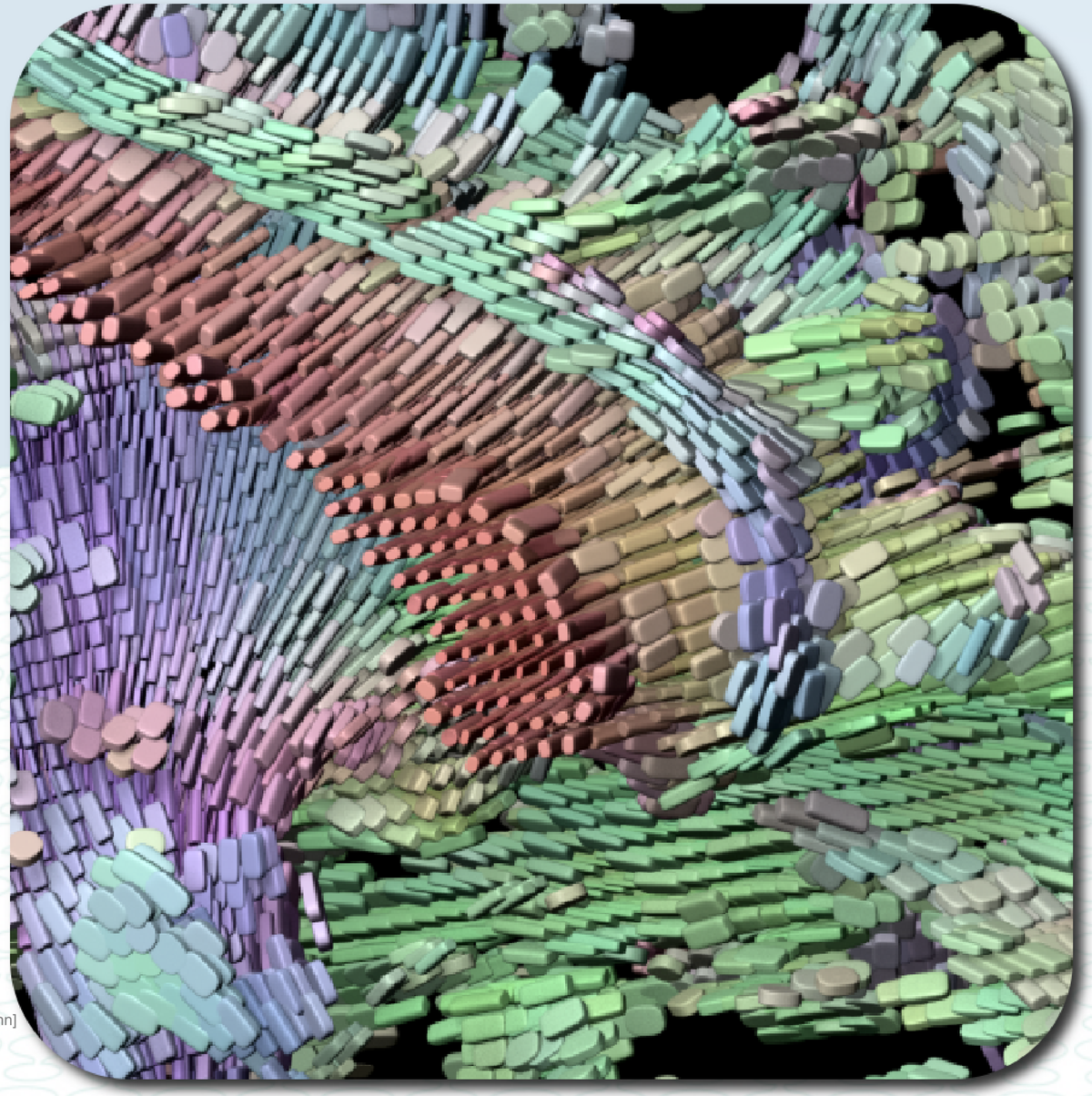


[Kindlmann]

Introduction to Scientific Visualization

Télécom ParisTech

tierny@telecom-paristech.fr

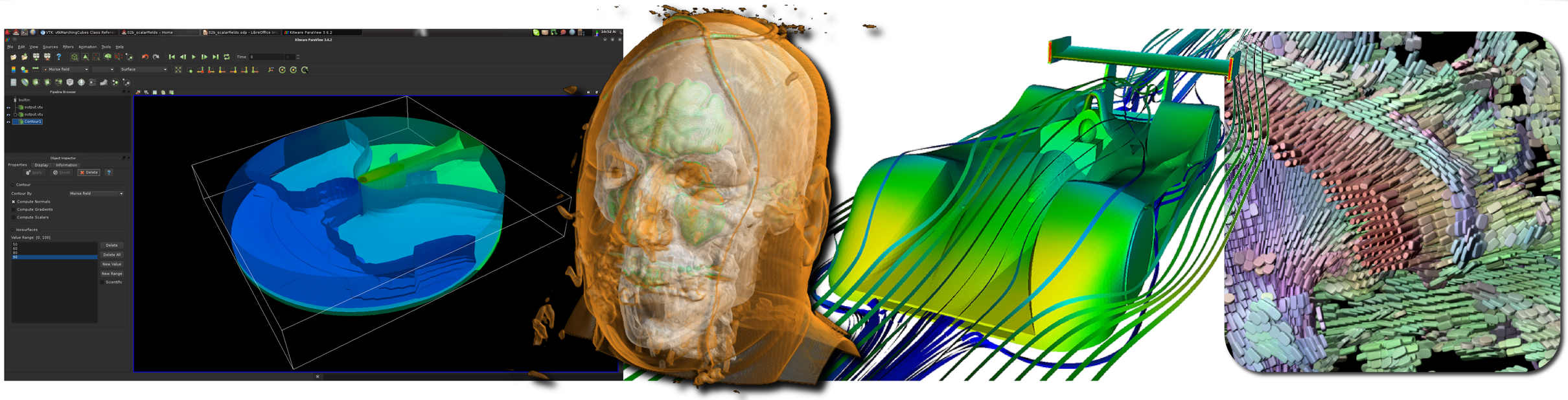
What?

- Generation of intelligible and interactive graphical representations

What?

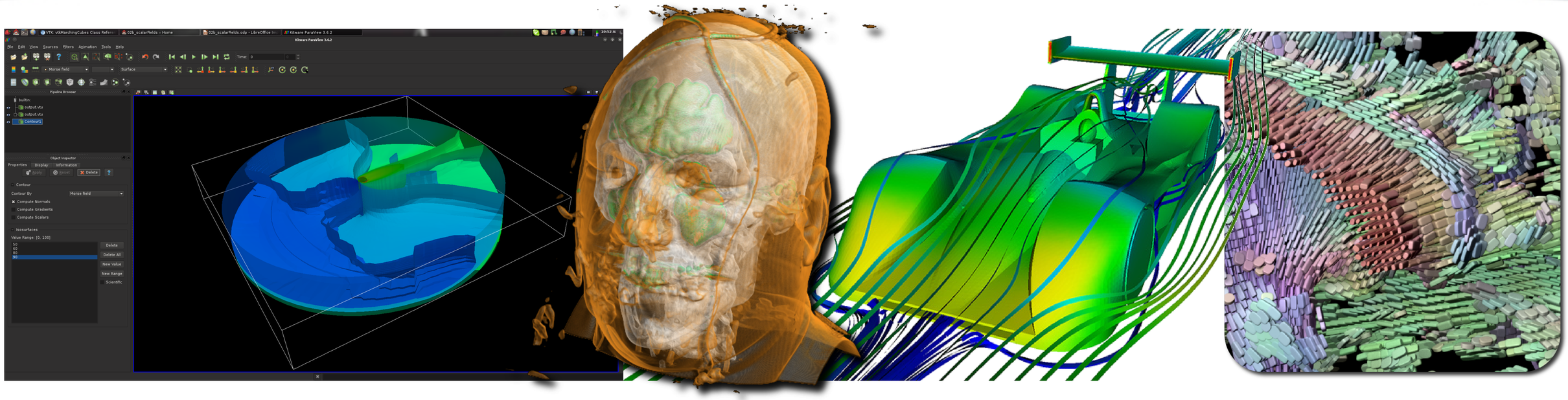
- Generation of intelligible and interactive graphical representations
  - Of scientific data

What?



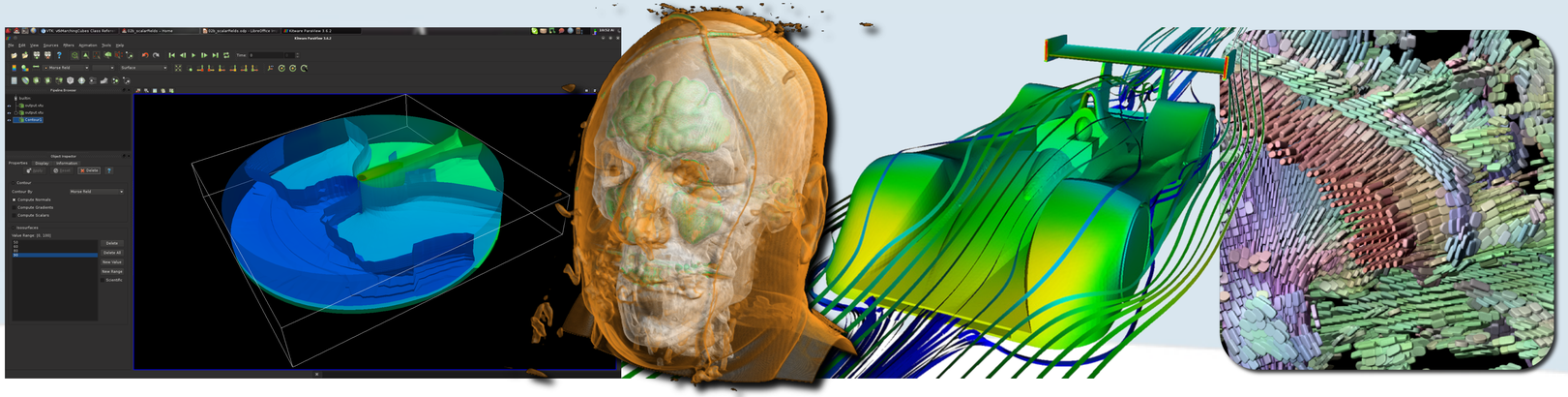
- Generation of intelligible and interactive graphical representations
  - Of scientific data

What?



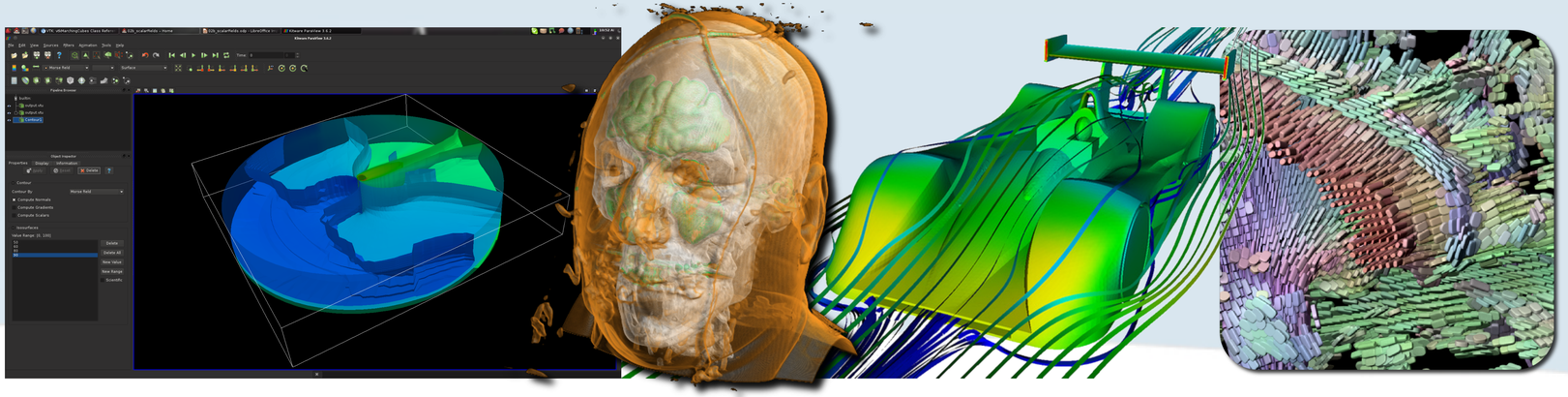
- Generation of intelligible and interactive graphical representations
  - Of scientific data
- Computer graphics for grown-ups

What?



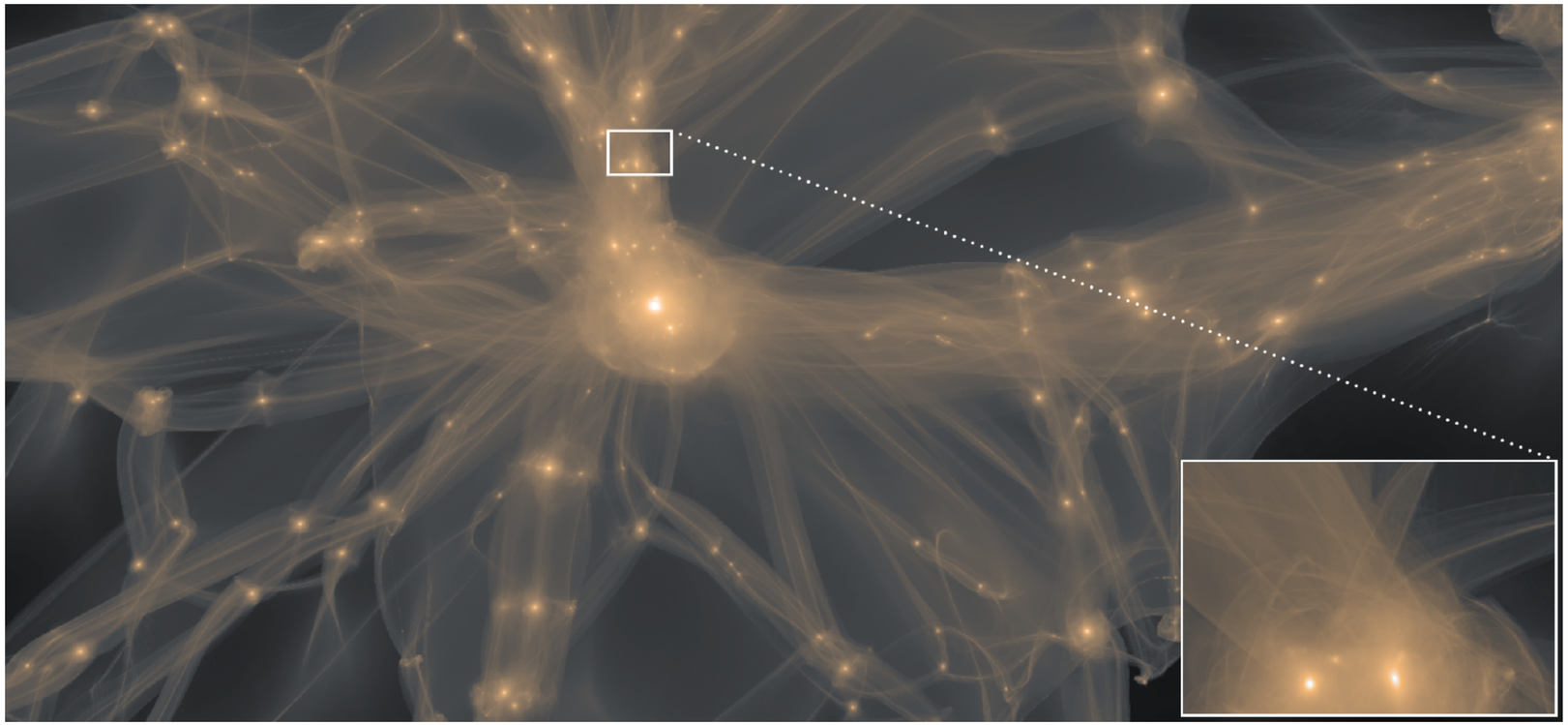
- Scientific data-sets
  - Results of simulations

What?

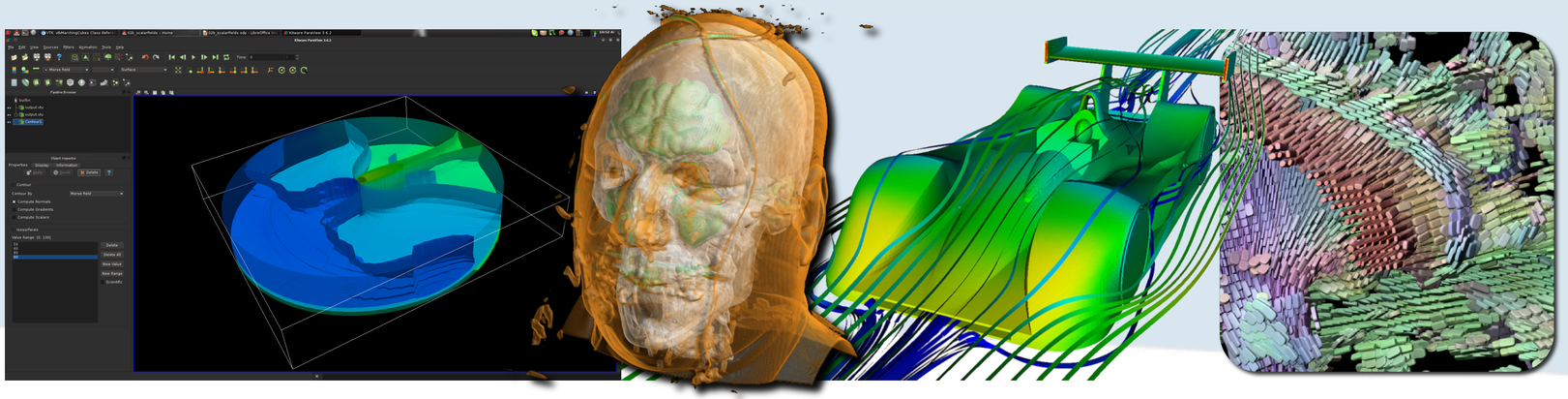


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics

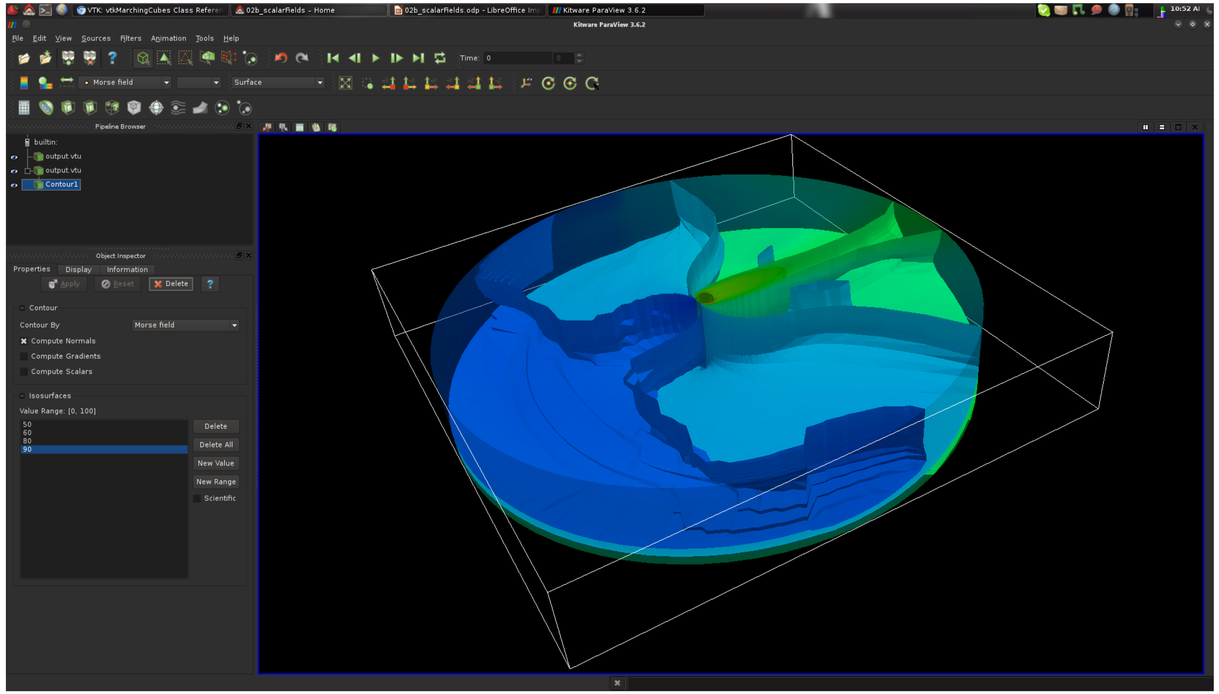
[Kaehler]



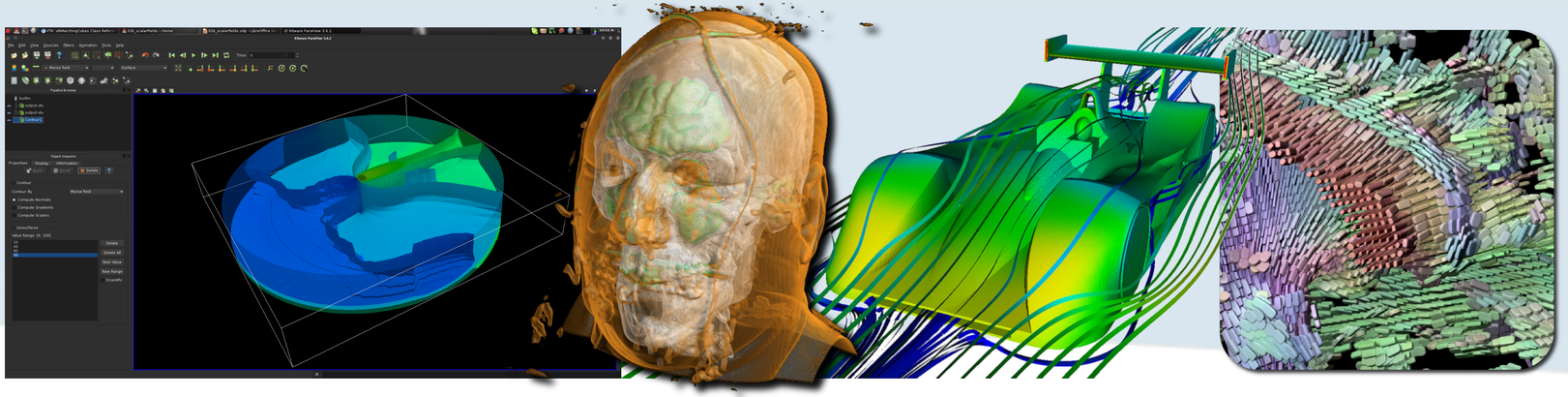
What?



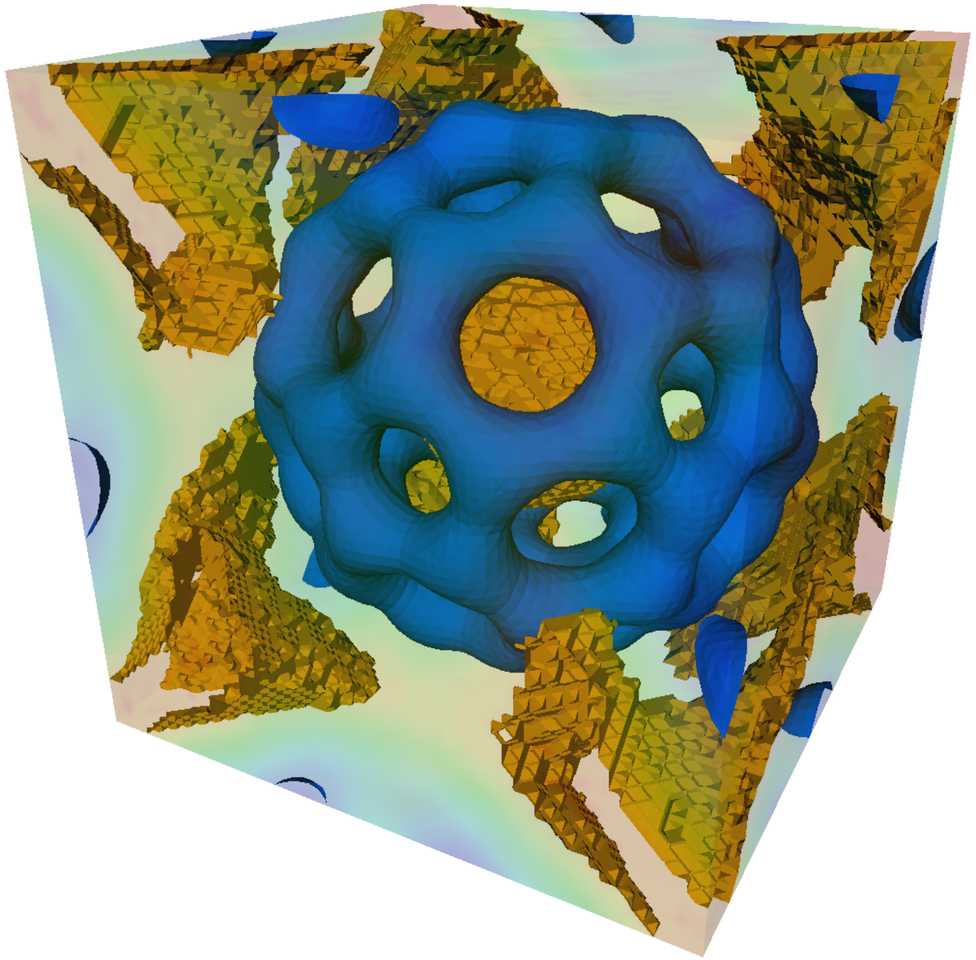
- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics



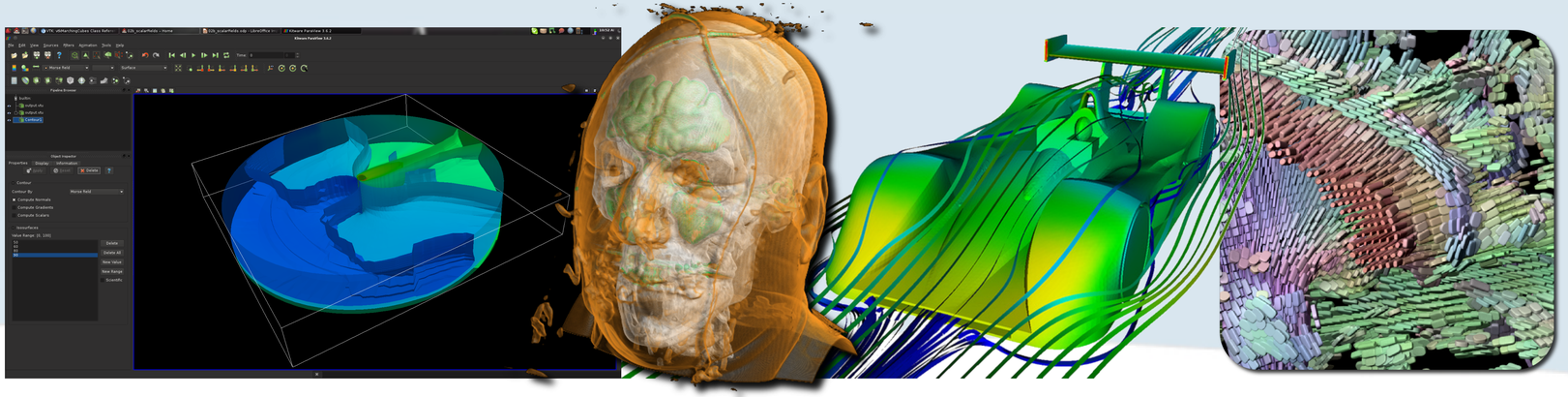
What?



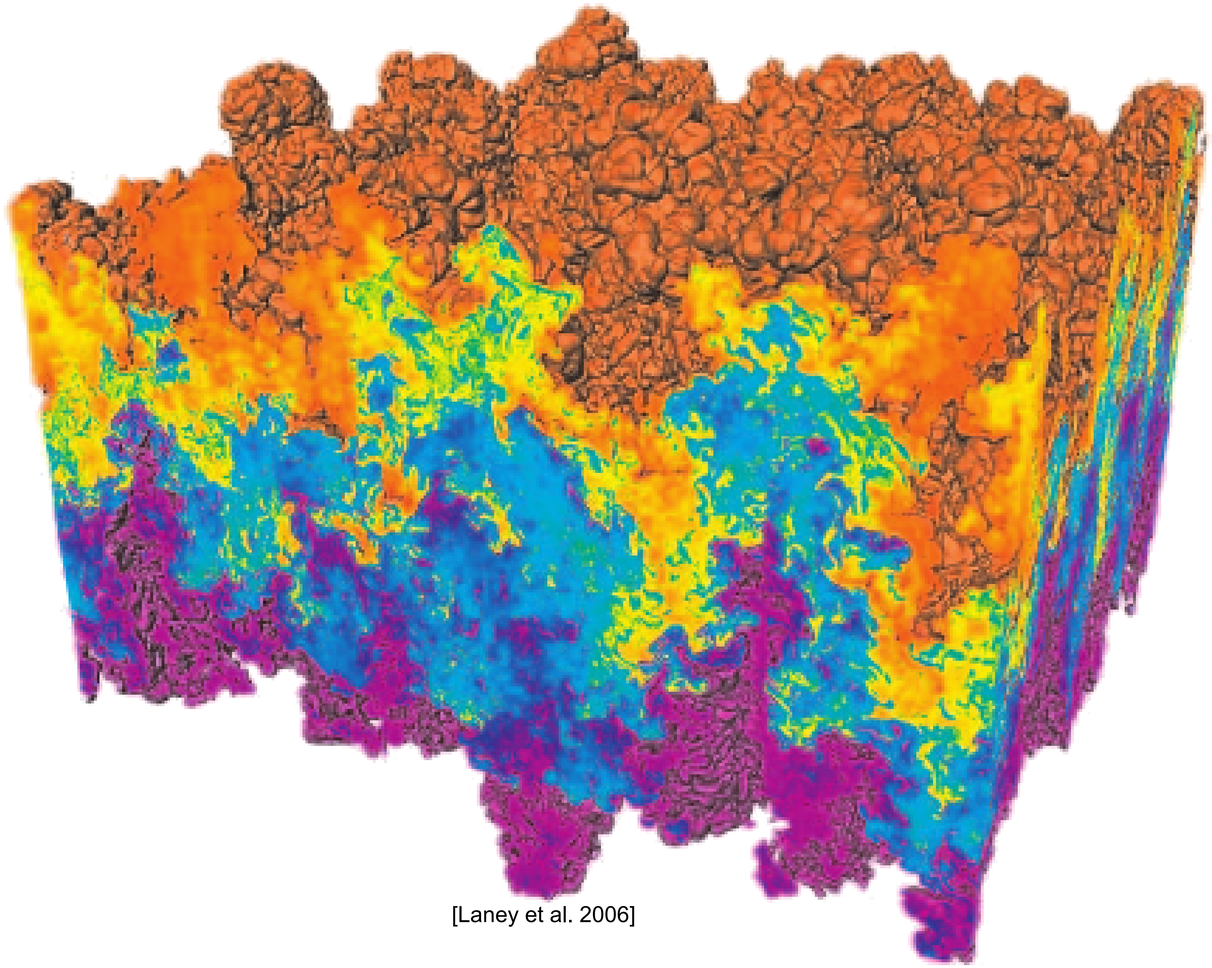
- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics



What?

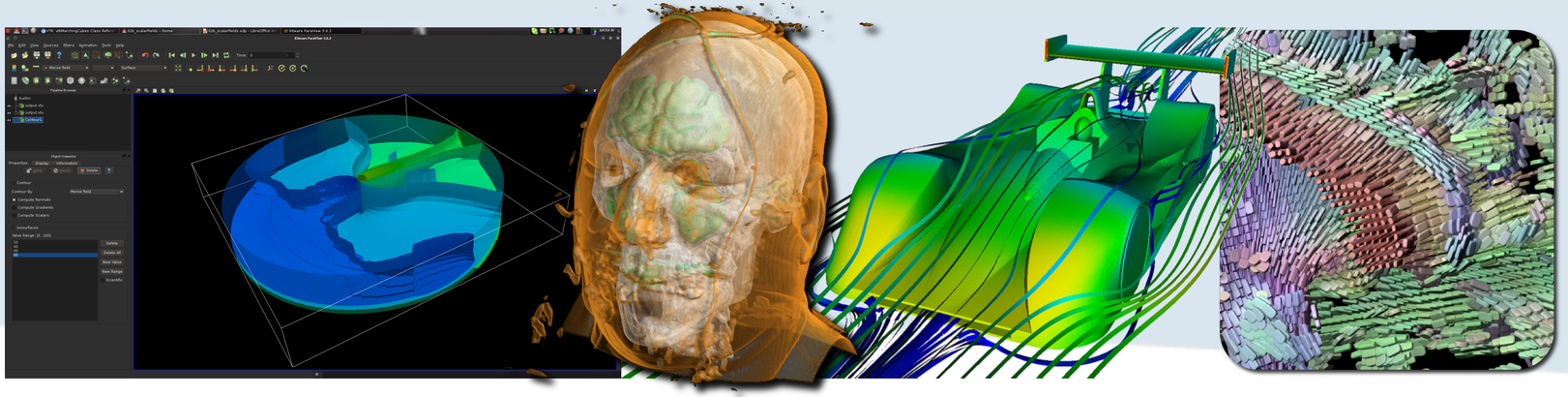


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics

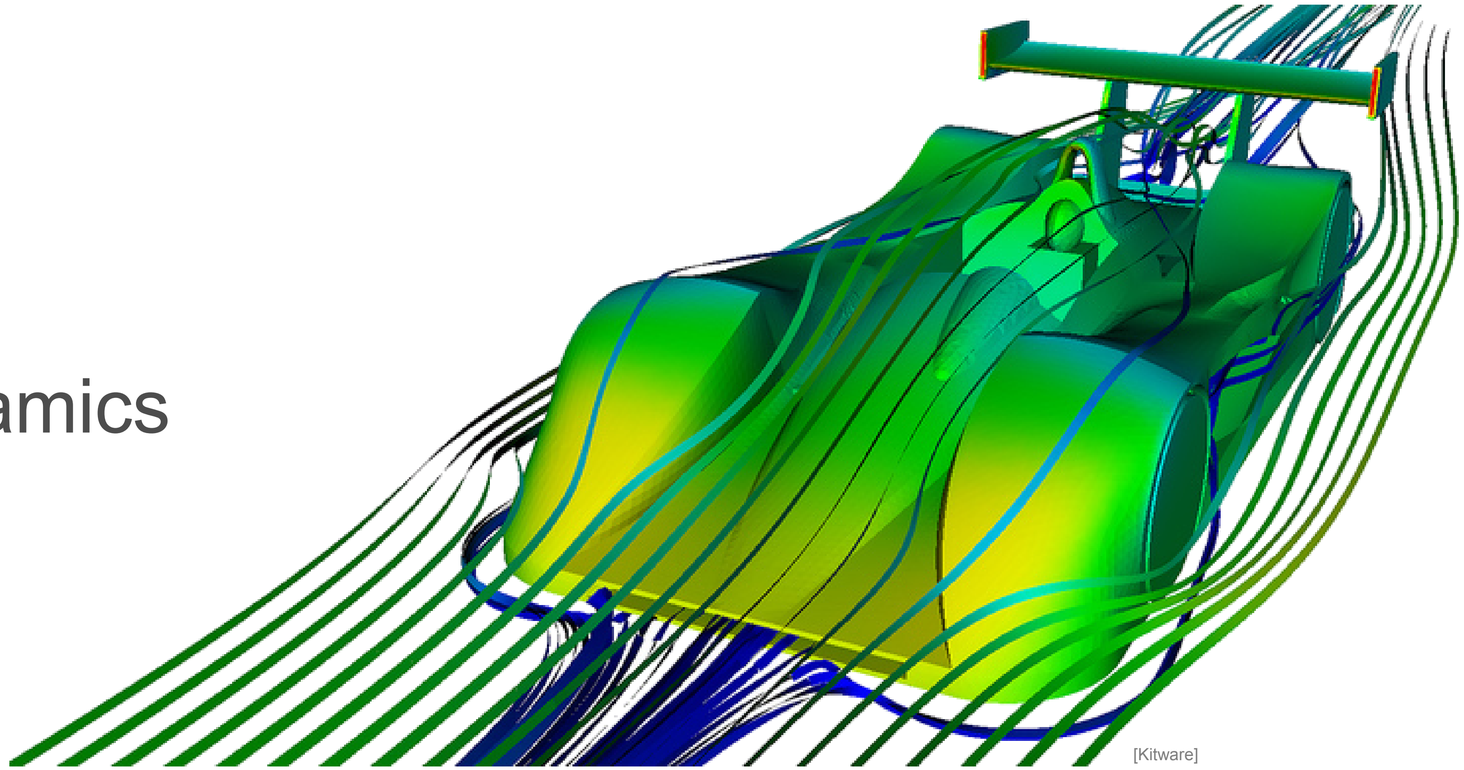


[Laney et al. 2006]

What?

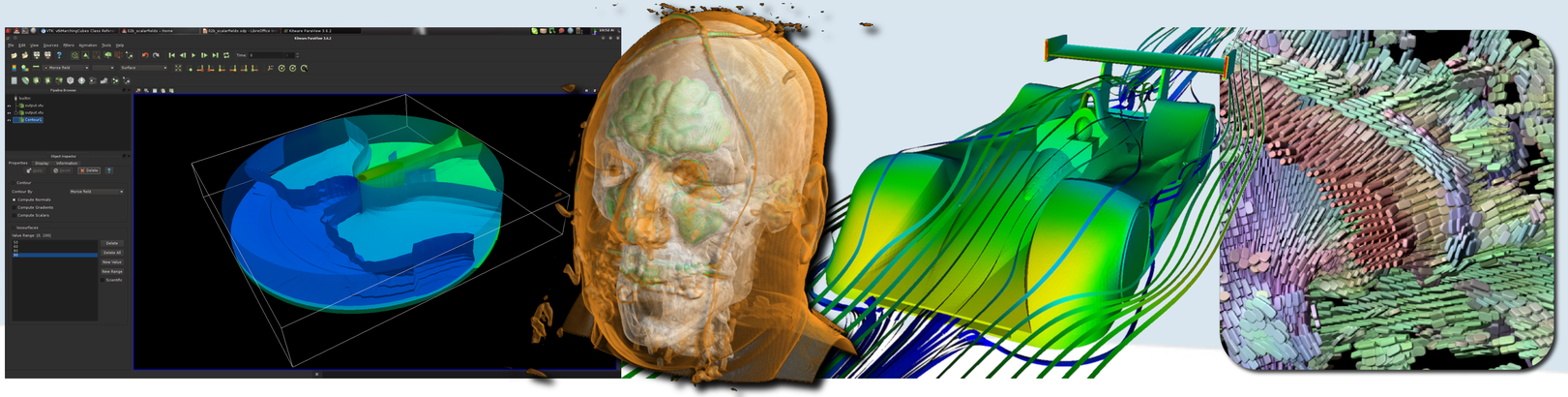


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics

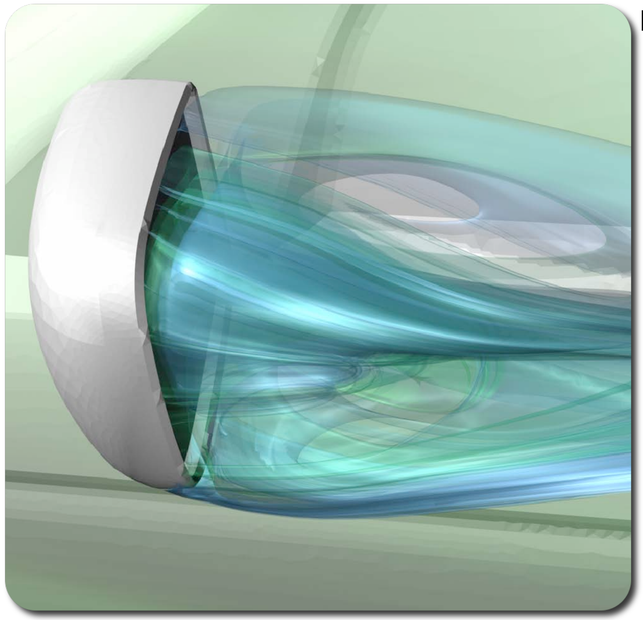


[Kitware]

What?

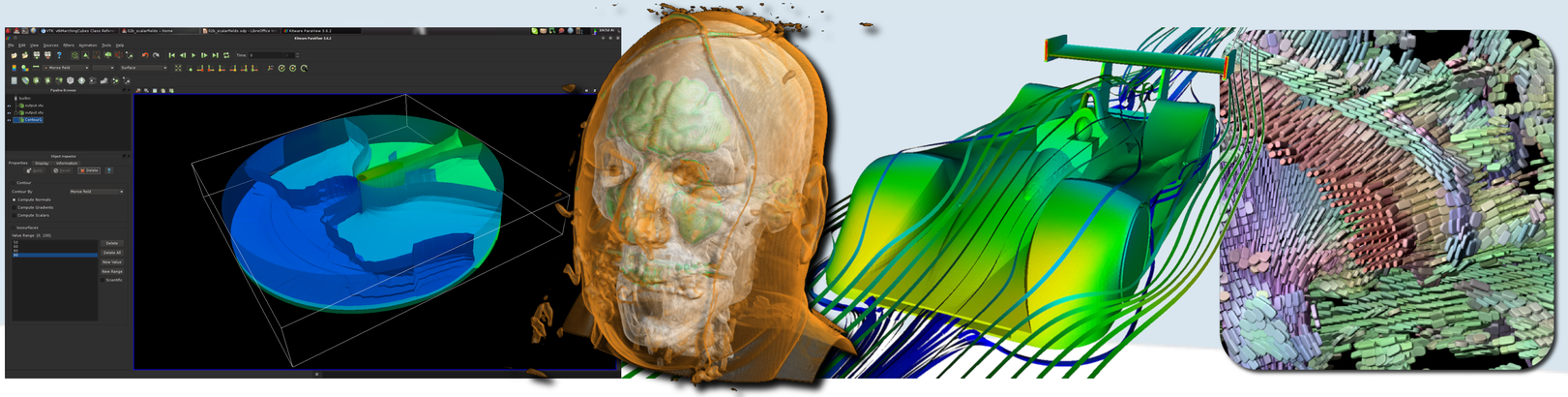


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics

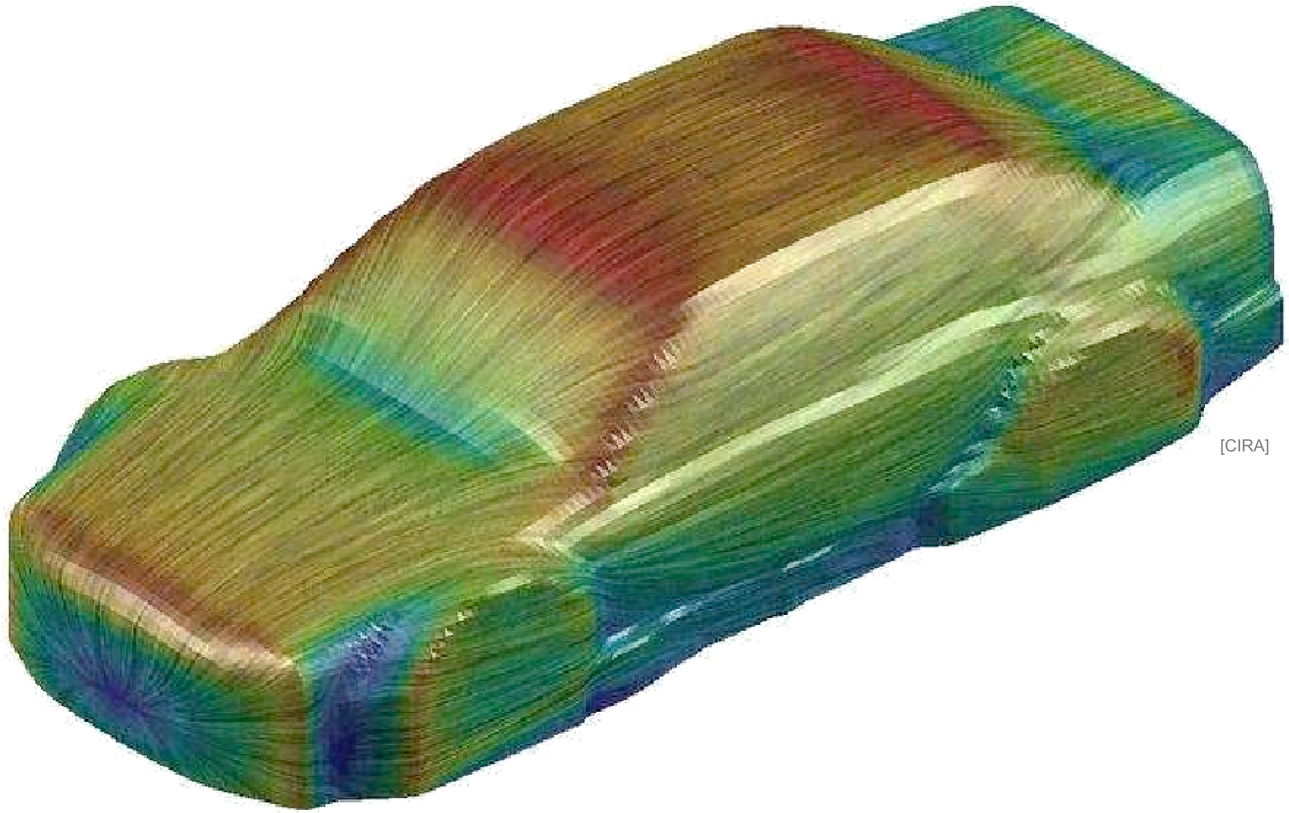


[GarthVIS08]

What?

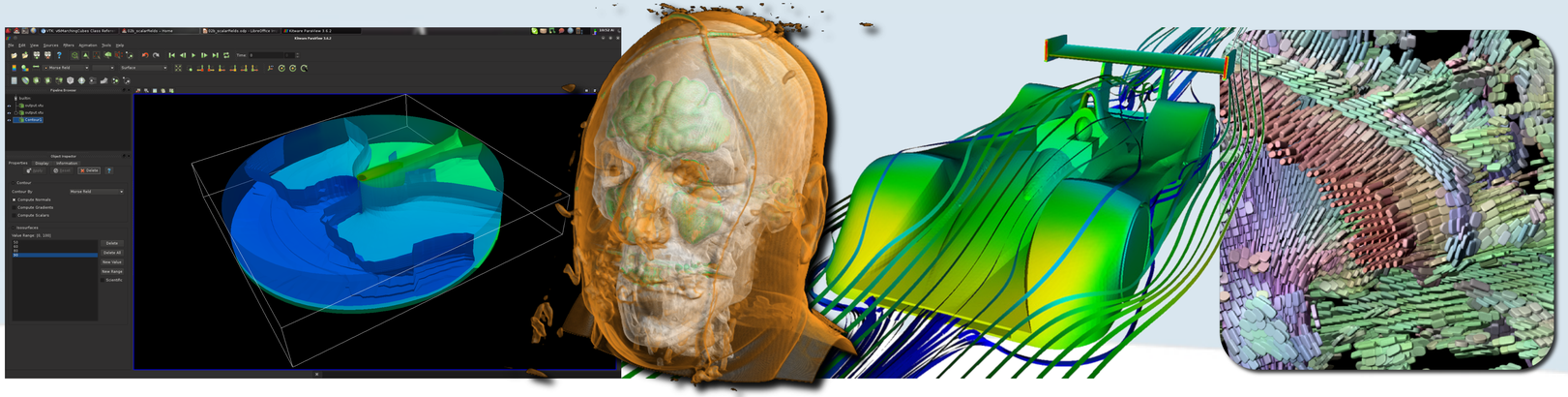


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics

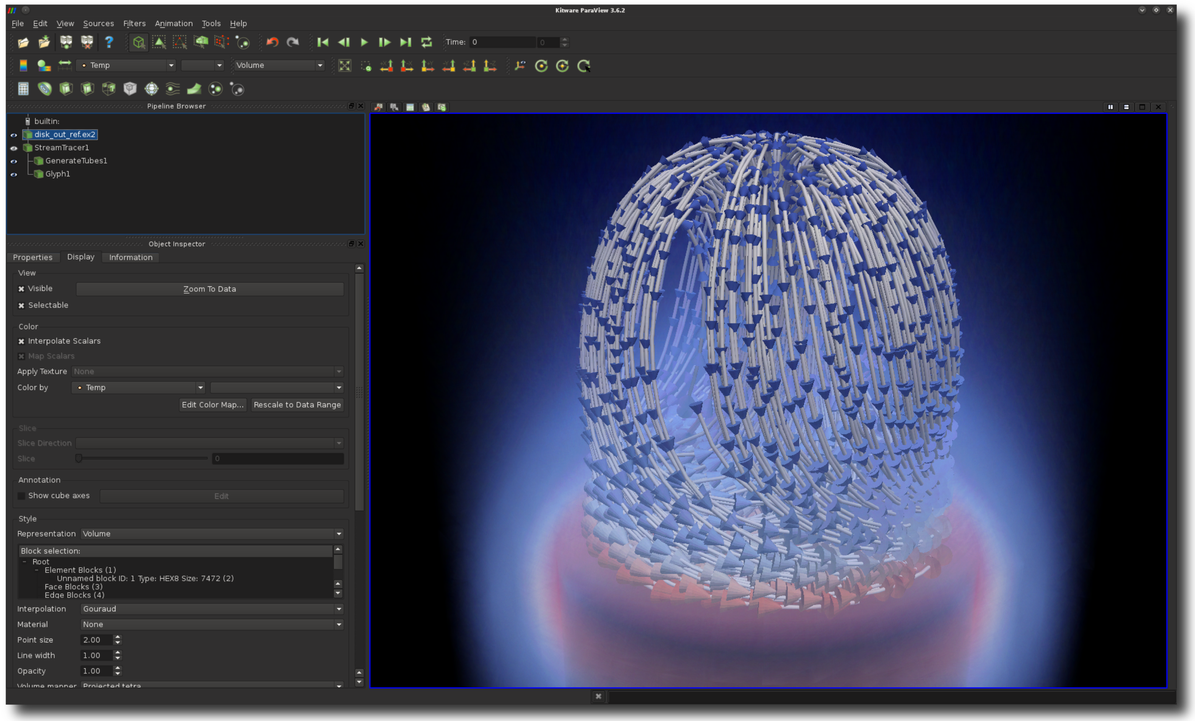


[CIRA]

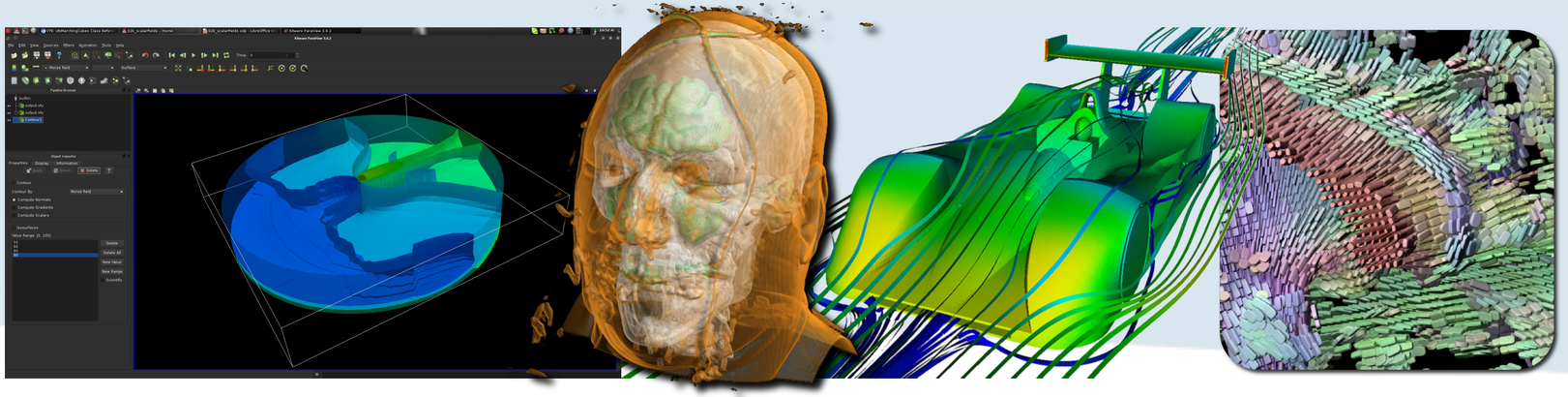
What?



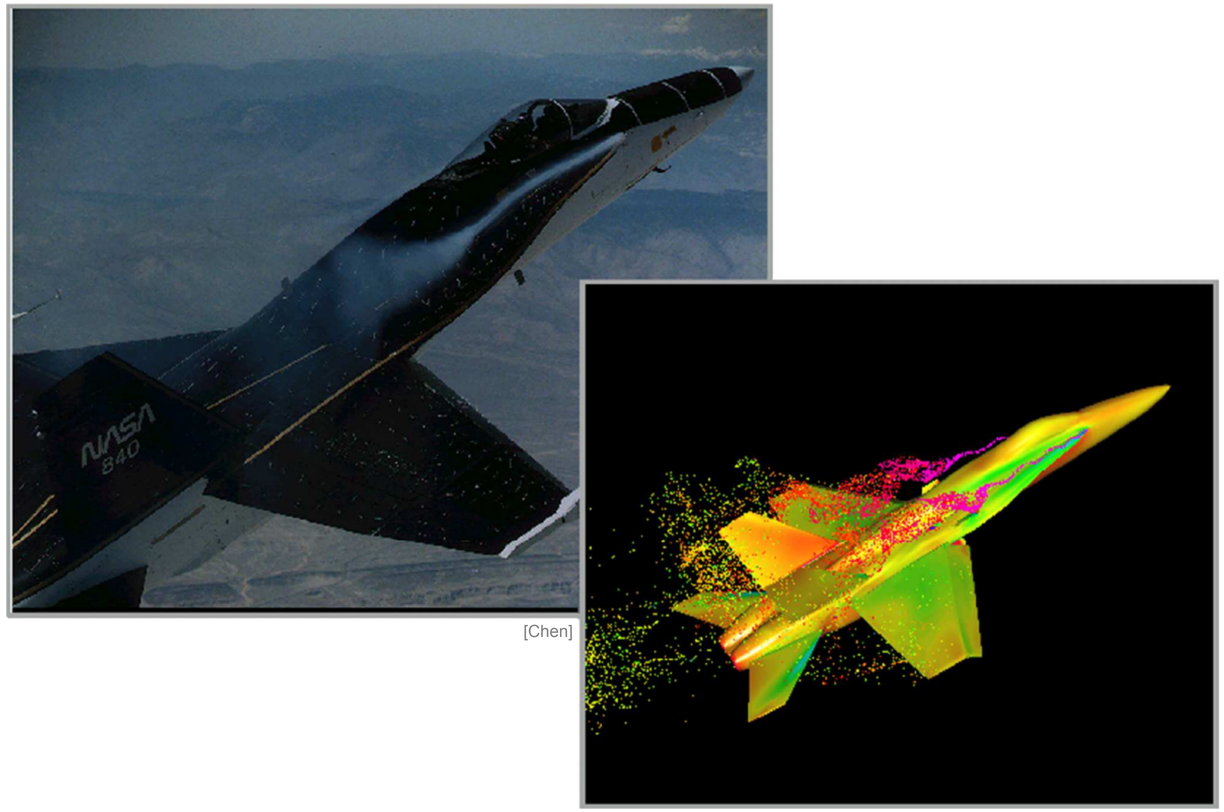
- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics



What?

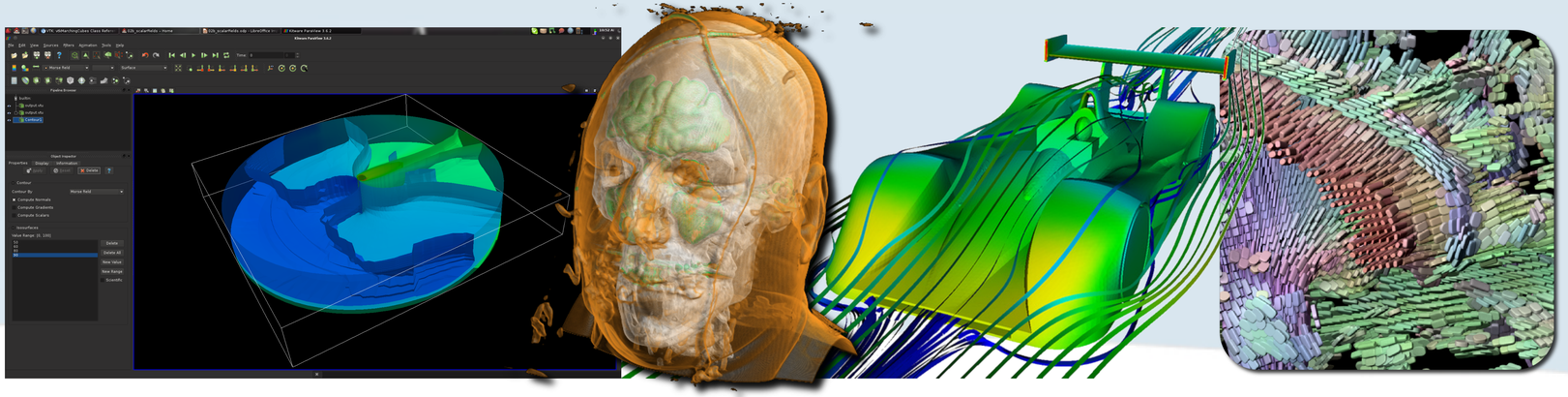


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics

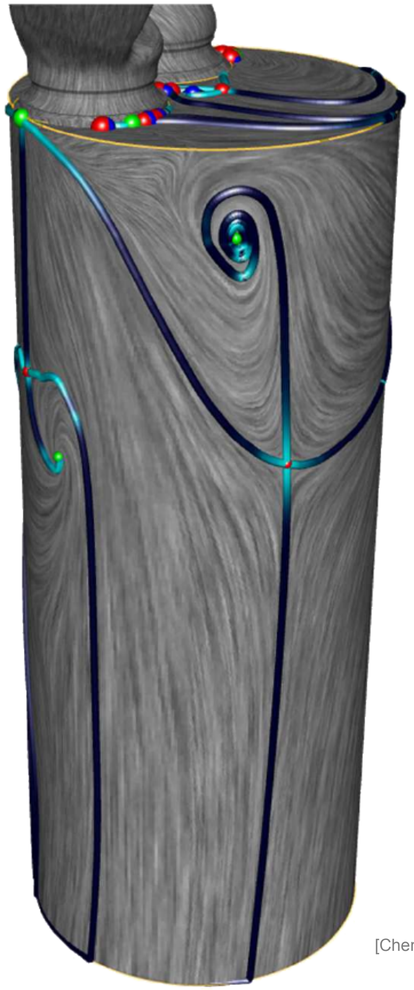


[Chen]

What?

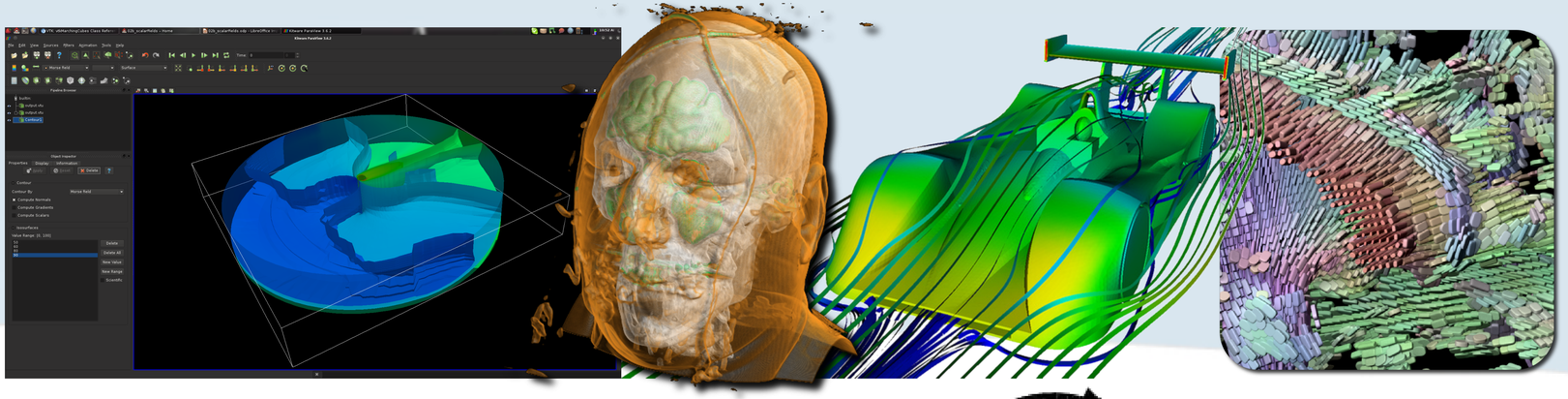


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics
    - Computer assisted design

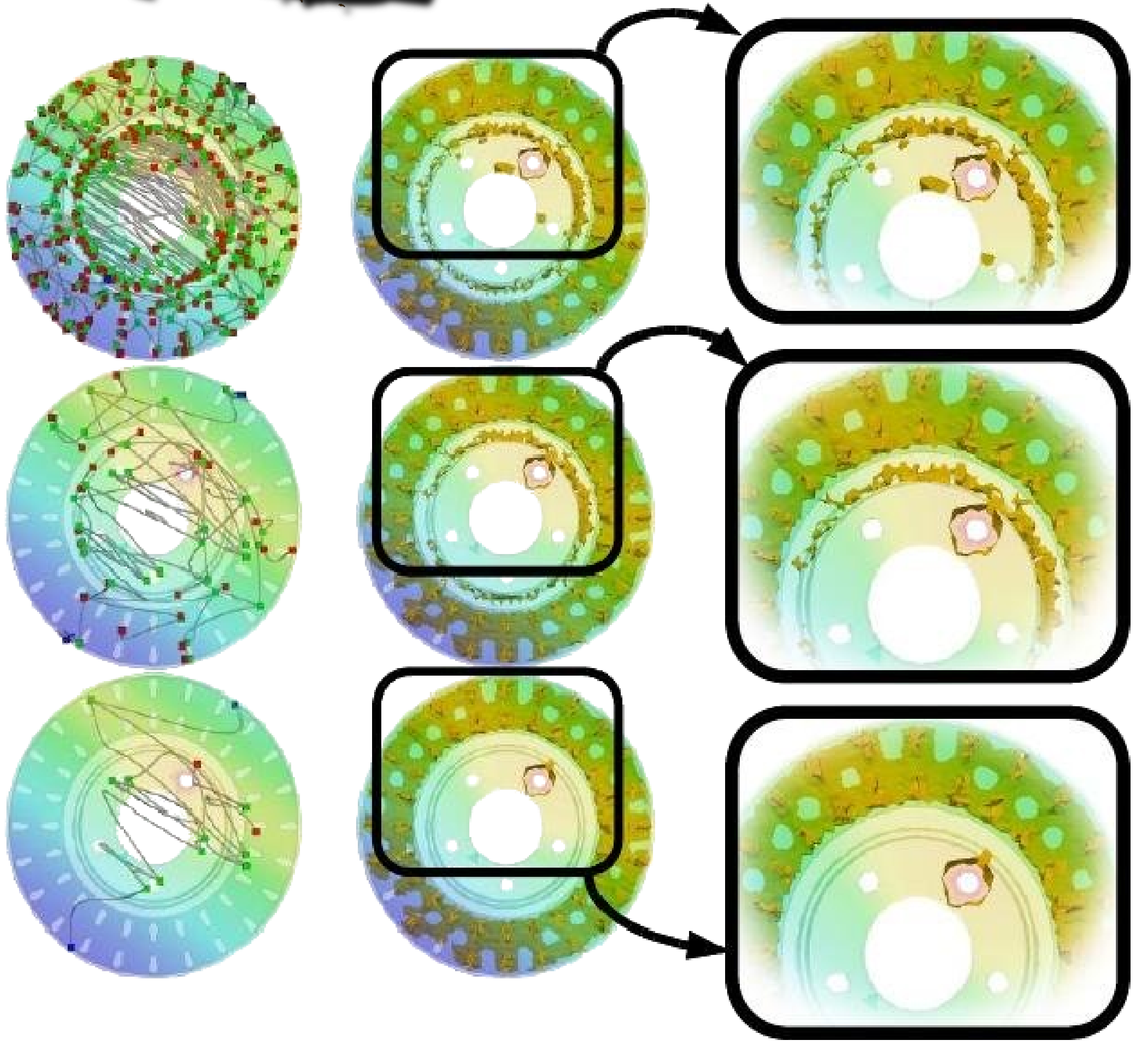


[Chen]

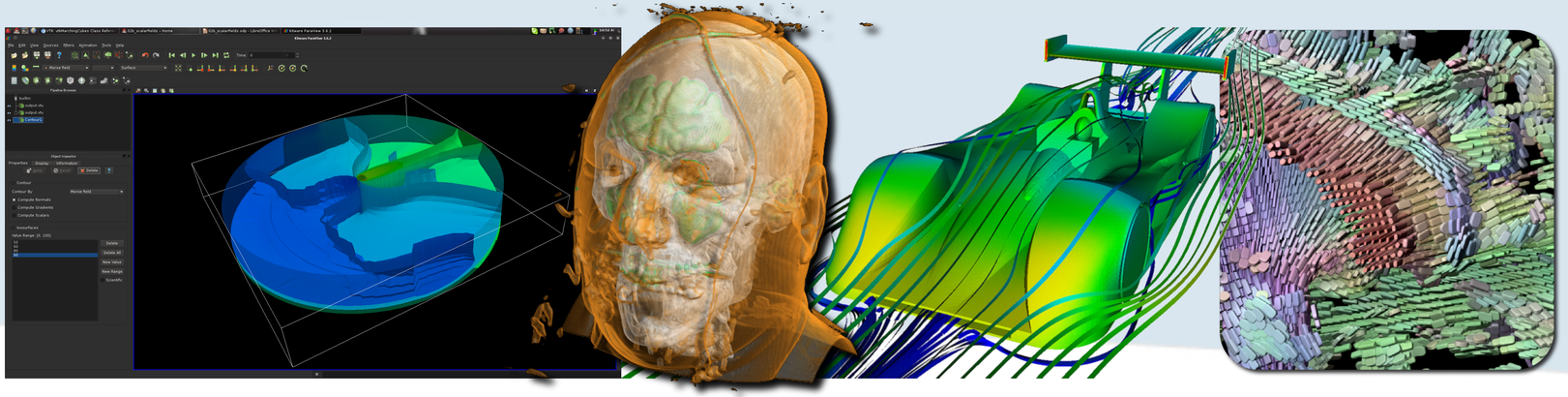
What?



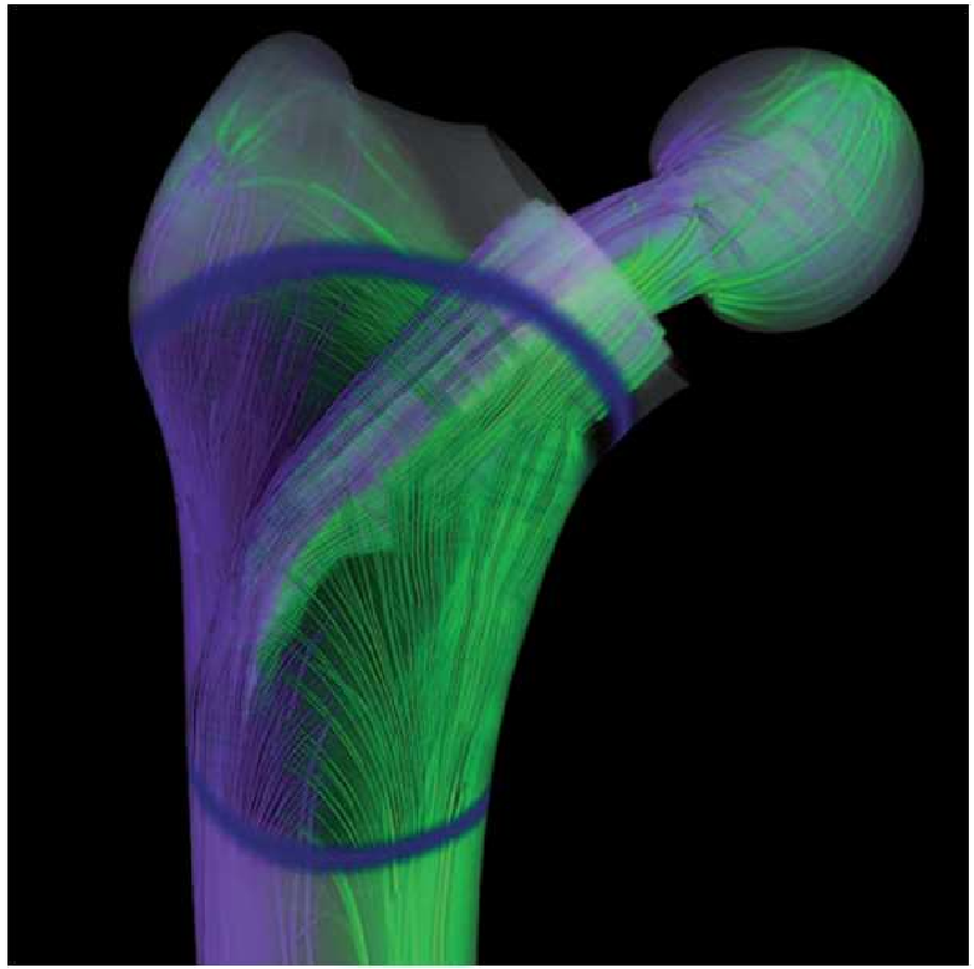
- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics
    - Computer assisted design



What?

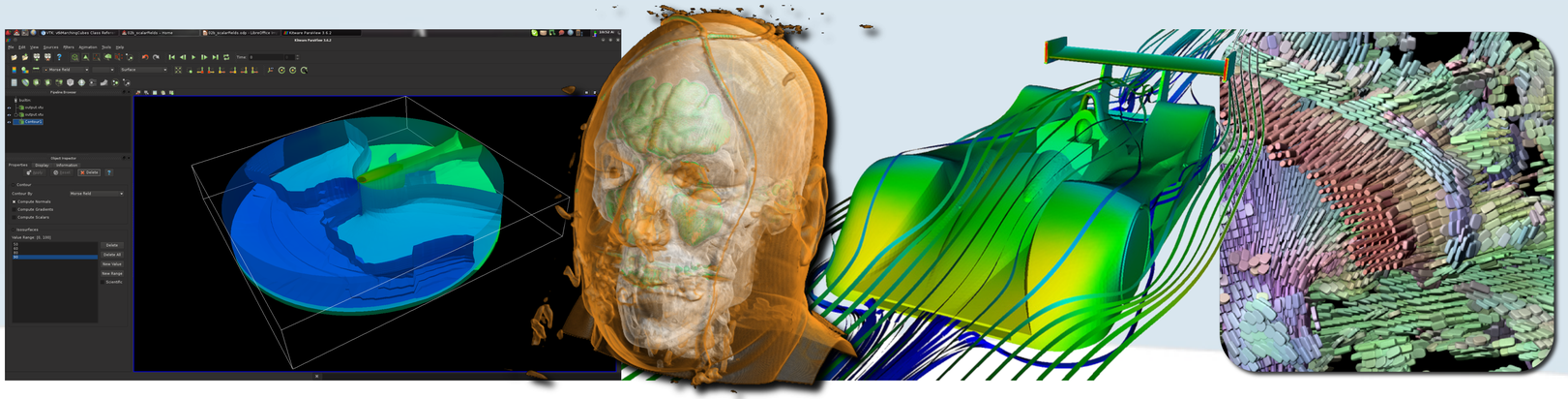


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics
    - Computer assisted design

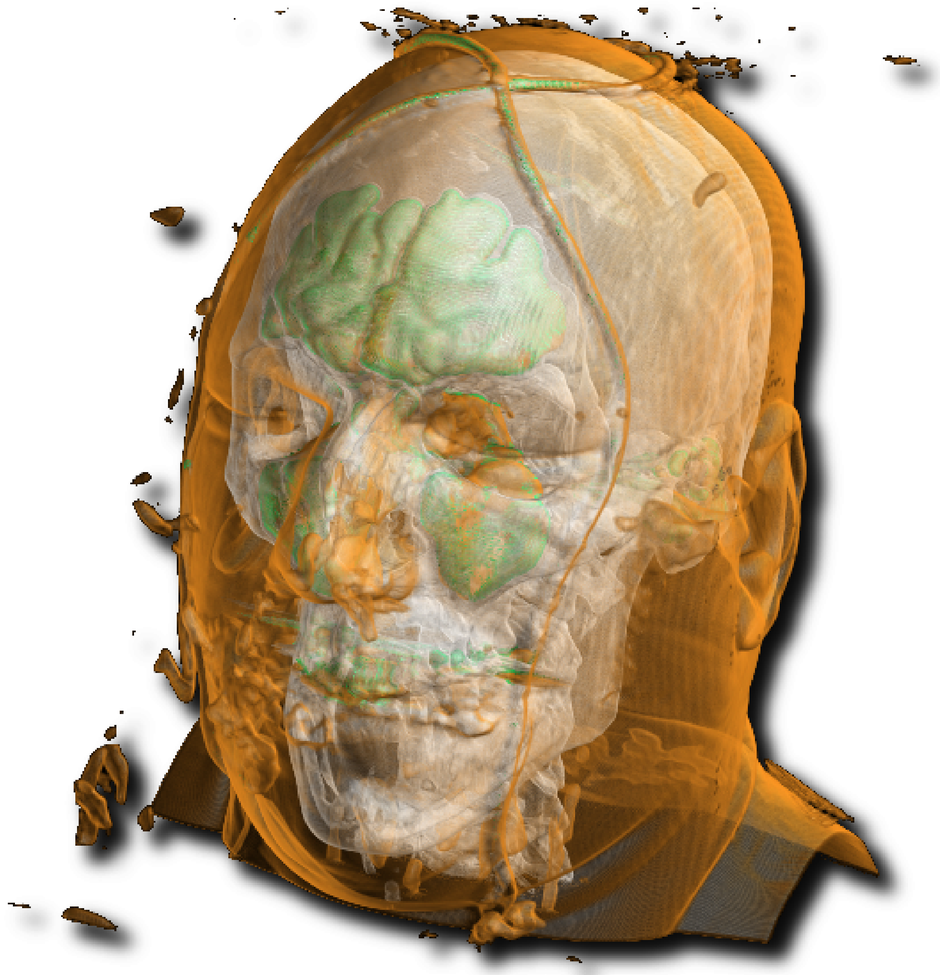


[Dick et al 09]

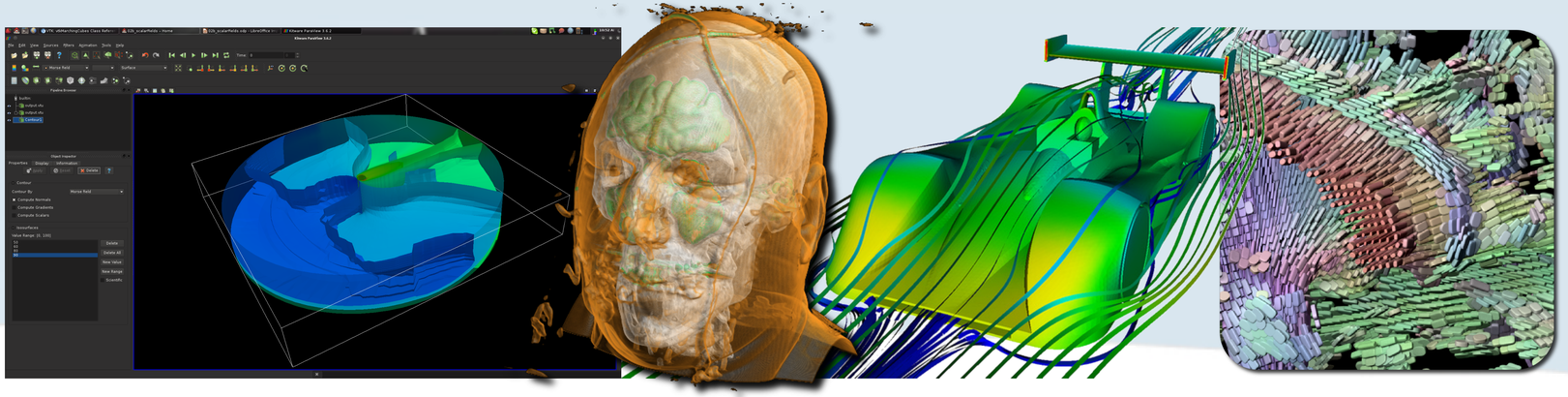
What?



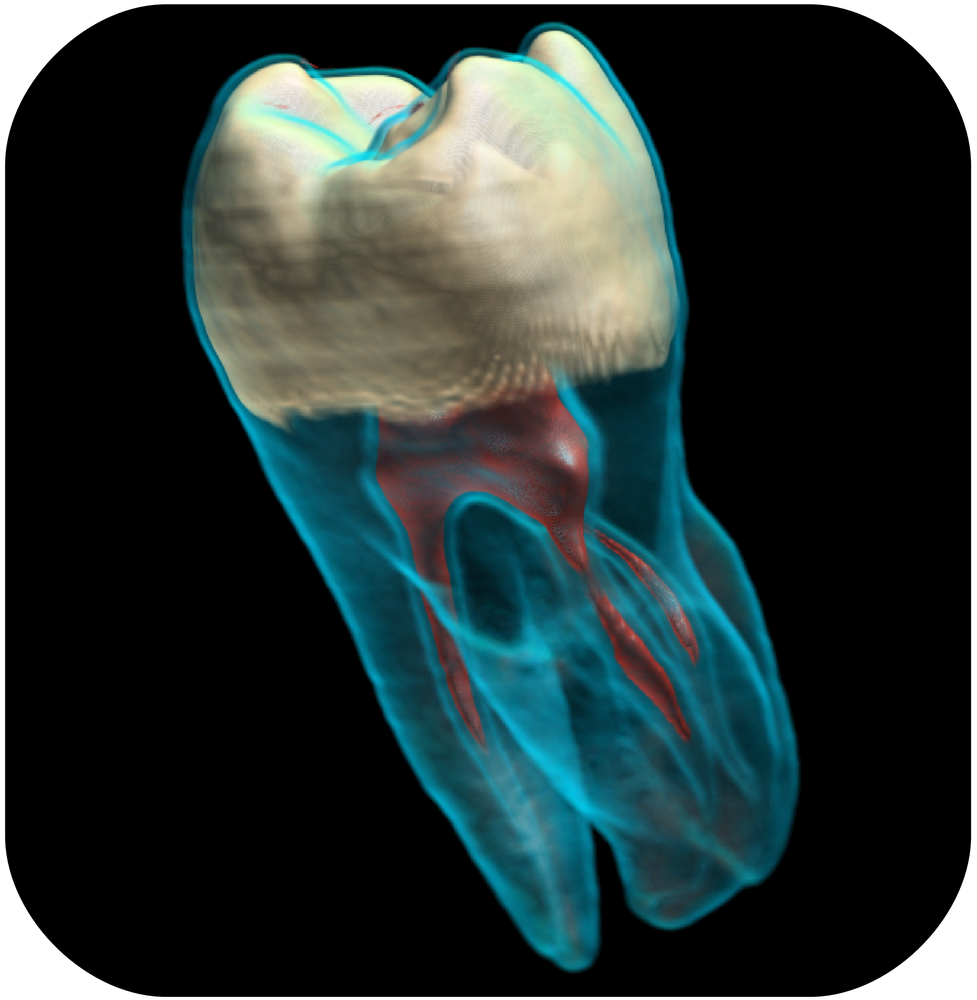
- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics
    - Computer assisted design
  - Results of acquisitions
    - Medical imaging



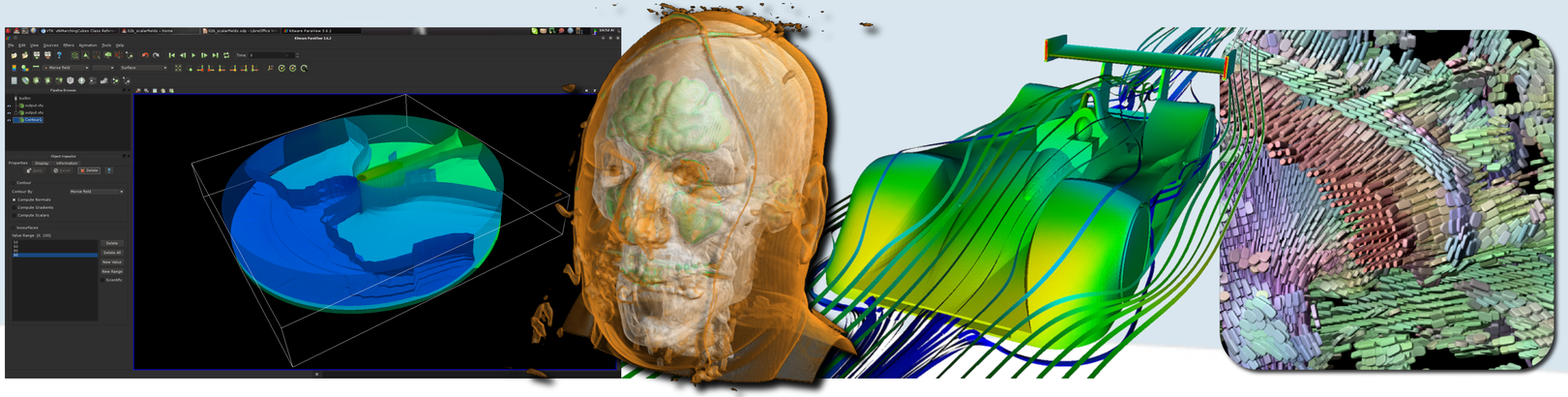
What?



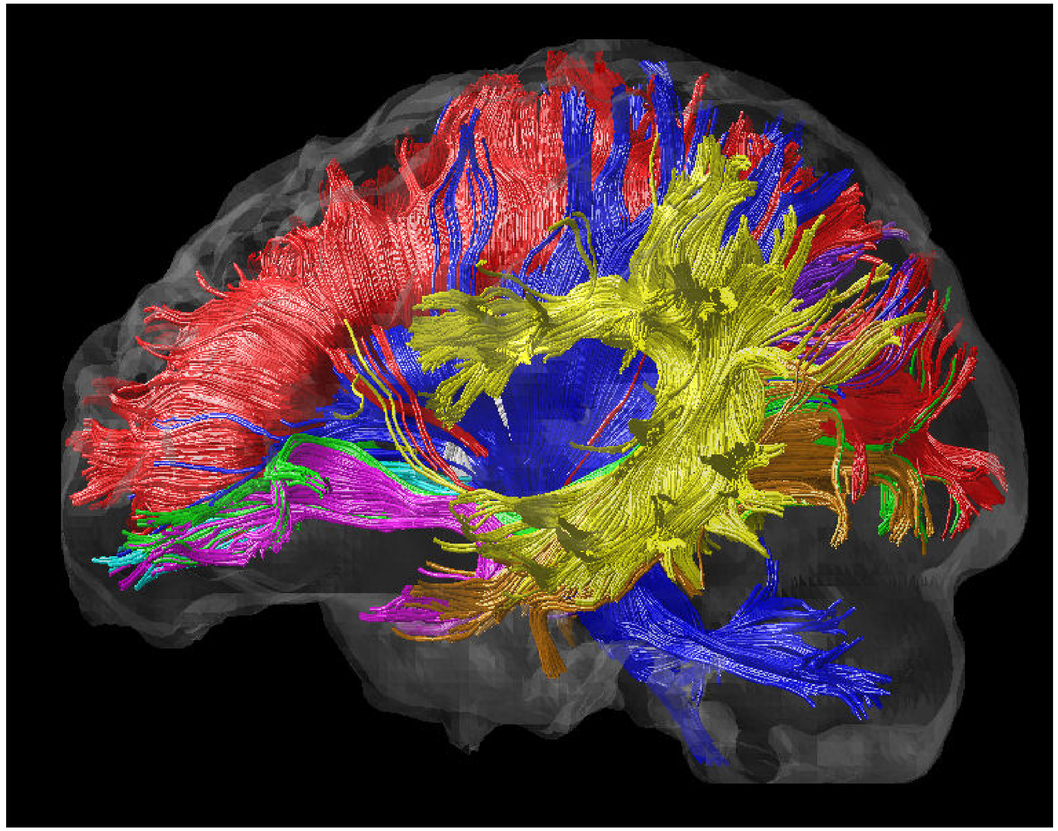
- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics
    - Computer assisted design
  - Results of acquisitions
    - Medical imaging



What?

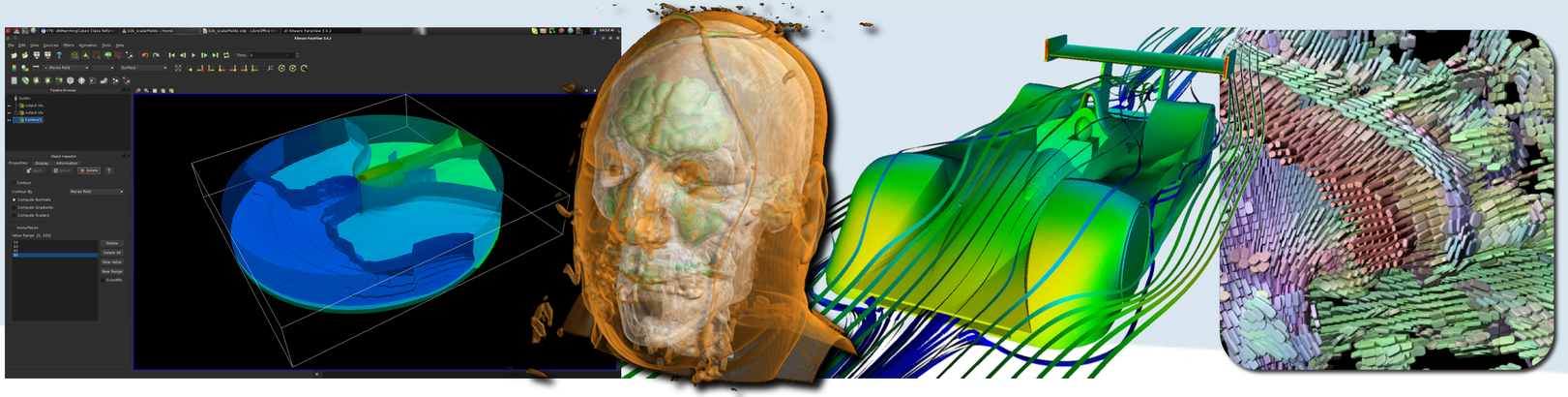


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics
    - Computer assisted design
  - Results of acquisitions
    - Medical imaging

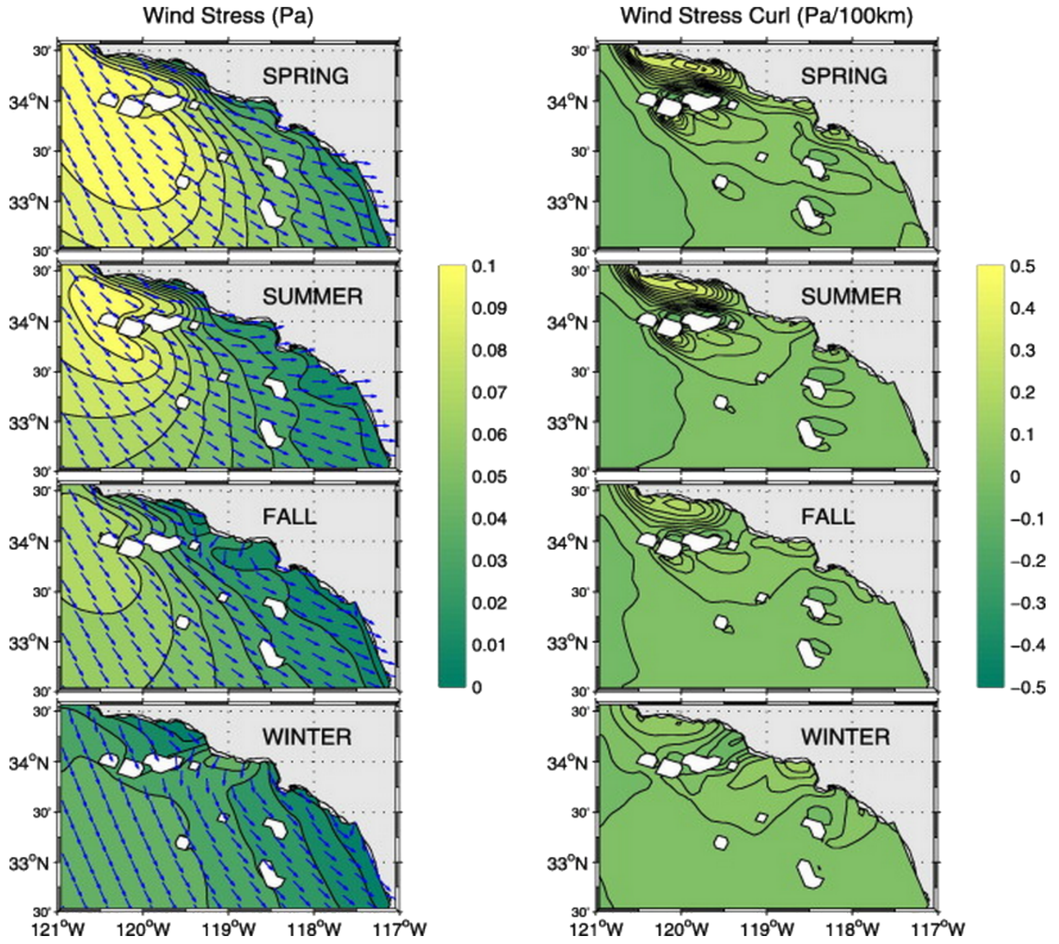


[http://www.neuroimaging.tau.ac.il]

What?

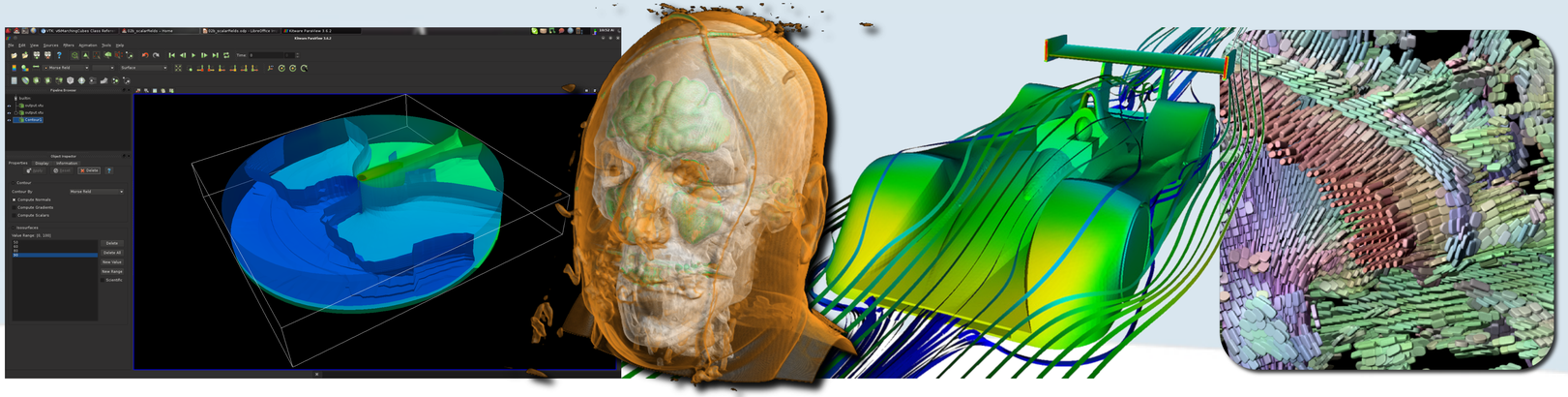


- Scientific data-sets
  - Results of simulations
    - Chemistry, Physics
    - Computational fluid dynamics
    - Computer assisted design
  - Results of acquisitions
    - Medical imaging
    - Seismology, oceanography, etc.



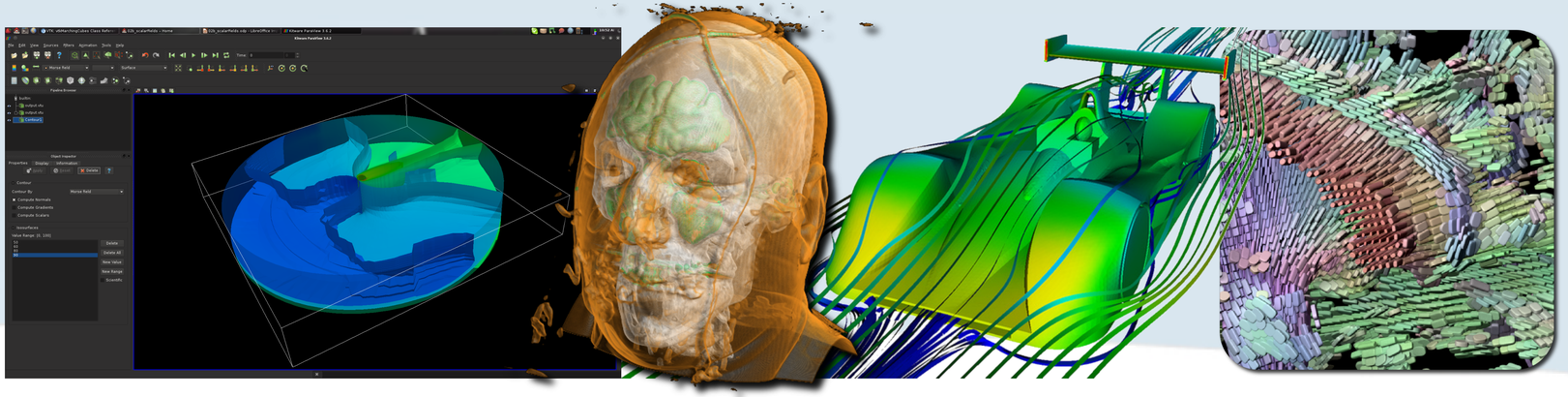
[Dong09]

What for?



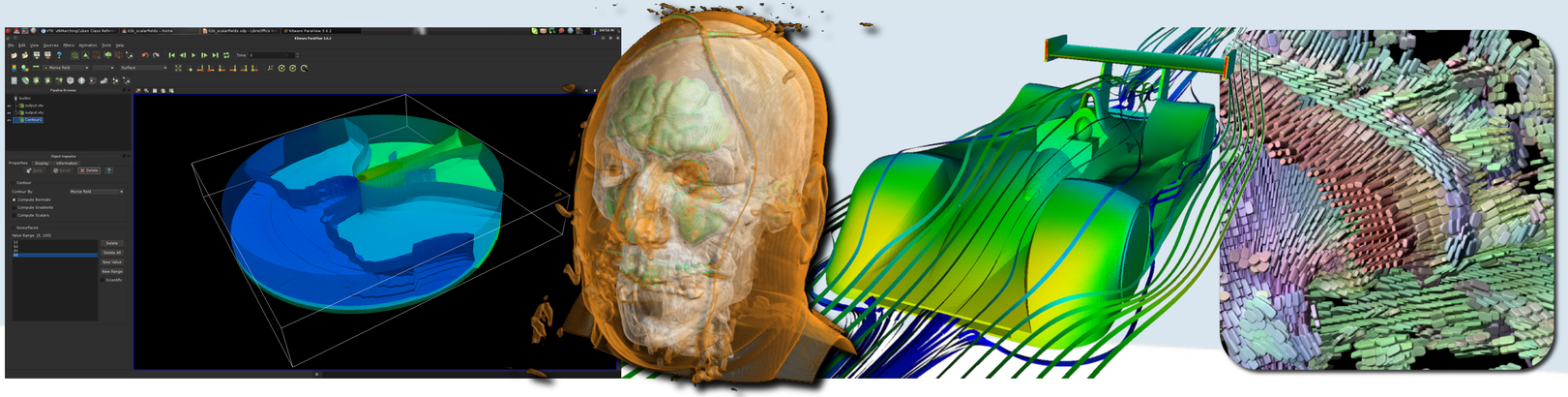
- Visual exploration of scientific data

What for?



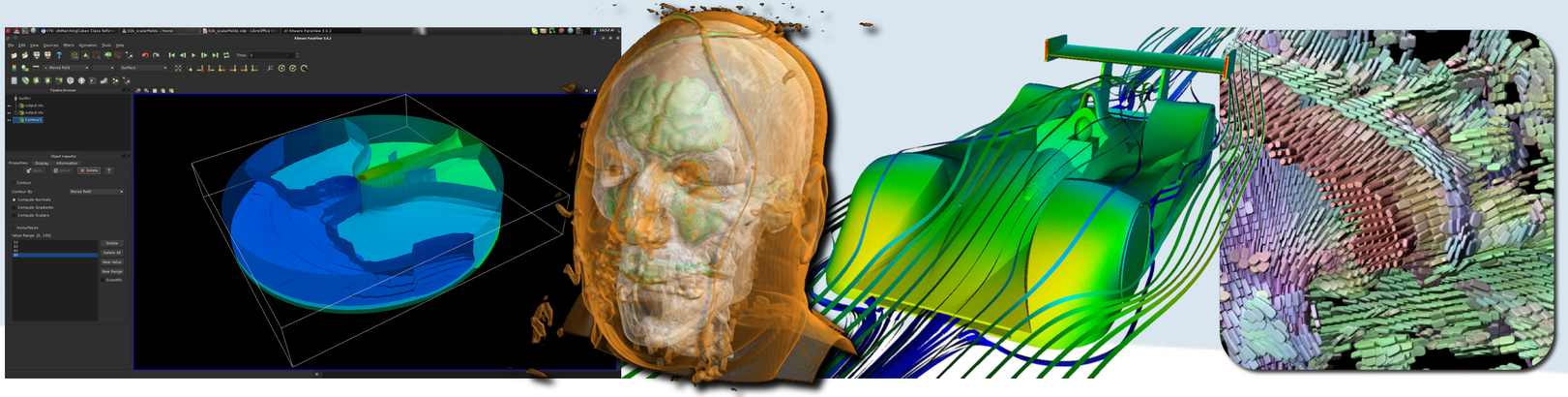
- Visual exploration of scientific data
  - Hypothesis formulation

What for?



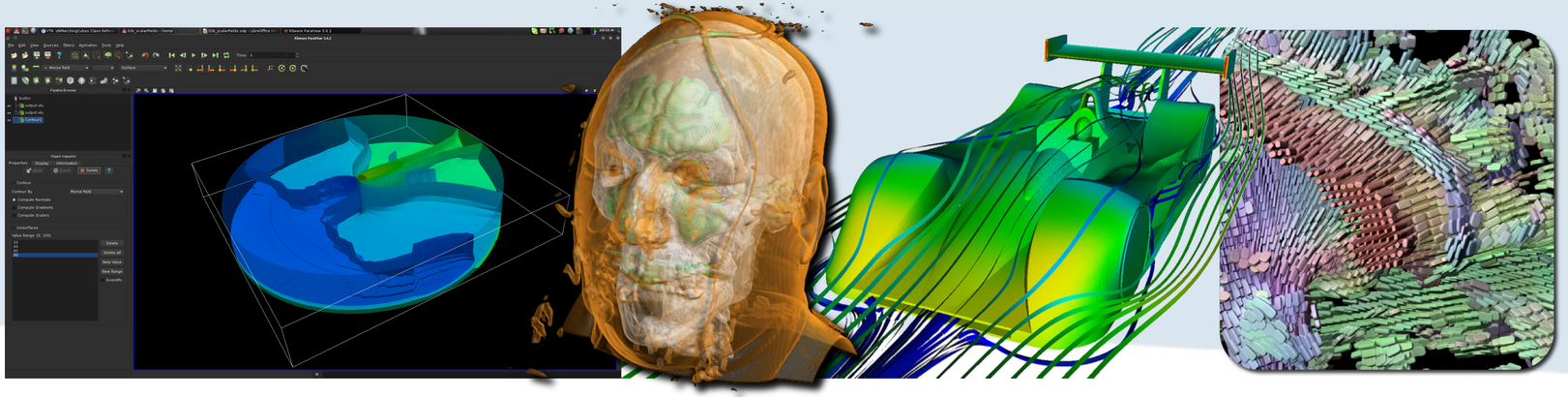
- Visual exploration of scientific data
  - Hypothesis formulation
  - Model verification

What for?



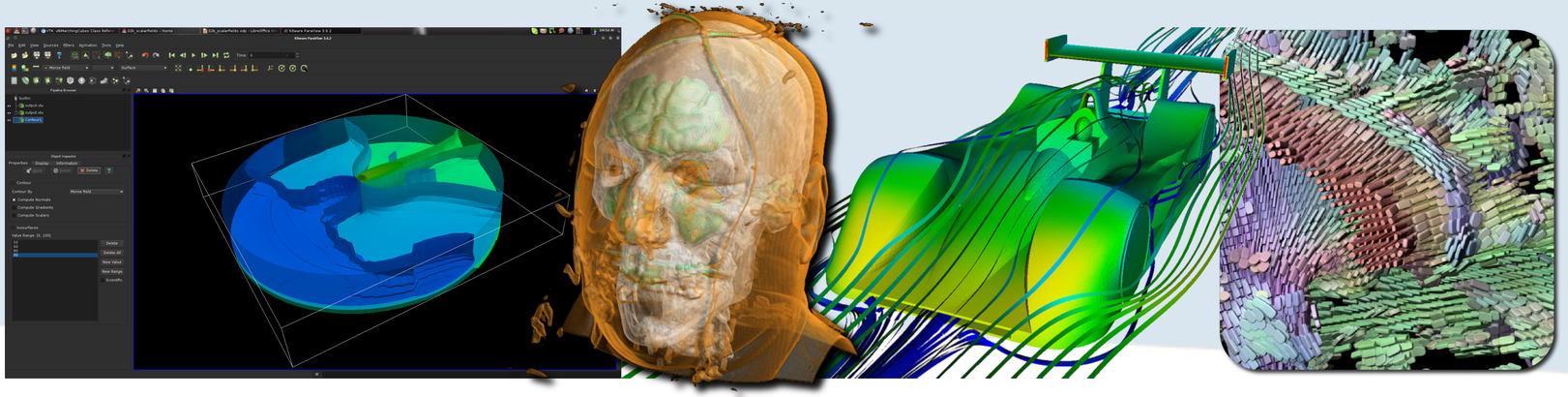
- Visual exploration of scientific data
  - Hypothesis formulation
  - Model verification
  - Intuition validation

What for?



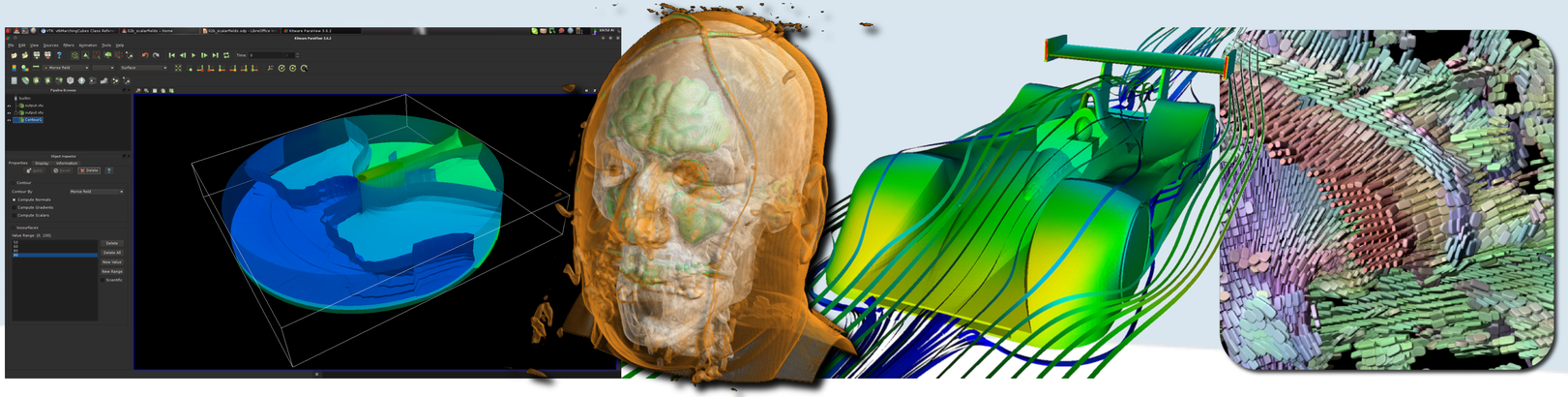
- Visual exploration of scientific data
  - Hypothesis formulation
  - Model verification
  - Intuition validation
- Geometrical analysis
  - Result interpretation

What for?

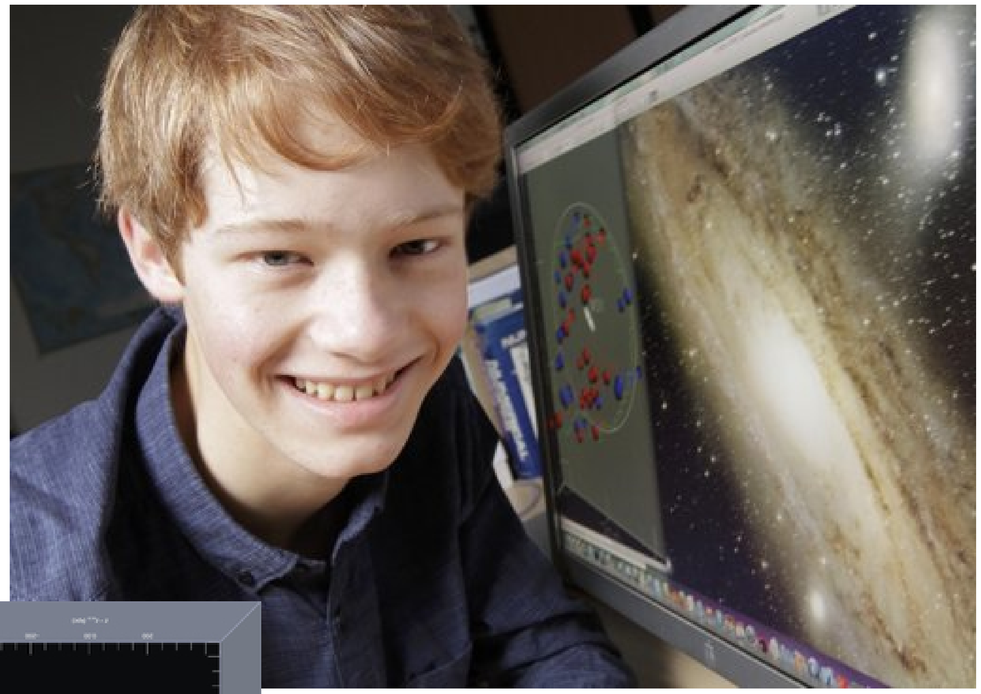


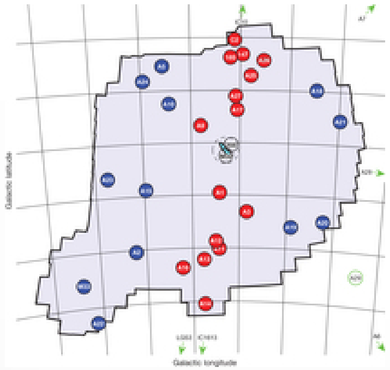
- Visual exploration of scientific data
  - Hypothesis formulation
  - Model verification
  - Intuition validation
- Geometrical analysis
  - Result interpretation
- Communication of scientific results
  - Interactive and graphical material

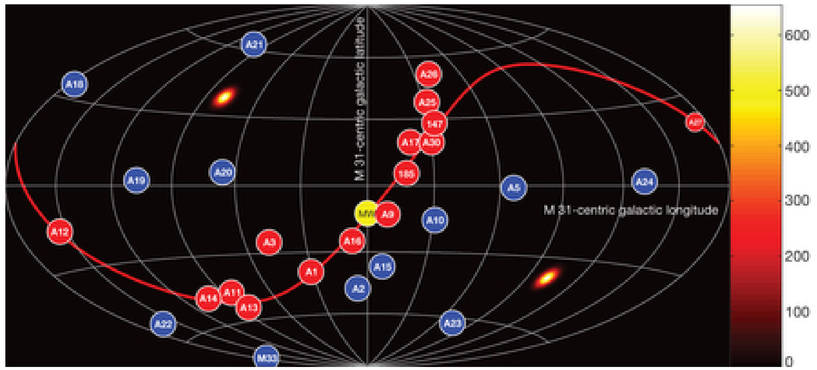
What for?

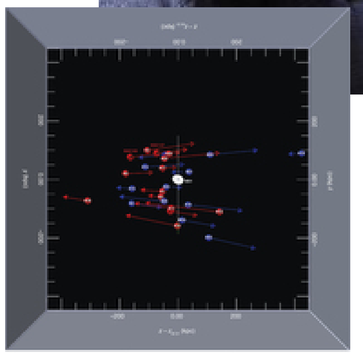


- Dwarf galaxies orbiting the Andromeda Galaxy
- *Nature*, January 3rd, 2013

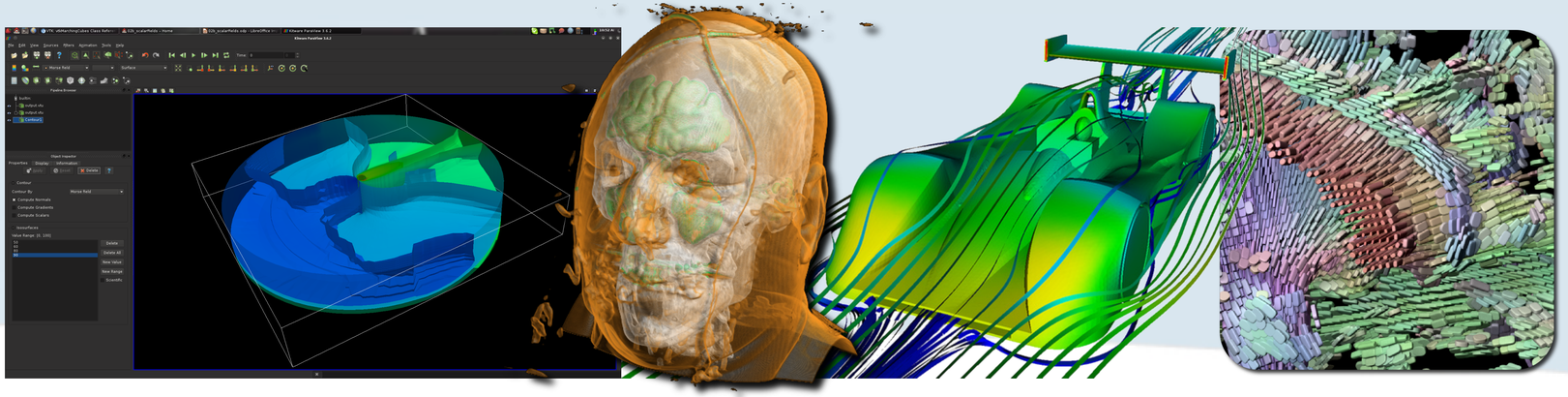




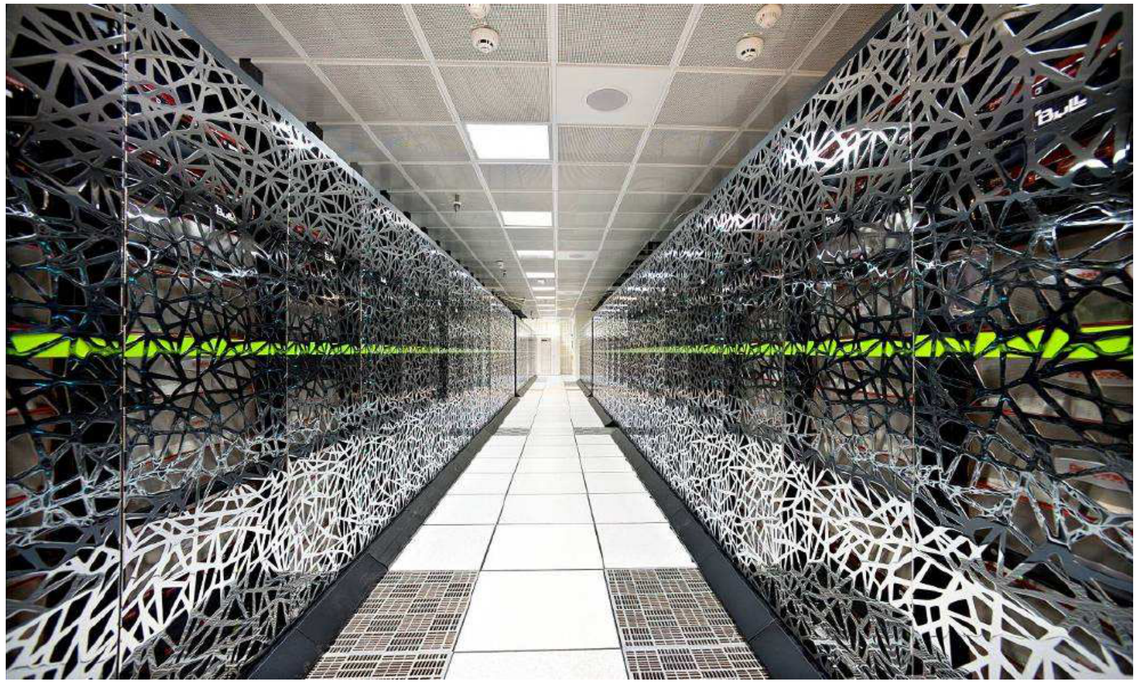




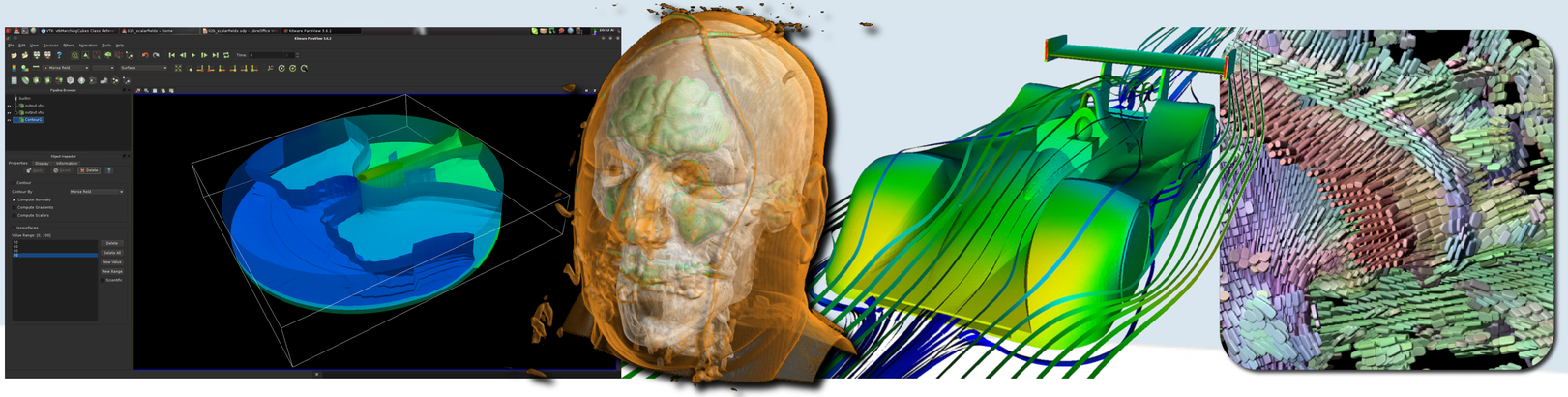
What for?



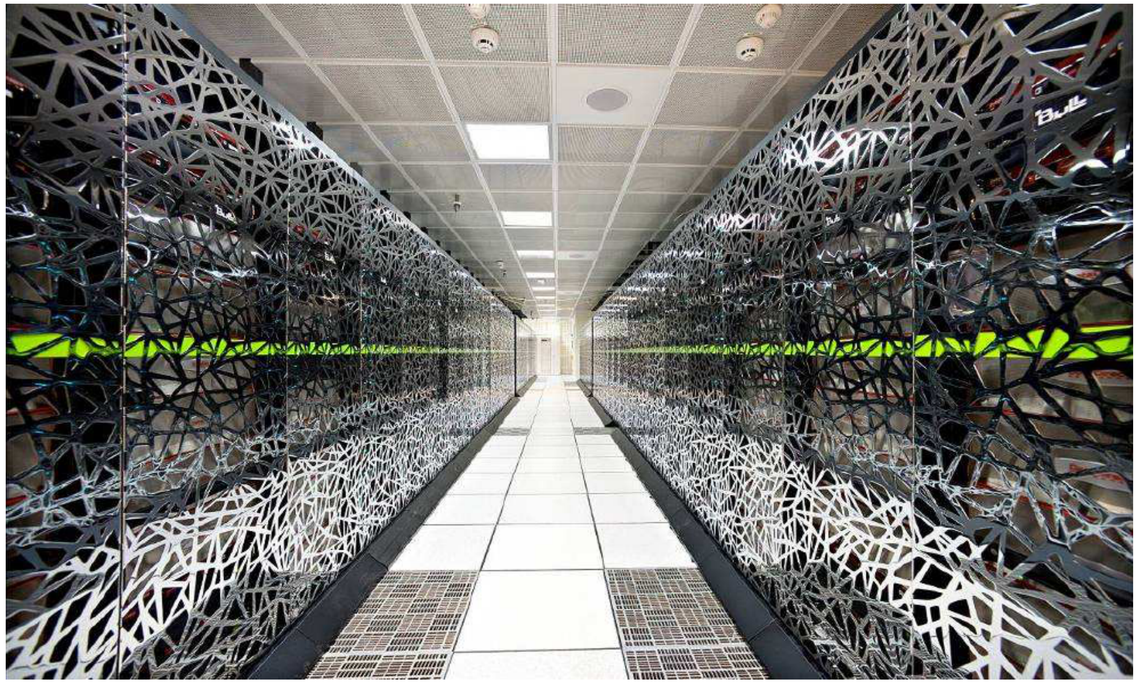
- Research and development activities



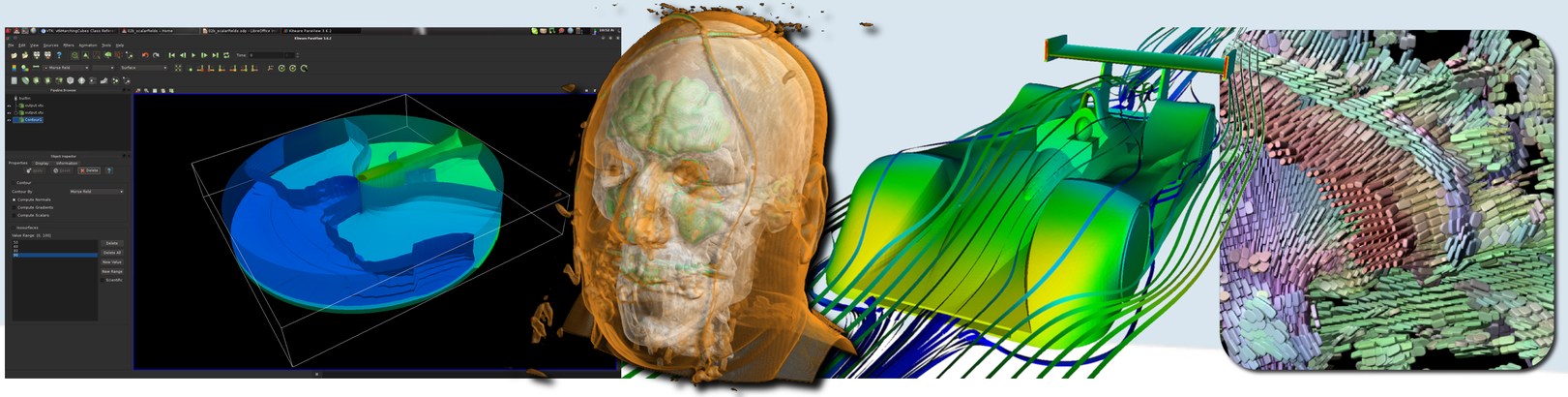
What for?



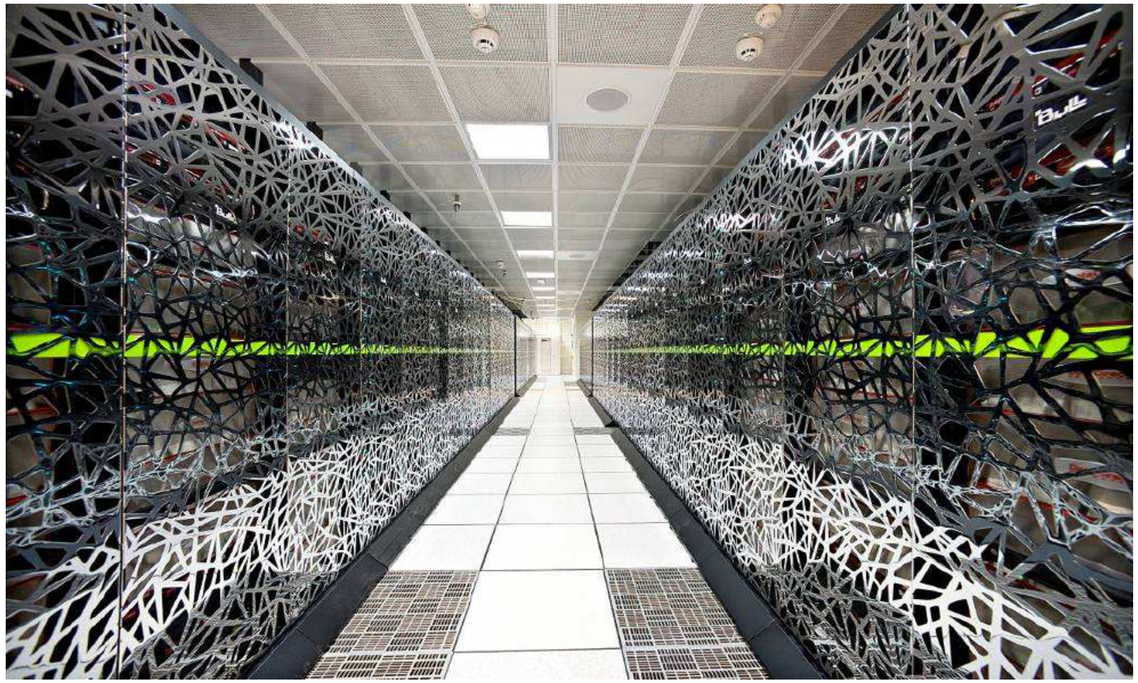
- Research and development activities
  - Academic
  - Industrial



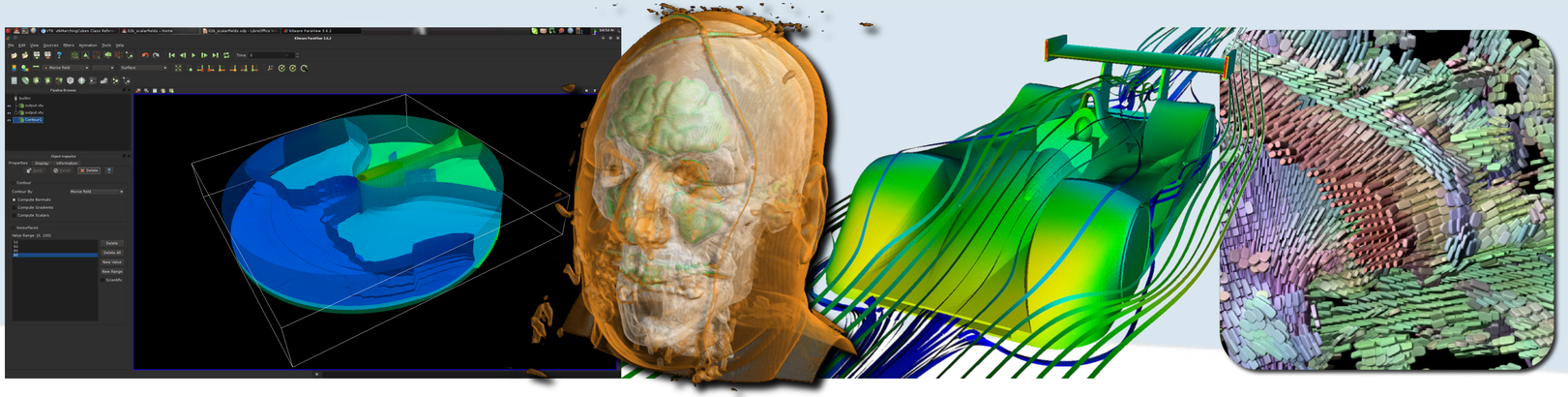
What for?



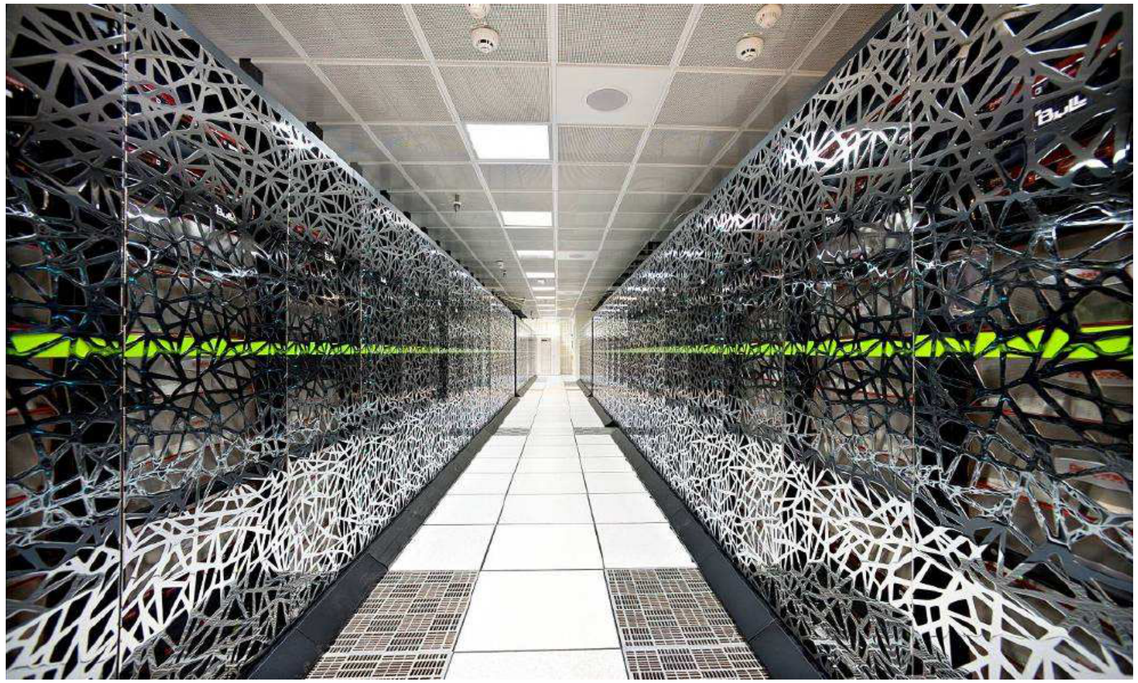
- Research and development activities
  - Academic
  - Industrial
    - CEA,
    - EDF,
    - Total,
    - Dassault Systèmes



What for?

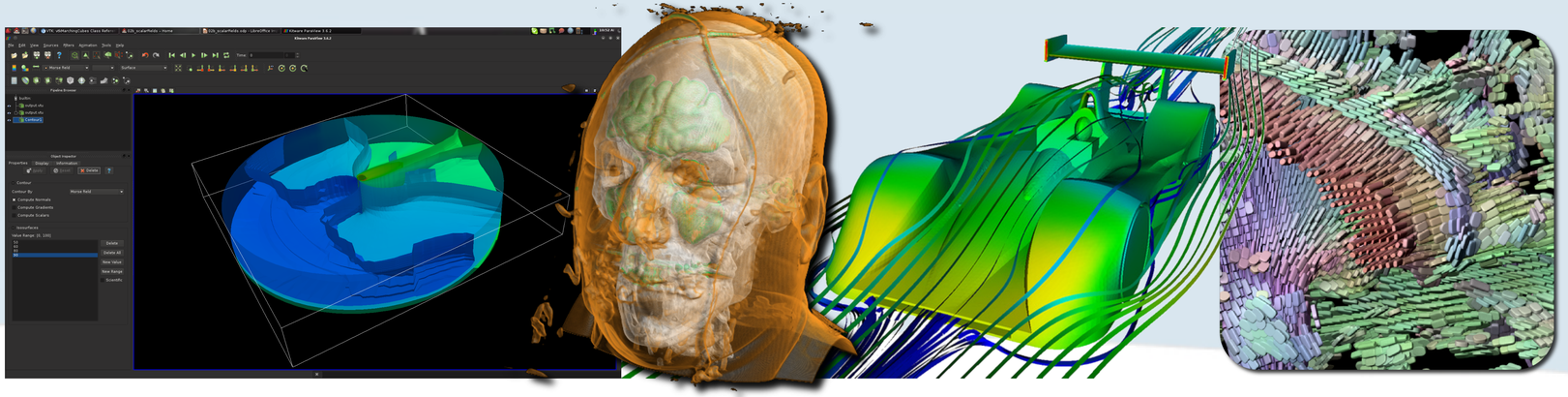


- Research and development activities
  - Academic
  - Industrial
    - CEA,
    - EDF,
    - Total,
    - Dassault Systèmes



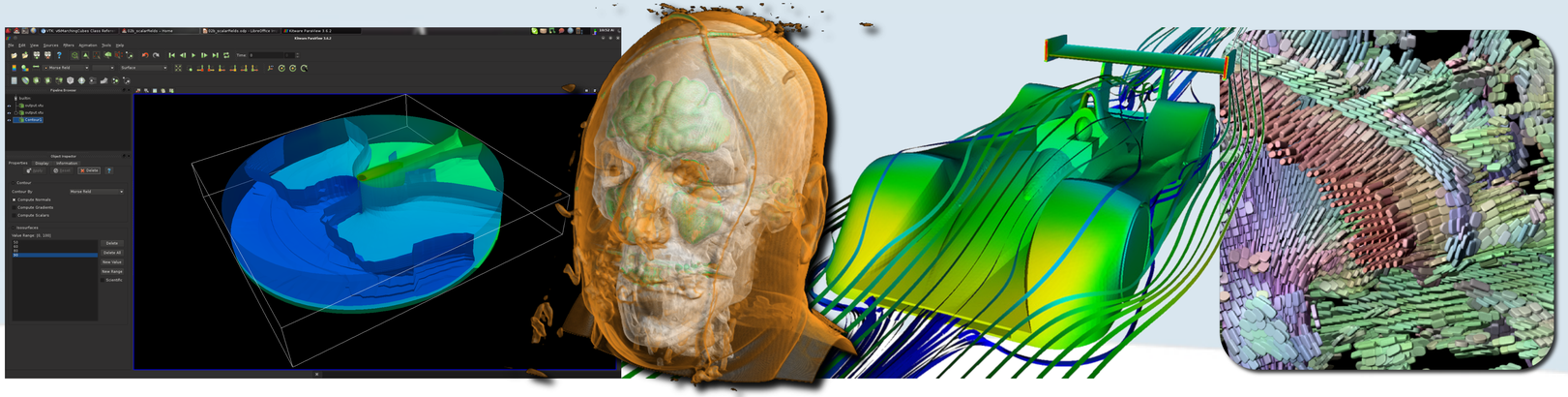
- 3D simulation, rendering and analysis software industry

What about?



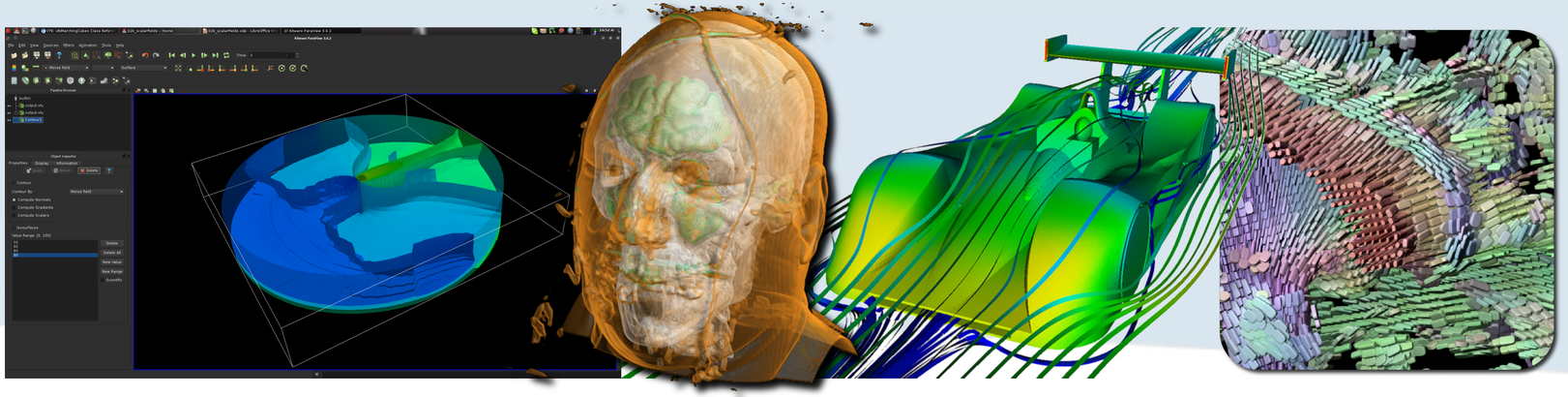
- Covered topics

What about?



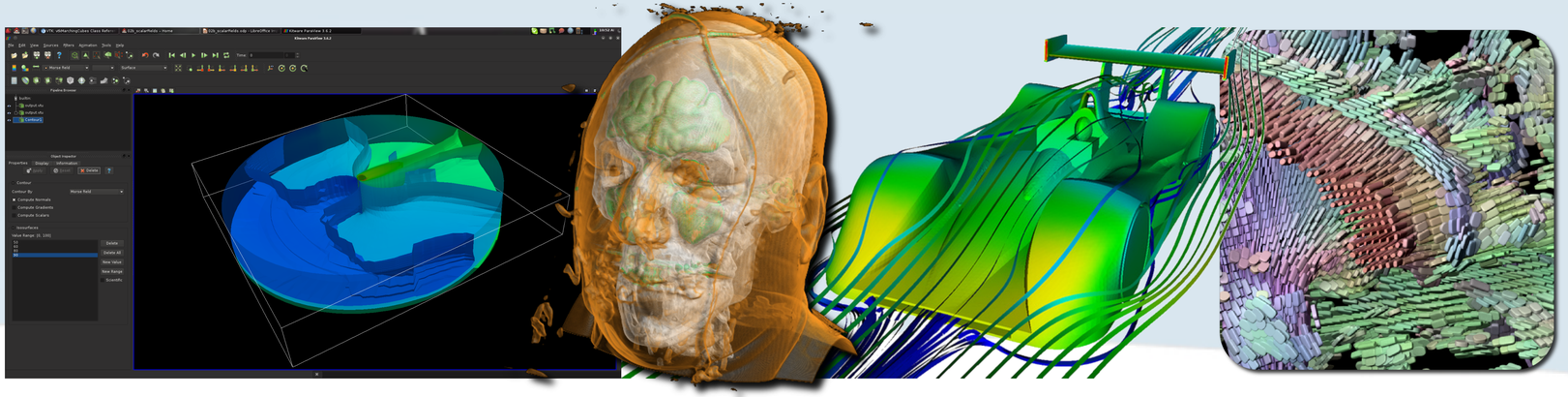
- Covered topics
  - 3D Rendering

What about?



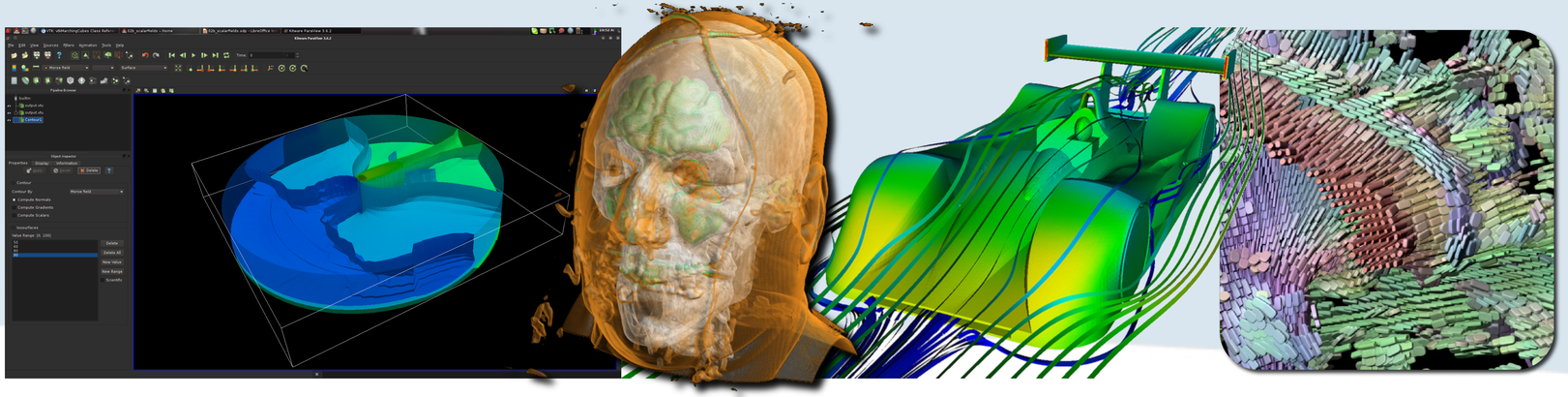
- Covered topics
  - 3D Rendering
  - Interactive systems

What about?



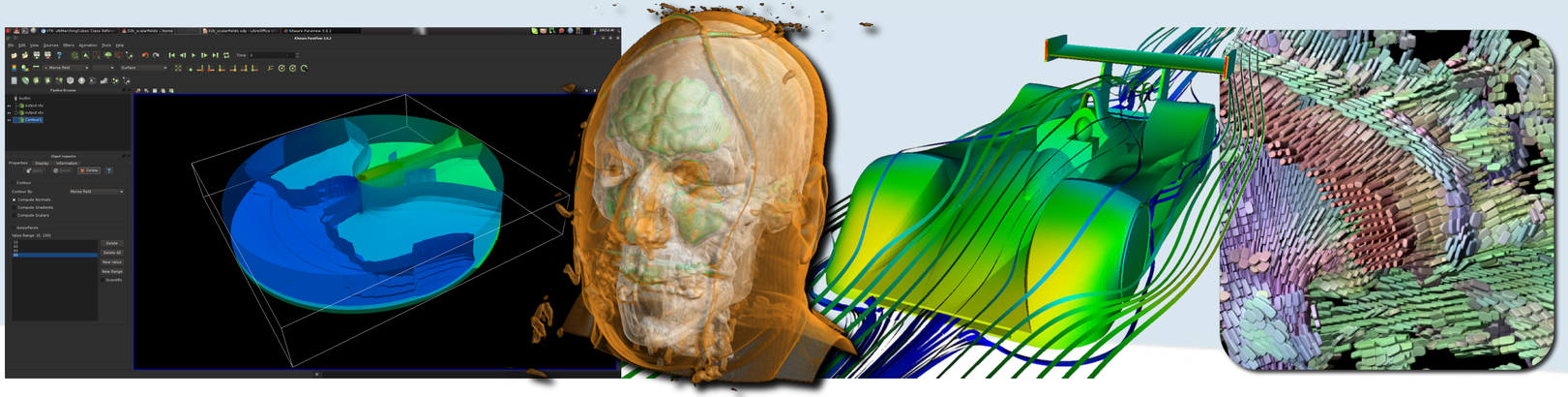
- Covered topics
  - 3D Rendering
  - Interactive systems
  - **Geometrical and topological analysis**

Who?



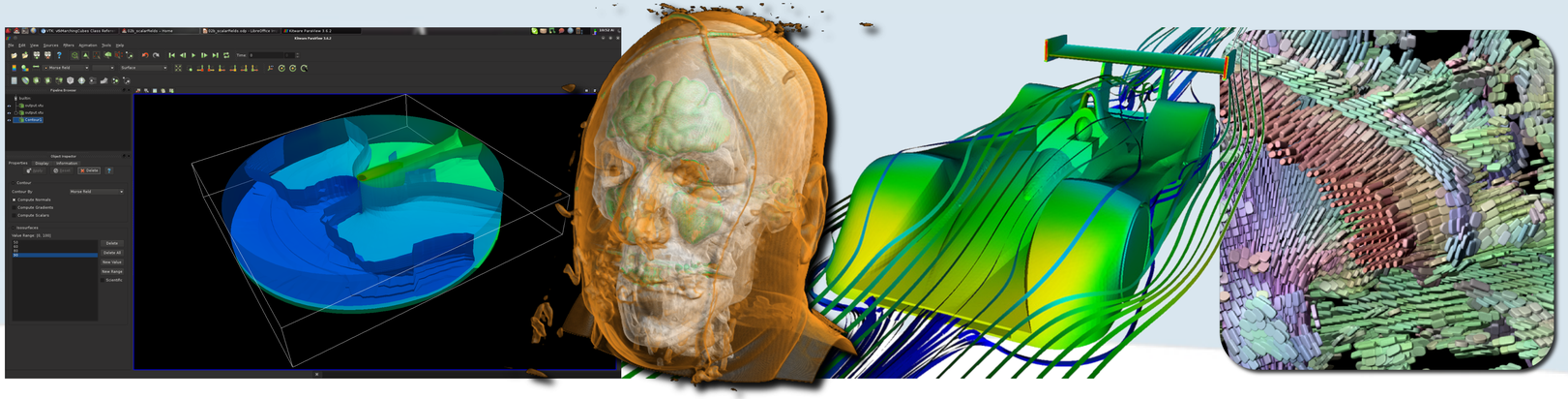
- Students targeting

Who?



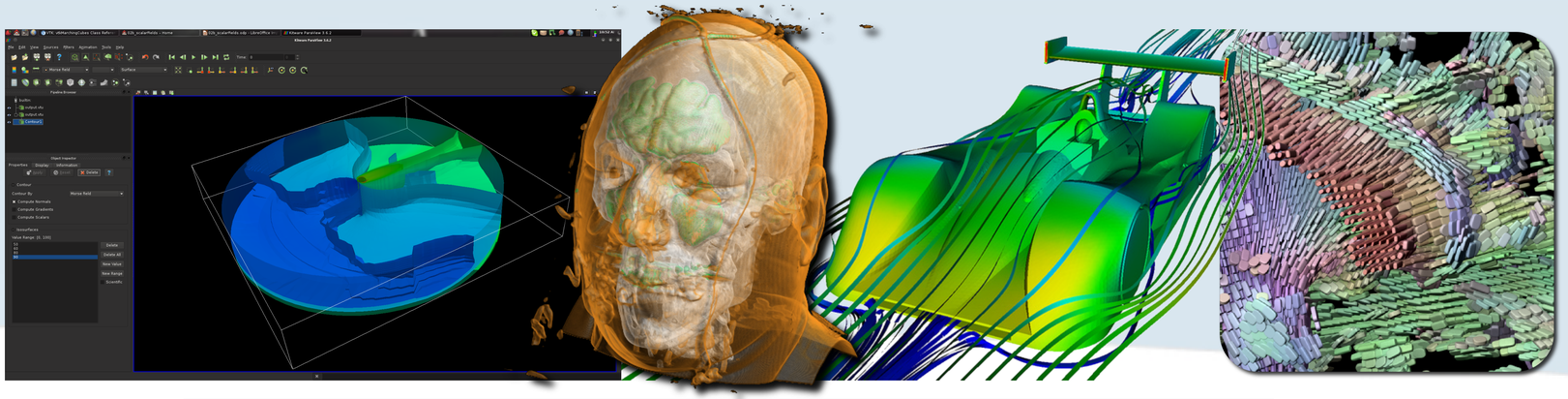
- Students targeting
  - R&D activities

Who?



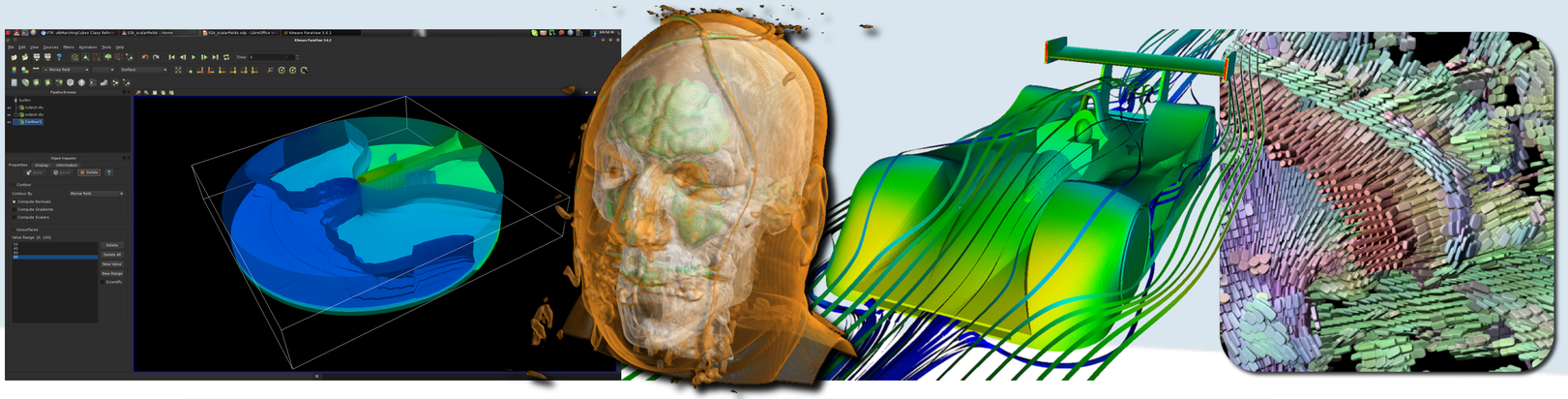
- Students targeting
  - R&D activities
- Students liking

Who?



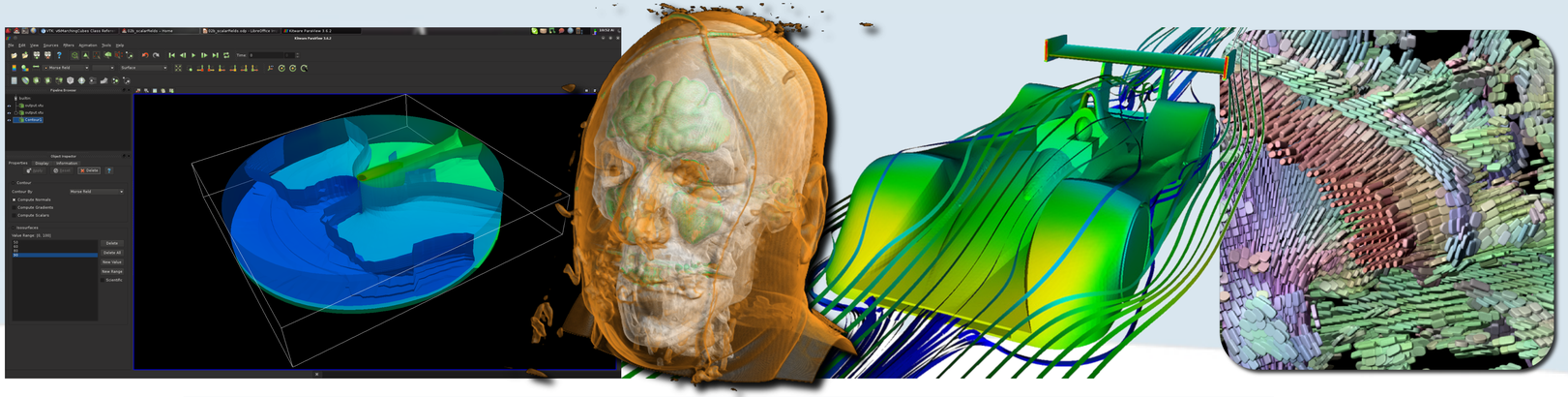
- Students targeting
  - R&D activities
- Students liking
  - Science

Who?



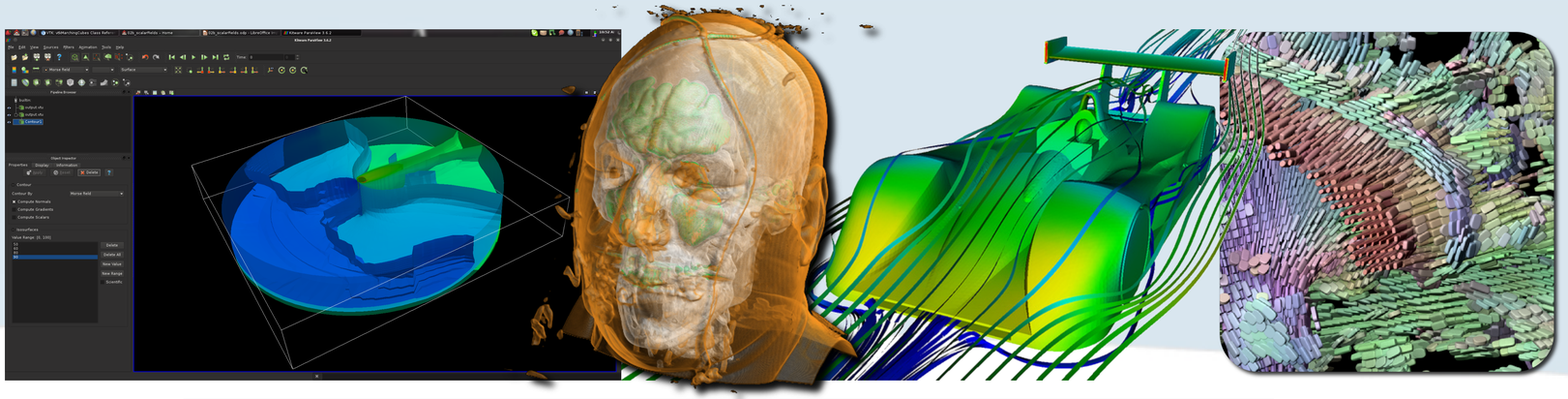
- Students targeting
  - R&D activities
- Students liking
  - Science
  - Geometry

Who?



- Students targeting
  - R&D activities
- Students liking
  - Science
  - Geometry
  - Coding

Who?



- Students targeting
  - R&D activities
- Students liking
  - Science
  - Geometry
  - Coding
  - Cool pictures

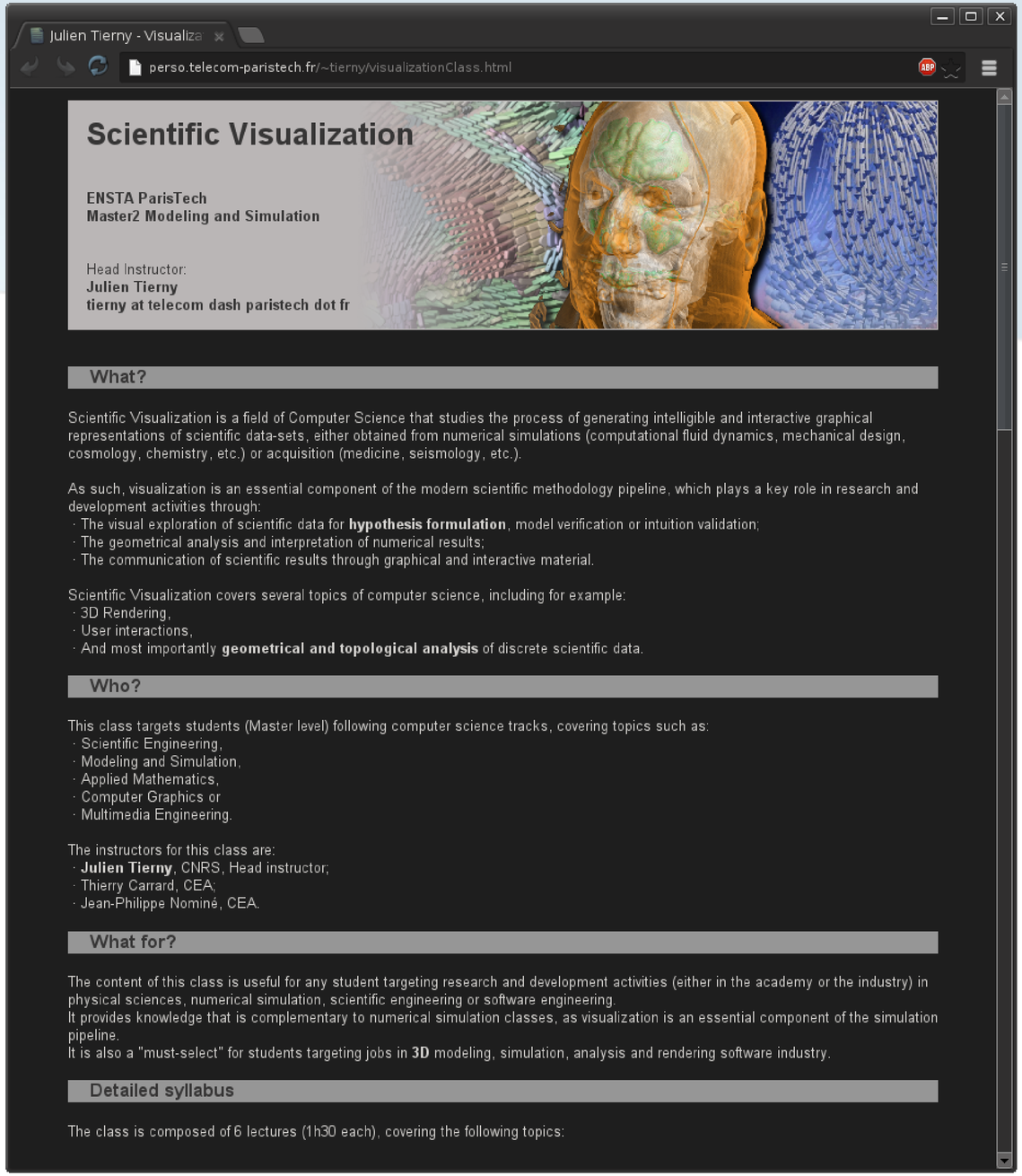
How?

How?

- First, follow this class

How?

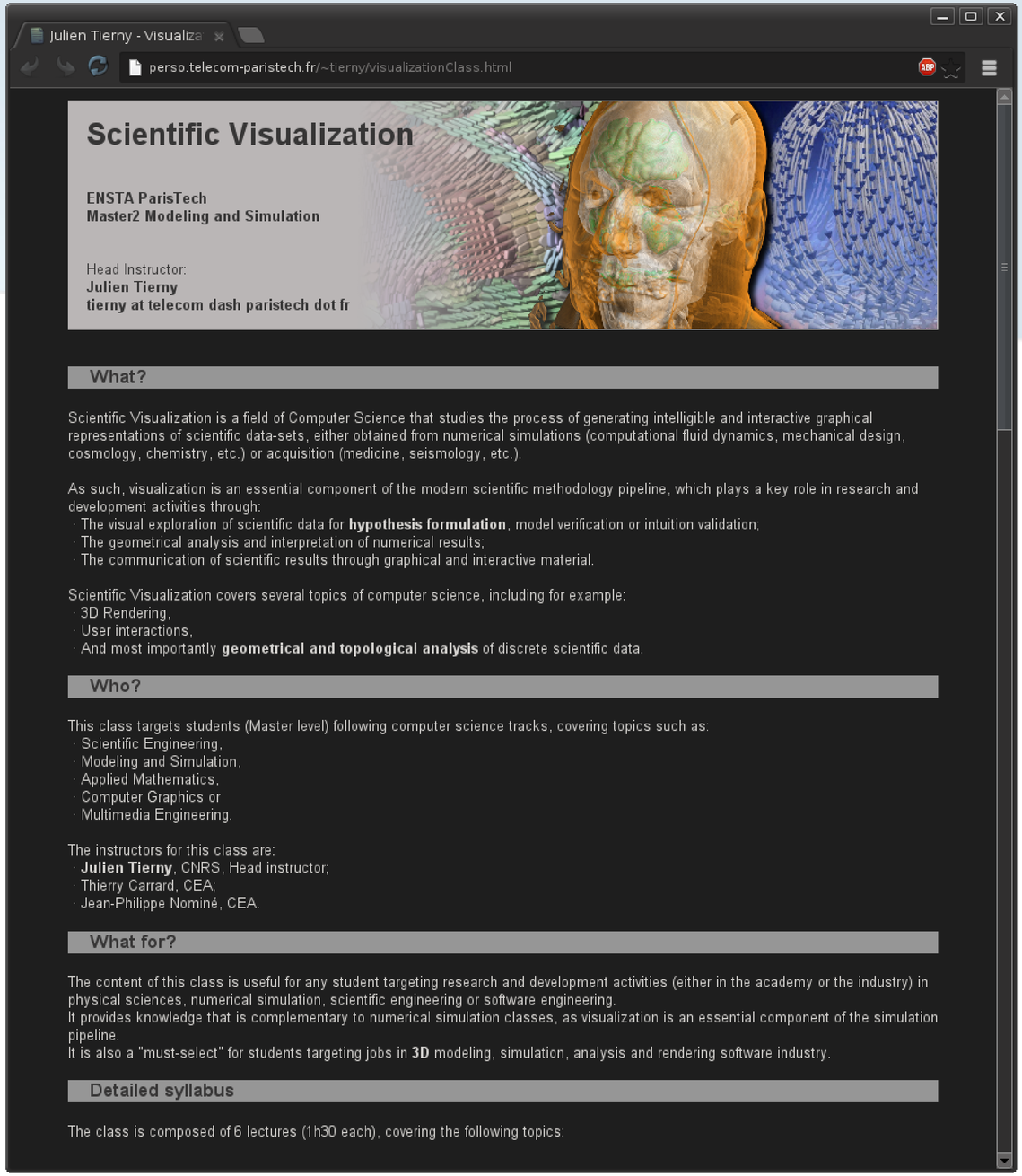
- First, follow this class



- Advanced class
  - http://www.telecom-paristech.fr/~tierny/visualizationClass.html

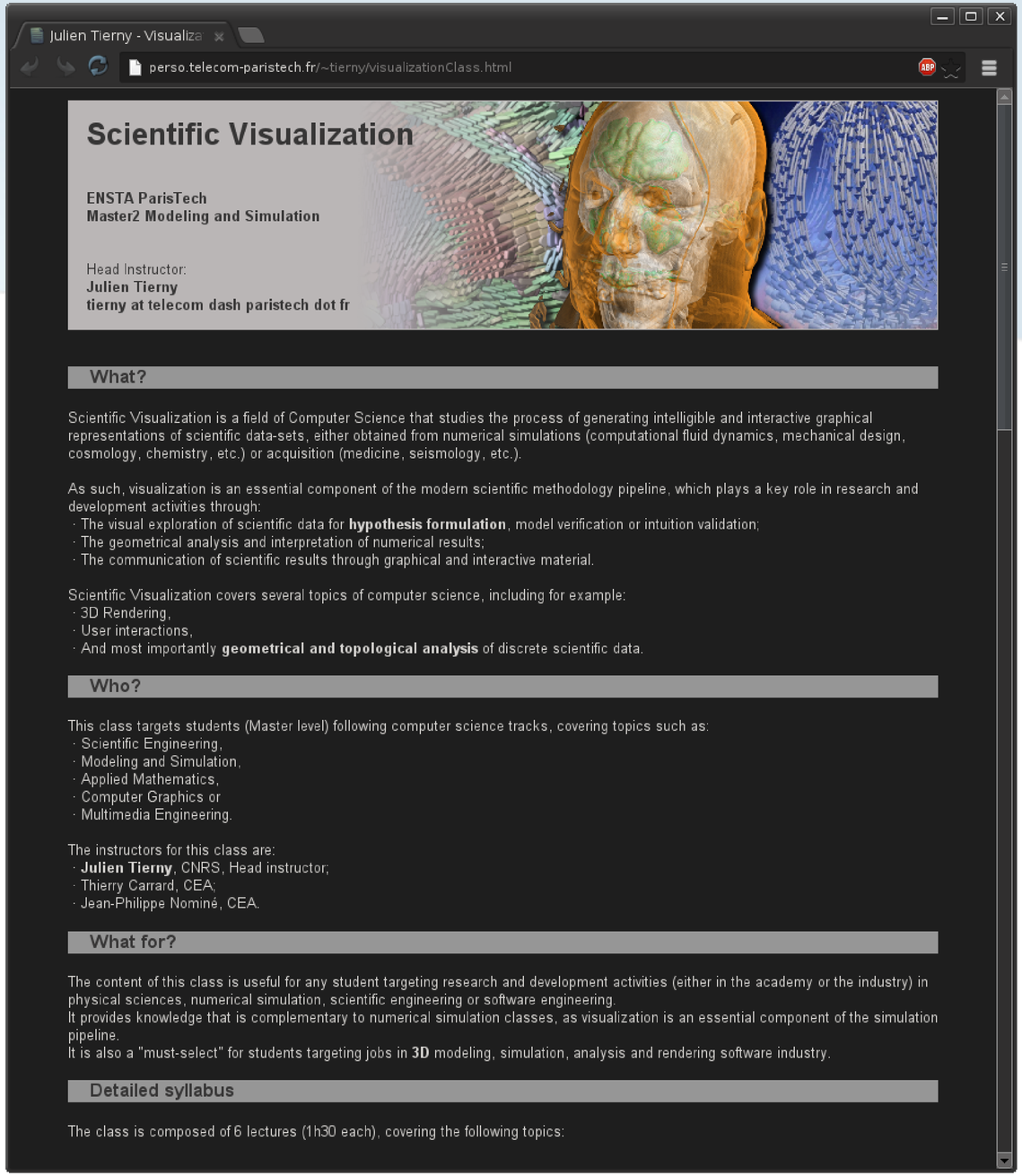
How?

- First, follow this class
- Modeling & Simulation Master
  - ENSTA ParisTech
- Advanced class
  - http://www.telecom-paristech.fr/~tierny/visualizationClass.html

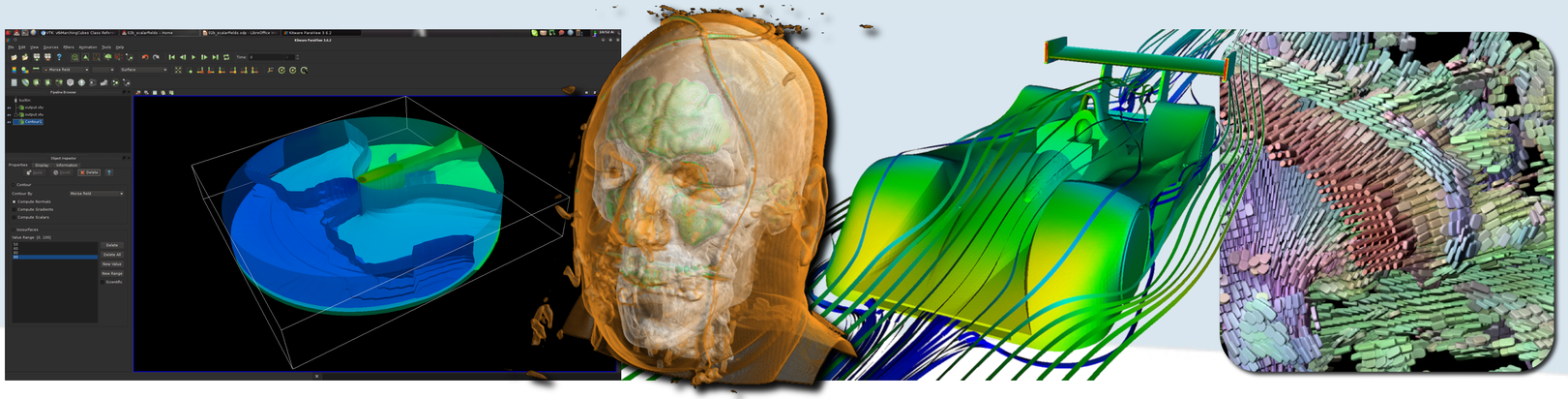


How?

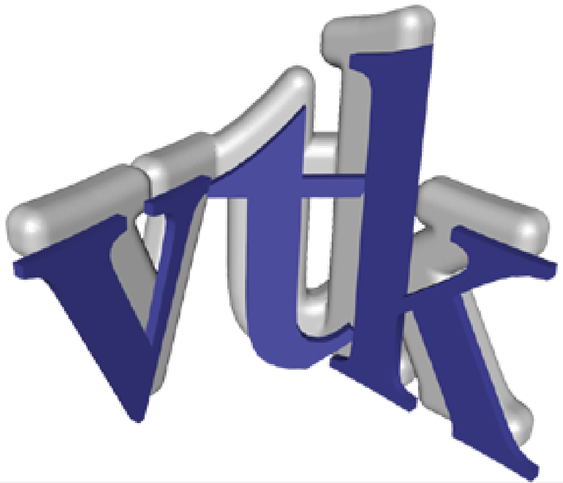
- First, follow this class
- Modeling & Simulation Master
  - ENSTA ParisTech
- Projects, internships, Ph.D. thesis
- Advanced class
  - http://www.telecom-paristech.fr/~tierny/visualizationClass.html

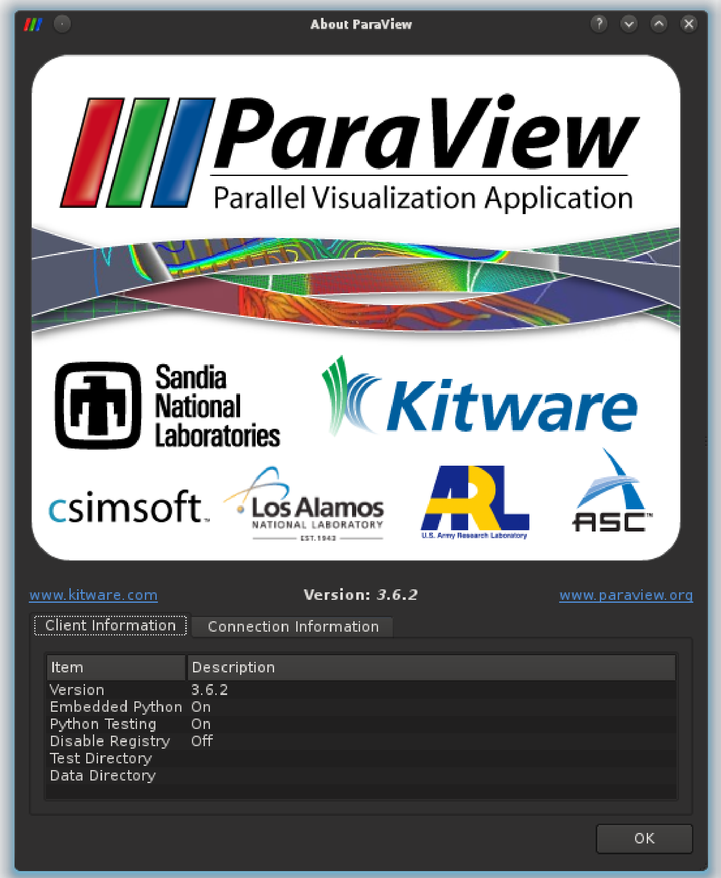


How?

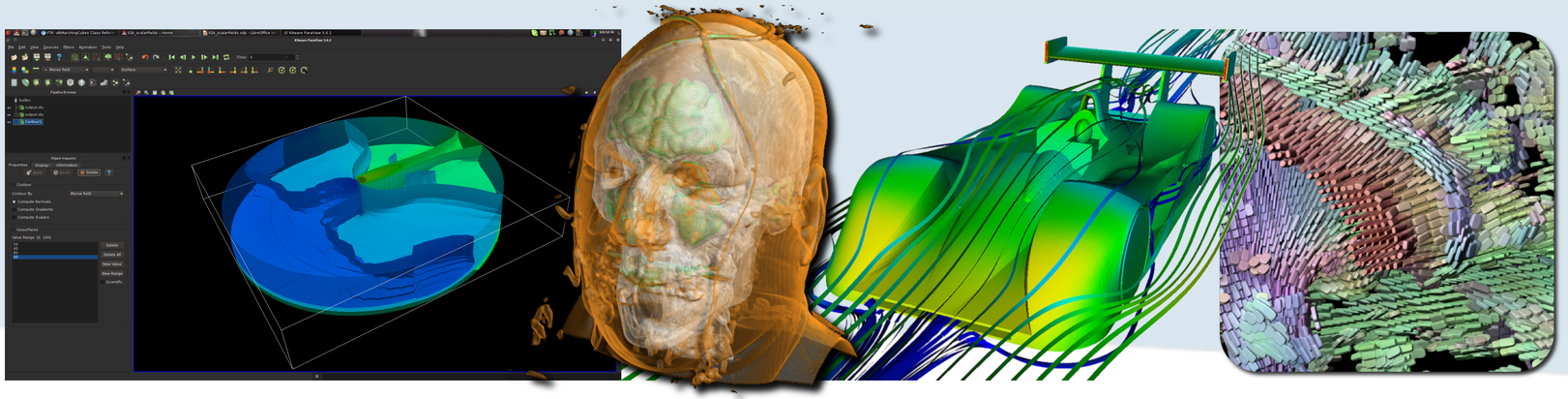


- Everything (well most of it) is already coded for you!

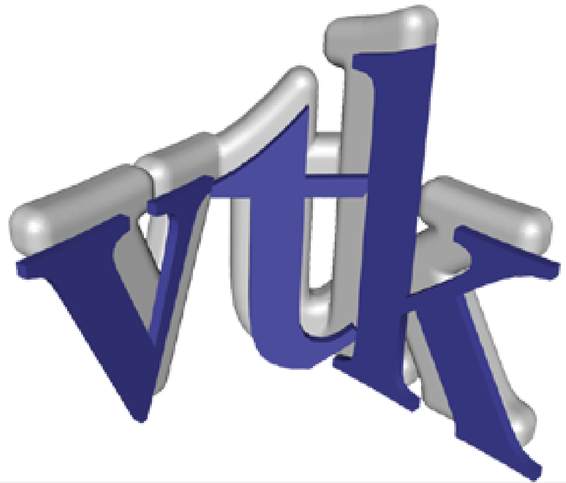


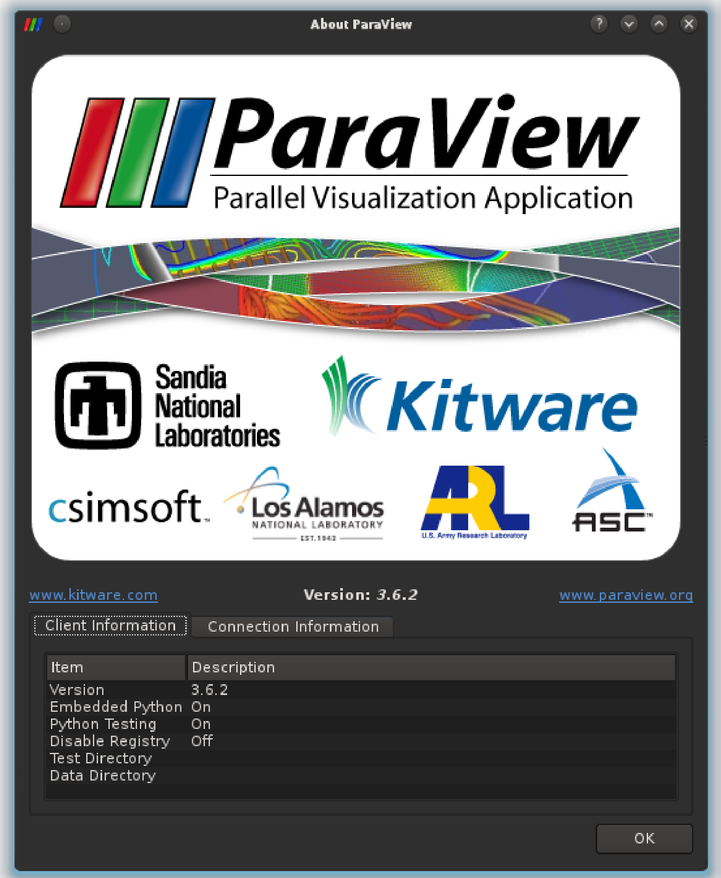


How?

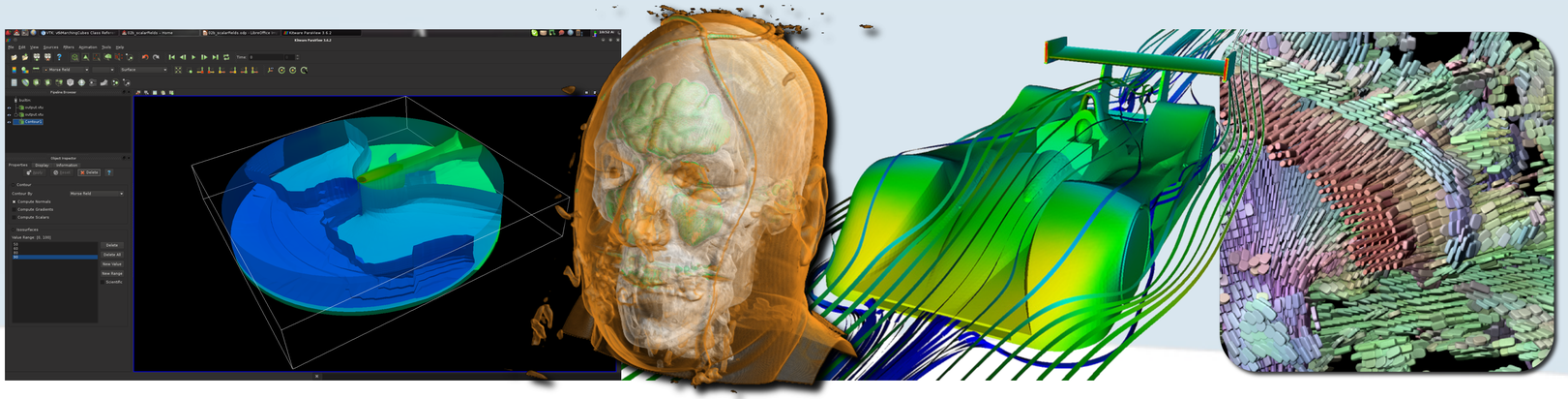


- Everything (well most of it) is already coded for you!
  - Visualization Tool Kit
    - Open Source C++ library
    - Started in 1993
    - Over a million lines of code
    - Tens of thousands of users

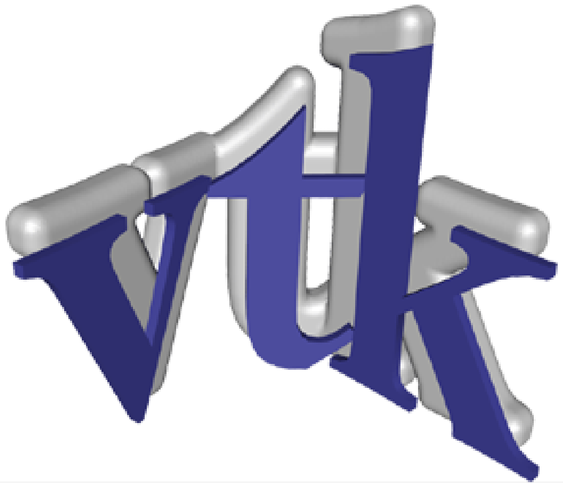


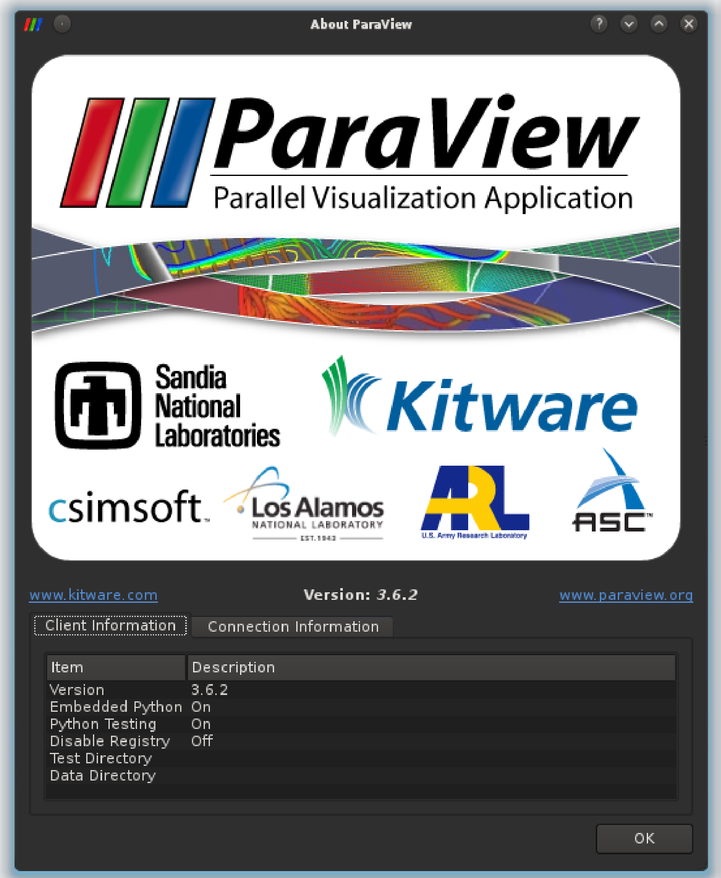


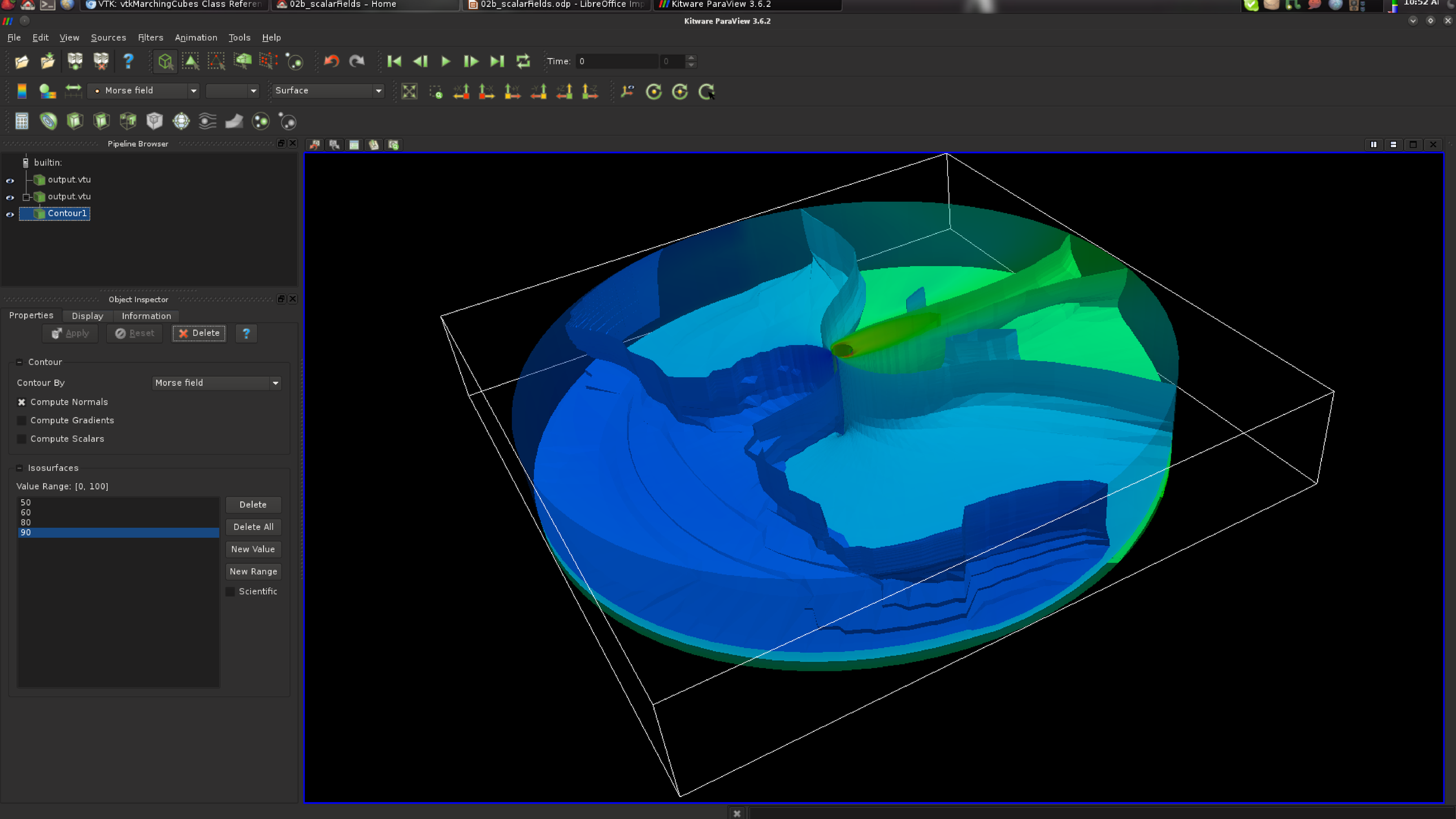
How?



- Everything (well most of it) is already coded for you!
  - Visualization Tool Kit
    - Open Source C++ library
    - Started in 1993
    - Over a million lines of code
    - Tens of thousands of users
  - Paraview
    - User interface front end



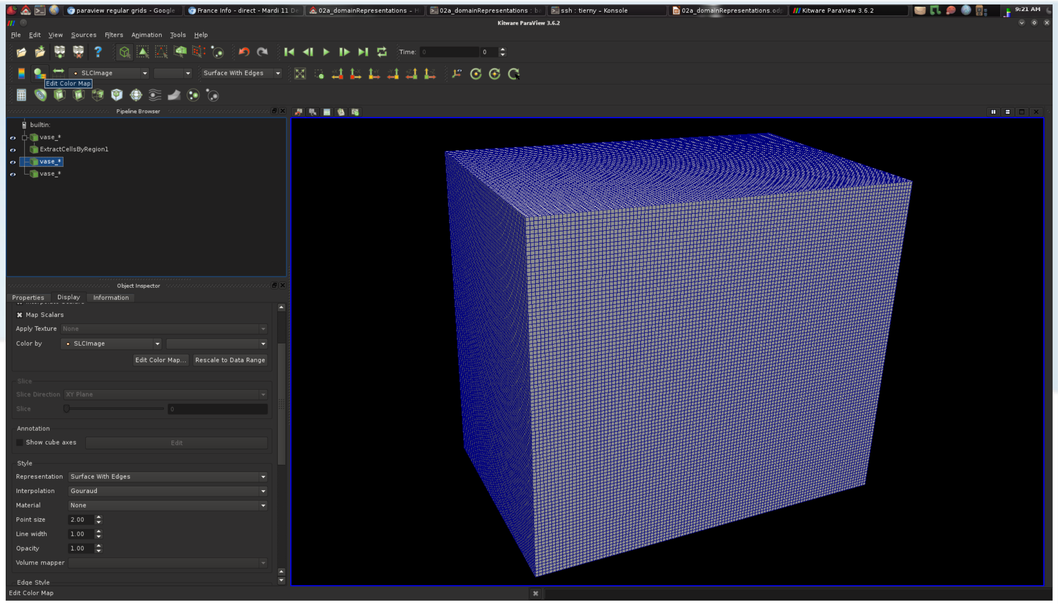



VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties 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

How does it work?

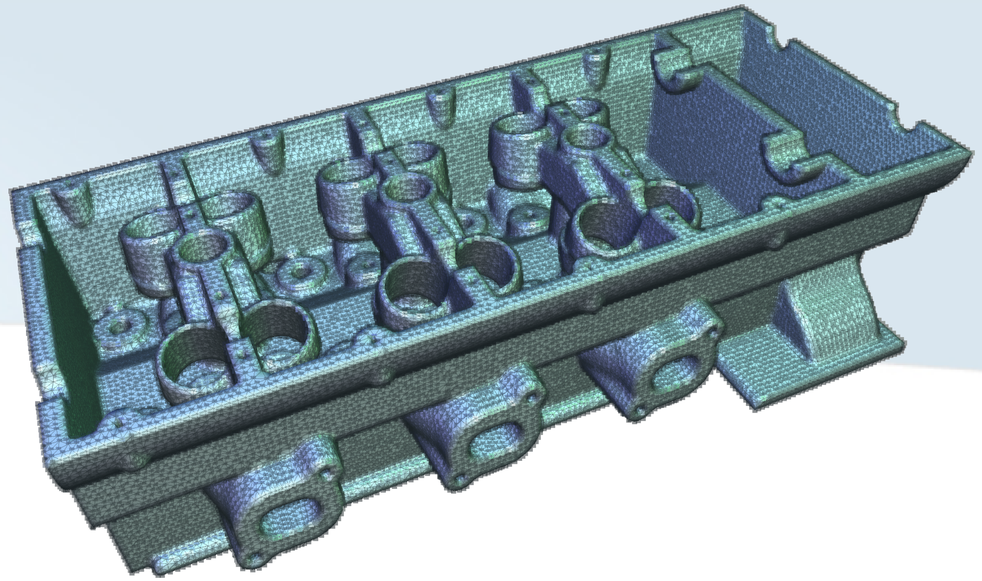
- It all starts with a numerical domain

How does it work?

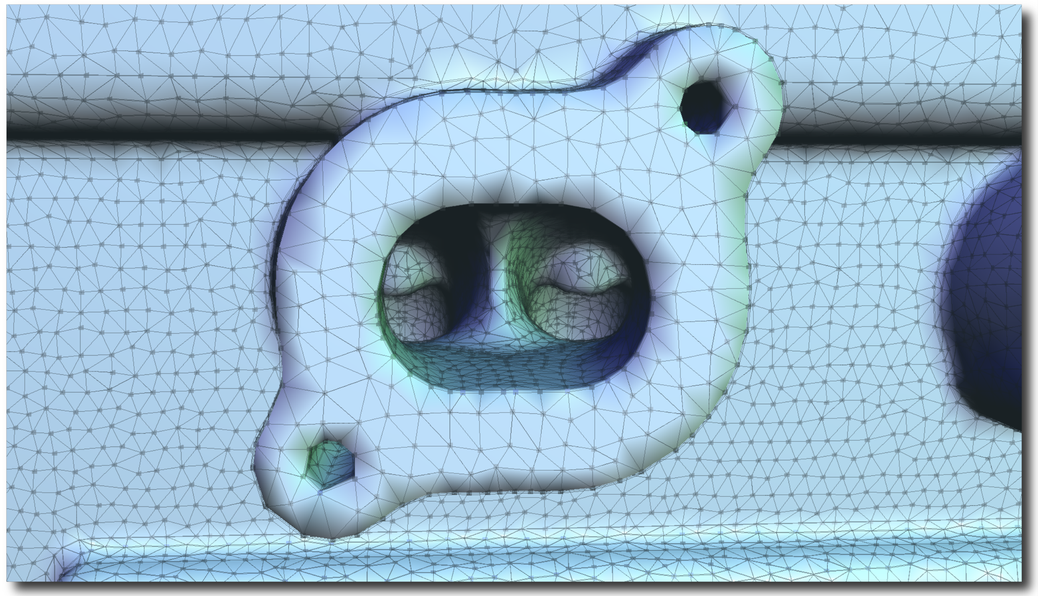


- It all starts with a numerical domain
  - Sub-set of euclidean spaces (1D, 2D, 3D, nD)
    - Regular Grids (pixels, voxels)

How does it work?

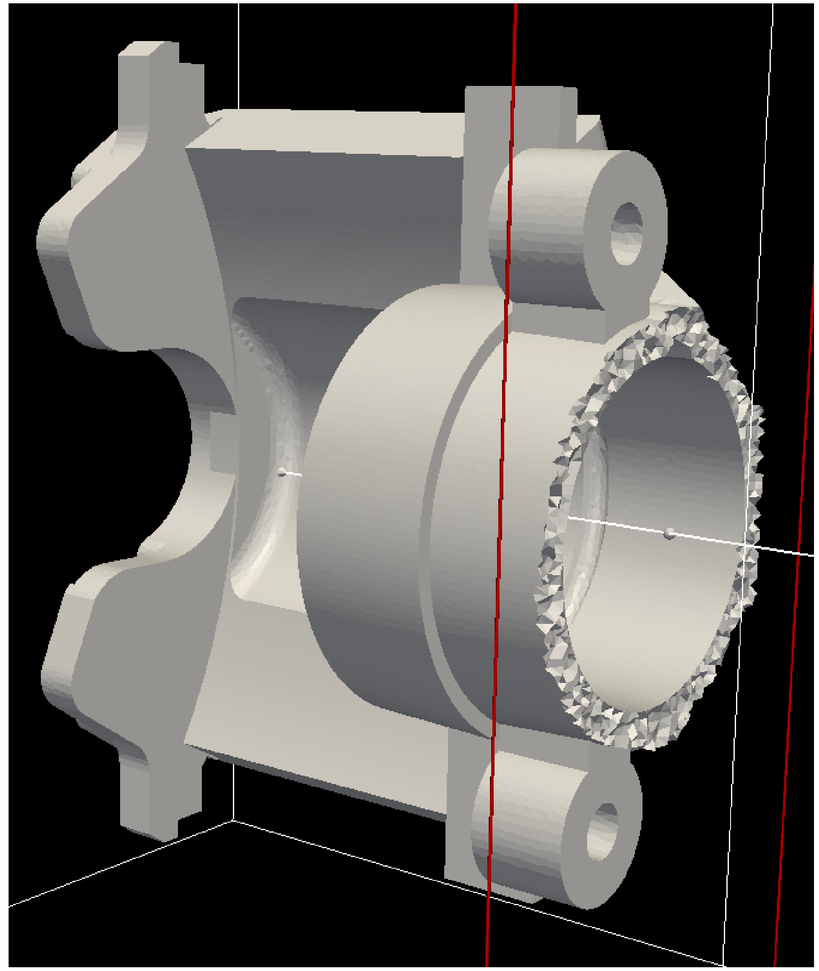


- It all starts with a numerical domain
  - Sub-set of euclidean spaces (1D, 2D, 3D, nD)
    - Regular Grids (pixels, voxels)
  - Non-euclidean spaces (1D, 2D, 3D, nD)
    - Notion of piecewise linear manifold
    - Triangle surface, Tetrahedral mesh



How does it work?

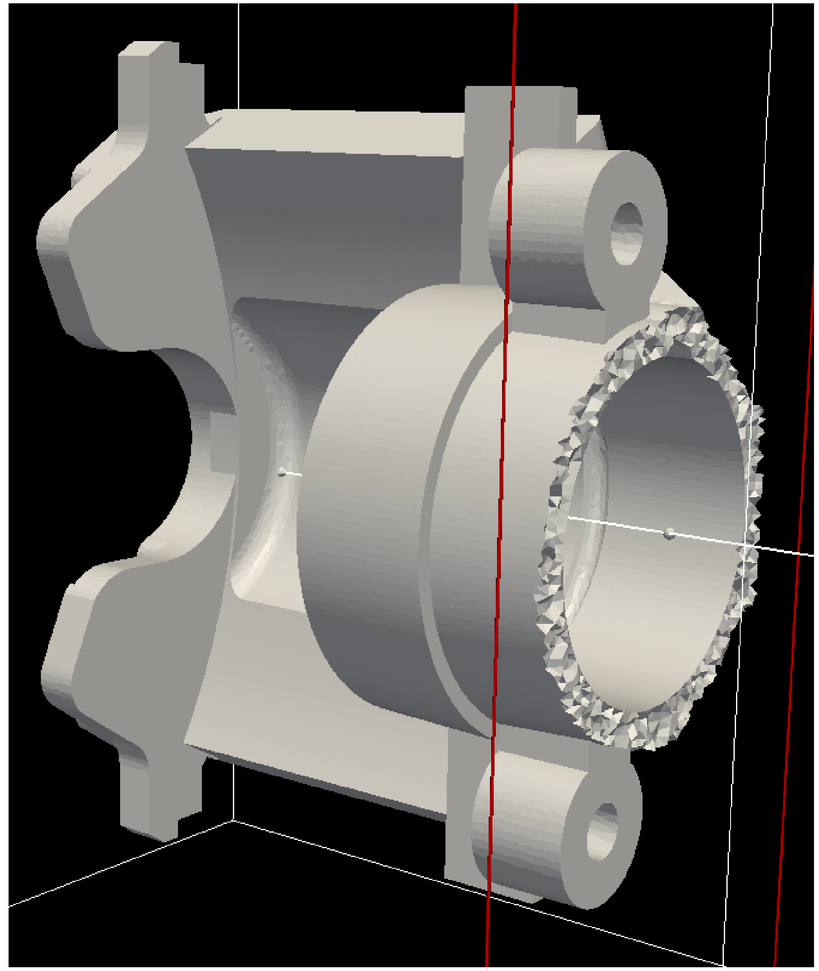
- It all starts with a numerical domain
  - Sub-set of euclidean spaces (1D, 2D, 3D, nD)
    - Regular Grids (pixels, voxels)
  - Non-euclidean spaces (1D, 2D, 3D, nD)
    - Notion of piecewise linear manifold
    - Triangle surface, Tetrahedral mesh



How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- It all starts with a numerical domain
  - Sub-set of euclidean spaces (1D, 2D, 3D, nD)
    - Regular Grids (pixels, voxels)
  - Non-euclidean spaces (1D, 2D, 3D, nD)
    - Notion of piecewise linear manifold
    - Triangle surface, Tetrahedral mesh
- Simulation or acquisition
  - Compute **something** on the domain



How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

How does it work?

$f : \mathcal{D} \rightarrow \mathcal{A}$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization

How does it work?

$f : \mathcal{D} \rightarrow \mathcal{A}$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization

How does it work?

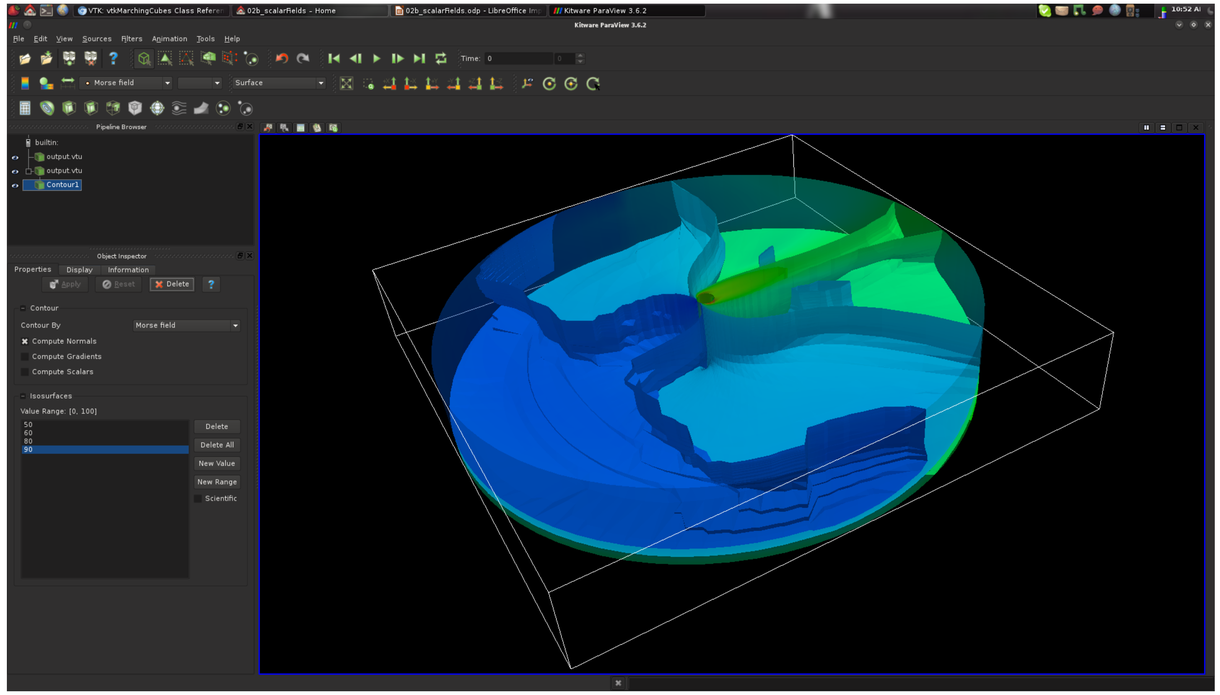
$f : \mathcal{D} \rightarrow \mathcal{A}$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization

How does it work?

$$f : \mathcal{D} \to \mathcal{A}$$

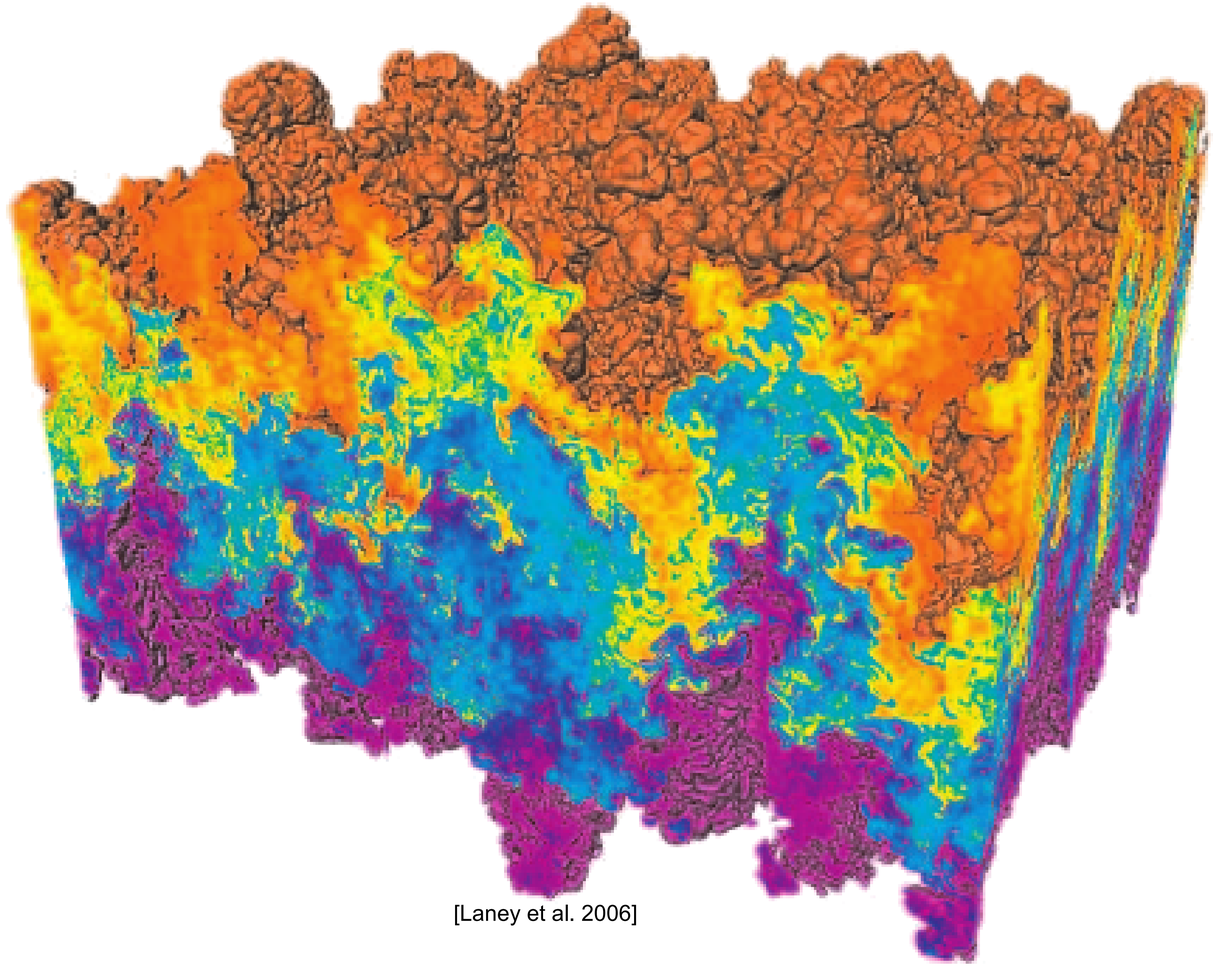
- $f : \mathcal{D} \to \mathbb{R}$
  - **Scalar field visualization**
- $f : \mathcal{D} \to \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \to \mathbb{M}_{d \times d}$
  - Tensor field visualization



How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - **Scalar field visualization**
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization

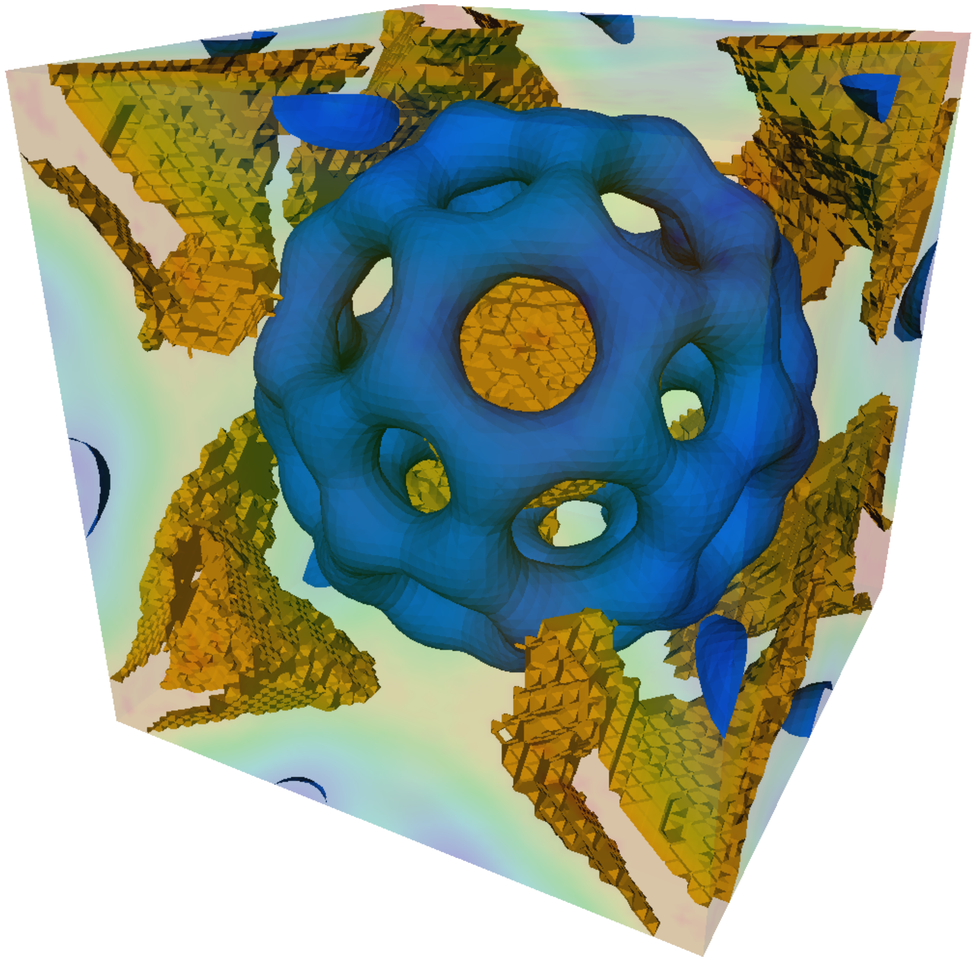


[Laney et al. 2006]

How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

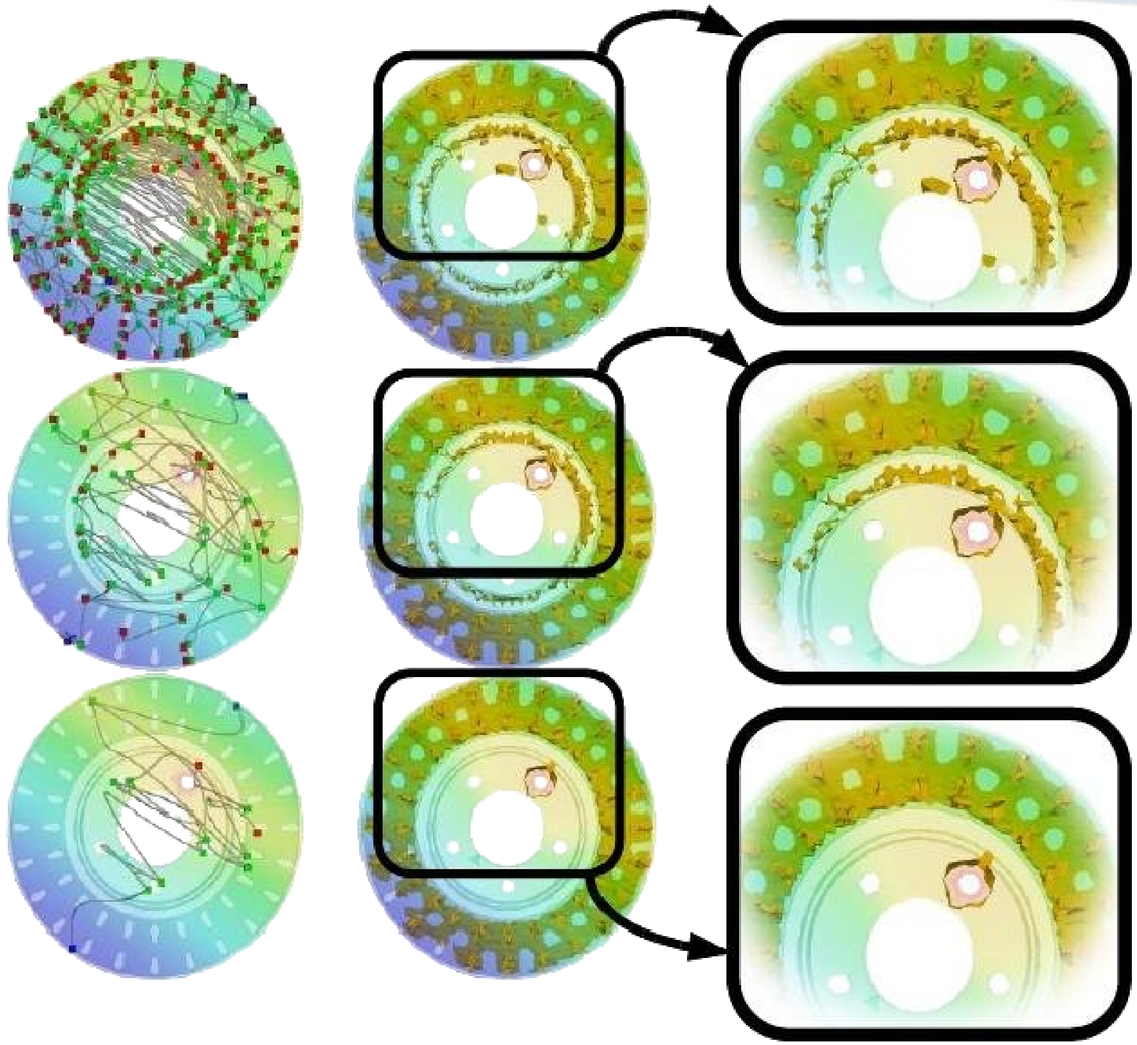
- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - **Scalar field visualization**
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization



How does it work?

$f : \mathcal{D} \rightarrow \mathcal{A}$

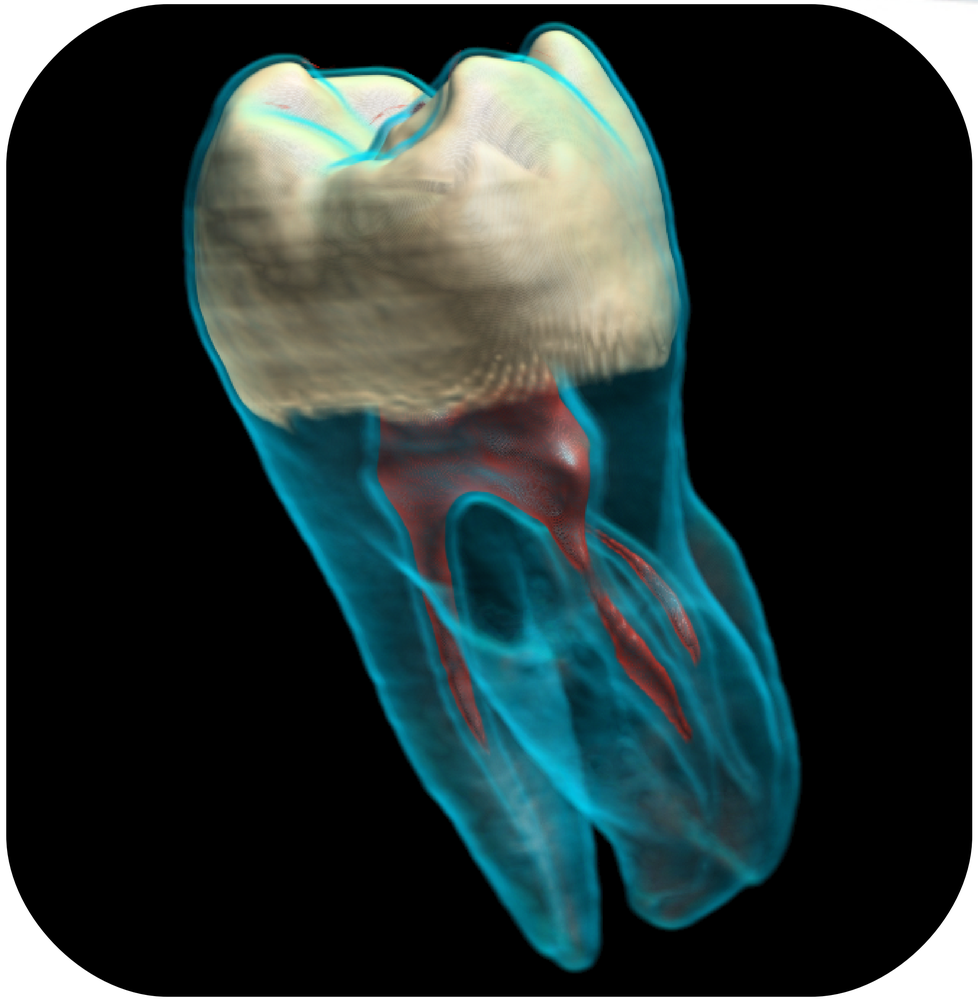
- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - **Scalar field visualization**
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization



How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - **Scalar field visualization**
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization



How does it work?

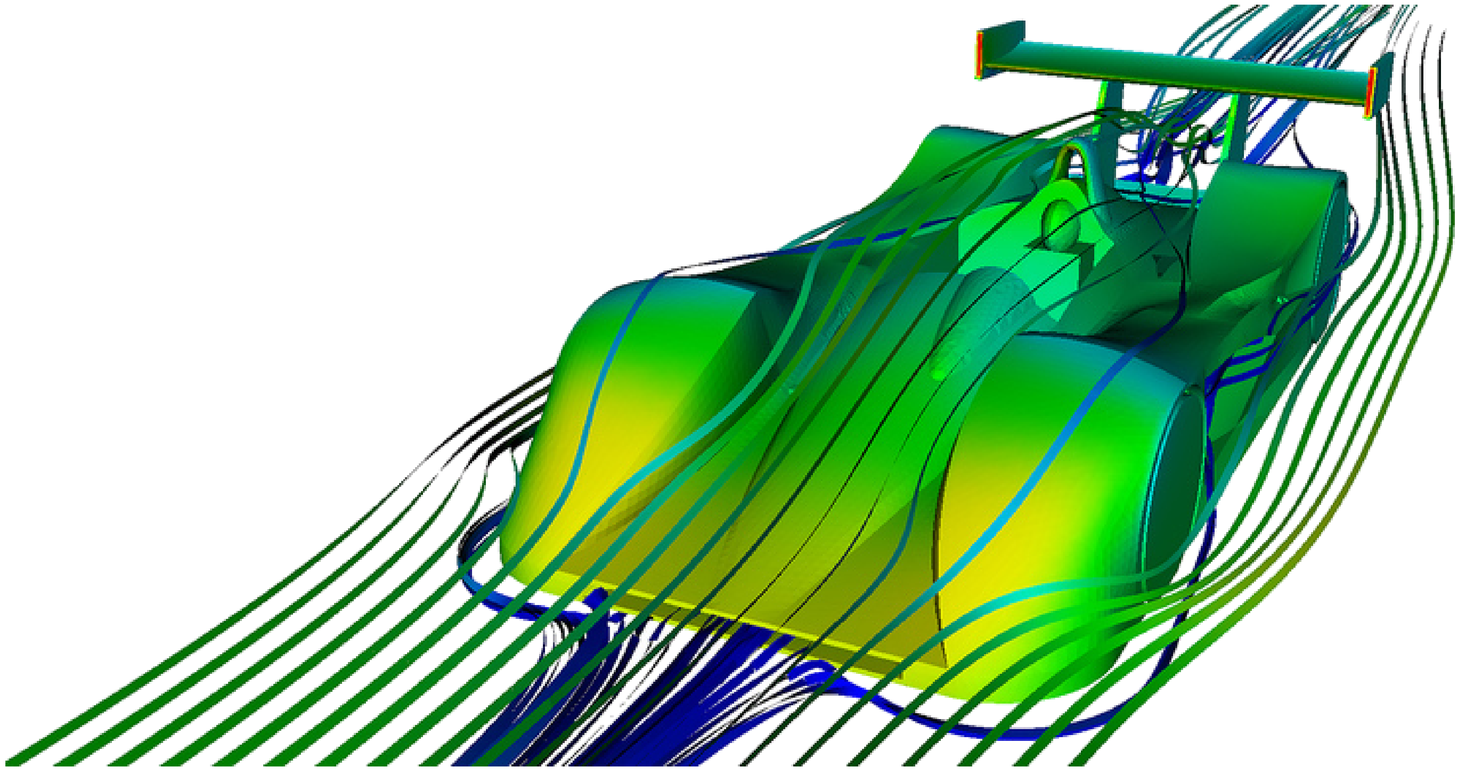
$f : \mathcal{D} \rightarrow \mathcal{A}$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - **Vector field visualization**
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization

How does it work?

