Data Representation

outline

Continuous Data

Sampled Data

Discrete Datasets

Cell Types

- vertex, line, triangle, quad, tetrahedron, hexahedron

Grid Types

- Uniform, rectlinear, structured, Unstructured

Attributes

- Scalar, Vector, Color, Tensor, Non-numerical

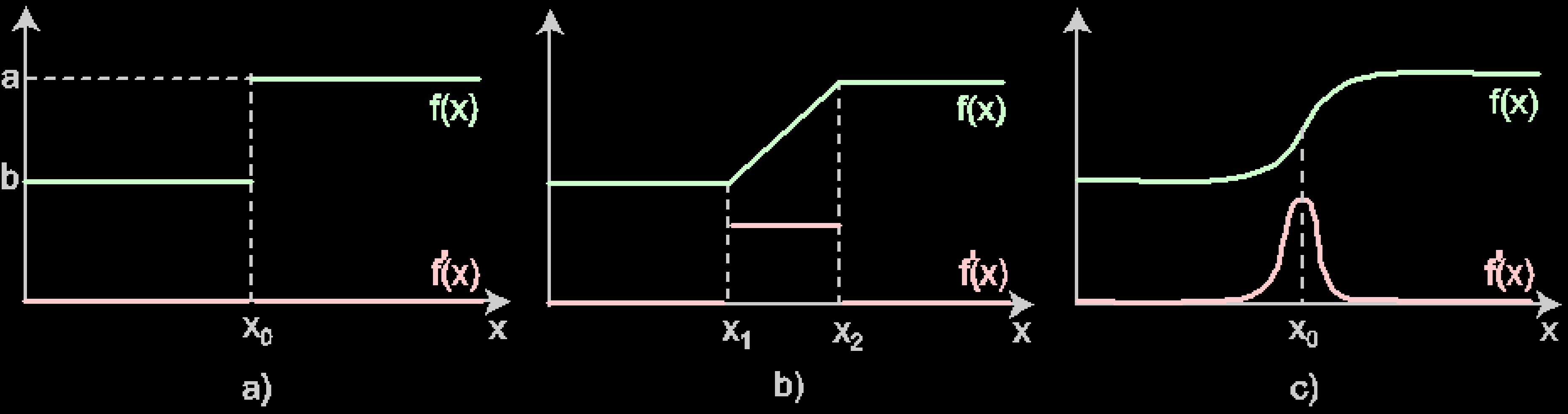
Continuous Data versus Discrete Data

- Continuous Data
    - Most scientific quantities are continuous in nature
    - Scientific Visualization, or *scivis*
- Discrete Data
    - E.g., text, images and others that can not be interpolated or scaled
    - Information Visualization, or *infovis*

- Continuous data, when represented by computers, are always in discrete form
    - These are called "sampled data"
        - Originated from continuous data
        - Intended to approximate the continuous quantity through visualization

Continuous Data

$$f(x), f'(x) = \frac{df(x)}{dx}, f''(x) = \frac{d^2 f(x)}{dx^2}$$



a)

b)

c)

a: discontinuous function

b: first-order continuous function: first-order derivative is not continuous

c: high-order continuous function

Continuous Data

Continuous data can be modeled as:

$$f : \mathrm{D} \rightarrow \mathrm{C}$$

$$\mathrm{D} \in \boldsymbol{R}^{d}$$

$$\mathrm{C} \in \boldsymbol{R}^{c}$$

$$(y_1, y_2 \ldots. y_c) = f(x_1, x_2 \ldots. x_d)$$

f is a d-dimension, c-valued function
D: function domain
C: function co-domain

Continuous Data

Cauchy criterion of continuity

$$\forall \varepsilon > 0, \exists \delta > 0$$

such that if

$$\| \mathrm{x} - \mathrm{p} \| < \delta$$

then

$$\| \mathrm{f(x)} - \mathrm{f(p)} \| < \varepsilon$$

In words, small changes in the input result in small changes in the output

Graphically, a function is continuous if the graph of the function is a connected surface without “holes” or “jumps”

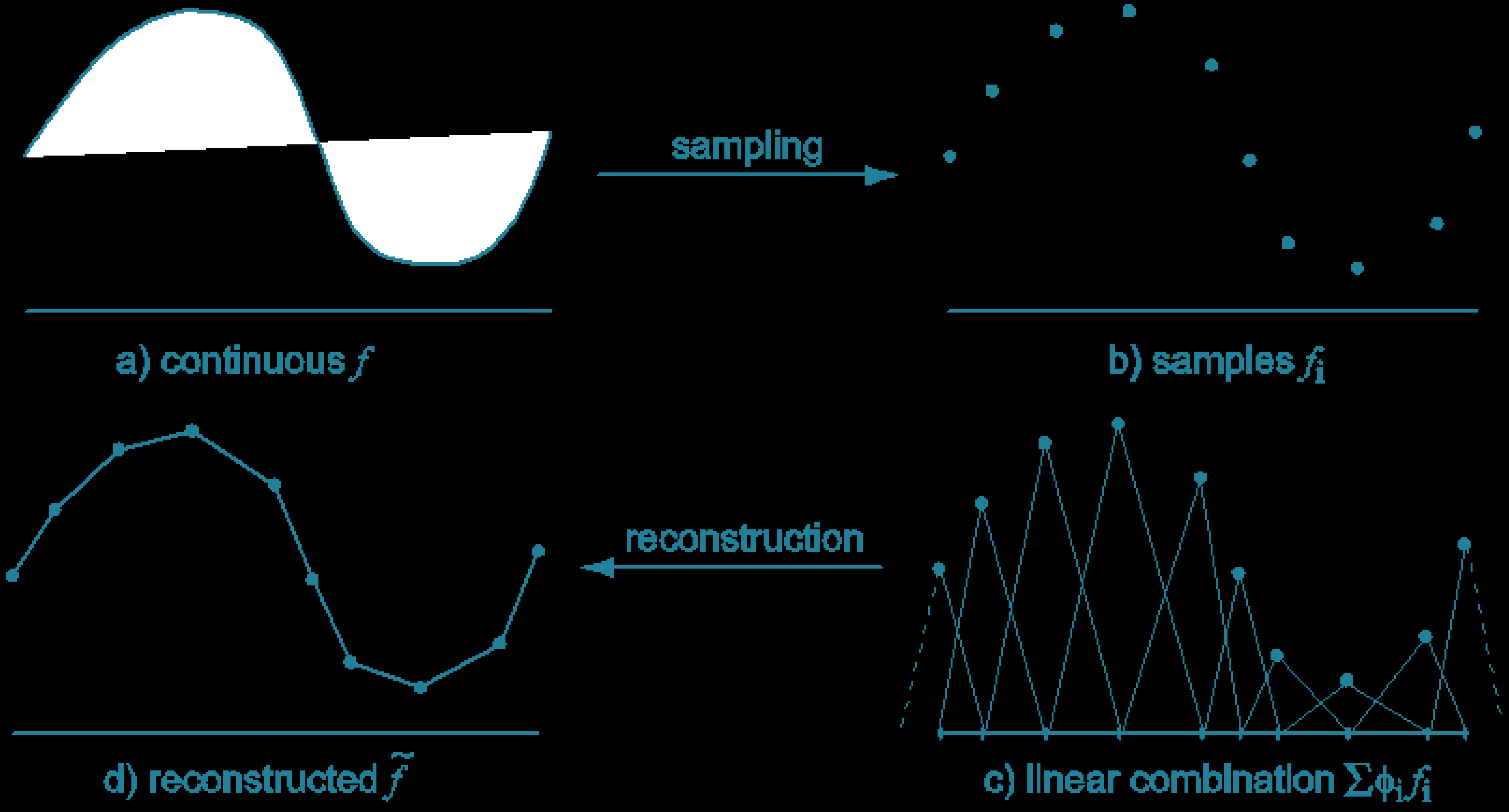
A function is continuous of order k if the function itself and all its derivative up to order k are also continuous

$$D_c = (\mathrm{D}, \mathrm{C}, \mathrm{f})$$

1. D: Function domain
2. C: Function co-domain
3. f: Function itself

- Geometric dimension: d
    - the space into which the function domain D is embedded
    - It is always 3 in the usual Euclidean space: d=3
- Topological dimension: s
    - The function domain D itself
    - A line or curve: s=1, d=3
    - A plane or curved surface: s=2, d=3
- Dataset dimension refers to the topological dimension
- Function values in the co-domain are called dataset attributes
- Attribute dimension: dimension of the function co-domain

Sampling and Reconstruction

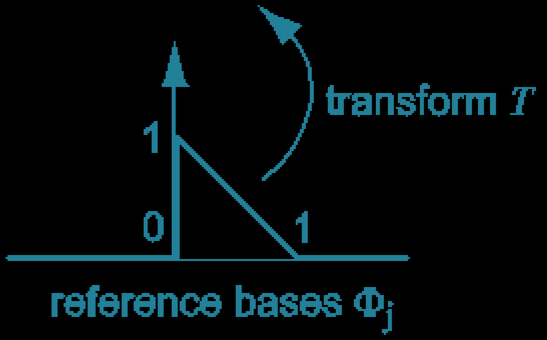


a) continuous f

b) samples f_i

d) reconstructed $\tilde{f}$

c) linear combination $\sum \phi_i f_i$



reference bases Φ_j

$$f \rightarrow f_i \rightarrow \tilde{f}$$

Sampled data

- Sampling: from continuous dataset
  to Sampled data
- Reconstruction:
  from Sampled data
  to recover/approximate
  continuous dataset

