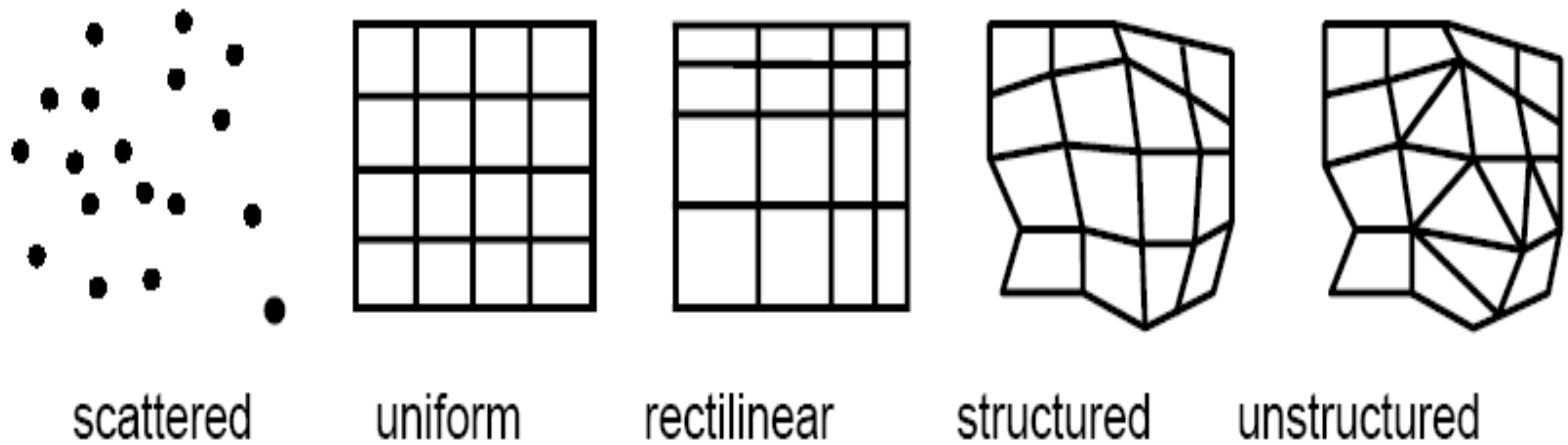


Same geometry (vertex positions), different topology (connectivity)

Data structure

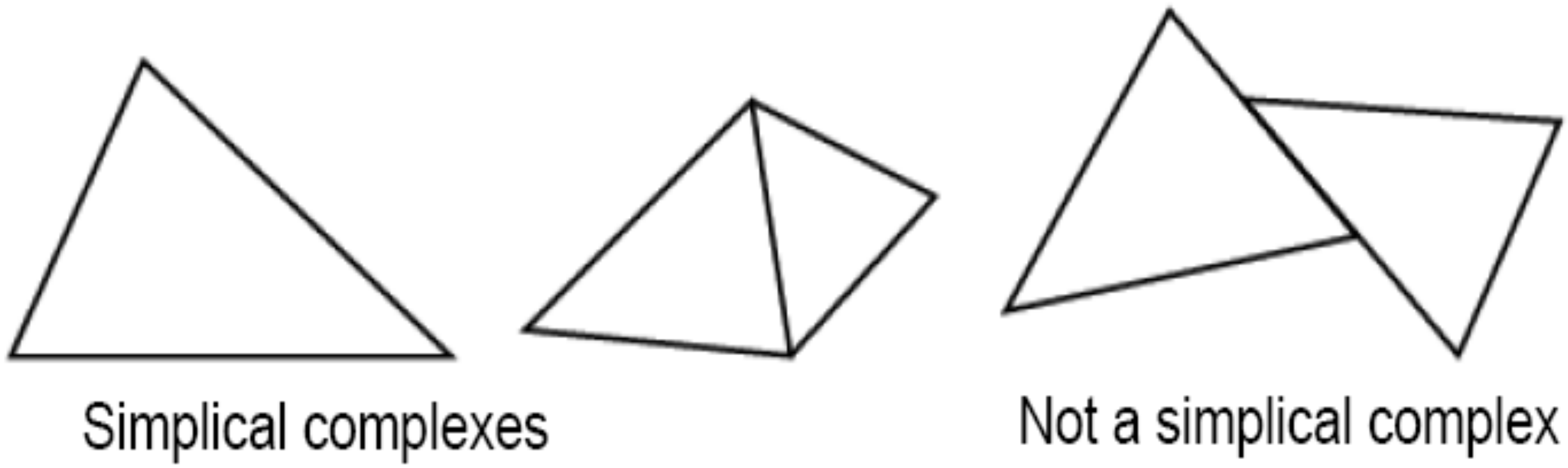
■ Grid types

- Grids differ substantially in the simplical elements (or cells)they are constructed from and in the way the inherent topological information is given



Data structure

- A simplex in R^n
  - The convex hull of n+1 affinely independent points
  - 0:points,1:lines,2:triangles,3:tetrahedra
- Partitions via simplices are called triangulations
- Simplical complex is a collection of simplices with:
  - Every face of an element of C is also in C
  - The intersection of two elements of C is empty or it is a face of both elements
- Simplical complex is a space with a triangulation



Data structure

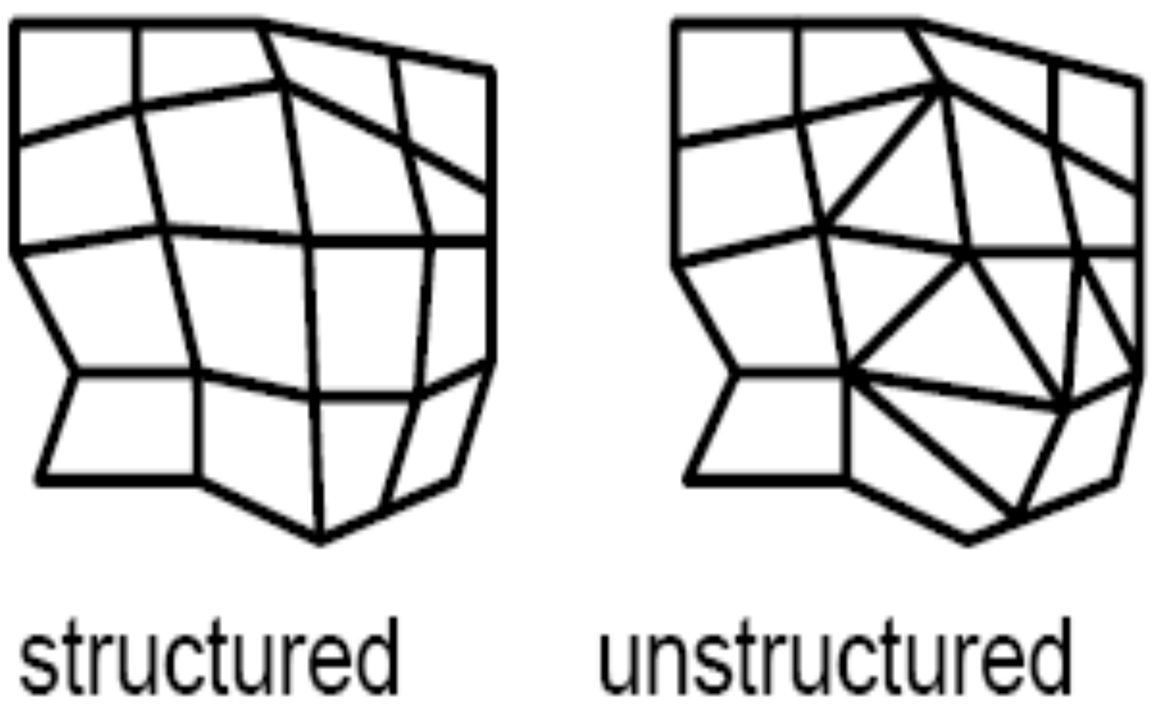
- Structured and unstructured grids can be distinguished by the way the elements or cells meet

- Structured grids

   - Have a regular topology and regular / irregular geometry

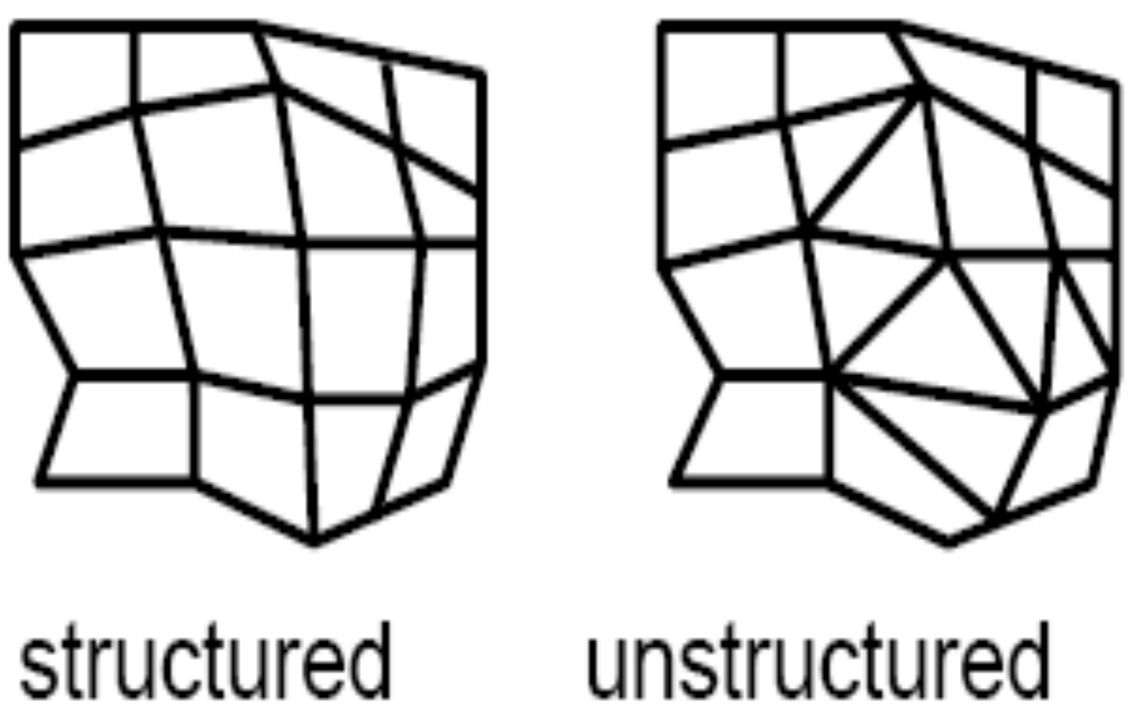
- Unstructured grids

   - Have irregular topology and geometry



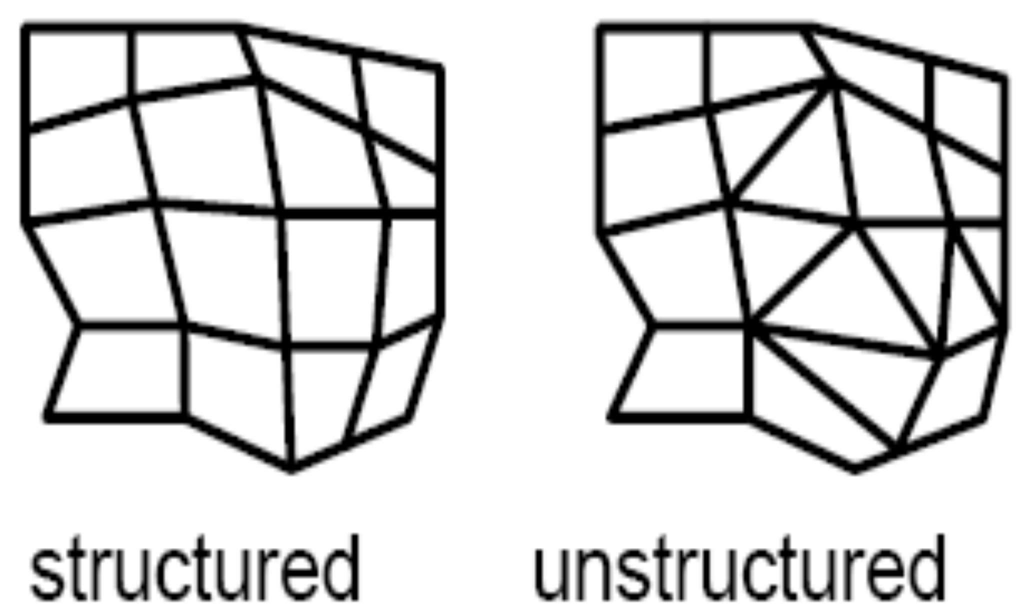
Data structure

- Characteristics of structured grids
  - Easier to compute with
  - Often composed of sets of connected parallelograms(hexahedra),with cells being equal or distorted with respect to (non-linear) transformations
  - May require more elements or badly shaped elements in order to precisely cover the underlying domain
  - Topology is represented implicitly by n-vector of dimensions
  - Geometry is represented explicitly by an array of points
  - Every interior point has the same number of neighbors



Data structure

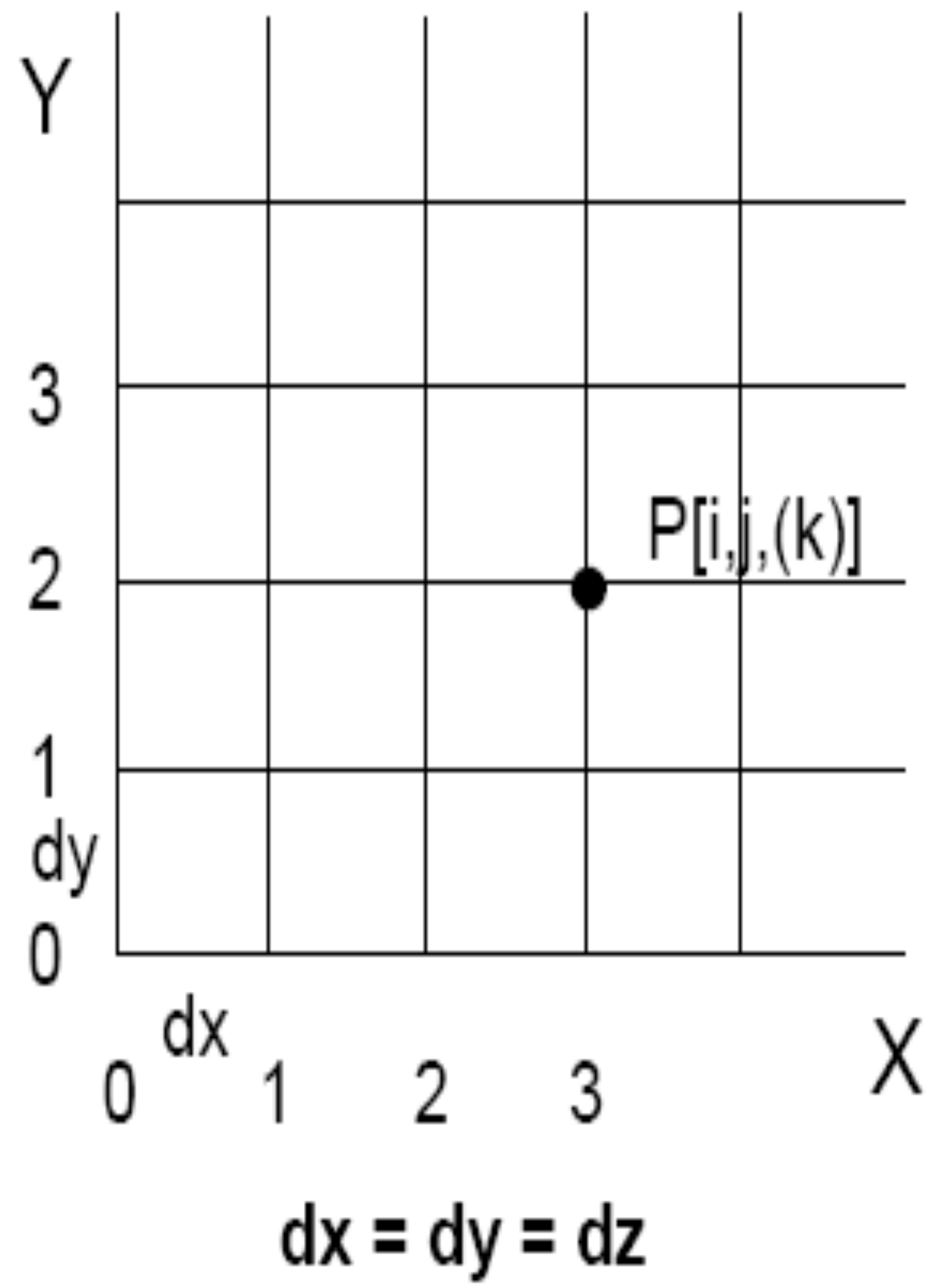
- If no implicit topological (connectivity)information is given the grids are termed unstructured grids
  - Unstructured grids are often computed using quadtrees(recursive domain partitioning for data clustering),or by triangulation of points sets
  - The task is often to create a grid from scattered points
- Characteristics of unstructured grids
  - Grid point geometry and connectivity must be stored
  - Dedicated data structures needed to allow for efficient traversal and thus data retrieval
  - Often composed of triangles or tetrahedra
  - Less elements are needed to cover the domain



Data structure

- Cartesian or equidistant grids

  - Structured grid
  - Cells and points are numbered sequentially with respect to increasing X,then Y,then Z ,or vice versa
  - Number of points=Nx·Ny·Nz
  - Number of cells =(Nx-1)·(Ny-1)·(Nz-1)



Data structure

- Cartesian grids

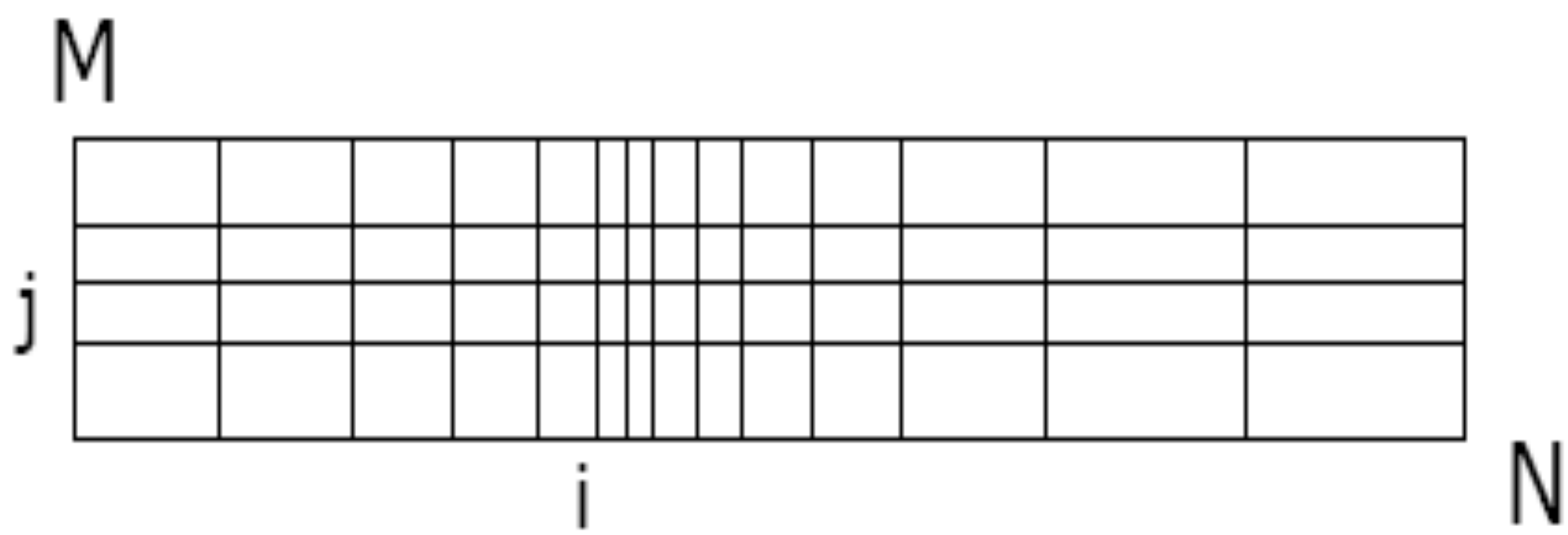
  - Vertex positions are given implicitly from [i,j,k]:
    - $P[i,j,k].x = origin + i \cdot dx$
    - $P[i,j,k].y = origin + j \cdot dy$
    - $P[i,j,k].z = origin + k \cdot dz$
  - Global vertex index $l[i,j,k] = k \cdot Ny \cdot Nx + j \cdot Nx + i$
    - $K = l/(Ny \cdot Nx)$
    - $j = (l\%(Ny \cdot Nx))/Nx$
    - $i = (l\%(Ny \cdot Nx))\%Nx$
  - Global index allows for linear storage scheme
    - Wrong access pattern might destroy cache coherence

Uniform grids

- Similar to Cartesian grids
- Consist of equal cells but with different resolution in at least one dimension($dx \neq dy(\neq dz)$)
- Spacing between grid points is constant in each dimension → same indexing scheme as for Cartesian grids
- Typical example : medical volume data consisting of slice images
- Most likely to occur in applications where the data is generated by a 3D imaging device providing different sampling rates in each dimension
  - Slice images with square pixels($dx=dy$)
  - Larger slice distance ($dz>dx=dy$)

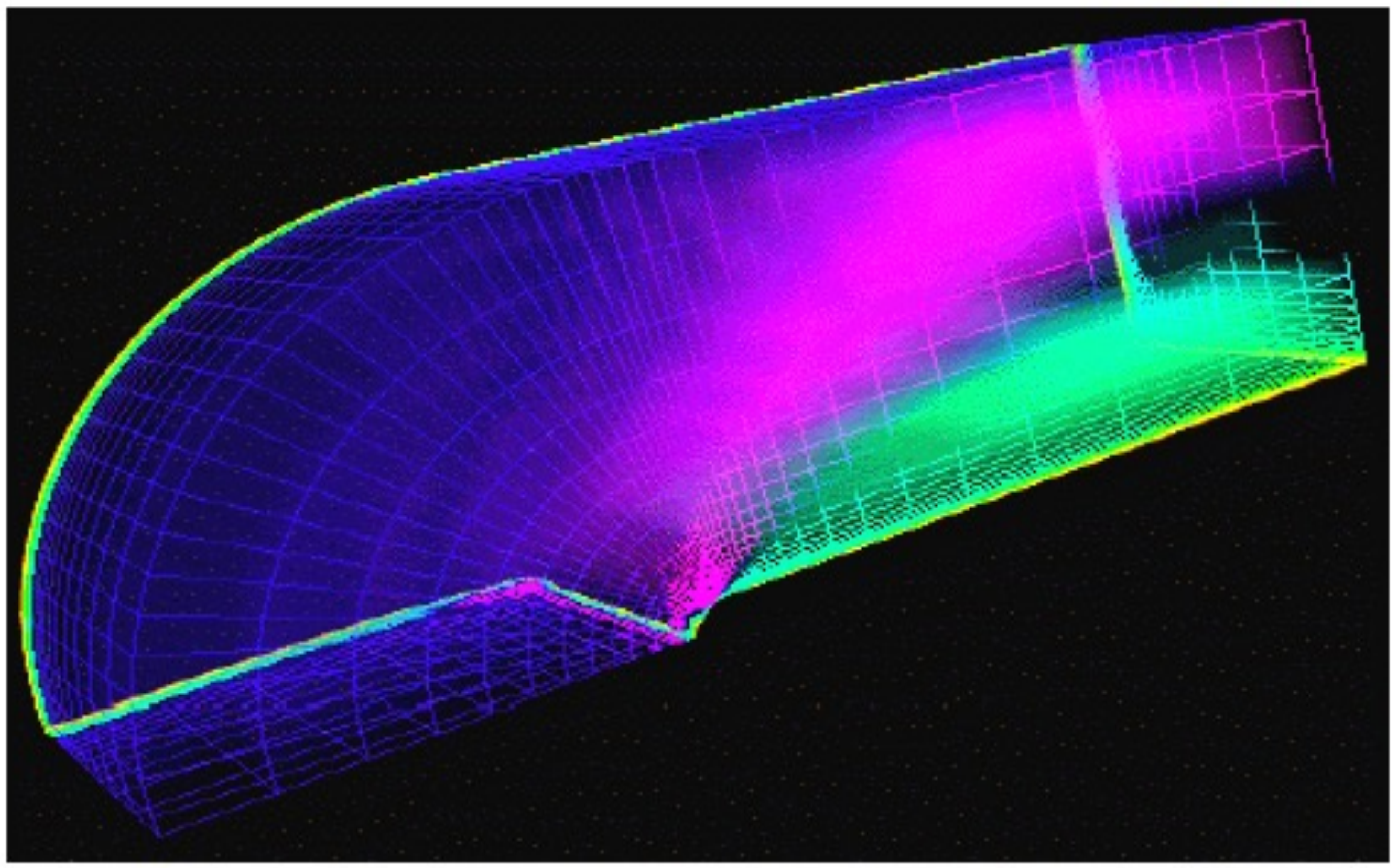
Rectilinear grids

- Topology is still regular but irregular spacing between grid points
    - Non-linear scaling of positions along either axis
    - Spacing, x_coorrd[L],y_coord[M],z_coord[N],must be stored explicitly
- Topology is still implicit