$f : \mathcal{D} \rightarrow \mathcal{A}$

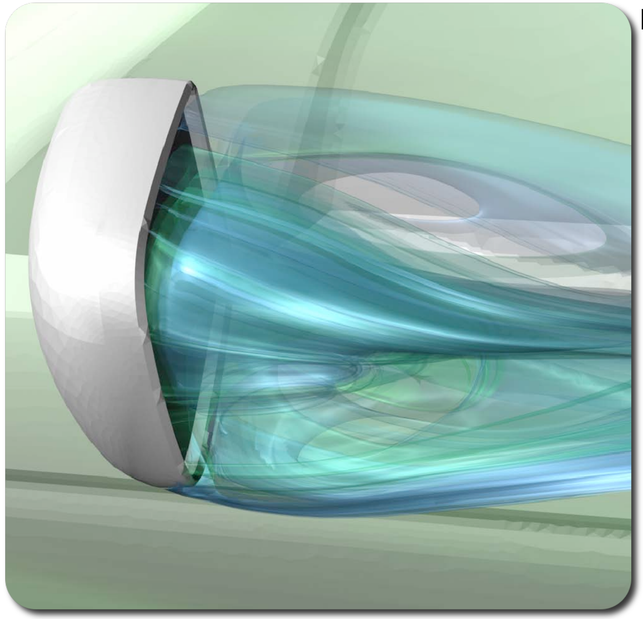
- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - **Vector field visualization**
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization



How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - **Vector field visualization**
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization

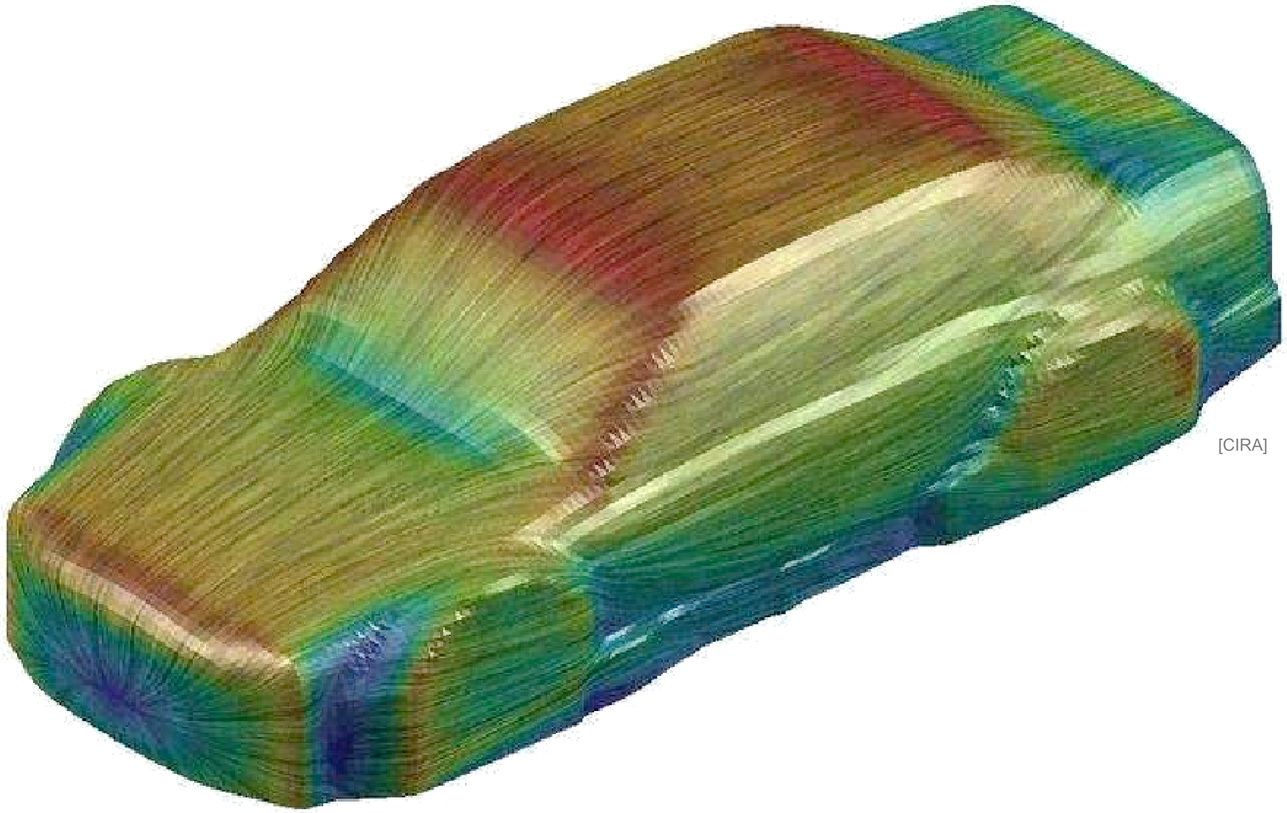


[GarthVIS08]

How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - **Vector field visualization**
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization

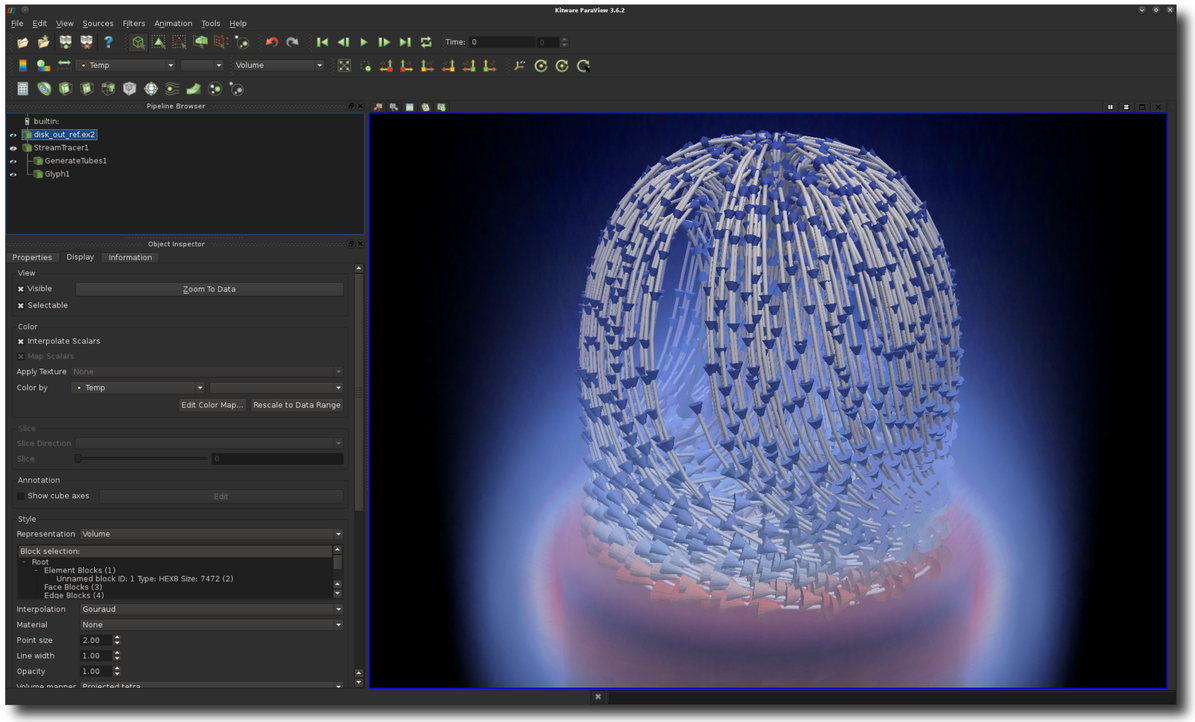


[CIRA]

How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

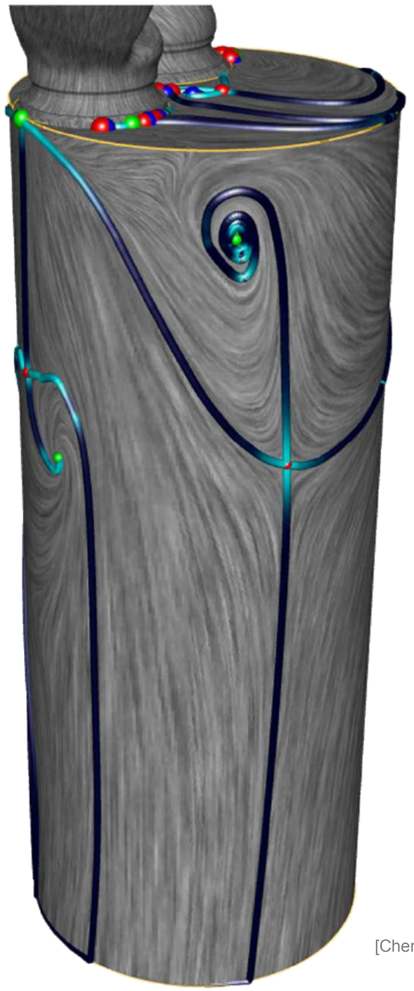
- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - **Vector field visualization**
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization



How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - **Vector field visualization**
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - Tensor field visualization

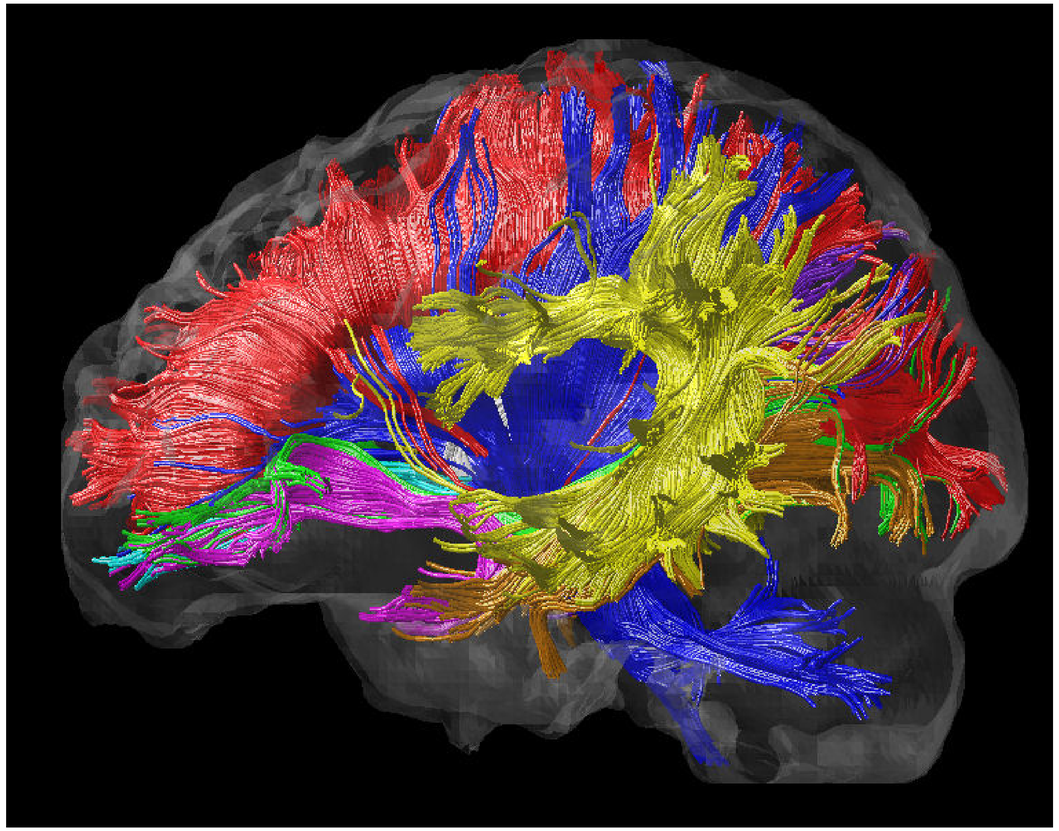


[Chen]

How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - **Tensor field visualization**

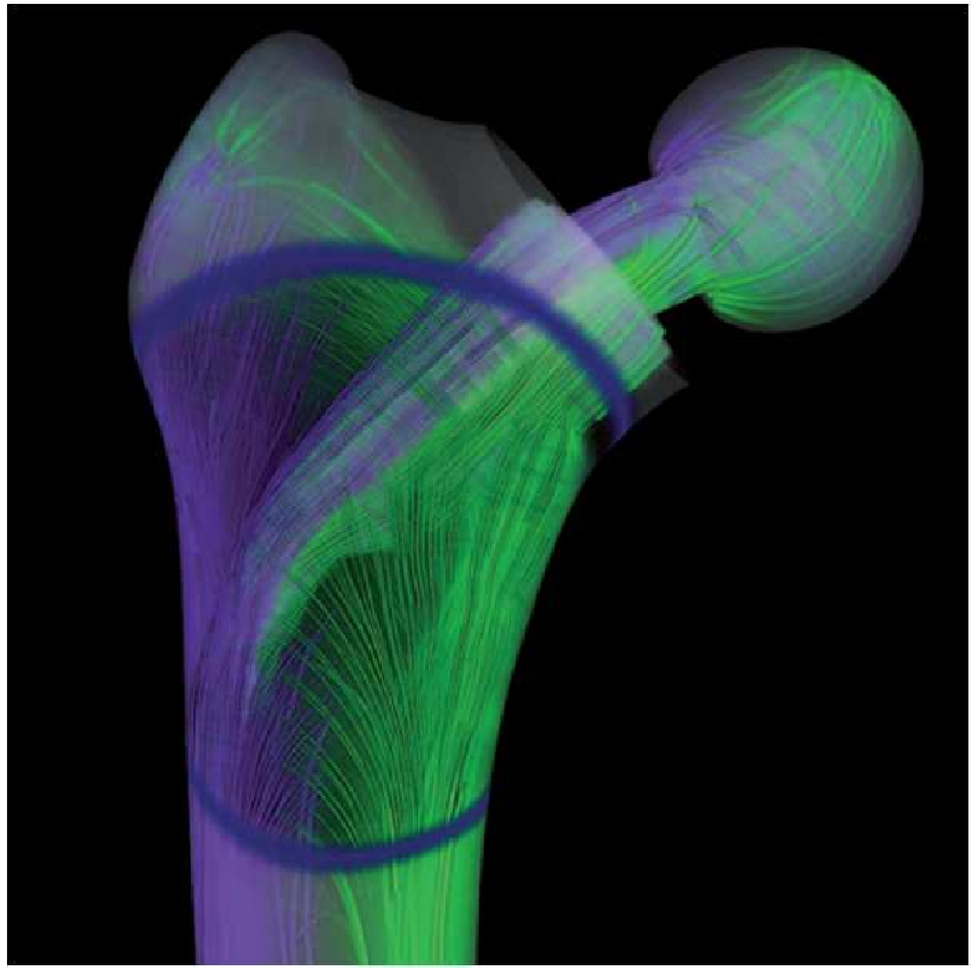


[http://www.neuroimaging.tau.ac.il]

How does it work?

$$f : \mathcal{D} \rightarrow \mathcal{A}$$

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D})$
  - Vector field visualization
- $f : \mathcal{D} \rightarrow \mathbb{M}_{d \times d}$
  - **Tensor field visualization**



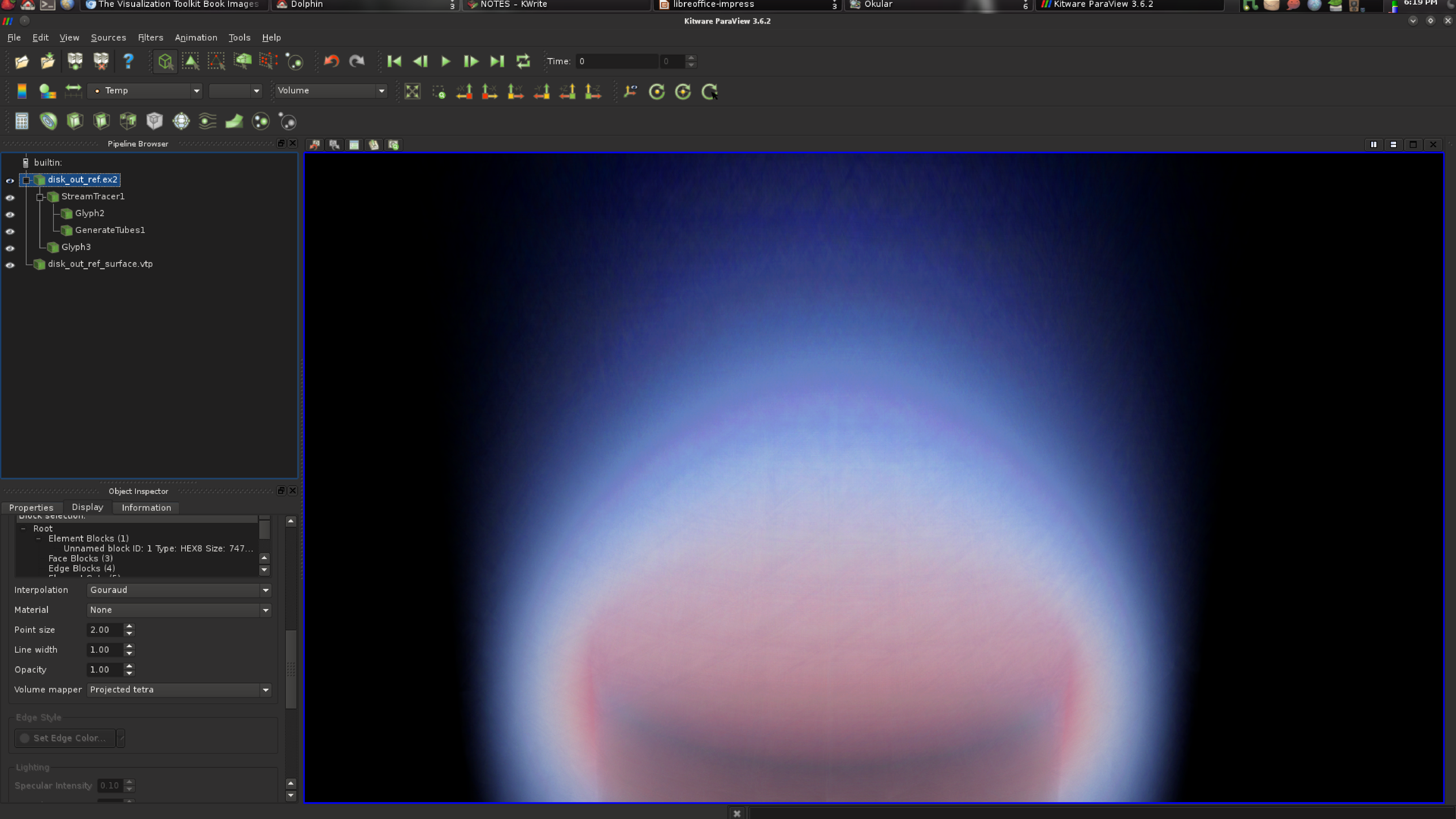
[Dick et al 09]

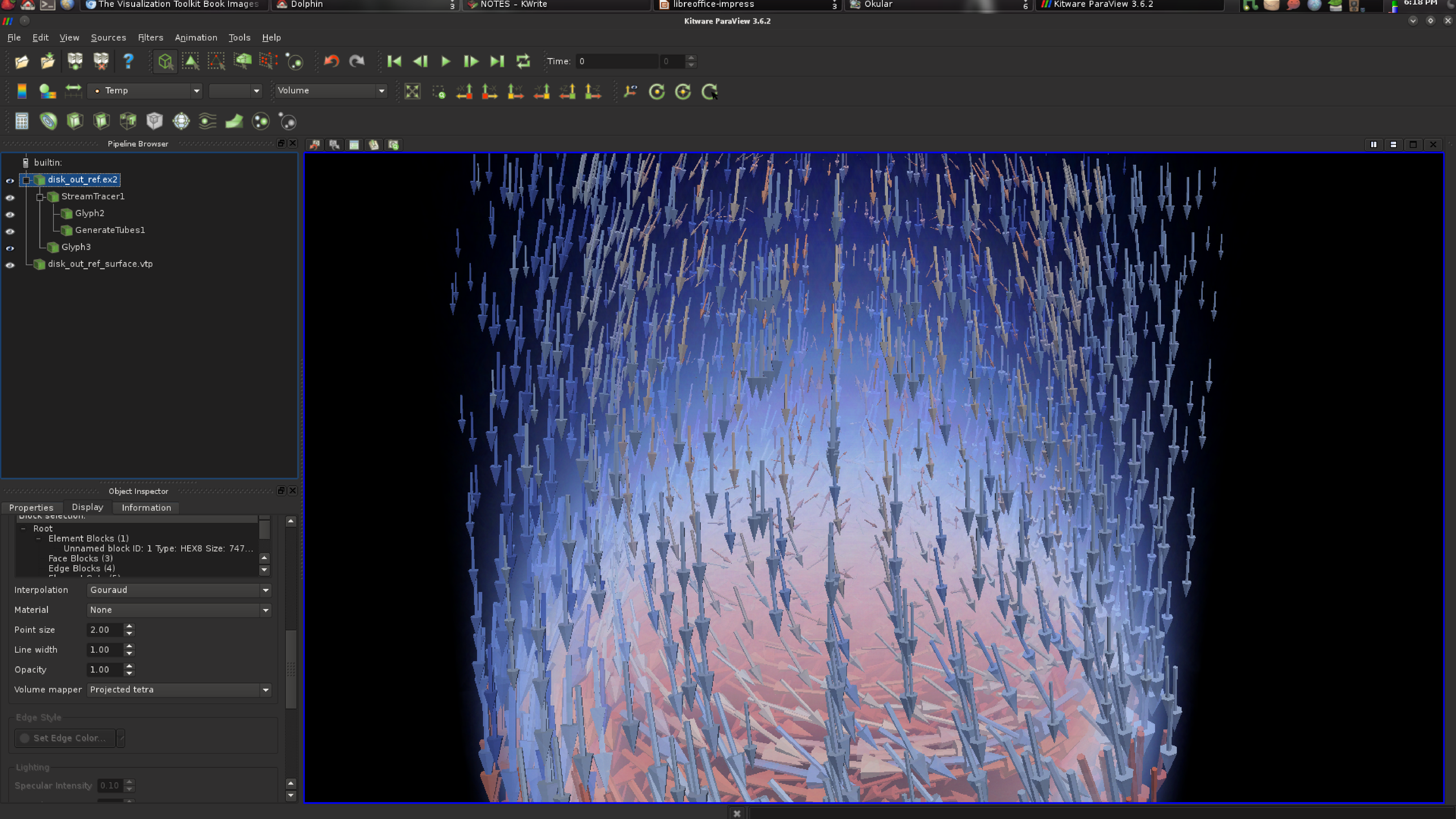
Challenges

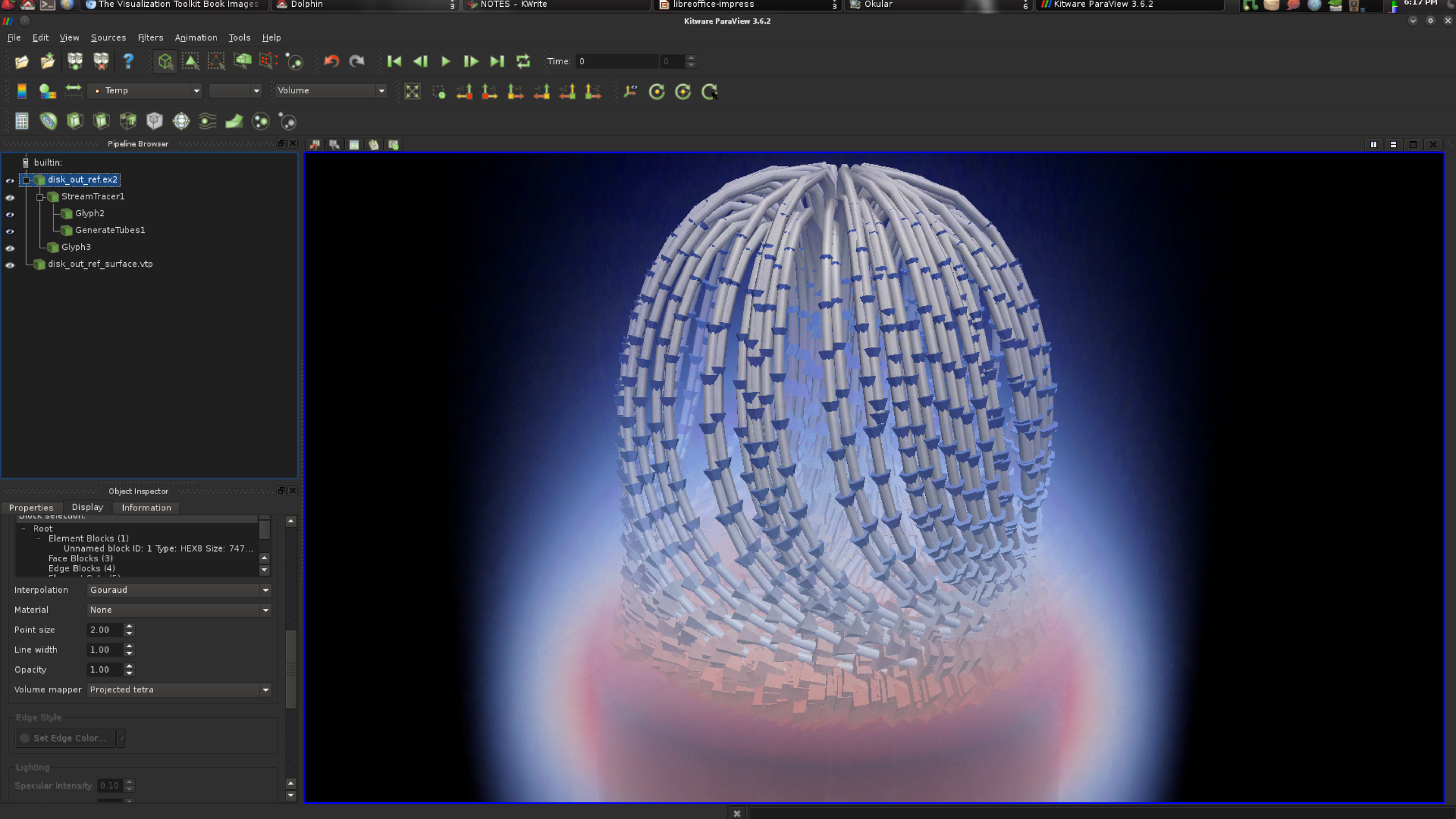
- These are complex data-sets:

Challenges

- These are complex data-sets:
  - What should we show?


The Visualization Toolkit Book Images
Dolphin
NOTES - KWrite
libreoffice-impress
Okular
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
Glyph2
GenerateTubes1
Glyph3
disk_out_ref_surface.vtp
Object Inspector
Properties Display Information
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
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Edge Style
Set Edge Color...
Lighting
Specular Intensity 0.10


The Visualization Toolkit Book Images
Dolphin
NOTES - KWrite
libreoffice-impress
Okular
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
Glyph2
GenerateTubes1
Glyph3
disk_out_ref_surface.vtp
Object Inspector
Properties Display Information
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
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Edge Style
Set Edge Color...
Lighting
Specular Intensity 0.10

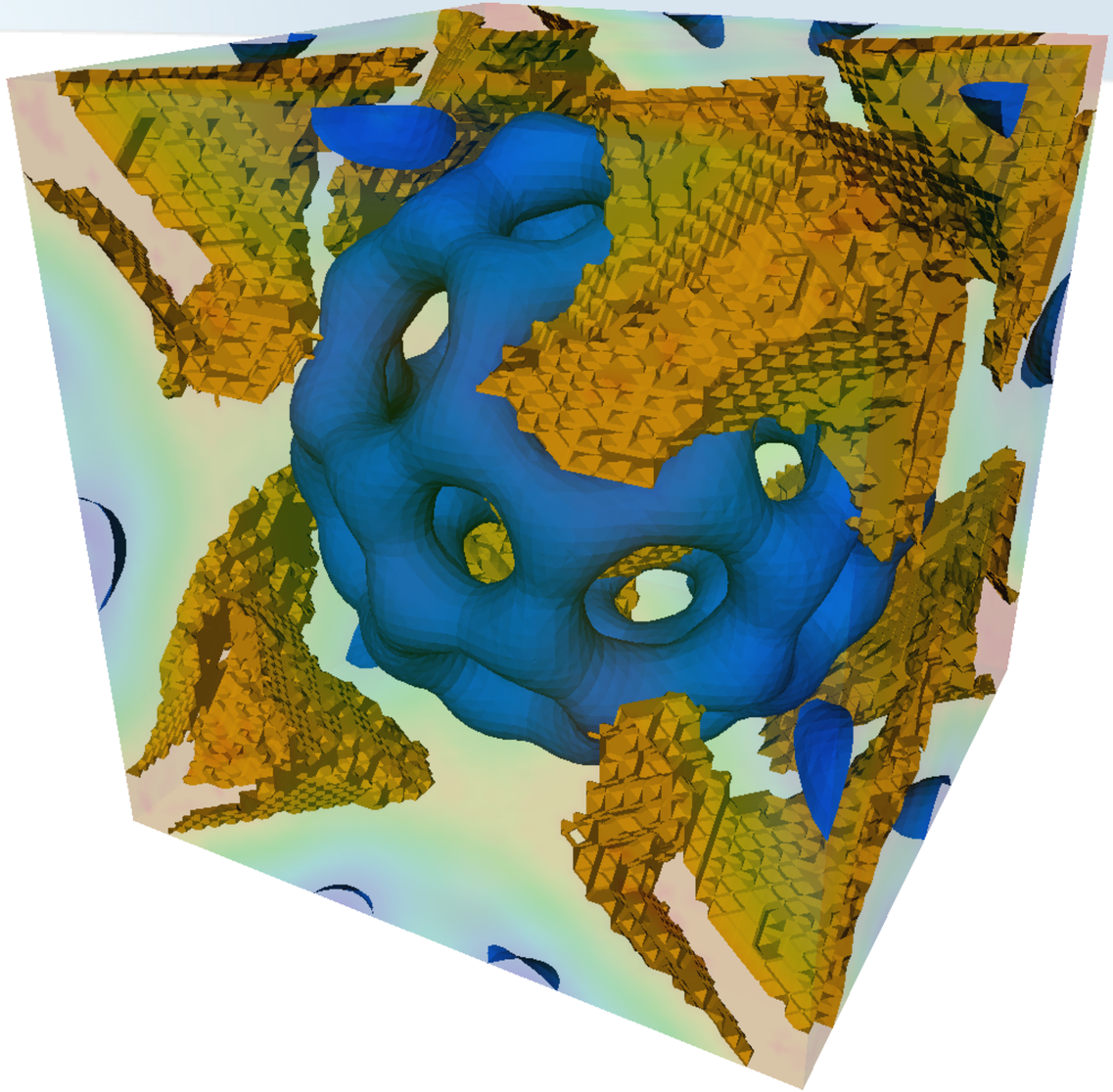

The Visualization Toolkit Book Images
Dolphin
NOTES - KWrite
libreoffice-impress
Okular
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Temp
Volume
Pipeline Browser
builtin:
disk_out_ref.ex2
StreamTracer1
Glyph2
GenerateTubes1
Glyph3
disk_out_ref_surface.vtp
Object Inspector
Properties Display Information
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
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Edge Style
Set Edge Color...
Lighting
Specular Intensity 0.10

Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?

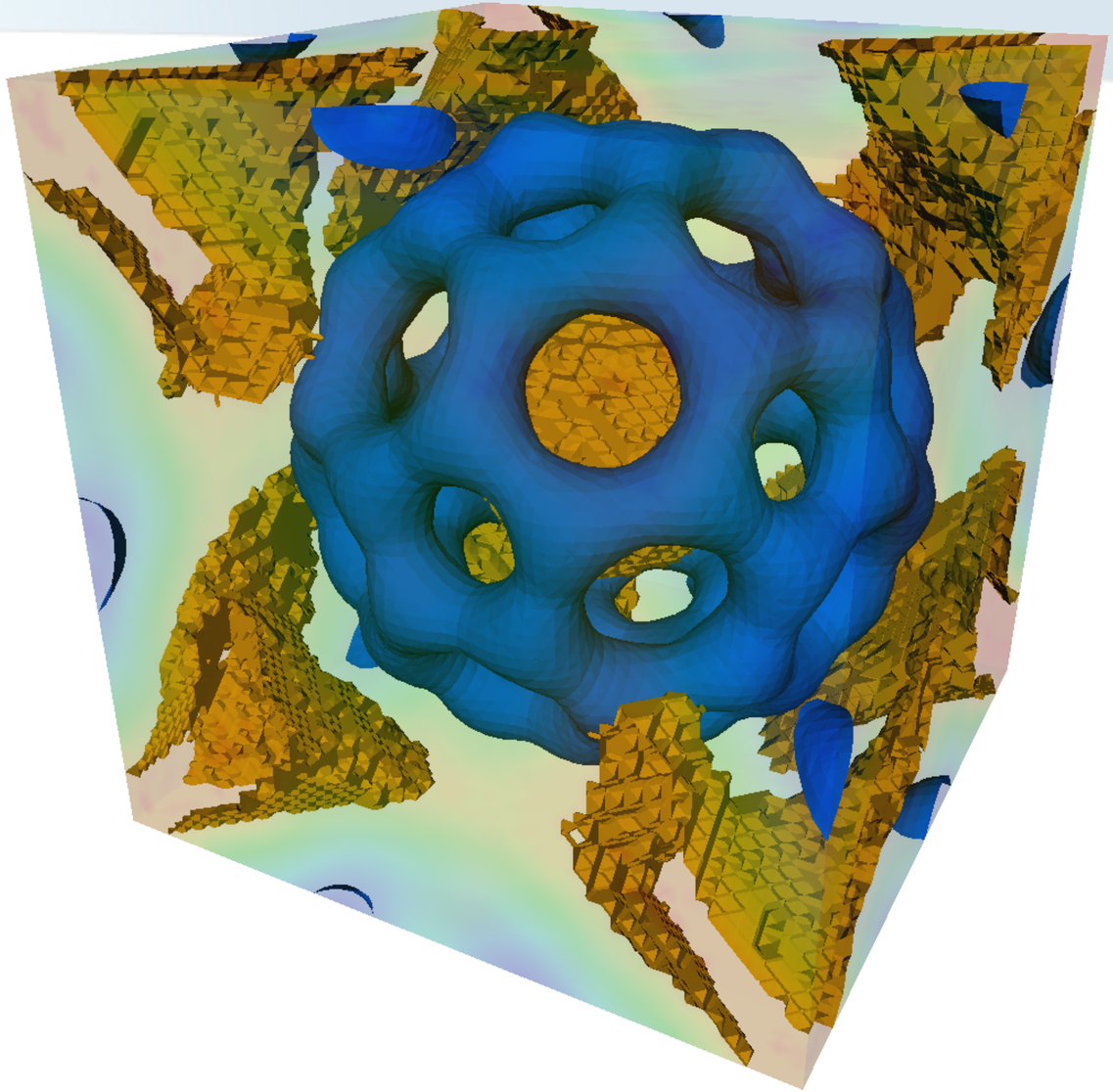
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?



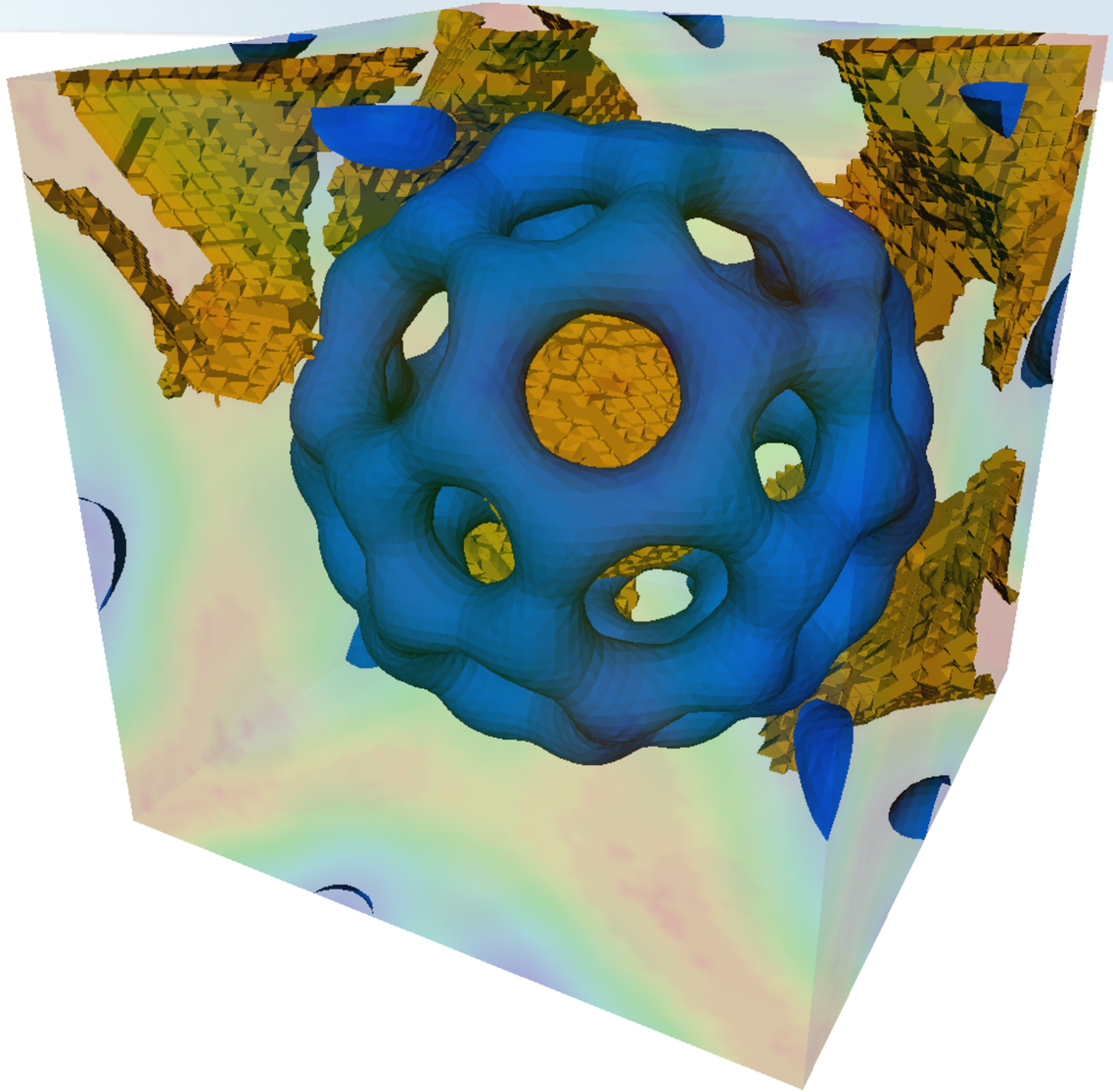
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?



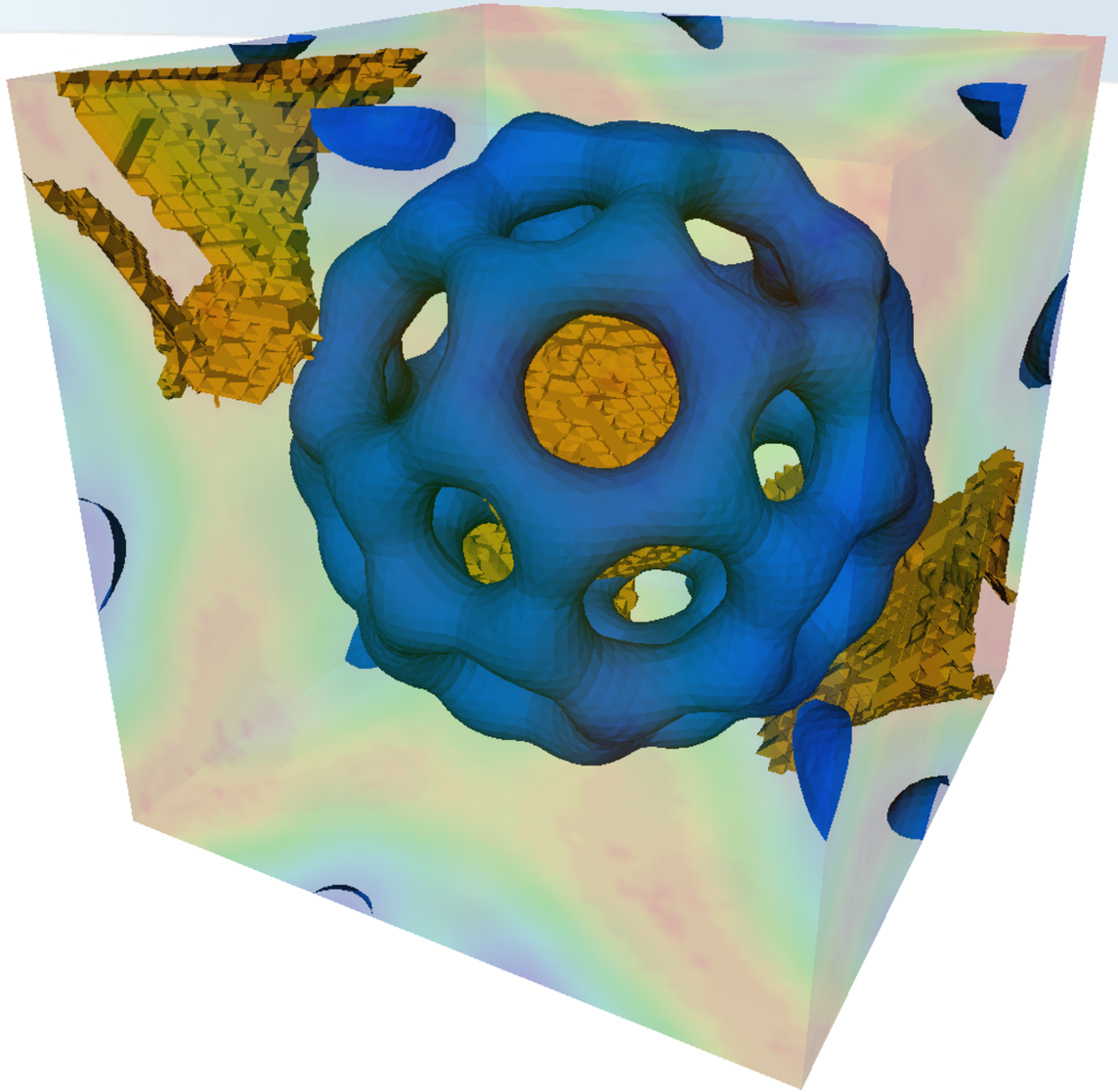
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?



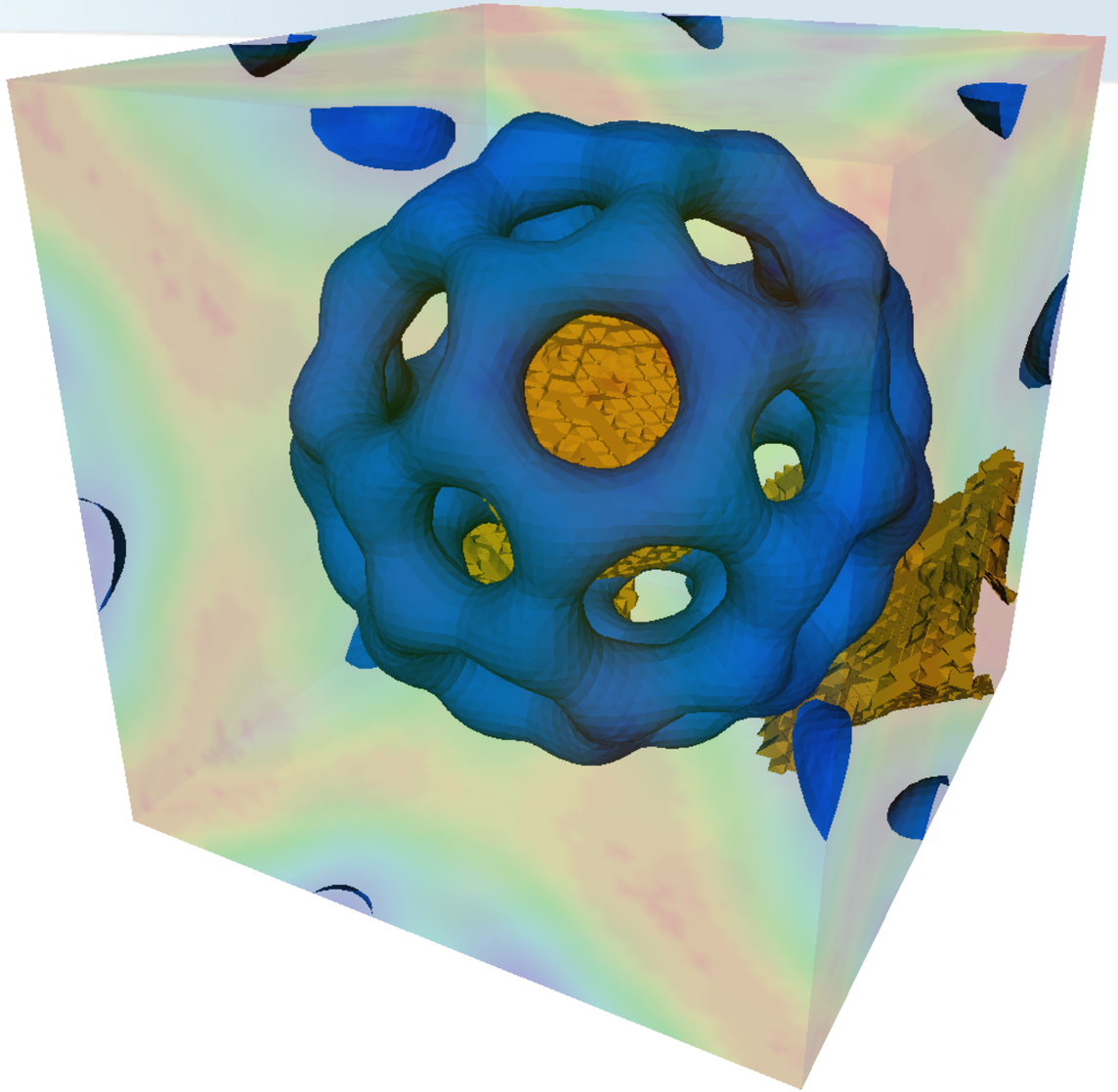
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?



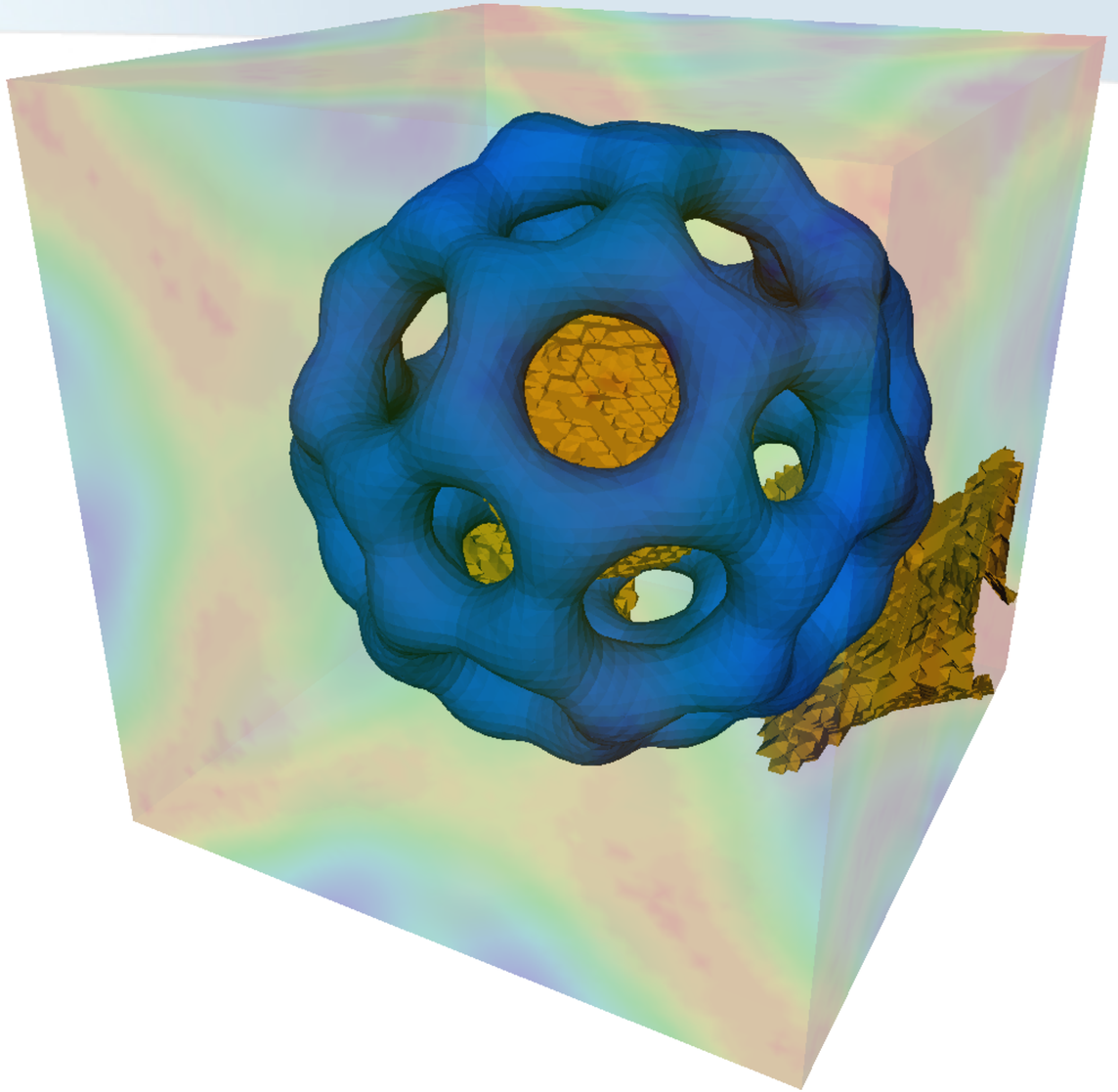
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?



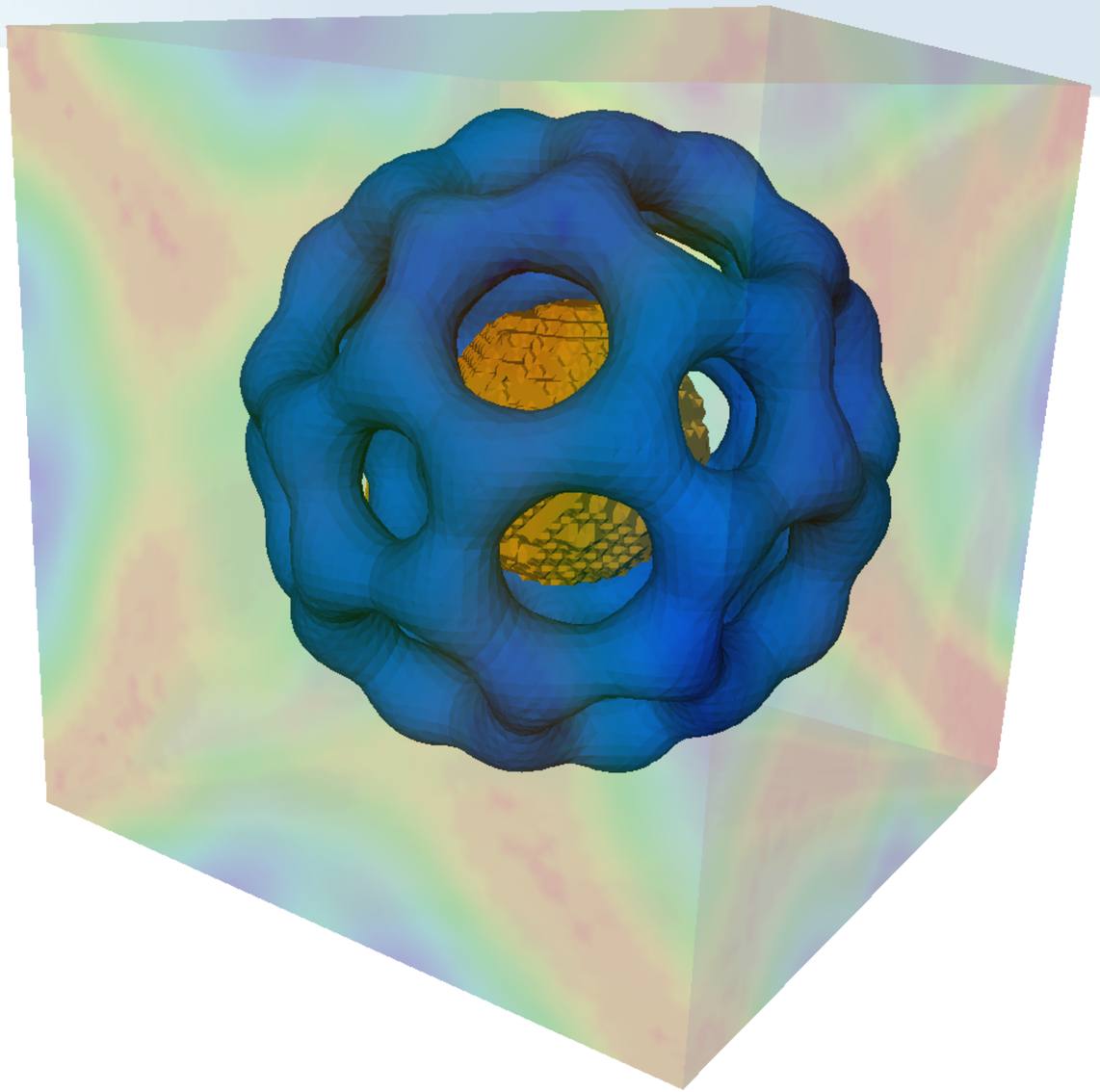
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?



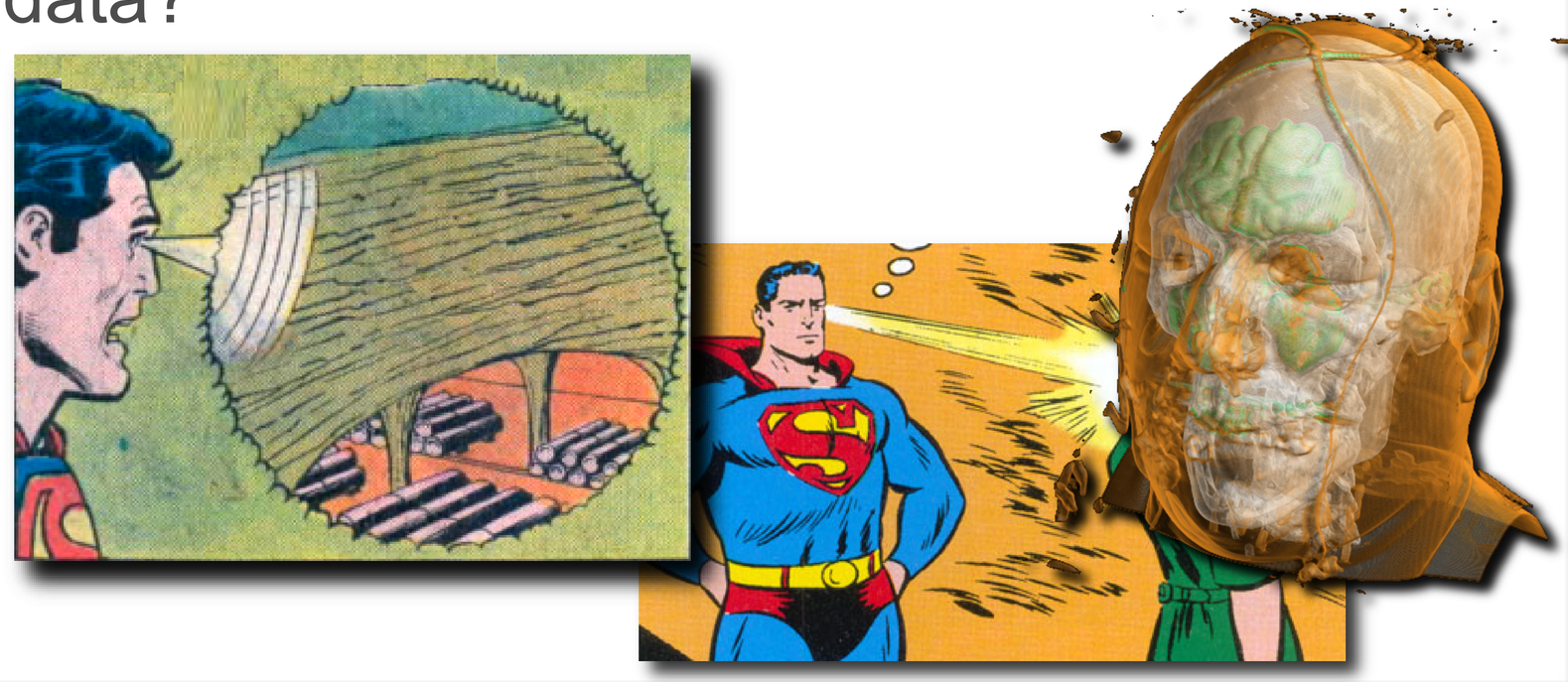
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?



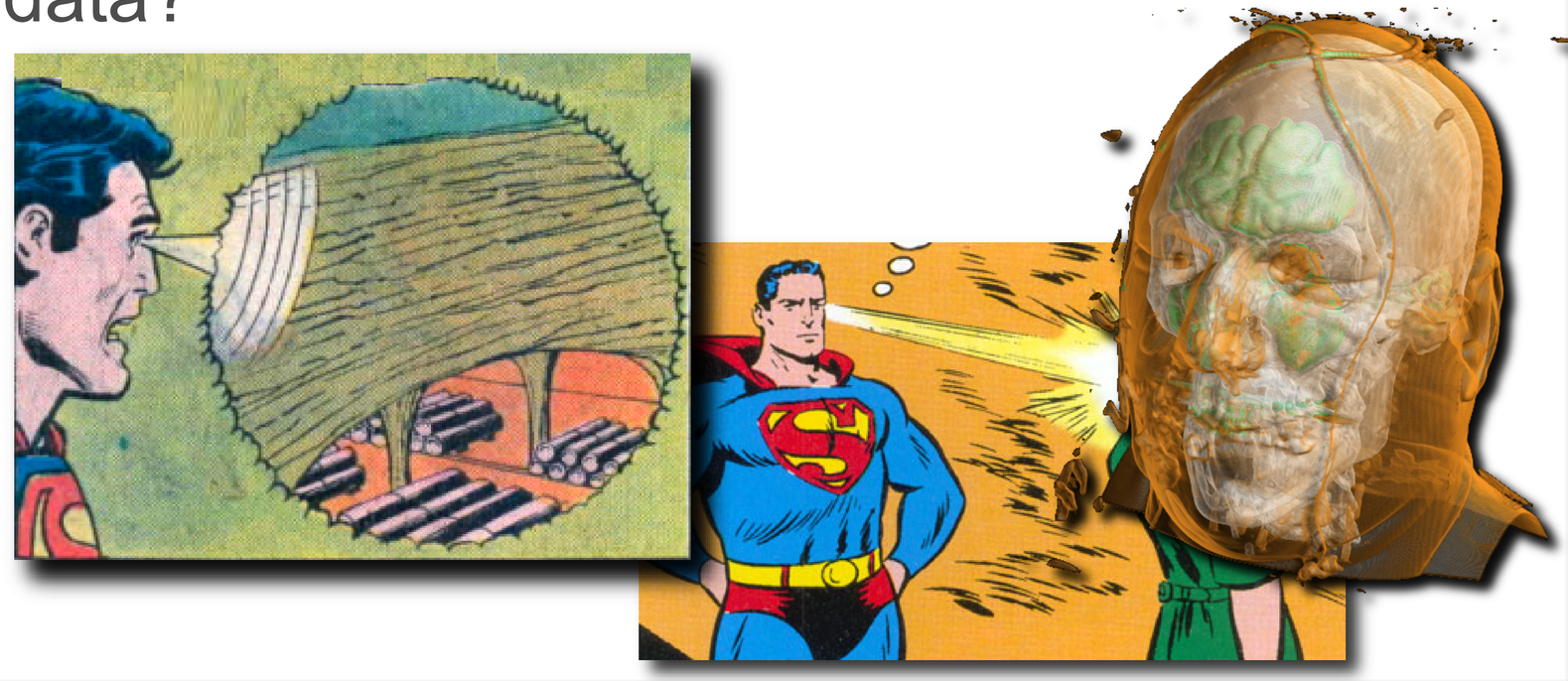
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?
- Dimensionality curse



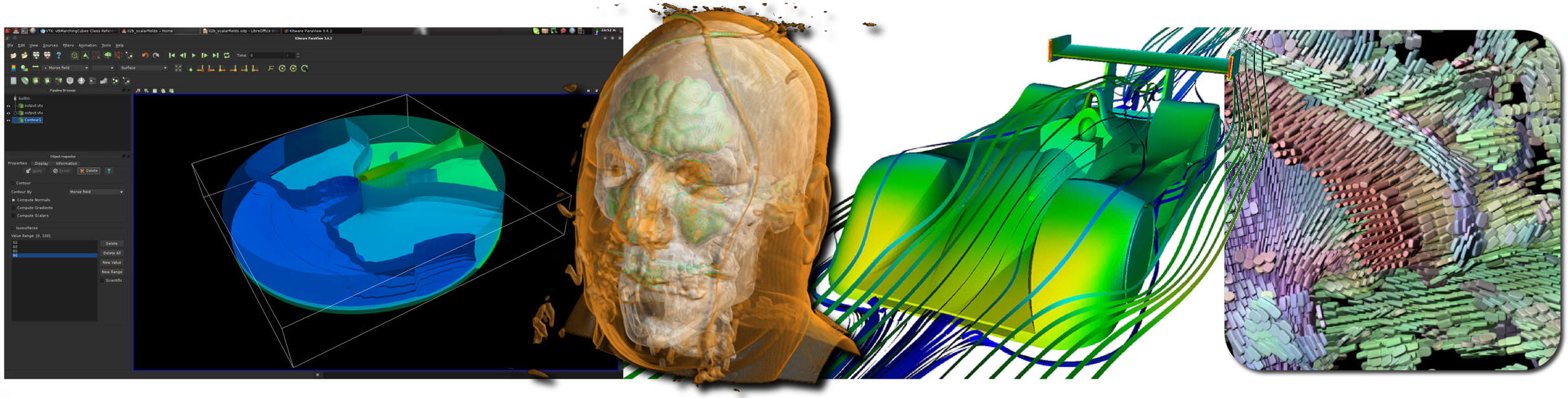
Challenges

- These are complex data-sets:
  - What should we show?
  - How can we explore the data?
- Dimensionality curse
- How can we do it fast?



Challenges

- Be creative
- Be efficient



Summary

Summary

- Numerical domain representations

Summary

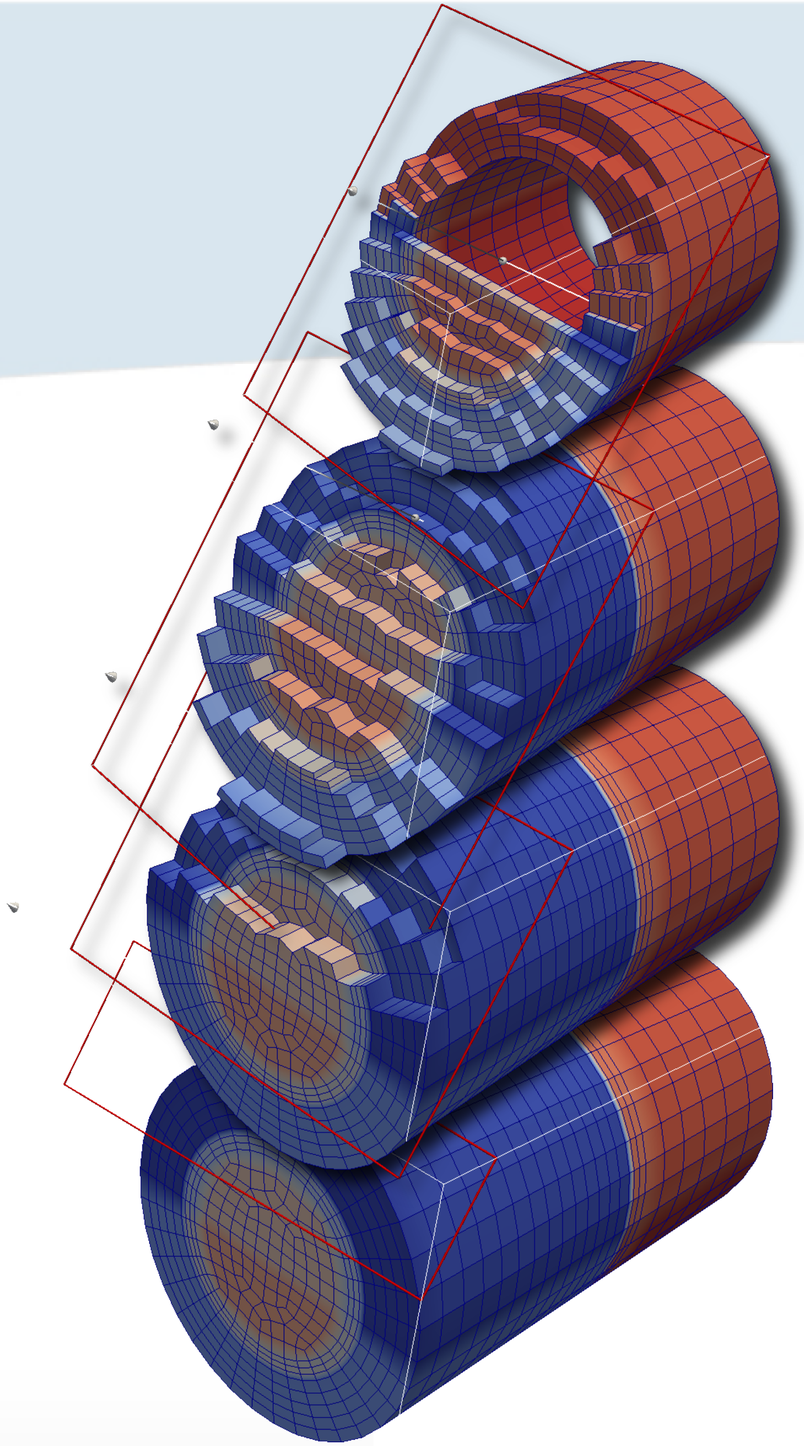
- Numerical domain representations
- Scalar field visualization
  - Level set extraction

Summary

- Numerical domain representations
- Scalar field visualization
  - Level set extraction
- Vector field visualization
  - Integral line extraction
  - Line Integral Convolution

Summary

- Numerical domain representations
- Scalar field visualization
  - Level set extraction
- Vector field visualization
  - Integral line extraction
  - Line Integral Convolution
- Tensor field visualization
  - Interpolation & convolution



Domain Representations

What do we mean by domain?

What do we mean by domain?

- Continuous discrete representation of a space $\mathbb{X}$

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes
  - Connectivity of the samples

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes
  - Connectivity of the samples
    - Continuity of the domain

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes
  - Connectivity of the samples
    - Continuity of the domain
    - Cellular elements (dimension of $\mathbb{X}$)

What do we mean by domain?

- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes
  - Connectivity of the samples
    - Continuity of the domain
    - Cellular elements (dimension of $\mathbb{X}$)
  - Cell interpolation scheme

What do we mean by domain?

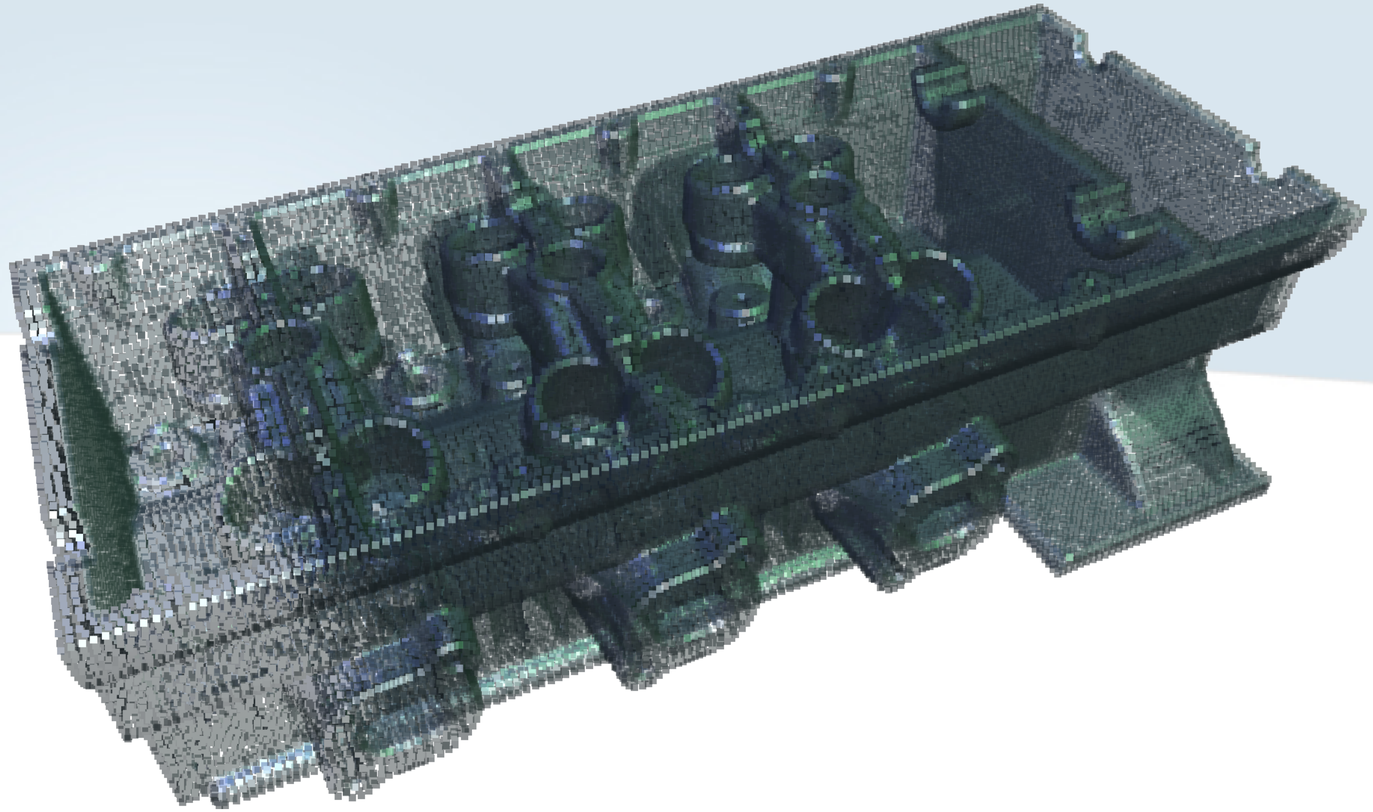
- *Continuous discrete* representation of a space $\mathbb{X}$
  - Finite set of samples
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes
  - Connectivity of the samples
    - Continuity of the domain
    - Cellular elements (dimension of $\mathbb{X}$)
  - Cell interpolation scheme
    - Continuity of the attributes

What do we mean by domain?

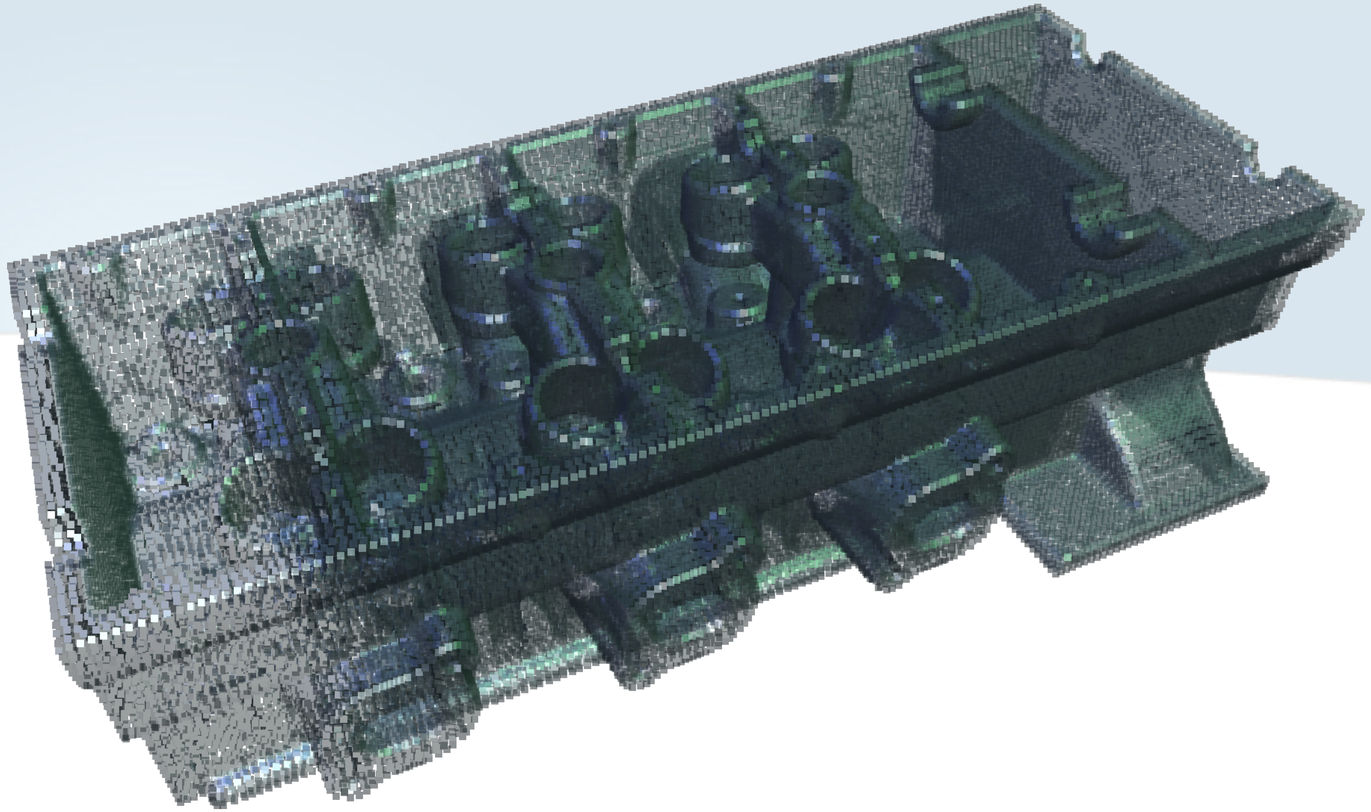
- *Continuous discrete* representation of a space $\mathbb{X}$
  - **Finite set of samples**
    - Positional information in an embedding space $\mathbb{E}$
    - Attributes
  - **Connectivity of the samples**
    - Continuity of the domain
    - Cellular elements (dimension of $\mathbb{X}$)
  - **Cell interpolation scheme**
    - Continuity of the attributes

Example

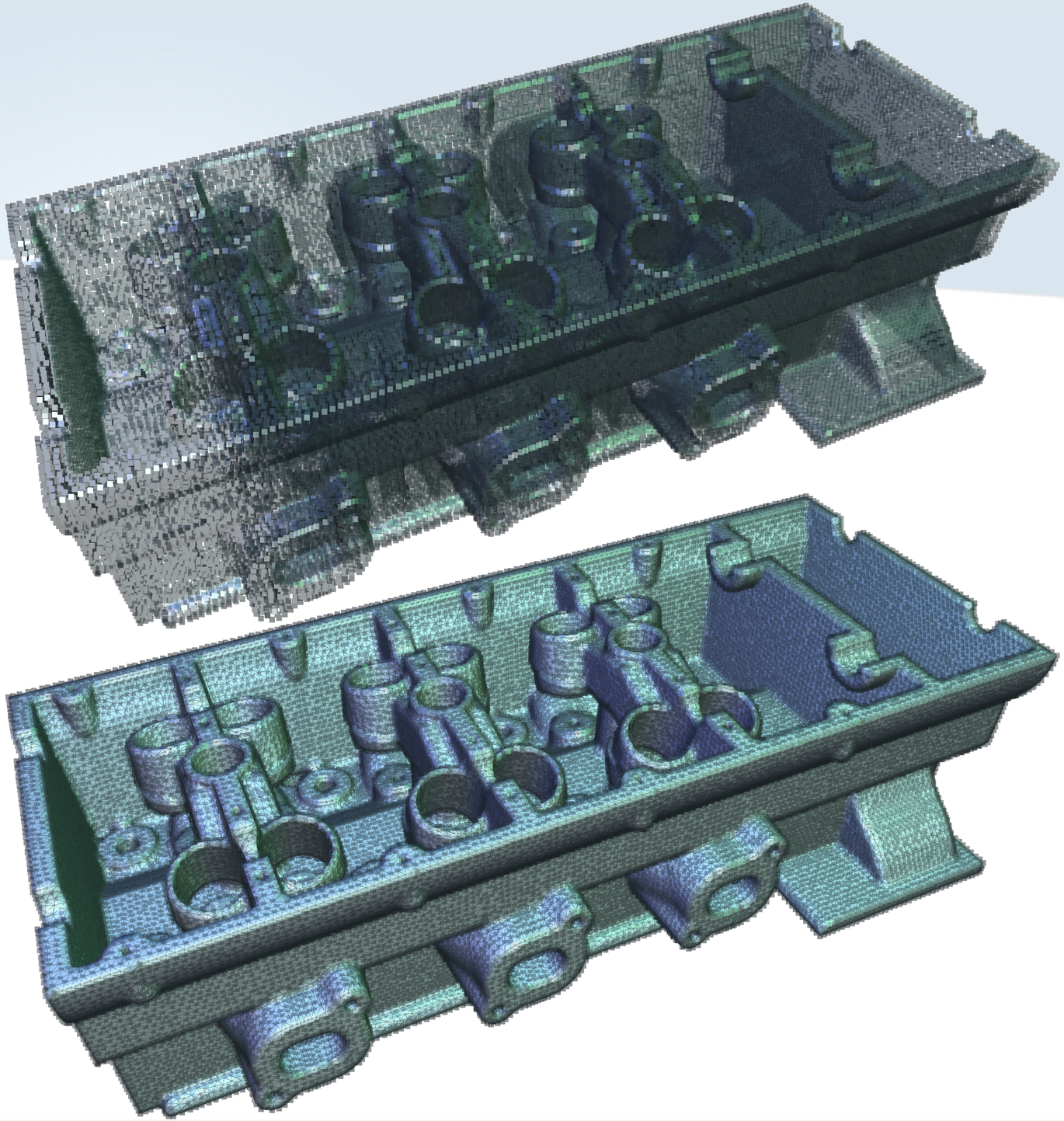
Example



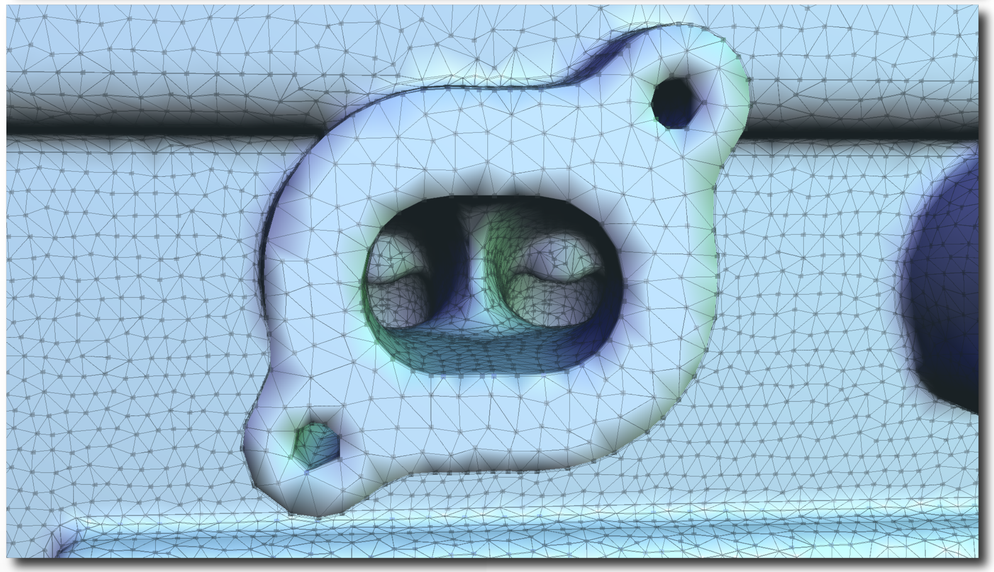
Example

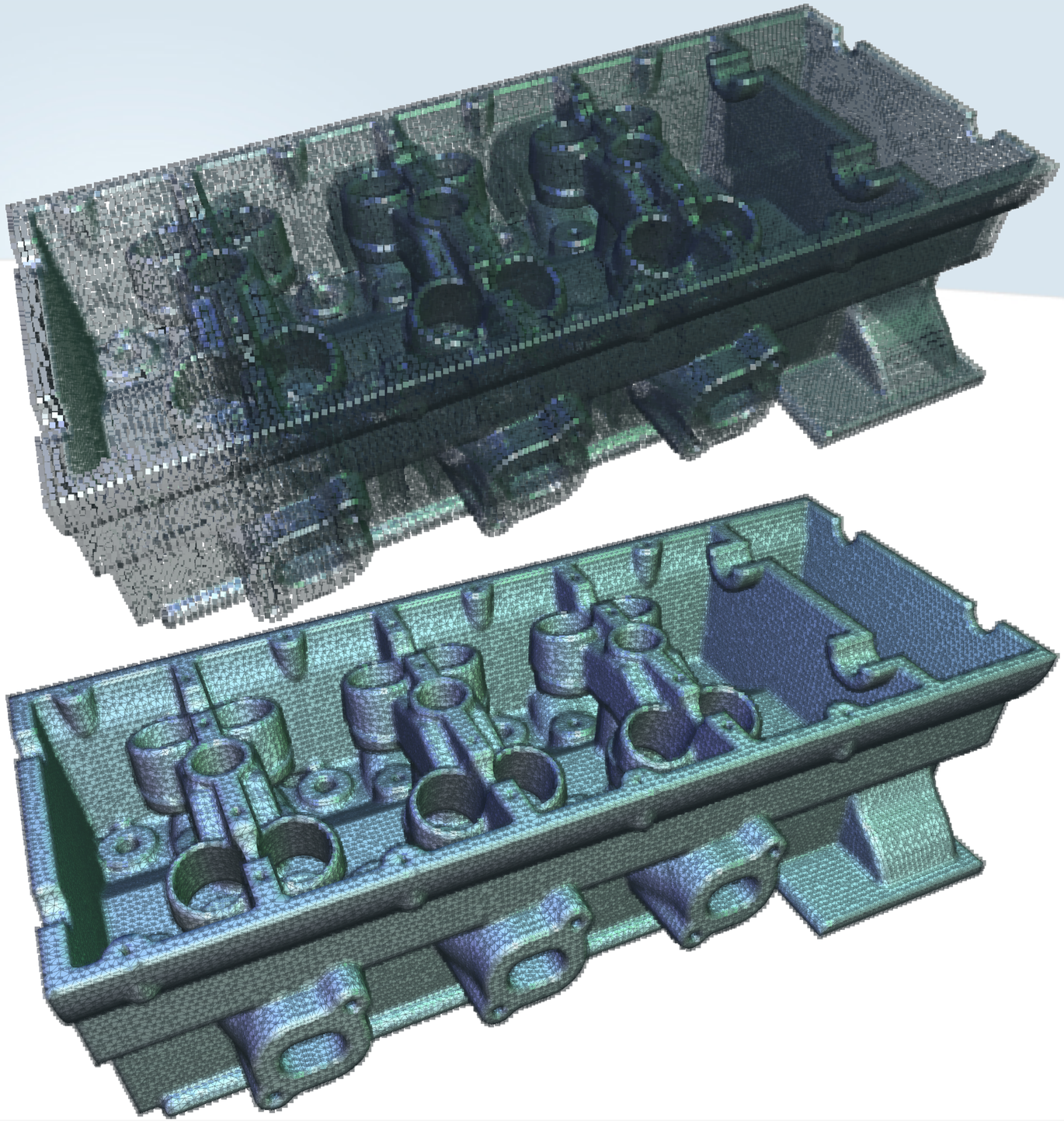


Example

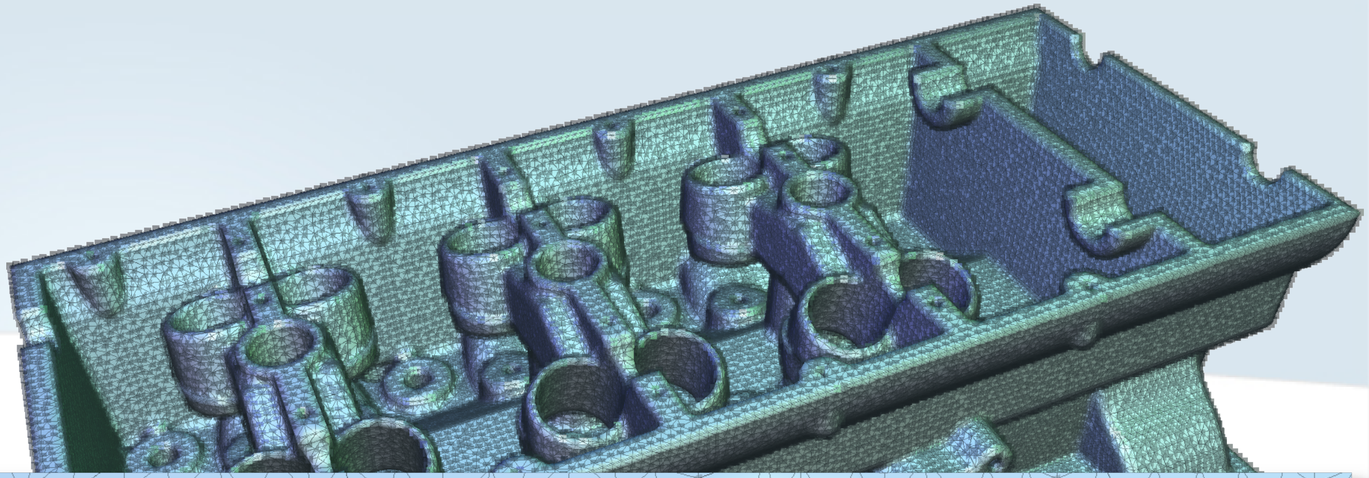


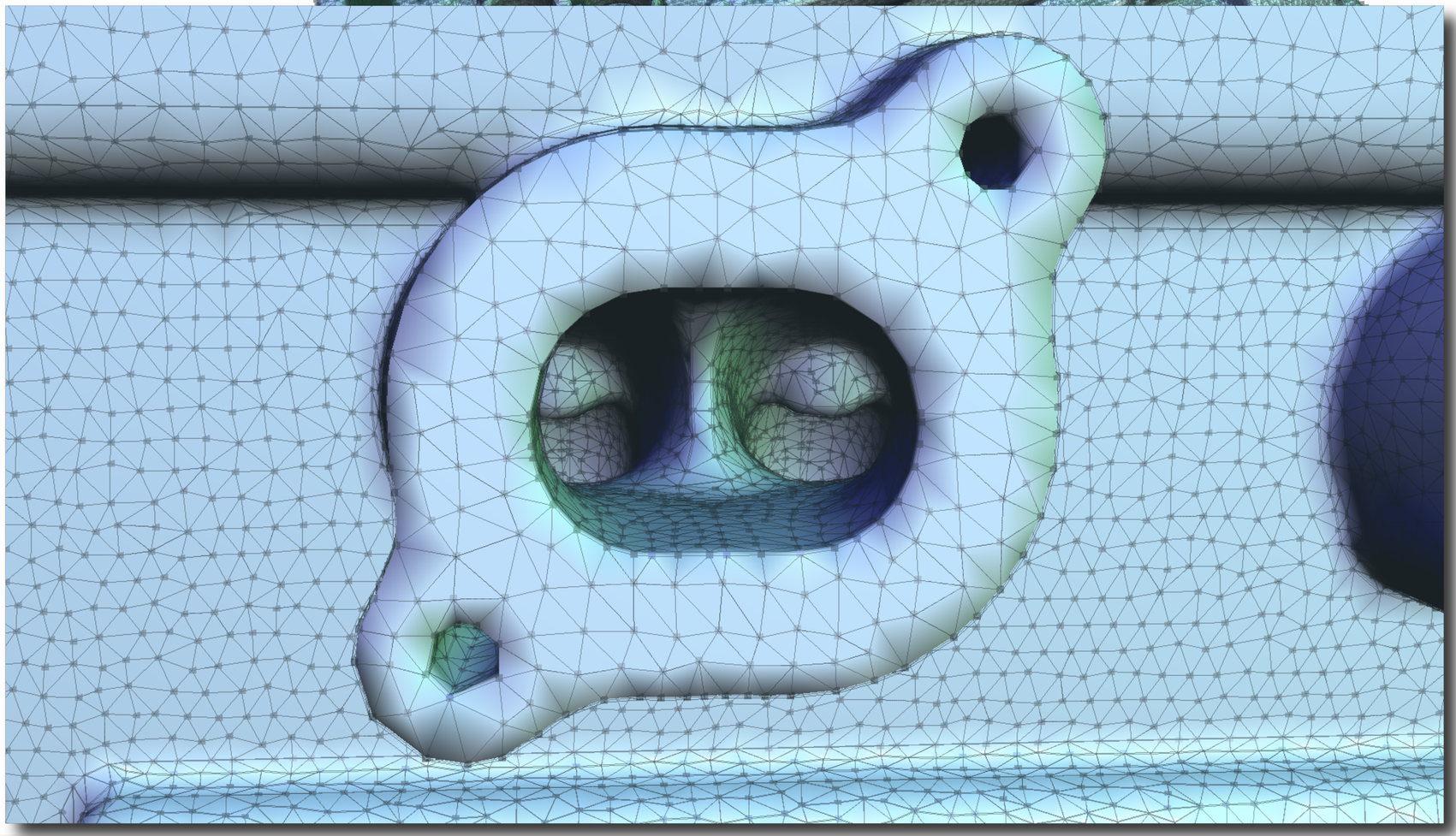
Example



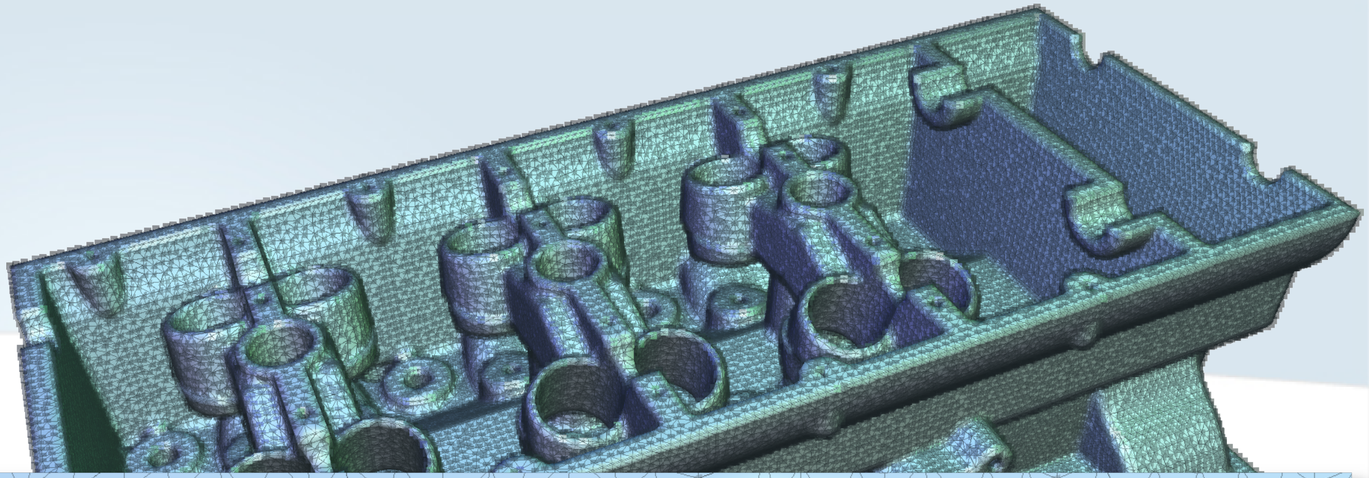


Example

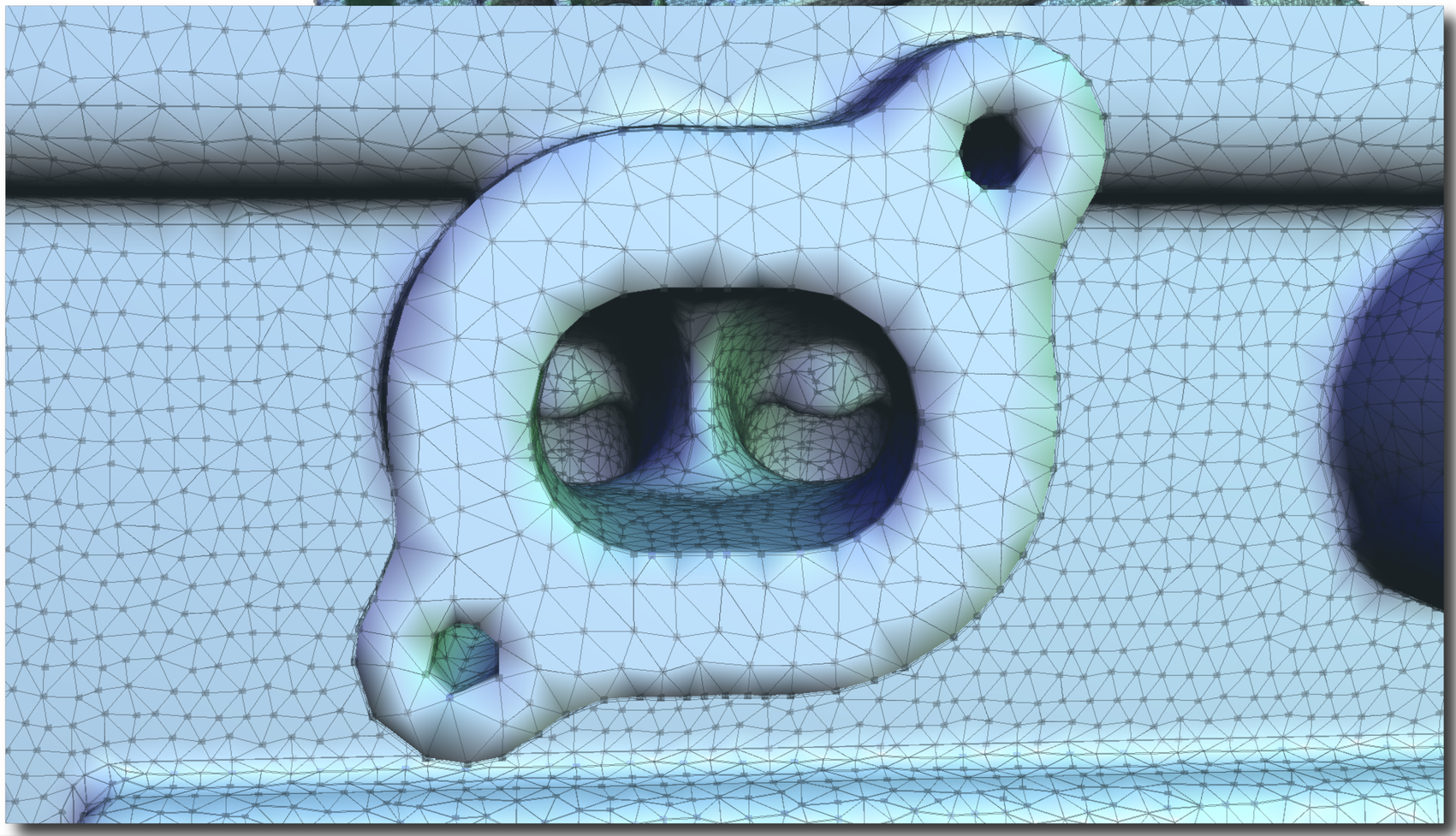




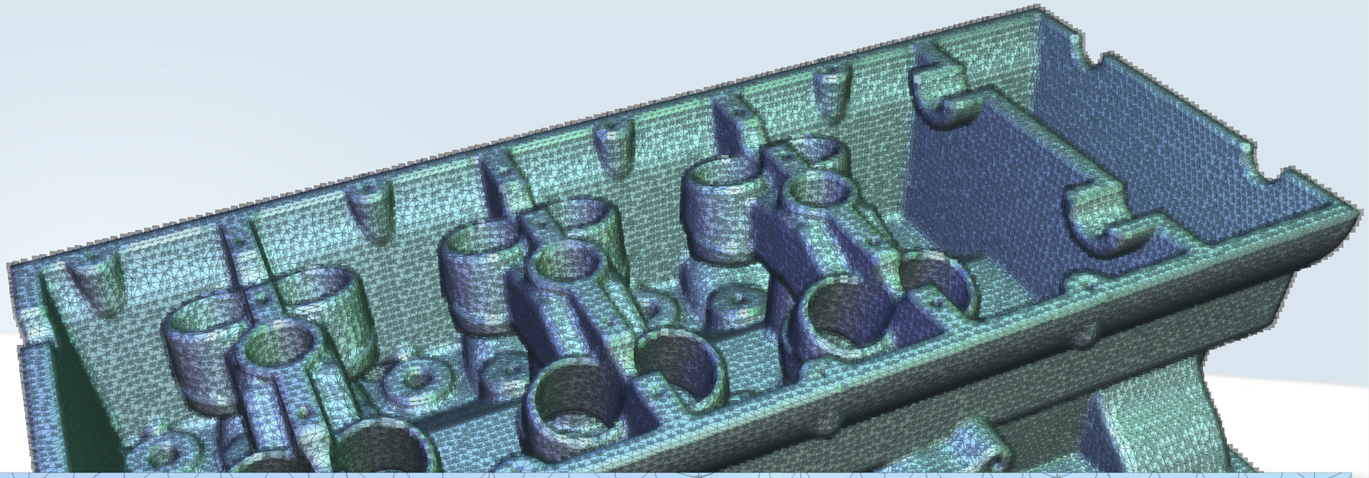
Example



- Domain
- Embedding space
- Cellular elements

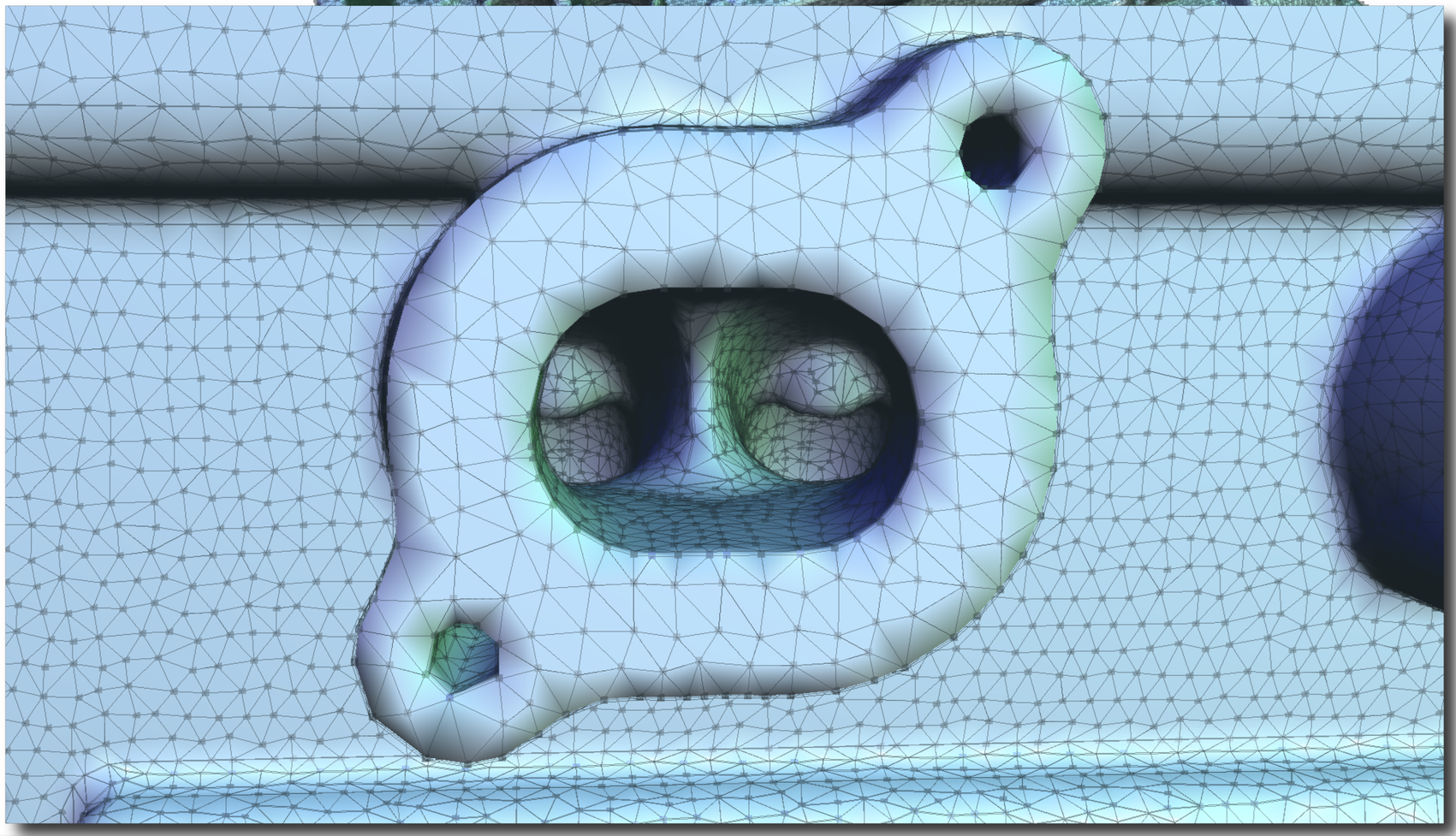


Example

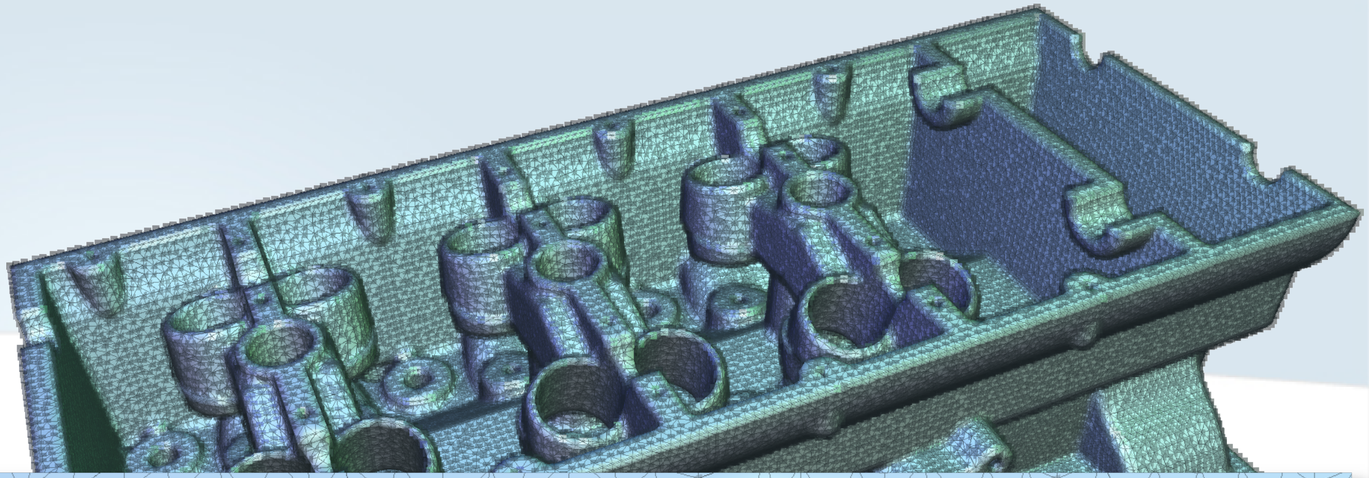


- Domain
  - 2-manifold $\mathcal{S}$
- Embedding space

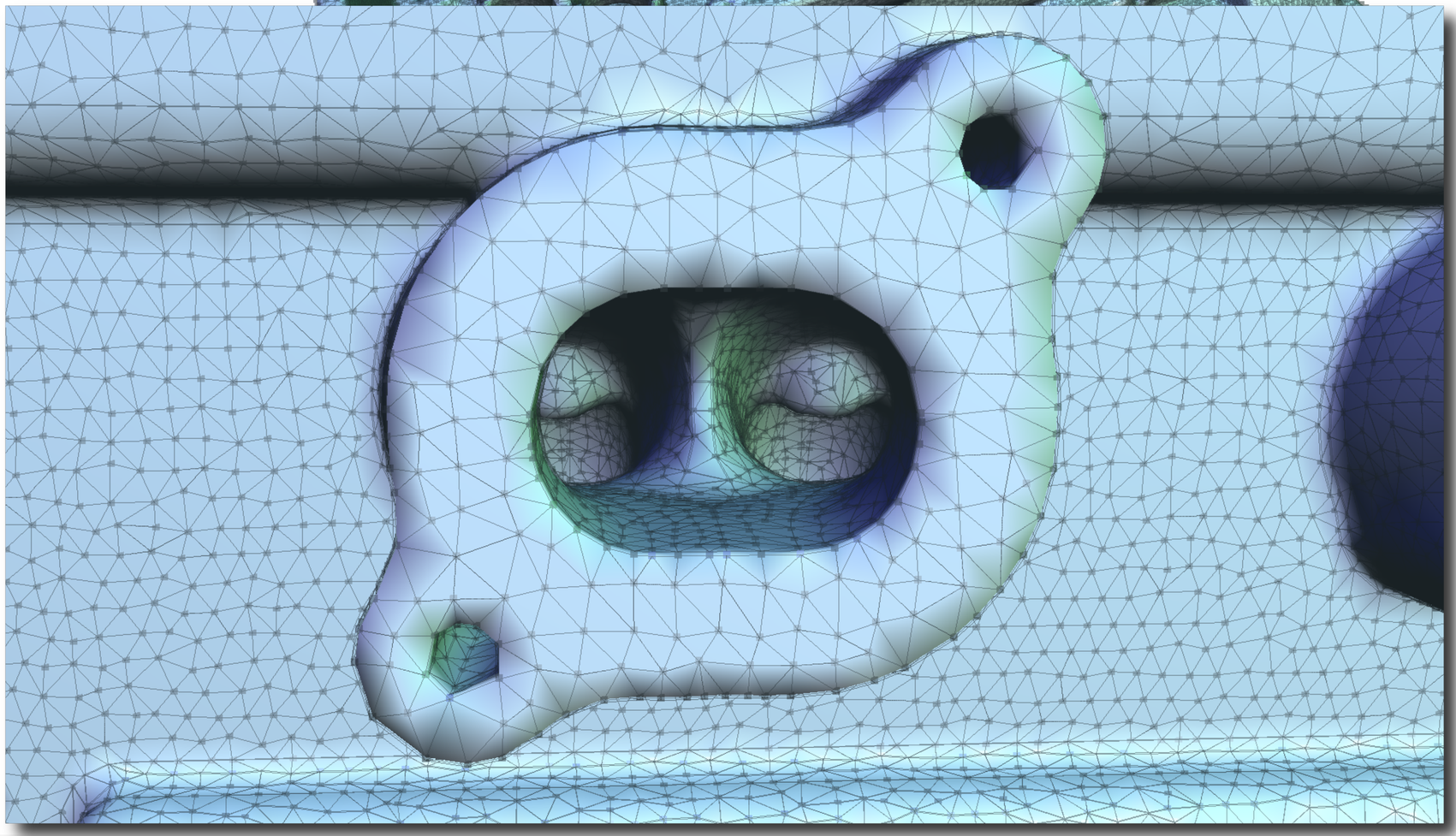
- Cellular elements



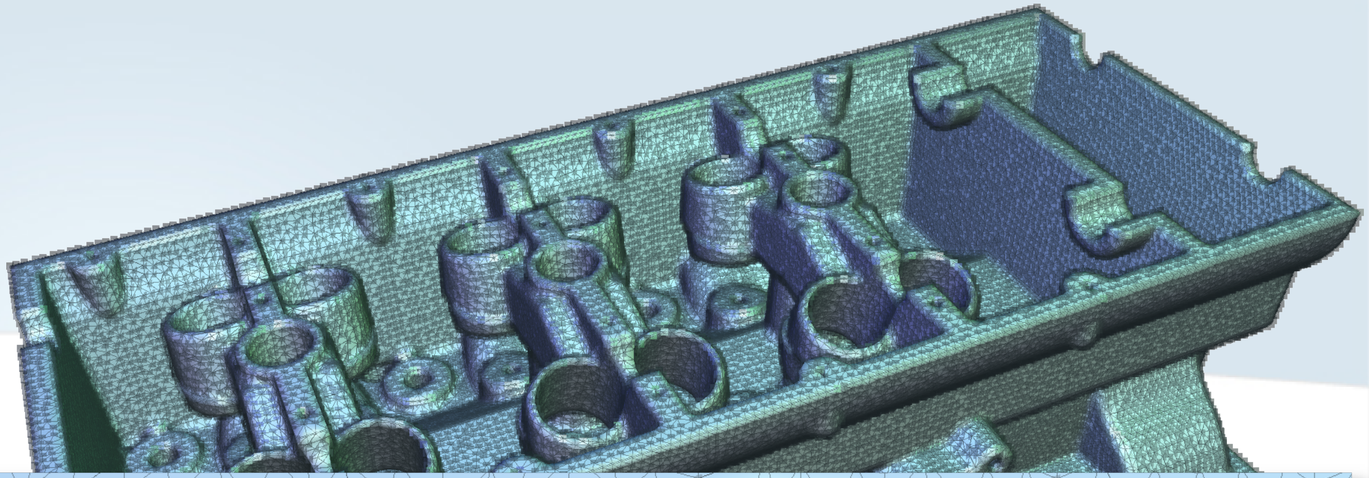
Example



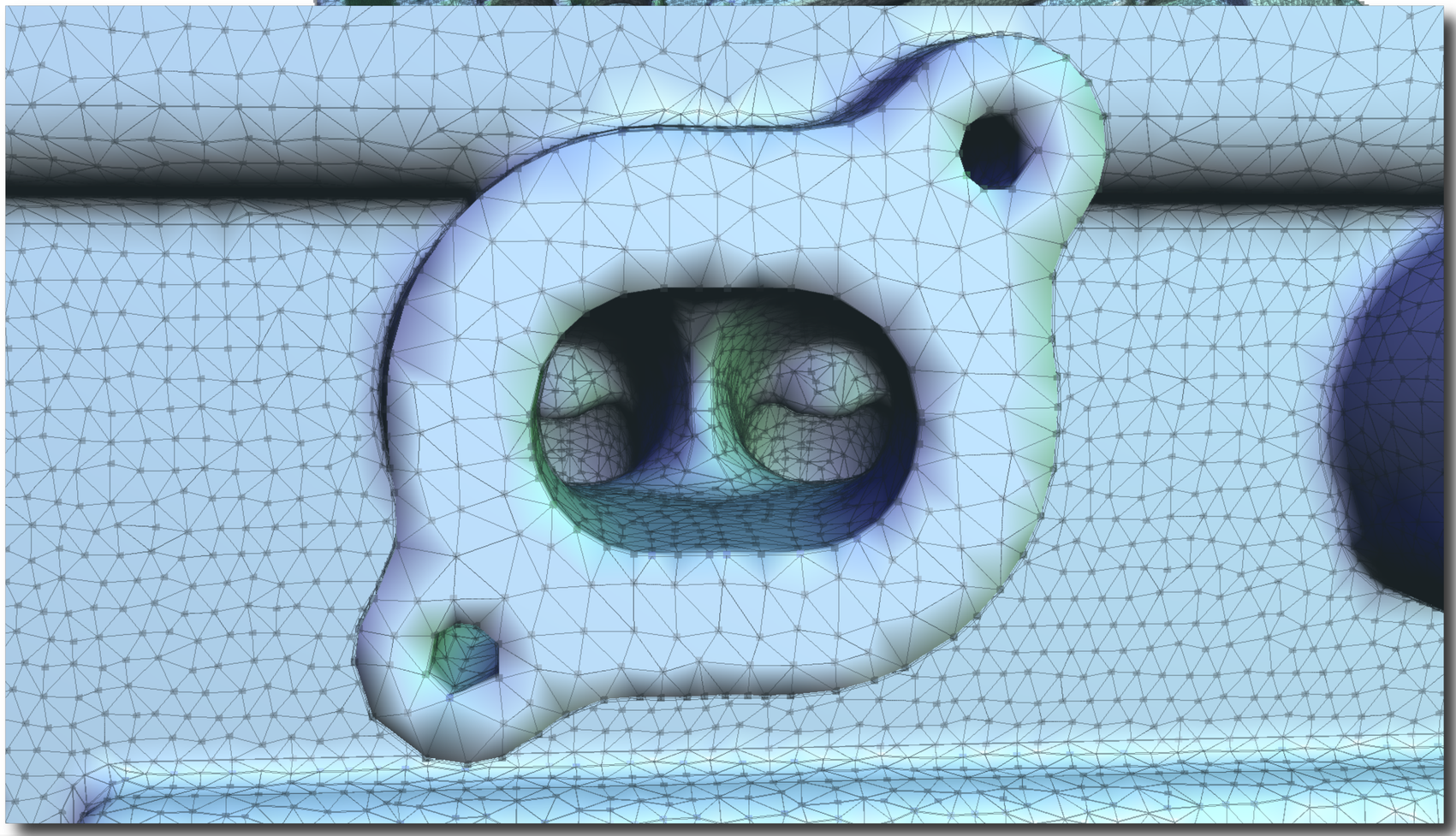
- Domain
  - 2-manifold $\mathcal{S}$
- Embedding space
  - $\mathbb{R}^3$
- Cellular elements



Example



- Domain
  - 2-manifold $\mathcal{S}$
- Embedding space
  - $\mathbb{R}^3$
- Cellular elements
  - Triangles



Domains of interest: Manifolds

Domains of interest: Manifolds

- d-manifold

Domains of interest: Manifolds

- d-manifold
  - Topological space such that:

Domains of interest: Manifolds

- d-manifold
  - Topological space such that:
    - Any of its open set is homeomorphic to $\mathbb{R}^d$

Domains of interest: Manifolds

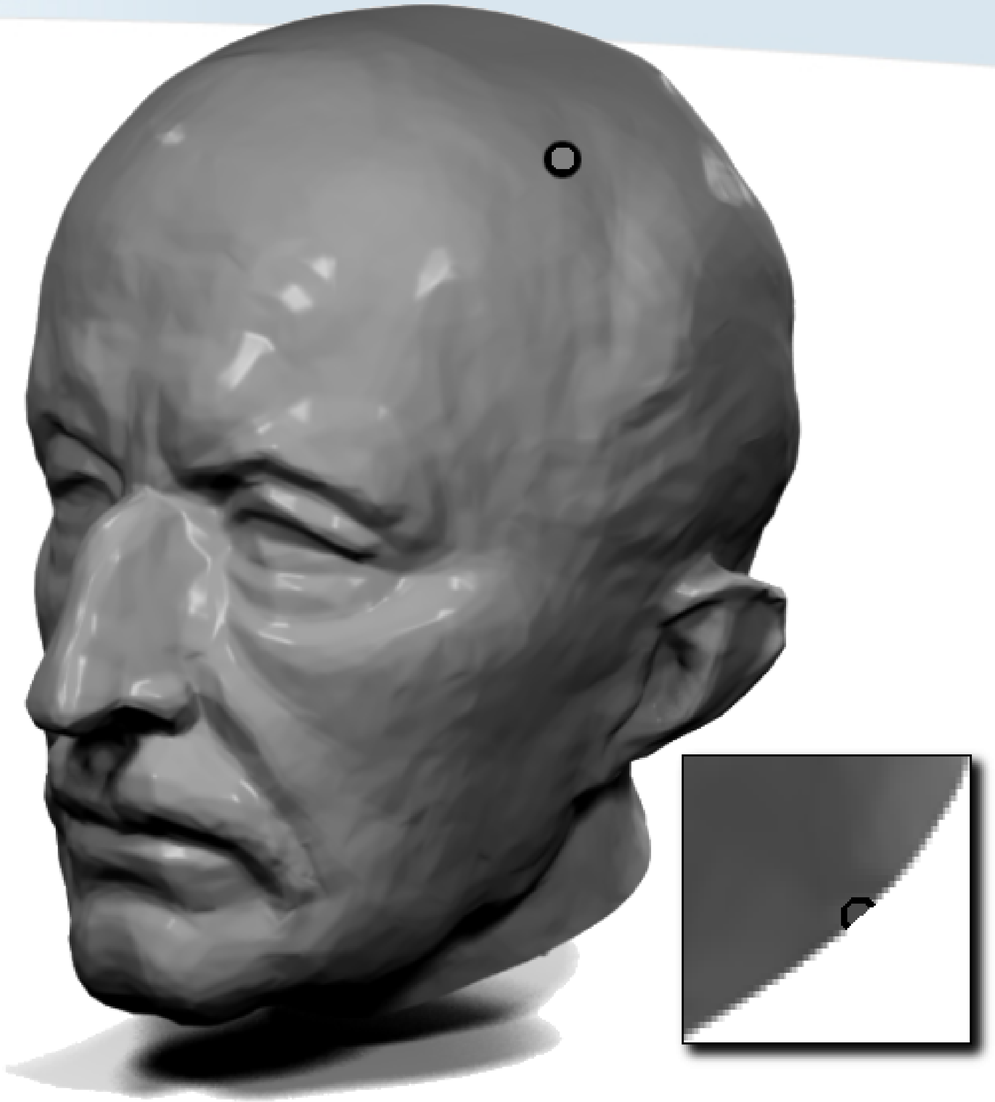
- d-manifold
  - Topological space such that:
    - Any of its open set is homeomorphic to $\mathbb{R}^d$

Domains of interest: Manifolds

- d-manifold with boundary
  - Topological space such that:
    - Any of its open set is homeomorphic to $\mathbb{R}^d$ *or* its half space

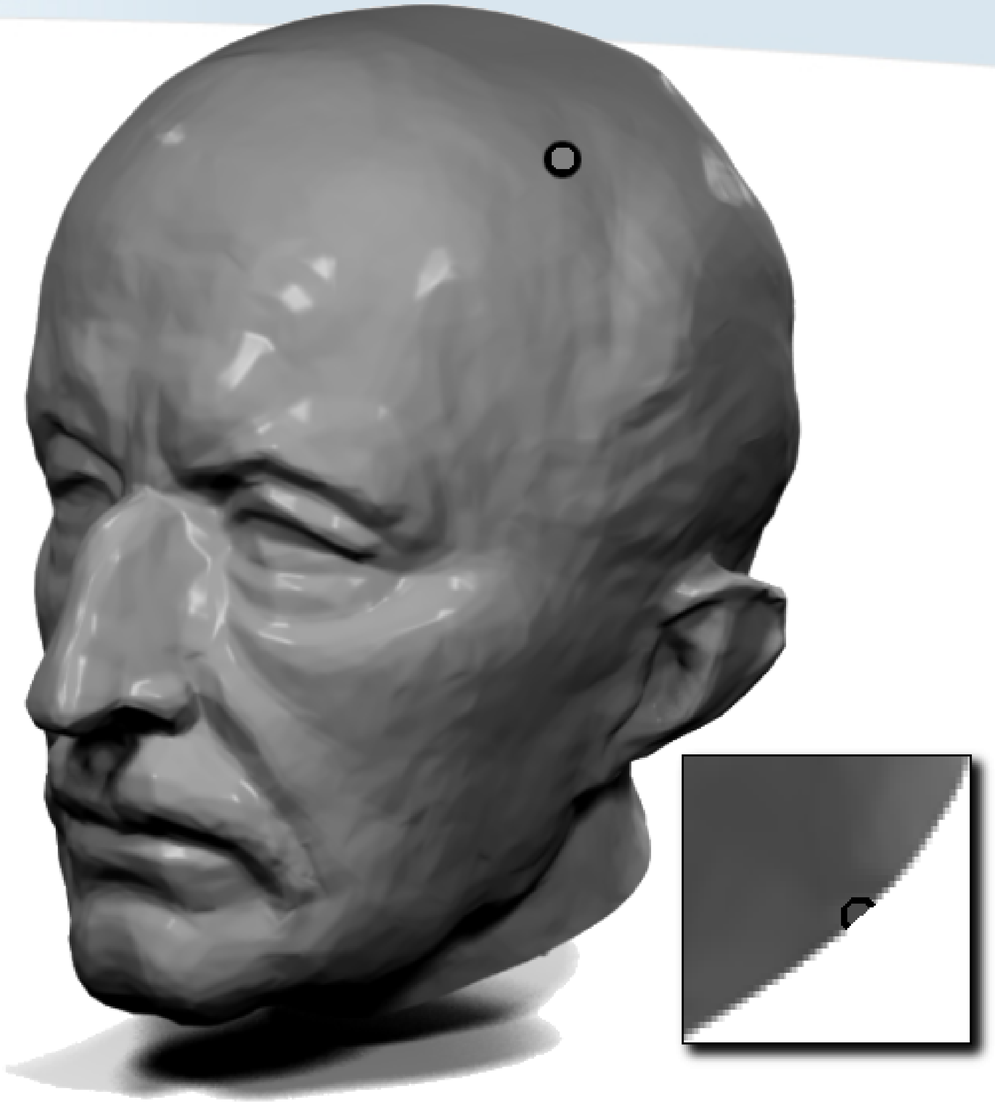
Domains of interest: Manifolds

- d-manifold with boundary
  - Topological space such that:
    - Any of its open set is homeomorphic to $\mathbb{R}^d$ *or* its half space



Domains of interest: Manifolds

- d-manifold with boundary
  - Topological space such that:
    - Any of its open set is homeomorphic to $\mathbb{R}^d$ *or* its half space
- Boundary: closed (d-1)-manifold



Manifold examples

Manifold examples

- Dimension?
- Embedding space?
- Boundary?

Manifold examples

$\mathbb{R}$

- Dimension?
- Embedding space?
- Boundary?

Manifold examples

$\mathbb{R}$

$[0, 1]$

- Dimension?
- Embedding space?
- Boundary?

Manifold examples

$$\mathbb{R}$$

$$[0, 1]$$

$$\{p \in \mathbb{R}^2 \ / \ \sqrt{p_x^2 + p_y^2} = r\}$$

- Dimension?
- Embedding space?
- Boundary?

Manifold examples

$$\mathbb{R}^2$$

- Dimension?
- Embedding space?
- Boundary?

$$[0, 1]$$

$$\{p \in \mathbb{R}^2 \ / \ \sqrt{p_x^2 + p_y^2} = r\}$$

Manifold examples

$$\mathbb{R}^2$$

$$[0,1] \times [0,1]$$

$$\{p \in \mathbb{R}^2 \ / \ \sqrt{p_x^2 + p_y^2} = r\}$$

- Dimension?
- Embedding space?
- Boundary?

Manifold examples

$$\mathbb{R}^2$$

$$[0,1] \times [0,1]$$

$$\{p \in \mathbb{R}^2 \ / \ \sqrt{p_x^2 + p_y^2} \leq r\}$$

- Dimension?
- Embedding space?
- Boundary?

Manifold examples

$$\mathbb{R}^3$$

$$[0,1] \times [0,1]$$

$$\{p \in \mathbb{R}^2 \ / \ \sqrt{p_x^2 + p_y^2} \leq r\}$$

- Dimension?
- Embedding space?
- Boundary?

Manifold examples

- Dimension?
- Embedding space?
- Boundary?

$$\mathbb{R}^3$$

$$[0,1] \times [0,1] \times [0,1]$$

$$\{p \in \mathbb{R}^2 \ / \ \sqrt{p_x^2 + p_y^2} \leq r\}$$

Manifold examples

$$\mathbb{R}^3$$

- Dimension?
- Embedding space?
- Boundary?

$$[0,1] \times [0,1] \times [0,1]$$

$$\{p \in \mathbb{R}^3 \ / \ \sqrt{p_x^2 + p_y^2 + p_z^2} = r\}$$

Manifold examples

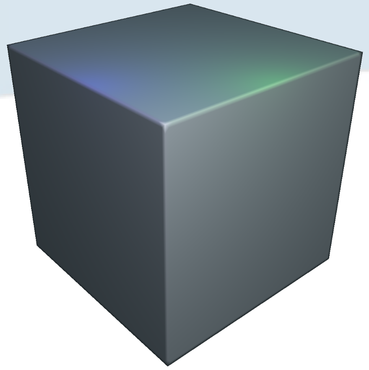
- Dimension?
- Embedding space?
- Boundary?

$$\mathbb{R}^3$$

$$[0,1] \times [0,1] \times [0,1]$$

$$\{p \in \mathbb{R}^3 \ / \ \sqrt{p_x^2 + p_y^2 + p_z^2} \leq r\}$$

Manifold examples



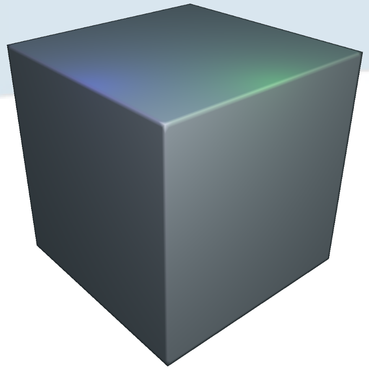
$$\mathbb{R}^3$$

- Dimension?
- Embedding space?
- Boundary?

$$[0,1] \times [0,1] \times [0,1]$$

$$\{p \in \mathbb{R}^3 \ / \ \sqrt{p_x^2 + p_y^2 + p_z^2} \leq r\}$$

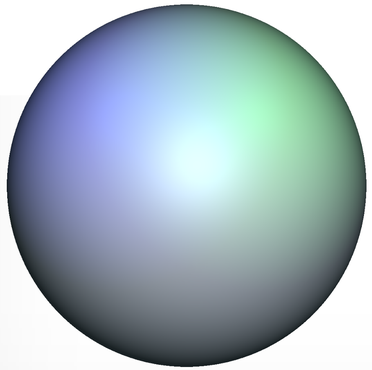
Manifold examples



$$\mathbb{R}^3$$

- Dimension?
- Embedding space?
- Boundary?

$$[0,1] \times [0,1] \times [0,1]$$



$$\{p \in \mathbb{R}^3 \ / \ \sqrt{p_x^2 + p_y^2 + p_z^2} \leq r\}$$

Euclidean spaces on a computer

Euclidean spaces on a computer

- Given an origin

Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$

Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis

Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$

Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_2 = (0, 1, 0, \ldots, 0) \in \mathbb{R}^n$

Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_2 = (0, 1, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_3 = (0, 0, 1, \ldots, 0) \in \mathbb{R}^n$

Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_2 = (0, 1, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_3 = (0, 0, 1, \ldots, 0) \in \mathbb{R}^n$
  - $\ldots$
  - $v_n = (0, 0, 0, \ldots, 1) \in \mathbb{R}^n$

Euclidean spaces on a computer

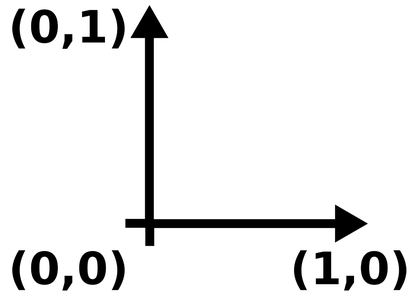
- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_2 = (0, 1, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_3 = (0, 0, 1, \ldots, 0) \in \mathbb{R}^n$
  - $\ldots$
  - $v_n = (0, 0, 0, \ldots, 1) \in \mathbb{R}^n$
- Points can be identified uniquely in that space

Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_2 = (0, 1, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_3 = (0, 0, 1, \ldots, 0) \in \mathbb{R}^n$
  - $\ldots$
  - $v_n = (0, 0, 0, \ldots, 1) \in \mathbb{R}^n$
- Points can be identified uniquely in that space

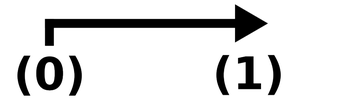
Euclidean spaces on a computer

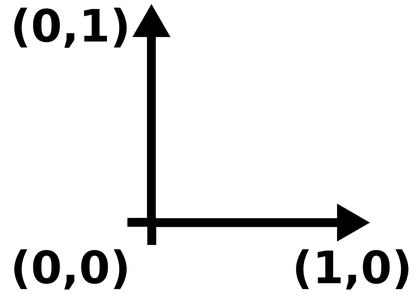
- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_2 = (0, 1, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_3 = (0, 0, 1, \ldots, 0) \in \mathbb{R}^n$
  - $\ldots$
  - $v_n = (0, 0, 0, \ldots, 1) \in \mathbb{R}^n$
- Points can be identified uniquely in that space

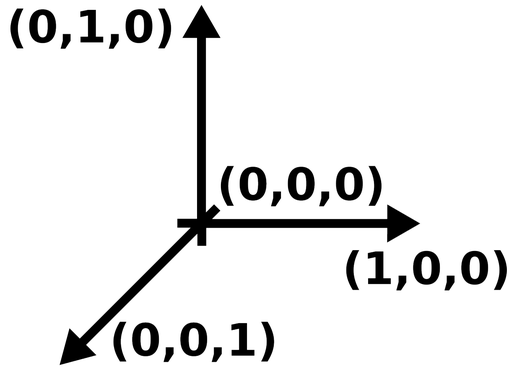


Euclidean spaces on a computer

- Given an origin
  - $o = (0, 0, \ldots, 0) \in \mathbb{R}^n$
- And an orthonormal basis
  - $v_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_2 = (0, 1, 0, \ldots, 0) \in \mathbb{R}^n$
  - $v_3 = (0, 0, 1, \ldots, 0) \in \mathbb{R}^n$
  - $\ldots$
  - $v_n = (0, 0, 0, \ldots, 1) \in \mathbb{R}^n$
- Points can be identified uniquely in that space

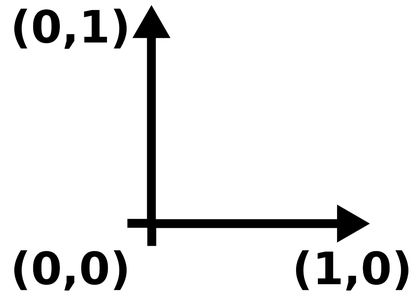


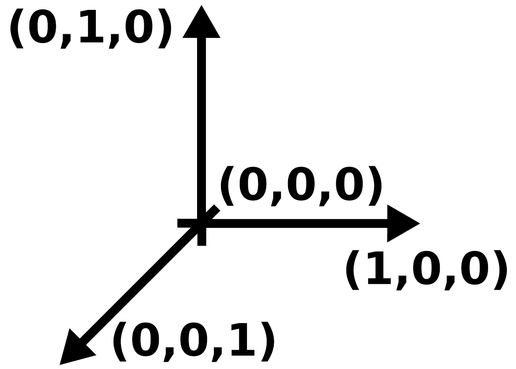




Euclidean spaces on a computer

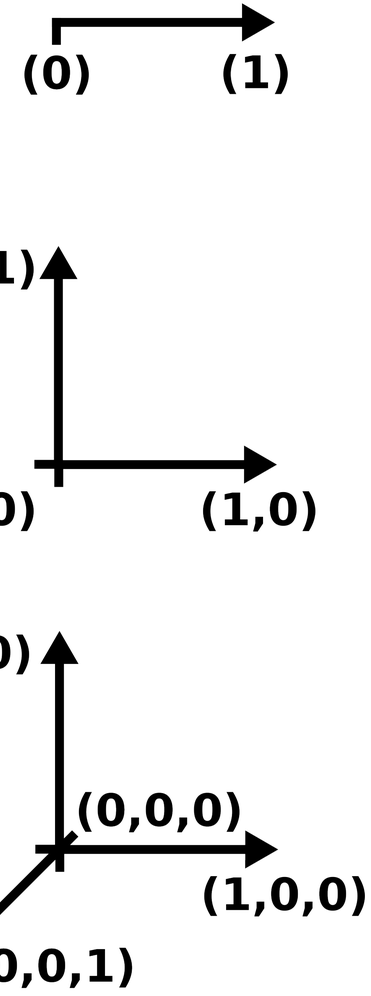
- Direct product of closed unit intervals





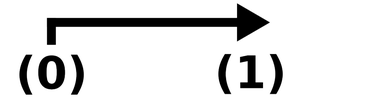
Euclidean spaces on a computer

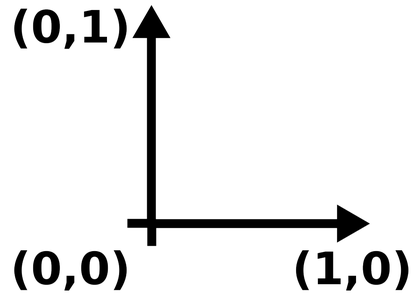
- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$

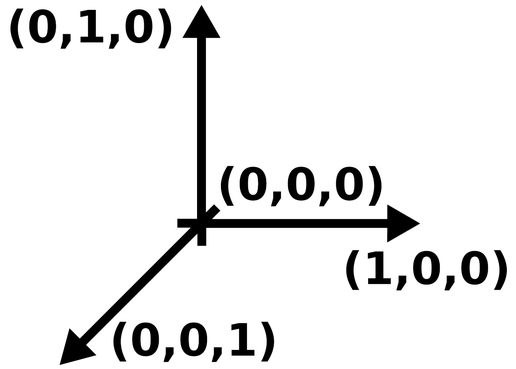


Euclidean spaces on a computer

- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$
  - By construction

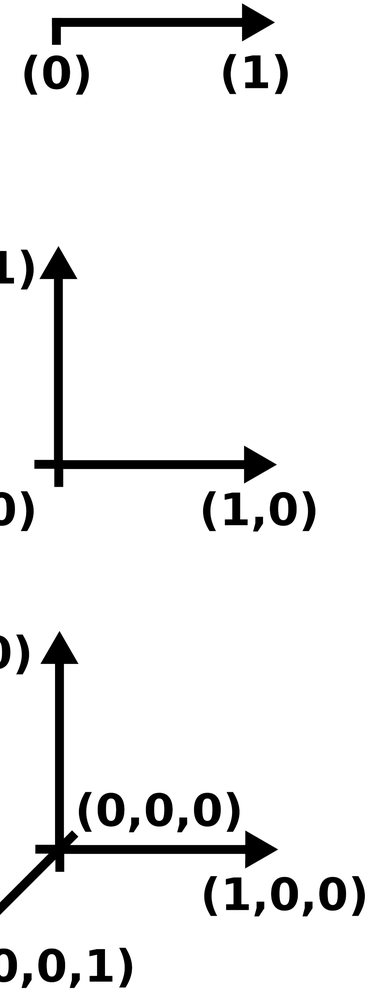






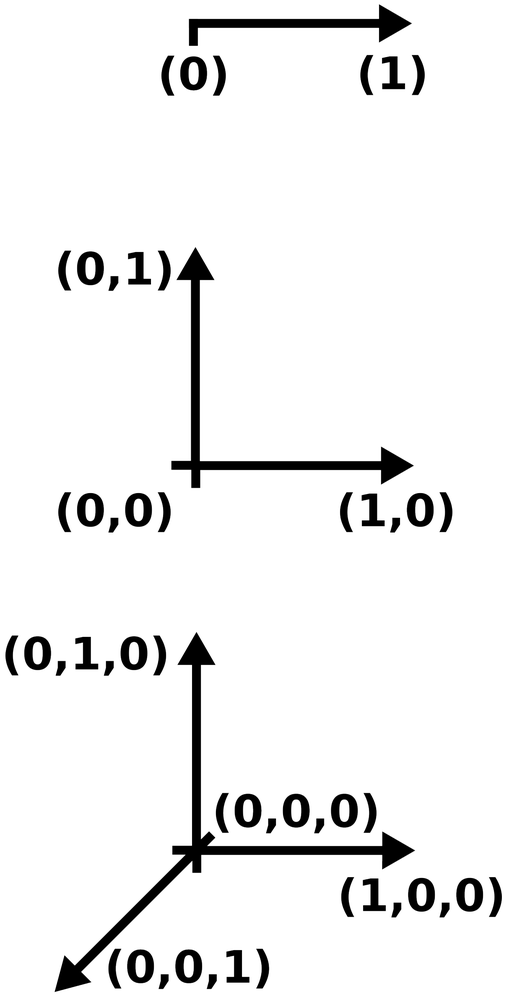
Euclidean spaces on a computer

- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$
  - By construction
    - Unit translations along the vectors of the orthonormal basis



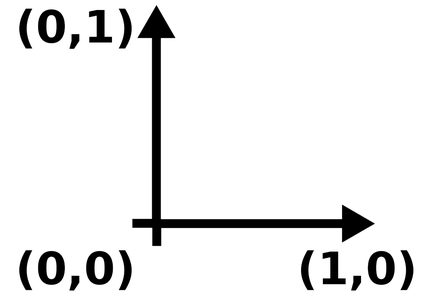
Euclidean spaces on a computer

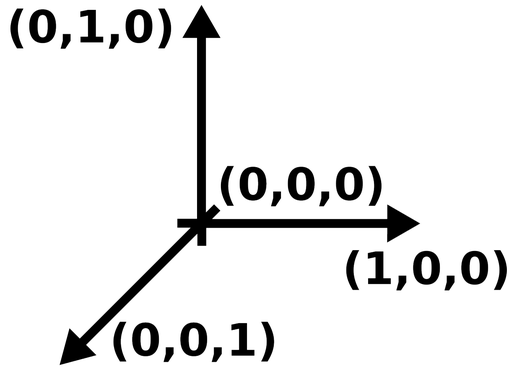
- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$
  - By construction
    - Unit translations along the vectors of the orthonormal basis
    - Covering a bounded, compact region of $\mathbb{R}^n$



Euclidean spaces on a computer

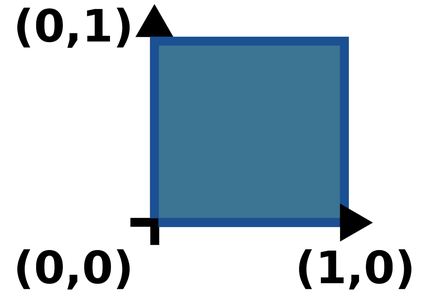
- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$
  - By construction
    - Unit translations along the vectors of the orthonormal basis
    - Covering a bounded, compact region of $\mathbb{R}^n$

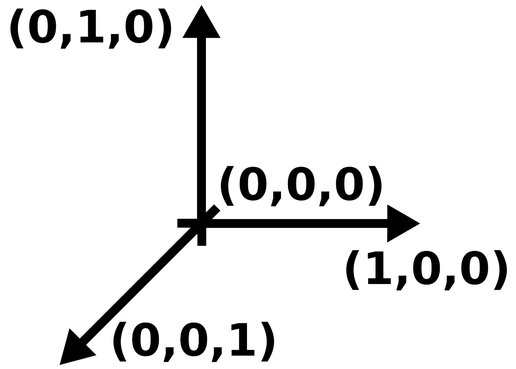




Euclidean spaces on a computer

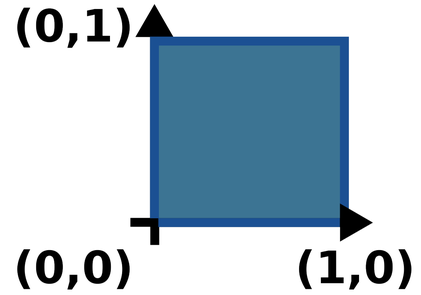
- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$
  - By construction
    - Unit translations along the vectors of the orthonormal basis
    - Covering a bounded, compact region of $\mathbb{R}^n$

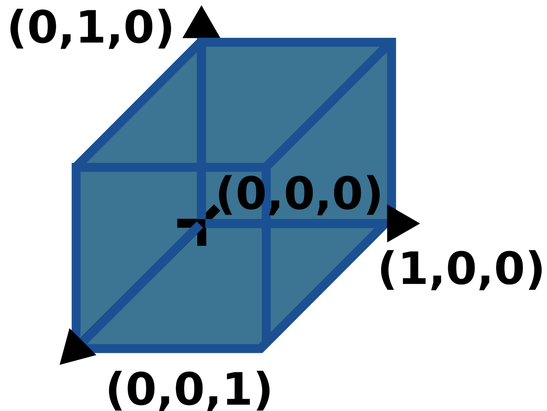




Euclidean spaces on a computer

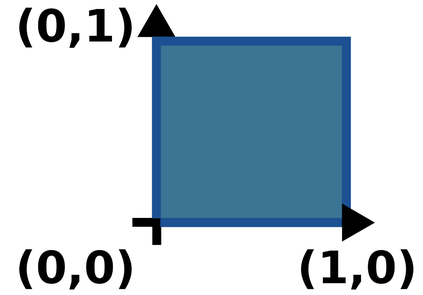
- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$
  - By construction
    - Unit translations along the vectors of the orthonormal basis
    - Covering a bounded, compact region of $\mathbb{R}^n$

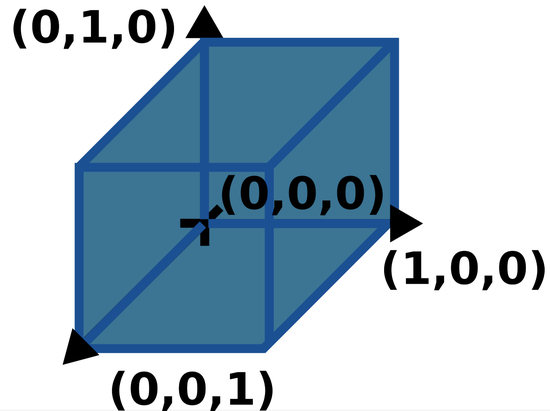




Euclidean spaces on a computer

- Direct product of closed unit intervals
  - $[0,1] \times [0,1] \times \cdots \times [0,1]$
  - By construction
    - Unit translations along the vectors of the orthonormal basis
    - Covering a bounded, compact region of $\mathbb{R}^n$

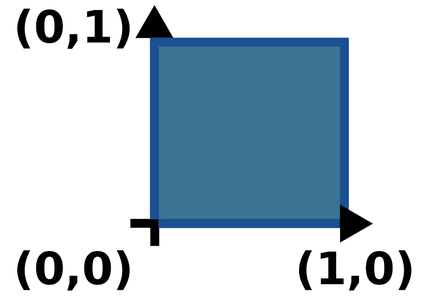


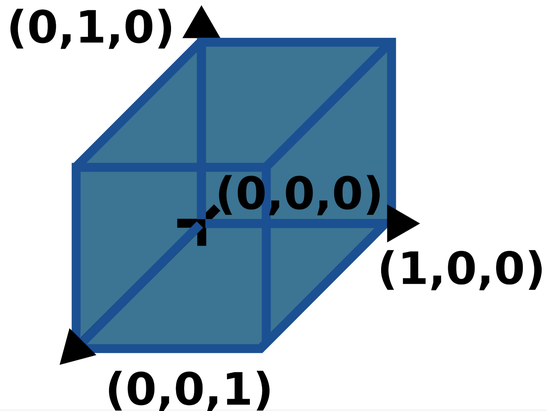


- **Unit cell**

Euclidean spaces on a computer

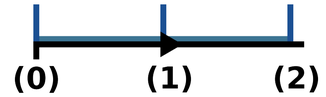
- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$

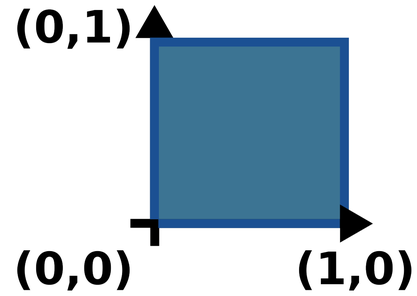


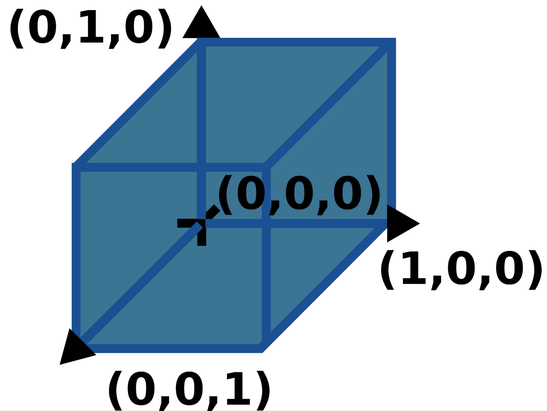


Euclidean spaces on a computer

- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$

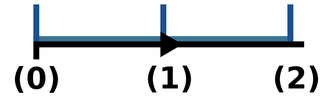


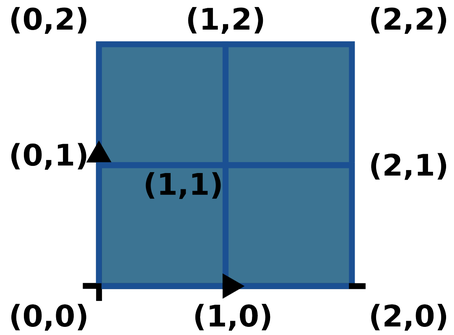


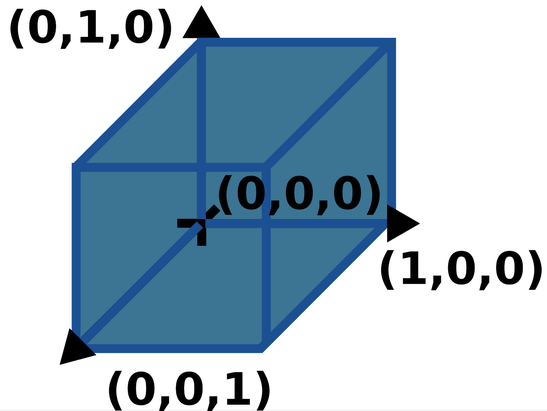


Euclidean spaces on a computer

- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$

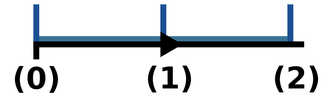


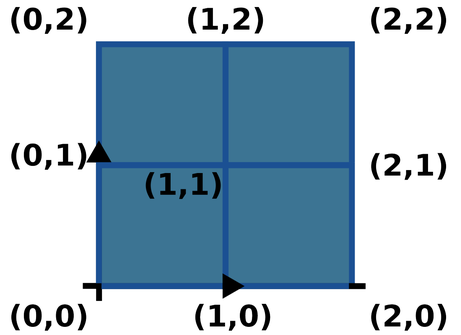


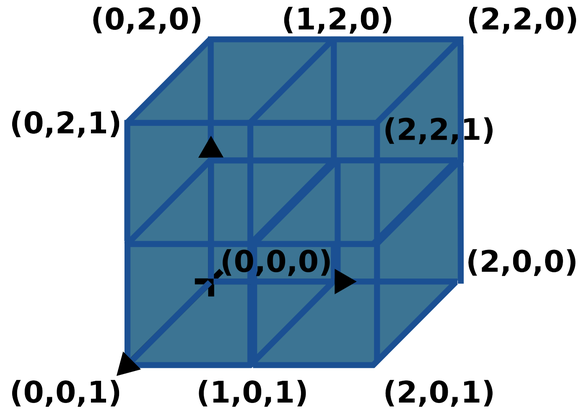


Euclidean spaces on a computer

- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$

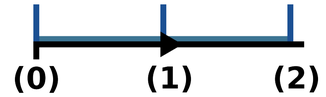


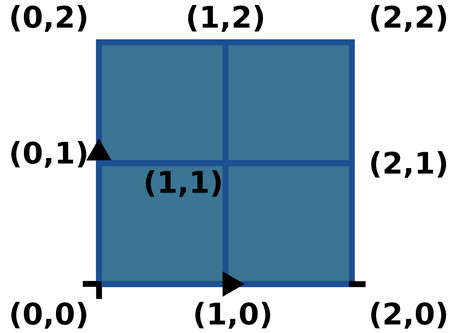


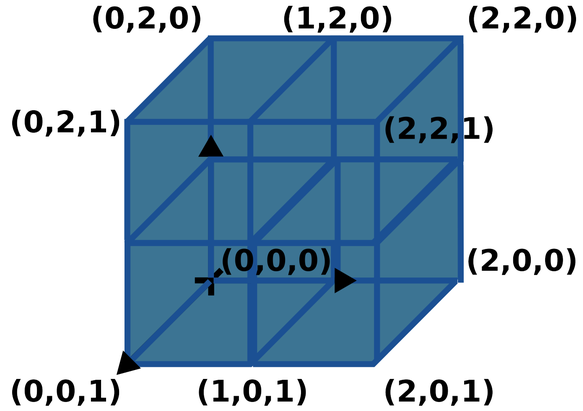


Euclidean spaces on a computer

- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$
  - How many unit cells in $[0, k_1] \times [0, k_2] \times [0, k_3]$?

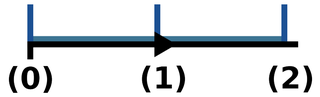


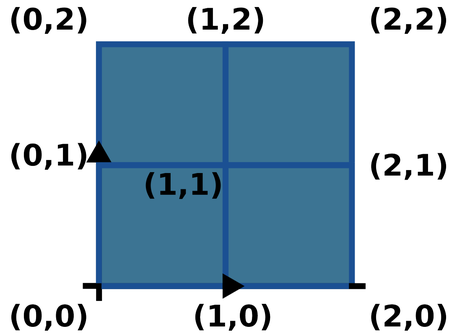


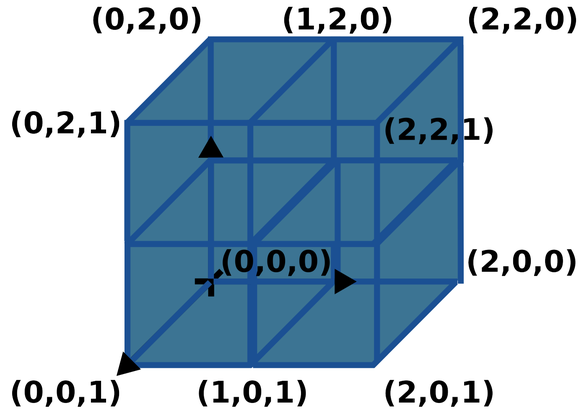


Euclidean spaces on a computer

- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$
  - How many unit cells in $[0, k_1] \times [0, k_2] \times [0, k_3]$?
    - $k_1.k_2.k_3$

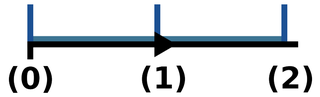


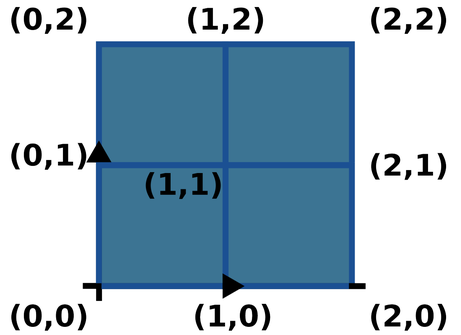


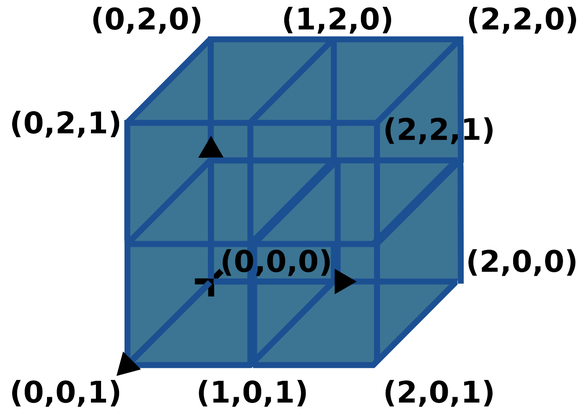


Euclidean spaces on a computer

- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$
  - How many unit cells in $[0, k_1] \times [0, k_2] \times [0, k_3]$?
    - $k_1.k_2.k_3$
  - How many samples?

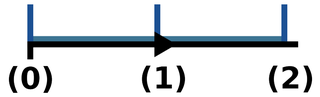


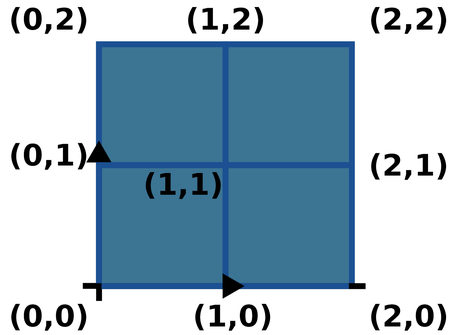


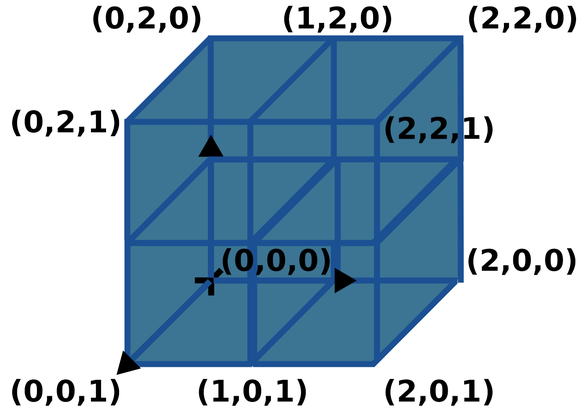


Euclidean spaces on a computer

- Notion of **regular grid** $\mathcal{G}$
  - Finite collection of unit cells
  - Entirely covering the direct product of closed intervals, such that
    - $[0, k_1] \times [0, k_2] \times \cdots \times [0, k_n]$
    - $k_i \in \mathbb{N}$
  - How many unit cells in $[0, k_1] \times [0, k_2] \times [0, k_3]$?
    - $k_1.k_2.k_3$
  - How many samples? $(k_1 + 1).(k_2 + 1).(k_3 + 1)$





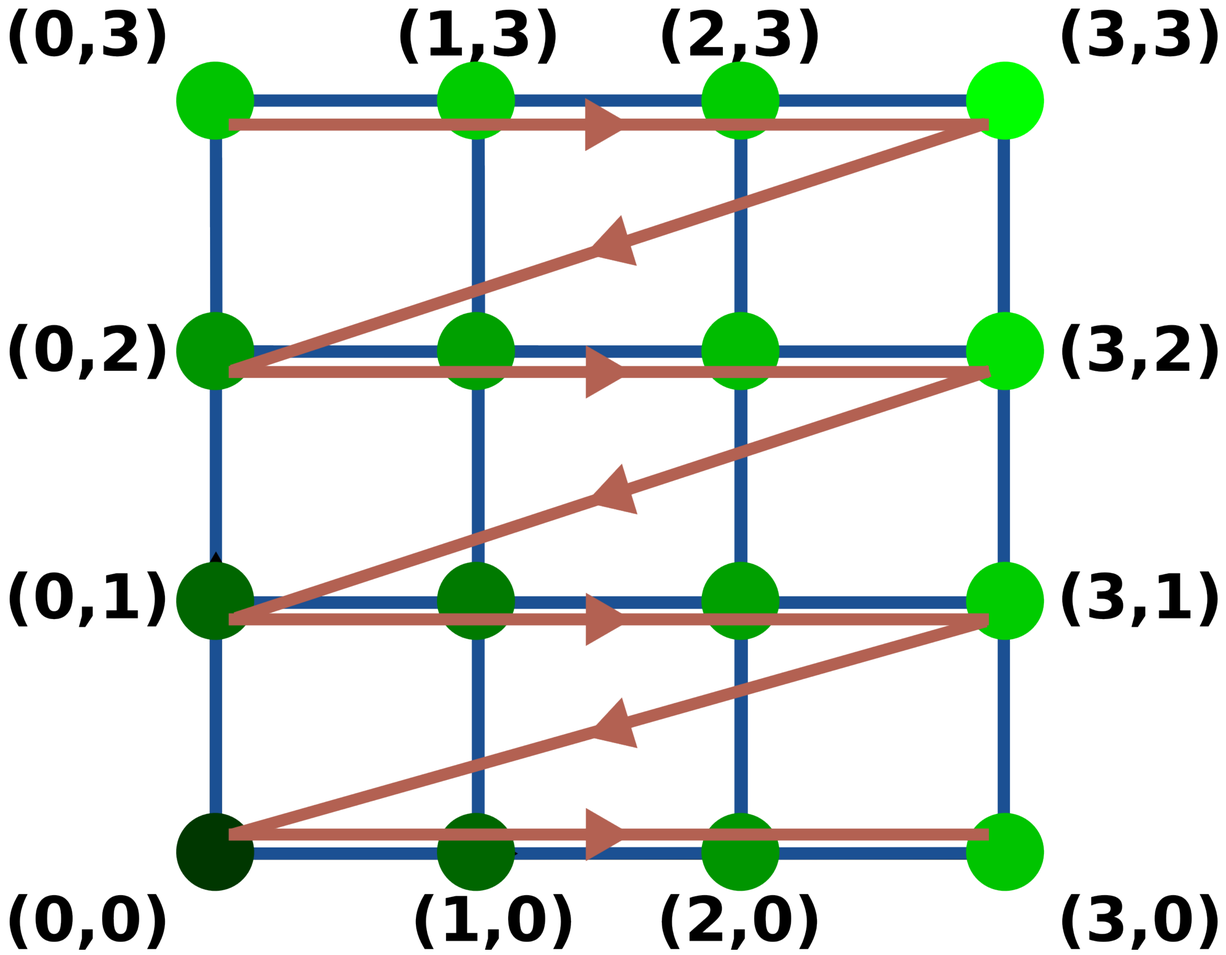


Regular grids

- 1D regular grids
  - Arrays

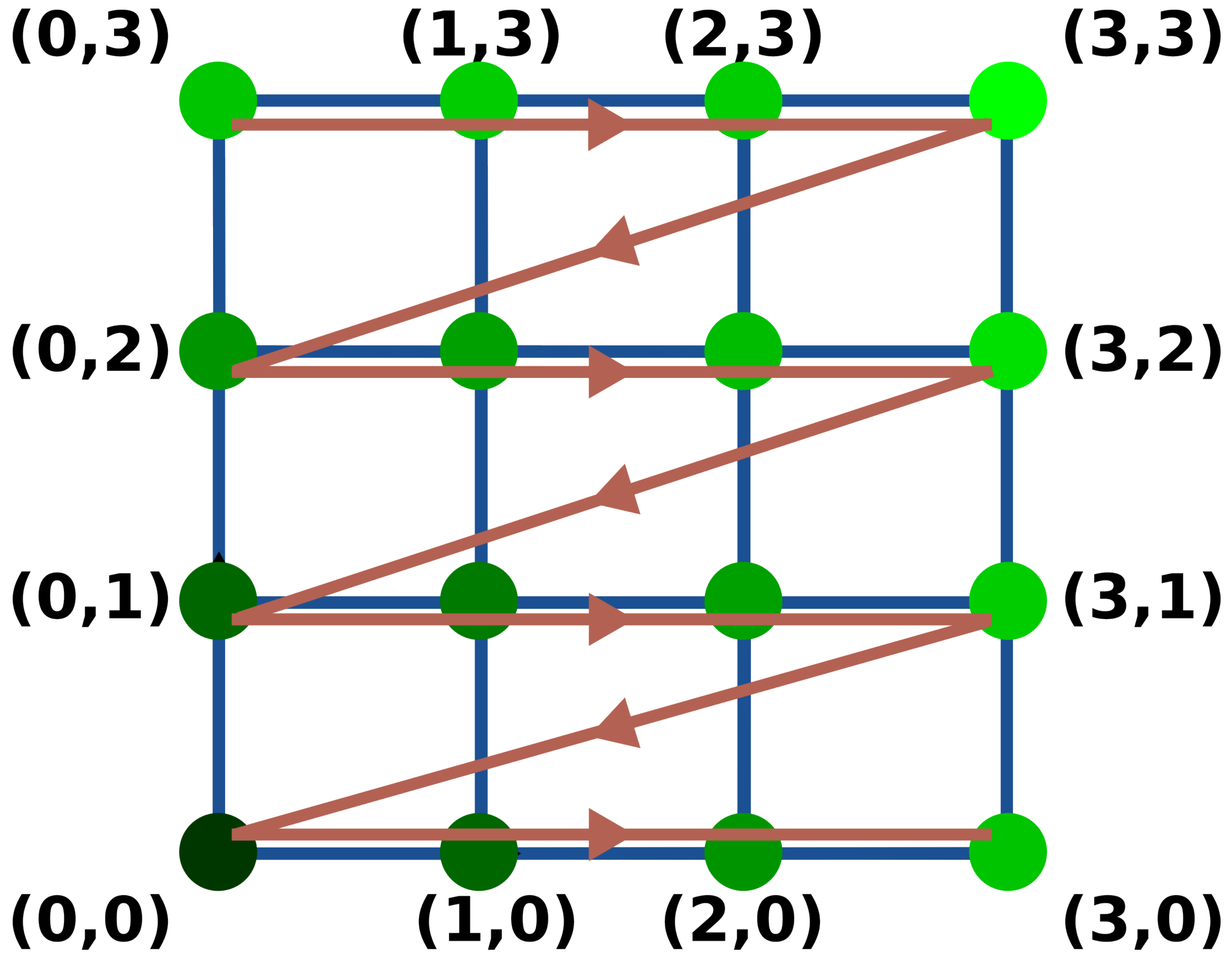
Regular grids

- 1D regular grids
  - Arrays
- 2D regular grids
  - Unit cell: pixel
  - Collection of arrays



Regular grids

- 1D regular grids
  - Arrays
- 2D regular grids
  - Unit cell: pixel
  - Collection of arrays
- 3D regular grids
  - Unit cell: voxel
  - Collection of arrays



Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$

Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$

Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples

Interpolants for regular grids

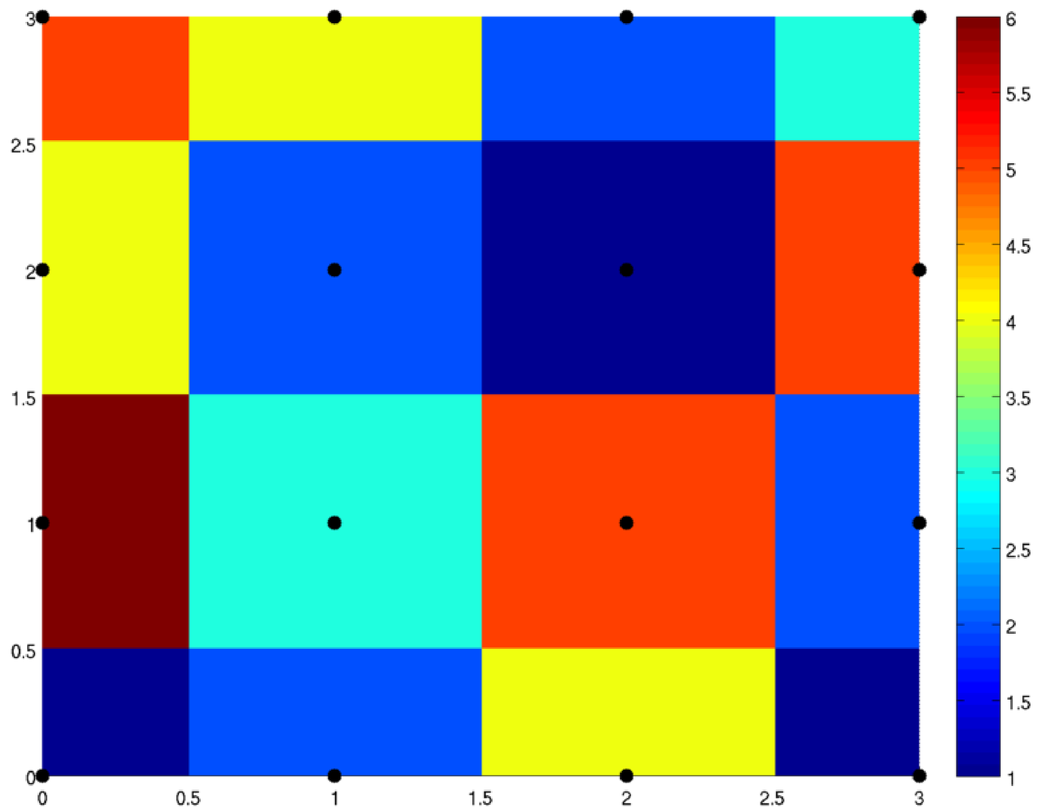
- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Piecewise constant
    - Nearest neighbor

Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Piecewise constant
    - Nearest neighbor
      - Value of the nearest vertex

Interpolants for regular grids

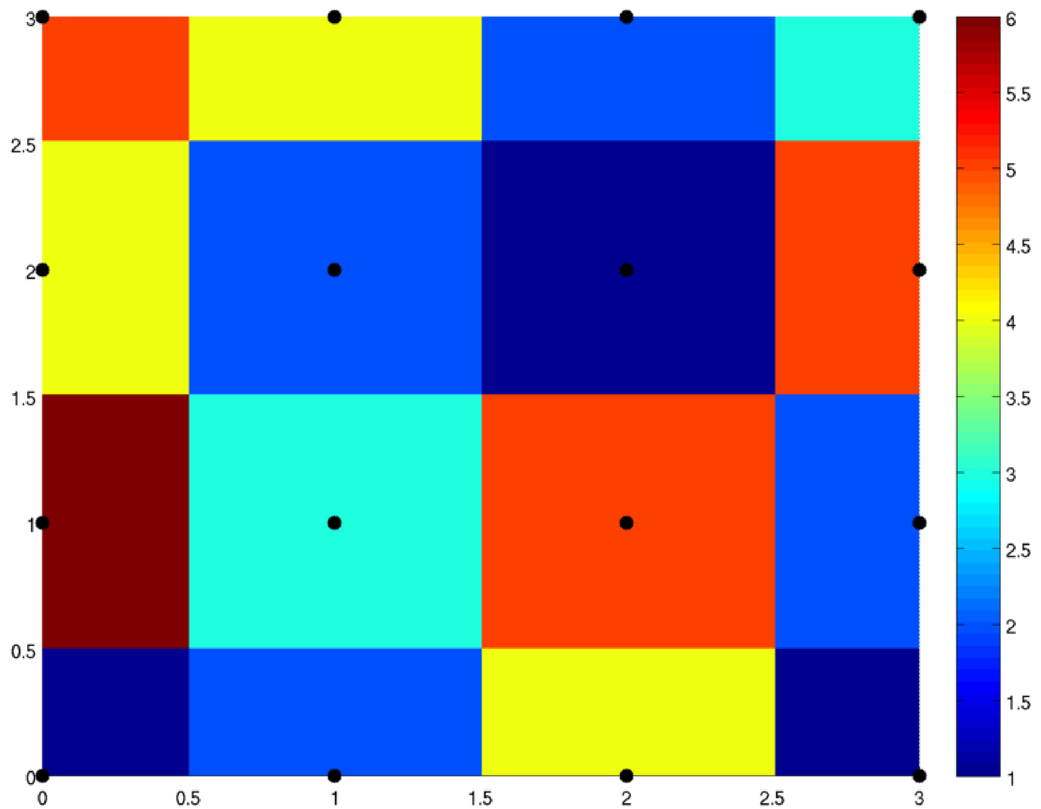
- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Piecewise constant
    - Nearest neighbor
      - Value of the nearest vertex



[Wikipedia]

Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      - One dimension at a time

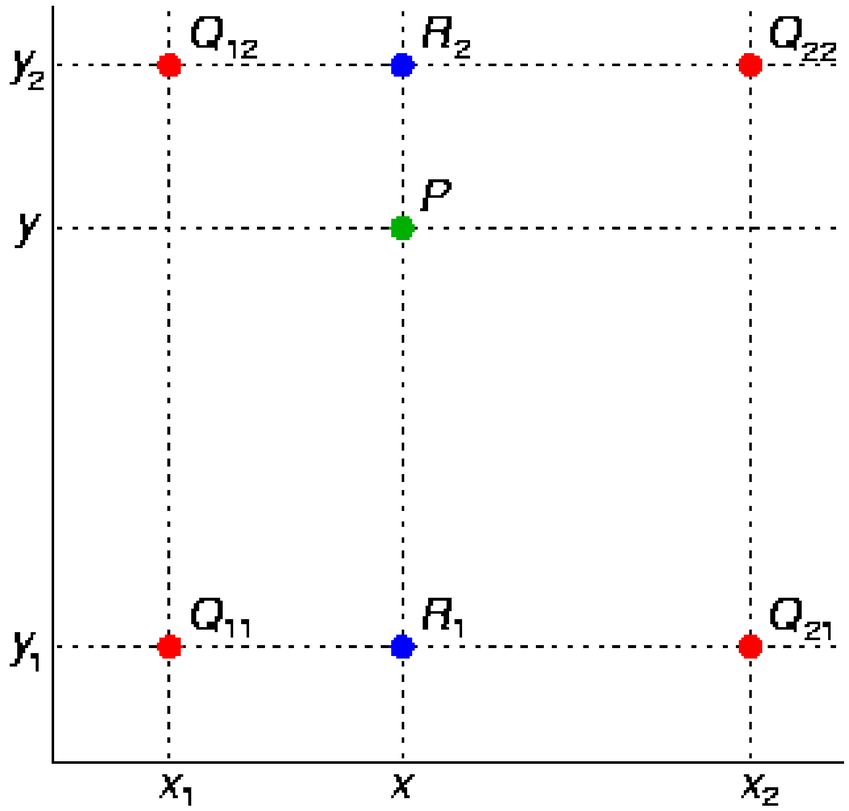


[Wikipedia]

Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      - One dimension at a time

$$f_{\mathbb{I}}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)$$



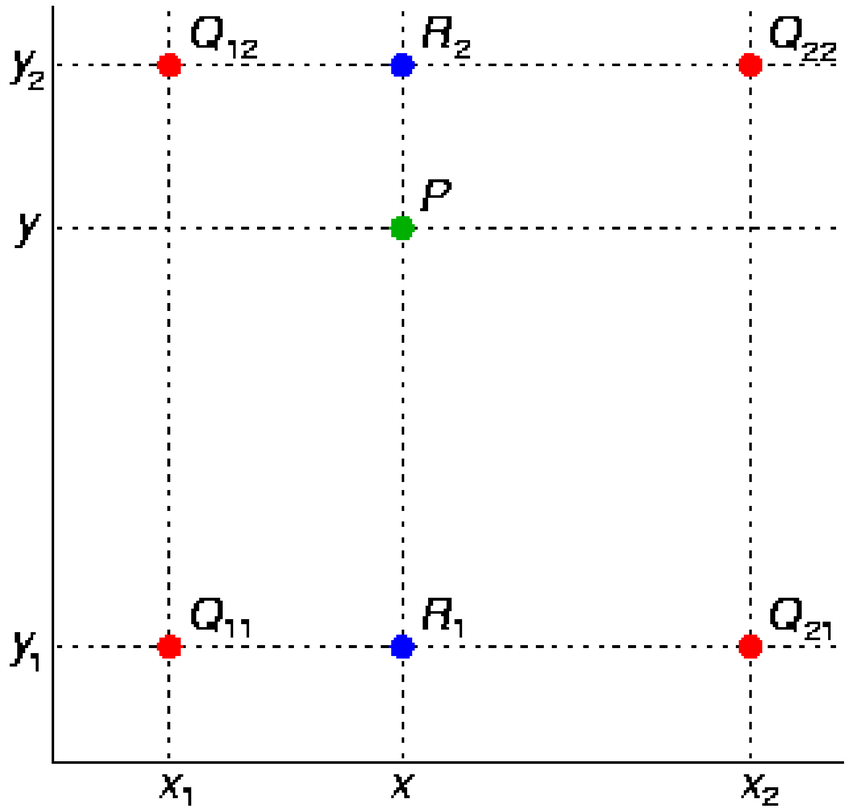
[Wikipedia]

Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      - One dimension at a time

$$f_{\mathbb{I}}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$



[Wikipedia]

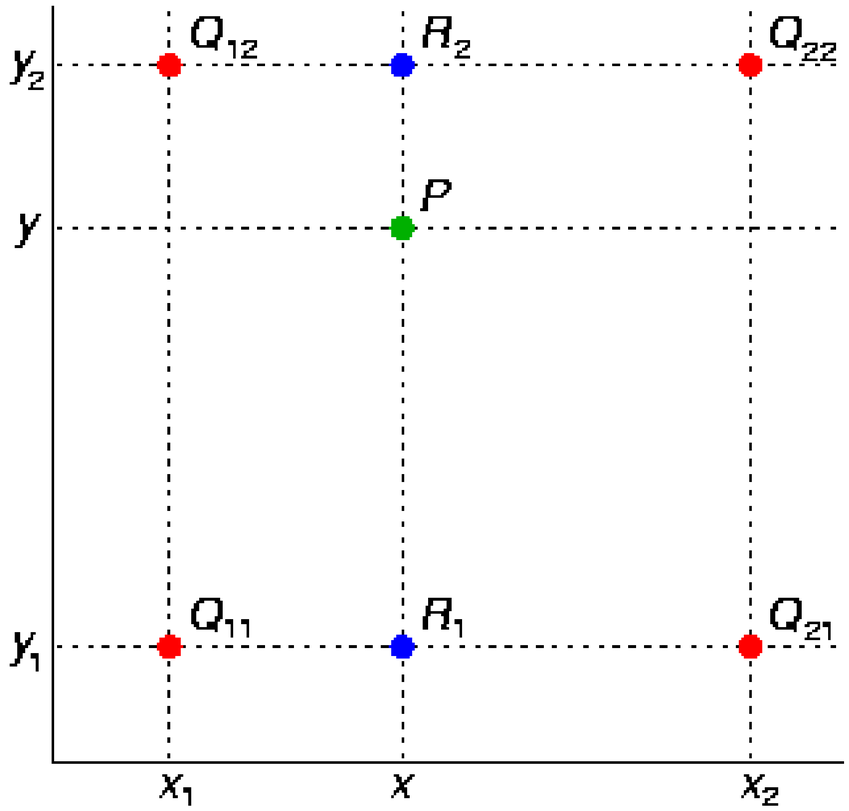
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      - One dimension at a time

$$f_{\mathbb{I}}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

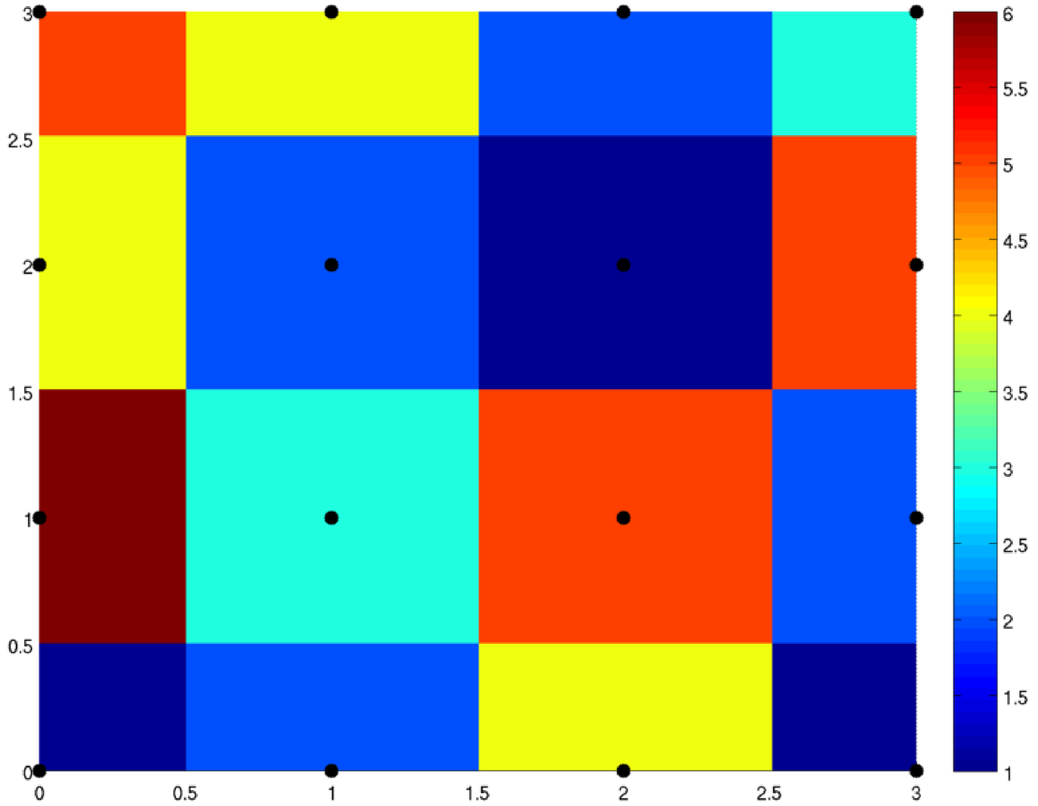
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      - One dimension at a time

$$f_{\mathbb{I}}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

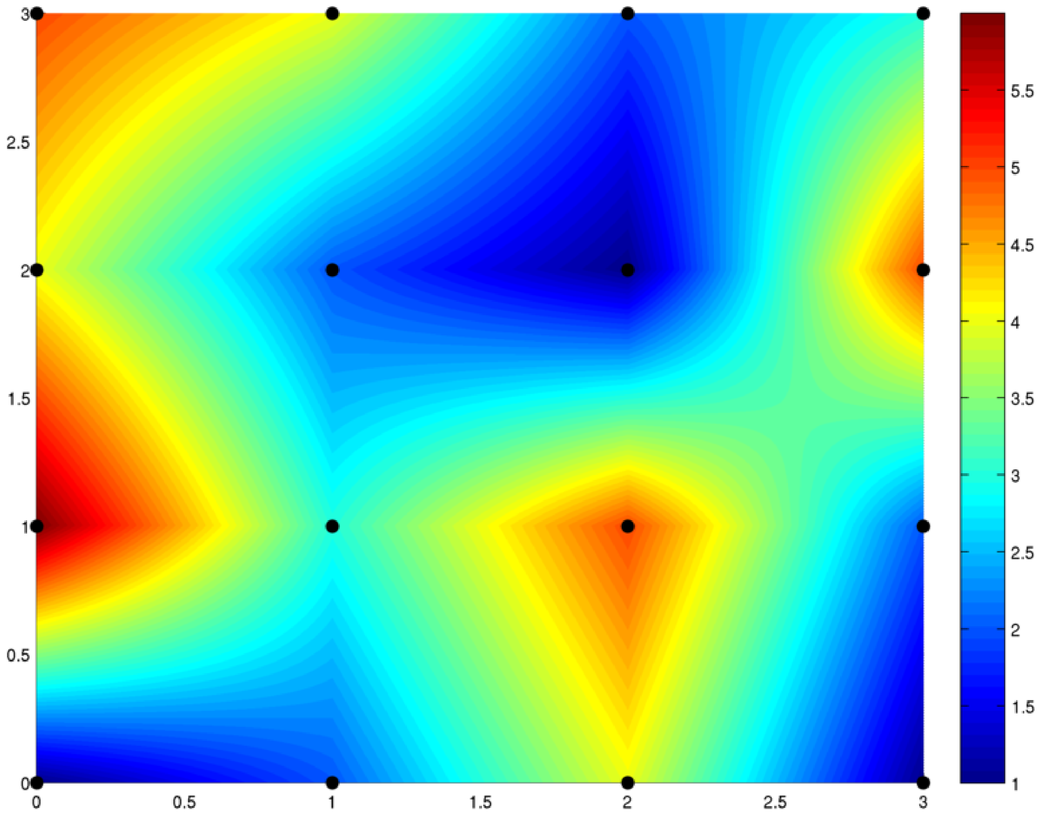
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      - One dimension at a time

$$f_{\mathbb{I}}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

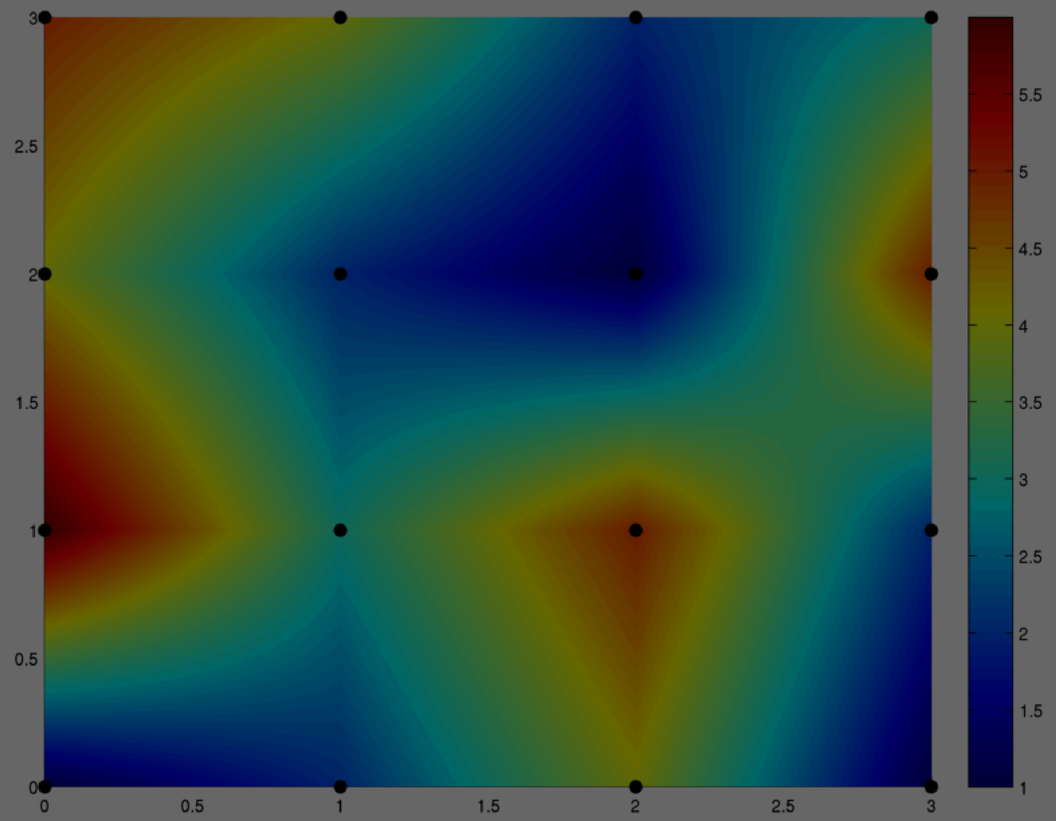
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      – One dimension at a time

$$f_{\mathbb{I}}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

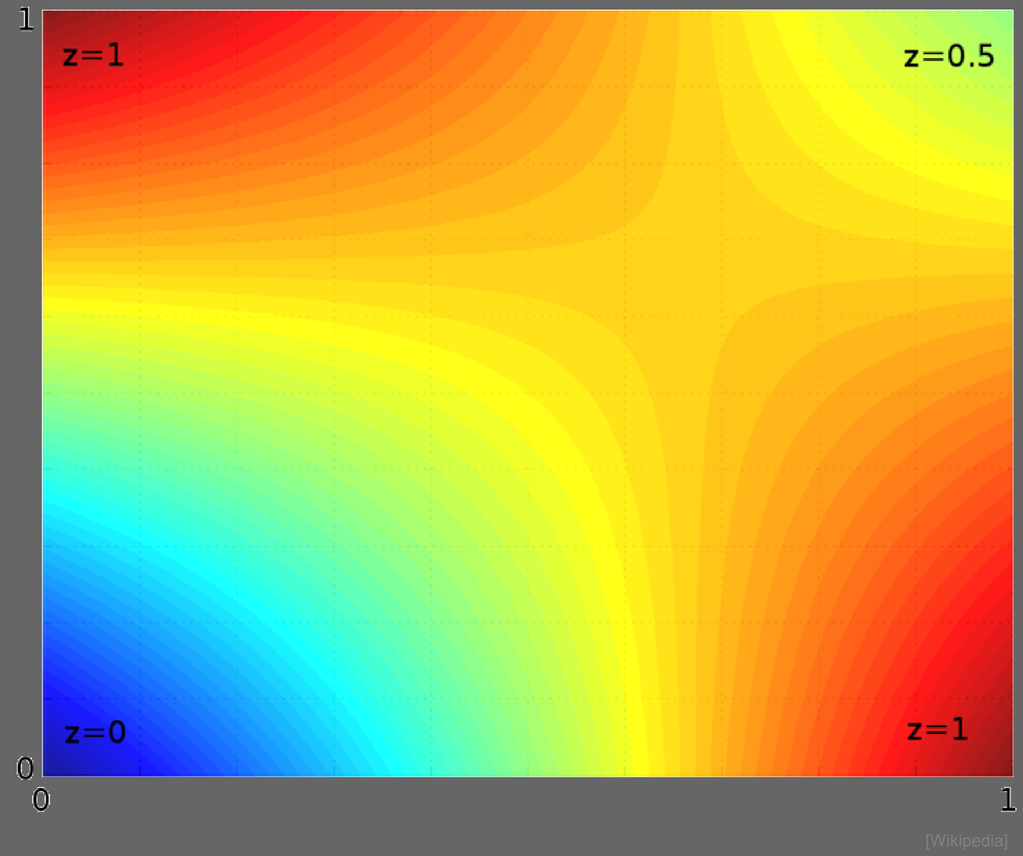
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bilinear interpolation
      – One dimension at a time

$$f_{\mathbb{I}}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

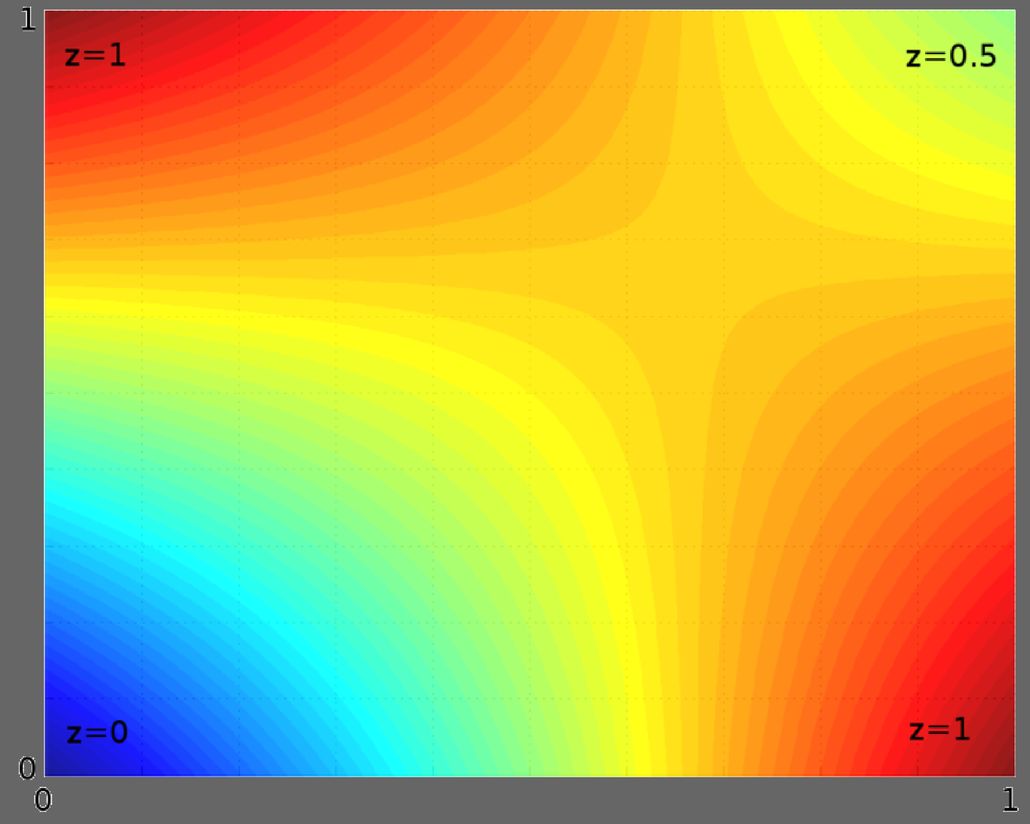
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - Problem

$$\frac{x_2 - x_1}{} (x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

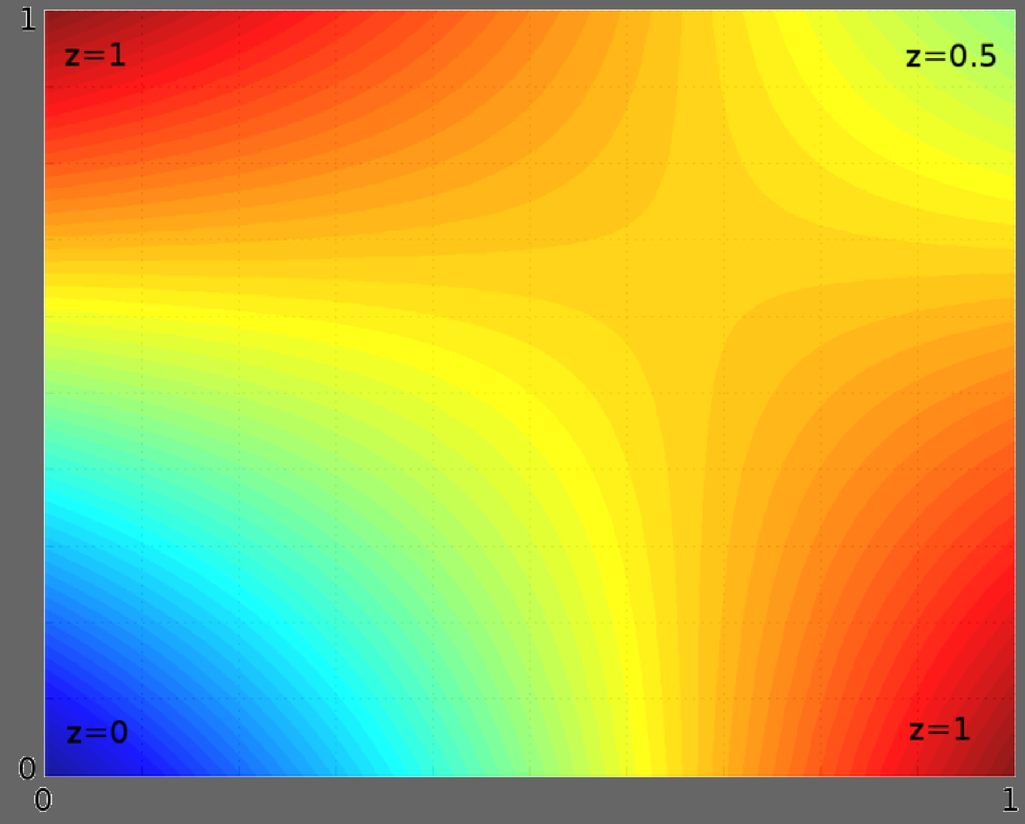
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - Problem
    - Critical points in the interior

$$x_2 - x_1 \qquad (x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

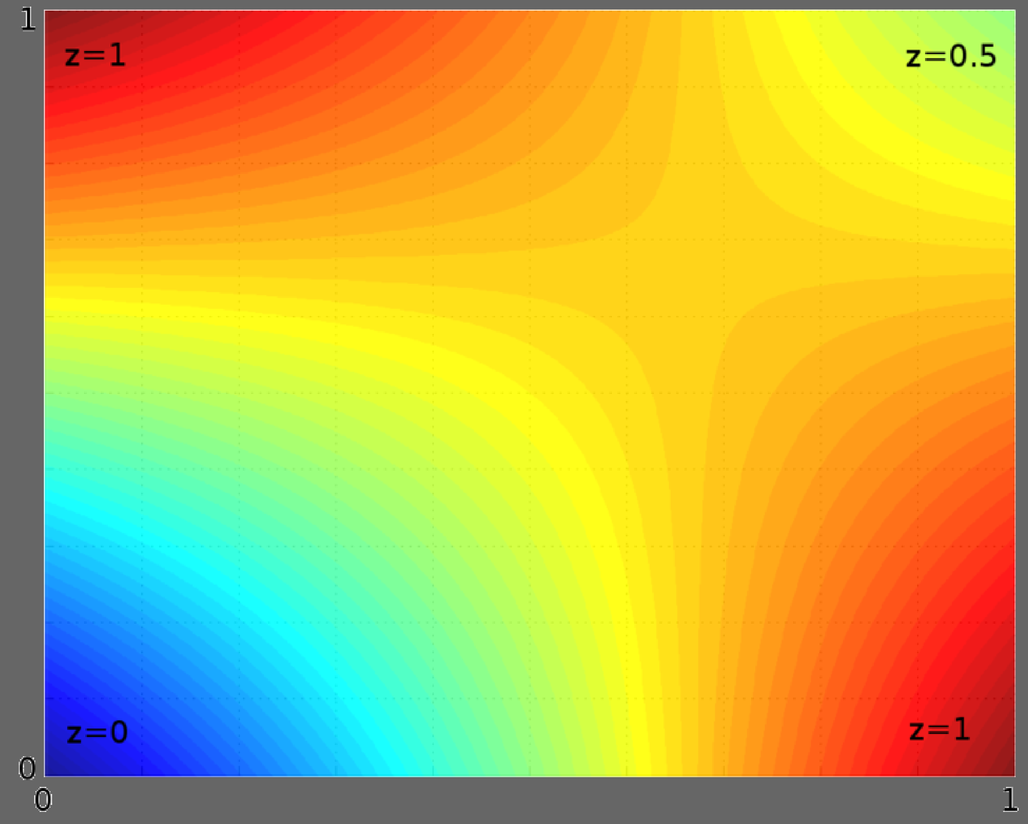
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - Problem
    - Critical points in the interior
    - Source of many ambiguities in visualization

$$\frac{x_2 - x_1}{} (x_1, y_1)$$

$$f_{\mathbb{I}}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)$$

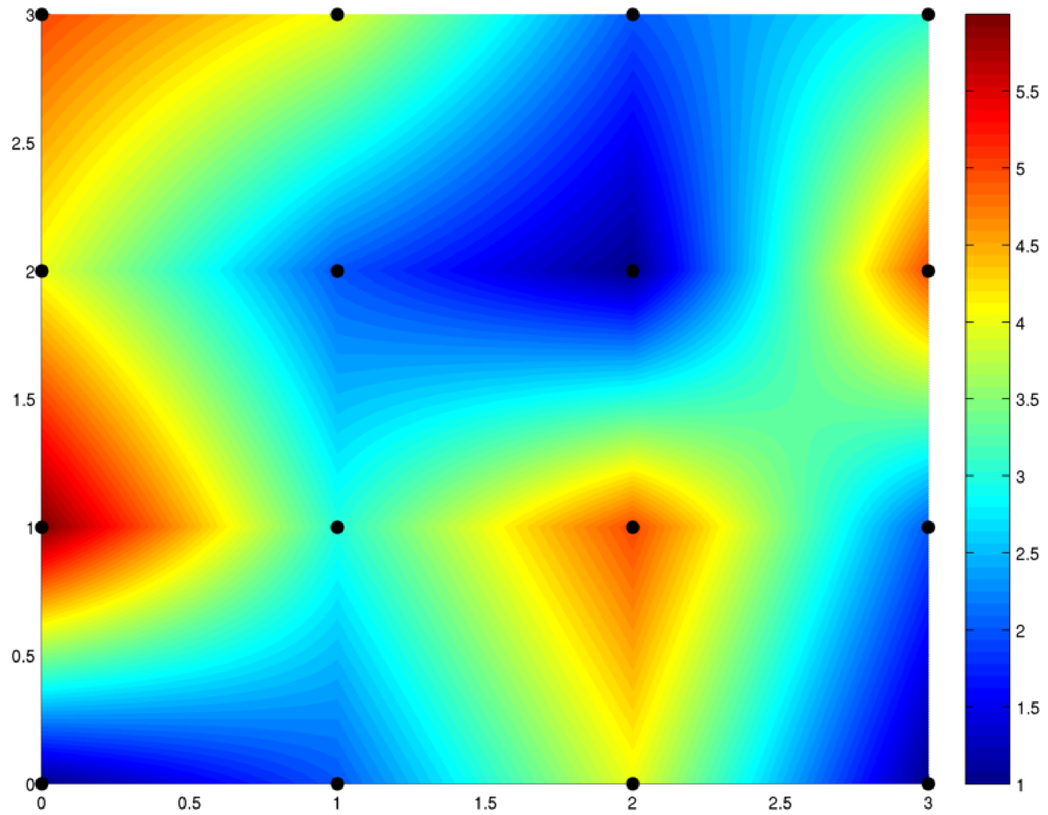
$$f_{\mathbb{I}}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)$$



[Wikipedia]

Interpolants for regular grids

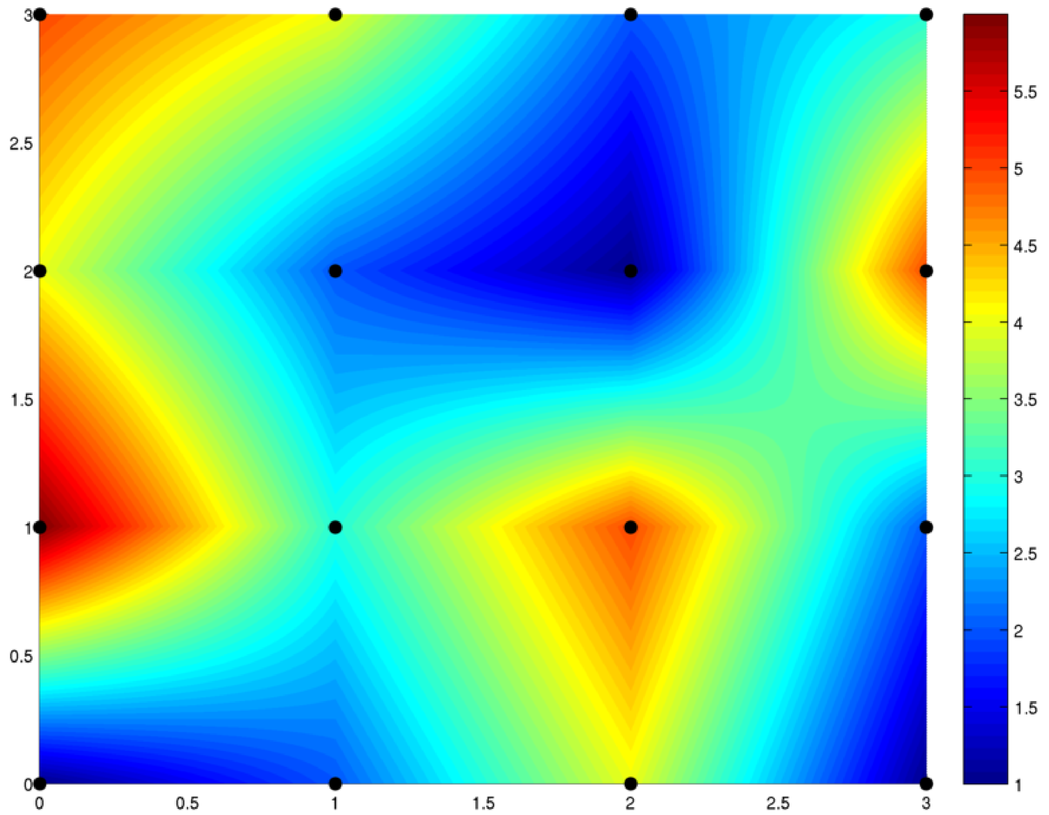
- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bicubic interpolation



[Wikipedia]

Interpolants for regular grids

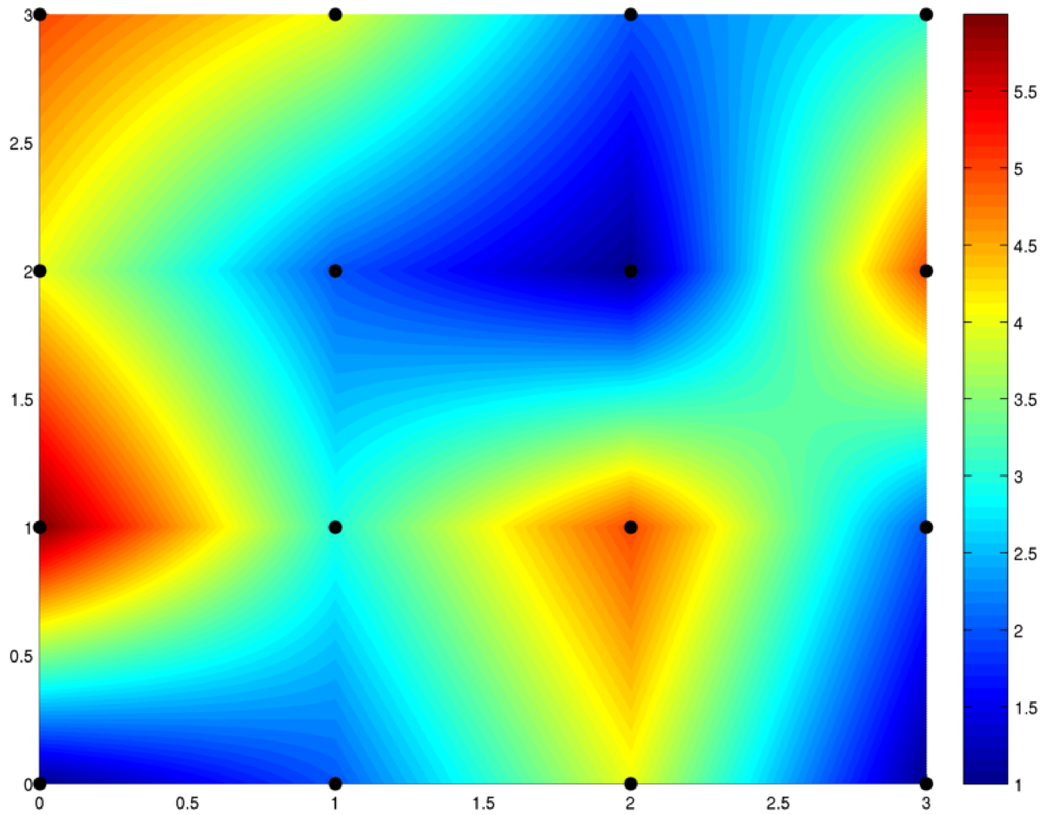
- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bicubic interpolation
      - – Piecewise polynomial



[Wikipedia]

Interpolants for regular grids

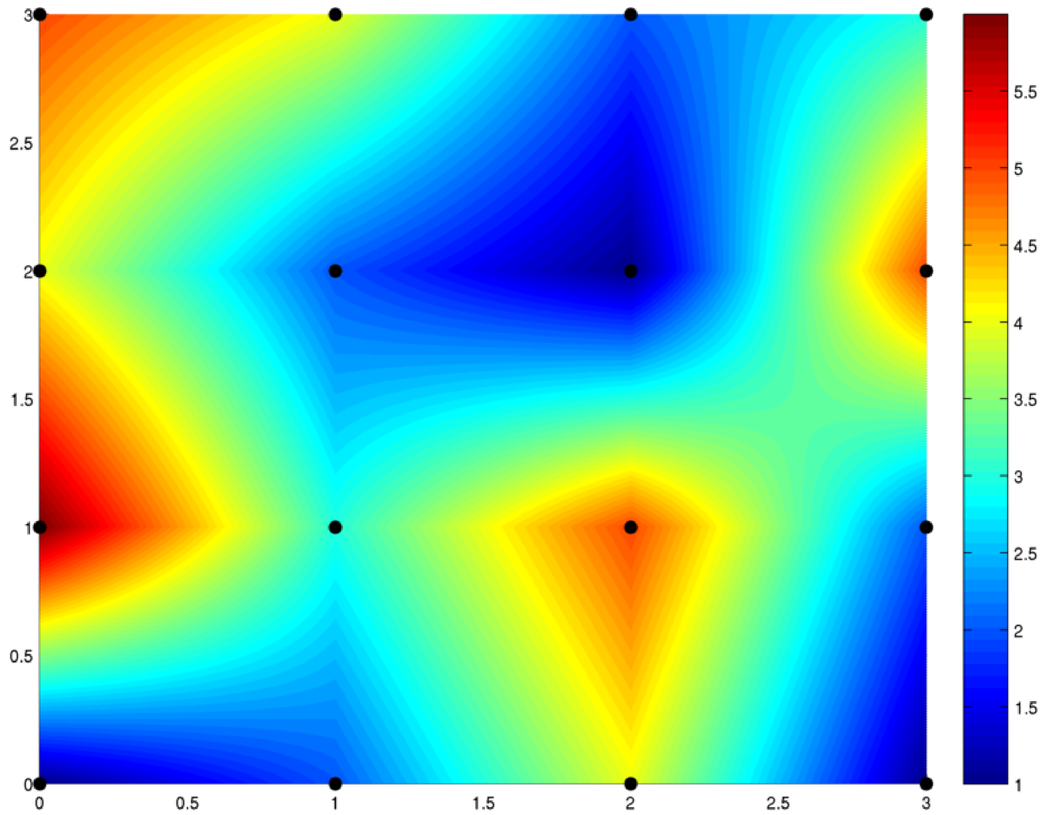
- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bicubic interpolation
      - – Piecewise polynomial
      - – Smoother interpolant



[Wikipedia]

Interpolants for regular grids

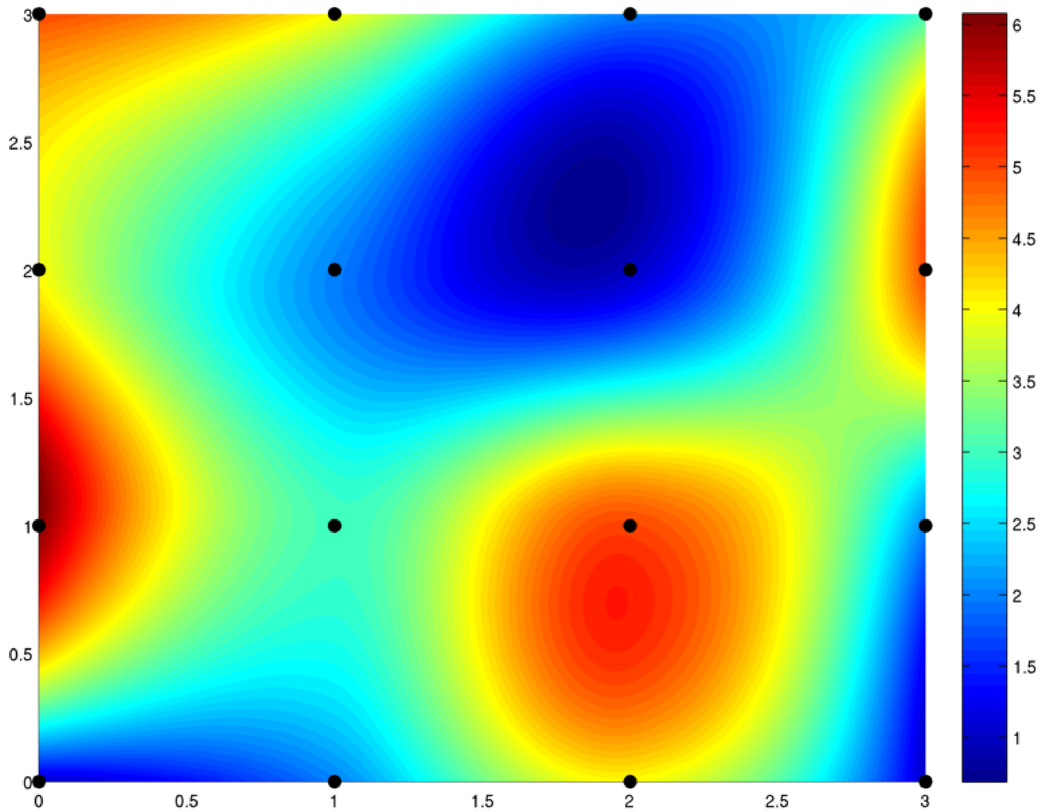
- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bicubic interpolation
      - Piecewise polynomial
      - Smoother interpolant
      - Takes adjacent cells into account



[Wikipedia]

Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Bicubic interpolation
      - Piecewise polynomial
      - Smoother interpolant
      - Takes adjacent cells into account



[Wikipedia]

Interpolants for regular grids

- For regular grids of $\mathbb{R}^3$

Interpolants for regular grids

- For regular grids of $\mathbb{R}^3$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \times \ ]z_1, z_2[\ \rightarrow \mathbb{R}$

Interpolants for regular grids

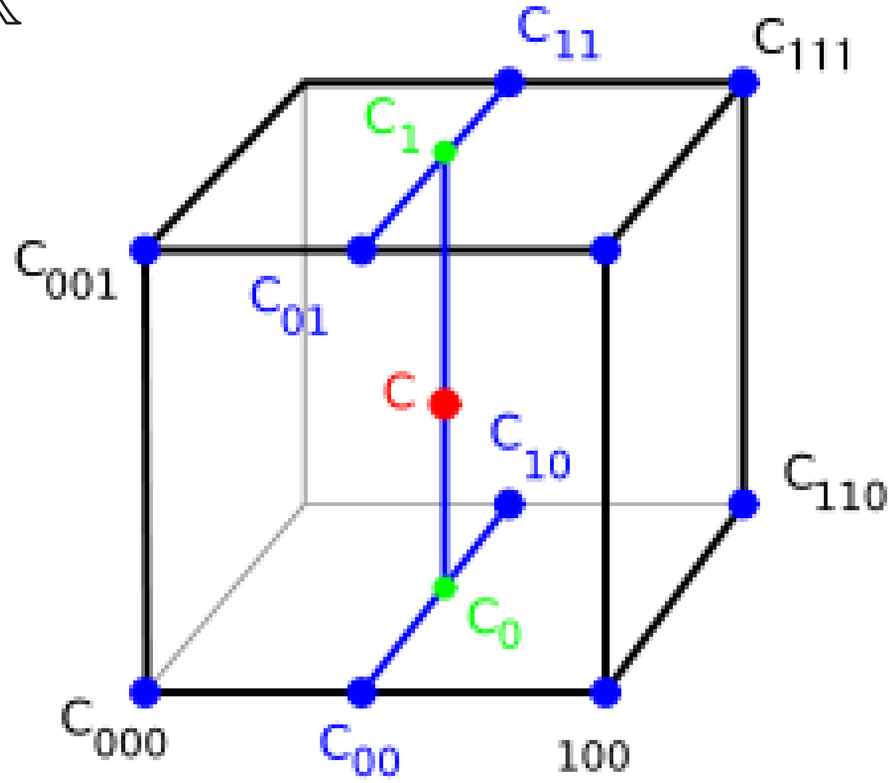
- For regular grids of $\mathbb{R}^3$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \times \ ]z_1, z_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Piecewise constant
      - Nearest neighbor
    - Trilinear
    - Tricubic

Interpolants for regular grids

- For regular grids of $\mathbb{R}^3$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \times \ ]z_1, z_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Piecewise constant
      - Nearest neighbor
    - **Trilinear**
    - Tricubic

Interpolants for regular grids

- For regular grids of $\mathbb{R}^3$
  - $f_{\mathbb{I}} : \ ]x_1, x_2[\ \times \ ]y_1, y_2[\ \times \ ]z_1, z_2[\ \rightarrow \mathbb{R}$
  - Examples
    - Piecewise constant
      - Nearest neighbor
    - **Trilinear**
    - Tricubic

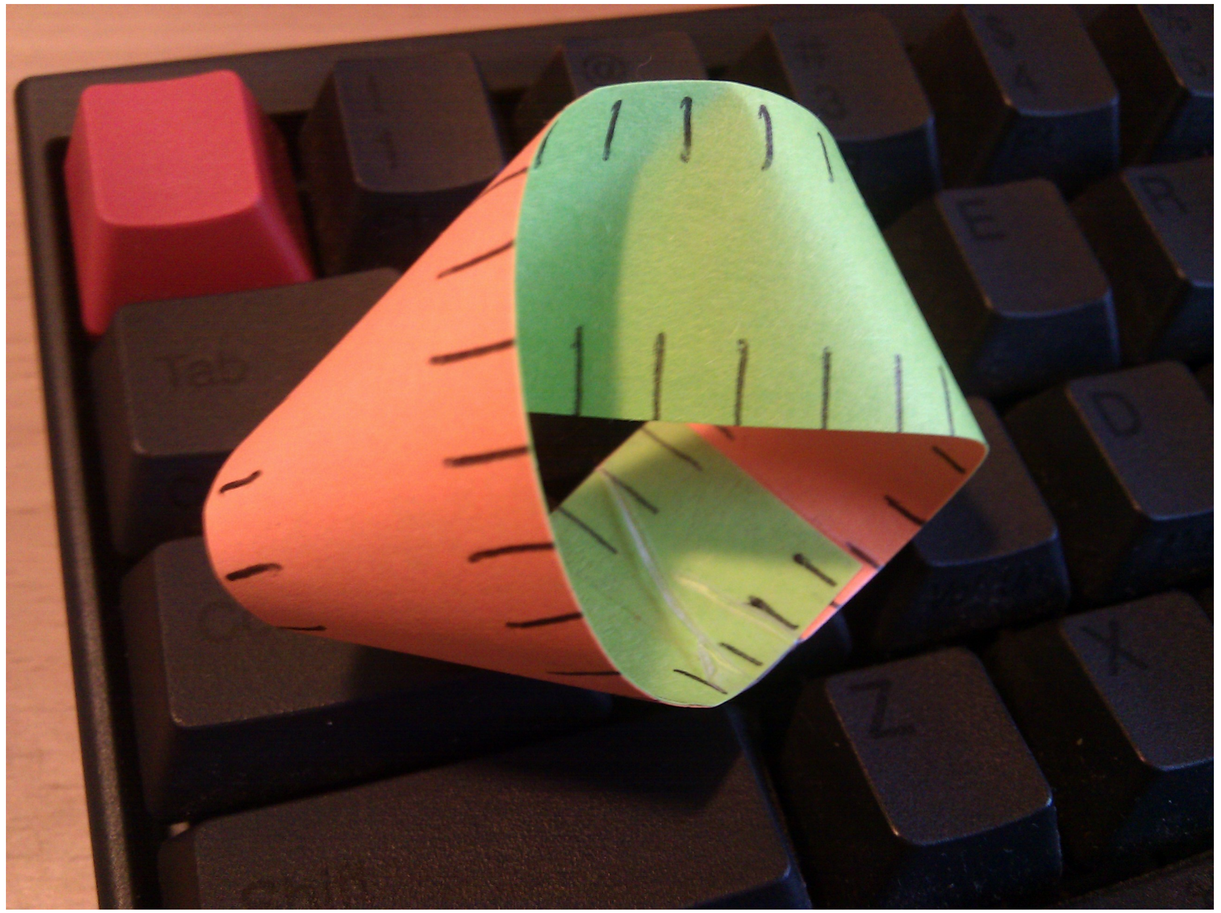


[Wikipedia]

Manifolds on a computer

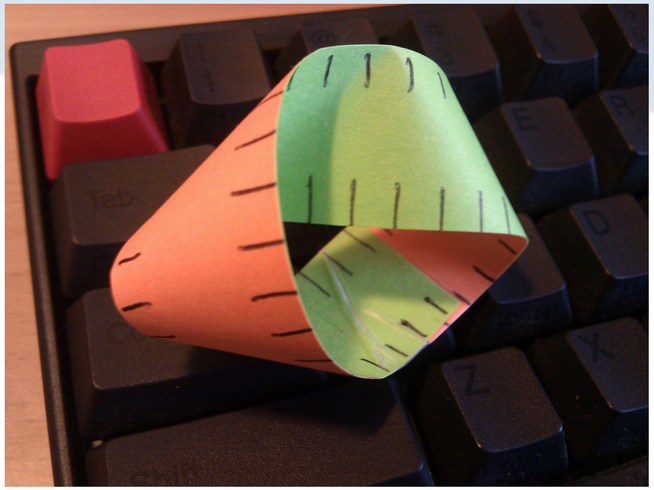
Manifolds on a computer

Manifolds on a computer



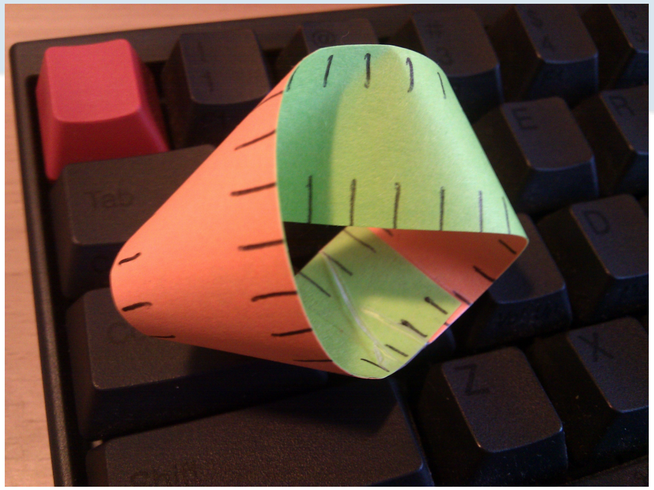
Manifolds on a computer

- Notion of simplex

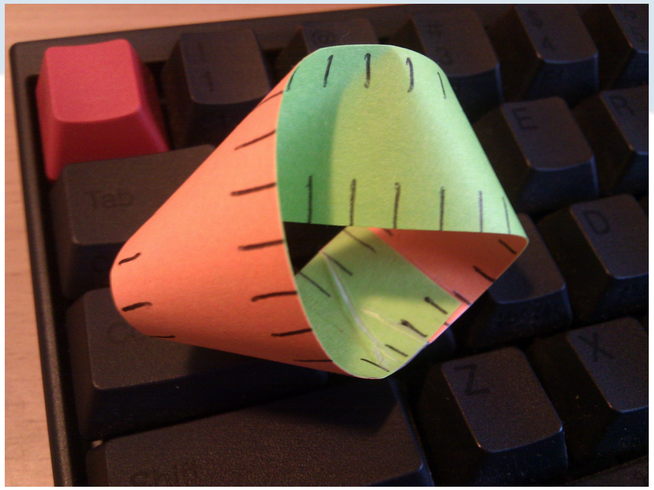


Manifolds on a computer

- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$

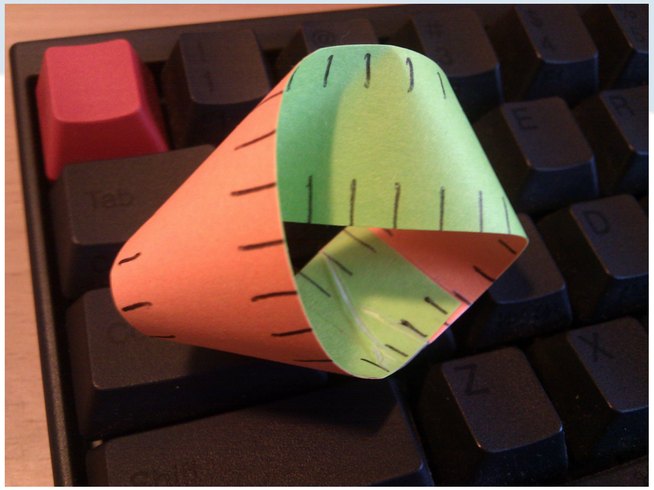


Manifolds on a computer



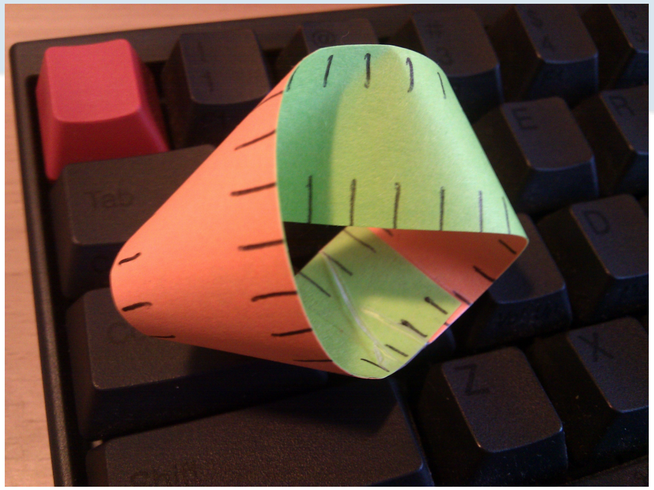
- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$

Manifolds on a computer



- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$
  - d is the dimension of σ

Manifolds on a computer



- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$
  - d is the dimension of σ
  - Smallest combinatorial construction to represent a cell of dimension d

Manifolds on a computer

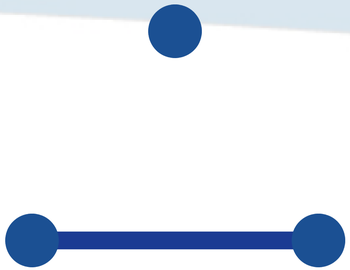
- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$

Manifolds on a computer

- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$
  - 0-simplex: *vertex*

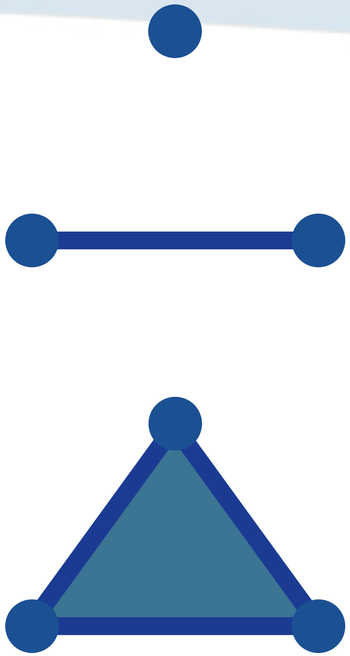
Manifolds on a computer

- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$
  - 0-simplex: *vertex*
  - 1-simplex: *edge*



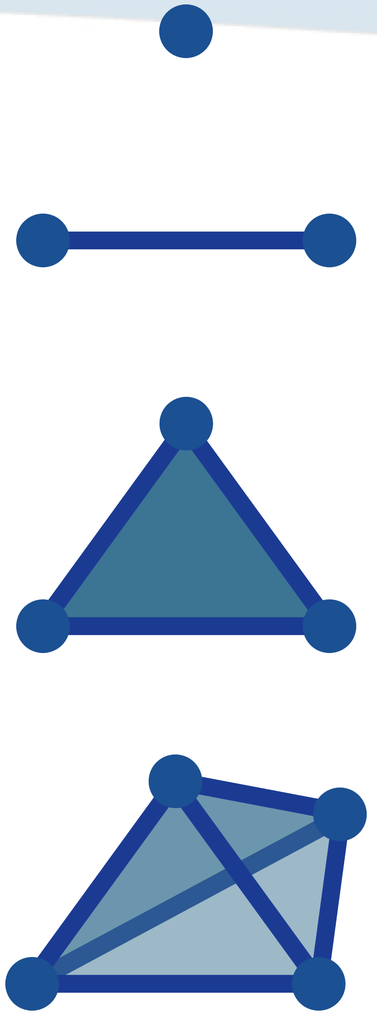
Manifolds on a computer

- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$
  - 0-simplex: *vertex*
  - 1-simplex: *edge*
  - 2-simplex: *triangle*



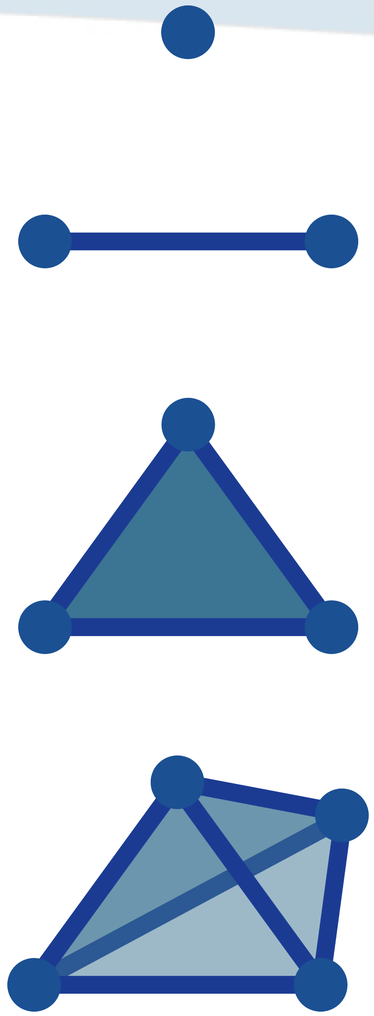
Manifolds on a computer

- Notion of simplex
  - A d-simplex σ is the convex hull of (d + 1) affinely independent points of an euclidean space $\mathbb{R}^n$
  - $0 \leq d \leq n$
  - 0-simplex: *vertex*
  - 1-simplex: *edge*
  - 2-simplex: *triangle*
  - 3-simplex: *tetrahedron*



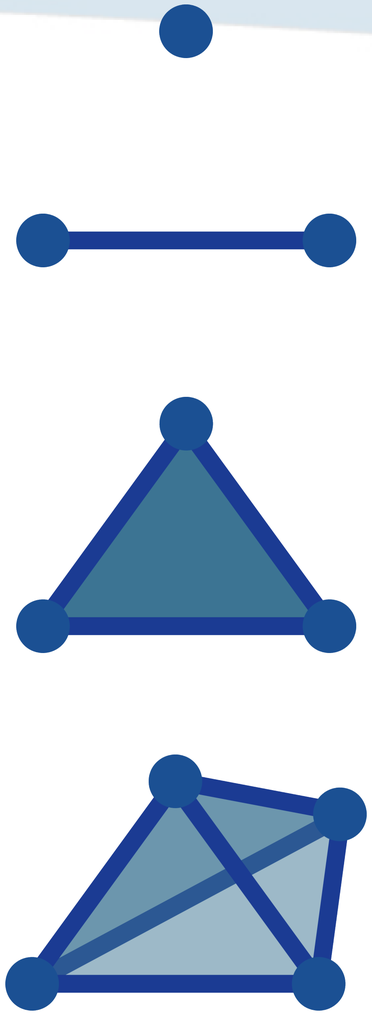
Manifolds on a computer

- Notion of face



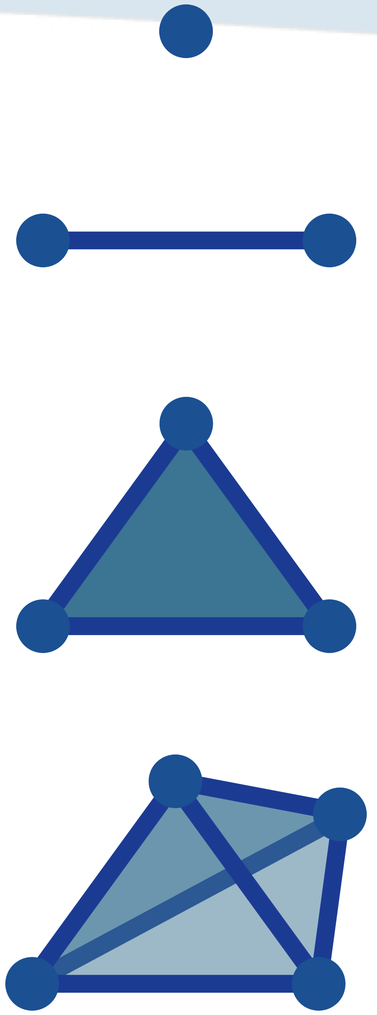
Manifolds on a computer

- Notion of face
  - A face τ of a d-simplex σ is the simplex defined by a non-empty subset of the d+1 points of σ
  - Noted $\tau \leq \sigma$



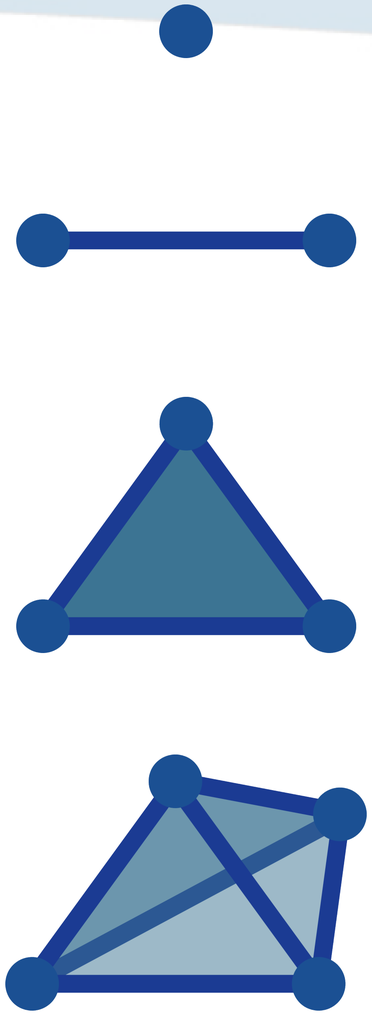
Manifolds on a computer

- Notion of face
  - A face τ of a d-simplex σ is the simplex defined by a non-empty subset of the d+1 points of σ
  - Noted $\tau \leq \sigma$
- Recursive construction!



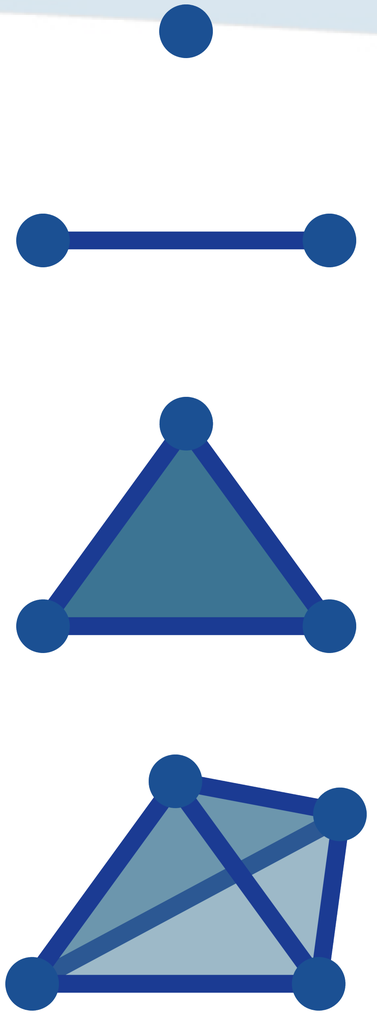
Manifolds on a computer

- Notion of face
  - A face τ of a d-simplex σ is the simplex defined by a non-empty subset of the d+1 points of σ
  - Noted $\tau \leq \sigma$
- Recursive construction!
  - A 3-simplex has 4 2-simplices as faces



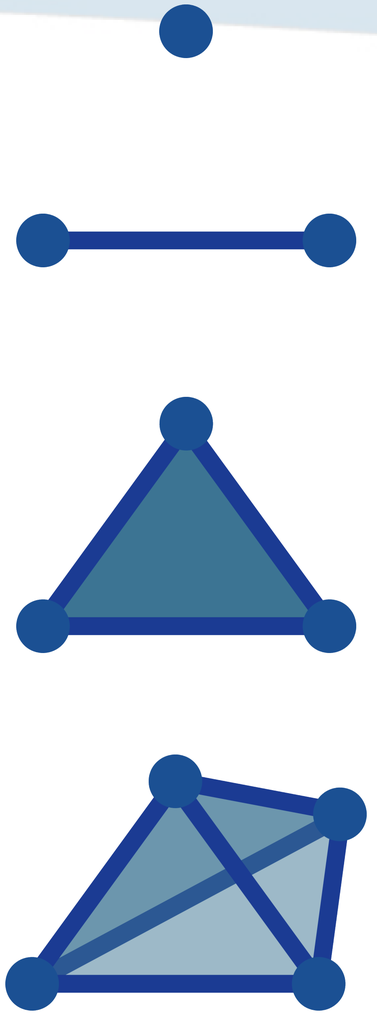
Manifolds on a computer

- Notion of face
  - A face τ of a d-simplex σ is the simplex defined by a non-empty subset of the d+1 points of σ
  - Noted $\tau \leq \sigma$
- Recursive construction!
  - A 3-simplex has 4 2-simplices as faces
  - A 2-simplex has 3 1-simplices as faces



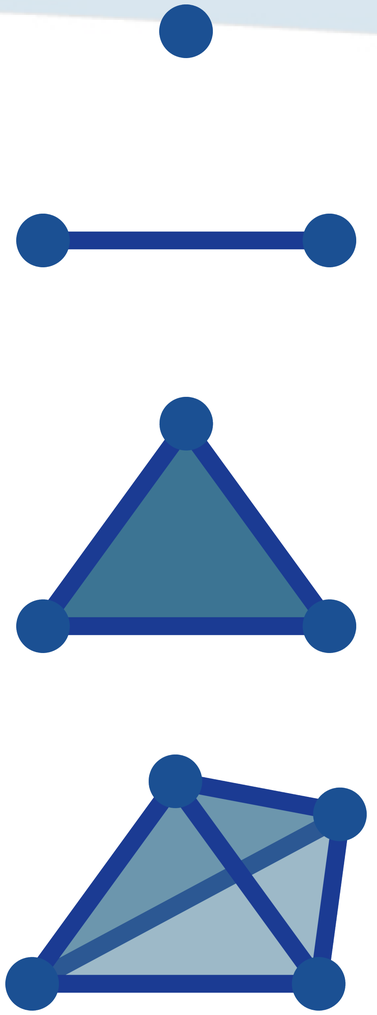
Manifolds on a computer

- Notion of face
  - A face τ of a d-simplex σ is the simplex defined by a non-empty subset of the d+1 points of σ
  - Noted $\tau \leq \sigma$
- Recursive construction!
  - A 3-simplex has 4 2-simplices as faces
  - A 2-simplex has 3 1-simplices as faces
  - A 1-simplex has 2 0-simplices as faces



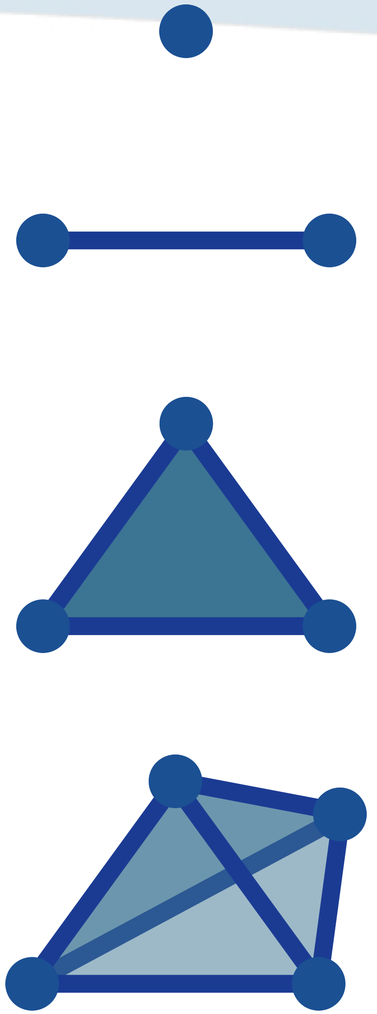
Manifolds on a computer

- Notion of face
  - A face τ of a d-simplex σ is the simplex defined by a non-empty subset of the d+1 points of σ
  - Noted $\tau \leq \sigma$
- Recursive construction!
  - A 3-simplex has 4 2-simplices as faces
  - A 2-simplex has 3 1-simplices as faces
  - A 1-simplex has 2 0-simplices as faces
  - How many faces in a 3-simplex (total)?



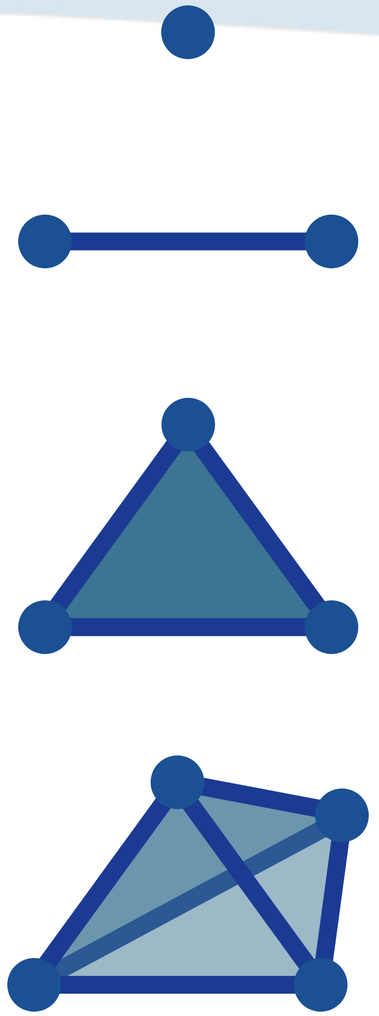
Manifolds on a computer

- Notion of face
  - A face τ of a d-simplex σ is the simplex defined by a non-empty subset of the d+1 points of σ
  - Noted $\tau \leq \sigma$
- Recursive construction!
  - A 3-simplex has 4 2-simplices as faces
  - A 2-simplex has 3 1-simplices as faces
  - A 1-simplex has 2 0-simplices as faces
  - How many faces in a 3-simplex (total)? 15



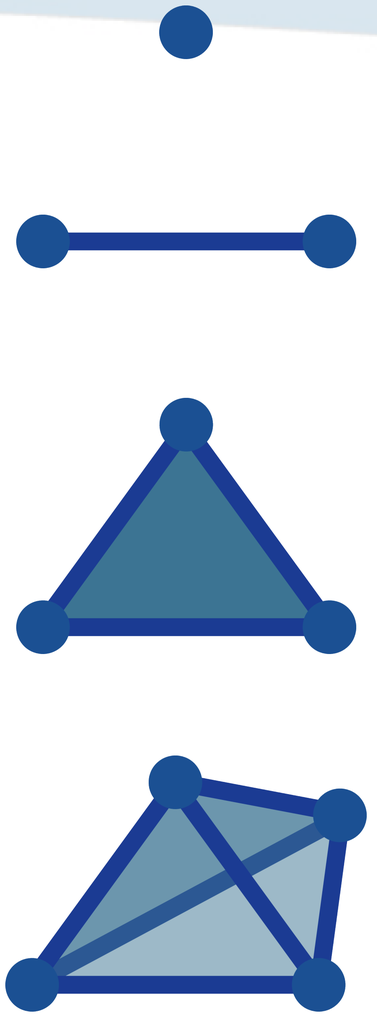
Manifolds on a computer

- Notion of **simplicial complex**



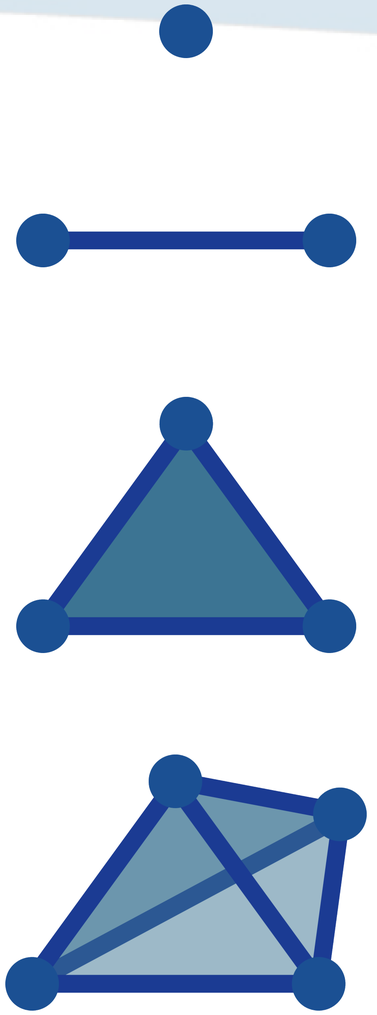
Manifolds on a computer

- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that



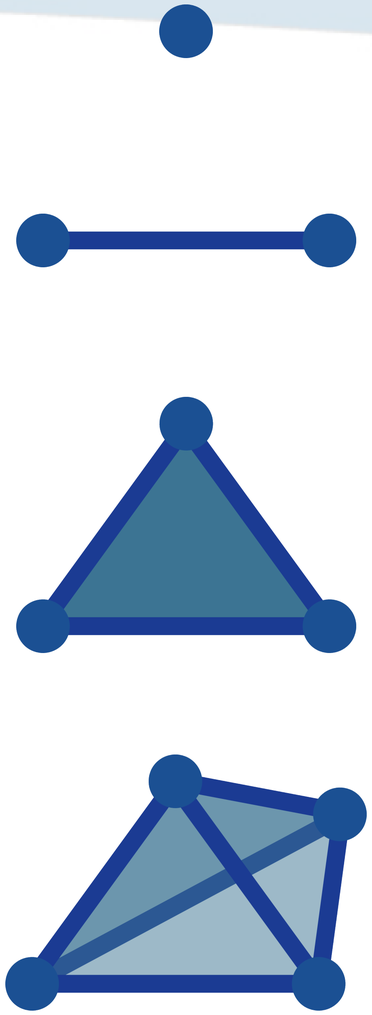
Manifolds on a computer

- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$

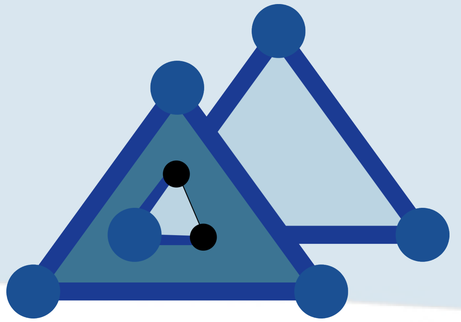


Manifolds on a computer

- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$
    - Any two simplices σ_i and σ_j intersect in a common face or not at all.

