Scientific Visualization: Topology, Flow and Vectors

CS 6630, Fall 2015 — Alex Lex
Aaron Knoll, guest lecturer

Slides thanks to:
Joshua Levina, Clemson University
Guoning Chen, University of Texas at Houston
Gordon Kindlmann, University of Chicago
Robert Laramee, Swansea University
Christoph Garth, University of Kaiserslautern

www.sci.utah.edu

Recap from last sci-vis lecture

- Isosurfaces
  - implicit vs explicit surfaces
  - contours, isosurfaces and level sets
- Marching cubes
  - How it works: 15-case lookup table
  - Improvements to marching cubes, and particle isosurface extraction methods
  - What are strengths / limitations of these approaches?
- Direct isosurface visualization
  - splatting
  - ray casting
  - What are advantages of direct vs indirect approaches?
  - When are isosurfaces better than volume rendering, and visa versa?

Today

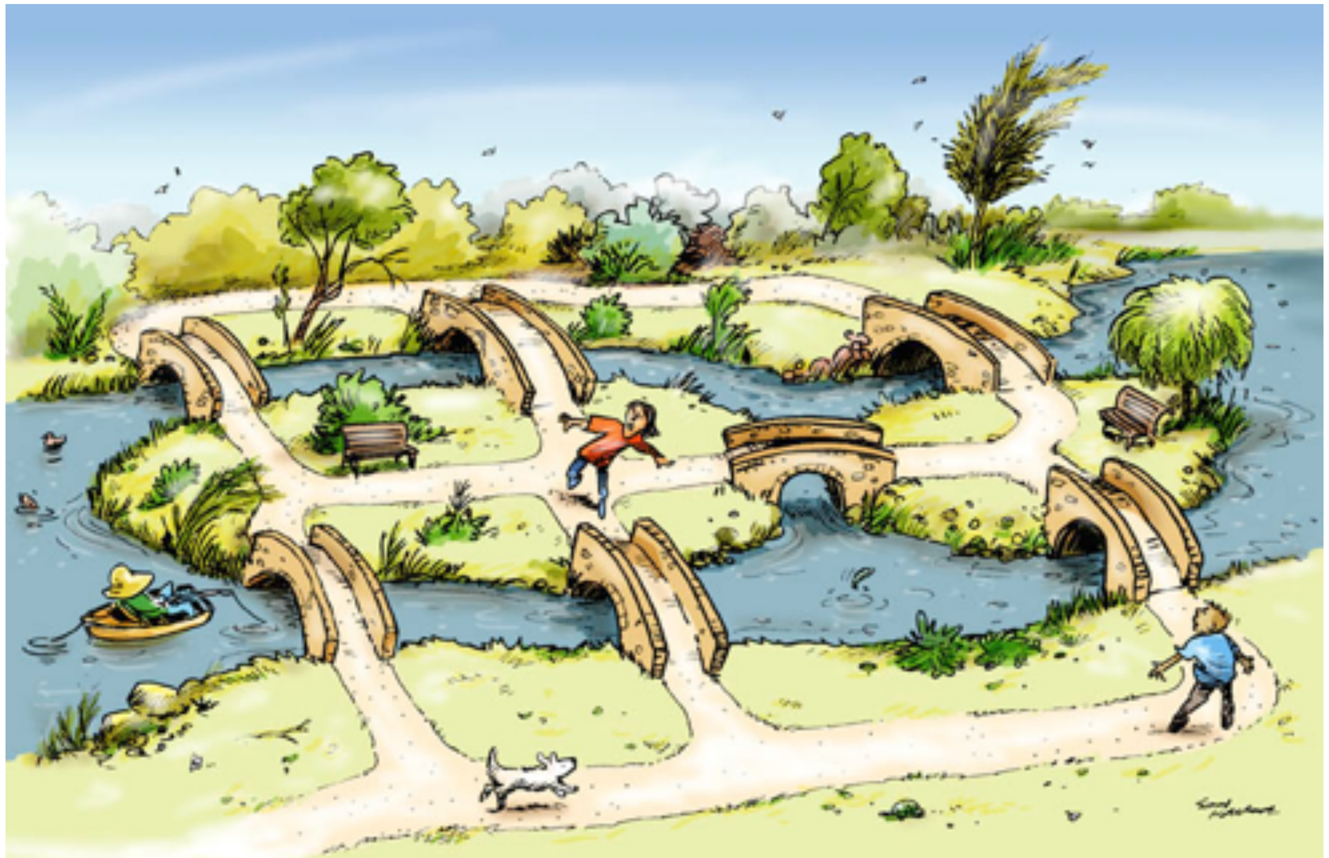
- *"Advanced Topics in Visualization" in one lecture!*
- Topology
  - Critical Points
  - Reeb Graphs and Contour Trees
  - Morse-Smale Complexes
- Flow and Vector Field Visualization
  - Fluid dynamics and a bit of math
  - Geometric methods: streamlines, streaklines, timelines, pathlines
  - Image-based methods: spot noise, LIC
  - Physically-based methods: Schlieren photography, Virtual Rheoscopic Fluids
  - Finite-Time Lyapunov Exponent

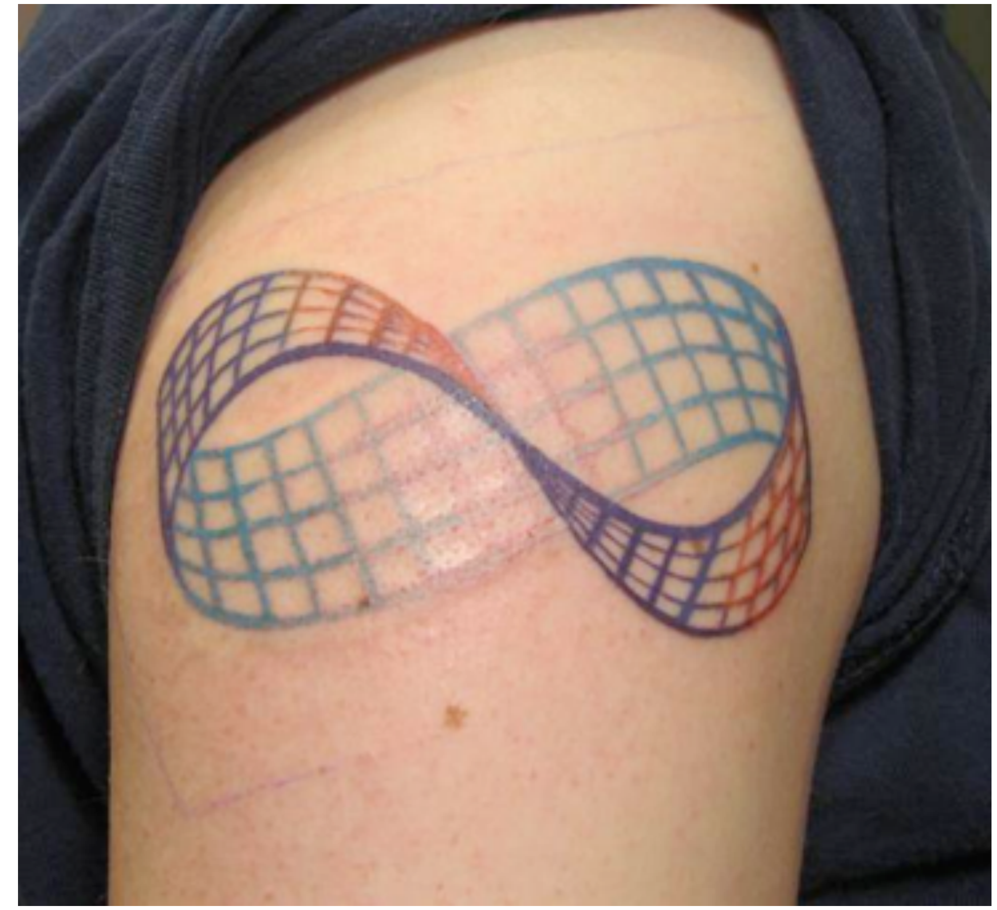
Topology

What is Topology?

- Field of mathematics which studies properties which are **preserved under continuous transformations**.
  - Stretching, bending = continuous changes.
  - Tearing, gluing = discontinuous changes.
- Also called: “Rubber sheet” geometry.
- Studies the connectedness of a space.

http://simonkneebone.files.wordpress.com/2011/11/konigsberg-puzzle.jpg



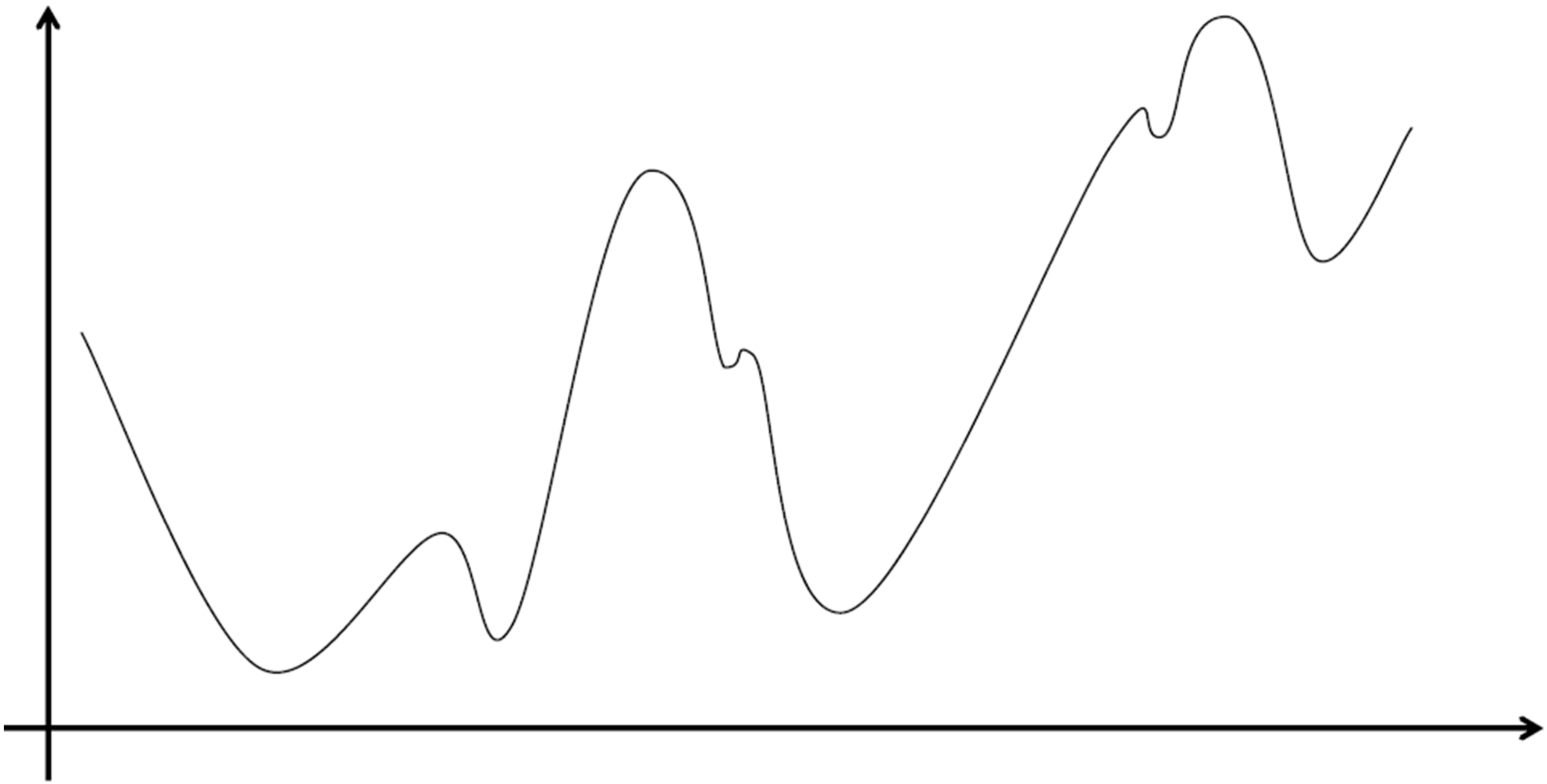


http://talklikeaphysicist.com/wp-content/uploads/2008/09/image-497.jpg

http://math.arizona.edu/~models/Topology/source/2.html

1D Case

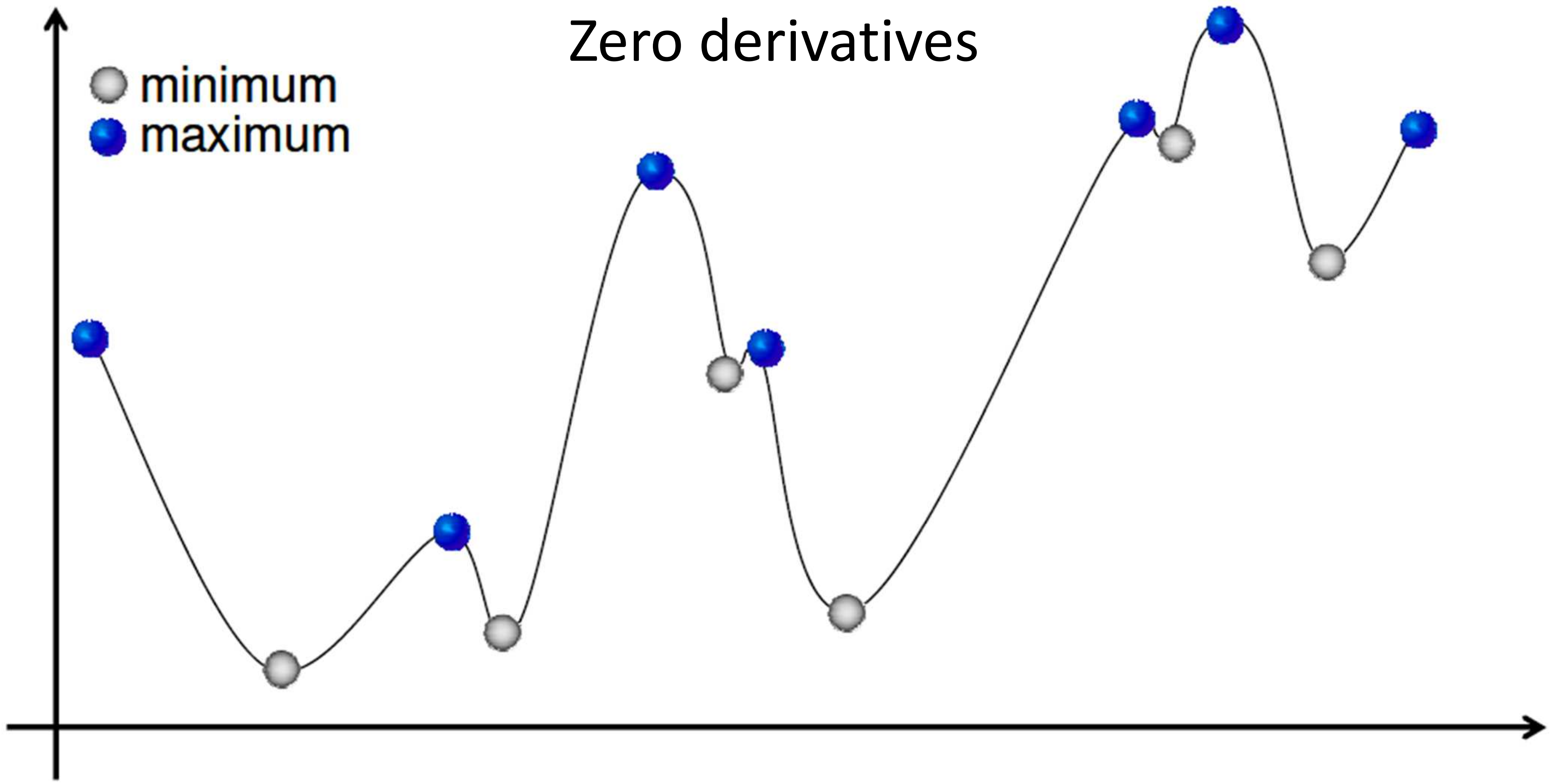
- Let us get back to the simple 1D case



1D Case

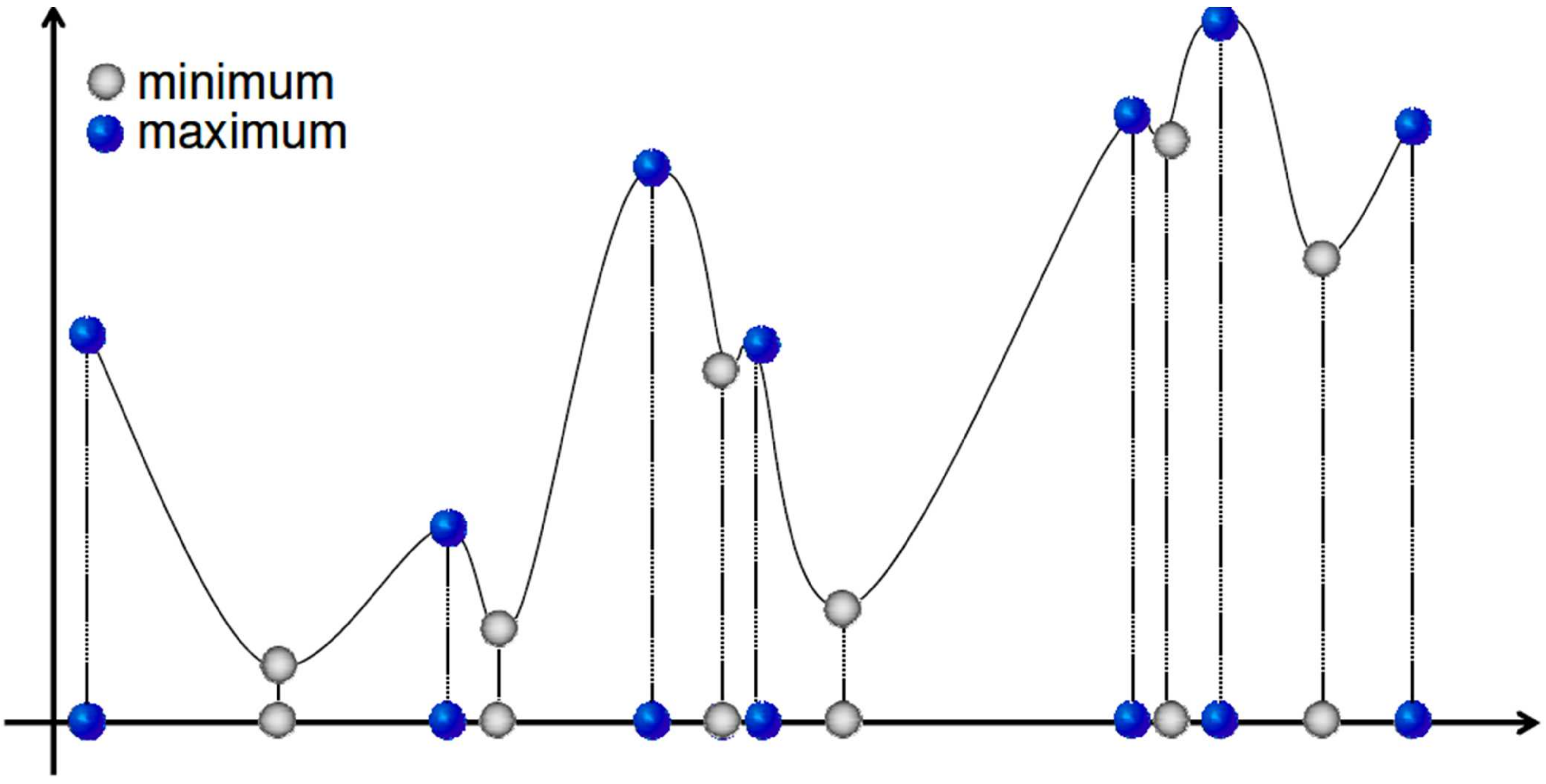
- Let us find out the local minimum/maximum

Zero derivatives



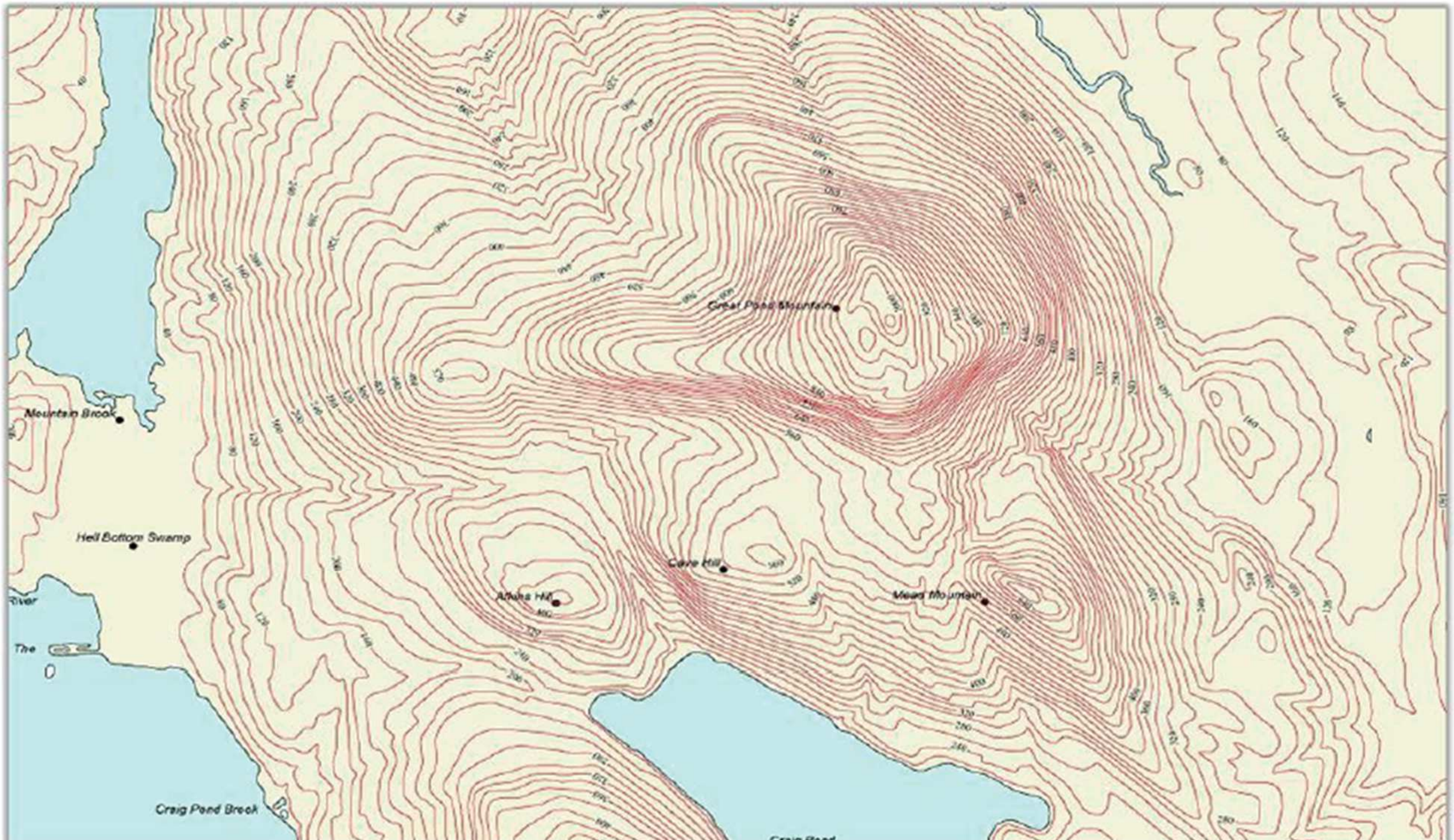
1D Case

- They partition the domain into monotonic regions



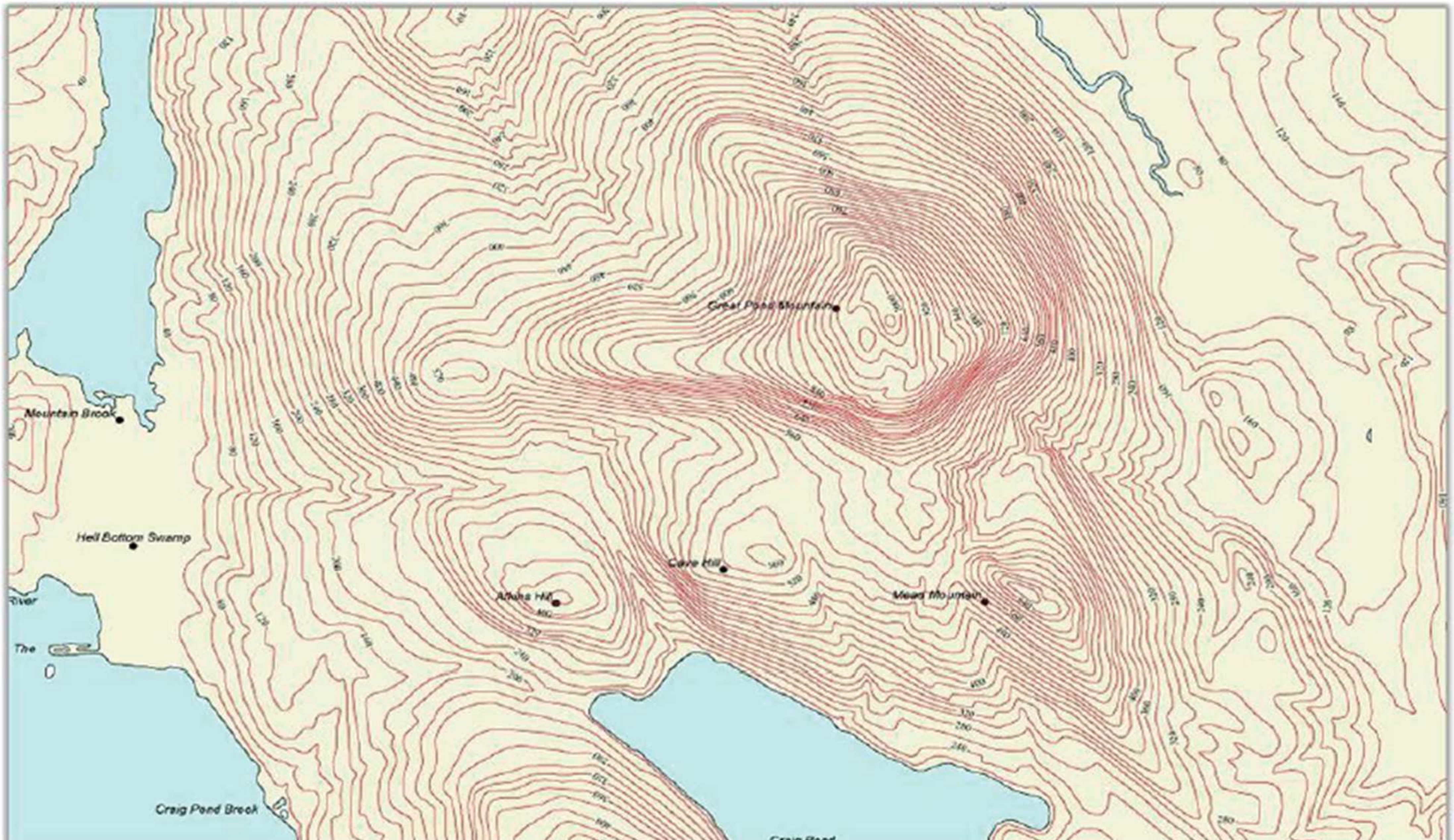
How About 2D Case?

