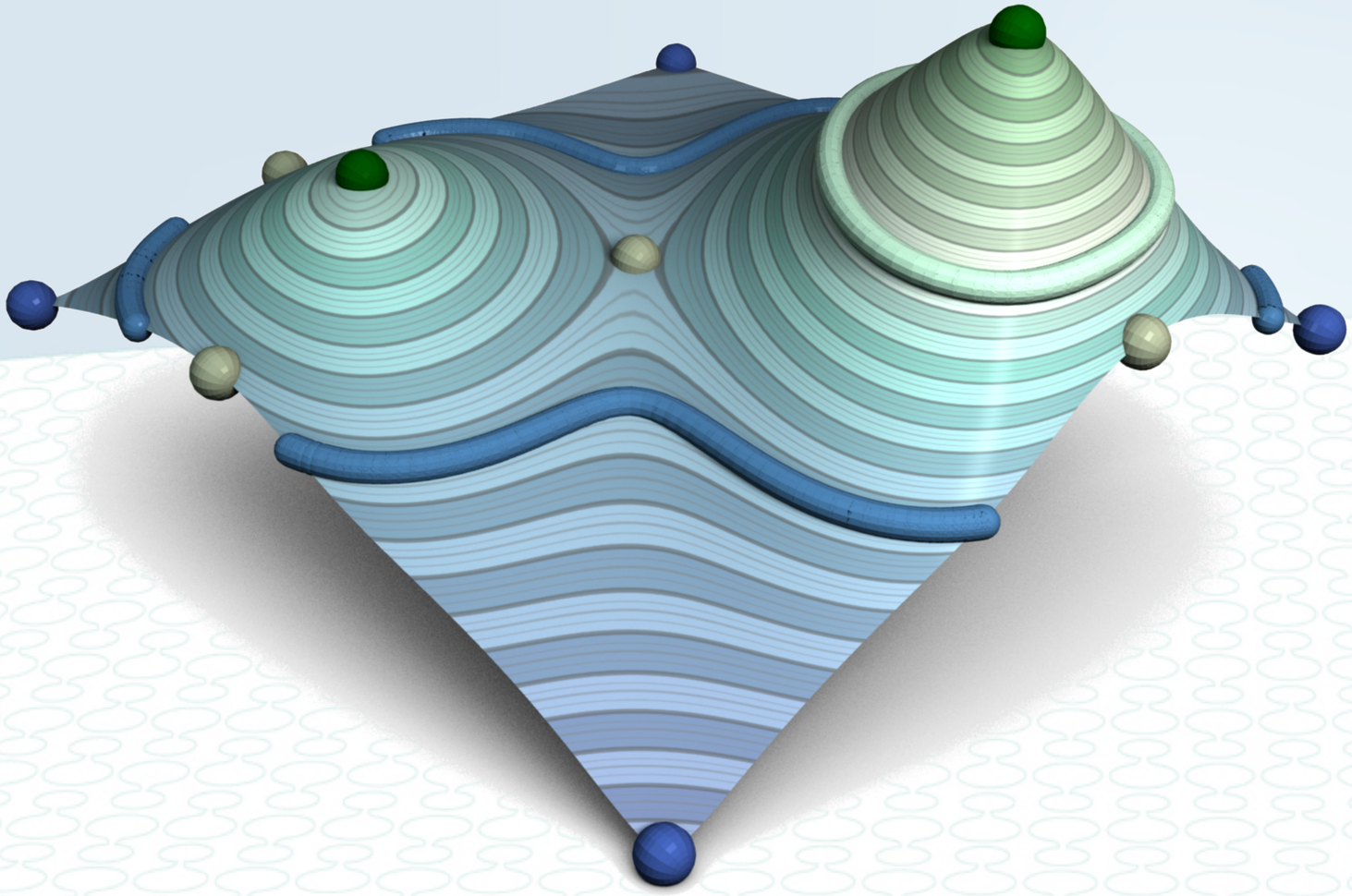
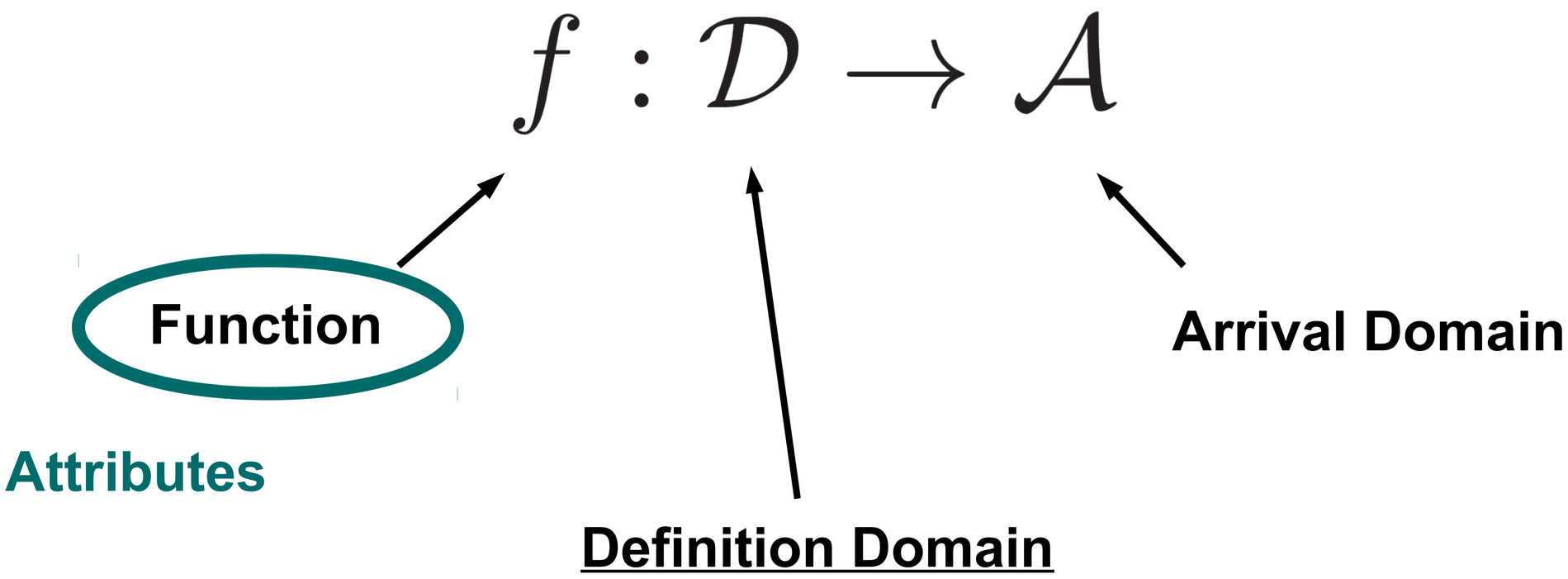


Scalar Field Visualization

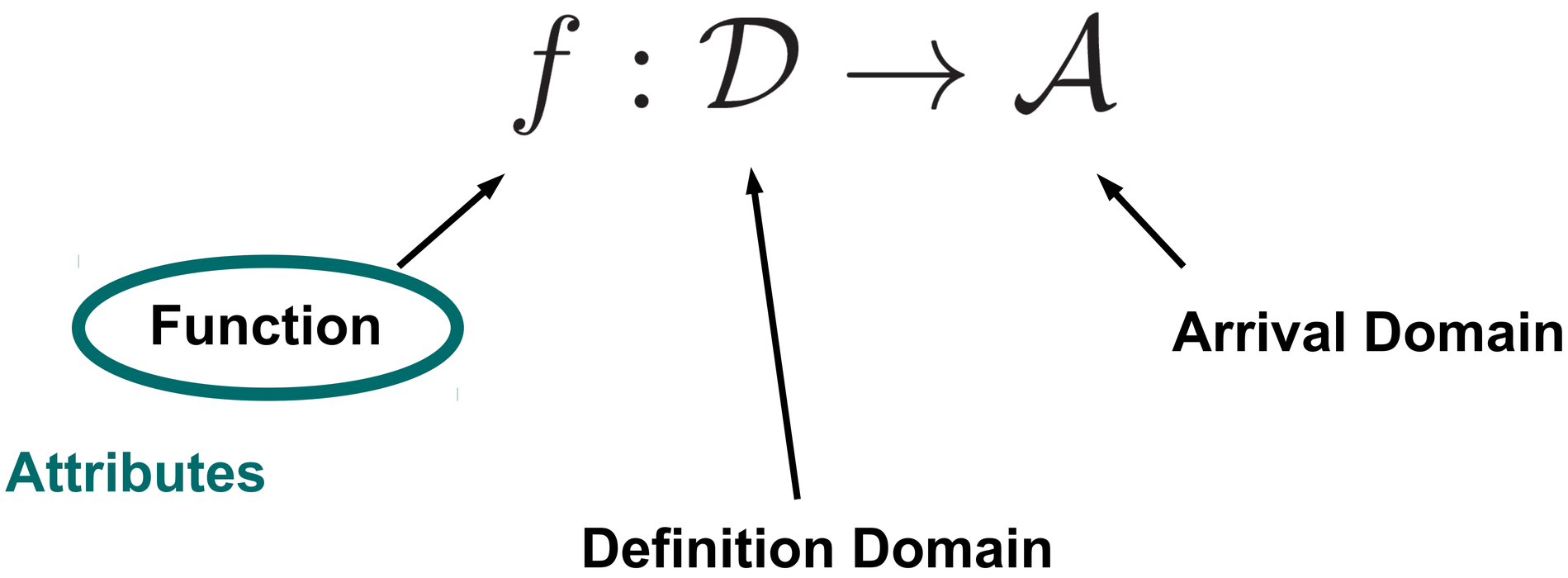
Master M2S

J. Tierny

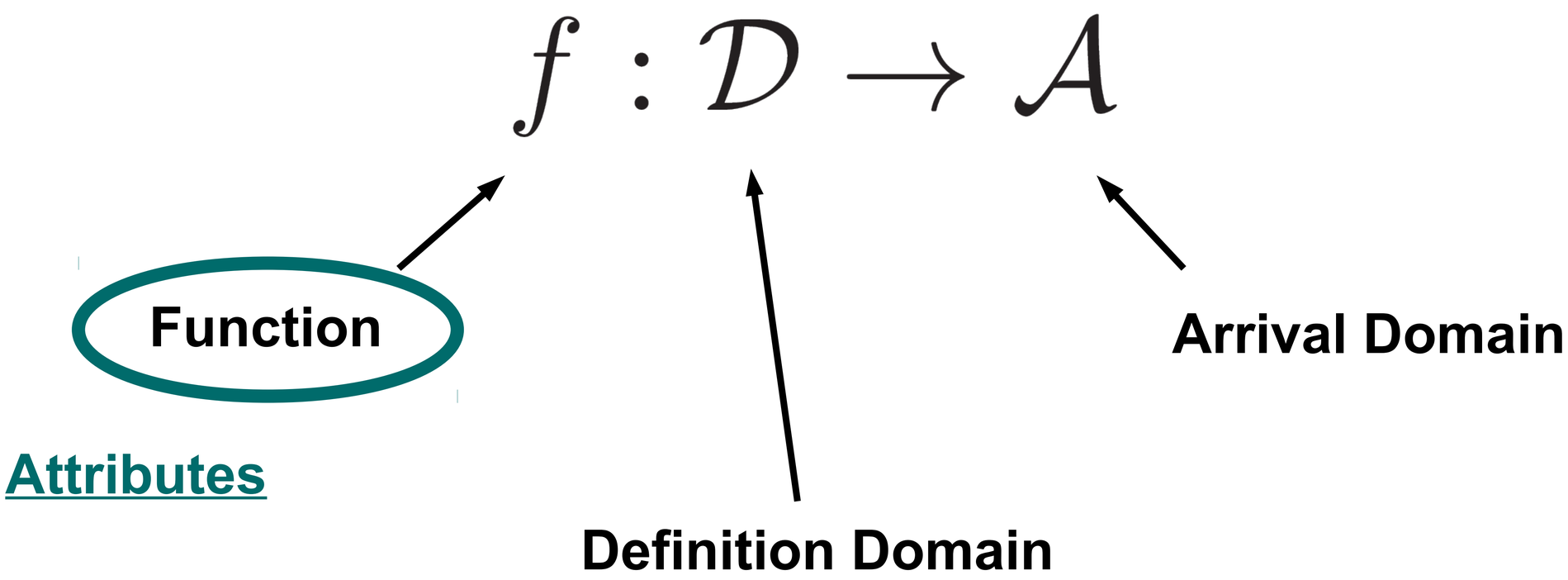
Previously


$f : \mathcal{D} \to \mathcal{A}$
Function
Attributes
Definition Domain
Arrival Domain

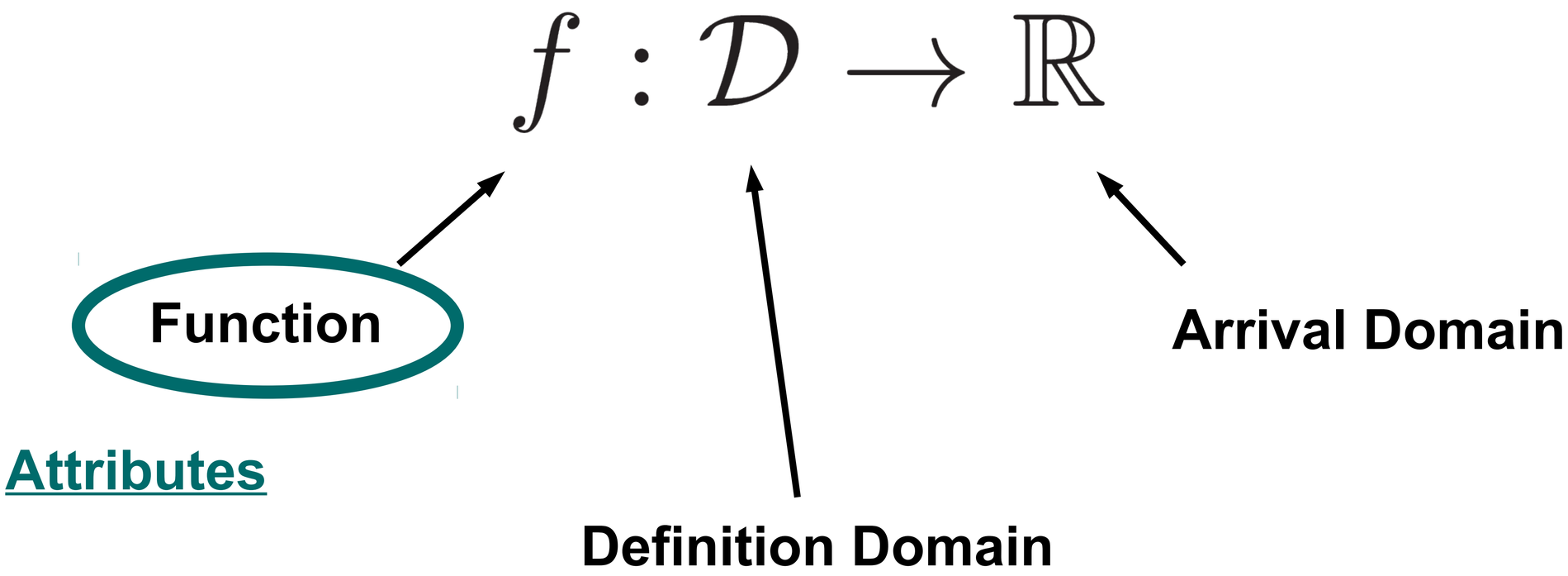
Previously


$f : \mathcal{D} \to \mathcal{A}$
Function
Attributes
Definition Domain
Arrival Domain

Previously


$f : \mathcal{D} \to \mathcal{A}$
Function
Attributes
Definition Domain
Arrival Domain

Previously


$f : \mathcal{D} \to \mathbb{R}$
Function
Attributes
Definition Domain
Arrival Domain

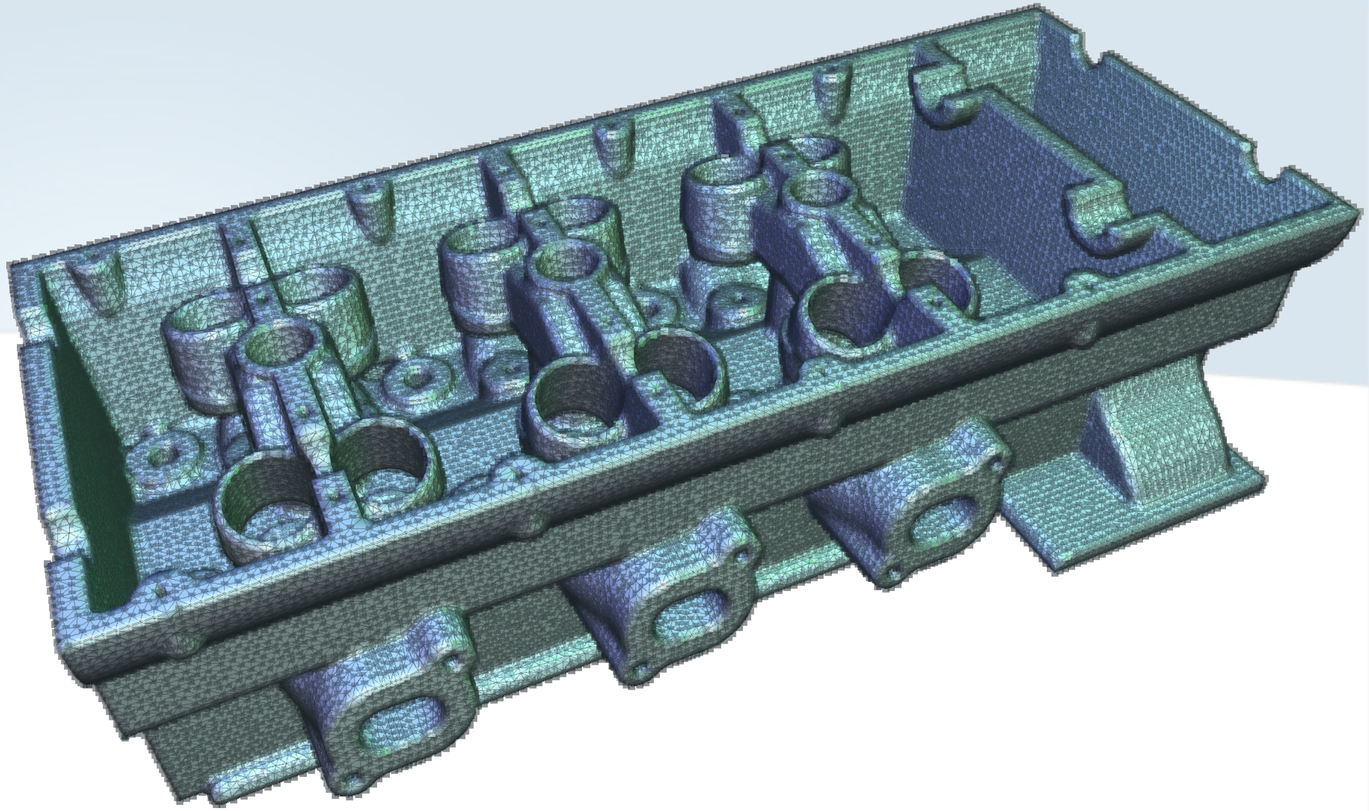
In practice

In practice

- Given a domain $\mathcal{D}$

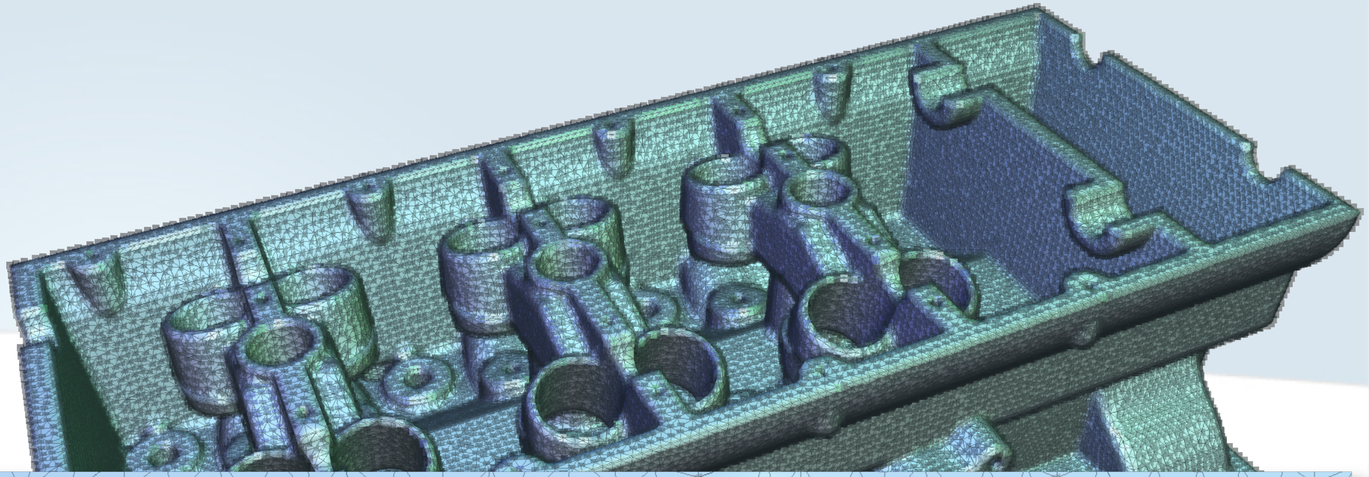
In practice

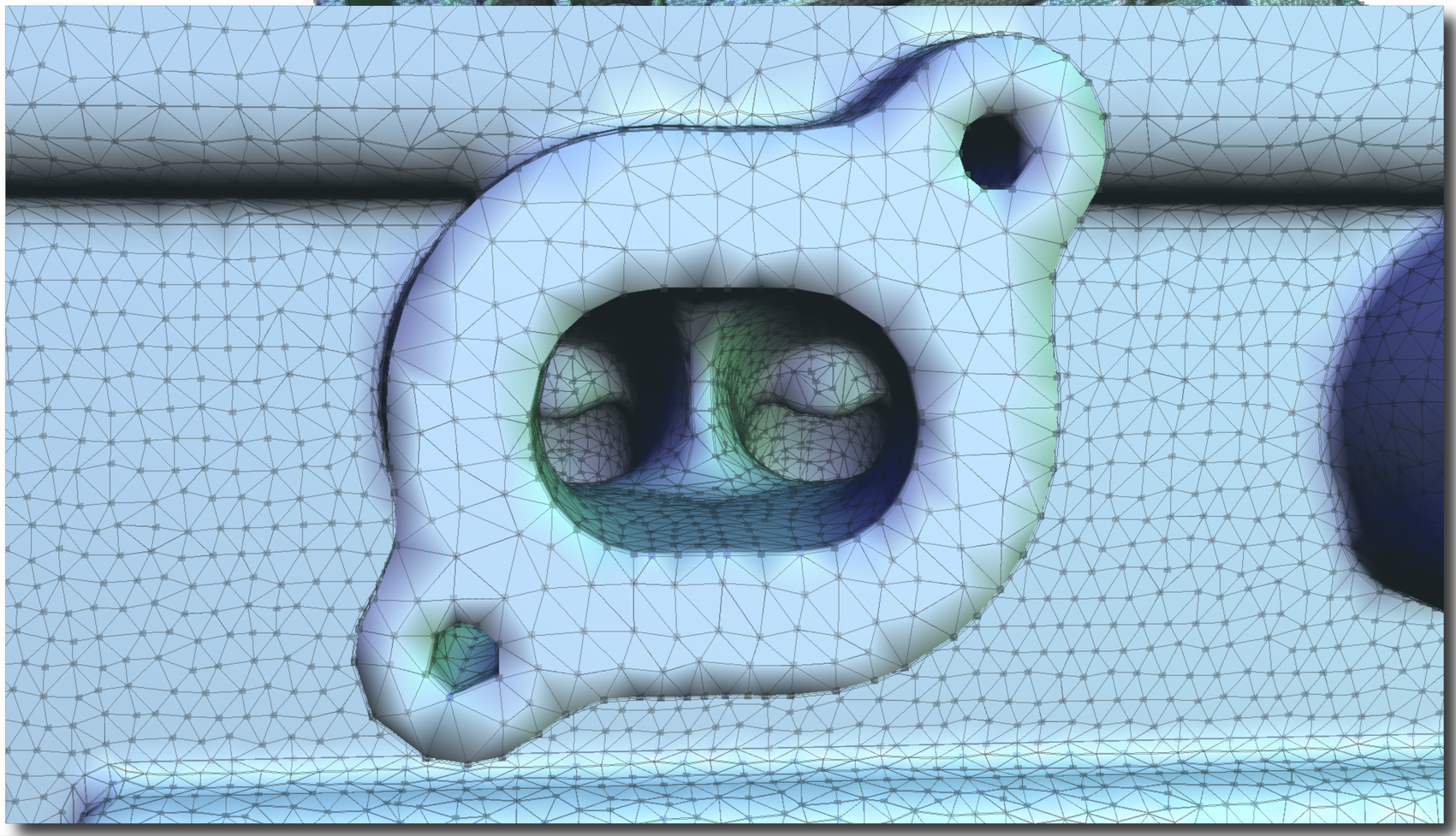
- Given a domain $\mathcal{D}$



In practice

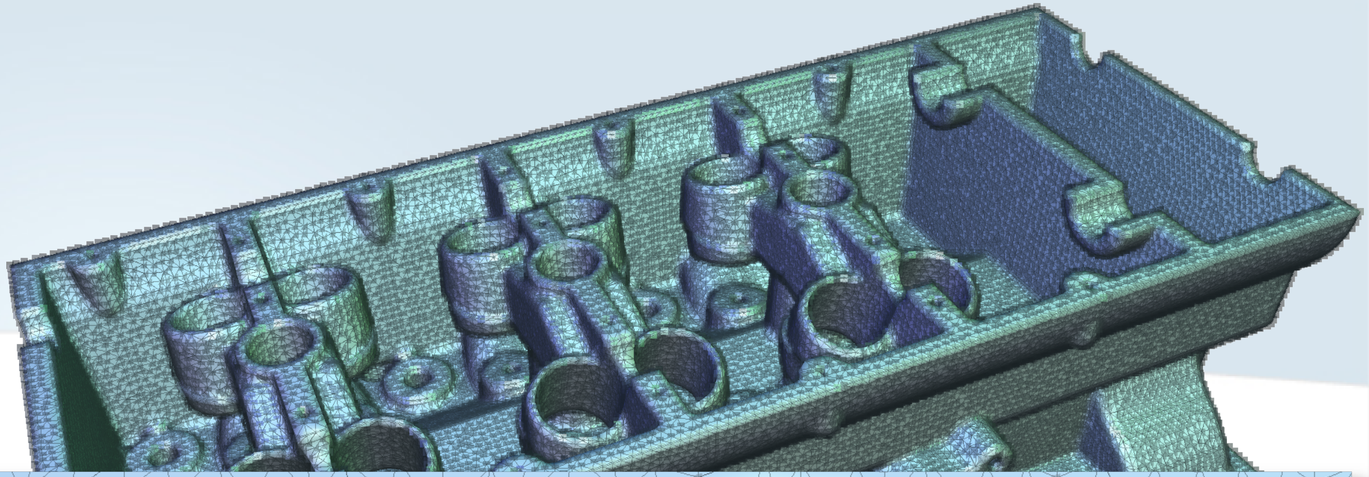
- Given a domain $\mathcal{D}$

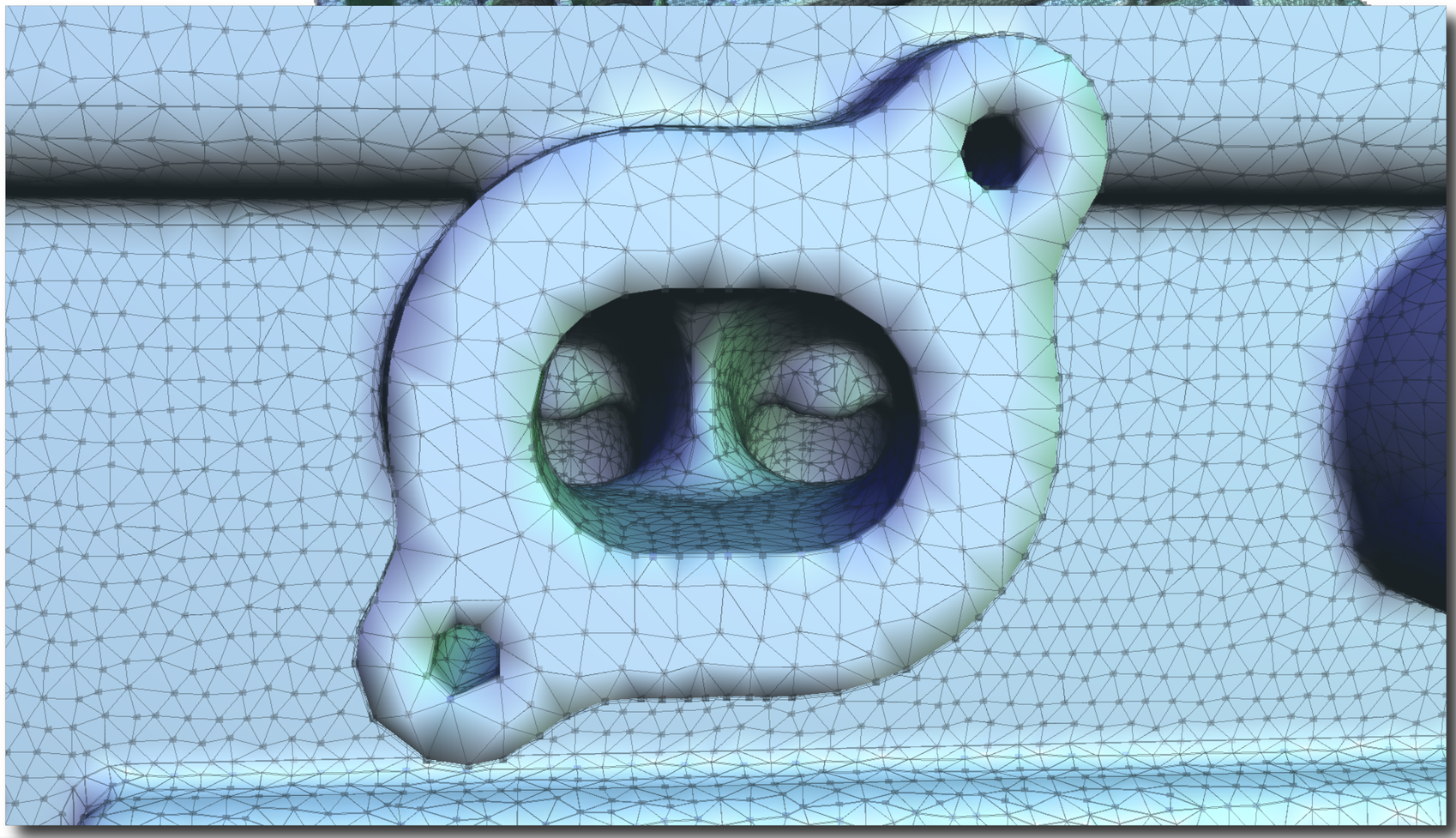




In practice

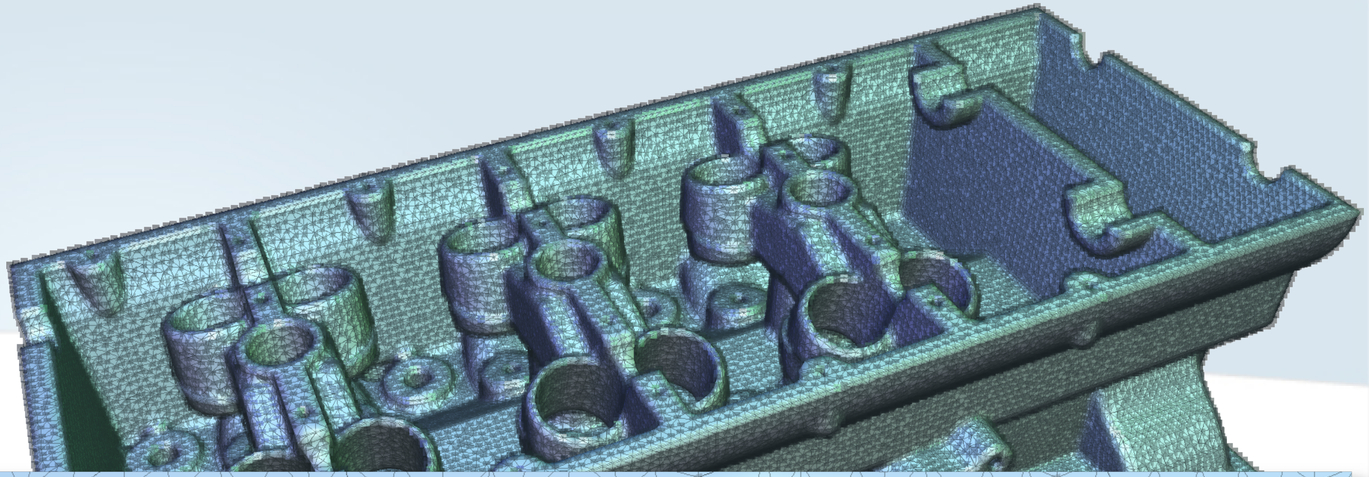
- Given a domain $\mathcal{D}$
- One scalar $f(v)$

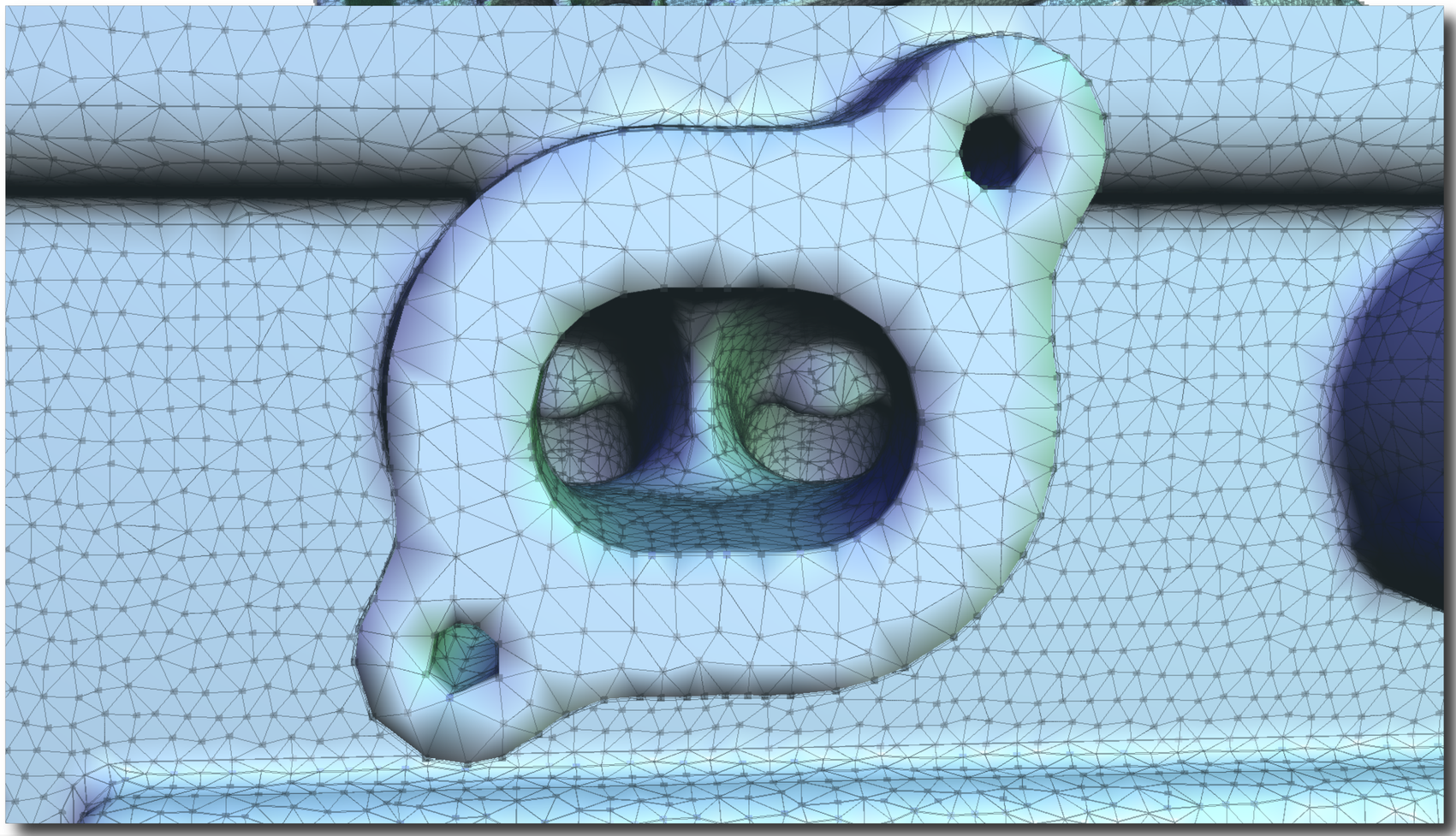




In practice

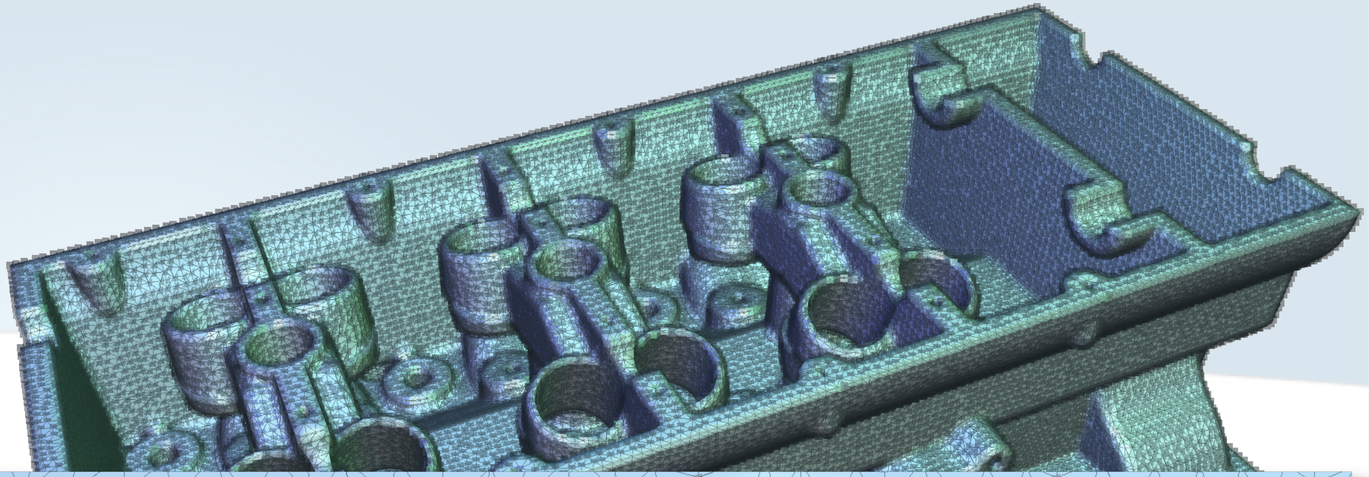
- Given a domain $\mathcal{D}$
- One scalar $f(v)$
- For each vertex v

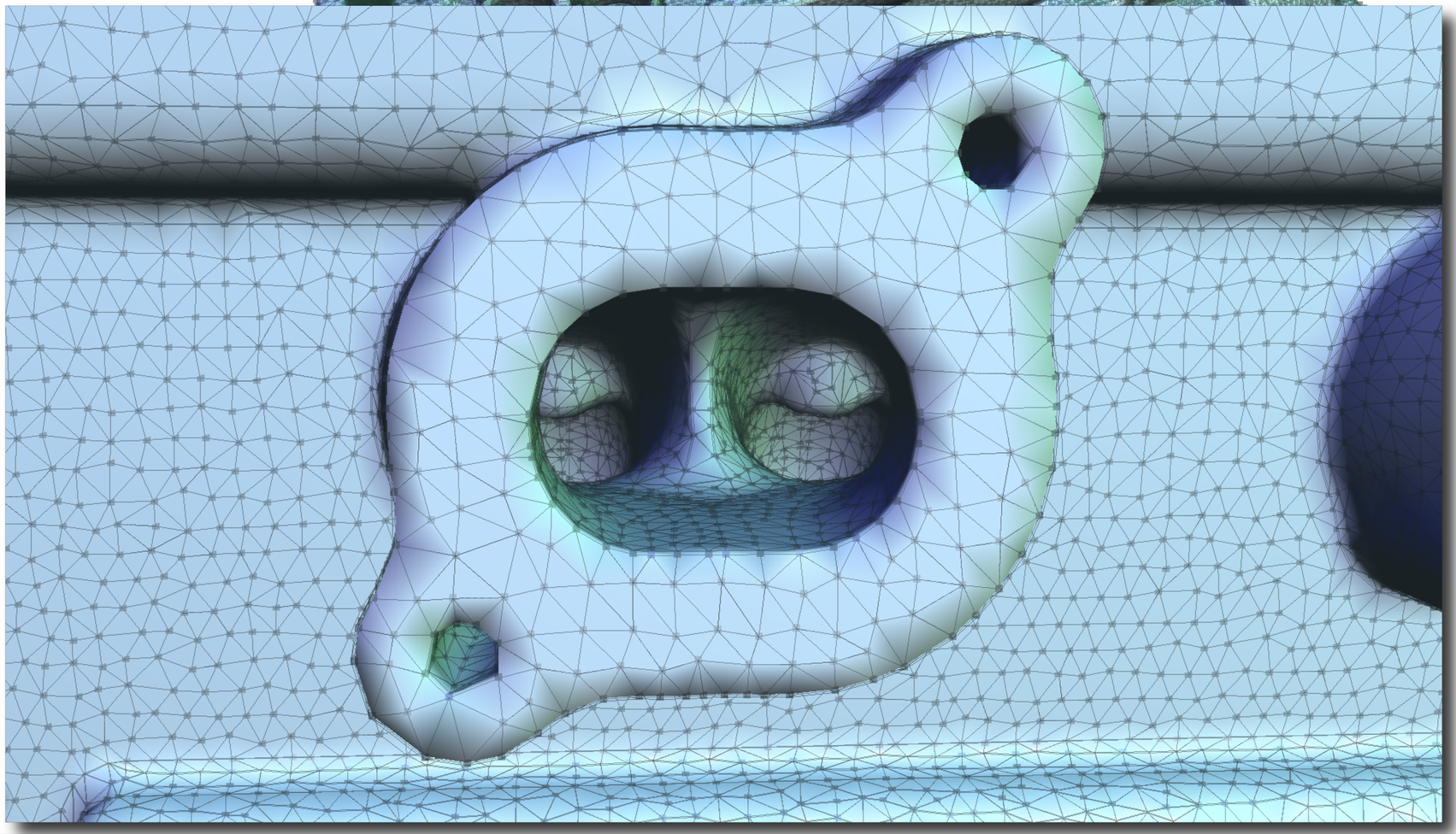




In practice

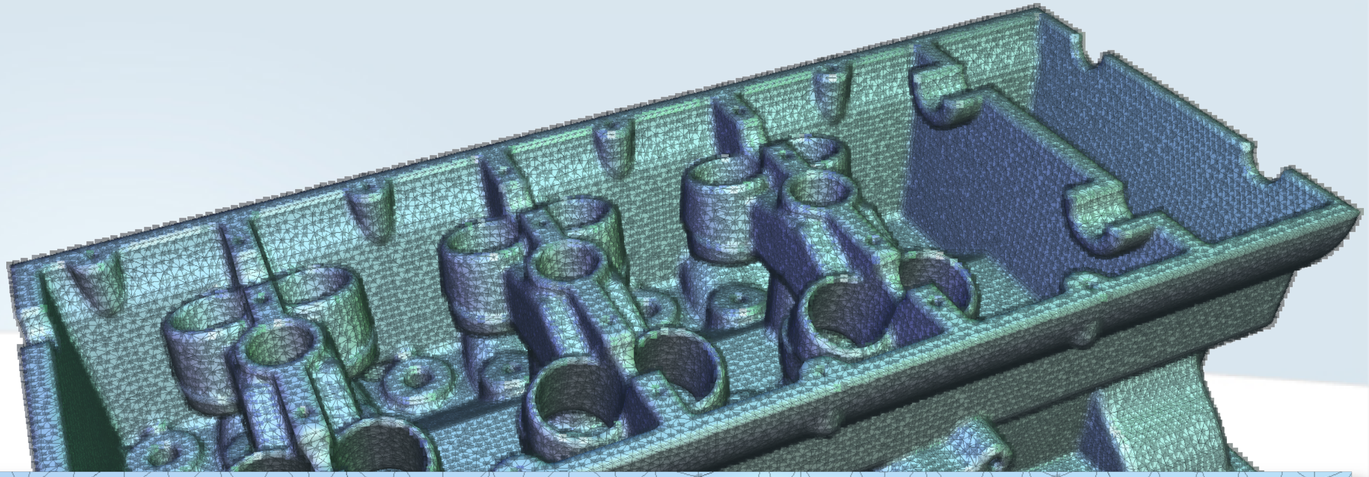
- Given a domain $\mathcal{D}$
- One scalar $f(v)$
- For each vertex v
- Interpolation on the other simplices

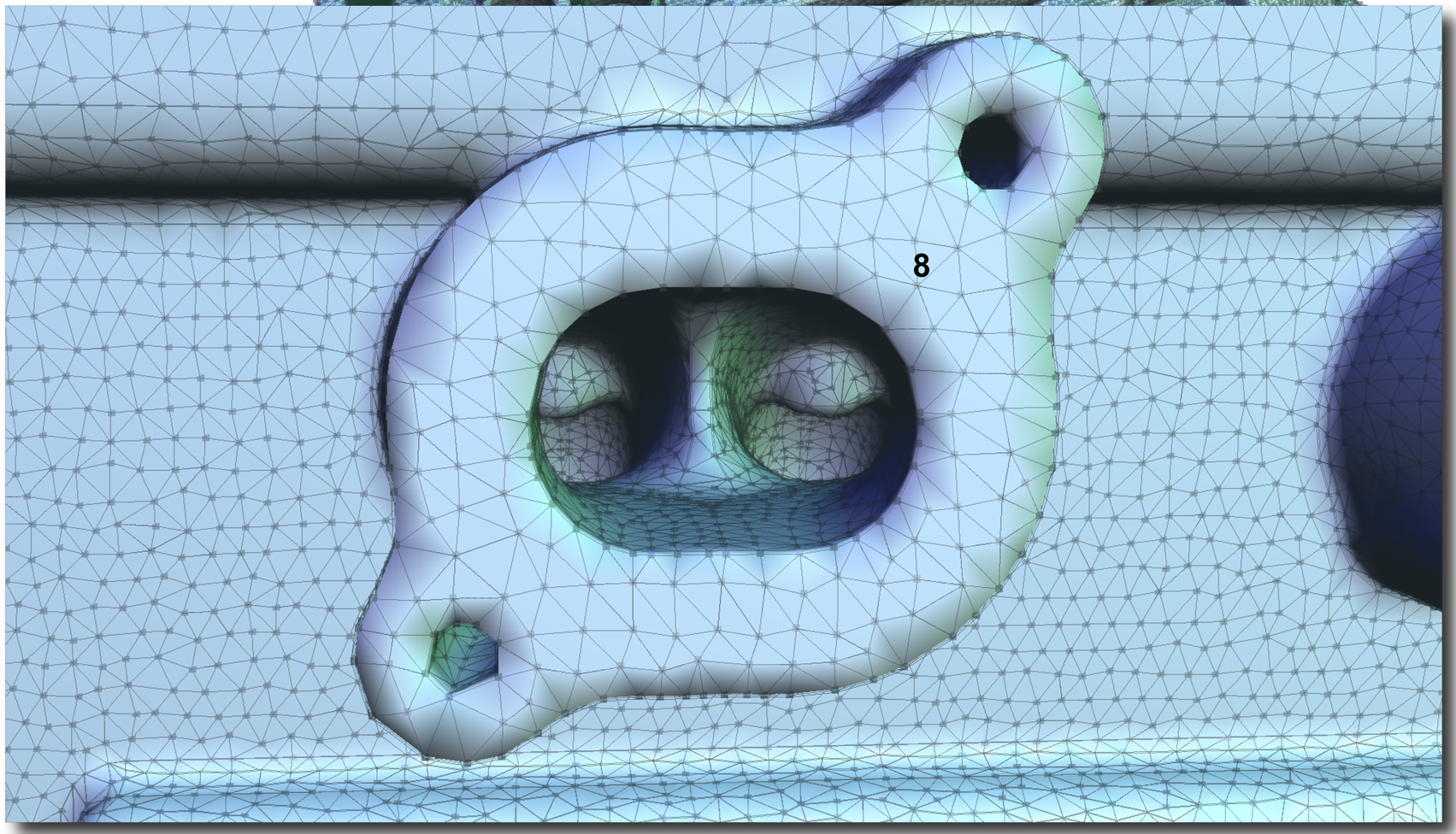




In practice

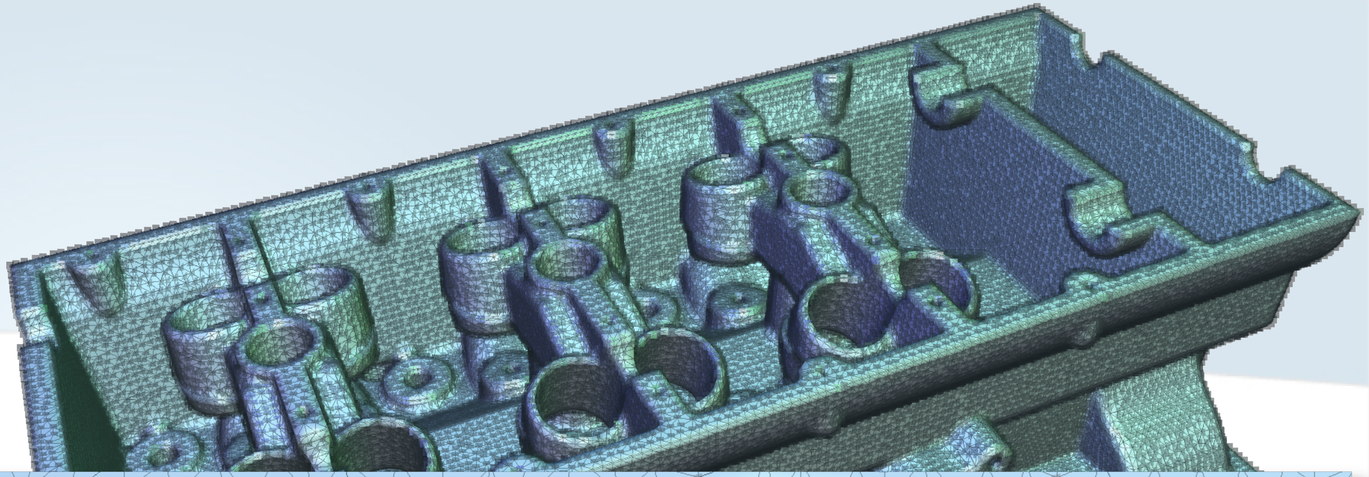
- Given a domain $\mathcal{D}$
- One scalar $f(v)$
- For each vertex v
- Interpolation on the other simplices

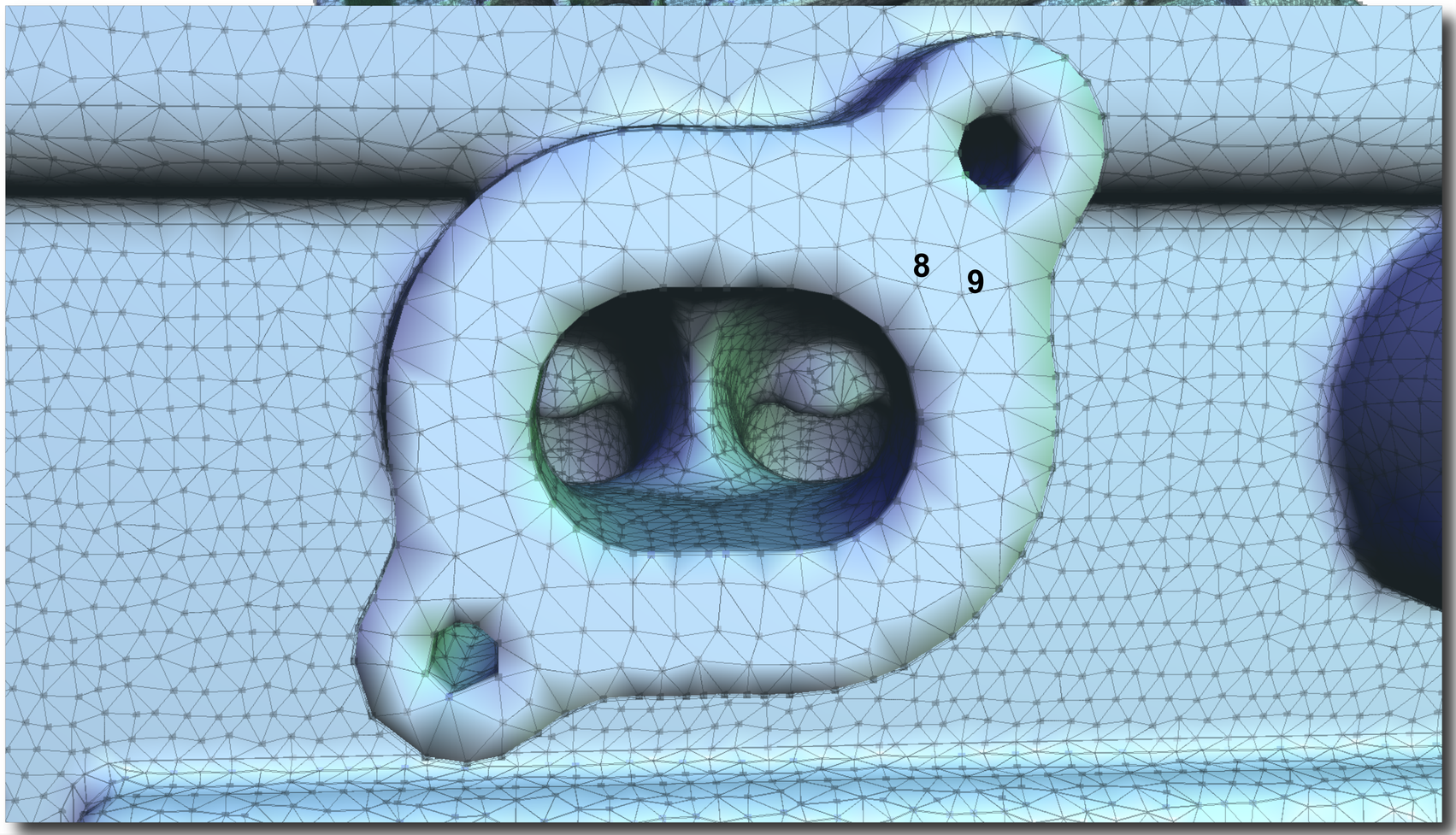




In practice

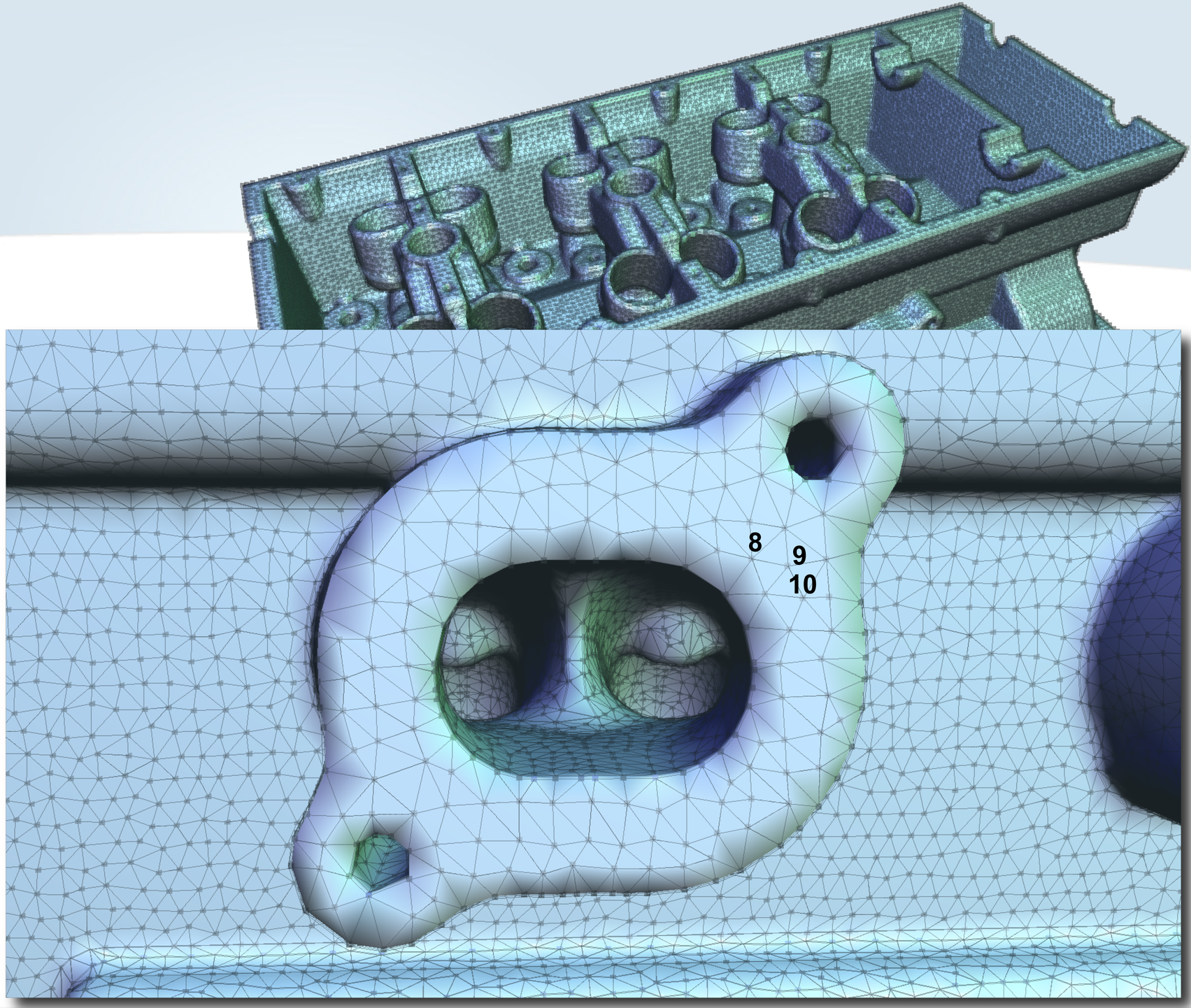
- Given a domain $\mathcal{D}$
- One scalar $f(v)$
- For each vertex v
- Interpolation on the other simplices





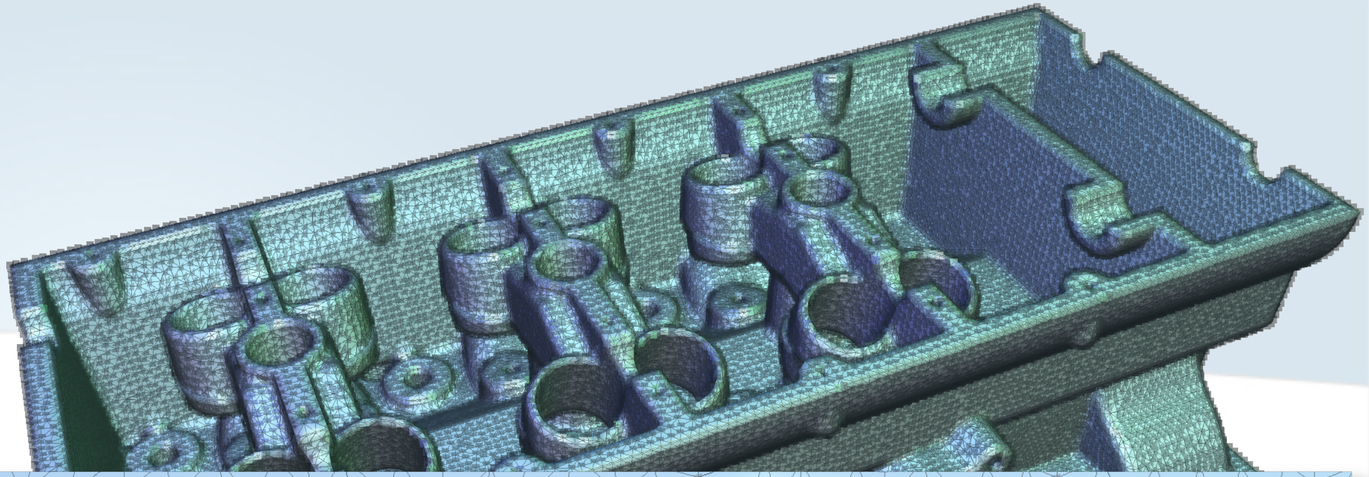
In practice

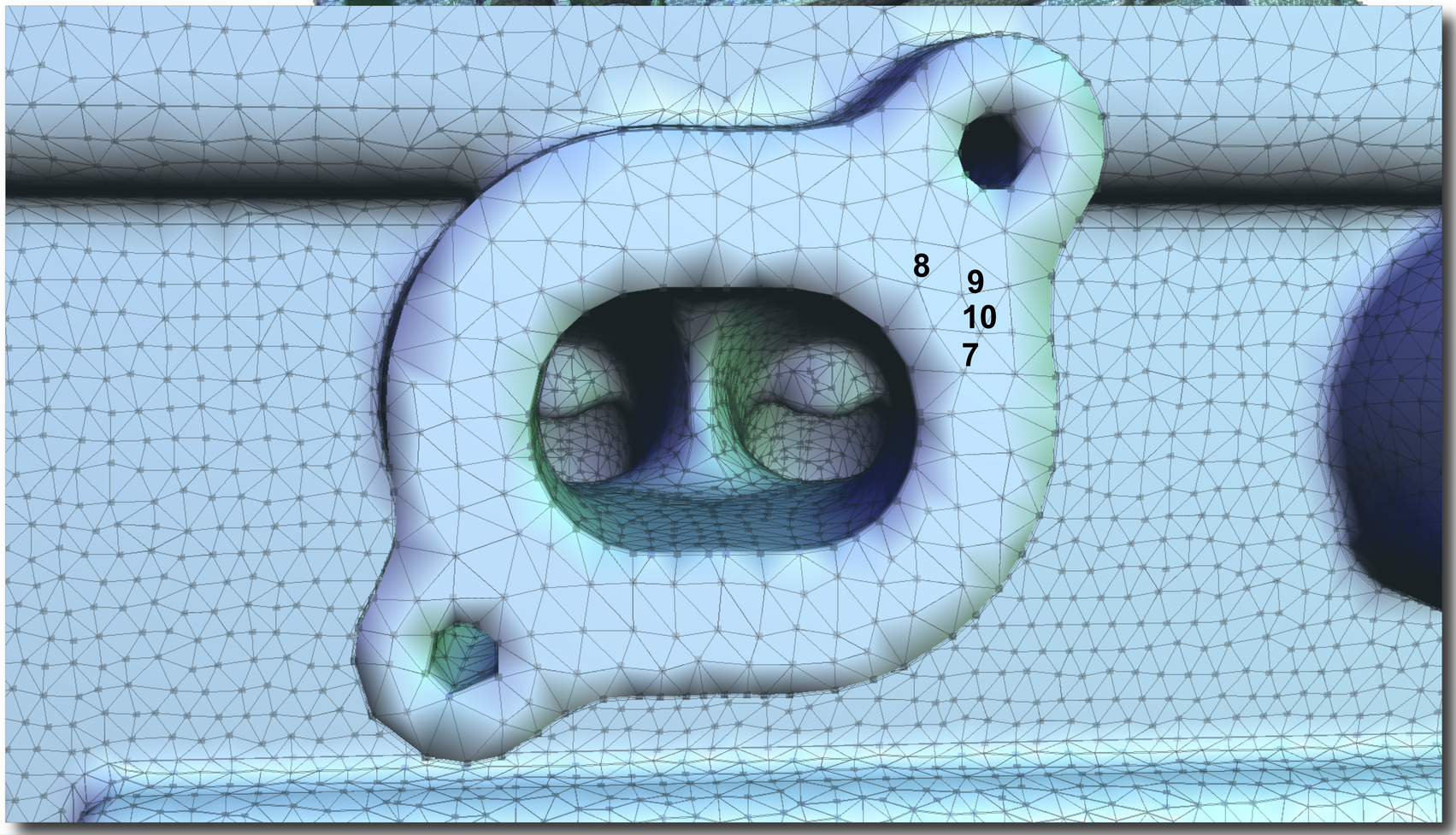
- Given a domain $\mathcal{D}$
- One scalar $f(v)$
- For each vertex v
- Interpolation on the other simplices



In practice

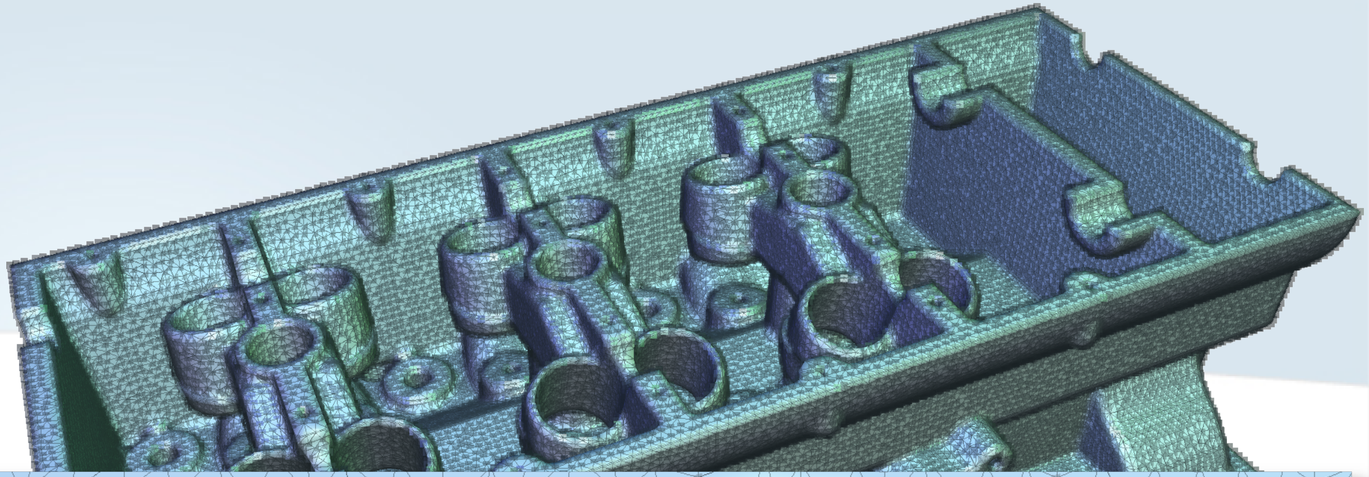
- Given a domain $\mathcal{D}$
- One scalar $f(v)$
- For each vertex v
- Interpolation on the other simplices

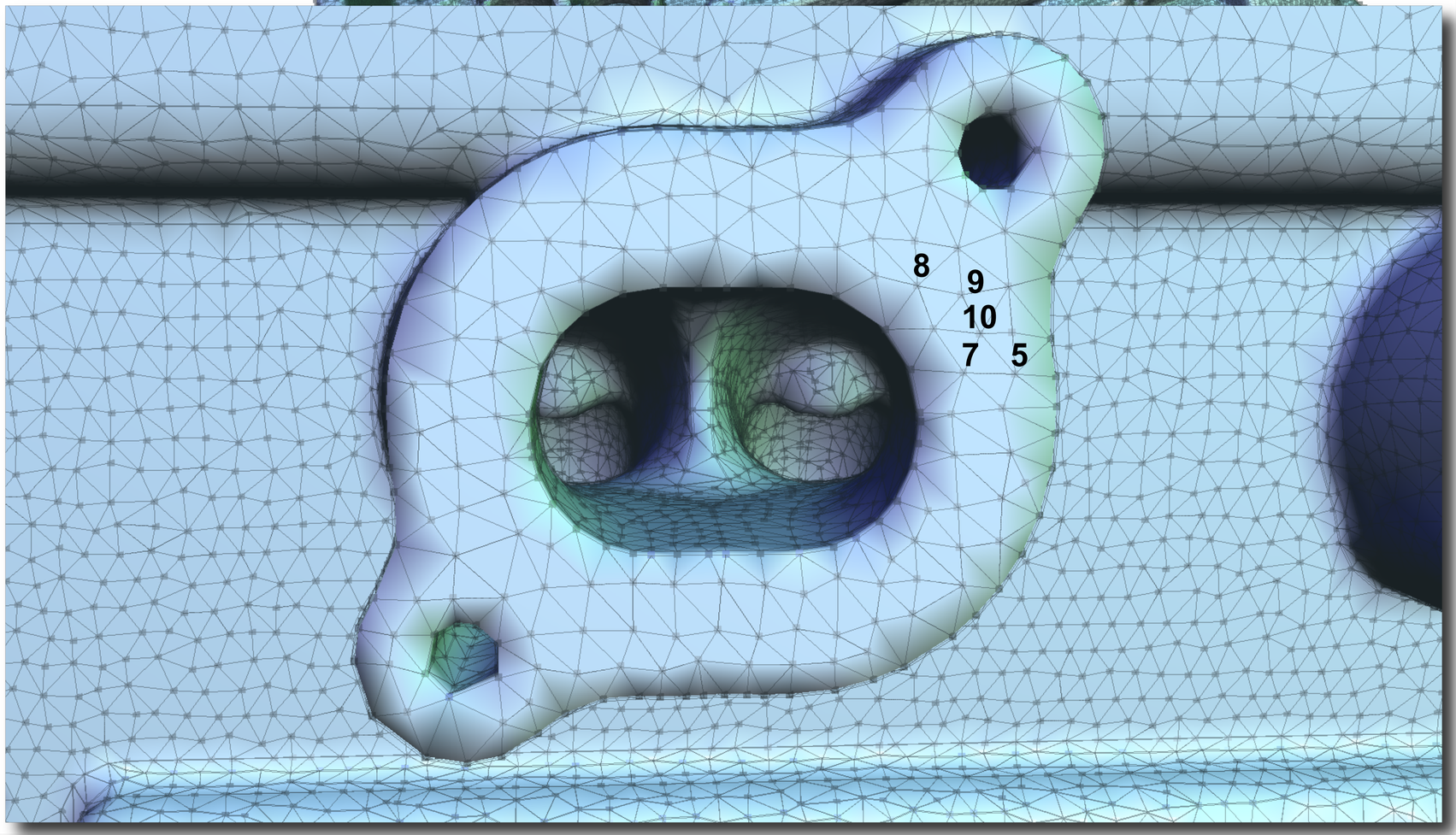




In practice

- Given a domain $\mathcal{D}$
- One scalar $f(v)$
- For each vertex v
- Interpolation on the other simplices





Basic idea

- Intuitive technique
  - Exploit additional dimensions for the embedding

Basic idea

- Intuitive technique
  - Exploit additional dimensions for the embedding

$$f : [0, 24] \rightarrow \mathbb{R}$$

Basic idea

- Intuitive technique
  - Exploit additional dimensions for the embedding

$$f : [0, 24] \rightarrow \mathbb{R}$$

$$x = x$$

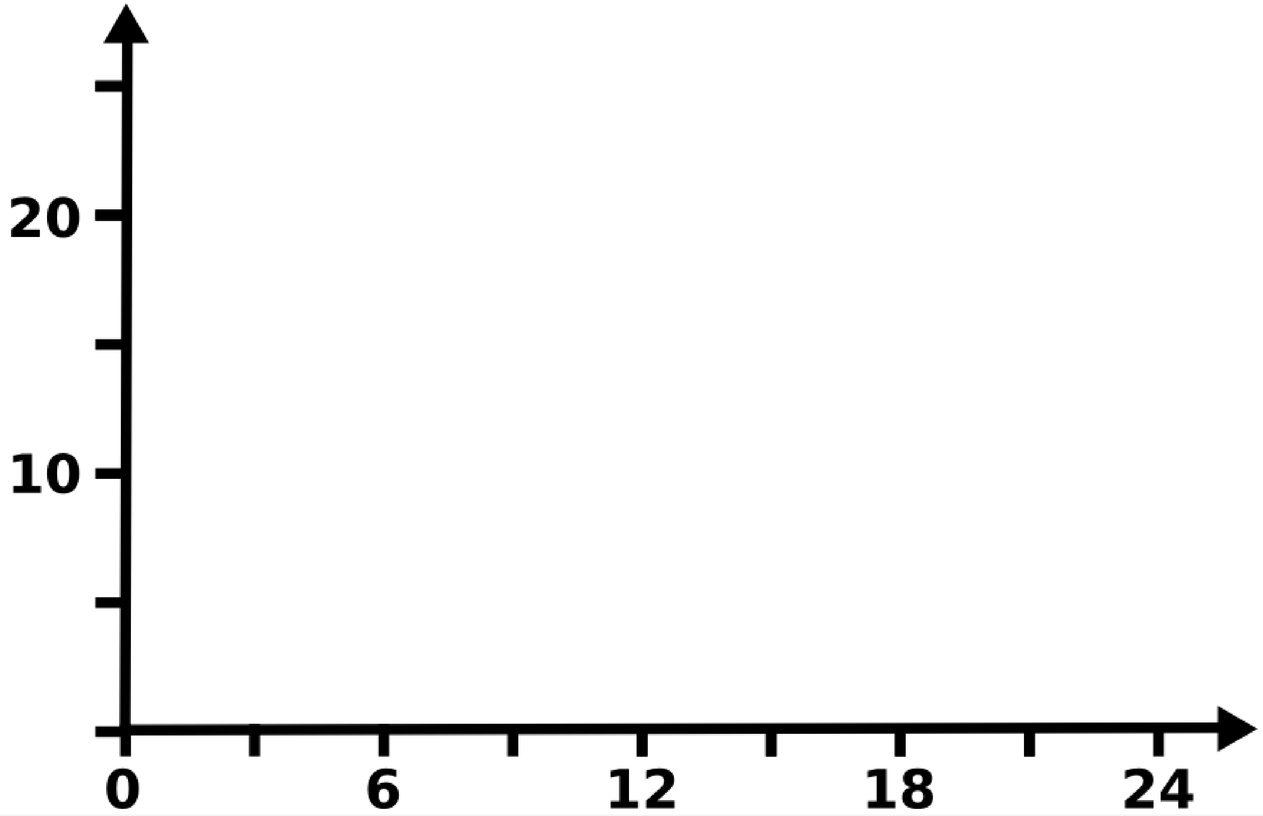
Basic idea

- Intuitive technique
  - Exploit additional dimensions for the embedding

$$f : [0, 24] \rightarrow \mathbb{R}$$

$$x = x$$

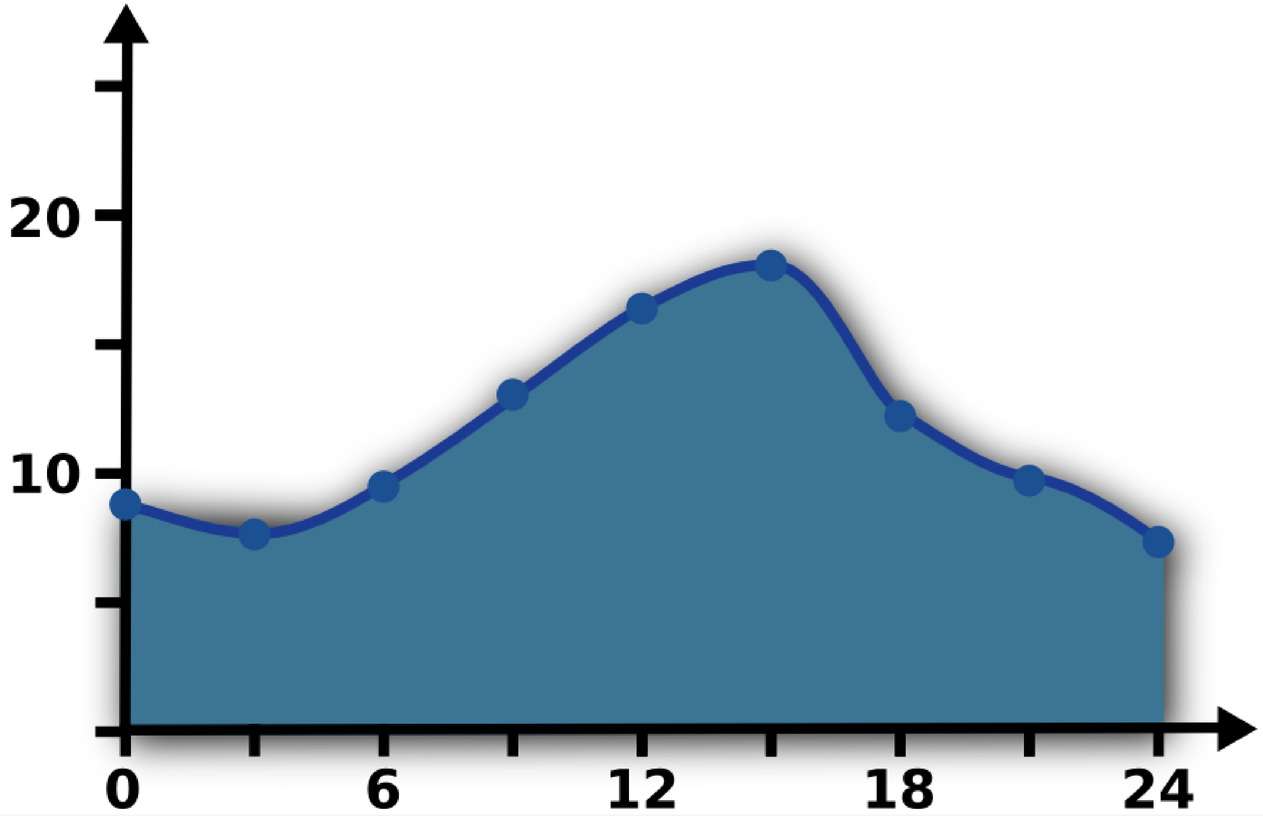
$$y = f(x)$$



Basic idea

- Intuitive technique
  - Exploit additional dimensions for the embedding

$$f : [0, 24] \rightarrow \mathbb{R}$$
$$x = x$$
$$y = f(x)$$



Basic idea

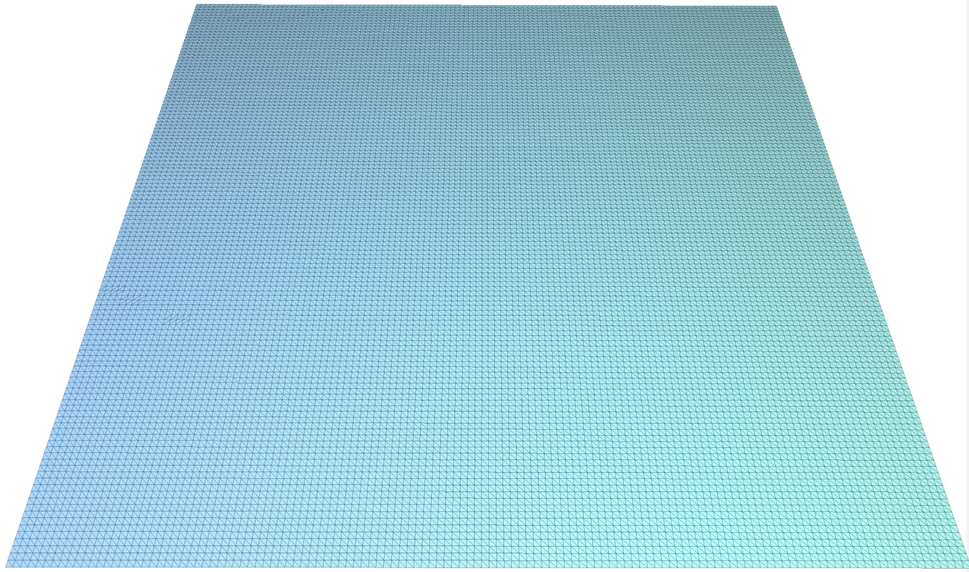
- Intuitive technique
  - Exploit additional dimensions for the embedding

$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$

Basic idea

- Intuitive technique
  - Exploit additional dimensions for the embedding

$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$



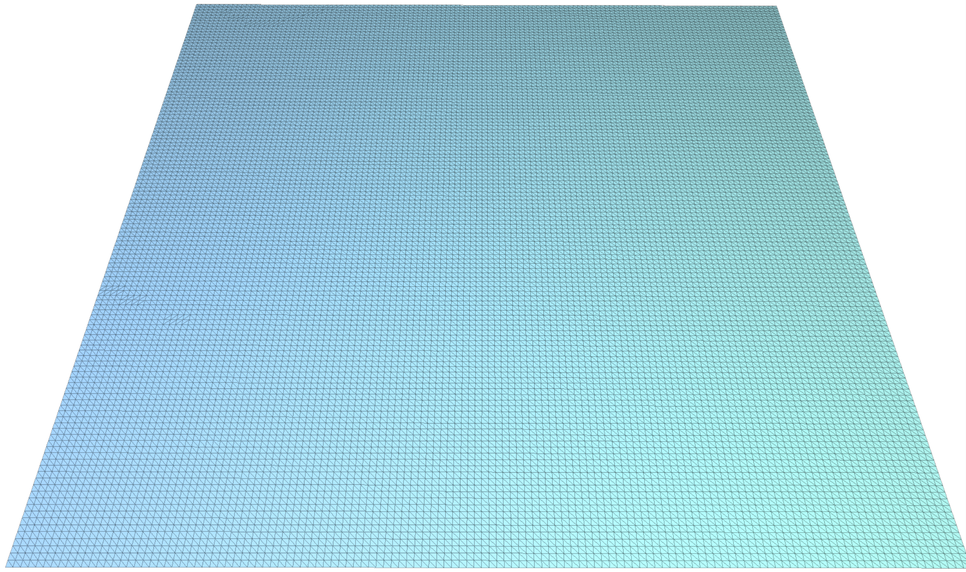
Basic idea

- Intuitive technique
  - Exploit additional dimensions for the embedding

$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$x = x$$

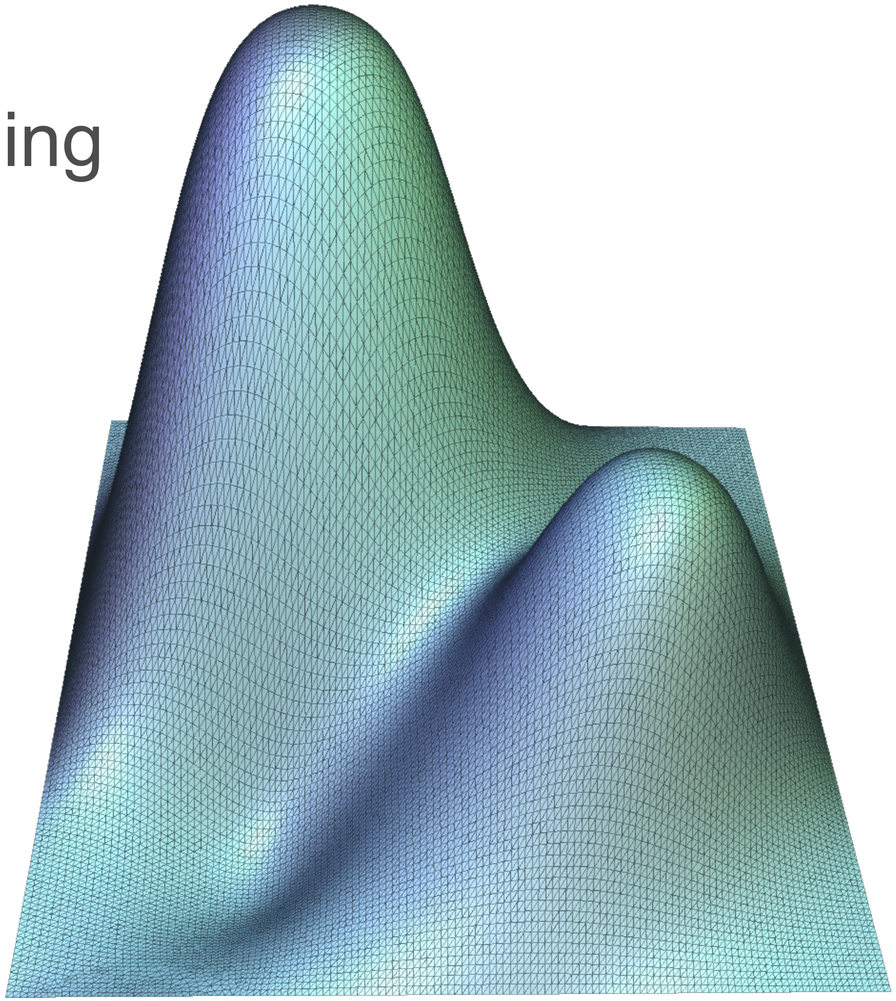
$$y = y$$



Basic idea

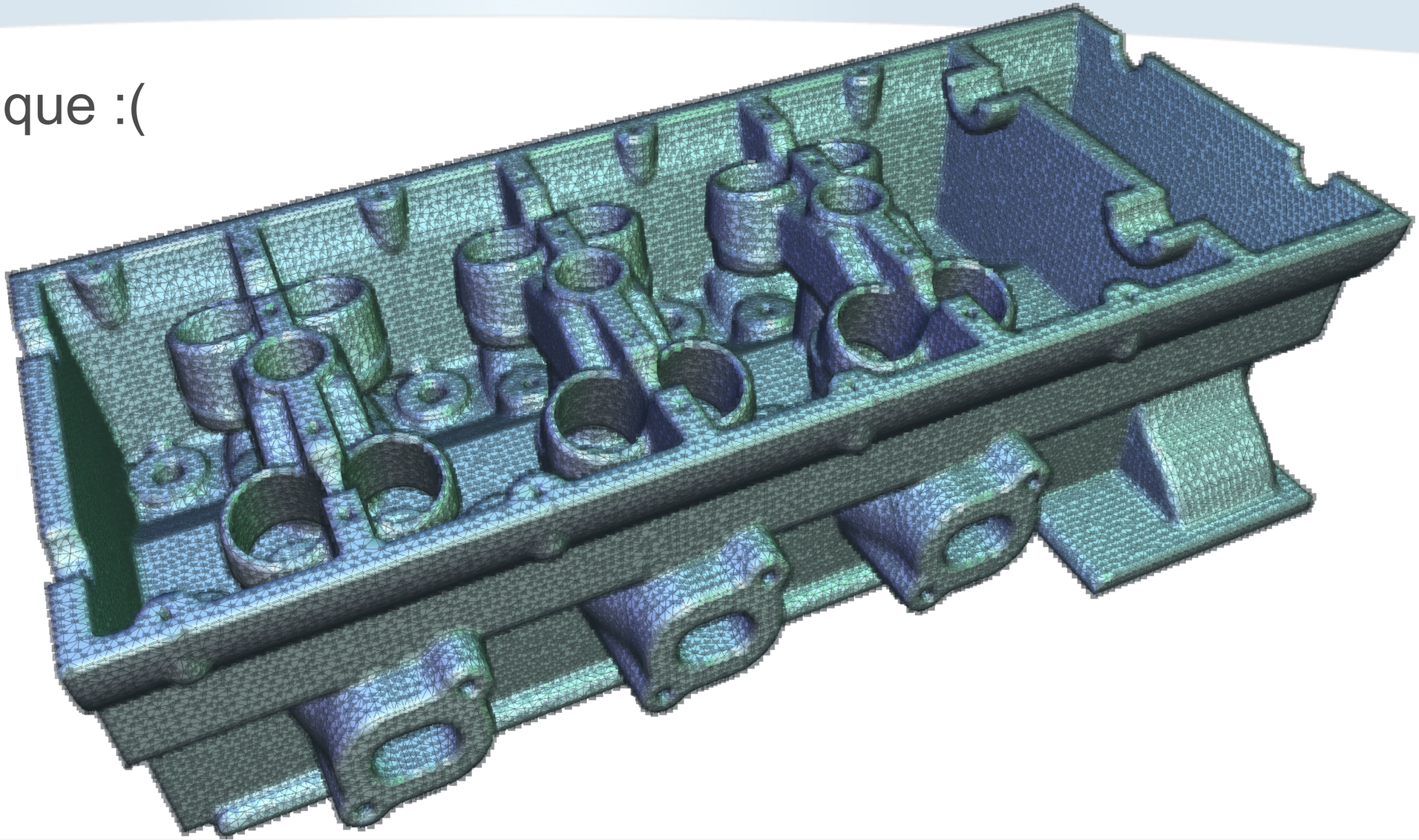
- Intuitive technique
  - Exploit additional dimensions for the embedding

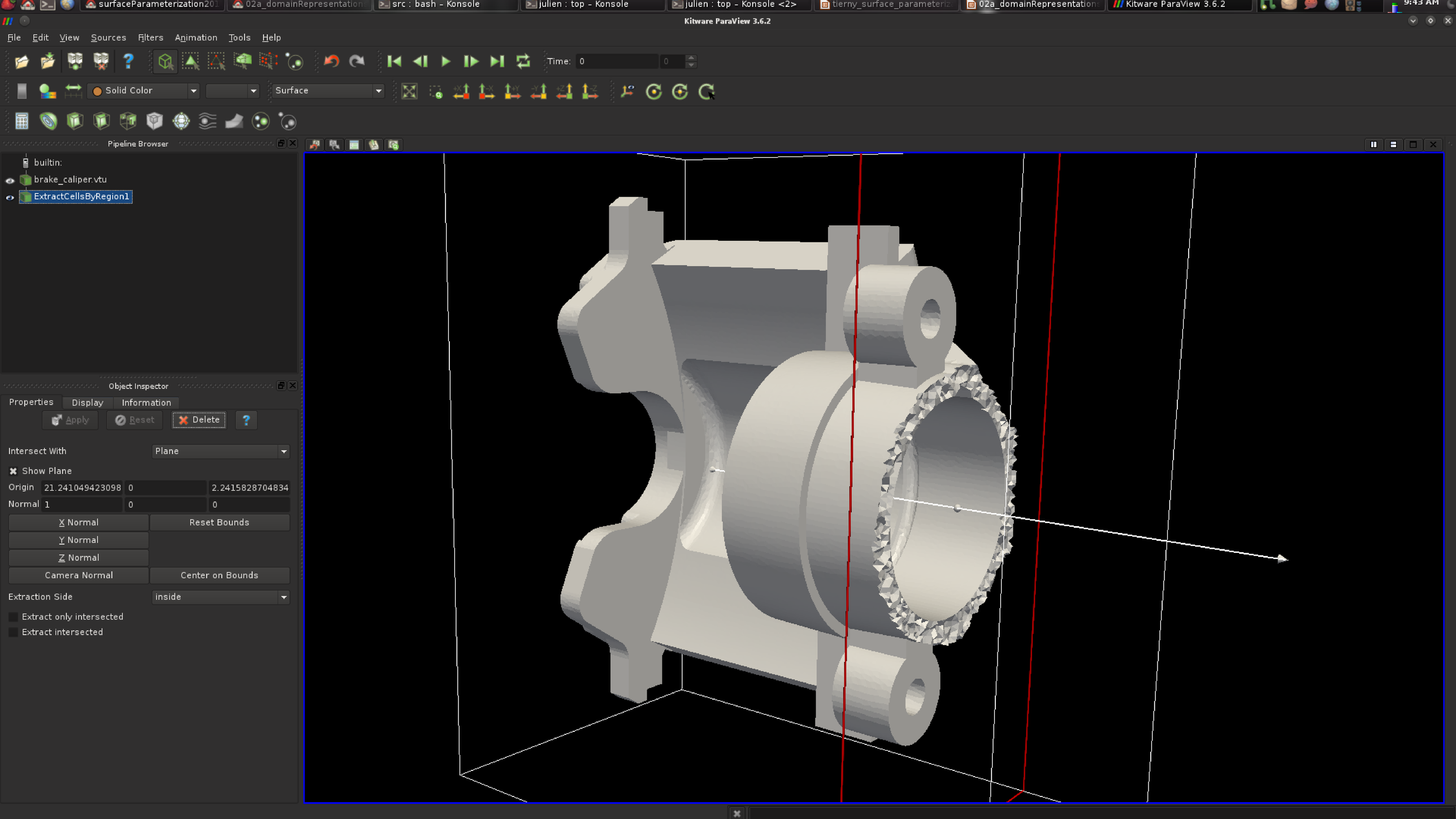
$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$
$$x = x$$
$$y = y$$
$$z = f(x, y)$$



Basic idea

- Intuitive technique :(




Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Solid Color
Surface
Pipeline Browser
builtin:
brake_caliper.vtu
ExtractCellsByRegion1
Object Inspector
Properties Display Information
Apply Reset Delete
Intersect With Plane
Show Plane
Origin 21.241049423098 0 2.2415828704834
Normal 1 0 0
X Normal
Reset Bounds
Y Normal
Z Normal
Camera Normal
Center on Bounds
Extraction Side inside
Extract only intersected
Extract intersected

Additional information channels

- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?

Additional information channels

- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?
- Intuitive idea
  - Use a perceptual channel

Additional information channels

- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?
- Intuitive idea
  - Use a perceptual channel **Color maps**

Additional information channels

- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?
- Intuitive idea
  - Use a perceptual channel **Color maps**
    - Arrange continuously a palette of colors

Additional information channels

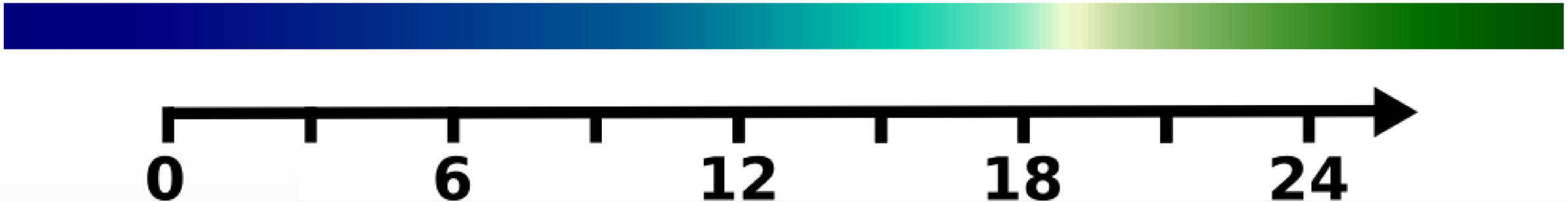
- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?
- Intuitive idea
  - Use a perceptual channel **Color maps**
    - Arrange continuously a palette of colors

Additional information channels

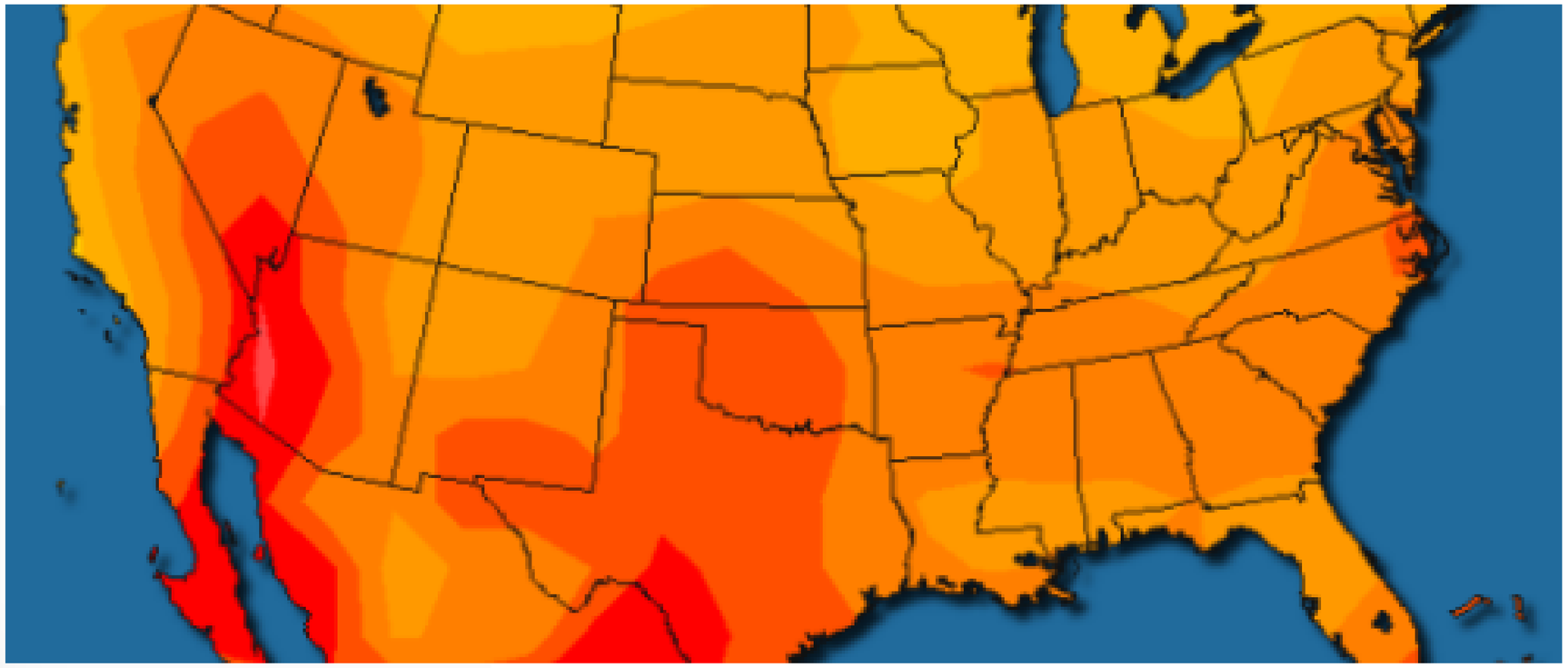
- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?
- Intuitive idea
  - Use a perceptual channel **Color maps**
    - Arrange continuously a palette of colors
    - Map it to the real line $(\mathcal{A})$

Additional information channels

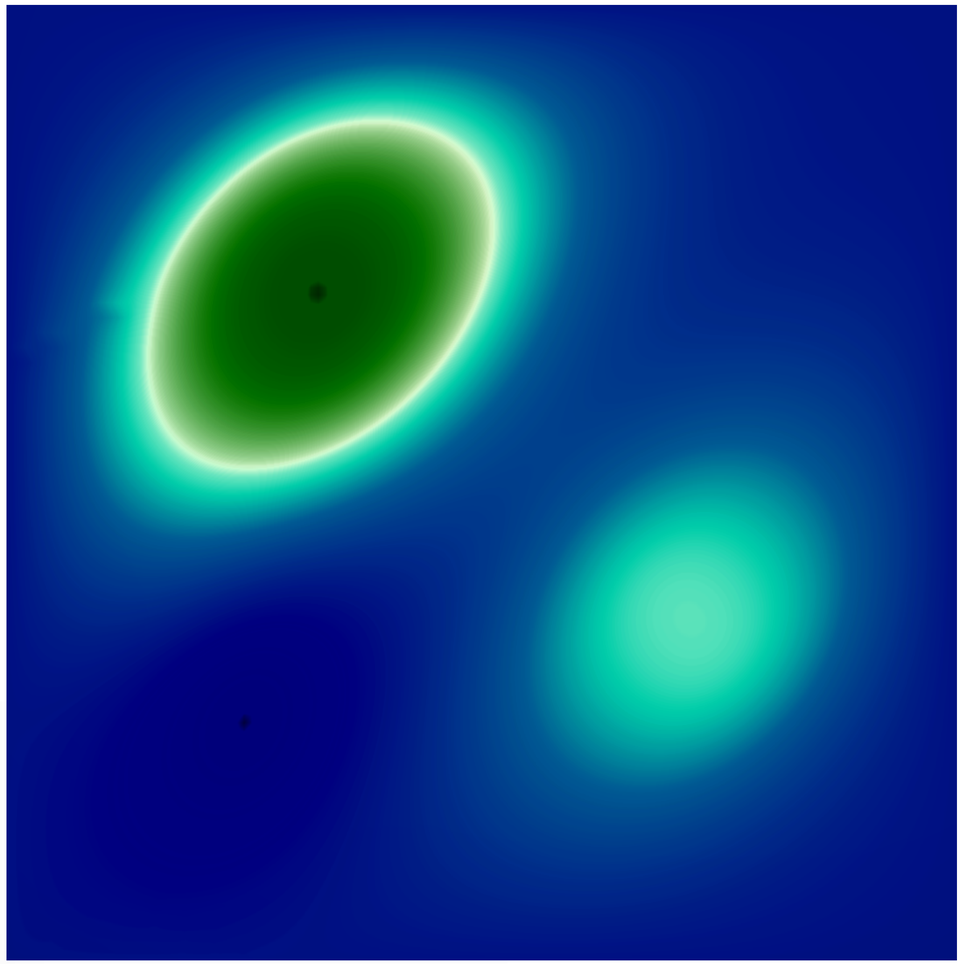
- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?
- Intuitive idea
  - Use a perceptual channel **Color maps**
    - Arrange continuously a palette of colors
    - Map it to the real line $(\mathcal{A})$



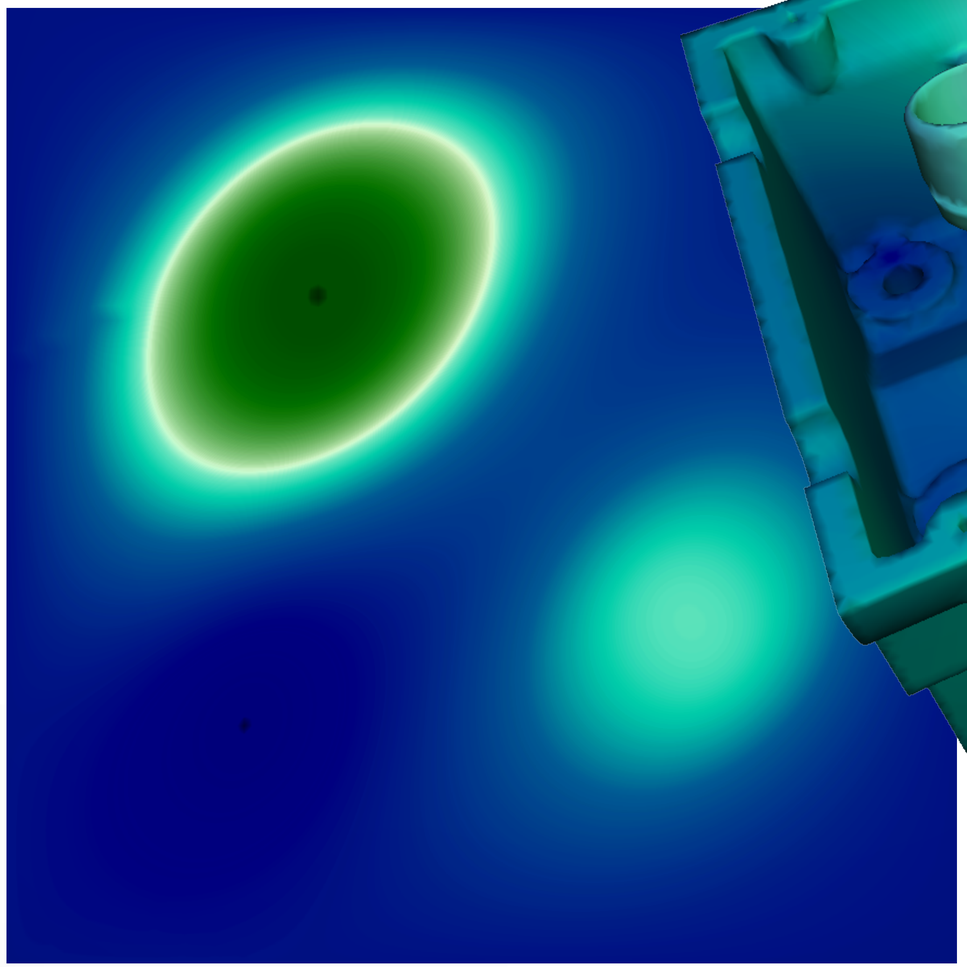
Color maps

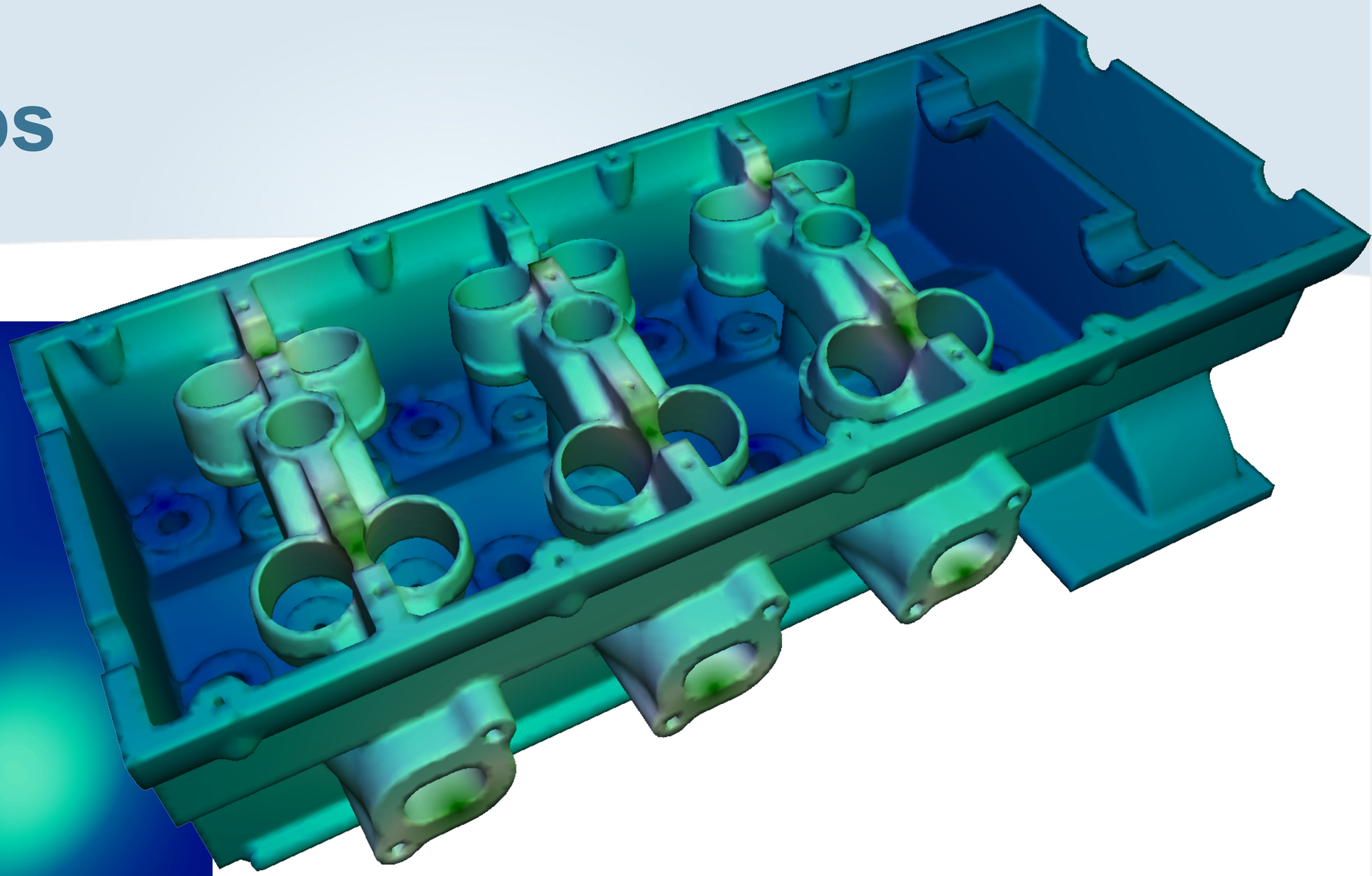


Color maps



Color maps



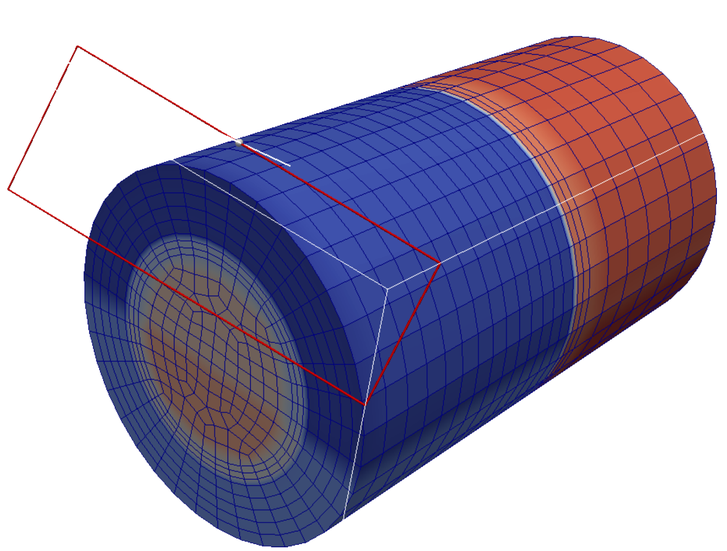


Color maps

- Intuitive idea :(
  - What about volumetric domains?

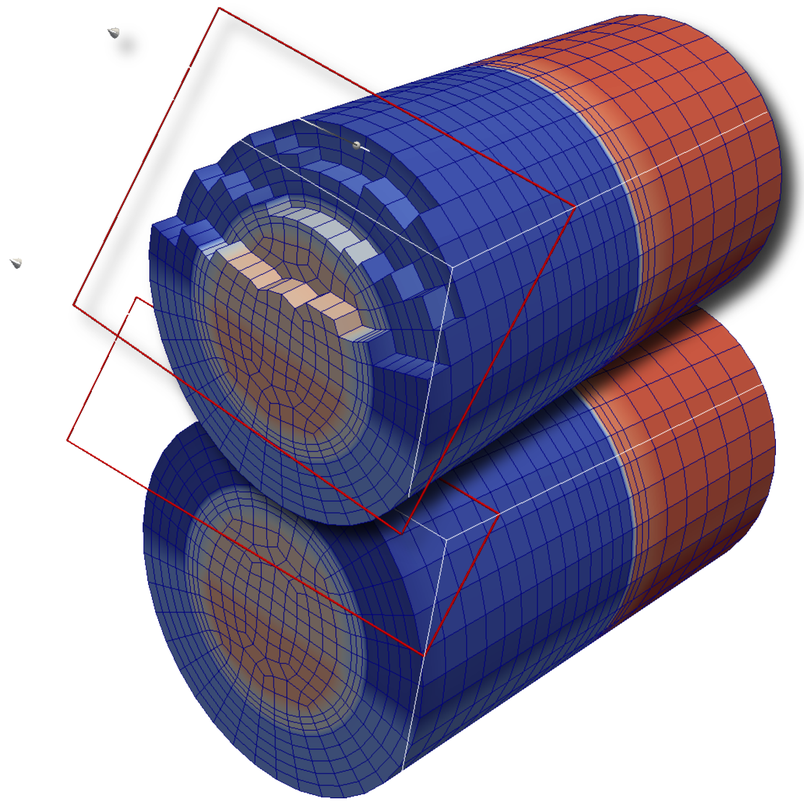
Color maps

- Intuitive idea :(
  - What about volumetric domains?



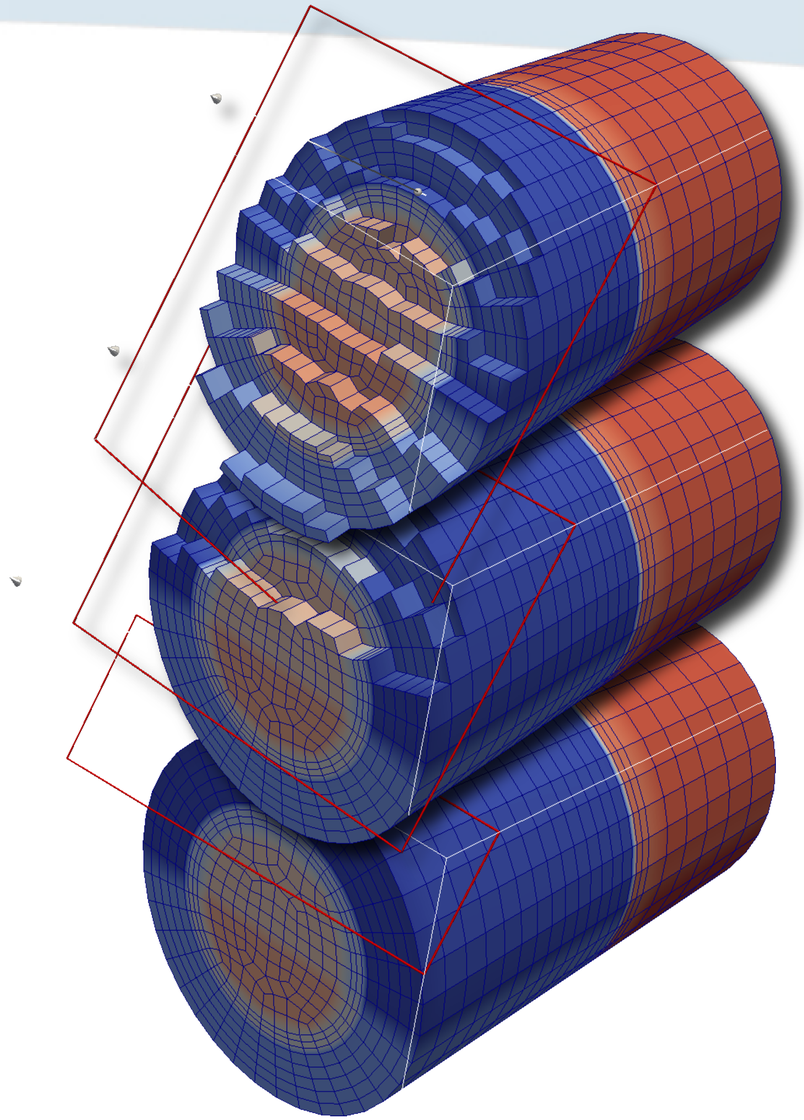
Color maps

- Intuitive idea :(
  - What about volumetric domains?



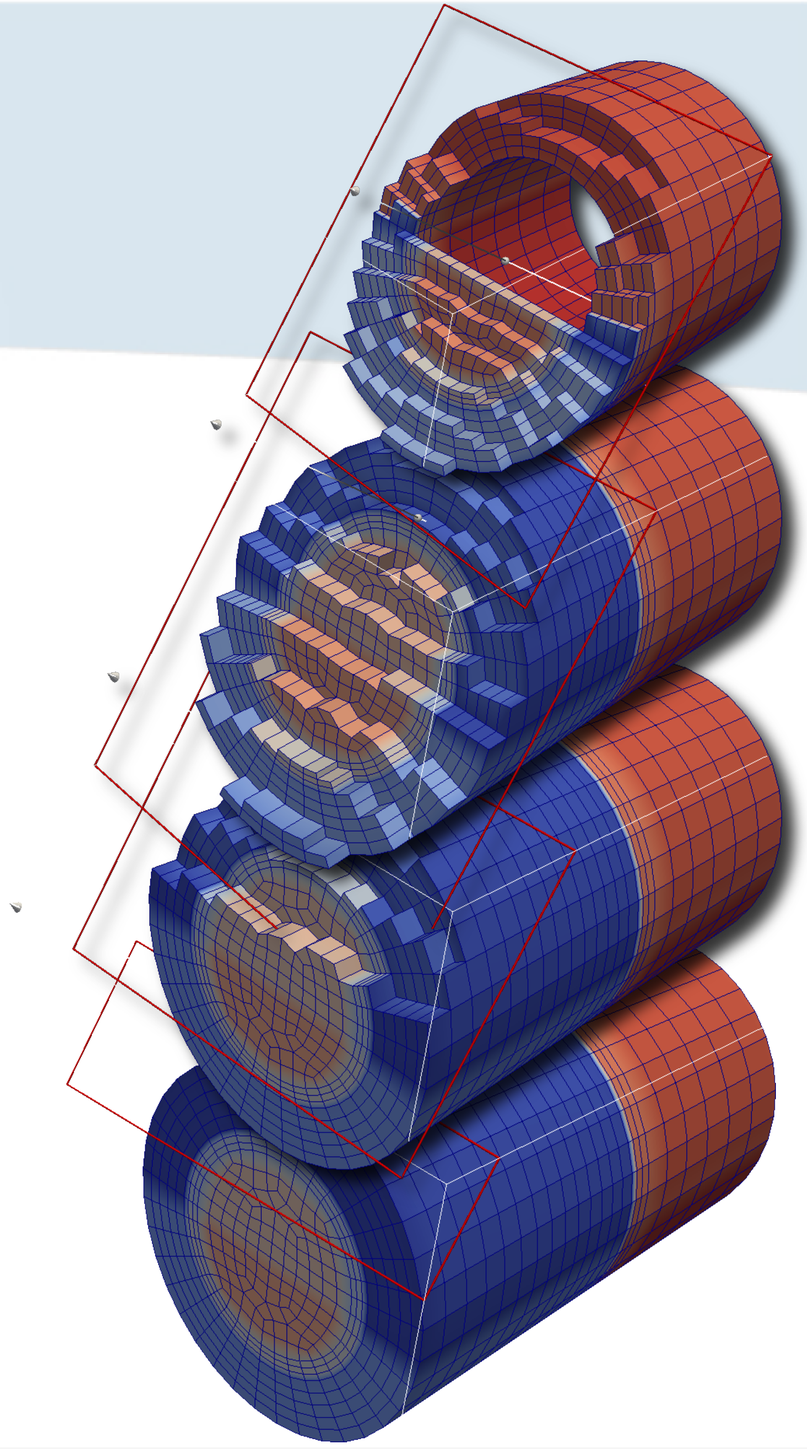
Color maps

- Intuitive idea :(
  - What about volumetric domains?



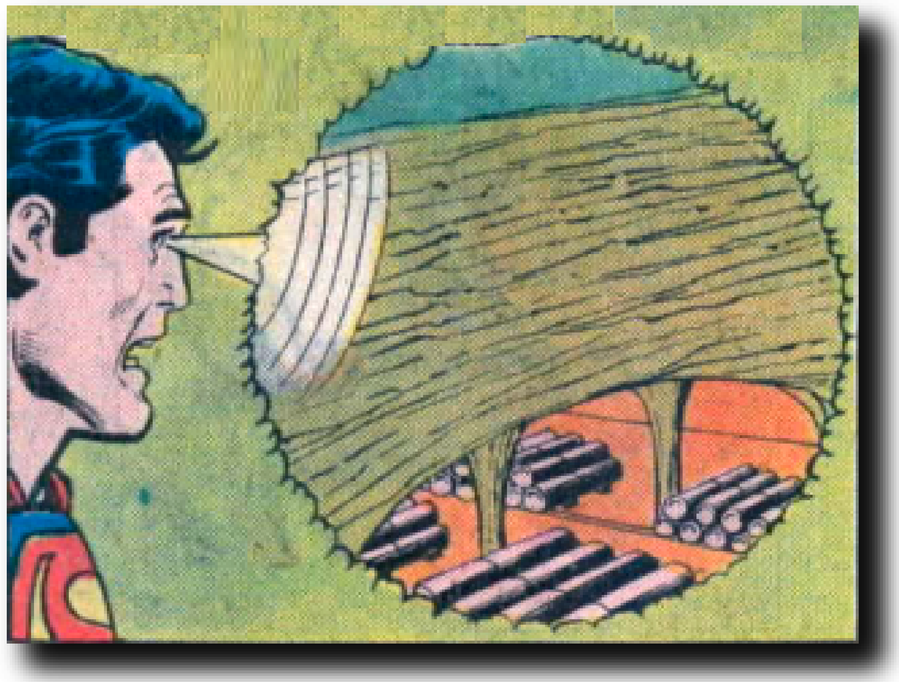
Color maps

- Intuitive idea :(
  - What about volumetric domains?



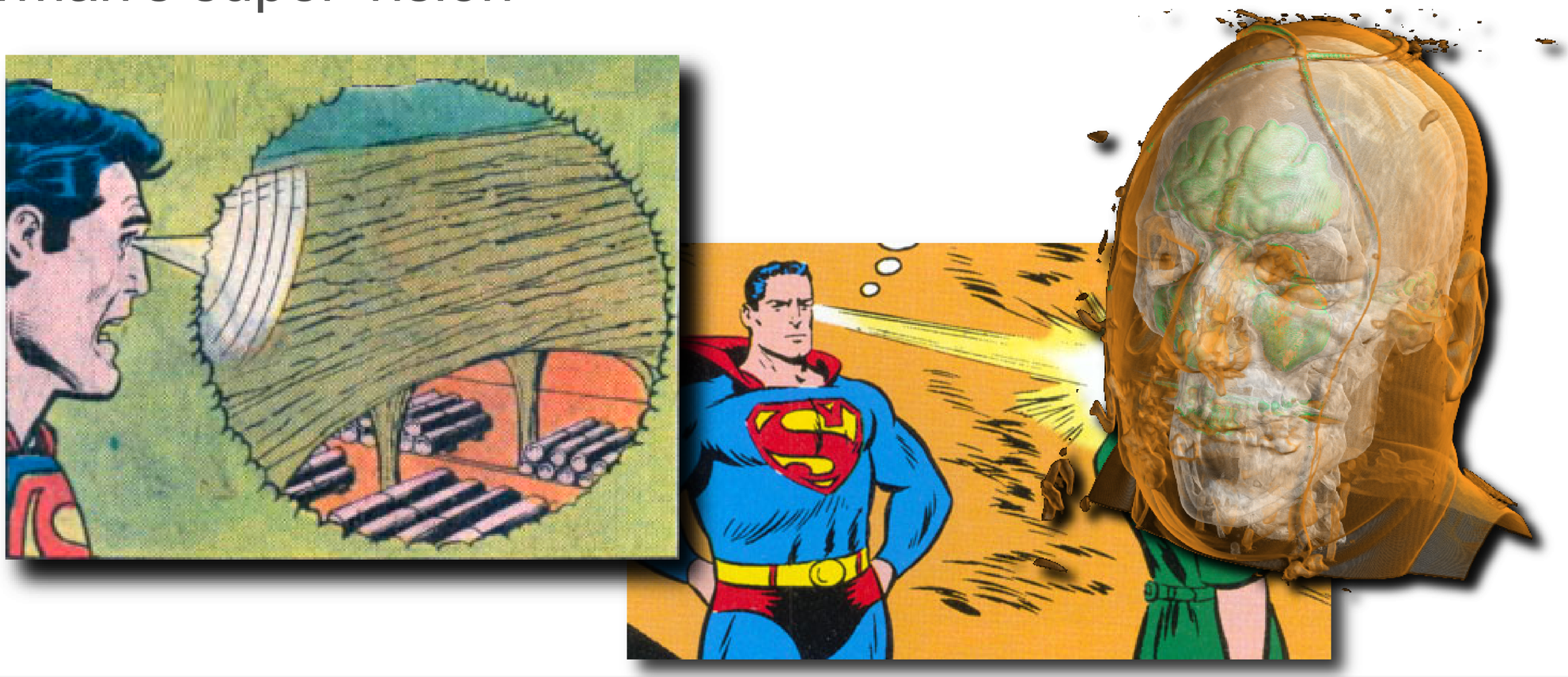
Color maps

- Intuitive idea :(
  - Mimic Superman's super-vision



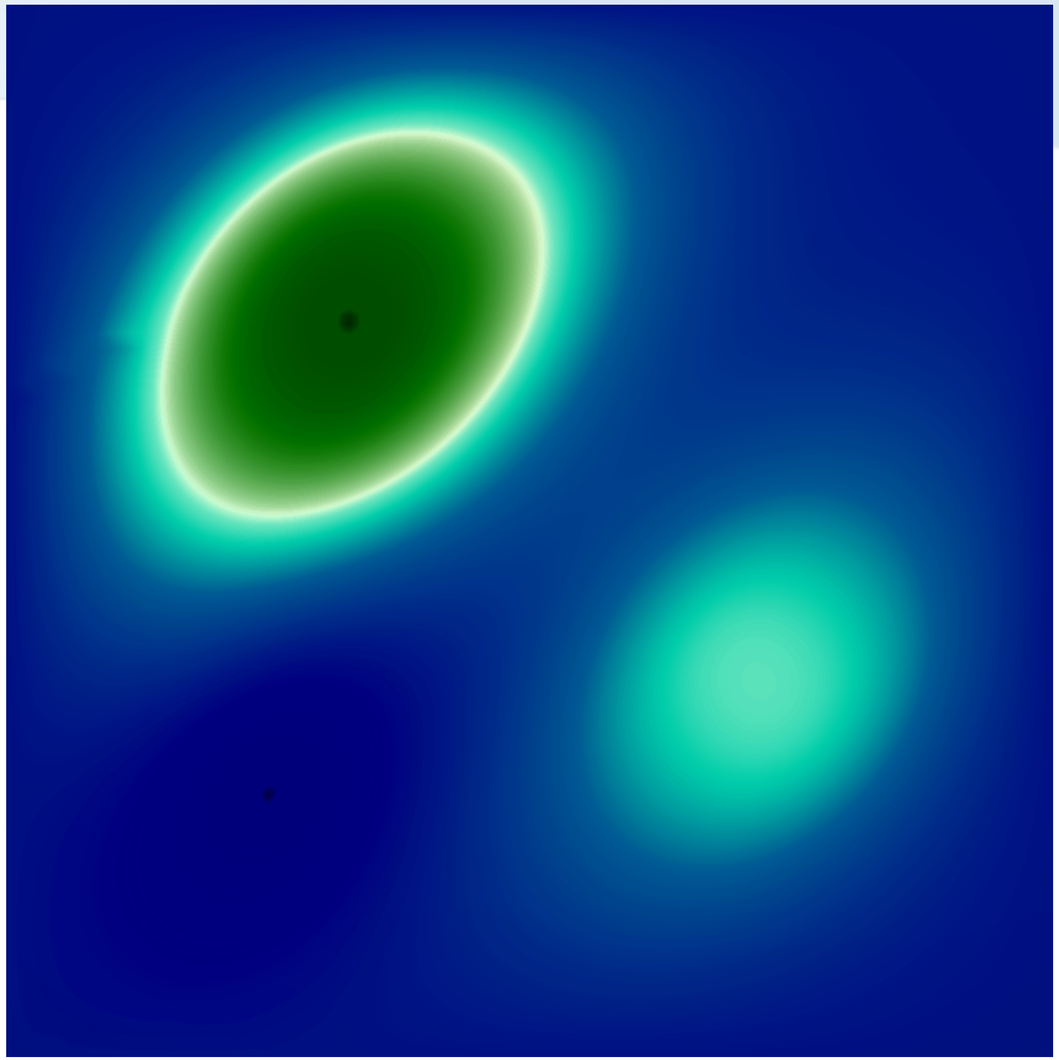
Color maps

- Intuitive idea :(
  - Mimic Superman's super-vision



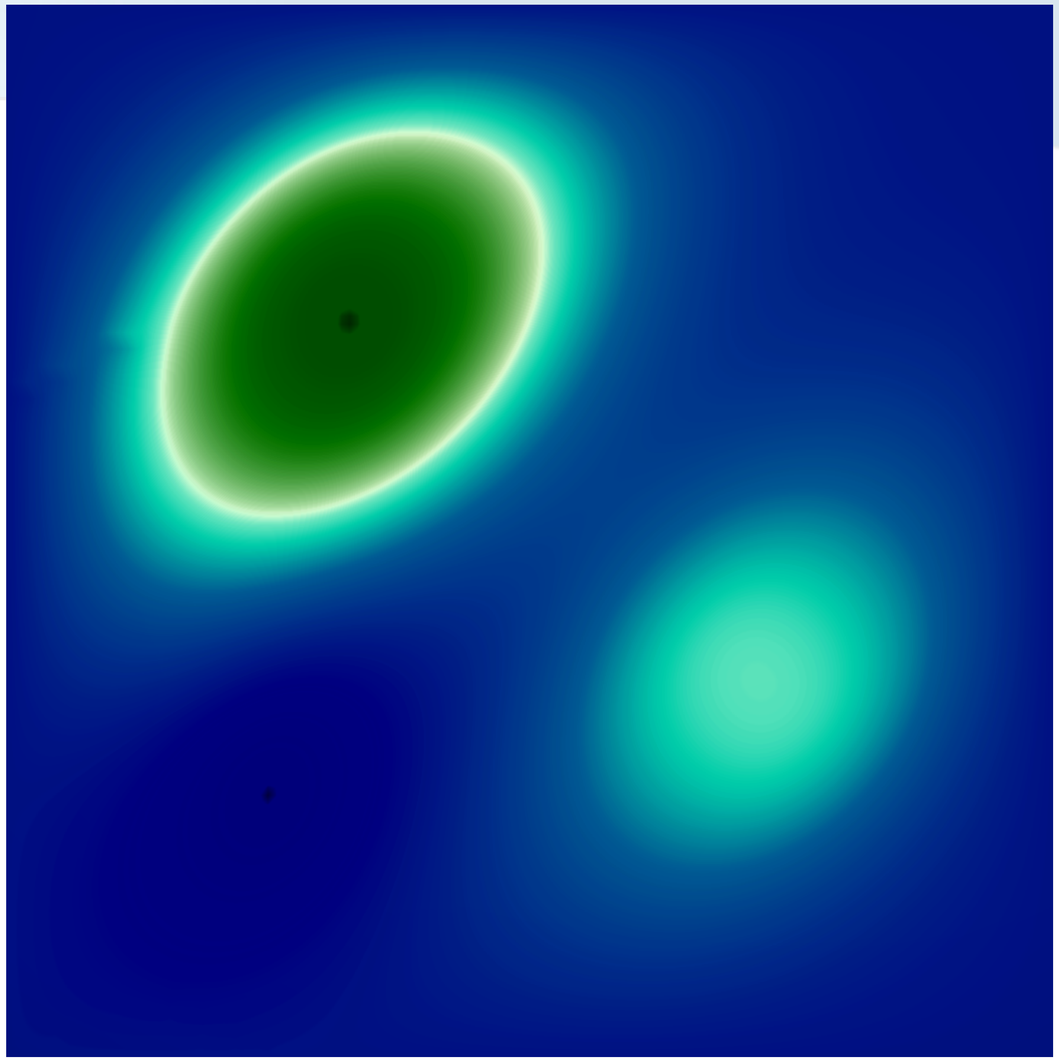
Scalar field geometry

- Continuous color maps



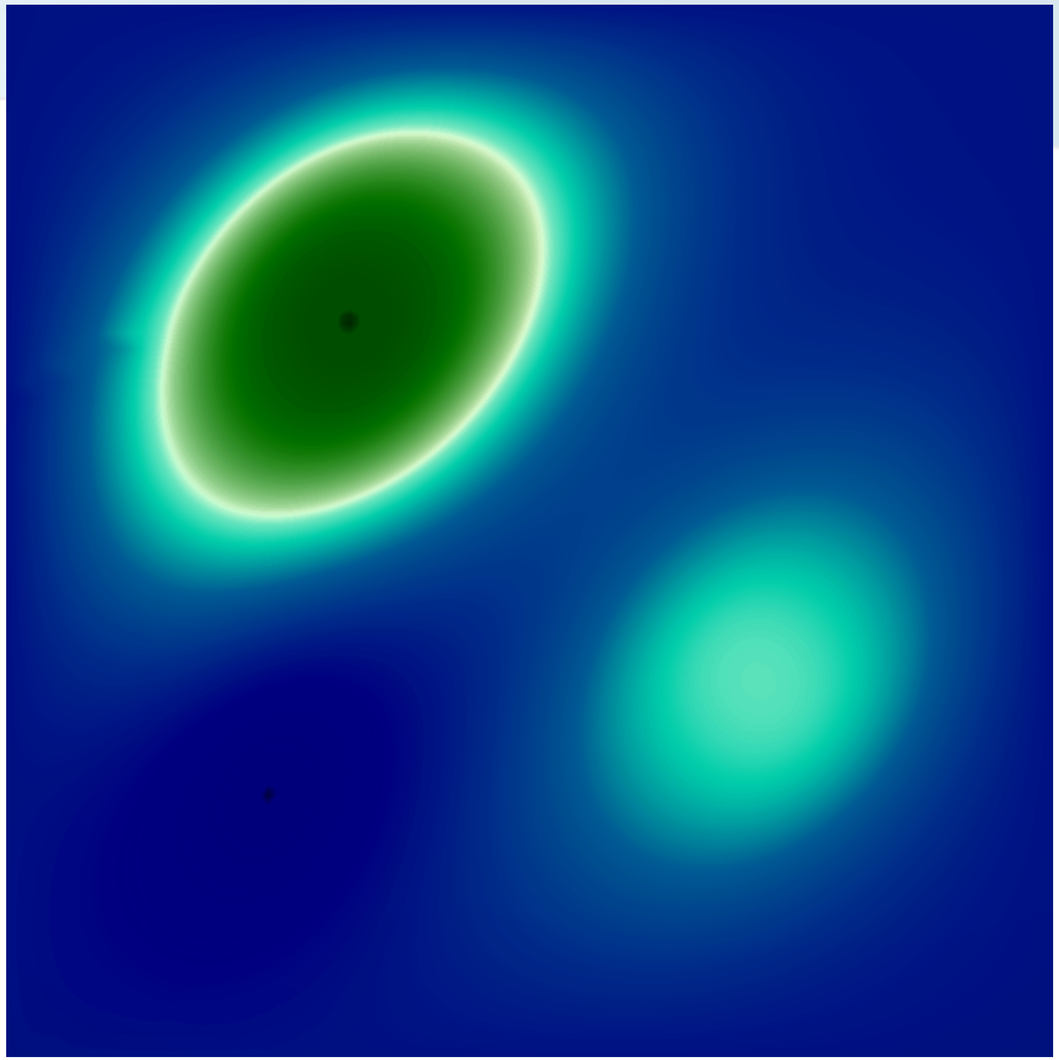
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry



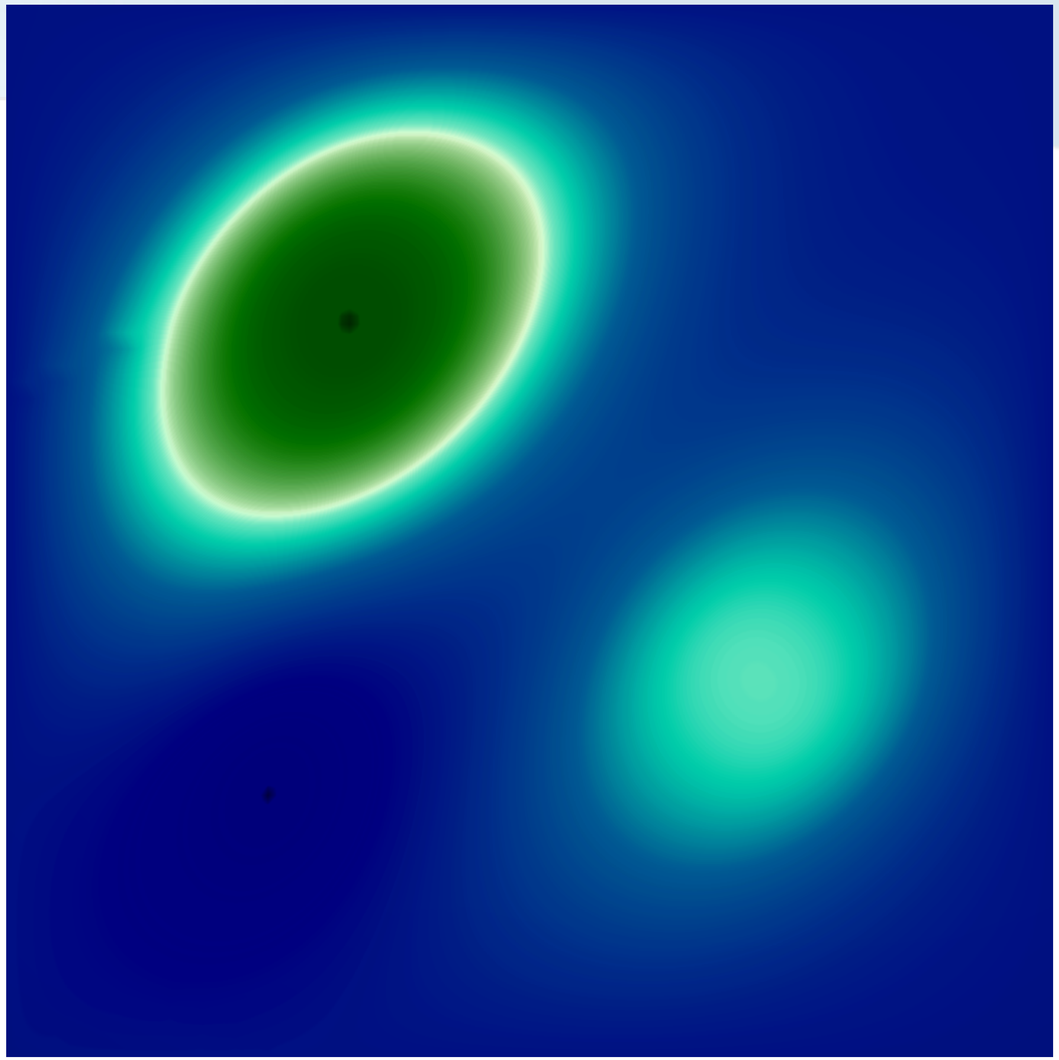
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry
    - Locally (gradient)
    - Globally (level sets)



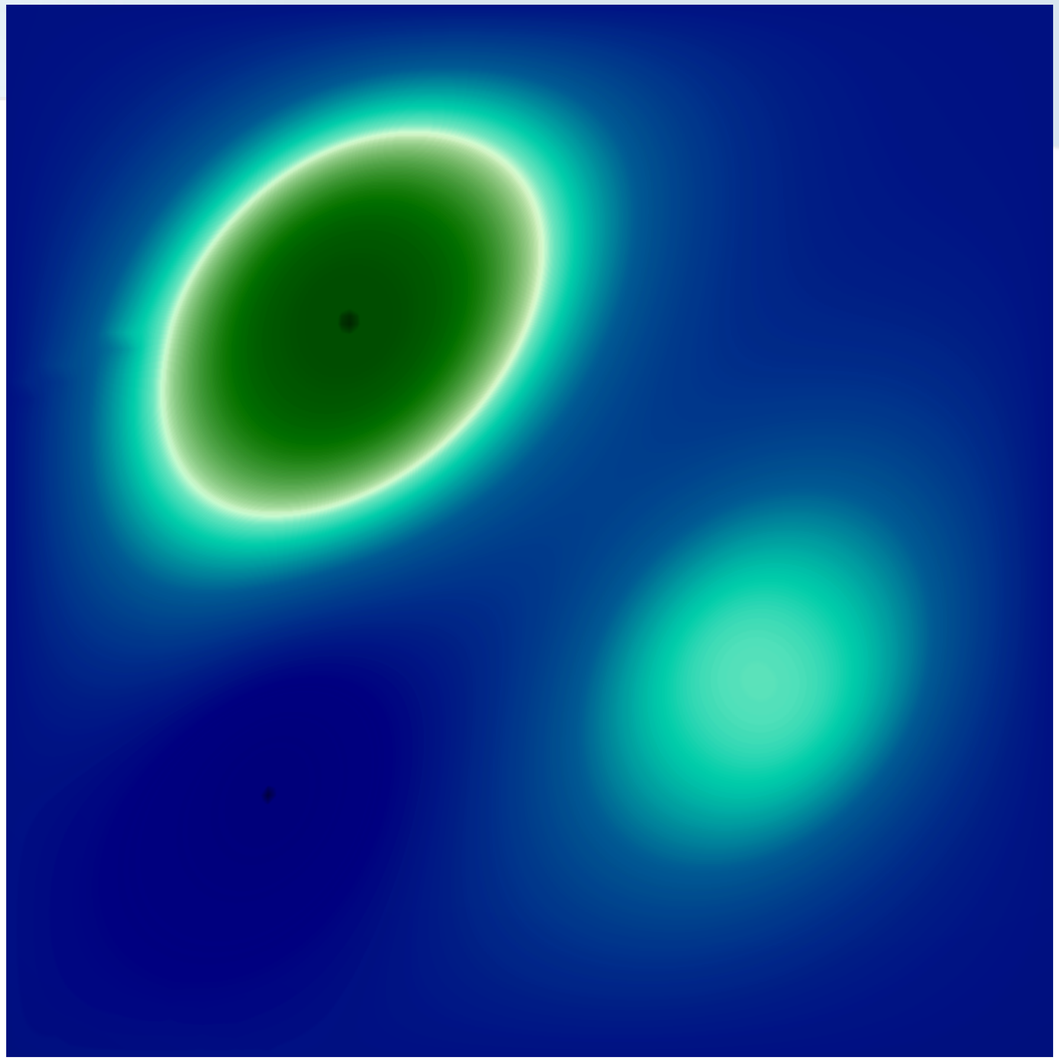
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry
    - Locally (gradient)
    - Globally (level sets)
- Level sets



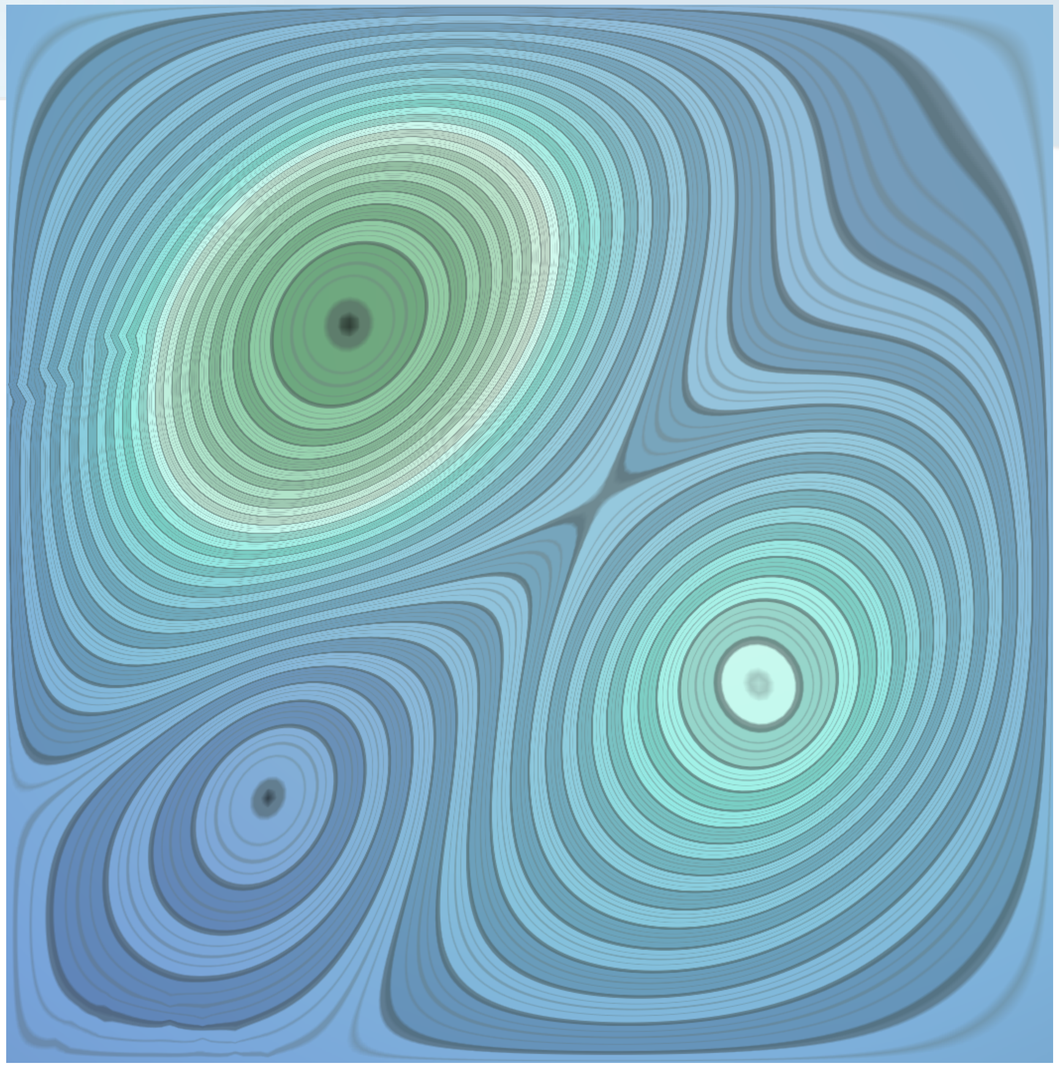
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry
    - Locally (gradient)
    - Globally (level sets)
- Level sets
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$



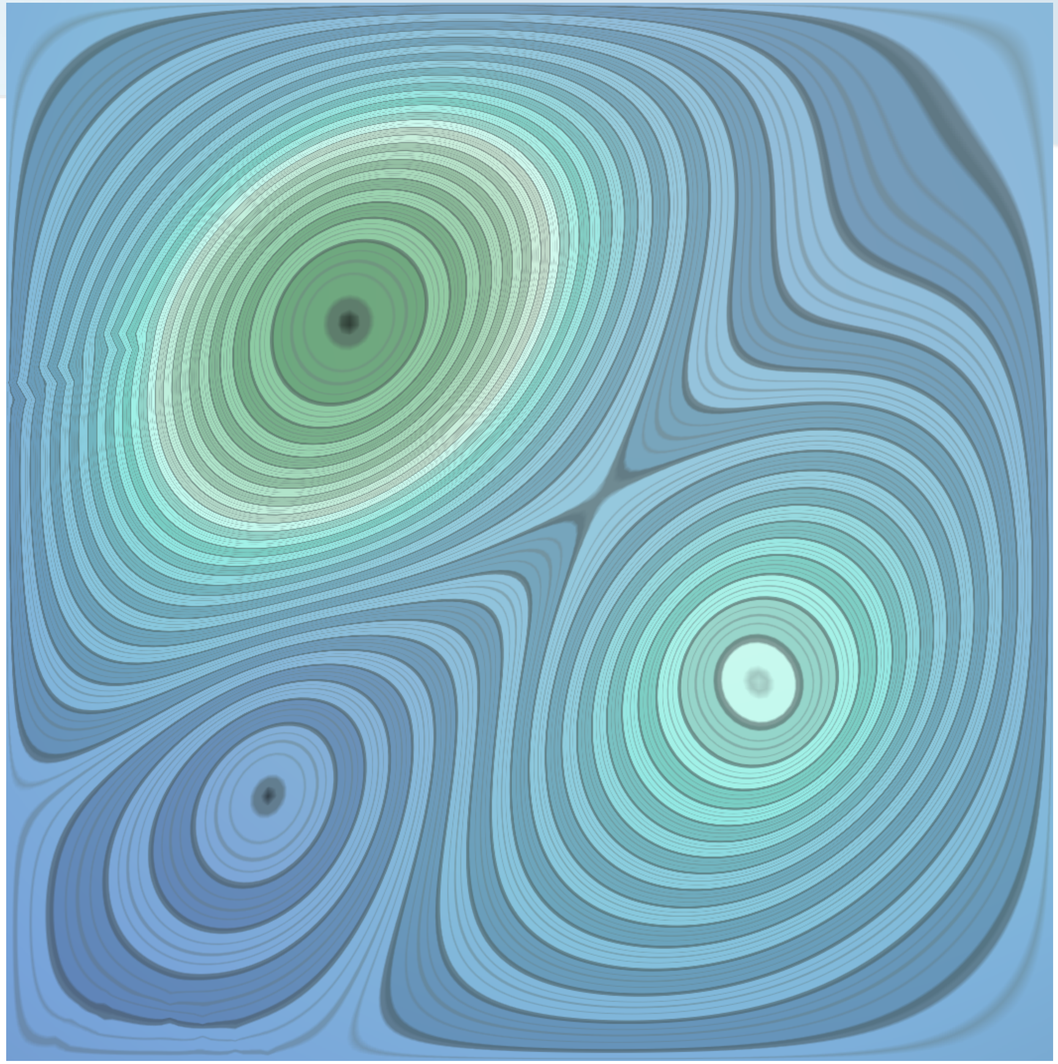
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry
    - Locally (gradient)
    - Globally (level sets)
- Level sets
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$



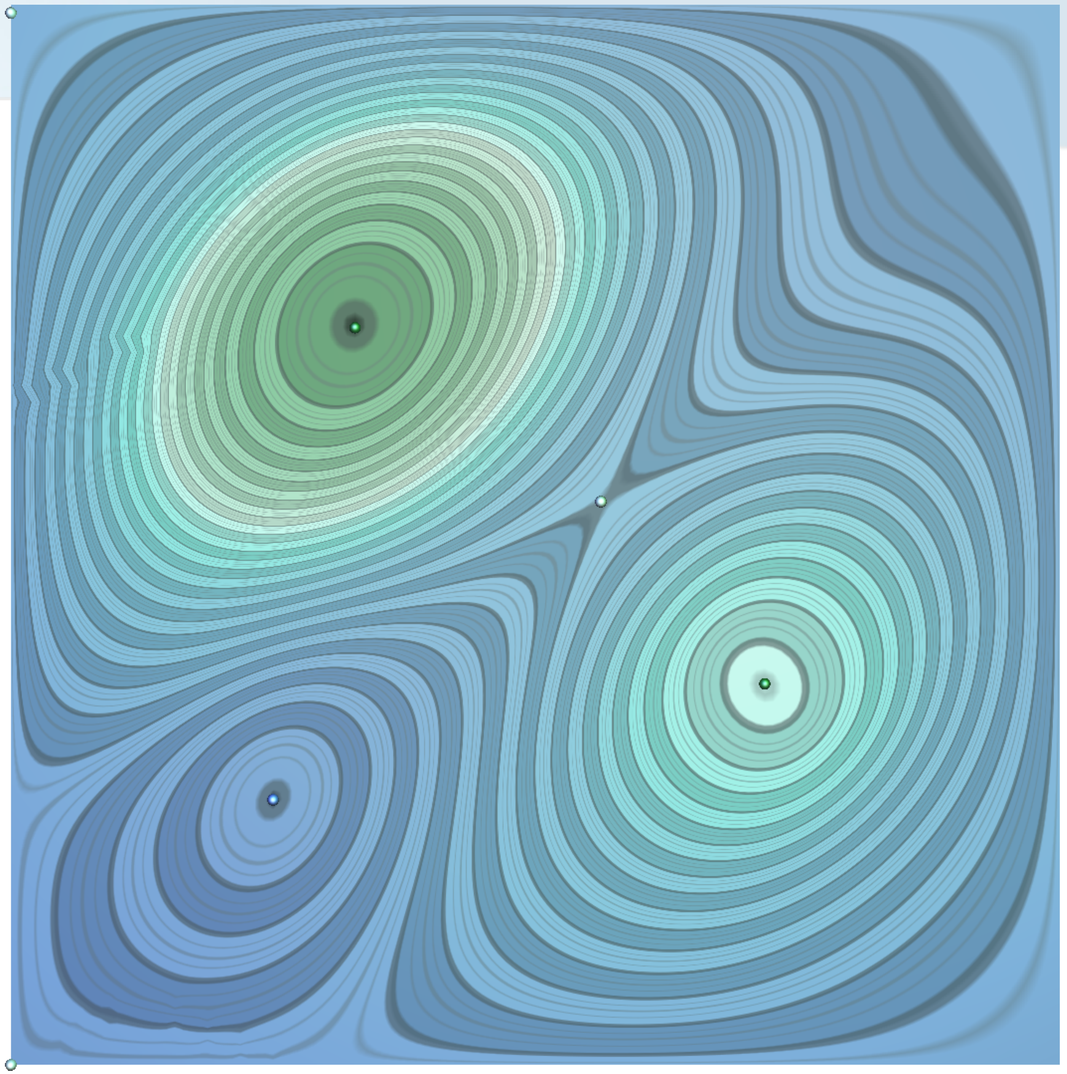
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry
    - Locally (gradient)
    - Globally (level sets)
- Level sets
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
  - Critical points
    - Where ∇f vanishes



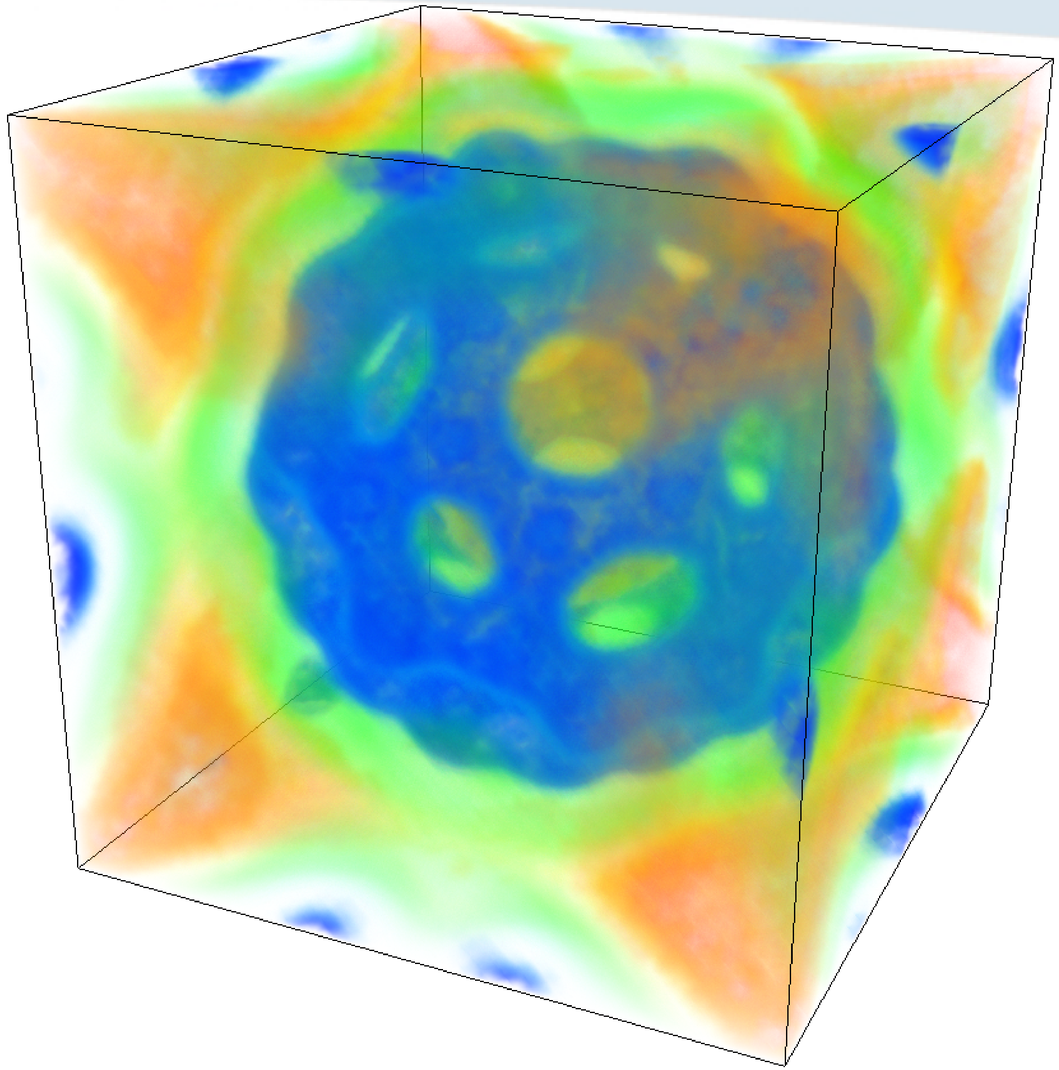
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry
    - Locally (gradient)
    - Globally (level sets)
- Level sets
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
  - Critical points
    - Where ∇f vanishes



