Introduction to Scientific Visualization

Télécom ParisTech

tierny@telecom-paristech.fr
What?

- Generation of intelligible and interactive graphical representations
• 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
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
What?

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

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

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

- Scientific data-sets
- Results of simulations
  - Chemistry, Physics
  - Computational fluid dynamics
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
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/](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.
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
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

The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'eureka!' but 'that's funny...'

- Isaac Asimov

[Image of Facebook post and other graphics related to science and technology]
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
• 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
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
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?

- 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?

\[ f : D \rightarrow A \]
How does it work?

- $f : \mathcal{D} \rightarrow \mathbb{R}$
- Scalar field visualization
How does it work?

\[ f : \mathcal{D} \rightarrow \mathbb{A} \]

- \( f : \mathcal{D} \rightarrow \mathbb{R} \)
  - Scalar field visualization
- \( f : \mathcal{D} \rightarrow \mathcal{T}(\mathcal{D}) \)
  - Vector field visualization
How does it work?

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

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

- $f : \mathcal{D} \rightarrow \mathcal{M}_{d \times d}$
  - Tensor field visualization
How does it work?

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

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

- $f : \mathcal{D} \rightarrow \mathcal{M}_{d \times d}$
  - Tensor field visualization
How does it work?

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - 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?

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - 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?

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

- $f : D \rightarrow T(D)$
  - Vector field visualization

- $f : D \rightarrow M_{d \times d}$
  - Tensor field visualization
How does it work?

- \( f : \mathcal{D} \rightarrow \mathbb{R} \)
  - 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?

\[ f : \mathcal{D} \rightarrow \mathbb{A} \]

- \( f : \mathcal{D} \rightarrow \mathbb{R} \)
  - 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?

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - 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?

- \( f : \mathcal{D} \rightarrow \mathbb{R} \)
  - 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

\[ f : \mathcal{D} \rightarrow \mathcal{A} \]
How does it work?

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - 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?

- $f : D \rightarrow \mathbb{R}$
  - Scalar field visualization
- $f : D \rightarrow T(D)$
  - Vector field visualization
- $f : D \rightarrow M_{d \times d}$
  - Tensor field visualization
How does it work?

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

- \( f : \mathcal{D} \rightarrow \mathcal{T} (\mathcal{D}) \)
  - Vector field visualization

- \( f : \mathcal{D} \rightarrow \mathcal{M}_{d \times d} \)
  - Tensor field visualization
How does it work?

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

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

- $f : \mathcal{D} \rightarrow \mathcal{M}_{d \times d}$
  - Tensor field visualization

[http://www.neuroimaging.tau.ac.il/](http://www.neuroimaging.tau.ac.il/)
How does it work?

- $f : \mathcal{D} \rightarrow \mathbb{R}$
  - 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
Challenges

• These are complex data-sets:
Challenges

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

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

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

- 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
What do we mean by domain?

- *Continuous discrete* representation of a space ☒
What do we mean by domain?

- Continuous discrete representation of a space
- Finite set of samples
What do we mean by domain?

- Continuous discrete representation of a space ✗
- Finite set of samples
  - Positional information in an embedding space ☑
What do we mean by domain?

• *Continuous discrete* representation of a space  
• Finite set of samples
  • Positional information in an embedding space
  • Attributes
What do we mean by domain?

- Continuous discrete representation of a space \( X \)
- 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 \( X \)
• Finite set of samples
  • Positional information in an embedding space \( \mathbb{R} \)
  • Attributes
• Connectivity of the samples
What do we mean by domain?

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

- Domain
- Embedding space
- Cellular elements
Example

- Domain
  - 2-manifold $S$
- Embedding space
- Cellular elements
Example

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

- Domain
  - 2-manifold $S$
- 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

- Dimension?
- Embedding space?
- Boundary?
Manifold examples

• Dimension?
• Embedding space?
• Boundary?

\[ [0, 1] \]
Manifold examples

\[ \mathbb{R} \]

\[ [0, 1] \]

\[ \{ p \in \mathbb{R}^2 \mid \sqrt{p_x^2 + p_y^2} = r \} \]

- Dimension?
- Embedding space?
- Boundary?
Manifold examples

\[ \mathbb{R}^2 \]

\[ [0, 1] \]

\[ \{ p \in \mathbb{R}^2 \mid \sqrt{p_x^2 + p_y^2} = r \} \]

- Dimension?
- Embedding space?
- Boundary?
Manifold examples

$\mathbb{R}^2$

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

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

- Dimension?
- Embedding space?
- Boundary?
Manifold examples

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

\[ \{ p \in \mathbb{R}^2 \mid \sqrt{p_x^2 + p_y^2} \leq r \} \]
Manifold examples

\[ \mathbb{R}^3 \]

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

\[ \{ p \in \mathbb{R}^2 \mid \sqrt{p_x^2 + p_y^2} \leq r \} \]
Manifold examples

\[ \mathbb{R}^3 \]

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

\[ \{ p \in \mathbb{R}^2 \mid \sqrt{p_x^2 + p_y^2} \leq r \} \]