Pre-image of an iso-value: Iso-contours



We Want to Extract Similar Information

Q: Which iso-contours are interesting?
Q: Summarize the evolution of iso-contours?



Topology

- These local minimum and maximum are called "critical points" of the scalar functions.
- Their connection forms the topology of the scalar field, which provides a partition scheme of the spatial domain.
- Each segment has the equivalent homogeneous behavior, e.g. monotonic for 1D case.
- This is similar for 2D and 3D scalar fields

Scalar Field Analysis

- Here is a more formal definition
- Given a scalar field f
  - Gradient vector

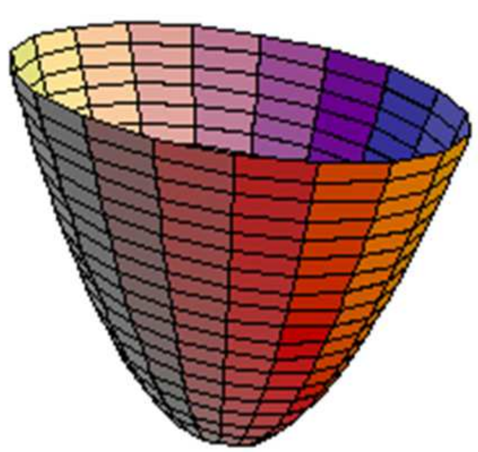
$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{bmatrix}$$

    - When not zero
      - Points in the direction of quickest ascend
      - Always perpendicular to the iso-contours (or level sets) of f
    - If $\nabla f\,(p)=0$,
      - p is a critical point
      - $f(p)$ is a critical value

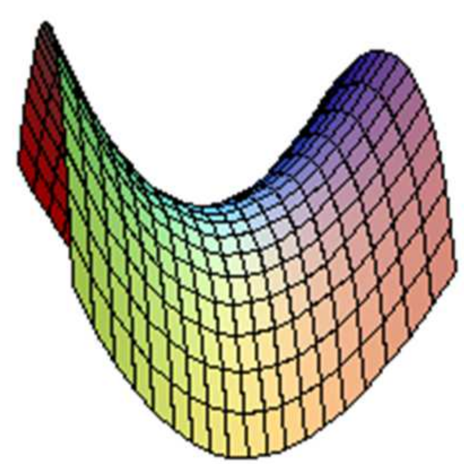
Scalar Field Analysis

- A critical point p is isolated if there exists a neighborhood of p such that p is the only critical point in the neighborhood

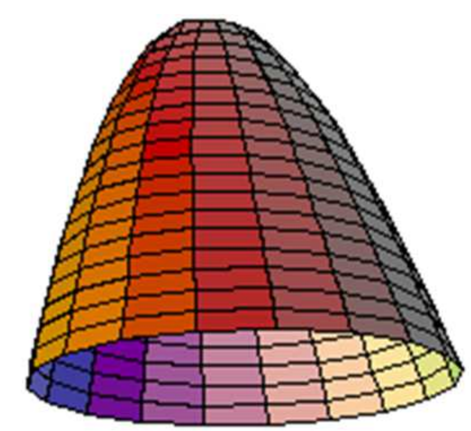
- Classification of fundamental critical points in 2D



Local minima



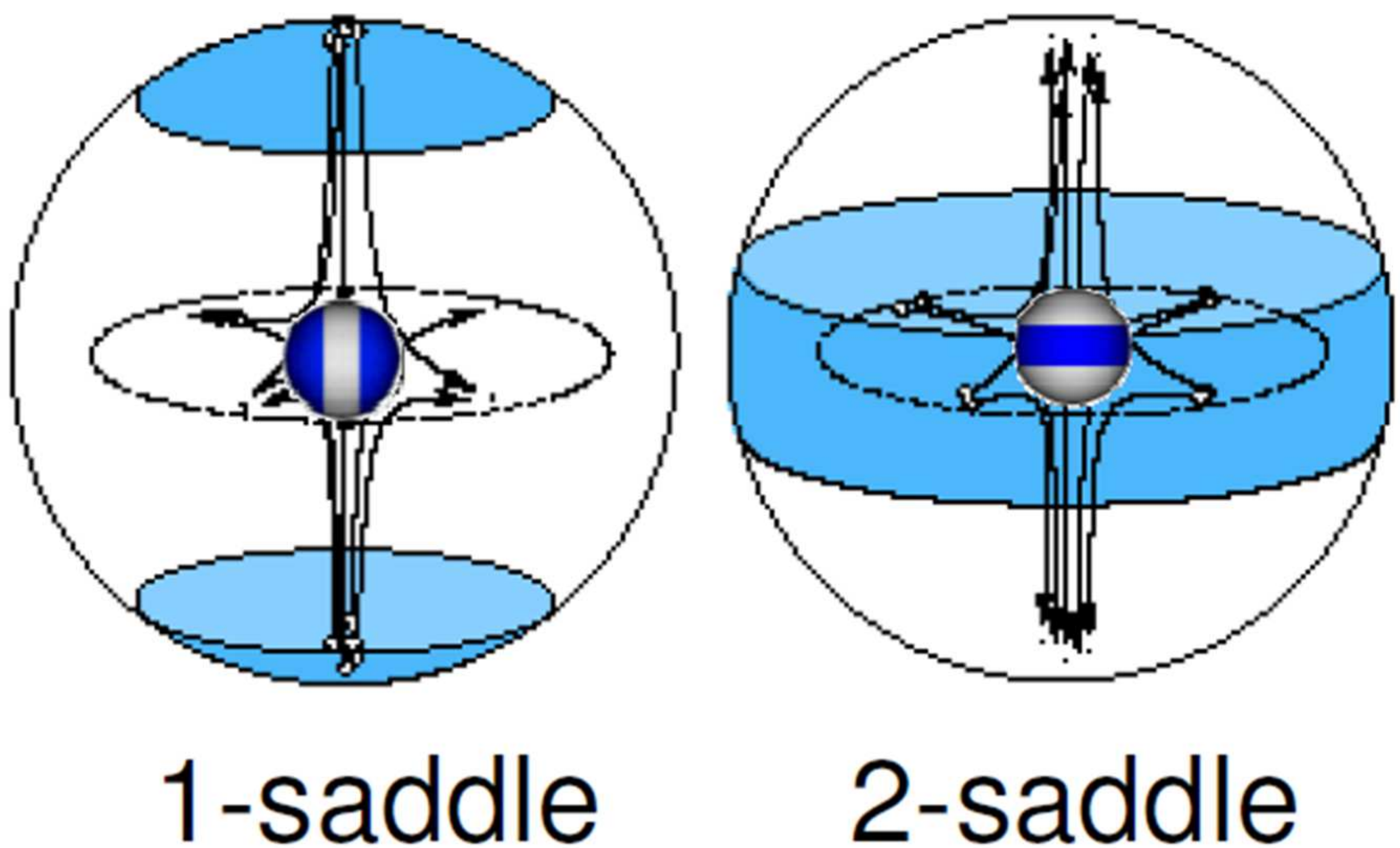
Saddle



Local maxima

Detection of Critical Points

3D saddles can have two distinct configurations



Scalar Field Analysis

- A function is a Morse function if it is smooth and all of its critical points are isolated and non-degenerate
  - Typically a good assumption for scientific data
  - A non-Morse function can be made Morse by adding small but random noise

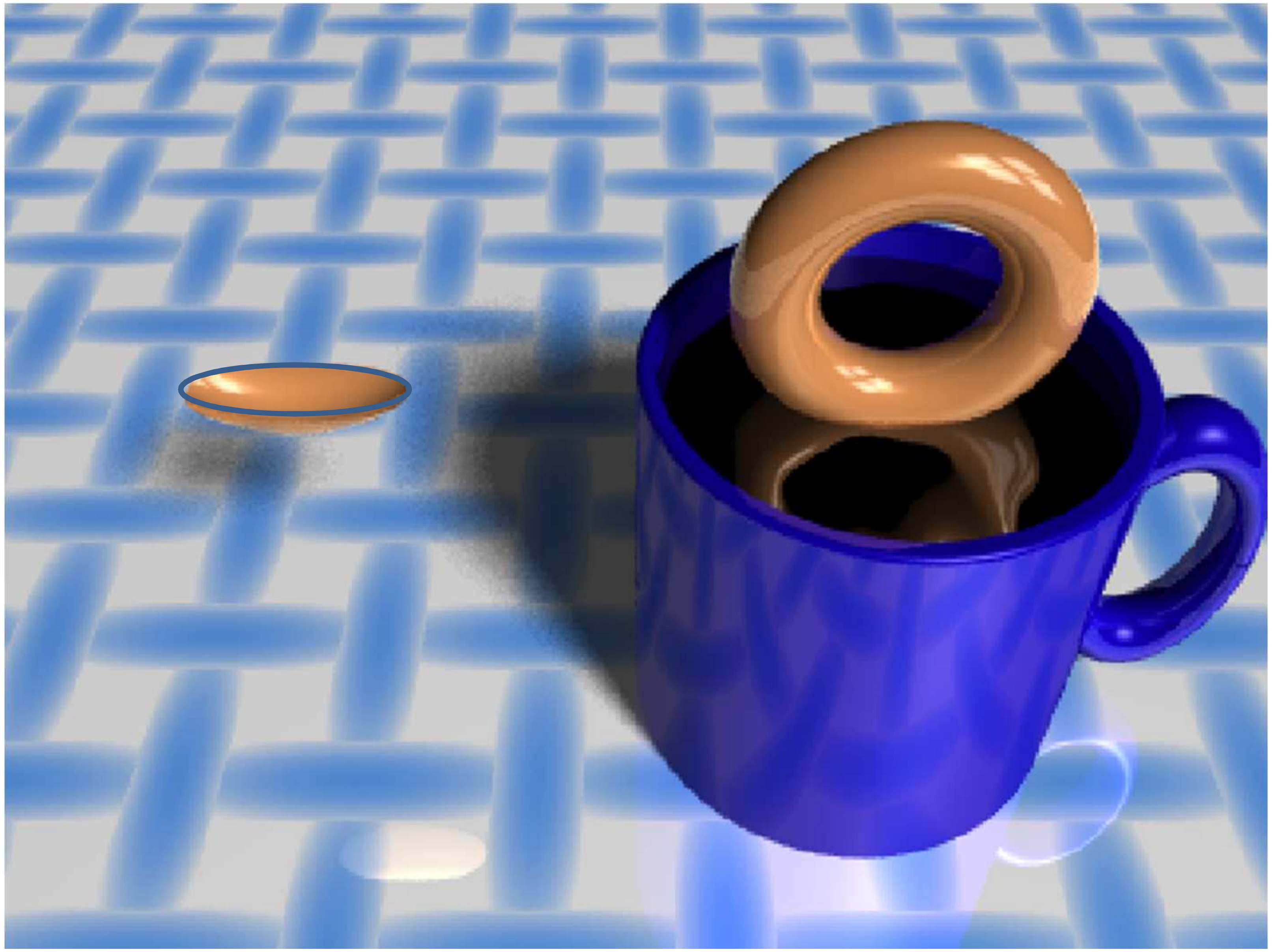
Level-Set Topology Reeb Graphs, Contour Trees, and Merge Trees

Example – dunking a doughnut

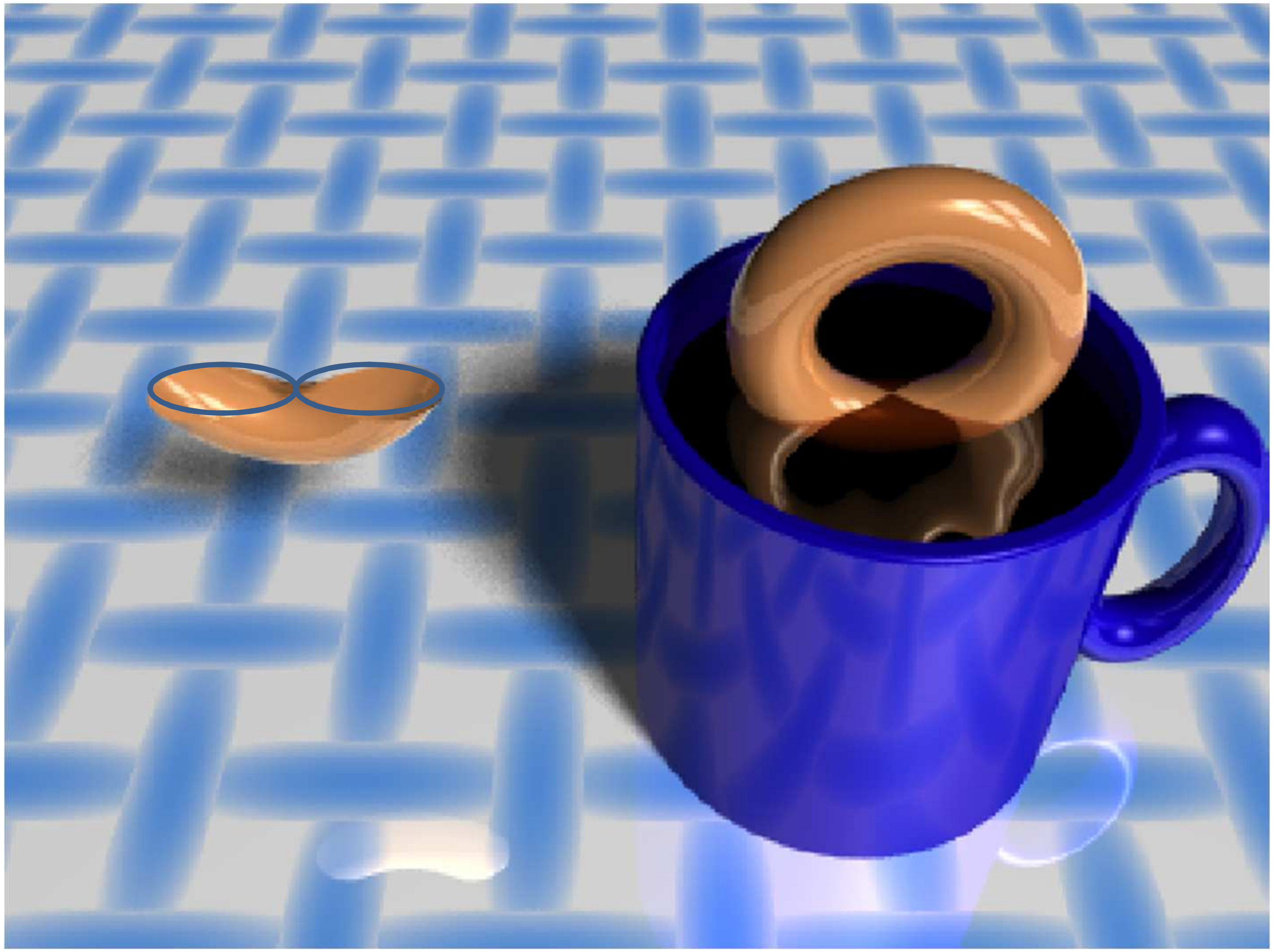
- $f(\mathbf{p}) = z$ (height function)

Shape analysis is a special case of scalar field analysis

Example – dunking a doughnut



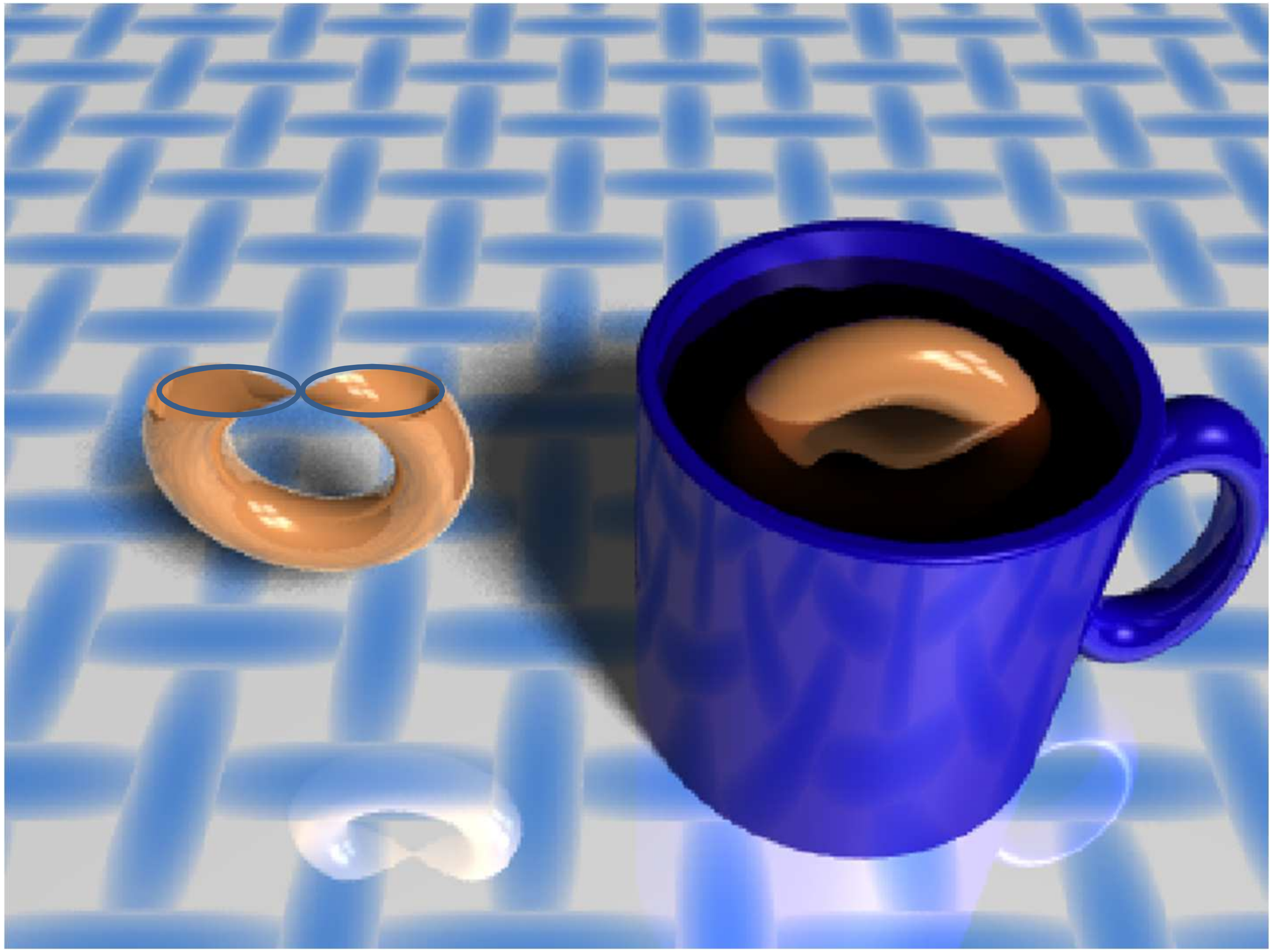
Example – dunking a doughnut



Example – dunking a doughnut

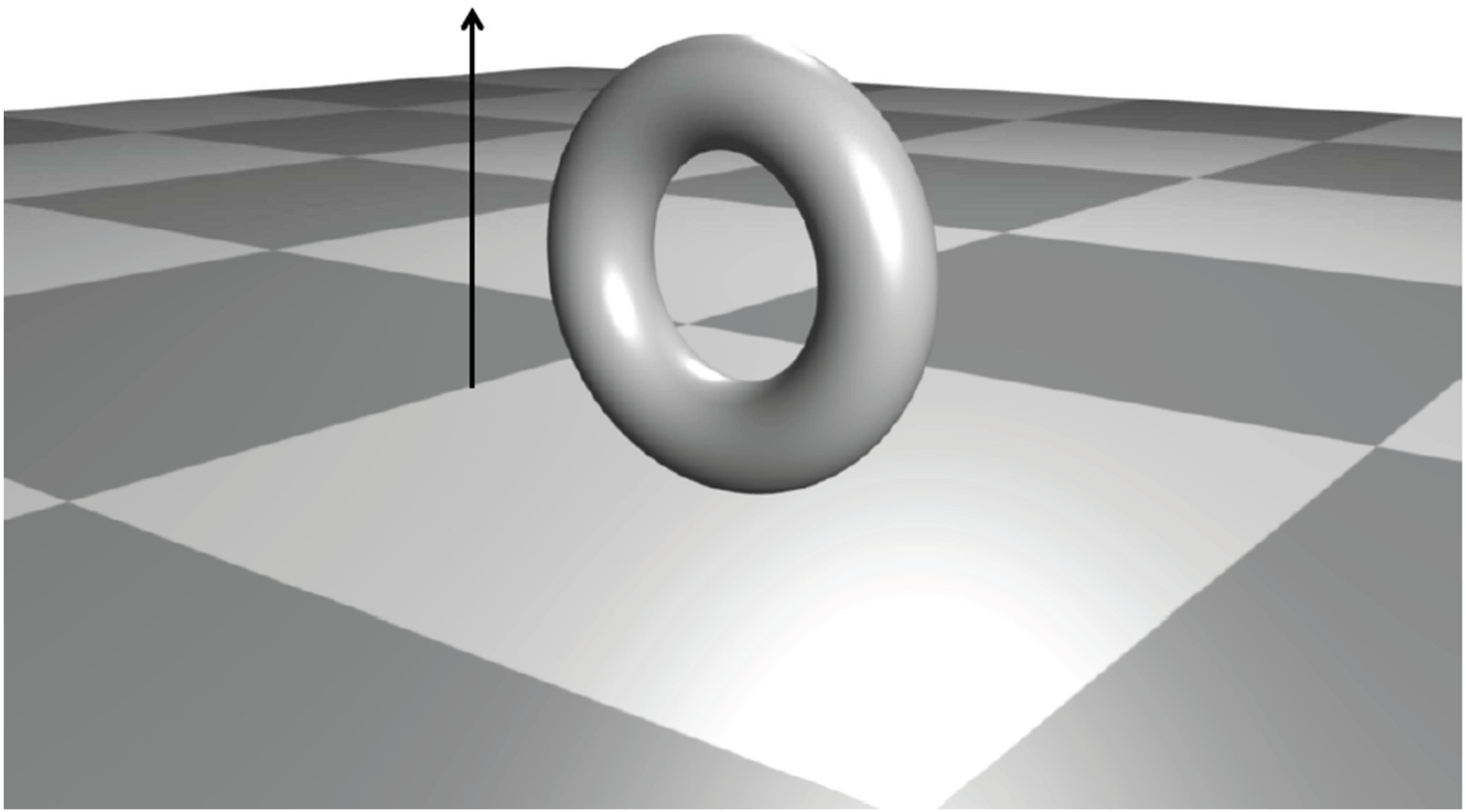
DUNKIN' DONUTS
AMERICA RUNS ON DUNKIN'

Example – dunking a doughnut



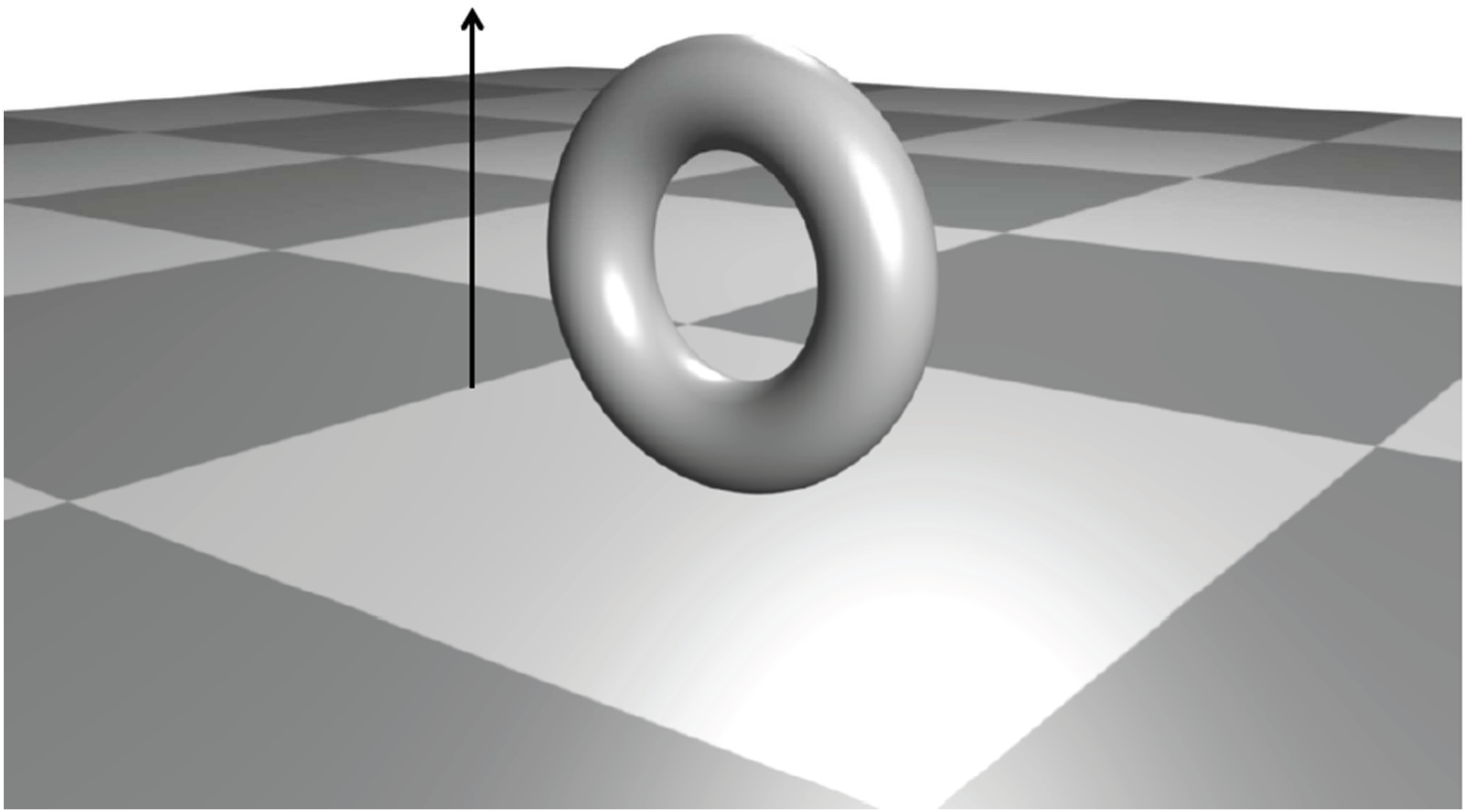
Example – dunking a doughnut

How Does it Work?