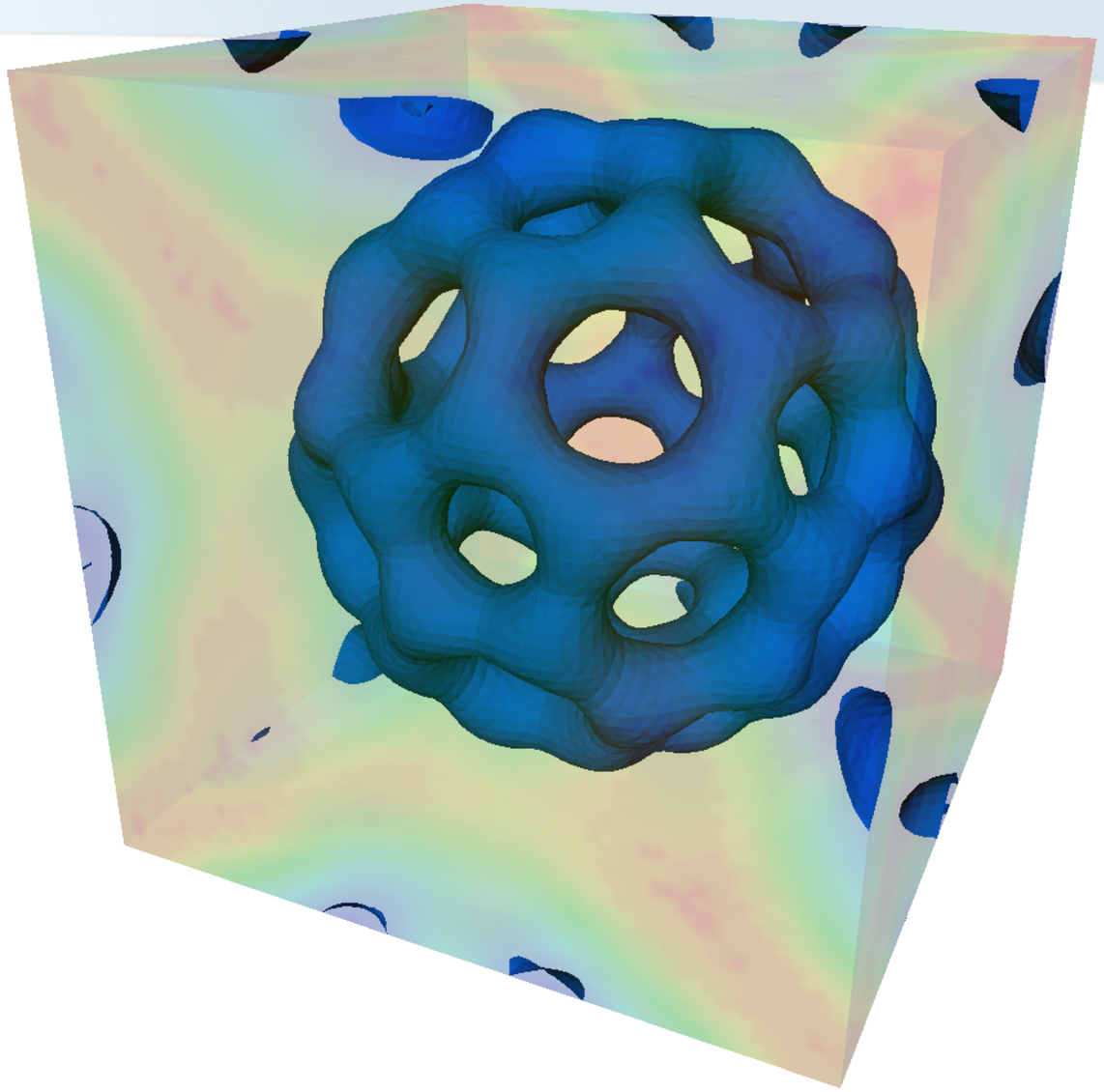
Query-driven visualization

- Visualize isolated regions of interests



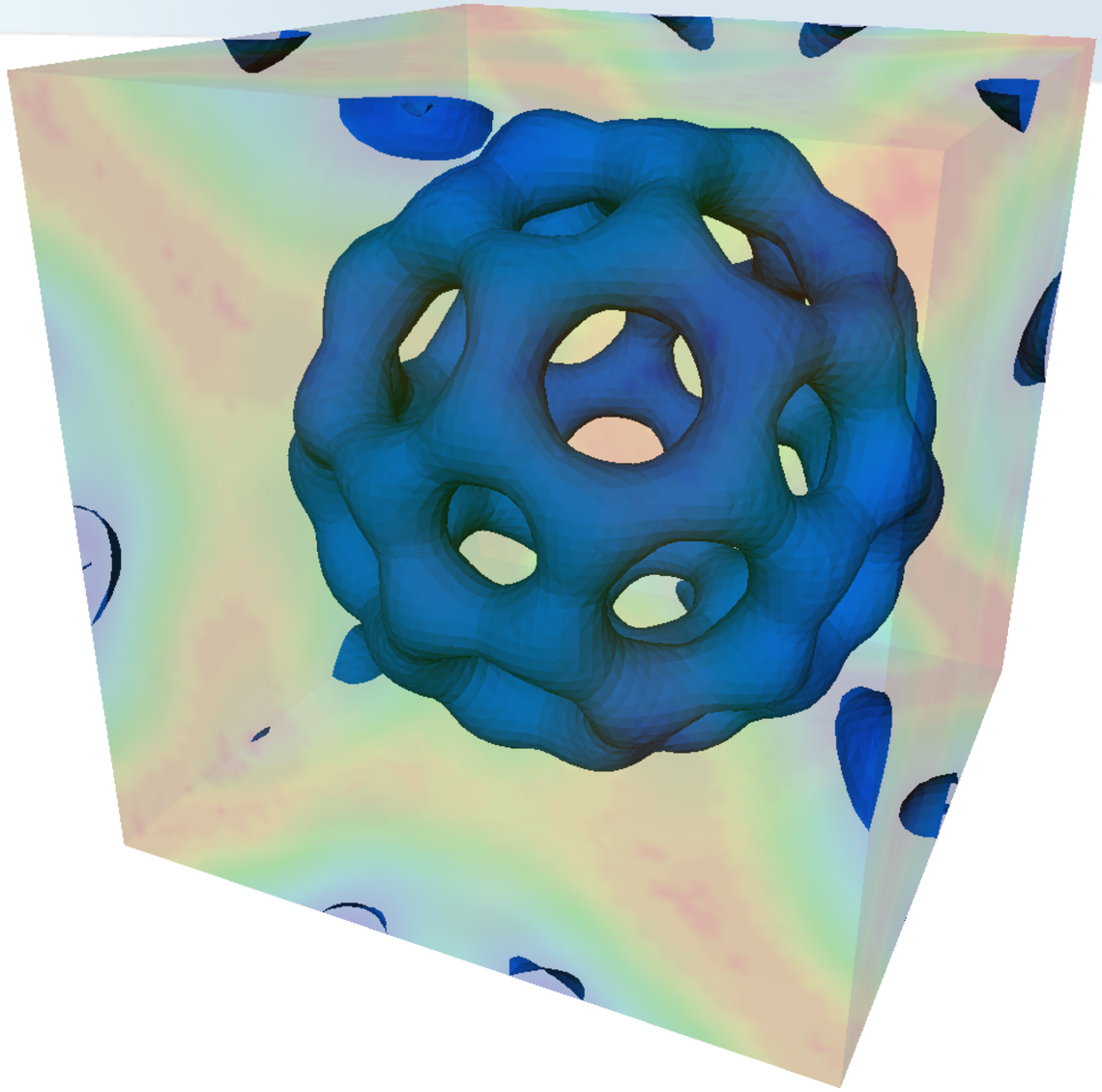
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction



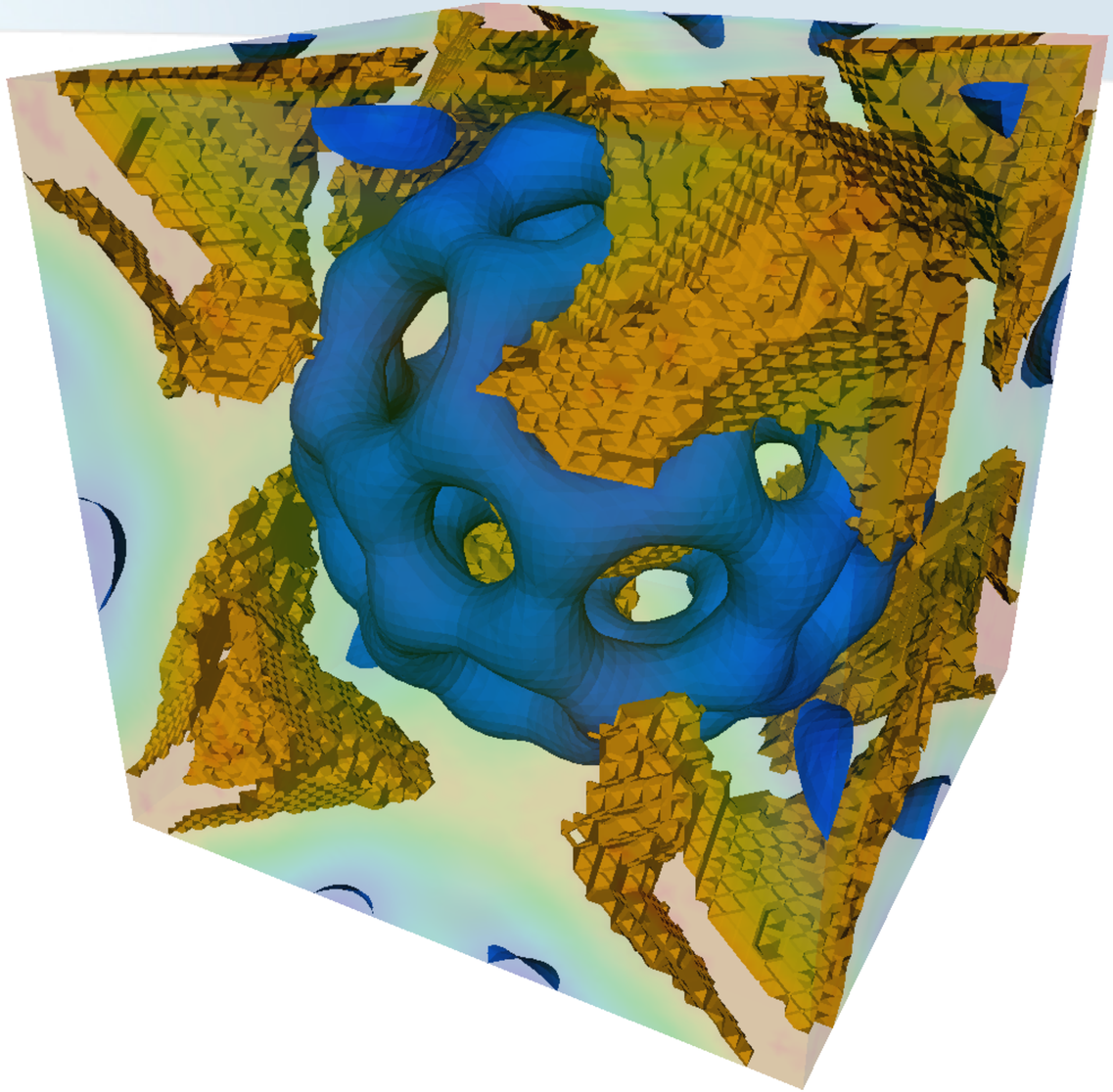
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization



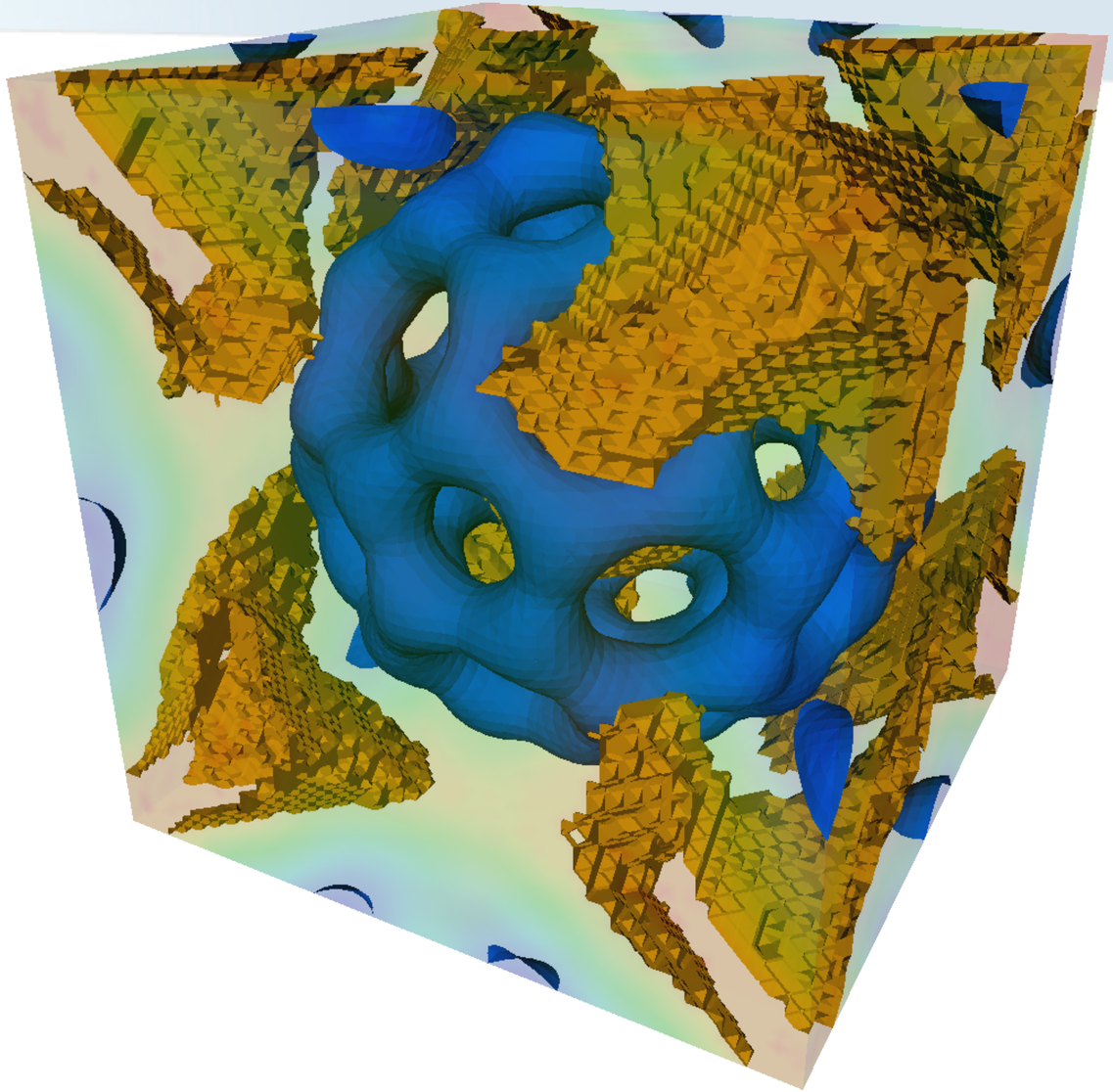
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization



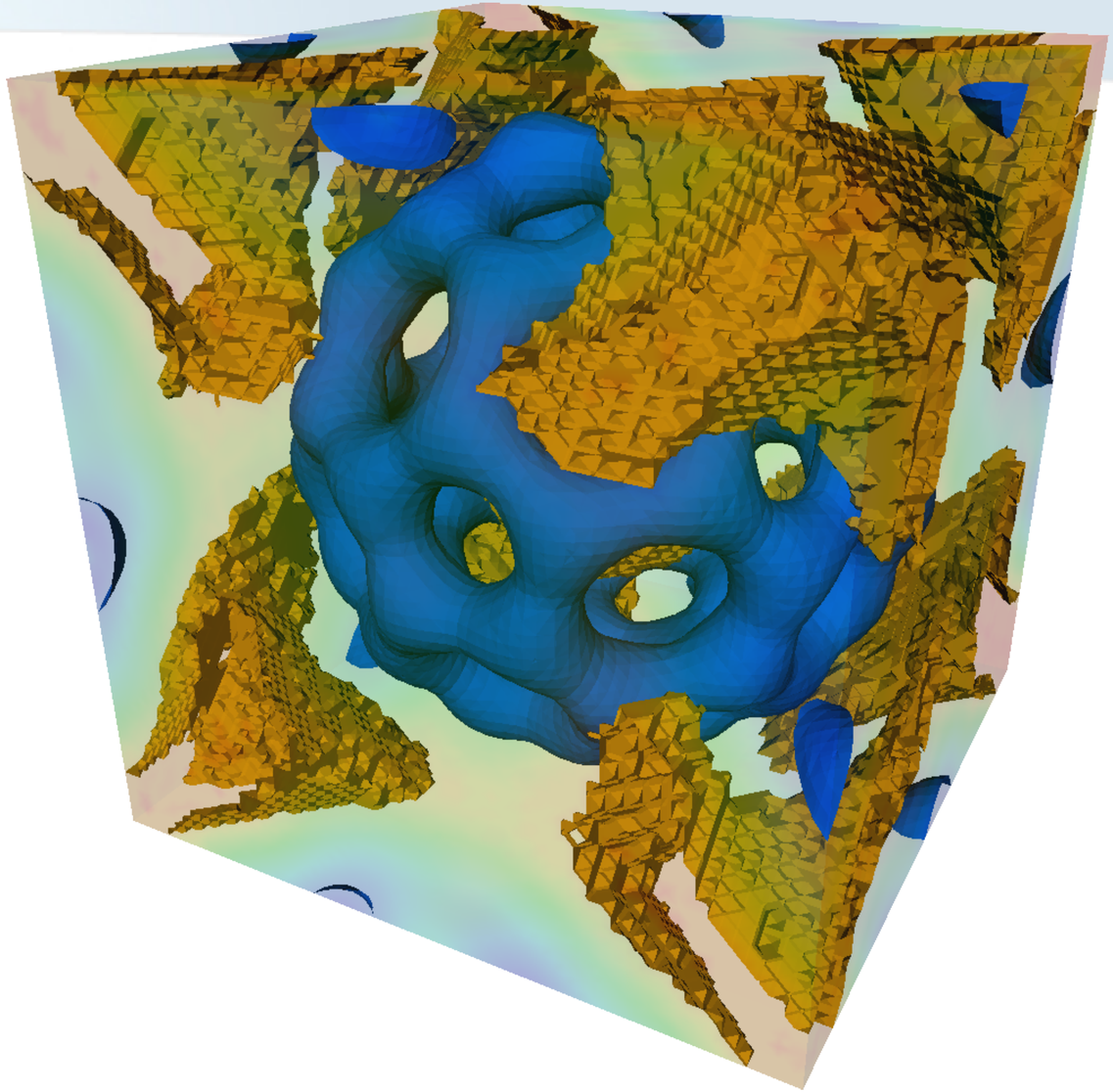
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



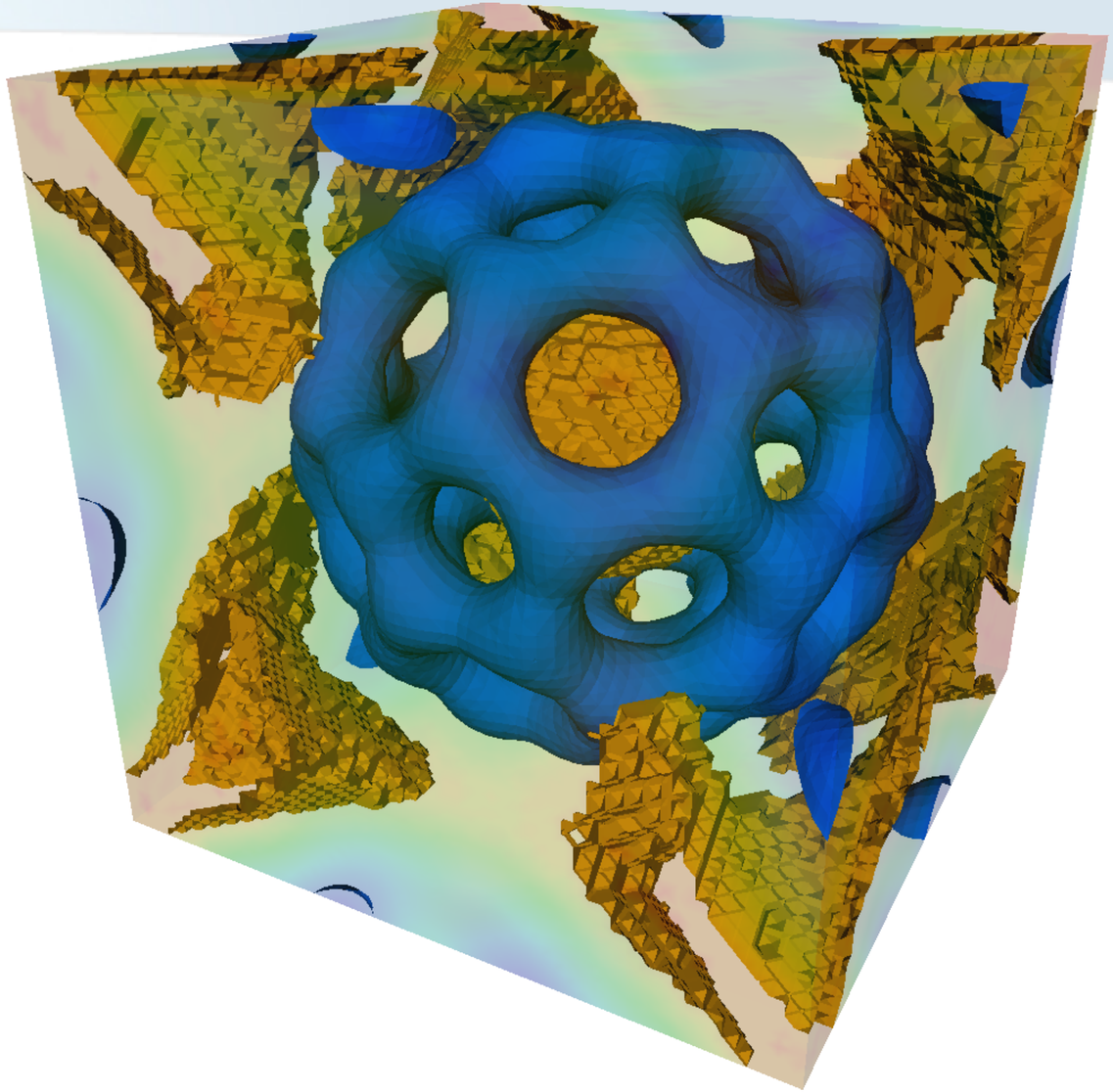
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



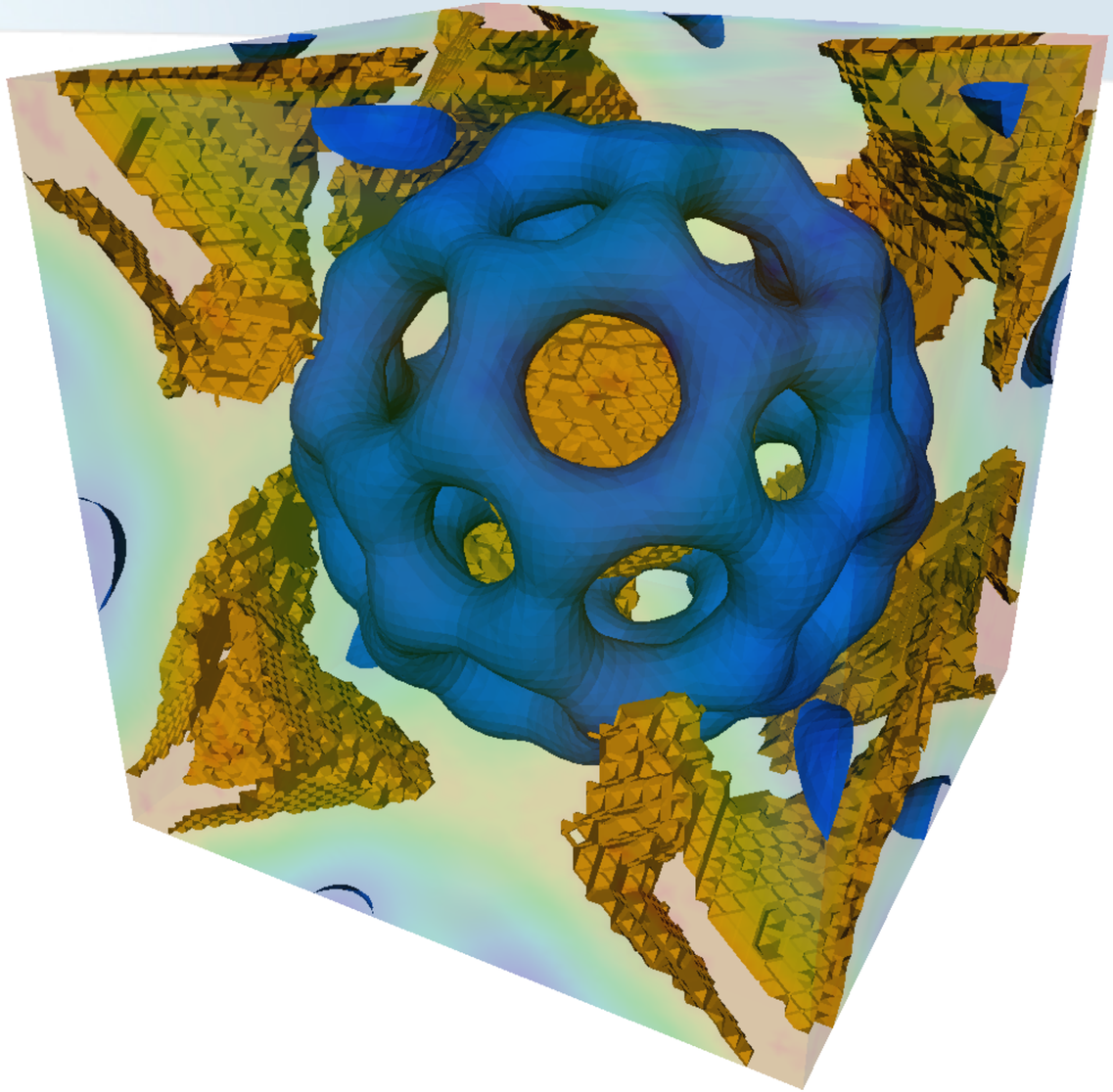
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



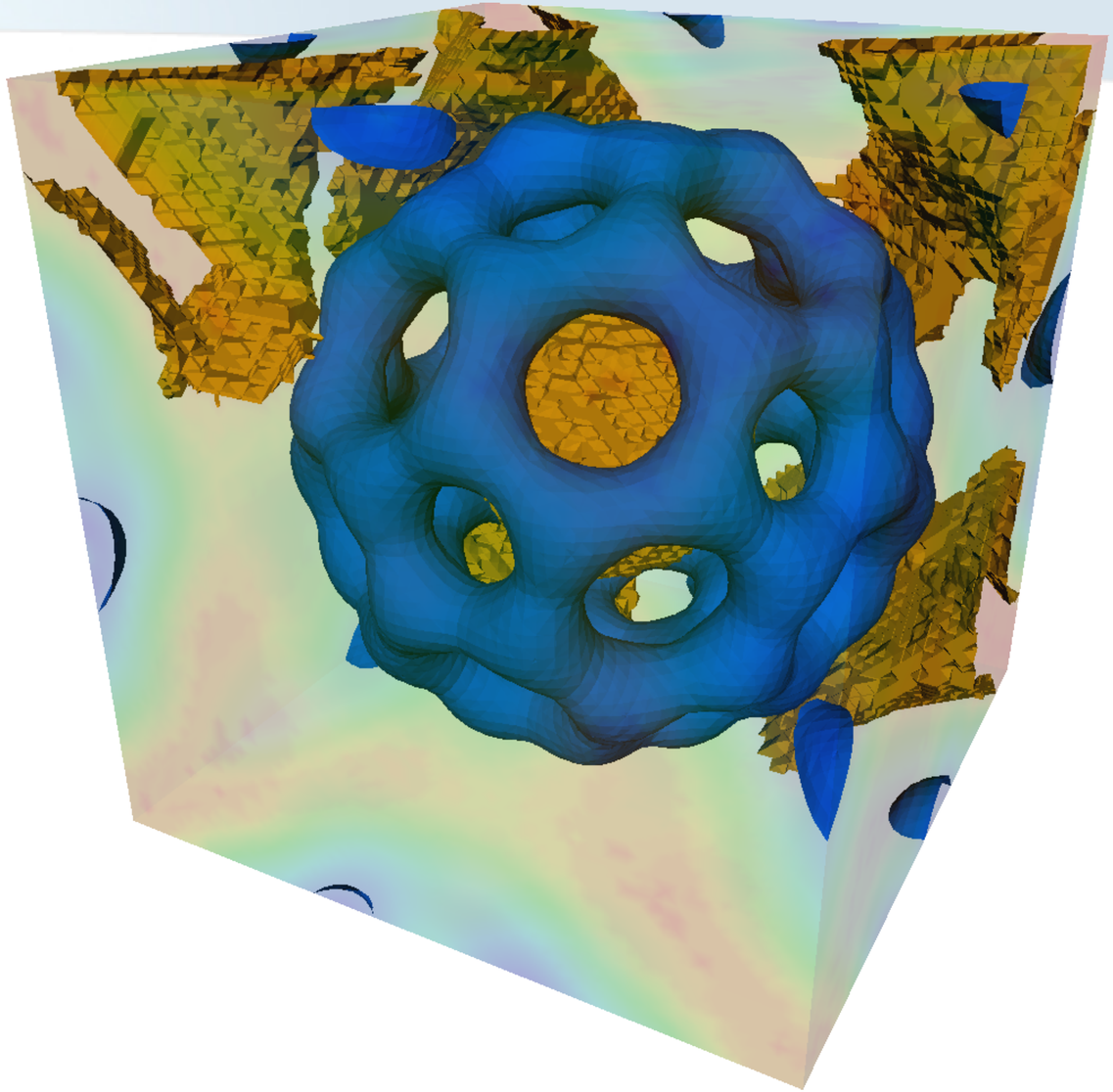
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



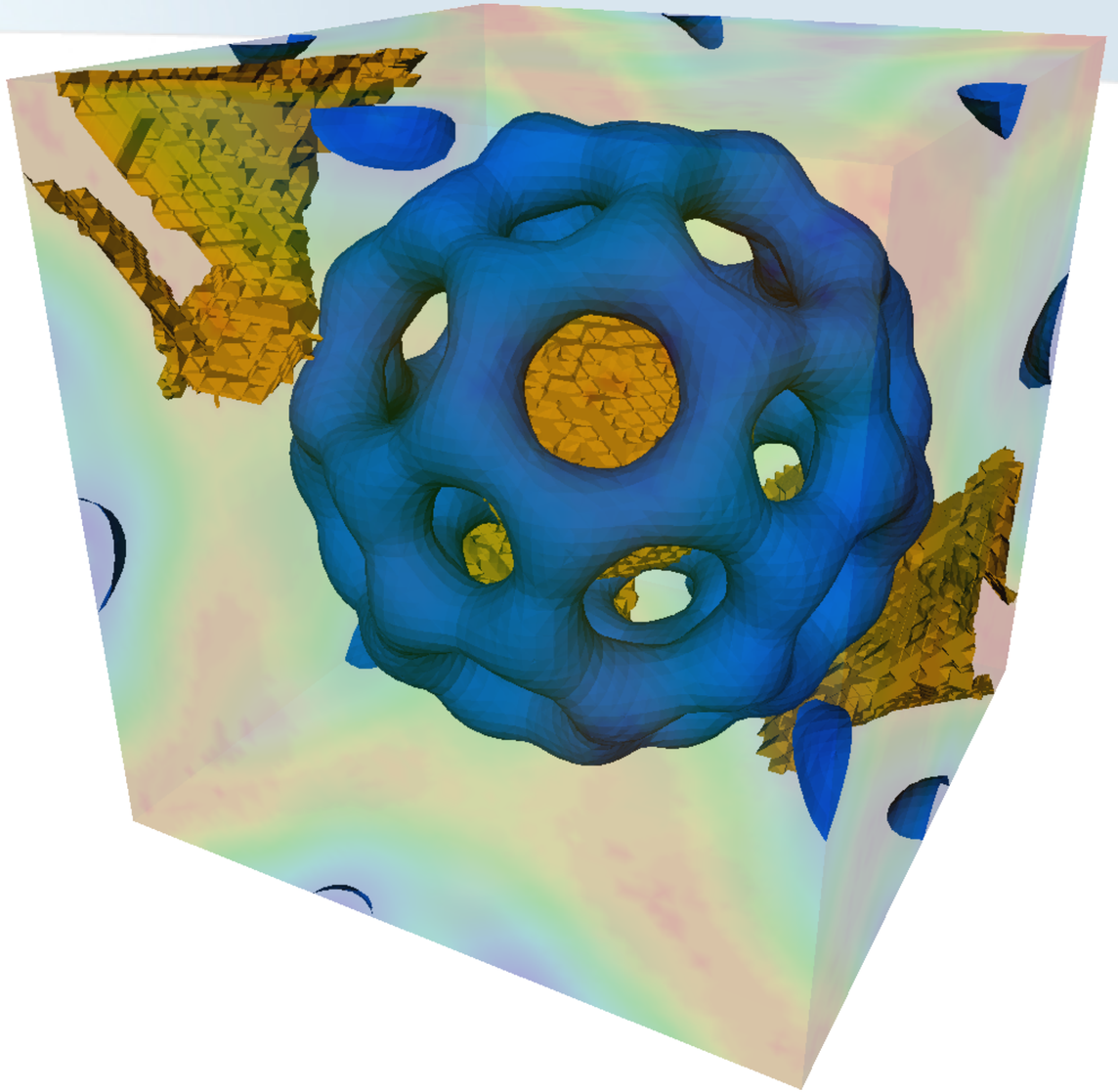
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



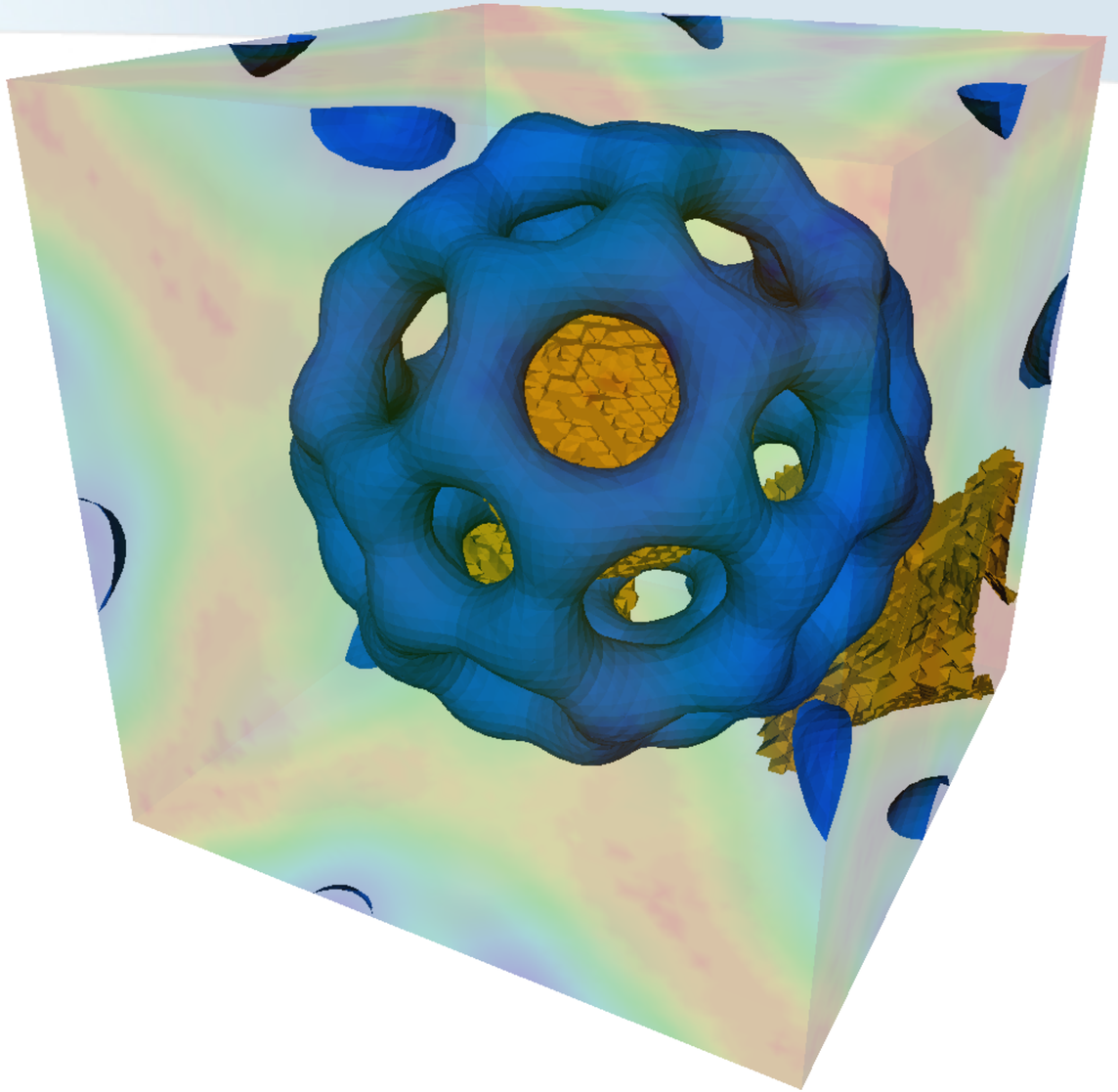
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



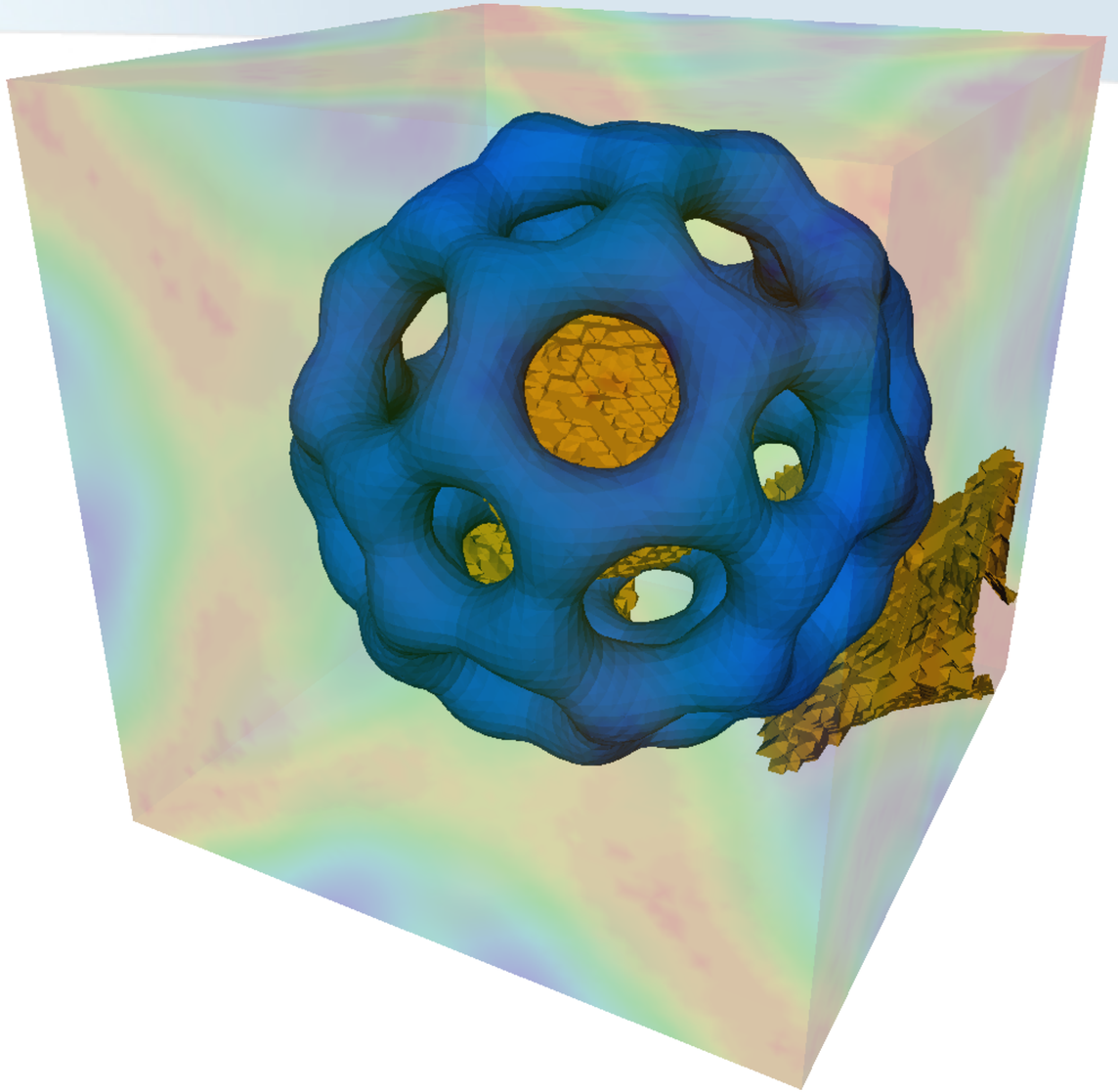
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



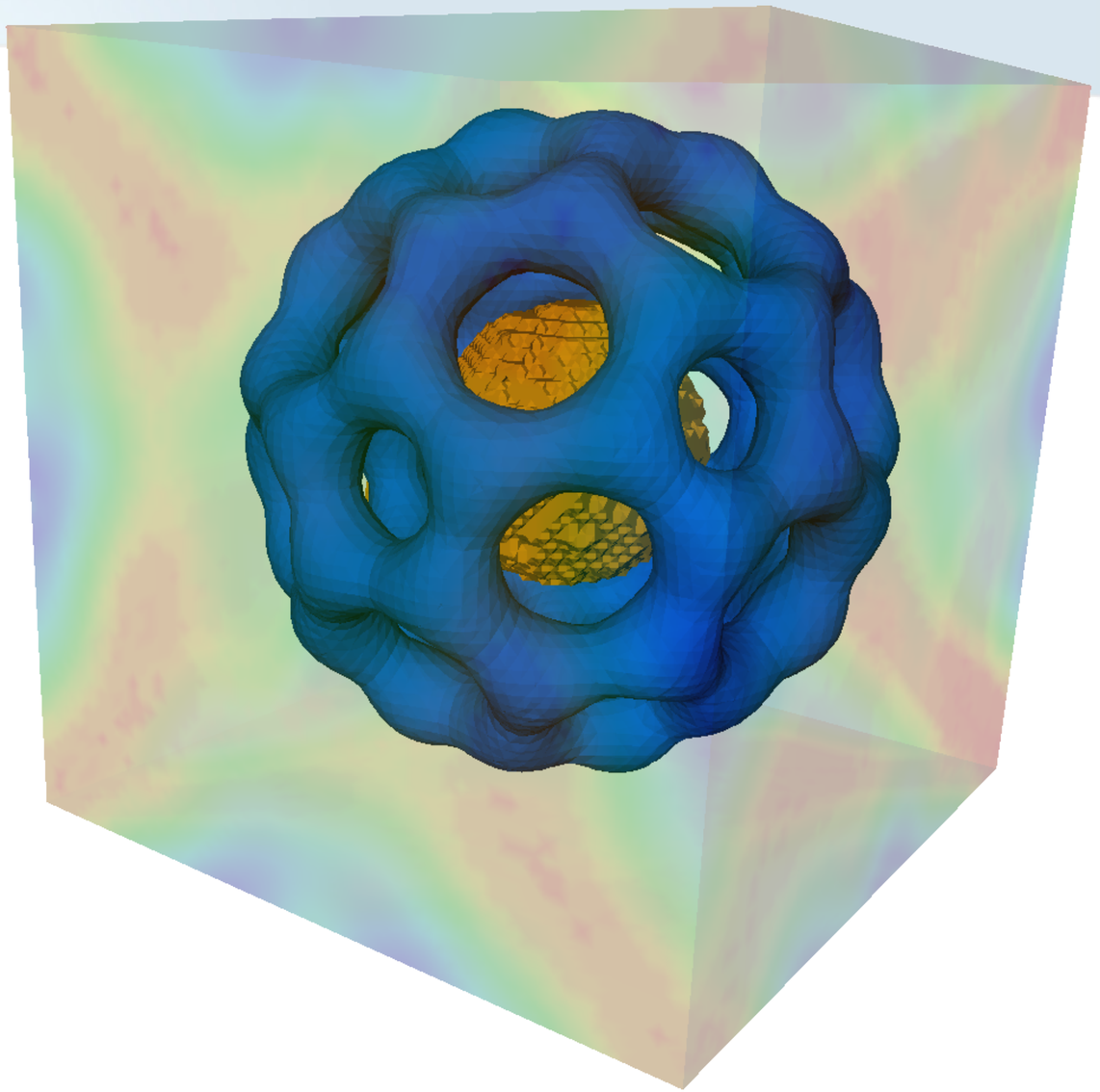
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



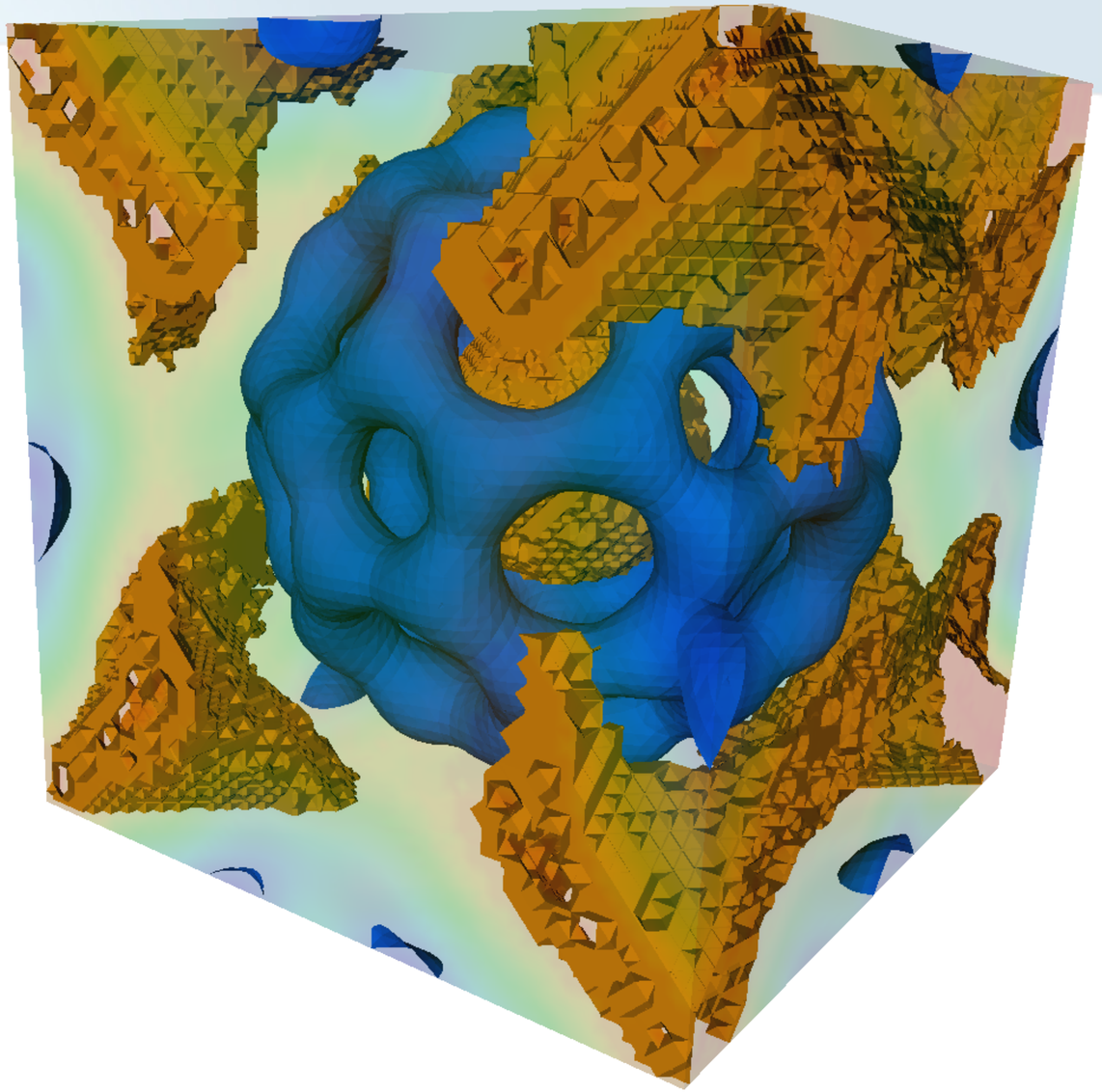
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



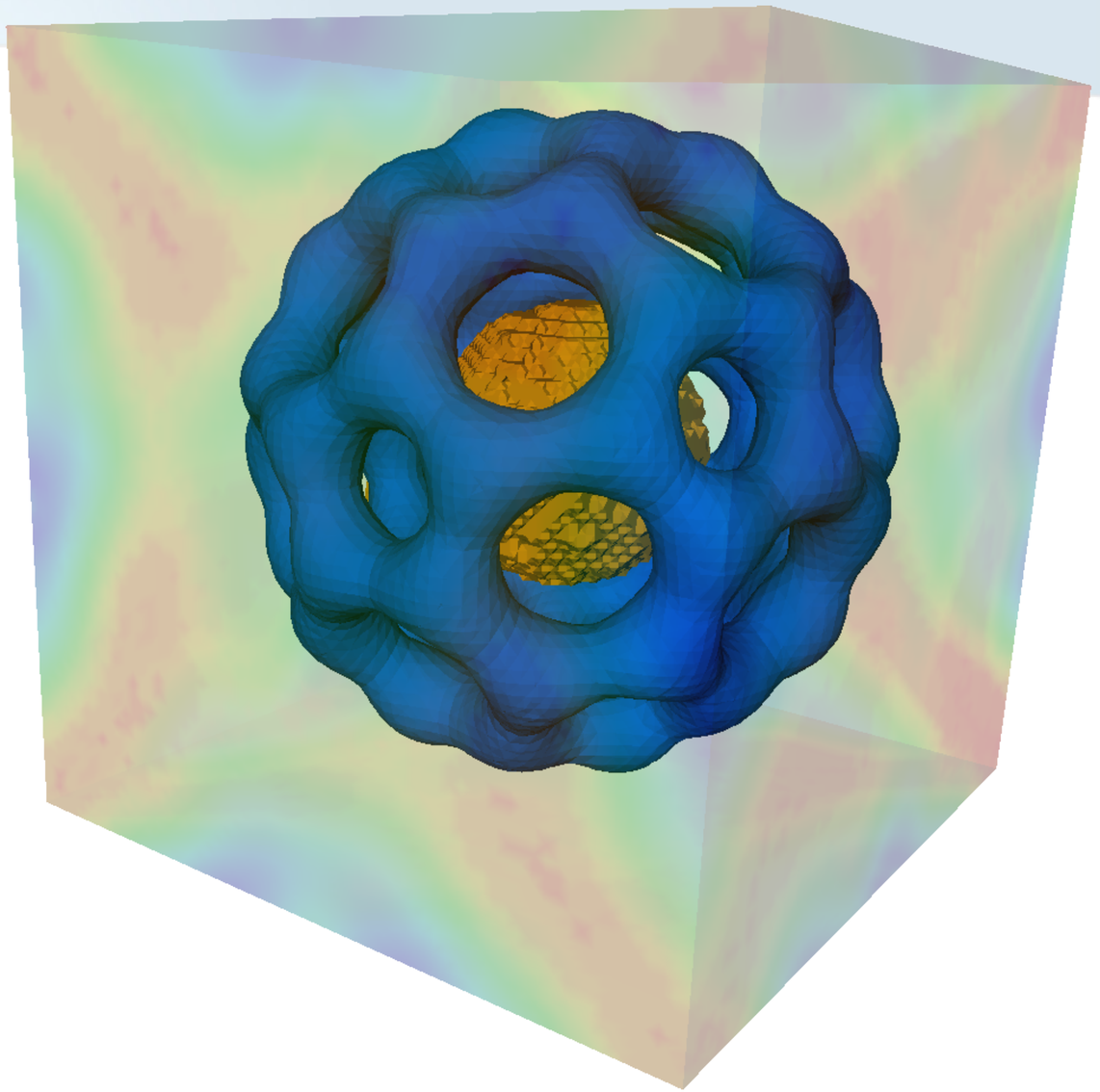
Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



Query-driven visualization

- Visualize isolated regions of interests
  - Level set extraction
  - Exact geometry visualization
  - Simplify occluders



Summary

Summary

- Higher dimension embeddings

Summary

- Higher dimension embeddings
- Color maps and volume rendering

Summary

- Higher dimension embeddings
- Color maps and volume rendering
- Level set computation

Summary

- Higher dimension embeddings
- Color maps and volume rendering
- Level set computation
  - PL-manifold domains
  - Regular grids

Summary

- Higher dimension embeddings
- Color maps and volume rendering
- Level set computation
  - PL-manifold domains
  - Regular grids
- Fast level set computation
- Topological simplification

Summary

- Higher dimension embeddings
- Color maps and volume rendering
- Level set computation
  - PL-manifold domains
  - Regular grids
- Fast level set computation
- Topological simplification

- Implementation examples

Higher dimensional embedding

- Given 1D regular grid
  - Create a 1-triangulation embedded in $\mathbb{R}^2$

$$f : [0, 24] \to \mathbb{R}$$
$$x = x$$
$$y = f(x)$$

Higher dimensional embedding

- Given 1D regular grid
  - Create a 1-triangulation embedded in $\mathbb{R}^2$

$$f : [0, 24] \rightarrow \mathbb{R}$$

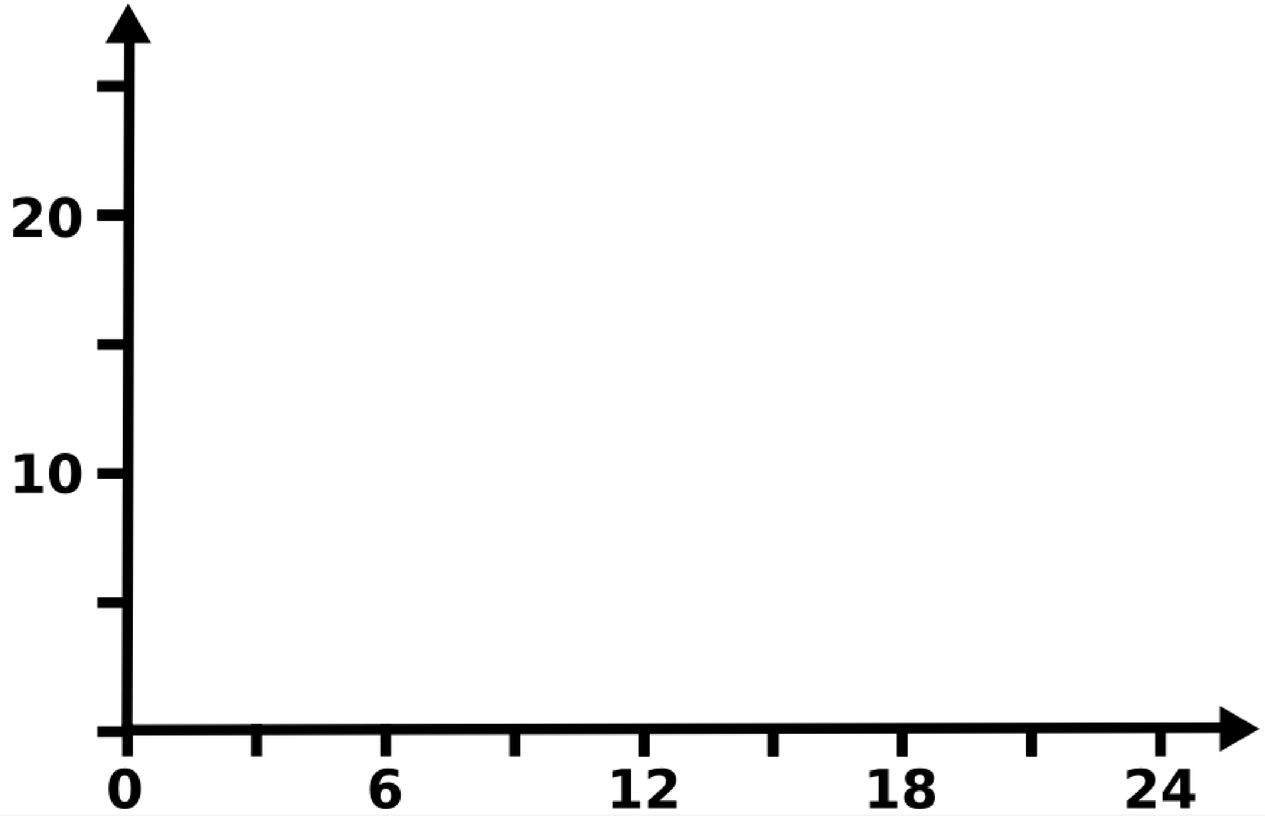
$$x = x$$

$$y = f(x)$$

Higher dimensional embedding

- Given 1D regular grid
  - Create a 1-triangulation embedded in $\mathbb{R}^2$

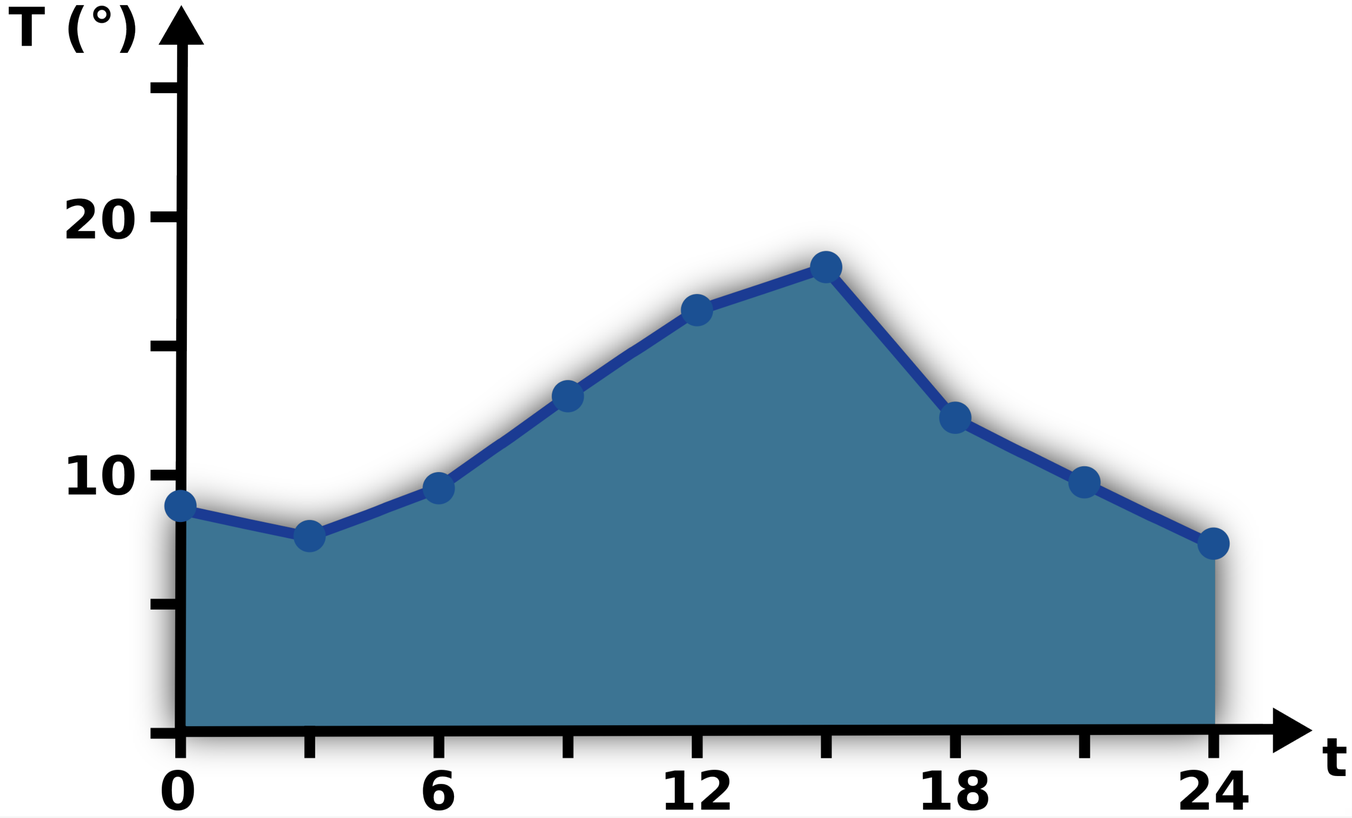
$$f : [0, 24] \rightarrow \mathbb{R}$$
$$x = x$$
$$y = f(x)$$



Higher dimensional embedding

- Given 1D regular grid
  - Create a 1-triangulation embedded in $\mathbb{R}^2$

$$f : [0, 24] \rightarrow \mathbb{R}$$
$$x = x$$
$$y = f(x)$$



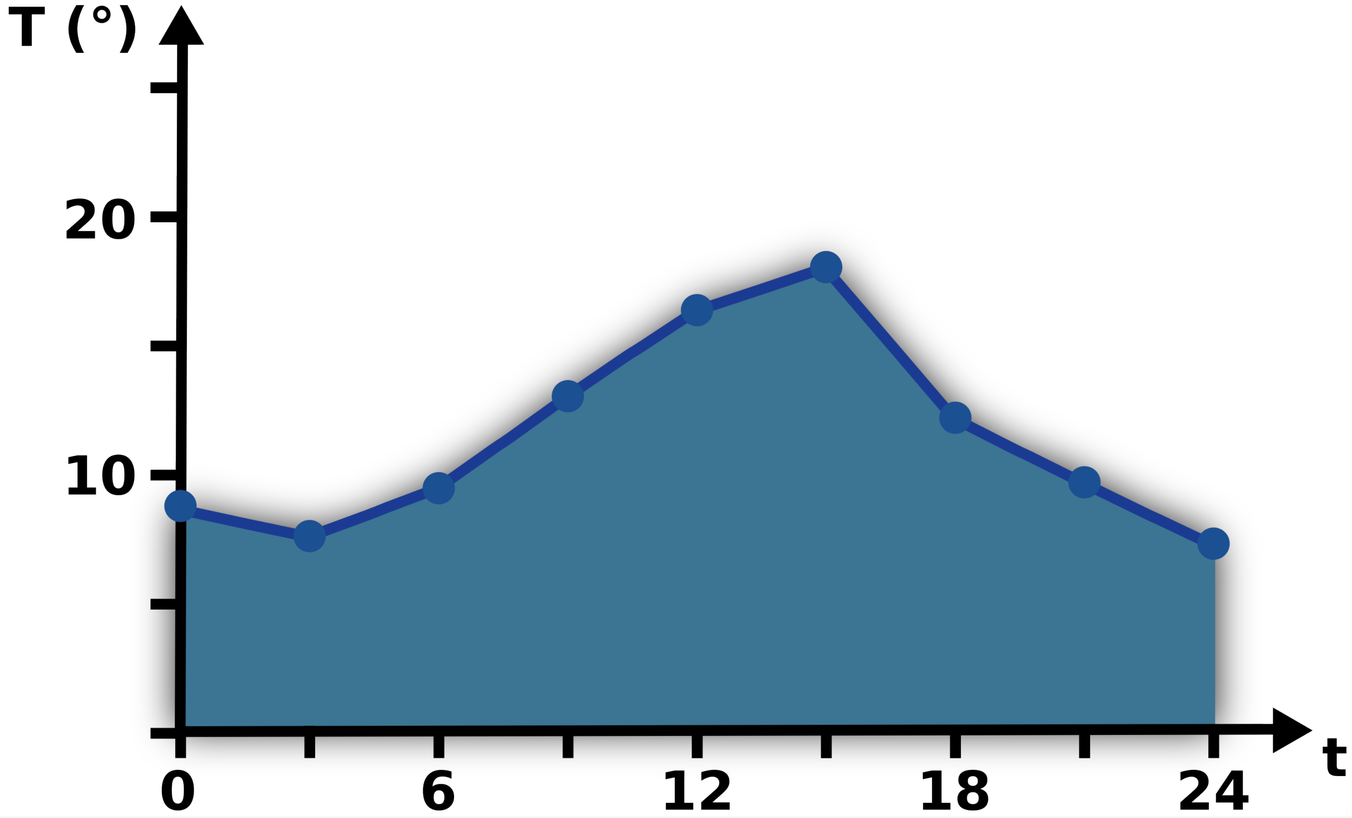
Higher dimensional embedding

- Given 1D regular grid
  - Create a 1-triangulation embedded in $\mathbb{R}^2$
  - **VTK PolyData**

$$f : [0, 24] \rightarrow \mathbb{R}$$

$$x = x$$

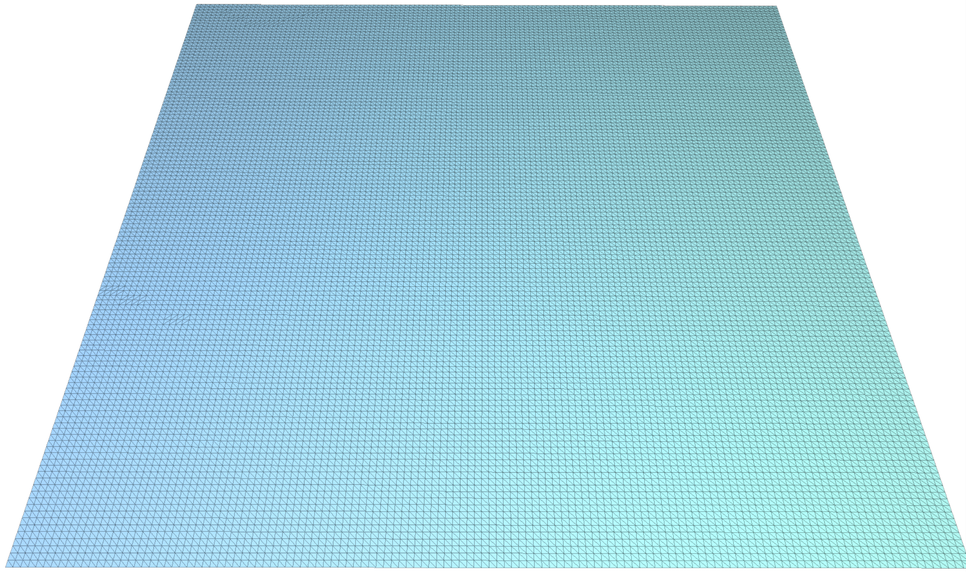
$$y = f(x)$$



Higher dimensional embedding

- Given 2D regular grid
  - Create a 2-triangulation embedded in $\mathbb{R}^3$

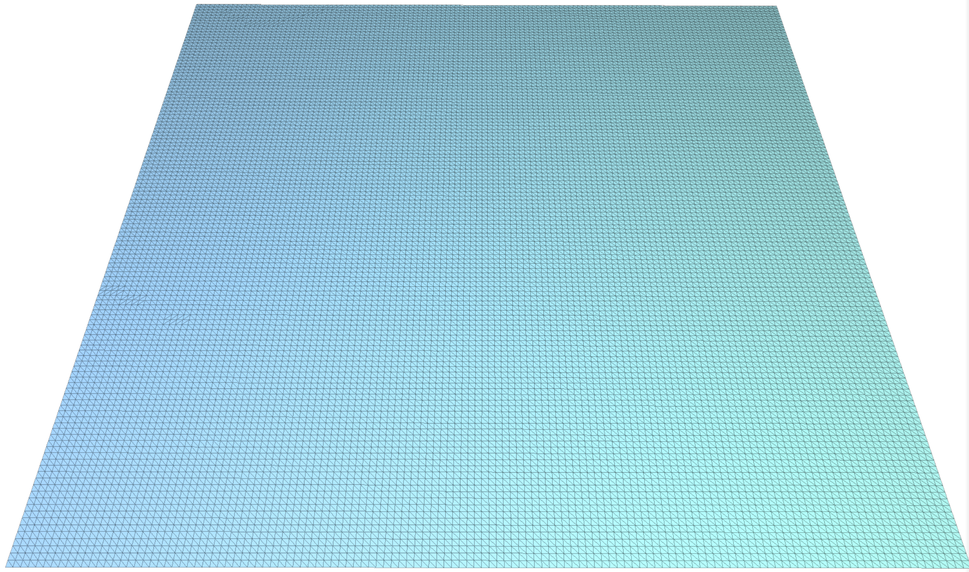
$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$



Higher dimensional embedding

- Given 2D regular grid
  - Create a 2-triangulation embedded in $\mathbb{R}^3$

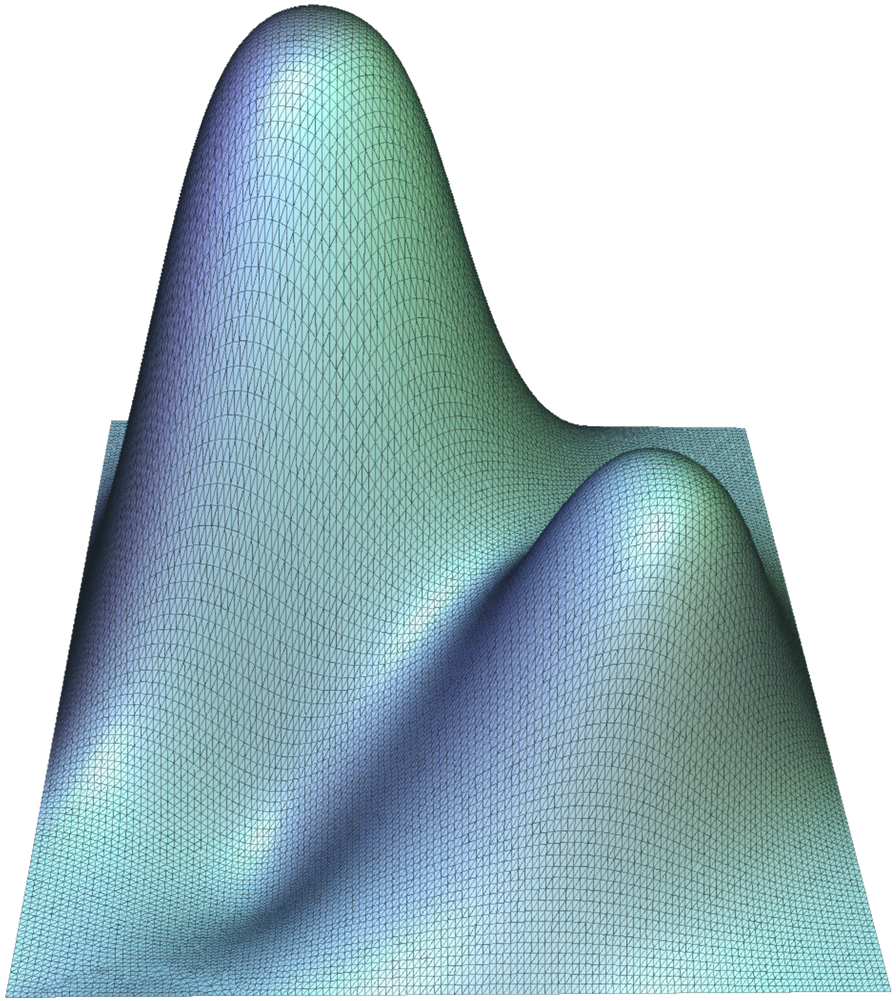
$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$
$$x = x$$
$$y = y$$
$$z = f(x, y)$$



Higher dimensional embedding

- Given 2D regular grid
  - Create a 2-triangulation embedded in $\mathbb{R}^3$

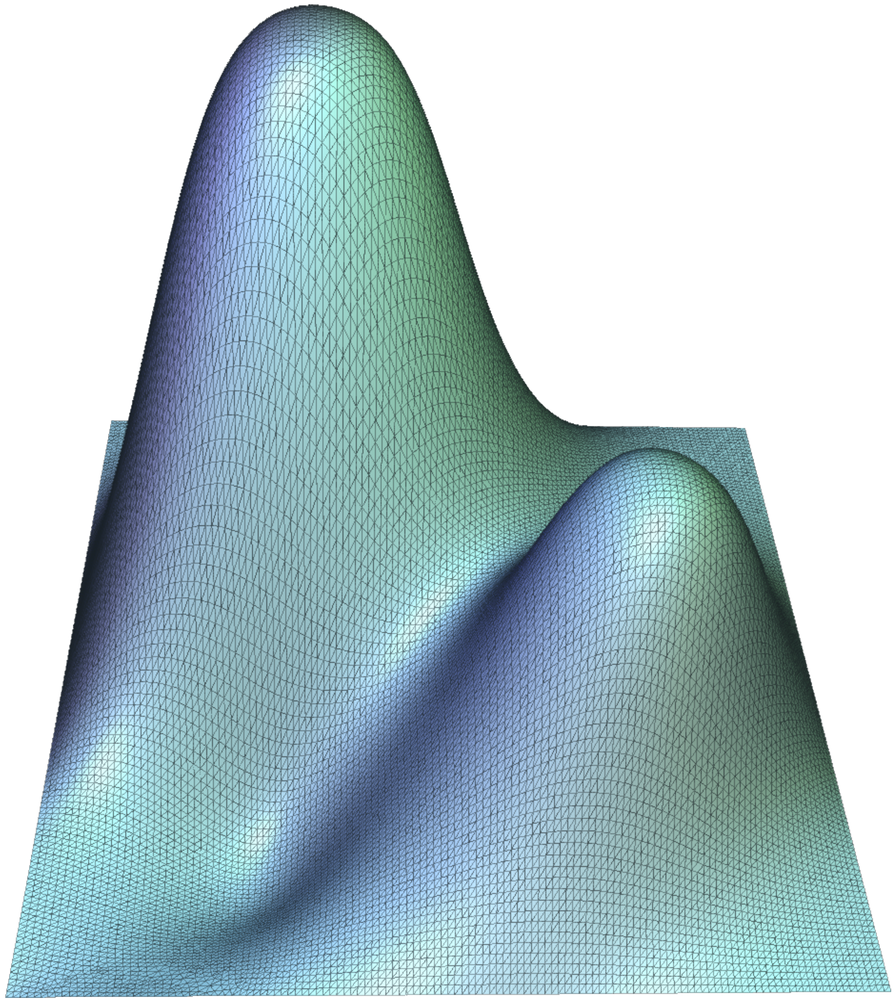
$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$
$$x = x$$
$$y = y$$
$$z = f(x, y)$$



Higher dimensional embedding

- Given 2D regular grid
  - Create a 2-triangulation embedded in $\mathbb{R}^3$
  - **VTK PolyData**

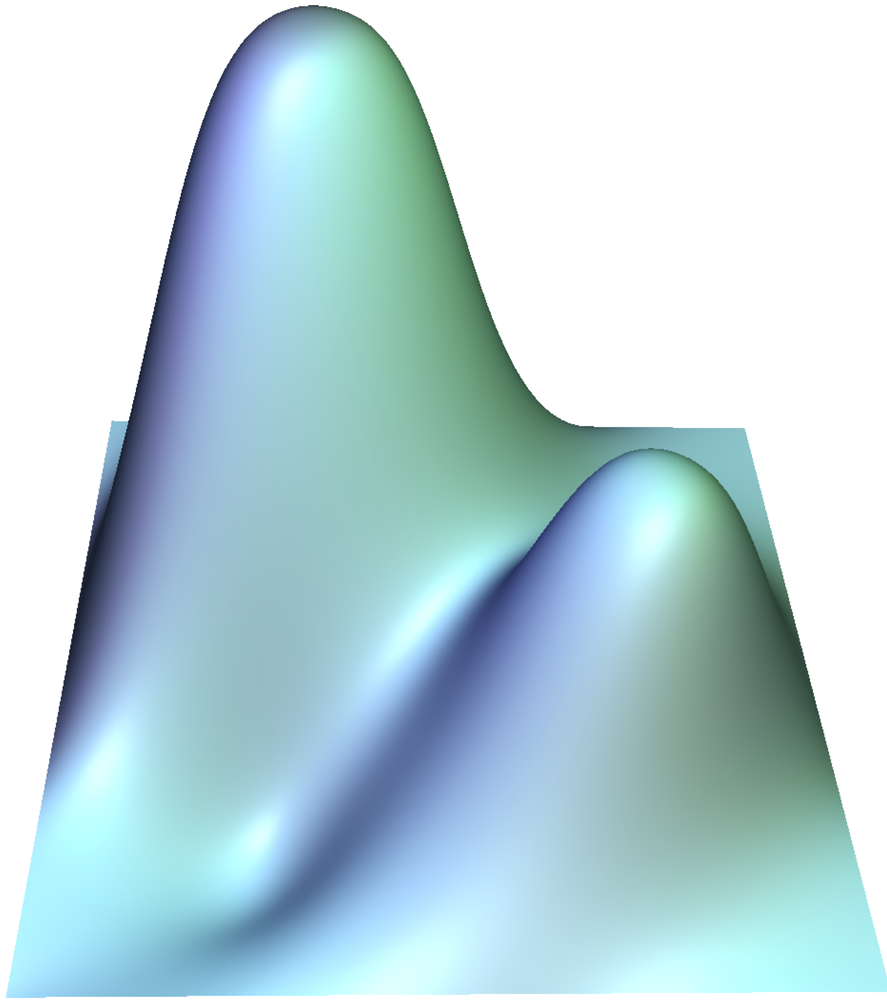
$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$
$$x = x$$
$$y = y$$
$$z = f(x, y)$$



Higher dimensional embedding

- Given 2D regular grid
  - Create a 2-triangulation embedded in $\mathbb{R}^3$
  - **VTK PolyData**

$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$
$$x = x$$
$$y = y$$
$$z = f(x, y)$$



Recap on projective rendering

- Interactive graphics
  - Direct interaction with the GPU device driver

Recap on projective rendering

- Interactive graphics
  - Direct interaction with the GPU device driver
    - OpenGL
    - DirectX

Recap on projective rendering

- Interactive graphics
  - Direct interaction with the GPU device driver
    - OpenGL
    - DirectX
  - Simplistic rendering

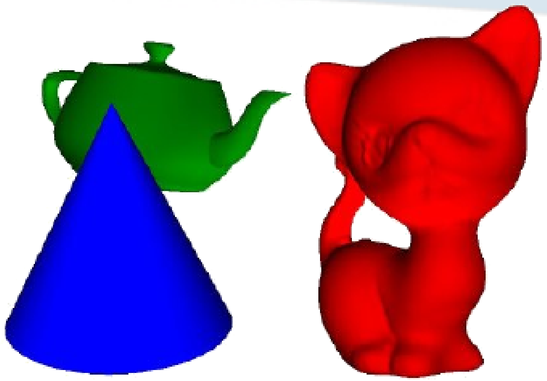
Recap on projective rendering

- Interactive graphics
  - Direct interaction with the GPU device driver
    - OpenGL
    - DirectX
  - Simplistic rendering
    - Especially for light interaction simulation

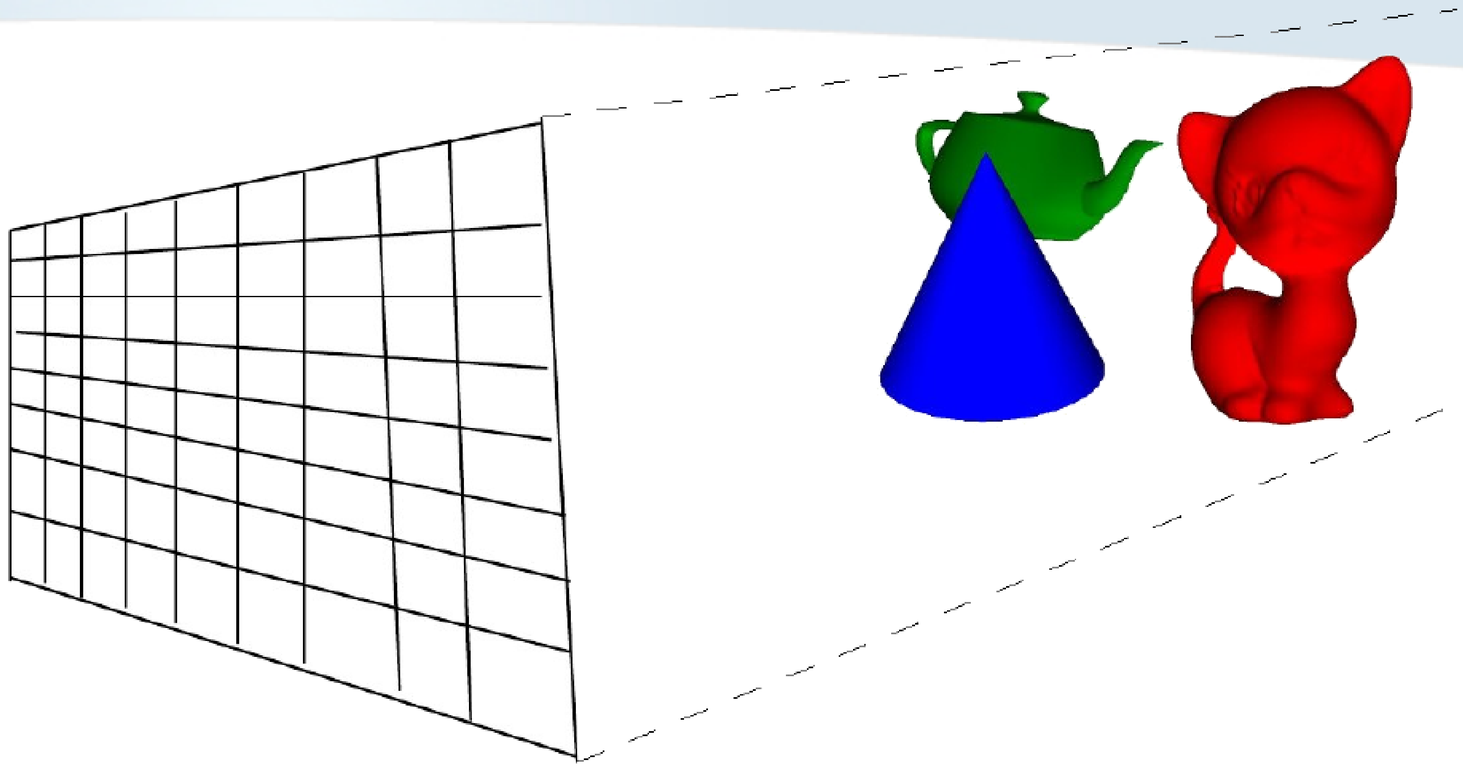
Recap on projective rendering

- Interactive graphics
  - Direct interaction with the GPU device driver
    - OpenGL
    - DirectX
  - Simplistic rendering
    - Especially for light interaction simulation
    - Guarantees interactive feedback

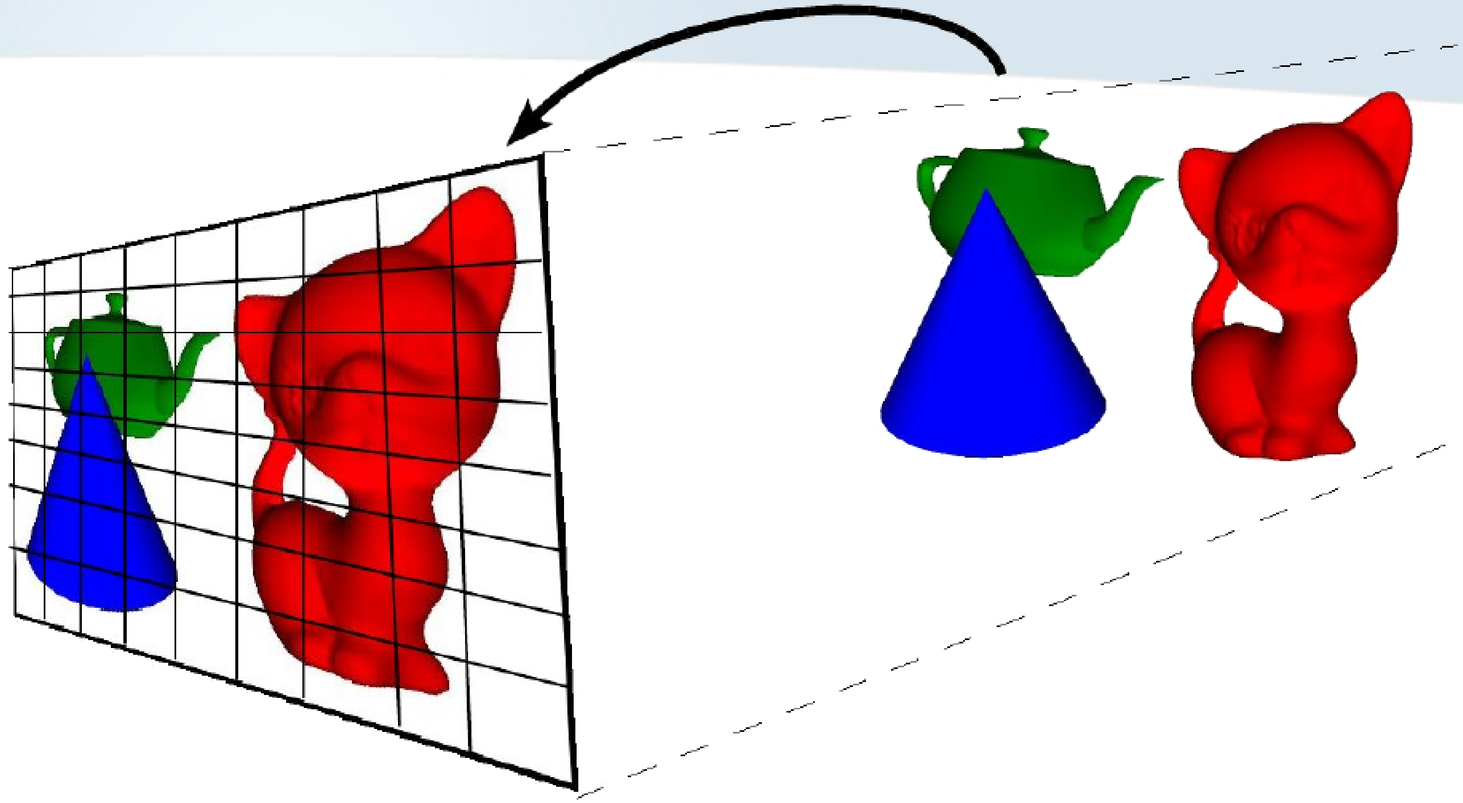
Recap on projective rendering



Recap on projective rendering

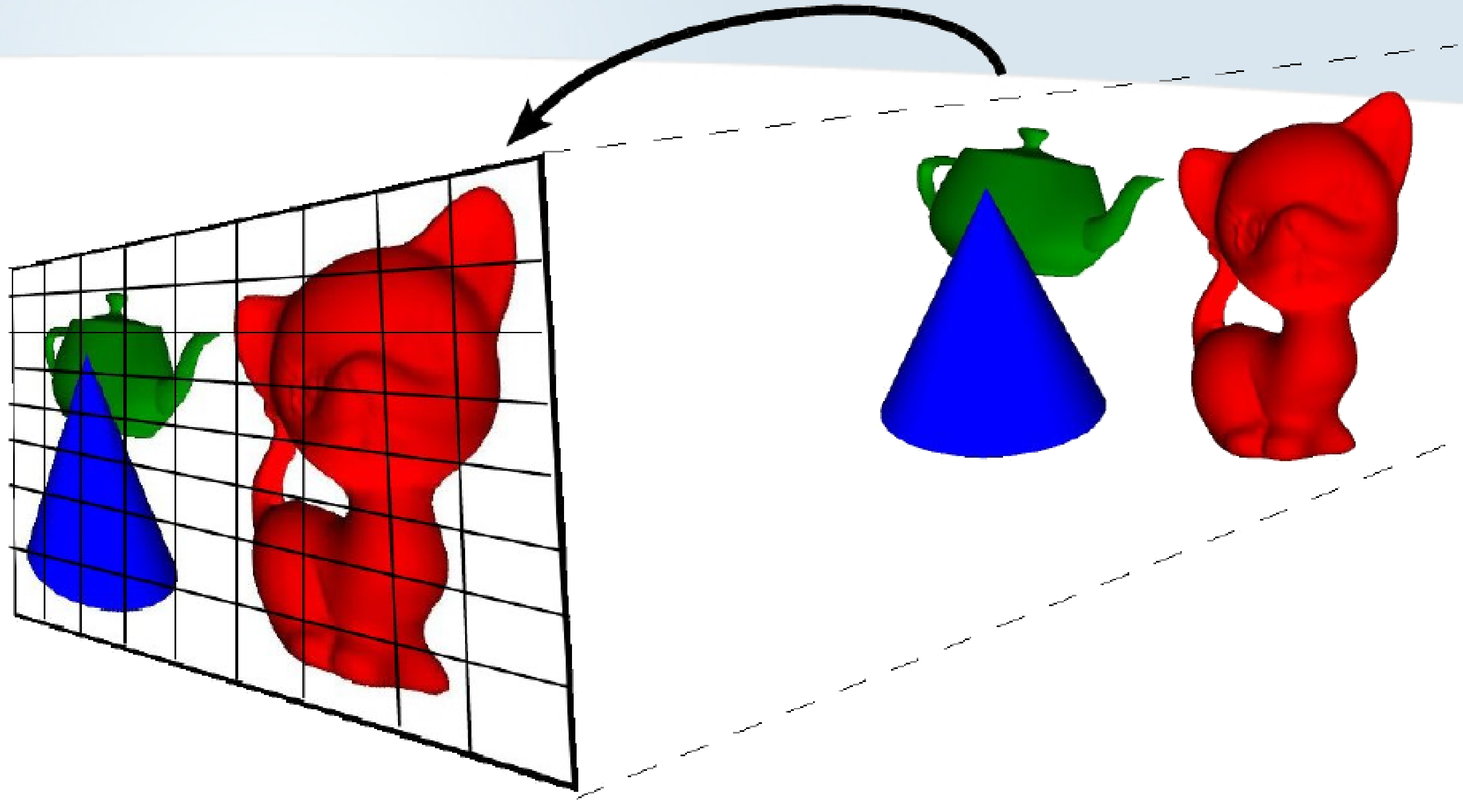


Recap on projective rendering



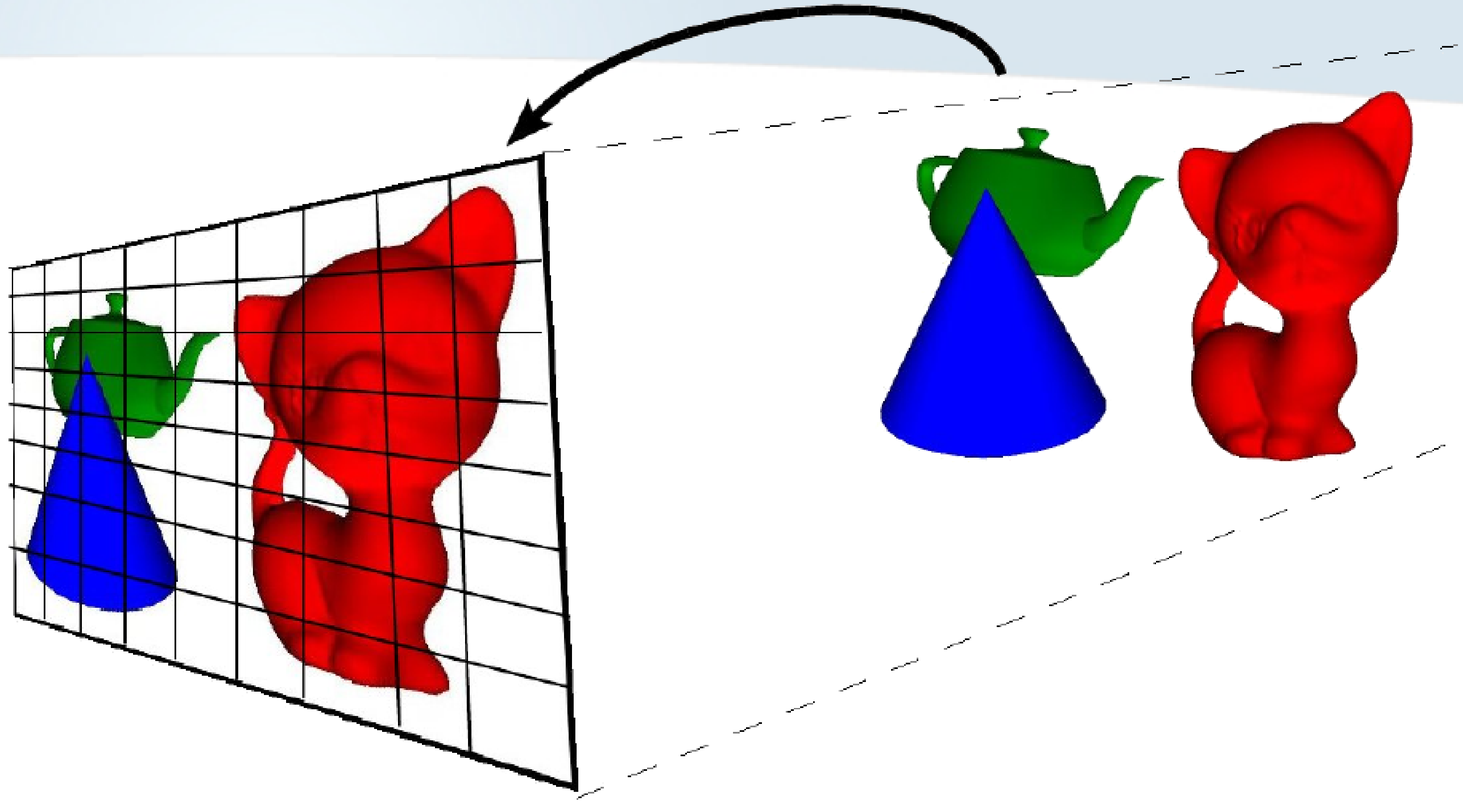
Recap on projective rendering

- Position matrices



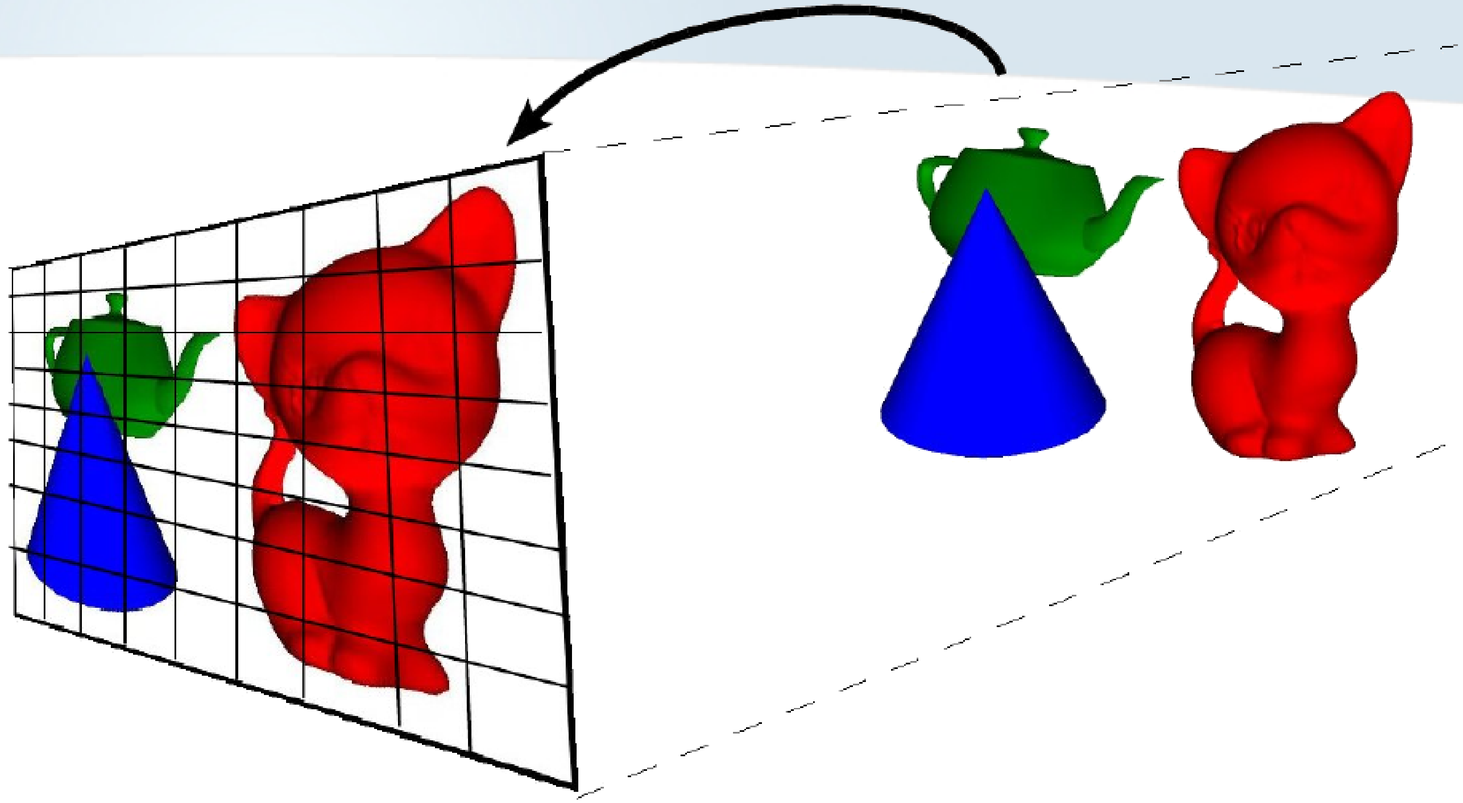
Recap on projective rendering

- Position matrices
- Color and light computation



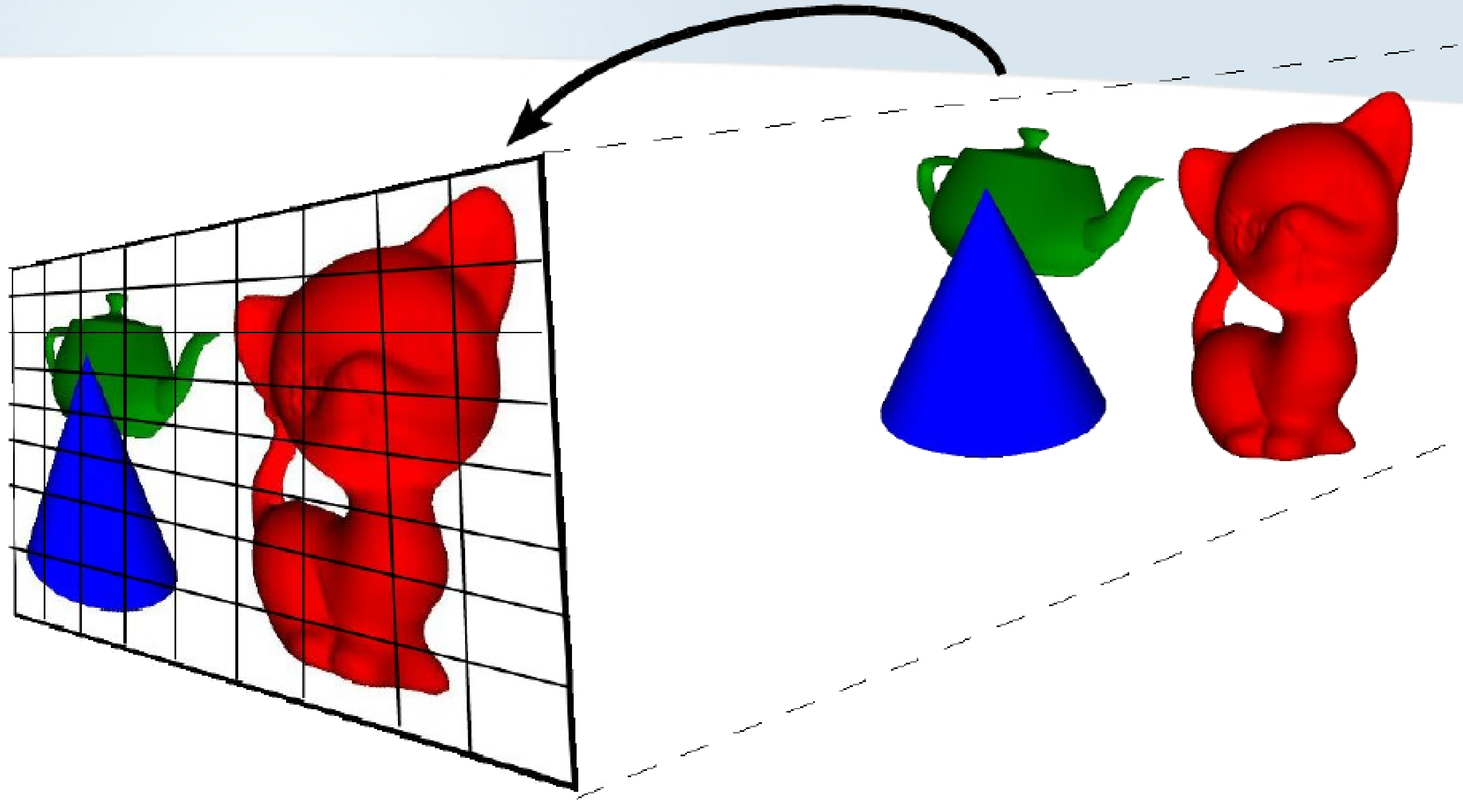
Recap on projective rendering

- Position matrices
- Color and light computation
- Projection matrix (camera)



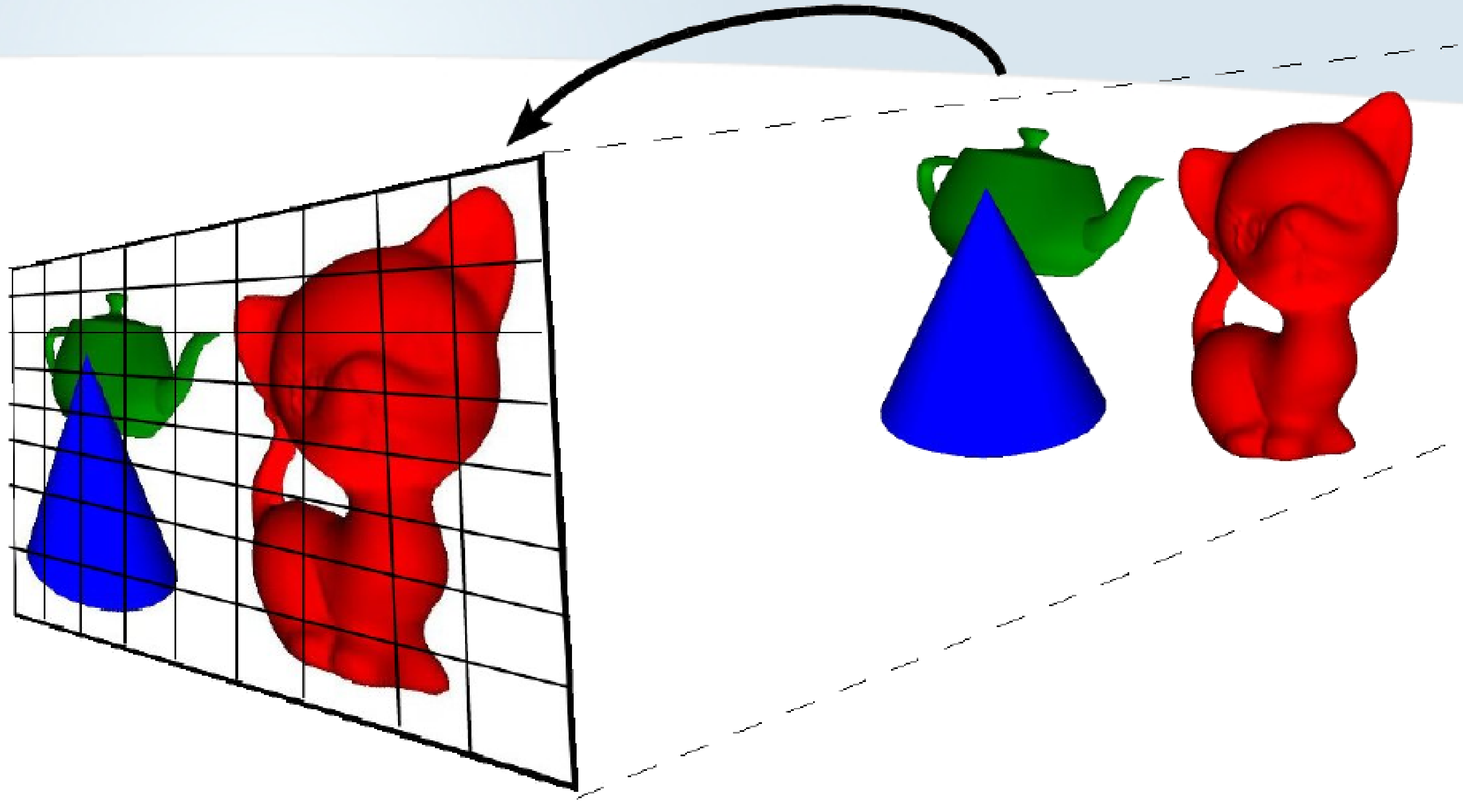
Recap on projective rendering

- Position matrices
- Color and light computation
- Projection matrix (camera)
  - **Matrix multiplication**



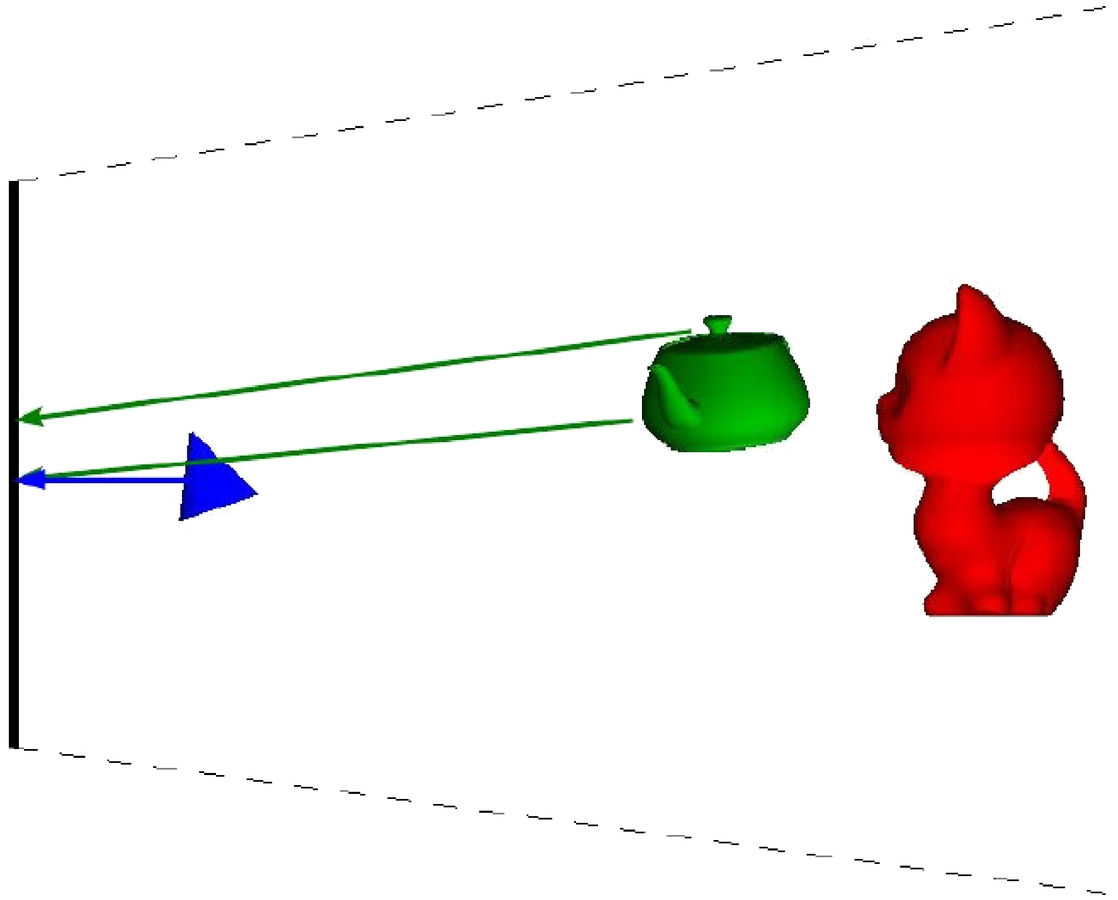
Recap on projective rendering

- Position matrices
- Color and light computation
- Projection matrix (camera)
  - **Matrix multiplication**
- Screen-space tests



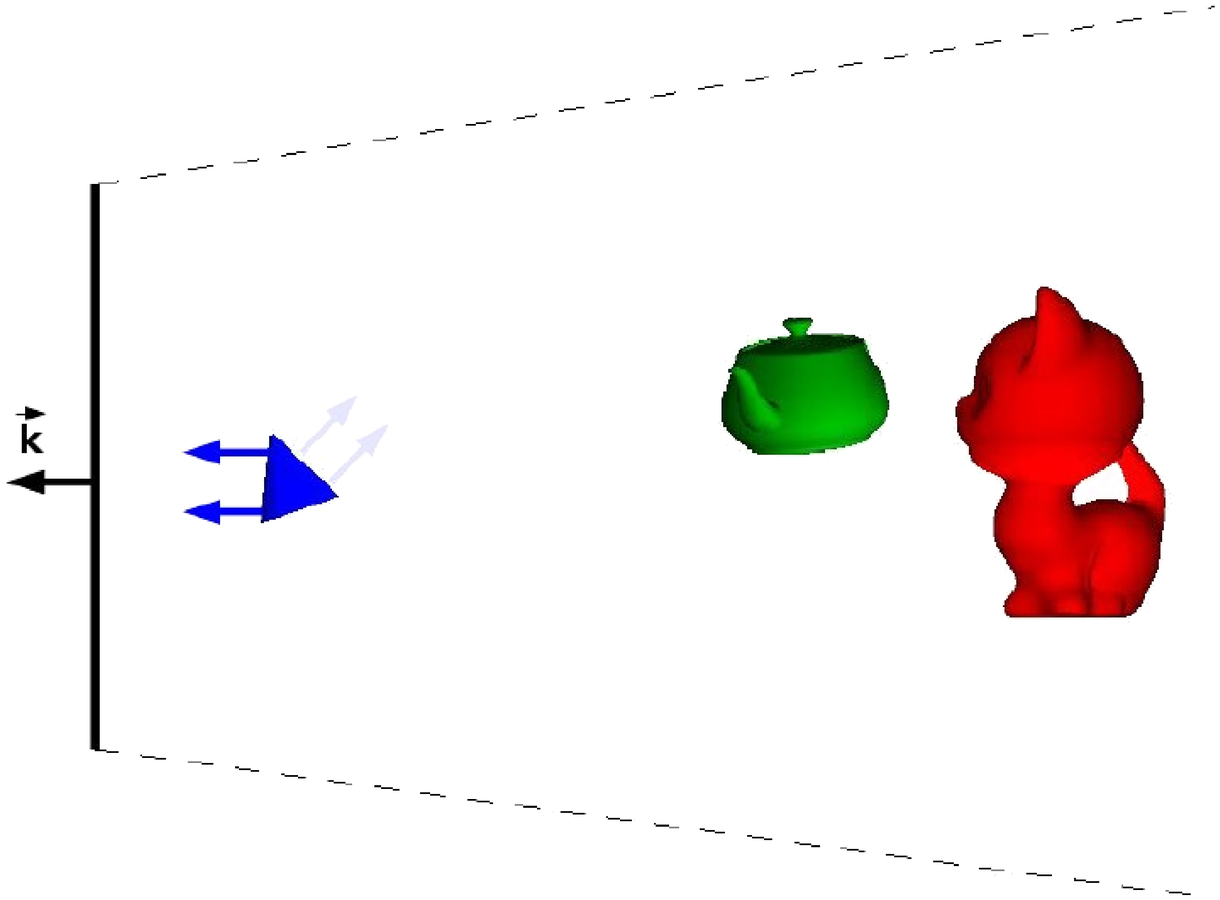
Recap on projective rendering

- Screen-space tests
  - Among others
    - Depth buffer
    - Back-face culling



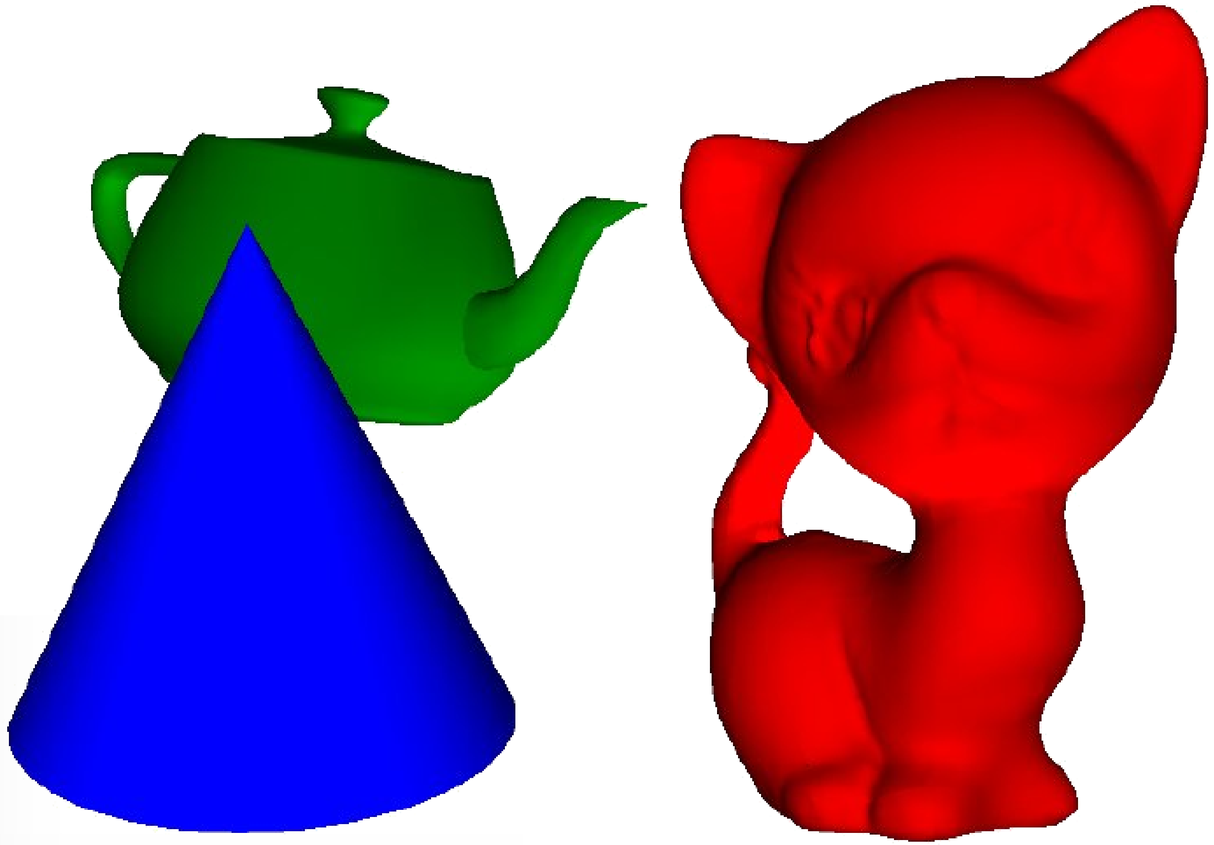
Recap on projective rendering

- Screen-space tests
  - Among others
    - Depth buffer
    - Back-face culling



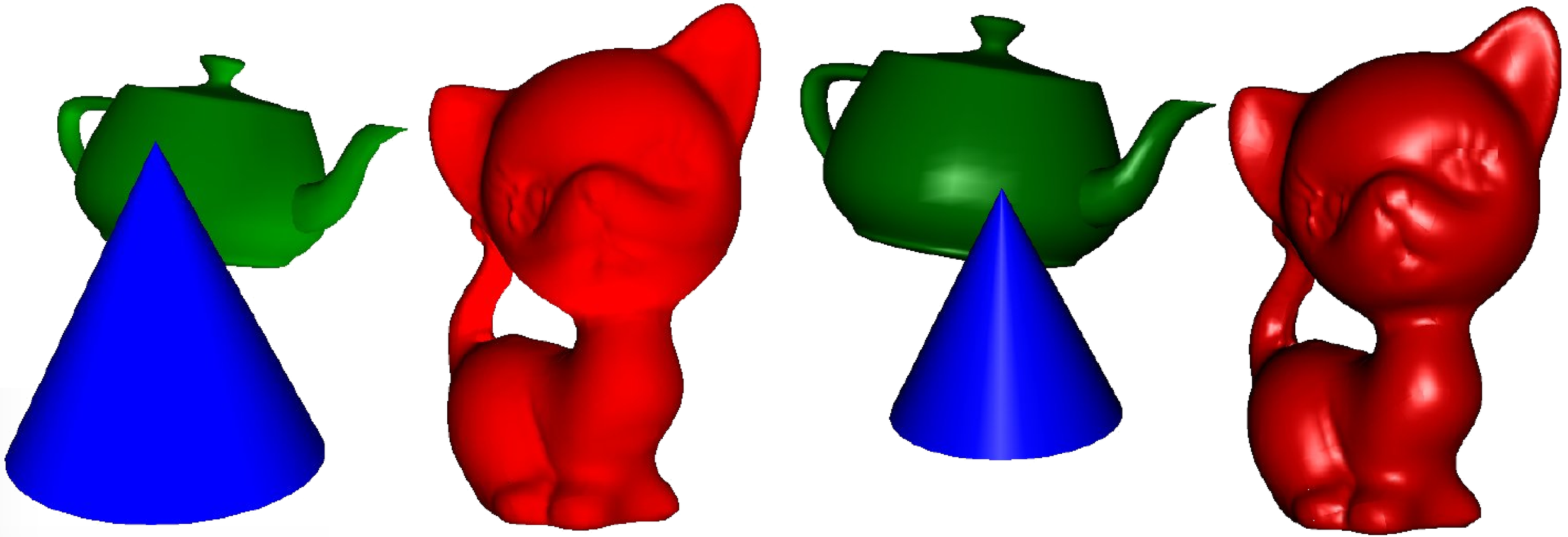
Recap on projective rendering

- Light modeling



Recap on projective rendering

- Light modeling

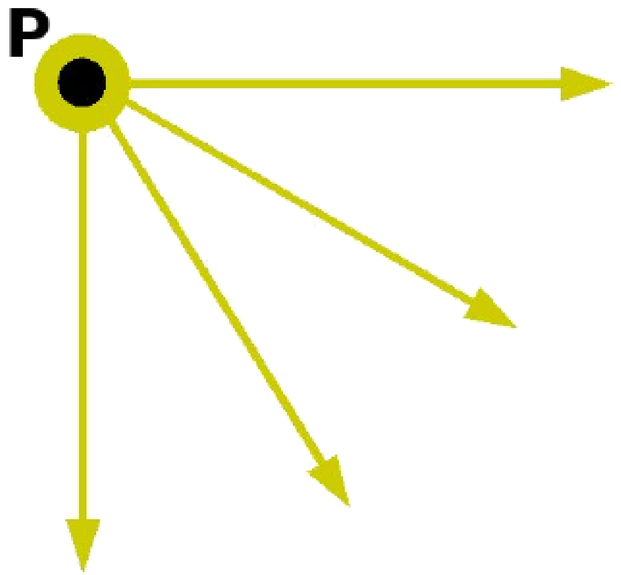


Recap on projective rendering

- Light modeling
  - Local model

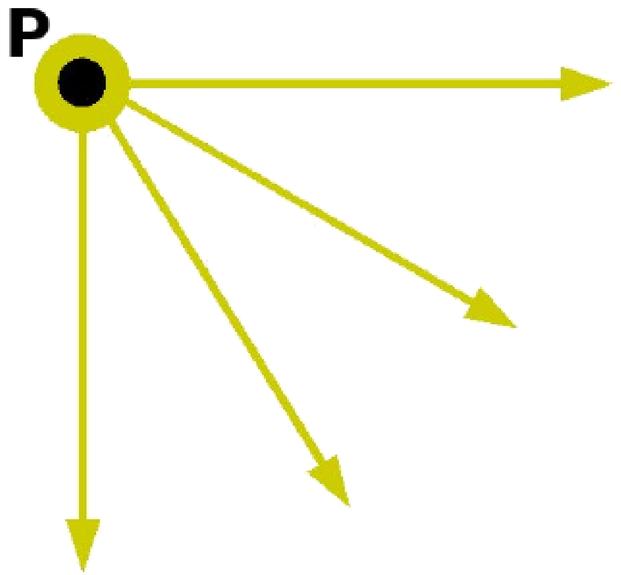
Recap on projective rendering

- Light modeling
  - Local model



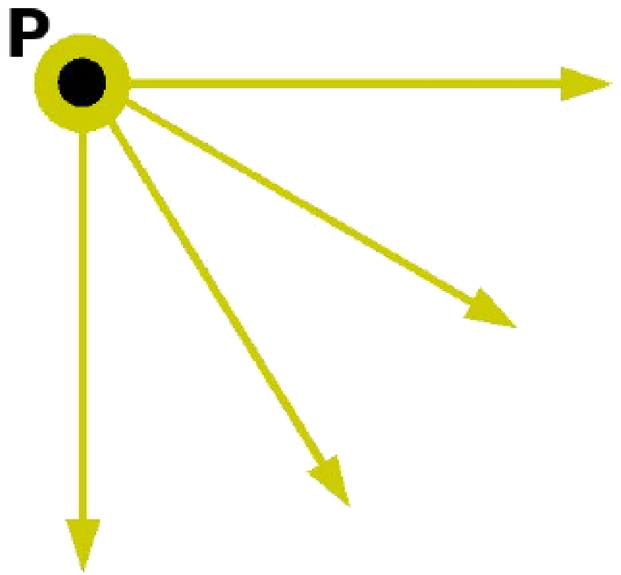
Recap on projective rendering

- Light modeling
  - Local model
  - Each object independently
  - Each polygon independently



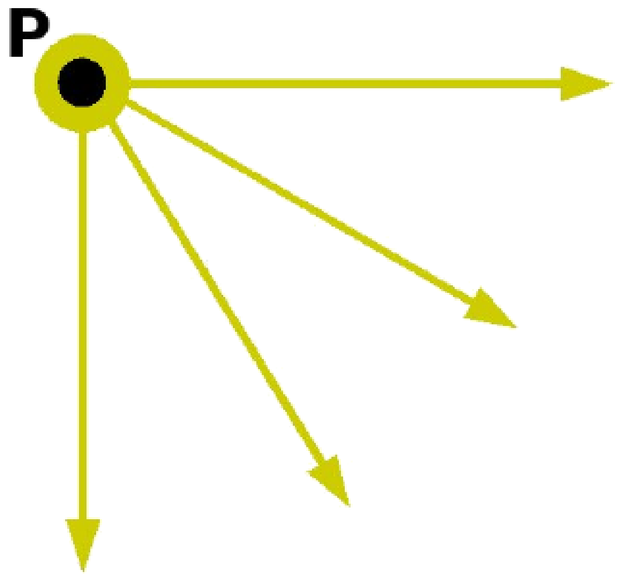
Recap on projective rendering

- Light modeling
  - Local model
  - Each object independently
  - Each polygon independently
    - No shadow
    - No inter-object interaction



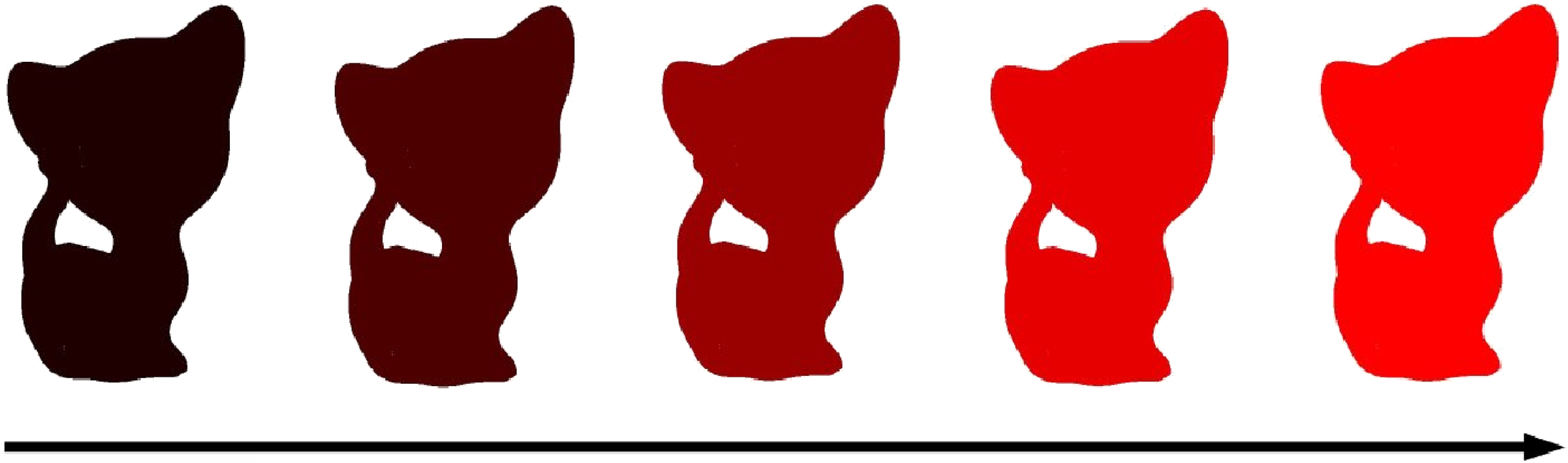
Recap on projective rendering

- Light modeling
  - Local model
  - Each object independently
  - Each polygon independently
    - No shadow
    - No inter-object interaction
  - Ambient attributes
  - Diffuse attributes
  - Specular attributes



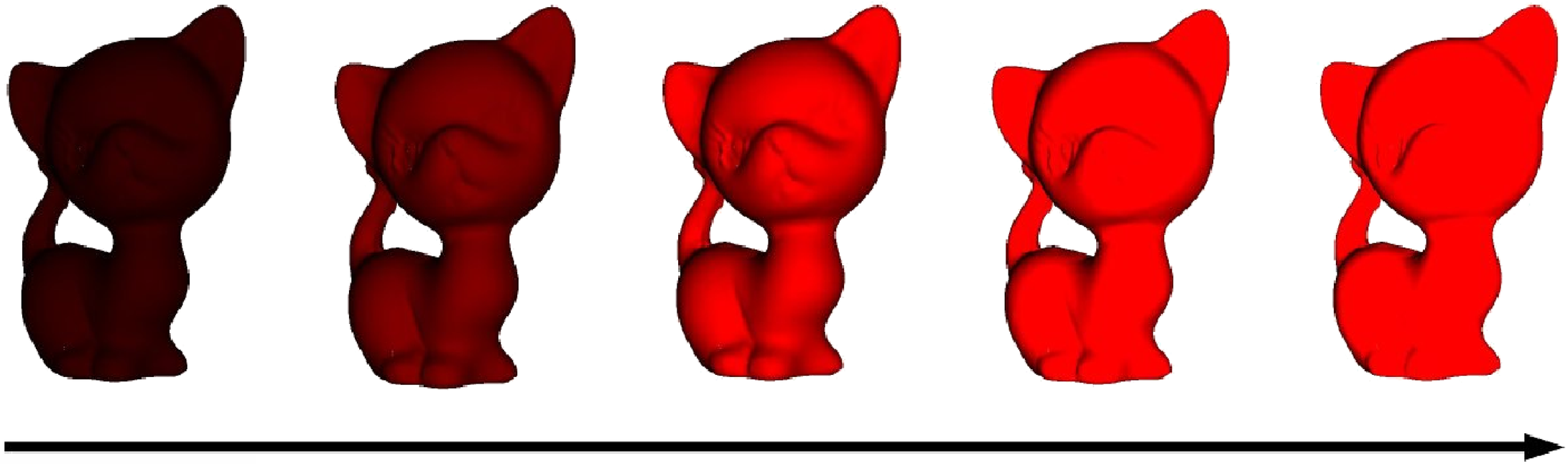
Recap on projective rendering

- Ambient contribution (the color of the object, per RGB channel)
  - $C_A(p) = k_a \times I_a$



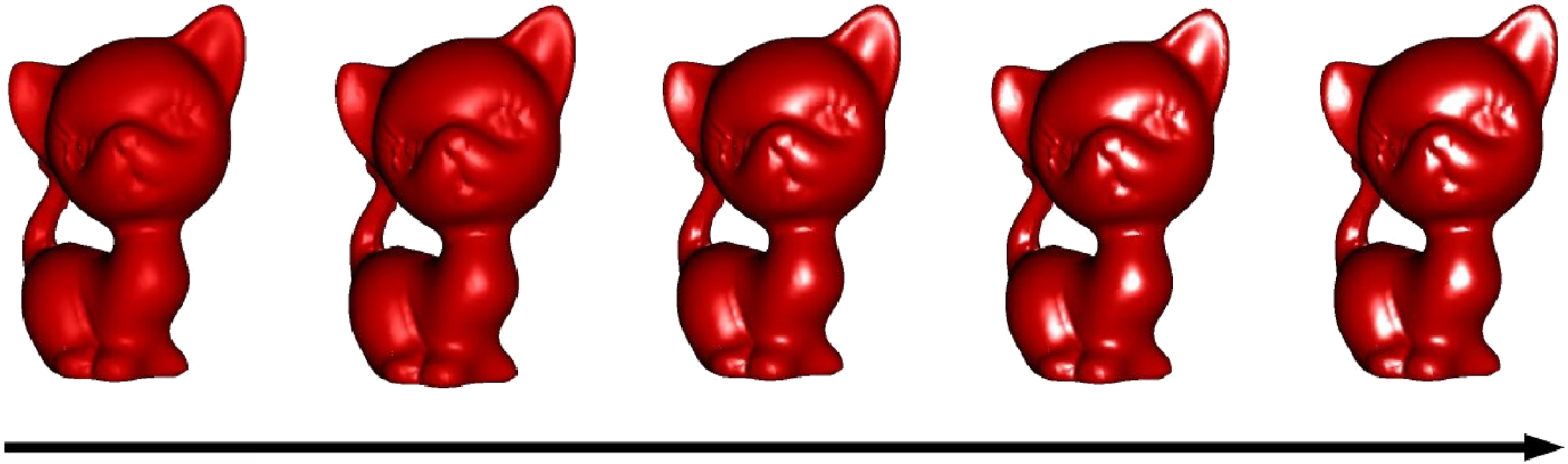
Recap on projective rendering

- Diffuse contribution (per RGB channel)
  - $C_D(p) = I_d \times k_d \times cos(\vec{N}, \vec{L}) \quad if\ \vec{N}.\vec{L} \geq 0$



Recap on projective rendering

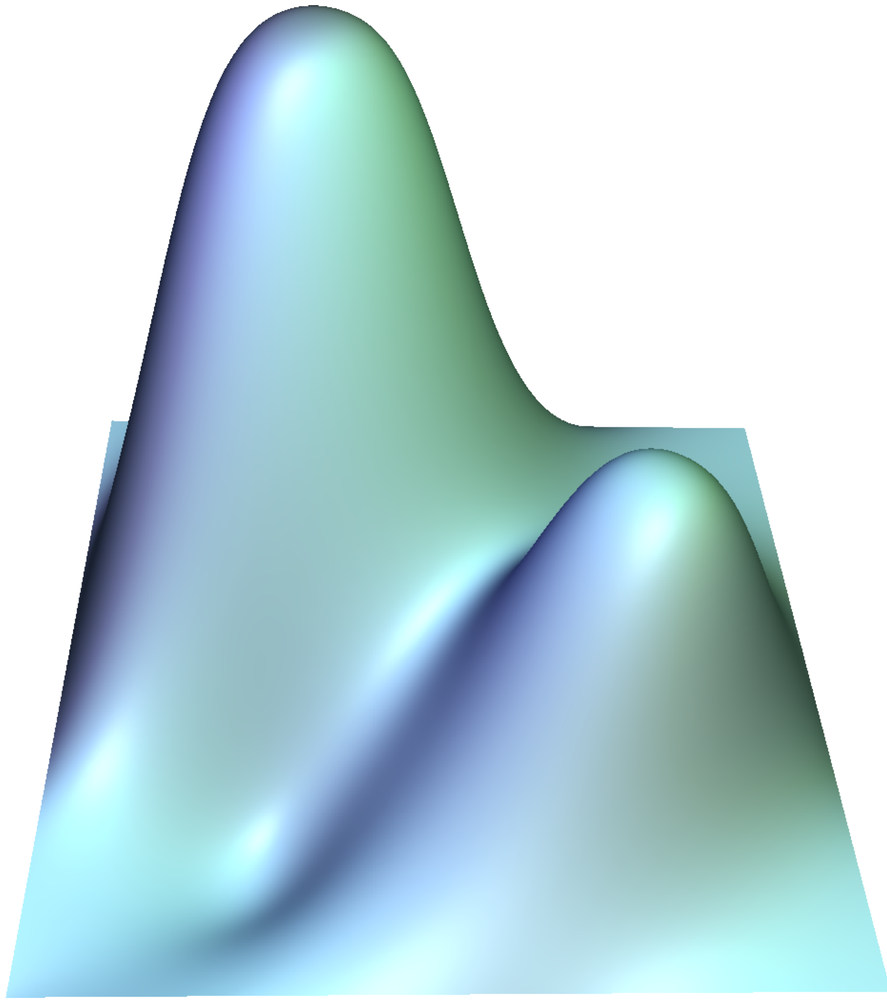
- Diffuse contribution (per RGB channel)
  - $C_S(p) = I_s \times k_s \times cos(\vec{V}, \vec{R})^{Sh}$



Higher dimensional embedding

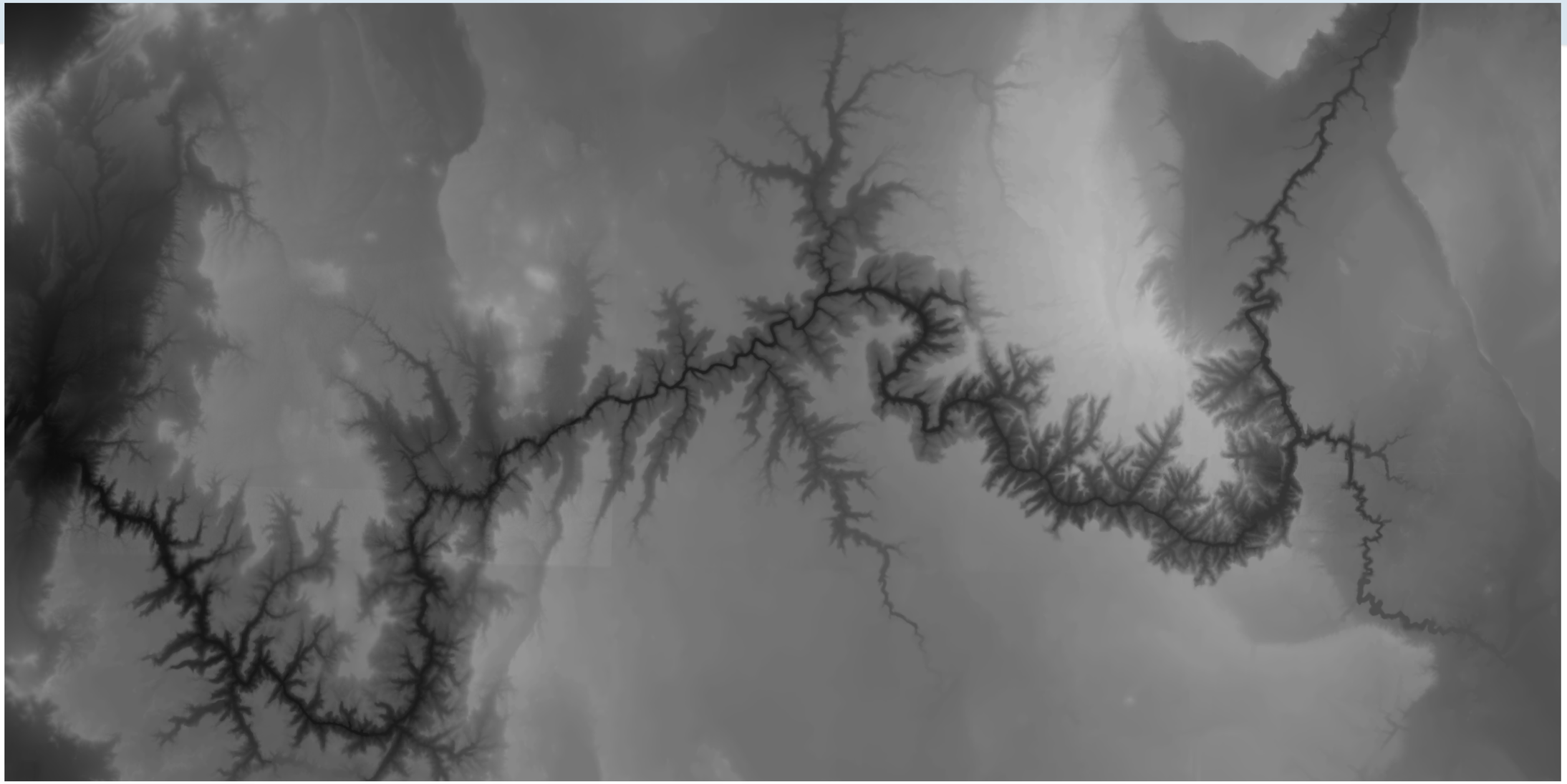
- Given 2D regular grid
  - Create a 2-triangulation embedded in $\mathbb{R}^3$
  - **VTK PolyData**

$$f : \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$$
$$x = x$$
$$y = y$$
$$z = f(x, y)$$

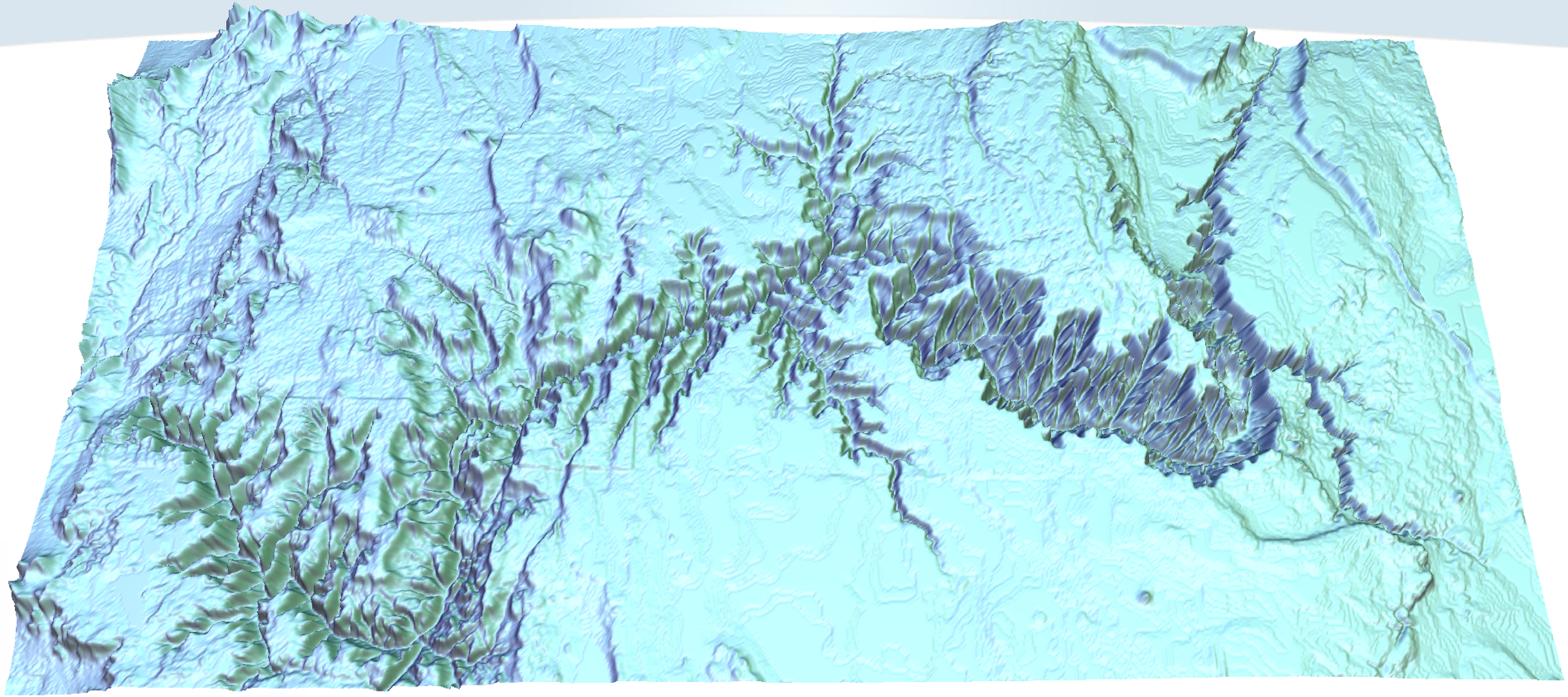


Elevation data-sets

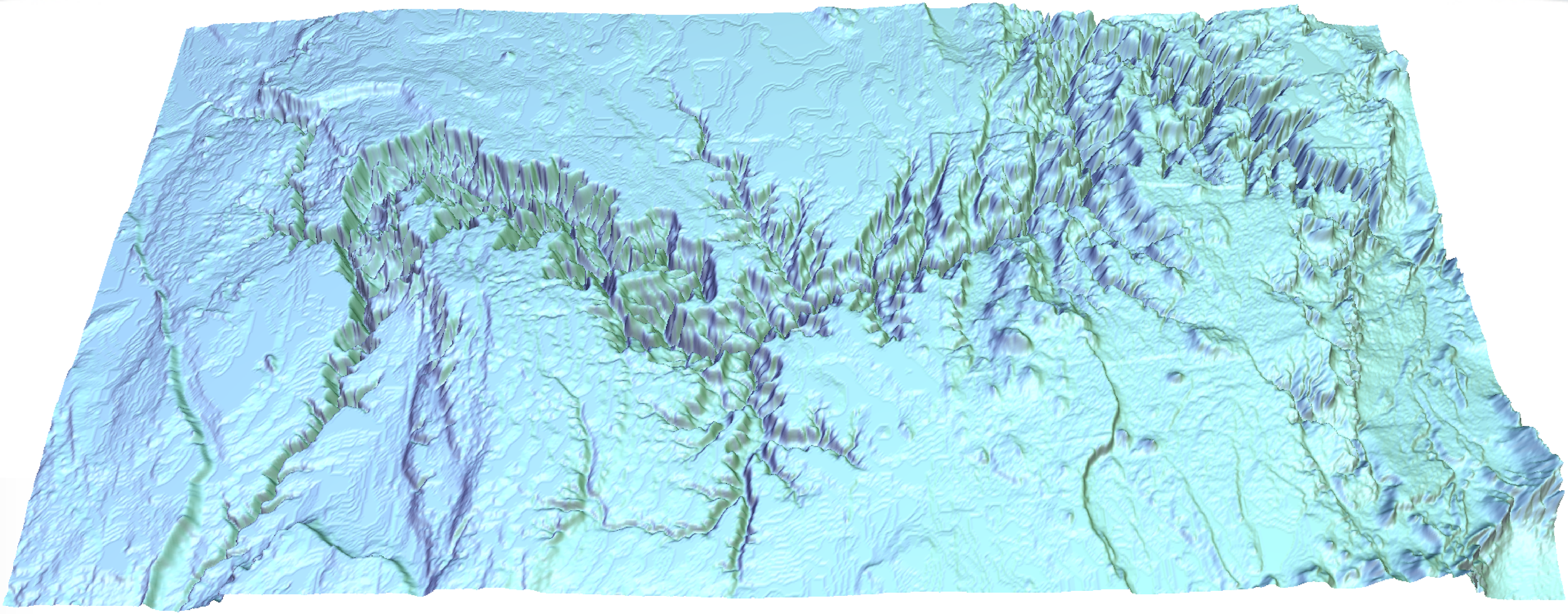
Elevation data-sets



Elevation data-sets

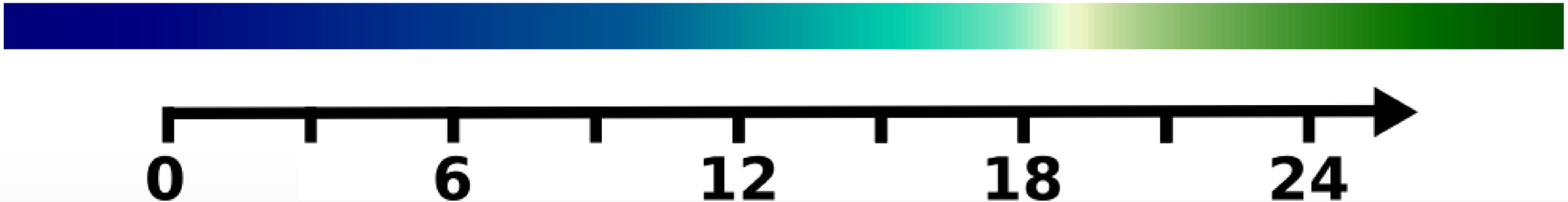


Elevation data-sets



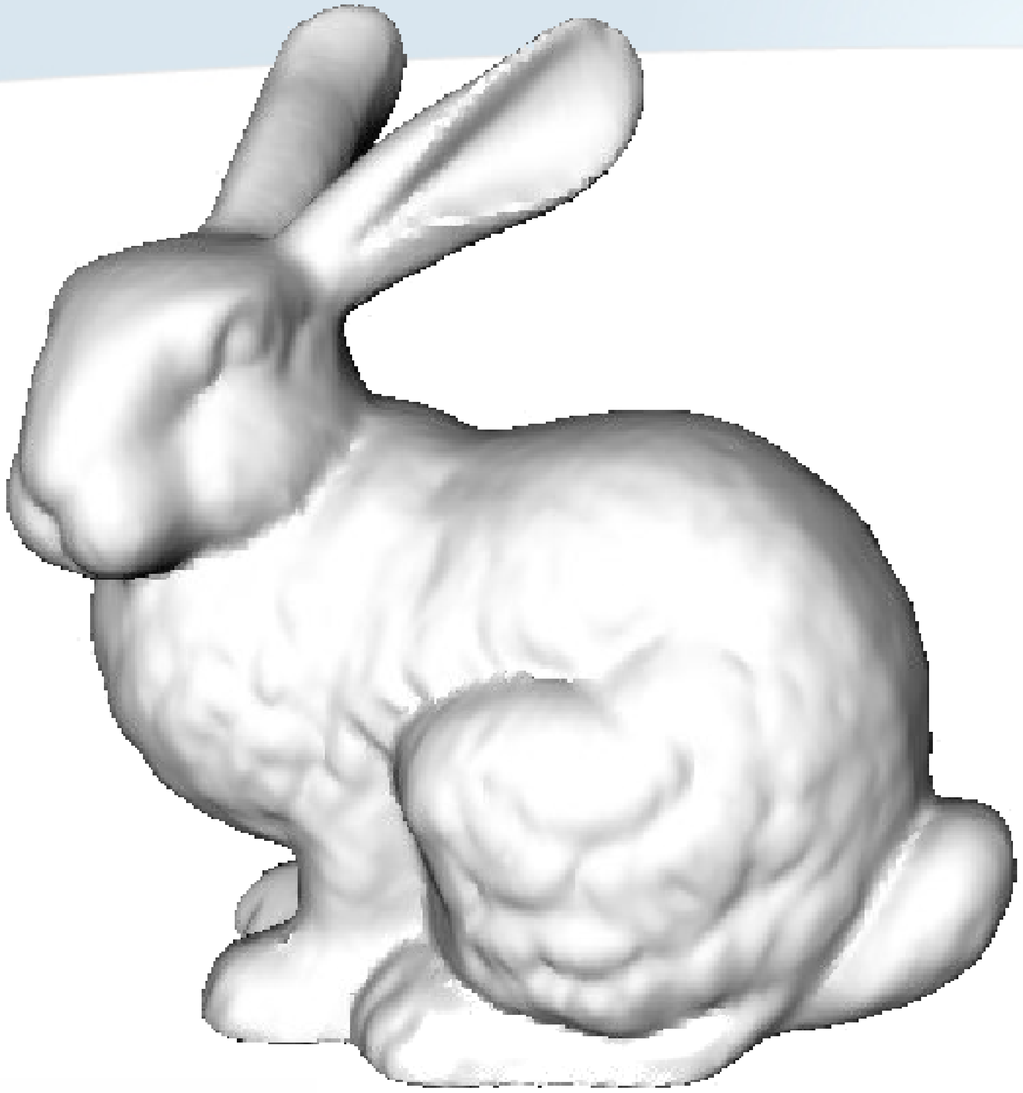
Additional information channels

- Domain with an embedding to $\mathbb{R}^3$
  - How can we encode scalar values?
- Intuitive idea
  - Use a perceptual channel **Color maps**
    - Arrange continuously a palette of colors
    - Map it to the real line $(\mathcal{A})$

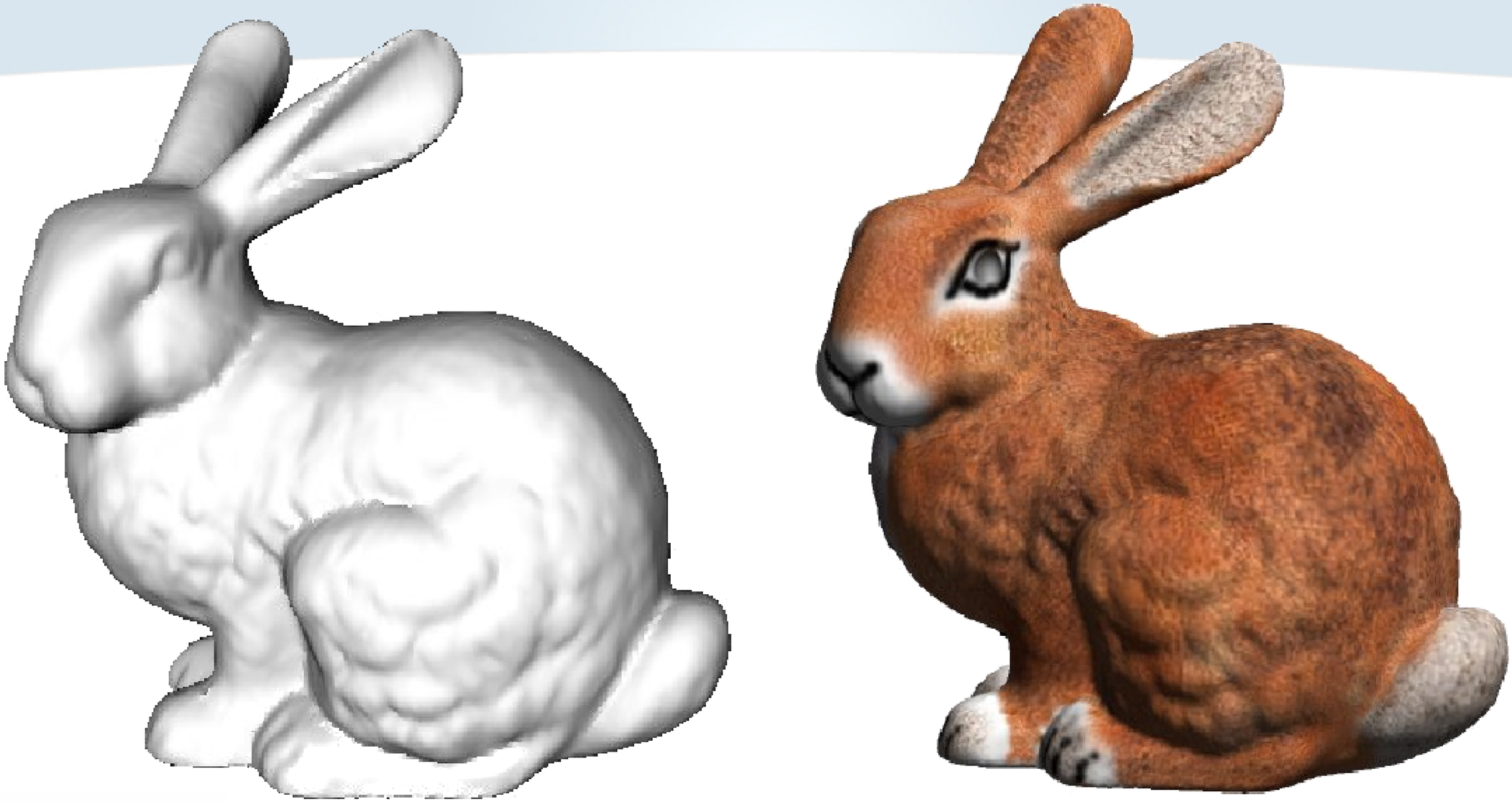


Recap on texture mapping

Recap on texture mapping



Recap on texture mapping



Recap on texture mapping

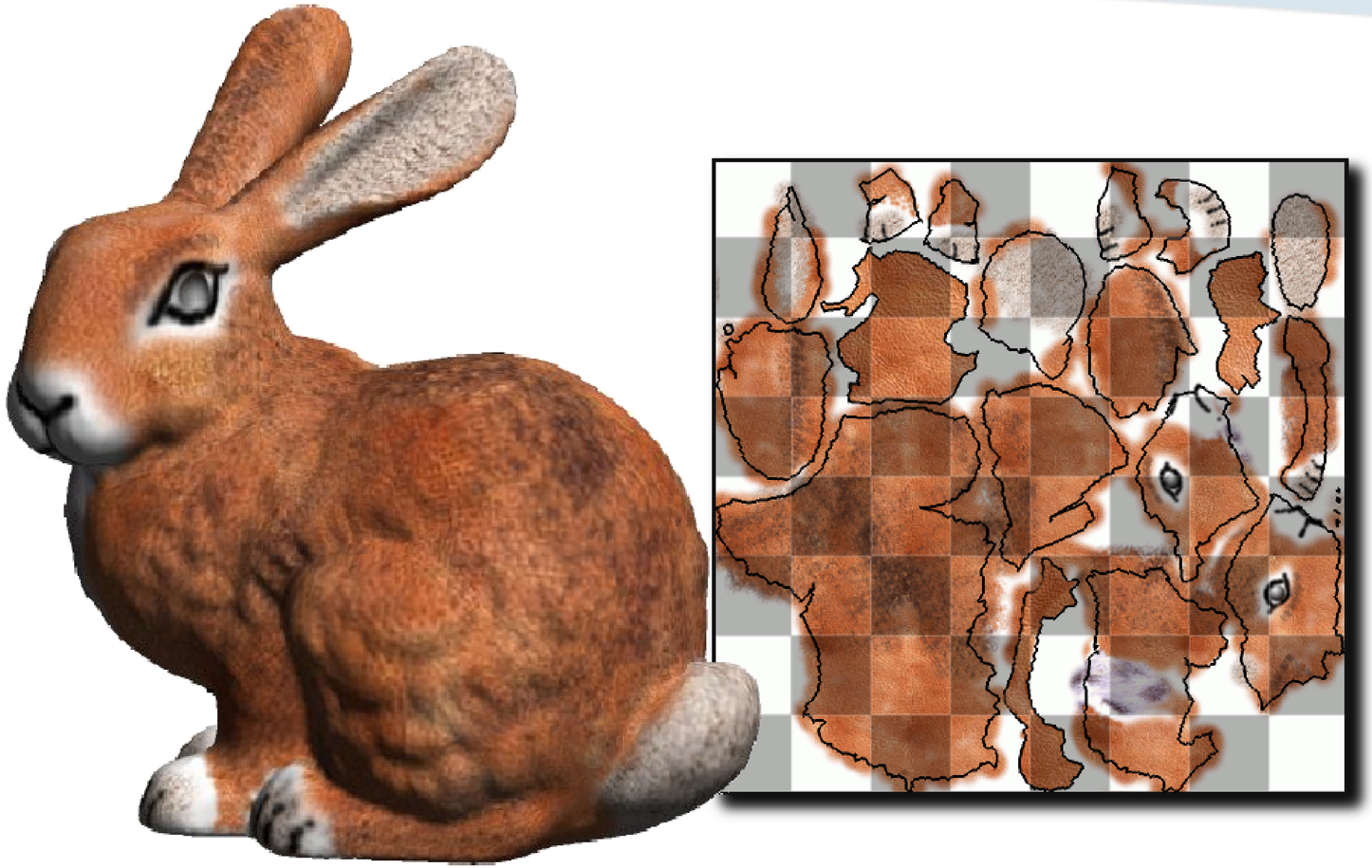
- Surface parameterization

Recap on texture mapping

- Surface parameterization
  - Bijective map
  - $\phi : \mathcal{S}_i \subset \mathcal{S} \rightarrow \mathcal{D} \subset \mathbb{R}^2$

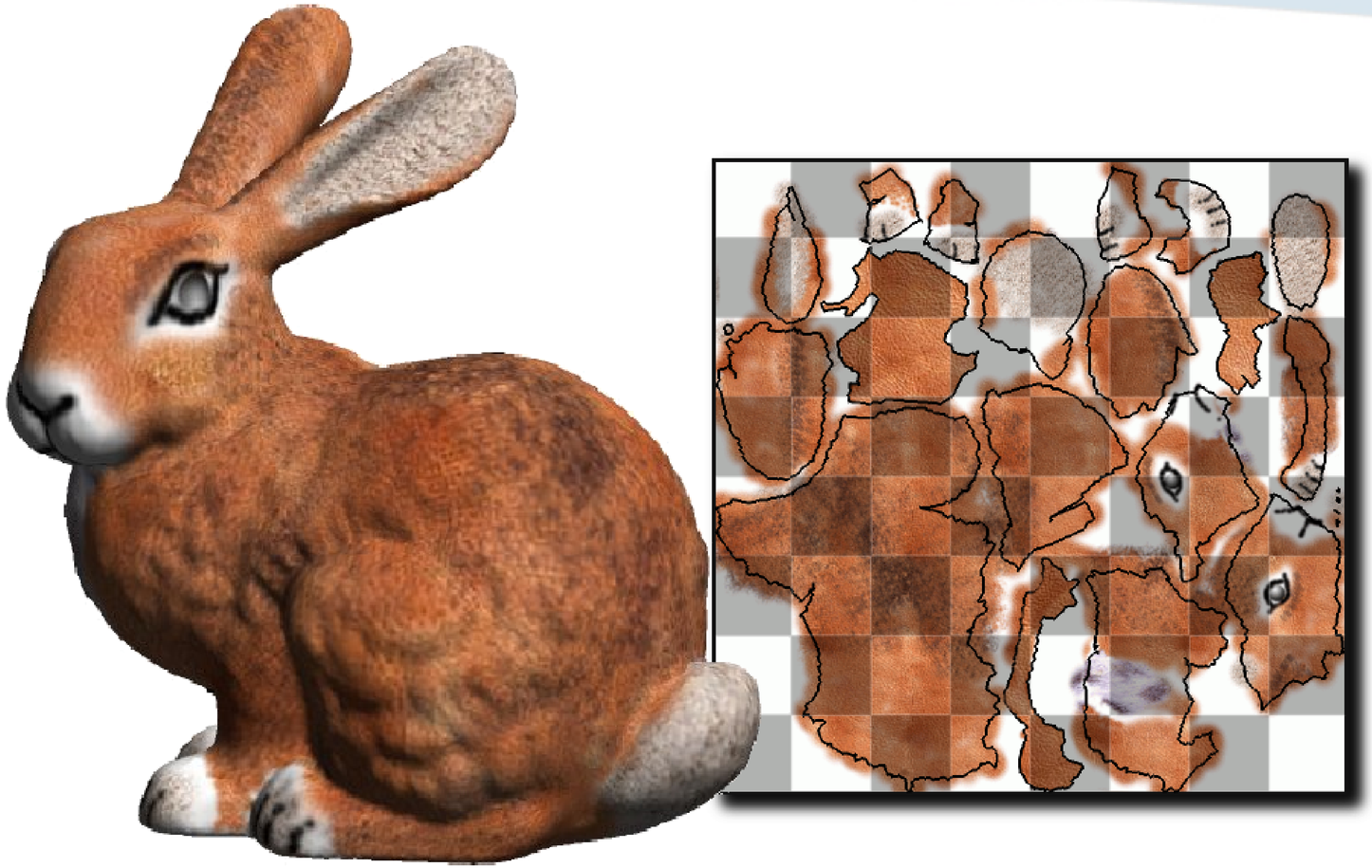
Recap on texture mapping

- Surface parameterization
  - Bijective map
  - $\phi : \mathcal{S}_i \subset \mathcal{S} \rightarrow \mathcal{D} \subset \mathbb{R}^2$



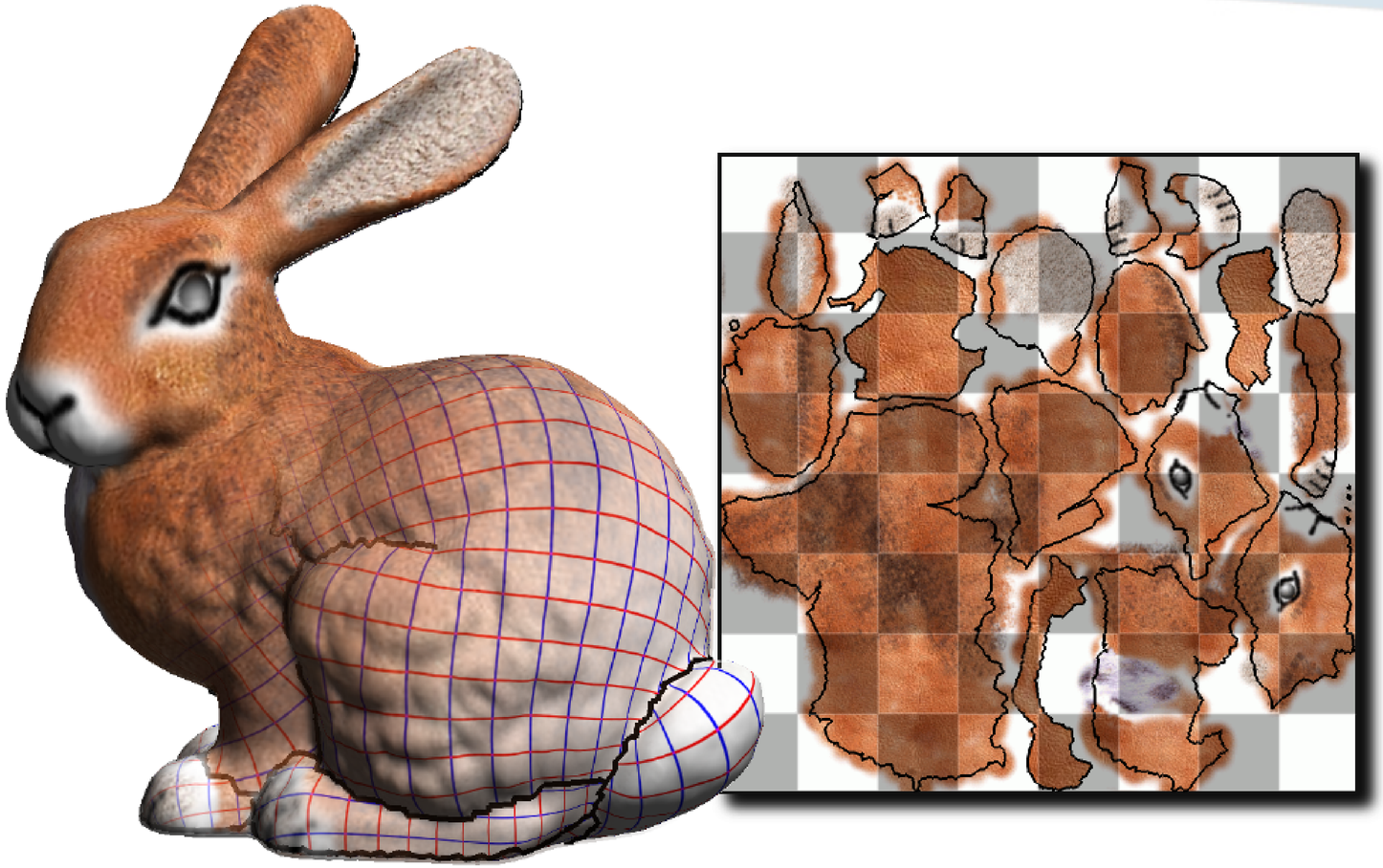
Recap on texture mapping

- Surface parameterization
  - Bijective map
  - $\phi : \mathcal{S}_i \subset \mathcal{S} \rightarrow \mathcal{D} \subset \mathbb{R}^2$
  - $\forall p \in \mathcal{S}_i \quad \phi(p) = (s, t)$



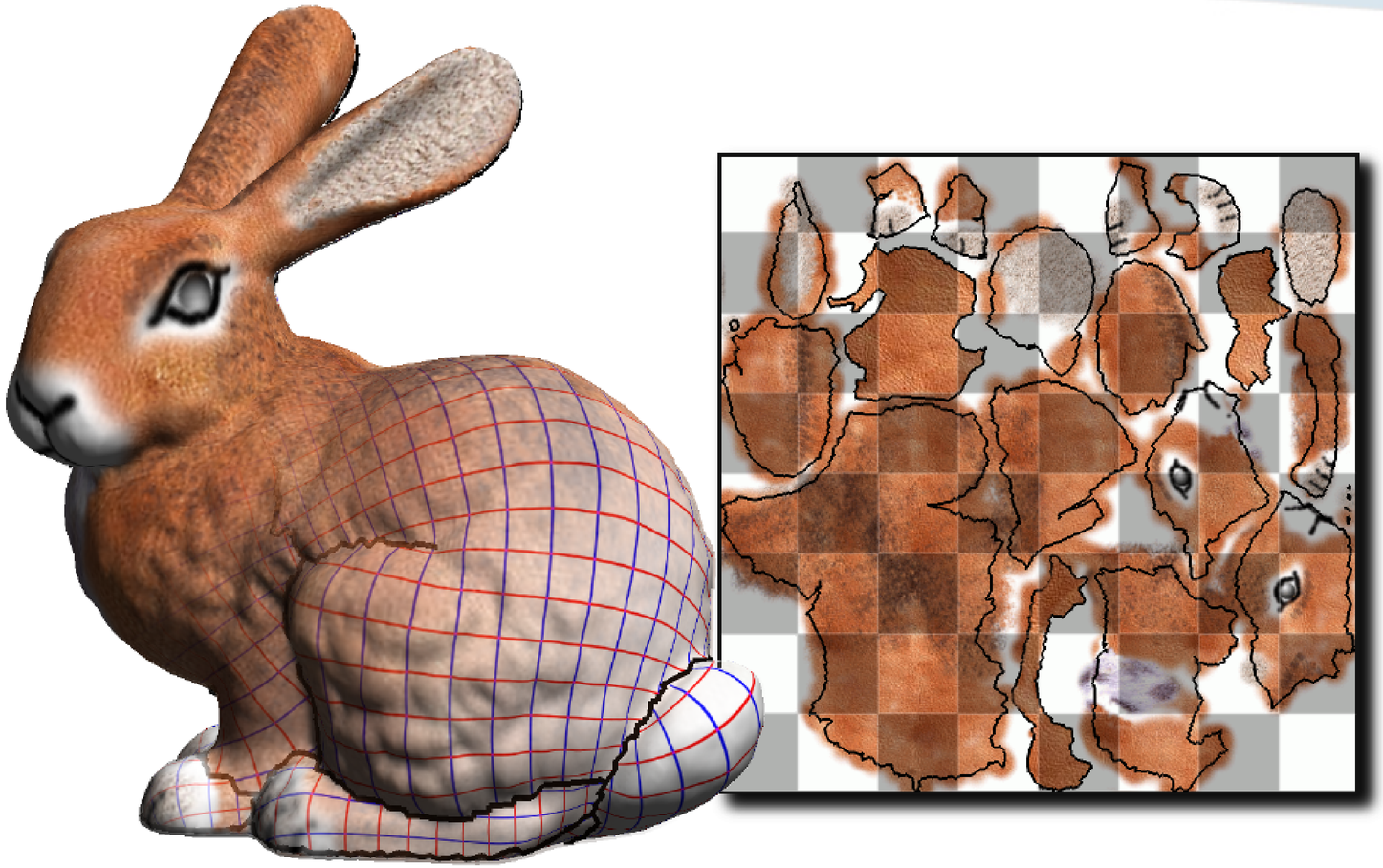
Recap on texture mapping

- Surface parameterization
  - Bijective map
  - $\phi : \mathcal{S}_i \subset \mathcal{S} \rightarrow \mathcal{D} \subset \mathbb{R}^2$
  - $\forall p \in \mathcal{S}_i \quad \phi(p) = (s, t)$



Recap on texture mapping

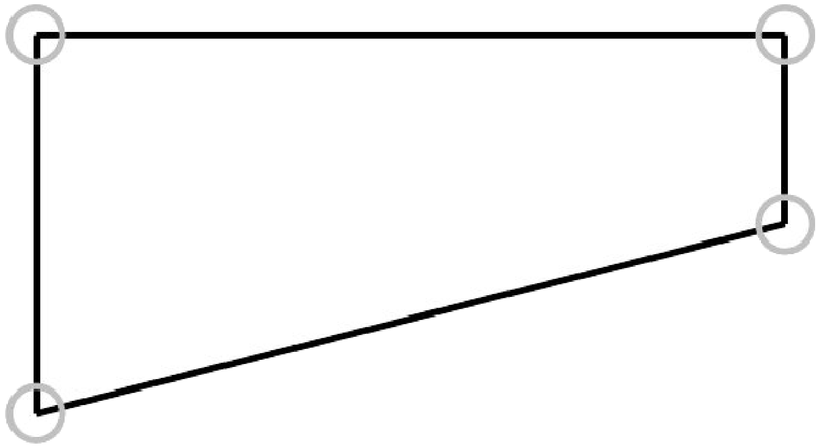
- Surface parameterization
  - Bijective map
  - $\phi : \mathcal{S}_i \subset \mathcal{S} \rightarrow \mathcal{D} \subset \mathbb{R}^2$
  - $\forall p \in \mathcal{S}_i \quad \phi(p) = (s, t)$
- Each vertex:
  - Embedding in $\mathbb{R}^3$
    - (x, y, z)
  - Unfolding to $\mathbb{R}^2$
    - (s, t)



Recap on texture mapping

- Each vertex:
  - Embedding in $\mathbb{R}^3$
    - (x, y, z)
  - Unfolding to $\mathbb{R}^2$
    - (s, t)

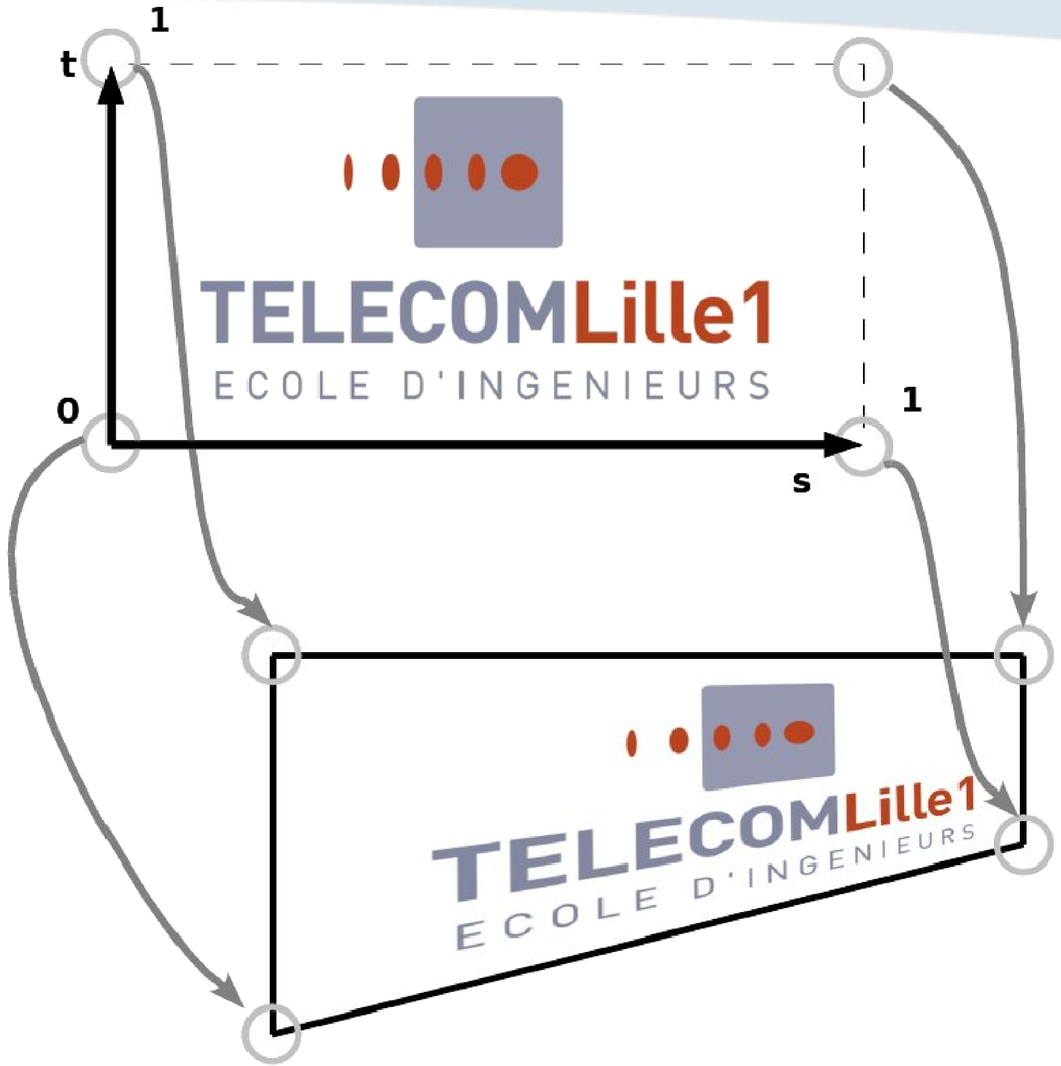
- What about the interior of the polygons?