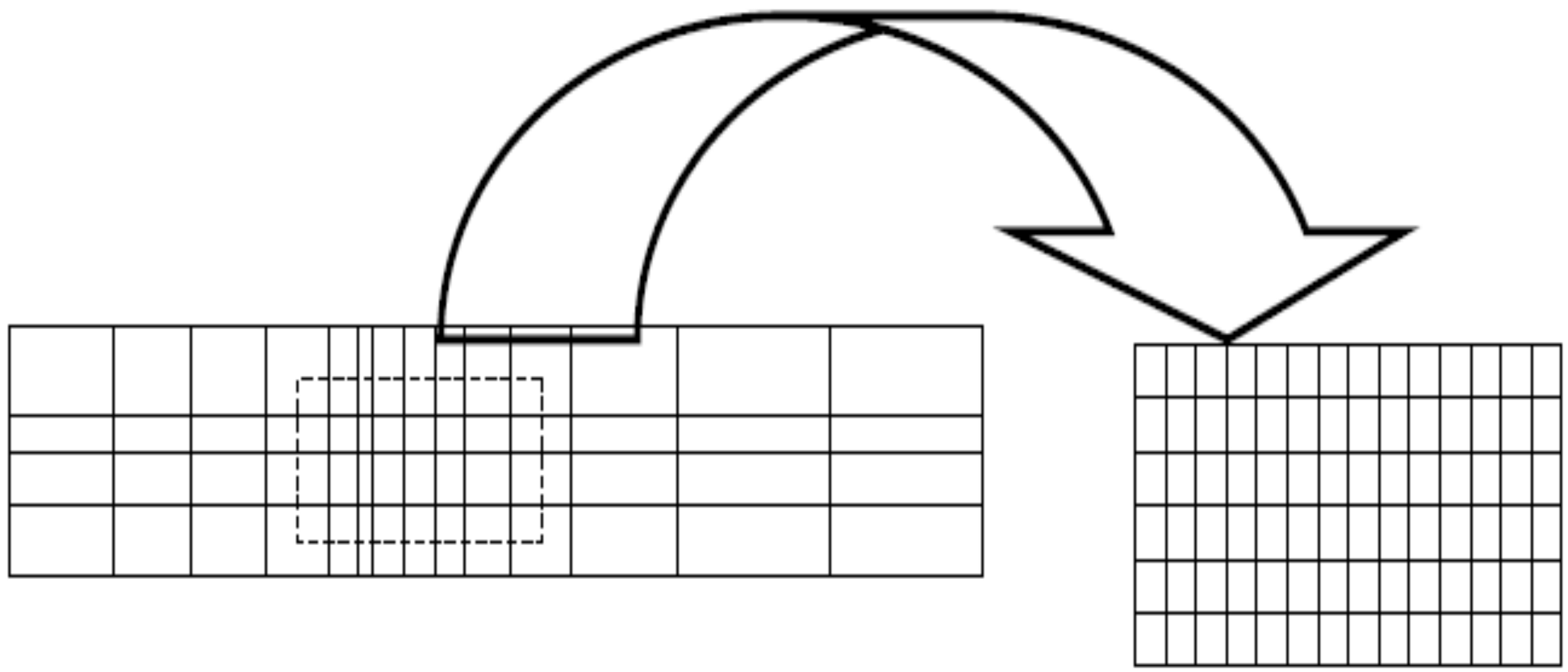
■ Curvilinear grids

- Topology is still regular but irregular spacing between grid points
    - Positions are non-linearly transformed
- Topology is still implicit , but vertex positions are explicitly stored
    - x_coord[L,M,N]
    - y_coord[L,M,N]
    - z_coord[L,M,N]
- Geometric structure might result in concave grids



Multigrids

- Focus in arbitrary areas to avoid wasted detail
- “blow up” regions of interest , i.e. finer grid
- Difficulties in the boundary region(i.e. interpolation)



Characteristics of structured grids

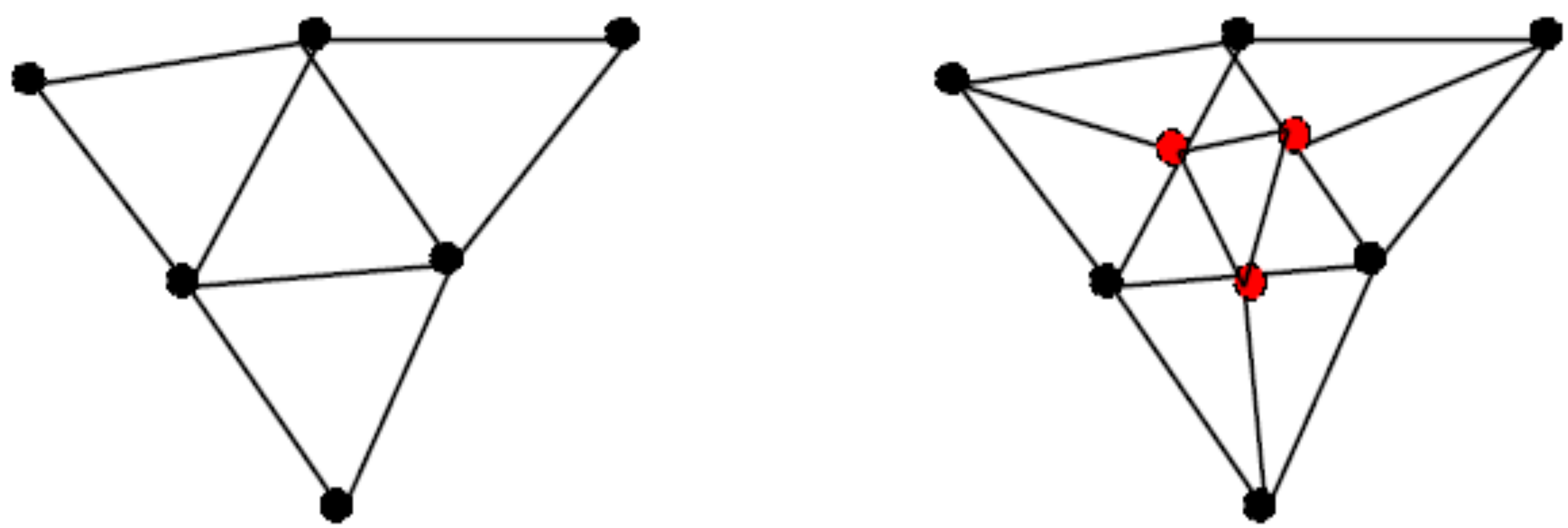
- Structured grids can be stored in a 2D/3D array
- Arbitrary samples can be directly accessed by indexing a particular entry in the array
- Topological information is implicitly coded
    - Direct access to adjacent elements at random
- Cartesian , uniform , and rectilinear grids are necessarily convex
- Their rigid layout prohibits the geometric structure to adapt to local features
- Curvilinear grids reveal a much more flexible alternative to model arbitrarily shaped objects
- However , this flexibility in the design of the geometric shape makes the sorting of grid elements a more complex procedure

- Typical implementation of structured grids

  - $DataType\ *data = new\ DataType[Nx \cdot \mathrm{Ny} \cdot \mathrm{Nz}];$
    val = data[i·(Ny·Nz) + j·Nz + k];

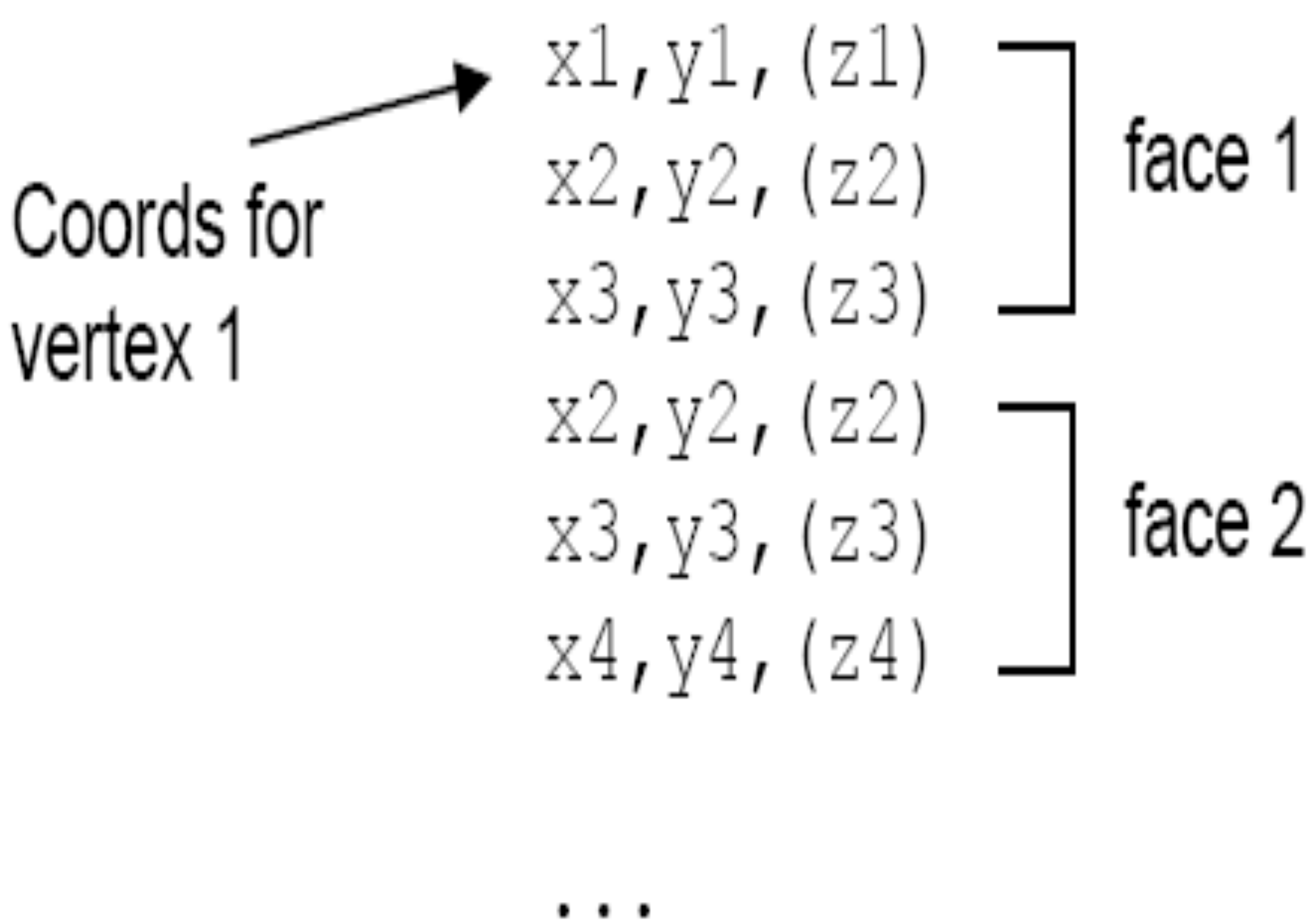
Unstructured grids

- Composed of arbitrarily positioned and connected elements
- Can be composed of one unique element type or they can be hybrid (tetras , hexas , prisms)
- Triangle meshes in 2D and tetrahedral grids in 3D are most common
- Can adapt to local features (small vs. large cells)
- Can be refined adaptively
- Simple linear interpolation in simplices



■ Typical implementations of unstructured grids

- Direct form



2D

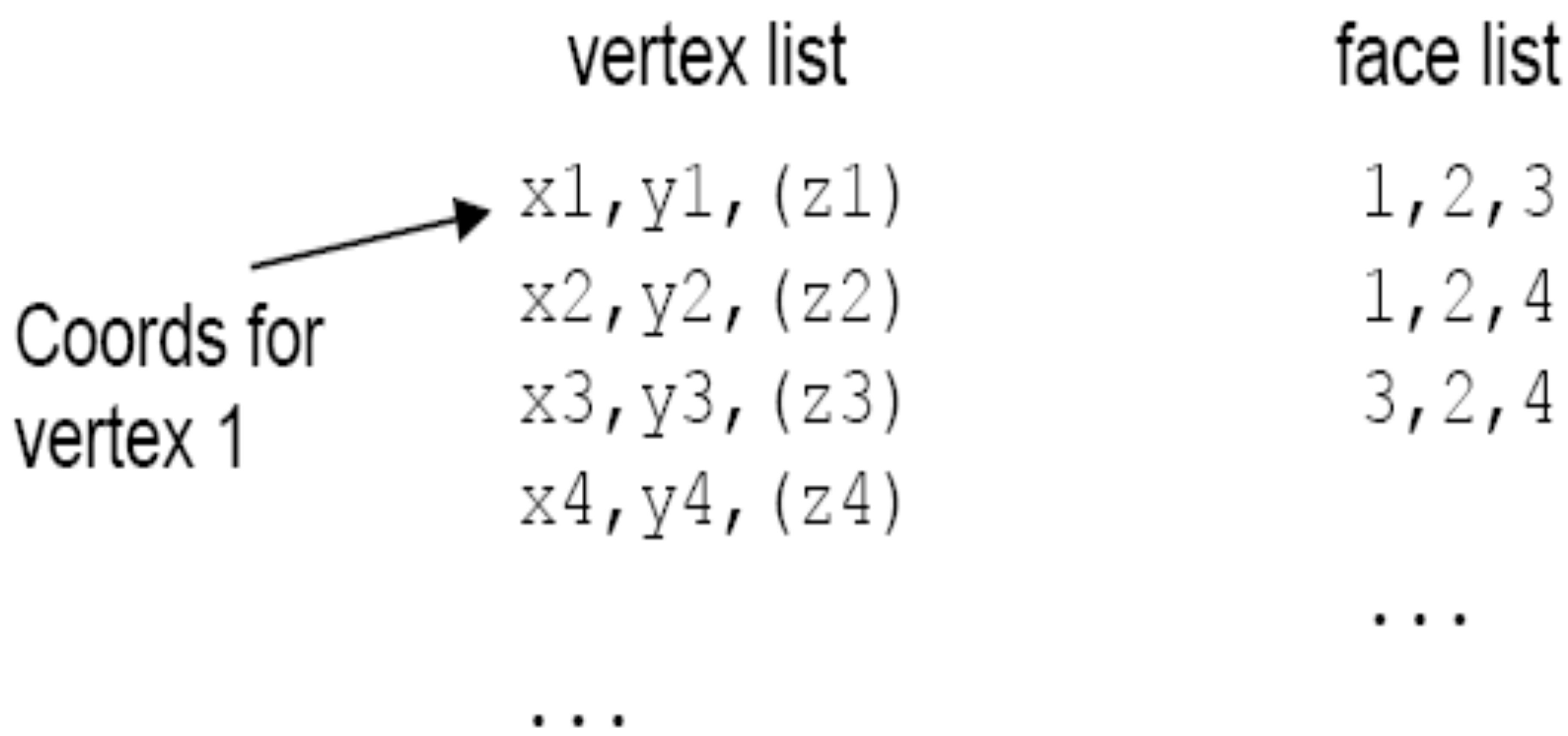
```
struct face
  float verts[3][2]
  DataType val;
```

3D

```
struct face
  float verts[3][3]
  DataType val;
```

- Additionally store the data values
- Problem : storage space , redundancy

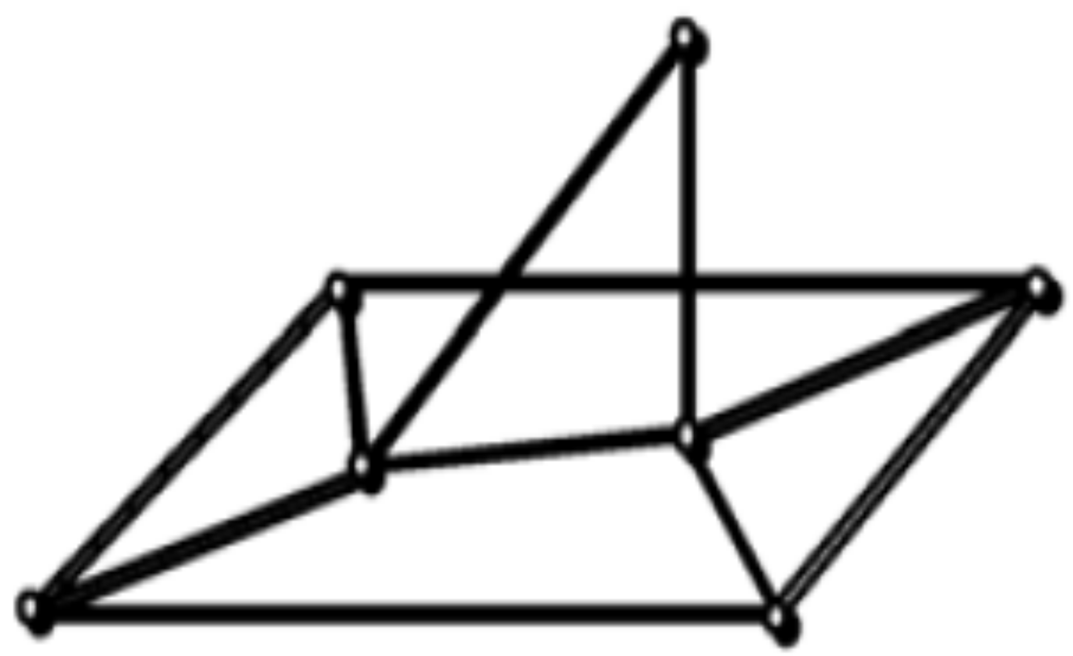
- Typical implementations of unstructured grids
  - Indirect form



  - Indexed face set
  - More efficient than direct approach in terms of memory requirements
  - But still have to be global search to find local information(i.e. what faces share an edge)

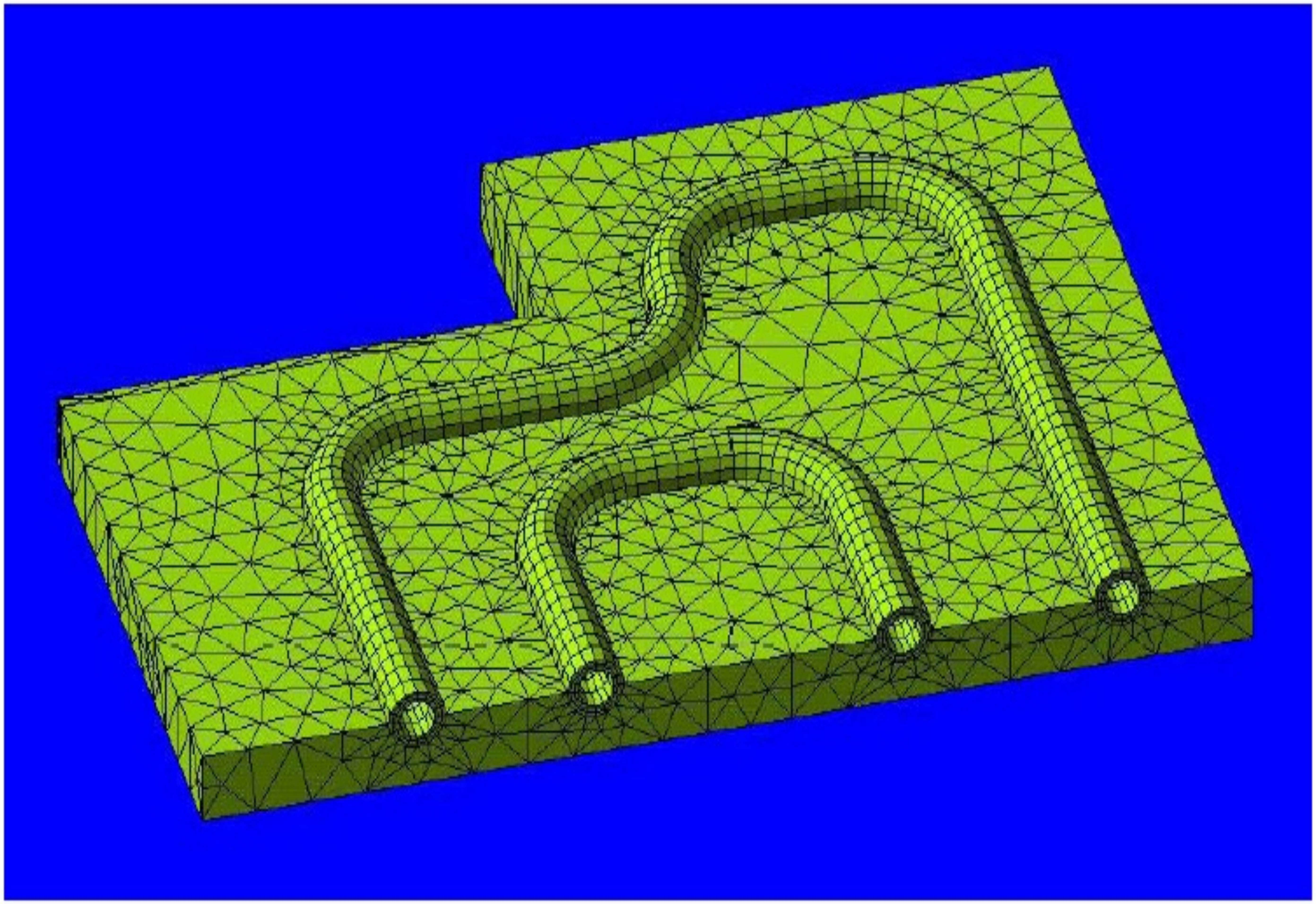
Manifold meshes

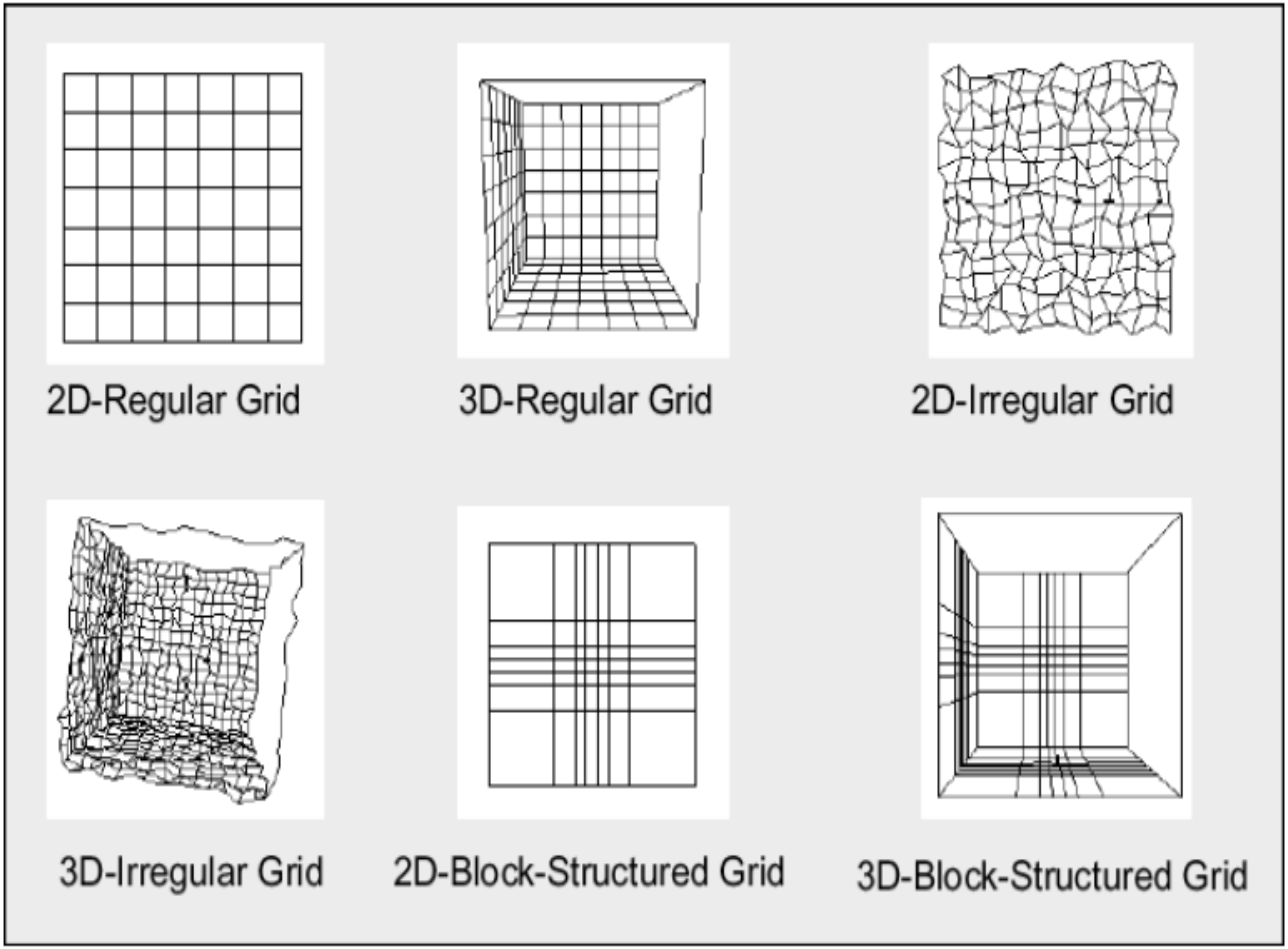
- 2-manifold is a surface where at every point on the surface a surrounding area can be found that looks like a disk
- Everything can be flattened out to a plane
- Sharp creased and edges are possible needs more than one normal per vertex
- Example for an non-manifold:

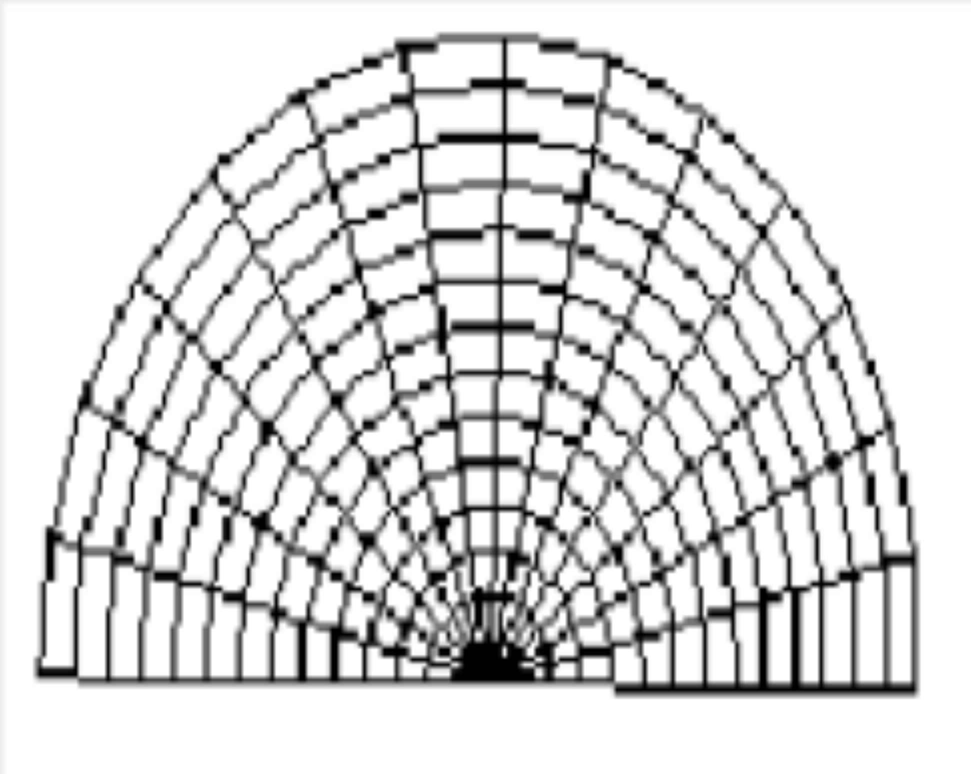


■ Hybrid grids

- ➢ Combination of different grid types

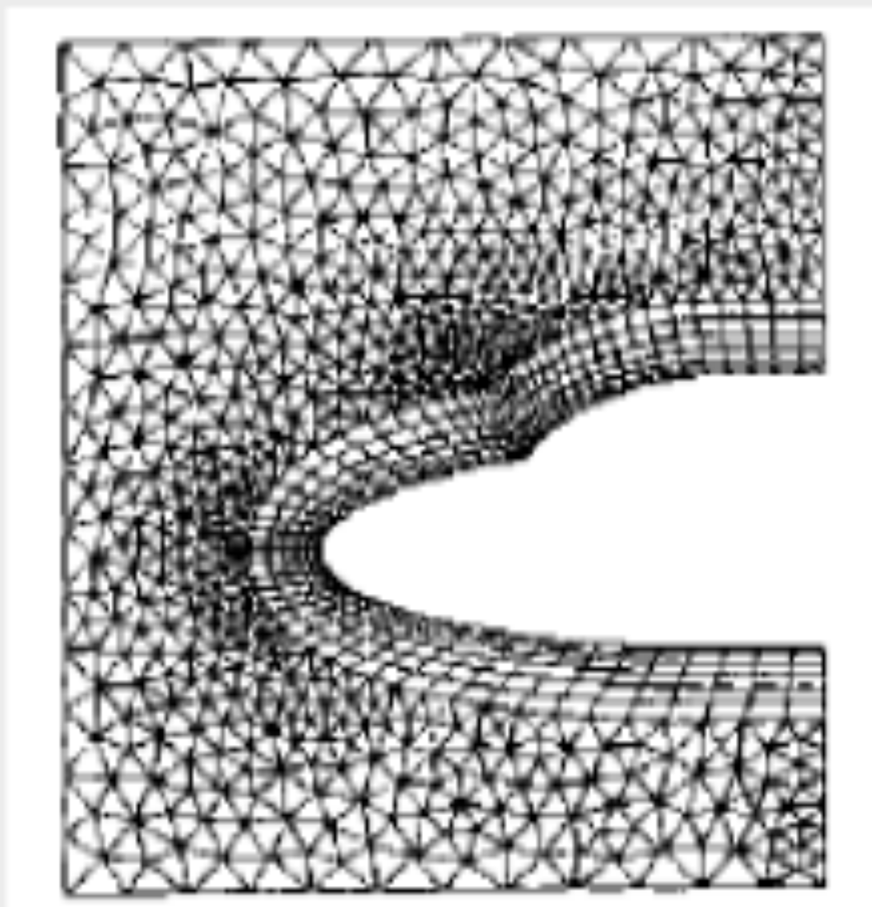



2D-Regular Grid
3D-Regular Grid
2D-Irregular Grid
3D-Irregular Grid
2D-Block-Structured Grid
3D-Block-Structured Grid

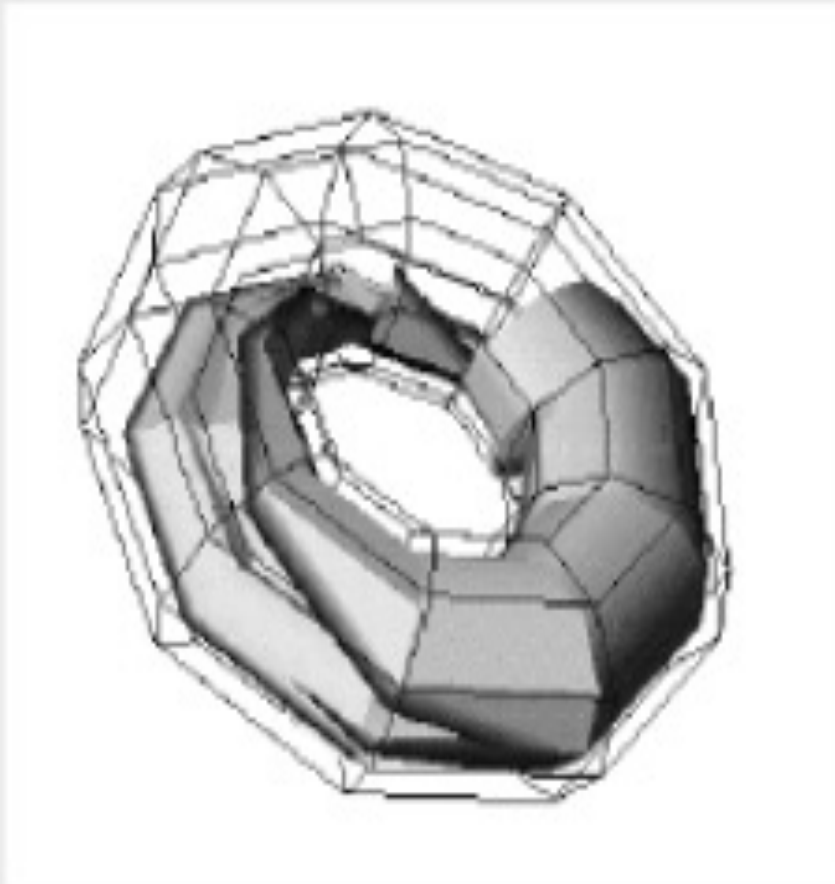


2D-Structured Grid

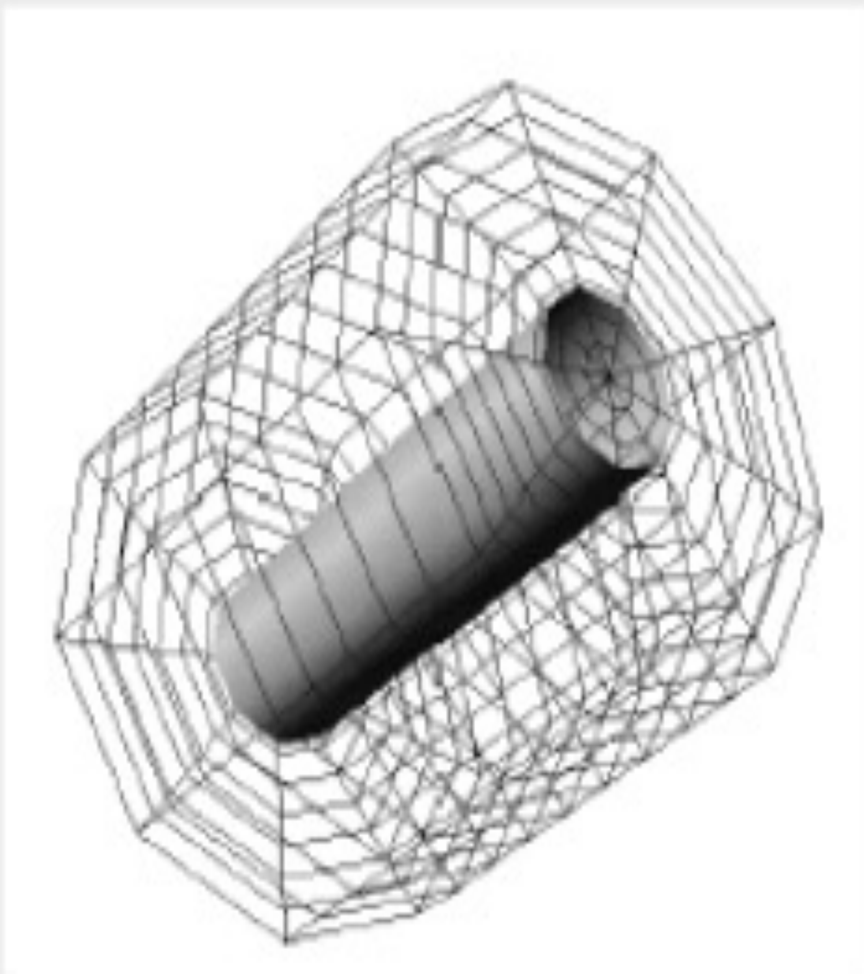
3D-Structured Grid



2D-Hybrid Grid

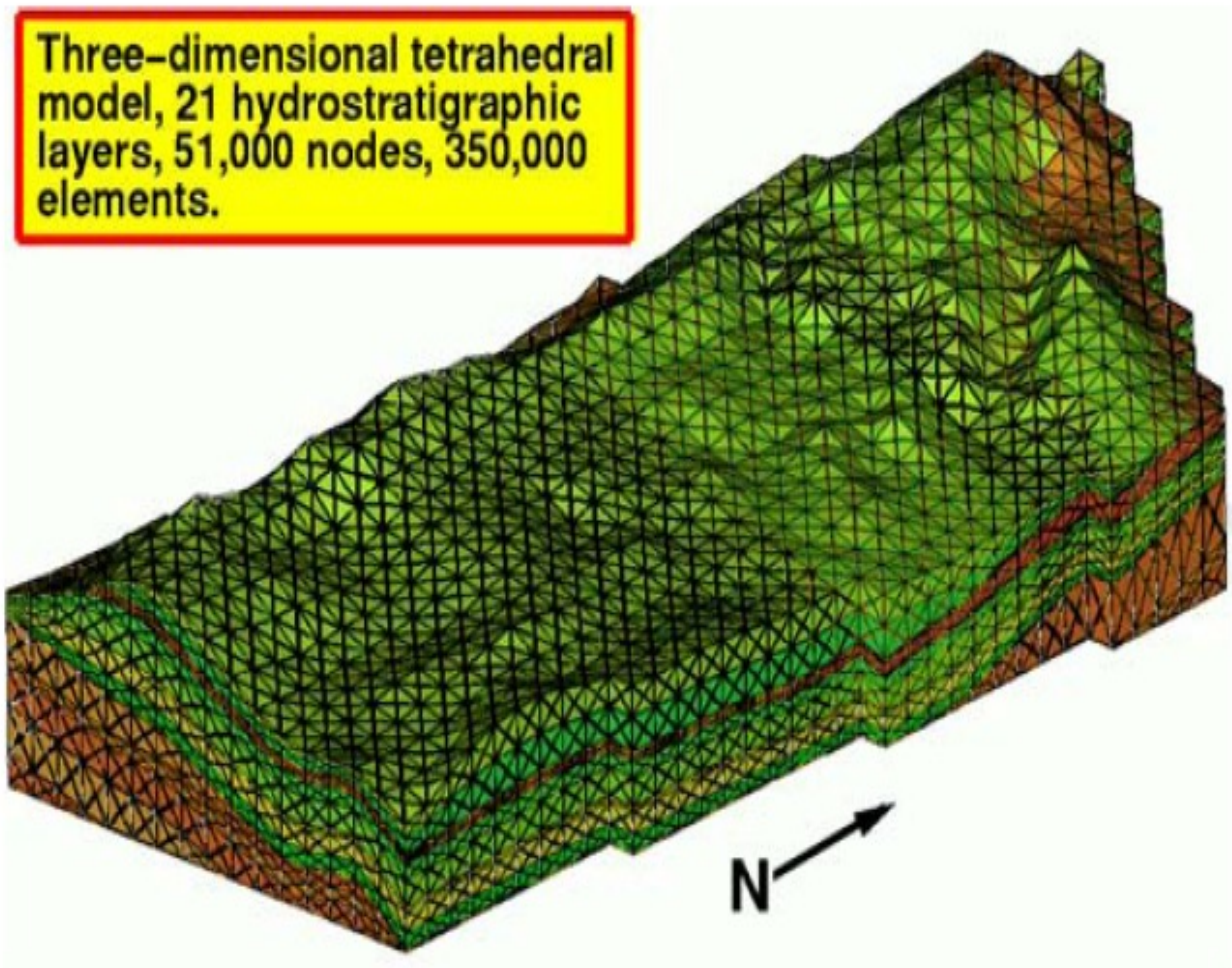


3D-Structured Grid “ Torus”

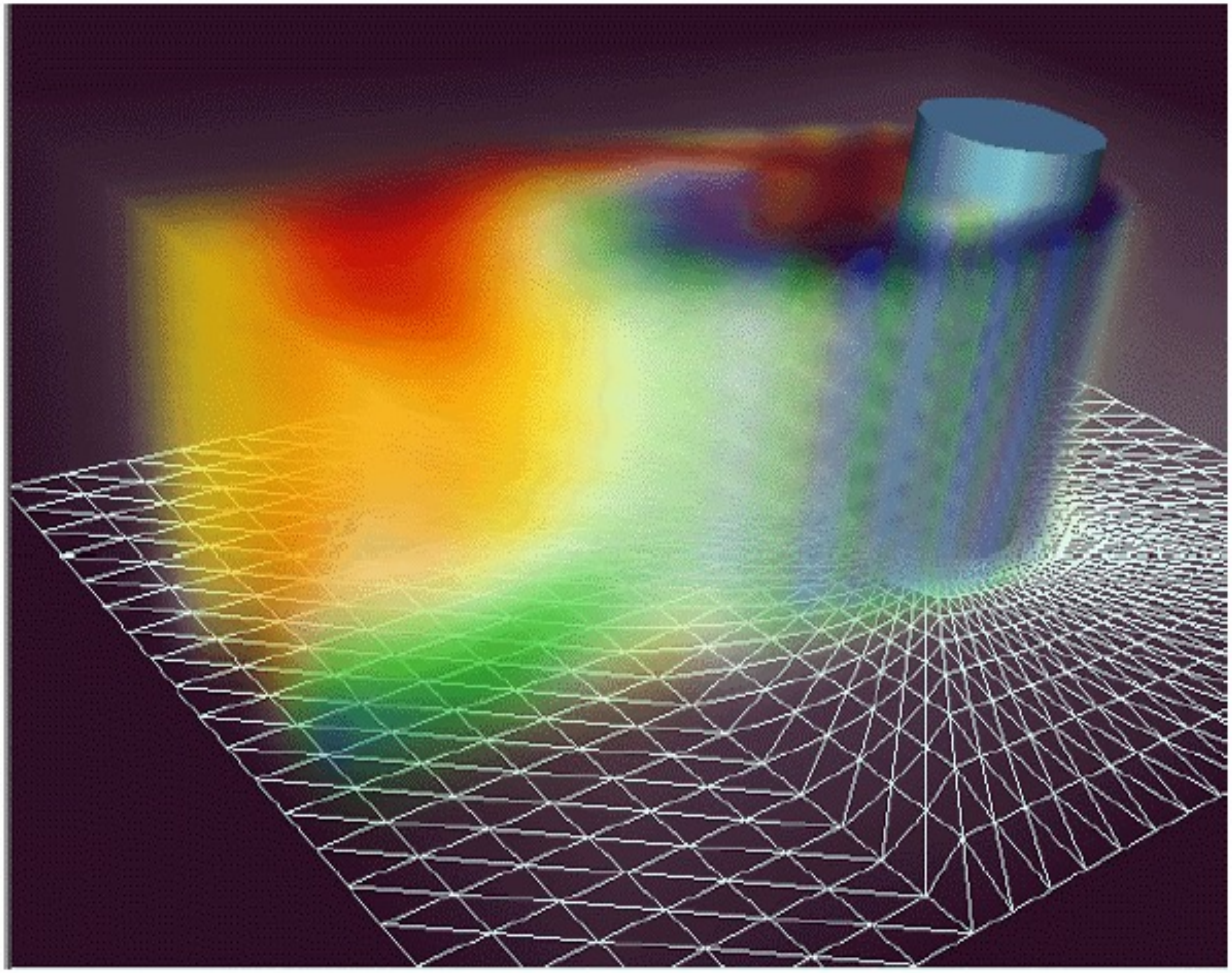


3D-structured Grid “full cylinder”

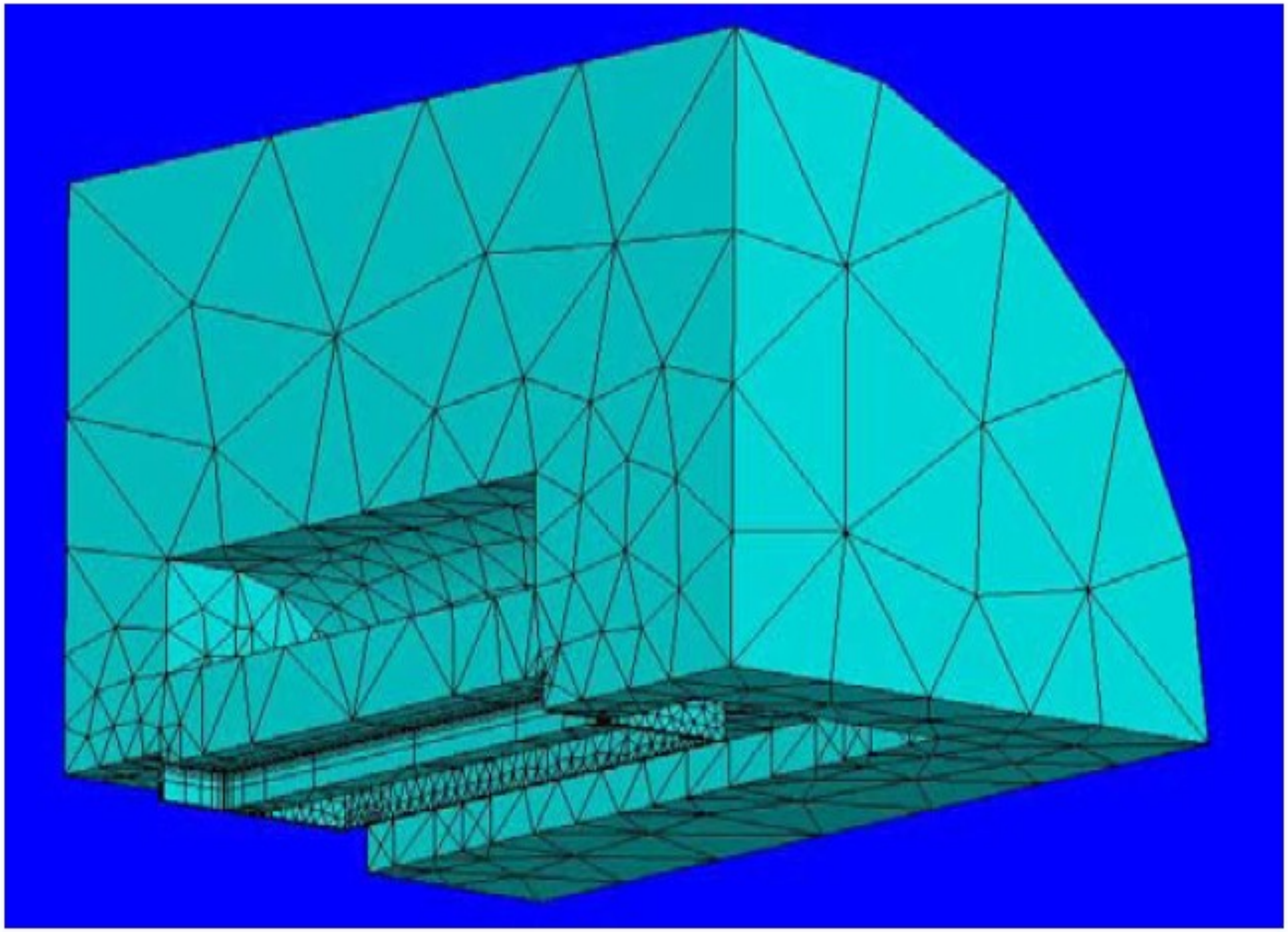
- Example


Three–dimensional tetrahedral model, 21 hydrostratigraphic layers, 51,000 nodes, 350,000 elements.
N