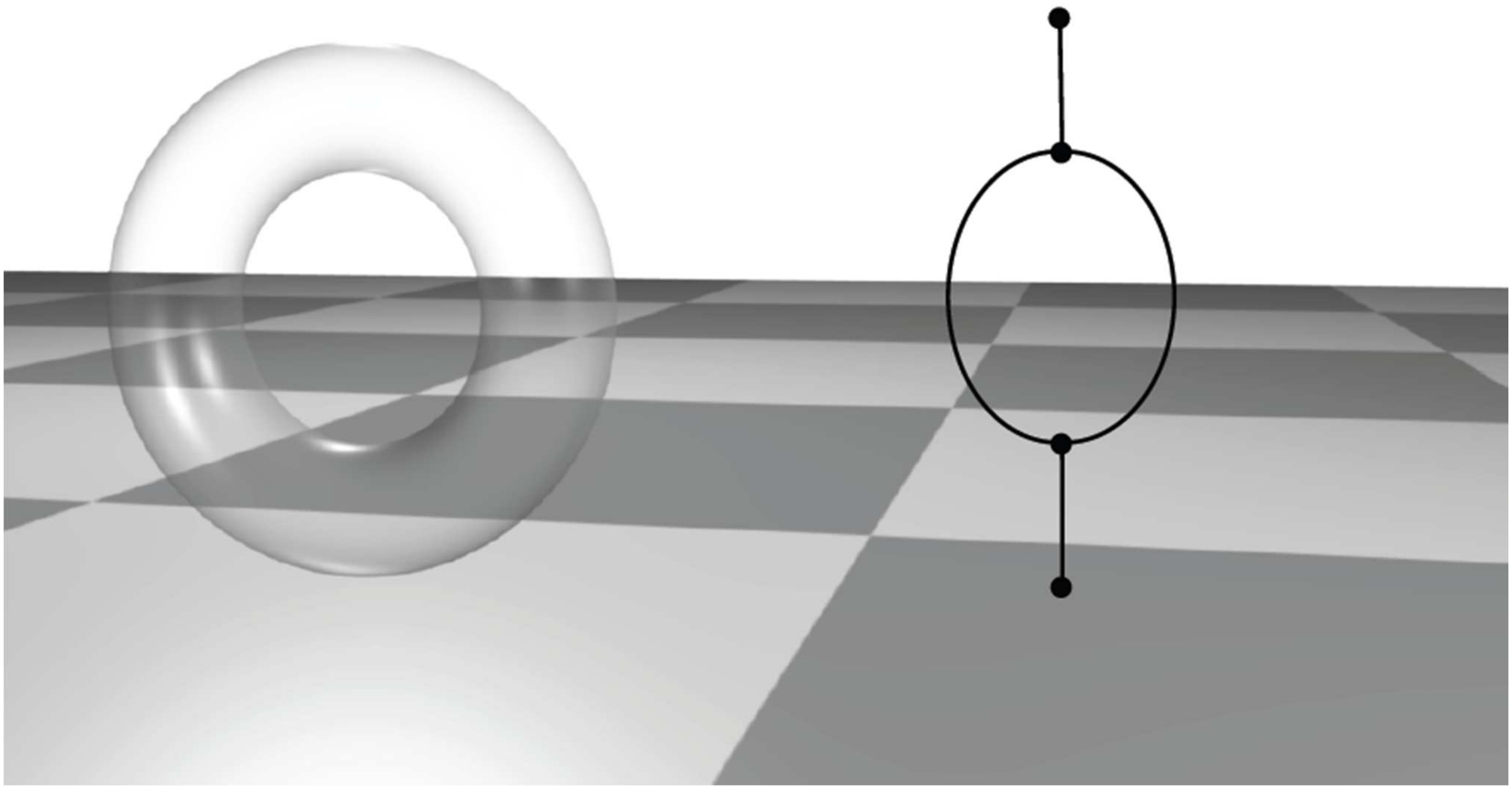


How Does it Work?

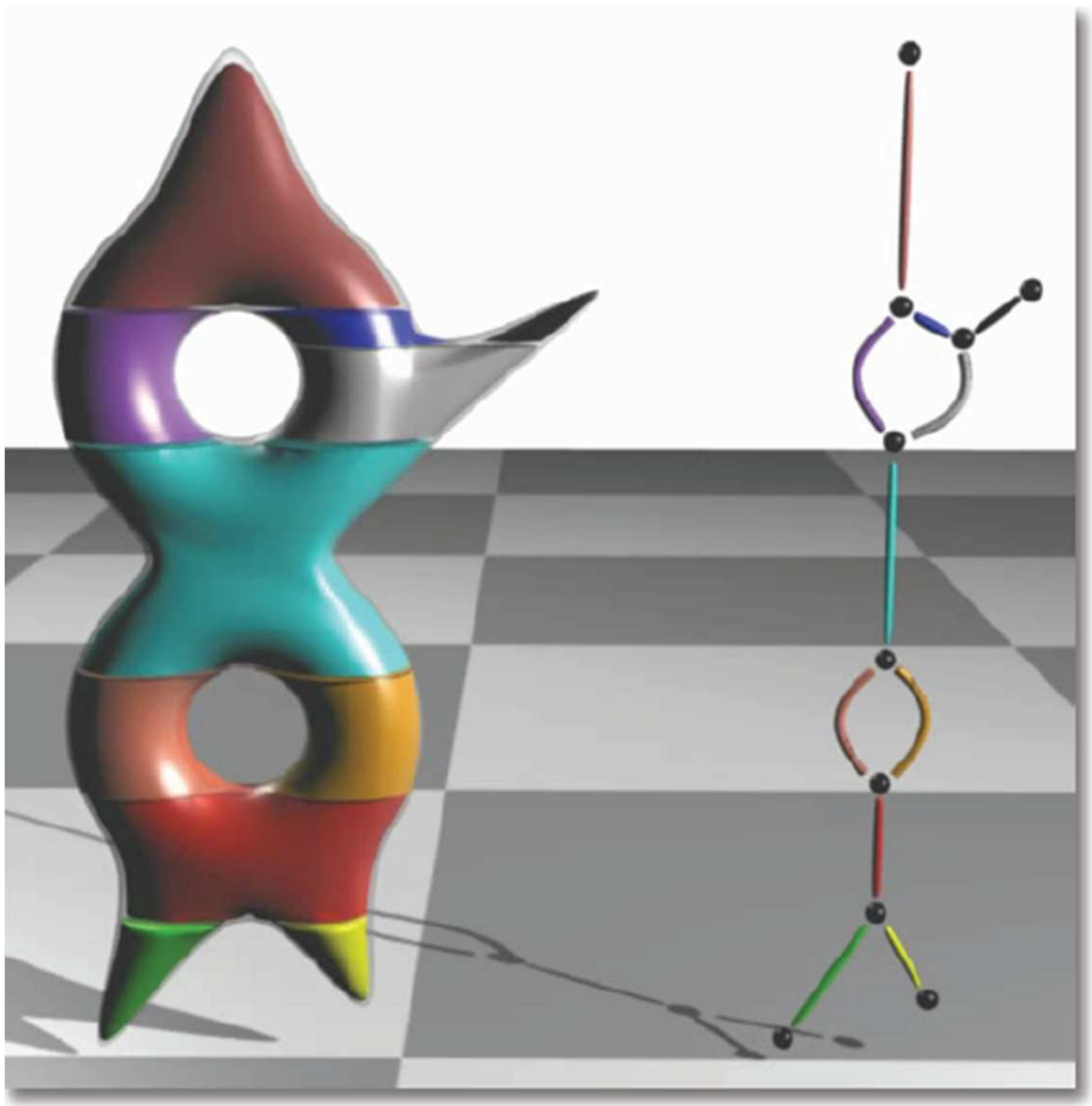
Level sets obtaining by sweeping along Z direction



Reeb Graph



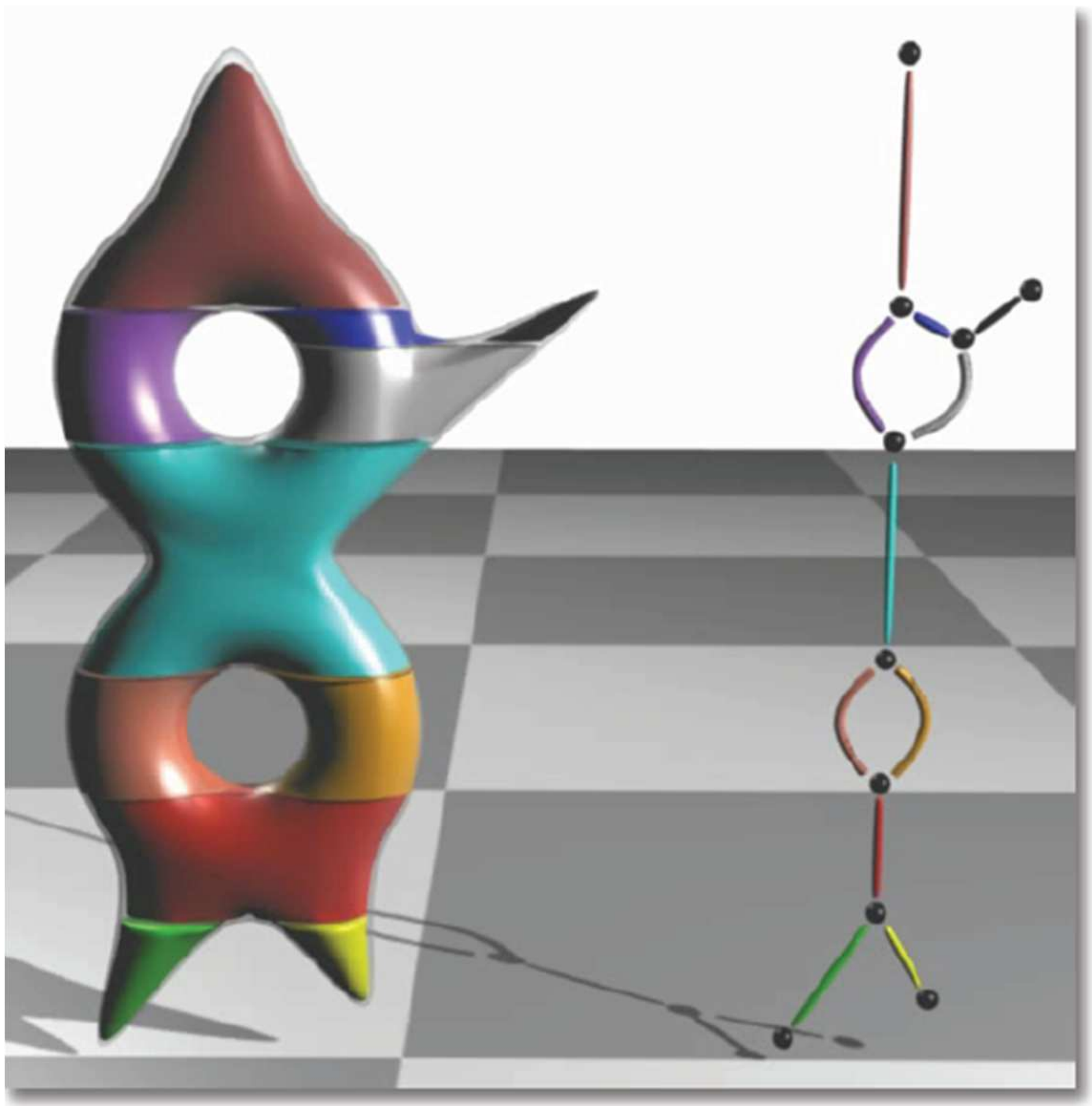
Reeb Graph



- Vertices of the graph are critical points
- Arcs of the graph are connected components (cylinders in domain)of the level sets of *f*, contracted to points

- Two-step algorithm
  - Locate critical points
  - Connect critical points

Reeb Graph



- Vertices of the graph are critical points
- Arcs of the graph are connected components (cylinders in domain)of the level sets of *f*, contracted to points

- Two-step algorithm
  - Locate critical points
  - Connect critical points

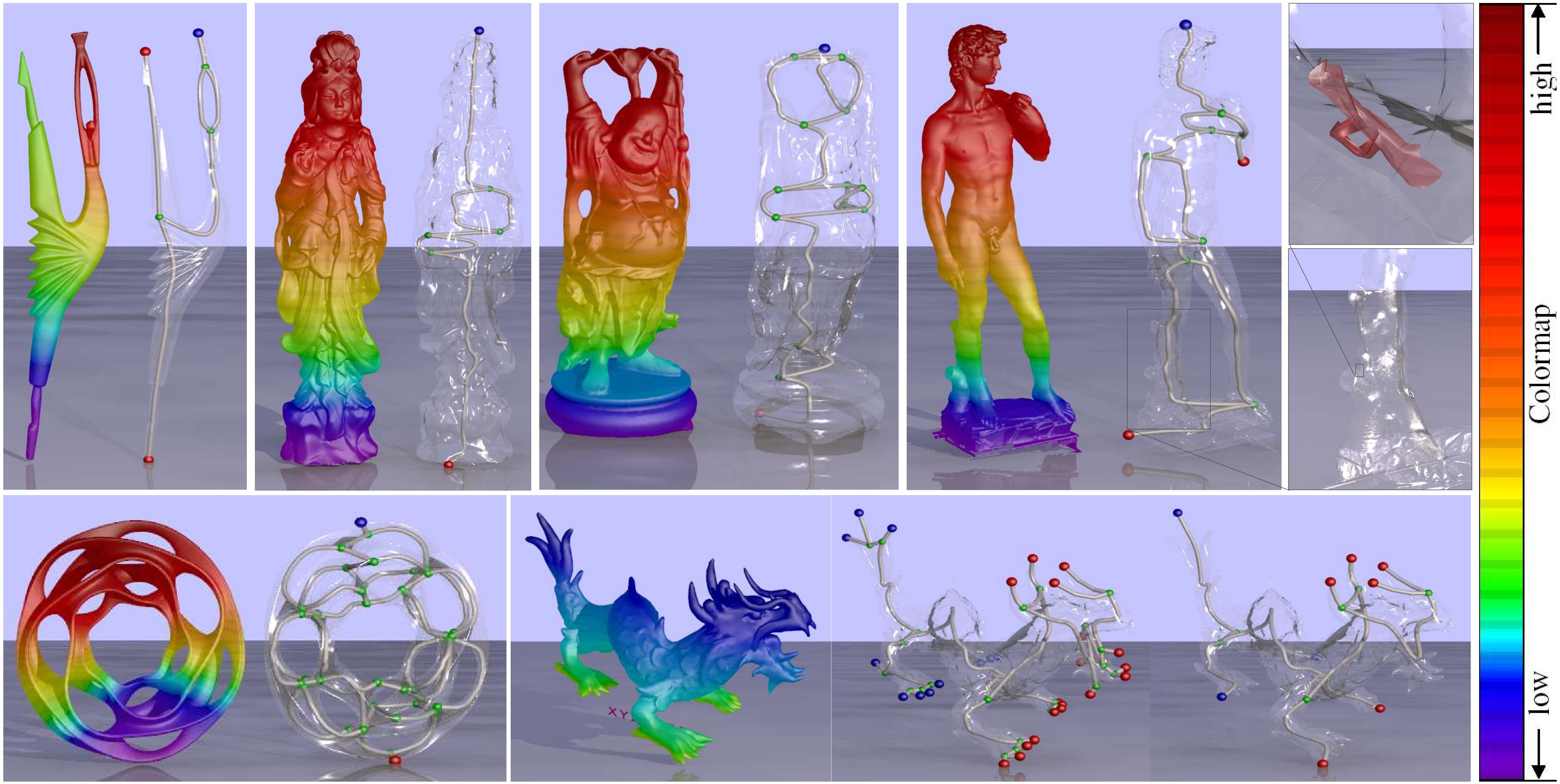
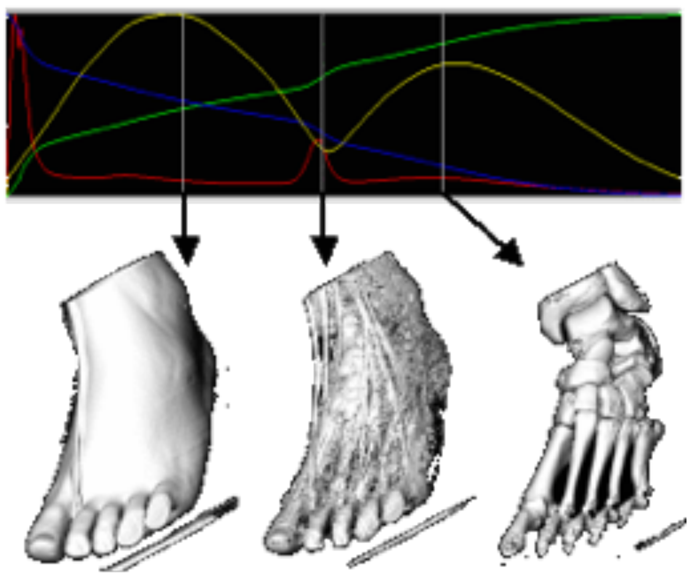


Figure 1: (Top row) Simplified Reeb graphs of the Dancer, Malaysian Goddess, Happy Buddha; and David together with two close-ups showing a tiny tunnel at the base of David's leg. The pseudo-colored surfaces show the function used for computing the Reeb graph. The transparent models show the structure of the Reeb graph and its embedding. (Bottom row) The Heptoroid model and two levels of resolution for the Reeb graph of the Asian Dragon model.

Valerio Pascucci, Giorgio Scorzelli, Peer-Timo Bremer, Ajith Mascarenhas: Robust on-line computation of Reeb graphs: simplicity and speed. ACM TOG. 26(3): 58 (2007)

Contour and merge trees



Bajaj et al. The Contour Spectrum. IEEE Vis 97

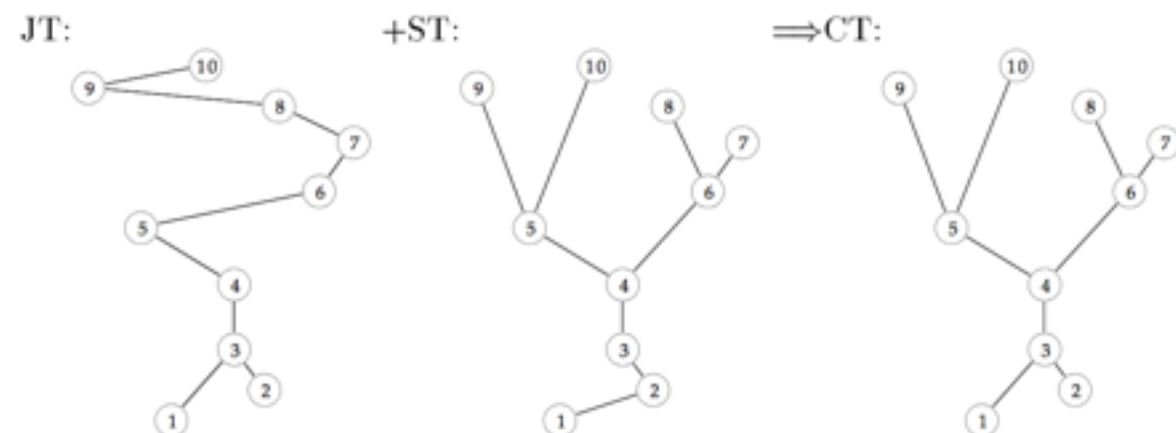
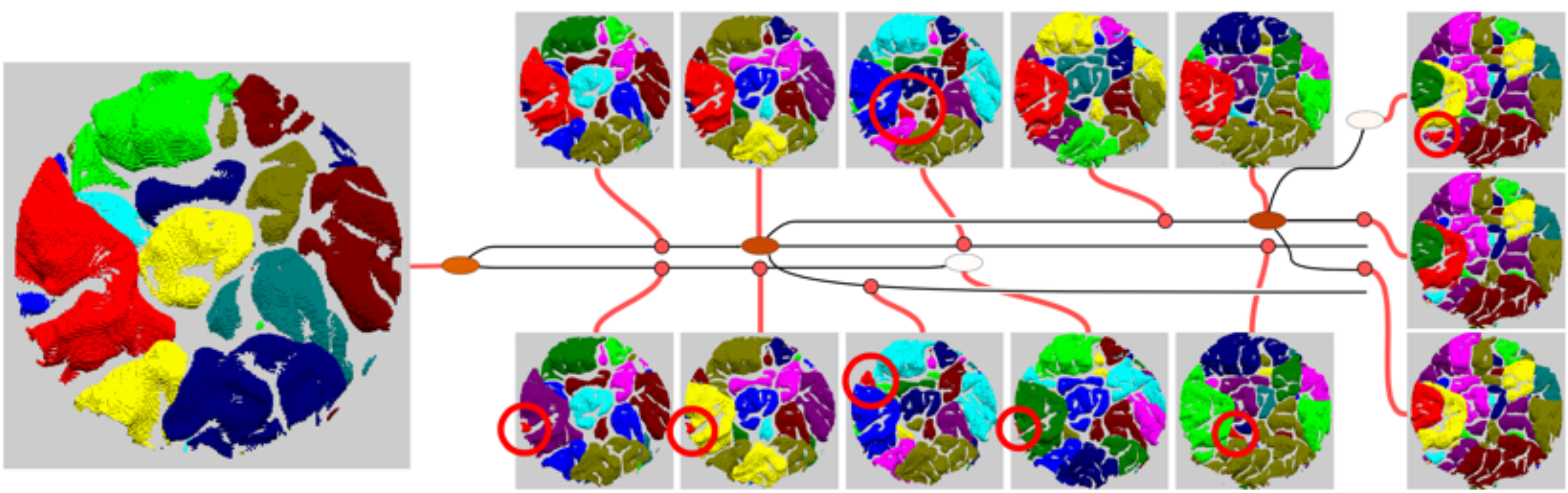


Figure 4: Augmented join and split trees merge to form the contour tree

Carr et al. Computing Contour Trees in All Dimensions. Computational Geometry, 2003.