Recap on texture mapping

- Each vertex:
  - Embedding in $\mathbb{R}^3$
    - (x, y, z)
  - Unfolding to $\mathbb{R}^2$
    - (s, t)

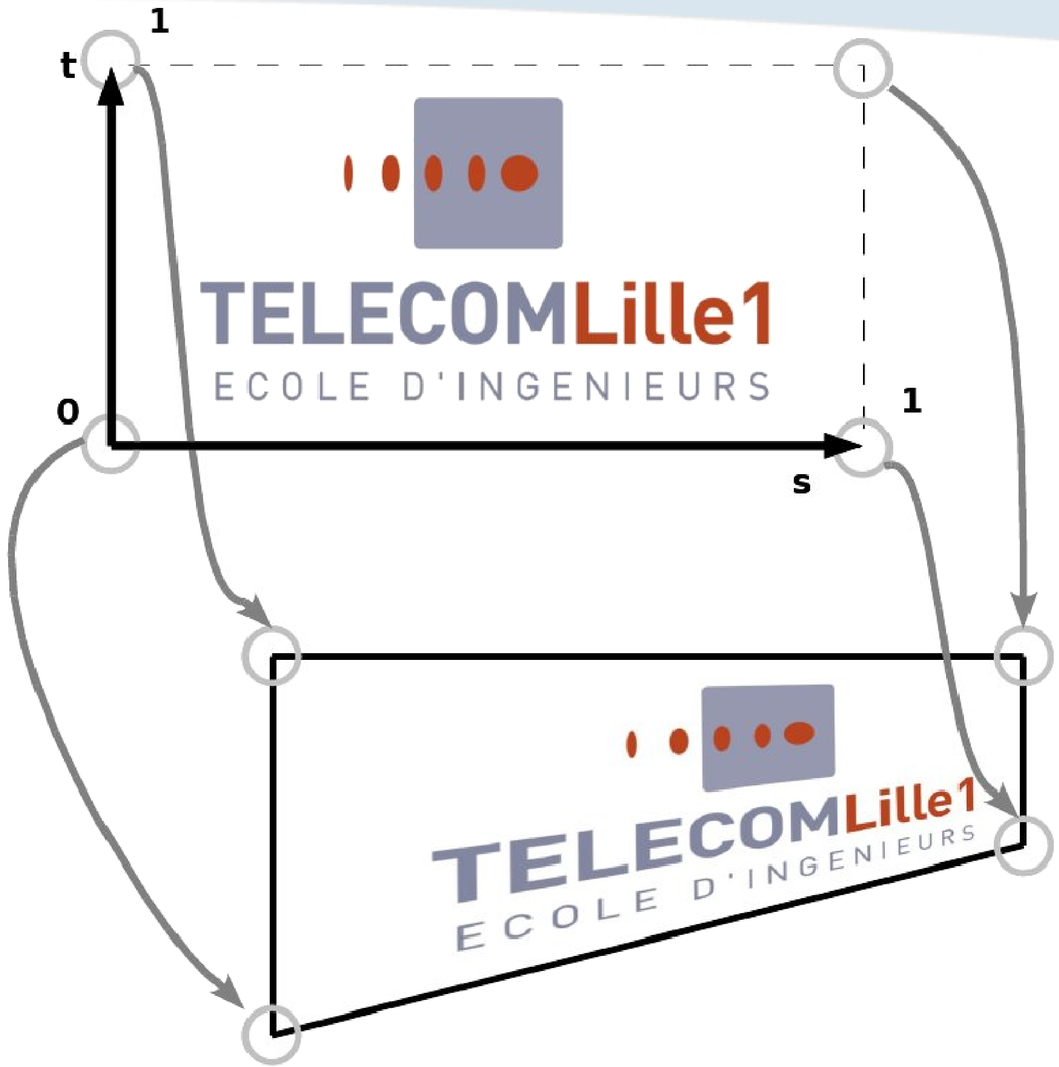
- What about the interior of the polygons?



Recap on texture mapping

- Each vertex:
  - Embedding in $\mathbb{R}^3$
    - (x, y, z)
  - Unfolding to $\mathbb{R}^2$
    - (s, t)

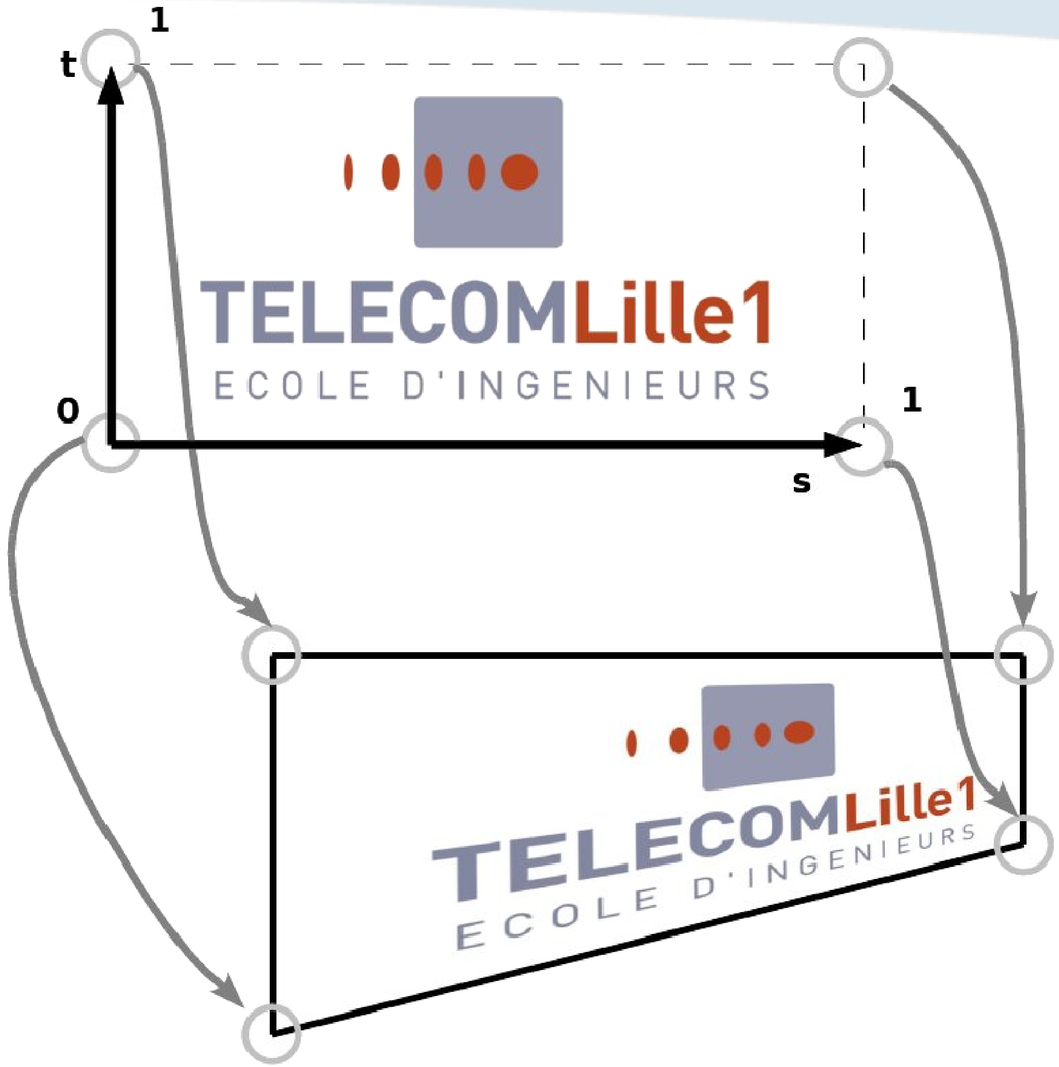
- What about the interior of the polygons?
  - **Linear interpolation**



Recap on texture mapping

- Each vertex:
  - Embedding in $\mathbb{R}^3$
    - (x, y, z)
  - Unfolding to $\mathbb{R}^2$
    - (s, t)

- What about the interior of the polygons?
  - **Linear interpolation** (GPU)



Color maps for scalar field visualization

- Arrange continuously a palette of colors

Color maps for scalar field visualization

- Arrange continuously a palette of colors

Color maps for scalar field visualization

- Arrange continuously a palette of colors

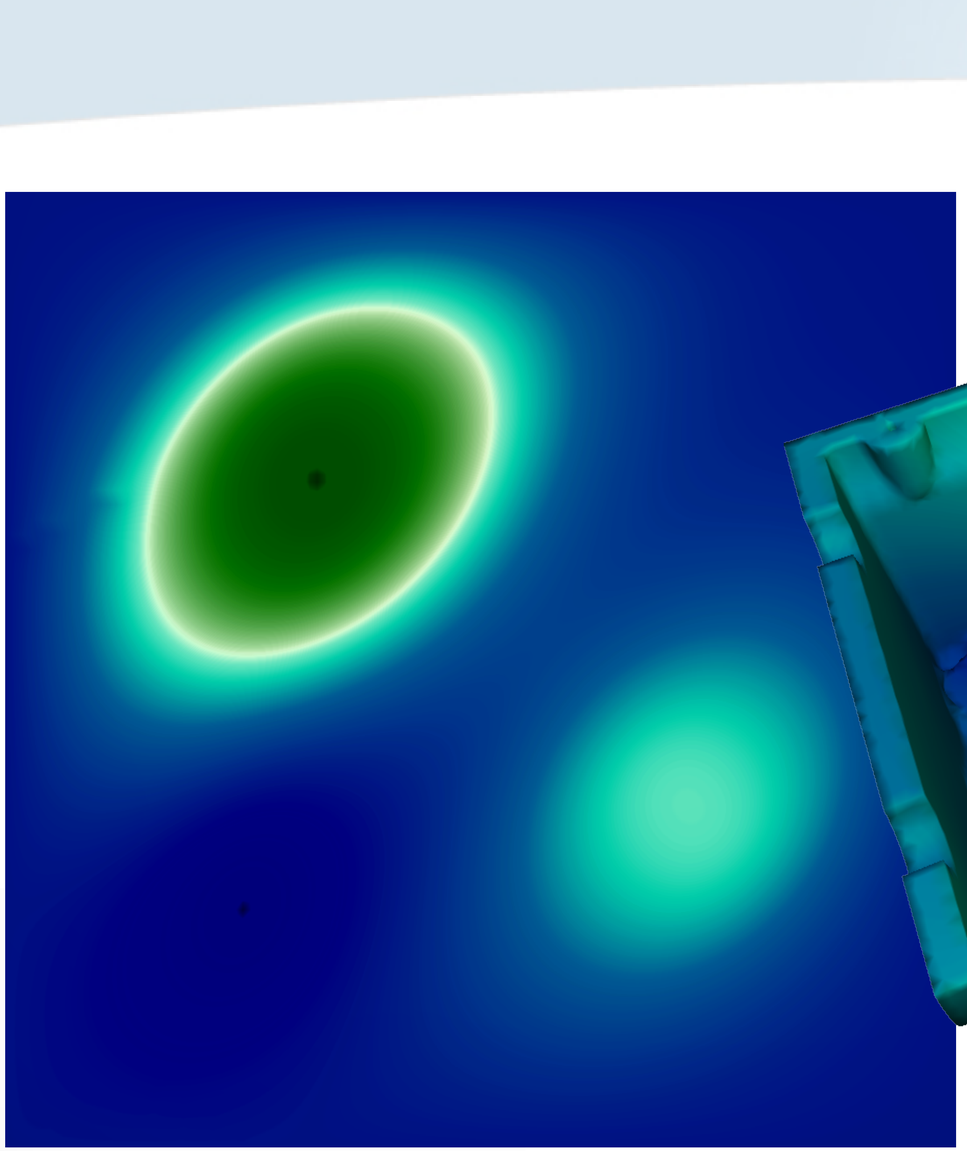
- For each vertex: (break the bijection)
  - $\phi(v) = (f(v), 0)$

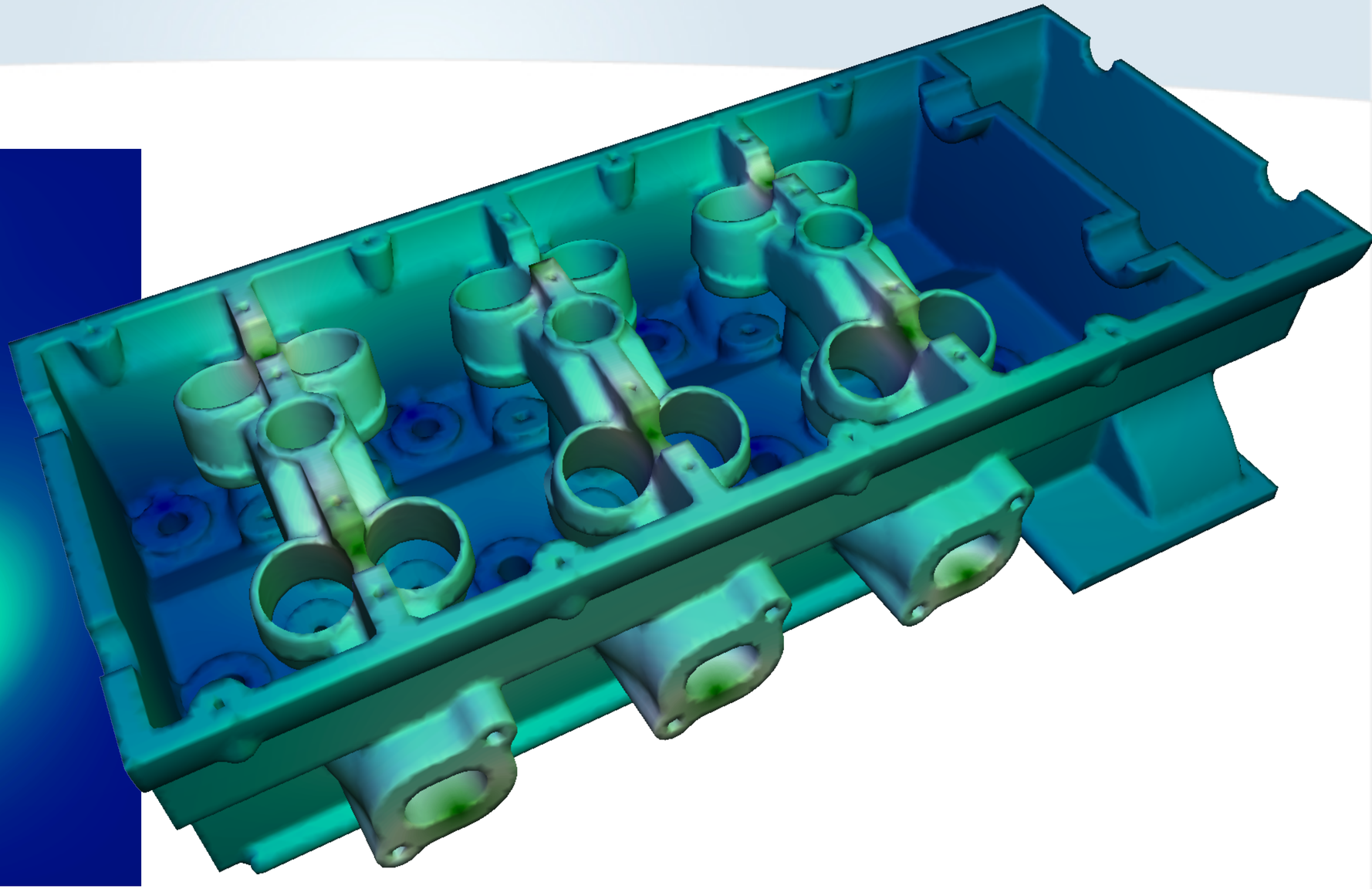
Color maps for scalar field visualization

- Arrange continuously a palette of colors

- For each vertex: (break the bijection)
  - $\phi(v) = (f(v), 0)$
  - Generic solution (arbitrary texture)

Color maps for scalar field visualization





Color maps for scalar field visualization

Color maps for scalar field visualization

- Continuously varying palettes
  - Difficult to evaluate variation

Color maps for scalar field visualization

- Continuously varying palettes
  - Difficult to evaluate variation

- Notion of level set
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$

Color maps for scalar field visualization

- Continuously varying palettes
  - Difficult to evaluate variation
- Notion of level set
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
  - **Global** and **local** information

Color maps for scalar field visualization

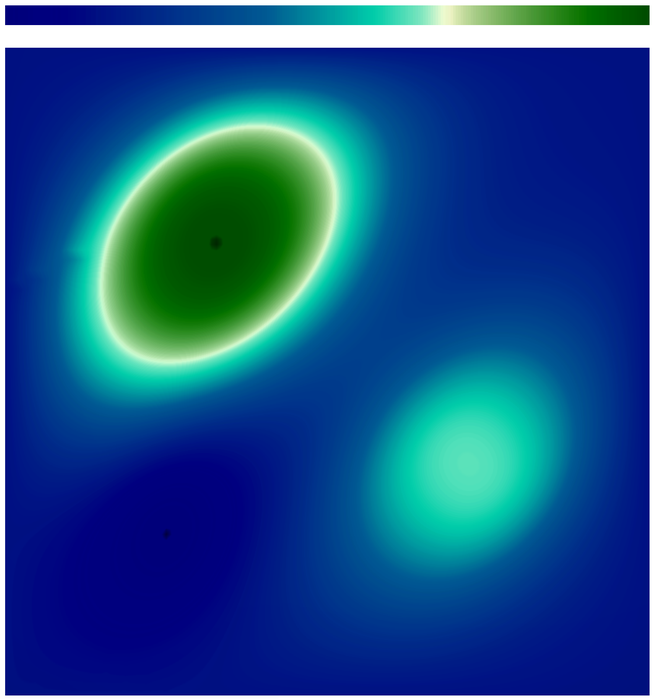
- Continuously varying palettes
  - Difficult to evaluate variation

- Notion of level set
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
  - **Global** and **local** information

- Explicit computation
  - A bit involved

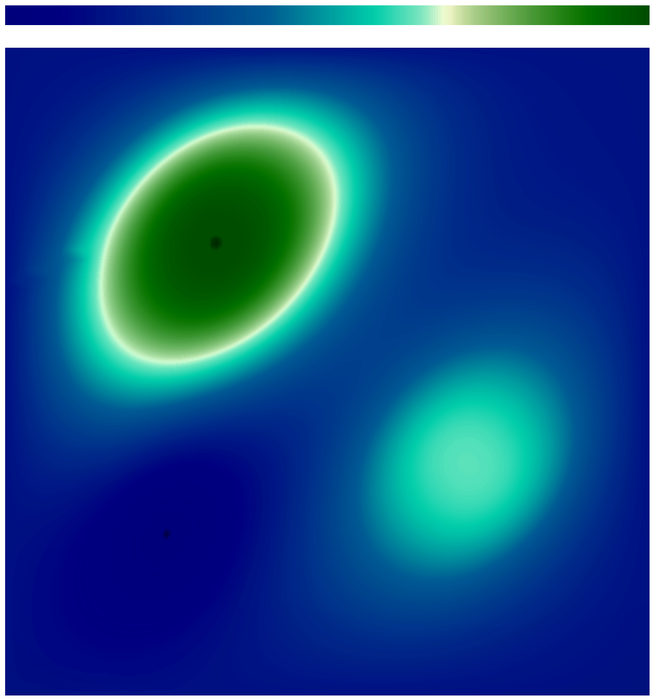
Color maps for scalar field visualization

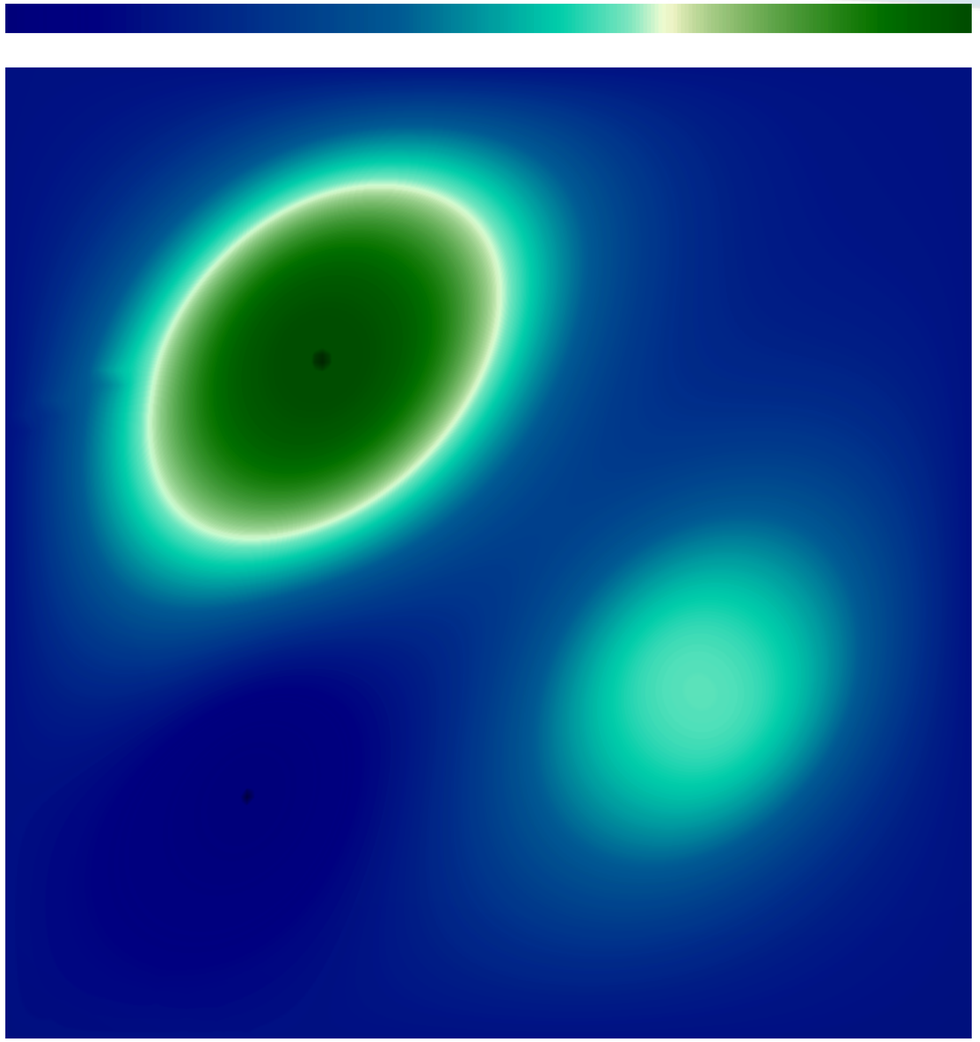
- Implicit level set visualization
  - Play with the texture?



Color maps for scalar field visualization

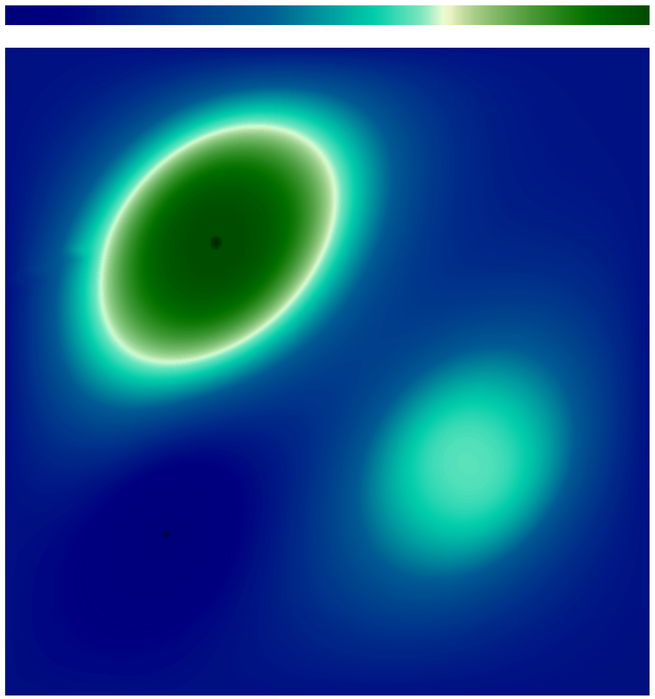
- Implicit level set visualization
  - Play with the texture?

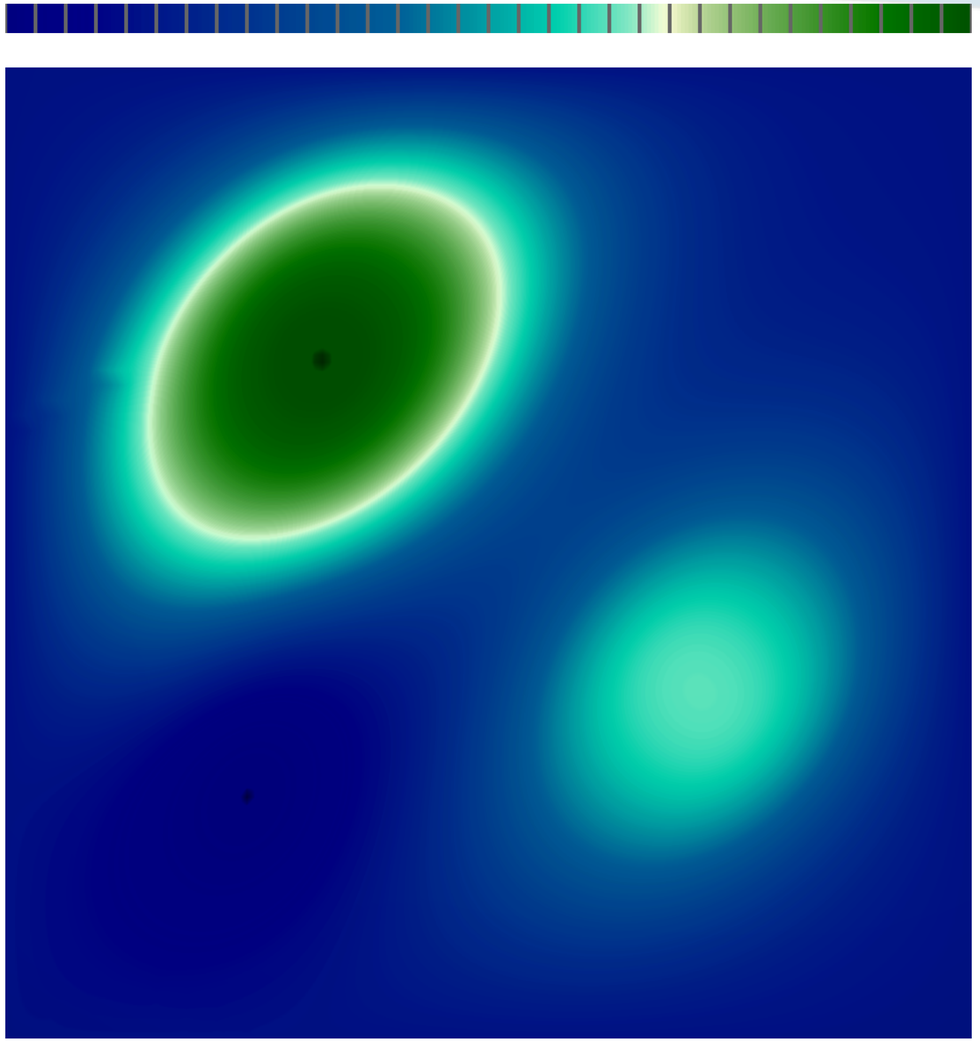




Color maps for scalar field visualization

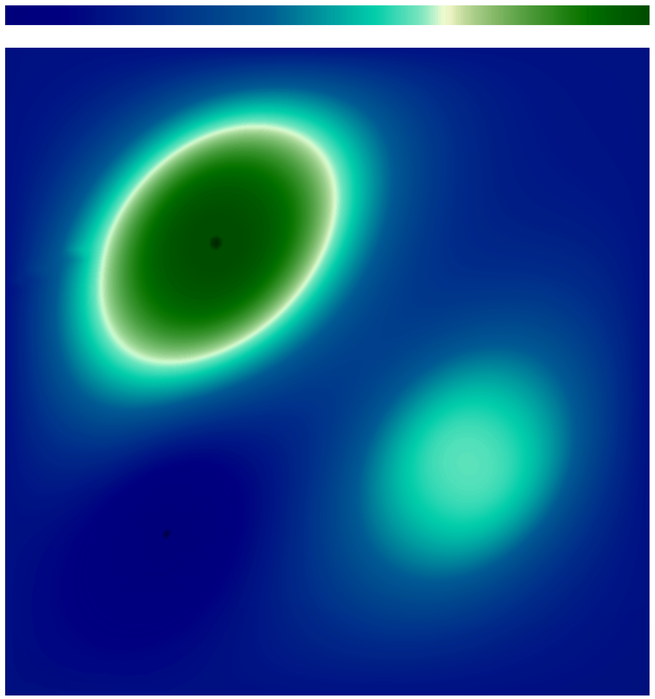
- Implicit level set visualization
  - Play with the texture?

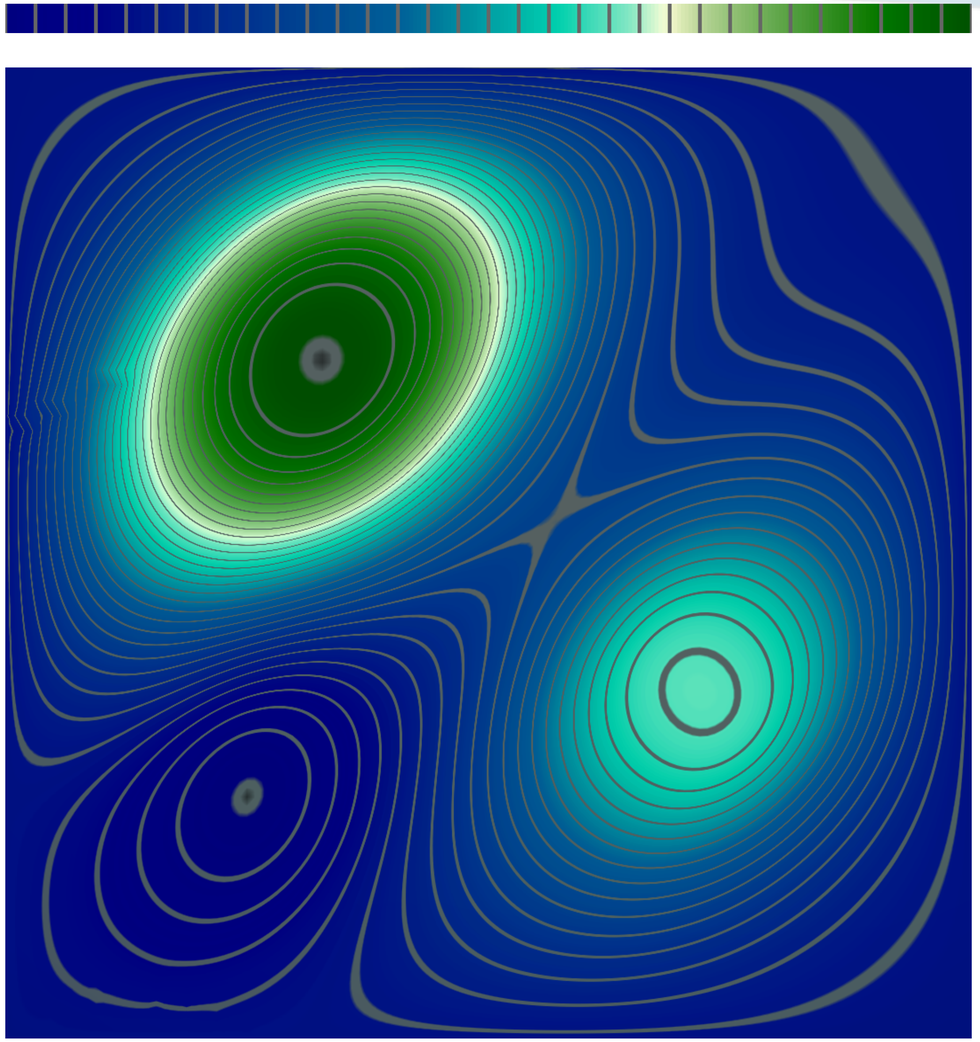




Color maps for scalar field visualization

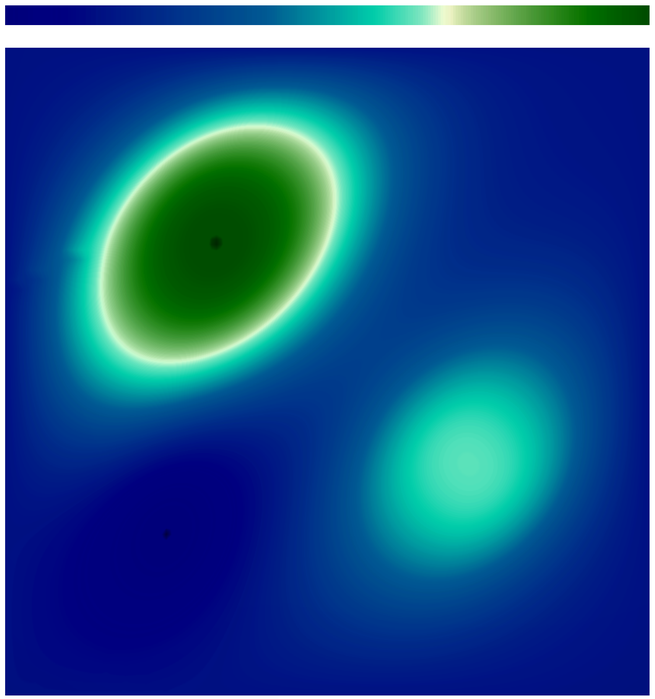
- Implicit level set visualization
  - Play with the texture?

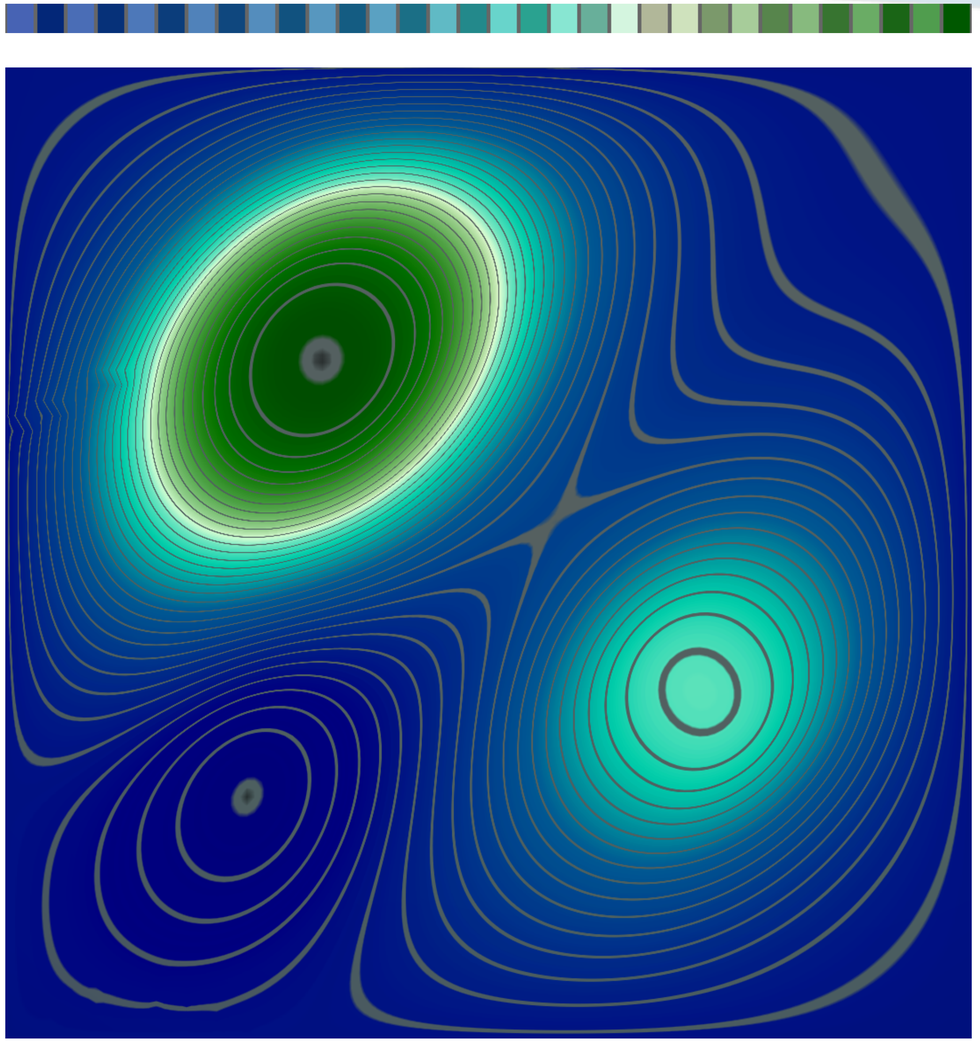




Color maps for scalar field visualization

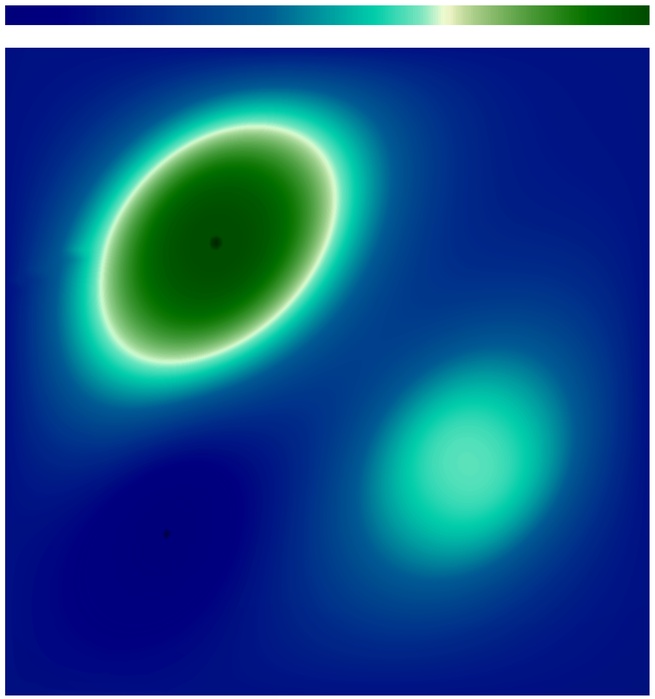
- Implicit level set visualization
  - Play with the texture?

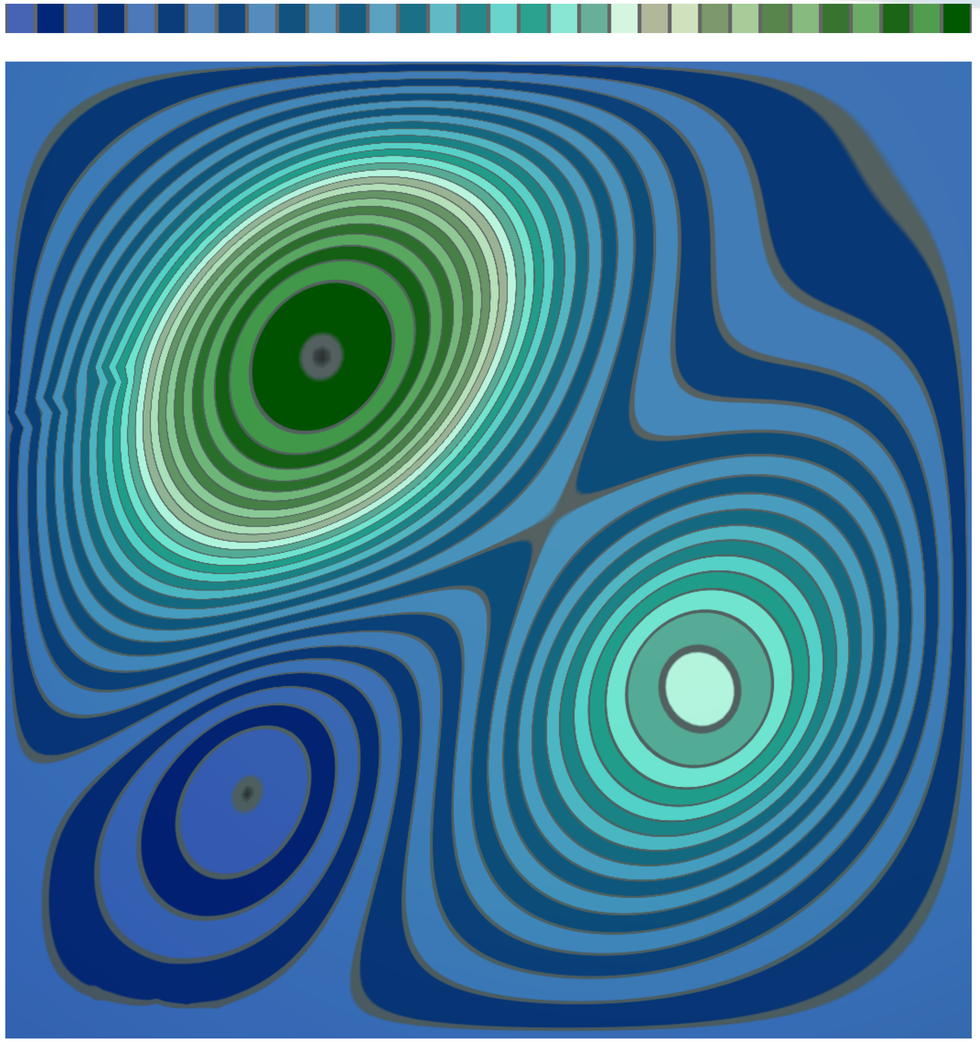




Color maps for scalar field visualization

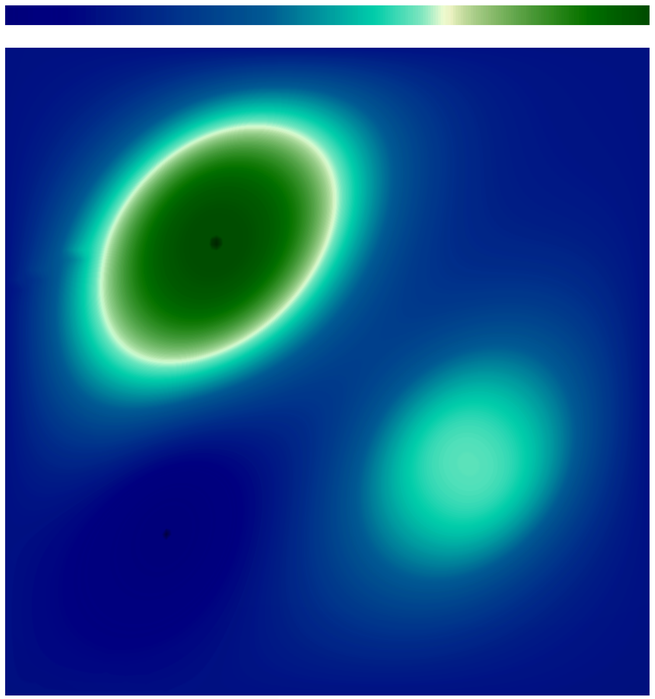
- Implicit level set visualization
  - Play with the texture?

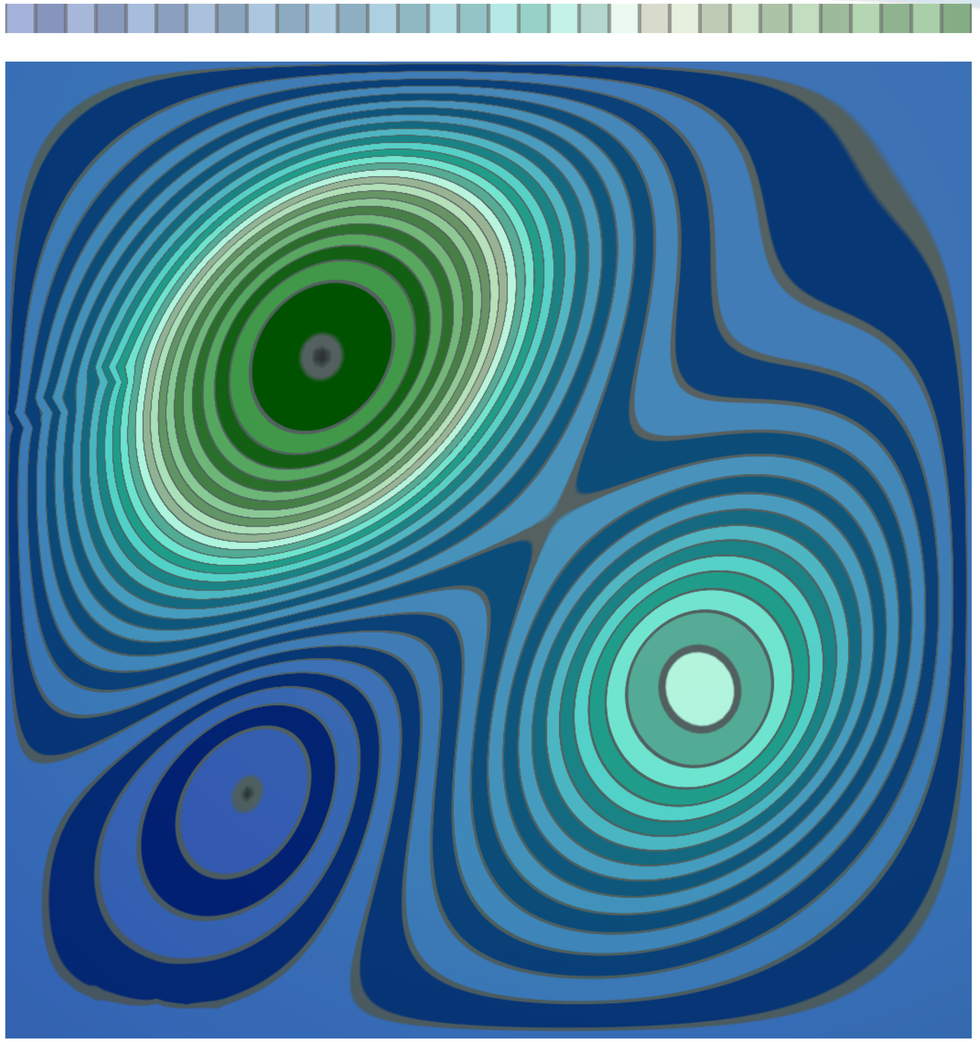




Color maps for scalar field visualization

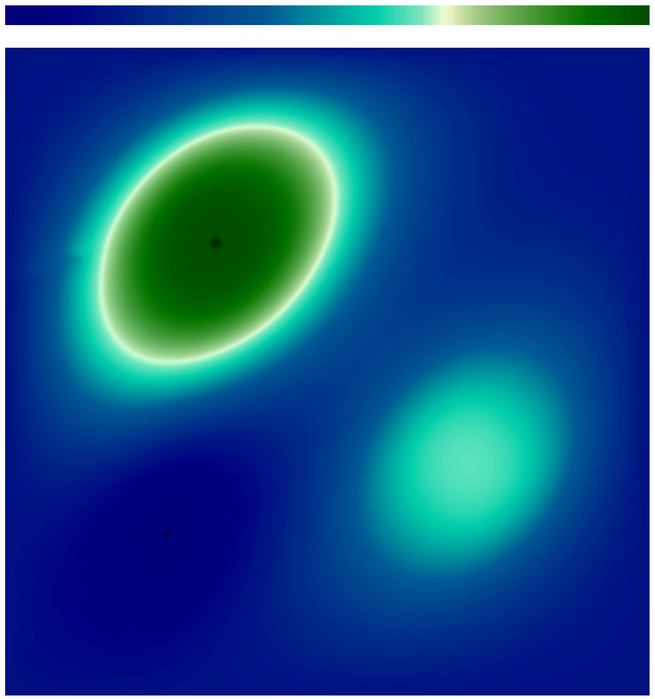
- Implicit level set visualization
  - Play with the texture?

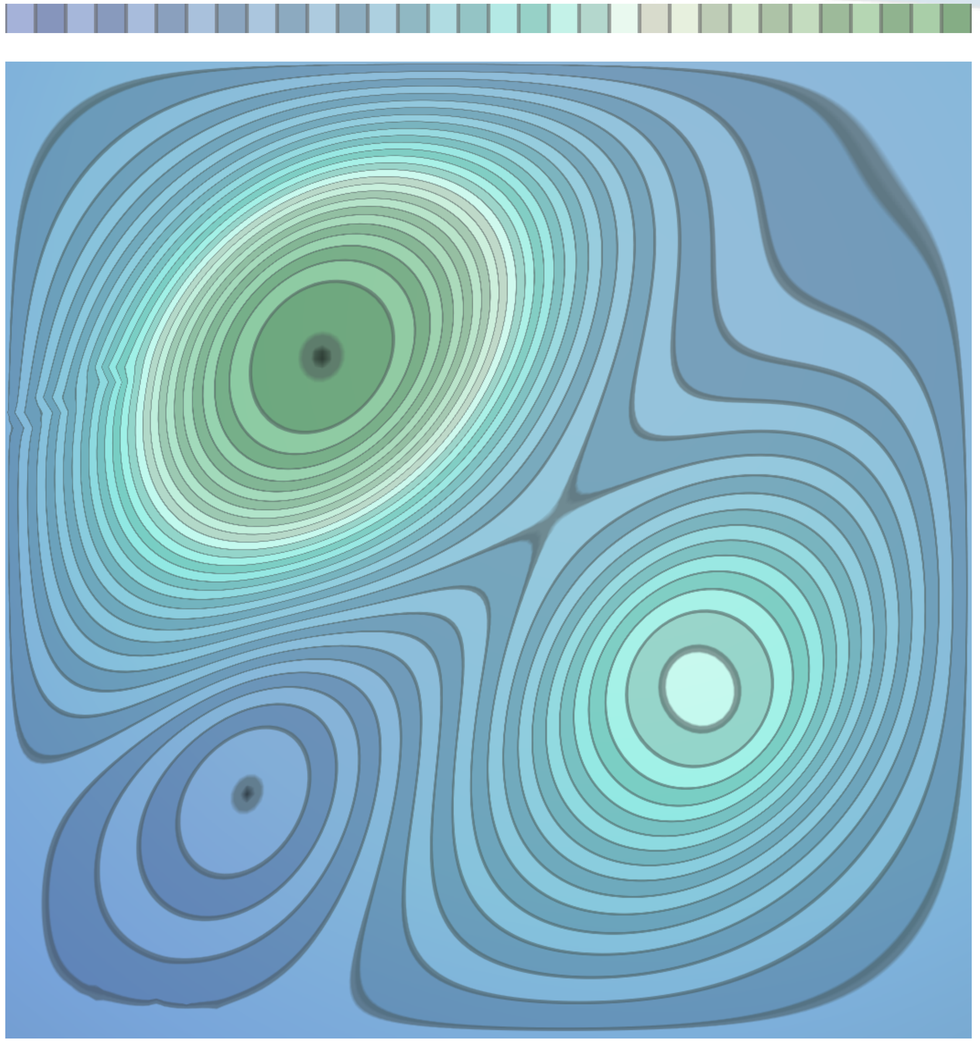




Color maps for scalar field visualization

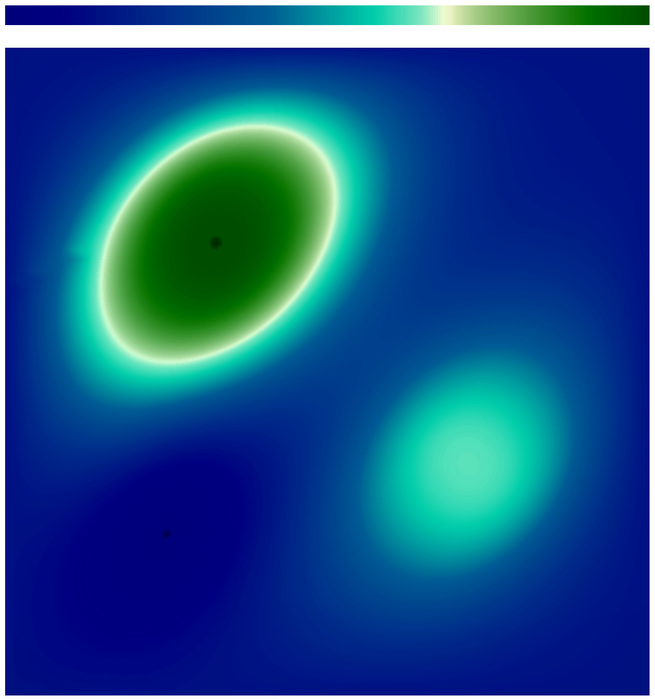
- Implicit level set visualization
  - Play with the texture?

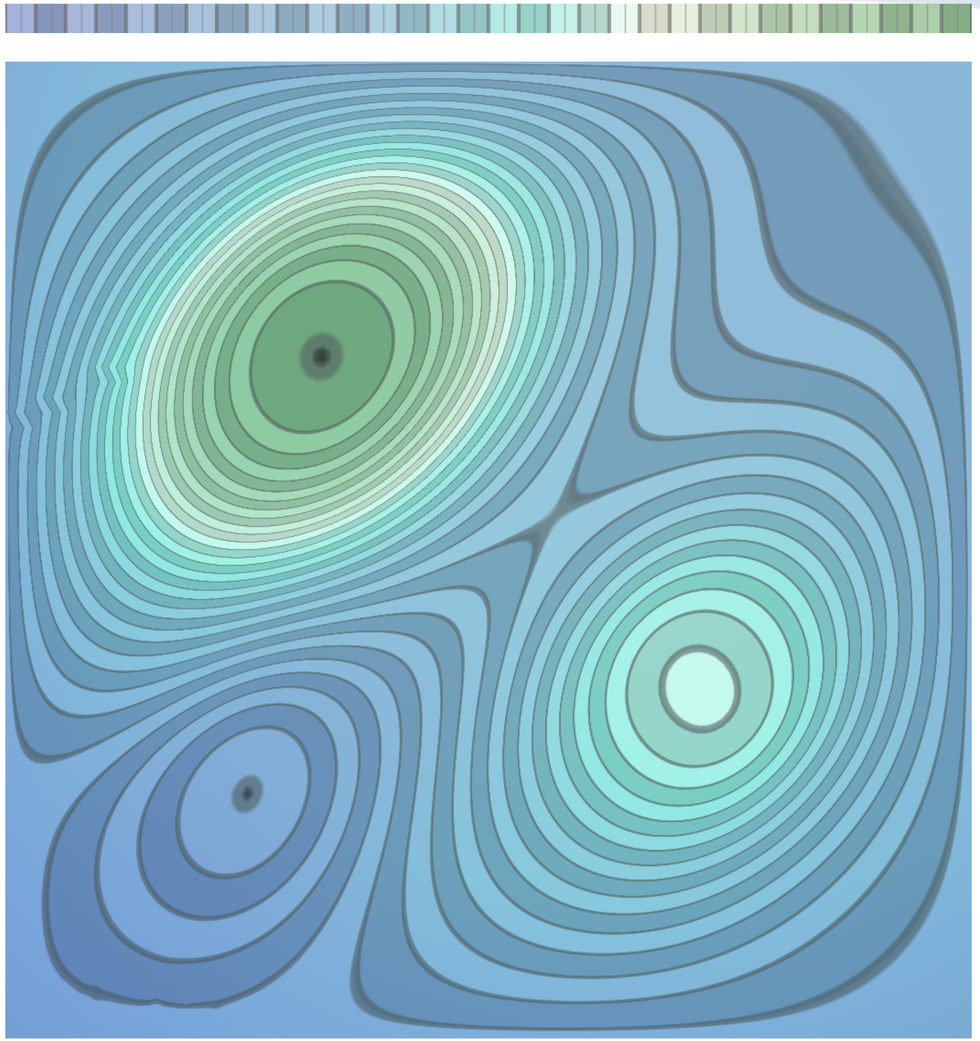




Color maps for scalar field visualization

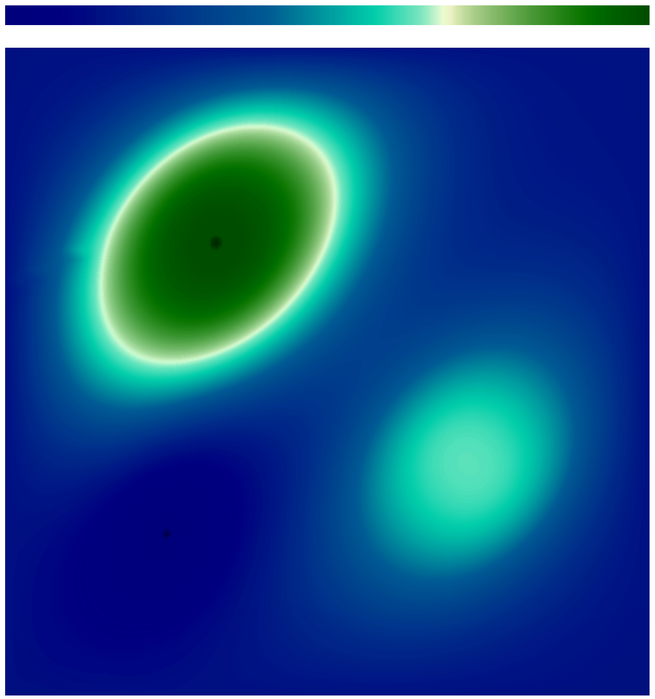
- Implicit level set visualization
  - Play with the texture?

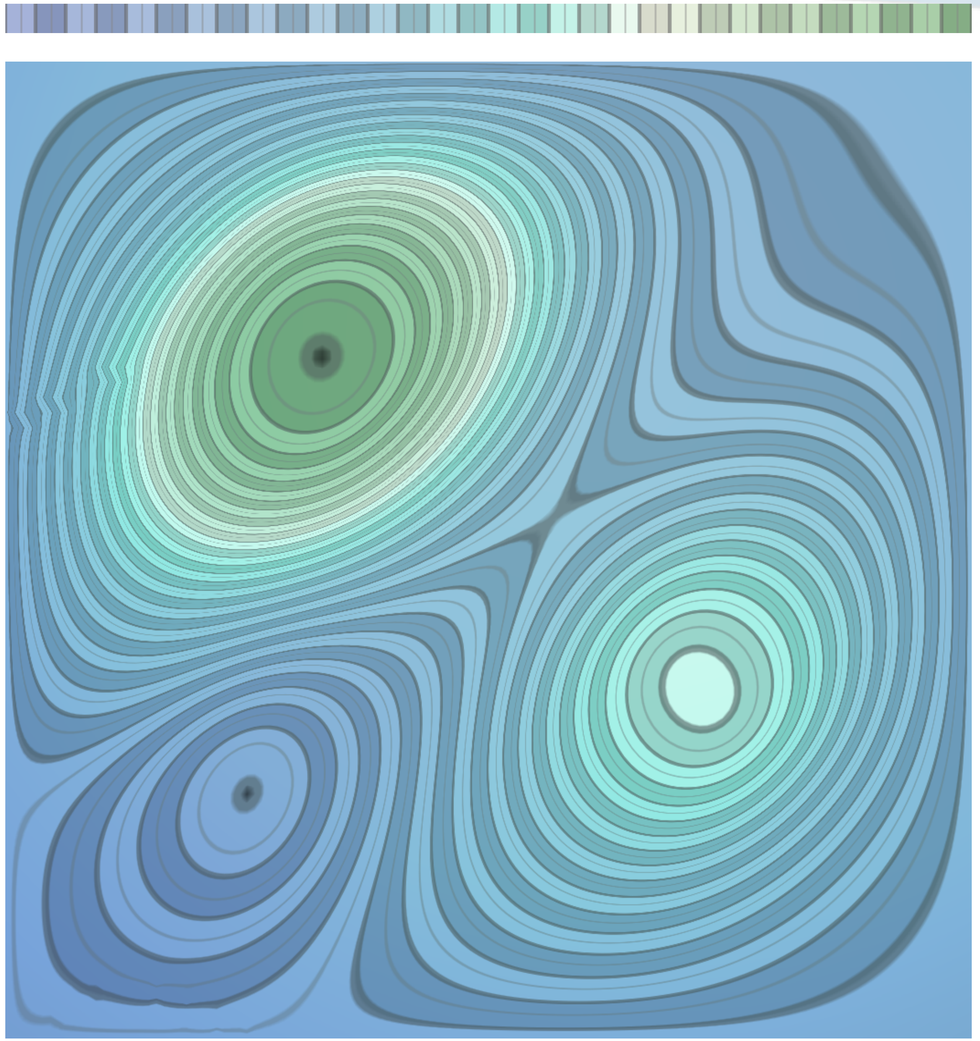




Color maps for scalar field visualization

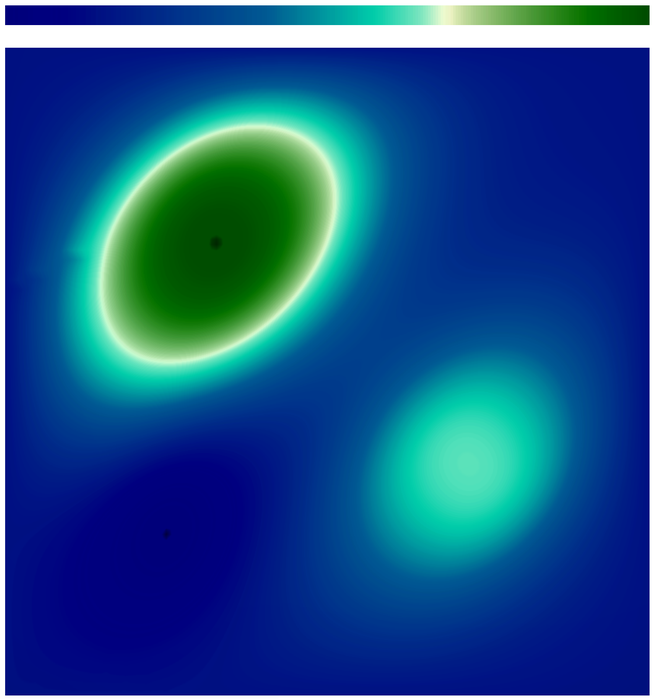
- Implicit level set visualization
  - Play with the texture?

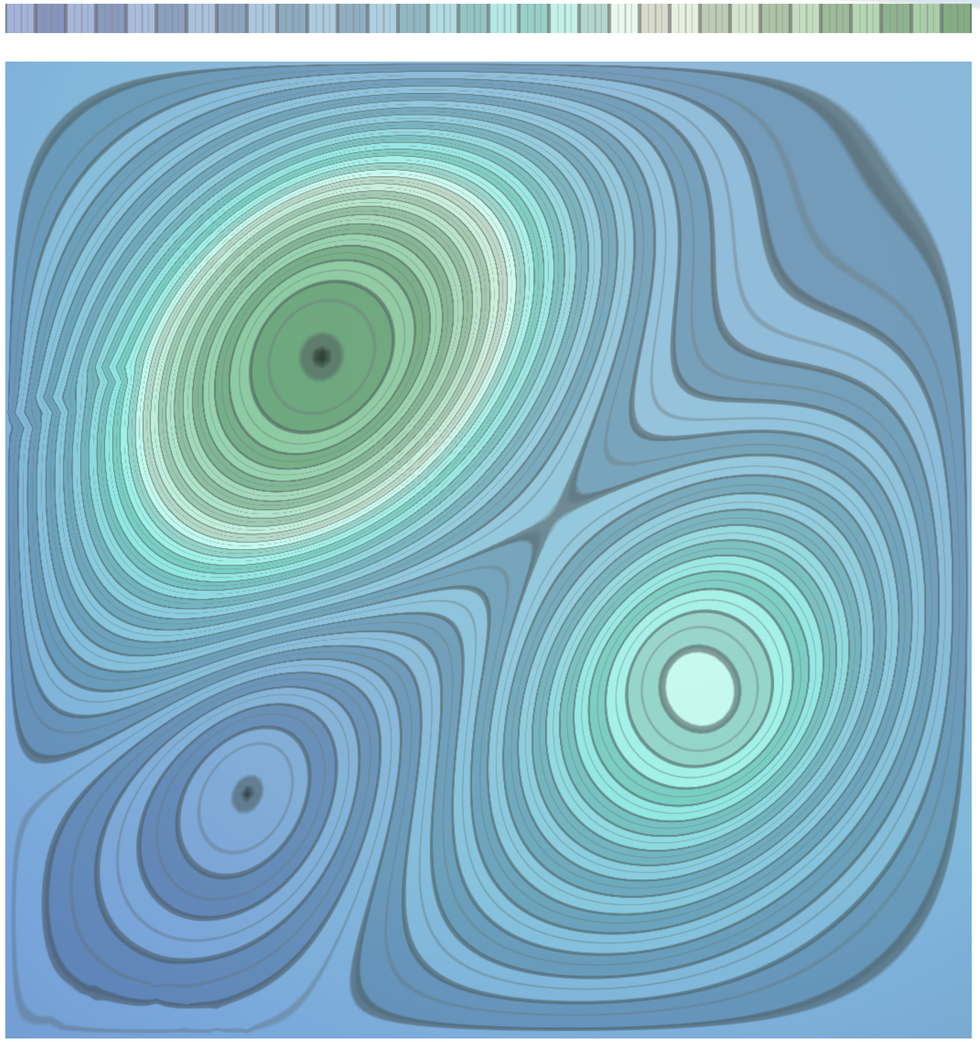




Color maps for scalar field visualization

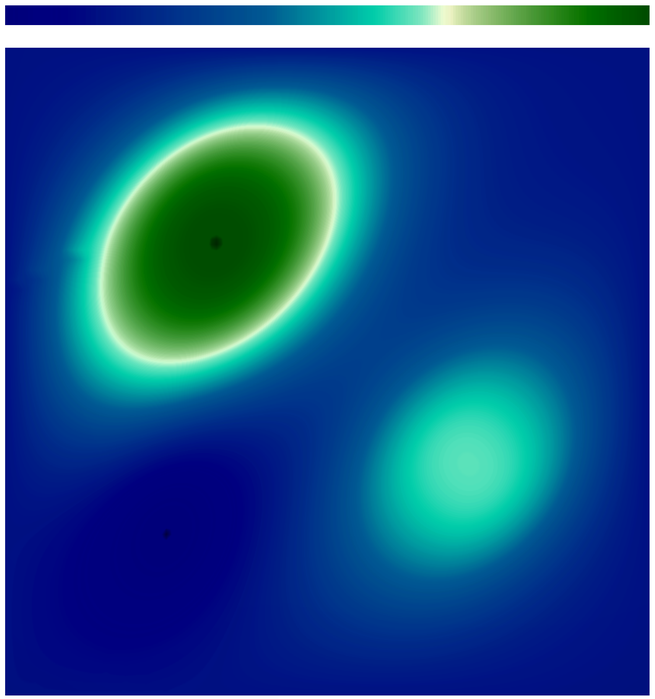
- Implicit level set visualization
  - Play with the texture?

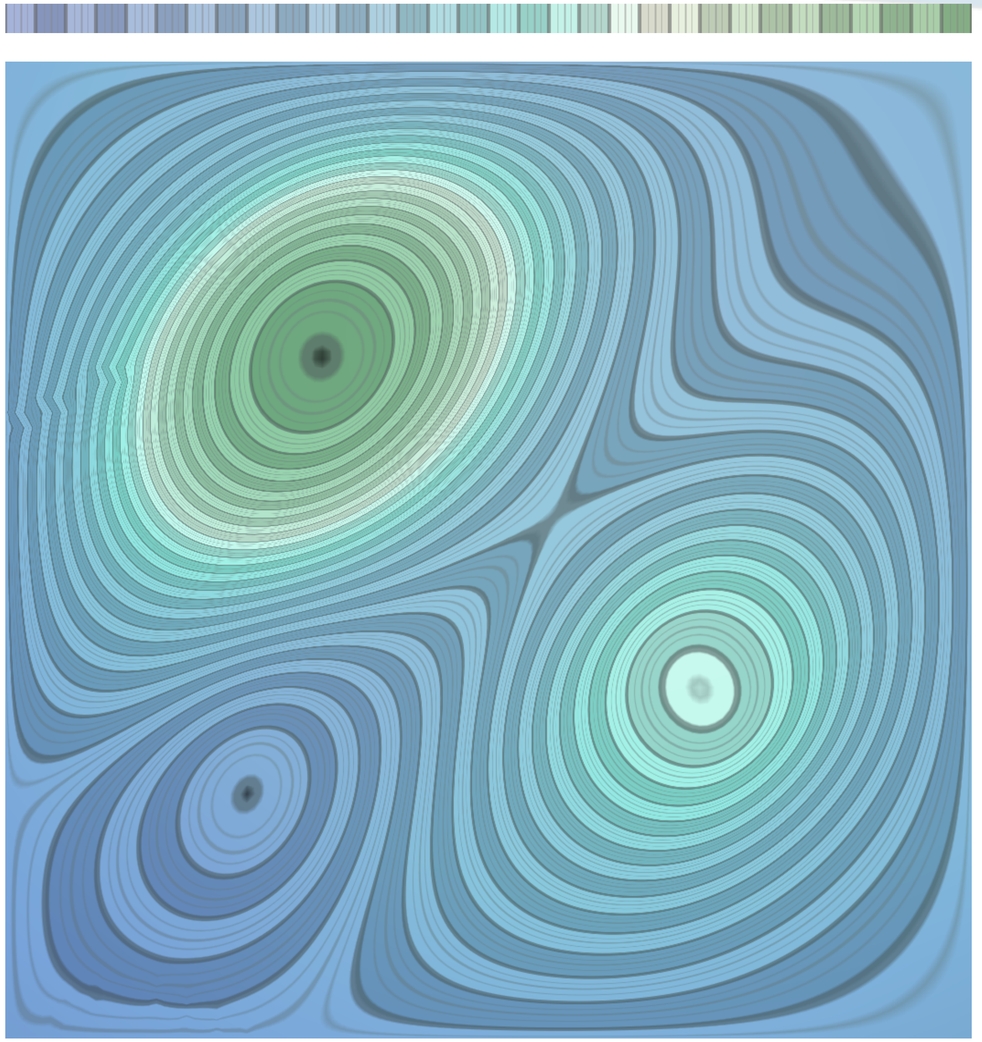




Color maps for scalar field visualization

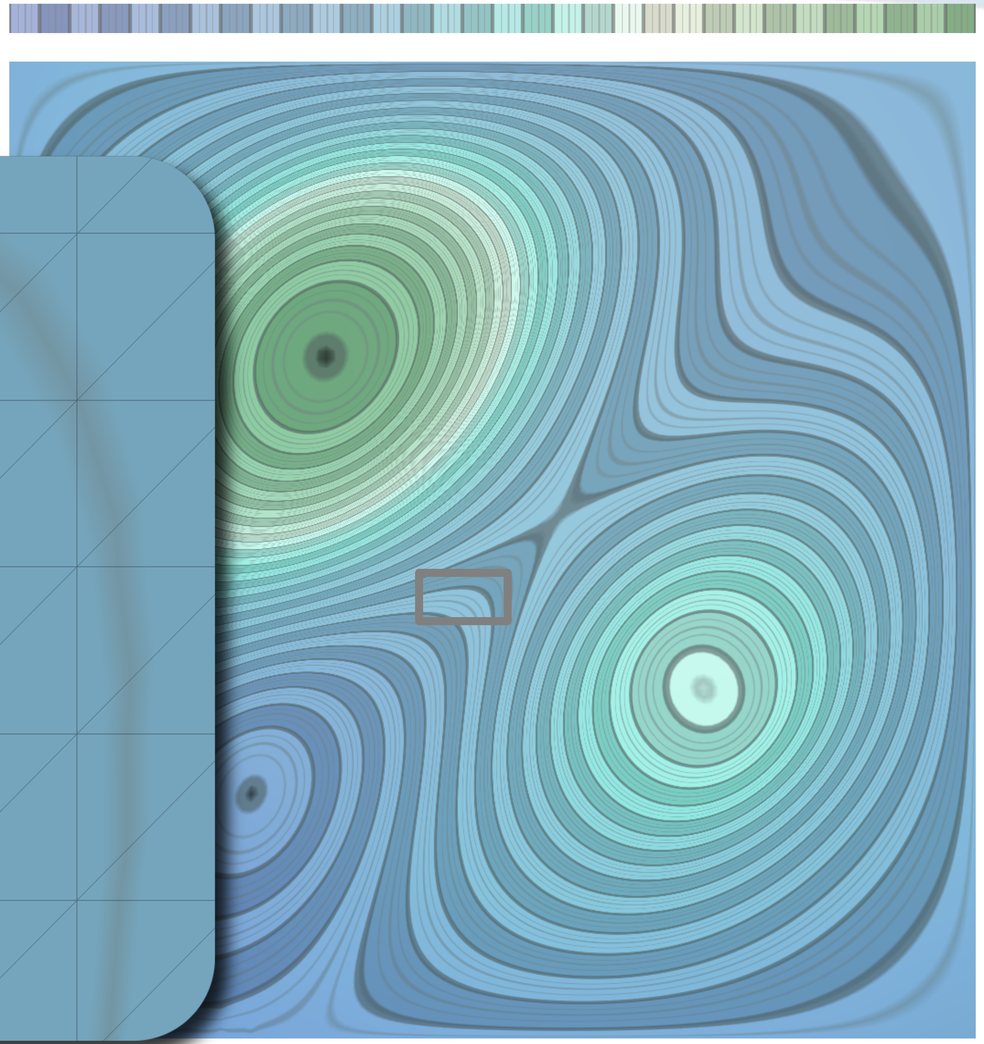
- Implicit level set visualization
  - Play with the texture?





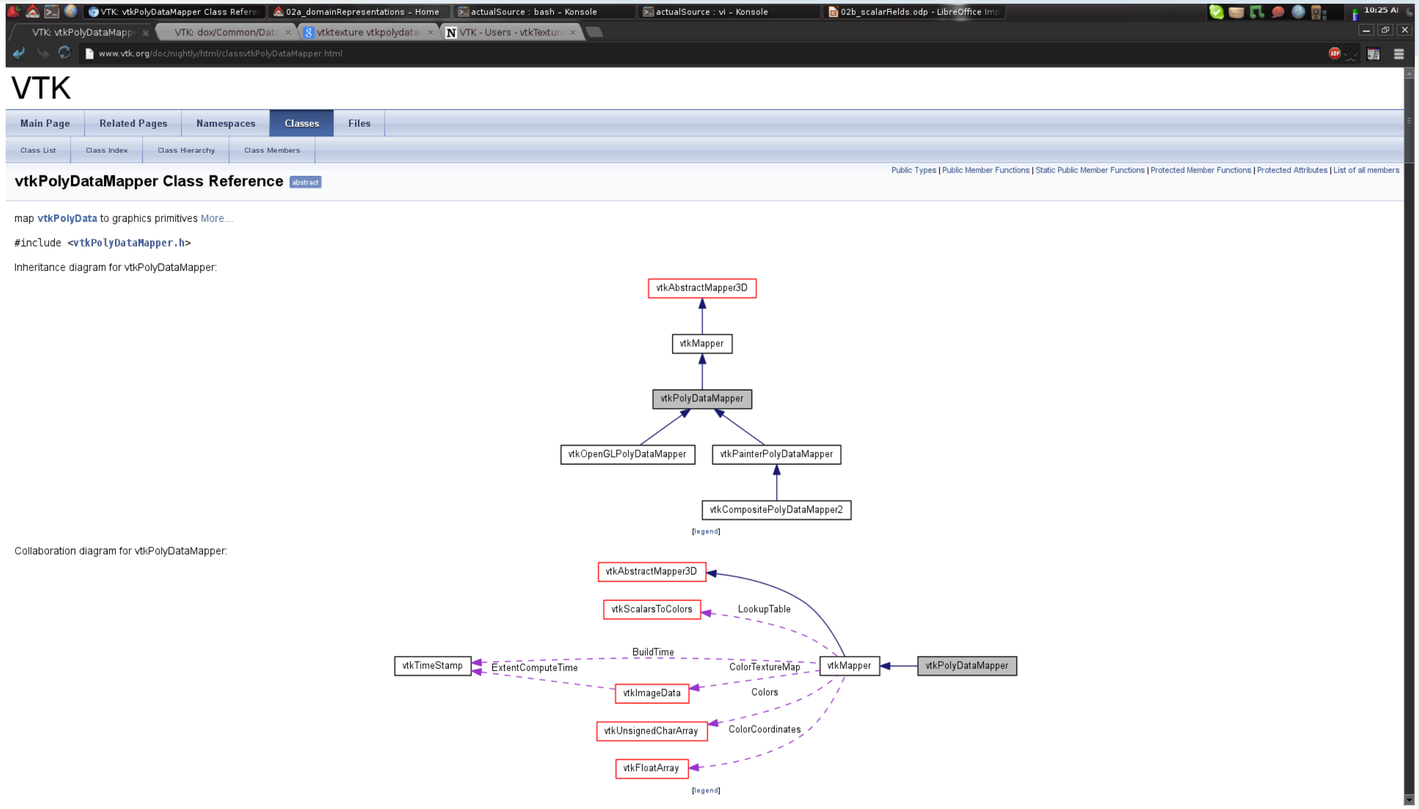
Color maps for scalar field visualization

- Implicit level set visualization



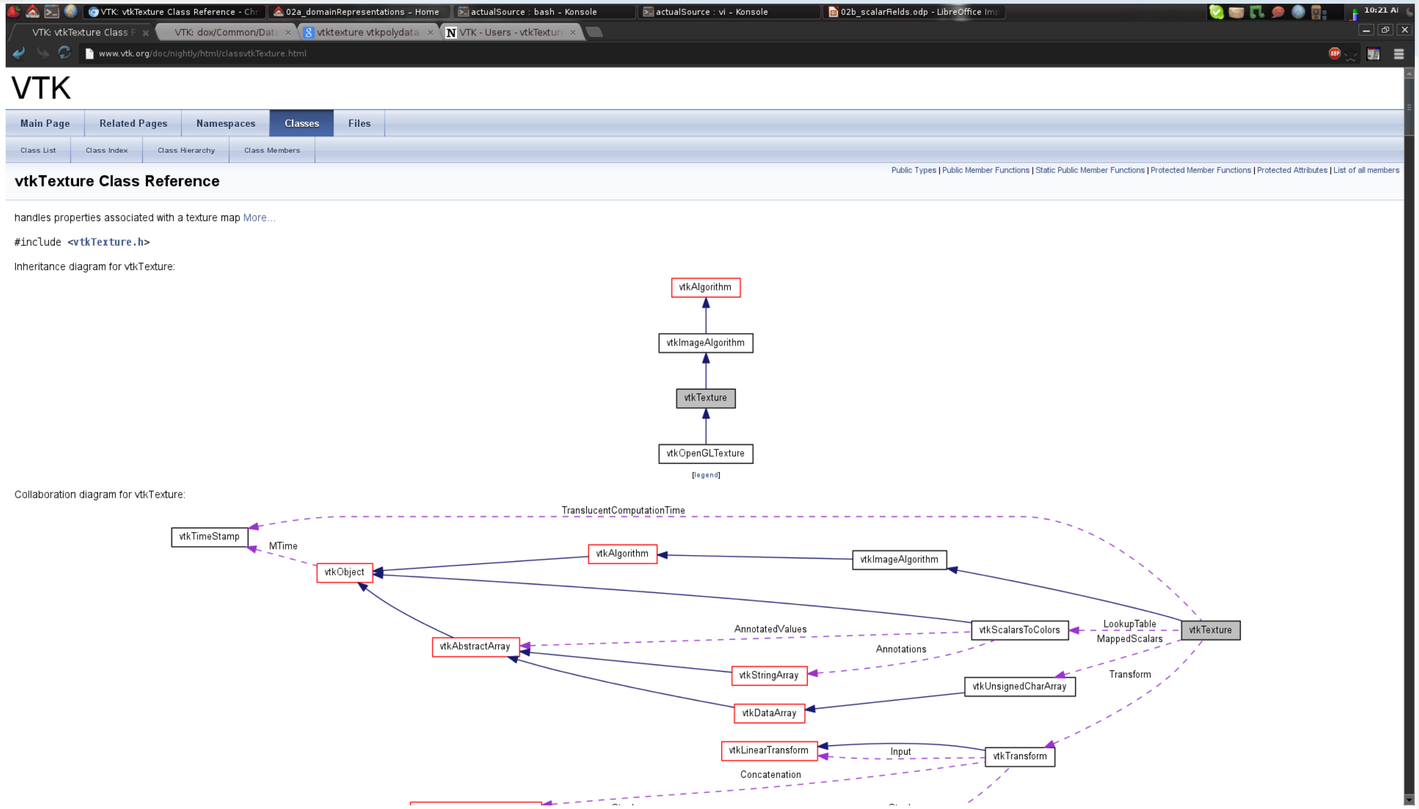
In practice

- VTK PolyDataMapper (2D)
  - Handles texture coordinates
  - Input: VTK PolyData



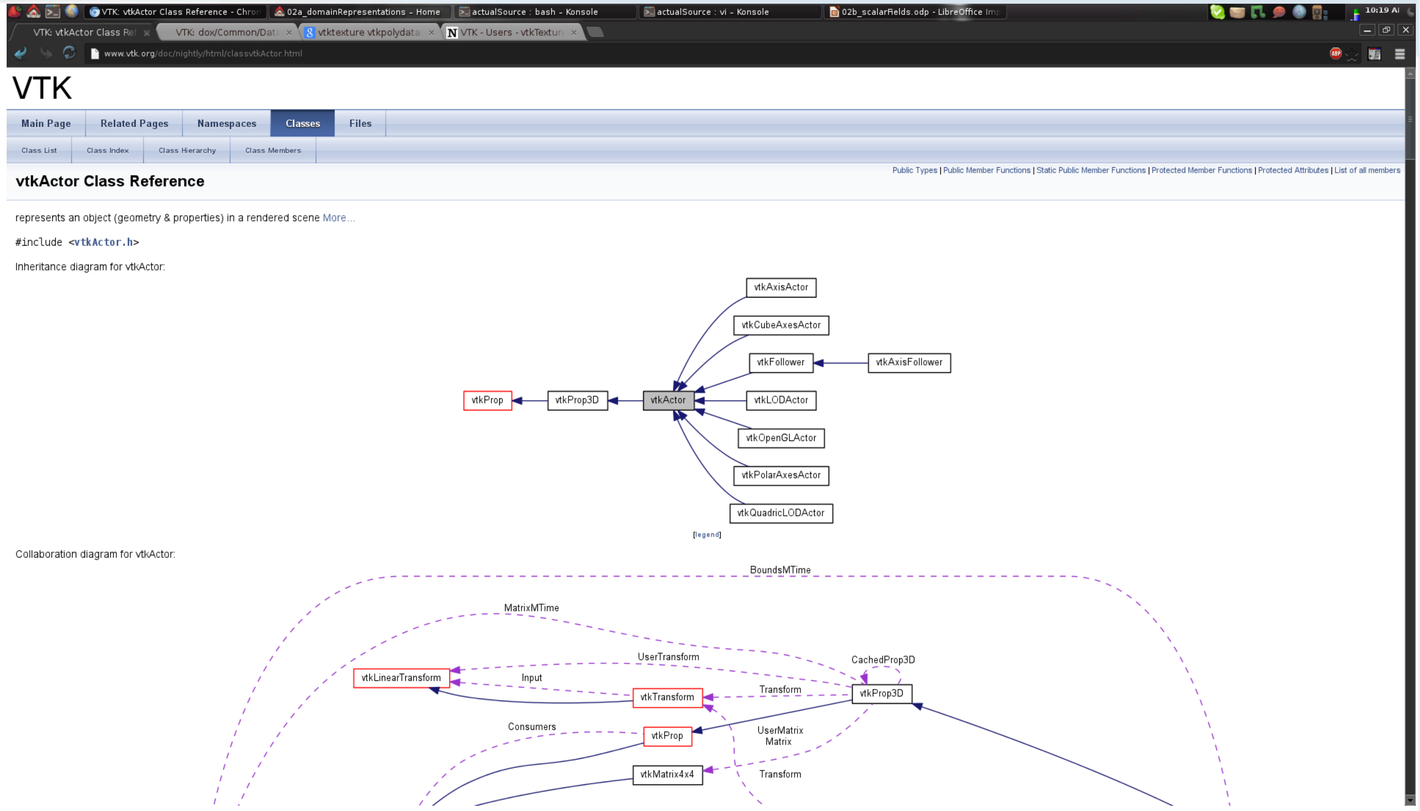
In practice

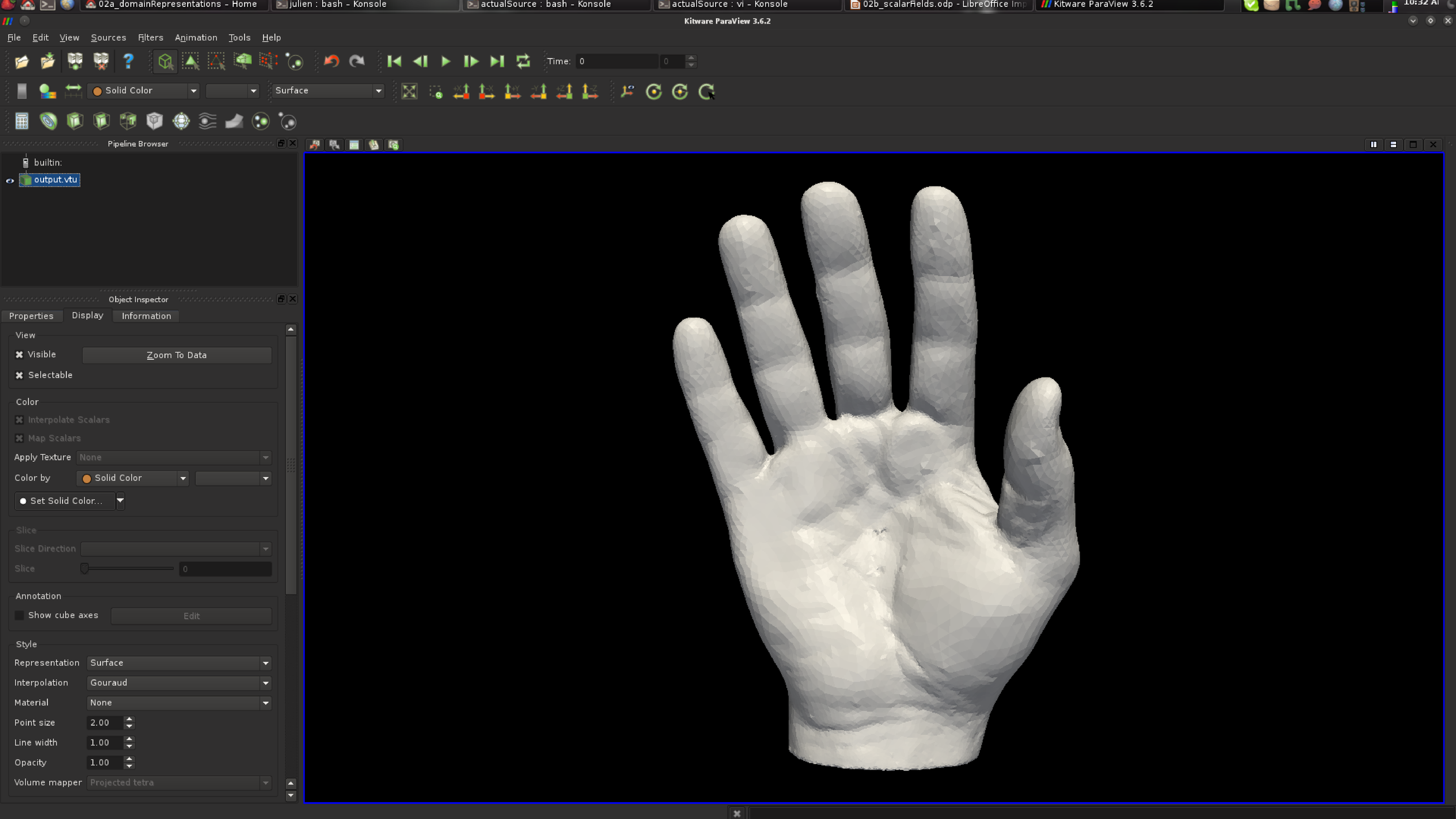
- VTK PolyDataMapper (2D)
  - Handles texture coordinates
  - Input: VTK PolyData
- VTK Texture
  - Represents a texture



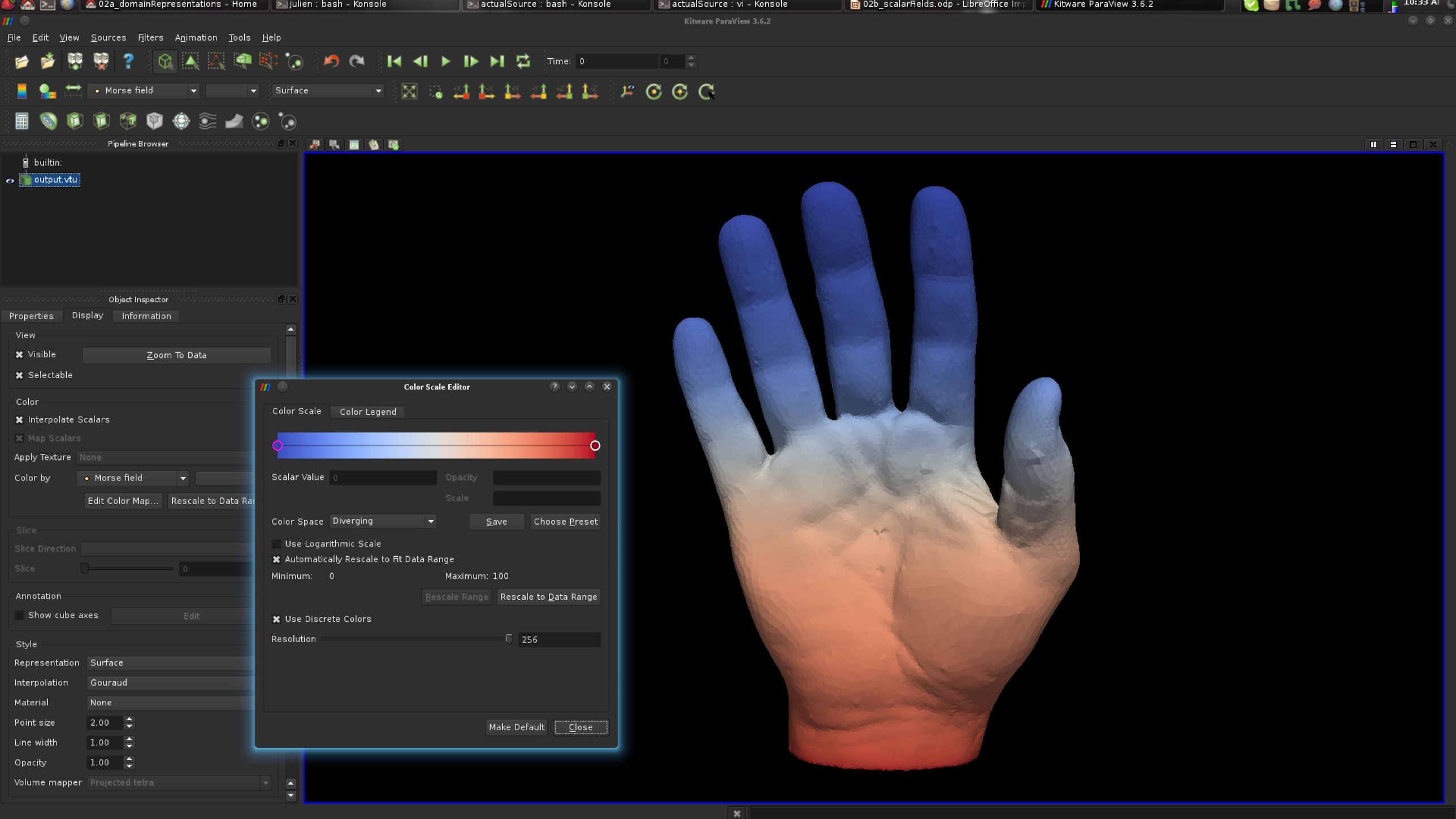
In practice

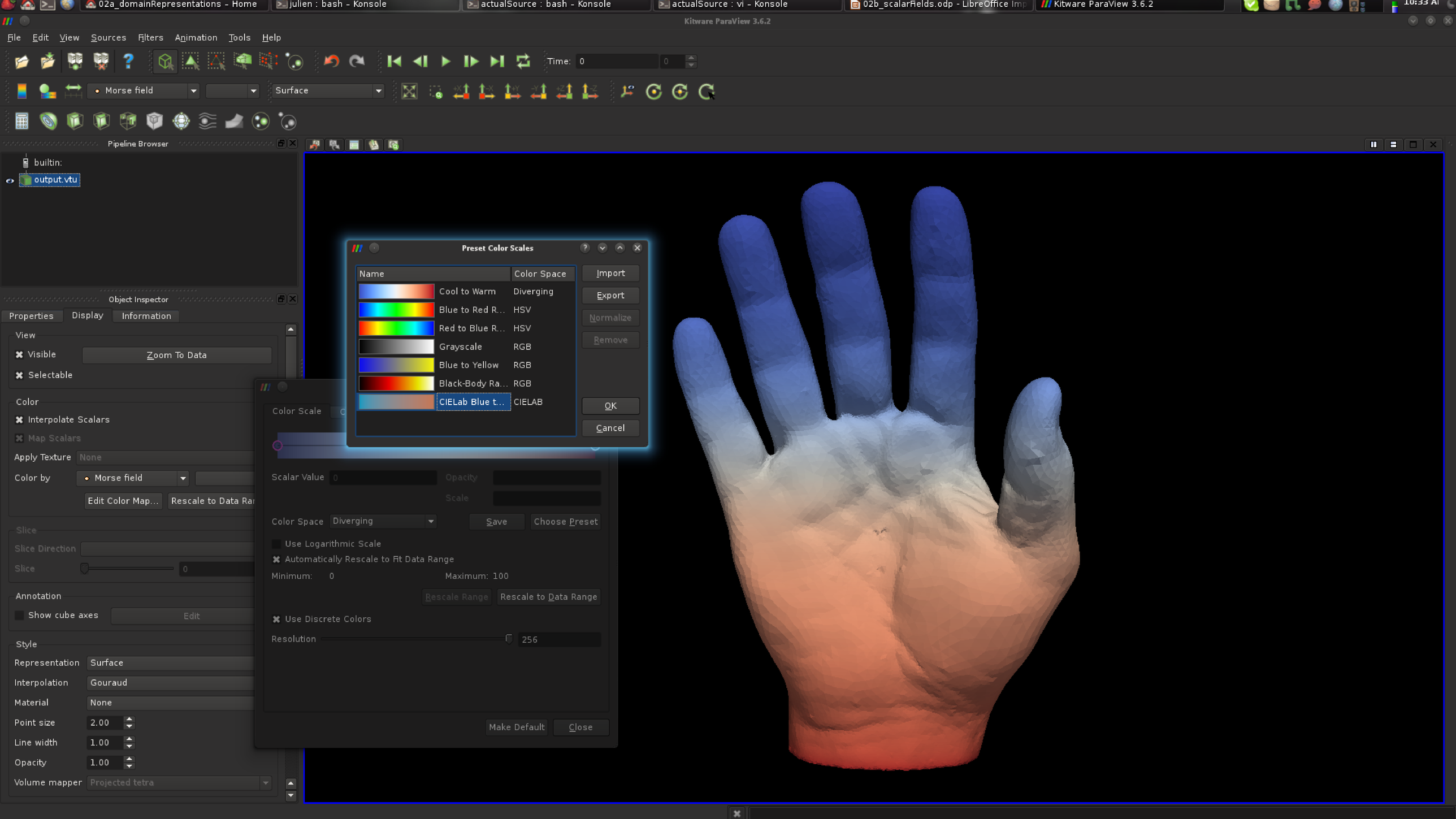
- VTK PolyDataMapper (2D)
  - Handles texture coordinates
  - Input: VTK PolyData
- VTK Texture
  - Represents a texture
- VTK Actor
  - Object representation for rendering
  - Inputs: texture and mapper

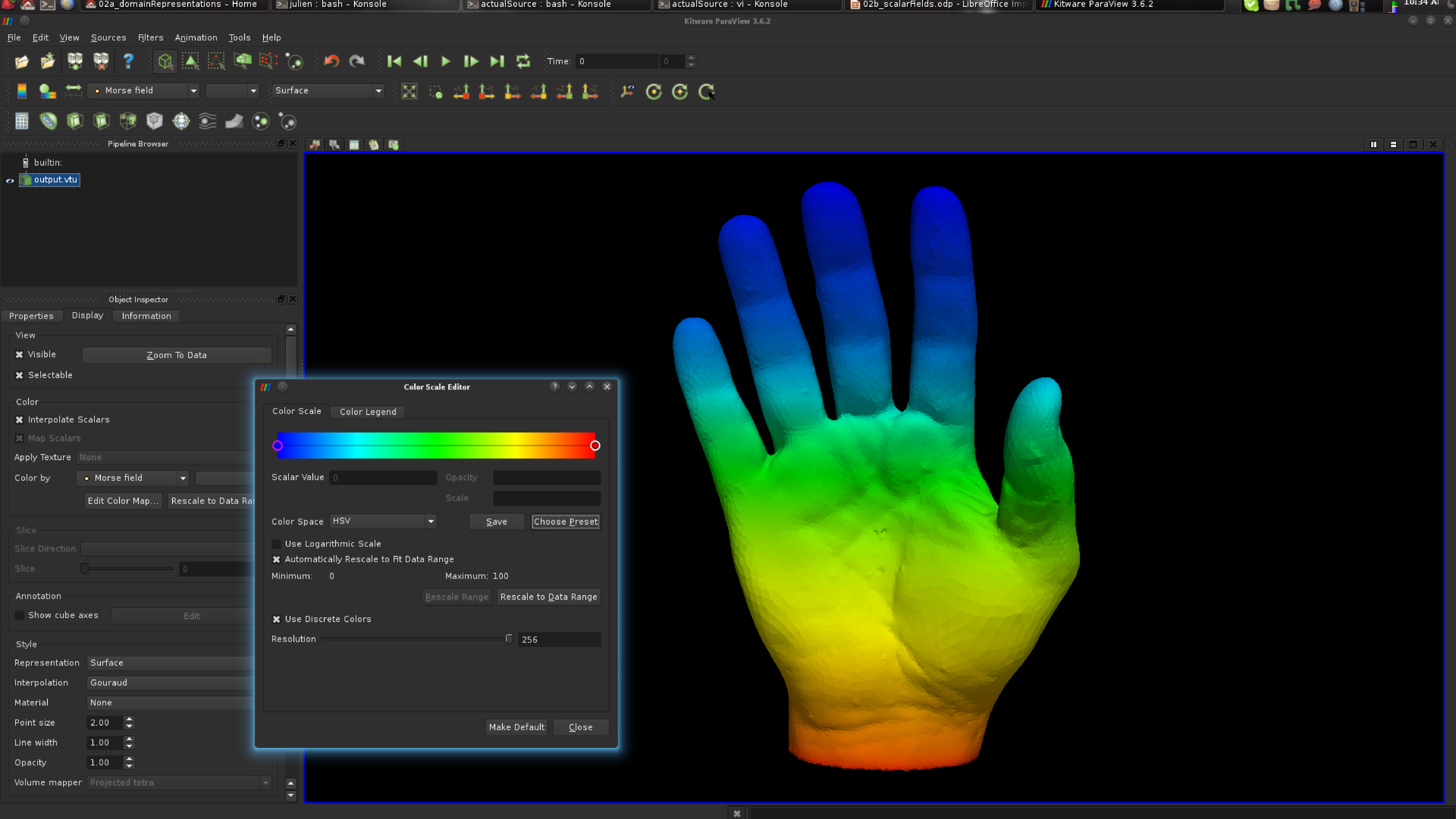



02a_domainRepresentations - Home
julien : bash - Konsole
actualSource : bash - Konsole
actualSource : vi - Konsole
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Solid Color
Surface
Pipeline Browser
builtin:
output.vtu
Object Inspector
Properties Display Information
View
Visible
Zoom To Data
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Solid Color
Set Solid Color...
Slice
Slice Direction
Slice 0
Annotation
Show cube axes
Edit
Style
Representation Surface
Interpolation Gouraud
Material None
Point size 2.00
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra

02a_domainRepresentations - Home
julien : bash - Konsole
actualSource : bash - Konsole
actualSource : vi - Konsole
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
Object Inspector
Properties Display Information
View
Visible
Zoom To Data
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Morse field
Edit Color Map...
Rescale to Data Range
Slice
Slice Direction
Slice 0
Annotation
Show cube axes
Edit
Style
Representation Surface
Interpolation Gouraud
Material None
Point size 2.00
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra

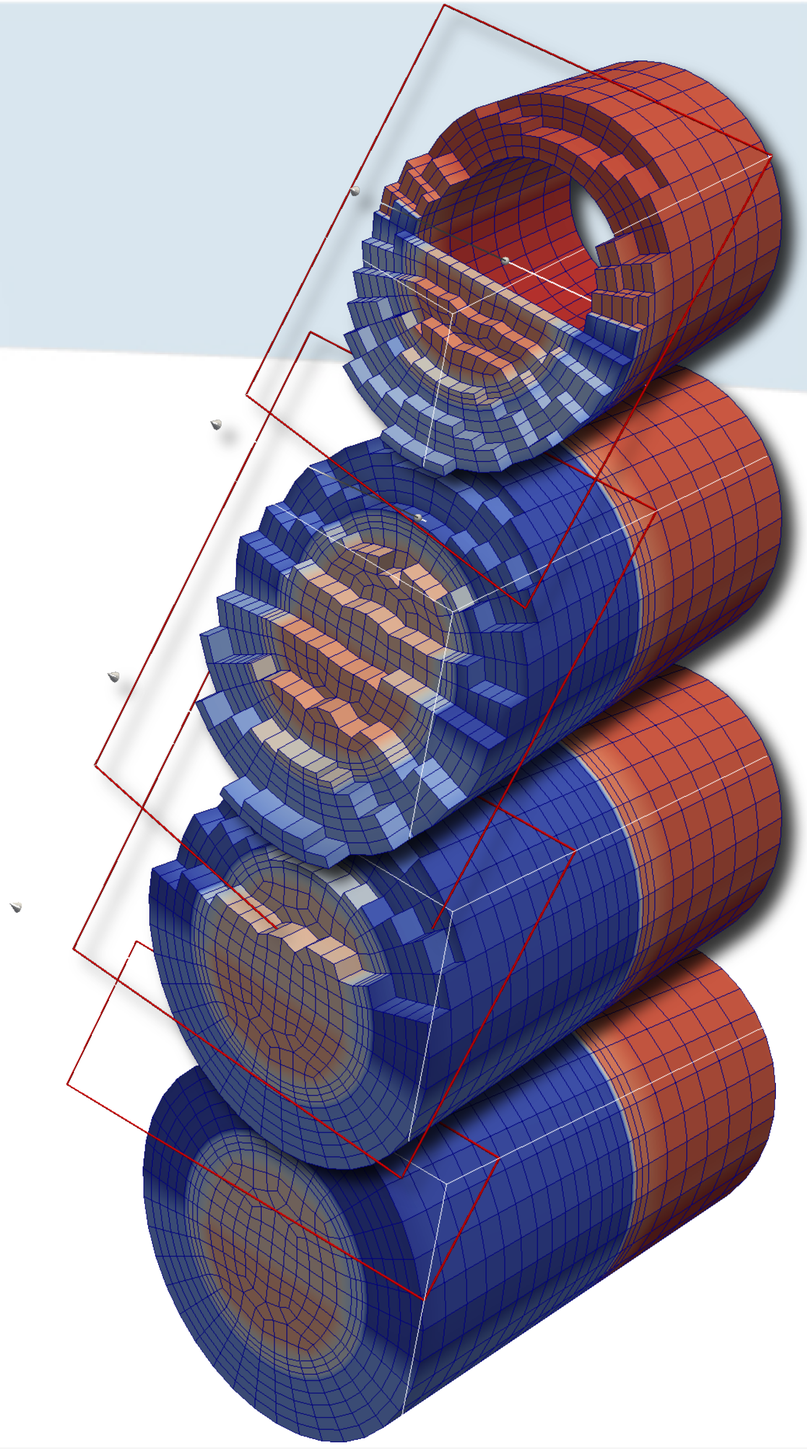

02a_domainRepresentations - Home
julien : bash - Konsole
actualSource : bash - Konsole
actualSource : vi - Konsole
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
Object Inspector
Properties Display Information
View
Visible
Zoom To Data
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Morse field
Edit Color Map...
Rescale to Data Ra
Slice
Slice Direction
Slice 0
Annotation
Show cube axes
Edit
Style
Representation Surface
Interpolation Gouraud
Material None
Point size 2.00
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Color Scale Editor
Color Scale
Color Legend
Scalar Value 0
Opacity
Scale
Color Space Diverging
Save
Choose Preset
Use Logarithmic Scale
Automatically Rescale to Fit Data Range
Minimum: 0
Maximum: 100
Rescale Range
Rescale to Data Range
Use Discrete Colors
Resolution 256
Make Default
Close


02a_domainRepresentations - Home
julien : bash - Konsole
actualSource : bash - Konsole
actualSource : vi - Konsole
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
Object Inspector
Properties Display Information
View
Visible
Zoom To Data
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Morse field
Edit Color Map...
Rescale to Data Ra
Slice
Slice Direction
Slice 0
Annotation
Show cube axes
Edit
Style
Representation Surface
Interpolation Gouraud
Material None
Point size 2.00
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Preset Color Scales
Name
Color Space
Cool to Warm Diverging
Blue to Red R... HSV
Red to Blue R... HSV
Grayscale RGB
Blue to Yellow RGB
Black-Body Ra... RGB
CIELab Blue t... CIELAB
Import
Export
Normalize
Remove
OK
Cancel
Color Scale
Scalar Value 0
Opacity
Scale
Color Space Diverging
Save
Choose Preset
Use Logarithmic Scale
Automatically Rescale to Fit Data Range
Minimum: 0
Maximum: 100
Rescale Range
Rescale to Data Range
Use Discrete Colors
Resolution 256
Make Default
Close


02a_domainRepresentations - Home
julien : bash - Konsole
actualSource : bash - Konsole
actualSource : vi - Konsole
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
Object Inspector
Properties Display Information
View
Visible
Zoom To Data
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Morse field
Edit Color Map...
Rescale to Data Ra
Slice
Slice Direction
Slice 0
Annotation
Show cube axes
Edit
Style
Representation Surface
Interpolation Gouraud
Material None
Point size 2.00
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Color Scale Editor
Color Scale Color Legend
Scalar Value 0
Opacity
Scale
Color Space HSV
Save
Choose Preset
Use Logarithmic Scale
Automatically Rescale to Fit Data Range
Minimum: 0
Maximum: 100
Rescale Range
Rescale to Data Range
Use Discrete Colors
Resolution 256
Make Default
Close

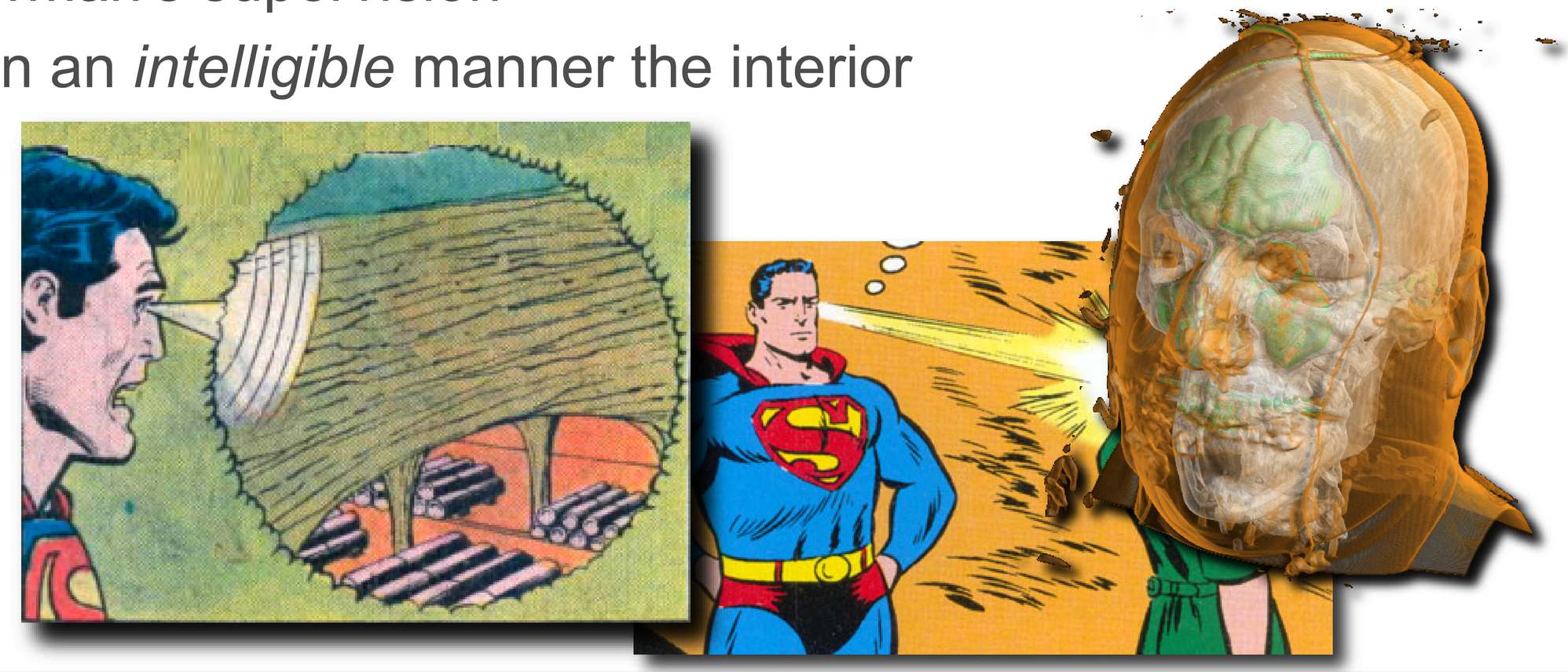
Color maps

- Not really useful for volumetric domains



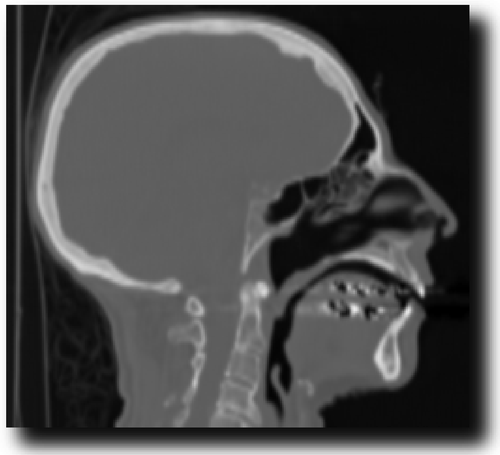
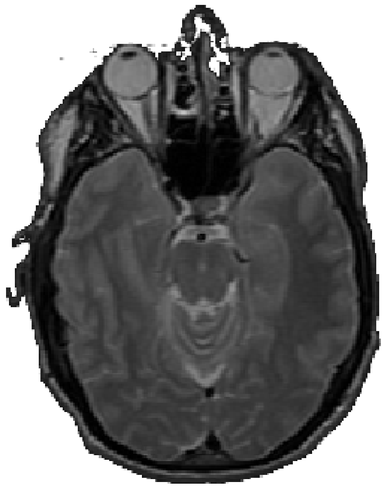
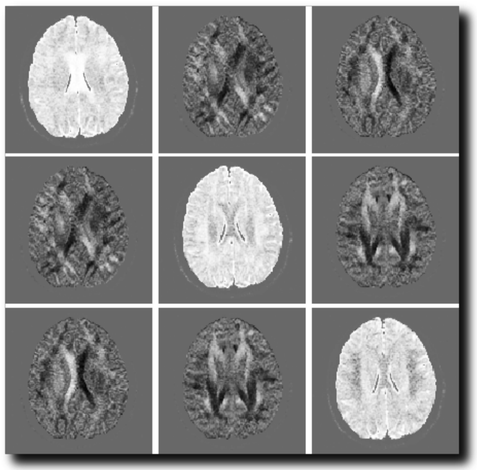
Color maps

- Not really useful for volumetric domains
  - Mimic Superman's supervision
  - Represent in an *intelligible* manner the interior



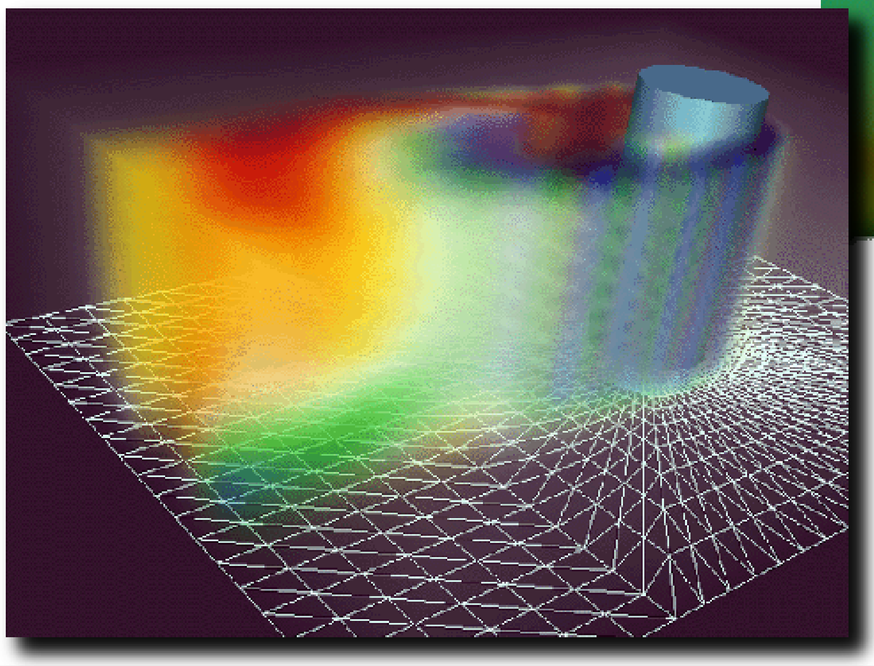
Volume rendering motivations

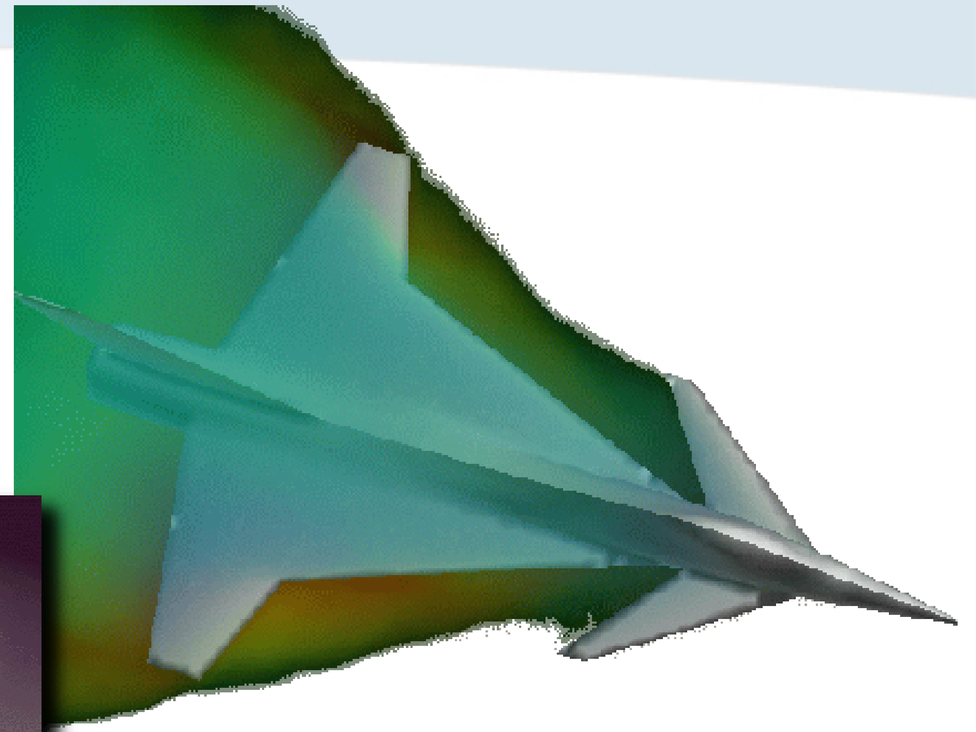
- Volumetric scalar fields
  - Acquired data-sets
  - Computed Tomography, Magnetic Reasonance, Ultrasound


Volume rendering motivations

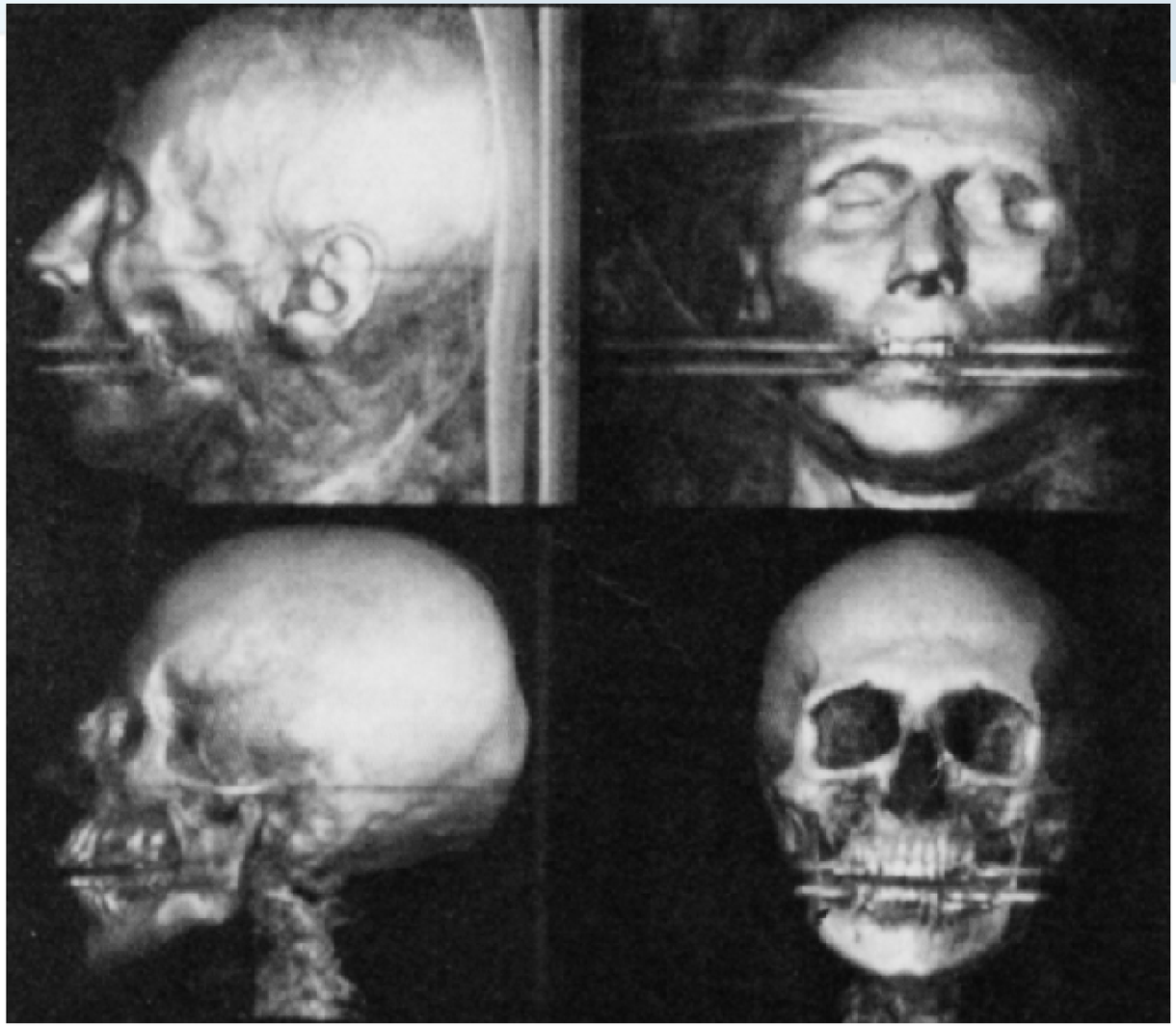
- Volumetric scalar fields
  - Simulated data-sets
    - Fluid dynamics
    - Pressure
    - Porosity
    - Temperature
    - Etc.