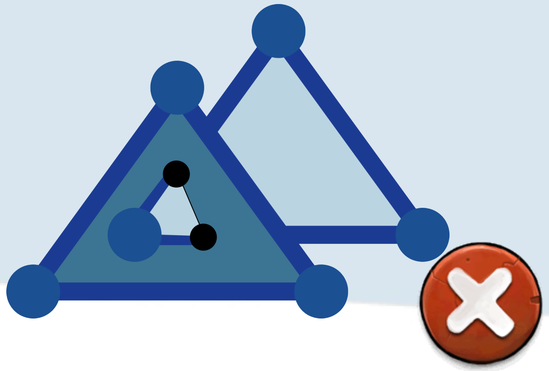


Manifolds on a computer



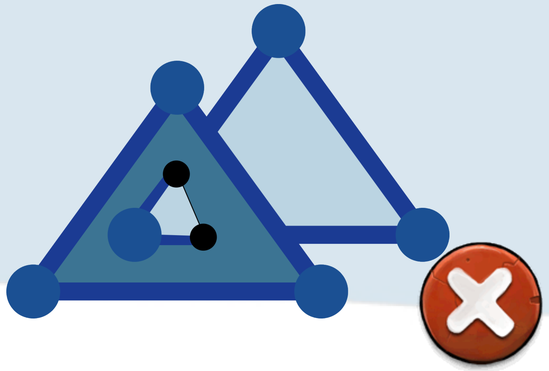
- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$
    - Any two simplices σ_i and σ_j intersect in a common face or not at all.

Manifolds on a computer

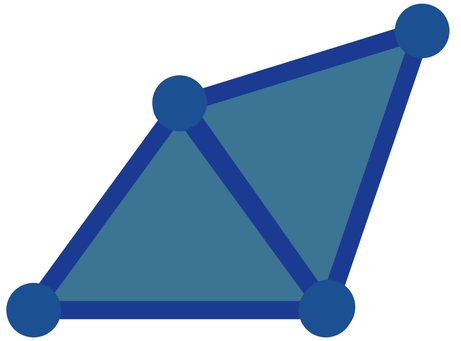


- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$
    - Any two simplices σ_i and σ_j intersect in a common face or not at all.

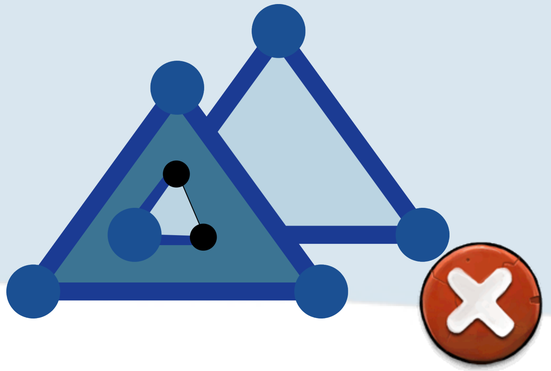
Manifolds on a computer



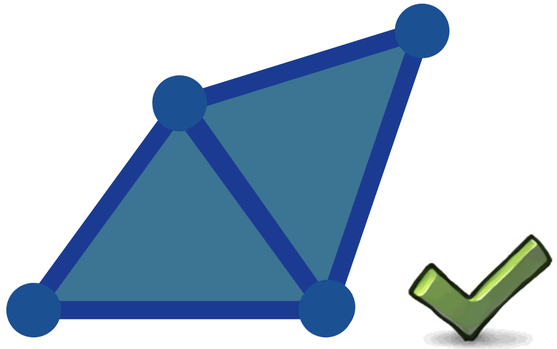
- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$
    - Any two simplices σ_i and σ_j intersect in a common face or not at all.



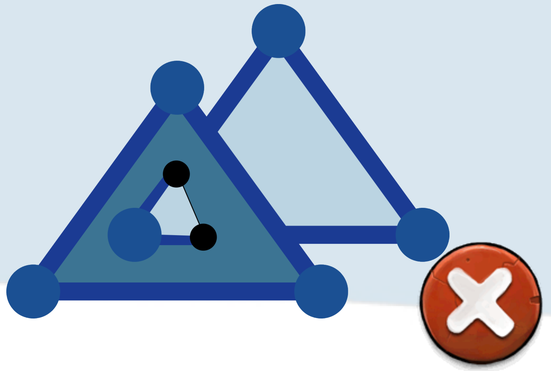
Manifolds on a computer



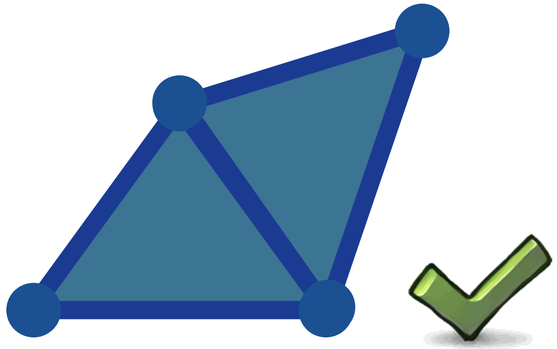
- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$
    - Any two simplices σ_i and σ_j intersect in a common face or not at all.

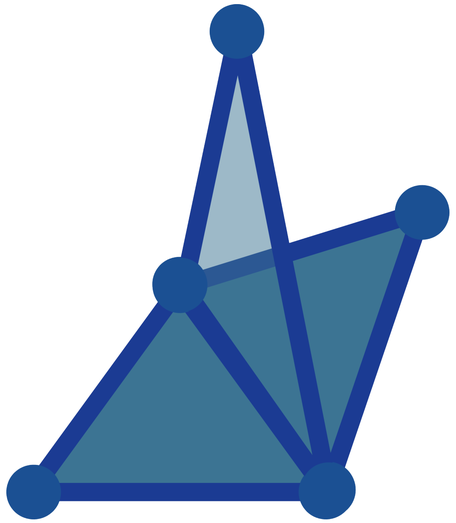


Manifolds on a computer

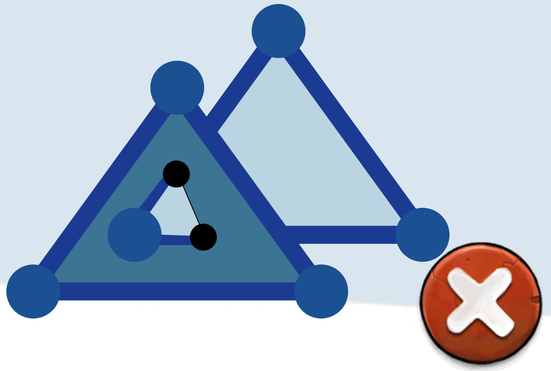


- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$
    - Any two simplices σ_i and σ_j intersect in a common face or not at all.

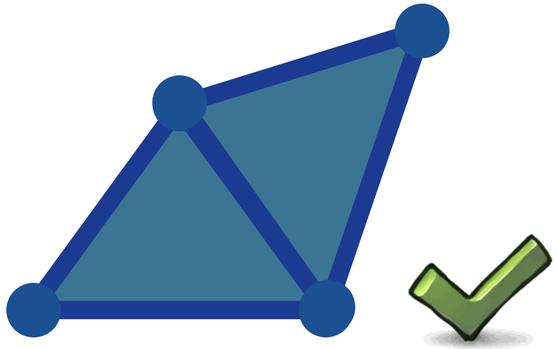


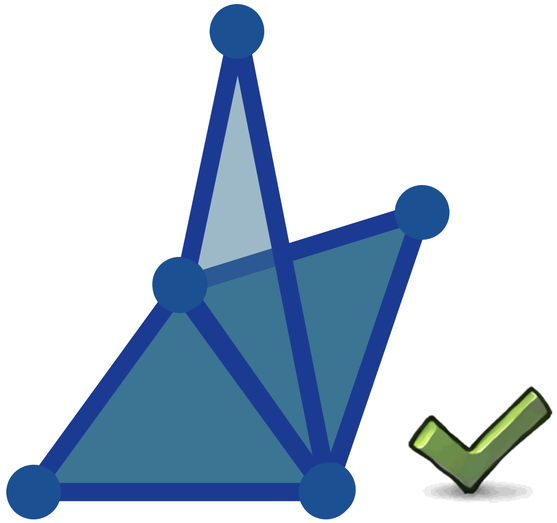


Manifolds on a computer



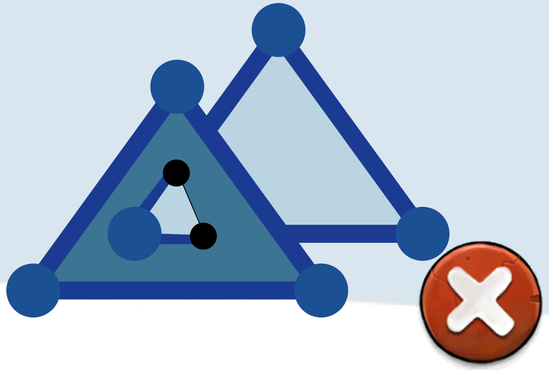
- Notion of **simplicial complex**
  - A simplicial complex $\mathcal{K}$ is a finite collection of non-empty simplices $\{\sigma_i\}$ such that
    - Every face τ of a simplex σ_i is also in $\mathcal{K}$
    - Any two simplices σ_i and σ_j intersect in a common face or not at all.

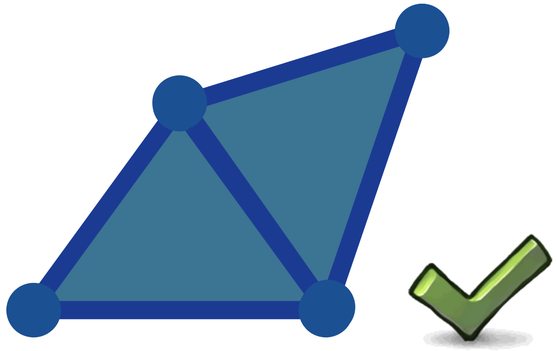


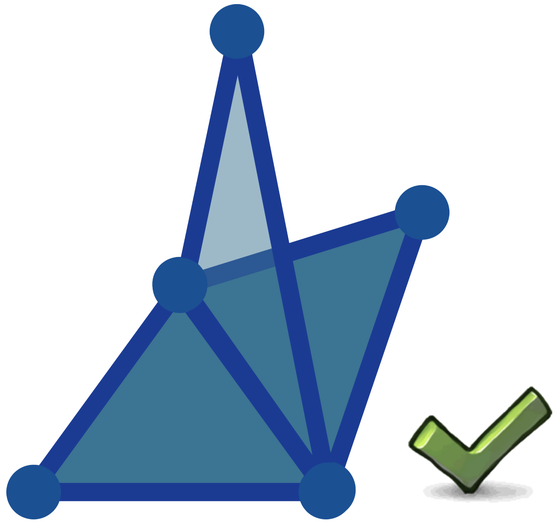


Manifolds on a computer

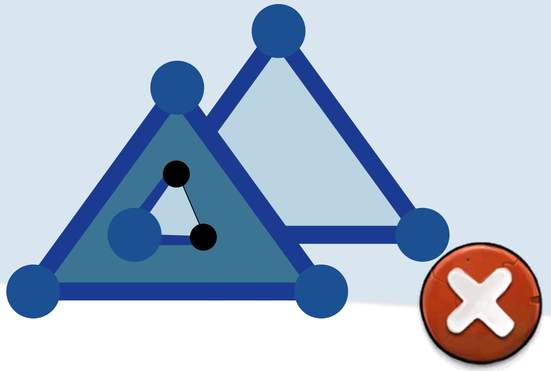
- Notion of **triangulation**



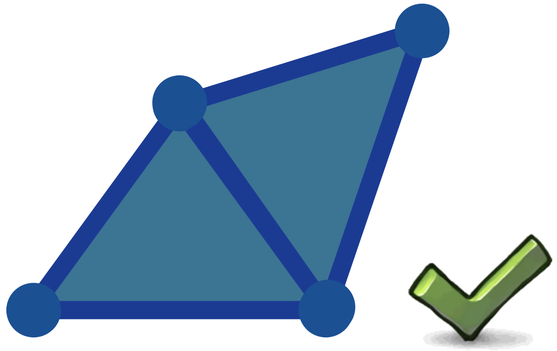


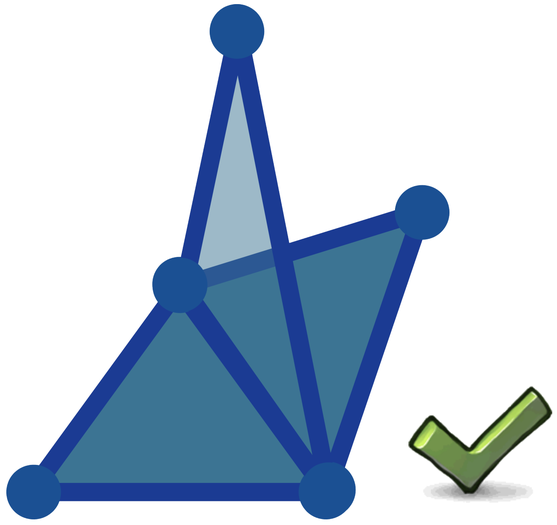


Manifolds on a computer

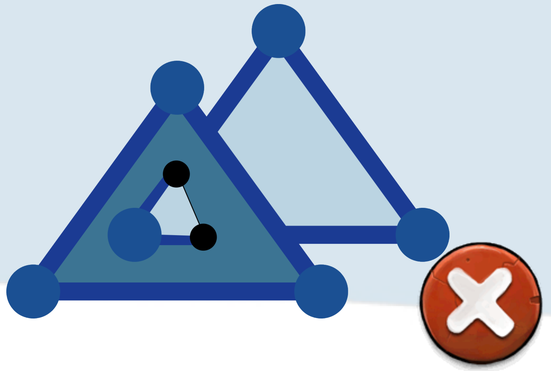


- Notion of **triangulation**
  - The triangulation of a d-manifold $\mathbb{M}$ is a simplicial complex $\mathcal{K}$ such that

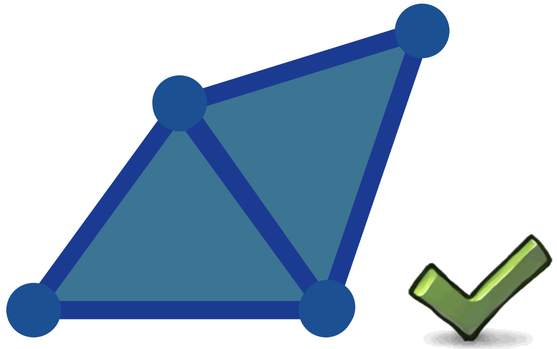


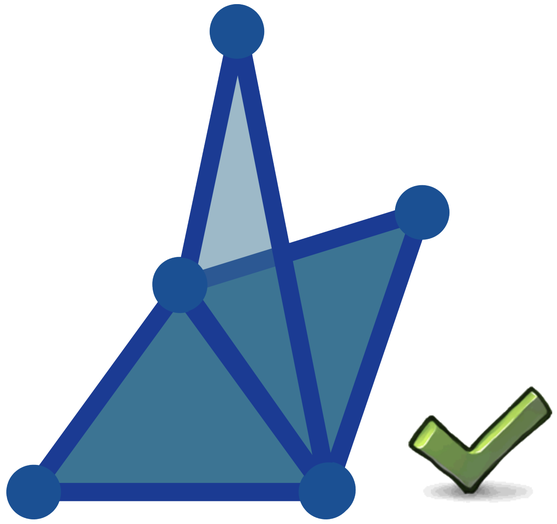


Manifolds on a computer

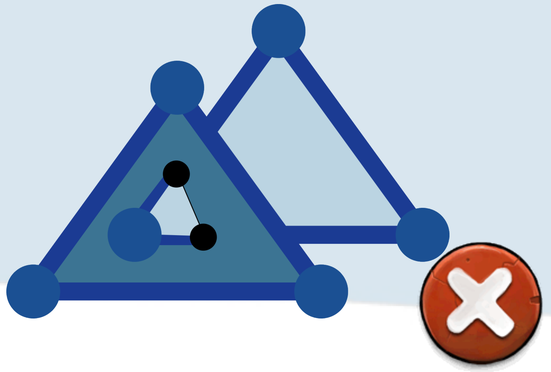


- Notion of **triangulation**
  - The triangulation of a d-manifold $\mathbb{M}$ is a simplicial complex $\mathcal{K}$ such that
    - The union $|\mathcal{K}| = \cup_{\sigma \in \mathcal{K}} \sigma$ of the simplices of $\mathcal{K}$ is homeomorphic to $\mathbb{M}$

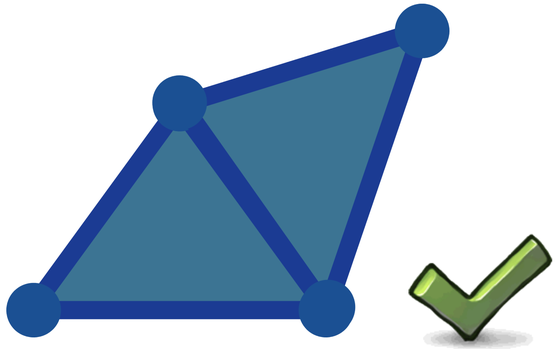


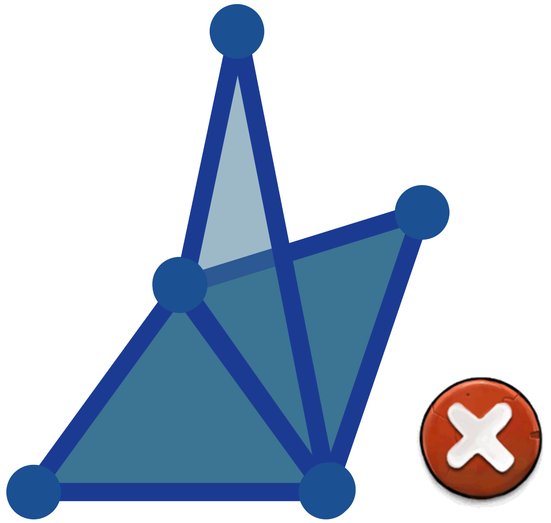


Manifolds on a computer

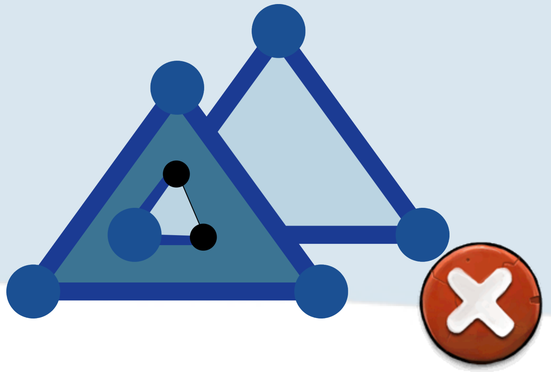


- Notion of **triangulation**
  - The triangulation of a d-manifold $\mathbb{M}$ is a simplicial complex $\mathcal{K}$ such that
    - The union $|\mathcal{K}| = \cup_{\sigma \in \mathcal{K}} \sigma$ of the simplices of $\mathcal{K}$ is homeomorphic to $\mathbb{M}$

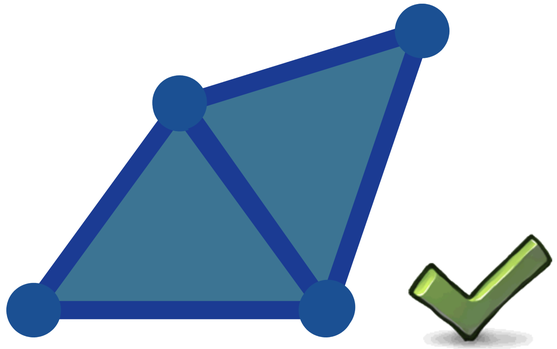


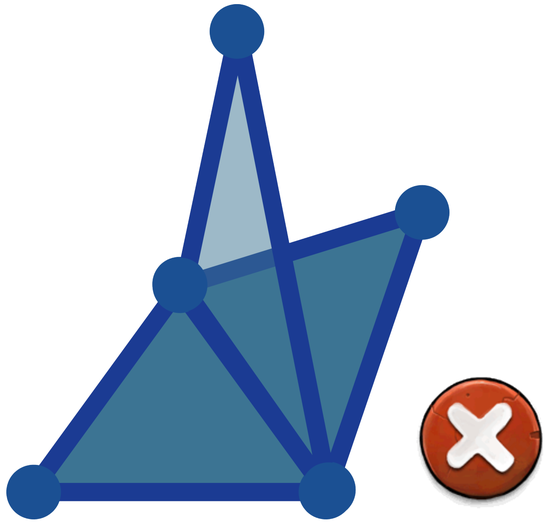


Manifolds on a computer

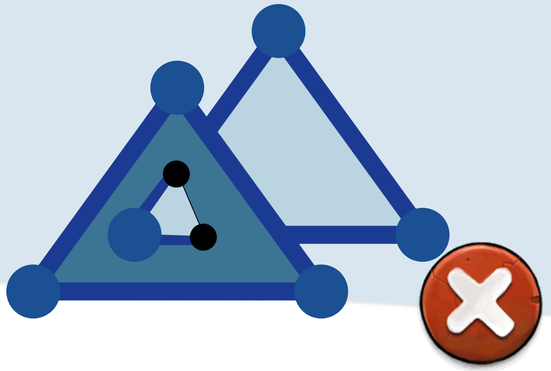


- Notion of **triangulation**
  - The triangulation of a d-manifold $\mathbb{M}$ is a simplicial complex $\mathcal{K}$ such that
    - The union $|\mathcal{K}| = \cup_{\sigma \in \mathcal{K}} \sigma$ of the simplices of $\mathcal{K}$ is homeomorphic to $\mathbb{M}$
  - Any open set of $|\mathcal{K}|$ is homeomorphic to $\mathbb{R}^d$

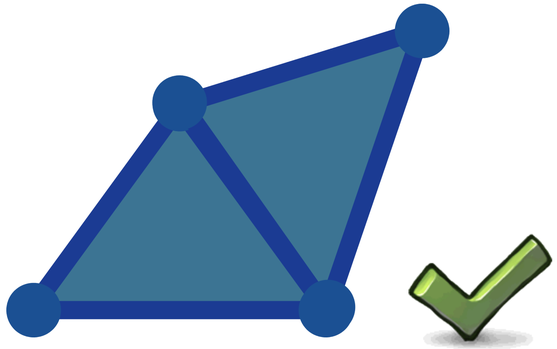




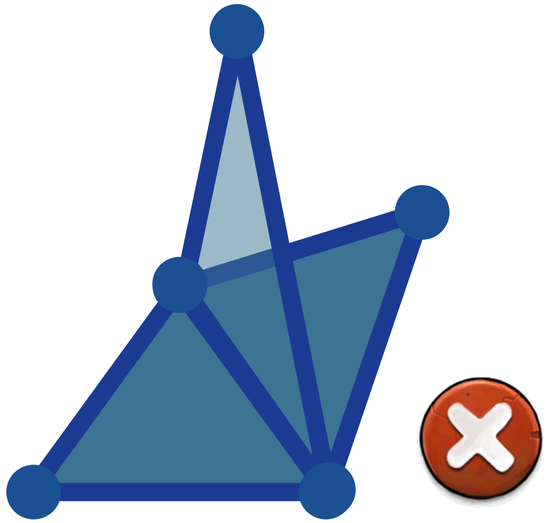
Manifolds on a computer

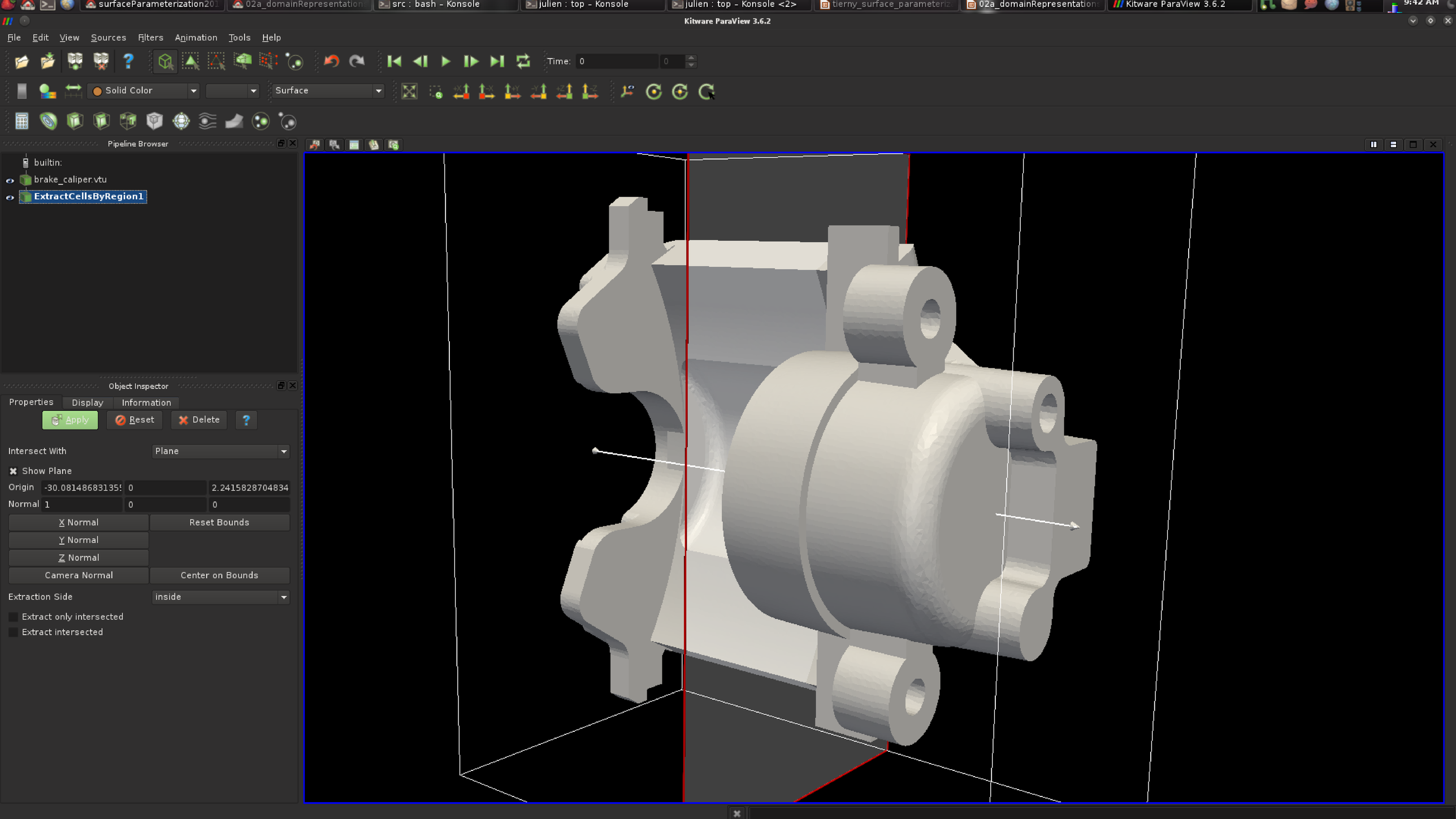


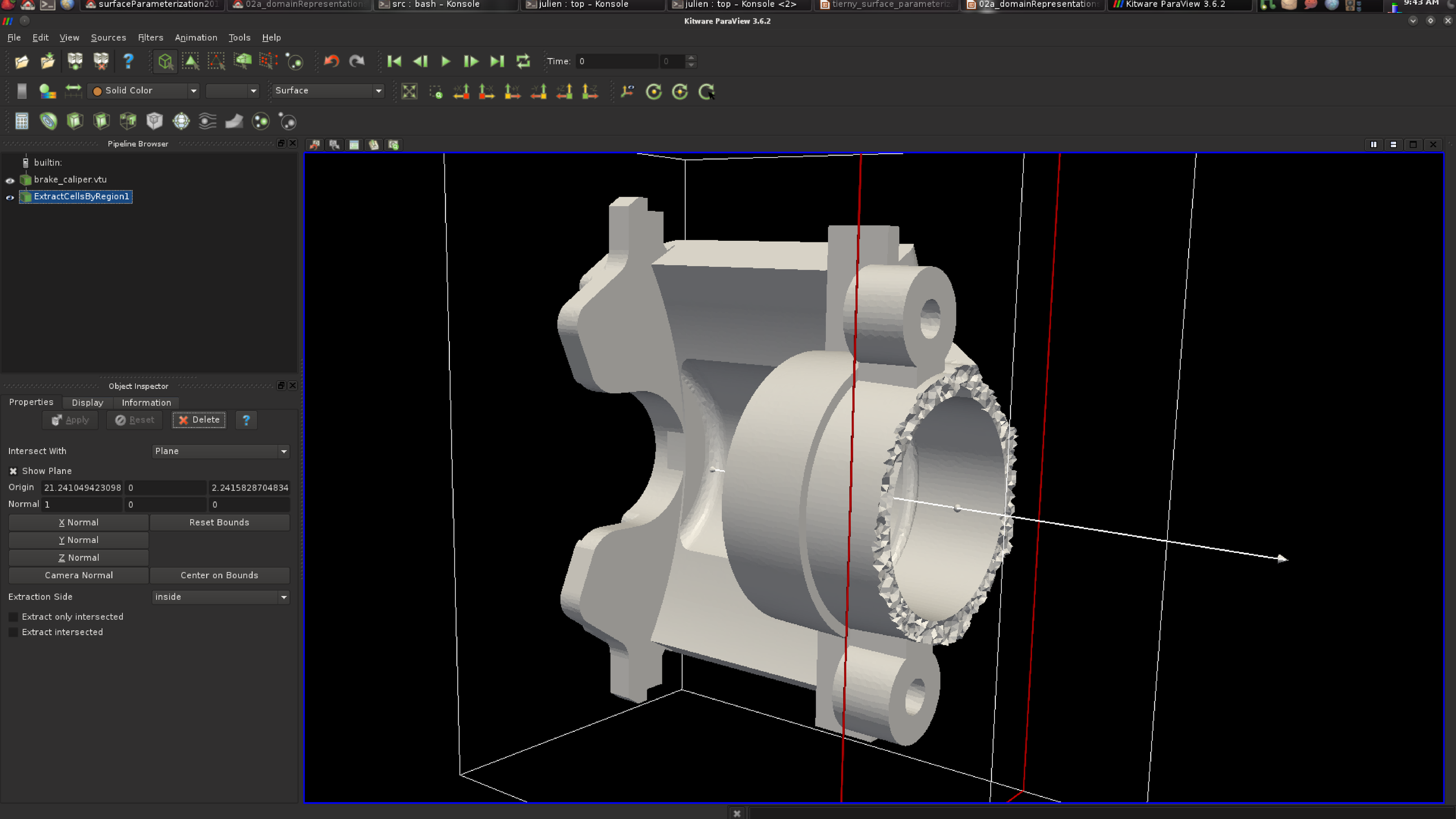
- Notion of **triangulation**
  - The triangulation of a d-manifold $\mathbb{M}$ is a simplicial complex $\mathcal{K}$ such that
    - The union $|\mathcal{K}| = \cup_{\sigma \in \mathcal{K}} \sigma$ of the simplices of $\mathcal{K}$ is homeomorphic to $\mathbb{M}$
  - Any open set of $|\mathcal{K}|$ is homeomorphic to $\mathbb{R}^d$

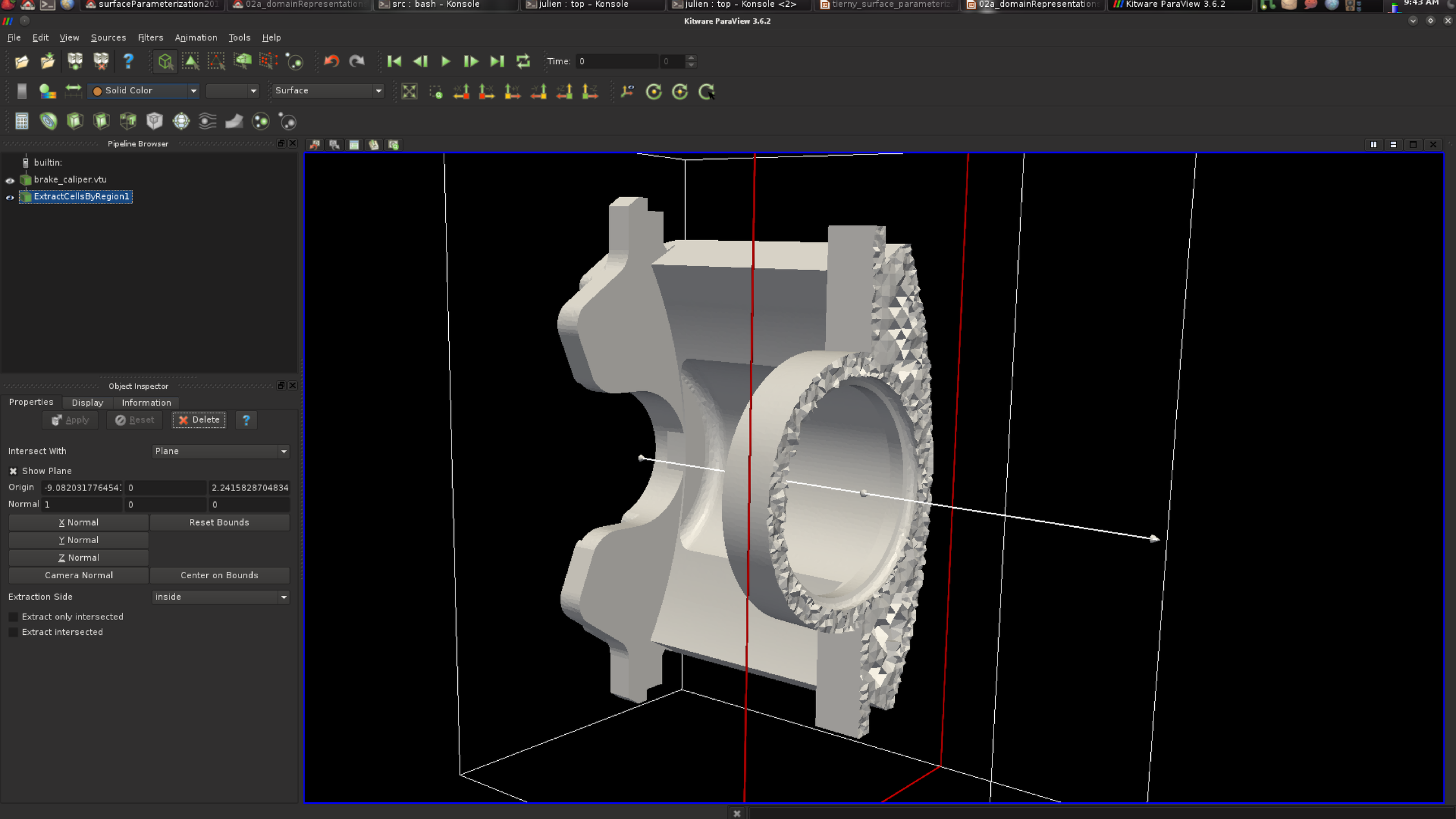


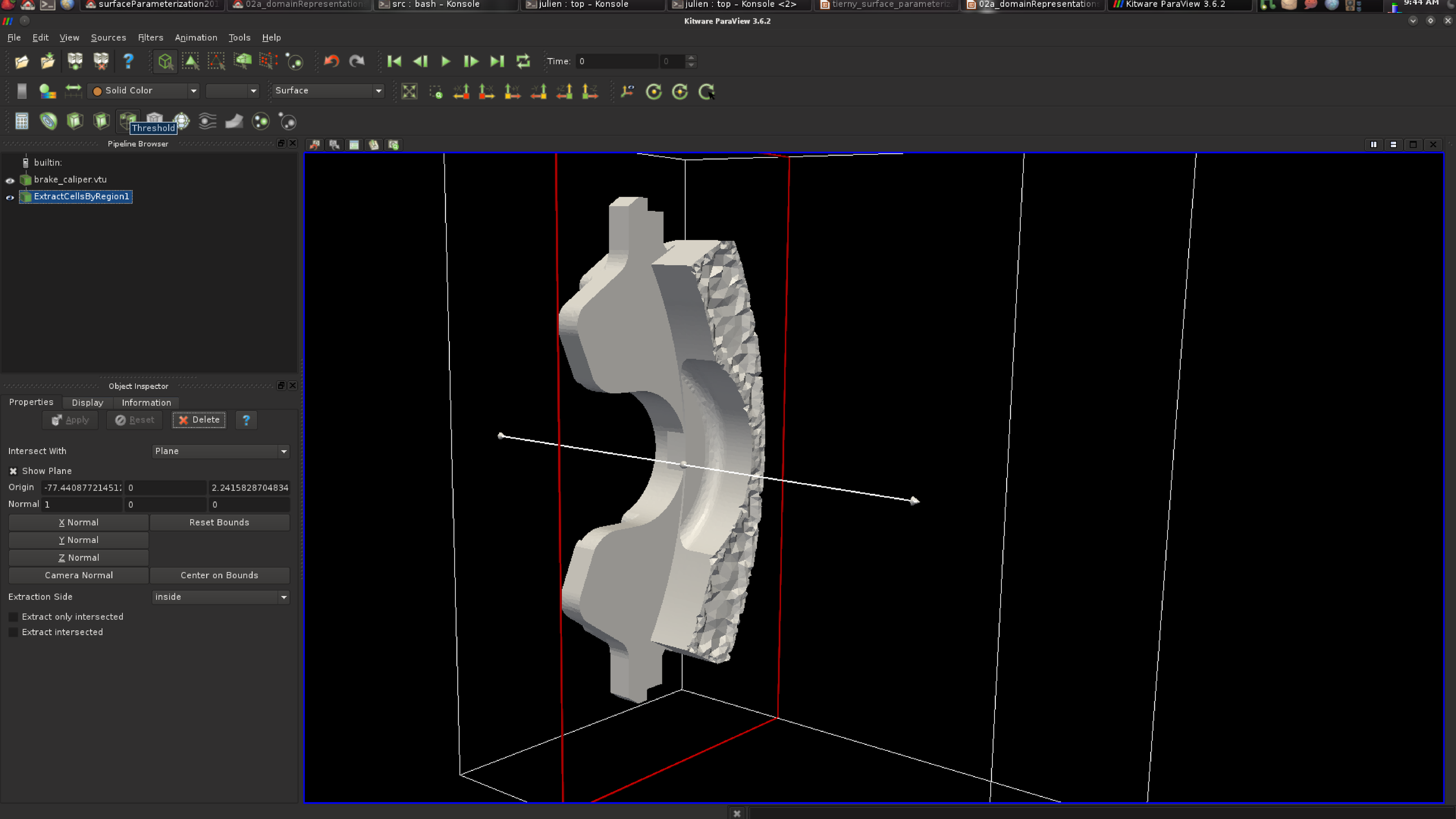
- 2-triangulation: *triangle mesh*
- 3-triangulation: *tetrahedral mesh*




surfaceParameterization201
02a_domainRepresentation
src : bash - Konsole
julien : top - Konsole
julien : top - Konsole <2>
tierny_surface_parameteriz
02a_domainRepresentation
Kitware ParaView 3.6.2
9:42 AM
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 -30.08148683135 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

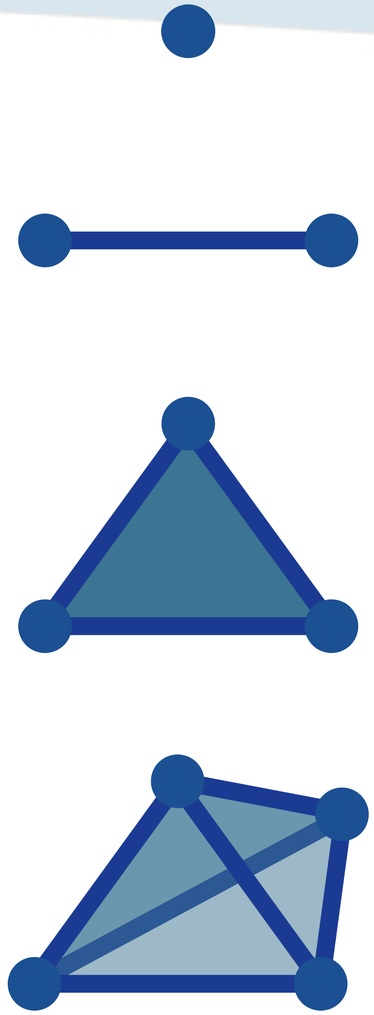

Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Solid Color
Surface
Pipeline Browser
builtin:
brake_caliper.vtu
ExtractCellsByRegion1
Object Inspector
Properties Display Information
Apply Reset Delete
Intersect With Plane
Show Plane
Origin 21.241049423098 0 2.2415828704834
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


surfaceParameterization201
02a_domainRepresentation
src : bash - Konsole
julien : top - Konsole
julien : top - Konsole <2>
tierny_surface_parameteriz
02a_domainRepresentation
Kitware ParaView 3.6.2
9:43 AM
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 -9.082031776454 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


surfaceParameterization201
02a_domainRepresentation
src : bash - Konsole
julien : top - Konsole
julien : top - Konsole <2>
tierny_surface_parameteriz
02a_domainRepresentation
Kitware ParaView 3.6.2
9:44 AM
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Solid Color
Surface
Threshold
Pipeline Browser
builtin:
brake_caliper.vtu
ExtractCellsByRegion1
Object Inspector
Properties Display Information
Apply Reset Delete
Intersect With Plane
Show Plane
Origin -77.44087721451 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

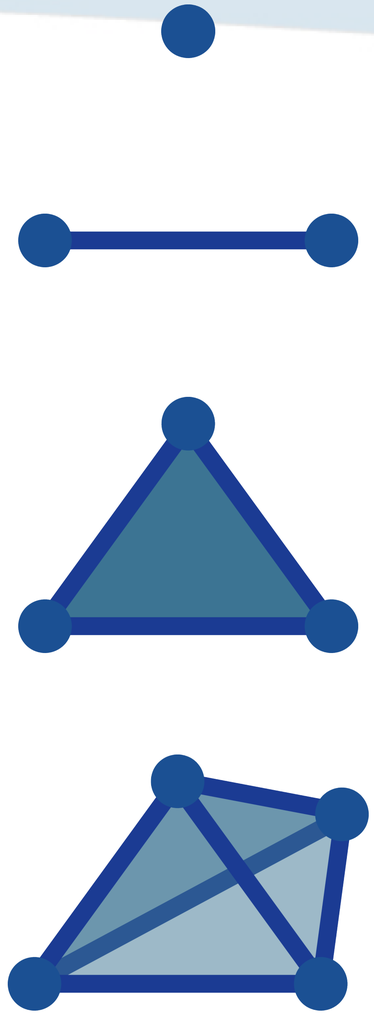
Interpolants for triangulations

- Like regular grids, several options



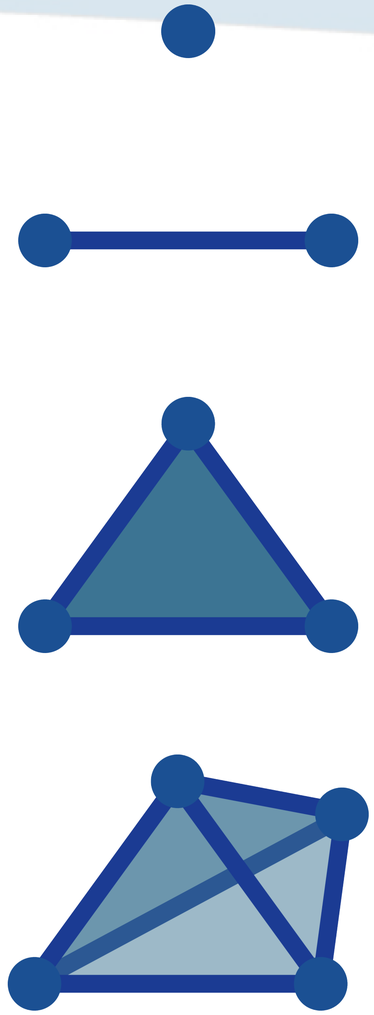
Interpolants for triangulations

- Like regular grids, several options
  - Piecewise constant
  - Piecewise linear
  - Piecewise polynomials,
  - etc.



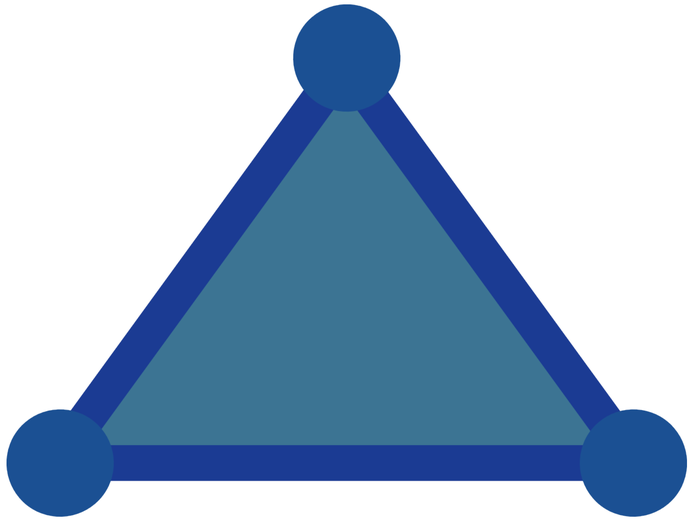
Interpolants for triangulations

- Like regular grids, several options
  - Piecewise constant
  - **Piecewise linear**
  - Piecewise polynomials,
  - etc.



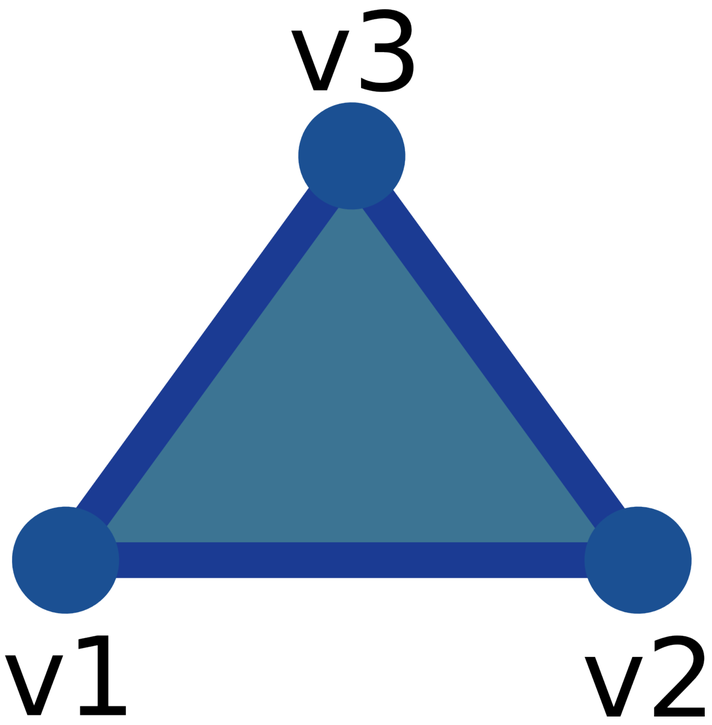
Piecewise linear interpolant on triangulations

- Interpolation as a boundary value problem



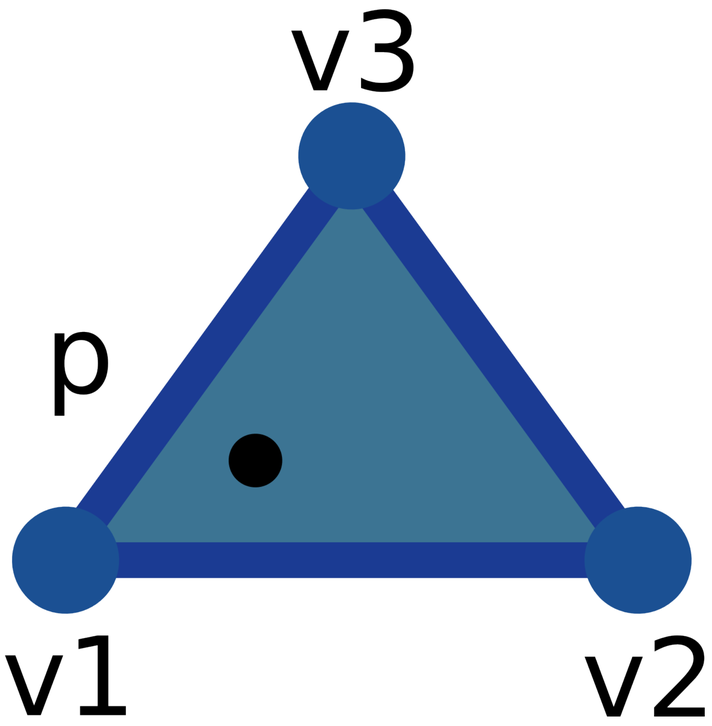
Piecewise linear interpolant on triangulations

- Interpolation as a boundary value problem
  - Given a simplex and its vertices



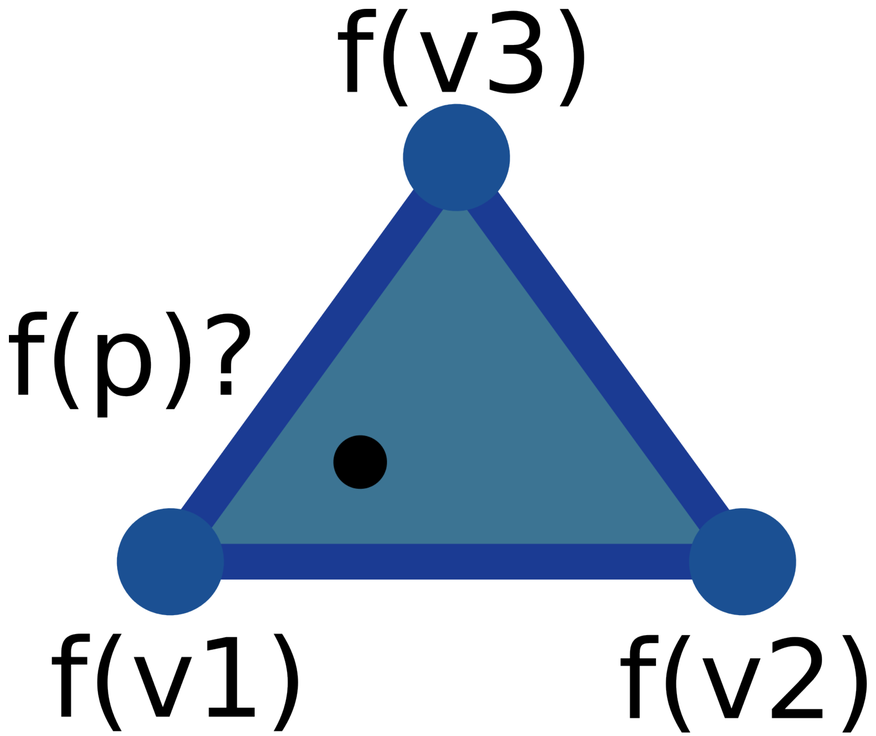
Piecewise linear interpolant on triangulations

- Interpolation as a boundary value problem
  - Given a simplex and its vertices
  - Express the value of any point



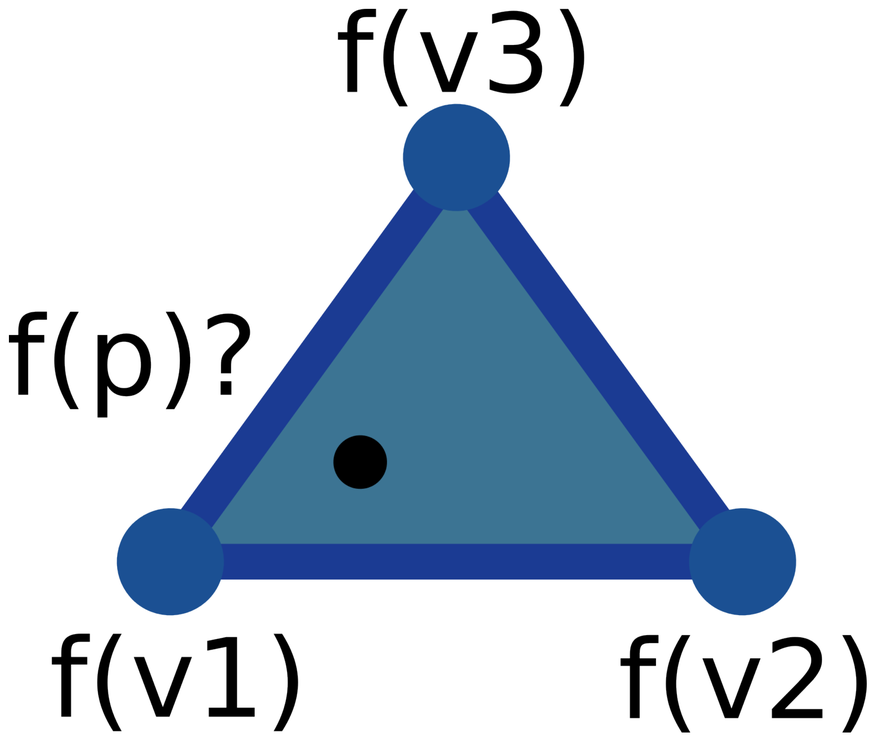
Piecewise linear interpolant on triangulations

- Interpolation as a boundary value problem
  - Given a simplex and its vertices
  - Express the value of any point
  - As a linear combination of those of the vertices



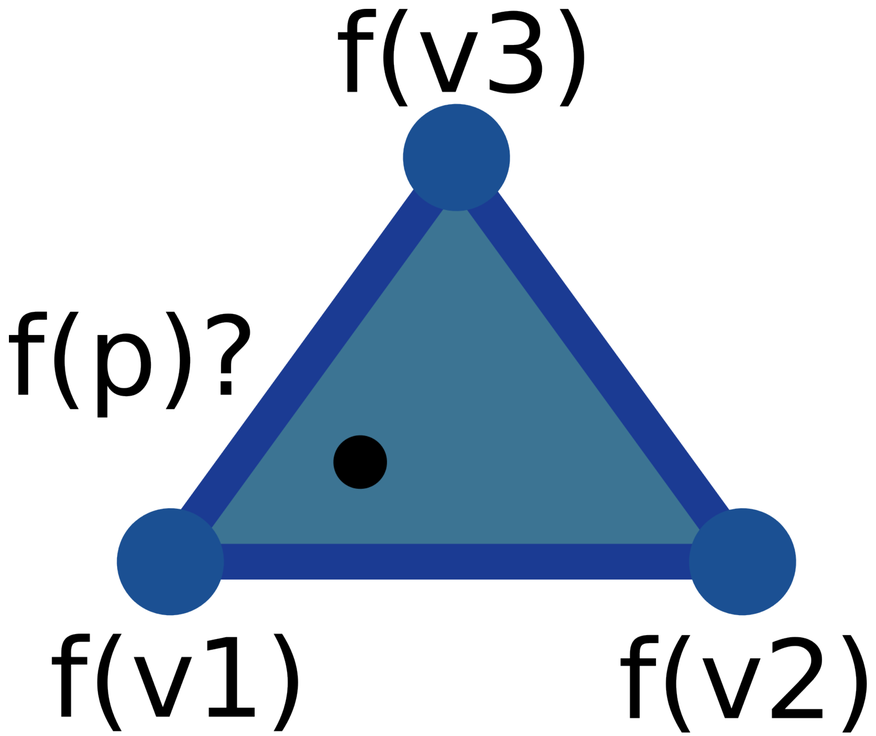
Piecewise linear interpolant on triangulations

- Interpolation as a boundary value problem
  - Given a simplex and its vertices
  - Express the value of any point
  - As a linear combination of those of the vertices
- Notion of barycentric coordinates:
  - $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$



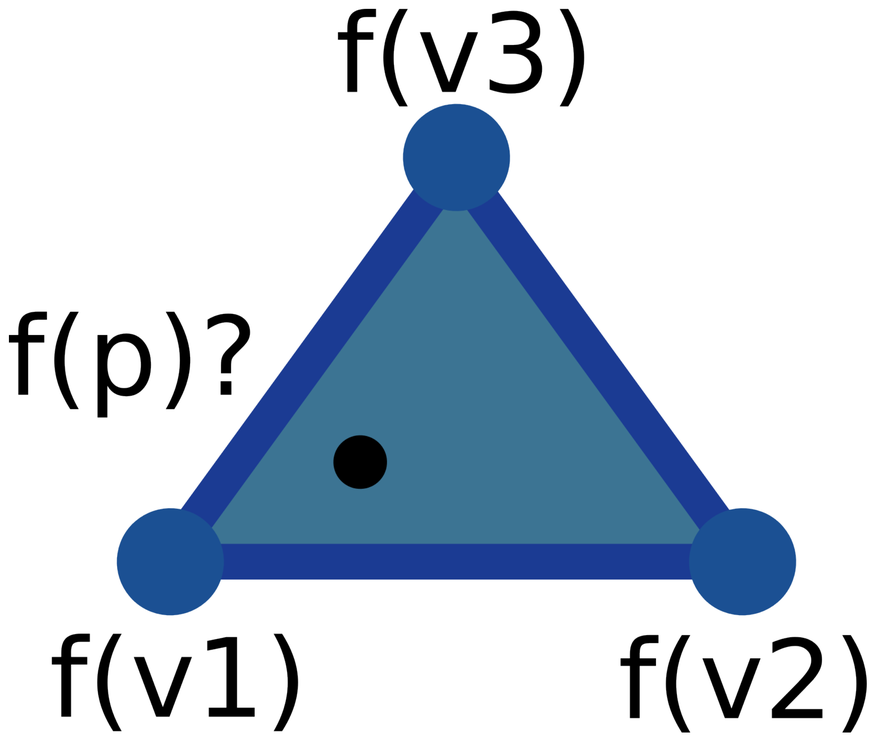
Barycentric coordinates

- There exists many forms
  - With specific properties



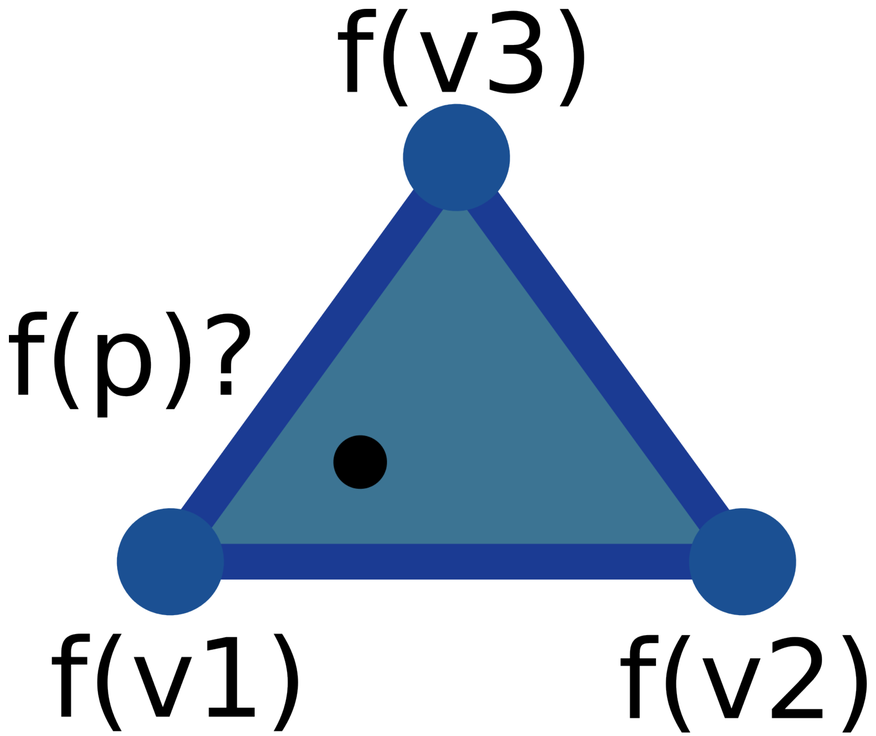
Barycentric coordinates

- There exists many forms
  - With specific properties
- We want to produce linear interpolations



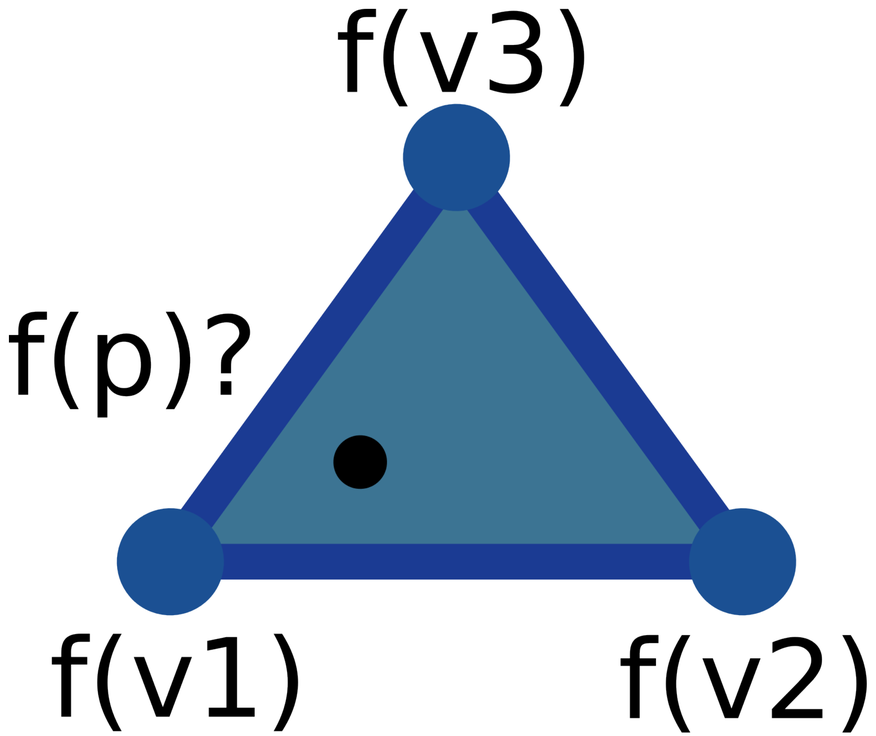
Barycentric coordinates

- There exists many forms
  - With specific properties
- We want to produce linear interpolations
  - $\alpha_1 + \alpha_2 + \alpha_3 = 1$



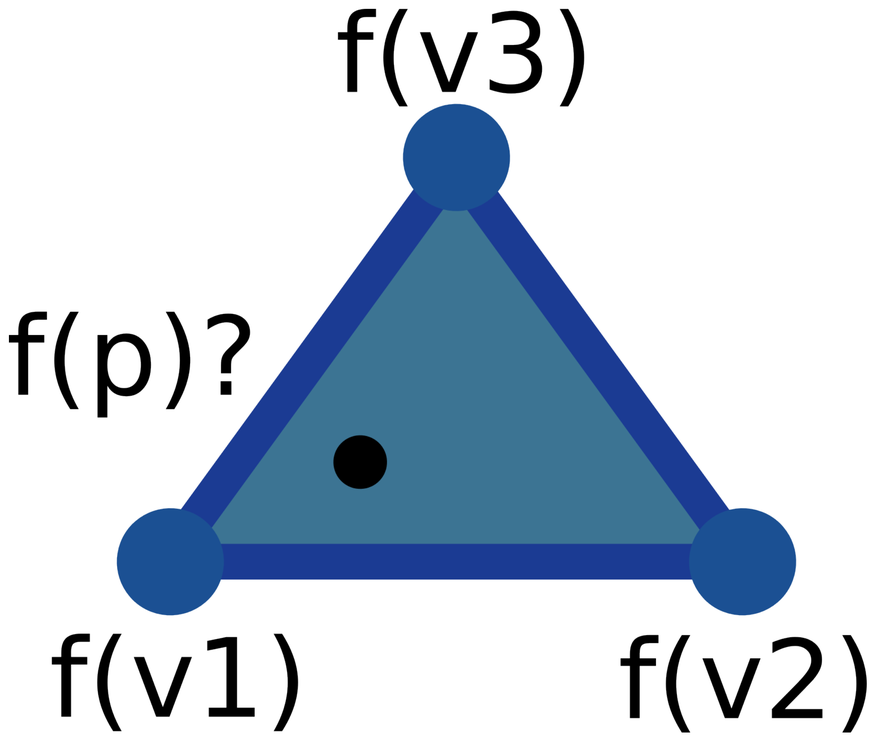
Barycentric coordinates

- There exists many forms
  - With specific properties
- We want to produce linear interpolations
  - $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- Notion of barycentric coordinates:
  - $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$



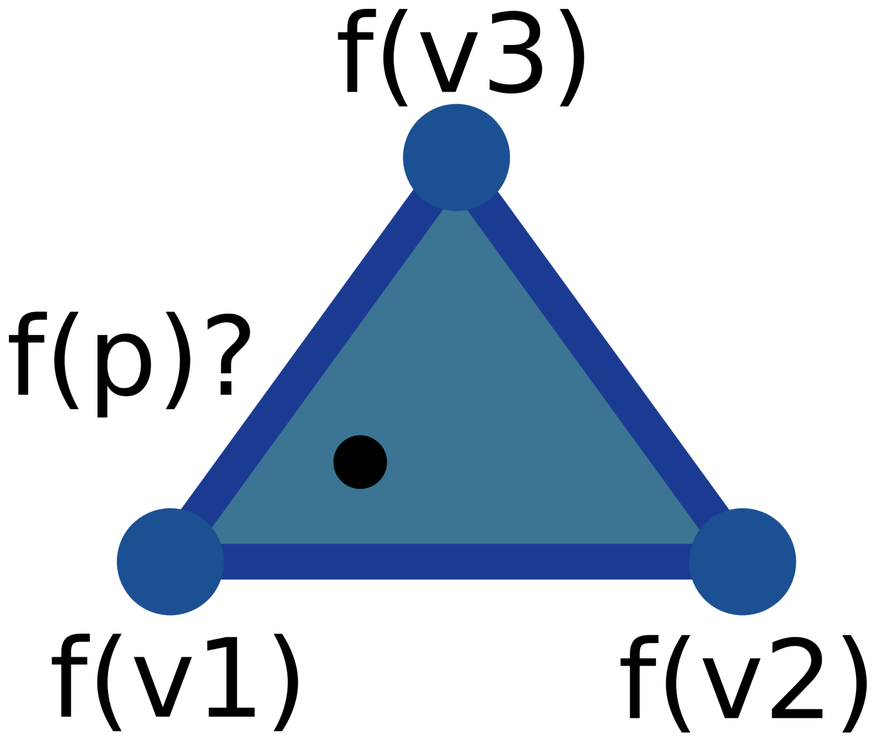
Barycentric coordinates

- There exists many forms
  - With specific properties
- We want to produce linear interpolations
  - $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- Notion of barycentric coordinates:
  - $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
  - Must hold for any $f : \mathcal{T} \rightarrow \mathbb{R}$



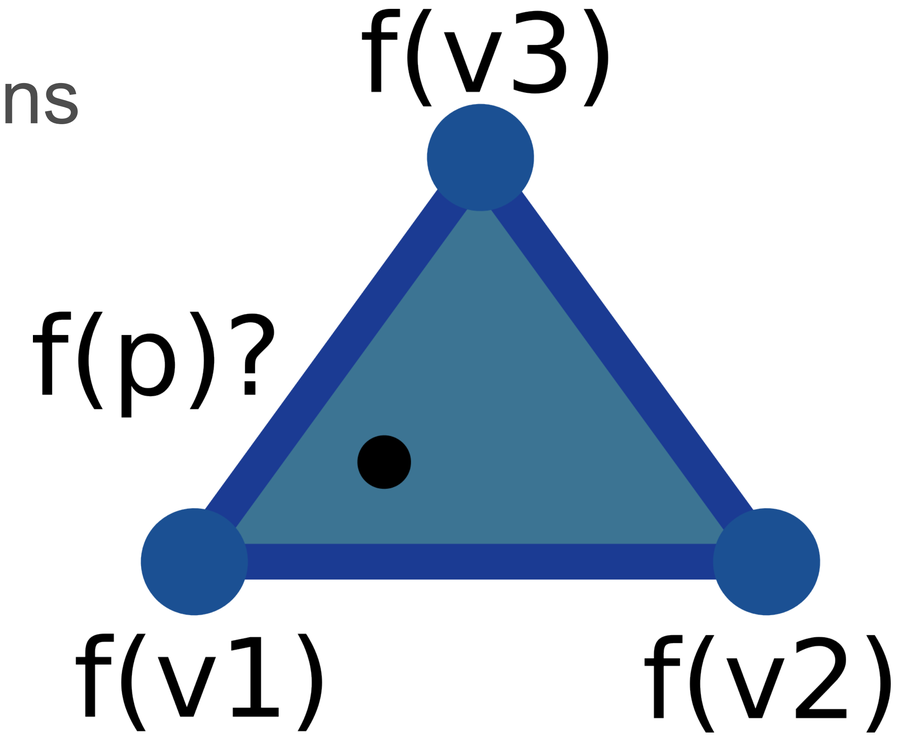
Barycentric coordinates

- In particular



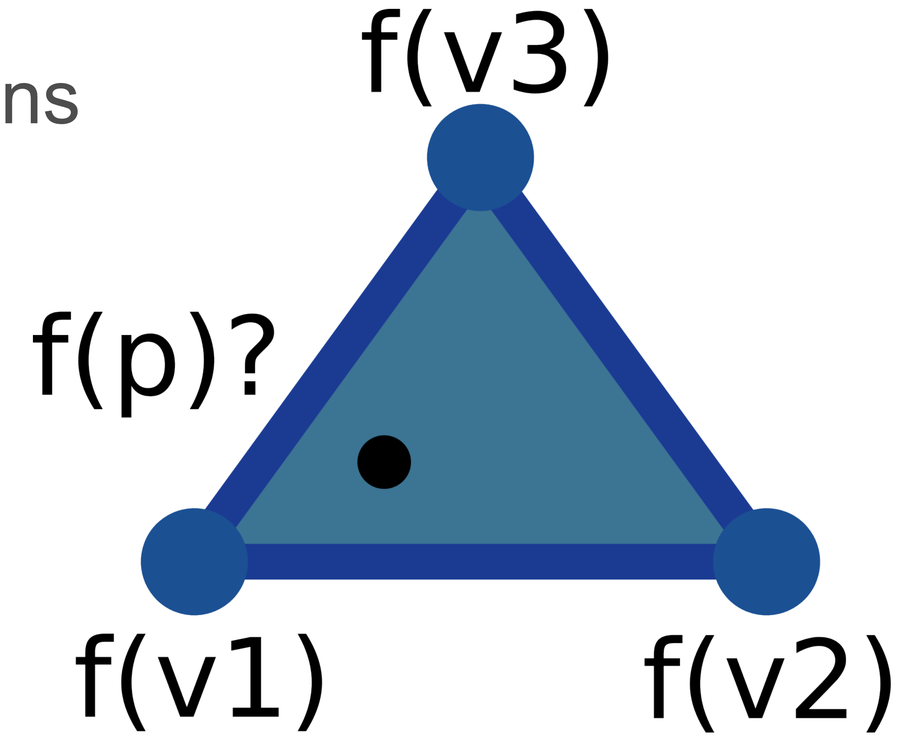
Barycentric coordinates

- In particular
  - It must also hold for the embedding functions



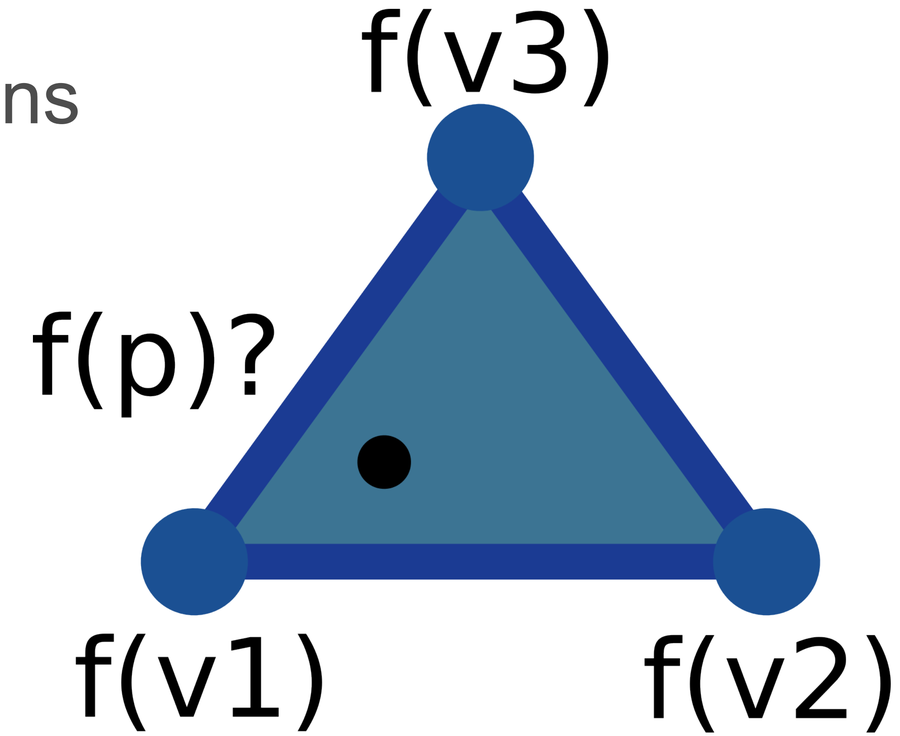
Barycentric coordinates

- In particular
  - It must also hold for the embedding functions
  - $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
  - $\alpha_1 + \alpha_2 + \alpha_3 = 1$
  - $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
  - $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$



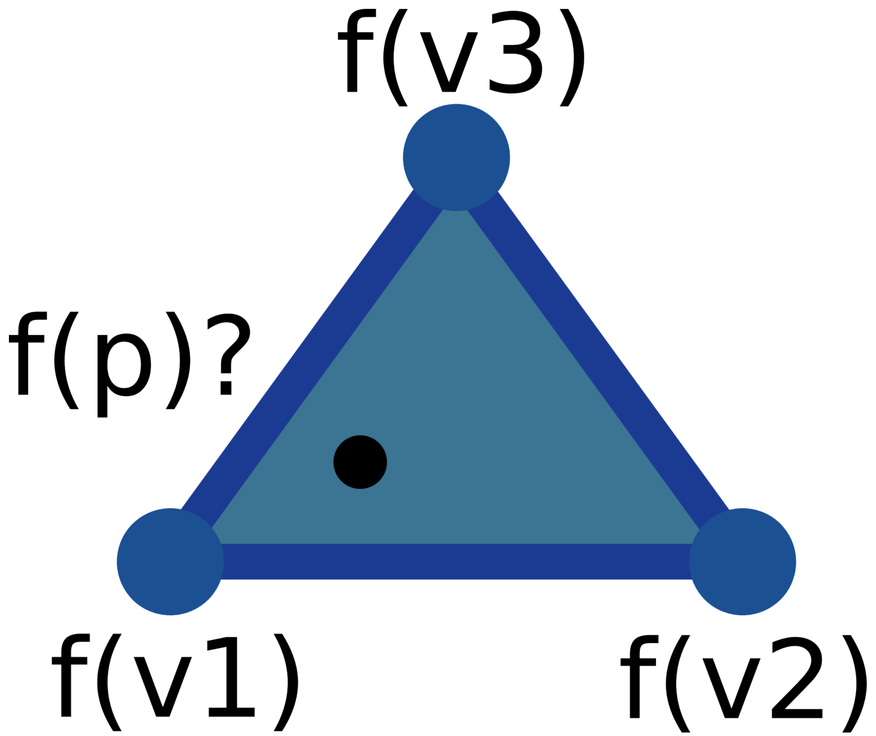
Barycentric coordinates

- In particular
  - It must also hold for the embedding functions
  - $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
  - $\alpha_1 + \alpha_2 + \alpha_3 = 1$
  - $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
  - $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$
- 3 linear equations with 3 unknowns



Barycentric coordinates

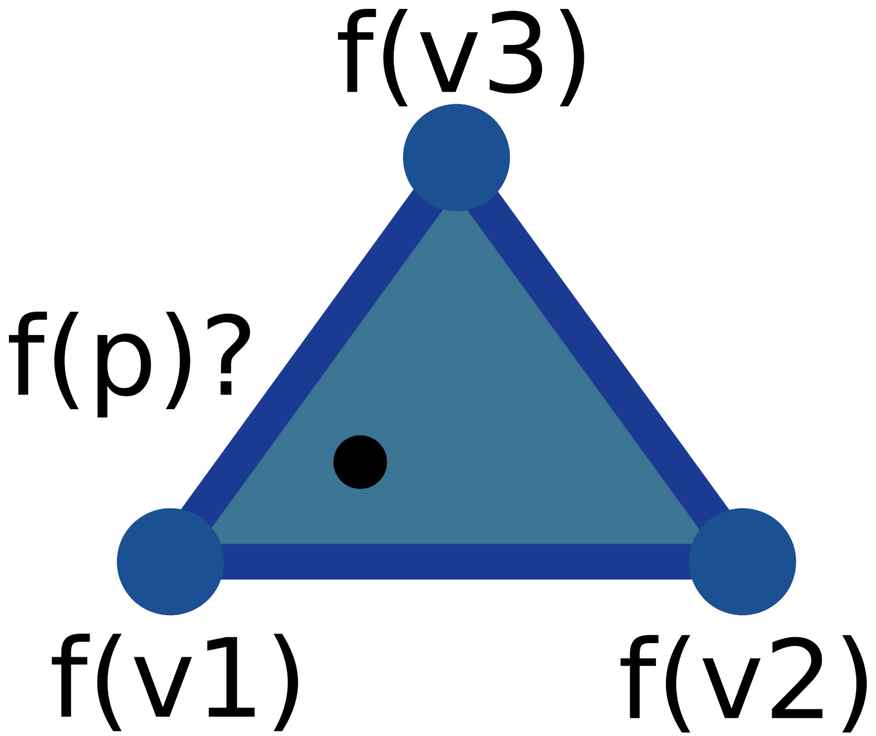
- $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
- $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
- $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$



Barycentric coordinates

- $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
- $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
- $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$

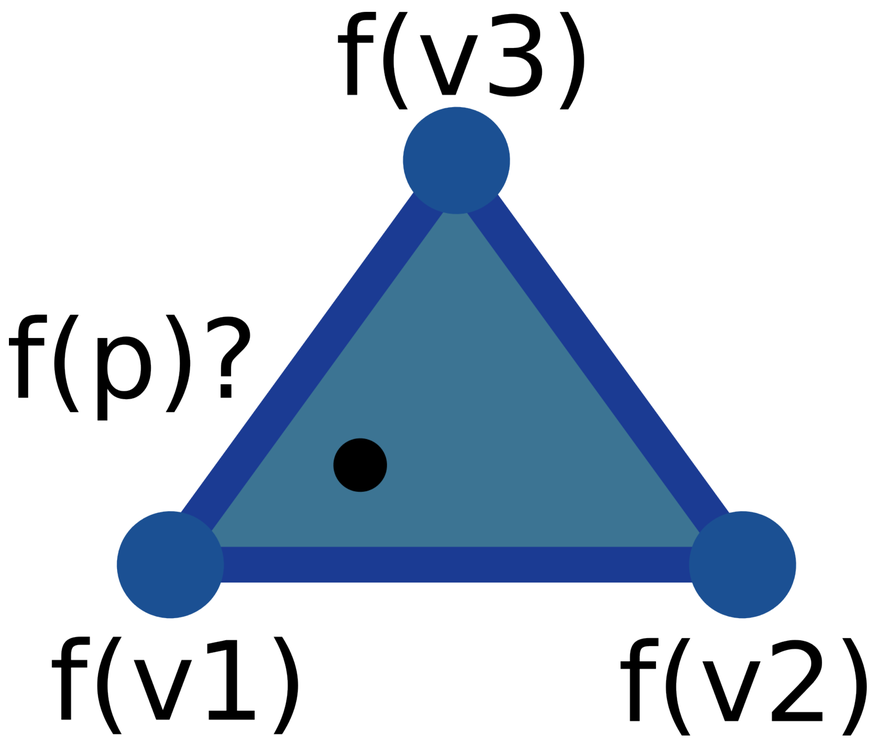
- Automatically interpolates on the edges



Barycentric coordinates

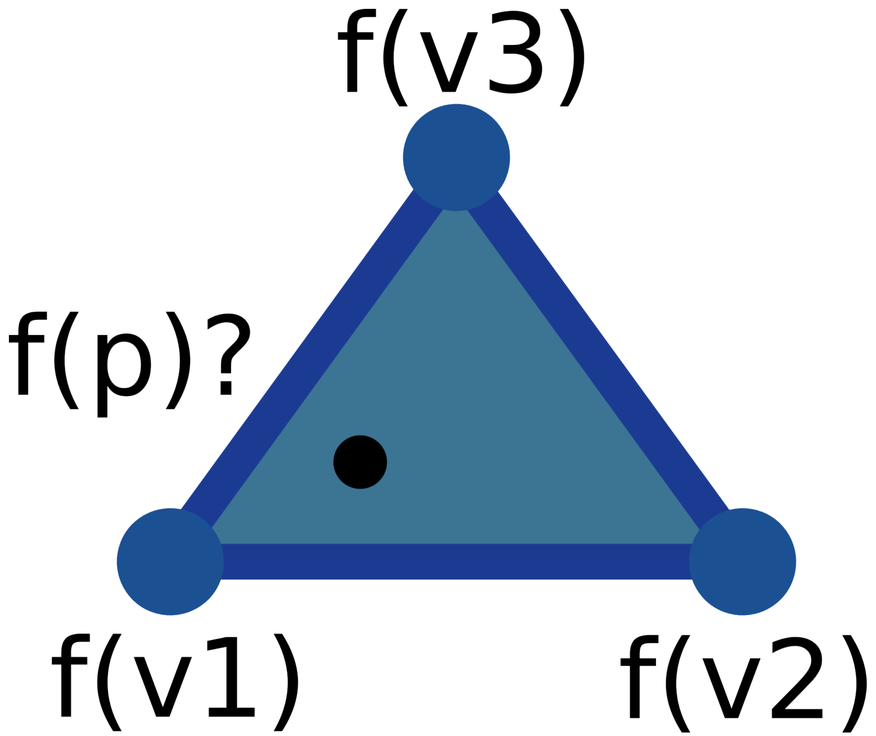
- $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
- $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
- $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$

- Automatically interpolates on the edges
- Similar reasoning for d-simplex



Barycentric coordinates

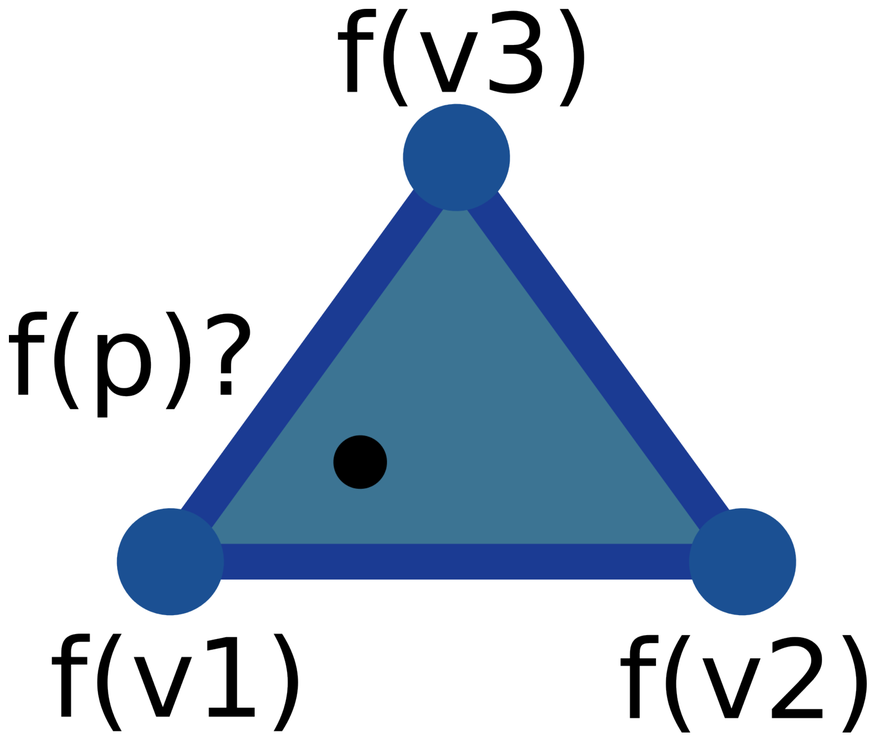
- $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
- $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
- $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$



- Automatically interpolates on the edges
- Similar reasoning for d-simplex
- Can be used to determine if a point lies within a simplex

Barycentric coordinates

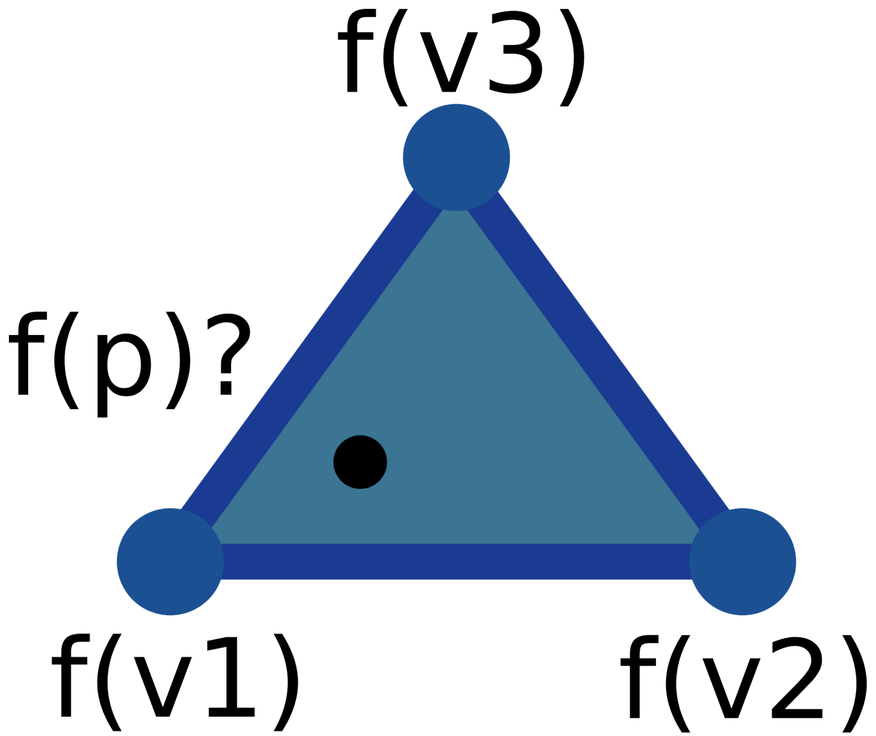
- $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
- $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
- $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$



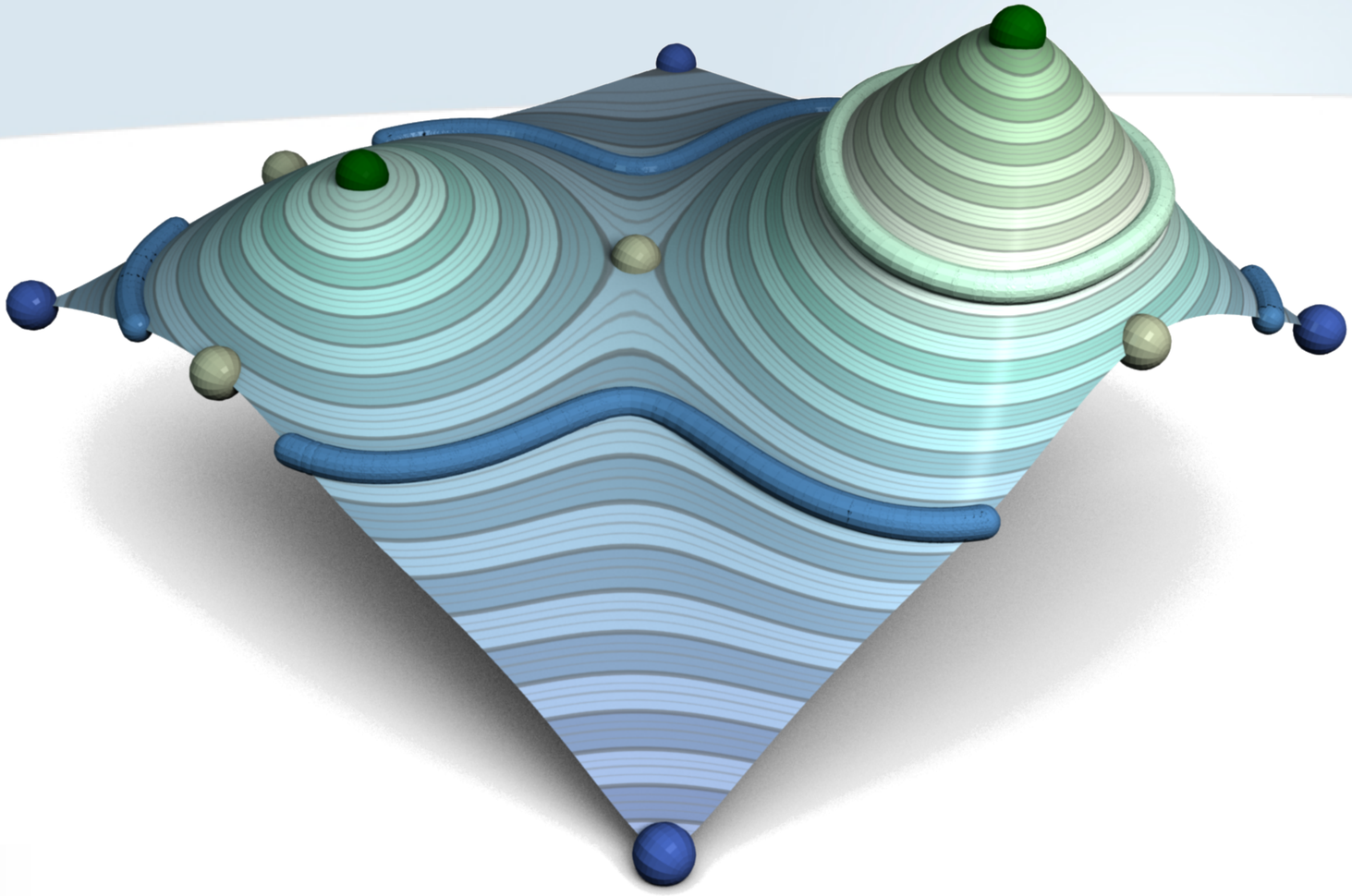
- Automatically interpolates on the edges
- Similar reasoning for d-simplex
- Can be used to determine if a point lies within a simplex $(\alpha_1, \alpha_2, \alpha_3) \in [0,1] \times [0,1] \times [0,1]$

Barycentric coordinates

- $f(p) = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \alpha_3 f(v_3)$
- $\alpha_1 + \alpha_2 + \alpha_3 = 1$
- $x(p) = \alpha_1 x(v_1) + \alpha_2 x(v_2) + \alpha_3 x(v_3)$
- $y(p) = \alpha_1 y(v_1) + \alpha_2 y(v_2) + \alpha_3 y(v_3)$



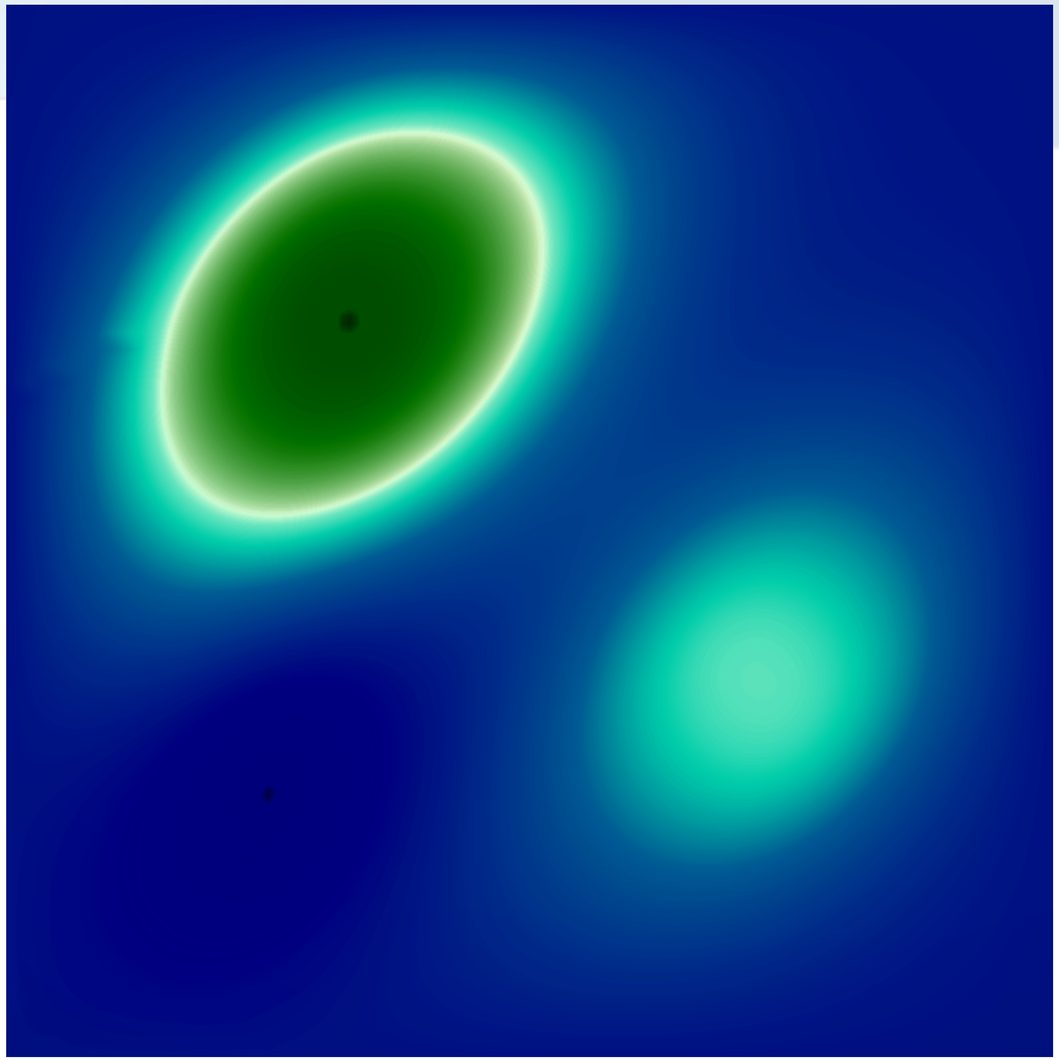
- Automatically interpolates on the edges
- Similar reasoning for d-simplex
- Can be used to determine if a point lies within a simplex $(\alpha_1, \alpha_2, \alpha_3) \in [0,1] \times [0,1] \times [0,1]$
- No critical point in the interior!



Scalar Field Visualization

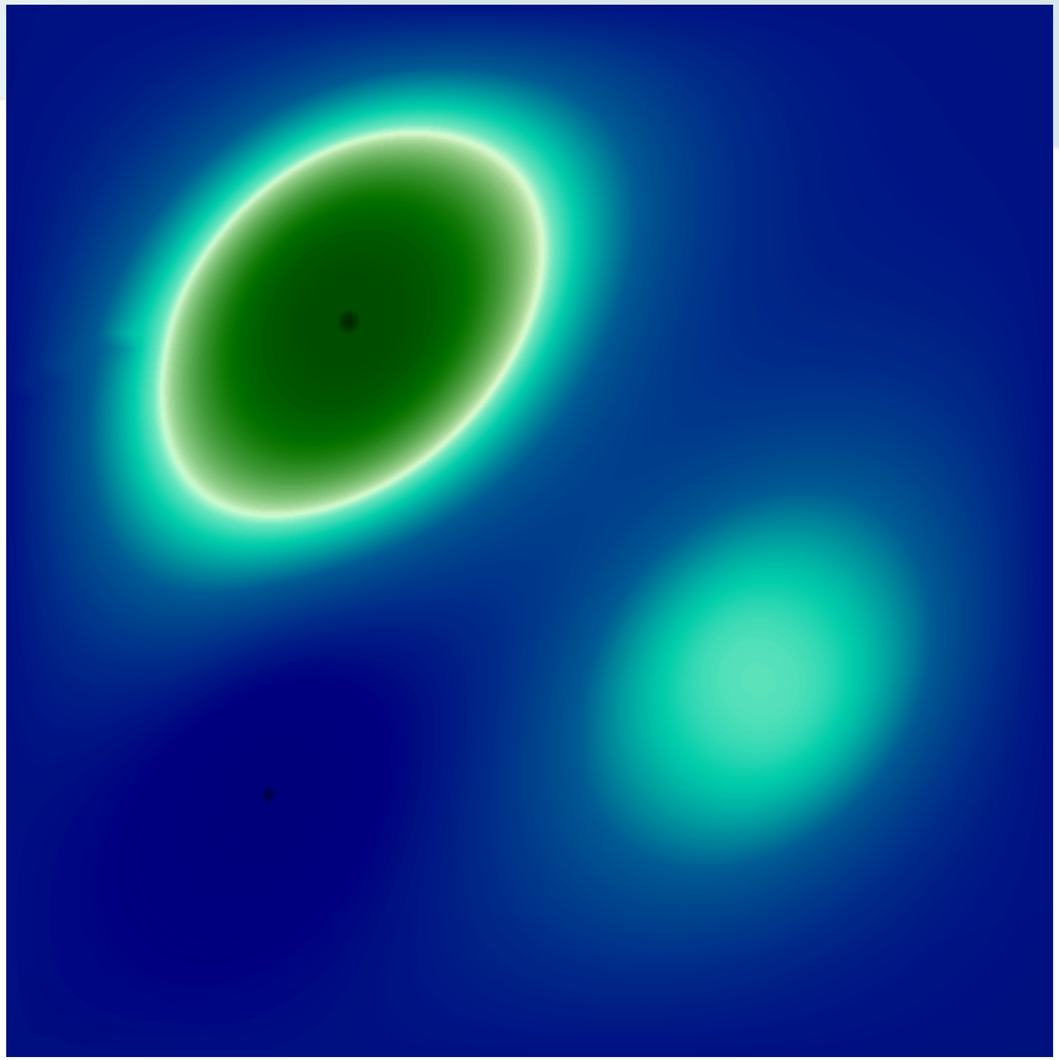
Scalar field geometry

- Continuous color maps



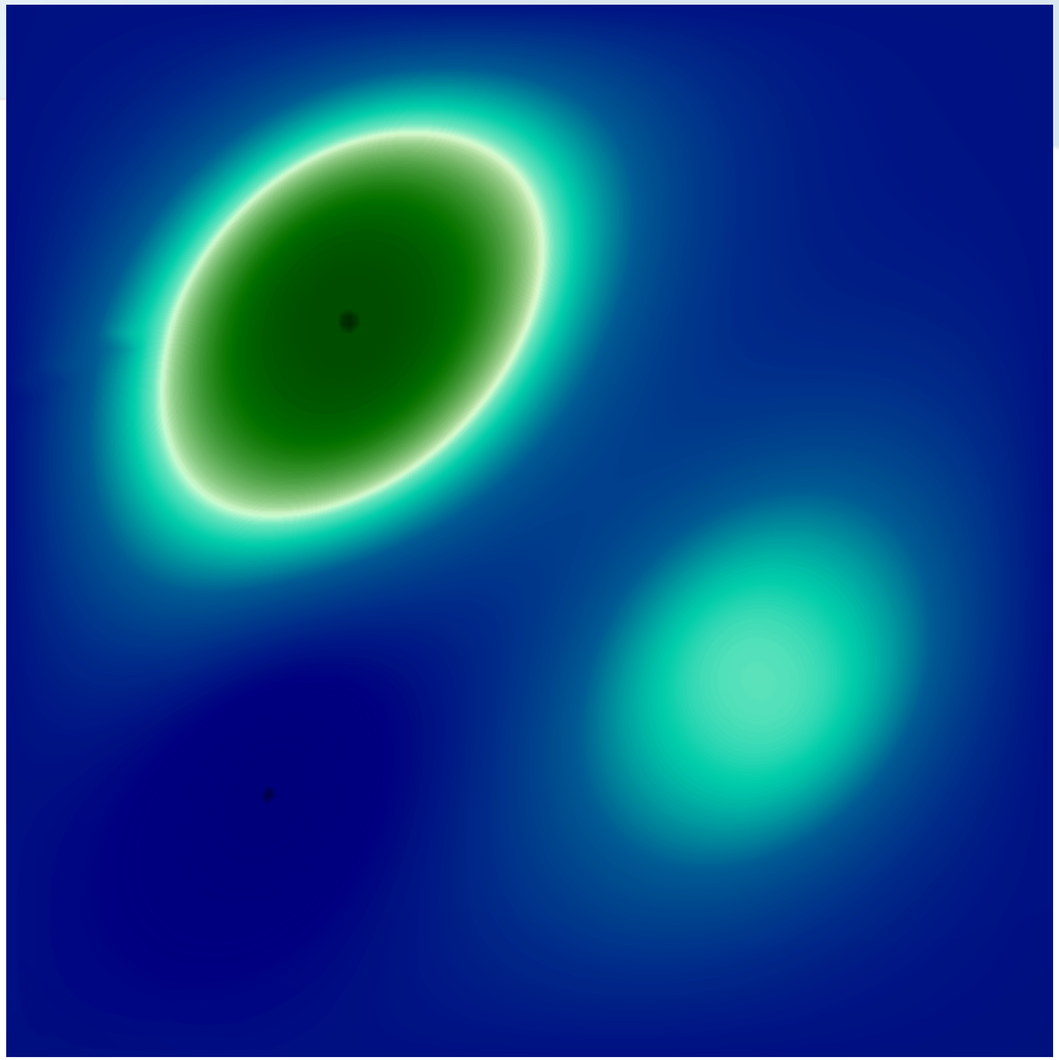
Scalar field geometry

- Continuous color maps
  - Difficulty to estimate the geometry



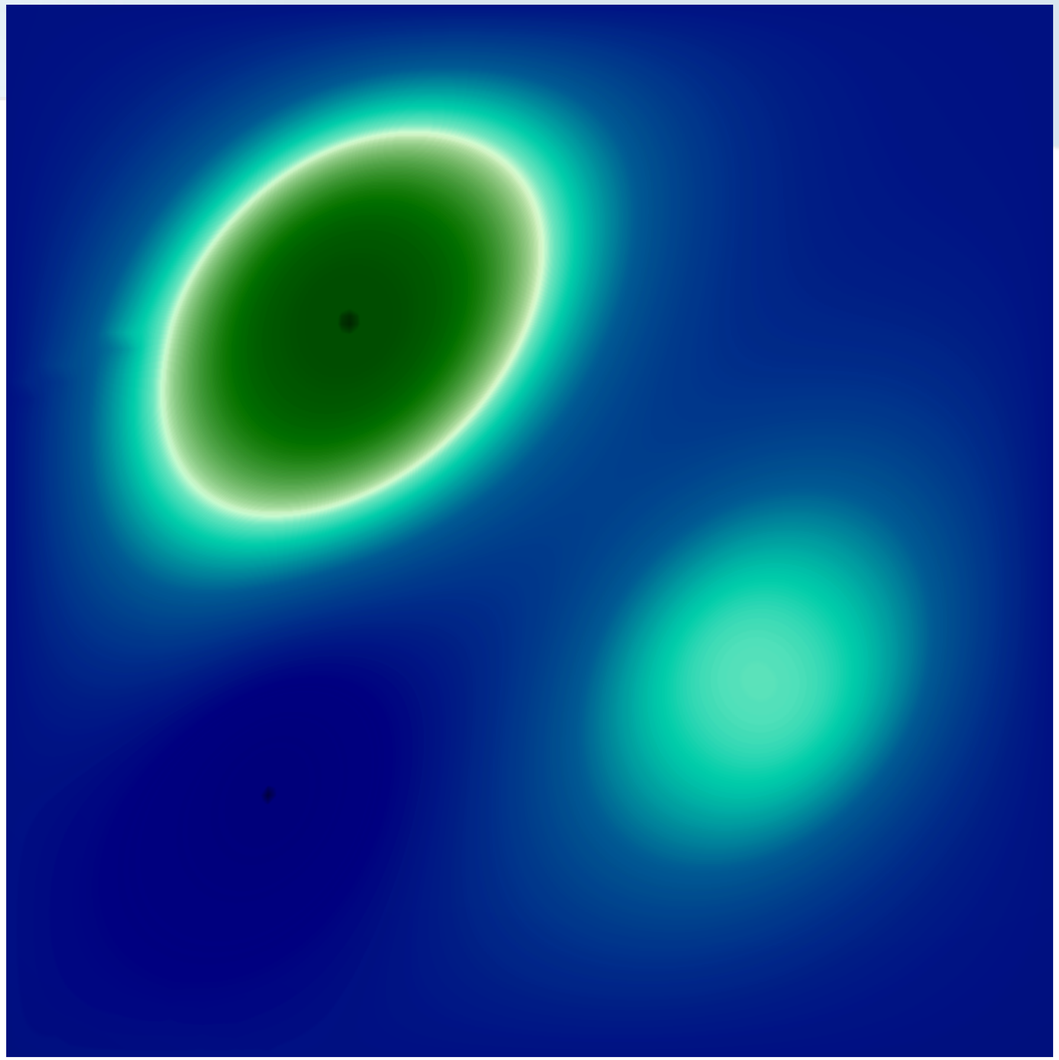
Scalar field geometry

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



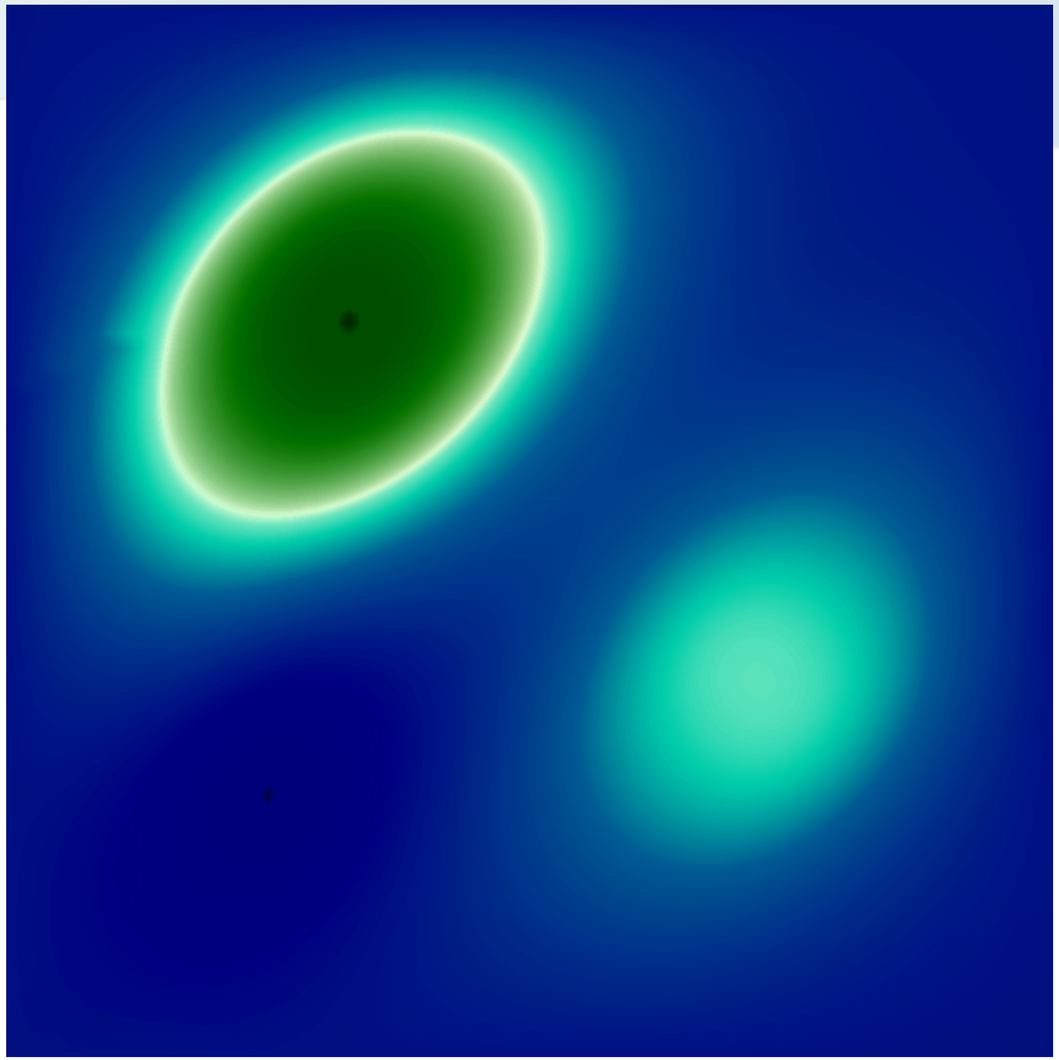
Scalar field geometry

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



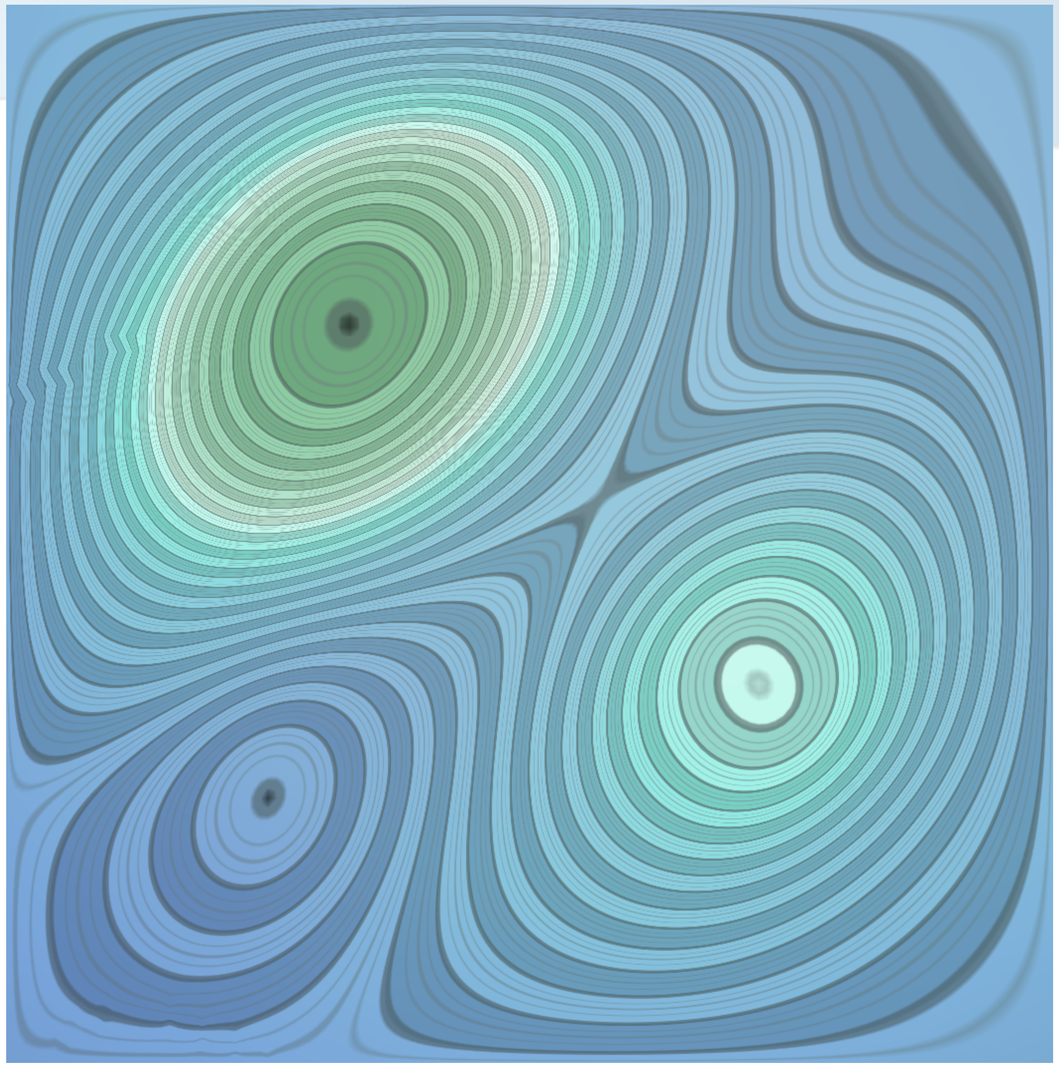
Scalar field geometry

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



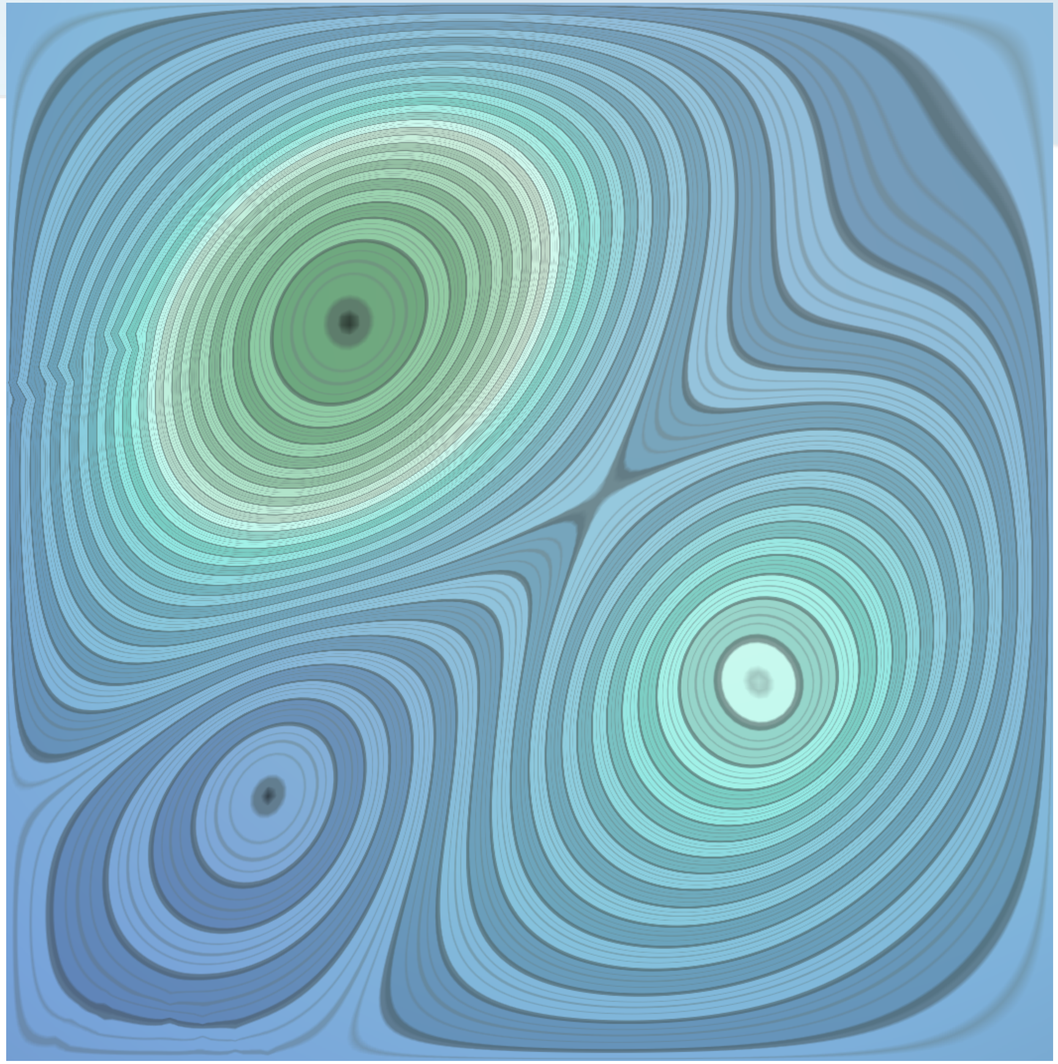
Scalar field geometry

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



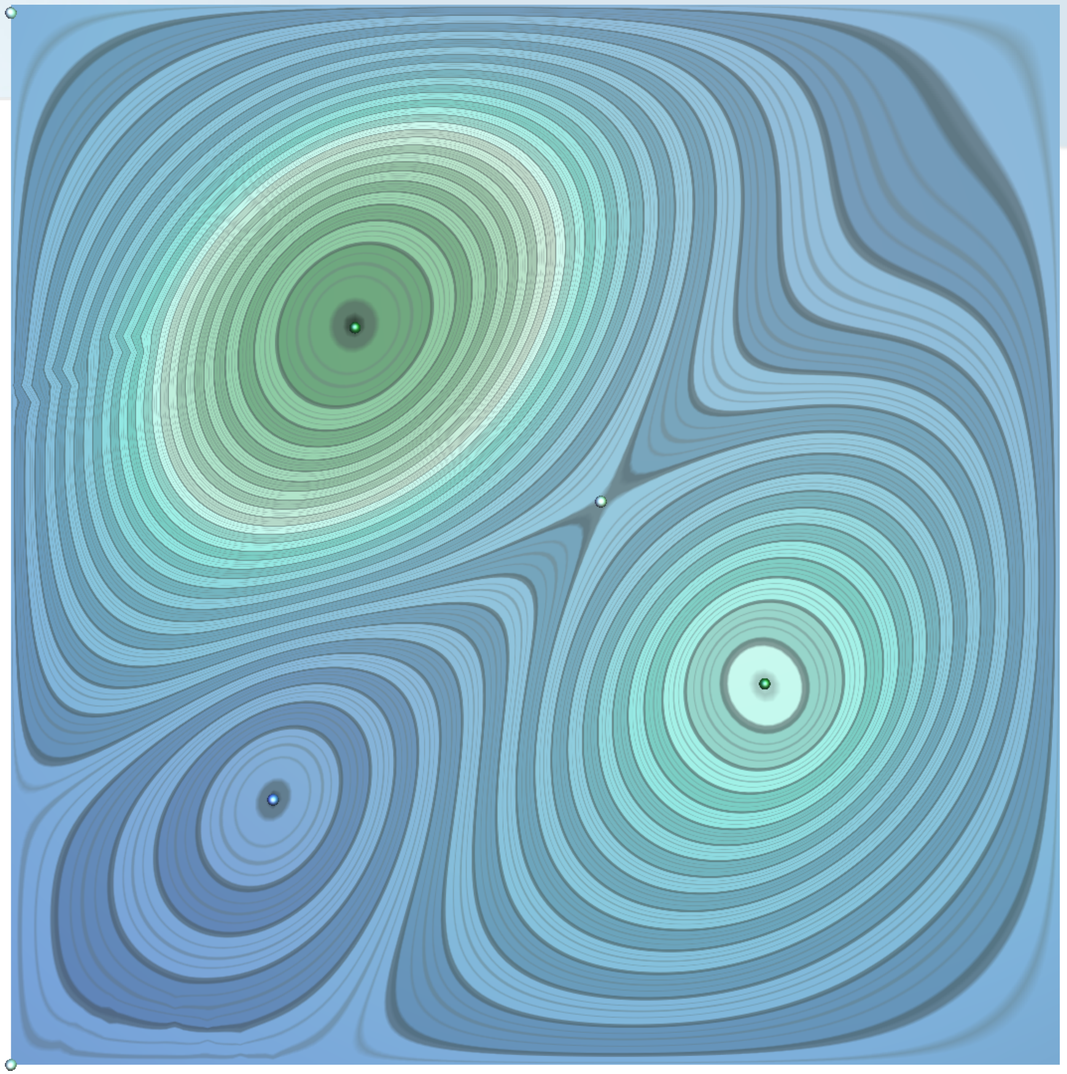
Scalar field geometry

- 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

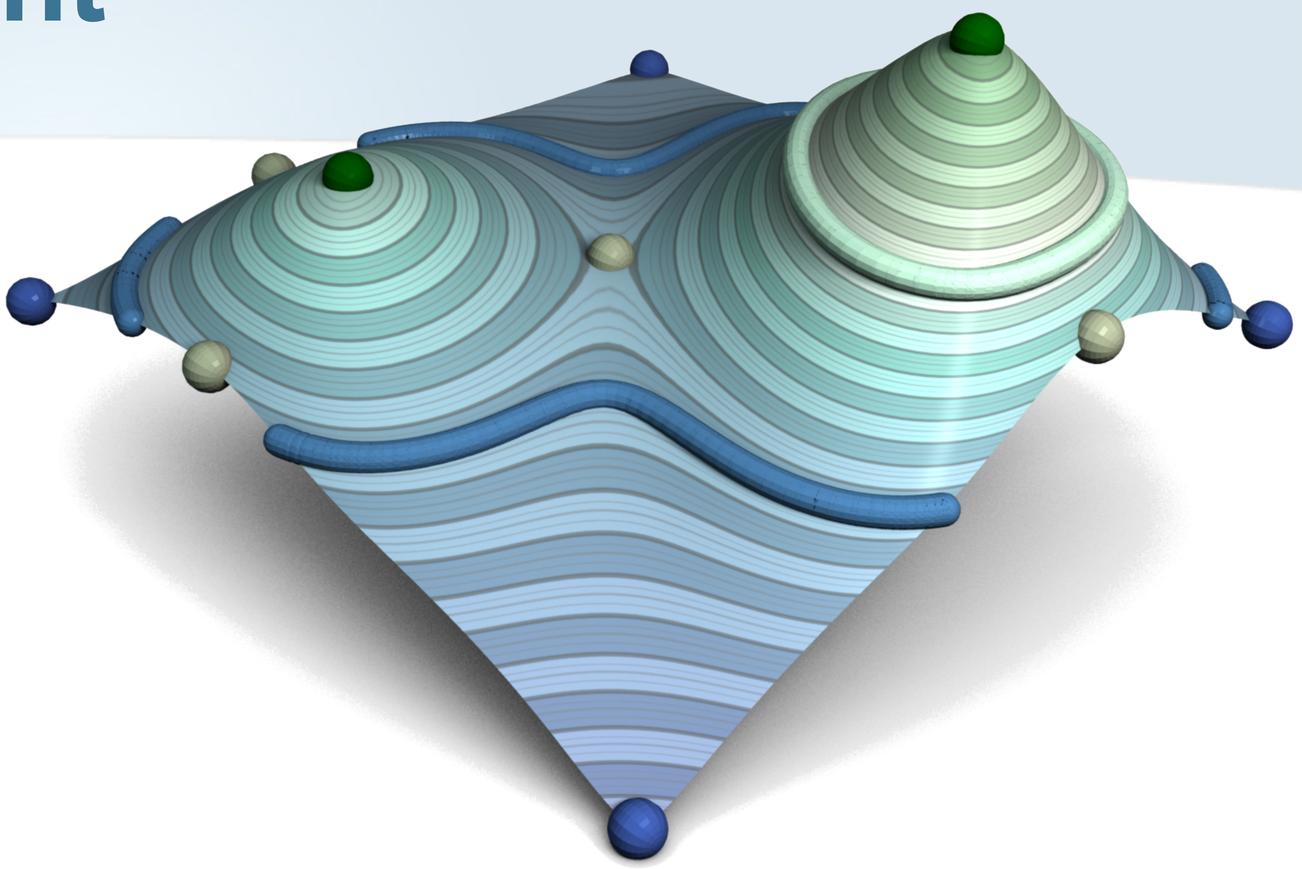


Scalar field geometry

- 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

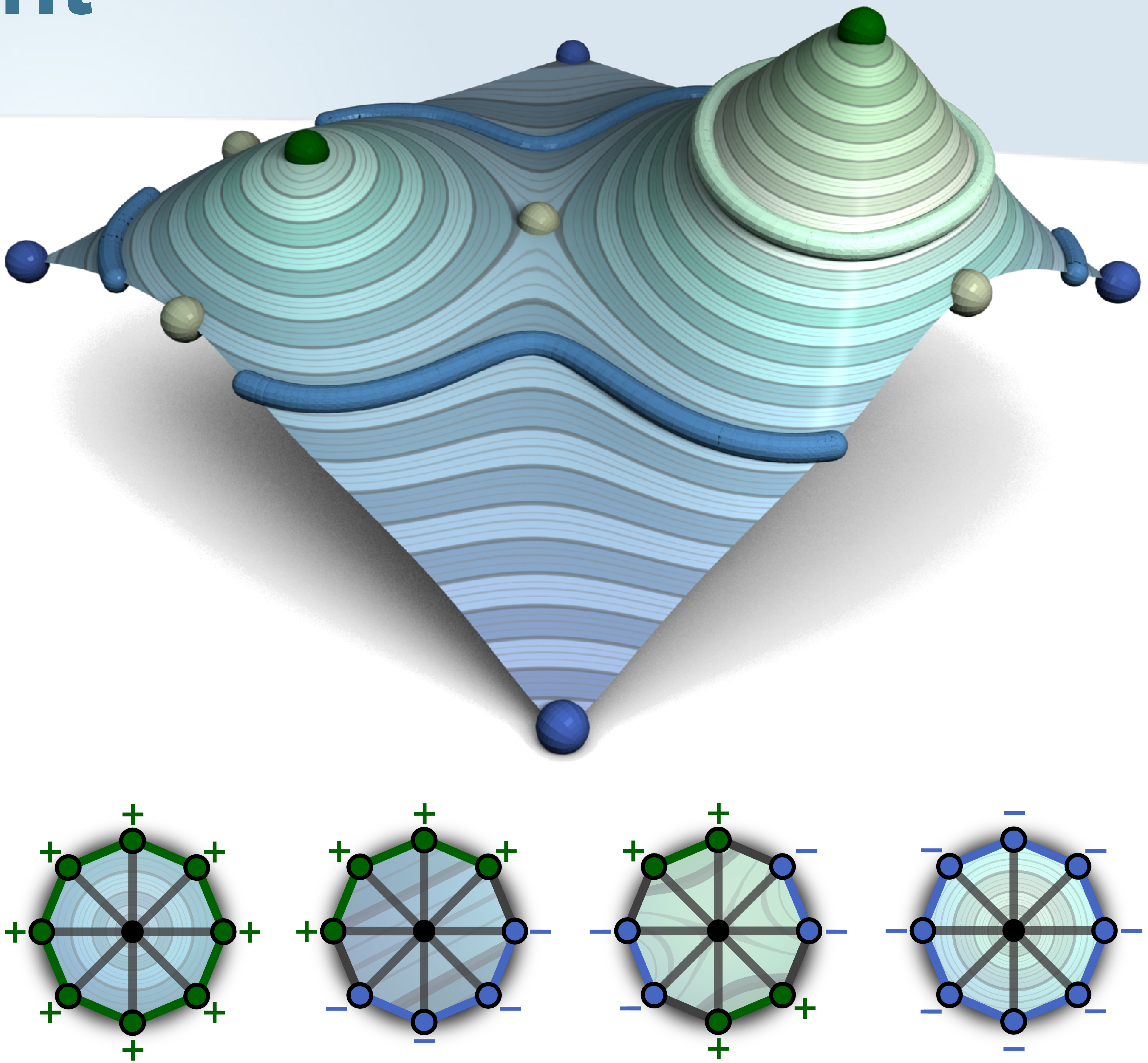


Notion of critical point



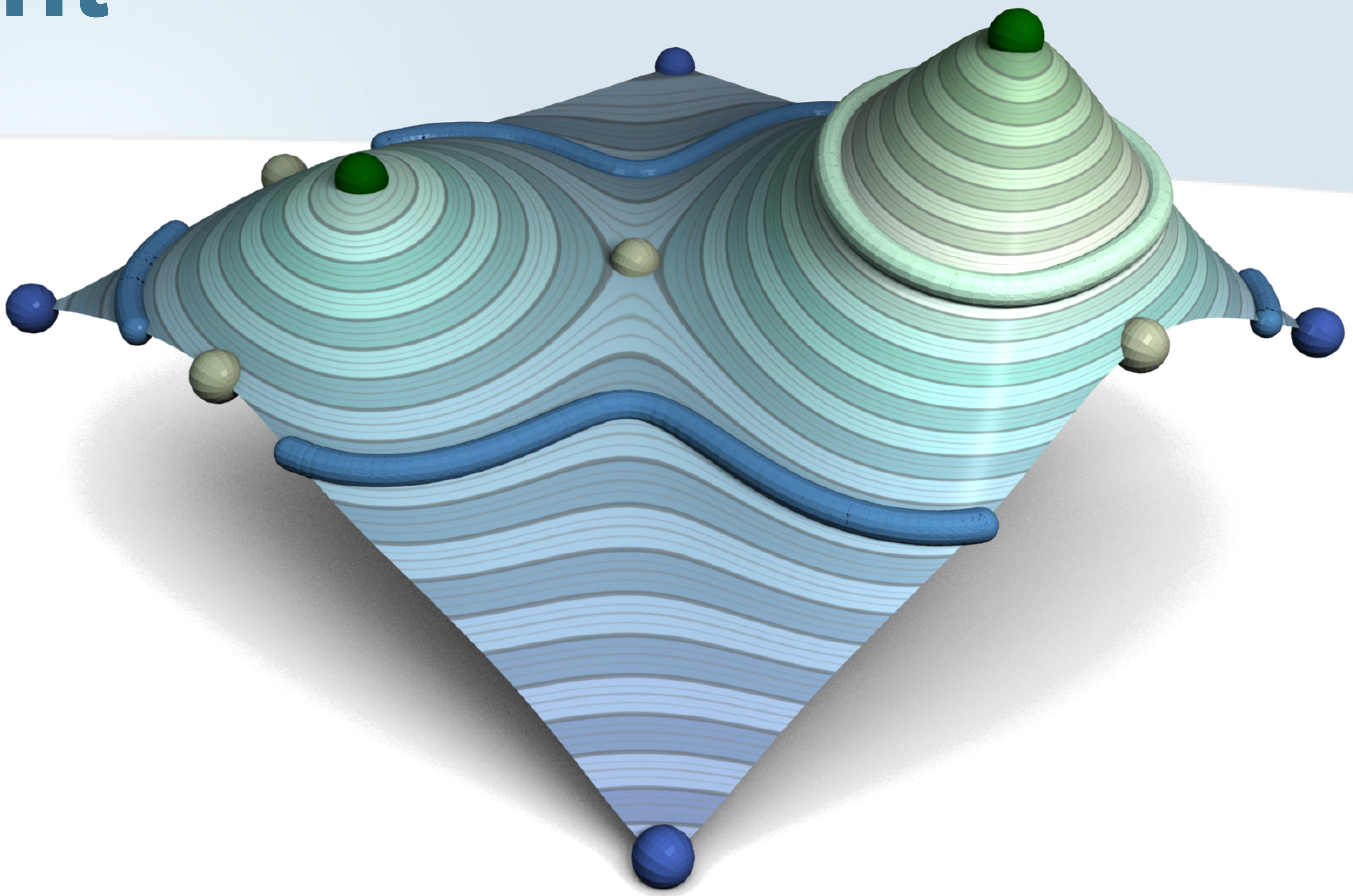
Notion of critical point

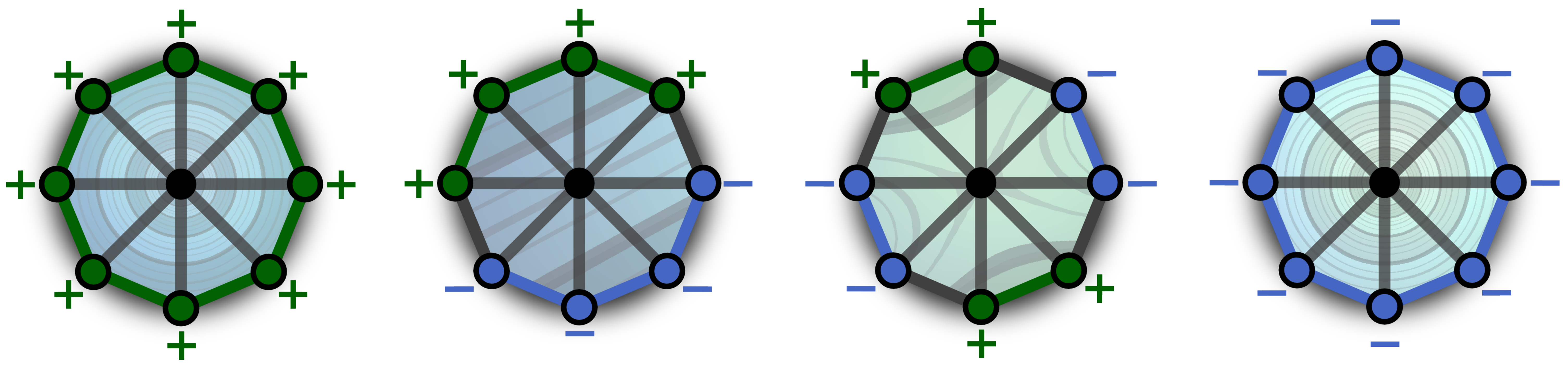
- Combinatorial identification



Notion of critical point

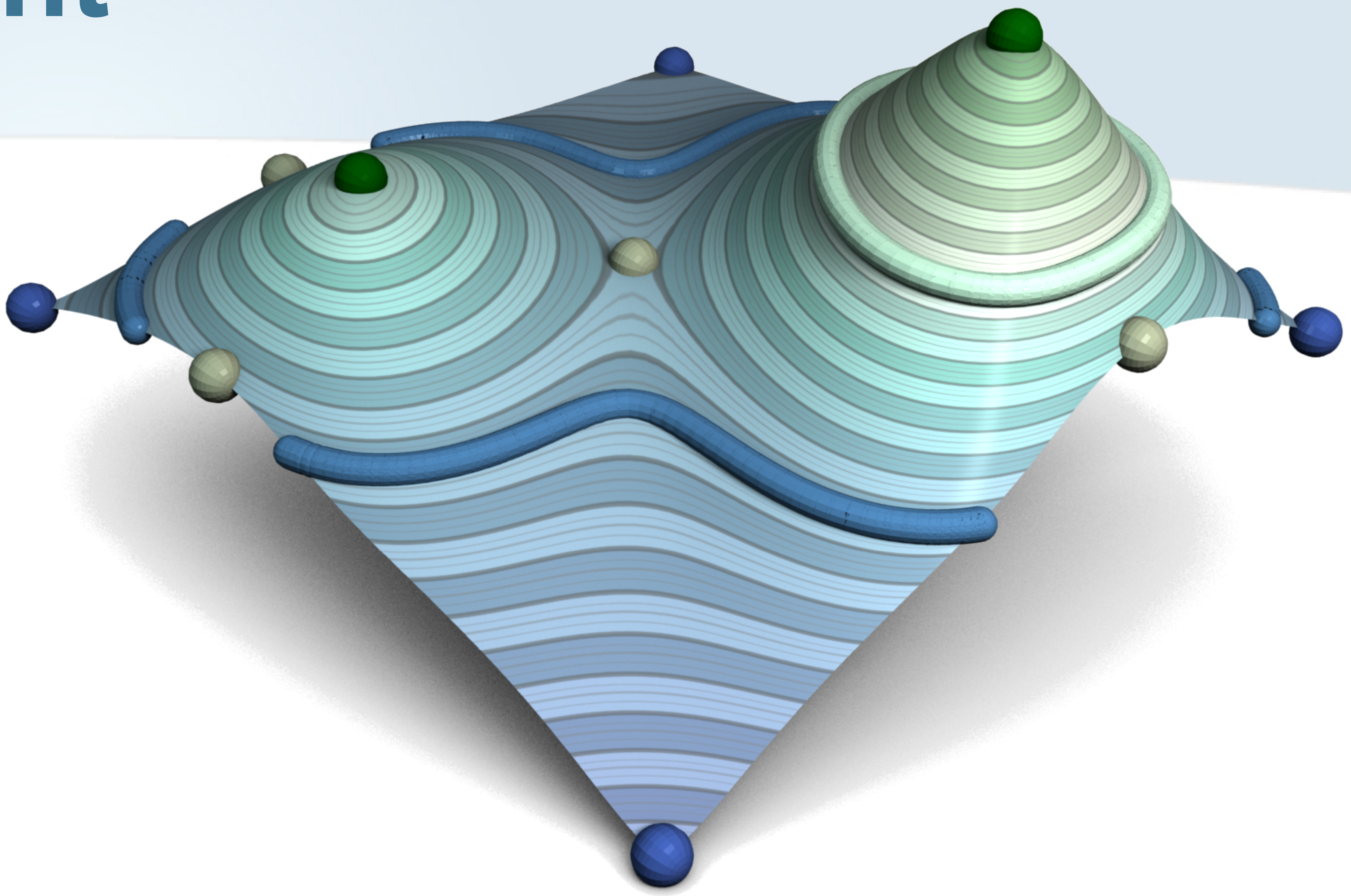
- Combinatorial identification
  - Star of a simplex σ

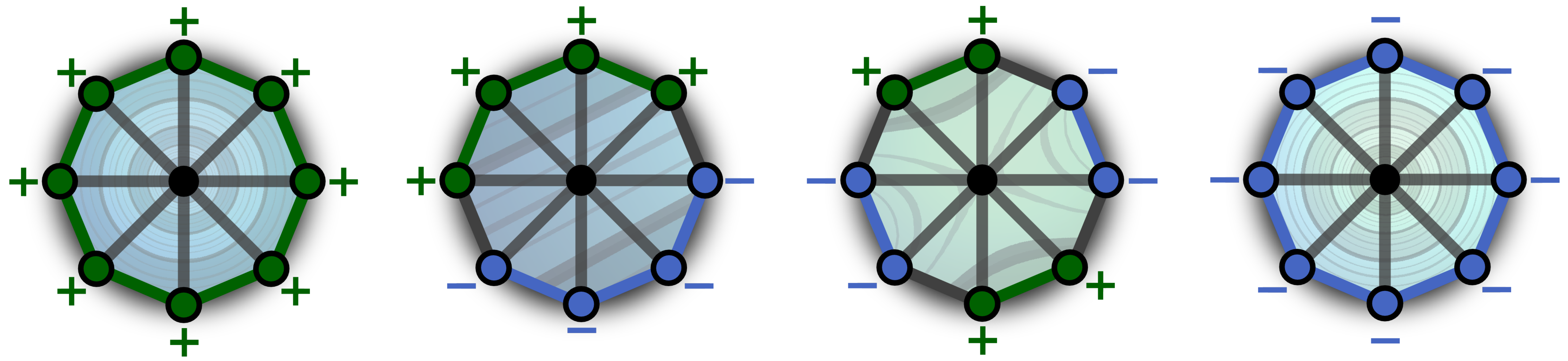




Notion of critical point

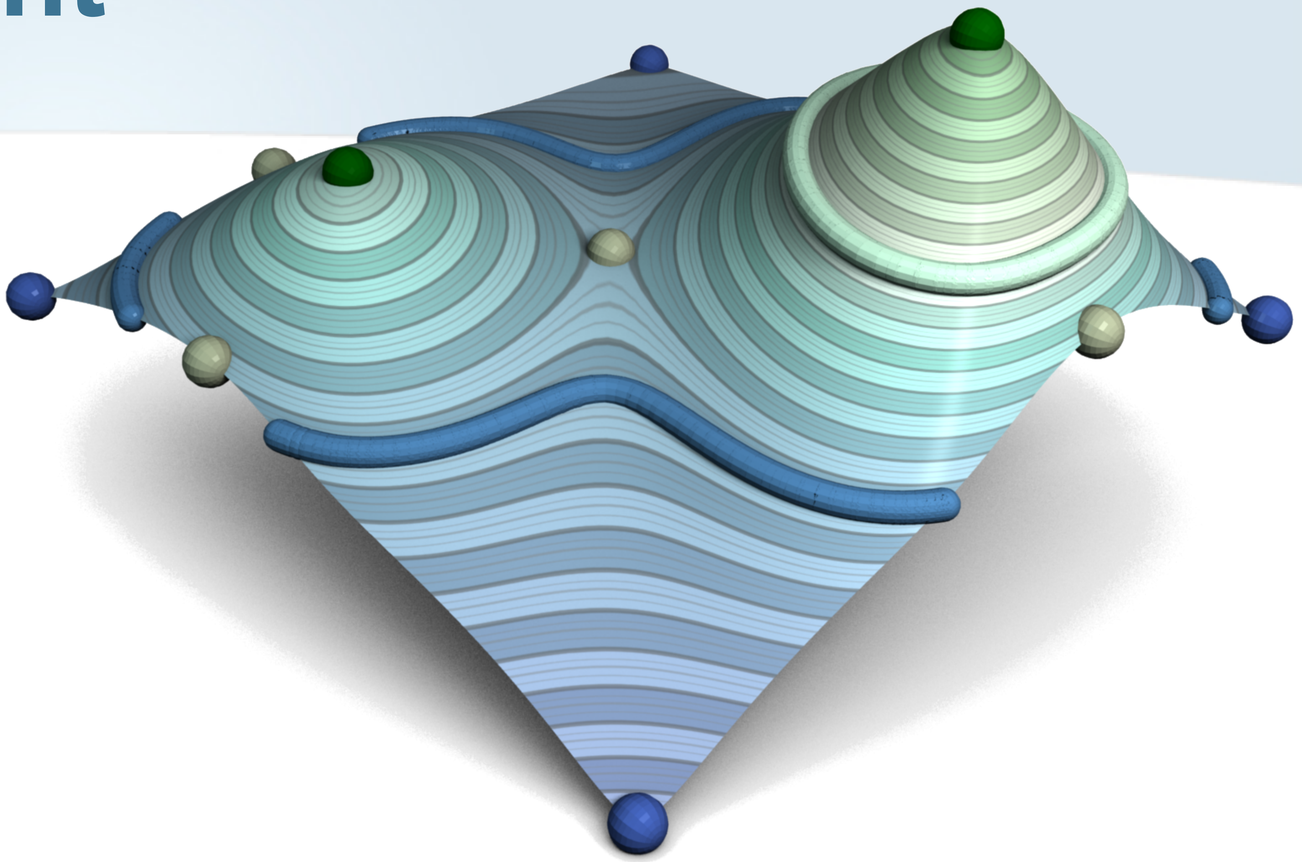
- Combinatorial identification
  - Star of a simplex σ
    - Simplices that contain σ as a face

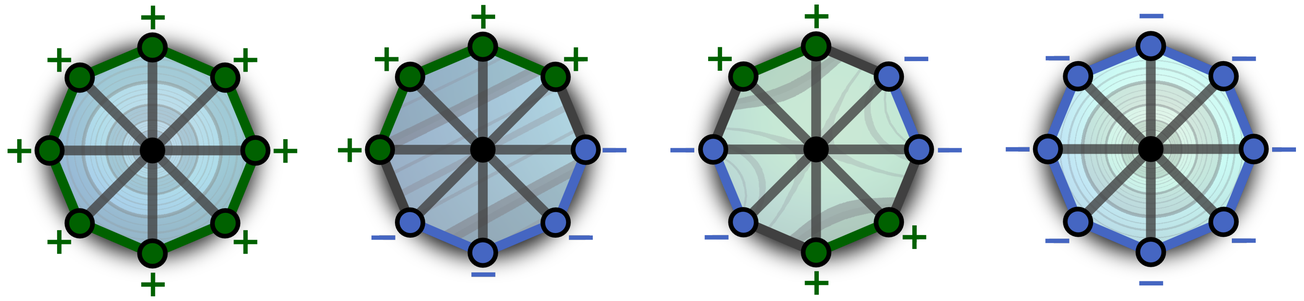




Notion of critical point

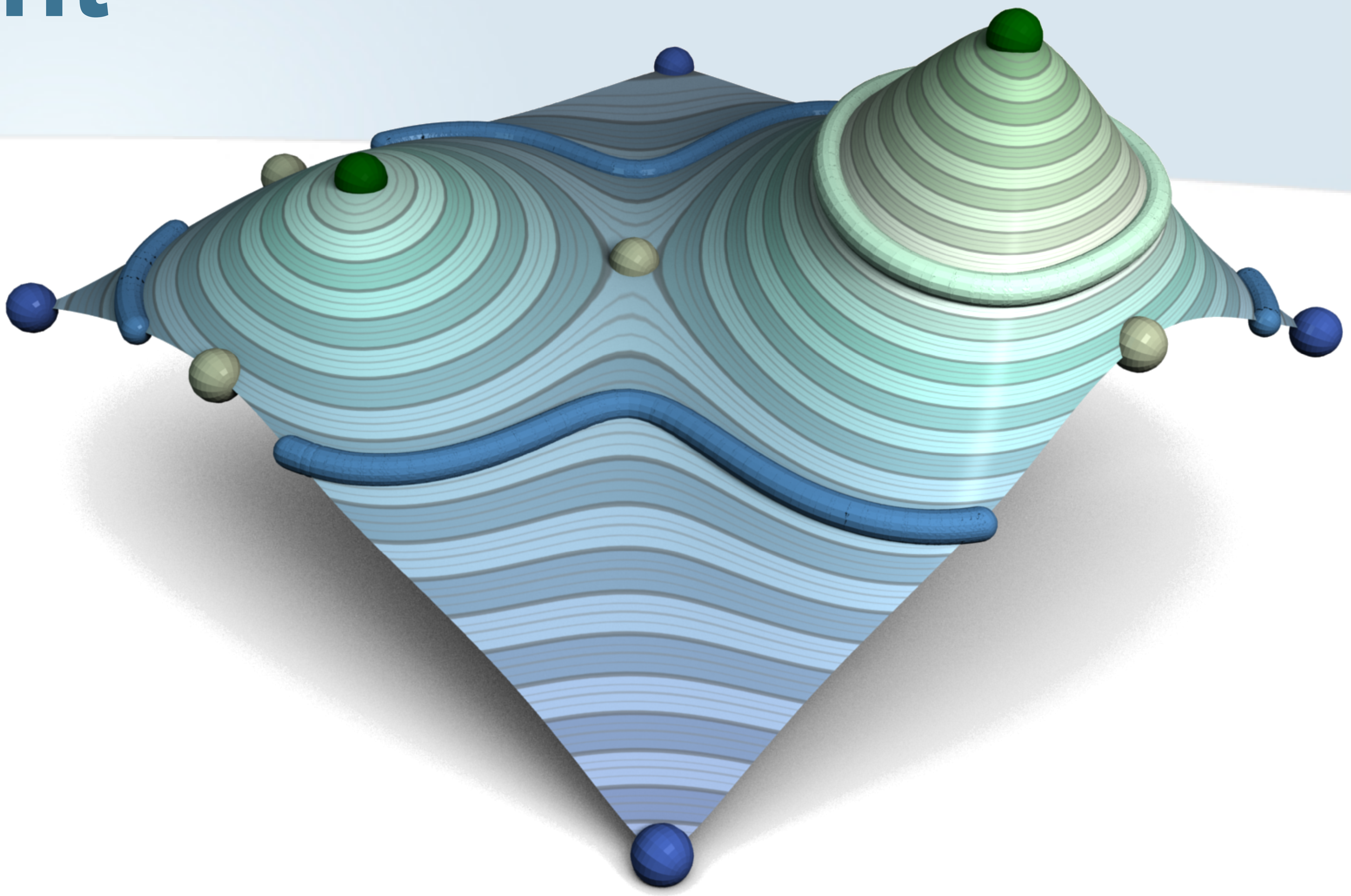
- Combinatorial identification
  - Star of a simplex σ
    - Simplices that contain σ as a face
  - Link of a simplex

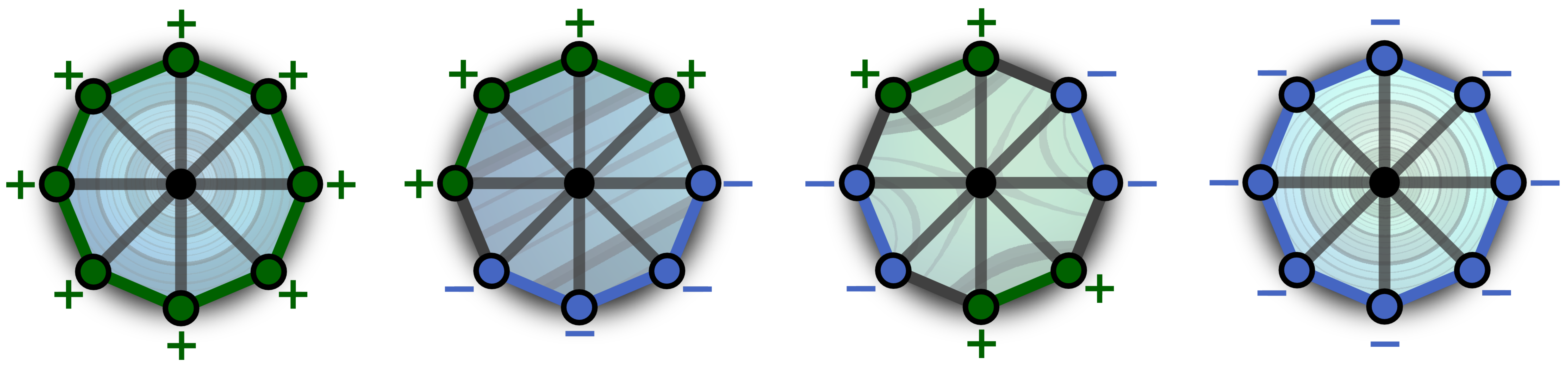




Notion of critical point

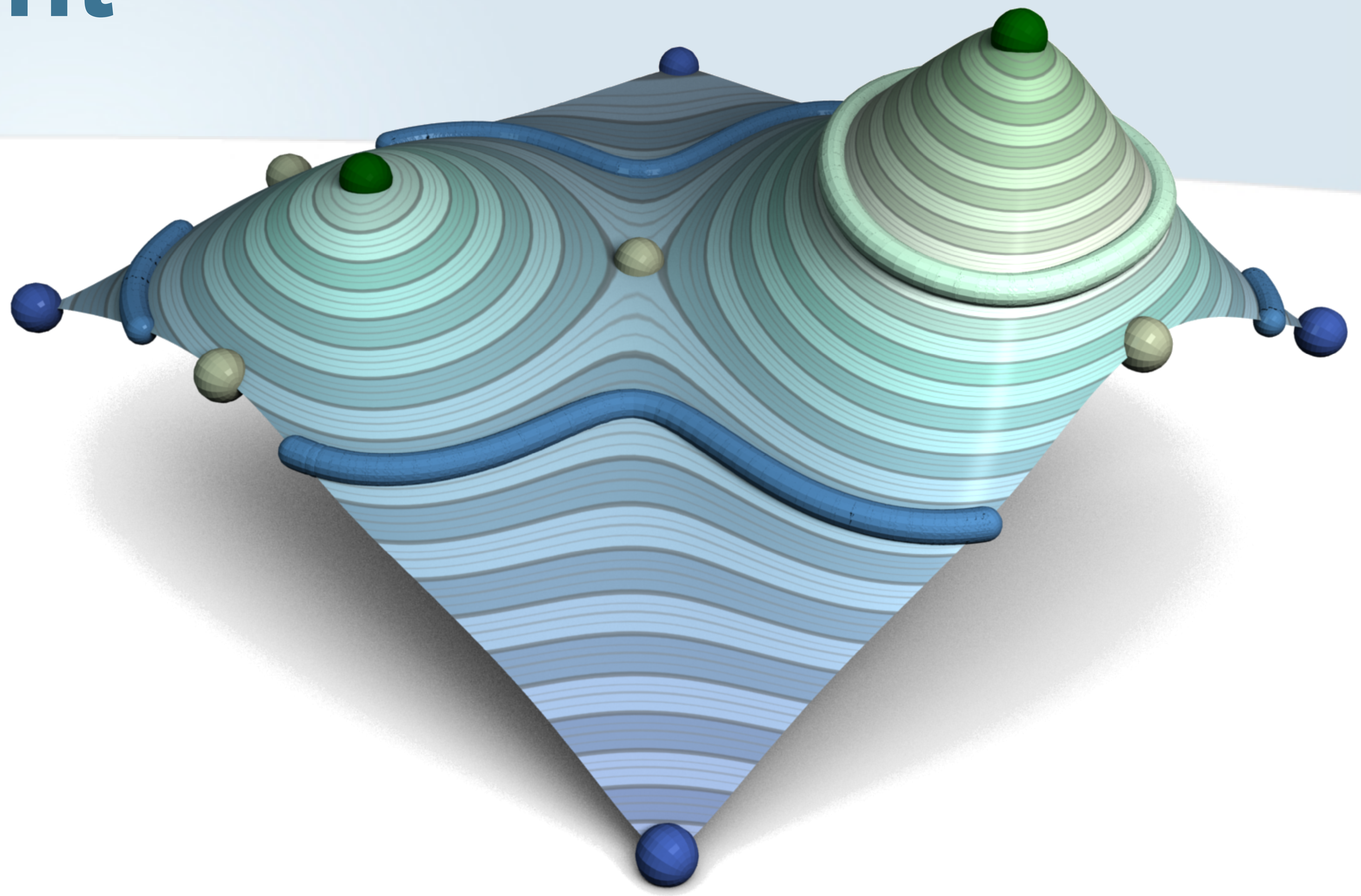
- Combinatorial identification
  - Star of a simplex σ
    - Simplices that contain σ as a face
  - Link of a simplex
    - Simplices of the closure of the star that do not contain σ as a face

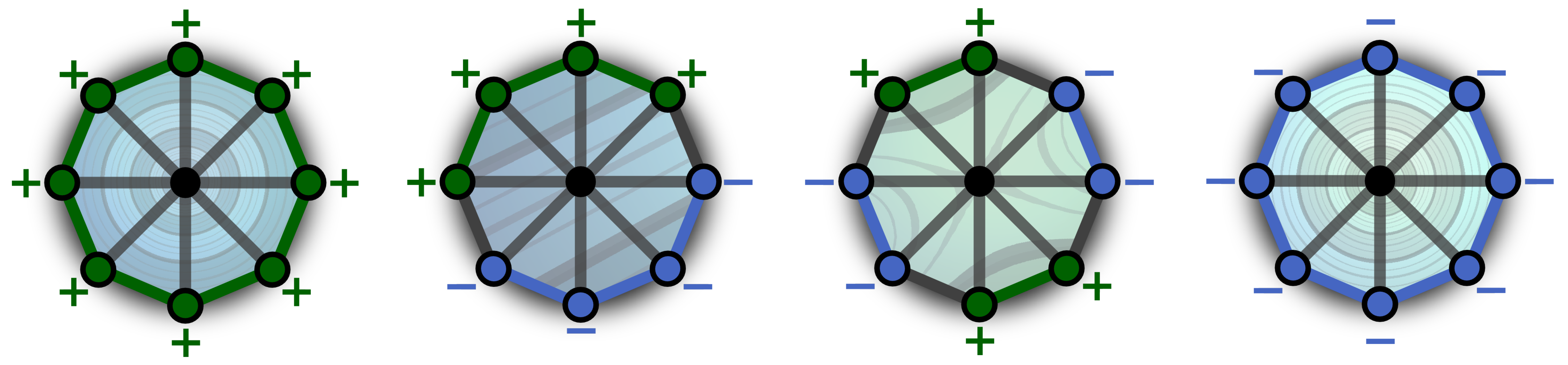




Notion of critical point

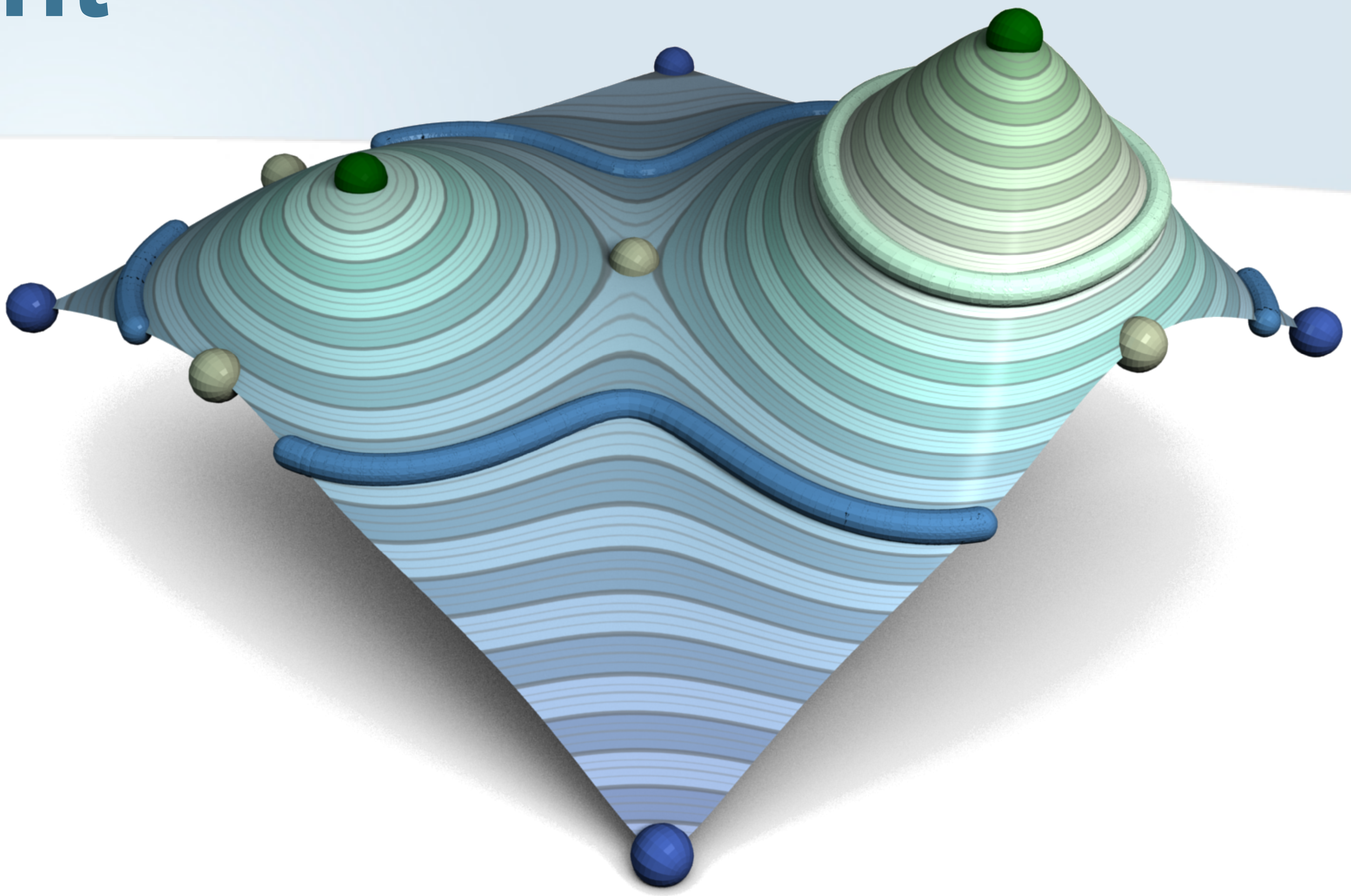
- Combinatorial identification
  - Lower link of v

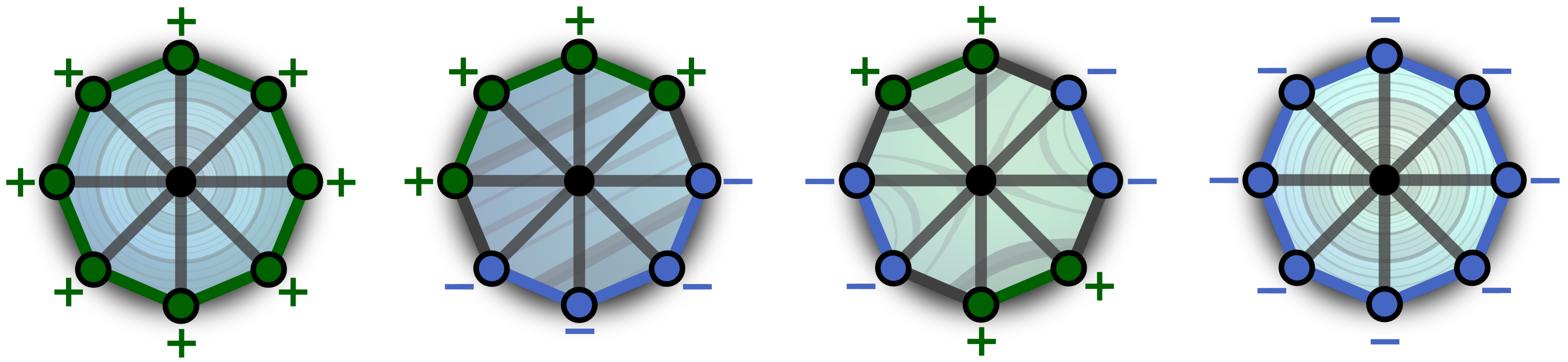




Notion of critical point

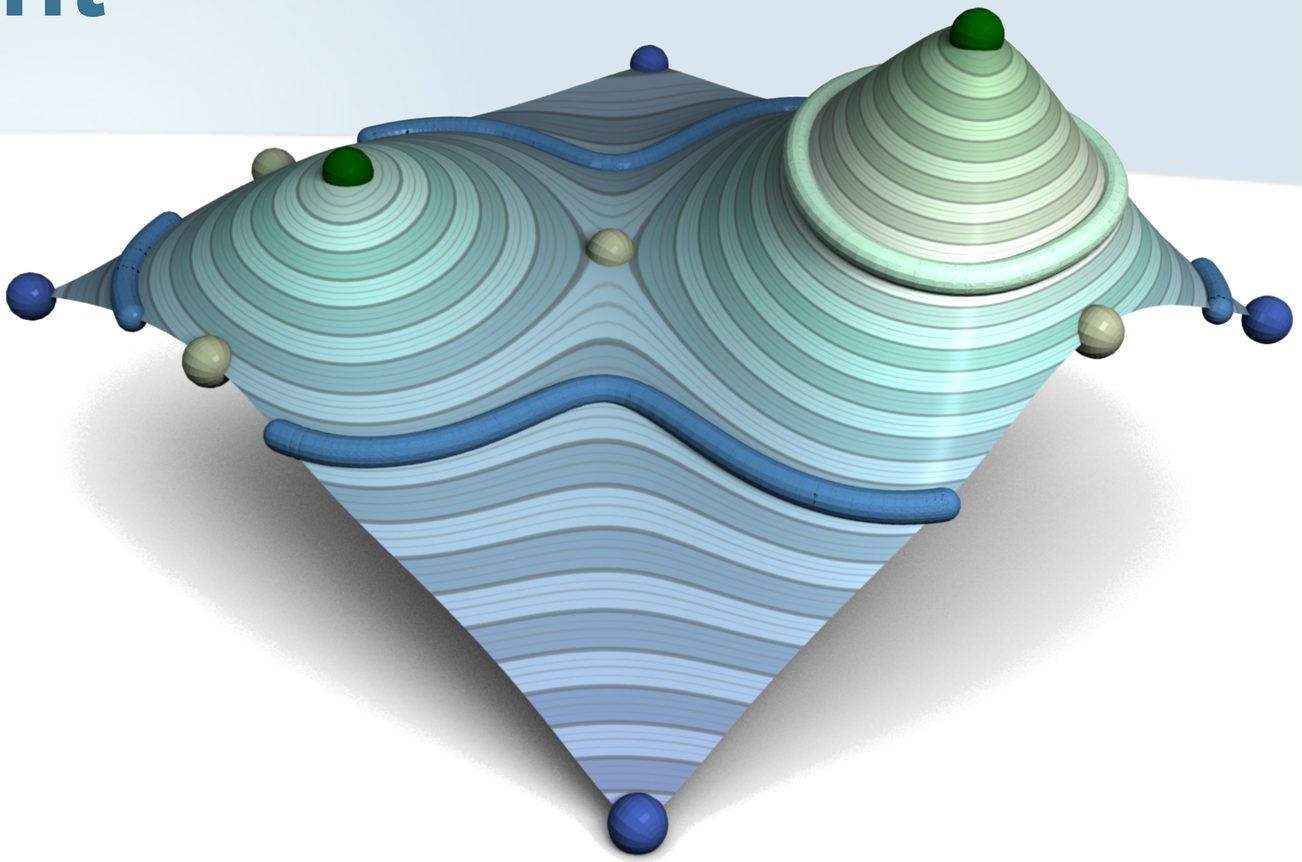
- Combinatorial identification
  - Lower link of v
    - Simplices whose function values are strictly below $f(v)$

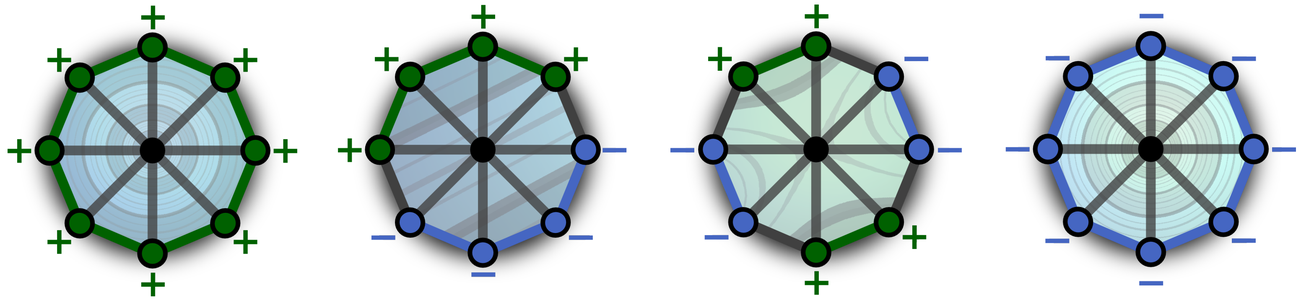




Notion of critical point

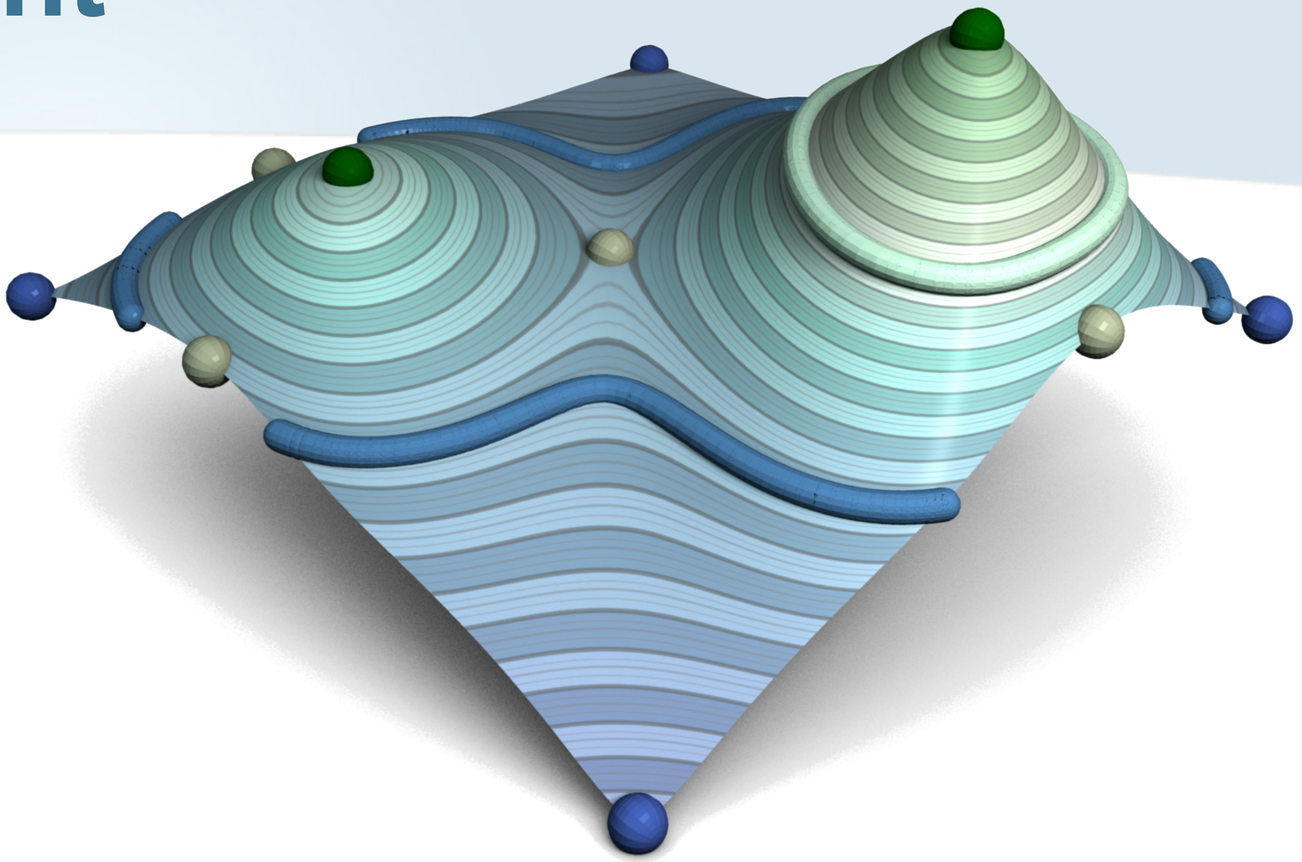
- Combinatorial identification
  - Lower link of v
    - Simplices whose function values are strictly below $f(v)$
  - Upper link of v

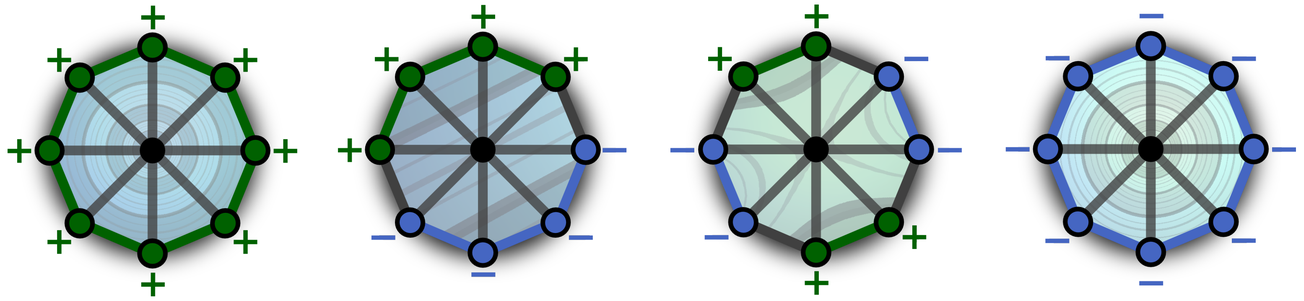




Notion of critical point

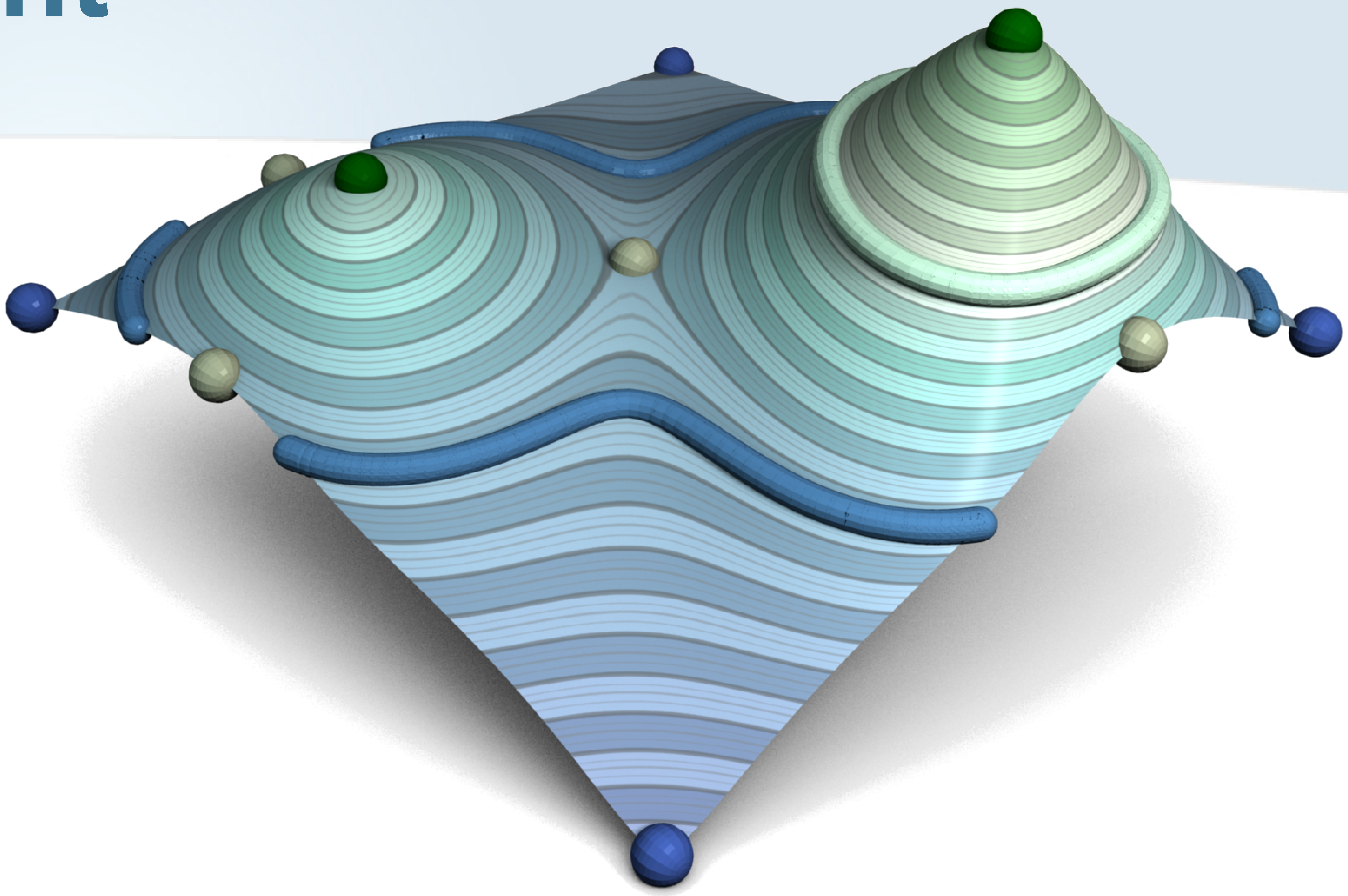
- Combinatorial identification
  - Lower link of v
    - Simplices whose function values are strictly below $f(v)$
  - Upper link of v
    - Simplices whose function values are strictly above $f(v)$

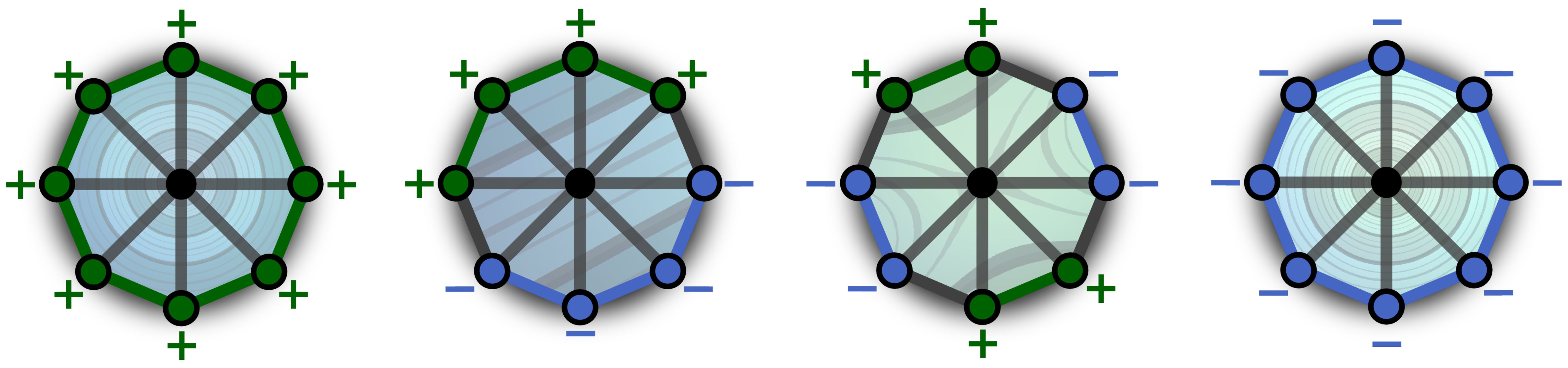




Notion of critical point

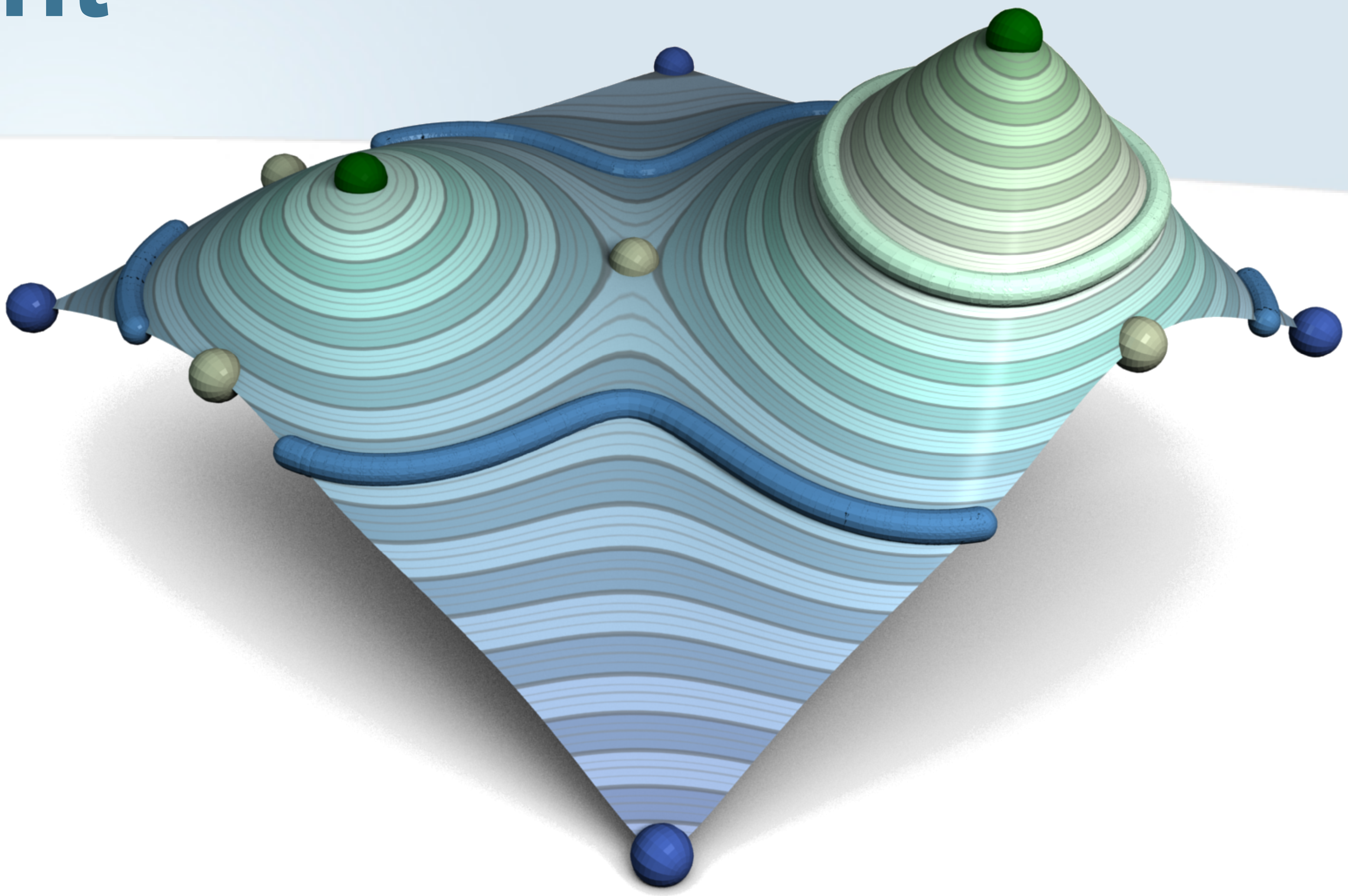
- Combinatorial identification

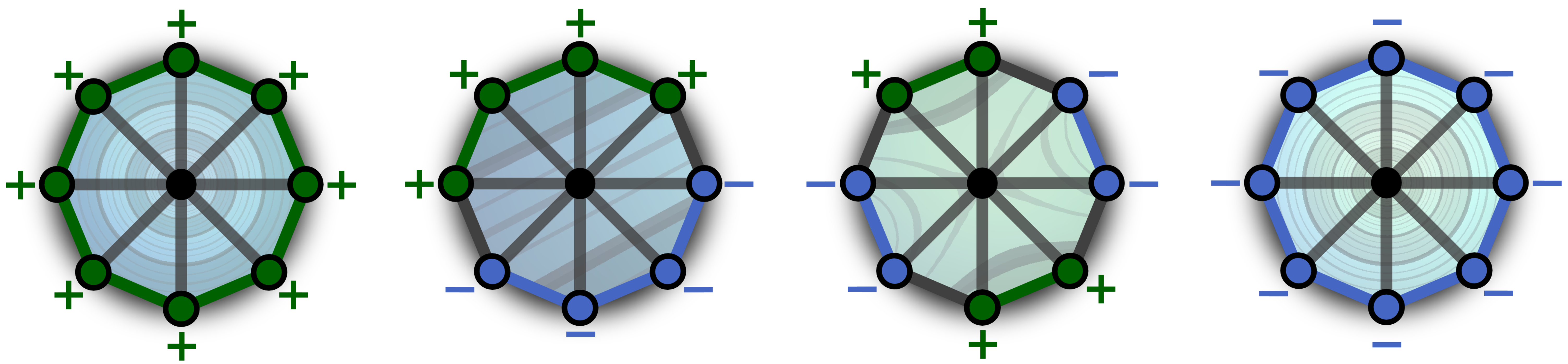




Notion of critical point

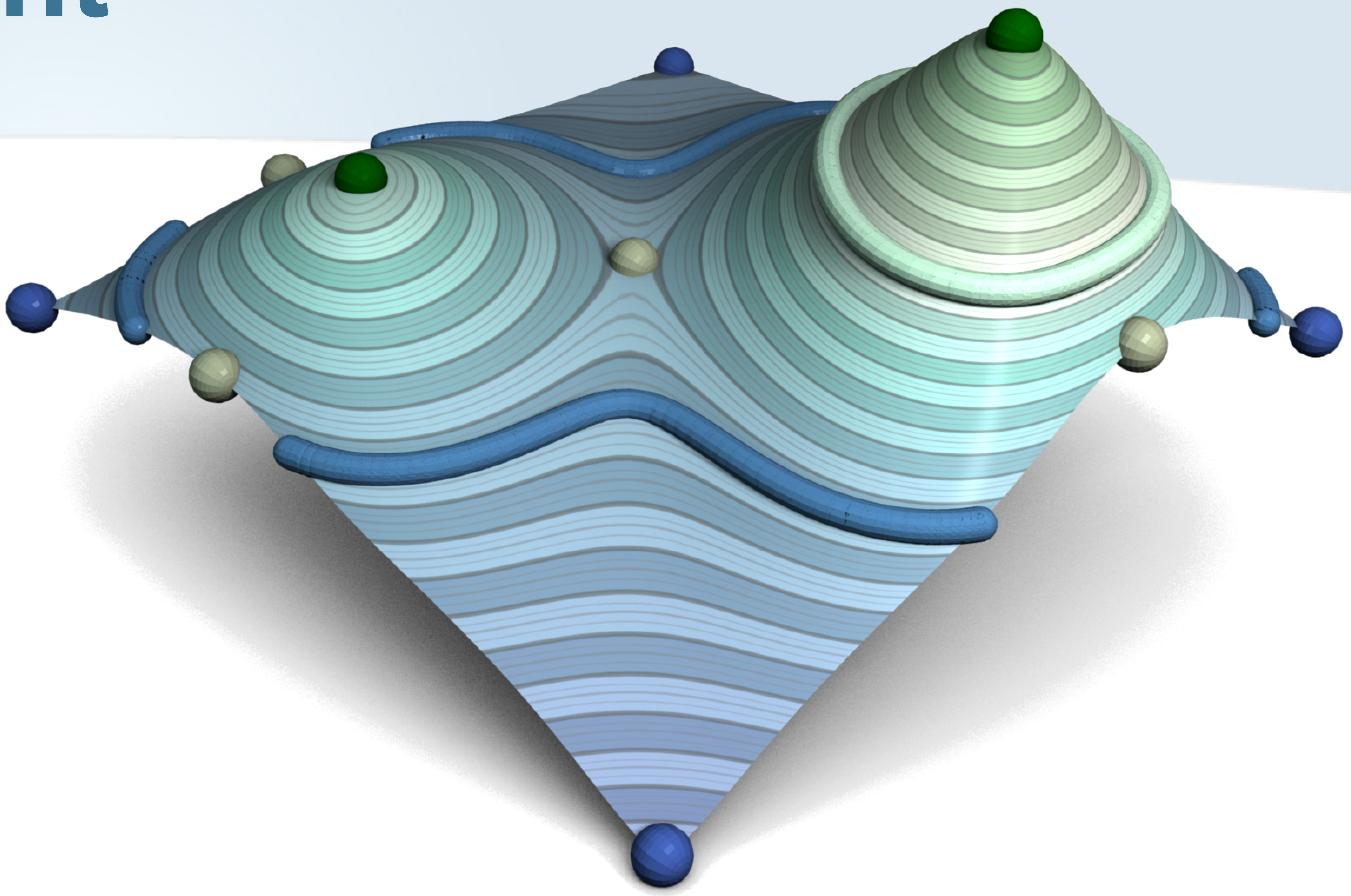
- Combinatorial identification
  - Minimum

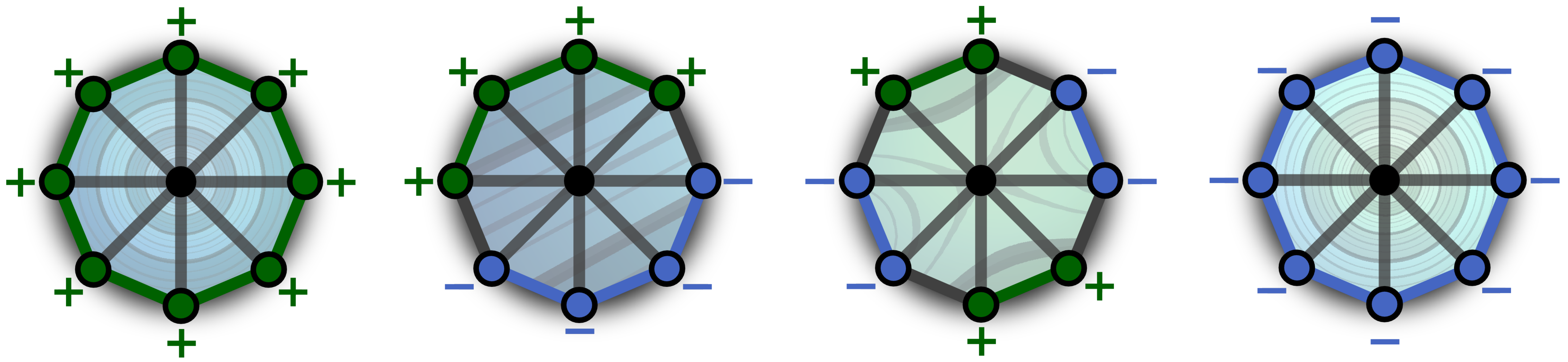




Notion of critical point

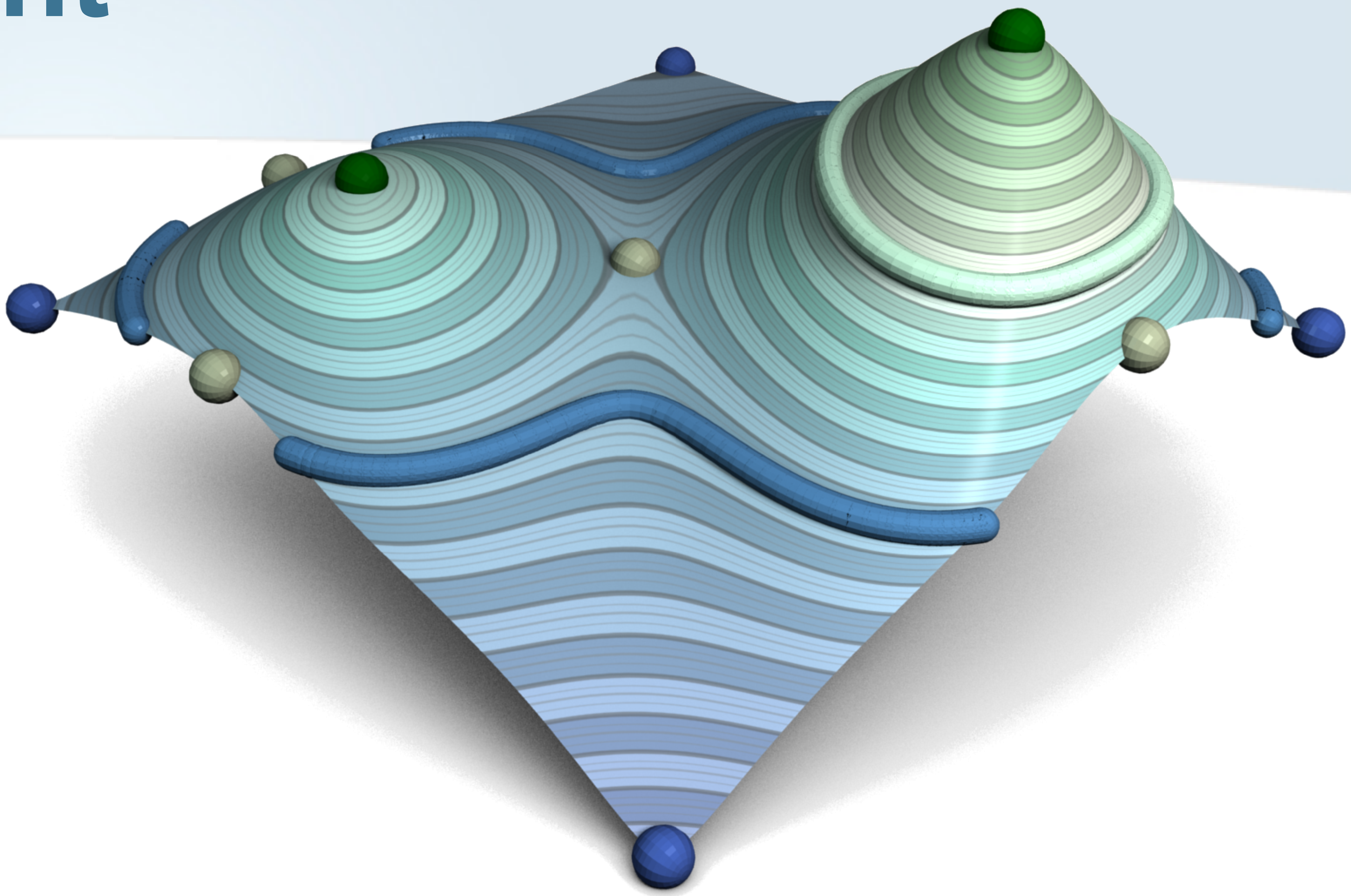
- Combinatorial identification
  - Minimum
    - Empty lower link

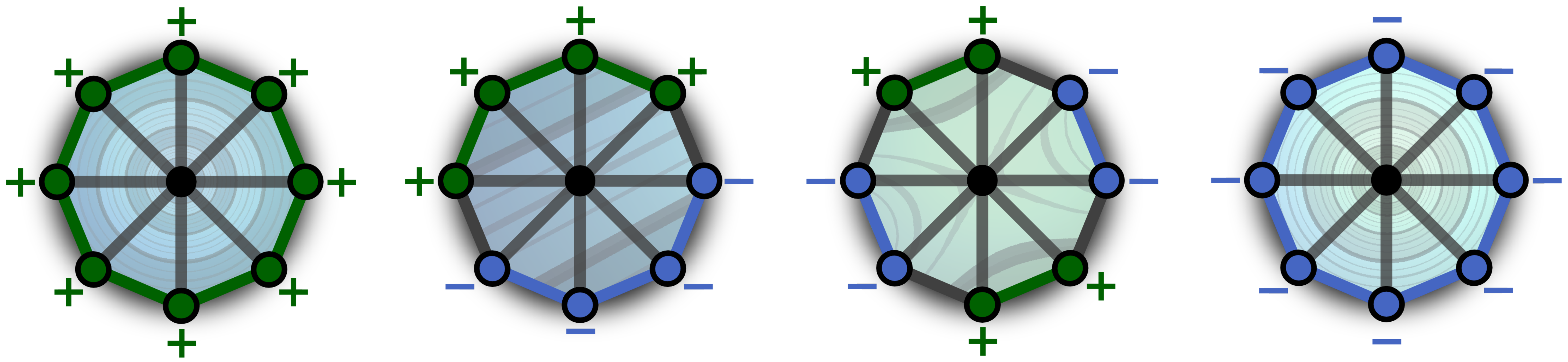




Notion of critical point

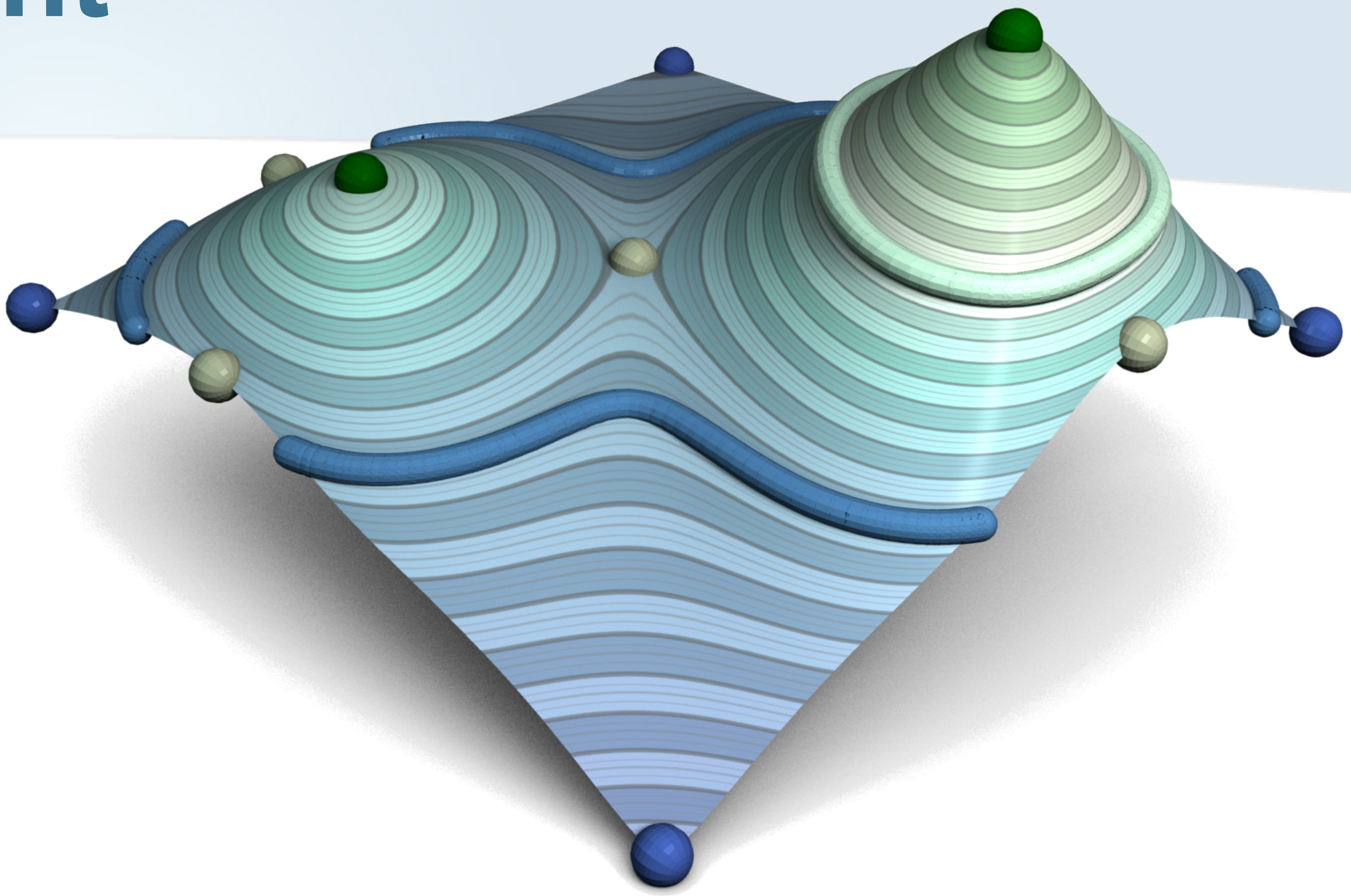
- Combinatorial identification
  - Minimum
    - Empty lower link
  - Maximum

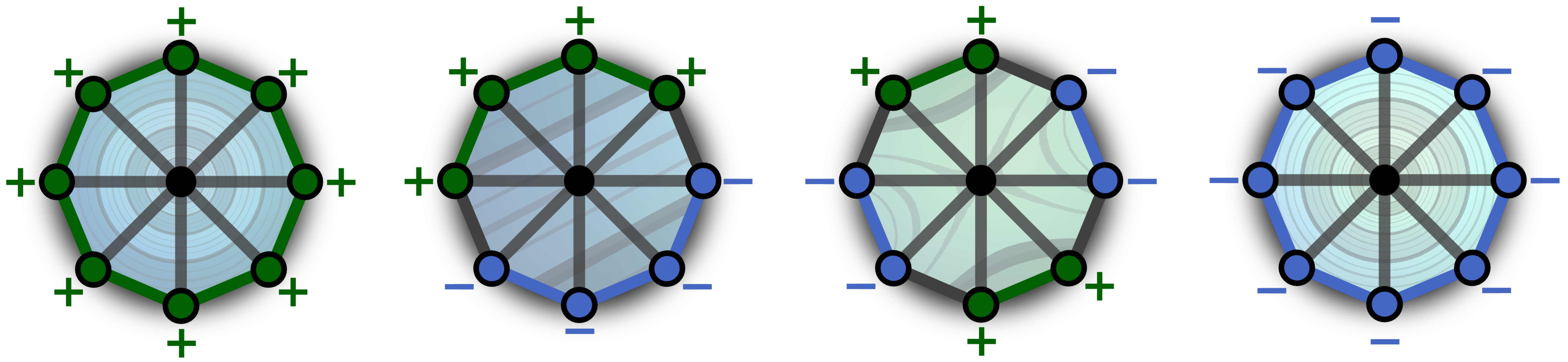




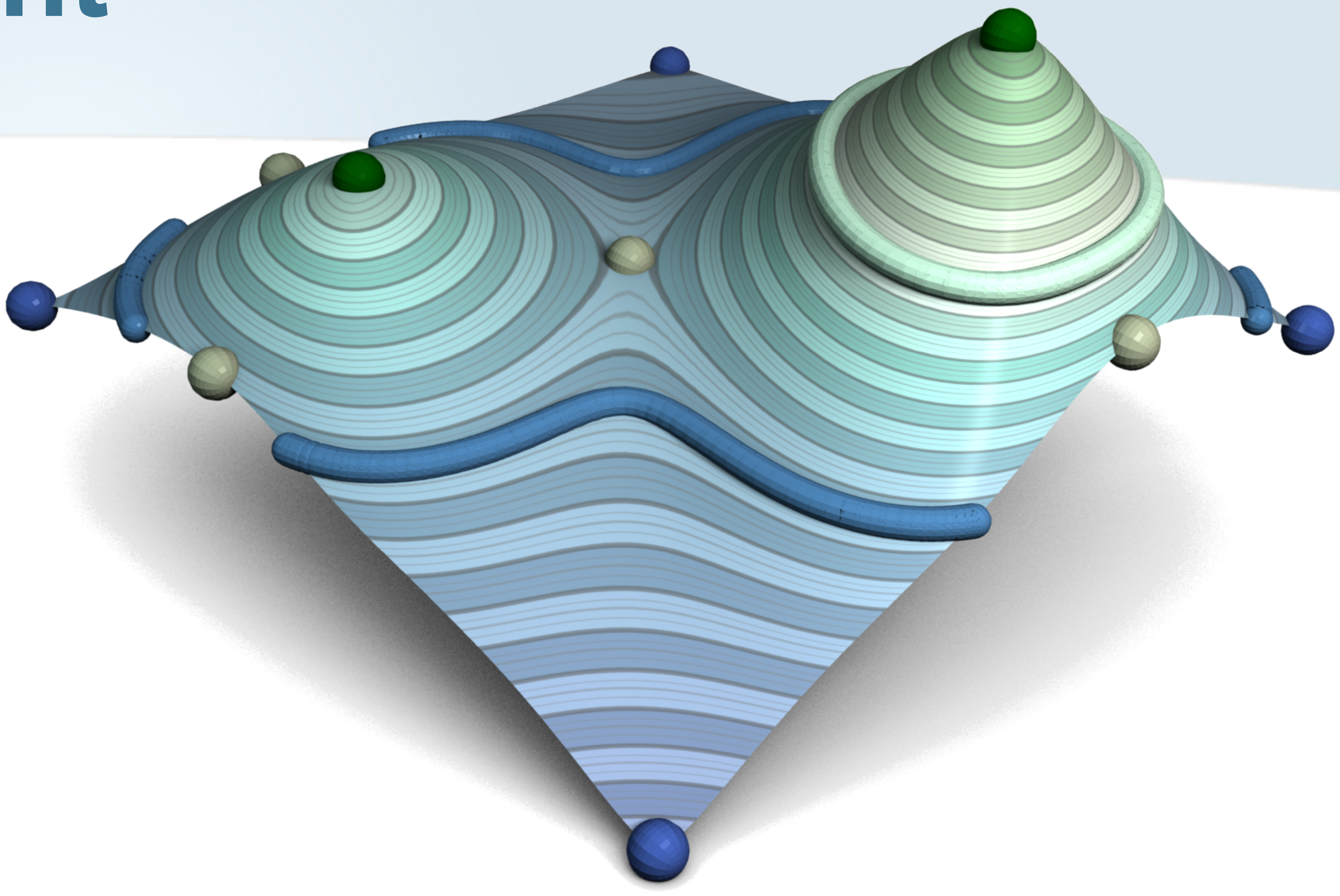
Notion of critical point

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

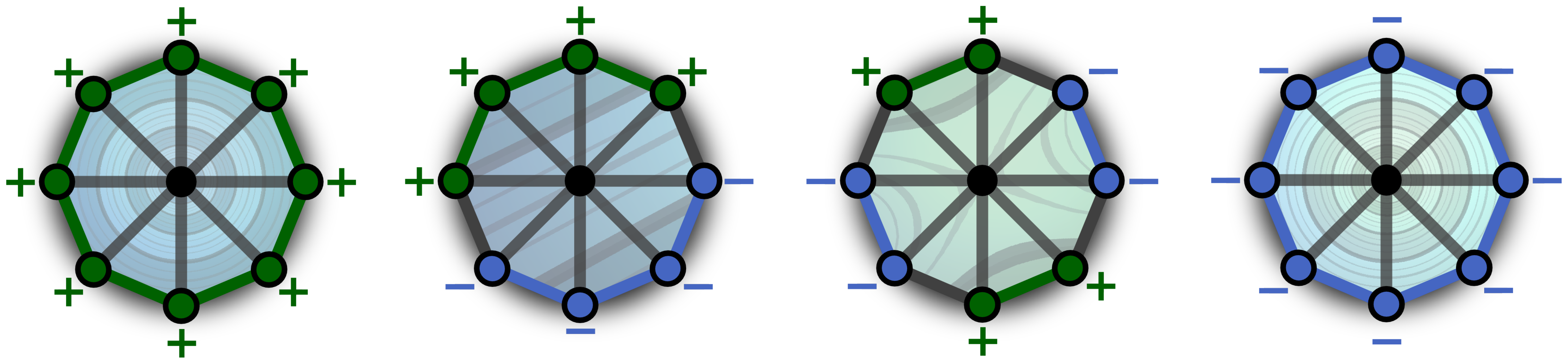




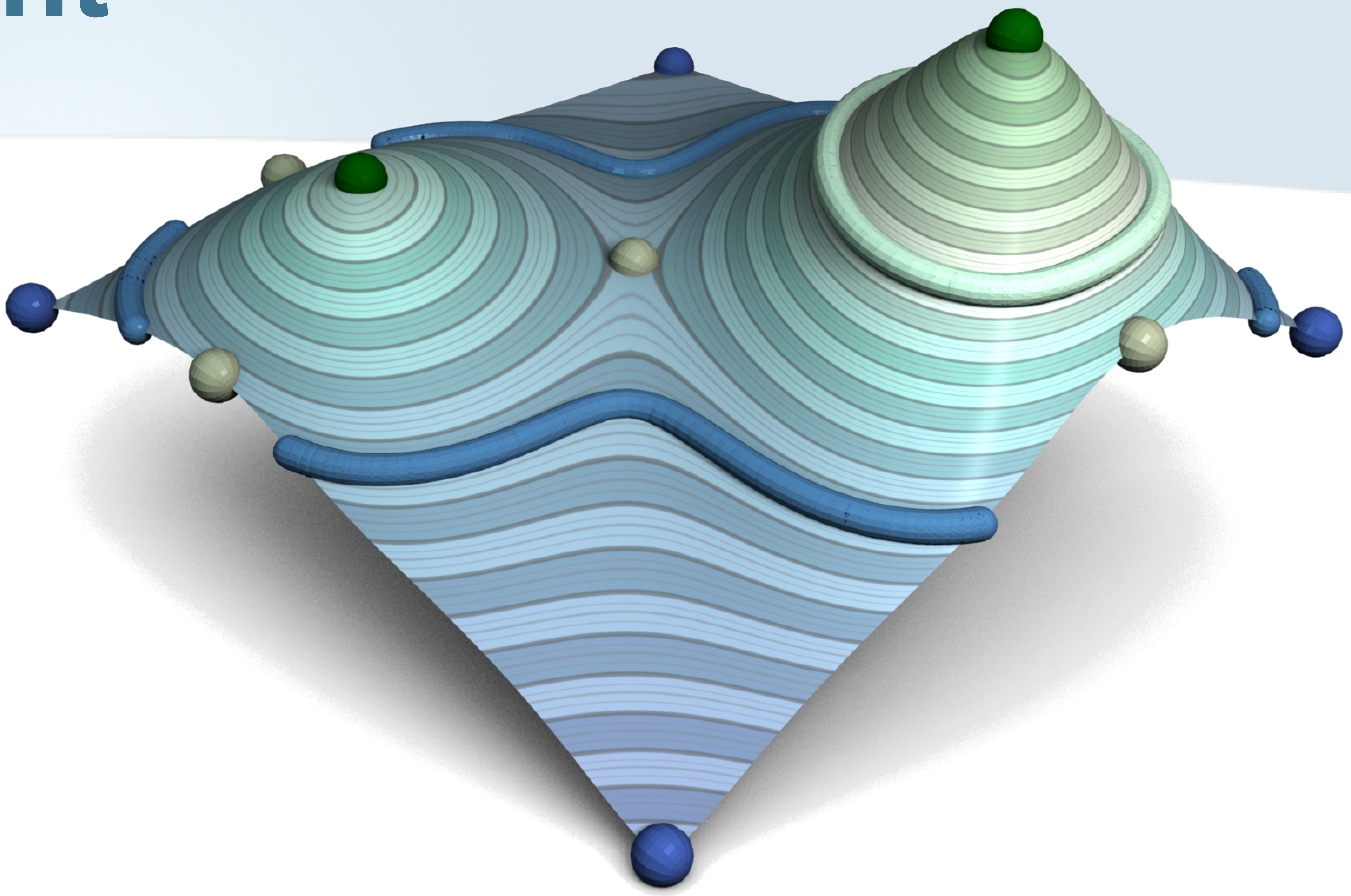
Notion of critical point



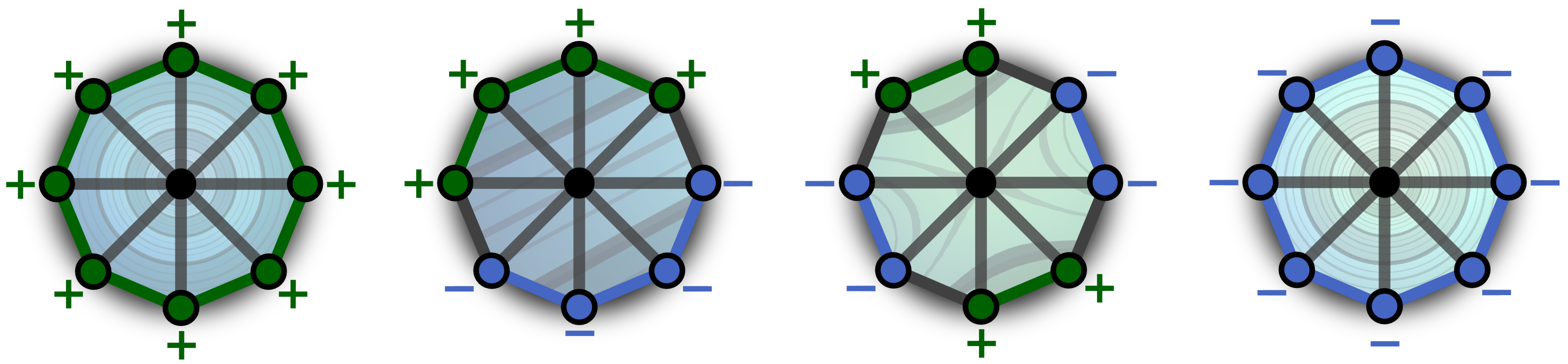
- Combinatorial identification
  - Minimum
    - Empty lower link
  - Maximum
    - Empty upper link
  - Regular point



Notion of critical point

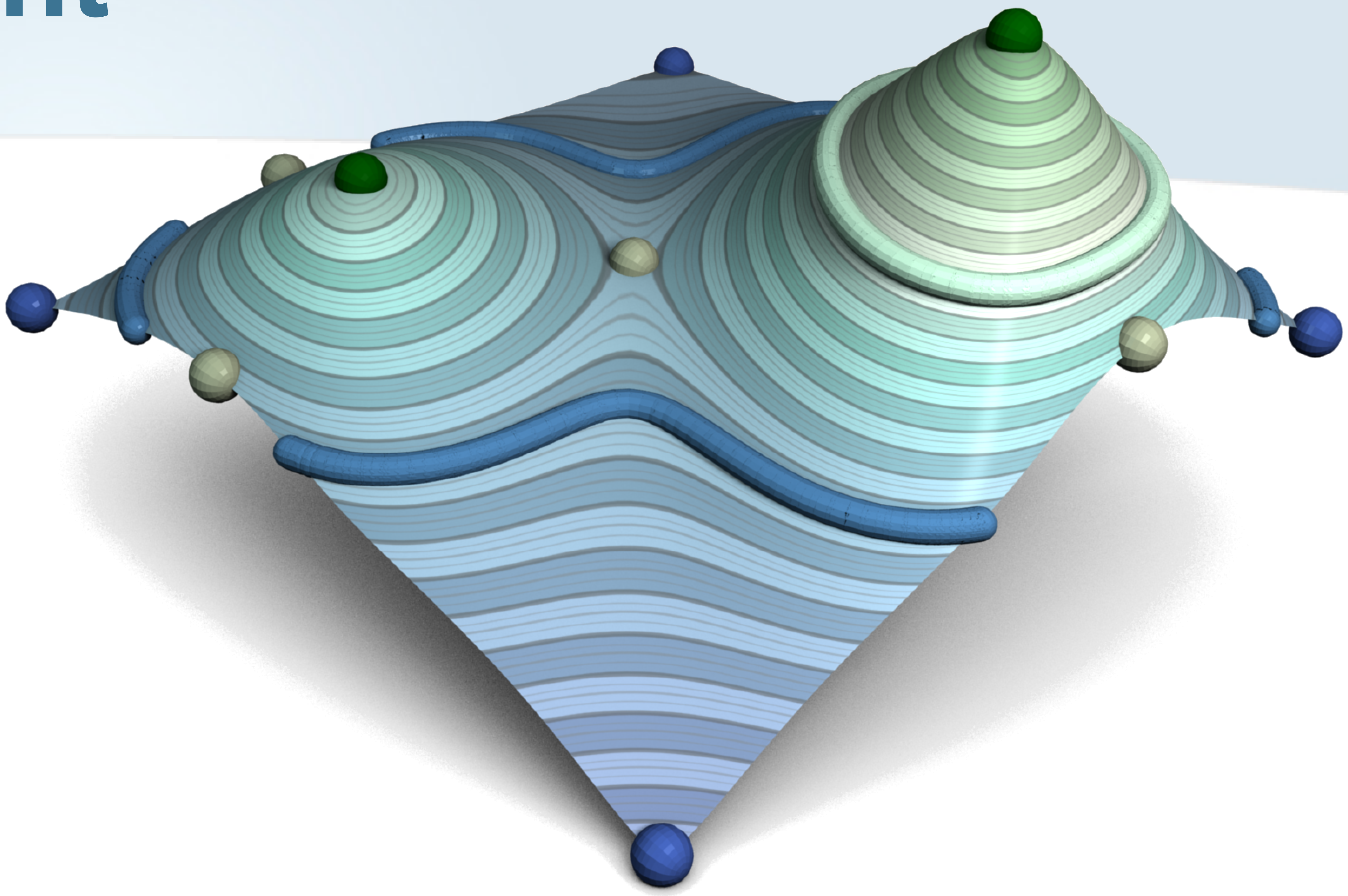


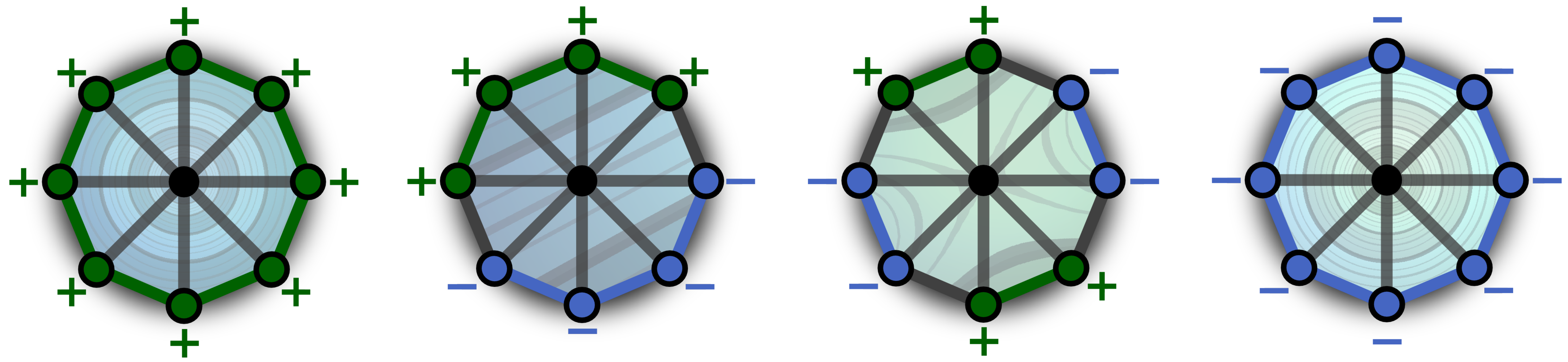
- Combinatorial identification
  - Minimum
    - Empty lower link
  - Maximum
    - Empty upper link
  - Regular point
    - Lower and upper links both *simply connected*



Notion of critical point

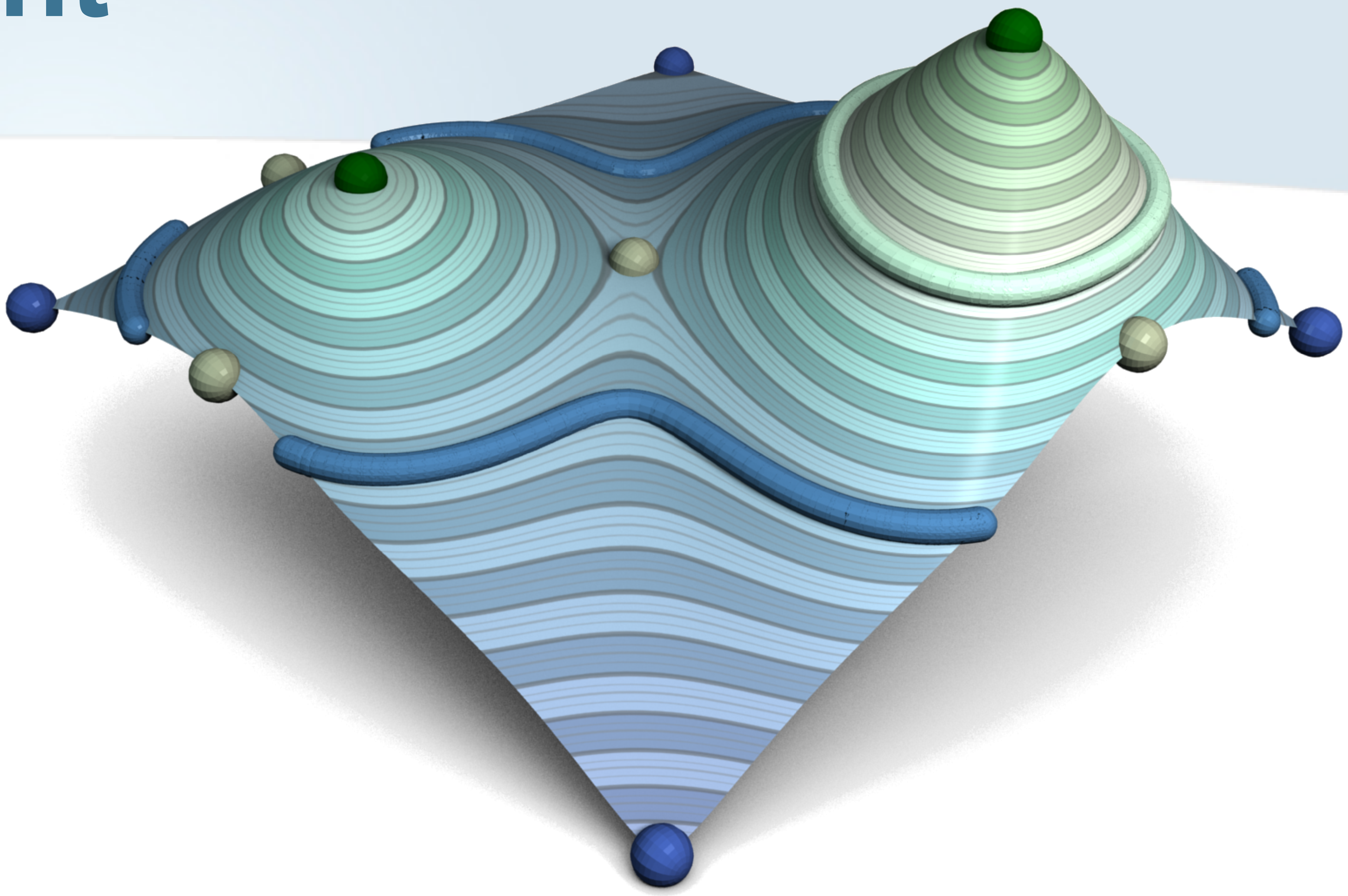
- Combinatorial identification
  - Everything else