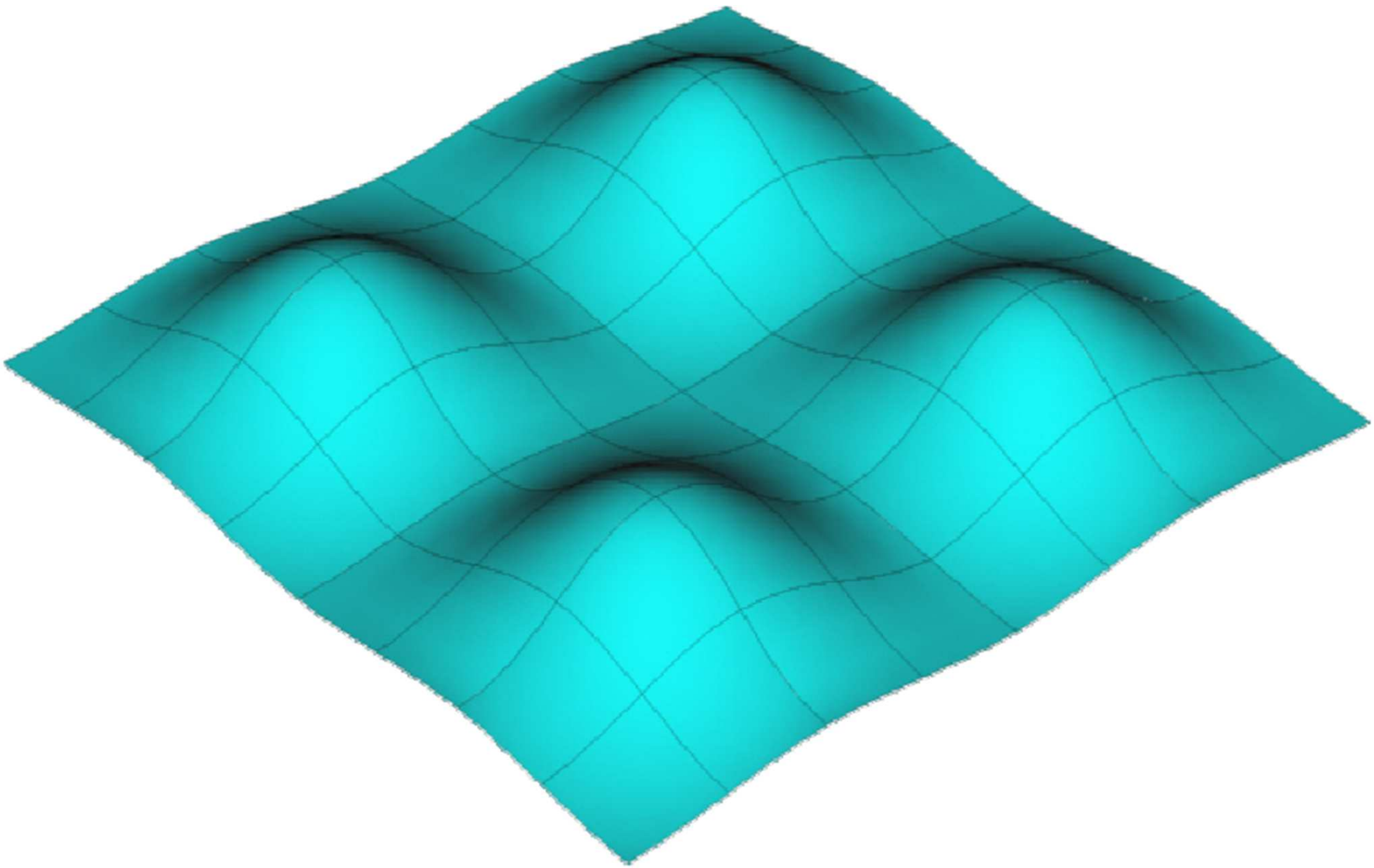


Join (Merge) trees - Bremer et al. Interactive Exploration and Analysis of Large Scale Simulations Using Topology-based Data Segmentation, IEEE TVCG 2011

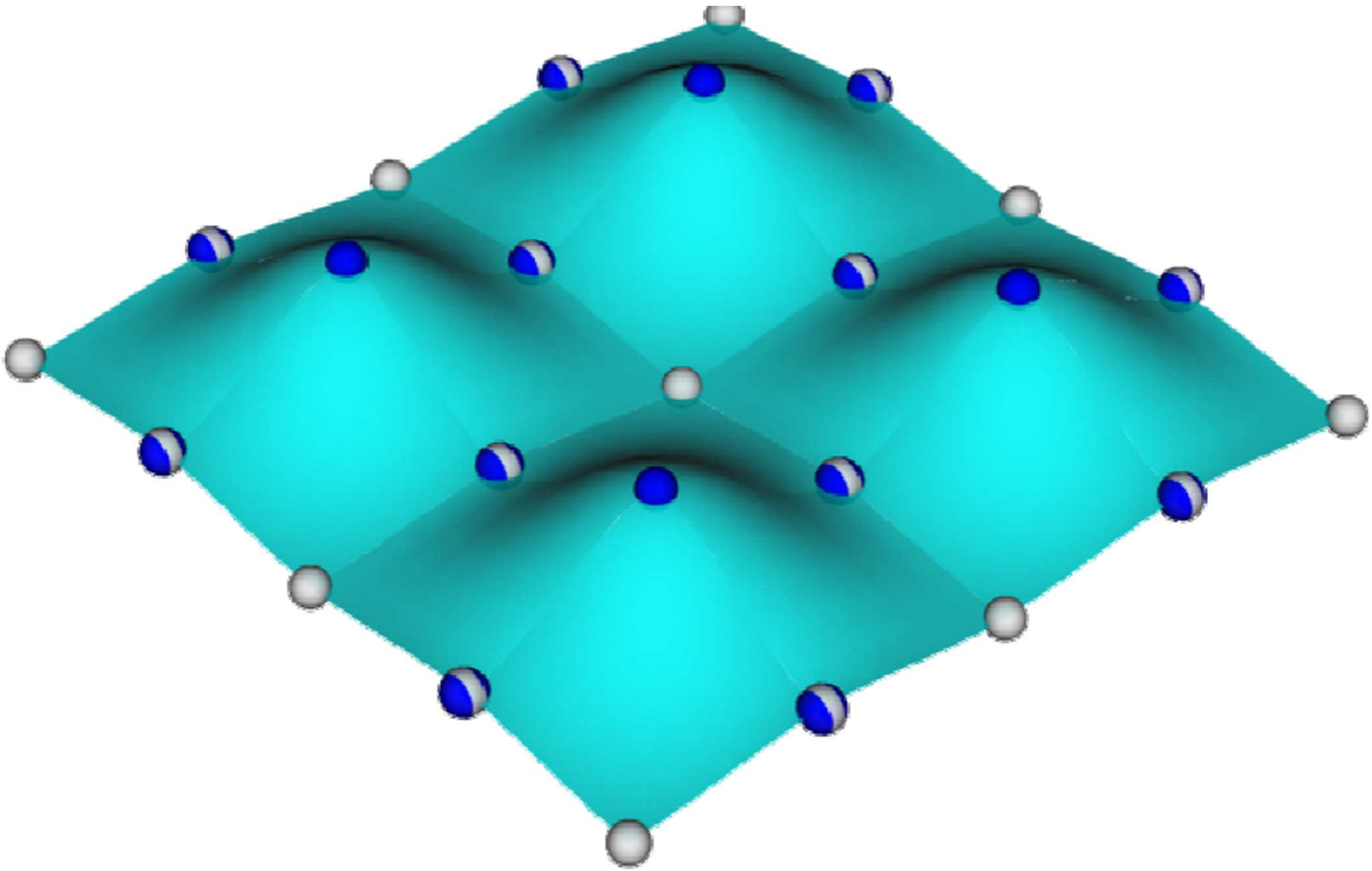
Gradient-Field Topology

Morse and Morse-Smale Complexes

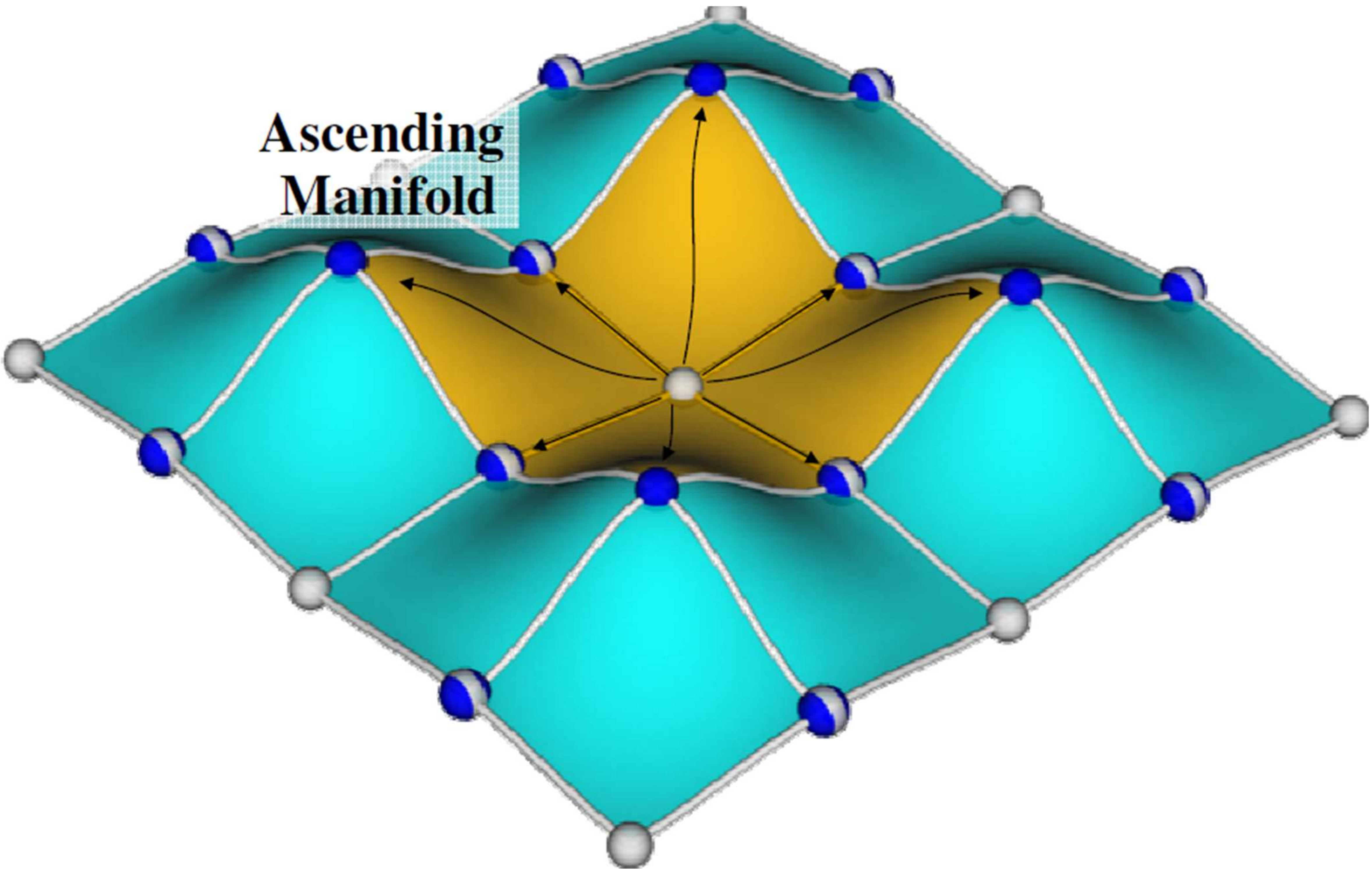
Morse-Smale Complex-2D



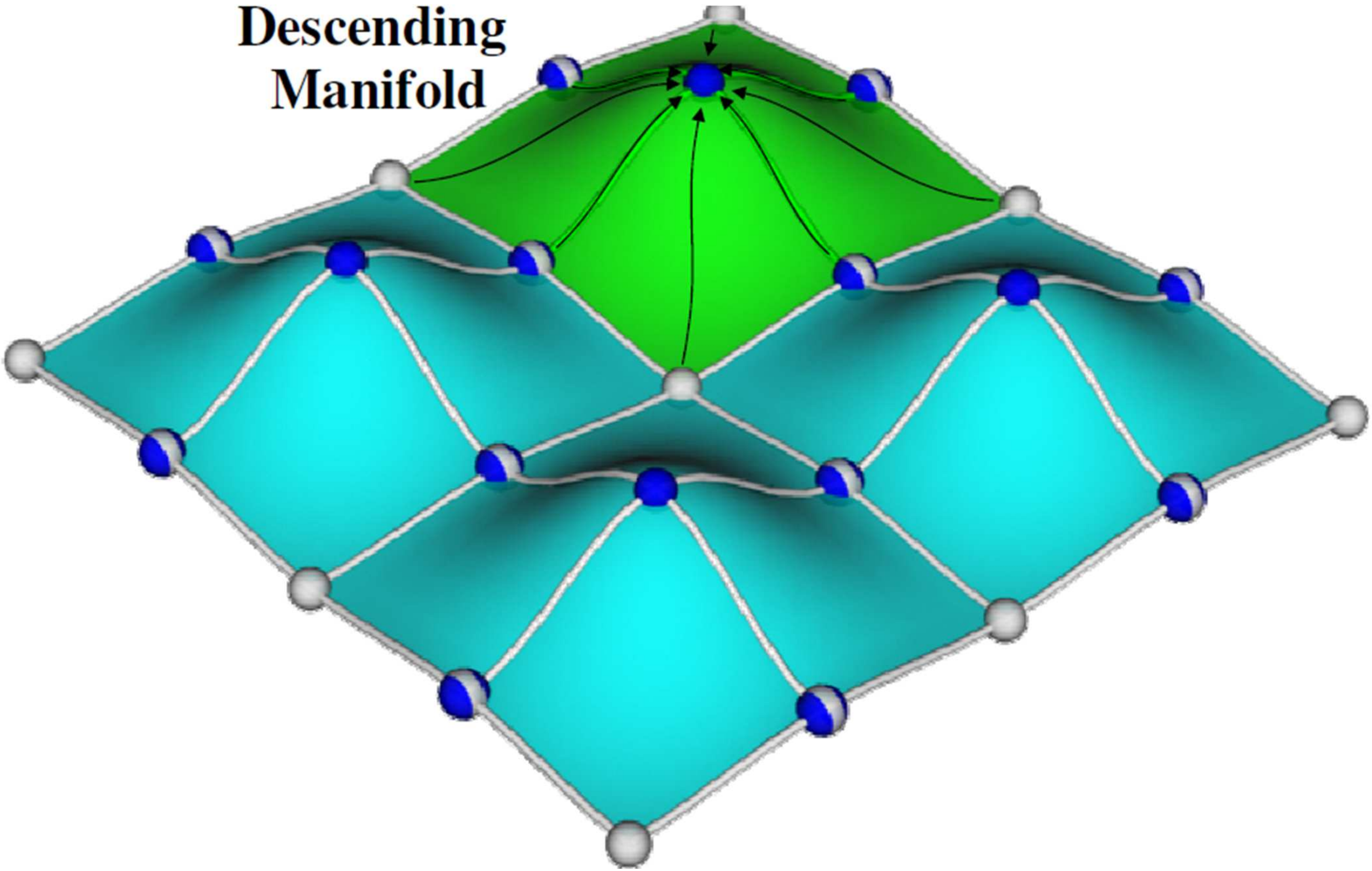
Morse-Smale Complex-2D



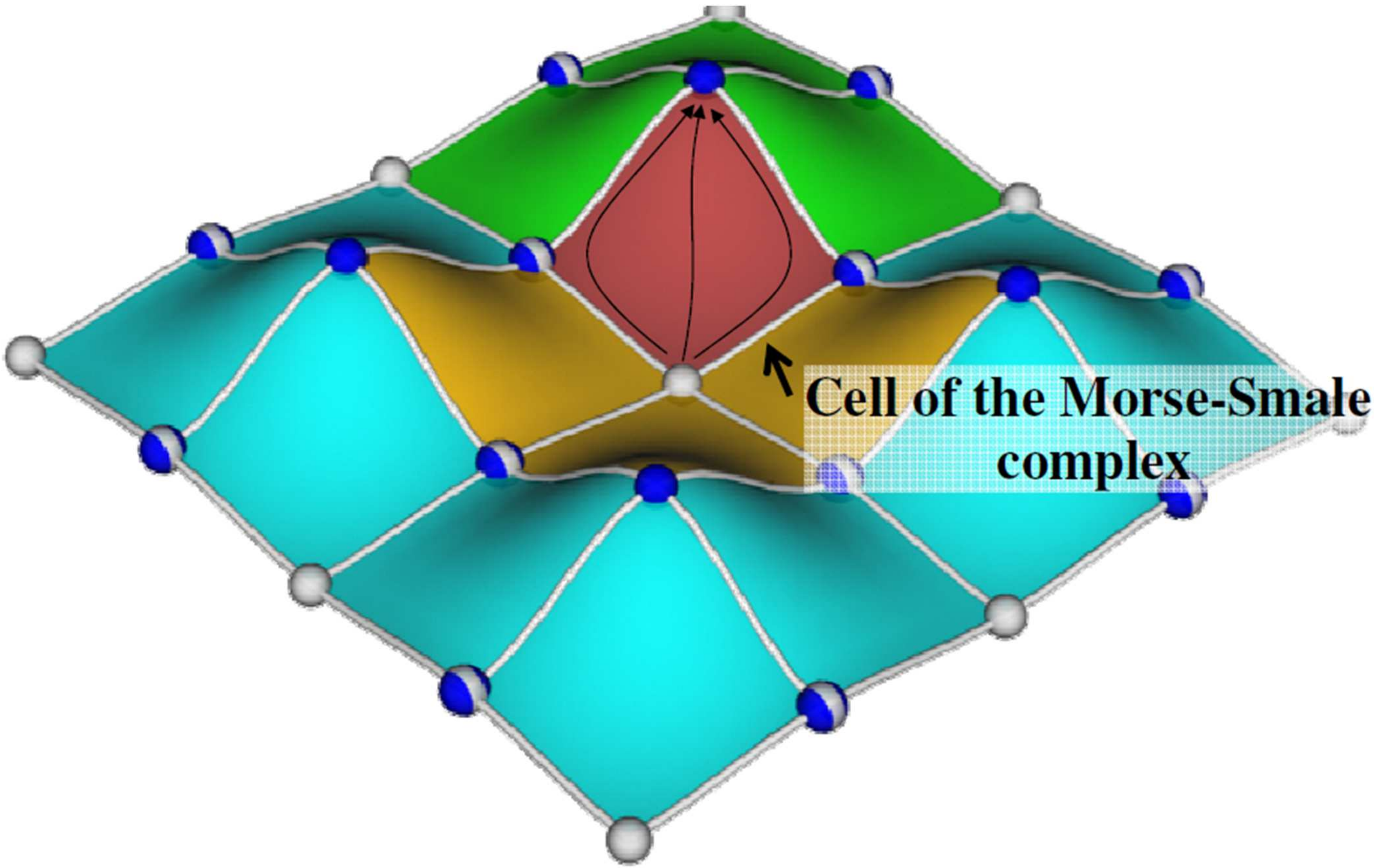
Morse-Smale Complex-2D



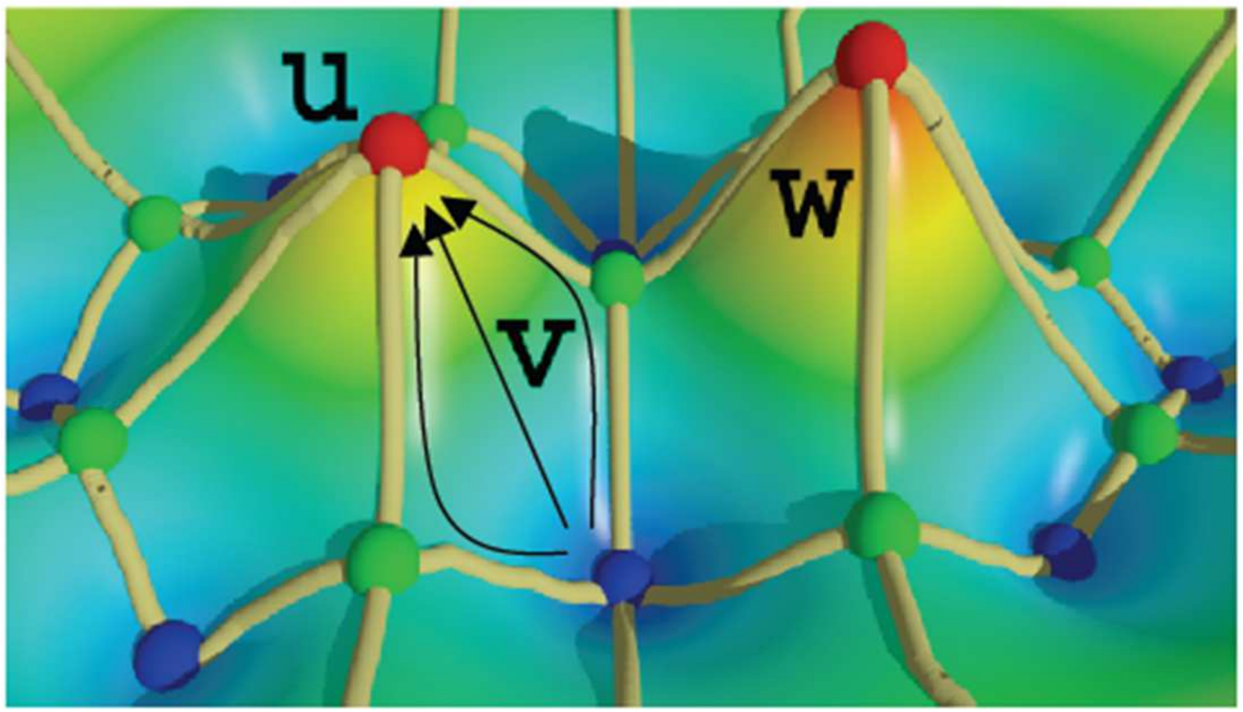
Morse-Smale Complex-2D

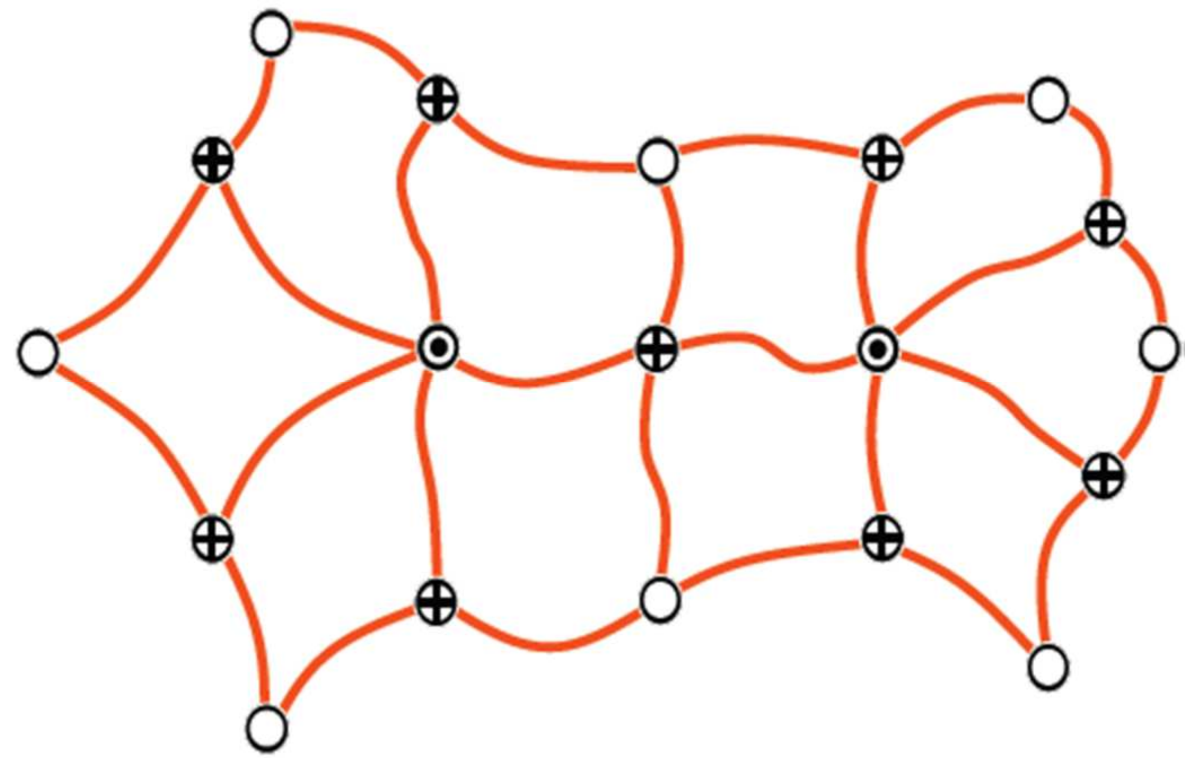


Morse-Smale Complex-2D



Morse-Smale Complex-2D





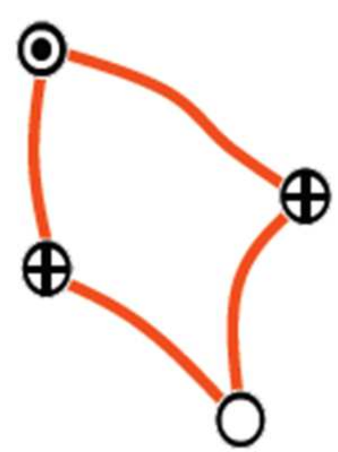
Decomposition into monotonic regions

Combinatorial Structure 2D

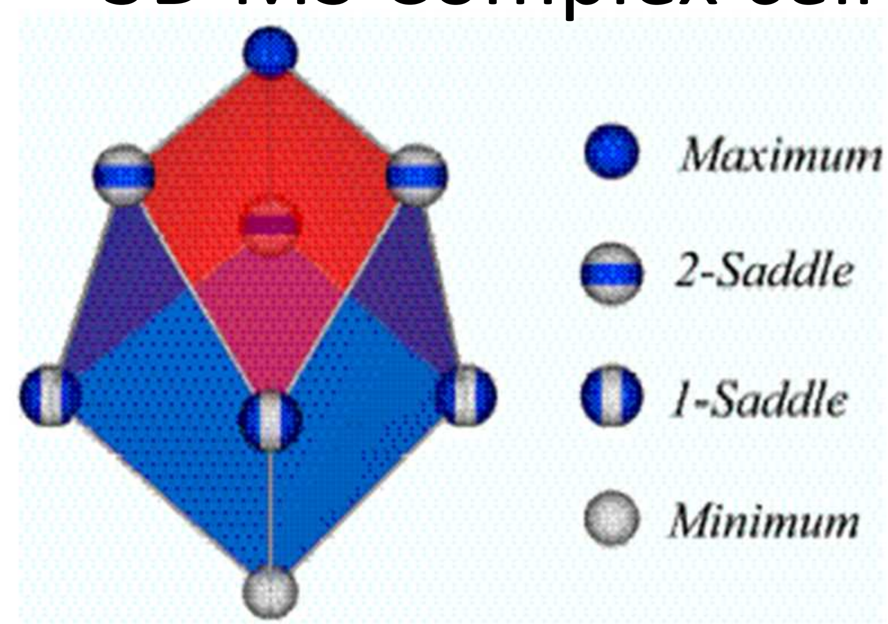
- Nodes of the MS complex are exactly the critical points of the Morse function
- Saddles have exactly four arcs incident on them

All regions are quads

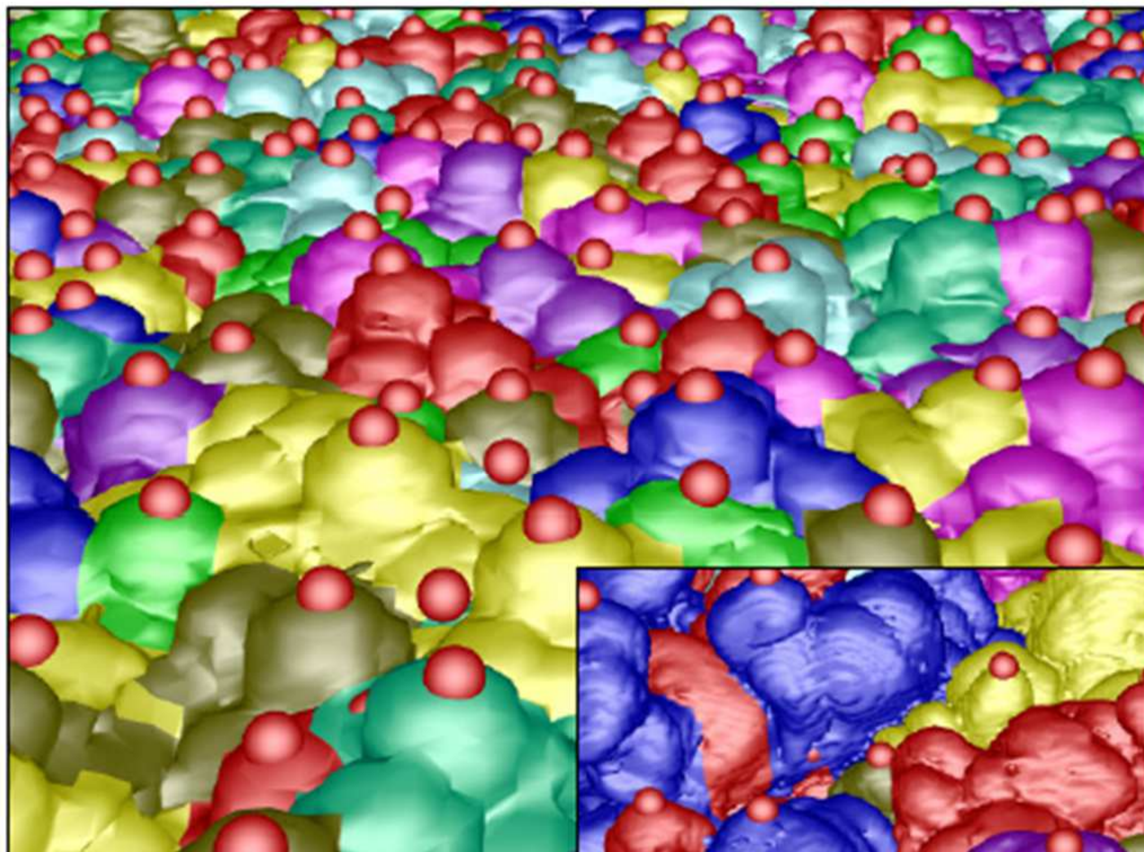
- Boundary of a region alternates between saddle-extremum
- 2k minima and maxima