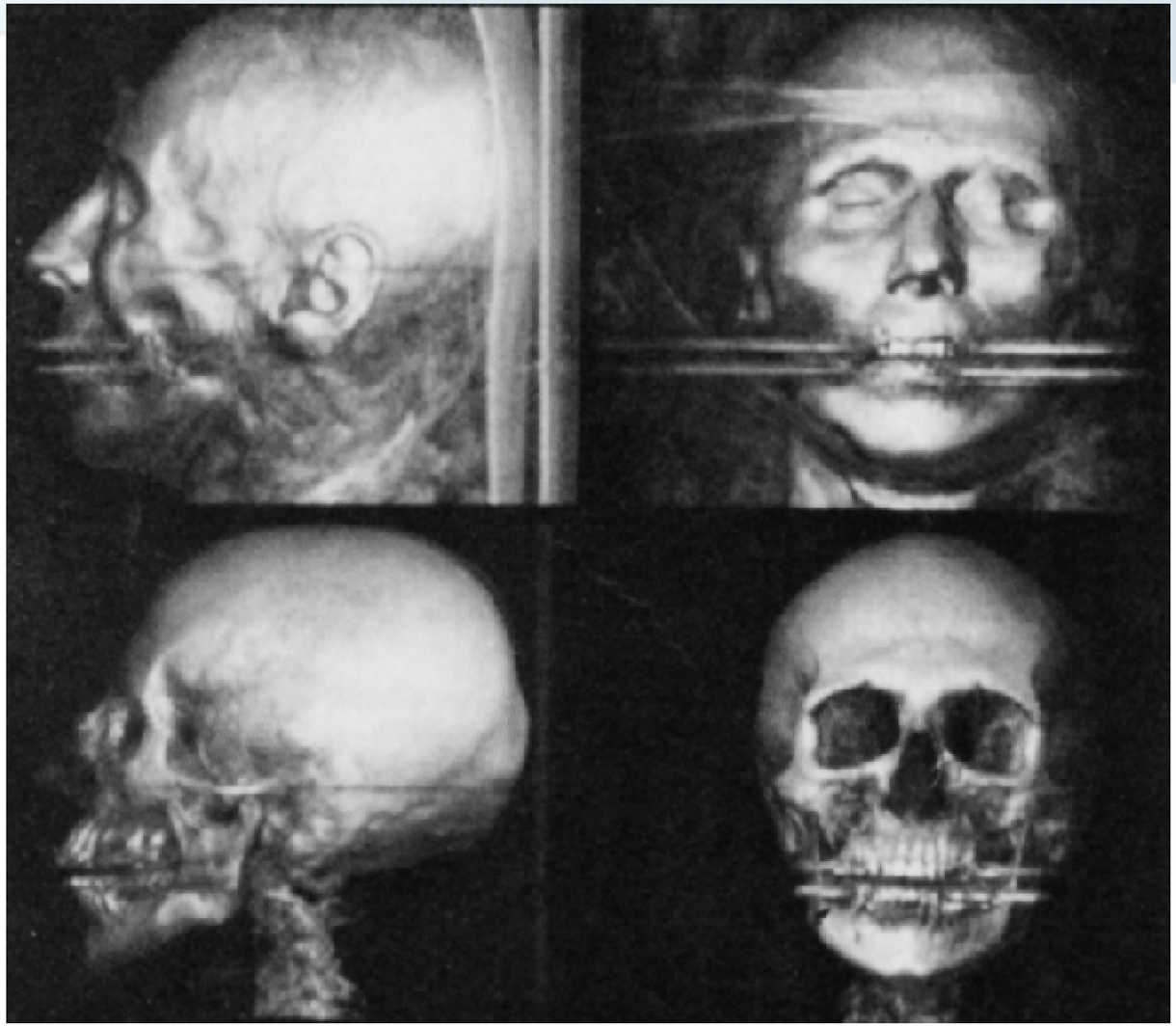
Volume rendering motivations

- Specific features
  - Represents all the field
    - Not only a few specific values
    - Many “layers”



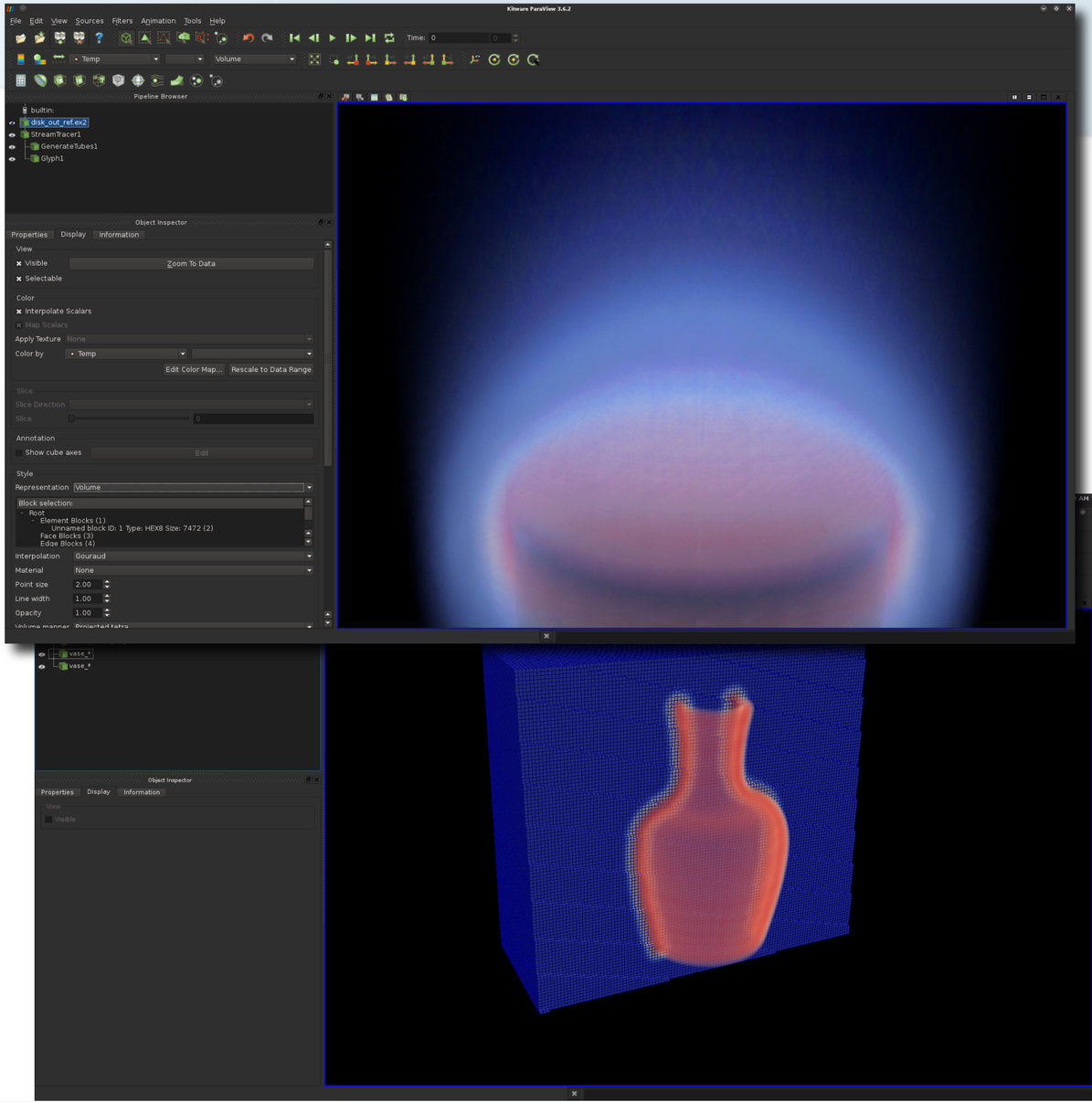
Volume rendering motivations

- Specific features
  - Represents all the field
    - Not only a few specific values
    - Many “layers”
- Key idea
    - Every cell should contribute
    - Blending by integration
    - Greater flexibility



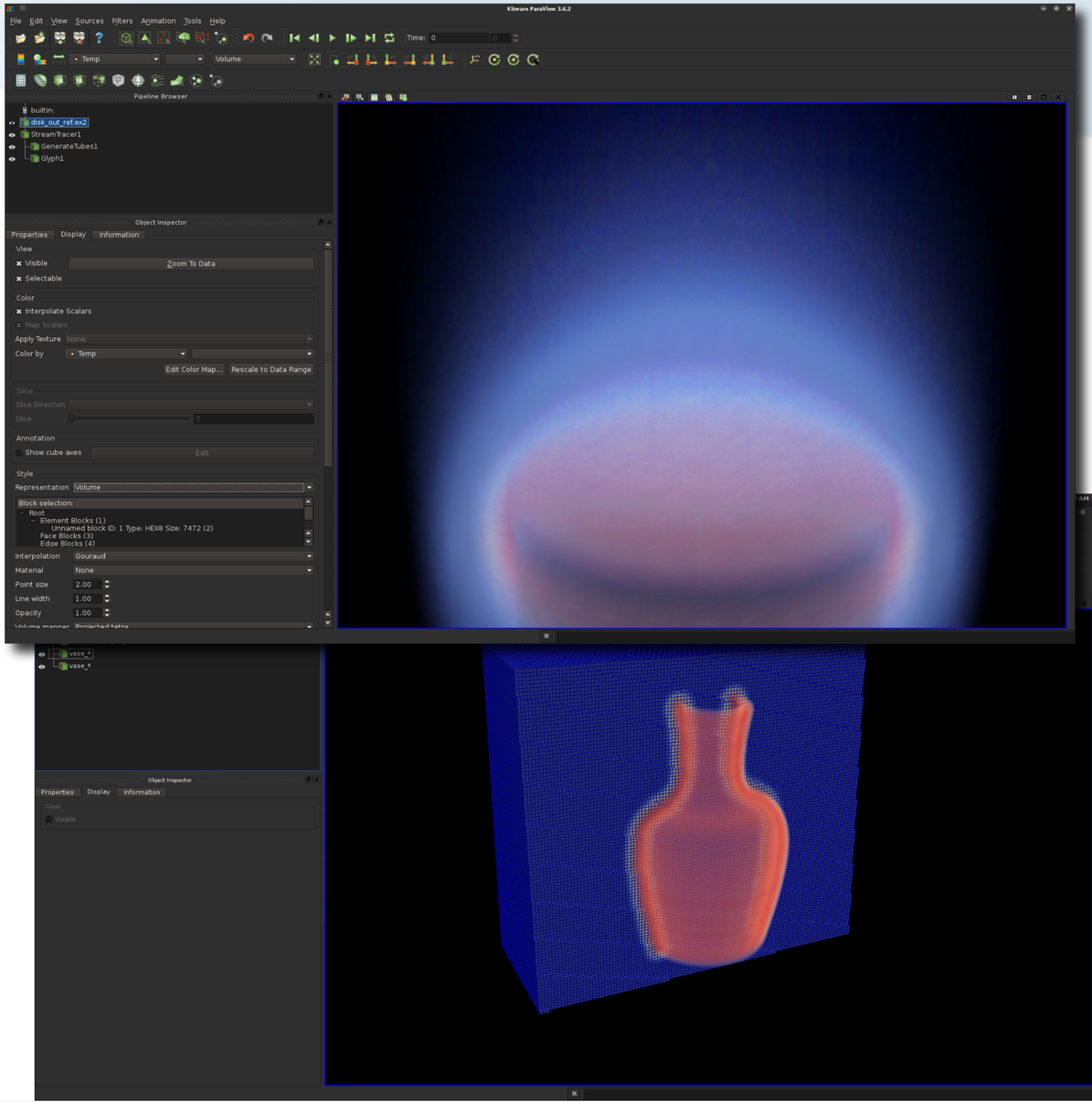
Volume rendering overview

- Rules to define



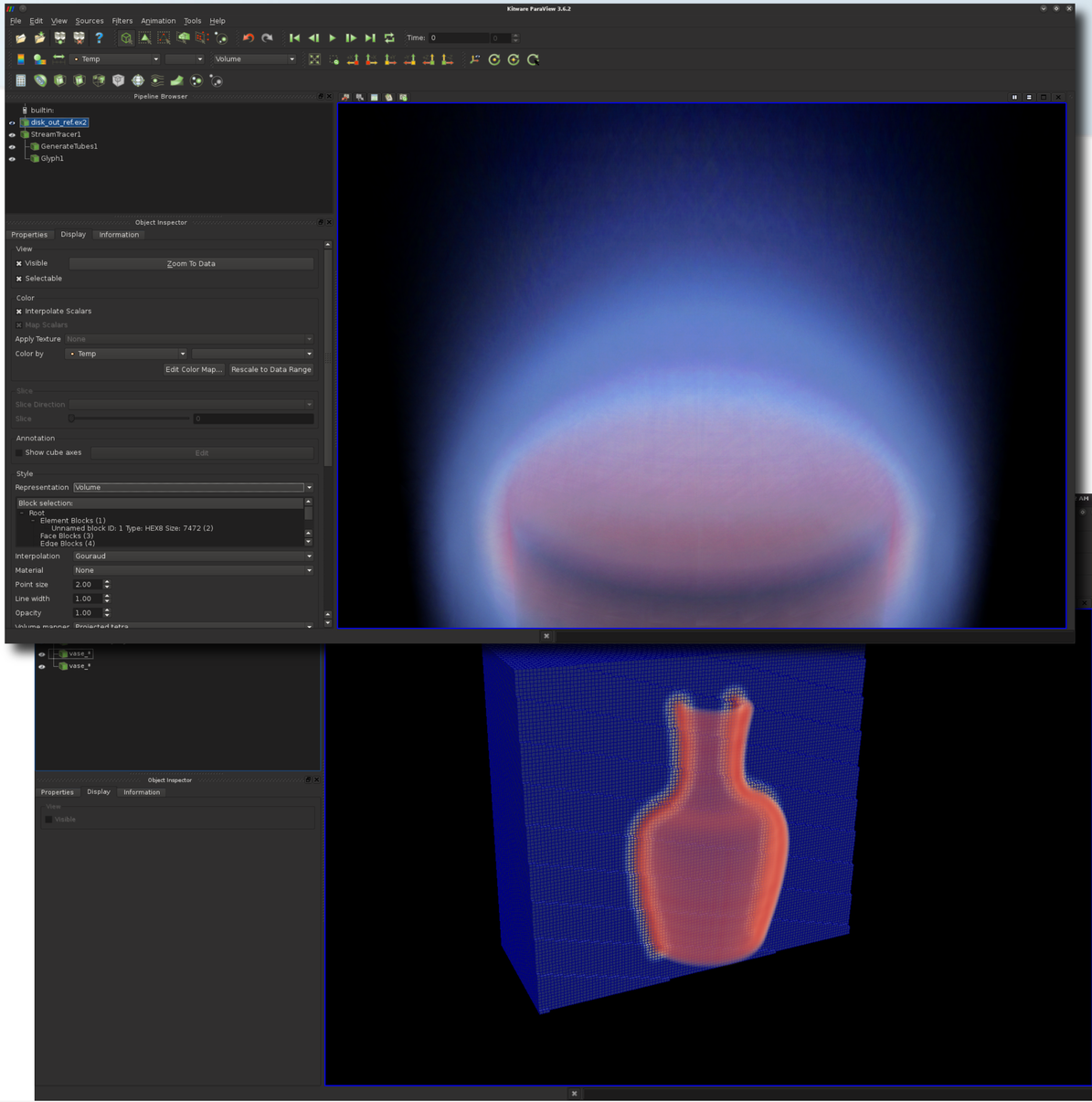
Volume rendering overview

- Rules to define
  - The properties of the “layers”



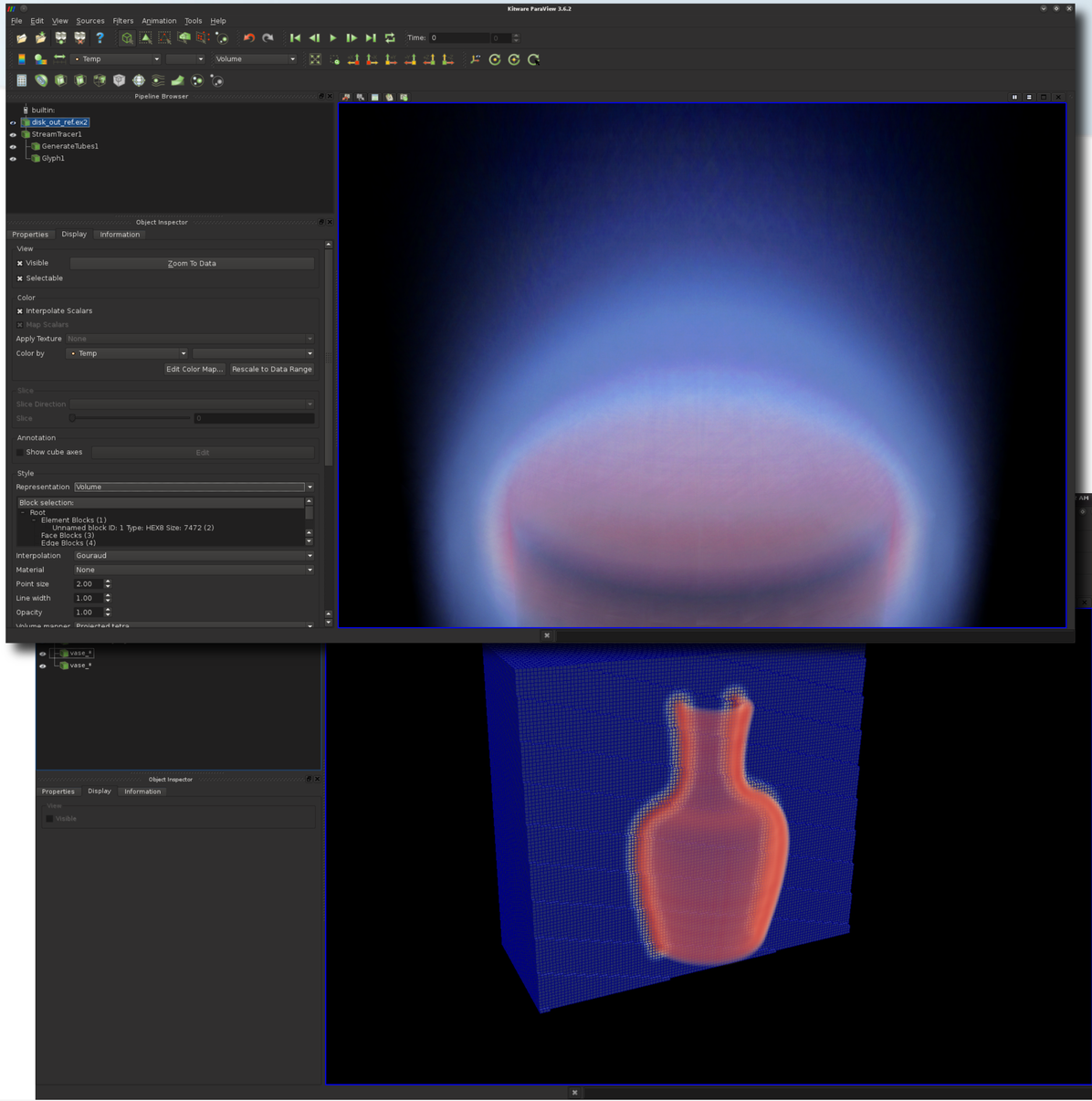
Volume rendering overview

- Rules to define
  - ~~The properties of the “layers”~~
  - The properties of the level sets



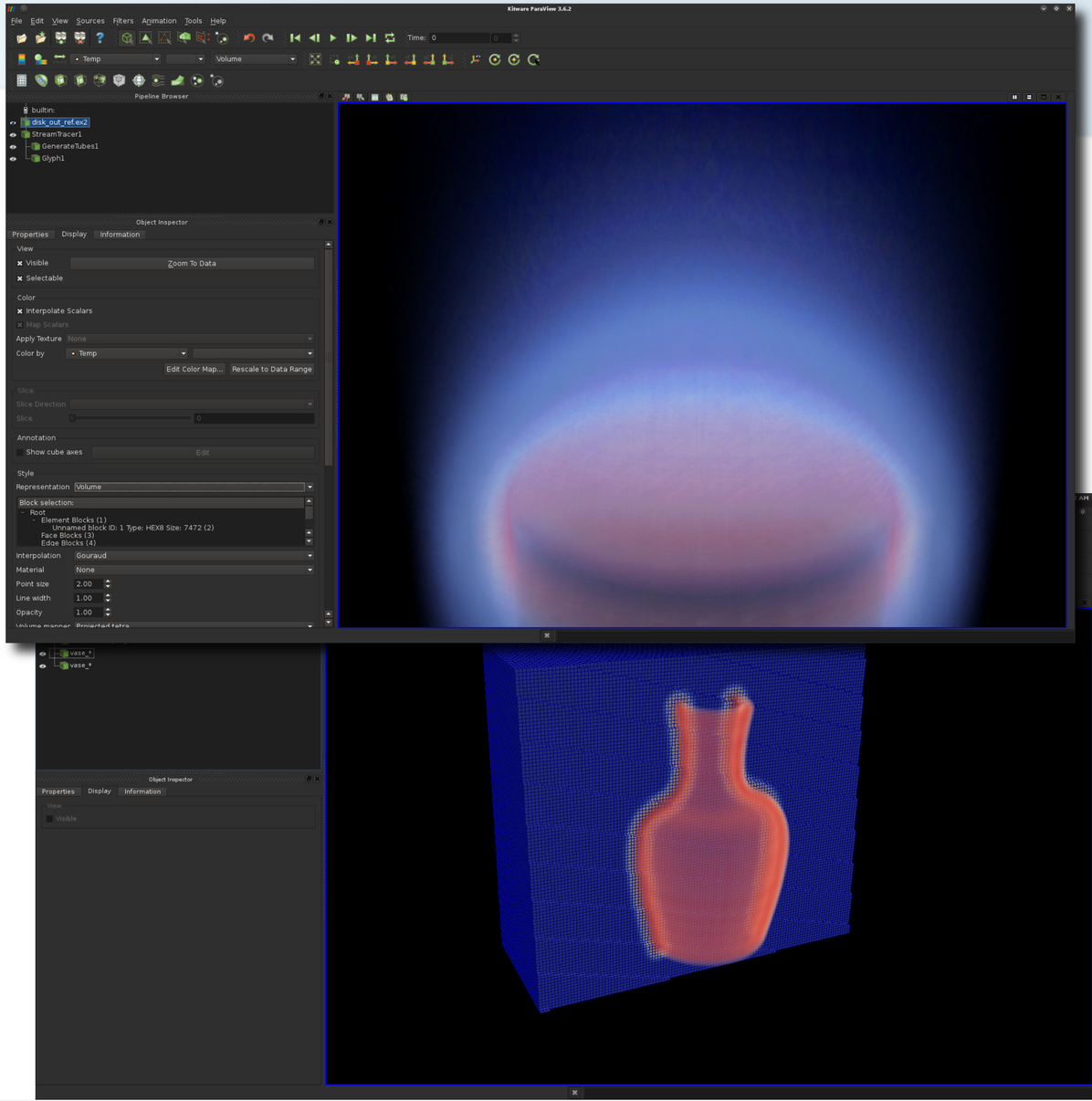
Volume rendering overview

- Rules to define
  - ~~The properties of the “layers”~~
  - The properties of the level sets
    - Color
    - Opacity



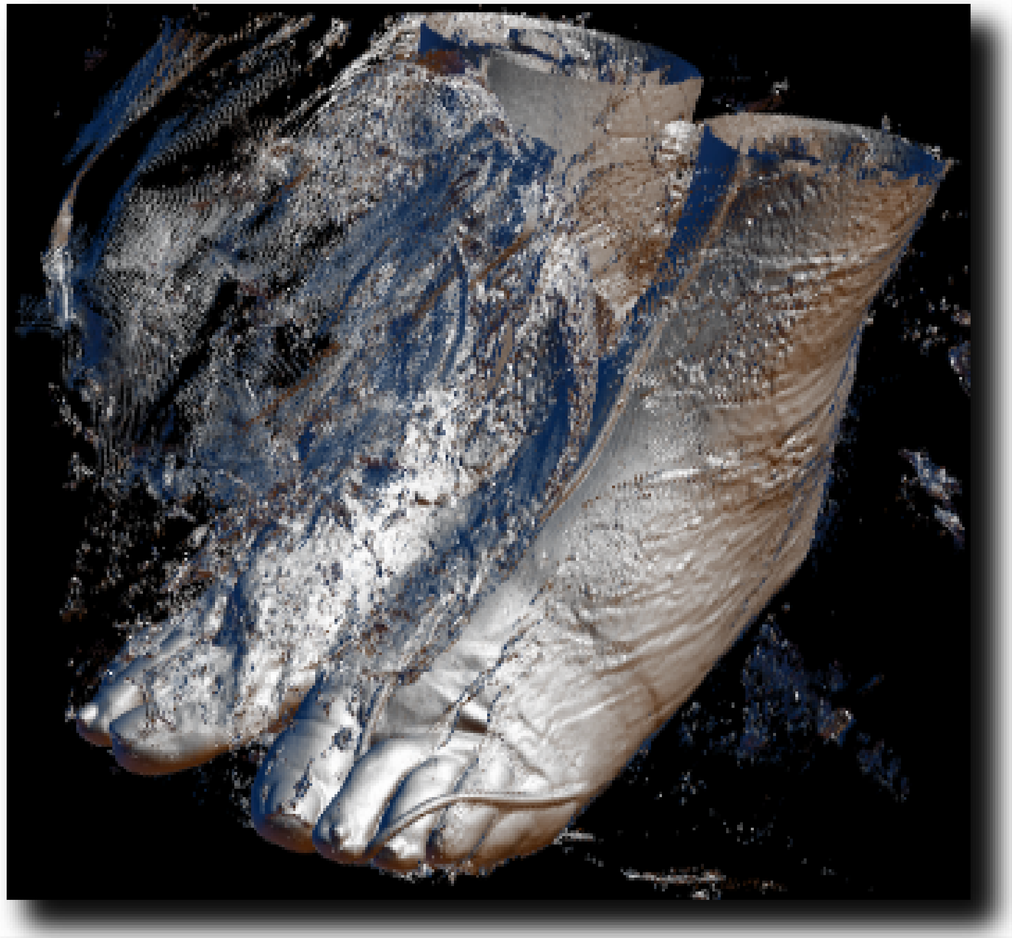
Volume rendering overview

- Rules to define
  - ~~The properties of the “layers”~~
  - The properties of the level sets
    - Color
    - Opacity
- Rendering
  - Accumulation process
  - View-dependent



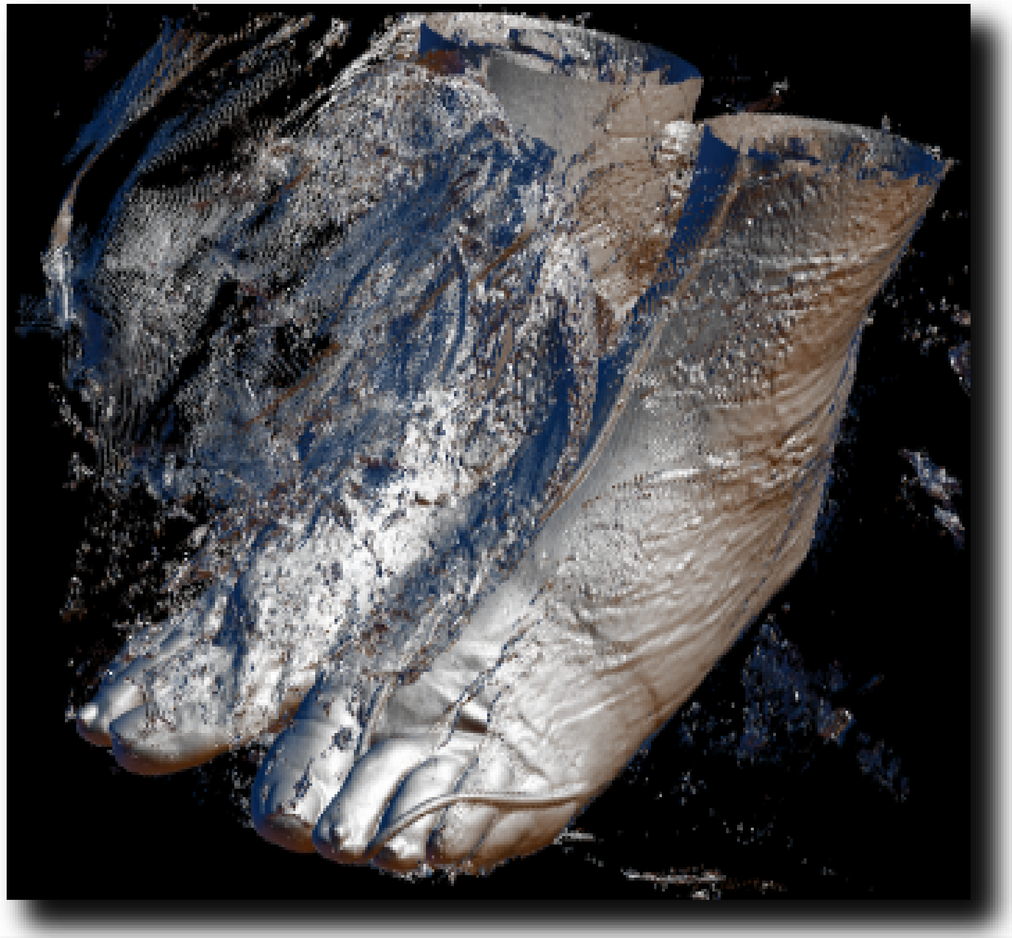
Transfer functions

- Assign distinct *materials* (function intervals)
  - To different properties
    - Color, Opacity



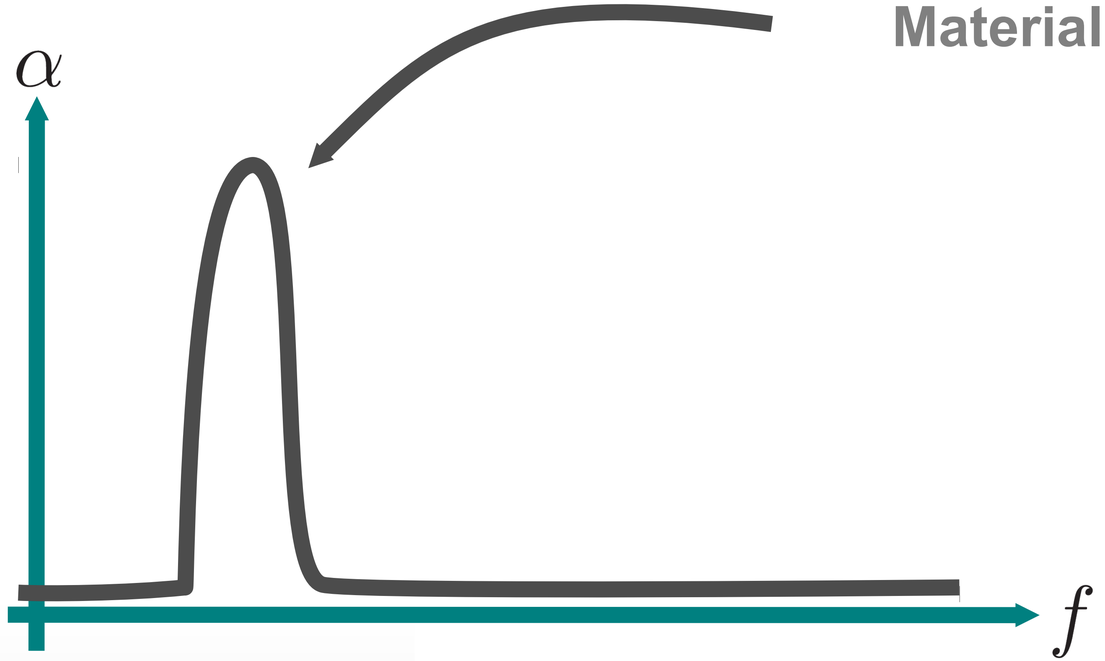
Transfer functions

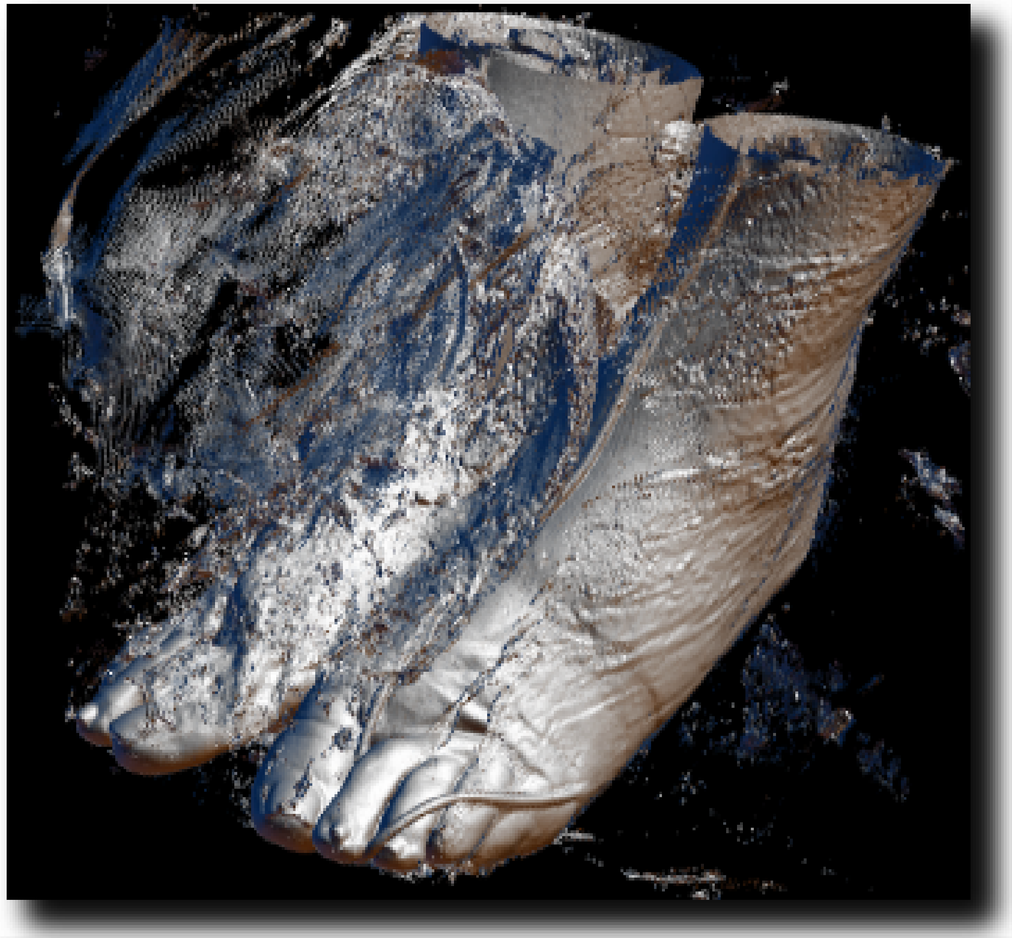
- Assign distinct *materials* (function intervals)
  - To different properties
    - Color, Opacity



Transfer functions

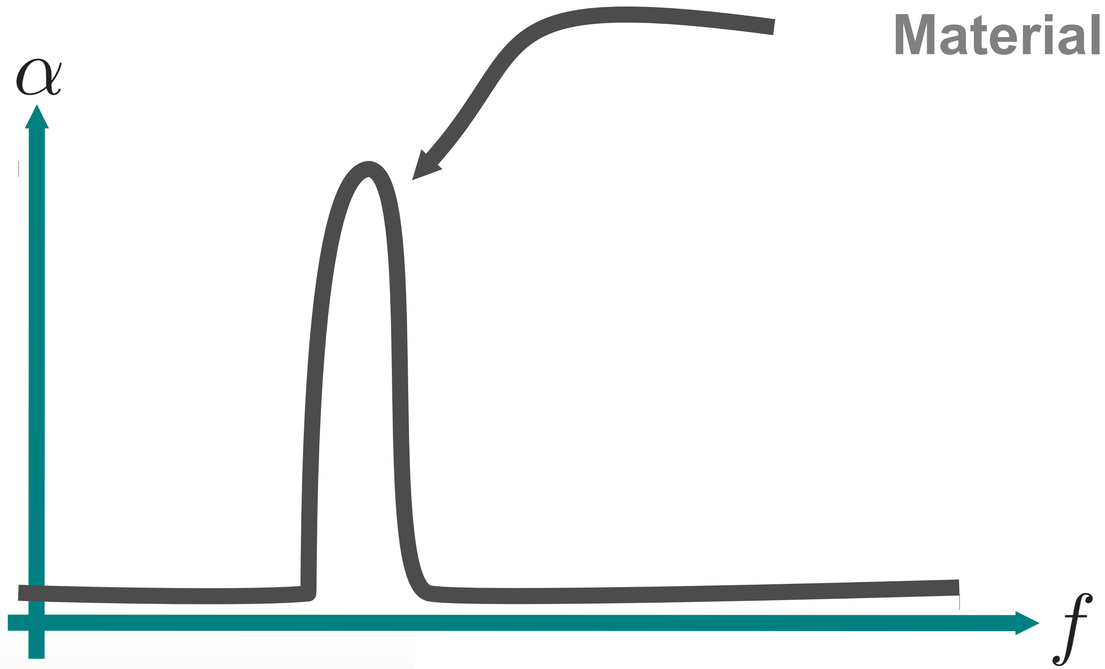
- Assign distinct *materials* (function intervals)
  - To different properties
    - Color, Opacity





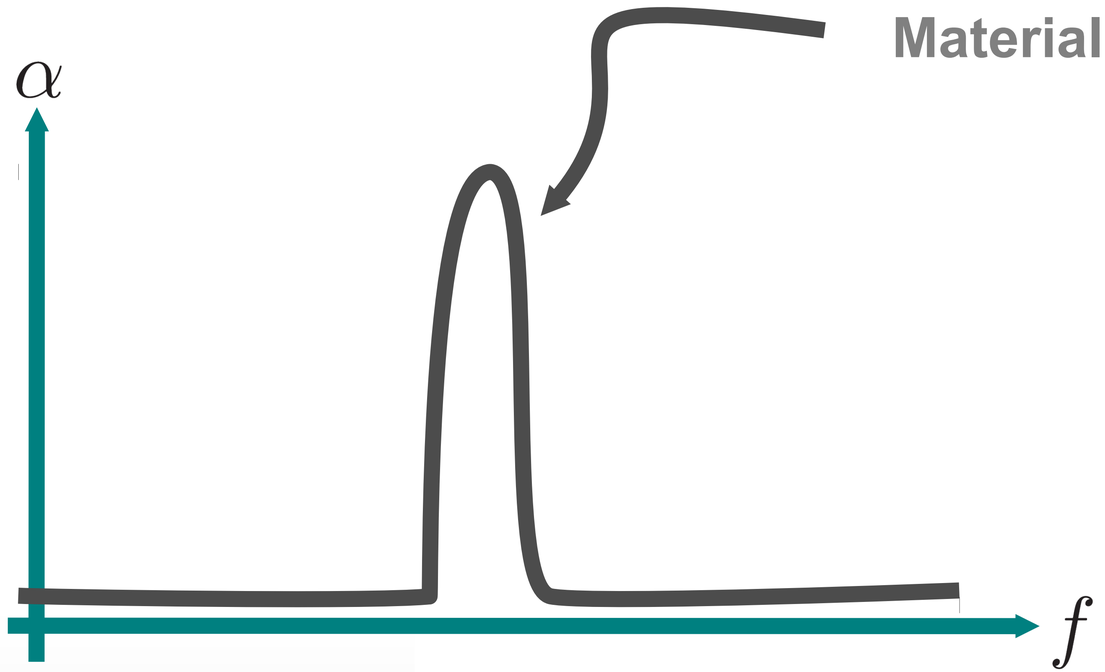
Transfer functions

- Assign distinct *materials* (function intervals)
  - To different properties
    - Color, Opacity



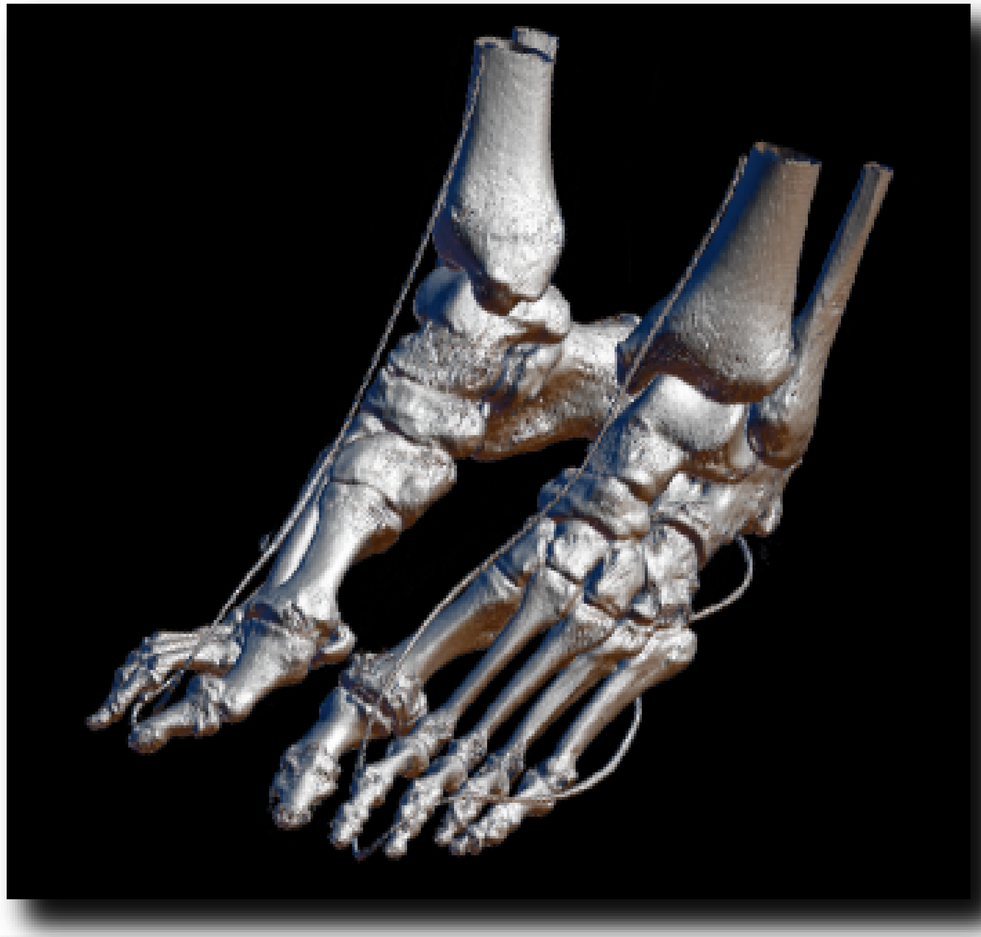
Transfer functions

- Assign distinct *materials* (function intervals)
  - To different properties
    - Color, Opacity

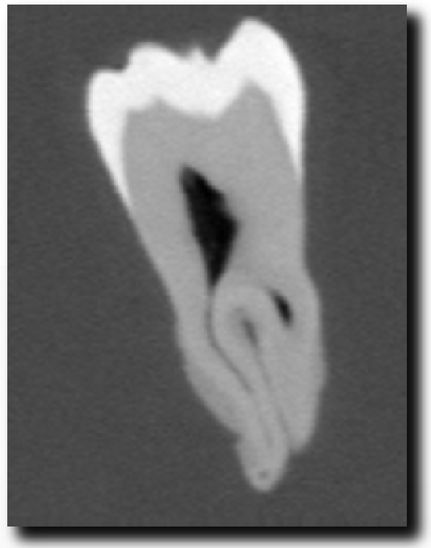


Transfer functions

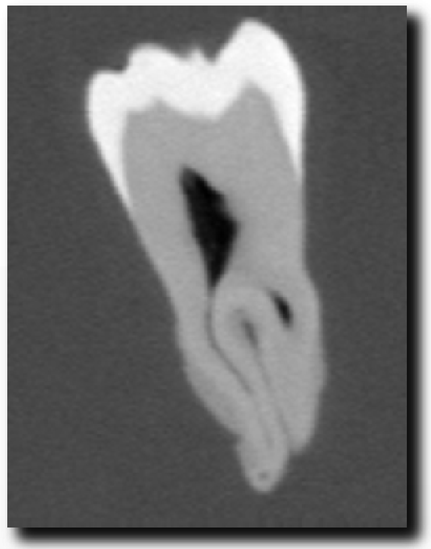
- Assign distinct *materials* (function intervals)
  - To different properties
    - Color, Opacity



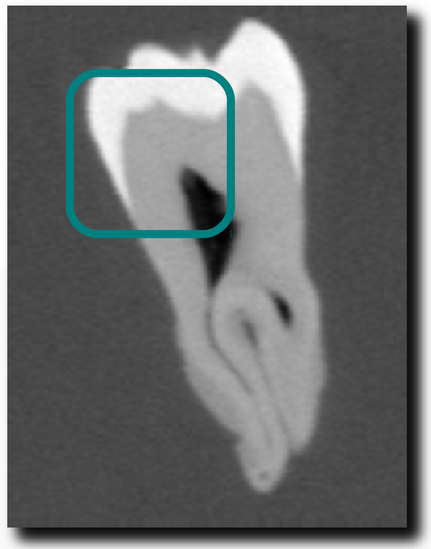
Transfer function examples



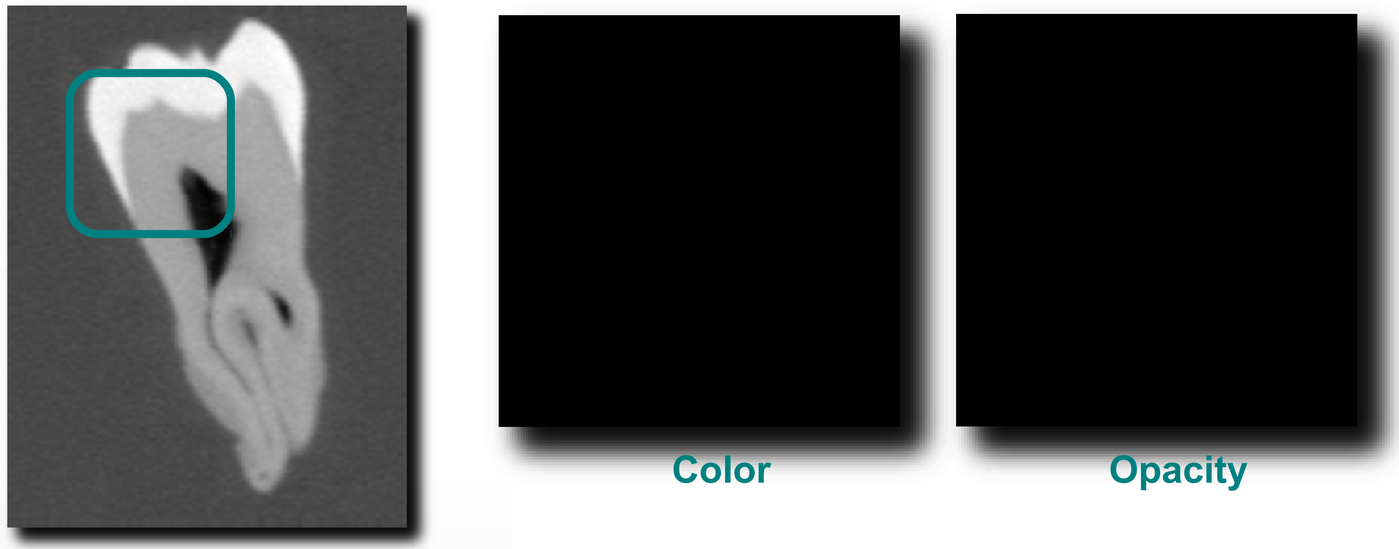
Transfer function examples



Transfer function examples



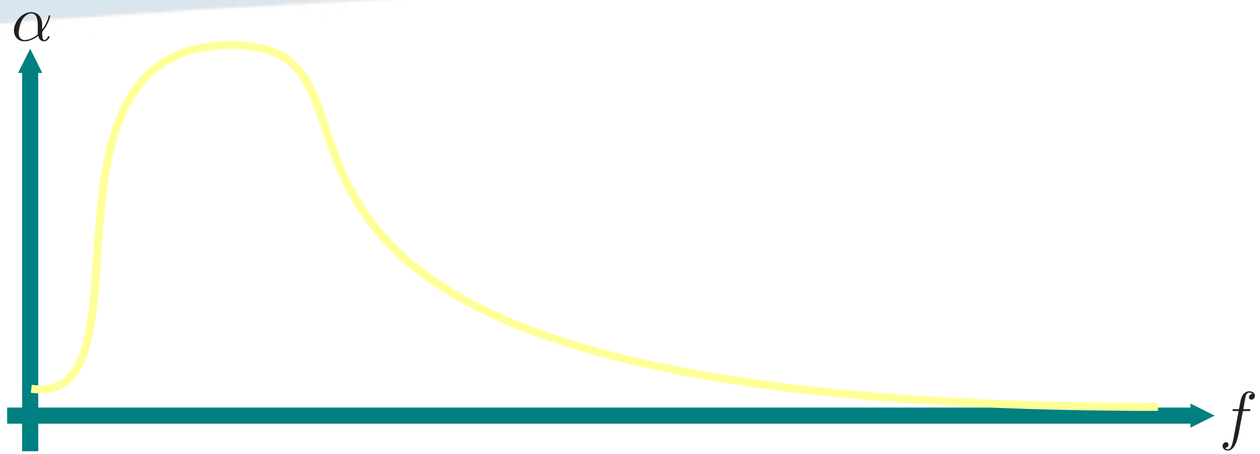
Transfer function examples

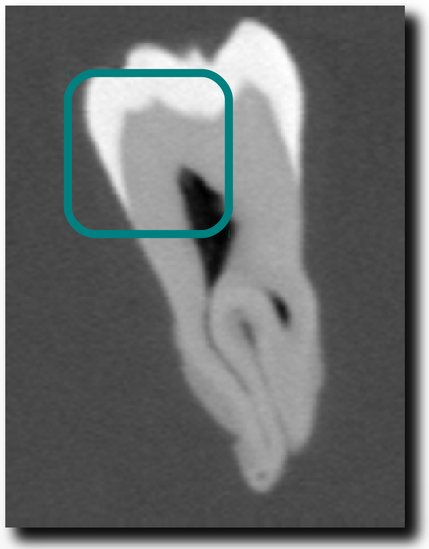


Color

Opacity

Transfer function examples

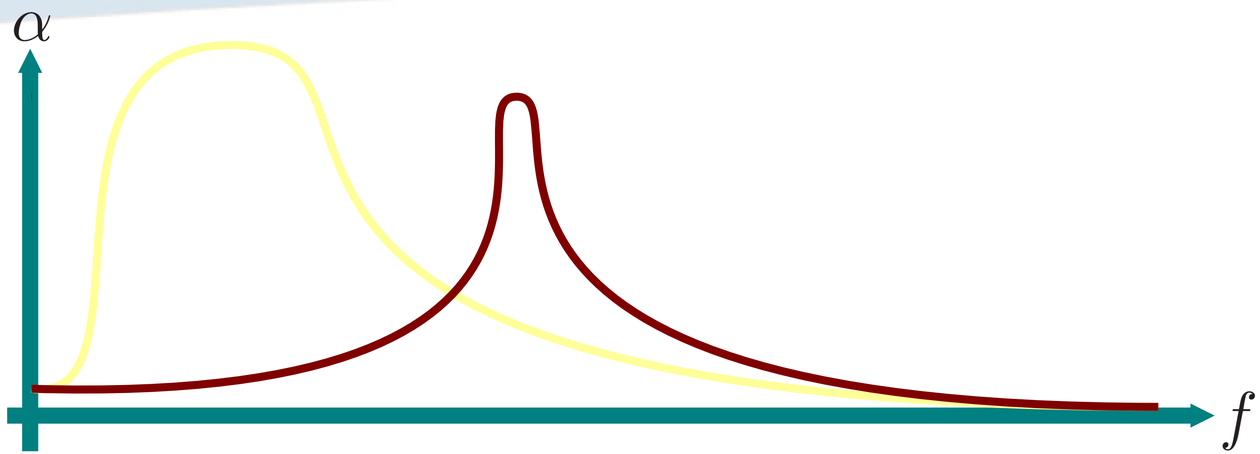


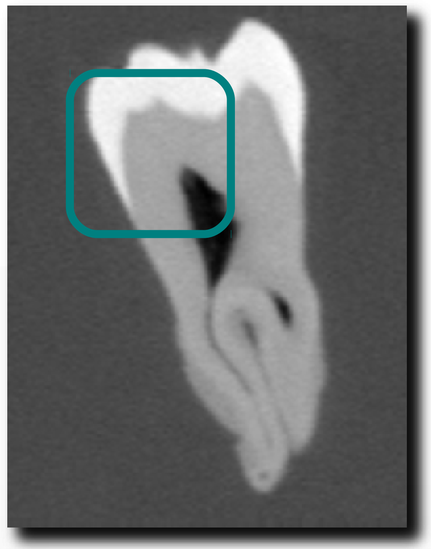


Color

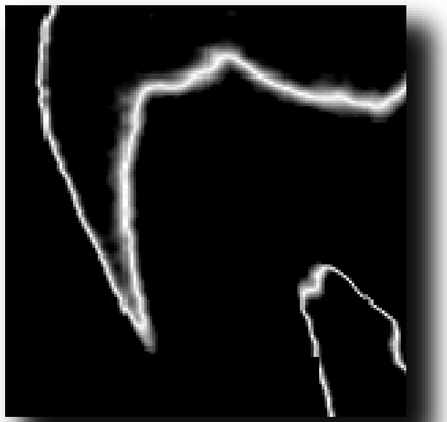
Opacity

Transfer function examples



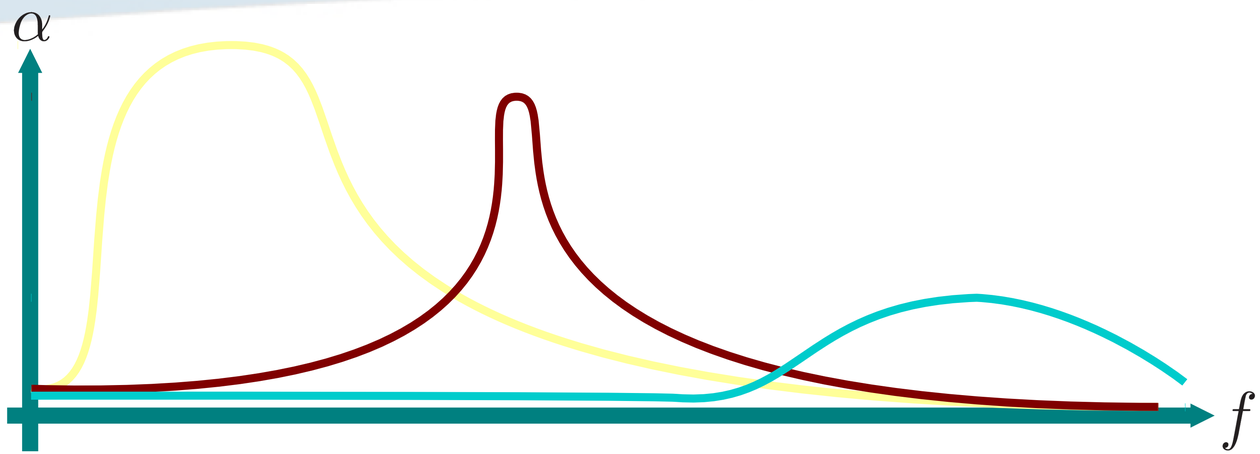


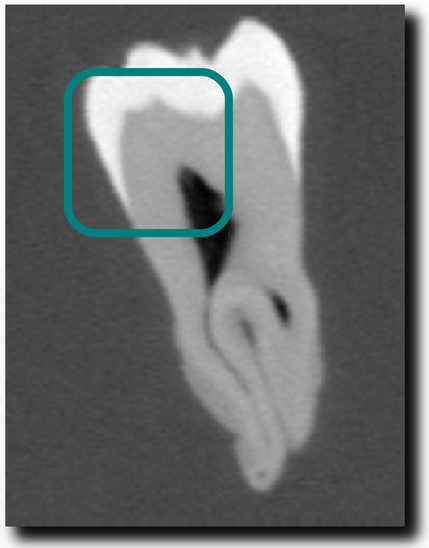
Color

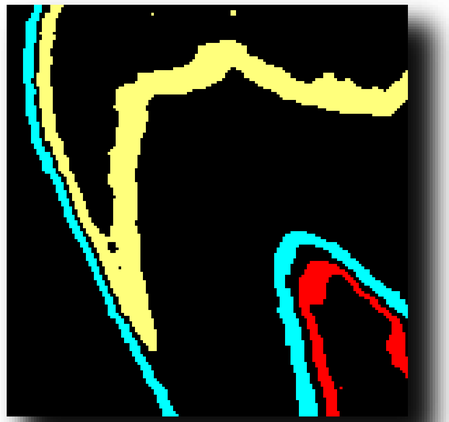


Opacity

Transfer function examples



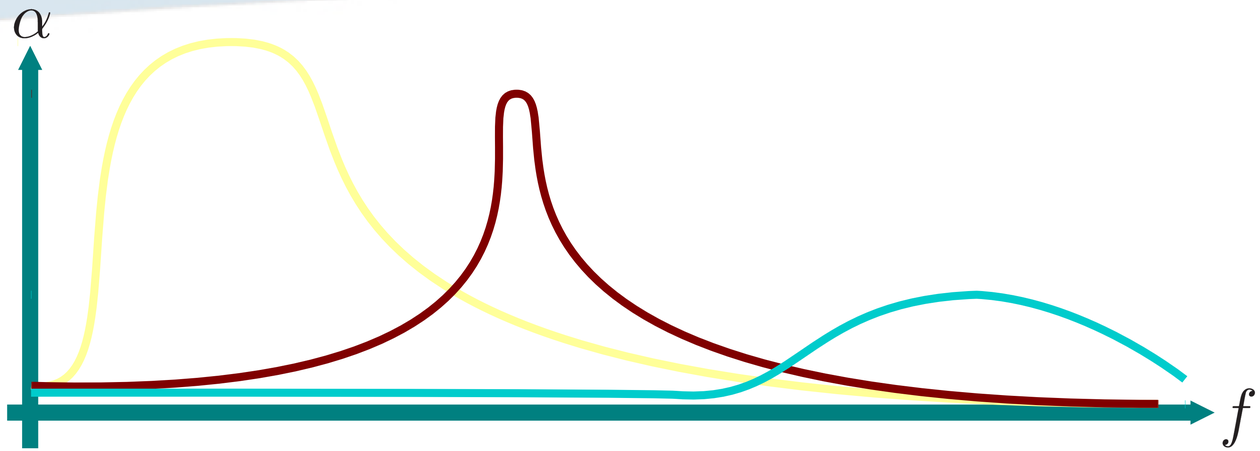


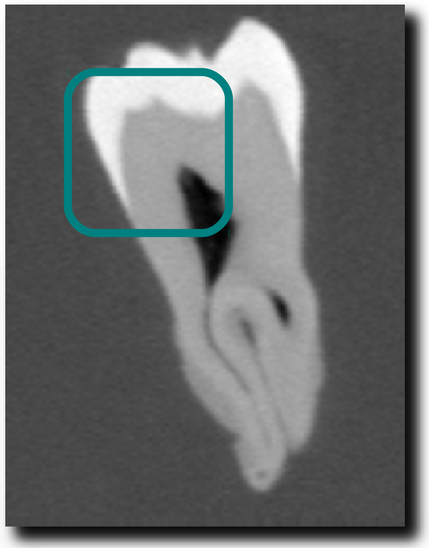


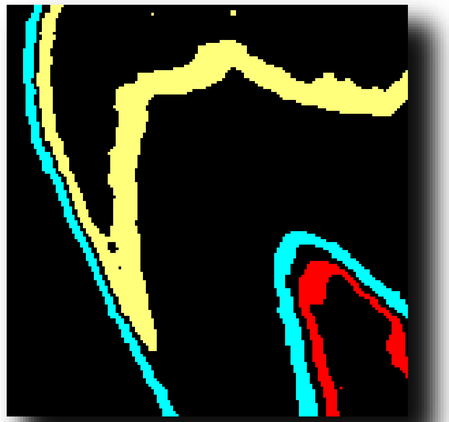
Color

Opacity

Transfer function examples

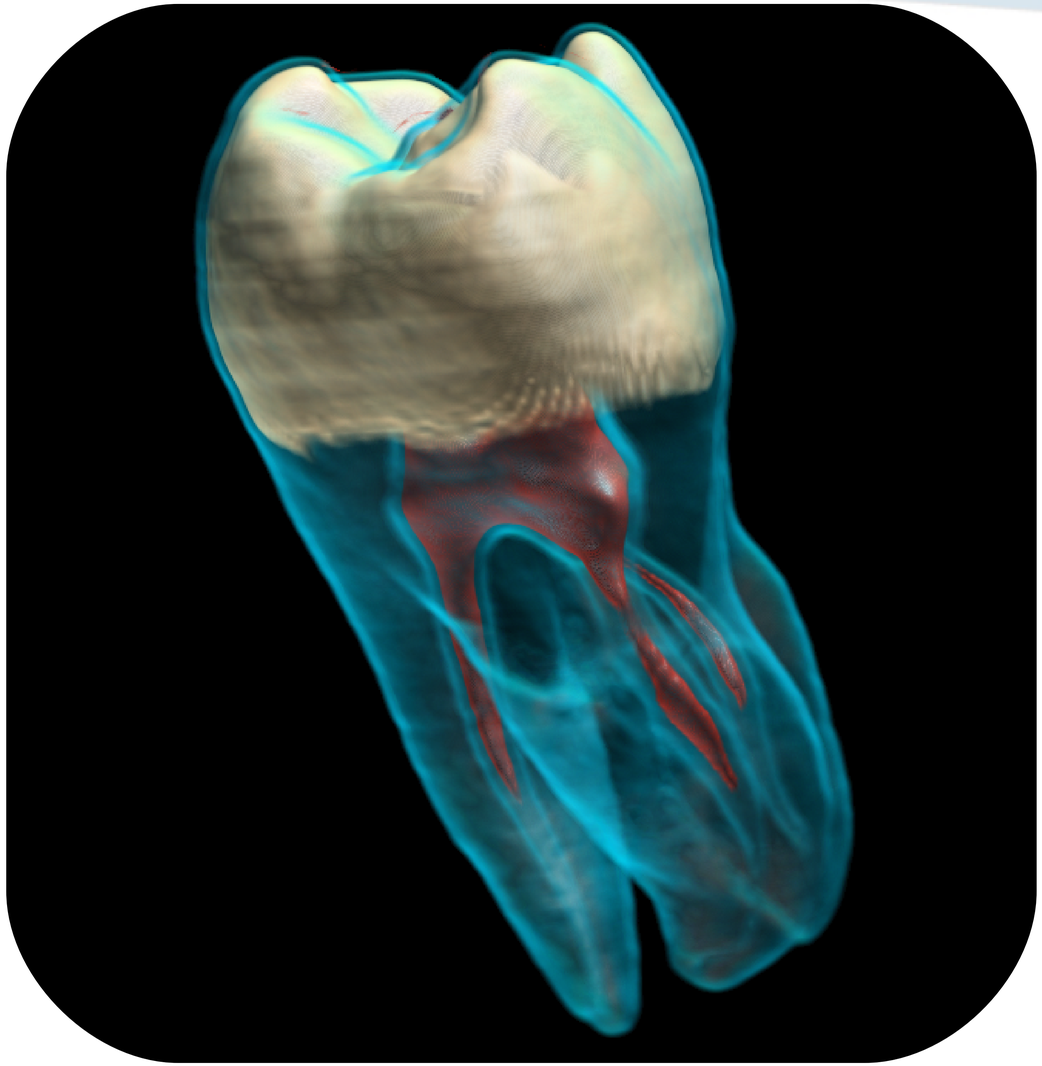






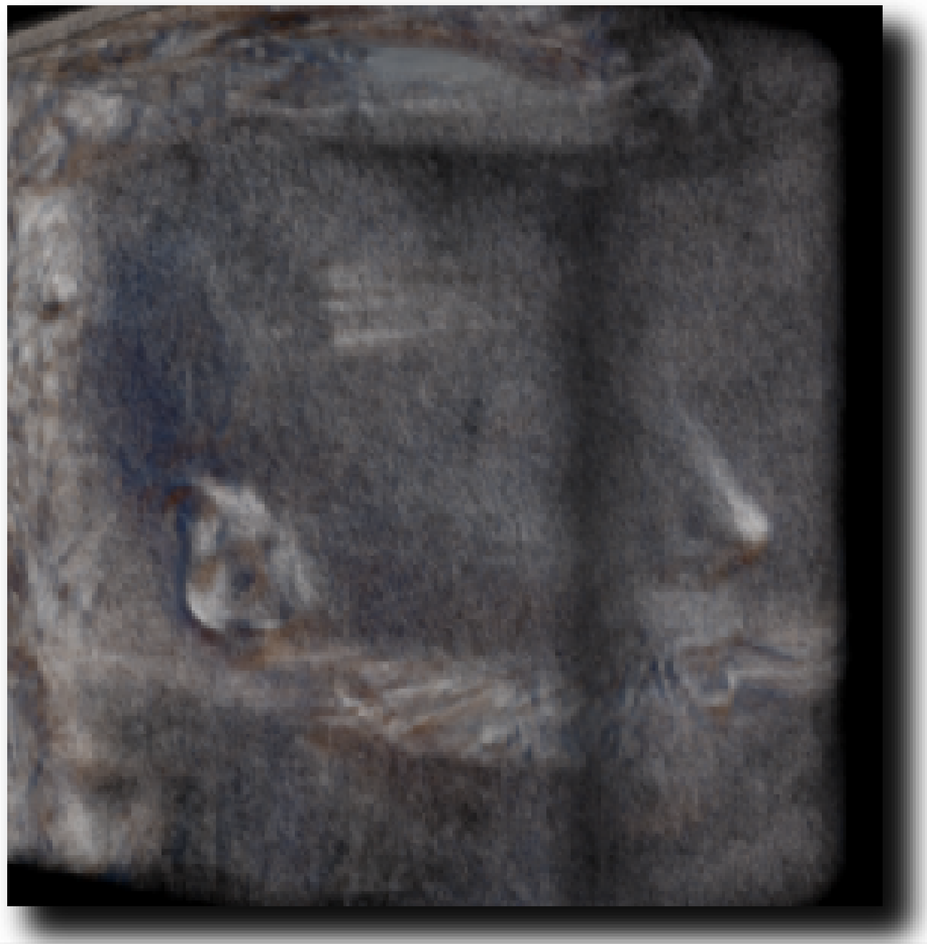
Color

Opacity



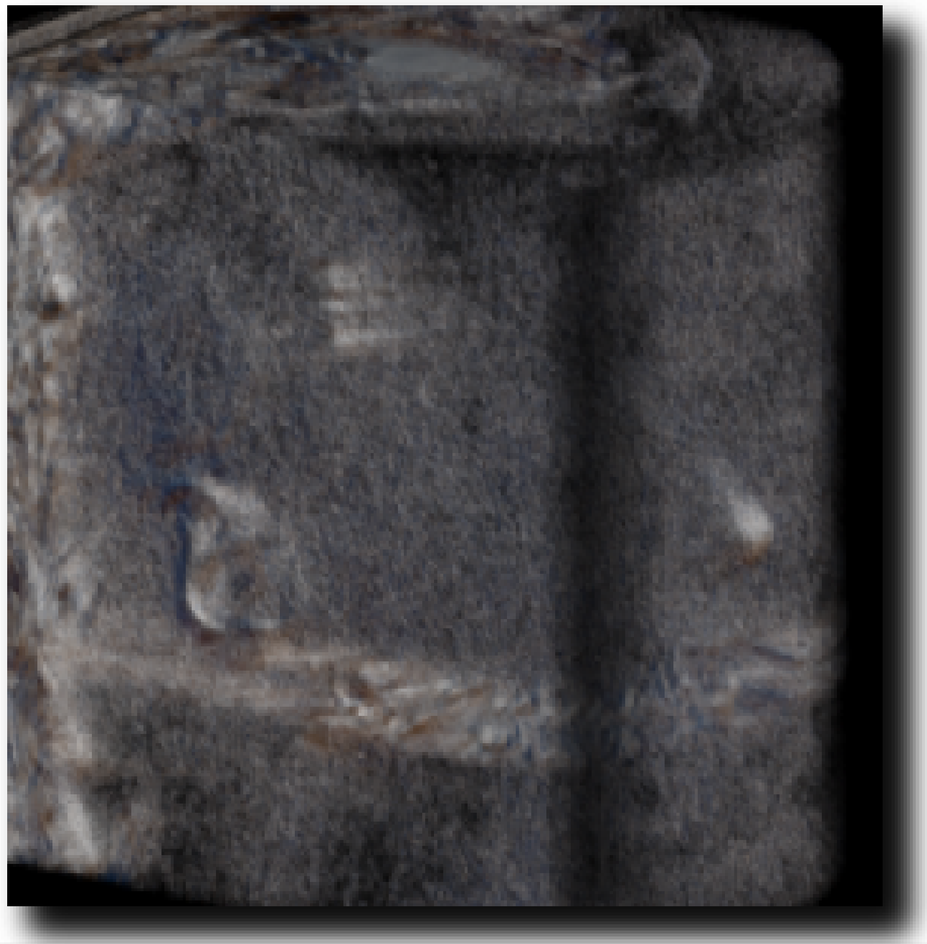
From a user's perspective

- Finding the “right” transfer function can be difficult
  - Experienced users
  - Knowledge of the data-set



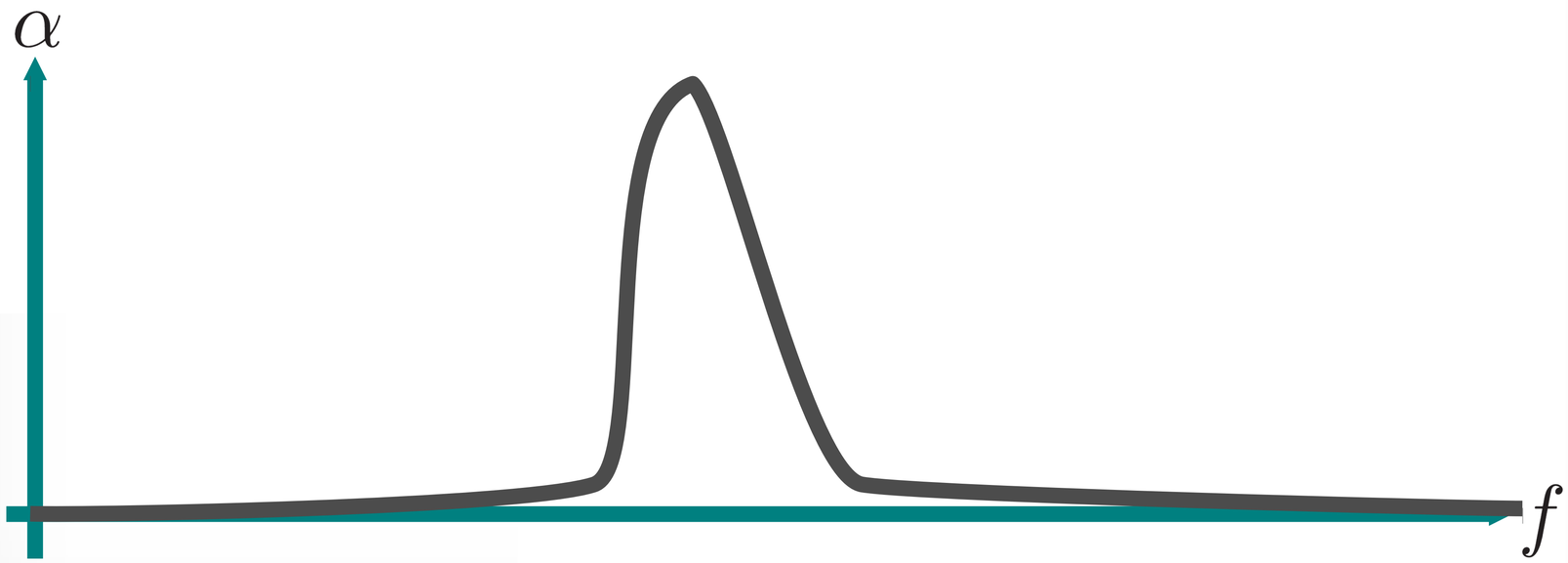
From a user's perspective

- Finding the “right” transfer function can be difficult
  - Experienced users
  - Knowledge of the data-set



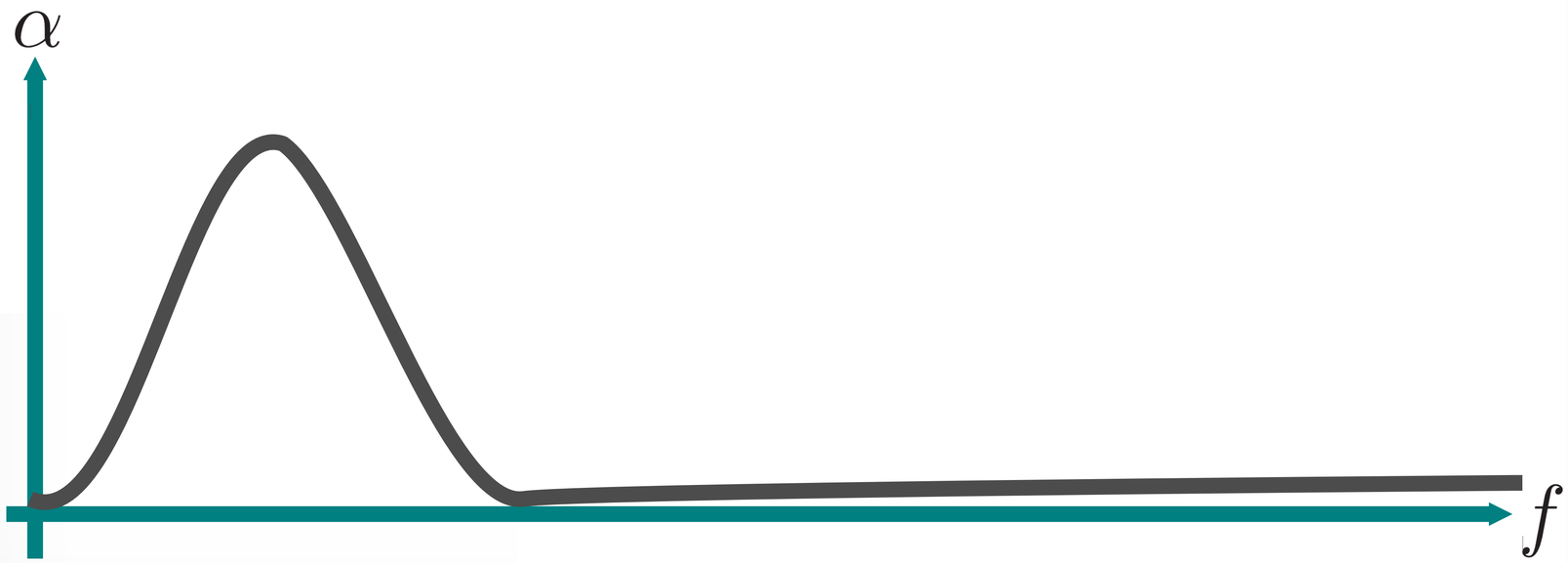
From a user's perspective

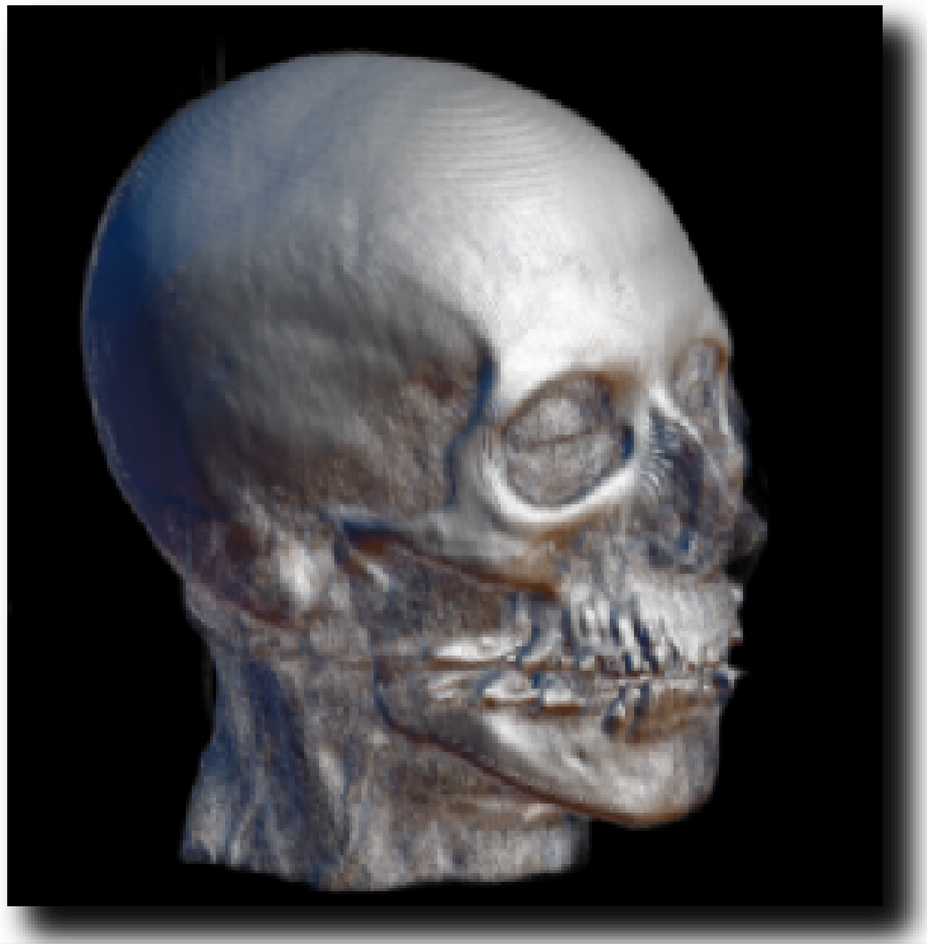
- Finding the “right” transfer function can be difficult
  - Experienced users
  - Knowledge of the data-set



From a user's perspective

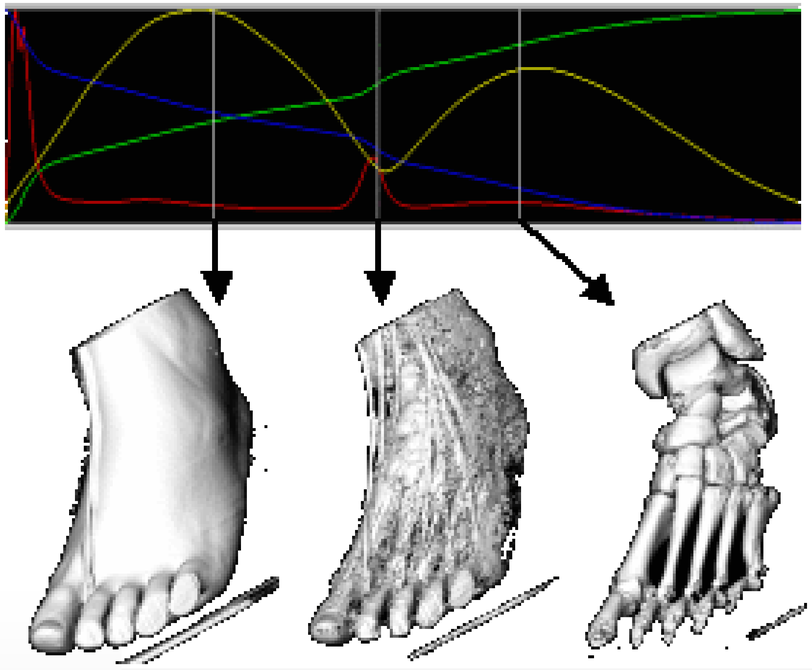
- Finding the “right” transfer function can be difficult
  - Experienced users
  - Knowledge of the data-set

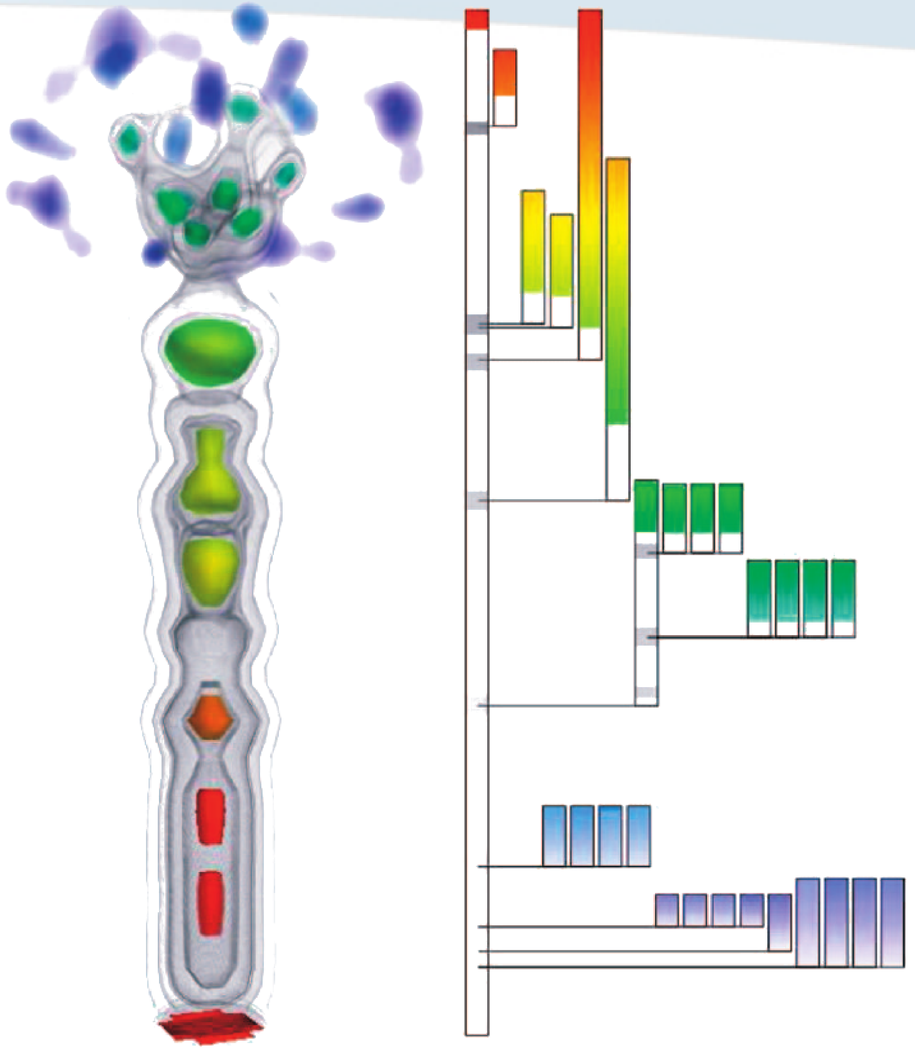




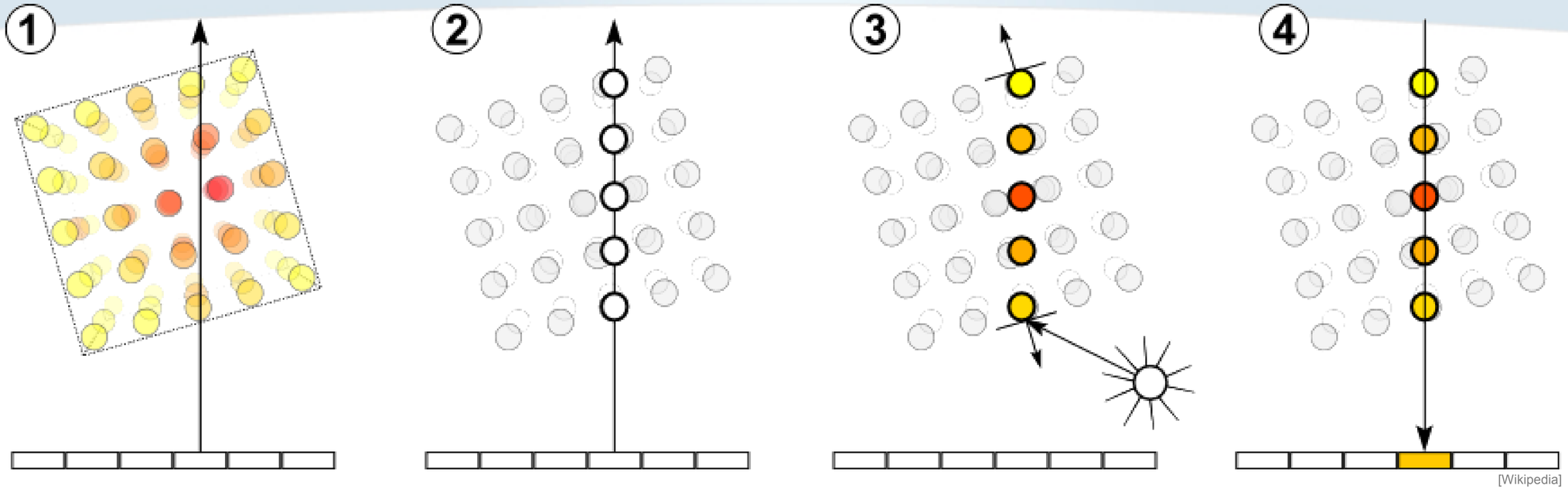
From a user's perspective

- Semi-automated techniques
  - Contour spectrum
  - Topology controlled volume rendering



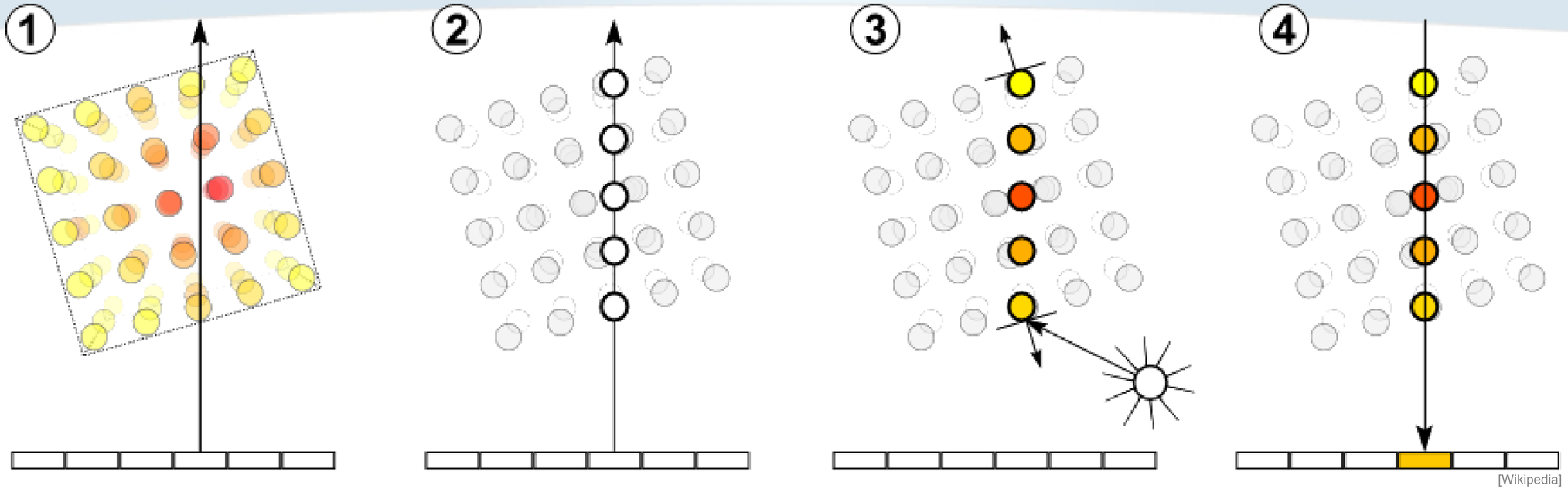


Volume rendering: how does it work?



[Wikipedia]

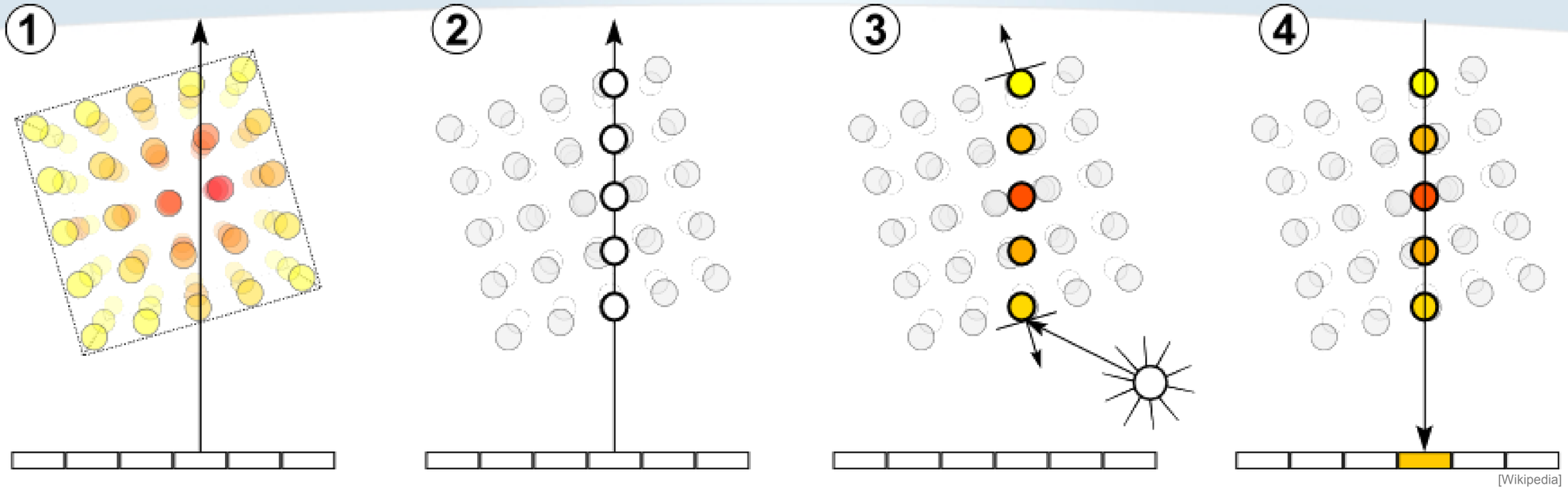
Volume rendering: how does it work?



[Wikipedia]

- Ray casting

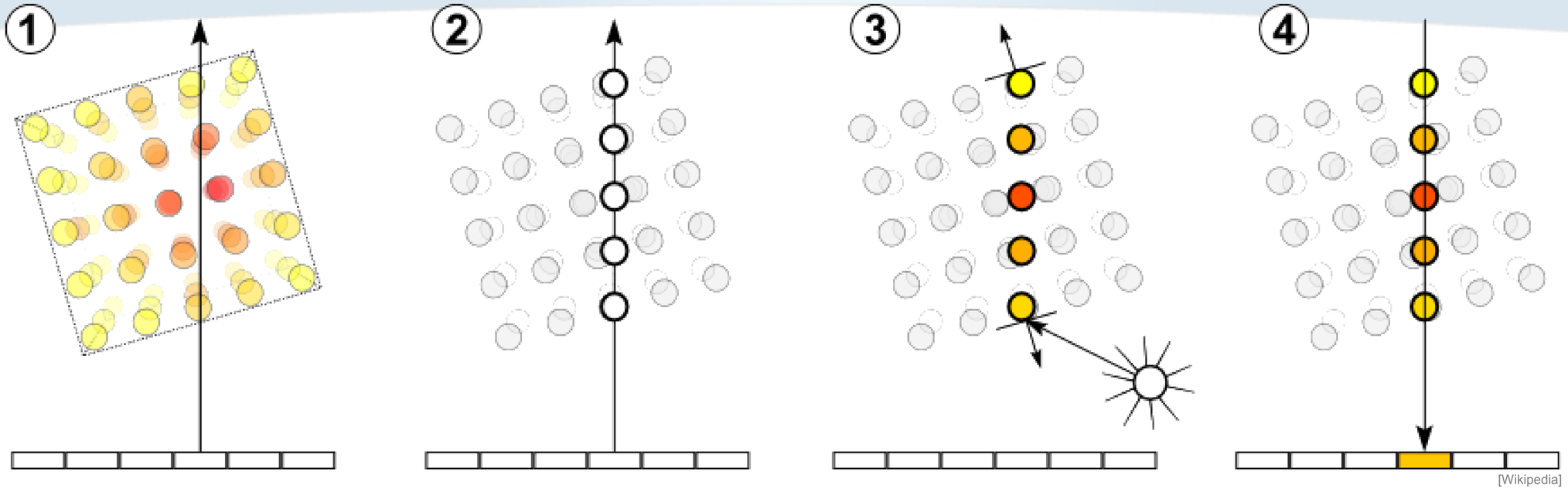
Volume rendering: how does it work?



[Wikipedia]

- Ray casting
- Sampling

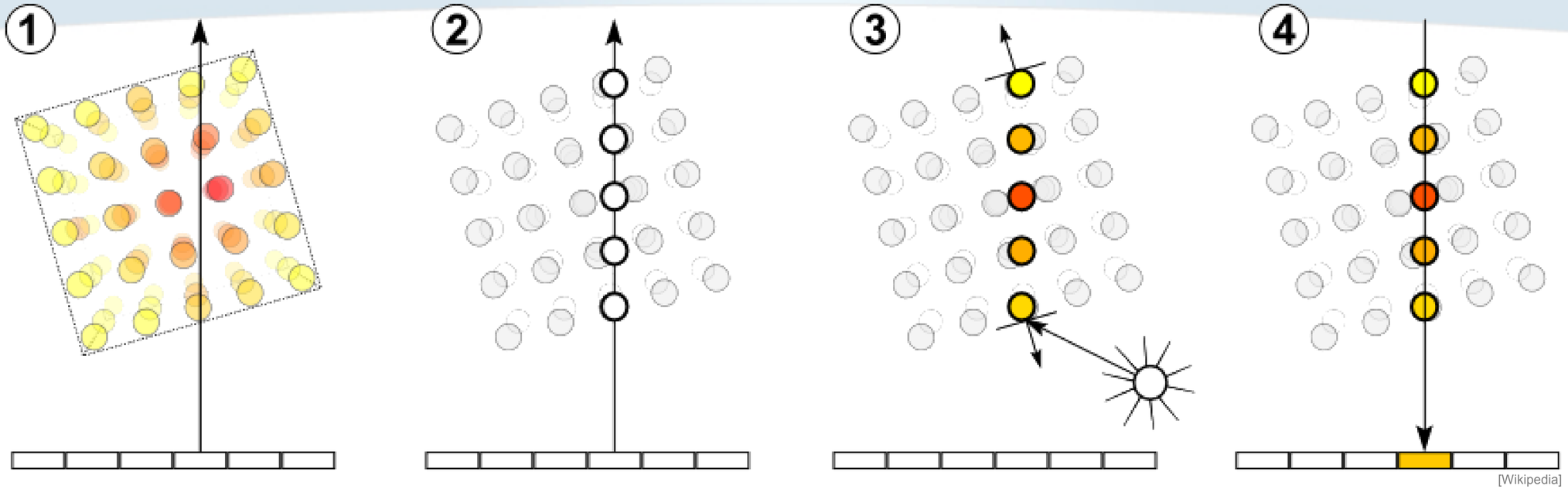
Volume rendering: how does it work?



[Wikipedia]

- Ray casting
- Sampling
- Shading

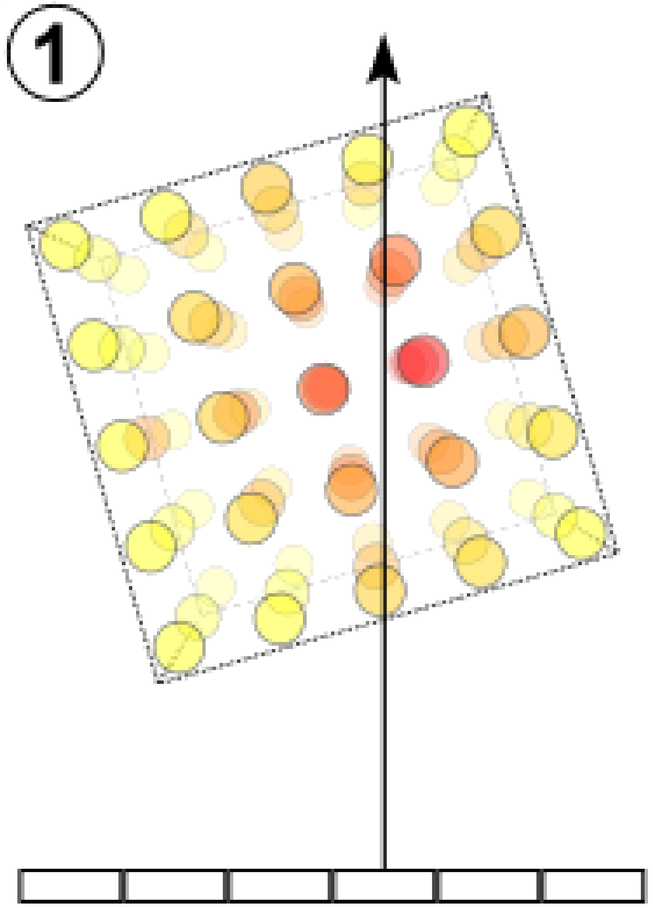
Volume rendering: how does it work?



[Wikipedia]

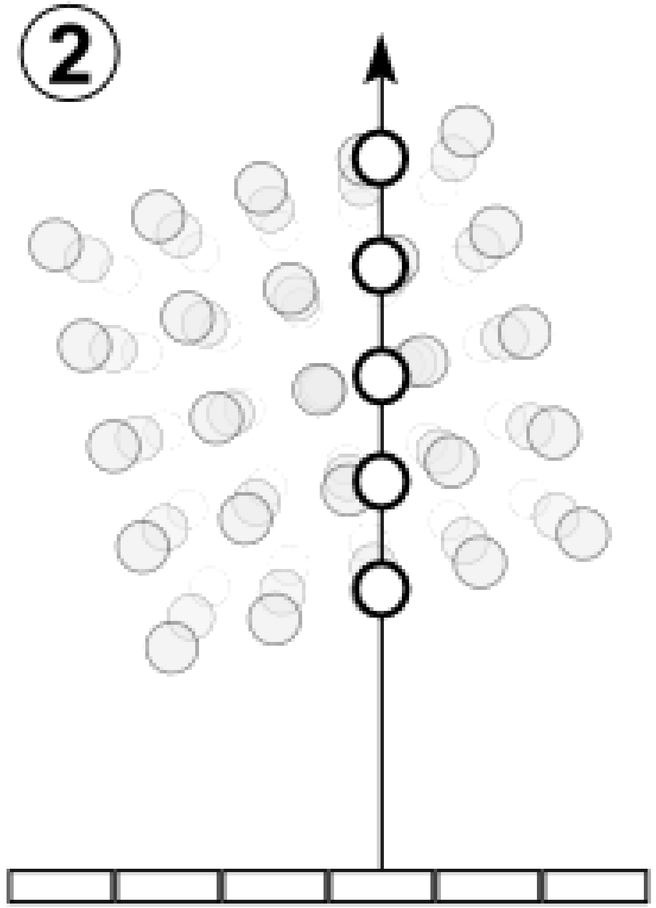
- Ray casting
- Sampling
- Shading
- Compositing

Ray casting



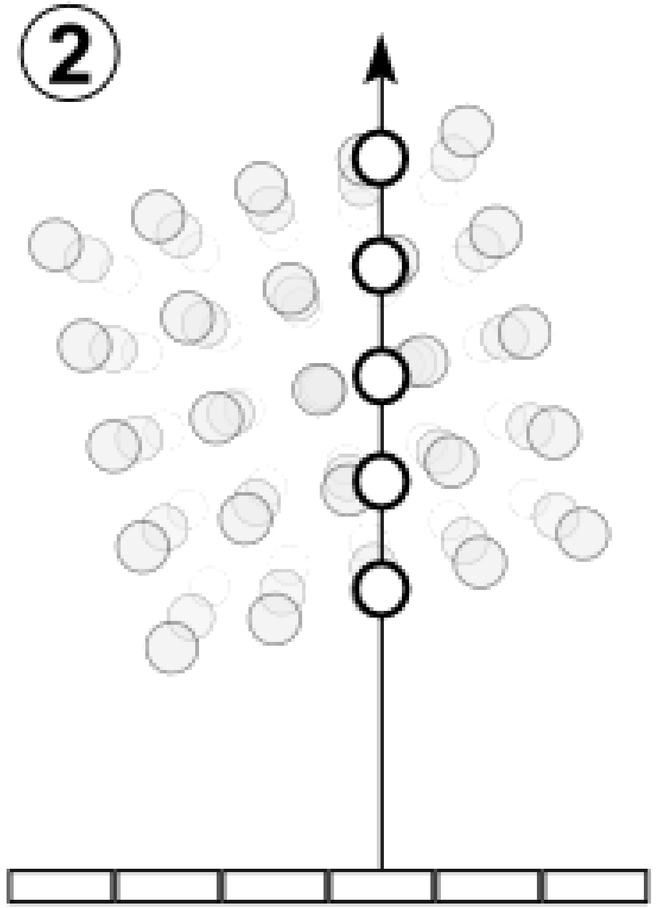
- For each pixel (screen space)
  - Cast a ray
  - Direction of observation
  - Intersection problem
    - Which cells are traversed by the ray?
    - Accelerating structures (octrees)

Sampling



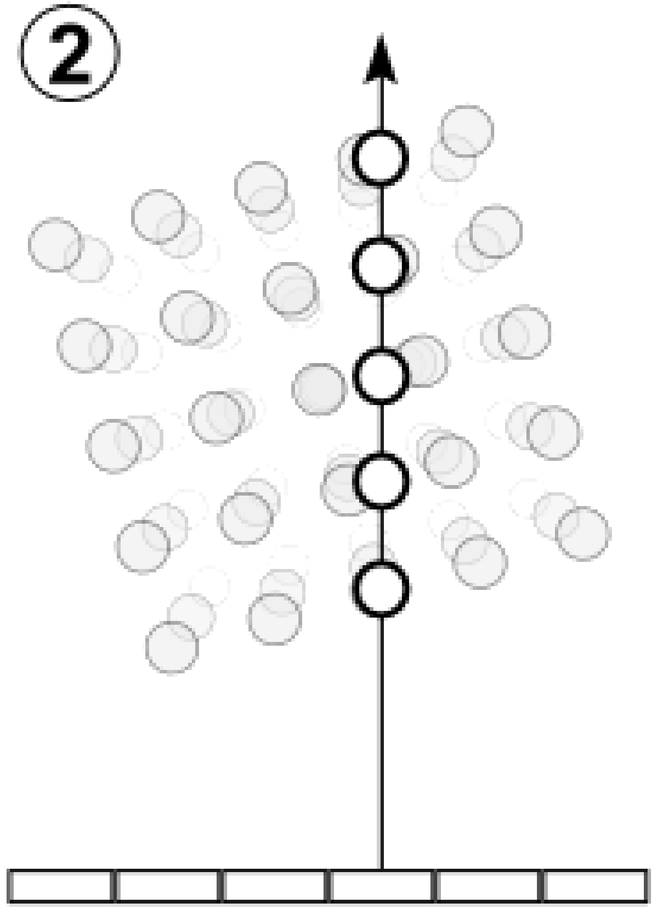
- Along each ray:

Sampling



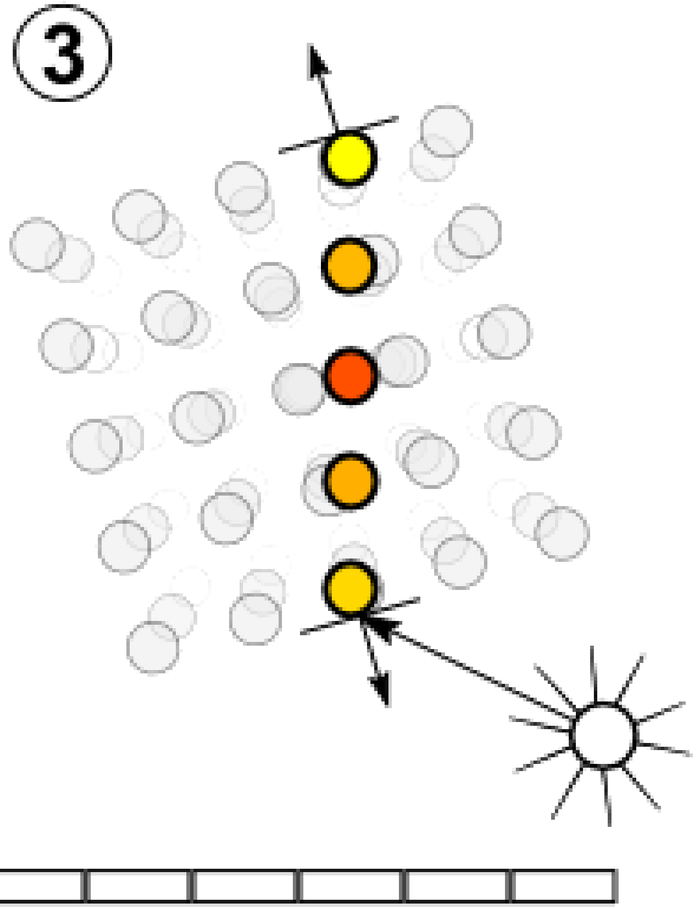
- Along each ray:
  - Sample the data along the ray
    - Intersection with the edges

Sampling



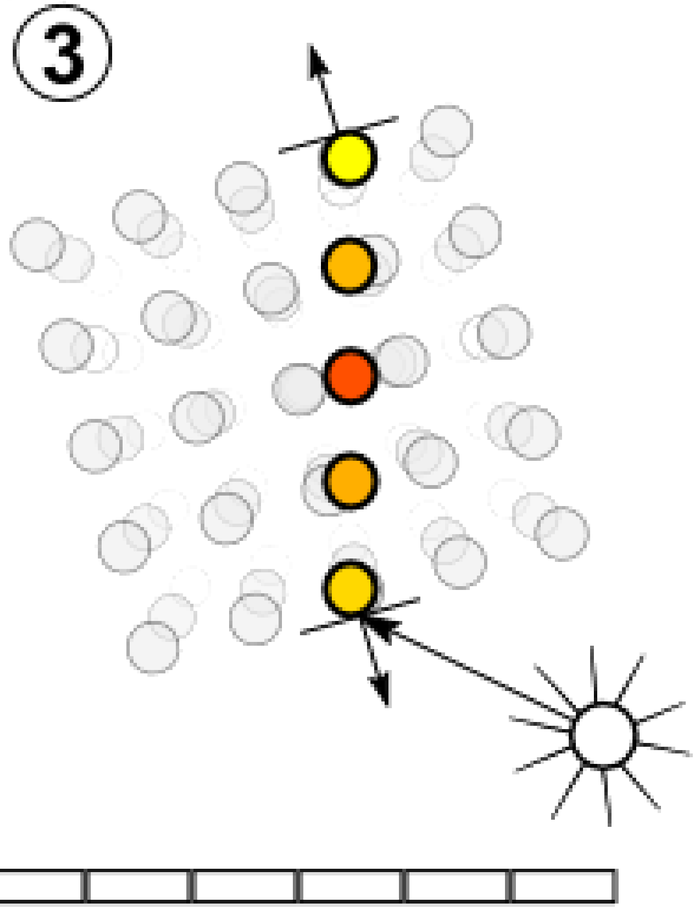
- Along each ray:
  - Sample the data along the ray
    - Intersection with the edges
  - Compute the function value on the samples
    - Based on the domain interpolant

Shading



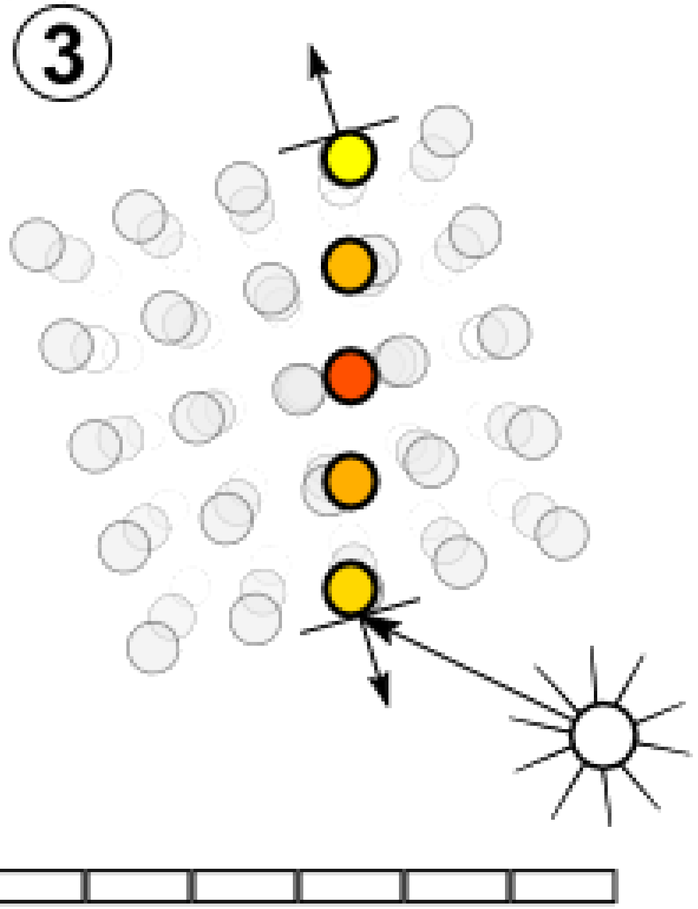
- Light computation for each sample

Shading



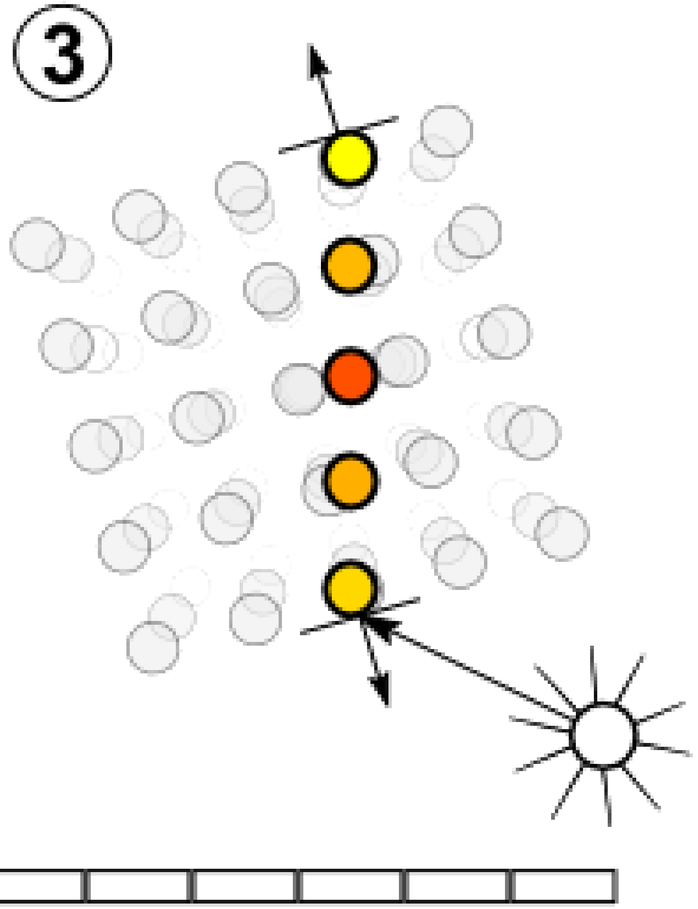
- Light computation for each sample
  - Retrieve the corresponding color (transfer)

Shading



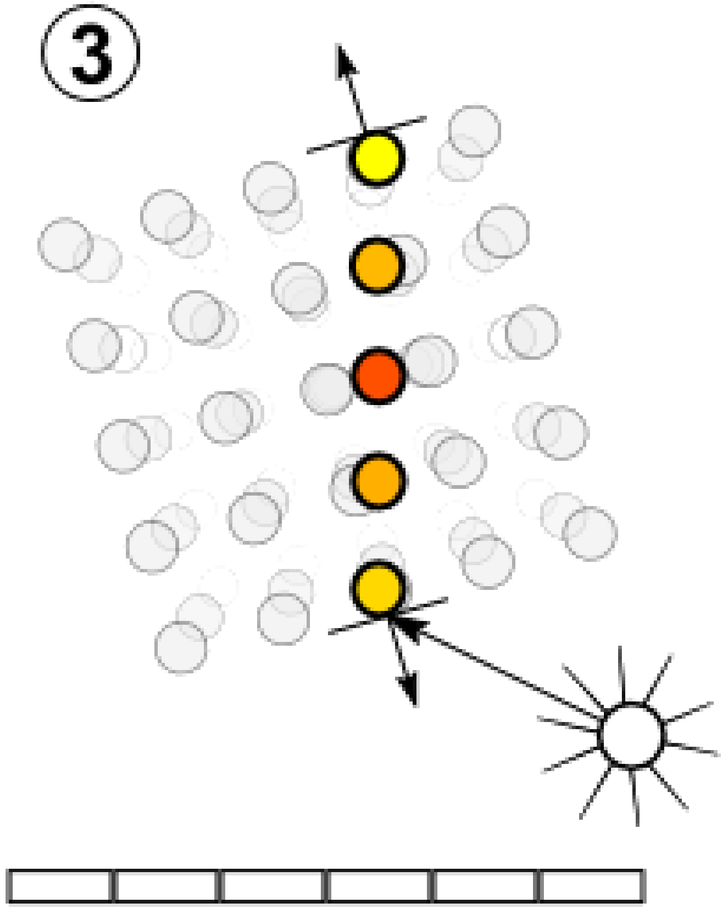
- Light computation for each sample
  - Retrieve the corresponding color (transfer)
  - Retrieve the surface normal

Shading



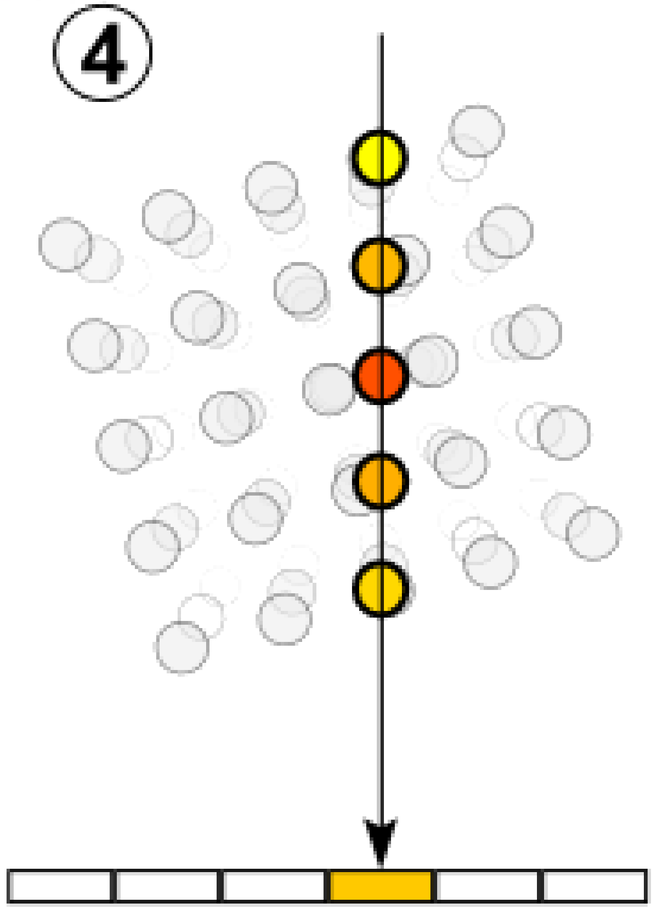
- Light computation for each sample
  - Retrieve the corresponding color (transfer)
  - Retrieve the surface normal
    - Normal to the isosurface

Shading



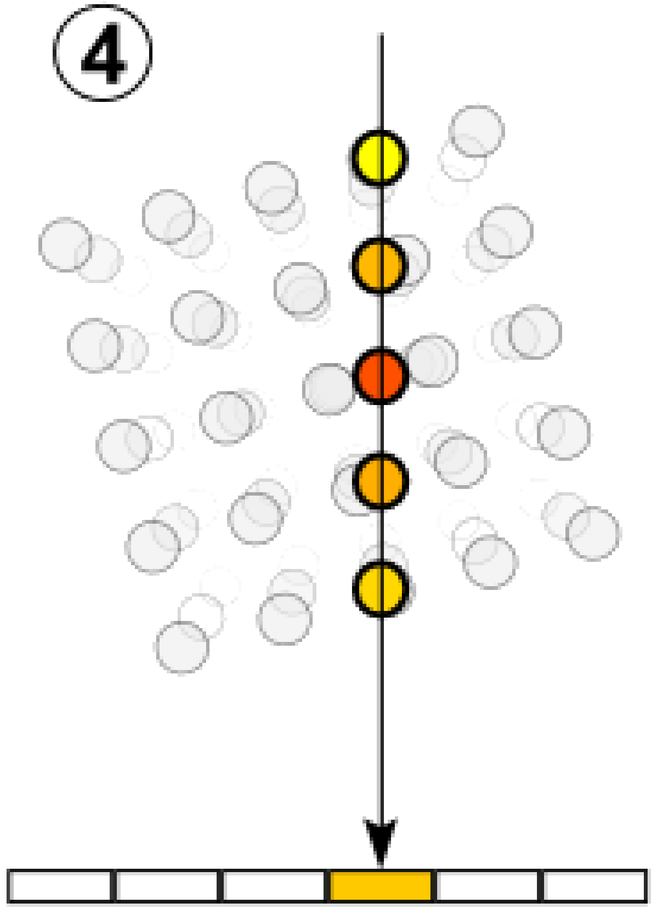
- Light computation for each sample
  - Retrieve the corresponding color (transfer)
  - Retrieve the surface normal
    - Normal to the isosurface
  - Shade the sample accordingly
    - Normal
    - View direction
    - Color

Compositing



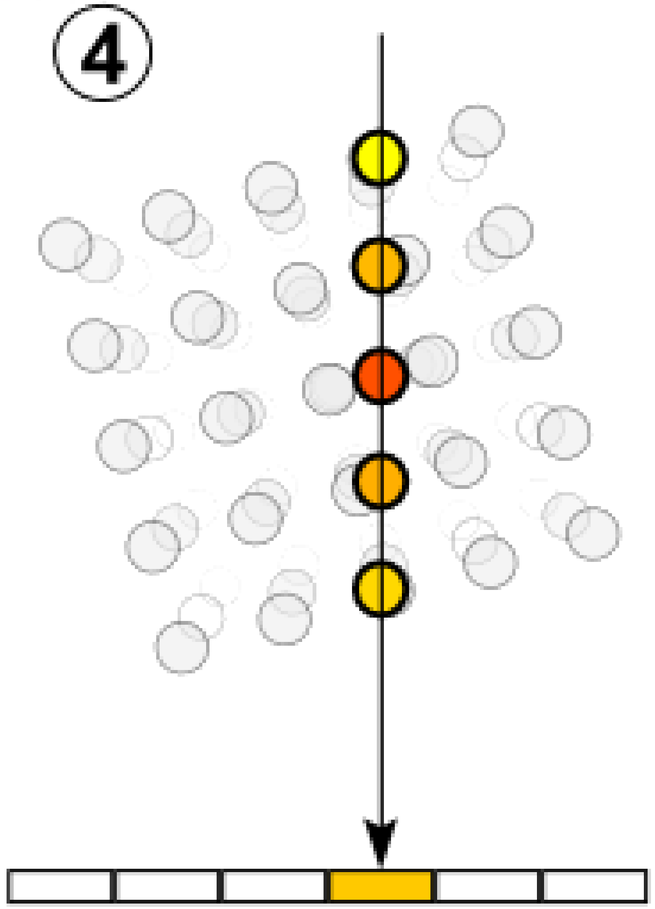
- Integrate all the samples' contributions

Compositing



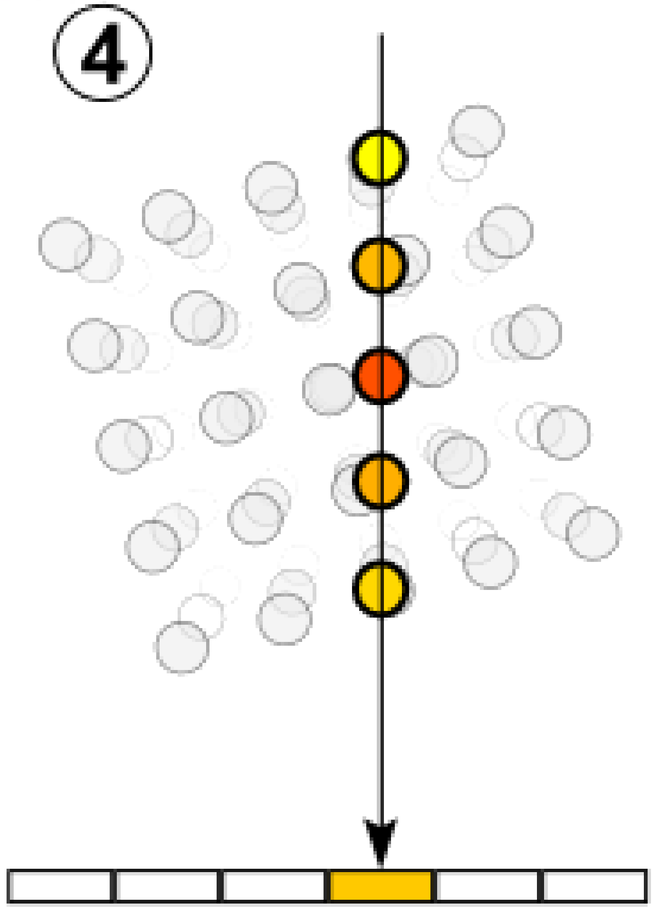
- Integrate all the samples' contributions
  - Along each ray
    - From the back to the front

Compositing



- Integrate all the samples' contributions
  - Along each ray
    - From the back to the front
    - At each sample
      - Retrieve the opacity value (transfer)

Compositing



- Integrate all the samples' contributions
  - Along each ray
    - From the back to the front
    - At each sample
      - Retrieve the opacity value (transfer)
  - Composition along the ray
    - Blend the color of the samples
    - Based on the opacity of the samples

Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

Color blending

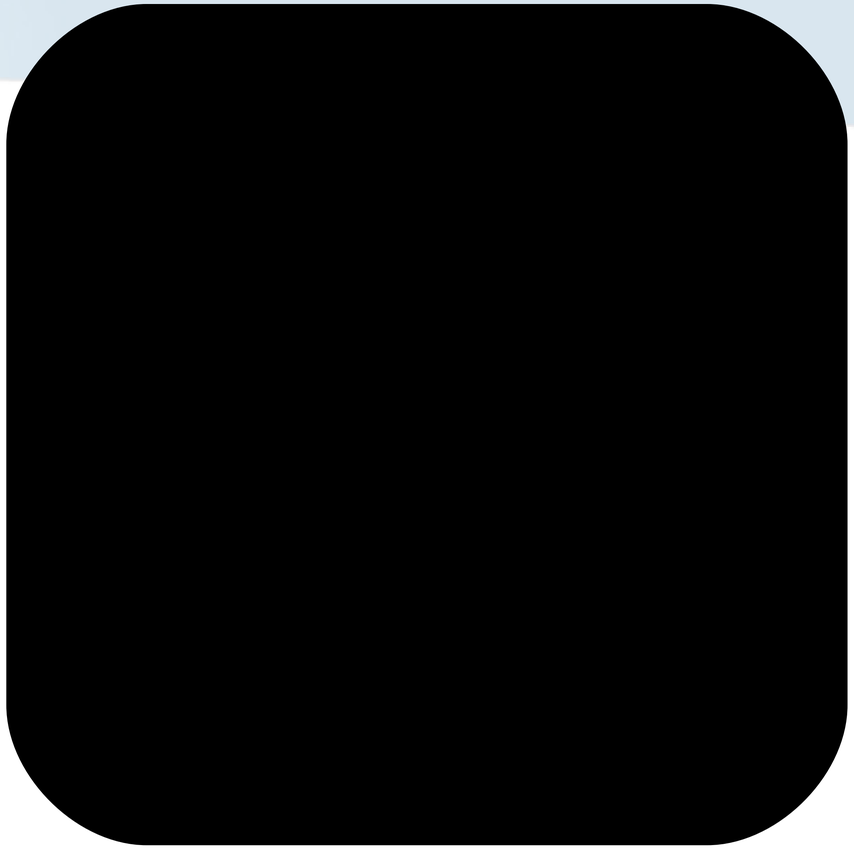
- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

- Order matters!

Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

- Order matters!
  - $C_a = (0, 0, 0),\ \alpha_a = 1$

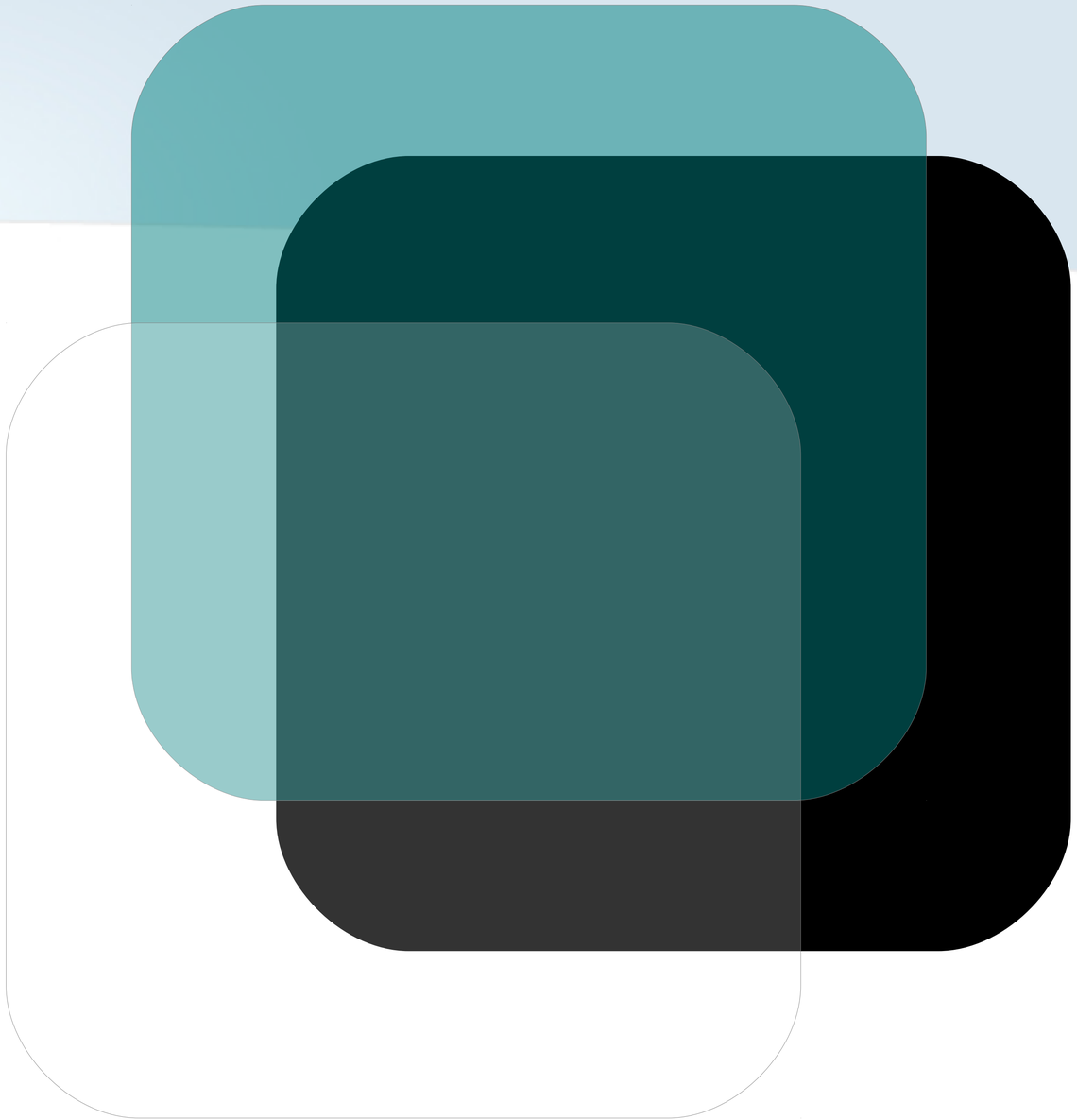


Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

- Order matters!
  - $C_a = (0, 0, 0),\ \alpha_a = 1$
  - $C_b = (0, 1, 0),\ \alpha_b = 0.5$

Color blending



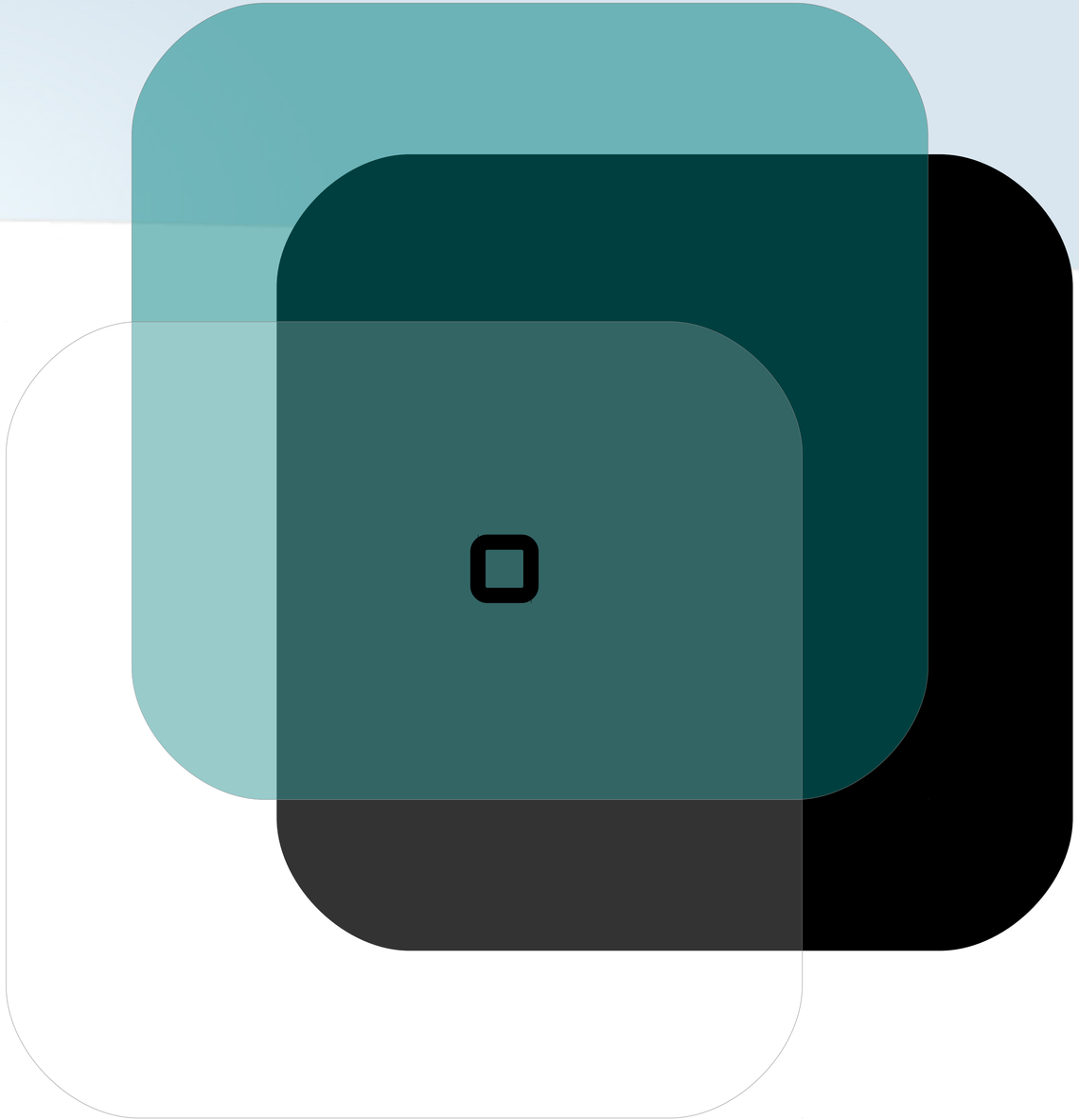
- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

- Order matters!
  - $C_a = (0, 0, 0),\ \alpha_a = 1$
  - $C_b = (0, 1, 0),\ \alpha_b = 0.5$
  - $C_c = (1, 1, 1),\ \alpha_c = 0.1$

Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

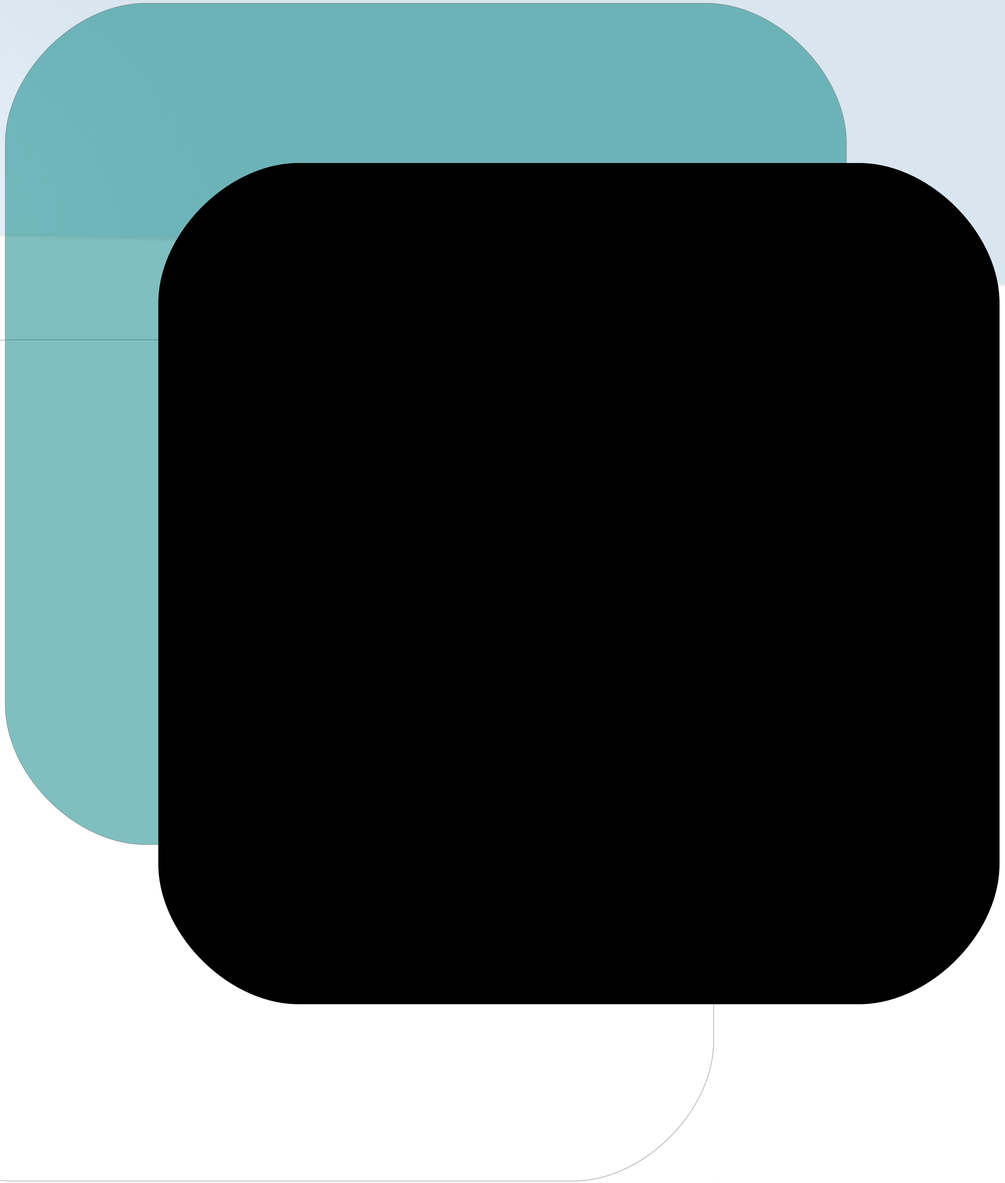
- Order matters!
  - $C_a = (0, 0, 0),\ \alpha_a = 1$
  - $C_b = (0, 1, 0),\ \alpha_b = 0.5$
  - $C_c = (1, 1, 1),\ \alpha_c = 0.1$



Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

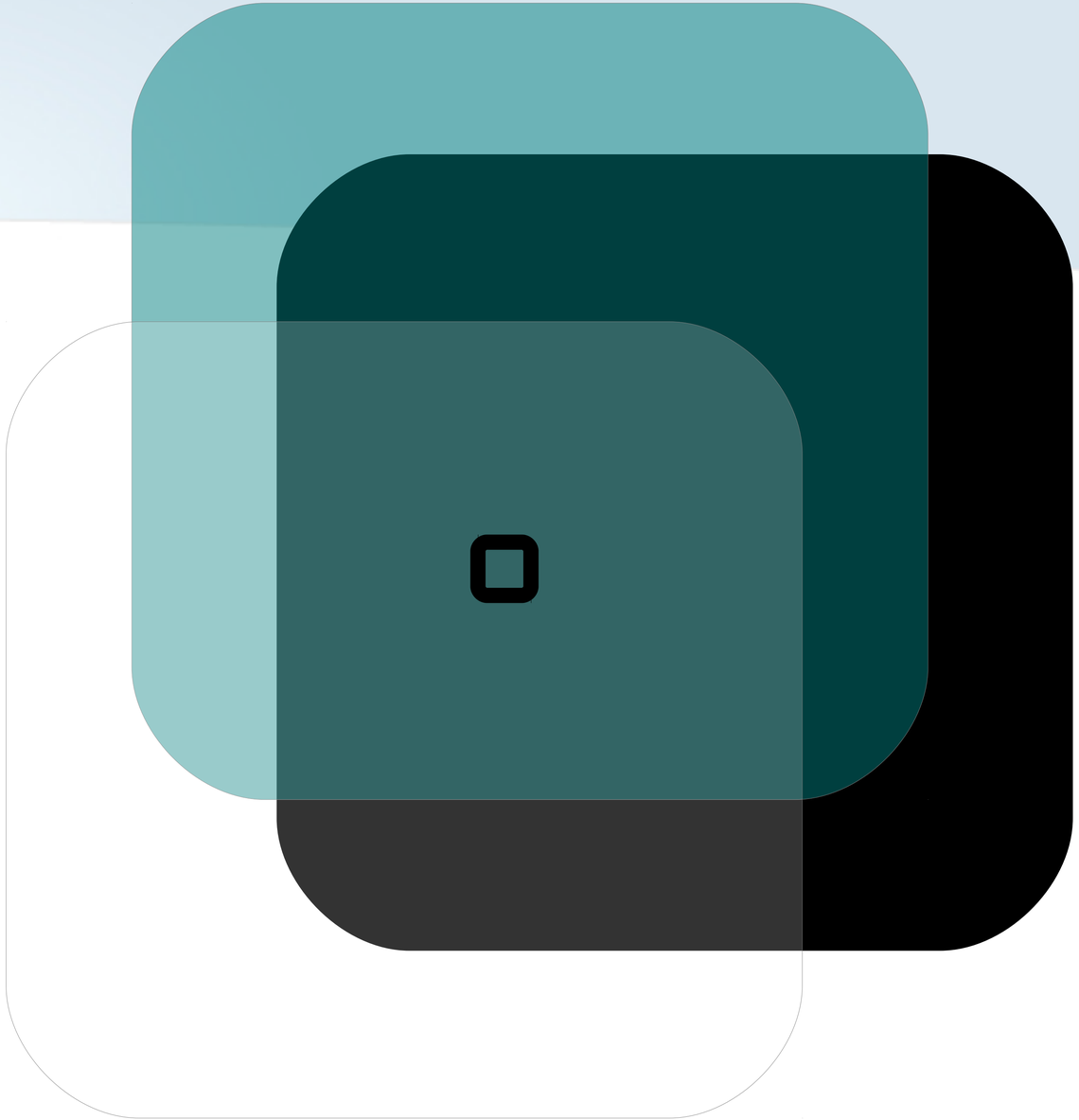
- Order matters!
  - $C_a = (0, 0, 0),\ \alpha_a = 1$
  - $C_b = (0, 1, 0),\ \alpha_b = 0.5$
  - $C_c = (1, 1, 1),\ \alpha_c = 0.1$



Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

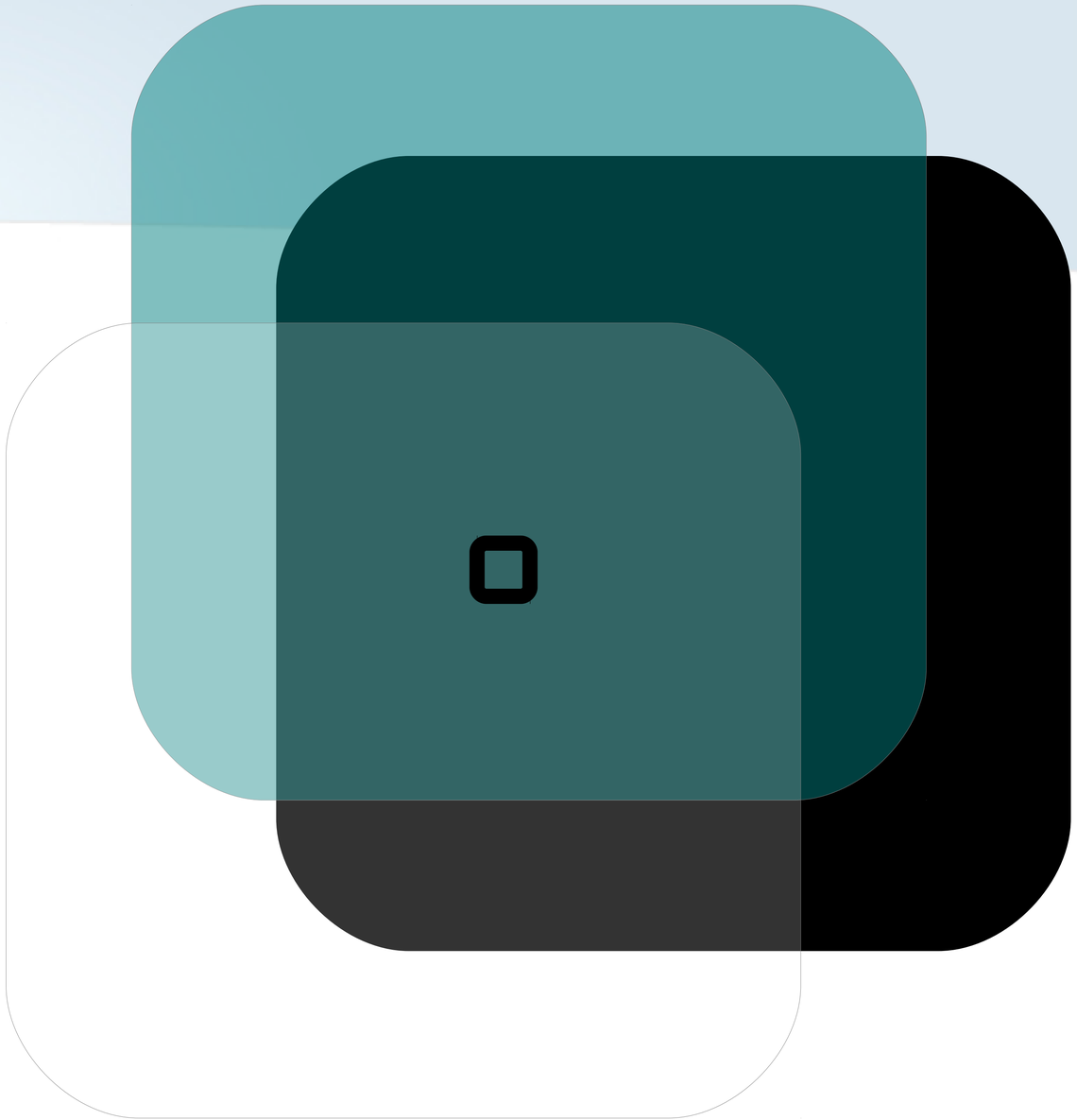
- Order matters!
  - $C_a = (0, 0, 0),\ \alpha_a = 1$
  - $C_b = (0, 1, 0),\ \alpha_b = 0.5$
  - $C_c = (1, 1, 1),\ \alpha_c = 0.1$



Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

- Order matters!