- Dimension?
- Embedding space?
- Boundary?
Manifold examples

$$\mathbb{R}^3$$

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

$$\{ p \in \mathbb{R}^3 \mid \sqrt{p^2_x + p^2_y + p^2_z} = r \}$$
Manifold examples

\[ \mathbb{R}^3 \]

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

\[ \{ p \in \mathbb{R}^3 \mid \sqrt{p_x^2 + p_y^2 + p_z^2} \leq r \} \]
Manifold examples

\[ \mathbb{R}^3 \]

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

\( \{ p \in \mathbb{R}^3 \mid \sqrt{p_x^2 + p_y^2 + p_z^2} \leq r \} \)

- Dimension?
- Embedding space?
- Boundary?
Manifold examples

- Dimension?
- Embedding space?
- Boundary?

\[ \mathbb{R}^3 \]

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

\[ \{ p \in \mathbb{R}^3 \mid \sqrt{p_x^2 + p_y^2 + p_z^2} \leq r \} \]
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 $G$
  • 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 $[0, k_1] \times [0, k_2] \times [0, k_3]$?
  - $k_1 \cdot k_2 \cdot k_3$
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 $[0, k_1] \times [0, k_2] \times [0, k_3]$?
  - $k_1 \cdot k_2 \cdot k_3$
- 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 $[0, k_1] \times [0, k_2] \times [0, k_3]$? $k_1 \cdot k_2 \cdot k_3$
- How many samples? $(k_1 + 1) \cdot (k_2 + 1) \cdot (k_3 + 1)$
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 $\mathbb{R}^2$

\[ f_\parallel : ]x_1, x_2[ \times ]y_1, y_2[ \rightarrow \mathbb{R} \]
Interpolants for regular grids

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

• Examples
Interpolants for regular grids

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

- $f_\Pi : [x_1, x_2[ \times [y_1, y_2[ \to \mathbb{R}$

Examples
- Piecewise constant
- Nearest neighbor
Interpolants for regular grids

• For regular grids of \( \mathbb{R}^2 \)
  
  \[ f_\Pi : [x_1, x_2] \times [y_1, y_2] \rightarrow \mathbb{R} \]

• Examples
  • Piecewise constant
  • Nearest neighbor
    – Value of the nearest vertex
Interpolants for regular grids

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

• Examples
  
  • Piecewise constant
  • Nearest neighbor
    – Value of the nearest vertex
Interpolants for regular grids

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

- Examples
  - Bilinear interpolation
    - One dimension at a time
Interpolants for regular grids

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

- Examples
  - Bilinear interpolation
  - One dimension at a time

\[ f_\parallel(x, y_1) = \frac{x-x_1}{x_2-x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1) \]
Interpolants for regular grids

- For regular grids of $\mathbb{R}^2$
  \[ f_{\|} : [x_1, x_2] \times [y_1, y_2] \rightarrow \mathbb{R} \]
- Examples
  - Bilinear interpolation
    - One dimension at a time
    \[
    f_{\|}(x, y_1) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1) \\
    f_{\|}(x, y_2) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)
    \]
Interpolants for regular grids

• For regular grids of $\mathbb{R}^2$

  \[ f_\Pi : [x_1, x_2] \times [y_1, y_2] \to \mathbb{R} \]

• Examples

  • Bilinear interpolation
    – One dimension at a time

  \[
  f_\Pi(x, y_1) = \frac{x-x_1}{x_2-x_1} (f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)
  \]

  \[
  f_\Pi(x, y_2) = \frac{x-x_1}{x_2-x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)
  \]

  \[
  f_\Pi(x, y) = \frac{y-y_1}{y_2-y_1} (f(x, y_1) - f(x, y_2)) + f(x, y_1)
  \]
Interpolants for regular grids

• For regular grids of $\mathbb{R}^2$
  \[ f_\square : ]x_1, x_2[ \times ]y_1, y_2[ \to \mathbb{R} \]

• Examples
  • Bilinear interpolation
    – One dimension at a time

\[
 f_{\square}(x, y_1) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)
\]

\[
 f_{\square}(x, y_2) = \frac{x - x_1}{x_2 - x_1}(f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)
\]

\[
 f_{\square}(x, y) = \frac{y - y_1}{y_2 - y_1}(f(x, y_1) - f(x, y_2)) + f(x, y_1)
\]
Interpolants for regular grids

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

- Examples
  - Bilinear interpolation
    - One dimension at a time
    \[
    f_{\|}(x, y_1) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)
    \]
    \[
    f_{\|}(x, y_2) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)
    \]
    \[
    f_{\|}(x, y) = \frac{y - y_1}{y_2 - y_1} (f(x, y_1) - f(x, y_2)) + f(x, y_1)
    \]
Interpolants for regular grids

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

  \[ f_{\parallel}: [x_1, x_2] \times [y_1, y_2] \rightarrow \mathbb{R} \]