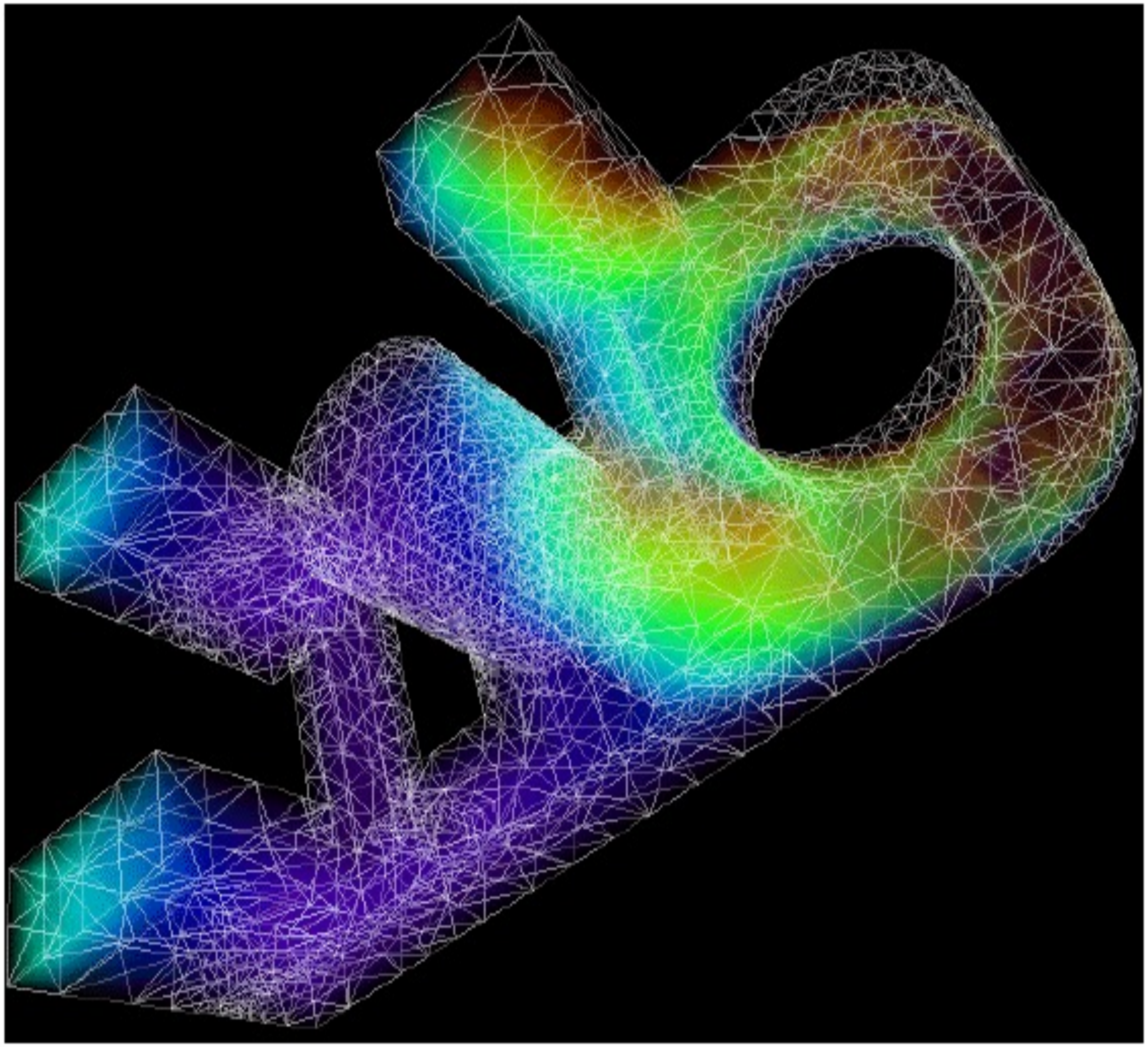
Example



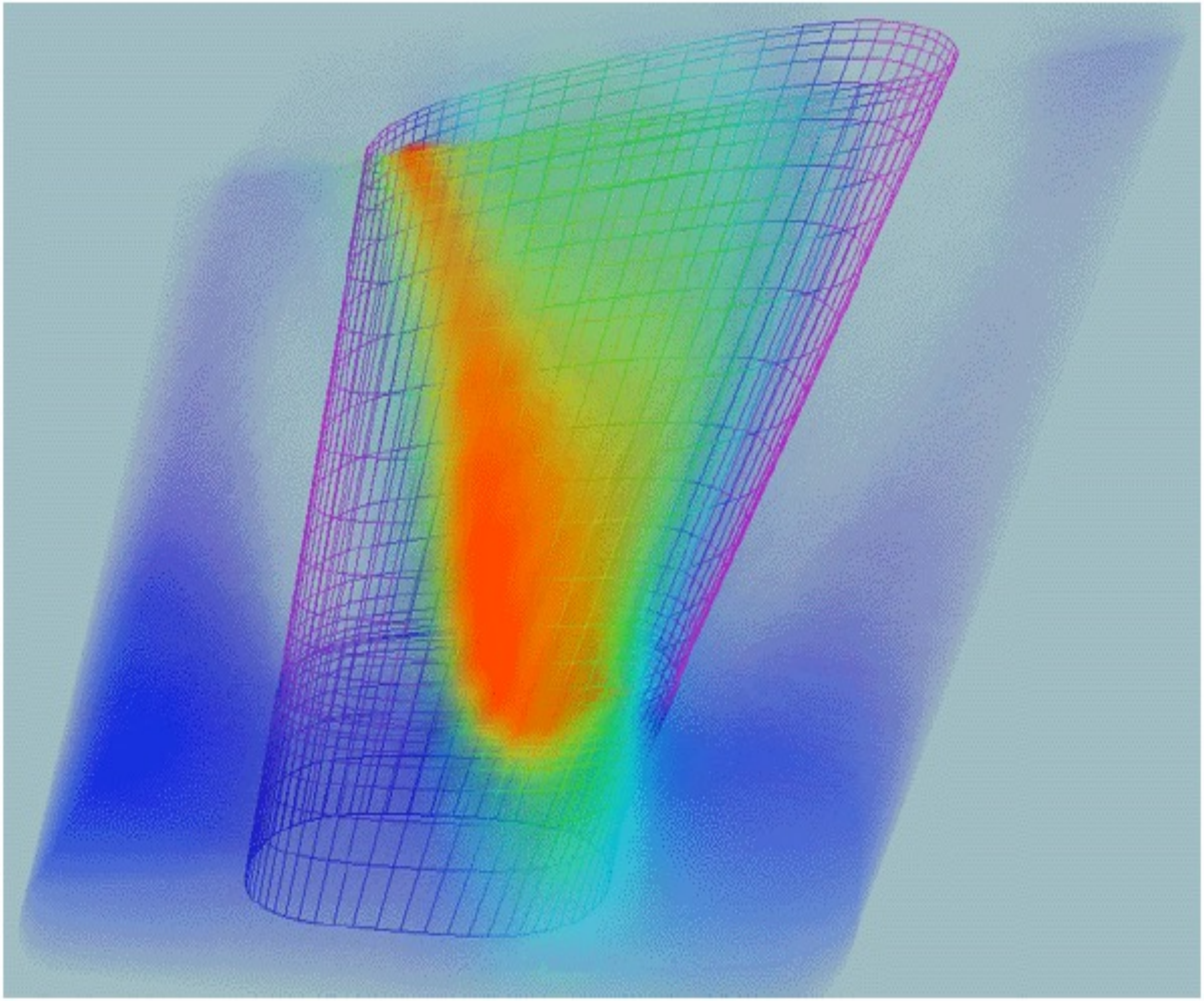
Example



Example

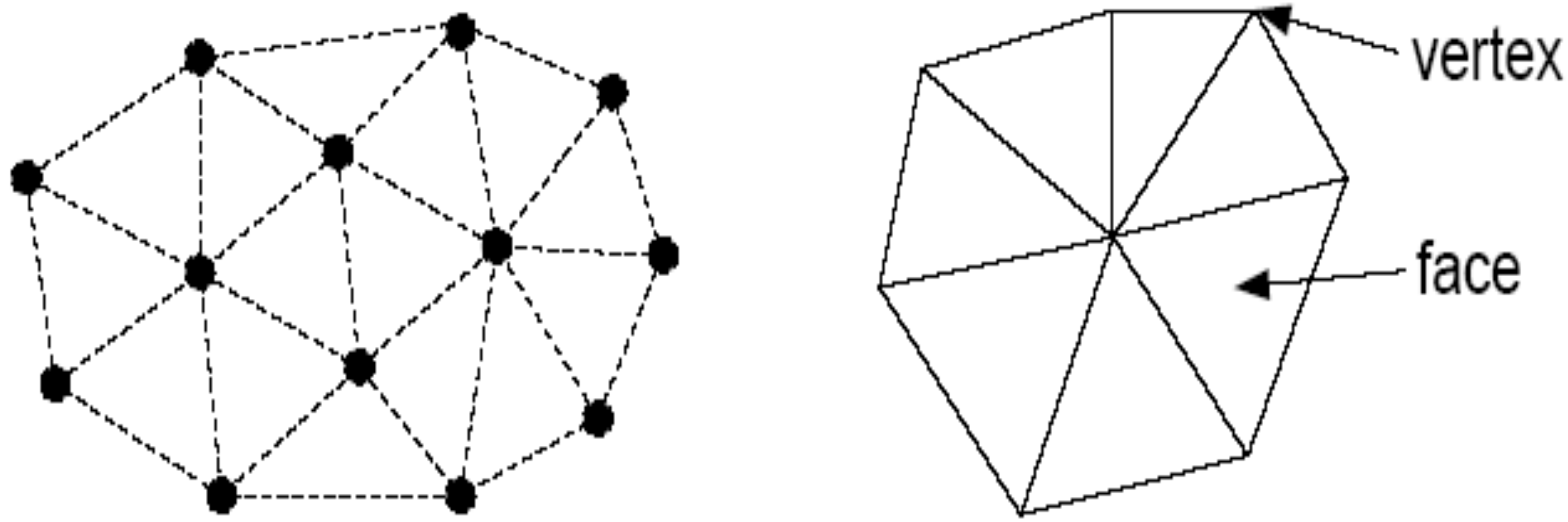


Example



■ Scattered data

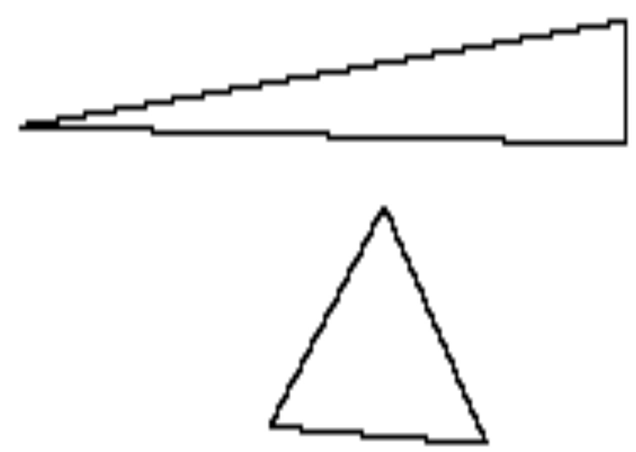
- Irregular distributed positions without connectivity information
- To get connectivity find a “good” triangulation
  (triangular / tetrahedral mesh with scattered points as vertices)

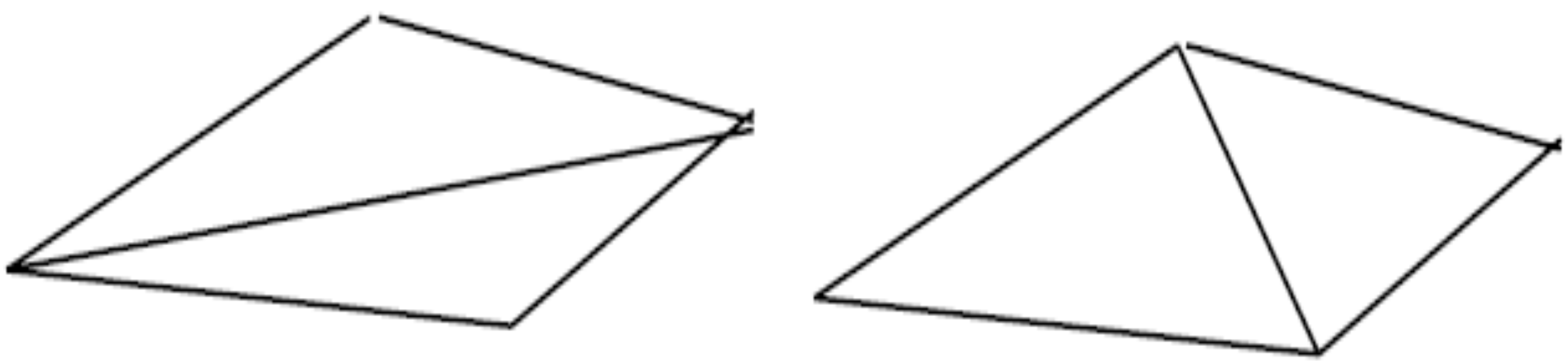


- For a set of points there are many possible triangulations
    - A measure for the quality of a triangulation is the aspect ratio of the so-defined triangles
    - Avoid long , thin ones

    - Delaunay triangulation (later in the course)

$$\frac{\text{radius of incircle}}{\text{radius of circumcircle}} \quad \text{or} \quad \text{maximal/minimal angle in triangle}$$





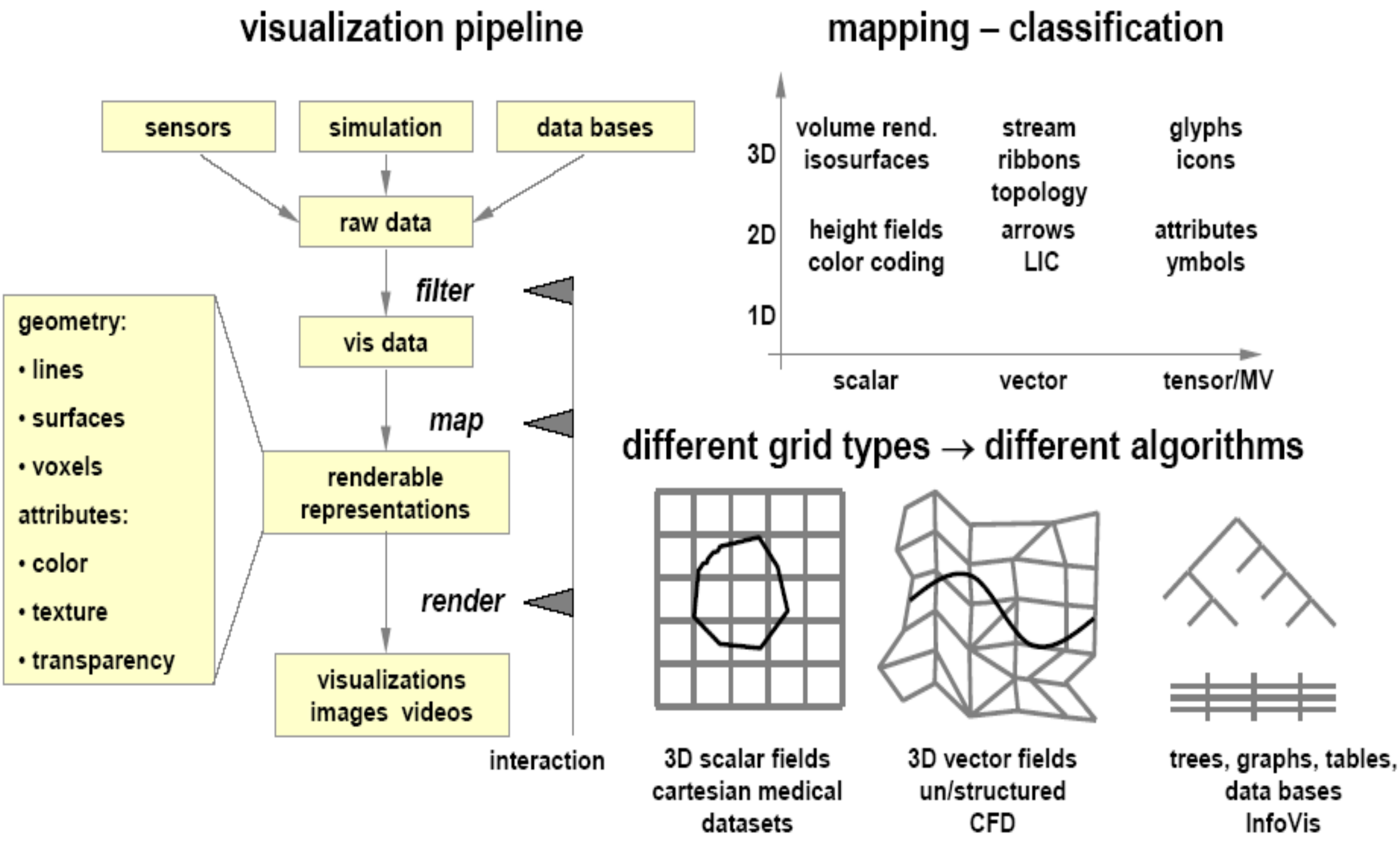
2

Data operation

- Interpolation
- Filtering

- Interpolation
- Filtering

The visualization pipeline



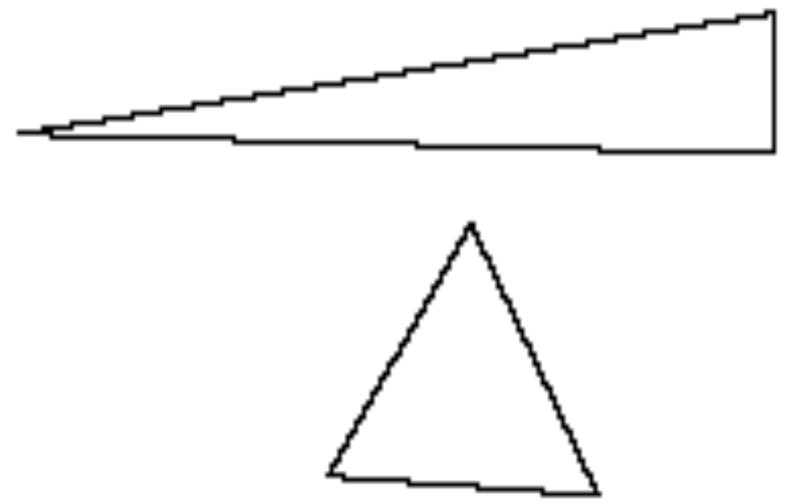
Interpolation and Filtering

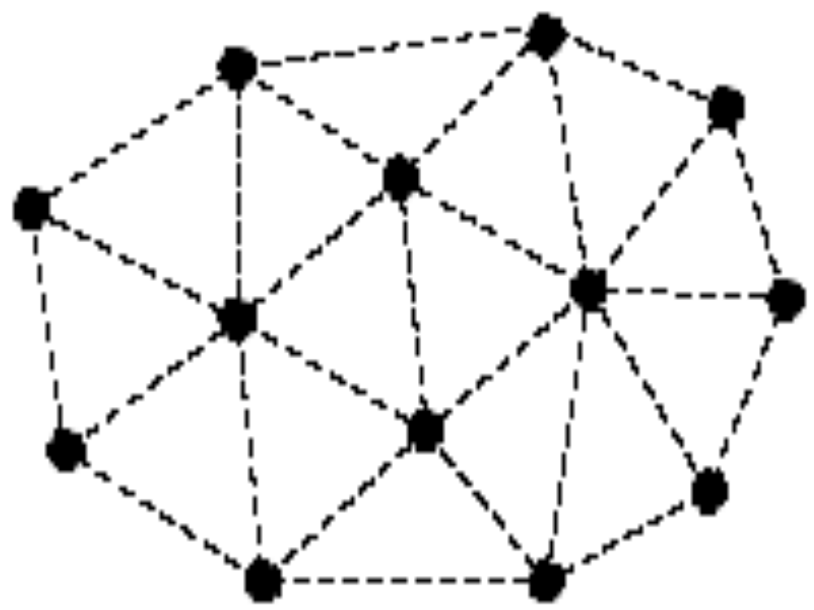
- Data is often discretized in space and / or time
- Finite number of samples
    - The continuous signal is usually known only at a few points (data points)
    - In general , data is needed in between these points
- By interpolation we obtain a representation that matches the function at the data points
    - Evaluation at any other point possible
- Reconstruction of signal at points that are not sampled
- Assumptions needed for reconstruction
    - Often smooth functions

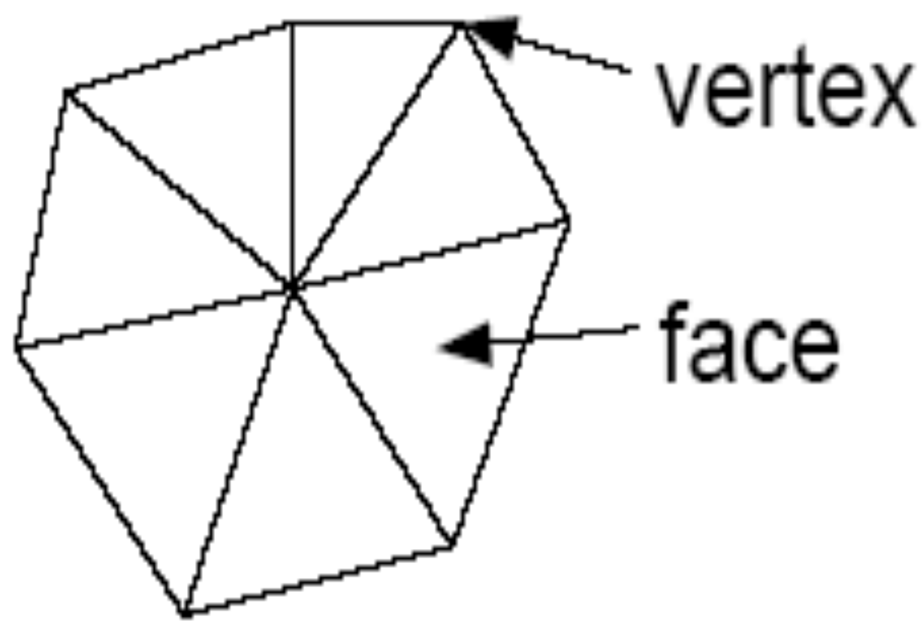
Voronoi Diagrams and Delaunay Triangulation

- Given irregularly distributed positions without connectivity information
- Problem : obtain connectivity to find a “good” triangulation
- For a set of points there are many possible triangulation

  - A measure for the quality of a triangulation is the aspect ratio of the so defined triangles
  - Avoid long , thin ones







Voronoi Diagrams and Delaunay Triangulation

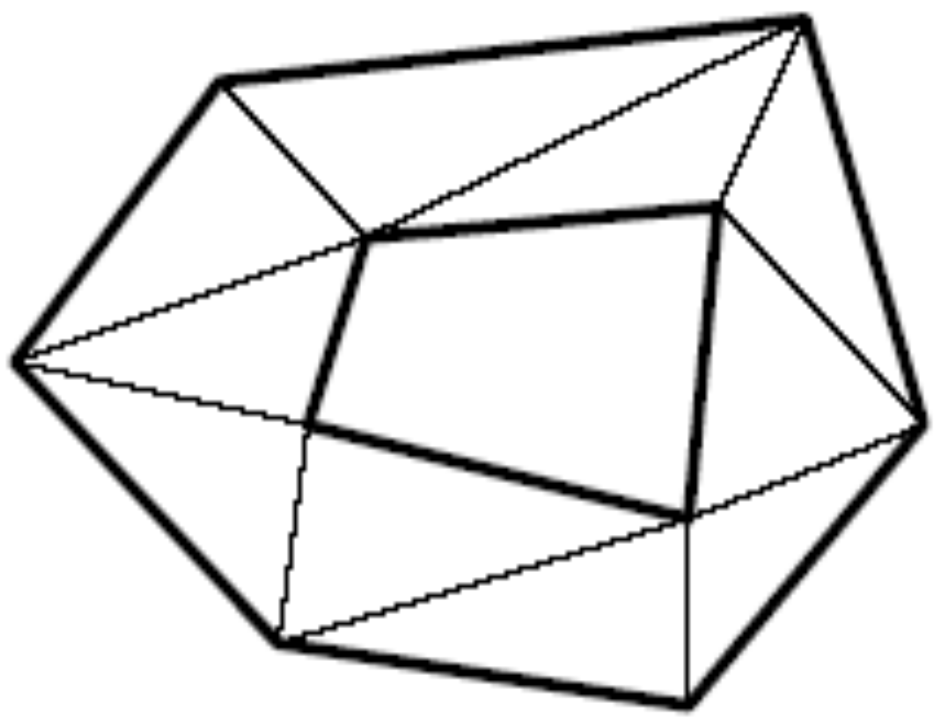
- Scattered data triangulation
  - A triangulation of data points $S = s_0, s_1, \dots, s_m \in R^2$ consists of
    - Vertices(0D)=S
    - Edges(1D)connecting two vertices
    - Faces (2D)connecting three vertices

- a triangulation must satisfy the following criteria
  - Ufaces =conv(S), i.e. the union of all faces including the boundary is the convex hull of all vertices
  - The intersection of two triangles is either empty , or a common vertex , or a common edge , or a common face(tetrahedra)

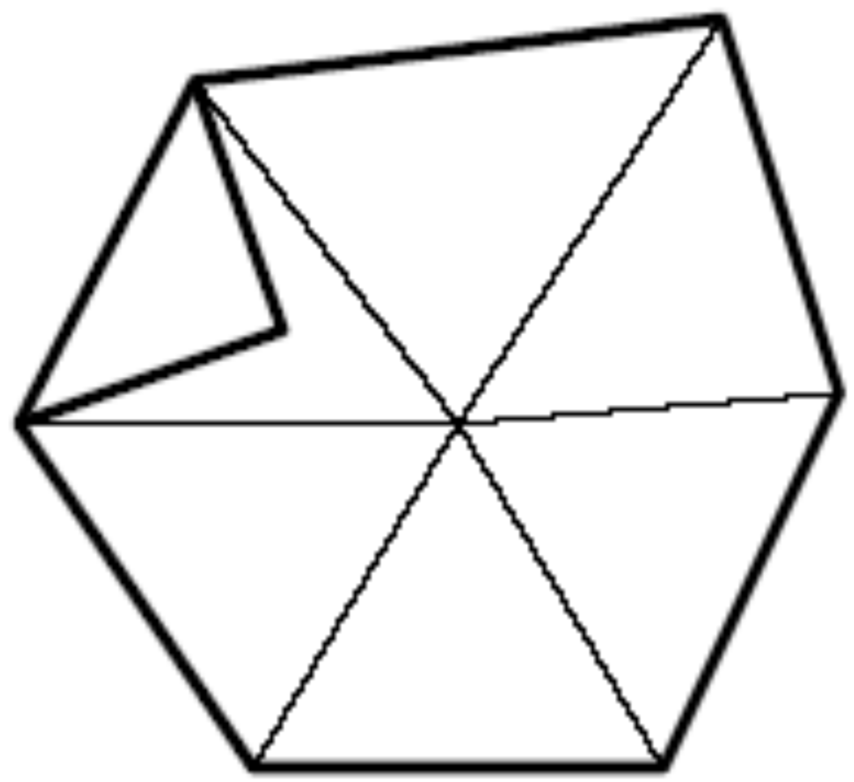
Voronoi Diagrams and Delaunay Triangulation

- Invalid triangulation

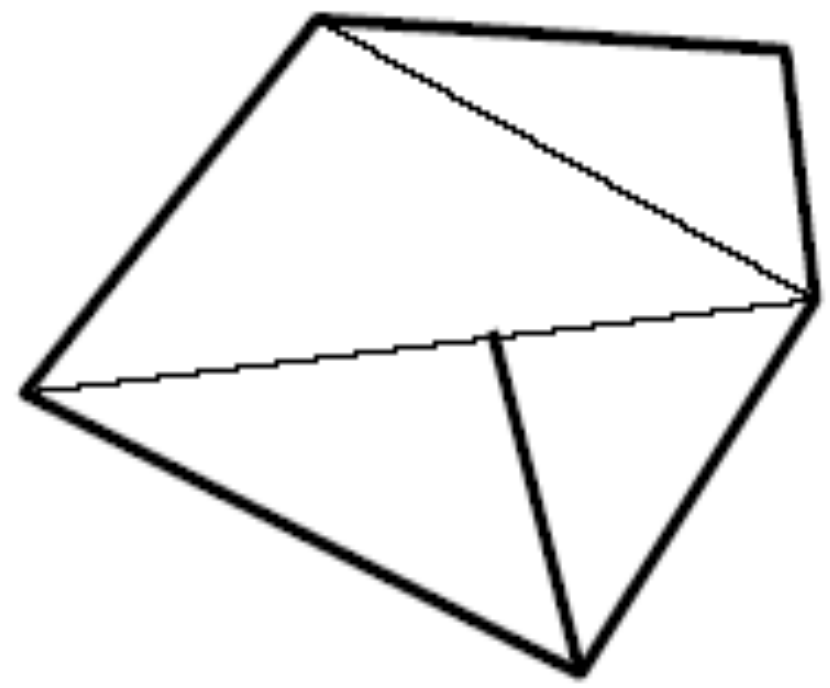
holes,



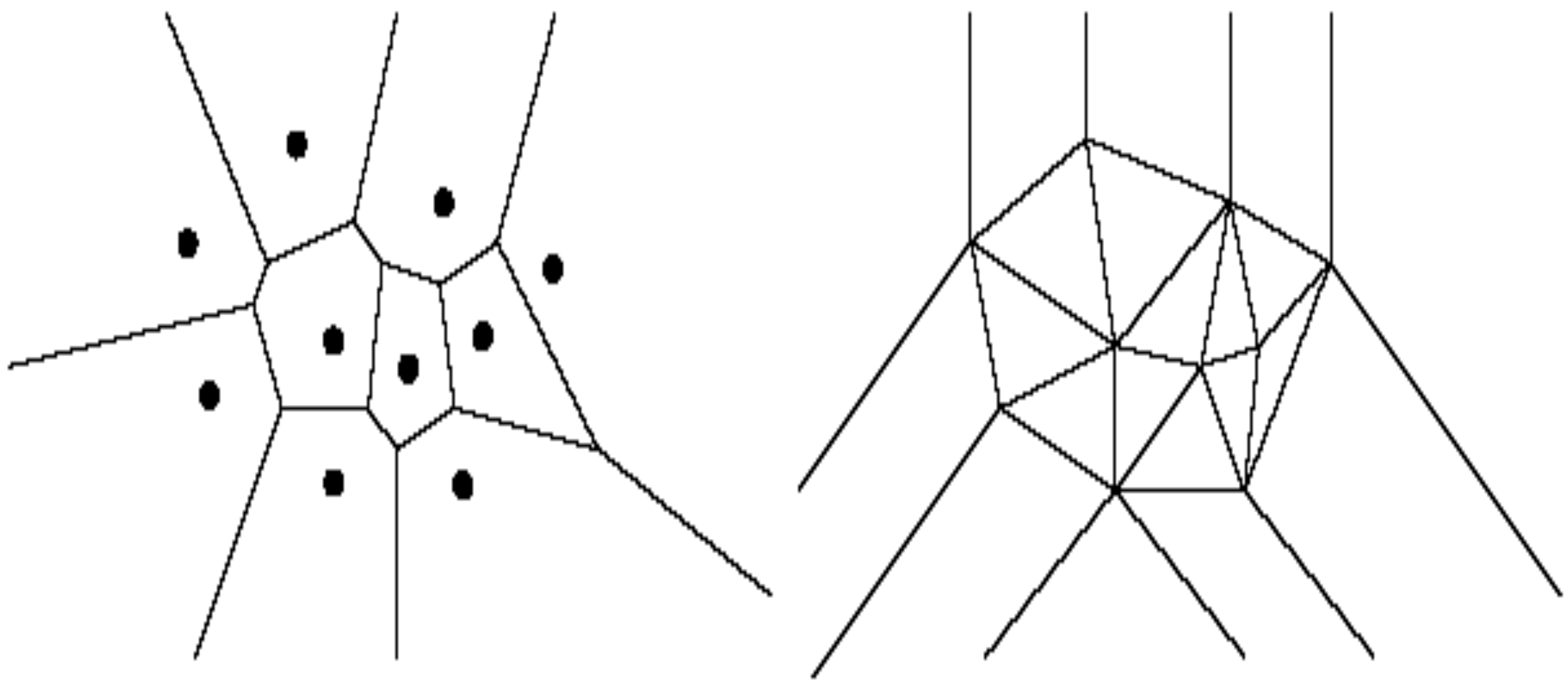
faces overlap,



T-vertices



Voronoi Diagrams and Delaunay Triangulation

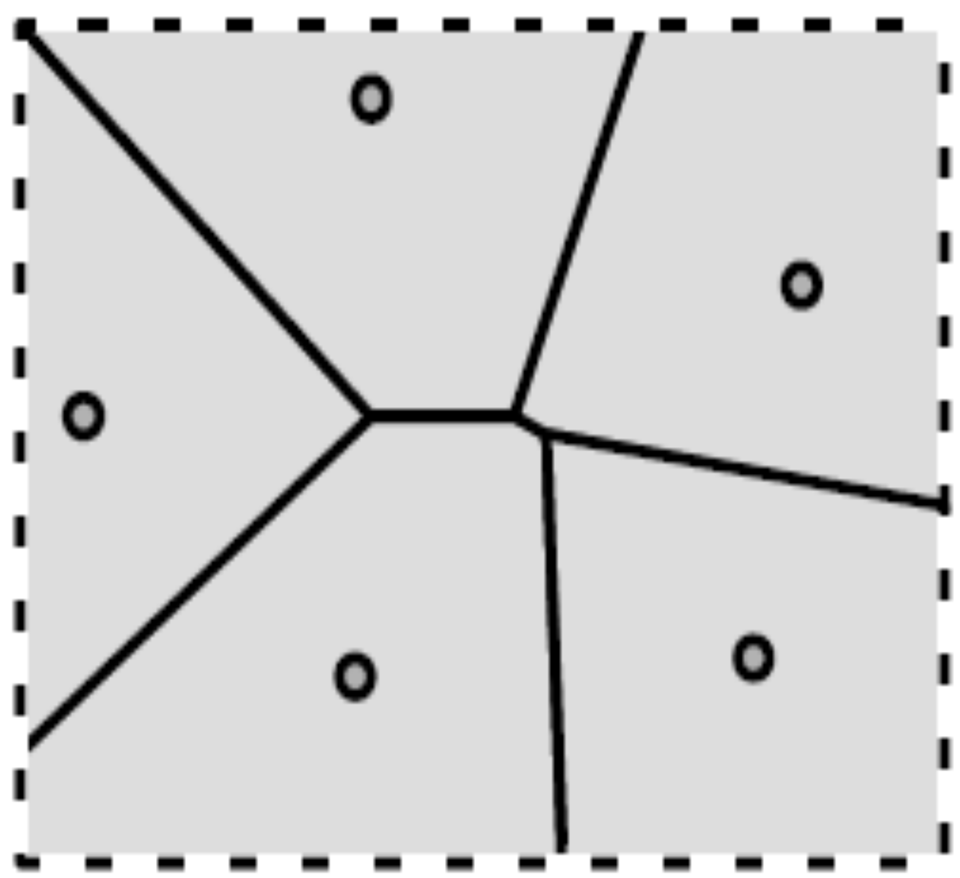


Voronoi Diagrams and Delaunay Triangulation

Voronoi diagram

- For each sample every point within a voronoi region is closer to it than to every other sample
- Given : a set of points $X = \{x_1, \dots, x_n\}$from R^d and a distance function dist(x,y)
- The voronoi diagram $Vor(X)$contains for each point x_i a cell V(x_i)with