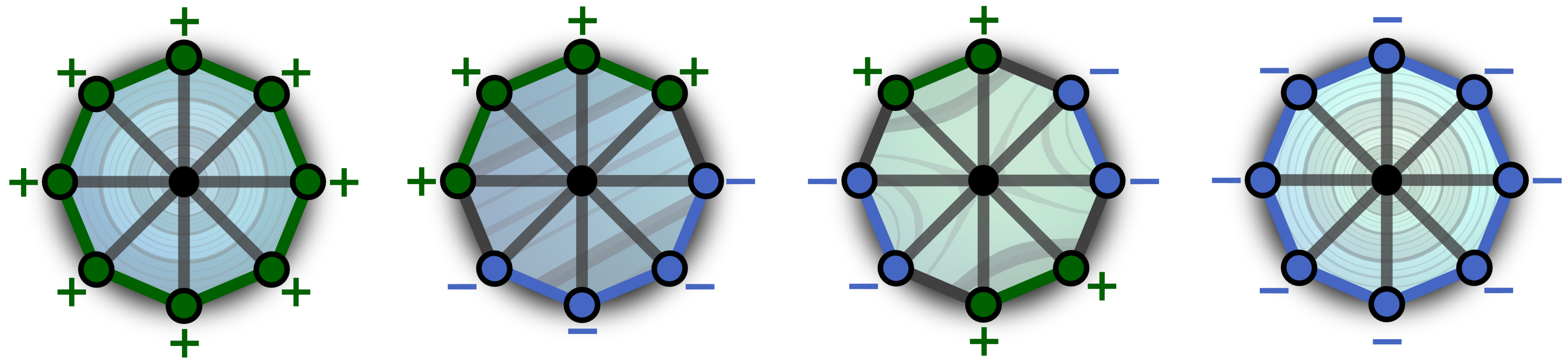




Notion of critical point

- Combinatorial identification
  - Everything else
    - Saddle

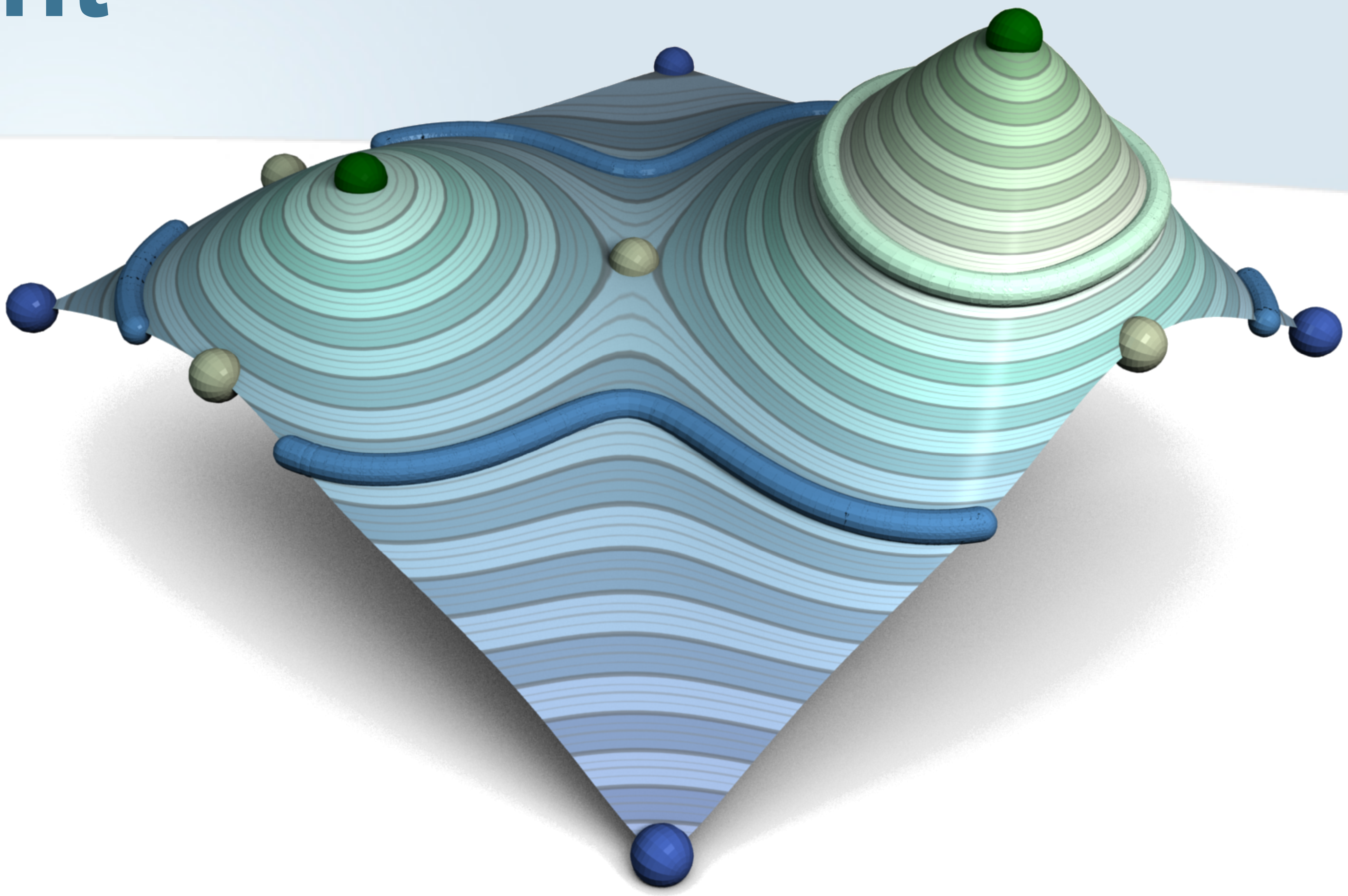


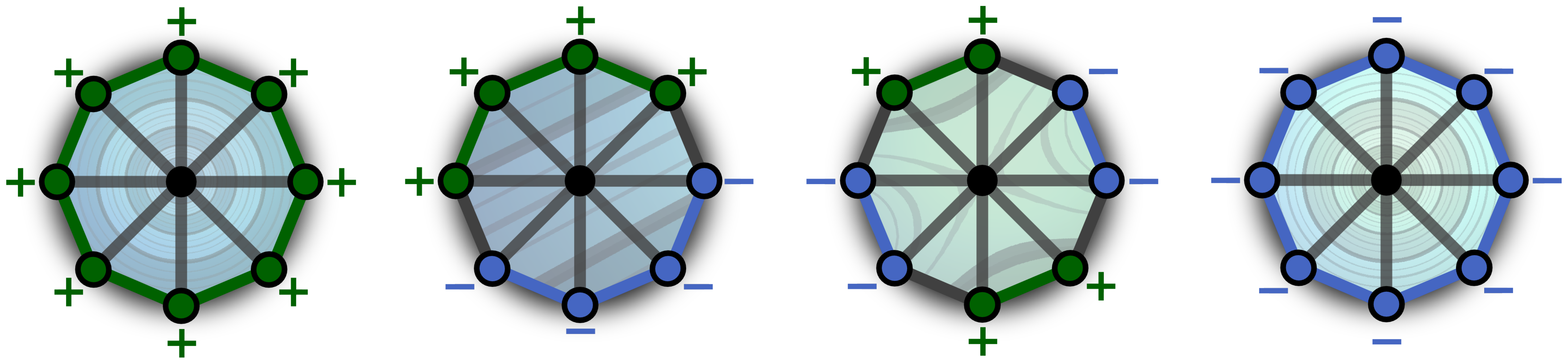


Notion of critical point

- Combinatorial identification
  - Everything else
    - Saddle

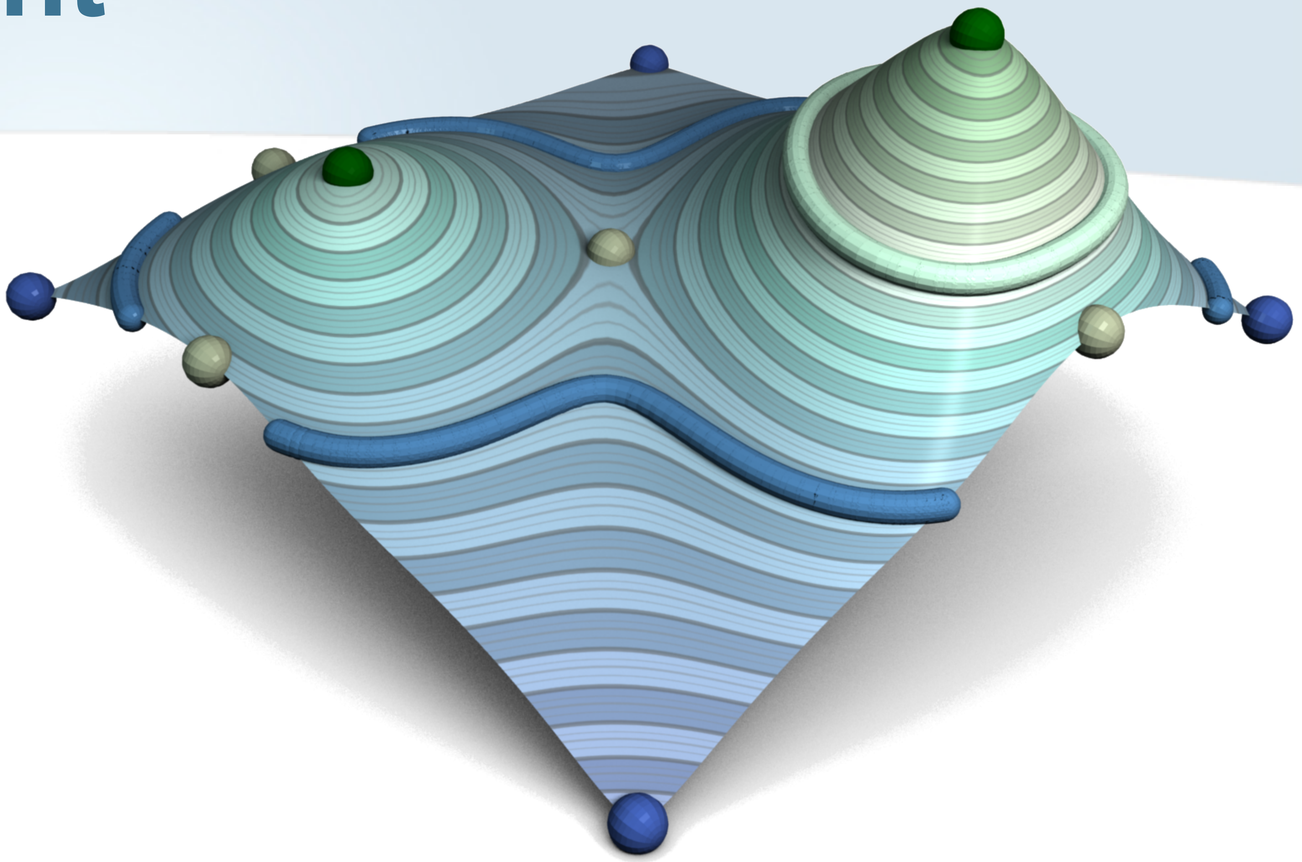
  - Works in arbitrary dimension

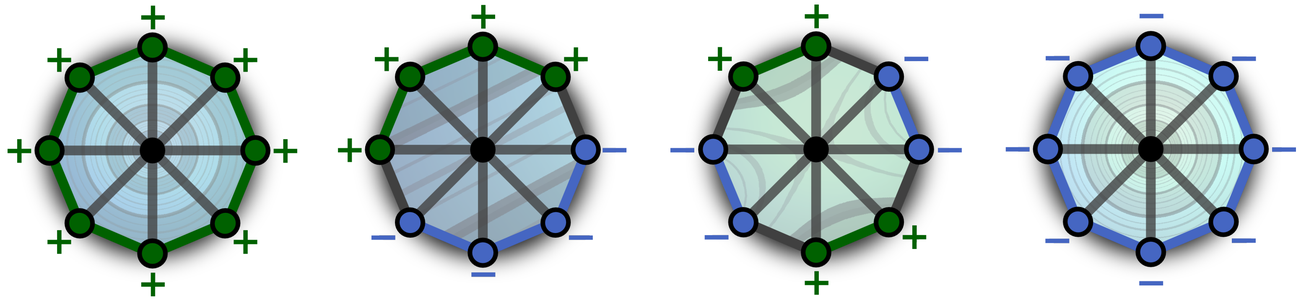




Notion of critical point

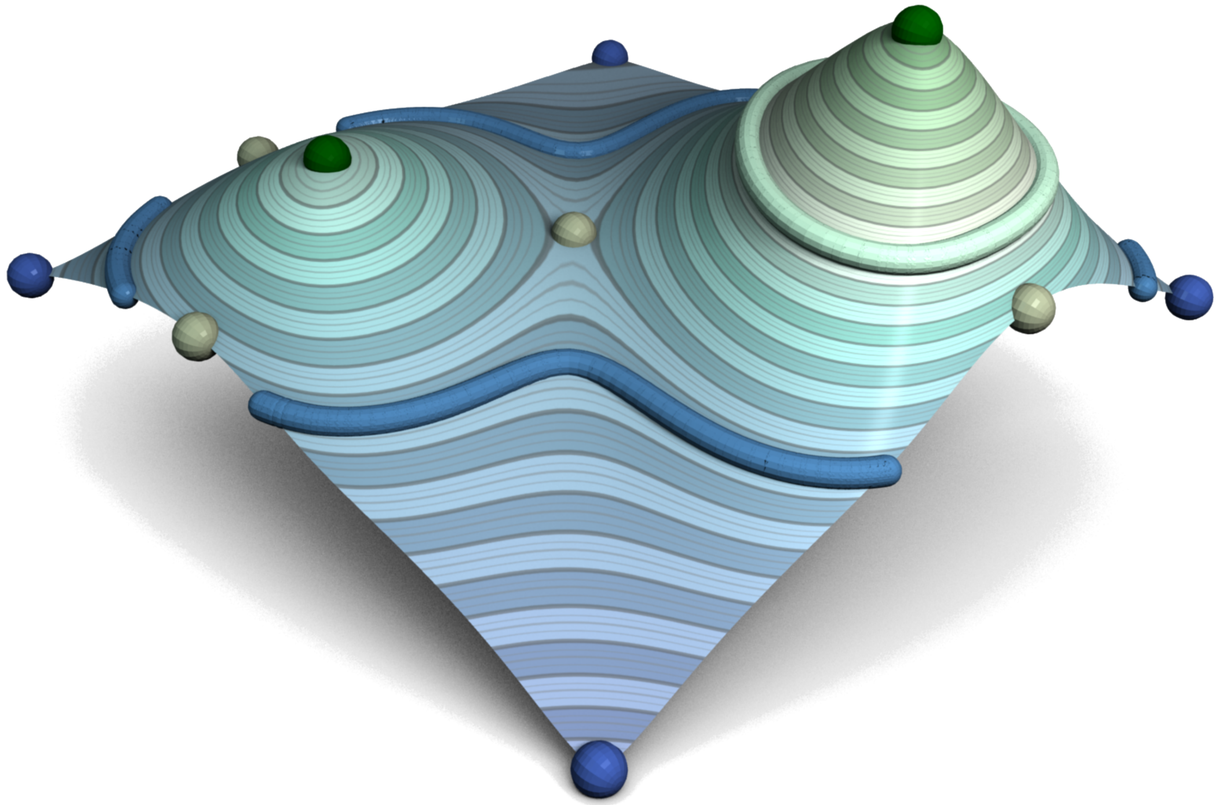
- Combinatorial identification
  - Everything else
    - Saddle
  - Works in arbitrary dimension
  - Value of a critical point
    - *Critical value*





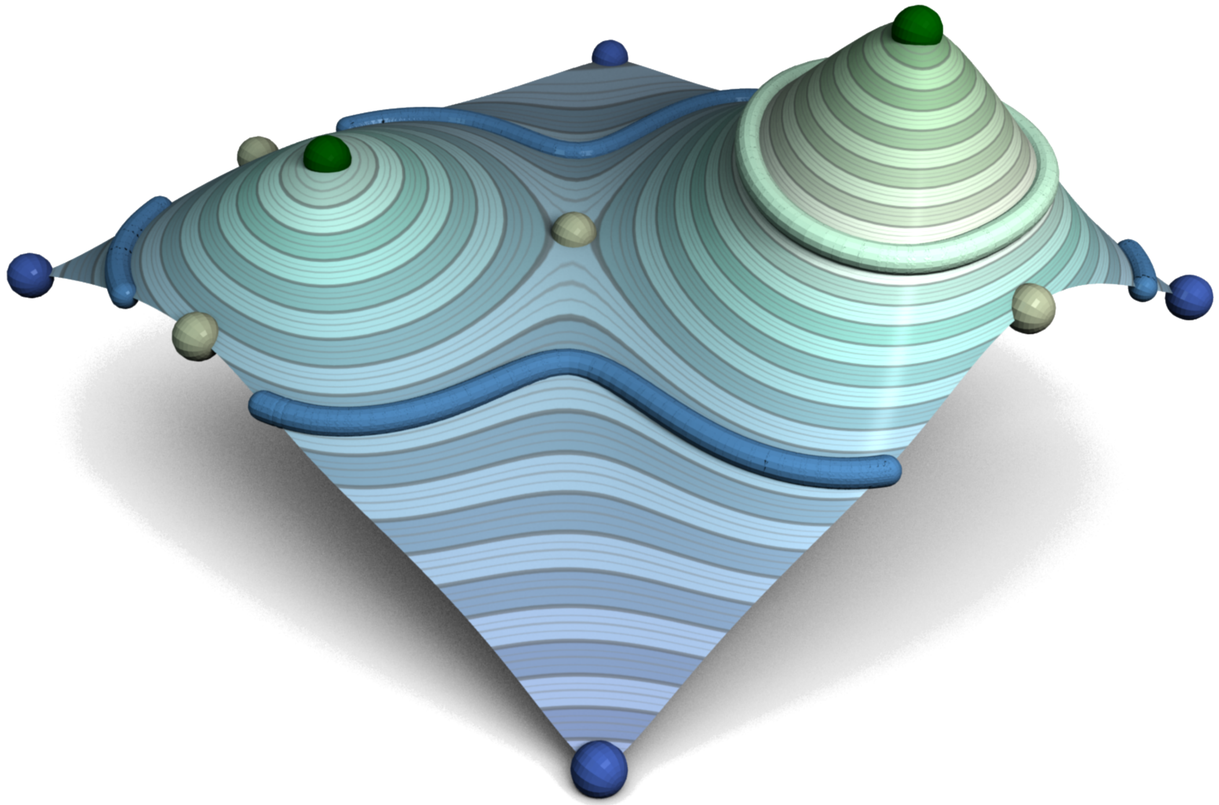
Level sets

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



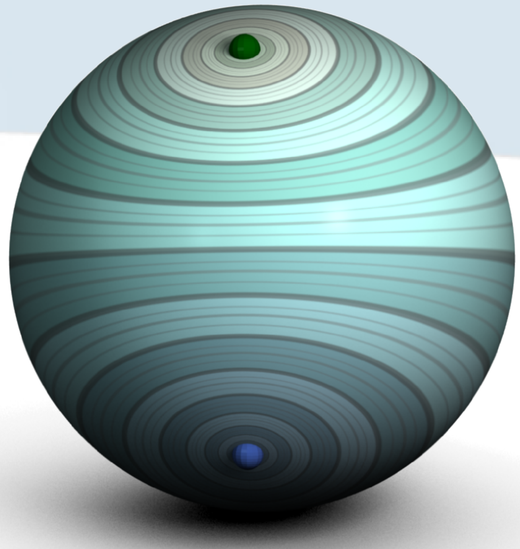
Level sets

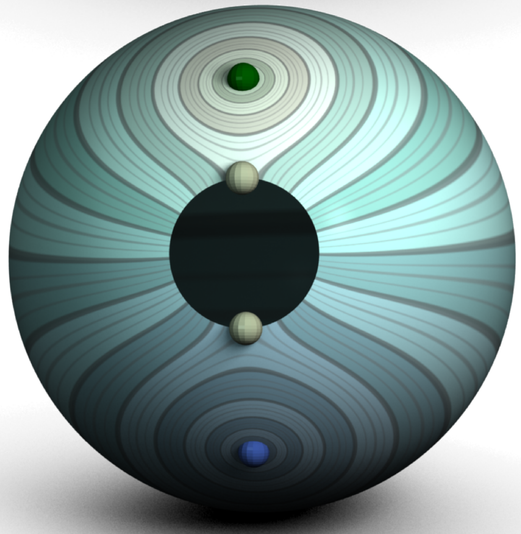
- Given a domain $\mathcal{D}$ of dimension d
  - Level set
    - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
  - If i is not a critical value
    - $f^{-1}(i)$ is a (d-1) manifold



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

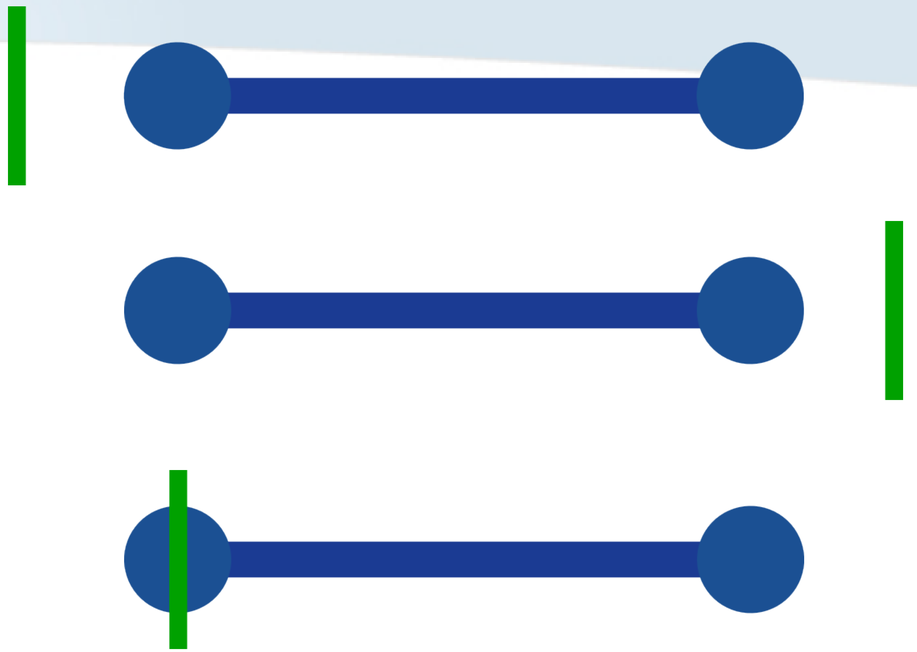
Level set in a 1-simplex

- 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
  - $f(v_0)$
  - $f(v_1)$
  - $f(v_0) < f(v_1)$

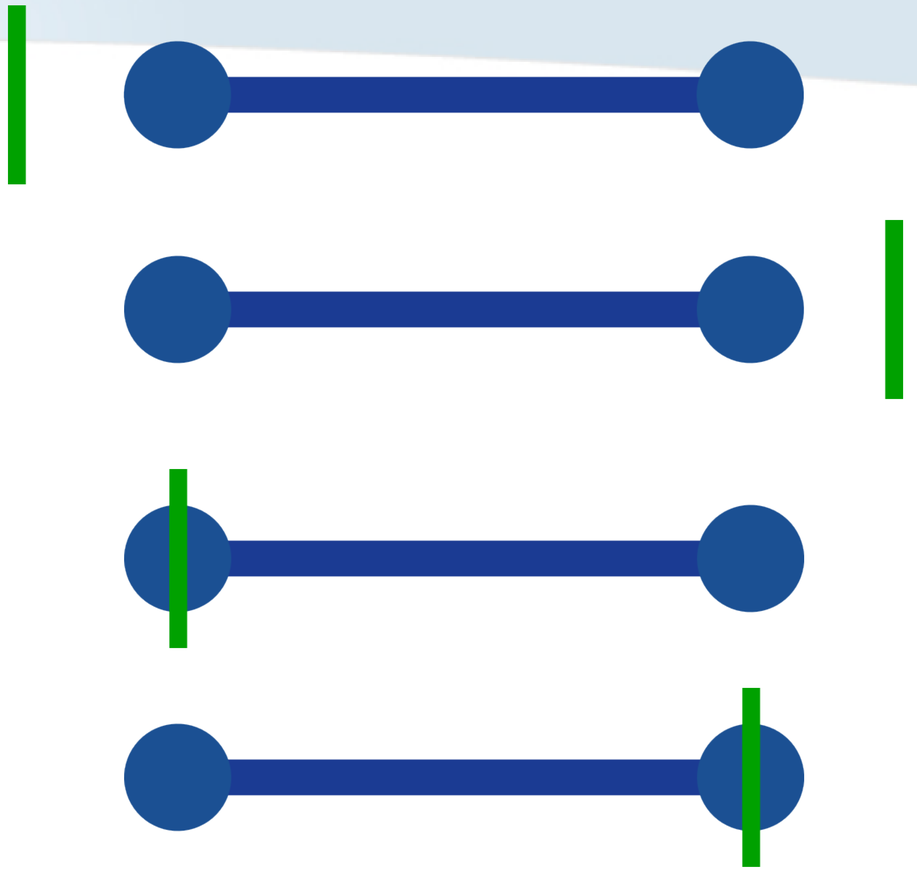


- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**

Level set in a 1-simplex

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

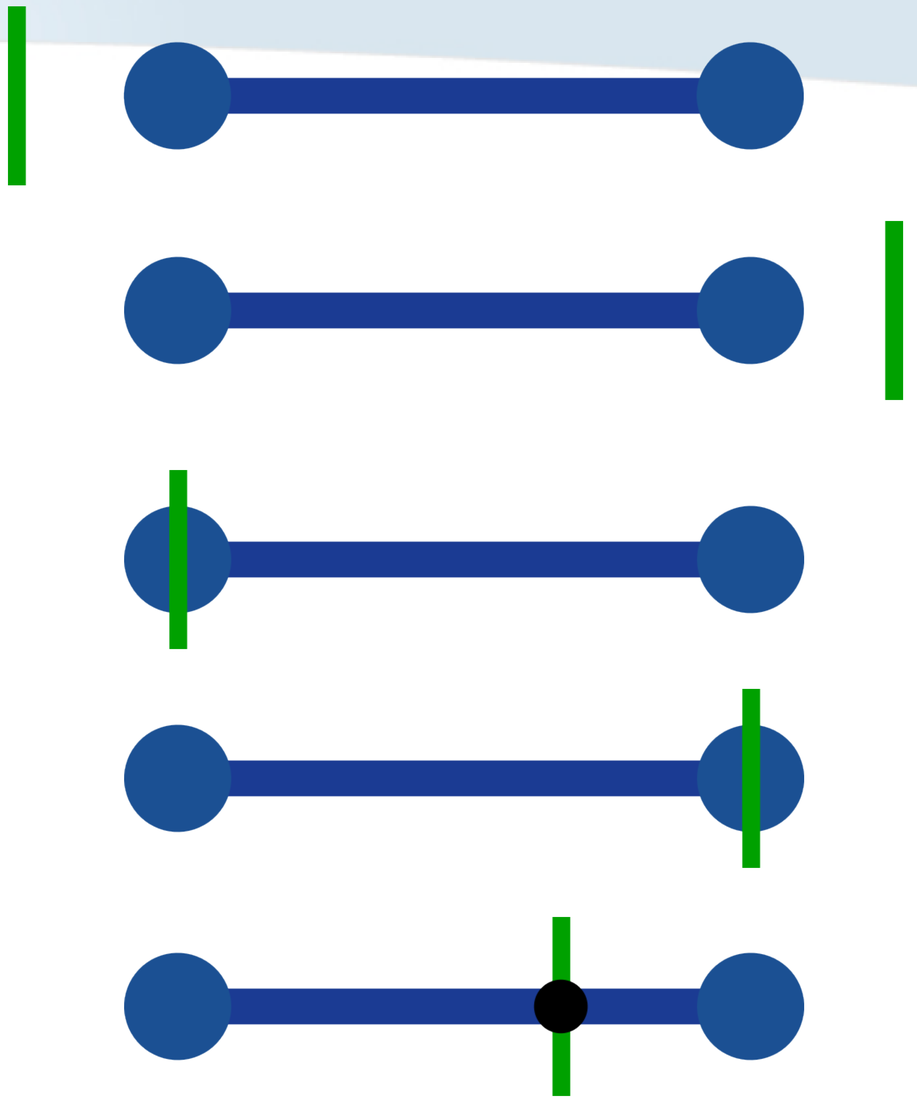
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**



Level set in a 1-simplex

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

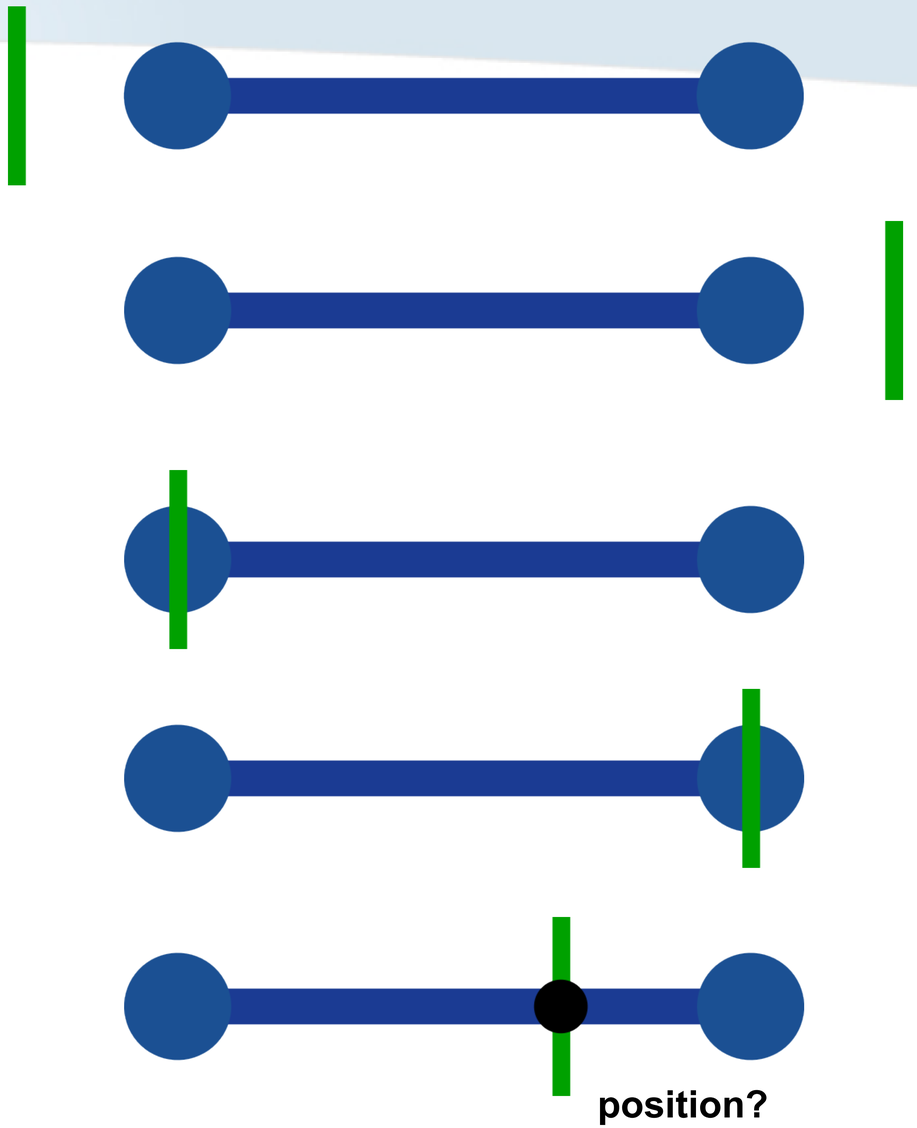
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - 0-manifold
  - If intersection, **3 cases only**



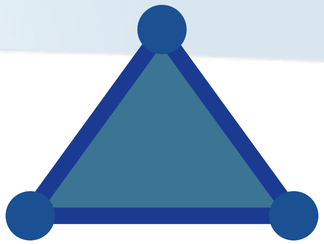
Level set in a 1-simplex

- 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**

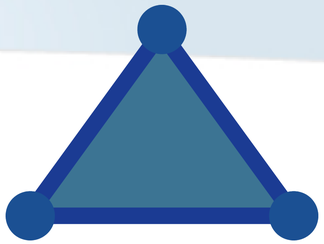


Level set in a 2-simplex



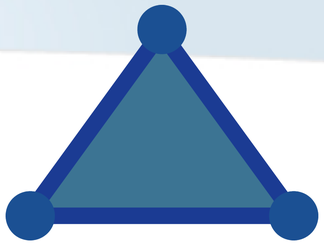
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$

Level set in a 2-simplex



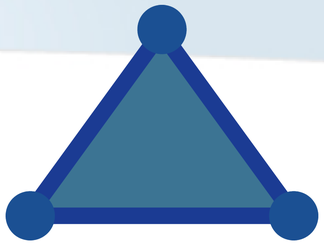
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$

Level set in a 2-simplex



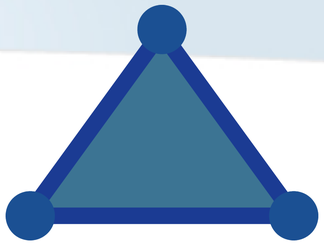
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**

Level set in a 2-simplex



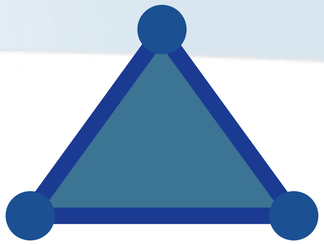
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold

Level set in a 2-simplex



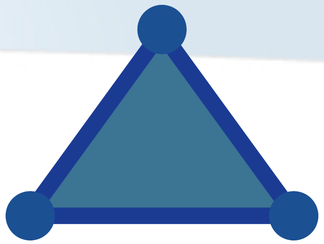
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary

Level set in a 2-simplex



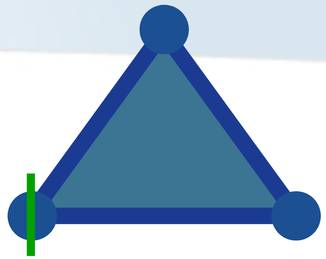
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



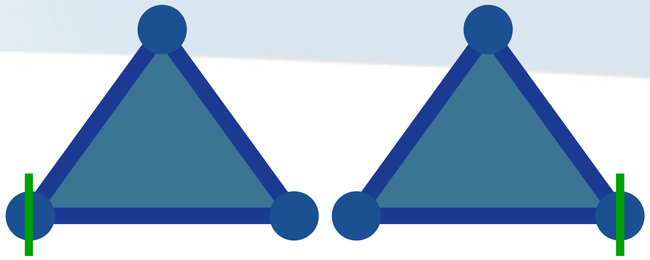
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



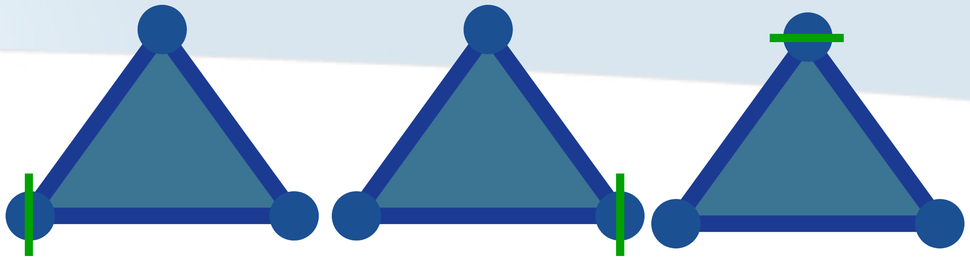
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



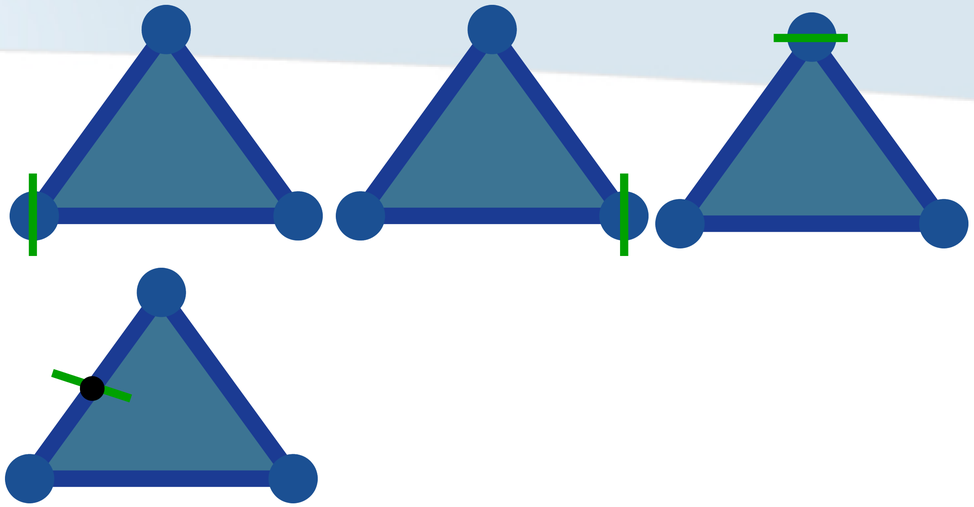
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



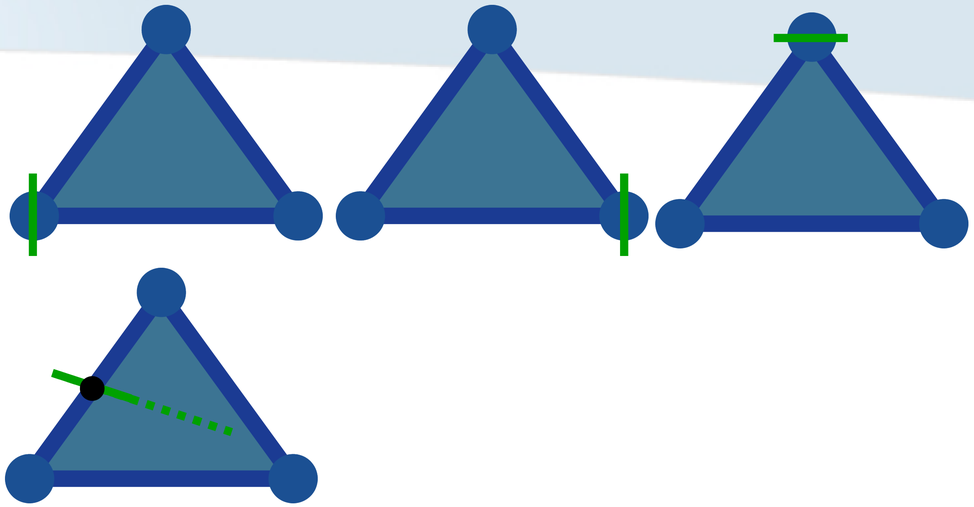
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



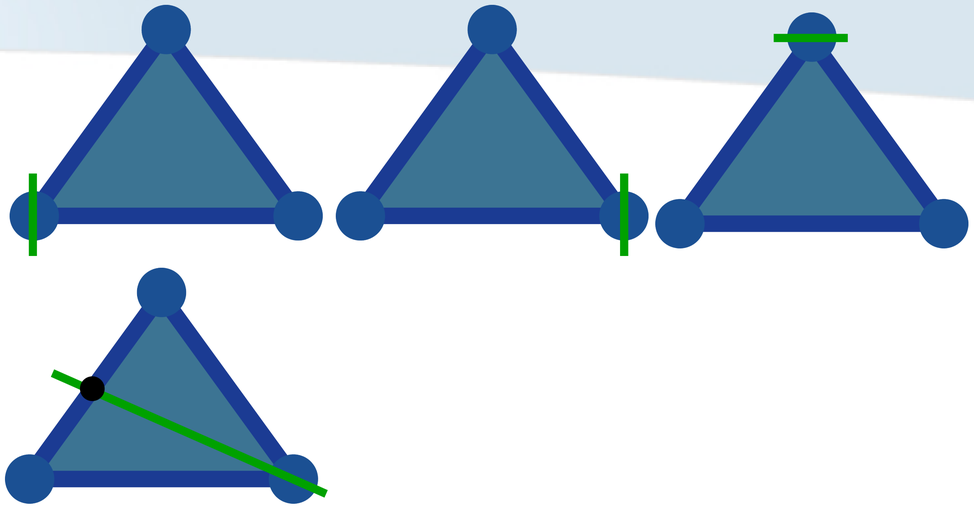
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



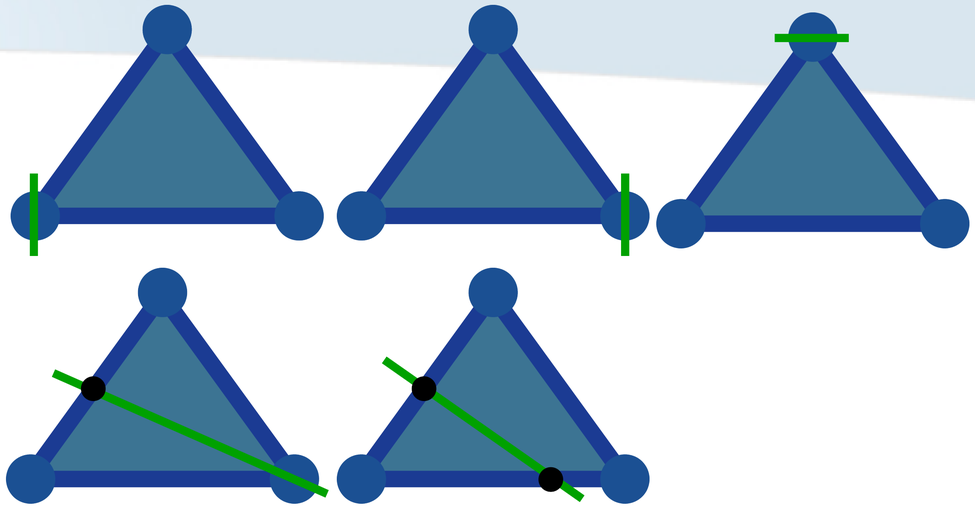
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



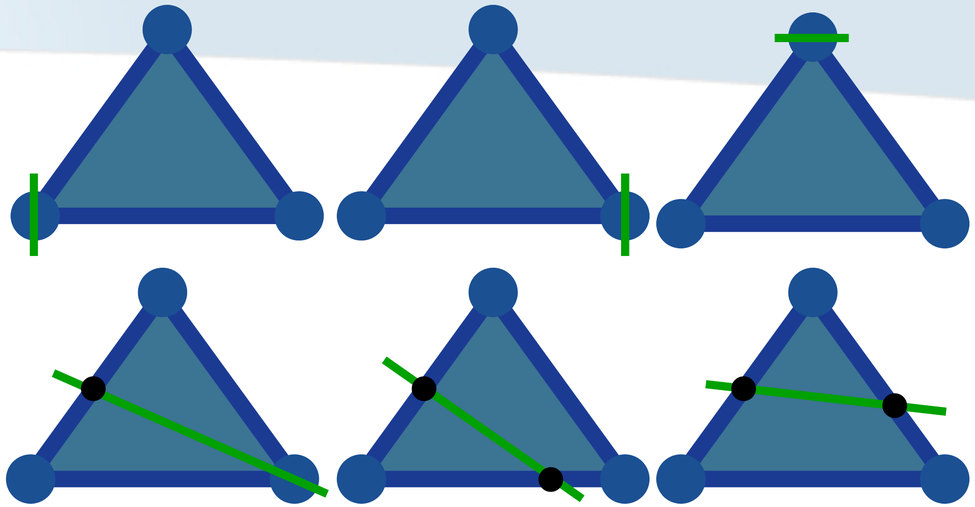
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



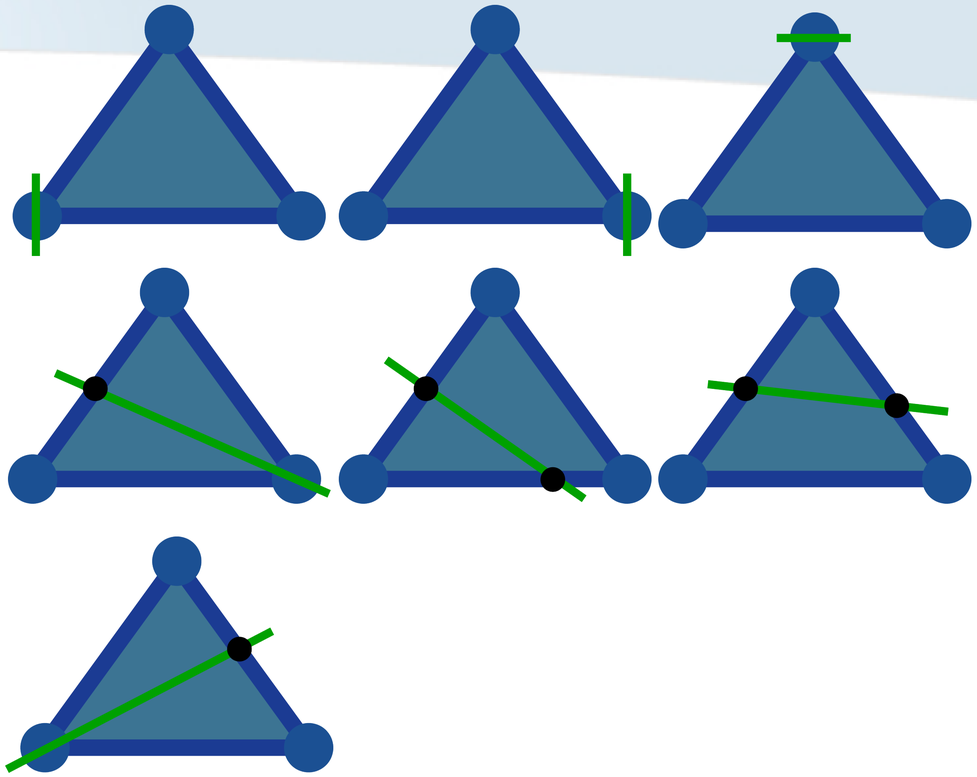
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



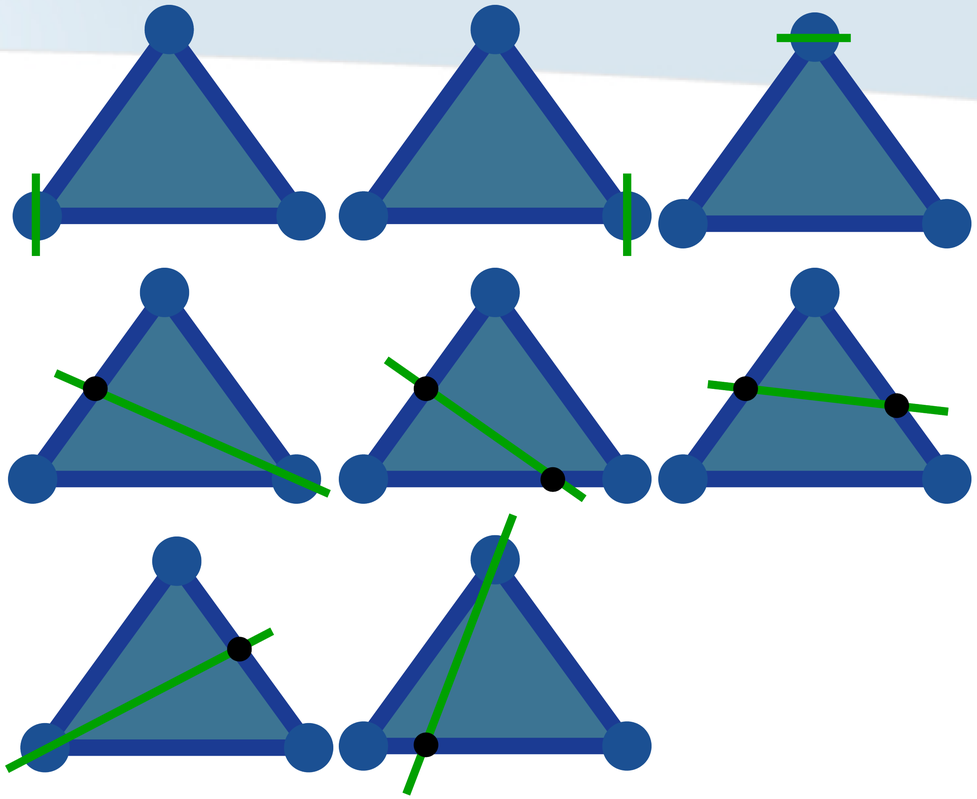
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

Level set in a 2-simplex



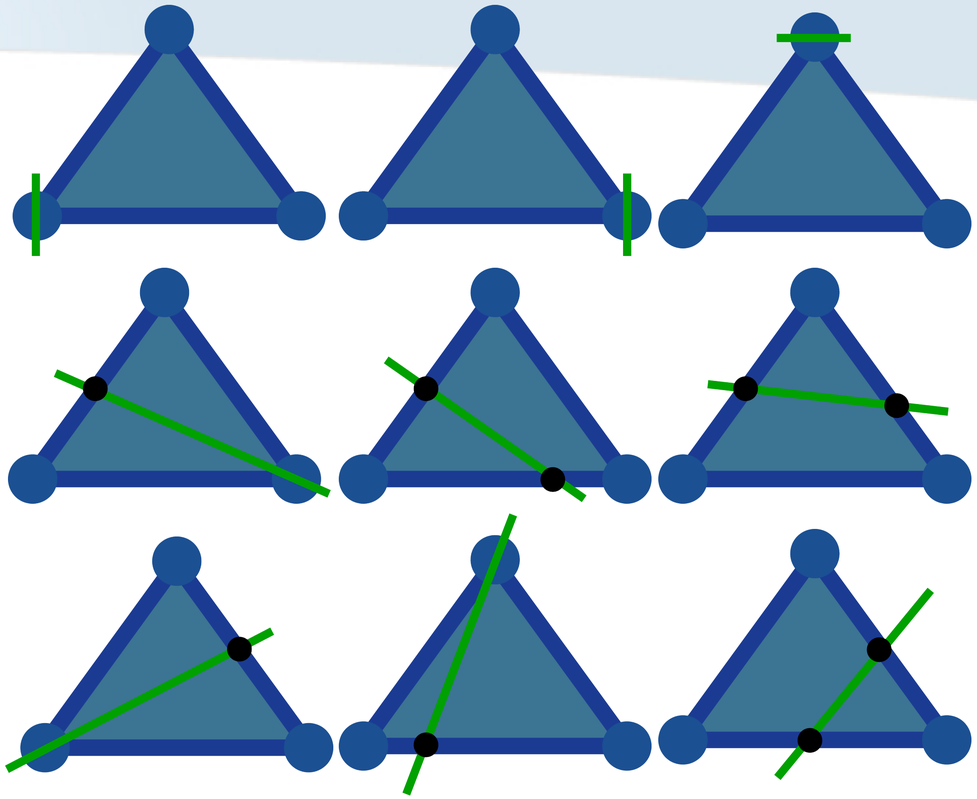
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

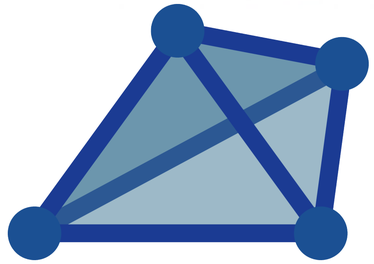
Level set in a 2-simplex



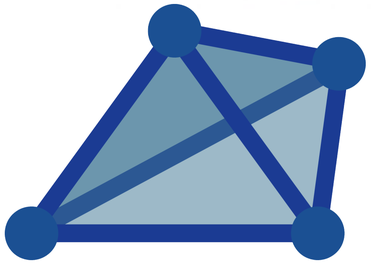
- Let $\mathcal{D}$ be a single triangle
  - $f(v_0), f(v_1), f(v_2)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the triangle!**
    - Simply connected, open, 1-manifold
    - Can be computed by only looking at the boundary
      - Level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

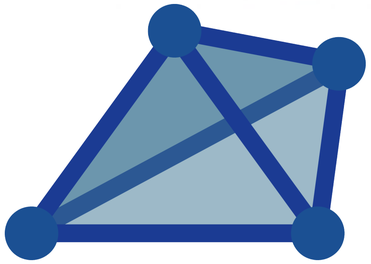


Level set in a 3-simplex



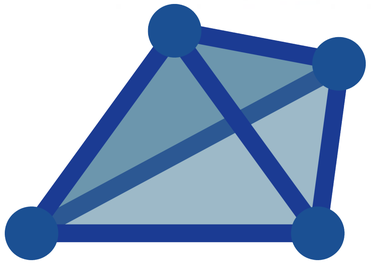
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$

Level set in a 3-simplex



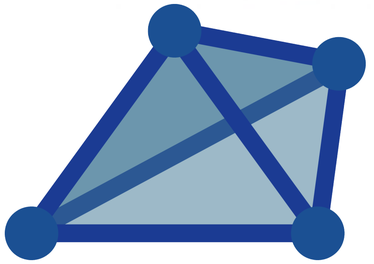
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**

Level set in a 3-simplex



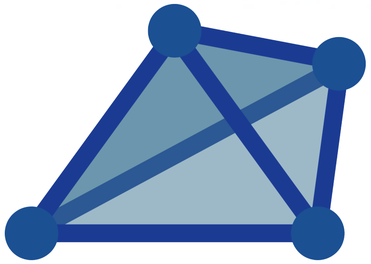
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold

Level set in a 3-simplex



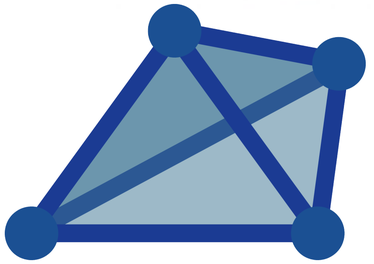
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary

Level set in a 3-simplex



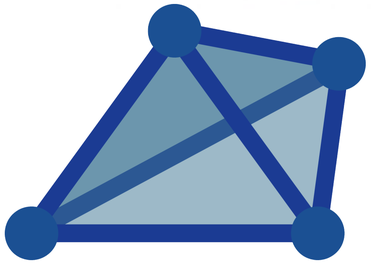
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

Level set in a 3-simplex



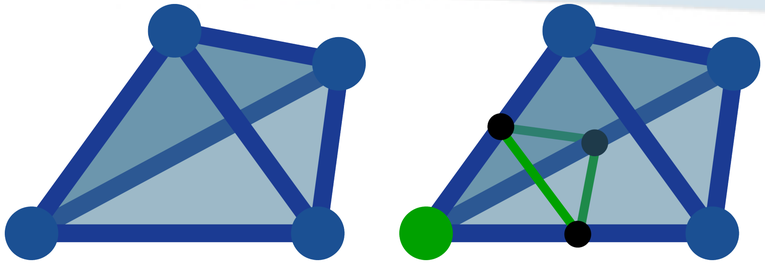
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

Level set in a 3-simplex



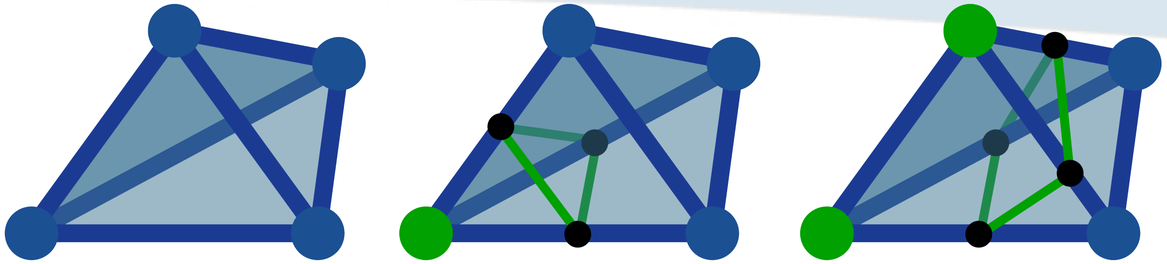
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

Level set in a 3-simplex



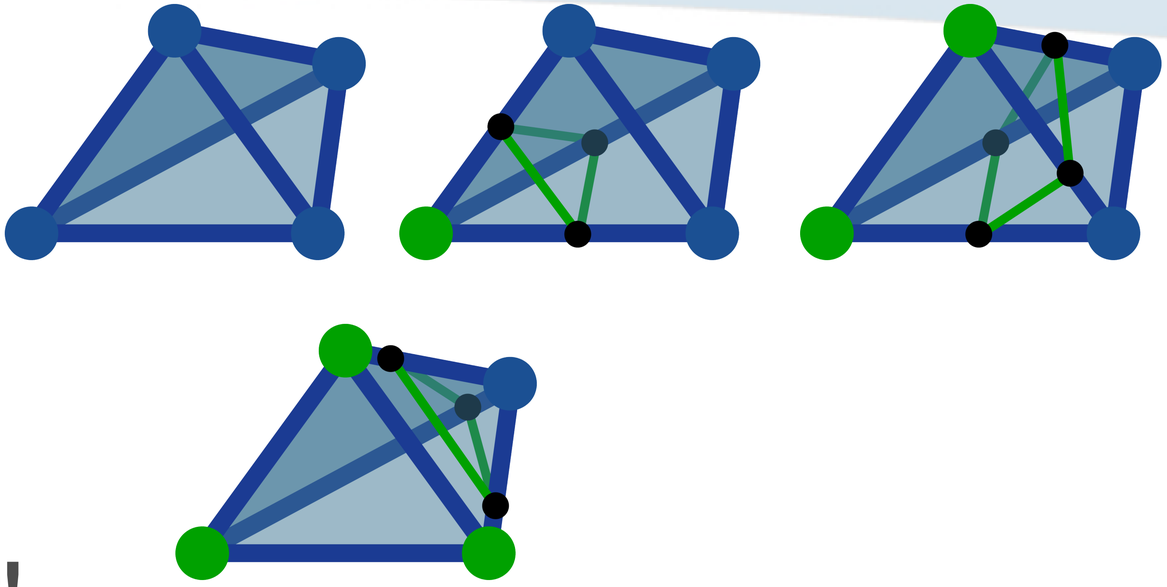
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

Level set in a 3-simplex



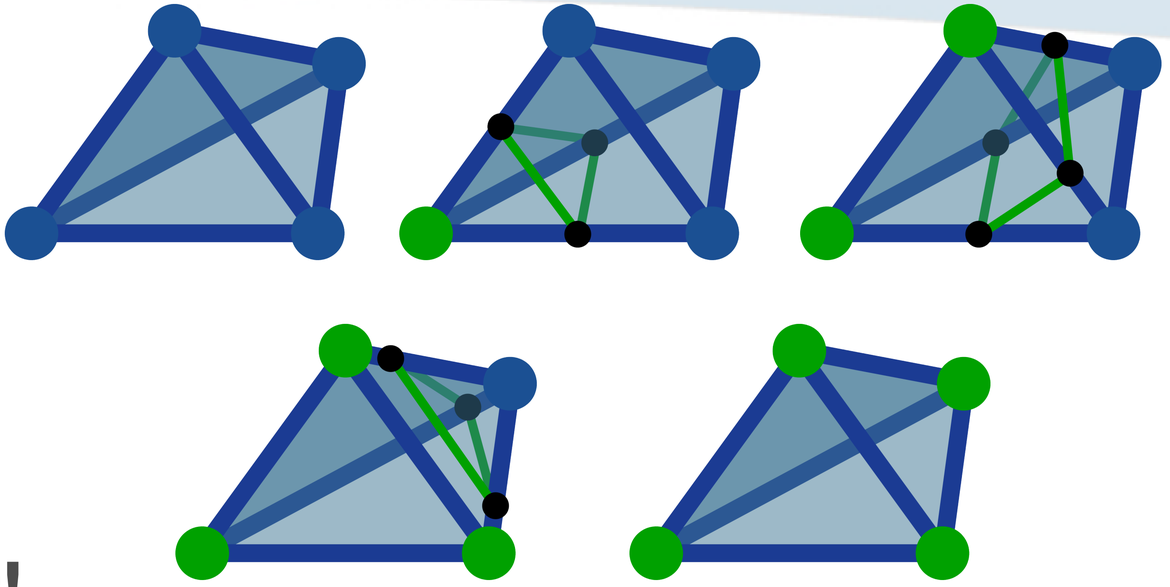
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

Level set in a 3-simplex



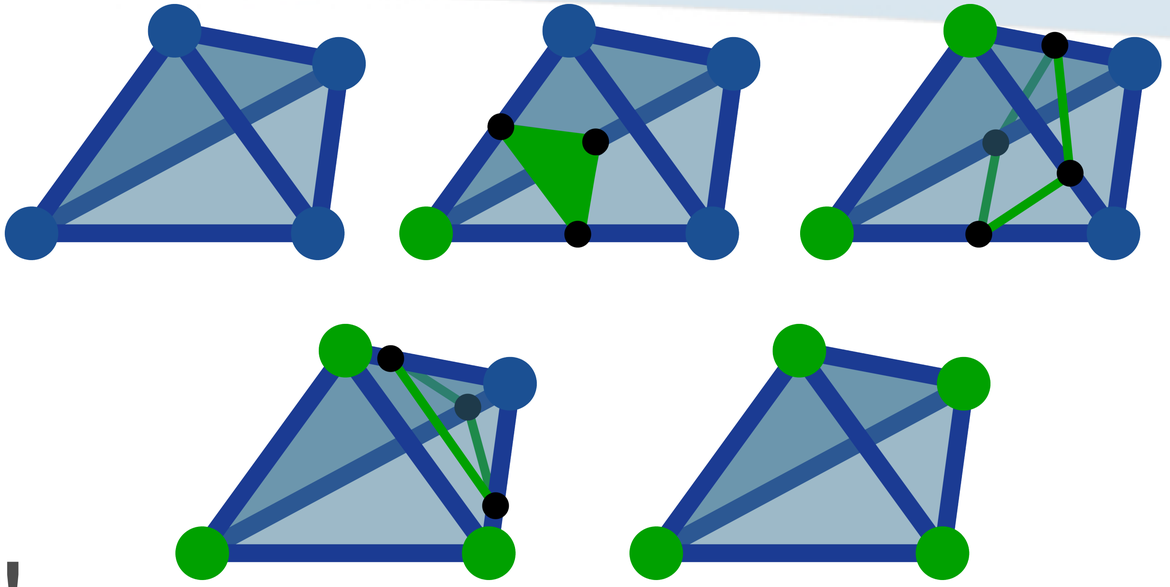
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

Level set in a 3-simplex



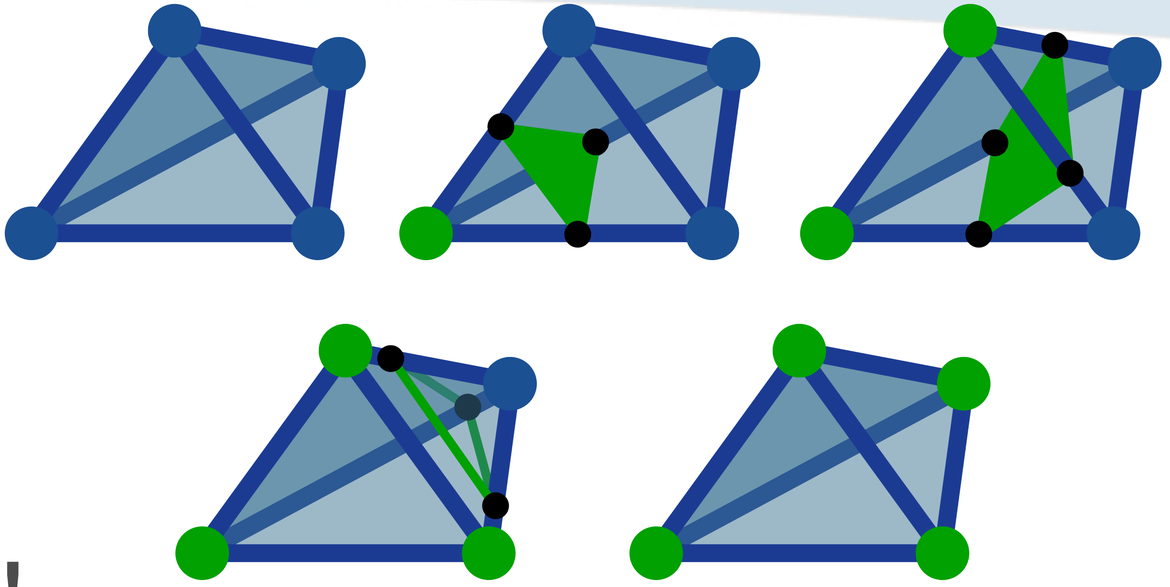
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

Level set in a 3-simplex



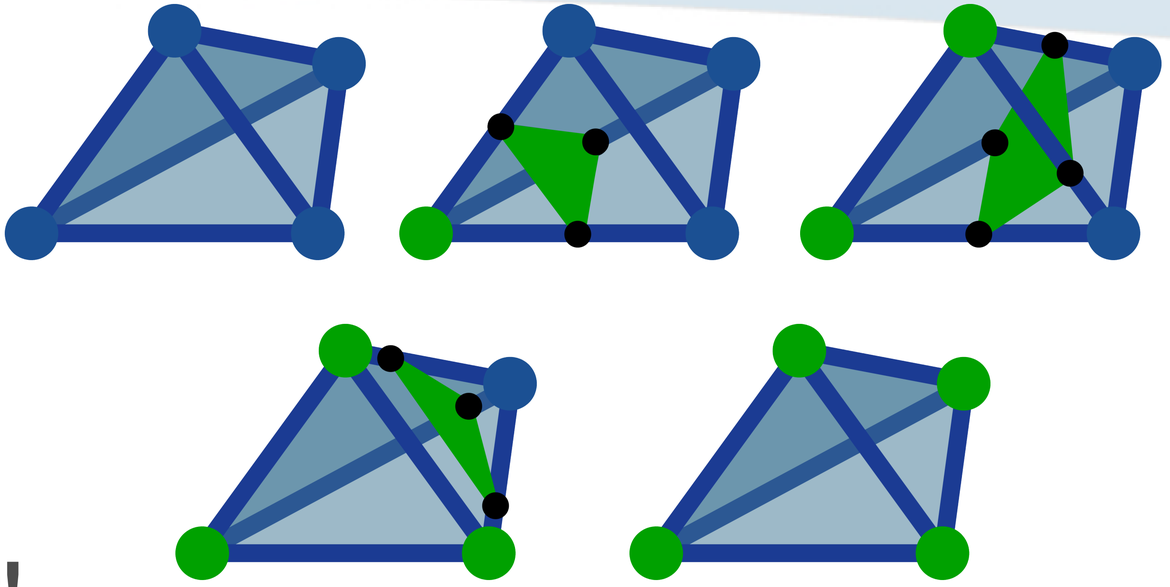
- Let $\mathcal{D}$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let i be an isovalue
  - $f^{-1}(i) = \{p \in \mathcal{D} \ / \ f(p) = i\}$
    - **No critical point inside the tet!**
    - Simply connected, open, 2-manifold
    - Can be computed by only looking at the boundary
  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

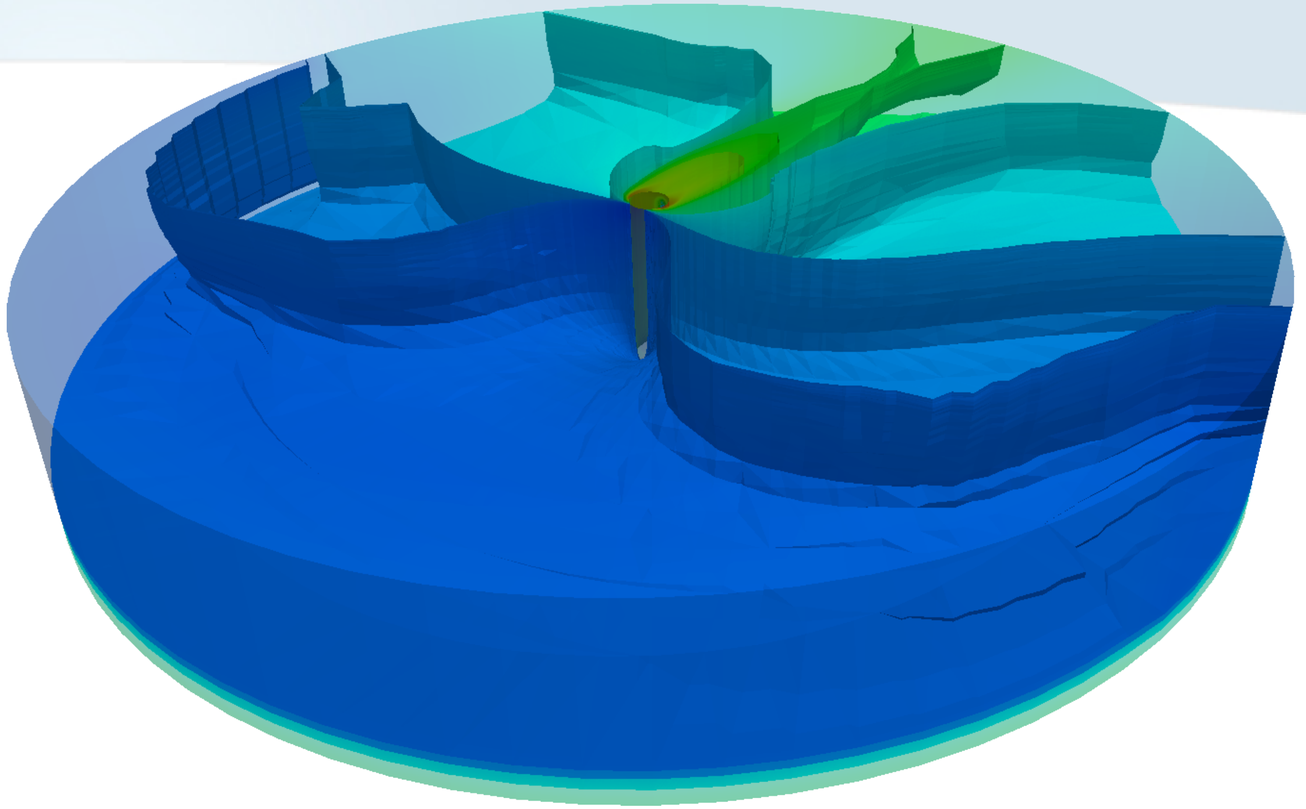
Level set in a d-simplex

- 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

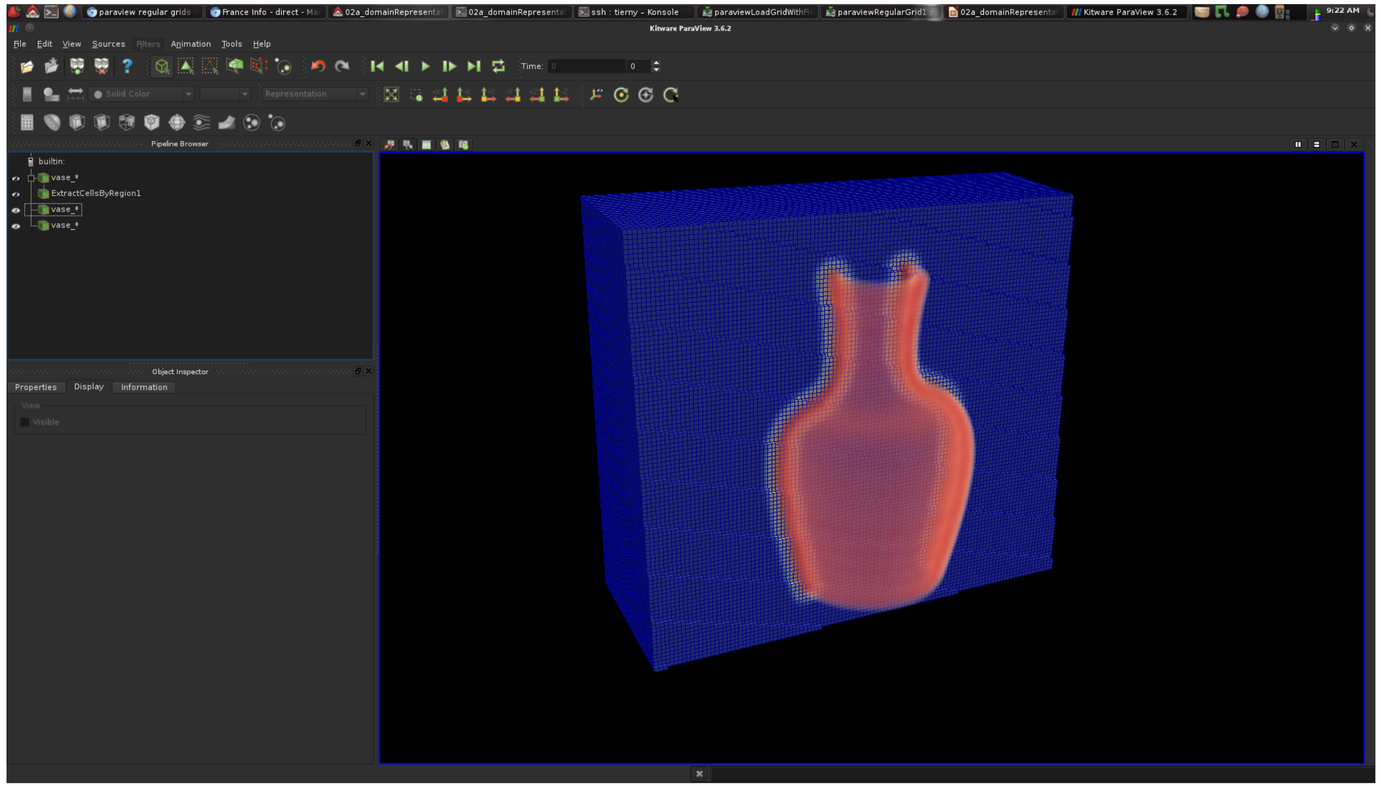
Level set extraction



- 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”*

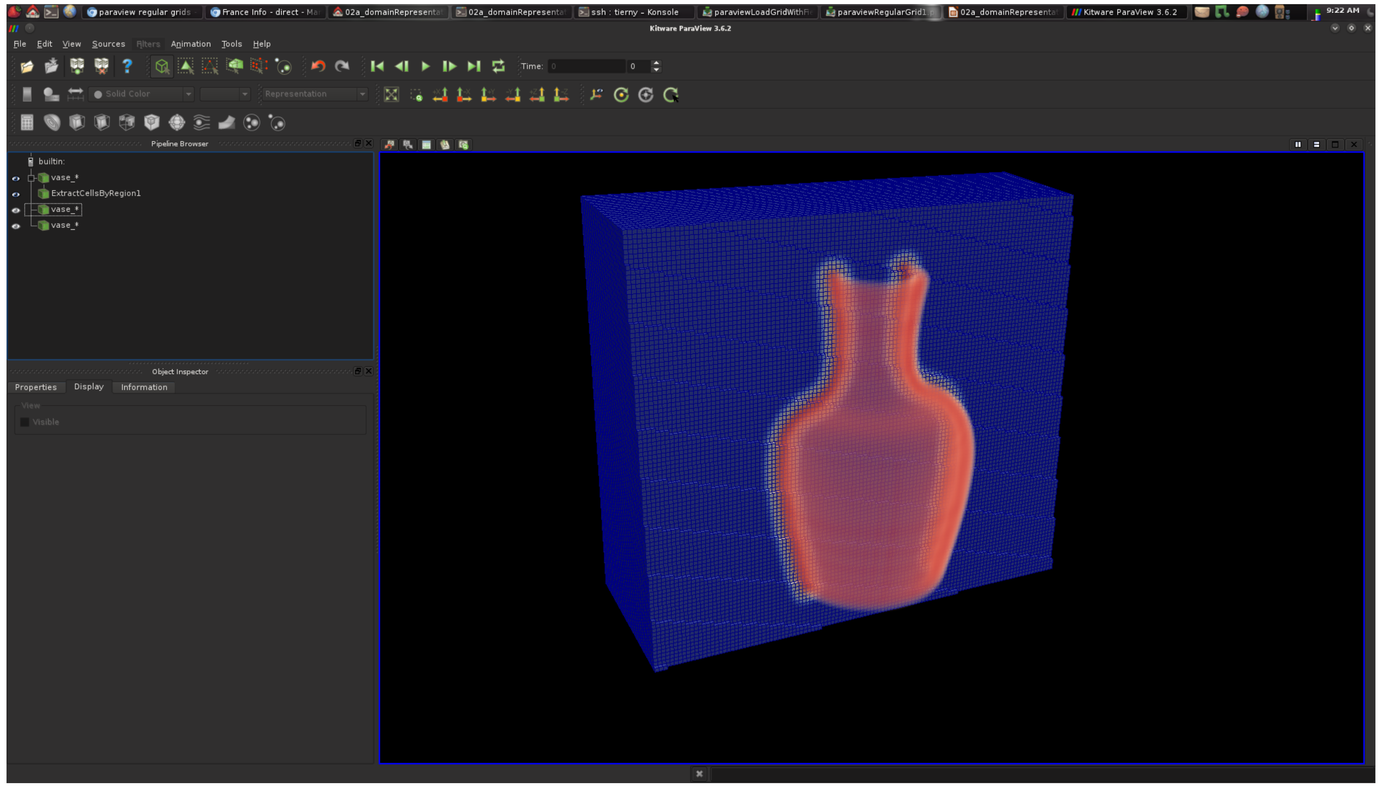
Level sets on regular grids

- What about regular grids?



Level sets on regular grids

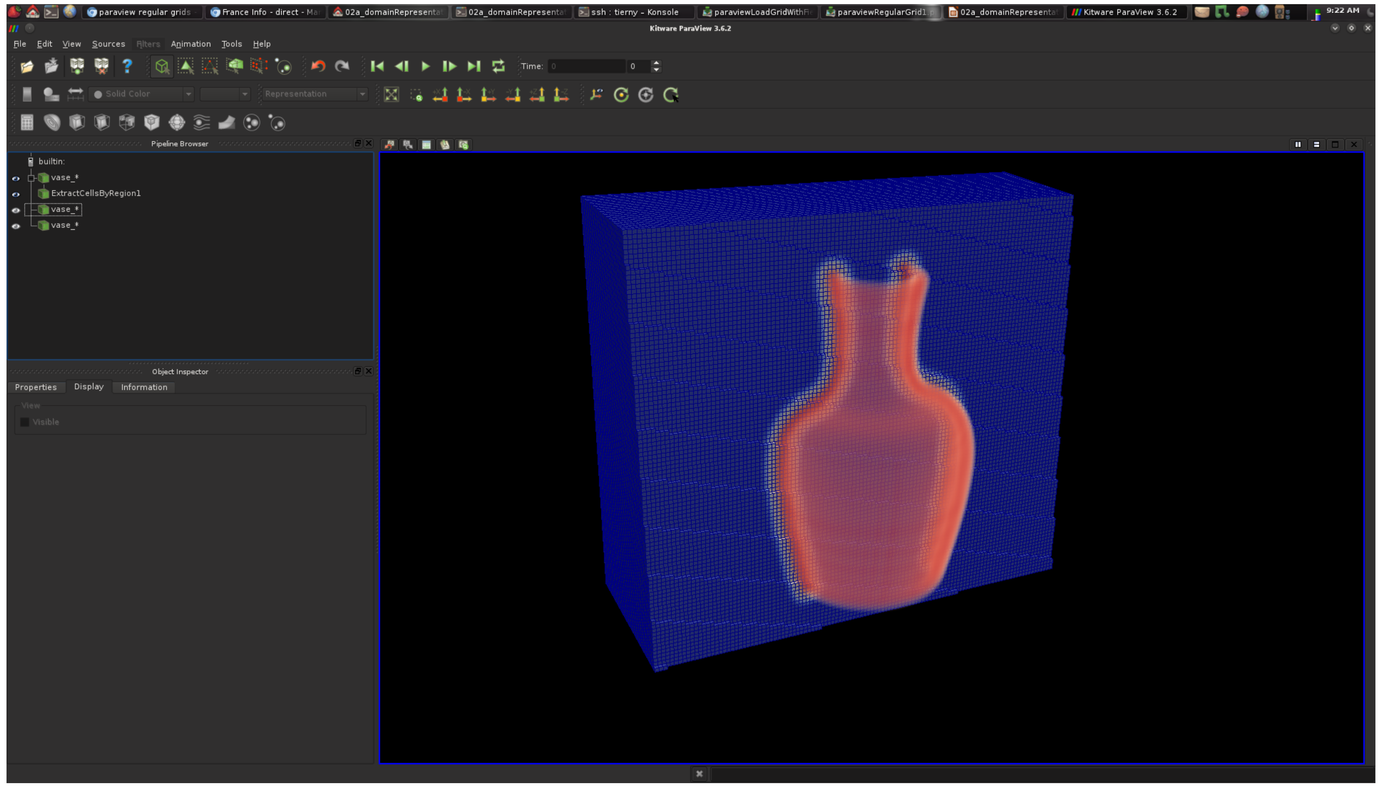
- What about regular grids?
  - In principle
    - Same rationale



Level sets on regular grids

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

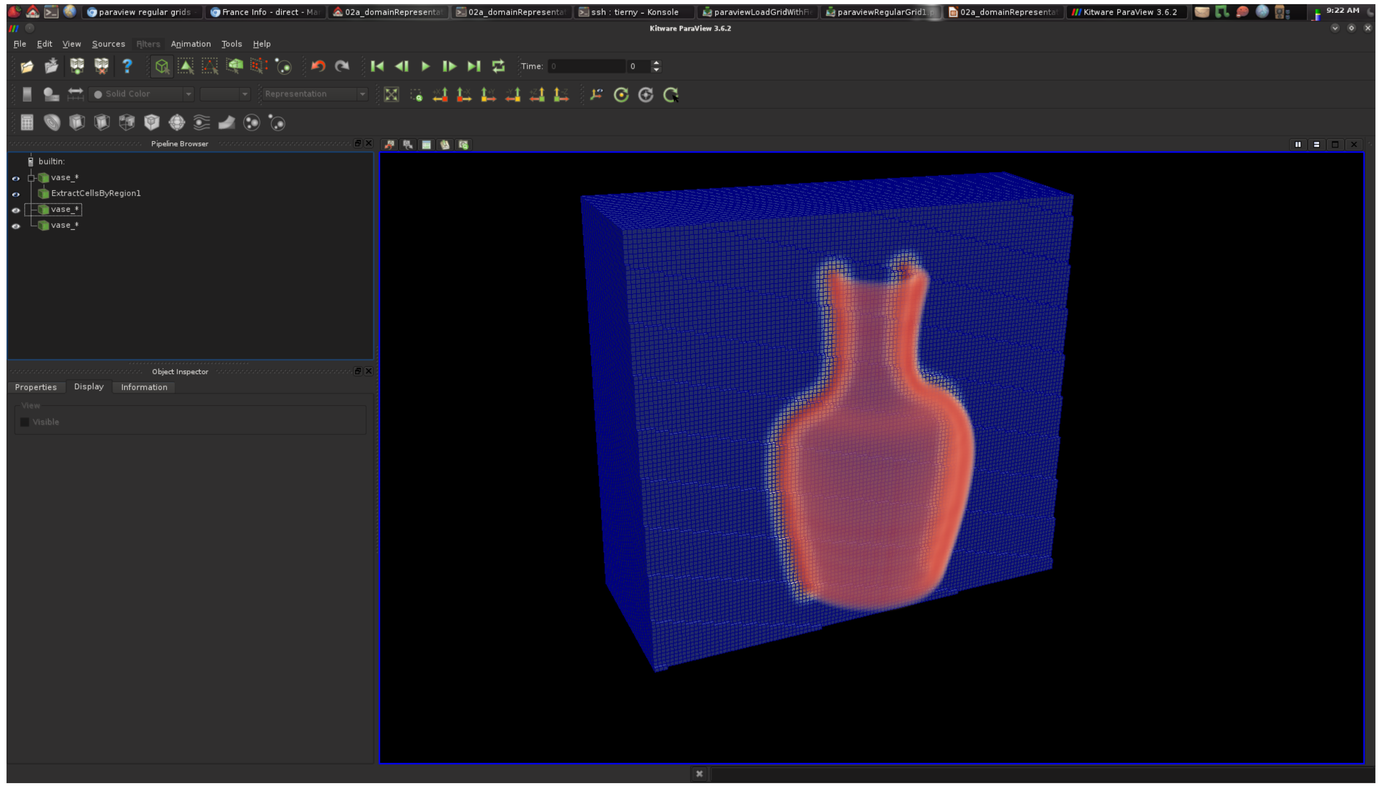
- 2D regular grids:
  - Marching squares



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

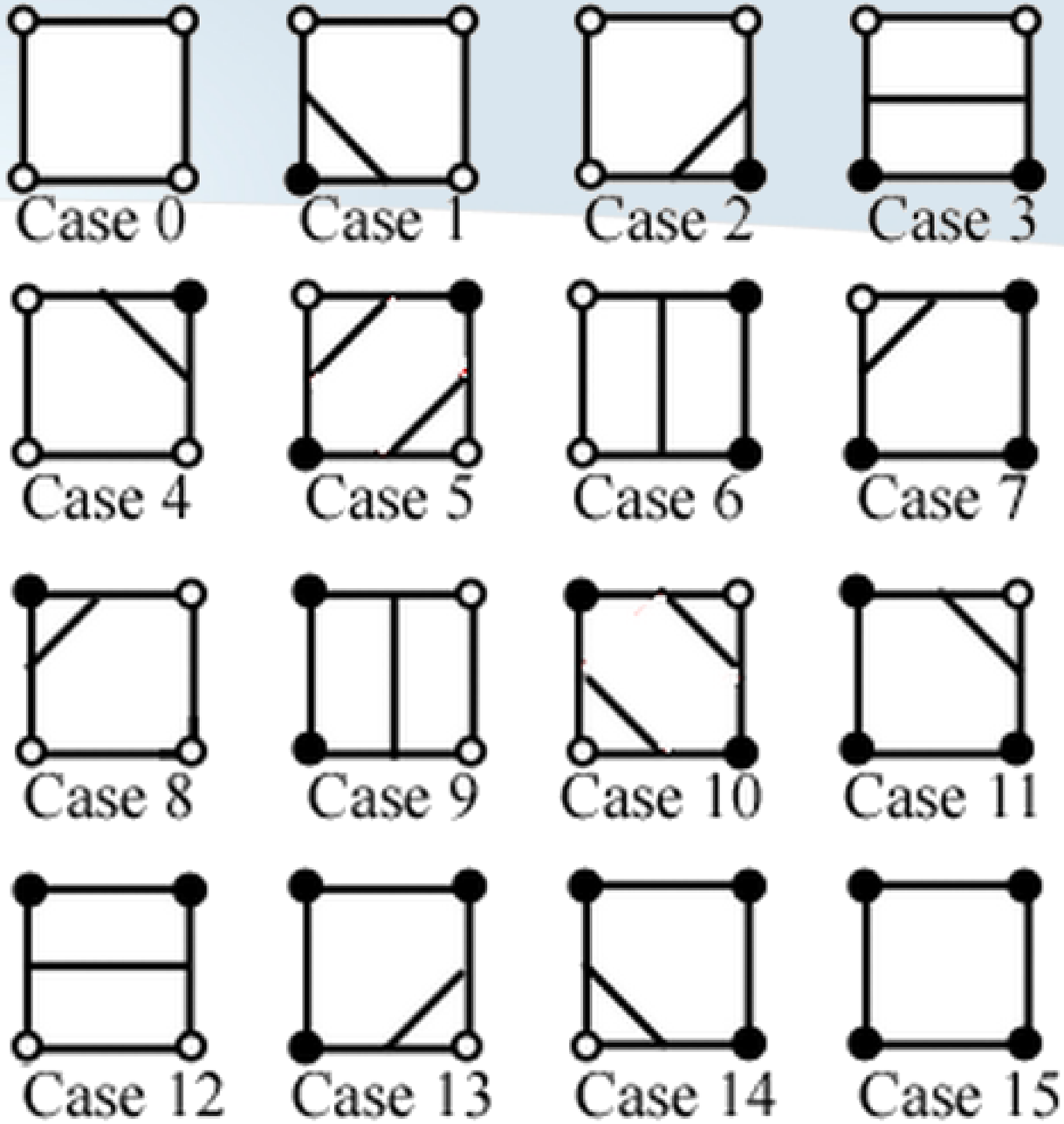
- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant

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

[Wikipedia]

Marching squares

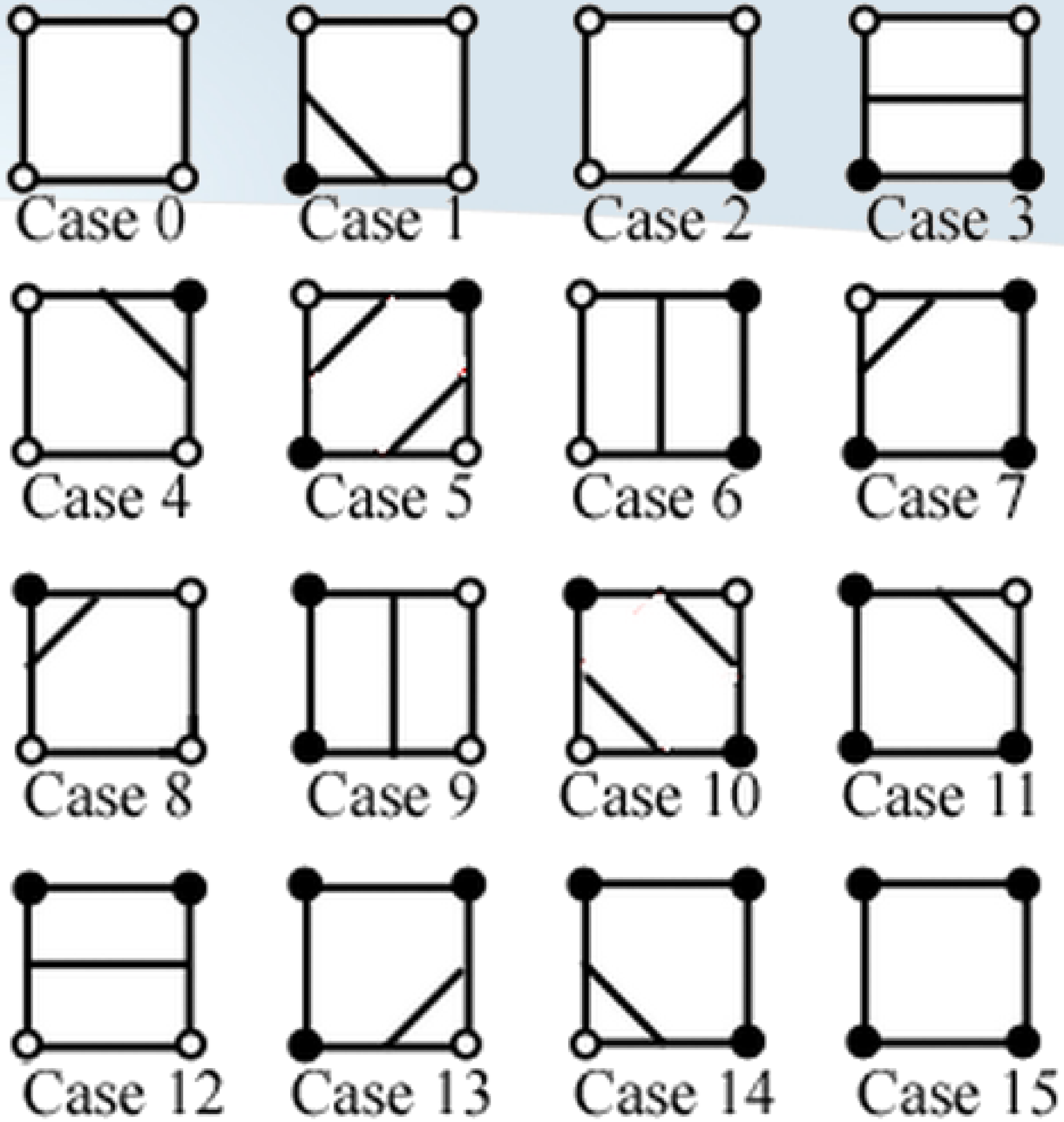
- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 2D unit cell



[Wikipedia]

Marching squares

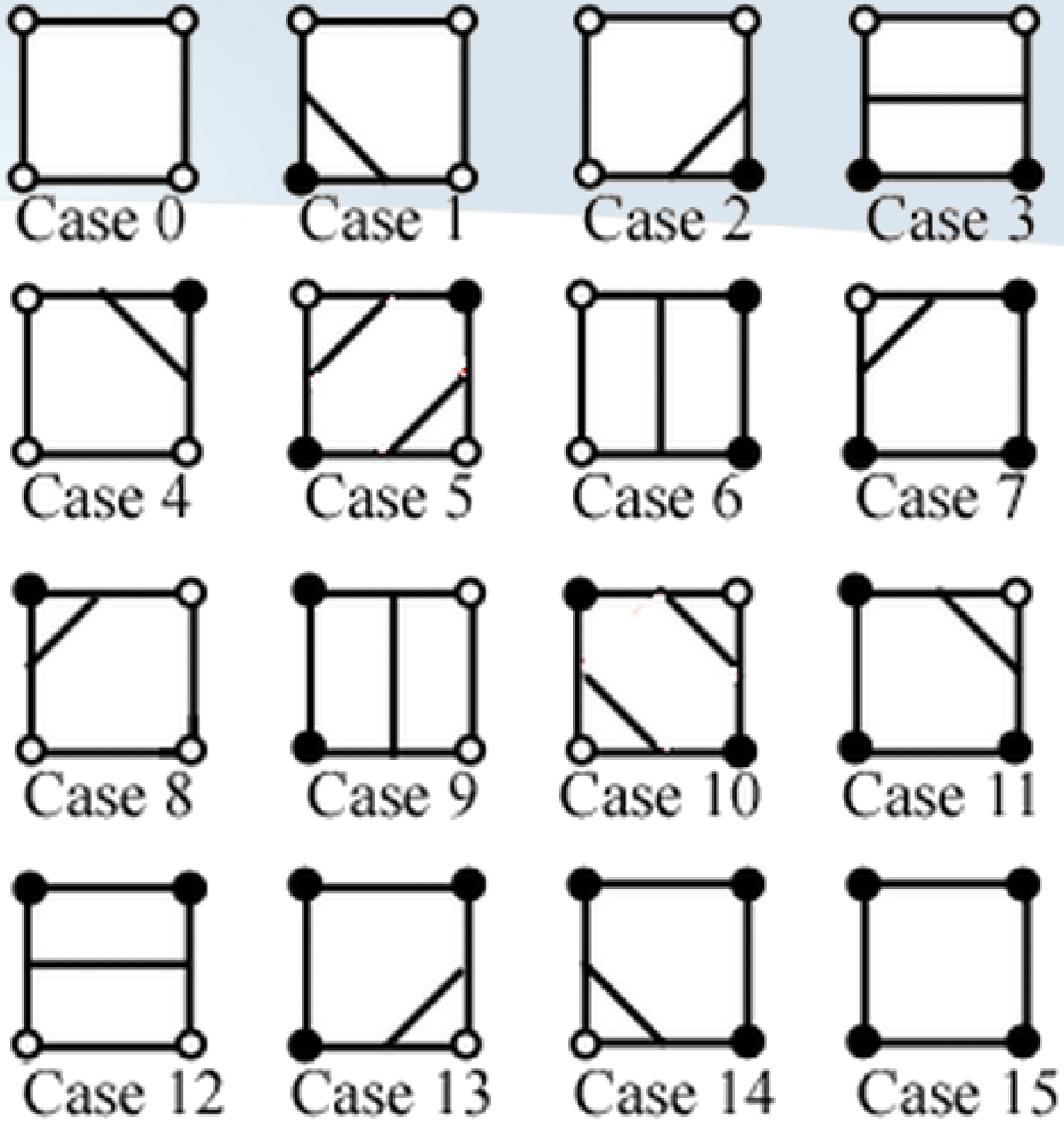
- Let $\mathcal{D}$ be a 2-regular grid
  - With bilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 2D unit cell
    - Much more than in a triangle



[Wikipedia]

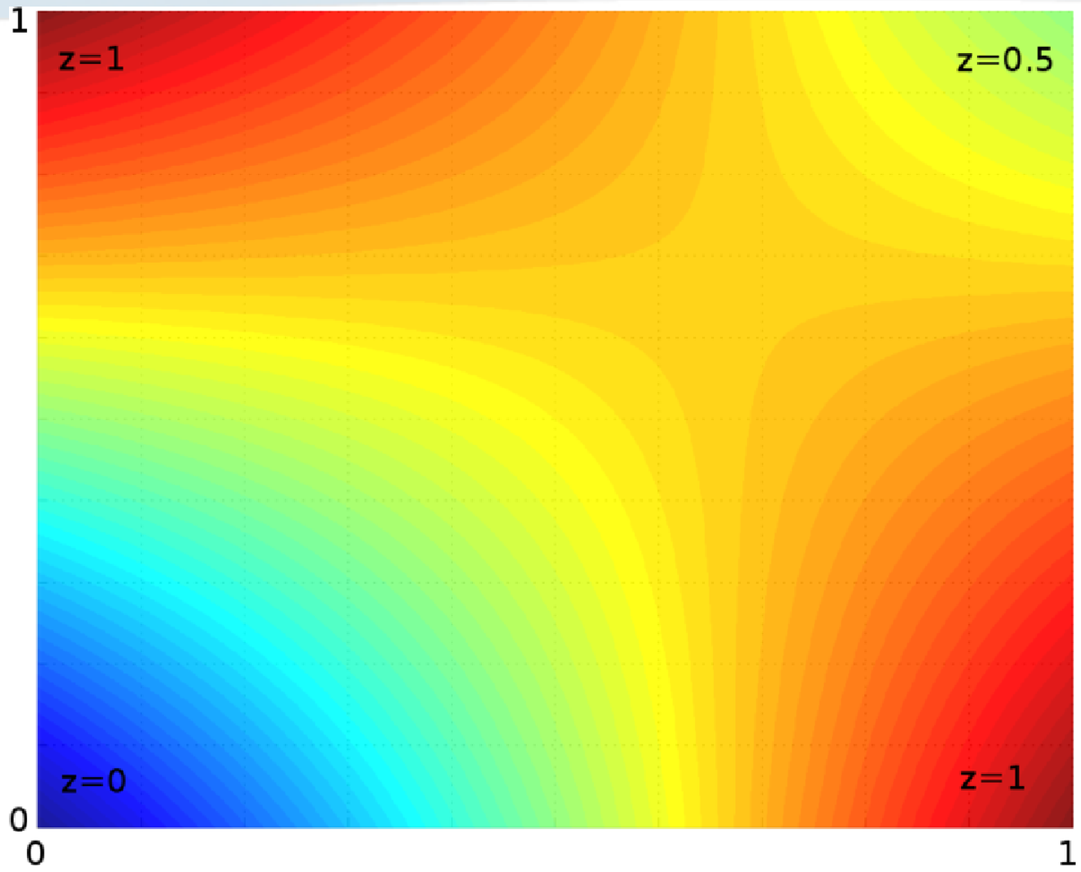
Marching squares

- 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!



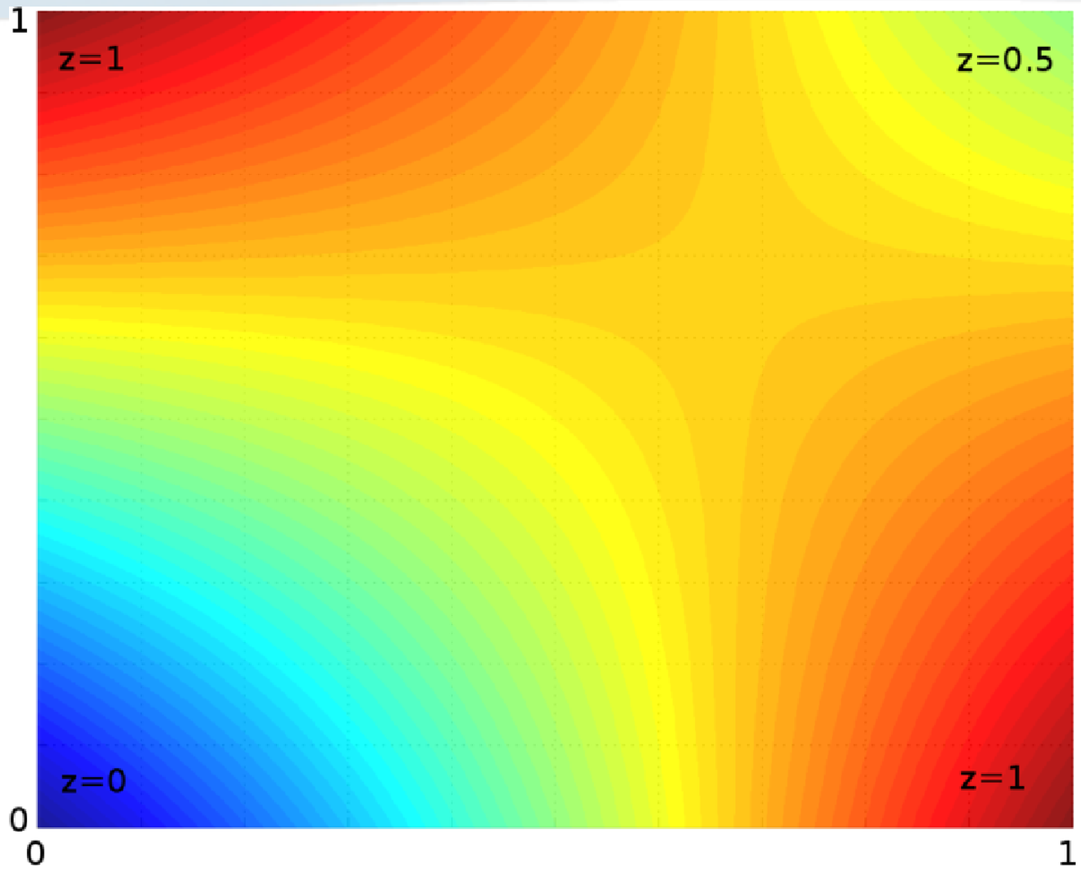
[Wikipedia]

Issues of marching squares



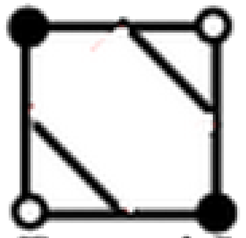
Issues of marching squares

$f^{-1}(0.95)$

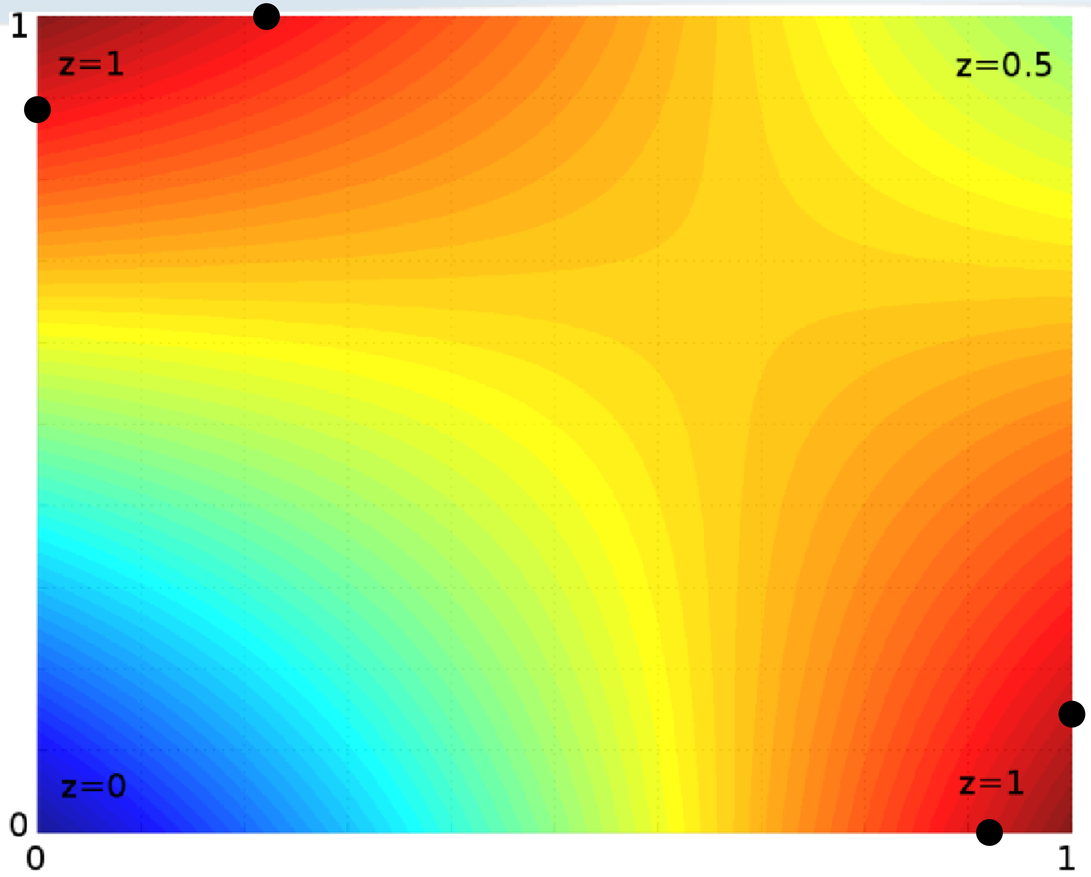


Issues of marching squares

$f^{-1}(0.95)$

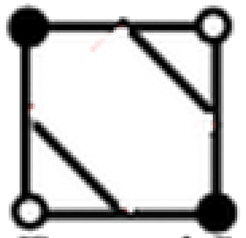


Case 10

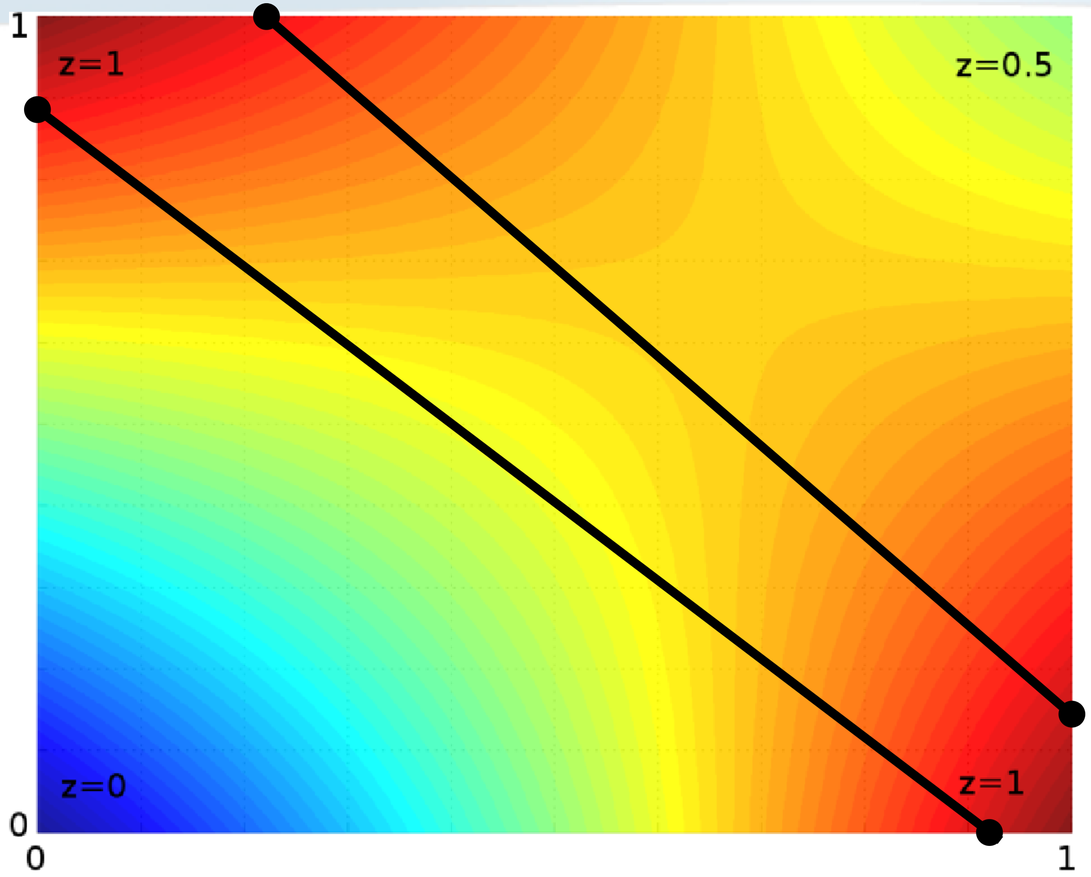


Issues of marching squares

$f^{-1}(0.95)$

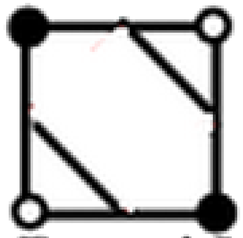


Case 10

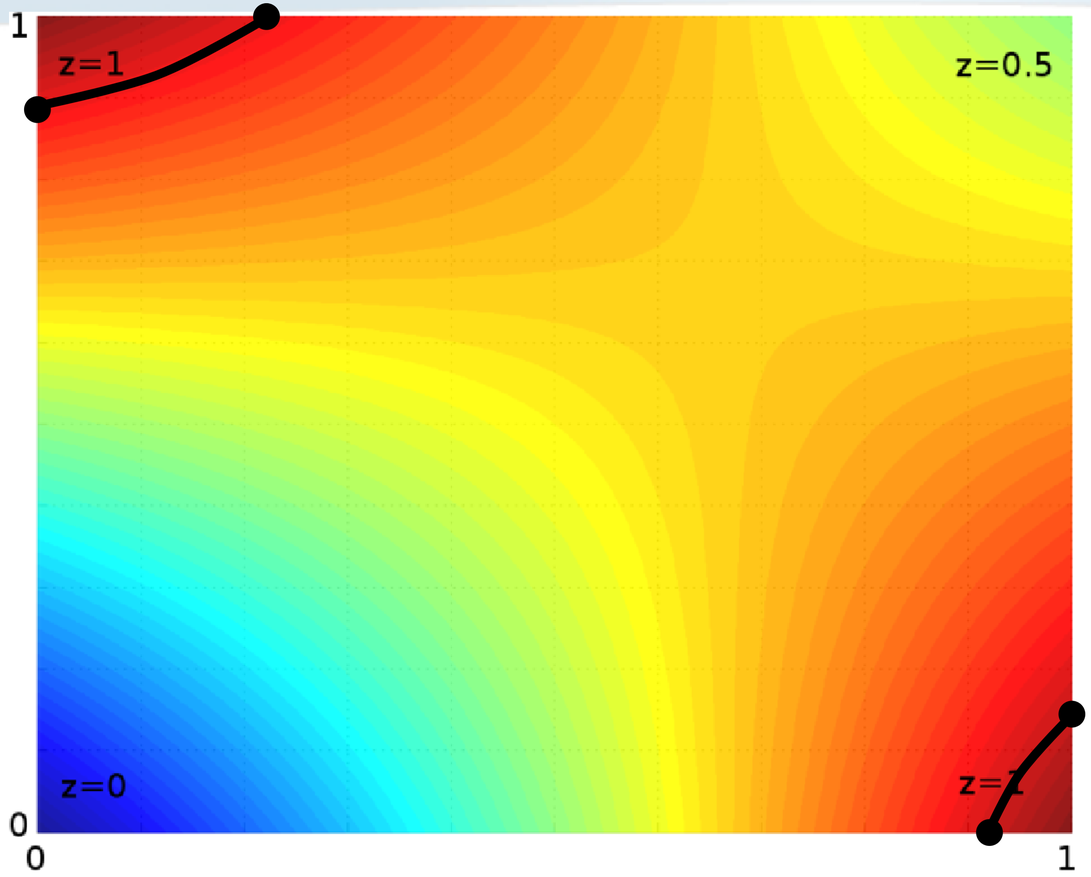


Issues of marching squares

$f^{-1}(0.95)$

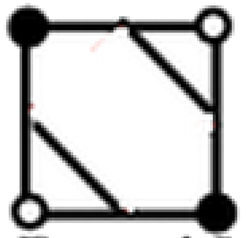


Case 10

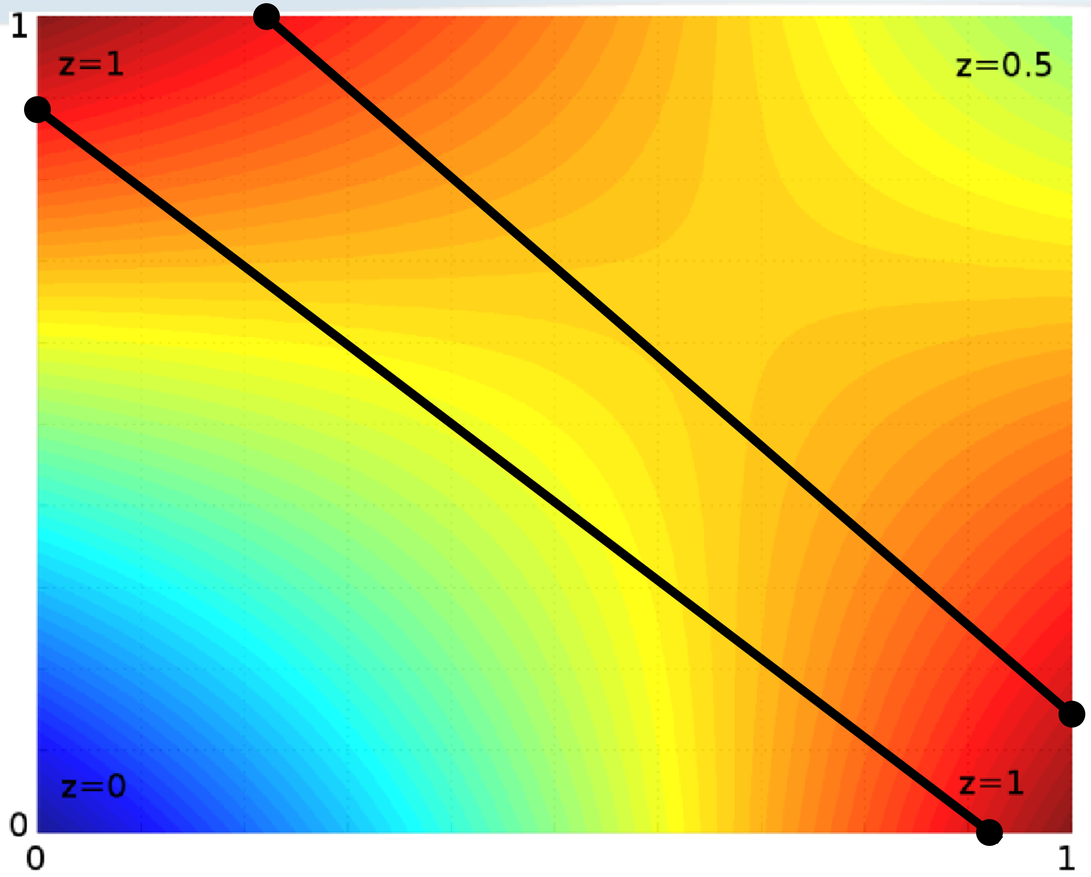


Issues of marching squares

$f^{-1}(0.95)$

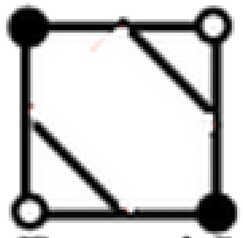


Case 10

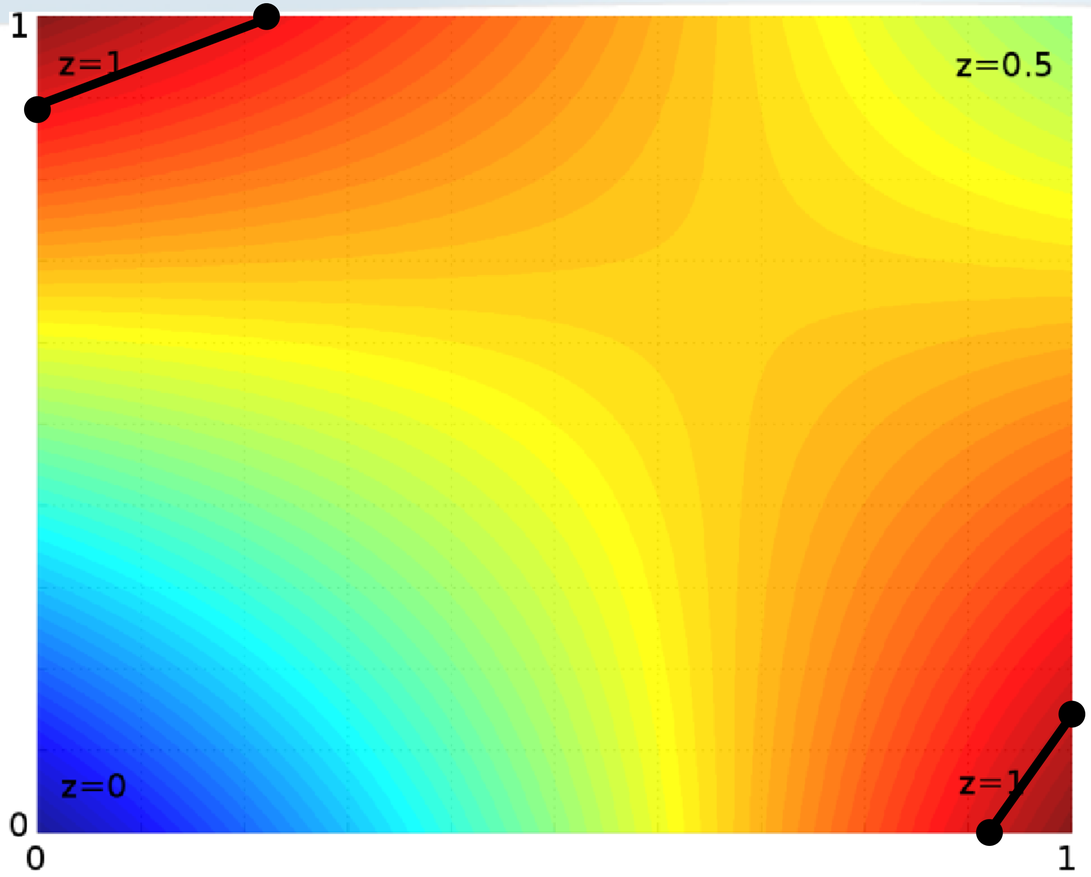


Issues of marching squares

$f^{-1}(0.95)$

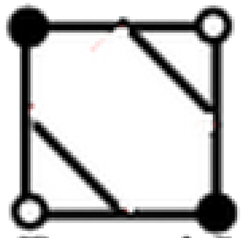


Case 10

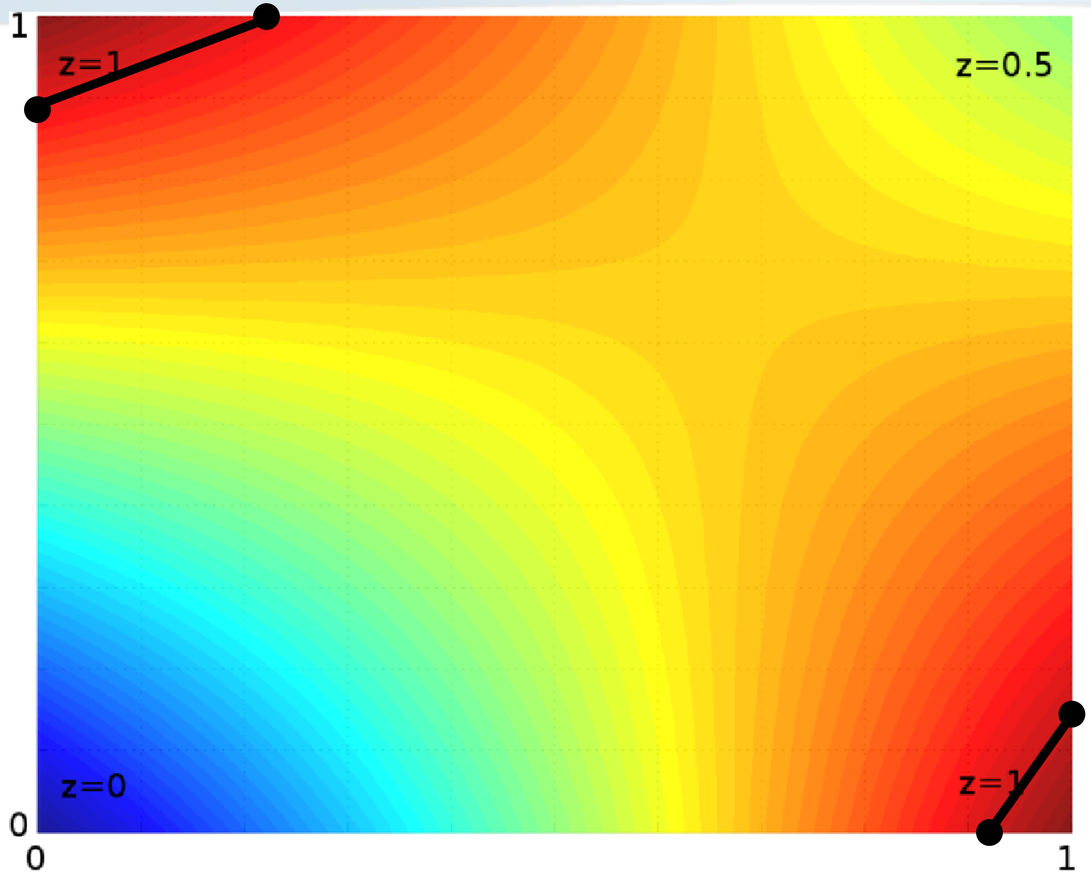


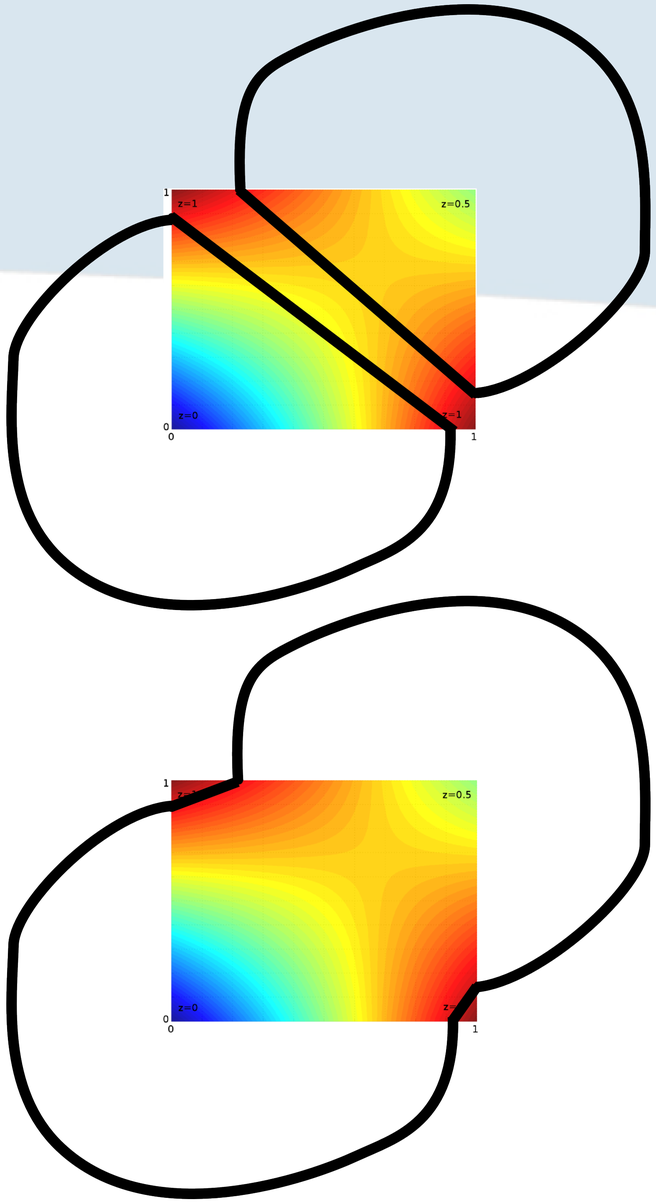
Issues of marching squares

$f^{-1}(0.95)$



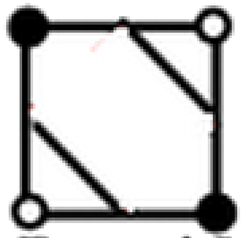
Case 10



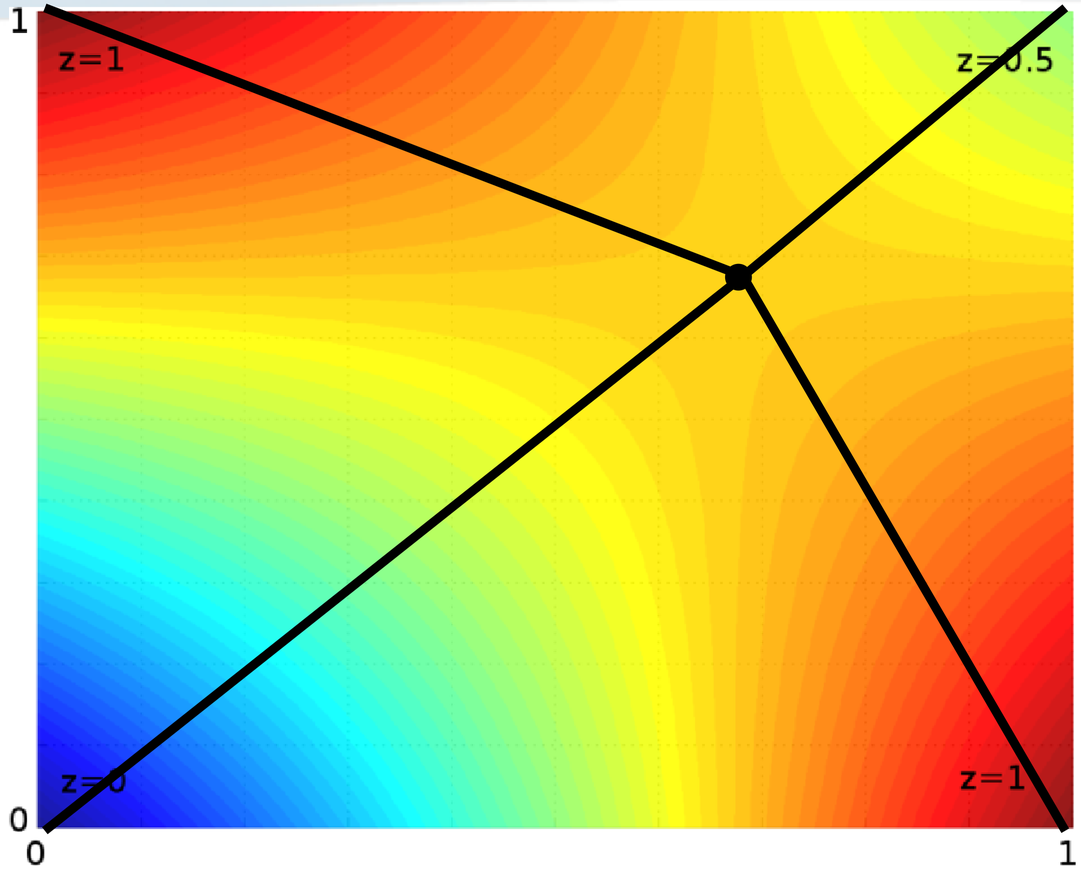


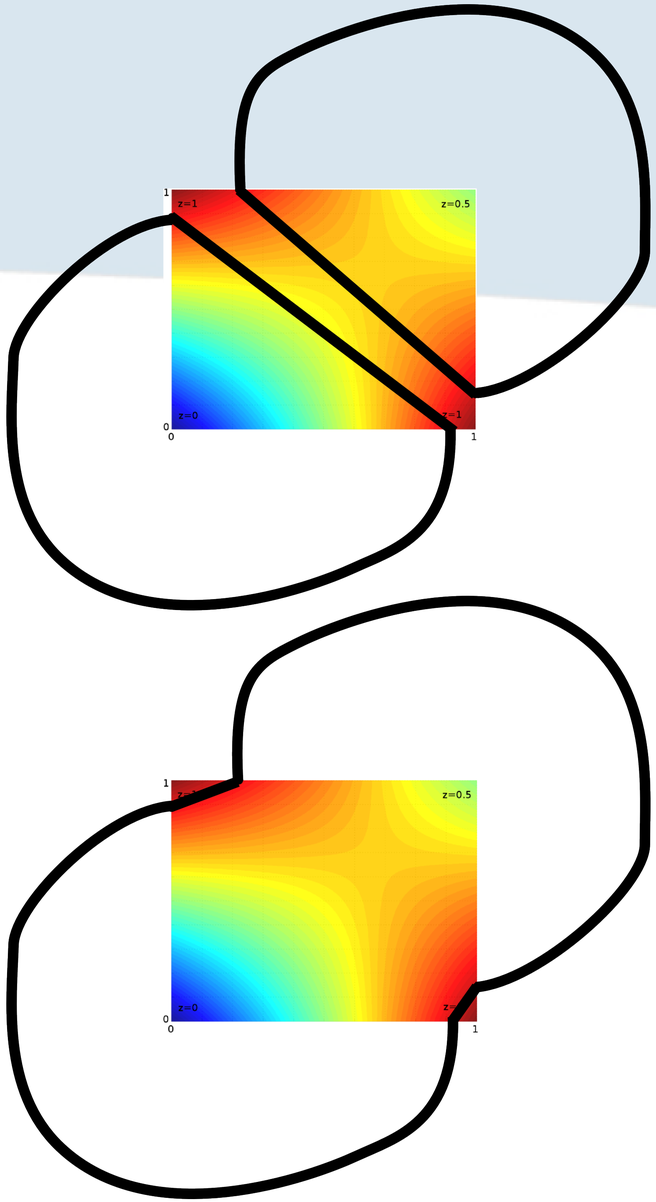
Issues of marching squares

$f^{-1}(0.95)$



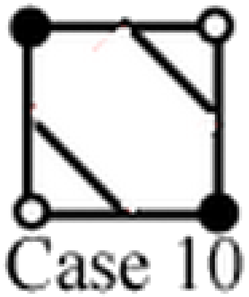
Case 10

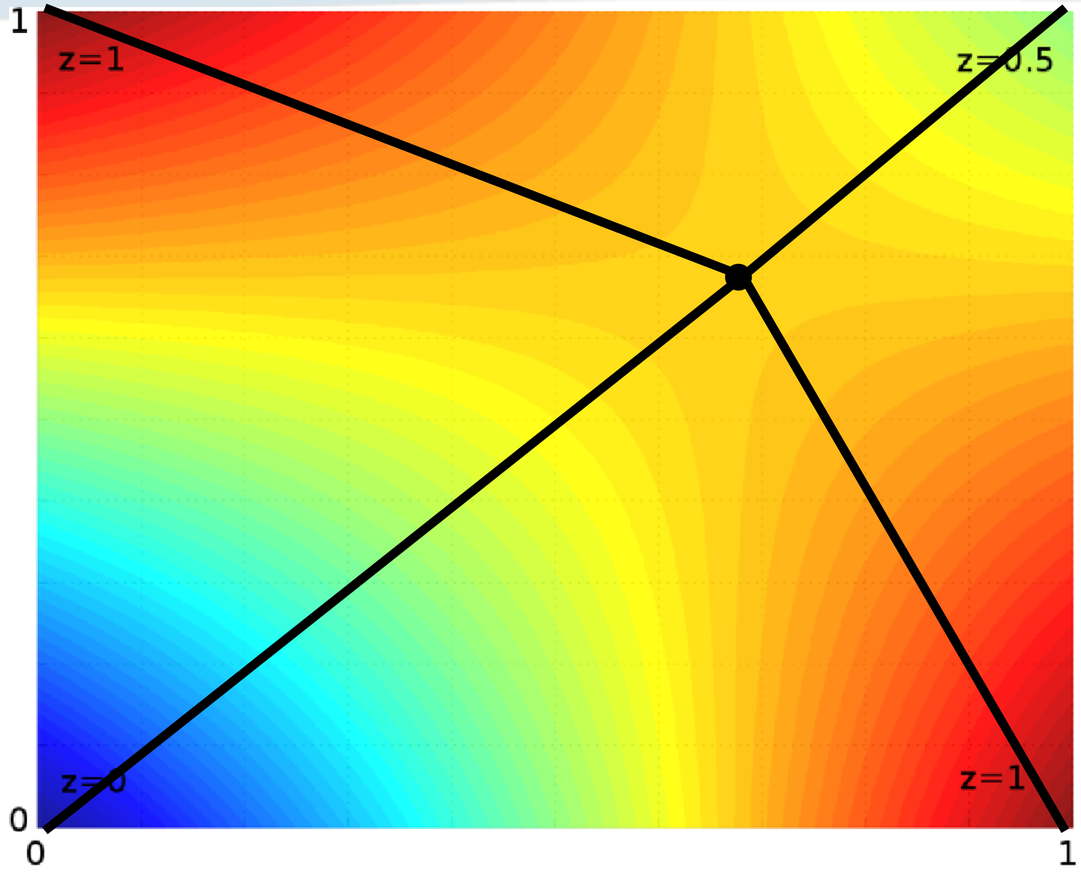


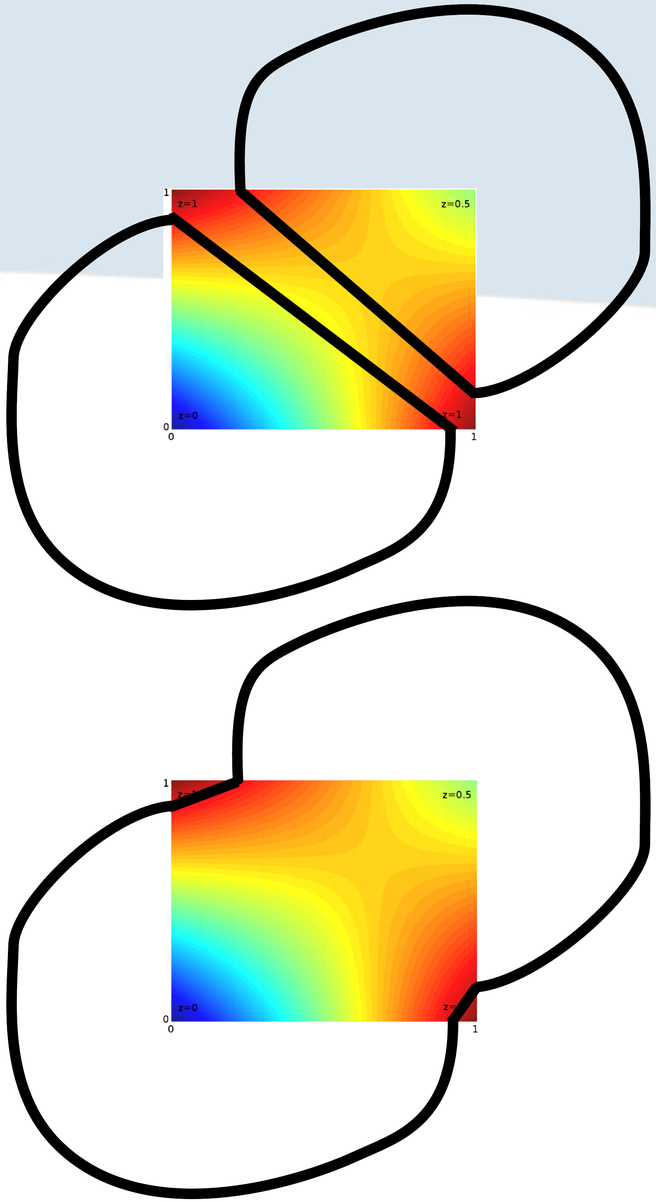


Issues of marching squares

$f^{-1}(0.95)$



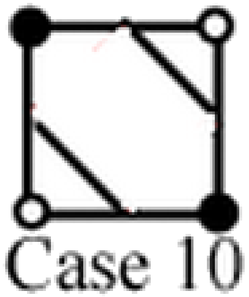


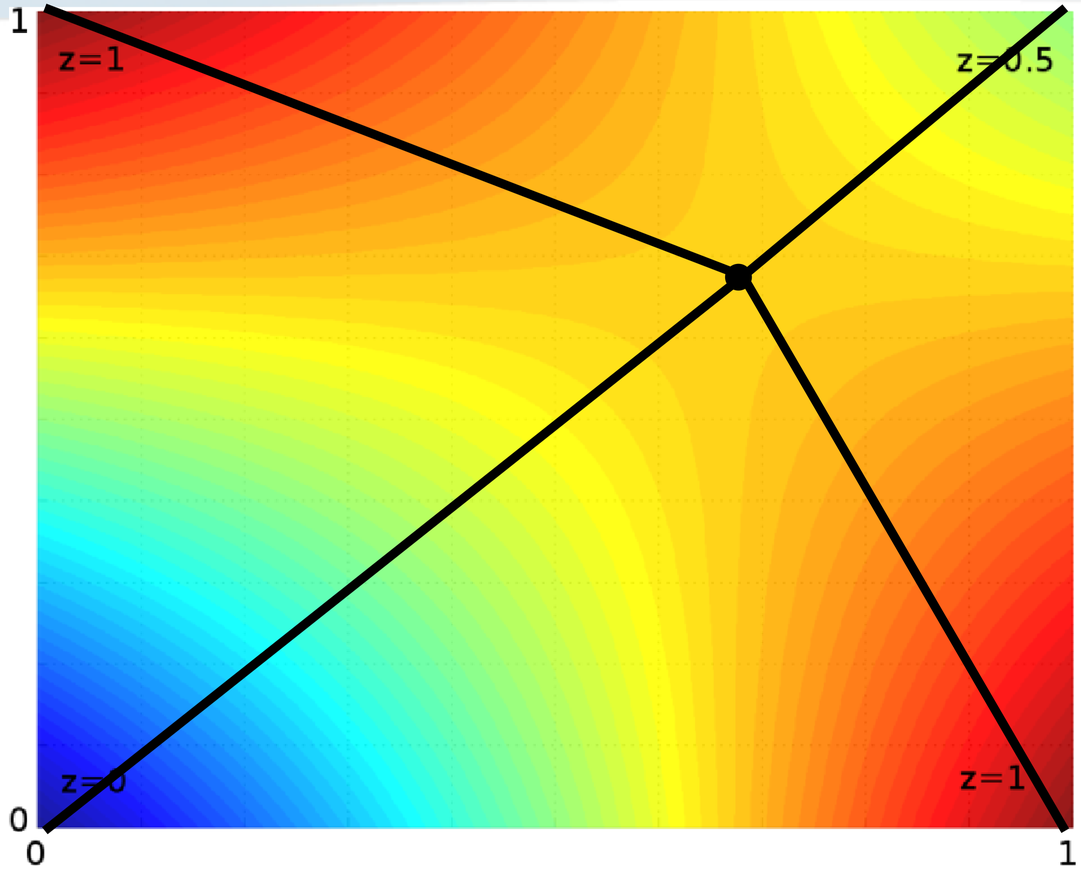


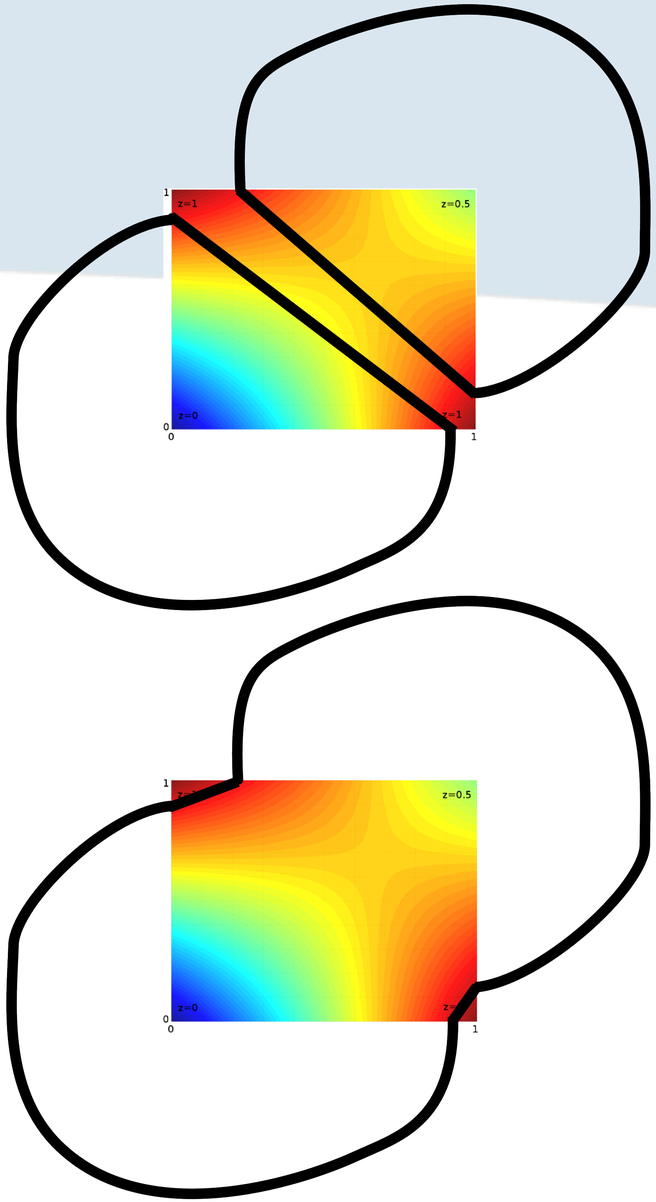
- Geometrically inaccurate (lines)

Issues of marching squares

$f^{-1}(0.95)$



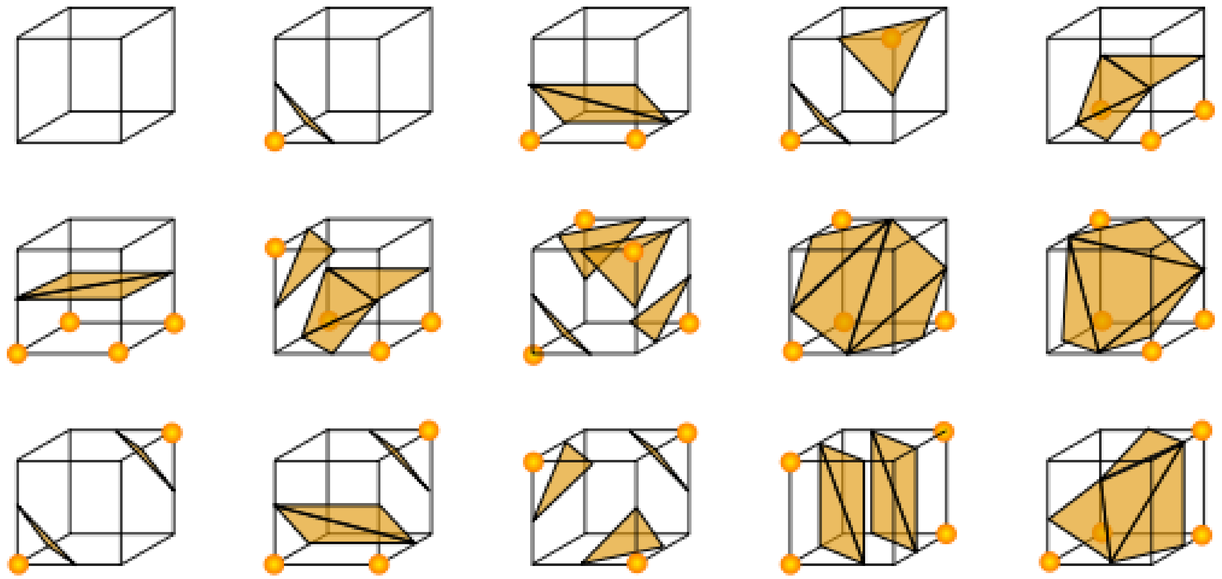




- Geometrically inaccurate (lines)
- Topologically inconsistent (**numerical** estimation of the saddle)

Marching cubes

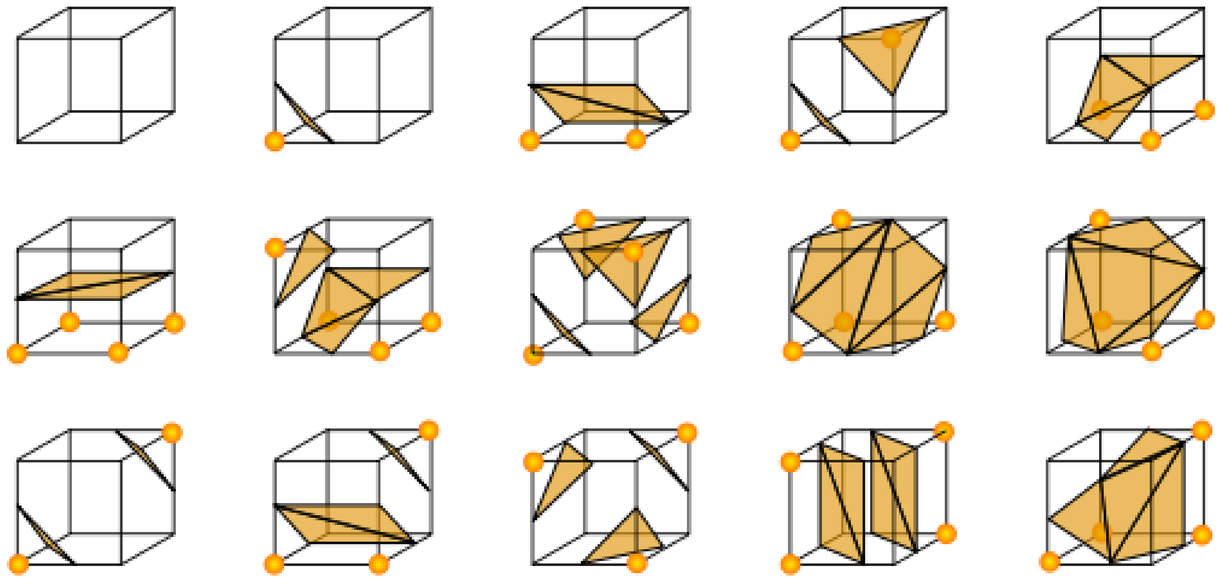
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant



[Wikipedia]

Marching cubes

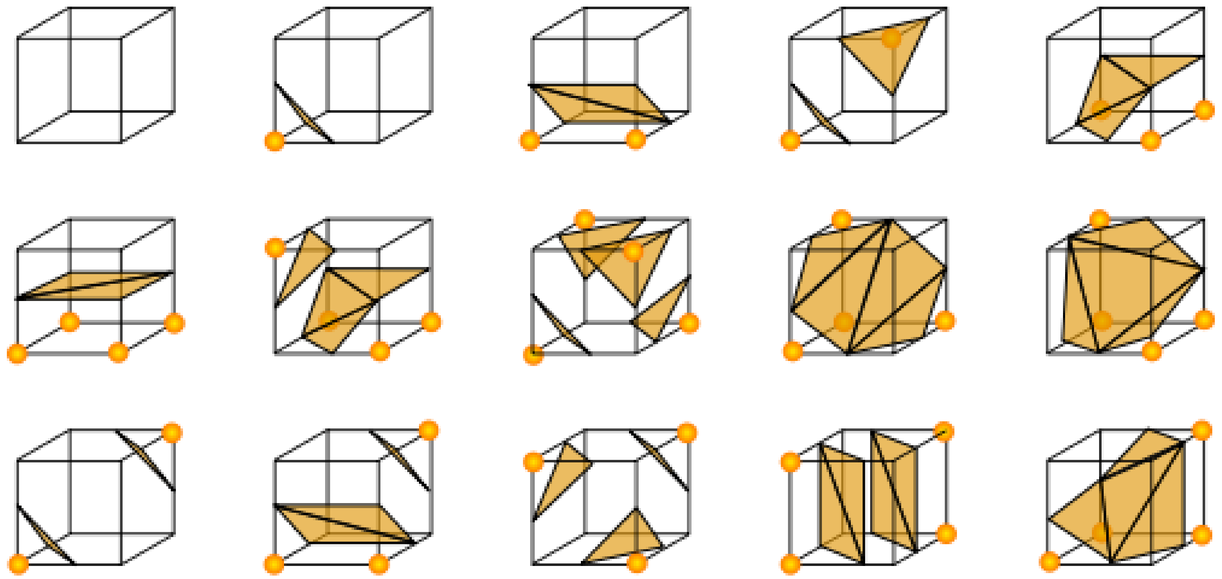
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells



[Wikipedia]

Marching cubes

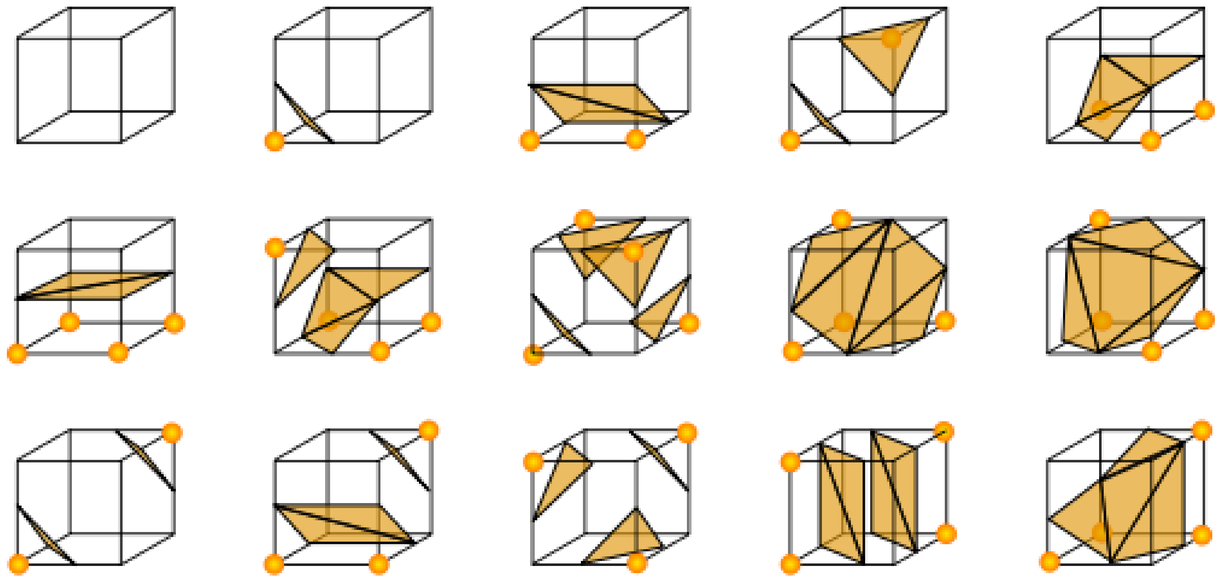
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 3D unit cell



[Wikipedia]

Marching cubes

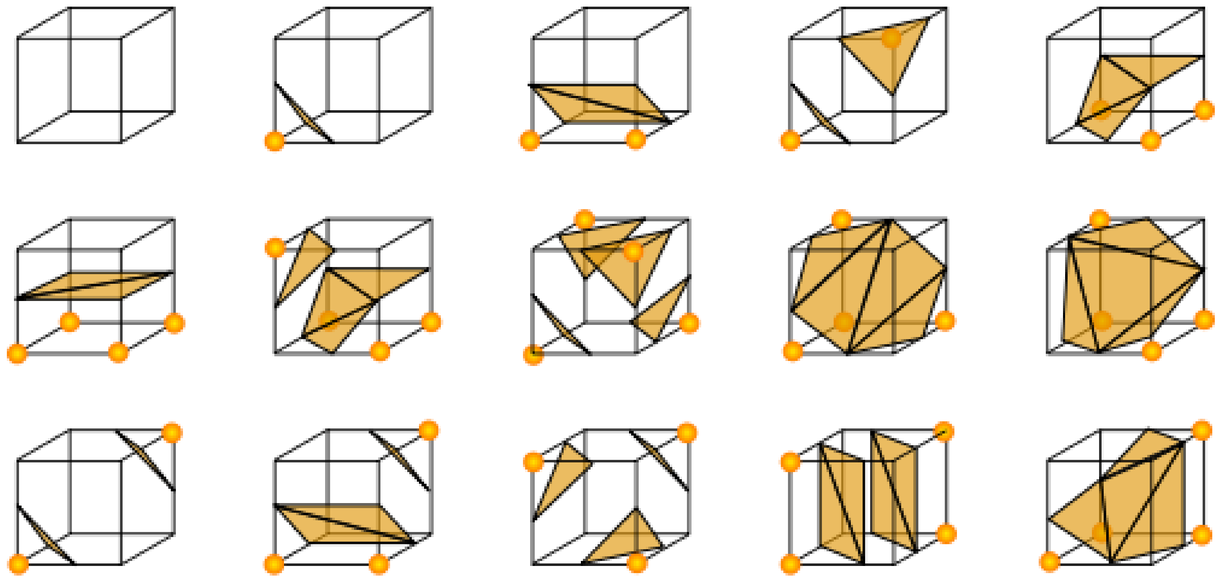
- Let $\mathcal{D}$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 3D unit cell
    - 256 cases



[Wikipedia]

Issues of marching cubes

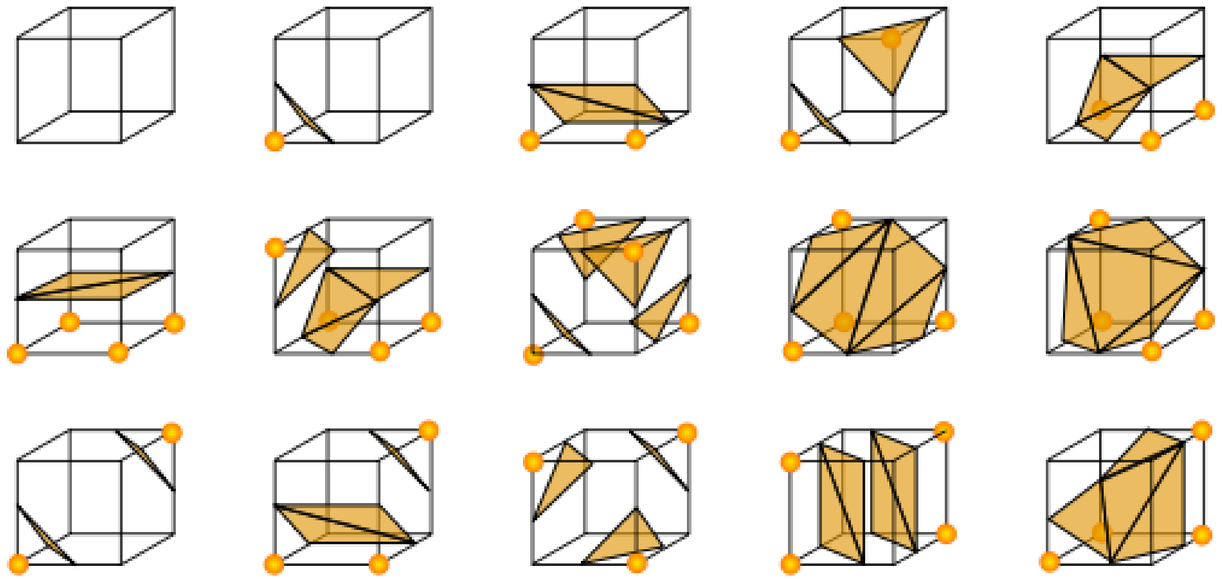
- Same as before



[Wikipedia]

Issues of marching cubes

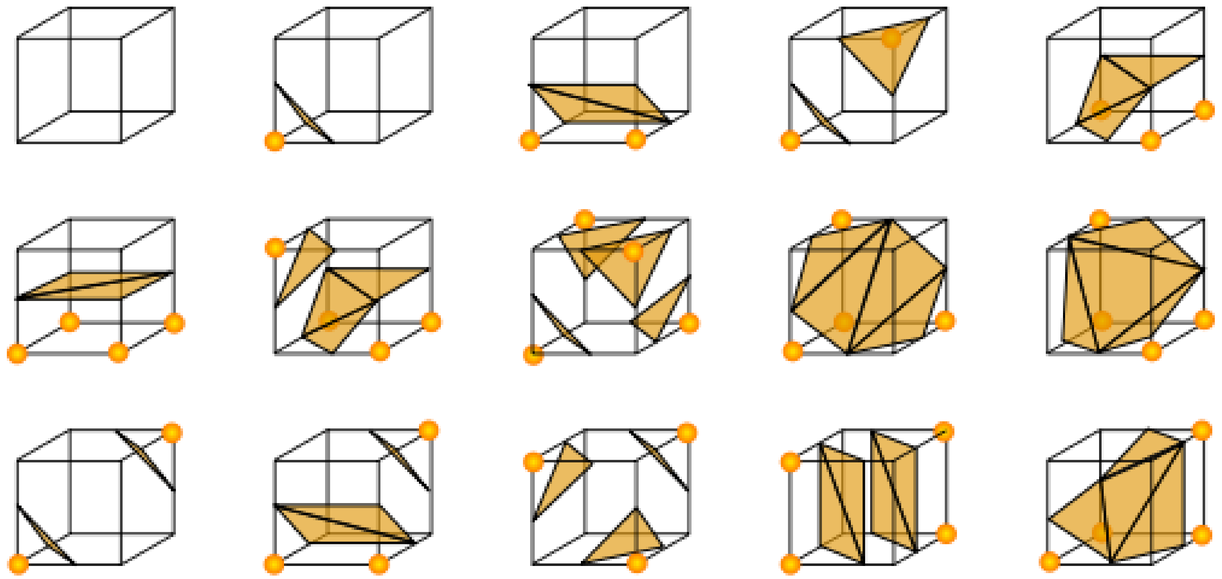
- Same as before
  - But worse!



[Wikipedia]

Issues of marching cubes

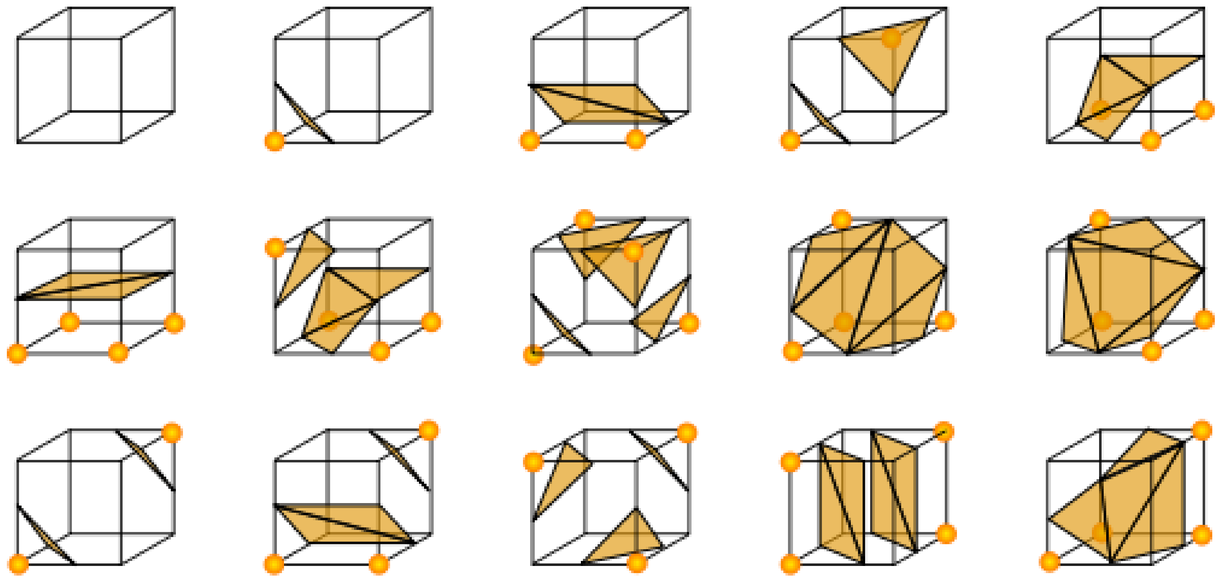
- Same as before
  - But worse!
- Lorensen and Cline 1987



[Wikipedia]

Issues of marching cubes

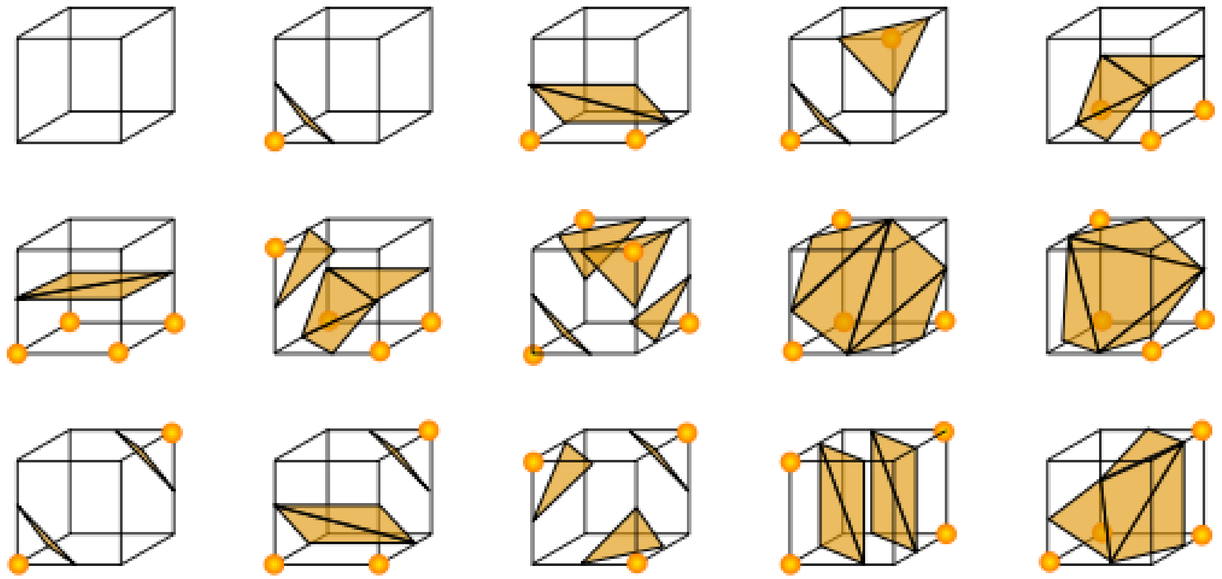
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)



[Wikipedia]

Issues of marching cubes

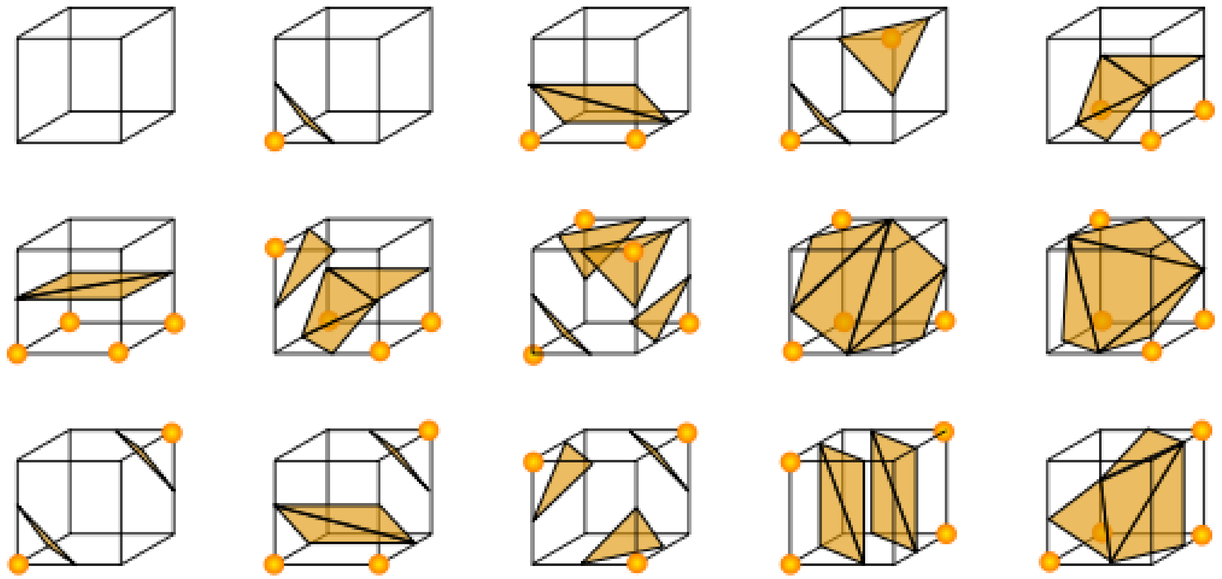
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks



[Wikipedia]

Issues of marching cubes

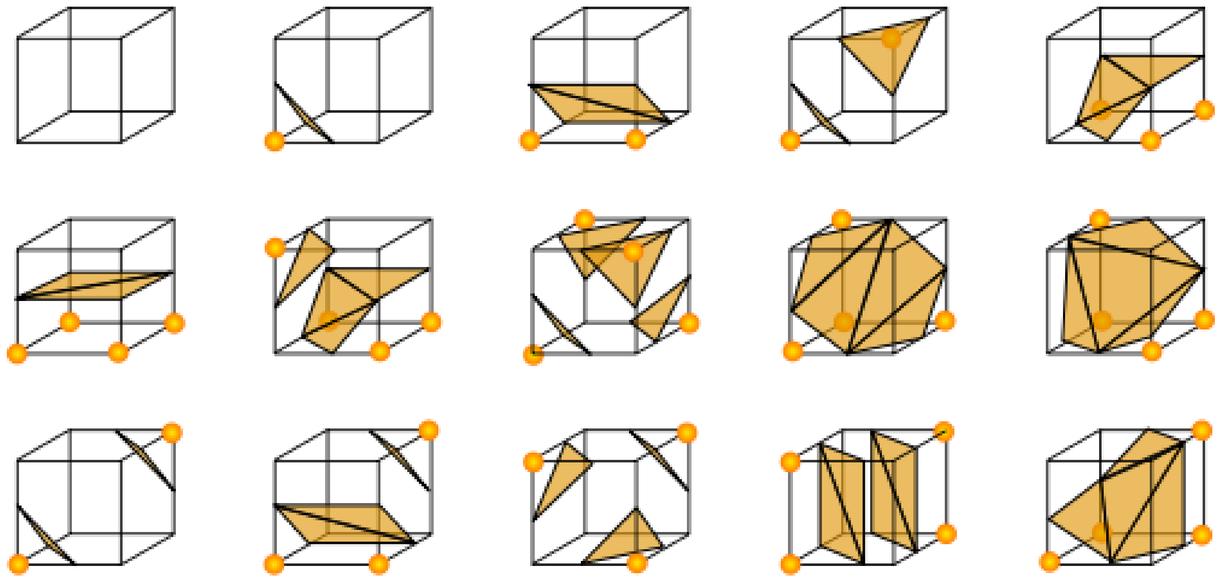
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases



[Wikipedia]

Issues of marching cubes

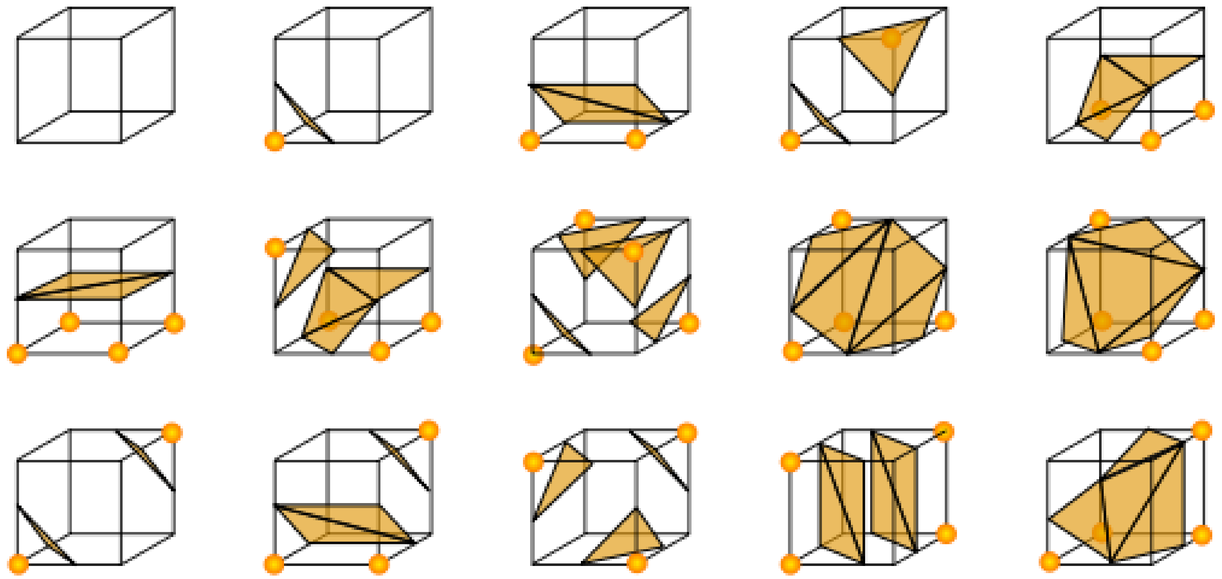
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases
- Saddle curse (components and genus):



[Wikipedia]

Issues of marching cubes

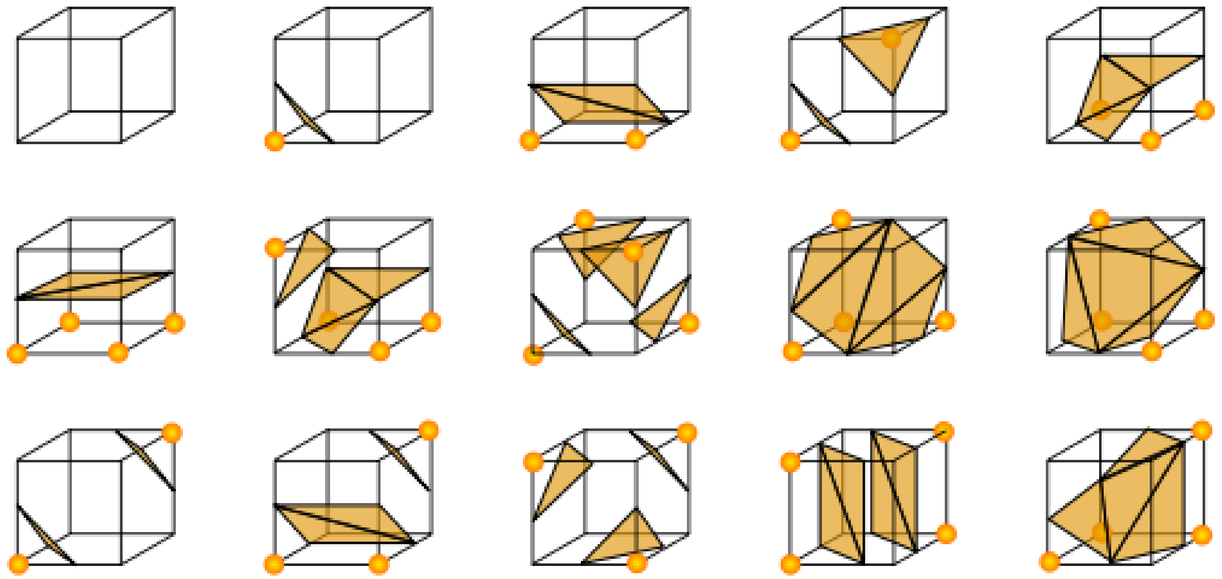
- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases
- Saddle curse (components and genus):
  - Nielson and Hamann 1991



[Wikipedia]

Issues of marching cubes

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

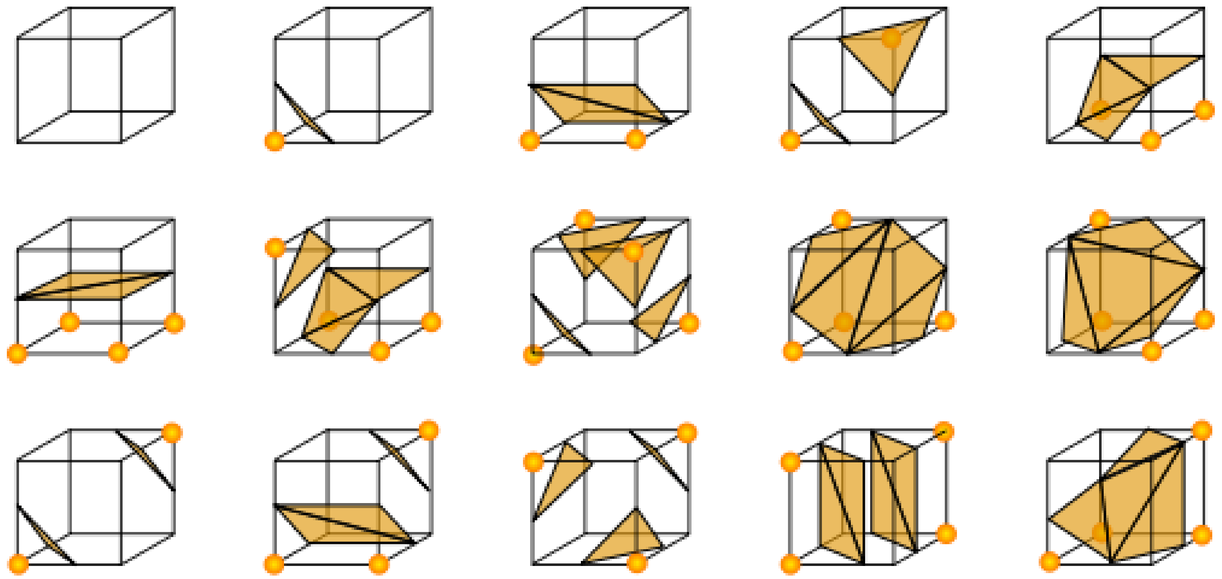


[Wikipedia]

- 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

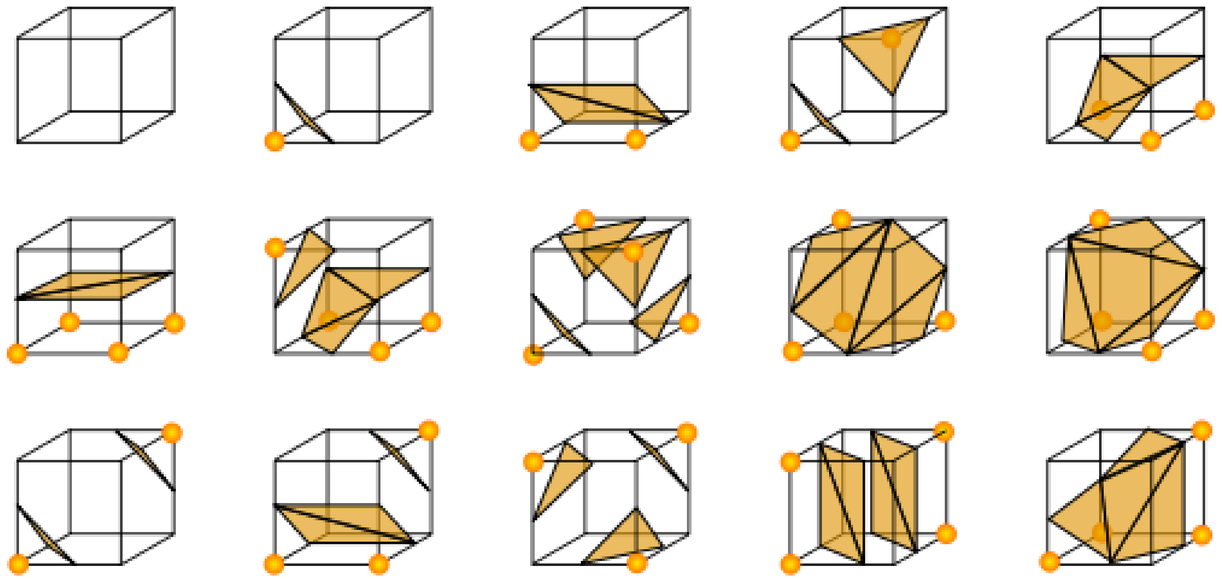


[Wikipedia]

- 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

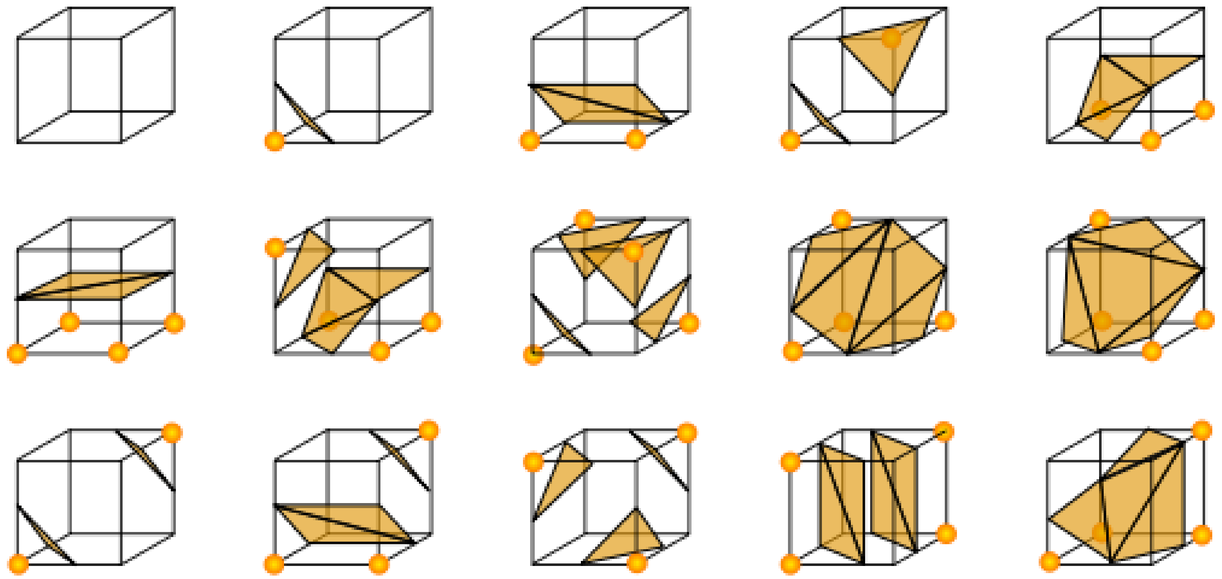


[Wikipedia]

- 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

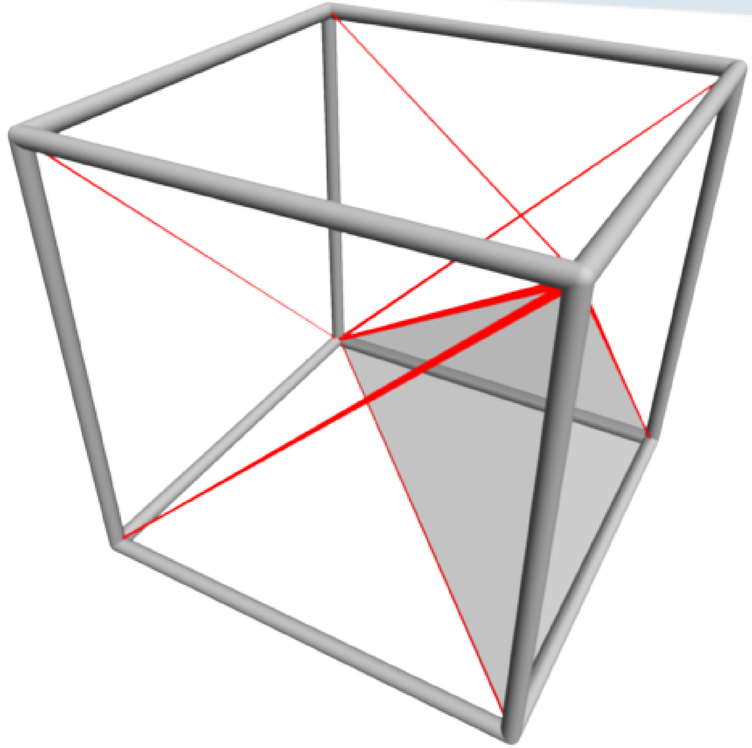


[Wikipedia]

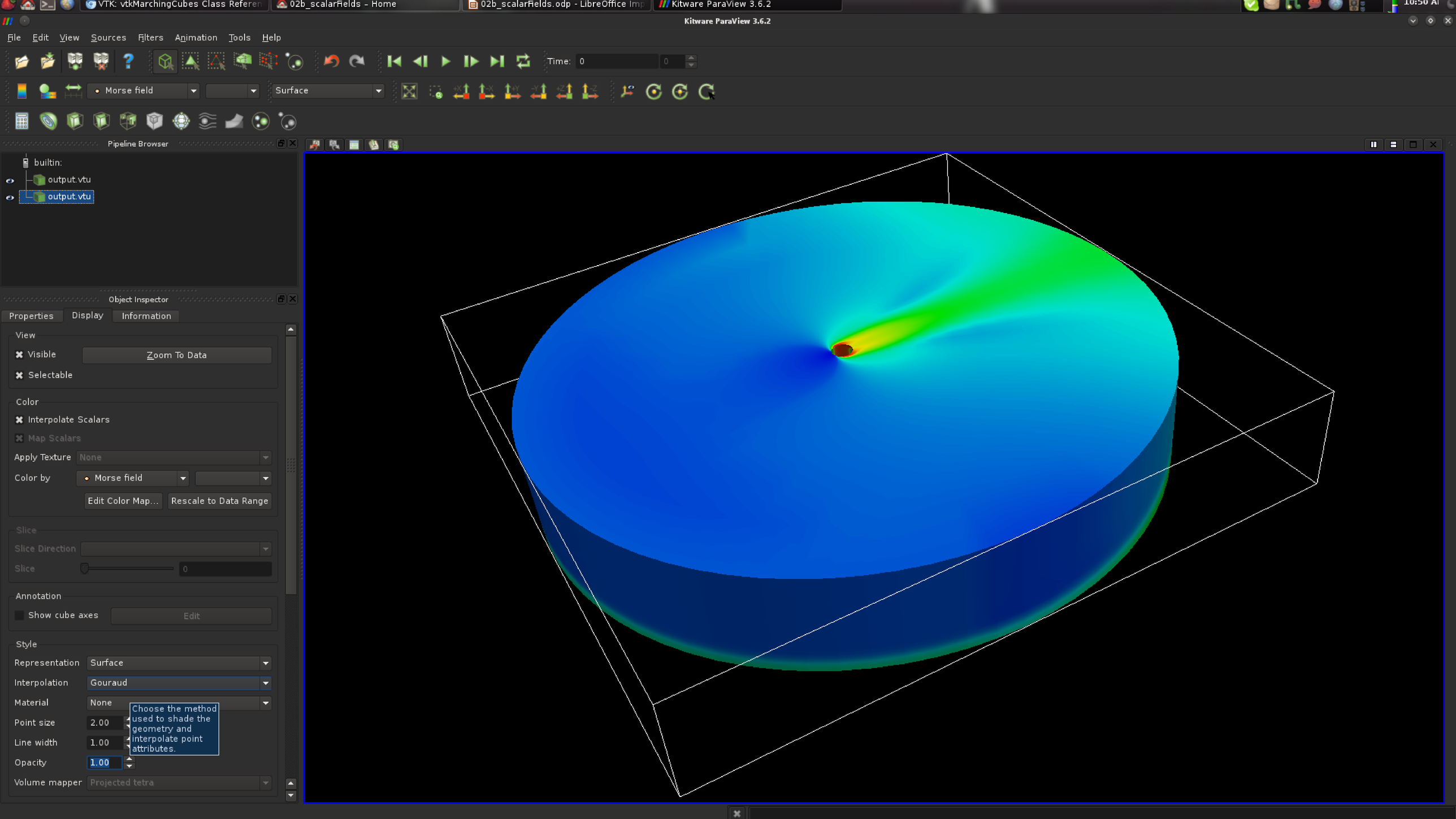
- Saddle curse (components and genus):
  - Nielson and Hamann 1991, Natarajan 1994, Chernyaev 1995, Lewiner 2003, *Etiene 2012*

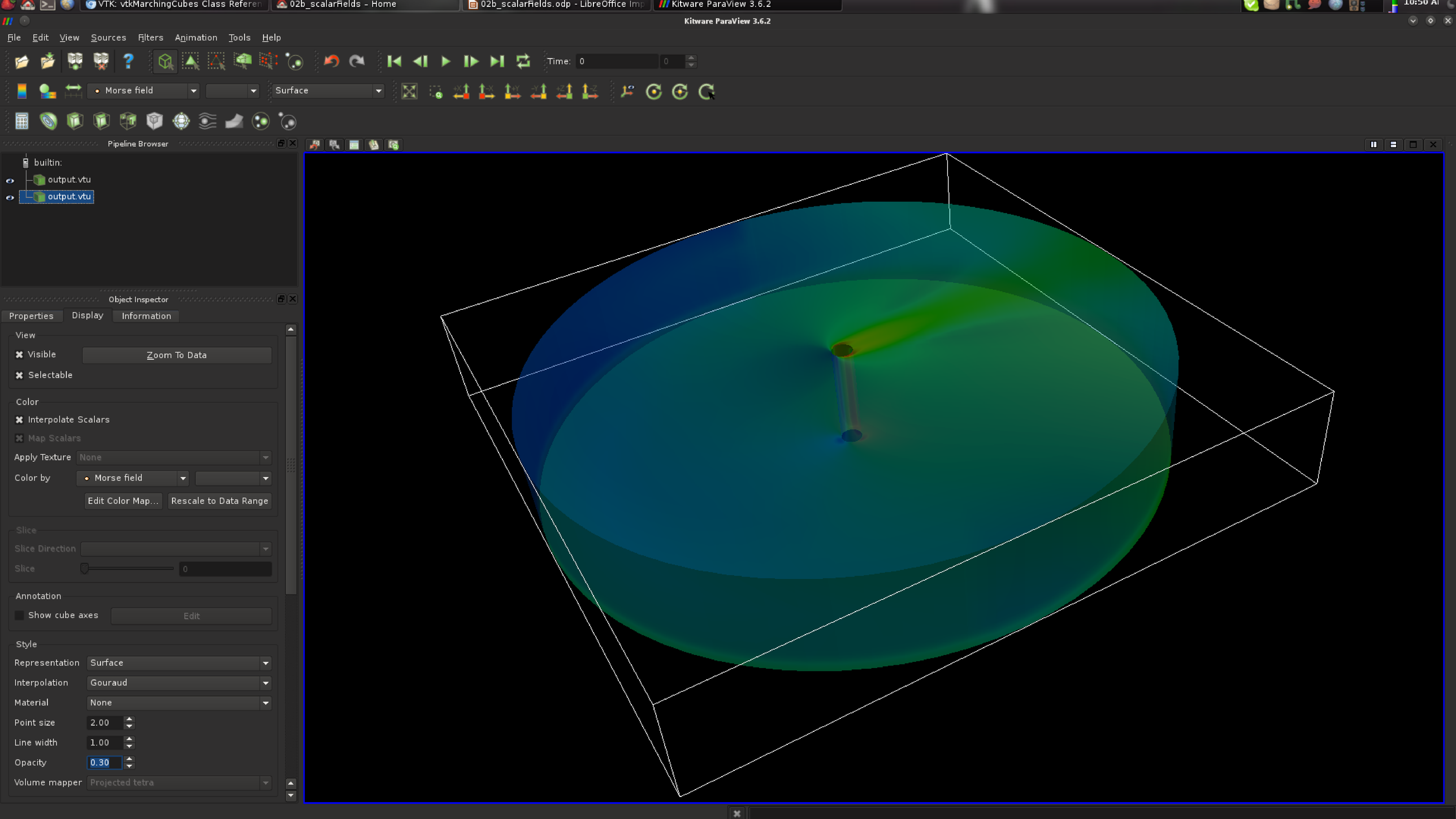
Issues of marching cubes

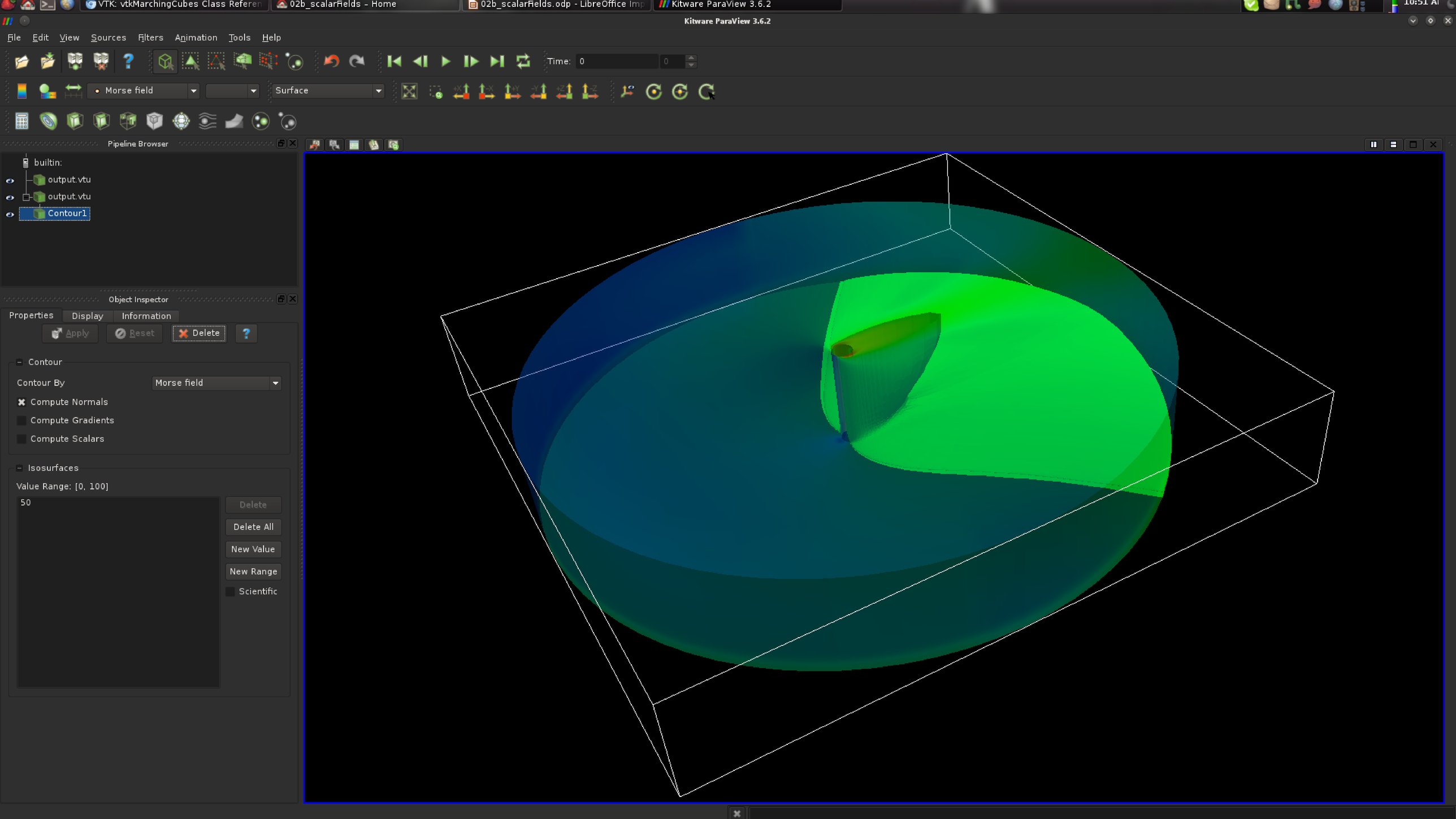
- Same as before
  - But worse!
- Lorensen and Cline 1987
    - 16 cases (symmetries)
    - Possible cracks
    - Improvement 28 cases
- Saddle curse (components and genus):
  - Nielson and Hamann 1991, Natarajan 1994, Chernyaev 1995, Lewiner 2003, *Etiene 2012*

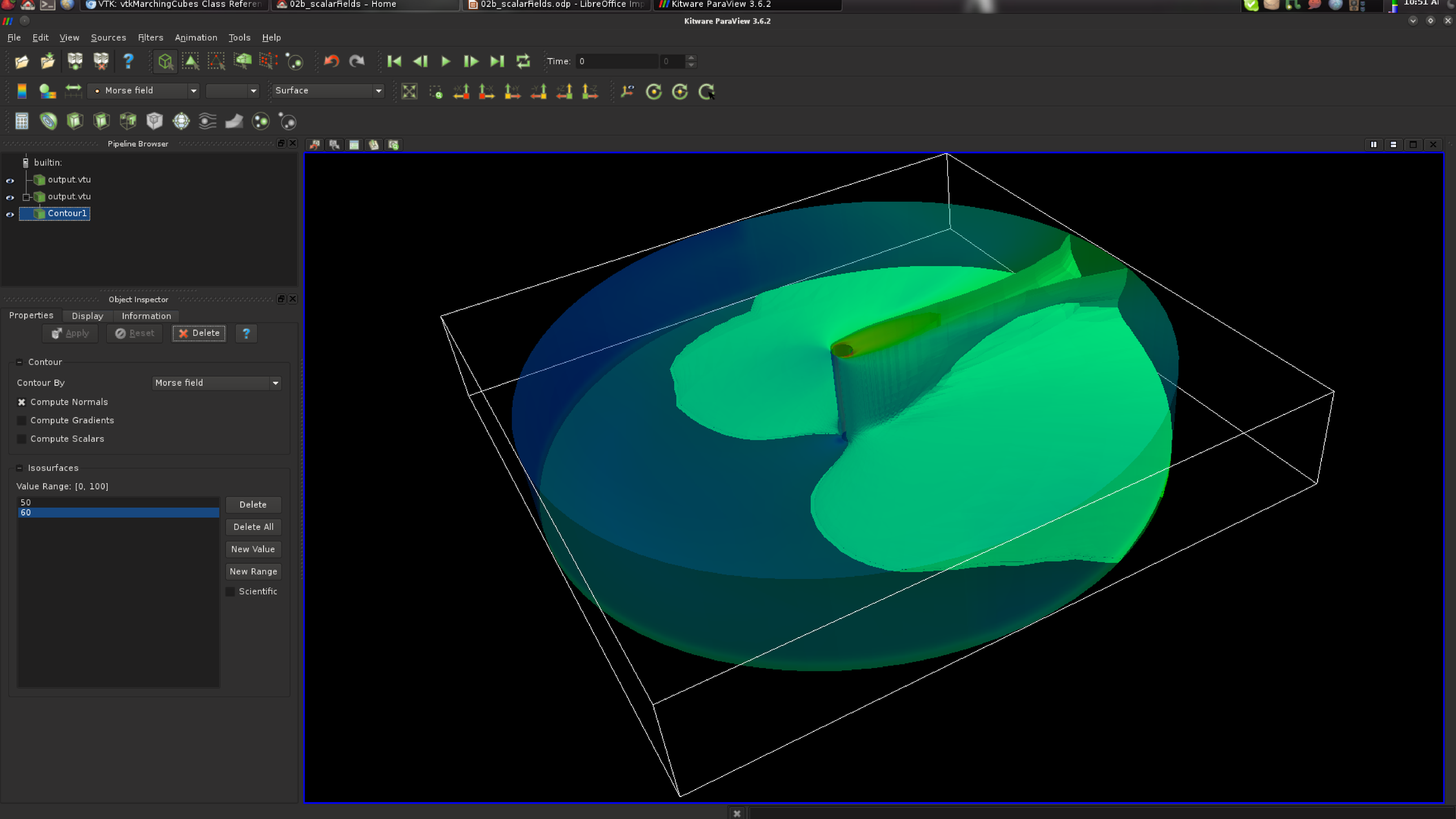


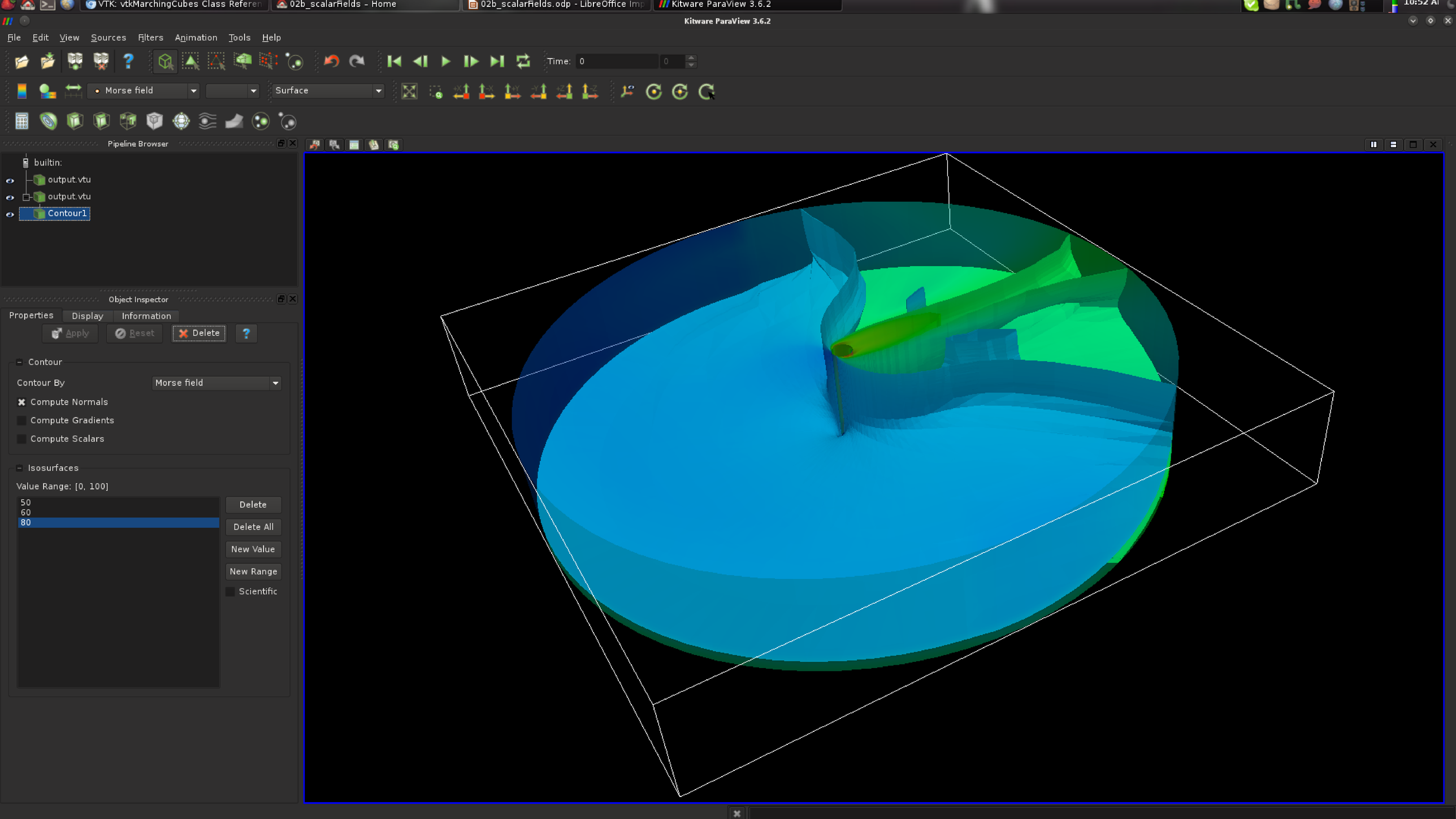
[Wikipedia]

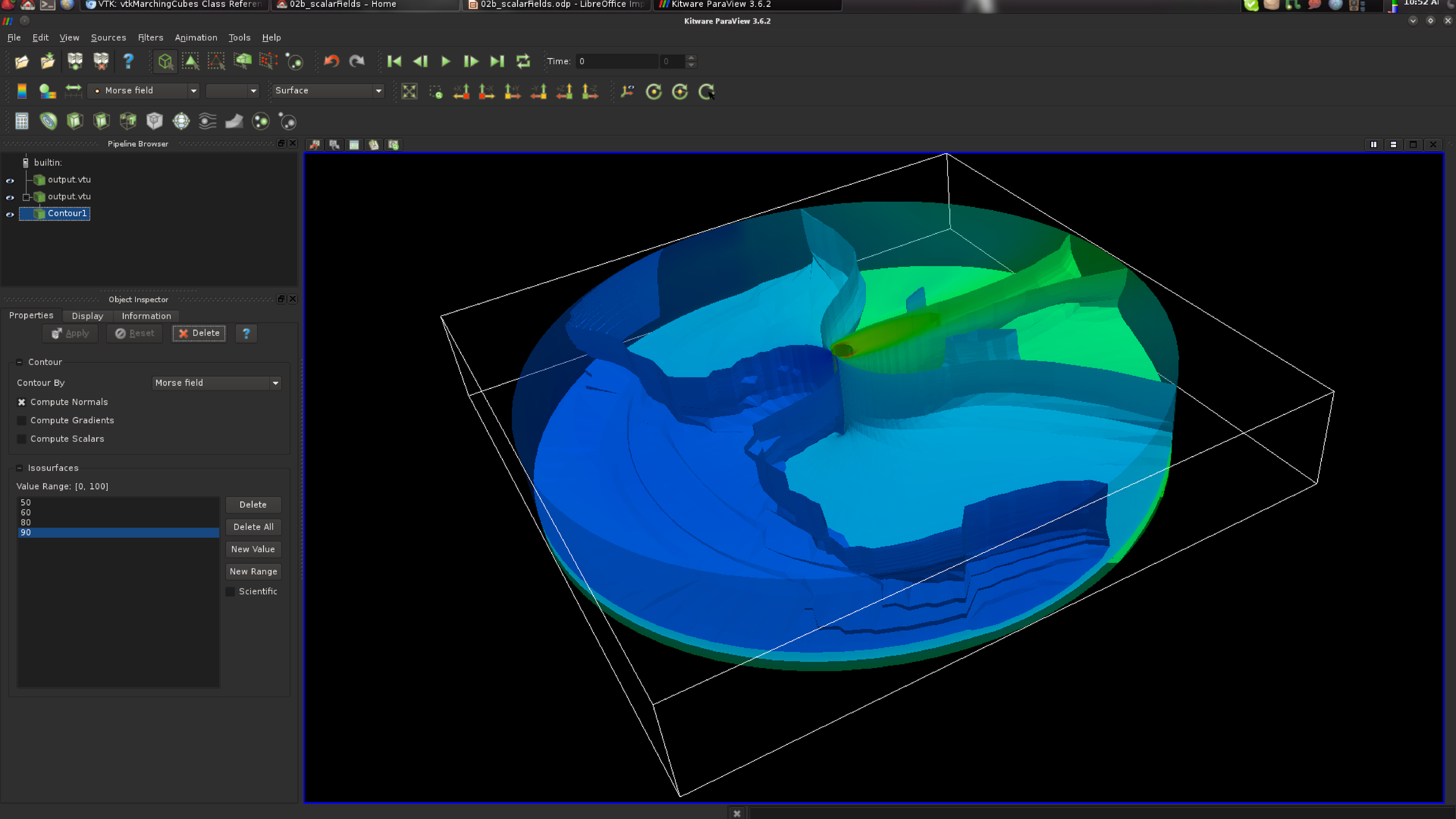

VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Object Inspector
Properties 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

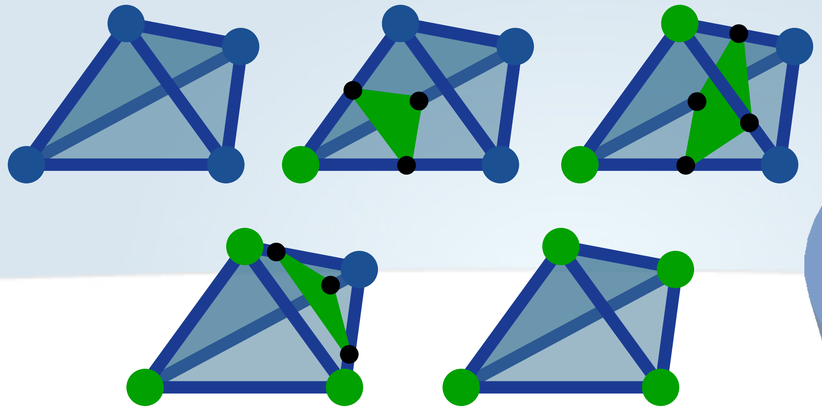

VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties 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

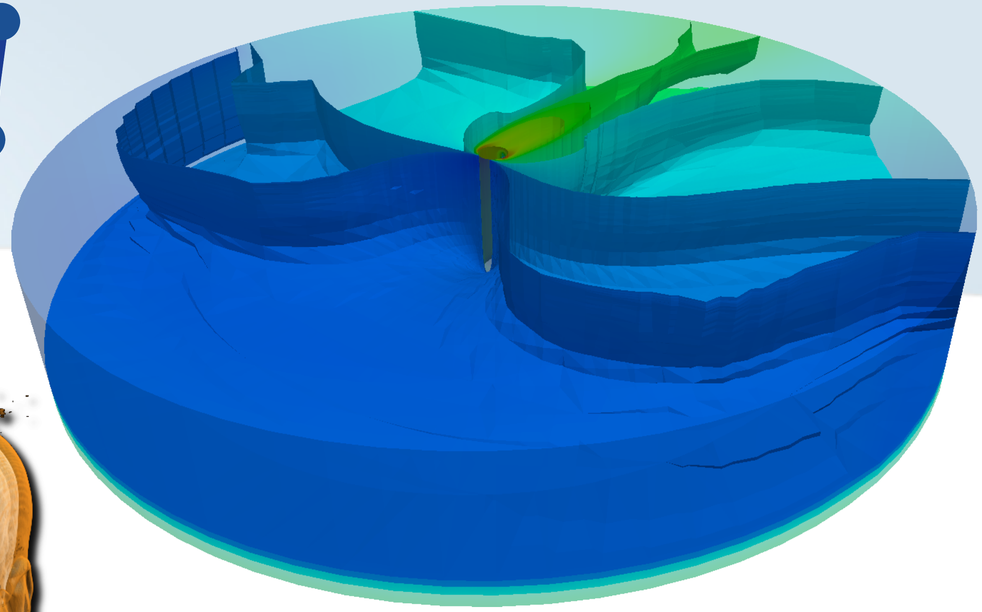

VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties 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

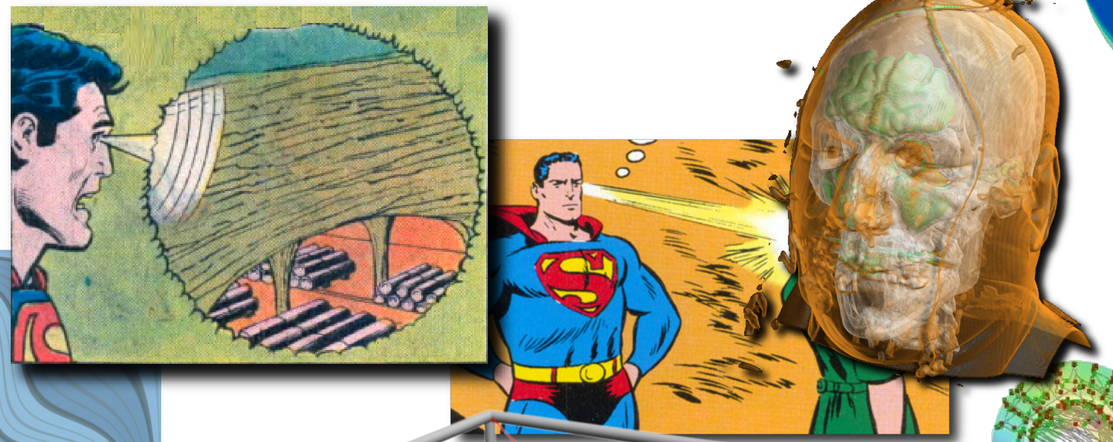

VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties 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

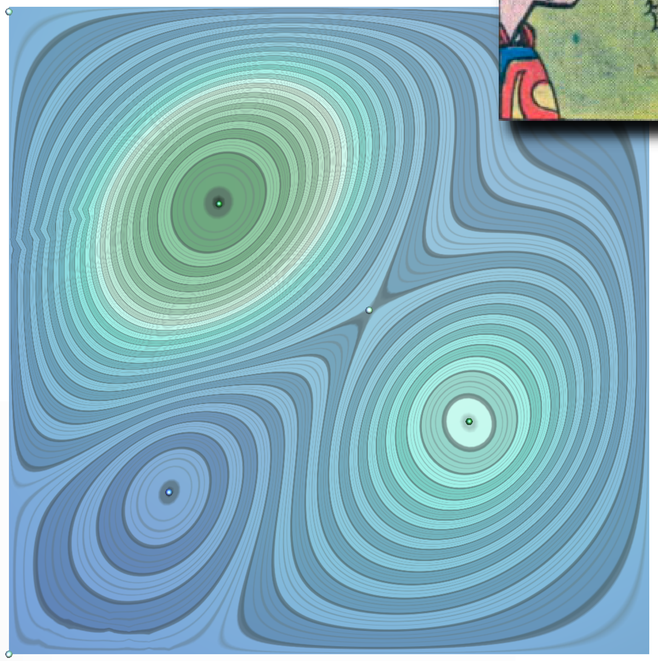

VTK: vtkMarchingCubes Class Referen
02b_scalarFields - Home
02b_scalarFields.odp - LibreOffice Imp
Kitware ParaView 3.6.2
Kitware ParaView 3.6.2
File Edit View Sources Filters Animation Tools Help
Time: 0
Morse field
Surface
Pipeline Browser
builtin:
output.vtu
output.vtu
Contour1
Object Inspector
Properties 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

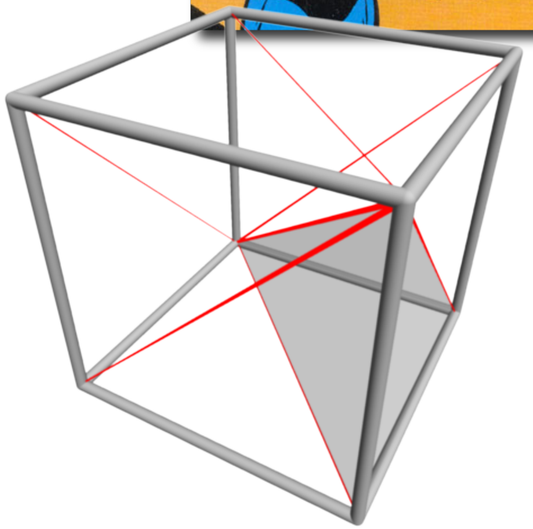
Scalars

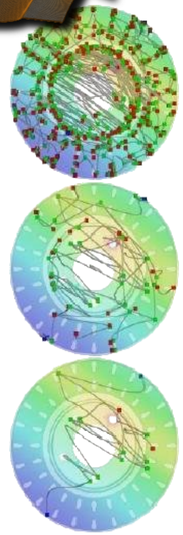


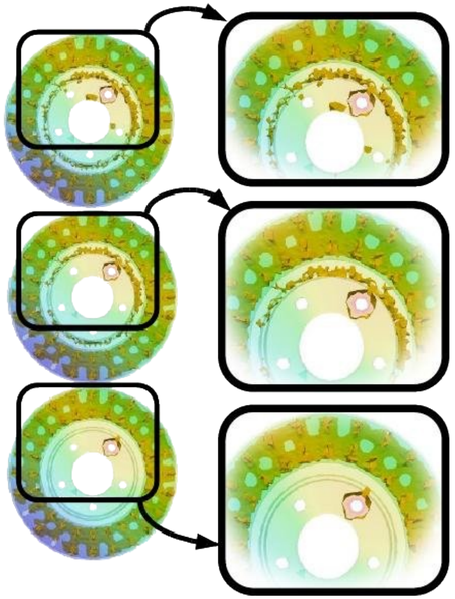


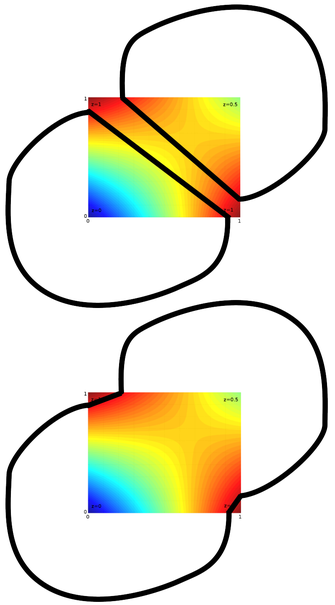


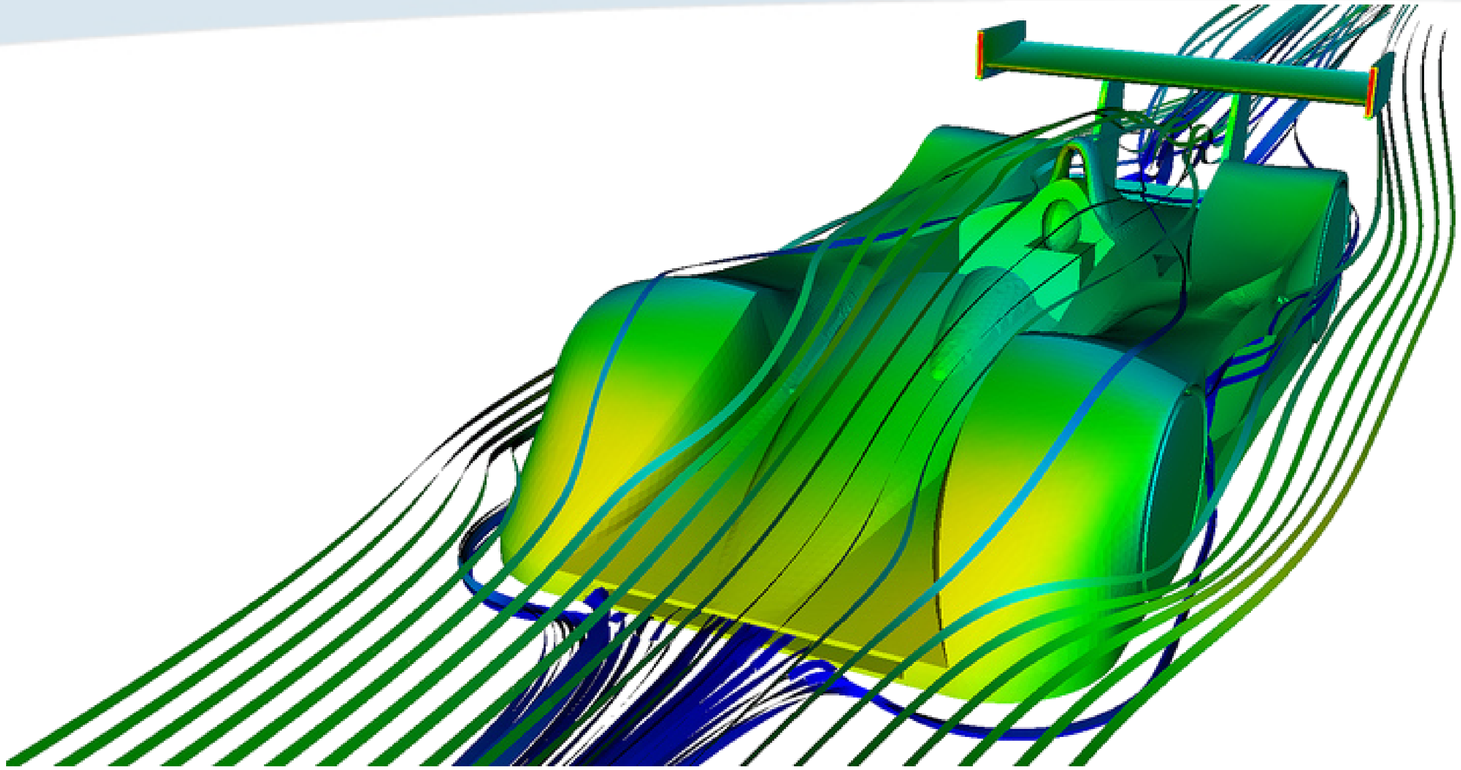












[Kitware]

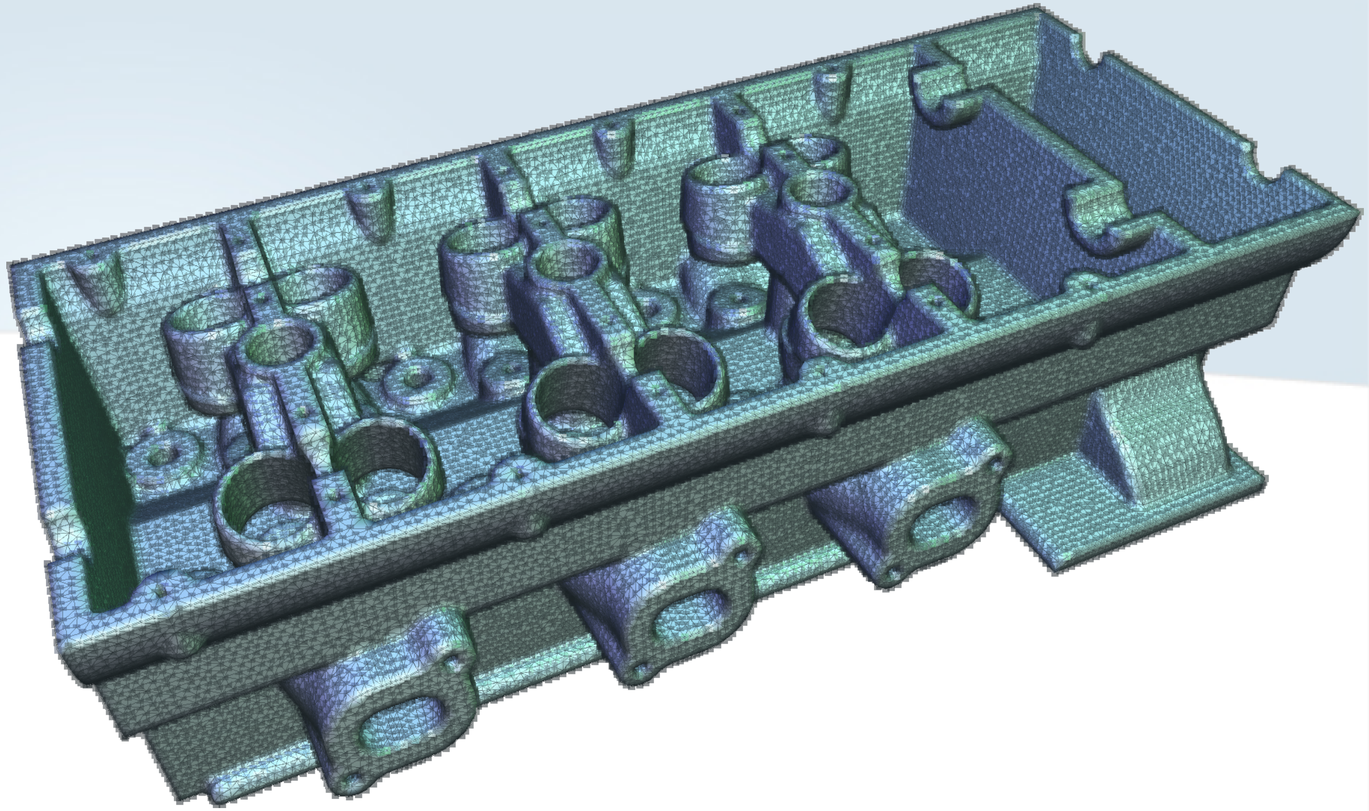
Vector Field Visualization

In practice

- Given a domain $\mathcal{D}$

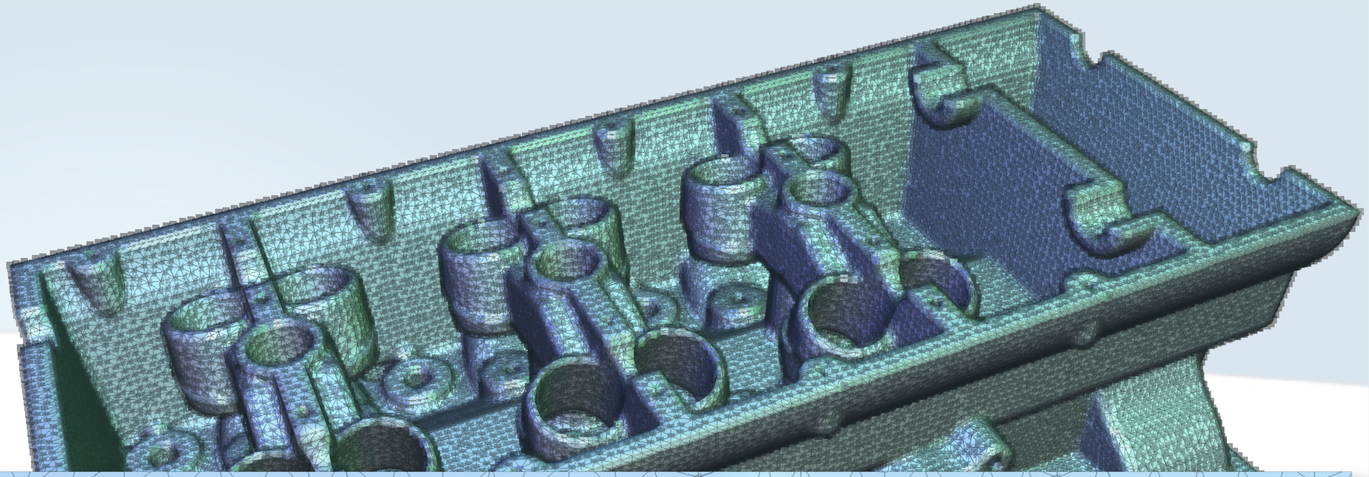
In practice

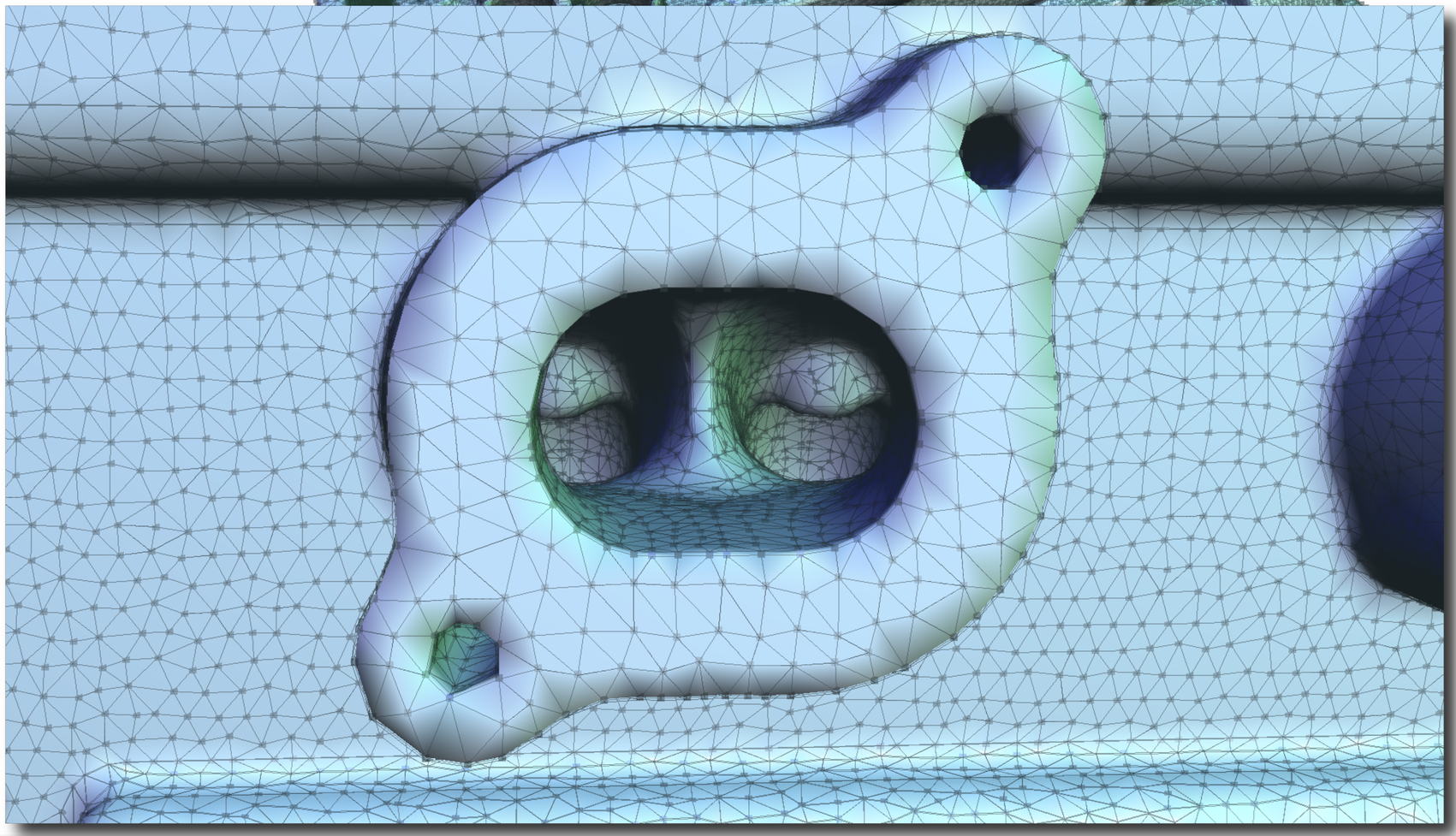
- Given a domain $\mathcal{D}$



In practice

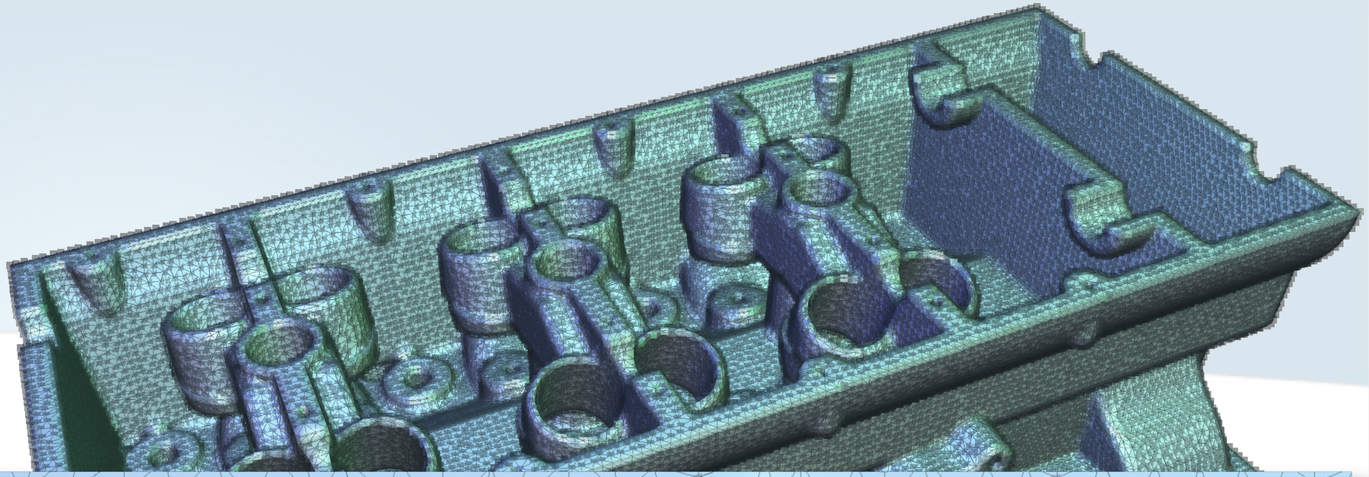
- Given a domain $\mathcal{D}$

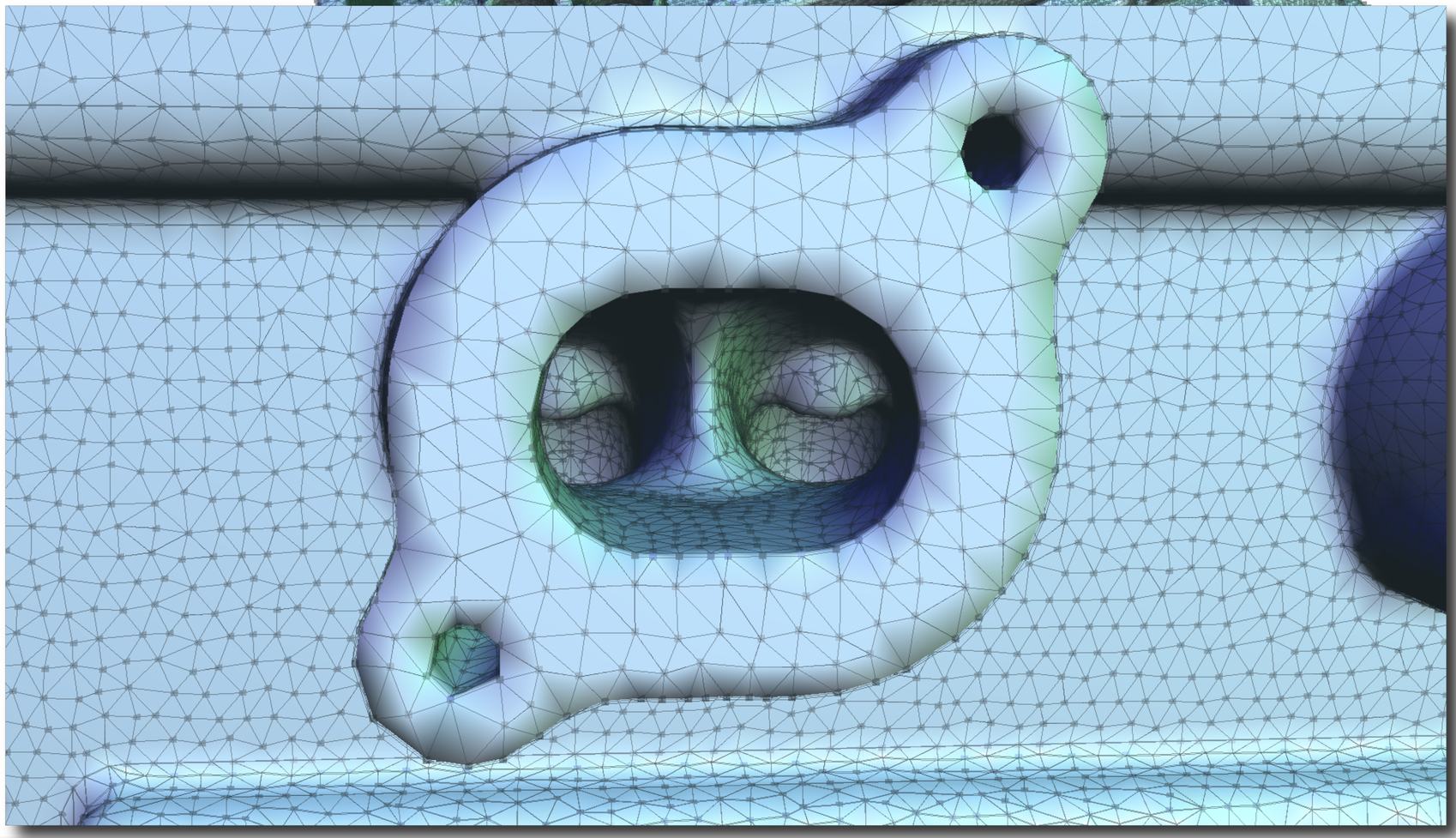




In practice

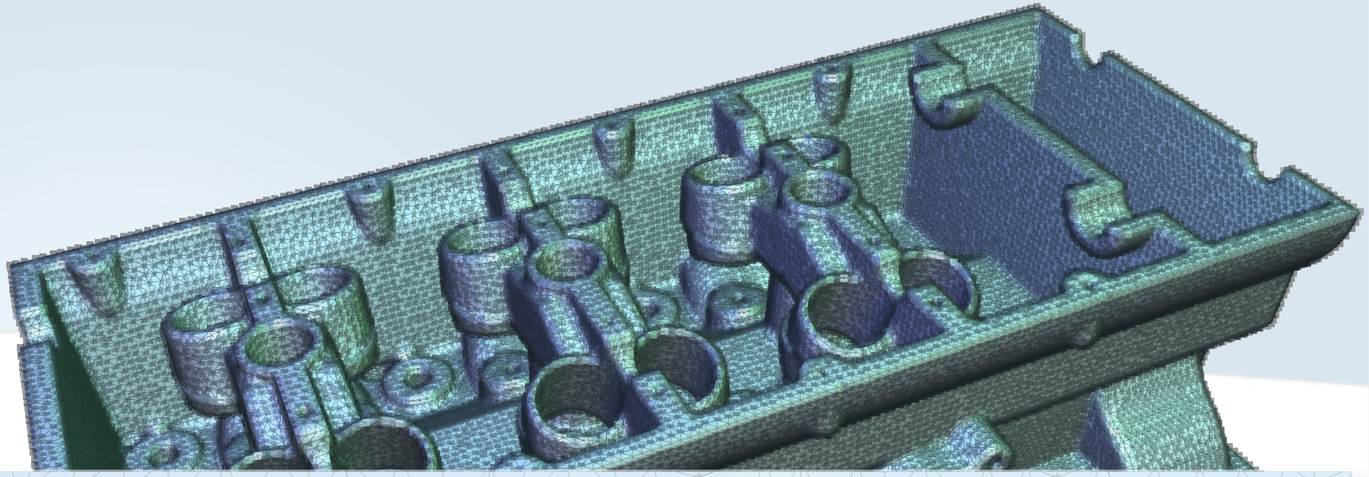
- Given a domain $\mathcal{D}$
- For each vertex v