- Examples

  - Bilinear interpolation
  
  - One dimension at a time

  \[
  f_{\parallel}(x, y_1) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1)
  \]

  \[
  f_{\parallel}(x, y_2) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)
  \]

  \[
  f_{\parallel}(x, y) = \frac{y - y_1}{y_2 - y_1} (f(x, y_1) - f(x, y_2)) + f(x, y_1)
  \]
Interpolants for regular grids

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

- Examples
  - Bilinear interpolation
    - One dimension at a time

\[
\begin{align*}
f_\Pi(x, y_1) &= \frac{x - x_1}{x_2 - x_1} (f(x_2, y_1) - f(x_1, y_1)) + f(x_1, y_1) \\
f_\Pi(x, y_2) &= \frac{x - x_1}{x_2 - x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2) \\
f_\Pi(x, y) &= \frac{y - y_1}{y_2 - y_1} (f(x, y_1) - f(x, y_2)) + f(x, y_1)
\end{align*}
\]
Interpolants for regular grids

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

\[ f^\parallel(x, y_2) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2) \]

\[ f^\parallel(x, y) = \frac{y - y_1}{y_2 - y_1} (f(x, y_1) - f(x, y_2)) + f(x, y_1) \]
Interpolants for regular grids

• For regular grids of $\mathbb{R}^2$

- Problem
  - Critical points in the interior

\[
\begin{align*}
  f_{\|}(x, y_2) &= \frac{x - x_1}{x_2 - x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2) \\
  f_{\|}(x, y) &= \frac{y - y_1}{y_2 - y_1} (f(x, y_1) - f(x, y_2)) + f(x, y_1)
\end{align*}
\]
Interpolants for regular grids

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

  - Problem
    - Critical points in the interior
    - Source of many ambiguities in visualization

\[
f(x, y) = \frac{y - y_1}{y_2 - y_1} (f(x, y_1) - f(x, y_2)) + f(x, y_1)
\]

\[
f(x, y_2) = \frac{x - x_1}{x_2 - x_1} (f(x_2, y_2) - f(x_1, y_2)) + f(x_1, y_2)
\]
Interpolants for regular grids

• For regular grids of $\mathbb{R}^2$
  $f_{\parallel} : [x_1, x_2] \times [y_1, y_2] \to \mathbb{R}$

• Examples
  • Bicubic interpolation
Interpolants for regular grids

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

• Examples
  • Bicubic interpolation
    – Piecewise polynomial
Interpolants for regular grids

• For regular grids of $\mathbb{R}^2$

$$f_{\Pi} : [x_1, x_2] \times [y_1, y_2] \rightarrow \mathbb{R}$$

• Examples
  • Bicubic interpolation
    – Piecewise polynomial
    – Smoother interpolant
Interpolants for regular grids

• For regular grids of $\mathbb{R}^2$
  
  \[
  f_{\Pi} : \quad \prod \{x_1, x_2\} \times \prod \{y_1, y_2\} \rightarrow \mathbb{R}
  \]

• Examples
  
  • Bicubic interpolation
    – Piecewise polynomial
    – Smoother interpolant
    – Takes adjacent cells into account
Interpolants for regular grids

• For regular grids of $\mathbb{R}^2$

  $f_{II} : \times [x_1, x_2] [y_1, y_2] \rightarrow \mathbb{R}$

• Examples

  • Bicubic interpolation
    – Piecewise polynomial
    – Smoother interpolant
    – Takes adjacent cells into account
Interpolants for regular grids

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

• For regular grids of $\mathbb{R}^3$

  \[ f_{\Pi} : [x_1, x_2] \times [y_1, y_2] \times [z_1, z_2] \rightarrow \mathbb{R} \]
Interpolants for regular grids

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

• Examples
  • Piecewise constant
    – Nearest neighbor
  • Trilinear
  • Tricubic
Interpolants for regular grids

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

• Examples
  • Piecewise constant
    – Nearest neighbor
  • Trilinear
  • Tricubic
Interpolants for regular grids

• For regular grids of $\mathbb{R}^3$
  \[ f_3 : [x_1, x_2] \times [y_1, y_2] \times [z_1, z_2] \rightarrow \mathbb{R} \]

• Examples
  • Piecewise constant
    – Nearest neighbor
  • Trilinear
  • Tricubic
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 $\sigma$ 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 $\sigma$ 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 $\sigma$ 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 $\sigma$
Manifolds on a computer

• Notion of simplex
• A $d$-simplex $\sigma$ 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 $\sigma$
• Smallest combinatorial construction to represent a cell of dimension $d$
Manifolds on a computer

• Notion of simplex
  • A d-simplex $\sigma$ 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 $\sigma$ 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 $\sigma$ 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 \( \sigma \) 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 $\sigma$ 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 $\tau$ of a $d$-simplex $\sigma$ is the simplex defined by a non-empty subset of the $d+1$ points of $\sigma$
  • Noted $\tau \leq \sigma$
Manifolds on a computer