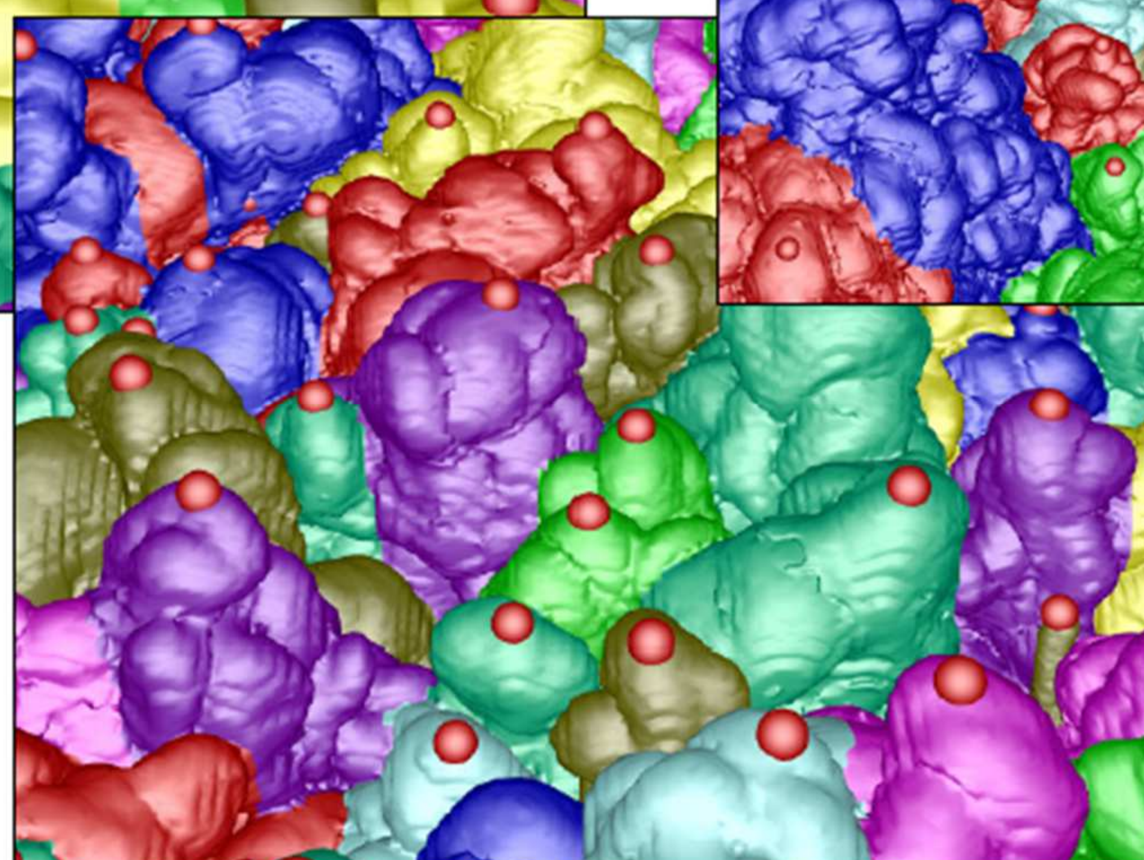
3D MS Complex cell



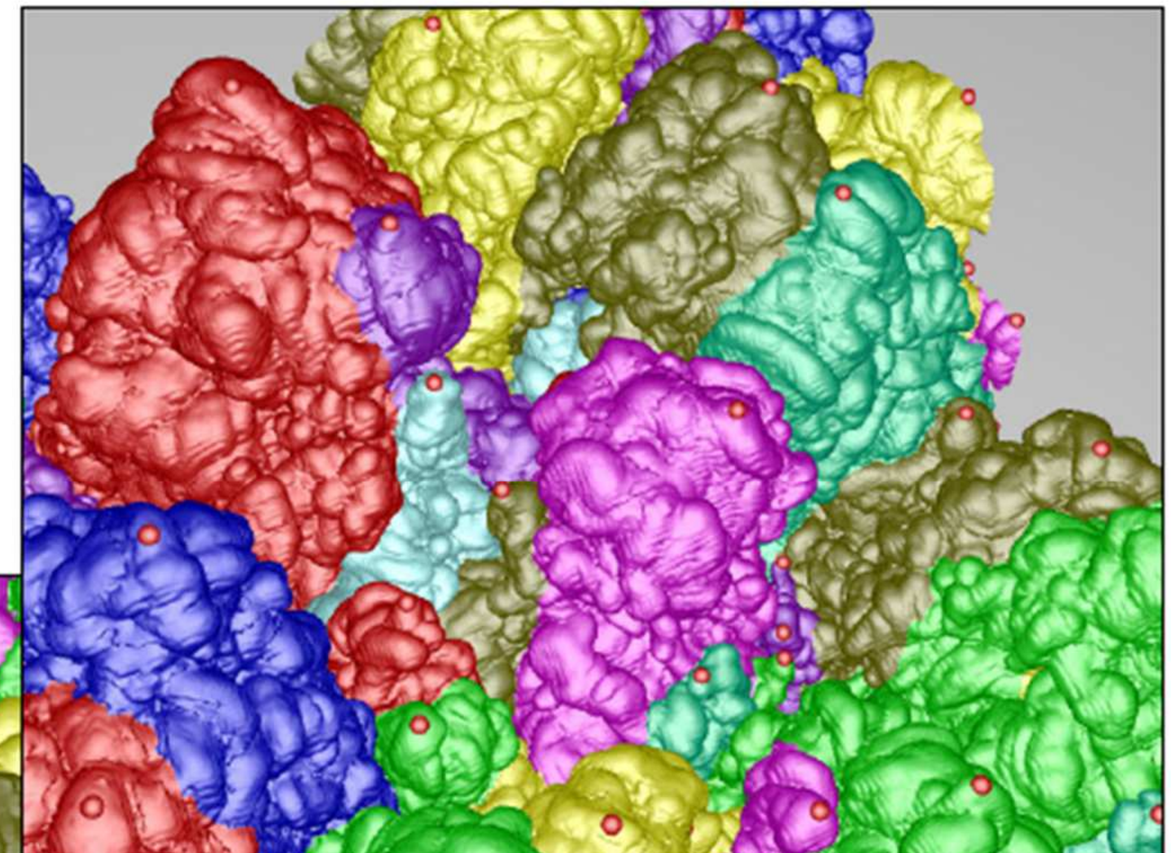
Applications



T=100



T=353



T=700

Rayleigh-Taylor turbulence analysis

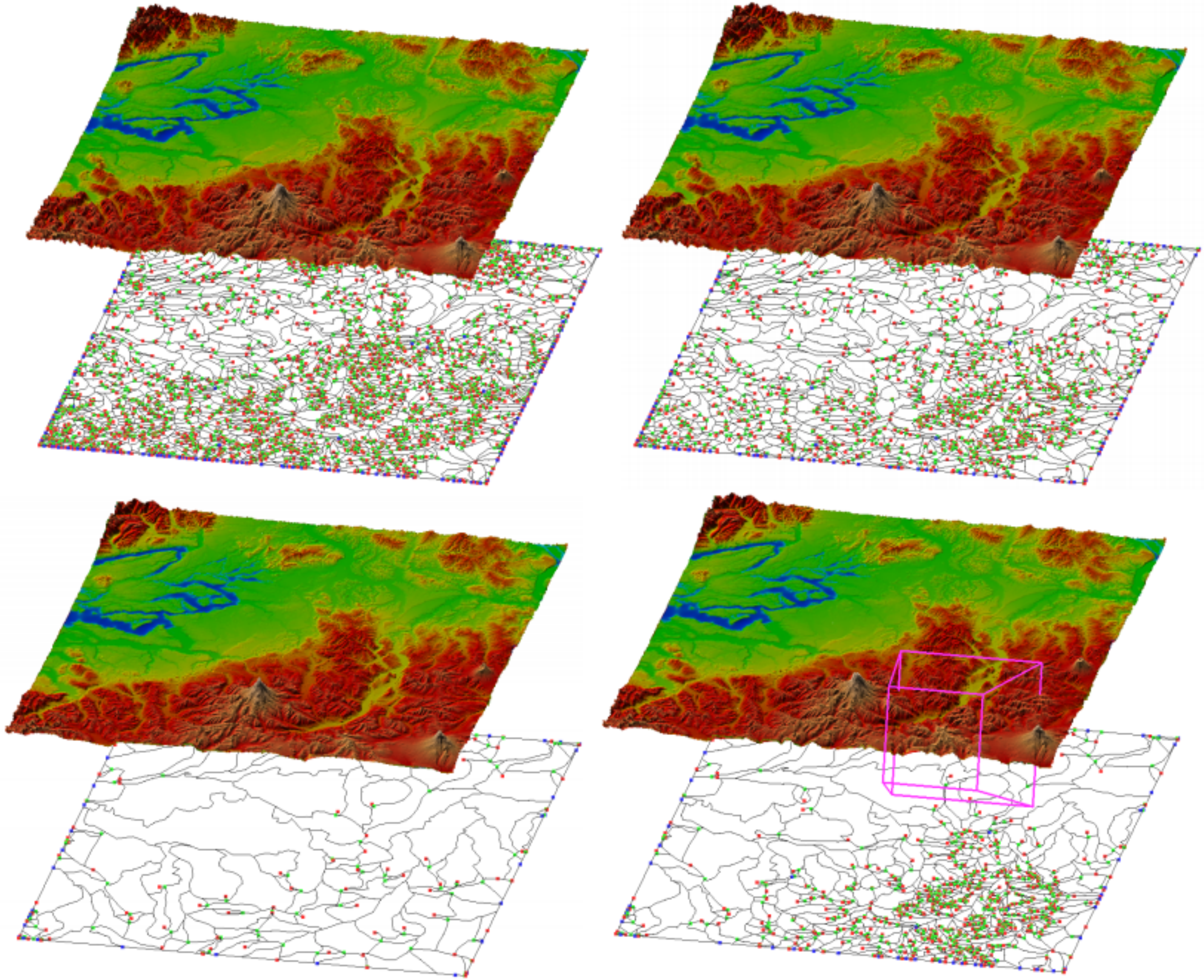


Figure 11: (Upper-left) Puget Sound data after topological noise removal. (Upper-right) Data at persistence of 1.2% of the maximum height. (Lower-left) Data at persistence 20% of the maximum height. (Lower-right) View-dependent re□nement (purple: view frustum).

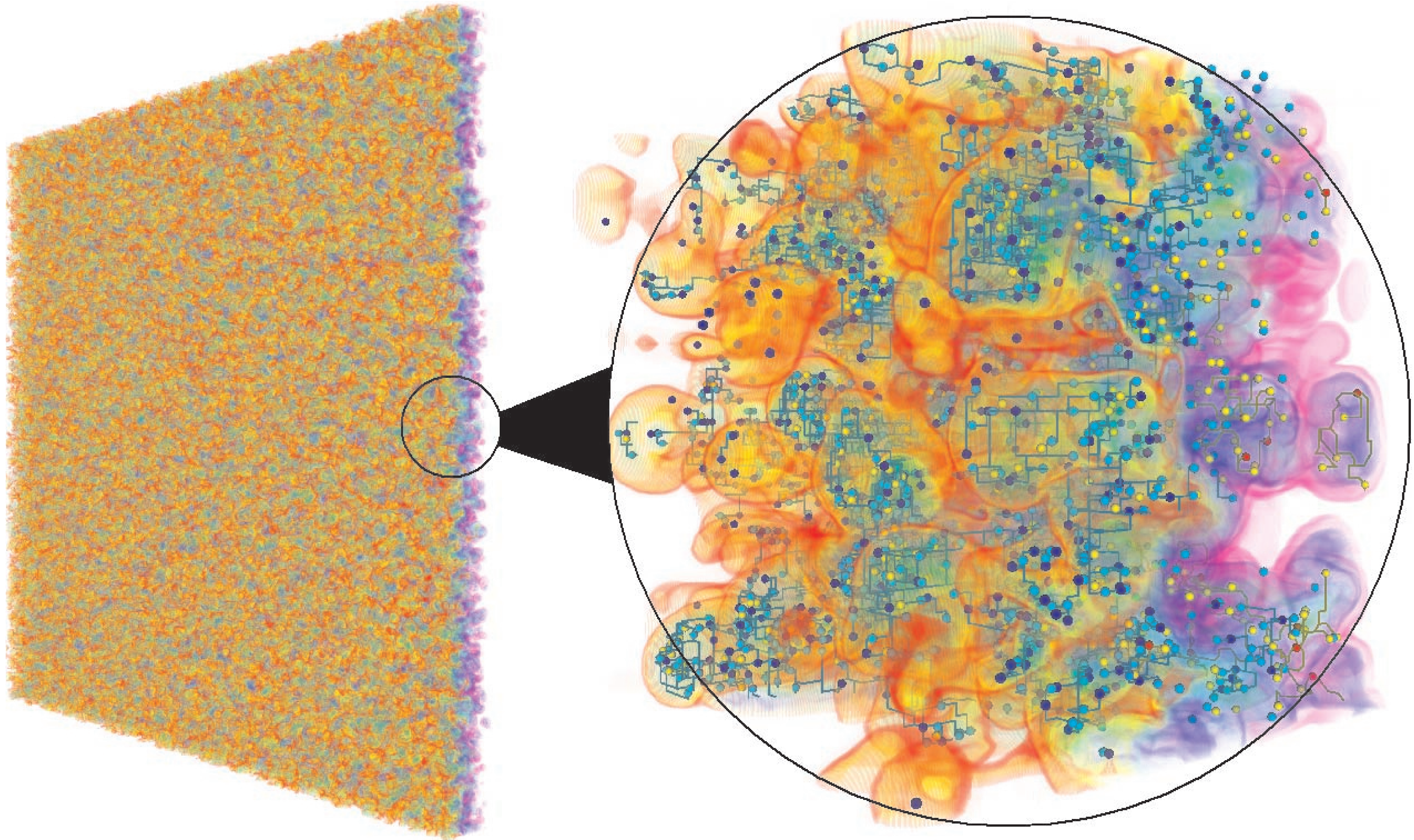
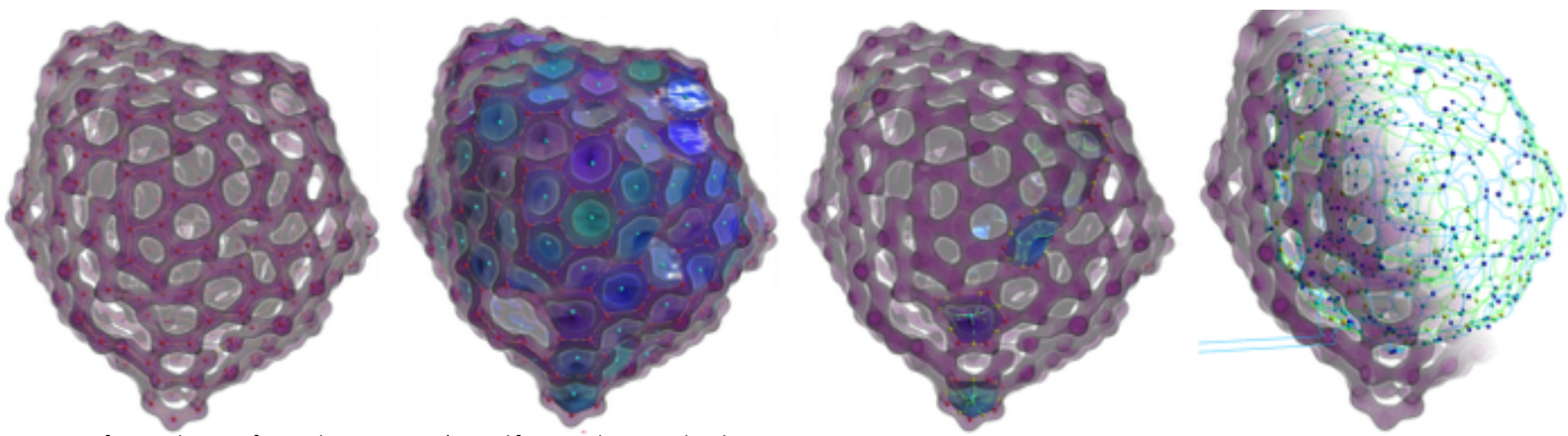


Fig. 5. A single timestep of a dataset of a simulated Raleigh-Taylor instability simulating the mixing of two fluids. This timestep has a resolution of $1152 \times 1152 \times 1000$ and is an early timestep of the simulation. The data is noisy, therefore we perform a 5% persistence simplification to remove "excess features." We compute the complex for the entire dataset, and the inset shows a small subsection of the data with selected nodes and arcs of the complex. Minima and maxima (blue and red spheres) and their saddle connections trace out the bubble structure in the data. The maxima represent isolated pockets of high-density fluid that have crossed the boundary between the two fluids. The structural complexity is overwhelming, but our prototype allows interactive exploration and visualization, and selective inclusion/omission of user-specified components of the MS complex.

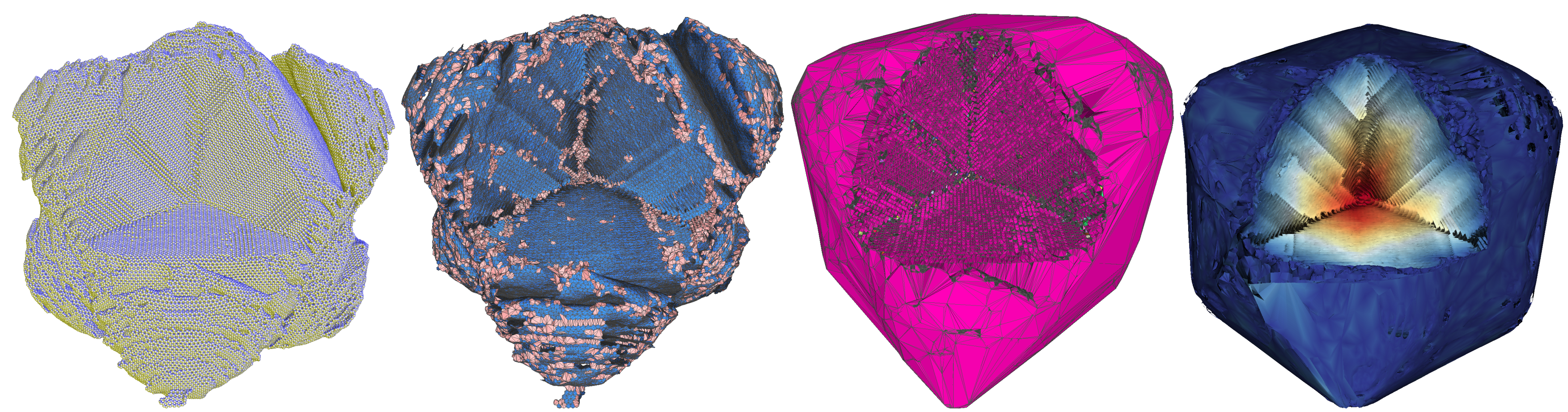
Gyulassy, Bremer, Hamann, Pascucci, 2008

Morse-Smale Battery Analysis



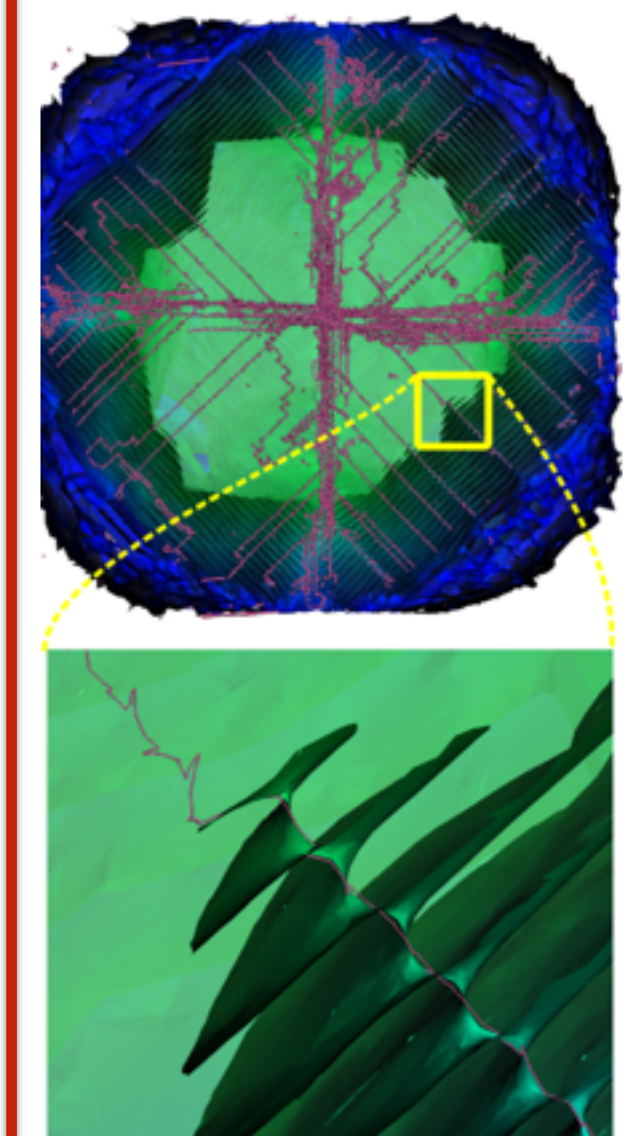
wavefunction of carbon nanosphere | classify carbon chains with MS complex | identify defect sites | MS complex

Gyulassy et al. Morse-Smale Analysis of Ion Diffusion in Ab Initio Battery Materials Simulations. TopoInVis 2015



atom geometry (LAMMPS) | classified blocking and non-blocking defect sites | Li-accessible regions of nanosphere | Li diffusion distance for saturation

Gyulassy et al. Interstitial and Interlayer Ion Diffusion Geometry Extraction in Graphitic Nanosphere Battery Materials. IEEE Visualization 2015

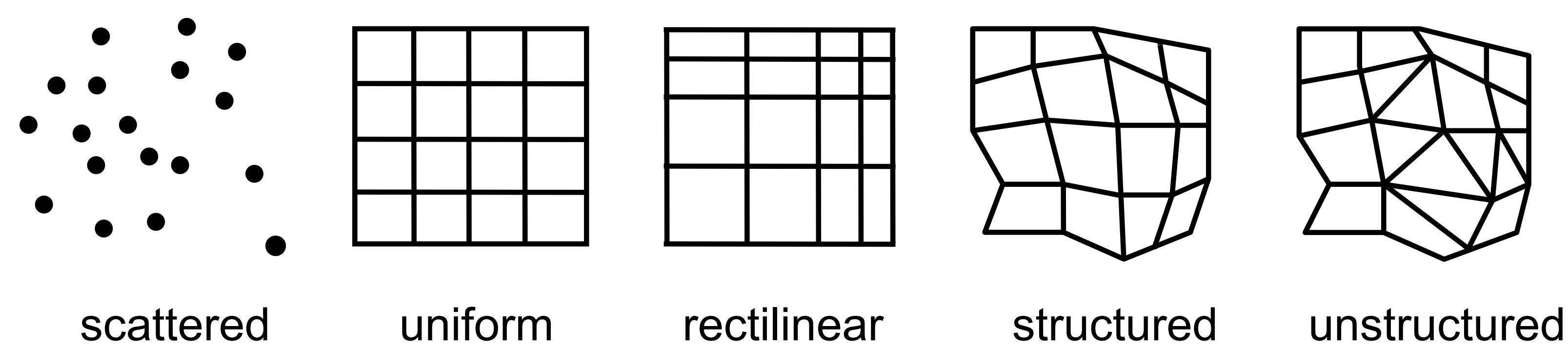


New finding: most ion movement occurs through large faults in the structure.