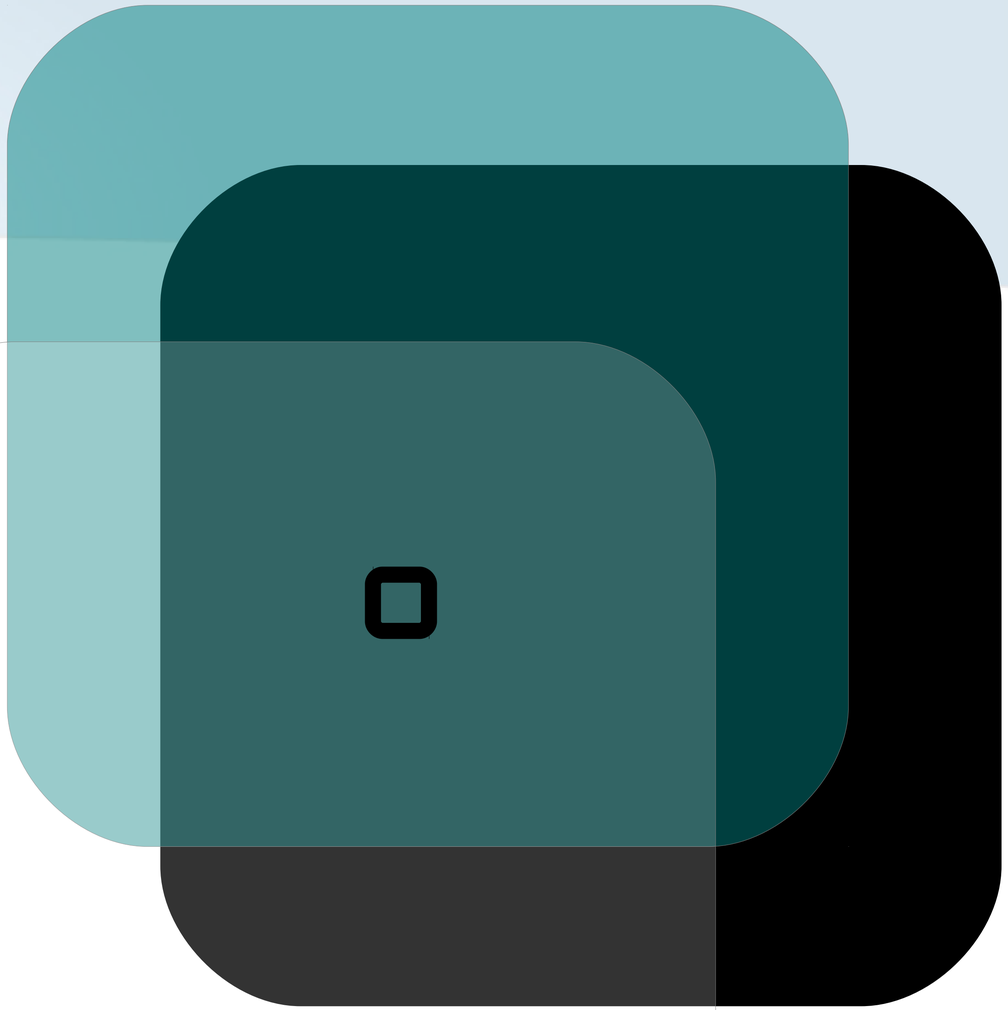


$$C'(i) = \alpha(i) \times C(i) + (1 - \alpha(i)) \times \alpha(i-1) \times C(i-1)$$

Color blending

- Screen space operation
  - Blend the colors of the samples
  - Based on their opacity

- Order matters!



$$\alpha'(i) = \alpha(i) + (1 - \alpha(i)) \times \alpha(i-1)$$

$$C'(i) = \alpha(i) \times C(i) + (1 - \alpha(i)) \times \alpha(i-1) \times C(i-1)$$

Compositing schemes

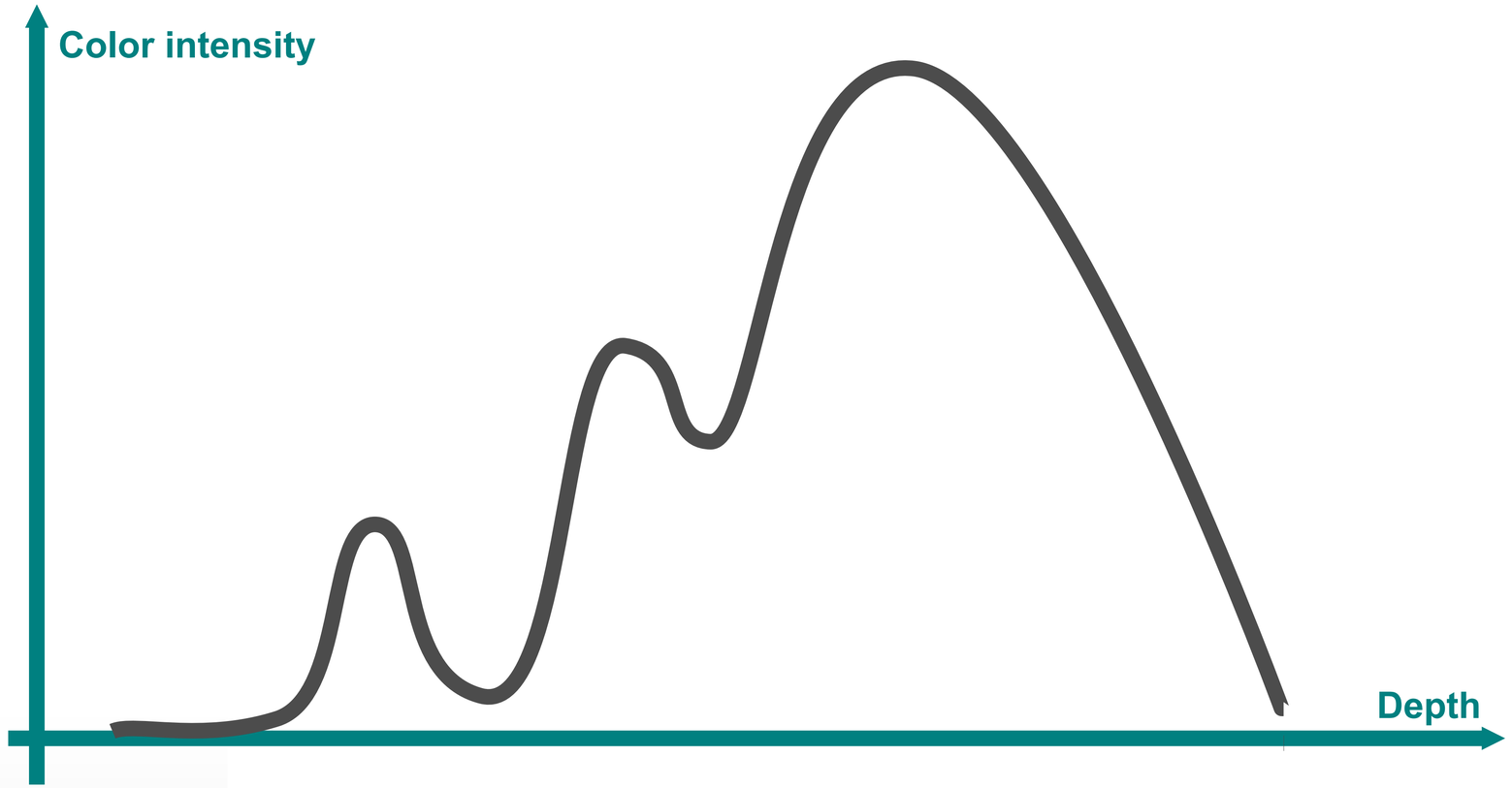
- Color intensity along the ray

Compositing schemes

- Color intensity along the ray

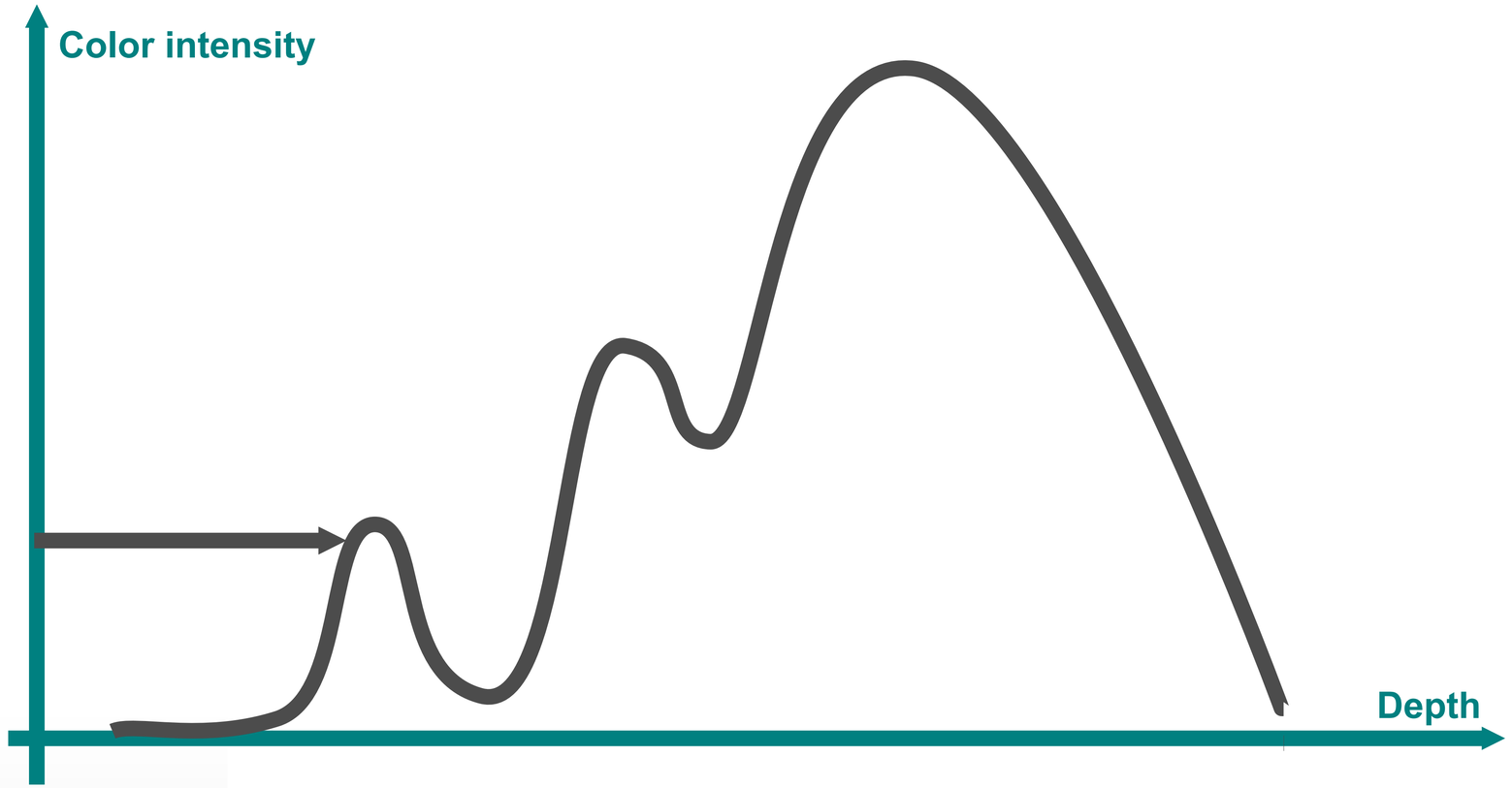
Compositing schemes

- Color intensity along the ray



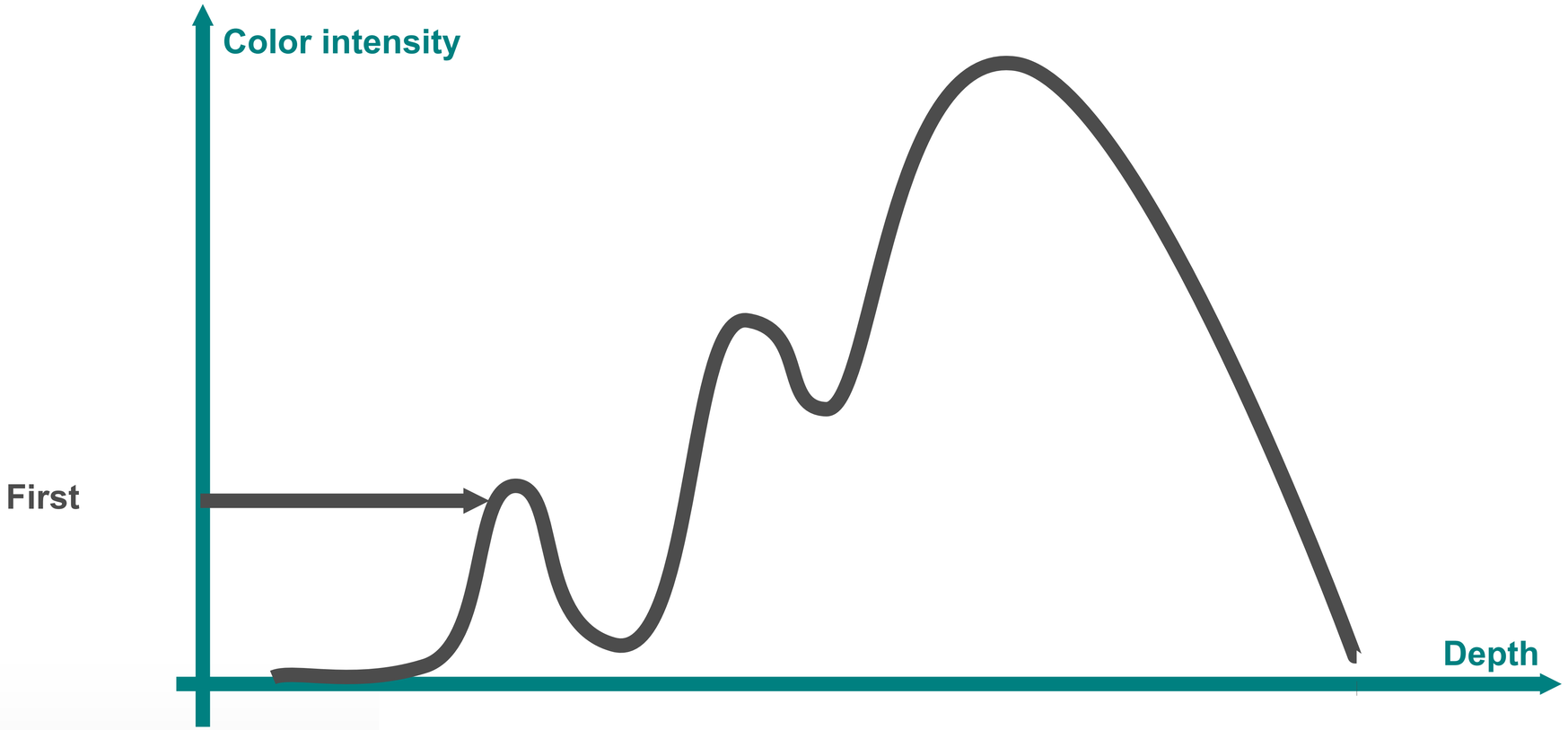
Compositing schemes

- Color intensity along the ray



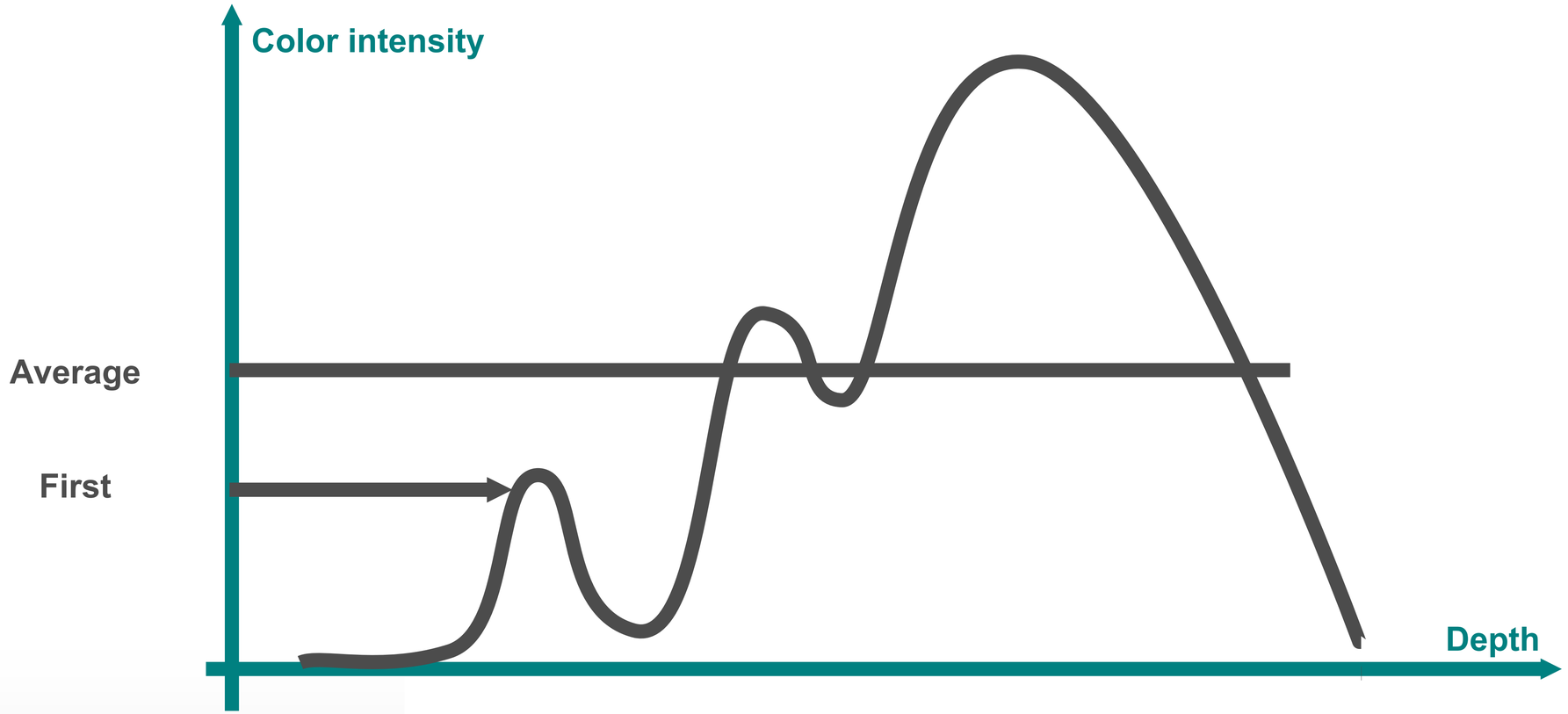
Compositing schemes

- Color intensity along the ray



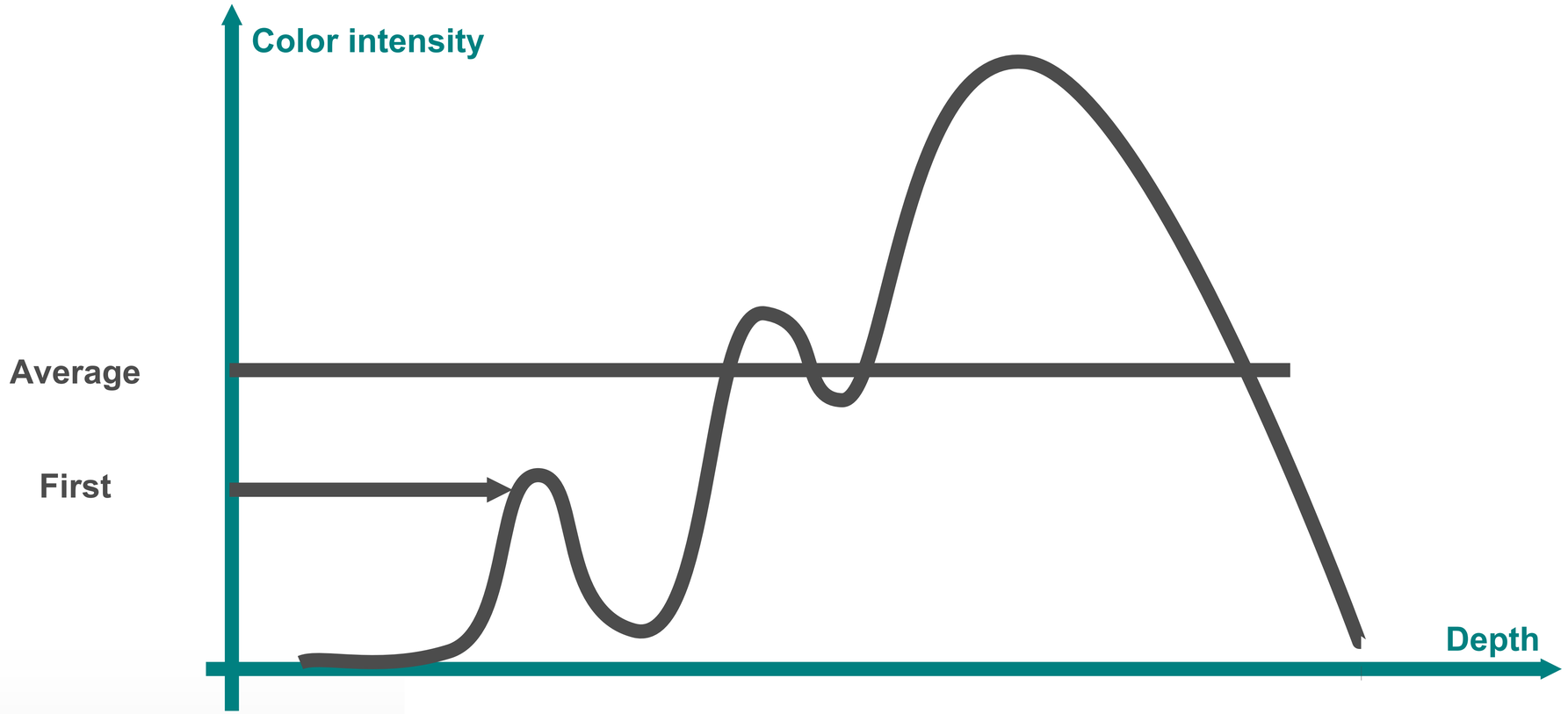
Compositing schemes

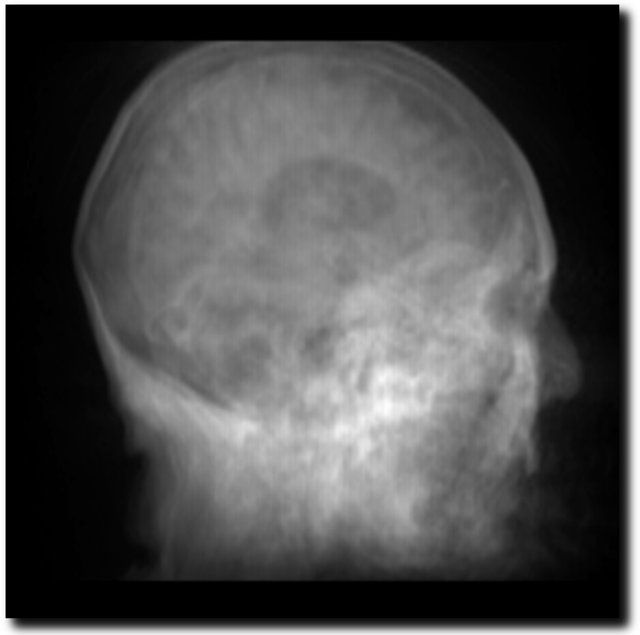
- Color intensity along the ray



Compositing schemes

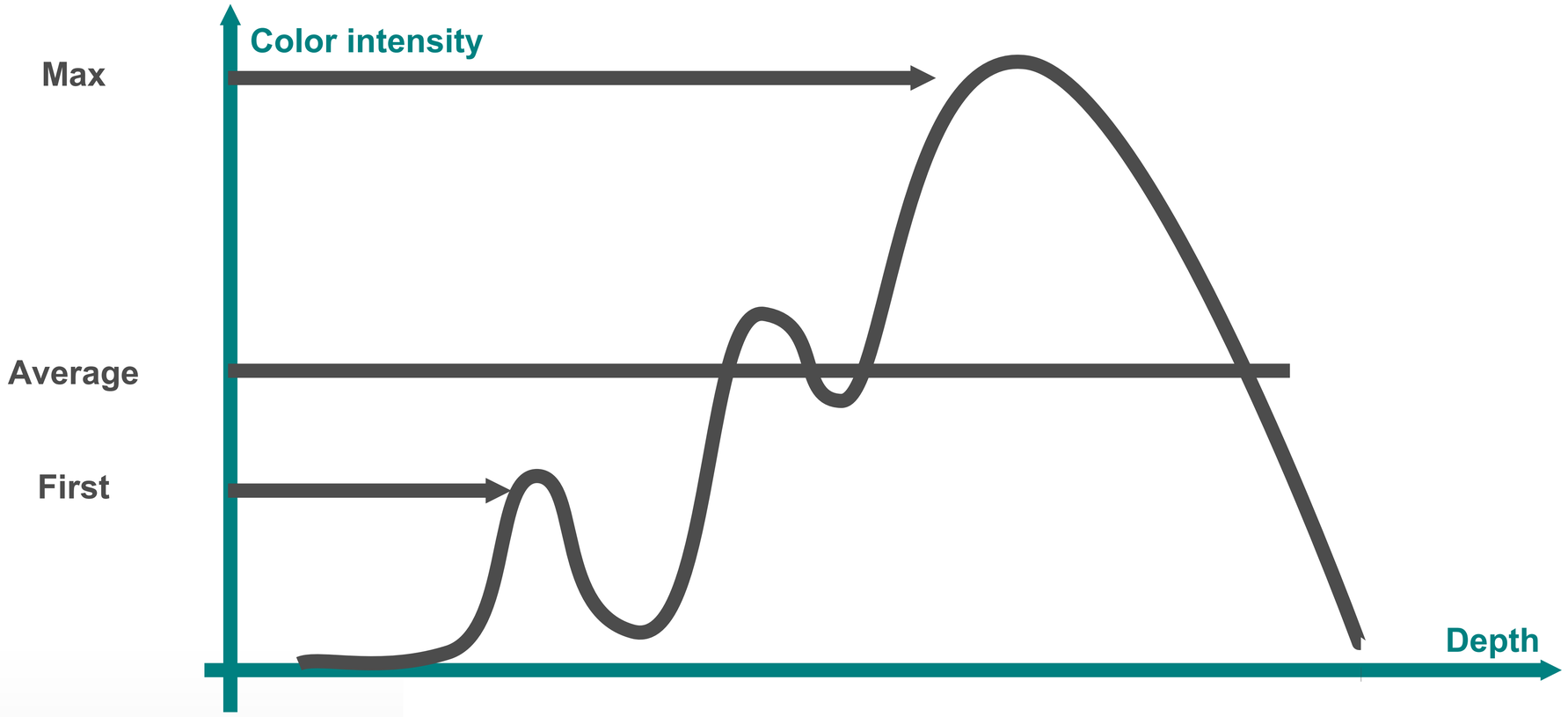
- Color intensity along the ray

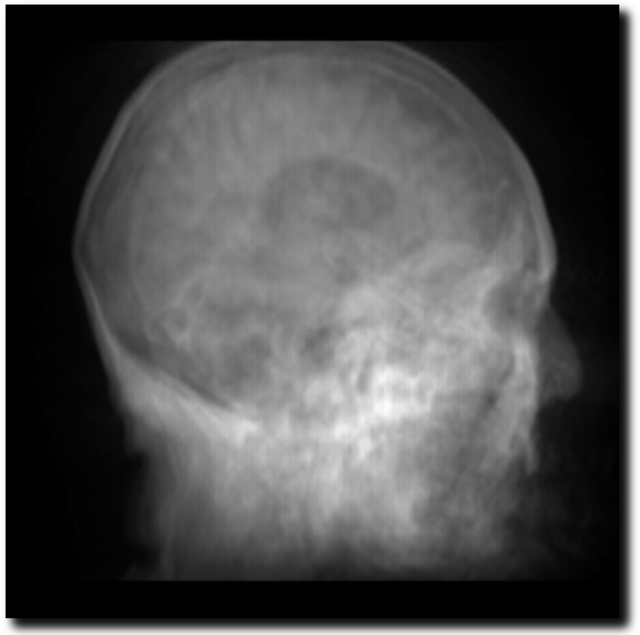




Compositing schemes

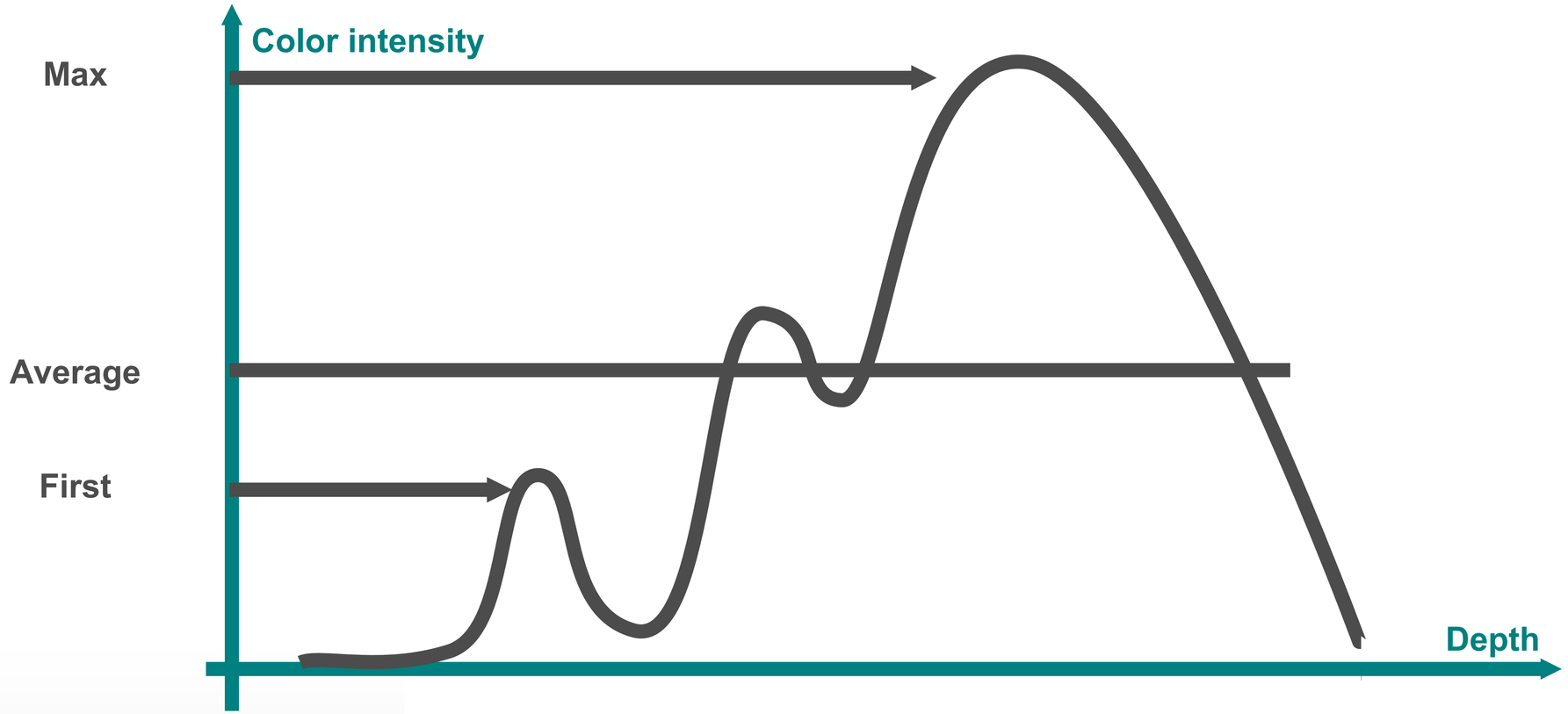
- Color intensity along the ray

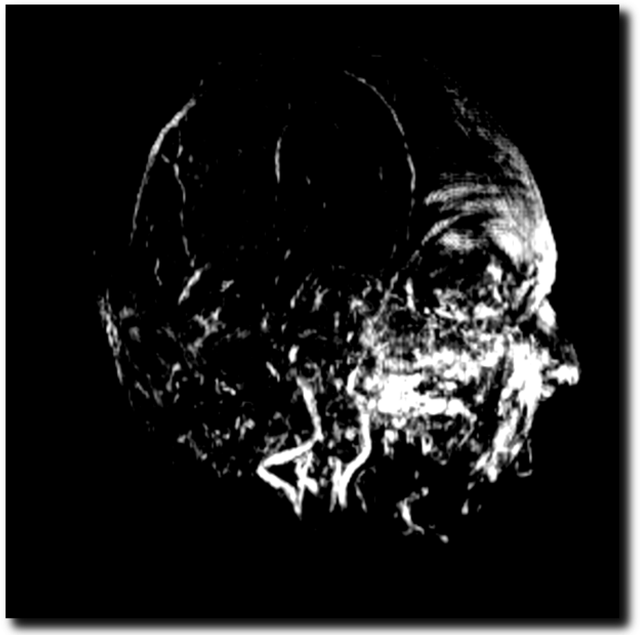




Compositing schemes

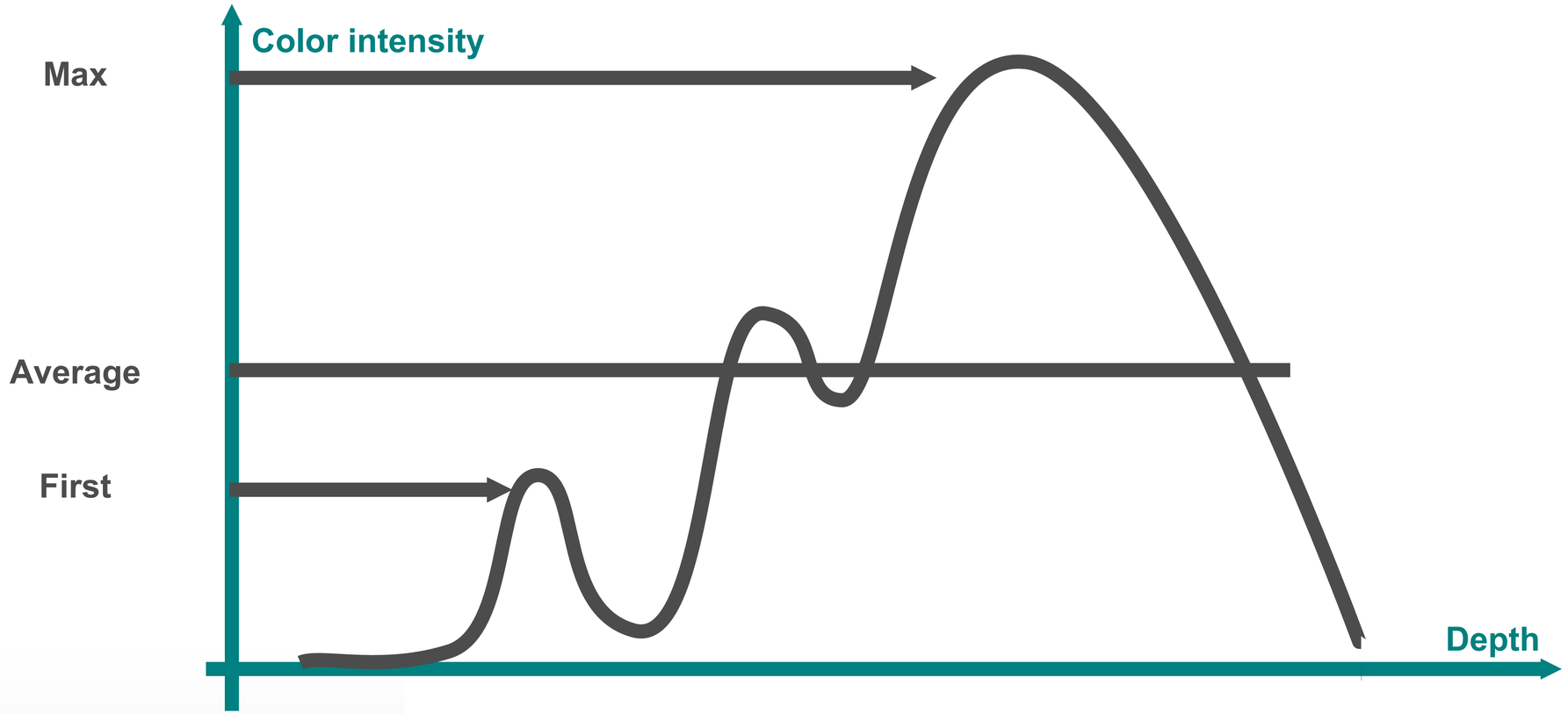
- Color intensity along the ray

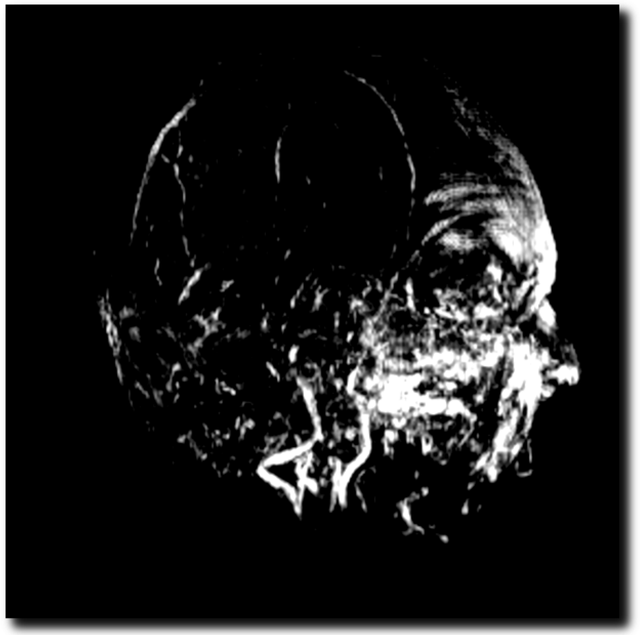




Compositing schemes

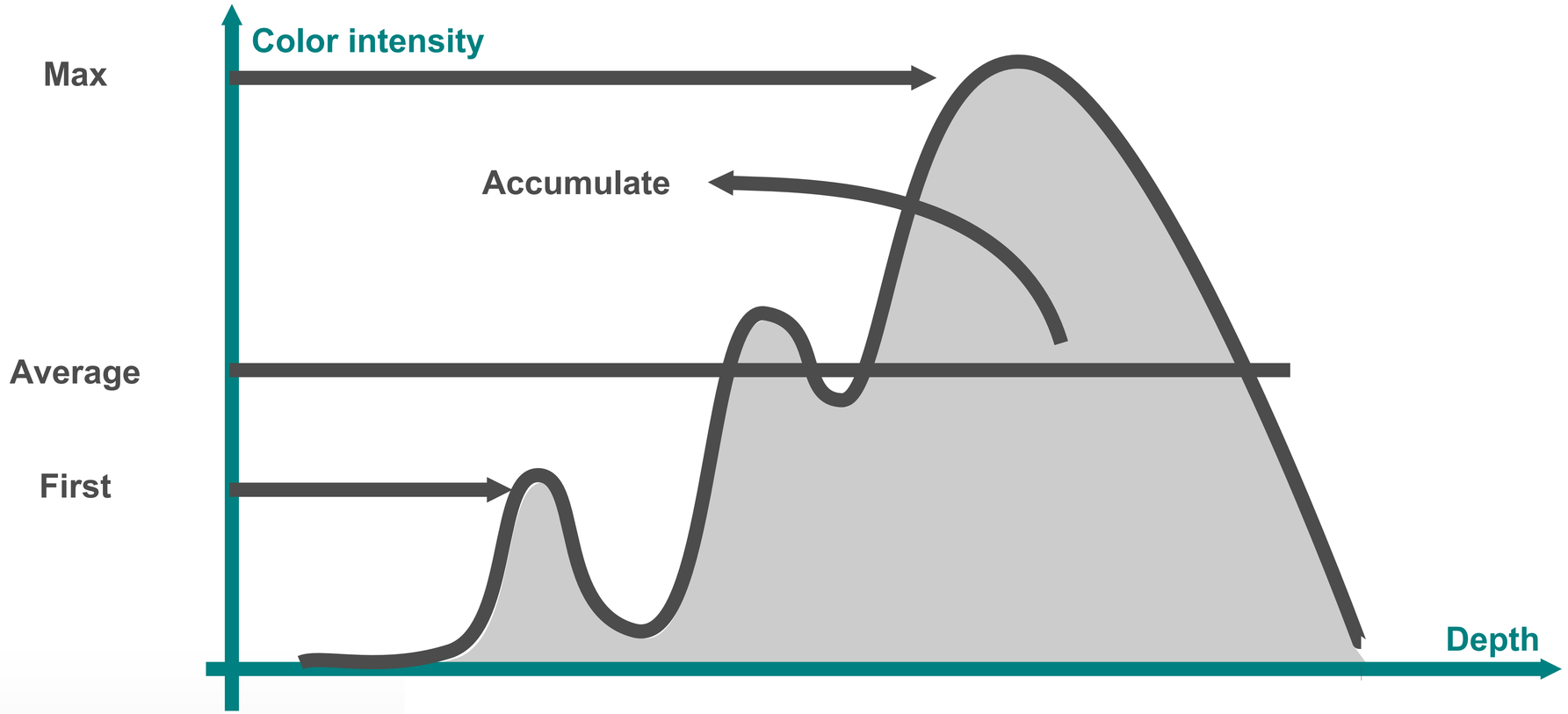
- Color intensity along the ray

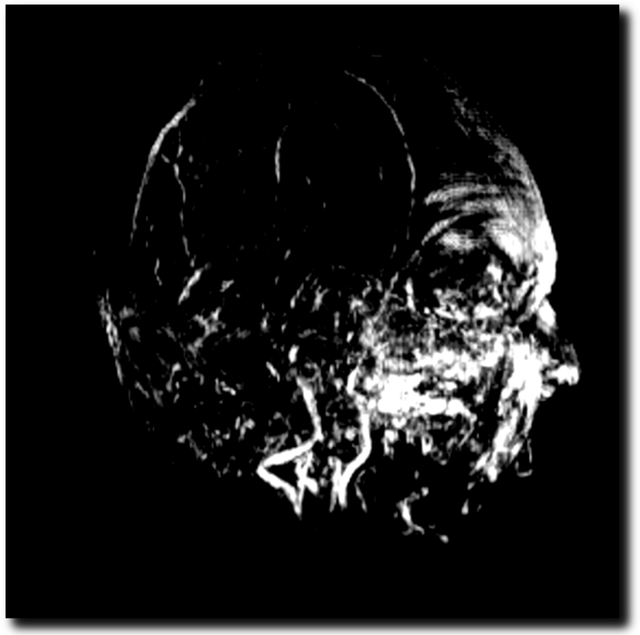




Compositing schemes

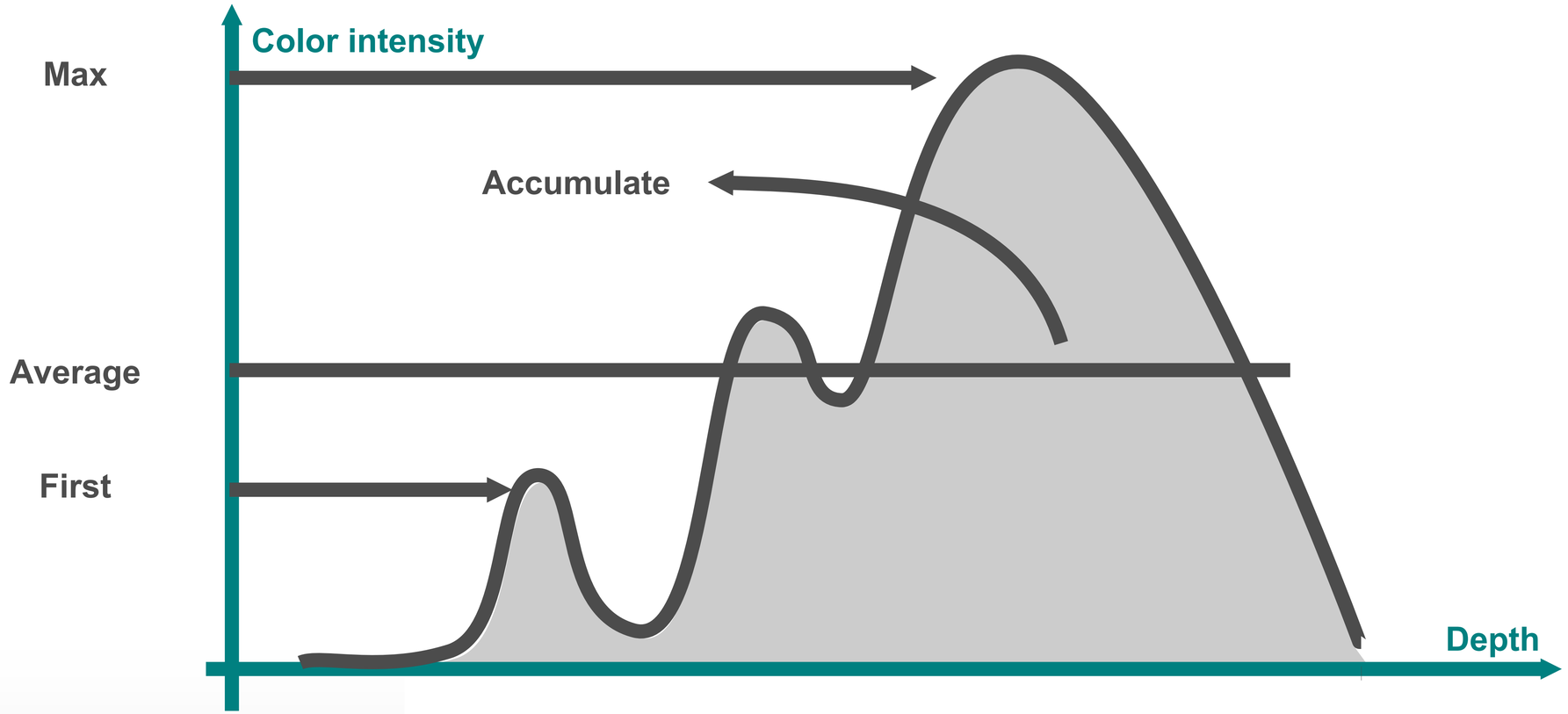
- Color intensity along the ray

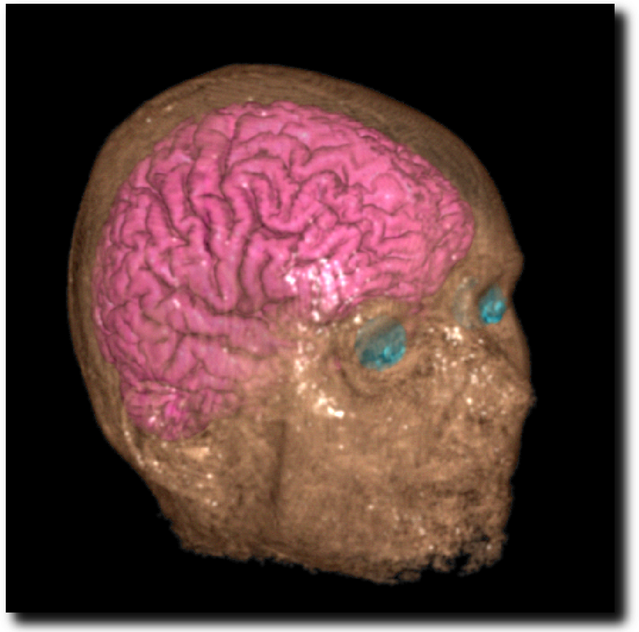




Compositing schemes

- Color intensity along the ray





Accumulation compositing

- From back to front

Accumulation compositing

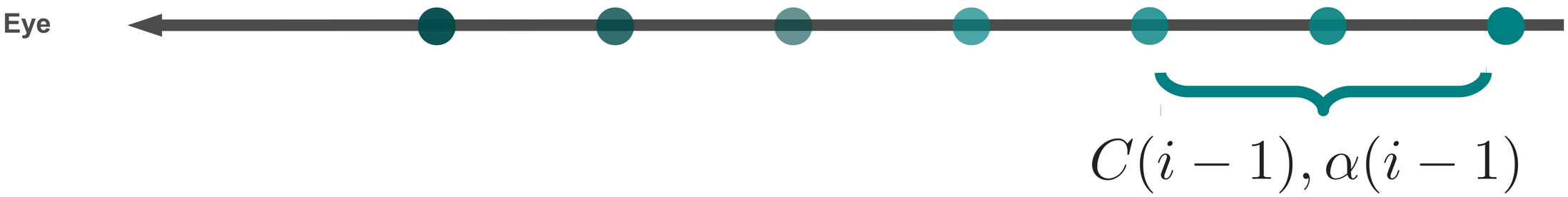
- From back to front

Accumulation compositing

- From back to front

Accumulation compositing

- From back to front

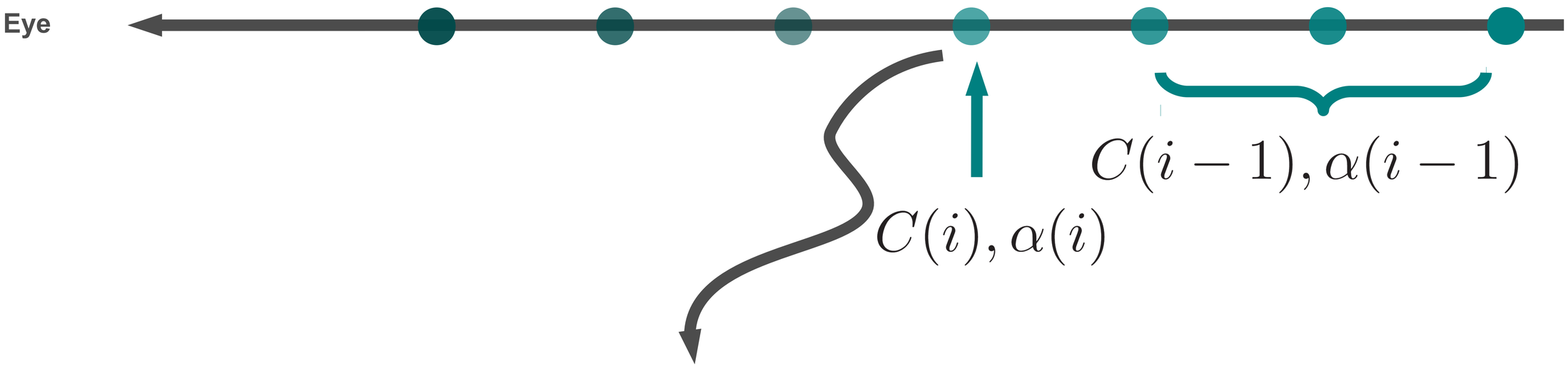


Accumulation compositing

- From back to front

Accumulation compositing

- From back to front



$$C'(i) = \alpha(i) \times C(i) + (1 - \alpha(i)) \times \alpha(i-1) \times C(i-1)$$
$$\alpha'(i) = \alpha(i) + (1 - \alpha(i)) \times \alpha(i-1)$$

VTK

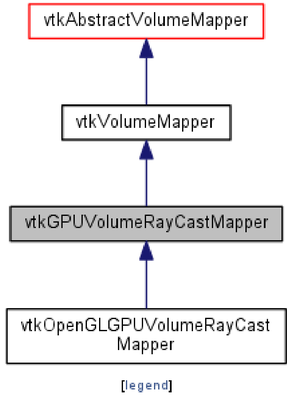
vtkGPUVolumeRayCastMapper Class Reference

abstract

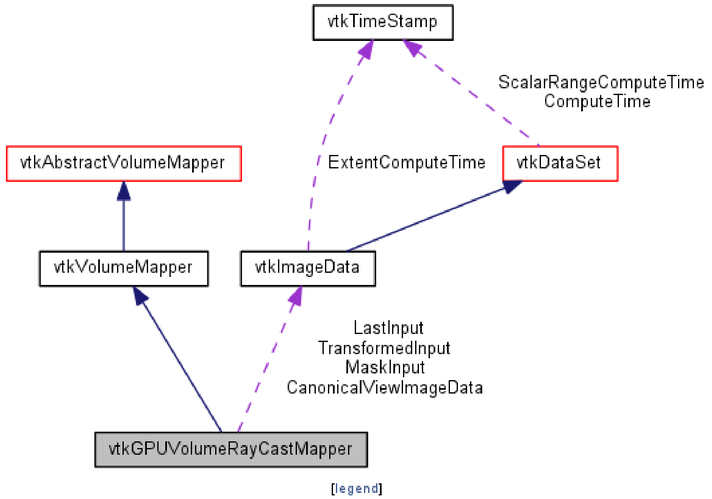
Ray casting performed on the GPU. More...

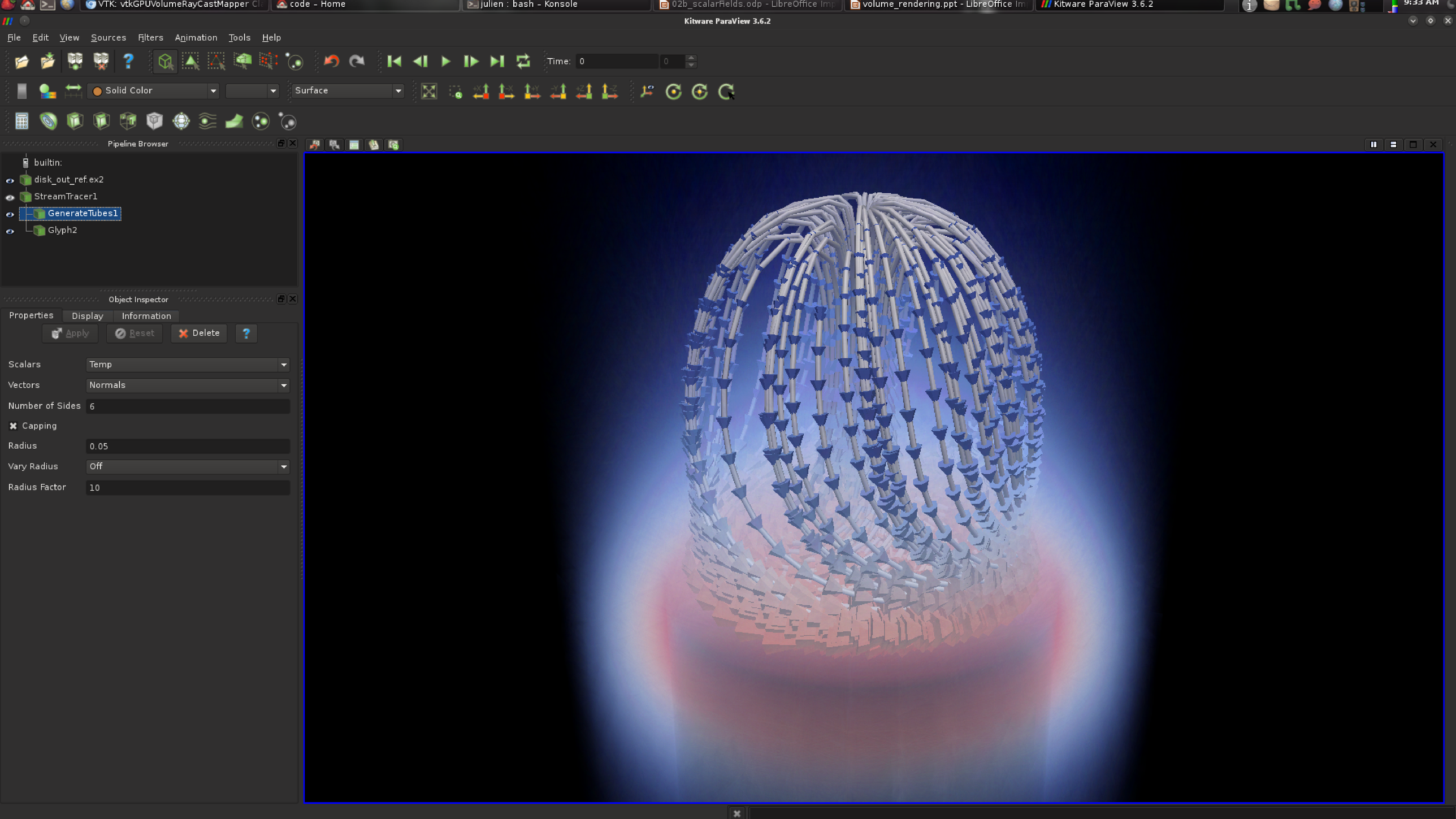
#include <vtkGPUVolumeRayCastMapper.h>

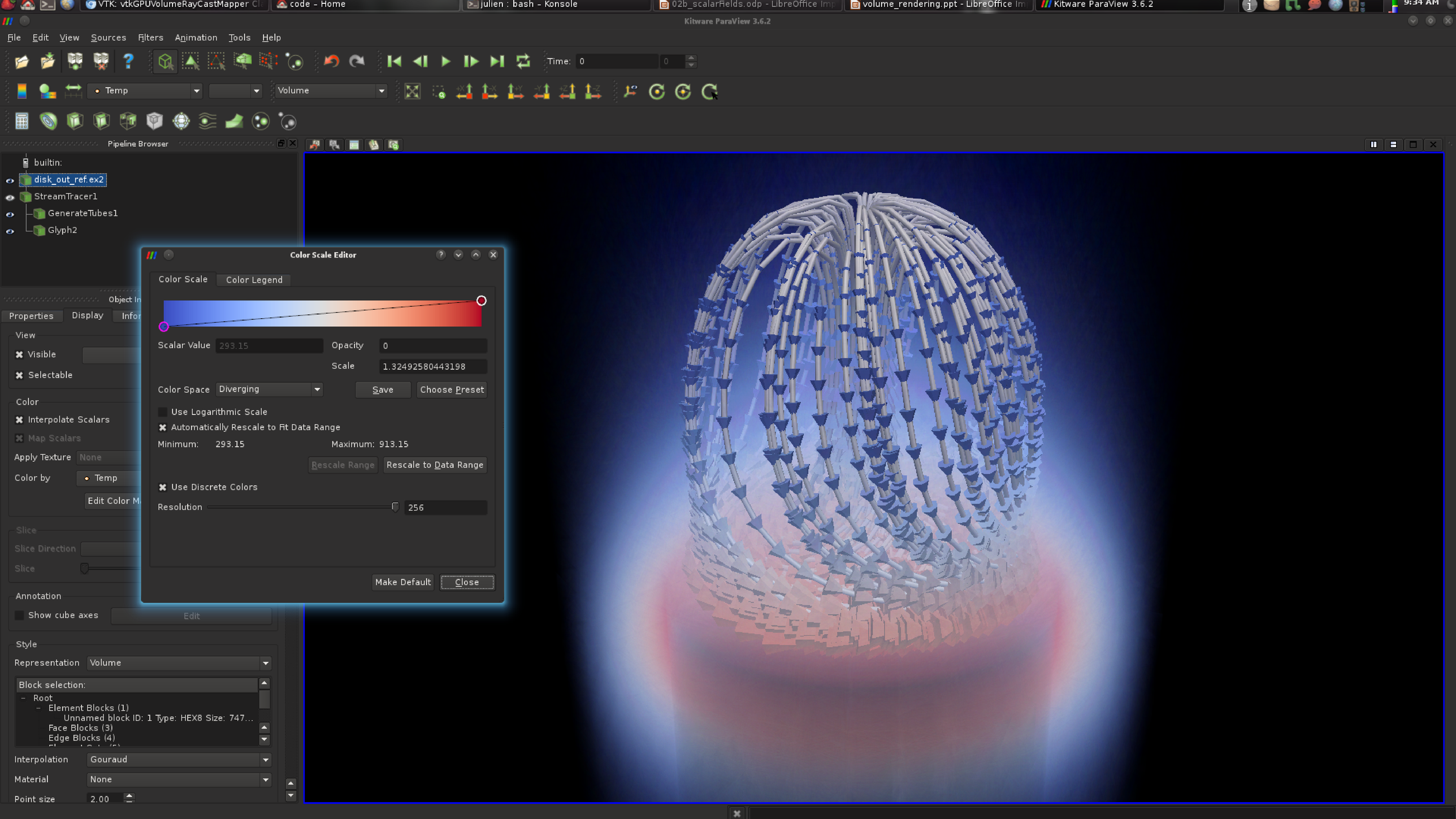
Inheritance diagram for vtkGPUVolumeRayCastMapper:

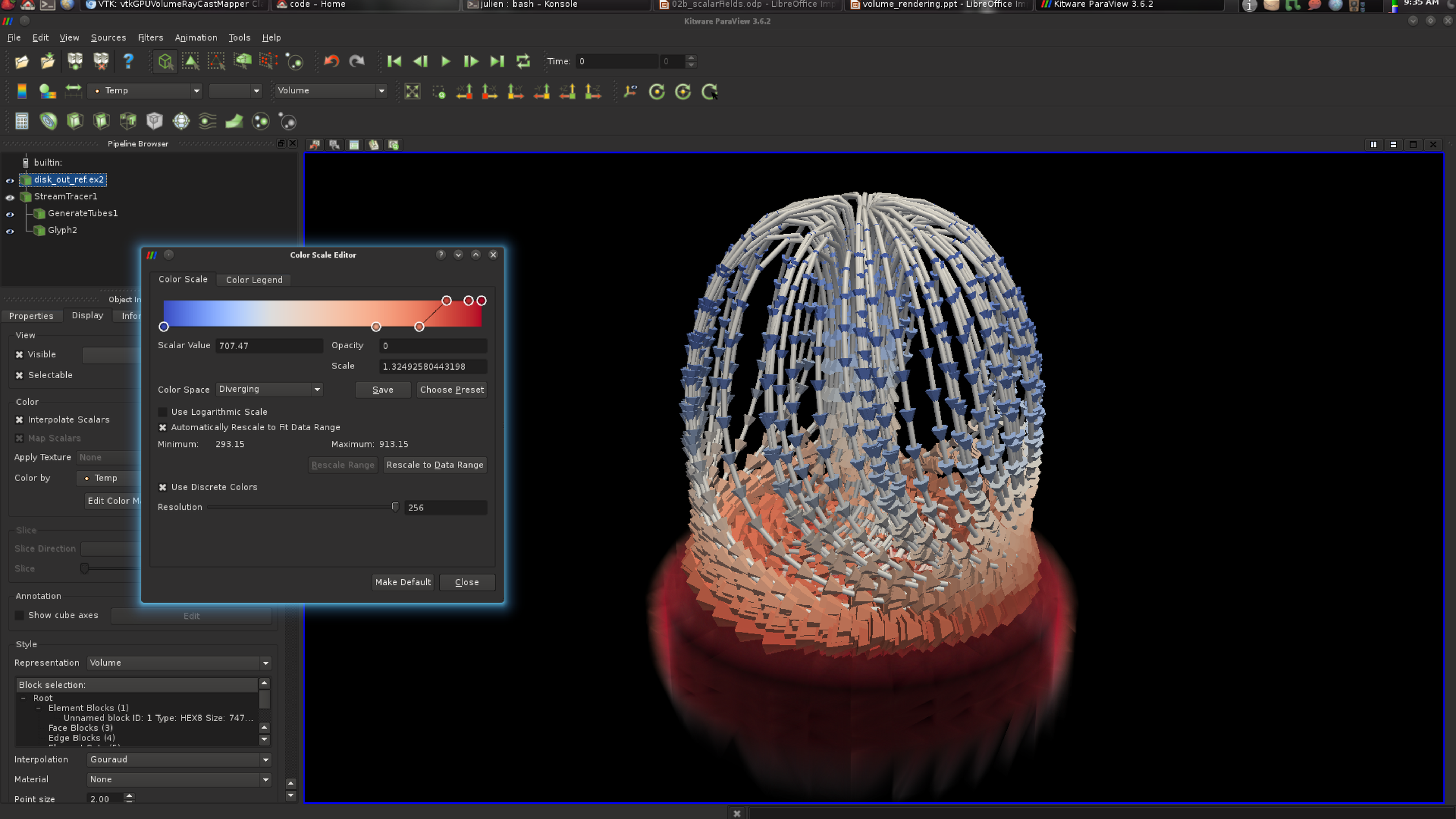


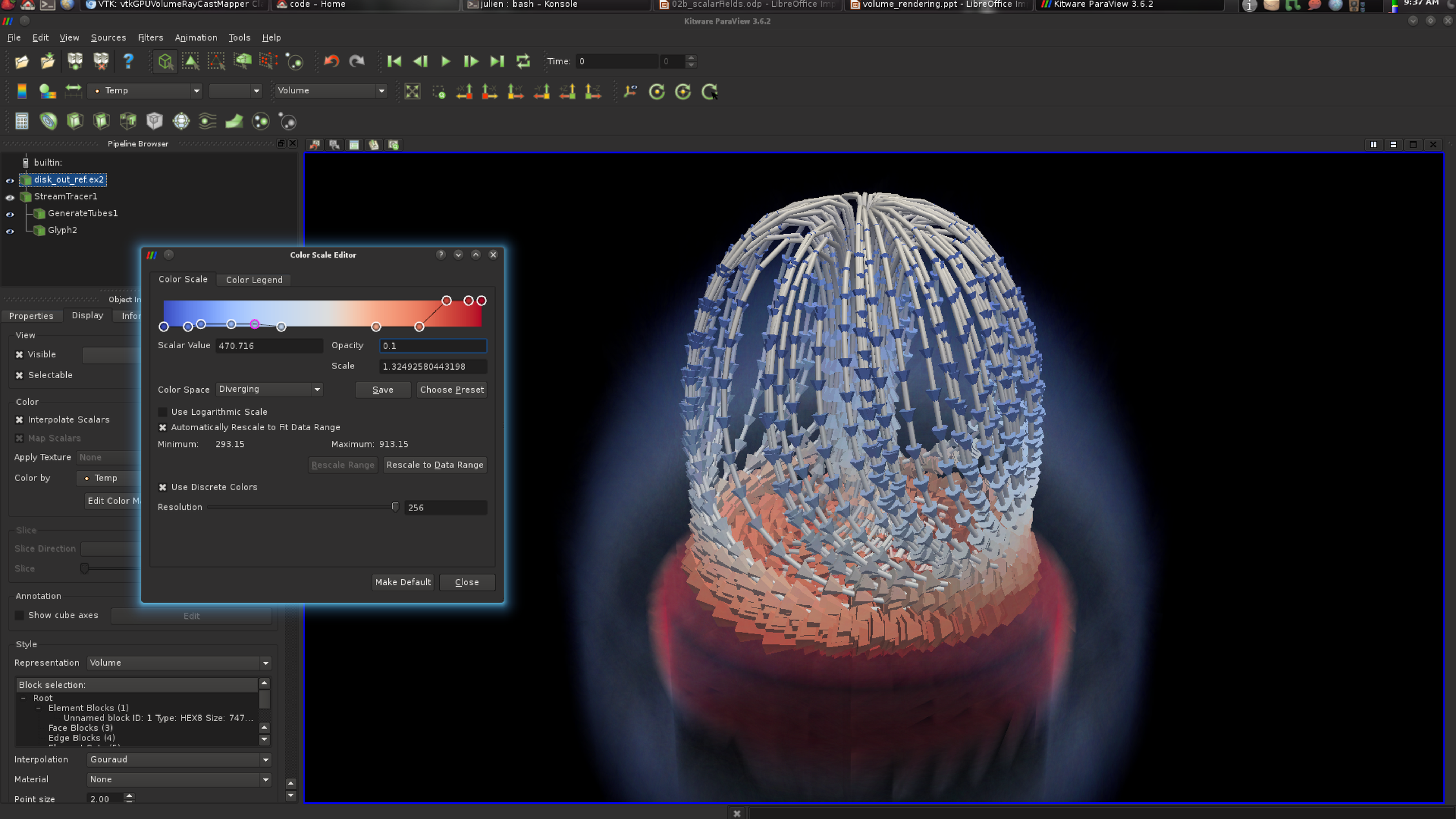
Collaboration diagram for vtkGPUVolumeRayCastMapper:

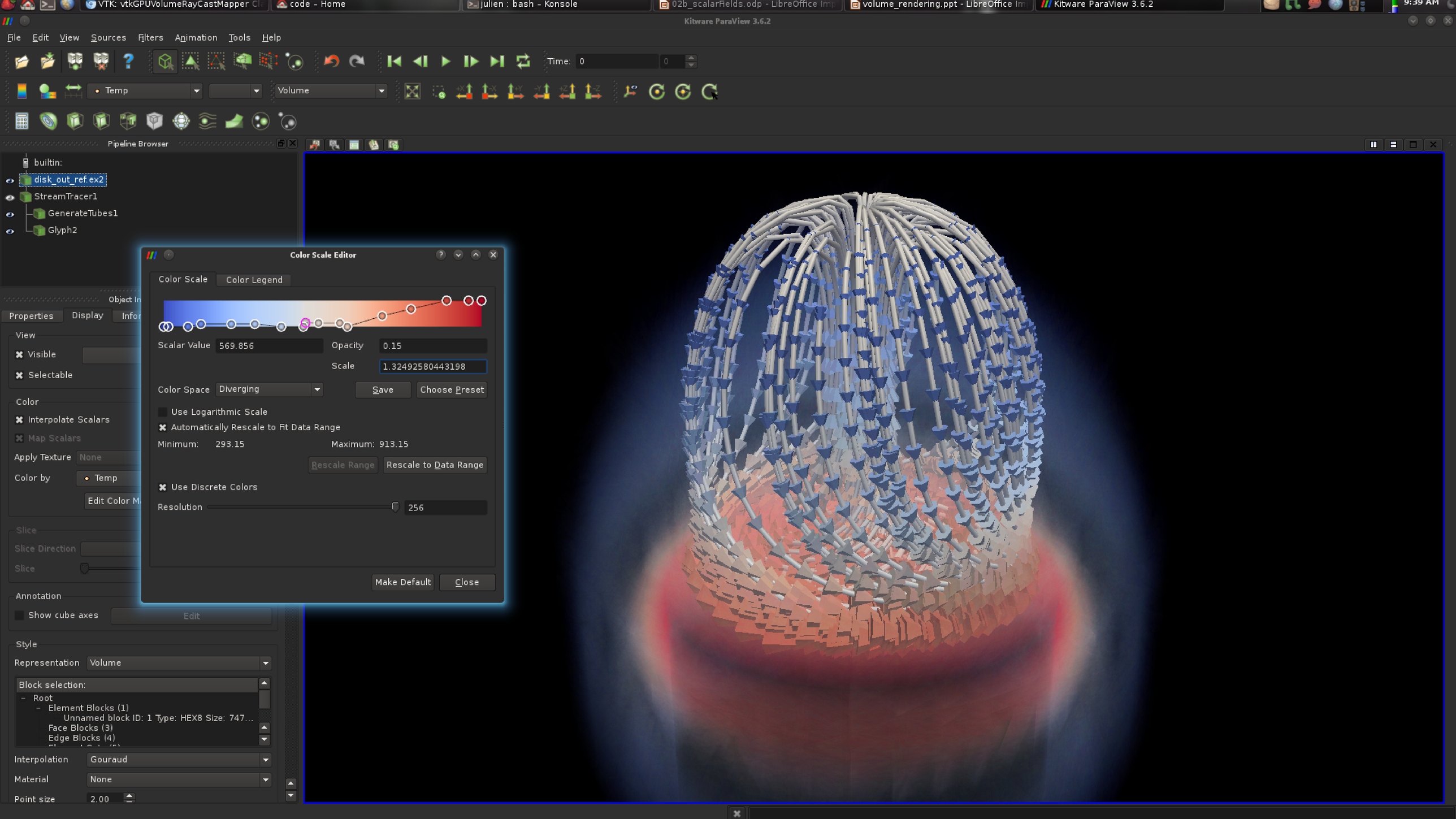



Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Solid Color
Surface
Pipeline Browser
builtin:
disk_out_ref.ex2
StreamTracer1
GenerateTubes1
Glyph2
Object Inspector
Properties Display Information
Apply Reset Delete
Scalars Temp
Vectors Normals
Number of Sides 6
Capping
Radius 0.05
Vary Radius Off
Radius Factor 10


VTK: vtkGPUVolumeRayCastMapper
code - Home
julien : bash - Konsole
02b_scalarFields.odp - LibreOffice Imp
volume_rendering.ppt - LibreOffice Im
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Temp
Volume
Pipeline Browser
builtin:
disk_out_ref.ex2
StreamTracer1
GenerateTubes1
Glyph2
Color Scale Editor
Color Scale
Color Legend
Scalar Value 293.15
Opacity 0
Scale 1.32492580443198
Color Space Diverging
Save
Choose Preset
Use Logarithmic Scale
Automatically Rescale to Fit Data Range
Minimum: 293.15
Maximum: 913.15
Rescale Range
Rescale to Data Range
Use Discrete Colors
Resolution 256
Make Default
Close
Properties
Display
View
Visible
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Temp
Edit Color M
Slice
Slice Direction
Slice
Annotation
Show cube axes
Edit
Style
Representation Volume
Block selection:
Root
Element Blocks (1)
Unnamed block ID: 1 Type: HEX8 Size: 747...
Face Blocks (3)
Edge Blocks (4)
Interpolation Gouraud
Material None
Point size 2.00

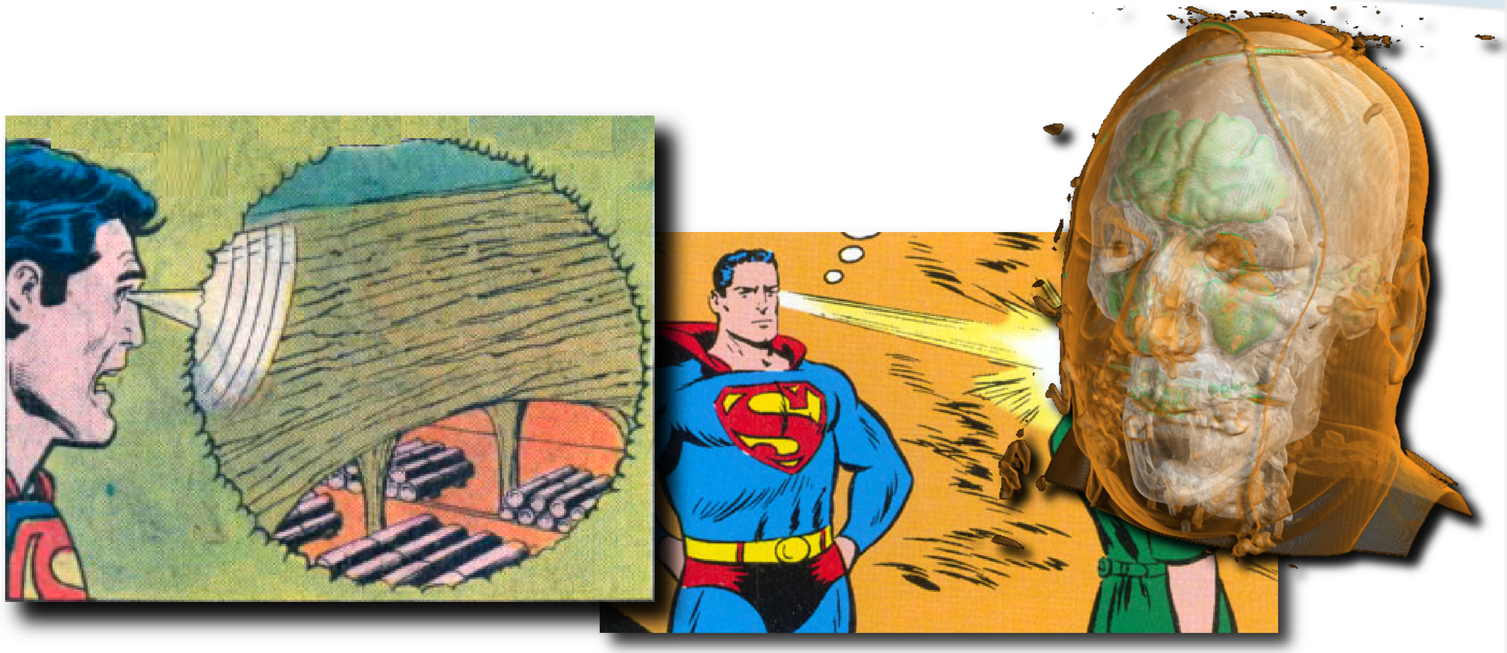

VTK: vtkGPUVolumeRayCastMapper
code - Home
julien : bash - Konsole
02b_scalarfields.odp - LibreOffice Imp
volume_rendering.ppt - LibreOffice Im
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Temp
Volume
Pipeline Browser
builtin:
disk_out_ref.ex2
StreamTracer1
GenerateTubes1
Glyph2
Color Scale Editor
Color Scale
Color Legend
Scalar Value 707.47
Opacity 0
Scale 1.32492580443198
Color Space Diverging
Save
Choose Preset
Use Logarithmic Scale
Automatically Rescale to Fit Data Range
Minimum: 293.15
Maximum: 913.15
Rescale Range
Rescale to Data Range
Use Discrete Colors
Resolution 256
Make Default
Close
Properties
Display
View
Visible
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Temp
Edit Color M
Slice
Slice Direction
Slice
Annotation
Show cube axes
Edit
Style
Representation Volume
Block selection:
Root
Element Blocks (1)
Unnamed block ID: 1 Type: HEX8 Size: 747...
Face Blocks (3)
Edge Blocks (4)
Interpolation Gouraud
Material None
Point size 2.00


Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Temp
Volume
Pipeline Browser
builtin:
disk_out_ref.ex2
StreamTracer1
GenerateTubes1
Glyph2
Color Scale Editor
Color Scale
Color Legend
Scalar Value 470.716
Opacity 0.1
Scale 1.32492580443198
Color Space Diverging
Save
Choose Preset
Use Logarithmic Scale
Automatically Rescale to Fit Data Range
Minimum: 293.15
Maximum: 913.15
Rescale Range
Rescale to Data Range
Use Discrete Colors
Resolution 256
Make Default
Close
Properties
Display
View
Visible
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Temp
Slice
Slice Direction
Annotation
Show cube axes
Edit
Style
Representation Volume
Block selection:
Root
Element Blocks (1)
Unnamed block ID: 1 Type: HEX8 Size: 747...
Face Blocks (3)
Edge Blocks (4)
Interpolation Gouraud
Material None
Point size 2.00


Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Temp
Volume
Pipeline Browser
builtin:
disk_out_ref.ex2
StreamTracer1
GenerateTubes1
Glyph2
Color Scale Editor
Color Scale
Color Legend
Scalar Value 569.856
Opacity 0.15
Scale 1.32492580443198
Color Space Diverging
Save
Choose Preset
Use Logarithmic Scale
Automatically Rescale to Fit Data Range
Minimum: 293.15
Maximum: 913.15
Rescale Range
Rescale to Data Range
Use Discrete Colors
Resolution 256
Make Default
Close
Properties
Display
View
Visible
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Temp
Edit Color M
Slice
Slice Direction
Slice
Annotation
Show cube axes
Edit
Style
Representation Volume
Block selection:
Root
Element Blocks (1)
Unnamed block ID: 1 Type: HEX8 Size: 747...
Face Blocks (3)
Edge Blocks (4)
Interpolation Gouraud
Material None
Point size 2.00

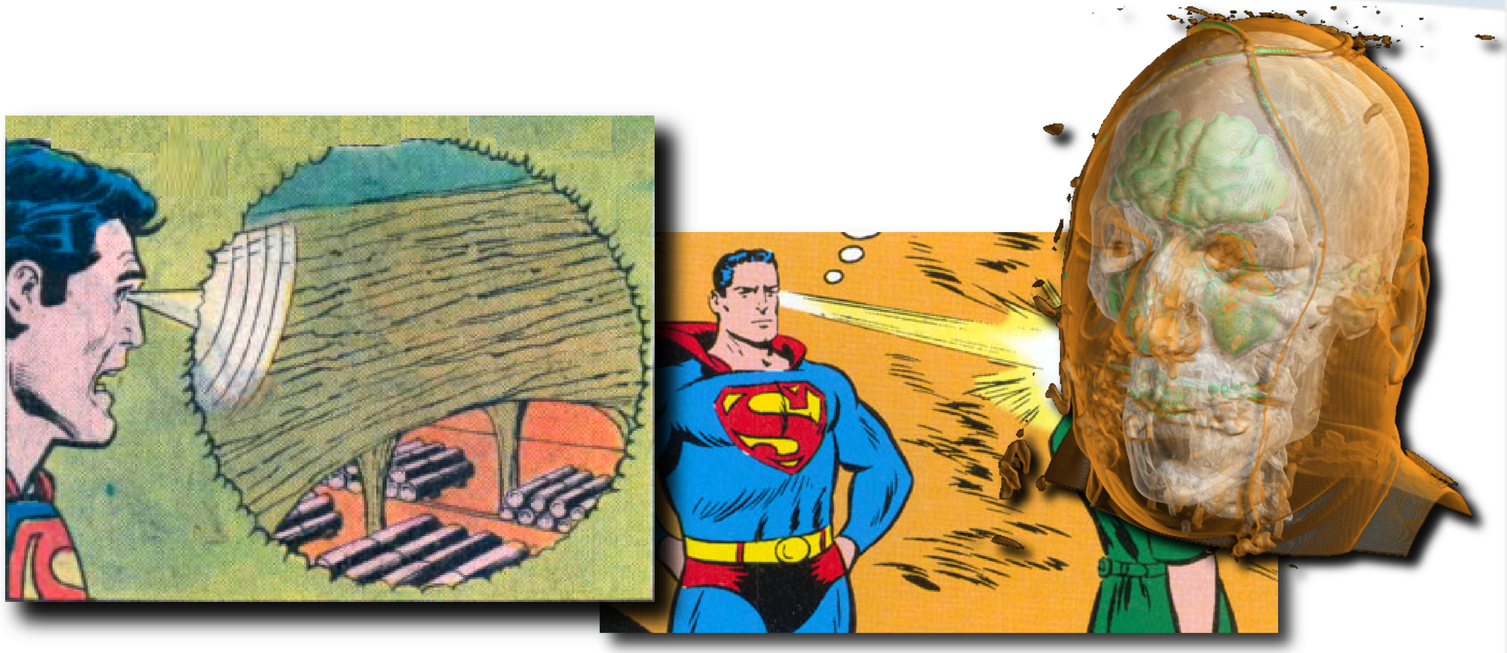
Pros and Cons

- Simple algorithm
- Inherently parallel
- High quality renderings



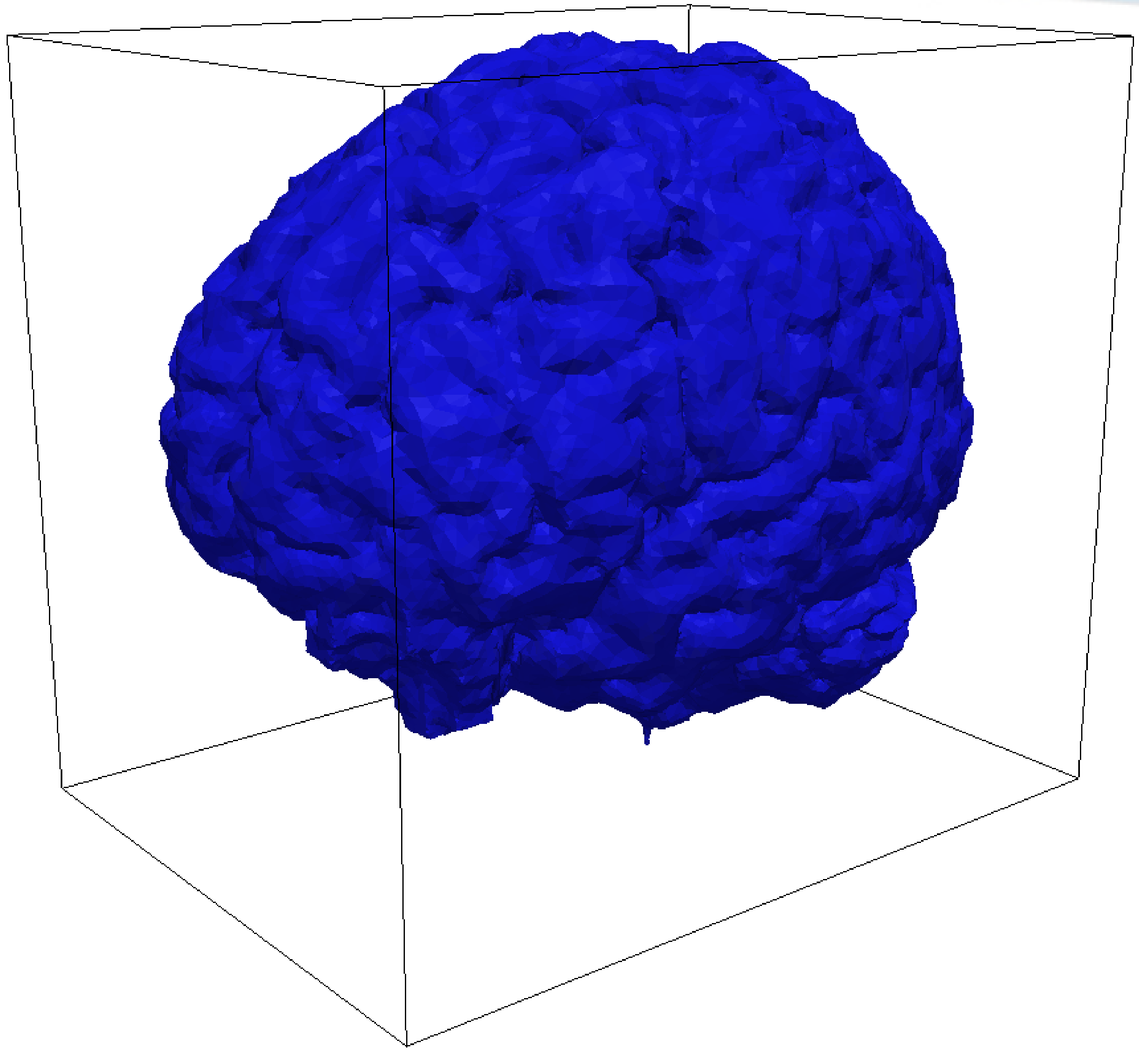
Pros and Cons

- Simple algorithm
- Inherently parallel
- High quality renderings



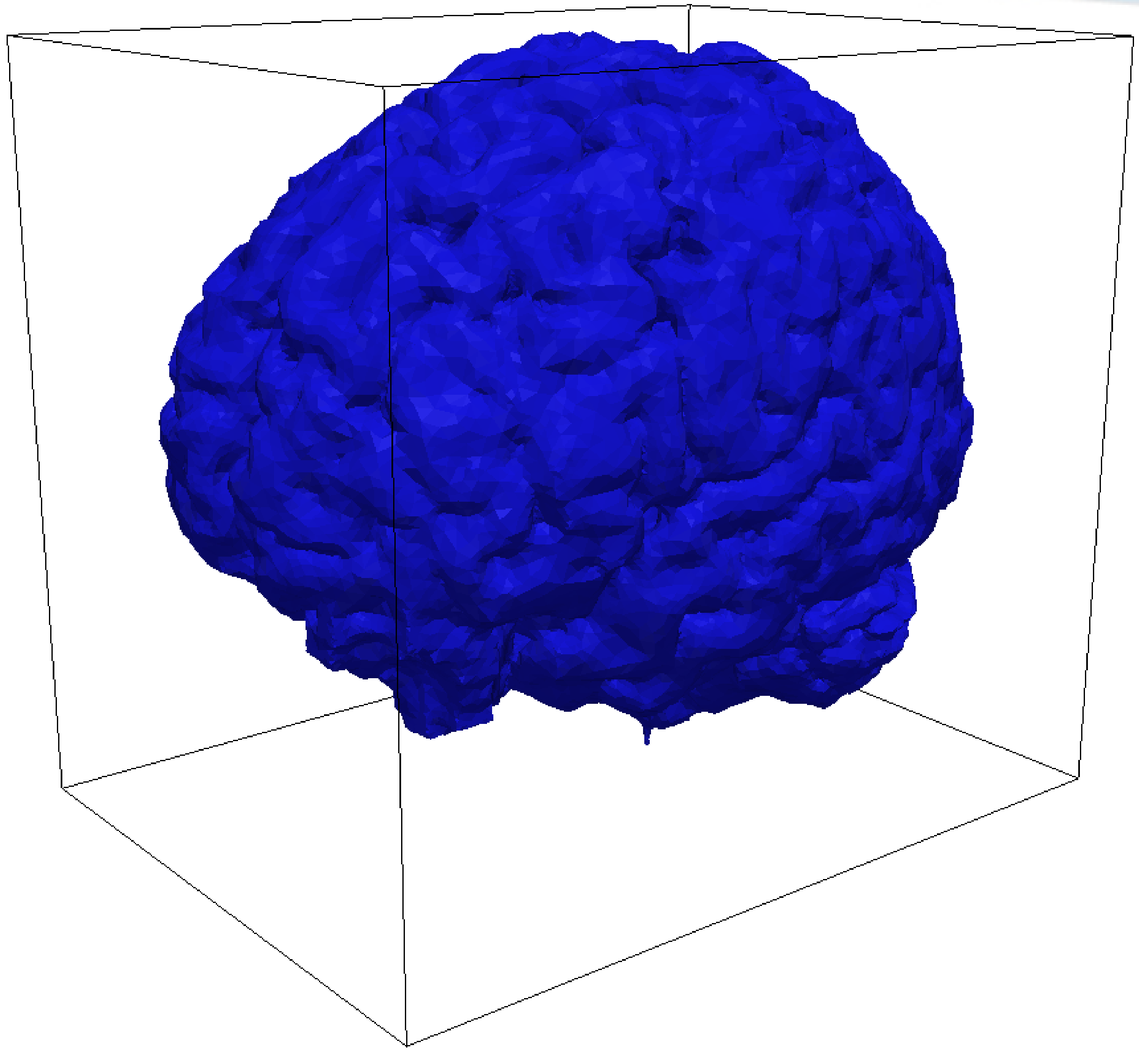
- SLOW!
  - Lots of rays
  - Lots of samples
- Transfer function design may not be intuitive

Level set extraction



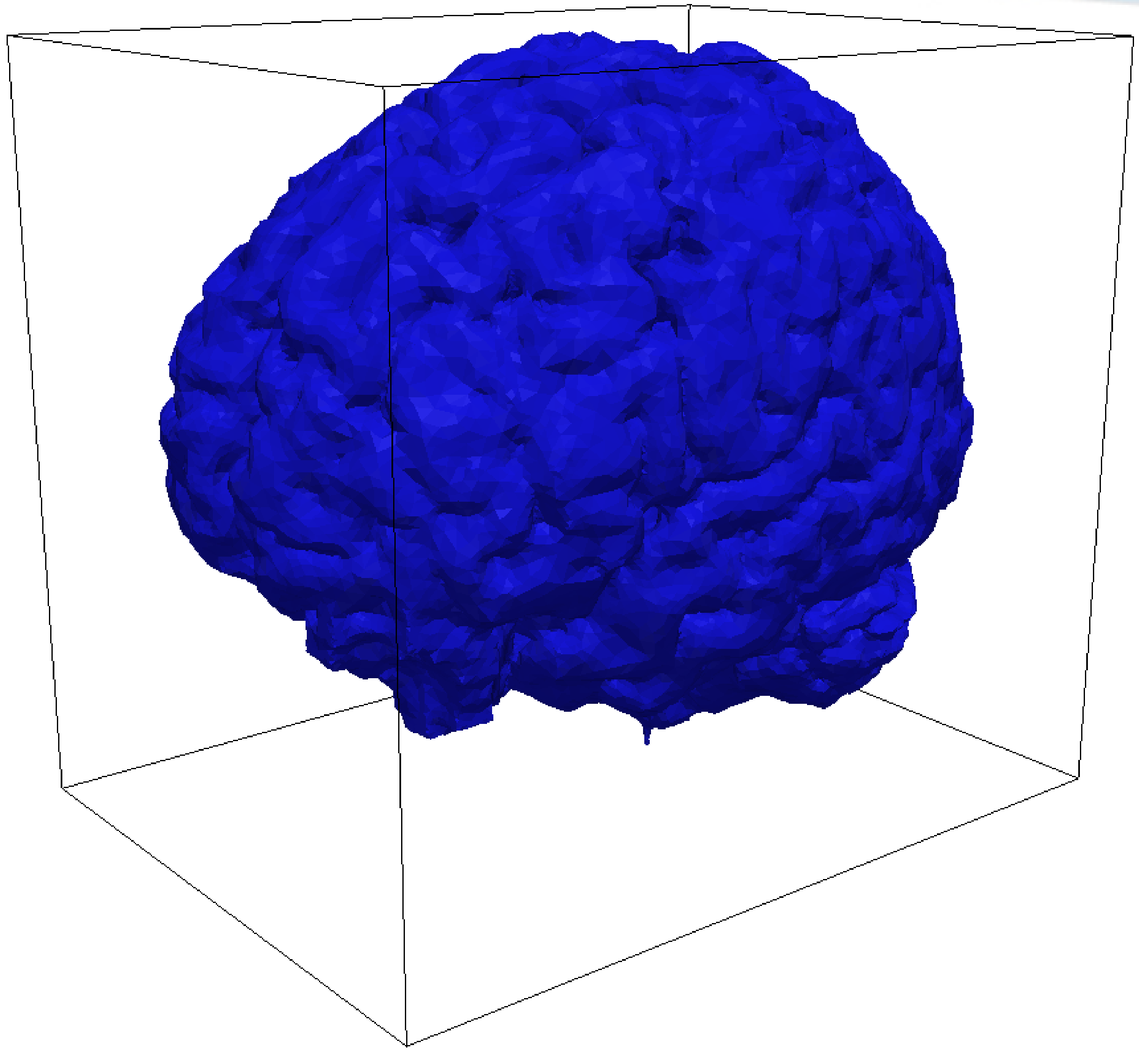
Level set extraction

- Exact geometry visualization



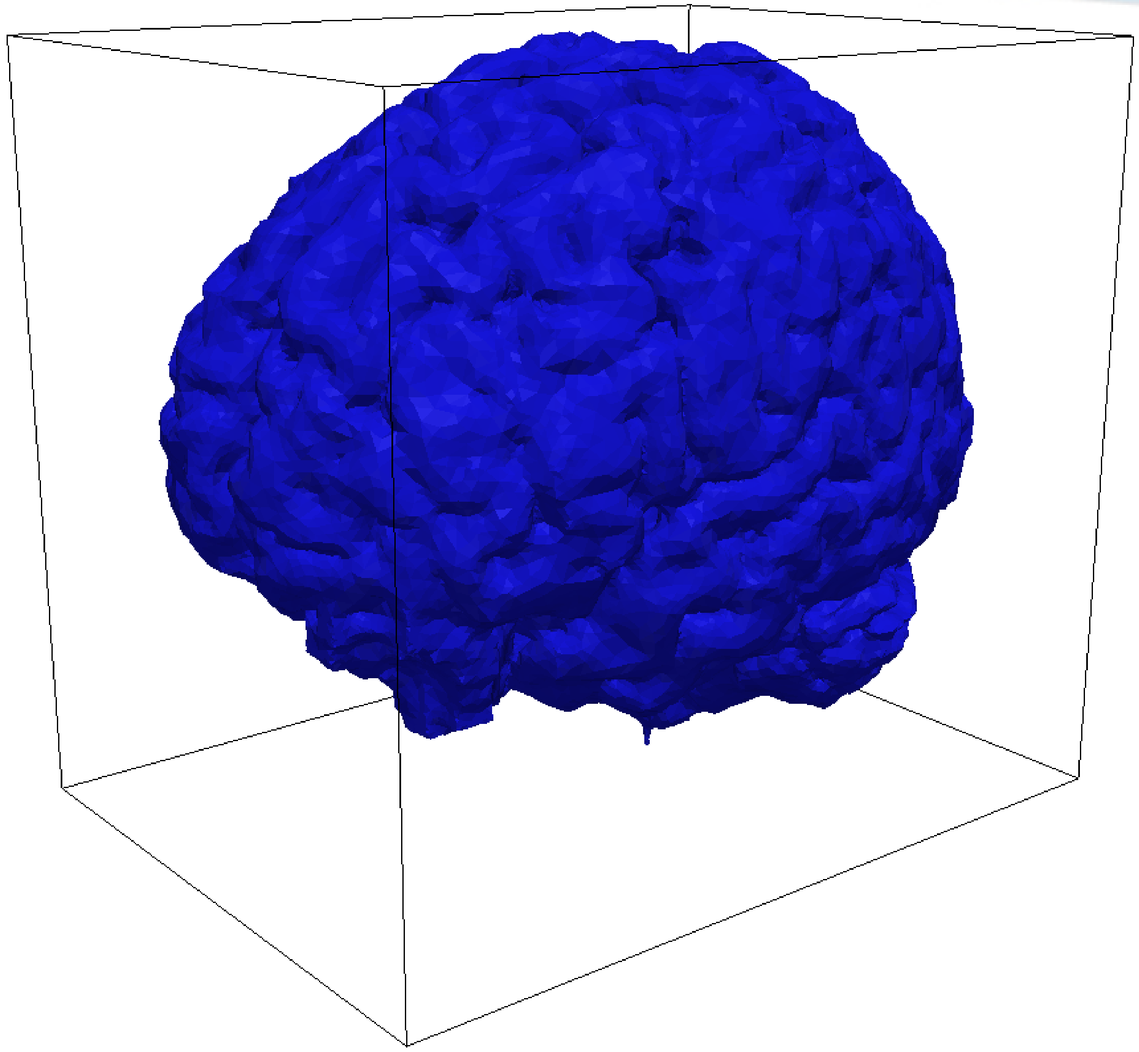
Level set extraction

- Exact geometry visualization
- Regions of interests



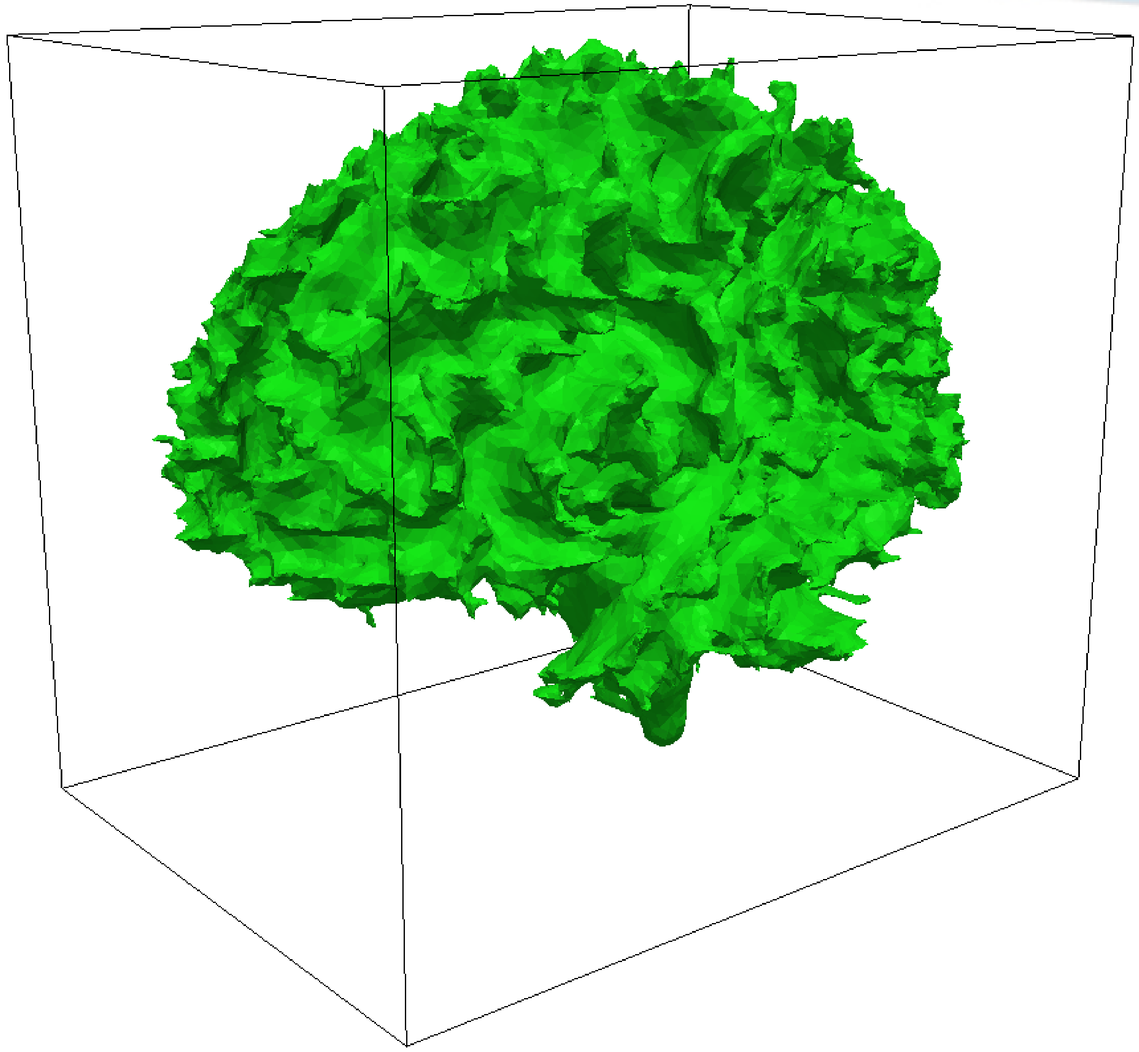
Level set extraction

- Exact geometry visualization
- Regions of interests
  - Identify
  - Study
  - Visualize



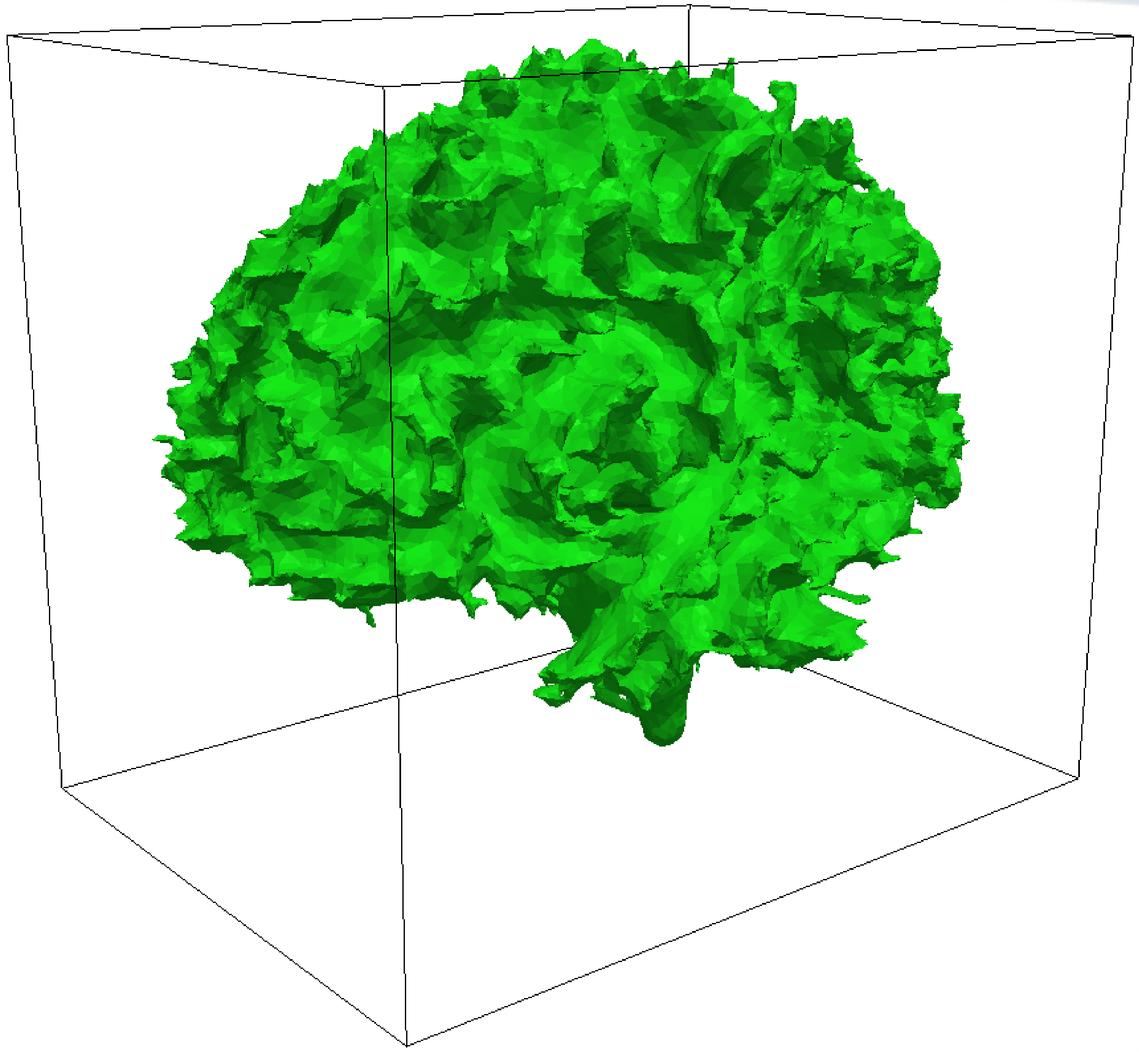
Level set extraction

- Exact geometry visualization
- Regions of interests
  - Identify
  - Study
  - Visualize
    - Different layers



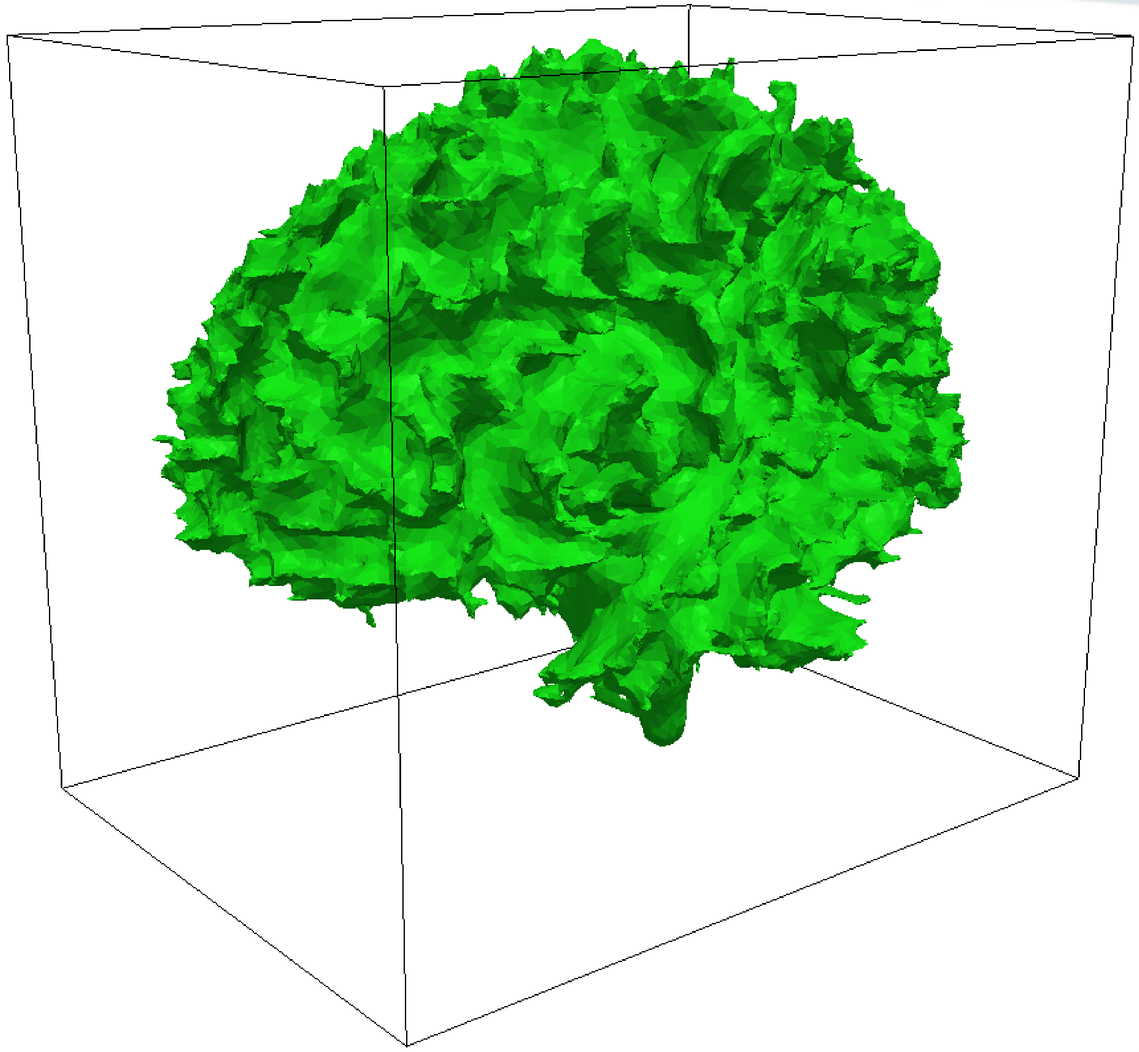
Level set extraction

- Exact geometry visualization
- Regions of interests
  - Identify
  - Study
  - Visualize
    - Different layers
    - Different level sets



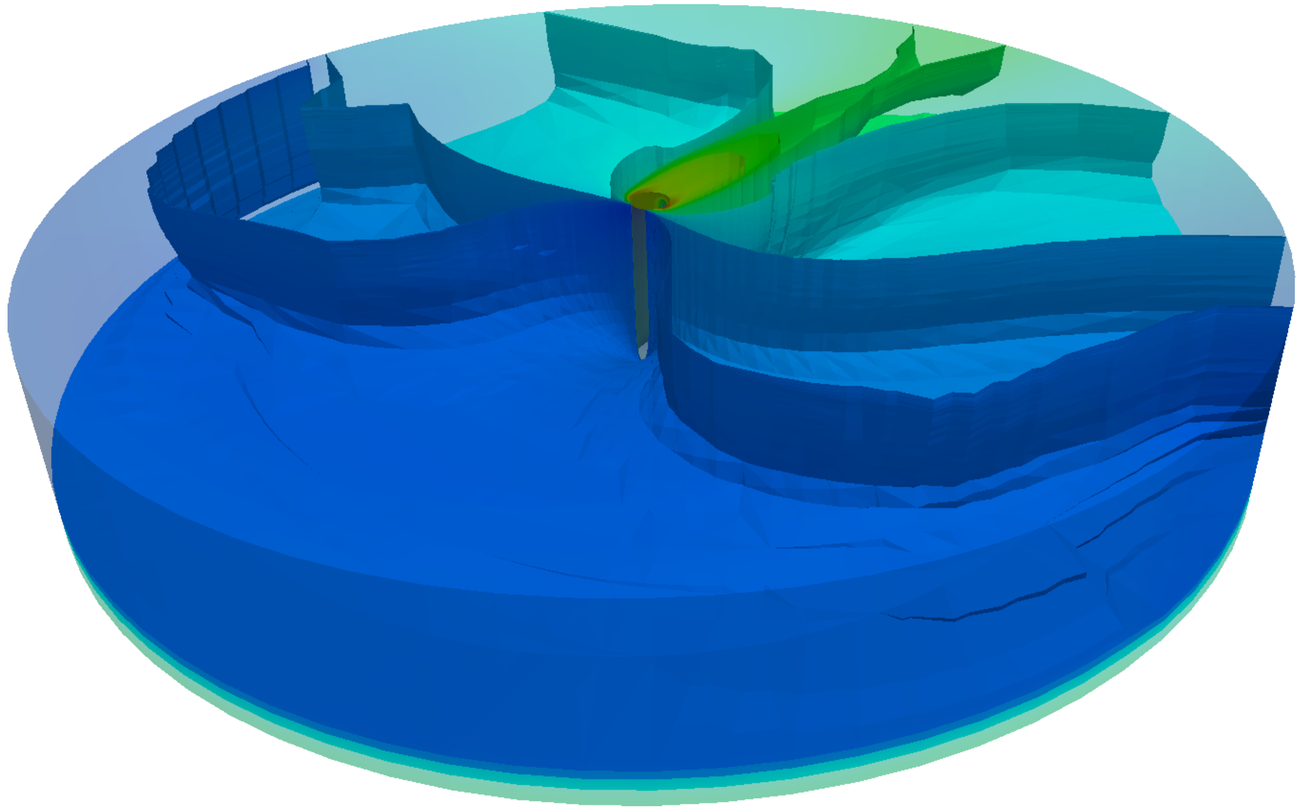
Level set extraction

- Exact geometry visualization
- Regions of interests
  - Identify
  - Study
  - Visualize
    - Different layers
    - Different level sets
- Towards data analysis

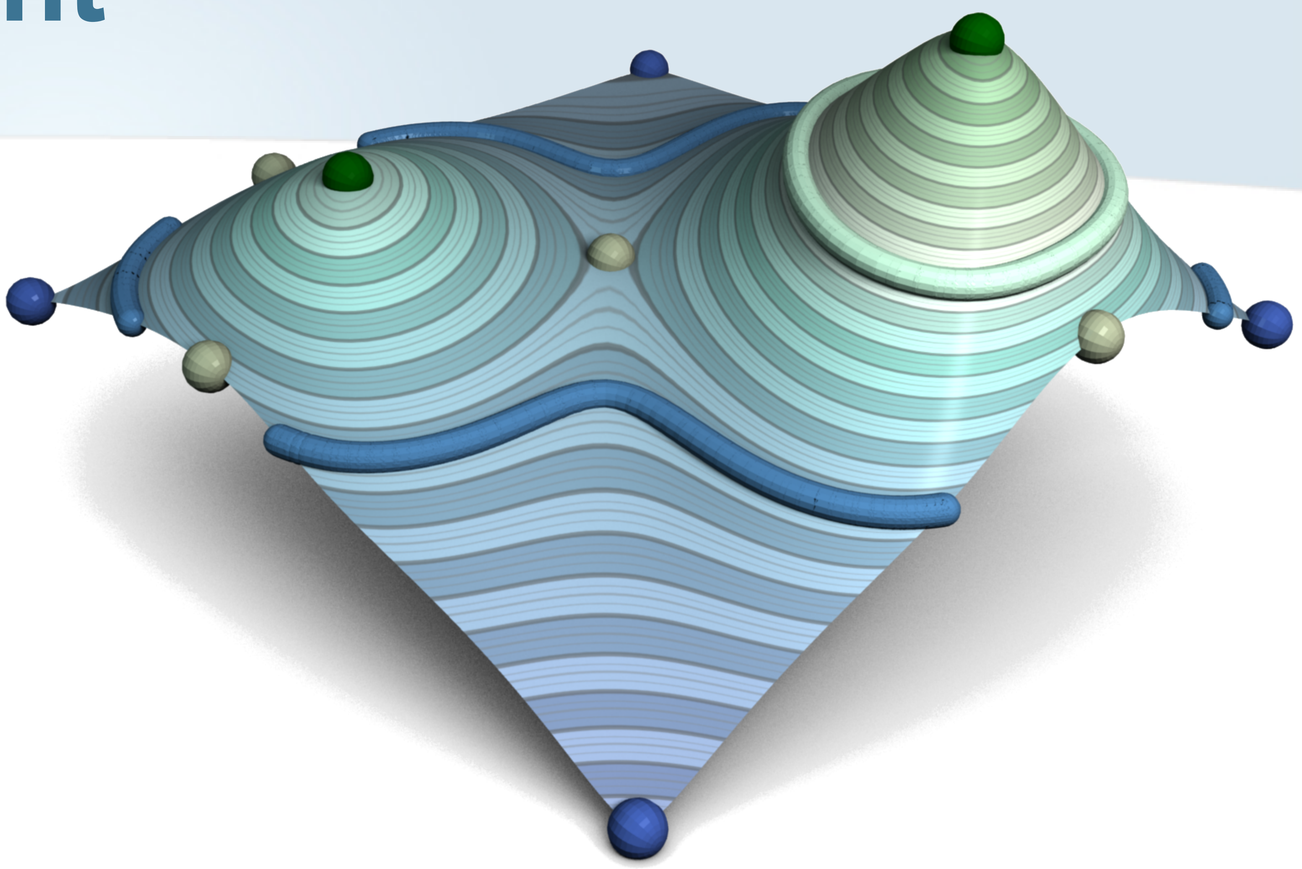


Level set summary

- Level set extraction
  - PL-manifold domains
  - Regular grids
- Acceleration structures
- Topological cleaning

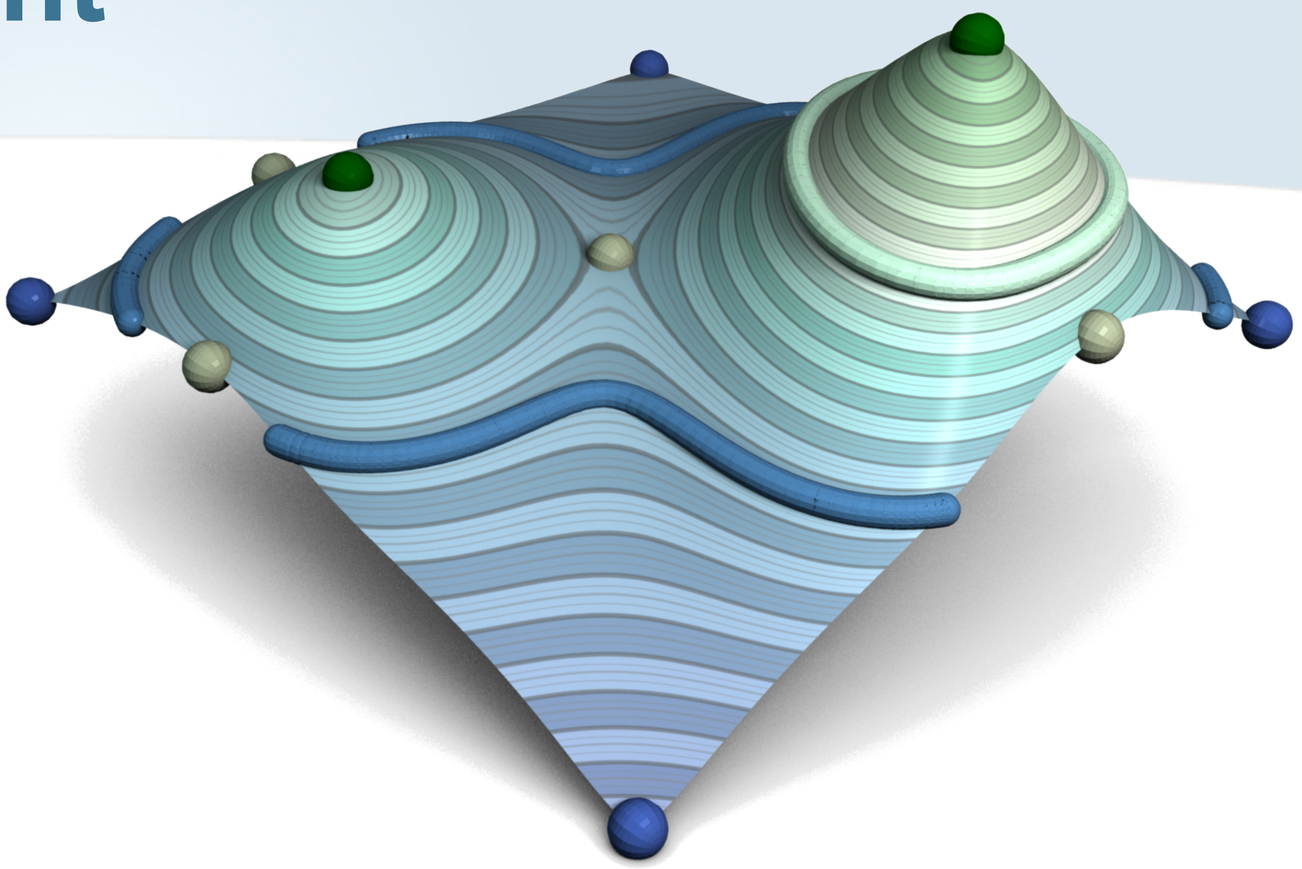


Notion of critical point



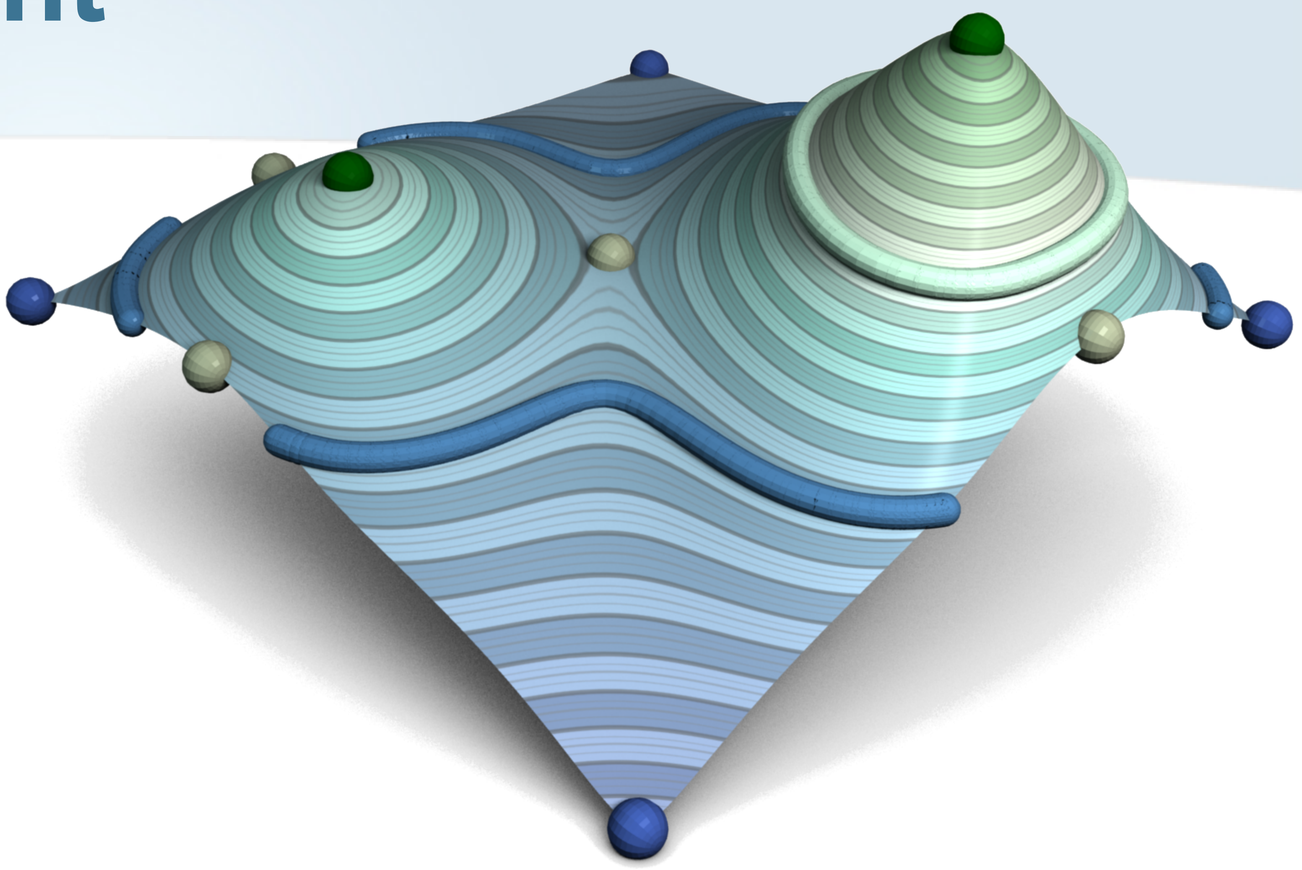
Notion of critical point

- Points where ∇f vanishes



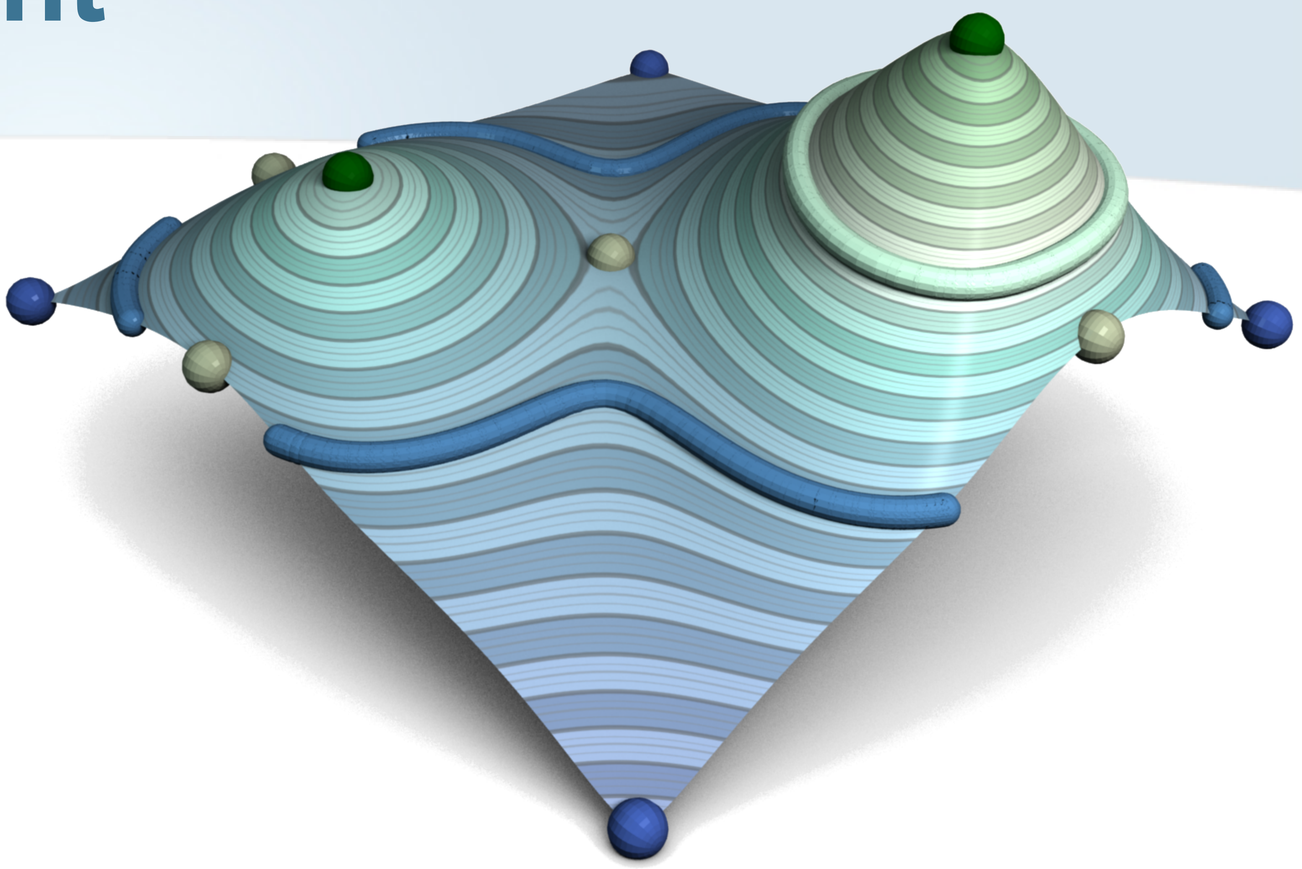
Notion of critical point

- Points where ∇f vanishes
  - Minima, maxima, saddles



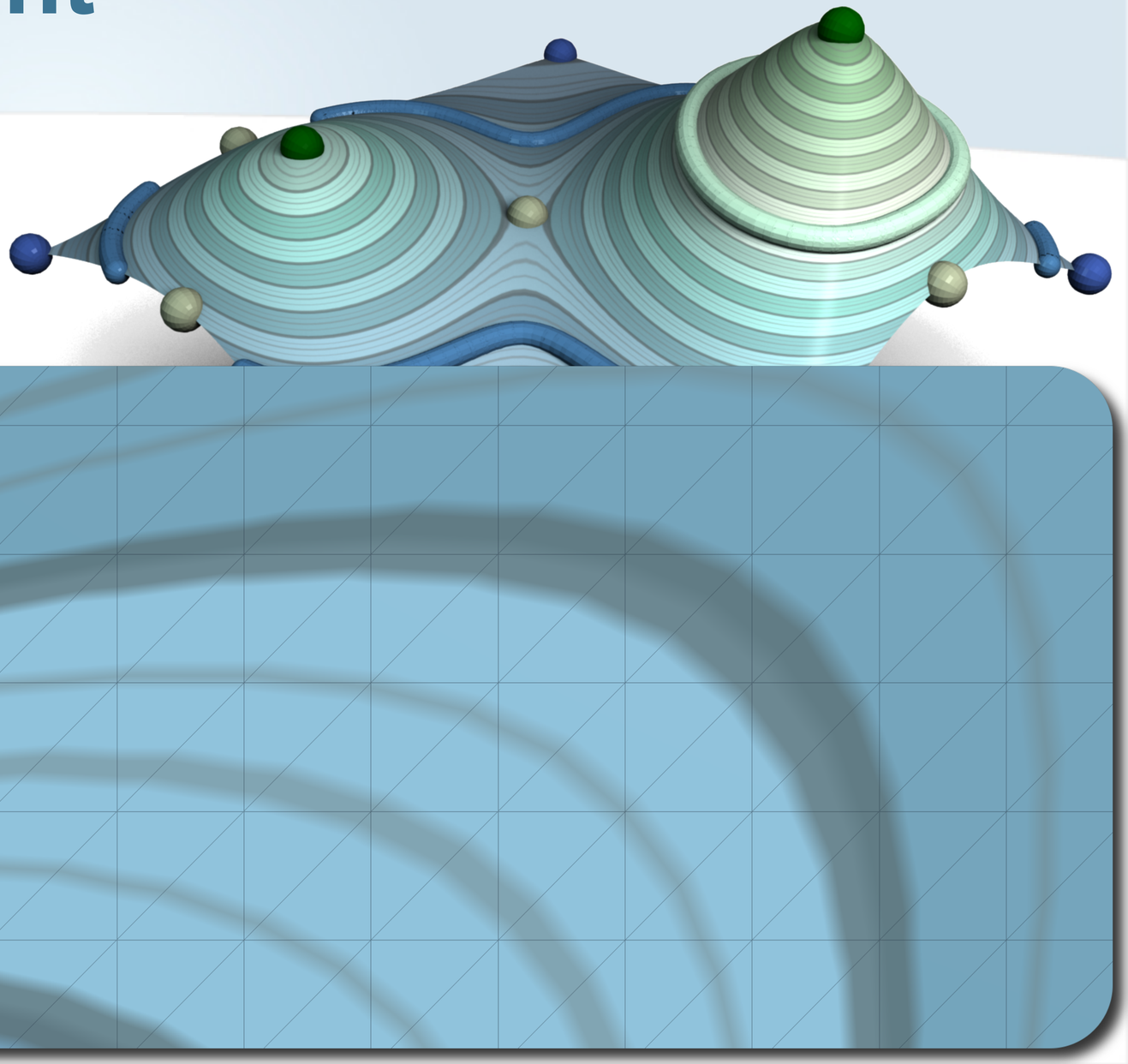
Notion of critical point

- Points where ∇f vanishes
  - Minima, maxima, saddles
  - **Topological changes**



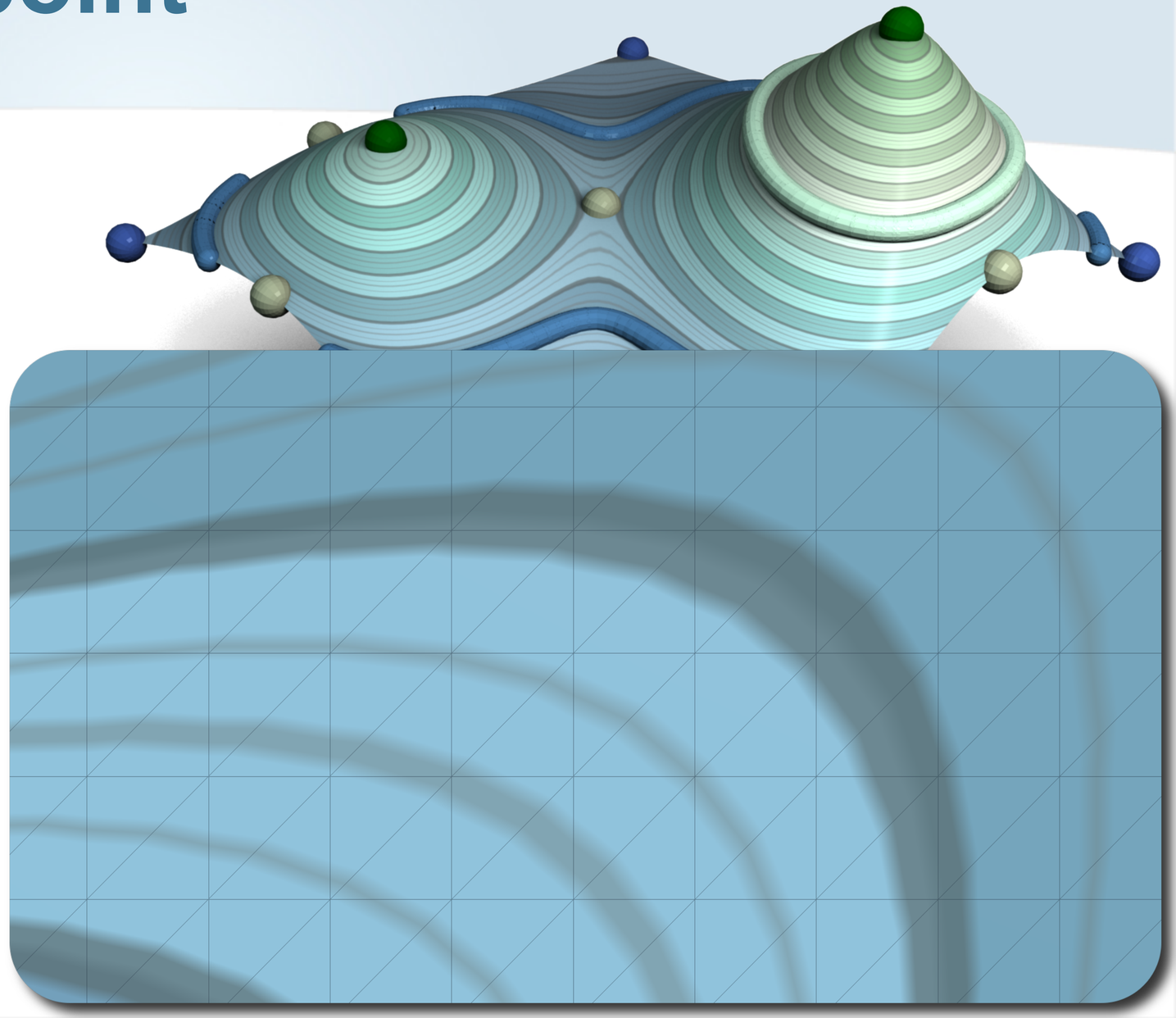
Notion of critical point

- Points where ∇f vanishes
  - Minima, maxima, saddles
  - **Topological changes**
  - PL interpolant



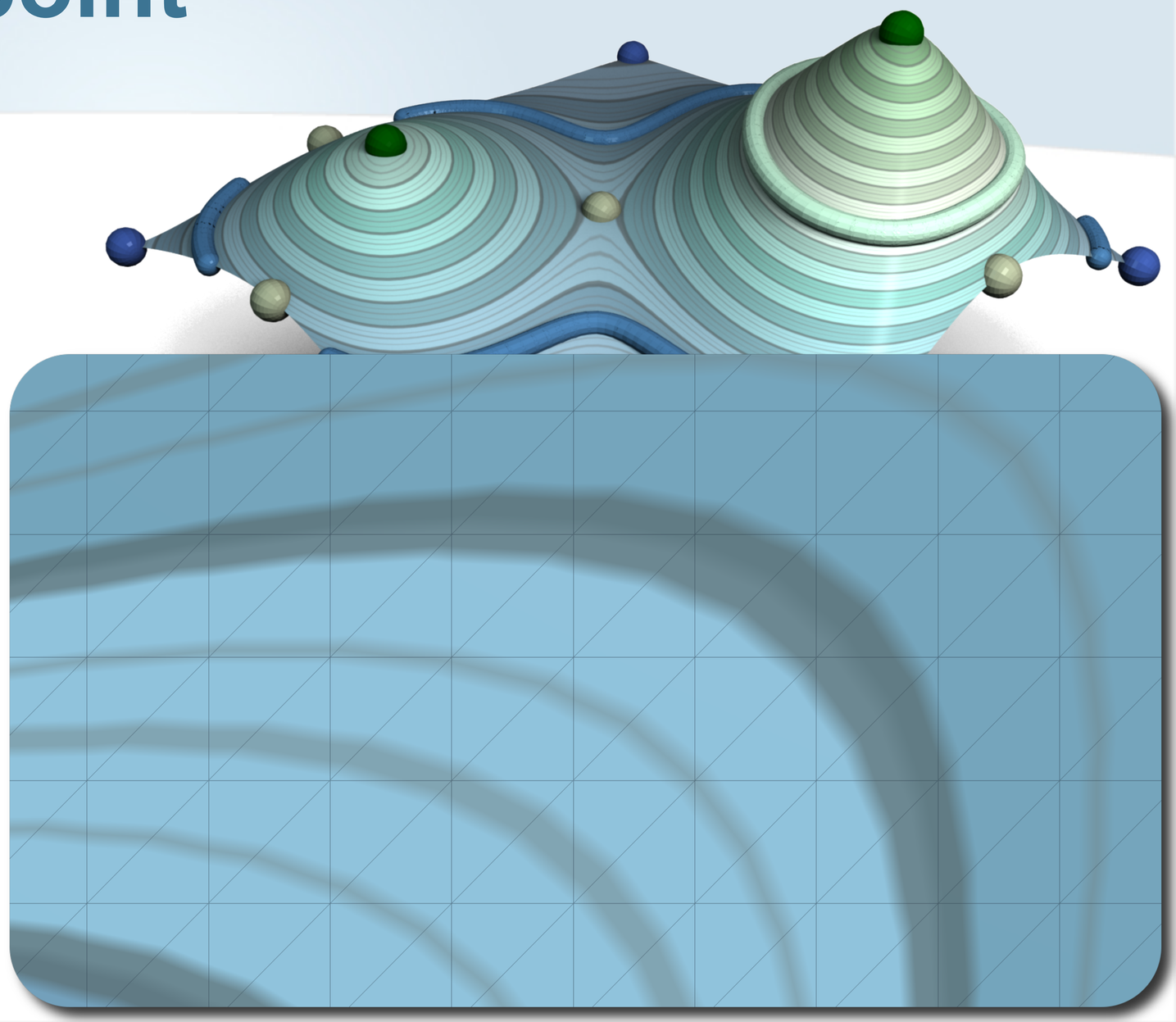
Notion of critical point

- Points where ∇f vanishes
  - Minima, maxima, saddles
  - **Topological changes**
  - PL interpolant
  - Barycentric coordinates on simplices



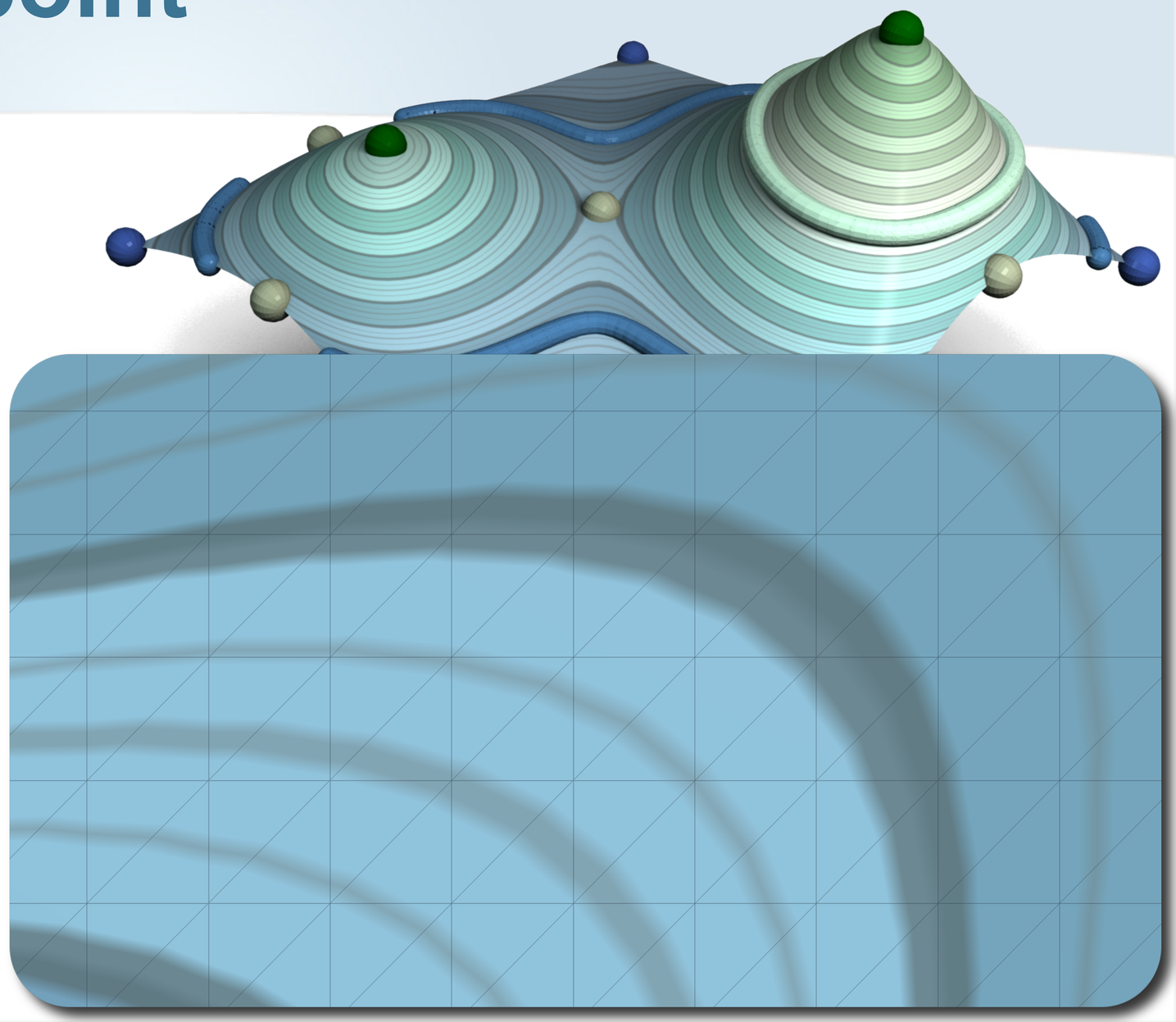
Notion of critical point

- Points where ∇f vanishes
  - Minima, maxima, saddles
  - **Topological changes**
  - PL interpolant
  - Barycentric coordinates on simplices
  - **Only on vertices**



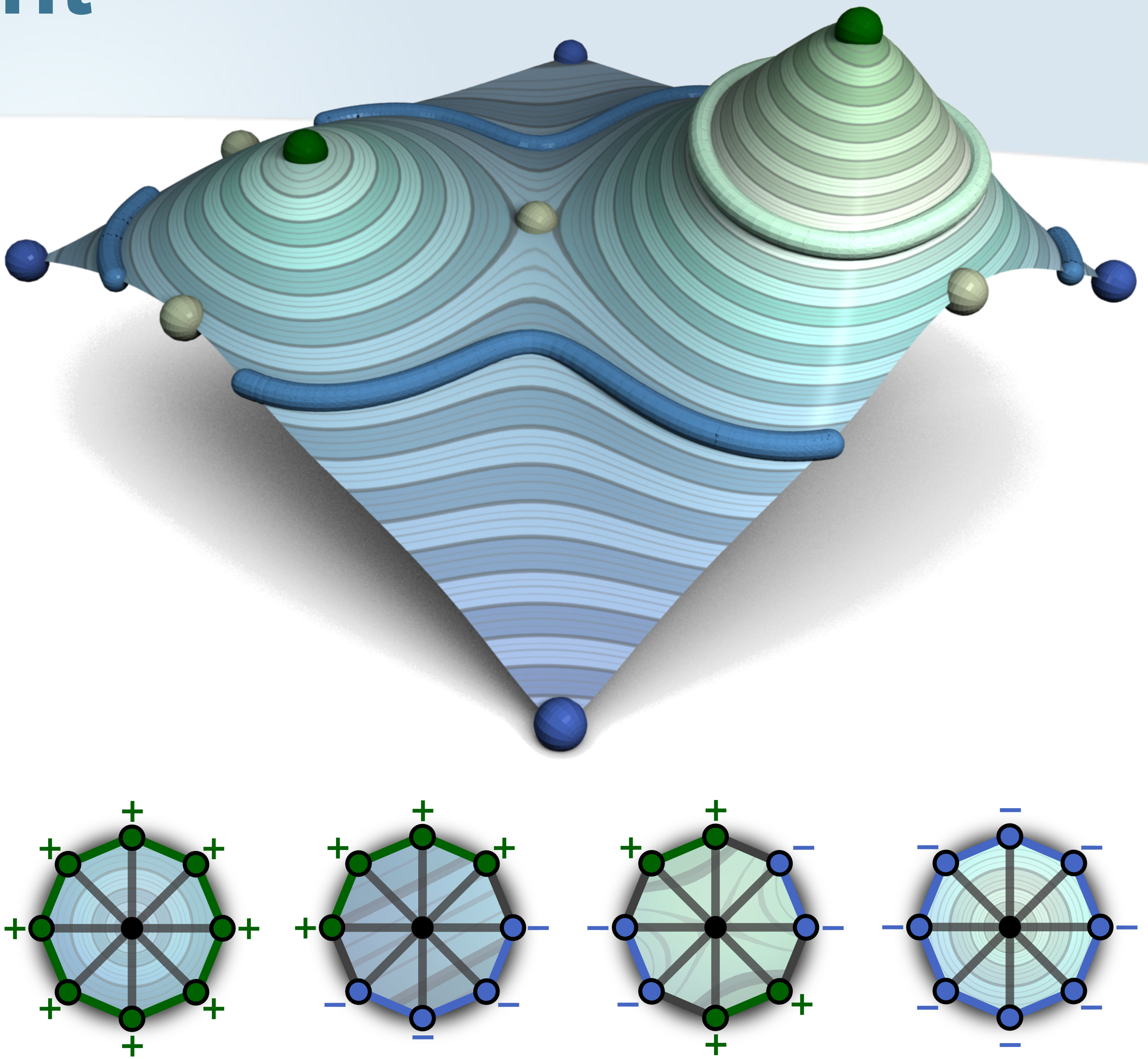
Notion of critical point

- Points where ∇f vanishes
  - Minima, maxima, saddles
  - **Topological changes**
  - PL interpolant
  - Barycentric coordinates on simplices
  - **Only on vertices**
    - Combinatorial characterization



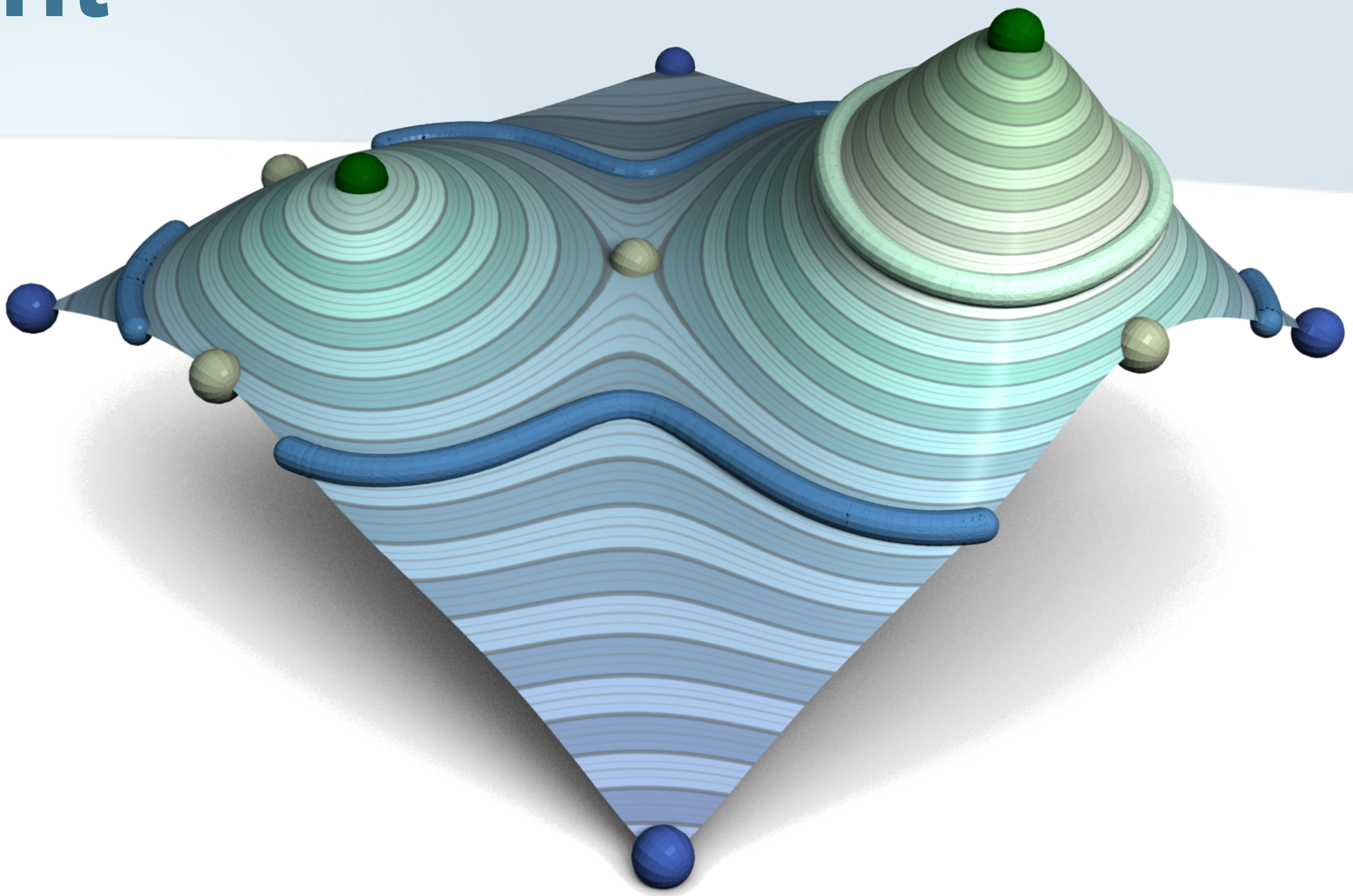
Notion of critical point

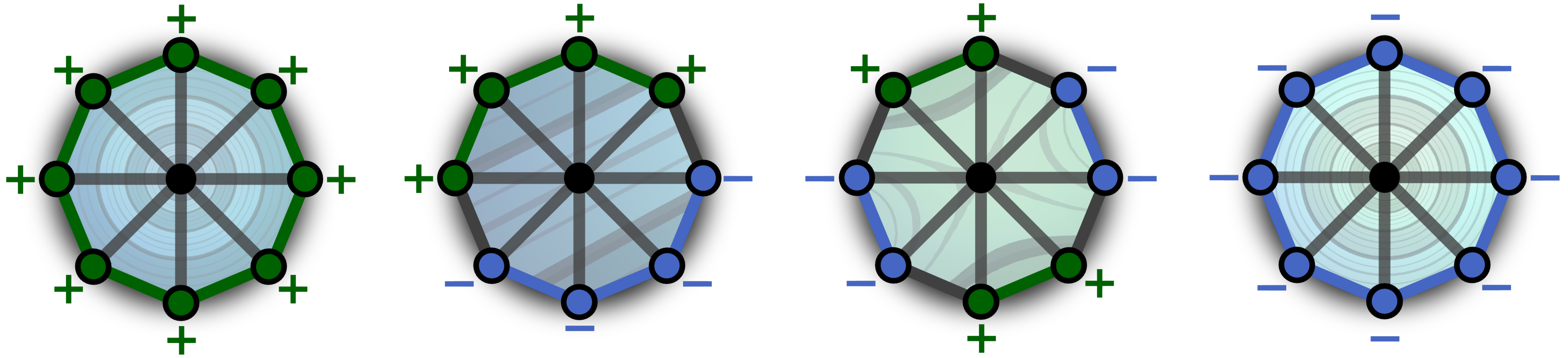
- Combinatorial identification



Notion of critical point

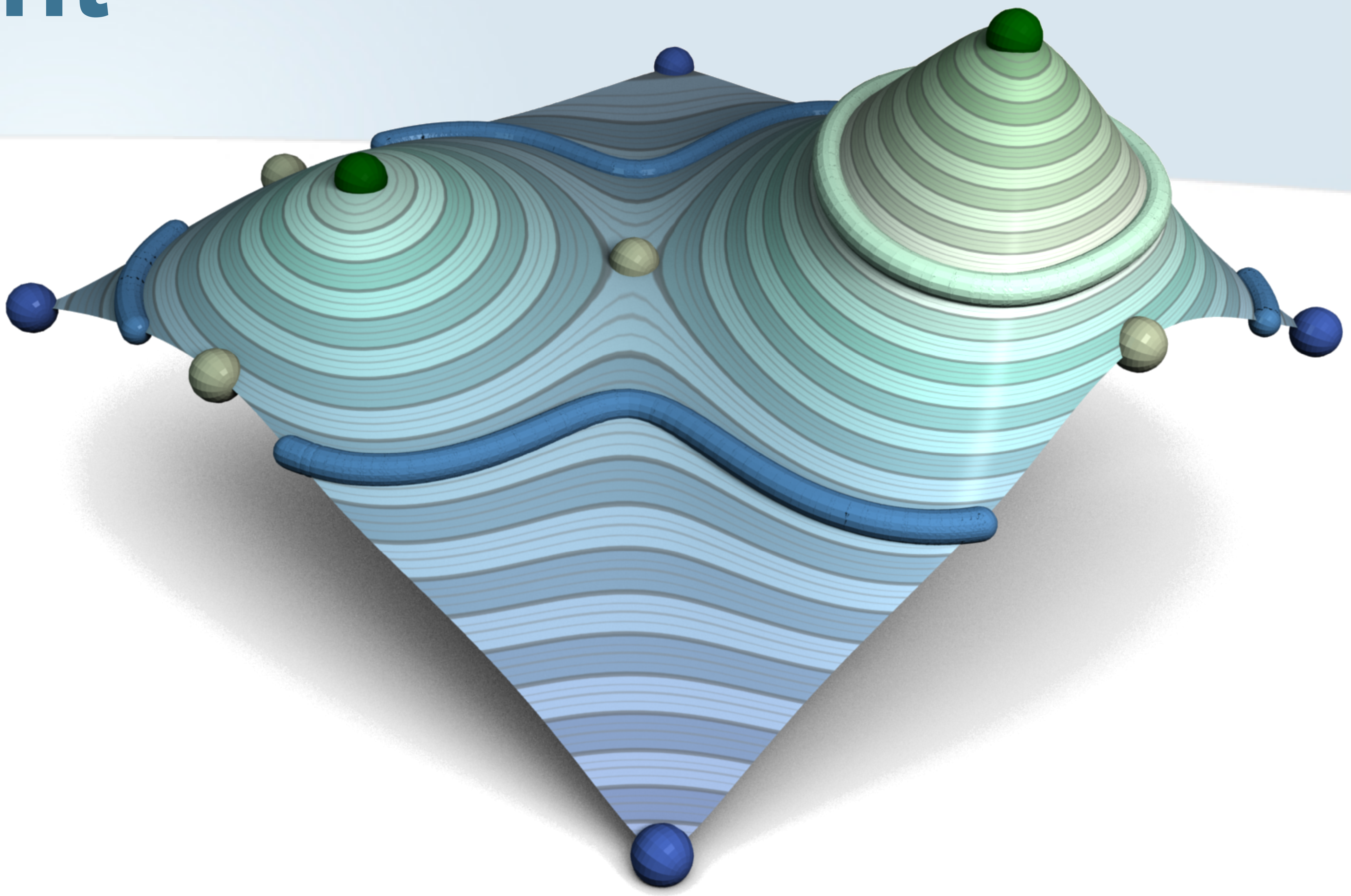
- Combinatorial identification
  - Star of a simplex σ

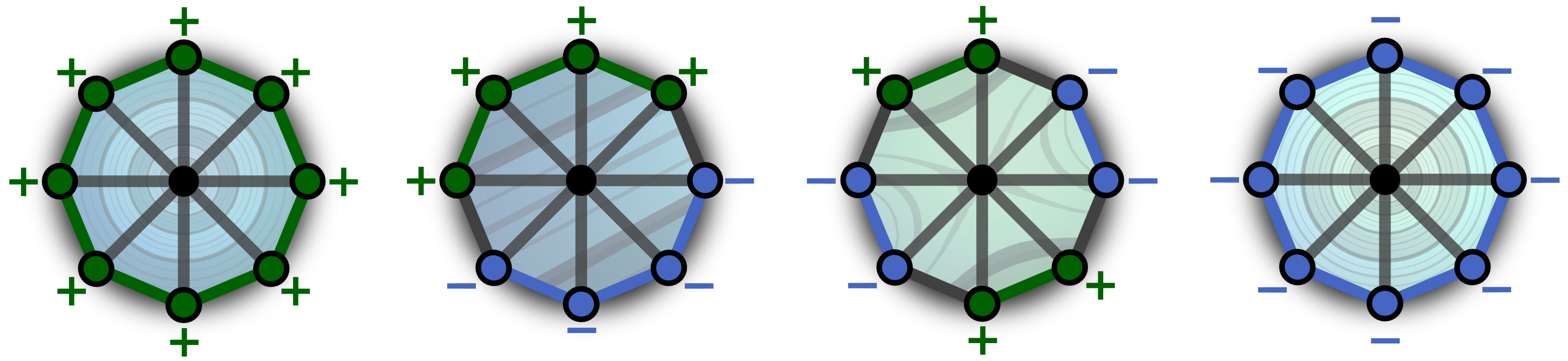




Notion of critical point

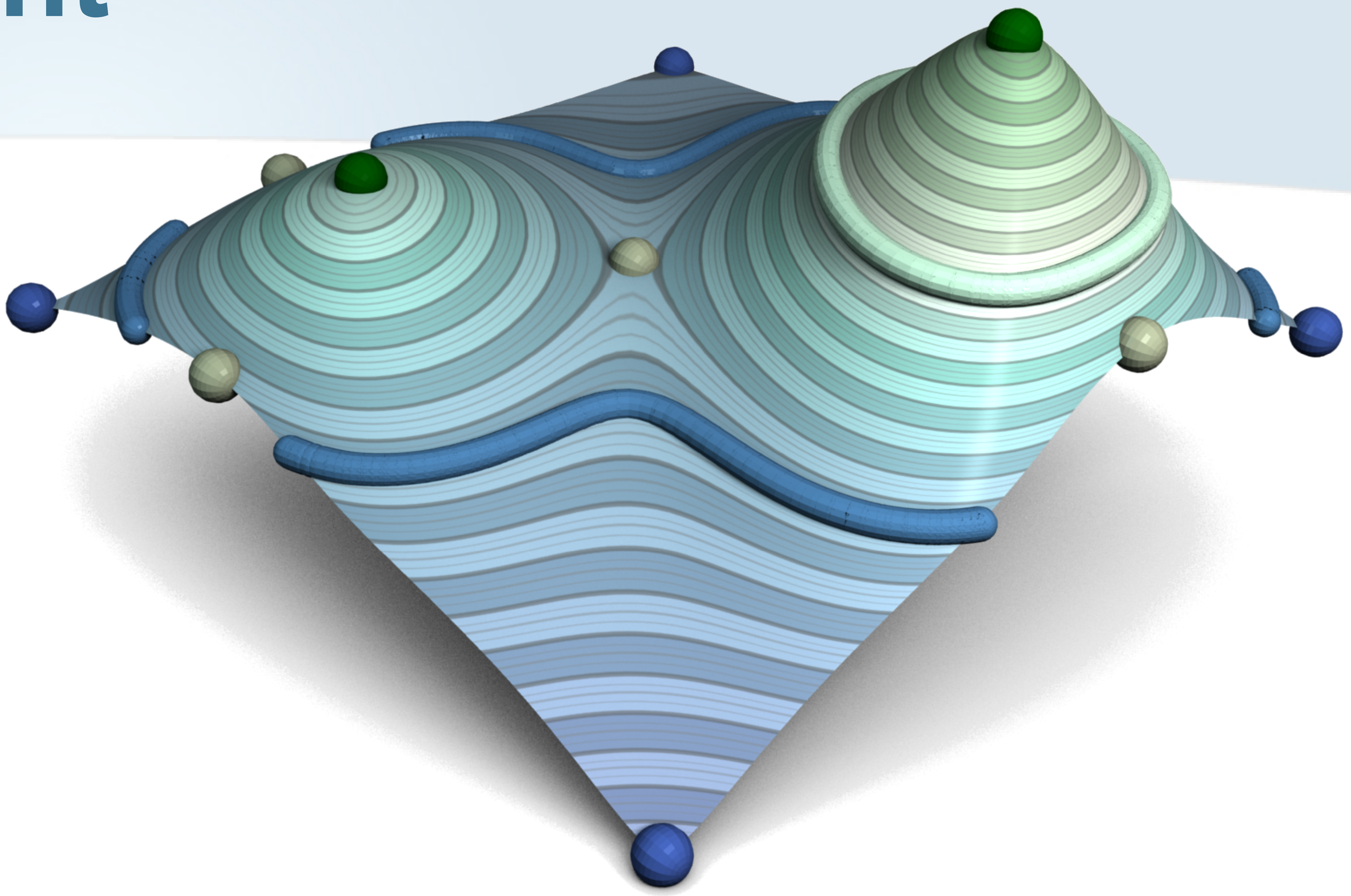
- Combinatorial identification
  - Star of a simplex σ
    - Simplices that contain σ as a face

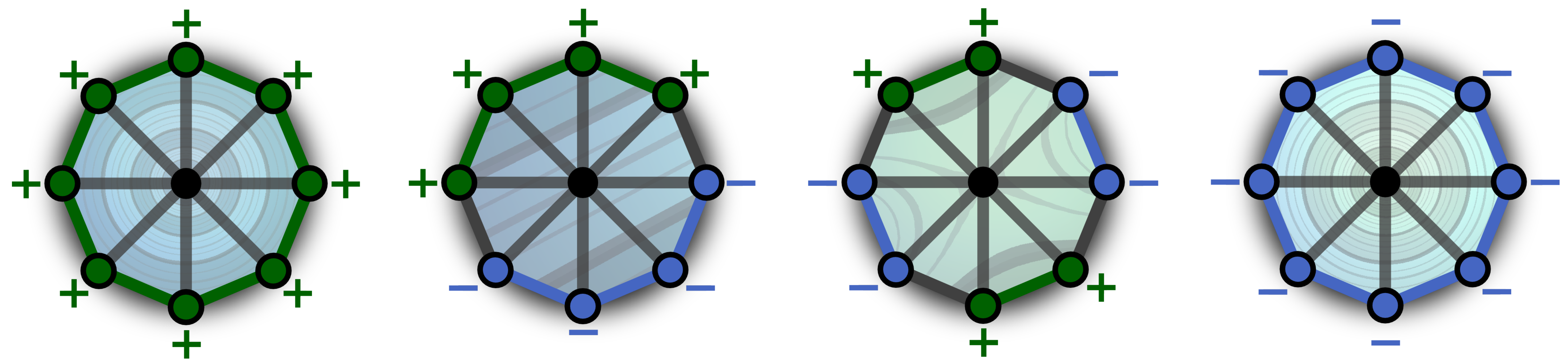




Notion of critical point

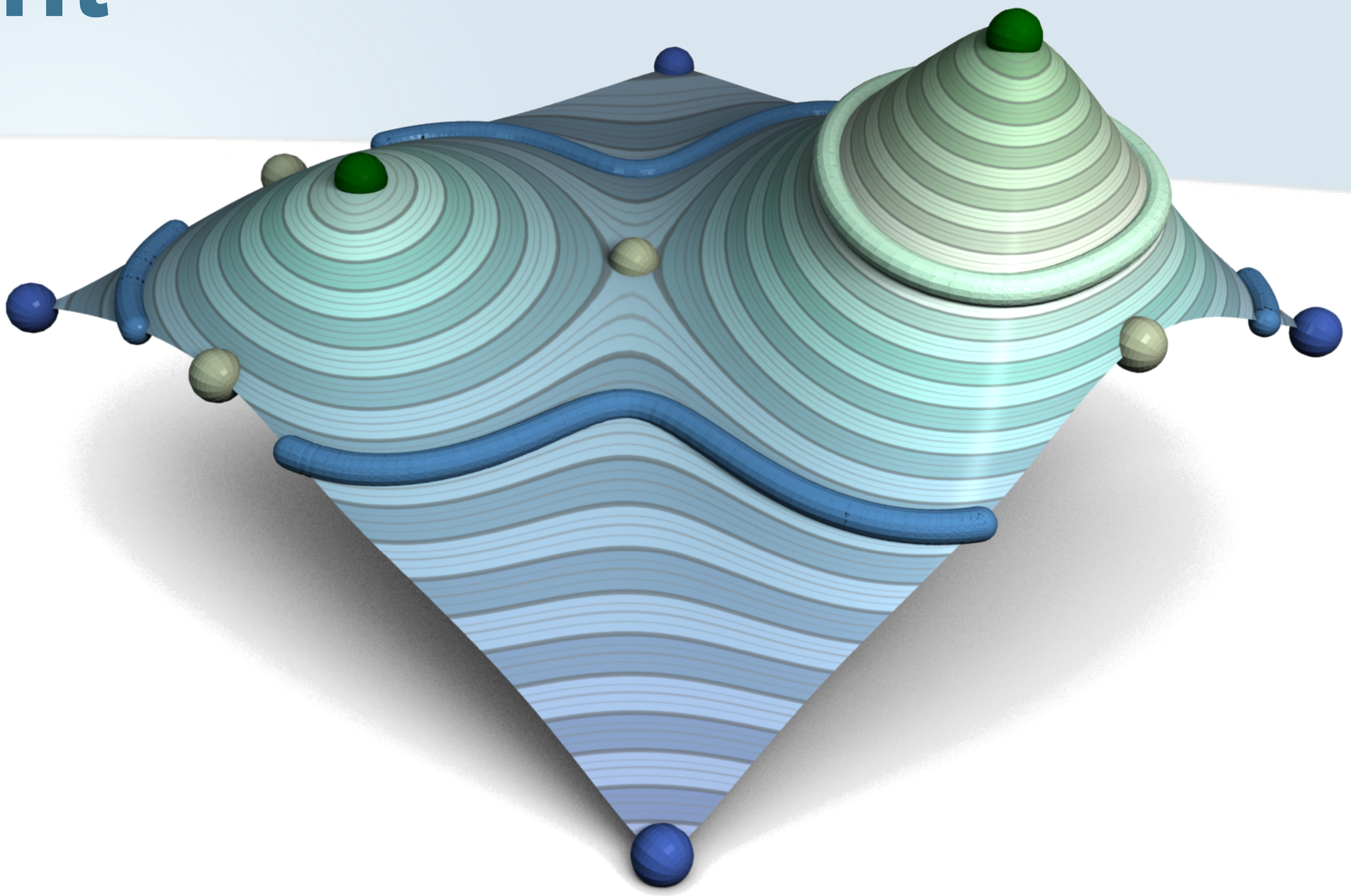
- Combinatorial identification
  - Star of a simplex σ
    - Simplices that contain σ as a face
  - Link of a simplex

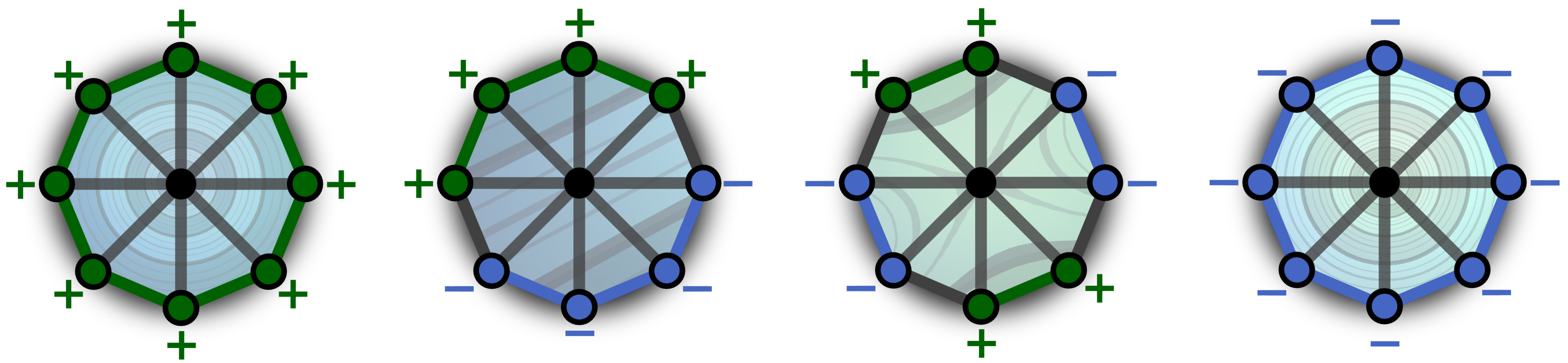




Notion of critical point

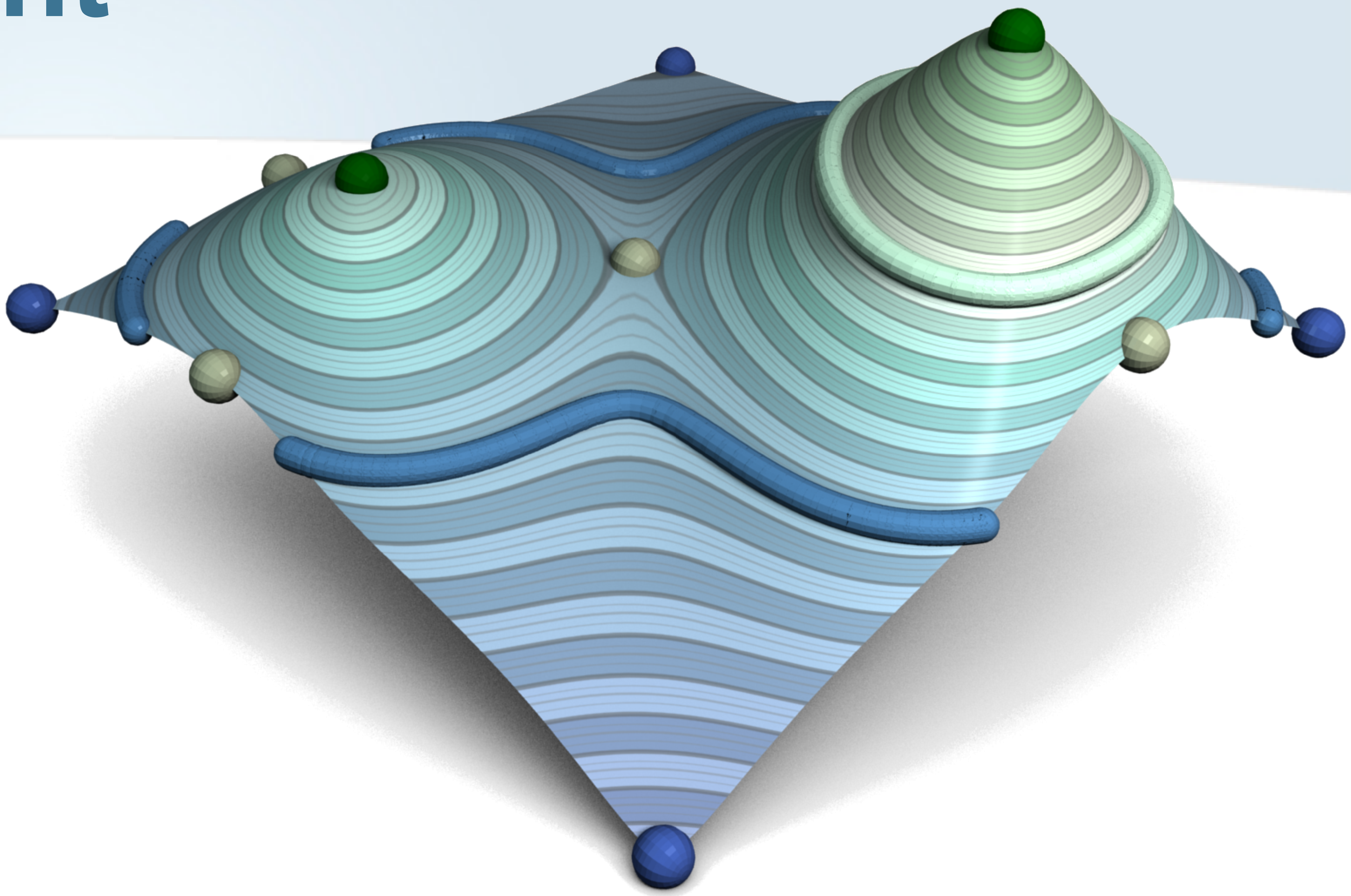
- Combinatorial identification
  - Star of a simplex σ
    - Simplices that contain σ as a face
  - Link of a simplex
    - Simplices of the closure of the star that do not contain σ as a face

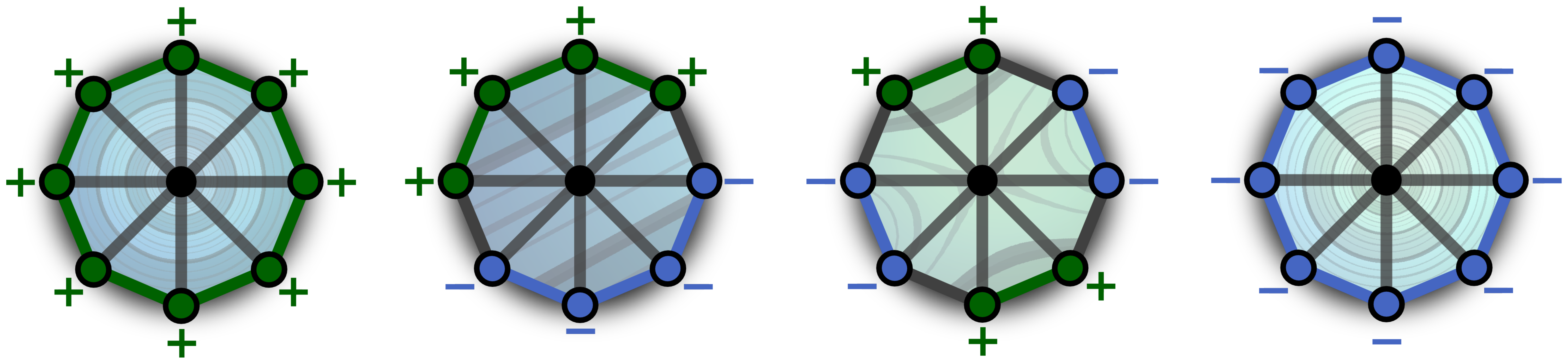




Notion of critical point

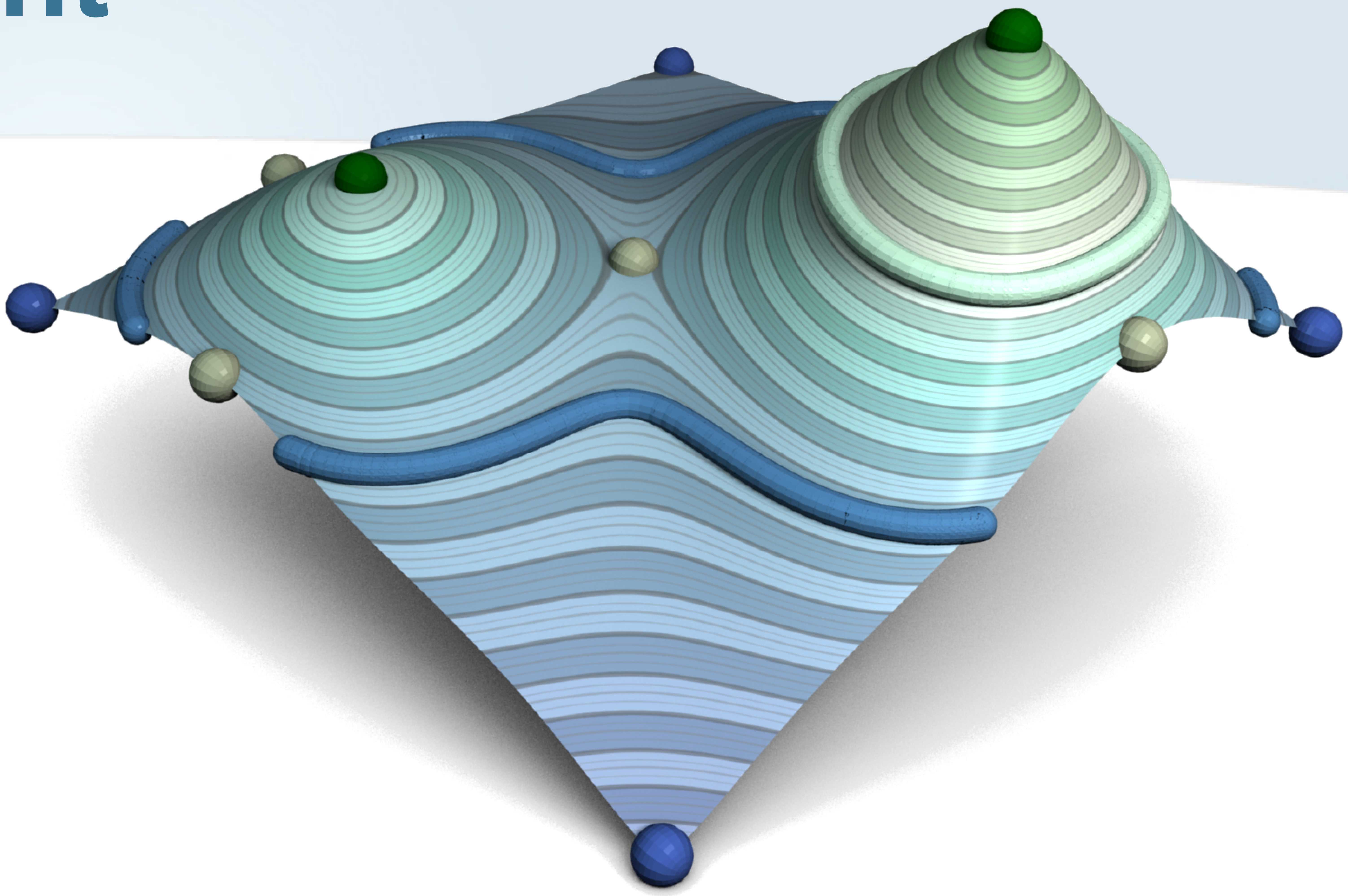
- Combinatorial identification
  - Lower link of v