  - V(x_i) $= \{x | dist(x, x_i) < dist(x, x_j) \forall j \neq i\}$

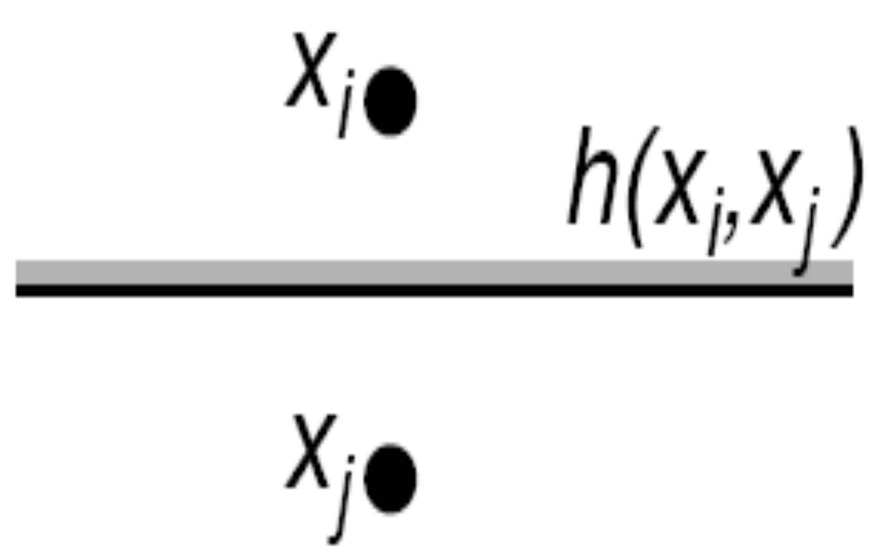


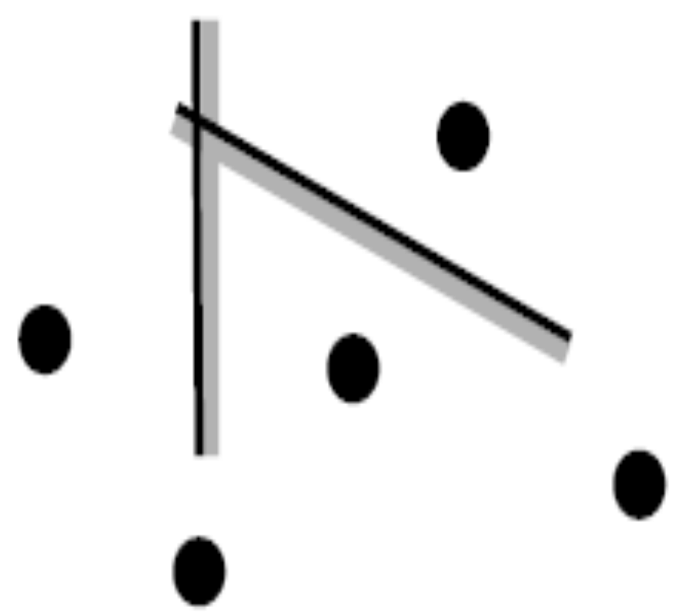
Voronoi Diagrams and Delaunay Triangulation

- Voronoi cells
  - The half space $h(x_i, x_j)$is separated by the perpendicular bisector between x_i and x_j
  - $h(x_i, x_j)$contais x_i
  - Voronoi cell

$$V(x_i) = \cap_{j \neq i} h(x_i, x_j)$$

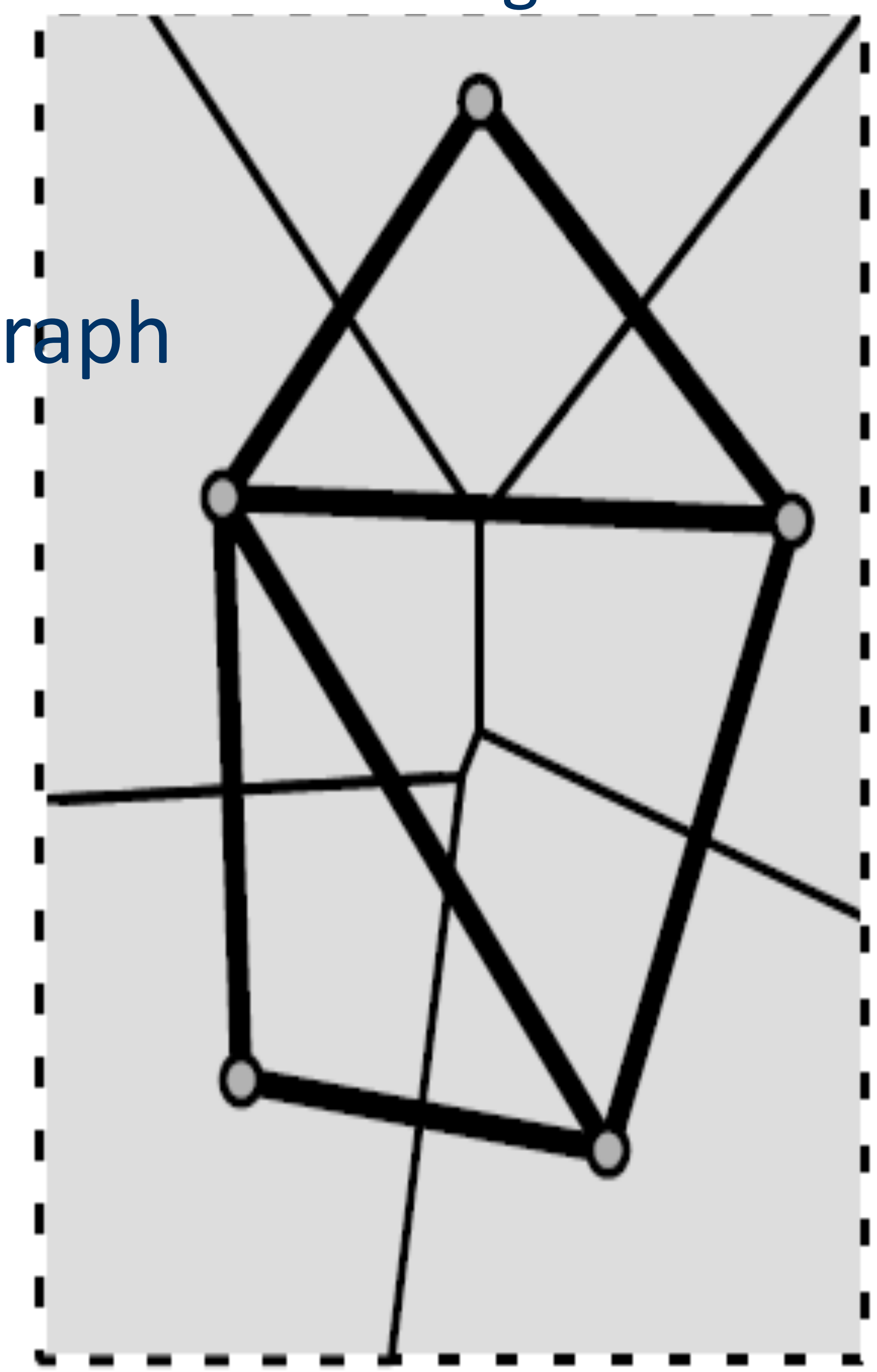
  - Voronoi cells are convex





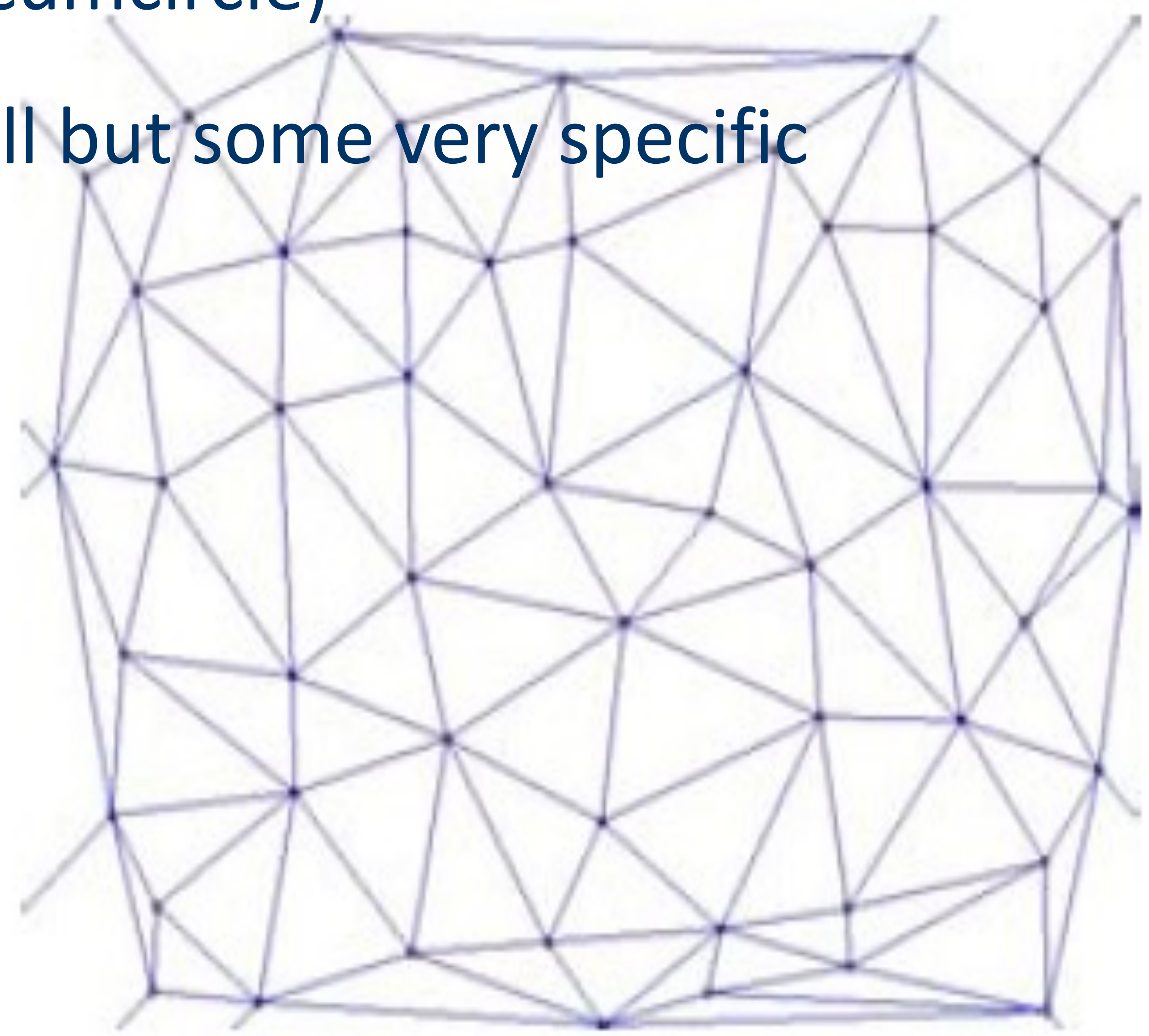
Voronoi Diagrams and Delaunay Triangulation

- Delaunay graph $Del(X)$
  - The geometric dual (topologically equal) of the Voronoi diagram $Vor(X)$
  - Points in X are nodes
  - Two nodes x_i and x_j connected iff the Voronoi cells $V(x_j)$ share a same edge
- Delaunay cells are convex
- Delaunay triangulation=triangulation of the Delaunay graph



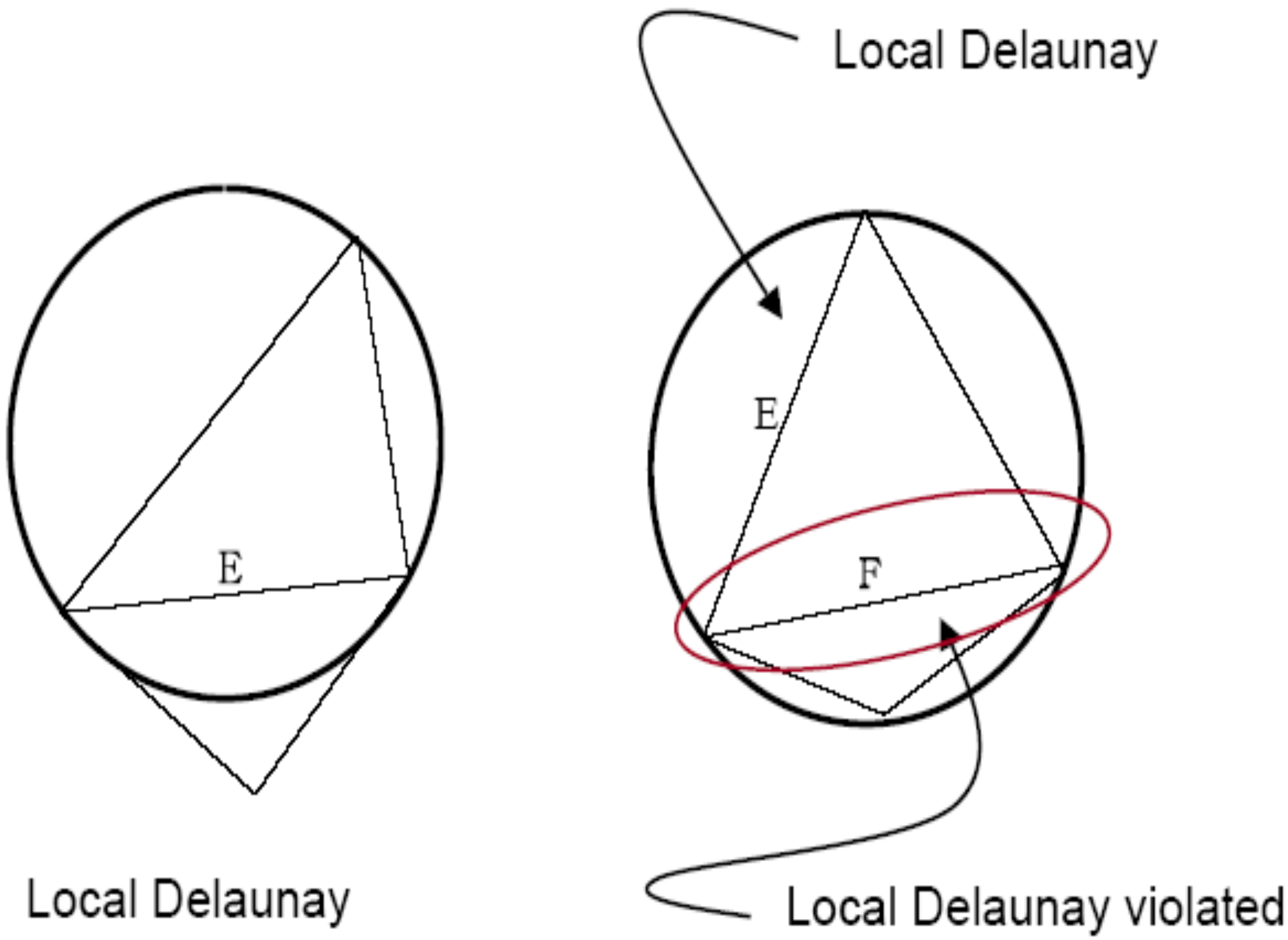
Voronoi Diagrams and Delaunay Triangulation

- Delaunay triangulation in 2D
  - Three points x_i, x_j, x_k in X belong to a face from Del(X)
    iff no further point lies inside the circle around x_i, x_j, x_k
  - Two points x_i, x_j form an edge
    iff there is a circle around x_i, x_j that does not contain a third point from X
  - For each triangle the circumcircle does not contain any other sample
  - Maximized the smallest angle
  - Maximized the ratio of (radius of incircle)/(radius of circumcircle)
  - It is unique (independent of the order of samples) for all but some very specific cases



Voronoi Diagrams and Delaunay Triangulation

- Local delaunay property



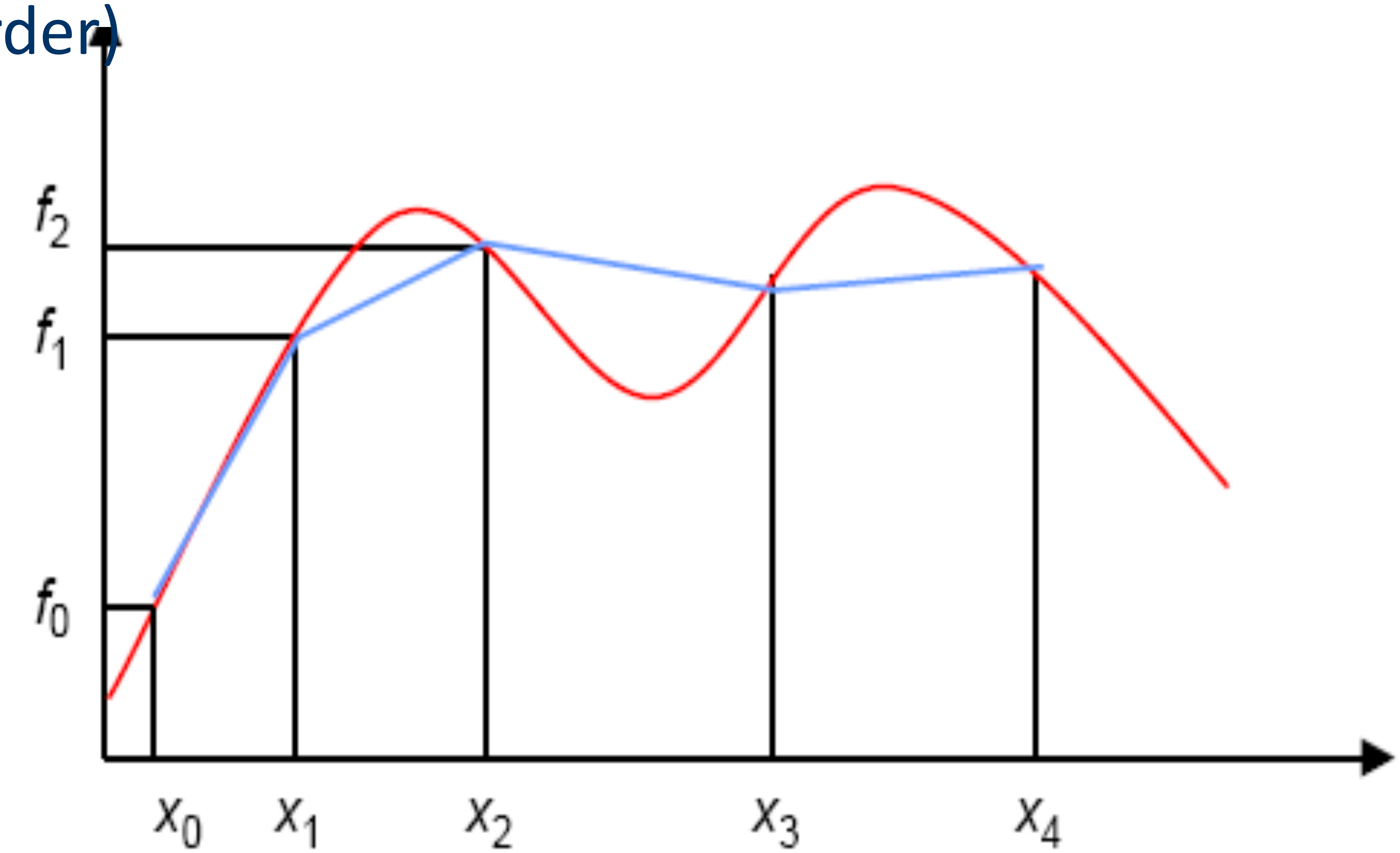
Voronoi Diagrams and Delaunay Triangulation

- Algorithm

  - Get the detailed algorithm from the Internet if you are interest

Univariate Interpolation

- Univariate interpolation : interpolation for one variable
  - Nearest neighbor (0 order)
  - Linear (first order)
  - Smooth (higher order)



Univariate Interpolation

- Taylor interpolation
- Basis functions : monomer basis (polynomials)
  - $m_i = x^i$, with $i \epsilon N_0$
- $P^m = \{1, x, x^2, \dots, x^m\}$is m+1-dimensional vector space of all polynomials with maximum degree m
- Coefficients c_i with $f = \sum_i c_i \cdot x^i$
- Representation of samples :
  - $f(x_j) = f_j \; \forall j = 1 \dots n$
- Interpolation problem

samples

$$\boldsymbol{V} \cdot \boldsymbol{c} = \boldsymbol{f}$$

coefficients (to be solved)

with the Vandermond matrix $\boldsymbol{V}_{ij} = x_i^{\,j-1}$

Univariate Interpolation

- Generic interpolation problem
  - given are n sampled points $X = \{x_i\} \subseteq \Omega \subseteq R^d$ with function values f_i
  - n-dimensional function space $\Phi_n{}^d(\Omega)$ with basis $\{\phi_{i=1,\ldots,n}\}$
  - representation of samples:

$$f(x_j) = f_j \quad \forall j = 1..n$$

  - Solving the linear system of equations

$$\boldsymbol{M} \cdot \boldsymbol{c} = \boldsymbol{f}$$

$$\text{With } \boldsymbol{M}_{ji} = \phi_i\,(X_j), \boldsymbol{c}_i = c_i, \text{and } \boldsymbol{f}_j = f_j$$

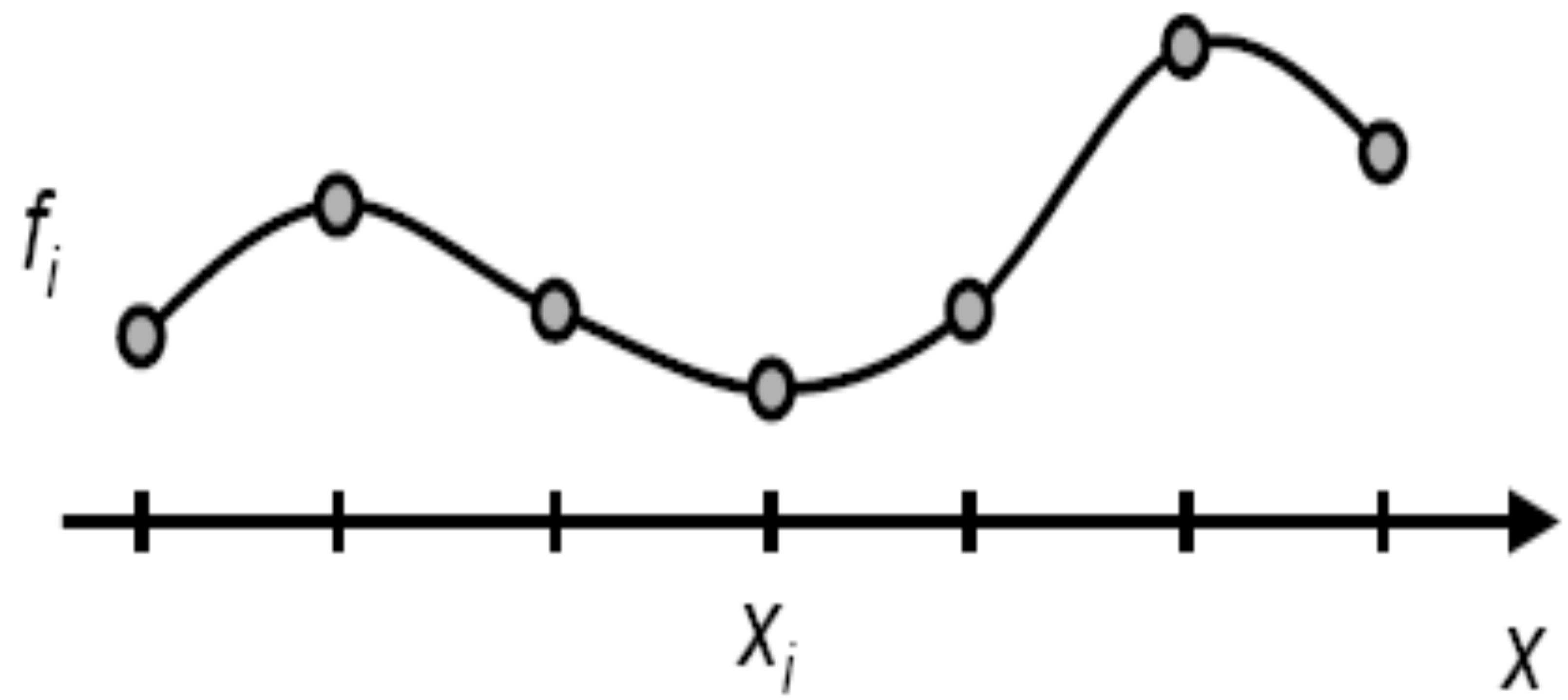
  - Note: number of points n determines dimension of vector space (=degree of polynomials)