Sampled Dataset

Sampled dataset

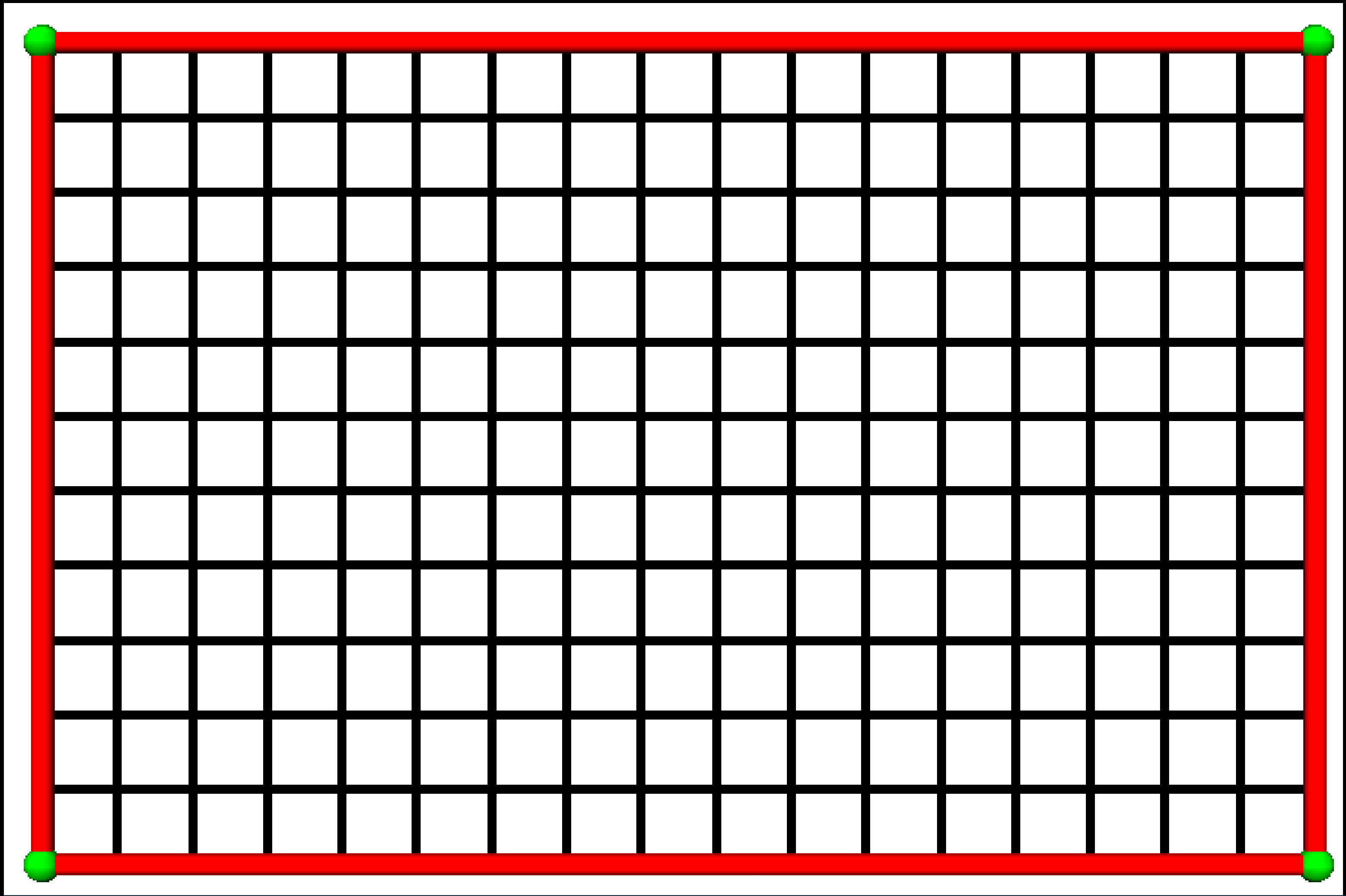
$$\boldsymbol{D_s} = (\{p_i\}, \{C_i\}, \{f_i\}, \{\Phi_i^k\})$$

1. p: sampling points
2. c: cells
3. f: sampled values
4. Φ: basis function or interpolation function

Continuous dataset

$$\boldsymbol{D} = (\mathrm{D}, \mathrm{C}, \mathrm{f})$$

Sampling



Point,
Cell,
Grid

A signal domain is sampled in a grid that contains a set of cells defined by the sample points

Sampling

Point $p_i \in \boldsymbol{R}^d\,,$

Cell $ci = \{p_1, p_2 \ldots .p_d\}$

$c_i \cap c_j = 0, \forall i \neq j$

Grid $U_i c_i = \mathrm{D}$

Reconstruction

$$\{f_i\}, i \in \{1,2\ldots\}$$

$$\tilde{f} = \sum_{i=1}^{N} f_i \phi_i$$

where ϕ_i is called basis function

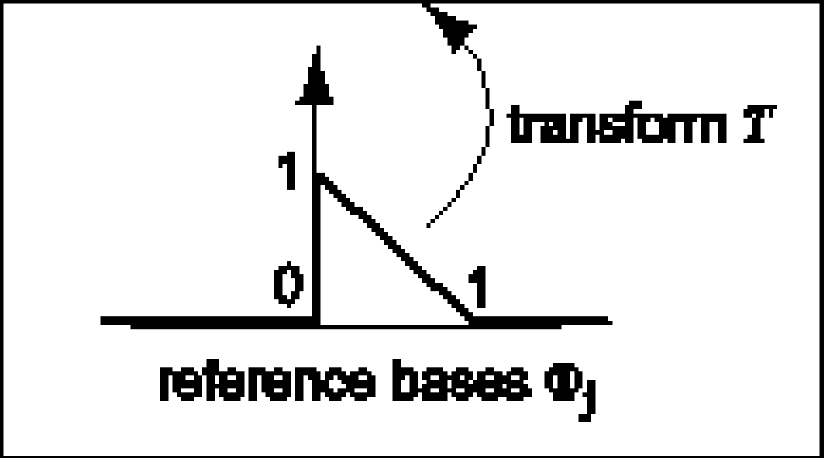
or interpolation function

Piecewise fitting: one cell one time

Reconstruction

Linear basis function

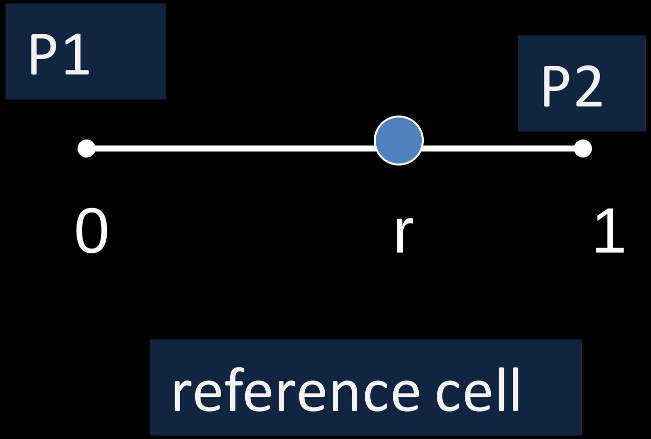
For 1-D line



$$\Phi_1^1(r) = 1 - r,$$

$$\Phi_2^1(r) = r,$$

$$f\ (r) = f_1(1-r) + f_2 r$$


P1

P2

0

r

1

reference cell

Basis Function

Basis function shall be orthonormal

1.Orthogonal: only vertex points within the same cell have contribution to the interpolated value
2.Normal: the sum of the basic functions of the vertices shall be unity.