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

• Recursive construction!
Manifolds on a computer

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

• Notion of face
  • A face $\tau$ of a $d$-simplex $\sigma$ is the simplex defined by a non-empty subset of the $d+1$ points of $\sigma$
  • 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 $\tau$ of a $d$-simplex $\sigma$ is the simplex defined by a non-empty subset of the $d+1$ points of $\sigma$
  • 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 $\tau$ of a $d$-simplex $\sigma$ is the simplex defined by a non-empty subset of the $d+1$ points of $\sigma$
  • 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 $\tau$ of a $d$-simplex $\sigma$ is the simplex defined by a non-empty subset of the $d+1$ points of $\sigma$
  • 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 $\tau$ of a simplex $\sigma_i$ 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 $\tau$ of a simplex $\sigma_i$ is also in $\mathcal{K}$
    - Any two simplices $\sigma_i$ and $\sigma_j$ 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 $\tau$ of a simplex $\sigma_i$ is also in $\mathcal{K}$
  • Any two simplices $\sigma_i$ and $\sigma_j$ 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 $\tau$ of a simplex $\sigma_i$ is also in $\mathcal{K}$
    • Any two simplices $\sigma_i$ and $\sigma_j$ 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 \( \tau \) of a simplex \( \sigma_i \) is also in \( \mathcal{K} \)
    - Any two simplices \( \sigma_i \) and \( \sigma_j \) 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 \( \tau \) of a simplex \( \sigma_i \) is also in \( \mathcal{K} \)
    - Any two simplices \( \sigma_i \) and \( \sigma_j \) 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 $\tau$ of a simplex $\sigma_i$ is also in $\mathcal{K}$
    • Any two simplices $\sigma_i$ and $\sigma_j$ 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 $\tau$ of a simplex $\sigma_i$ is also in $\mathcal{K}$
    - Any two simplices $\sigma_i$ and $\sigma_j$ 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 $\mathcal{M}$ is a simplicial complex $\mathcal{K}$ such that
Manifolds on a computer

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

Diagram: A triangle with vertices labeled v1, v2, and v3.
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 $f : \mathcal{T} \to \mathbb{R}$
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
  - \( f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \} \)
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 \( \nabla f \) vanishes
Scalar field geometry

- Continuous color maps
- Difficulty to estimate the geometry
  - Locally (gradient)
  - Globally (level sets)
- Level sets
  \[ f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \} \]
- Critical points
  - Where \( \nabla f \) vanishes
Notion of critical point
Notion of critical point

• Combinatorial identification
Notion of critical point

• Combinatorial identification
• Star of a simplex $\sigma$
Notion of critical point

- Combinatorial identification
- Star of a simplex $\sigma$
  - Simplices that contain $\sigma$ as a face
Notion of critical point

- Combinatorial identification
- Star of a simplex $\sigma$
  - Simplices that contain $\sigma$ as a face
- Link of a simplex
Notion of critical point

• Combinatorial identification
  • Star of a simplex $\sigma$
    • Simplices that contain $\sigma$ as a face
  • Link of a simplex
    • Simplices of the closure of the star that do not contain $\sigma$ as a face
Notion of critical point

• Combinatorial identification
• Lower link of $\nu$
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
  - $f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \}$
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 $D$ 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
  • $f^{-1}(i) = \{p \in \mathcal{D} \mid f(p) = i\}$
    • 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
  - $f^{-1}(i) = \{p \in \mathcal{D} \mid f(p) = i\}$
    - 0-manifold
    - If intersection, 3 cases only
Level set in a 1-simplex

• Let $D$ be a single edge
  • $f(v_0)$
  • $f(v_1)$
  • $f(v_0) < f(v_1)$

• Let $i$ be an isovalue
  • $f^{-1}(i) = \{ p \in D \mid f(p) = i \}$
    • 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
  - $f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \}$
    - 0-manifold
    - If intersection, 3 cases only
Level set in a 1-simplex

• Let $D$ be a single edge
  • $f(v_0)$
  • $f(v_1)$
  • $f(v_0) < f(v_1)$

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

- Let $i$ be an isovalue
  - $f^{-1}(i) = \{ p \in D \mid f(p) = i \}$
    - 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 $D$ 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 iso-value
  • $f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \}$
Level set in a 3-simplex

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

- $f(v_0), f(v_1), f(v_2), f(v_3)$

Let $i$ be an iso-value

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

  - 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 $D$ be a single tetrahedron
  • $f(v_0), f(v_1), f(v_2), f(v_3)$

• Let $i$ be an iso-value
  • $f^{-1}(i) = \{p \in D / f(p) = i\}$
    • 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
  - $f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \}$
    - 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 iso-value
  - $f^{-1}(i) = \{p \in \mathcal{D} / f(p) = i\}$
    - 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 $D$ be a single tetrahedron
  - $f(v_0), f(v_1), f(v_2), f(v_3)$
- Let $i$ be an isovalue
  - $f^{-1}(i) = \{ p \in D \mid f(p) = i \}$
  - 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
  • $f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \}$
    • 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 $D$ be a single d-simplex
  - $f(v_0), f(v_1), \ldots f(v_d)$
- Let $i$ be an isovalue
  - $f^{-1}(i) = \{ p \in D \mid f(p) = i \}$
  - **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
  - $f^{-1}(i) = \{ p \in \mathcal{D} \mid f(p) = i \}$
  - 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
  - Apply 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 $D$ be a 2-regular grid
Marching squares