Univariate Interpolation

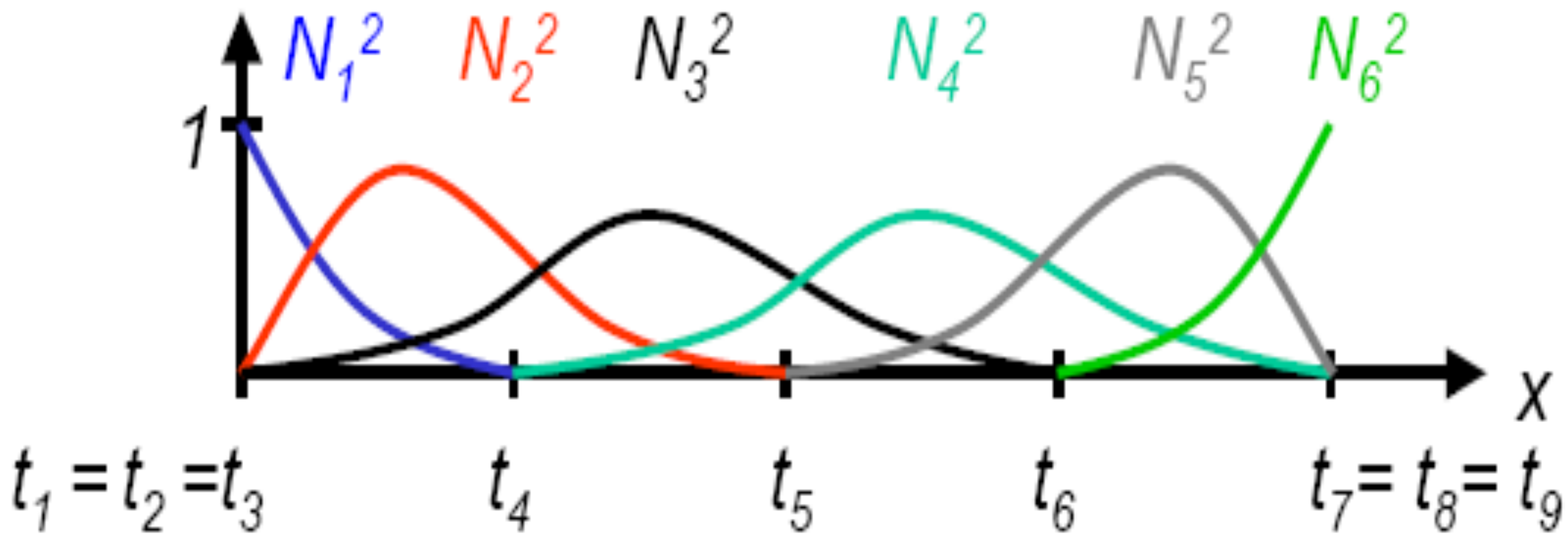
- Other basis functions result in other interpolations schemes :
  - Lagrange interpolation
  - Newton interpolation
  - Bernstein basis : Bezier curves(approximation)
  - Hermite basis

Univariate Interpolation

- Problem： coupling of number of samples n and degree of polynomials



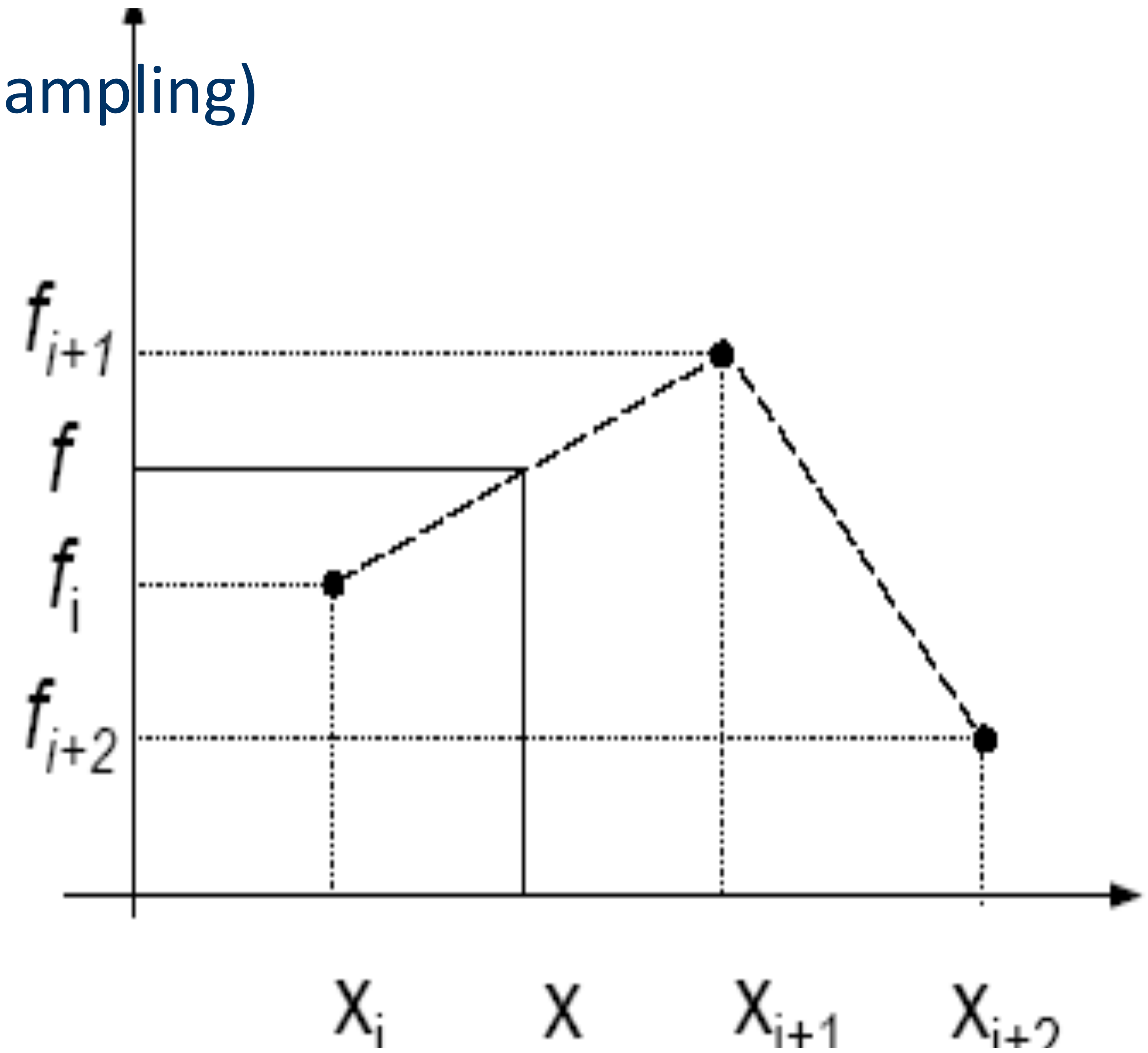
- Solution: spline interpolation
  - Smooth piecewise polynomial function
  - Continuity / smoothness at segment boundaries!



B-Spline

Univariate Interpolation

- Piecewise linear interpolation
  - Simplest approach(except for nearest-neighbor sampling)
  - Fast to compute
  - Often used in visualization applications
  - C^0 continuity at segment boundaries
  - Data points : $(x_0, f_0), \dots, (x_n, f_n)$
  - For any point x with

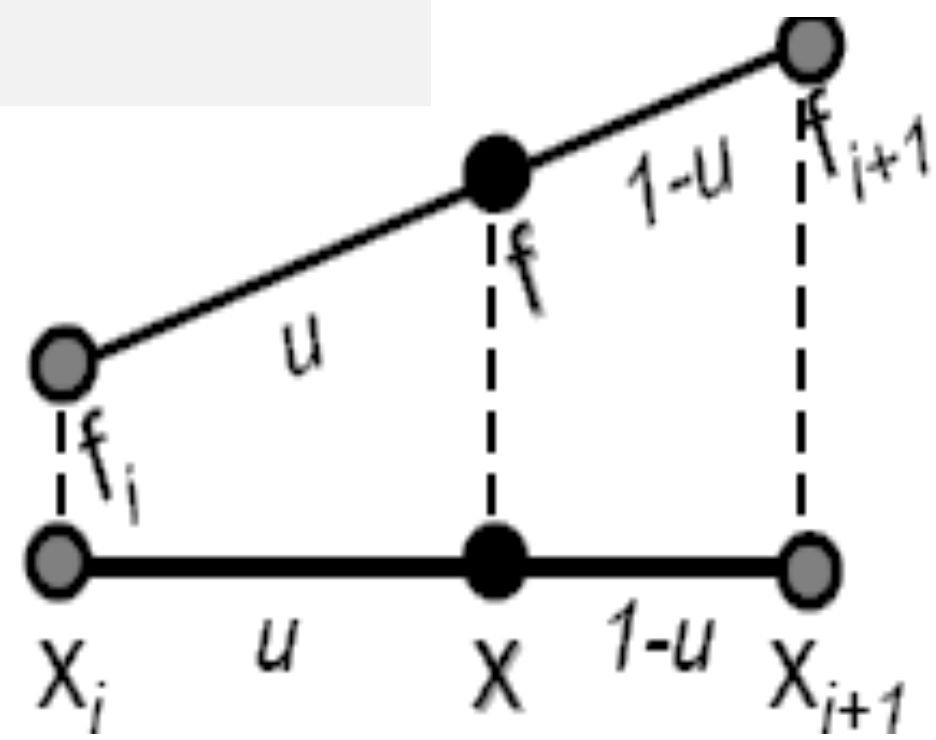


$$x_i \le x \le x_{i+1}$$

Described by local coordinate $u = \frac{x-x_i}{x_{i+1}-x_i} \in [0,1]$;

that is $x = x_i + u(x_{i+1} - x_i) = (1-u)x_i$+u$x_{i+1}$;

evaluate $f(x) = (1-u)f_i$+uf_{i+1};



Differentiation on Grids

- First approach
  - Replace differential by finite differences
  - Note that approximating the derivative by

$$f'(x) = \frac{df}{dx} \rightarrow \frac{\Delta f}{\Delta x}$$

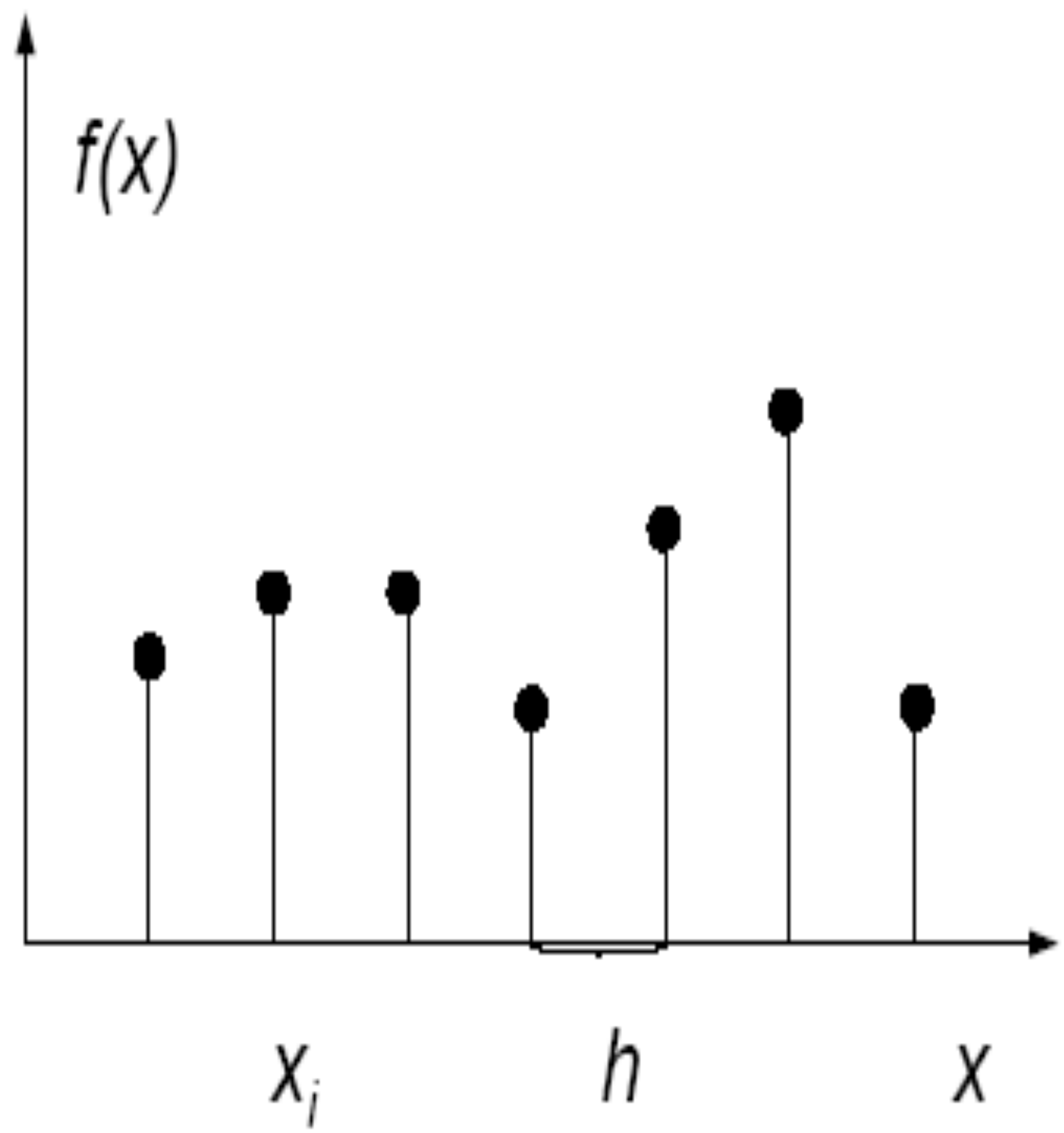
Caused subtractive cancellation and large rounding errors for small h

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- Second approach
  - Approximate / interpolate(locally) by differentiable function and differentiate this function

Differentiation on Grids

- Finite differences on uniform grids with grid size h(1D case)



Differentiation on Grids

- Finite differences on uniform grids with grid size h (1D case)
  - Forward differences

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$$

  - Backward differences

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h}$$

  - Cnetral differences

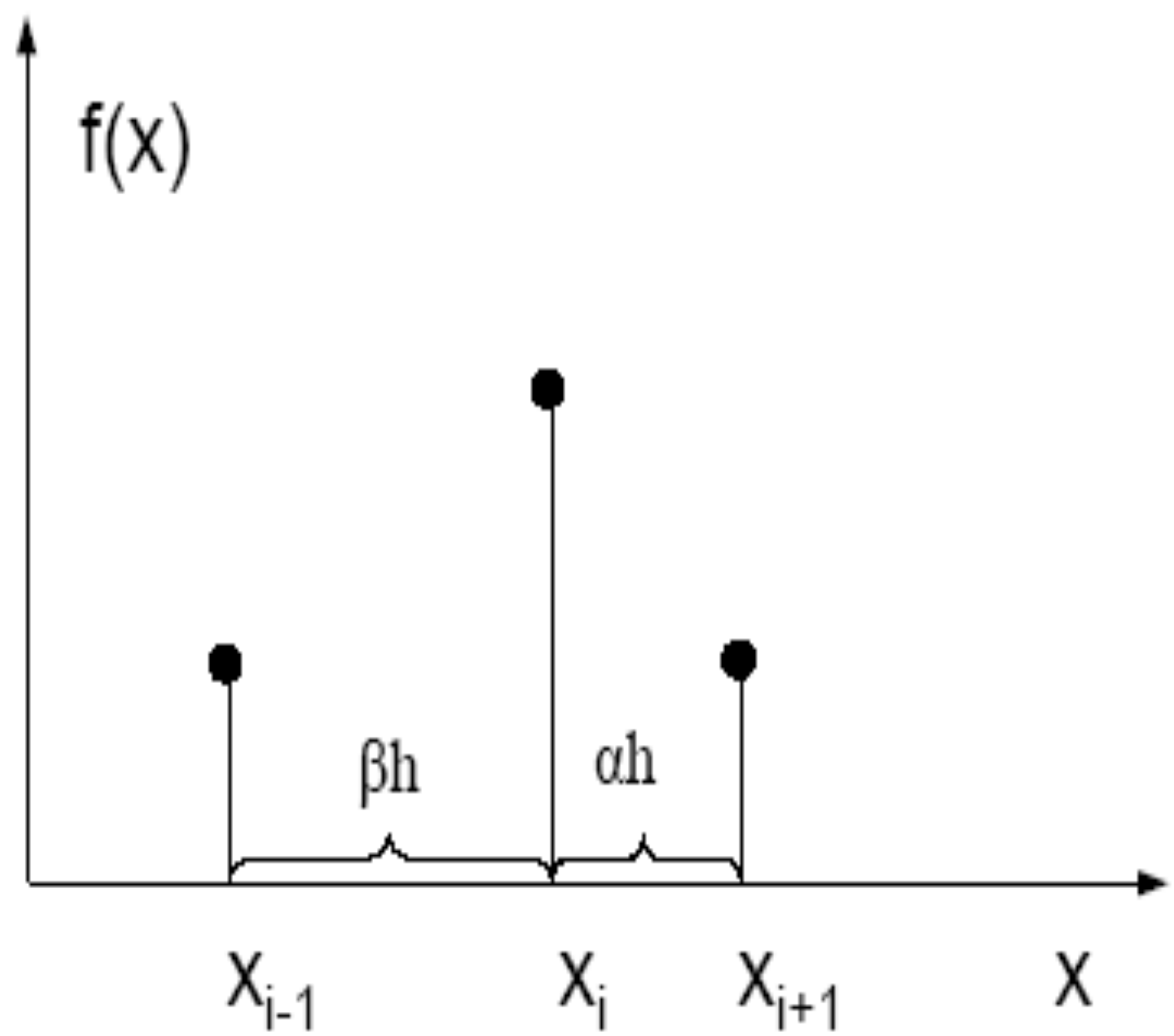
$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

- Error estimation
  - Forward / backward differences are first order
  - Central differences are second order

Differentiation on Grids

- Finite differences on non-uniform grids
  - Forward and backward differences as for uniform grids with

$$(x_{i+1} - x_i = \alpha h)$$
$$(x_i - x_{i-1} = \beta h)$$



Differentiation on Grids

- Finite differences on non-uniform grids
  - Central differences by Taylor expansion around the point x_i

$$f(x_{i+1}) = f(x_i) + \alpha h f'(x_i) + \frac{(\alpha h)^2}{2} f''(x_i) + \ldots$$

$$f(x_{i-1}) = f(x_i) - \beta h f'(x_i) + \frac{(\beta h)^2}{2} f''(x_i) + \ldots$$

$$\Rightarrow \frac{1}{\alpha^2}(f(x_{i+1}) - f(x_i)) - \frac{1}{\beta^2}(f(x_{i-1}) - f(x_i)) = \frac{h}{\alpha} f'(x_i) + \frac{h}{\beta} f'(x_i) + O(h^3)$$

  - The final approximation of the derivative

$$f'(x_i) = \frac{1}{h(\alpha+\beta)}\left(\frac{\beta}{\alpha} f(x_{i+1}) - \frac{\alpha}{\beta} f(x_{i-1}) + \frac{\alpha^2 - \beta^2}{\alpha - \beta} f(x_i)\right)$$

Differentiation on Grids

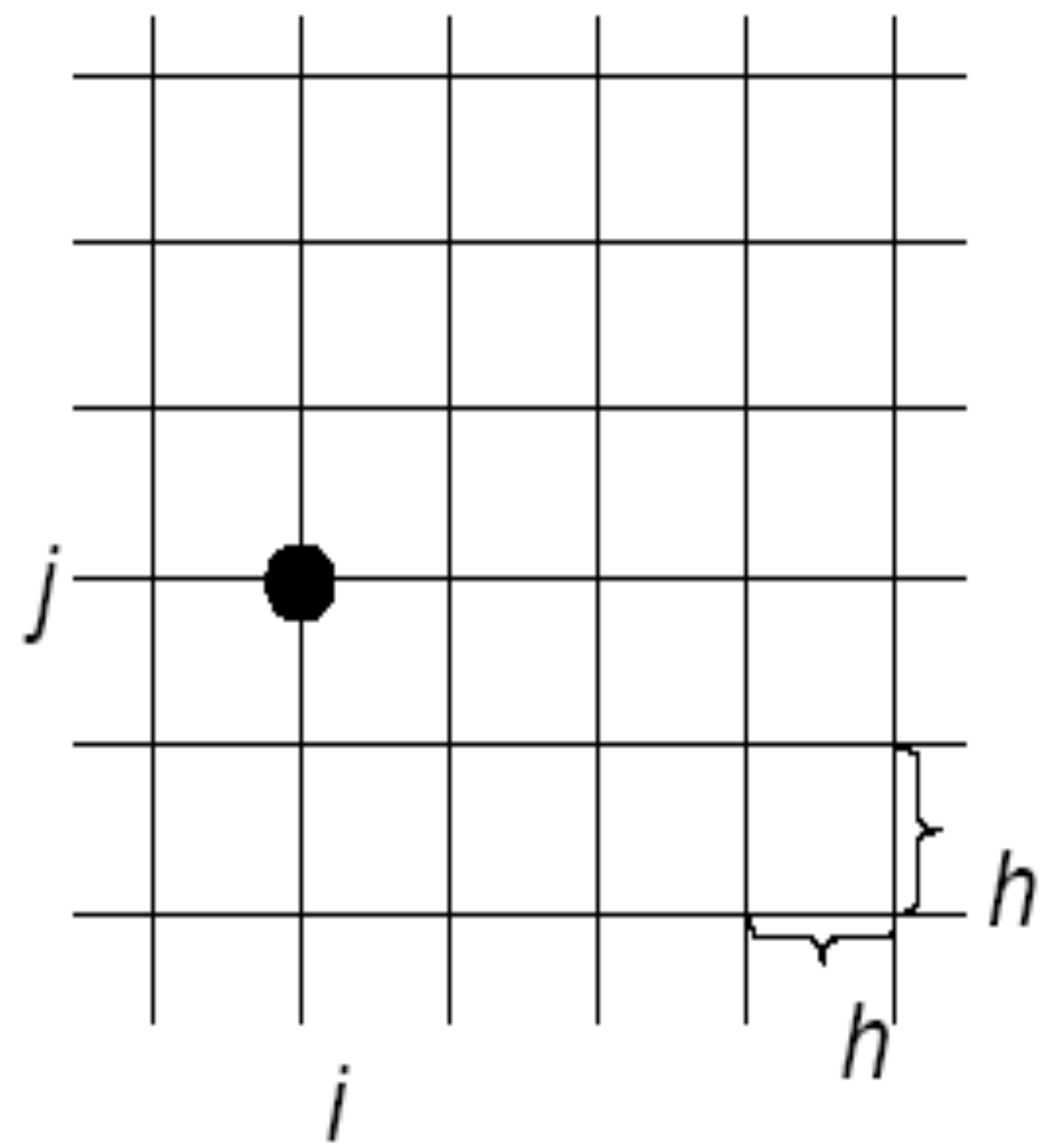
- 2D or 3D uniform or rectangular grids

  - Partial derivatives

$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$$

  - Same as in univariate case along each coordinate axis

  - Example : gradient in a 3D uniform grid



$$grad\ f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{f_{i+1,j,k} - f_{i-1,j,k}}{2h} \\ \frac{f_{i,j+1,k} - f_{i,j-1,k}}{2h} \\ \frac{f_{i,j,k+1} - f_{i,j,k-1}}{2h} \end{pmatrix}$$