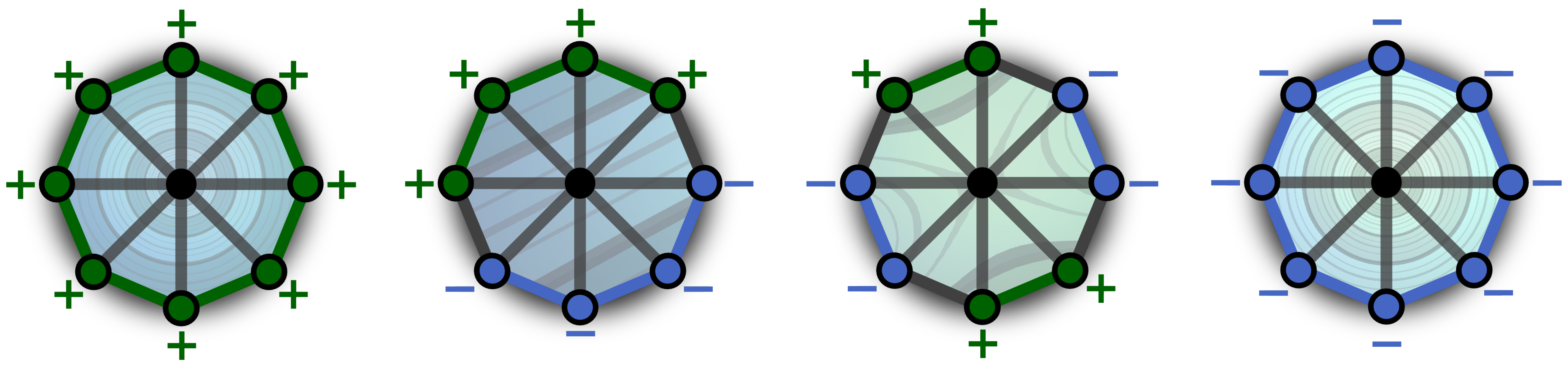




Notion of critical point

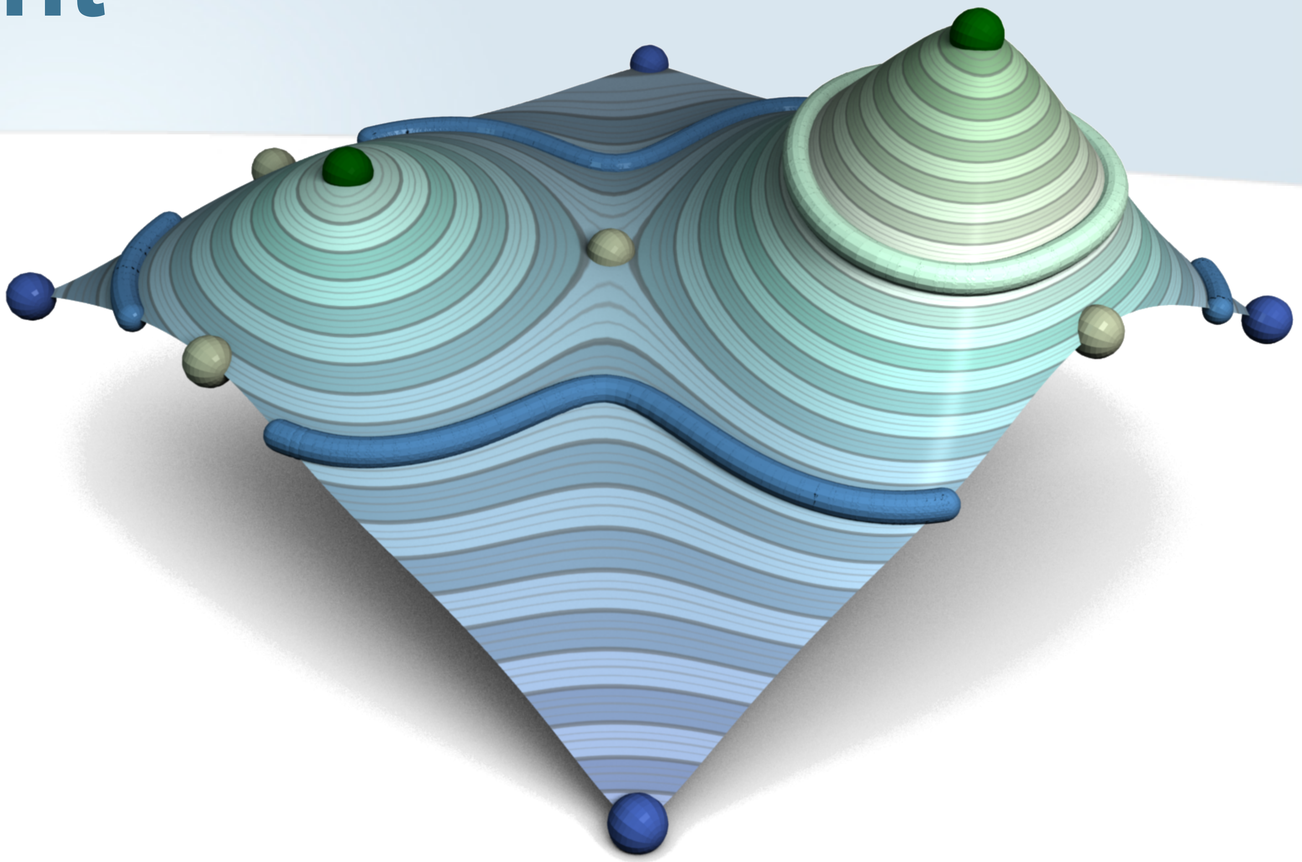
- Combinatorial identification
  - Lower link of v
    - Simplices whose function values are strictly below $f(v)$

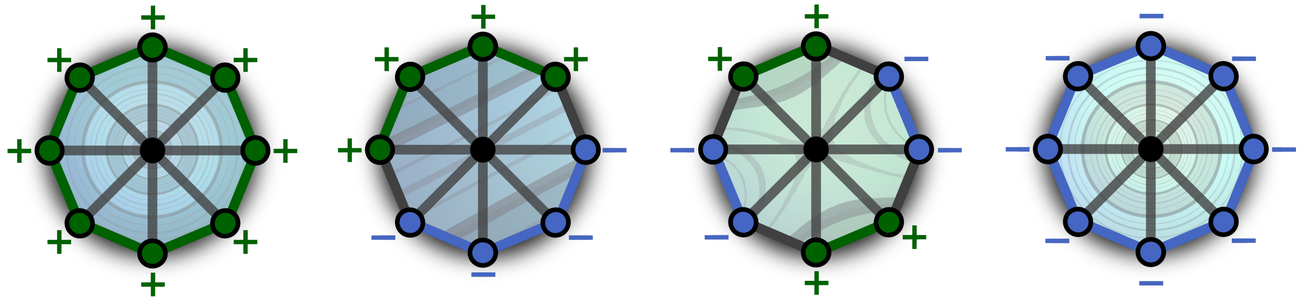




Notion of critical point

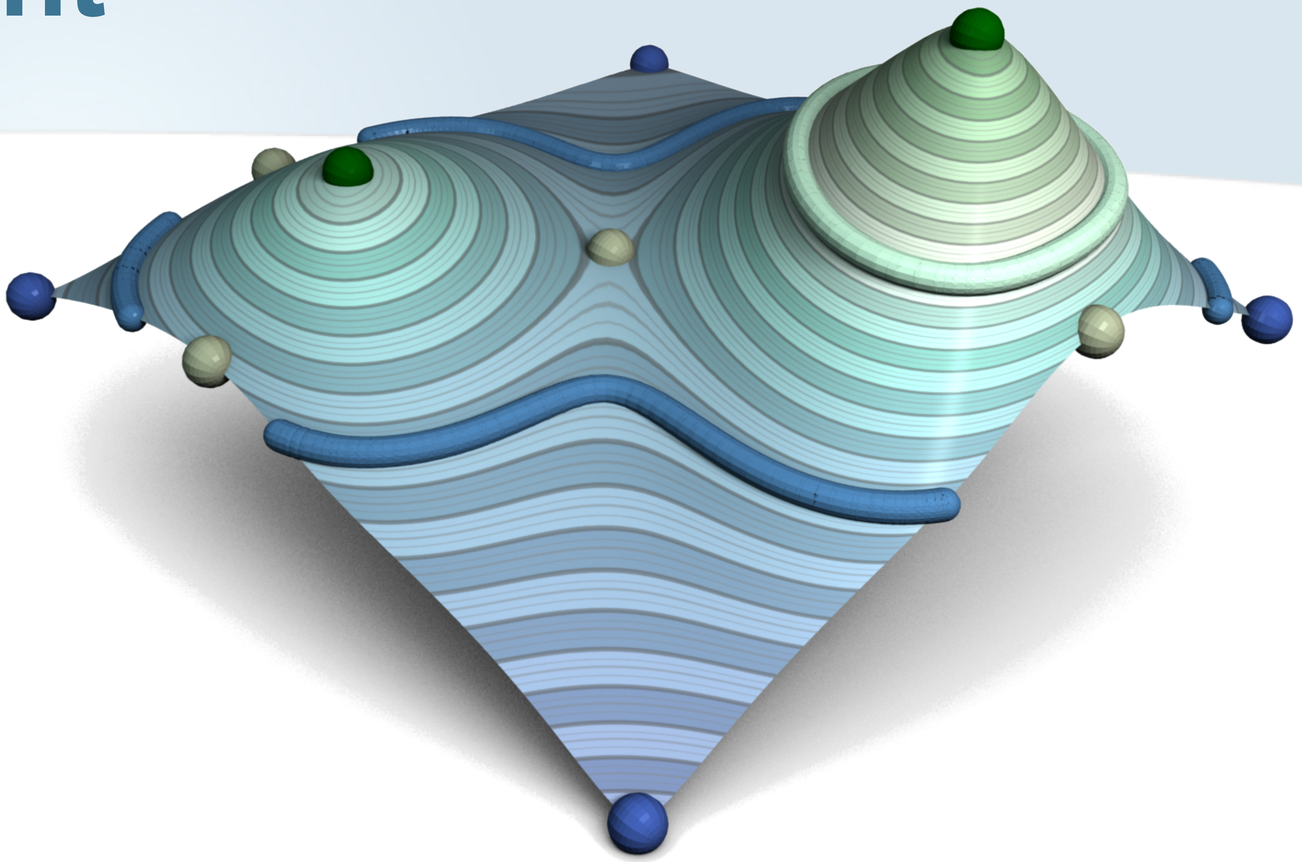
- Combinatorial identification
  - Lower link of v
    - Simplices whose function values are strictly below $f(v)$
  - Upper link of v

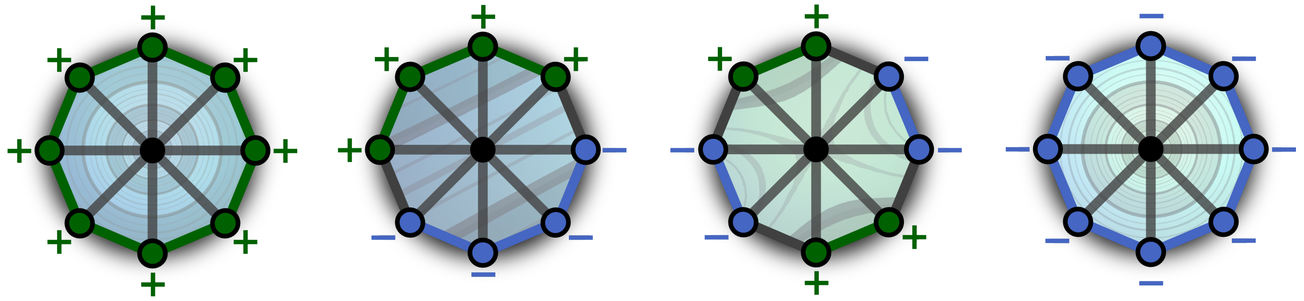




Notion of critical point

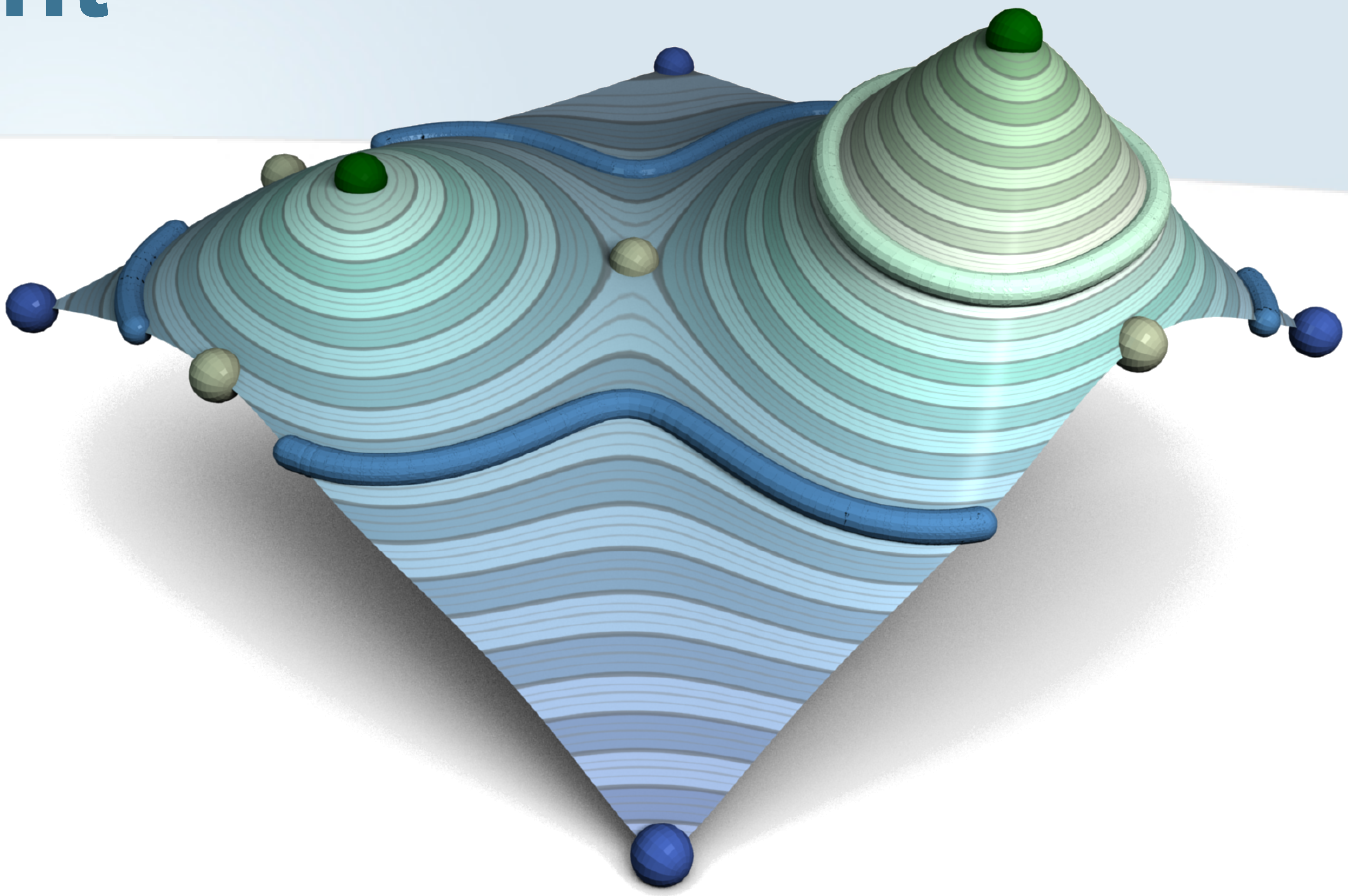
- Combinatorial identification
  - Lower link of v
    - Simplices whose function values are strictly below $f(v)$
  - Upper link of v
    - Simplices whose function values are strictly above $f(v)$

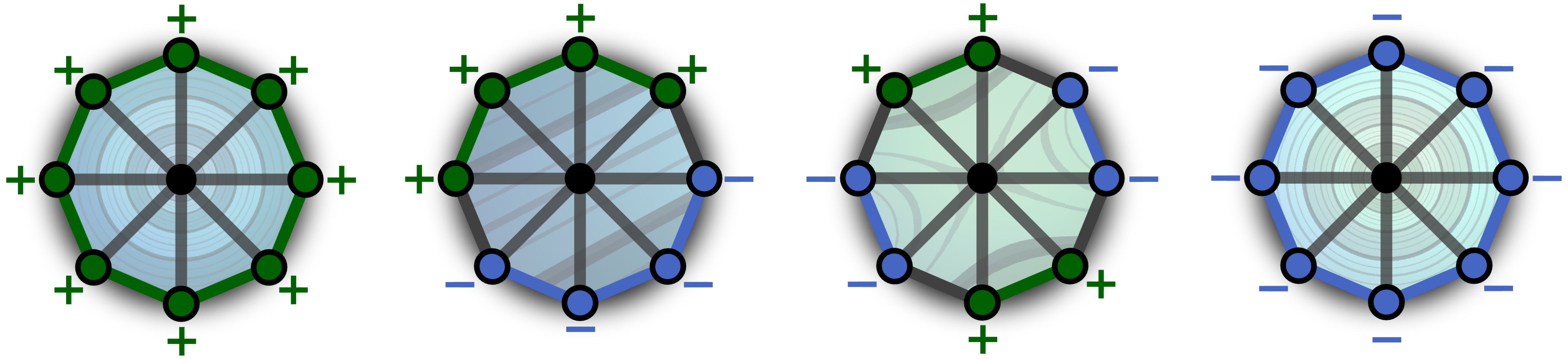




Notion of critical point

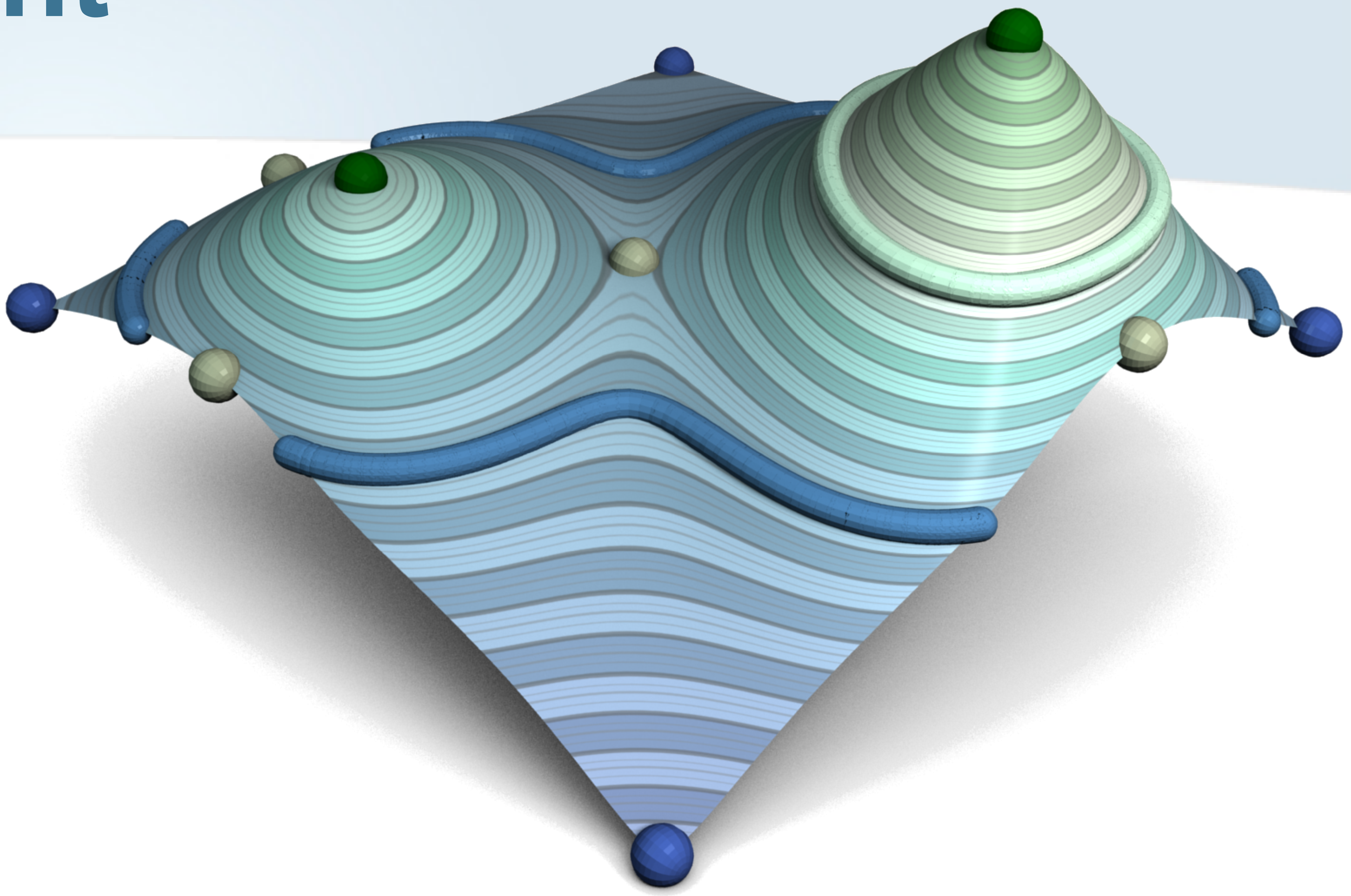
- Combinatorial identification

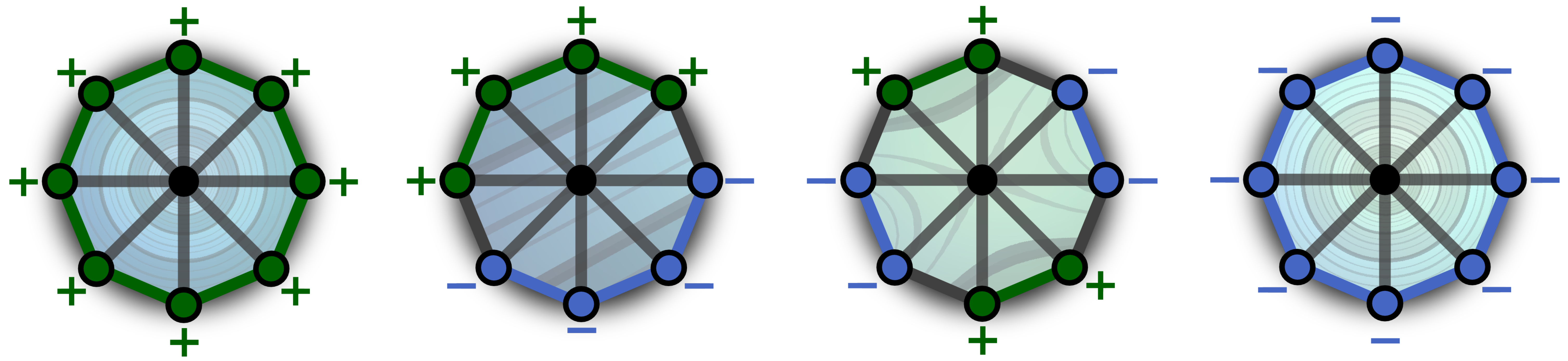




Notion of critical point

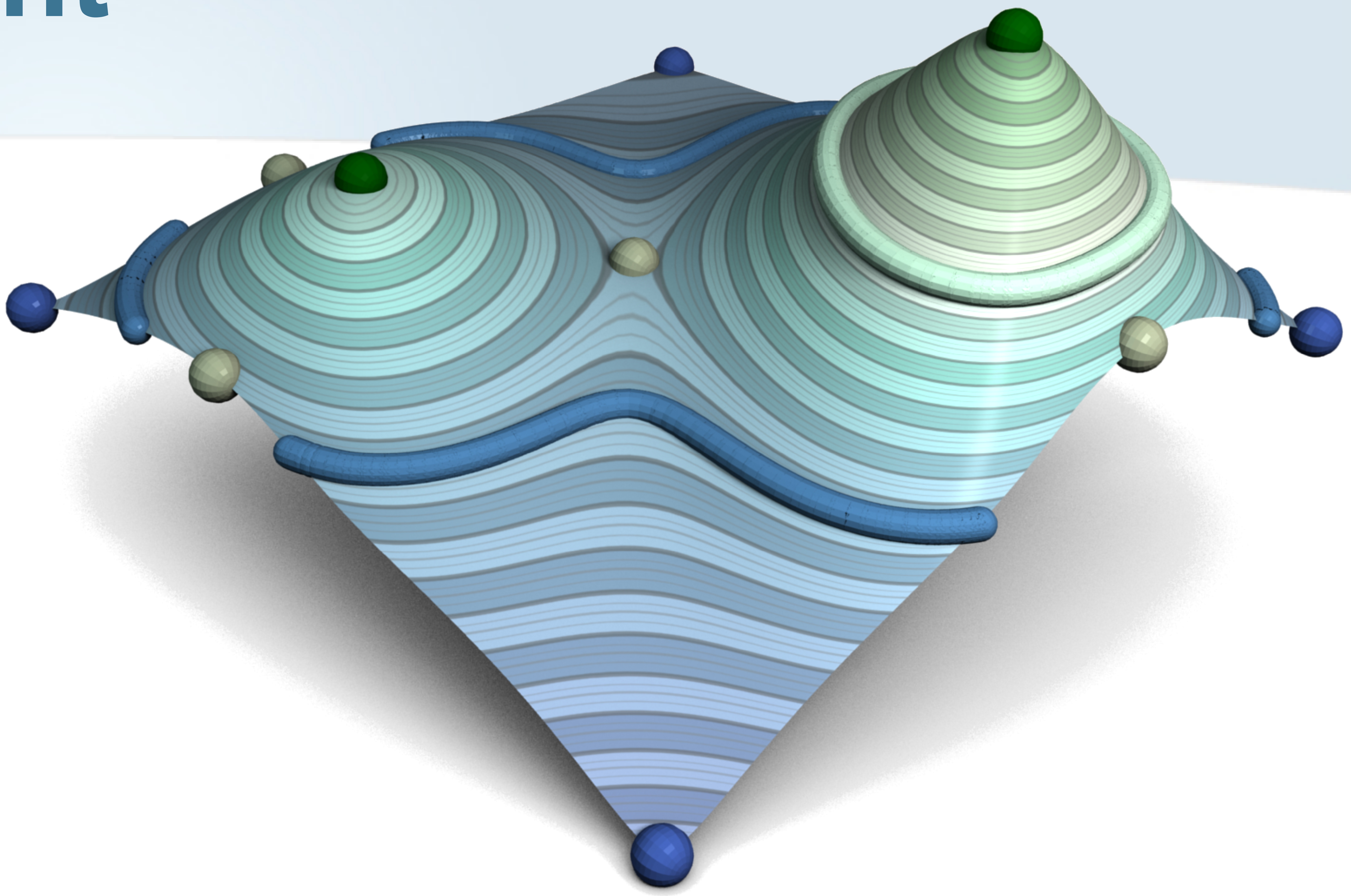
- Combinatorial identification
  - Minimum

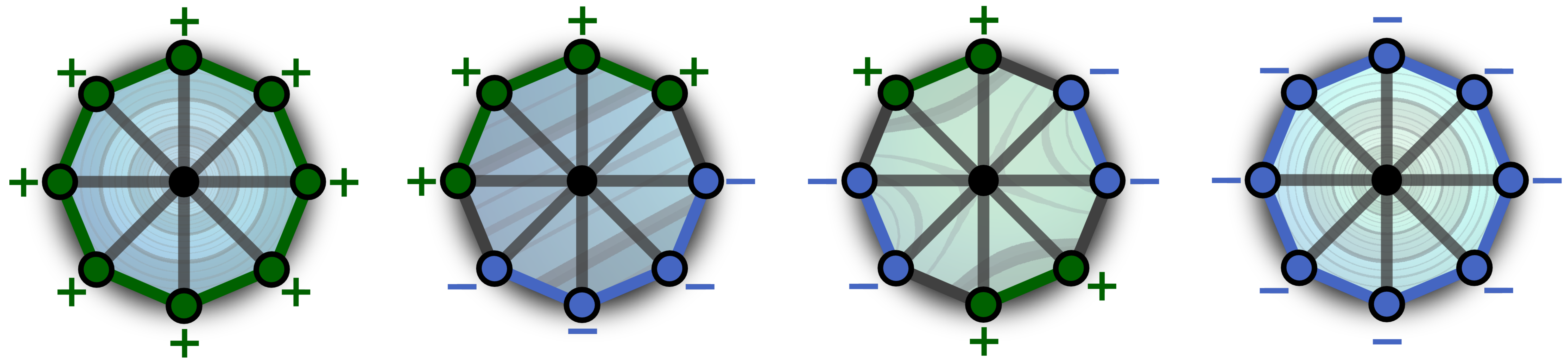




Notion of critical point

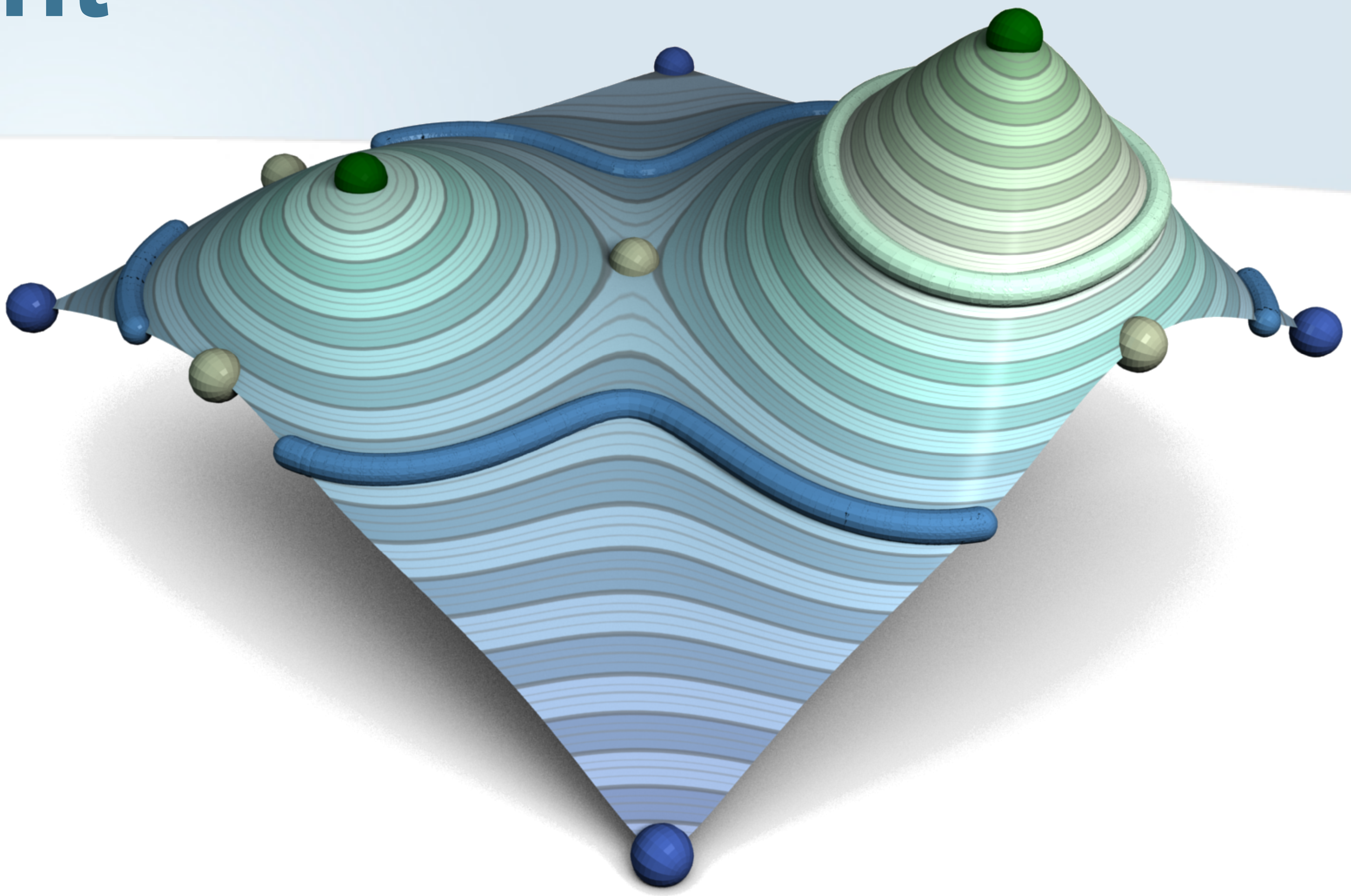
- Combinatorial identification
  - Minimum
    - Empty lower link

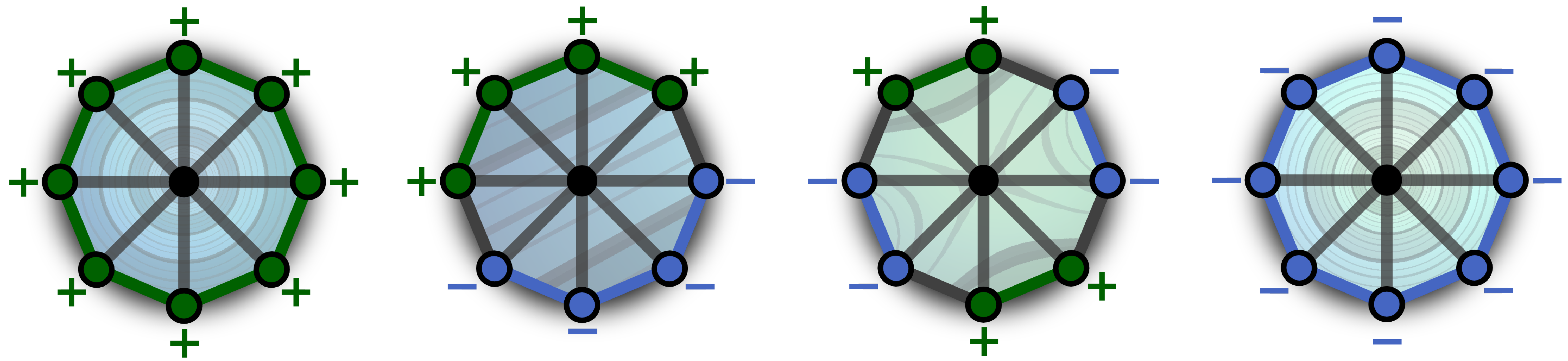




Notion of critical point

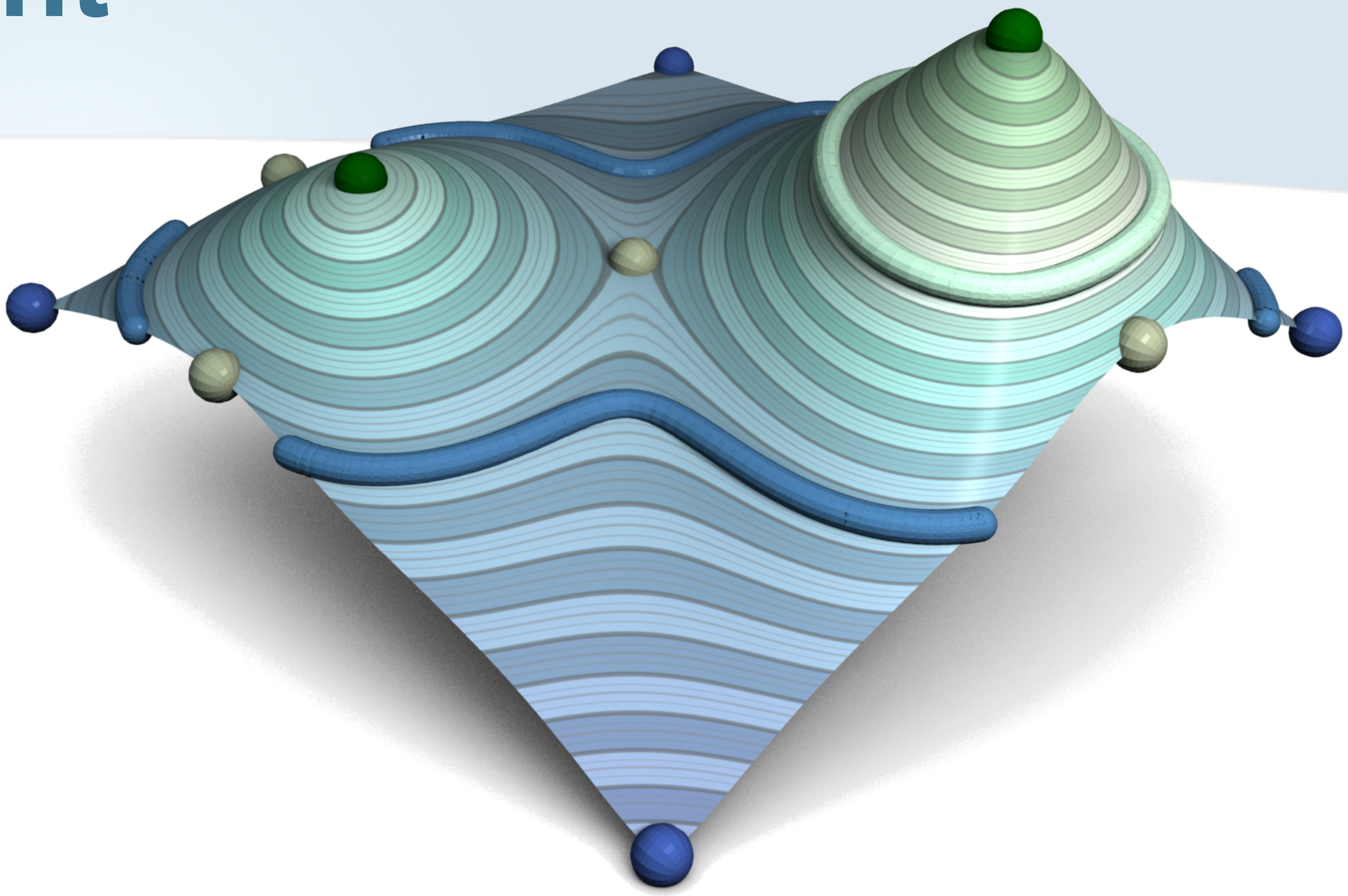
- Combinatorial identification
  - Minimum
    - Empty lower link
  - Maximum

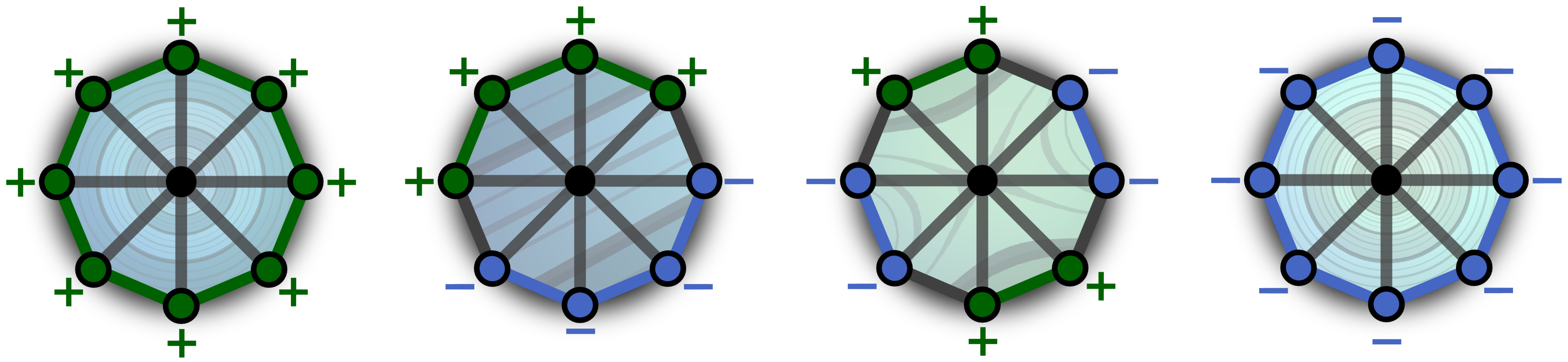




Notion of critical point

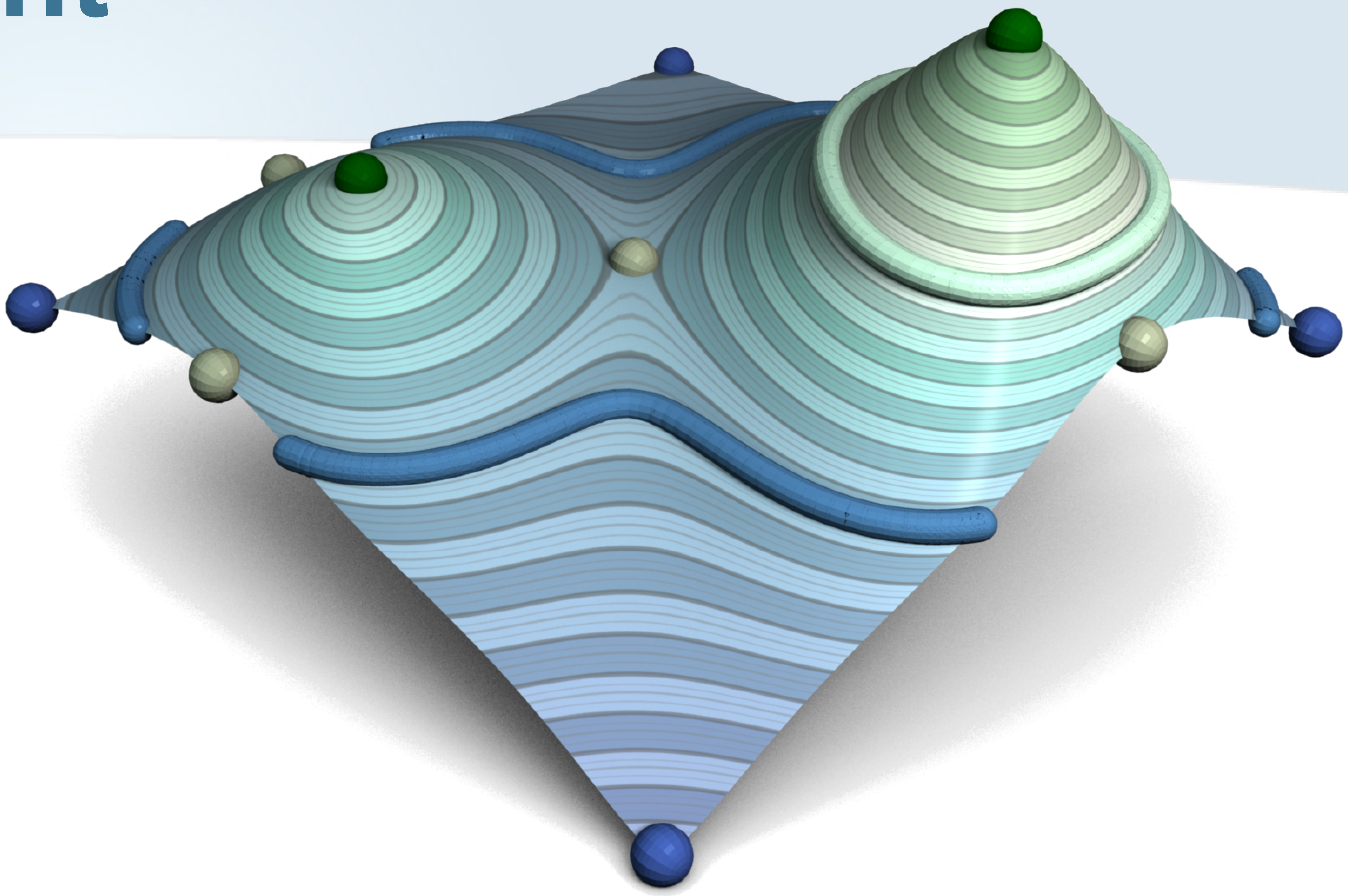
- Combinatorial identification
  - Minimum
    - Empty lower link
  - Maximum
    - Empty upper link

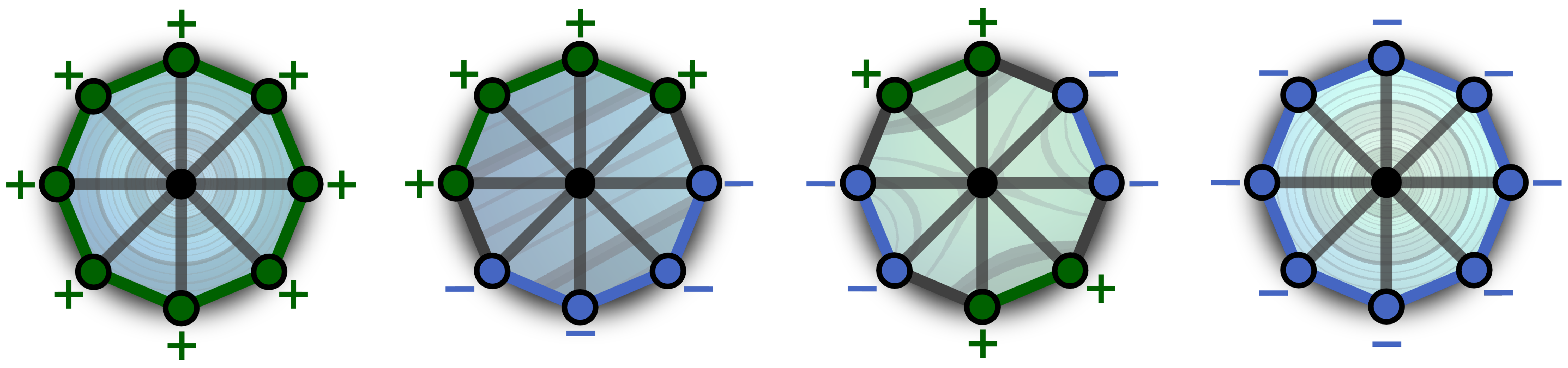




Notion of critical point

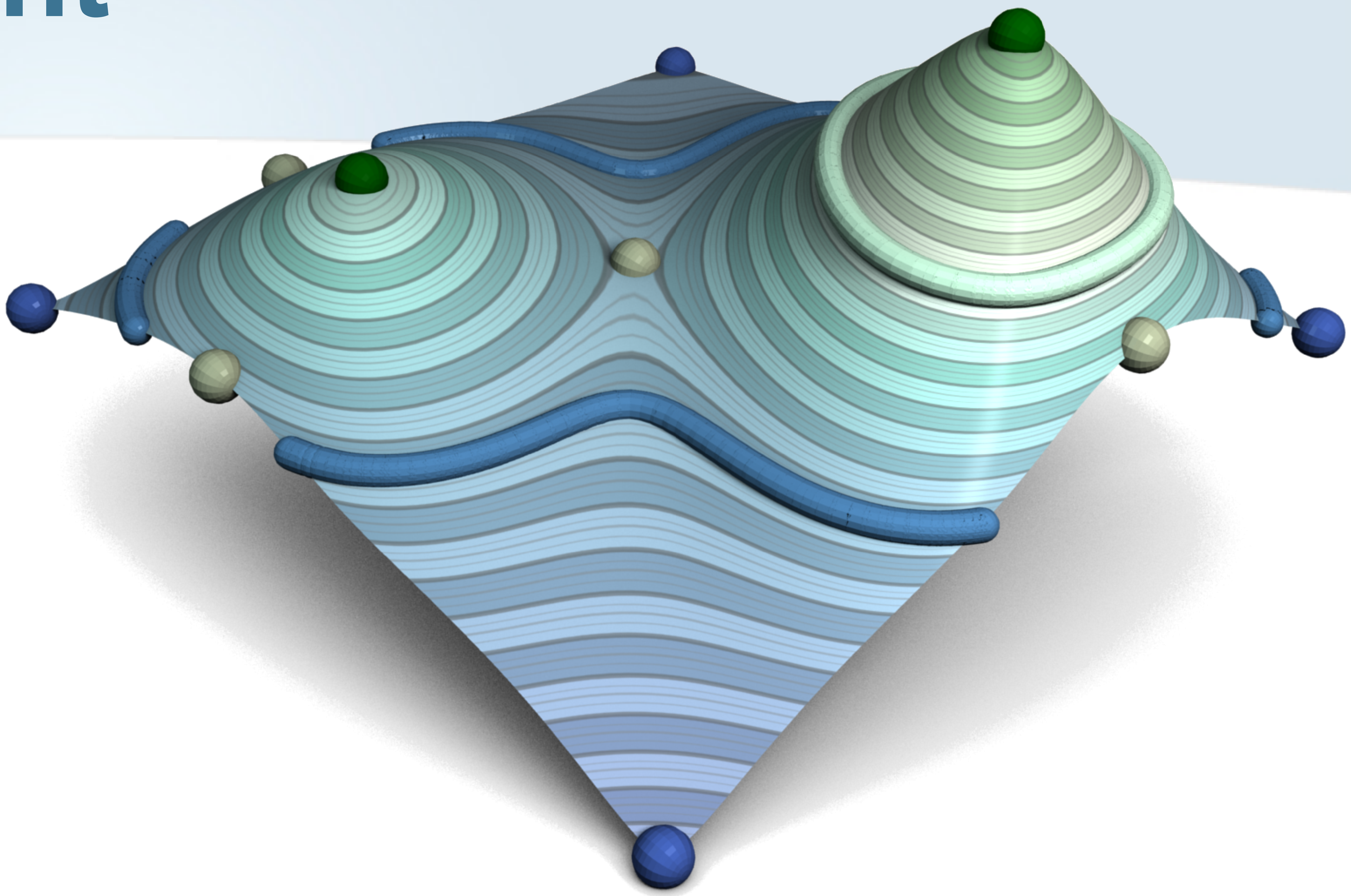
- Combinatorial identification
  - Minimum
    - Empty lower link
  - Maximum
    - Empty upper link
  - Regular point

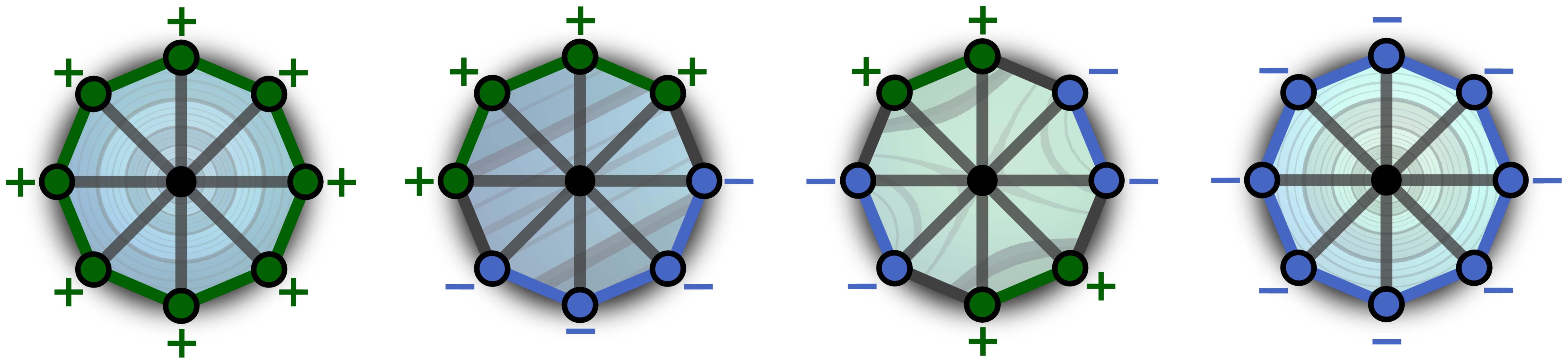




Notion of critical point

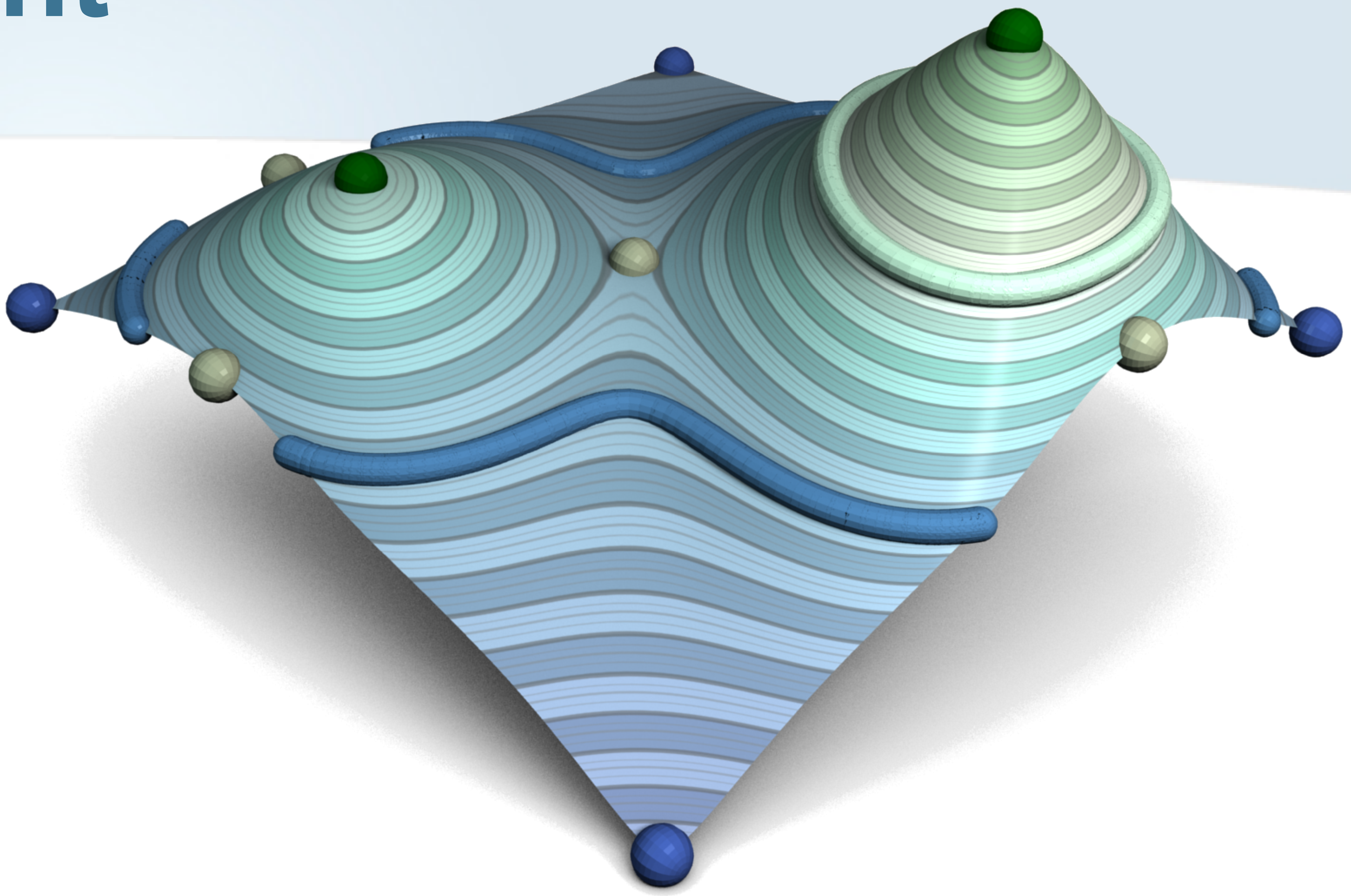
- Combinatorial identification
  - Minimum
    - Empty lower link
  - Maximum
    - Empty upper link
  - Regular point
    - Lower and upper links both *simply connected*

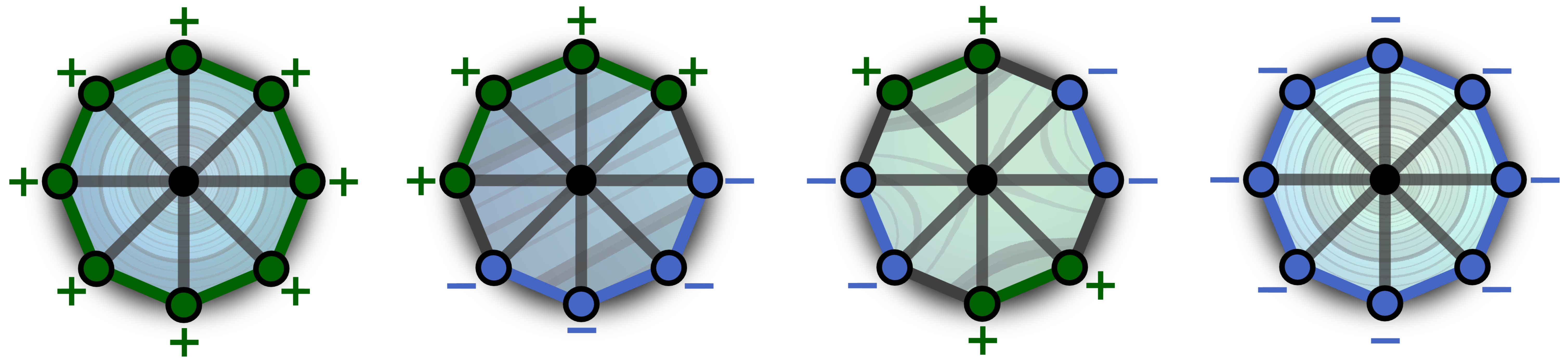




Notion of critical point

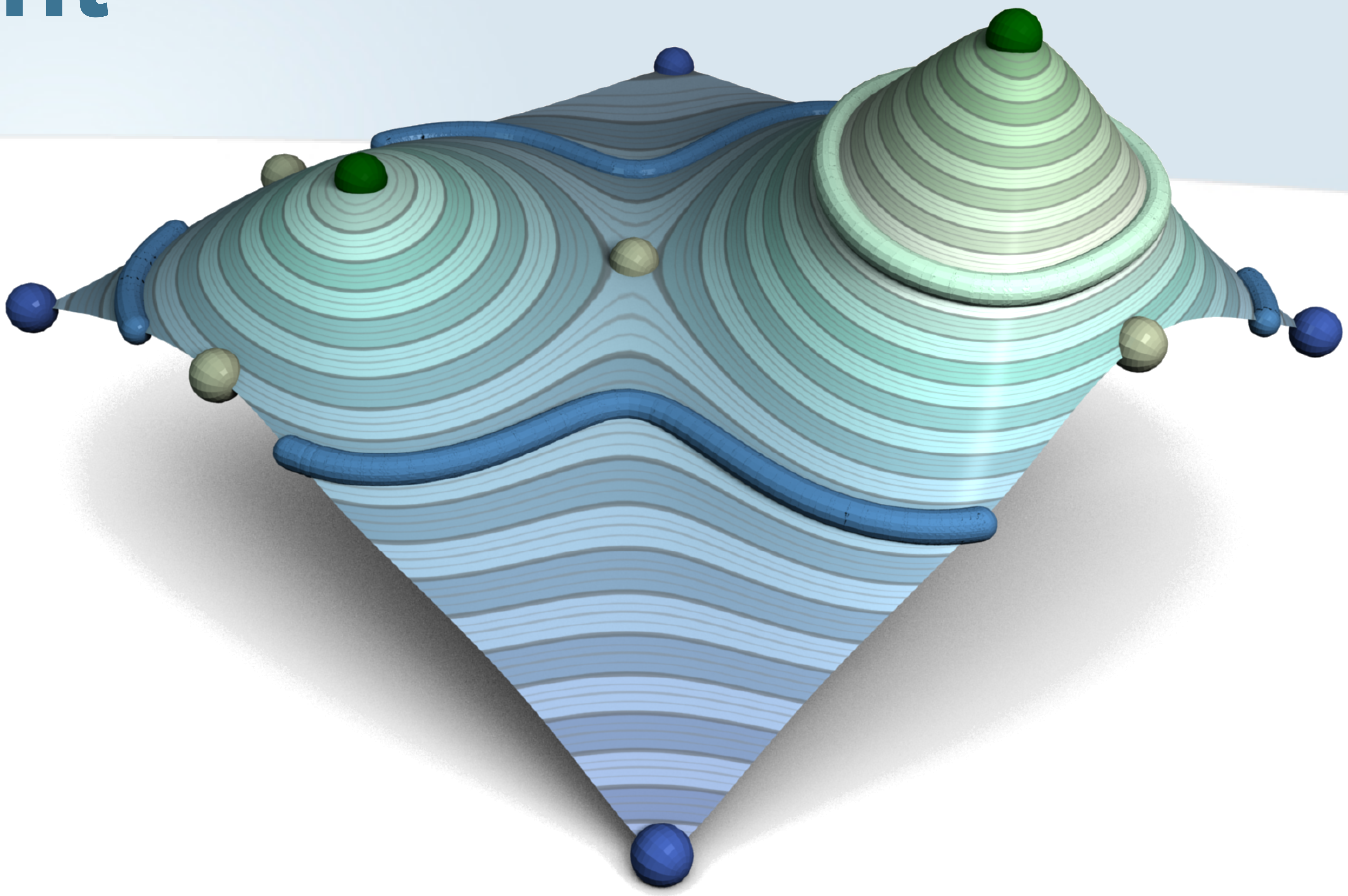
- Combinatorial identification
  - Everything else

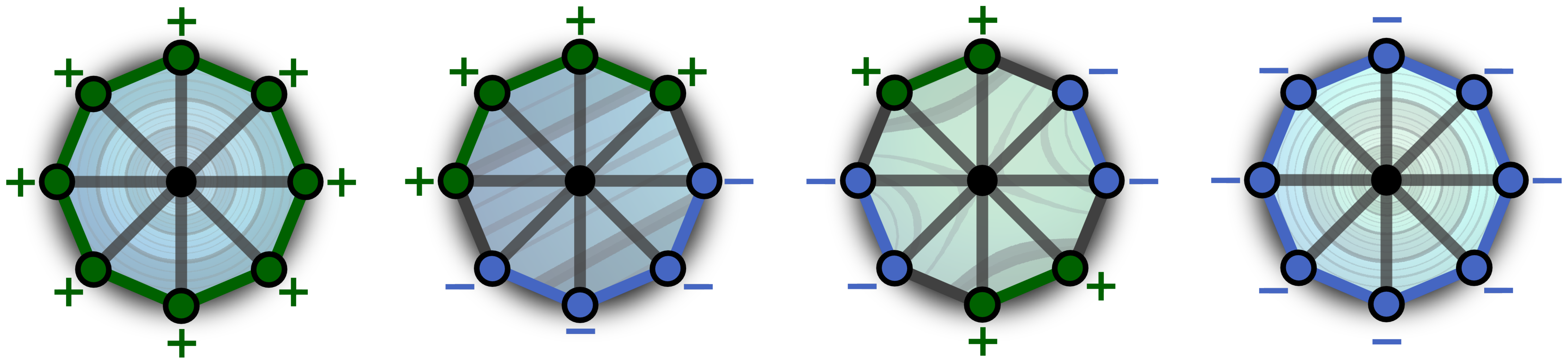




Notion of critical point

- Combinatorial identification
  - Everything else
    - Saddle

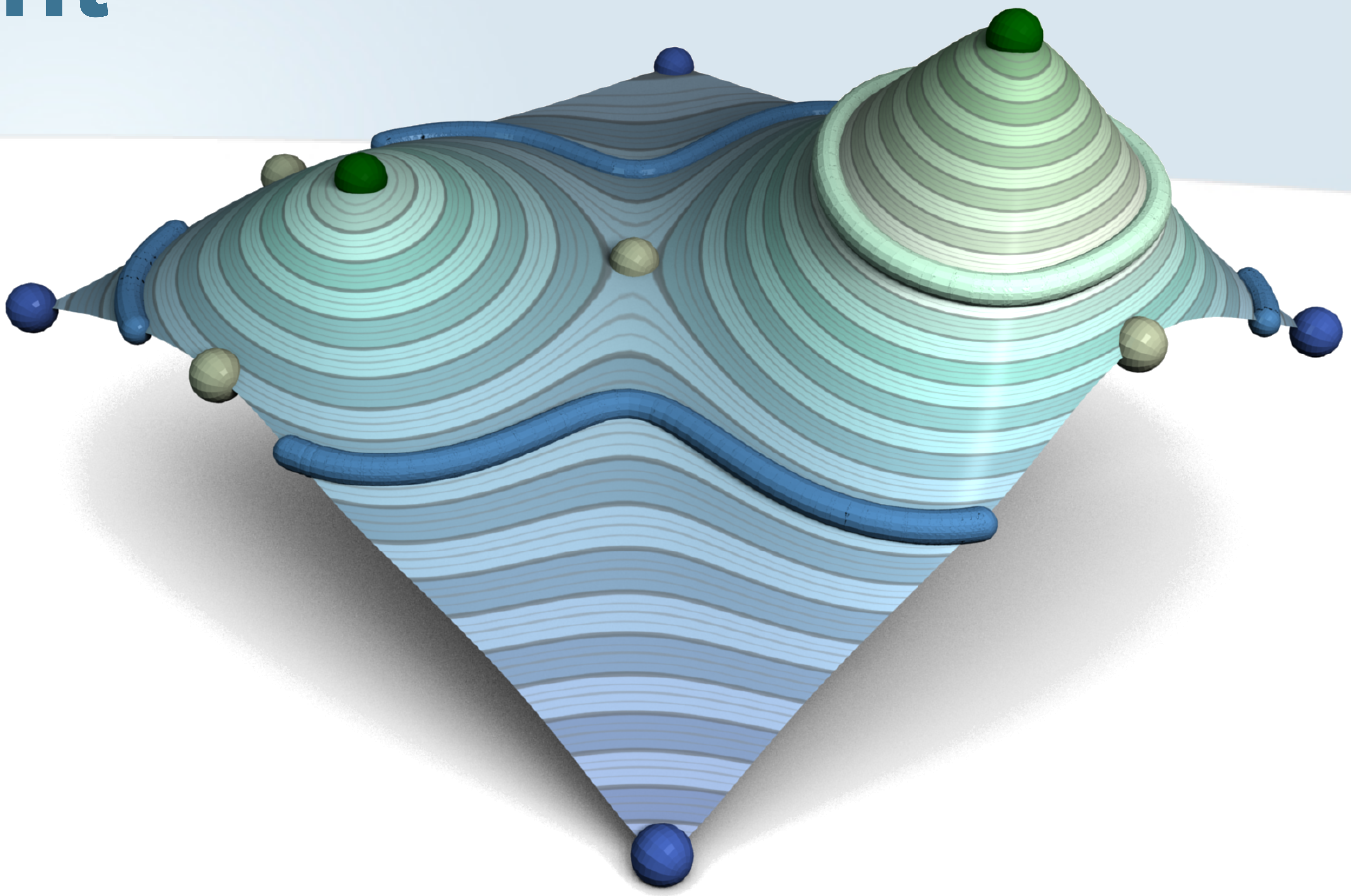


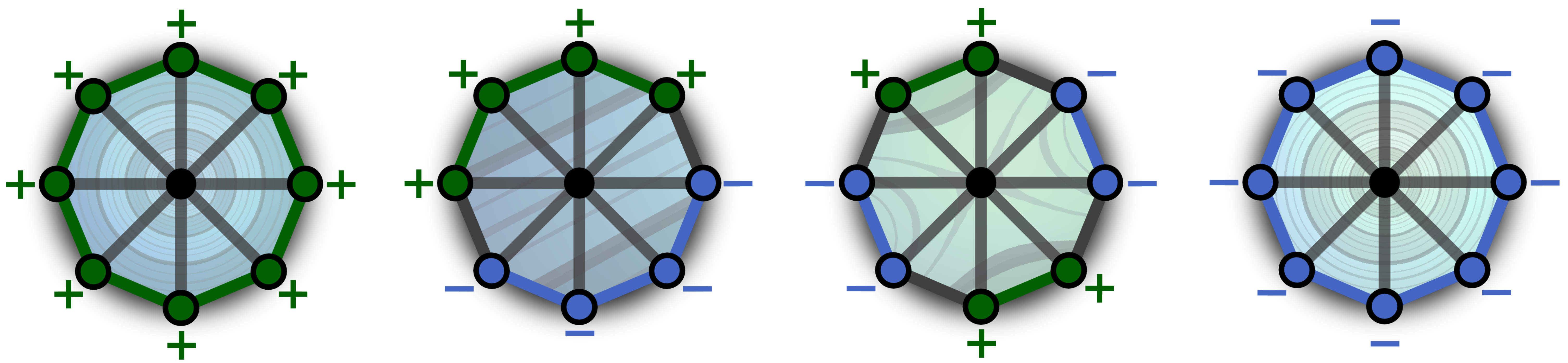


Notion of critical point

- Combinatorial identification
  - Everything else
    - Saddle

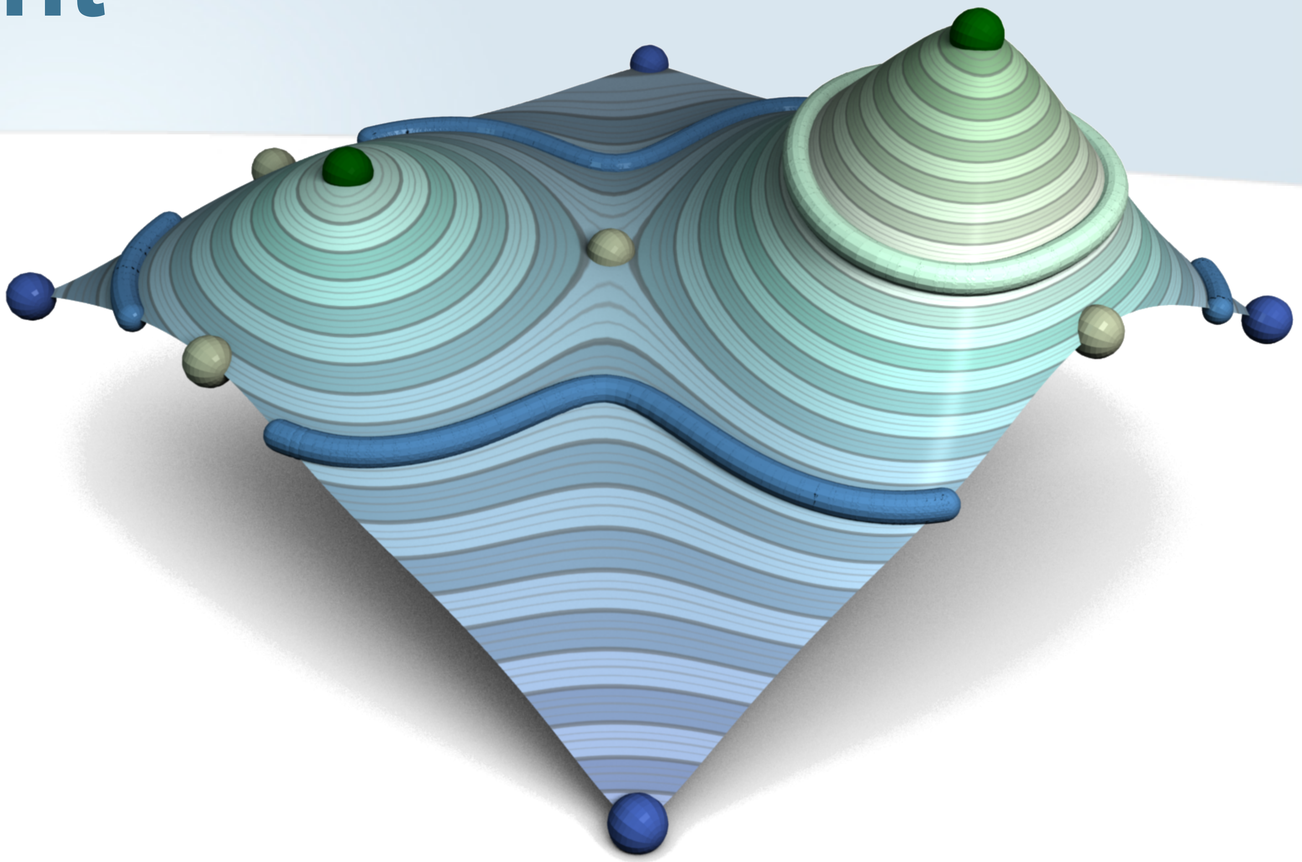
  - Works in arbitrary dimension

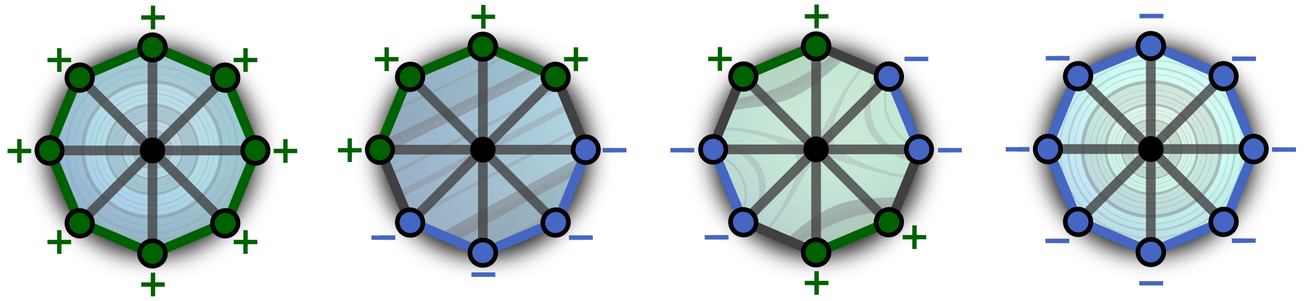




Notion of critical point

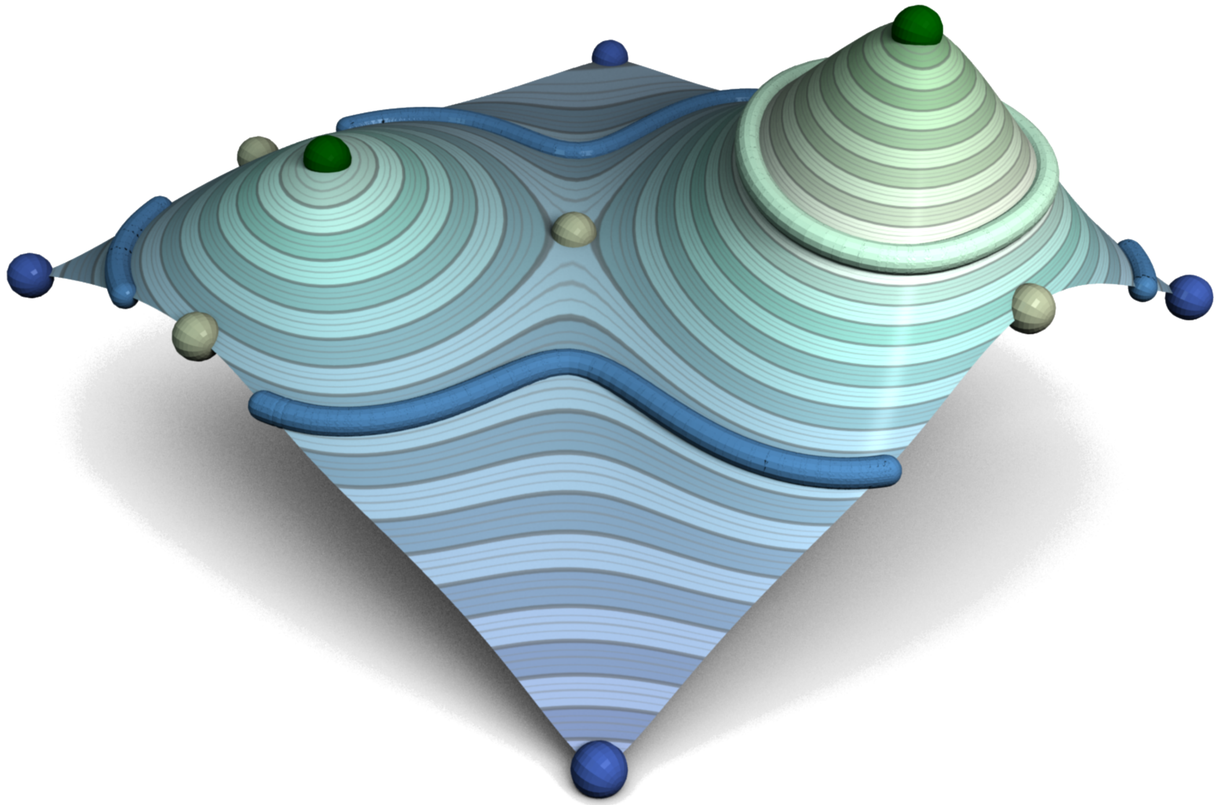
- Combinatorial identification
  - Everything else
    - Saddle
  - Works in arbitrary dimension
  - Value of a critical point
    - *Critical value*





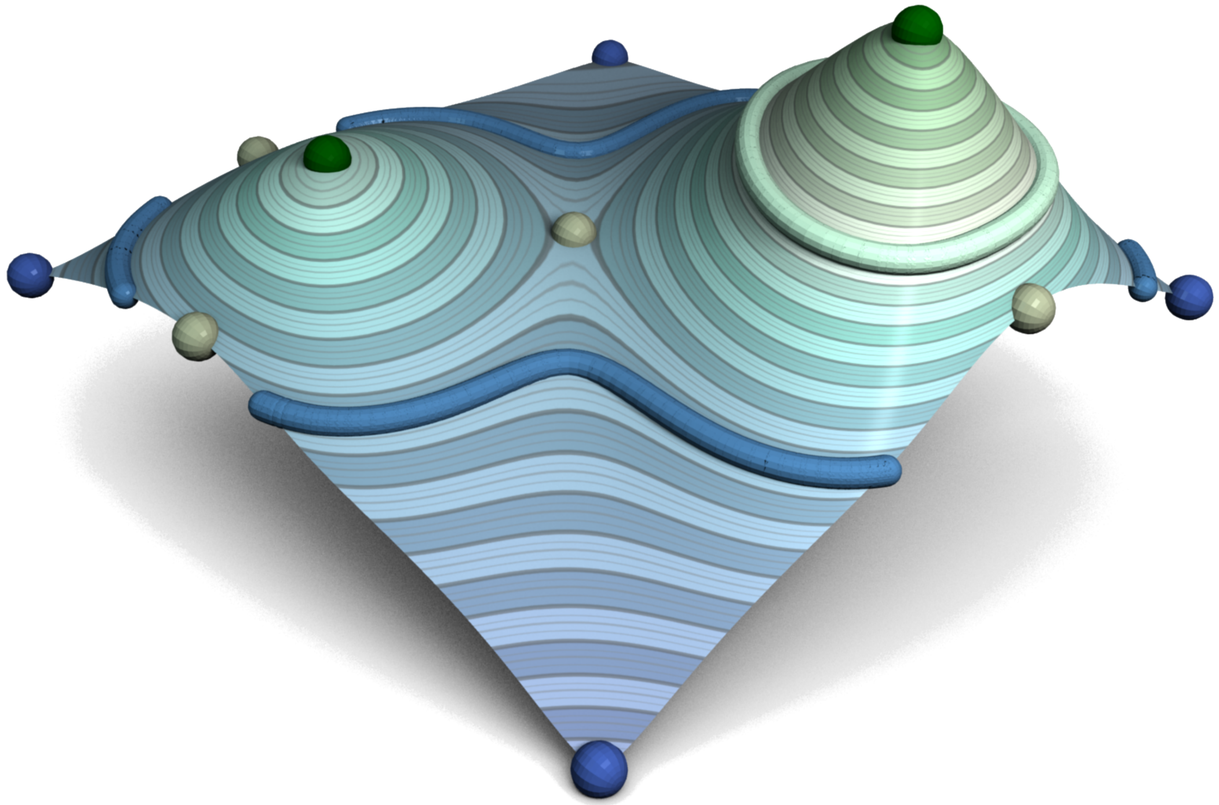
Level sets

- Given a domain $\mathcal{D}$ of dimension d
  - Level set
    - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$



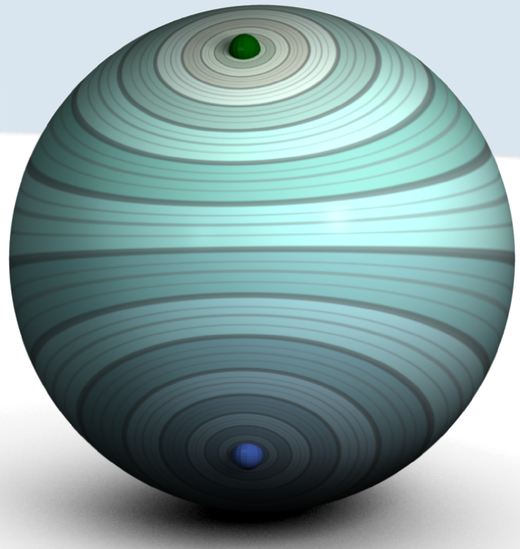
Level sets

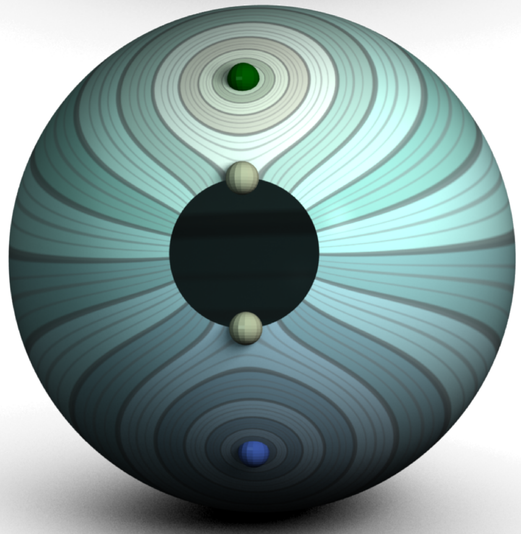
- Given a domain $\mathcal{D}$ of dimension d
  - Level set
    - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
  - If i is not a critical value
    - $f^{-1}(i)$ is a (d-1) manifold



Level sets

- Given a domain $\mathcal{D}$ of dimension d
  - Level set
    - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
  - If i is not a critical value
    - $f^{-1}(i)$ is a (d-1) manifold
  - If $\mathcal{D}$ is closed, $f^{-1}(i)$ is closed
    - Otherwise it may be open





Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

- Let i be an isovalue

Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold

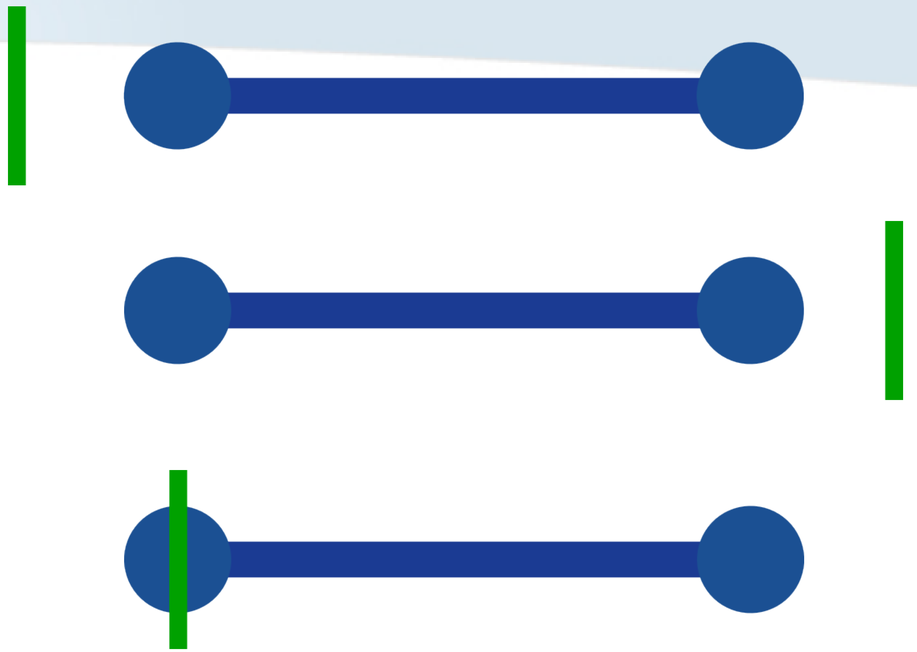
Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**

Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

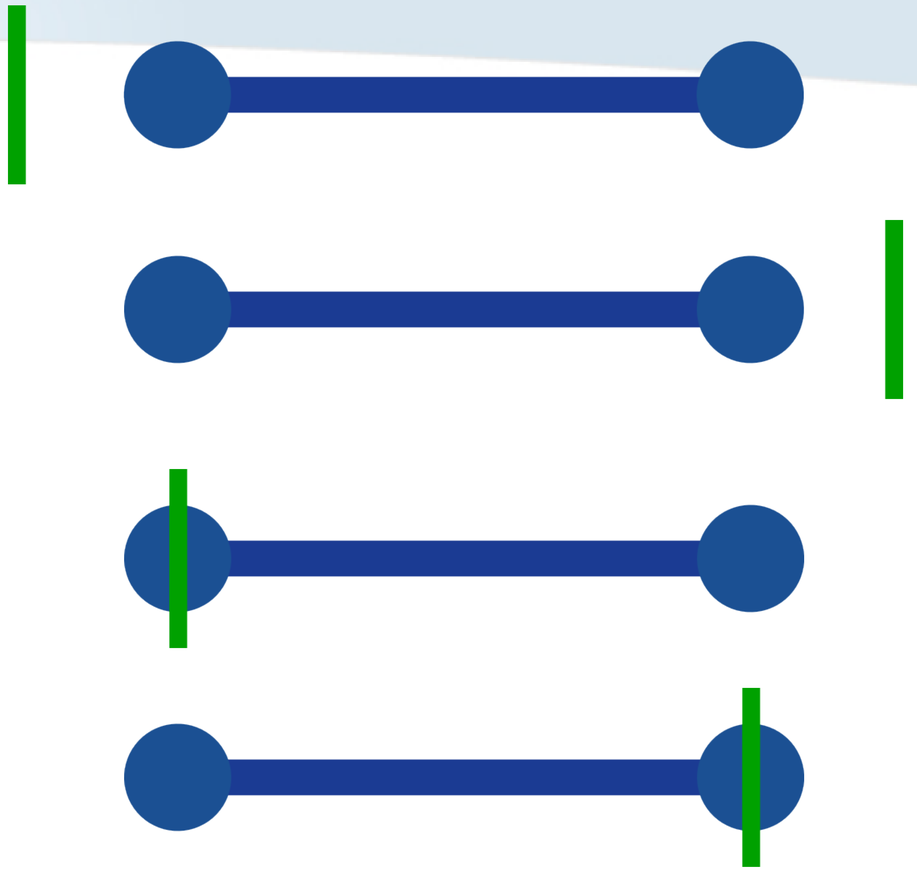


- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**

Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

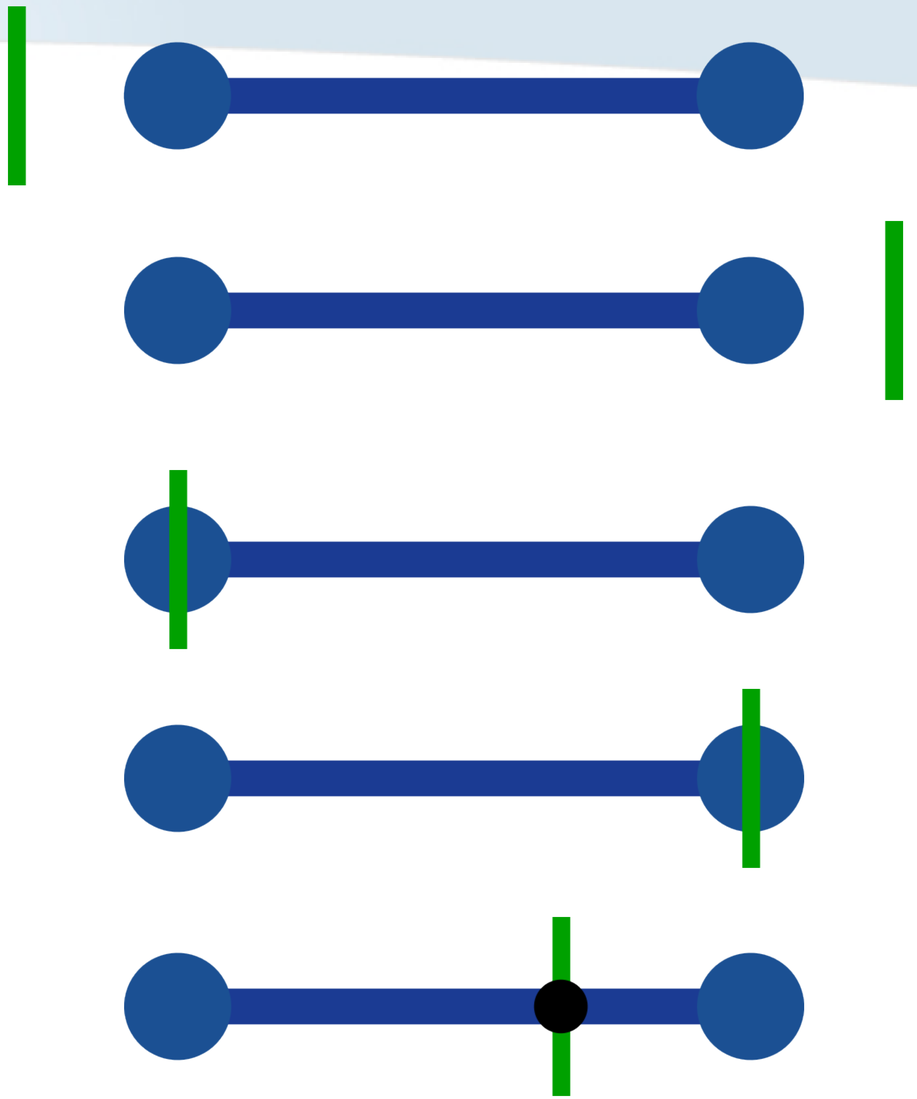
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**



Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

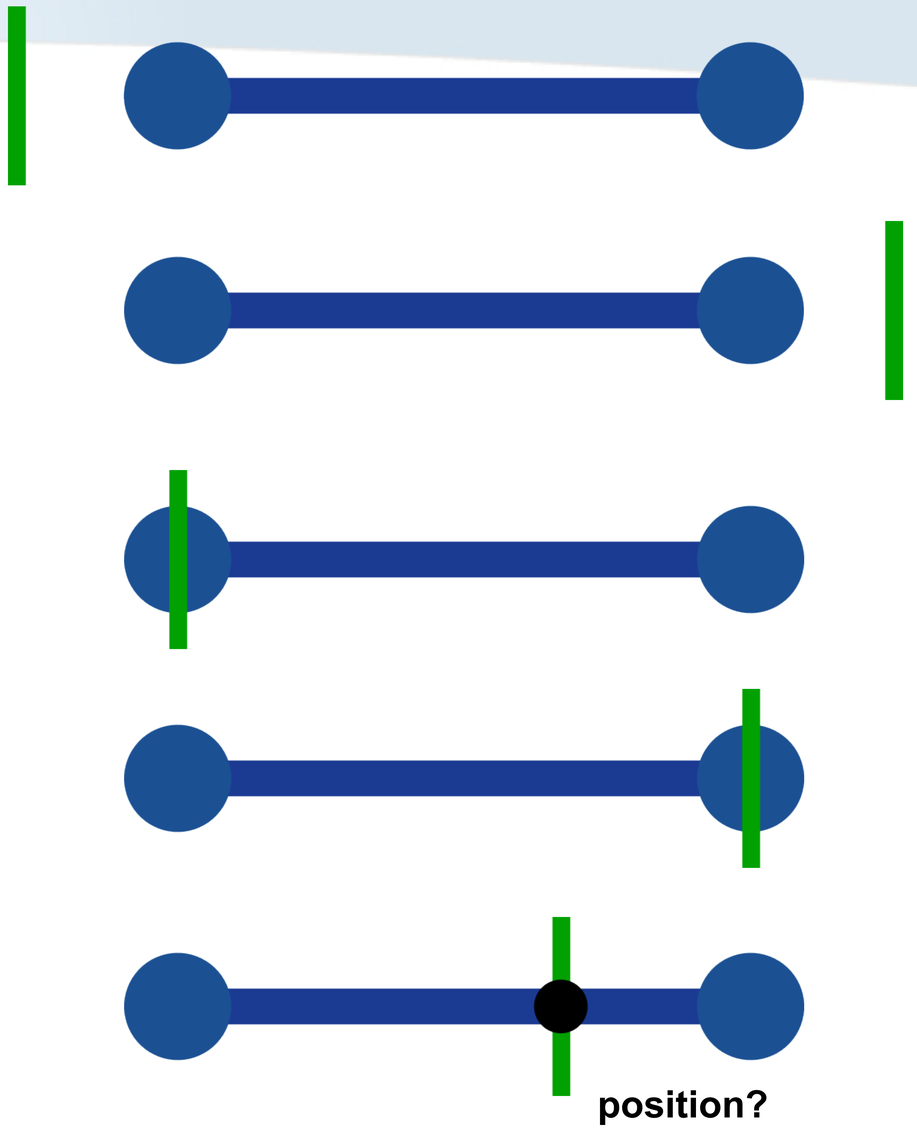
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**



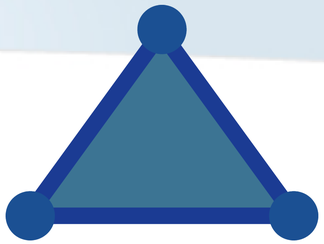
Level set in a 1-simplex

- Let $\mathcal{D}$ be a single edge
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**

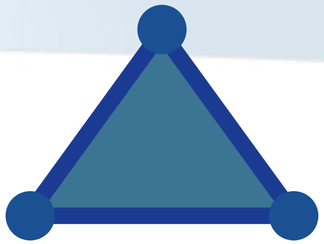


Level set in a 2-simplex



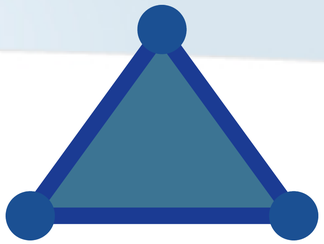
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$

Level set in a 2-simplex



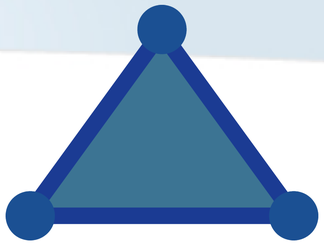
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$

Level set in a 2-simplex



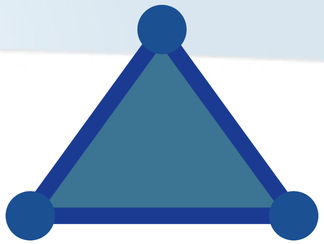
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**

Level set in a 2-simplex



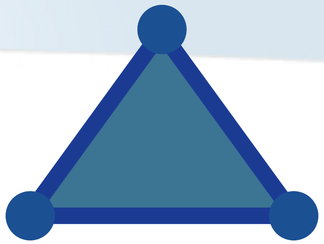
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold

Level set in a 2-simplex



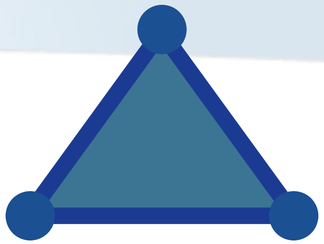
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary

Level set in a 2-simplex



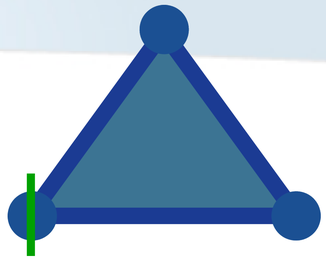
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set

Level set in a 2-simplex



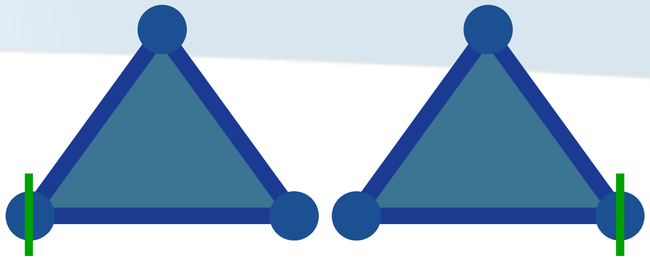
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



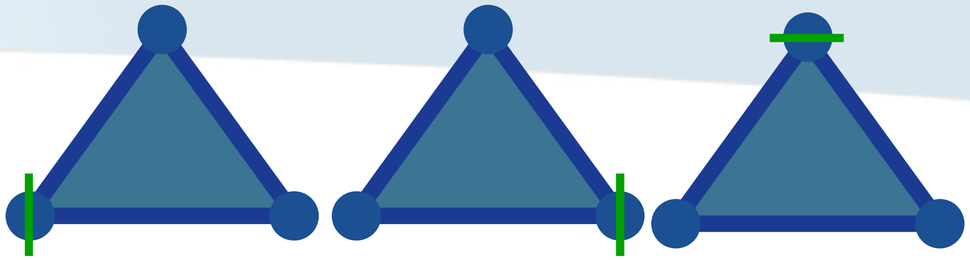
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



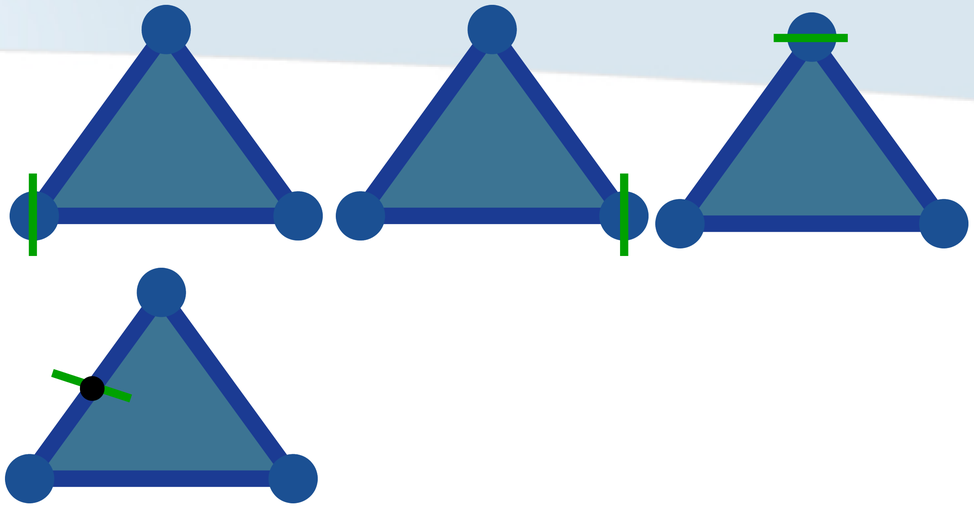
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



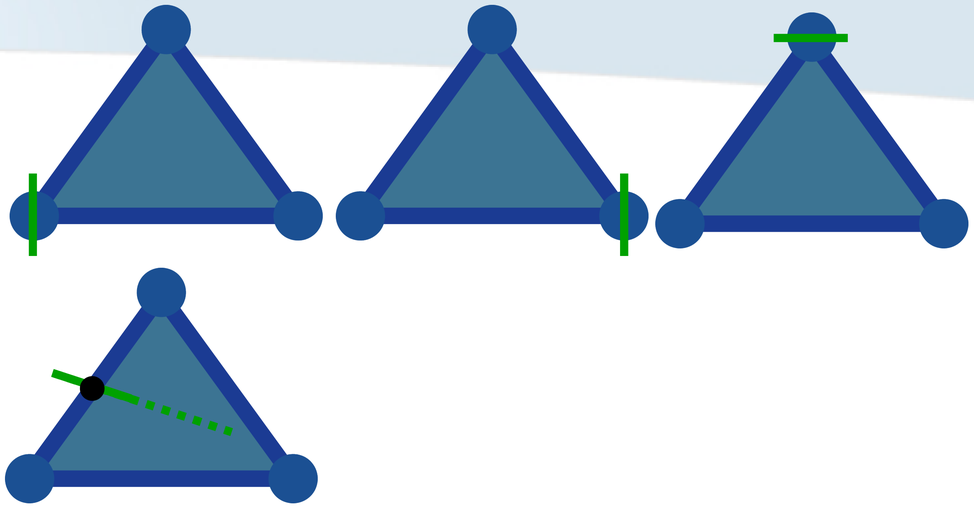
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



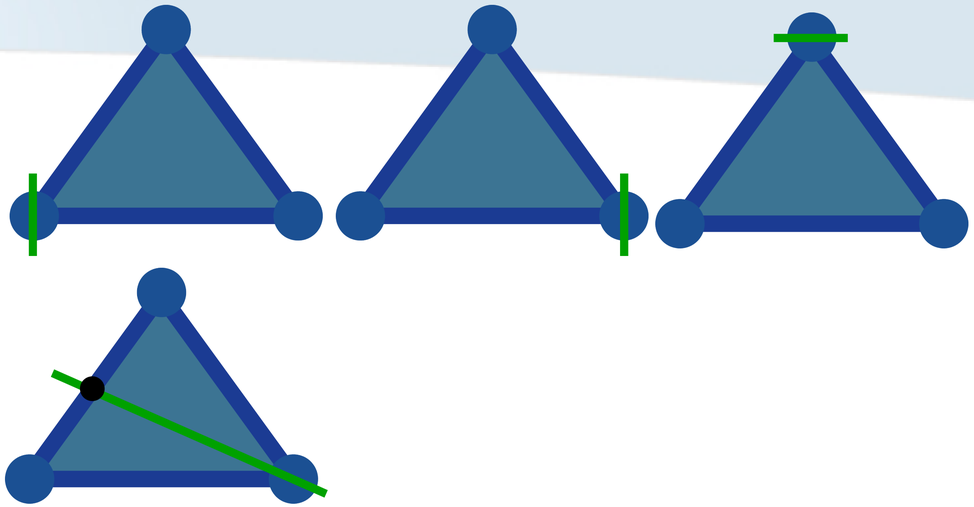
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



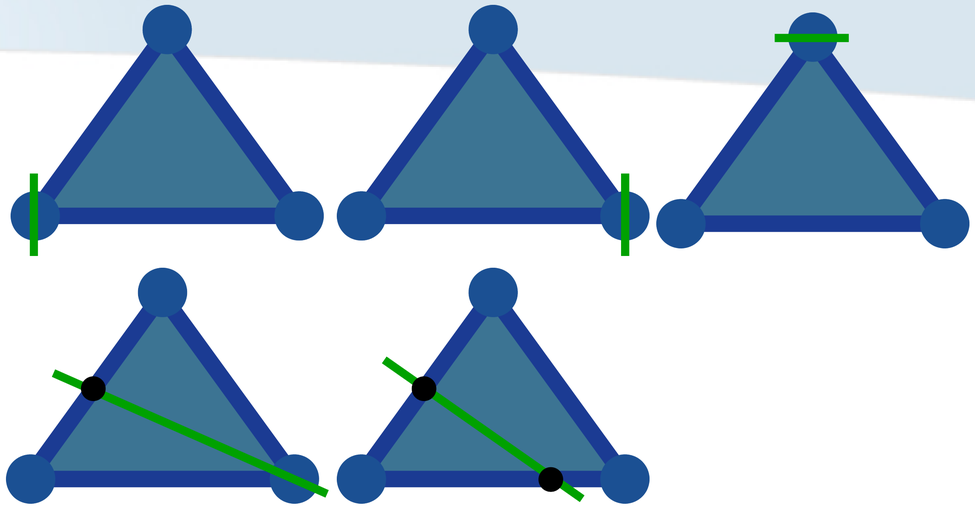
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



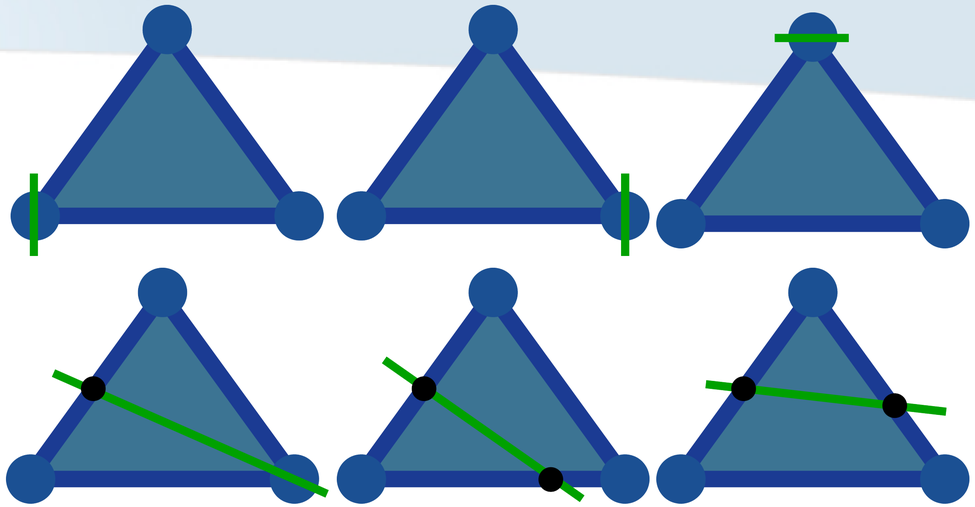
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



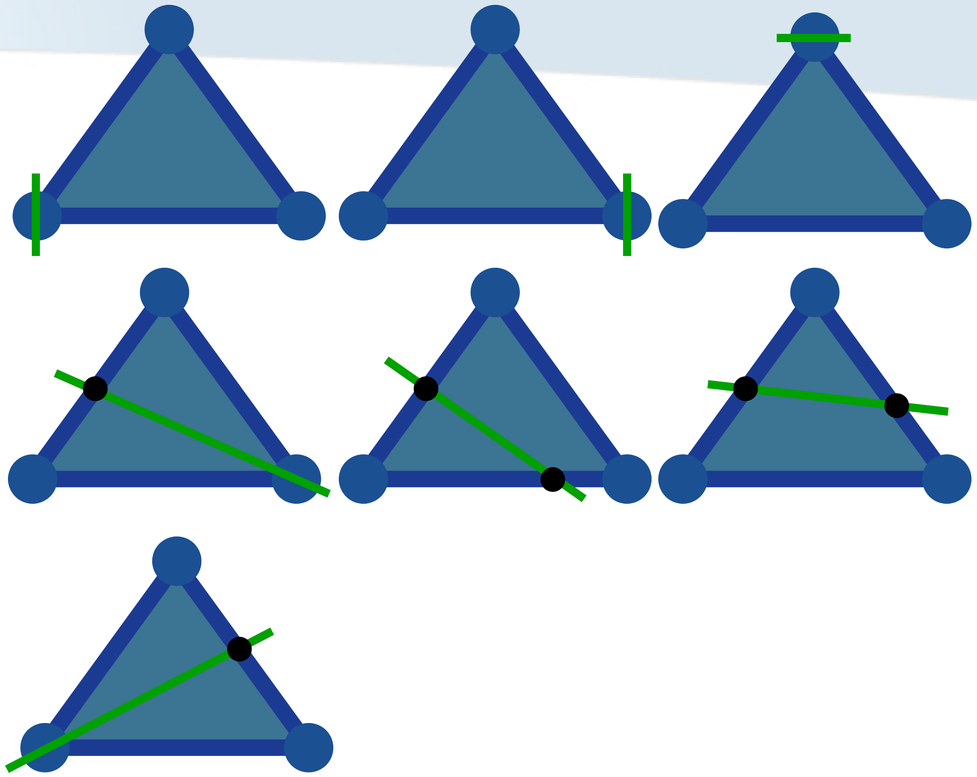
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



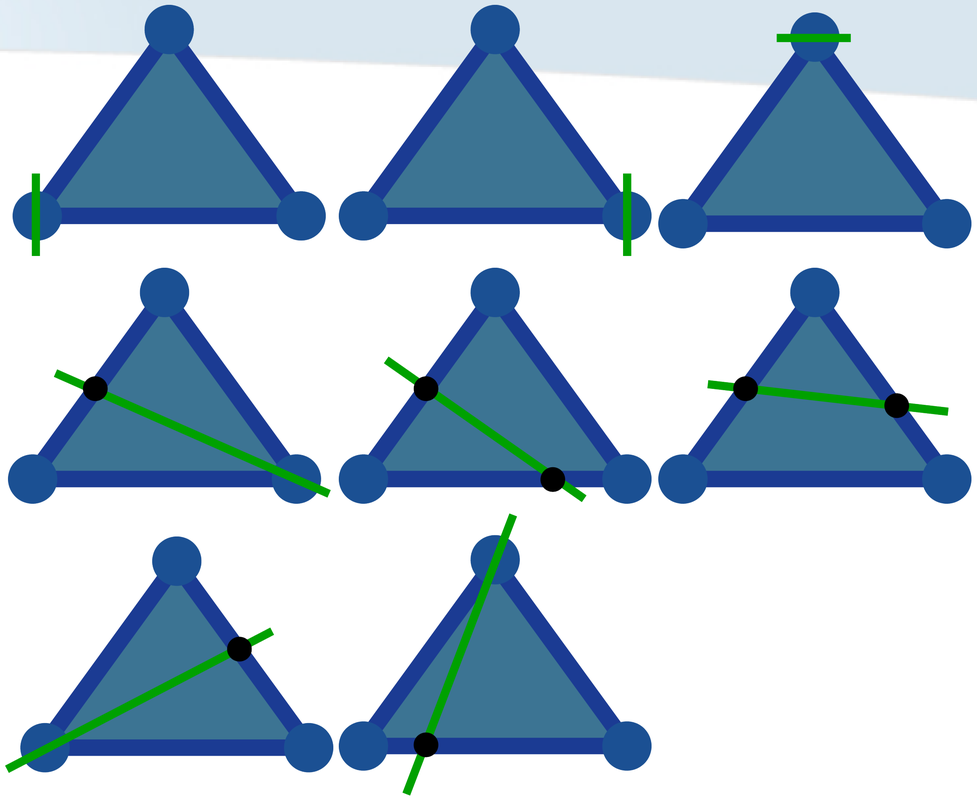
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



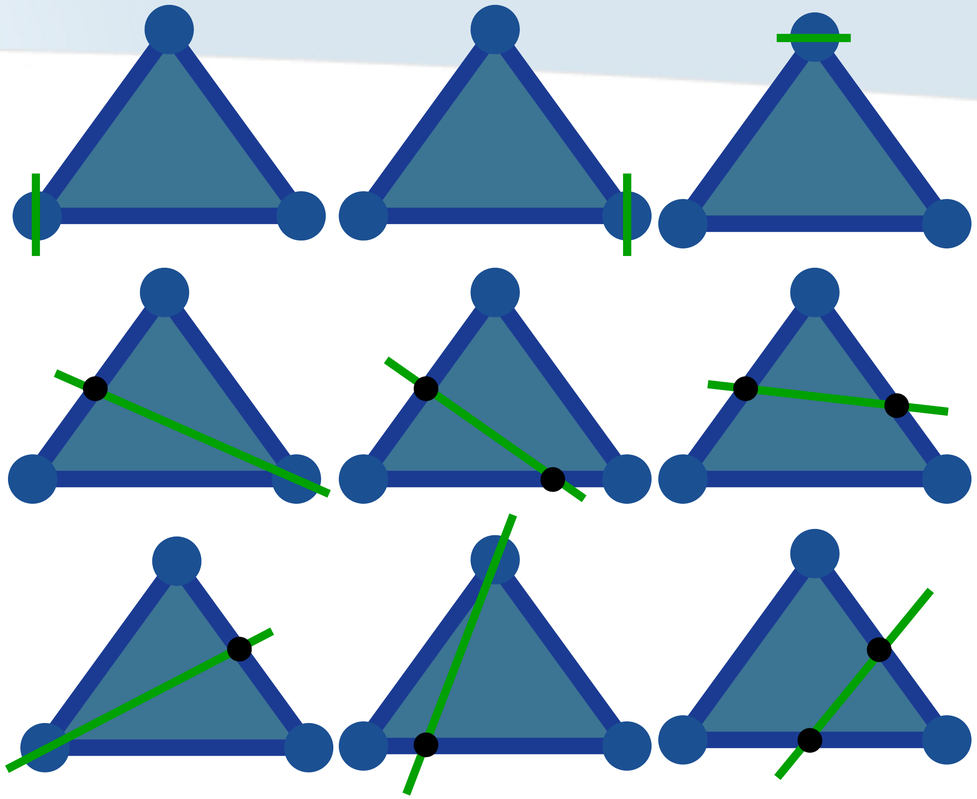
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 2-simplex



- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

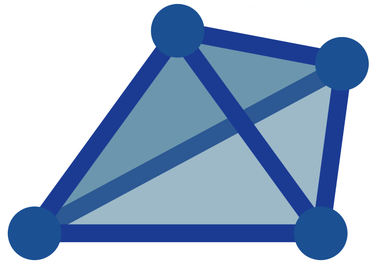
Level set in a 2-simplex



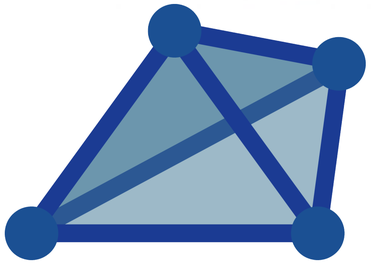
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level sets on edges: boundary of the level set
  - If edge-intersections, **9 cases only**

Level set in a 3-simplex

- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$

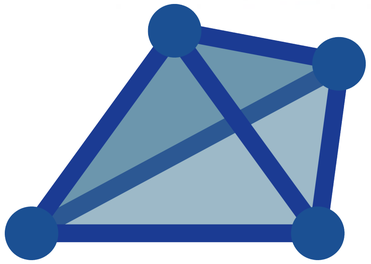


Level set in a 3-simplex



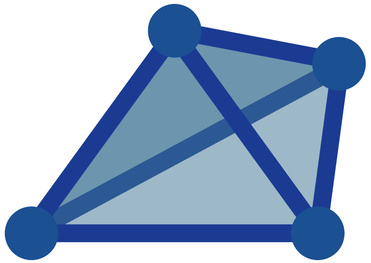
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$

Level set in a 3-simplex



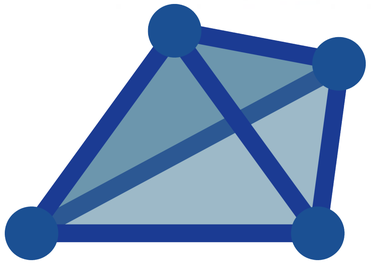
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**

Level set in a 3-simplex



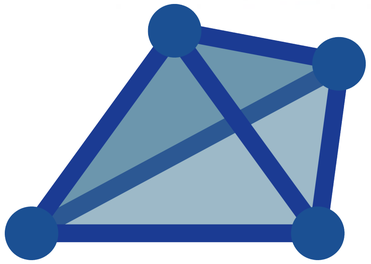
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold

Level set in a 3-simplex



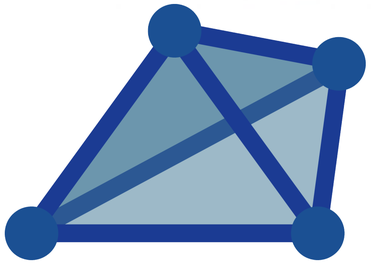
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary

Level set in a 3-simplex



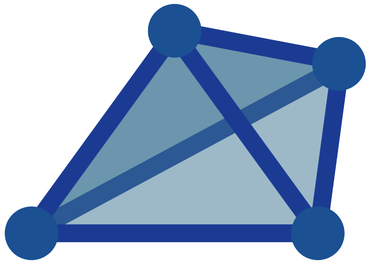
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set

Level set in a 3-simplex



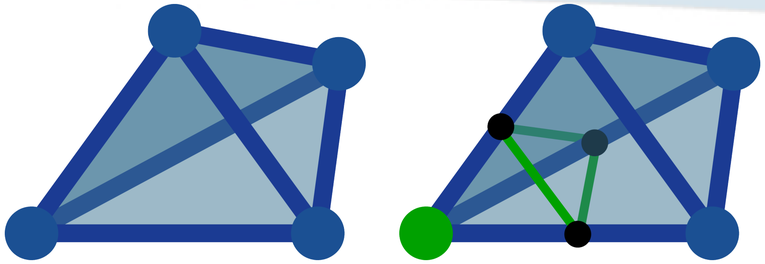
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases

Level set in a 3-simplex



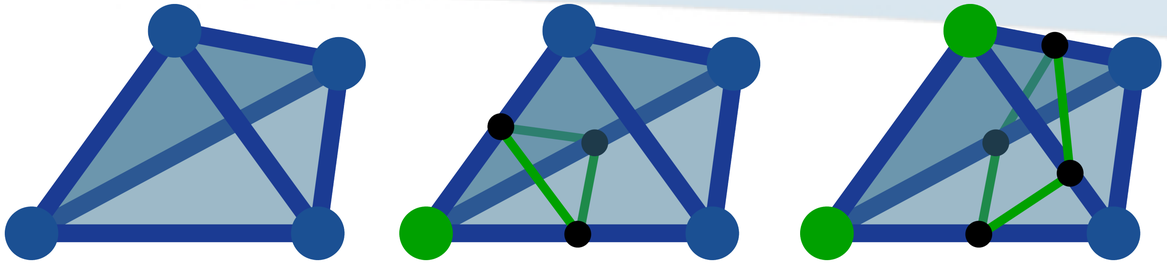
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a 3-simplex



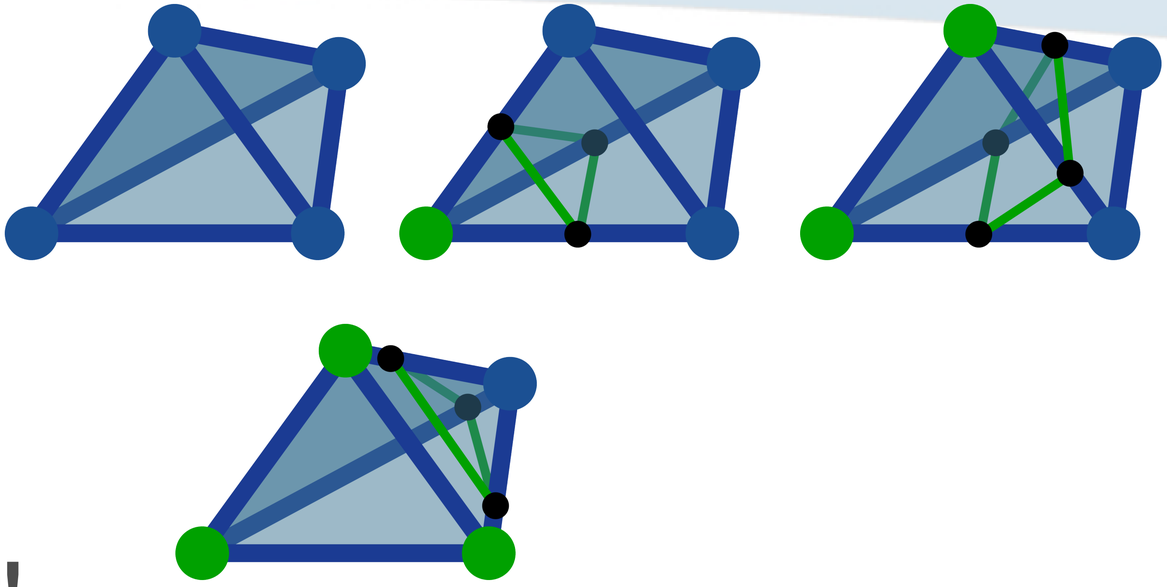
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a 3-simplex



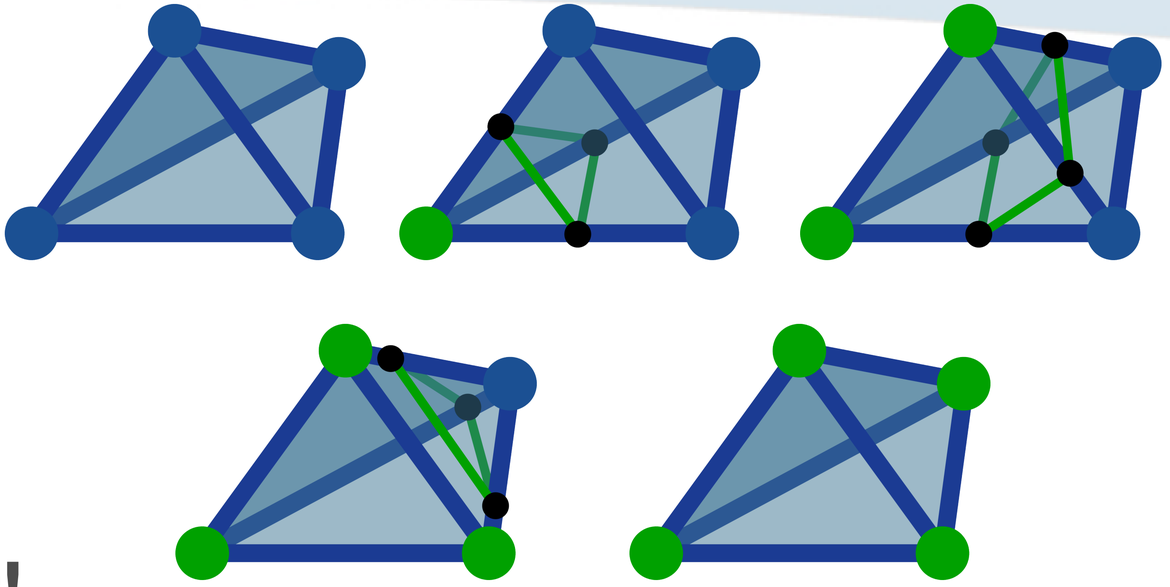
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a 3-simplex



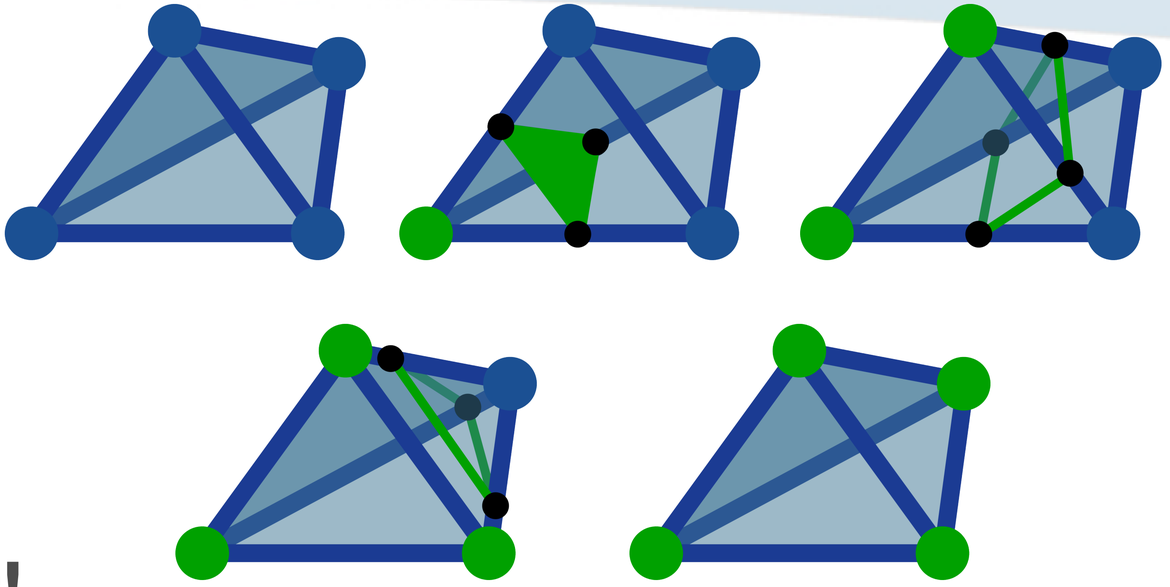
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a 3-simplex



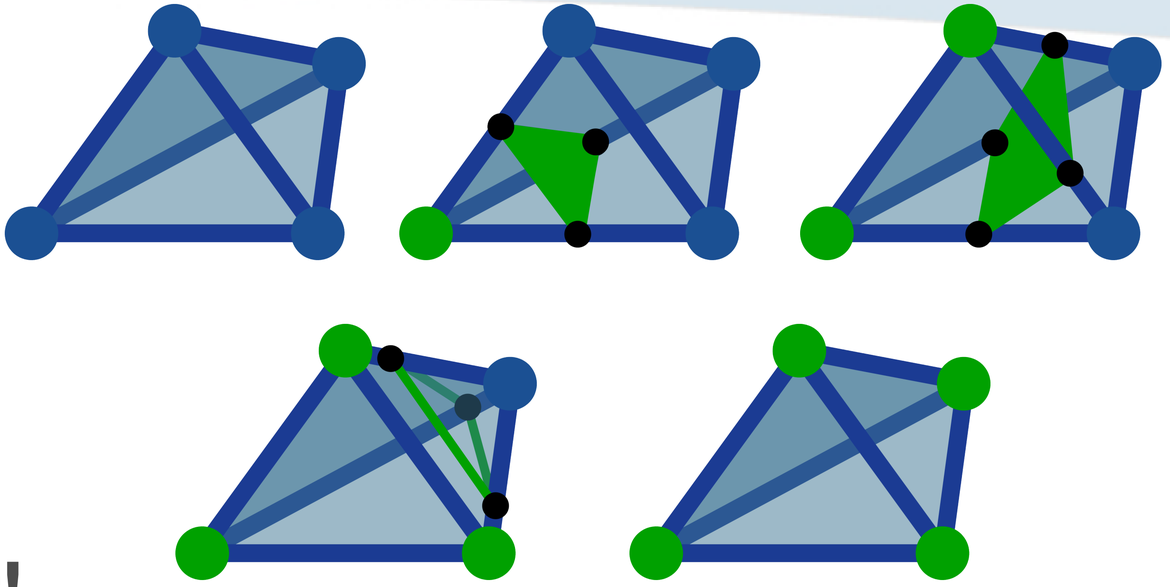
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a 3-simplex



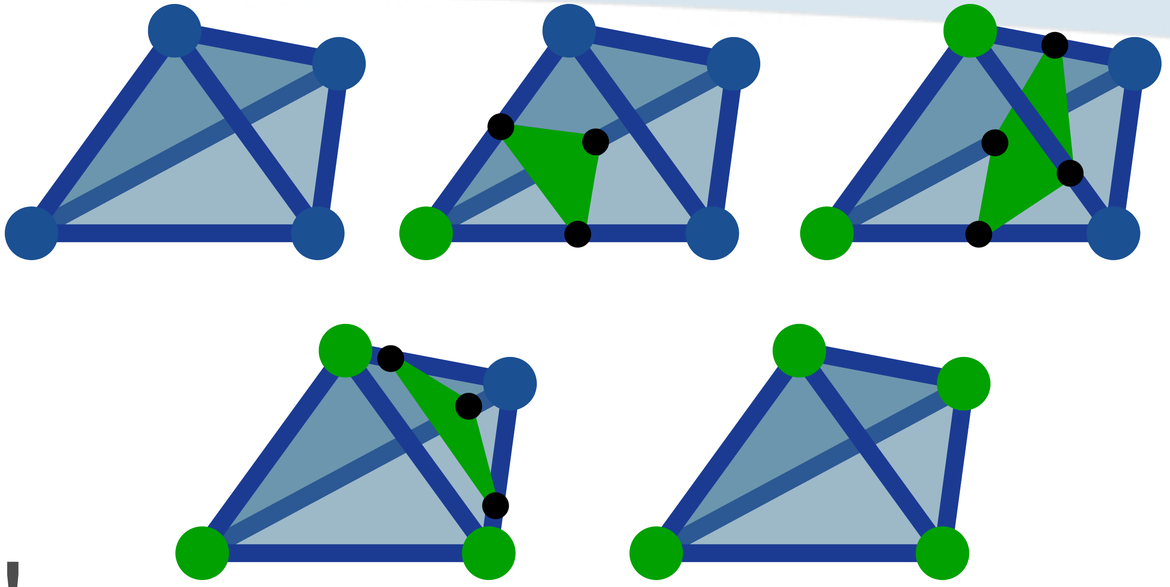
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a 3-simplex



- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a 3-simplex



- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
      - Level sets on triangles: boundary of the level set
  - Many cases, in terms of function value: only 5

Level set in a d-simplex

- Let $\mathcal{D}$ be a single d-simplex
  - $f(v_0), f(v_1), \ldots f(v_d)$

Level set in a d-simplex

- Let $\mathcal{D}$ be a single d-simplex
  - $f(v_0), f(v_1), \ldots f(v_d)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the simplex!**

Level set in a d-simplex

- Let $\mathcal{D}$ be a single d-simplex
  - $f(v_0), f(v_1), \ldots f(v_d)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the simplex!**
    - Simply connected, open, (d-1)-manifold

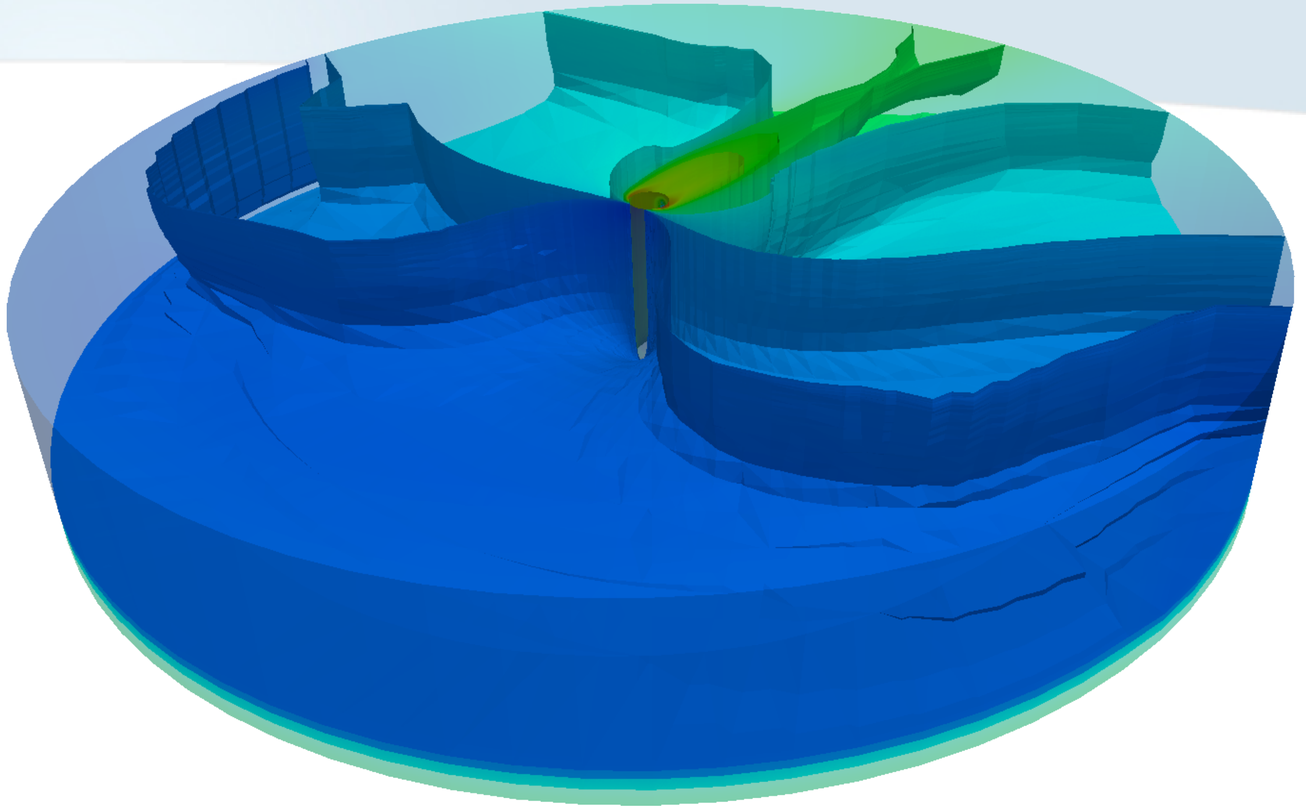
Level set in a d-simplex

- Let $\mathcal{D}$ be a single d-simplex
  - $f(v_0), f(v_1), \ldots f(v_d)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the simplex!**
    - Simply connected, open, (d-1)-manifold
    - Can be computed by only looking at the boundary
      - Level sets on (d-1)-faces: boundary of the level set

Level set in a d-simplex

- Let $\mathcal{D}$ be a single d-simplex
  - $f(v_0), f(v_1), \ldots f(v_d)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the simplex!**
    - Simply connected, open, (d-1)-manifold
    - Can be computed by only looking at the boundary
      - Level sets on (d-1)-faces: boundary of the level set
      - Recursive process

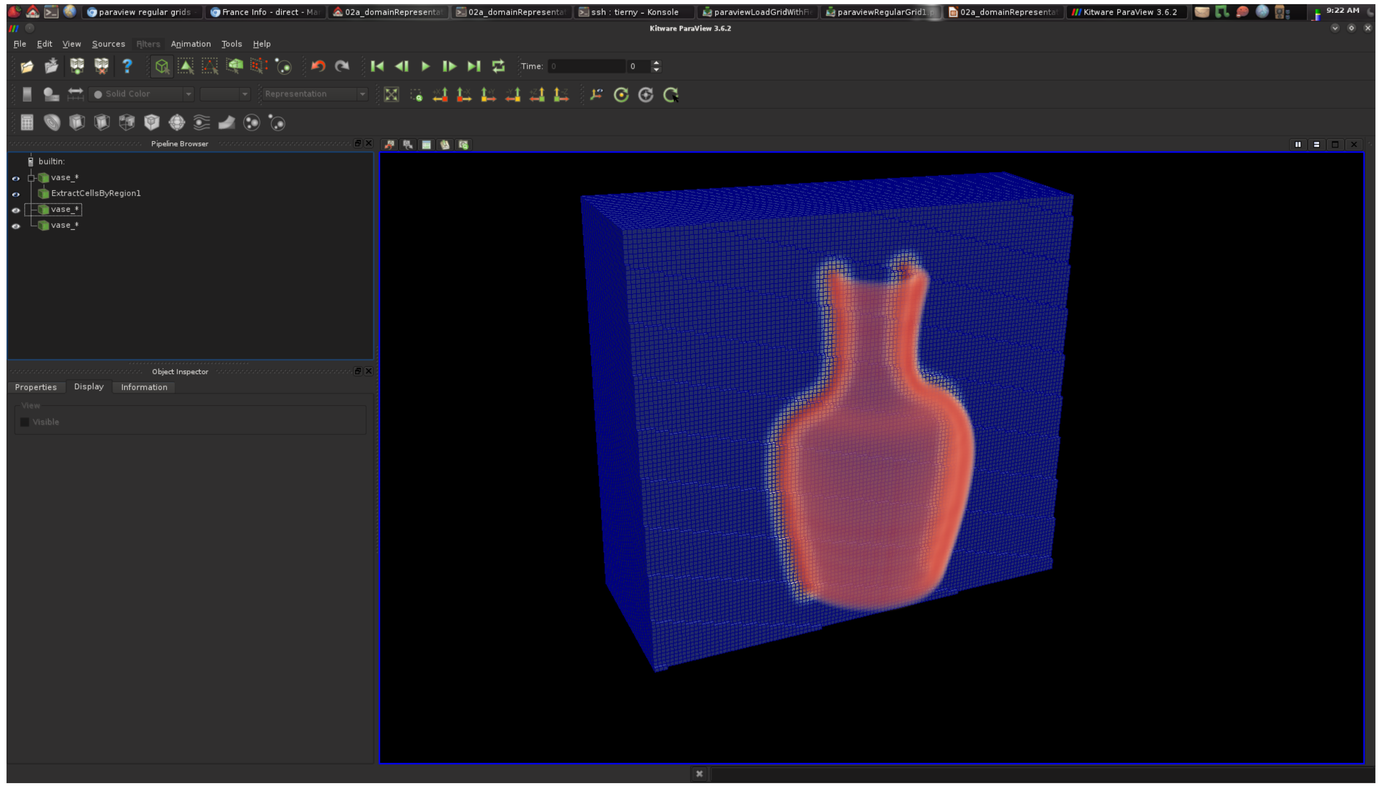
Level set extraction



- Now we know
  - How to compute a level set
  - In arbitrary dimension
  - But in only one d-simplex :(
- General algorithm
  - Flip through the list of d-simplex
  - Appy the algorithm on a per d-simplex basis
  - *“Marching Tetrahedra”*

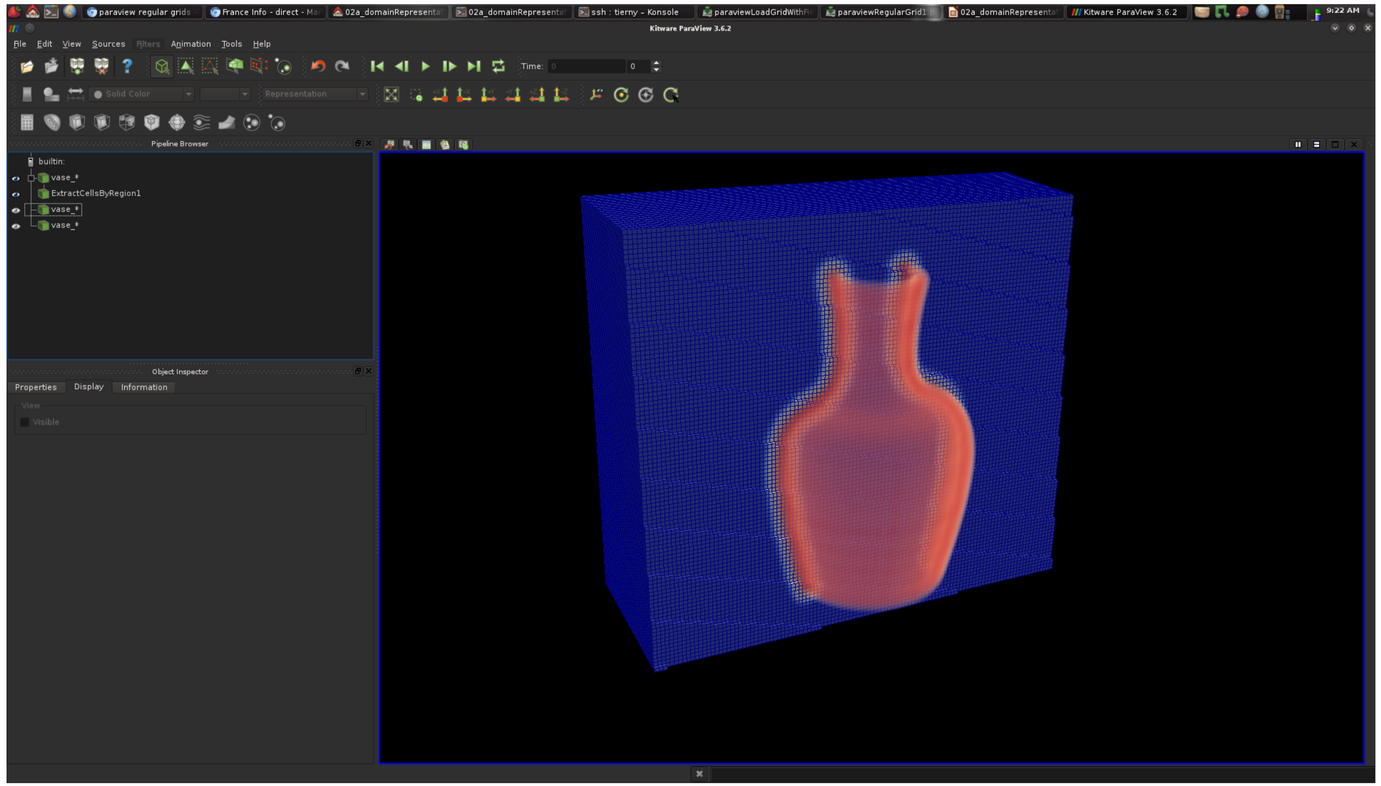
Level sets on regular grids

- What about regular grids?



Level sets on regular grids

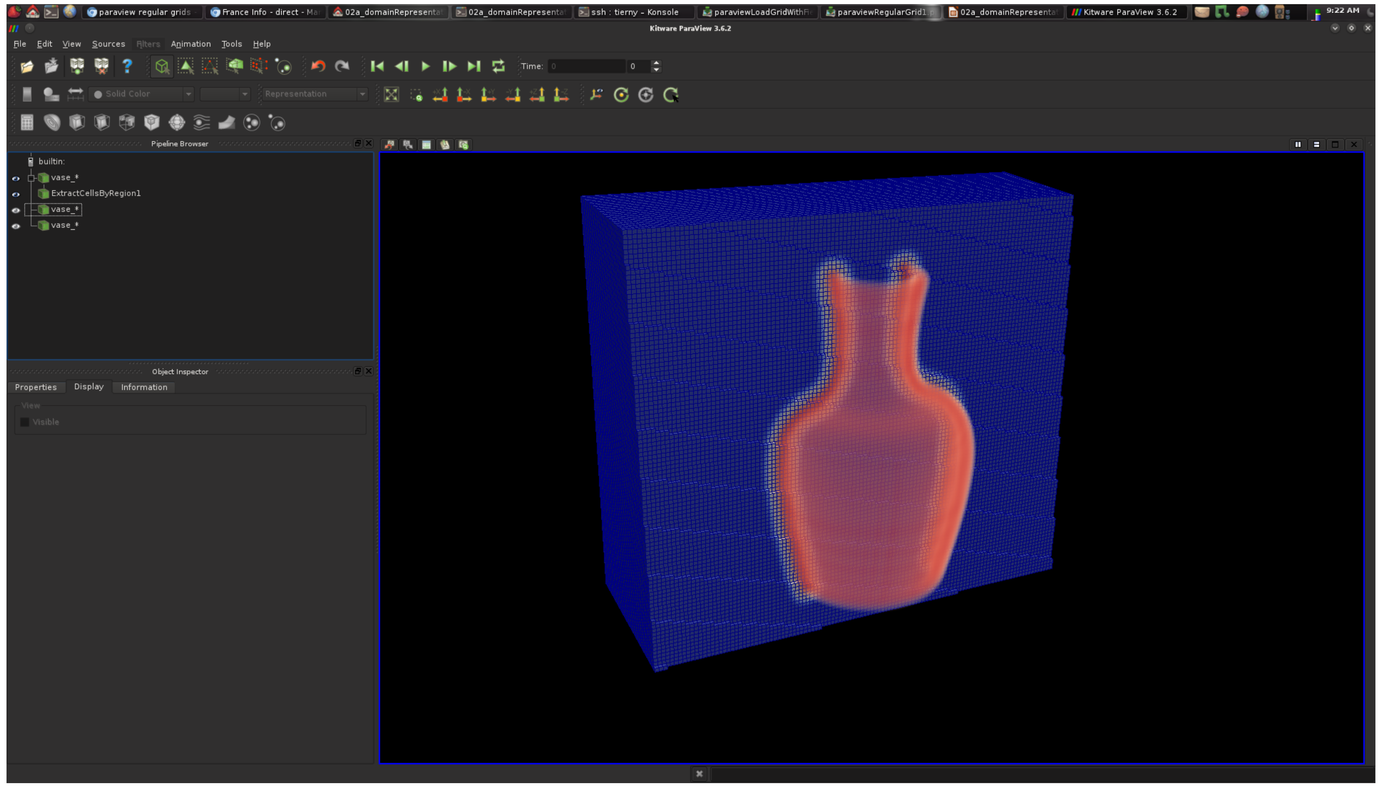
- What about regular grids?
  - In principle
    - Same rationale



Level sets on regular grids

- What about regular grids?
  - In principle
    - Same rationale

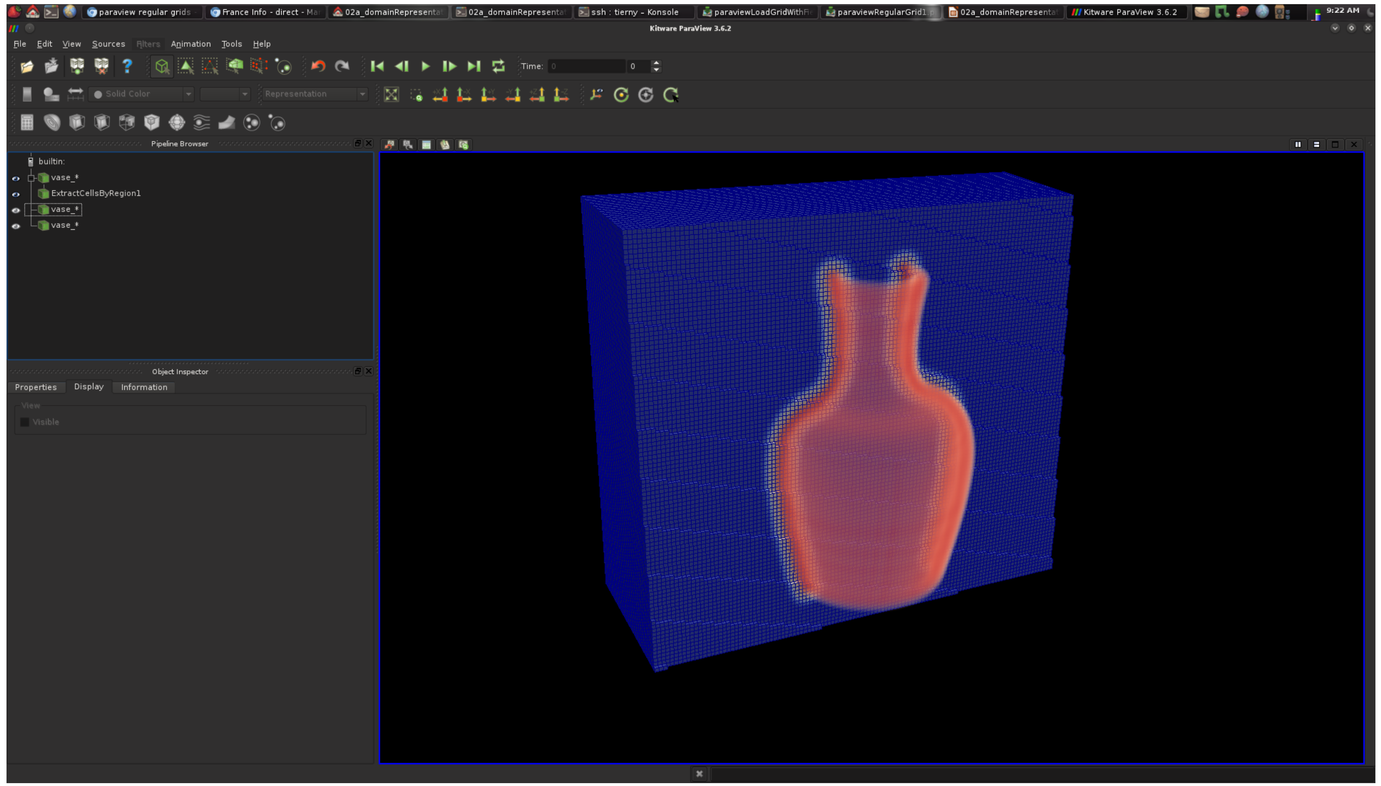
- 2D regular grids:
  - Marching squares



Level sets on regular grids

- What about regular grids?
  - In principle
    - Same rationale

- 2D regular grids:
  - Marching squares
- 3D regular grids:
  - Marching cubes



Marching squares

- Let $\mathcal{D}$ be a 2-regular grid

[Wikipedia]

Marching squares

- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant

[Wikipedia]

Marching squares

- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant

- Level set extraction
  - Loop over the unit cells

[Wikipedia]

Marching squares

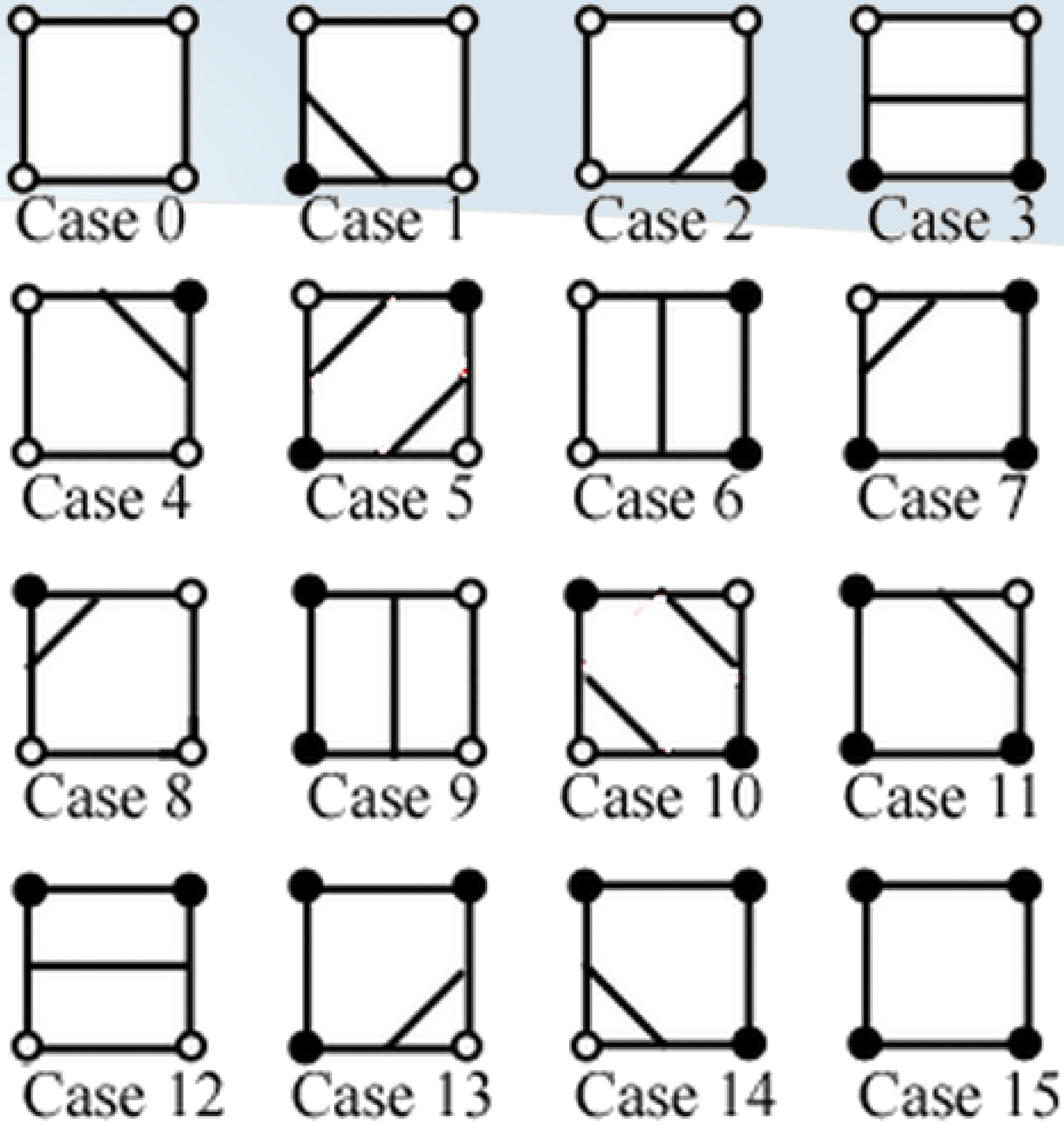
- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant

- Level set extraction
  - Loop over the unit cells
  - Cases on a 2D unit cell

[Wikipedia]

Marching squares

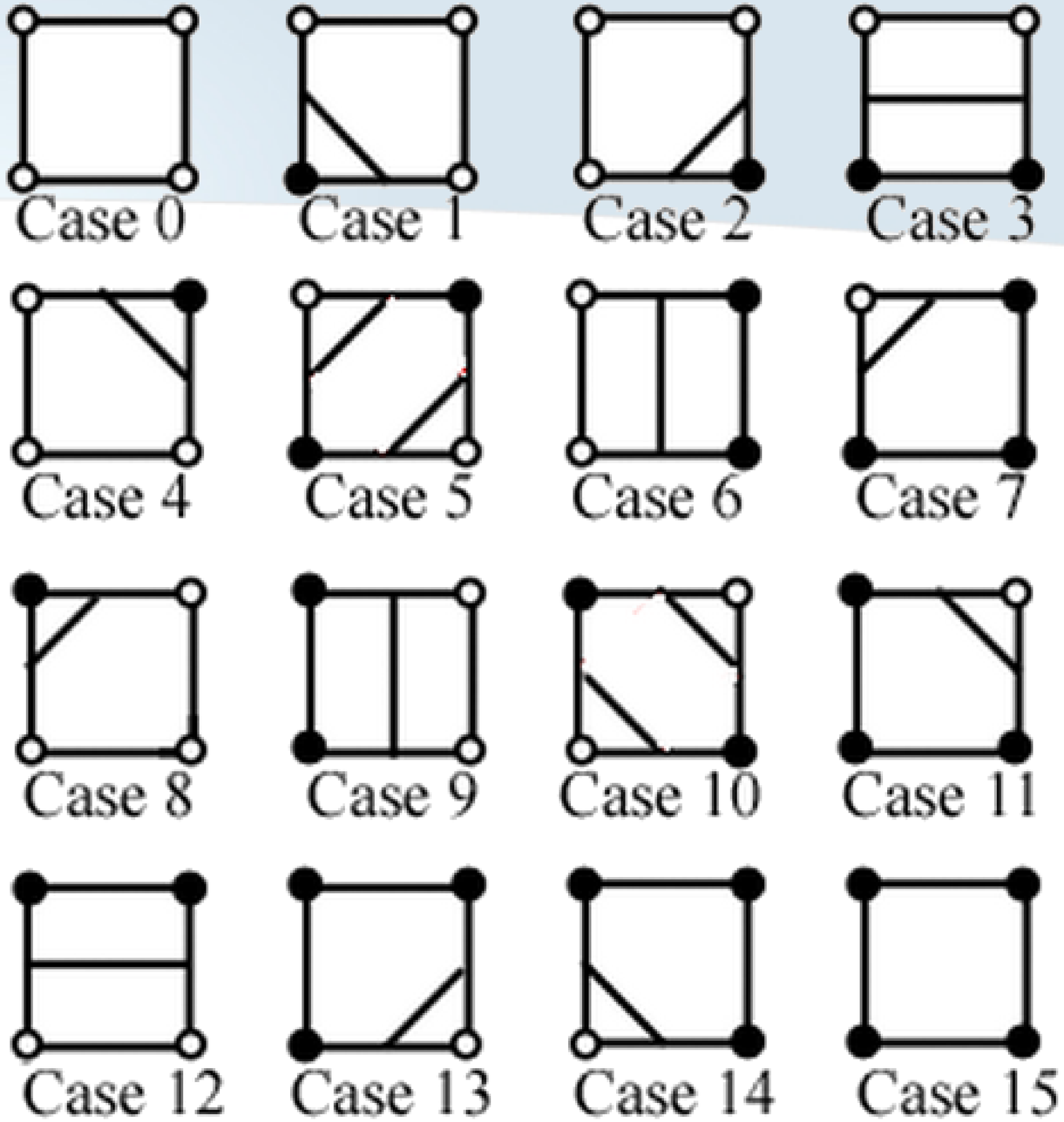
- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 2D unit cell



[Wikipedia]

Marching squares

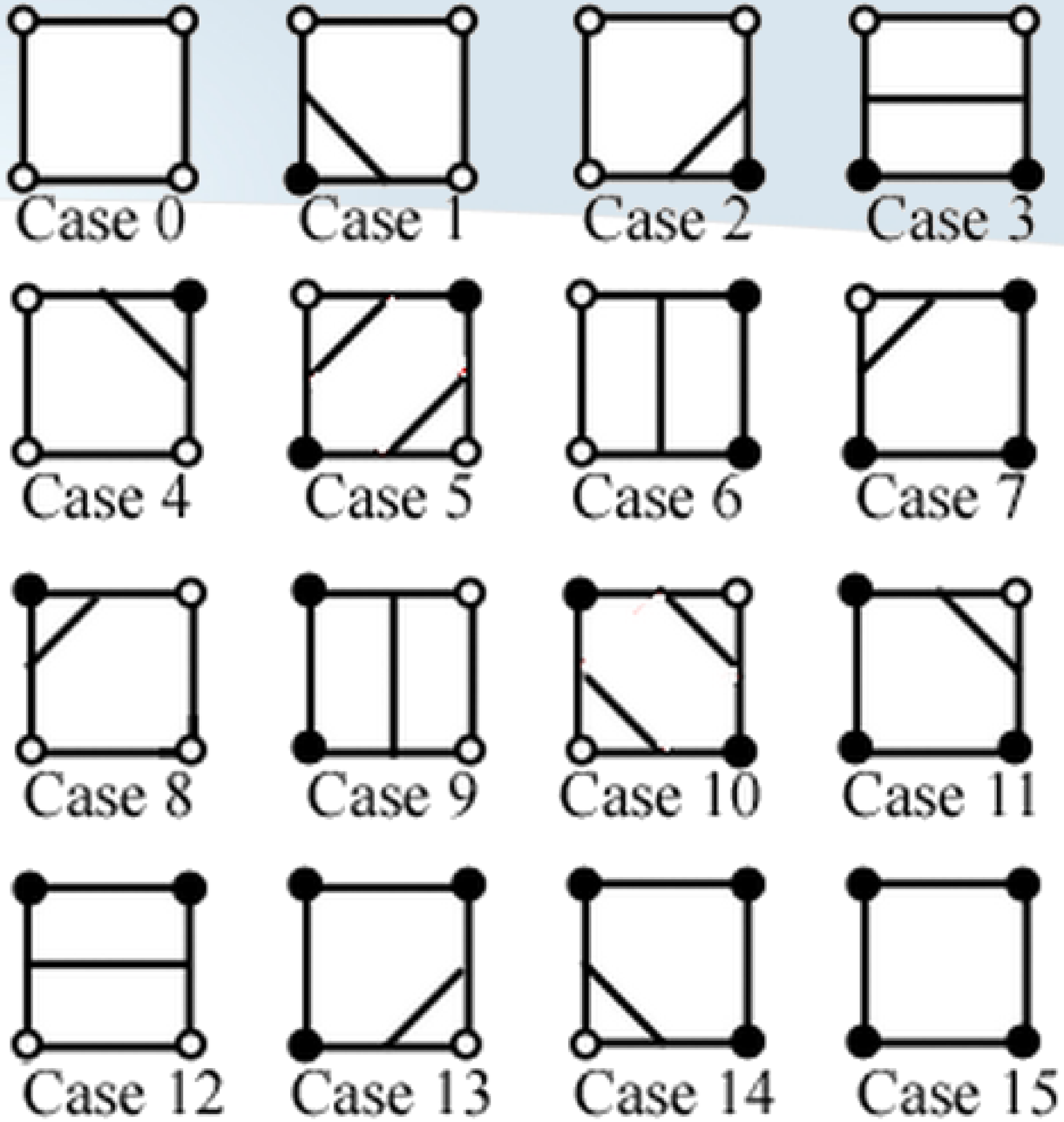
- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 2D unit cell
    - Much more than in a triangle



[Wikipedia]

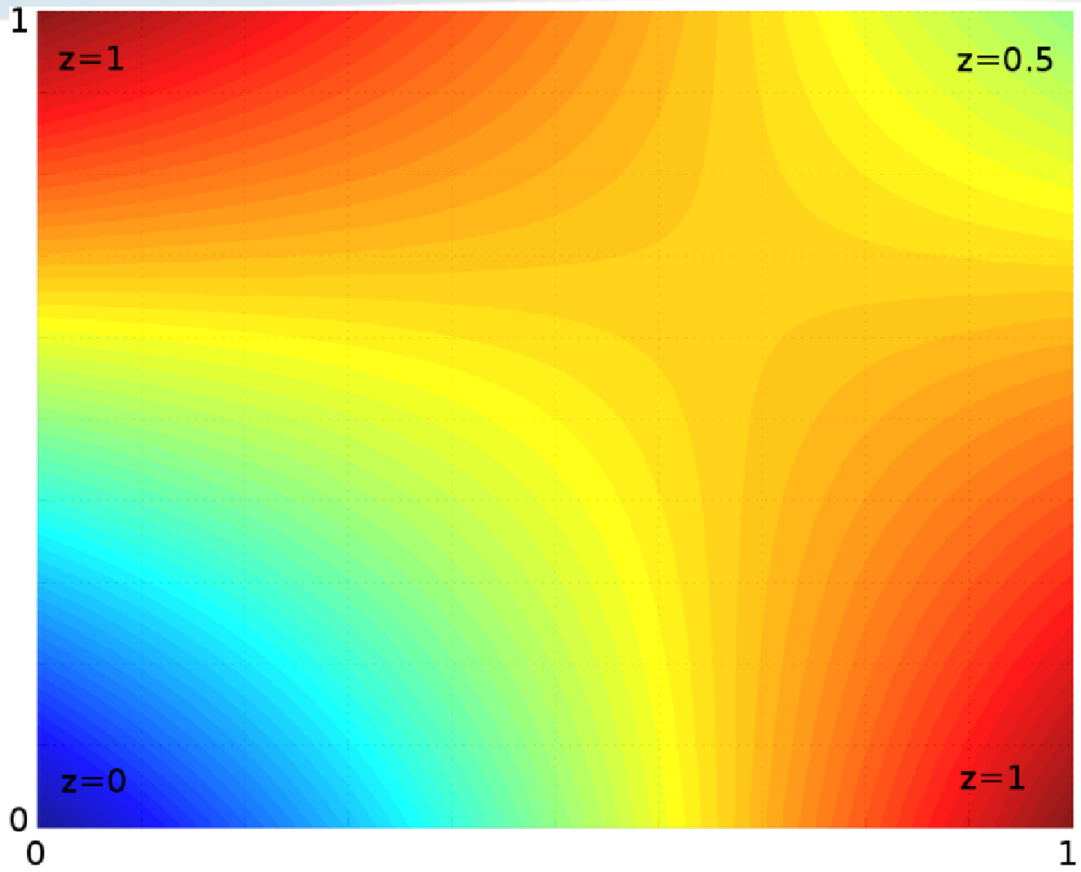
Marching squares

- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 2D unit cell
    - Much more than in a triangle
    - More than in a tet!