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

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

- Level set extraction
  - Loop over the unit cells
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
Marching squares

- Let $D$ be a 2-regular grid
  - With bilinear interpolant

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

• Let $D$ 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
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!
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
Issues of marching squares
Issues of marching squares
Issues of marching squares
Issues of marching squares
Issues of marching squares
Issues of marching squares

- Geometrically inaccurate (lines)
Issues of marching squares

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

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

- Let $D$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells
Marching cubes

- Let $D$ be a 3-regular grid
  - With trilinear interpolant
- Level set extraction
  - Loop over the unit cells
  - Cases on a 3D unit cell
Marching cubes

- Let $D$ be a 3-regular grid
  - With trilinear interpolant

- Level set extraction
  - Loop over the unit cells
  - Cases on a 3D unit cell
    - 256 cases
Issues of marching cubes

• Same as before
Issues of marching cubes

- Same as before
- But worse!
Issues of marching cubes

• Same as before
  • But worse!
• Lorensen and Cline 1987
Issues of marching cubes

• Same as before
  • But worse!
• Lorensen and Cline 1987
  • 16 cases (symmetries)
Issues of marching cubes

• Same as before
  • But worse!
• Lorensen and Cline 1987
  • 16 cases (symmetries)
  • Possible cracks
Issues of marching cubes

- Same as before
  - But worse!
- Lorensen and Cline 1987
  - 16 cases (symmetries)
  - Possible cracks
  - Improvement 28 cases
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):
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
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
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):
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):
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):
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):
In practice
In practice
In practice
Scalars
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 $\mathbf{f}(v)$
  • 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
What is there to visualize?
• The simple way: glyphs
• Data overload
• Occlusion issues
• Small scale details
What is there to visualize?

• Getting inspiration from engineering sciences
What is there to visualize?

• Getting inspiration from engineering sciences
What is there to visualize?

- Getting inspiration from engineering sciences
- Localized visualization
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
What is there to visualize?

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

- Analogy to scalar fields:
  - Isocontours
  - Isosurfaces
Streamlines

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

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

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

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

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

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

• Solution to an ODE
  • $c : C \to [0, 1]$
Streamlines

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

• Solution to an ODE
  • $c : \mathcal{C} \to [0, 1]$ 
  • $p = c^{-1}(0) + \int_0^{c(p)} \frac{f(c^{-1}(u))}{du}$ 
  • $\forall p \in \mathcal{C}$
  • Unique solution
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 order approximations
Streamlines on a computer

• Numerical integration of the ODE
  • Euler method
  • Runge-Kutta
    • Higher order approximations
Streamlines on a computer

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

- Numerical integration of the ODE
  - Euler method
  - Runge-Kutta
    - Higher order approximations
- Discretization
  - \( c : \mathcal{C} \rightarrow [0, 1] \)
  - \( p = c^{-1}(0) + \int_0^{c(p)} \mathbf{f}(c^{-1}(u)) du \)
  - \( p = c^{-1}(0) + \lim_{q \to \infty} \sum_{u \in [0, q]} \frac{\mathbf{f}(c^{-1}(u \cdot \frac{c(p)}{q+1}))}{\| \mathbf{f}(c^{-1}(u \cdot \frac{c(p)}{q+1})) \|} \cdot \frac{c(p)}{q+1}, \quad u \in \mathbb{N} \)
Streamlines on a computer

- Numerical integration of the ODE
  - Euler method
  - Runge-Kutta
    - Higher order approximations
- Discretization
  - \( c : \mathcal{C} \to [0, 1] \)
  - \( p = c^{-1}(0) + \int_0^c(p) \vec{f}(c^{-1}(u))du \)
  - \( p = c^{-1}(0) + \lim_{q \to \infty} \sum_{u \in [0, q]} \frac{\vec{f}(c^{-1}(u, \frac{c(p)}{q+1})))}{\| \vec{f}(c^{-1}(u, \frac{c(p)}{q+1}))) \|} \cdot \frac{c(p)}{q + 1} \), \( u \in \mathbb{N} \)
Streamlines on a computer

• Numerical integration of the ODE
  • Euler method
  • Runge-Kutta
    • Higher oder approximations
• Discretization
  • $c : C \to [0, 1]$
  • $p = c^{-1}(0) + \int_{0}^{c(p)} \vec{f}(c^{-1}(u)) \, du$
  • $p = c^{-1}(0) + \lim_{q \to \infty} \sum_{u \in [0, q]} \frac{\vec{f}(c^{-1}(u, \frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u, \frac{c(p)}{q+1}))||} \cdot \frac{c(p)}{q + 1}, \quad u \in \mathbb{N}$
Streamlines on a computer