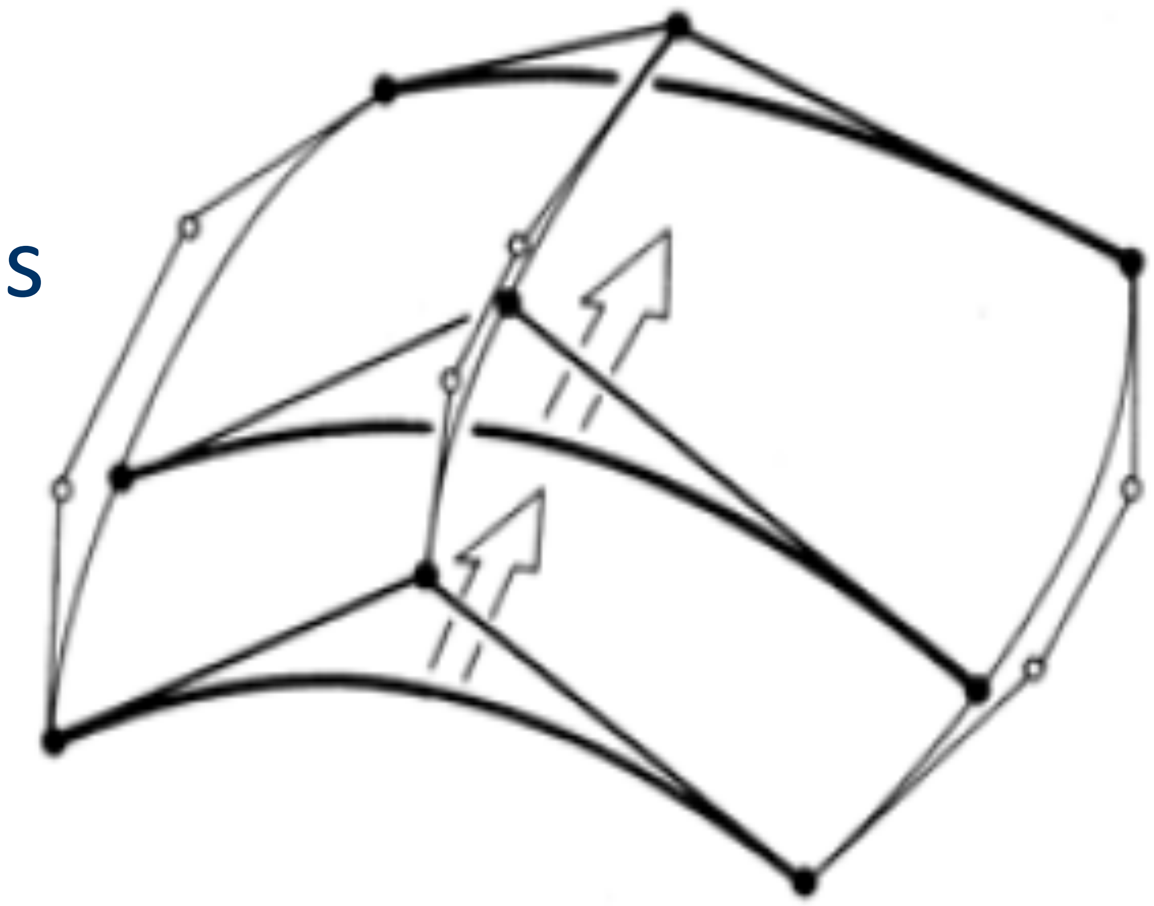
Interpolation on Grids

- Manifolds with more than 1D
- Tensor product
- Combination of several univariate interpolations
- Example for 2D surface:
  - N-m values f_{ji} with j=1..n and i=1..m
    given at points X*Y=($x_1, \ldots, x_n$)×（$y_1, \ldots, y_m$）
  - n univariate basis functions $\xi_j(x)$ on X
  - m univariate basis functions $\psi_i(y)$ on Y
  - n-m basis functions on X*Y:

$$\phi_{ij}(x,y) = \xi_i(x)\psi_j(y)$$

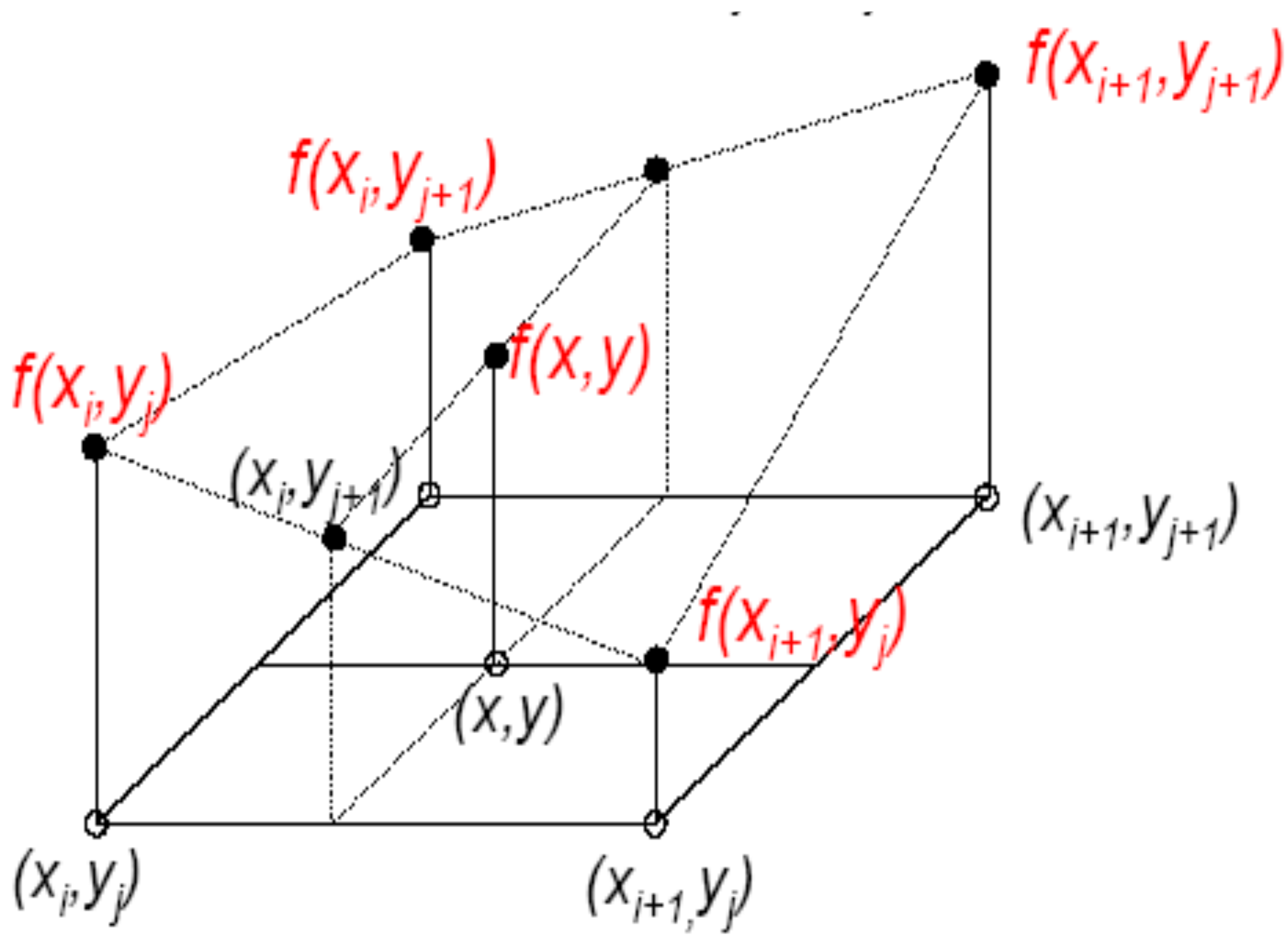
  - Tensor product:

$$f(x,y) = \sum_{i=1,j=1}^{n,m} \phi_{ij}(x,y)c_{ij}$$



Interpolation on Grids

- Bilinear interpolation on a rectangle
  - Tensor product for two linear interpolations
  - 2D local interpolation in a cell
  - Known solution of the linear system of equations for the coefficients c_{ij}
  - Four data points (x_i, y_j),…, (x_{i+1}, y_{j+1}) with scalar values $f_{i,j} = f\left(x_p, y_j\right)$, …
  - Bilinear interpolation of points $(x\,, y\,)$ with $x_i \leq x \leq x_{i+1}$ and$y_i \leq y \leq \; y_{i+1}$



Interpolation on Grids

- Bilinear interpolation on a rectangle

$$f(x,y) = (1-\beta)\left[(1-\alpha)f_{i,j} + \alpha f_{i+1,j}\right] + \beta\left[(1-\alpha)f_{i,j+1} + \alpha f_{i+1,j+1}\right]$$

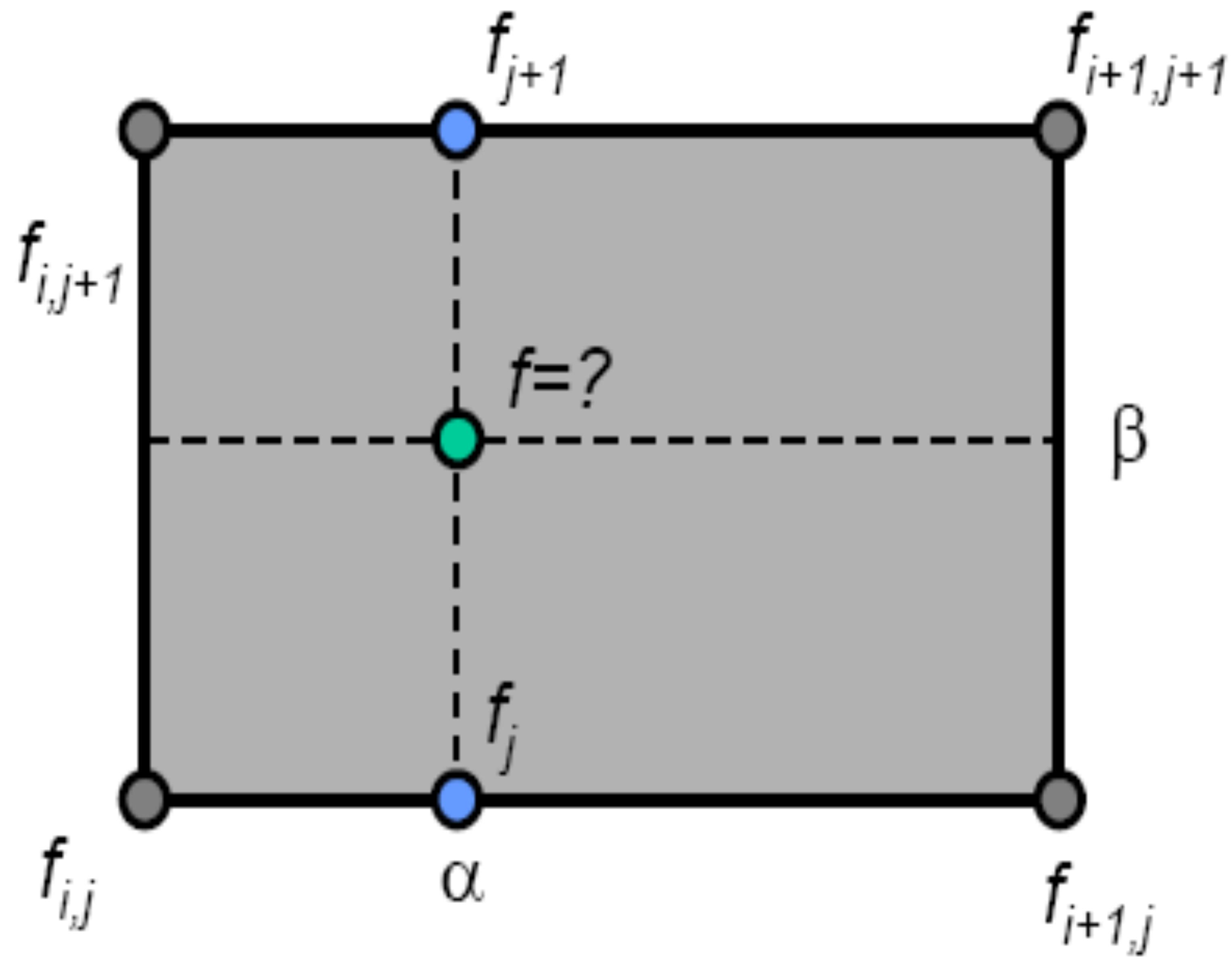
=(1- $-\beta)f_j + \beta f_{j+1}$

With $f_j = (1-\alpha)f_{i,j} + \alpha f_{i+1,j}$

$f_{j+1} = (1-\alpha)f_{i,j+1} + \alpha f_{i+1,j+1}$

And local coordinates

$$\alpha = \frac{x - x_i}{x_{i+1} - x_i} \quad \beta = \frac{y - y_i}{y_{i+1} - y_i} \qquad \alpha, \beta \epsilon [0,1]$$



Interpolation on Grids

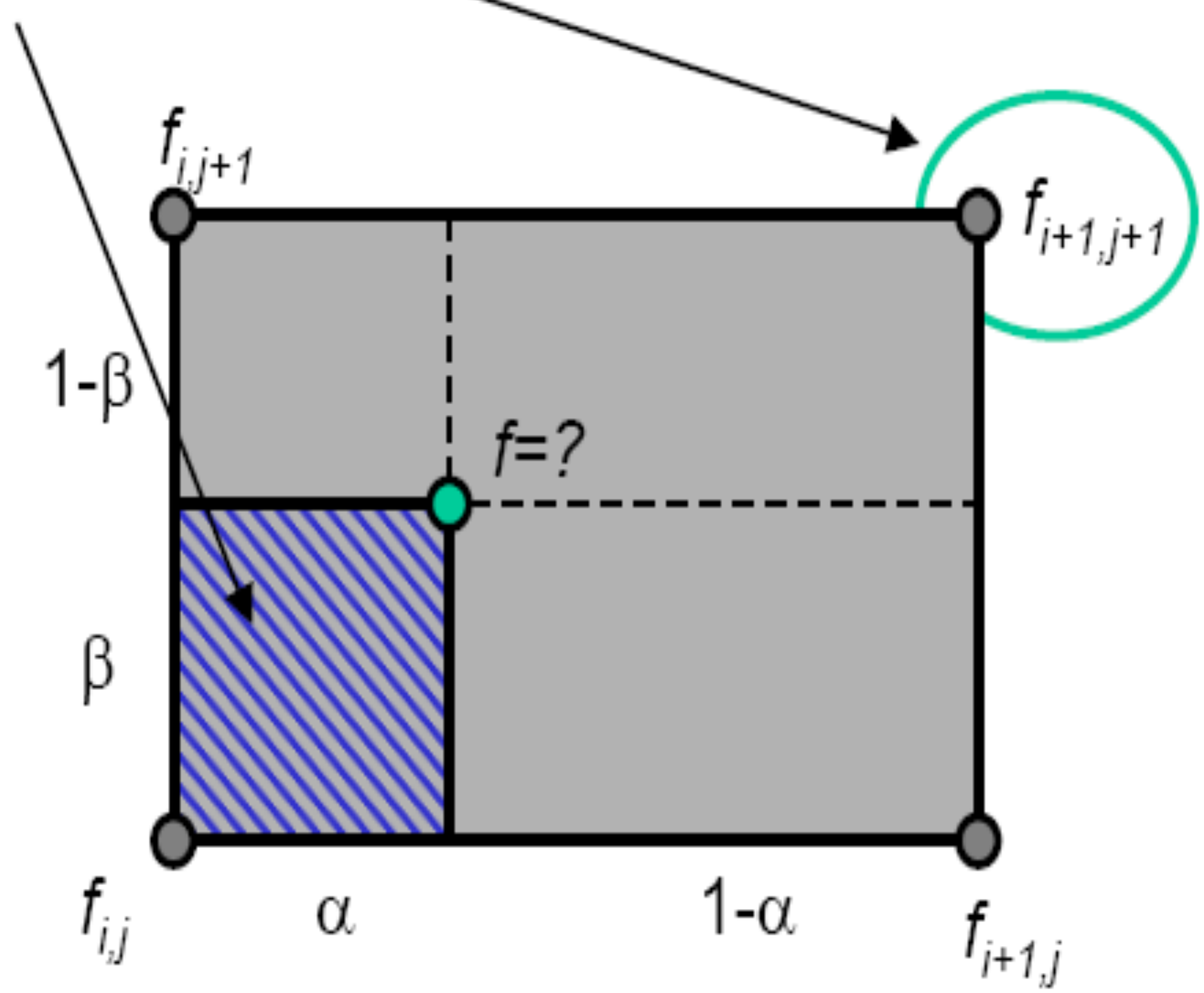
- Bilinear interpolation on a rectangle

$$f(x,y) = (1-\alpha)(1-\beta)f_{i,j} + \alpha(1-\beta)f_{i+1,j} + (1-\alpha)\beta f_{i,j+1} + \alpha\beta f_{i+1,j+1}$$

Weighted by local areas of the opposite point

Bilinear interpolation is not linear (but quadratic)!

Cannot be inverted easily!



Interpolation on Grids

- Trilinear interpolation on a 3D uniform grid
  - Straightforward extension of bilinear interpolation
  - Three local coordinates α,β,γ
  - Known solution of the linear system of equations for the coefficients c_{ij}
  - Efficient evalution :
    f(α,β,γ)=a+ α(b+ β(e+h γ))+ β(c+f γ)+ γ(d+g α)
    with coefficients a,b,c,d,e,f,g from data at the corner vertices
- Extension to higher order of continuity
  - Piecewise cubic interpolation in 1D
  - Piecewise bicubic interpolation in 2D
  - Piecewise tricubic interpolation in 3D
  - Based on Hermite polynomials

Interpolation on Grids

- Affine combination of points x(in Euclidean space):
  - Straightforward extension of bilinear interpolation

    $0 \leq \alpha_i \leq 1, \forall i$

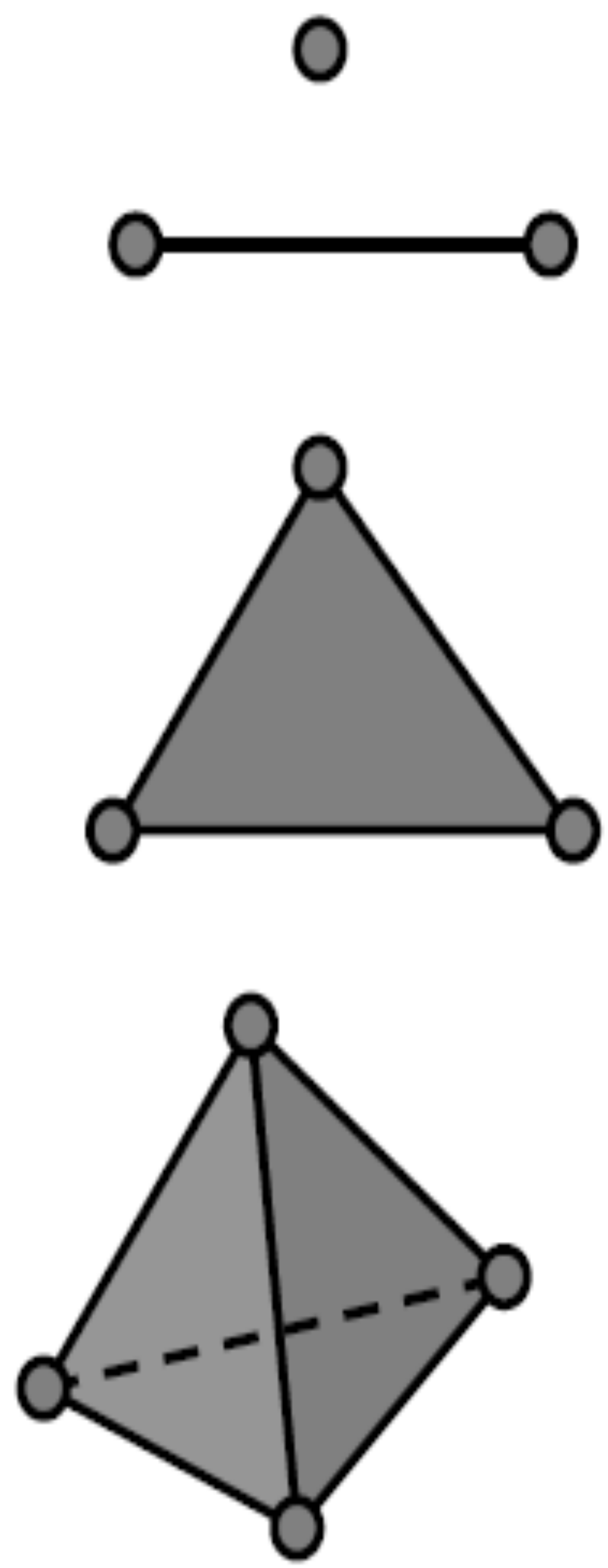
    $\sum_i \alpha_i = 1$

    α_i are barycentric coordinates

- Affinely independent set of points:
  - No point can be expressed as affine combination of the other points
  - Mamimum number of points is d+1 in R^d

Interpolation on Grids

- Simplex in R^d
  - d+1 affinely independent points
  - Span of these points
  - 0D :point
  - 1D:line
  - 2D:triangle
  - 3D: tetrahedron



Interpolation on Grids

- Barycentric interpolation on a simplex
  - d+1 points x_i with function values f_i
  - Point x within the simplex described as affine combination of x_i
  - Possible approach:
    solve for coefficients α_i based on x=$\sum_i \alpha_i \cdot x_i$ and $\sum_i \alpha_i = 1$
  - Function value at x: f=$\sum_i \alpha_i \cdot f_i$ is affine combination of f_i
- Barycentric coordinates from area / volume considerations:

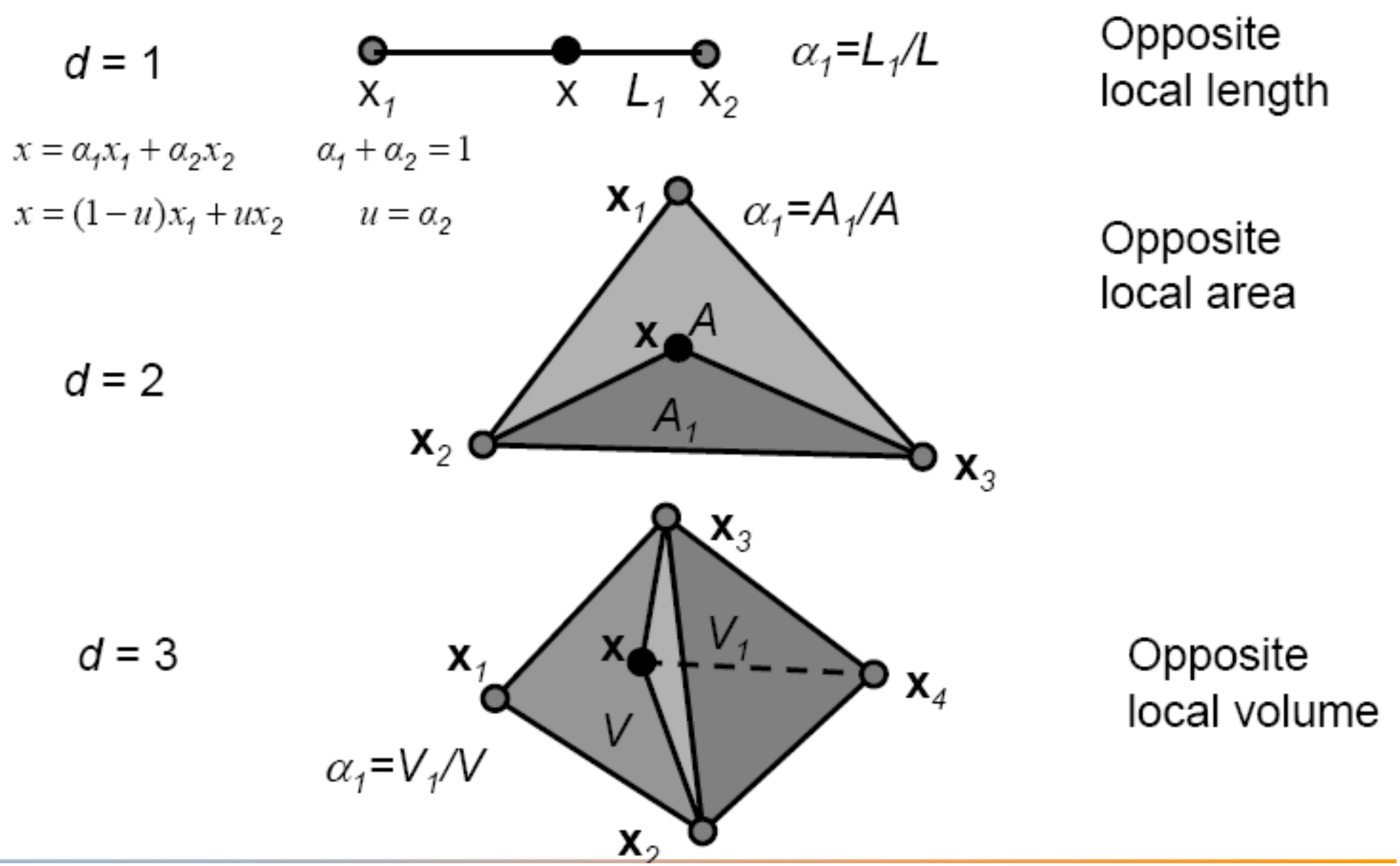
$$\alpha_i = \frac{\mathbf{Vol}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{d+1})}{\mathbf{Vol}(\mathbf{x}_1, \ldots, \mathbf{x}_{d+1})}$$

$$\mathbf{Vol}(\mathbf{x}_1, \ldots, \mathbf{x}_{d+1}) = \mathbf{det}\begin{pmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_{d+1} \\ 1 & \cdots & 1 \end{pmatrix}$$

generalized measure for area/volume

Interpolation on Grids

- Barycentric coordinates from area / volume considerations:

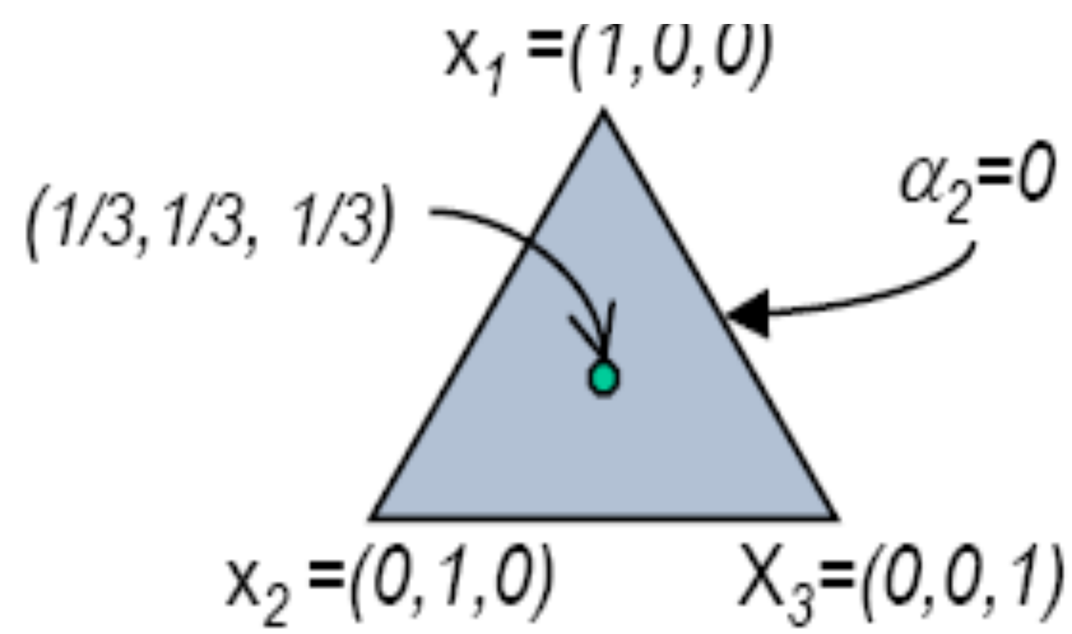


Interpolation on Grids

- Barycentric interpolation in a triangle
  - Geometrically , barycentric coordinates are given by the ratios of the area of the whole triangle and the subtriangles defined by x and any two points of $x_1, x_2\,, x_3$

$$\mathbf{Vol}(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) = \mathbf{det}\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} =$$

$$= \pm 2\,\mathrm{Area}(\Delta(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3))$$