Basis Function

Linear basis function

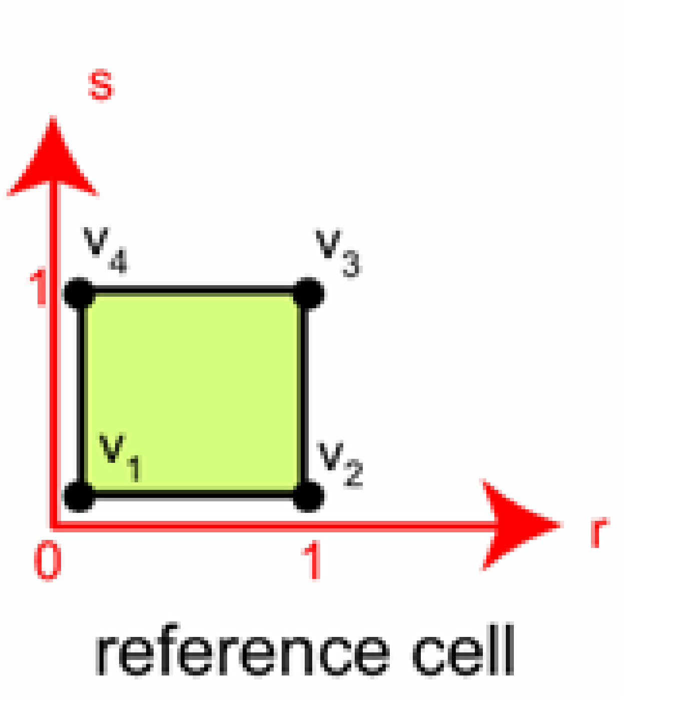
For 2-D quad

$$\Phi_1^1(r,s) = (1-r)(1-s)$$

$$\Phi_2^1(r,s) = r(1-s)$$

$$\Phi_3^1(r,s) = rs$$

$$\Phi_4^1(r,s) = (1-r)s$$



Sampled Dataset

Sampled dataset

$$\boldsymbol{D_s} = (\{p_i\}, \{C_i\}, \{f_i\}, \{\Phi_i^k\})$$

p: sampling points

c: cells

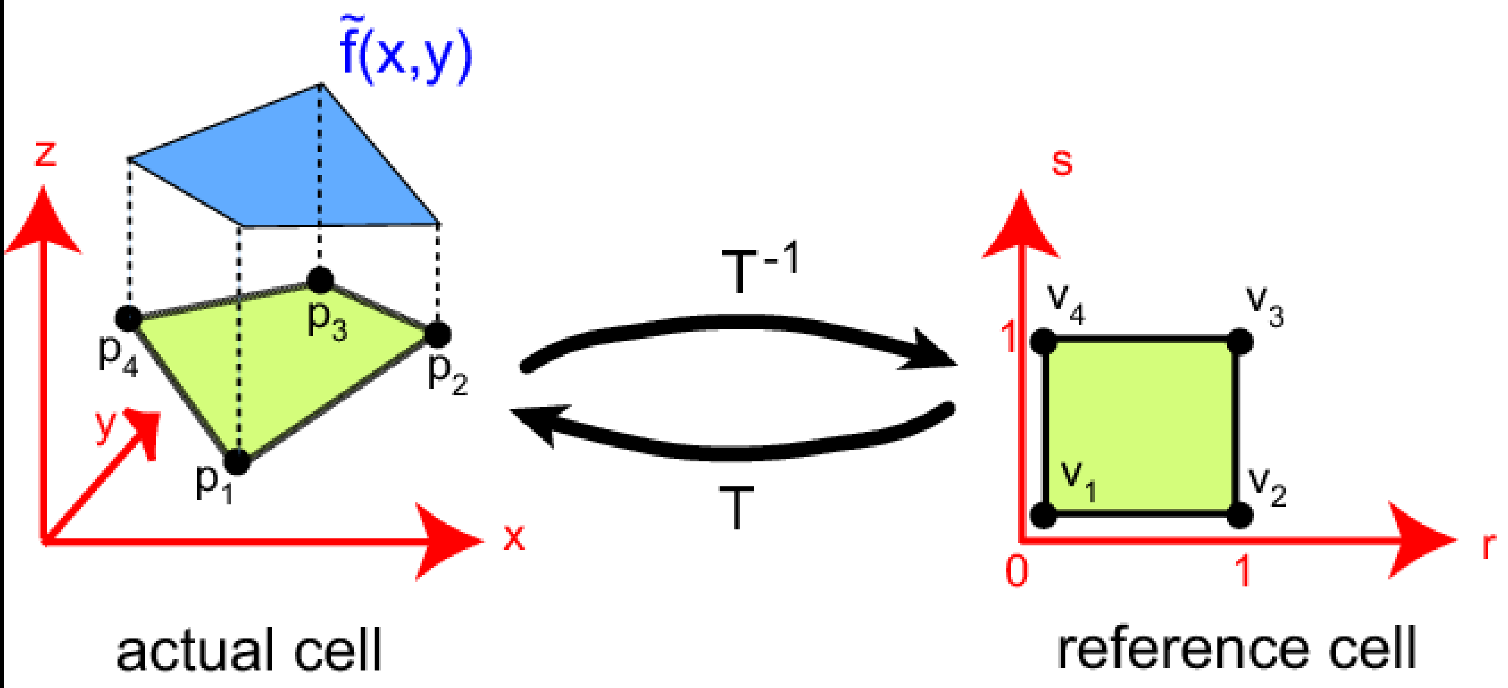
f: sampled values

Φ: basis function or interpolation function

Continuous dataset

$$\boldsymbol{D} = (\mathrm{D}, \mathrm{C}, \mathrm{f})$$

Basis Function



cell=(p1,p2,p3,p4)
D: (x,y,z)

cell=(v1,v2,v3,v4)
D: (r,s,t) and t=0

Coordinate Transformation

- Basis function is defined in reference cell
- Reference cell: axis-aligned unit cell, e.g., unit square in 2-D, unit line in 1-D
- Data are sampled at actual (world) cells
- Mapping between actual cell and reference cell

$$(x, y, z) = T(r, s, t)$$

$$(r, s, t) = T^{-1}(x, y, z)$$

$$\phi(x, y, z) \Leftrightarrow \Phi(r, s, t) = \Phi(T^{-1}(x, y, z))$$

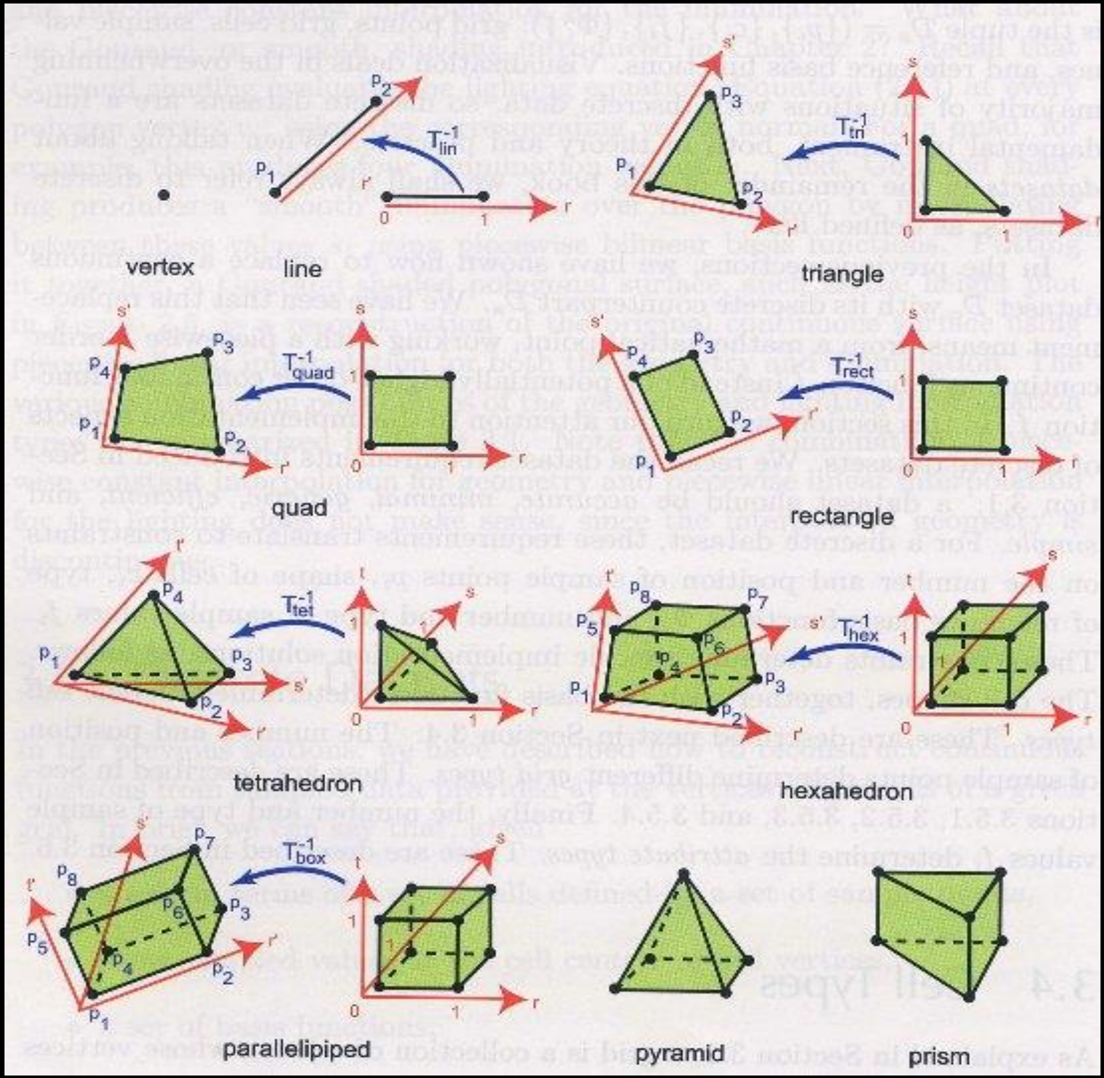
$$\tilde{f}(x, y, z) = \sum_{i=1}^{N} f_i \Phi_i^1(T^{-1}(x, y, z))$$

Discrete Datasets

- A Grid = cells + sample points
- Sample Values at cell centers/vertices
- Basis functions

Cell types

- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism



Vertex

d=0

$$c = \{v_1\}$$

$$\Phi_1^0(r,s) = 1$$

- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

Line

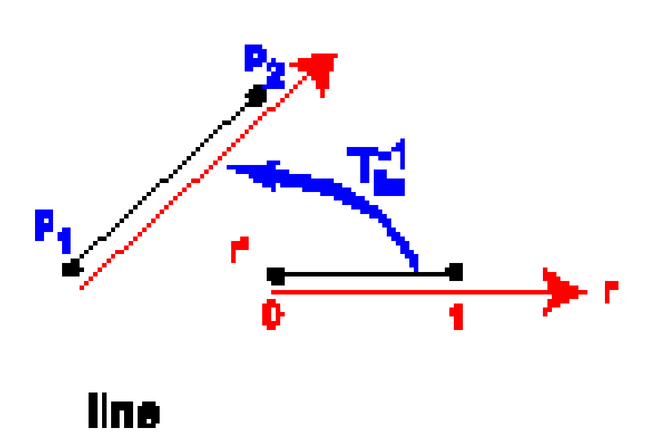
d=1

$$c = \{v_1, v_2\}$$

$$\Phi_1^1(r) = 1 - r$$

$$\Phi_2^1(r) = r$$

$$T^{-1}(x, y, z) = \frac{(\vec{p} - \vec{p}_1) \cdot (\vec{p}_2 - \vec{p}_1)}{\| \vec{p}_2 - \vec{p}_1 \|^2}$$



- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

Line (cont.)

Actual line d=1

$$\vec{p}_1 = x_1, \vec{p}_2 = x_2, \vec{p} = x$$

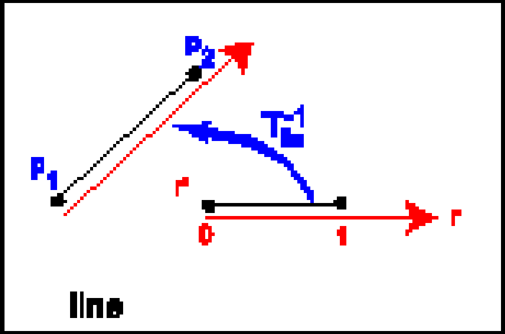
$$r = \frac{x - x_1}{x_2 - x_1}$$

$$f = f_1 \frac{x_2 - x}{x_2 - x_1} + f_2 \frac{x - x_1}{x_2 - x_1}$$

Actual line d=2

$$\vec{p}_1 = (x_1, y_1), \vec{p}_2 = (x_2, y_2), \vec{p} = (x, y)$$

$$r = \frac{(x - x_1)(x_2 - x_1\ ) + (y - y_1)(y_2 - y_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

Triangle

d=2

$$c = \{v_1, v_2, v_3\}$$

$$\Phi_1^1(r,s) = 1 - r - s$$

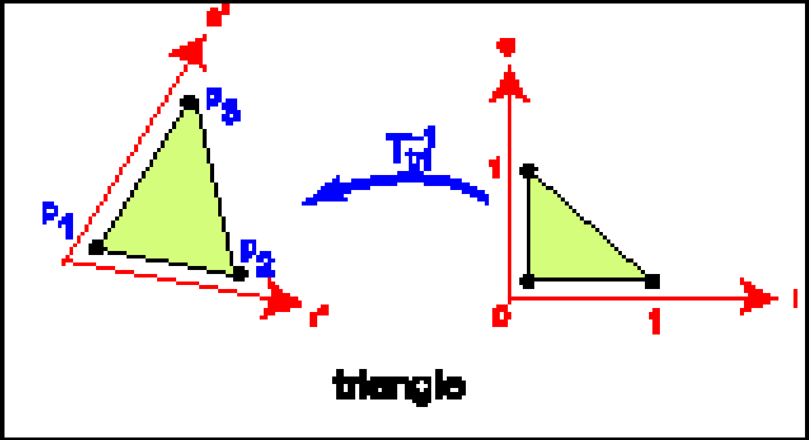
$$\Phi_2^1(r,s) = r$$

$$\Phi_3^1(r,s) = s$$

$$T^{-1}(x,y,z) = (r,s)$$

$$r = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_2 - \vec{p}_1)}{\| \vec{p}_2 - \vec{p}_1 \|^2}$$

$$s = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_3 - \vec{p}_1)}{\| \vec{p}_3 - \vec{p}_1 \|^2}$$



- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

Quad

d=2

$$c = \{v_1, v_2, v_3, v_4\}$$

$$\Phi_1^1(r,s) = (1-r)(1-s)$$

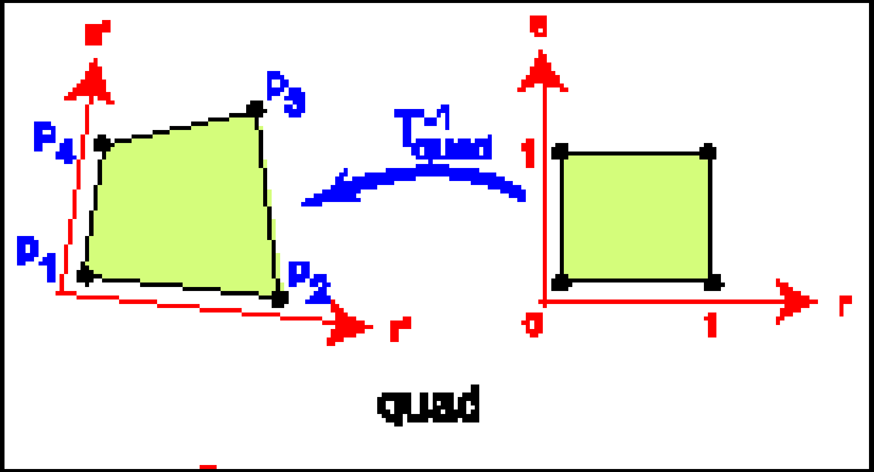
$$\Phi_2^1(r,s) = r(1-s)$$

$$\Phi_3^1(r,s) = rs$$

$$\Phi_4^1(r,s) = (1-r)s$$

$$r = \frac{(\vec{p}-\vec{p}_1)\cdot(\vec{p}_2-\vec{p}_1)}{\|\vec{p}_2-\vec{p}_1\|^2}$$

$$s = \frac{(\vec{p}-\vec{p}_1)\cdot(\vec{p}_4-\vec{p}_1)}{\|\vec{p}_4-\vec{p}_1\|^2}$$



- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

Tetrahedron

d=3

$$c = \{v_1, v_2, v_3, v_4\}$$

$$\Phi_1^1(r,s) = 1 - r - s - t$$

$$\Phi_2^1(r,s) = r$$

$$\Phi_3^1(r,s) = s$$

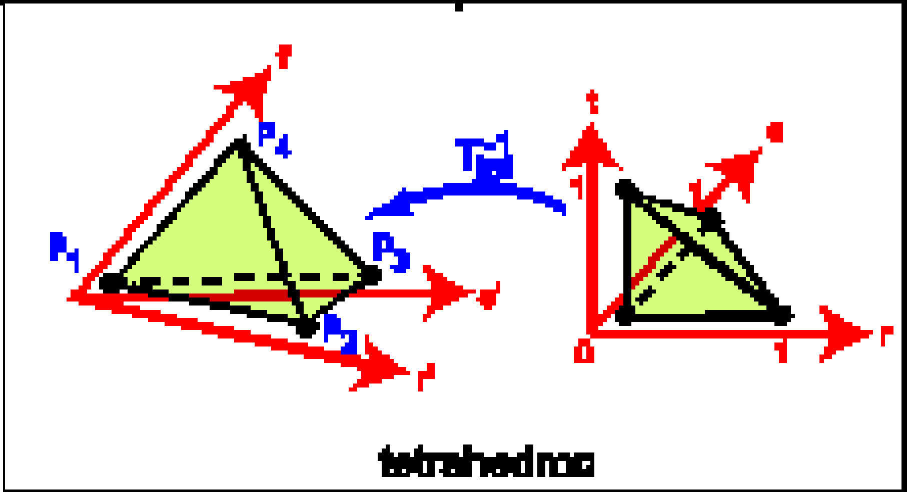
$$\Phi_4^1(r,s) = t$$

$$T^{-1}(x,y,z) = (r,s,t)$$

$$r = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_2 - \vec{p}_1)}{\|\vec{p}_2 - \vec{p}_1\|^2}$$

$$s = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_3 - \vec{p}_1)}{\|\vec{p}_3 - \vec{p}_1\|^2}$$

$$t = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_4 - \vec{p}_1)}{\|\vec{p}_4 - \vec{p}_1\|^2}$$



- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

Hexahedron

d=3

$$c = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$$

$$\Phi_1^1(r,s) = (1-r)(1-s)(1-t)$$

$$\Phi_2^1(r,s) = r(1-s)(1-t)$$

$$\Phi_3^1(r,s) = rs(1-t)$$

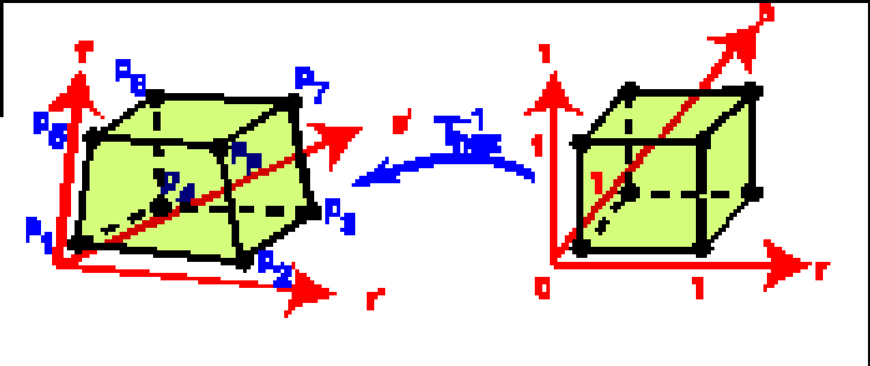
$$\Phi_4^1(r,s) = (1-r)(1-t)$$

$$\Phi_5^1(r,s) = (1-r)(1-s)t$$

$$\Phi_6^1(r,s) = r(1-s)t$$

$$\Phi_7^1(r,s) = rst$$

$$\Phi_8^1(r,s) = (1-r)st$$



hexahedron

- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

Hexahedron (cont.)

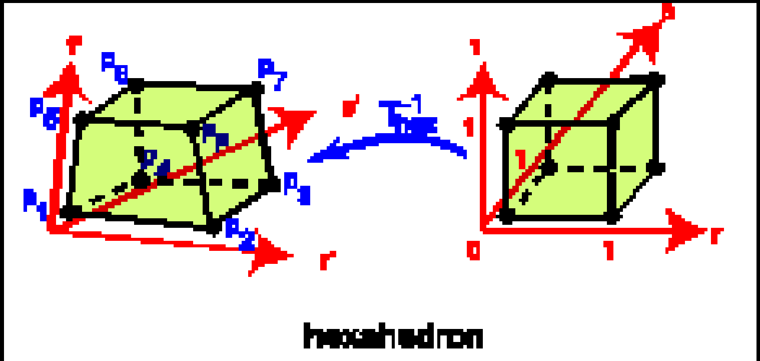
d=3

$$T^{-1}(x, y, z) = (r, s, t)$$

$$r = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_2 - \vec{p}_1)}{\| \vec{p}_2 - \vec{p}_1 \|^2}$$

$$s = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_4 - \vec{p}_1)}{\| \vec{p}_4 - \vec{p}_1 \|^2}$$

$$t = \frac{(\vec{p} - \vec{p}_1)\cdot(\vec{p}_8 - \vec{p}_1)}{\| \vec{p}_8 - \vec{p}_1 \|^2}$$

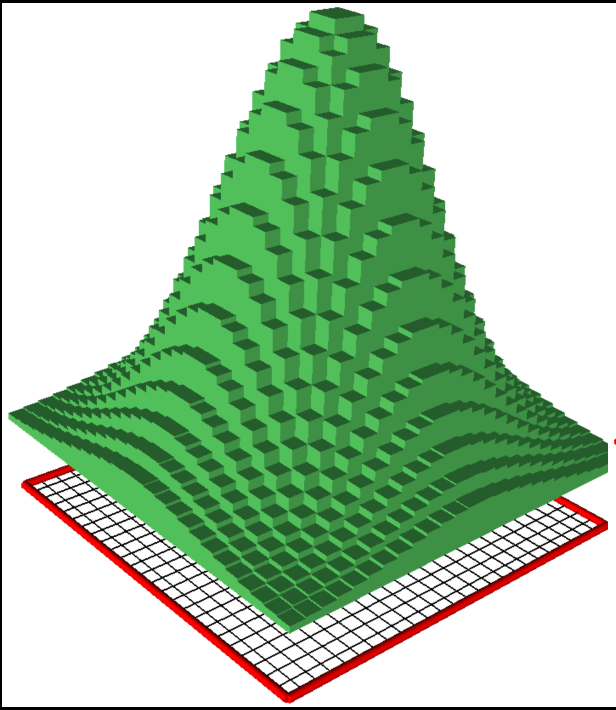


- Vertex
- Line
- Triangle
- Quad
- Rectangle
- Tetrahedron
- Hexahedron
- Parallelipiped
- Pyramid
- prism

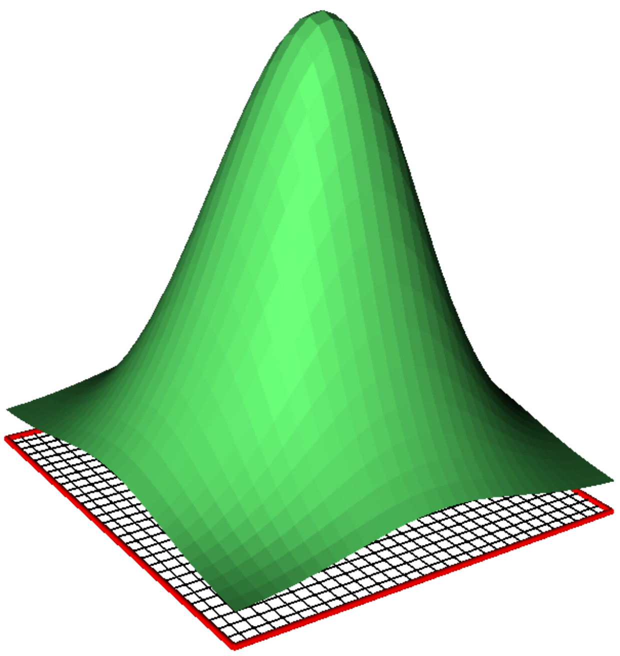
Effect of Reconstruction

	Geometry: Constant	Geometry: Linear
Lighting: Constant	Staircase shading	Flat Shading
Lighting: Linear	---------	Smooth (Gouraud) shading