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

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

- Numerical integration of the ODE
  - Euler method
  - Runge-Kutta
    - Higher order approximations
- Discretization
  - $c : C \rightarrow [0, 1]$
  - $p = c^{-1}(0) + \int_0^{c(p)} \vec{f}(c^{-1}(u))du$
  - $p = c^{-1}(0) + \lim_{q \to \infty} \sum_{u \in [0, q]} \frac{\vec{f}(c^{-1}(u \cdot \frac{c(p)}{q+1}))}{||\vec{f}(c^{-1}(u \cdot \frac{c(p)}{q+1}))||} \cdot \frac{c(p)}{q+1}, \quad u \in \mathbb{N}$
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 $\vec{f}(c^{-1}(u \cdot c(p)/q)$
Streamlines on a computer

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

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

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

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

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

• Euler integration algorithm
  • Given a seed point $c^{-1}(0)$
  • $u \leftarrow 0$
  • Repeat
    • Evaluate $\mathbf{f}(c^{-1}(u \cdot c(p)/q))$
    • $c^{-1}((u + 1) \cdot c(p)/q) \leftarrow c^{-1}(u \cdot c(p)/q) + \frac{\mathbf{f}(c^{-1}(u \cdot c(p)/q))}{\|\mathbf{f}(c^{-1}(u \cdot c(p)/q))\|} \cdot \frac{c(p)}{q}$
    • $u \leftarrow u + 1$
  • Until $u \leq q$
Streamlines on a computer

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

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

Repeat
- Evaluate \( \vec{f}(c^{-1}(u \cdot c(p)/q)) \)
- \( c^{-1}((u + 1) \cdot c(p)/q) \leftarrow c^{-1}(u \cdot c(p)/q) + \frac{\vec{f}(c^{-1}(u \cdot c(p)/q))}{||\vec{f}(c^{-1}(u \cdot c(p)/q))||} \cdot \frac{c(p)}{q} \)
- \( u \leftarrow u + 1 \)

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
Streamlines on a computer

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

[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
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
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
Line Integral Convolution

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

• Algorithm
  • Compute a noise field $\mathcal{N} : \mathcal{D} \to [0, 1]$
    • Random intensity for each vertex
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
Line Integral Convolution

- Algorithm
  - Compute a noise field $\mathcal{N} : \mathcal{D} \to [0, 1]$
    - 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
Line Integral Convolution

• Algorithm
  • Compute a noise field $\mathbf{N} : \mathcal{D} \rightarrow [0, 1]$
    • 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
      $$LIC(p) = \int_{c(p)-L}^{c(p)+L} k(u) \cdot \mathbf{N}(c^{-1}(u))du$$
Line Integral Convolution

• Convolution kernel $k(u)$
Line Integral Convolution

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

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

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

\[\int_{c(p) - L}^{c(p) + L} k(u) \, du = 1\]
Line Integral Convolution
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
LIC for volumetric domains

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

- Also,
  - What is LIC in the end?
LIC for volumetric domains

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

- Also,
  - What is LIC in the end?
  - A scalar field
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
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
Global visualization of the direction of the flow
Line Integral Convolution

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

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

- Combine the magnitude with the LIC
Derived scalar fields

- Combine the magnitude with the LIC
- Color map of the magnitude
Derived scalar fields

• Combine the magnitude with the LIC
• Color map of the magnitude
• For each channel (RGB)
  • Multiply by the LIC value
Derived scalar fields

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

- 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

• 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

• 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

• 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

- 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

• 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

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

- Divergence of the flow, for instance
  \[ f : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \]
Flow orientation

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

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

\[
div f = \nabla \cdot f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}
\]
Flow orientation

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

\[
div f = \nabla \cdot f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}
\]
Angular speed
Angular speed

• Magnitude of the curl, for instance
  \[ f : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \]
Angular speed

- Magnitude of the curl, for instance

\[ f : \mathbb{R}^3 \to \mathbb{R}^3 \]

\[ \nabla \times f = \]

\[ \left( \frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z} \right) \hat{i} + \left( \frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x} \right) \hat{j} \]

\[ + \left( \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right) \hat{k} \]
Angular speed

• Magnitude of the curl, for instance

\[ f : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \]

\[ \nabla \times f = \]

\[ \left( \frac{\partial f_y}{\partial y} - \frac{\partial f_z}{\partial z} \right) \hat{i} + \left( \frac{\partial f_z}{\partial z} - \frac{\partial f_x}{\partial x} \right) \hat{j} \]

\[ + \left( \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right) \hat{k} \]
Angular speed

• Magnitude of the curl, for instance

\[ f : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \]

\[ \nabla \times f = \]

\[ \left( \frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z} \right)\hat{i} + \left( \frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x} \right)\hat{j} + \left( \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right)\hat{k} \]
Angular speed

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

\[
\mathbf{\nabla \times f} = \\
\left( \frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z} \right) \mathbf{i} + \\
\left( \frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x} \right) \mathbf{j} + \\
\left( \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right) \mathbf{k}
\]