Flow Visualization

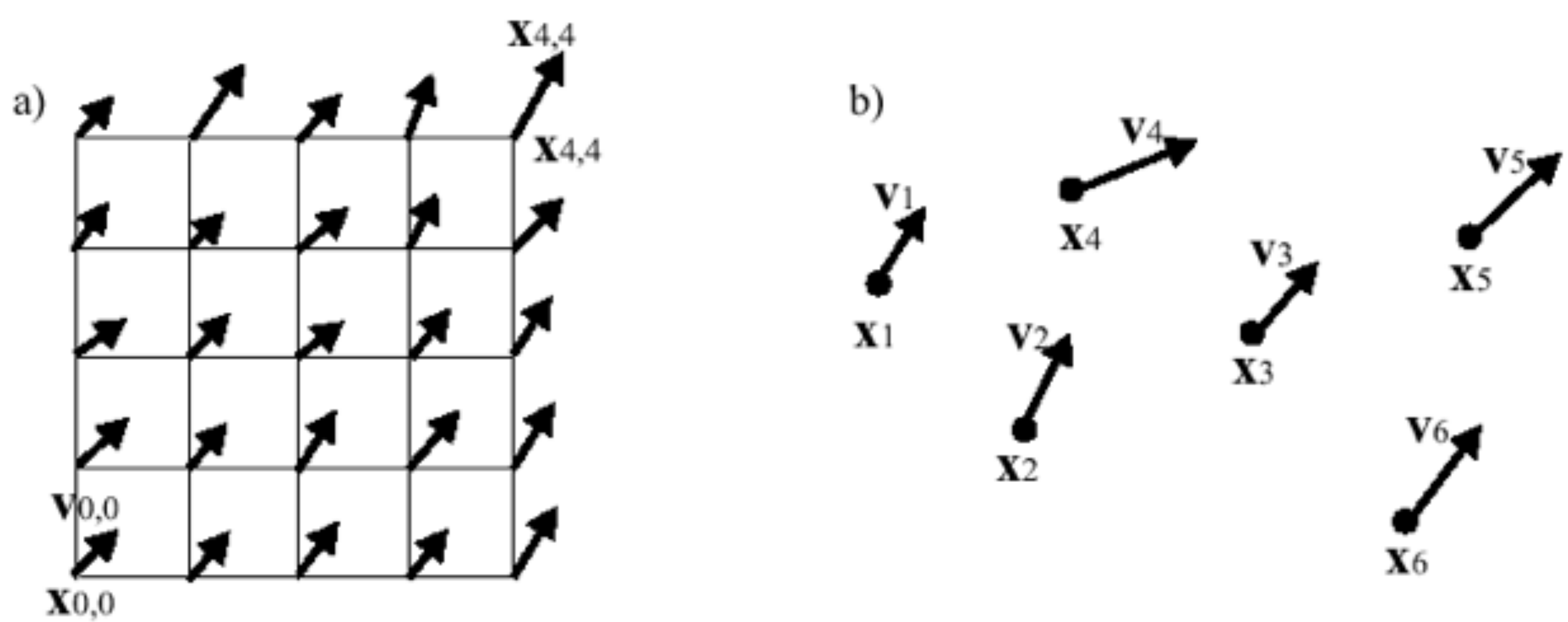
Vector fields

- Vector data on a 2D or 3D grid



- Additional scalar data may be defined per grid point

- Example on a regular grid (a) or scattered data points (b)



More formally

scalar field

$$s : \mathbb{E}^n \rightarrow \mathbb{R}$$

vector field

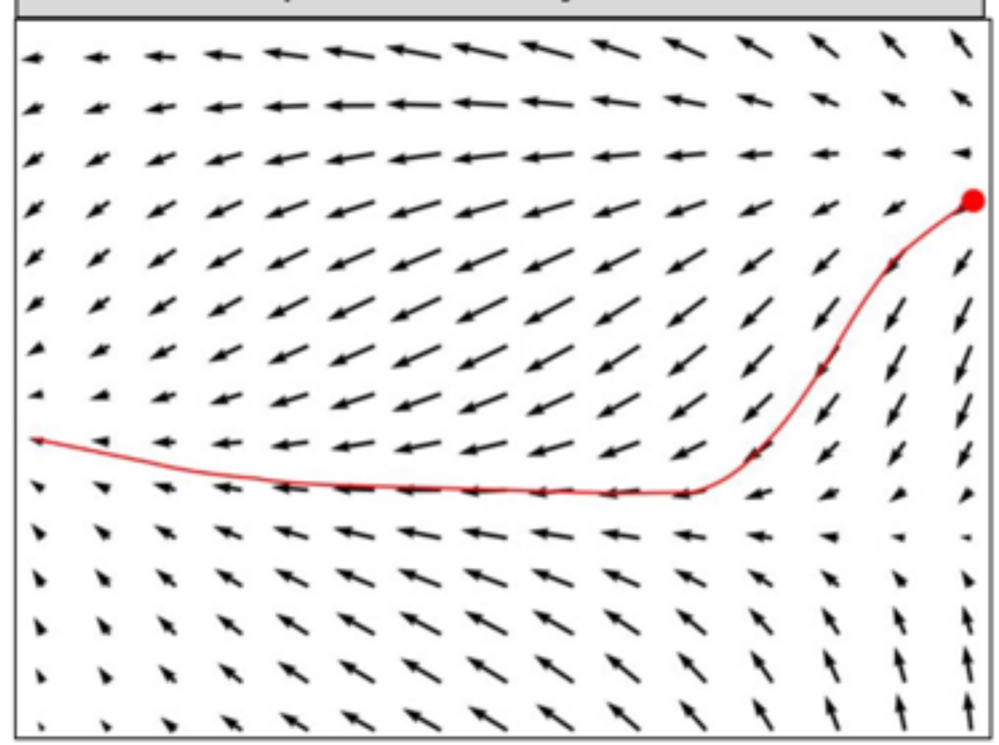
$$\mathbf{v} : \mathbb{E}^n \rightarrow \mathbb{R}^m$$

- m=n *usually* — but not always.
- *The vector is the element of the field*
  (in contrast to multifields)
- Typically, the vector field can be expressed as an ordinary differential equation (ODE), e.g.,

$$\frac{d\varphi(x)}{dt} = V(x)$$

- Solving (integrating) this ODE results in **flow,** i.e. the set of particle trajectories in this field.
- *Flow vis is about how we select and show these trajectories.*

A simple 2D **steady** vector field



- Main application of vector field visualization is flow visualization
  - Motion of fluids (gas, liquids)
  - Geometric boundary conditions
  - Velocity (flow) field $\boldsymbol{v}(\boldsymbol{x},t)$
  - Pressure p
  - Temperature T
  - Vorticity $\nabla \times \boldsymbol{v}$
  - Density ρ
  - Conservation of mass, energy, and momentum
  - Navier-Stokes equations
  - CFD (Computational Fluid Dynamics)

Experimental flow visualization

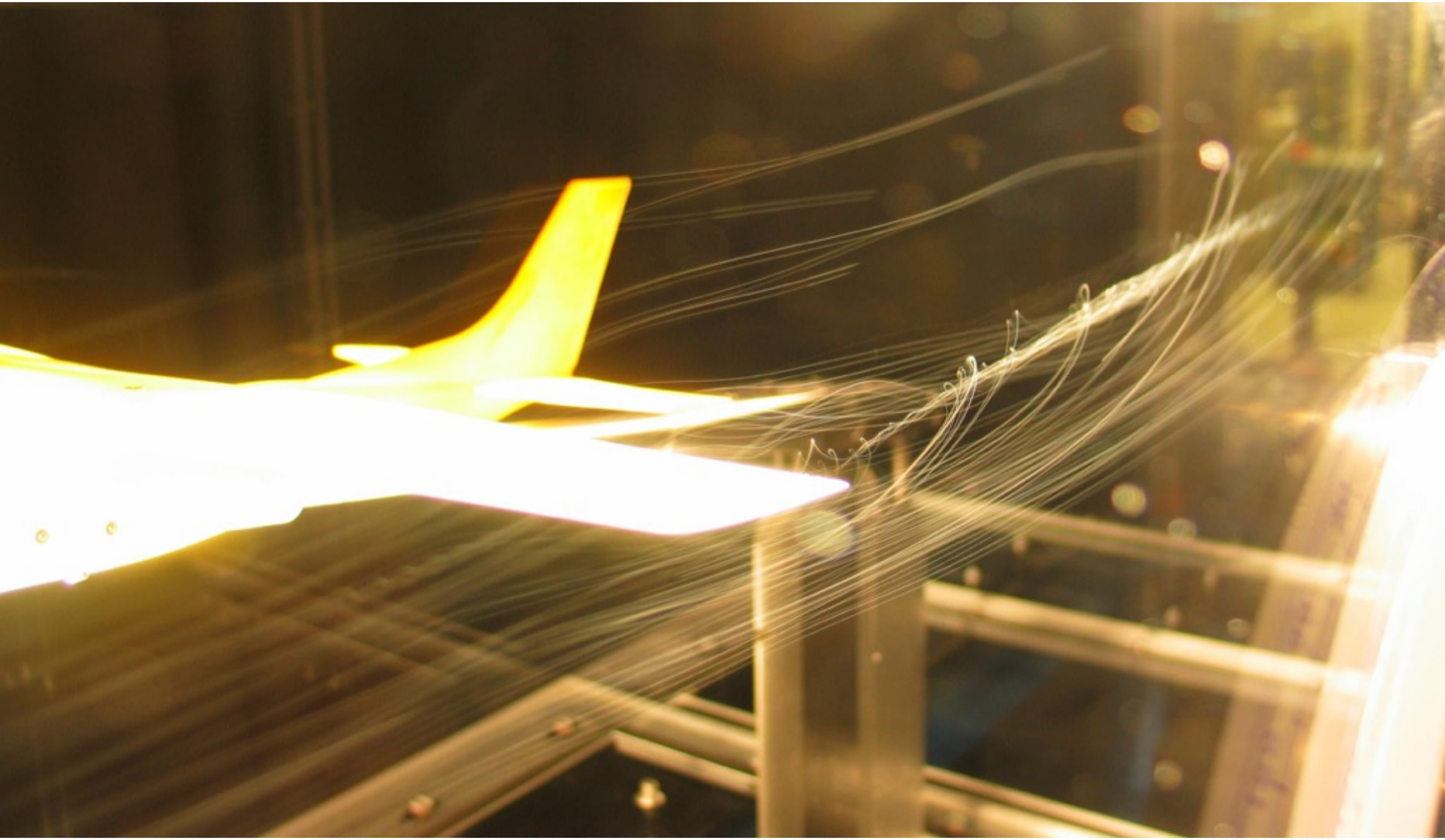
Milestones in Flight History
Dryden Flight Research Center

L-1011

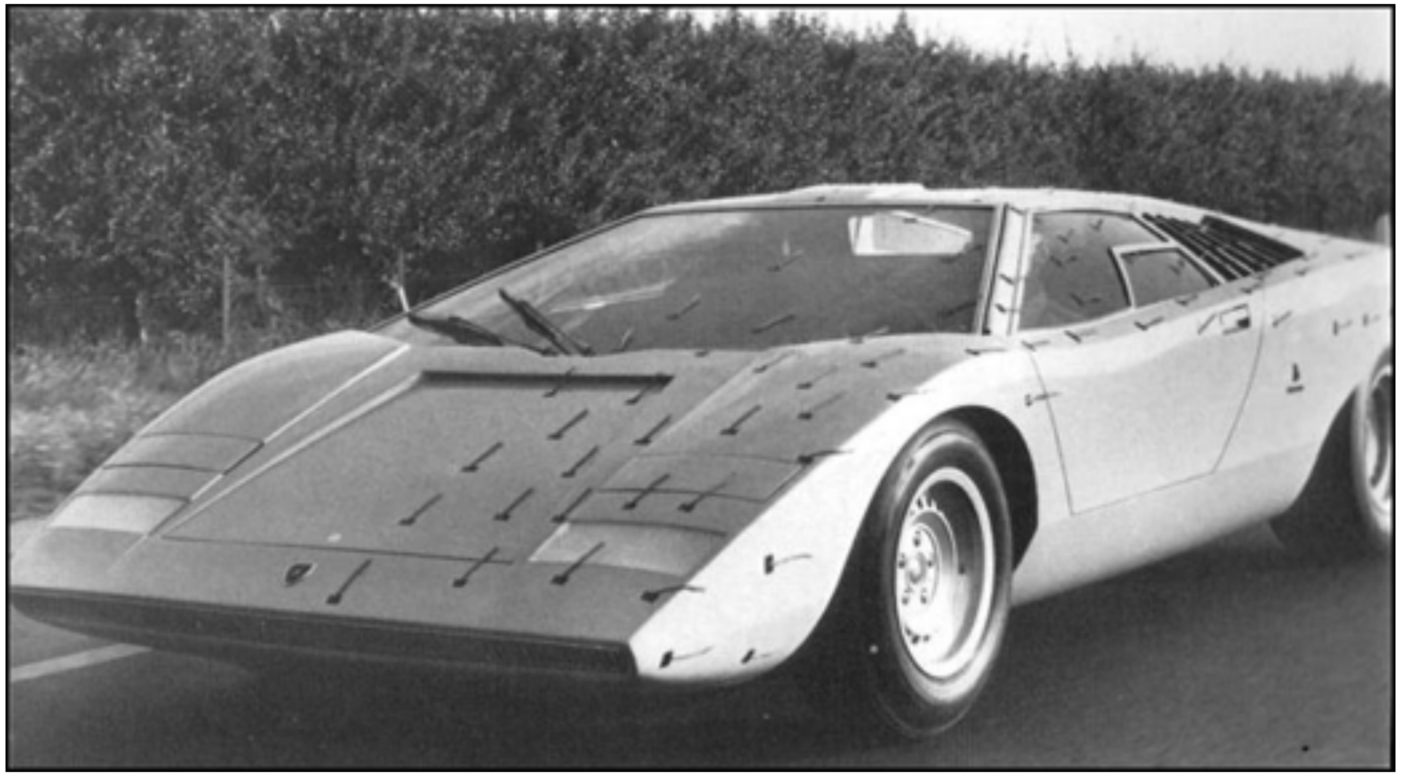
Airliner Wing Vortice Tests at Langley

Circa 1970s

Smoke angel
A C-17 Globemaster III from the 14th Airlift Squadron, Charleston Air Force Base, S.C. flies off after releasing flares over the Atlantic Ocean near Charleston, S.C., during a training mission on Tuesday, May 16, 2006. The "smoke angel" is caused by the vortex from the engines.
(U.S. Air Force photo/Tech. Sgt. Russell E. Cooley IV)



A wind tunnel model of a Cessna 182 showing a wingtip vortex.
Tested in the RPI (Rensselaer Polytechnic Institute) Subsonic Wind Tunnel.
By Ben FrantzDale (2007).



http://autospeed.com/cms/A_108677/article.html

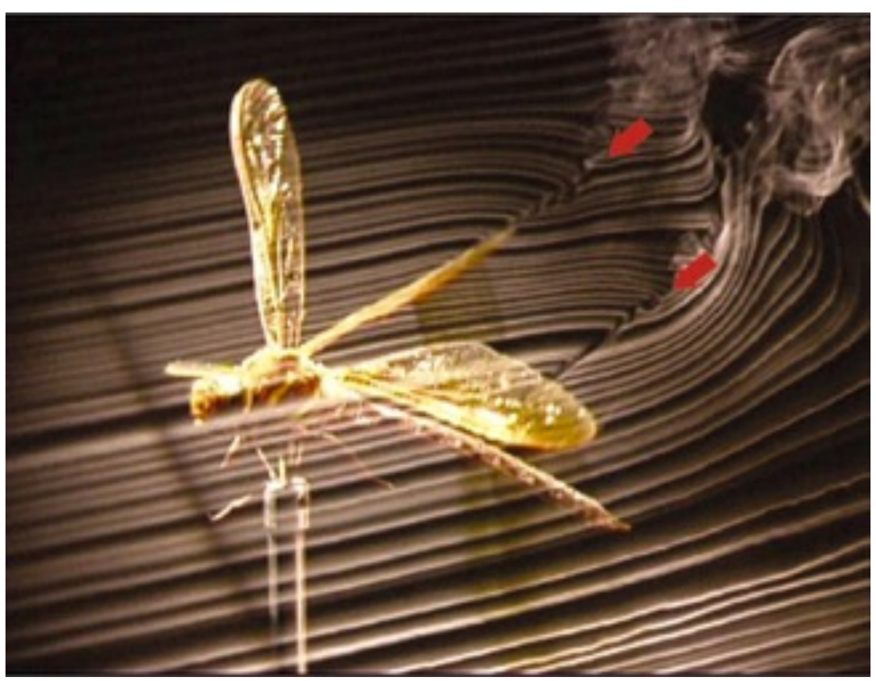
http://autospeed.com/cms/A_108677/article.html

Wool Tufts

Smoke Injection

http://autospeed.com/cms/A_108677/article.html

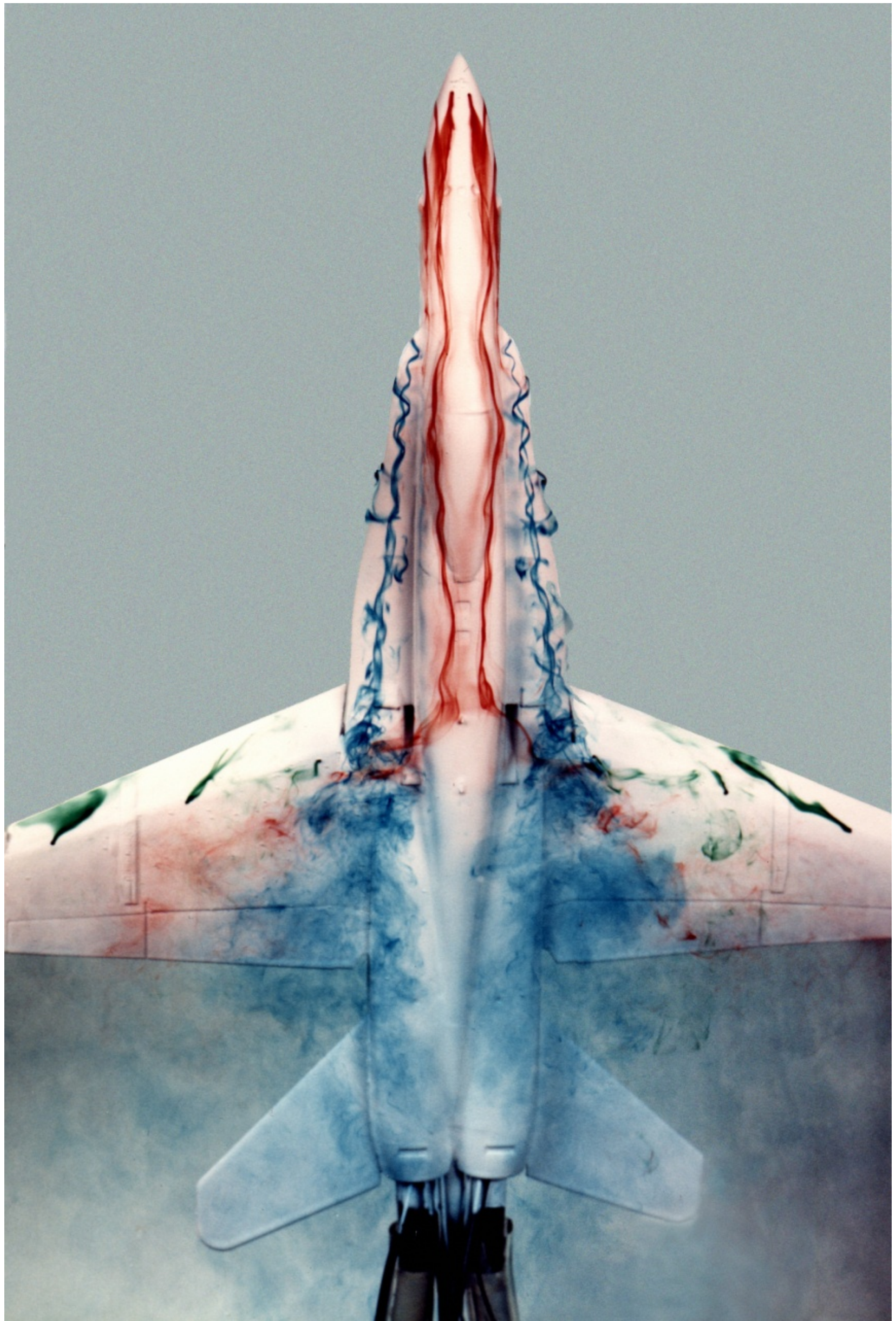
Smoke Nozzles



[NASA, J. Exp. Biol.]

http://autospeed.com/cms/A_108677/article.html http://de.wikipedia.org/wiki/Bild:Airplane_vortex_edit.jpg

Streaklines in Experimental Flow Vis



NASA Dryden Flight Research Center Photo Collection
http://www.dfrc.nasa.gov/gallery/photo/index.html
NASA Photo: ECN-33298-03 Date: 1985

1/48-scale model of an F-18 aircraft in Flow Visualization Facility (FVF)

Dryden Flight Research Center ECN 33298-47 Photographed 1985
F-18 water tunnel test in Flow Visualization Facility NASA/Dryden

Computational fluid dynamics

Fluid dynamics

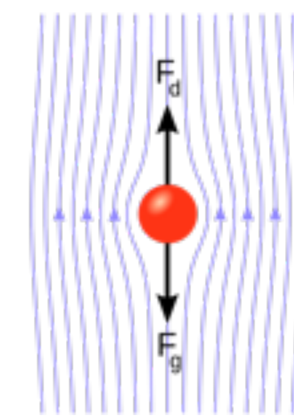
- **Navier-Stokes equations:** a set of PDE's modeling the behavior of fluids.
  Example for compressible fluids:

$$\underbrace{\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)}_{1} = \underbrace{-\nabla p}_{2} + \underbrace{\nabla\cdot\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)}_{3} + \underbrace{\mathbf{F}}_{4}$$

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0$$

Continuity equation

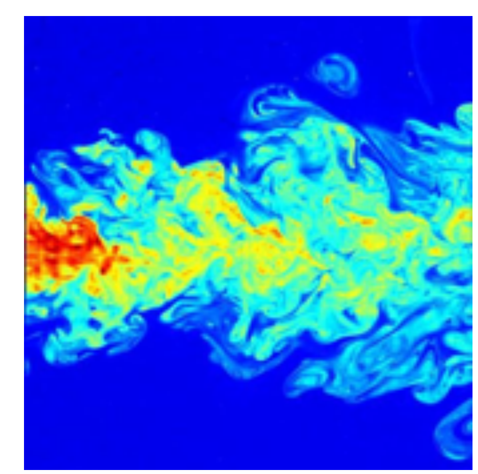
  where u is the fluid velocity, p is the fluid pressure, ρ is the fluid density, and μ is the viscoscity.



Laminar flow

- **Conservation of mass**, **momentum**, **energy** (relate to 2nd law of thermodynamics).
- **Viscosity** is the measure of the fluid's resistance to deformation, from shear or tensile stress.
  (A stress tensor with 9 degrees of freedom!)
- Flow can be **steady** (time derivative $\frac{\partial \rho}{\partial t} = 0$) or **unsteady** (or **transient,** i.e. high time derivative)
- Also **laminar** (flows in predictable, parallel layers) or **turbulent** (eddies, vortices, random chaos).
- **Reynolds number** indicates the turbulence of flow = inertial forces / viscous forces.

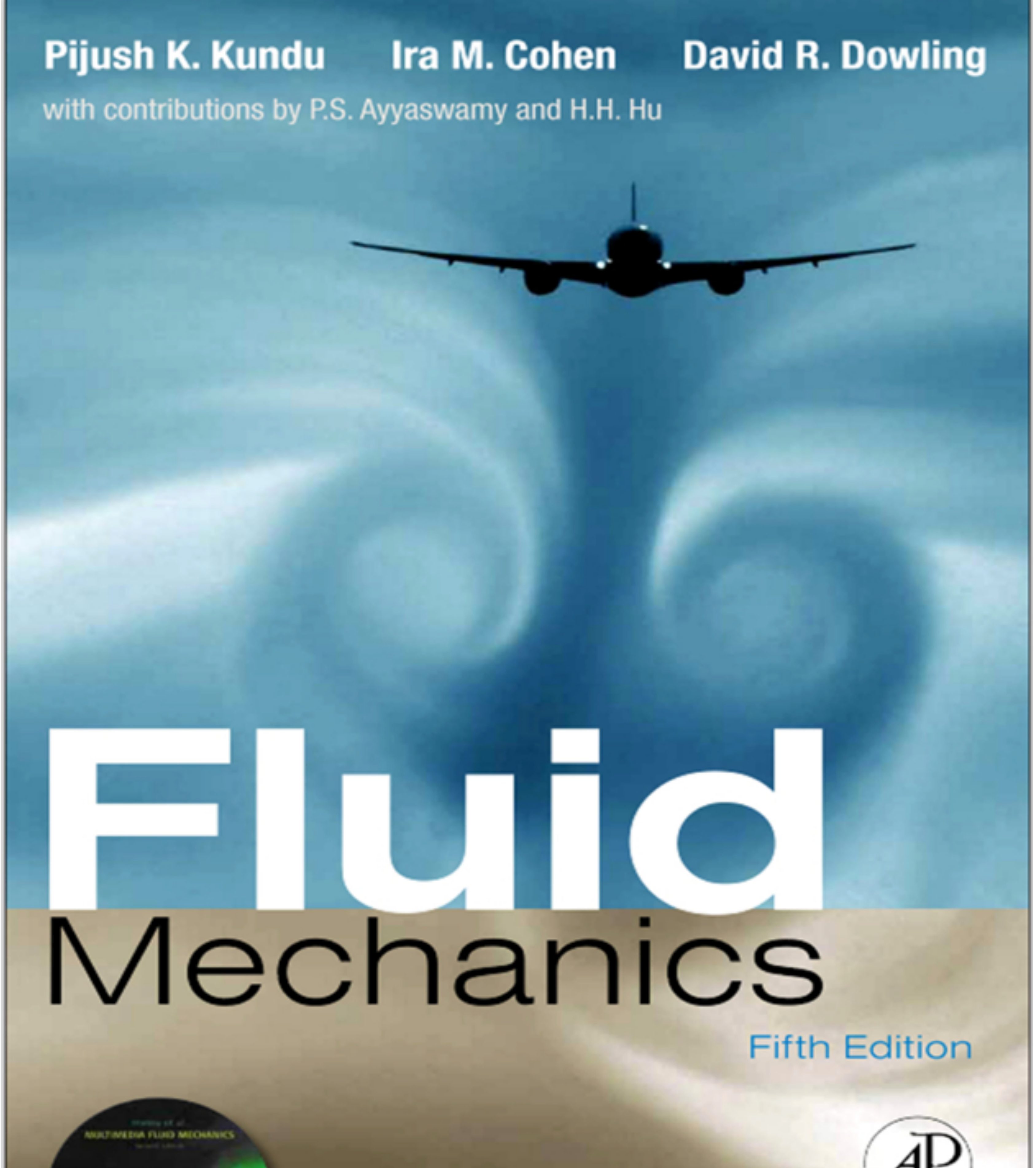
$$\mathrm{Re} = \frac{\text{inertial forces}}{\text{viscous forces}} = \frac{\rho\mathbf{v}L}{\mu} = \frac{\mathbf{v}L}{\nu}$$



Turbulent flow

https://www.comsol.com/multiphysics/navier-stokes-equations
https://en.wikipedia.org/wiki/Navier–Stokes_equations
https://en.wikipedia.org/wiki/Fluid_dynamics
https://en.wikipedia.org/wiki/Chaos_theory