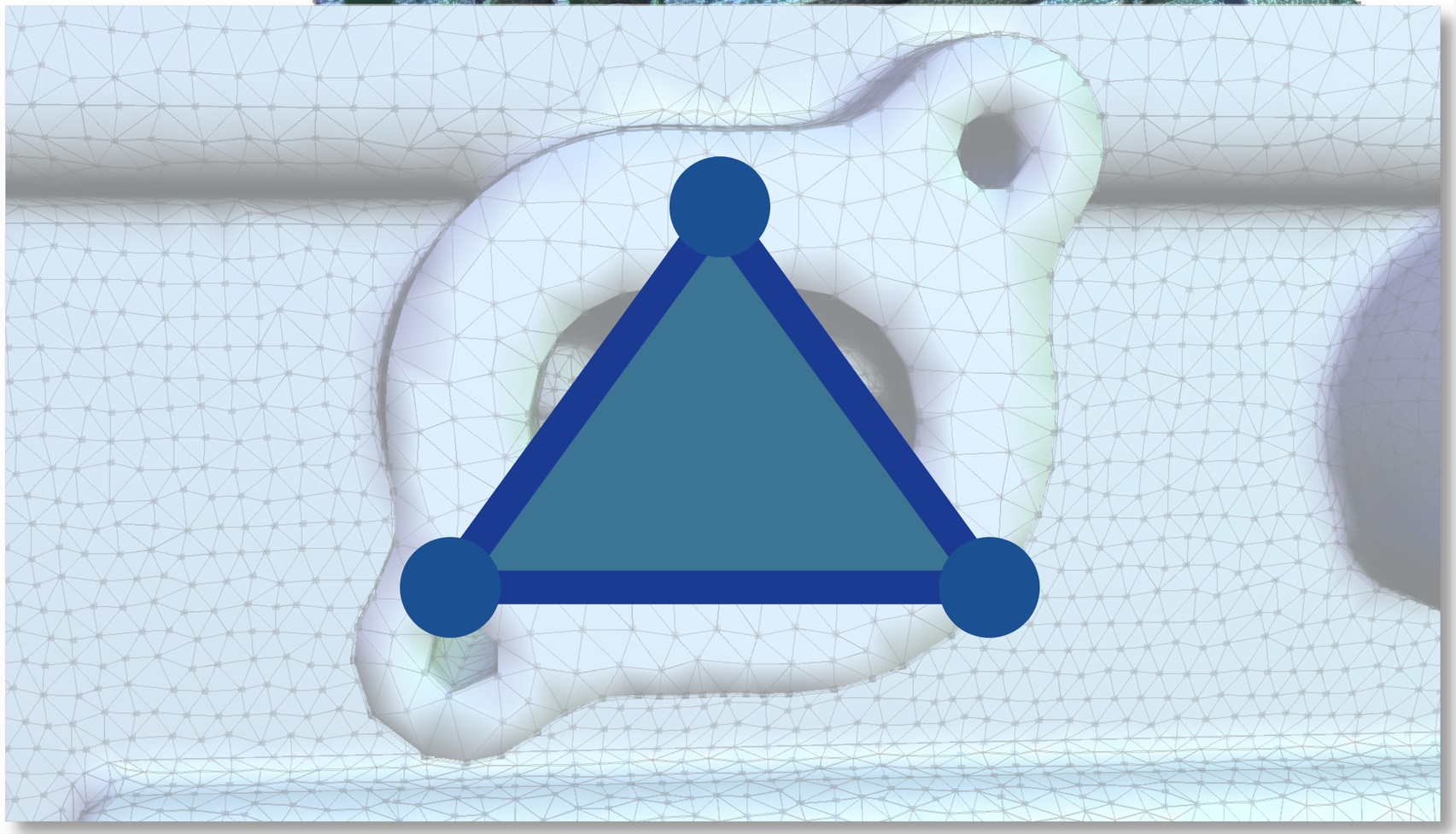




In practice

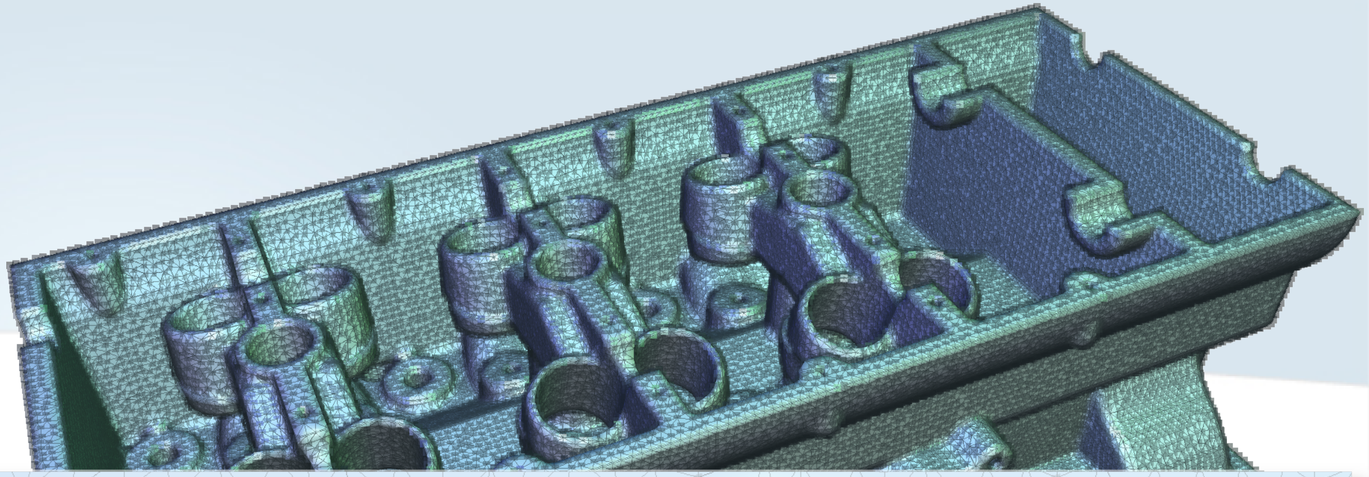
- Given a domain $\mathcal{D}$
- For each vertex v

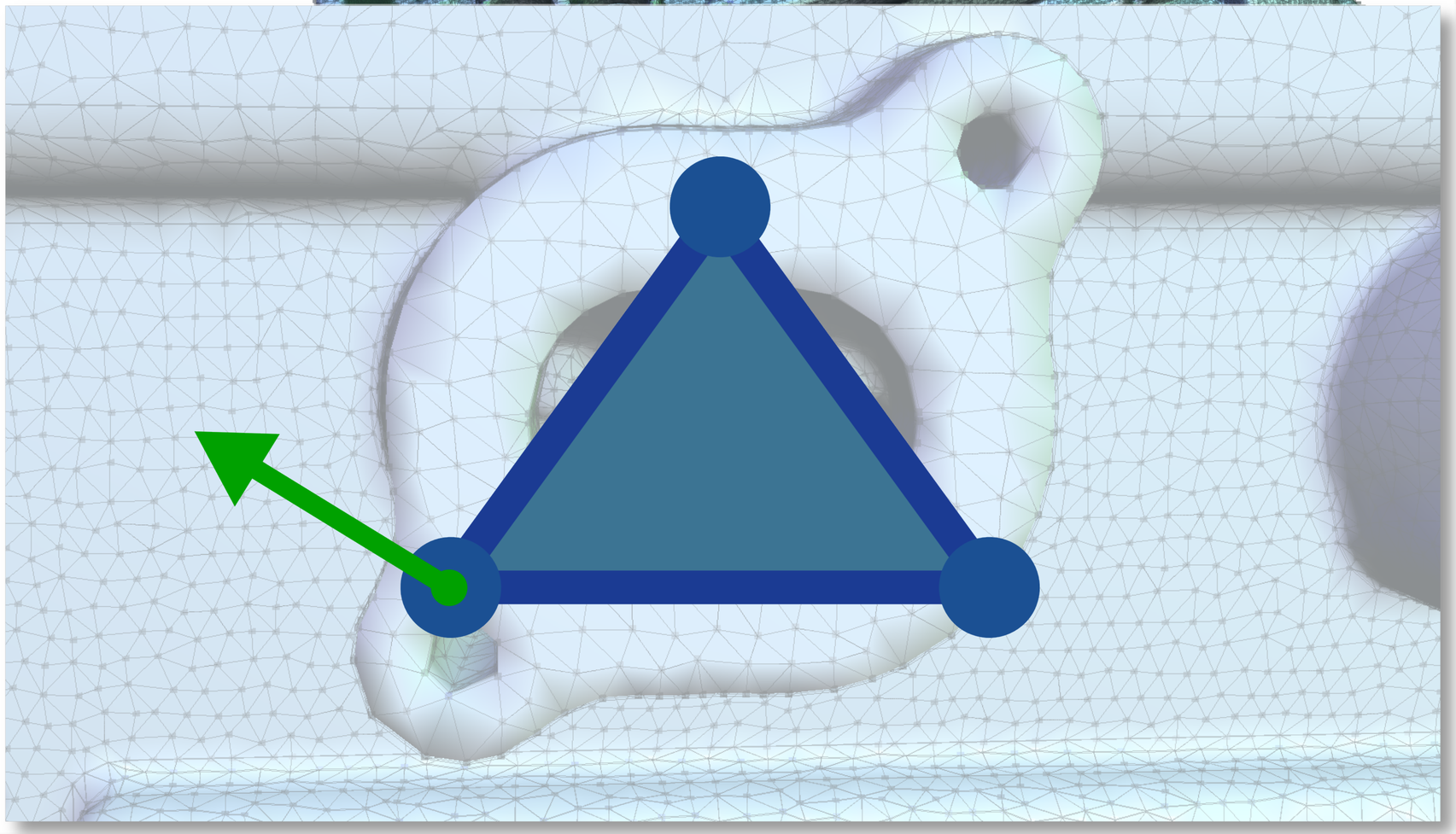




In practice

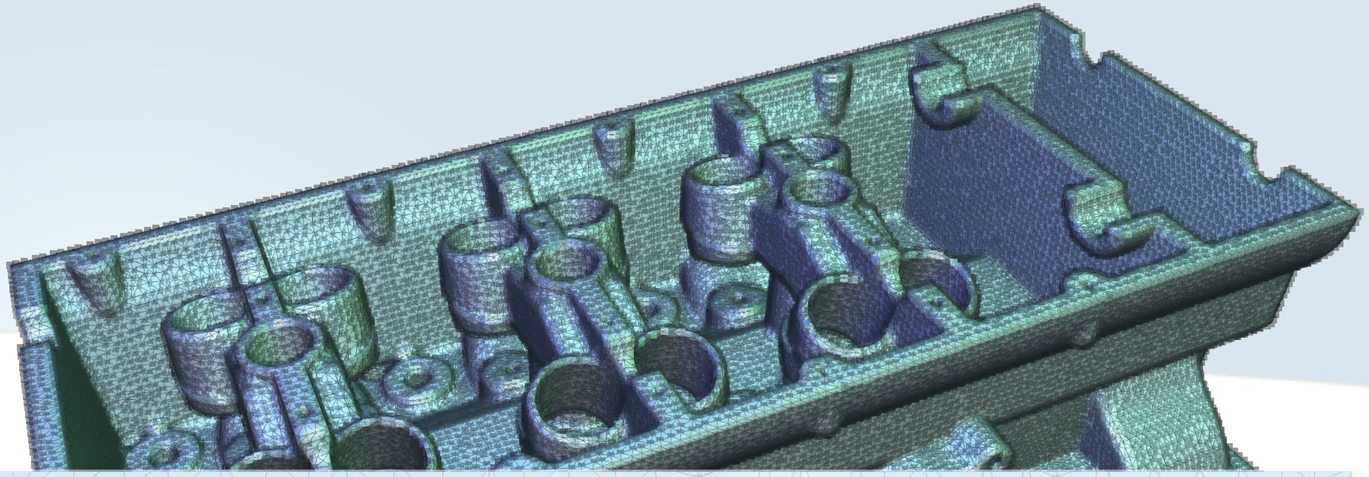
- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$

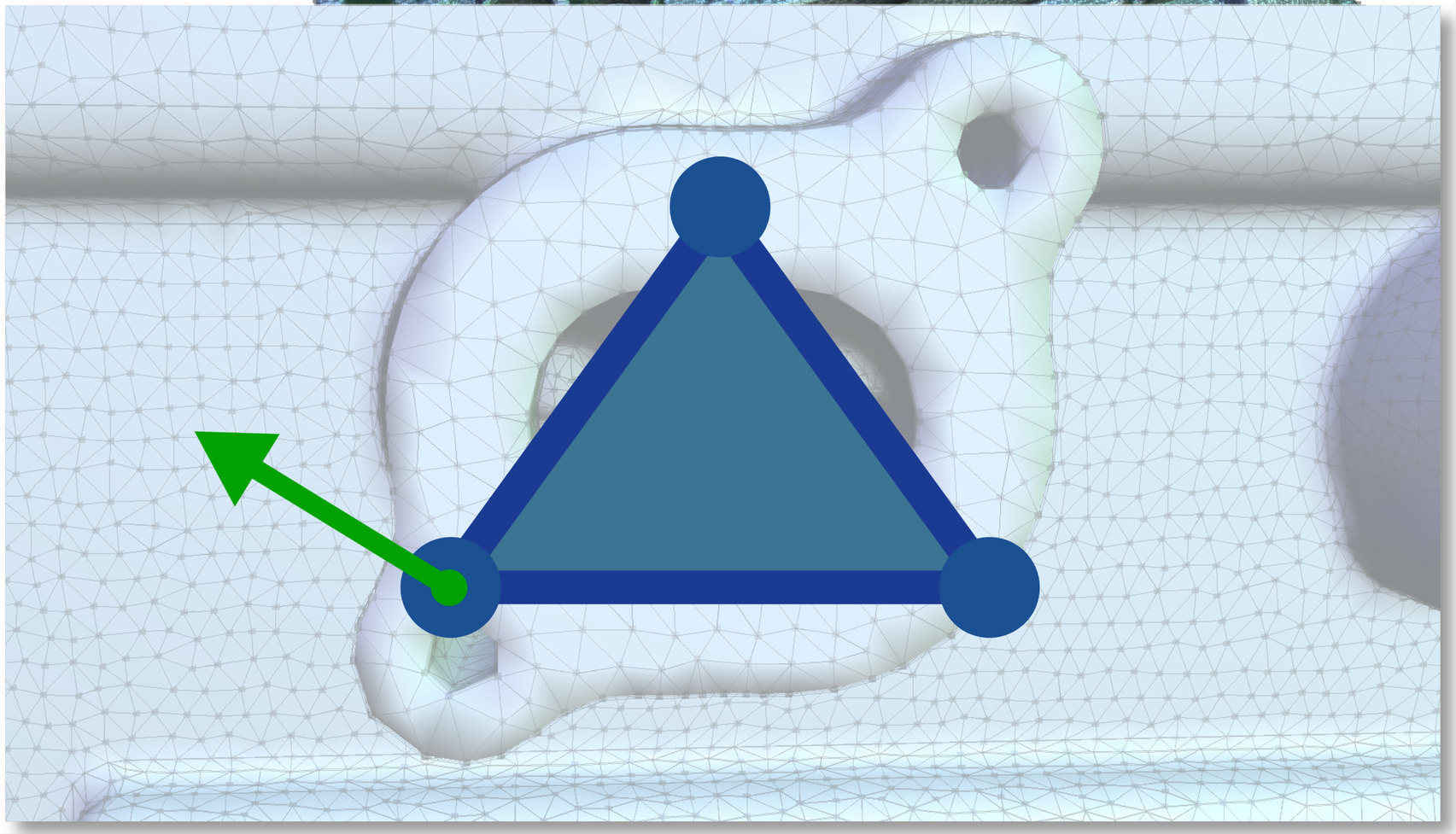




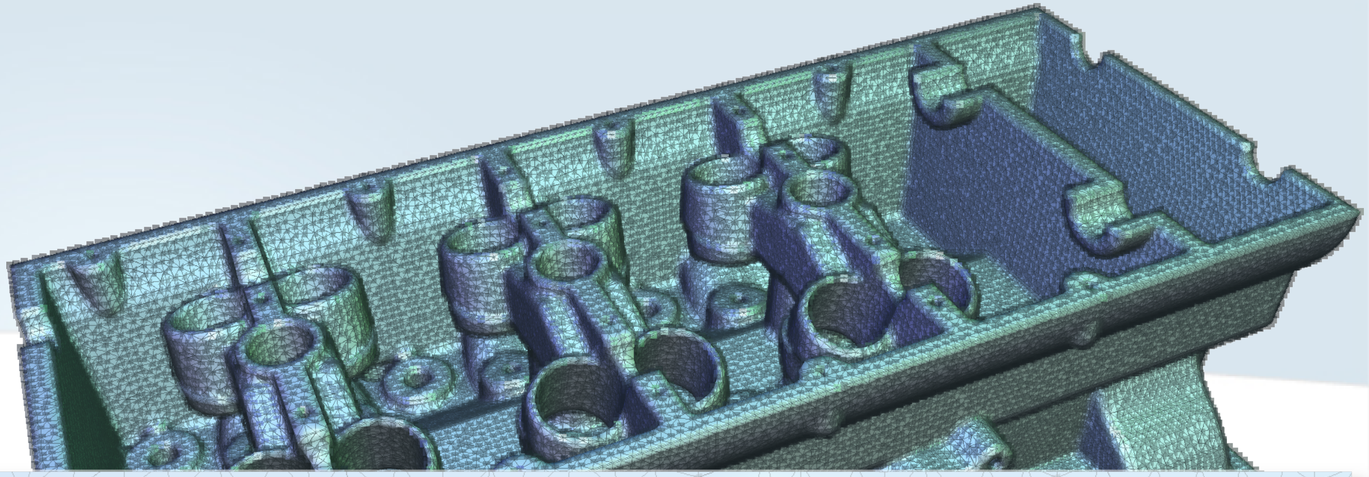
In practice

- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$
  - *Coordinates in* $\mathbb{E}$

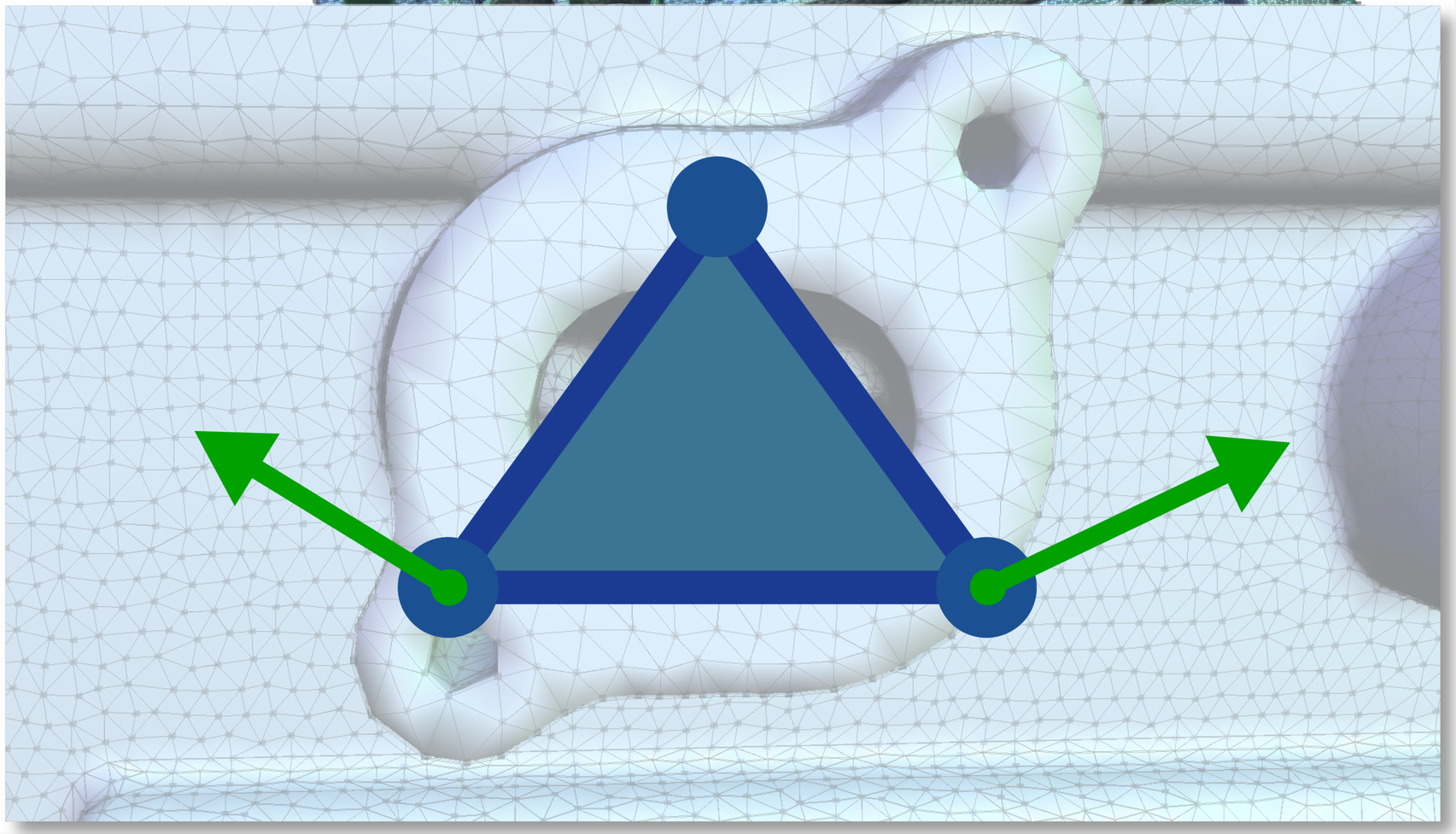




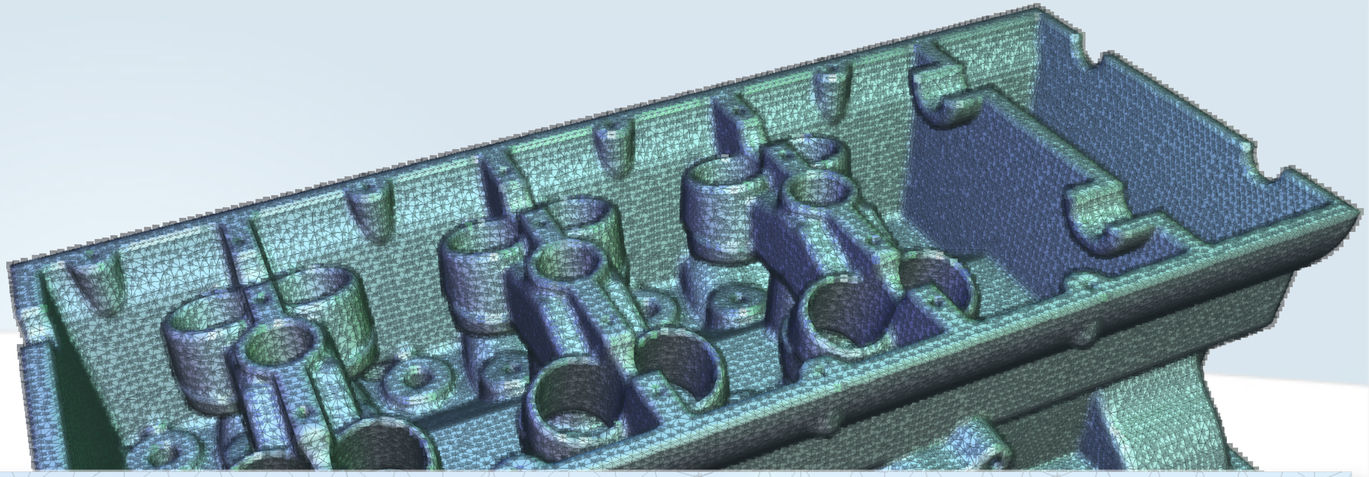
In practice



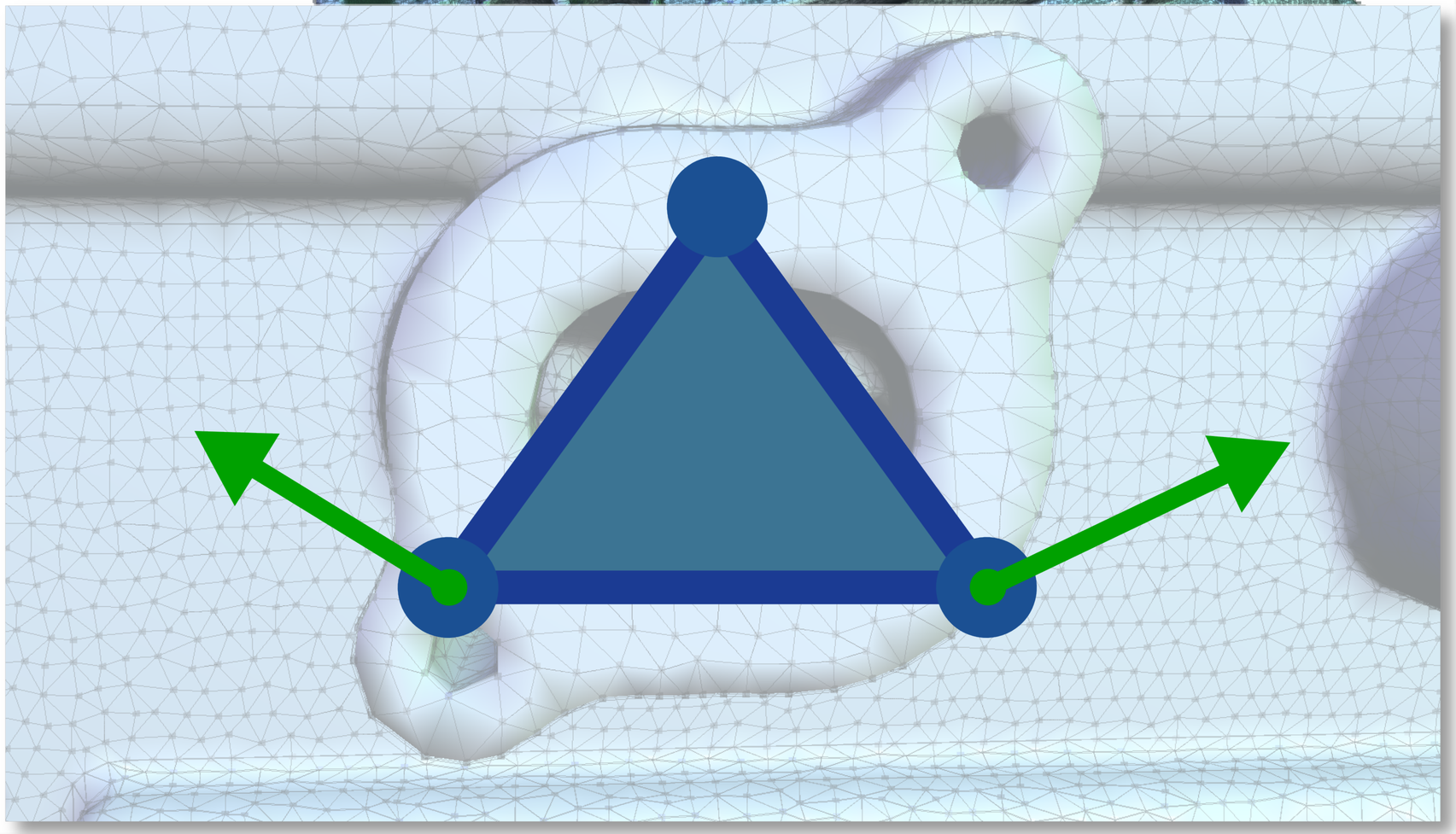
- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$
  - Coordinates in $\mathbb{E}$
- Interpolation on the other simplices:



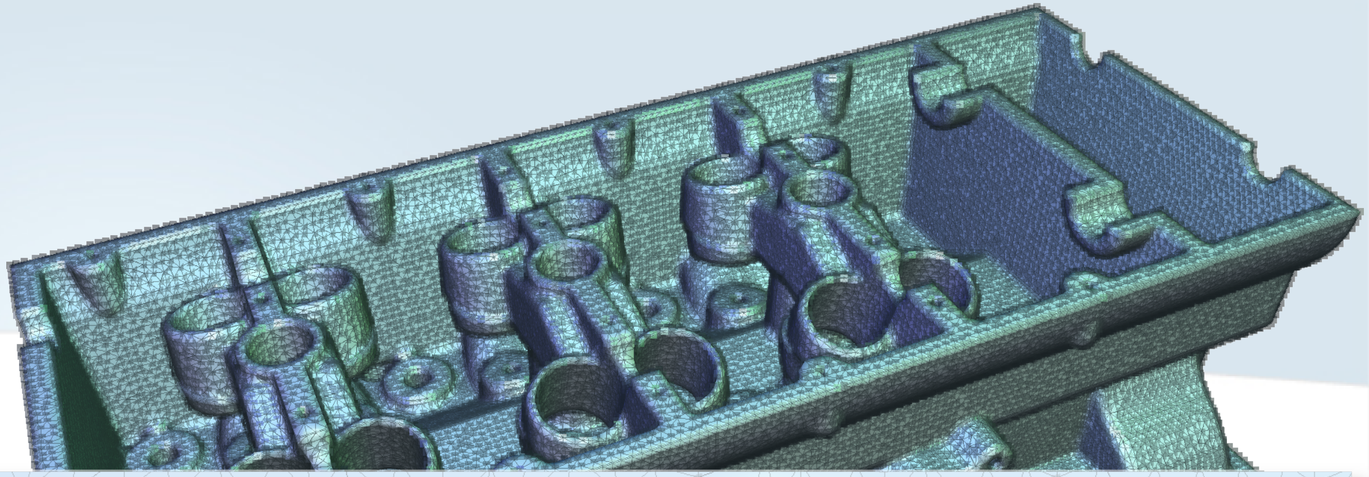
In practice



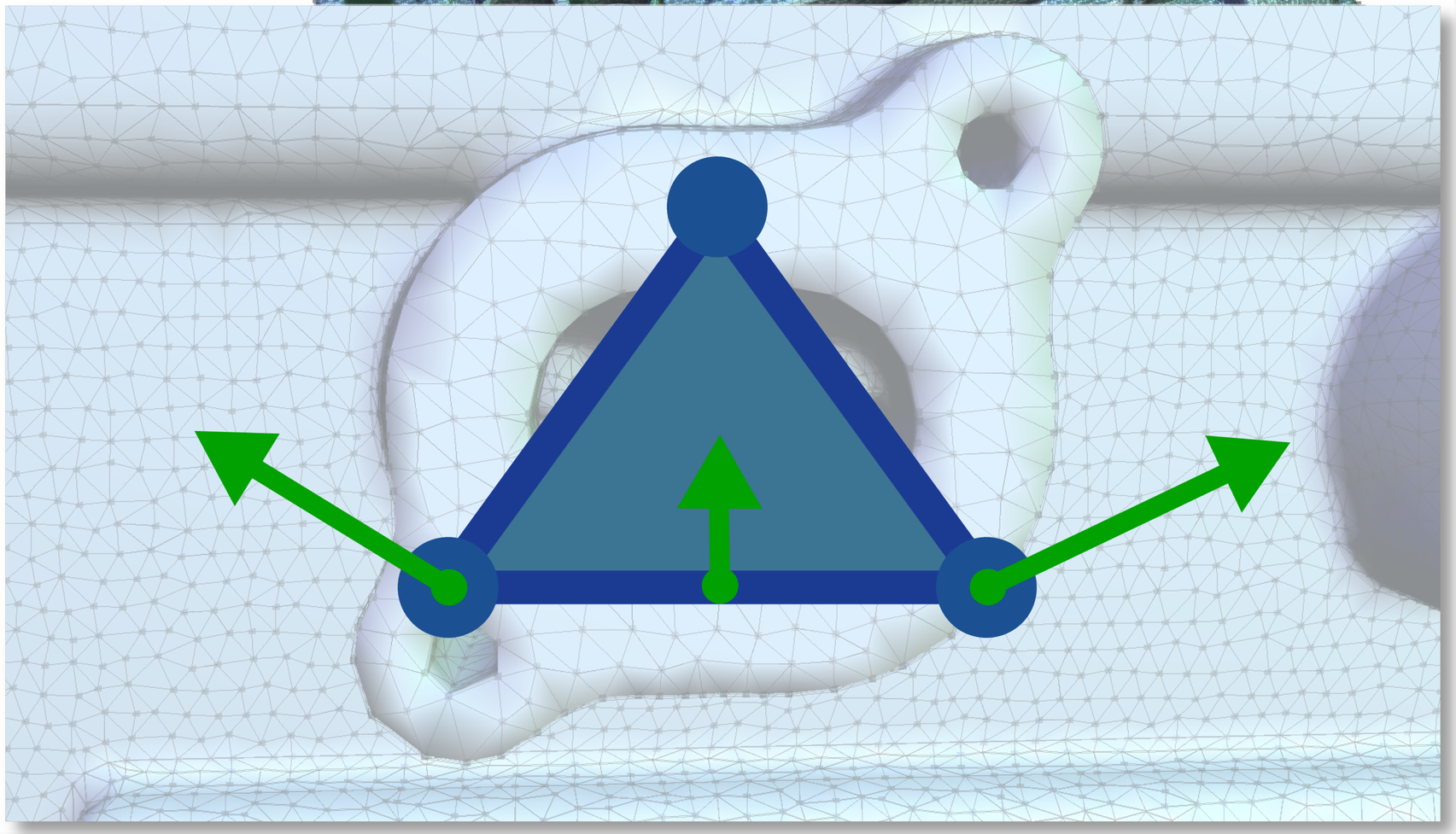
- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$
  - Coordinates in $\mathbb{E}$
- Interpolation on the other simplices:
  - Coordinates



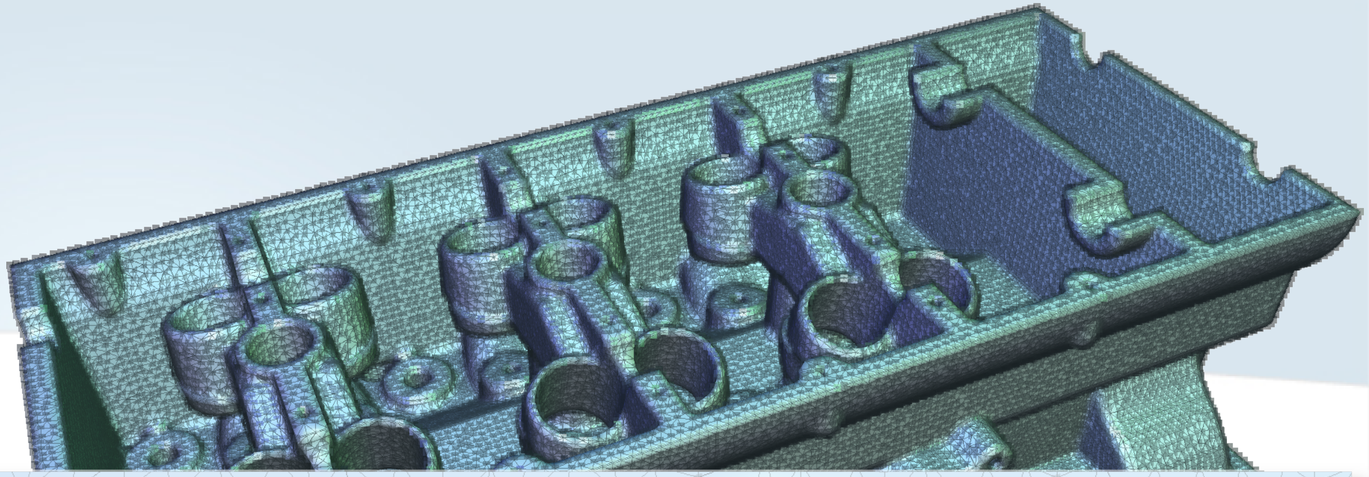
In practice



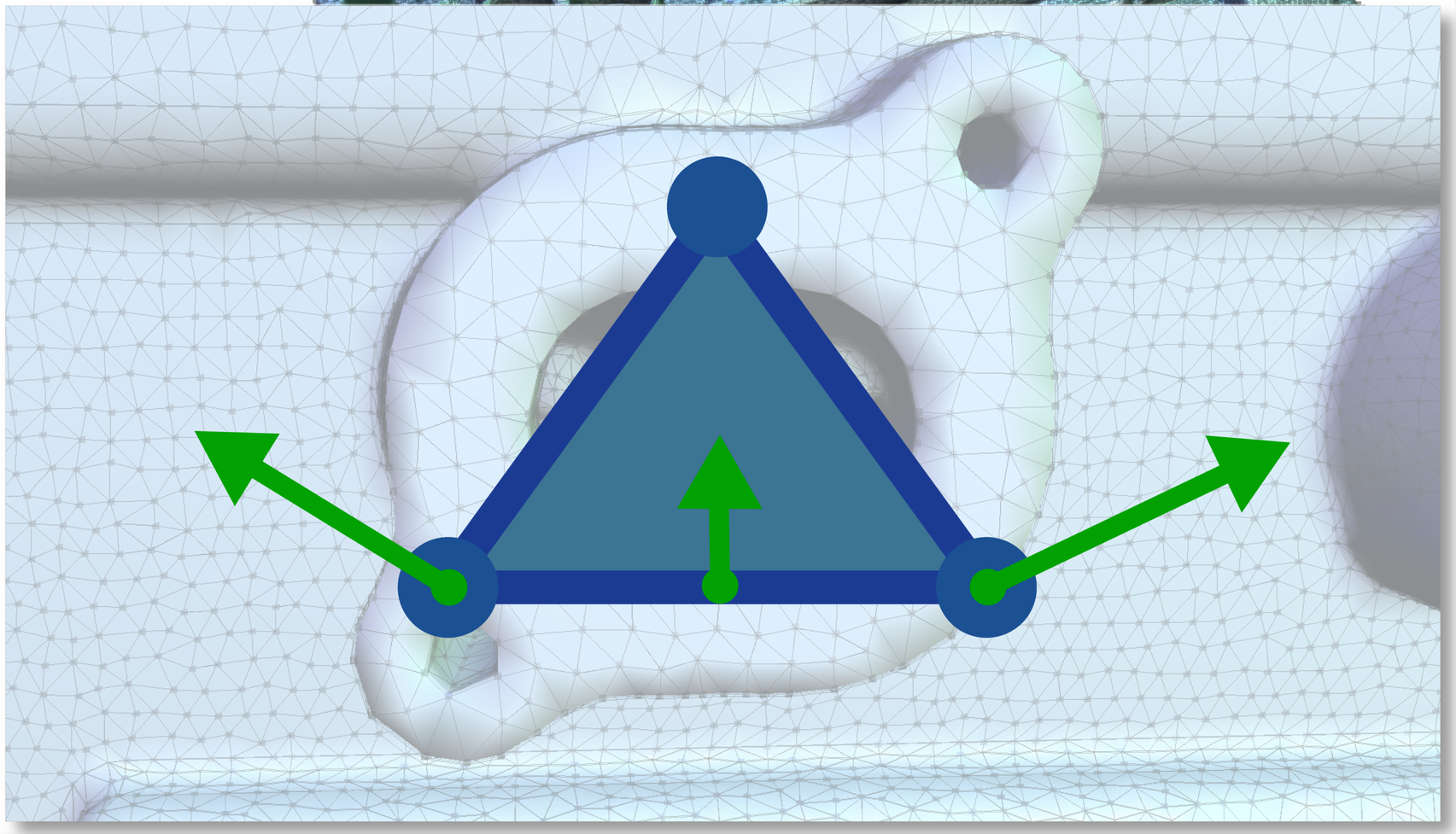
- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$
  - Coordinates in $\mathbb{E}$
- Interpolation on the other simplices:
  - Coordinates



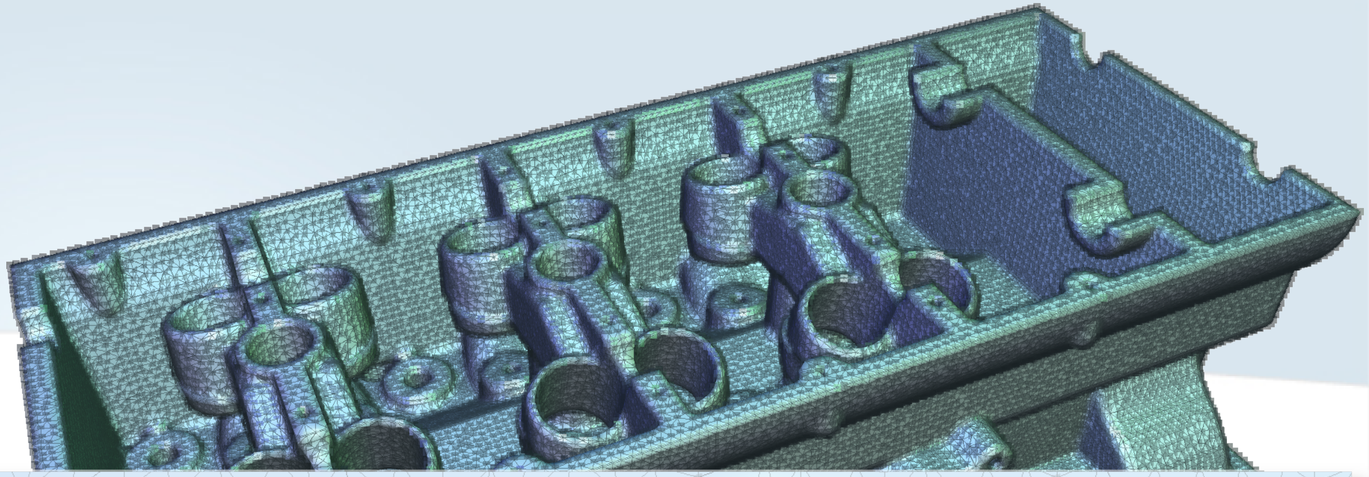
In practice



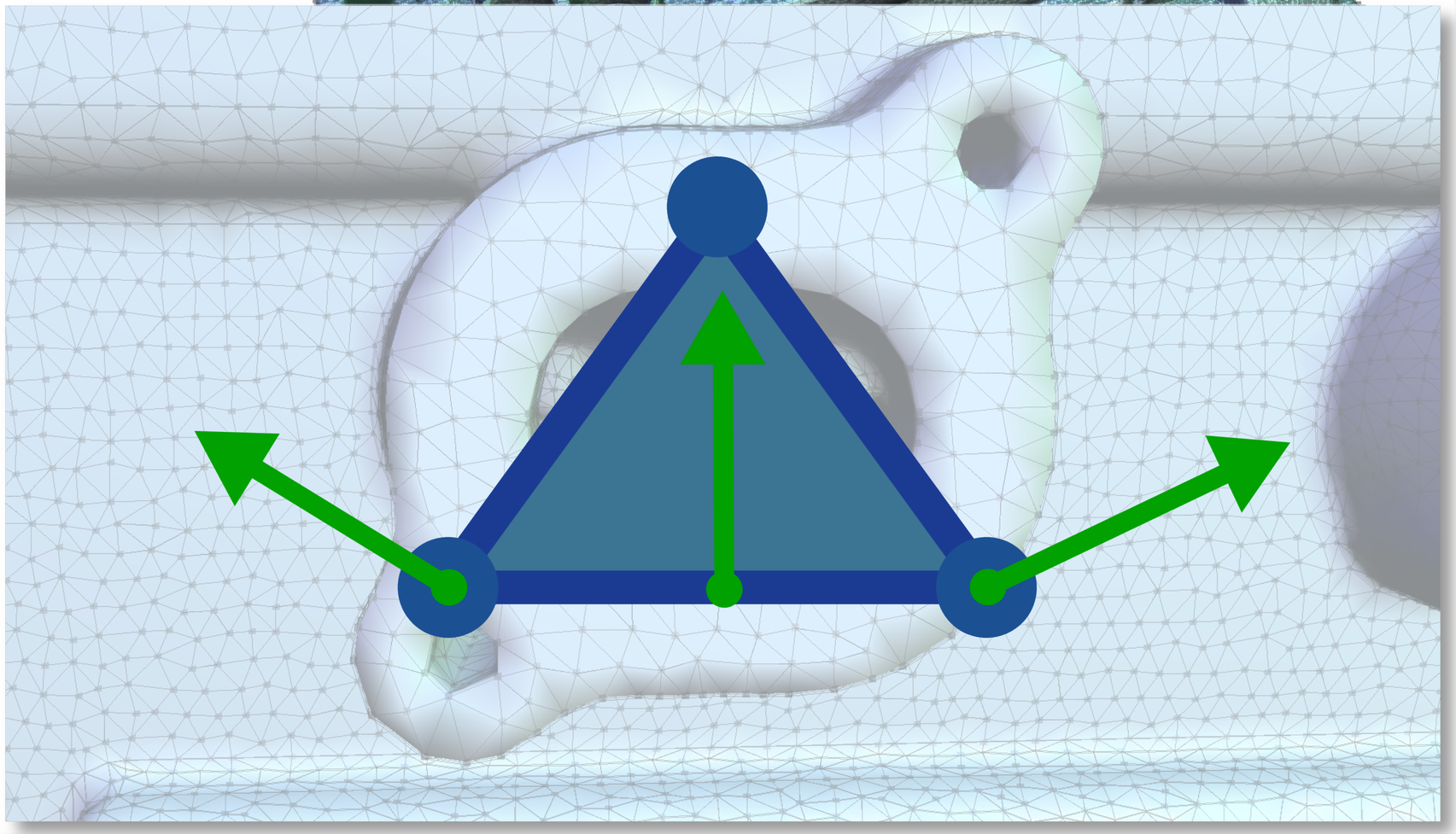
- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$
  - Coordinates in $\mathbb{E}$
- Interpolation on the other simplices:
  - Coordinates
  - Magnitude/angle



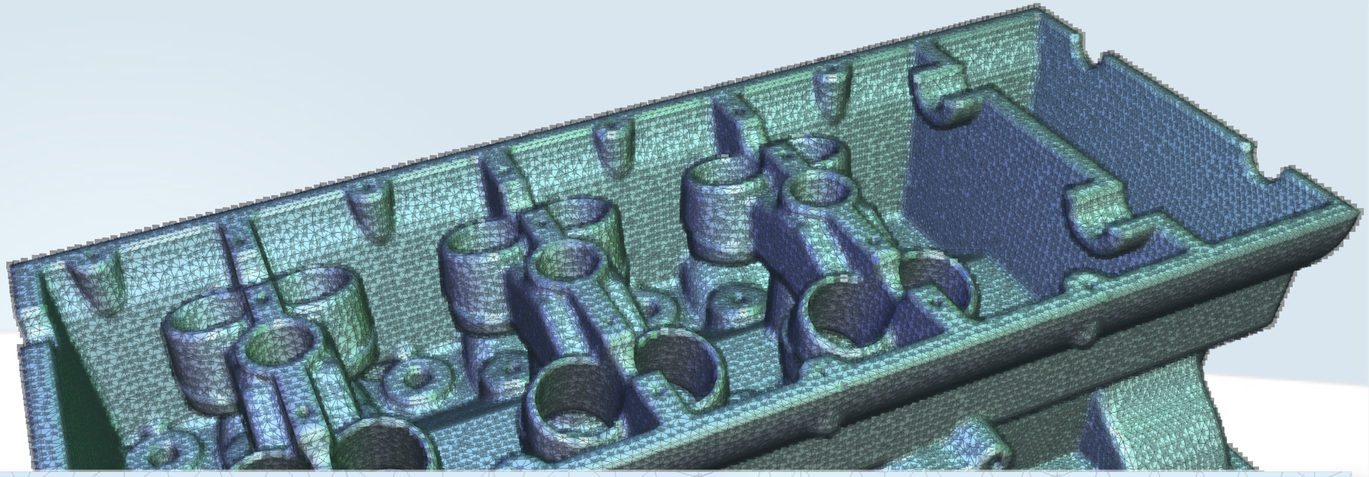
In practice



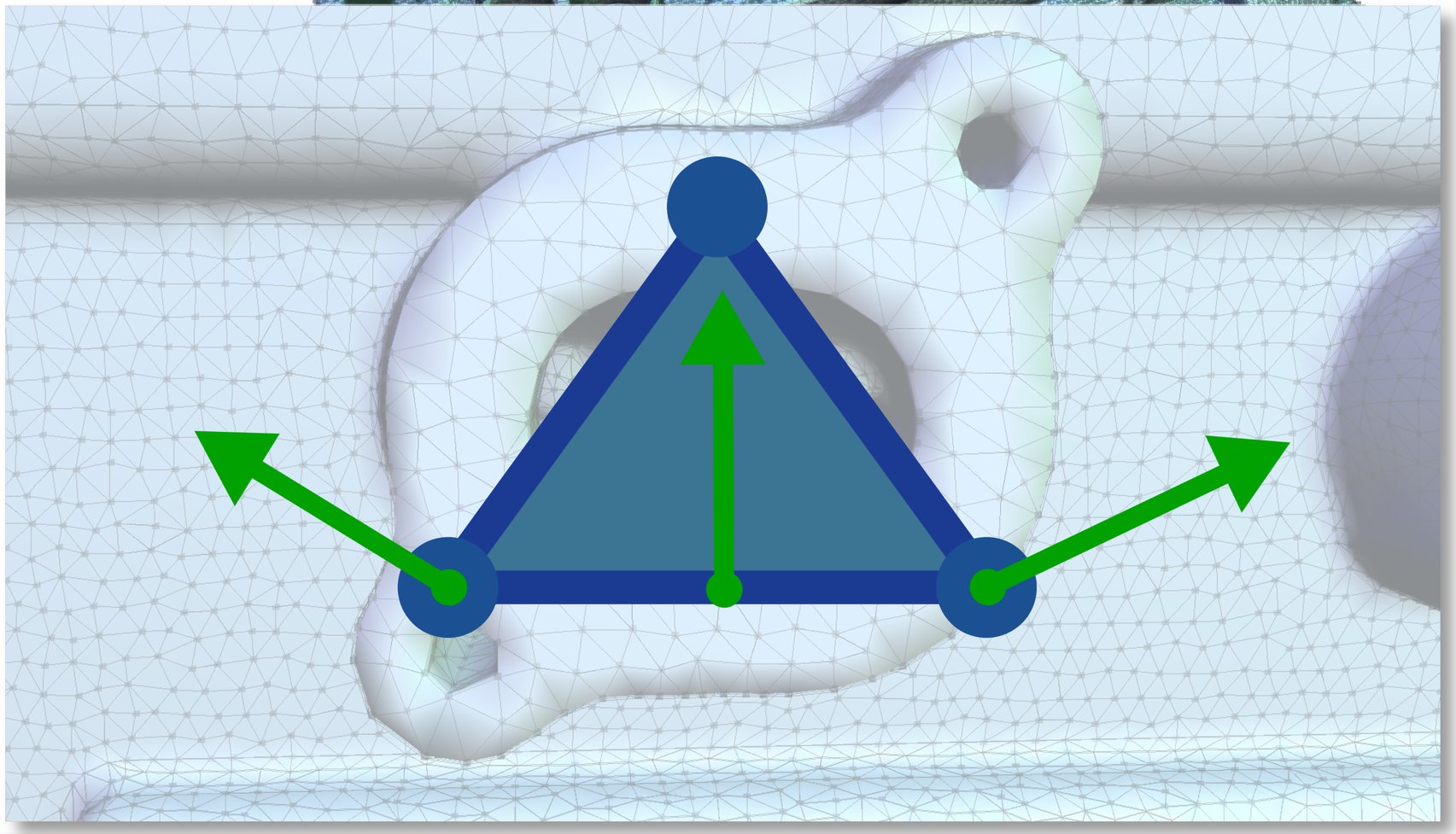
- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$
  - Coordinates in $\mathbb{E}$
- Interpolation on the other simplices:
  - Coordinates
  - Magnitude/angle

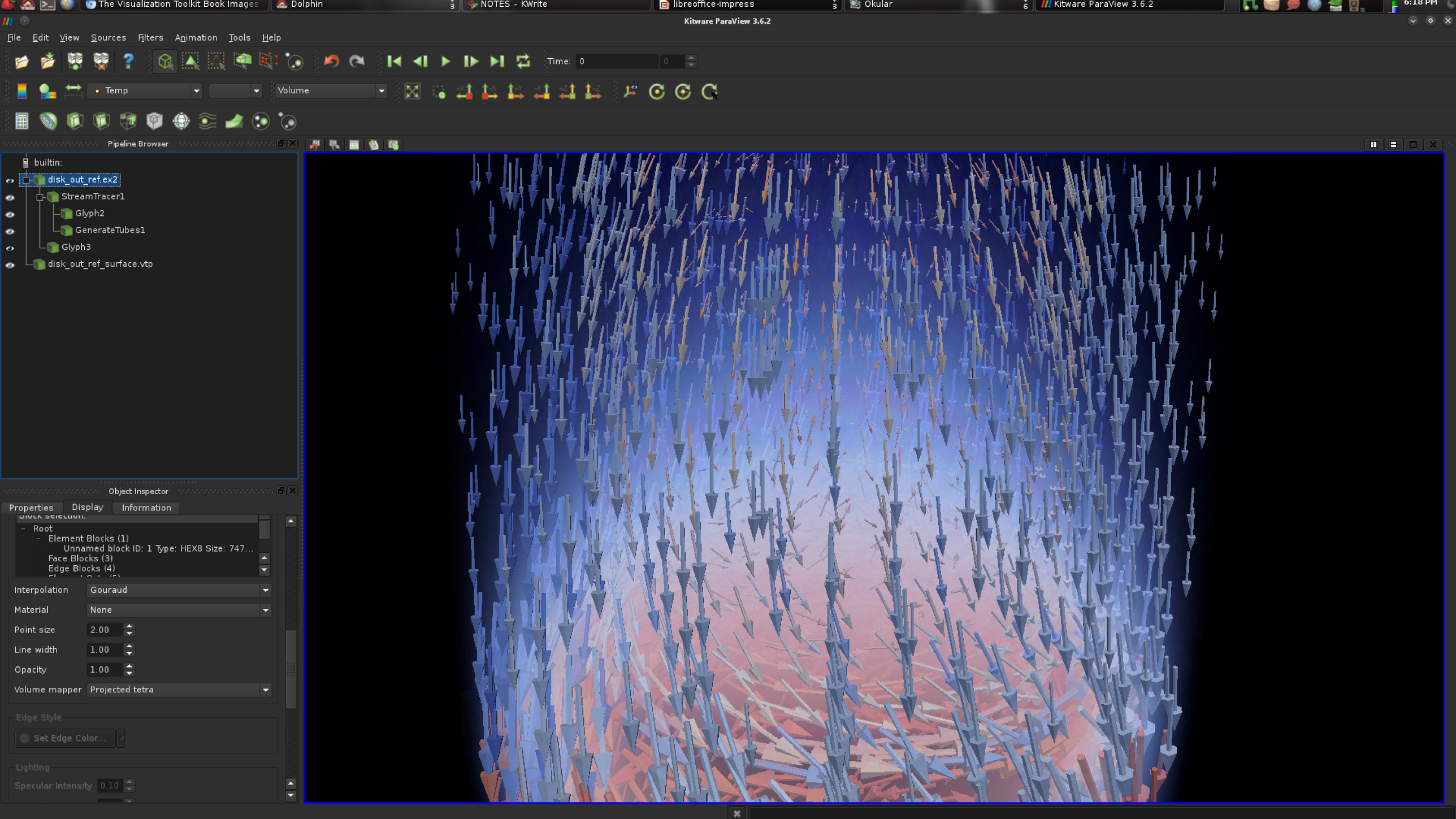


In practice



- Given a domain $\mathcal{D}$
- For each vertex v
- One vector $\vec{f}(v)$
  - Coordinates in $\mathbb{E}$
- Interpolation on the other simplices:
  - Coordinates
  - **Magnitude/angle**




The Visualization Toolkit Book Images
Dolphin
NOTES - KWrite
libreoffice-impress
Okular
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
Glyph2
GenerateTubes1
Glyph3
disk_out_ref_surface.vtp
Object Inspector
Properties Display Information
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
Line width 1.00
Opacity 1.00
Volume mapper Projected tetra
Edge Style
Set Edge Color...
Lighting
Specular Intensity 0.10

What is there to visualize?

- Getting inspiration from engineering sciences

What is there to visualize?

- Getting inspiration from engineering sciences

[www.speedhunter.com]

What is there to visualize?

- Getting inspiration from engineering sciences
  - Localized visualization

[www.speedhunter.com]

What is there to visualize?

- Getting inspiration from engineering sciences
  - Localized visualization
  - Explicit representations

[www.speedhunter.com]

What is there to visualize?

- Getting inspiration from engineering sciences
  - Localized visualization
  - Explicit representations
  - *Stream lines and surfaces*

[www.speedhunter.com]

What is there to visualize?

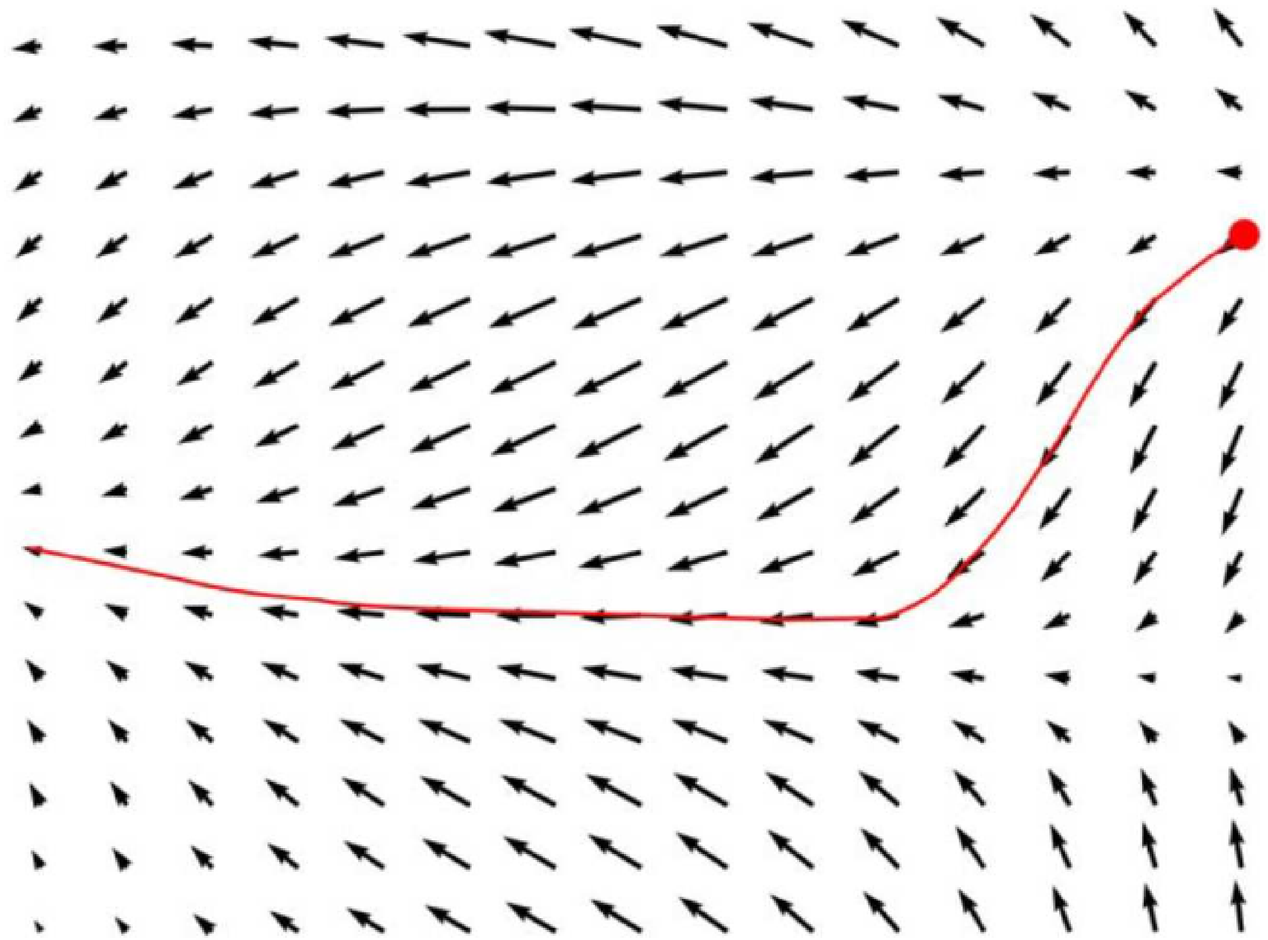
- Getting inspiration from engineering sciences
  - Localized visualization
  - Explicit representations
  - *Stream lines and surfaces*

- Analogy to scalar fields:
  - Isocontours
  - Isosurfaces

[www.speedhunter.com]

Streamlines

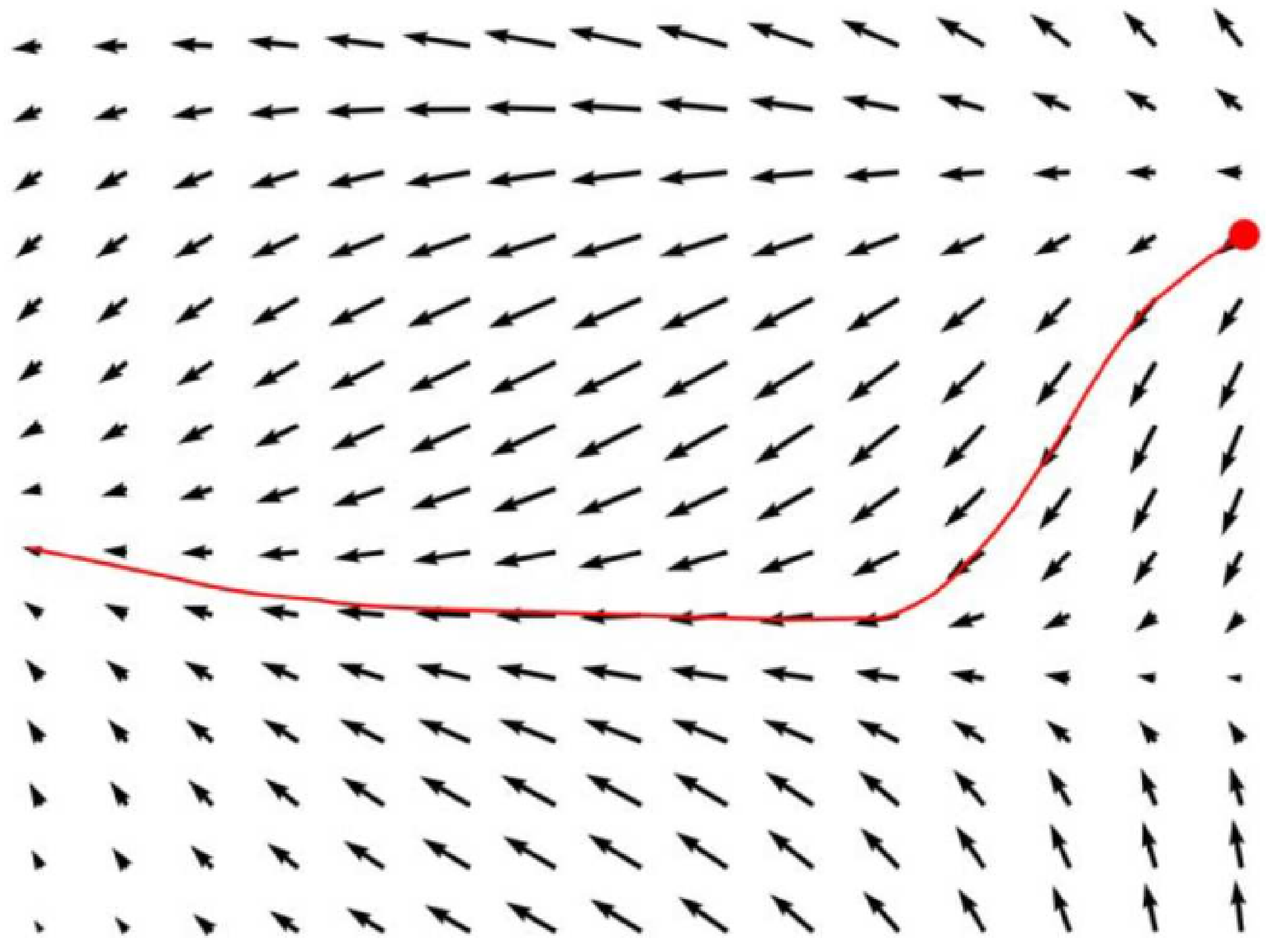
- What is there to visualize?
  - Integral curves
  - “Streamlines”



[Chen]

Streamlines

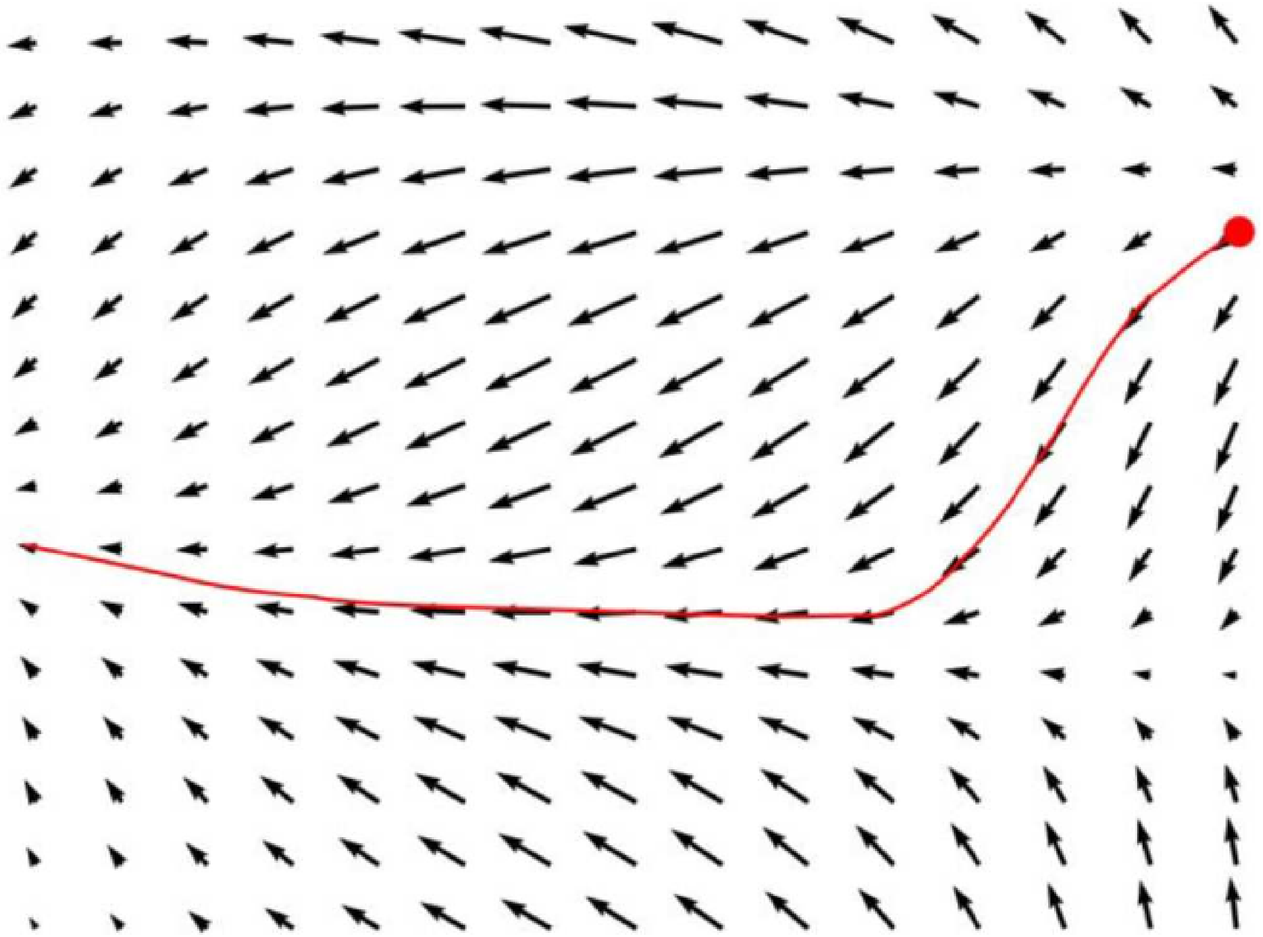
- What is there to visualize?
  - Integral curves
  - “Streamlines”
- Curve $\mathcal{C} \in \mathcal{D}$ such that:



[Chen]

Streamlines

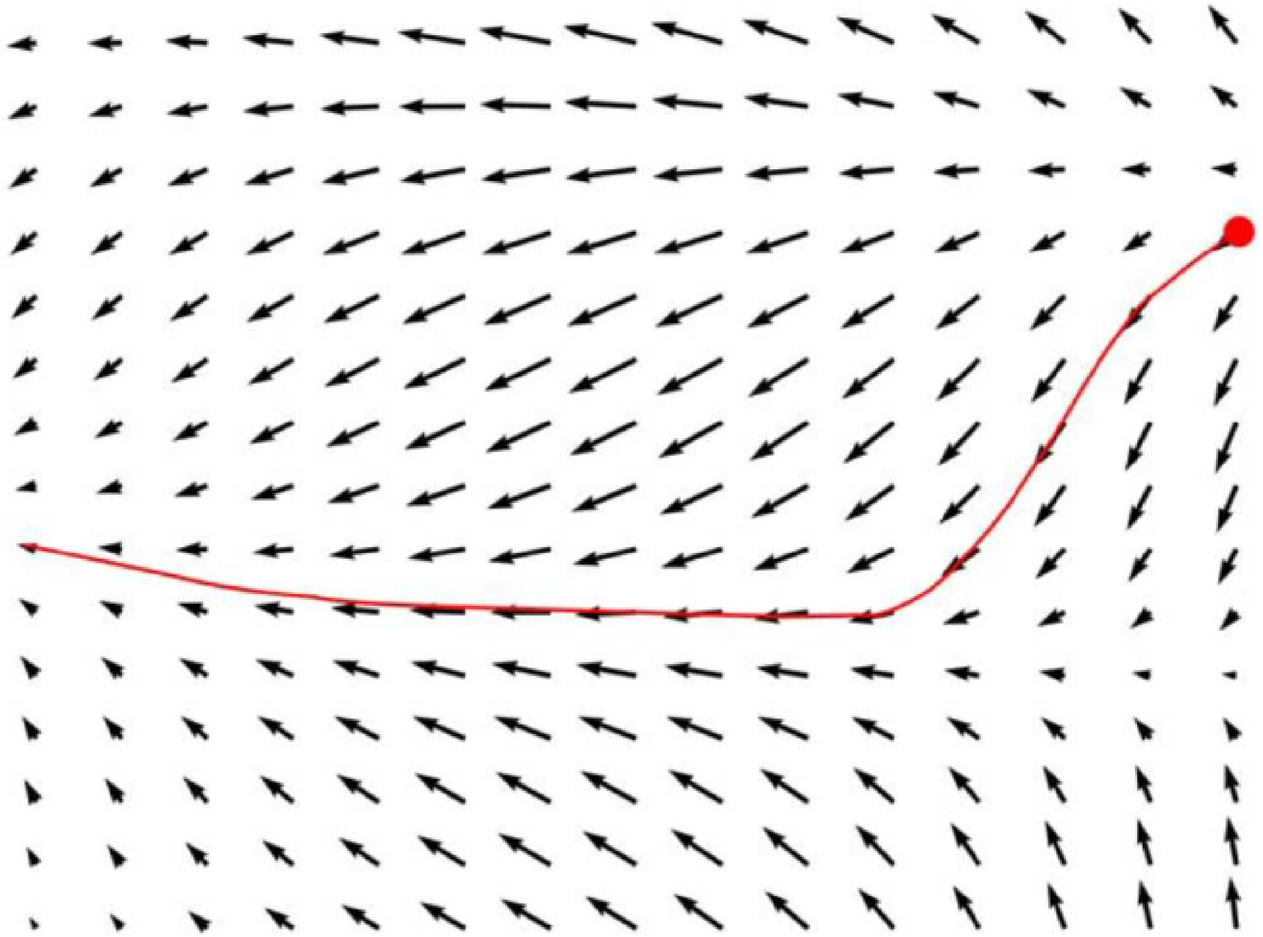
- What is there to visualize?
  - Integral curves
  - “Streamlines”
- Curve $\mathcal{C} \in \mathcal{D}$ such that:
  - Given a bijection
    - $c : \mathcal{C} \rightarrow [0, 1]$



[Chen]

Streamlines

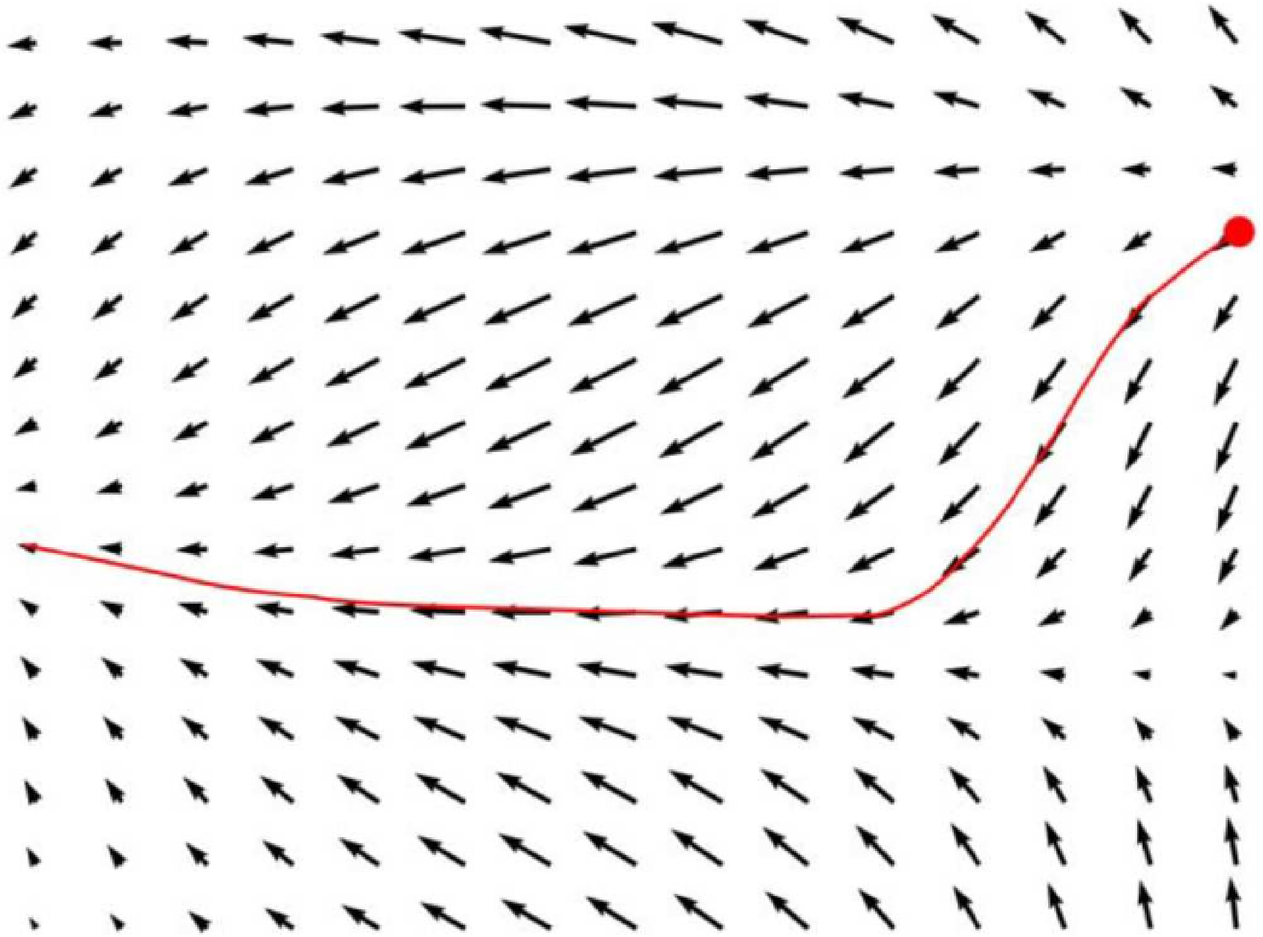
- What is there to visualize?
  - Integral curves
  - “Streamlines”
- Curve $\mathcal{C} \in \mathcal{D}$ such that:
  - Given a bijection
    - $c : \mathcal{C} \rightarrow [0, 1]$
  - $\frac{\partial p}{\partial c} \times \vec{f}(p) = \vec{0}, \quad \forall p \in \mathcal{C}$



[Chen]

Streamlines

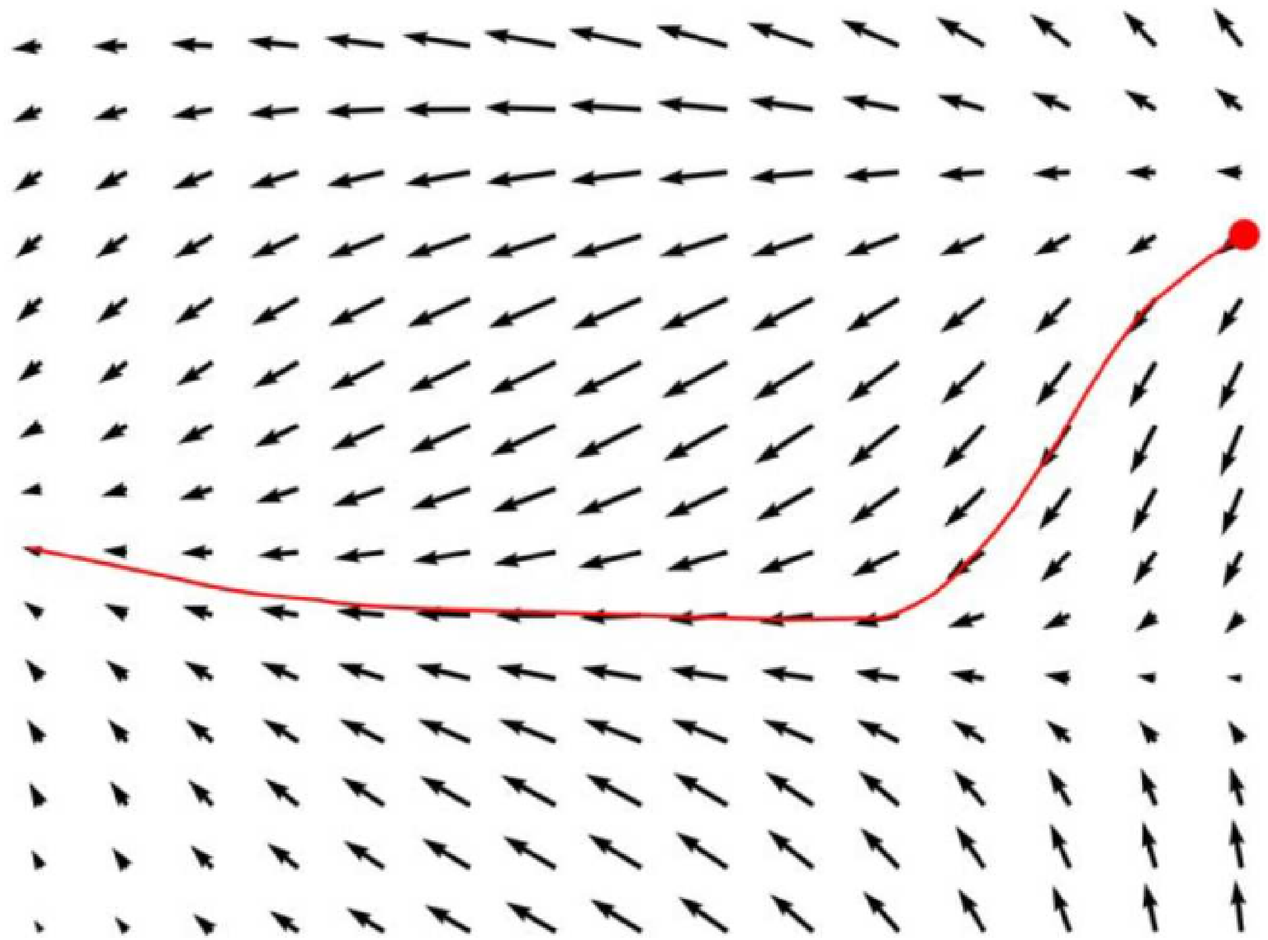
- What is there to visualize?
  - Integral curves
  - “Streamlines”
- Curve $\mathcal{C} \in \mathcal{D}$ such that:
  - Given a bijection
    - $c : \mathcal{C} \rightarrow [0, 1]$
  - $\frac{\partial p}{\partial c} \times \vec{f}(p) = \vec{0}, \quad \forall p \in \mathcal{C}$
- Everywhere tangential to the flow



[Chen]

Streamlines

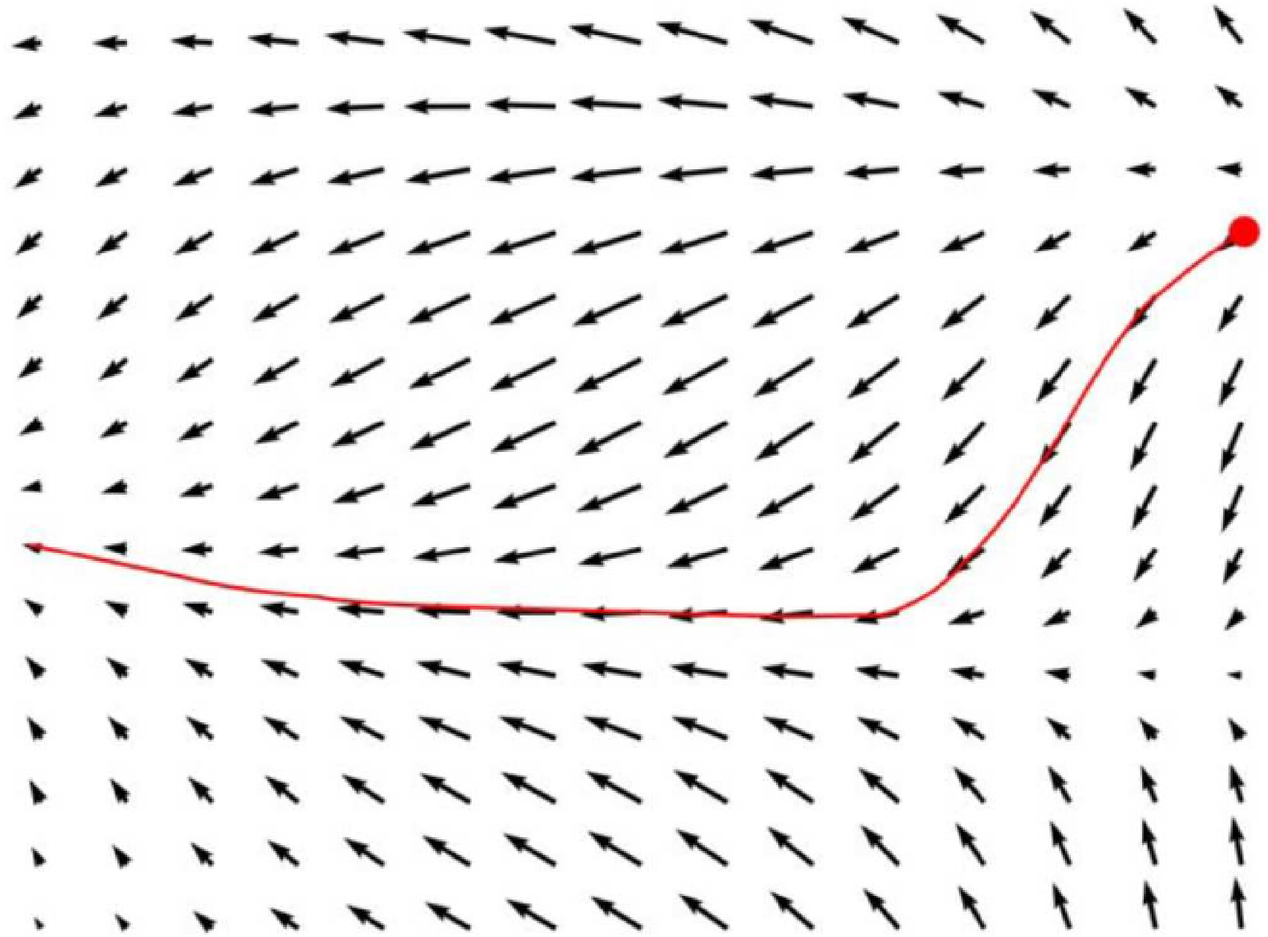
- What is there to visualize?
  - Integral curves
  - “Streamlines”



[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”
- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$

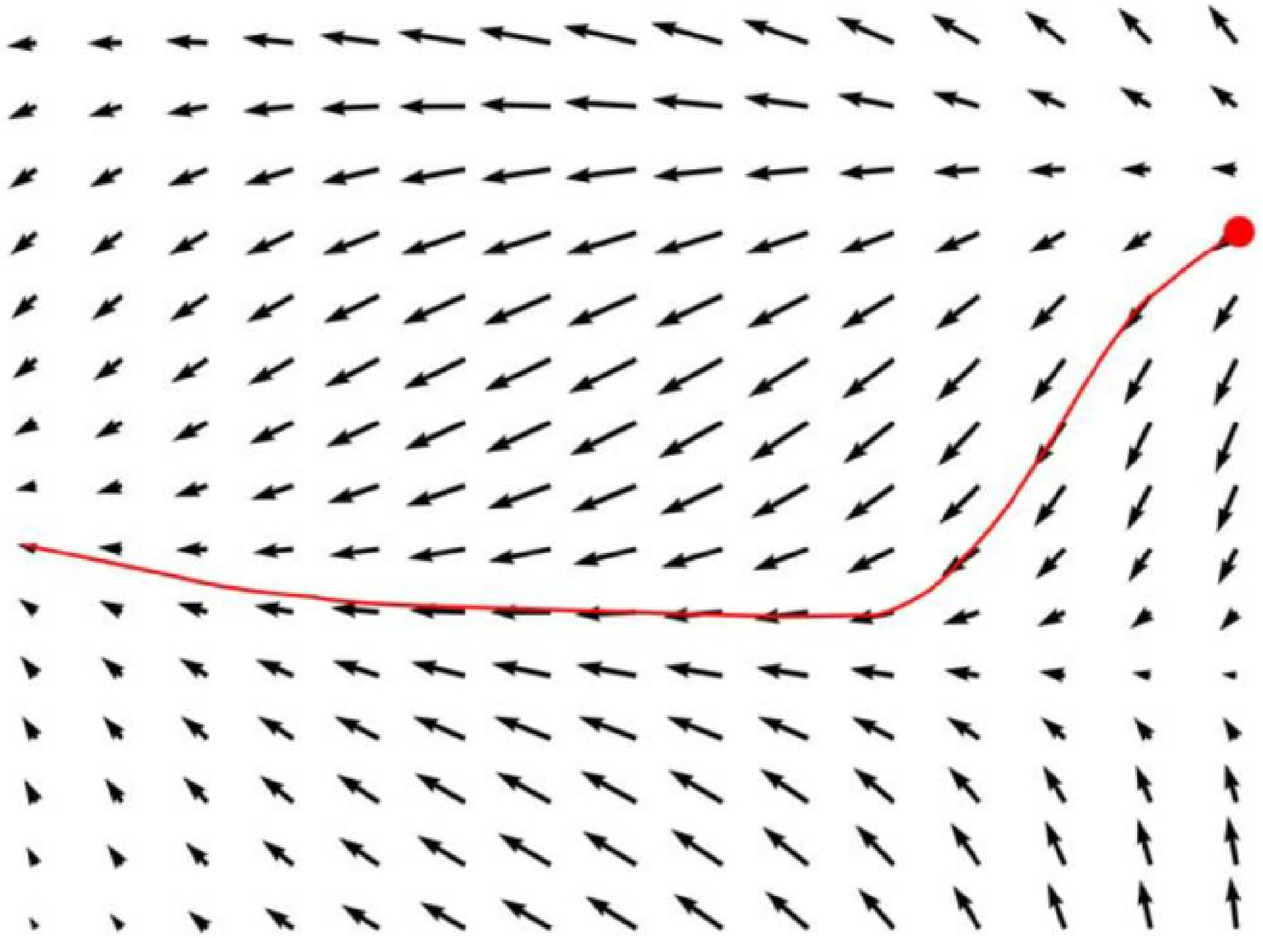


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$

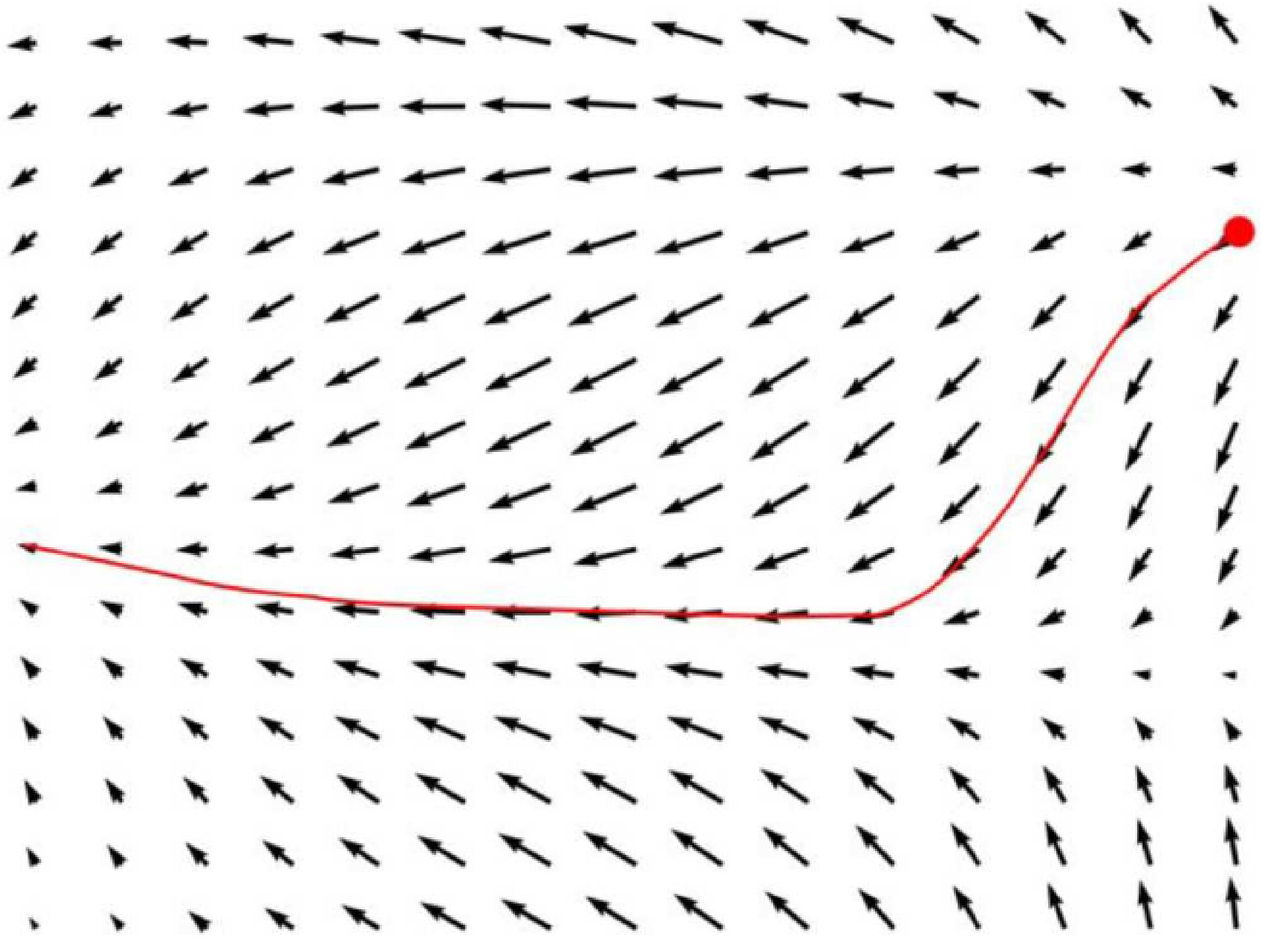


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

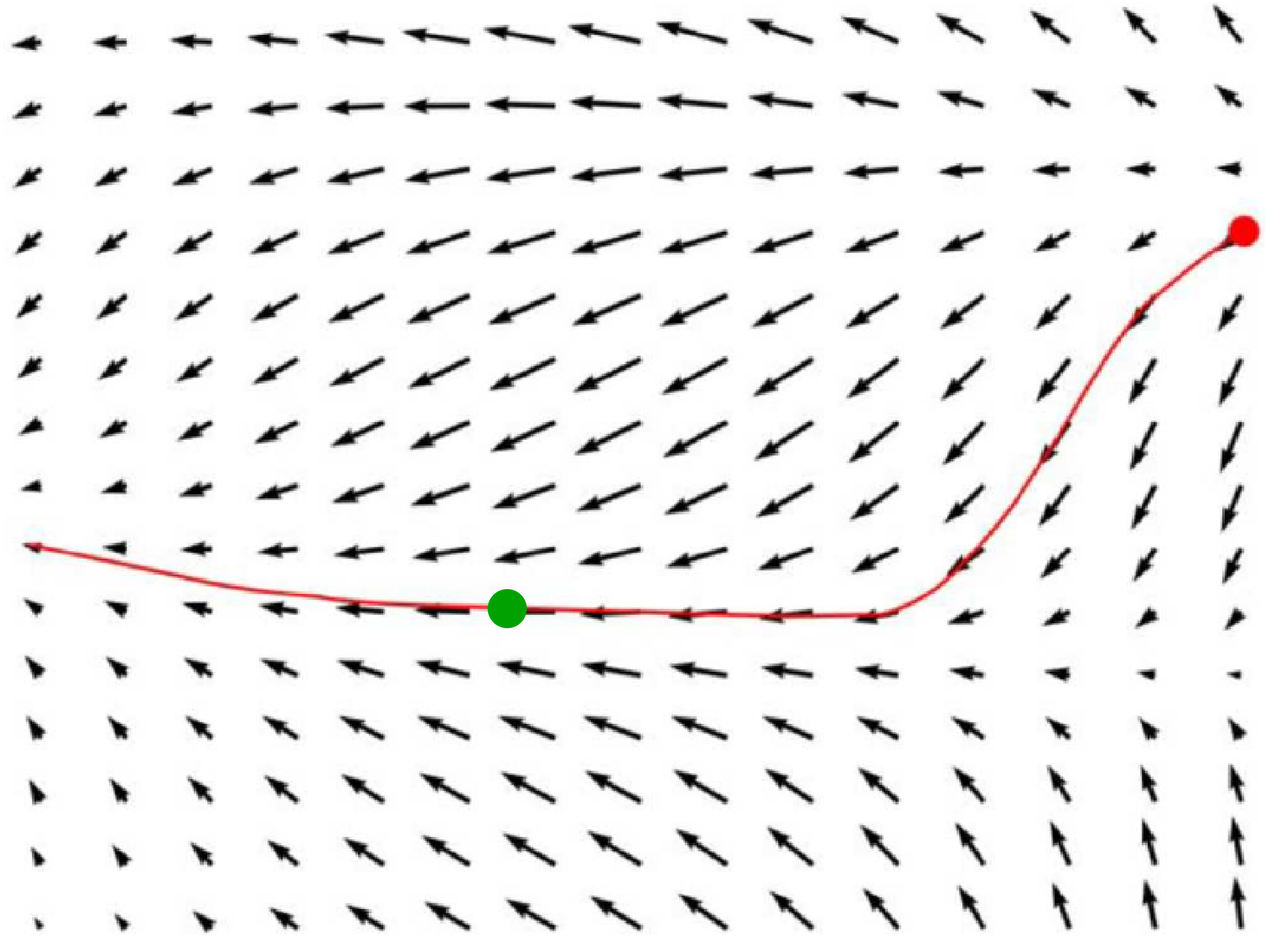


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

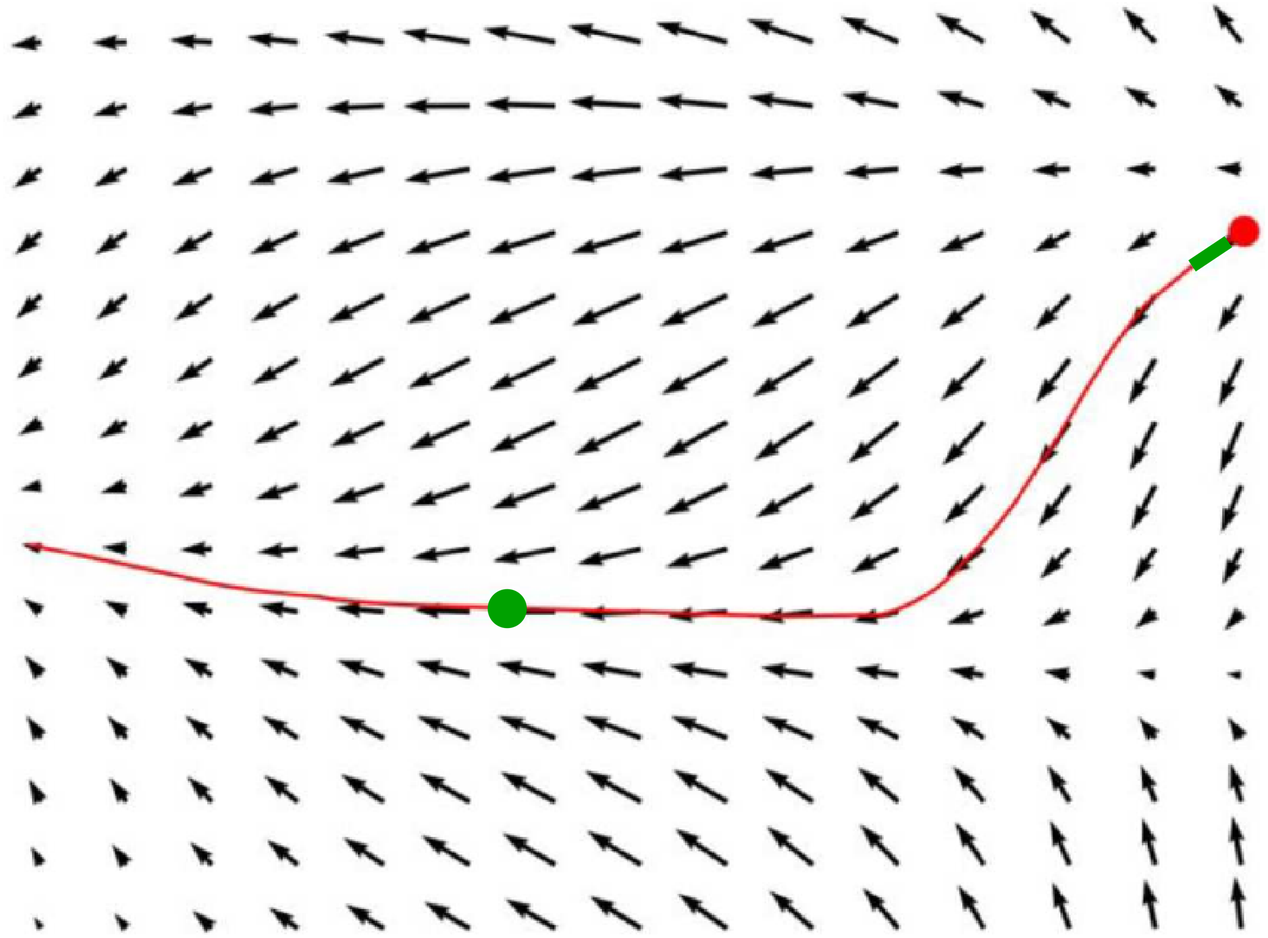


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

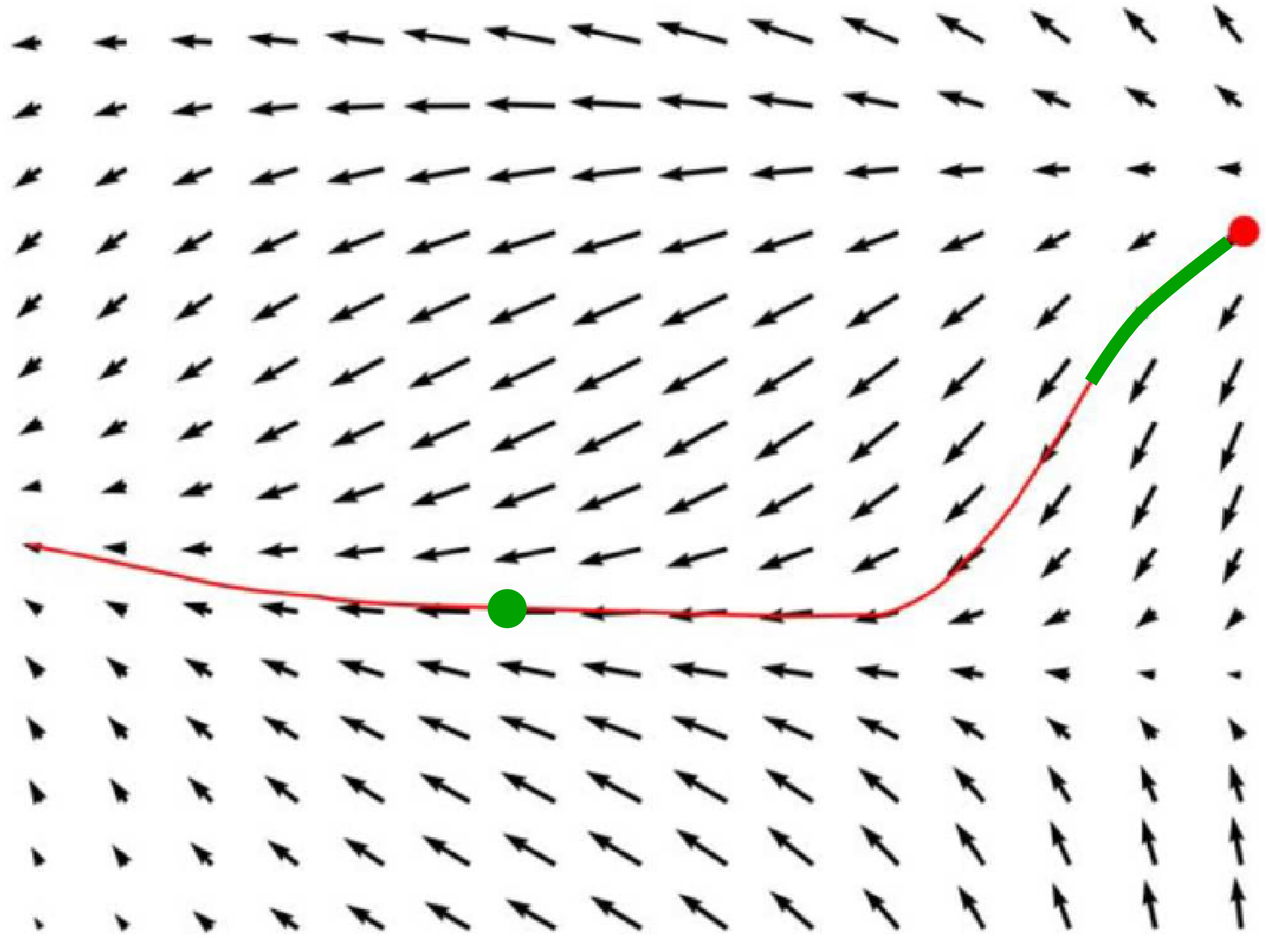


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

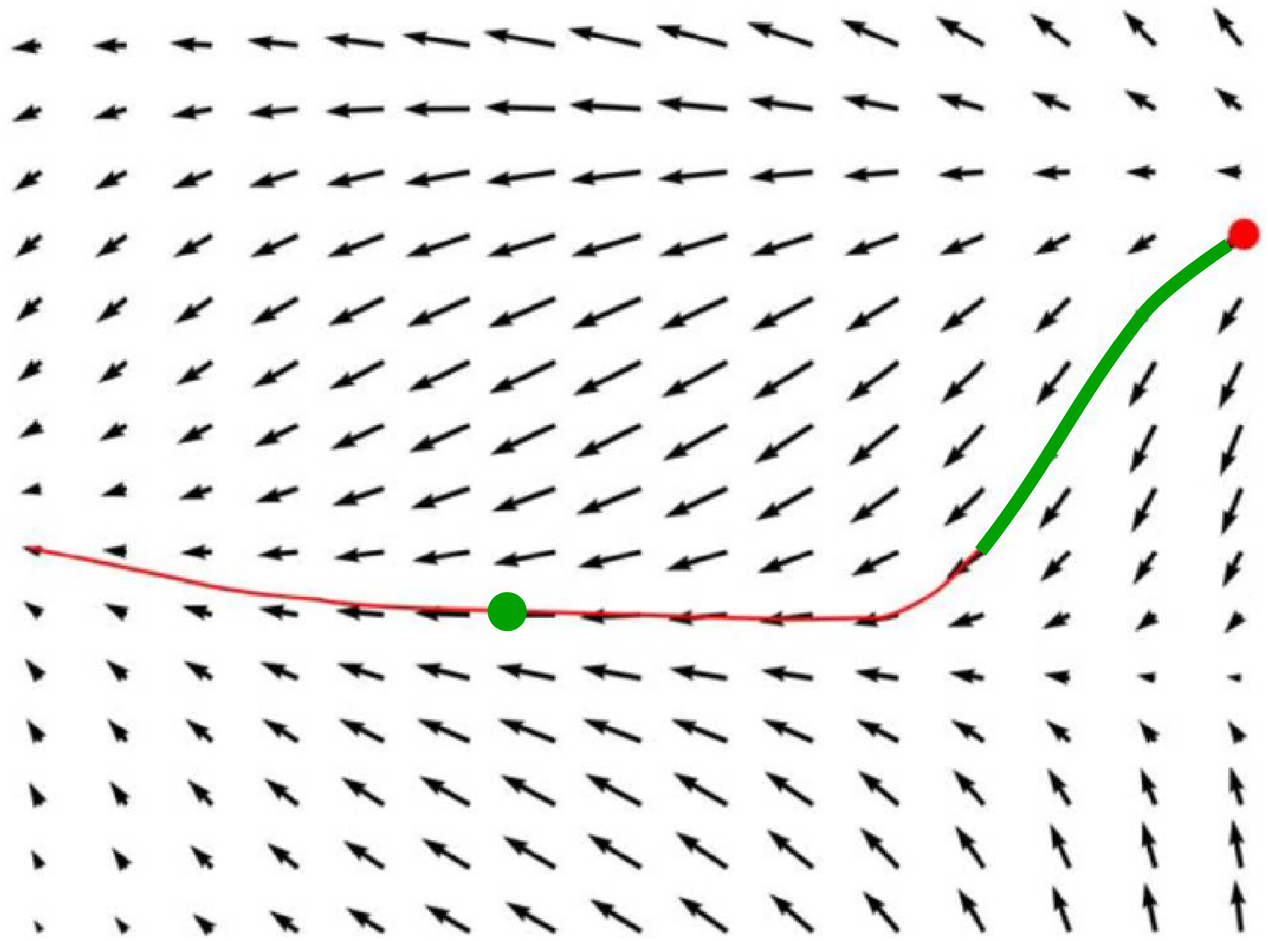


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

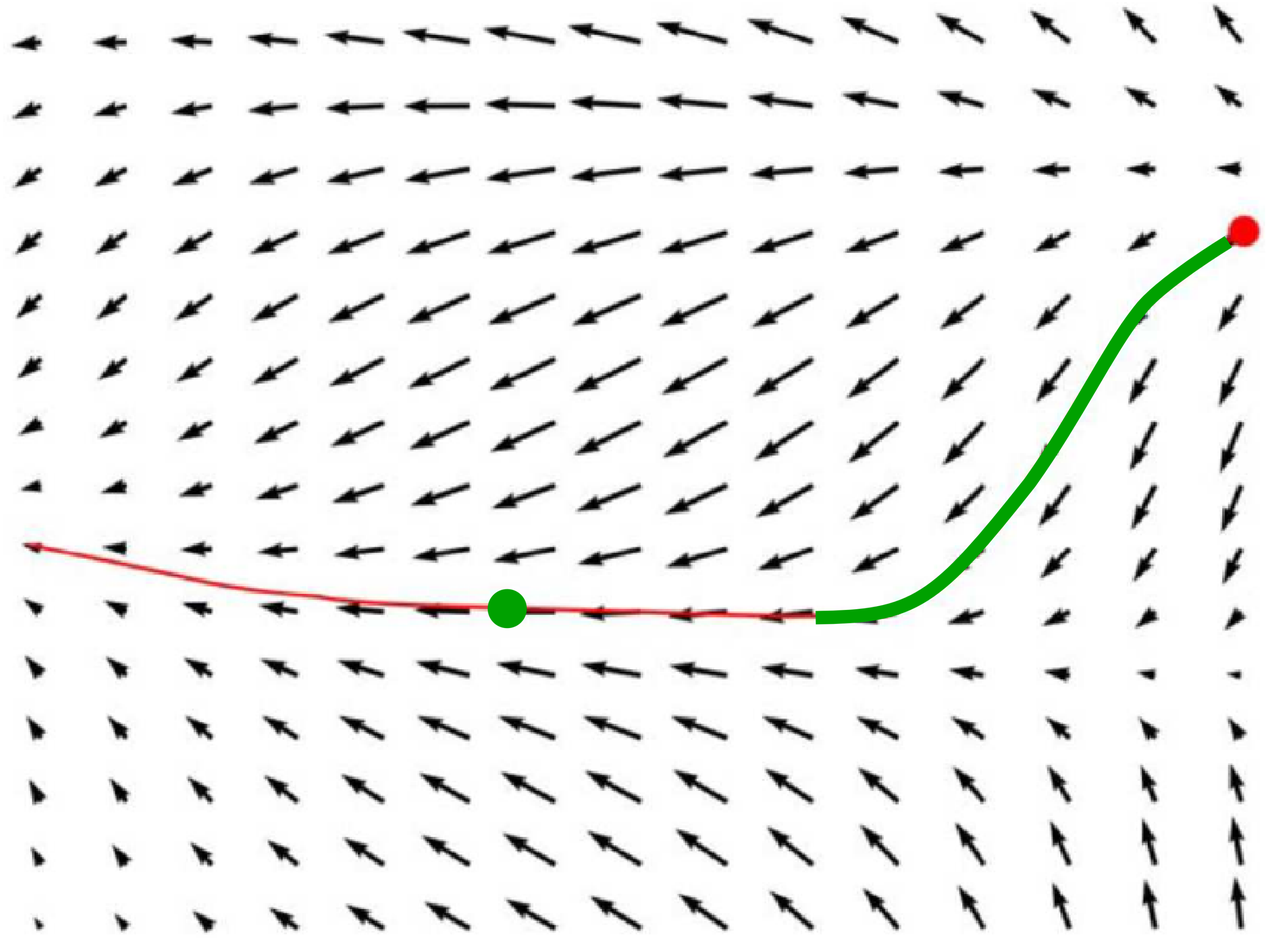


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

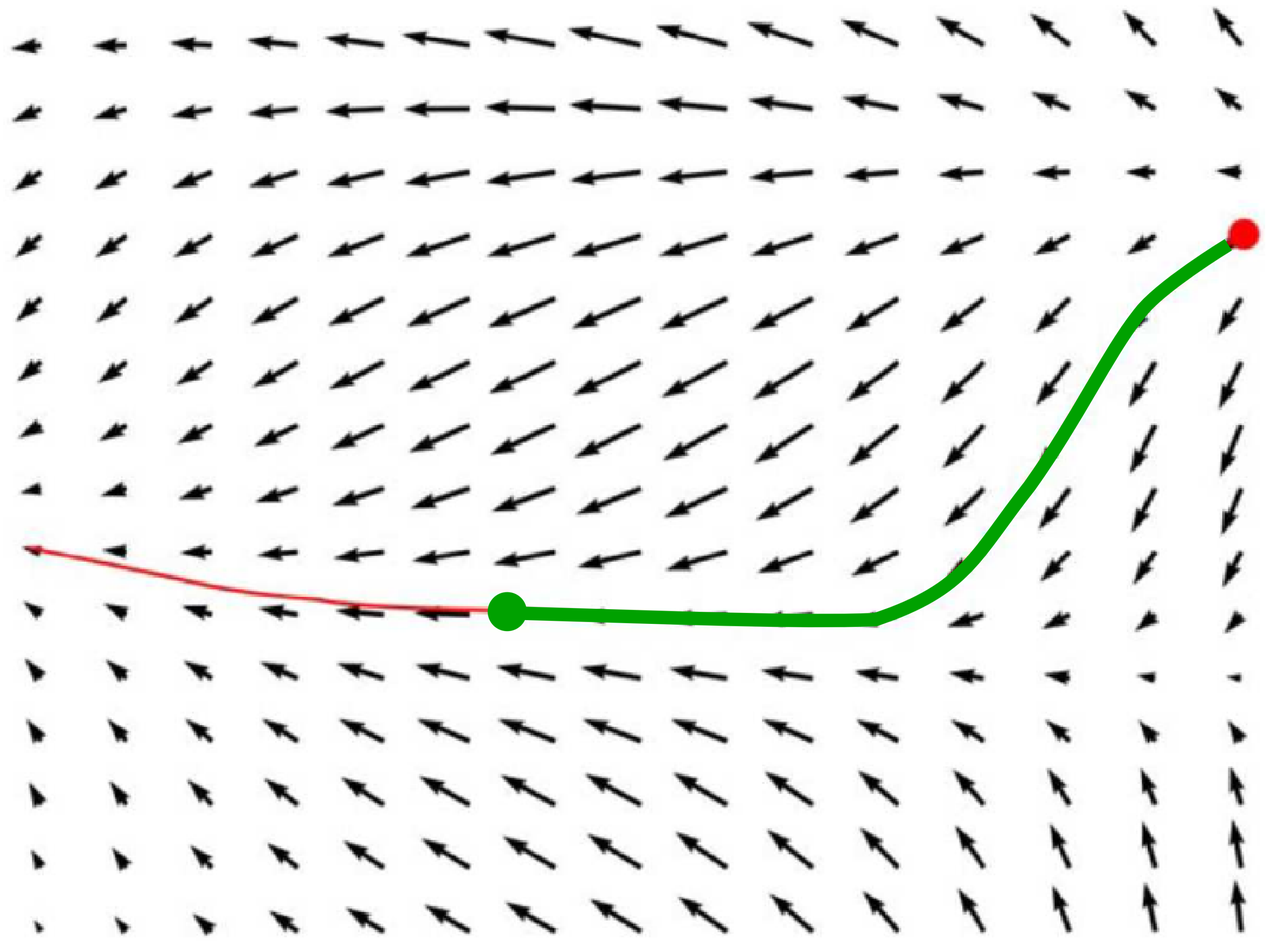


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

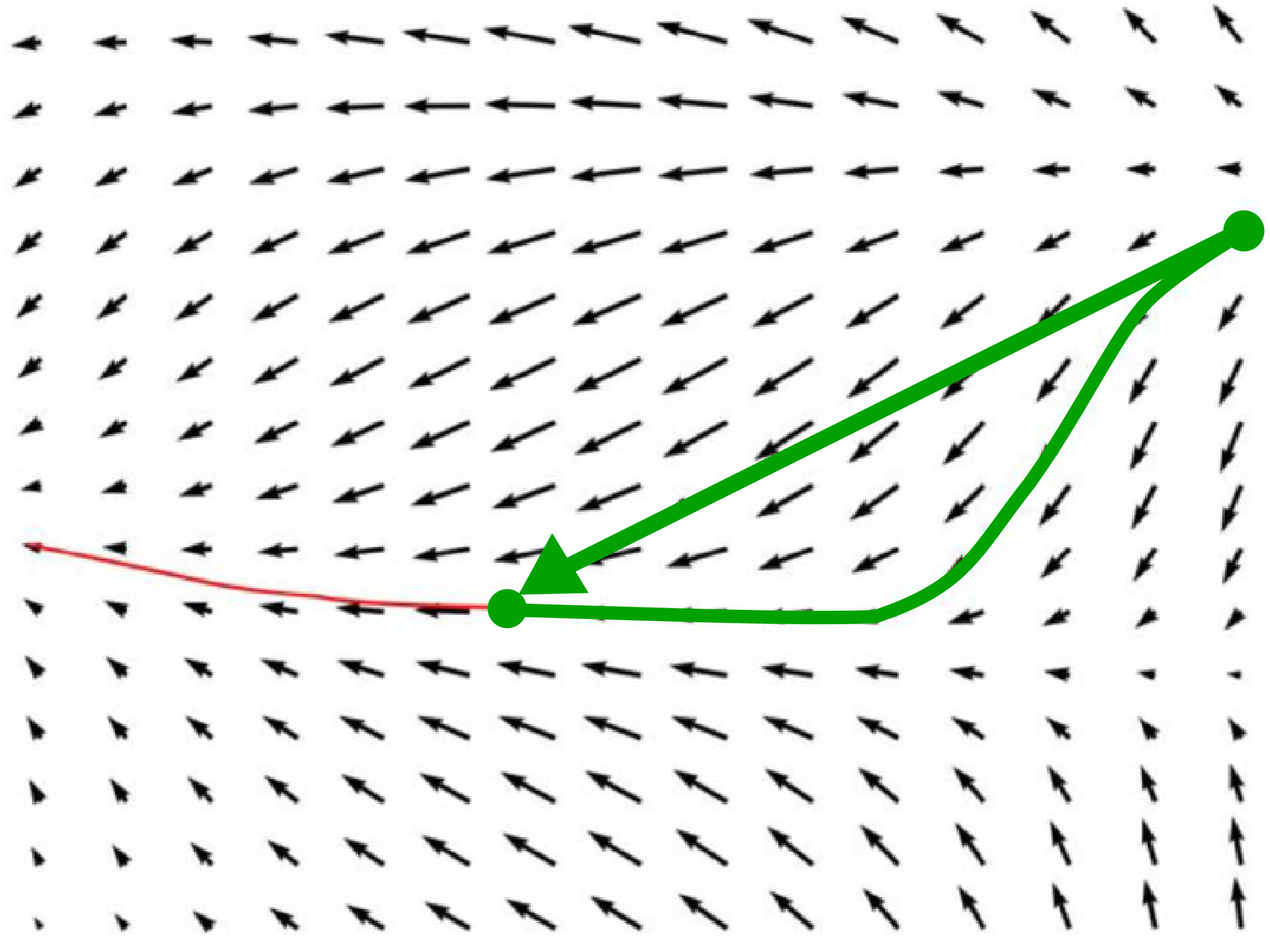


[Chen]

Streamlines

- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$

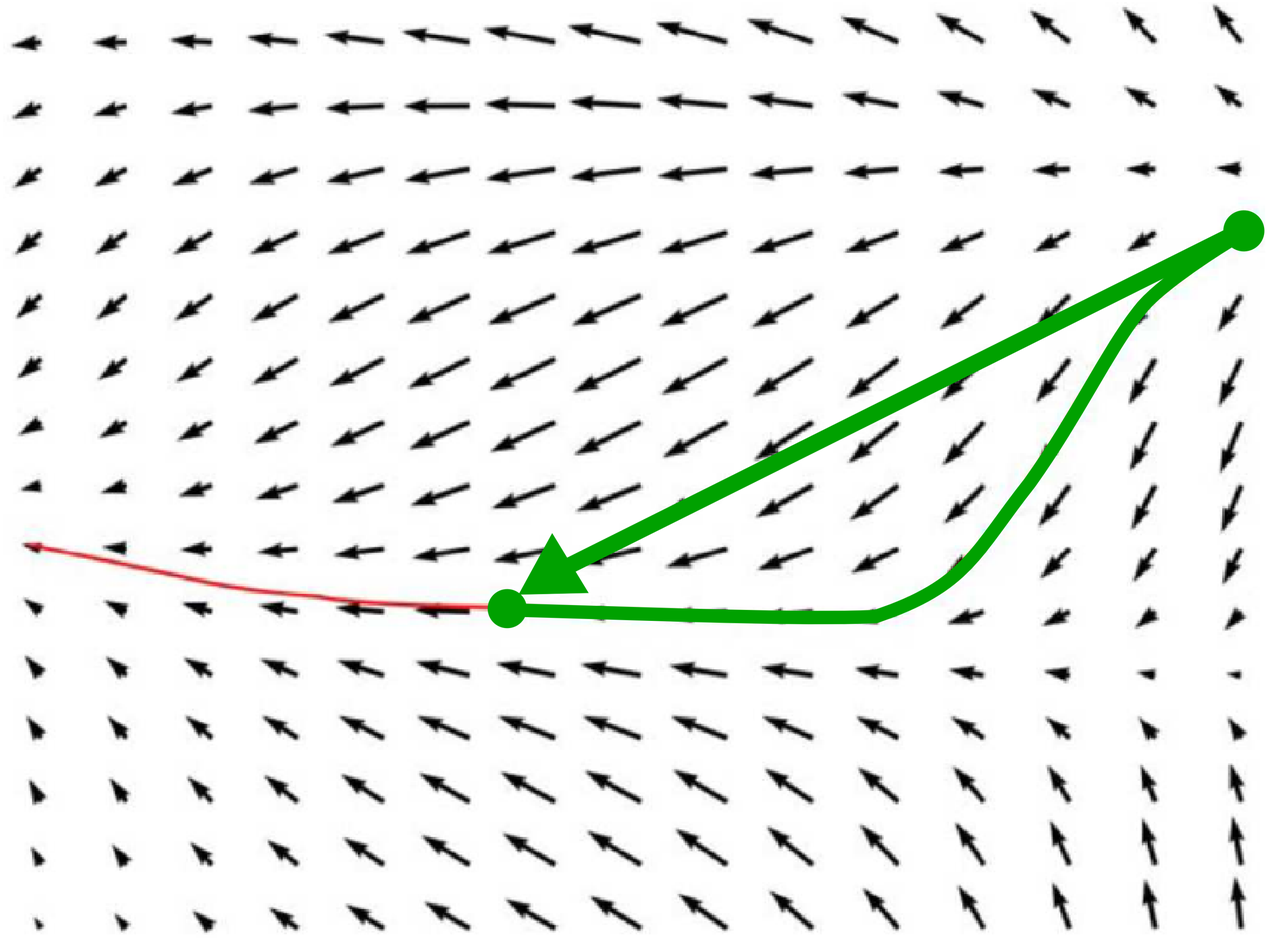


[Chen]

Streamlines

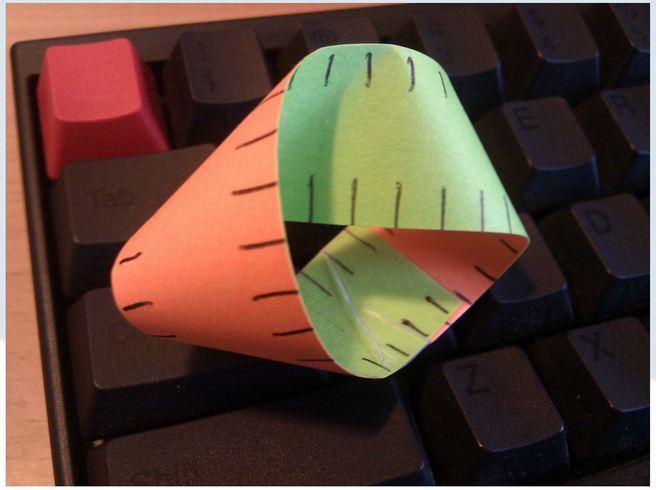
- What is there to visualize?
  - Integral curves
  - “Streamlines”

- Solution to an ODE
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $\forall p \in \mathcal{C}$
  - Unique solution

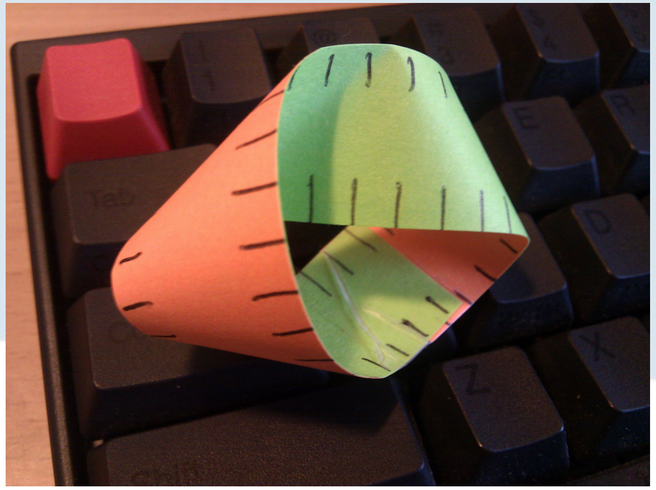


[Chen]

Streamlines on a computer

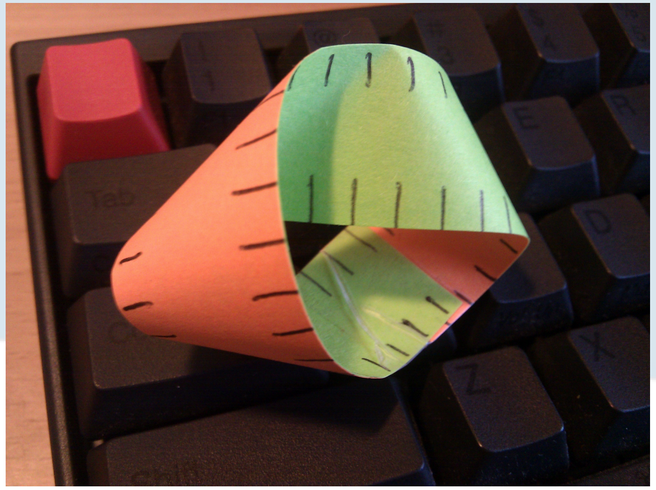


Streamlines on a computer



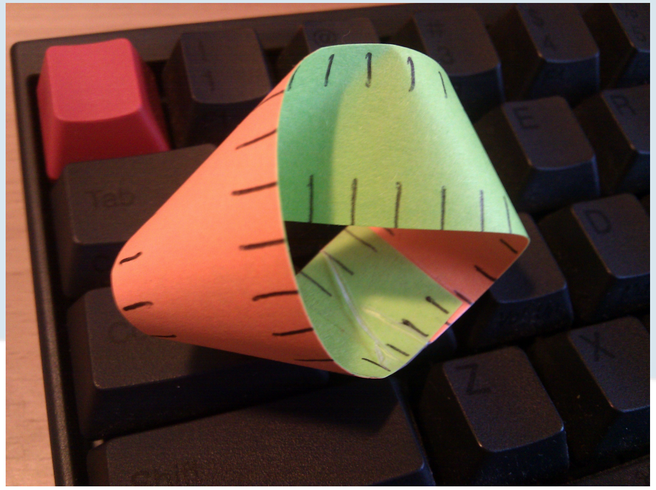
- Numerical integration of the ODE

Streamlines on a computer



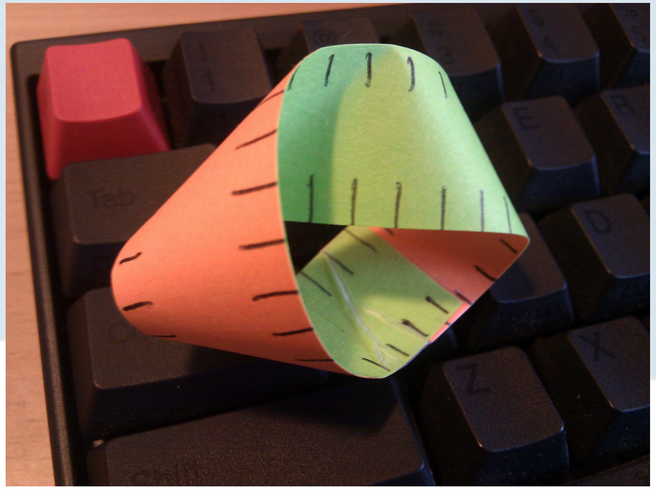
- Numerical integration of the ODE
  - Euler method

Streamlines on a computer



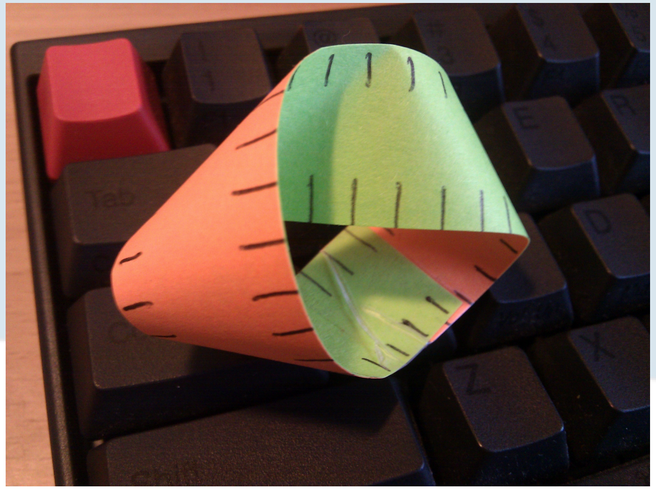
- Numerical integration of the ODE
  - Euler method
  - Runge-Kutta
    - Higher oder approximations

Streamlines on a computer



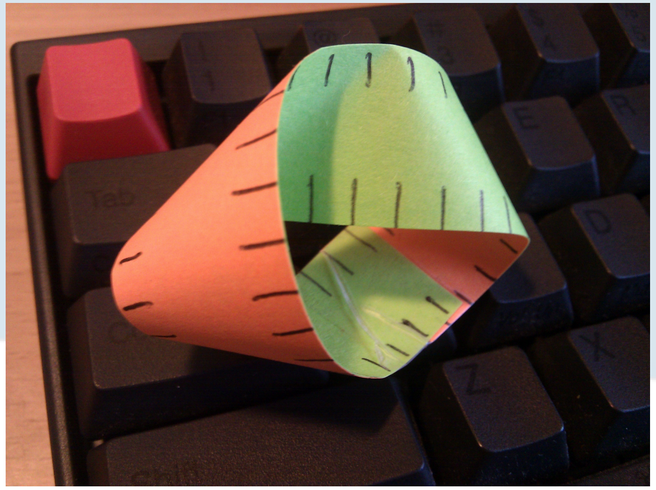
- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations

Streamlines on a computer



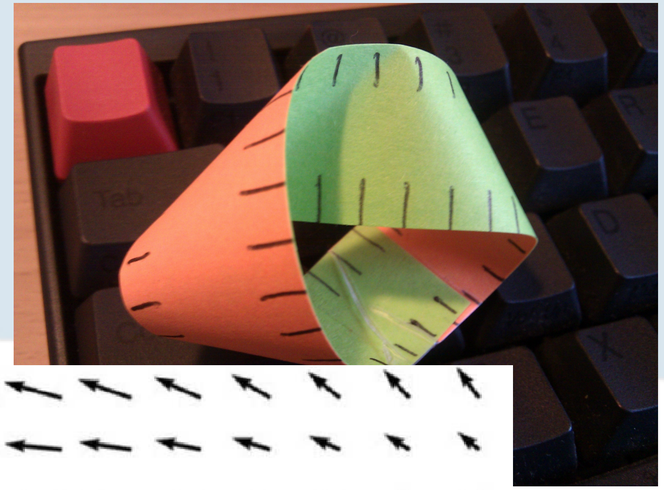
- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$

Streamlines on a computer

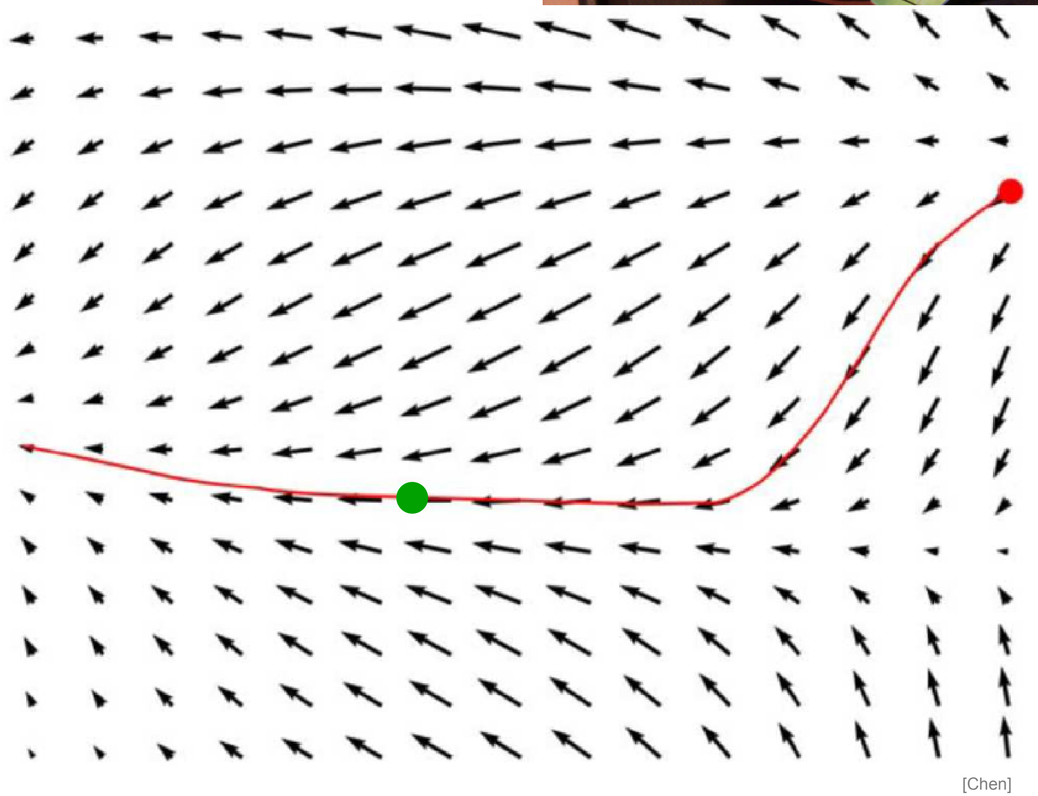


- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + lim_{q\rightarrow\infty} \sum_{u\in[0,q]} \frac{\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))||}.\frac{c(p)}{q+1}, \quad u \in \mathbb{N}$

Streamlines on a computer

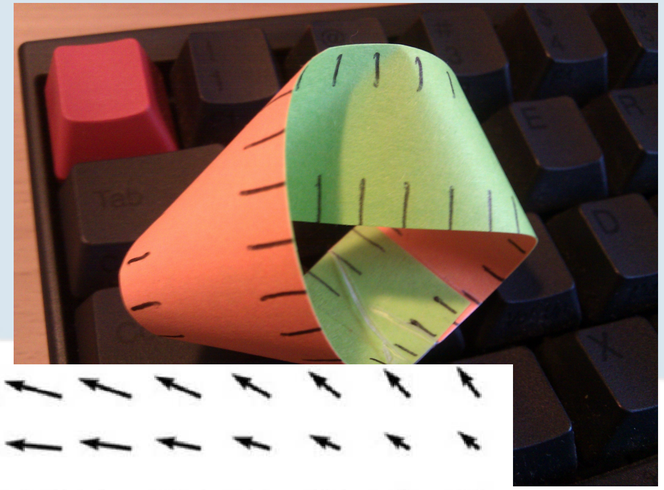


- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + lim_{q\rightarrow\infty} \sum_{u\in[0,q]} \frac{\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))||}.\frac{c(p)}{q+1}, \quad u \in \mathbb{N}$

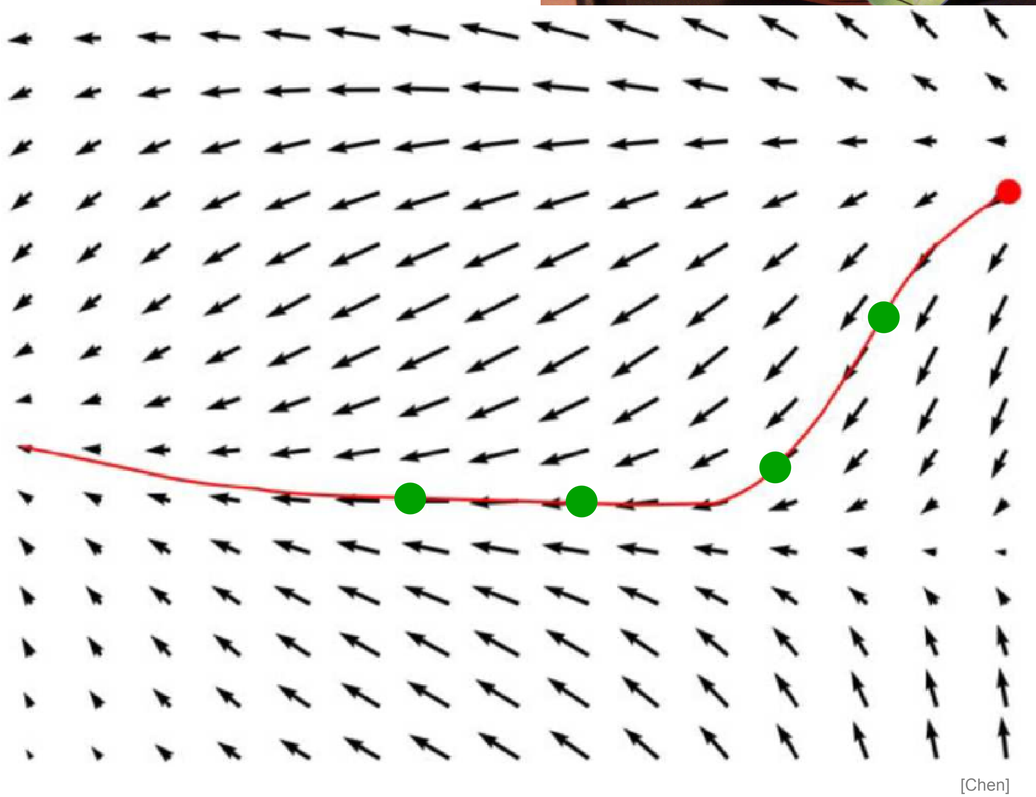


[Chen]

Streamlines on a computer

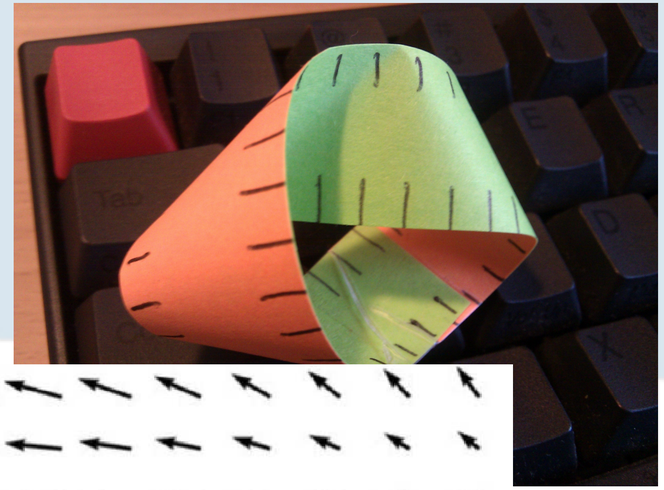


- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + lim_{q\rightarrow\infty} \sum_{u\in[0,q]} \frac{\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))||}.\frac{c(p)}{q+1}, \quad u \in \mathbb{N}$

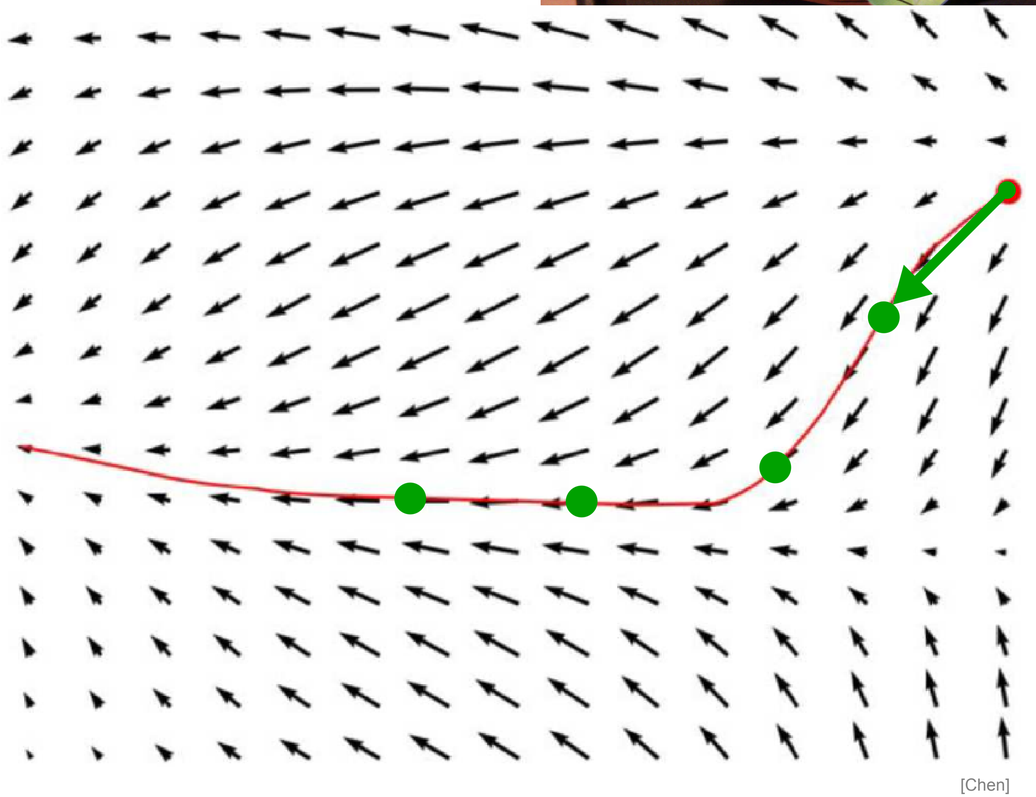


[Chen]

Streamlines on a computer

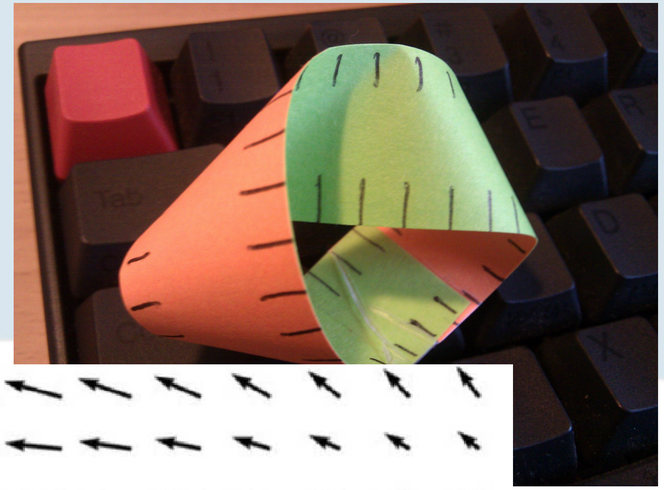


- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + lim_{q\rightarrow\infty} \sum_{u\in[0,q]} \frac{\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))||}.\frac{c(p)}{q+1}, \quad u \in \mathbb{N}$

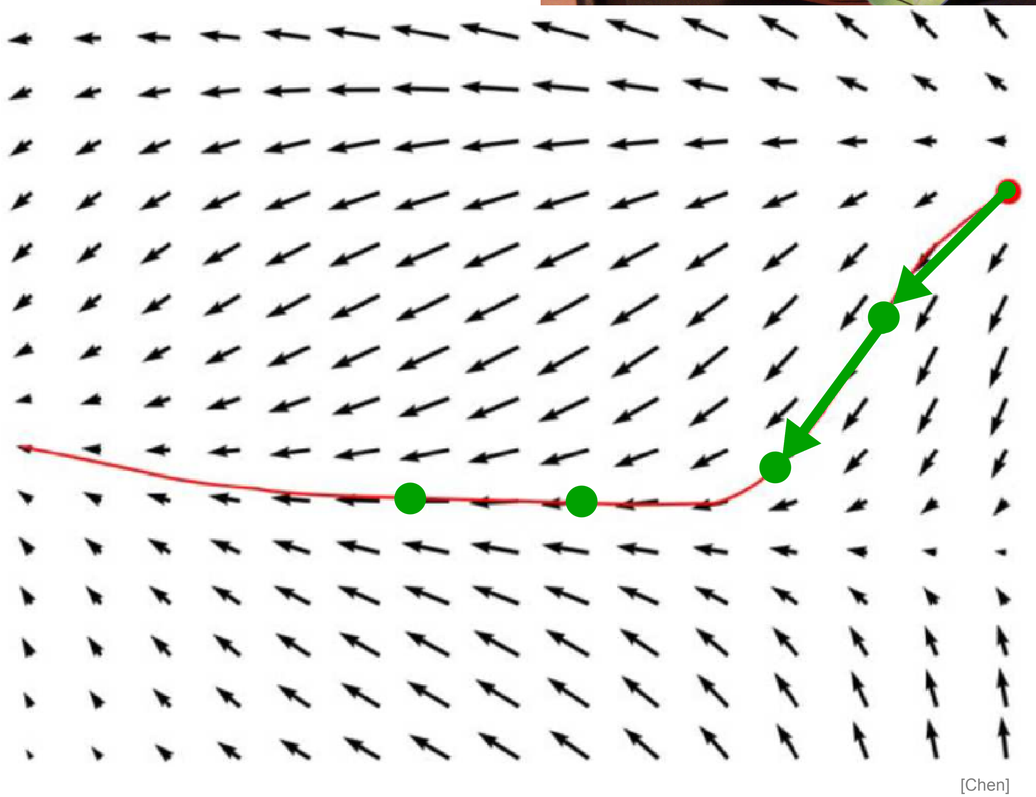


[Chen]

Streamlines on a computer

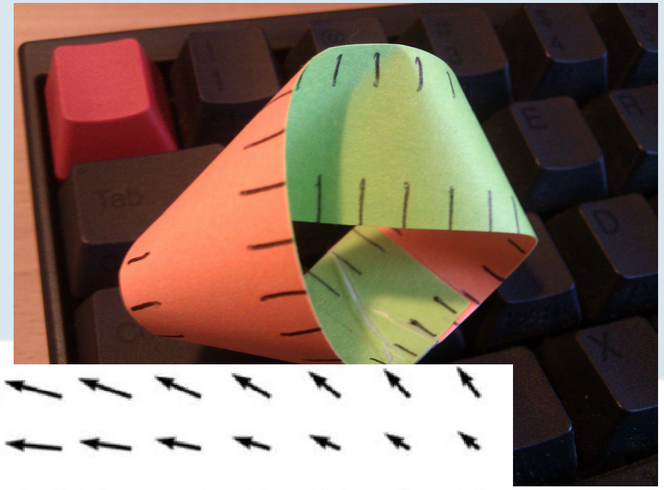


- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + lim_{q\rightarrow\infty} \sum_{u\in[0,q]} \frac{\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))||}.\frac{c(p)}{q+1}, \quad u \in \mathbb{N}$

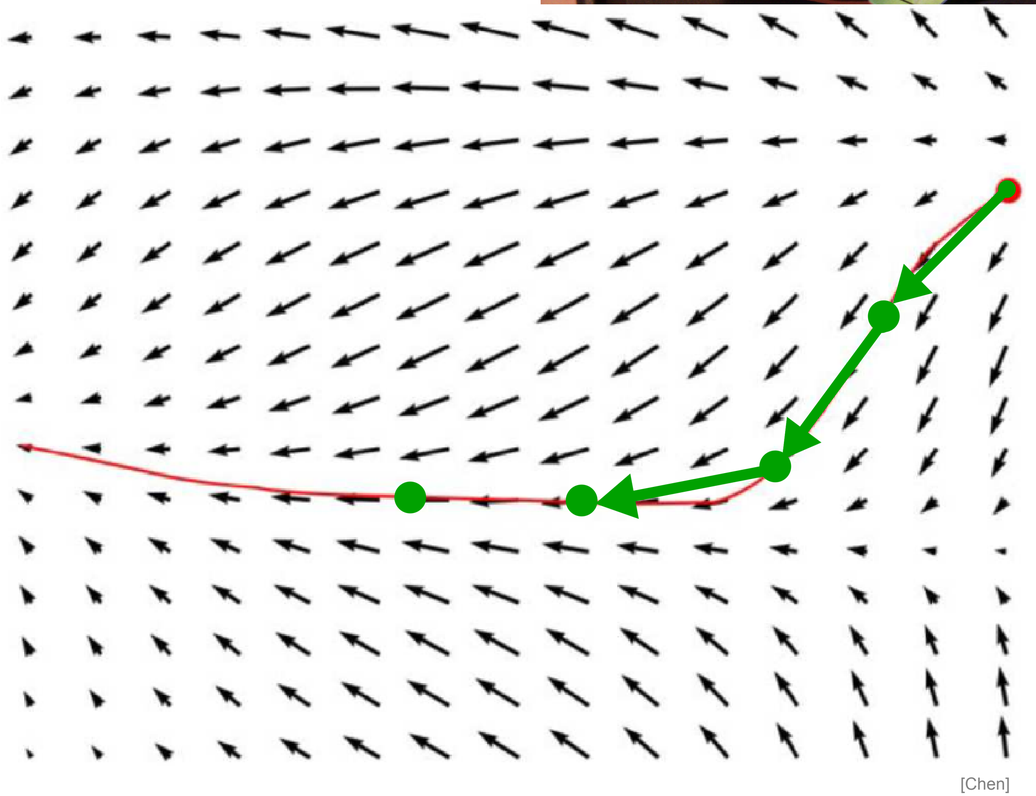


[Chen]

Streamlines on a computer

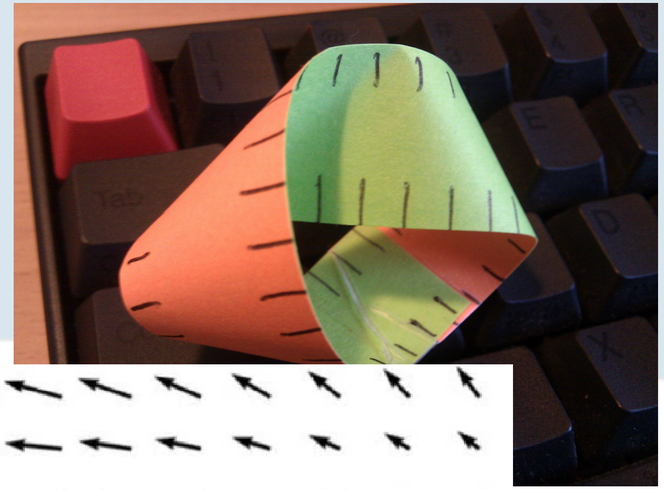


- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + lim_{q\rightarrow\infty} \sum_{u\in[0,q]} \frac{\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))||}.\frac{c(p)}{q+1}, \quad u \in \mathbb{N}$

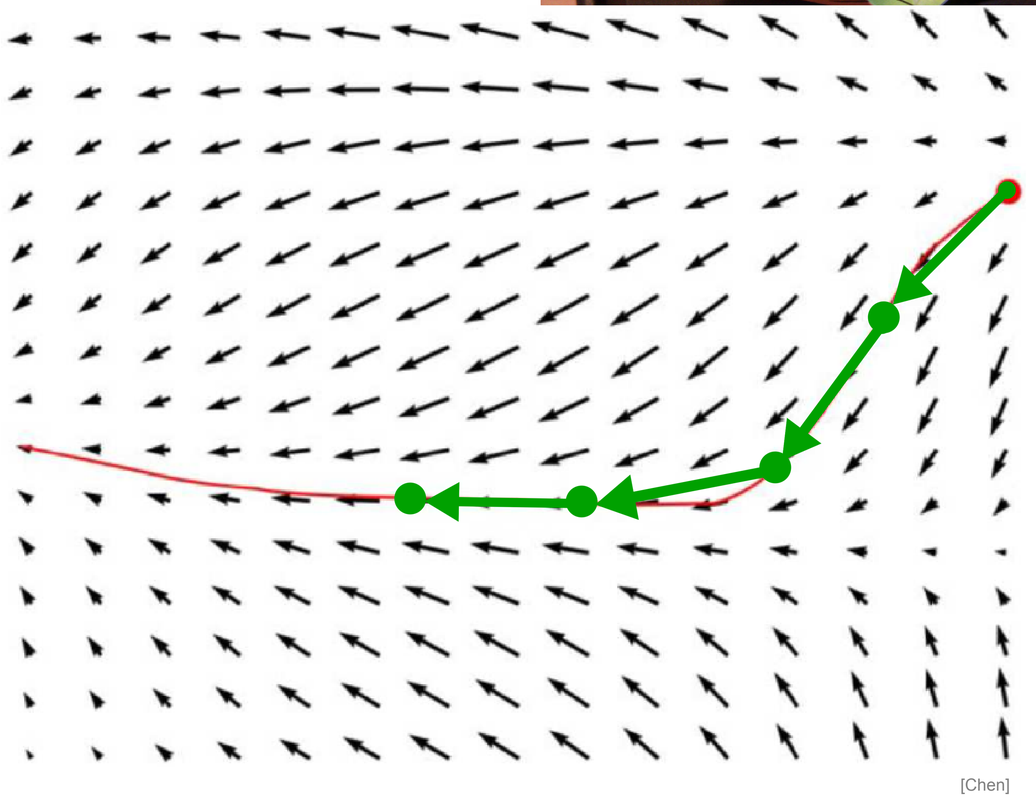


[Chen]

Streamlines on a computer



- Numerical integration of the ODE
  - **Euler method**
  - Runge-Kutta
    - Higher oder approximations
- Discretization
  - $c : \mathcal{C} \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + lim_{q\rightarrow\infty} \sum_{u\in[0,q]} \frac{\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u.\frac{c(p)}{q+1}))||}.\frac{c(p)}{q+1}, \quad u \in \mathbb{N}$



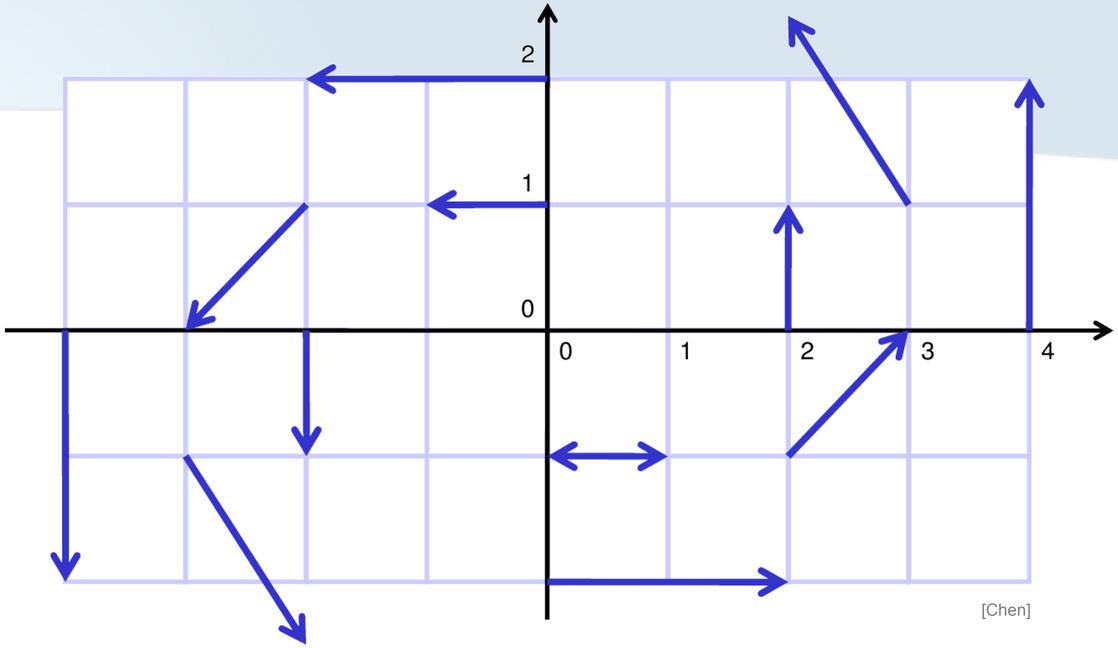
[Chen]

Streamlines on a computer

- Euler integration algorithm

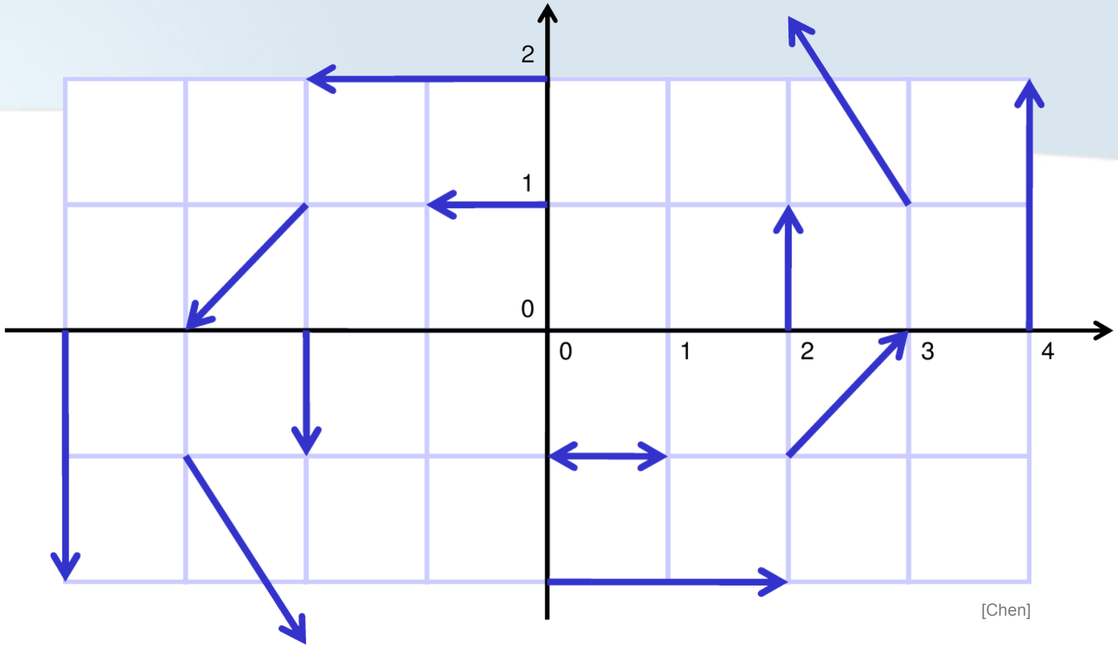
Streamlines on a computer

- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$



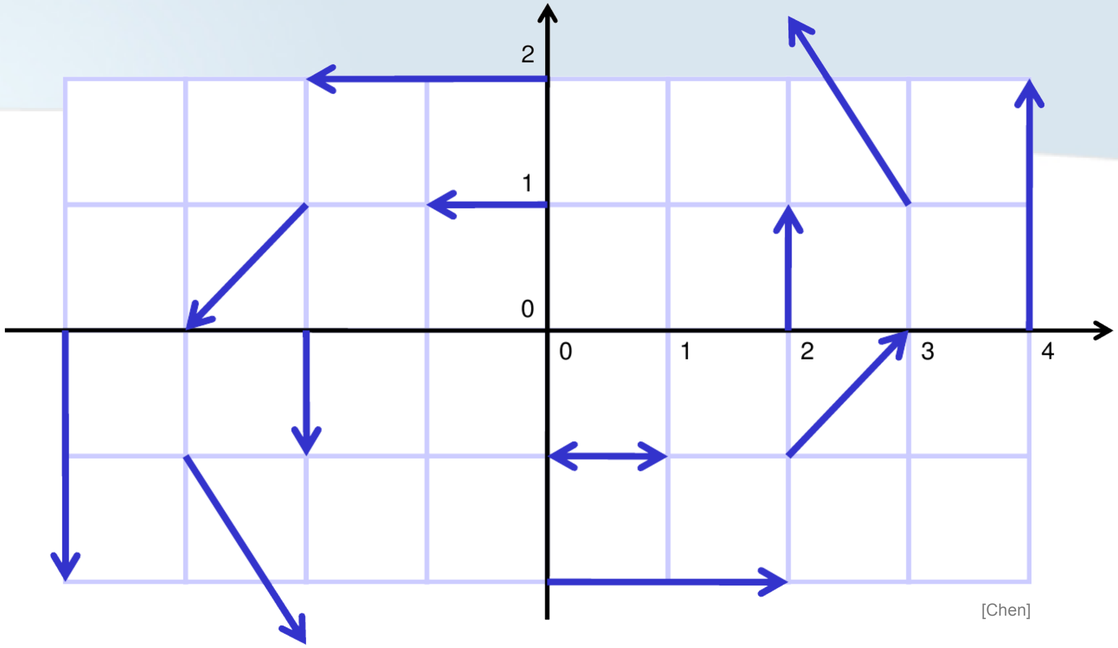
Streamlines on a computer

- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$



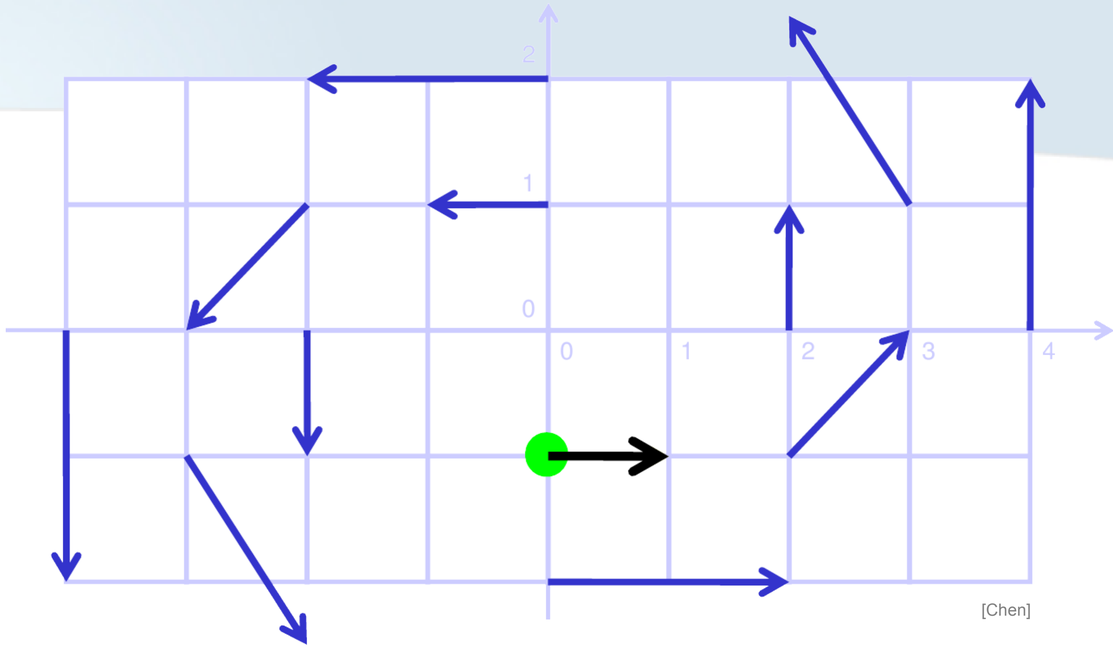
Streamlines on a computer

- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat



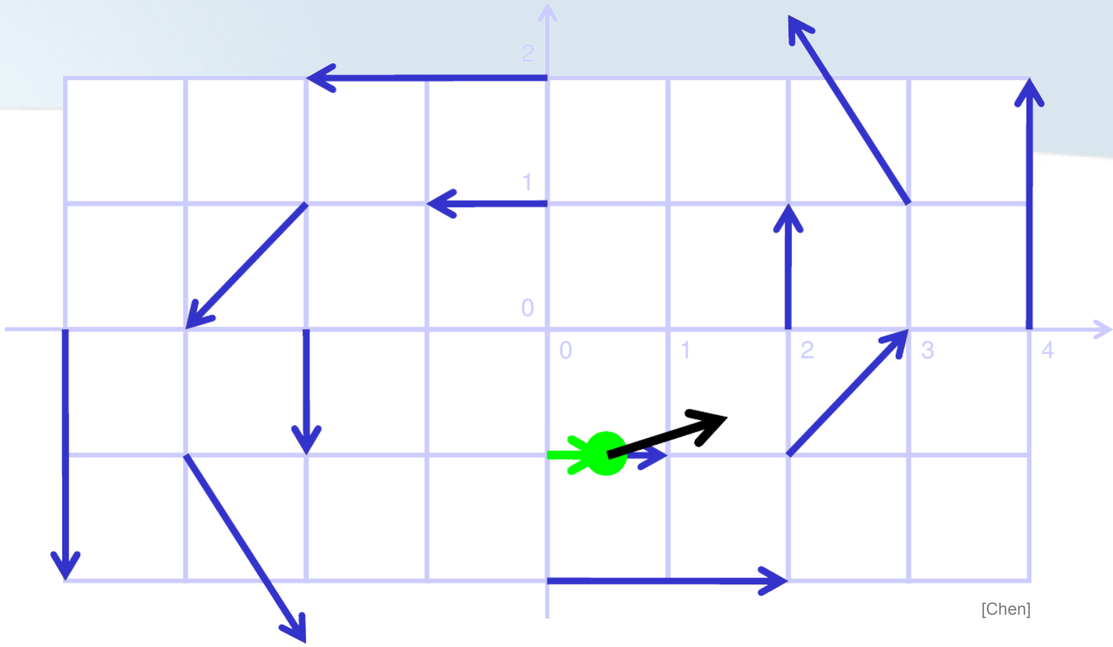
Streamlines on a computer

- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$



Streamlines on a computer

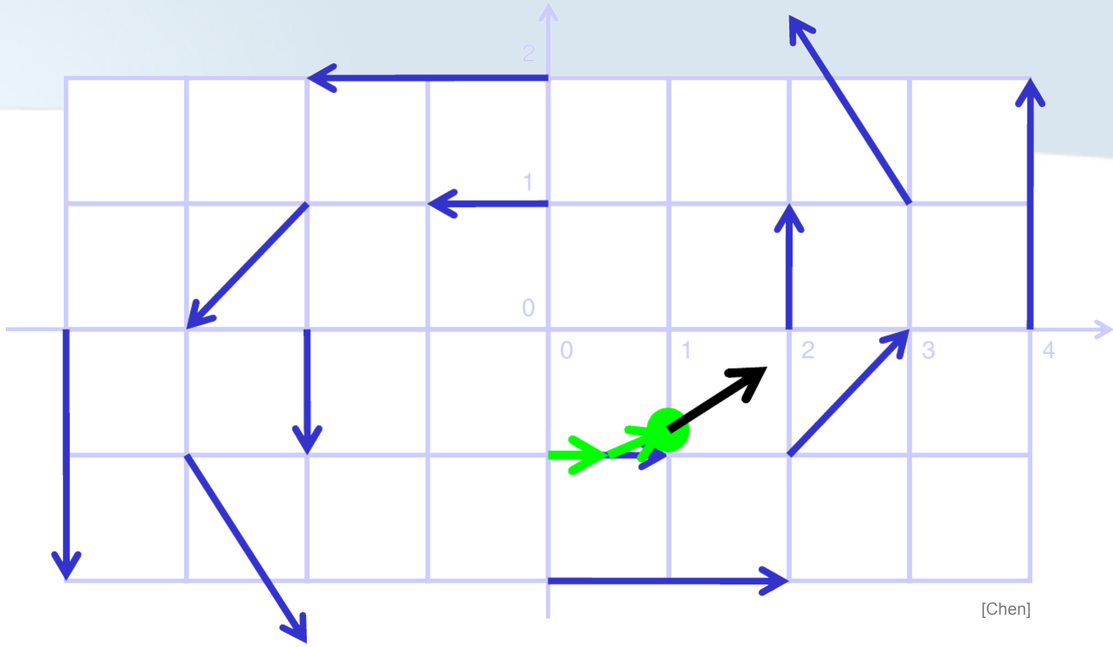
- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$



[Chen]

Streamlines on a computer

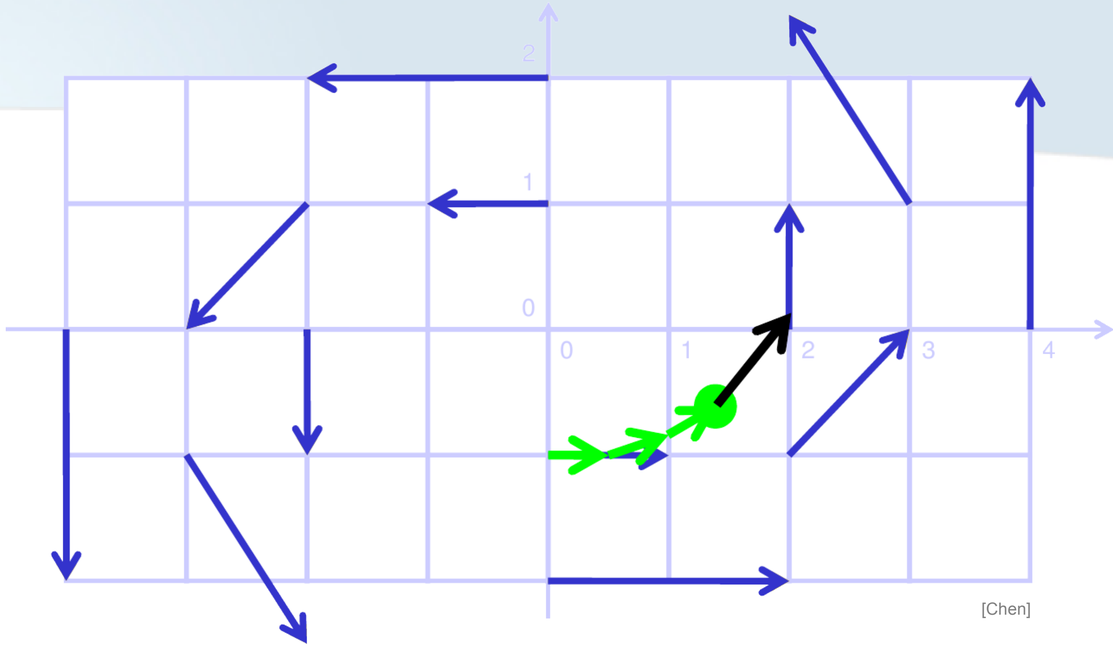
- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$
    - $u \leftarrow u+1$



[Chen]

Streamlines on a computer

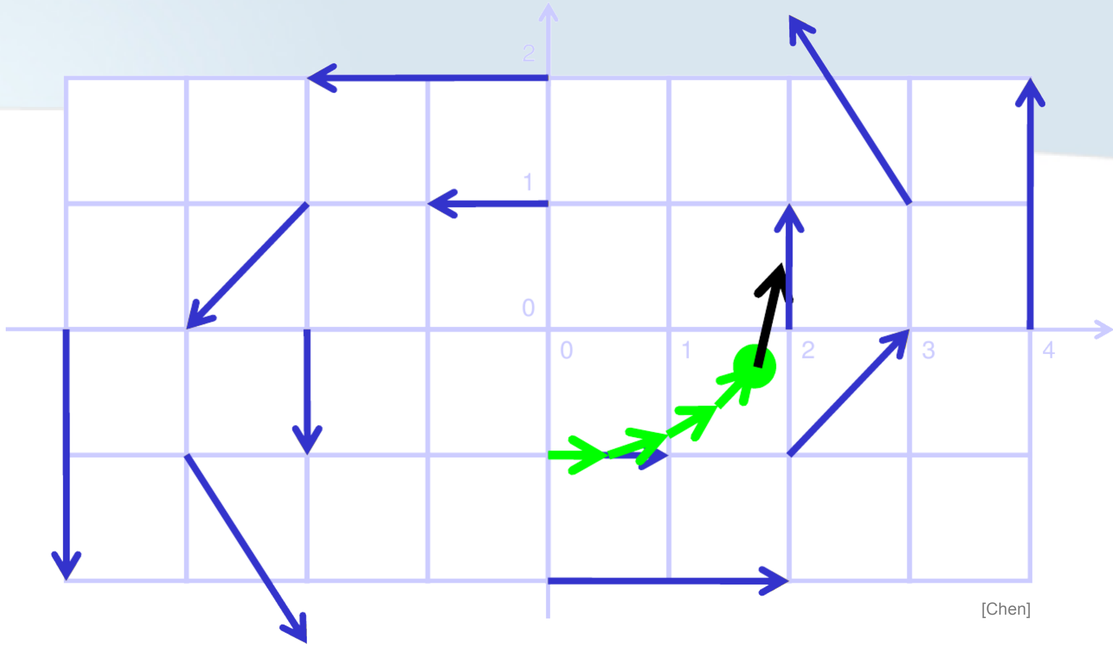
- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$
    - $u \leftarrow u+1$
  - Until $u \leq q$



[Chen]

Streamlines on a computer

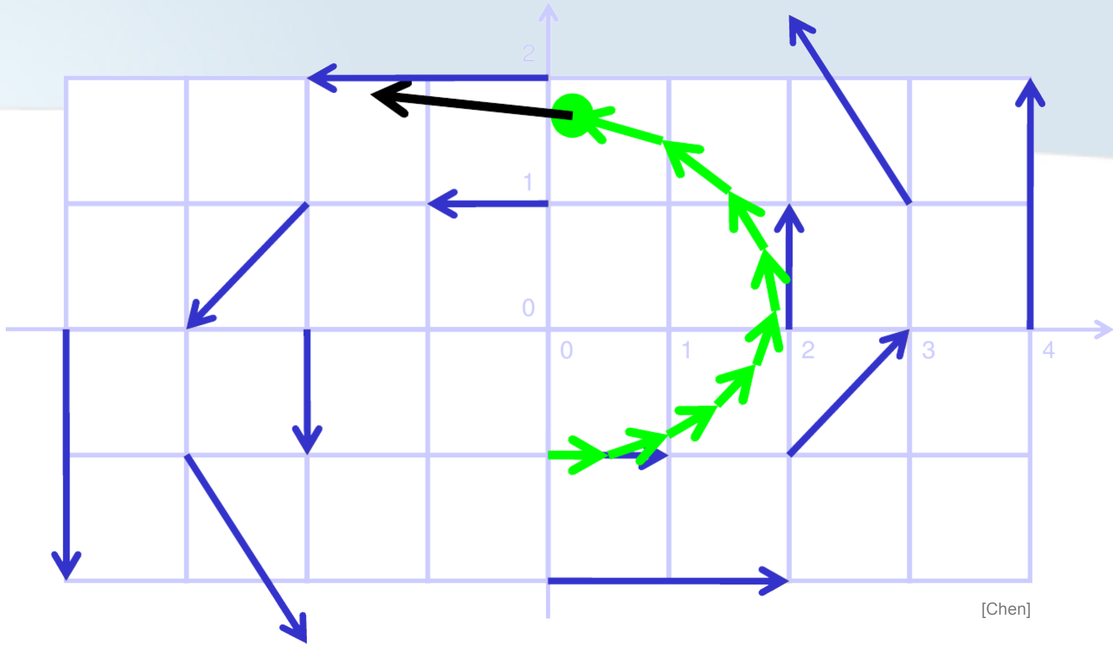
- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$
    - $u \leftarrow u+1$
  - Until $u \leq q$



[Chen]

Streamlines on a computer

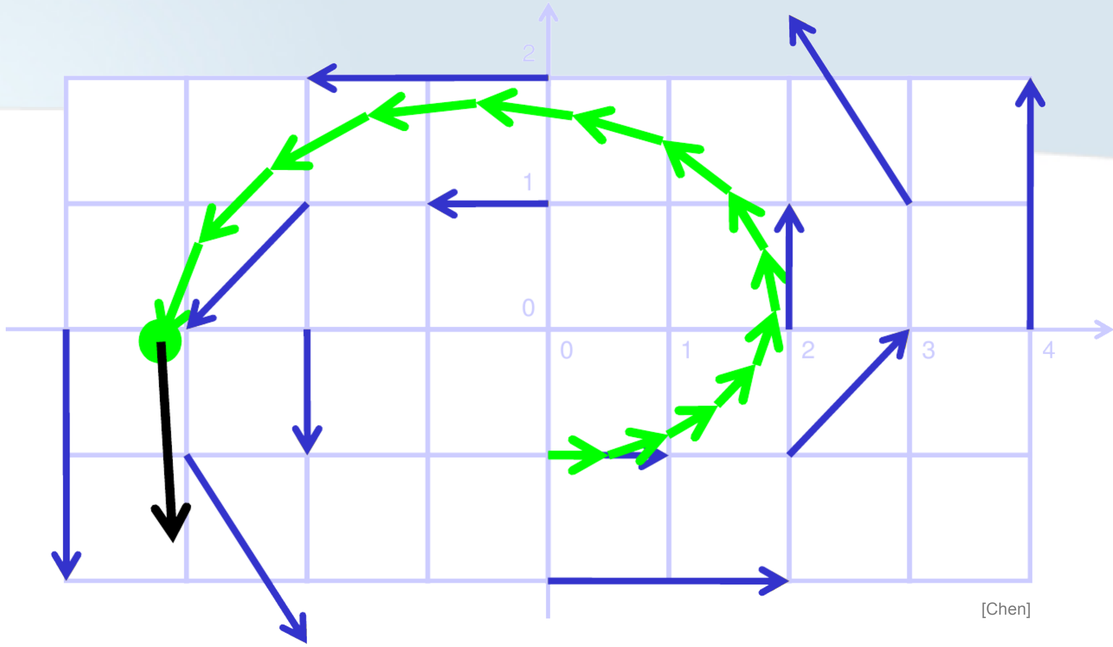
- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$
    - $u \leftarrow u+1$
  - Until $u \leq q$



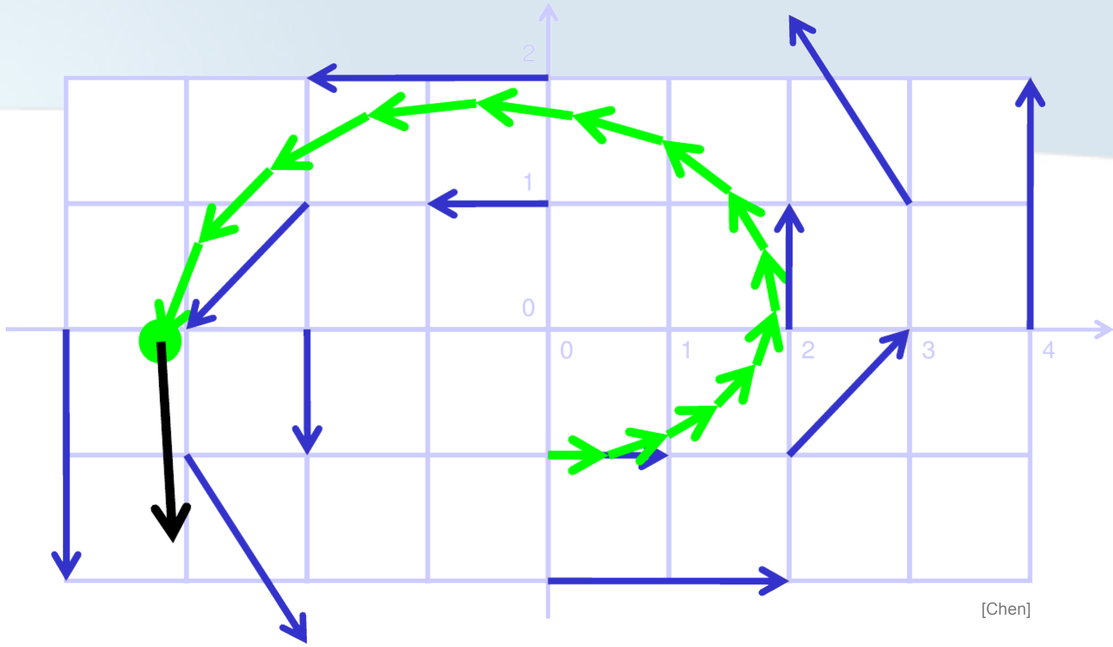
[Chen]

Streamlines on a computer

- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$
    - $u \leftarrow u+1$
  - Until $u \leq q$



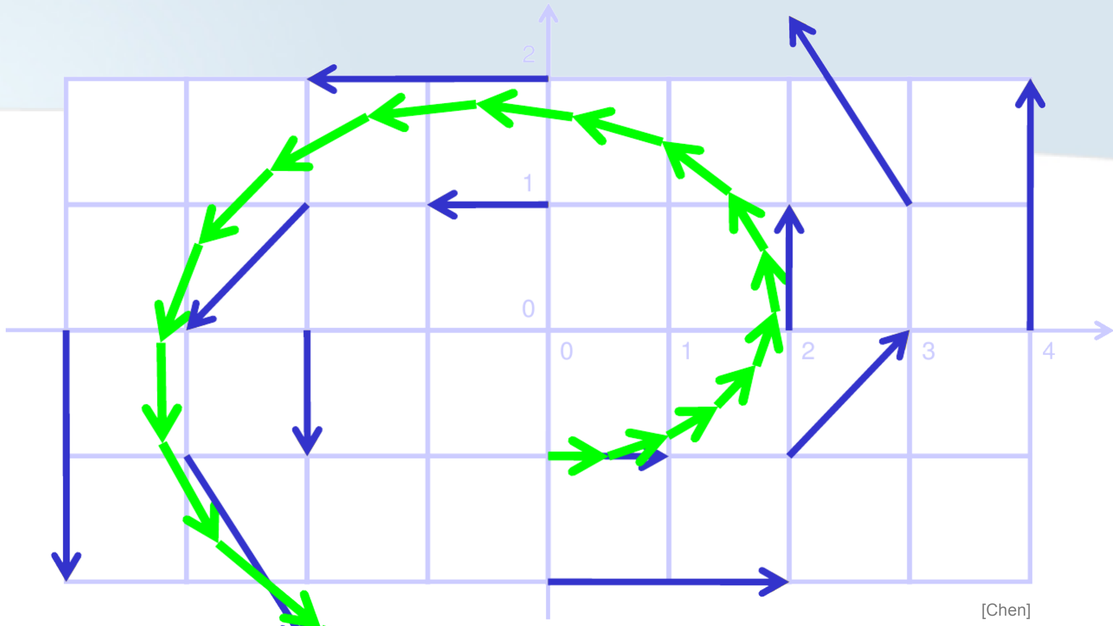
Streamlines on a computer



- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$
    - $u \leftarrow u + 1$
  - Until $u \leq q$
  - Does it look right to you?

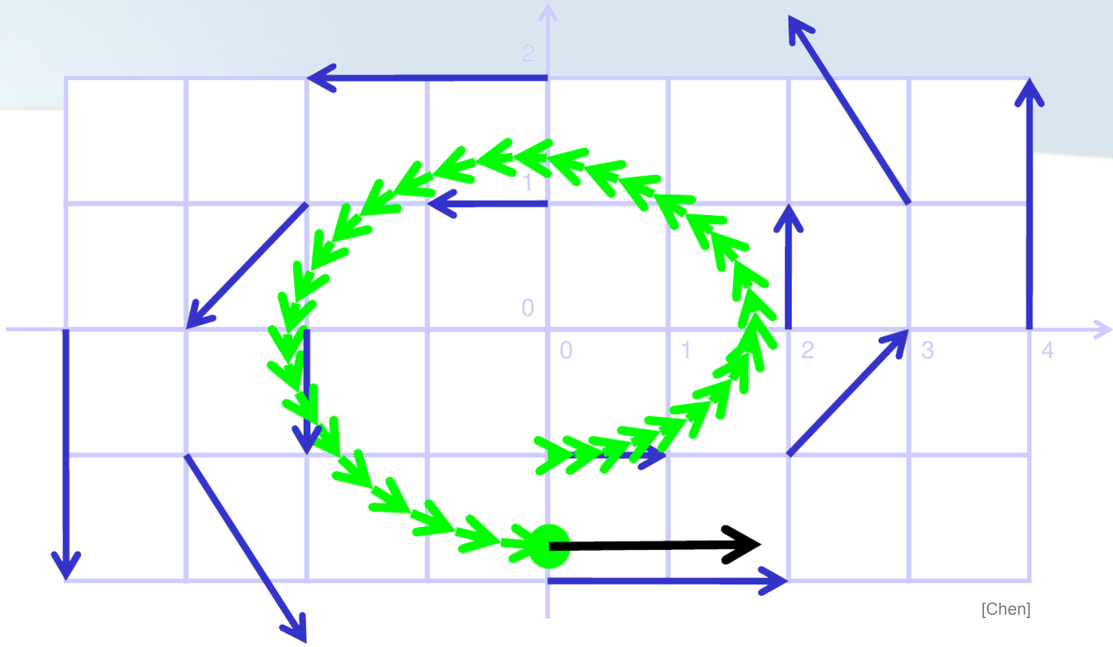
Streamlines on a computer

- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} \cdot \frac{c(p)}{q}$
    - $u \leftarrow u+1$
  - Until $u \leq q$
  - Does it look right to you?



[Chen]

Streamlines on a computer



- Euler integration algorithm
  - Given a seed point $c^{-1}(0)$
  - $u \leftarrow 0$
  - Repeat
    - Evaluate $\vec{f}(c^{-1}(u.c(p)/q)$
    - $c^{-1}((u+1).c(p)/q) \leftarrow c^{-1}(u.c(p)/q) + \frac{\vec{f}(c^{-1}(u.c(p)/q))}{||\vec{f}(c^{-1}(u.c(p)/q))||} . \frac{c(p)}{q}$
    - $u \leftarrow u + 1$
  - Until $u \leq q$
  - Does it look right to you?

Streamlines on a computer

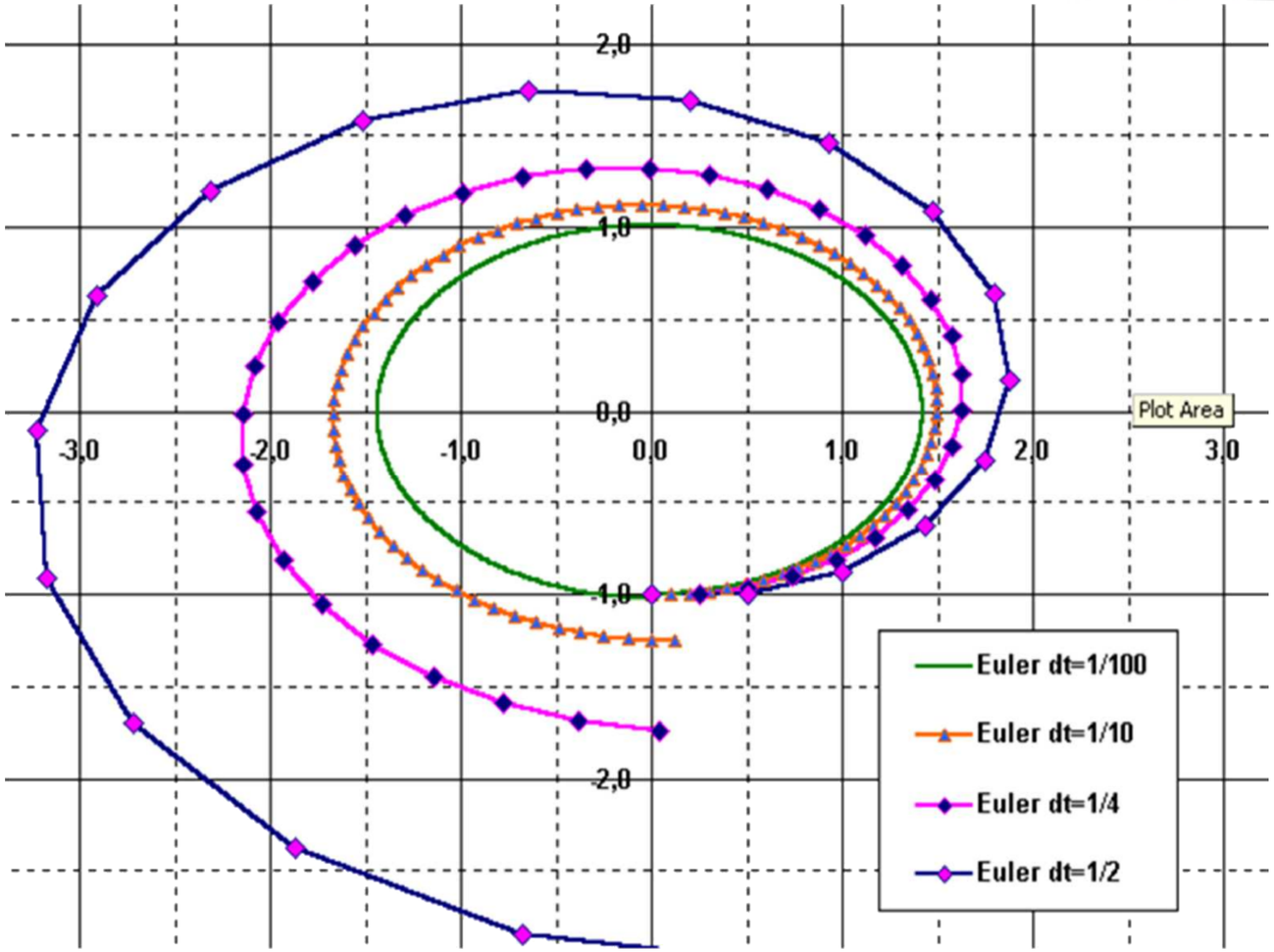
- Euler integration algorithm

Streamlines on a computer

- Euler integration algorithm
  - Fixed step size
  - Ratio of the magnitude

Streamlines on a computer

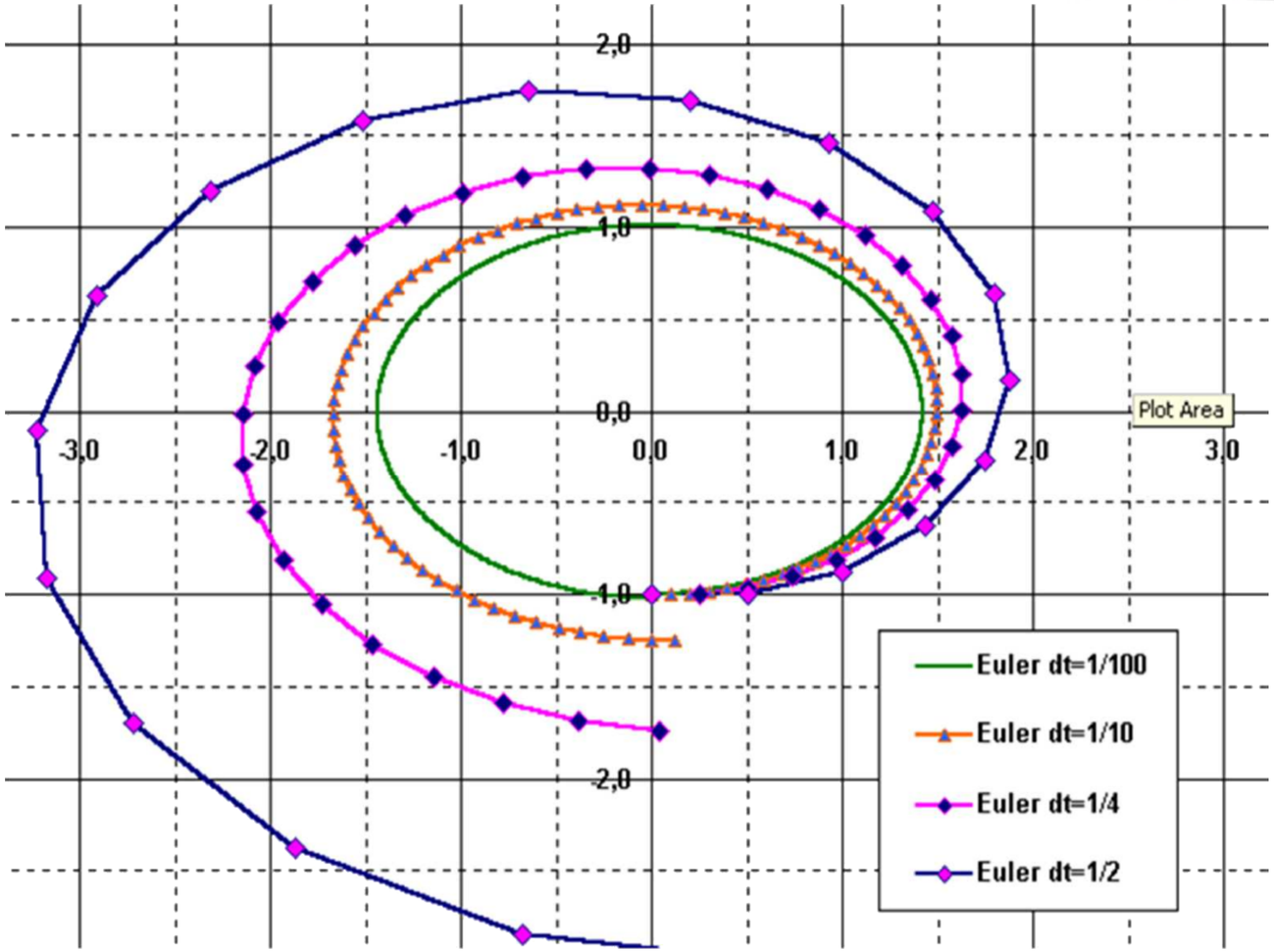
- Euler integration algorithm
  - Fixed step size
  - Ratio of the magnitude



[Chen]

Streamlines on a computer

- Euler integration algorithm
  - Fixed step size
  - Ratio of the magnitude
- Trade-off
  - Sampling
  - Approximation quality



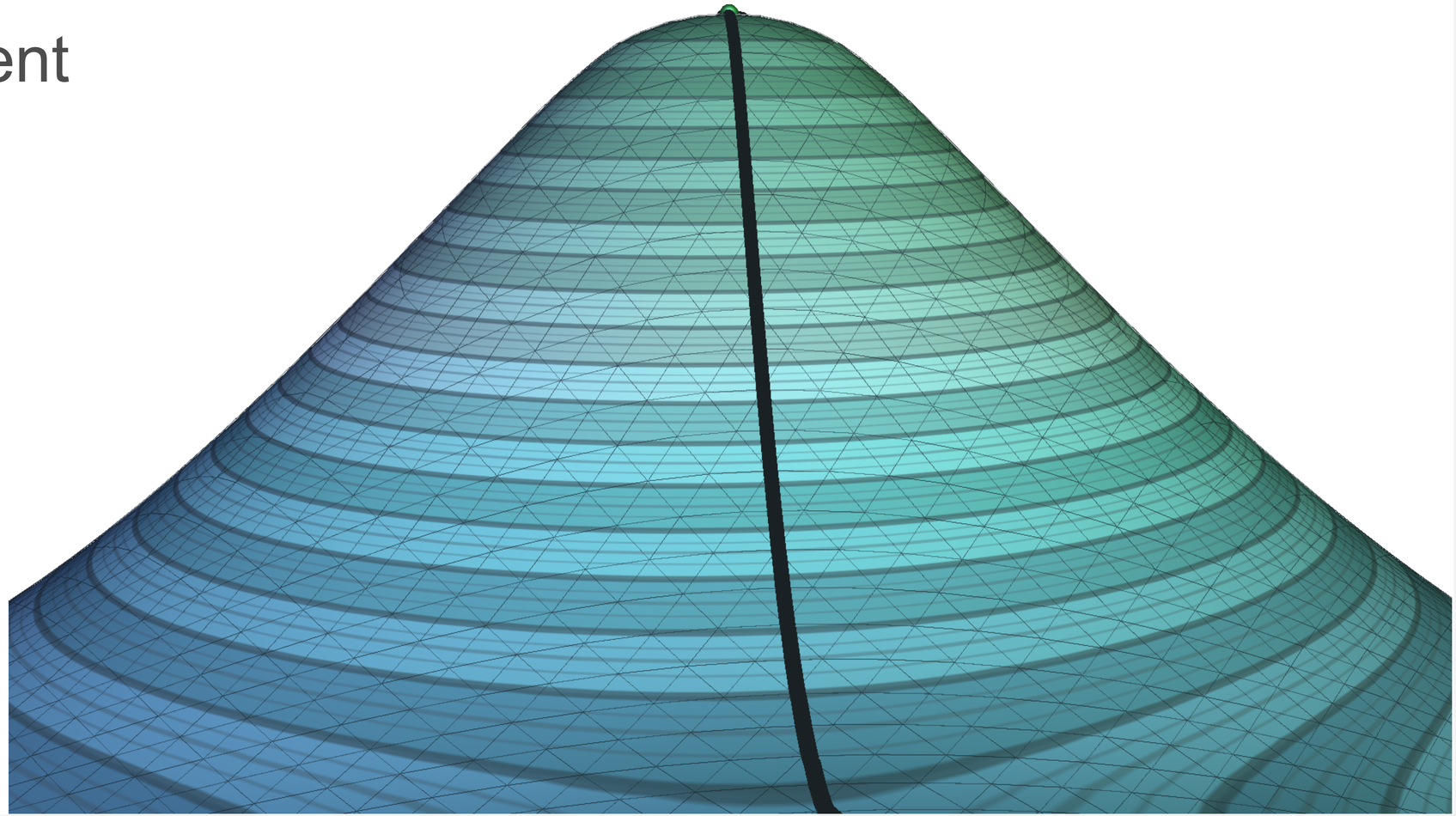
[Chen]

Special case

- Gradient fields on PL-manifolds

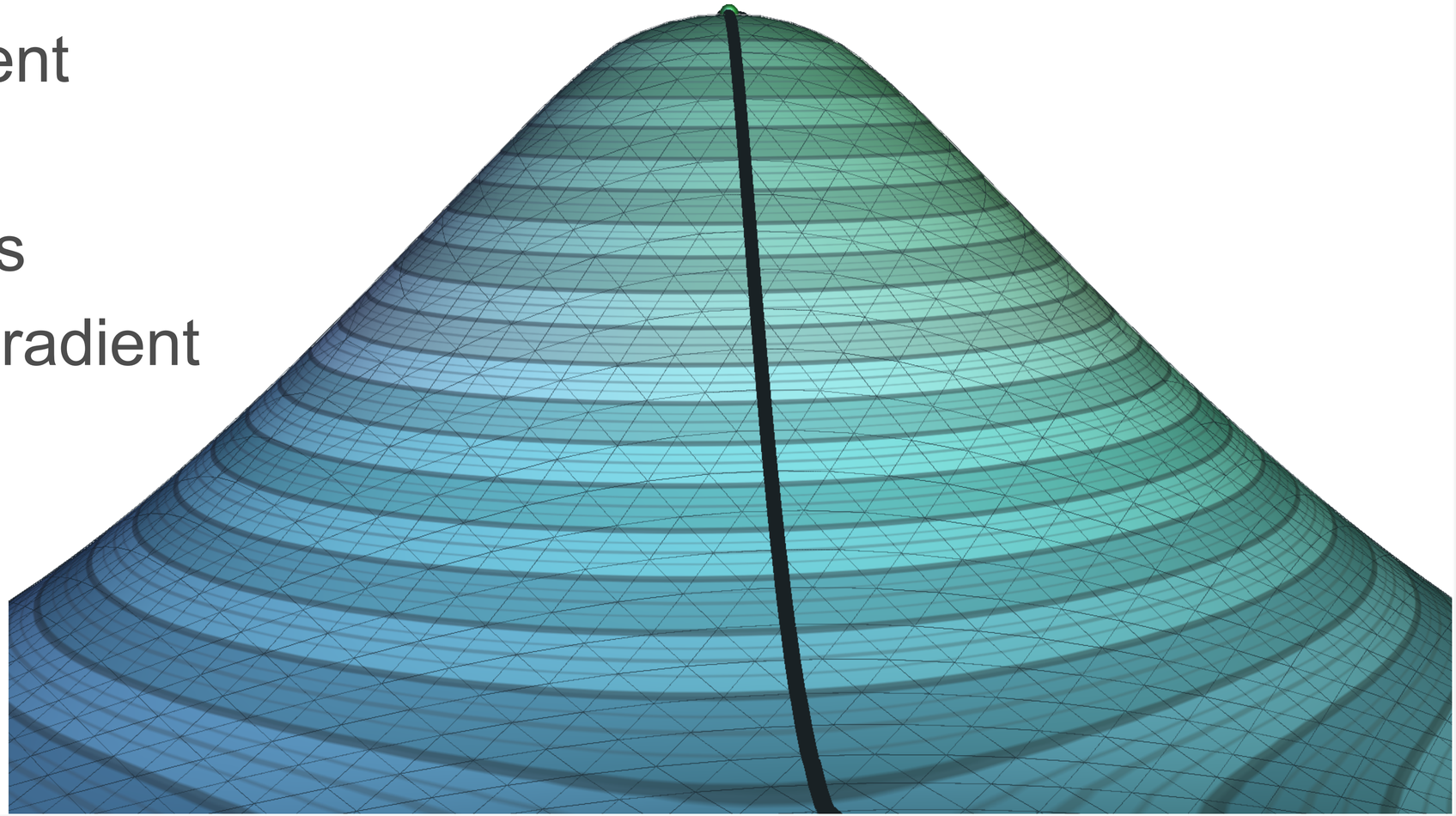
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent



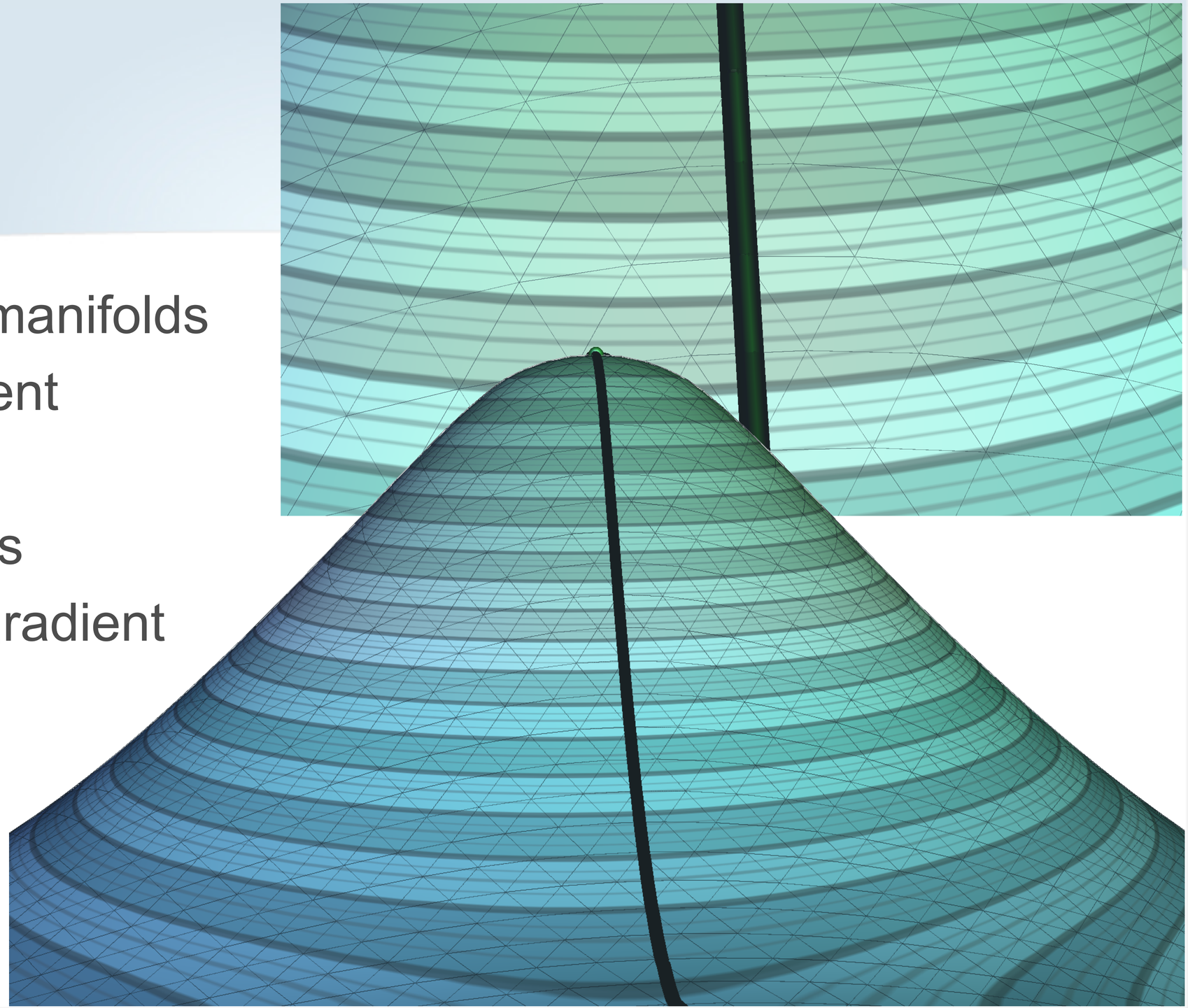
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent
- Barycentric coordinates
  - Piecewise constant gradient



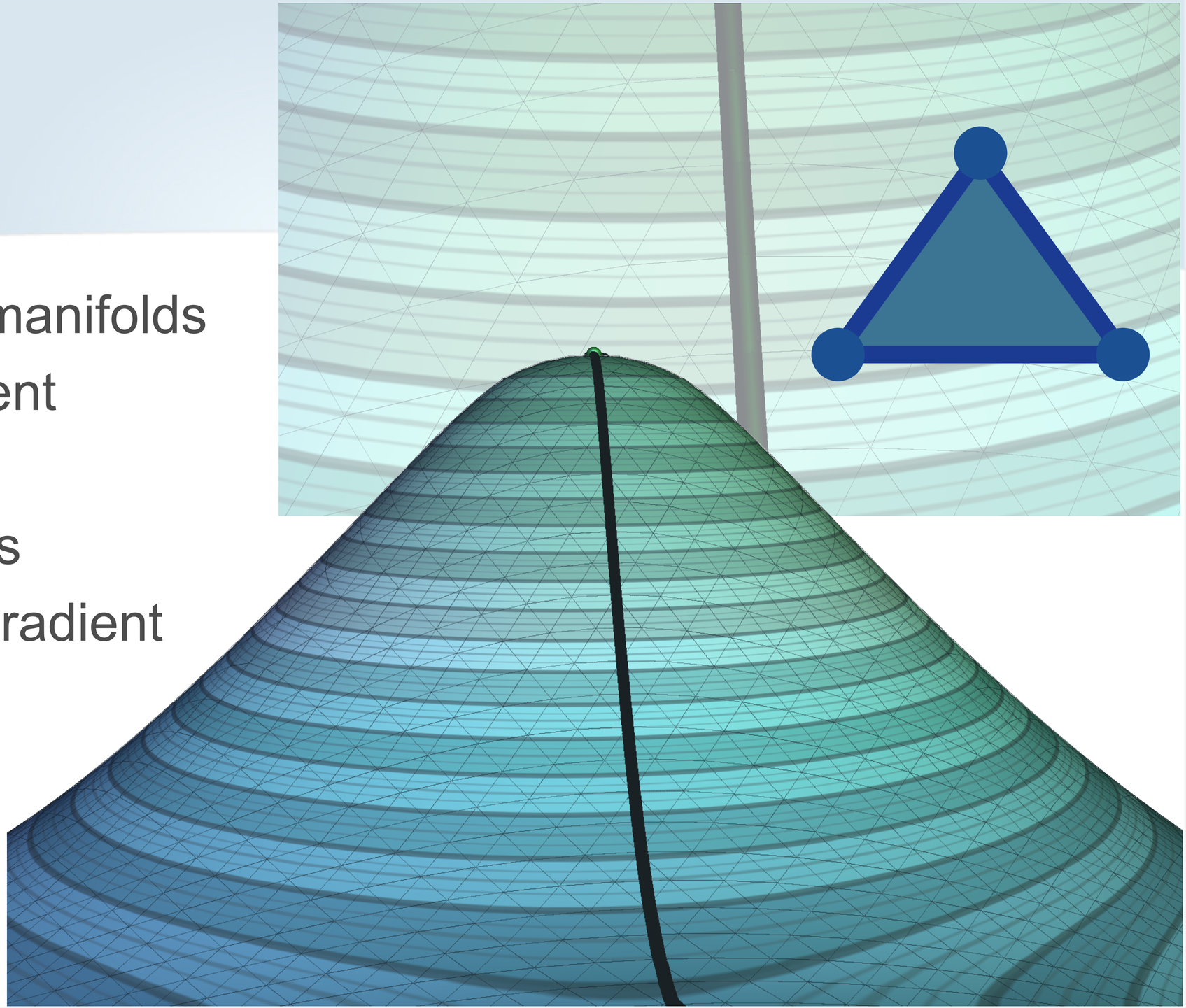
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent
- Barycentric coordinates
  - Piecewise constant gradient



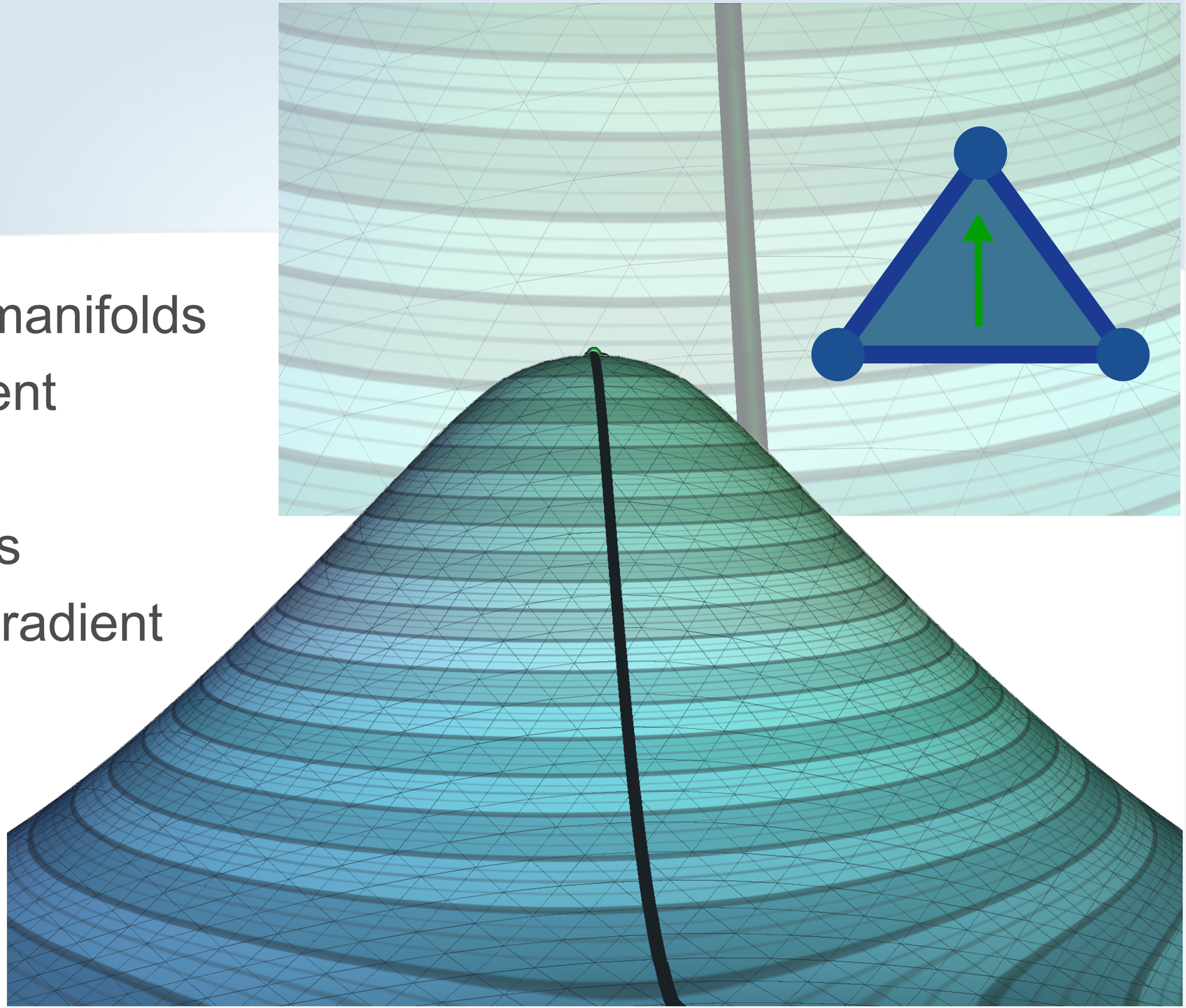
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent
- Barycentric coordinates
  - Piecewise constant gradient



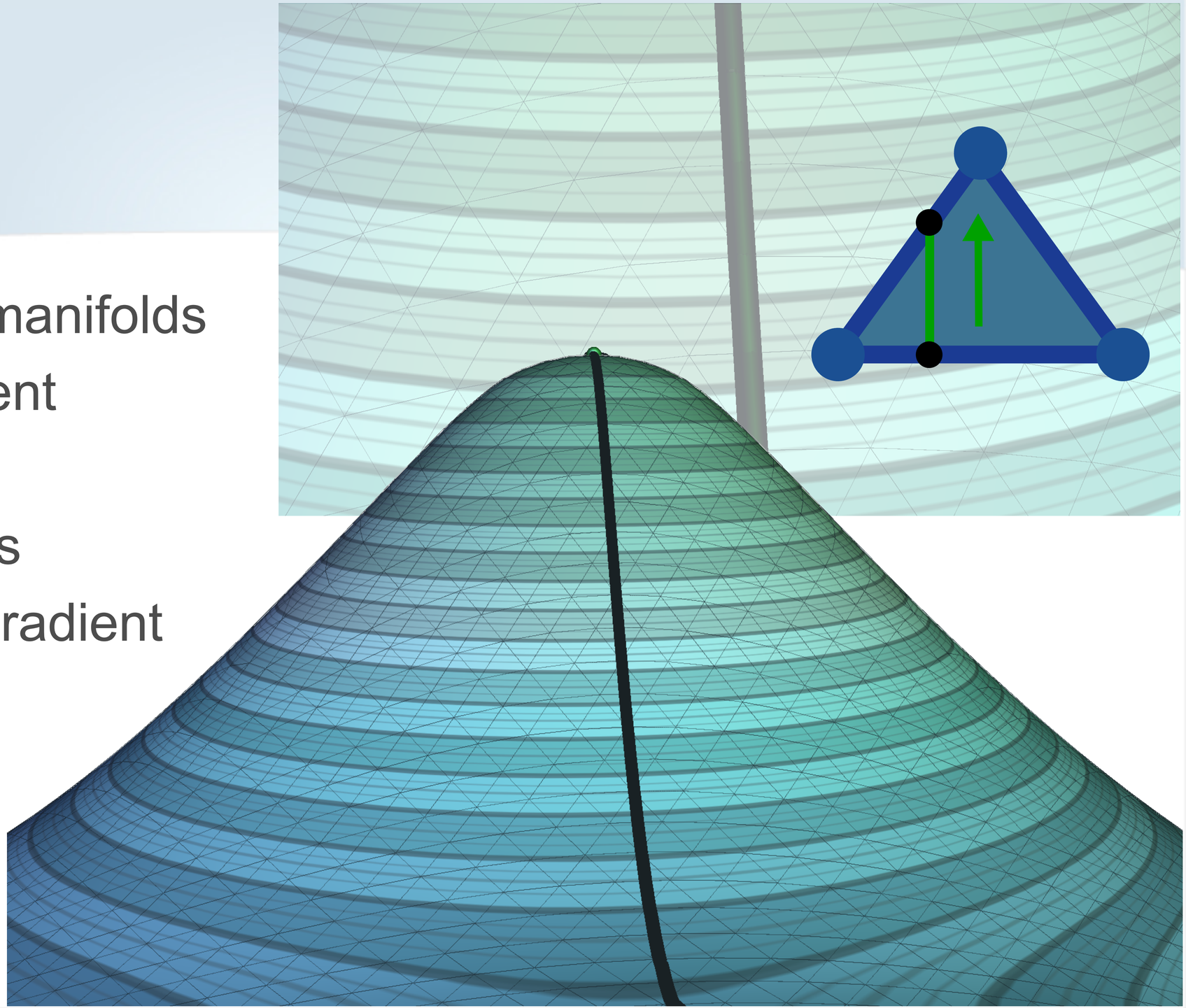
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent
- Barycentric coordinates
  - Piecewise constant gradient



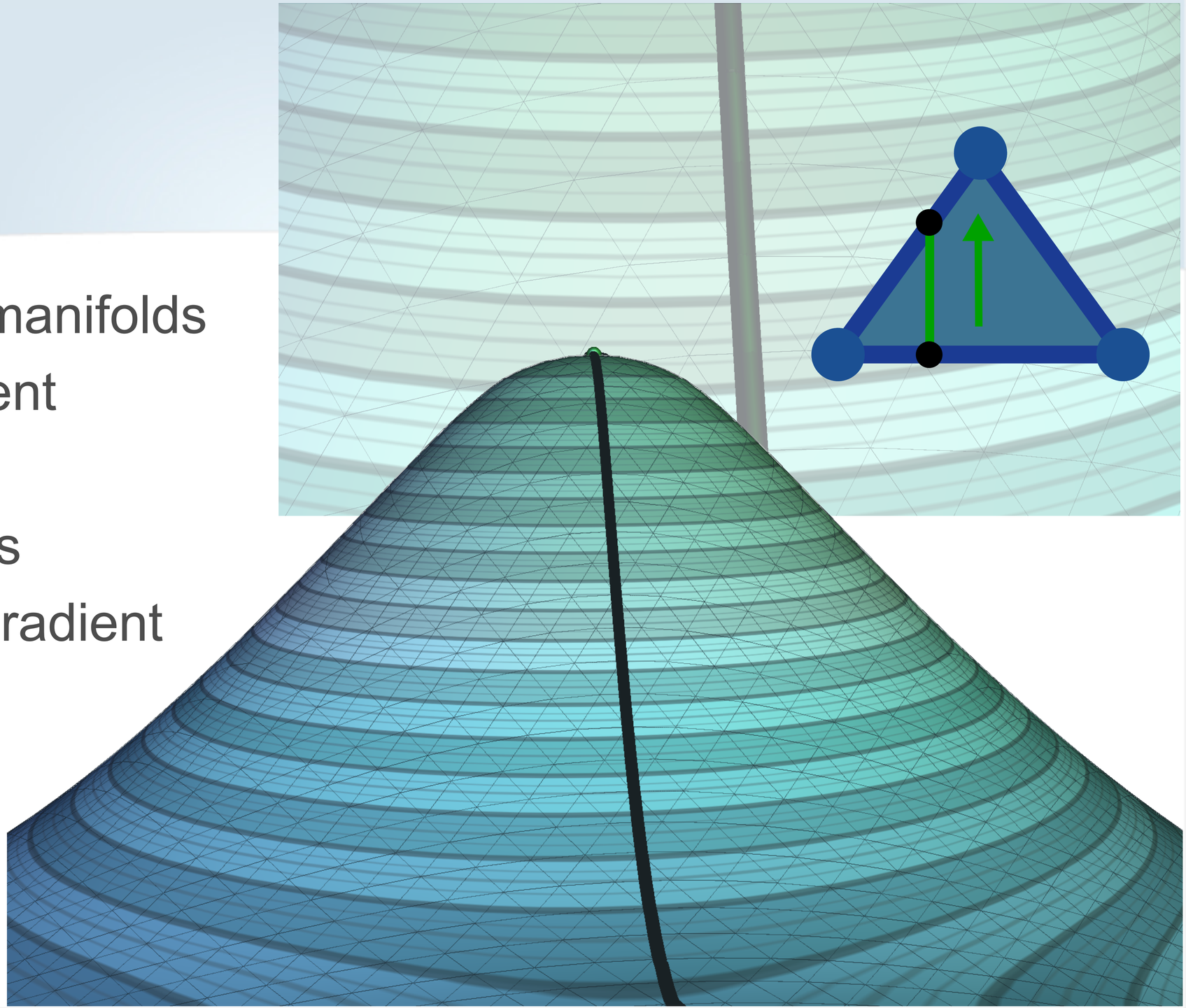
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent
- Barycentric coordinates
  - Piecewise constant gradient



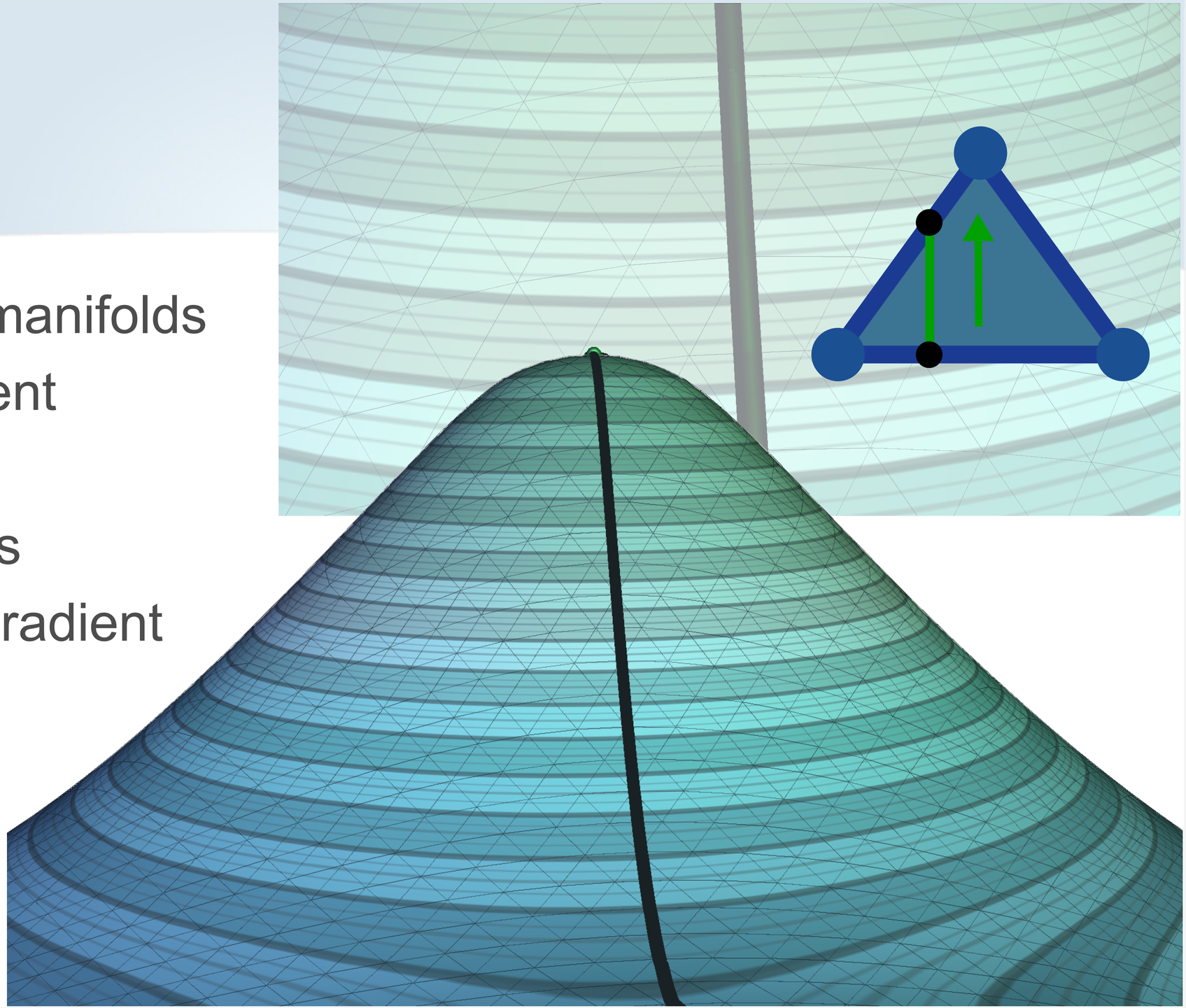
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent
- Barycentric coordinates
  - Piecewise constant gradient
  - No integration error
  - Nearly no ambiguity



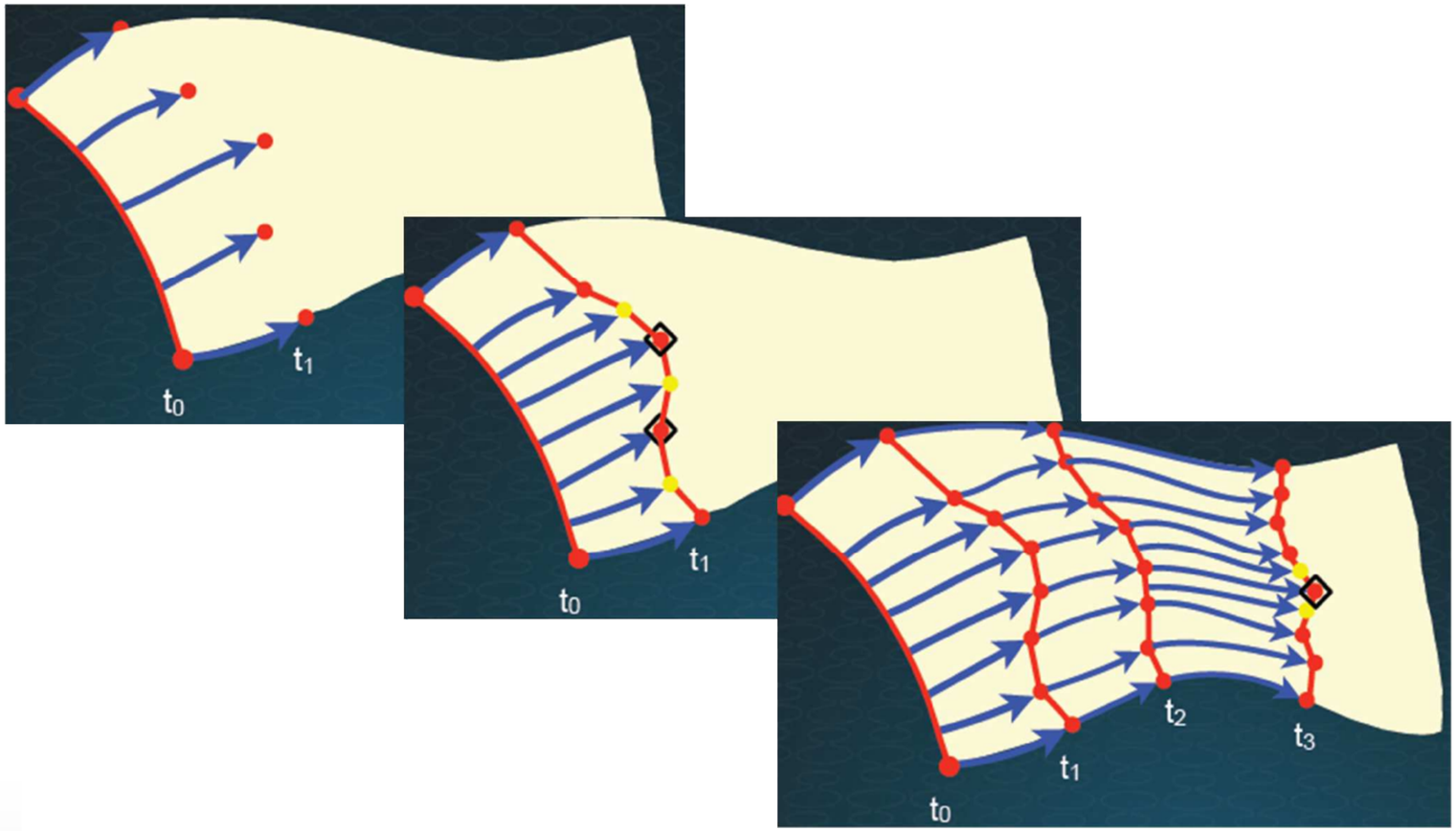
Special case

- Gradient fields on PL-manifolds
  - Path of steepest ascent
- Barycentric coordinates
  - Piecewise constant gradient
  - No integration error
  - Nearly no ambiguity
- Primal/dual meshes



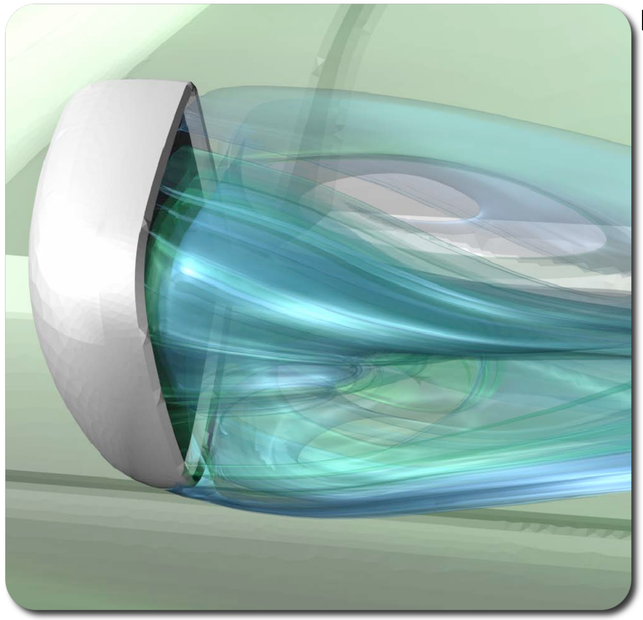
Streamsurfaces

- Adaptive sampling depending on the curvature of the seed curve

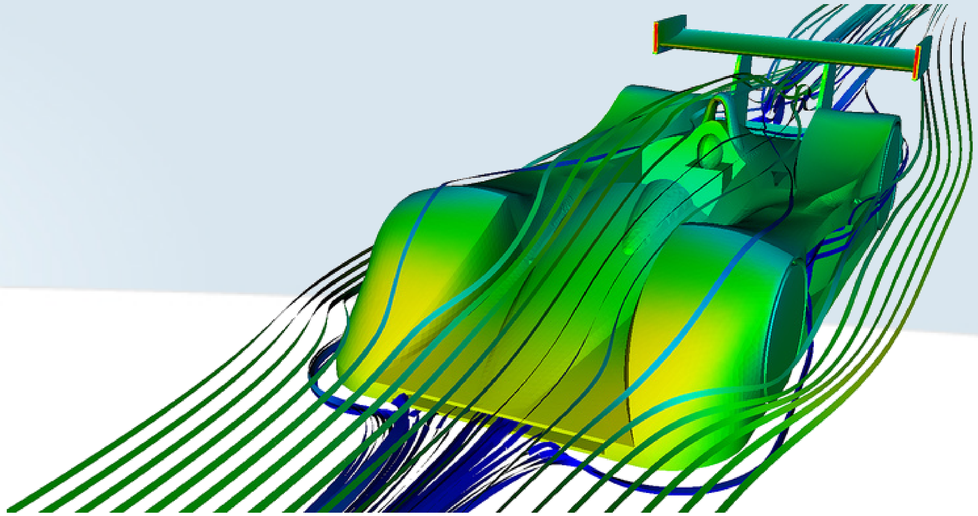


[Garth 09]

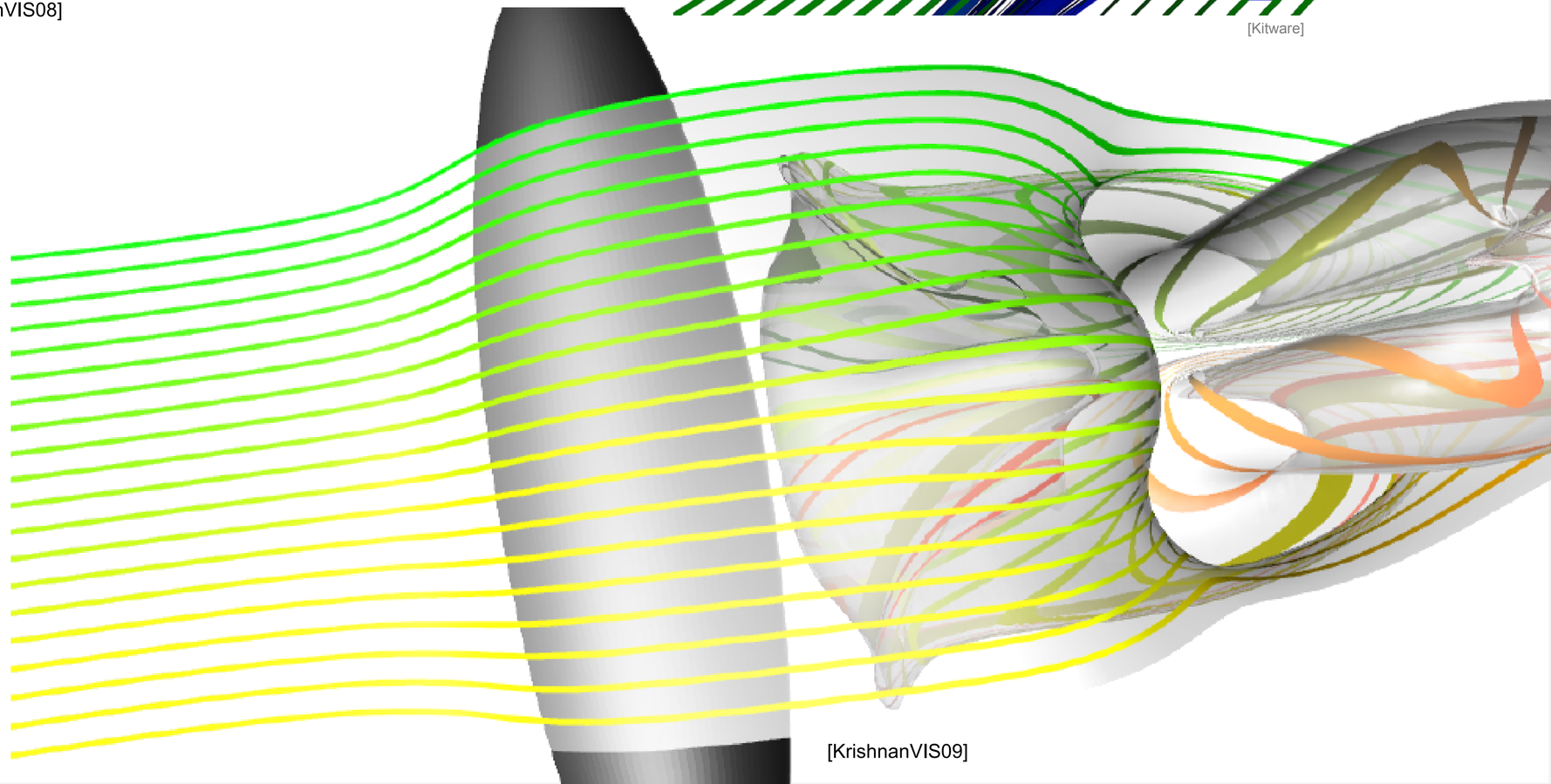
Examples



[GarthVIS08]



[Kitware]



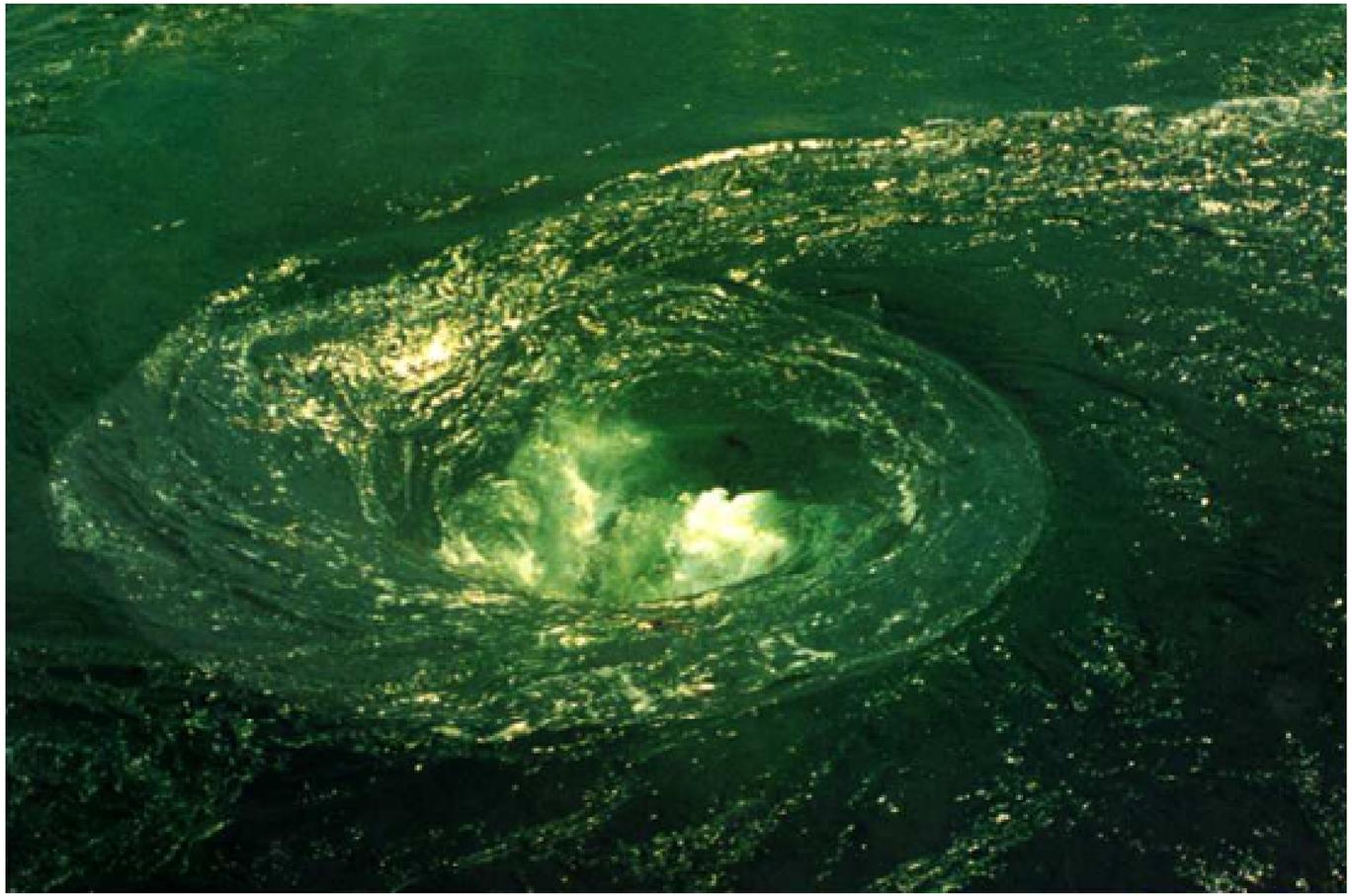
[KrishnanVIS09]

Steady vector fields

- What else is there to visualize?