- On PL-manifolds?
So far

• Extract geometrical features
  • Streamlines
So far

• Extract geometrical features
• Streamlines
So far

• Extract geometrical features
  • Streamlines
  • Seeding

(Jobard et al. 97)
So far

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

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

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

- Extract geometrical measures
  - Magnitude, orientation, angular speed, distortion
So far

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

Gradient field topology
Gradient field topology

• For instance
  • \( \mathcal{D} \subset \mathbb{R}^n \)
  • \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \)
Gradient field topology

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

Gradient field, example
- $g : \mathbb{R}^n \to \mathbb{R}$
  - $f = \nabla g = \left( \frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \ldots, \frac{\partial g}{\partial x_n} \right)$
Gradient field topology

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

Gradient field, example
- $g : \mathbb{R}^n \rightarrow \mathbb{R}$
- $f = \nabla g = \left( \frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \ldots, \frac{\partial g}{\partial x_n} \right)$

PL scalar fields
- Piecewise constant gradient field
Gradient field topology

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

- Gradient field, example
  - $g : \mathbb{R}^n \rightarrow \mathbb{R}$
    - $f = \nabla g = \left( \frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \ldots, \frac{\partial g}{\partial x_n} \right)$

- 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 stop?
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 $g$
Gradient field topology

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

• Where does the gradient stop?
  • 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
Gradient field topology

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

- What's the relation to streamlines?
Gradient field topology

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

• What's the relation to streamlines?
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
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
Gradient field topology

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

- Understanding the structure of the critical points
- Several streamlines can have the same extremities
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
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
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
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?
Gradient field topology

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

• For example
  • \( f : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \)
Critical points of a vector field

• For example
  • \( f : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \)

• 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
  - $f(p) = p$

[Diagrams of source, sink, and saddle points]
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
Classifying critical points

Source
Sink
Saddle

Center
Classifying critical points

- Jacobian of the vector field
Classifying critical points

- Jacobian of the vector field

\[
J = \begin{bmatrix}
\frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\
\frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y}
\end{bmatrix}
\]
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
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  \quad Sink  \quad Saddle  \quad Center
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
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
    - \( R_1, I_1 \) - \( R_2, I_2 \)
Classifying critical points

- Jacobian of the vector field
  \[ J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix} \]
  \[ R_1 > 0, R_2 > 0 \]
  \[ I_1 = 0, I_2 = 0 \]

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - Symmetric, antisymmetric
    - \( R_1, I_1 \) - \( R_2, I_2 \)

Source
Sink
Saddle
Center

Chen
Classifying critical points

- Jacobian of the vector field
  \[ J = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{bmatrix} \]
  - \( R_1 > 0, R_2 > 0 \)
  - \( I_1 = 0, I_2 = 0 \)
  - \( R_1 < 0, R_2 < 0 \)
  - \( I_1 = 0, I_2 = 0 \)

- Eigenvalues of J
  - 2 eigenvalues
  - For each
    - Real and imaginary parts
      - Symmetric, antisymmetric
    - \( R_1, I_1 \) - \( R_2, I_2 \)
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
    - \( R_1, I_1 \) - \( R_2, I_2 \)
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
    - \( R_1, I_1 \) - \( R_2, I_2 \)
Classifying critical points

• Jacobian of the vector field

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

- \( R_1 > 0, R_2 > 0 \)  \( I_1 = 0, I_2 = 0 \)  \( I_1 \neq 0, I_2 \neq 0 \)  \( R_1 < 0, R_2 < 0 \)  \( I_1 = 0, I_2 = 0 \)  \( I_1 \neq 0, I_2 \neq 0 \)  \( R_1 R_2 < 0 \)  \( I_1 = 0, I_2 = 0 \)

• Eigenvalues of J
  • 2 eigenvalues
  • For each
    • Real and imaginary parts
      – Symmetric, antisymmetric
    • \( R_1, I_1 \) - \( R_2, I_2 \)
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
    - \( R_1, I_1 \) - \( R_2, I_2 \)
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
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
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!
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
Non planar domains

• PL 2-manifolds in $\mathbb{R}^3$
  • Trivial extension
  • Numerical evaluations slightly more involved
• Volumetric domains
  • Similar process
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
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
Vector fields
Tensor Field Visualization
Notion of tensor
Notion of tensor

• Generalization of scalars and vectors
Notion of tensor

- Generalization of scalars and vectors
- Describes linear relations between
Notion of tensor

• Generalization of scalars and vectors
• Describes linear relations between
  • Scalars
  • Vectors
  • Or other tensors
Notion of tensor

- Generalization of scalars and vectors
- Describes linear relations between
  - Scalars
  - Vectors
  - Or other tensors
- Multi-dimensional array of numerical values
Notion of tensor

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

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

- Order of a tensor
Notion of tensor

• Order of a tensor
  • Scalar value: $0^{\text{th}}$ order tensor
Notion of tensor

- Order of a tensor
  - Scalar value: 0\textsuperscript{th} order tensor
  - Vector field: 1\textsuperscript{st} order tensor
Notion of tensor