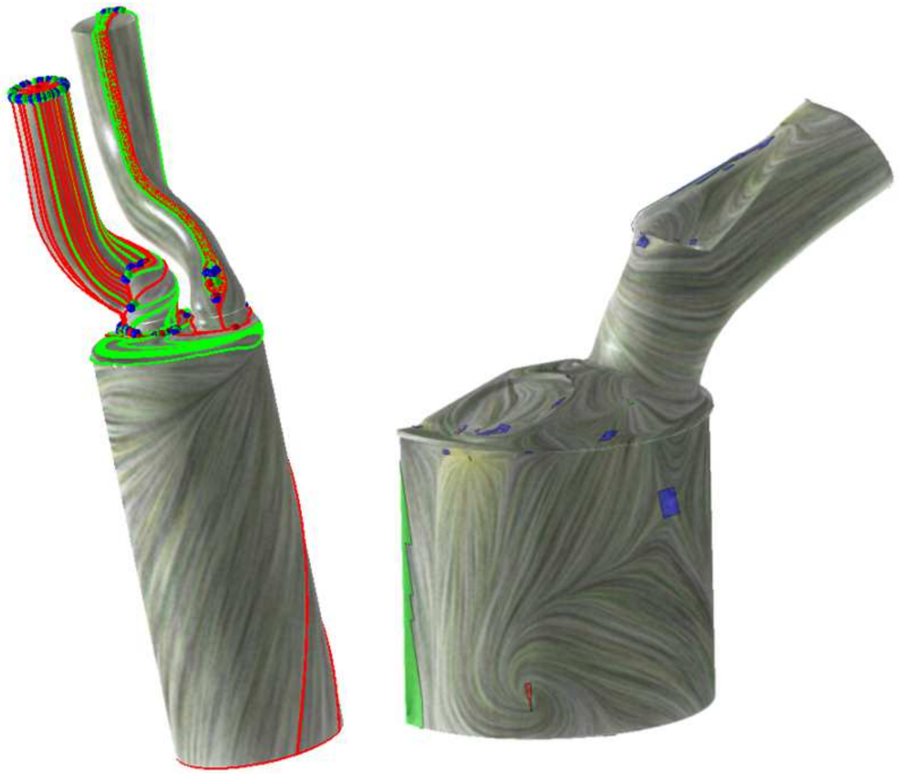
www.sci.utah.edu



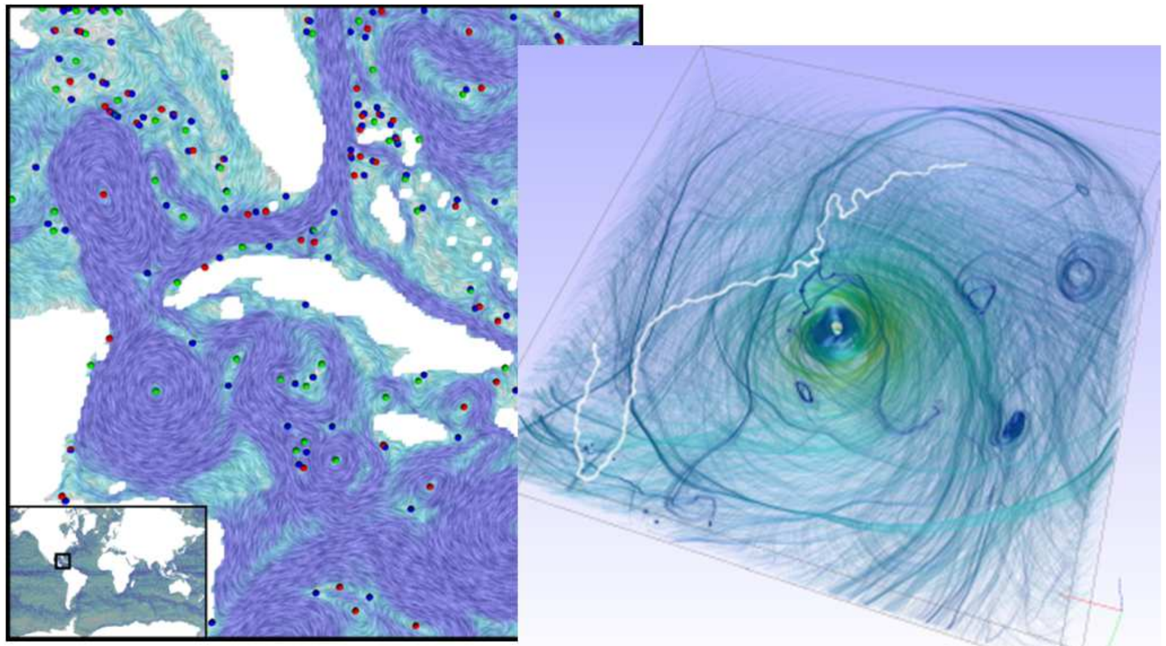
http://www3.nd.edu/~fthomas/Kundu_Fluid_Mechanics.pdf

www.sci.utah.edu

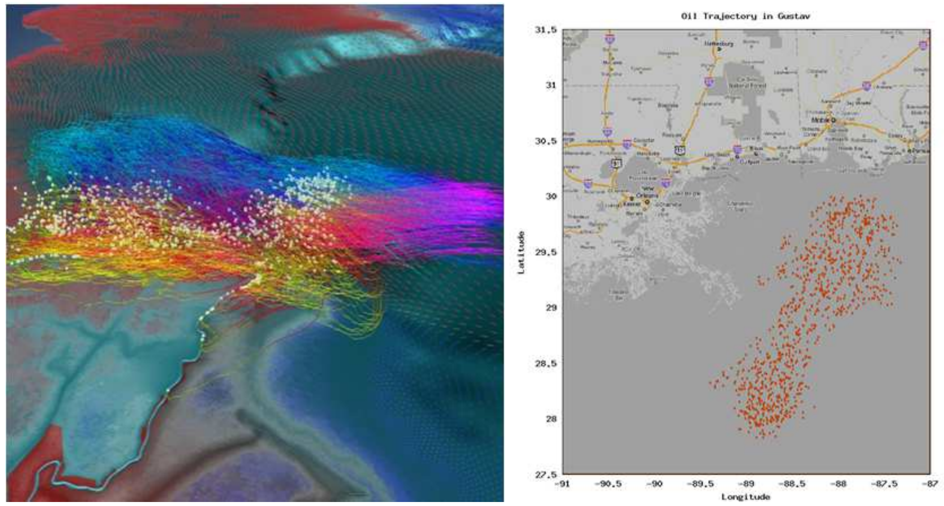
Vector Fields in Engineering and Science



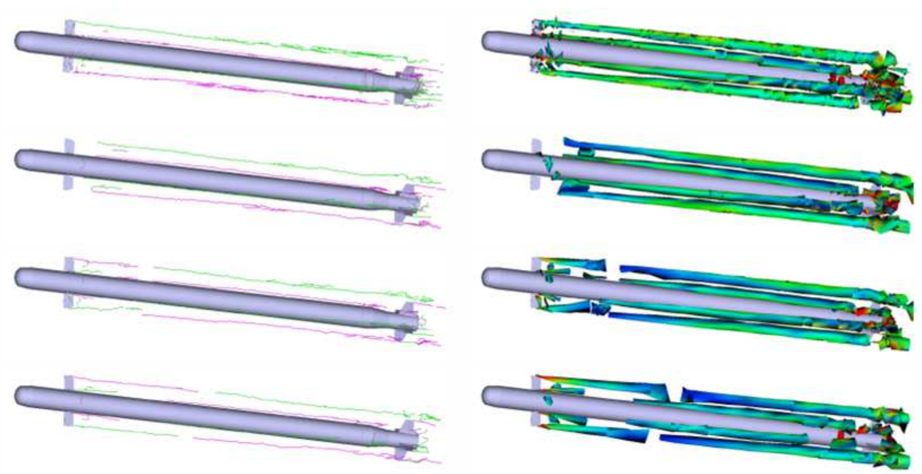
Automotive design
[**Chen** et al. TVCG07,TVCG08]



Weather study [Bhatia and Chen et al. TVCG11]

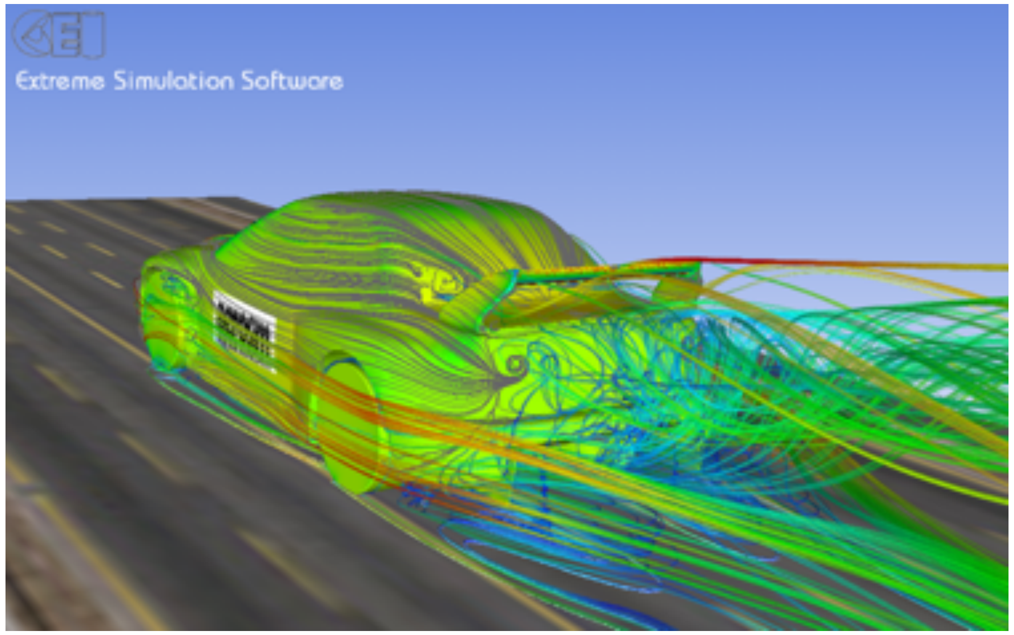


Oil spill trajectories [Tao et al. EMI2010]



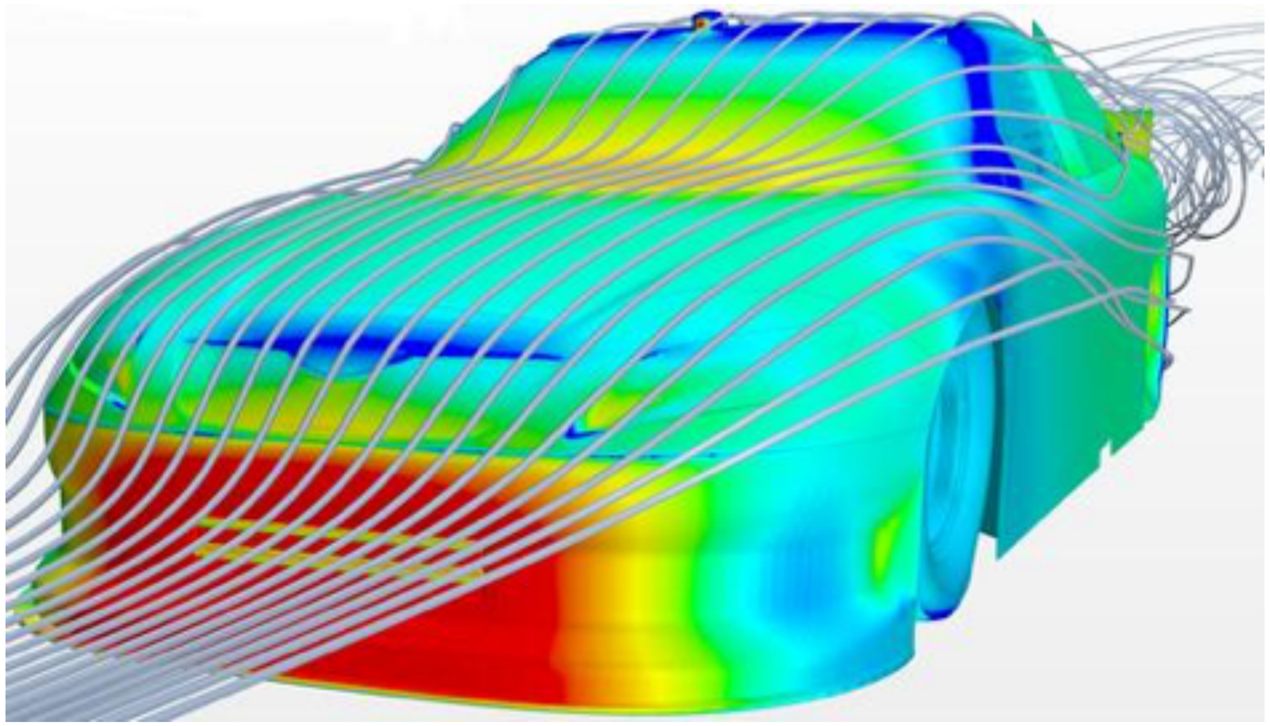
Aerodynamics around missiles [Kelly et al. Vis06]

Automotive body CFD simulations

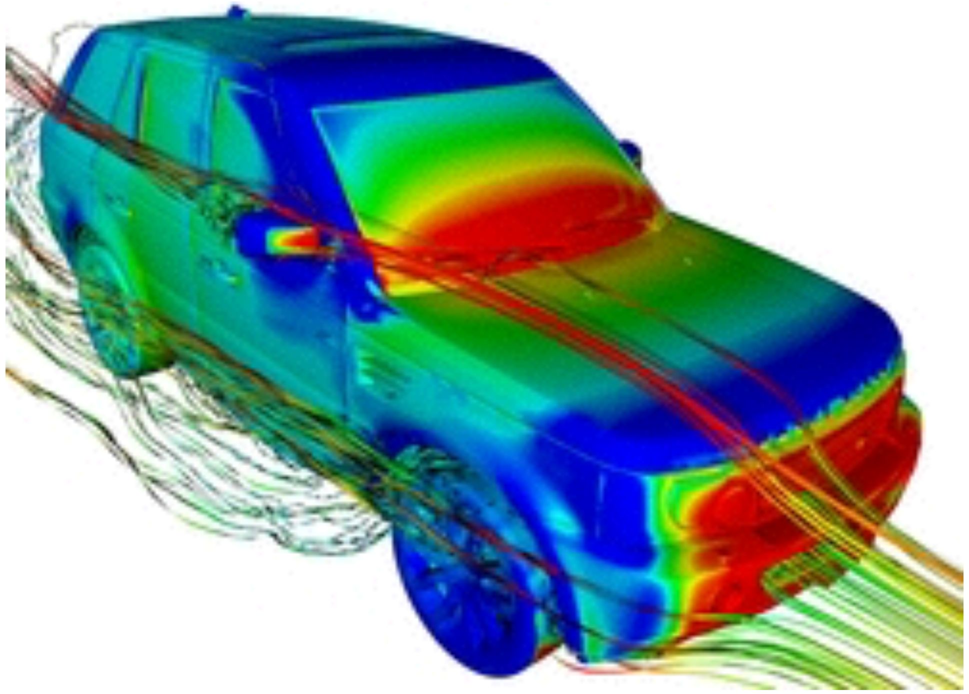


Flow visualization in Ensight
http://gallery.ensight.com/keyword/external%20aero;simulation/

F1 RANS simulation
http://www.symscape.com/blog/car-design-cfd

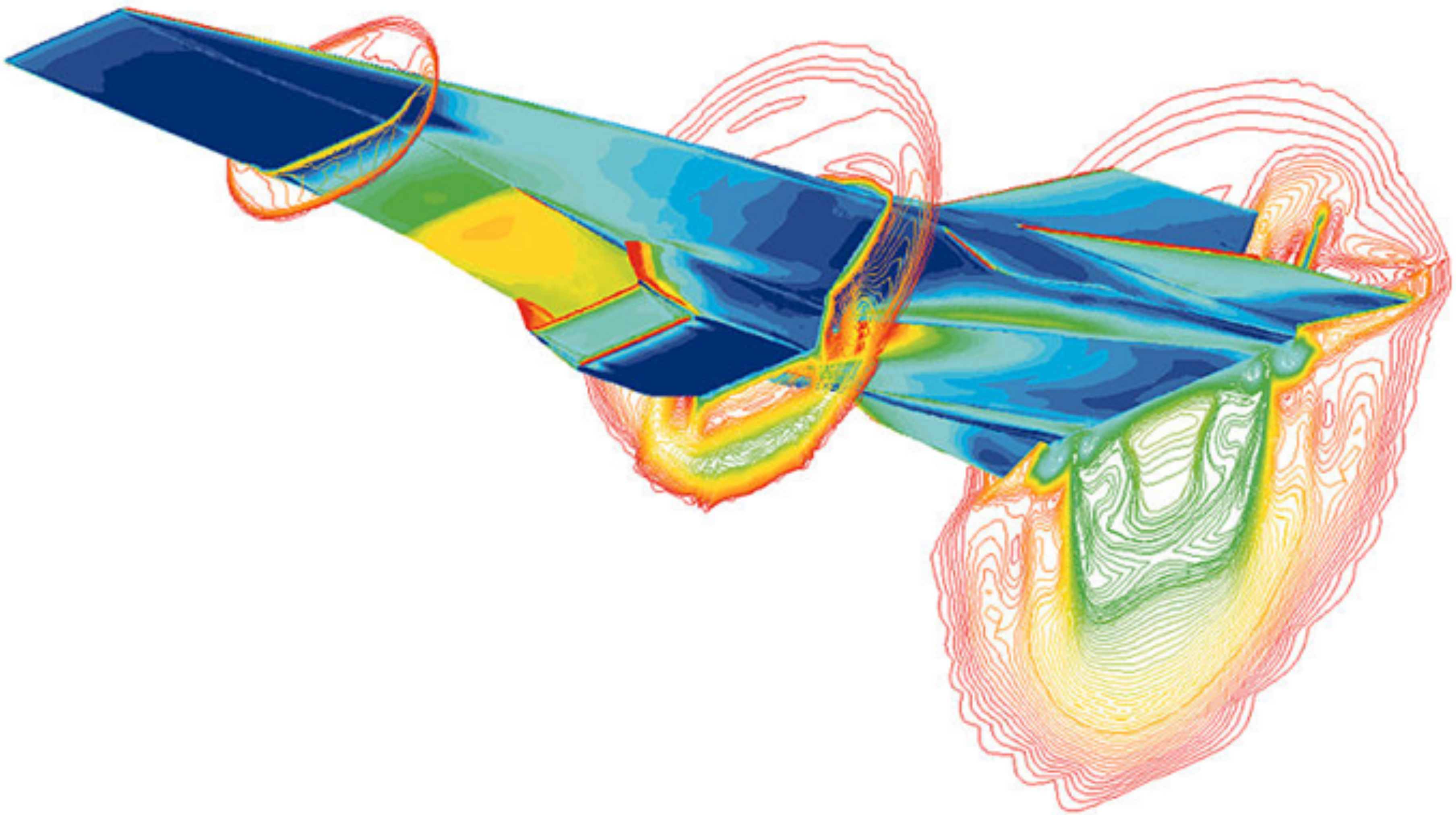


Michael Waltrip NASCAR
flow analysis in CD-adapco Star-CCM CFD tools

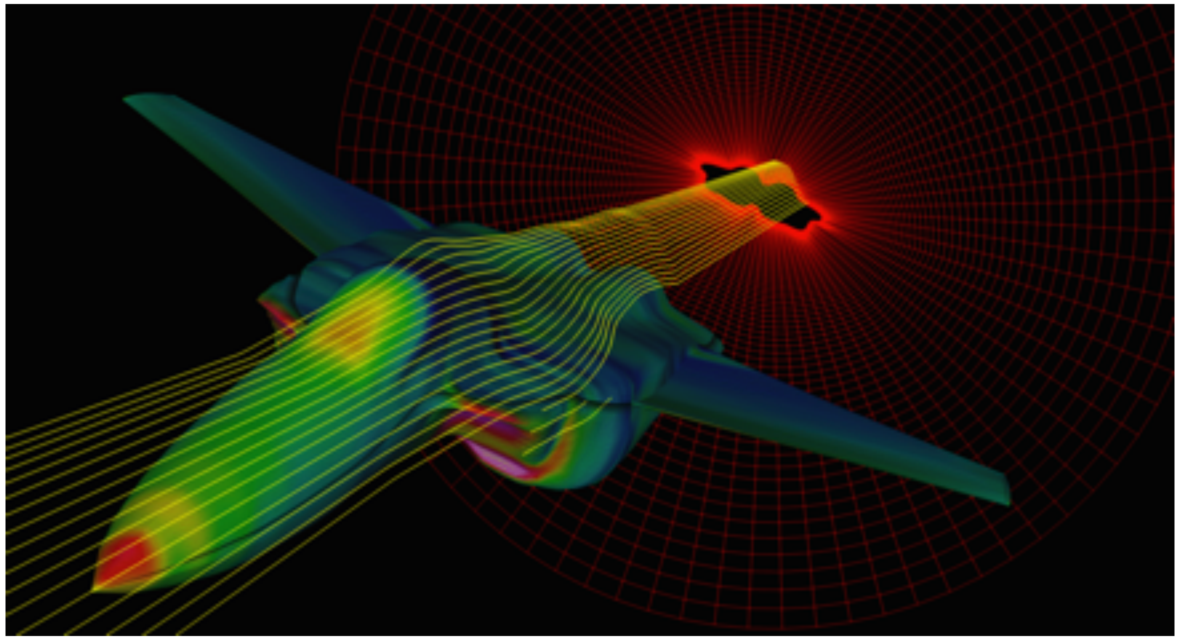


Jaguar Land Rover External Aerodynamic Simulation by Exa's PowerFLOW Software

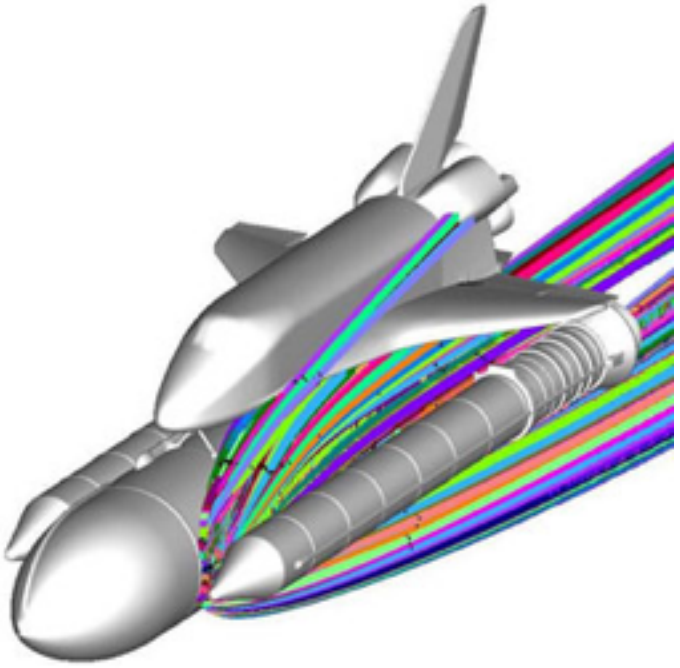
Aerospace



A simulation of the Hyper-X scramjet vehicle in operation at Mach-7. http://www.airports-worldwide.com/articles/article0523.php



FAST, http://www.openchannelfoundation.org



http://www.cesc.zju.edu.cn/learningcenter.htm

Flow visualization

Approaches to flow vis

- “How?”
  - **Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)**
  - Texture-based (LIC, spot noise)
  - Direct + geometry-based (hedehogs, glyphs)
  - Direct + heuristic (magnitude, Laplacian, FTLE)
  - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- “Where?”
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

Characteristic Curves of a Vector Field

- **Streamlines:** curve parallel (tangent) to the vector field in each point for a fixed time
- **Pathlines:** describes motion of a particles over time through a vector field
- **Streaklines:** trace of dye that is released into the flow at a fixed position
- **Timelines**: describes motion of particles set out on a line over time through a vector field

Characteristic Curves of a Vector Field

- **Streamlines:** curve parallel (tangent) to the vector field in each point for a fixed time

$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u))\ du$$

integrate over space
the "continuous" static velocity field

- **Pathlines:** describes motion of a particles over time through a vector field

- **Streaklines:** trace of dye that is released into the flow at a fixed position

- **Timelines**: describes motion of particles set out on a line over time through a vector field

Characteristic Curves of a Vector Field

- **Streamlines:** curve parallel (tangent) to the vector field in each point for a fixed time

$$s(t) = s_0 + \int_{0 \le u \le t} V(s(u))\ du$$

integrate over space
the “continuous” static velocity field

- **Pathlines:** describes motion of a particles over time through a vector field

$$s(t) = s_0 + \int_{0 \le u \le t} V(s(u), u)\ du$$

integrate over time and space
each point is like a new seed

- **Streaklines:** trace of dye that is released into the flow at a fixed position

- **Timelines**: describes motion of particles set out on a line over time through a vector field

Characteristic Curves of a Vector Field

- **Streamlines:** curve parallel (tangent) to the vector field in each point for a fixed time

$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u))\ du$$

integrate over space
the "continuous" static velocity field

- **Pathlines:** describes motion of a particles over time through a vector field

$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u), u)\ du$$

integrate over time and space
each point is like a new seed

- **Streaklines:** trace of dye that is released into the flow at a fixed position

$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u), u)\ du$$

integrate over time and space
seed(s) stay in the same place

- **Timelines**: describes motion of particles set out on a line over time through a vector field

Characteristic Curves of a Vector Field

- **Streamlines:** curve parallel (tangent) to the vector field in each point for a fixed time

$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u))\ du$$

integrate over space
the "continuous" static velocity field

- **Pathlines:** describes motion of a particles over time through a vector field

$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u), u)\ du$$

integrate over time and space
each point is like a new seed

- **Streaklines:** trace of dye that is released into the flow at a fixed position