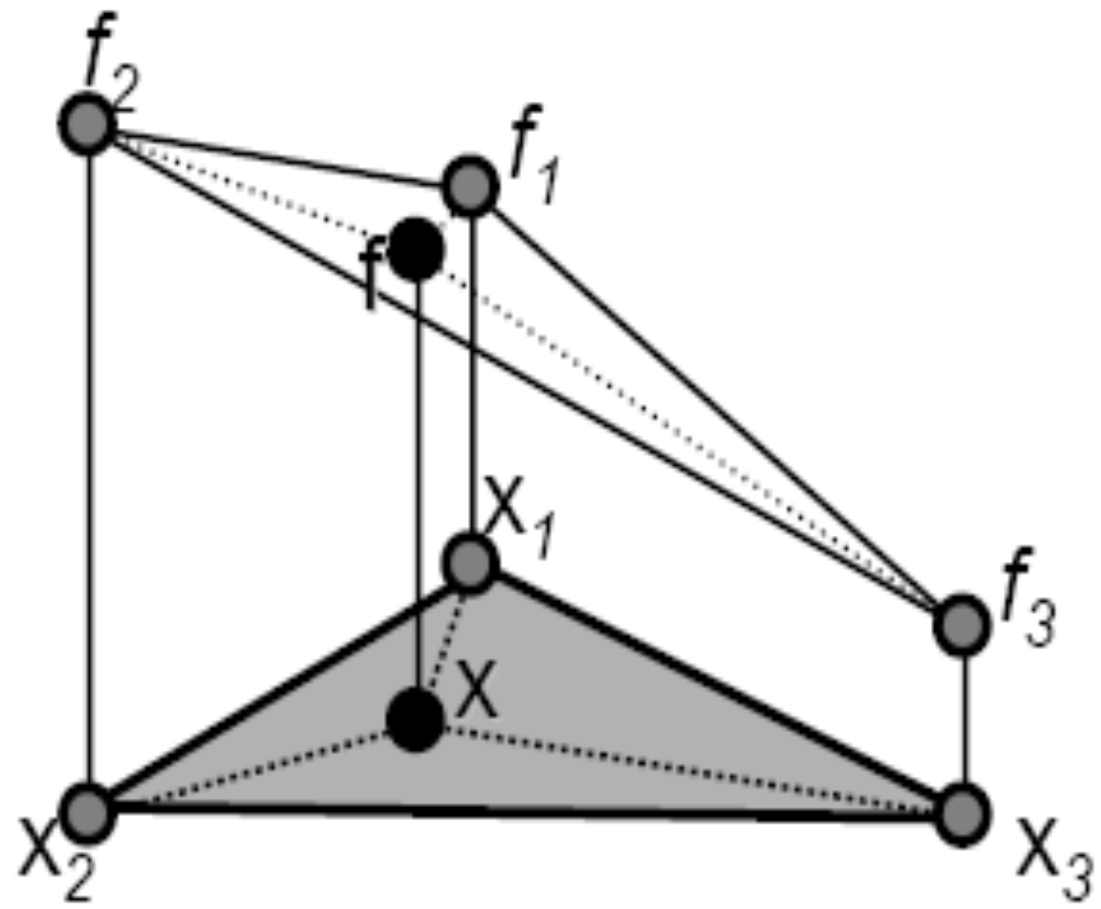


$$\alpha_1 = \frac{\mathbf{Vol}(\mathbf{x},\mathbf{x}_2,\mathbf{x}_3)}{\mathbf{Vol}(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)} \qquad \alpha_1 + \alpha_2 + \alpha_3 = 1$$

$$\mathbf{x} = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \alpha_3\mathbf{x}_3$$

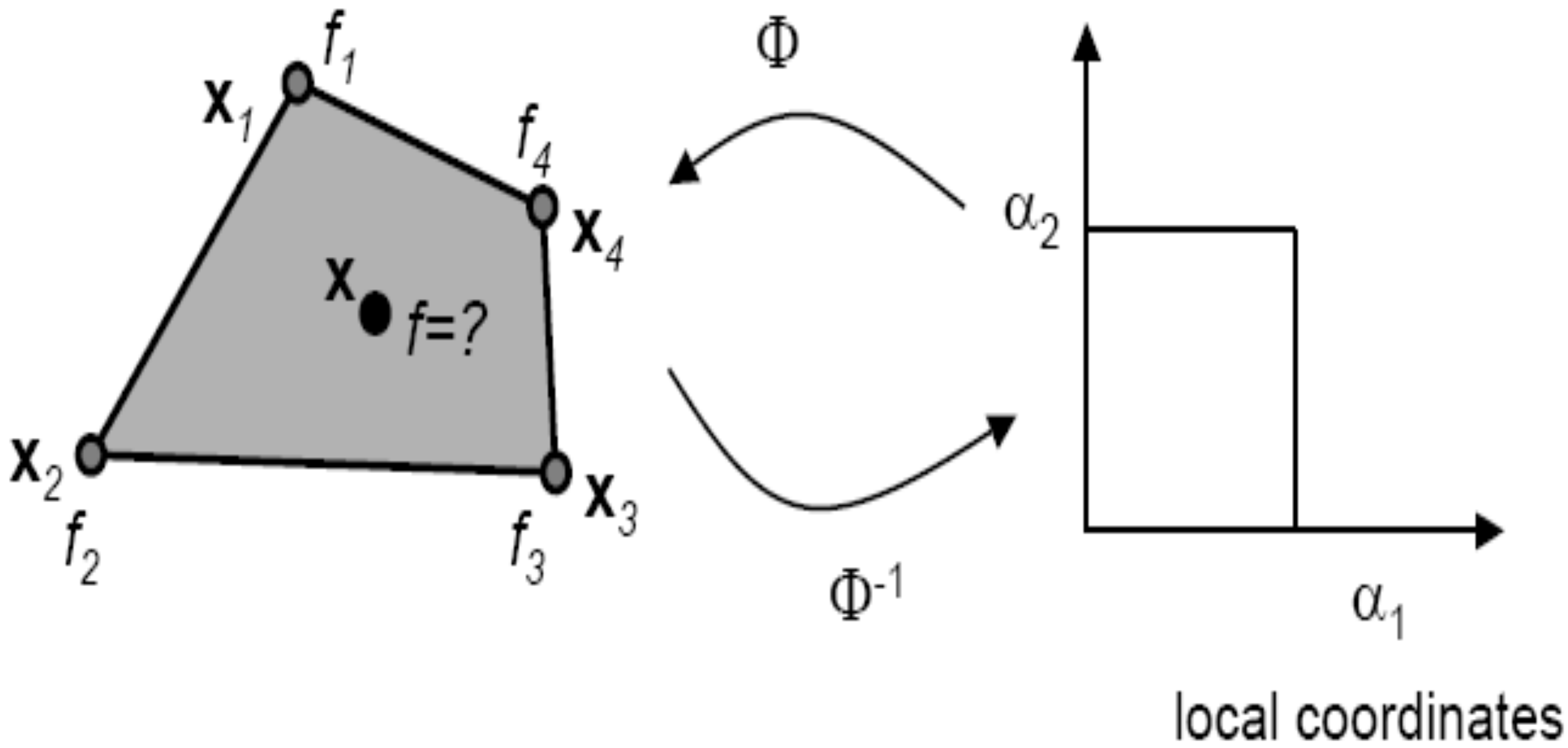
$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}; \quad \alpha_3 = 1 - \alpha_1 - \alpha_2$$

$$f(\mathbf{x}) = \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3$$



Interpolation on Grids

- Interpolation in a generic quadrilateral
  - Main application: curvilinear grids
  - Problem: find a parameterization for arbitrary quadrilaterals



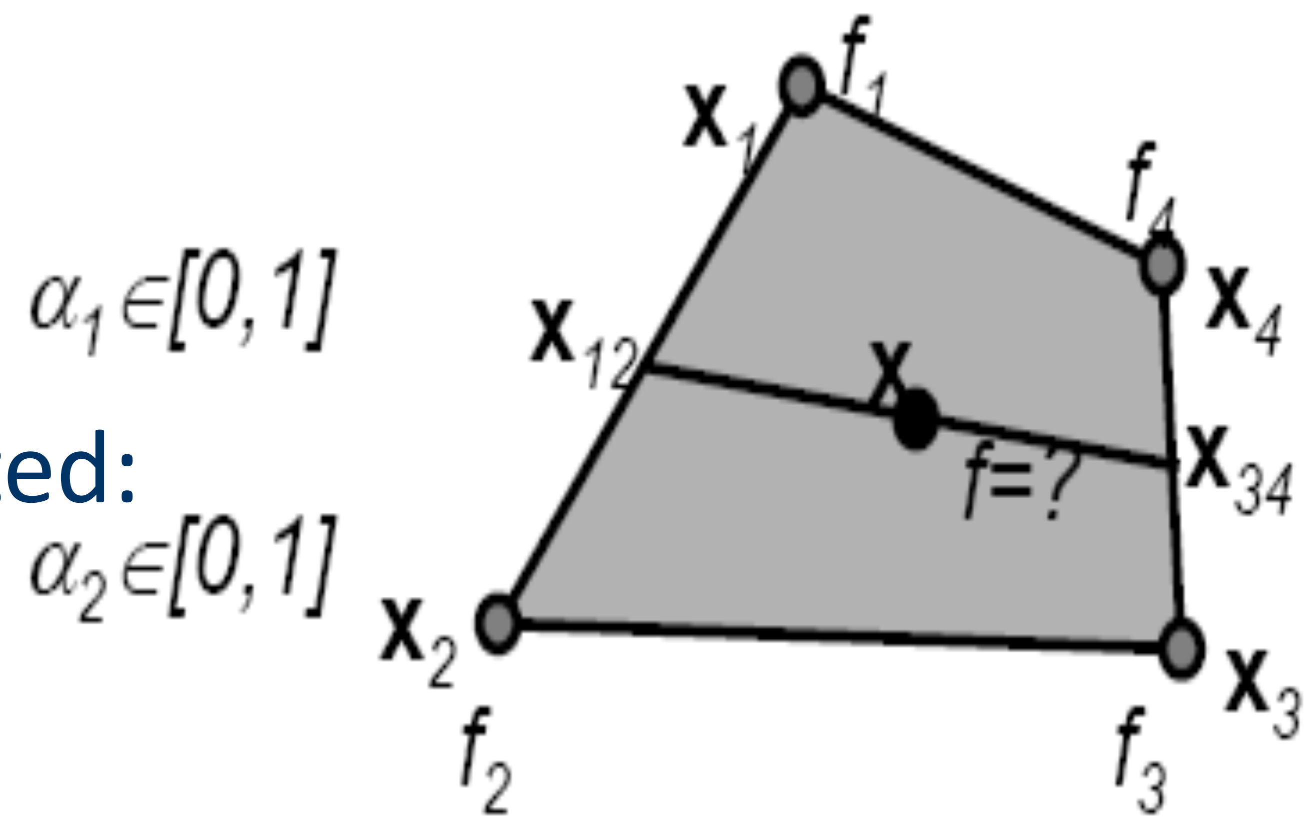
Interpolation on Grids

- Mapping ϕ from rectangular domain to quadratic domains is known: bilinear interpolation on a rectangle

$$\boldsymbol{X}_{12} = \alpha_1 \cdot \boldsymbol{X}_{\mathbf{1}} + (1 - \alpha_1) \cdot \boldsymbol{X}_{\mathbf{2}}$$

$$\boldsymbol{X}_{34} = \alpha_1 \cdot \boldsymbol{X}_{\mathbf{4}} + (1 - \alpha_1) \cdot \boldsymbol{X}_{\mathbf{3}}$$

$$\boldsymbol{X} = \alpha_2 \cdot \boldsymbol{X}_{\mathbf{12}} + (1 - \alpha_2) \cdot \boldsymbol{X}_{\mathbf{34}}$$



- Computing the inverse of ϕ is more complicated:
  - Analytically solve quadratic system for α_1, α_2
  - Or: numerical solution by Newton iteration

- Final value: $f = \alpha_2 \cdot (\alpha_1 \cdot f_1 + (1 - \alpha_1) \cdot f_2) + (1 - \alpha_2) \cdot (\alpha_1 \cdot f_4 + (1 - \alpha_1) \cdot f_3)$

Interpolation on Grids

- Jacobi matrix $J(\Phi)$
  - $J(\Phi)_{ij} = \partial\Phi_i / \partial\alpha_j$
  - $J(\Phi)_{.j}$ describes direction and speed of position changes of Φ when α_j are varied
- Newton iteration

Start with seed points as start configuration , e.g. , $\alpha_i = 1/2$

While ($\|\mathrm{x} - \Phi(\alpha_1, \alpha_2, \alpha_3)\| > \varepsilon$)

Compute $\mathrm{J}(\Phi(\alpha_1, \alpha_2, \alpha_3))$

transform $\mathbf{X}$ in coordinate system $\mathrm{J}(\Phi)$:

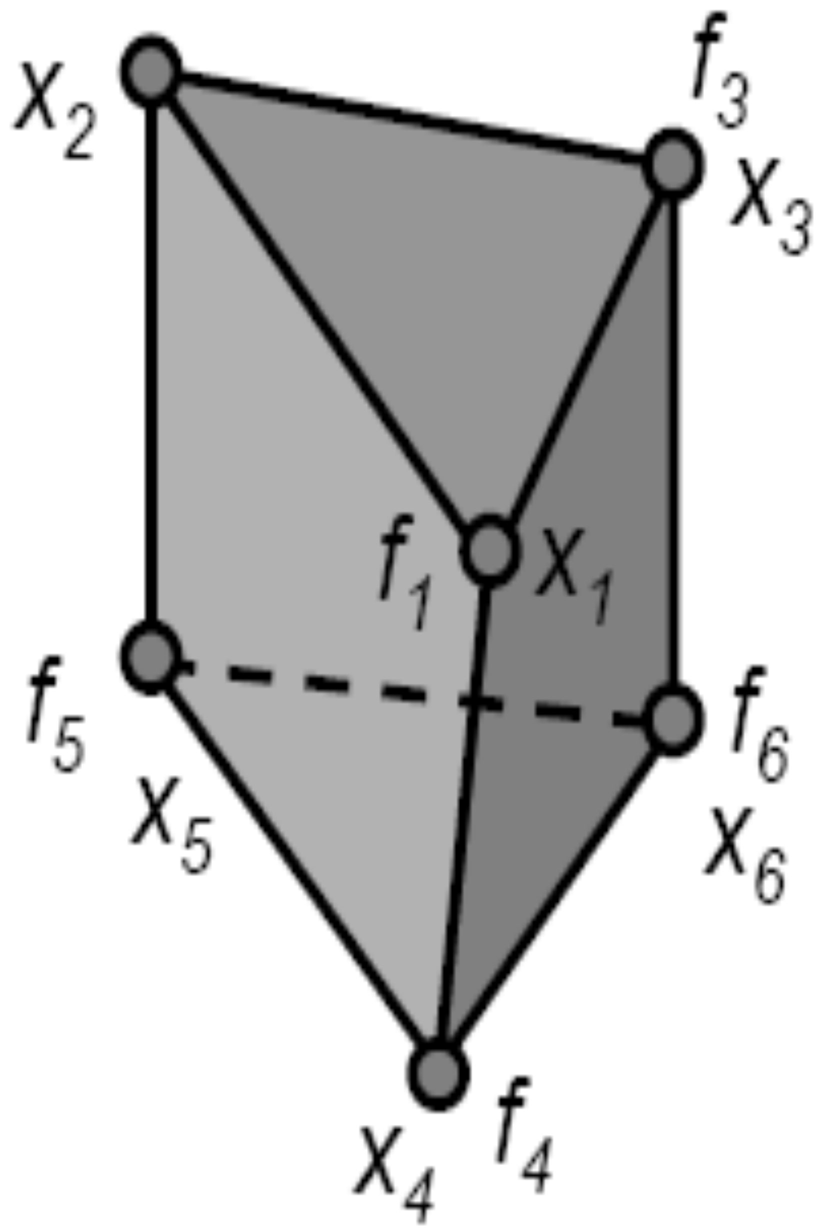
$$x_\alpha = \mathrm{J}\big(\Phi(\alpha_1, \alpha_2, \alpha_3)\big)^{-1} \cdot (x - \Phi(\alpha_1, \alpha_2, \alpha_3))$$

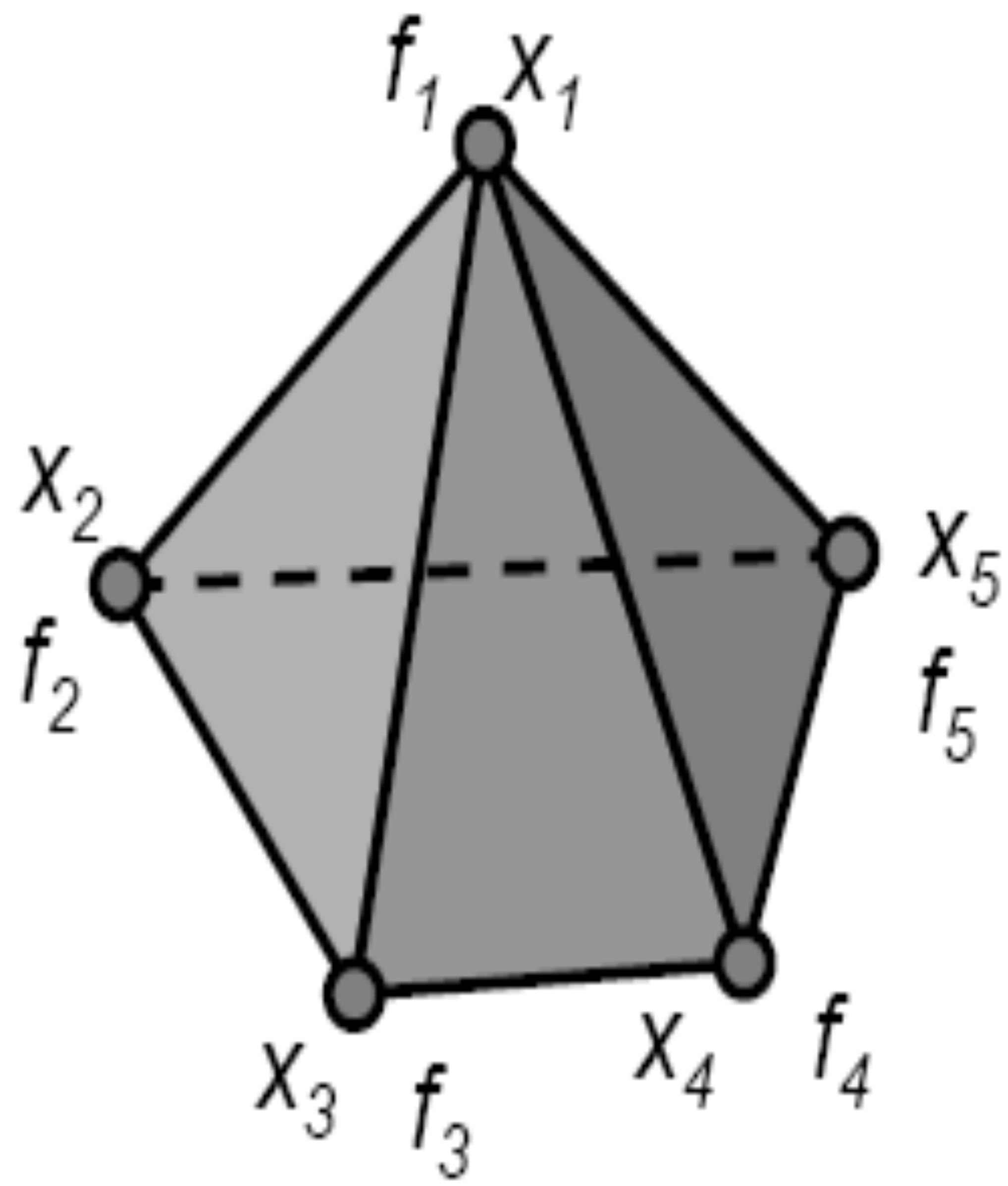
update $\alpha_i = \alpha_i + x_{\alpha,i}$

Maximum error ε

Interpolation on Grids

- Other primitive cell types possible





Prism:
- twice barycentric
- once linear

Pyramid:
- bilinear on base face
- then linear

Interpolation on Grids

- Radial basis functions (RBF)
  - n function values f_i given at n points $\boldsymbol{X}_i$
  - Interpolant $f(\boldsymbol{X}) = \sum_{i=1}^{n} \lambda_i \phi\left(\left|\left|\boldsymbol{X} - \boldsymbol{X}_i\right|\right|\right) + \sum_{m=0}^{k} c_m P_m(\boldsymbol{X})$
  - Univariate radial basis $\phi(r)$
  - Examples:
    - Polynomials r^v
    - Gaussians $\exp(r^{-2})$
  - Polynomial basis $\{p_m\}$ for (k+1)-dimensional vector space

Interpolation on Grids

- Radial basis functions (RBF)
  - Under-determined system:
    n equations for n+(k+1) unknowns
  - Additional constraints (orthogonality / side conditions):

$$\sum_{i=1}^{n} \lambda_i p_m(\boldsymbol{X}_i) = 0 \quad \forall\, m = 0 \ldots k$$

  - Well-defined system of linear equations (vector / matrix notation):

Coefficients for radial function

$$\mathbf{A}_{i,j} = \phi\left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)$$

Function values at sample points

$$\begin{pmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}$$

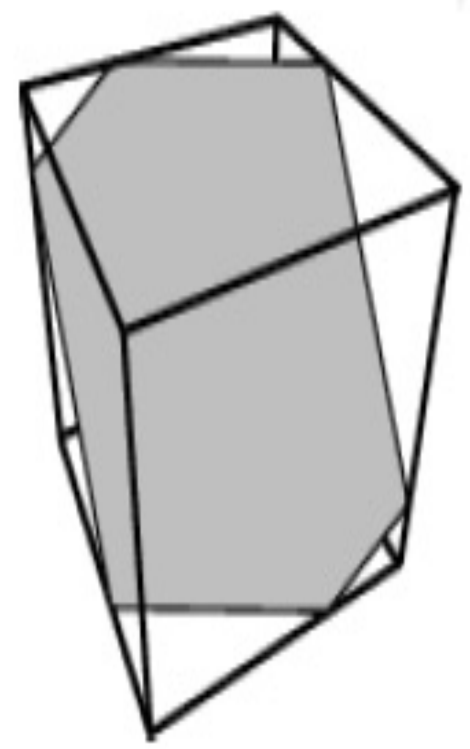
Polynomial basis

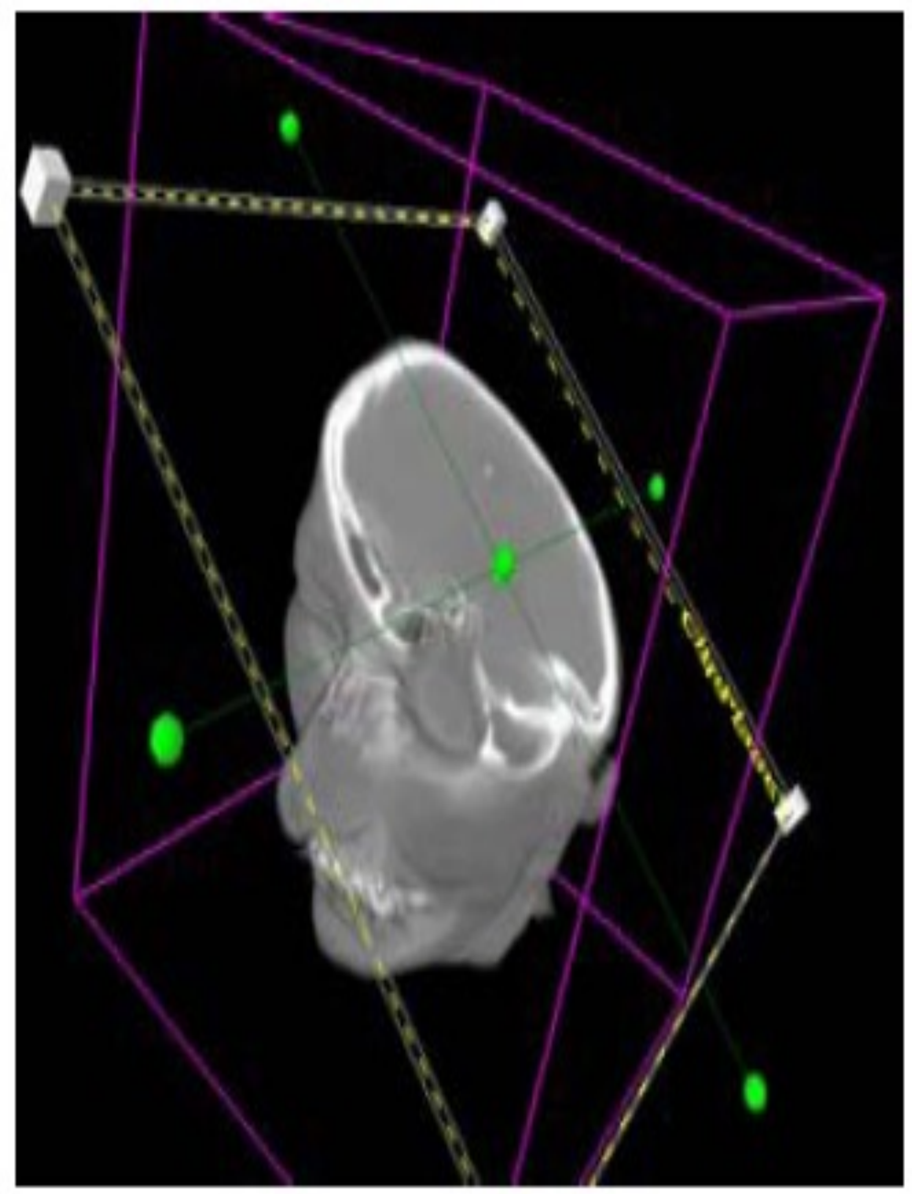
Coefficients for polynomials

- Interpolation
- Filtering

Filtering by Projection or Selection

- Remove the invisible or uninterested regions





Thanks