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

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

- Here, mostly symmetric 2\textsuperscript{nd} order tensors
In practice

• Given a domain $\mathcal{D}$
In practice

- Given a domain $\mathcal{D}$
In practice

• Given a domain $D$
In practice

• Given a domain $D$
• For each vertex $v$
In practice

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

• Given a domain \( D \)
• For each vertex \( v \)
• One matrix \( f(v) \)
In practice

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

\[
\begin{bmatrix}
a_0 & b_0 \\
c_0 & d_0
\end{bmatrix}
\]
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 $f(v)$
  - $(d \times d)$-matrix
  - $d$: dimension of $\mathcal{D}$
In practice

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

• Given a domain $\mathcal{D}$
• For each vertex $v$
• One matrix $f(v)$
  • $(d \times d)$-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 $f(v)$
  • $(d \times d)$-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
    • $f(v)_{ij} = f(v)_{ji}$, $\forall i, j$

• 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

\[
U \quad f(v) \quad U^T = \begin{bmatrix}
\lambda_1 & 0 & 0 \\
0 & \lambda_2 & 0 \\
0 & 0 & \lambda_3
\end{bmatrix}
\]
Tensor diagonalization

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

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

- Let's represent $f(v)$ by an ellipsoid
  - Semi-principal axes
    - Eigenvectors
  - Axis length: eigenvalue
  - Interpolation of the eigenvectors/values
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
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”
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?
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
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
    - \( f(v)_{ij} = f(v)_{ji}, \quad \forall i, j \)

- It can be diagonalized
  - \( \lambda_1, \lambda_2, \lambda_3 \)
    - Eigenvalues
  - \( U = (\vec{e}_1, \vec{e}_2, \vec{e}_3) \)
    - Eigenvectors

\[
U \cdot f(v) \cdot U^T = \begin{bmatrix}
\lambda_1 & 0 & 0 \\
0 & \lambda_2 & 0 \\
0 & 0 & \lambda_3 \\
\end{bmatrix}
\]
Tensor glyphs

• Ellipsoids
Tensor glyphs

• Ellipsoids
  • Semi-principal axes
  • Eigen vectors
Tensor glyphs

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

• Ellipsoids
  • Semi-principal axes
    • Eigen vectors
      – Direction, not a vector!
  • Axis length
    • Eigen values
Tensor glyphs

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

• Anisotropy information
Tensor glyphs

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

- Anisotropy information
Tensor glyphs

- Ellipsoids in 3D
Tensor glyphs

• Ellipsoids in 3D
Tensor glyphs

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

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

- Ellipsoids in 3D
- Information loss due to screen projection
- Continuous shading
  - Hard to evaluate variations across axes
• “Superquadrics”
• Parallelepiped with smooth edges
• Exaggerate shading variations across axis
Glyph packing

- How to distribute the glyphs
Glyph packing

• How to distribute the glyphs
• To avoid data overload
• Glyph overlap
Glyph packing

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

- Particle-based energy optimization
Glyph packing

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

• Particle-based energy optimization
  • Given a target number of particles
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
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
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
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
Glyph packing
Putting it all together
Tensor glyphs

• Provide important insights
Tensor glyphs

- Provide important insights
- Direction information
Tensor glyphs

• Provide important insights
• Direction information
• Anisotropy information
Tensor glyphs

• Provide important insights
  • Direction information
  • Anisotropy information

• Still
  • Occlusion issues
Tensor glyphs

- Provide important insights
  - Direction information
  - Anisotropy information

- Still
  - Occlusion issues
  - No global information
Tensor glyphs

- Provide important insights
  - Direction information
  - Anisotropy information

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

• What interesting vector fields could we consider?
Derived vector fields

• What interesting vector fields could we consider?
  • Eigenvectors
Derived vector fields

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

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

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

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

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

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

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

• How to combine the directions?
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 …
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
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
Hyper-streamlines

• For each of the d directions
Hyper-streamlines

• For each of the d directions
  • Streamline integration
Hyper-streamlines

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

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

• Problem
Hyper-streamlines

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

• Problem
  • No orientation
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
Hyper-streamlines

- Other than that...
Hyper-streamlines

• Other than that...
  • Same integration as for vector fields
Hyper-streamlines

• Other than that...
• Same integration as for vector fields
Hyper-streamlines

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

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

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

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

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

• Getting inspiration from vector fields
  • Line Integral Convolution
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
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
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
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
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
Tensor fields
The awful truth
The awful truth

• Simulations usually have a temporal component
The awful truth

- Simulations usually have a temporal component
The awful truth

• Simulations usually have a temporal component

• Simulations often come uncertainty evaluation
The awful truth

• Simulations usually have a temporal component

• Simulations often come uncertainty evaluation
The awful truth

• Simulations usually have a temporal component

• Simulations often come uncertainty evaluation

• Simulations often yield several fields per data-set
The awful truth

• Simulations usually have a temporal component

• Simulations often come uncertainty evaluation

• 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