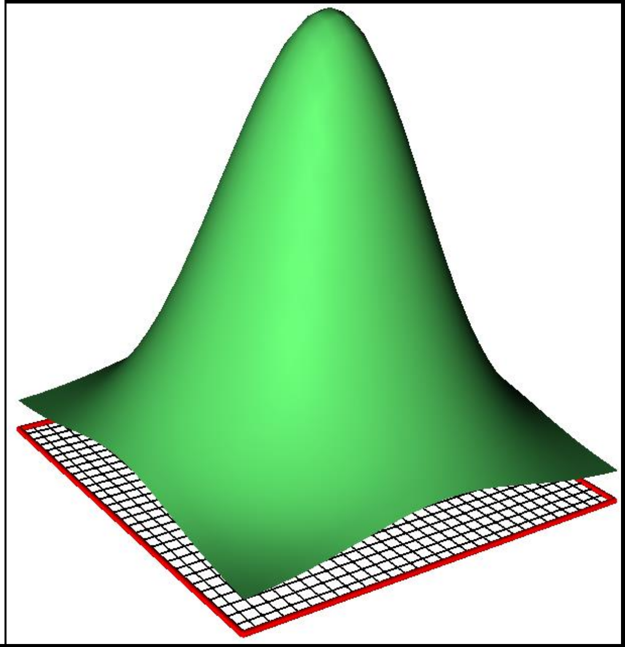
Effect of Reconstruction



Staircase Shading



Flat Shading



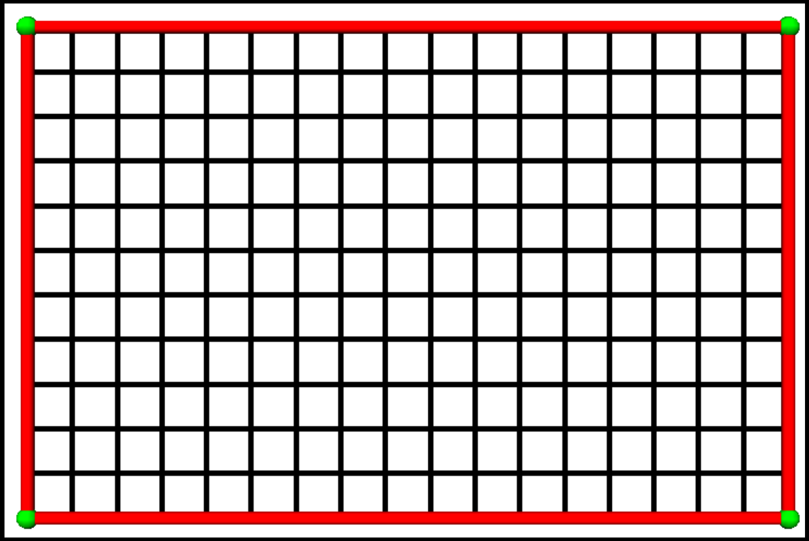
Smooth Shading

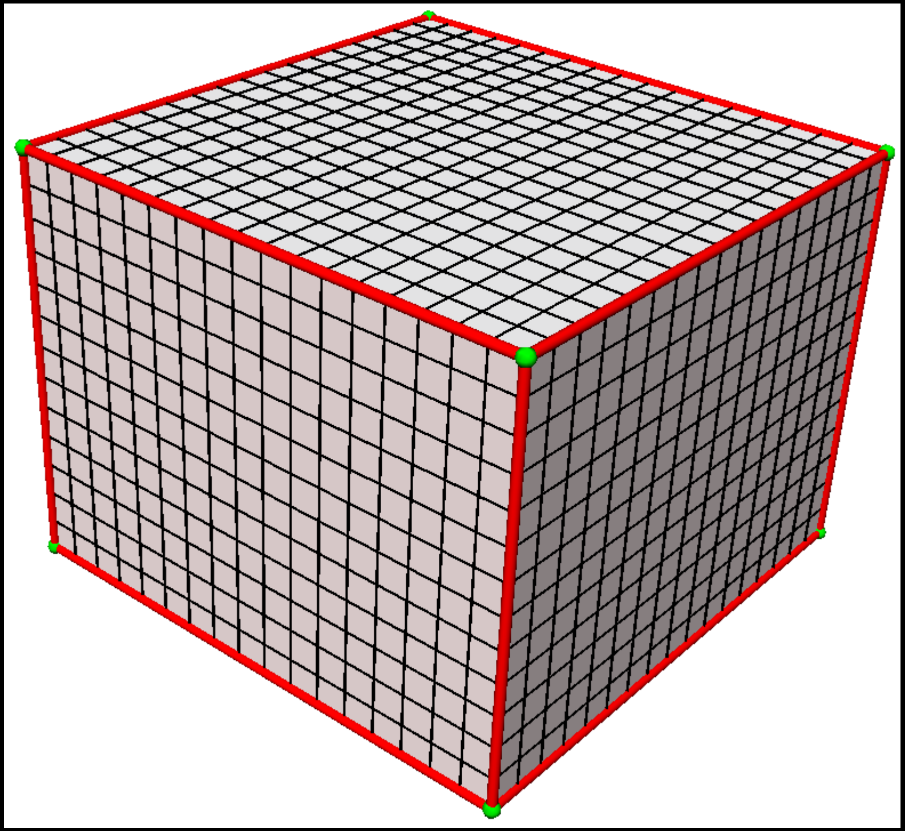
Grid types

- Grid is the pattern of cells in the data domain
- Grid is also called mesh

- Uniform grid
- Rectilinear grid
- Structured grid
- Unstructured grid

Uniform Grid





2-D

3-D

Uniform Grid

- The simplest grid type
- Domain D is usually an axis-aligned box
  - Line segment for d=1
  - Rectangle for d=2
  - parallelepiped for d=3
- Sample points are equally distributed on every axis
- **Structured coordinates**: the position of the sample points in the data domain are simply indicated by d integer coordinates (n_1,..n_d)
- Simple to implement
- Efficient to run (storage, memory and CPU)

Uniform Grid

- Data points are simply stored in the increasing order of the indices, e.g, an 1-D array
- Lexicographic order

$$i = n_1 + \sum_{k=2}^{d} (n_k \prod_{l=1}^{k-1} N_l)$$

$$\text{If d} = 2,$$

$$i = n_1 + n_2 N_1, \; or$$

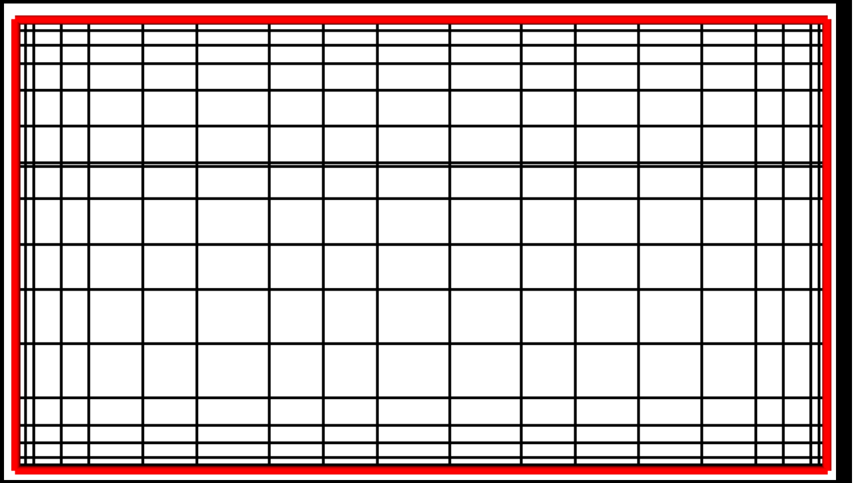
$$n_2 = i / N_1$$

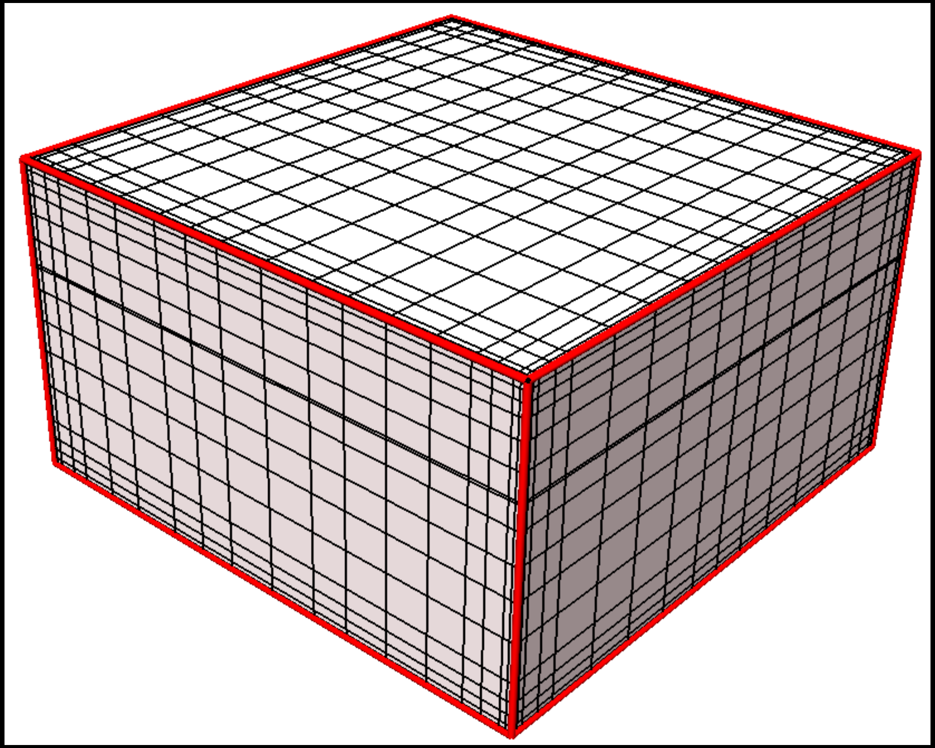
$$n_1 = i \ \text{ mod } \ (n_2 N_1)$$

$$\text{If d} = 3,$$

$$i = n_1 + n_2 N_1 + n_3 N_1 N_2$$

Rectilinear Grid





2-D

3-D

Rectilinear Grid

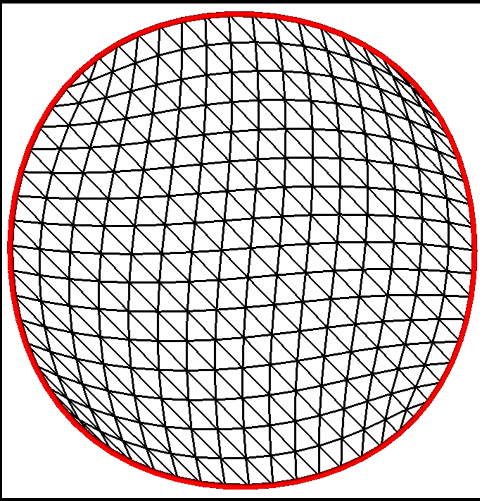
- Domain D is also an axis-aligned box
- However, the sampling step is not equal