Steady vector fields

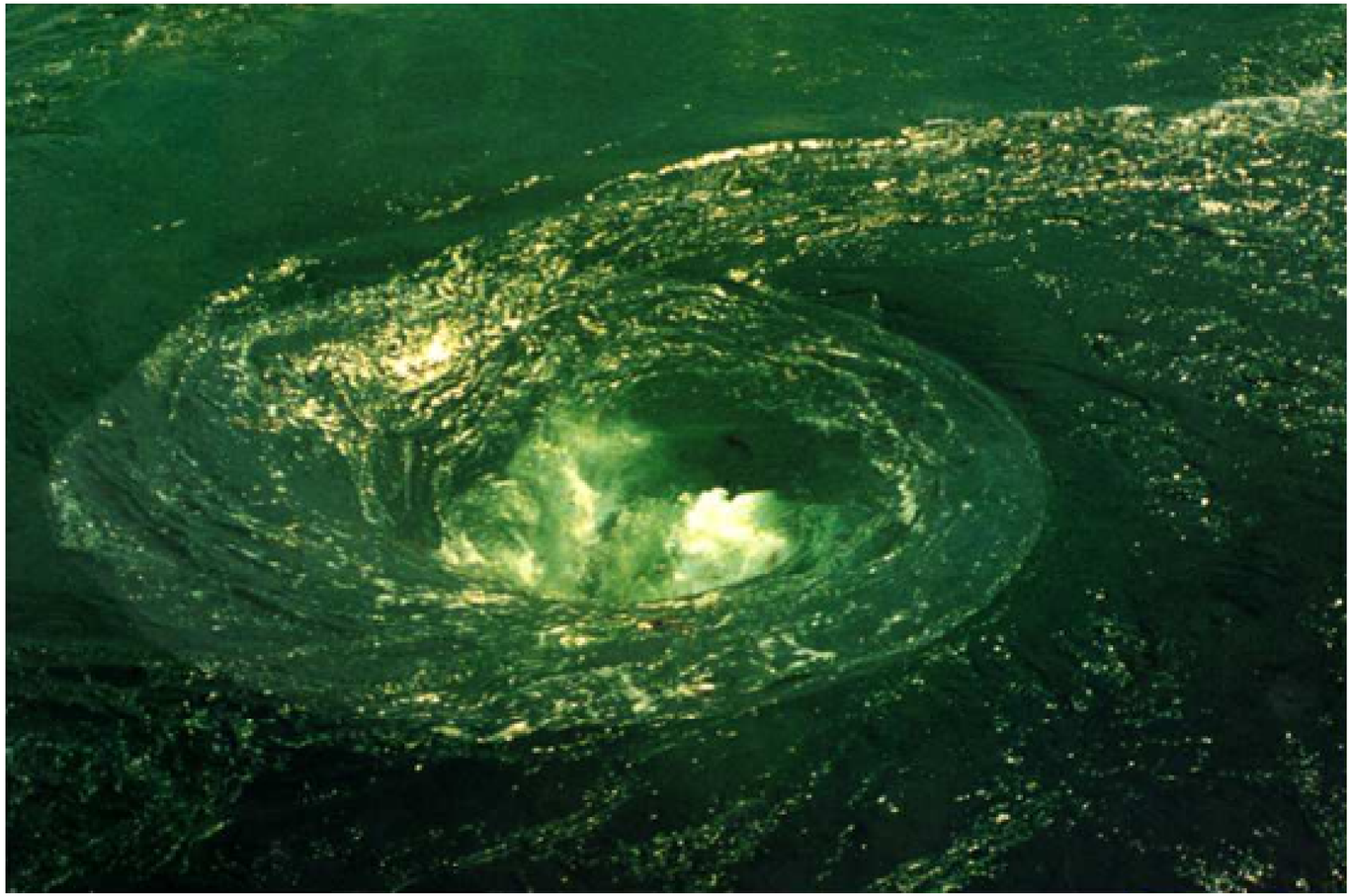
- What else is there to visualize?
  - Get inspiration from nature



[Post et al. 2003]

Steady vector fields

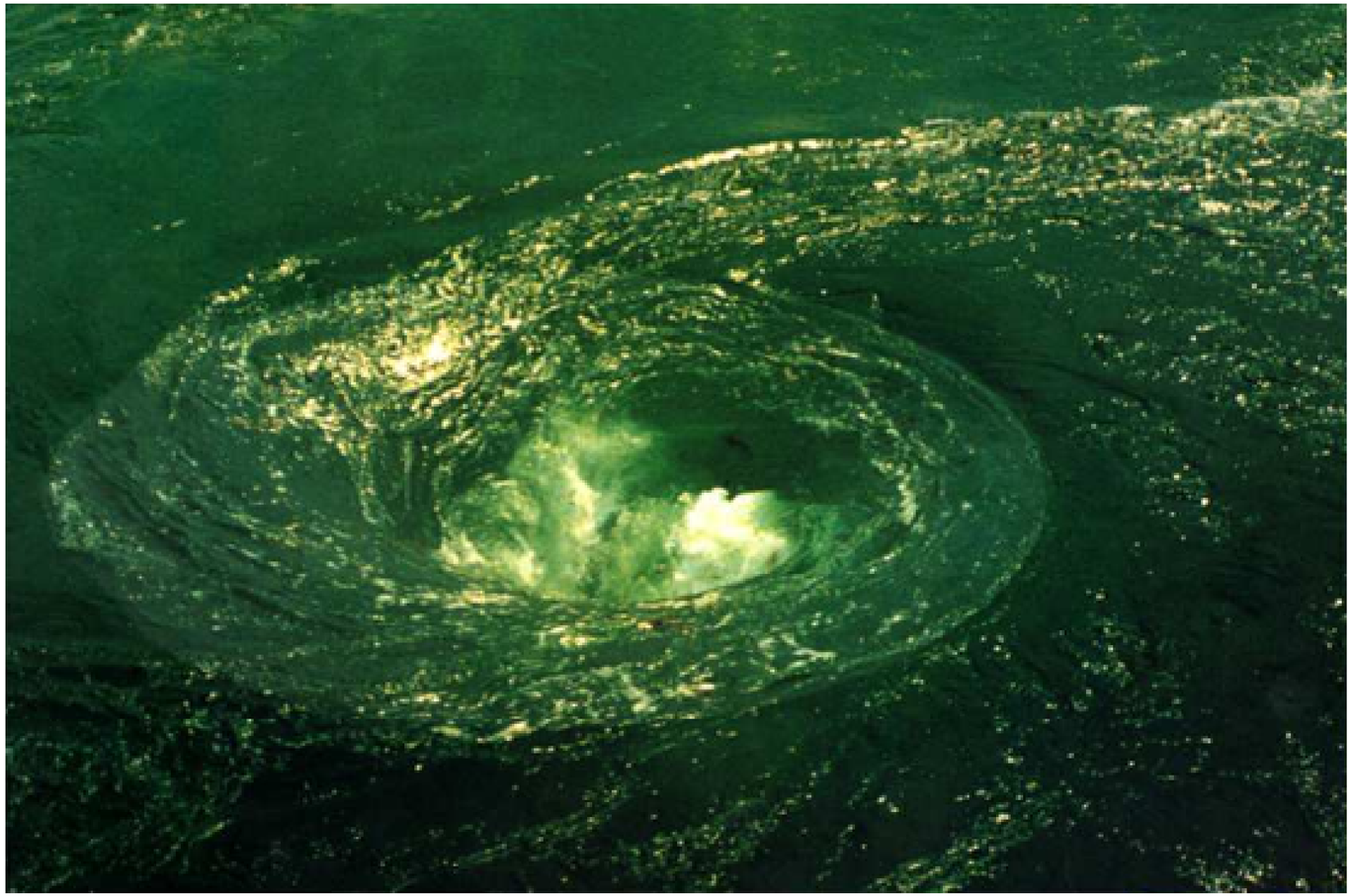
- What else is there to visualize?
  - Get inspiration from nature
  - Visualize the flow **globally**
  - For each point of the domain



[Post et al. 2003]

Steady vector fields

- What else is there to visualize?
  - Get inspiration from nature
  - Visualize the flow **globally**
  - For each point of the domain
  - Implicit visualization

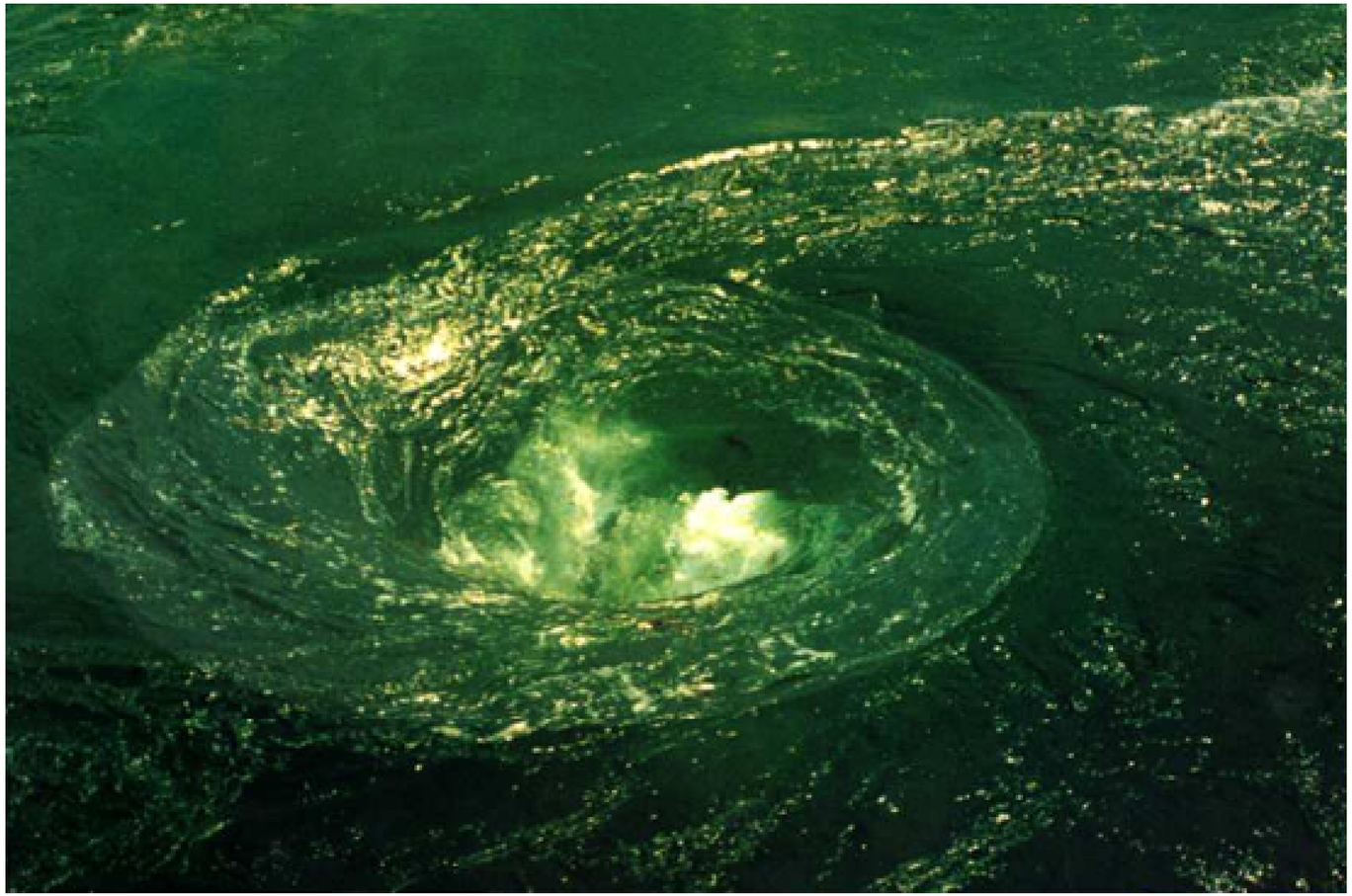


[Post et al. 2003]

Steady vector fields

- What else is there to visualize?
  - Get inspiration from nature
  - Visualize the flow **globally**
  - For each point of the domain
  - Implicit visualization

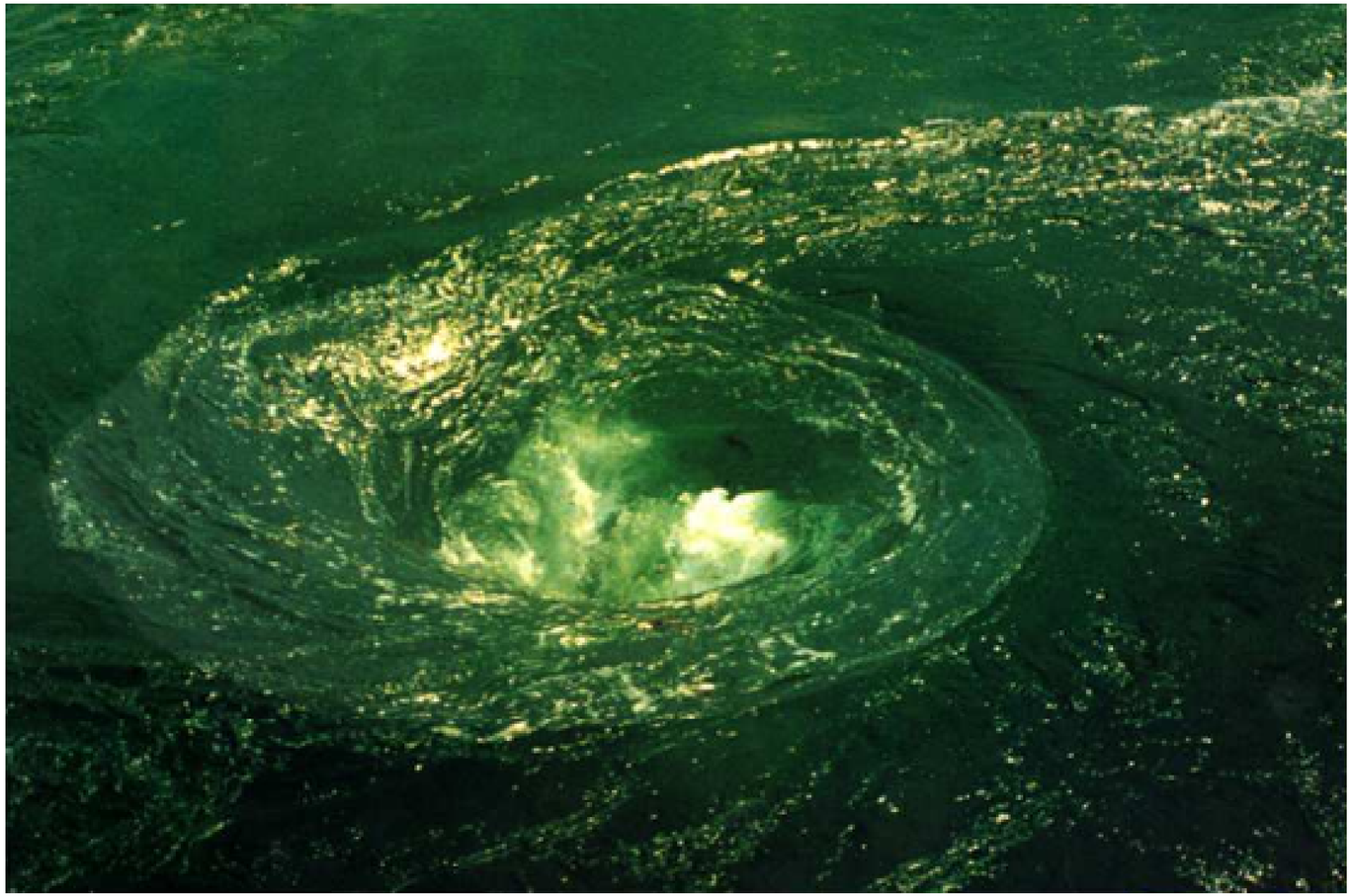
- Line Integral Convolution



[Post et al. 2003]

Line Integral Convolution

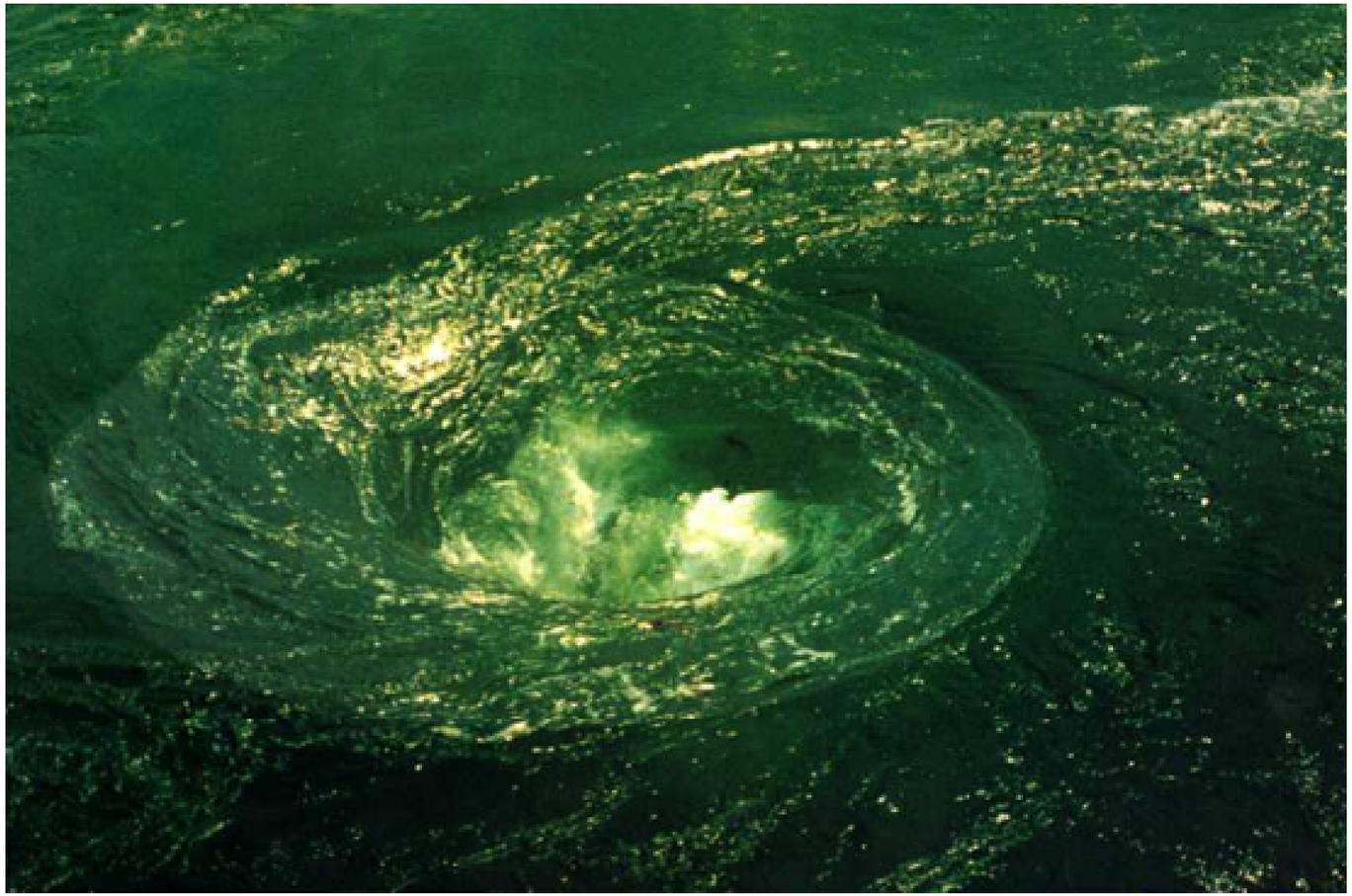
- Basic idea



[Post et al. 2003]

Line Integral Convolution

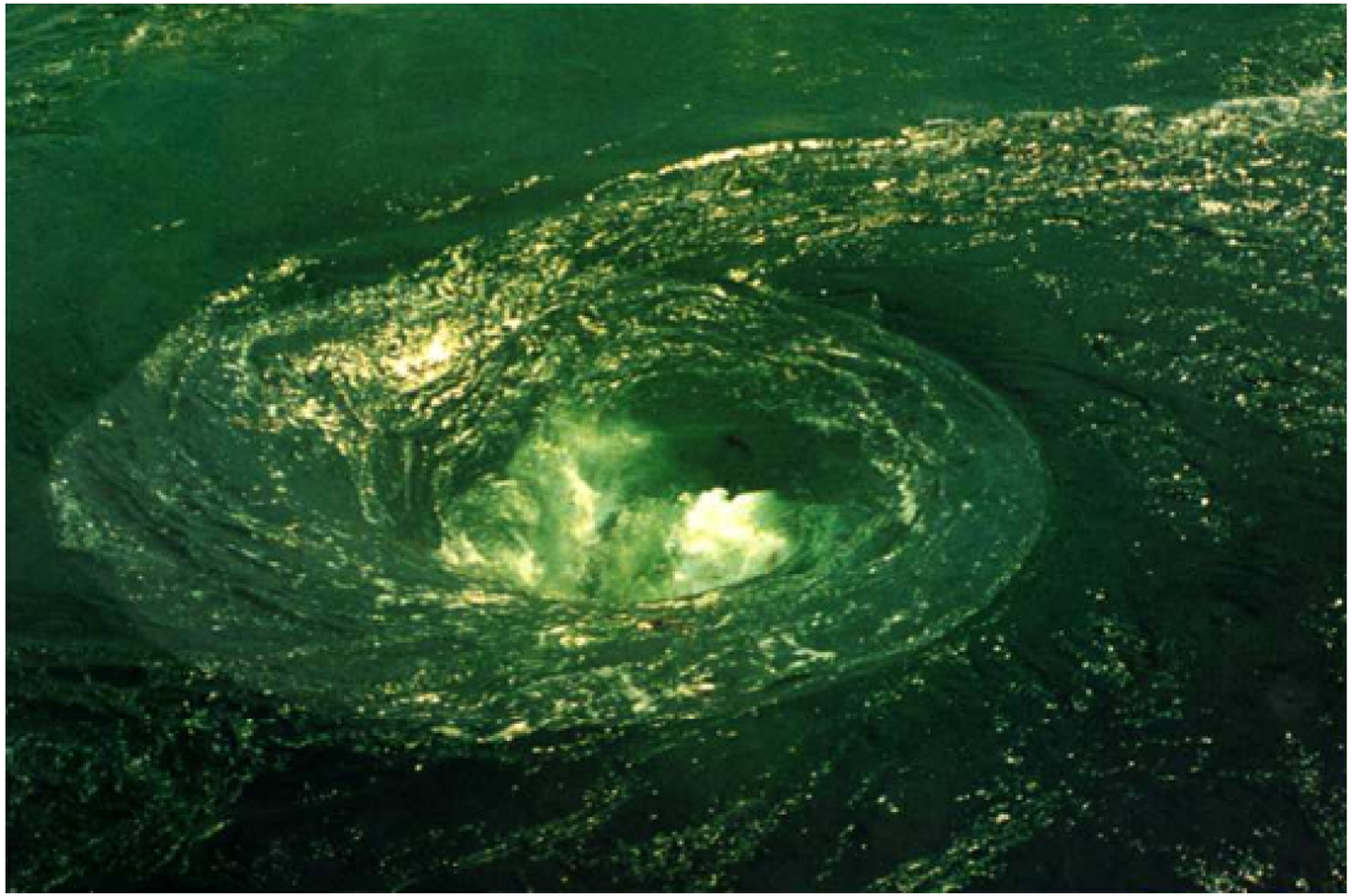
- Basic idea
  - Global visualization



[Post et al. 2003]

Line Integral Convolution

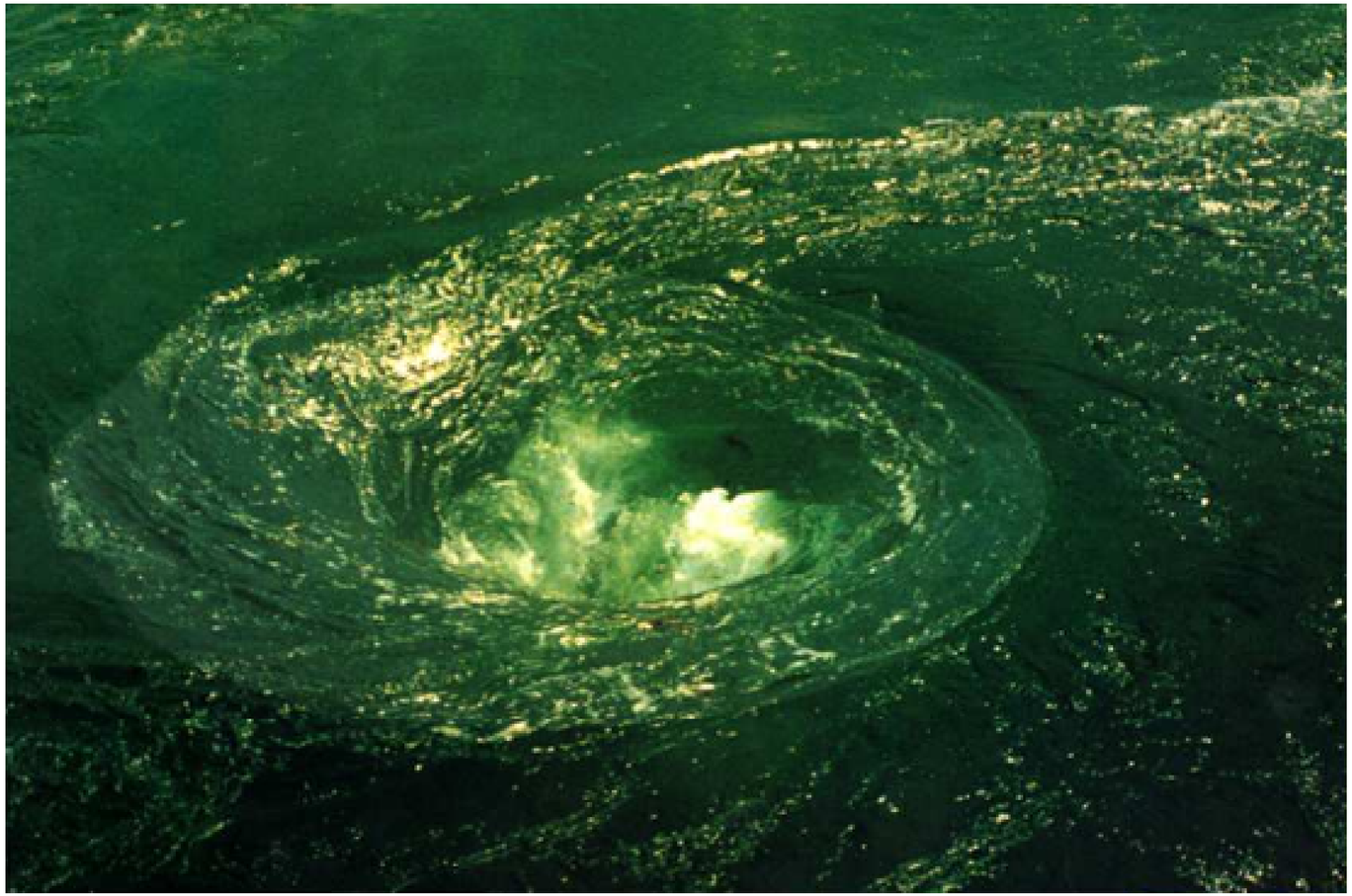
- Basic idea
  - Global visualization
    - Compute an integral curve for each point of the domain



[Post et al. 2003]

Line Integral Convolution

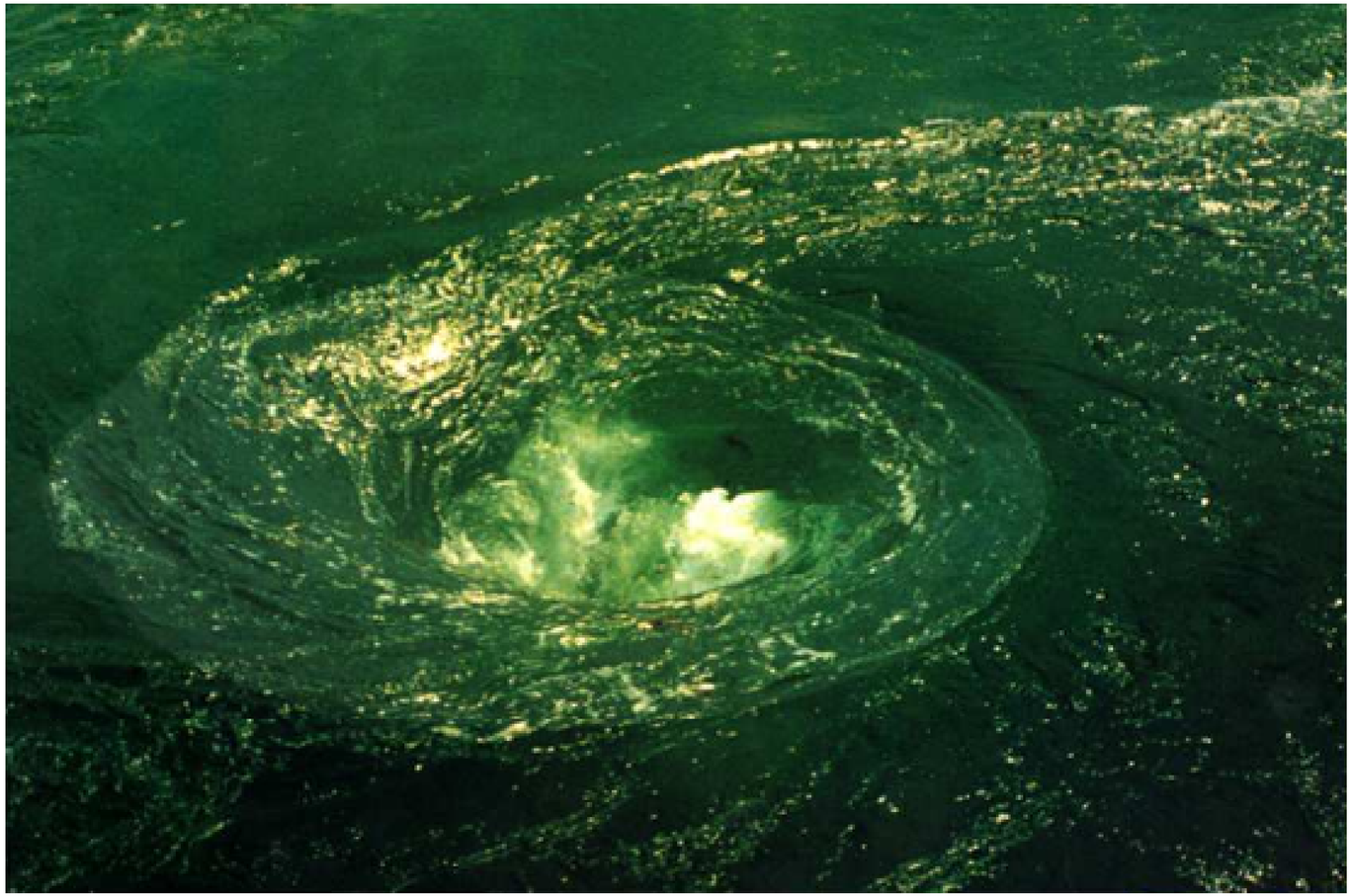
- Basic idea
  - Global visualization
    - Compute an integral curve for each point of the domain
  - Problem
    - Hard to see anything



[Post et al. 2003]

Line Integral Convolution

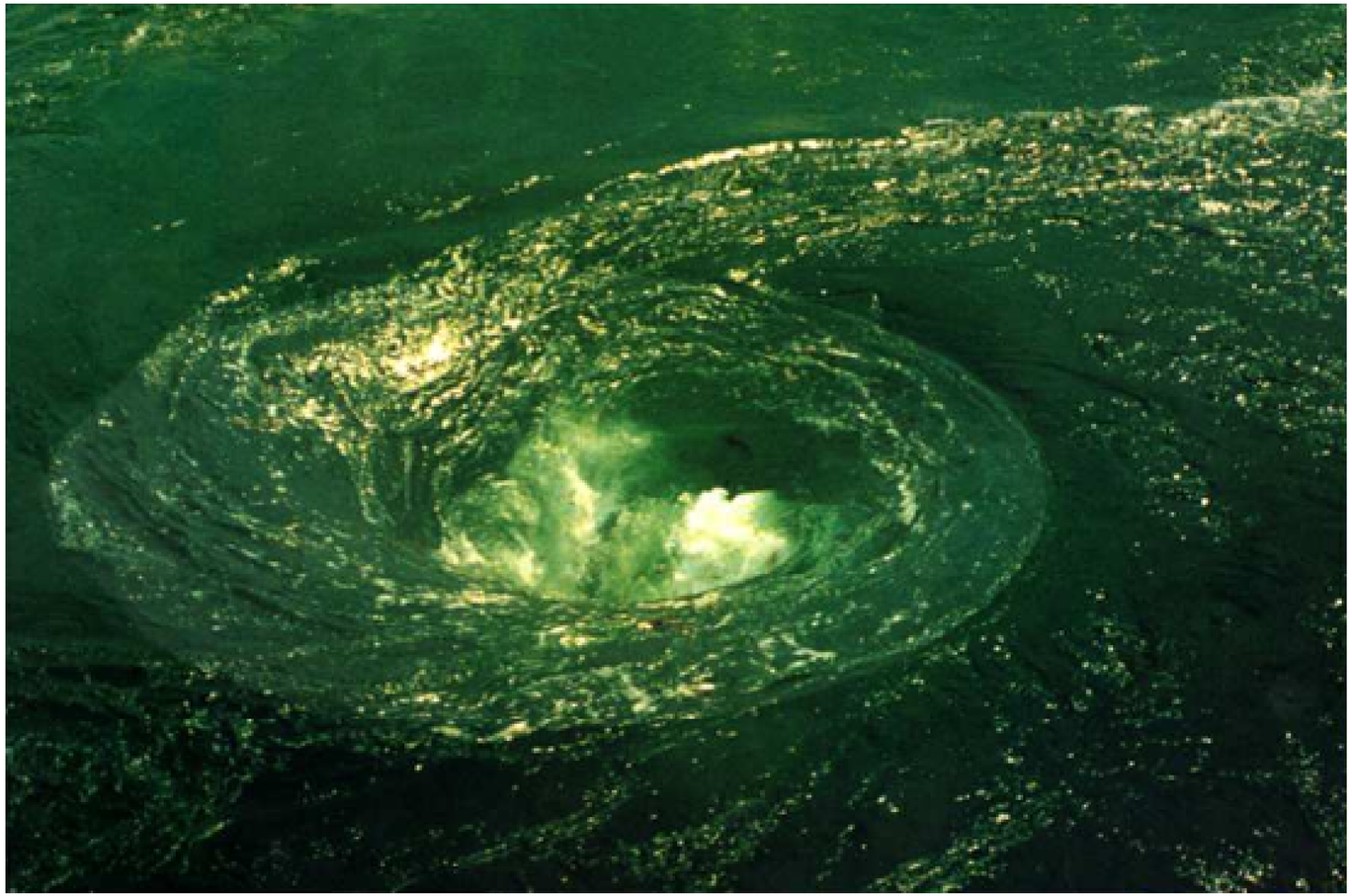
- Basic idea
  - Global visualization
    - Compute an integral curve for each point of the domain
  - Problem
    - Hard to see anything
  - Key idea
    - Mimic light variation (noise)



[Post et al. 2003]

Line Integral Convolution

- Basic idea
  - Global visualization
    - Compute an integral curve for each point of the domain
  - Problem
    - Hard to see anything
  - Key idea
    - Mimic light variation (noise)
    - Blend curves with noise



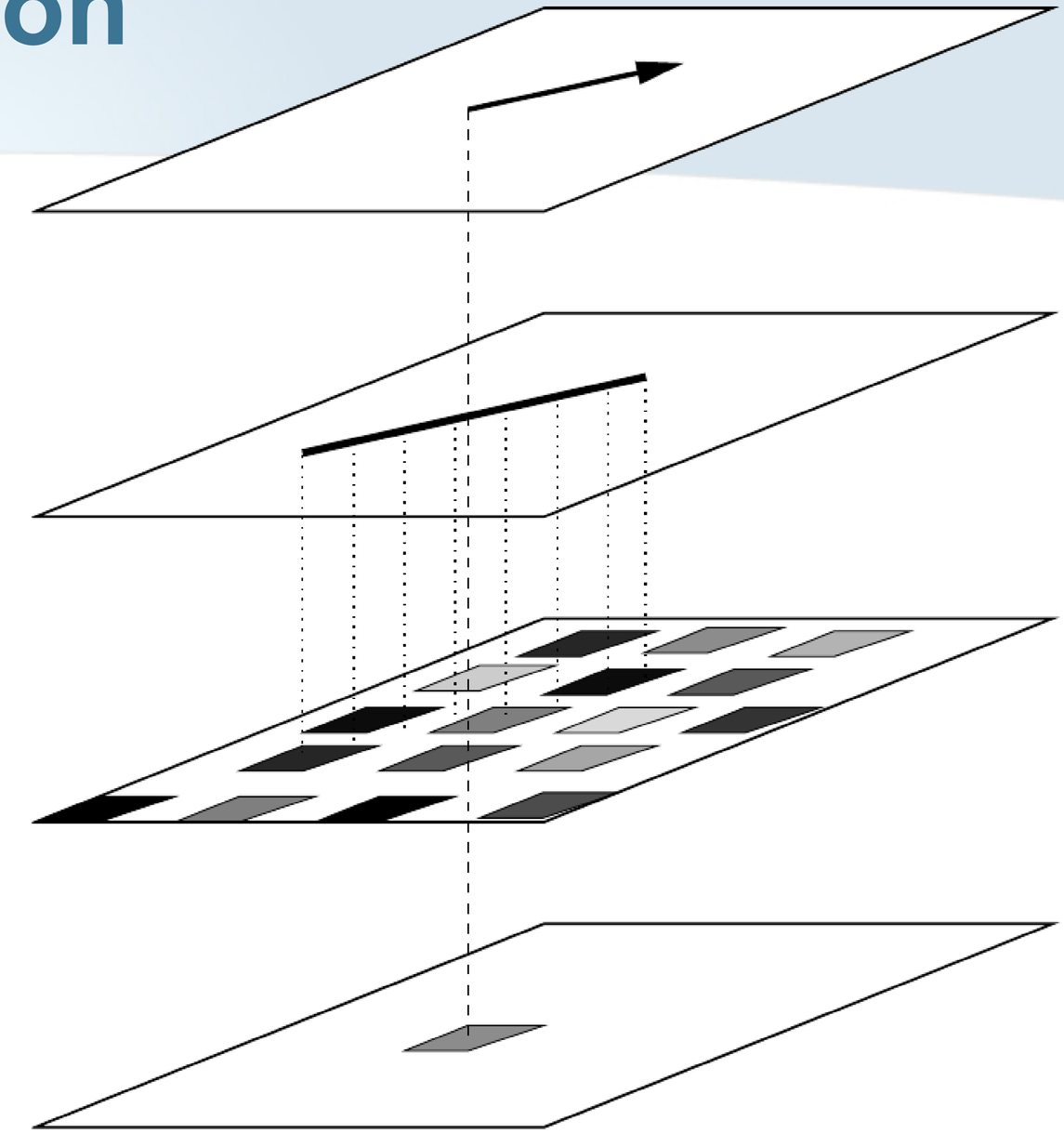
[Post et al. 2003]

Line Integral Convolution

- Algorithm

Line Integral Convolution

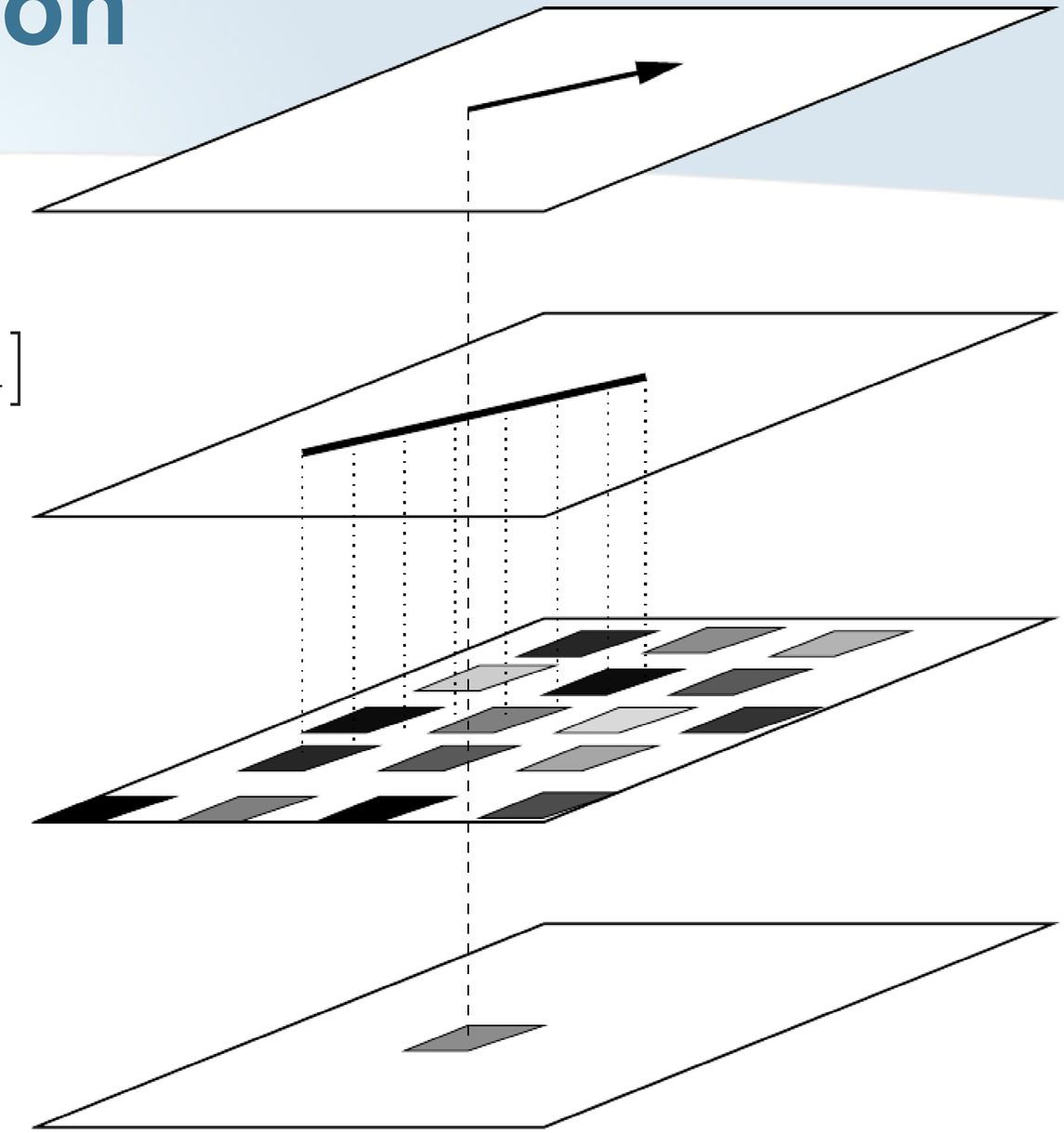
- Algorithm



[Helgeland]

Line Integral Convolution

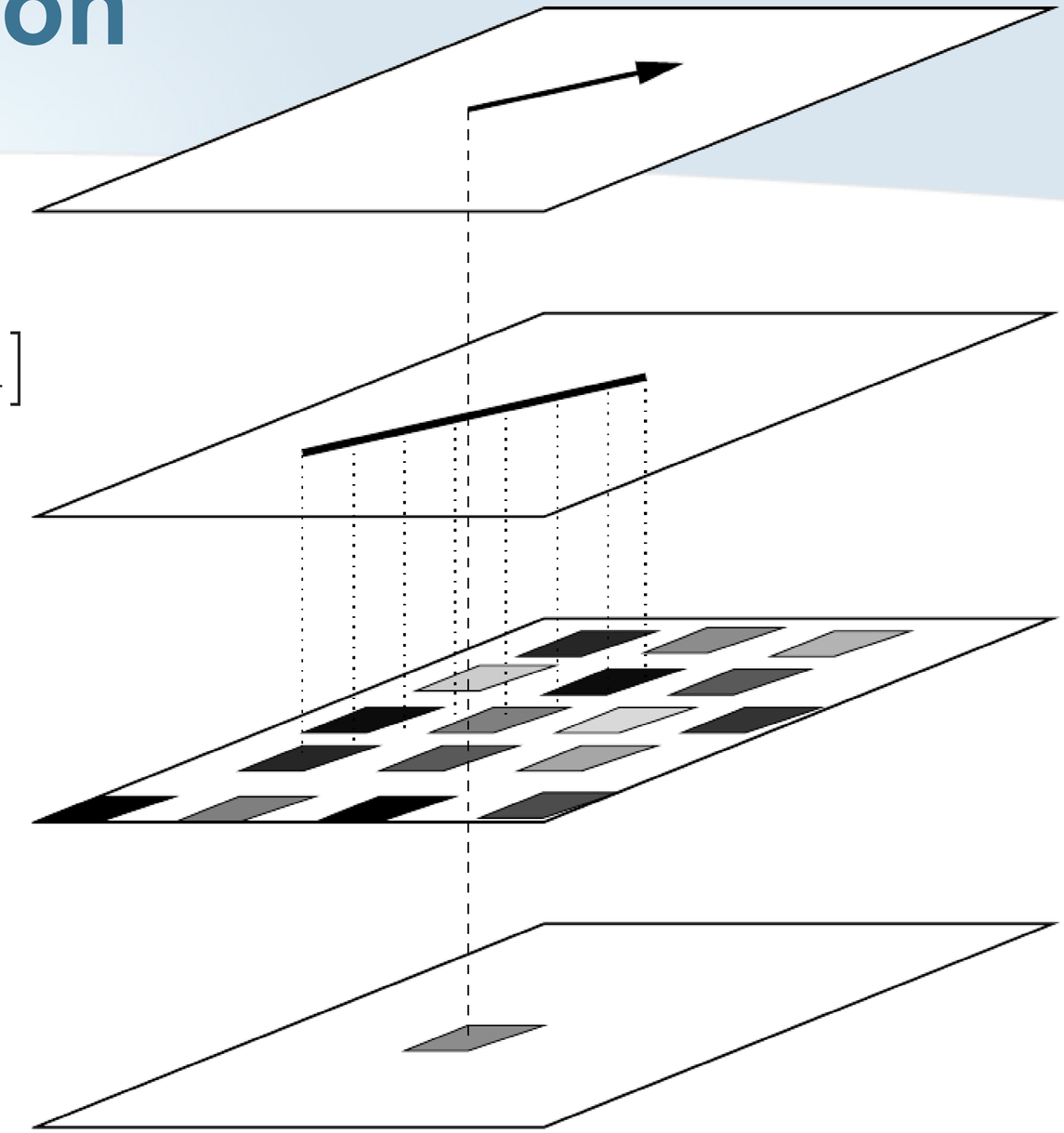
- Algorithm
  - Compute a noise field $\mathcal{N} : \mathcal{D} \rightarrow [0, 1]$



[Helgeland]

Line Integral Convolution

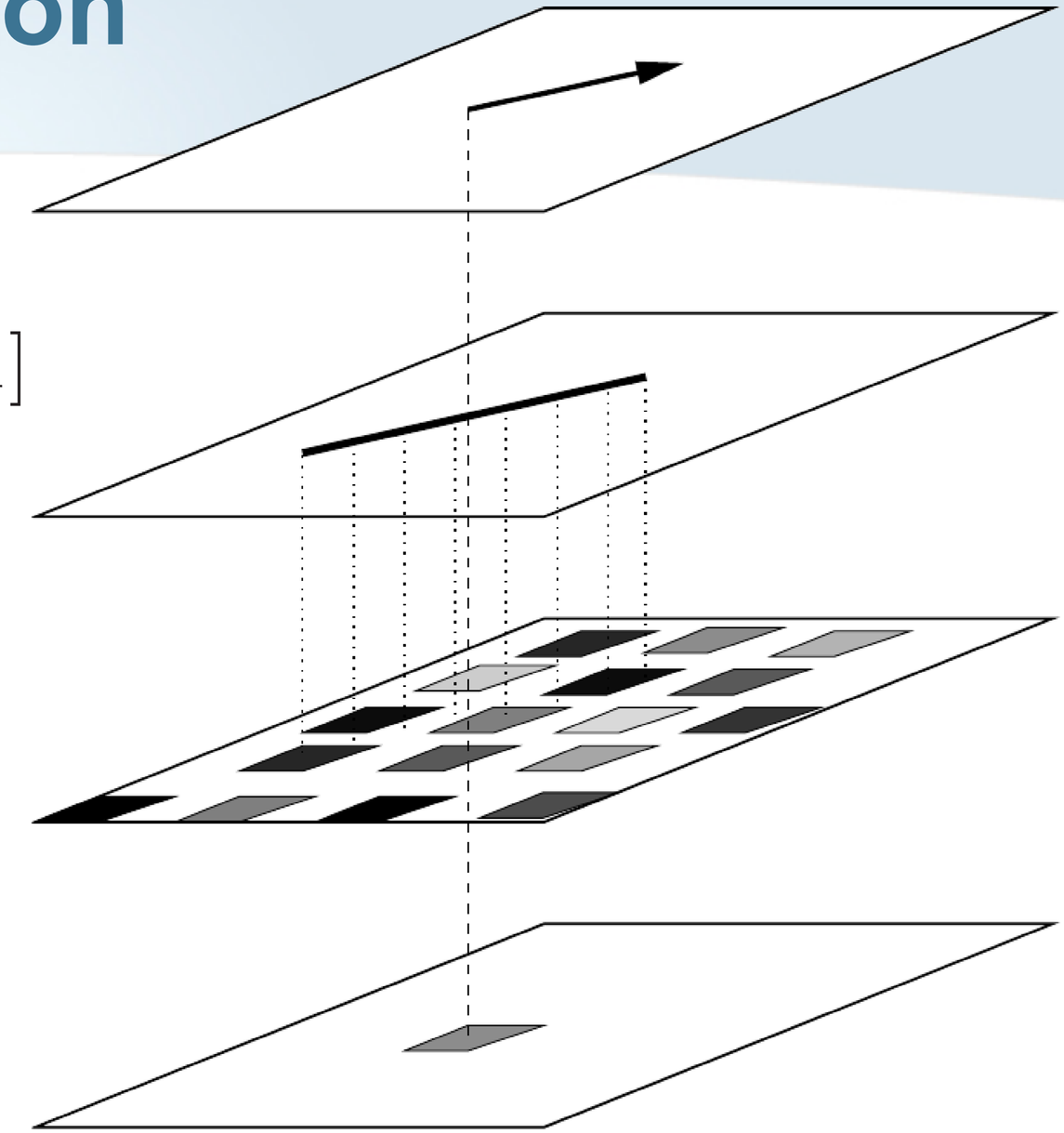
- Algorithm
  - Compute a noise field $\mathcal{N} : \mathcal{D} \rightarrow [0, 1]$
    - Random intensity for each vertex



[Helgeland]

Line Integral Convolution

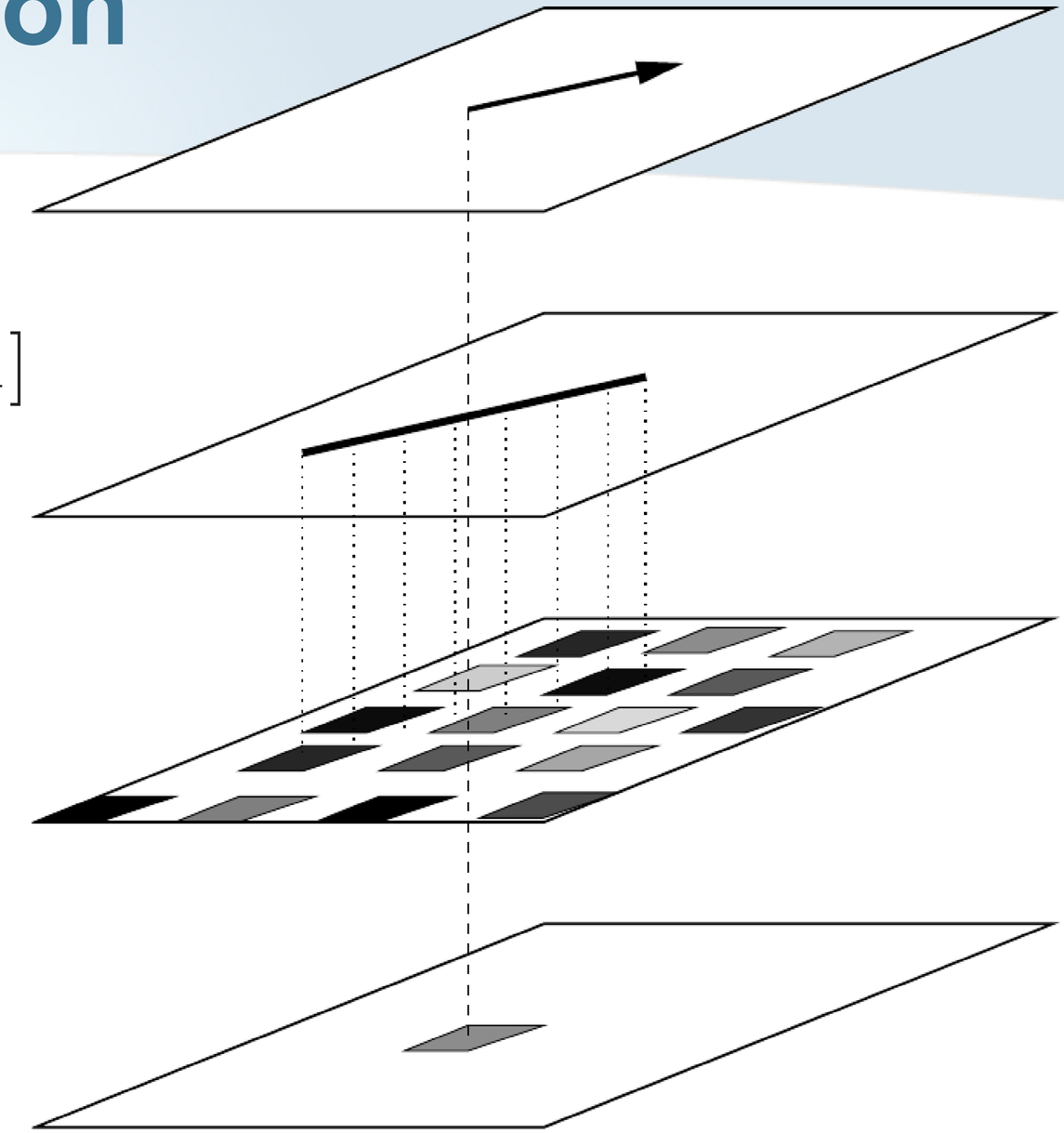
- Algorithm
  - Compute a noise field $\mathcal{N} : \mathcal{D} \rightarrow [0, 1]$
    - Random intensity for each vertex
  - For each vertex of the domain
    - Compute its integral curve $\mathcal{C}$
      - Backwards and forwards



[Helgeland]

Line Integral Convolution

- Algorithm
  - Compute a noise field $\mathcal{N} : \mathcal{D} \rightarrow [0, 1]$
    - Random intensity for each vertex
  - For each vertex of the domain
    - Compute its integral curve $\mathcal{C}$
      – Backwards and forwards
    - Convolution with the noise field

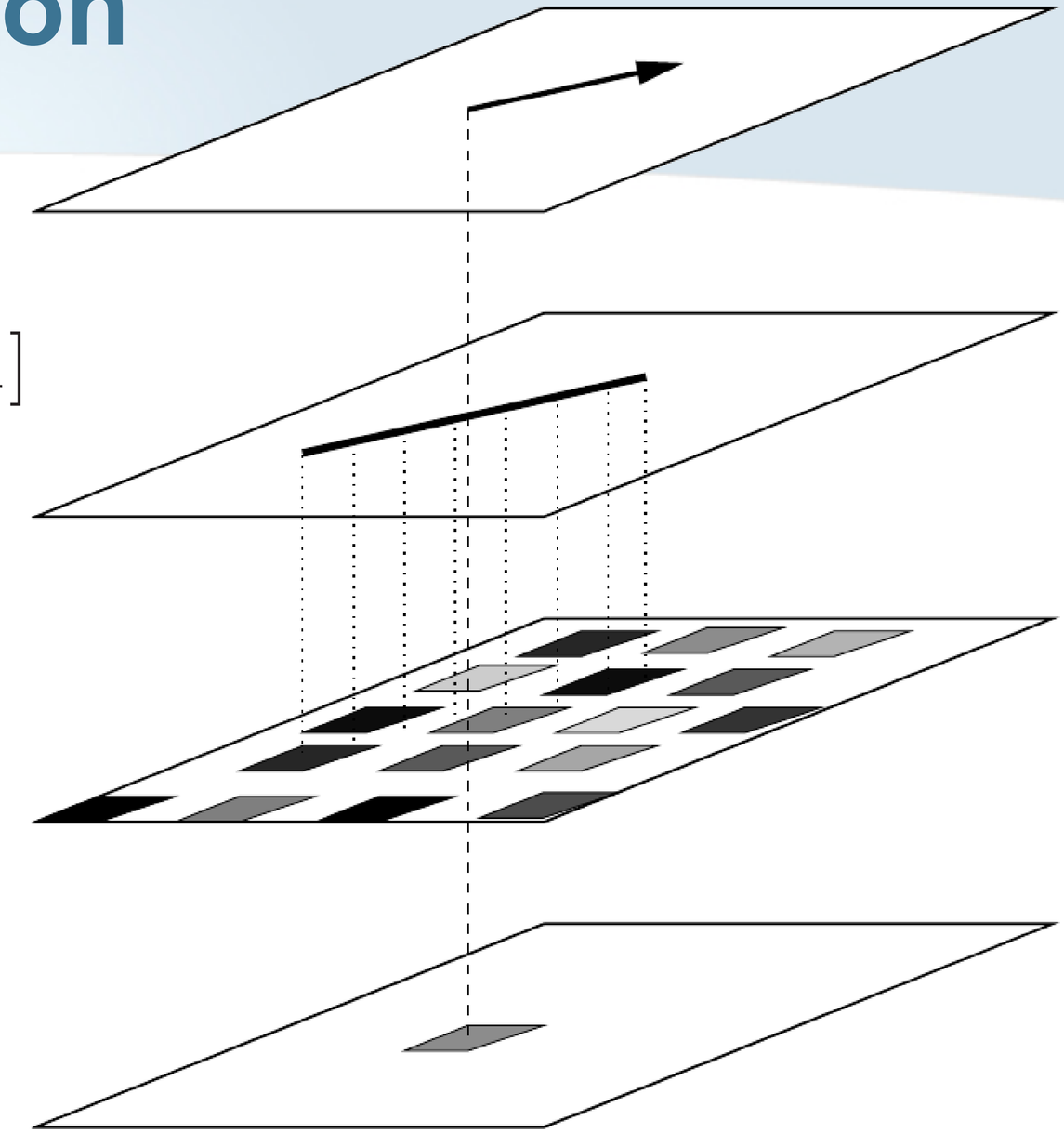


[Helgeland]

Line Integral Convolution

- Algorithm
  - Compute a noise field $\mathcal{N} : \mathcal{D} \rightarrow [0, 1]$
    - Random intensity for each vertex
  - For each vertex of the domain
    - Compute its integral curve $\mathcal{C}$
      – Backwards and forwards
    - Convolution with the noise field

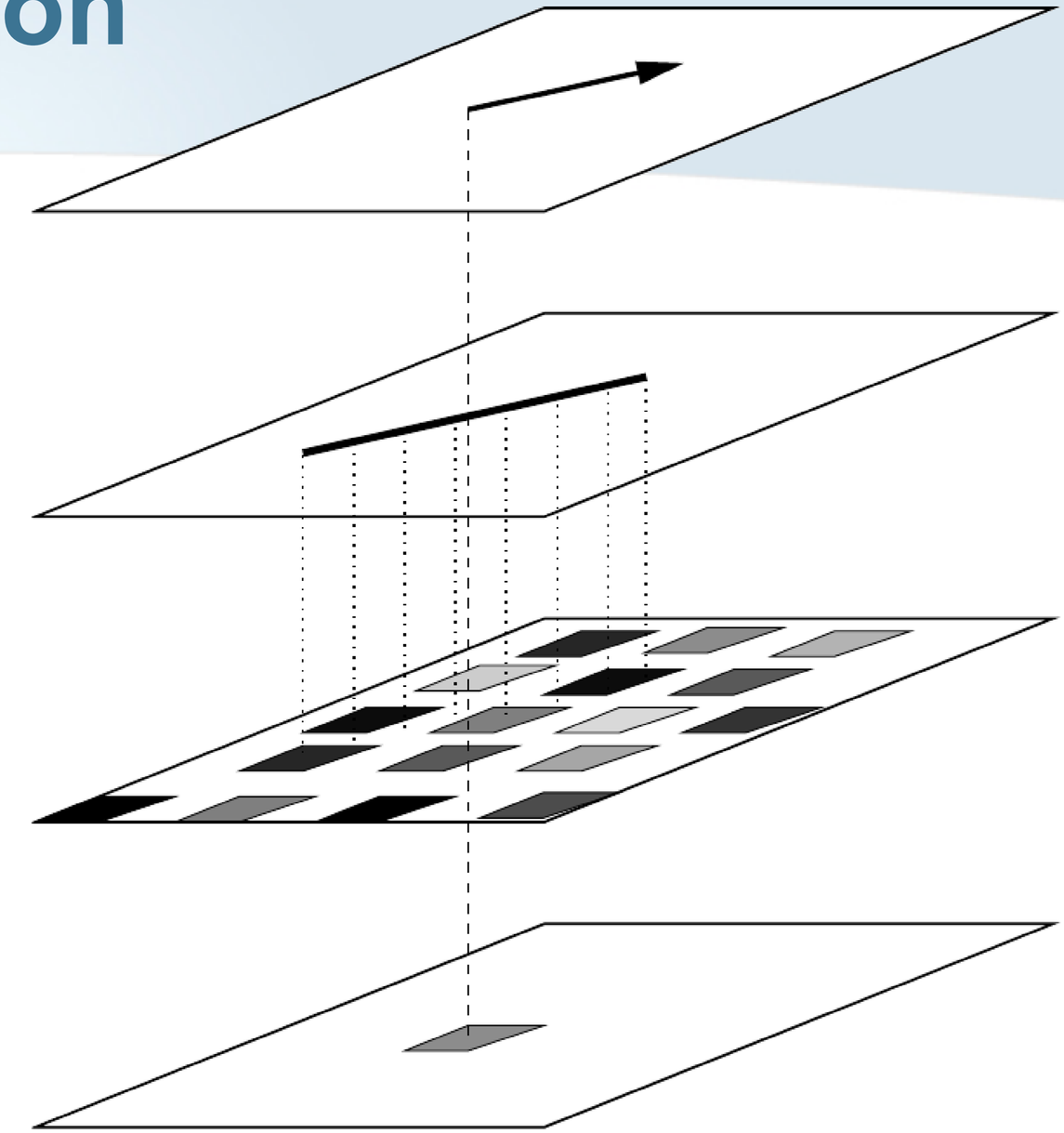
$$LIC(p) = \int_{c(p)-L}^{c(p)+L} k(u) \; . \; \mathcal{N}(c^{-1}(u))du$$



[Helgeland]

Line Integral Convolution

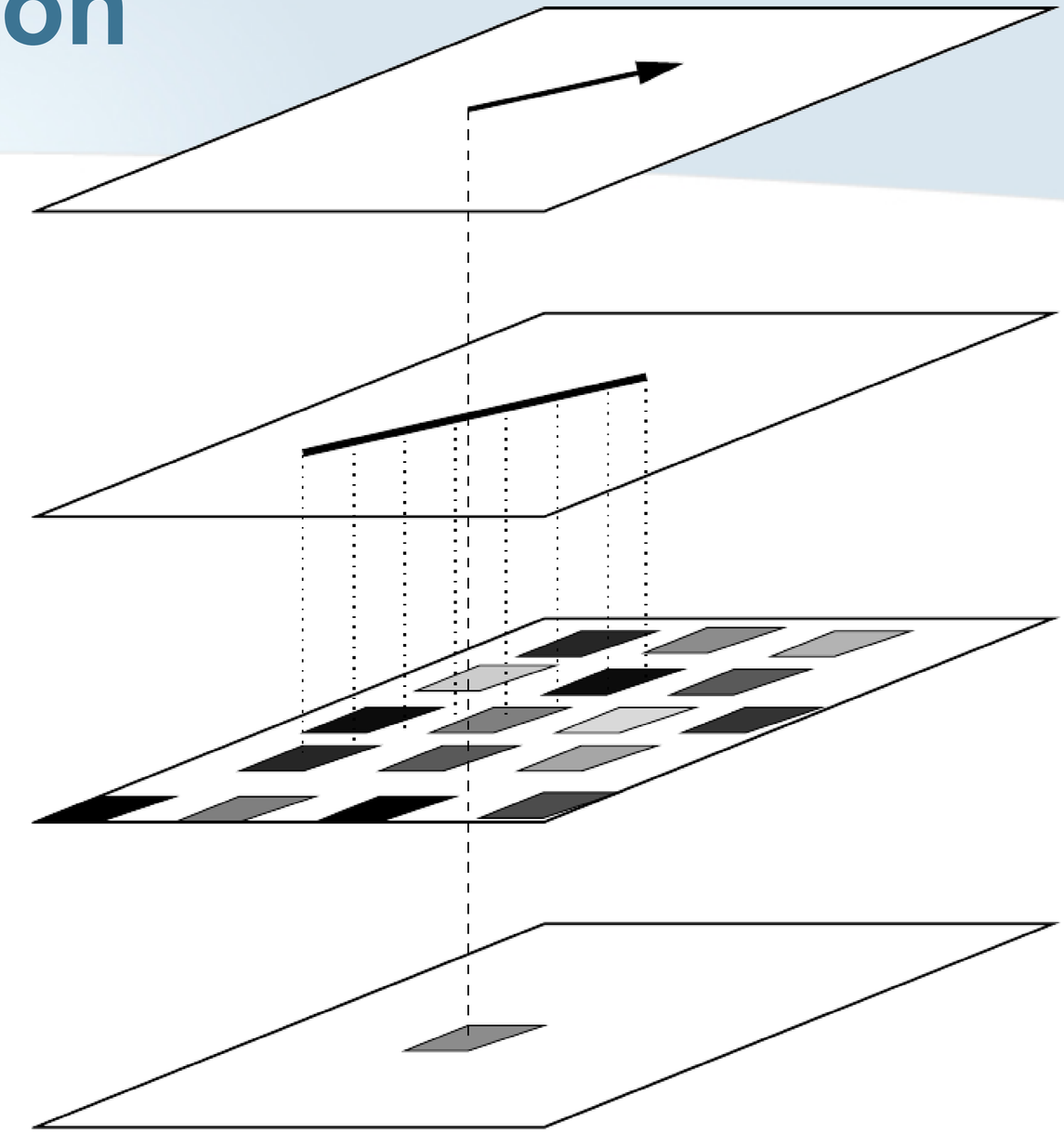
- Convolution kernel $k(u)$



[Helgeland]

Line Integral Convolution

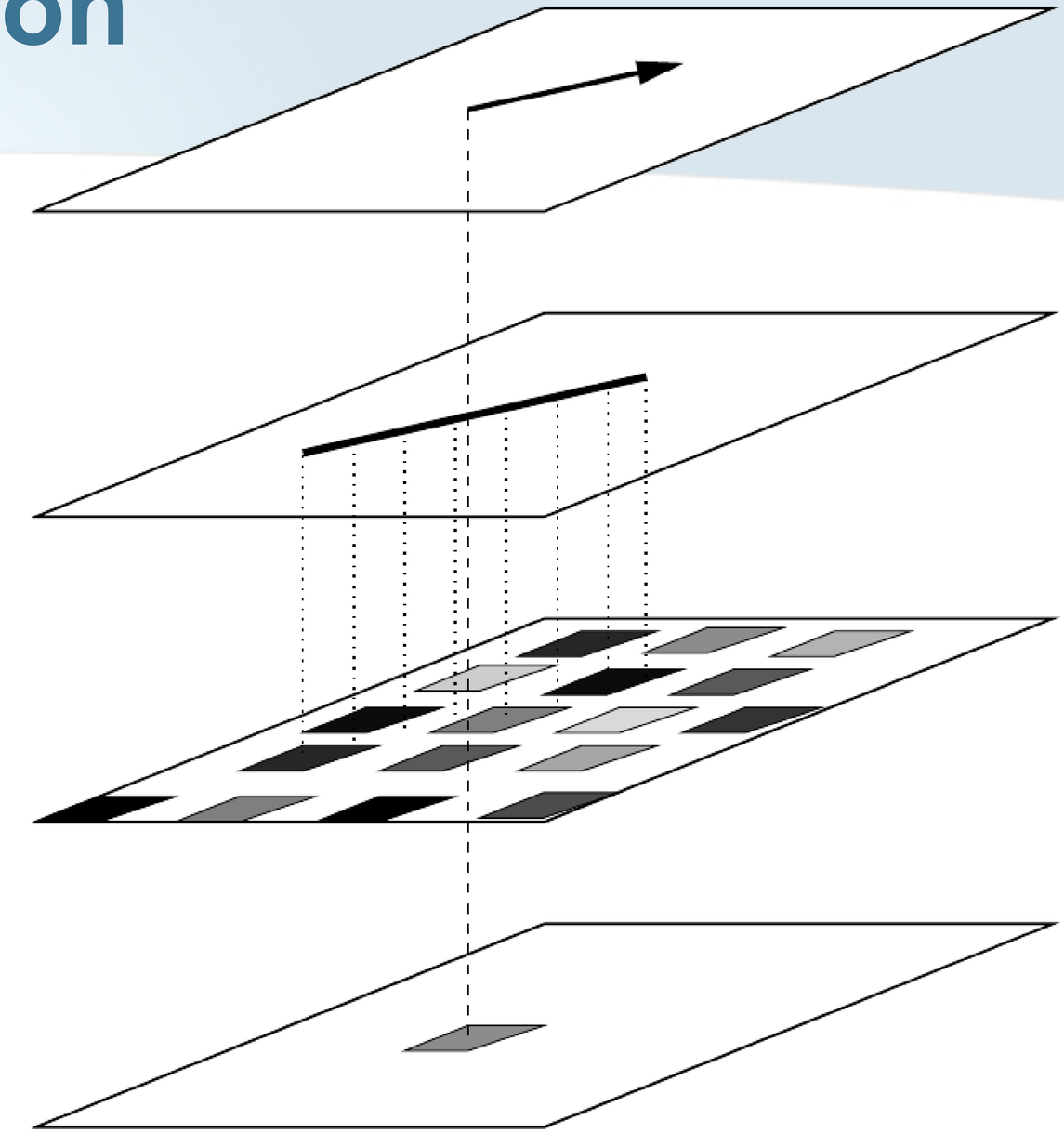
- Convolution kernel $k(u)$
  - Gaussian kernel



[Helgeland]

Line Integral Convolution

- Convolution kernel $k(u)$
  - Gaussian kernel
  - Finite support
    $[c(p) - L, c(p) + L]$

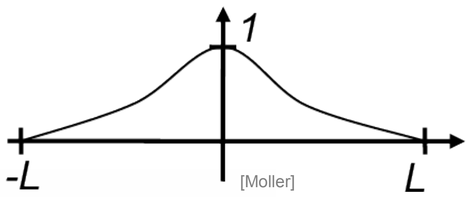


[Helgeland]

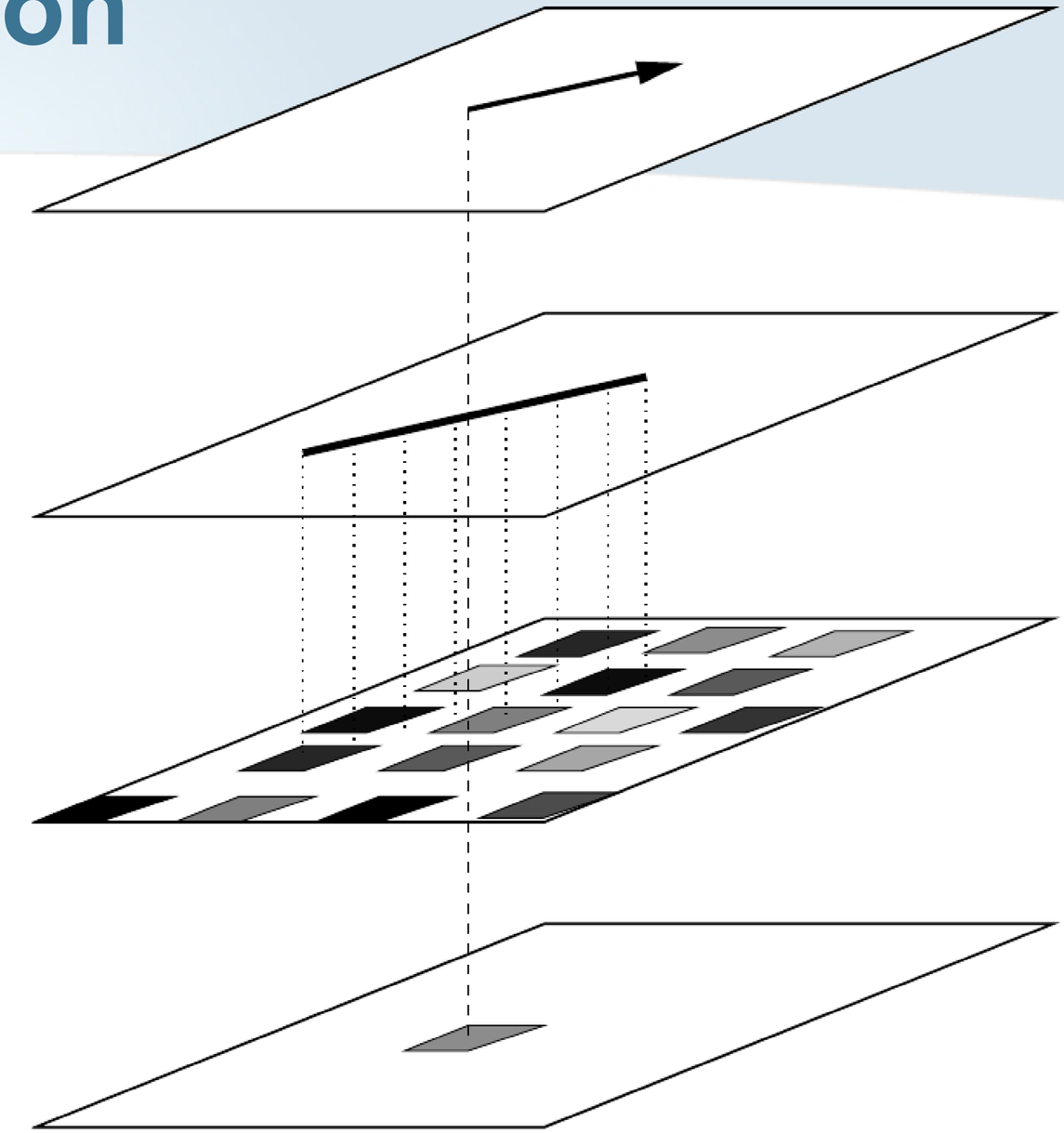
Line Integral Convolution

- Convolution kernel $k(u)$
  - Gaussian kernel
  - Finite support
    $[c(p) - L, c(p) + L]$
  - Normalized

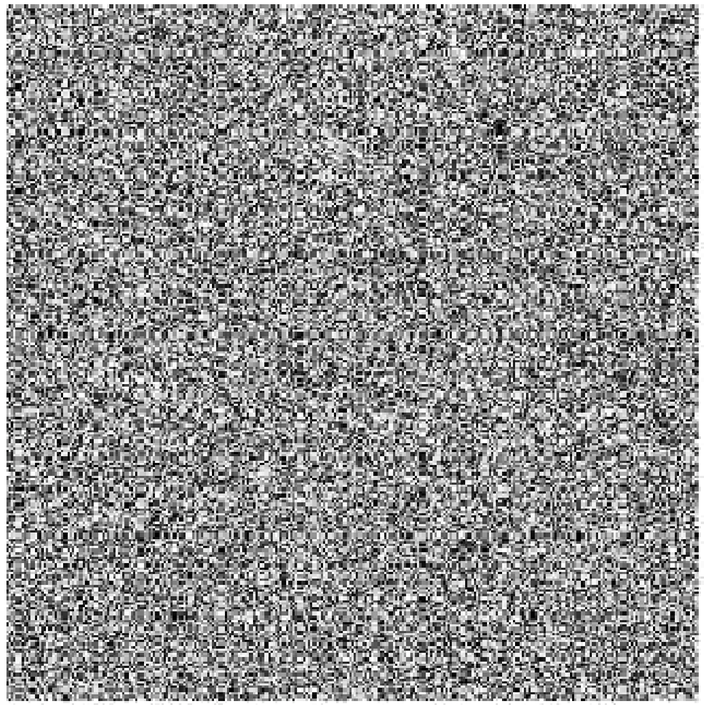
$$\int_{c(p)-L}^{c(p)+L} k(u)du = 1$$

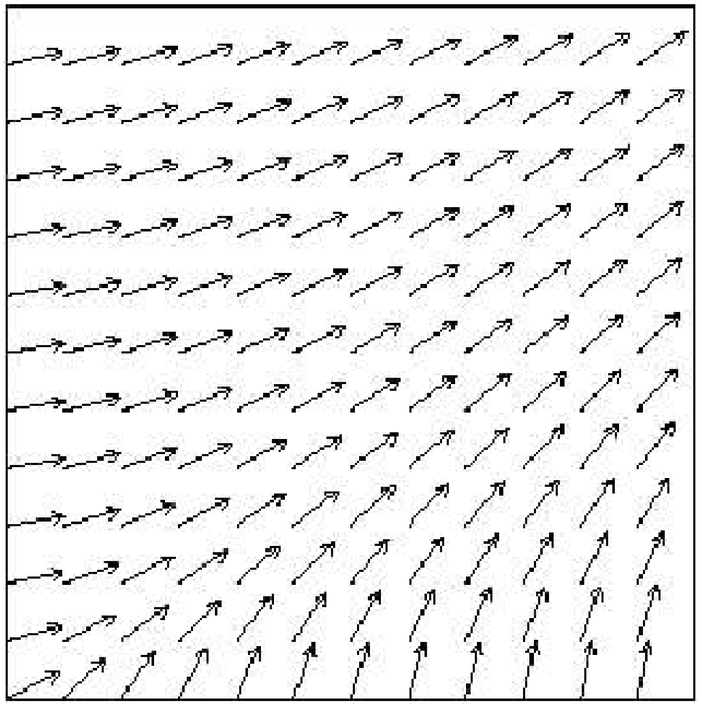


[Moller]


[Helgeland]

Line Integral Convolution





[Helgeland]

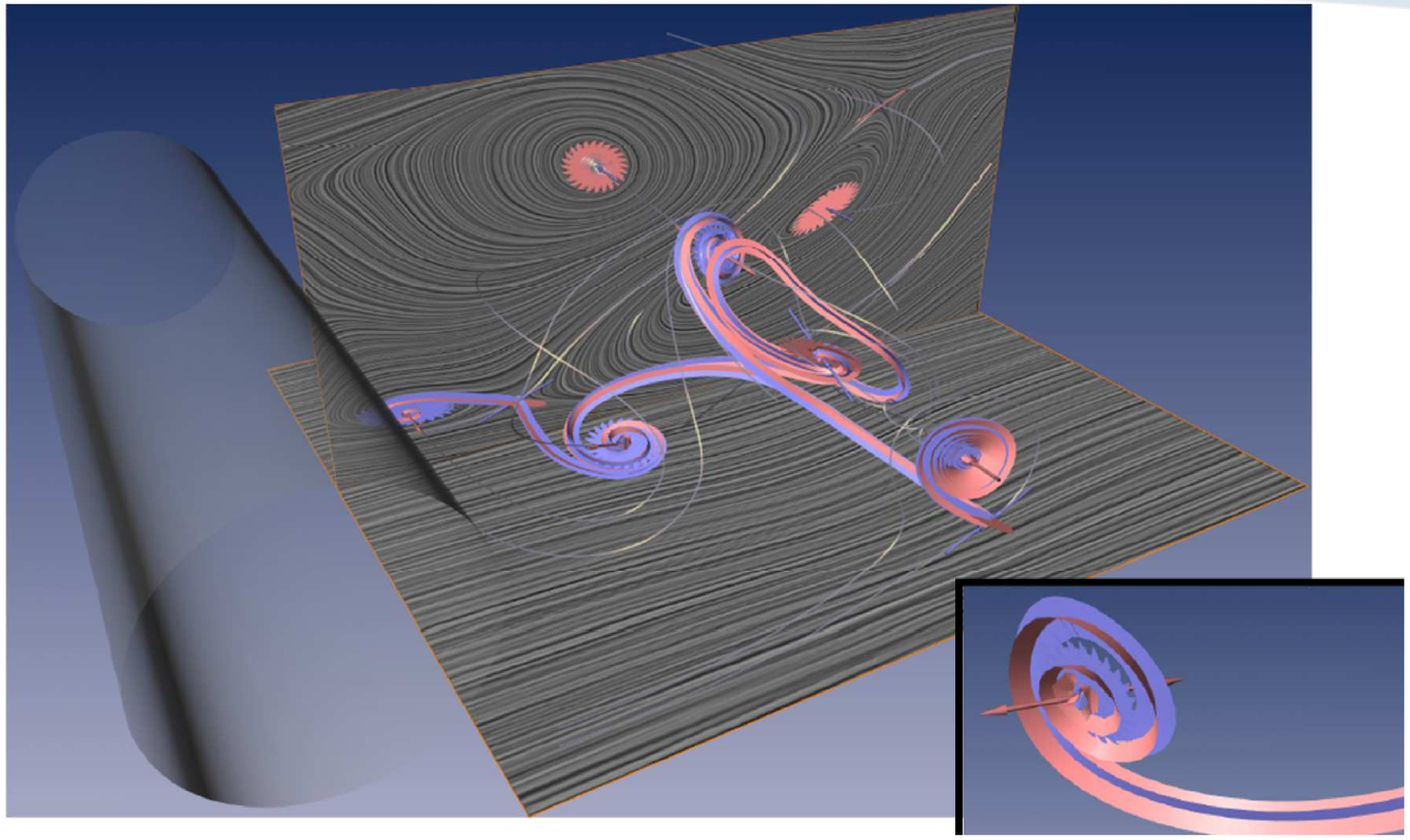
LIC for volumetric domains

LIC for volumetric domains

- Easy way
  - Slice the volume

LIC for volumetric domains

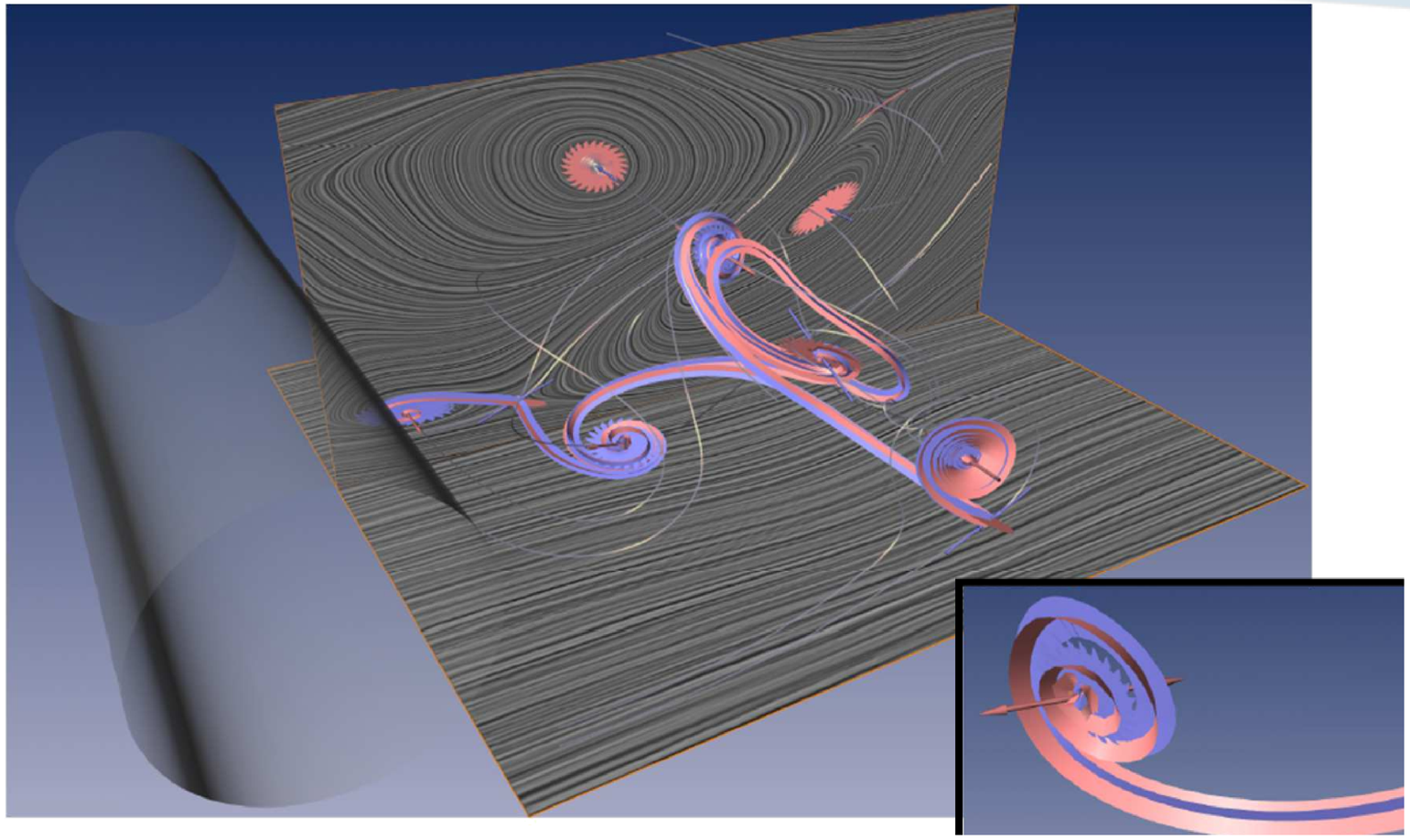
- Easy way
  - Slice the volume
  - LIC for each slice



[Theisel 03]

LIC for volumetric domains

- Easy way
  - Slice the volume
  - LIC for each slice
- Also,
  - What is LIC in the end?

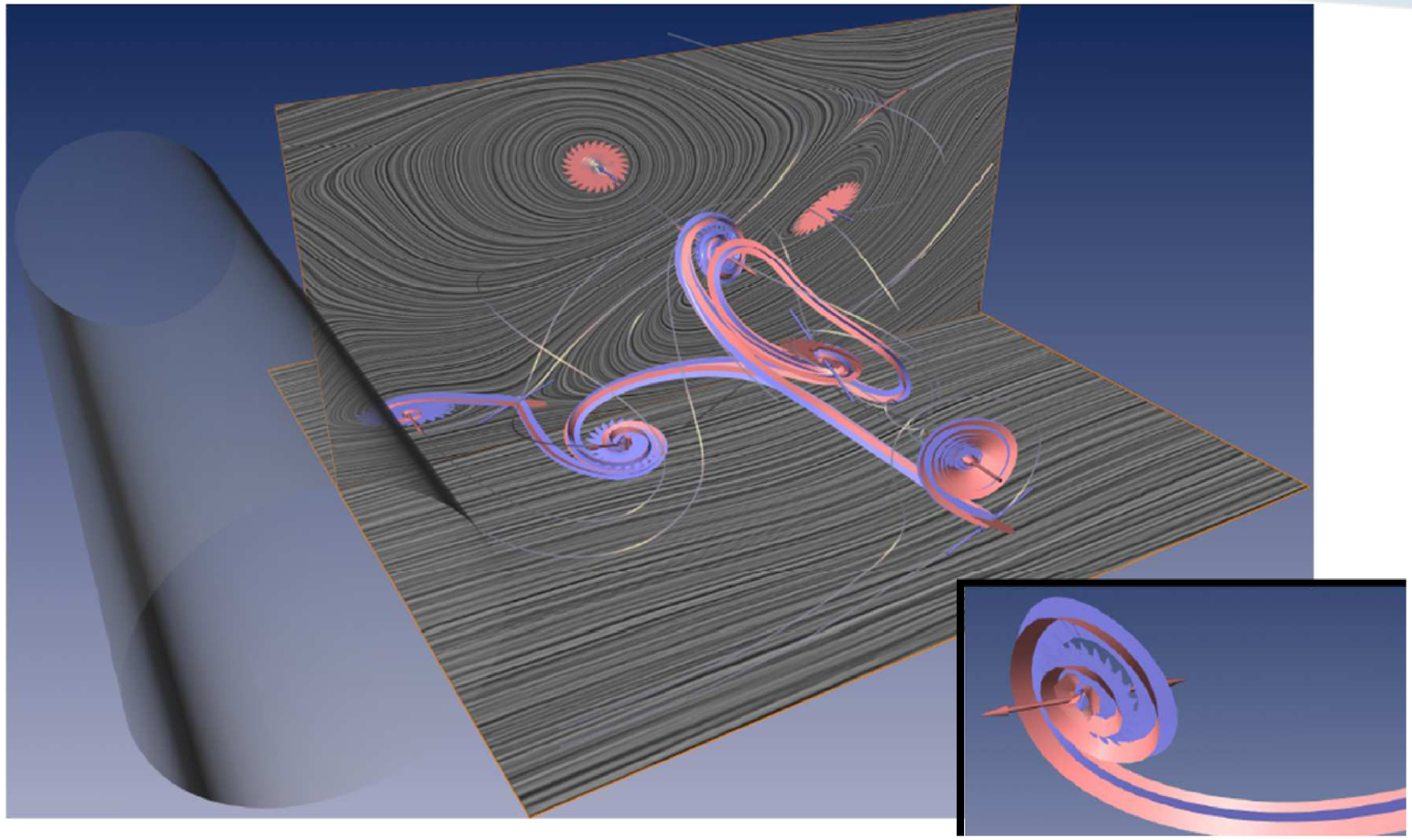


[Theisel 03]

LIC for volumetric domains

- Easy way
  - Slice the volume
  - LIC for each slice

- Also,
  - What is LIC in the end?
  - **A scalar field**

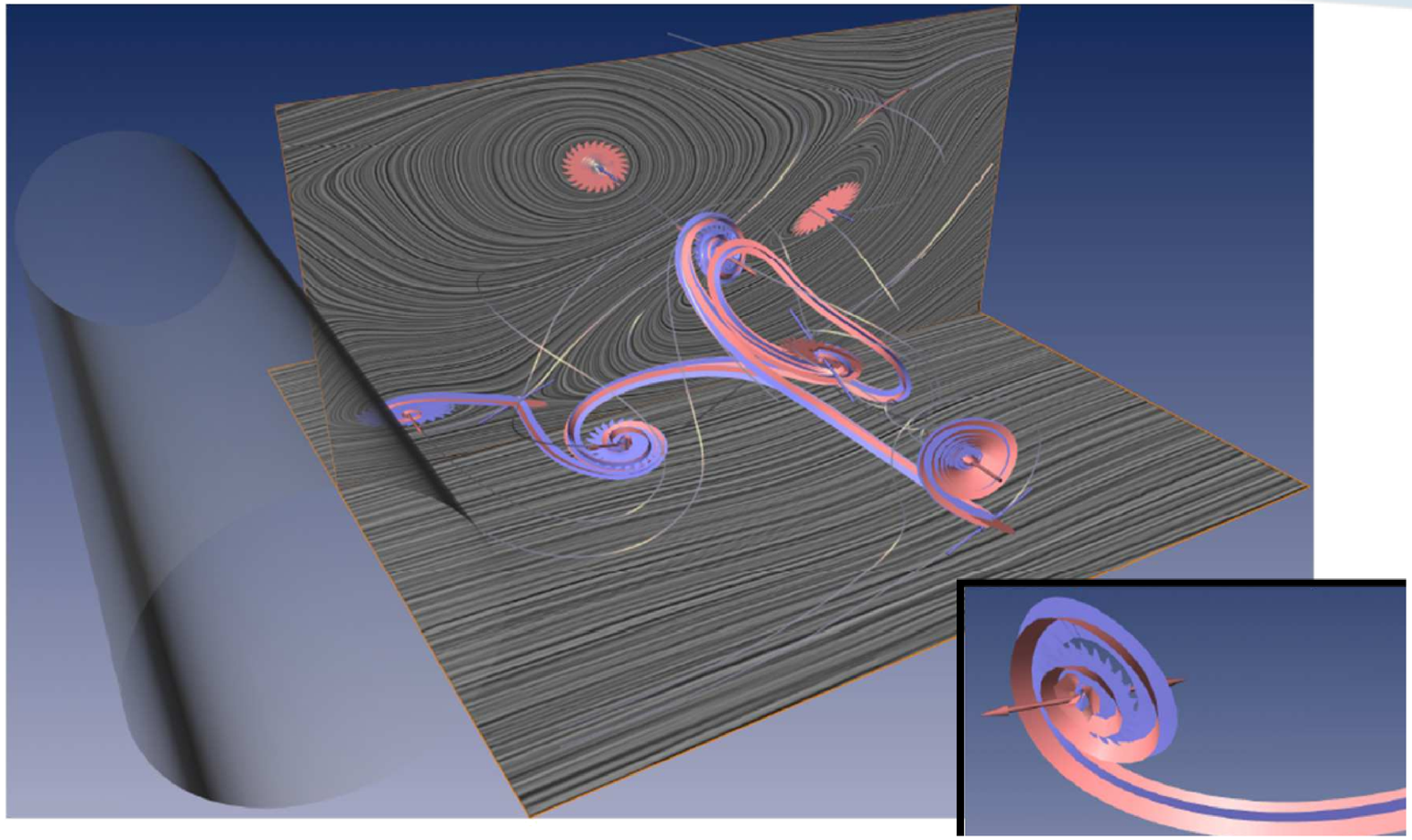


[Theisel 03]

LIC for volumetric domains

- Easy way
  - Slice the volume
  - LIC for each slice

- Also,
  - What is LIC in the end?
  - **A scalar field**
    - Volume rendering!

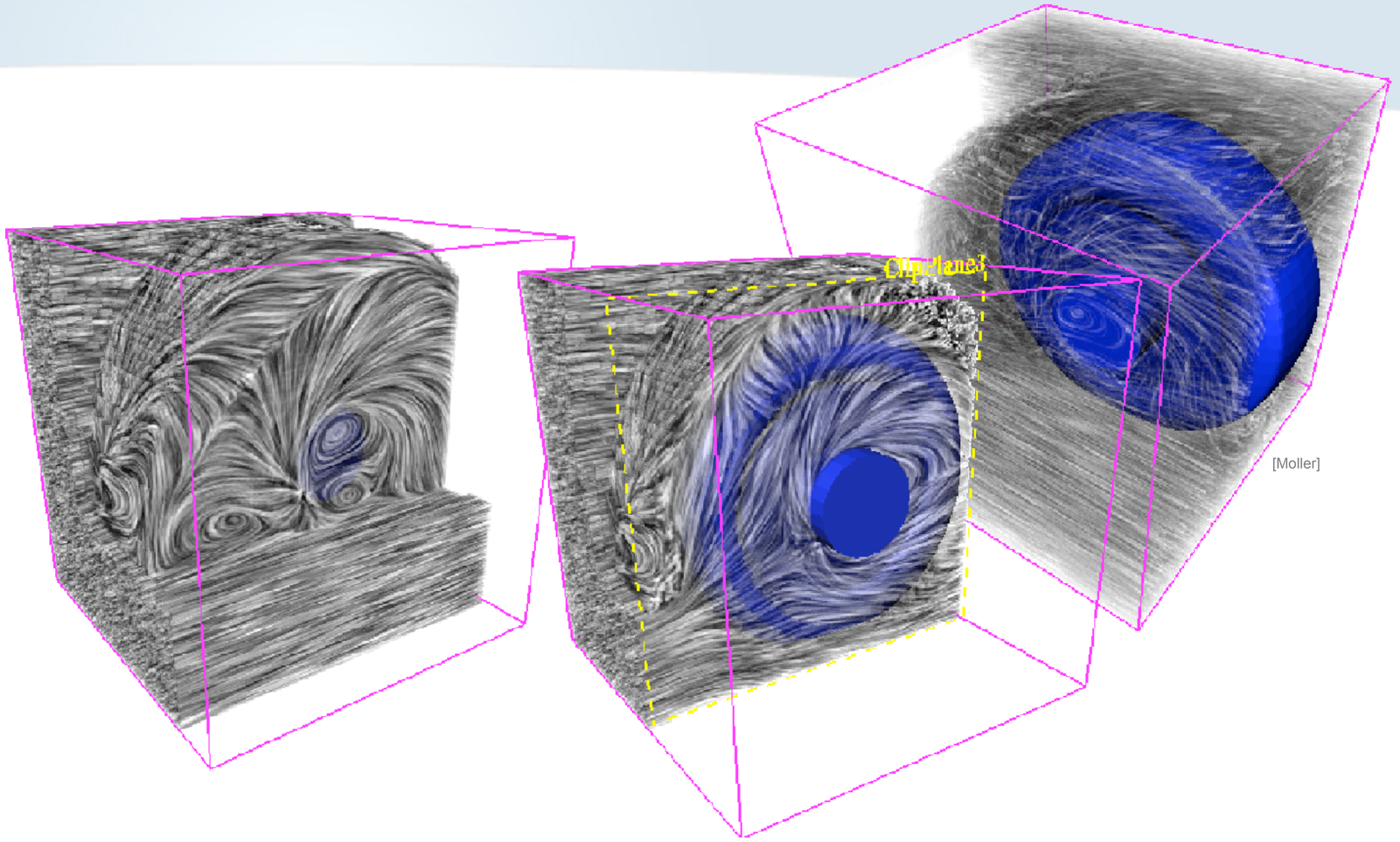


[Theisel 03]

LIC for volumetric domains

- Easy way
  - Slice the volume
  - LIC for each slice

- Also,
  - What is LIC in the end?
  - **A scalar field**
    - Volume rendering!
    - Clipping often necessary

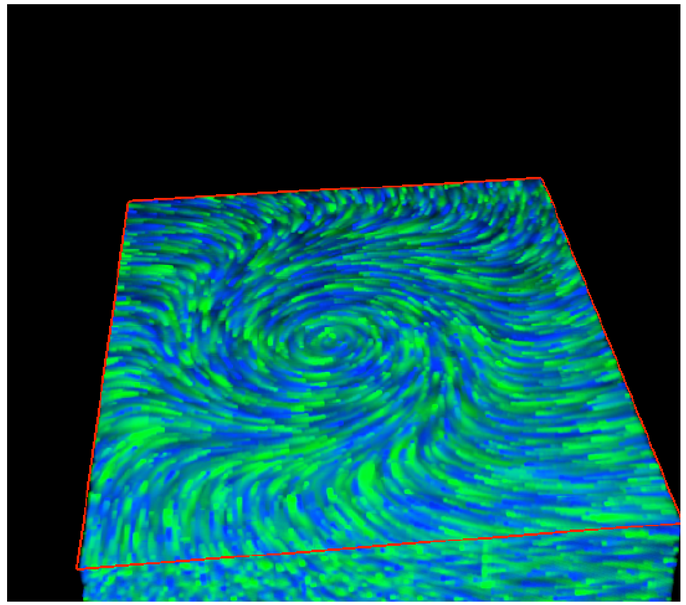


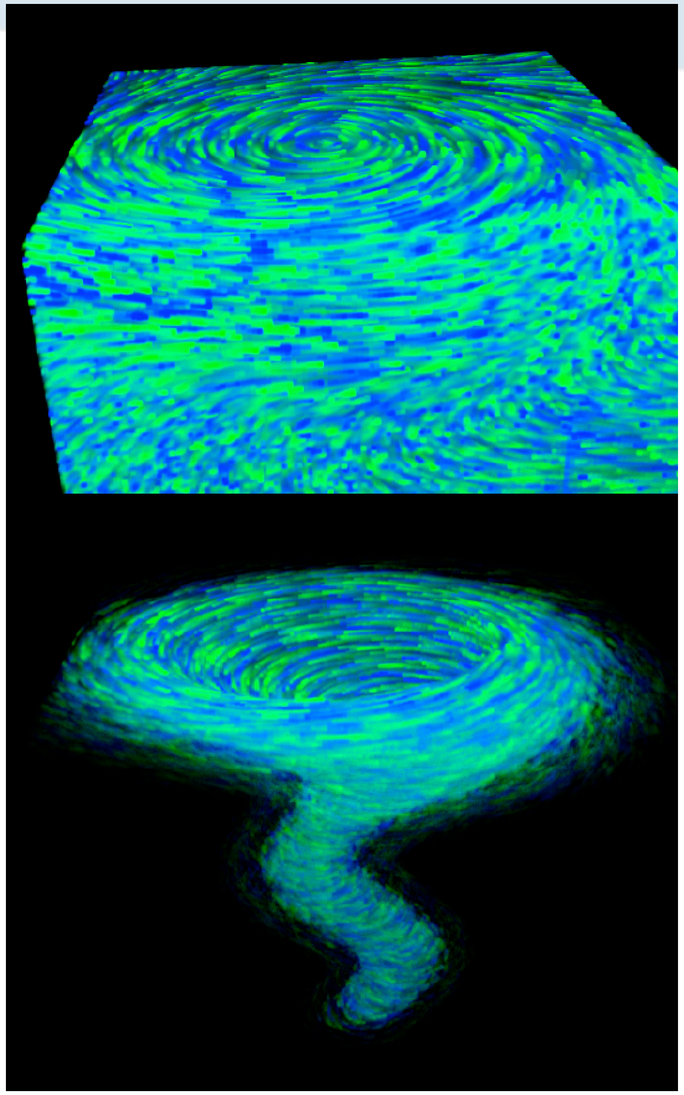
[Moller]

LIC for volumetric domains

- Easy way
  - Slice the volume
  - LIC for each slice

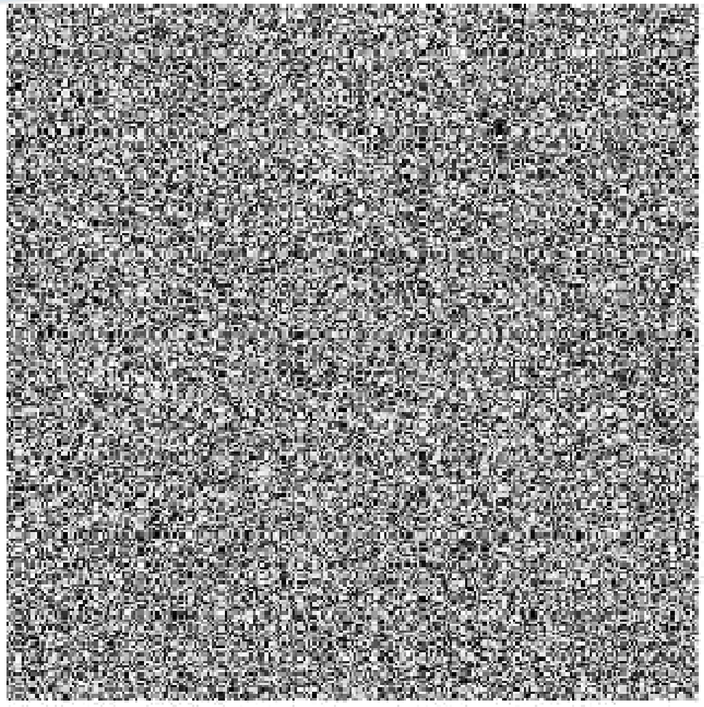
- Also,
  - What is LIC in the end?
  - **A scalar field**
    - Volume rendering!
    - Clipping often necessary

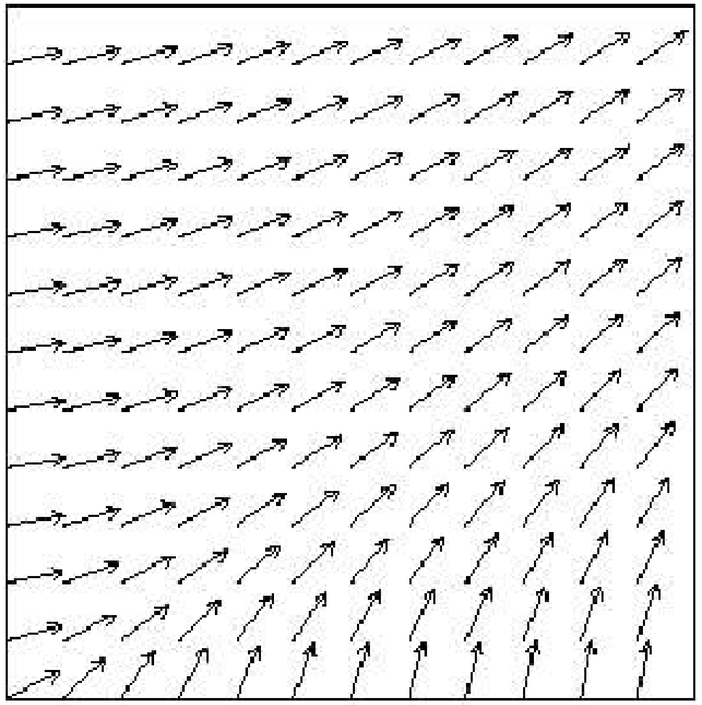




[Moller]

Line Integral Convolution

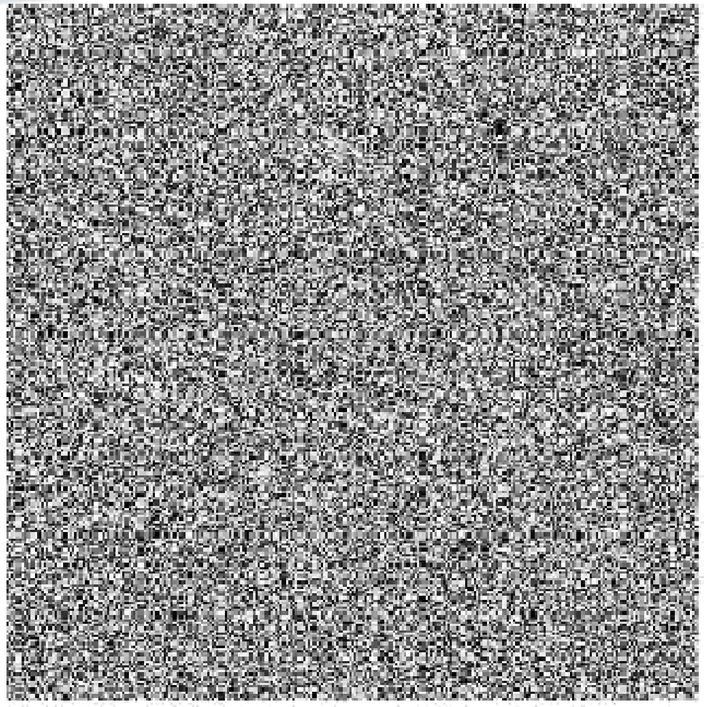


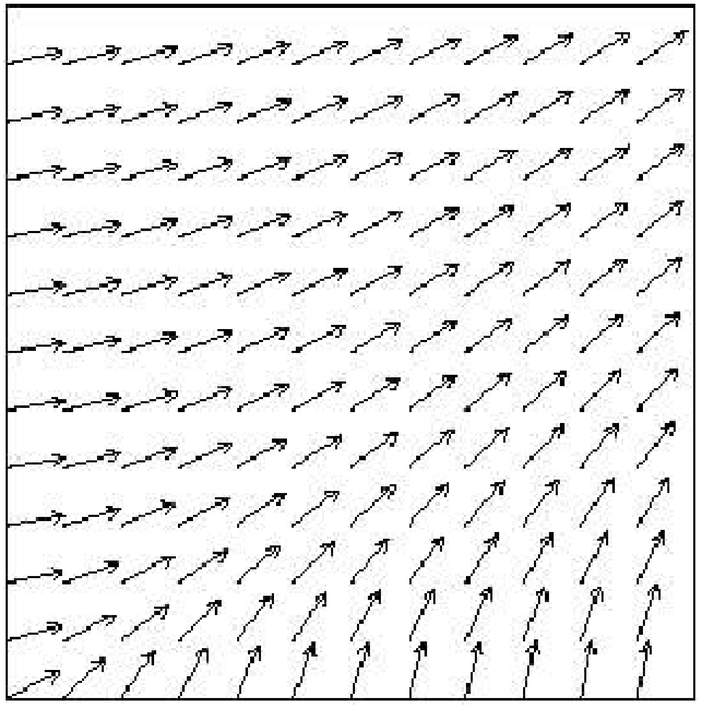


[Helgeland]

- Global visualization of the direction of the flow

Line Integral Convolution

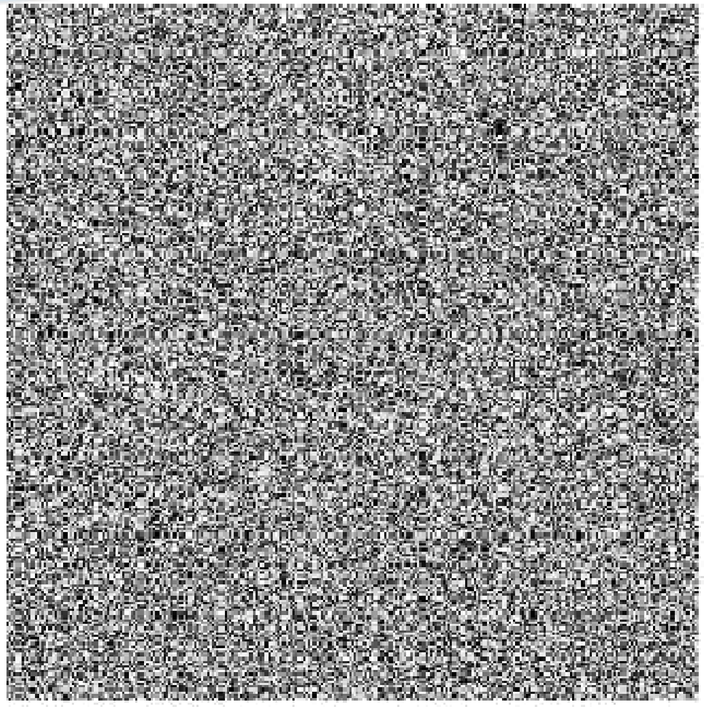


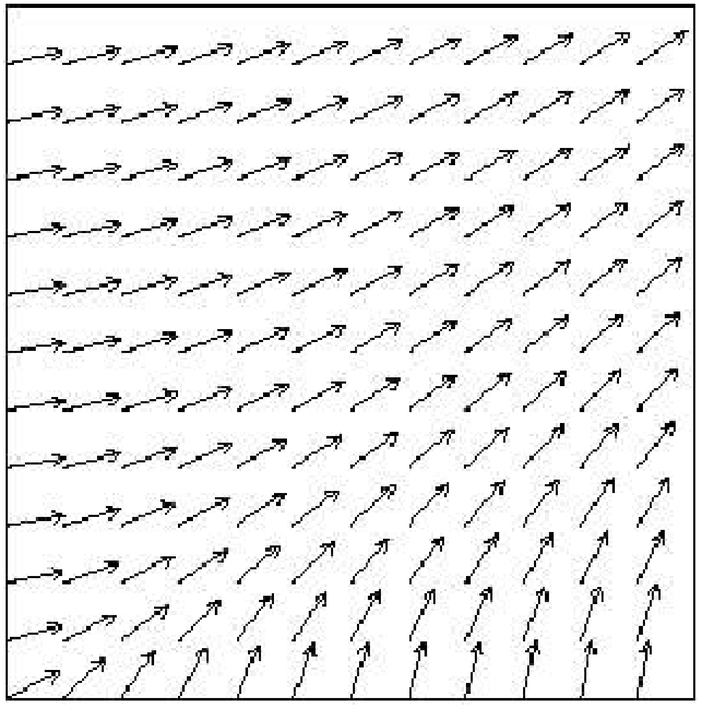


[Helgeland]

- Global visualization of the direction of the flow
  - What about its magnitude?

Line Integral Convolution



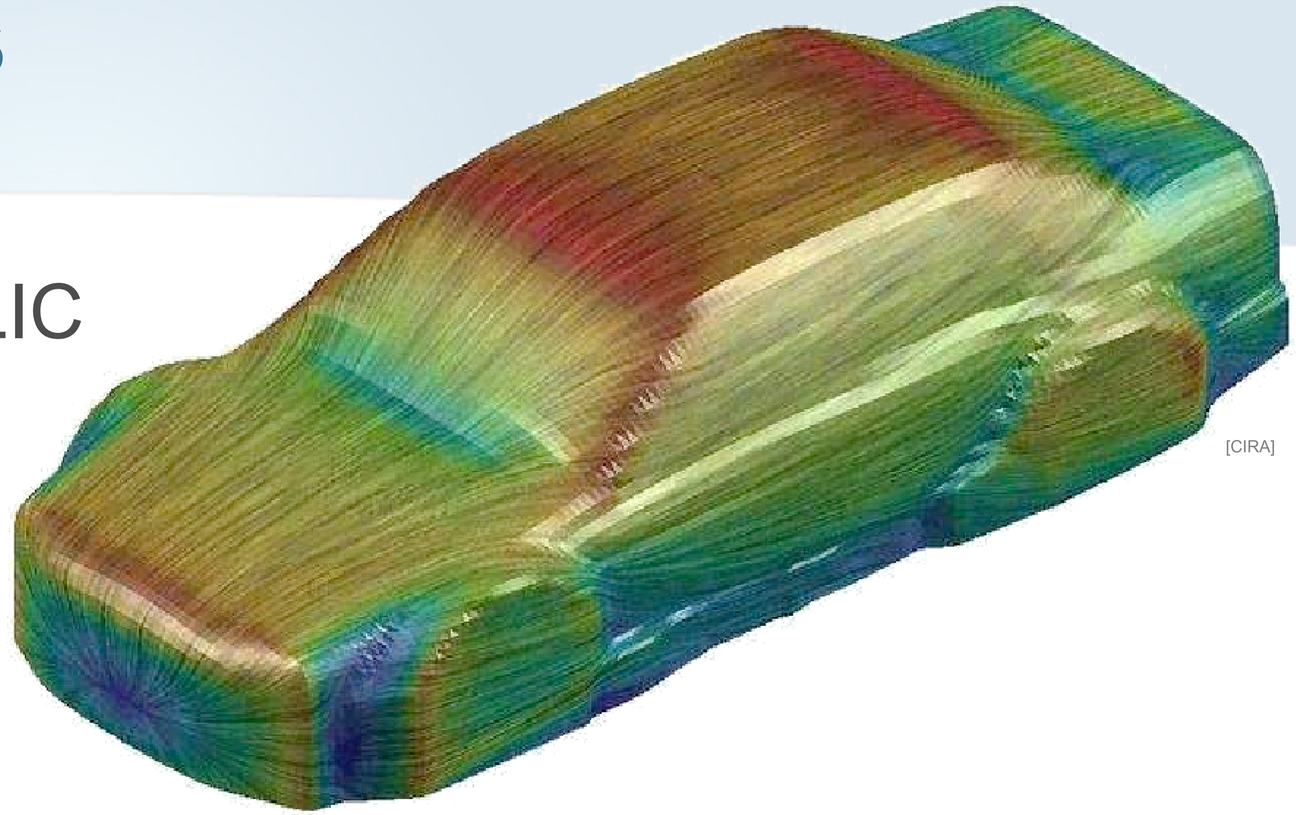


[Helgeland]

- Global visualization of the direction of the flow
  - What about its magnitude? Its orientation?

Derived scalar fields

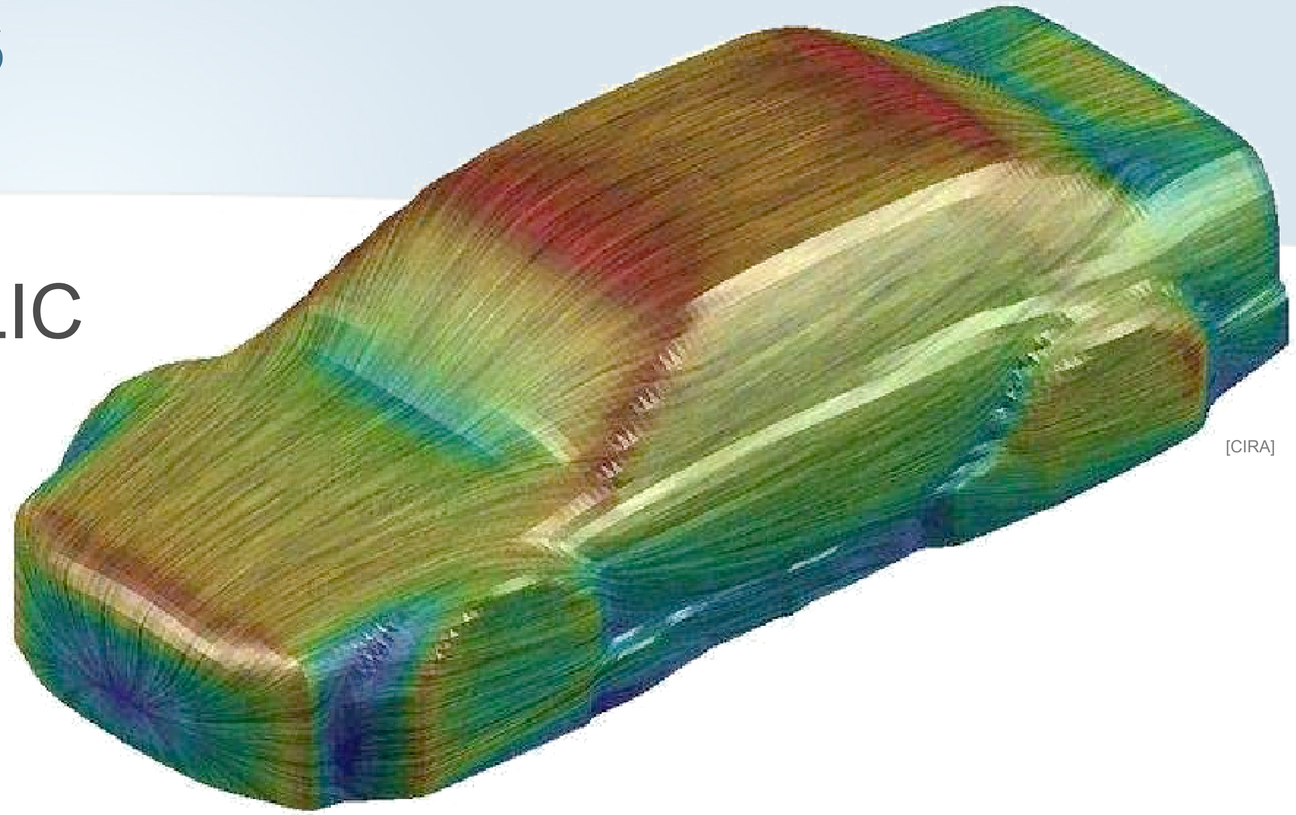
- Combine the magnitude with the LIC



[CIRA]

Derived scalar fields

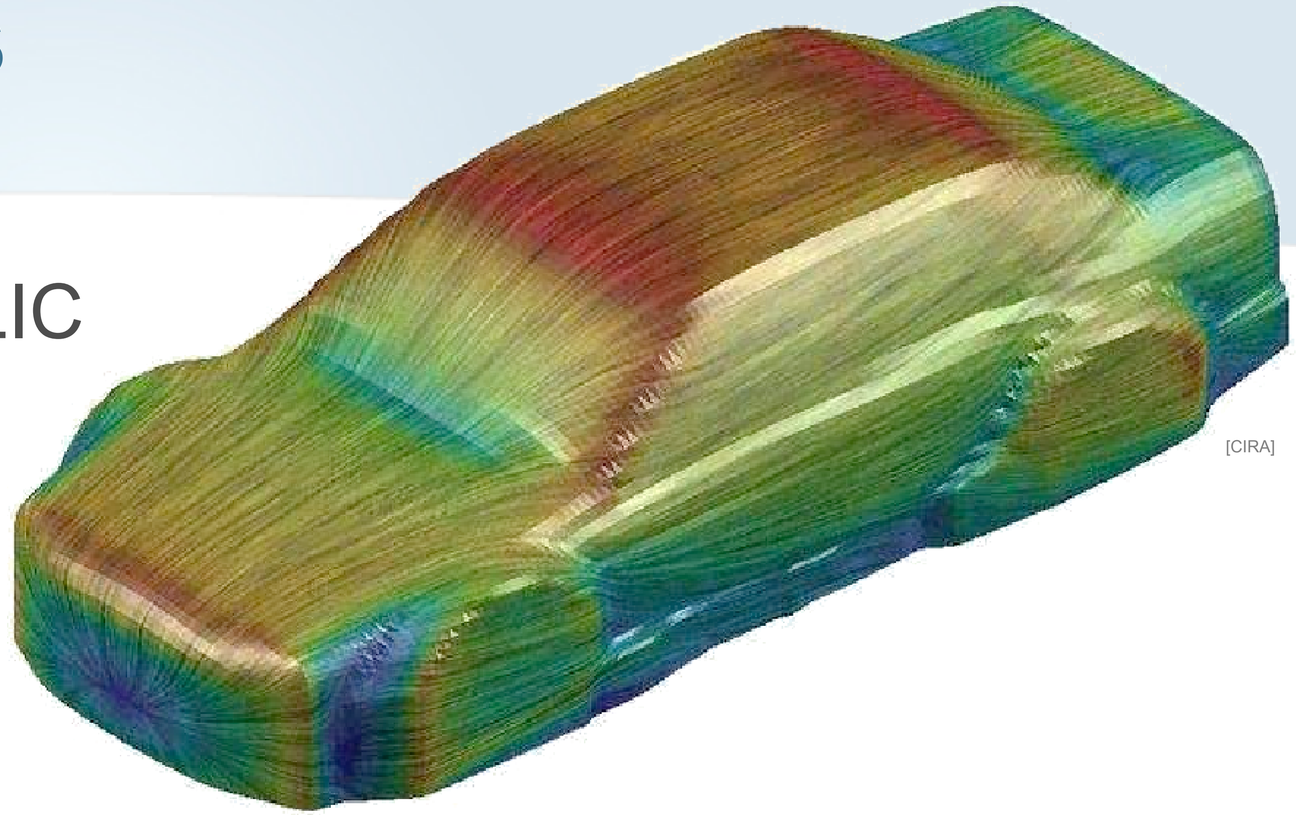
- Combine the magnitude with the LIC
  - Color map of the magnitude



[CIRA]

Derived scalar fields

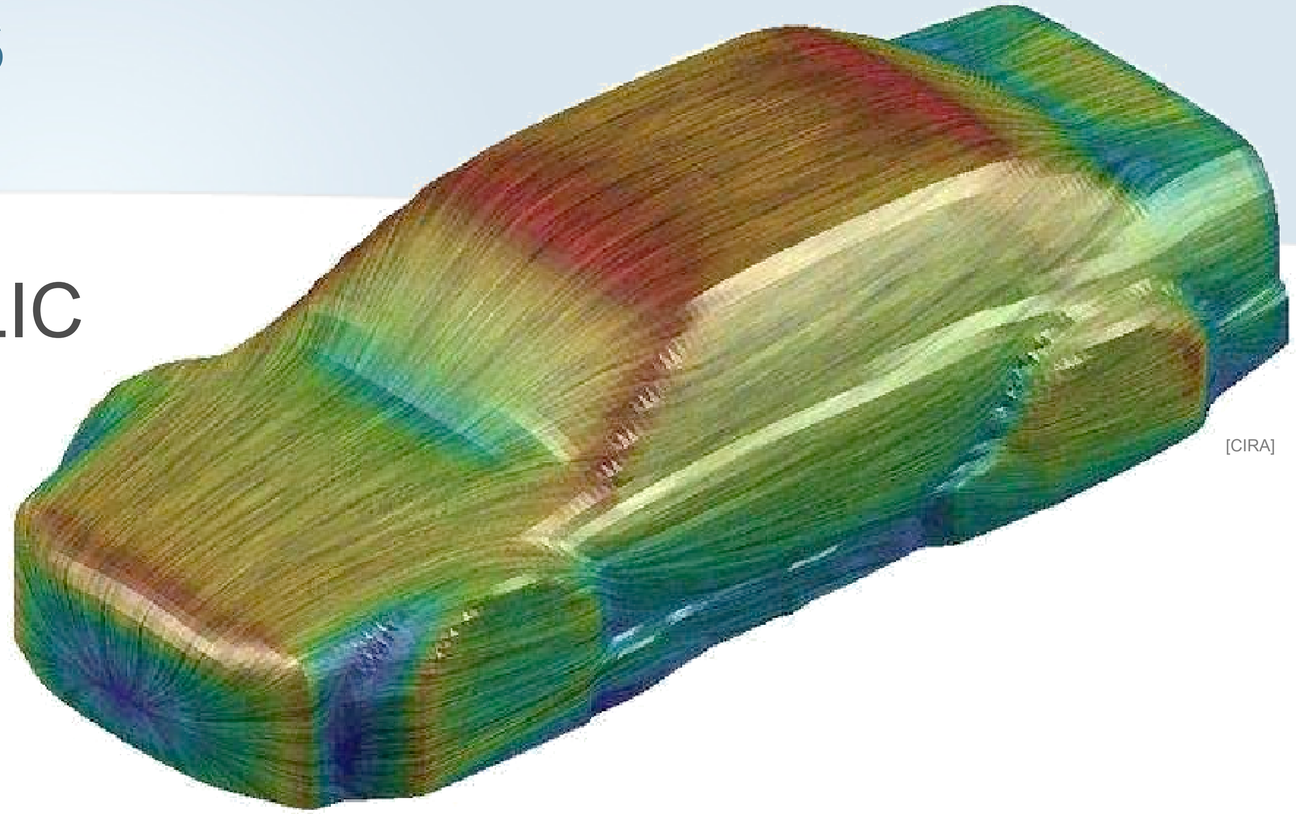
- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value



[CIRA]

Derived scalar fields

- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value

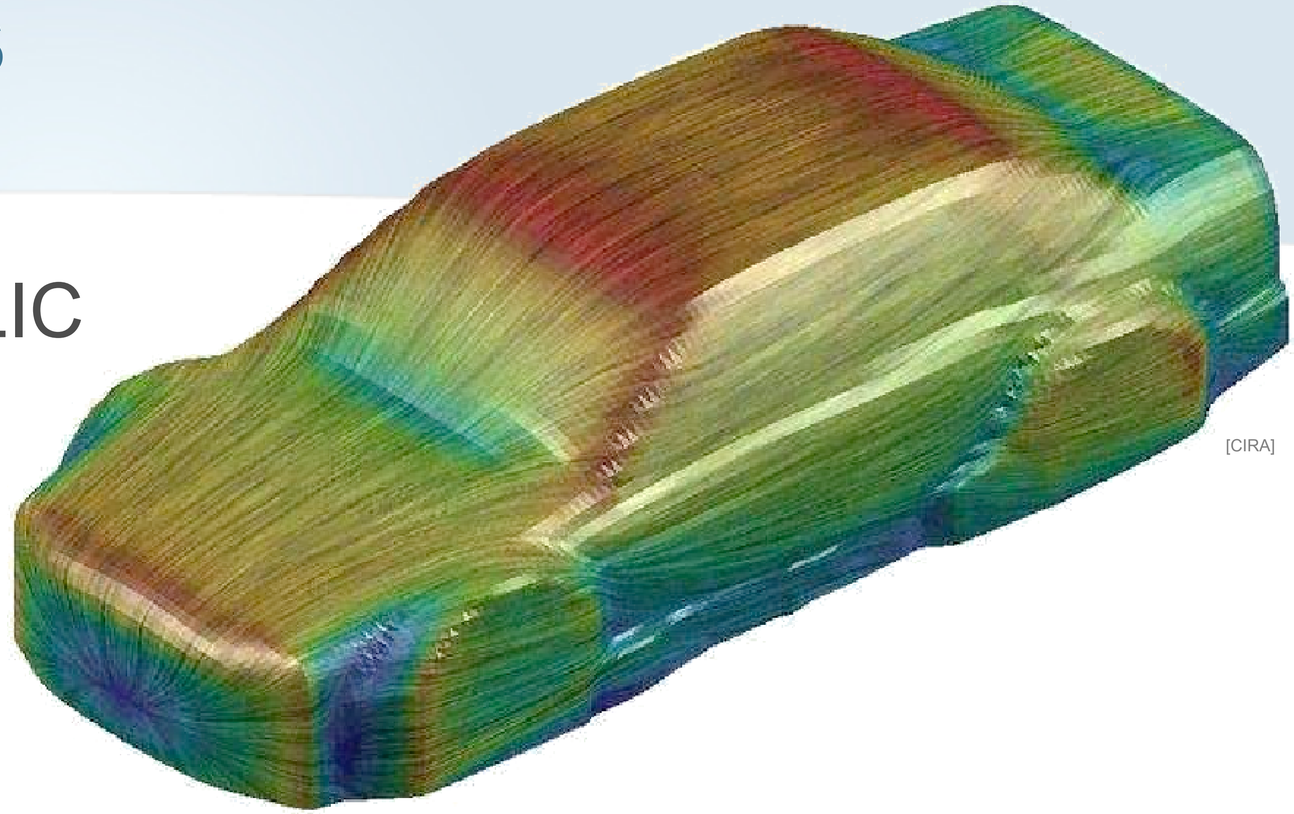


[CIRA]

- What other derived scalar fields would be interesting?

Derived scalar fields

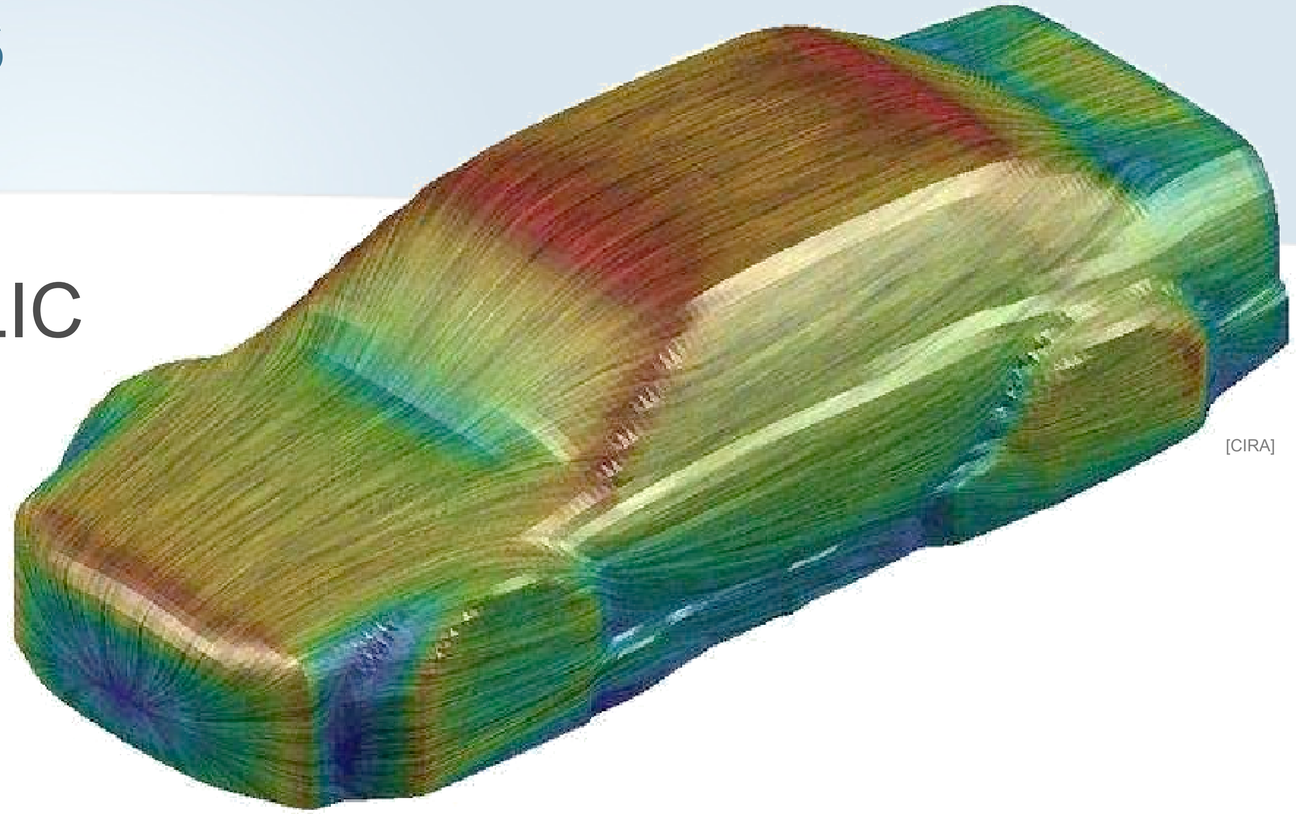
- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value


[CIRA]

- What other derived scalar fields would be interesting?
  - Flow orientation

Derived scalar fields

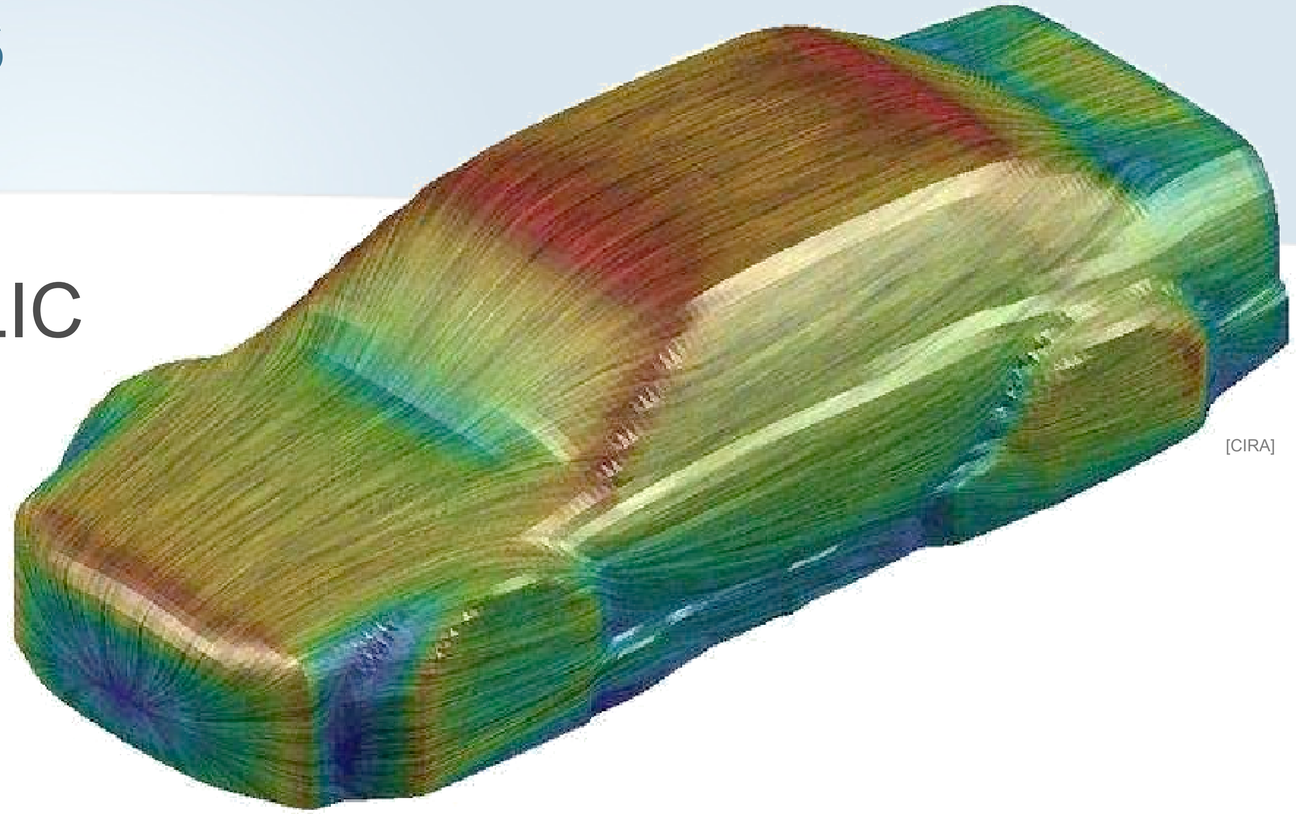
- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value


[CIRA]

- What other derived scalar fields would be interesting?
  - Flow orientation: divergence

Derived scalar fields

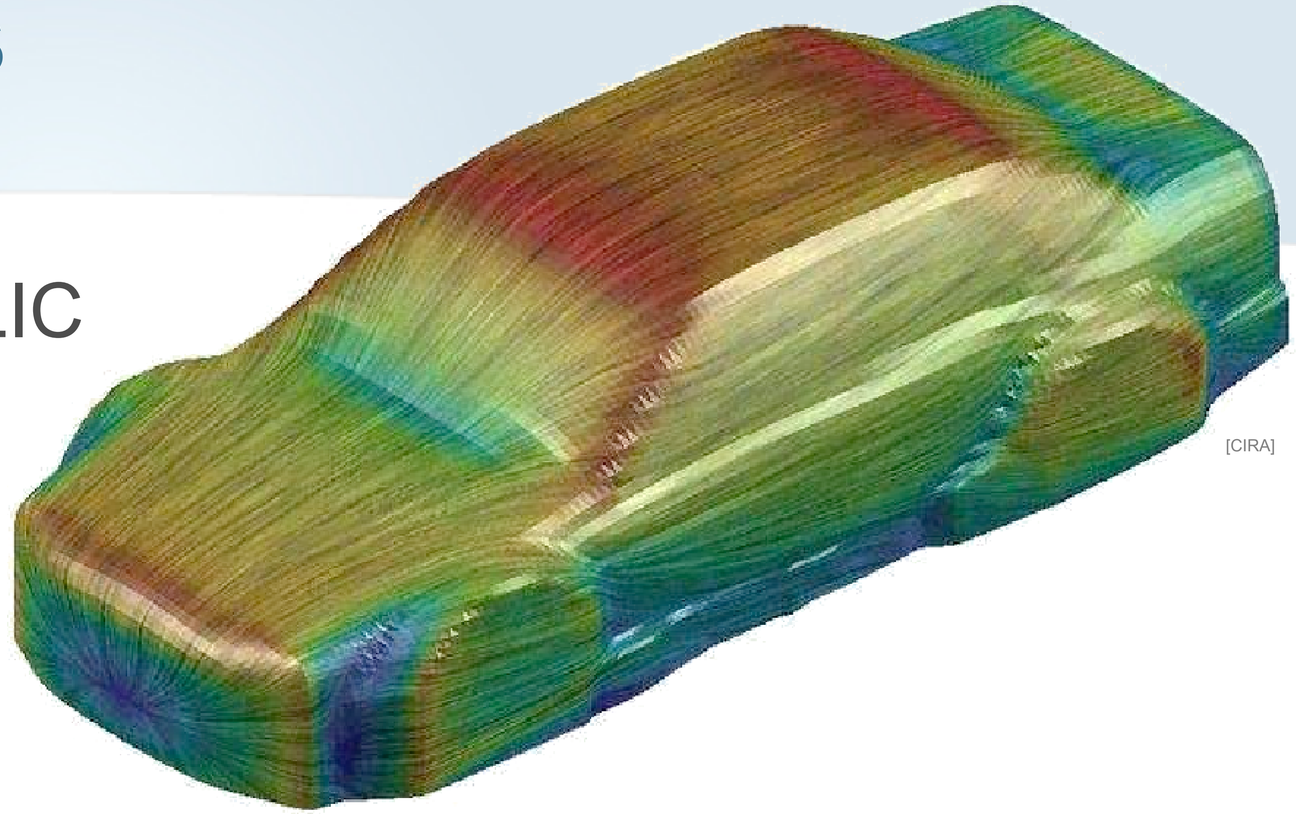
- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value


[CIRA]

- What other derived scalar fields would be interesting?
  - Flow orientation: divergence
  - Angular speed

Derived scalar fields

- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value

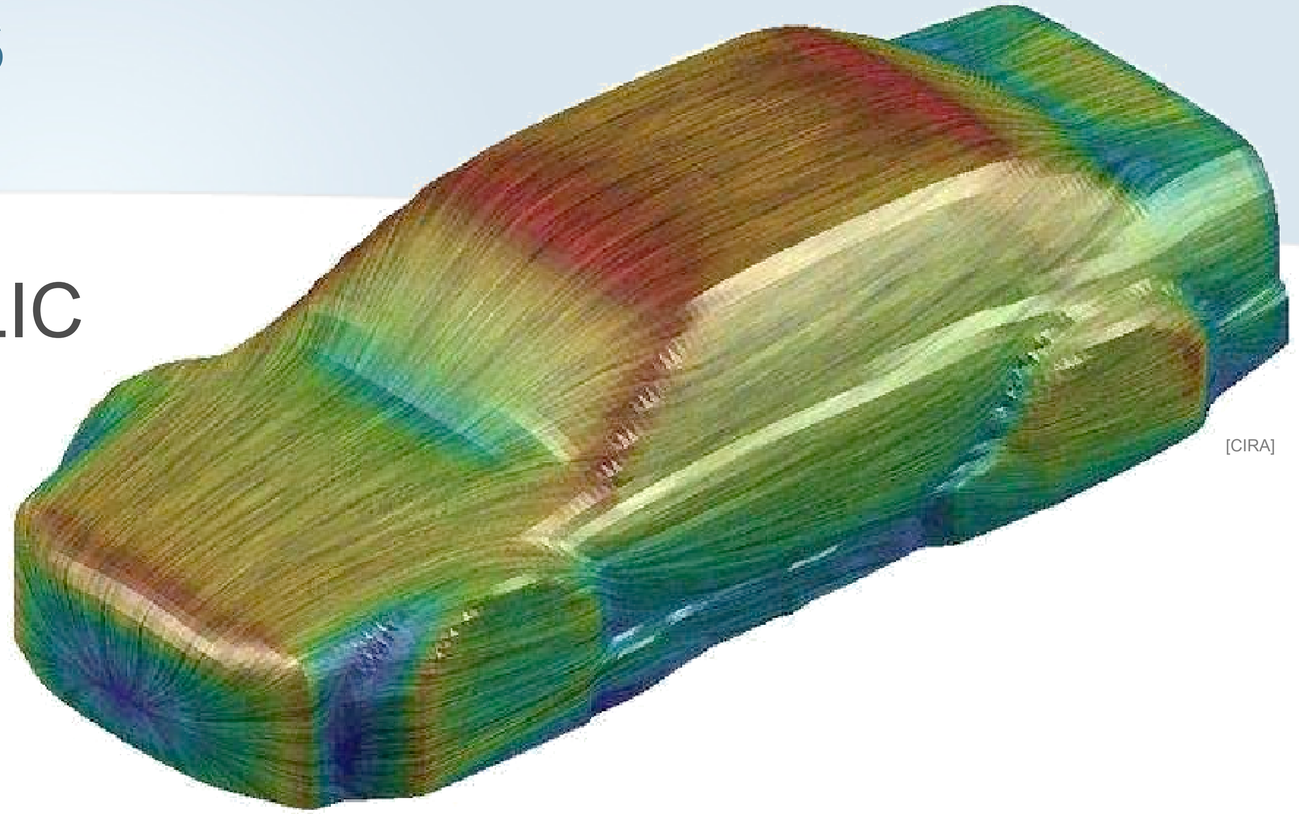


[CIRA]

- What other derived scalar fields would be interesting?
  - Flow orientation: divergence
  - Angular speed: magnitude of the curl

Derived scalar fields

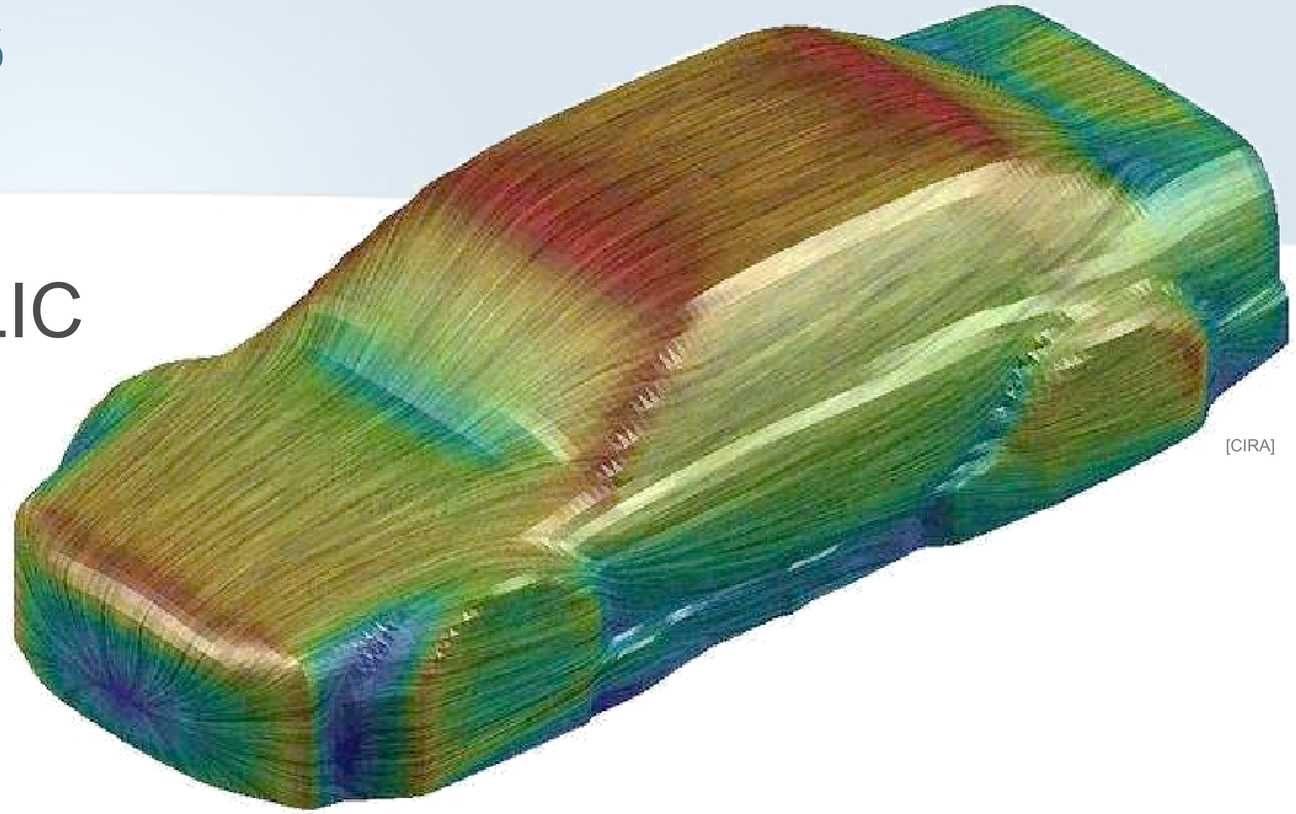
- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value


[CIRA]

- What other derived scalar fields would be interesting?
  - Flow orientation: divergence
  - Angular speed: magnitude of the curl
  - Flow distortion

Derived scalar fields

- Combine the magnitude with the LIC
  - Color map of the magnitude
  - For each channel (RGB)
    - Multiply by the LIC value



[CIRA]

- What other derived scalar fields would be interesting?
  - Flow orientation: divergence
  - Angular speed: magnitude of the curl
  - Flow distortion: Finite Time Lyaponov Exponent

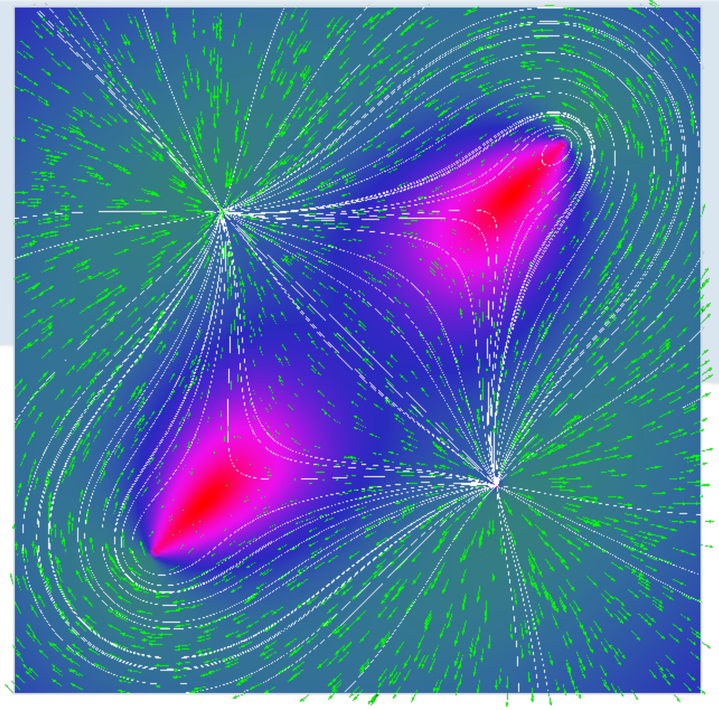
Flow orientation

Flow orientation

- Divergence of the flow, for instance
  - $f : \mathbb{R}^3 \to \mathbb{R}^3$

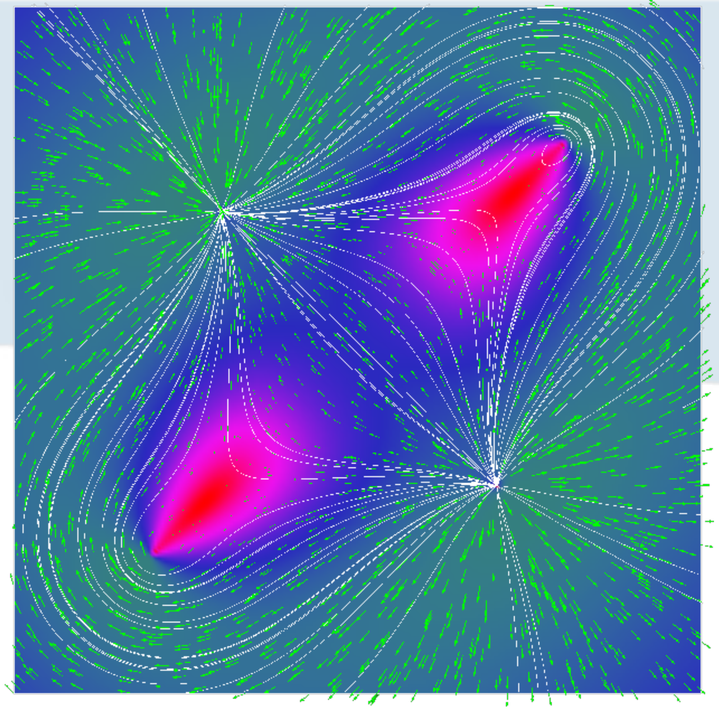
Flow orientation

- Divergence of the flow, for instance
  - $f : \mathbb{R}^3 \to \mathbb{R}^3$



[http://www.deepakantony.com/]

Flow orientation



- Divergence of the flow, for instance
  - $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$

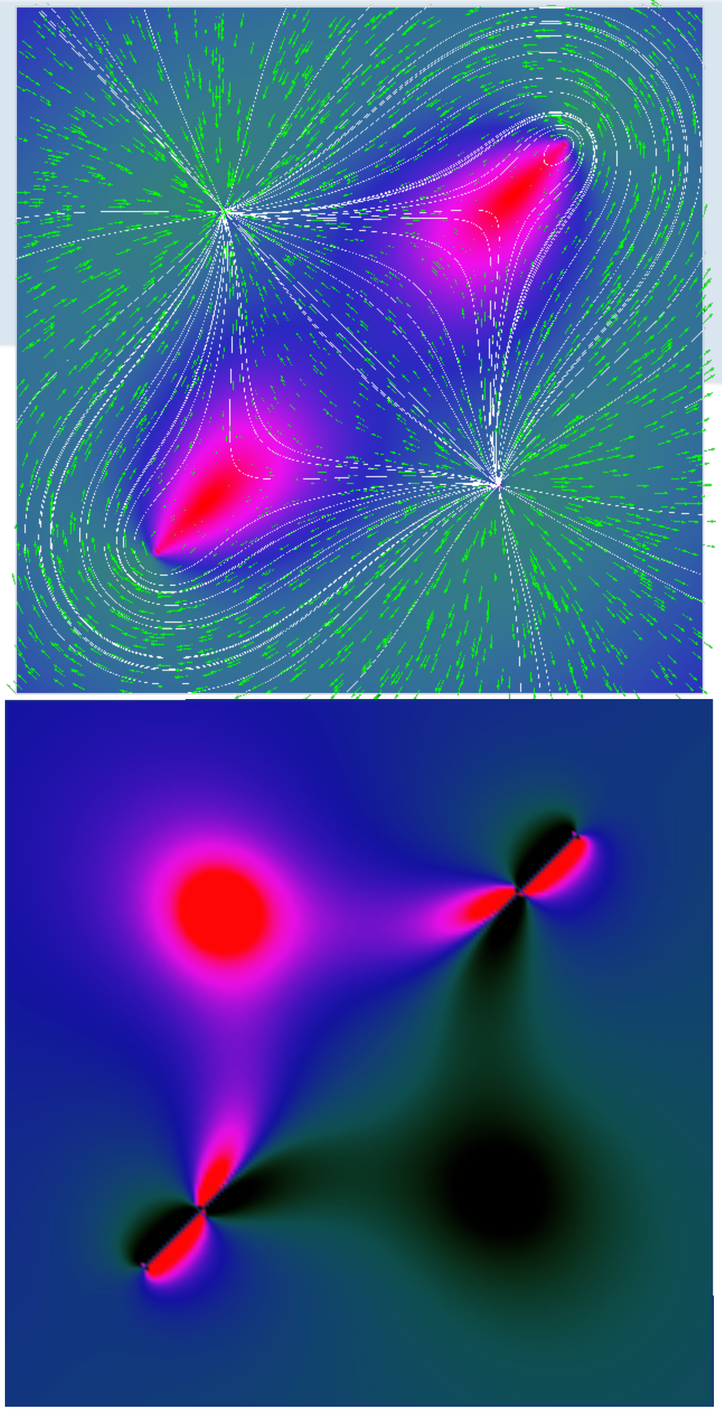
$$div f = \nabla . f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}$$

[http://www.deepakantony.com/]

Flow orientation

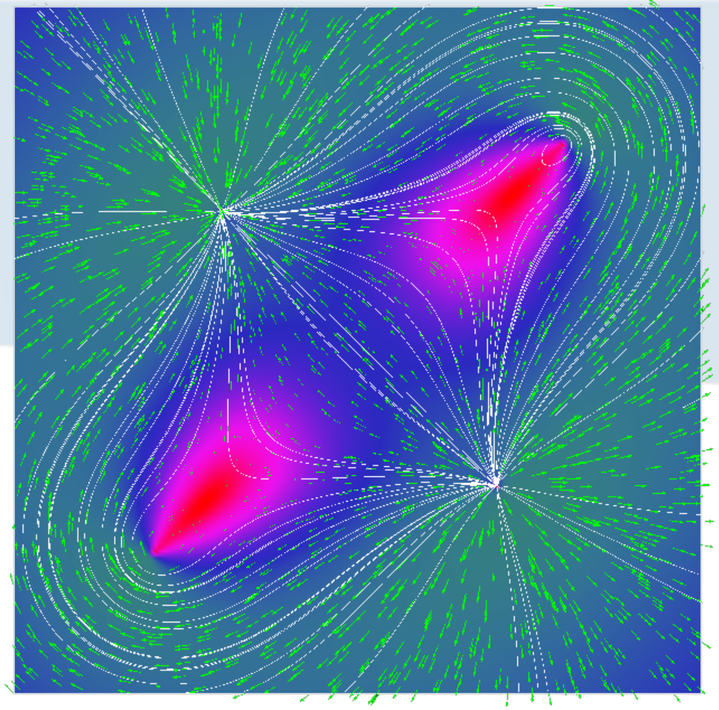
- Divergence of the flow, for instance
  - $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$

$$div f = \nabla . f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}$$



[http://www.deepakantony.com/]

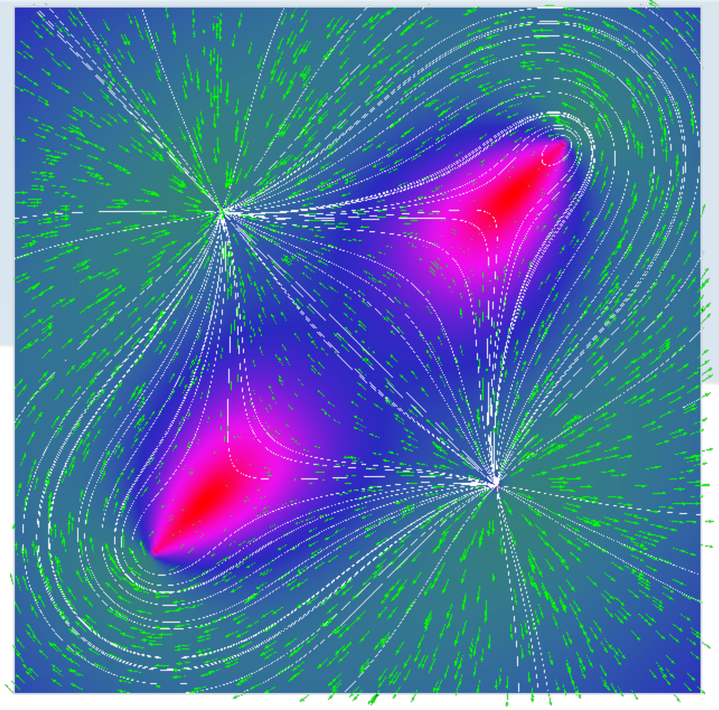
Angular speed



[http://www.deepakantony.com/]

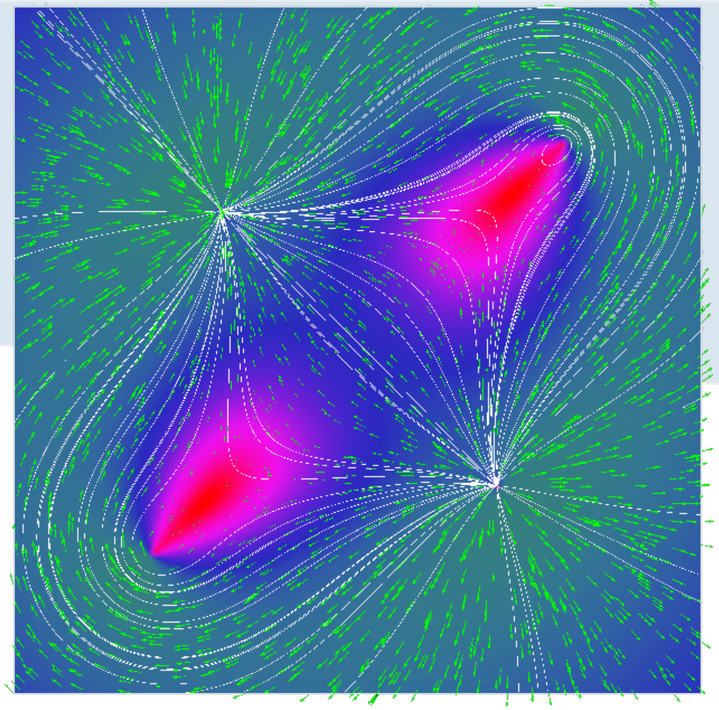
Angular speed

- Magnitude of the curl, for instance
  - $f : \mathbb{R}^3 \to \mathbb{R}^3$



[http://www.deepakantony.com/]

Angular speed



- Magnitude of the curl, for instance
  - $f : \mathbb{R}^3 \to \mathbb{R}^3$

$$\nabla \times f =$$

$$(\frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z})\vec{i} + (\frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x})\vec{j}$$

$$+(\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y})\vec{k}$$

[http://www.deepakantony.com/]

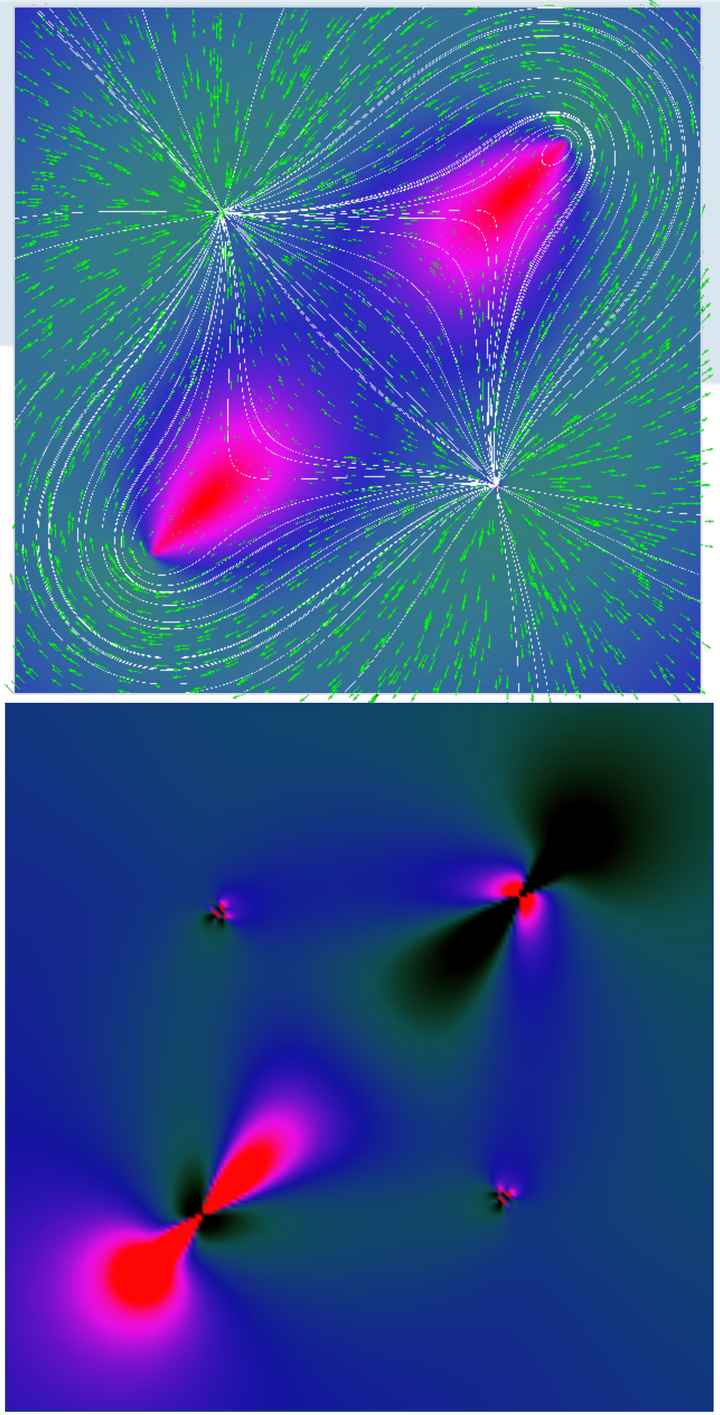
Angular speed

- Magnitude of the curl, for instance
  - $f : \mathbb{R}^3 \to \mathbb{R}^3$

$$\nabla \times f =$$

$$(\frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z})\vec{i} + (\frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x})\vec{j}$$

$$+(\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y})\vec{k}$$



[http://www.deepakantony.com/]

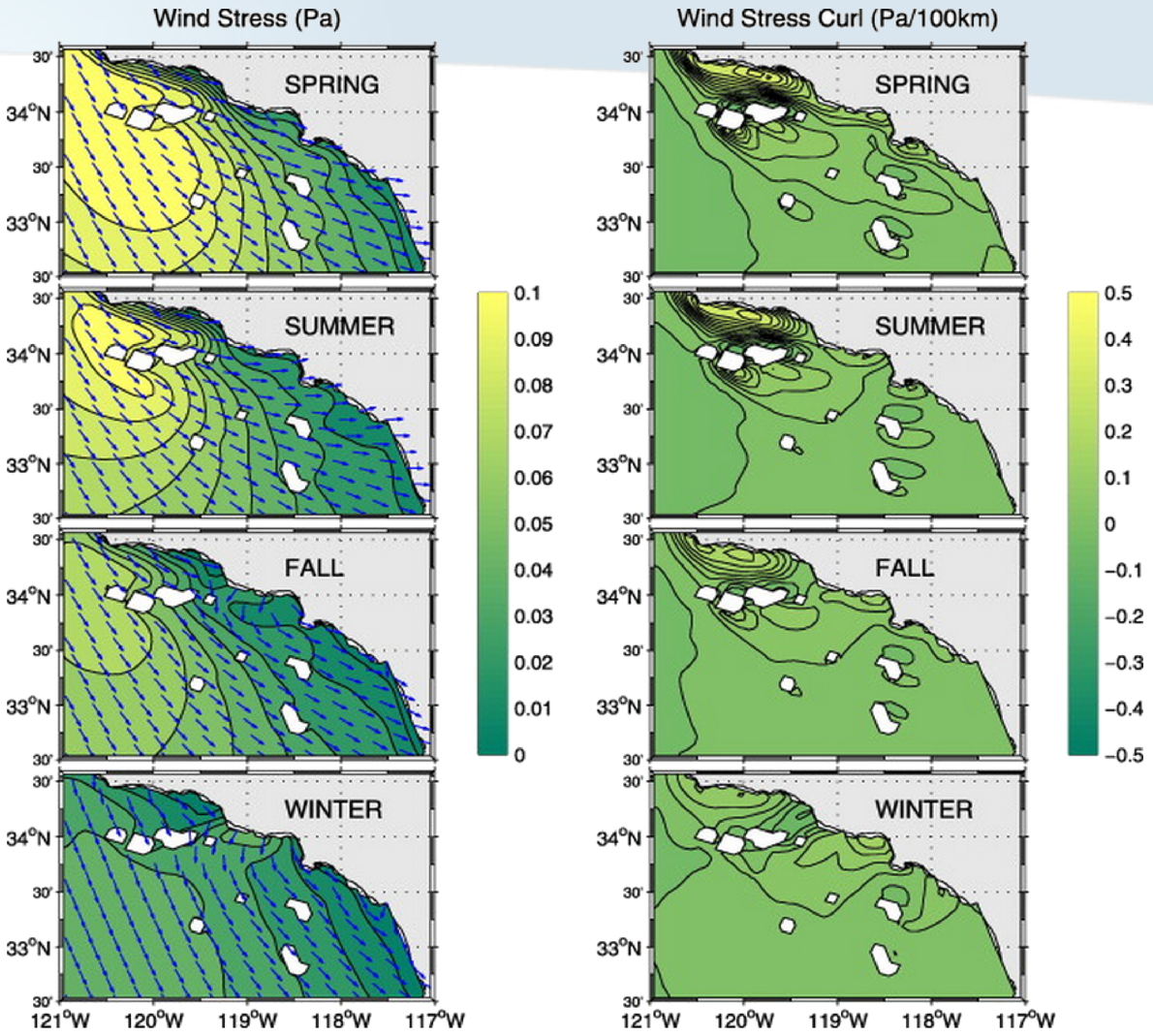
Angular speed

- Magnitude of the curl, for instance
  - $f : \mathbb{R}^3 \to \mathbb{R}^3$

$$\nabla \times f =$$

$$(\frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z})\vec{i} + (\frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x})\vec{j}$$

$$+(\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y})\vec{k}$$



[Dong09]

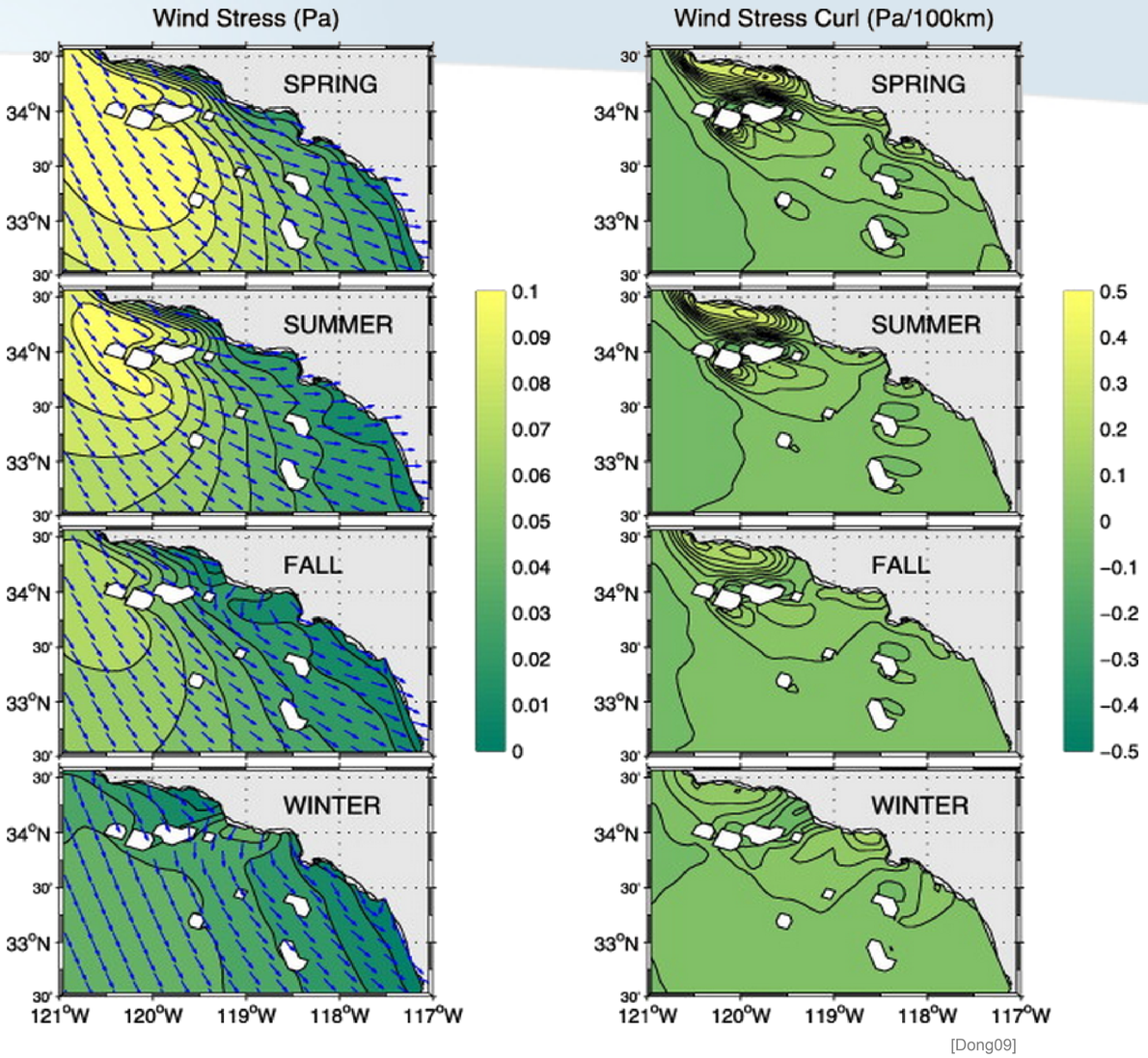
Angular speed

- Magnitude of the curl, for instance
  - $f : \mathbb{R}^3 \to \mathbb{R}^3$

$$\nabla \times f =$$

$$(\frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z})\vec{i} + (\frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x})\vec{j} + (\frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y})\vec{k}$$

  - On PL-manifolds?



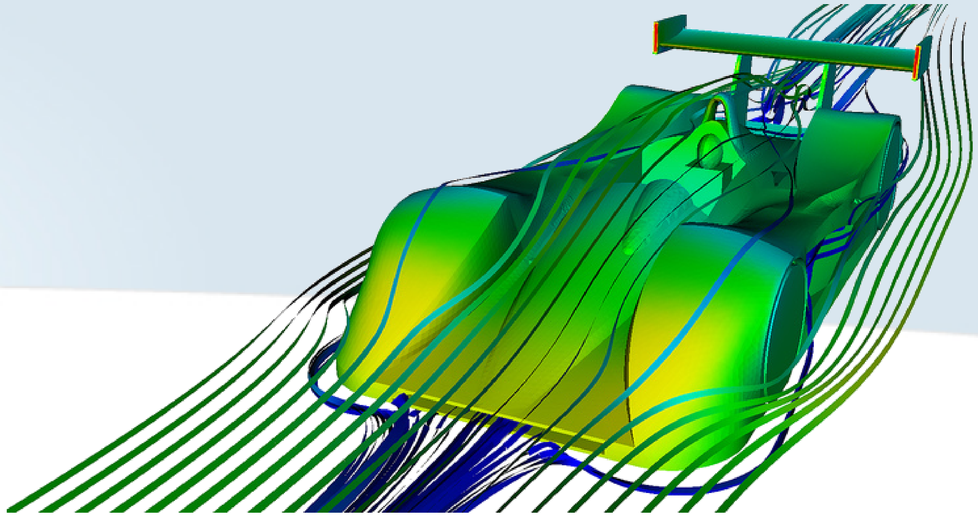
[Dong09]

So far

- Extract geometrical features
  - Streamlines

So far

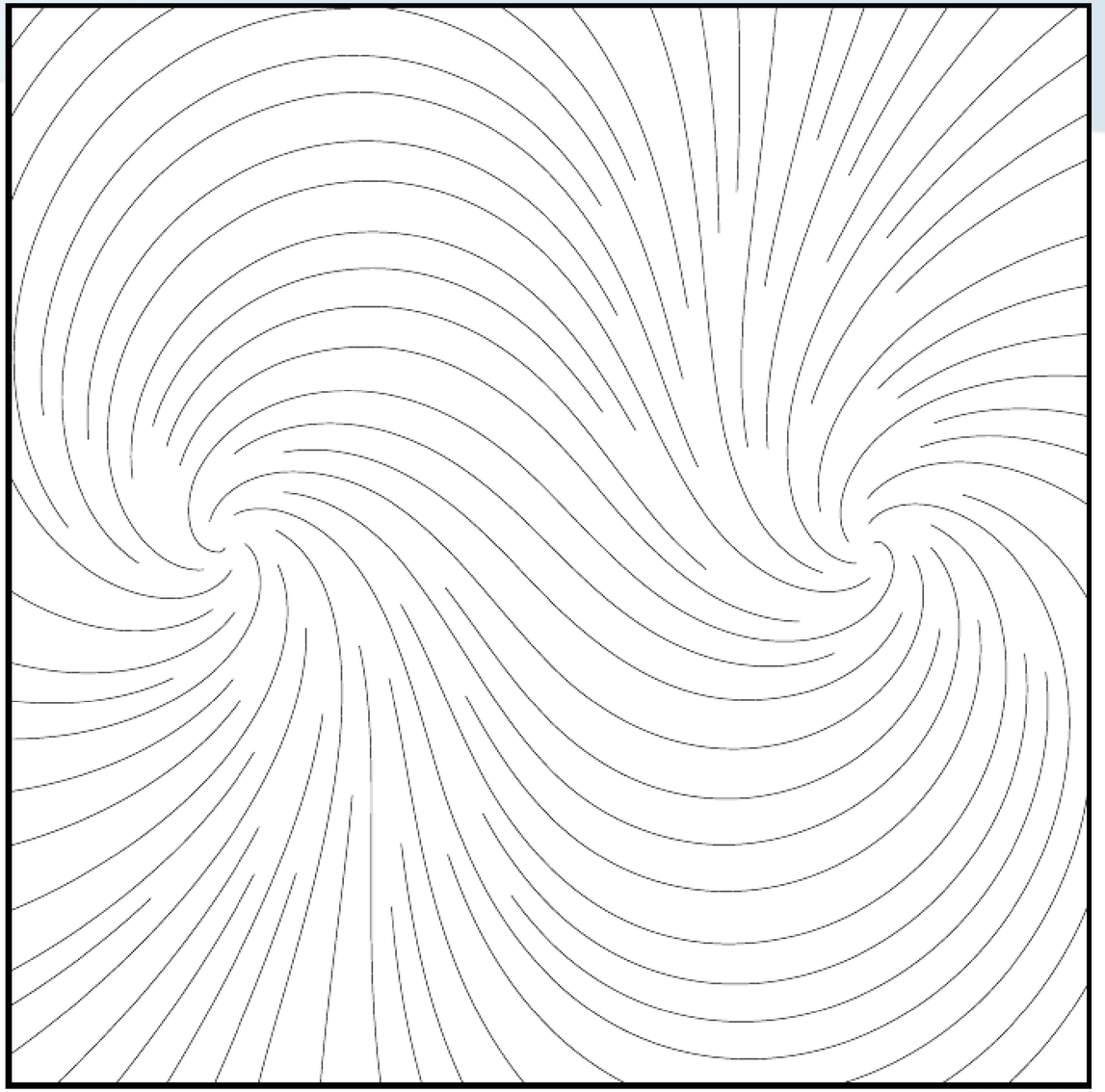
- Extract geometrical features
  - Streamlines



[Kitware]

So far

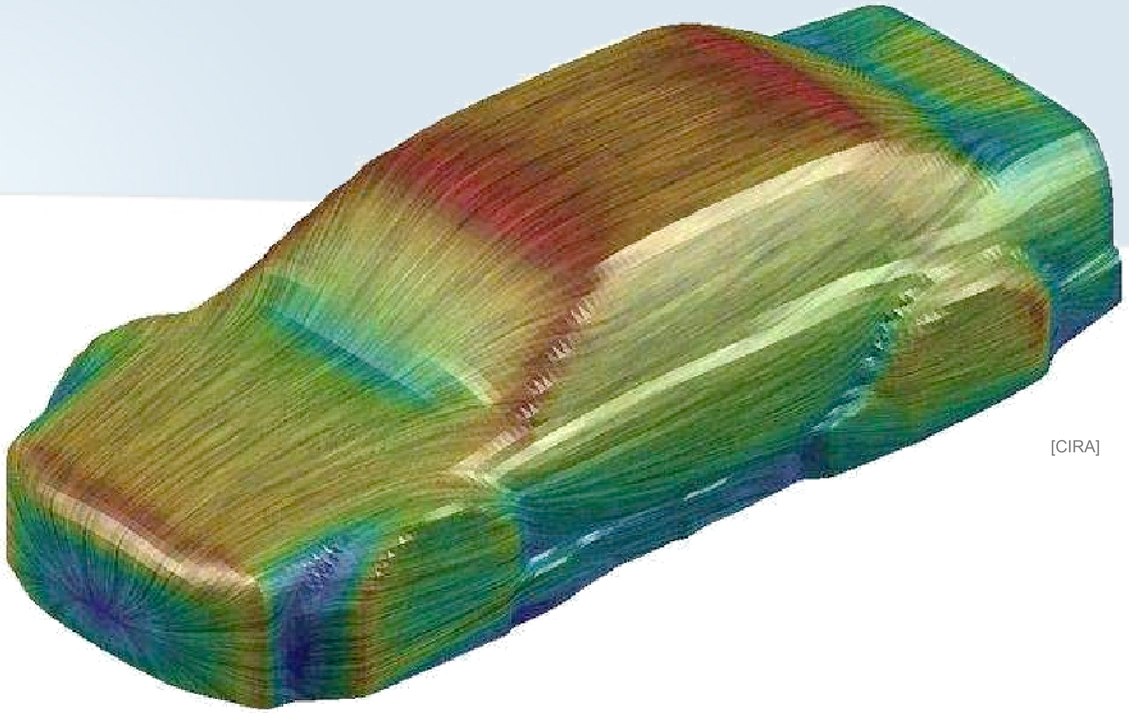
- Extract geometrical features
  - Streamlines
    - Seeding



[Jobard et al. 97]

So far

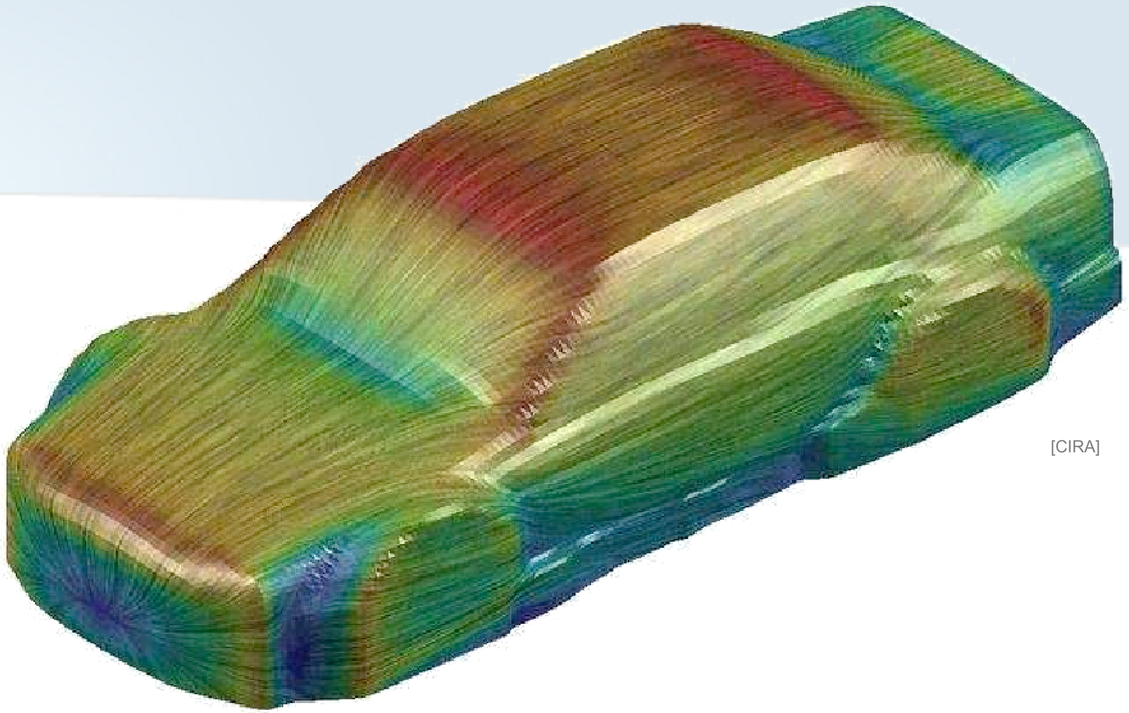
- Extract geometrical features
  - Streamlines
    - Seeding, Line Integral Convolution



[CIRA]

So far

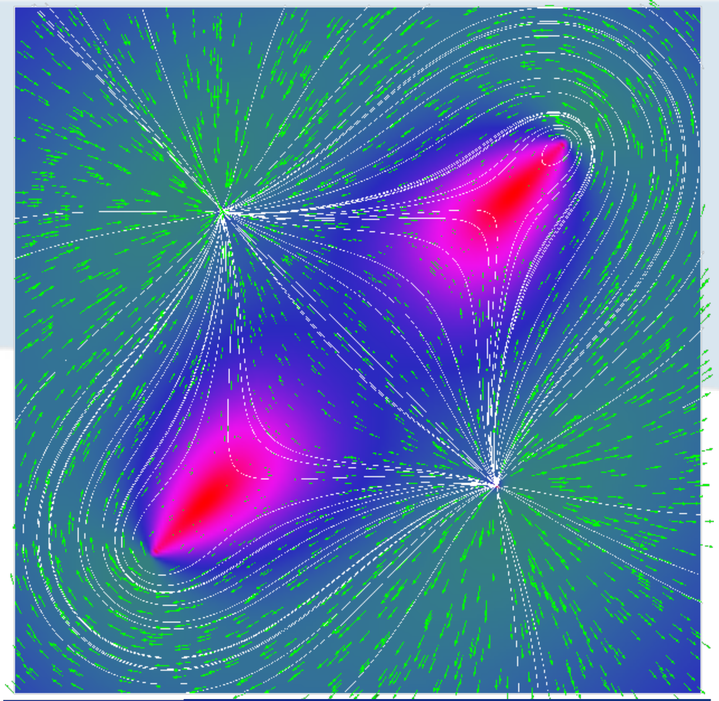
- Extract geometrical features
  - Streamlines
    - Seeding, Line Integral Convolution
- Extract geometrical measures

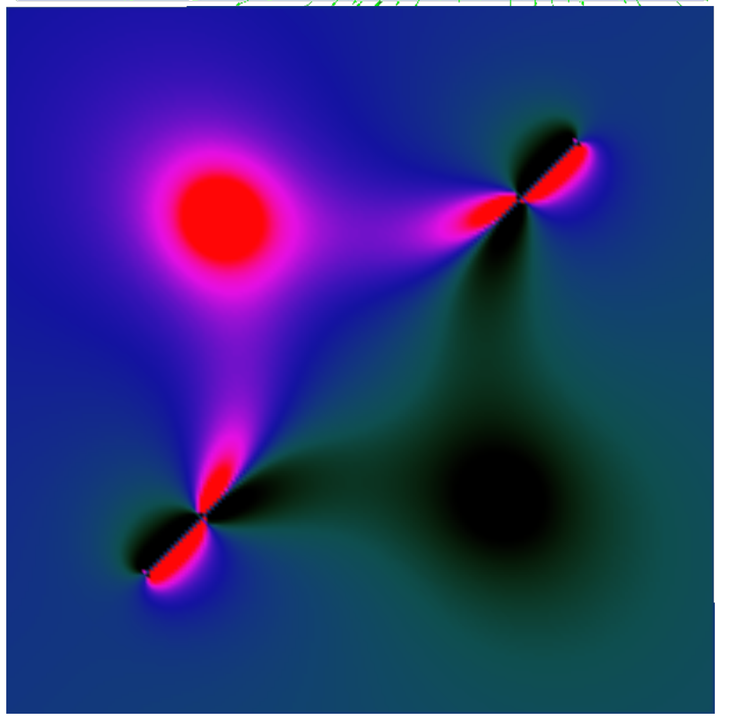


[CIRA]

So far

- Extract geometrical features
  - Streamlines
    - Seeding, Line Integral Convolution
- Extract geometrical measures
  - Magnitude, orientation, angular speed, distortion

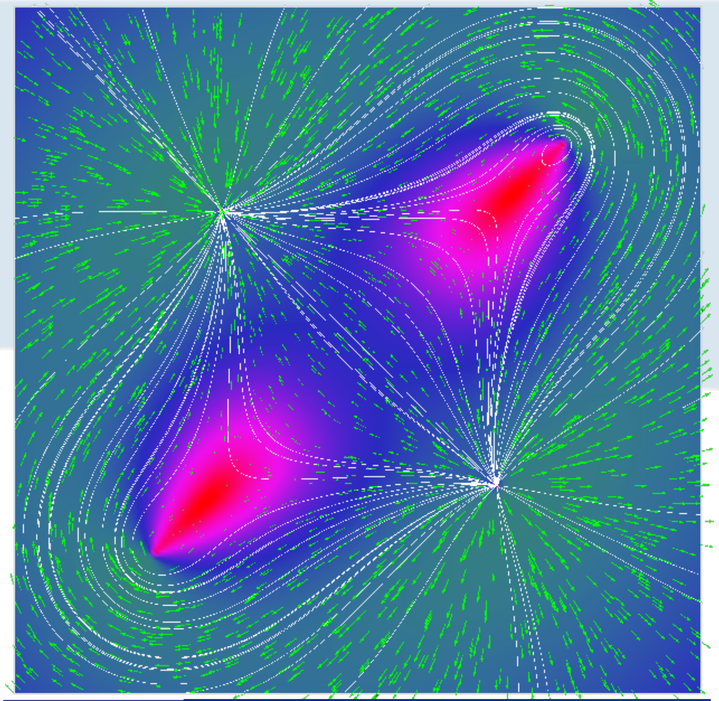


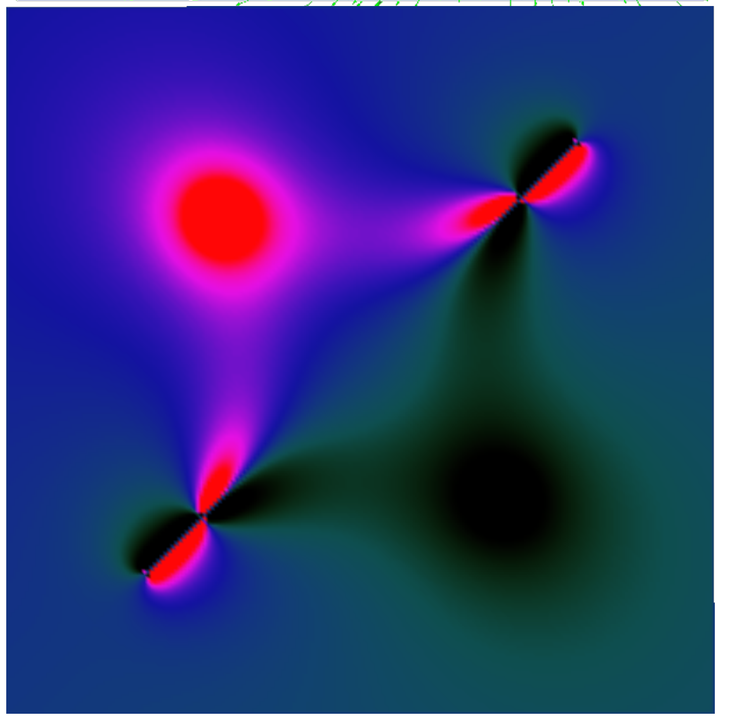


[http://www.deepakantony.com/]

So far

- Extract geometrical features
  - Streamlines
    - Seeding, Line Integral Convolution
    - **Where do they end/start?**
- Extract geometrical measures
  - Magnitude, orientation, angular speed, distortion



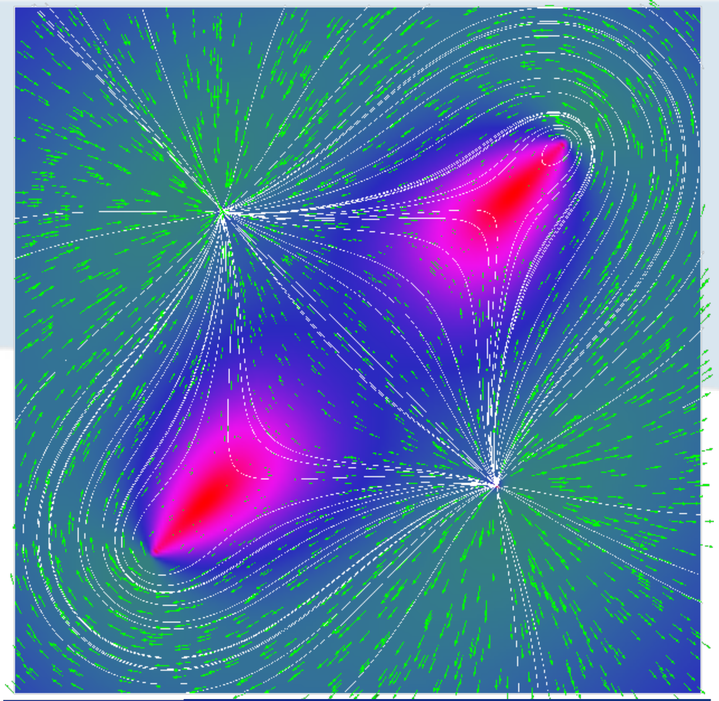


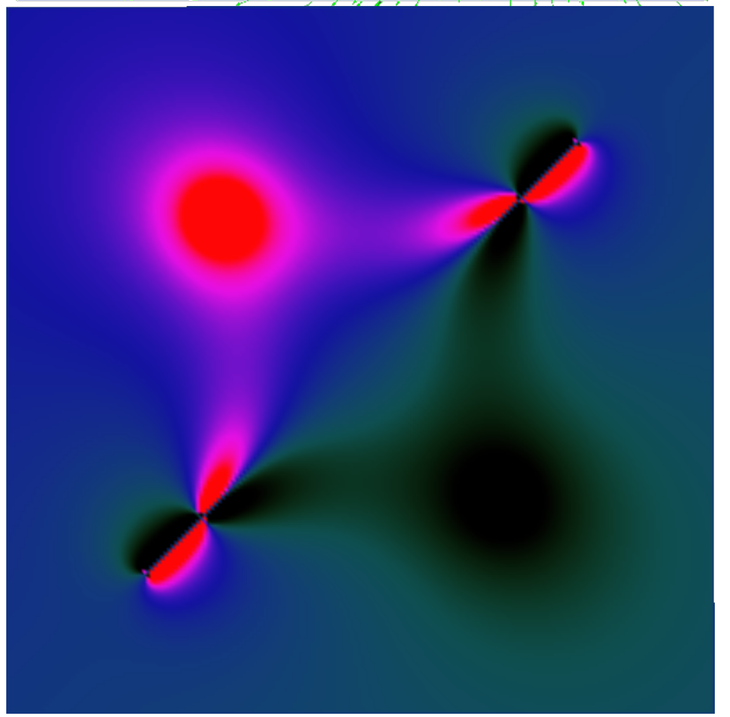
[http://www.deepakantony.com/]

So far

- Extract geometrical features
  - Streamlines
    - Seeding, Line Integral Convolution
    - **Where do they end/start?**
- Extract geometrical measures
  - Magnitude, orientation, angular speed, distortion

- Vector field topology



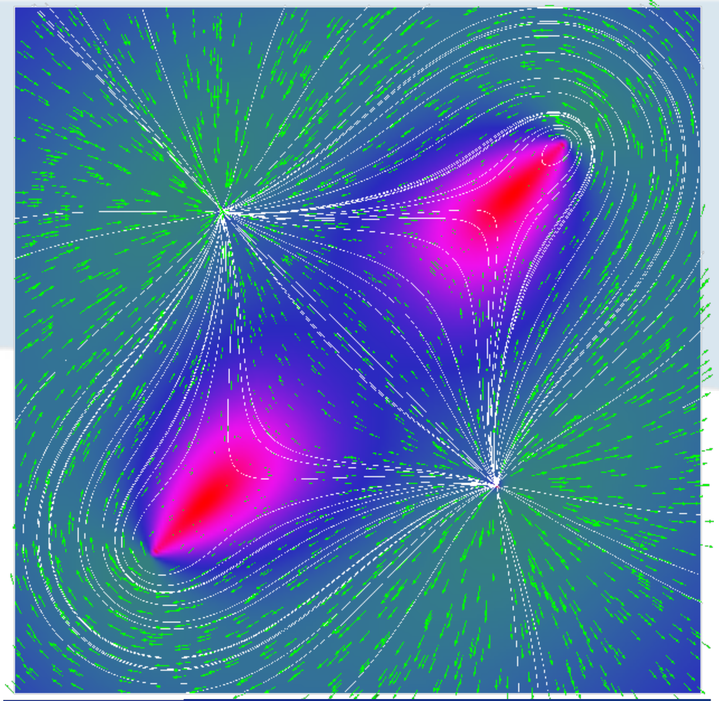


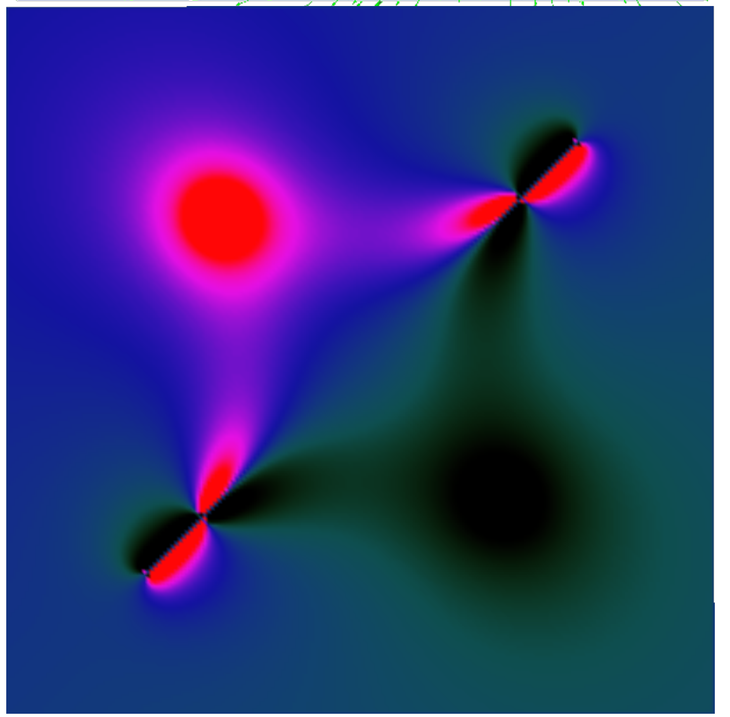
[http://www.deepakantony.com/]

So far

- Extract geometrical features
  - Streamlines
    - Seeding, Line Integral Convolution
    - **Where do they end/start?**
- Extract geometrical measures
  - Magnitude, orientation, angular speed, distortion

- Vector field topology
  - Summarizes all this information





[http://www.deepakantony.com/]

Gradient field topology

Gradient field topology

- For instance
  - $\mathcal{D} \subset \mathbb{R}^n$
    - $f : \mathbb{R}^n \to \mathbb{R}^n$

Gradient field topology

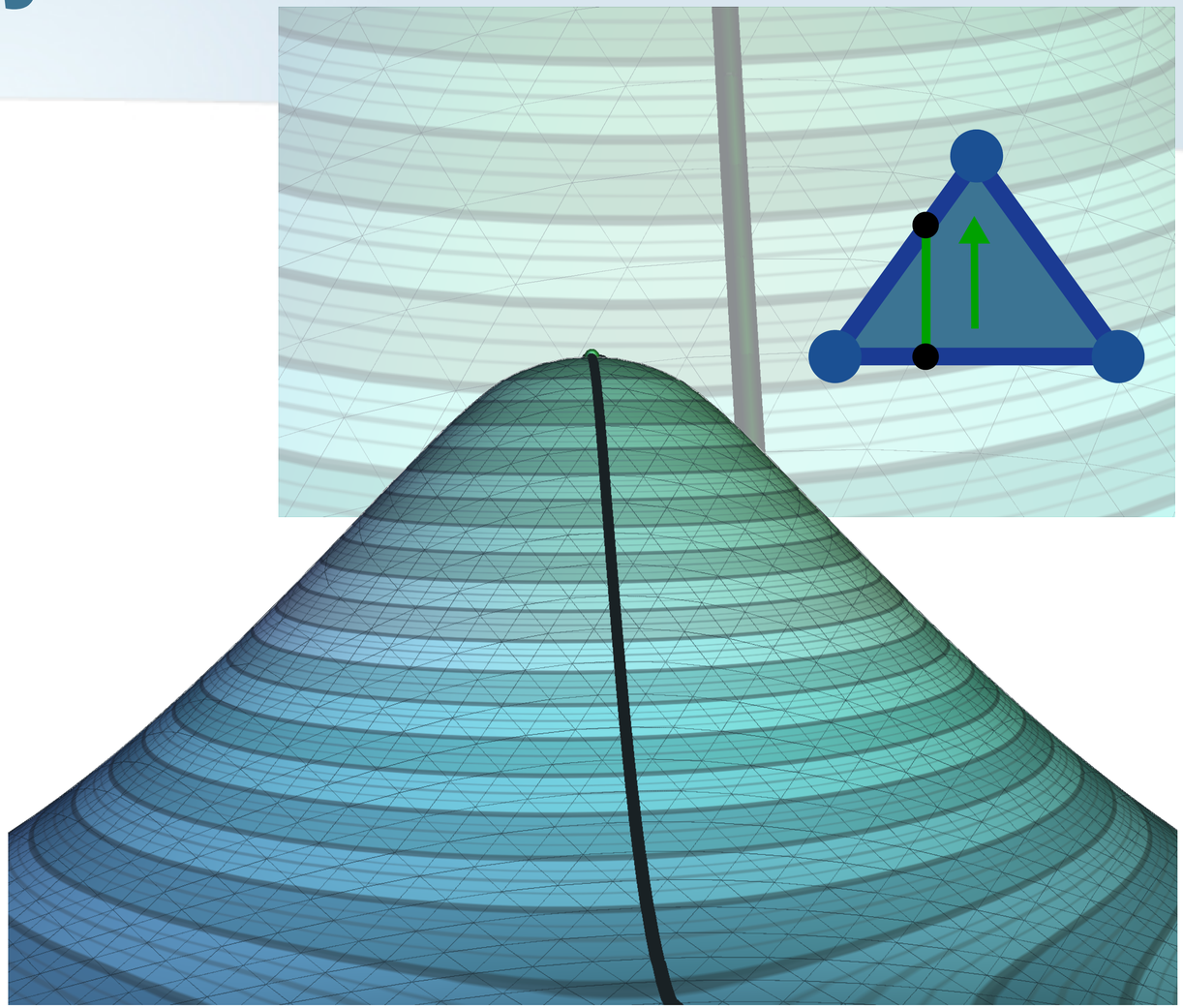
- For instance
  - $\mathcal{D} \subset \mathbb{R}^n$
    - $f : \mathbb{R}^n \to \mathbb{R}^n$
  - Gradient field, example
    - $g : \mathbb{R}^n \to \mathbb{R}$
    - $f = \nabla g = (\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \ldots, \frac{\partial g}{\partial x_n})$

Gradient field topology

- For instance
  - $\mathcal{D} \subset \mathbb{R}^n$
    - $f : \mathbb{R}^n \to \mathbb{R}^n$
  - Gradient field, example
    - $g : \mathbb{R}^n \to \mathbb{R}$
    - $f = \nabla g = (\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \ldots, \frac{\partial g}{\partial x_n})$
- PL scalar fields
  - Piecewise constant gradient field

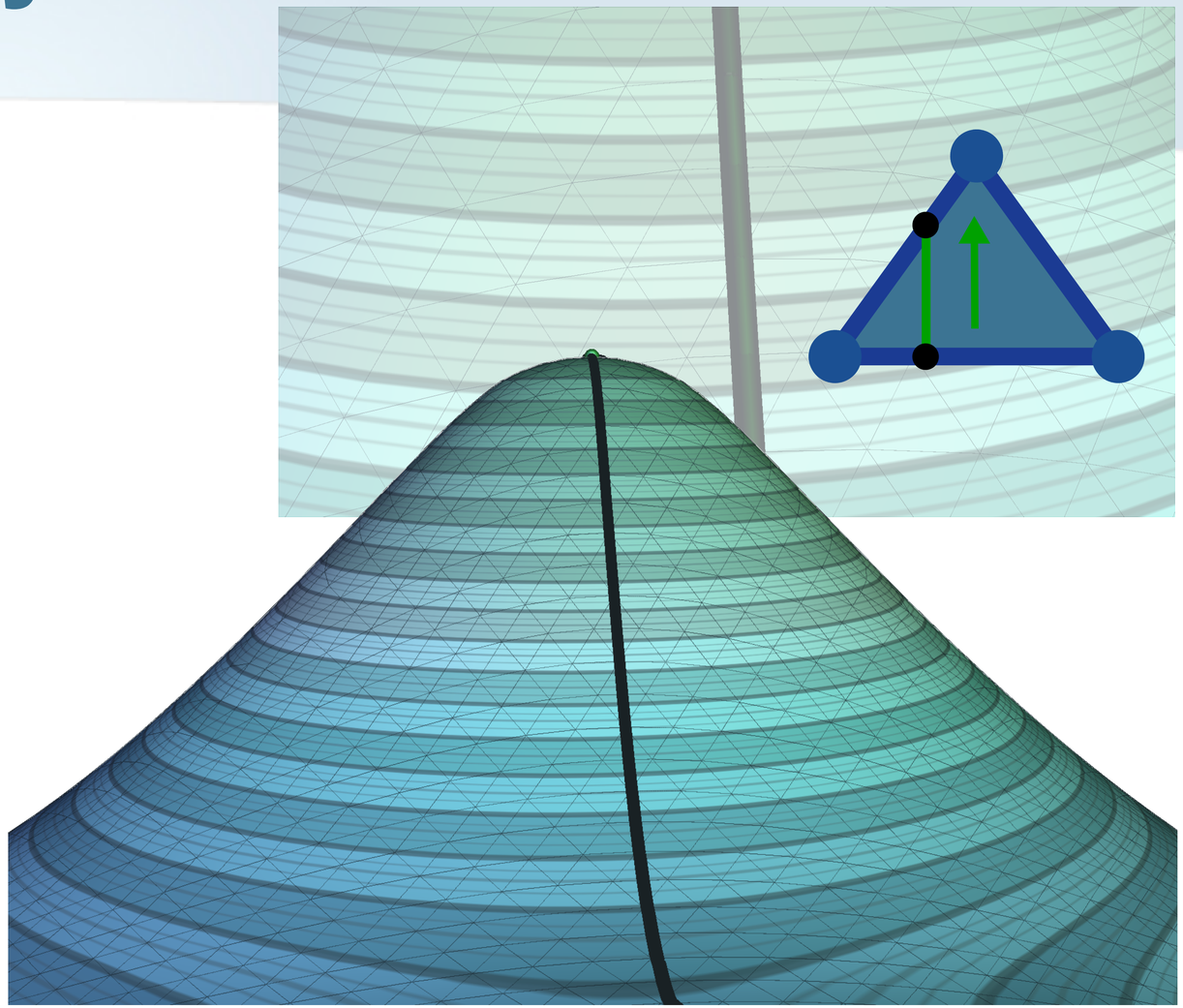
Gradient field topology

- For instance
  - $\mathcal{D} \subset \mathbb{R}^n$
    - $f : \mathbb{R}^n \to \mathbb{R}^n$
  - Gradient field, example
    - $g : \mathbb{R}^n \to \mathbb{R}$
    - $f = \nabla g = (\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \ldots, \frac{\partial g}{\partial x_n})$
- PL scalar fields
  - Piecewise constant gradient field



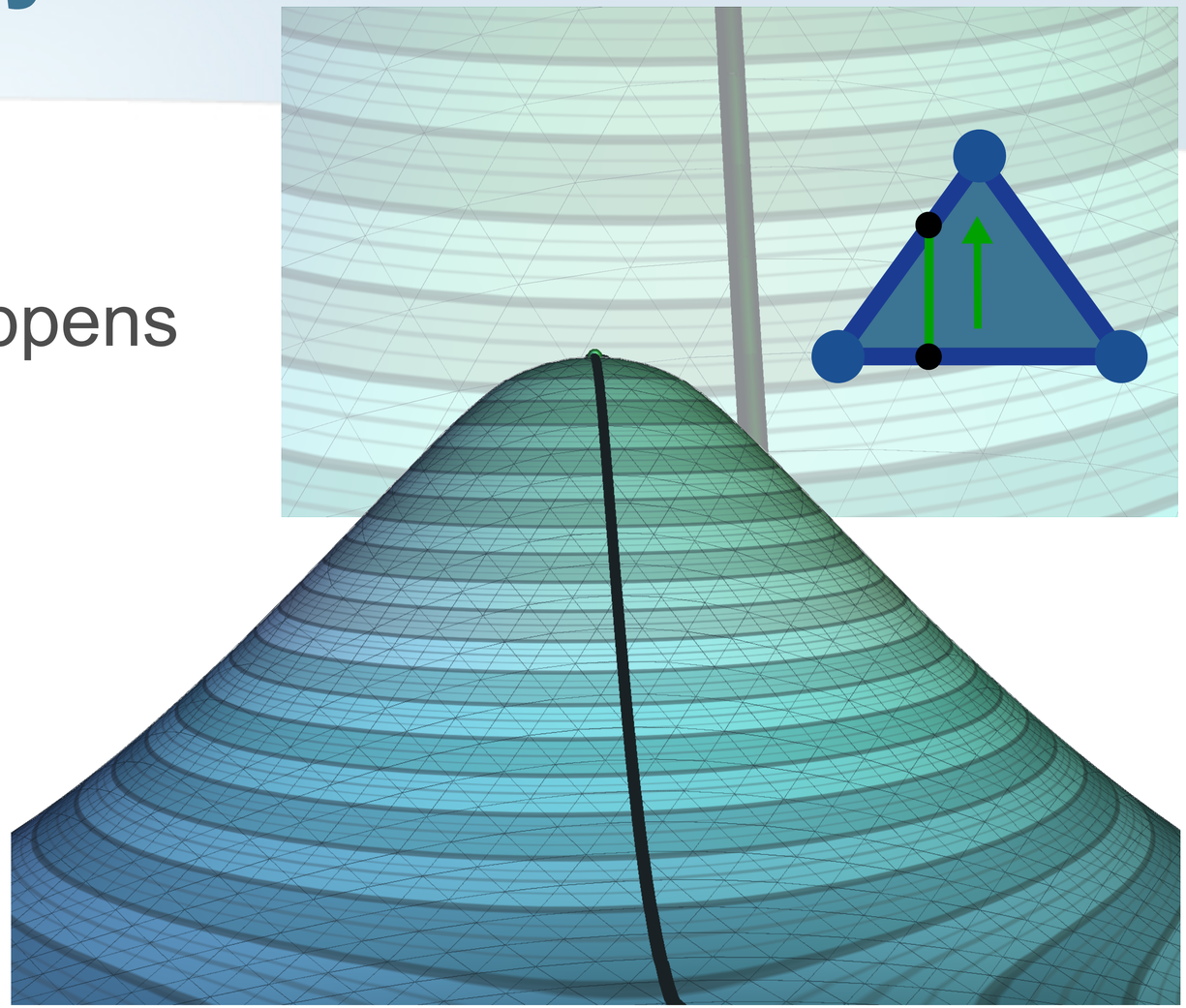
Gradient field topology

- Intuition of critical points



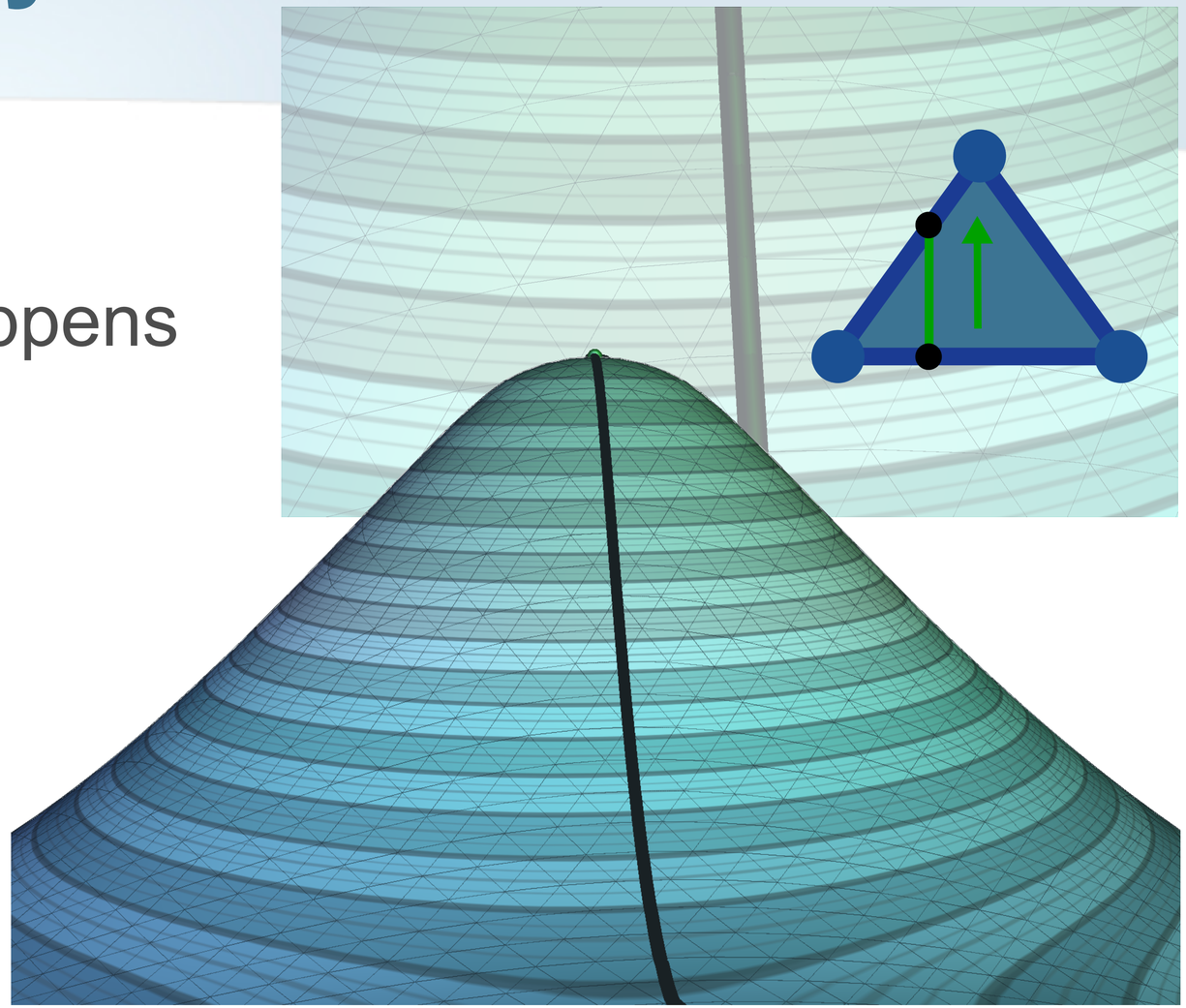
Gradient field topology

- Intuition of critical points
  - Points where something critical happens



Gradient field topology

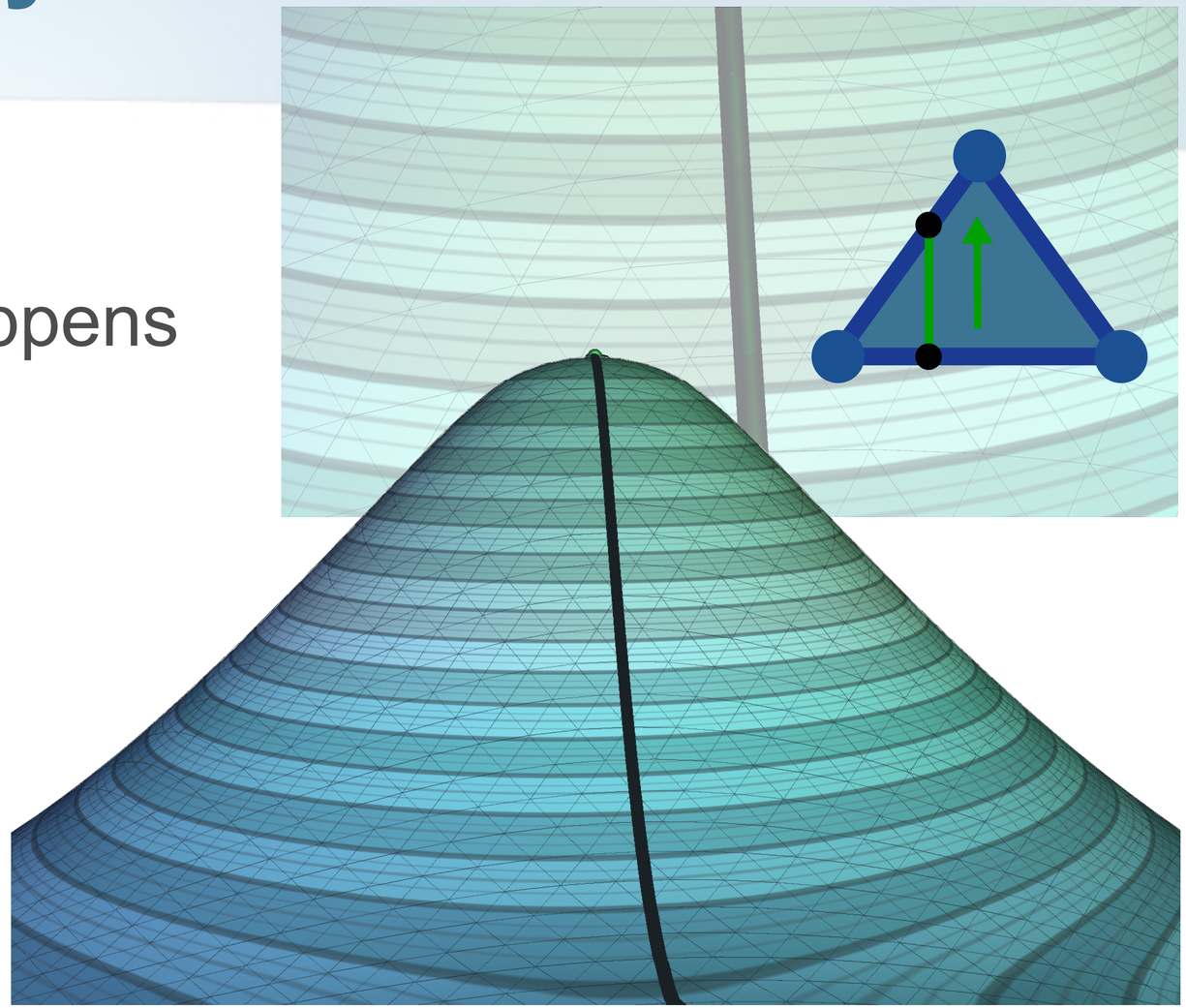
- Intuition of critical points
  - Points where something critical happens
  - Where the flow stops



Gradient field topology

- Intuition of critical points
  - Points where something critical happens
  - Where the flow stops

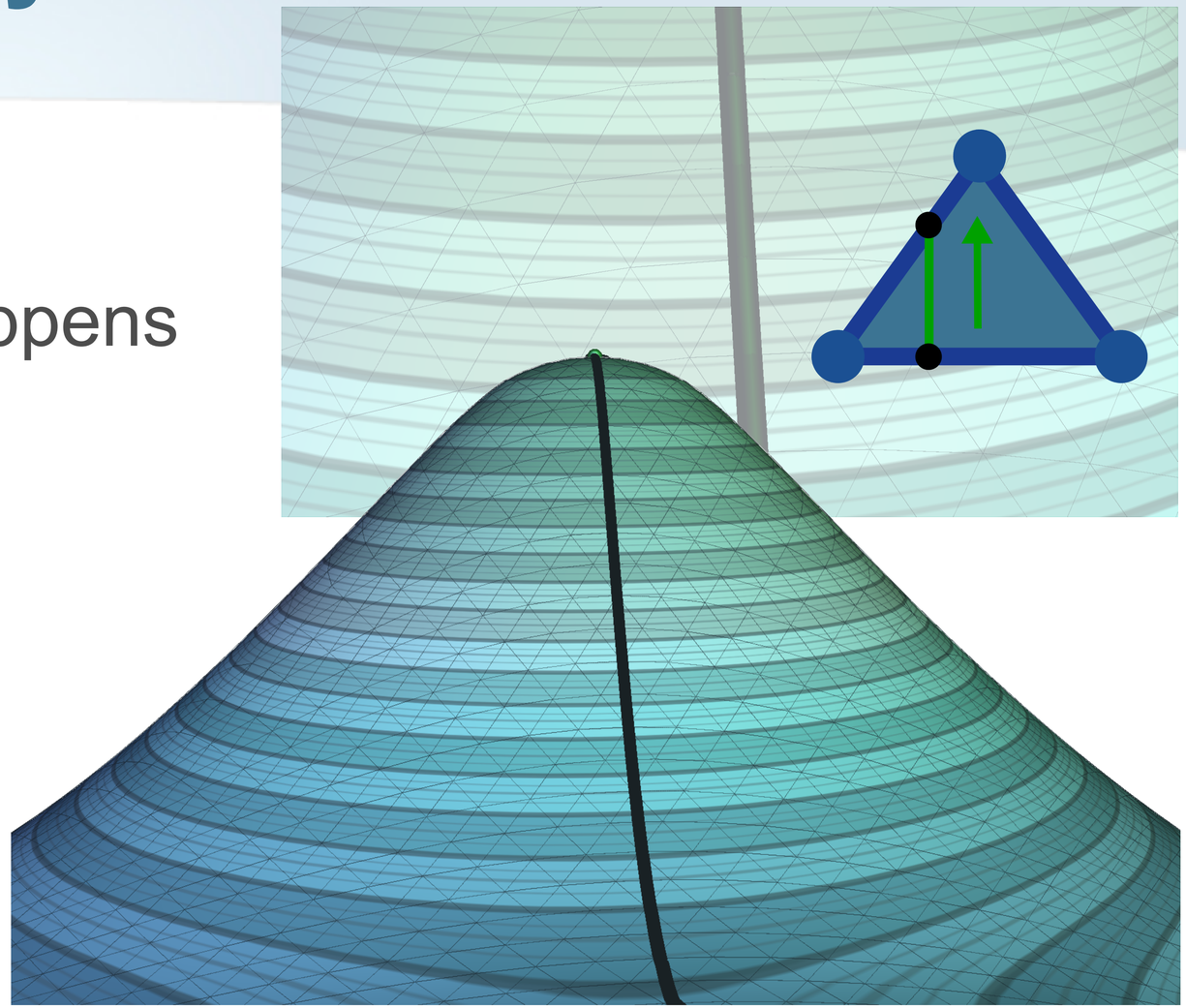
- Where does the gradient stops?