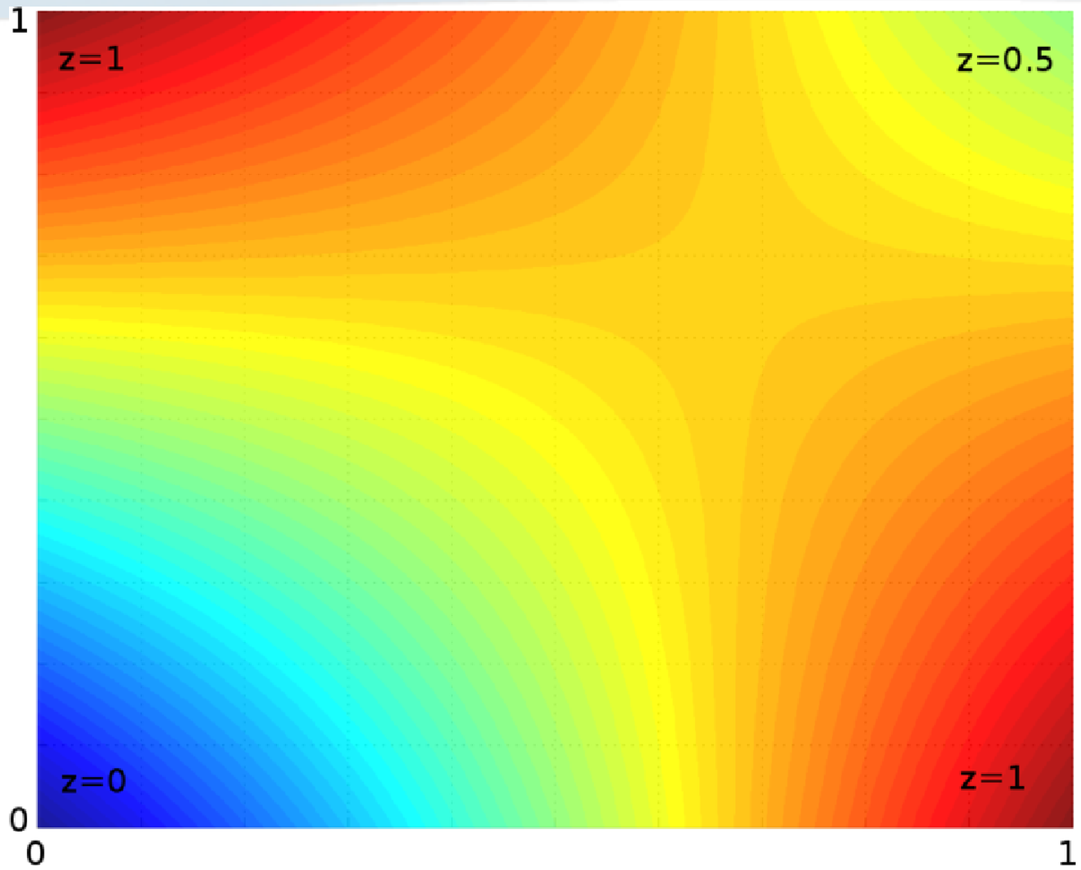
[Wikipedia]

Issues of marching squares



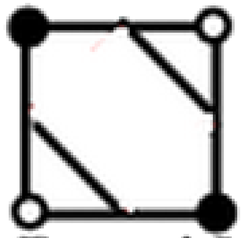
Issues of marching squares

$f^{-1}(0.95)$

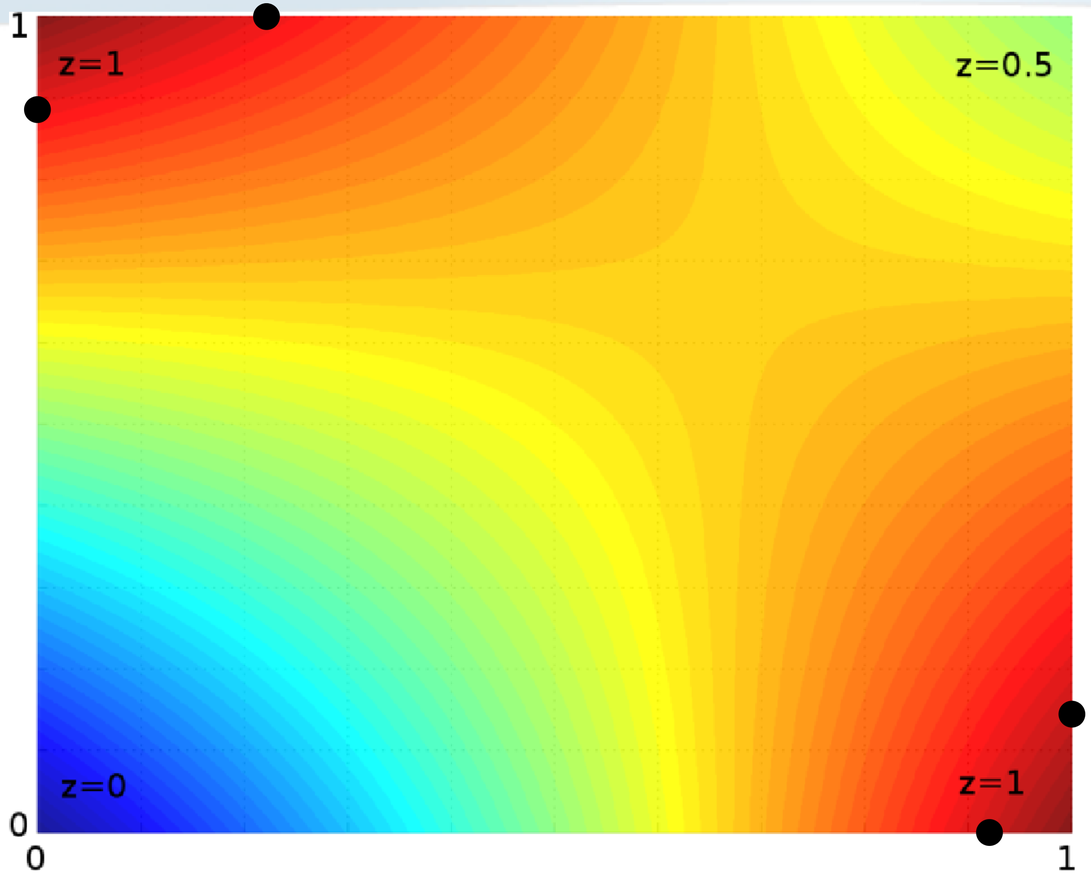


Issues of marching squares

$f^{-1}(0.95)$

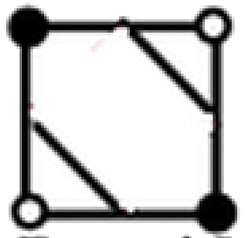


Case 10

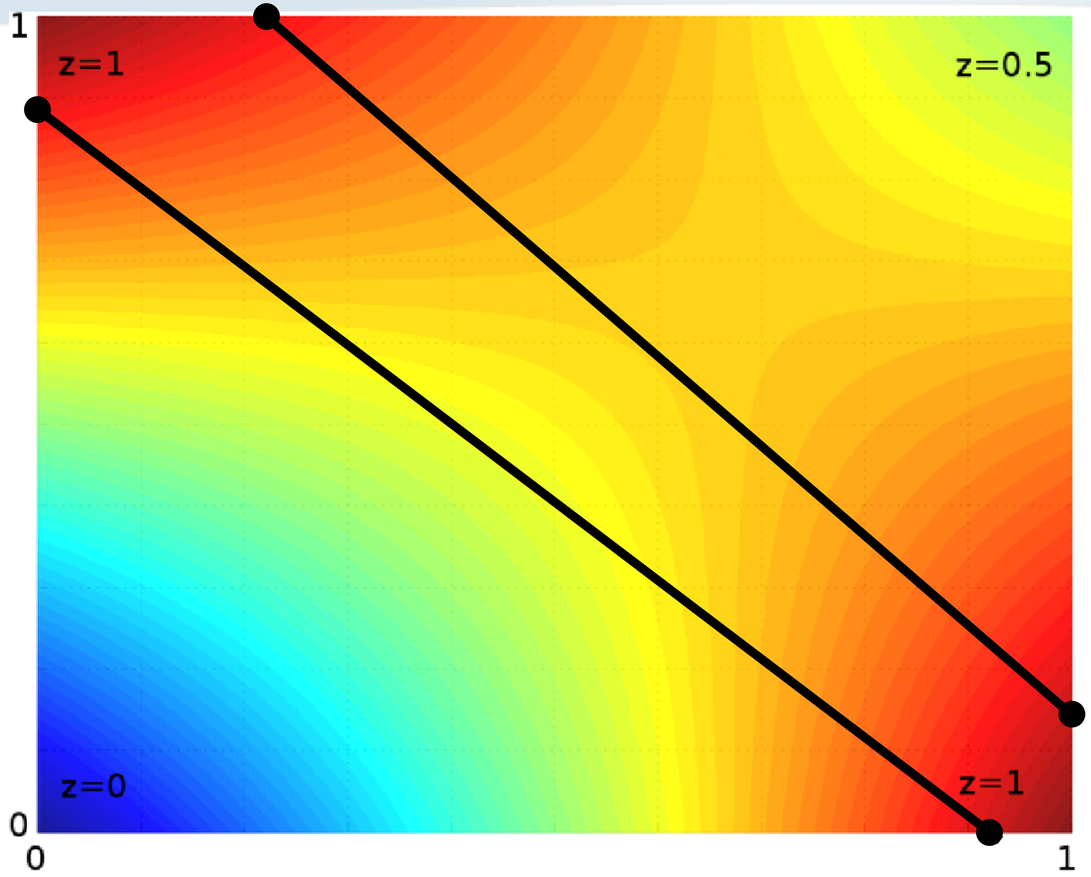


Issues of marching squares

$f^{-1}(0.95)$

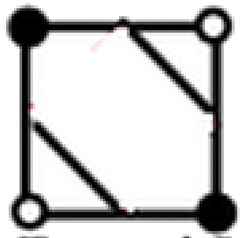


Case 10

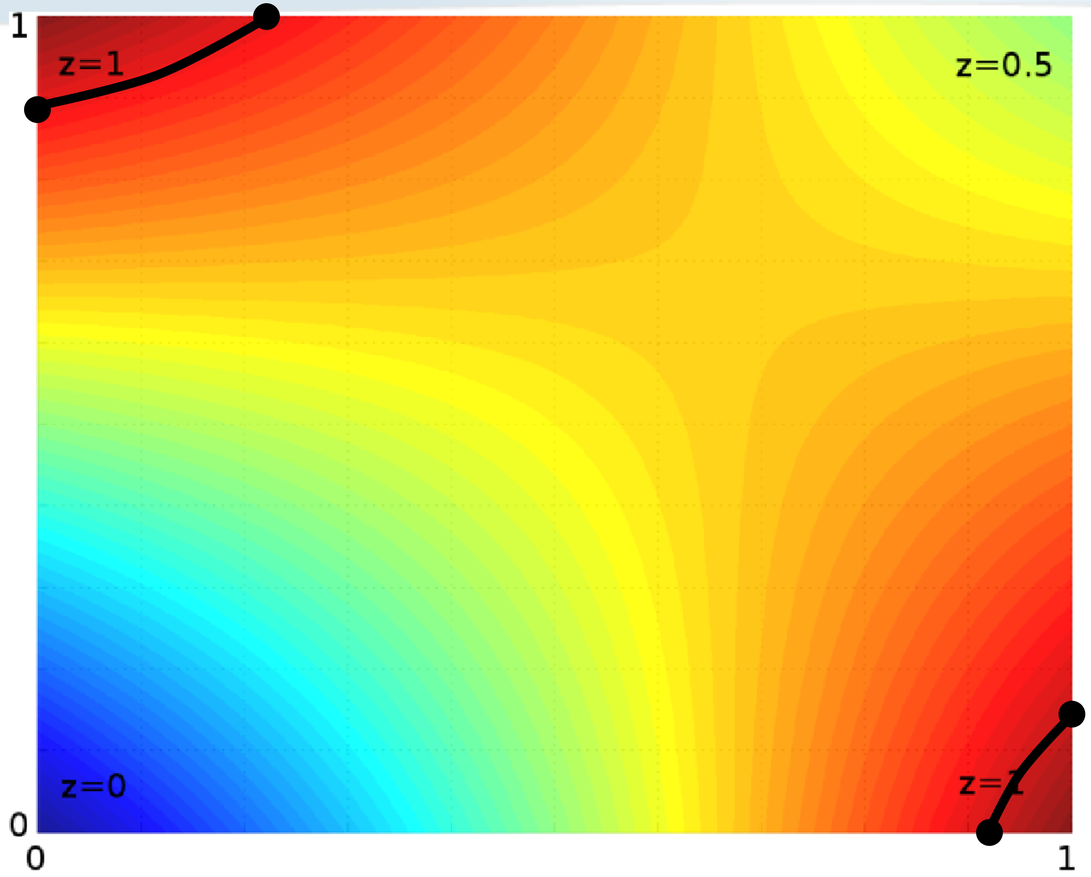


Issues of marching squares

$f^{-1}(0.95)$

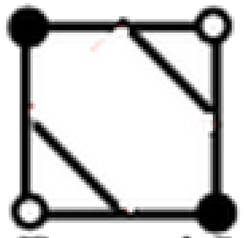


Case 10

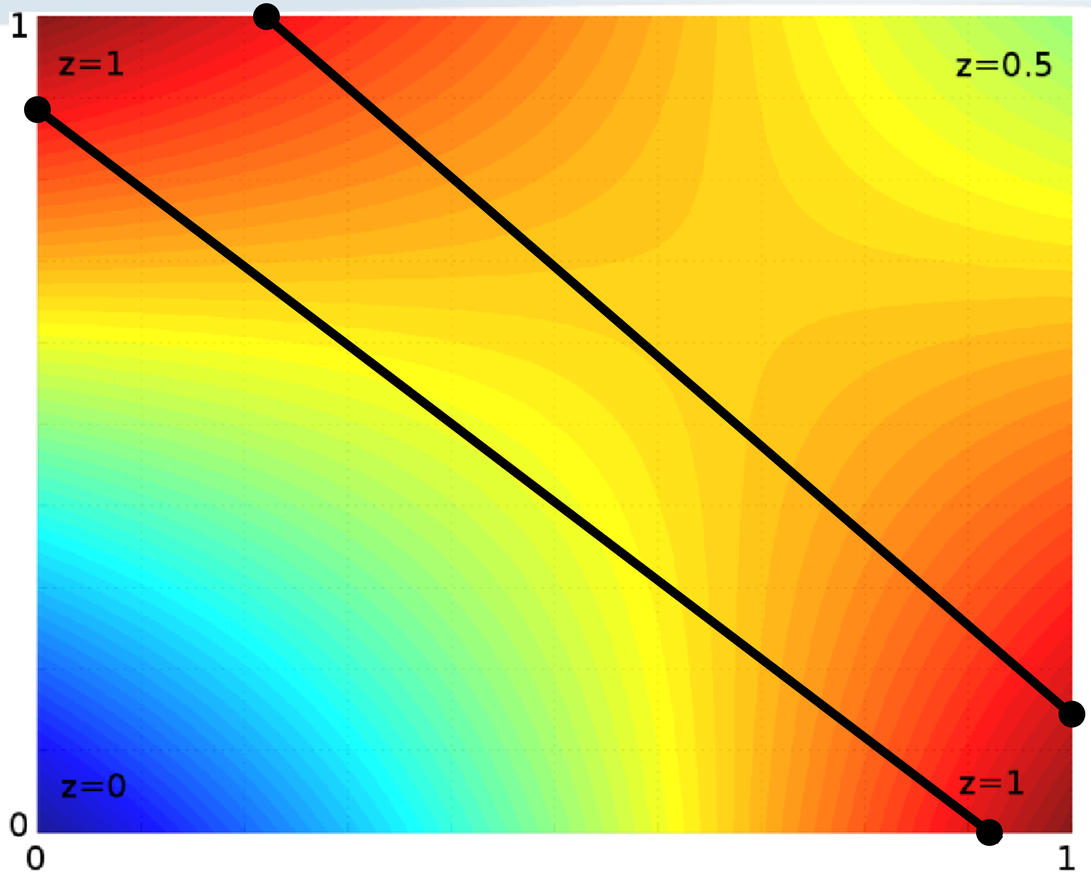


Issues of marching squares

$f^{-1}(0.95)$

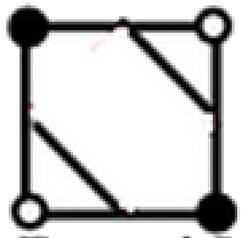


Case 10

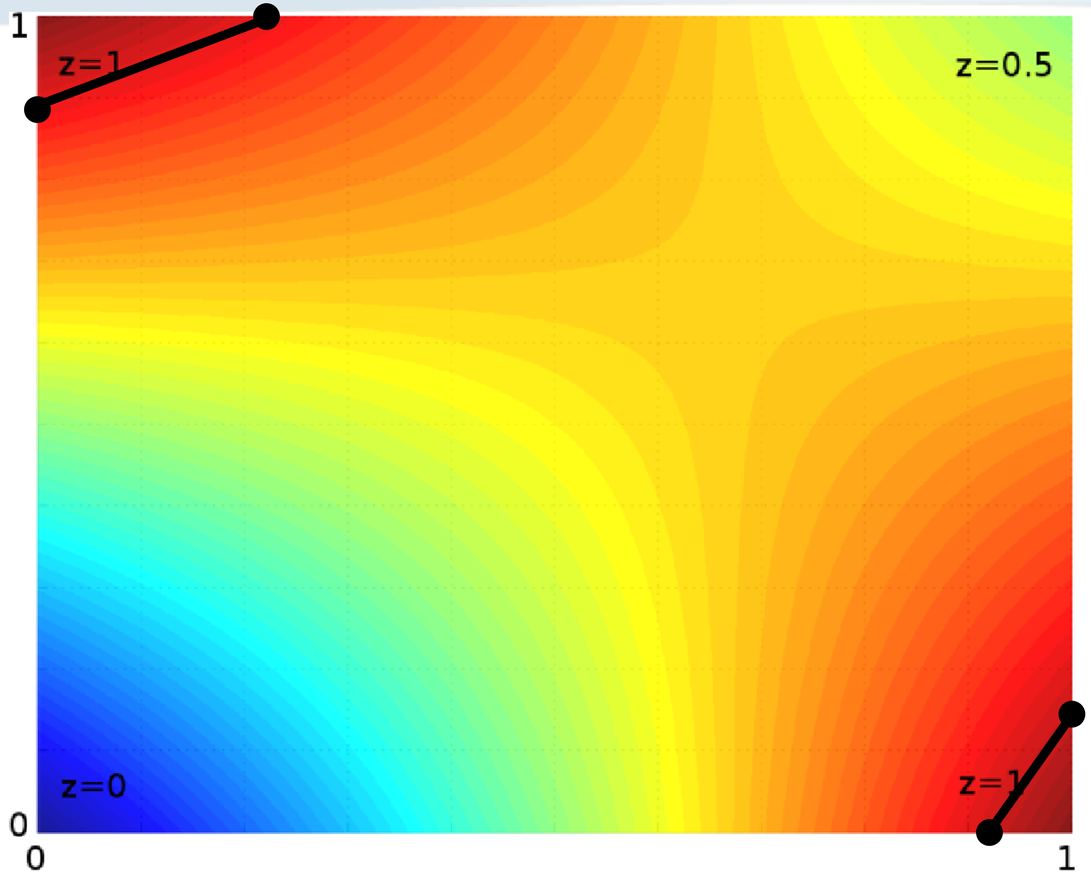


Issues of marching squares

$f^{-1}(0.95)$

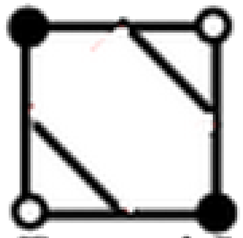


Case 10

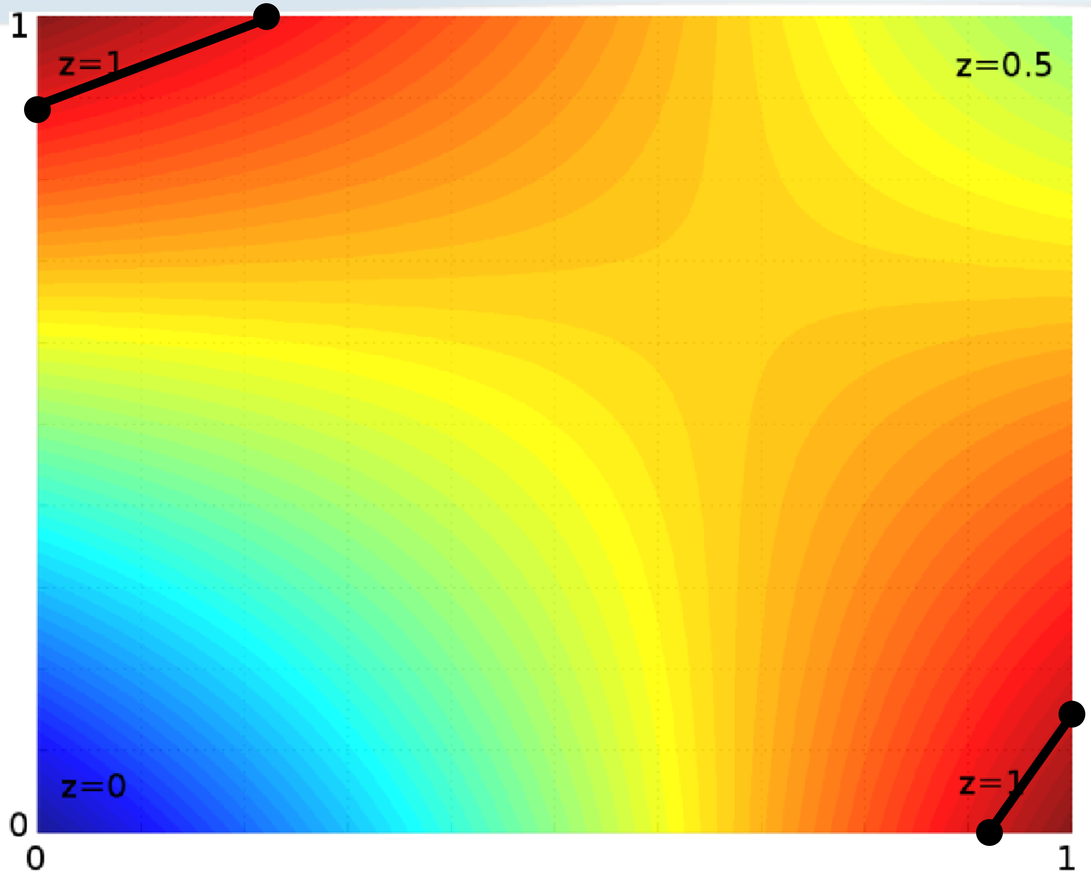


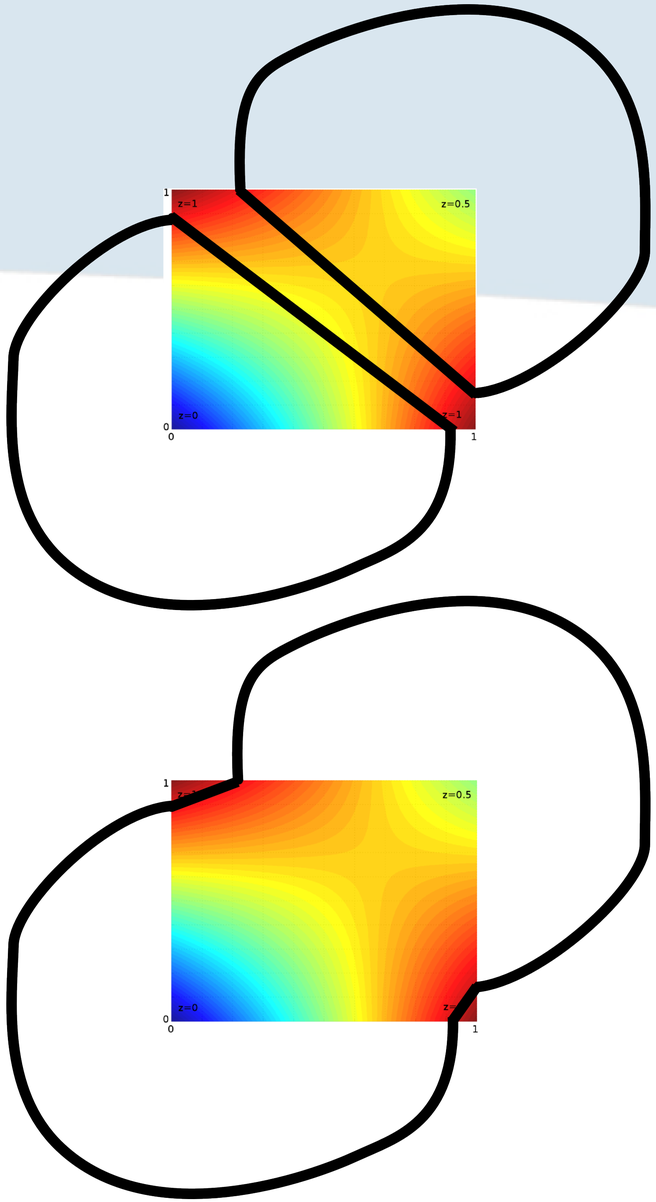
Issues of marching squares

$f^{-1}(0.95)$



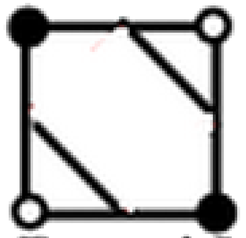
Case 10



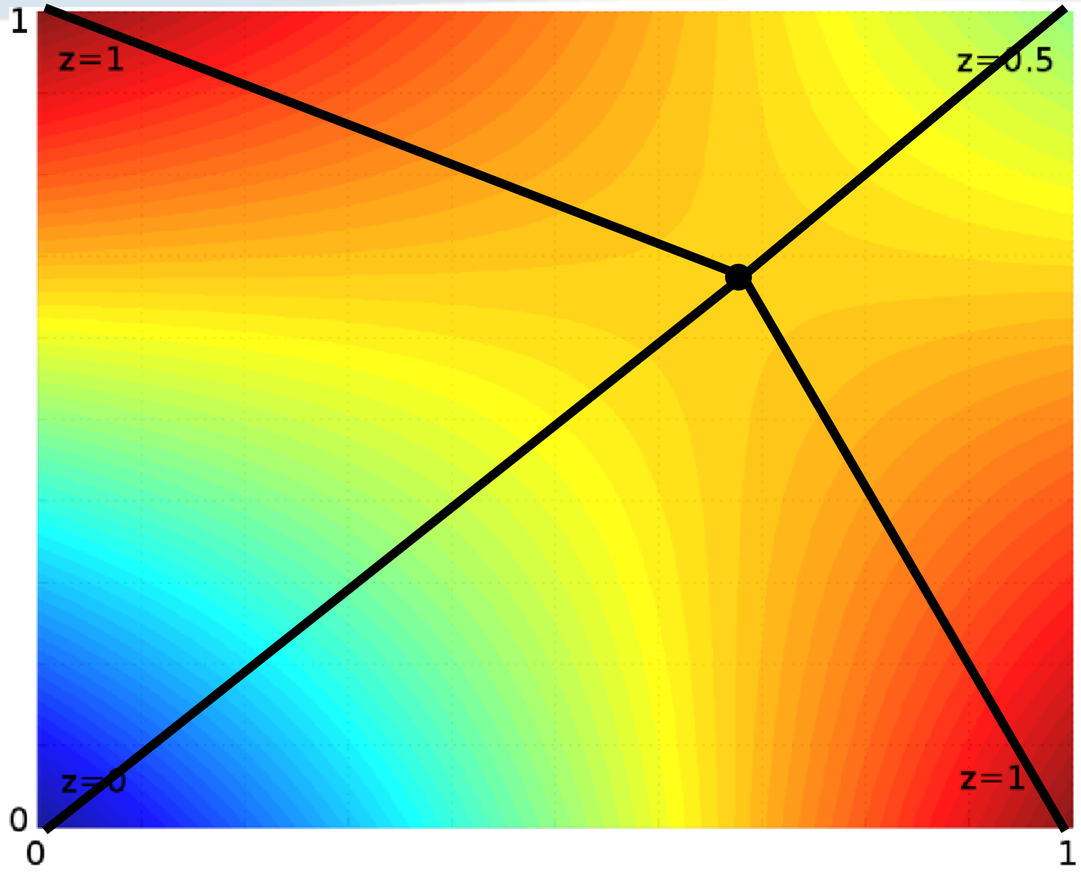


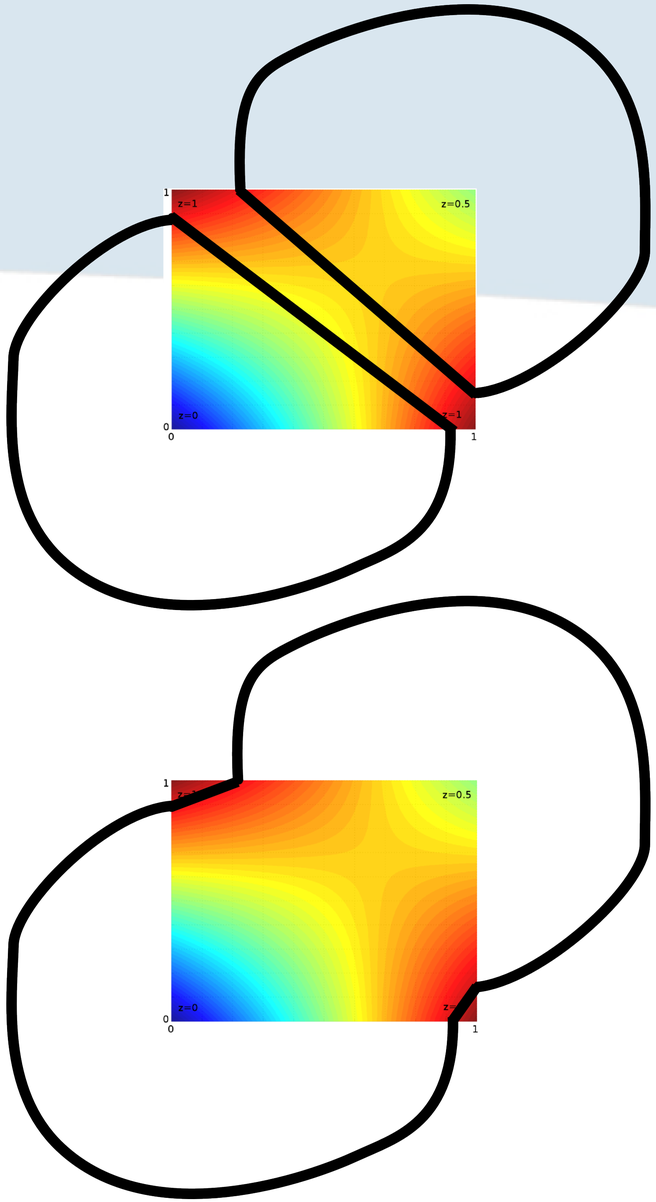
Issues of marching squares

$f^{-1}(0.95)$



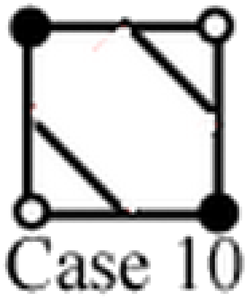
Case 10

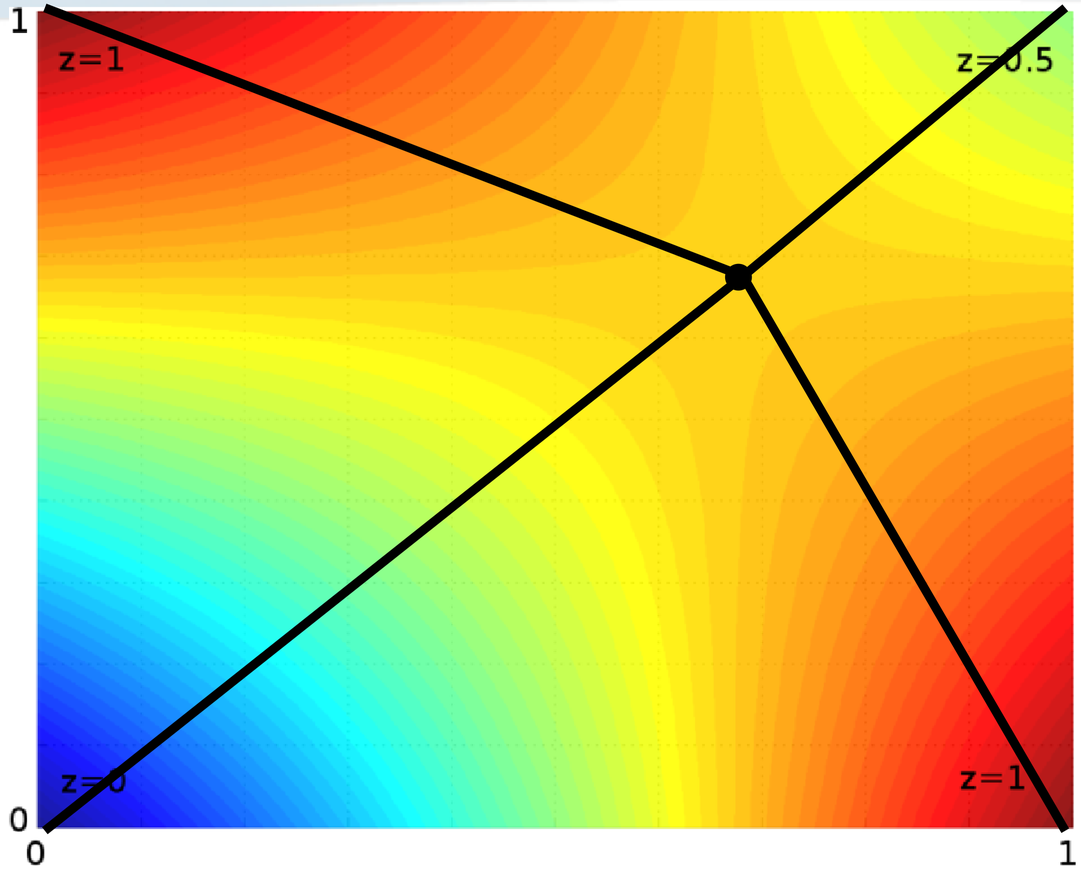


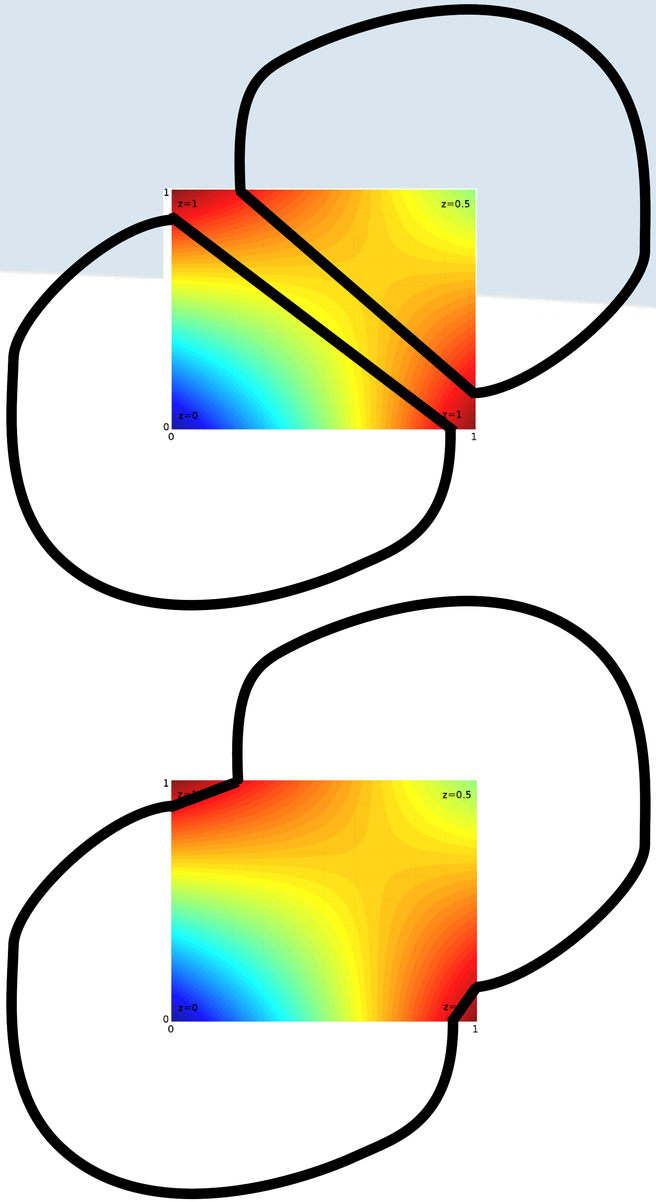


Issues of marching squares

$f^{-1}(0.95)$



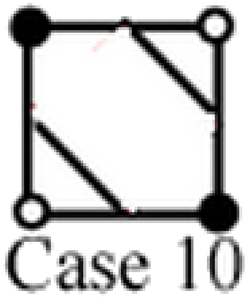


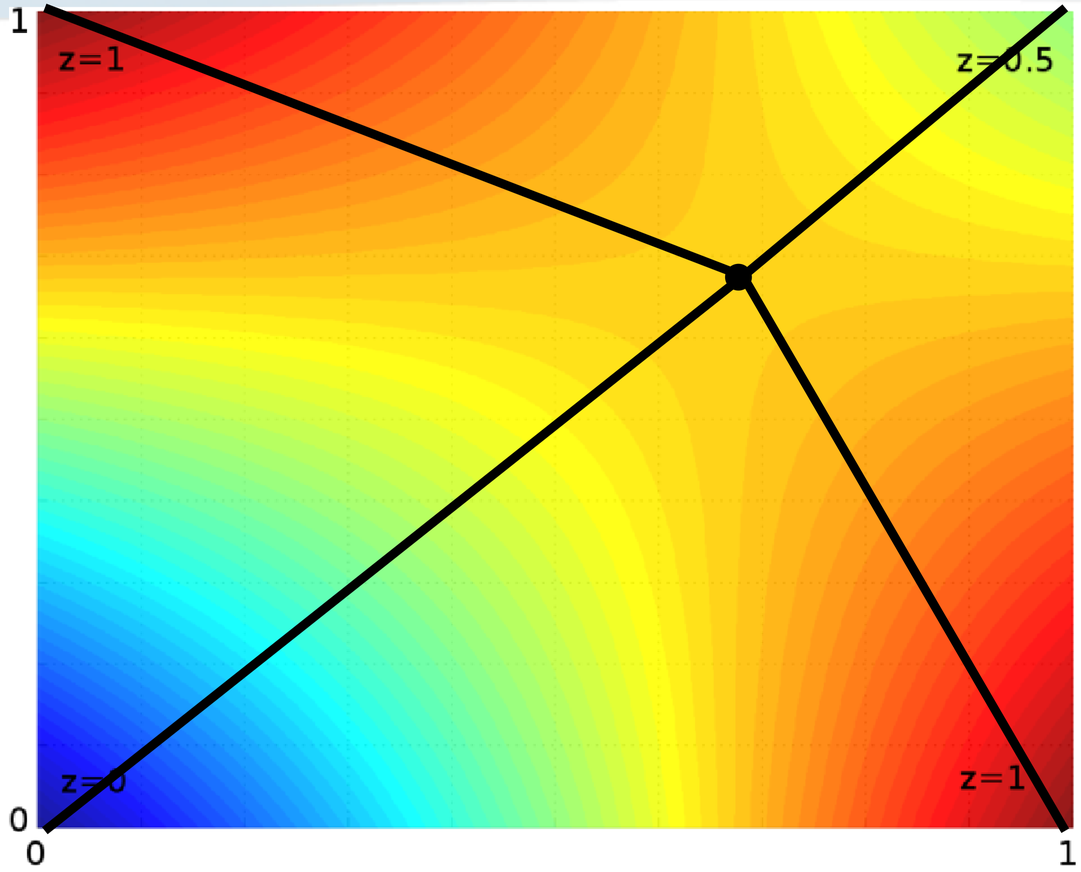


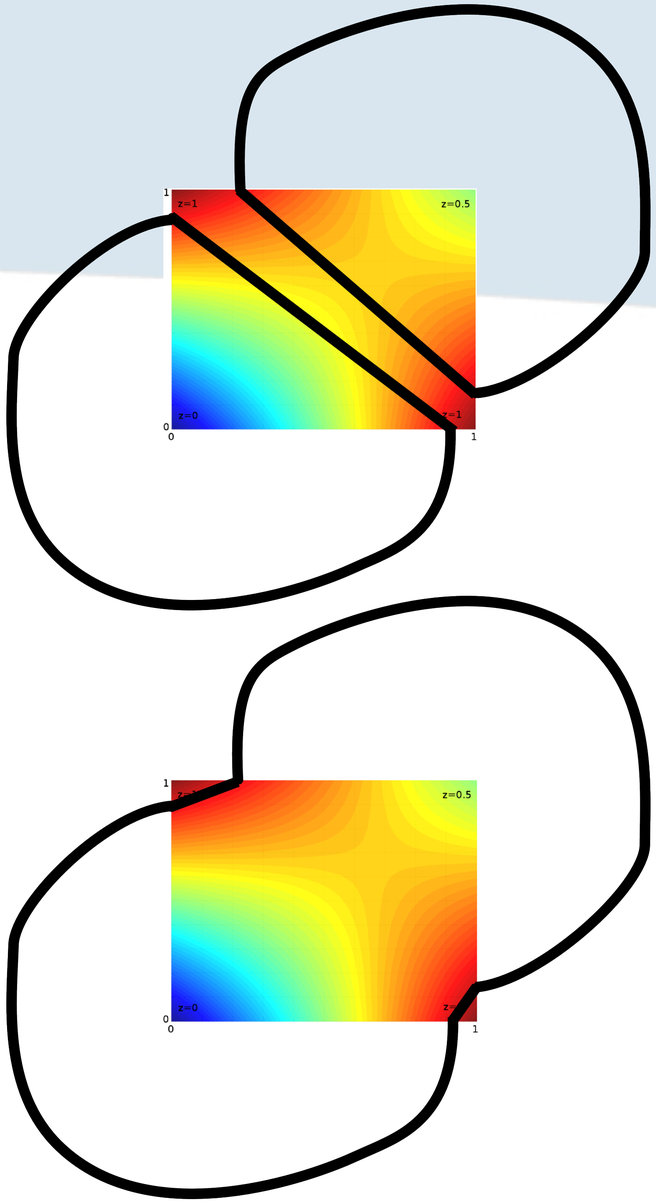
- Geometrically inaccurate (lines)

Issues of marching squares

$f^{-1}(0.95)$



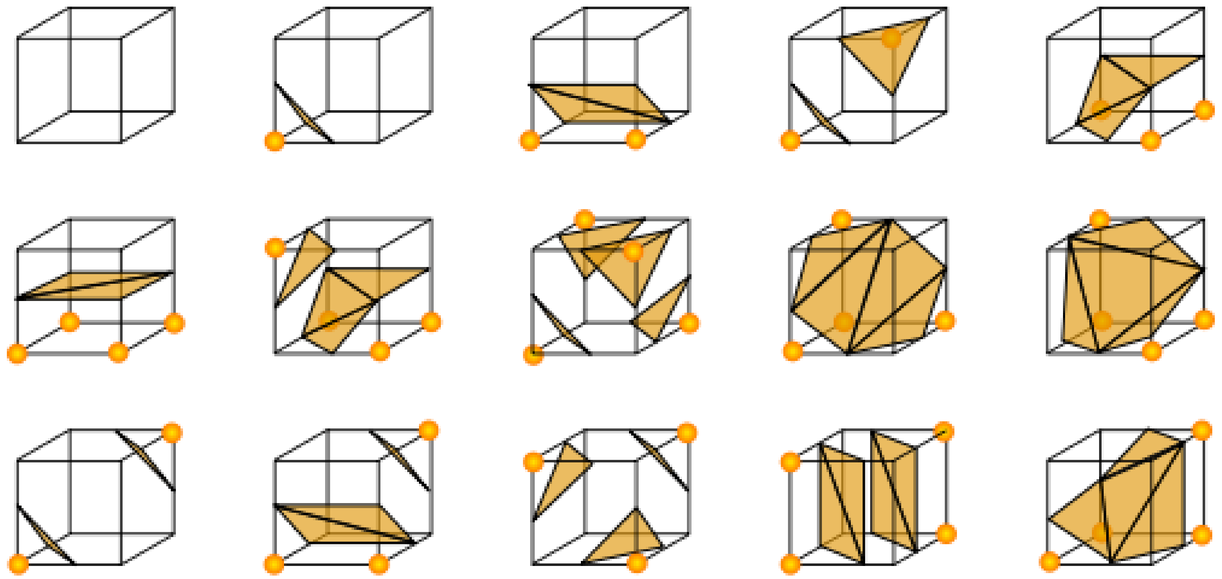




- Geometrically inaccurate (lines)
- Topologically inconsistent (**numerical** estimation of the saddle)

Marching cubes

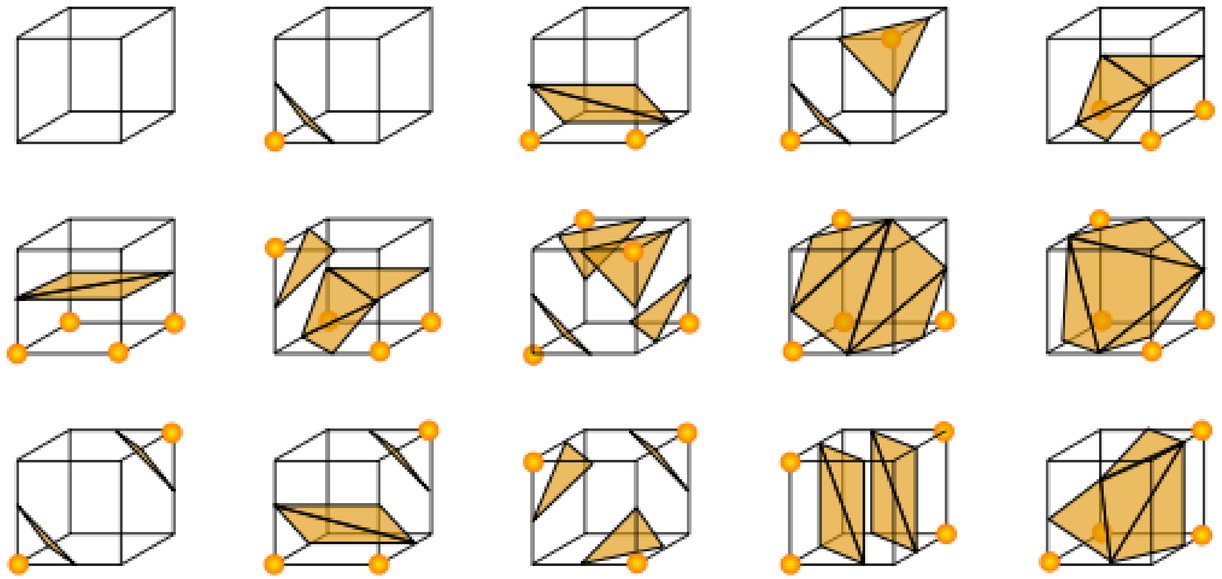
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant



[Wikipedia]

Marching cubes

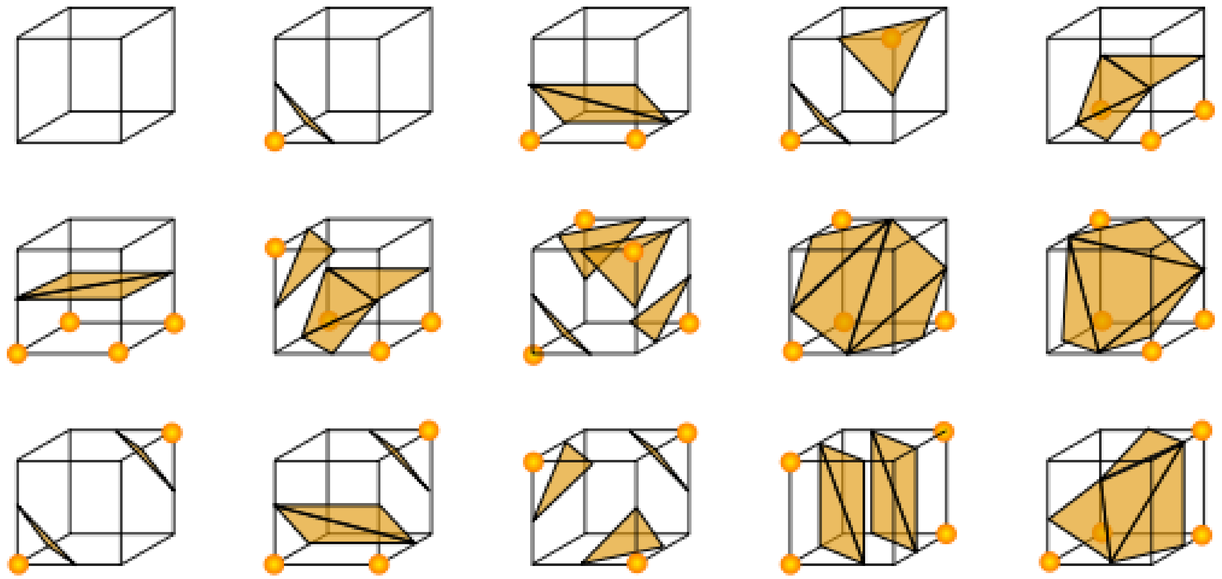
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells



[Wikipedia]

Marching cubes

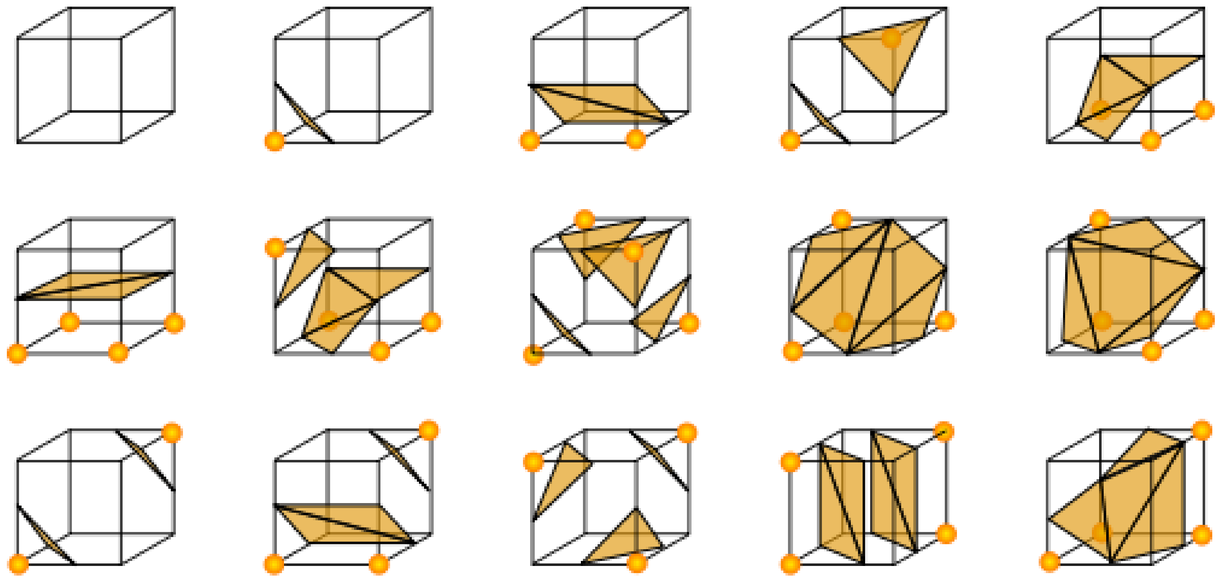
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 3D unit cell



[Wikipedia]

Marching cubes

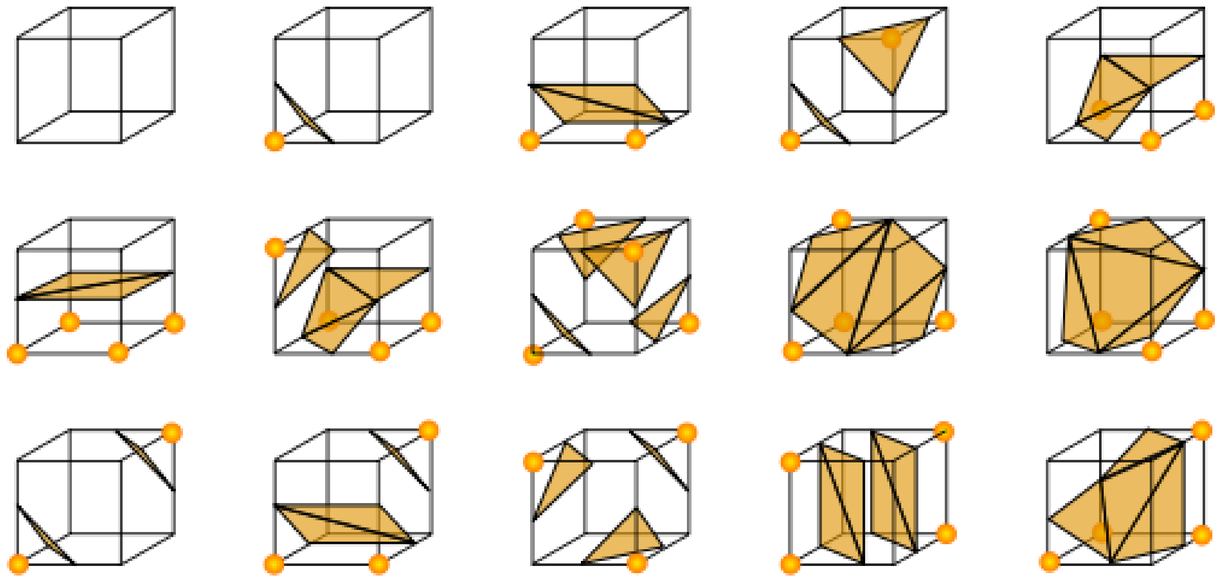
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 3D unit cell
    - 256 cases



[Wikipedia]

Issues of marching cubes

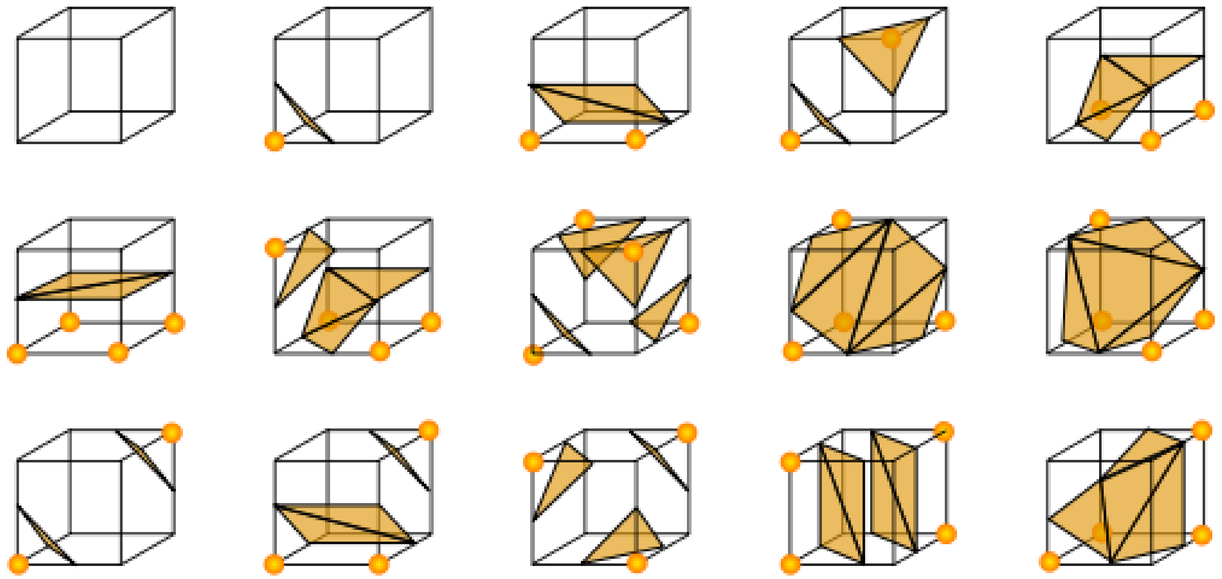
- Same as before



[Wikipedia]

Issues of marching cubes

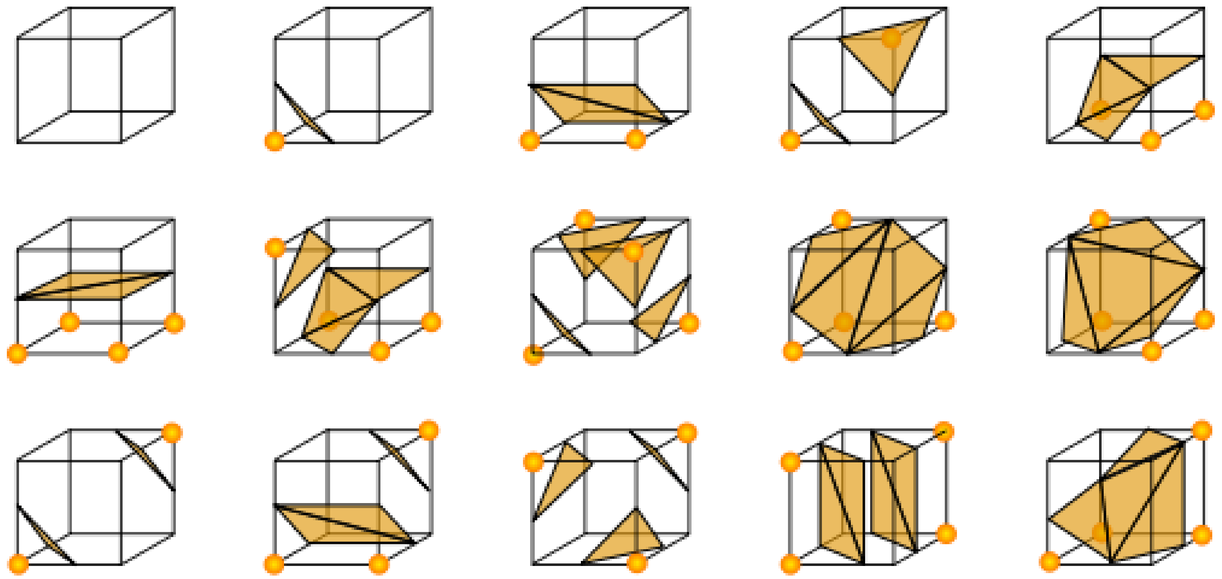
- Same as before
  - But worse!



[Wikipedia]

Issues of marching cubes

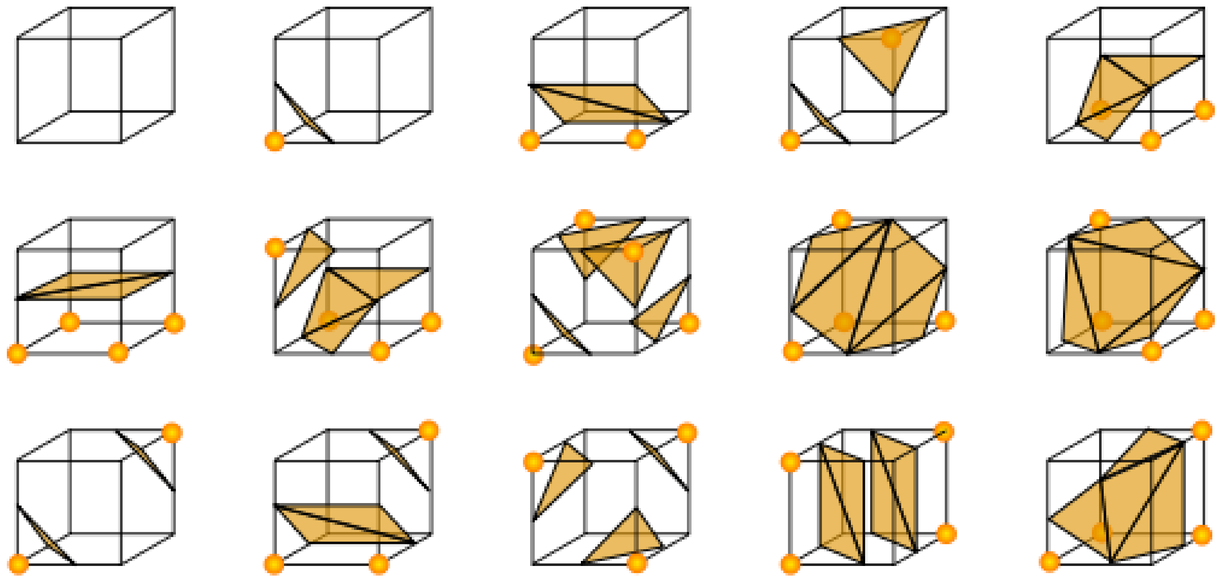
- Same as before
  - But worse!
- Lorensen and Cline 1987



[Wikipedia]

Issues of marching cubes

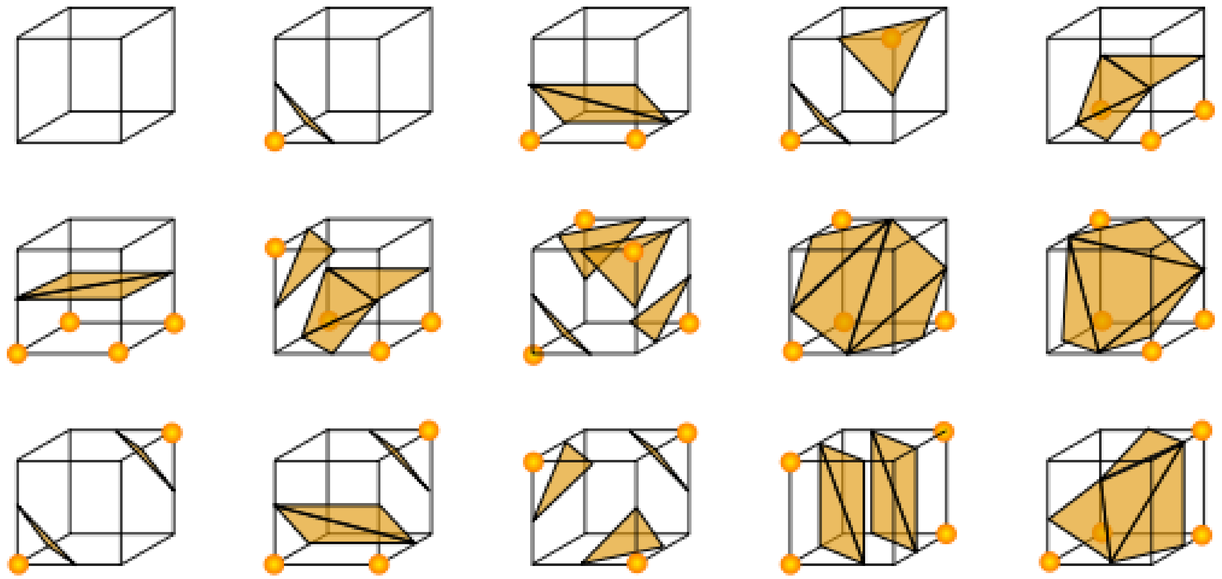
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)



[Wikipedia]

Issues of marching cubes

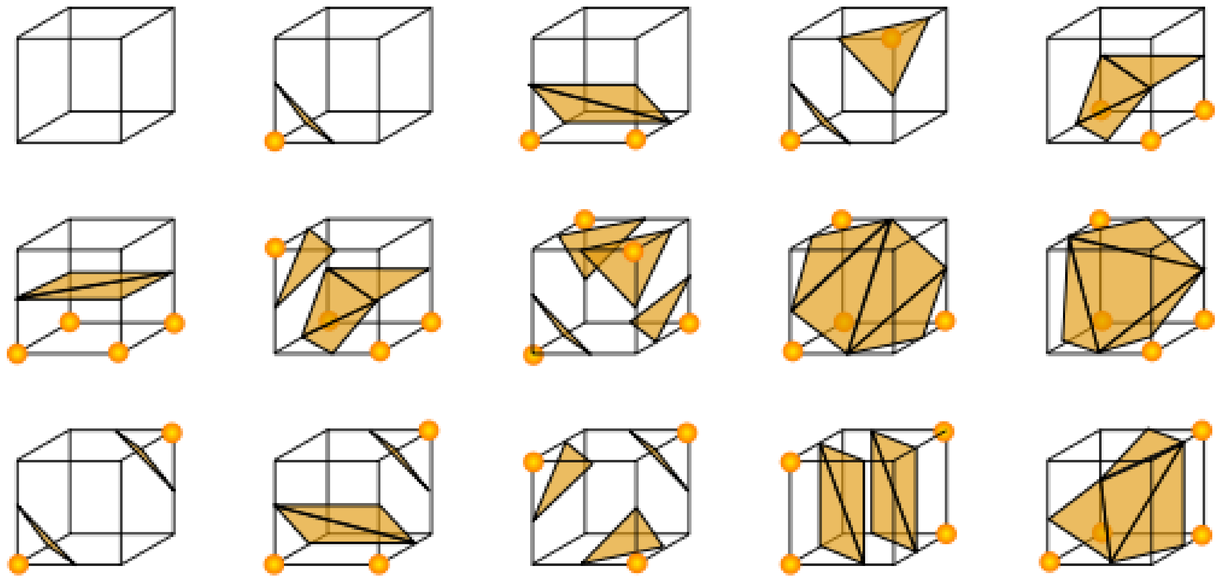
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks



[Wikipedia]

Issues of marching cubes

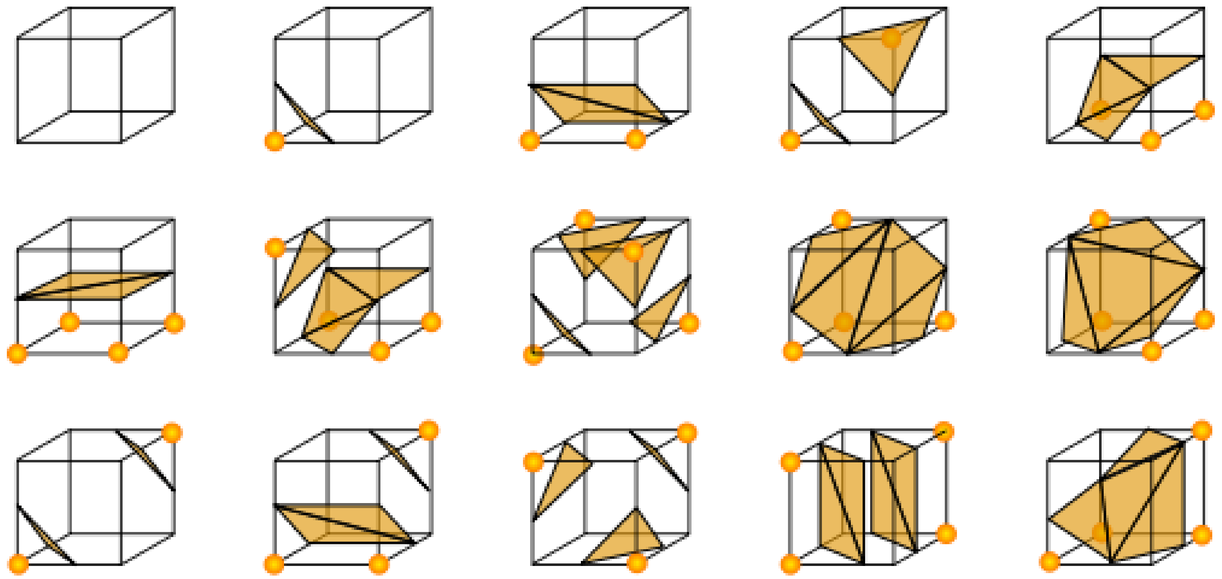
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases



[Wikipedia]

Issues of marching cubes

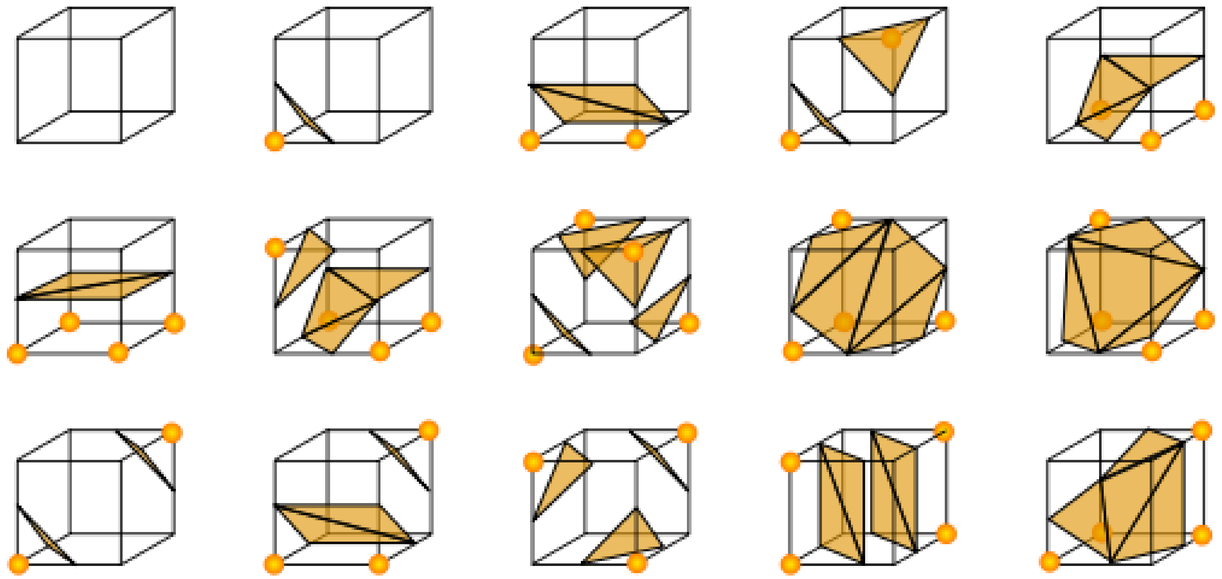
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases
- Saddle curse (components and genus):



[Wikipedia]

Issues of marching cubes

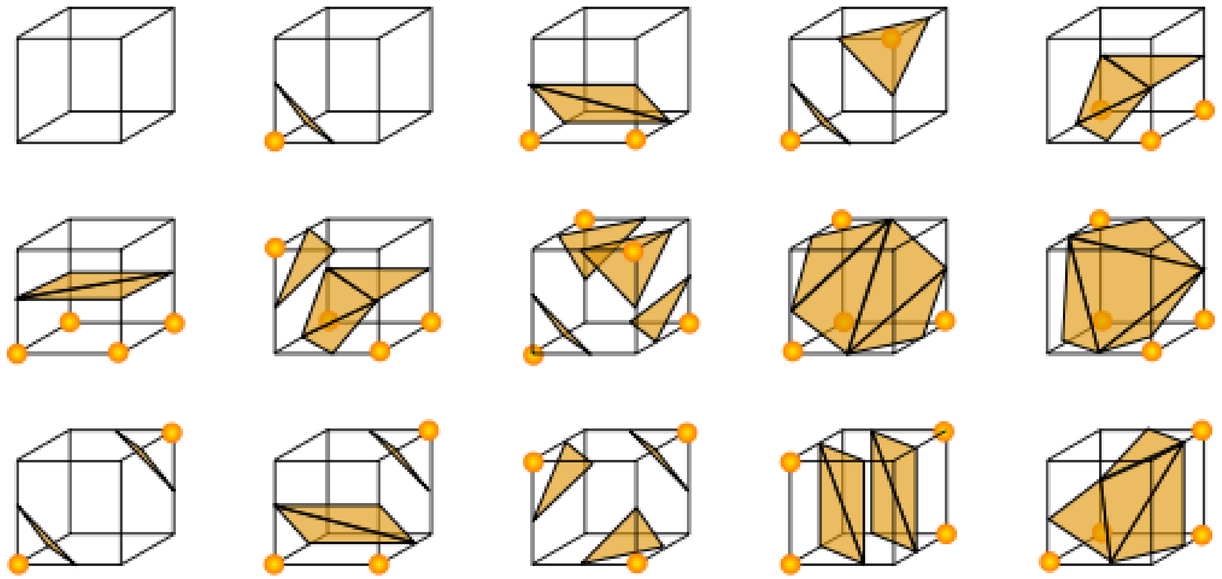
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases
- Saddle curse (components and genus):
  - Nielson and Hamann 1991



[Wikipedia]

Issues of marching cubes

- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases

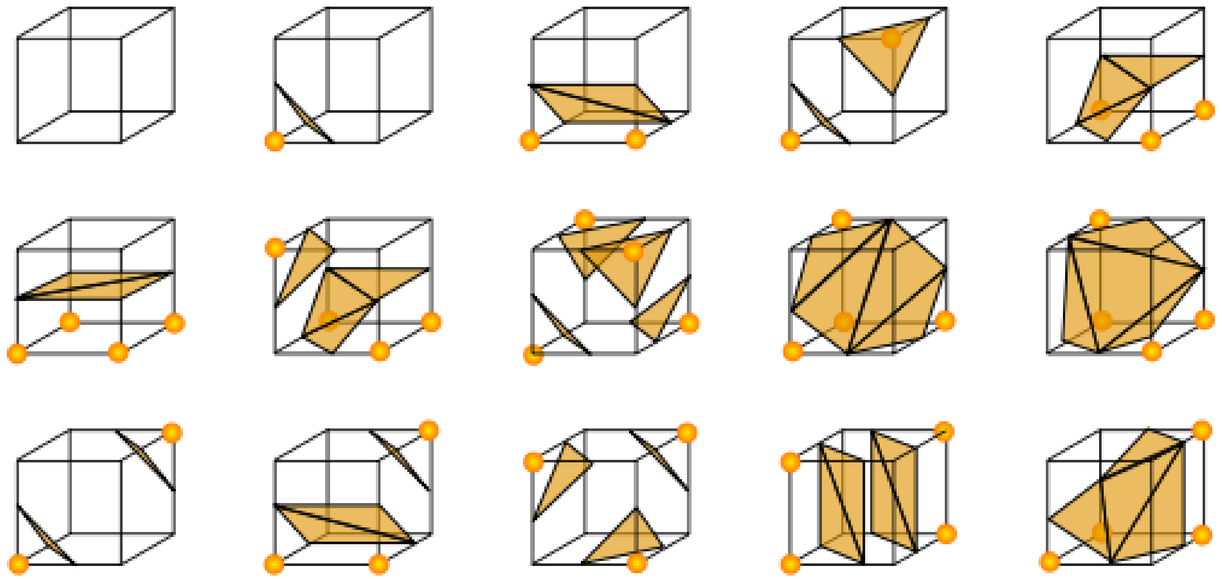


[Wikipedia]

- Saddle curse (components and genus):
  - Nielson and Hamann 1991, Natarajan 1994

Issues of marching cubes

- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases

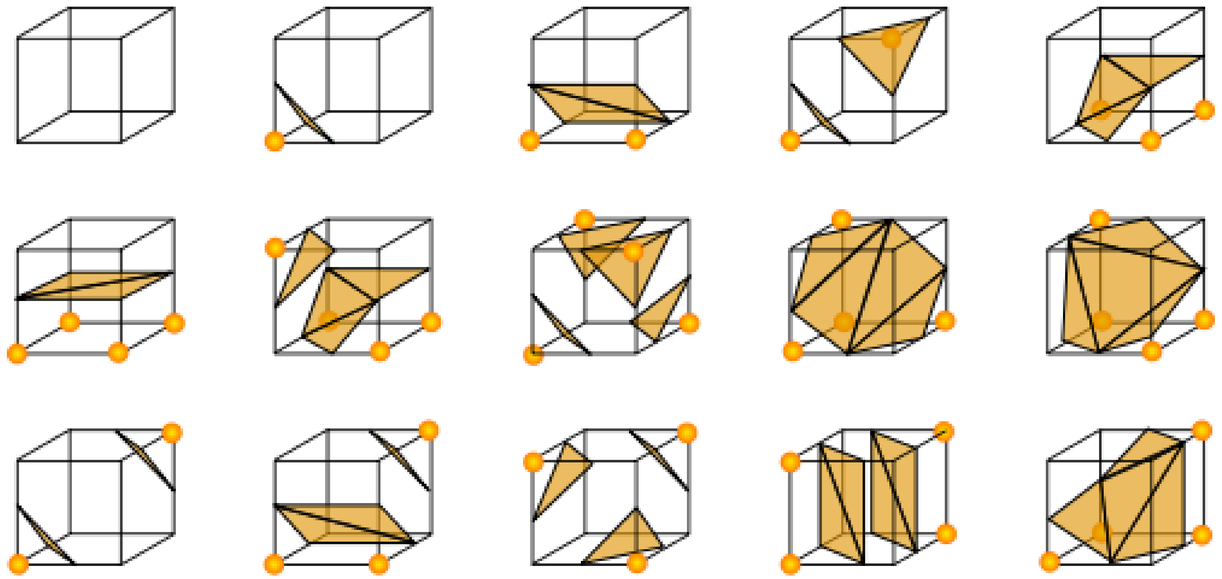


[Wikipedia]

- Saddle curse (components and genus):
  - Nielson and Hamann 1991, Natarajan 1994, Chernyaev 1995

Issues of marching cubes

- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases

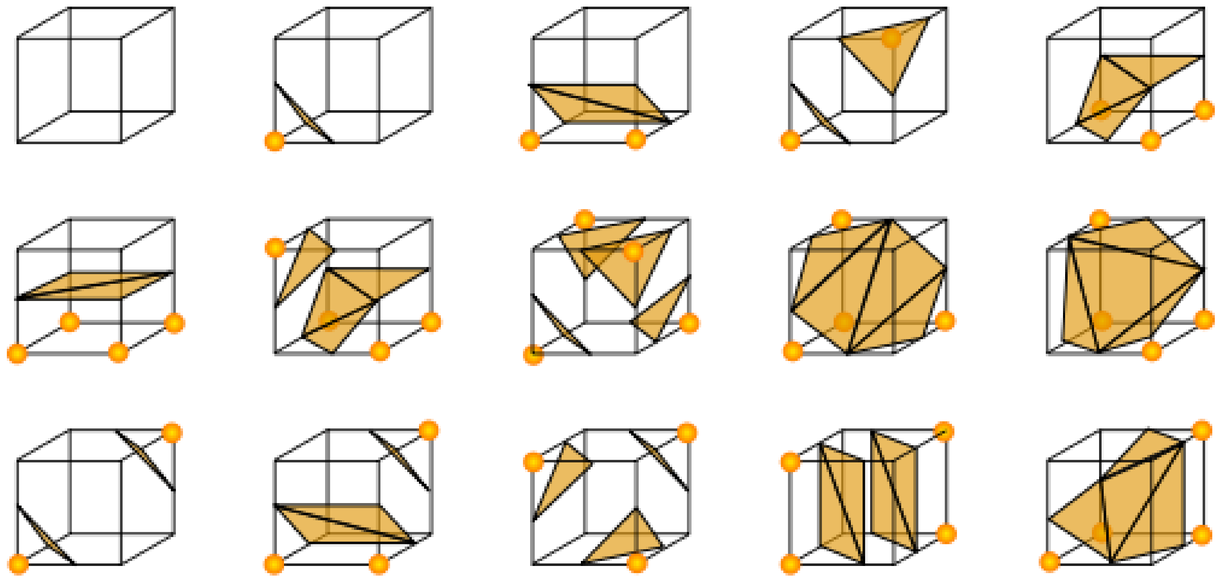


[Wikipedia]

- Saddle curse (components and genus):
  - Nielson and Hamann 1991, Natarajan 1994, Chernyaev 1995, Lewiner 2003

Issues of marching cubes

- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases

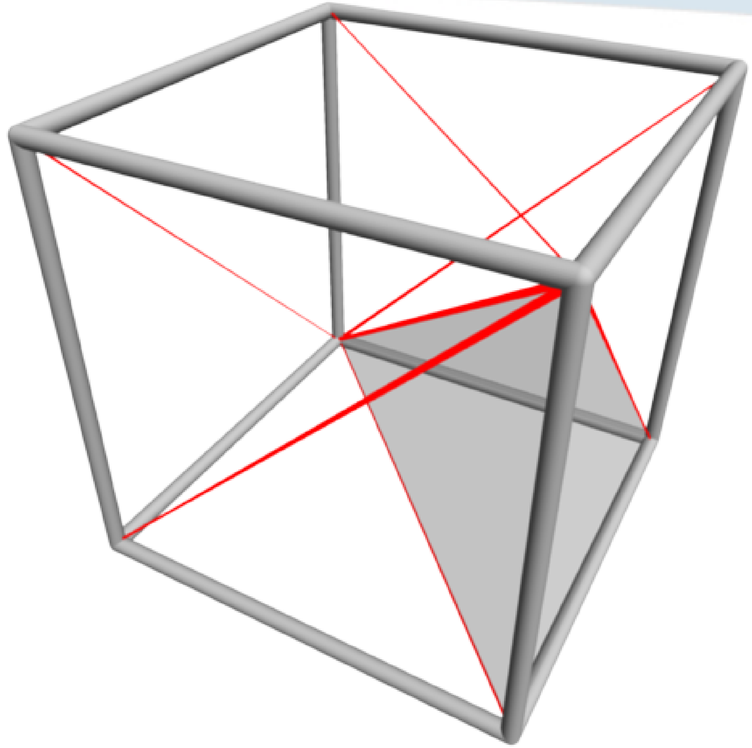


[Wikipedia]

- Saddle curse (components and genus):
  - Nielson and Hamann 1991, Natarajan 1994, Chernyaev 1995, Lewiner 2003, *Etiene 2012*

Issues of marching cubes

- Same as before
  - But worse!
- Lorensen and Cline 1987
    - 16 cases (symmetries)
    - Possible cracks
    - Improvement 28 cases
- Saddle curse (components and genus):
  - Nielson and Hamann 1991, Natarajan 1994, Chernyaev 1995, Lewiner 2003, *Etiene 2012*



[Wikipedia]

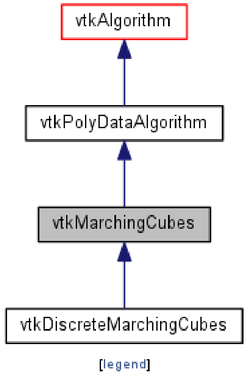
VTK

vtkMarchingCubes Class Reference

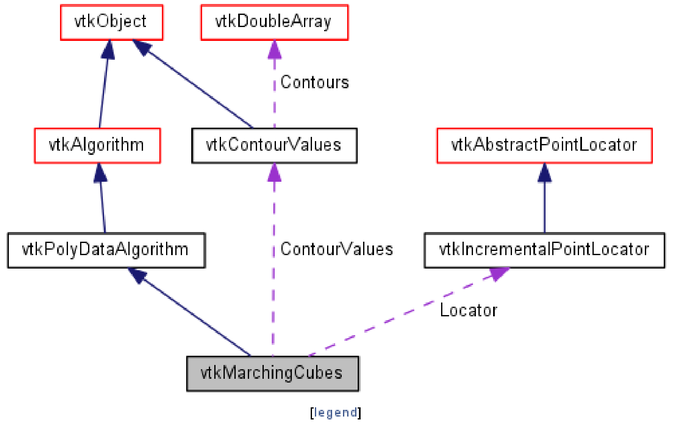
generate isosurface(s) from volume More...

#include <vtkMarchingCubes.h>

Inheritance diagram for vtkMarchingCubes:



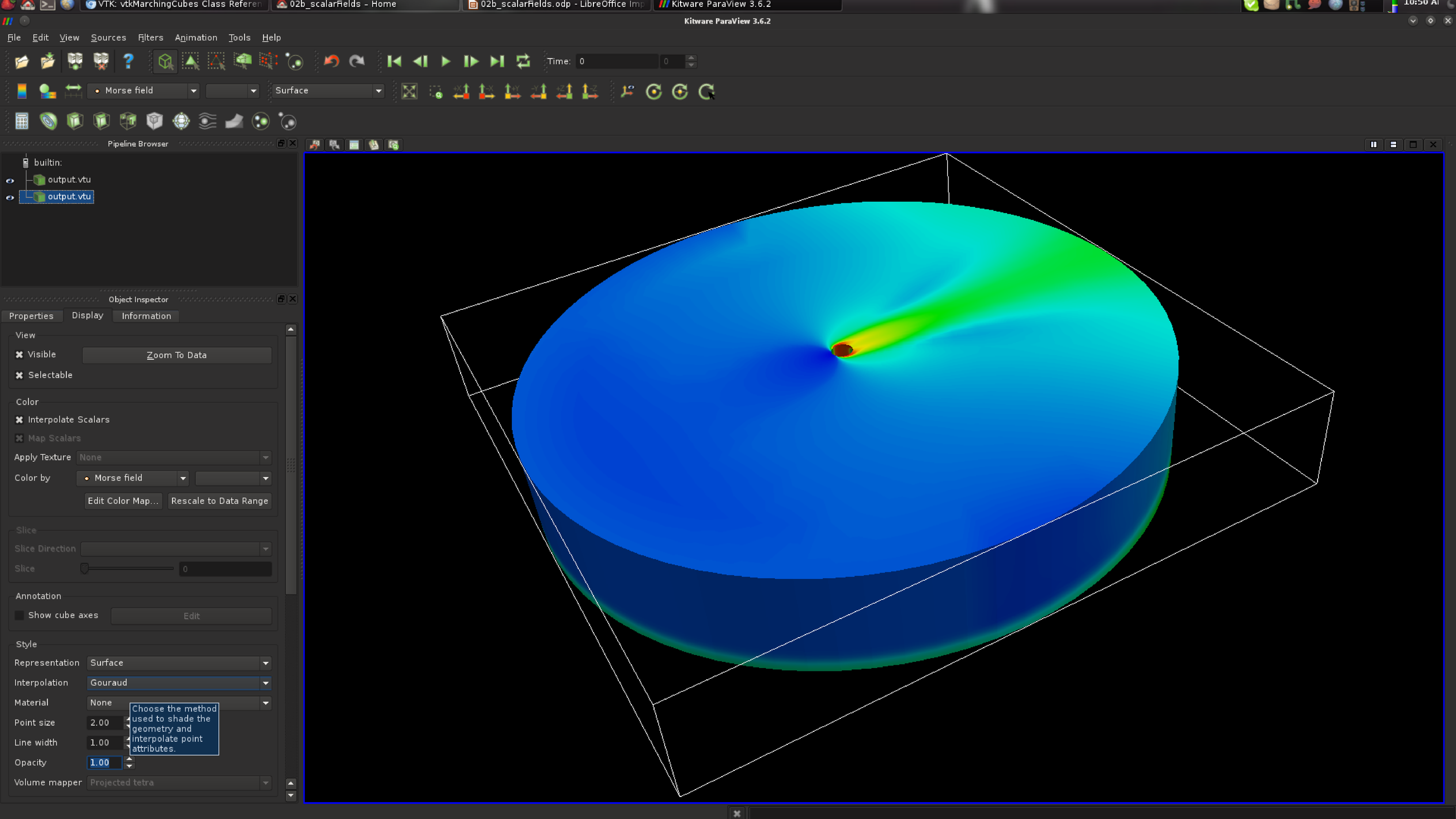
Collaboration diagram for vtkMarchingCubes:

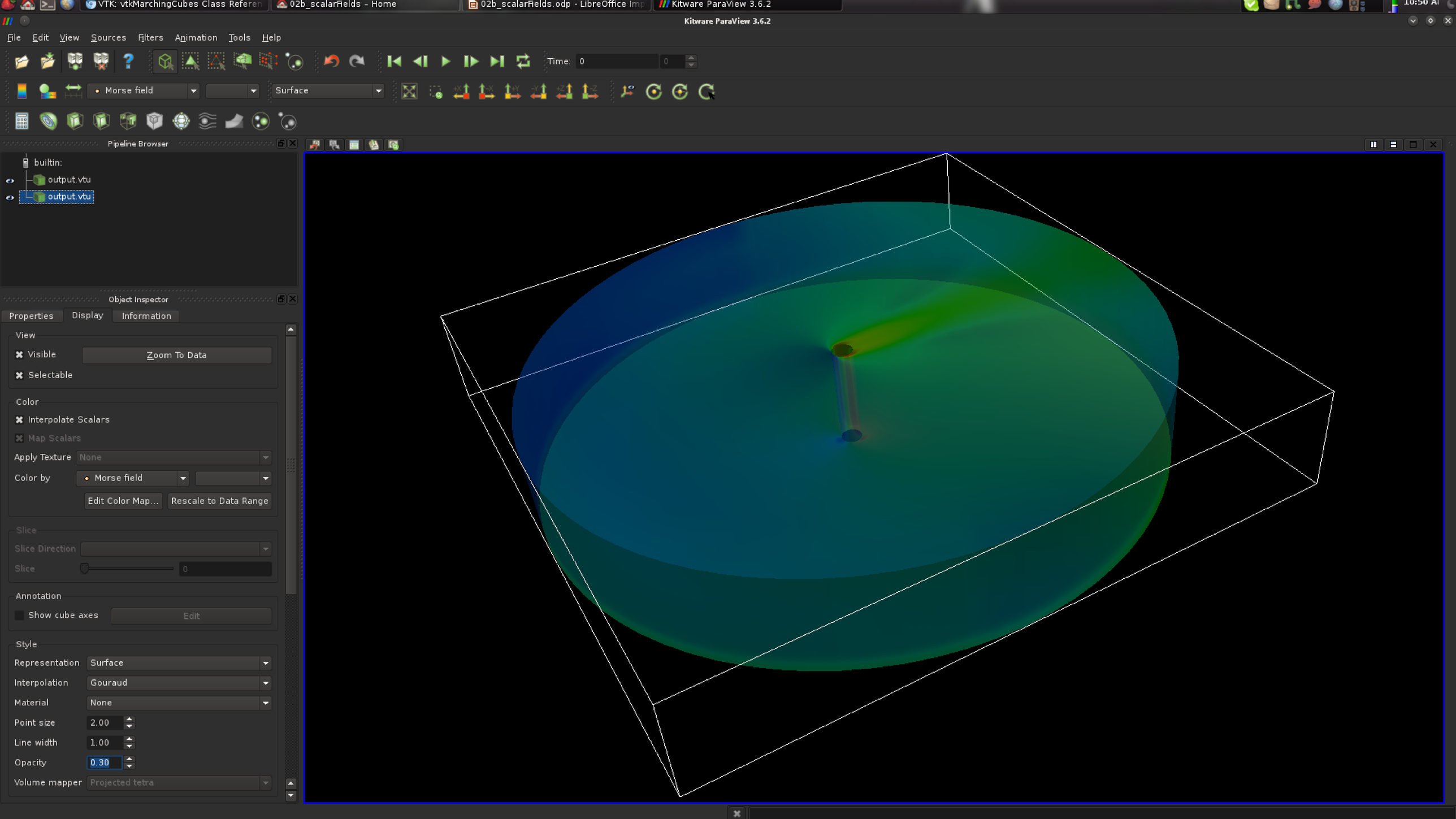


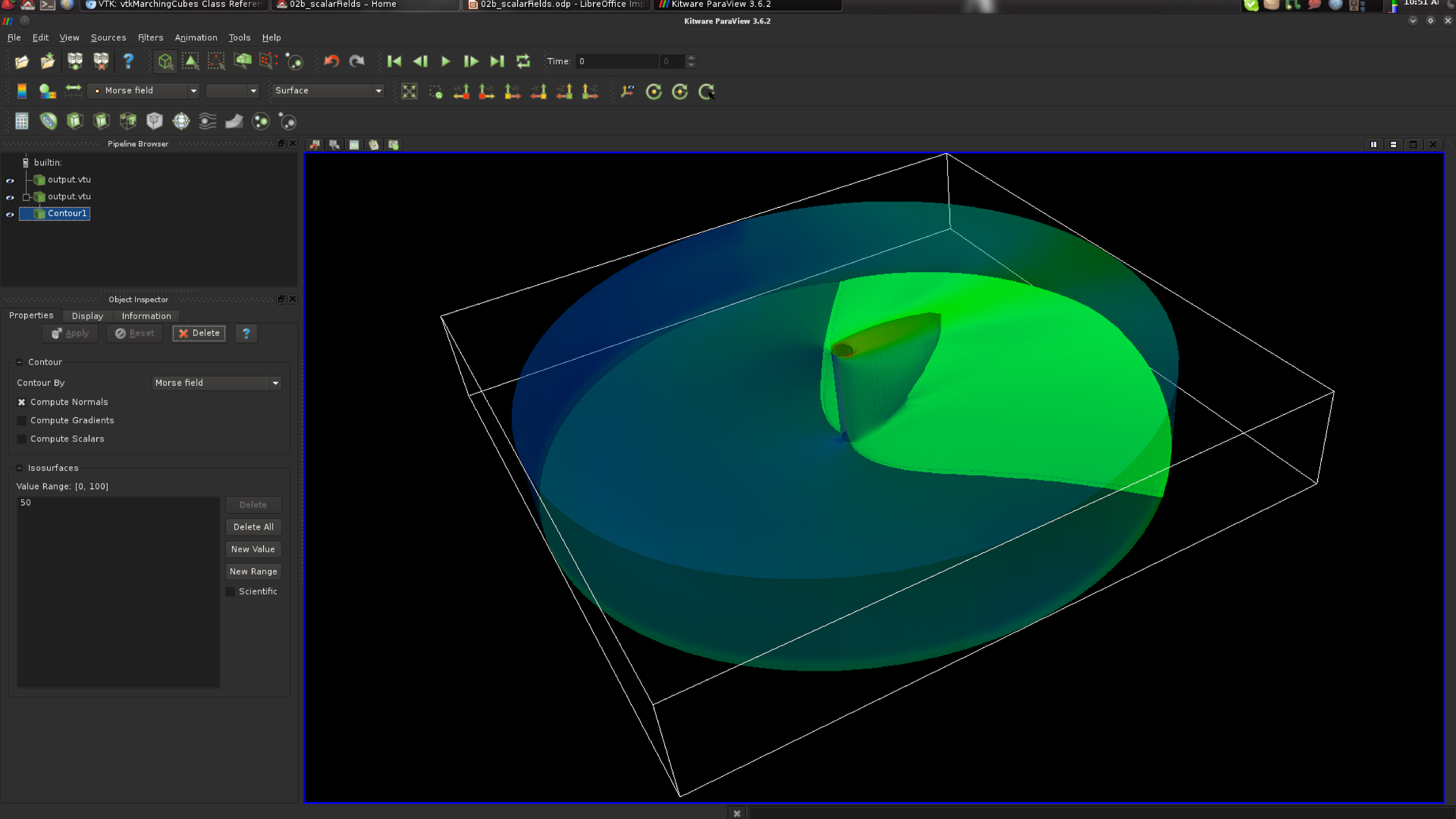
Public Types

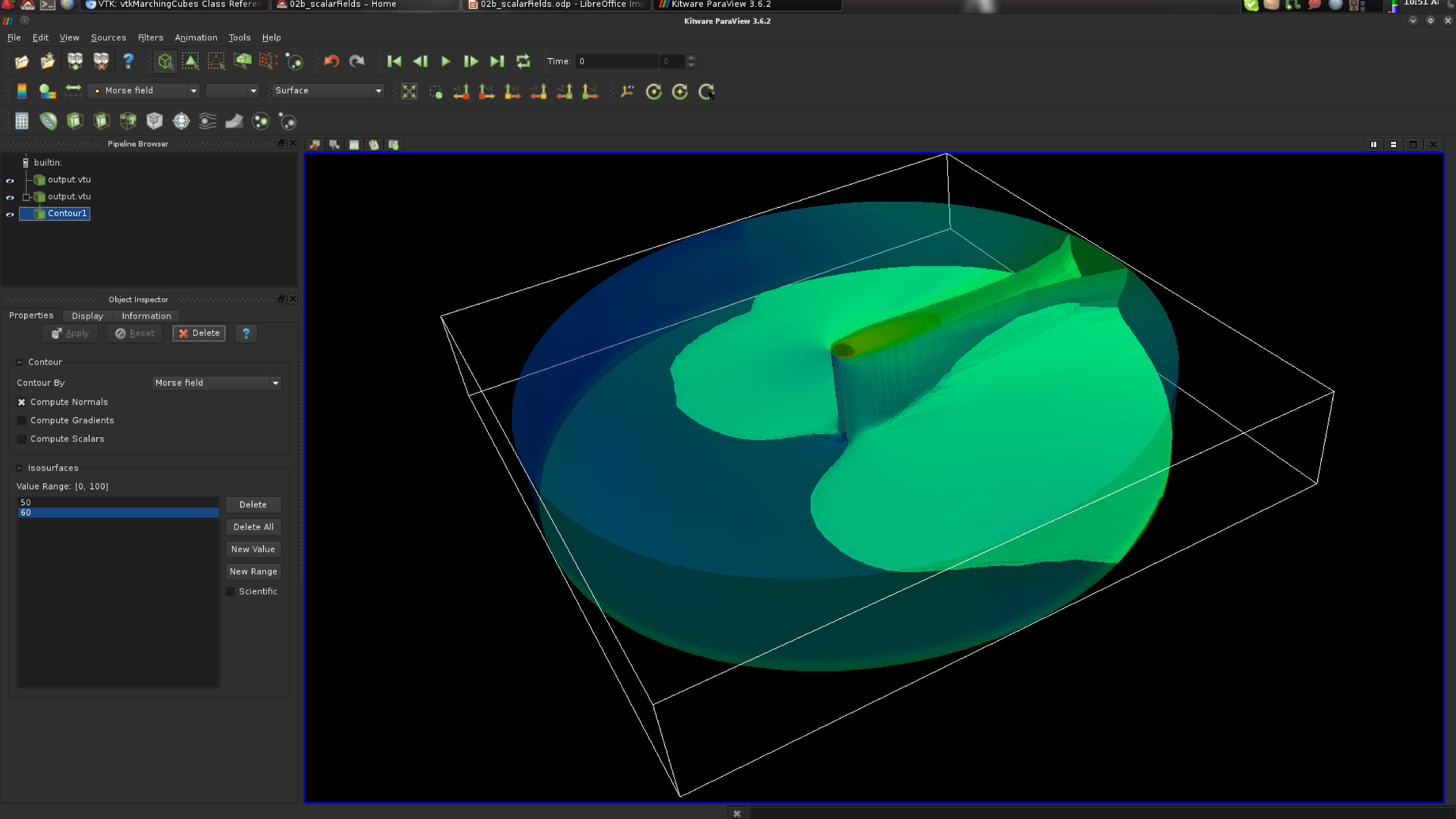
typedef vtkPolyDataAlgorithm Superclass

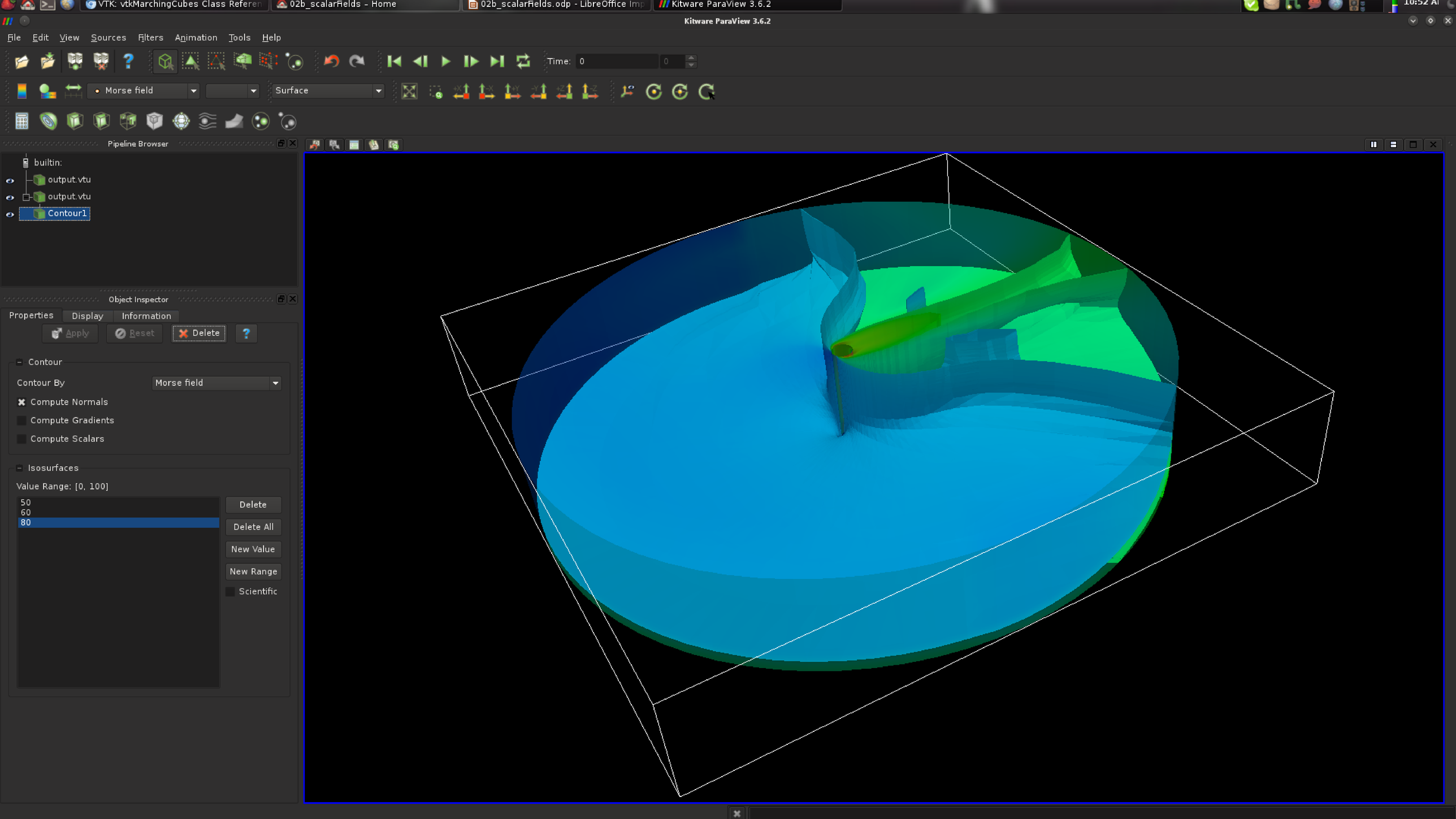
Public Types inherited from vtkPolyDataAlgorithm

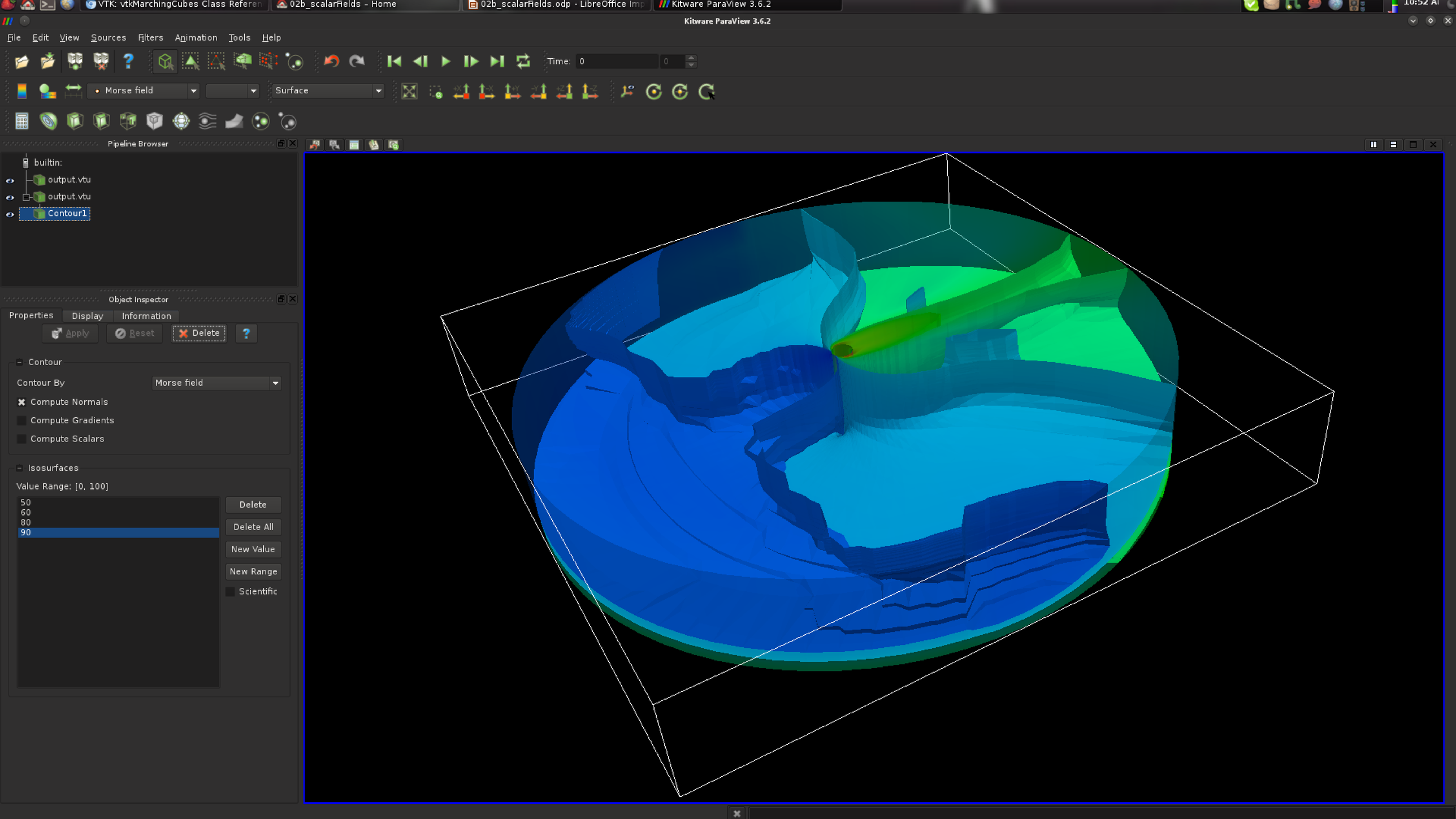

VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Object Inspector
Properties Display Information
View
Visible
Zoom To Data
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Morse field
Edit Color Map...
Rescale to Data Range
Slice
Slice Direction
Slice 0
Annotation
Show cube axes
Edit
Style
Representation Surface
Interpolation Gouraud
Material None
Point size 2.00
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Choose the method used to shade the geometry and interpolate point attributes.


VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Object Inspector
Properties Display Information
View
Visible
Zoom To Data
Selectable
Color
Interpolate Scalars
Map Scalars
Apply Texture None
Color by Morse field
Edit Color Map...
Rescale to Data Range
Slice
Slice Direction
Slice 0
Annotation
Show cube axes
Edit
Style
Representation Surface
Interpolation Gouraud
Material None
Point size 2.00
Line width 1.00
Opacity 0.30
Volume mapper Projected tetra


VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties Display Information
Apply Reset Delete
Contour
Contour By
Morse field
Compute Normals
Compute Gradients
Compute Scalars
Isosurfaces
Value Range: [0, 100]
50
Delete
Delete All
New Value
New Range
Scientific

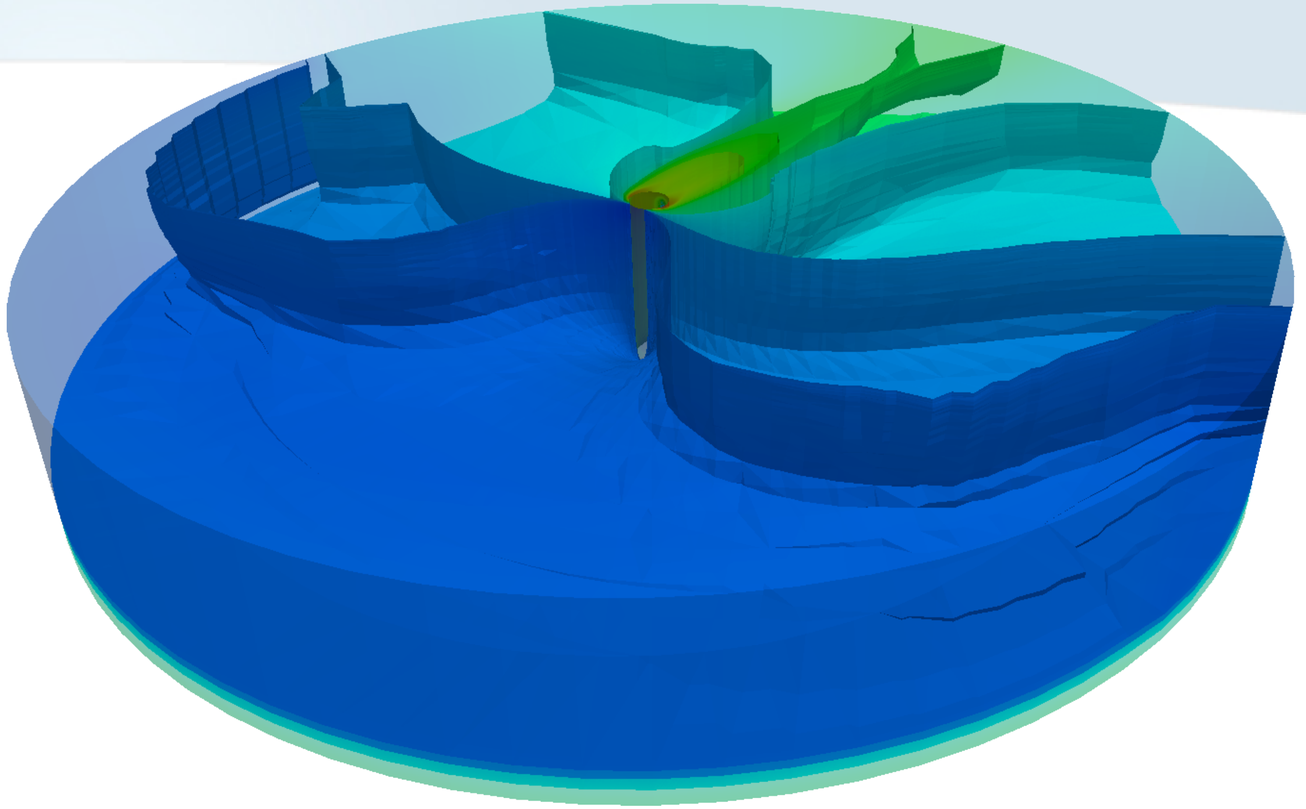

VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties Display Information
Apply Reset Delete
Contour
Contour By
Morse field
Compute Normals
Compute Gradients
Compute Scalars
Isosurfaces
Value Range: [0, 100]
50
60
Delete
Delete All
New Value
New Range
Scientific


VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties Display Information
Apply Reset Delete
Contour
Contour By
Morse field
Compute Normals
Compute Gradients
Compute Scalars
Isosurfaces
Value Range: [0, 100]
50
60
80
Delete
Delete All
New Value
New Range
Scientific


VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties Display Information
Apply Reset Delete
Contour
Contour By
Morse field
Compute Normals
Compute Gradients
Compute Scalars
Isosurfaces
Value Range: [0, 100]
50
60
80
90
Delete
Delete All
New Value
New Range
Scientific

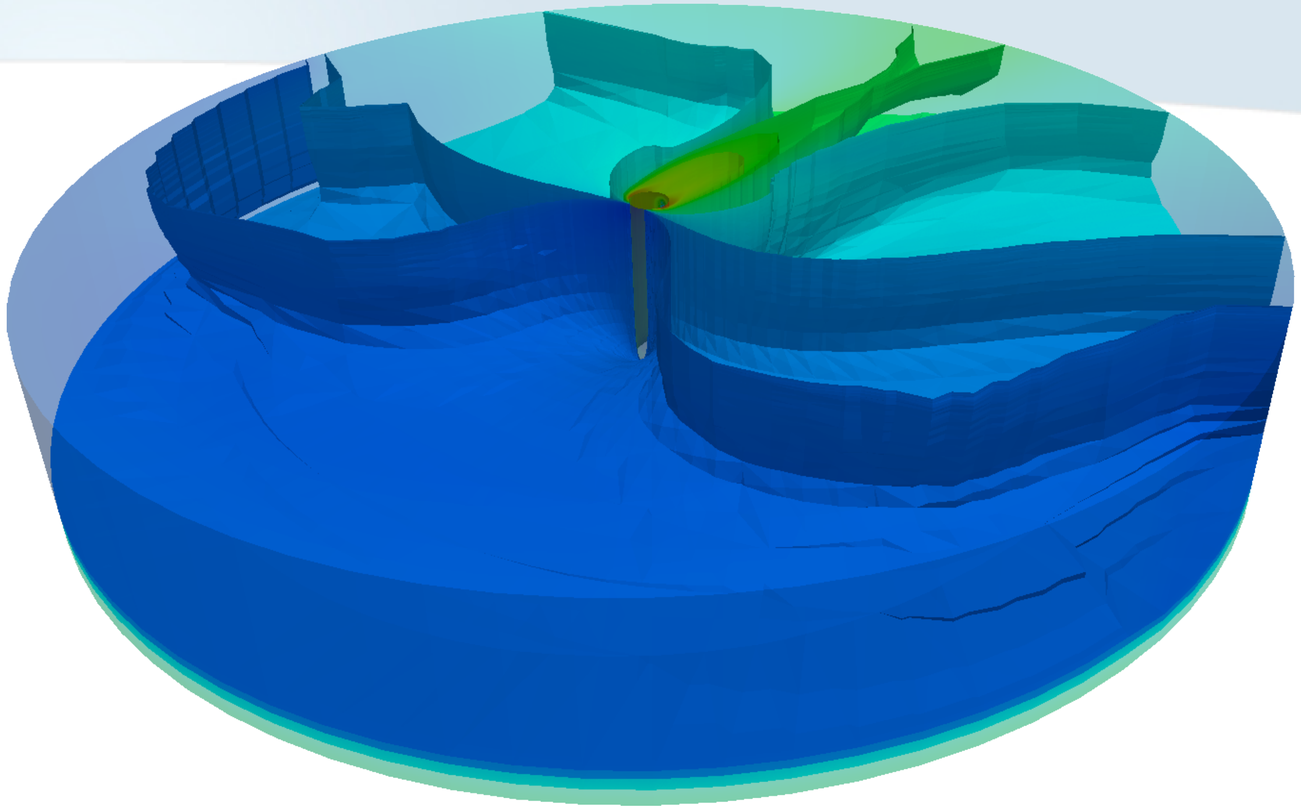
Fast level set extraction

- Marching xxxxxx
  - Visit all the unit cells
  - Per-cell extraction



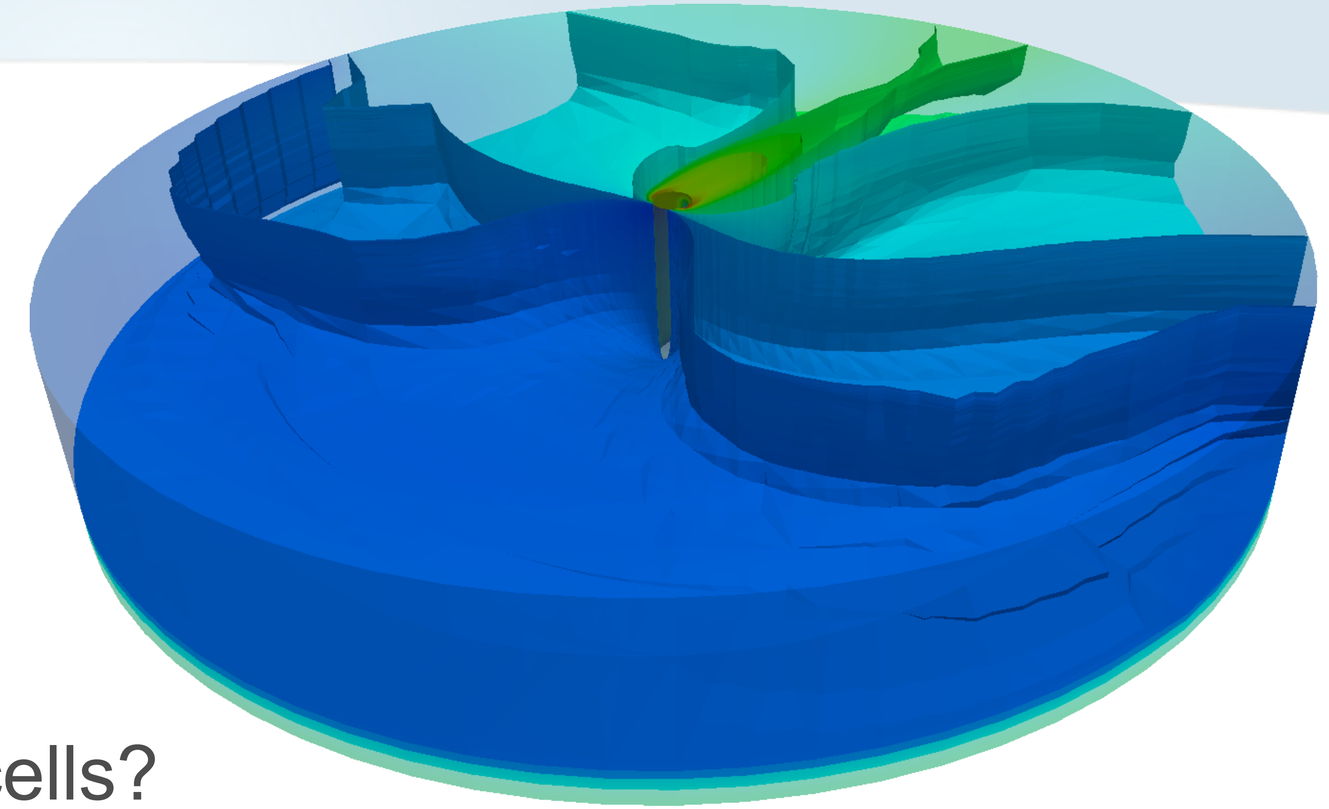
Fast level set extraction

- Marching xxxxxx
  - Visit all the unit cells
  - Per-cell extraction
- Speeding up extraction

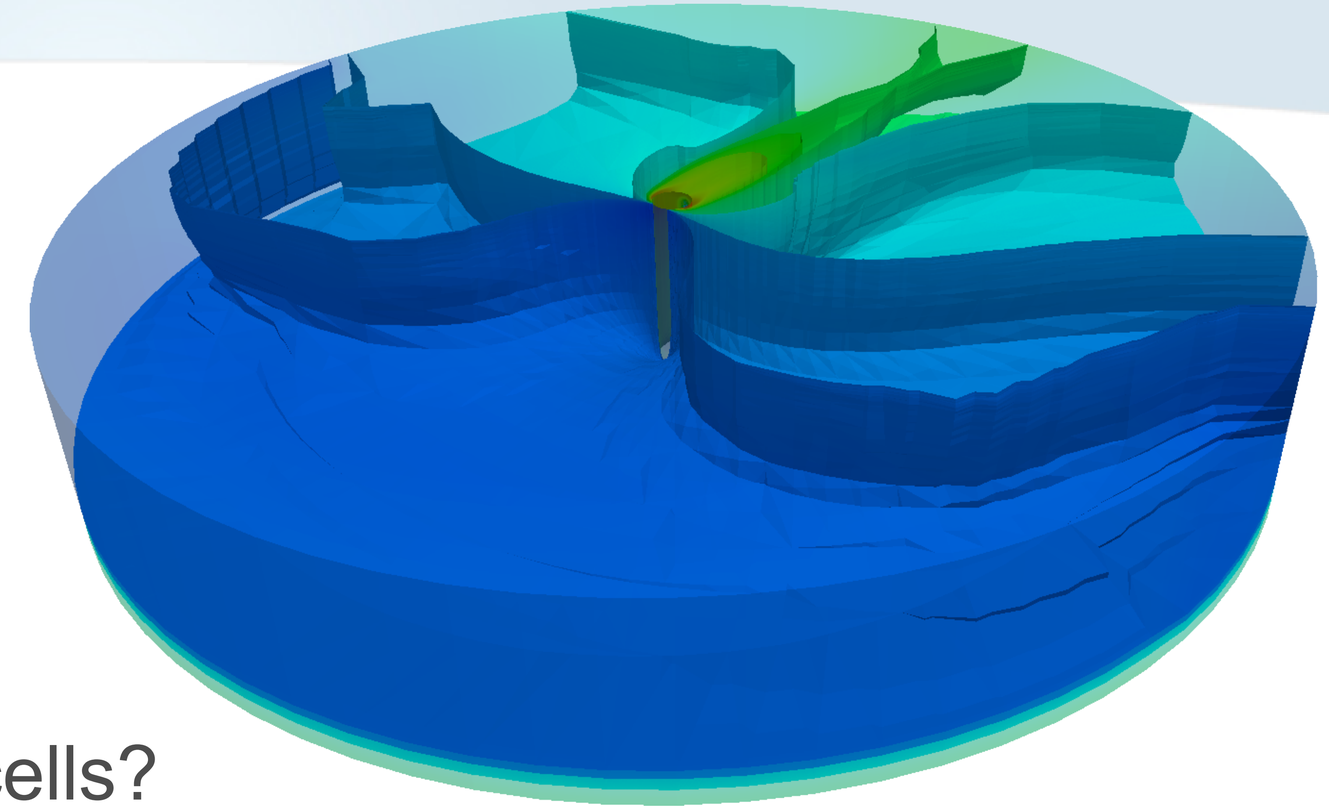


Fast level set extraction

- Marching xxxxxx
  - Visit all the unit cells
  - Per-cell extraction
- Speeding up extraction
  - Do we need to visit **ALL** the cells?

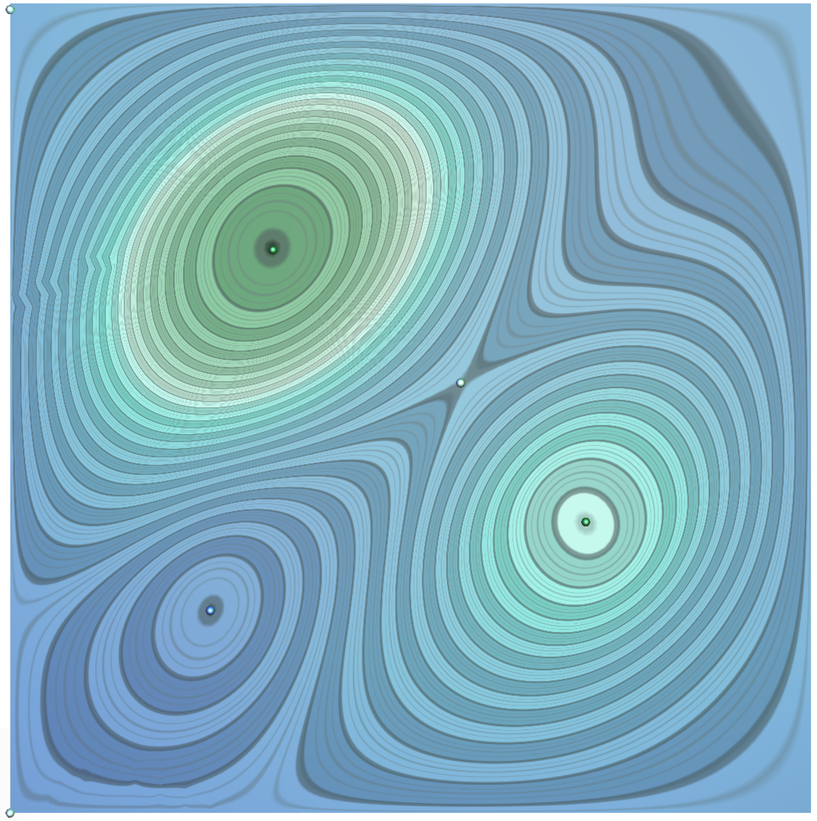


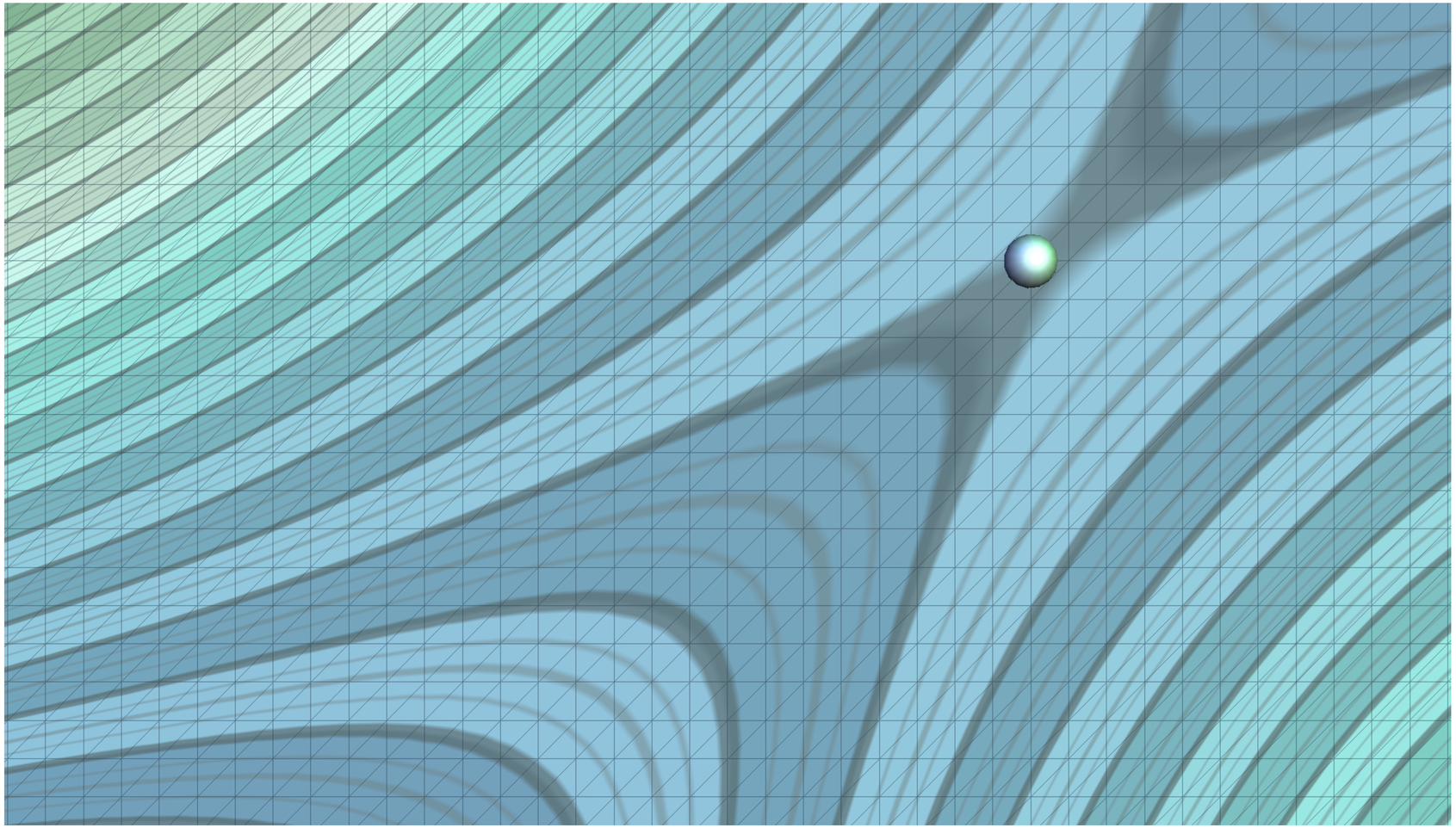
Fast level set extraction



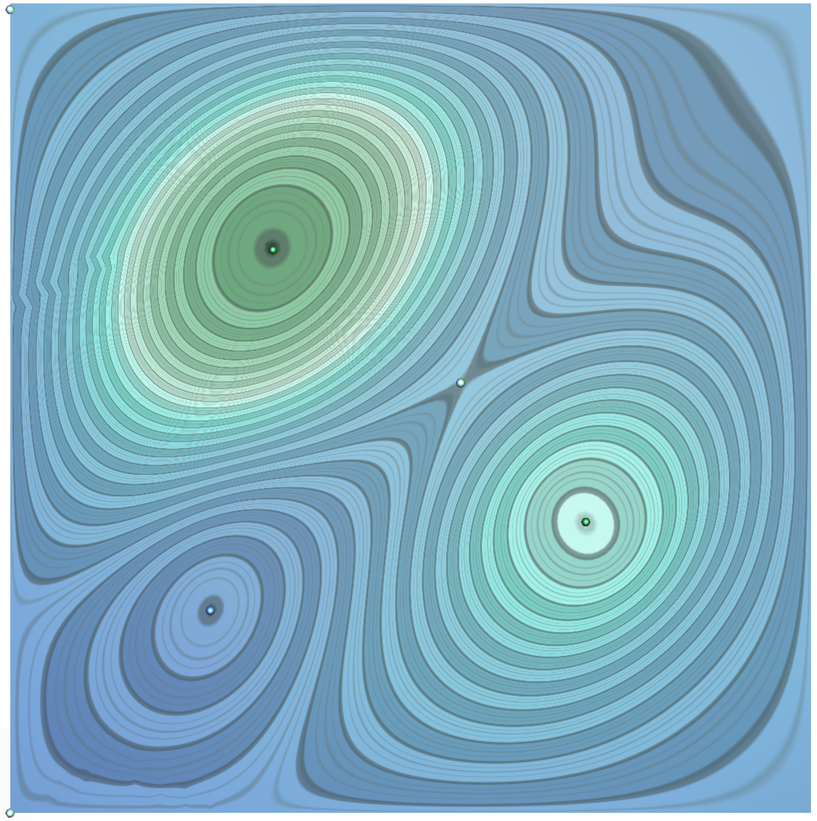
- Marching xxxxxx
  - Visit all the unit cells
  - Per-cell extraction
- Speeding up extraction
  - Do we need to visit **ALL** the cells?
  - Can't we just visit only the cells that are needed?