- It is not as simple or as efficient as the uniform grid
- However, improving modeling power

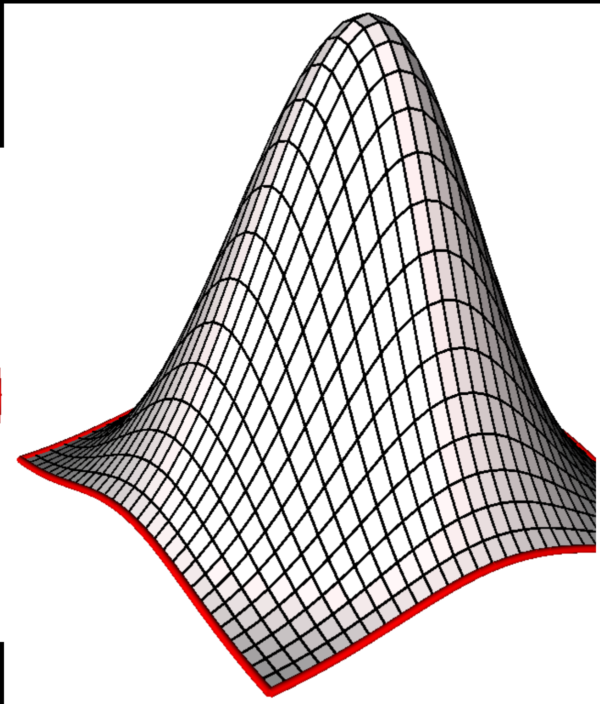
Structured Grid

- Further relaxing the constraint, a structured grid can be seen as the free deformation of a uniform or rectilinear grid
- The data domain can be non-rectangular
- It allows explicit placement of every sample points
- The matrix-like ordering of the sampling points are preserved
    - Topology is preserved
    - But, the geometry has changed