Gradient field topology

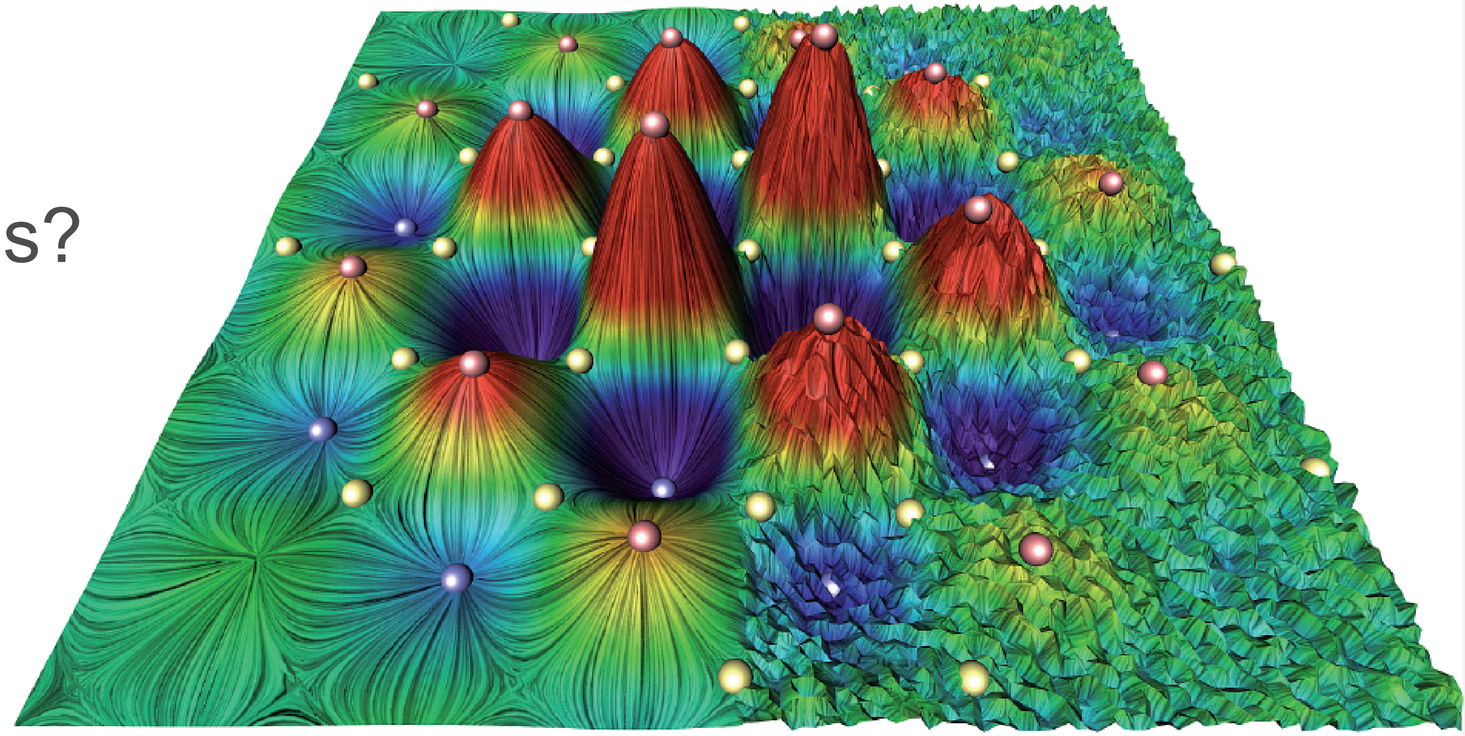
- Intuition of critical points
  - Points where something critical happens
  - Where the flow stops

- Where does the gradient stops?
  - At the critical points of g



Gradient field topology

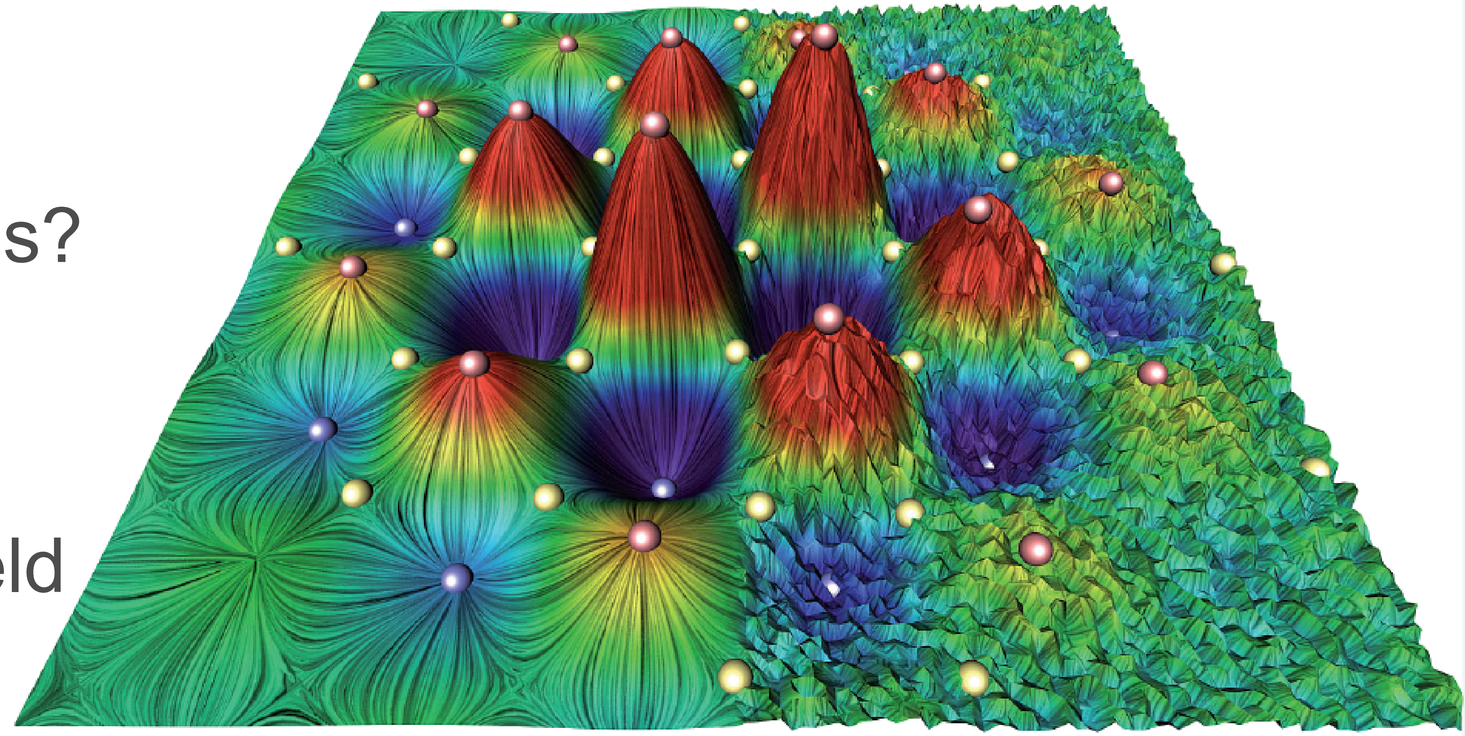
- Intuition of critical points
  - Points where something critical happens
  - Where the flow stops
- Where does the gradient stops?
  - At the critical points of g



[Reininghaus11]

Gradient field topology

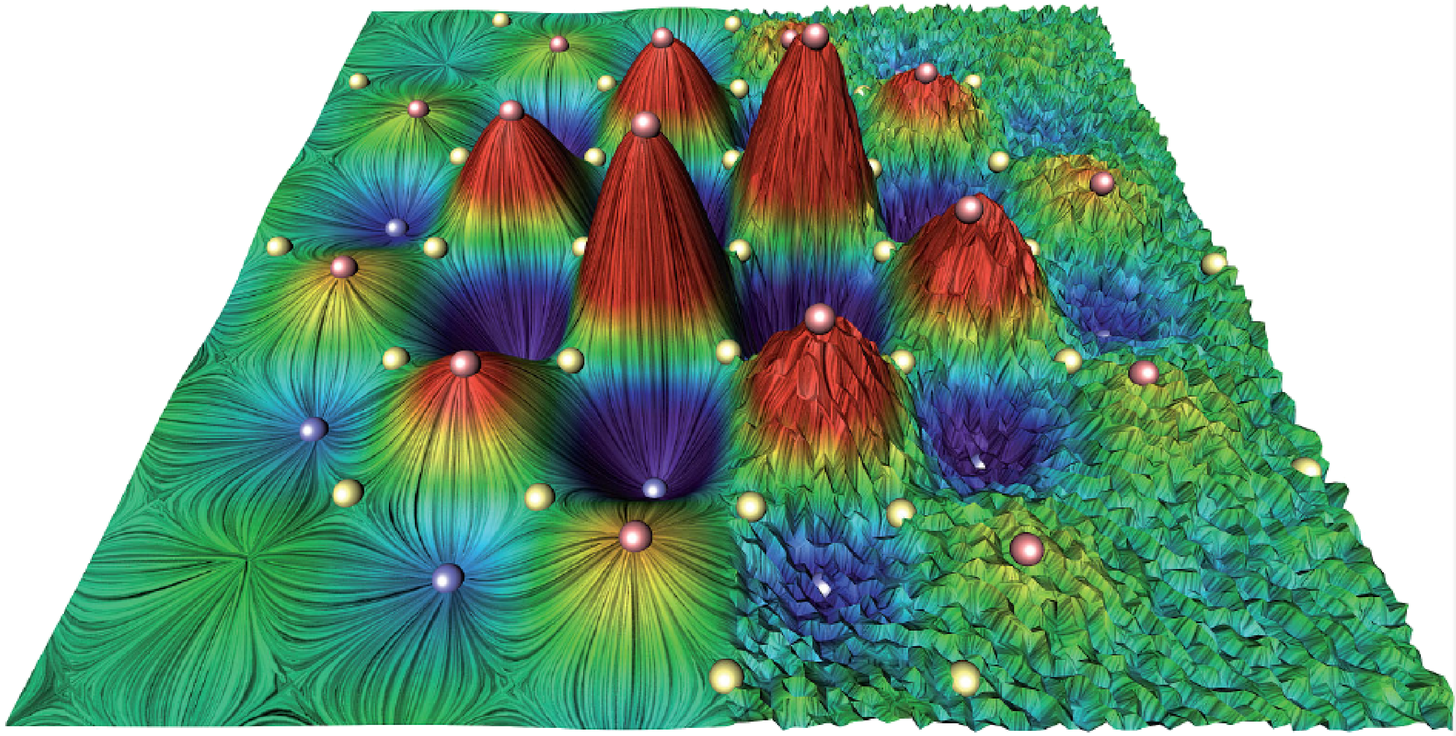
- Intuition of critical points
  - Points where something critical happens
  - Where the flow stops
- Where does the gradient stops?
  - At the critical points of g
- Critical points of a gradient field
  - Same as for scalar fields



[Reininghaus11]

Gradient field topology

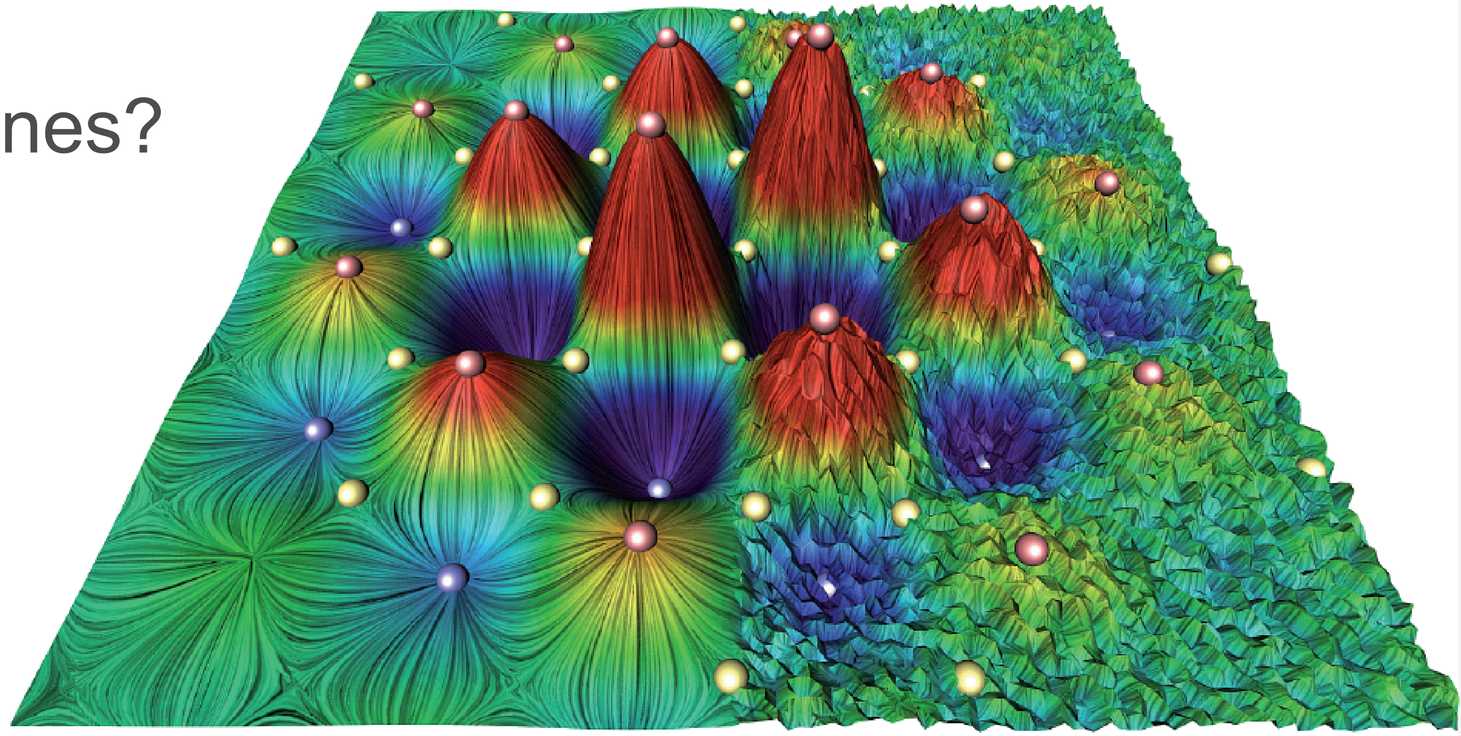
- Critical points of a vector field
  - Points where the magnitude vanishes



[Reininghaus11]

Gradient field topology

- Critical points of a vector field
  - Points where the magnitude vanishes
- What's the relation to streamlines?

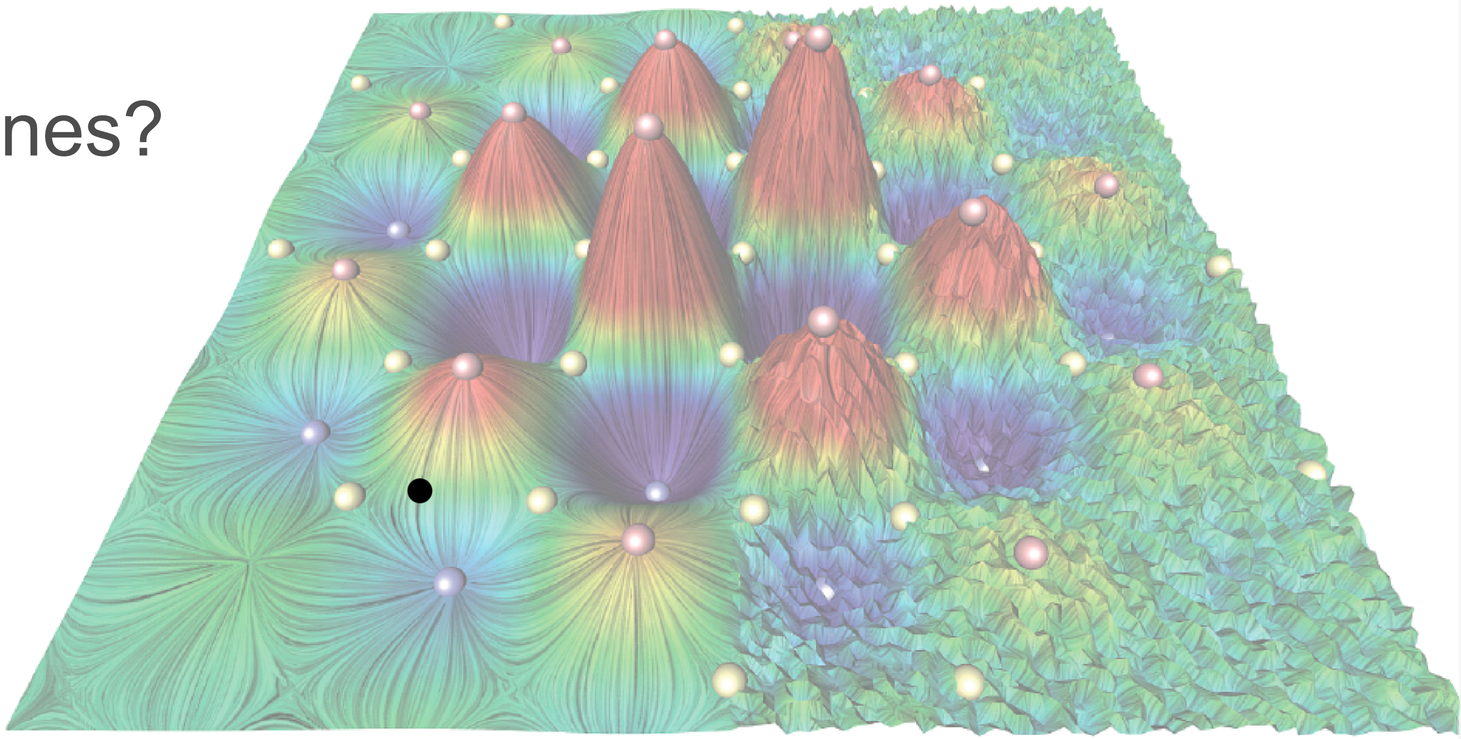


[Reininghaus11]

Gradient field topology

- Critical points of a vector field
  - Points where the magnitude vanishes

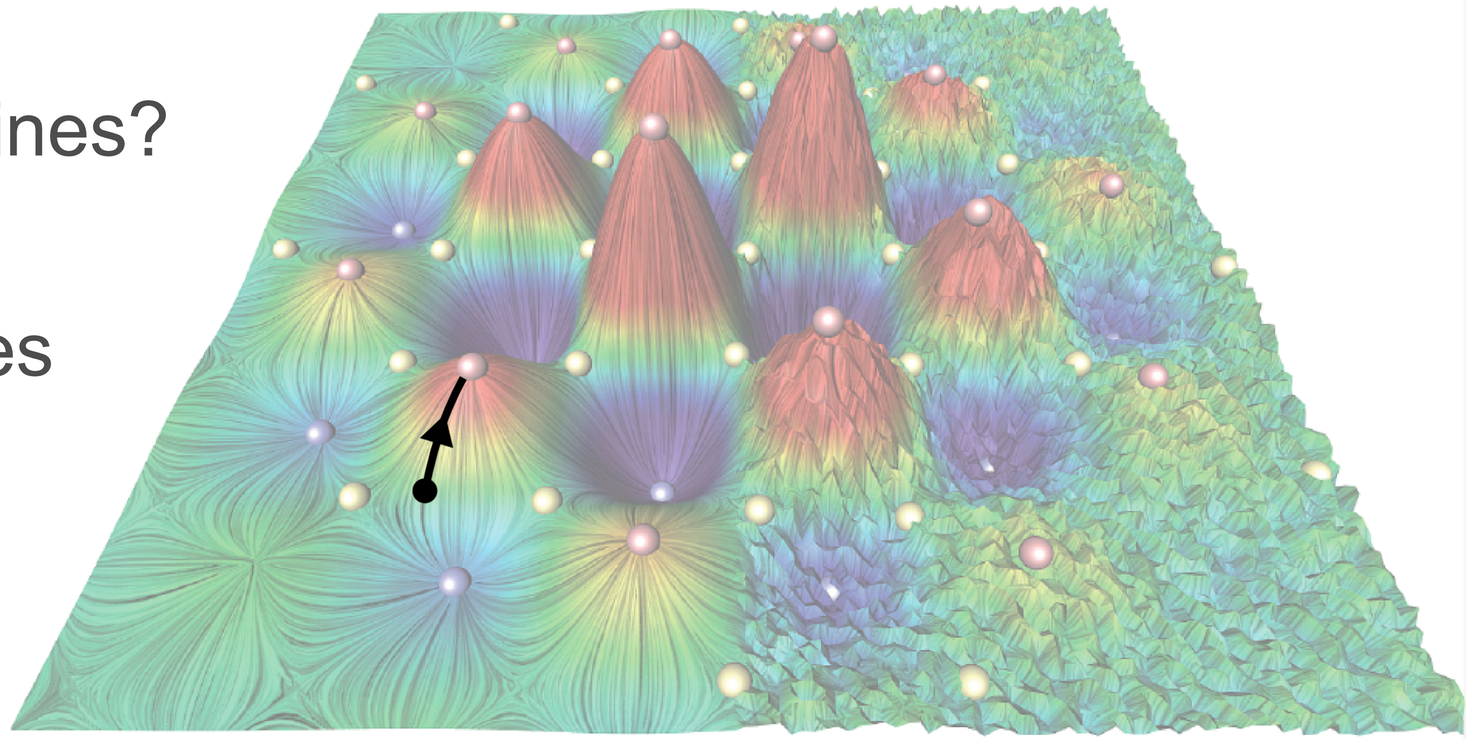
- What's the relation to streamlines?



[Reininghaus11]

Gradient field topology

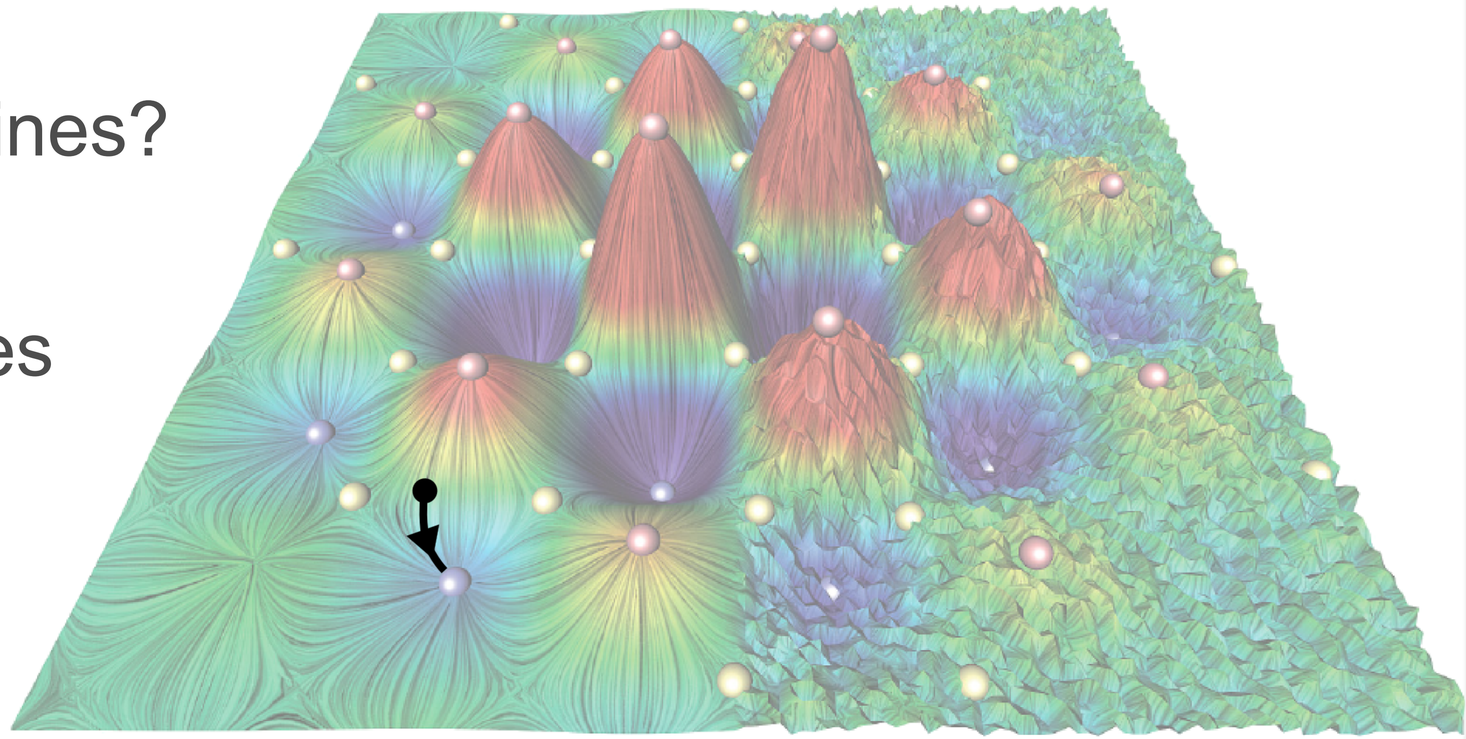
- Critical points of a vector field
  - Points where the magnitude vanishes
- What's the relation to streamlines?
  - On closed domains
  - Critical points are streamlines extremities



[Reininghaus11]

Gradient field topology

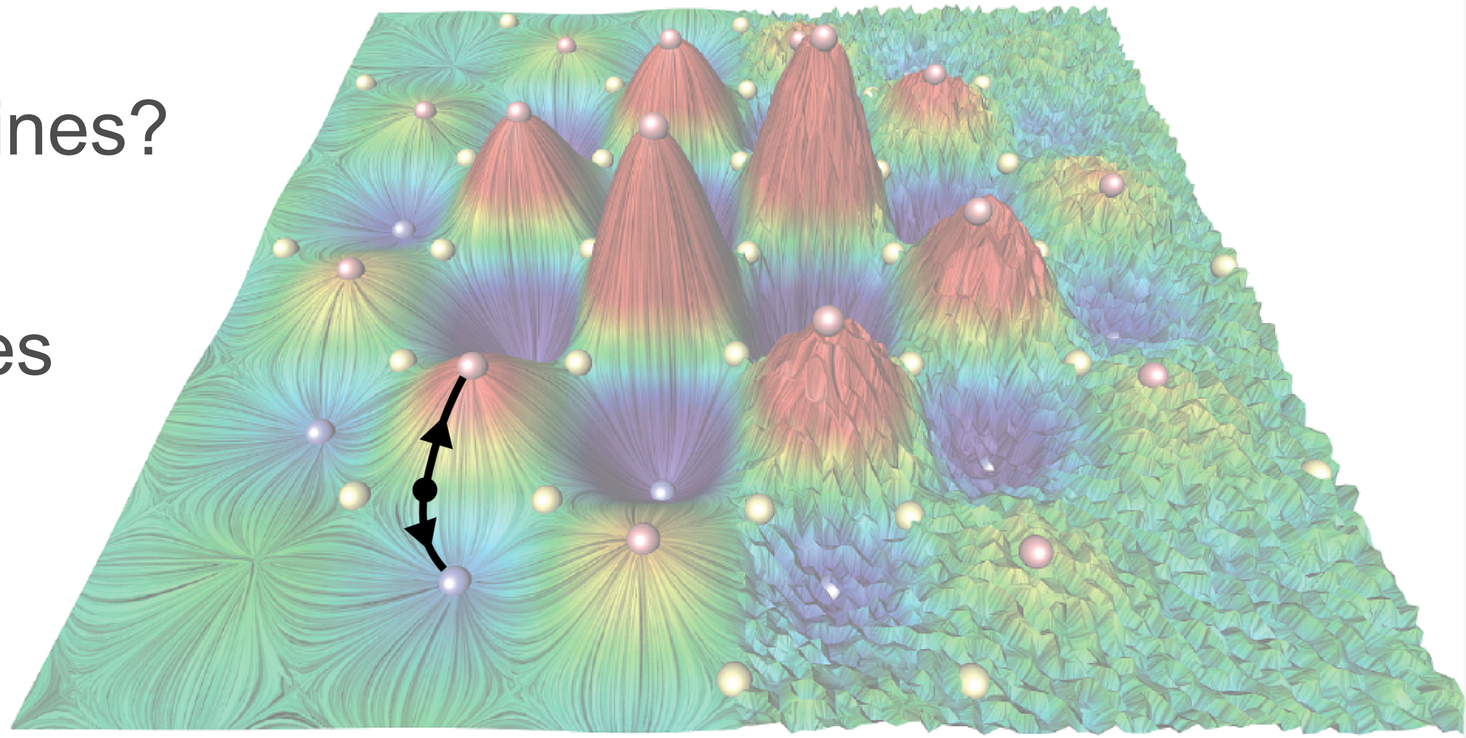
- Critical points of a vector field
  - Points where the magnitude vanishes
- What's the relation to streamlines?
  - On closed domains
  - Critical points are streamlines extremities
  - Forwards and backwards



[Reininghaus11]

Gradient field topology

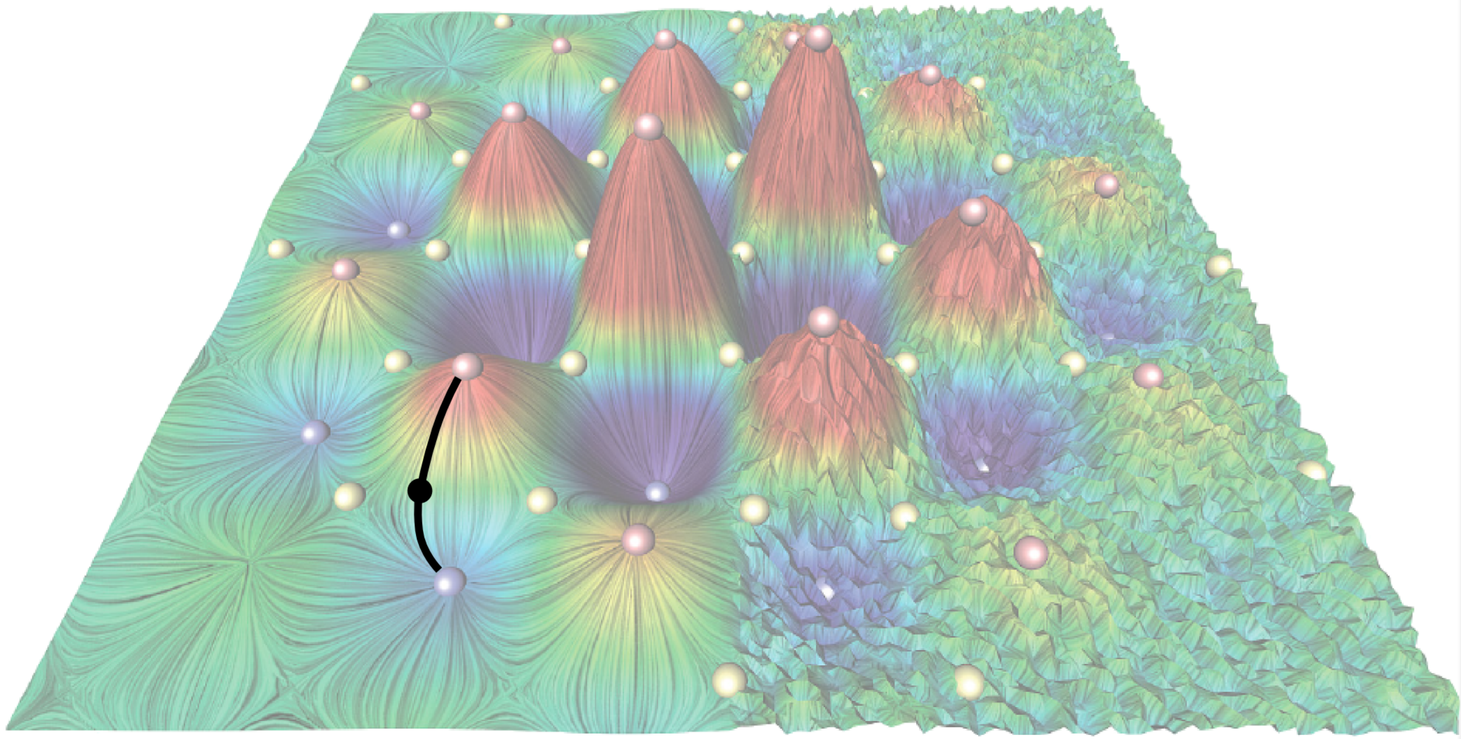
- Critical points of a vector field
  - Points where the magnitude vanishes
- What's the relation to streamlines?
  - On closed domains
  - Critical points are streamlines extremities
  - Forwards and backwards



[Reininghaus11]

Gradient field topology

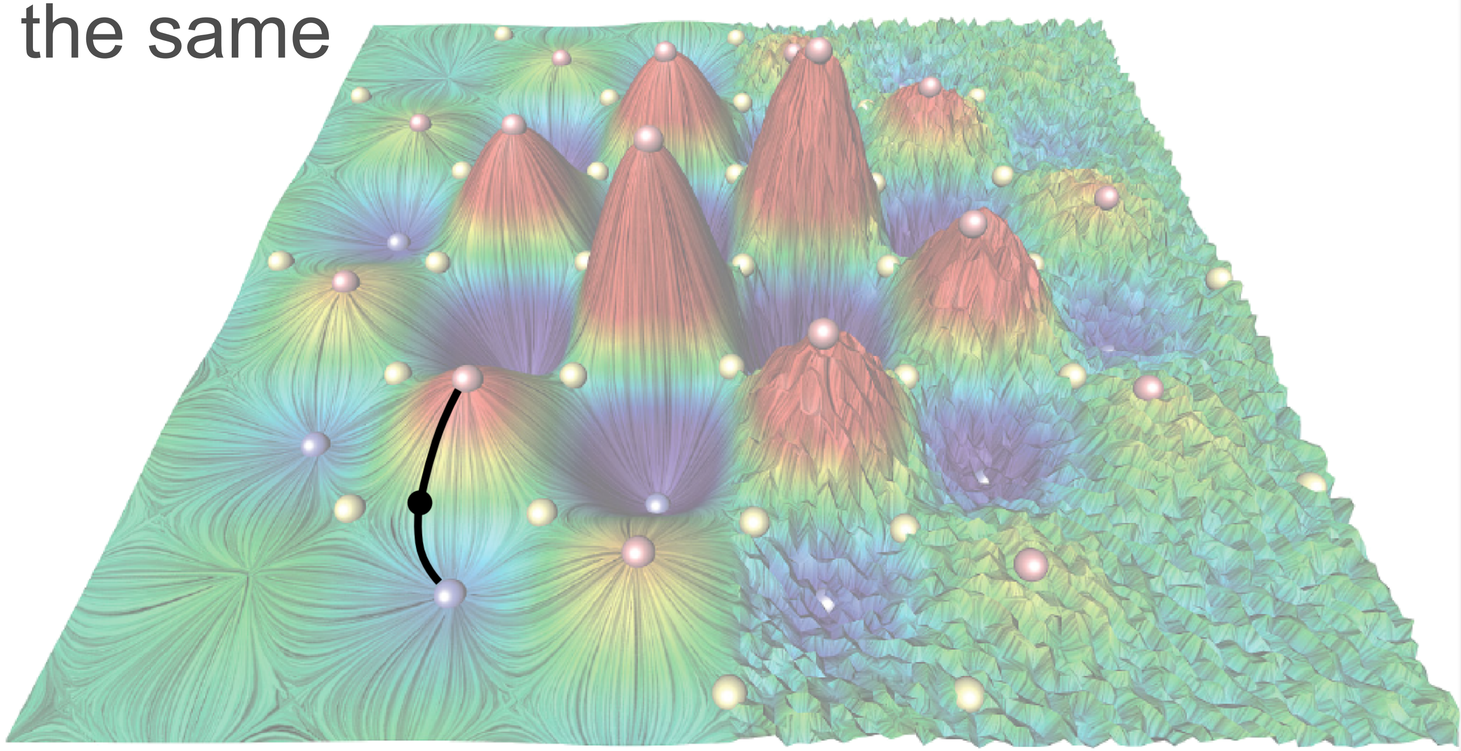
- Understanding the structure of the critical points



[Reininghaus11]

Gradient field topology

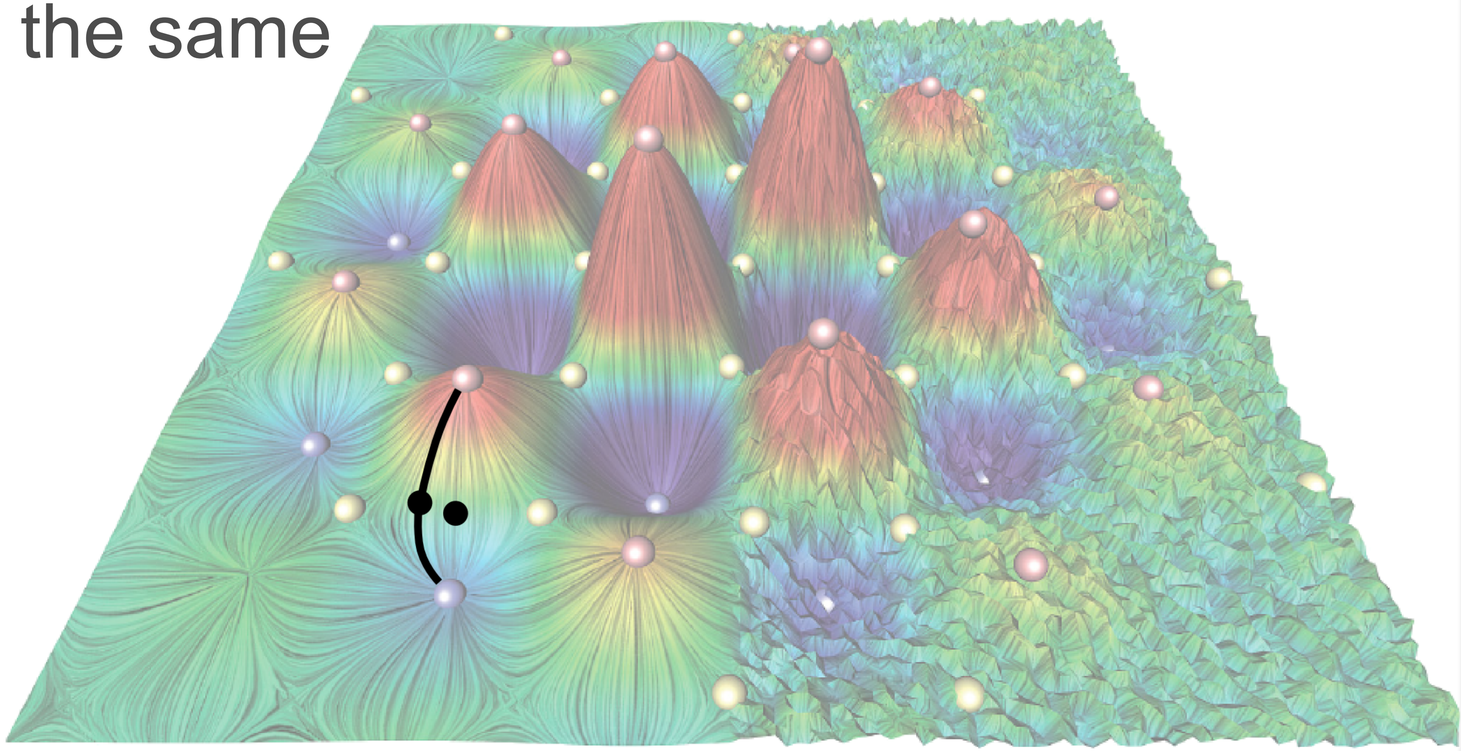
- Understanding the structure of the critical points
- Several streamlines can have the same extremities



[Reininghaus11]

Gradient field topology

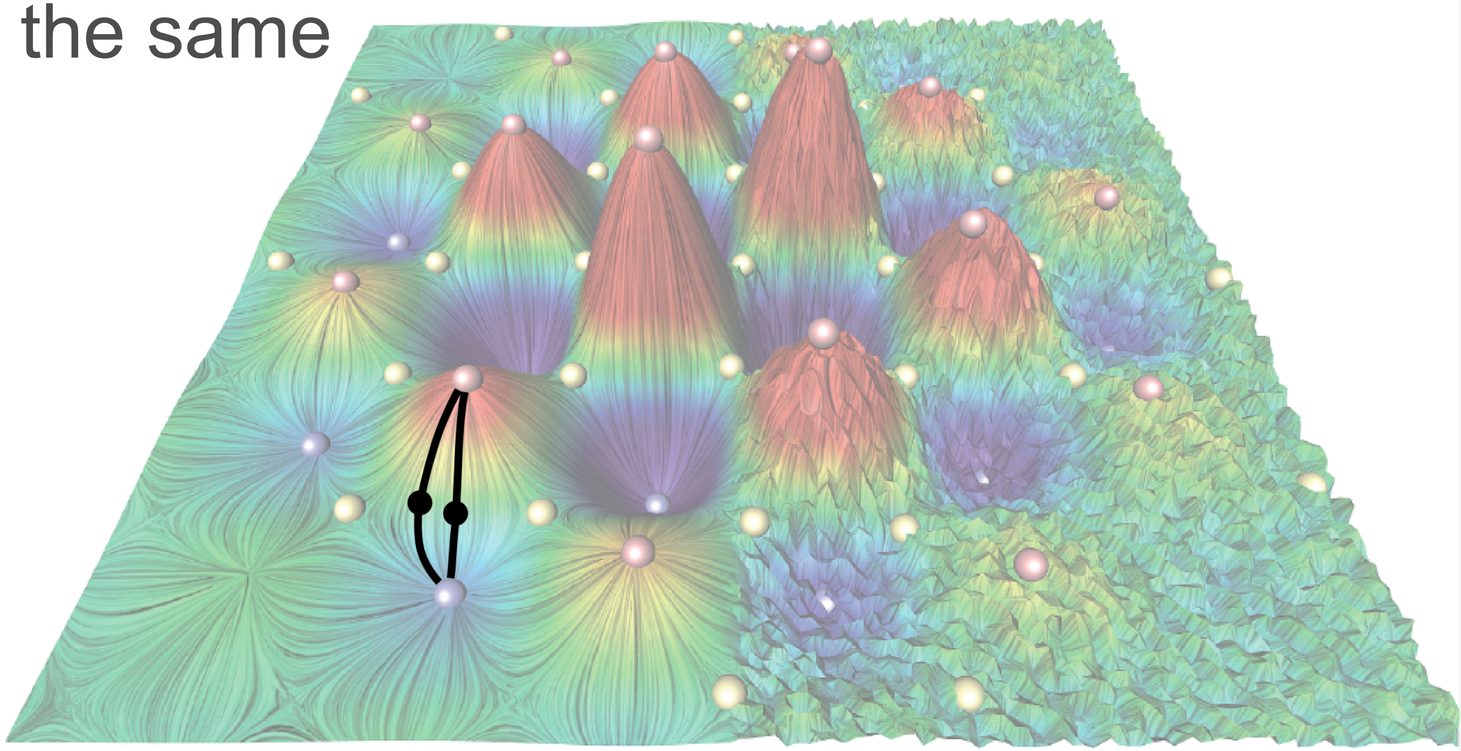
- Understanding the structure of the critical points
- Several streamlines can have the same extremities



[Reininghaus11]

Gradient field topology

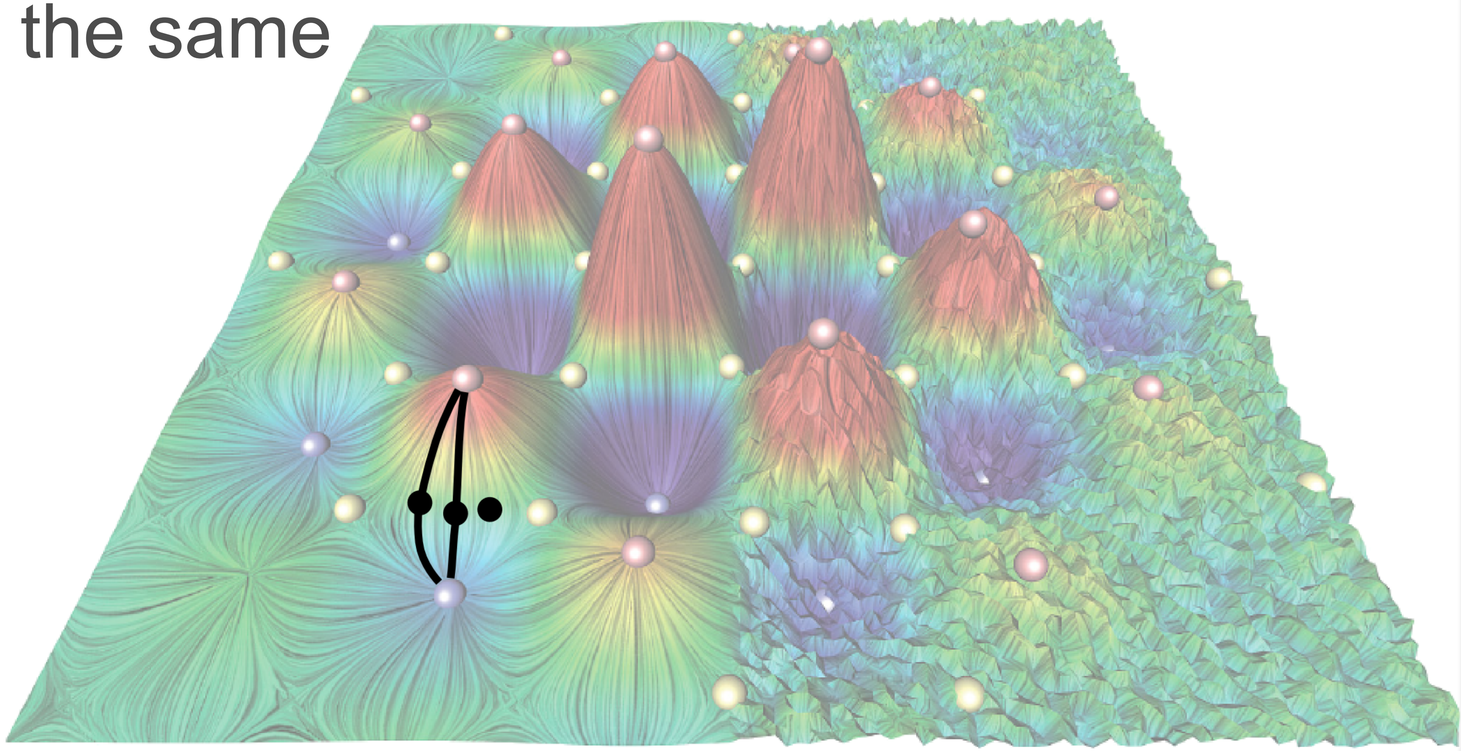
- Understanding the structure of the critical points
- Several streamlines can have the same extremities



[Reininghaus11]

Gradient field topology

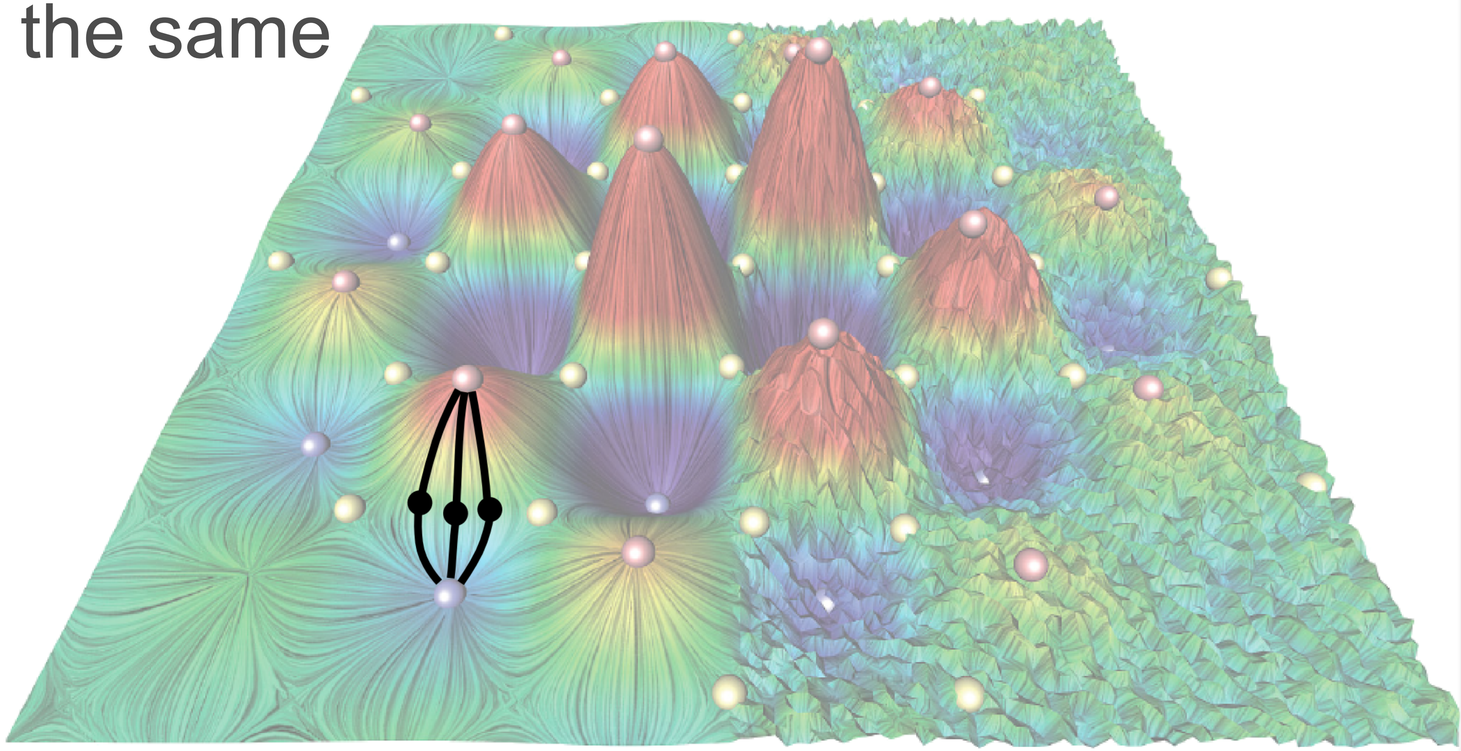
- Understanding the structure of the critical points
- Several streamlines can have the same extremities



[Reininghaus11]

Gradient field topology

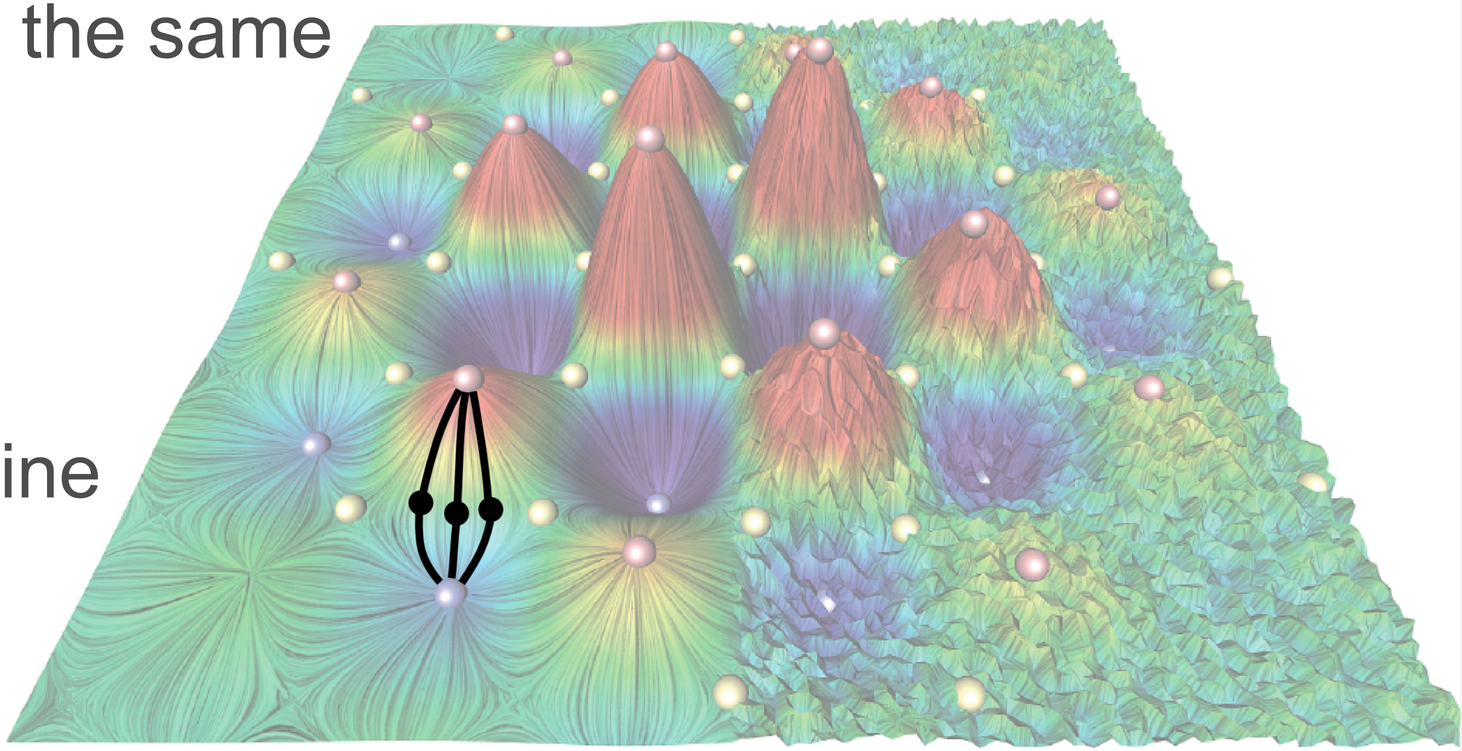
- Understanding the structure of the critical points
- Several streamlines can have the same extremities



[Reininghaus11]

Gradient field topology

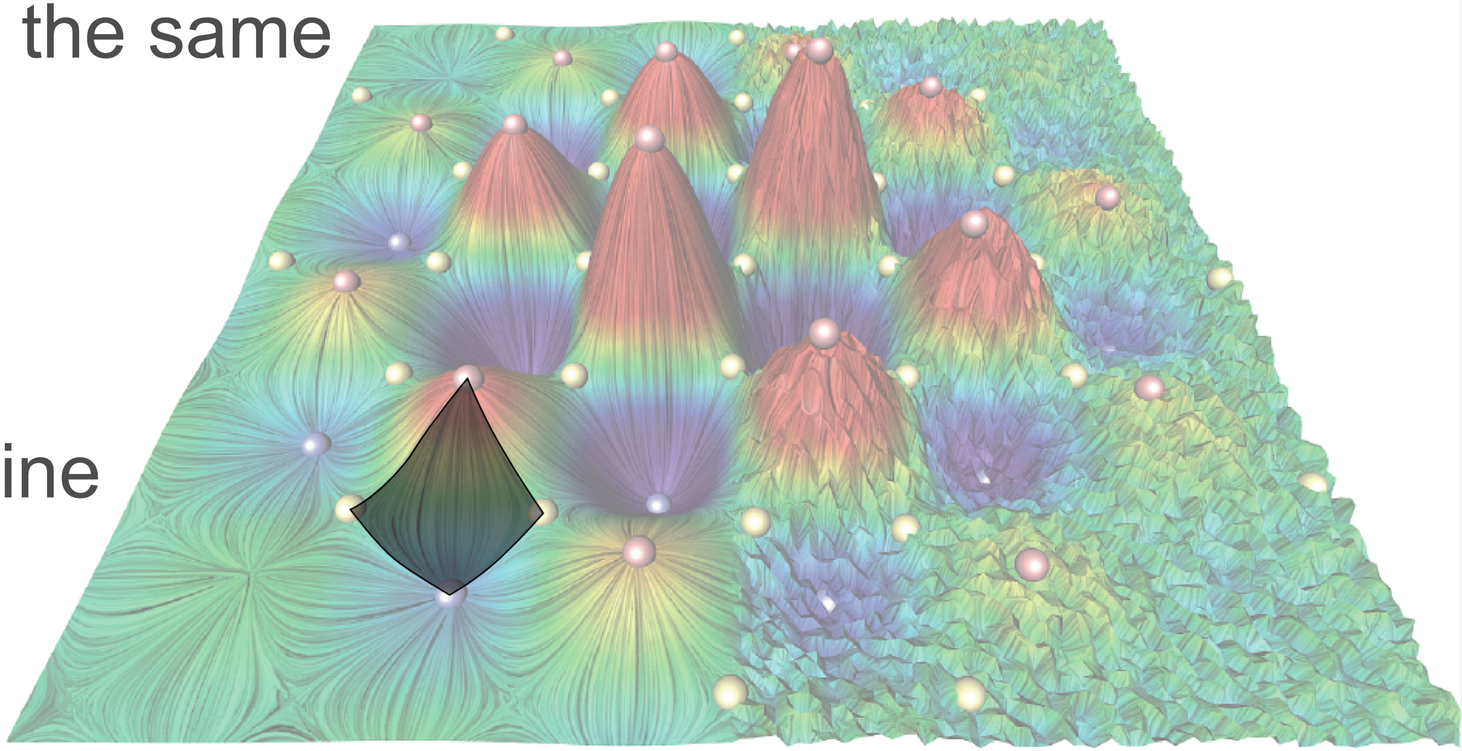
- Understanding the structure of the critical points
- Several streamlines can have the same extremities
- Equivalence relation
  - All the points whose streamline shares identical extremities



[Reininghaus11]

Gradient field topology

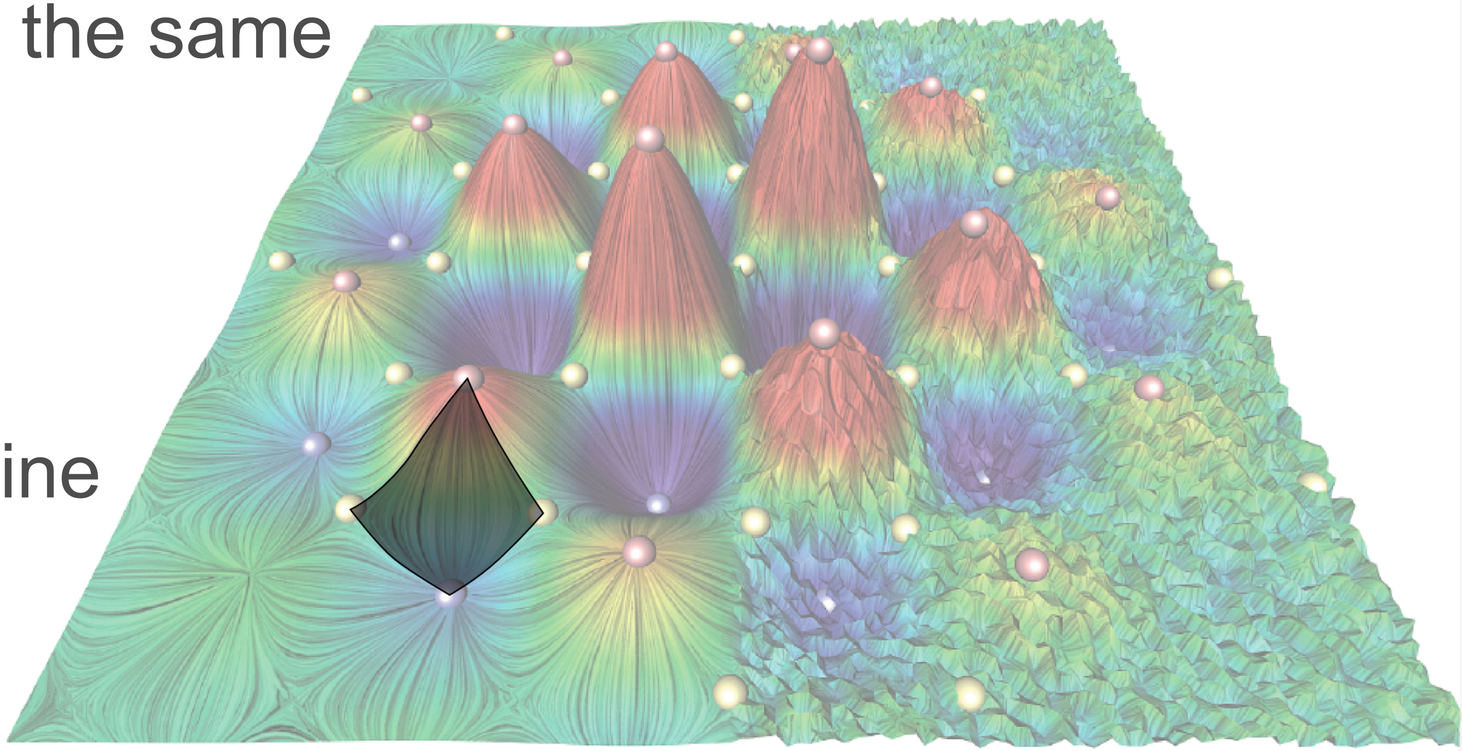
- Understanding the structure of the critical points
- Several streamlines can have the same extremities
- Equivalence relation
  - All the points whose streamline shares identical extremities



[Reininghaus11]

Gradient field topology

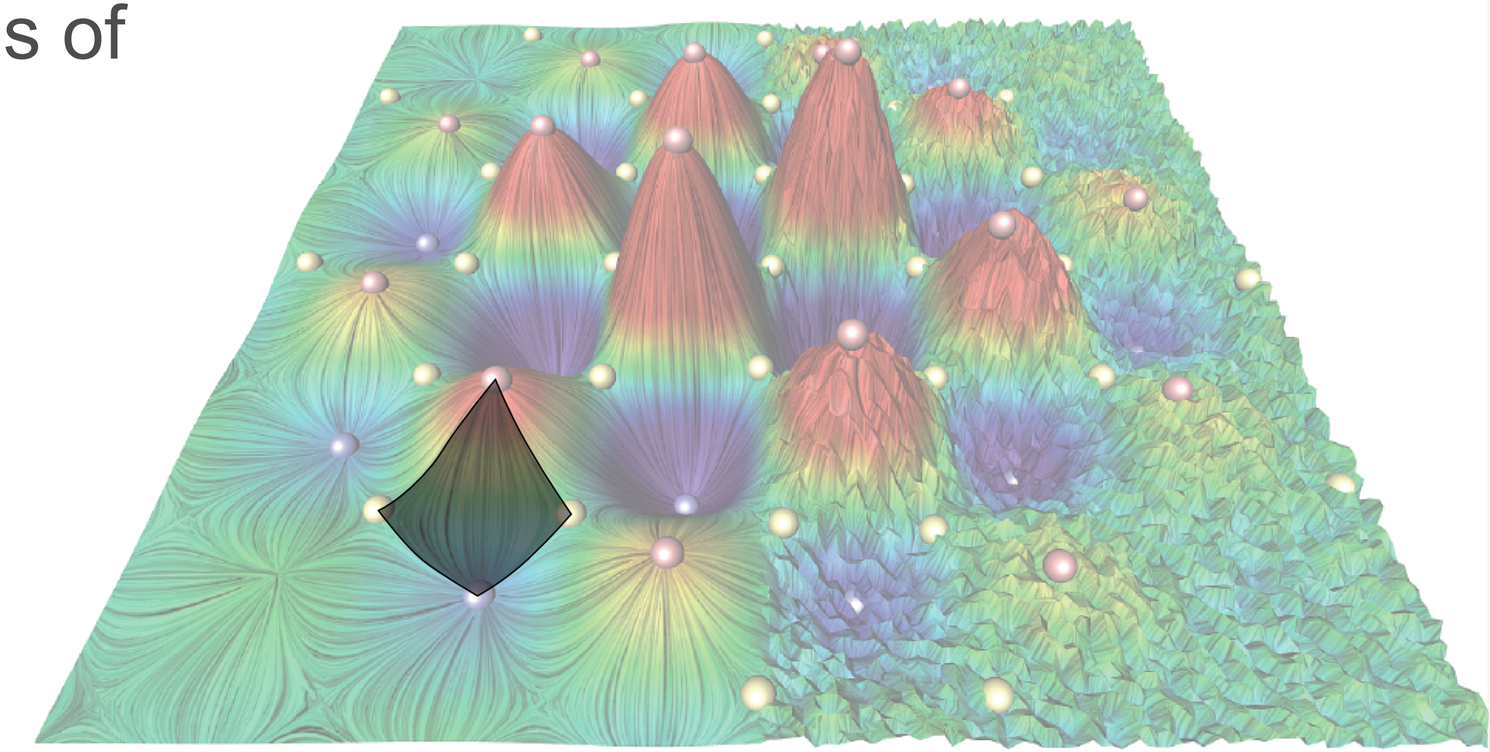
- Understanding the structure of the critical points
- Several streamlines can have the same extremities
- Equivalence relation
  - All the points whose streamline shares identical extremities
  - **Notion of flow cell**



[Reininghaus11]

Gradient field topology

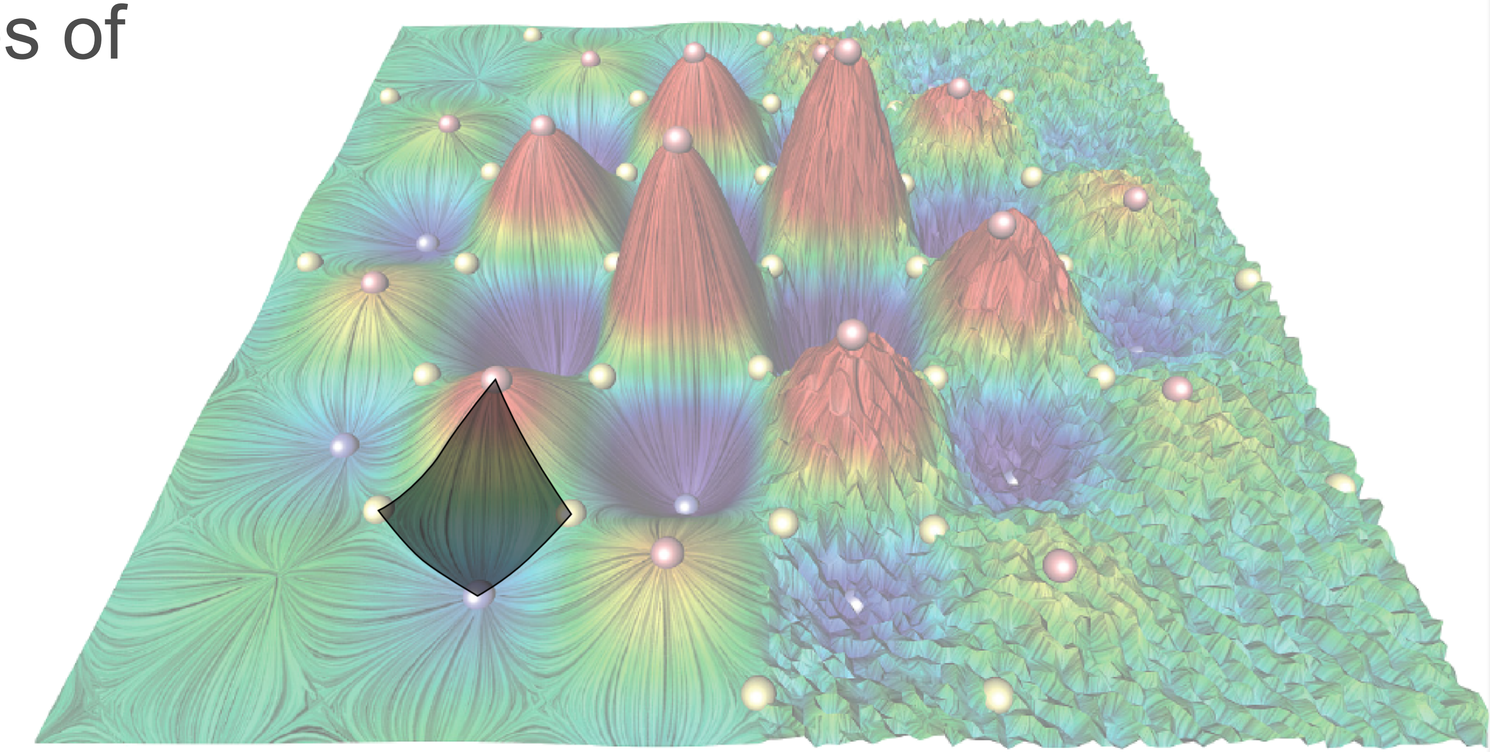
- Understanding the structure of the critical points
- Now, what are the boundaries of the flow cells?



[Reininghaus11]

Gradient field topology

- Understanding the structure of the critical points
- Now, what are the boundaries of the flow cells?
- Streamlines between critical points



[Reininghaus11]

Critical points of a vector field

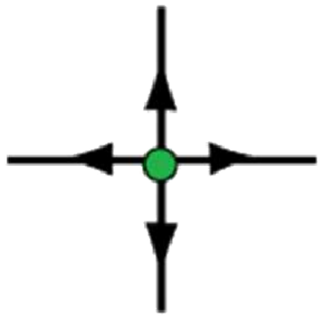
- For example
  - $f : \mathbb{R}^2 \to \mathbb{R}^2$

Critical points of a vector field

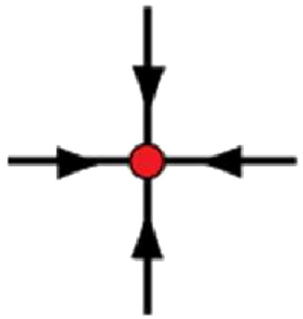
- For example
  - $f : \mathbb{R}^2 \to \mathbb{R}^2$
- Points where the magnitude vanishes
  - $f(p) = p$

Critical points of a vector field

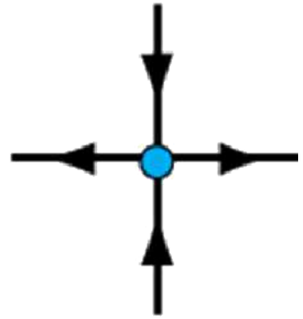
- For example
  - $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$
- Points where the magnitude vanishes
  - $f(p) = p$



Source



Sink

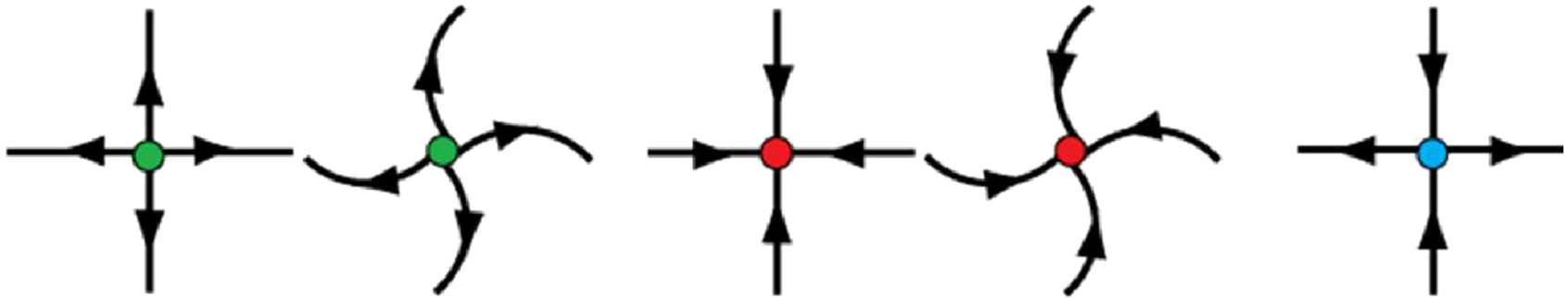


Saddle

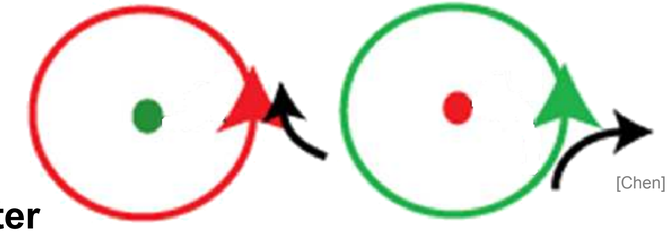
[Chen]

Critical points of a vector field

- For example
  - $f : \mathbb{R}^2 \to \mathbb{R}^2$
- Points where the magnitude vanishes
  - $f(p) = p$
- With curl
  - More critical points



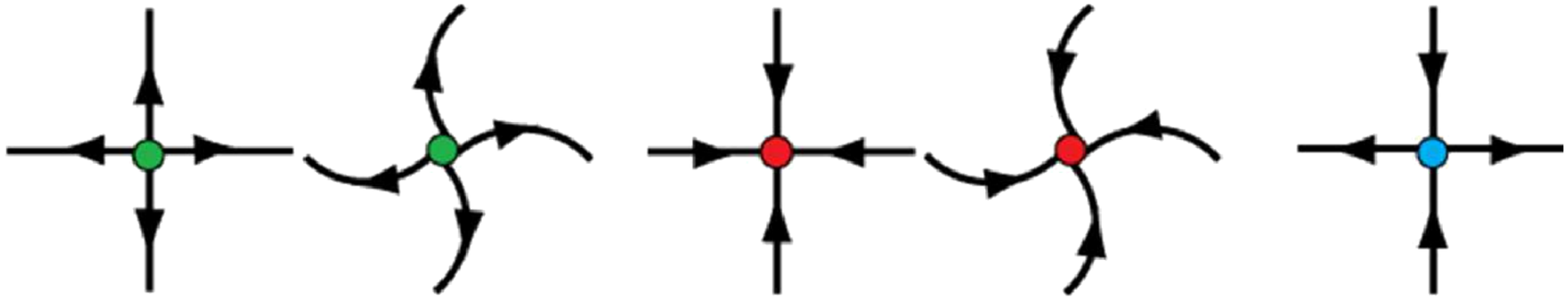
Source Sink Saddle



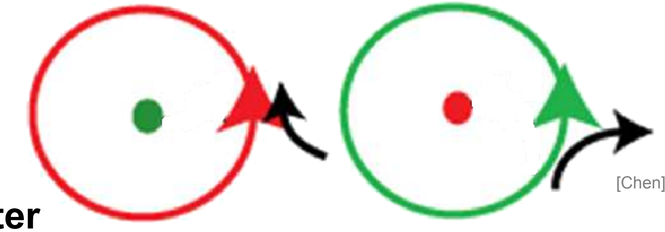
Center

[Chen]

Classifying critical points



Source Sink Saddle

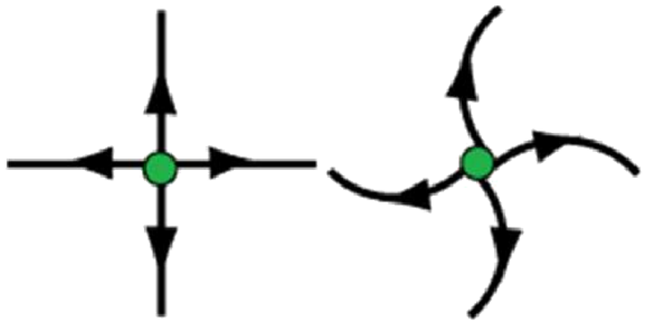


Center

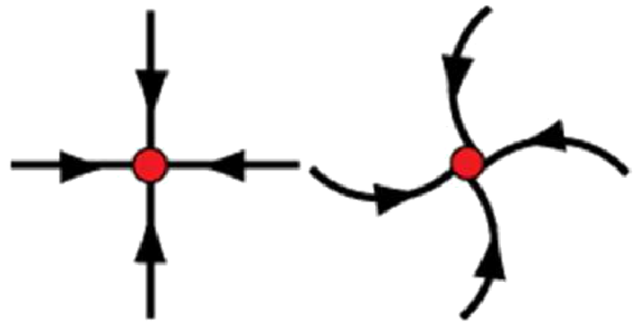
[Chen]

Classifying critical points

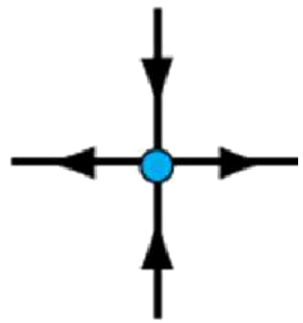
- Jacobian of the vector field



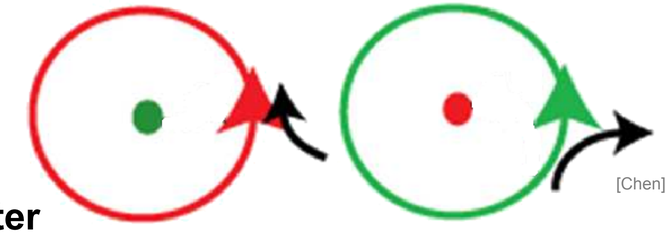
Source



Sink



Saddle



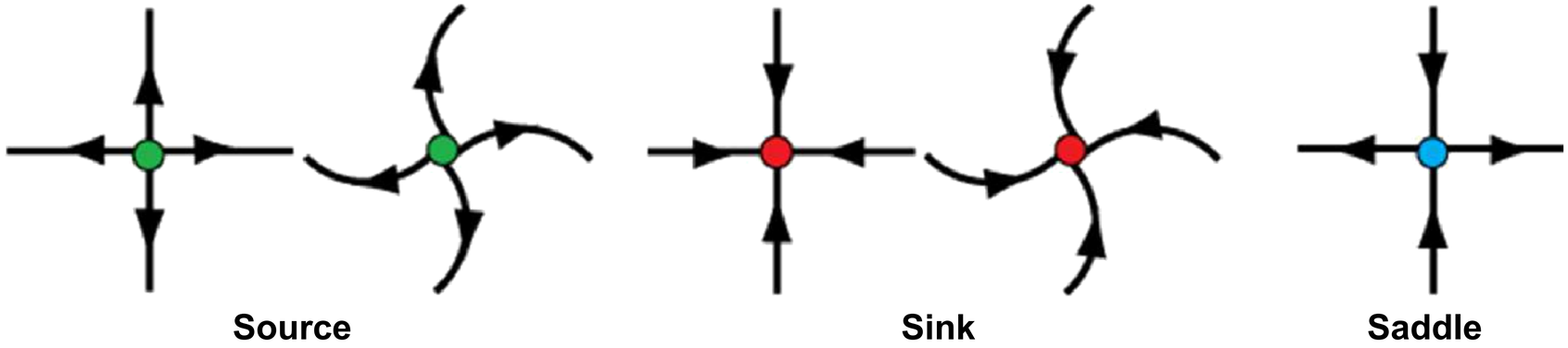
Center

[Chen]

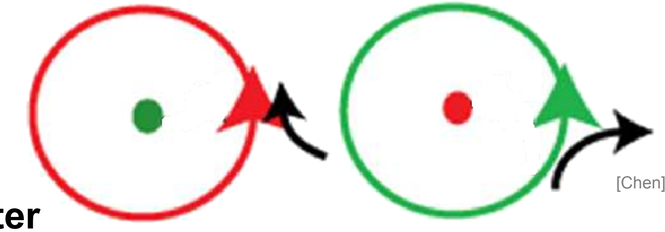
Classifying critical points

- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$



Source Sink Saddle



Center

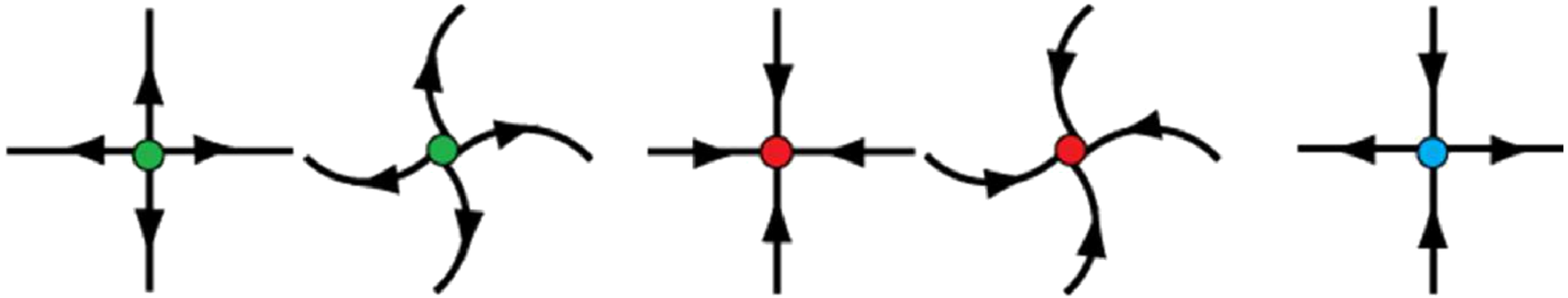
[Chen]

Classifying critical points

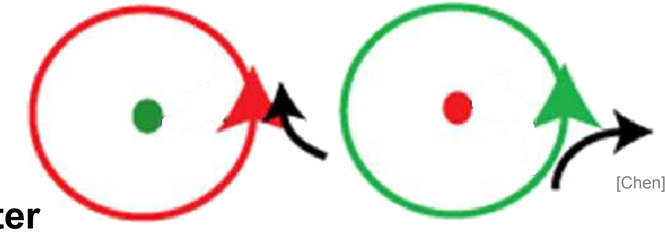
- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues



Source Sink Saddle



Center

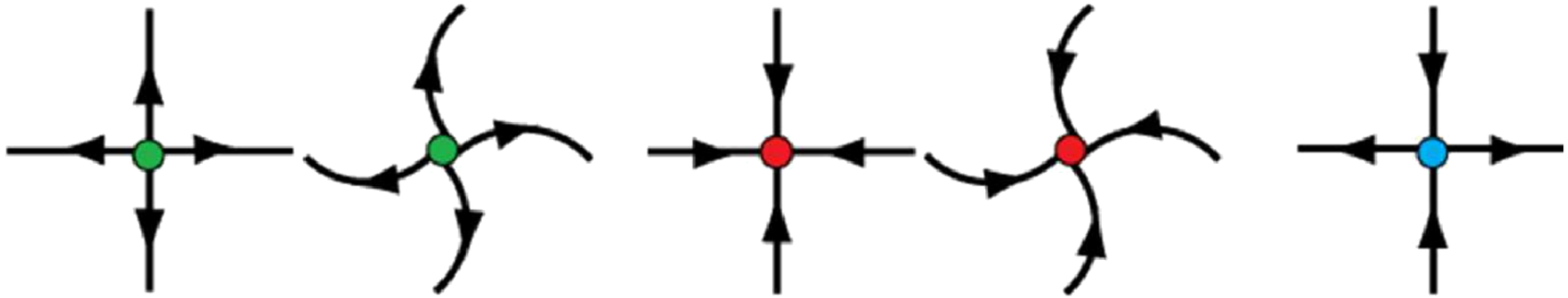
[Chen]

Classifying critical points

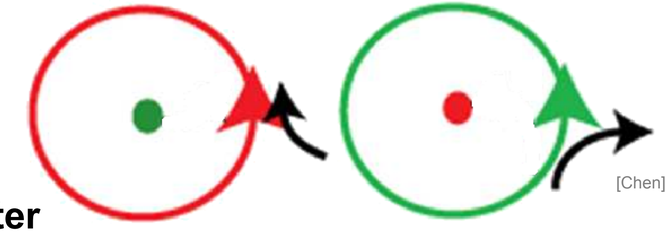
- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues
  - For each



Source Sink Saddle



Center

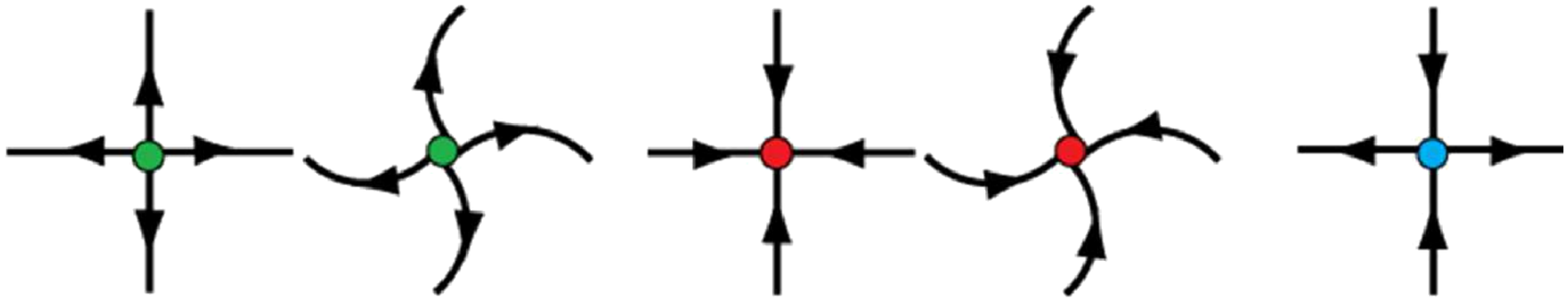
[Chen]

Classifying critical points

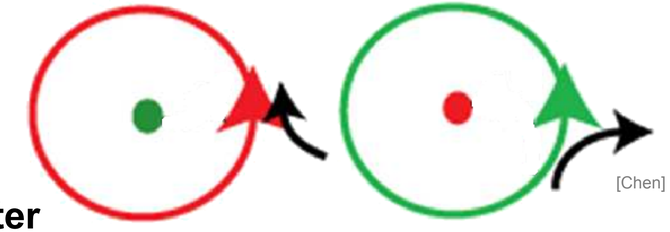
- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric



Source Sink Saddle



Center

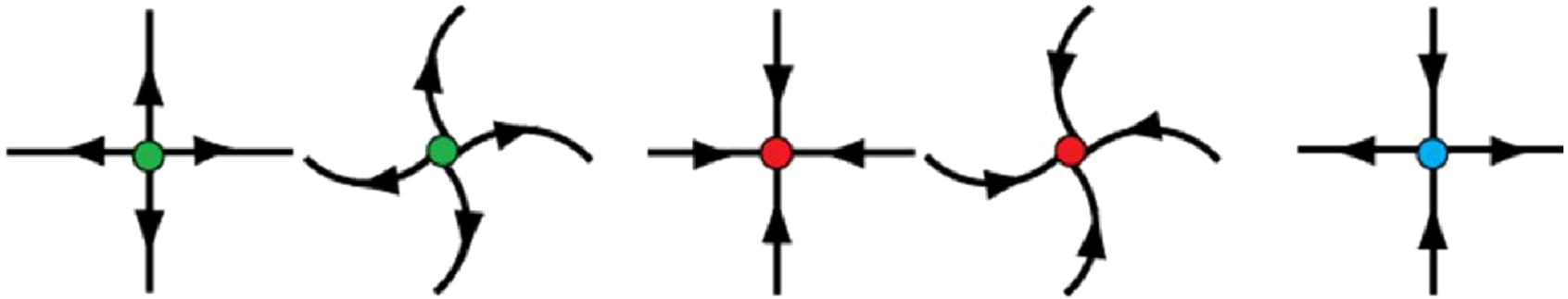
[Chen]

Classifying critical points

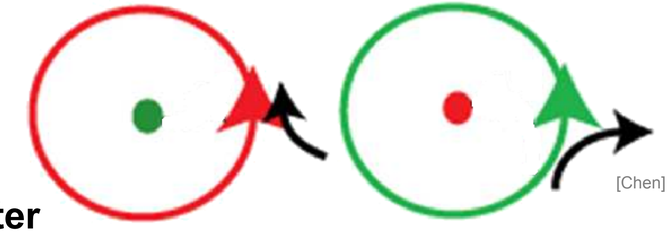
- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric
    - R1, I1 - R2, I2



Source Sink Saddle



Center

[Chen]

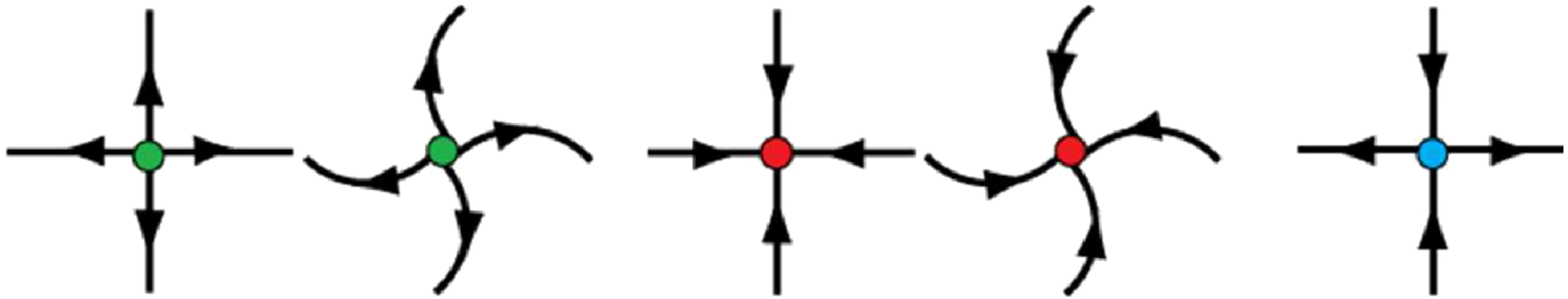
Classifying critical points

- Jacobian of the vector field

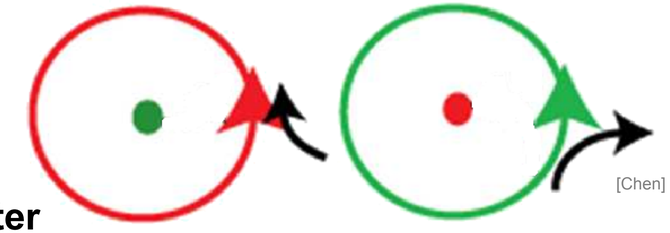
$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric
    - R1, I1 - R2, I2

$$R_1 > 0, R_2 > 0$$
$$I_1 = 0, I_2 = 0$$



Source Sink Saddle



Center

[Chen]

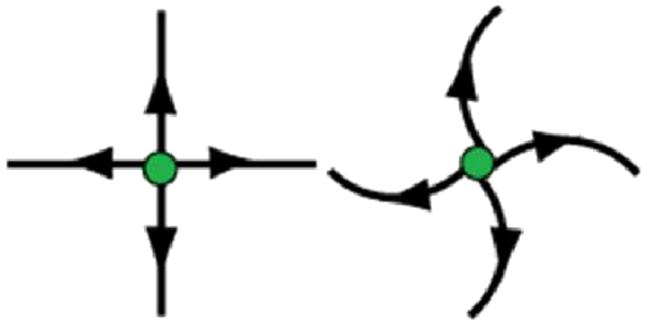
Classifying critical points

- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

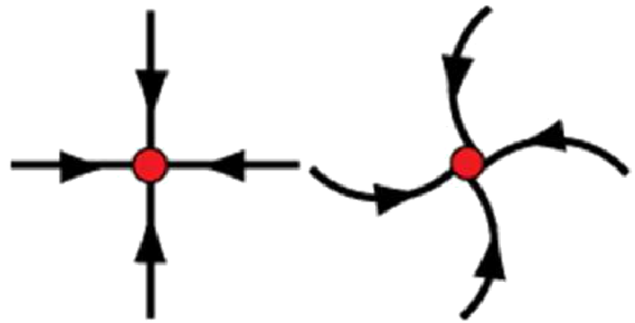
- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric
    - R1, I1 - R2, I2

$R_1 > 0, R_2 > 0$
$I_1 = 0, I_2 = 0$

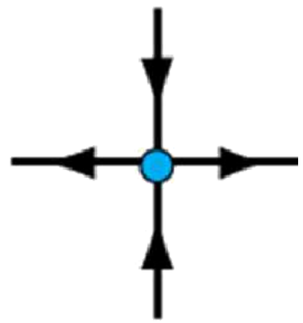


Source

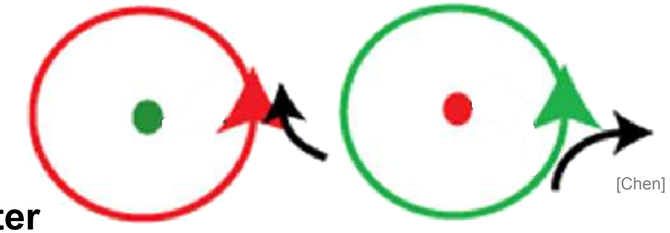
$R_1 < 0, R_2 < 0$
$I_1 = 0, I_2 = 0$



Sink



Saddle



Center

[Chen]

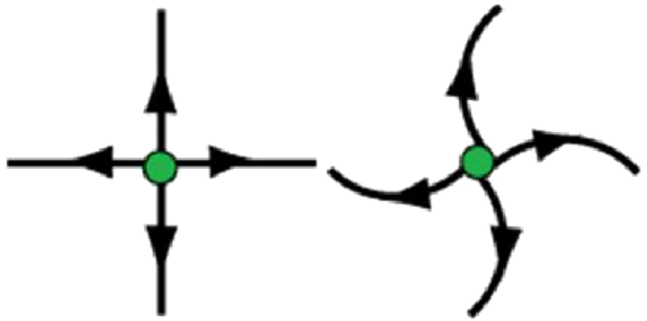
Classifying critical points

- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

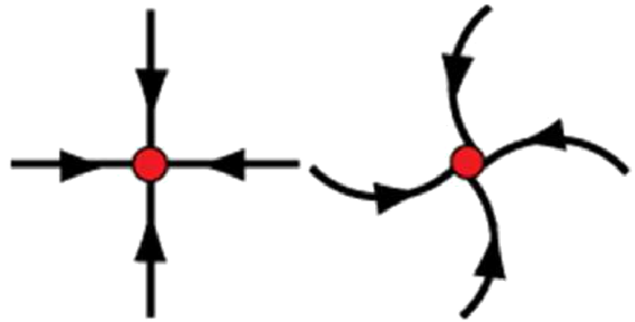
- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric
    - R1, I1 - R2, I2

$R_1 > 0, R_2 > 0$
$I_1 = 0, I_2 = 0$



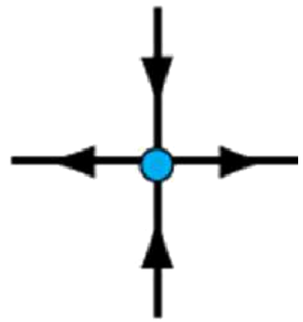
Source

$R_1 < 0, R_2 < 0$
$I_1 = 0, I_2 = 0$

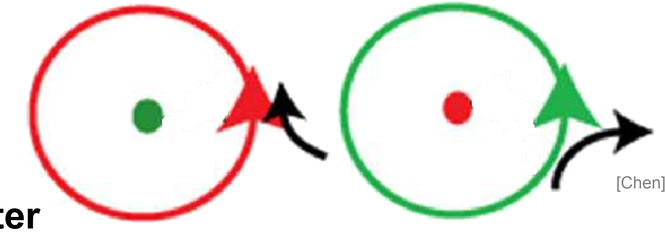


Sink

$R_1.R_2 < 0$
$I_1 = 0, I_2 = 0$



Saddle



Center

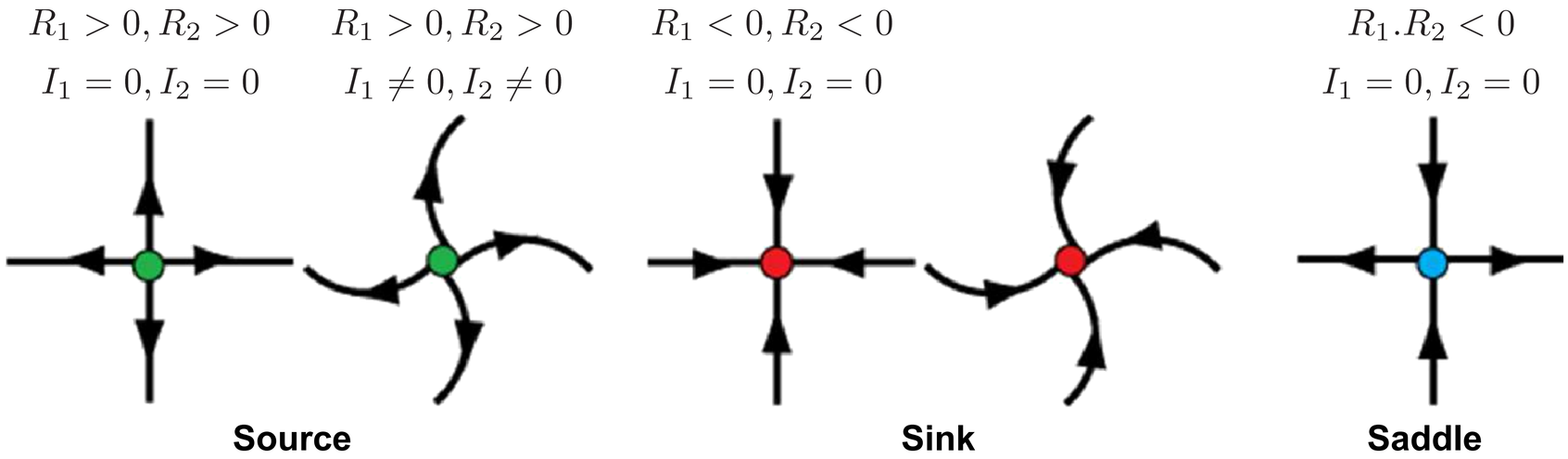
[Chen]

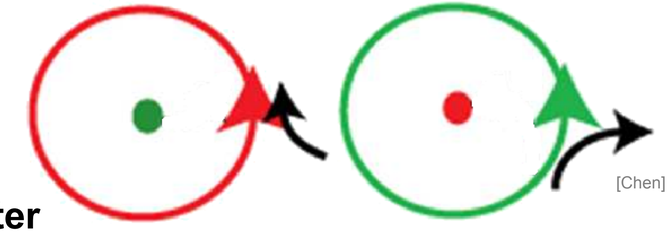
Classifying critical points

- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric
    - R1, I1 - R2, I2





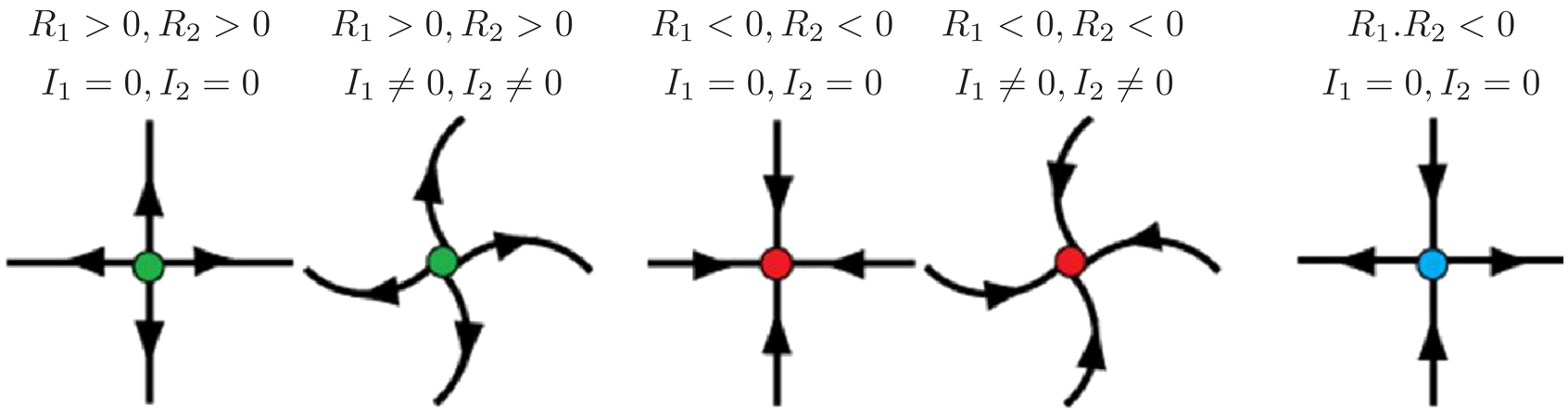
[Chen]

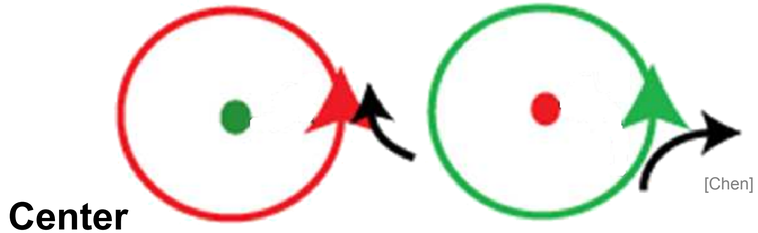
Classifying critical points

- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric
    - R1, I1 - R2, I2



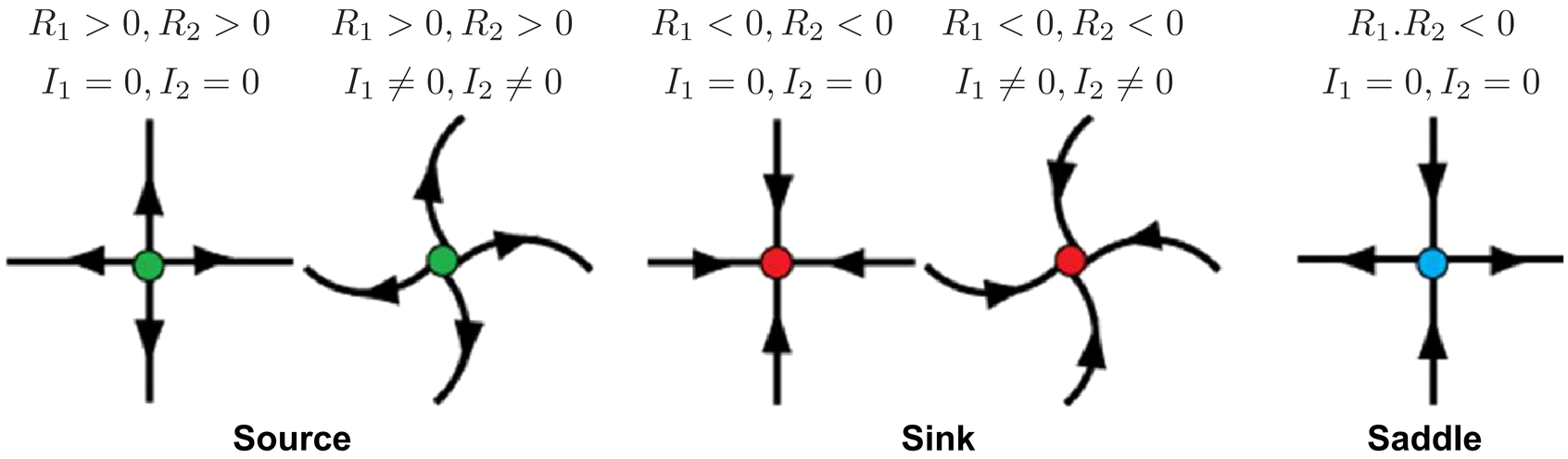


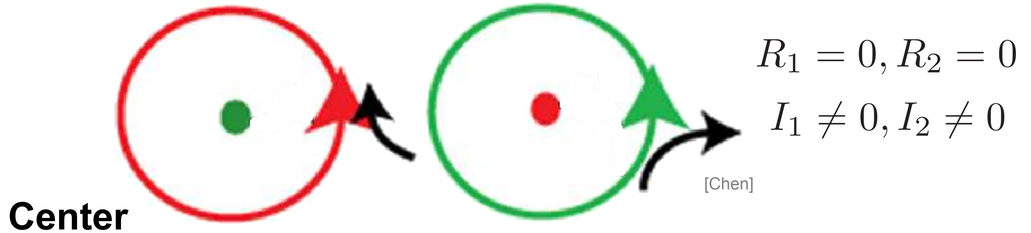
Classifying critical points

- Jacobian of the vector field

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix}$$

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - –Symmetric, antisymmetric
    - R1, I1 - R2, I2





Vector field decomposition

Vector field decomposition

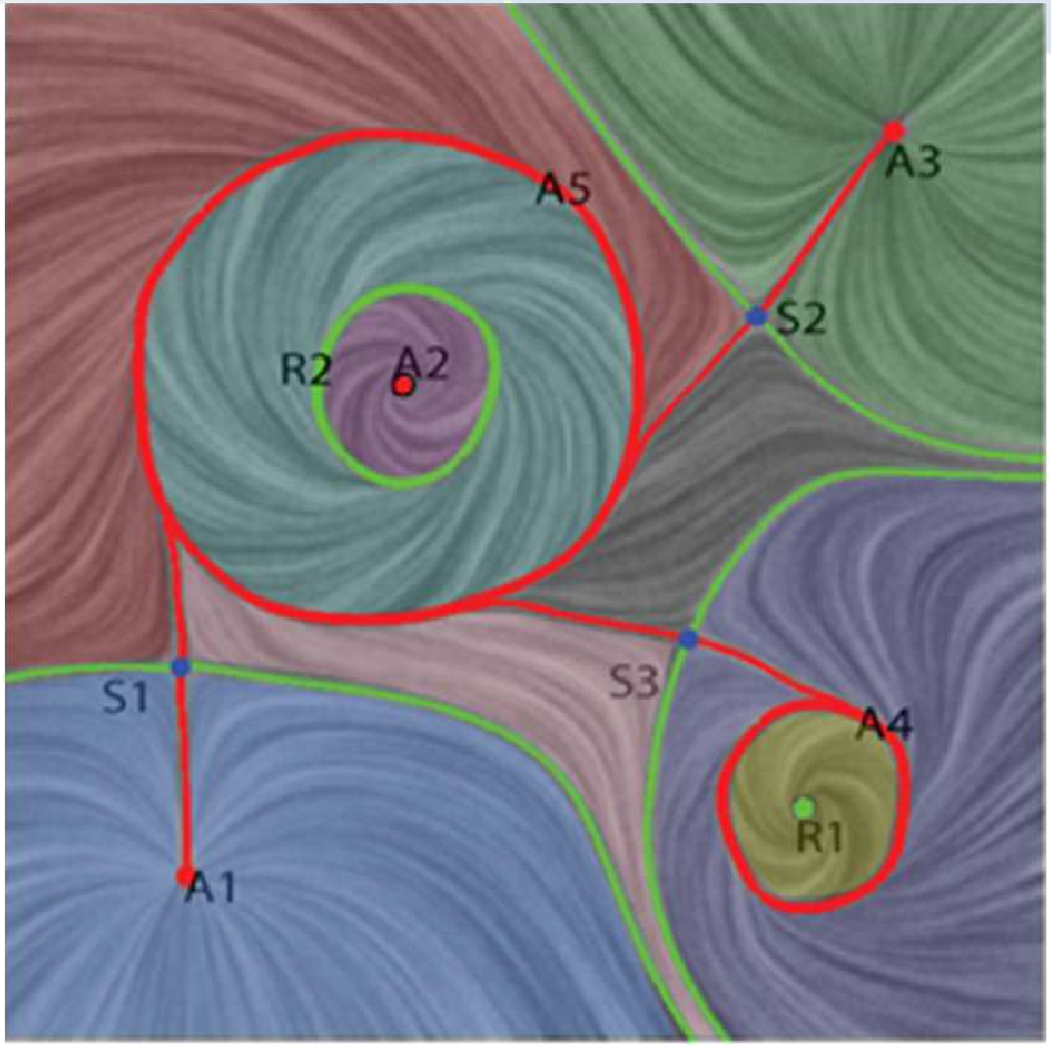
- Critical point extraction
  - Identify the cells of the domain containing critical points

Vector field decomposition

- Critical point extraction
  - Identify the cells of the domain containing critical points
  - Sub-sampling for accurate locations

Vector field decomposition

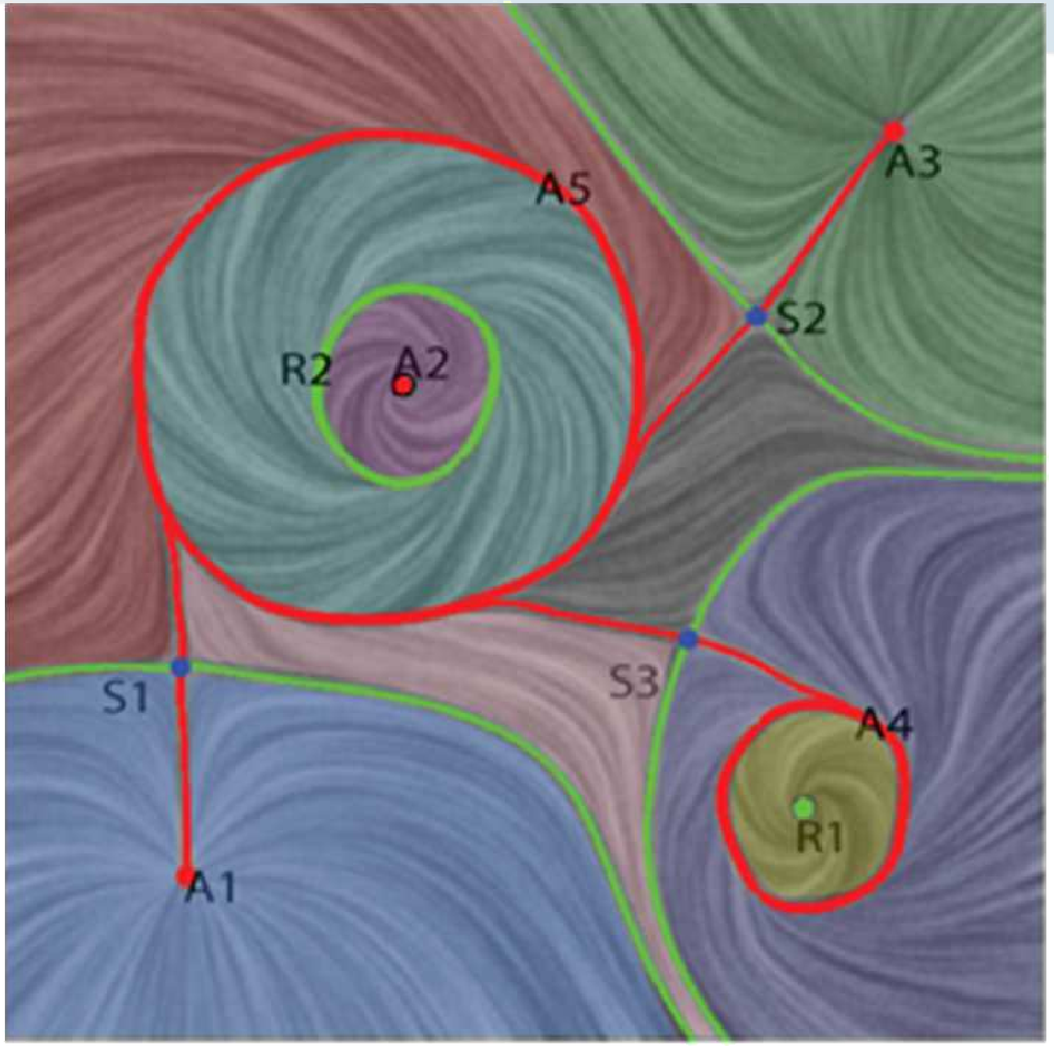
- Critical point extraction
  - Identify the cells of the domain containing critical points
  - Sub-sampling for accurate locations
- Decomposition



[Chen]

Vector field decomposition

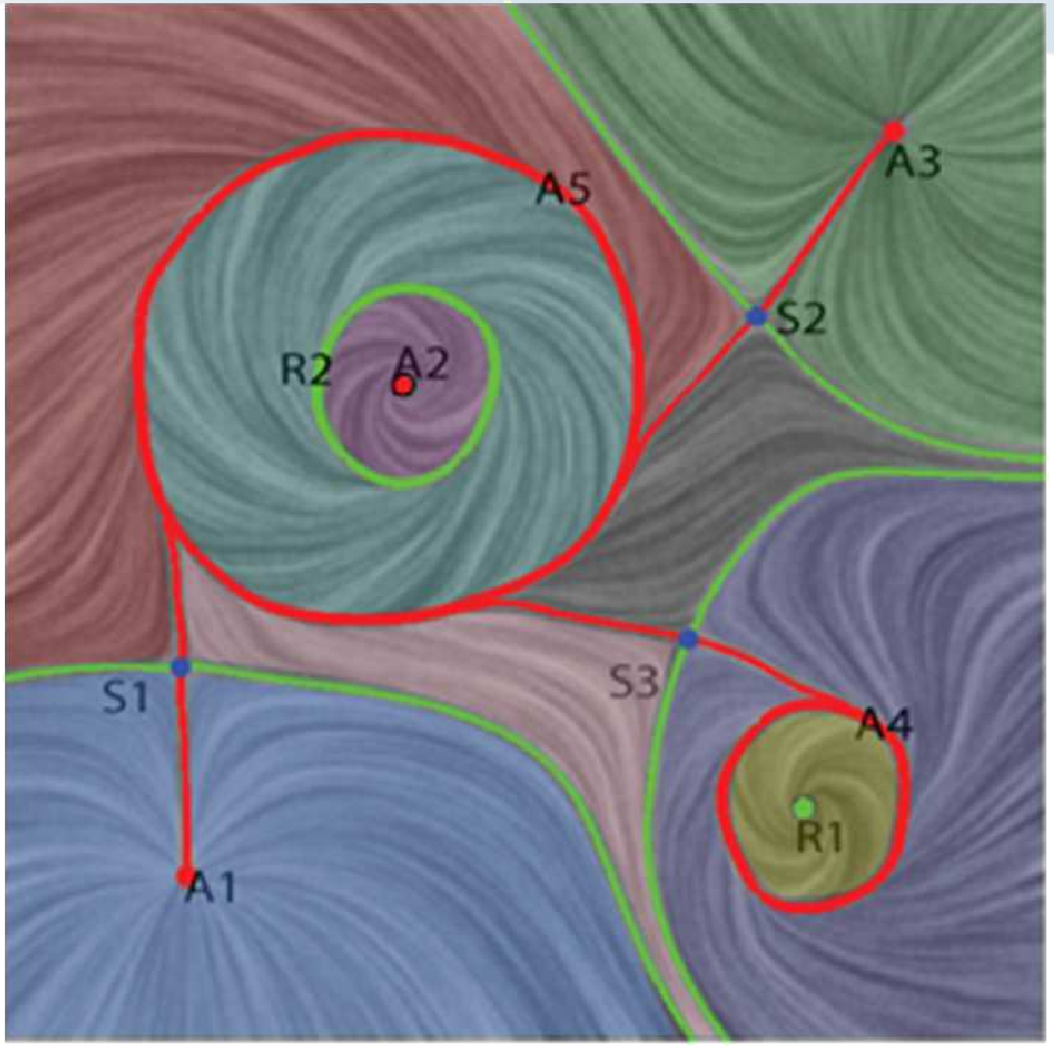
- Critical point extraction
  - Identify the cells of the domain containing critical points
  - Sub-sampling for accurate locations
- Decomposition
  - Backward and forward streamlines from the critical points



[Chen]

Vector field decomposition

- Critical point extraction
  - Identify the cells of the domain containing critical points
  - Sub-sampling for accurate locations
- Decomposition
  - Backward and forward streamlines from the critical points
  - Periodic orbits!



[Chen]

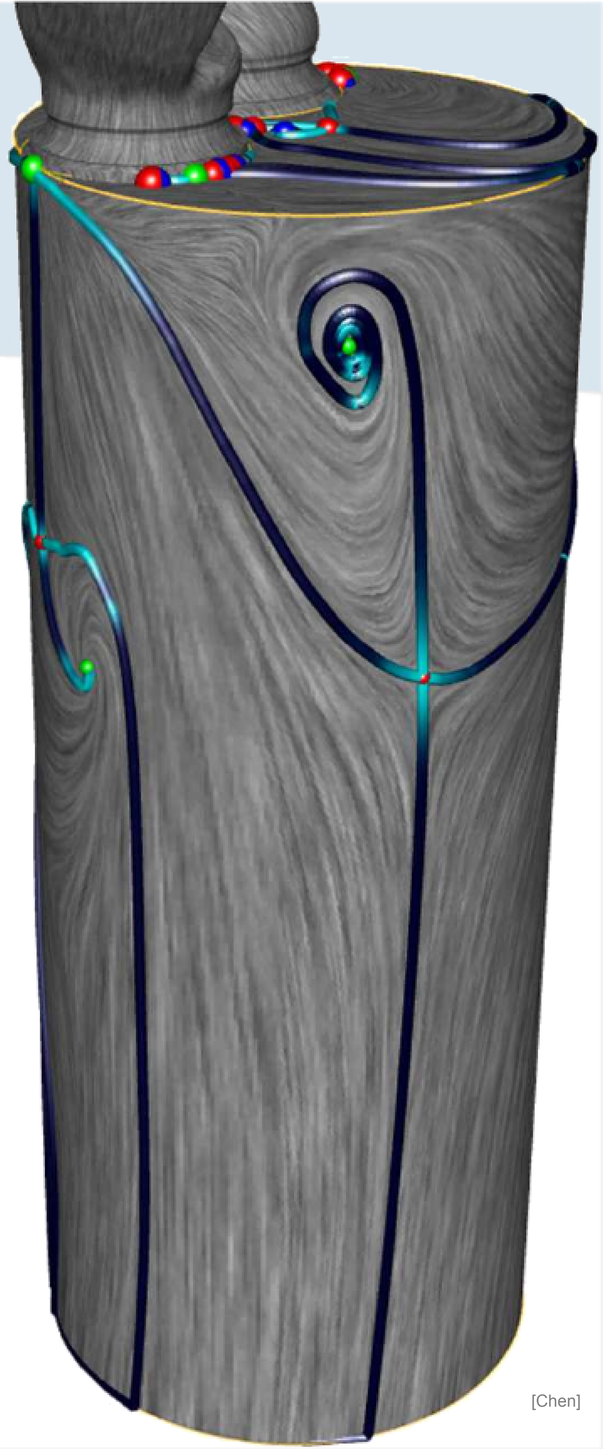
Non planar domains

Non planar domains

- PL 2-manifolds in $\mathbb{R}^3$

Non planar domains

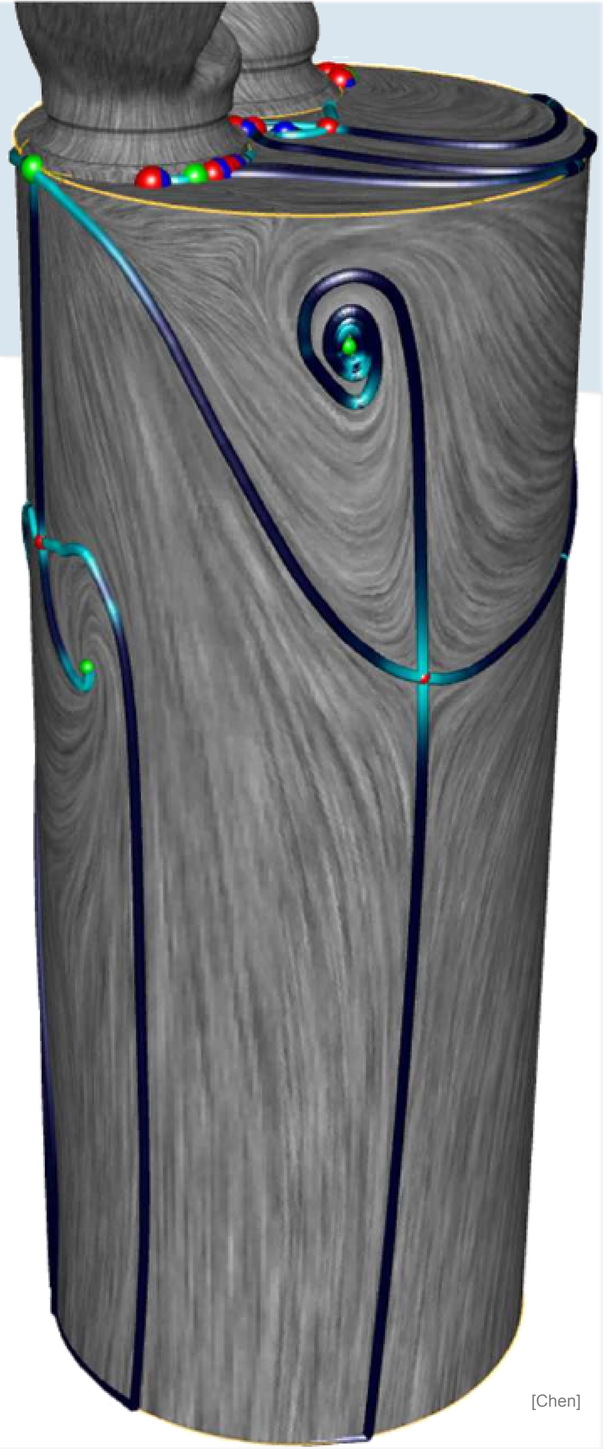
- PL 2-manifolds in $\mathbb{R}^3$
  - Trivial extension
  - Numerical evaluations slightly more involved



[Chen]

Non planar domains

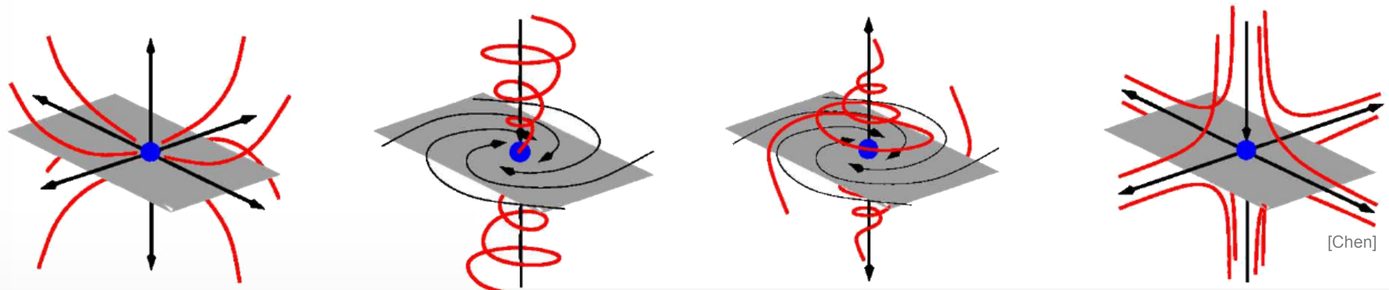
- PL 2-manifolds in $\mathbb{R}^3$
  - Trivial extension
  - Numerical evaluations slightly more involved
- Volumetric domains
  - Similar process



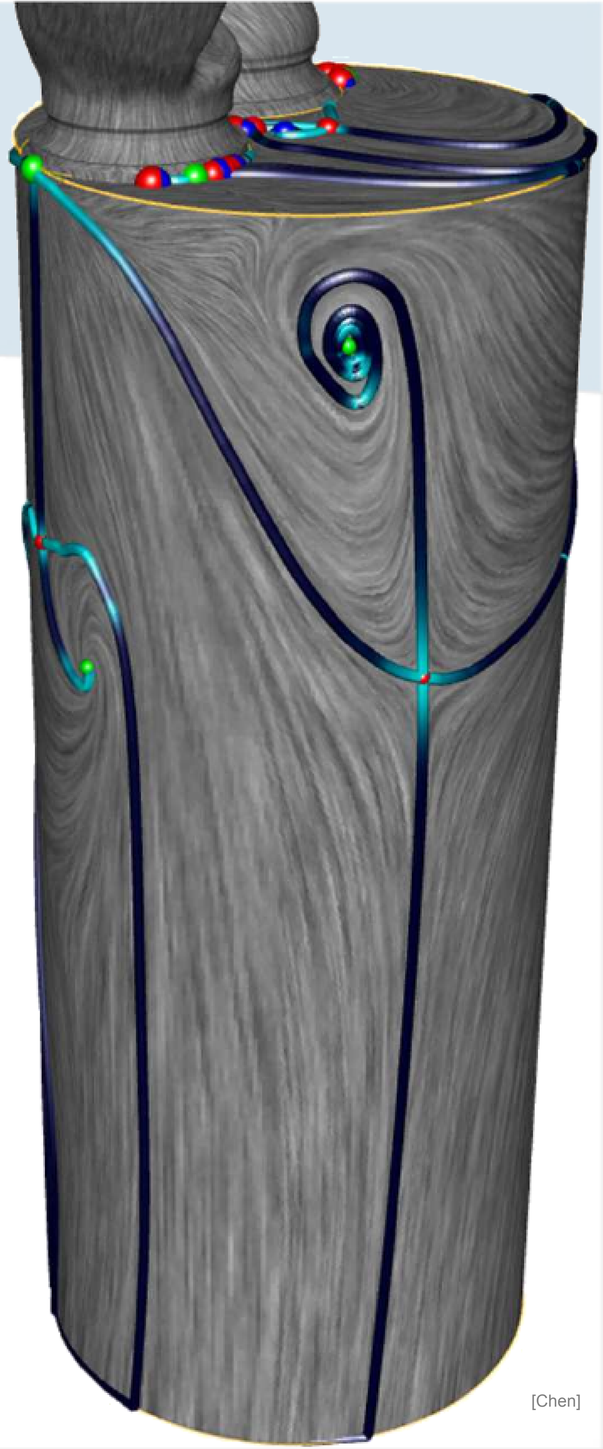
[Chen]

Non planar domains

- PL 2-manifolds in $\mathbb{R}^3$
  - Trivial extension
  - Numerical evaluations slightly more involved
- Volumetric domains
  - Similar process
    - Critical points: spiral effects



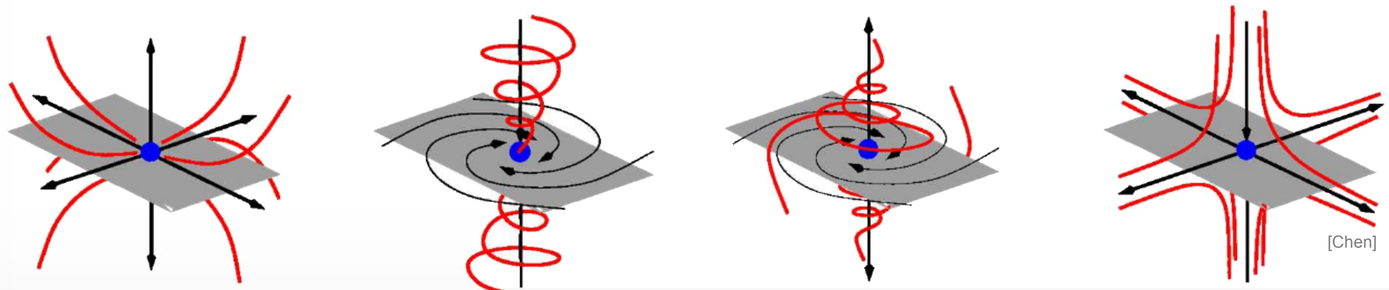
[Chen]



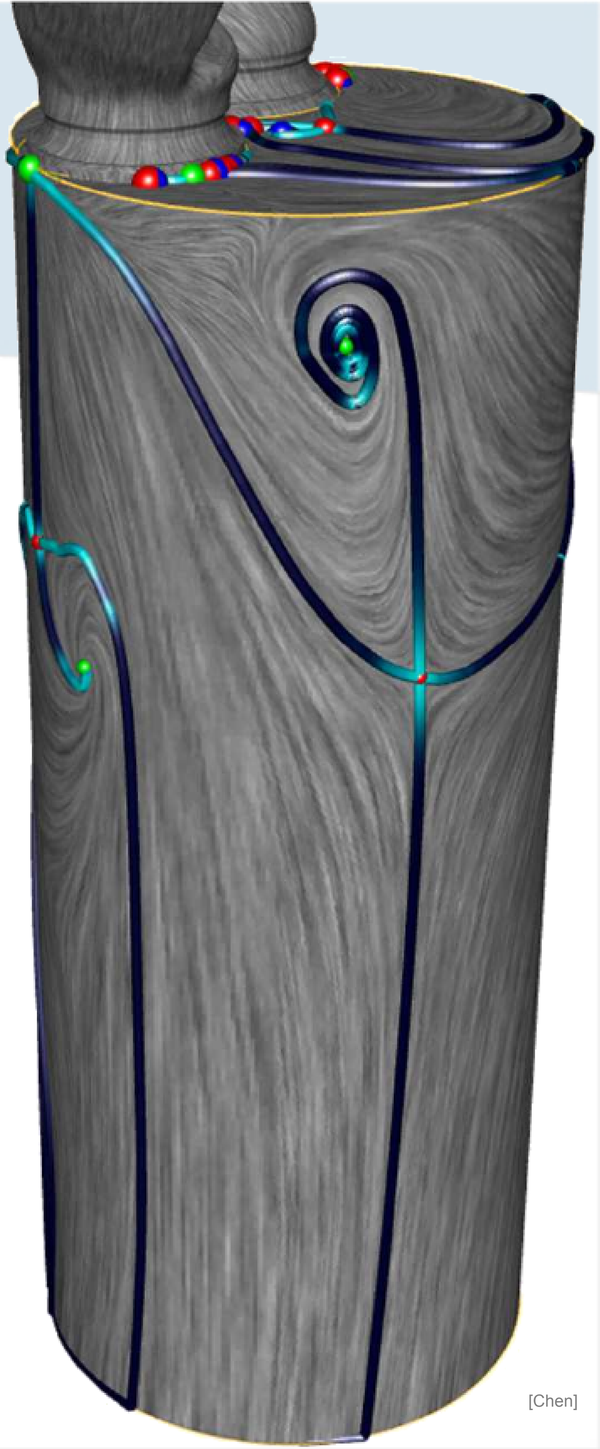
[Chen]

Non planar domains

- PL 2-manifolds in $\mathbb{R}^3$
  - Trivial extension
  - Numerical evaluations slightly more involved
- Volumetric domains
  - Similar process
    - Critical points: spiral effects
    - Streamlines and streamsurfaces as separatrices

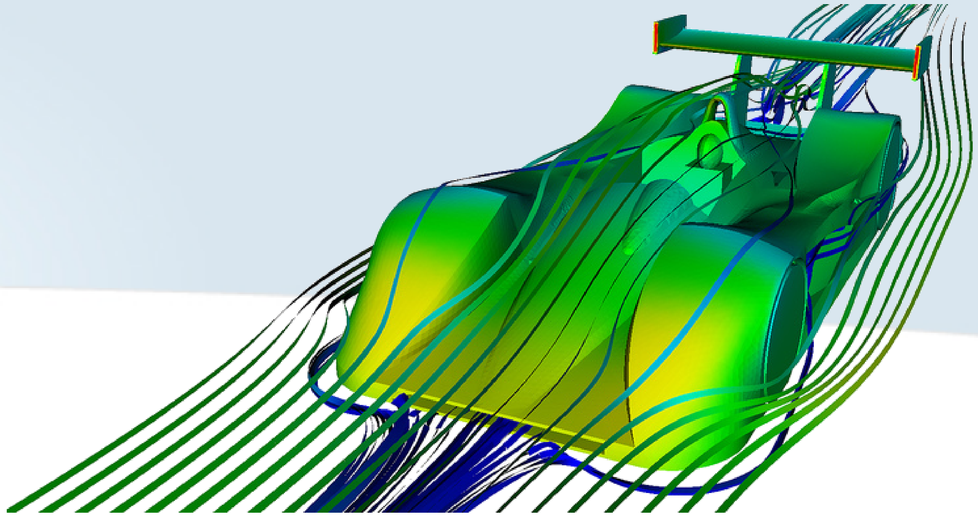


[Chen]

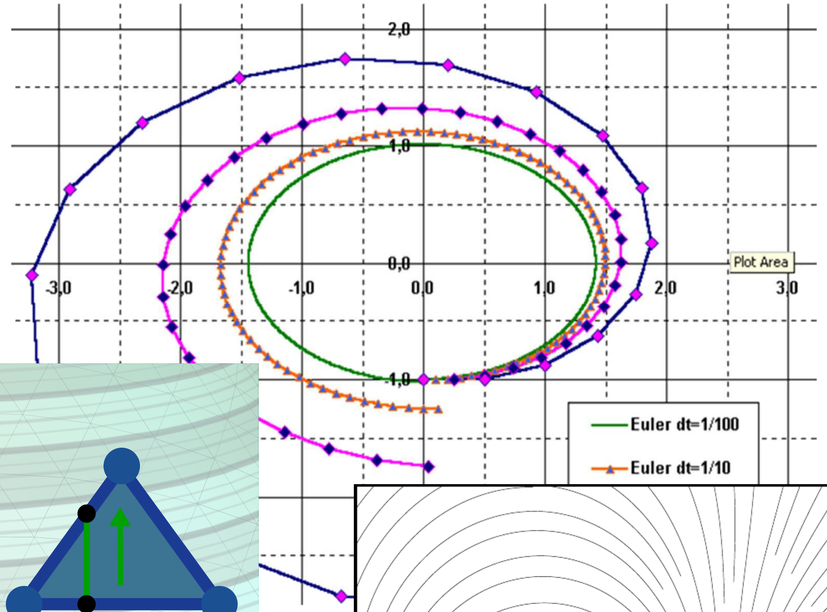


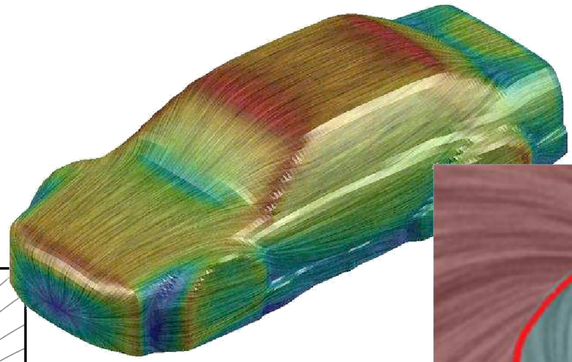
[Chen]

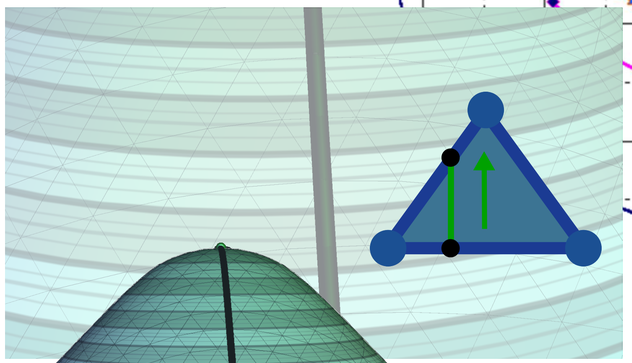
Vector fields

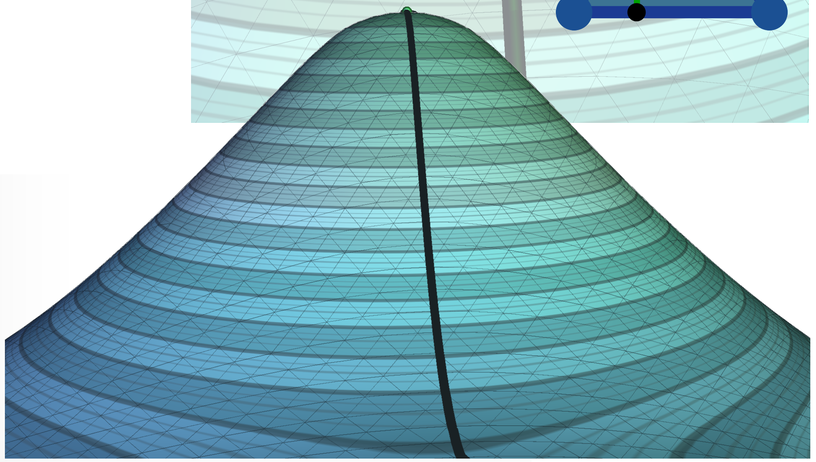


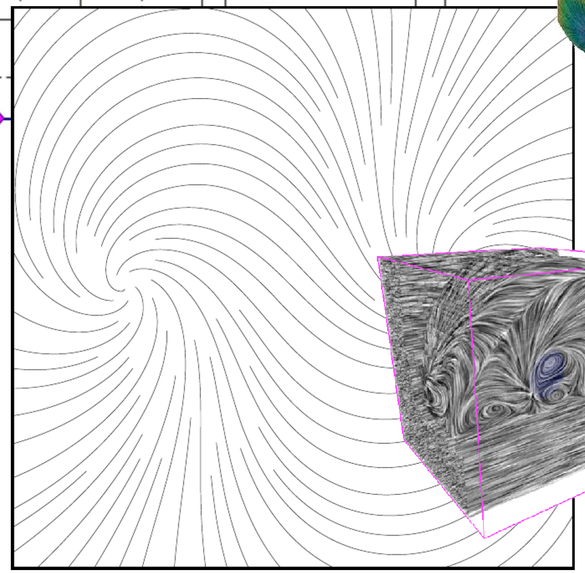
[Kitware]

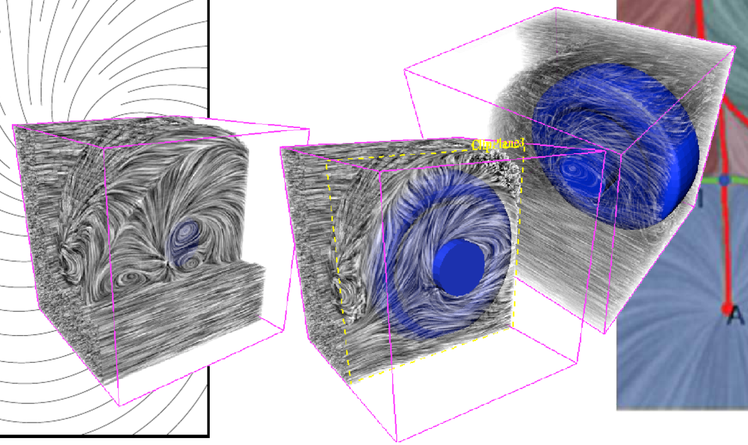


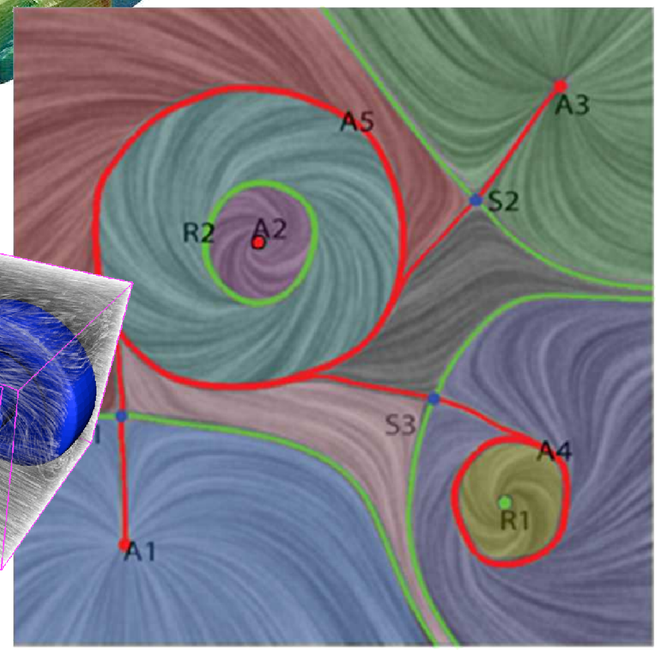


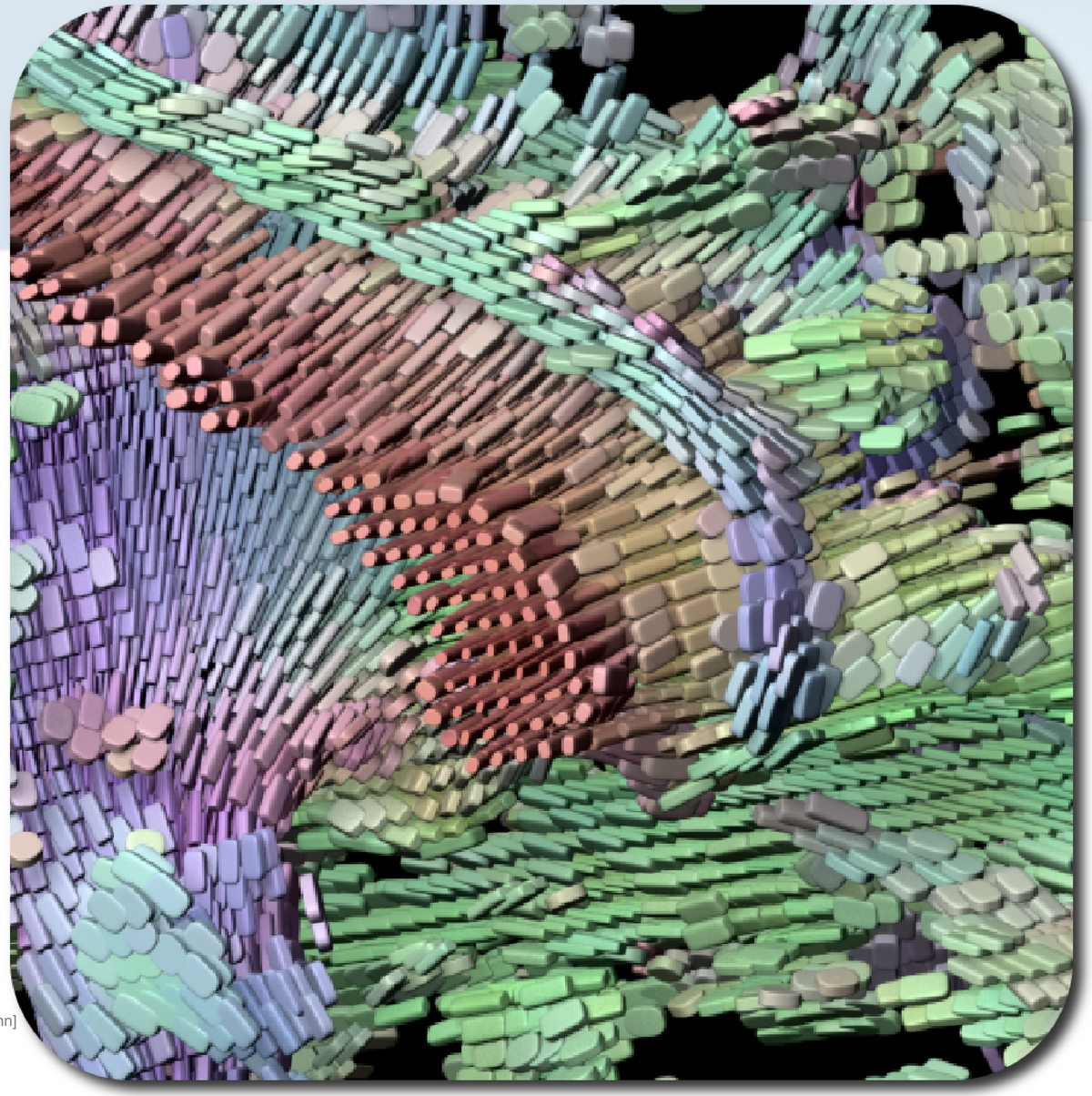












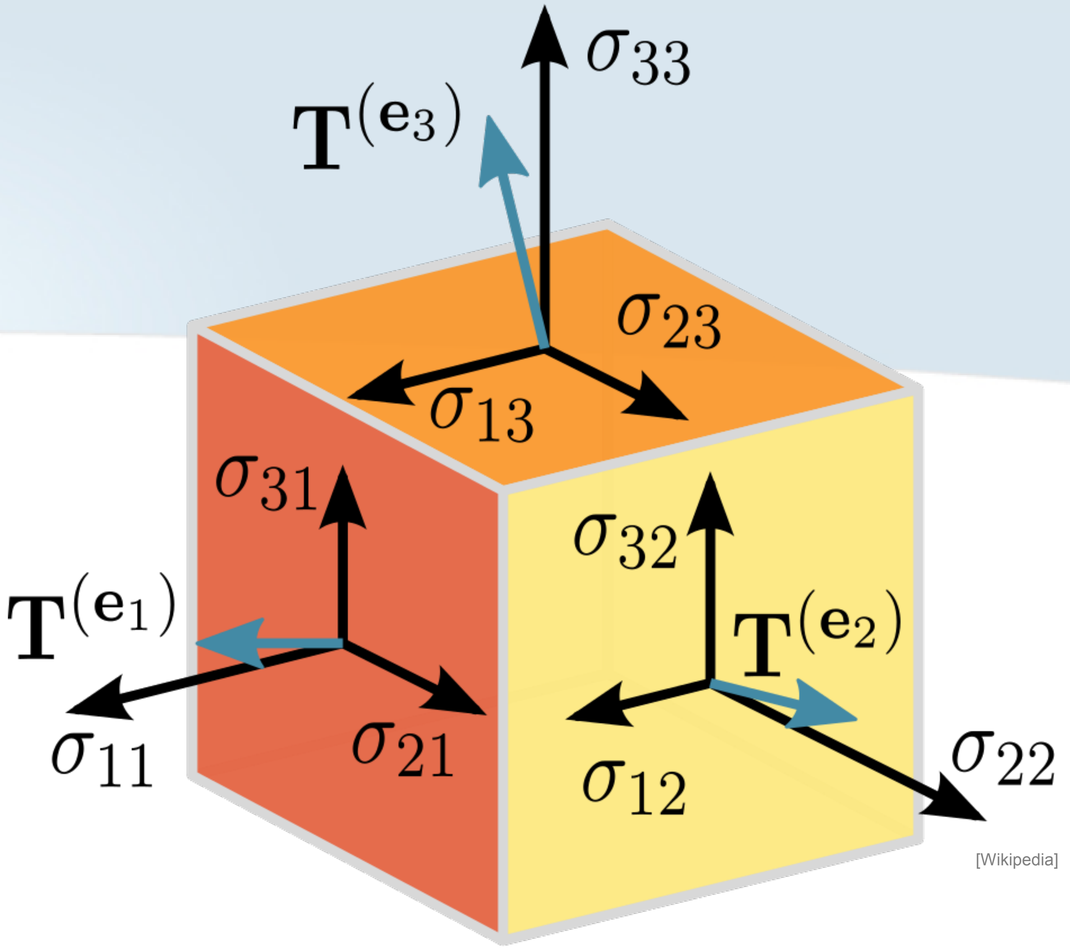
[Kindlmann]

Tensor Field Visualization

Notion of tensor

Notion of tensor

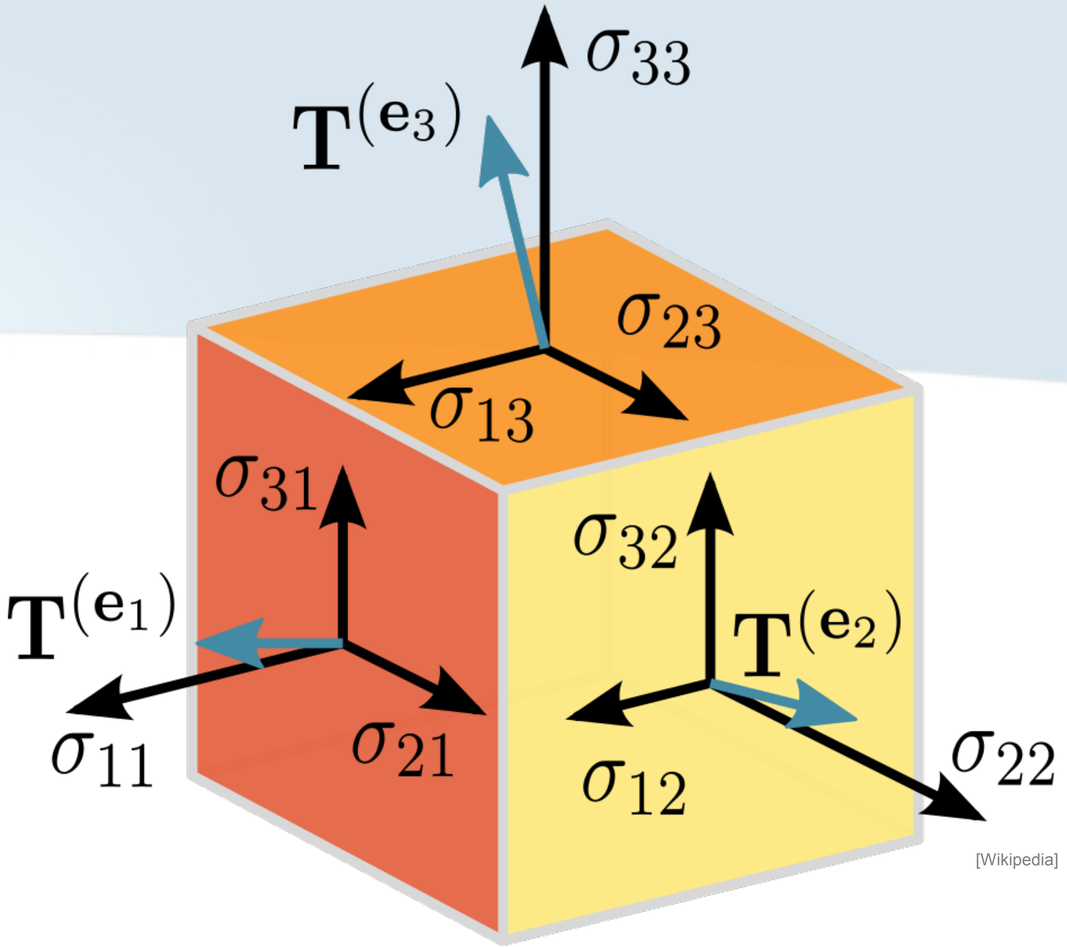
- Generalization of scalars and vectors



[Wikipedia]

Notion of tensor

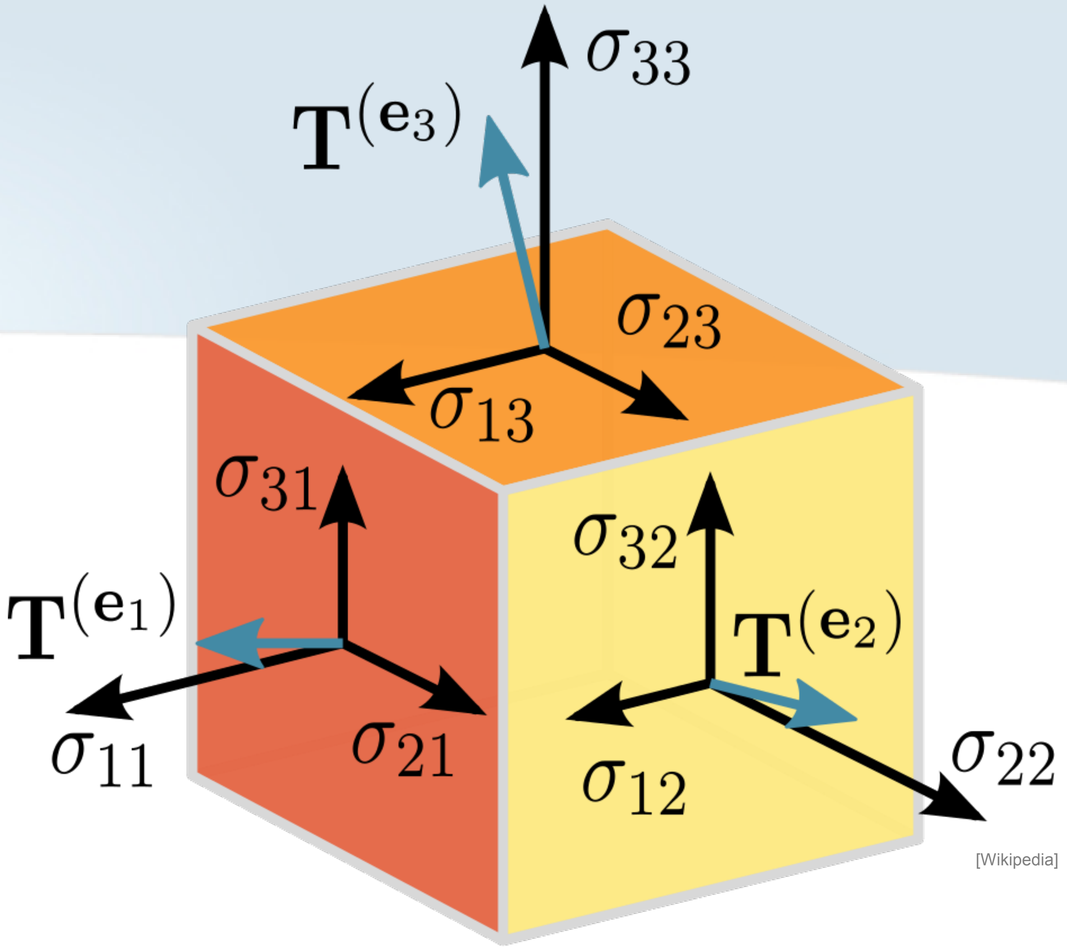
- Generalization of scalars and vectors
- Describes linear relations between



[Wikipedia]

Notion of tensor

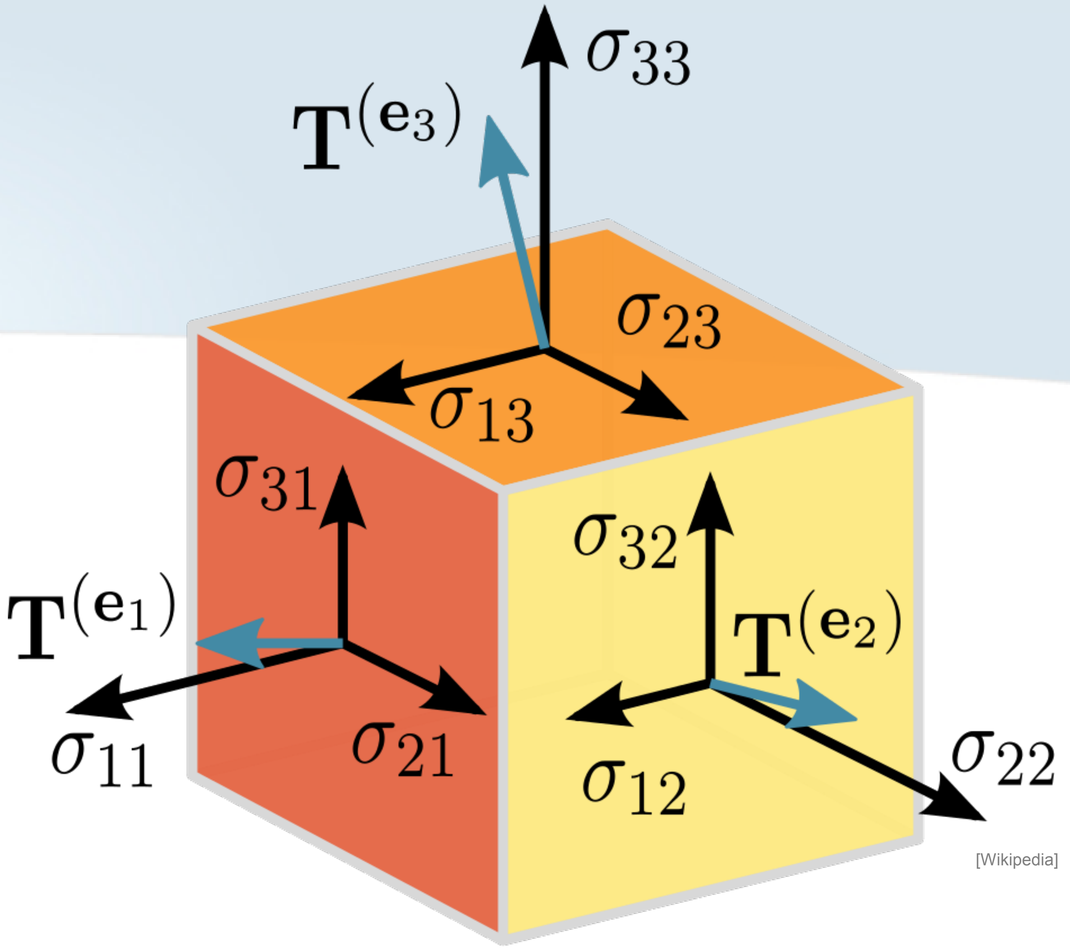
- Generalization of scalars and vectors
- Describes linear relations between
  - Scalars
  - Vectors
  - Or other tensors



[Wikipedia]

Notion of tensor

- Generalization of scalars and vectors
- Describes linear relations between
  - Scalars
  - Vectors
  - Or other tensors

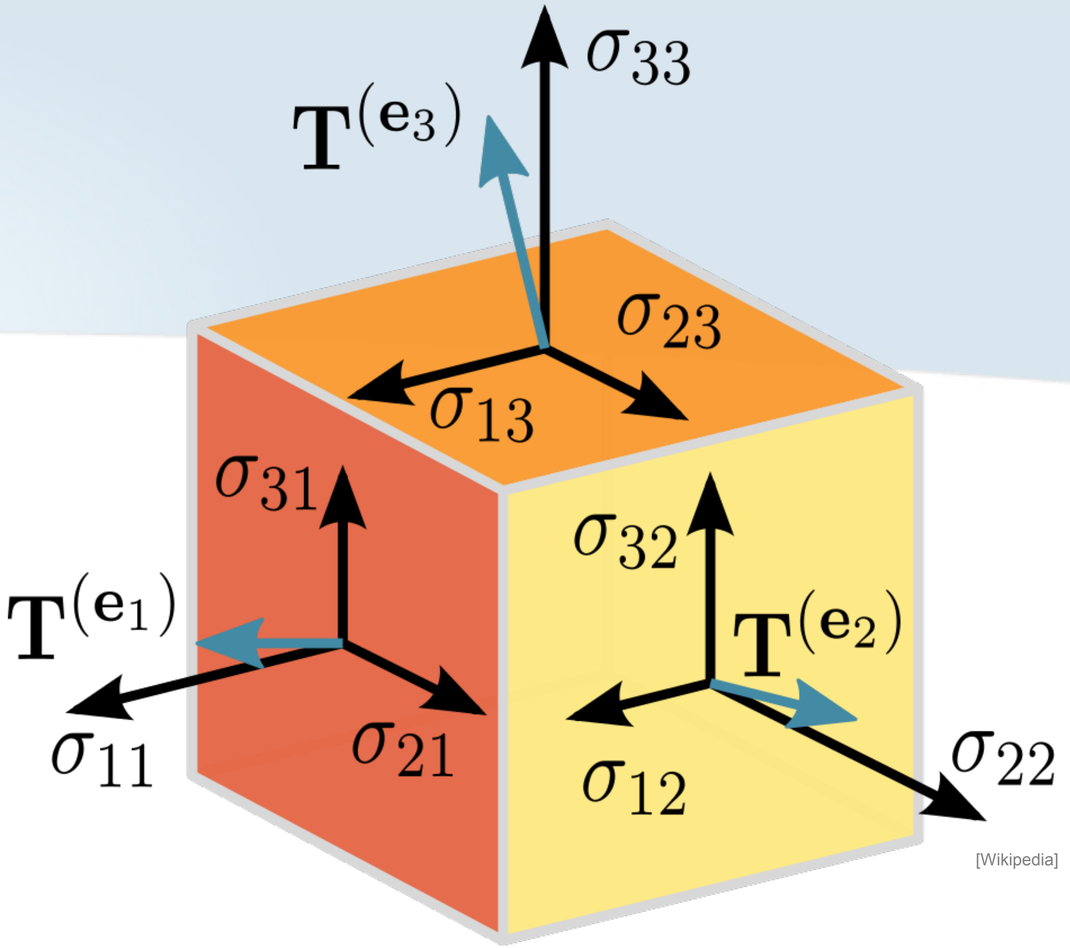


[Wikipedia]

- Multi-dimensional array of numerical values

Notion of tensor

- Generalization of scalars and vectors
- Describes linear relations between
  - Scalars
  - Vectors
  - Or other tensors

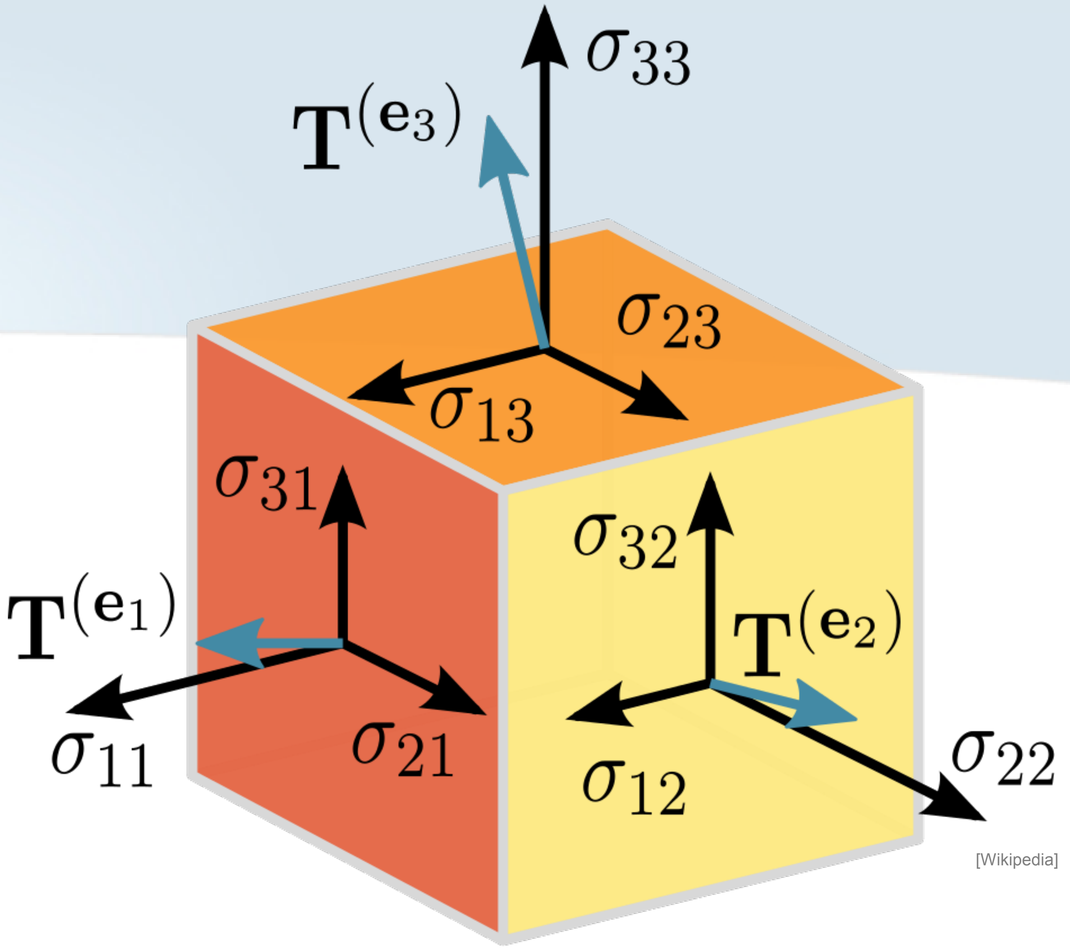


[Wikipedia]

- Multi-dimensional array of numerical values
  - Order of a tensor

Notion of tensor

- Generalization of scalars and vectors
- Describes linear relations between
  - Scalars
  - Vectors
  - Or other tensors

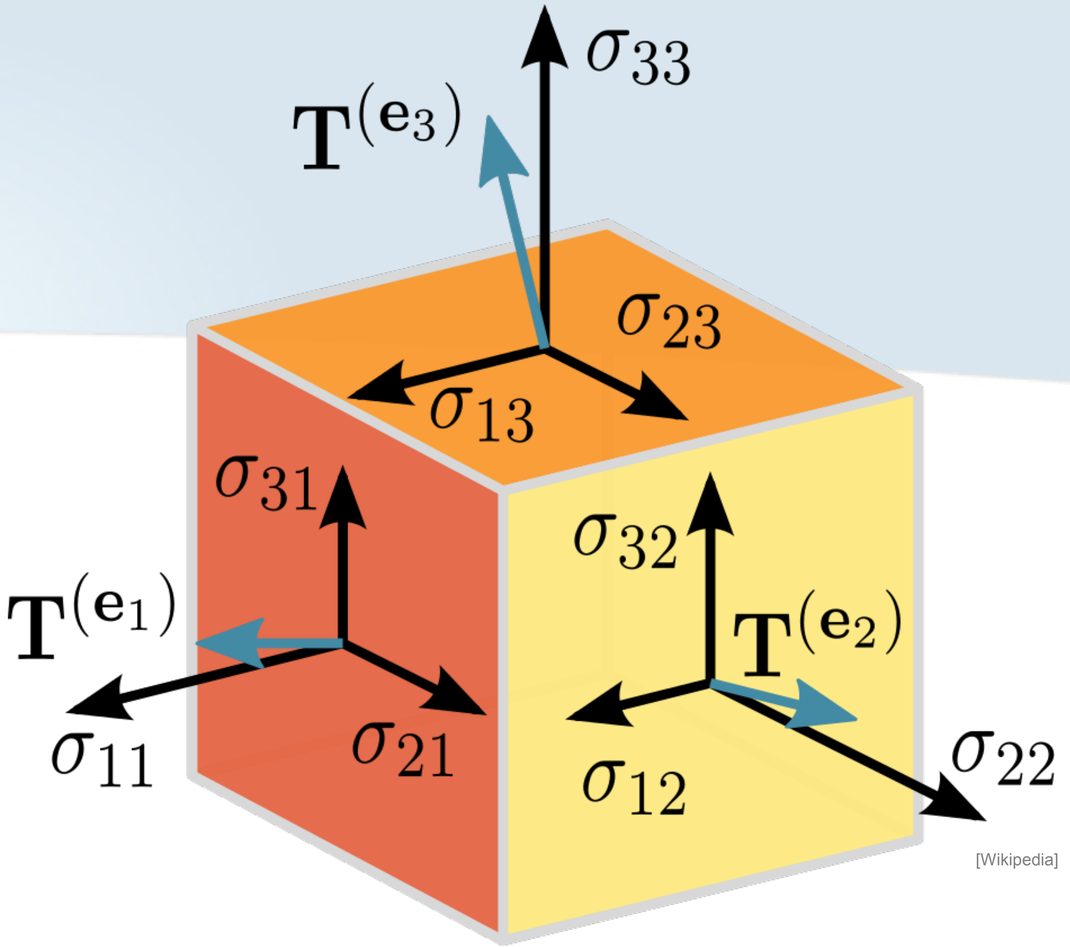


[Wikipedia]

- Multi-dimensional array of numerical values
  - Order of a tensor
    - Number of dimensions

Notion of tensor

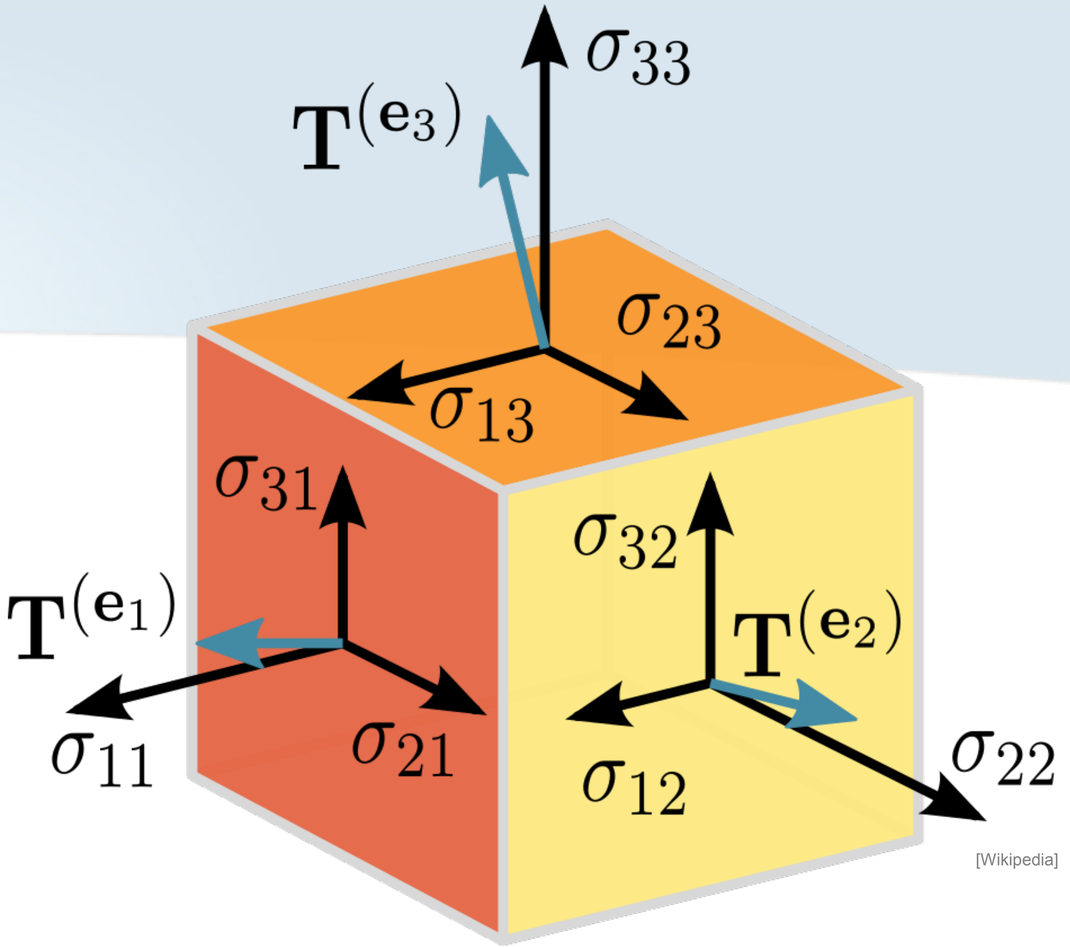
- Order of a tensor



[Wikipedia]

Notion of tensor

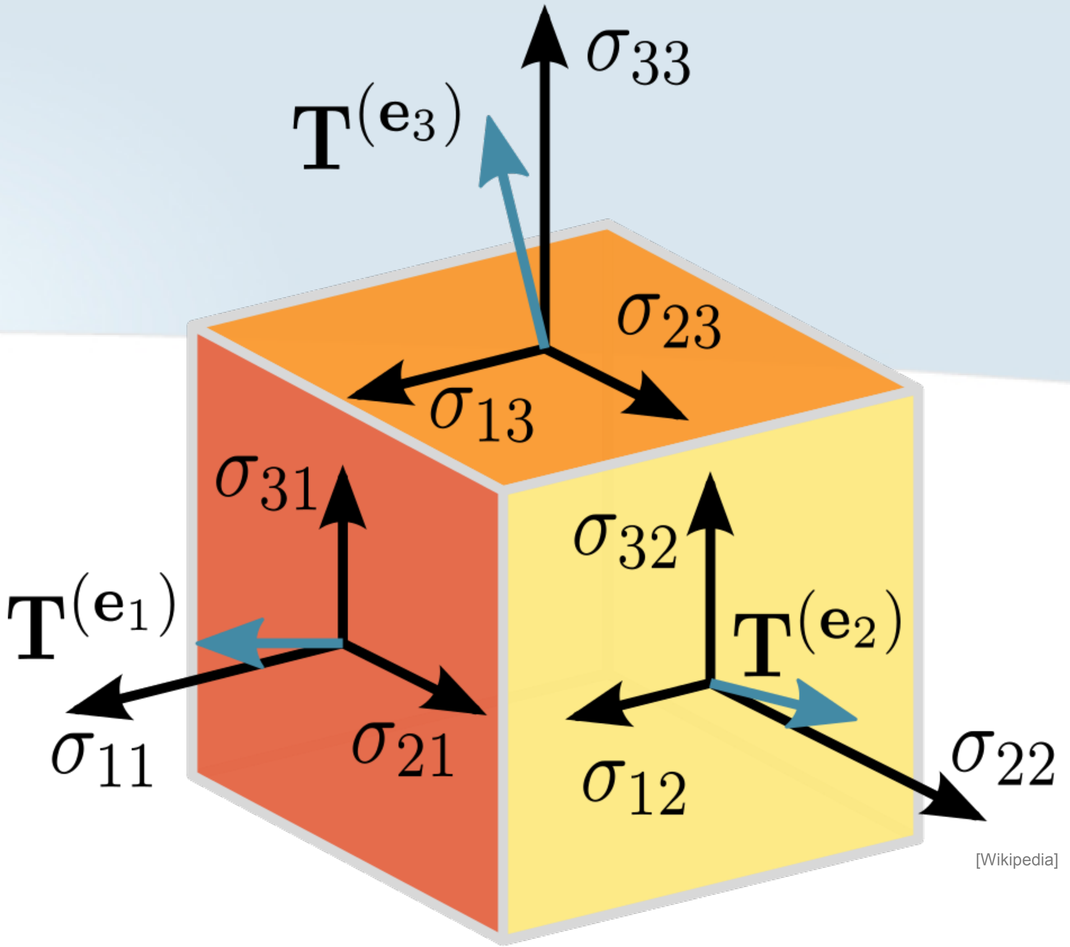
- Order of a tensor
  - Scalar value: 0^{th} order tensor



[Wikipedia]

Notion of tensor

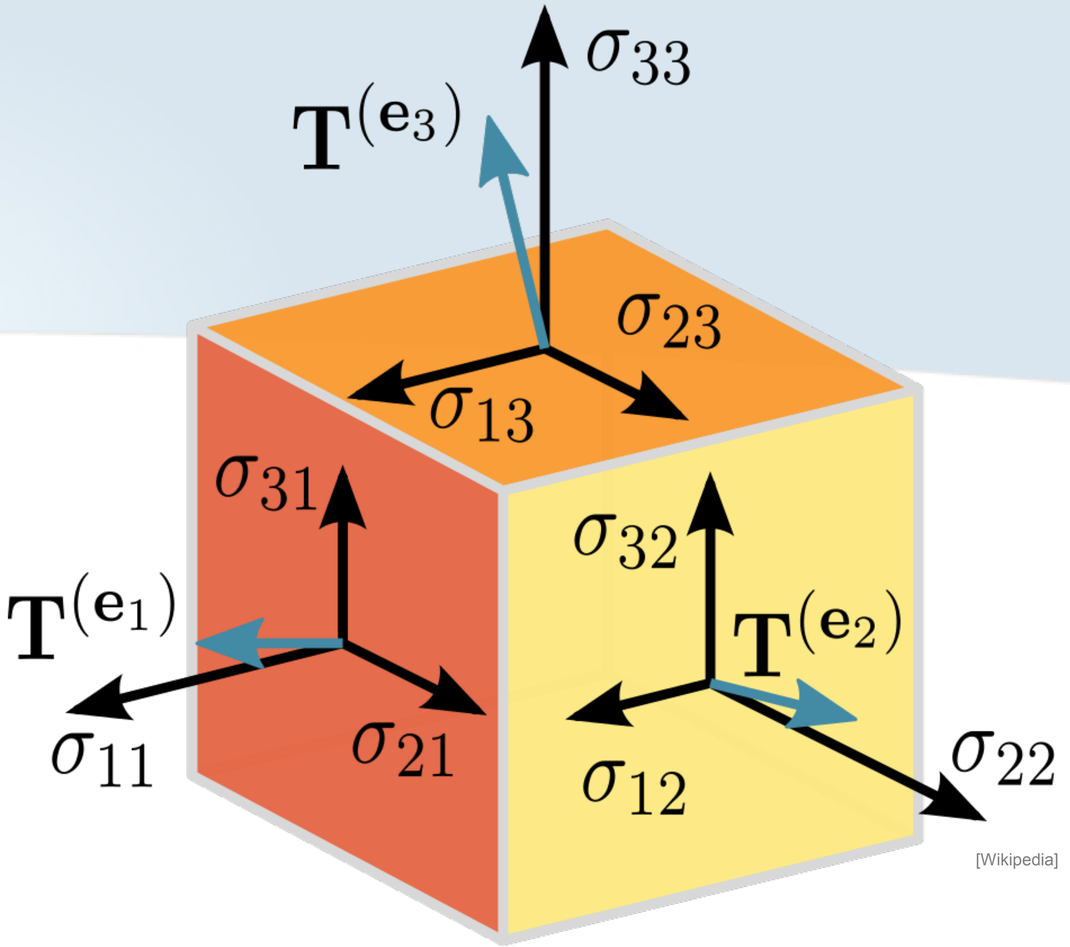
- Order of a tensor
  - Scalar value: 0th order tensor
  - Vector field: 1st order tensor



[Wikipedia]

Notion of tensor

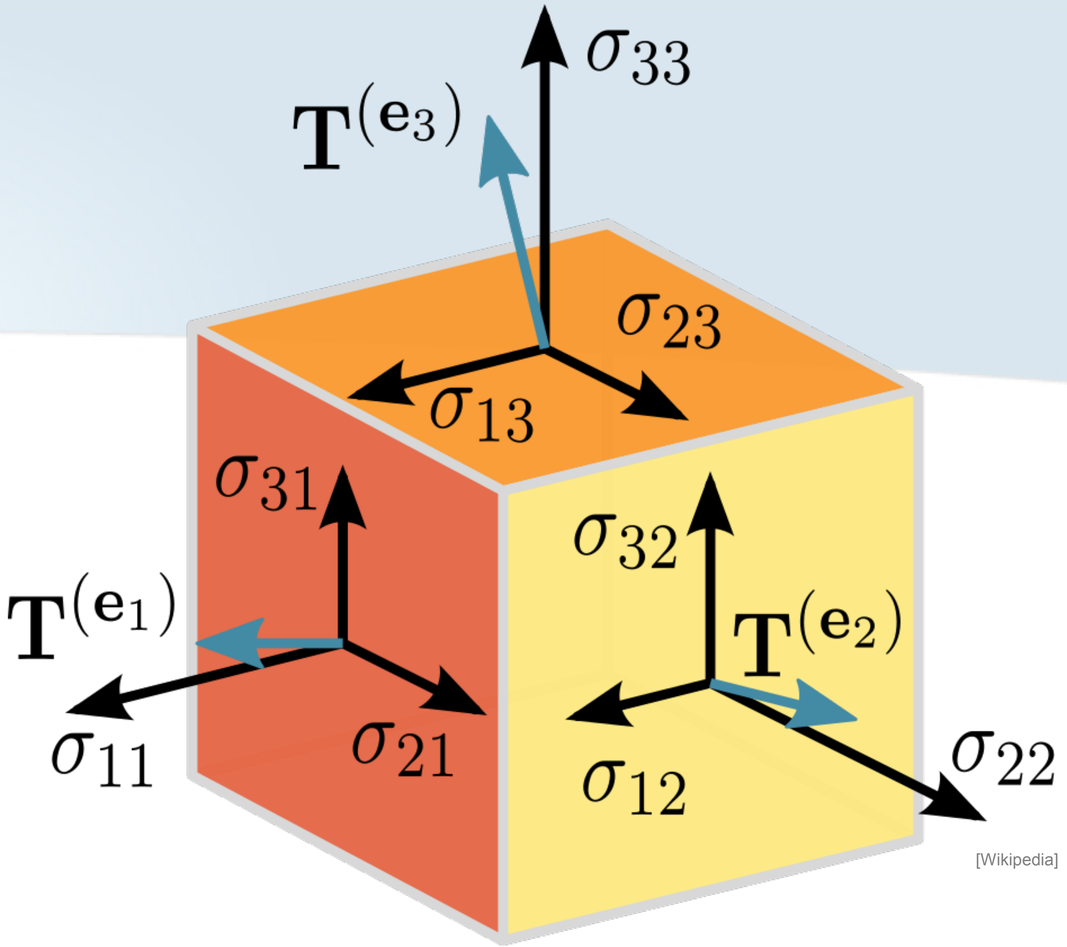
- Order of a tensor
  - Scalar value: 0th order tensor
  - Vector field: 1st order tensor
  - (dxd)-matrix: 2nd order tensor



[Wikipedia]

Notion of tensor

- Order of a tensor
  - Scalar value: 0th order tensor
  - Vector field: 1st order tensor
  - (dxd)-matrix: 2nd order tensor



[Wikipedia]

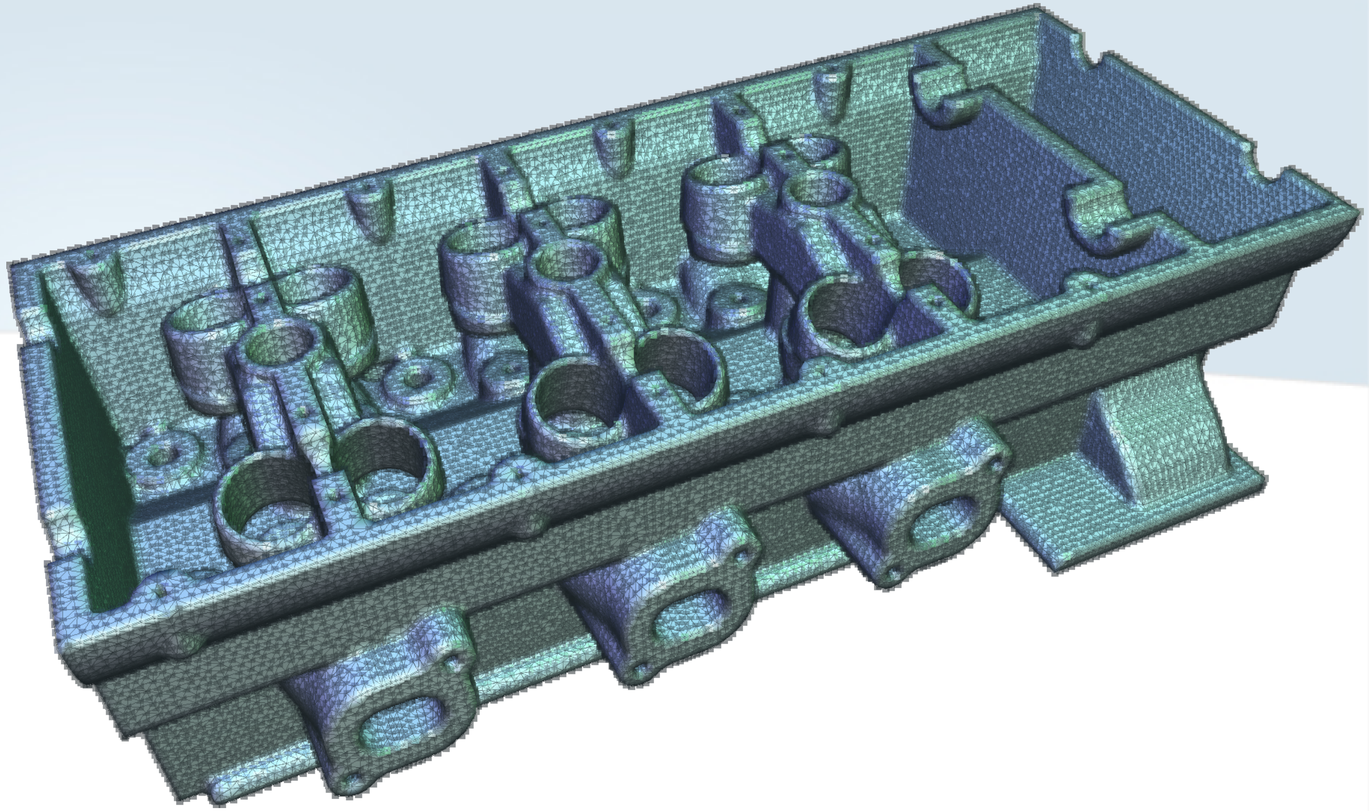
- Here, **mostly symmetric 2nd order tensors**

In practice

- Given a domain $\mathcal{D}$

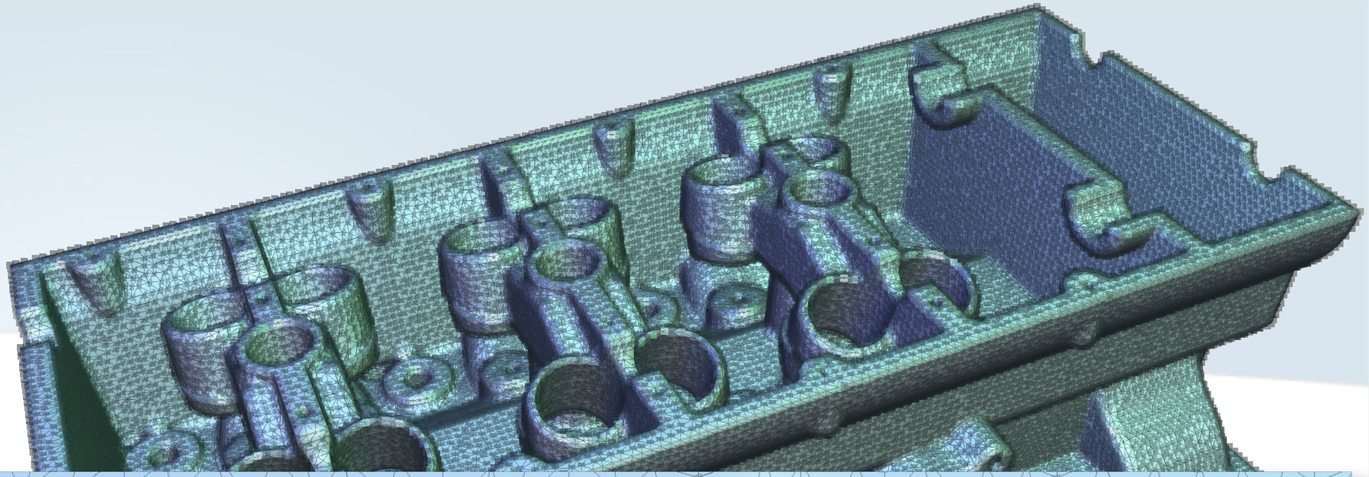
In practice

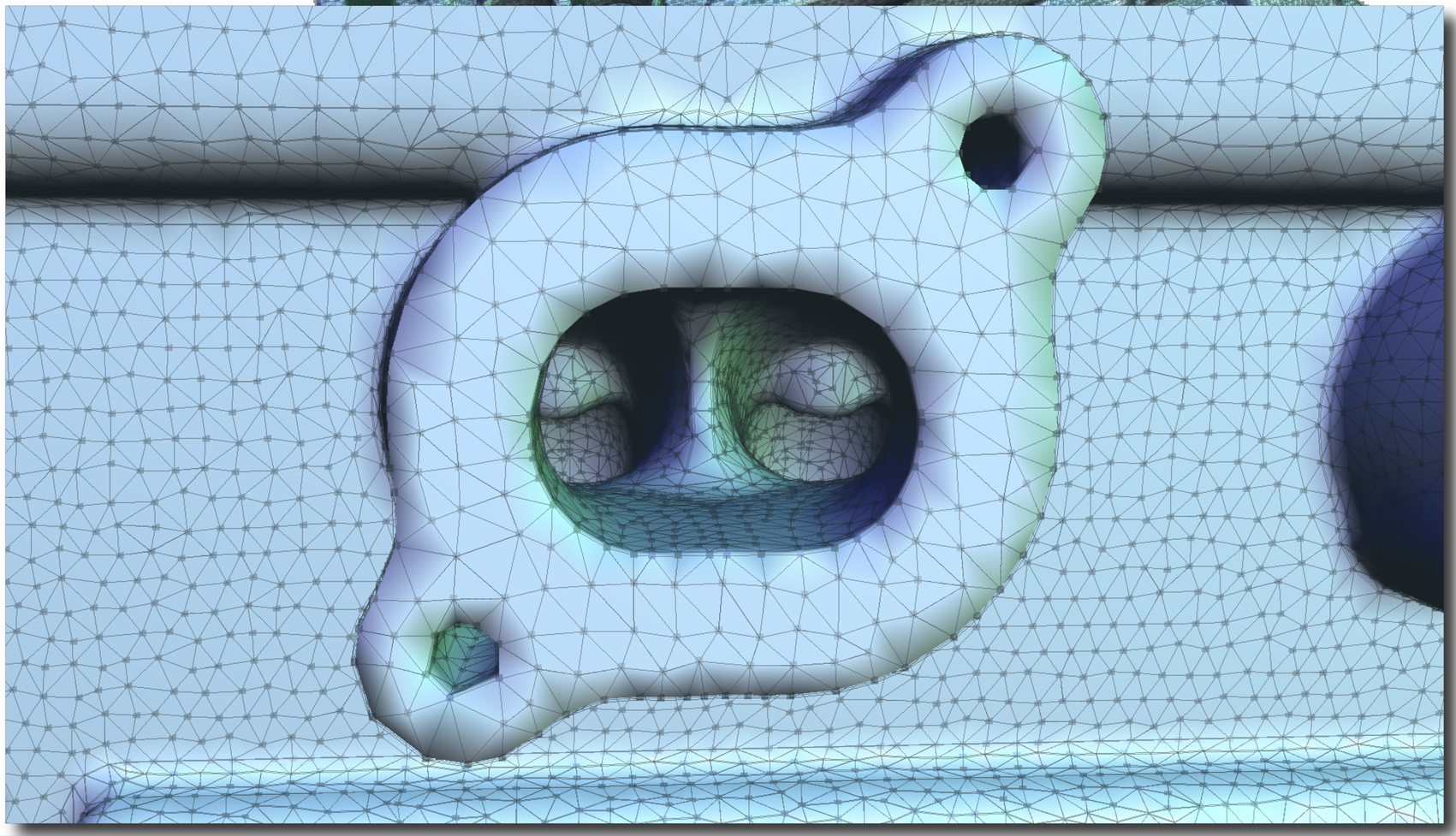
- Given a domain $\mathcal{D}$



In practice

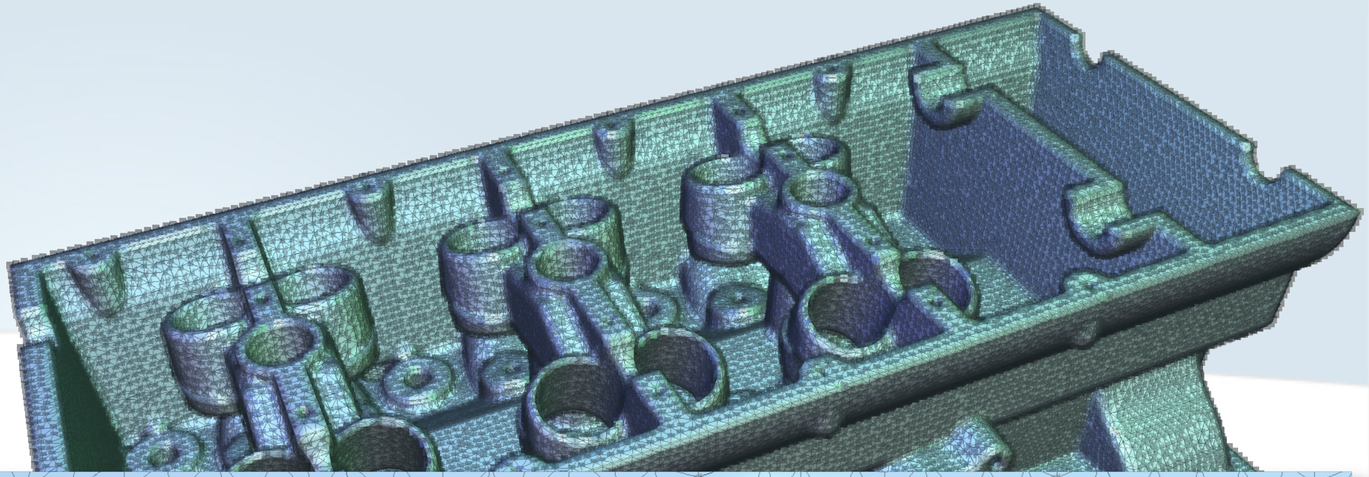
- Given a domain $\mathcal{D}$

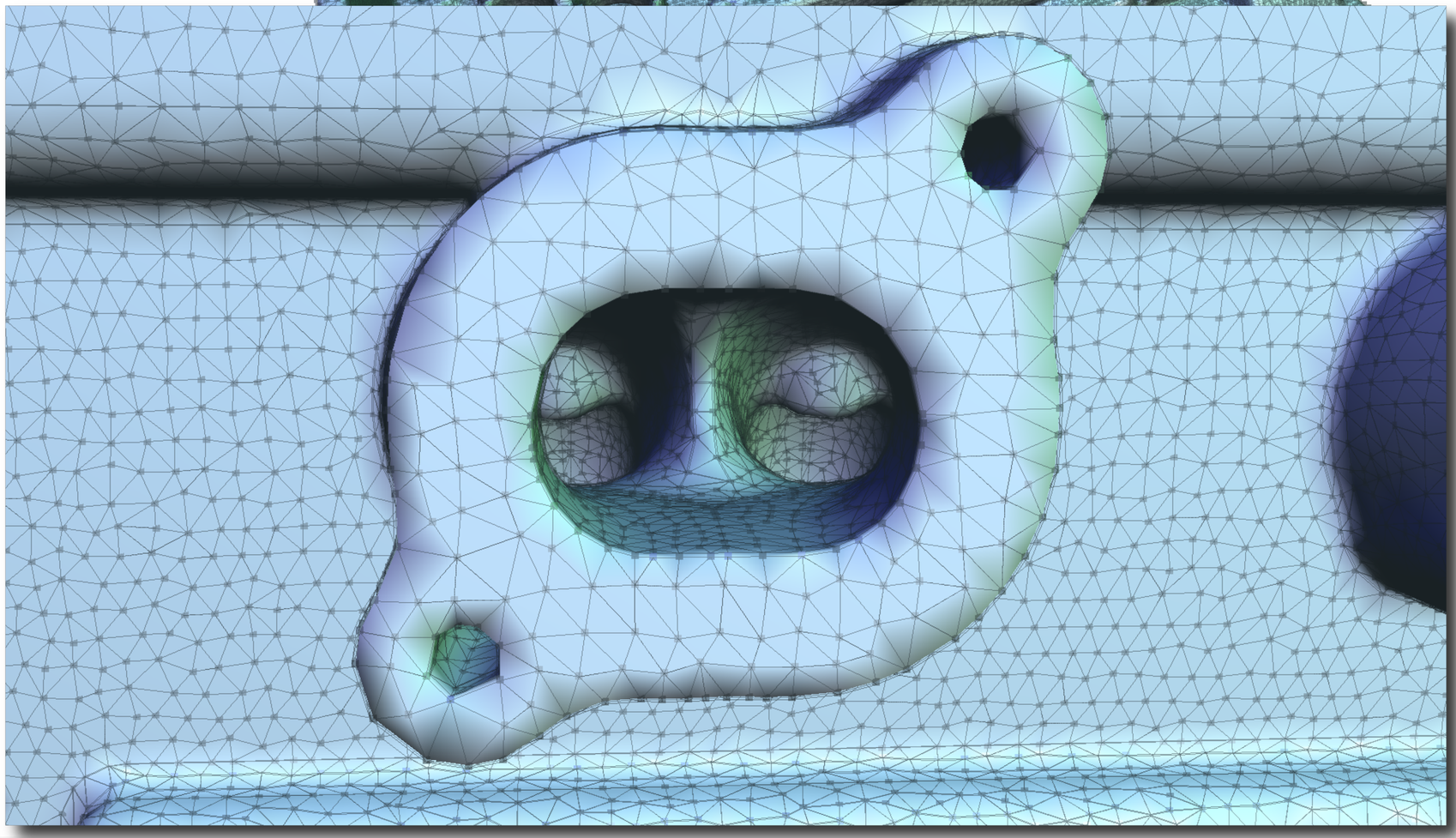




In practice

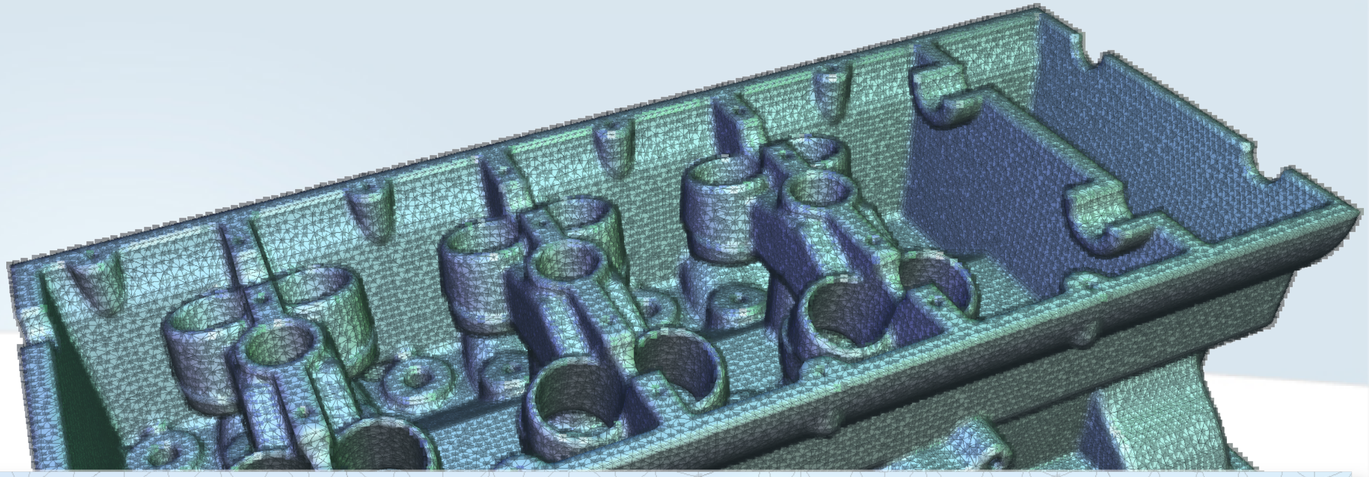
- Given a domain $\mathcal{D}$
- For each vertex v

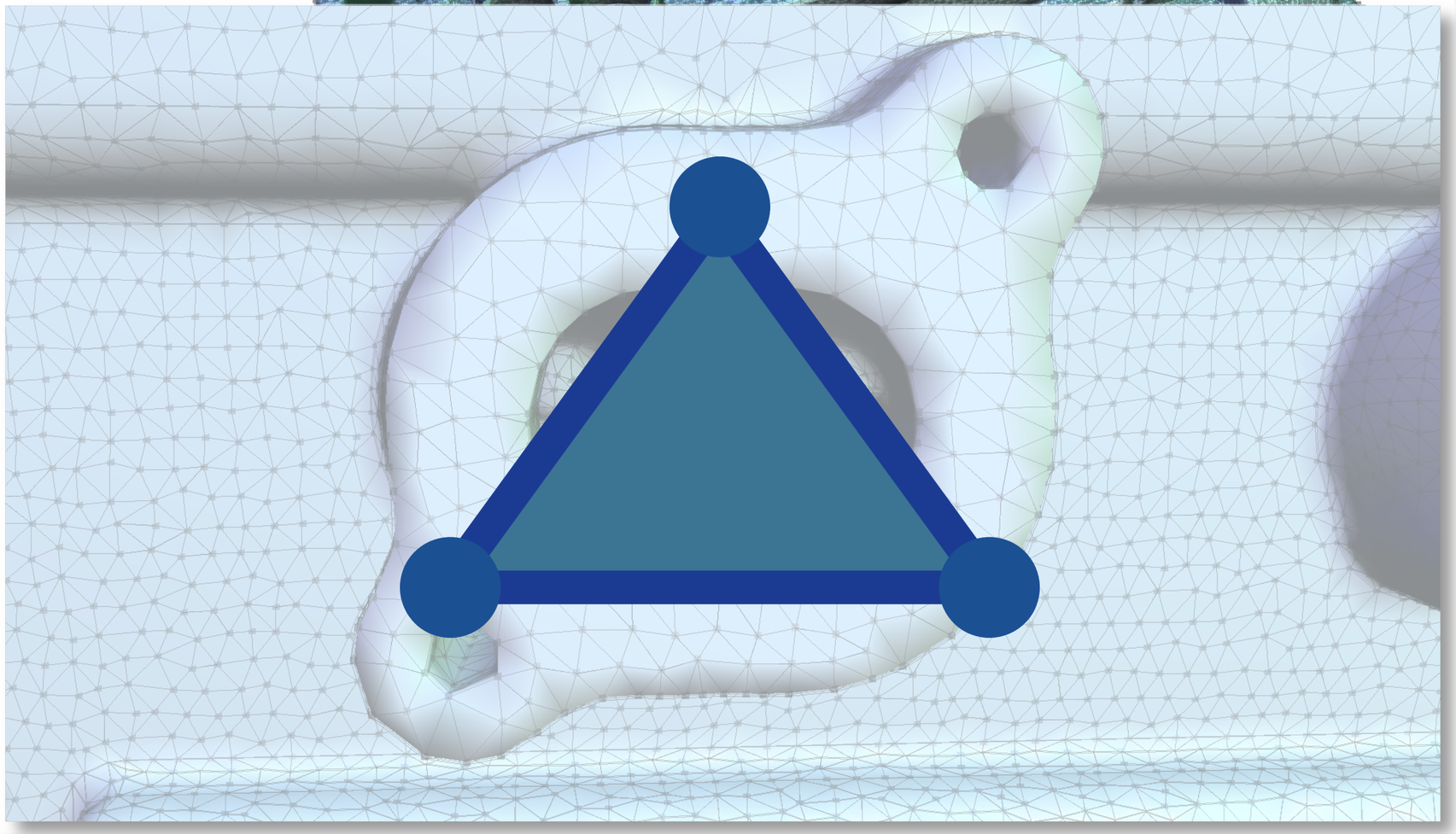




In practice

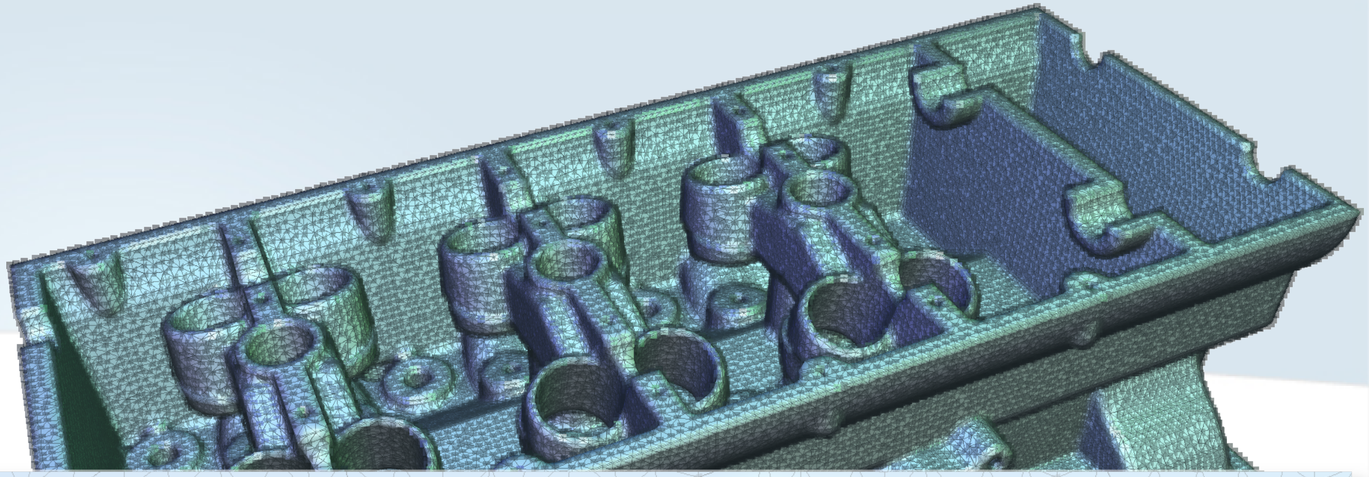
- Given a domain $\mathcal{D}$
- For each vertex v

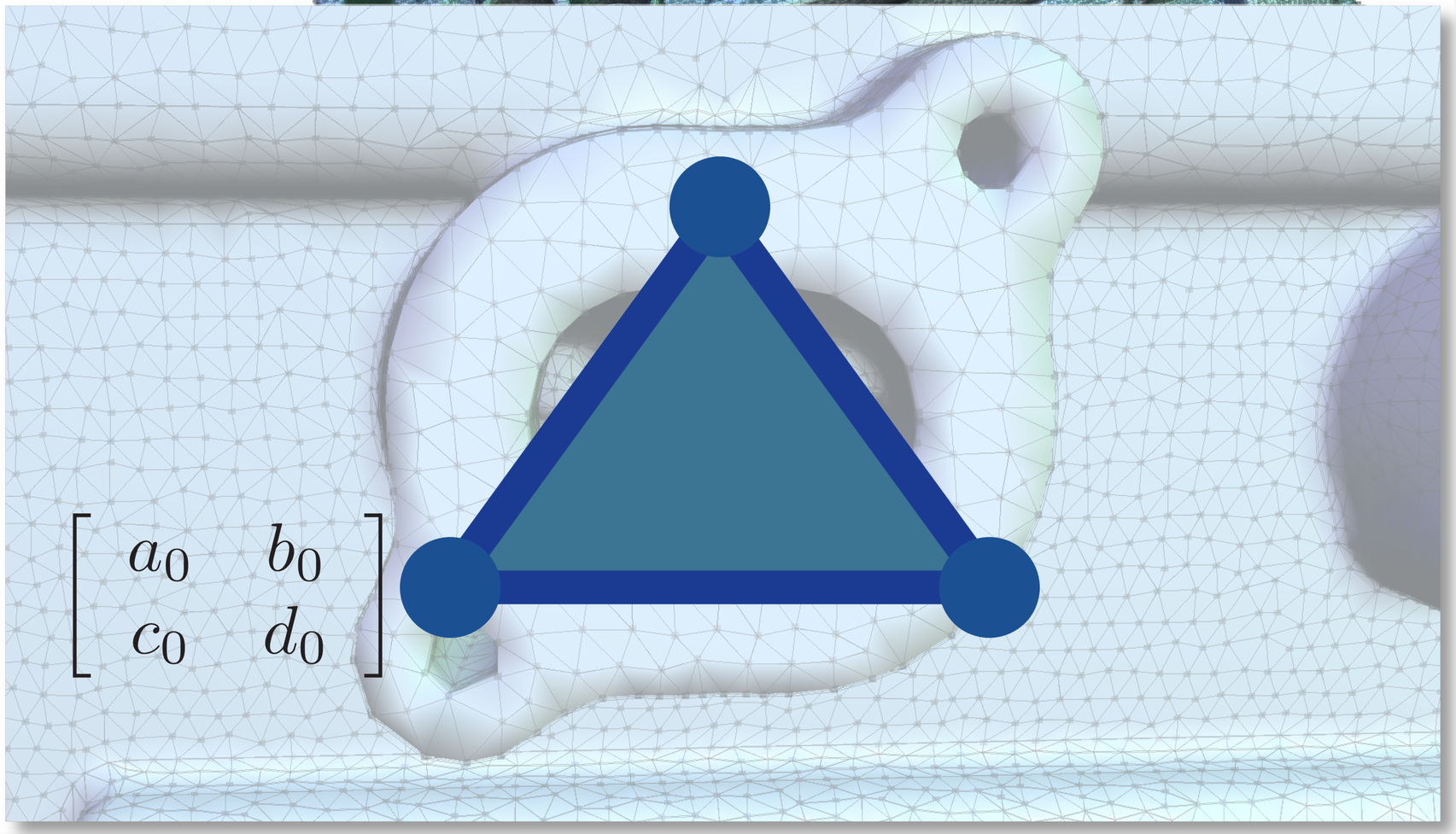




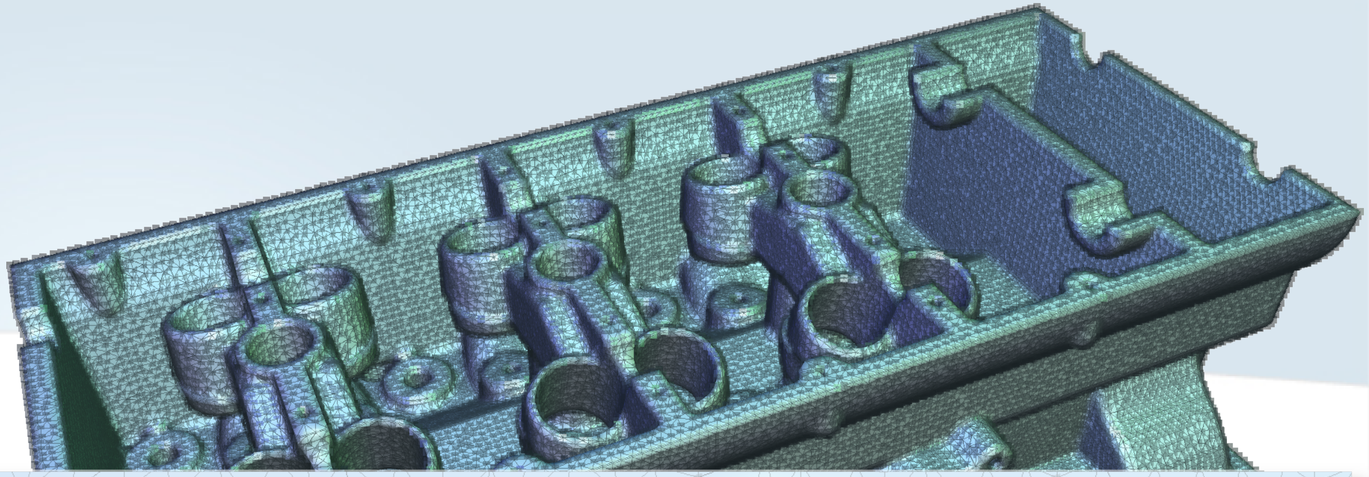
In practice

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

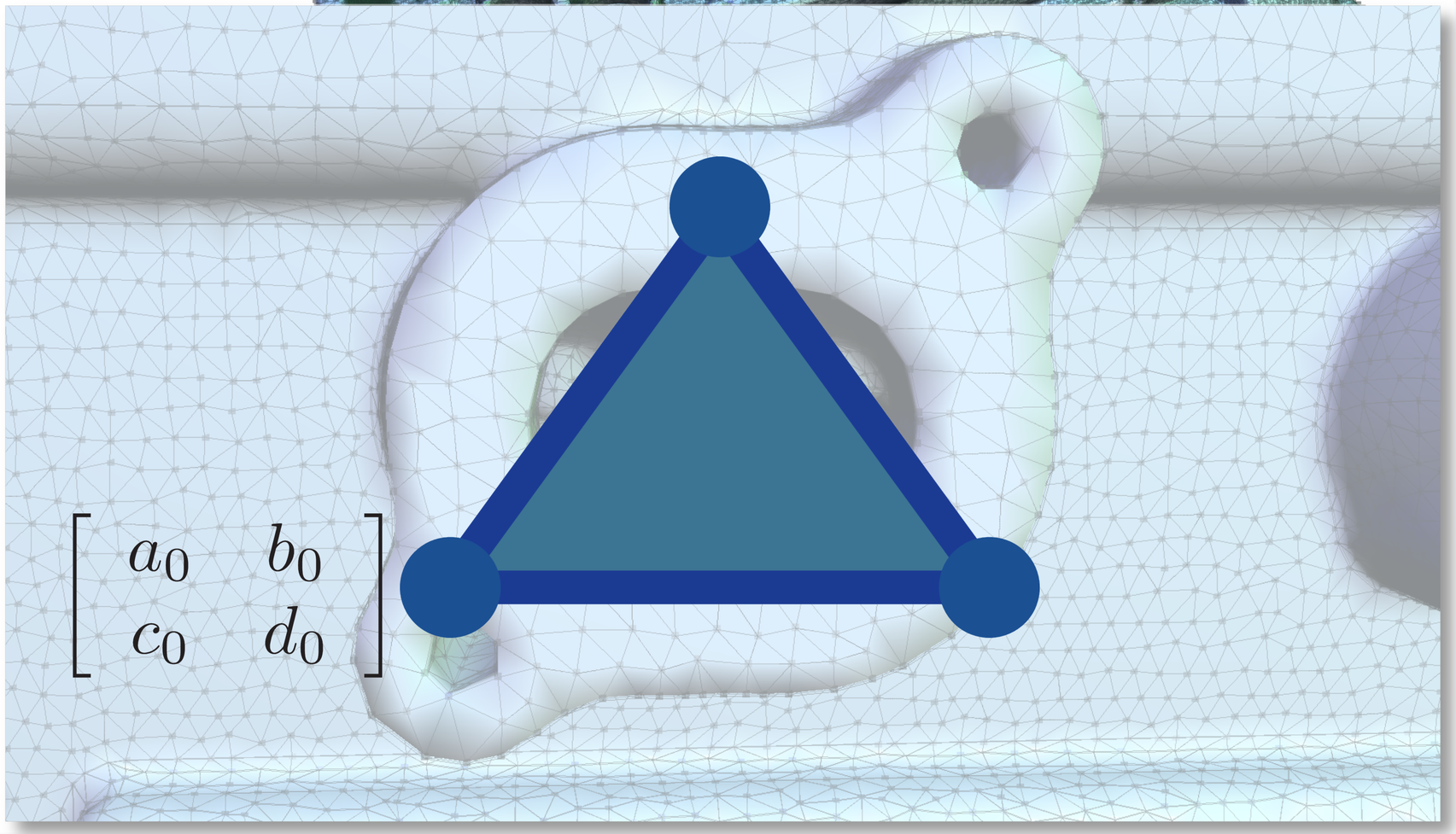




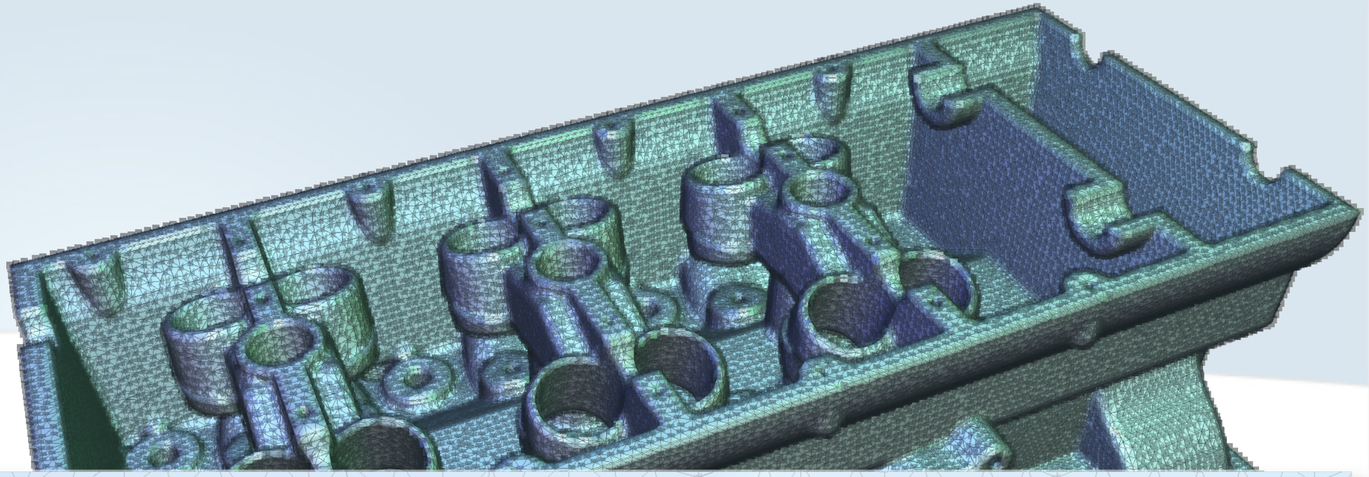
In practice



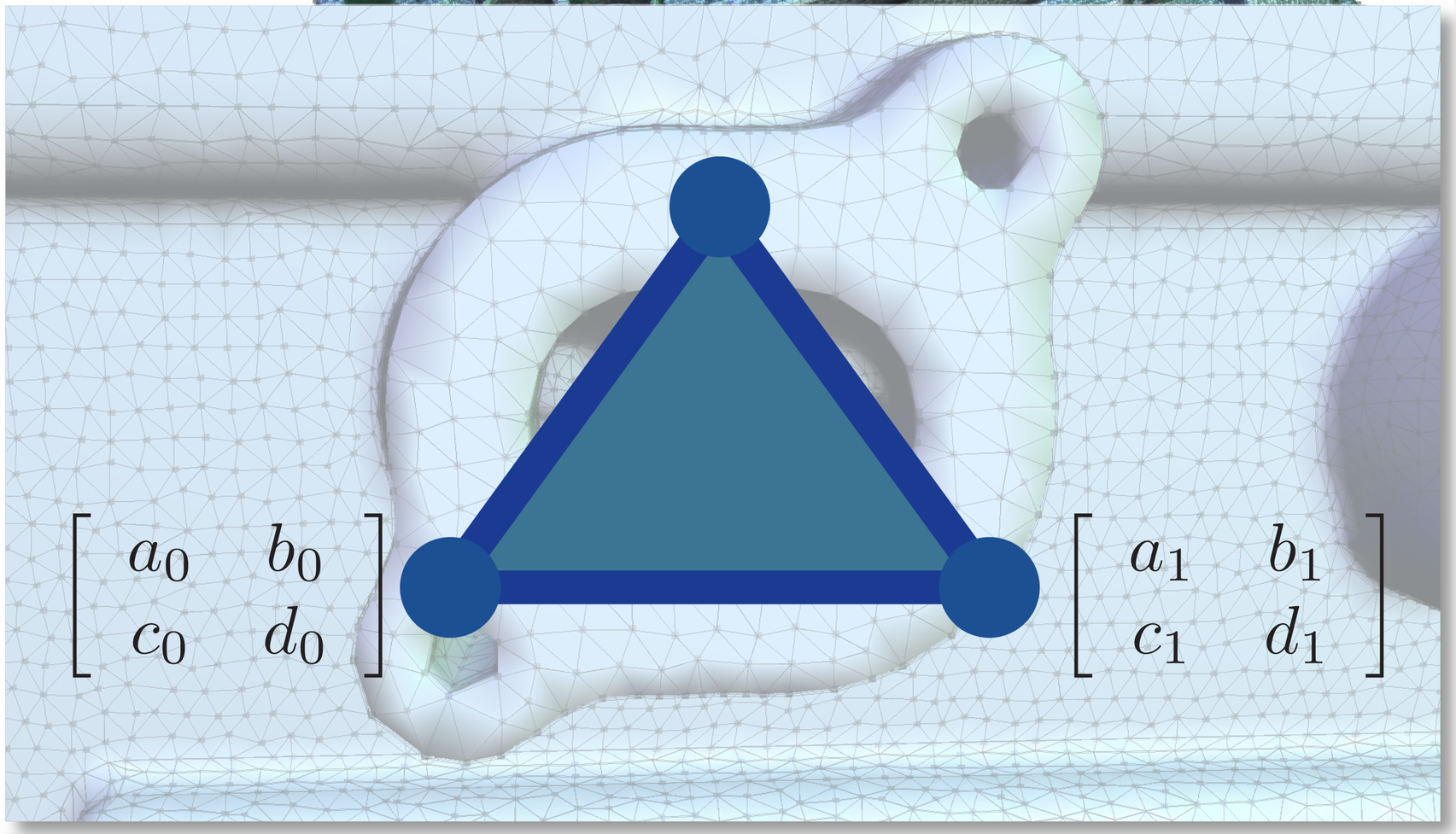
- Given a domain $\mathcal{D}$
- For each vertex v
- One matrix $f(v)$
  - (dxd)-matrix
  - d: dimension of $\mathcal{D}$



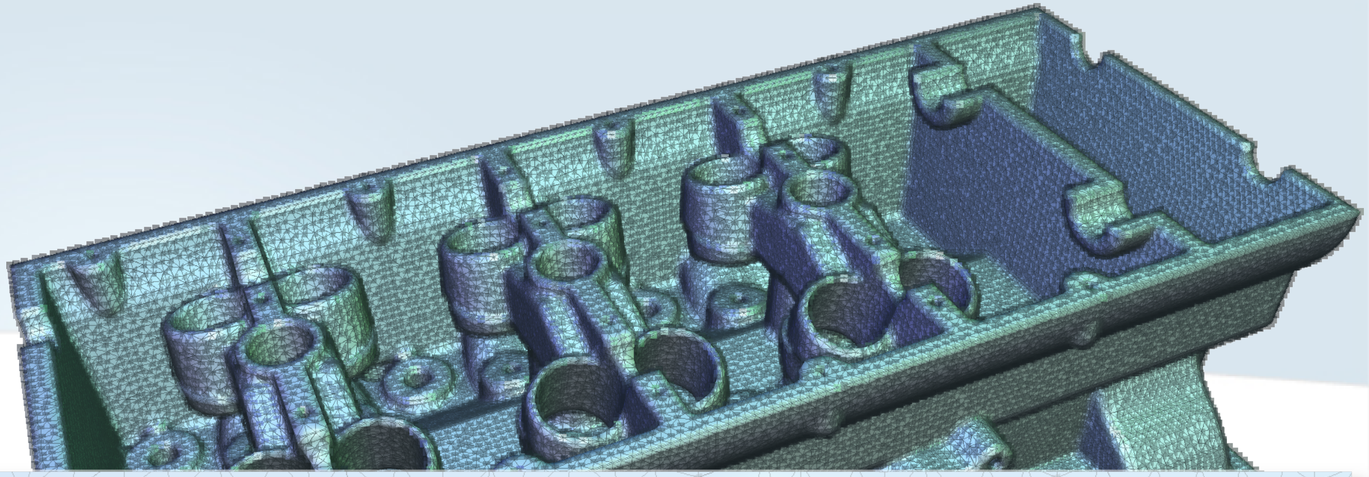
In practice



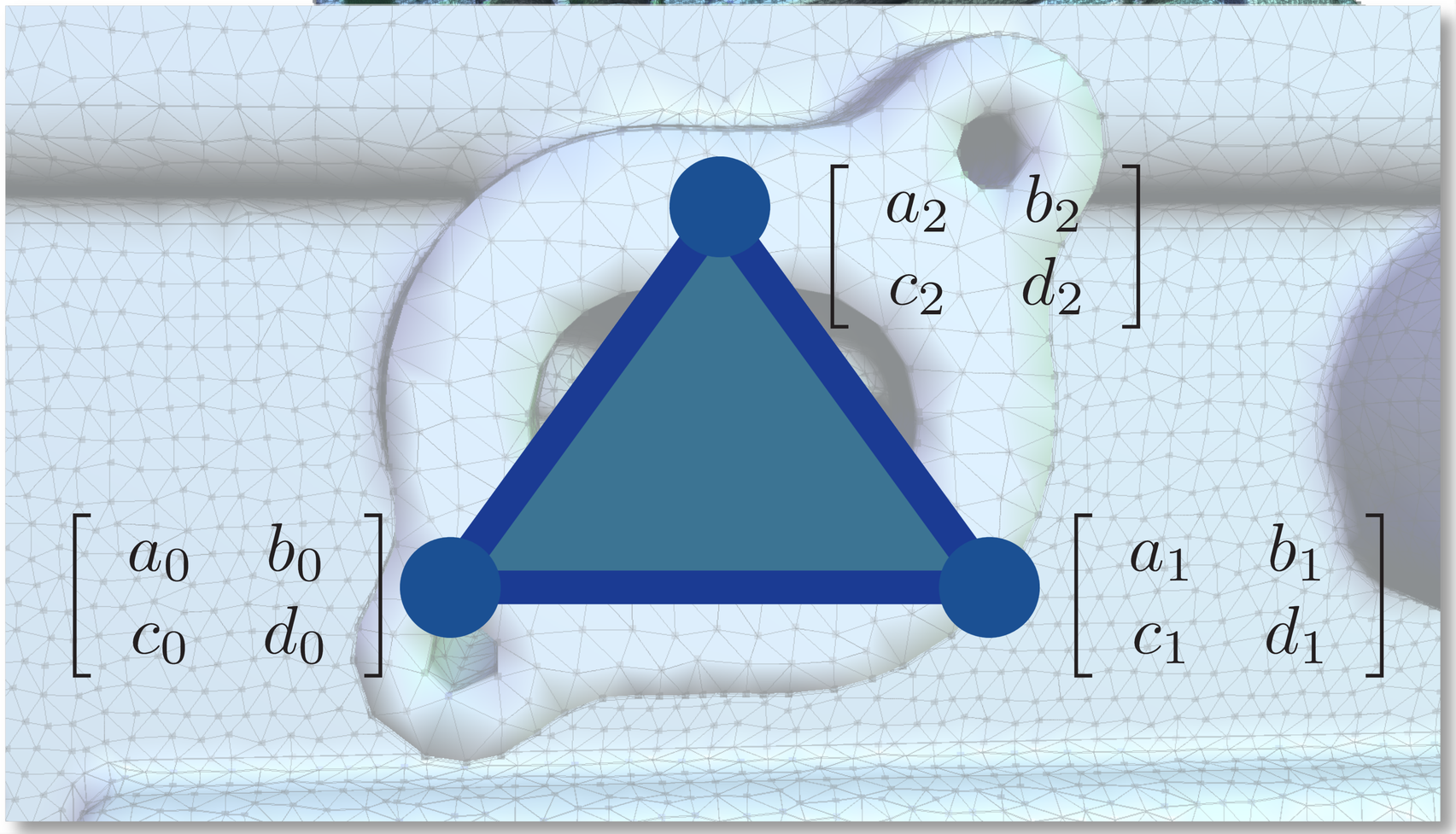
- Given a domain $\mathcal{D}$
- For each vertex v
- One matrix $f(v)$
  - (dxd)-matrix
  - d: dimension of $\mathcal{D}$