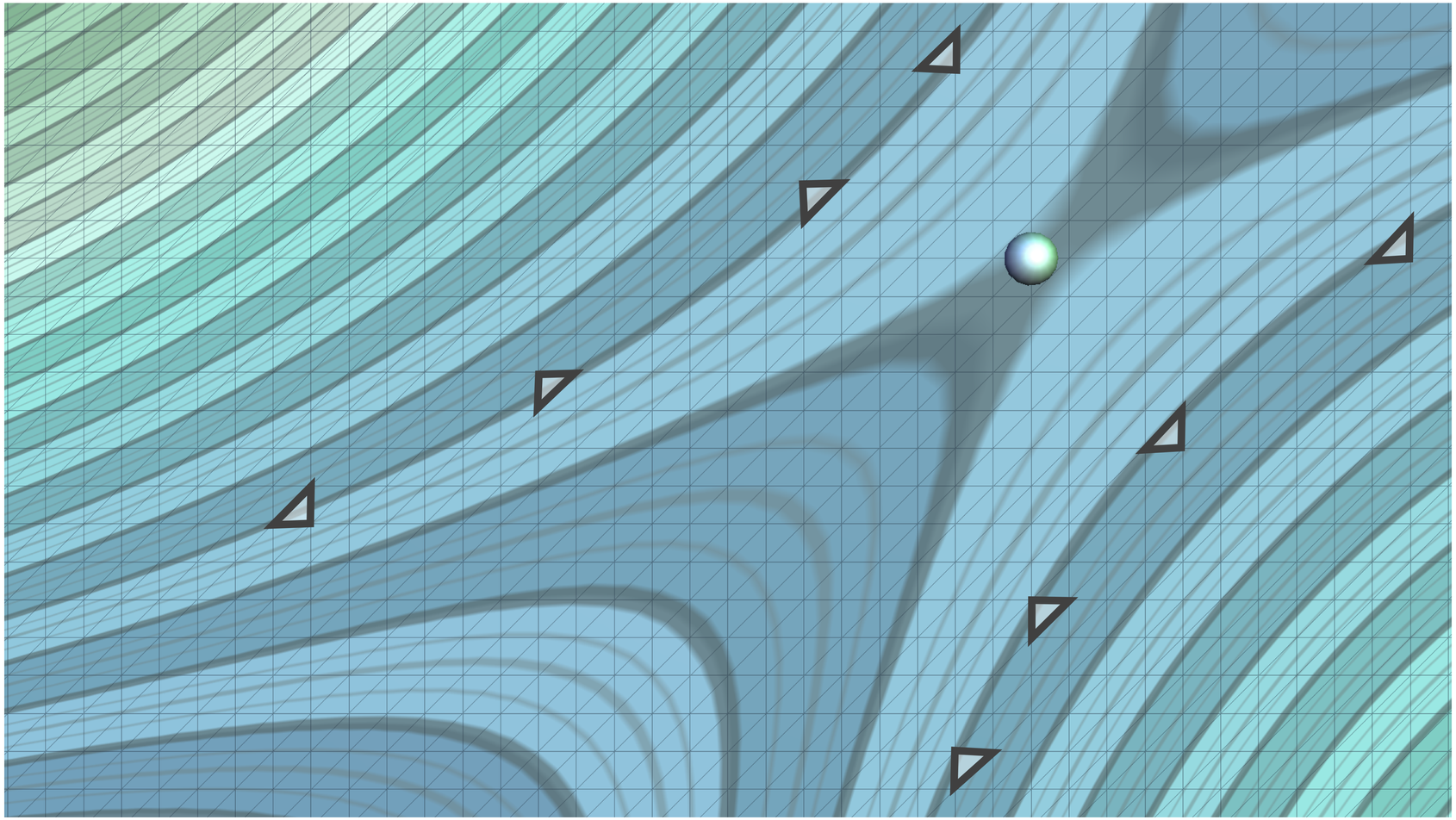
Seeds for level set extraction



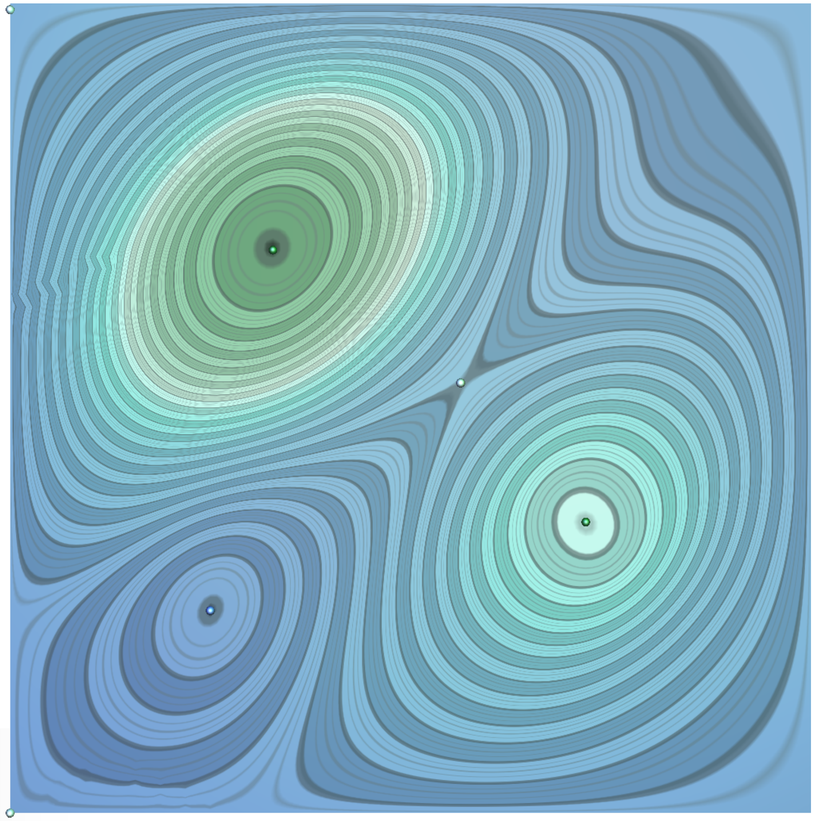


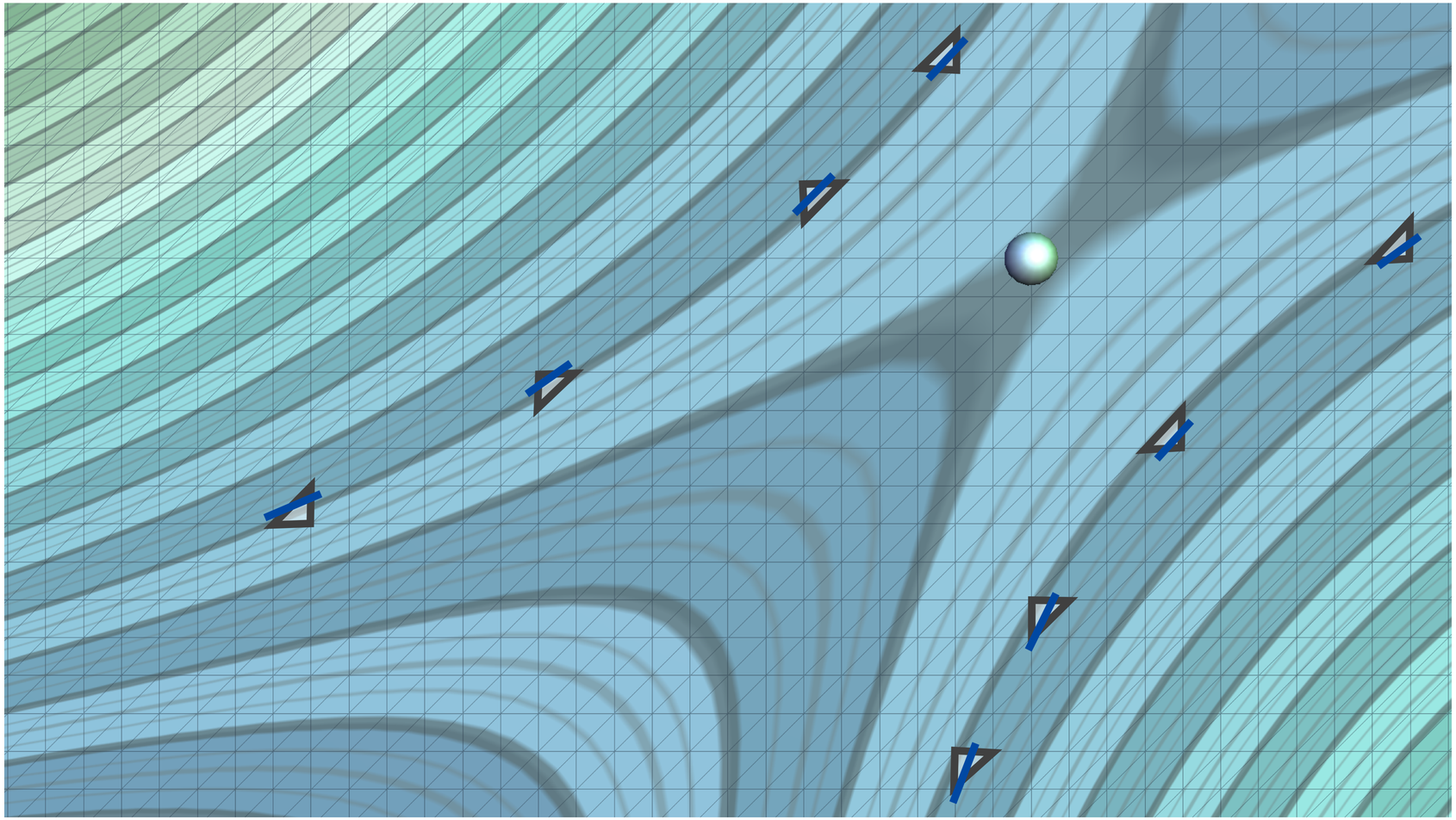
Seeds for level set extraction



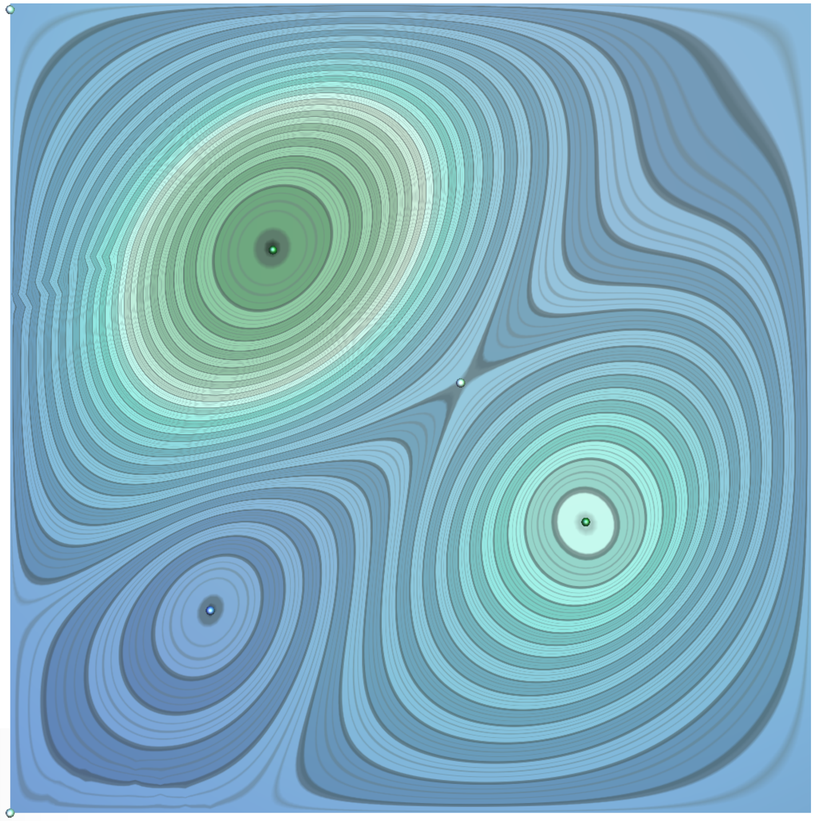


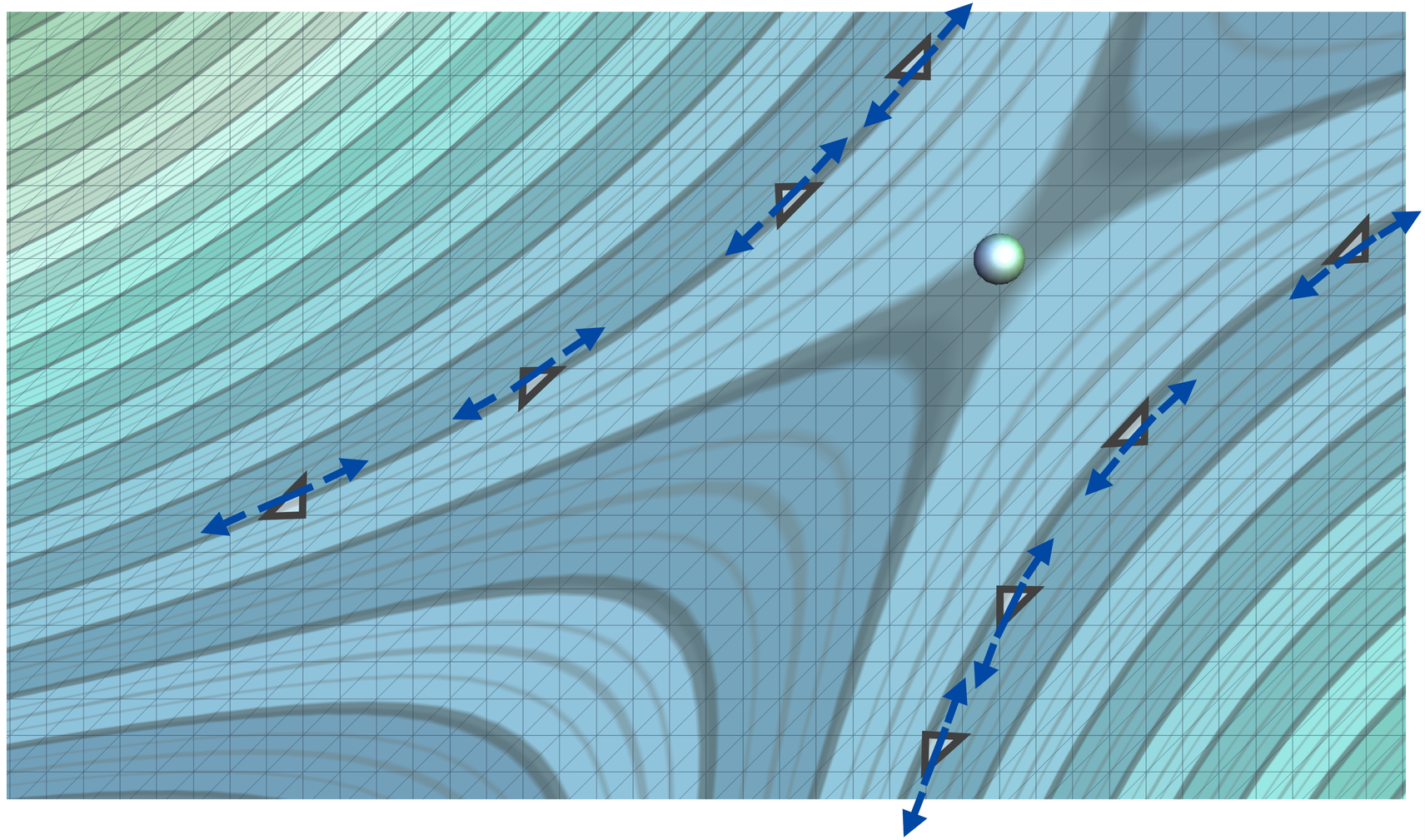
Seeds for level set extraction



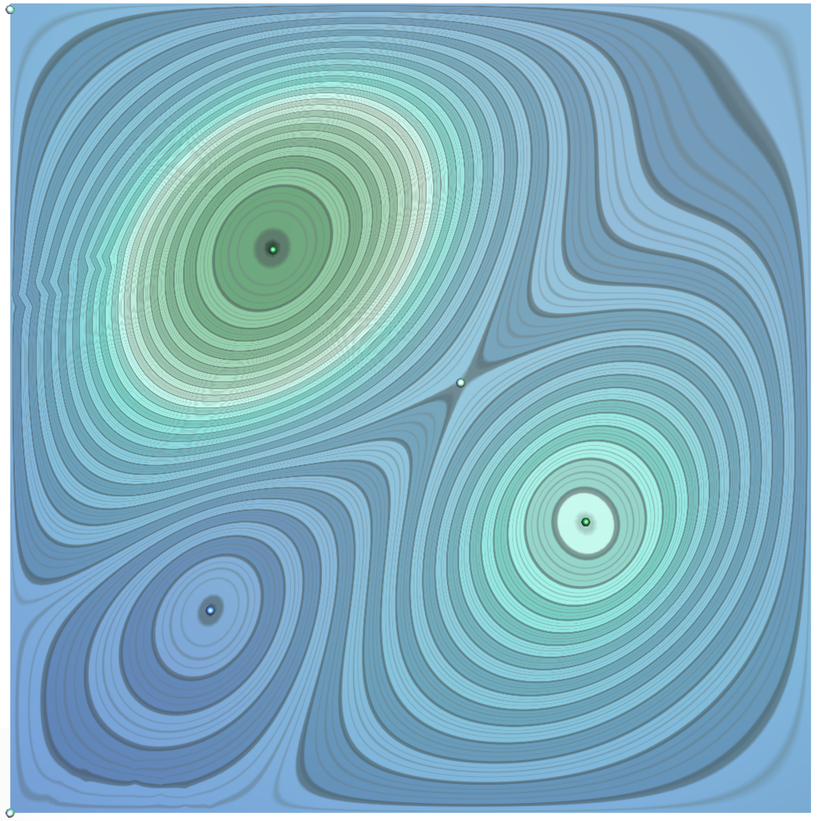


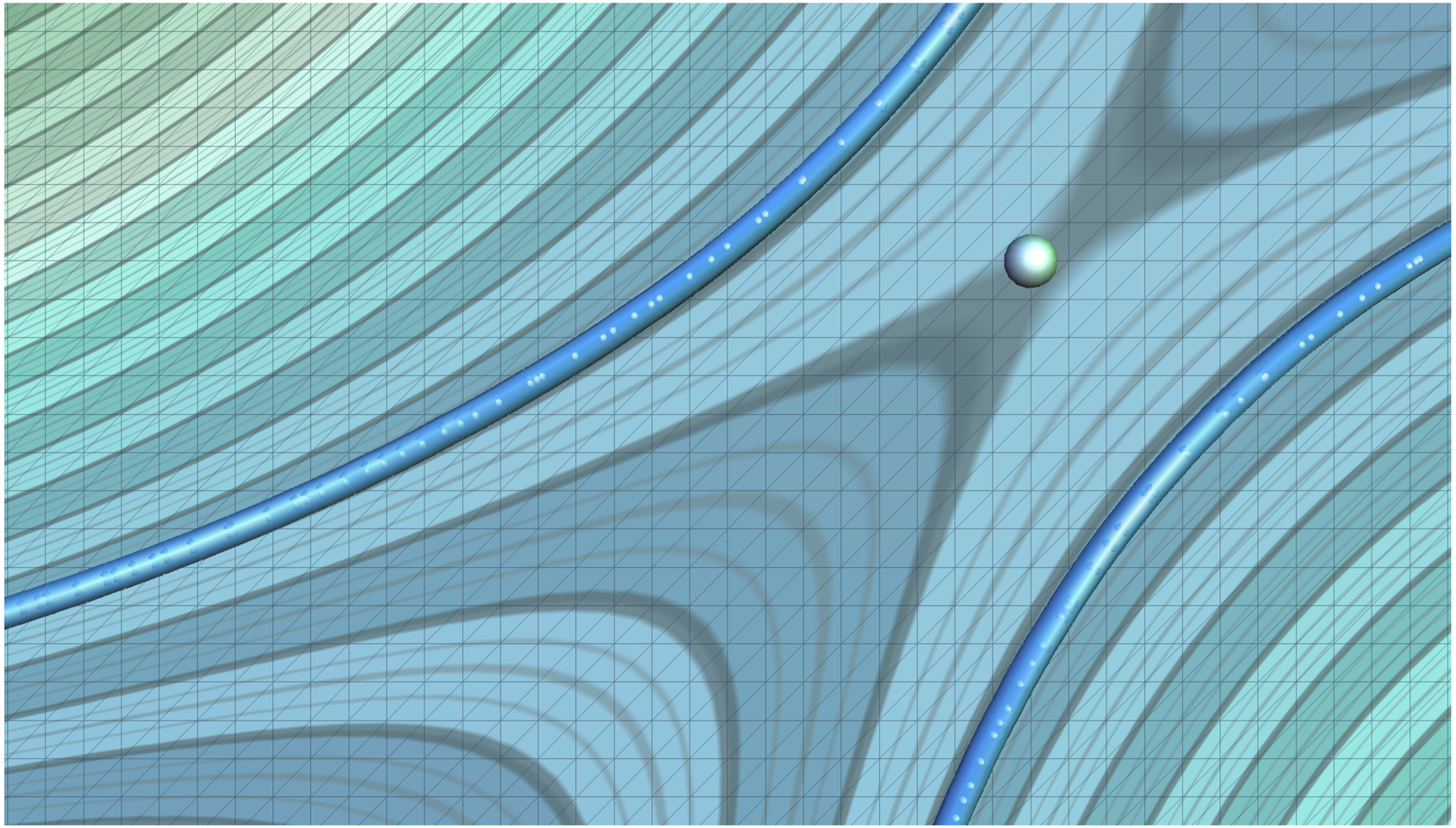
Seeds for level set extraction



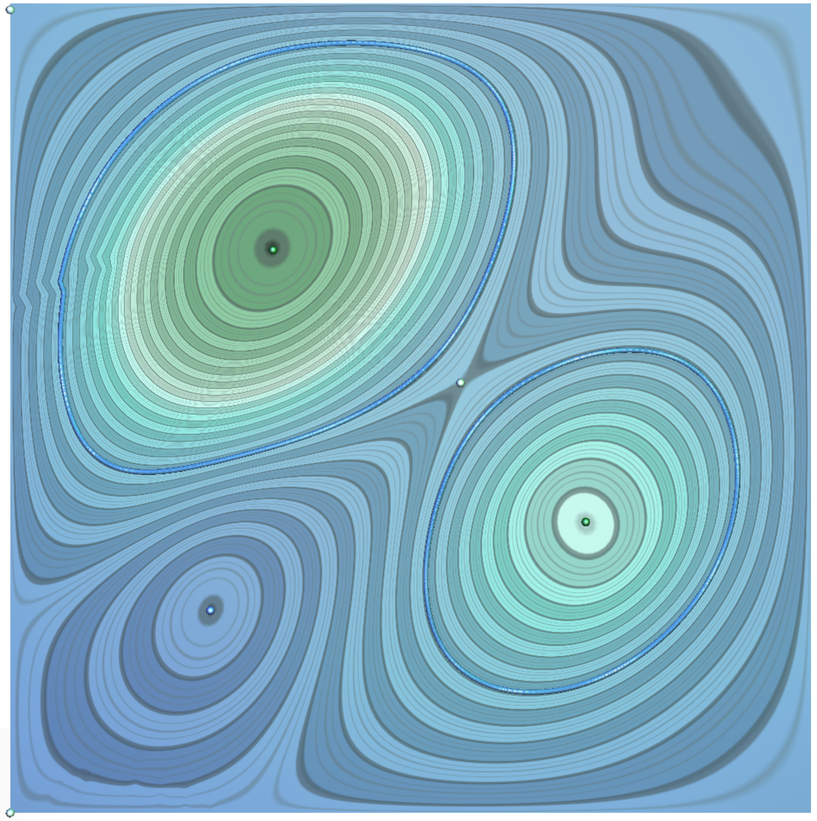


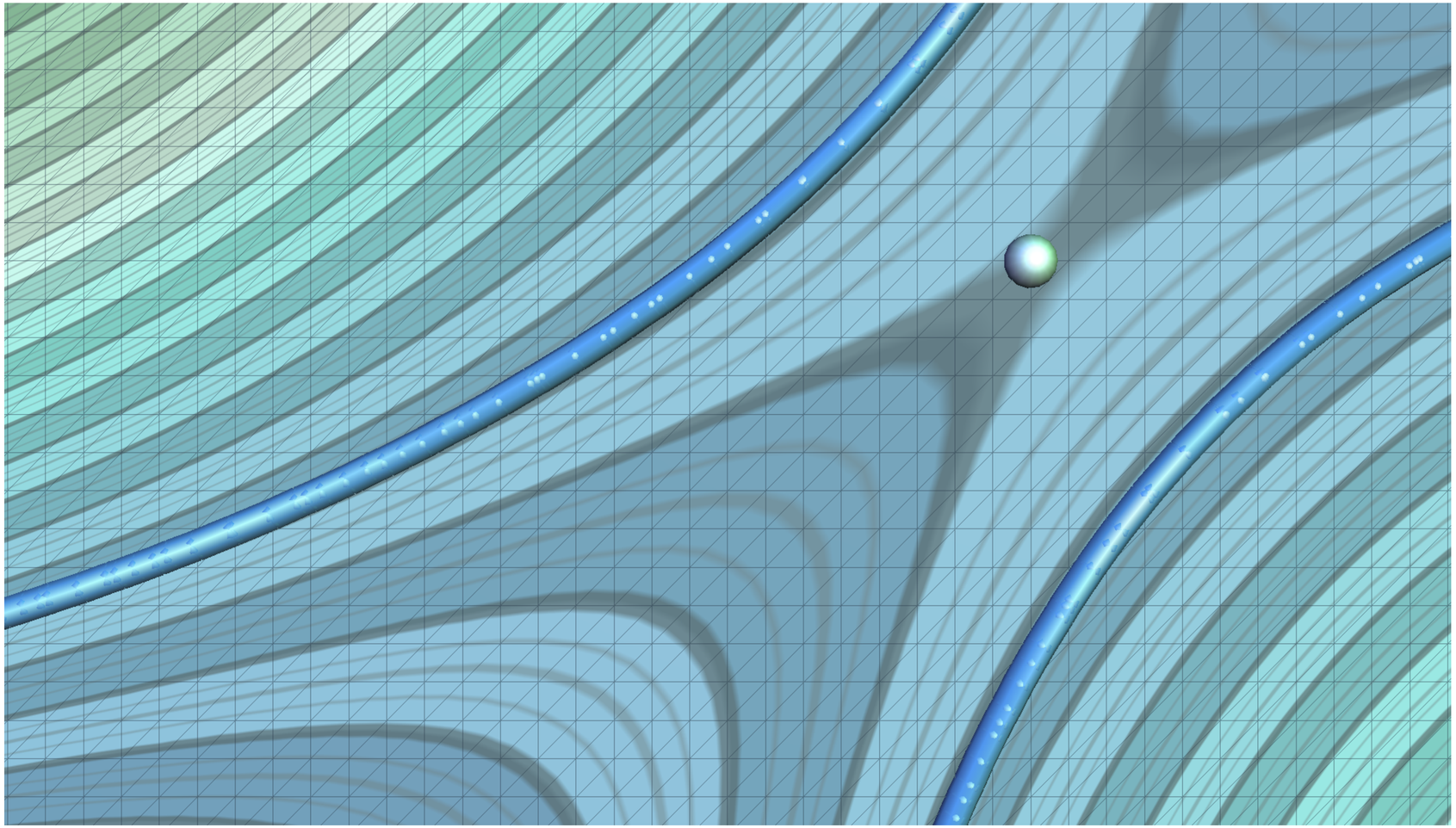
Seeds for level set extraction



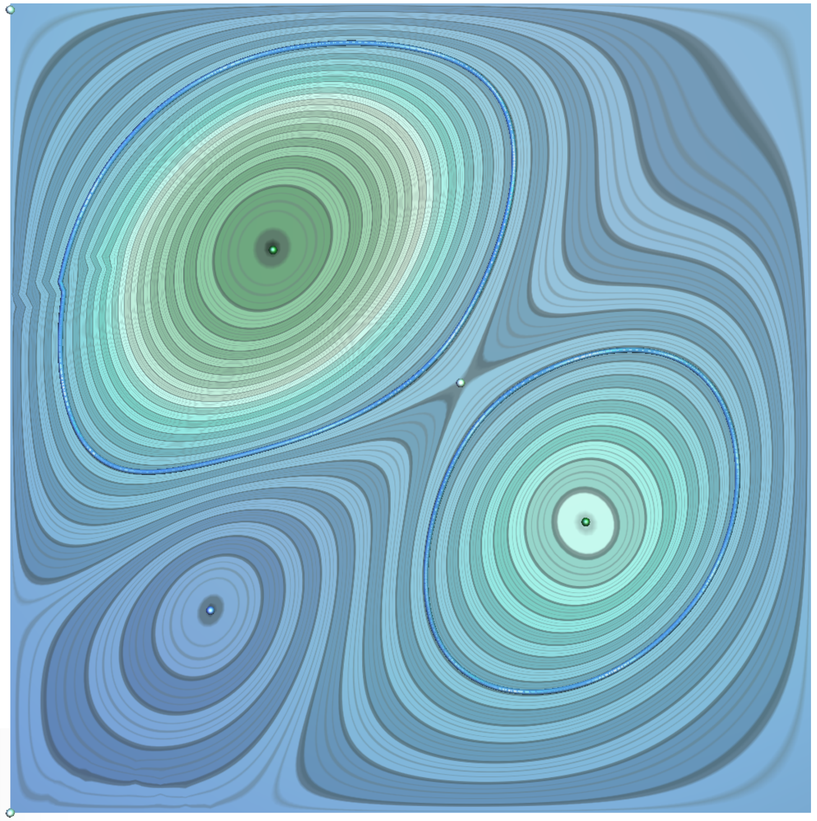


Seeds for level set extraction

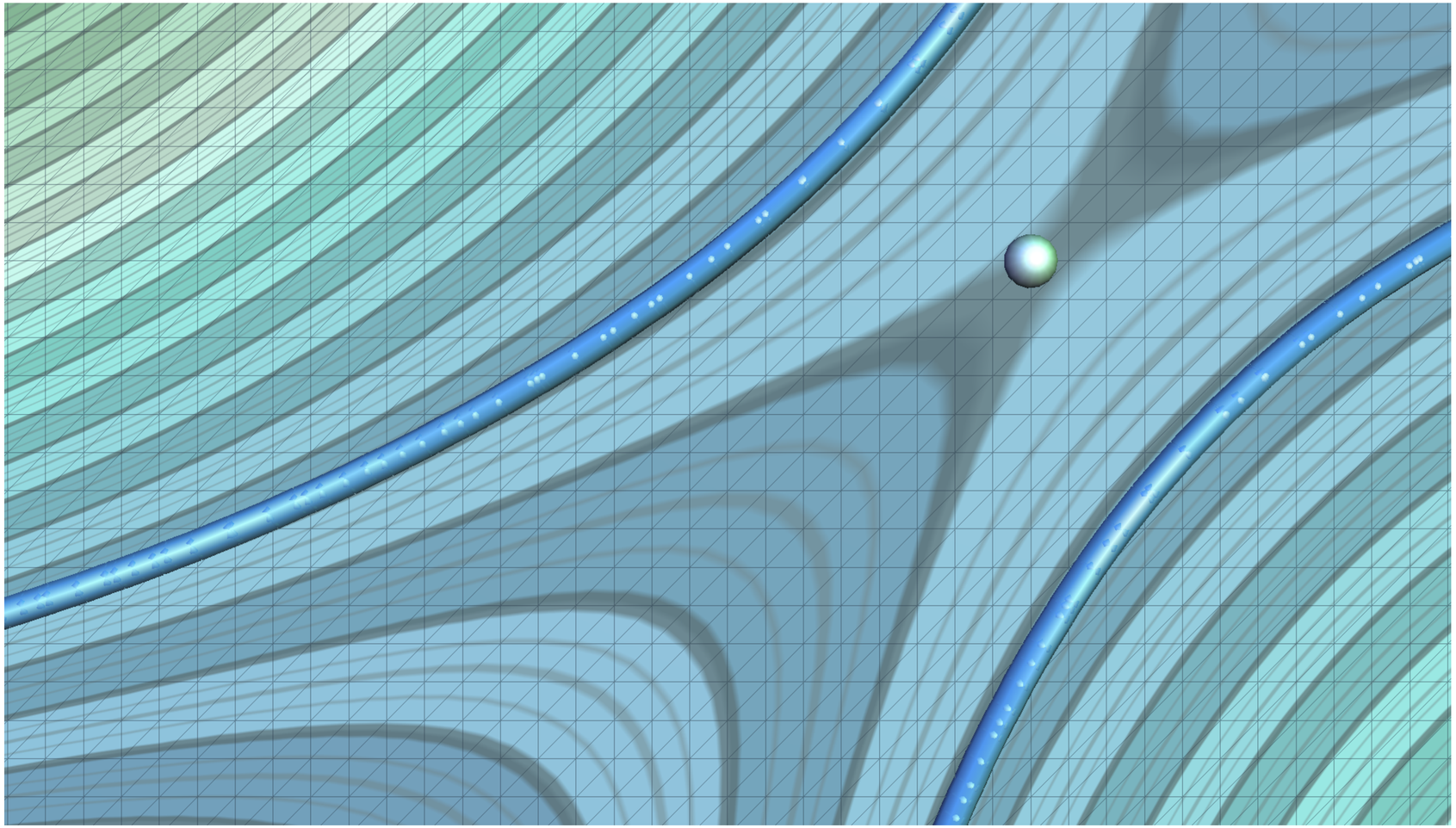




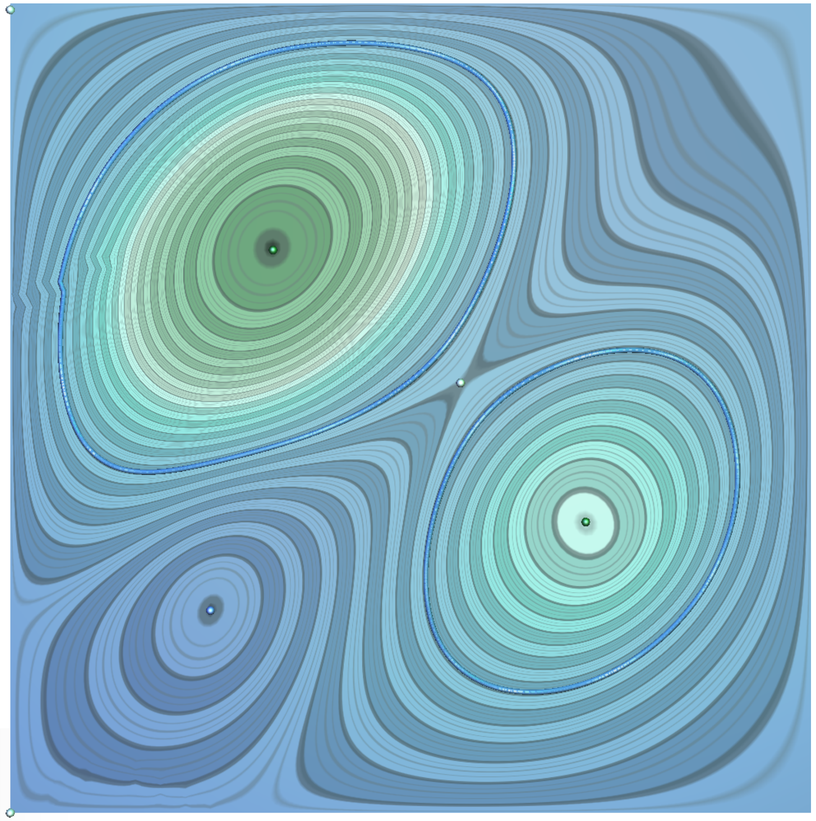
Seeds for level set extraction



Minimal number of seeds?

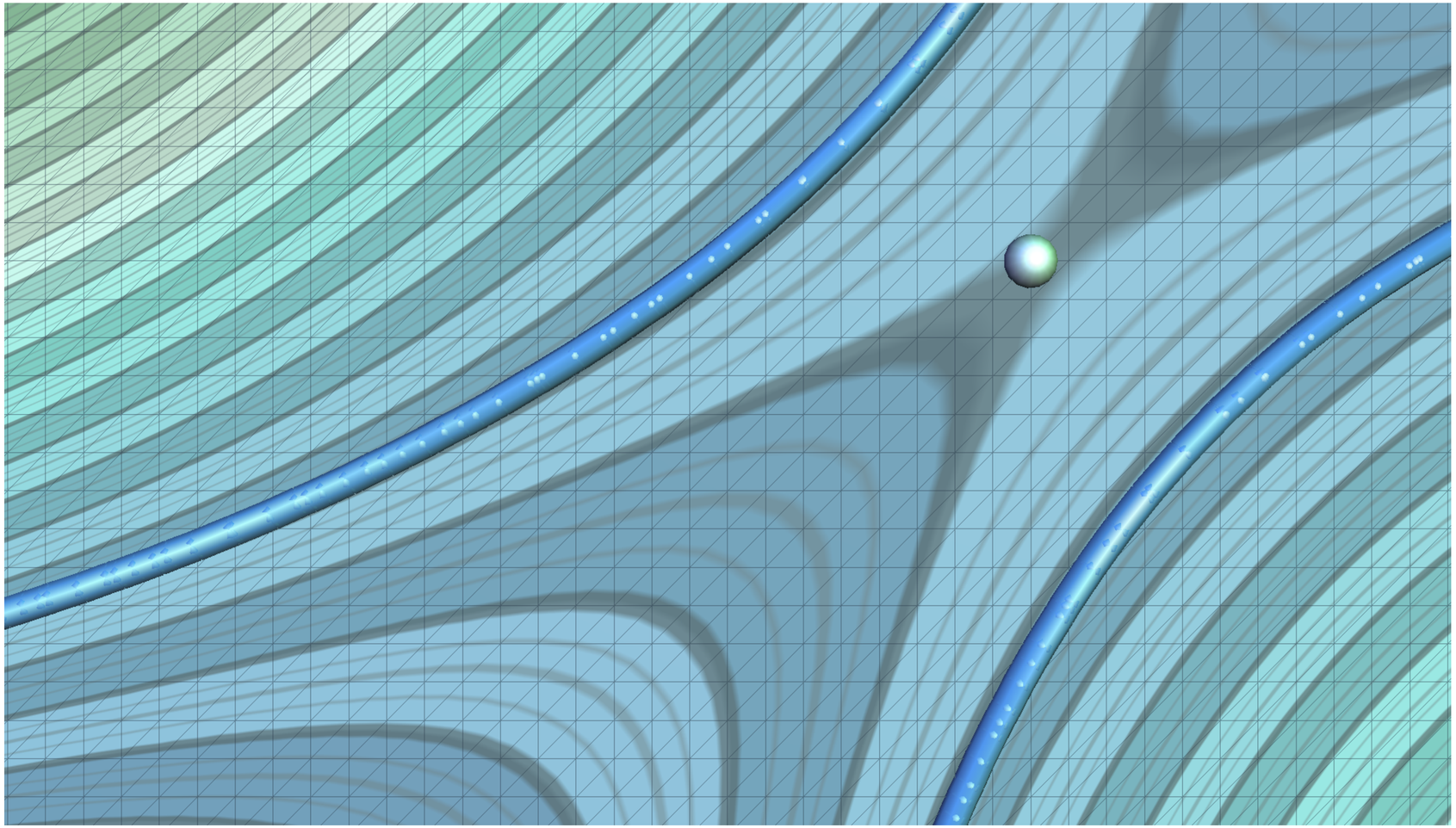


Seeds for level set extraction



Minimal number of seeds?

- **One per connected component of $f^{-1}(i)$**

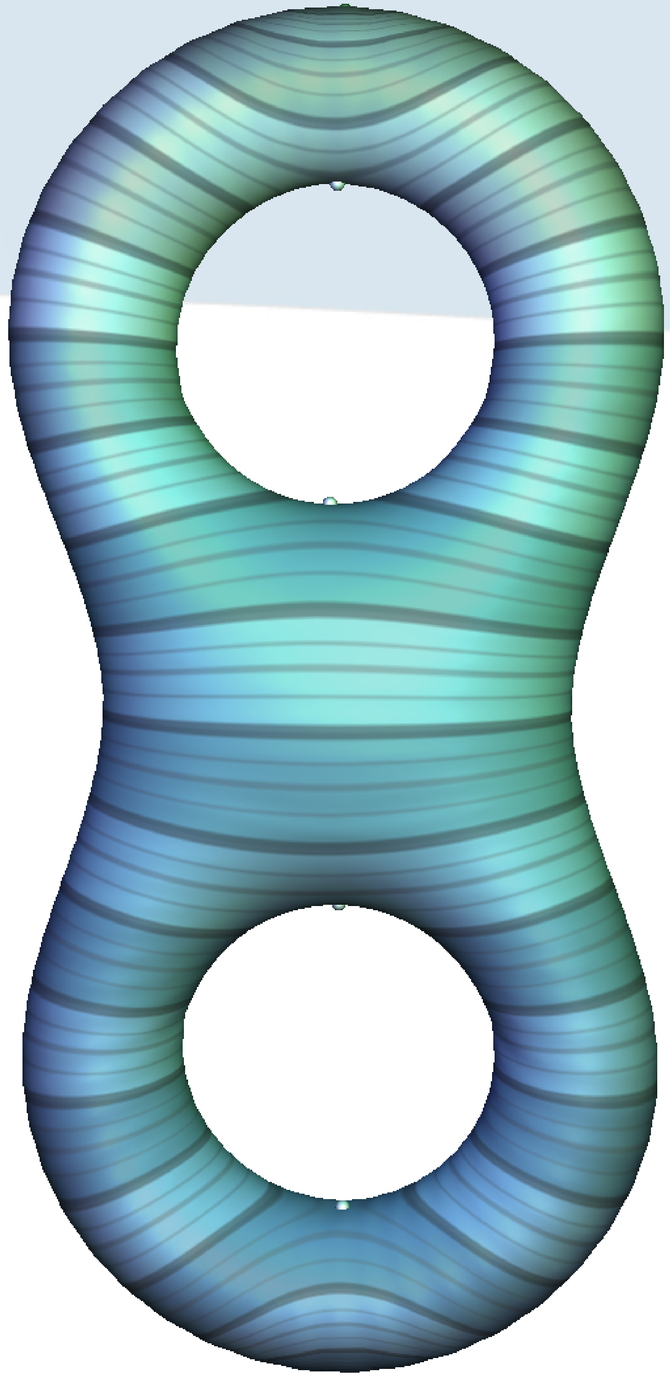


Indexing contours

- Notion of Reeb graph

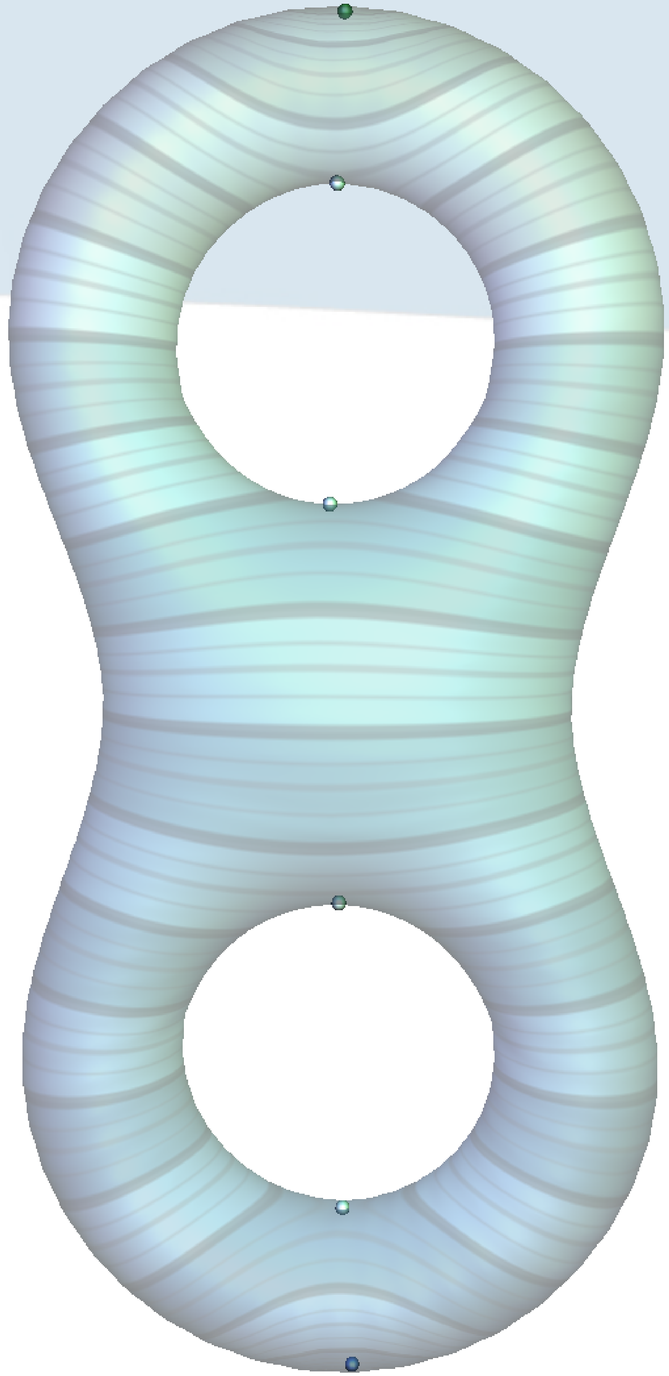
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function



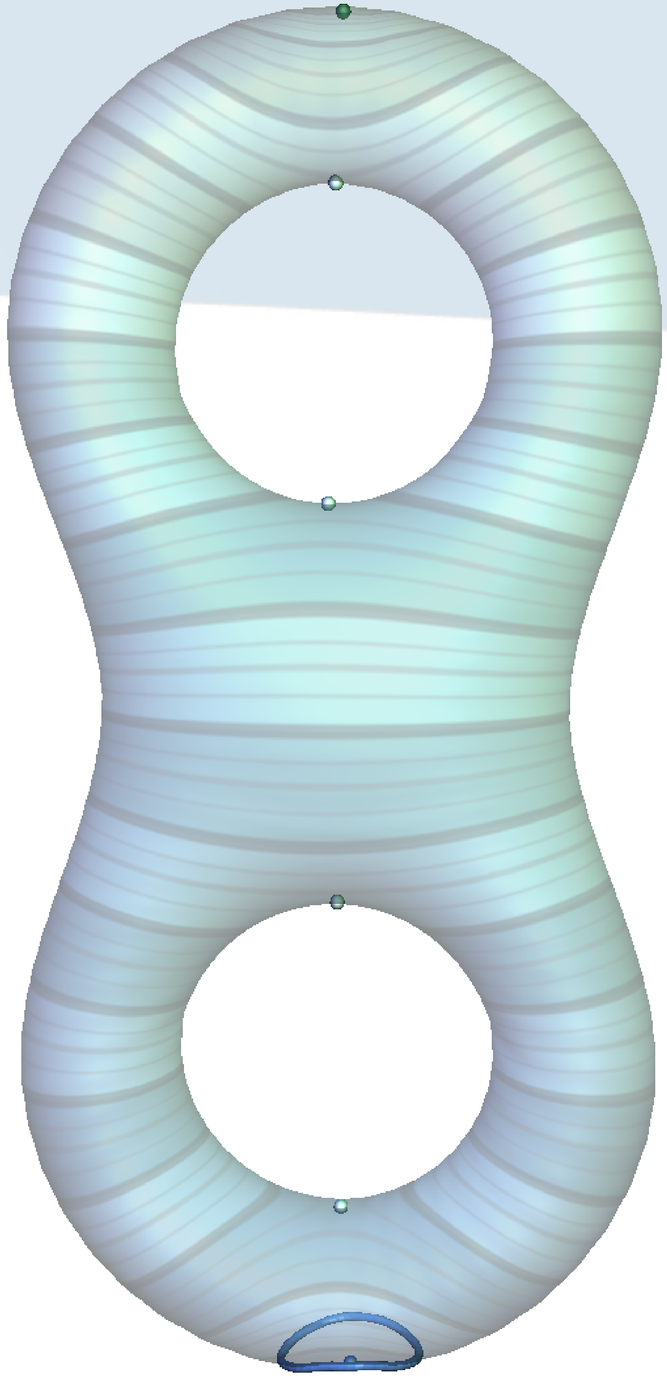
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function



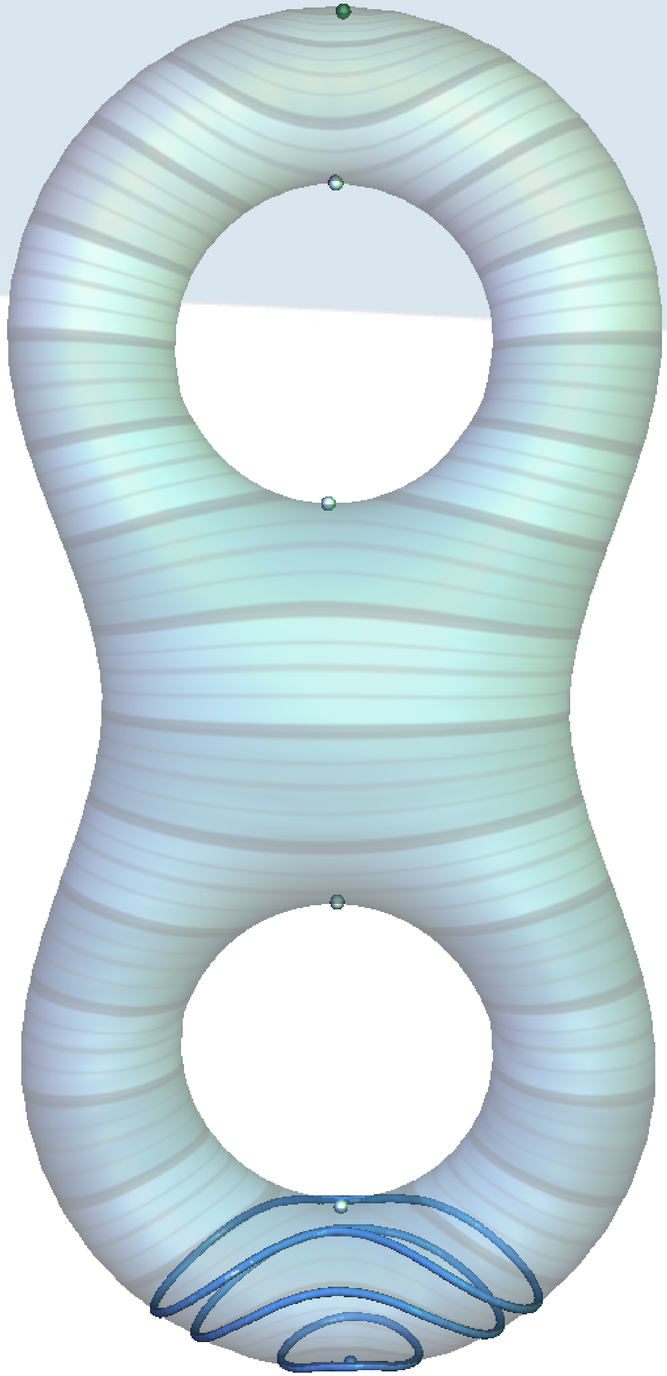
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function



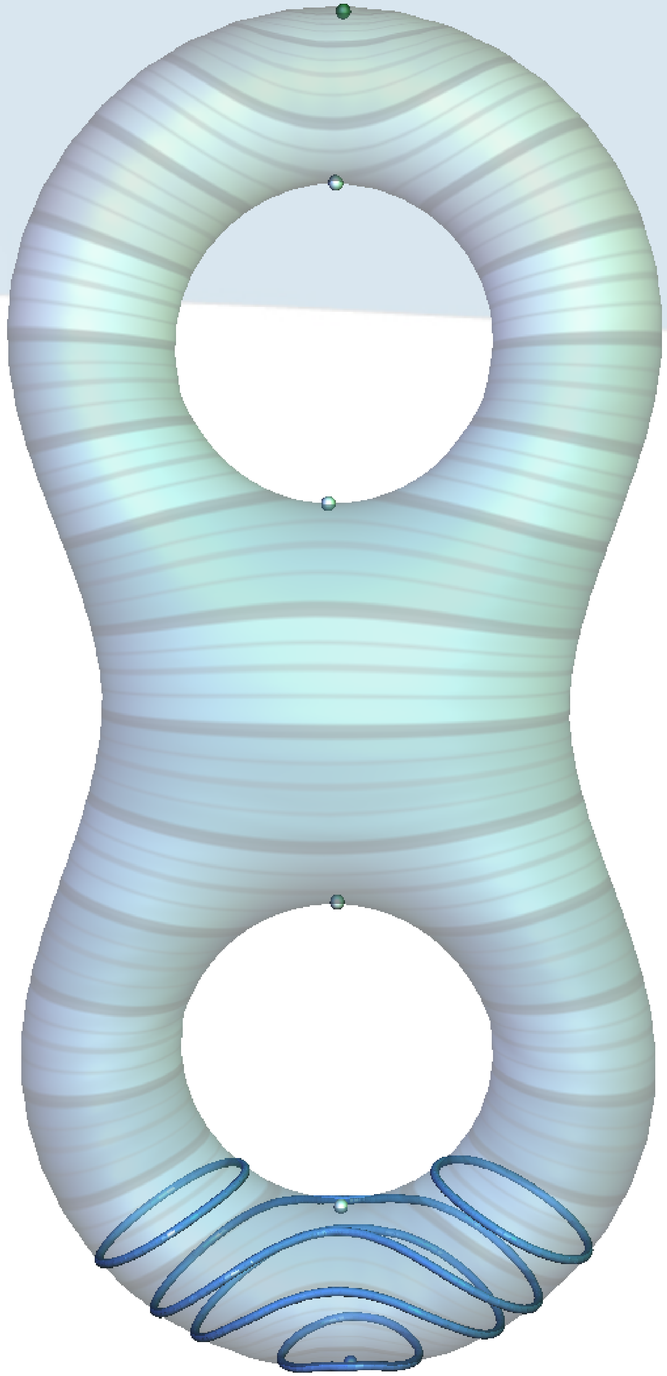
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function



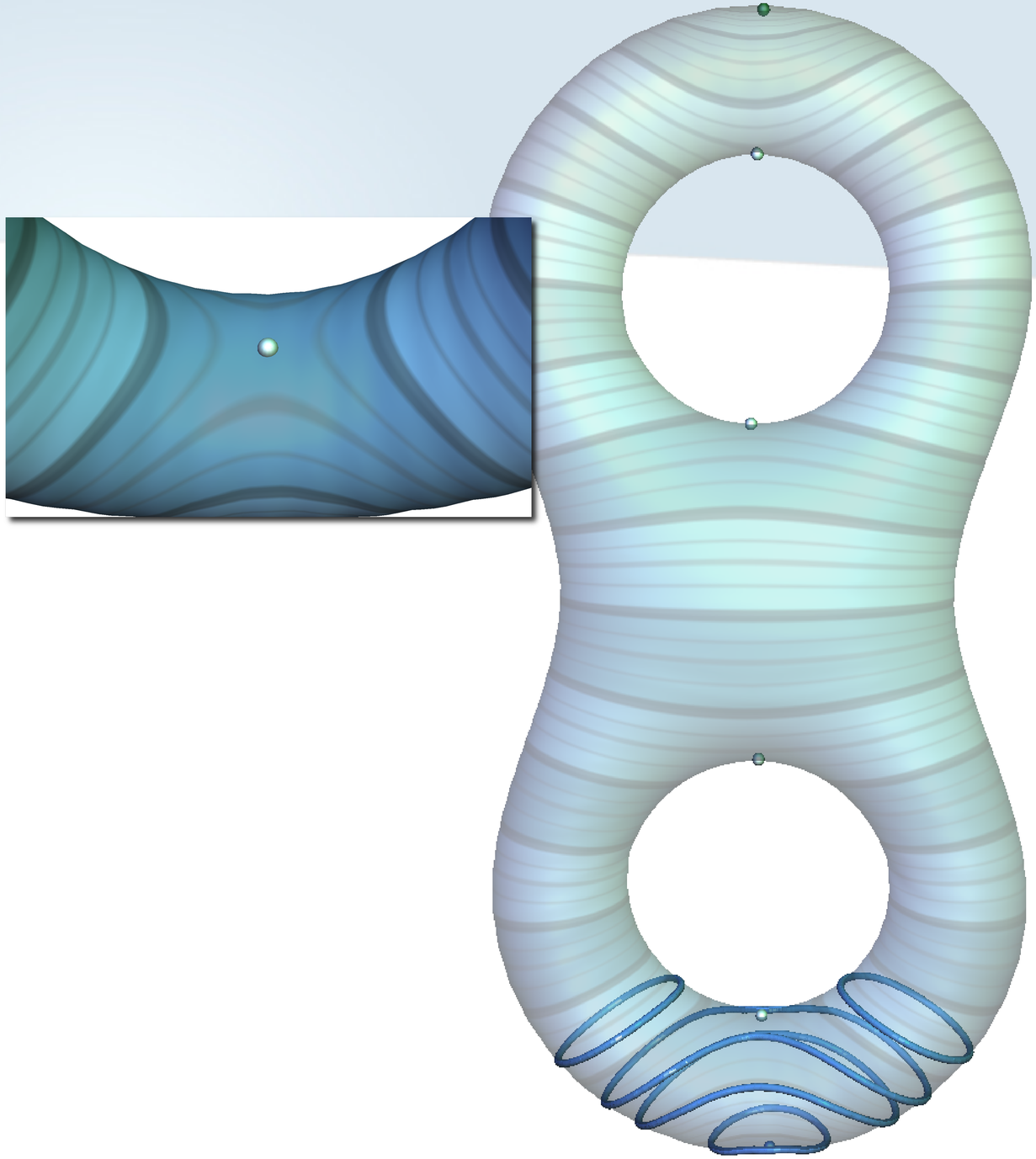
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function



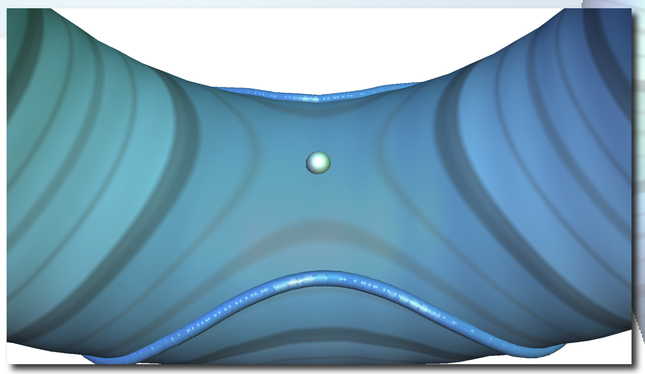
Indexing contours

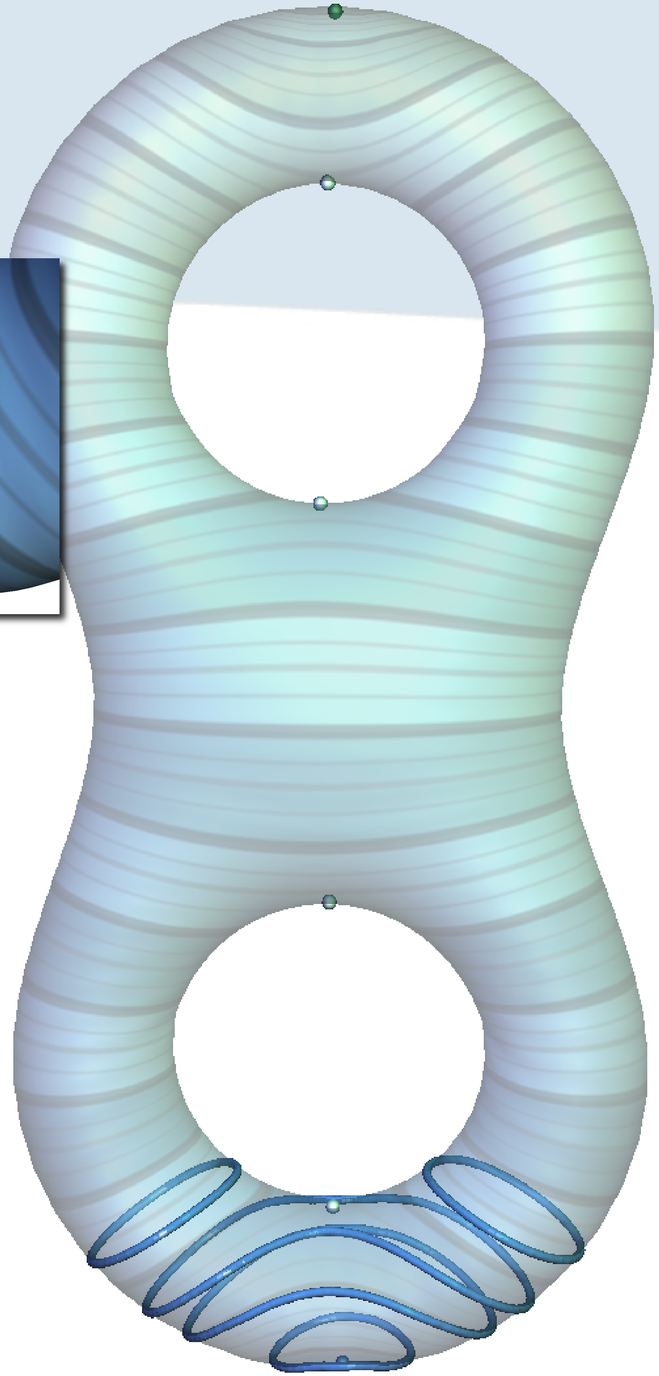
- Notion of Reeb graph
  - $f : \mathcal{M} \rightarrow \mathbb{R}$
  - *Morse* function



Indexing contours

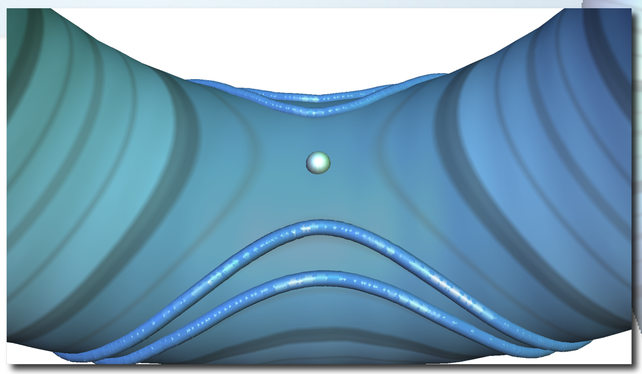
- Notion of Reeb graph
  - $f : \mathcal{M} \rightarrow \mathbb{R}$
  - *Morse* function

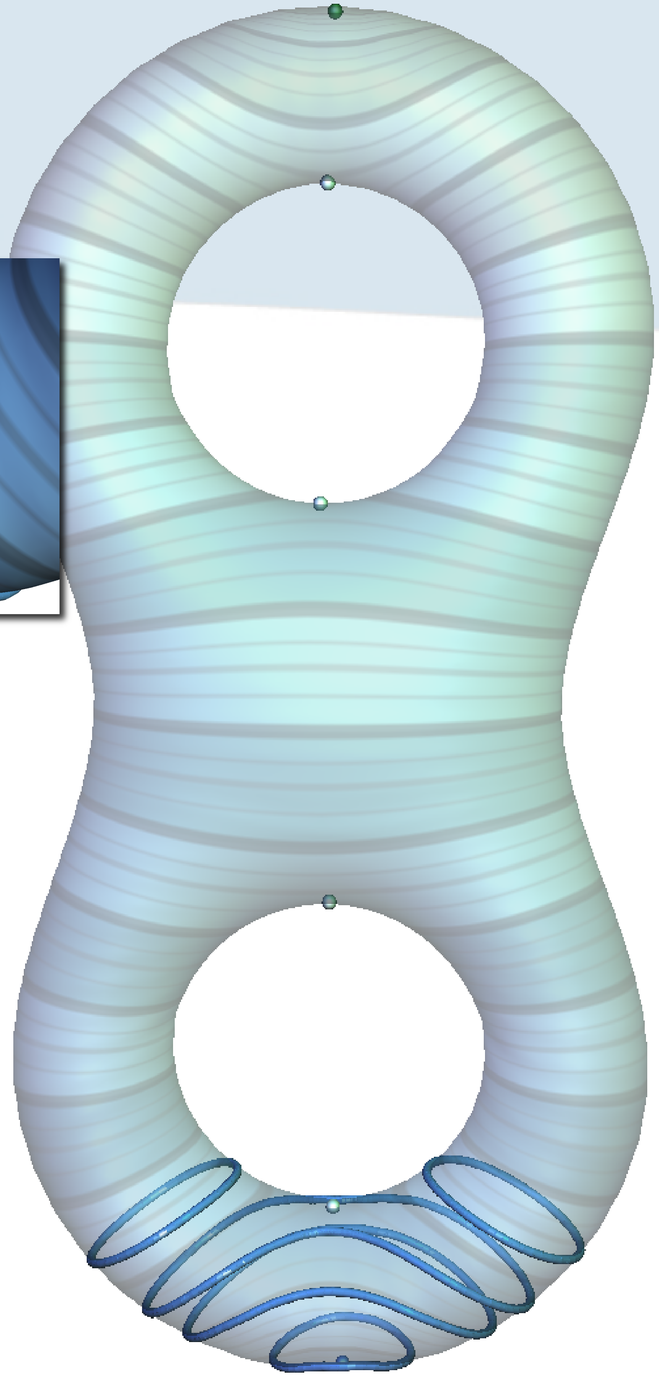




Indexing contours

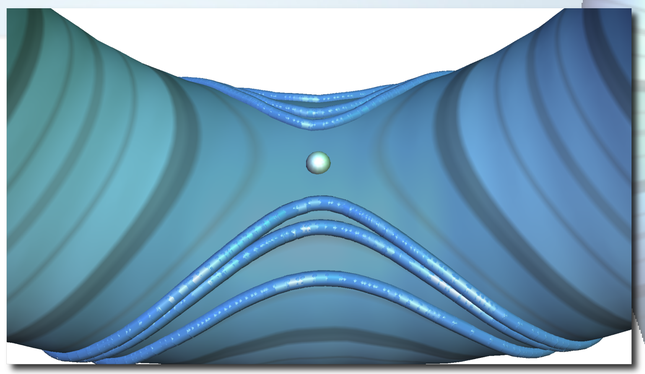
- Notion of Reeb graph
  - $f : \mathcal{M} \rightarrow \mathbb{R}$
  - *Morse* function

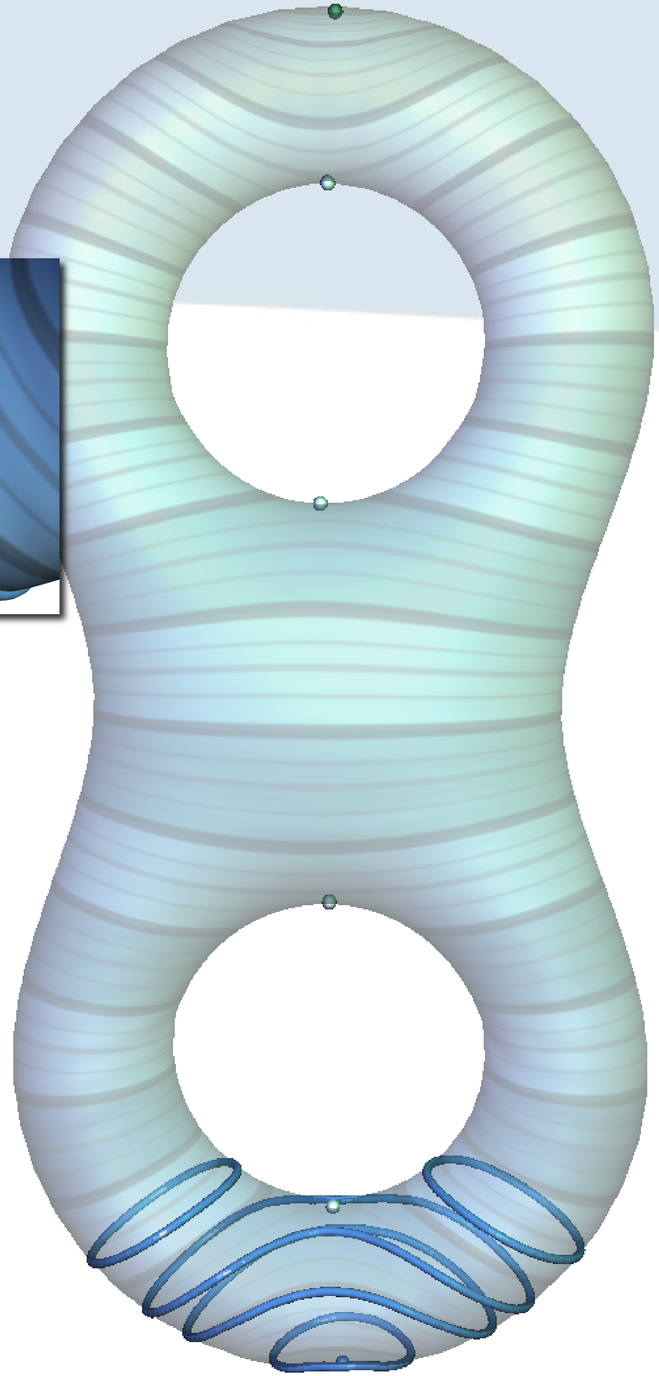




Indexing contours

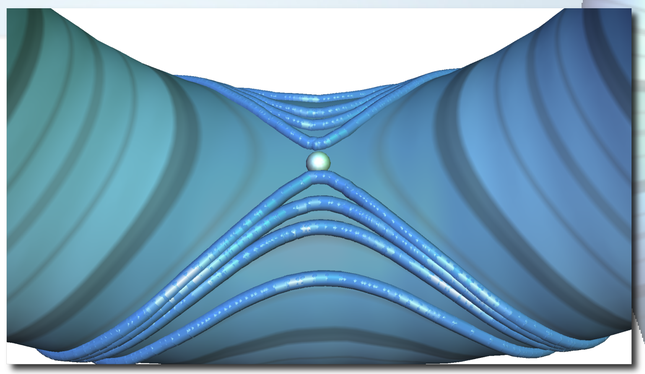
- Notion of Reeb graph
  - $f : \mathcal{M} \rightarrow \mathbb{R}$
  - *Morse* function

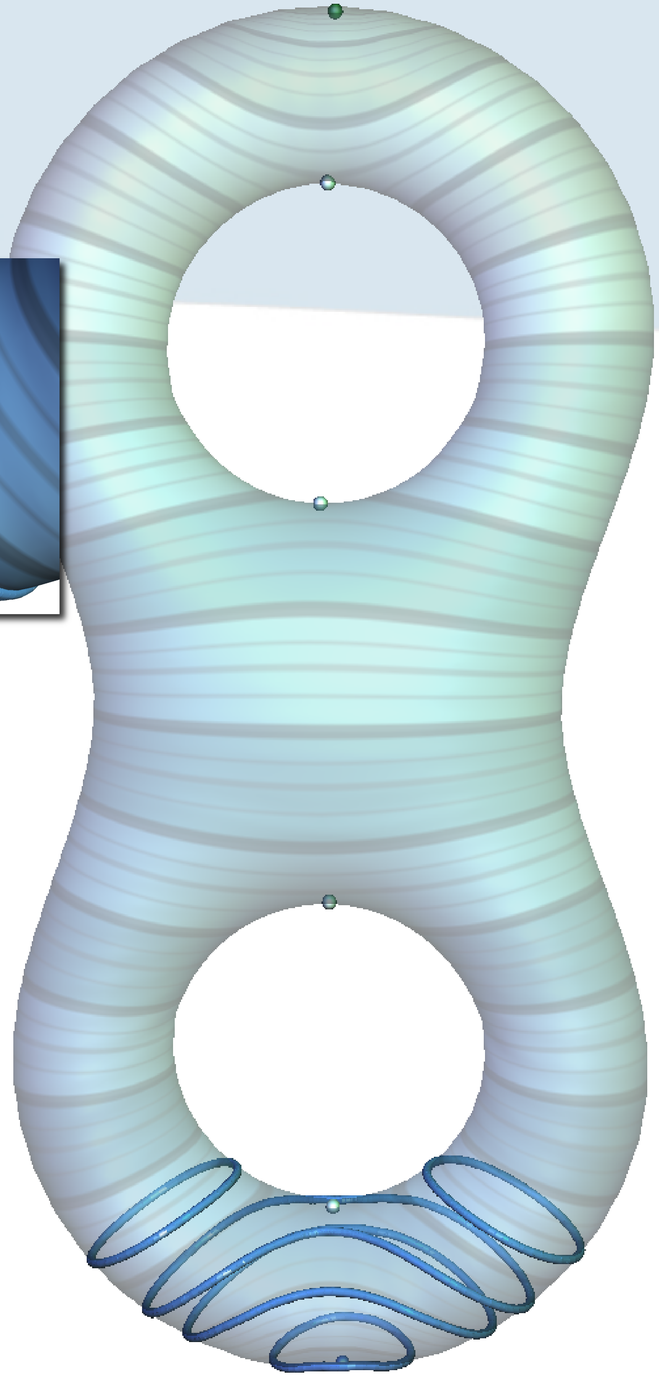




Indexing contours

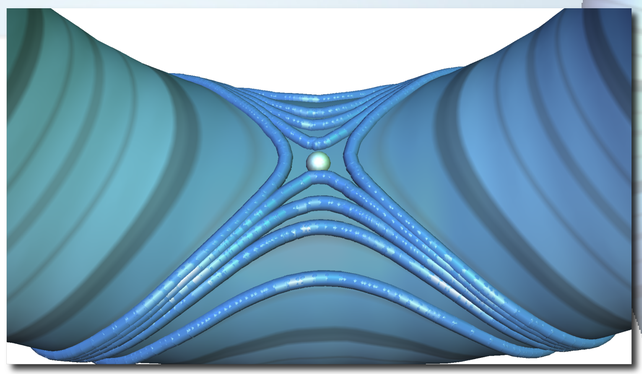
- Notion of Reeb graph
  - $f : \mathcal{M} \rightarrow \mathbb{R}$
  - *Morse* function

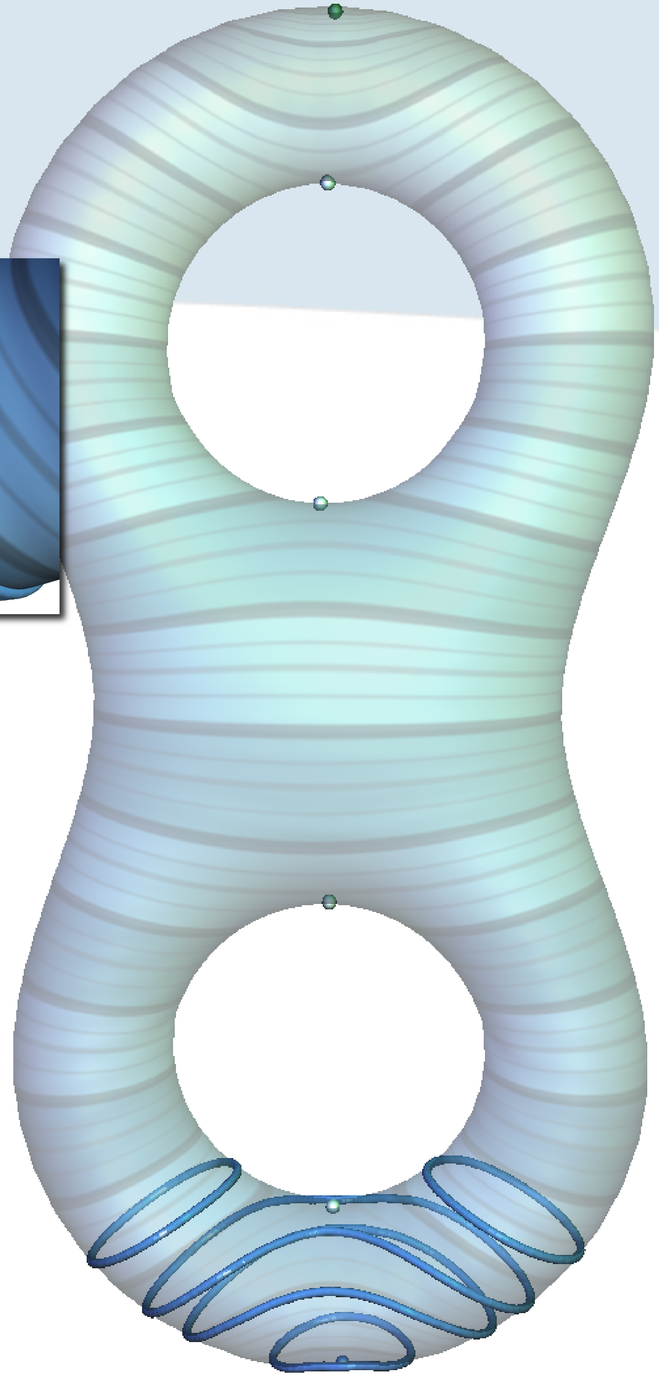




Indexing contours

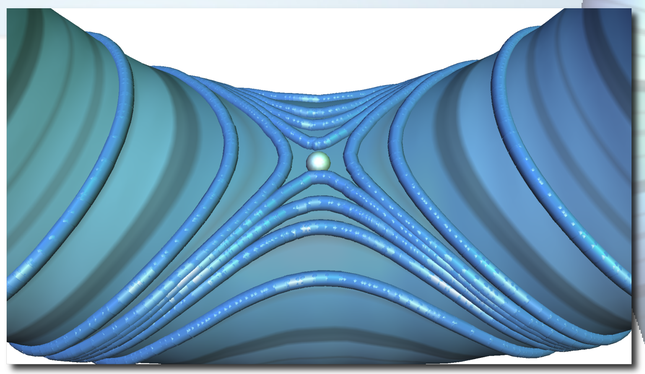
- Notion of Reeb graph
  - $f : \mathcal{M} \rightarrow \mathbb{R}$
  - *Morse* function





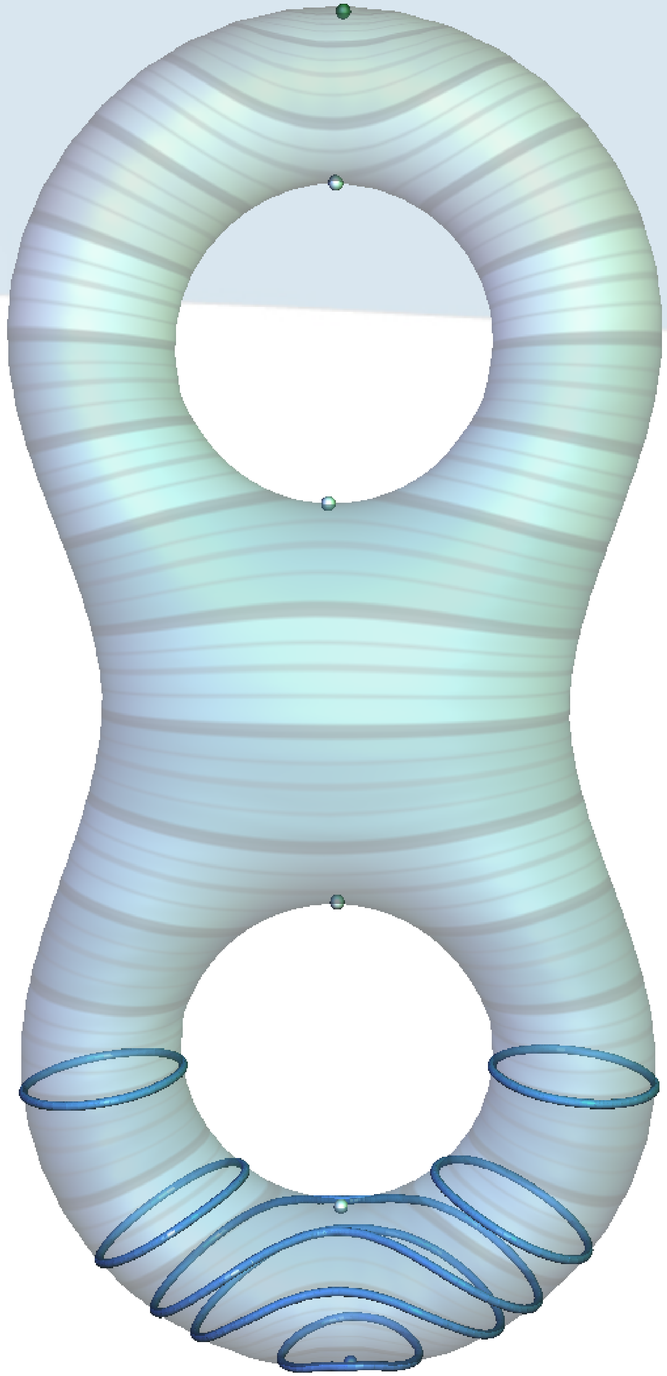
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \rightarrow \mathbb{R}$
  - *Morse* function



Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function



Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function

Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function

Indexing contours

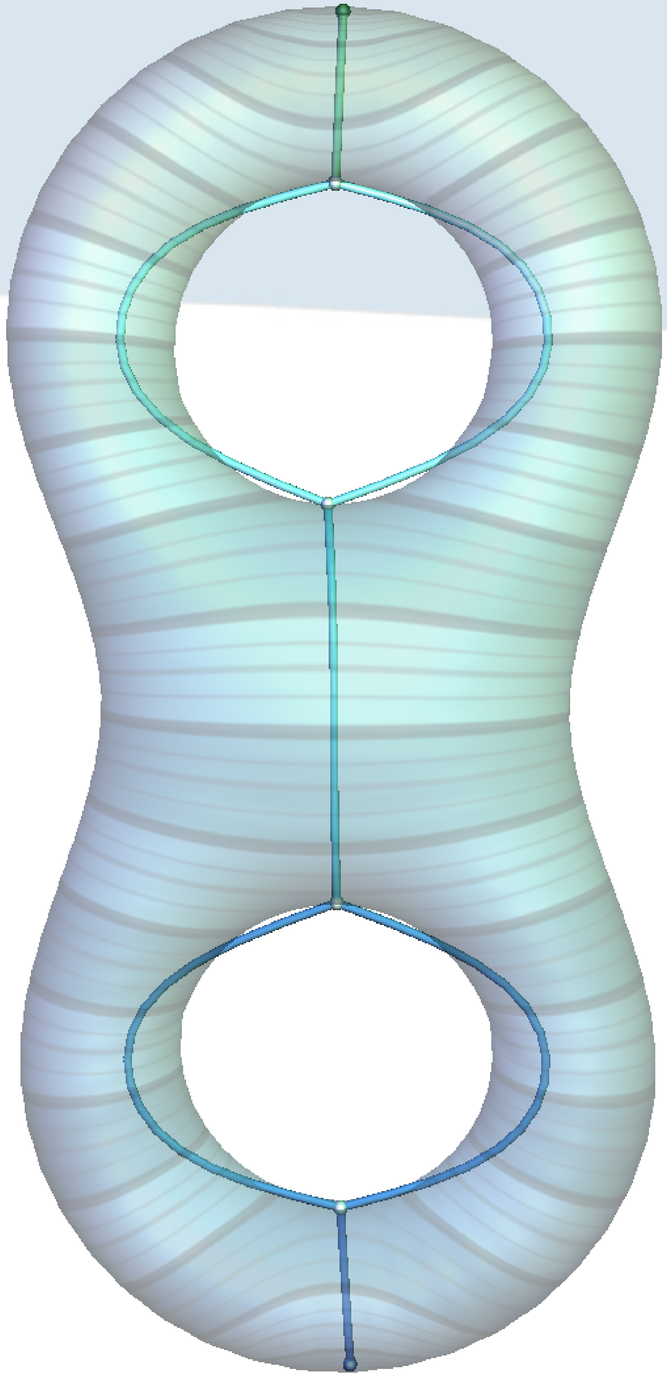
- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function

Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function

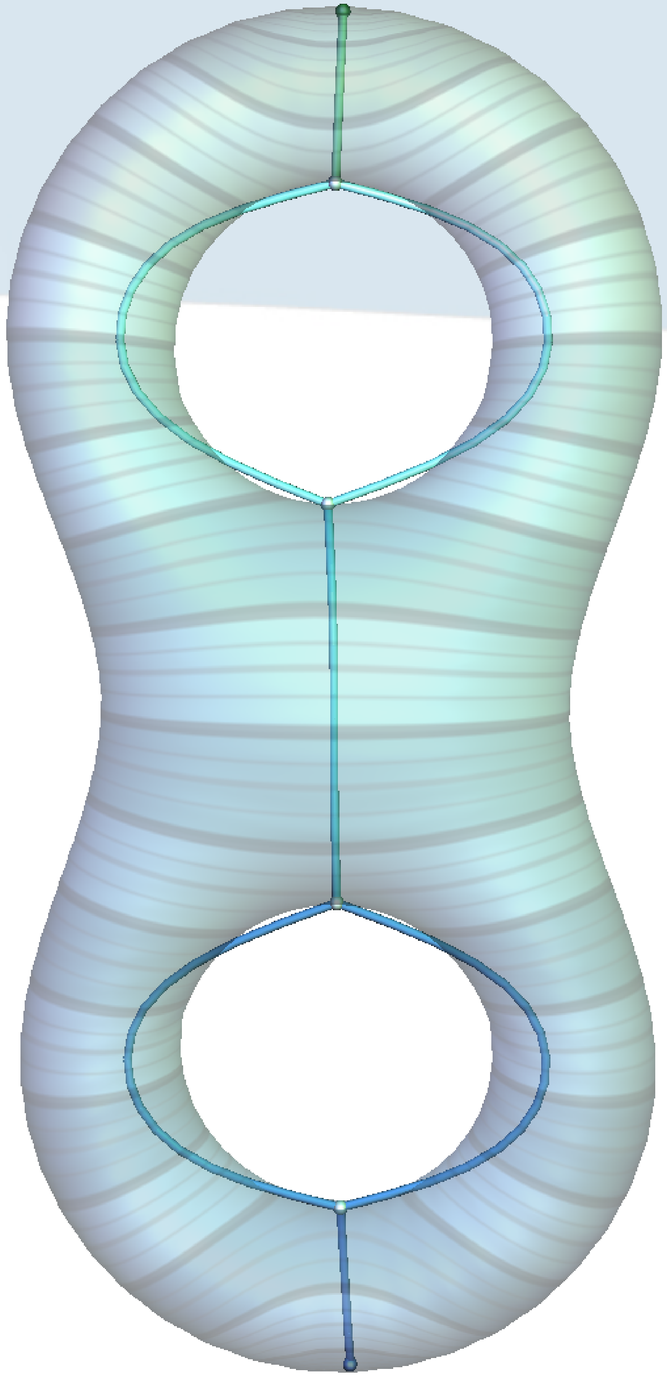
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function



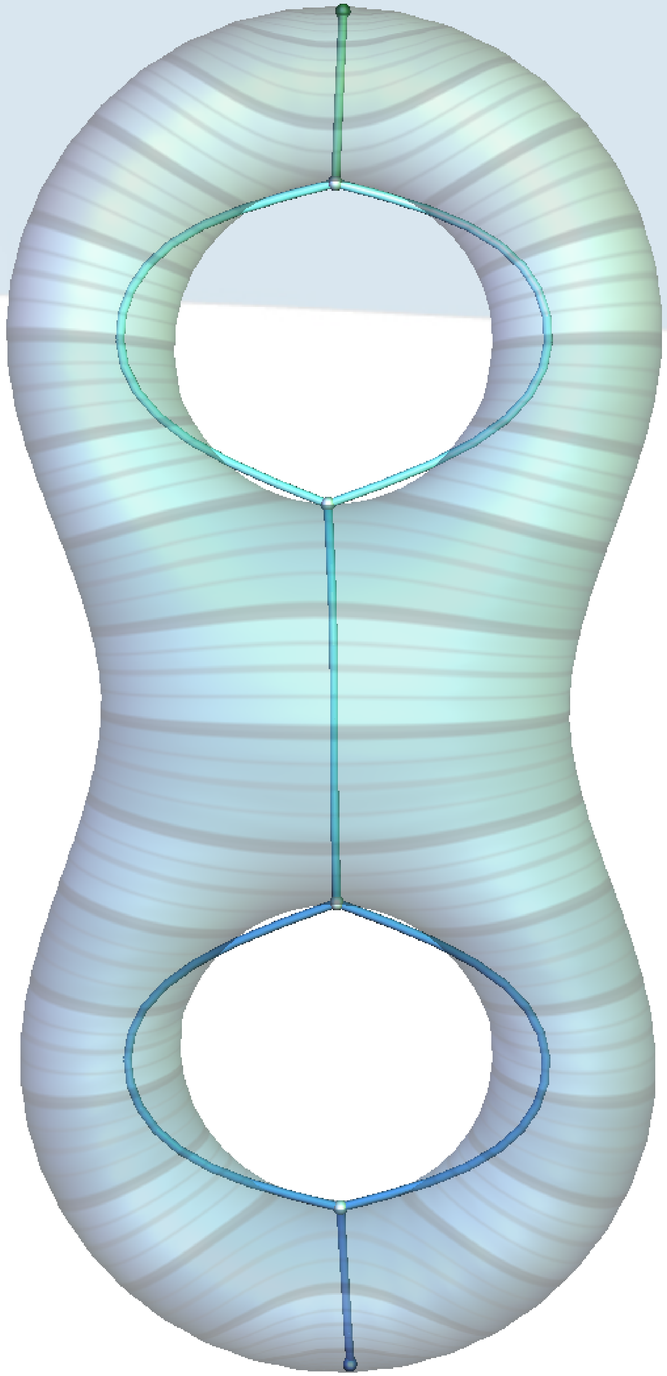
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function
- Contour retraction
  - Continuous map that retracts each contour to a single point



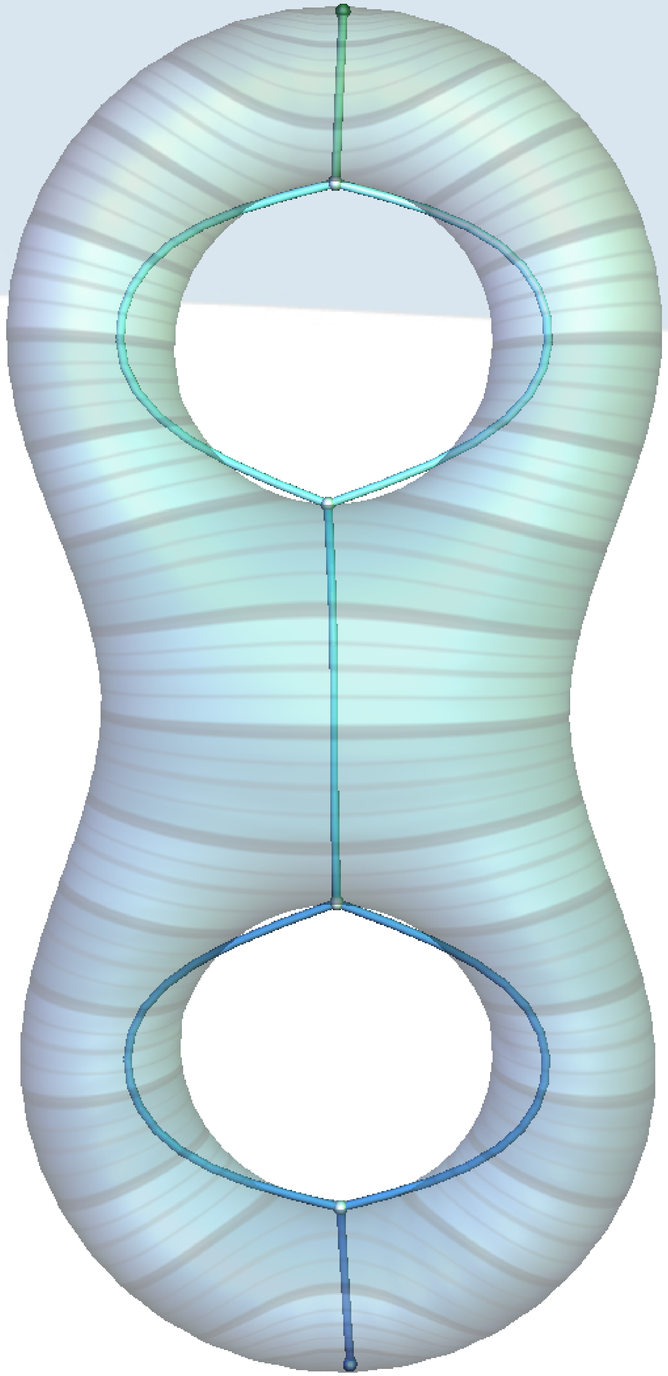
Indexing contours

- Notion of Reeb graph
  - $f : \mathcal{M} \to \mathbb{R}$
  - *Morse* function
- Contour retraction
  - Continuous map that retracts each contour to a single point
- Reeb graph $\mathcal{R}(f)$
  - Contour retraction of $\mathcal{M}$ under f



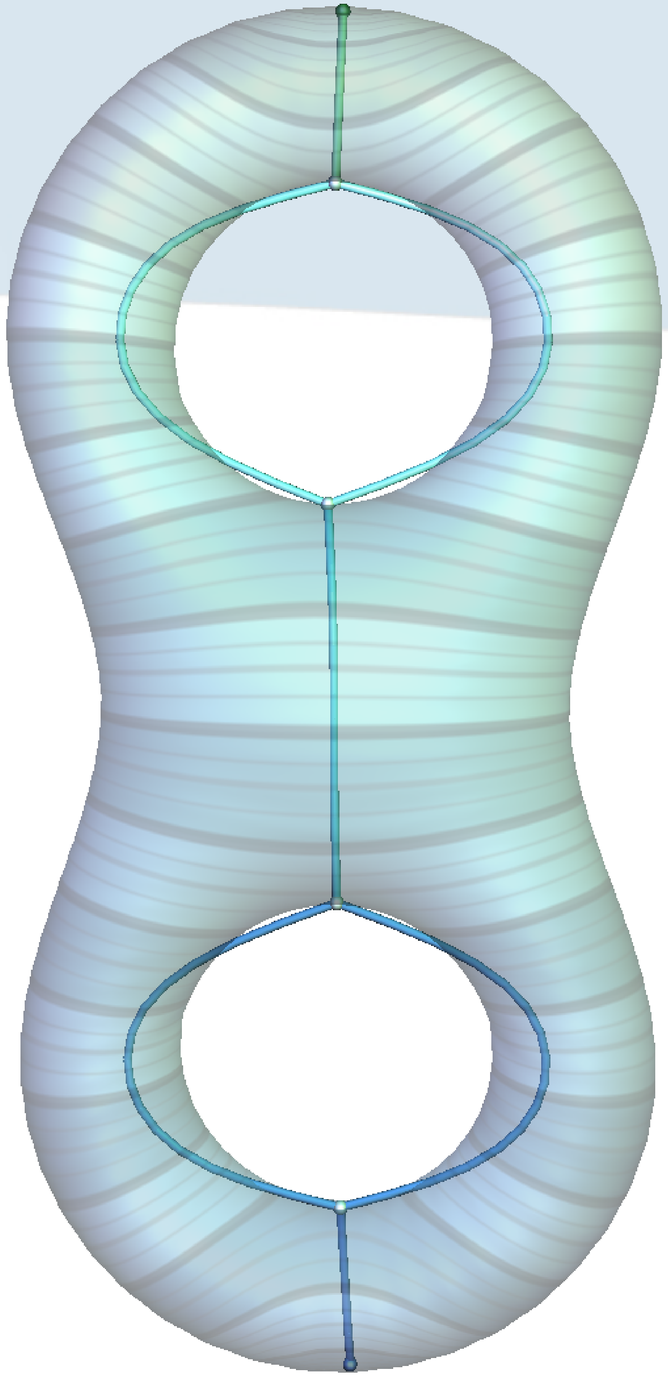
Indexing contours

- Reeb graph properties
  - Continuous 1-dimensional simplicial complex



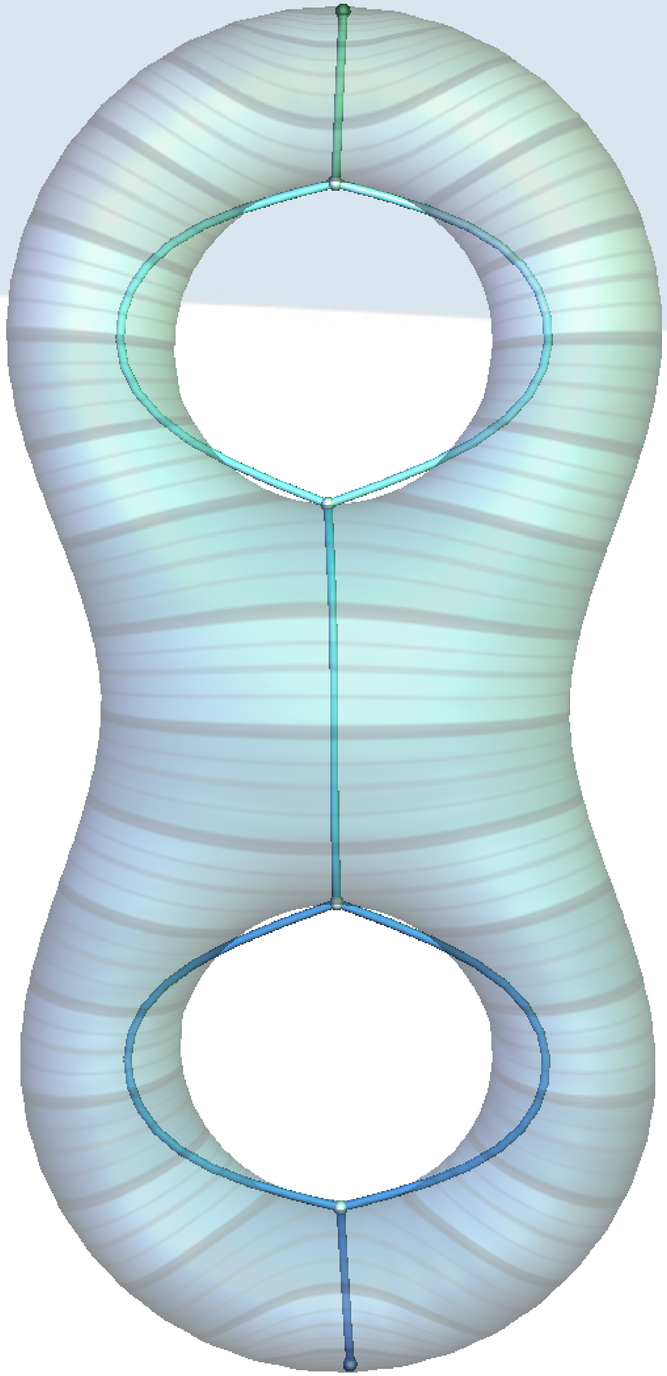
Indexing contours

- Reeb graph properties
  - Continuous 1-dimensional simplicial complex
  - Extrema: valence 1



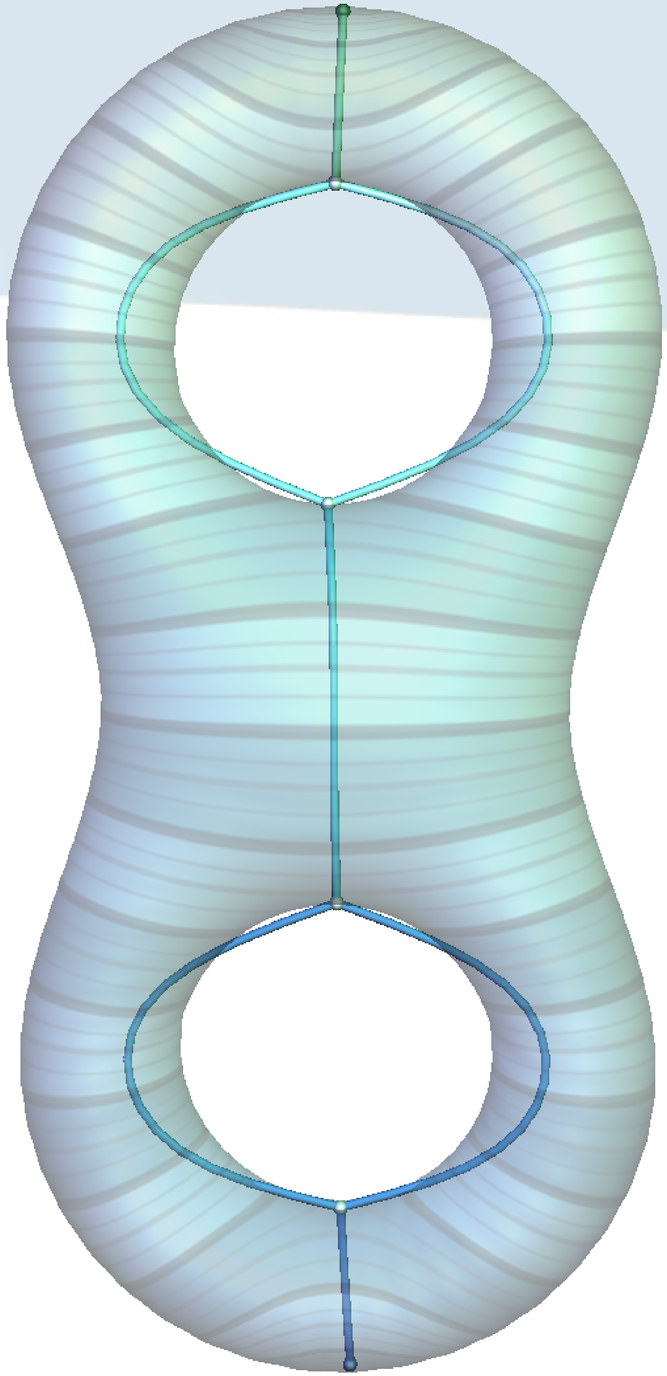
Indexing contours

- Reeb graph properties
  - Continuous 1-dimensional simplicial complex
  - Extrema: valence 1
  - Saddles in 2D: valence 3 or 4 (boundary)
  - Saddles in nD: valence 2 or 3



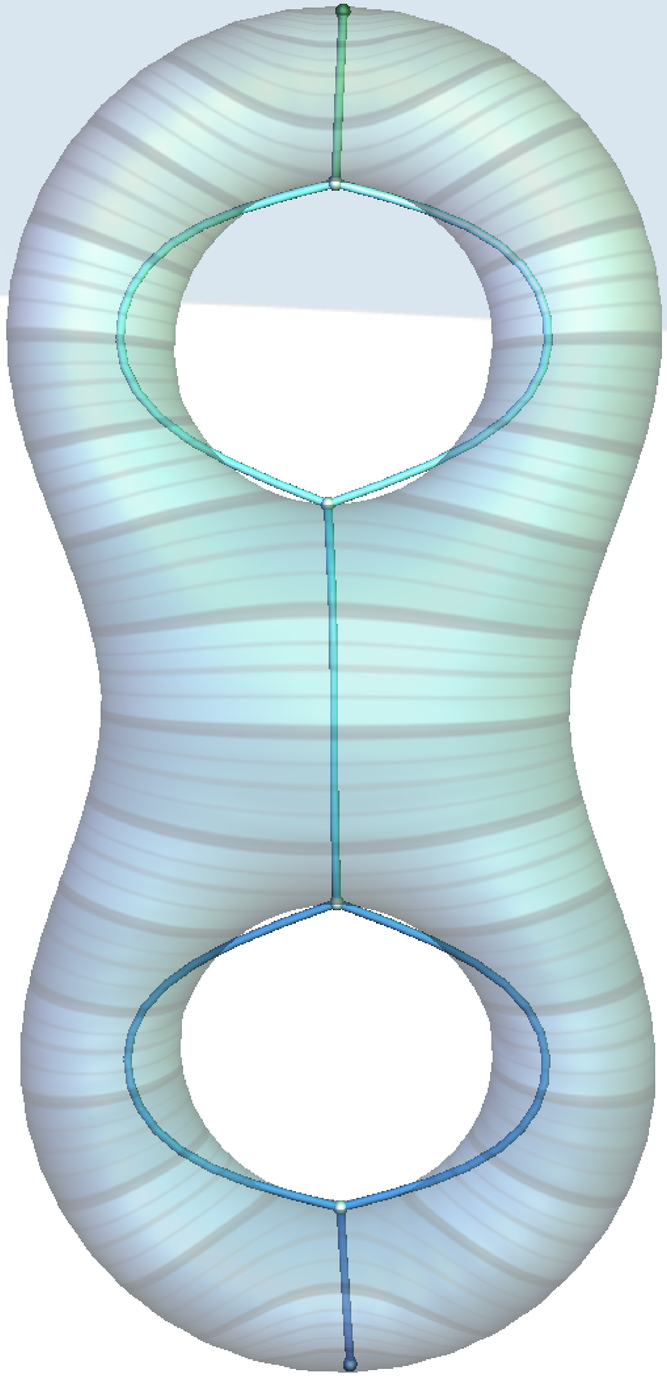
Indexing contours

- Reeb graph properties
  - Continuous 1-dimensional simplicial complex
  - Extrema: valence 1
  - Saddles in 2D: valence 3 or 4 (boundary)
  - Saddles in nD: valence 2 or 3
- In practice:
  - Arcs: collection of vertices



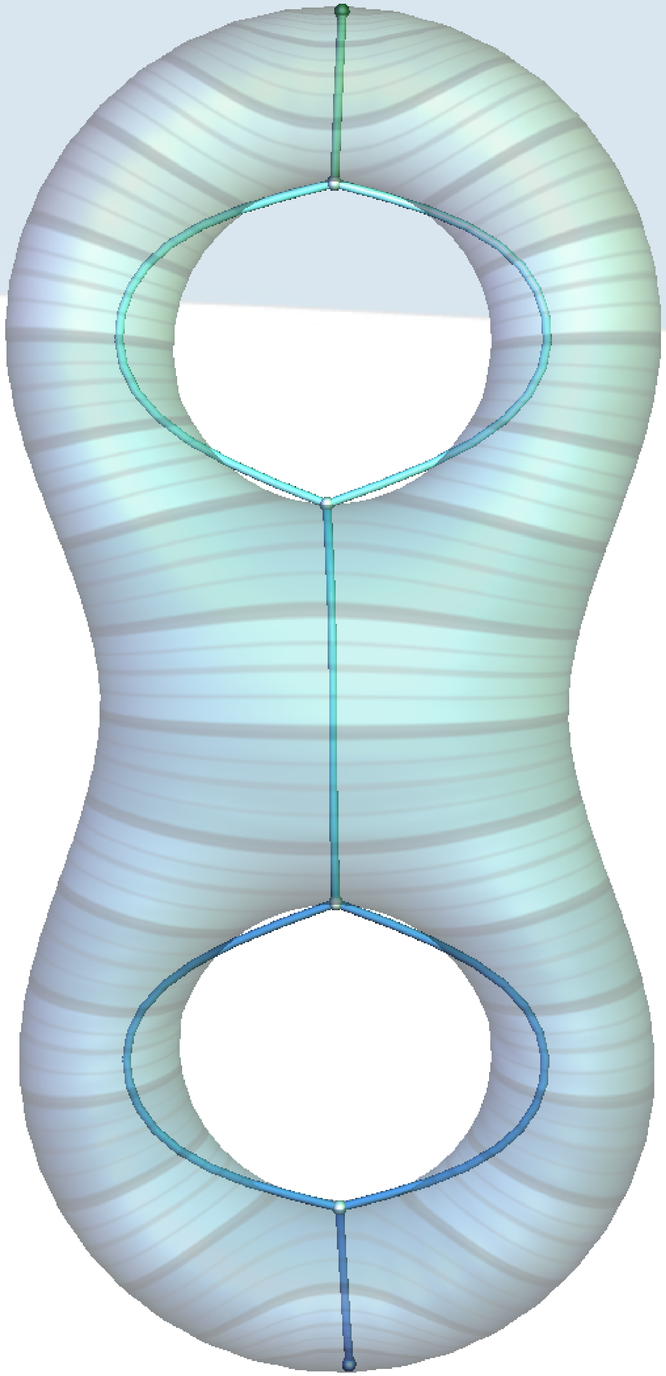
Indexing contours

- Reeb graph properties
  - Continuous 1-dimensional simplicial complex
  - Extrema: valence 1
  - Saddles in 2D: valence 3 or 4 (boundary)
  - Saddles in nD: valence 2 or 3
- In practice:
  - Arcs: collection of vertices
  - Simply connected domains: Carr00



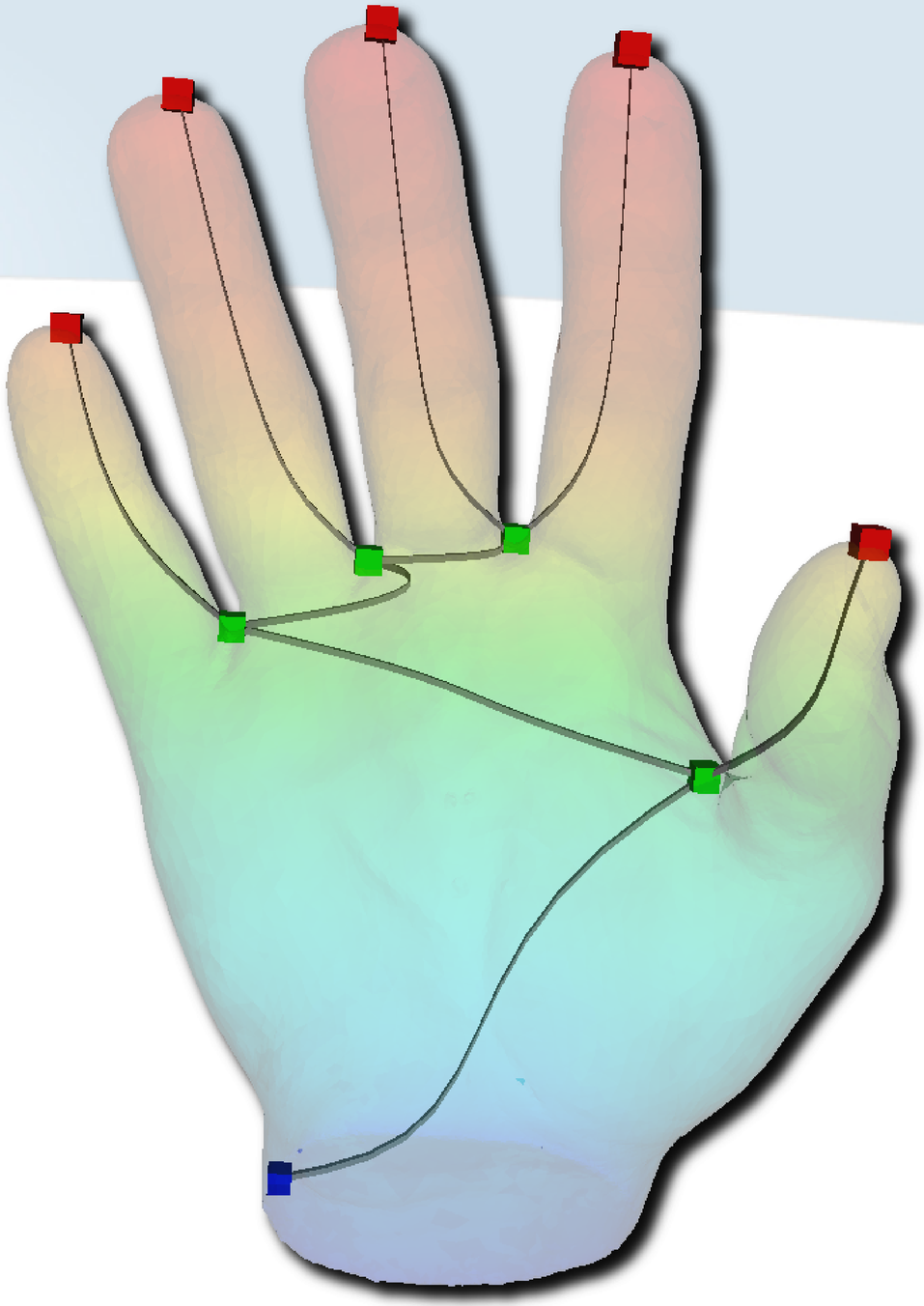
Indexing contours

- Reeb graph properties
  - Continuous 1-dimensional simplicial complex
  - Extrema: valence 1
  - Saddles in 2D: valence 3 or 4 (boundary)
  - Saddles in nD: valence 2 or 3
- In practice:
  - Arcs: collection of vertices
  - Simply connected domains: Carr00
  - Otherwise: Pascucci07, Tierny09, Parsa12



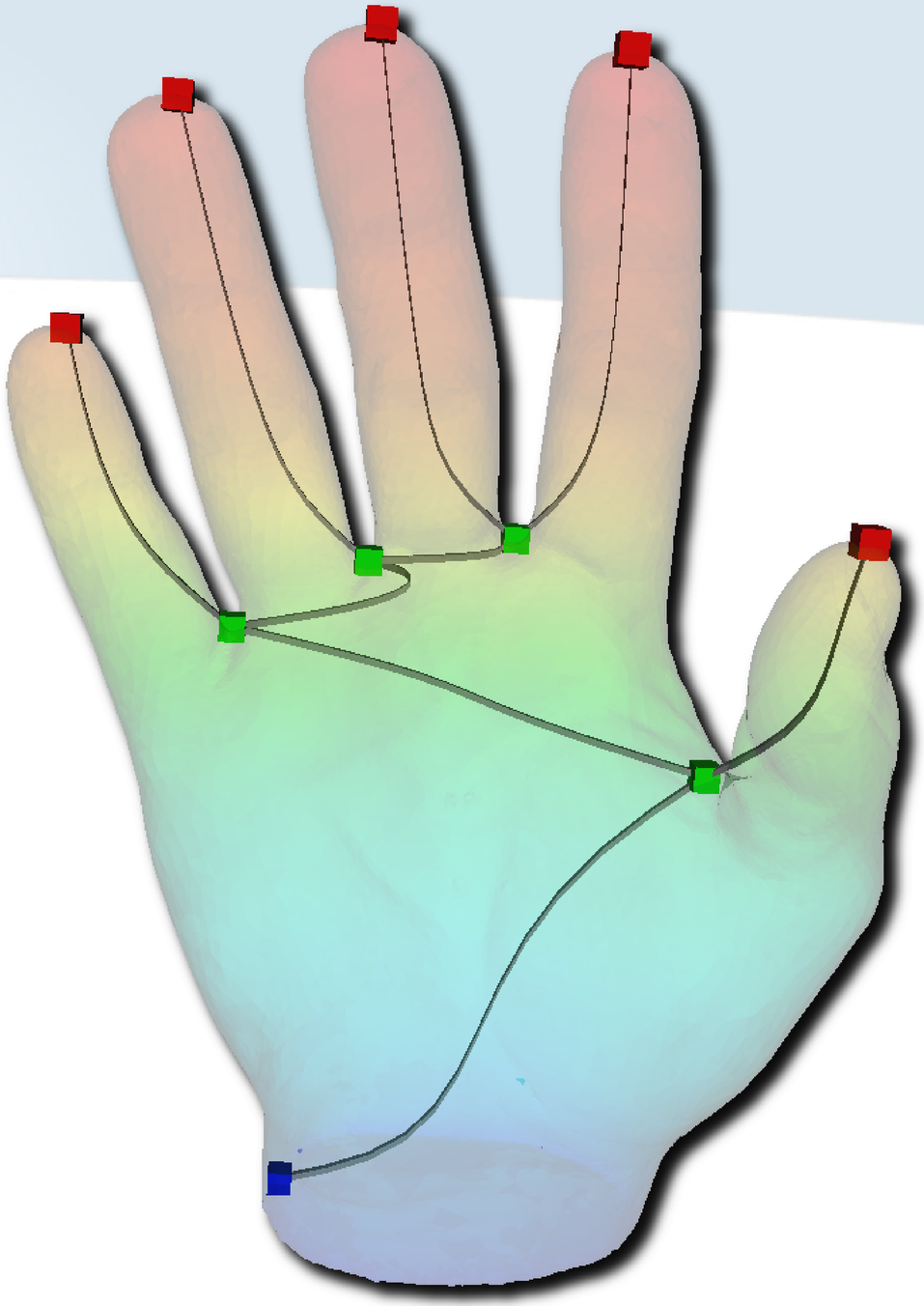
Online contour query

- Reeb graph computation
  - Once for all, matters of seconds



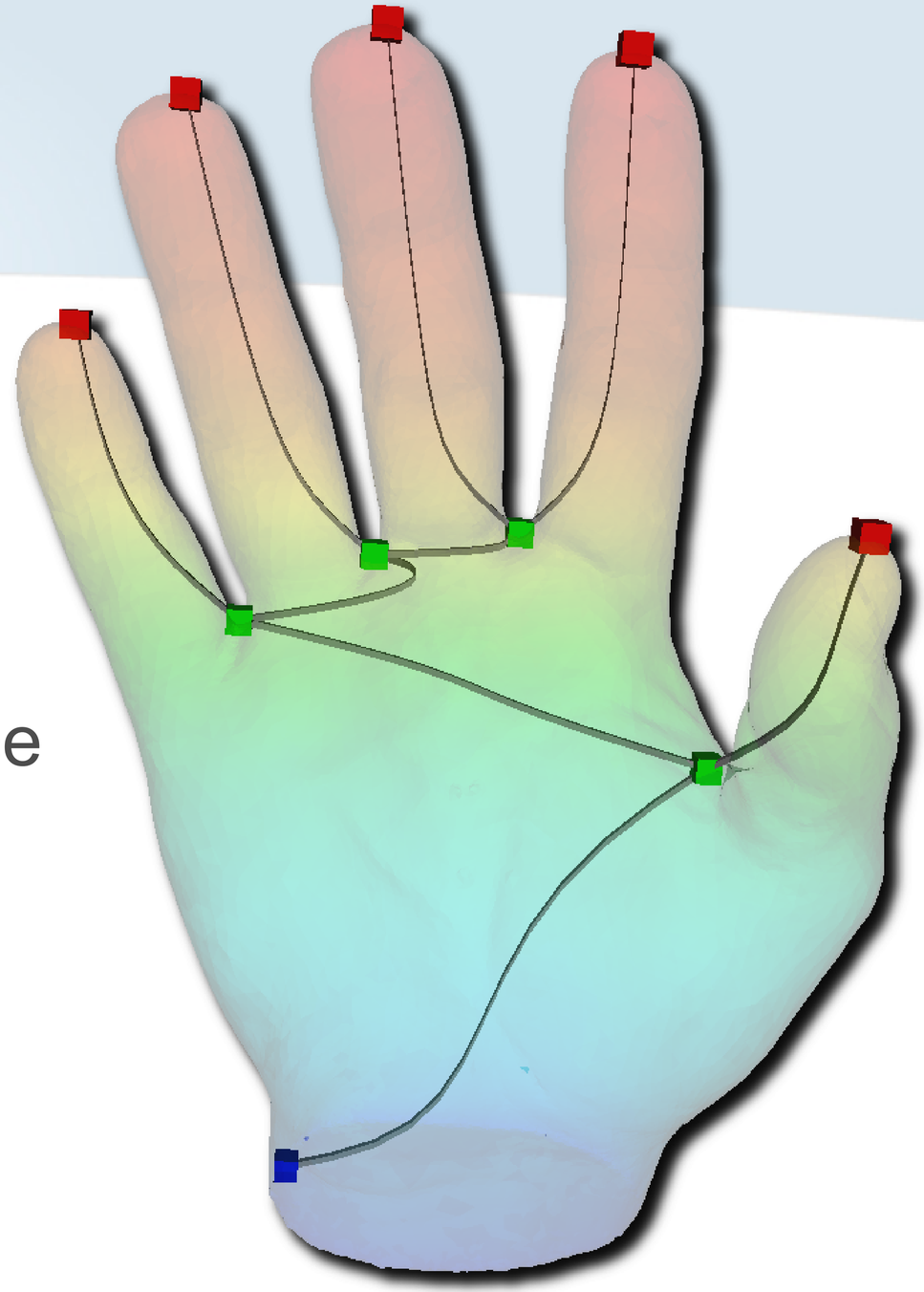
Online contour query

- Reeb graph computation
  - Once for all, matters of seconds
  - Store the arcs in balanced interval tree



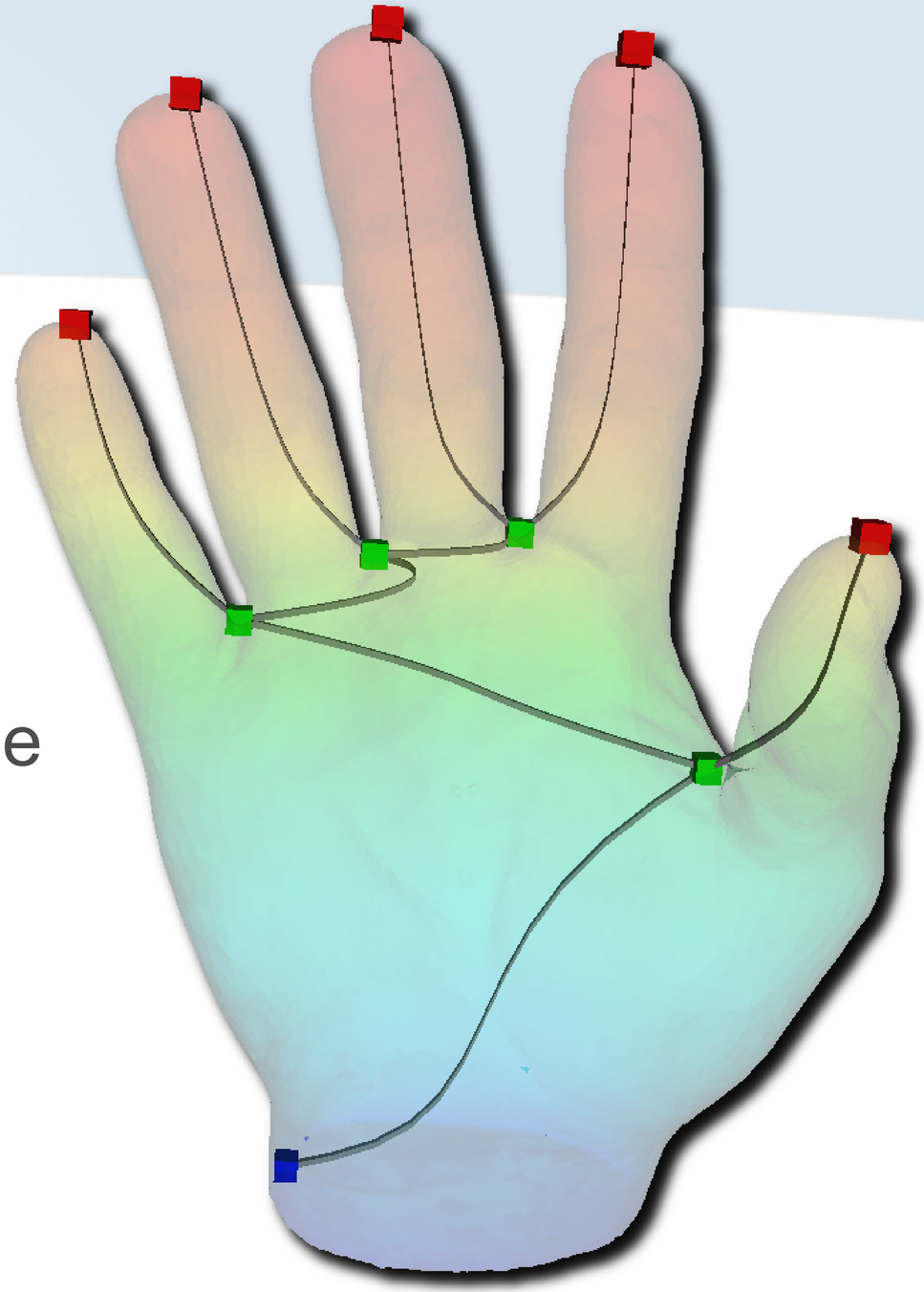
Online contour query

- Reeb graph computation
  - Once for all, matters of seconds
  - Store the arcs in balanced interval tree
  - For each arc, balanced binary search tree



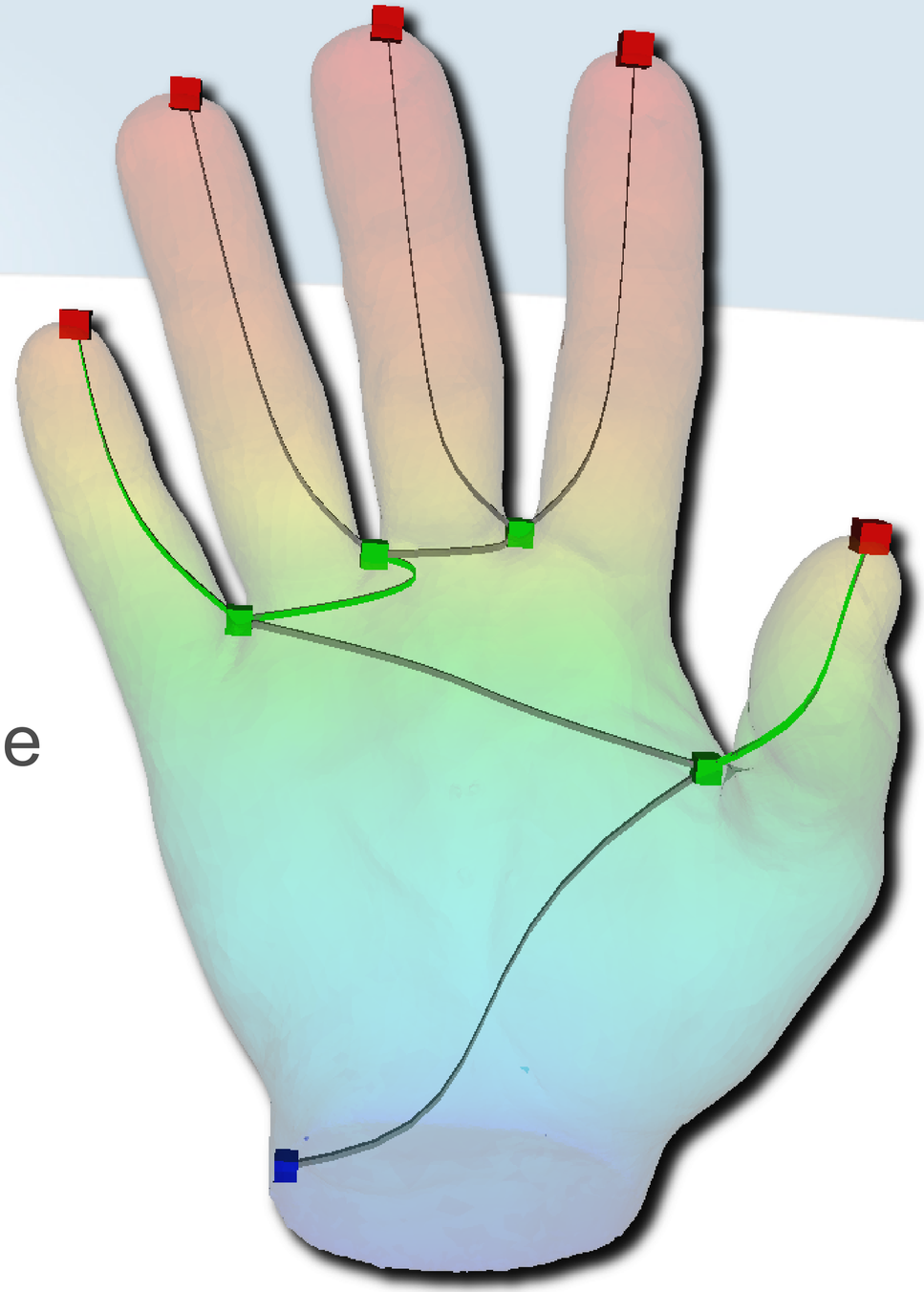
Online contour query

- Reeb graph computation
  - Once for all, matters of seconds
  - Store the arcs in balanced interval tree
  - For each arc, balanced binary search tree
- Isovalue query:



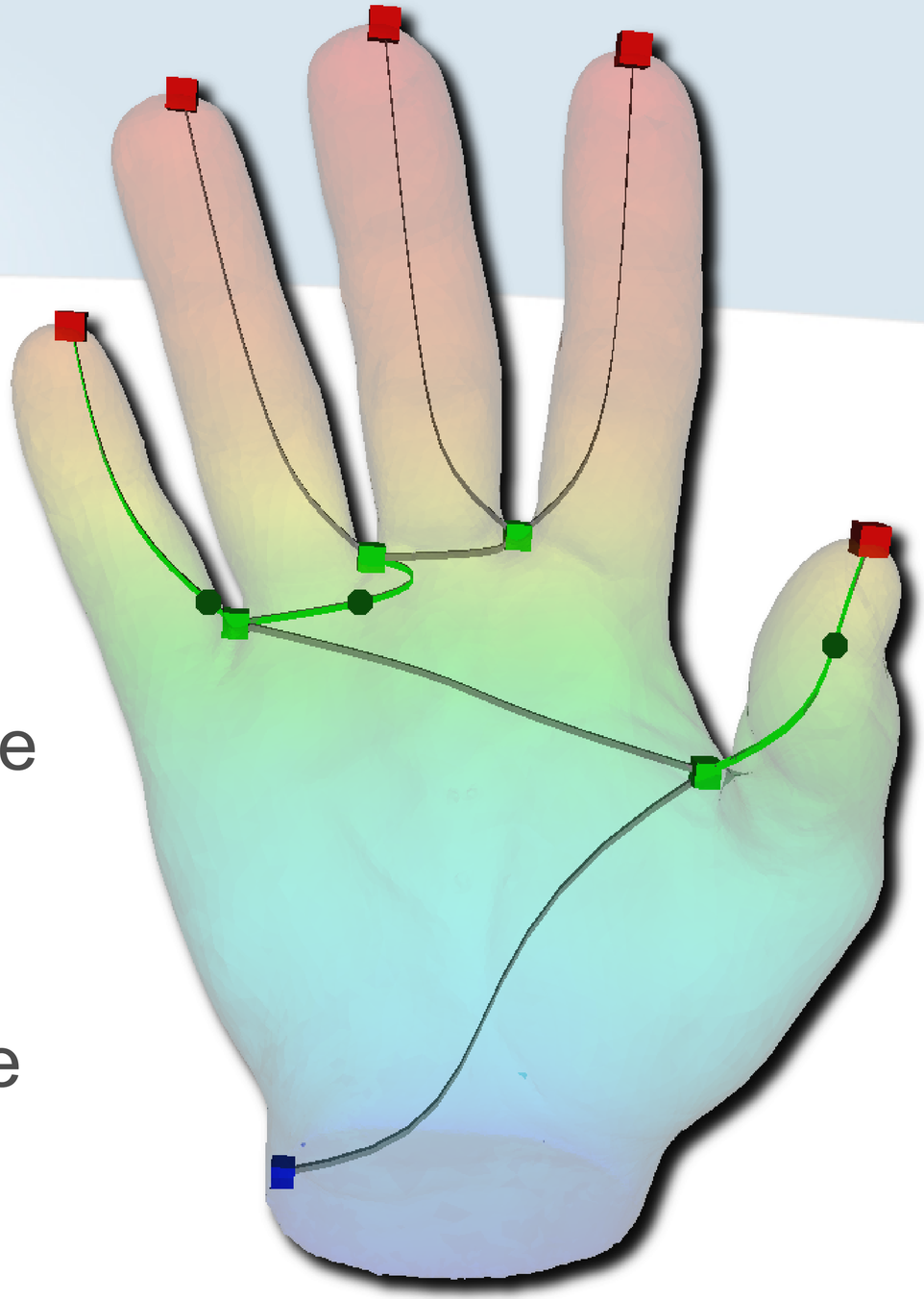
Online contour query

- Reeb graph computation
  - Once for all, matters of seconds
  - Store the arcs in balanced interval tree
  - For each arc, balanced binary search tree
- Isovalue query:
  - Find the arcs crossing the isovalue



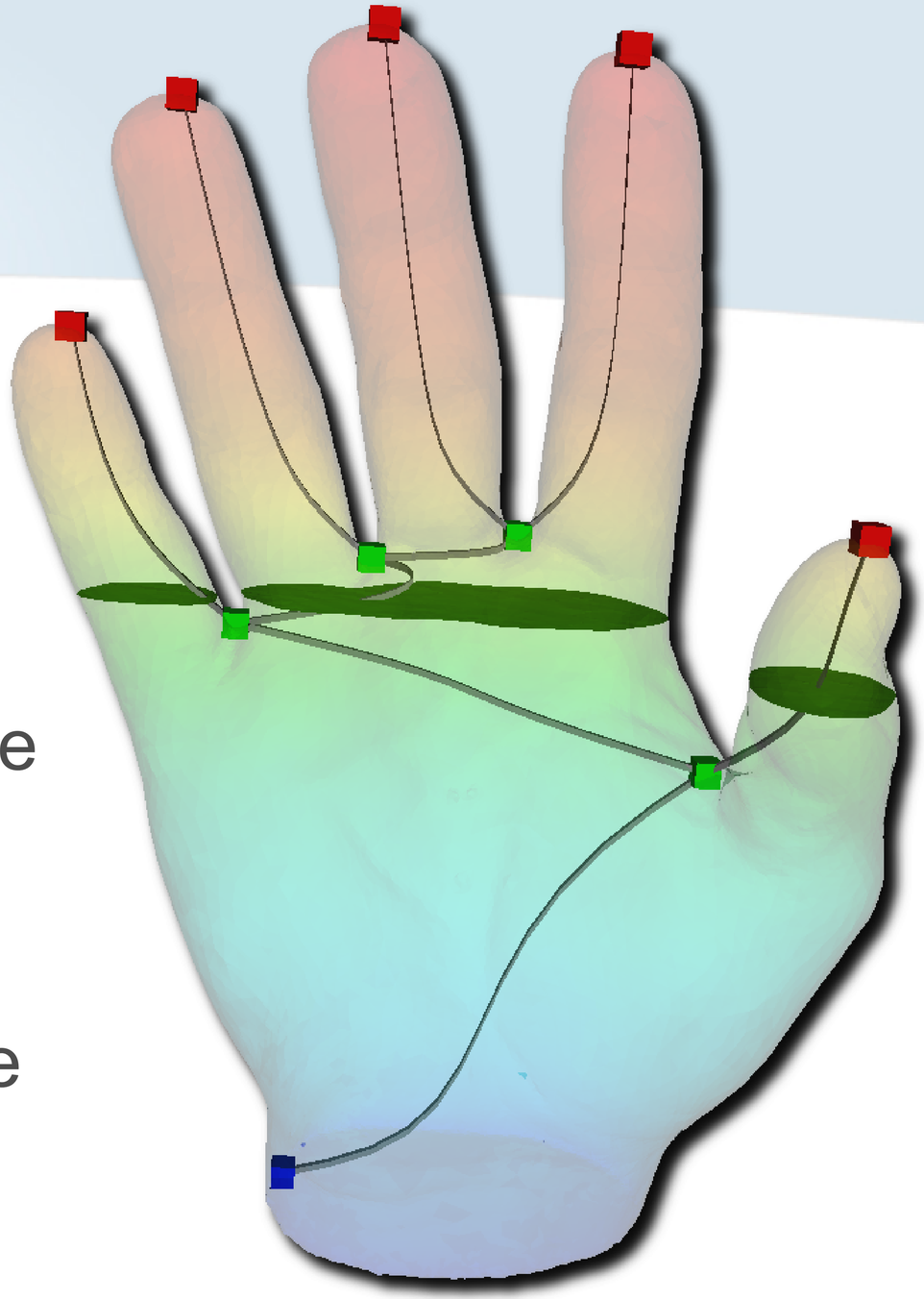
Online contour query

- Reeb graph computation
  - Once for all, matters of seconds
  - Store the arcs in balanced interval tree
  - For each arc, balanced binary search tree
- Isovalue query:
  - Find the arcs crossing the isovalue
    - Find the vertex right below the isovalue



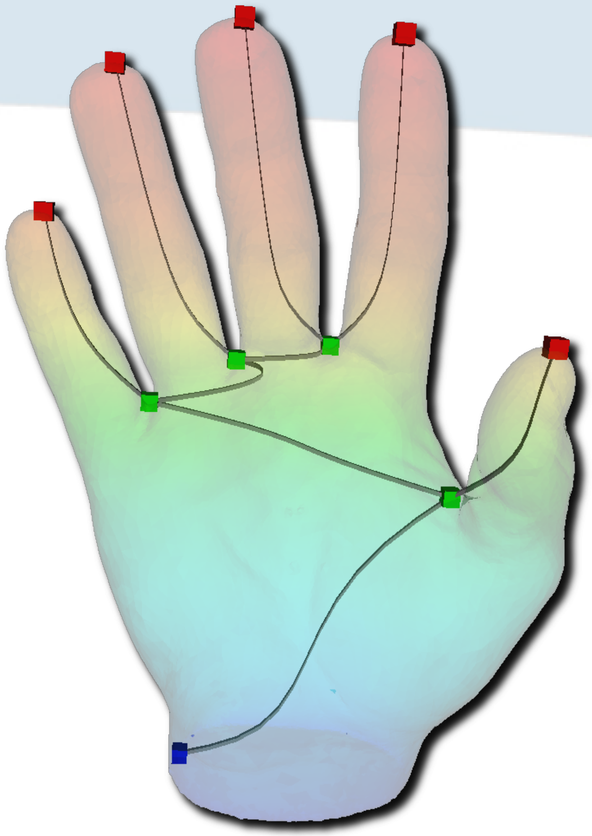
Online contour query

- Reeb graph computation
  - Once for all, matters of seconds
  - Store the arcs in balanced interval tree
  - For each arc, balanced binary search tree
- Isovalue query:
  - Find the arcs crossing the isovalue
    - Find the vertex right below the isovalue
    - Return its d-simplices



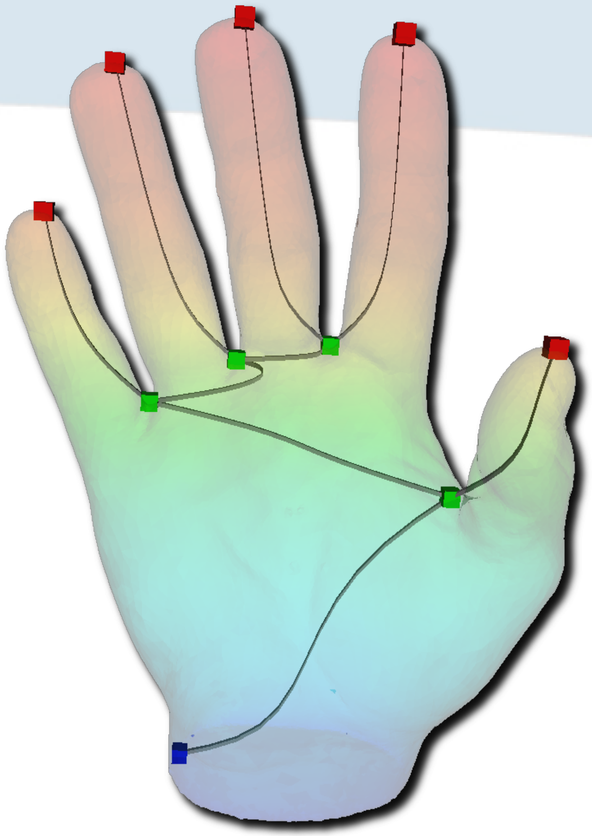
Query with topological simplification

- Reeb graph simplification



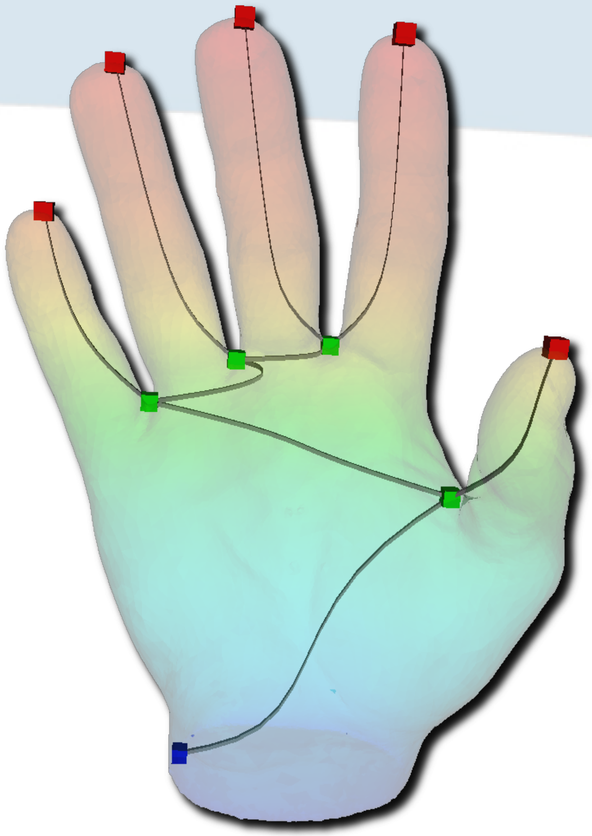
Query with topological simplification

- Reeb graph simplification
  - Persistent homology



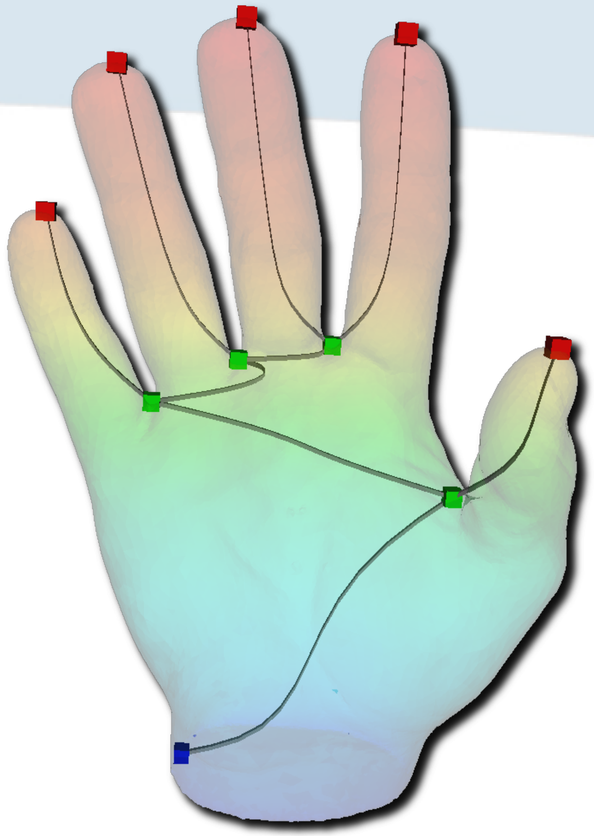
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold



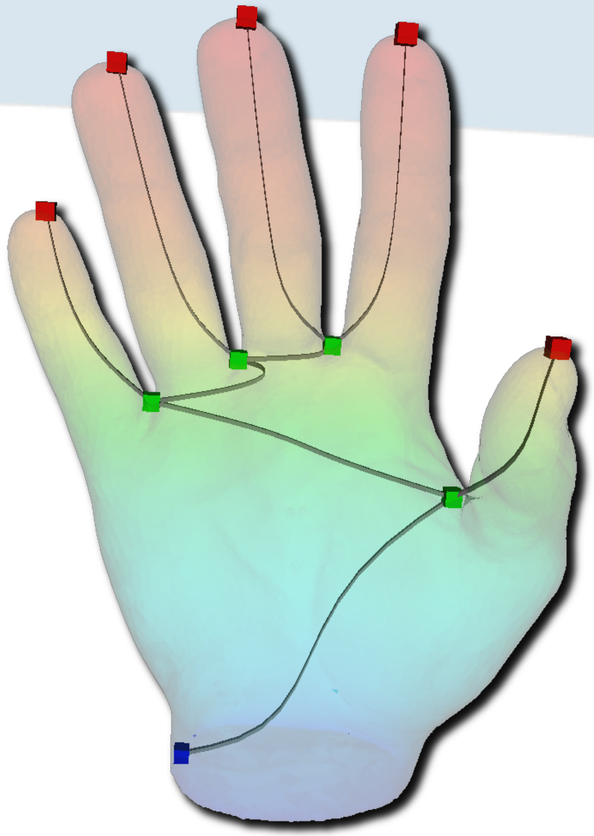
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



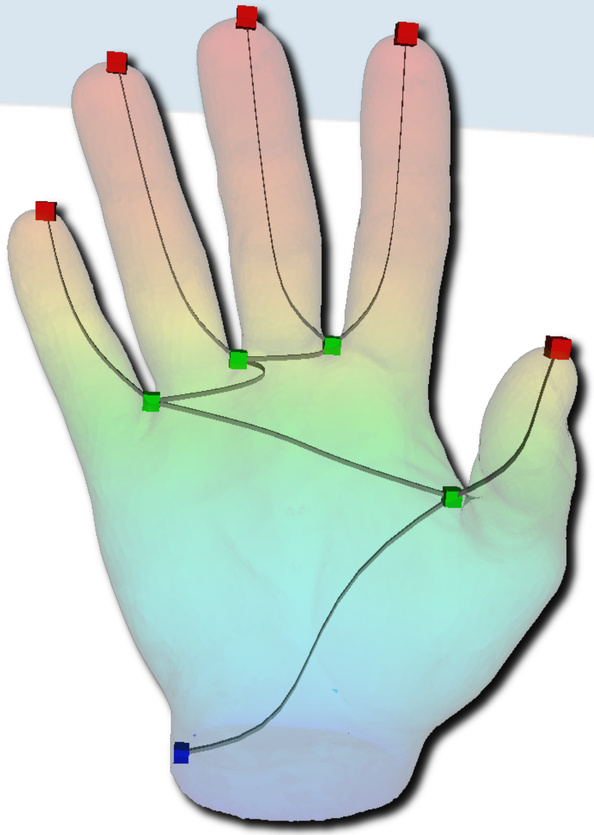
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



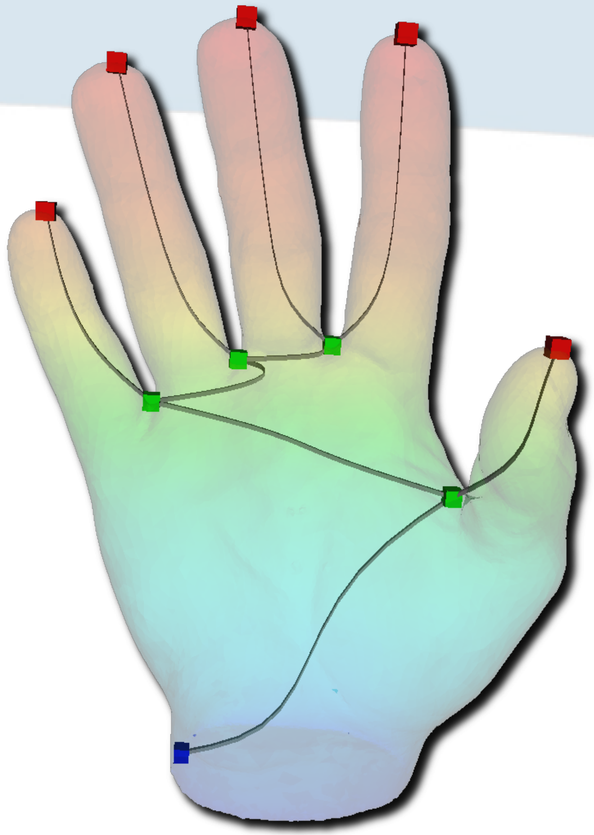
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



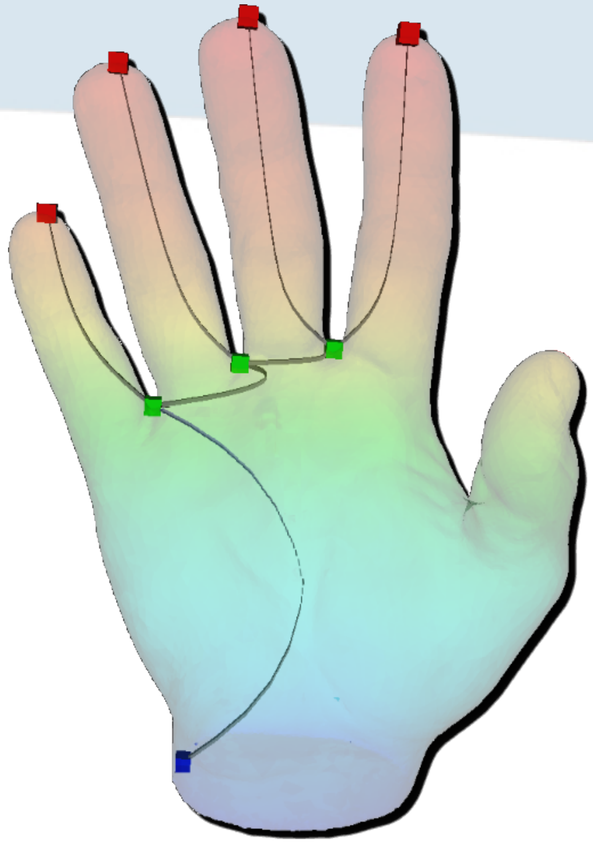
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



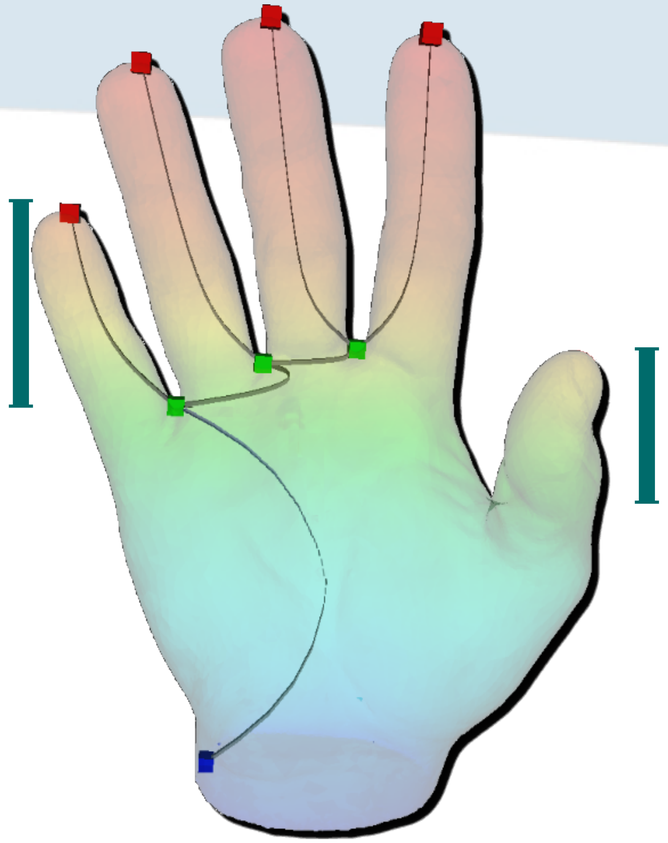
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



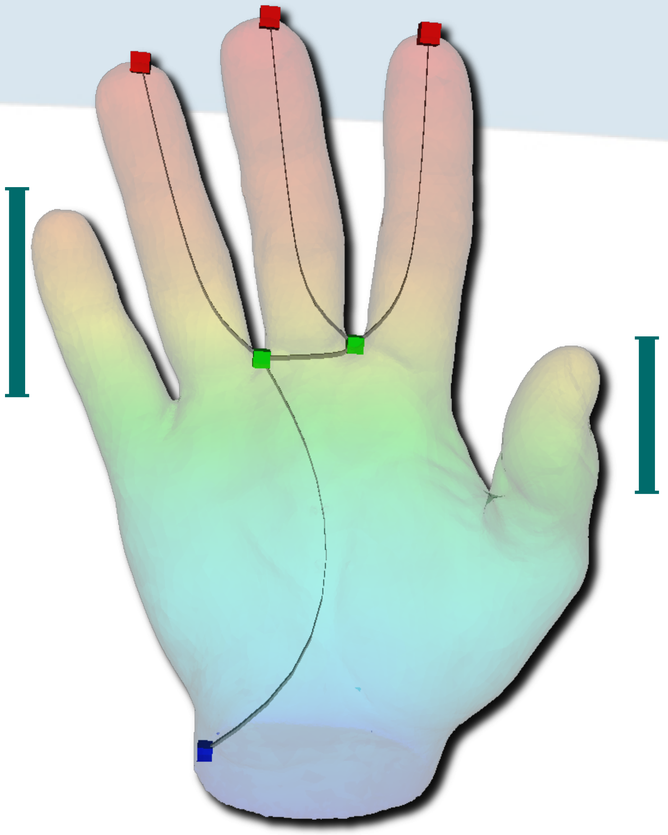
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



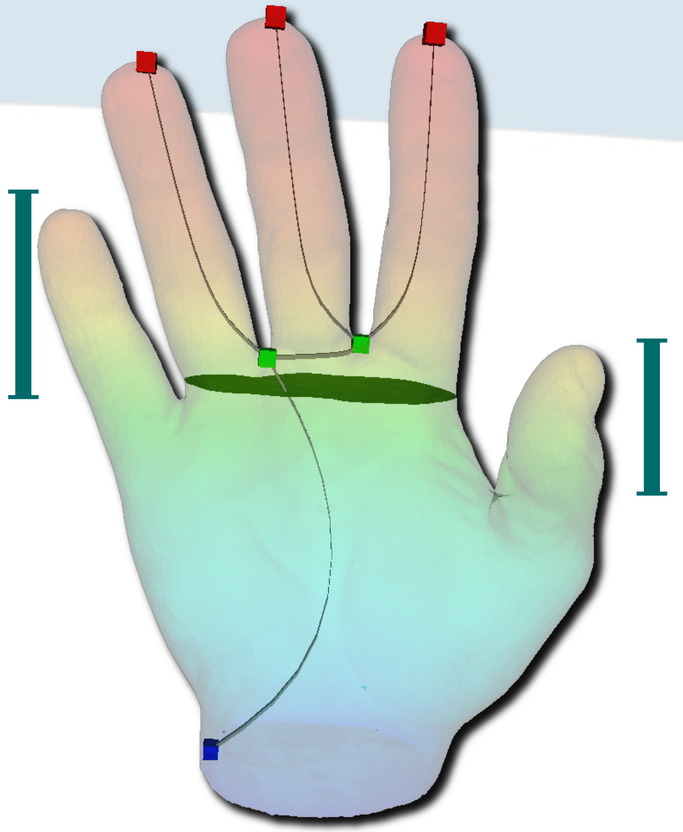
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



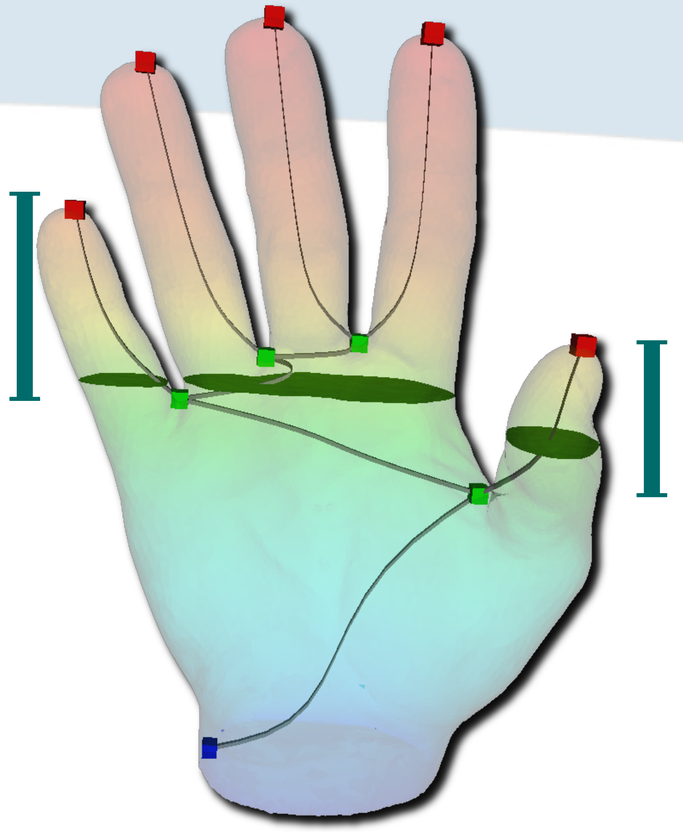
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



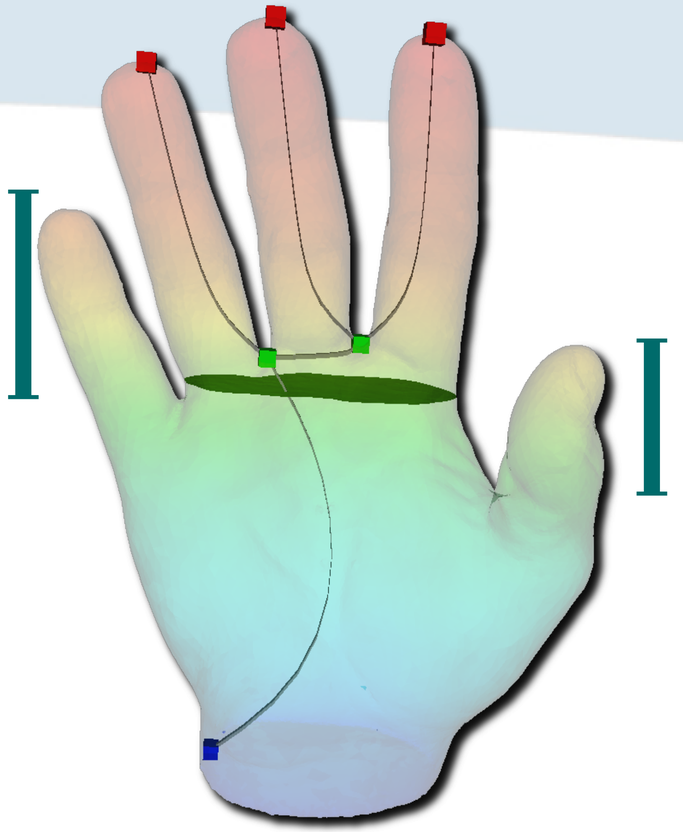
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only



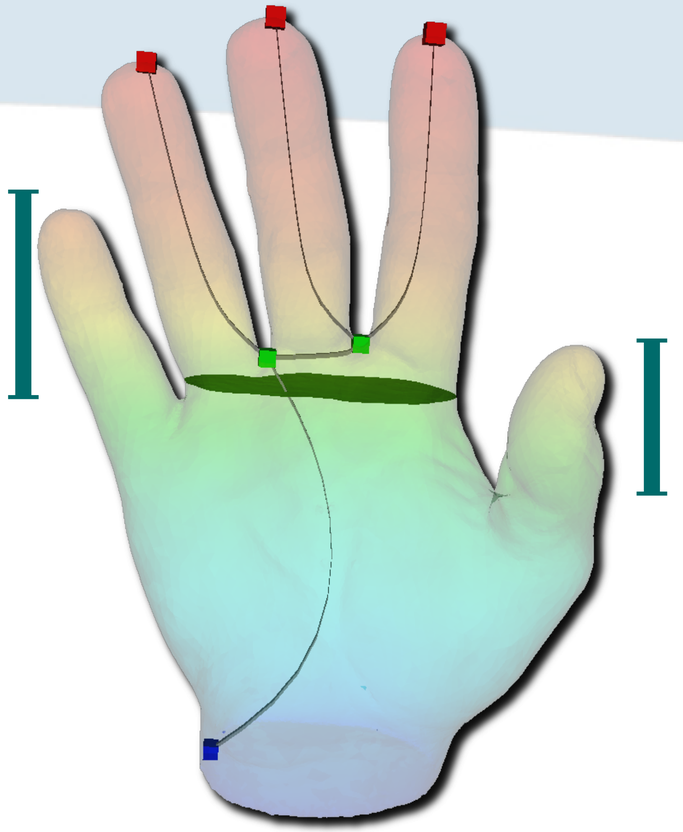
Query with topological simplification

- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only

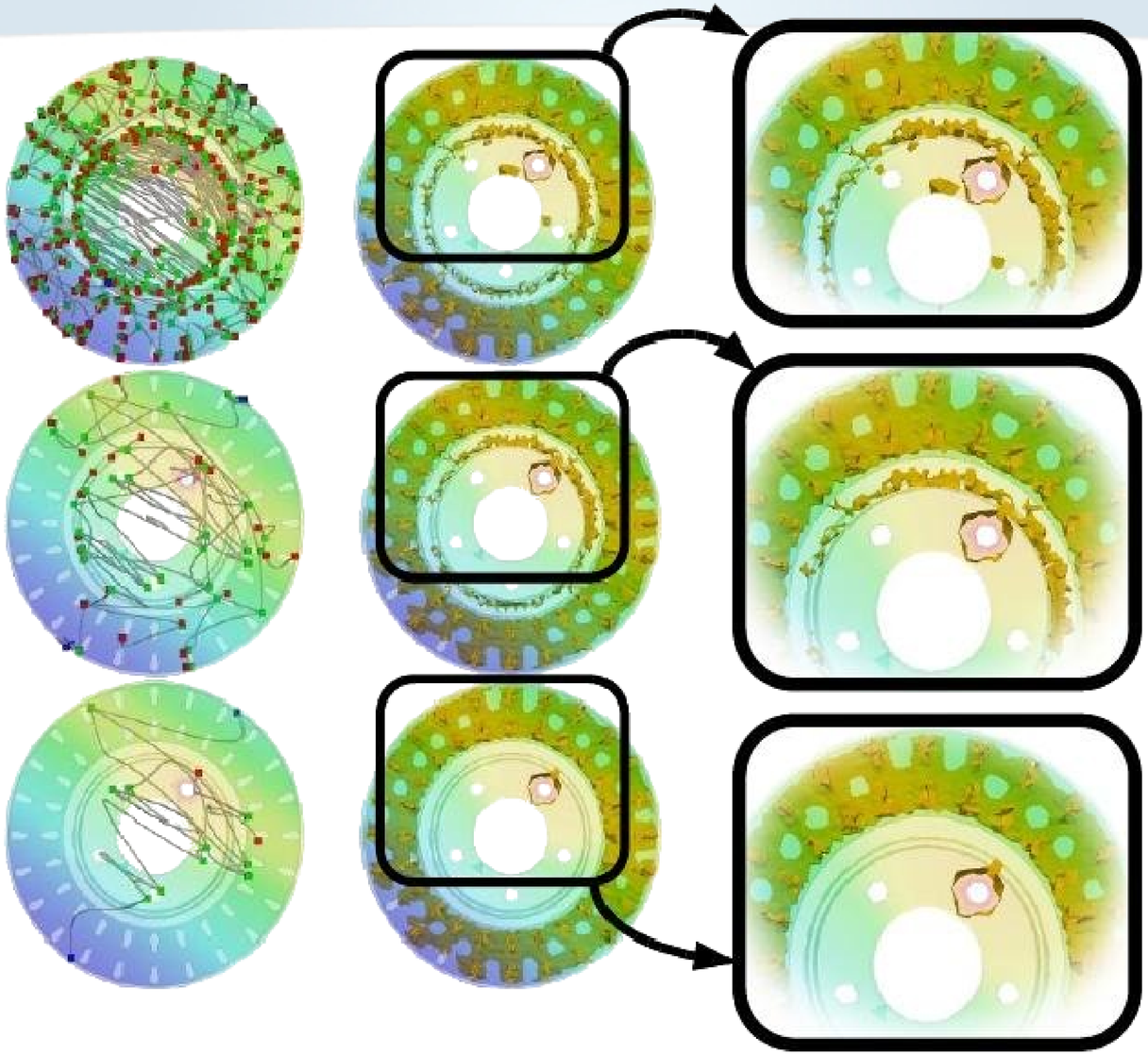


Query with topological simplification

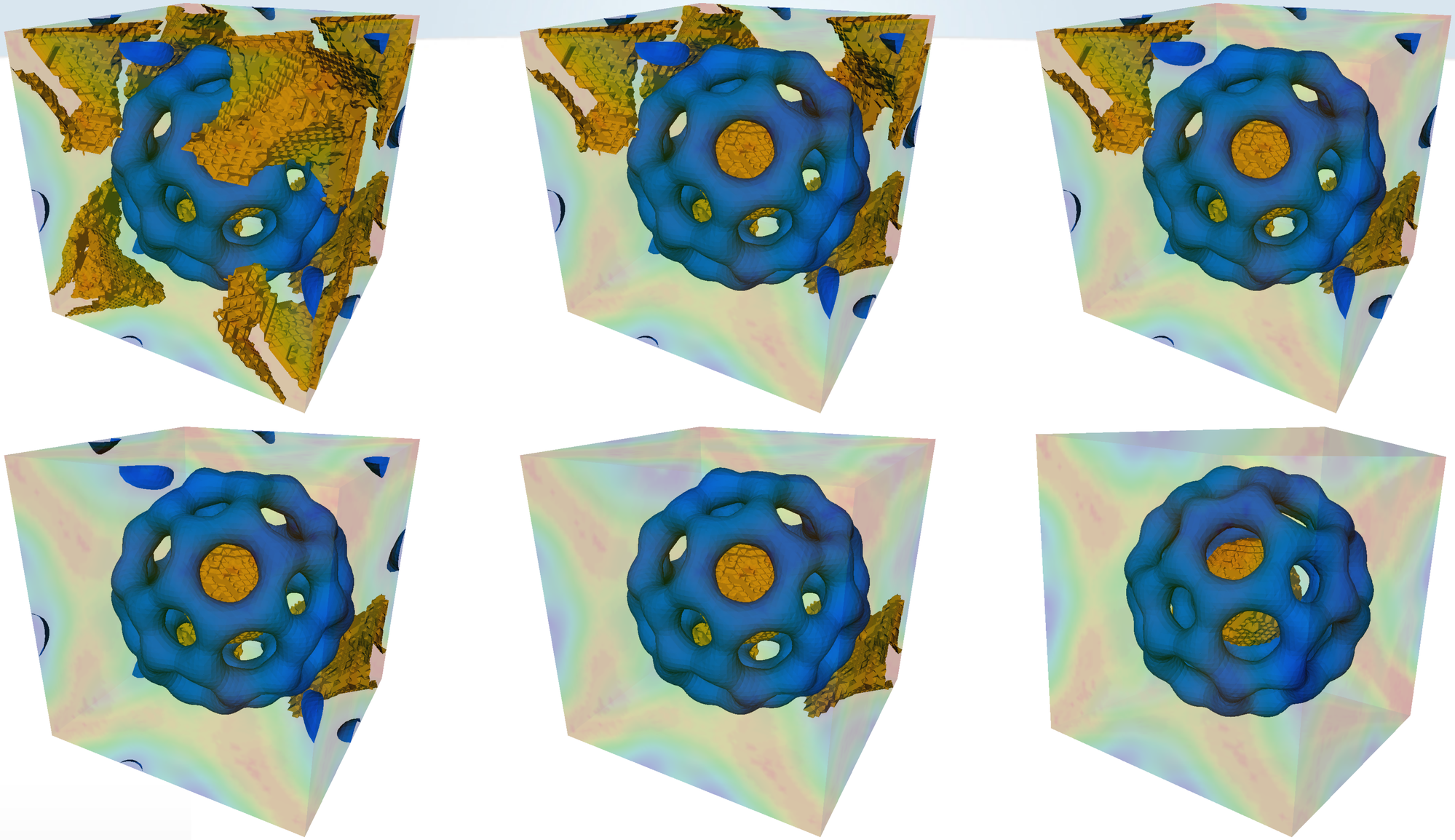
- Reeb graph simplification
  - Persistent homology
  - Given a criterion and a threshold
    - Remove the *smallest* arcs one by one
    - Arcs under the threshold only
- Criterion examples:
  - Function value difference
  - Volume/Area of the arc
  - ...



Query with topological simplification



Query with topological simplification



In conclusion

- Now you know

In conclusion

- Now you know
  - How to visualize scalar fields

In conclusion

- Now you know
  - How to visualize scalar fields
    - Higher dimensional embeddings

In conclusion

- Now you know
  - How to visualize scalar fields
    - Higher dimensional embeddings
    - Use of colors (maps, volume rendering)

In conclusion

- Now you know
  - How to visualize scalar fields
    - Higher dimensional embeddings
    - Use of colors (maps, volume rendering)
    - Level extraction
      - PL manifold domains
      - Regular grids

In conclusion

- Now you know
  - How to visualize scalar fields
    - Higher dimensional embeddings
    - Use of colors (maps, volume rendering)
    - Level extraction
      - PL manifold domains
      - Regular grids
    - Fast level set computation

In conclusion

- Now you know
  - How to visualize scalar fields
    - Higher dimensional embeddings
    - Use of colors (maps, volume rendering)
    - Level extraction
      - PL manifold domains
      - Regular grids
    - Fast level set computation
    - Topological simplification of level sets