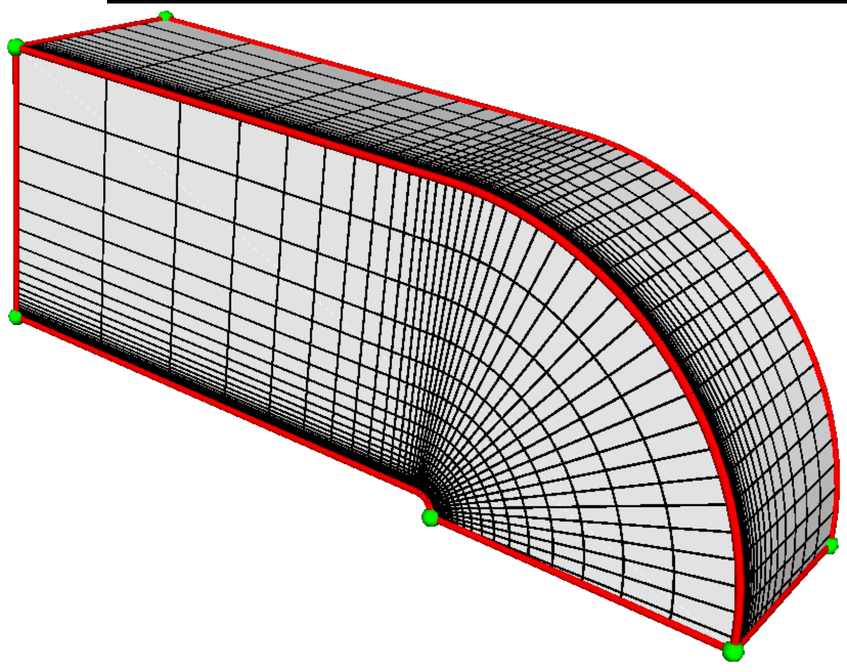
Structured Grid



Circular domain



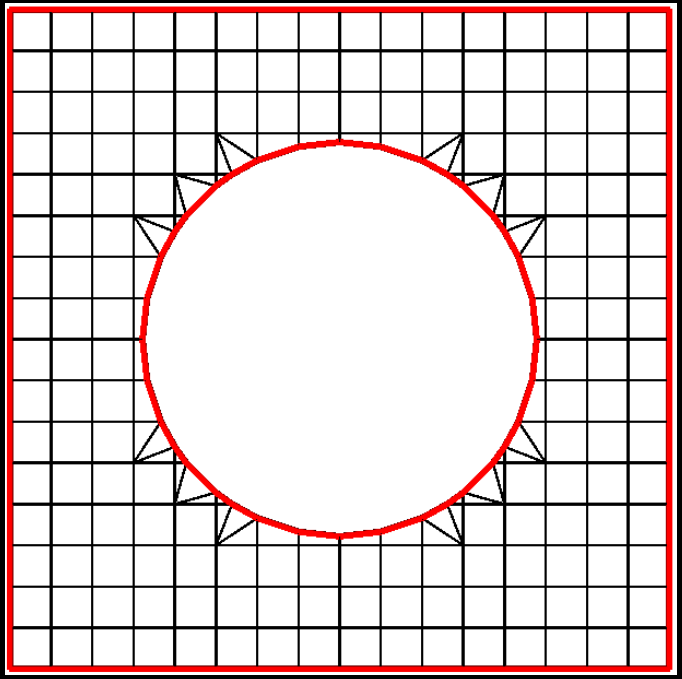
Curved Surface

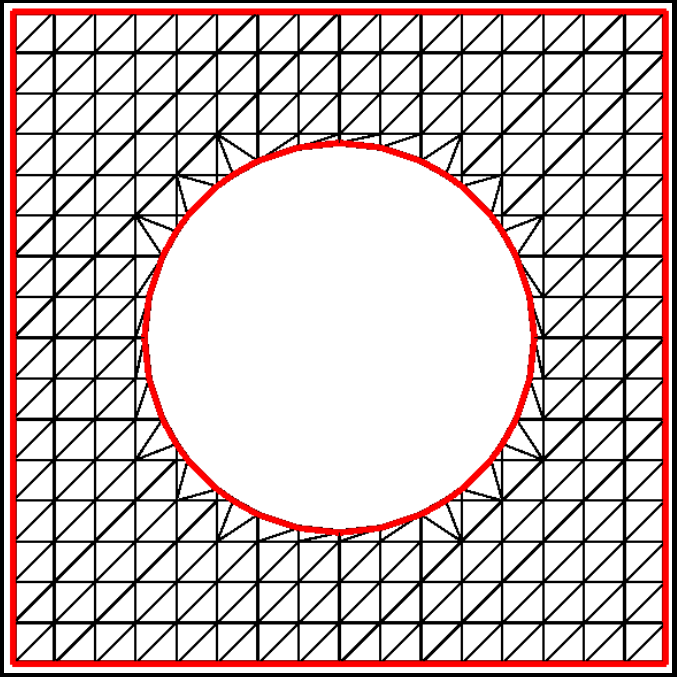


3D volume

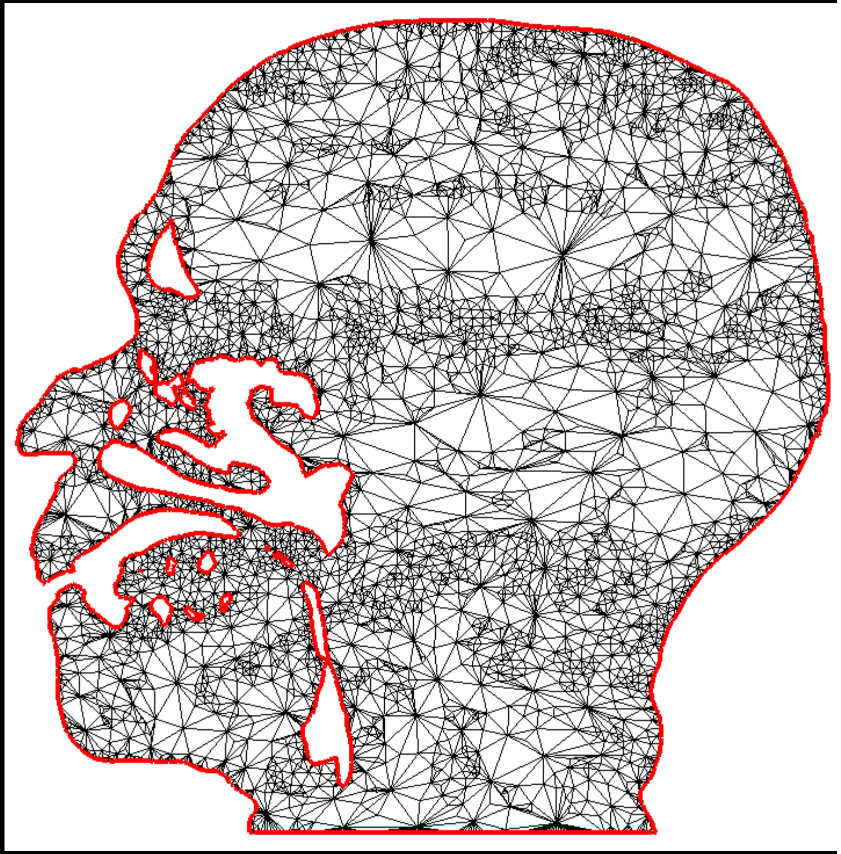
Unstructured Grid

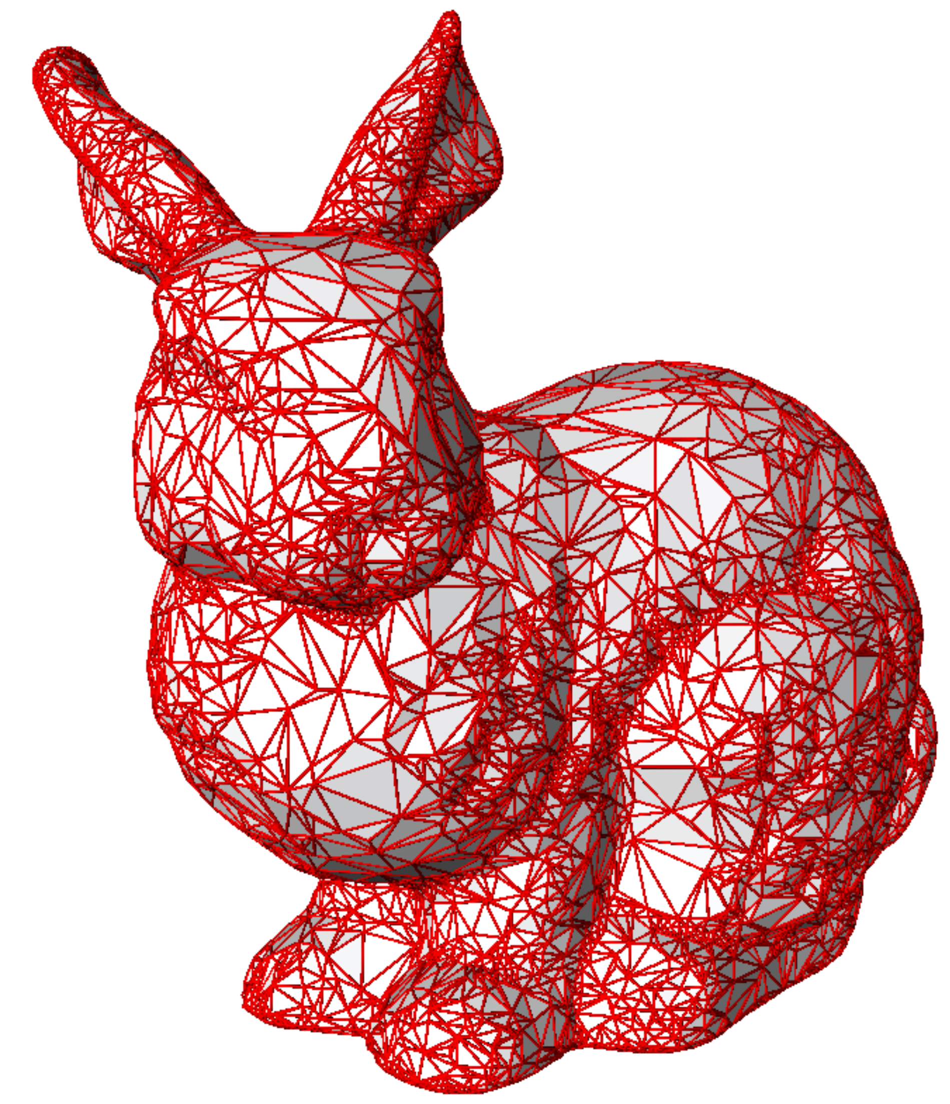
- It is allowed to define both sample points and cells explicitly
- The most general and flexible grid type
- However, it needs to store
  - The coordinates of all sample points p_i
  - For each cell, the set of vertex indices ci={v_{i1},…v_{iCi}), and for all cells {c1,c2…}





Unstructured Grid





Attributes

- Attribute data is the set of sample values of a sampled dataset

- Attribute = $\{f_i\}$

Sampled dataset

$$\boldsymbol{D_s} = (\{p_i\}, \{C_i\}, \{f_i\}, \{\Phi_i^k\})$$

Attribute Types

- Scalar Attribute $$\mathrm{C} \in \boldsymbol{R}^{c}$$ $$c = 1$$

- Vector Attribute $$\mathrm{C} \in \boldsymbol{R}^{c}$$ $$c = 2,\ \text{or}\ \ c = 3$$

- Color Attribute: c=3

- Tensor Attributes

- Non-Numerical Attributes

Scalar Attributes

$$f : R^2 \rightarrow R, \text{ or}$$

$$f : R^3 \rightarrow R$$

- E.g., temperature, density,

- **Scalar, Vector, Color, Tensor, Non-numerical**

Vector Attributes

$$f : R^2 \rightarrow R^2, \text{ or}$$
$$f : R^3 \rightarrow R^3$$

- E.g.,
- Normal
- Force
- velocity

- A vector has a magnitude and orientation

•**Scalar, Vector, Color, Tensor, Non-numerical**

Tensor Attributes

- A high-dimensional generalization of vectors

Tensor

$$\overleftrightarrow{V} = \vec{V}_A \vec{V}_B = \begin{pmatrix} V_{Ax}V_{Bx}, V_{Ax}V_{By} \\ V_{Ay}V_{Bx}, V_{Ay}V_{By} \end{pmatrix}$$

Vector

$$\vec{V} = (Vx, Vy)$$

Scalar

$$V = V$$

- A tensor describes physical quantities that depend on direction

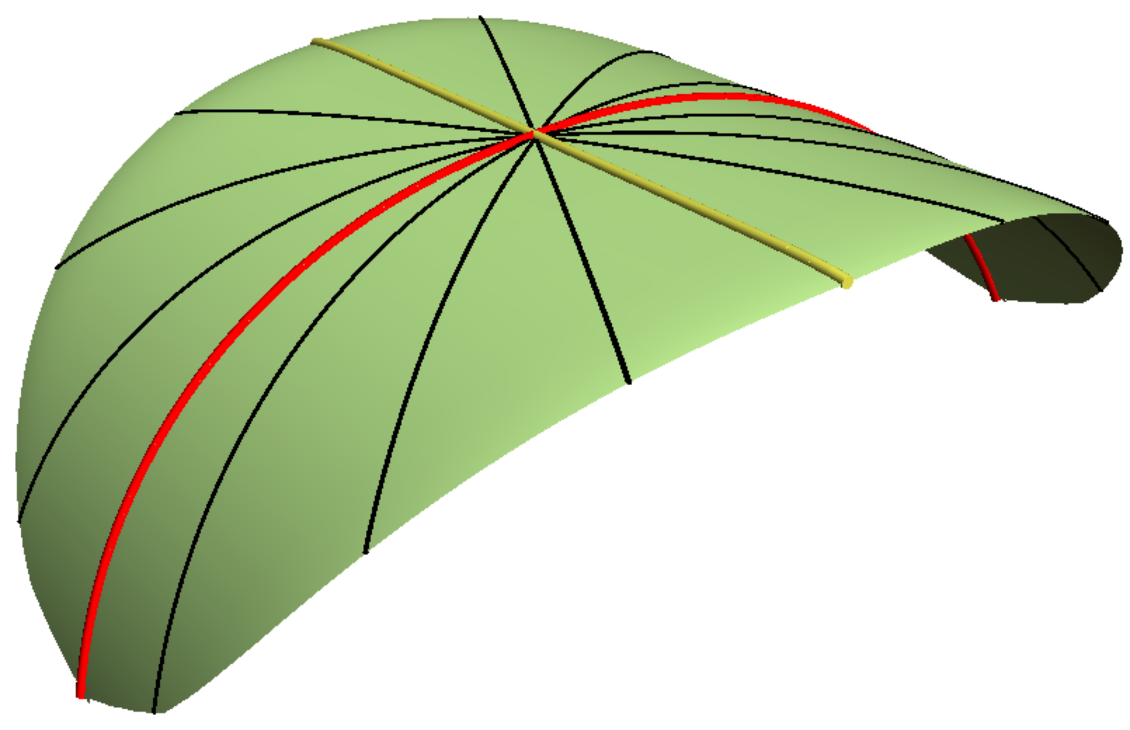
  Vector and scalar describes physical quantities that depend on position only

- Scalar, Vector, Color, Tensor, Non-numerical

Tensor Attributes

Tensor

- E.g. curvature of a 2-D surface



- E.g., diffusivity, conductivity, stress

- **Scalar, Vector, Color, Tensor, Non-numerical**

Non-numerical Attributes

- E.g. text, image, voice, and video
- Data can not be interpolated
- Therefore, the dataset has no basis function
- Domain of information of visualization (infovis)

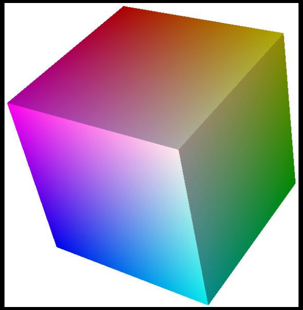
- **Scalar, Vector, Color, Tensor, Non-numerical**