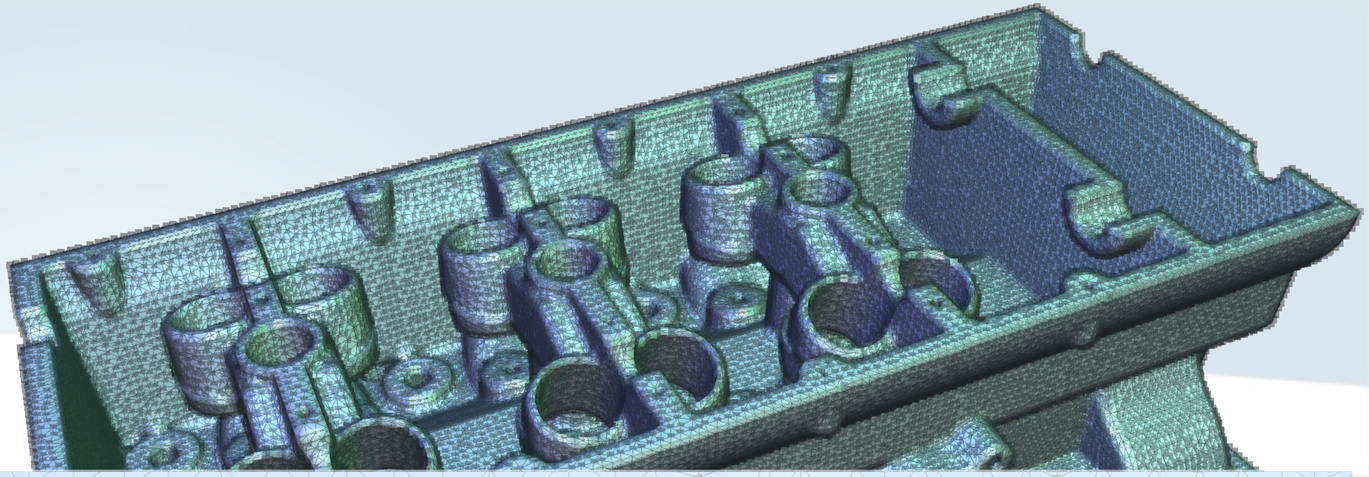
In practice



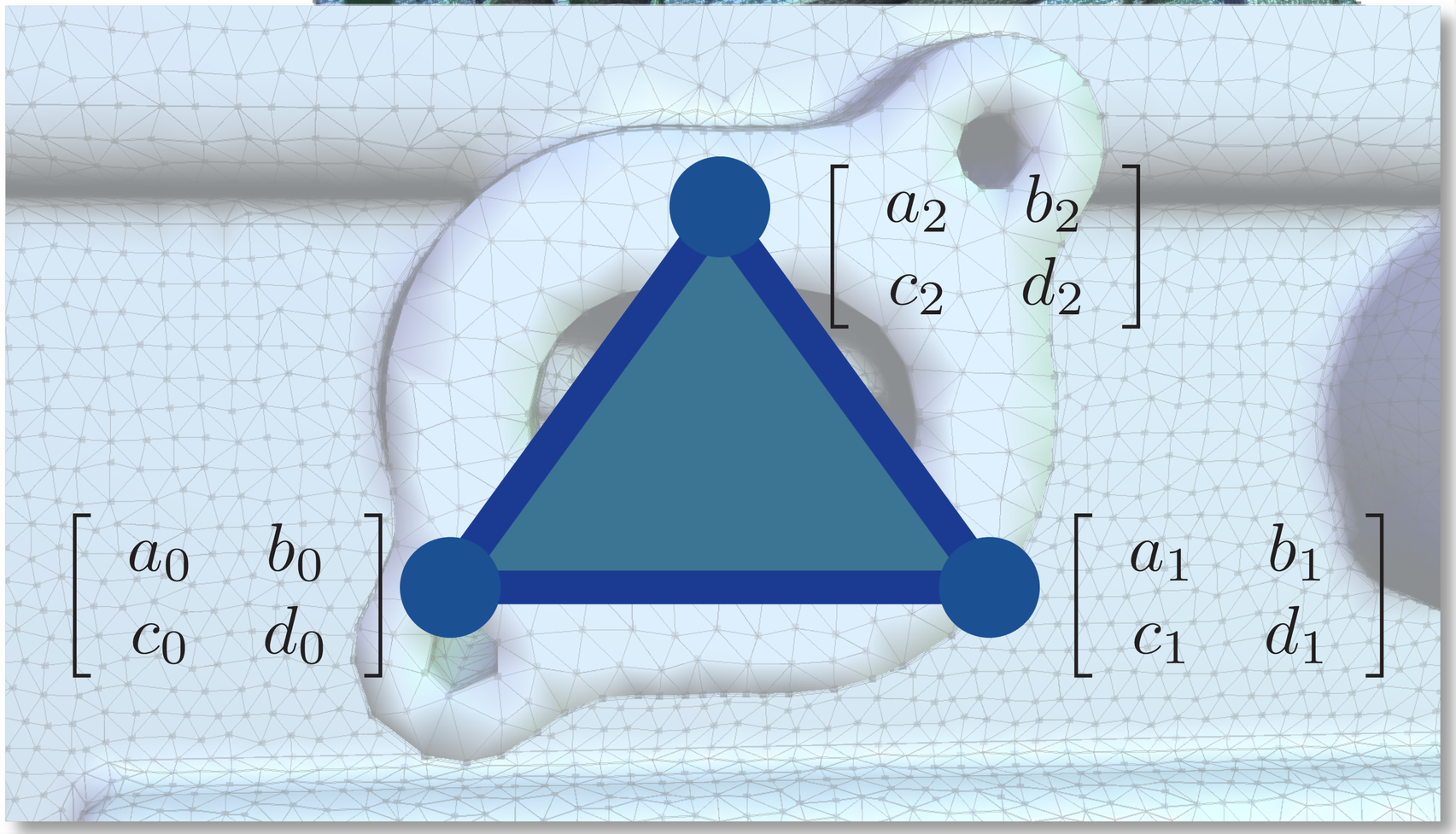
- Given a domain $\mathcal{D}$
- For each vertex v
- One matrix $f(v)$
  - (dxd)-matrix
  - d: dimension of $\mathcal{D}$



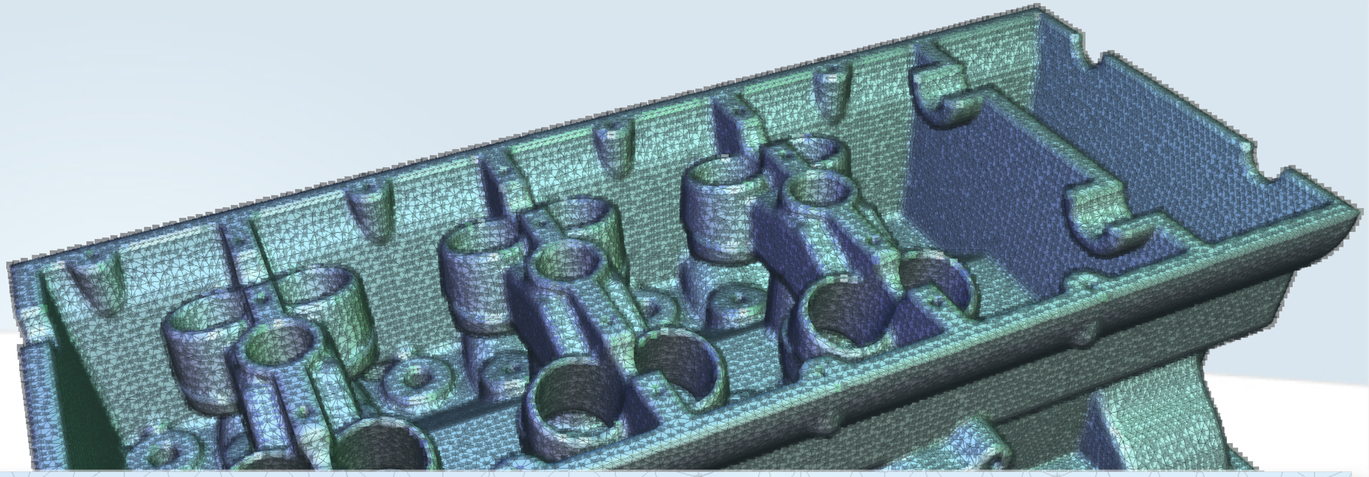
In practice



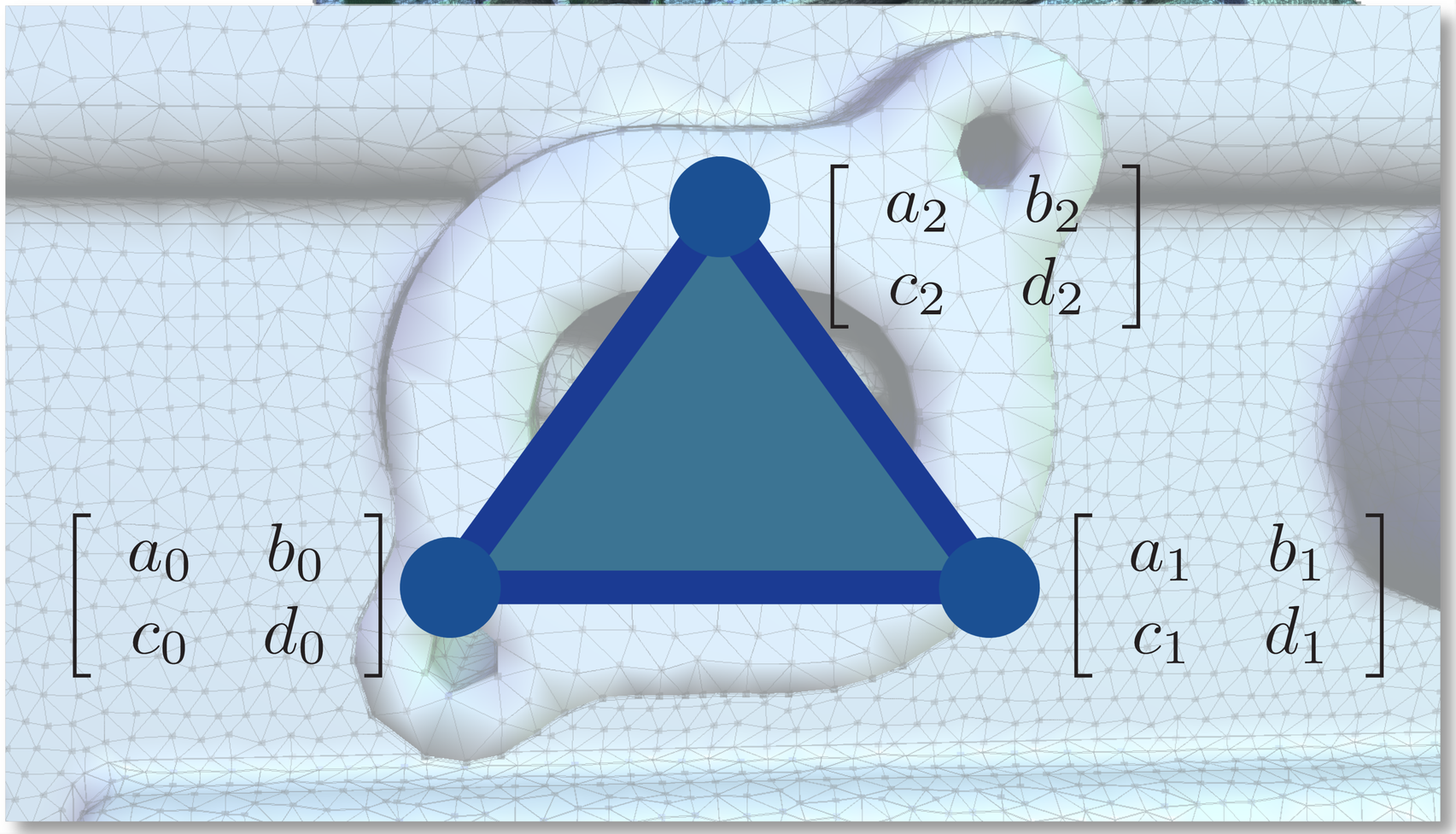
- Given a domain $\mathcal{D}$
- For each vertex v
- One matrix $f(v)$
  - (dxd)-matrix
  - d: dimension of $\mathcal{D}$
- Interpolation on the other simplices



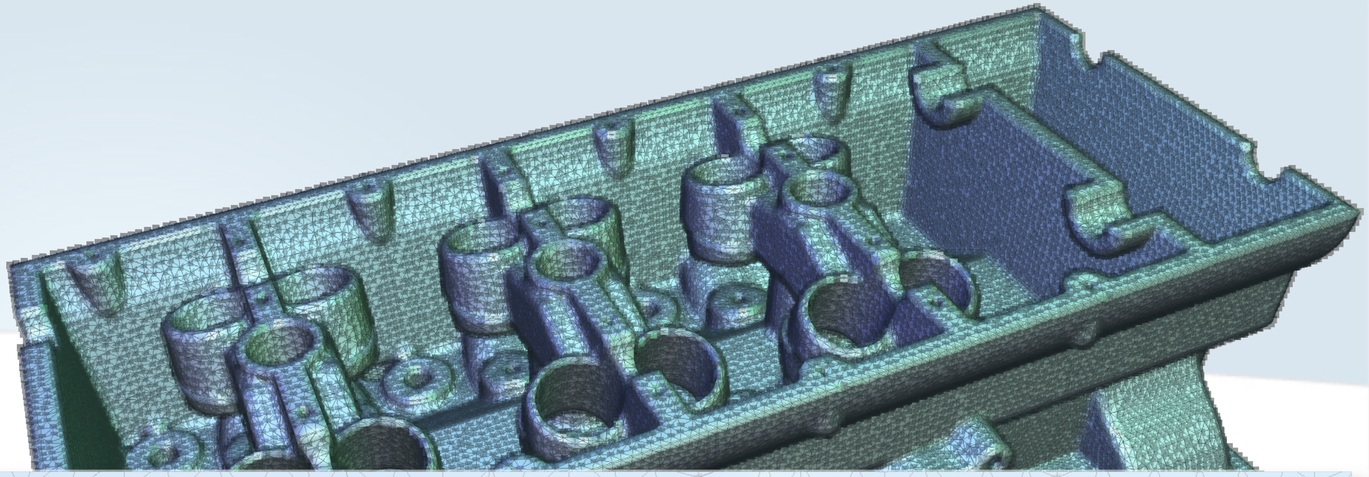
In practice



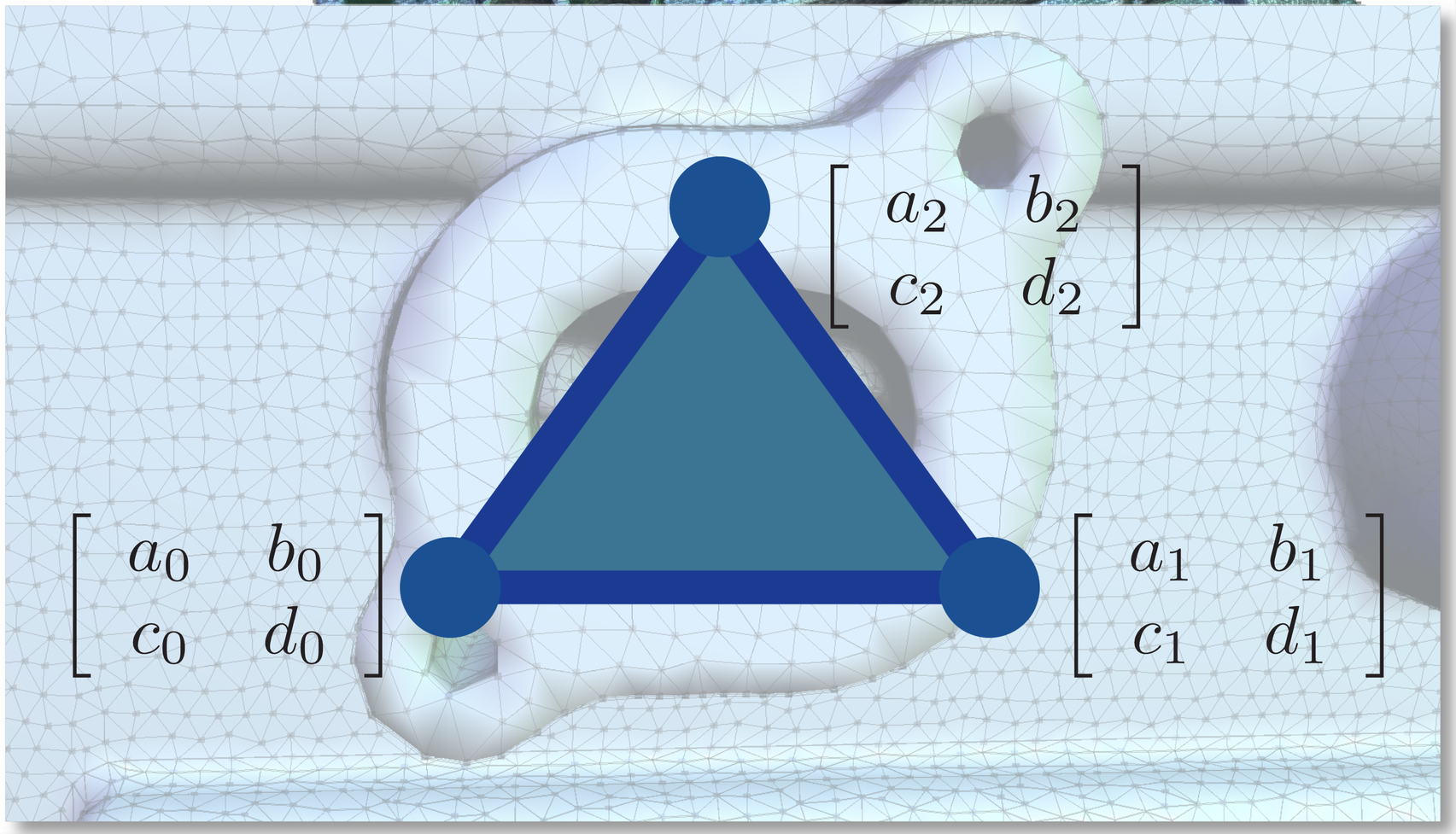
- Given a domain $\mathcal{D}$
- For each vertex v
- One matrix $f(v)$
  - (dxd)-matrix
  - d: dimension of $\mathcal{D}$
- Interpolation on the other simplices
  - Matrix coefficients



In practice



- Given a domain $\mathcal{D}$
- For each vertex v
- One matrix $f(v)$
  - (dxd)-matrix
  - d: dimension of $\mathcal{D}$
- Interpolation on the other simplices
  - Matrix coefficients
  - Eigenvector/values



Tensor diagonalization

- If for each vertex v

Tensor diagonalization

- If for each vertex v
  - If $f(v)$ is a **symmetric** matrix
    - $f(v)_{ij} = f(v)_{ji}, \quad \forall i, j$

Tensor diagonalization

- If for each vertex v
  - If $f(v)$ is a **symmetric** matrix
    - $f(v)_{ij} = f(v)_{ji}, \quad \forall i, j$
  - It can be diagonalized

Tensor diagonalization

- If for each vertex v
  - If $f(v)$ is a **symmetric** matrix
    - $f(v)_{ij} = f(v)_{ji}, \quad \forall i, j$
  - It can be diagonalized

$$U \ f(v) \ U^T = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Tensor diagonalization

- If for each vertex v
  - If $f(v)$ is a **symmetric** matrix
    - $f(v)_{ij} = f(v)_{ji}, \quad \forall i, j$
  - It can be diagonalized
    - $\lambda_1, \lambda_2, \lambda_3$
      - Eigenvalues

$$U \ f(v) \ U^T = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Tensor diagonalization

- If for each vertex v
  - If $f(v)$ is a **symmetric** matrix
    - $f(v)_{ij} = f(v)_{ji}, \quad \forall i, j$
  - It can be diagonalized
    - $\lambda_1, \lambda_2, \lambda_3$
      - Eigenvalues
    - $U = (\vec{e_1}, \vec{e_2}, \vec{e_3})$
      - Eigenvectors

$$U \ f(v) \ U^T = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Tensor field interpolation

Tensor field interpolation

- Let's represent $f(v)$ by an ellipsoid

Tensor field interpolation

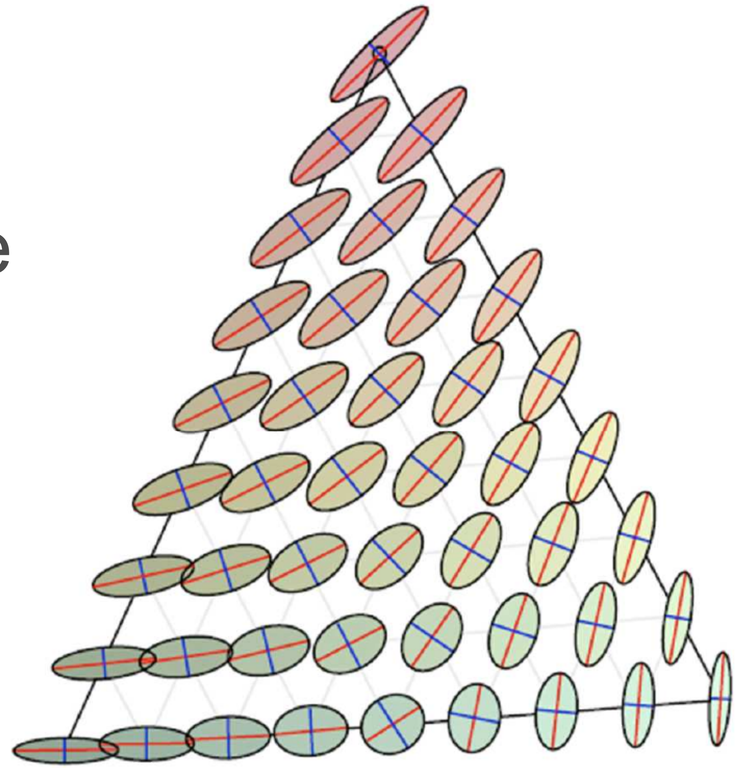
- Let's represent $f(v)$ by an ellipsoid
  - Semi-principal axes
    - Eigenvectors

Tensor field interpolation

- Let's represent $f(v)$ by an ellipsoid
  - Semi-principal axes
    - Eigenvectors
  - Axis length: eigenvalue

Tensor field interpolation

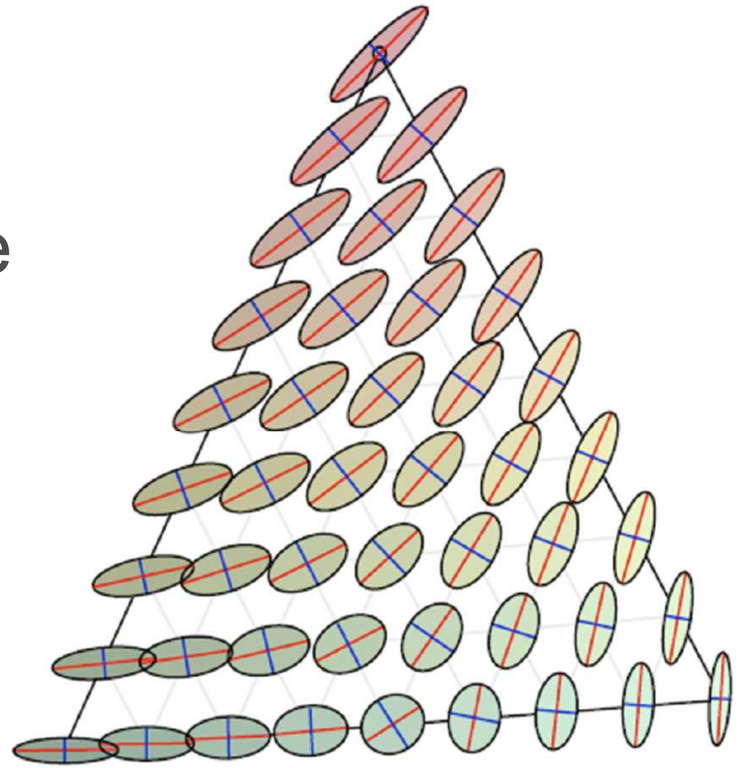
- Let's represent $f(v)$ by an ellipsoid
  - Semi-principal axes
    - Eigenvectors
  - Axis length: eigenvalue



[Hotz]

Tensor field interpolation

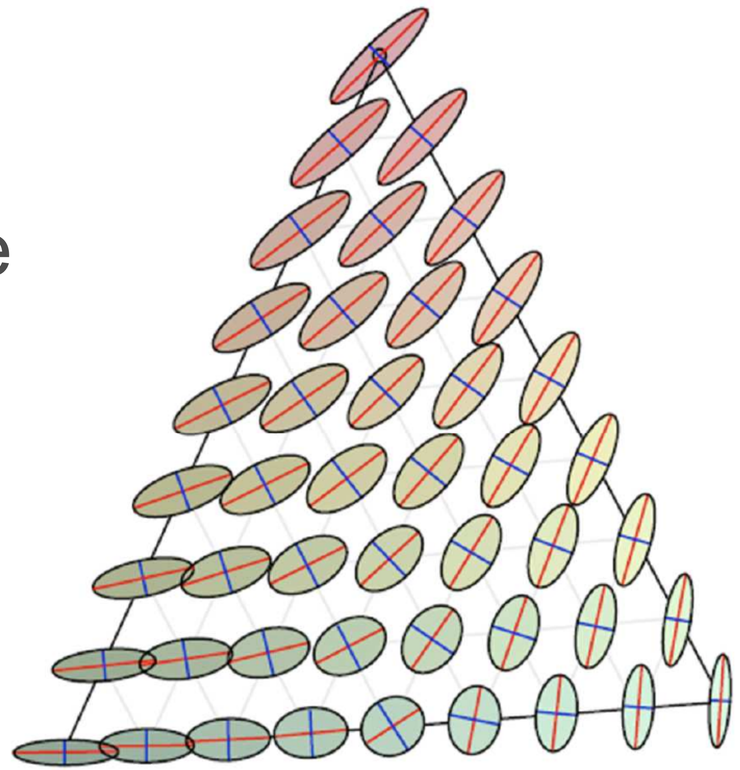
- Let's represent $f(v)$ by an ellipsoid
  - Semi-principal axes
    - Eigenvectors
  - Axis length: eigenvalue
  - Interpolation of the eigenvectors/values

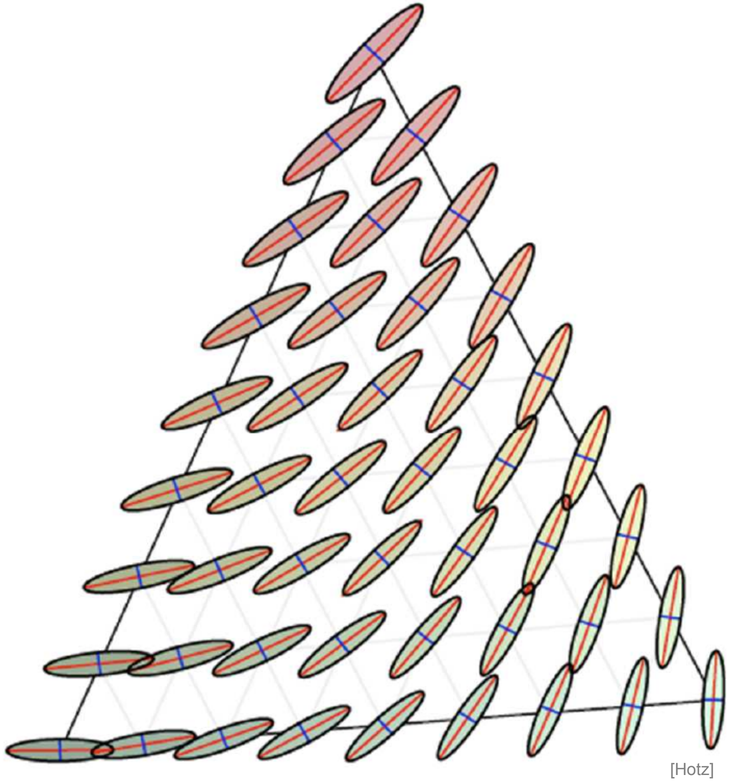


[Hotz]

Tensor field interpolation

- Let's represent $f(v)$ by an ellipsoid
  - Semi-principal axes
    - Eigenvectors
  - Axis length: eigenvalue
  - Interpolation of the eigenvectors/values

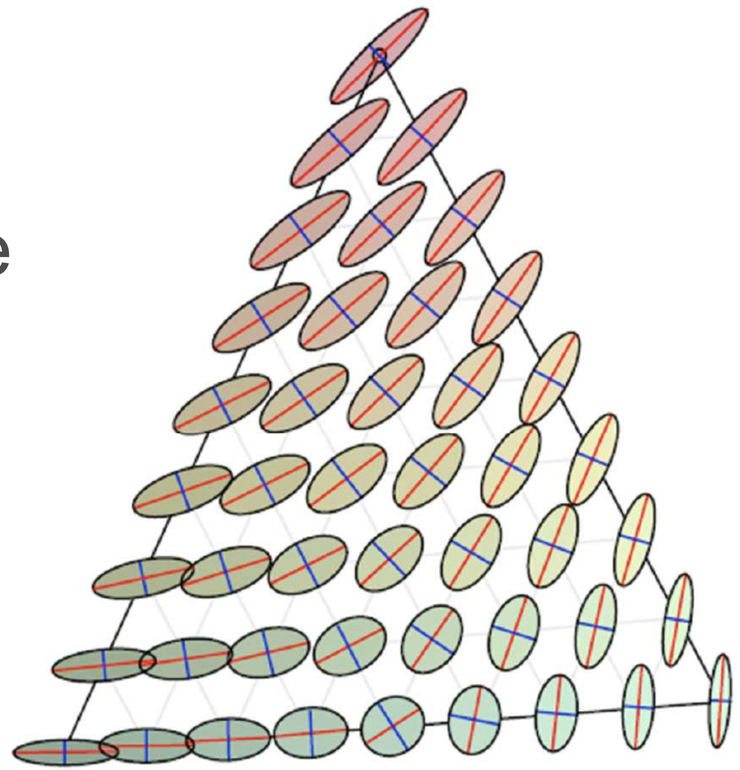


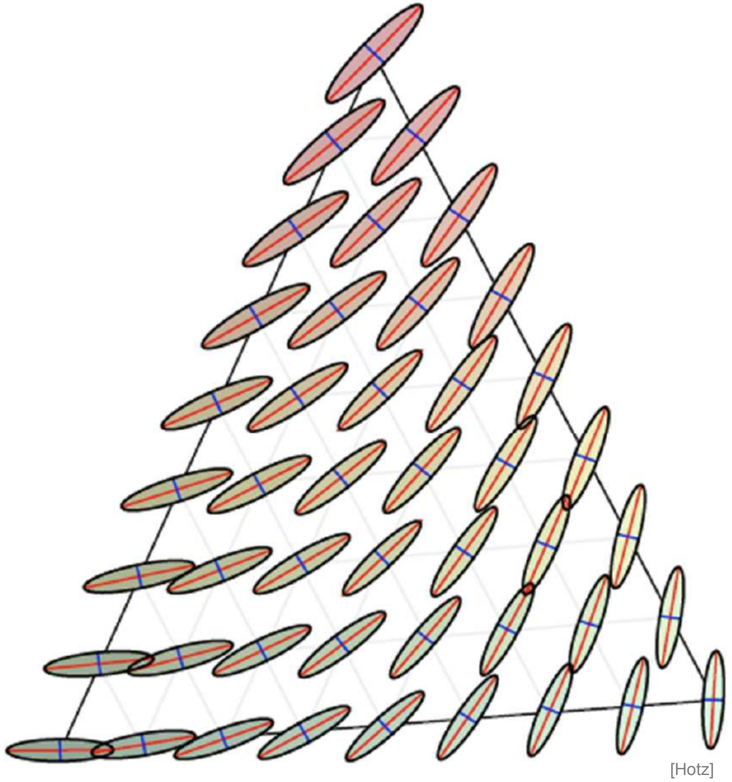


[Hotz]

Tensor field interpolation

- Let's represent $f(v)$ by an ellipsoid
  - Semi-principal axes
    - Eigenvectors
  - Axis length: eigenvalue
  - Interpolation of the eigenvectors/values
  - Similar to vector fields
    - Magnitude/angle





[Hotz]

Glyph packing

- Intuitive idea

Glyph packing

- Intuitive idea
  - Locally represent the tensor with a simple symbol

Glyph packing

- Intuitive idea
  - Locally represent the tensor with a simple symbol
    - Analogy with arrows for vector fields

Glyph packing

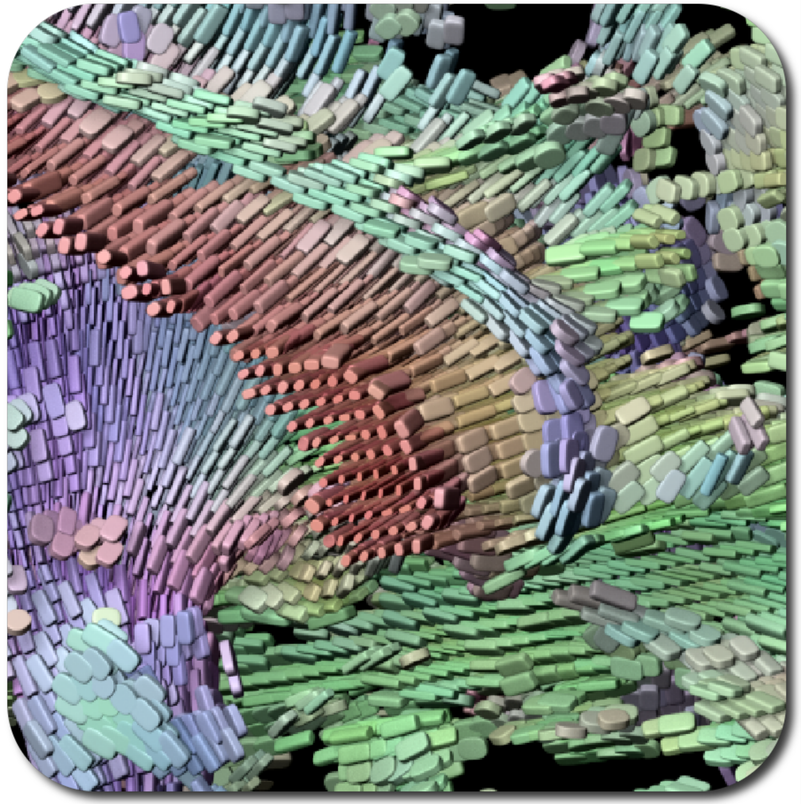
- Intuitive idea
  - Locally represent the tensor with a simple symbol
    - Analogy with arrows for vector fields
  - What kind of symbol?

Glyph packing

- Intuitive idea
  - Locally represent the tensor with a simple symbol
    - Analogy with arrows for vector fields
  - What kind of symbol?
    - Ellipsoids, “superquadrics”

Glyph packing

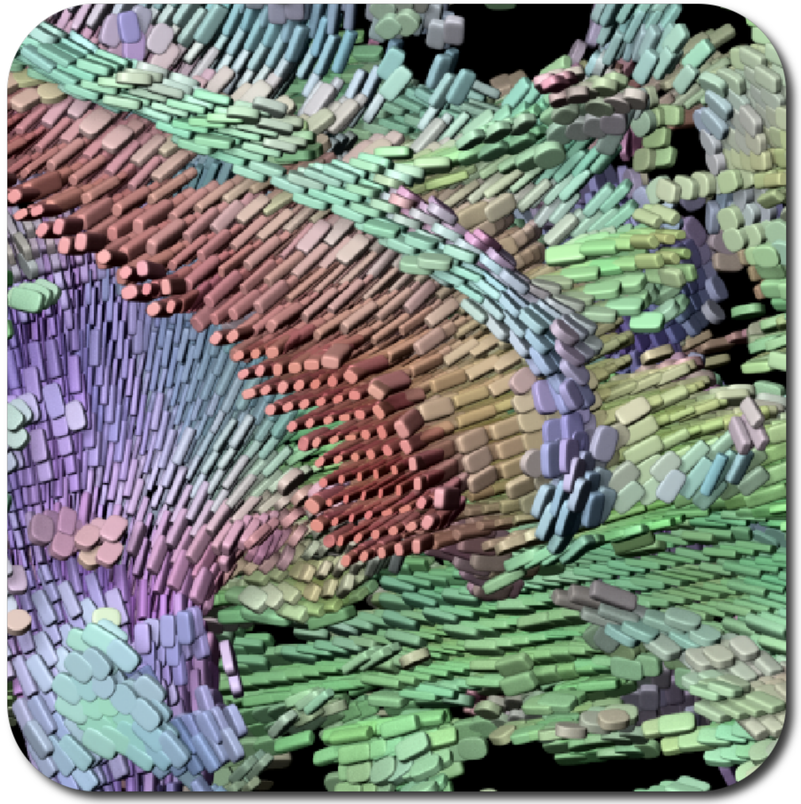
- Intuitive idea
  - Locally represent the tensor with a simple symbol
    - Analogy with arrows for vector fields
  - What kind of symbol?
    - Ellipsoids, “superquadrics”



[Kindlmann]

Glyph packing

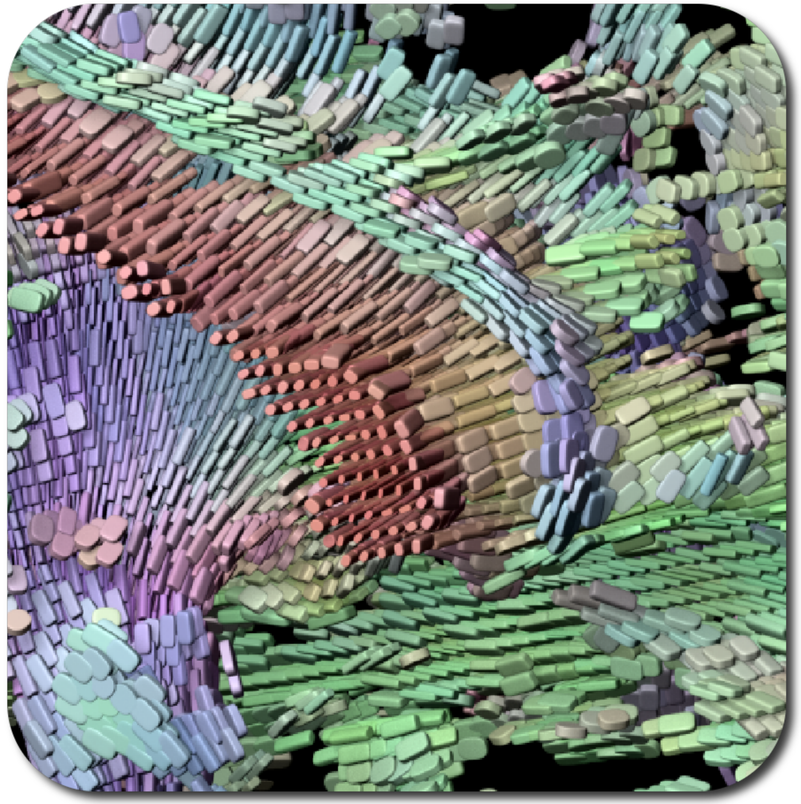
- Intuitive idea
  - Locally represent the tensor with a simple symbol
    - Analogy with arrows for vector fields
  - What kind of symbol?
    - Ellipsoids, “superquadrics”
  - What geometrical properties?



[Kindlmann]

Glyph packing

- Intuitive idea
  - Locally represent the tensor with a simple symbol
    - Analogy with arrows for vector fields
  - What kind of symbol?
    - Ellipsoids, “superquadrics”
  - What geometrical properties?
    - Eigen vectors, eigen values



[Kindlmann]

Tensor diagonalization

- If for each vertex v
  - If $f(v)$ is a **symmetric** matrix
    - $f(v)_{ij} = f(v)_{ji}, \quad \forall i, j$
  - It can be diagonalized
    - $\lambda_1, \lambda_2, \lambda_3$
      - Eigenvalues
    - $U = (\vec{e_1}, \vec{e_2}, \vec{e_3})$
      - Eigenvectors

$$U \ f(v) \ U^T = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Tensor diagonalization

- If for each vertex v
  - If $f(v)$ is a **symmetric** matrix
    - $f(v)_{ij} = f(v)_{ji}, \quad \forall i, j$

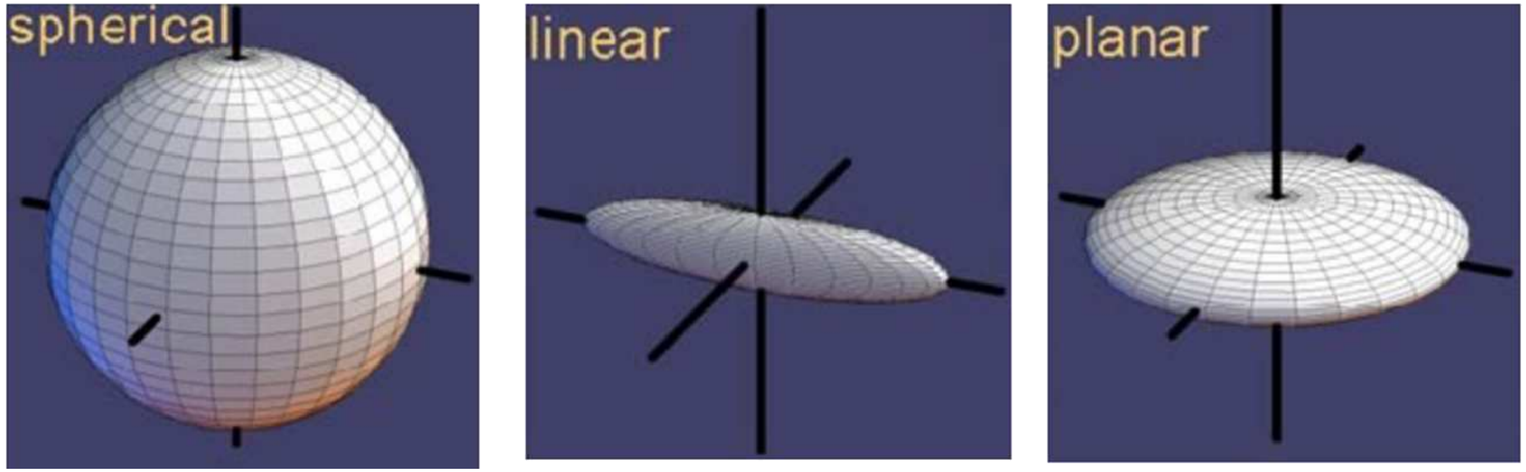
SVD or PCA

  - It can be diagonalized
    - $\lambda_1, \lambda_2, \lambda_3$
      - Eigenvalues
    - $U = (\vec{e_1}, \vec{e_2}, \vec{e_3})$
      - Eigenvectors

$$U \ f(v) \ U^T = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Tensor glyphs

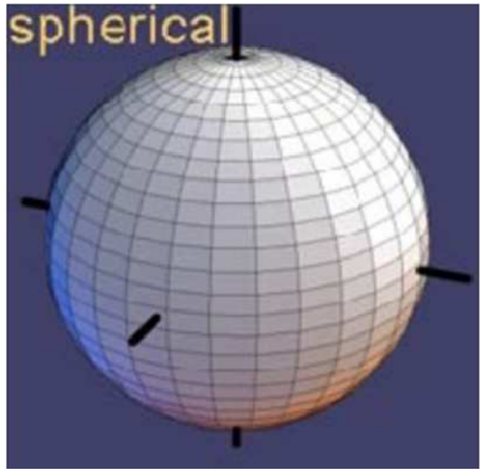
- Ellipsoids

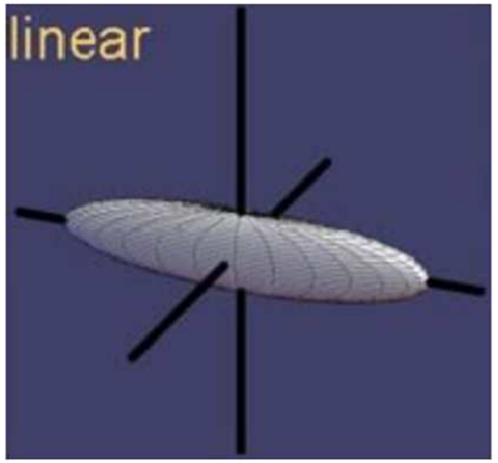


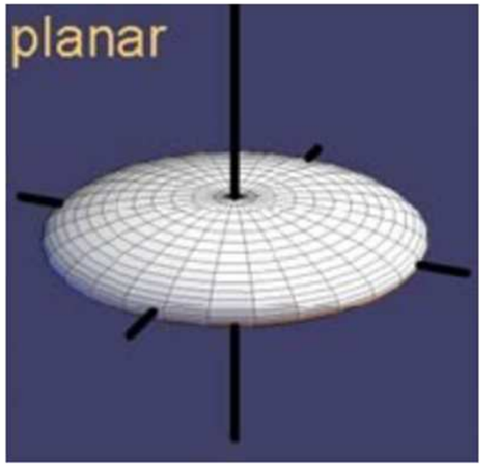
[Kindlmann]

Tensor glyphs

- Ellipsoids
  - Semi-principal axes
    - Eigen vectors



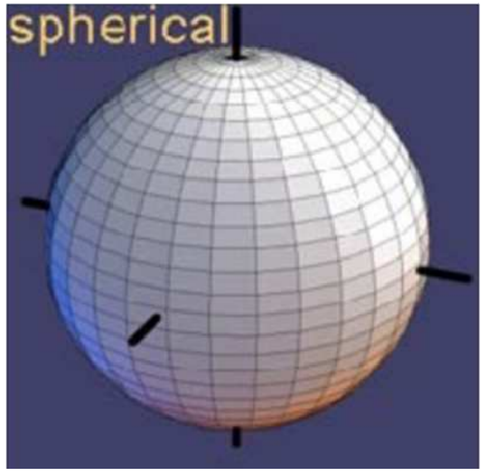


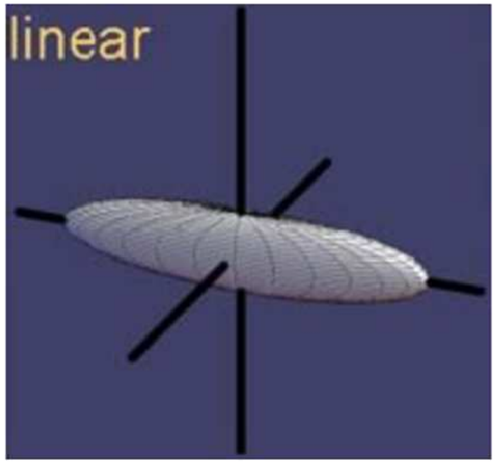


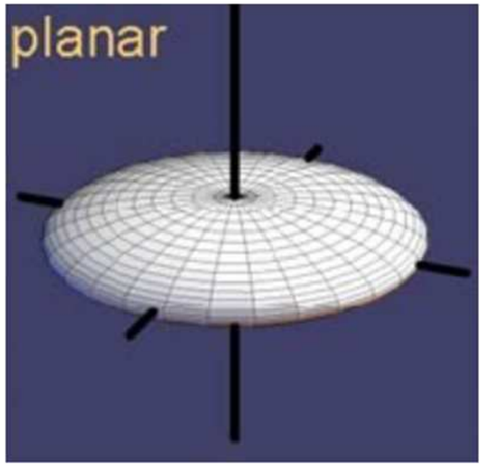
[Kindlmann]

Tensor glyphs

- Ellipsoids
  - Semi-principal axes
    - Eigen vectors
      - **Direction, not a vector!**



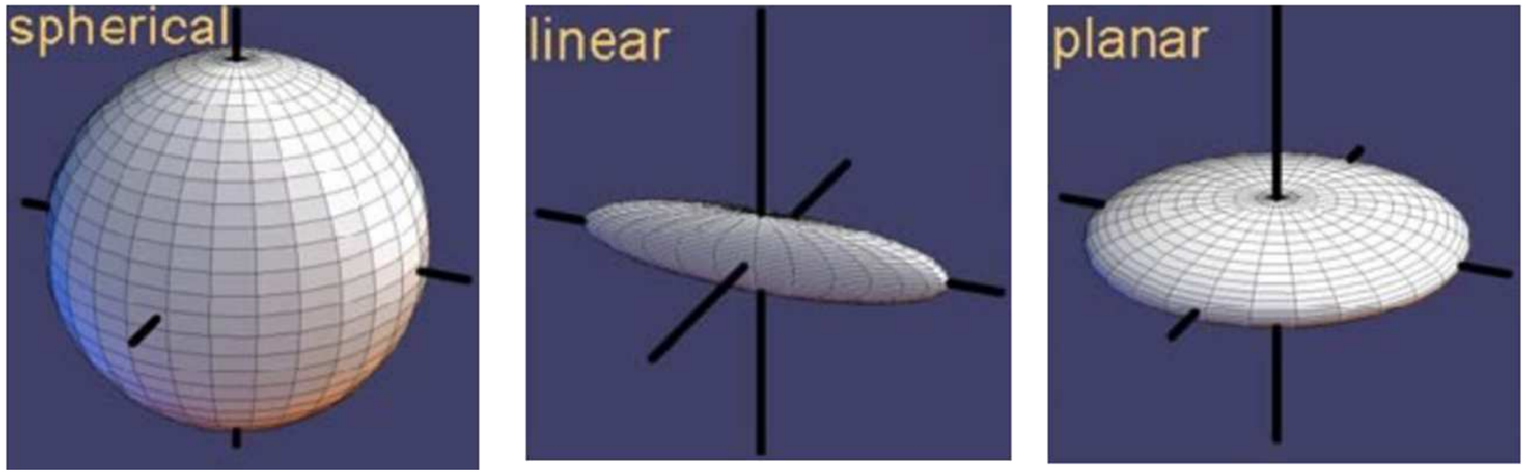




[Kindlmann]

Tensor glyphs

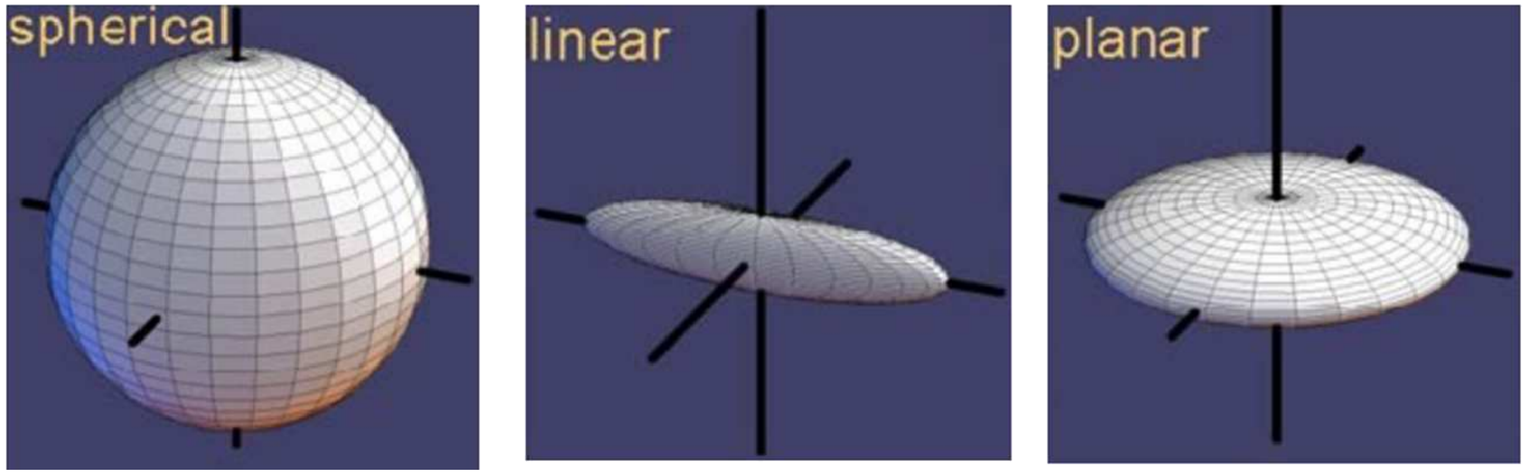
- Ellipsoids
  - Semi-principal axes
    - Eigen vectors
      - **Direction, not a vector!**
  - Axis length
    - Eigen values



[Kindlmann]

Tensor glyphs

- Ellipsoids
  - Semi-principal axes
    - Eigen vectors
      - **Direction, not a vector!**
  - Axis length
    - Eigen values
  - Anisotropy information

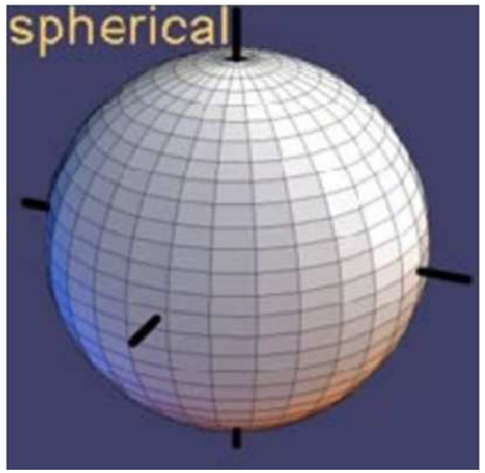


[Kindlmann]

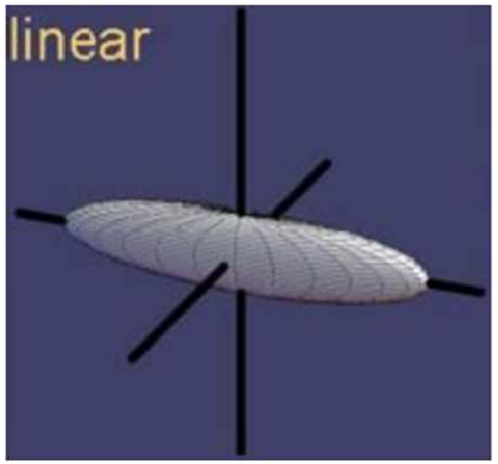
Tensor glyphs

- Ellipsoids
  - Semi-principal axes
    - Eigen vectors
      - **Direction, not a vector!**
  - Axis length
    - Eigen values
  - Anisotropy information

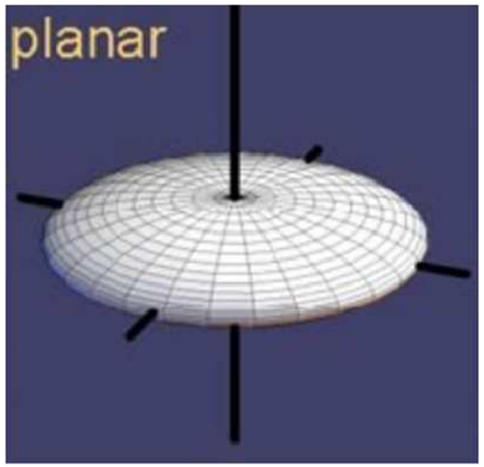
Isotropy



Axial anisotropy



Planar anisotropy

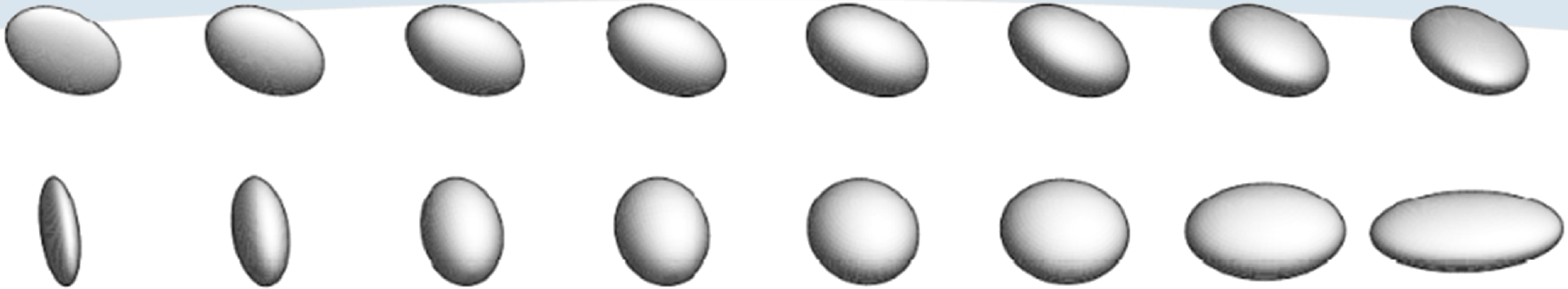


[Kindlmann]

Tensor glyphs

- Ellipsoids in 3D

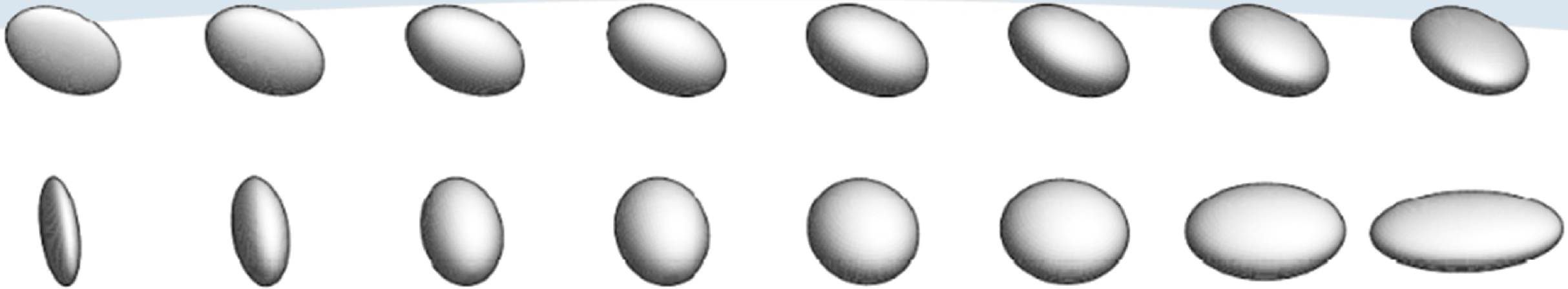
Tensor glyphs



[Kindlmann]

- Ellipsoids in 3D

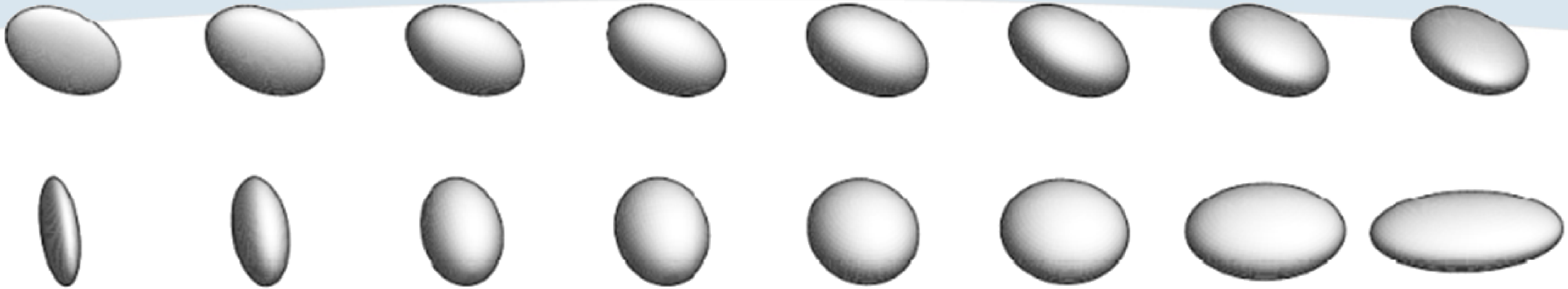
Tensor glyphs



[Kindlmann]

- Ellipsoids in 3D
  - Information loss due to screen projection

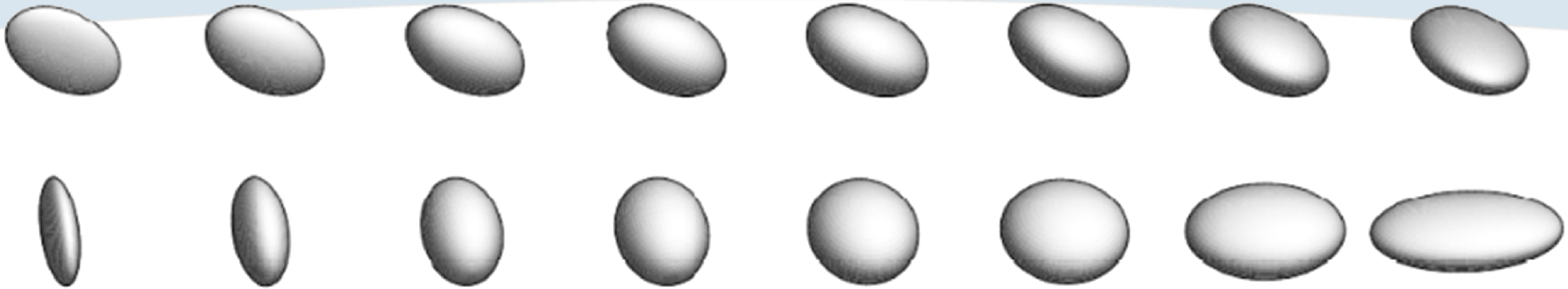
Tensor glyphs



[Kindlmann]

- Ellipsoids in 3D
  - Information loss due to screen projection
  - Continuous shading

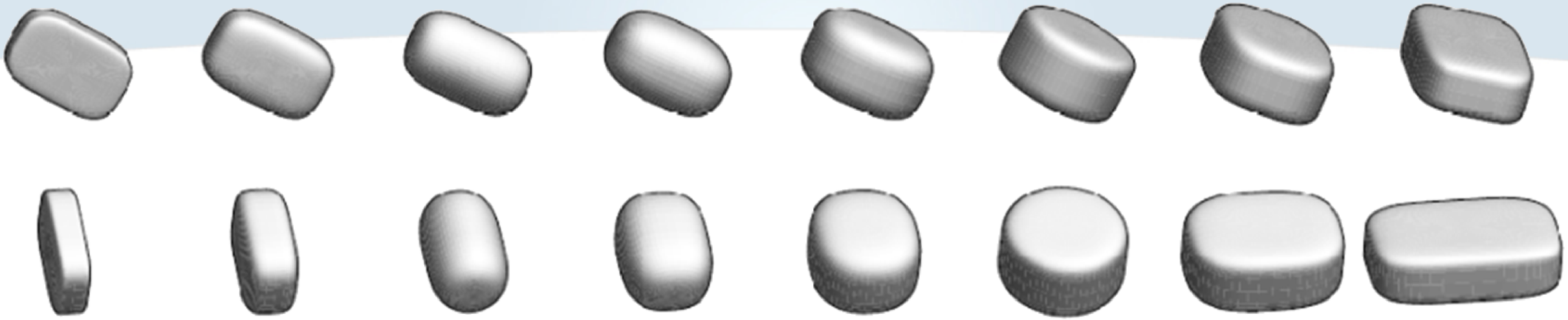
Tensor glyphs



[Kindlmann]

- Ellipsoids in 3D
  - Information loss due to screen projection
  - Continuous shading
    - Hard to evaluate variations across axes

Tensor glyphs

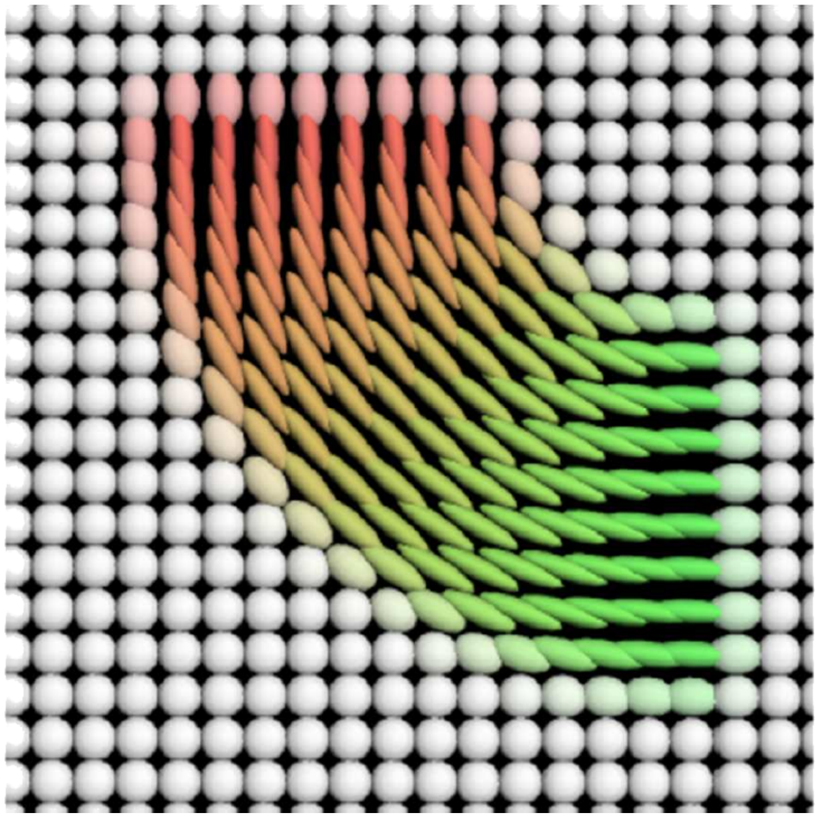


[Kindlmann]

- “Superquadrics”
  - Parallelepiped with smooth edges
  - Exaggerate shading variations across axis

Glyph packing

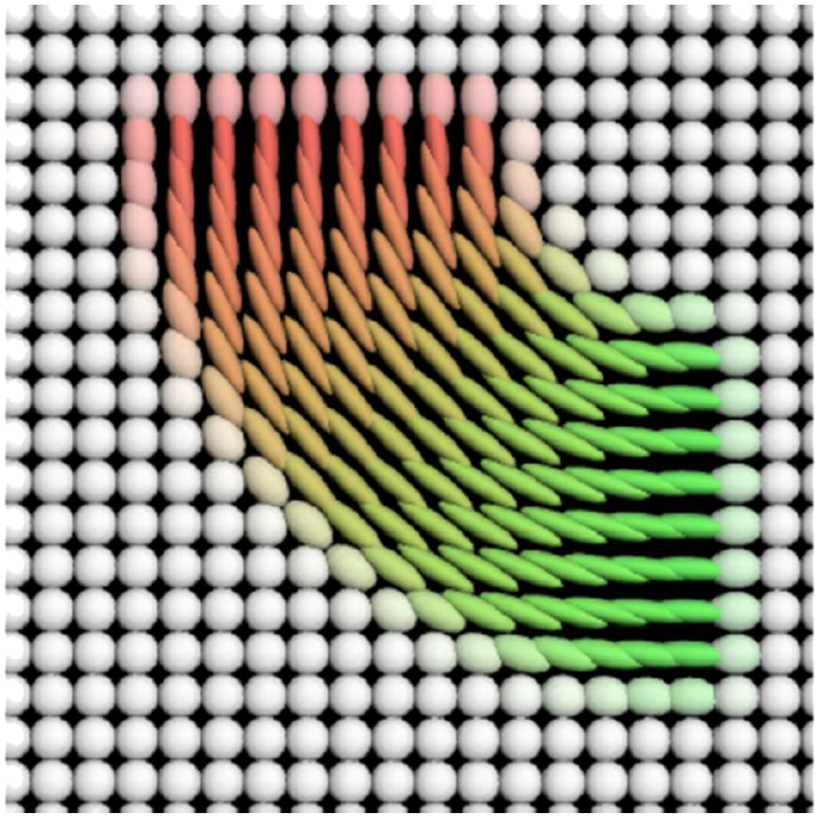
- How to distribute the glyphs



[Kindlmann]

Glyph packing

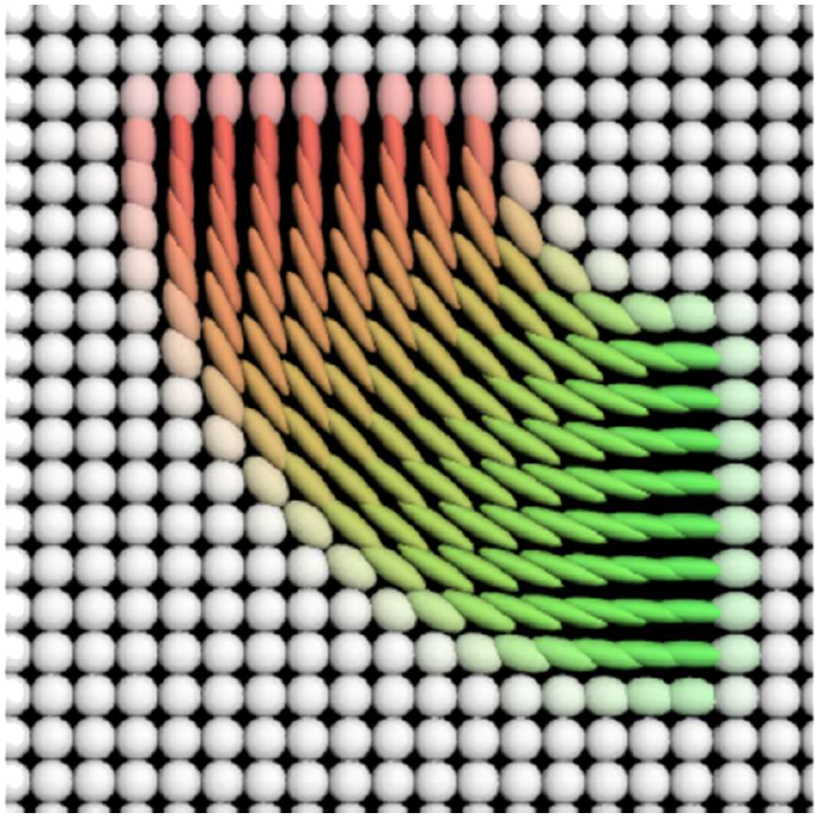
- How to distribute the glyphs
  - To avoid data overload
  - Glyph overlap



[Kindlmann]

Glyph packing

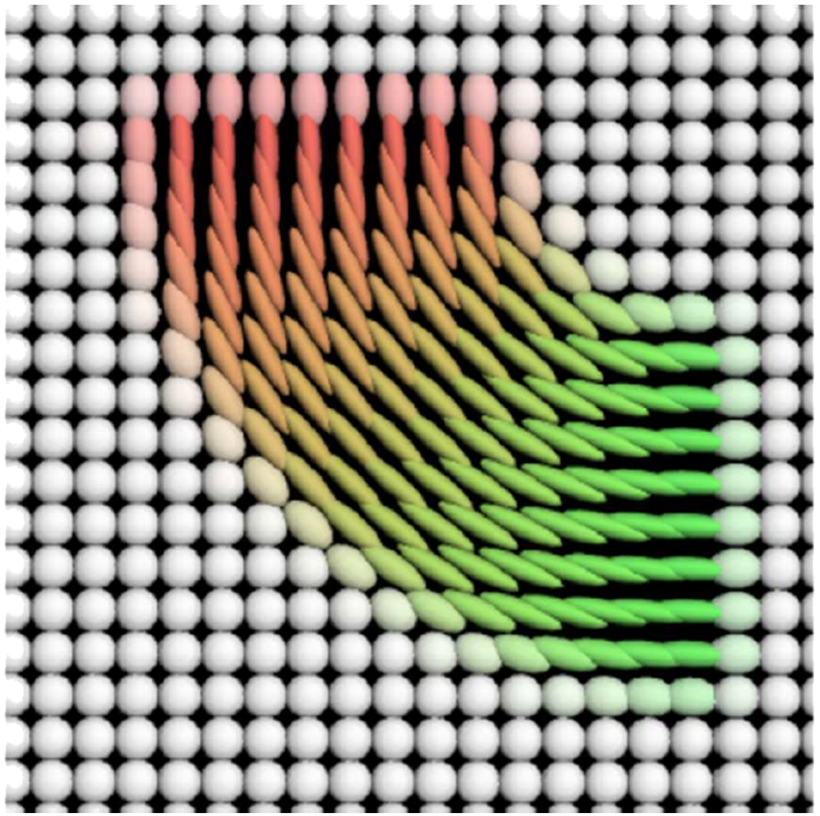
- How to distribute the glyphs
  - To avoid data overload
  - Glyph overlap
- Particle-based energy optimization



[KindImann]

Glyph packing

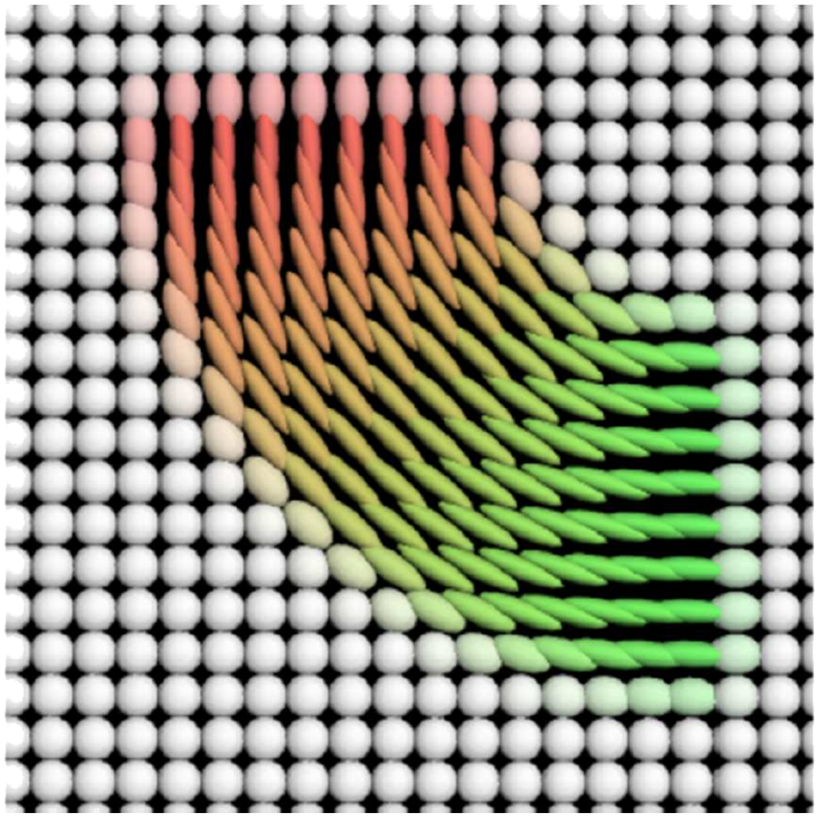
- How to distribute the glyphs
  - To avoid data overload
  - Glyph overlap
- Particle-based energy optimization
  - Given a target number of particles



[Kindlmann]

Glyph packing

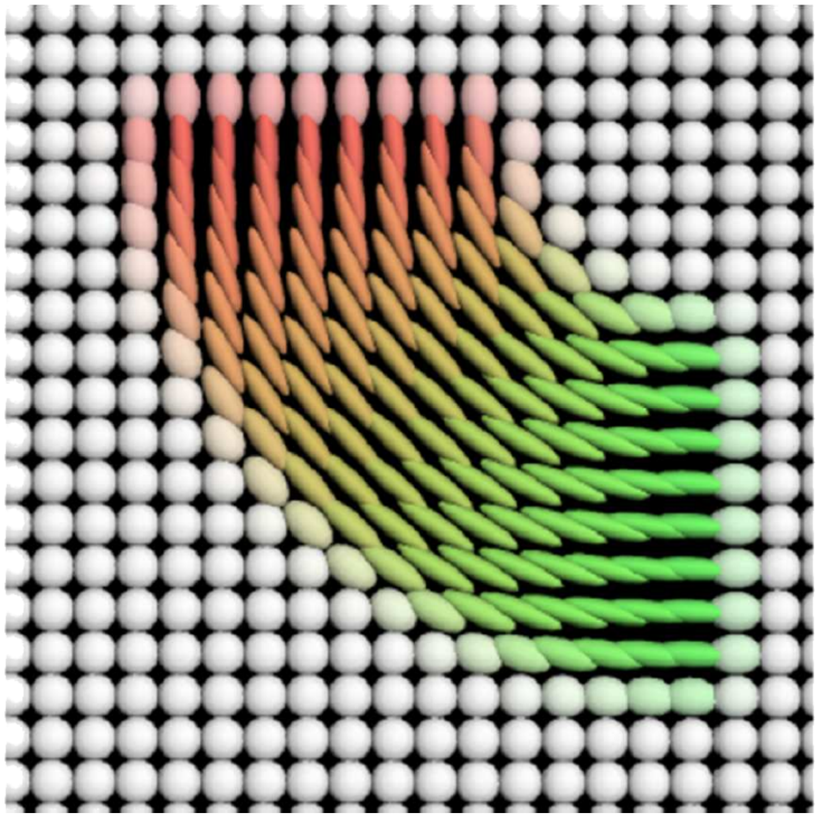
- How to distribute the glyphs
  - To avoid data overload
  - Glyph overlap
- Particle-based energy optimization
  - Given a target number of particles
  - Optimally (globally) scale and place them



[Kindlmann]

Glyph packing

- How to distribute the glyphs
  - To avoid data overload
  - Glyph overlap
- Particle-based energy optimization
  - Given a target number of particles
  - Optimally (globally) scale and place them
    - To minimize overlap

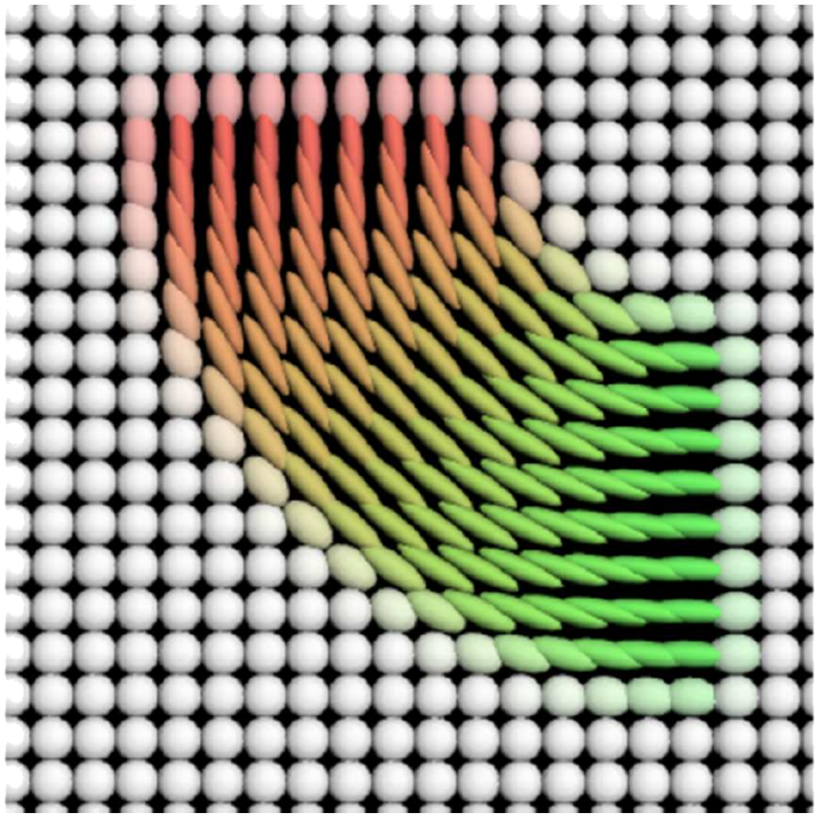


[KindImann]

Glyph packing

- How to distribute the glyphs
  - To avoid data overload
  - Glyph overlap

- Particle-based energy optimization
  - Given a target number of particles
  - Optimally (globally) scale and place them
    - To minimize overlap
  - Maximize the sum of distances between glyphs

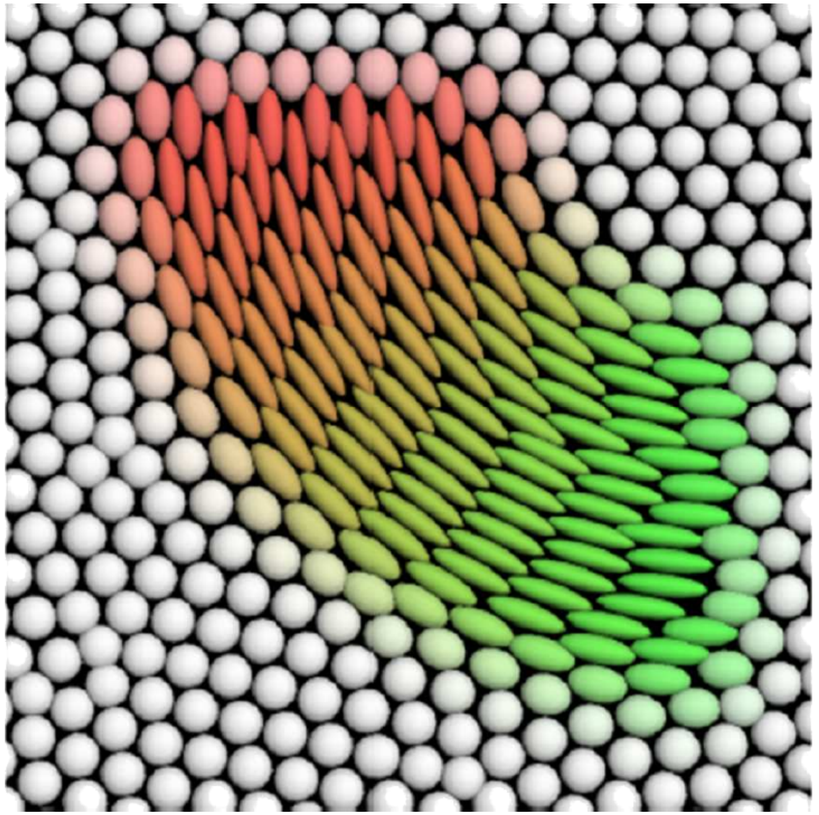


[Kindlmann]

Glyph packing

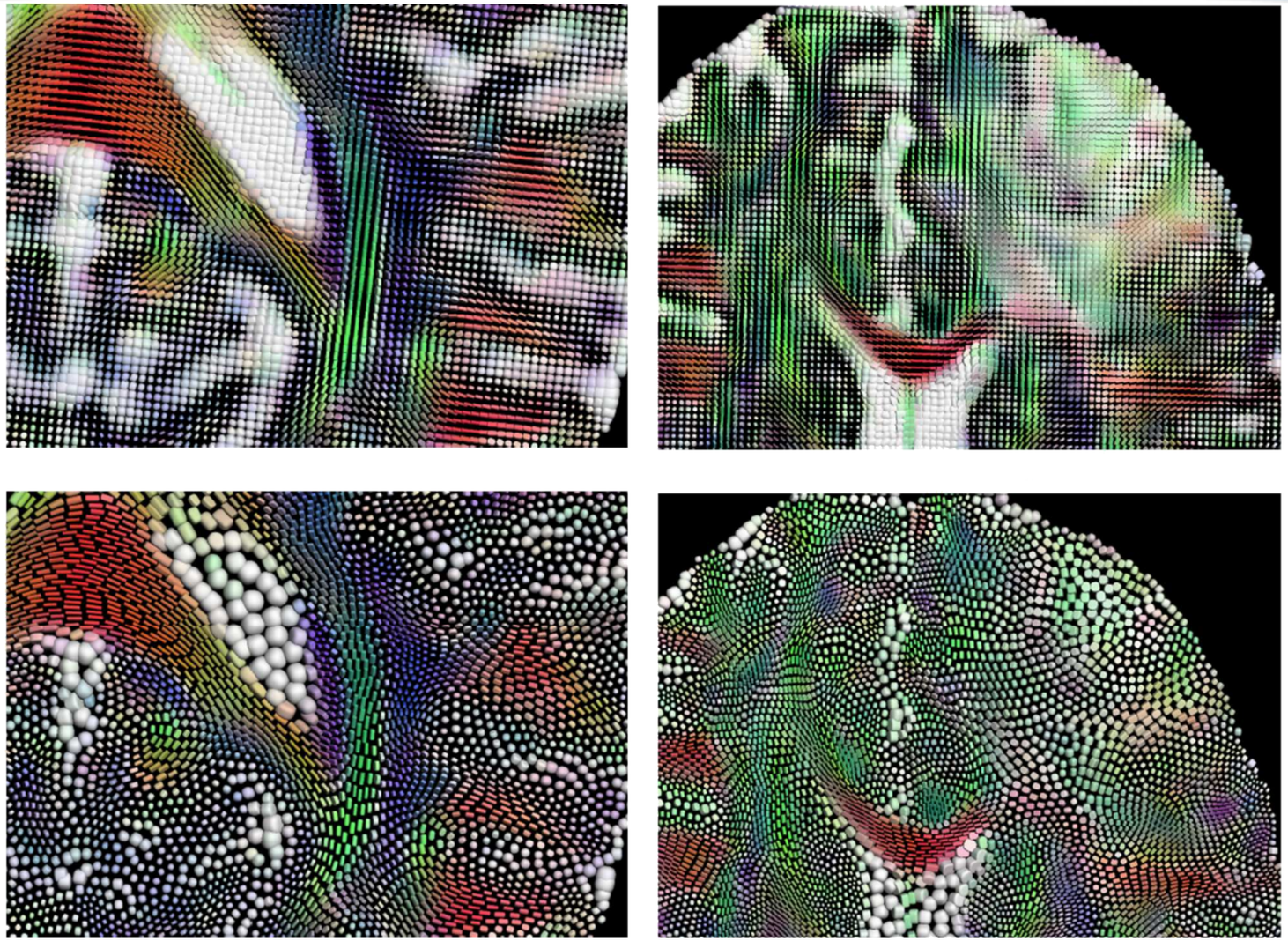
- How to distribute the glyphs
  - To avoid data overload
  - Glyph overlap

- Particle-based energy optimization
  - Given a target number of particles
  - Optimally (globally) scale and place them
    - To minimize overlap
  - Maximize the sum of distances between glyphs



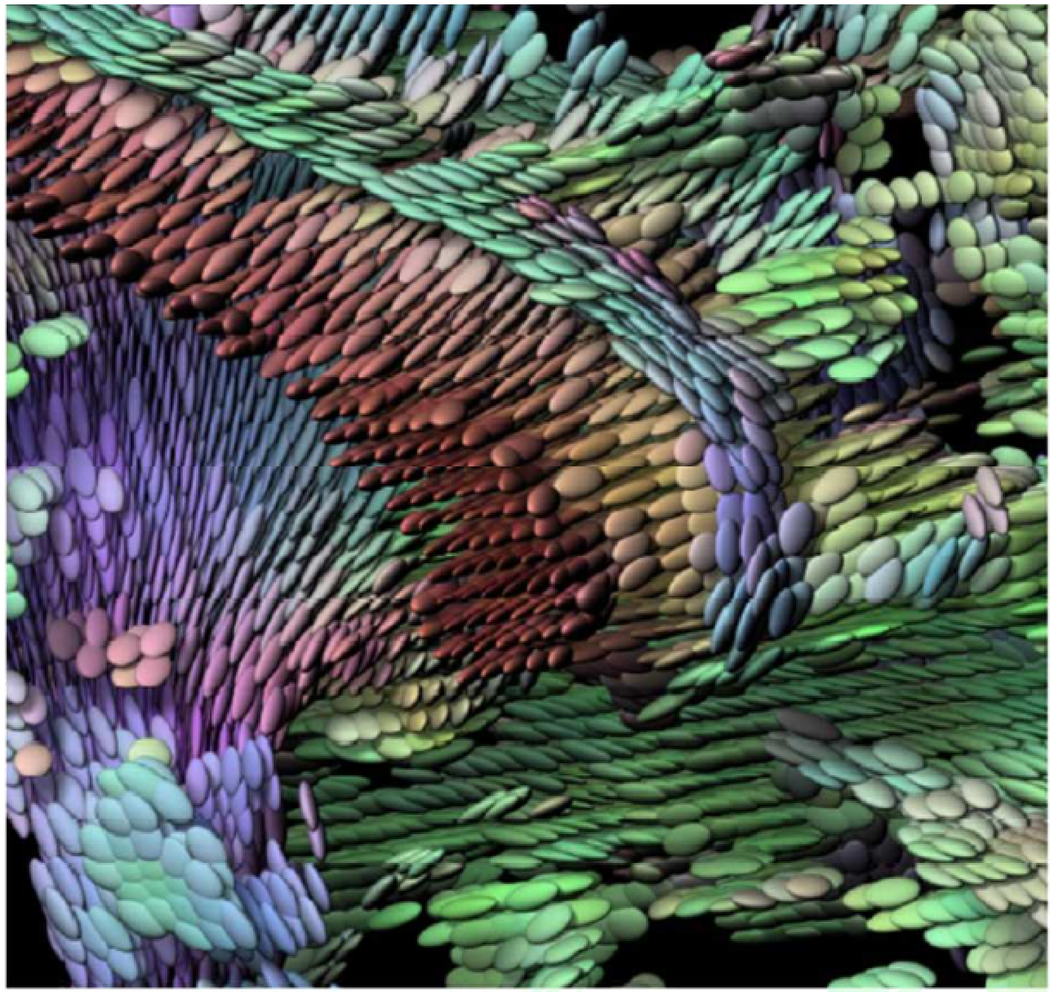
[KindImann]

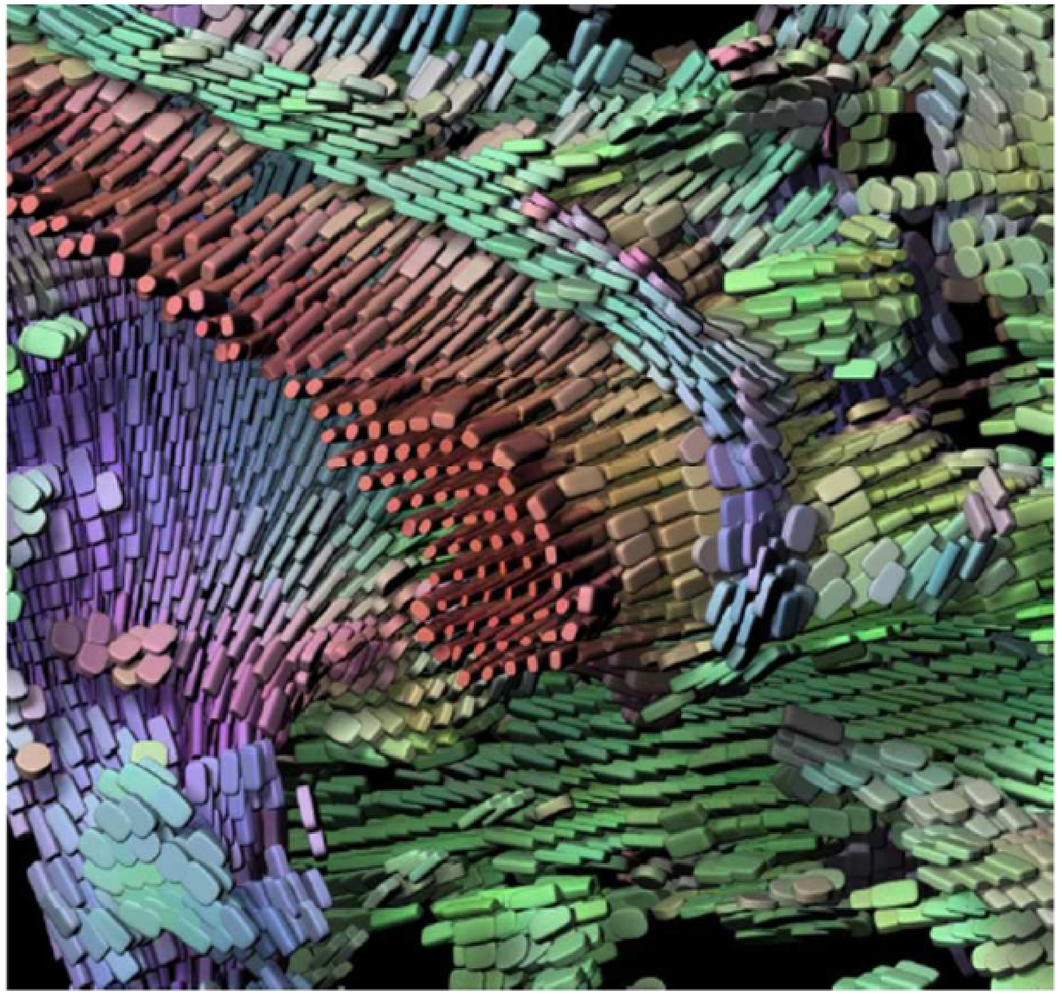
Glyph packing



[Kindlmann]

Putting it all together

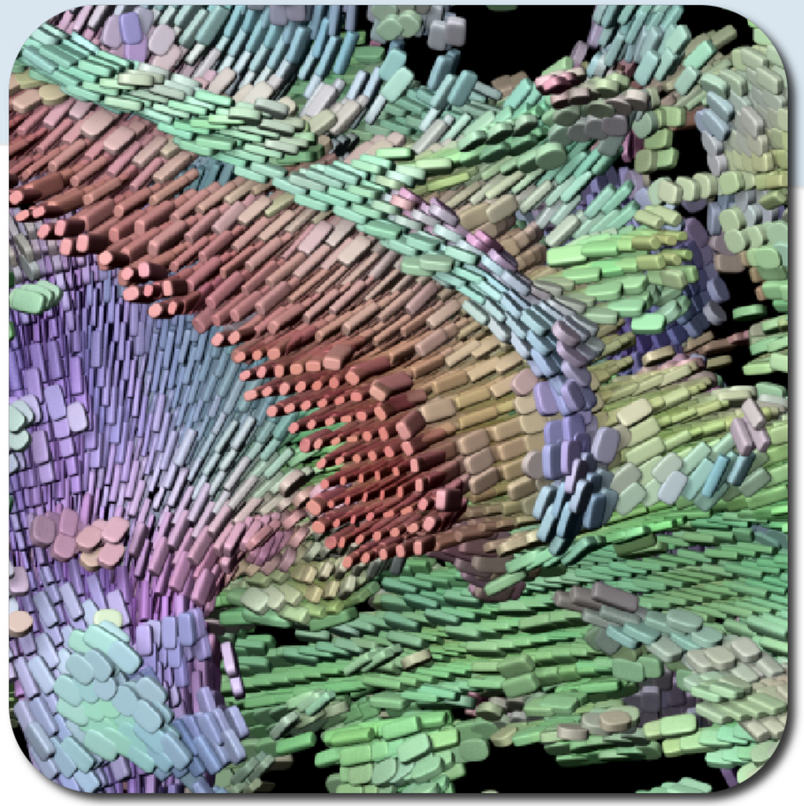




[Kindlmann]

Tensor glyphs

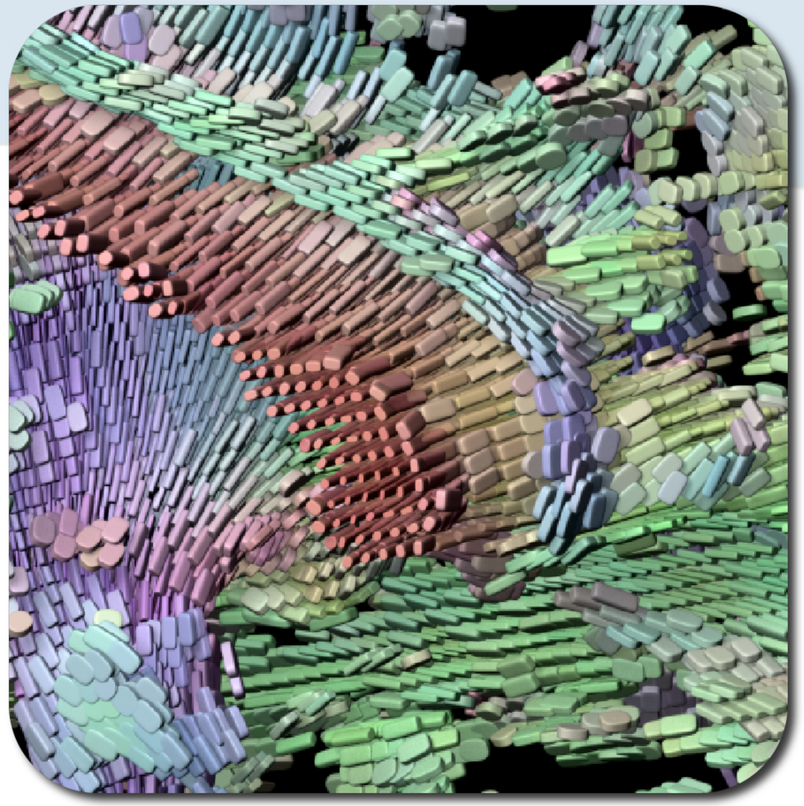
- Provide important insights



[Kindlmann]

Tensor glyphs

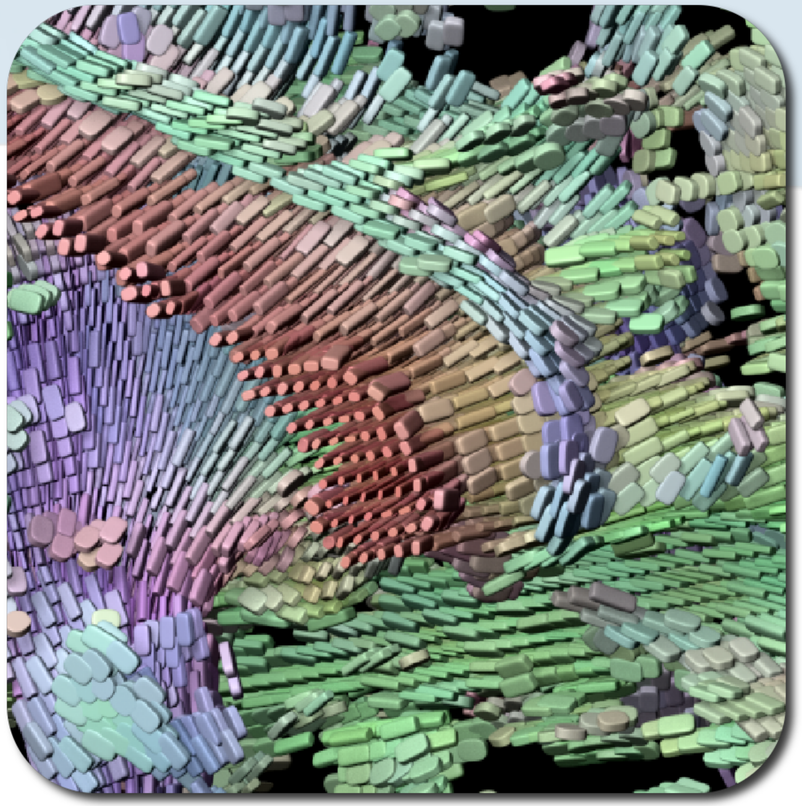
- Provide important insights
  - Direction information



[Kindlmann]

Tensor glyphs

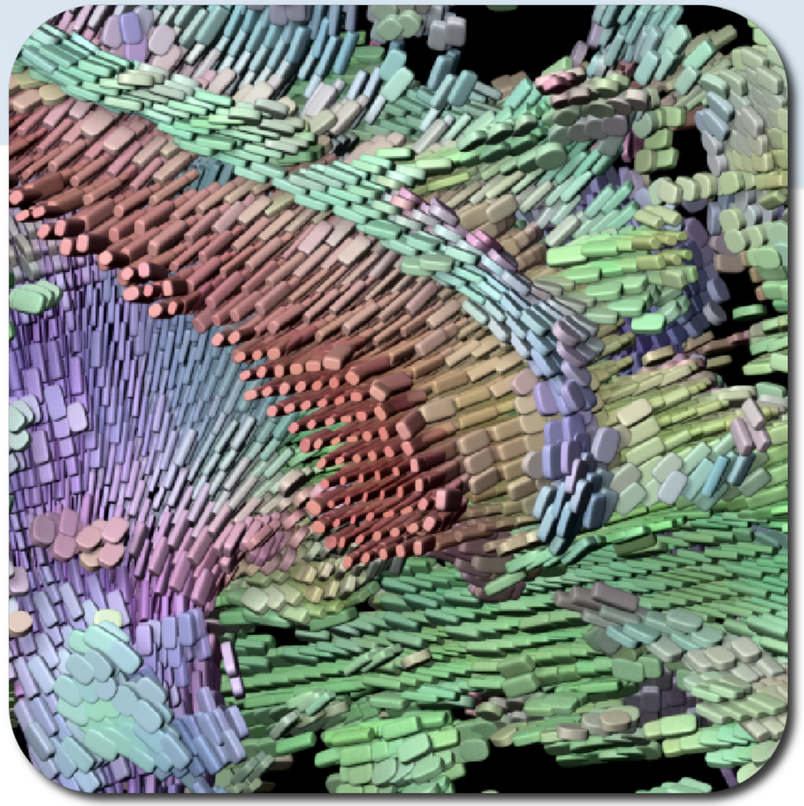
- Provide important insights
  - Direction information
  - Anisotropy information



[Kindlmann]

Tensor glyphs

- Provide important insights
  - Direction information
  - Anisotropy information
- Still
  - Occlusion issues

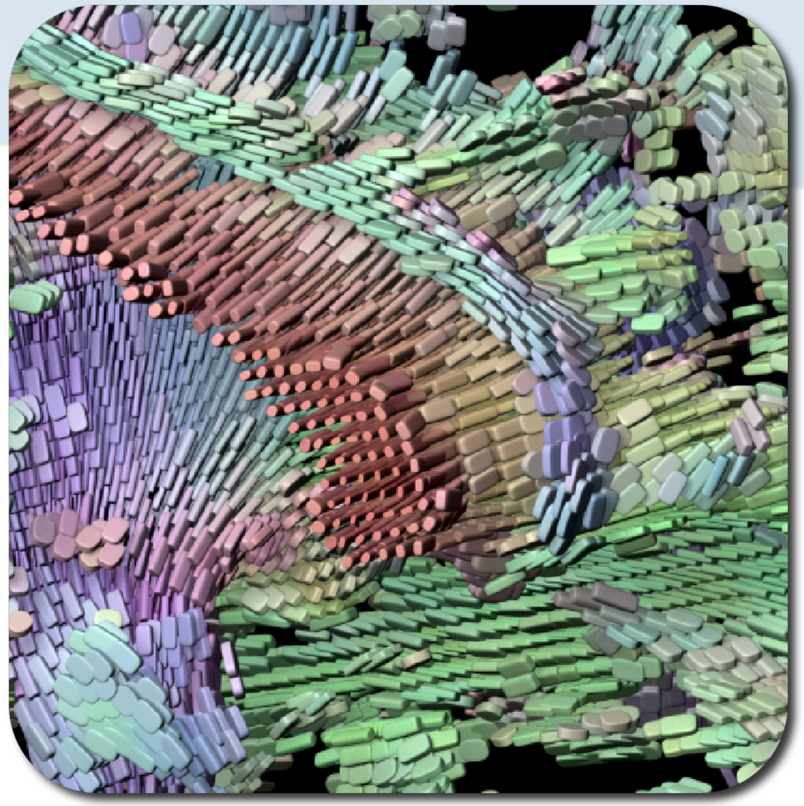


[Kindlmann]

Tensor glyphs

- Provide important insights
  - Direction information
  - Anisotropy information

- Still
  - Occlusion issues
  - No global information

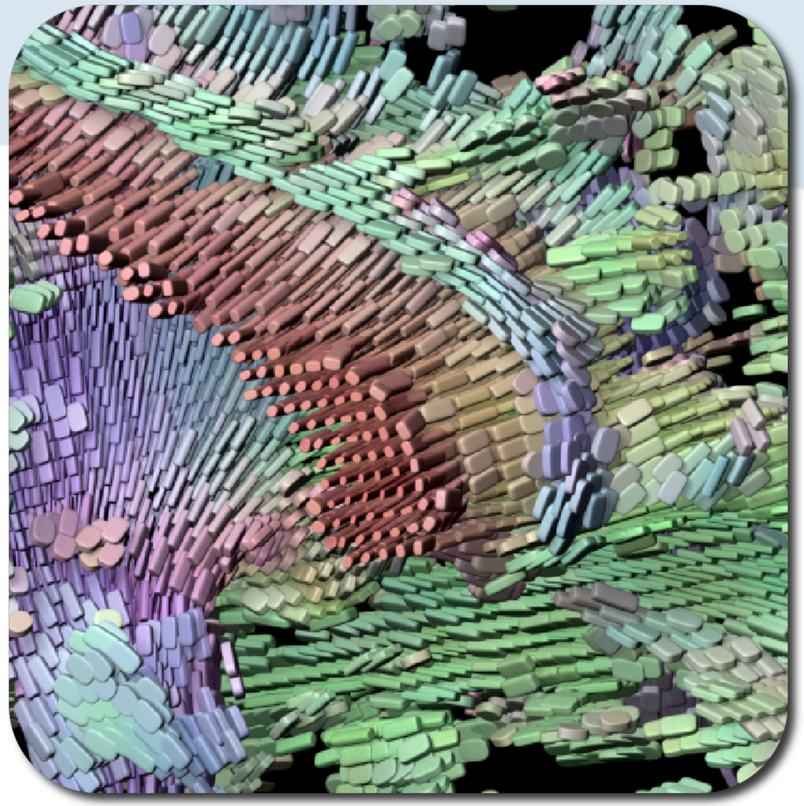


[Kindlmann]

Tensor glyphs

- Provide important insights
  - Direction information
  - Anisotropy information

- Still
  - Occlusion issues
  - No global information
  - Dependent on the number of glyphs
    - Trade-off



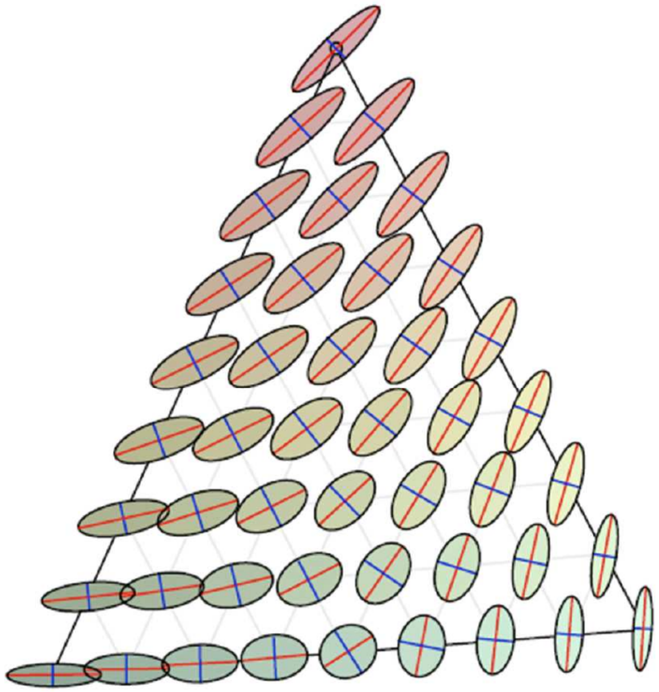
[Kindlmann]

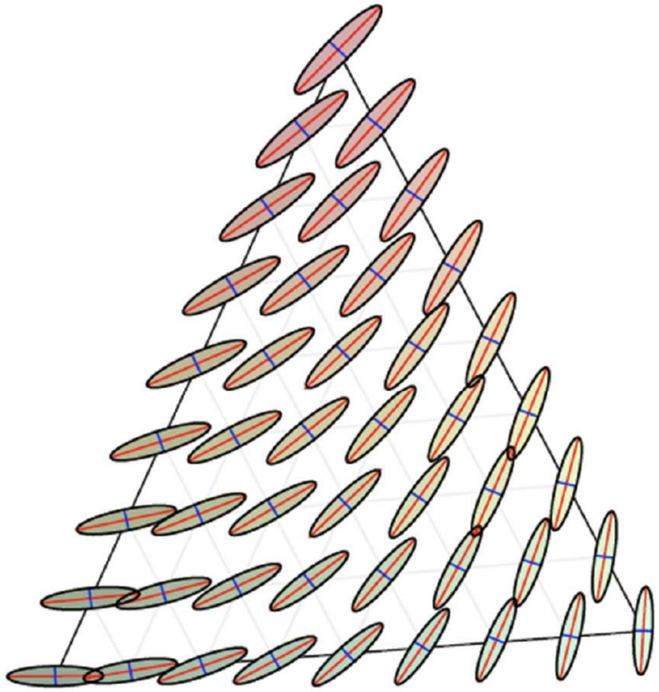
Derived vector fields

- What interesting vector fields could we consider?

Derived vector fields

- What interesting vector fields could we consider?
  - Eigenvectors

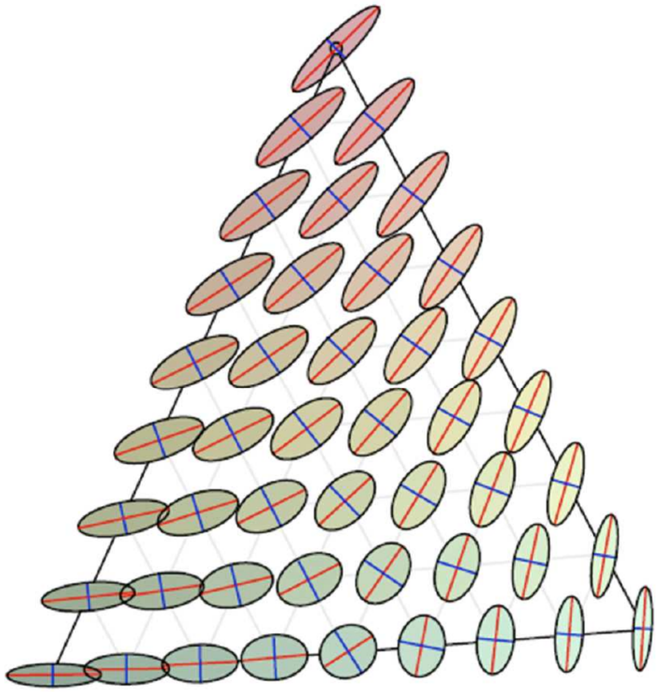


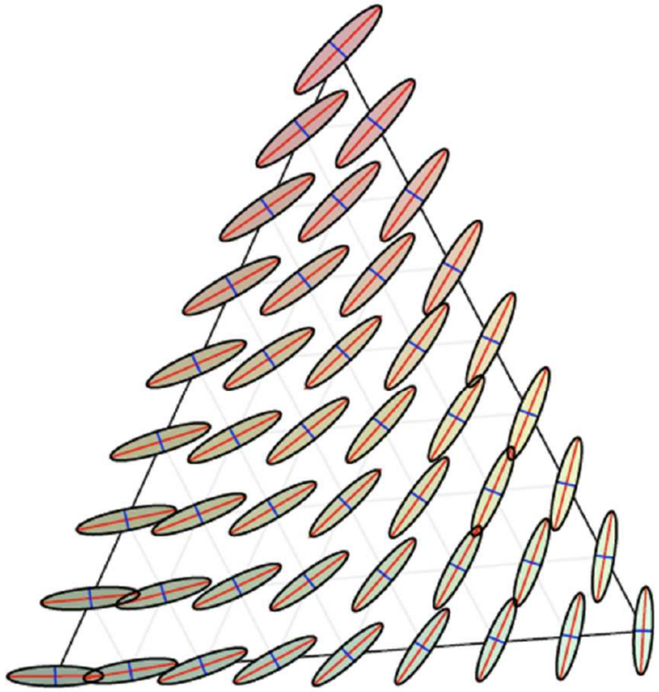


[Hotz]

Derived vector fields

- What interesting vector fields could we consider?
  - Eigenvectors
    - Not a real vector field

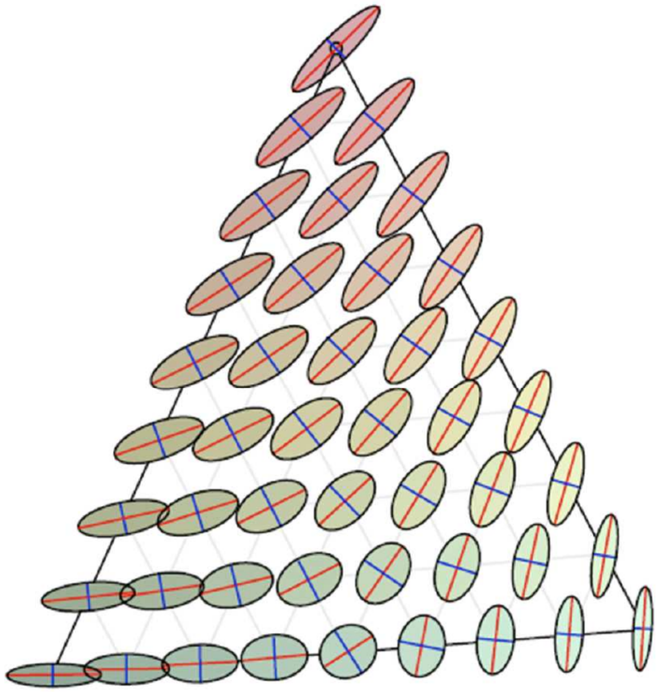


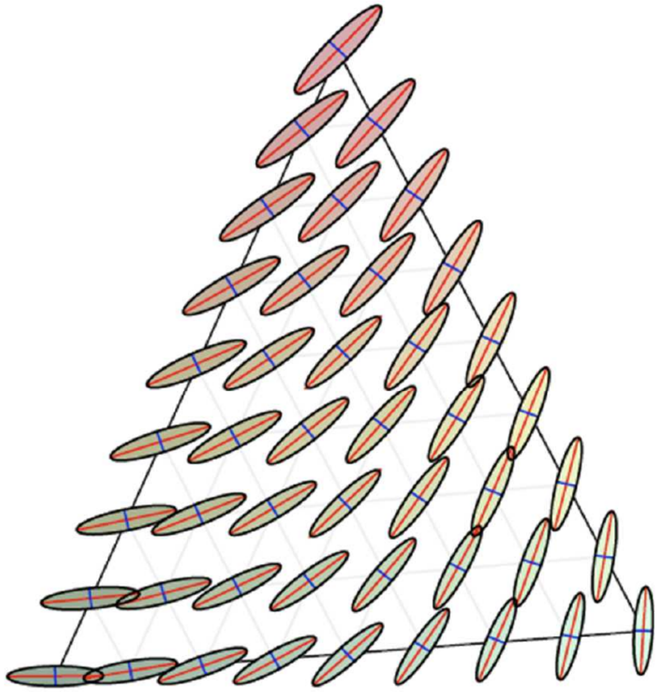


[Hotz]

Derived vector fields

- What interesting vector fields could we consider?
  - Eigenvectors
    - Not a real vector field
      - No magnitude
      - No orientation

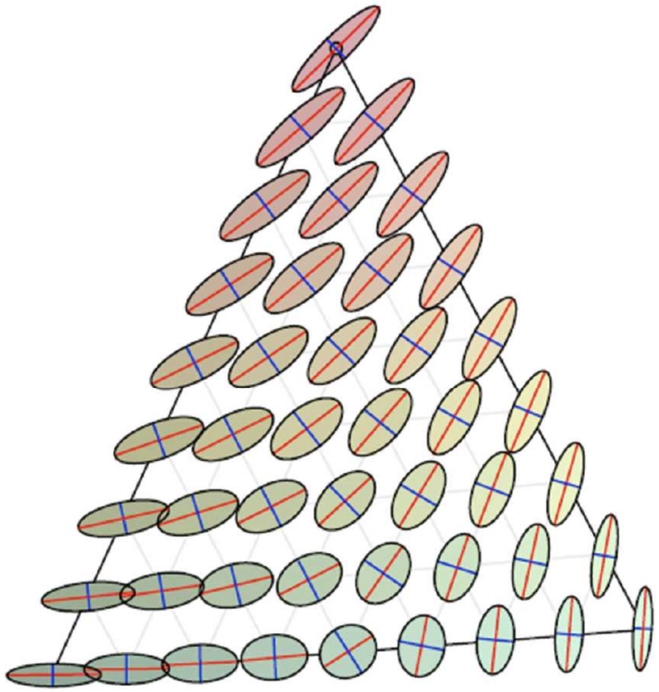


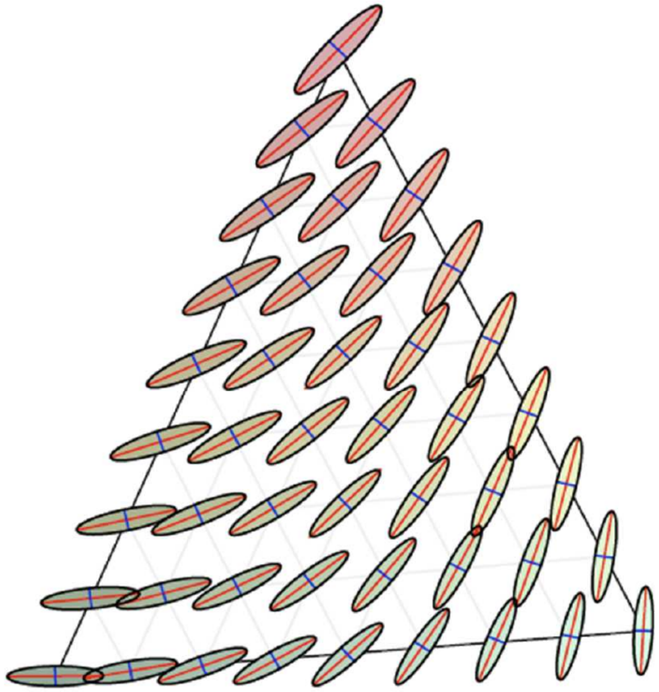


[Hotz]

Derived vector fields

- What interesting vector fields could we consider?
  - Eigenvectors
    - Not a real vector field
      - No magnitude
      - No orientation
  - Magnitude
    - Eigenvalues

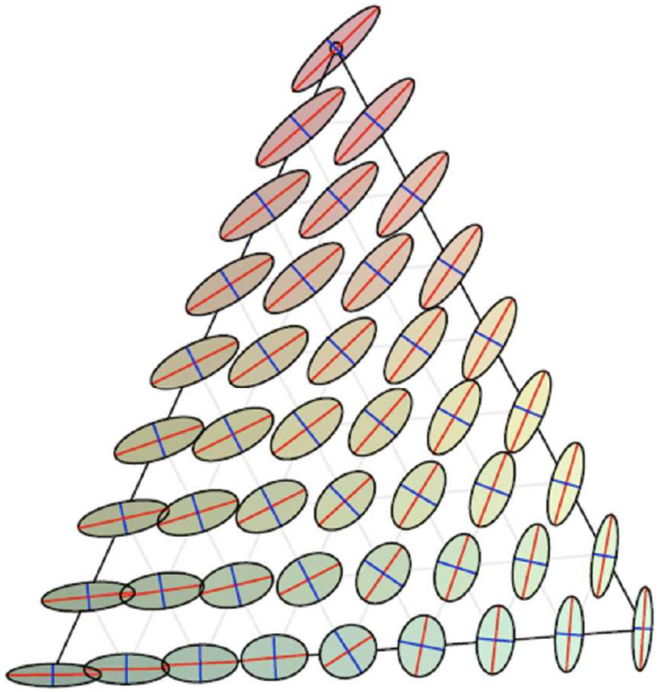


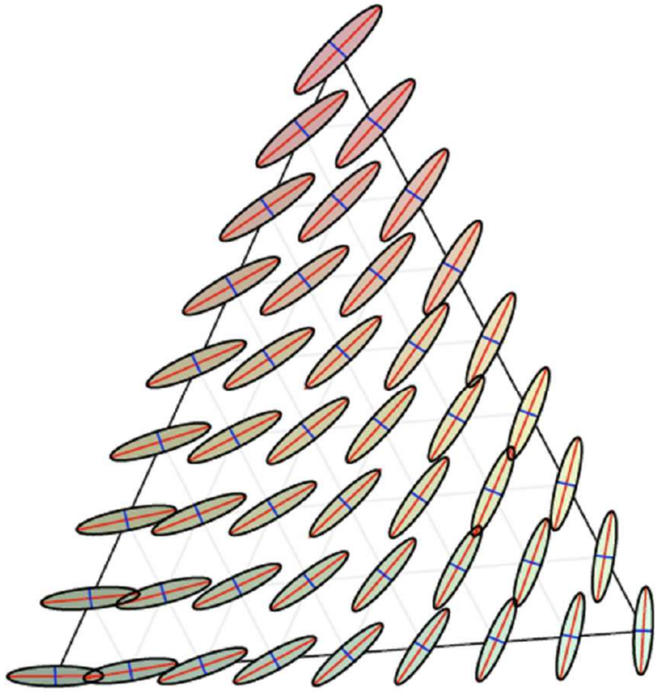


[Hotz]

Derived vector fields

- What interesting vector fields could we consider?
  - Eigenvectors
    - Not a real vector field
      - No magnitude
      - No orientation
    - Magnitude
      - Eigenvalues
  - Ambiguous entity

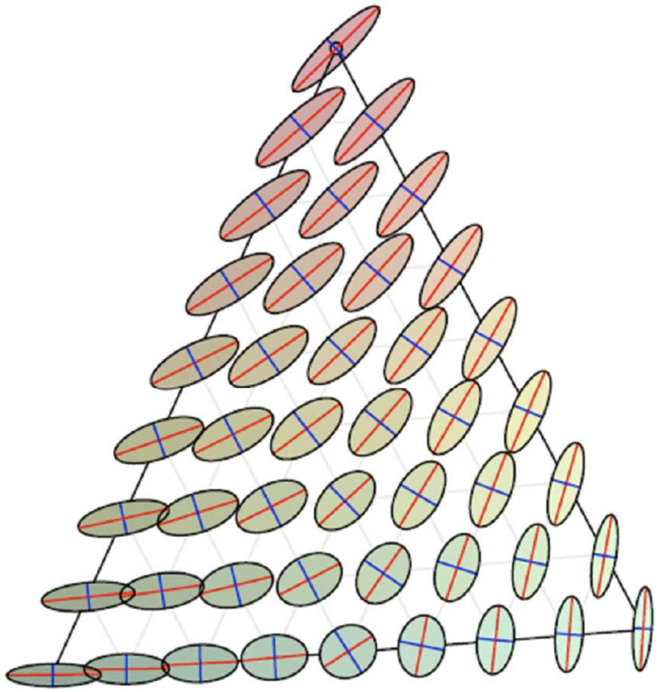


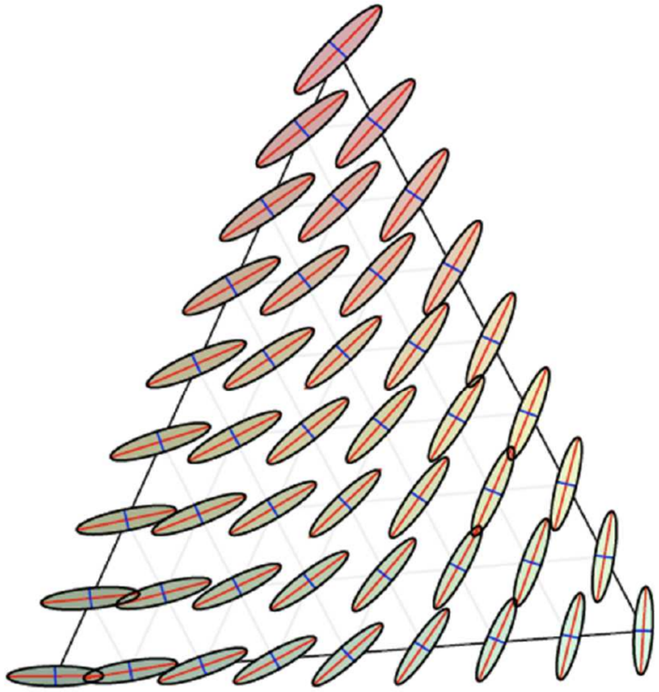


[Hotz]

Derived vector fields

- What interesting vector fields could we consider?
  - Eigenvectors
    - Not a real vector field
      - No magnitude
      - No orientation
  - Magnitude
    - Eigenvalues
  - Ambiguous entity
    - Notion of direction field

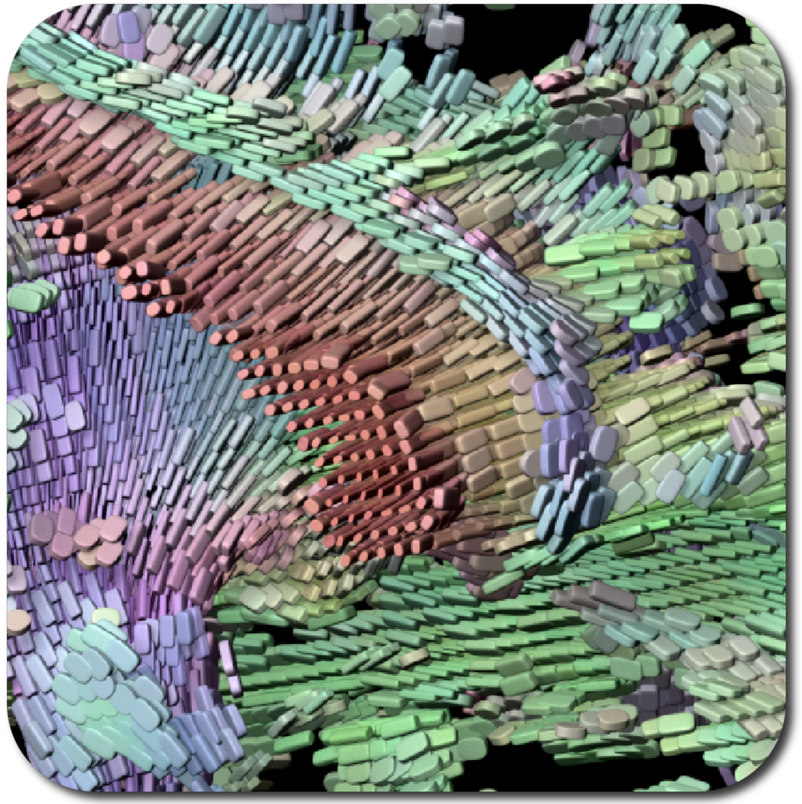




[Hotz]

Direction fields

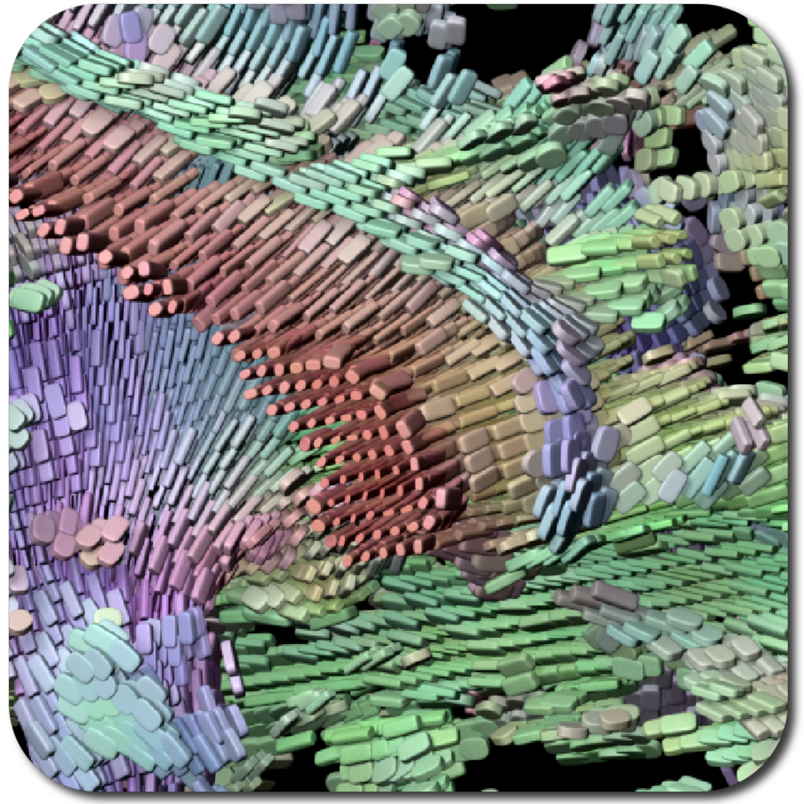
- Set of orthogonal “pseudo” vectors



[Kindlmann]

Direction fields

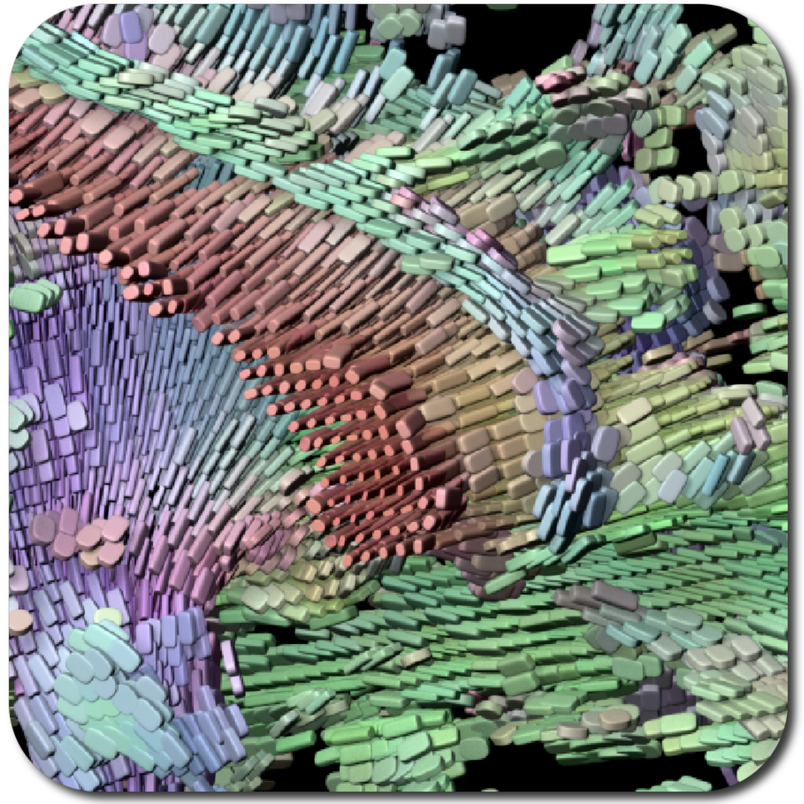
- Set of orthogonal “pseudo” vectors
  - Pulling all the vector field visualization techniques



[Kindlmann]

Direction fields

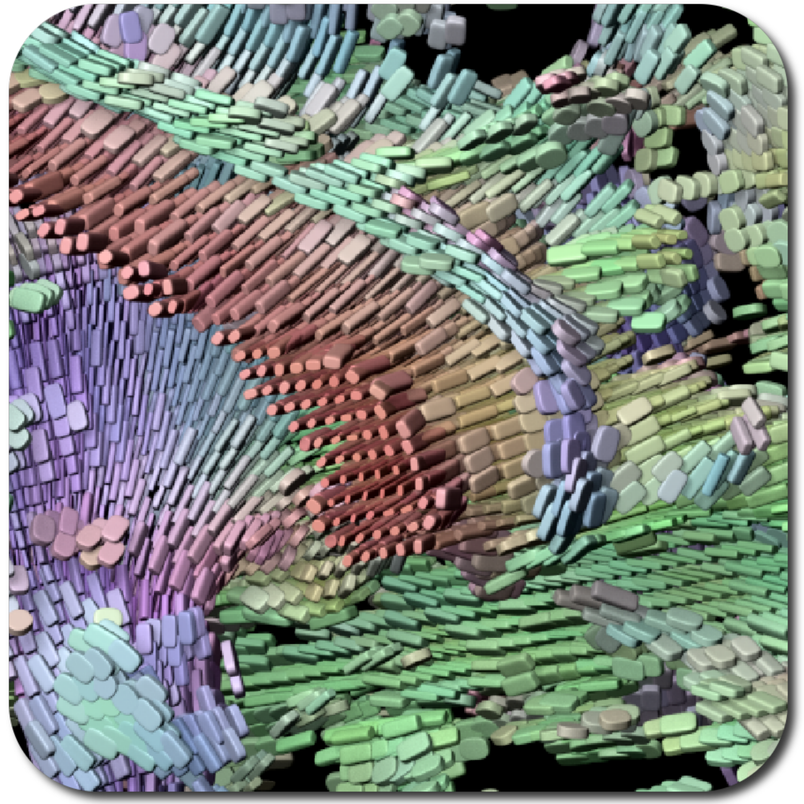
- Set of orthogonal “pseudo” vectors
  - Pulling all the vector field visualization techniques
    - Streamline computation
    - Streamline seeding
    - LIC



[Kindlmann]

Direction fields

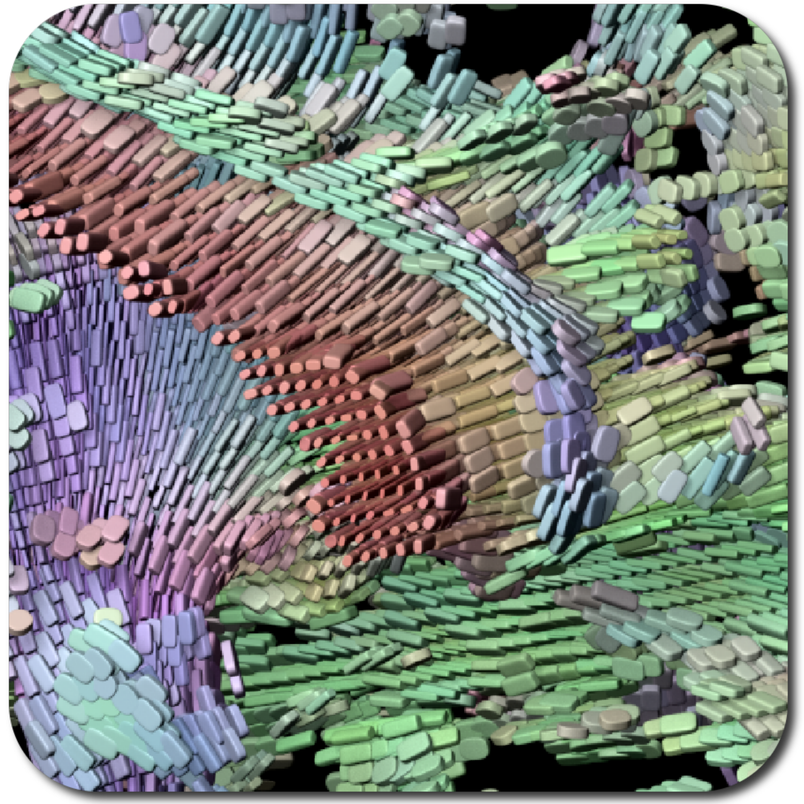
- Set of orthogonal “pseudo” vectors
  - Pulling all the vector field visualization techniques
    - Streamline computation
    - Streamline seeding
    - LIC
  - How to combine the directions?



[Kindlmann]

Direction fields

- Set of orthogonal “pseudo” vectors
  - Pulling all the vector field visualization techniques
    - Streamline computation
    - Streamline seeding
    - LIC
  - How to combine the directions?
    - Get inspiration from …

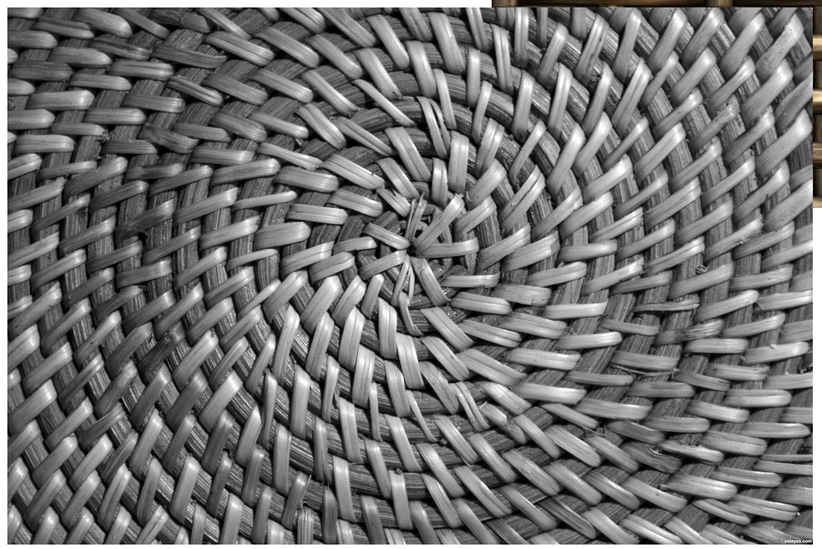


[Kindlmann]

Direction fields

- Set of orthogonal “pseudo” vectors
  - Pulling all the vector field visualization techniques
    - Streamline computation
    - Streamline seeding
    - LIC
  - How to combine the directions?
    - Get inspiration from … craft

[http://www.123rf.com]

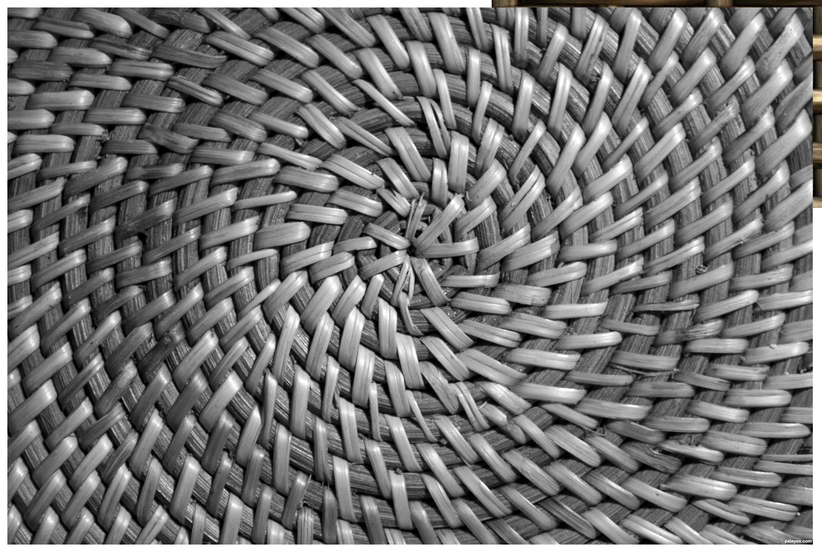


[http://www.pxleyes.com]

Direction fields

- Set of orthogonal “pseudo” vectors
  - Pulling all the vector field visualization techniques
    - Streamline computation
    - Streamline seeding
    - LIC
  - How to combine the directions?
    - Get inspiration from … craft
    - Overlay the directions

[http://www.123rf.com]

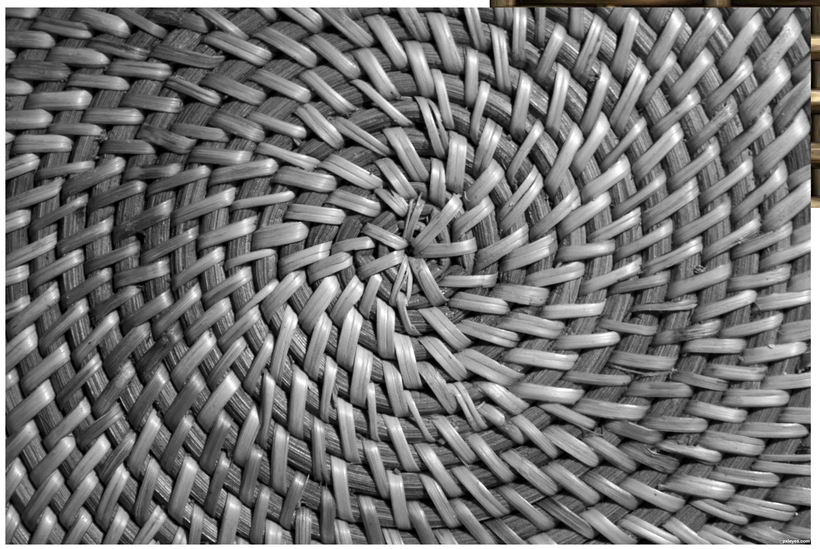


[http://www.pxleyes.com]

Hyper-streamlines

- For each of the d directions

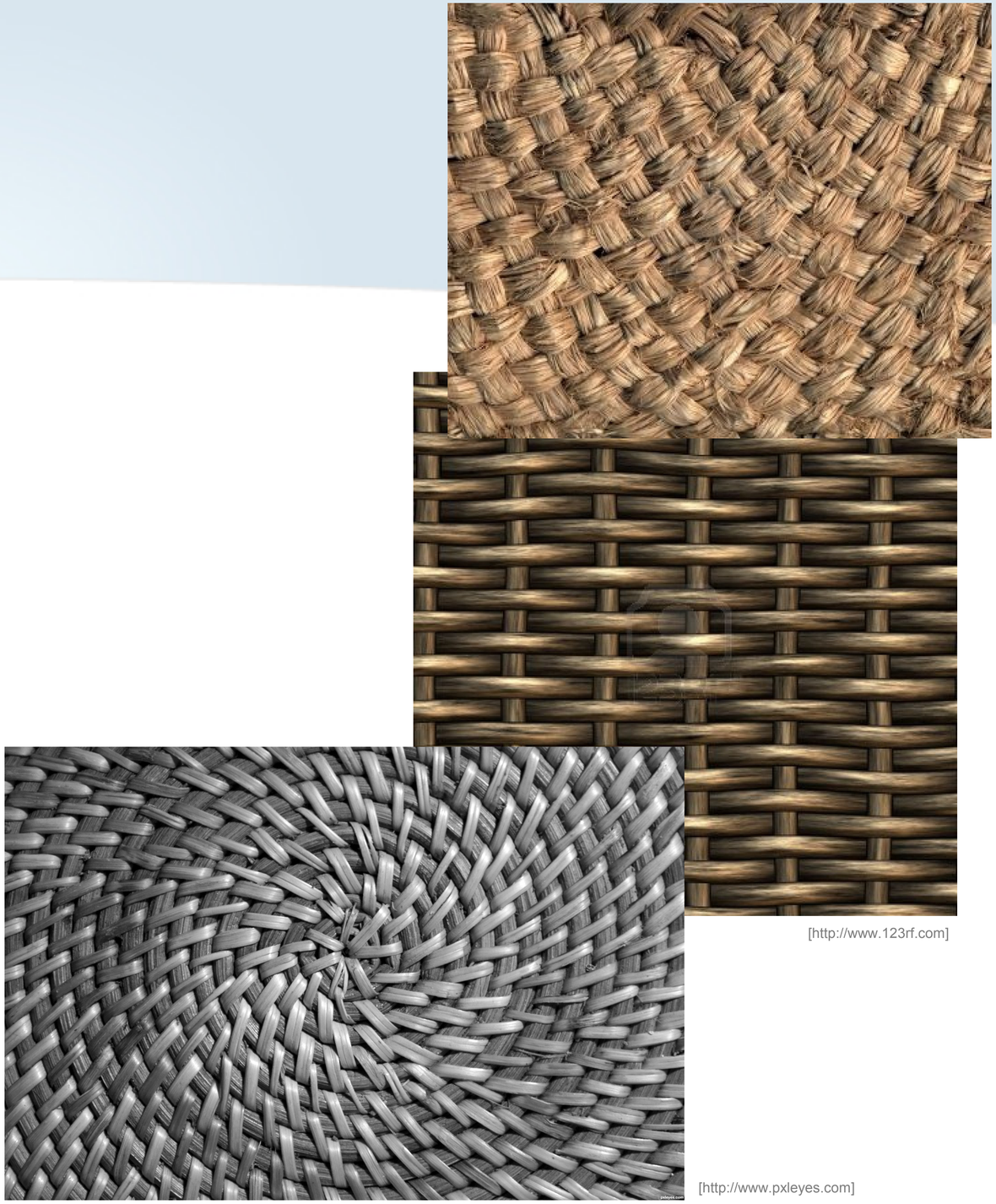
[http://www.123rf.com]



[http://www.pxleyes.com]

Hyper-streamlines

- For each of the d directions
  - Streamline integration

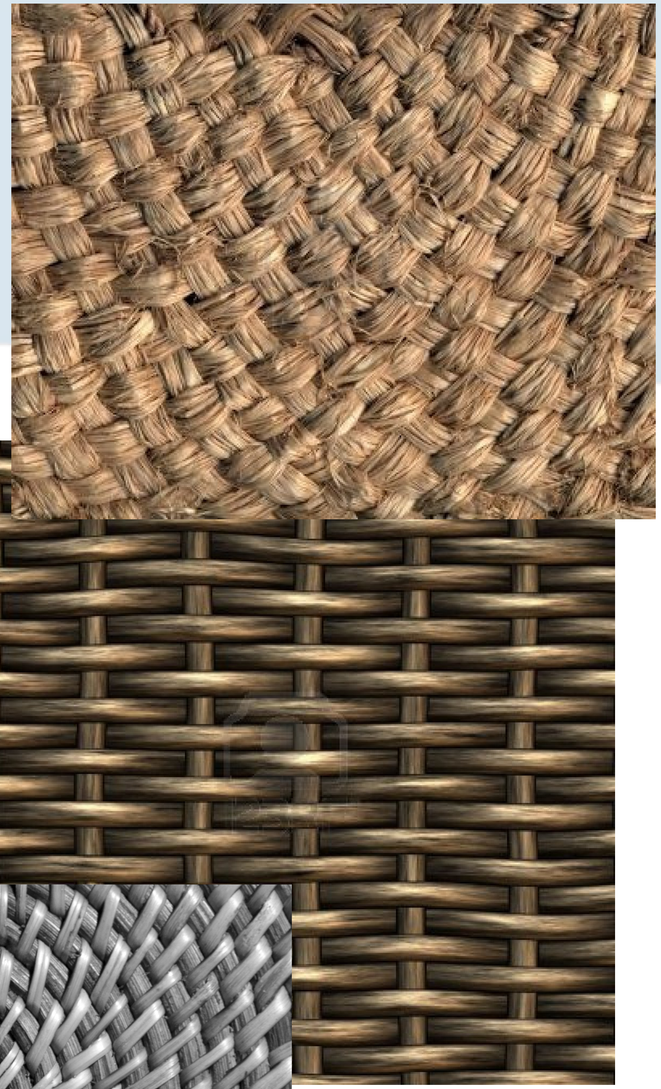


[http://www.123rf.com]

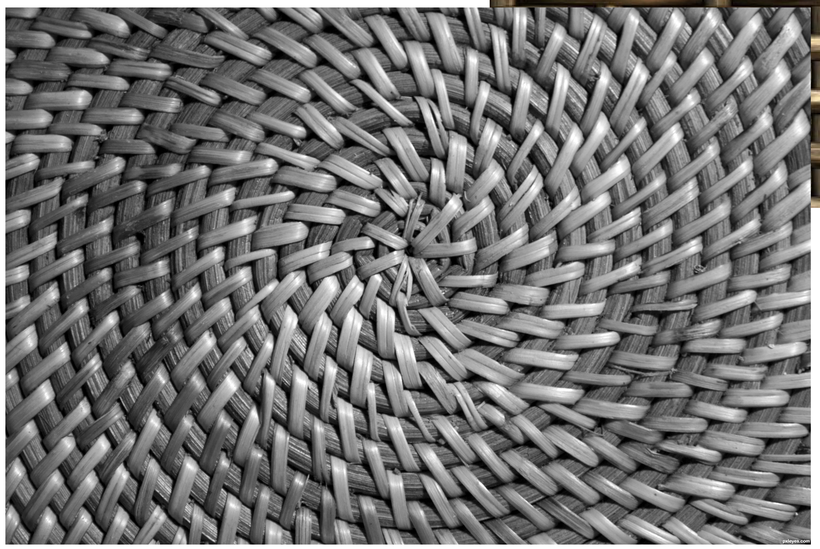
[http://www.pxleyes.com]

Hyper-streamlines

- For each of the d directions
  - Streamline integration
    - We know how to do it for vector fields



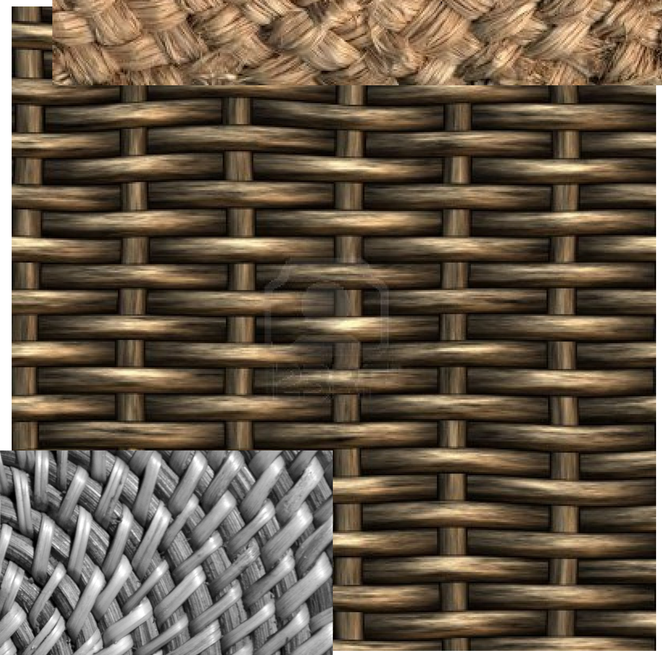
[http://www.123rf.com]



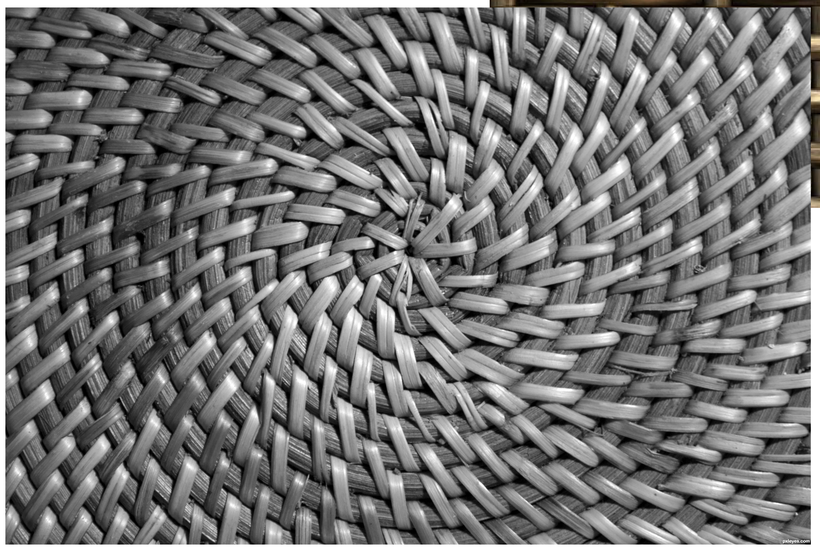
[http://www.pxleyes.com]

Hyper-streamlines

- For each of the d directions
  - Streamline integration
    - We know how to do it for vector fields
  - Problem



[http://www.123rf.com]

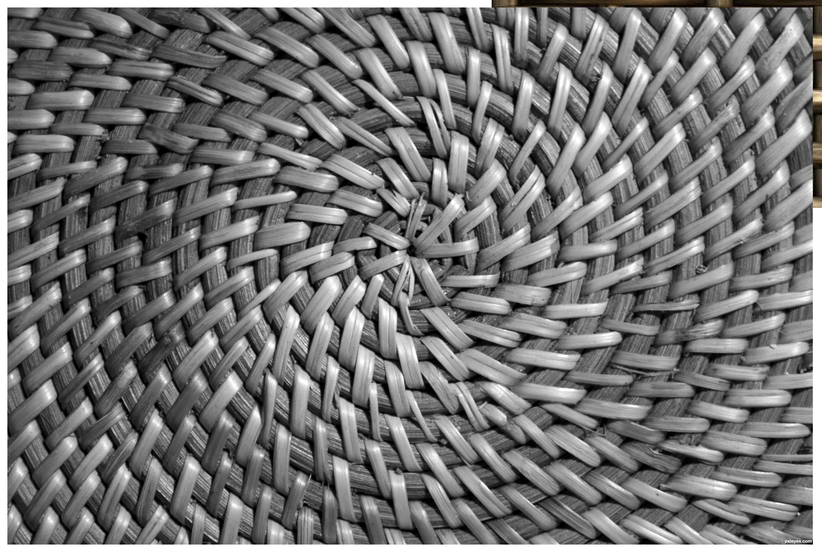


[http://www.pxleyes.com]

Hyper-streamlines

- For each of the d directions
  - Streamline integration
    - We know how to do it for vector fields
  - Problem
    - No orientation

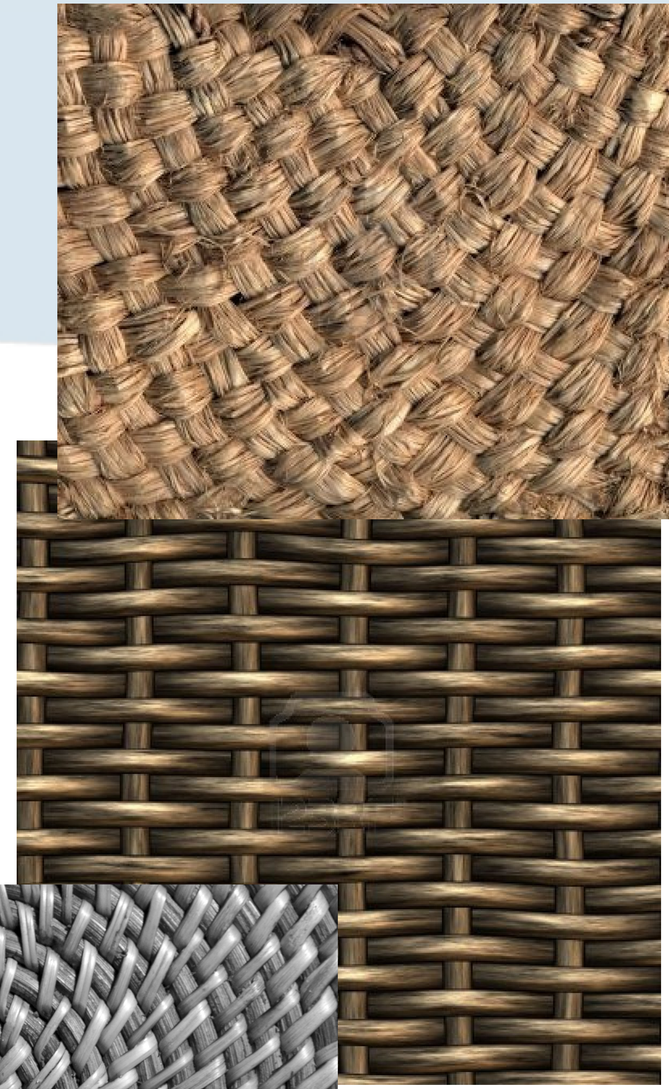
[http://www.123rf.com]



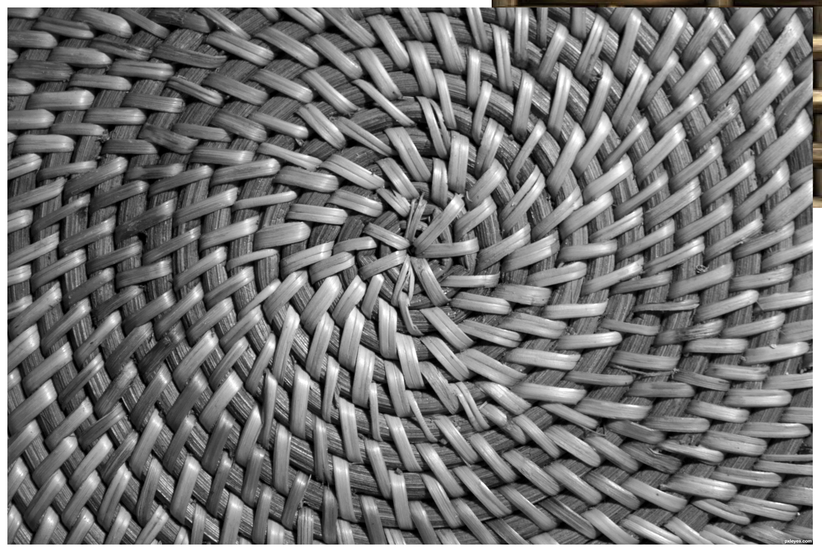
[http://www.pxleyes.com]

Hyper-streamlines

- For each of the d directions
  - Streamline integration
    - We know how to do it for vector fields
  - Problem
    - No orientation
    - No clear matches across cells



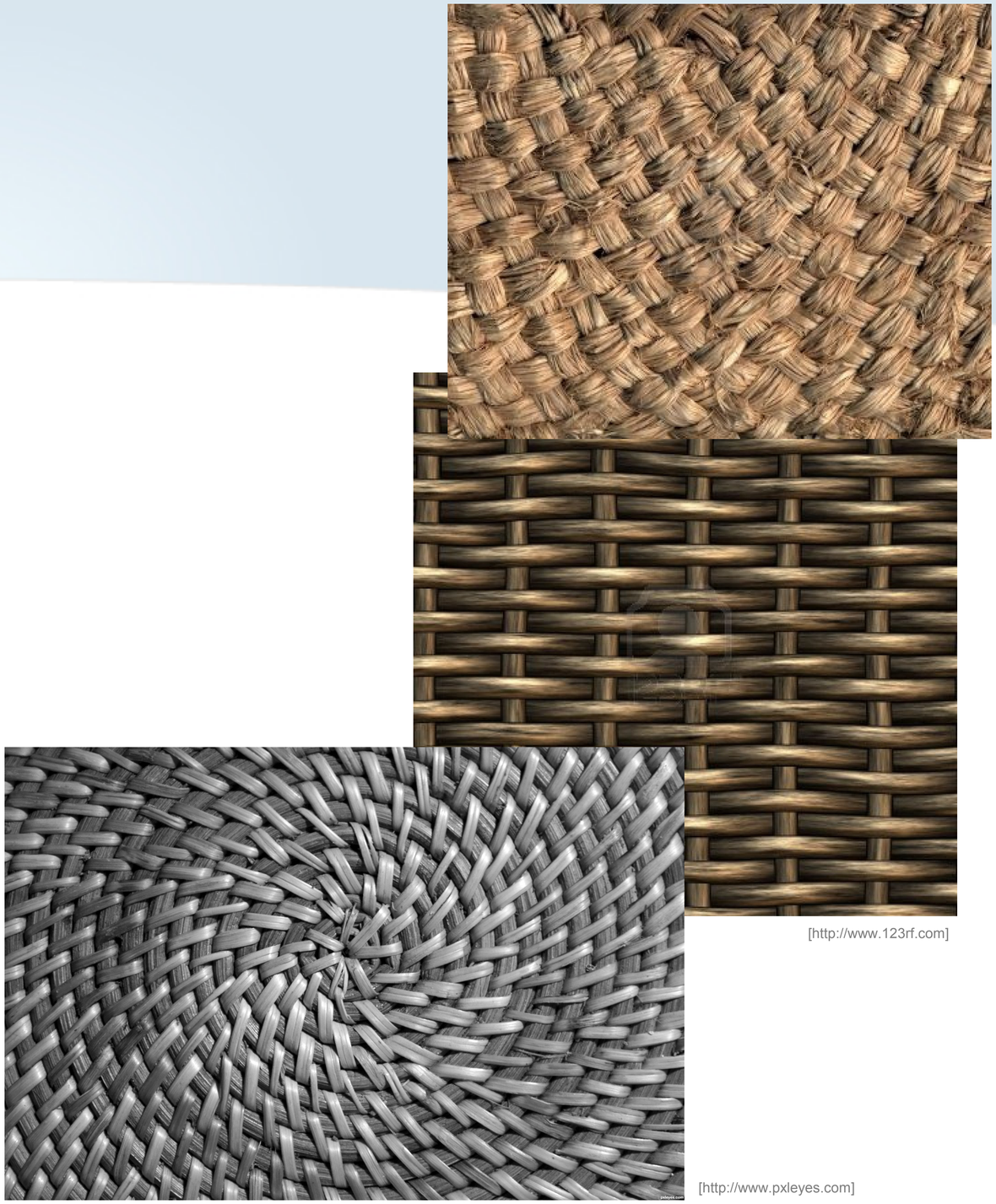
[http://www.123rf.com]



[http://www.pxleyes.com]

Hyper-streamlines

- Other than that...

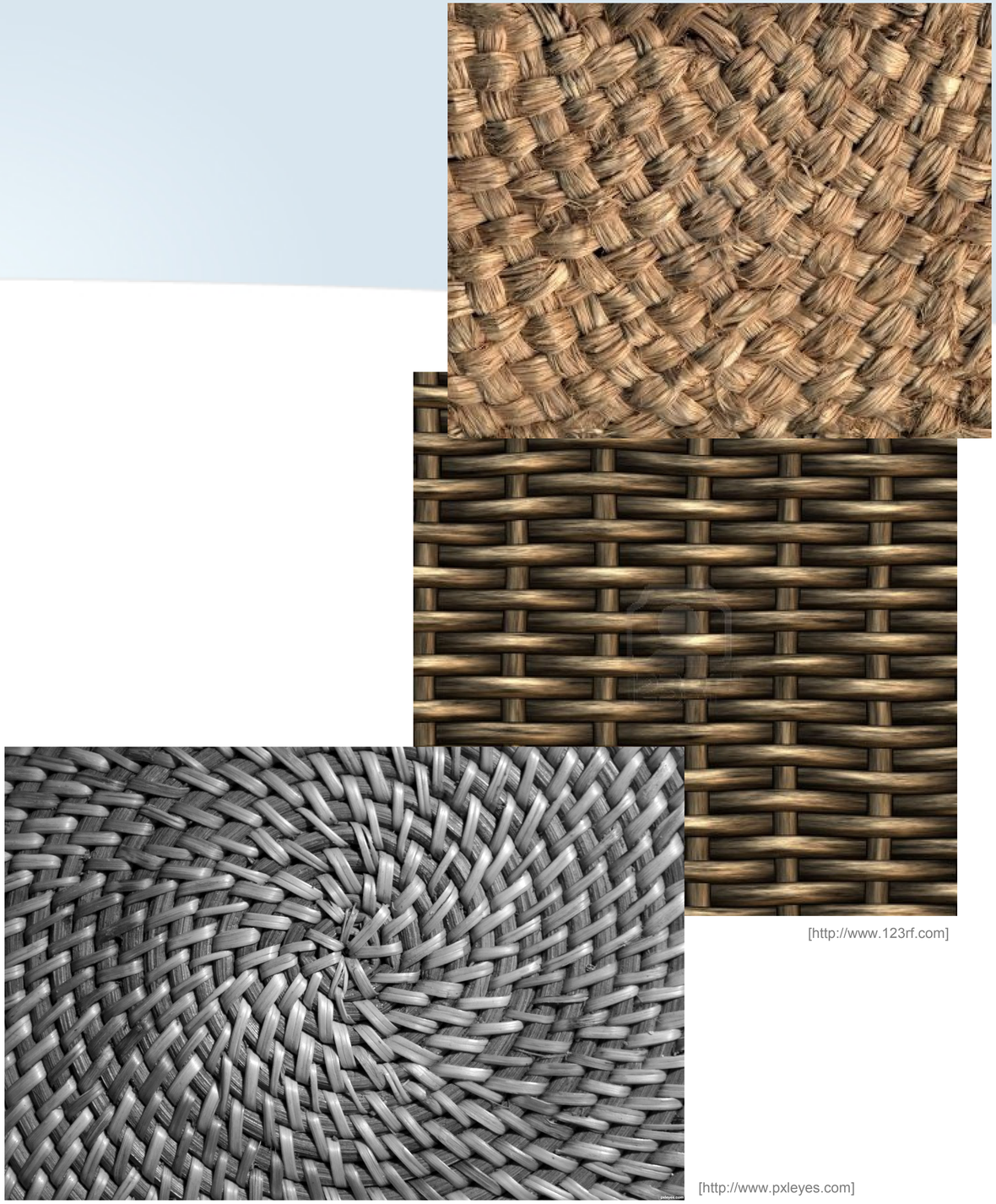


[http://www.123rf.com]

[http://www.pxleyes.com]

Hyper-streamlines

- Other than that...
  - Same integration as for vector fields

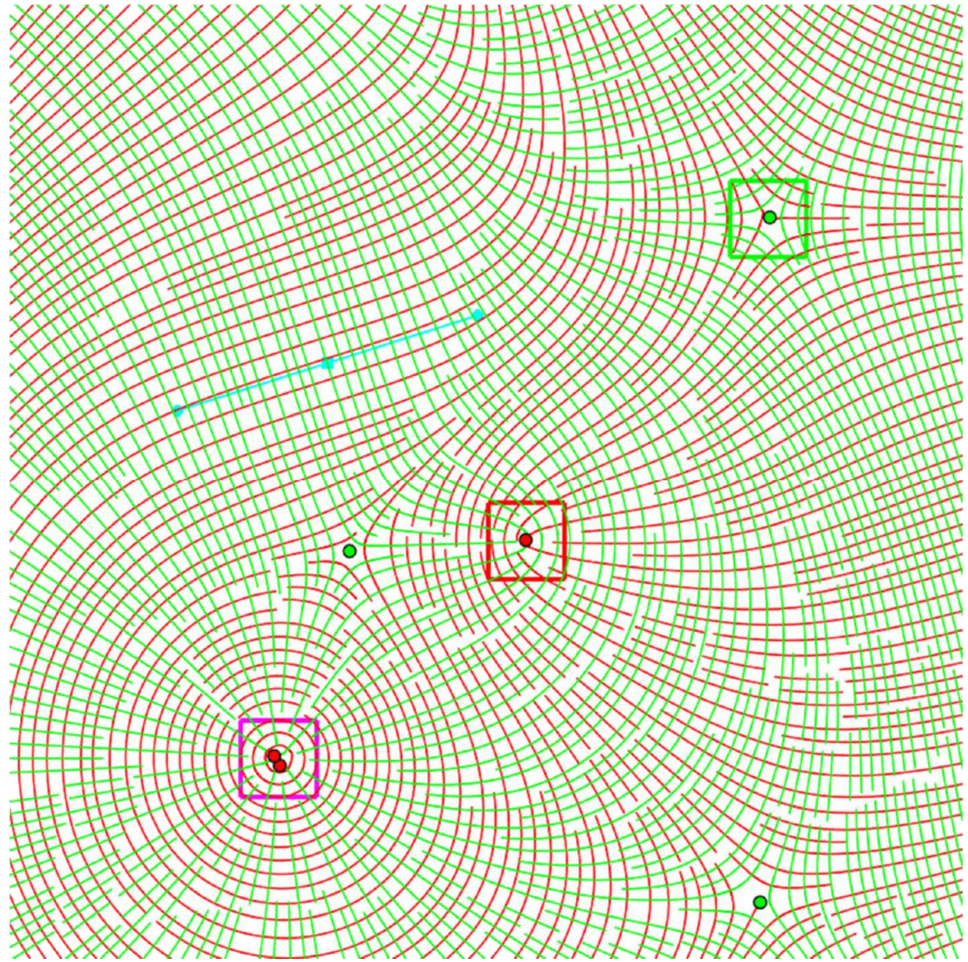


[http://www.123rf.com]

[http://www.pxleyes.com]

Hyper-streamlines

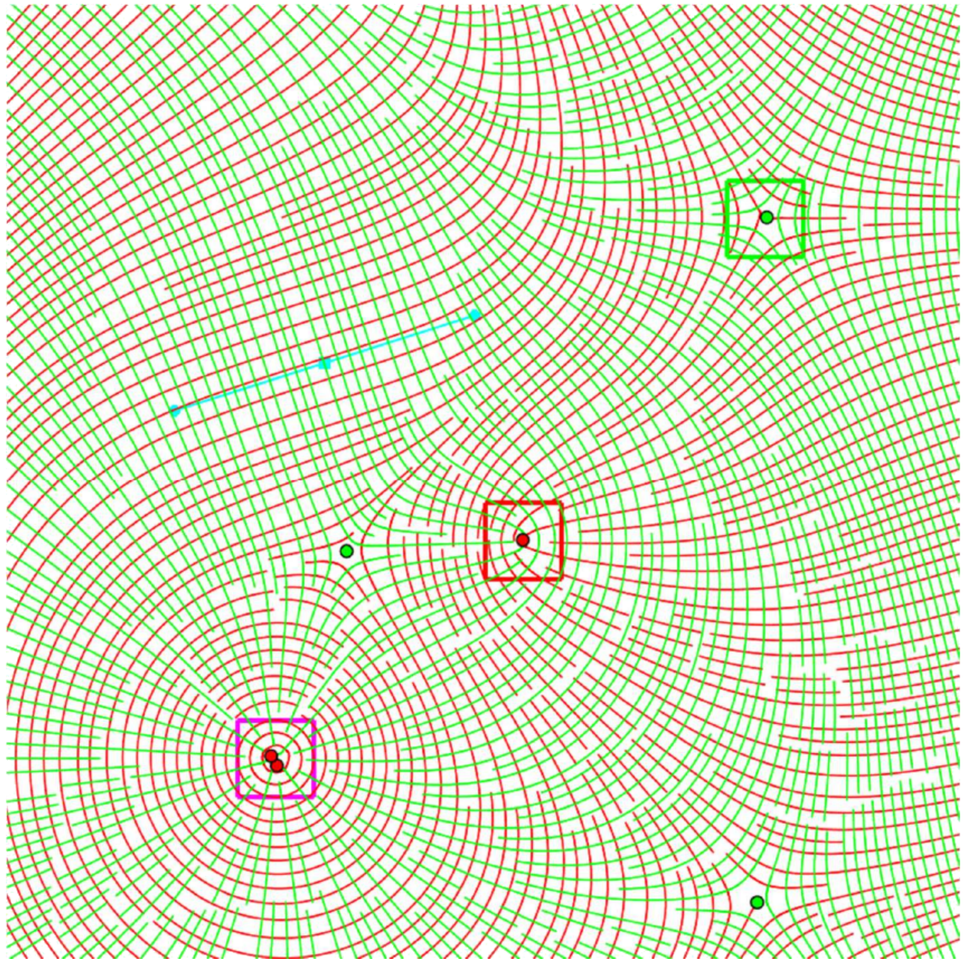
- Other than that...
  - Same integration as for vector fields



[Chen]

Hyper-streamlines

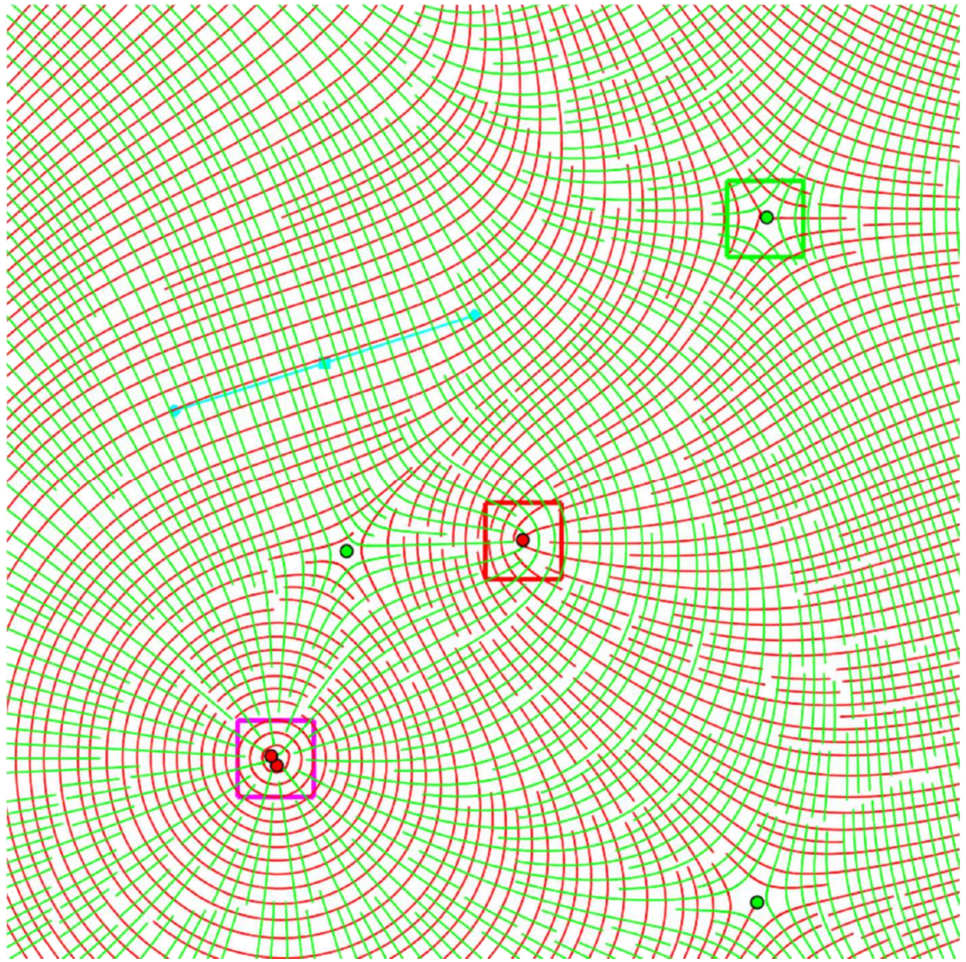
- Other than that...
  - Same integration as for vector fields
- Recap



[Chen]

Hyper-streamlines

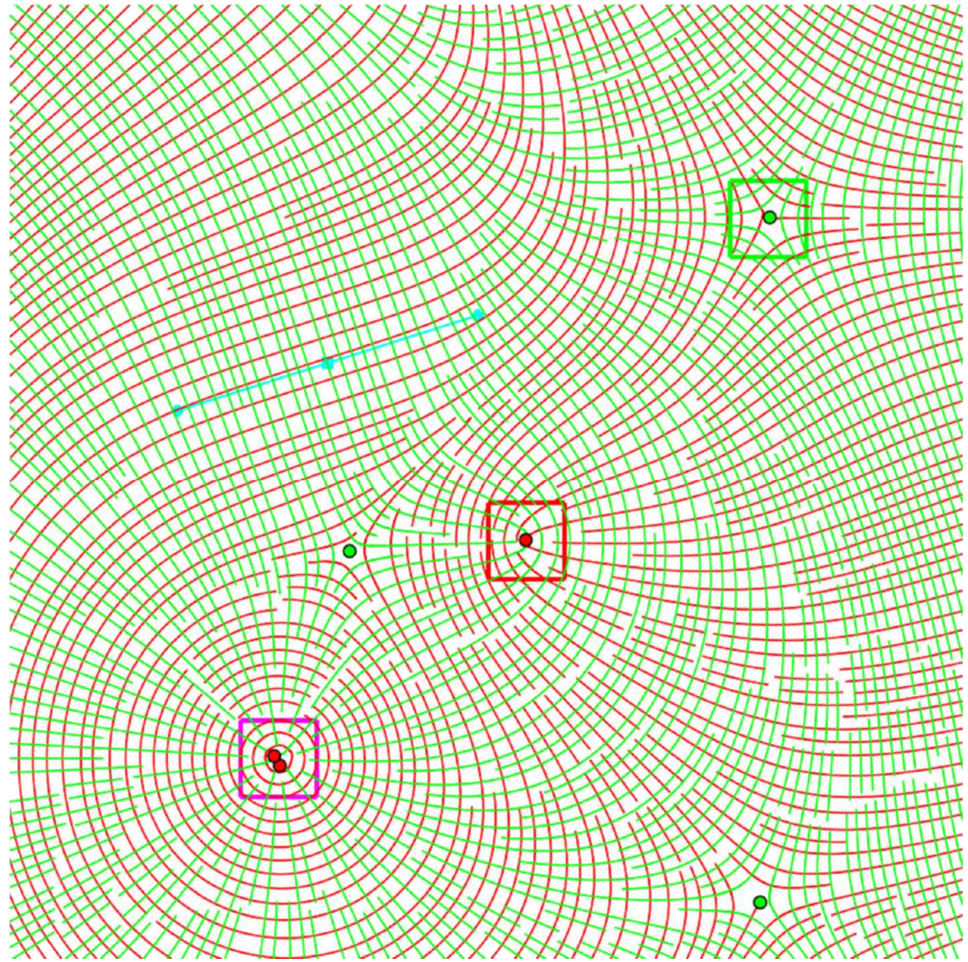
- Other than that...
  - Same integration as for vector fields
- Recap
  - Independent computation for minor/major



[Chen]

Hyper-streamlines

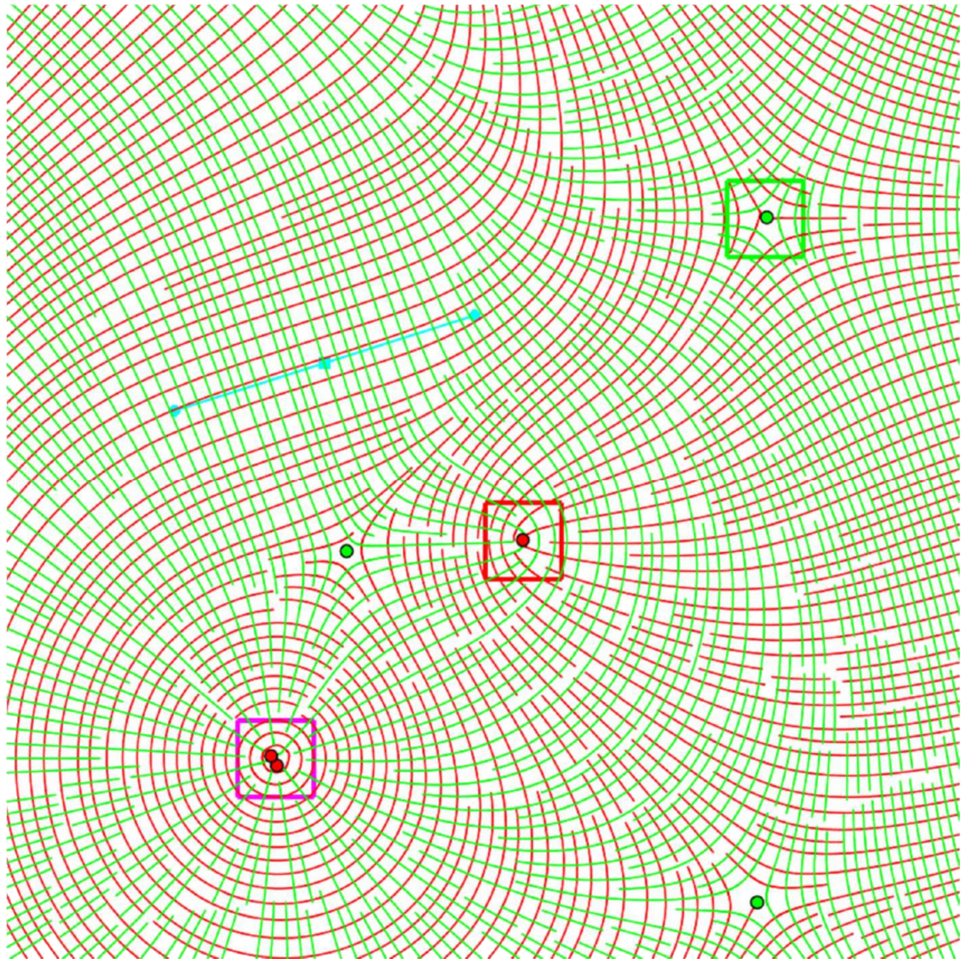
- Other than that...
  - Same integration as for vector fields
- Recap
  - Independent computation for minor/major
  - Seeding
    - Distance criterion



[Chen]

Hyper-streamlines

- Other than that...
  - Same integration as for vector fields
- Recap
  - Independent computation for minor/major
  - Seeding
    - Distance criterion
  - Integration
    - Euler
    - Runge-Kutta

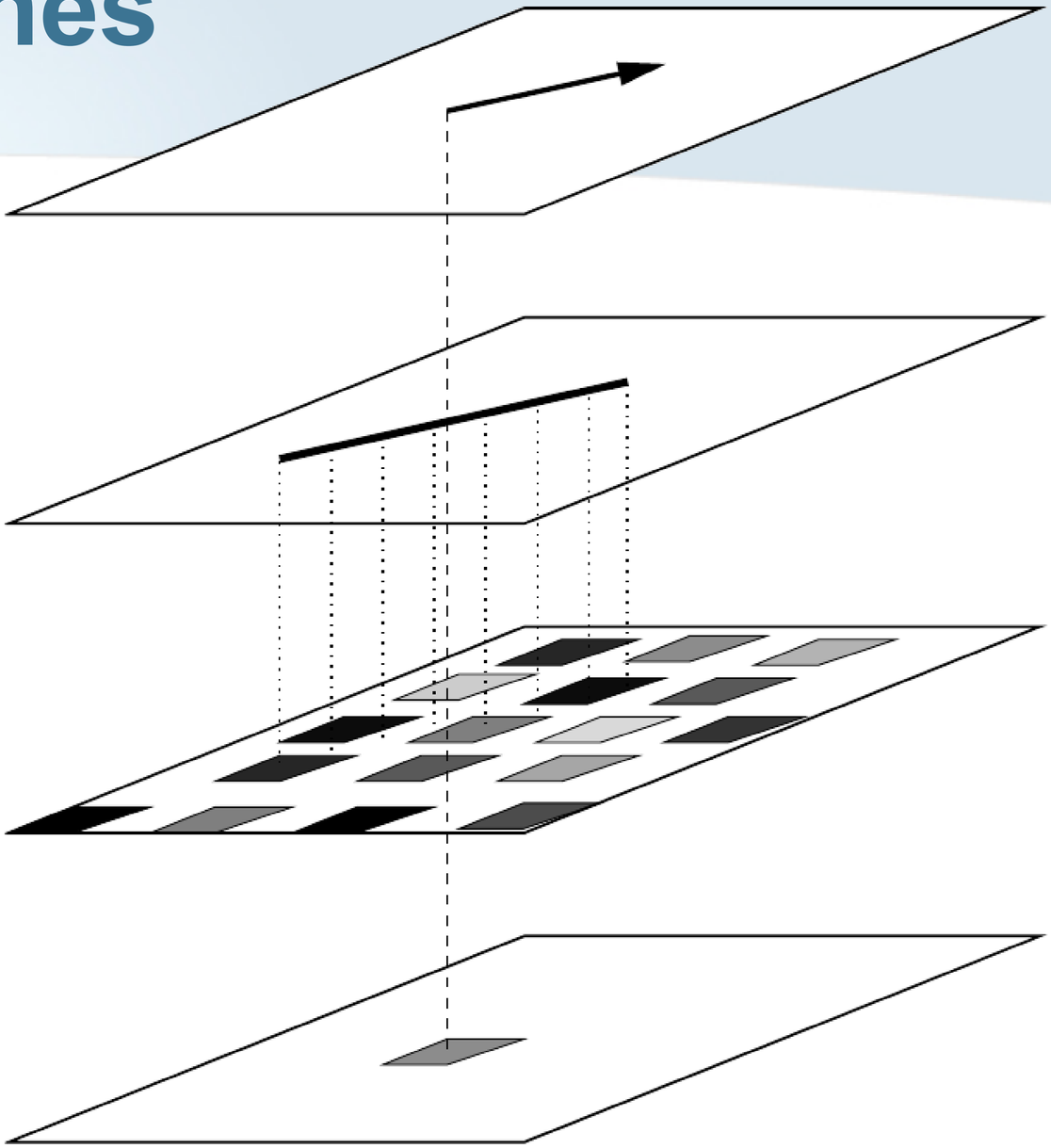


[Chen]

Beyond hyper-streamlines

- Now that we know how to extract hyper-streamlines
  - More global visualization

- Getting inspiration from vector fields
  - Line Integral Convolution

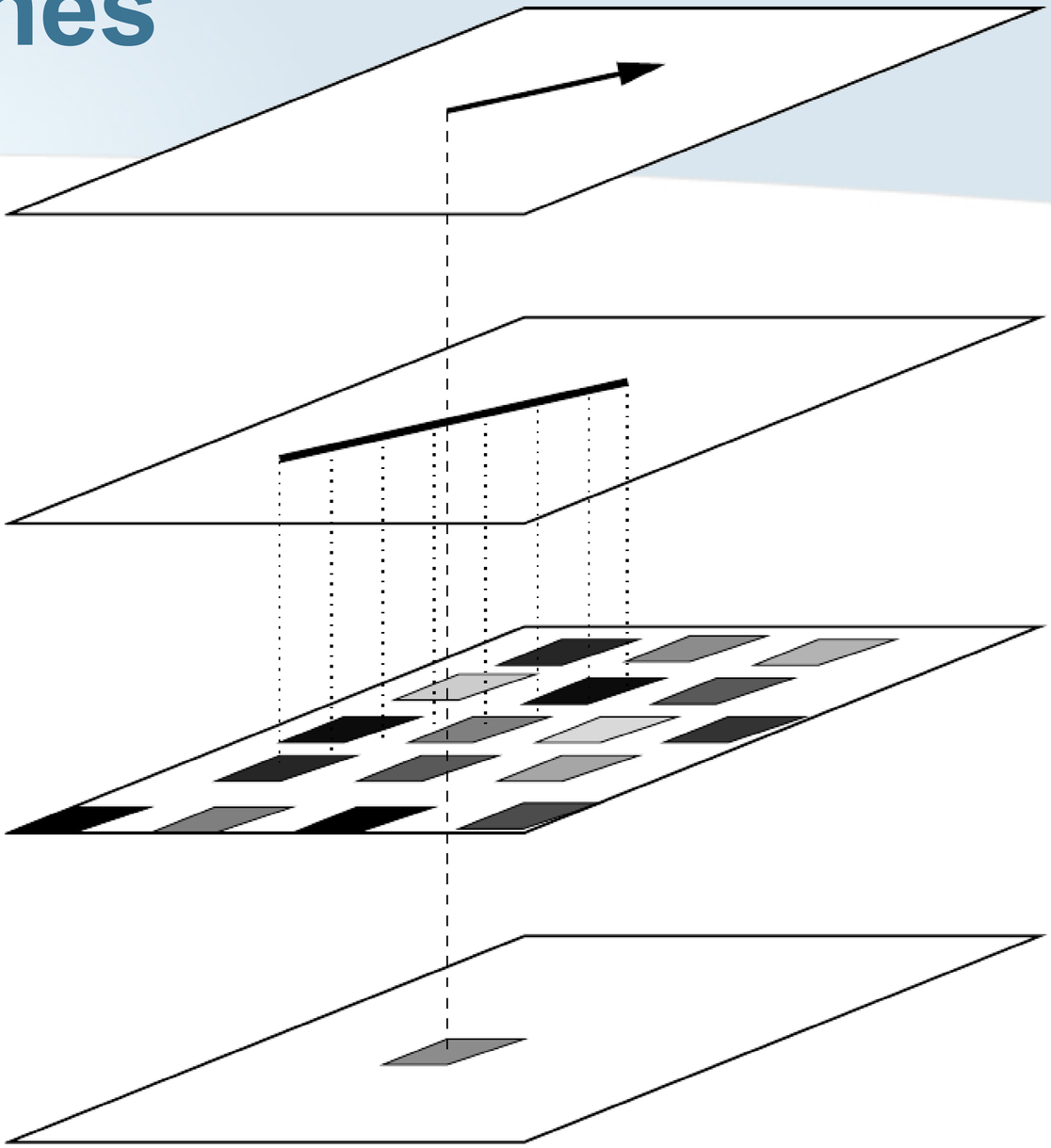


[Helgeland]

Beyond hyper-streamlines

- Now that we know how to extract hyper-streamlines
  - More global visualization

- Getting inspiration from vector fields
  - Line Integral Convolution
    - d-dimensional convolution kernel

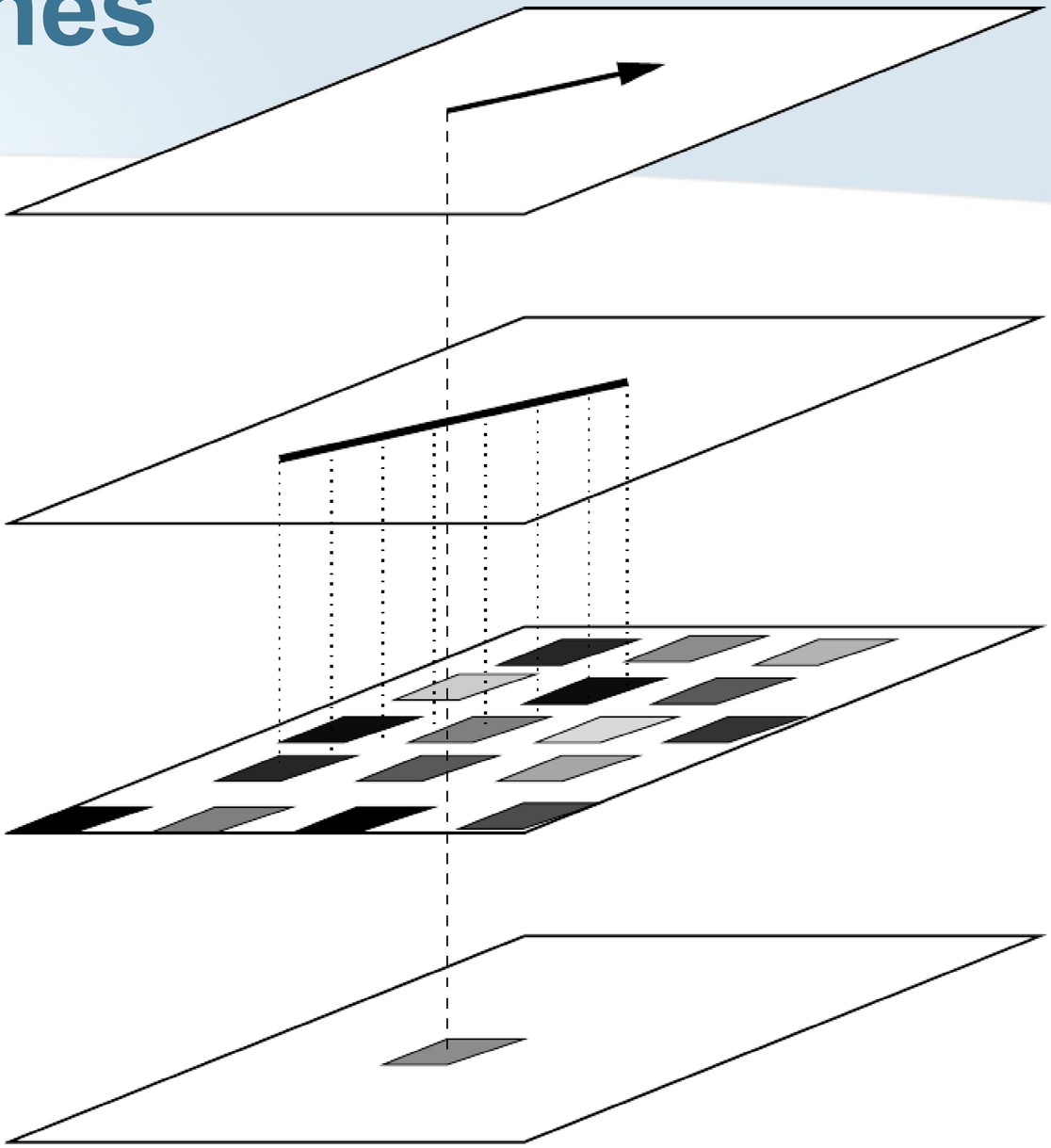


[Helgeland]

Beyond hyper-streamlines

- Now that we know how to extract hyper-streamlines
  - More global visualization

- Getting inspiration from vector fields
  - Line Integral Convolution
    - d-dimensional convolution kernel
    - Independent computations

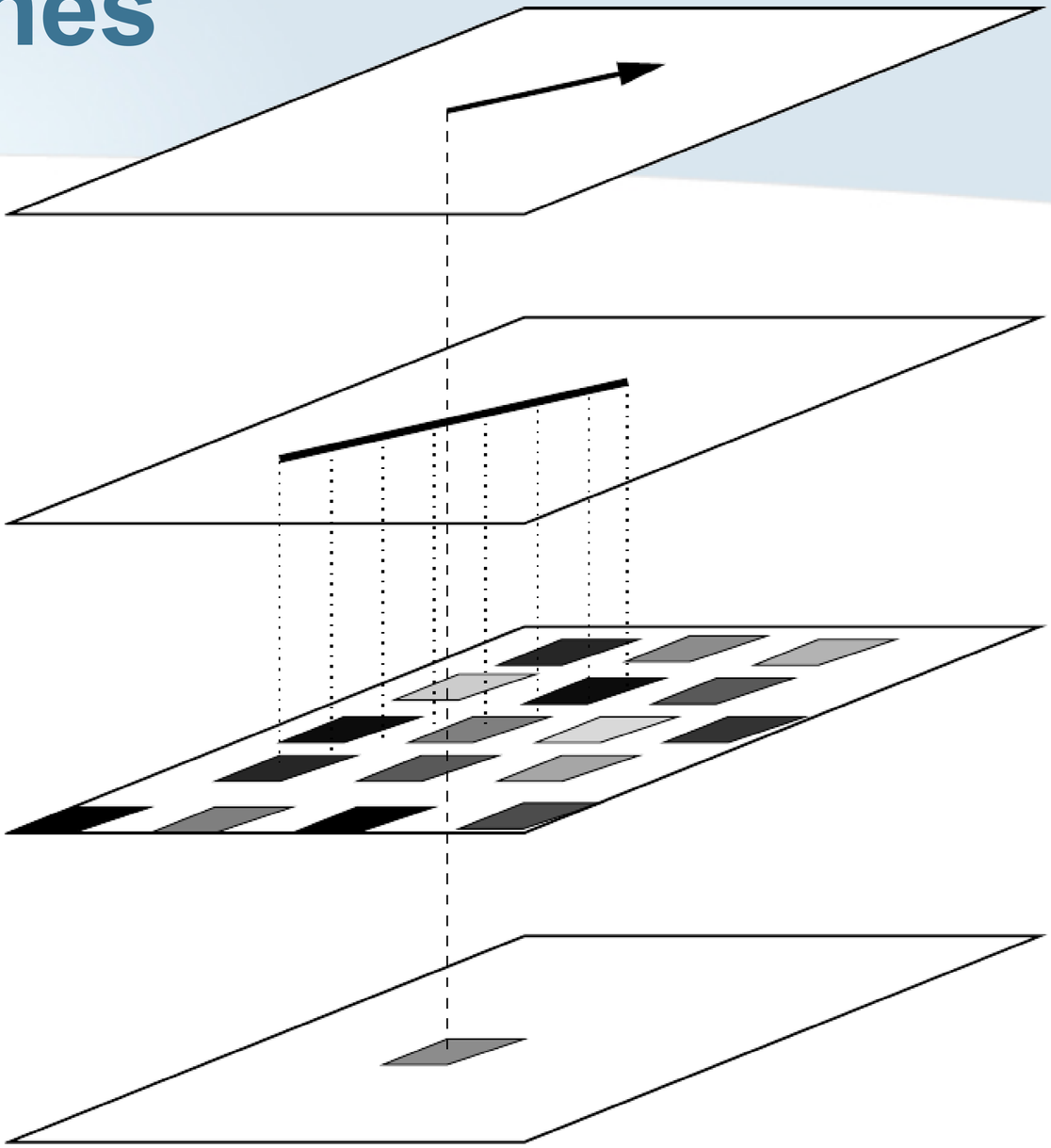


[Helgeland]

Beyond hyper-streamlines

- Now that we know how to extract hyper-streamlines
  - More global visualization

- Getting inspiration from vector fields
  - Line Integral Convolution
    - d-dimensional convolution kernel
    - Independent computations
      - Random blend minor/major

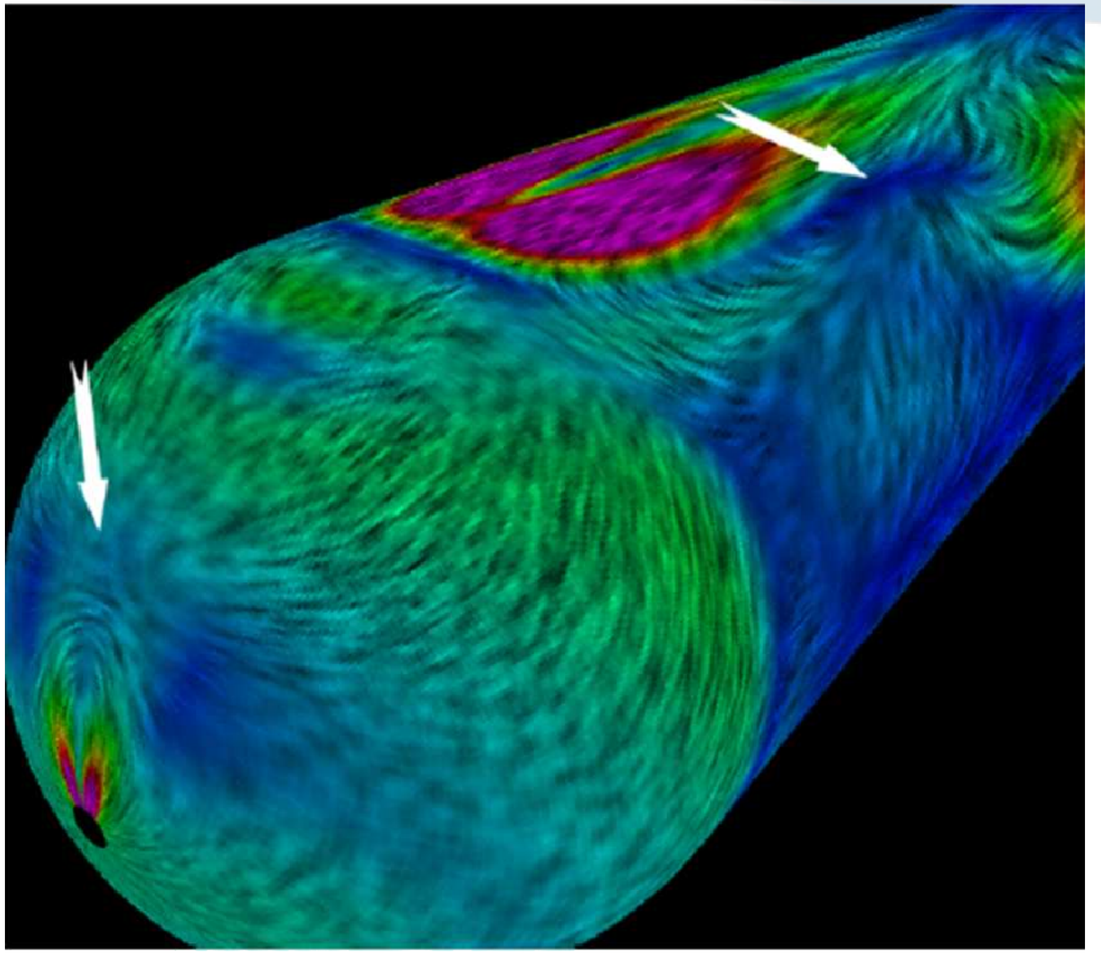


[Helgeland]

Beyond hyper-streamlines

- Now that we know how to extract hyper-streamlines
  - More global visualization

- Getting inspiration from vector fields
  - Line Integral Convolution
    - d-dimensional convolution kernel
    - Independent computations
      - Random blend minor/major

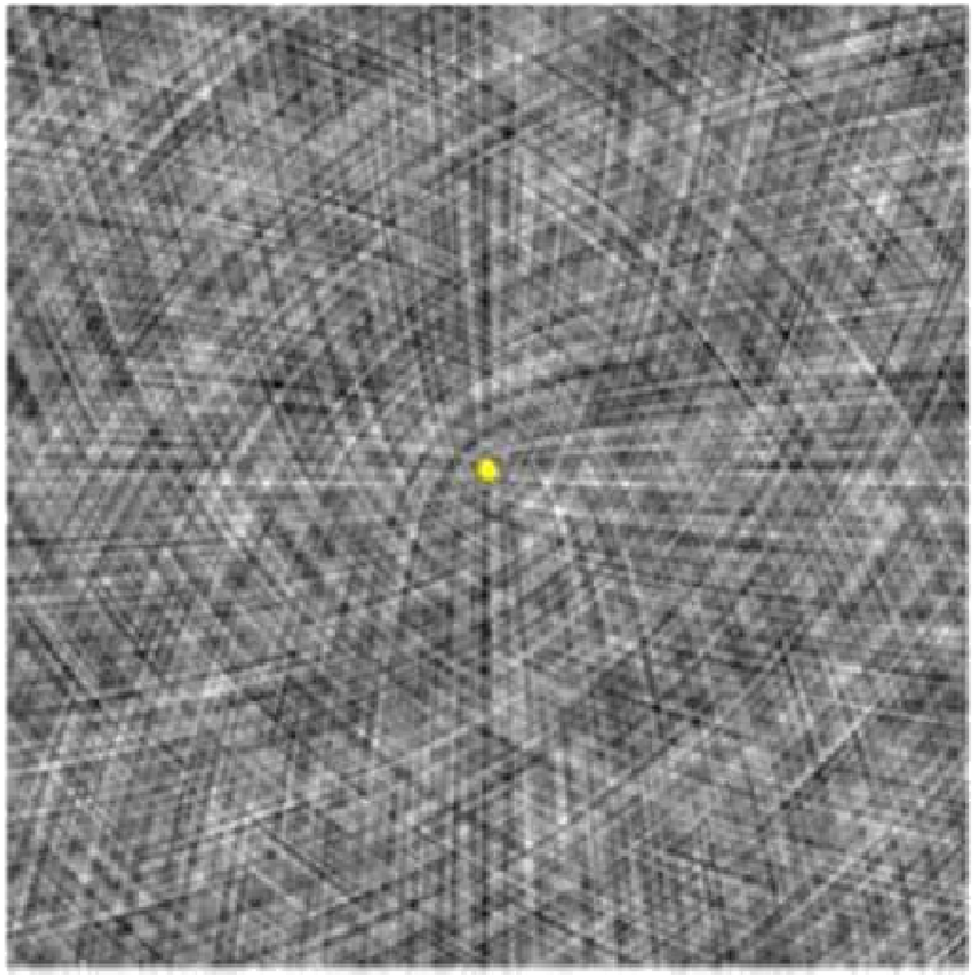


[Chen]

Beyond hyper-streamlines

- Now that we know how to extract hyper-streamlines
  - More global visualization

- Getting inspiration from vector fields
  - Line Integral Convolution
    - d-dimensional convolution kernel
    - Independent computations
      - Random blend minor/major

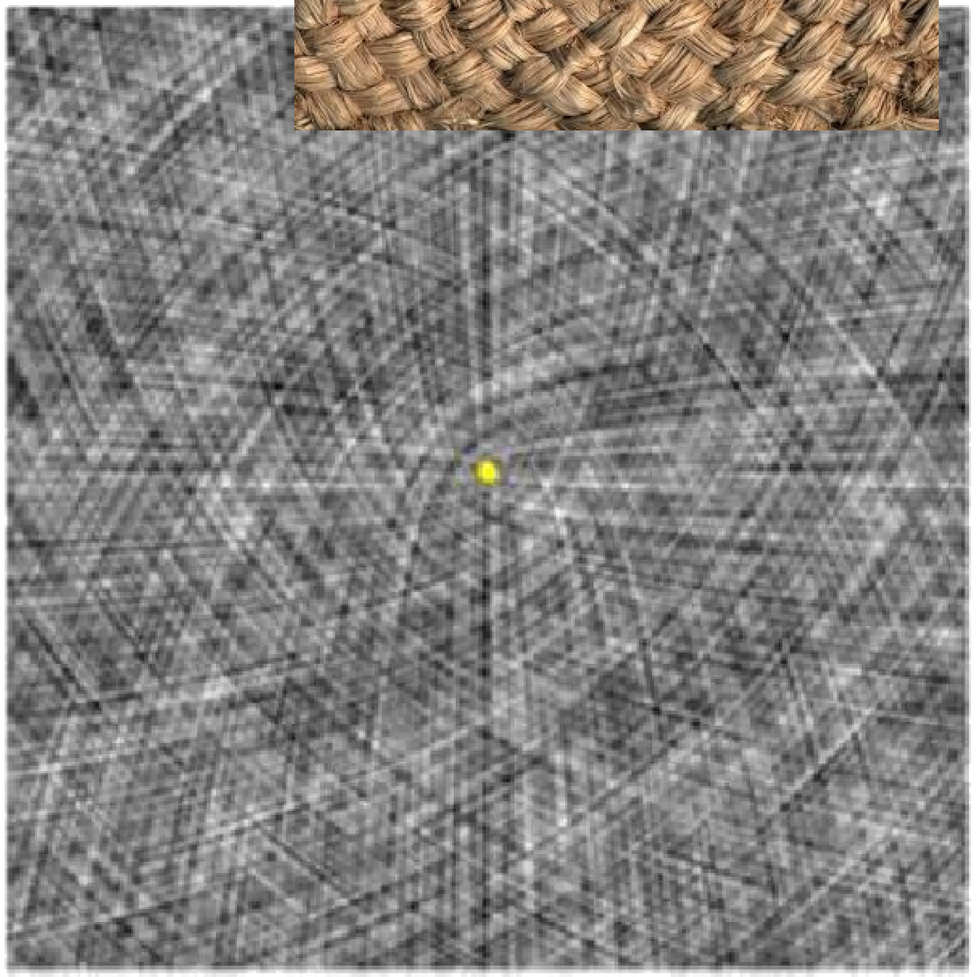


[Zhang]

Beyond hyper-streamlines

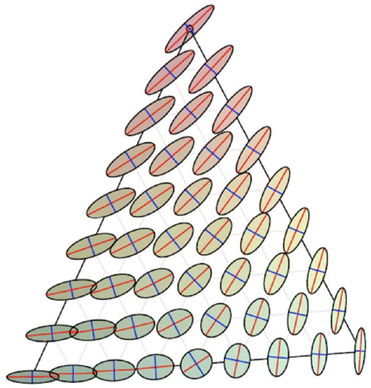
- Now that we know how to extract hyper-streamlines
  - More global visualization

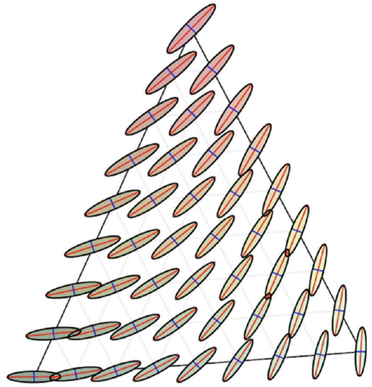
- Getting inspiration from vector fields
  - Line Integral Convolution
    - d-dimensional convolution kernel
    - Independent computations
      - Random blend minor/major

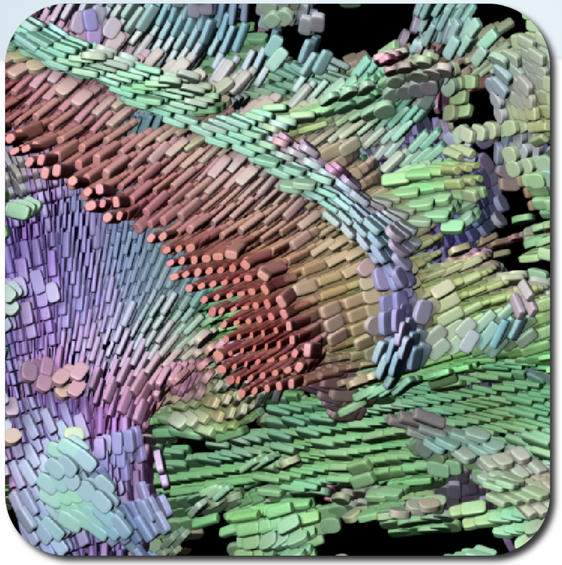


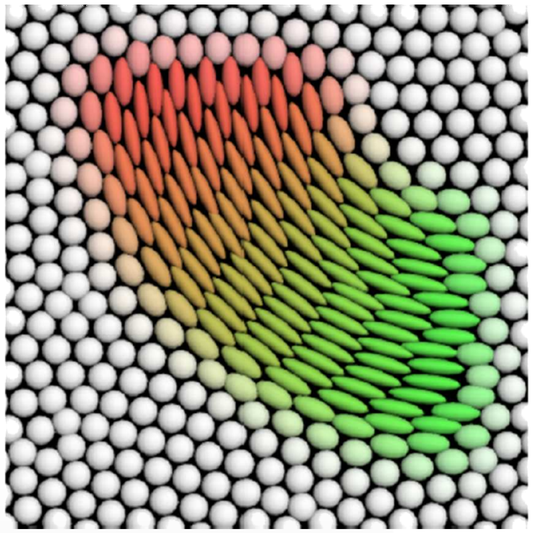
[Zhang]

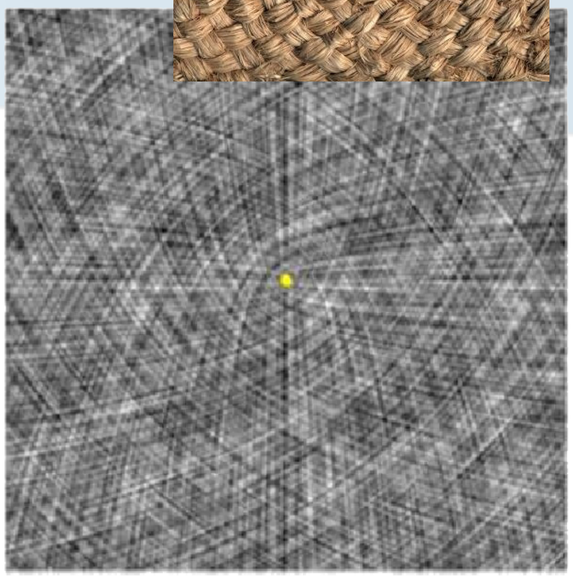
Tensor fields

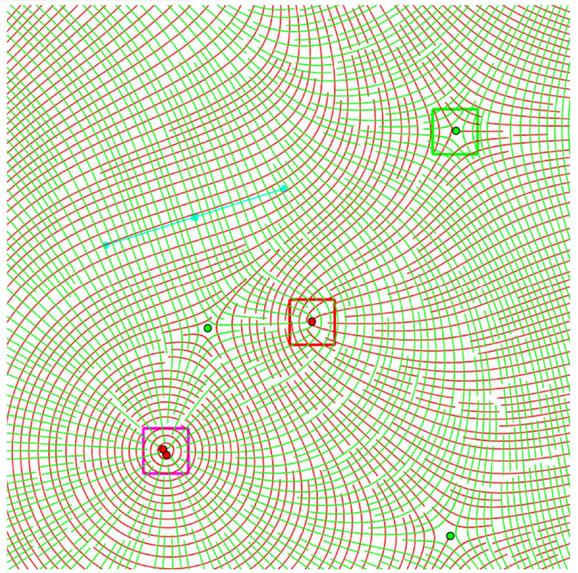


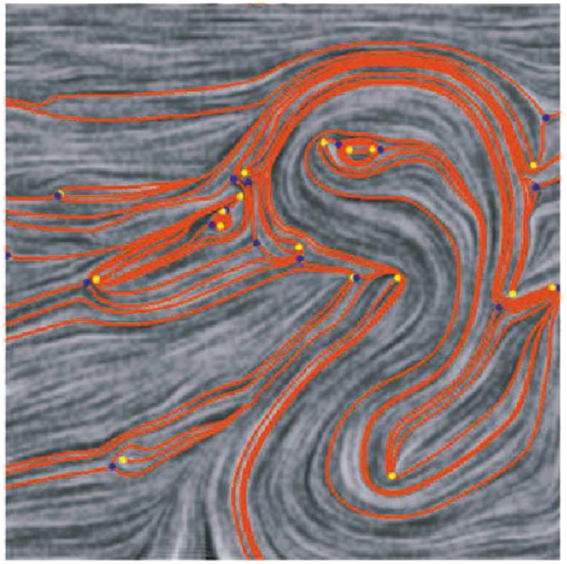










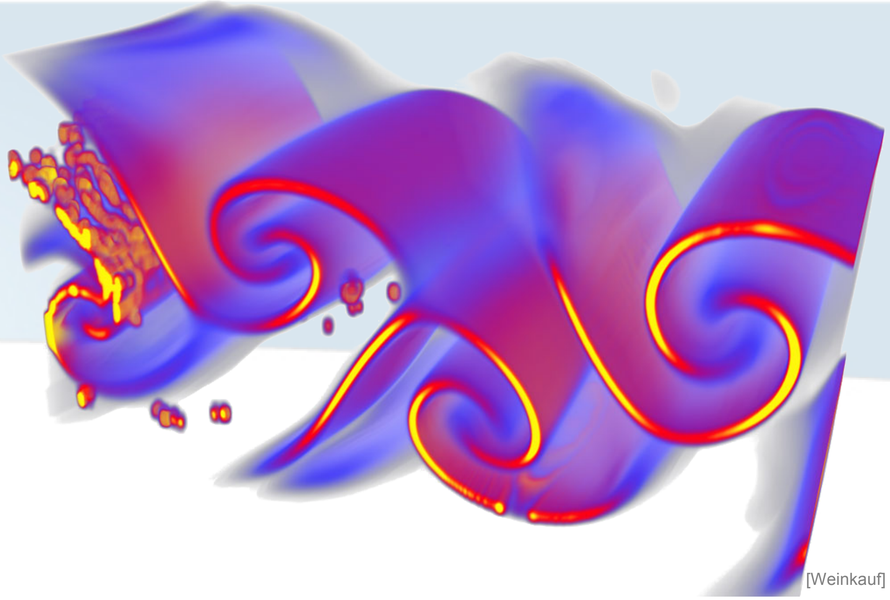


The awful truth

The awful truth

- Simulations usually have a temporal component

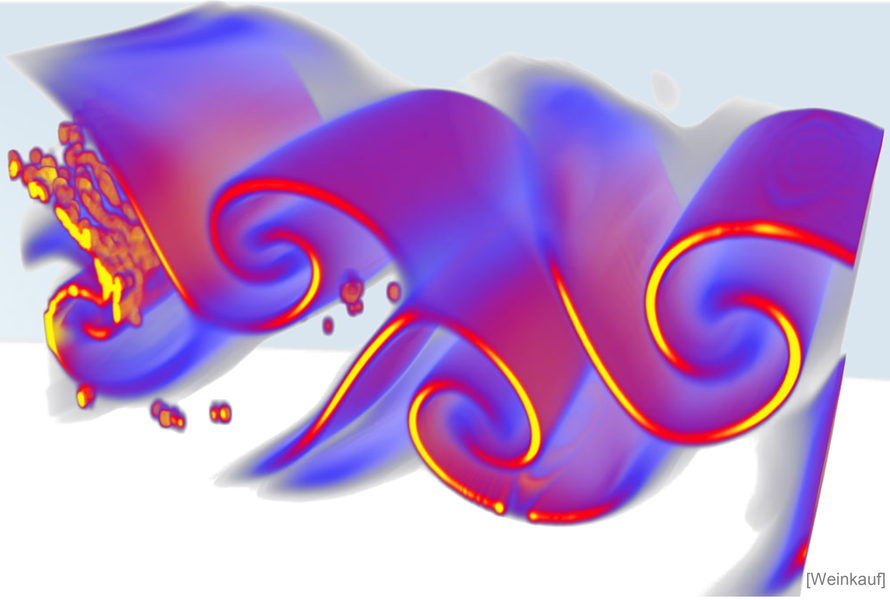
The awful truth



[Weinkauf]

- Simulations usually have a temporal component

The awful truth

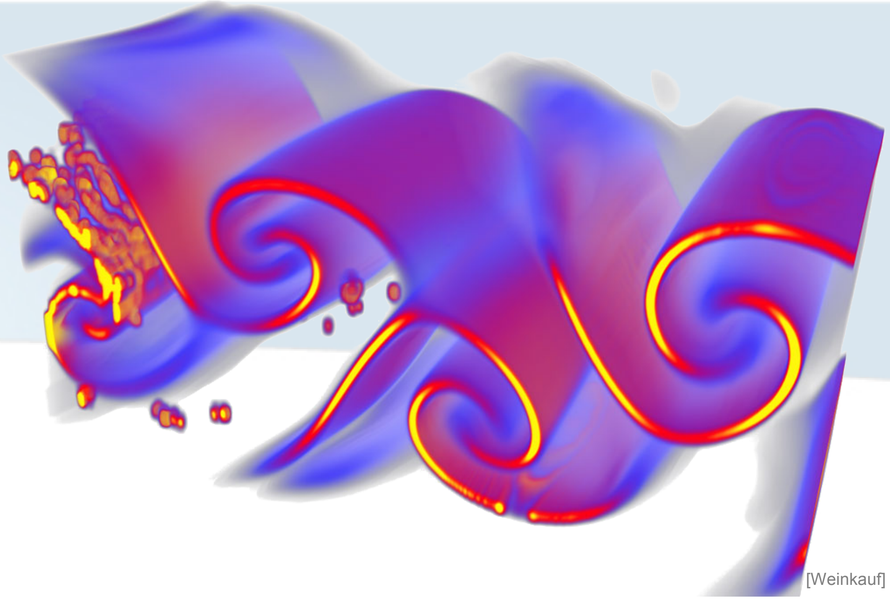


[Weinkauf]

- Simulations usually have a temporal component

- Simulations often come uncertainty evaluation

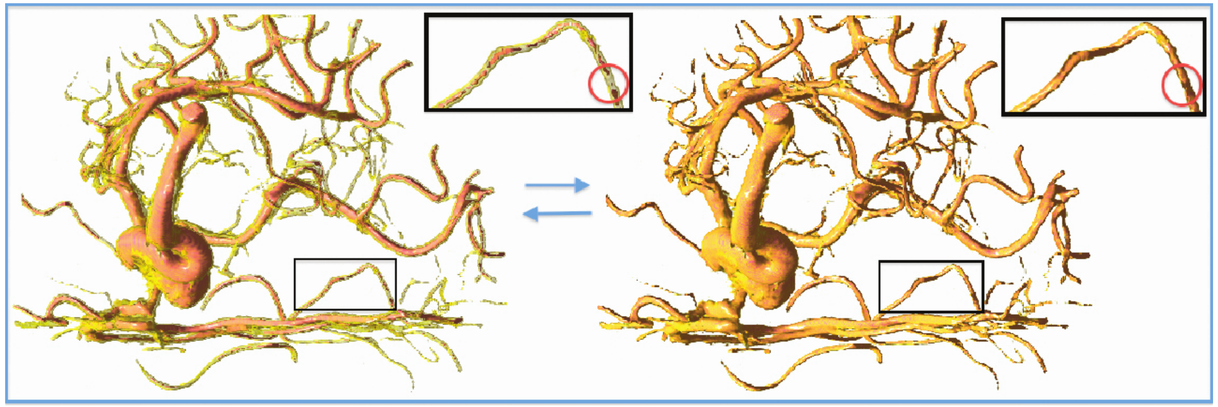
The awful truth



[Weinkauf]

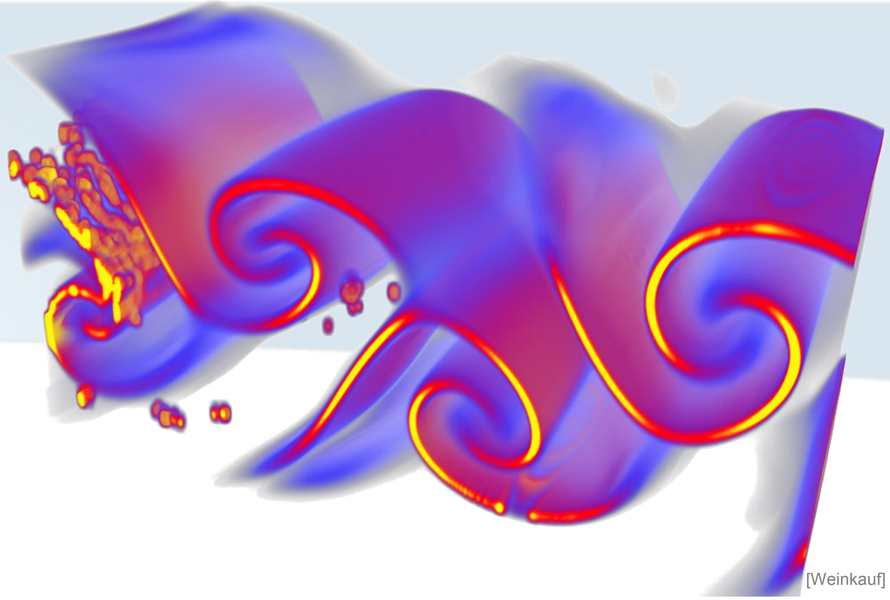
- Simulations usually have a temporal component

- Simulations often come uncertainty evaluation



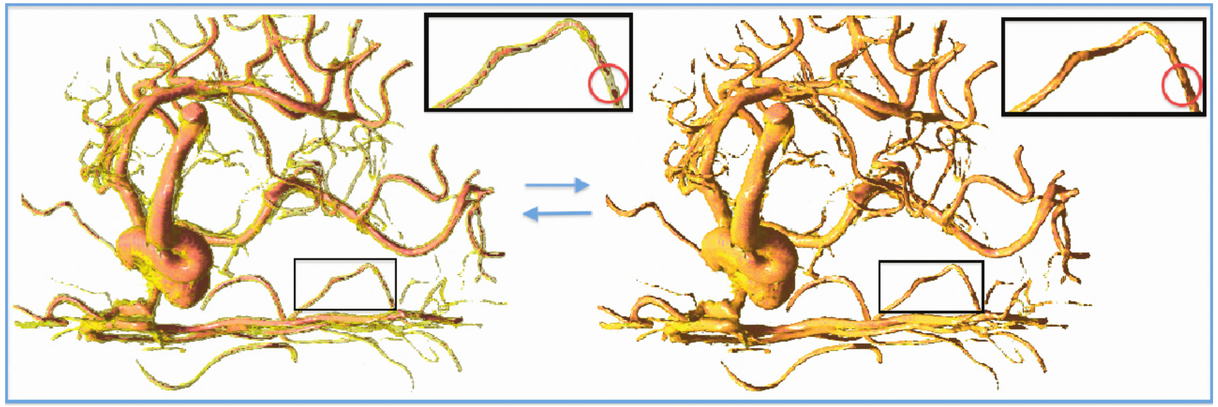
[Fout]

The awful truth



[Weinkauf]

- Simulations usually have a temporal component

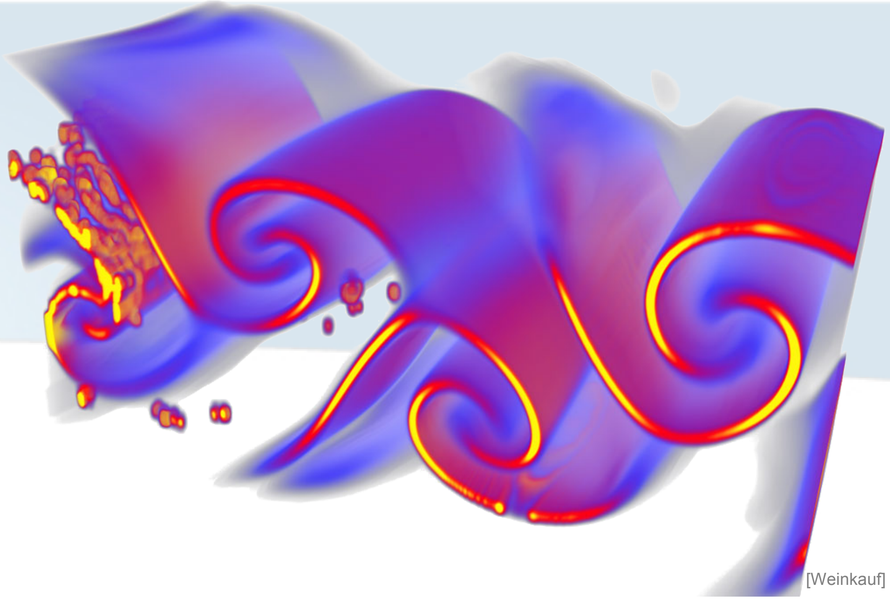


[Fout]

- Simulations often come uncertainty evaluation

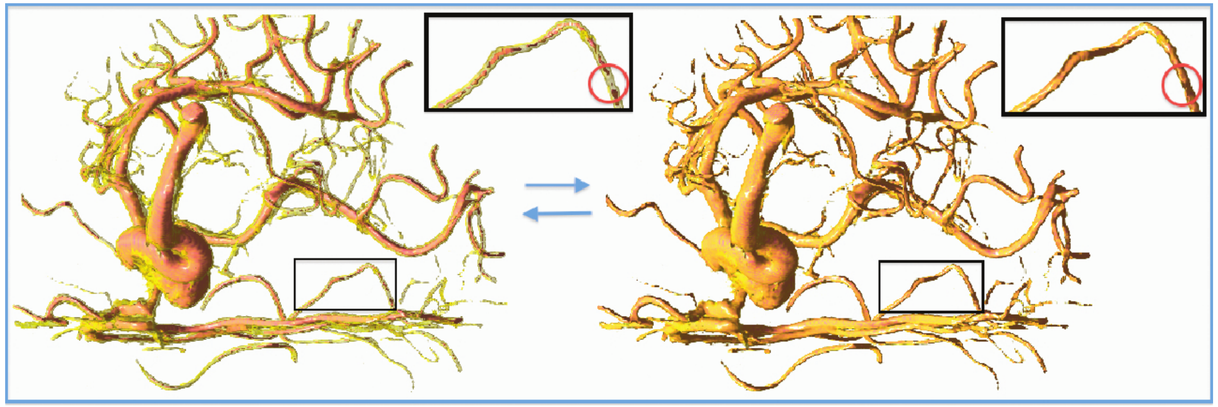
- Simulations often yield several fields per data-set

The awful truth



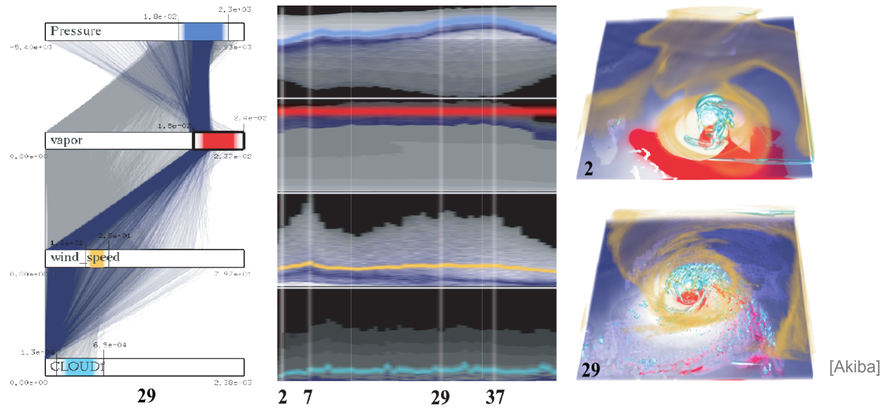
[Weinkauf]

- Simulations usually have a temporal component



[Fout]

- Simulations often come uncertainty evaluation



[Akiba]

- Simulations often yield several fields per data-set

References

- tierny@telecom-paristech.fr

- Projects, internships, Ph.D. thesis

- Online class
  - http://www.telecom-paristech.fr/~tierny/visualizationClass.html