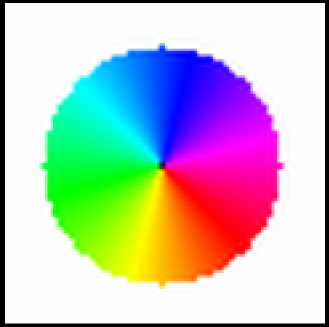
Color Attributes

- A special type of vector attributes with dimension c=3

- RGB system: convenient for hardware and implementation

  R: red
  G: green
  B: blue





- HSV system: intuitive for human user

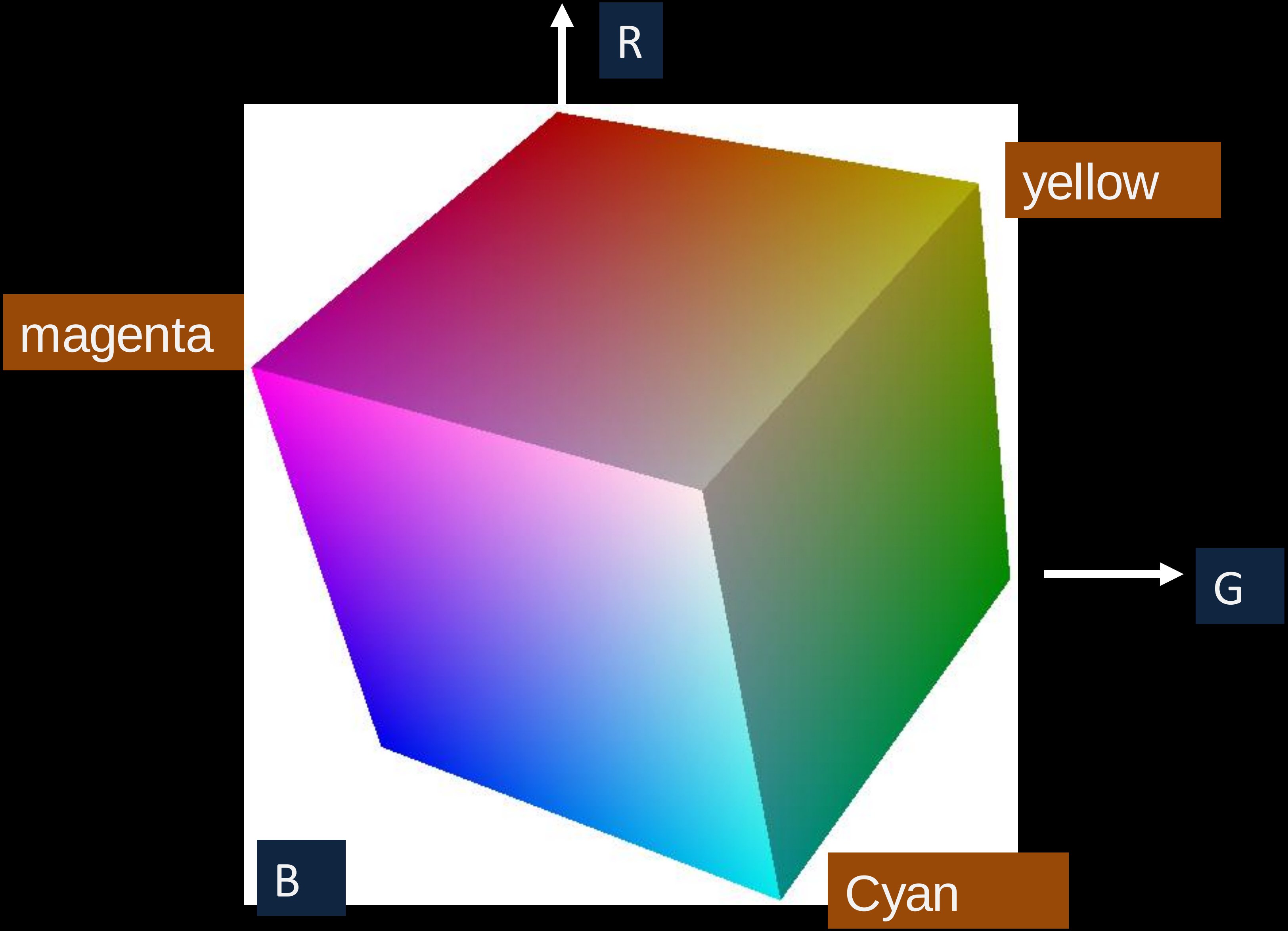
  H: Hue
  S: Saturation
  V: Value

Scalar, Vector, Color, Tensor, Non-numerical

RGB System

- Every color is represented as a mix of “pure” red, green and blue colors in different amount
- Equal amounts of the three colors determines gray shades
- RGB cube’s main diagonal line connecting the points (0,0,0) and (1,1,1) is the locus of all the grayscale value

RGB Cube



R

yellow

magenta

G

B

Cyan

HSV System

- Hue: distinguish between different colors of different wavelengths, from red to blue
- Saturation: represent the color of “purity”, or how much hue is diluted with white

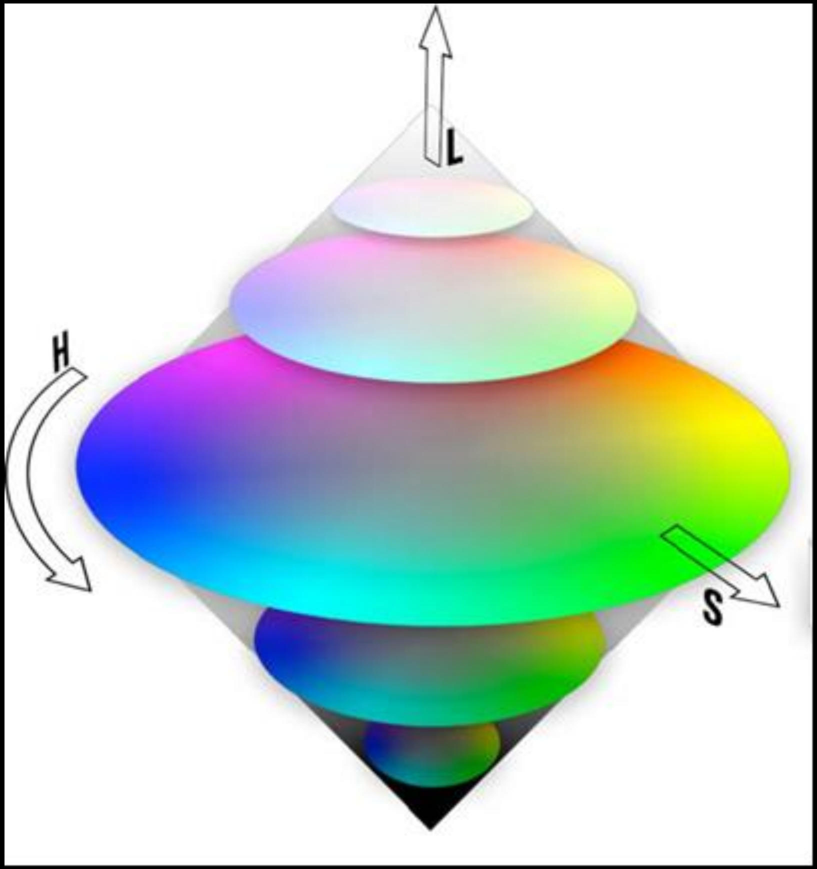
  S=1, pure, undiluted color

  S=0, white
- Value: represent the brightness, or luminance

  V=0, always dark

  V=1, brightest color for a given H and S

HSV System



HSV Color Cone

Color, Light, Electromagnetic Radiation

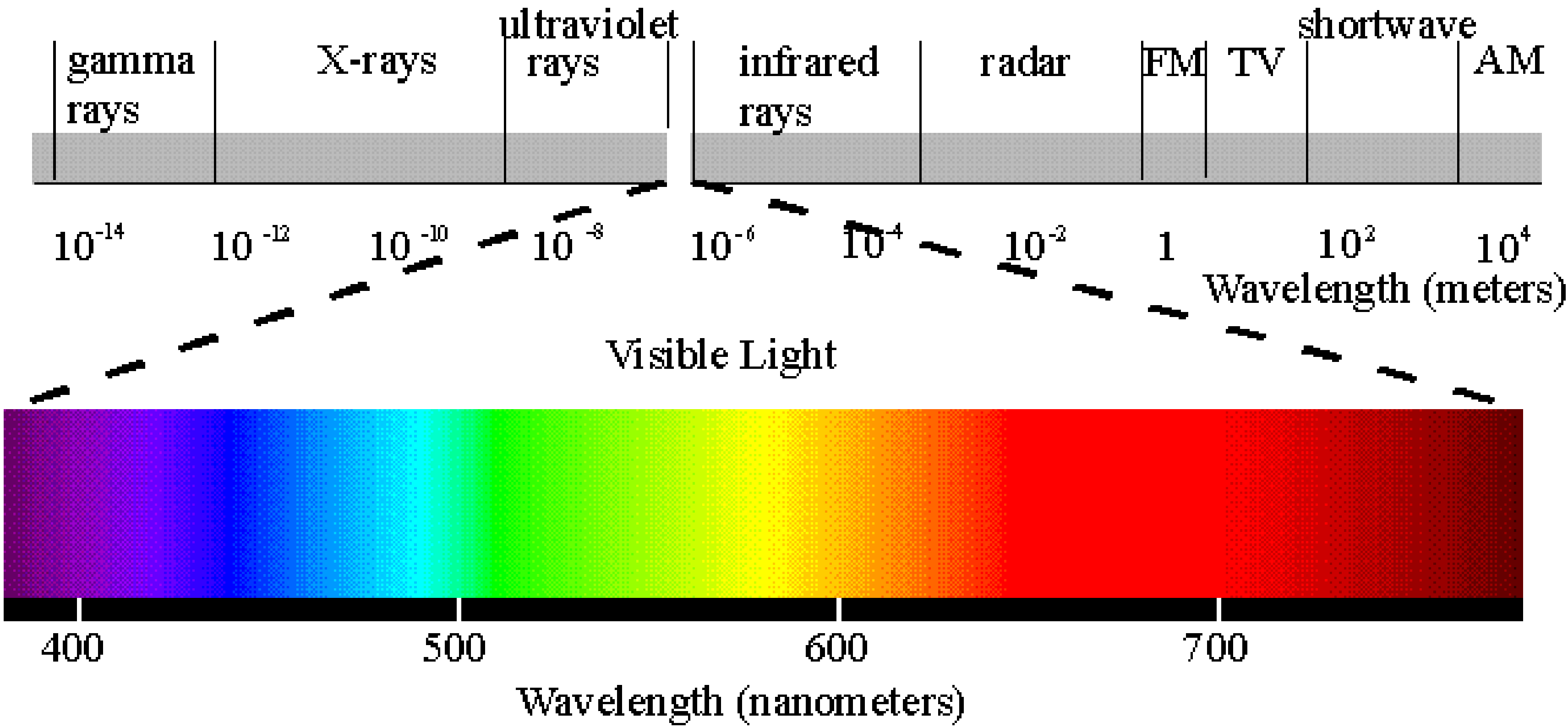


Table of Contents

Visual Stimulus

RGB to HSV

- All values are in [0,1]

max=max(R,G,B)
min=min(R,G,B)
diff=max-min

- V = max
    - largest RGB component
- S = diff/max
- For hue H, different cases
    - H = (G-B)/diff if R=max
    - H =2+(B-R)/diff if G=max
    - H =4+(R-G)/diff if B=max
    - then H=H/6
    - H=H+1 if H < 0

- Exp: Full Green Color
- (R,G,B)=(0,1,0) →
- (H,S,V)=(1/3, 1,1)

- Exp: Yellow Color
- (R,G,B)=(1,1,0) →
- (H,S,V)=(1/6, 1, 1)

HSV to RGB

- huecase = {int} (h*6)
- frac = 6*h – huecase

- lx= v*(1-s)
- ly= v*(1-s*frac)
- lz= v*(1-s(1-frac))

- huecase =6 (0<h<1/6): r=v, g=lz, b=lx
- huecase =1 (1/6<h<2/6): r=ly, g=v, b=lx
- huecase =2 (2/6<h<3/6): r=lx, g=v, b=lz
- huecase =3 (3/6<h<4/6): r=lx, g=ly, b=v
- huecase =4 (4/6<h<5/6): r=lz, g=lx, b=v
- huecase =5 (5/6<h<1): r=v, g=lx, b=ly

- Exp: Full Green Color
- (H,S,V)=(1/3,1,1) →
- (R,G,B)=(0,1,0)

- Exp: Yellow Color
- (H,S,V)=(1/6,1,1) →
- (R,G,B)=(1,1,0)

Conclusion

- Fundamental issues involved in representing data for visualization applications
- A set of data cells
- Data attributes, several types: scalar vector color and tensor
- Basis function: constant and linear

  Simplicity of implementation and direct support in the graphics hardware
- Grid Types: uniform, rectilinear, structured and unstructured grids