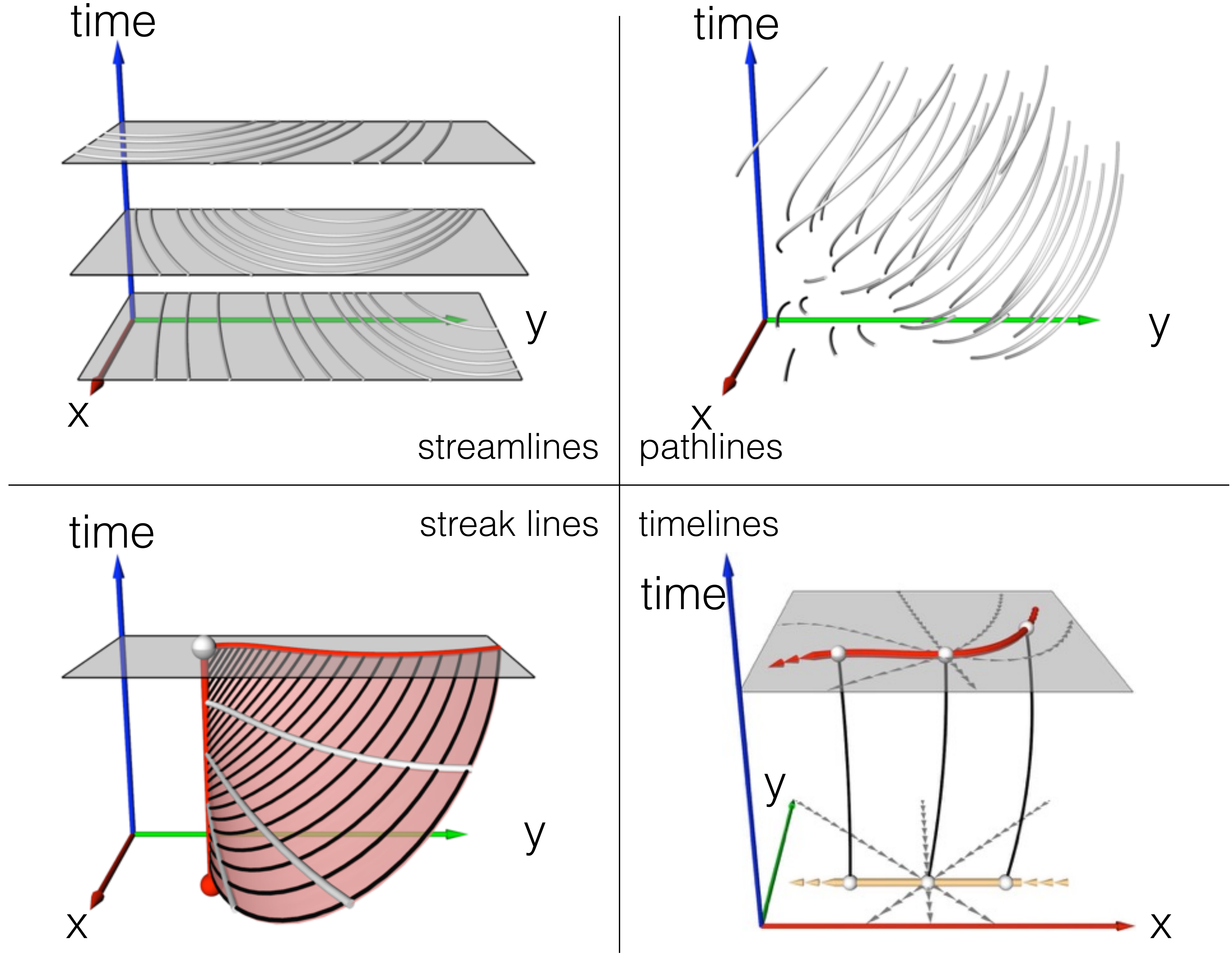
$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u), u)\ du$$

integrate over time and space
seed(s) stay in the same place

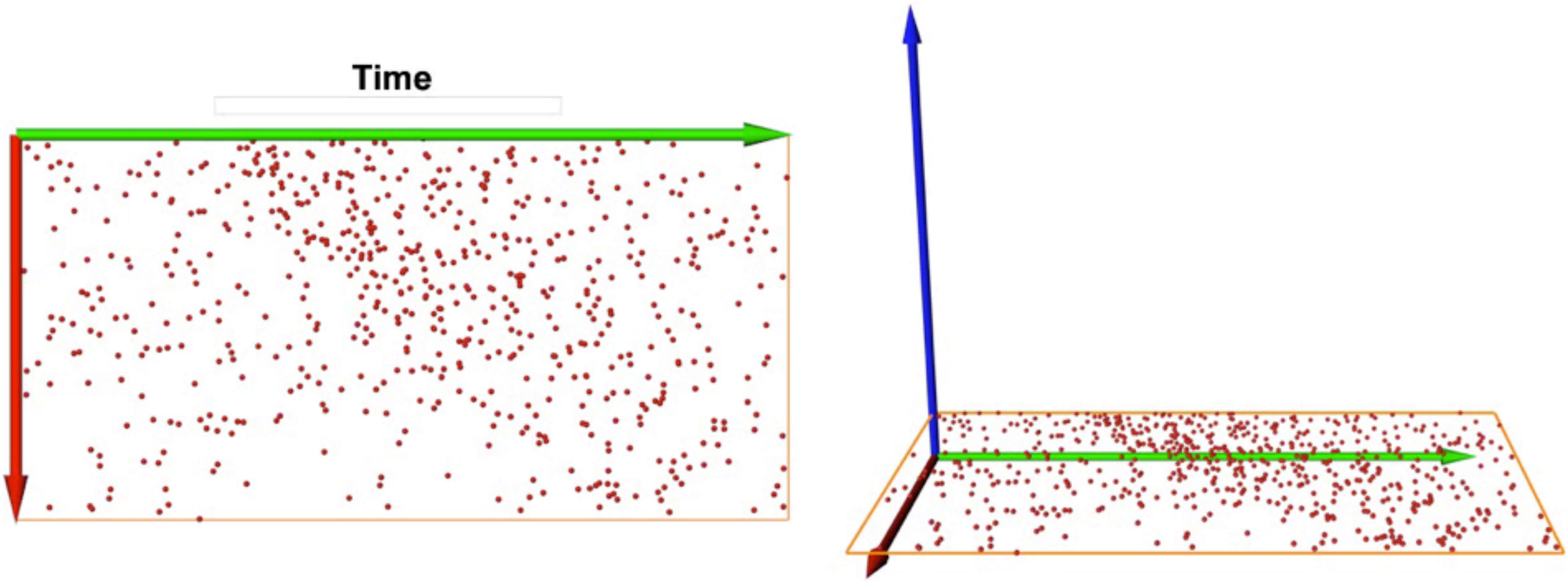
- **Timelines**: describes motion of particles set out on a line over time through a vector field

$$s(t) = s_0 + \int_{0 \leq u \leq t} V(s(u), u)\ du$$

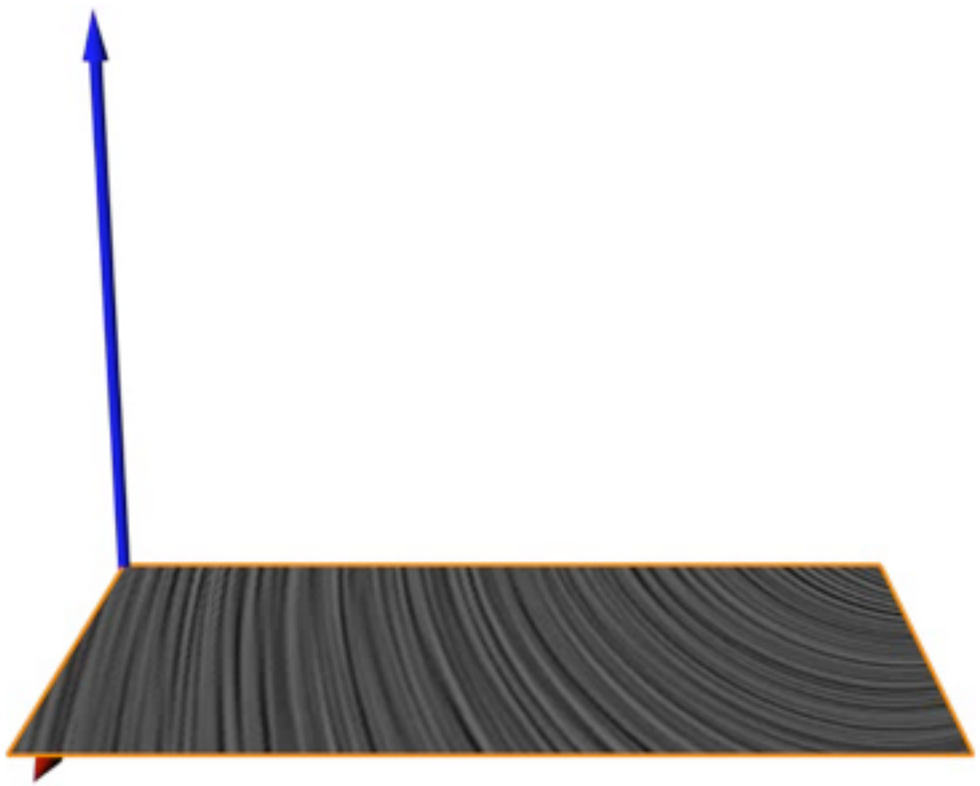
same as streaklines, but a "burst" in time

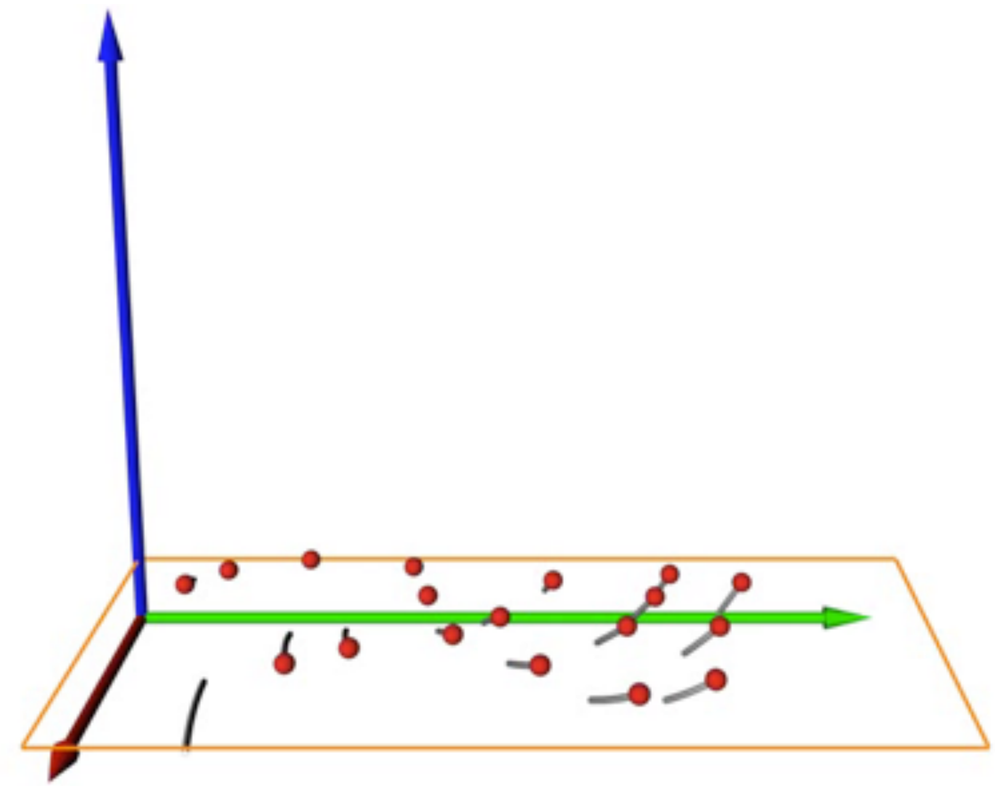


time
y
x
streamlines
time
y
x
pathlines
time
streak lines
y
x
timelines
time
y
x



2D time-dependent vector field
particle visualization





streamlines | pathlines

curve parallel to the vector field in each point for a **fixed time**

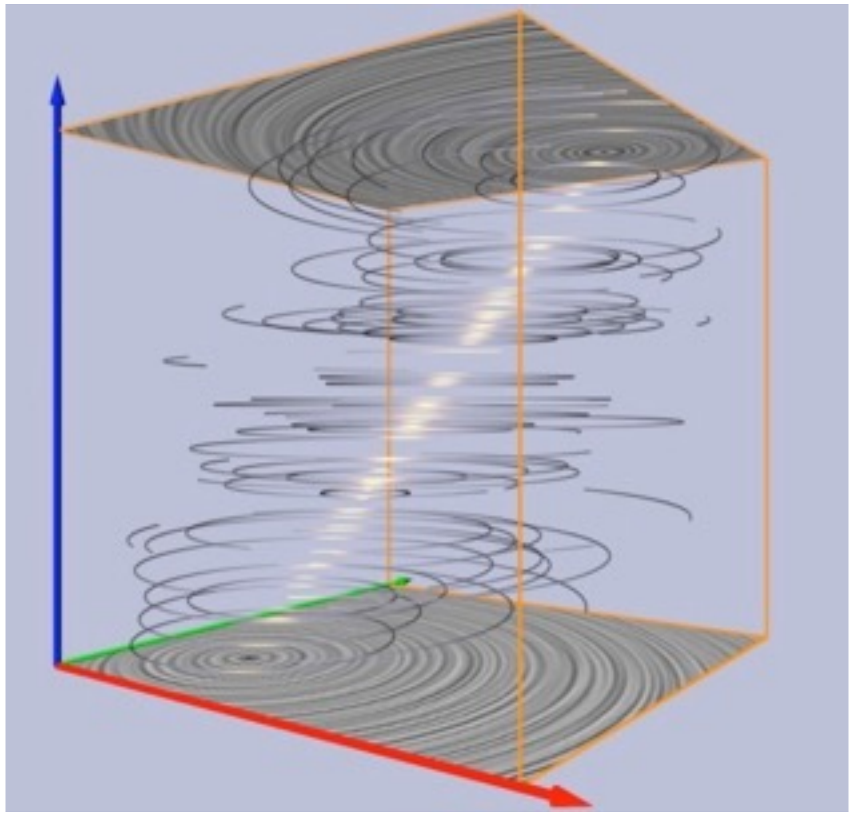
describes motion of a massless particle in an **steady** flow field

$$\mathbf{s}(t) = \mathbf{s}_0 + \int_{0\le u\le t} V(\mathbf{s}(u))\, du$$

curve parallel to the vector field in each point **over time**

describes motion of a massless particle in an **unsteady** flow field

$$\mathbf{s}(t) = \mathbf{s}_0 + \int_{0\le u\le t} V(\mathbf{s}(u), \mathbf{u})\, du$$

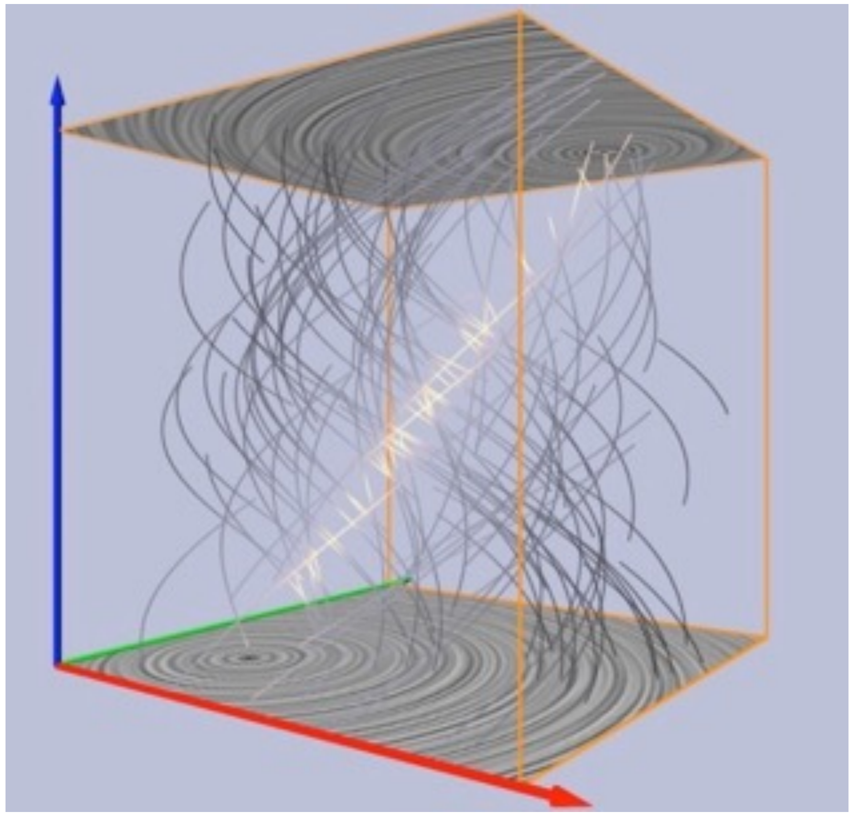


streamlines

curve parallel to the vector field in each point for a **fixed time**

describes motion of a massless particle in an **steady** flow field

$$s(t) = s_0 + \int_{0\leq u\leq t} V(s(u))\, du$$



pathlines

curve parallel to the vector field in each point **over time**

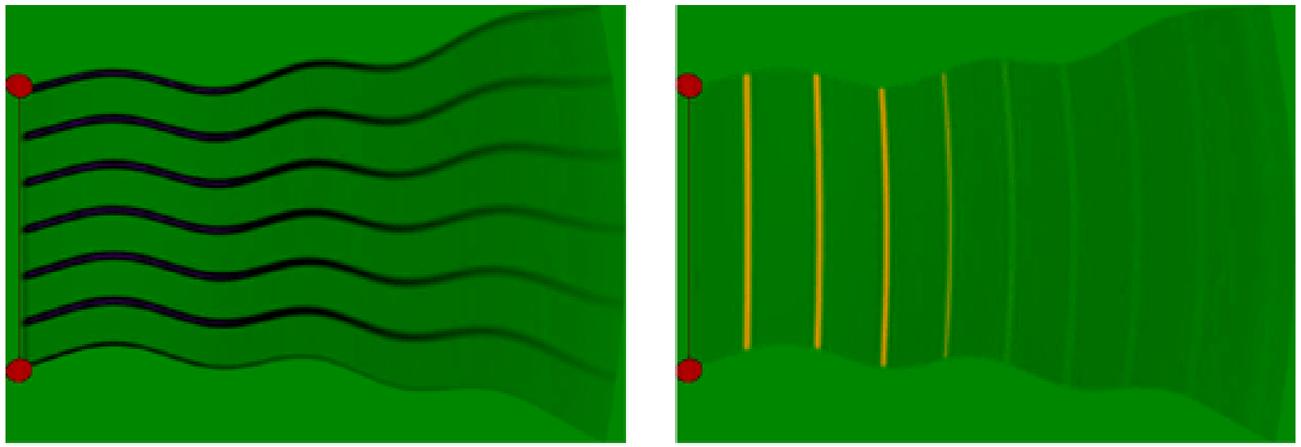
describes motion of a massless particle in an **unsteady** flow field

$$s(t) = s_0 + \int_{0\leq u\leq t} V(s(u), u)\, du$$

Weinkauf and Theisel, TVCG 2010

Other feature curve

- **Timelines**
  - Union of the current positions of particles released at the same time in space



(a) Coloring fixed rows in the array reveals streak lines.

(b) Coloring fixed columns in the array reveals time lines.

Source: doi.ieeecomputersociety.org

- **Stream and Path lines:**
  - Through all non-critical points (**x**,t) in space-time there is exactly one stream/path line passing through it.

- **Streak and Time lines:**
  - Many streak/time lines through every point (of the spatial domain)
  - ➔ makes it difficult to describe streak/time lines as tangent curves of some vector field
    - But it is possible. We may discuss it in a later session.

- Stream, Path, and Streak lines in a steady vector field.

?

- **Stream and Path lines:**
  - Through all non-critical points (**x**,t) in space-time there is exactly one stream/path line passing through it.

- **Streak and Time lines:**
  - Many streak/time lines through every point (of the spatial domain)
  - ➔ makes it difficult to describe streak/time lines as tangent curves of some vector field
    - But it is possible. We may discuss it in a later session.

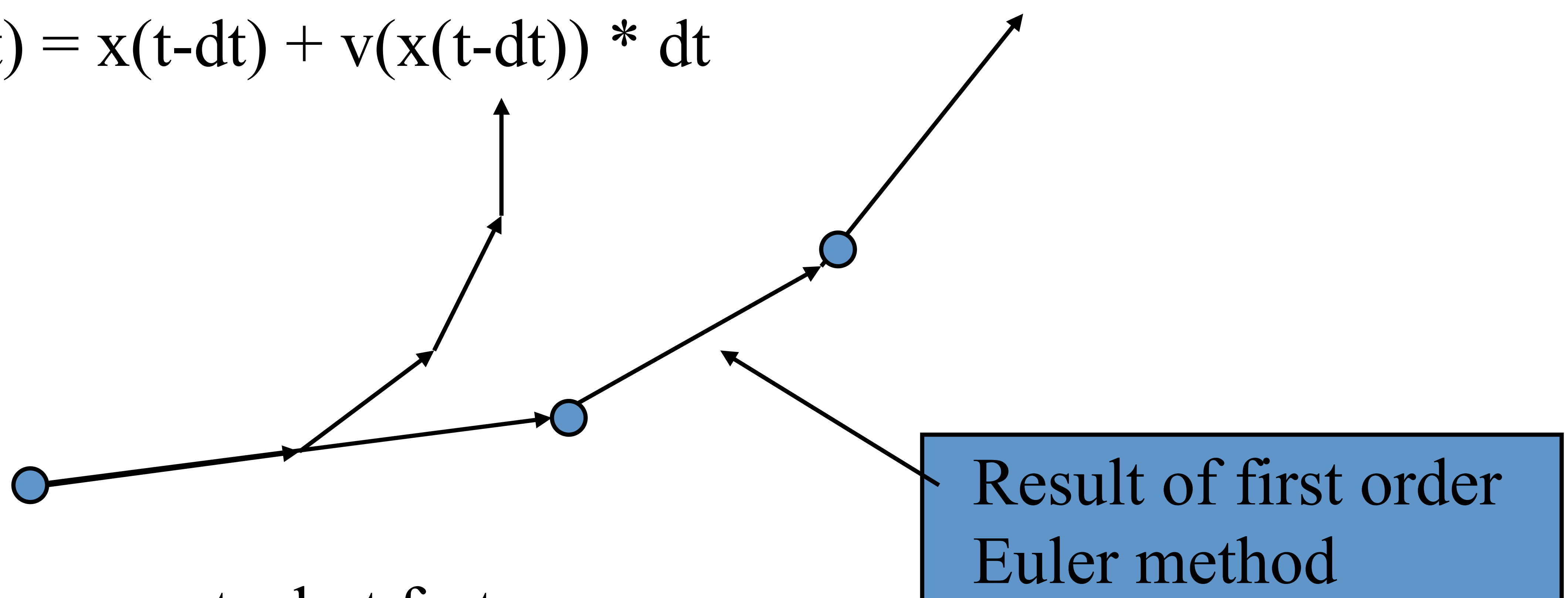
- Stream, Path, and Streak lines coincide in a steady vector field.

Integration Techniques

Numerical Integration

First Order Euler method:

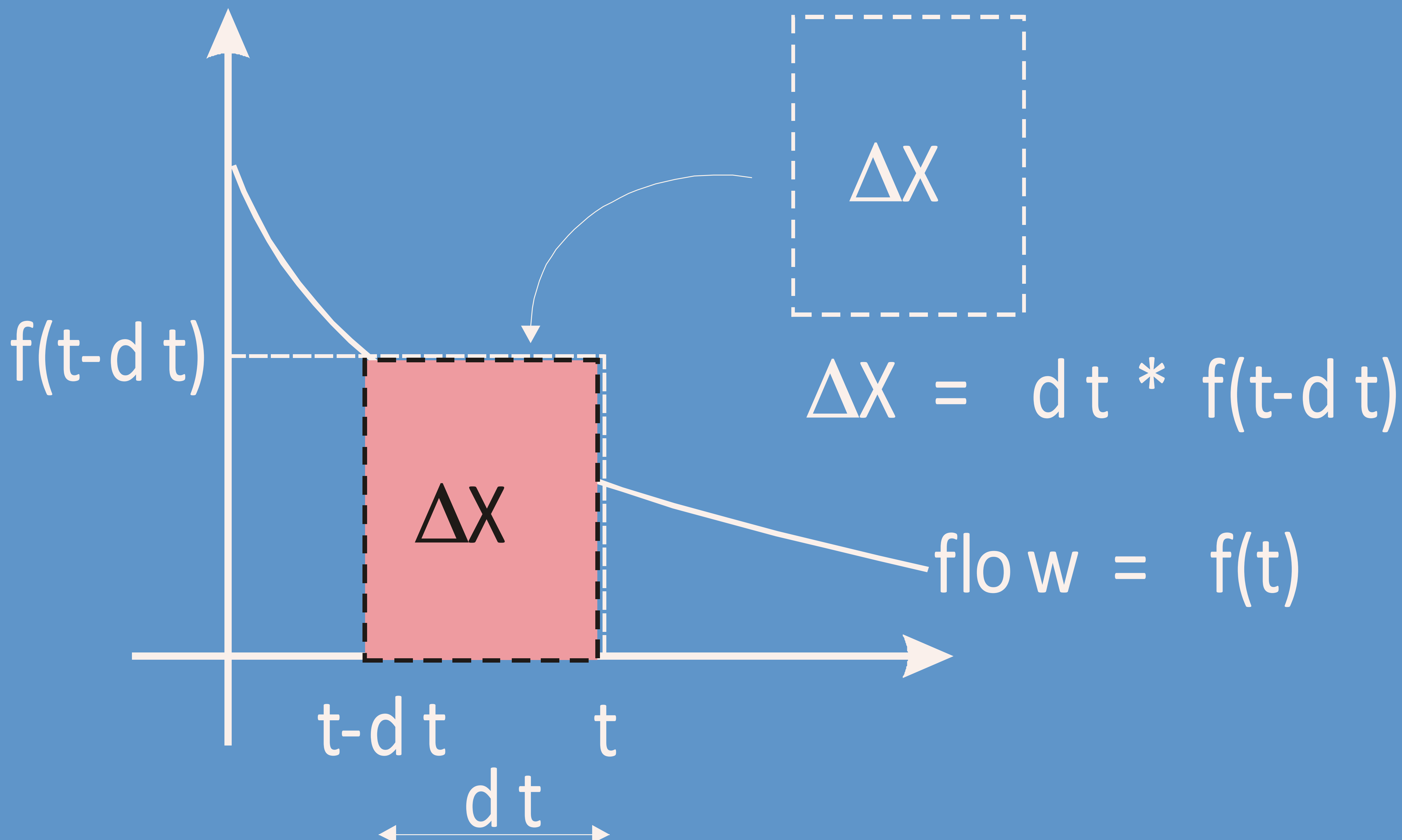
$$x(t) = x(t-dt) + v(x(t-dt)) * dt$$



- Not very accurate, but fast
- Other higher order methods are avilable: Runge-Kutta second and fourth order integration methods (more popular due to their accuracy)

Euler's Method

Assume flow = f(t)



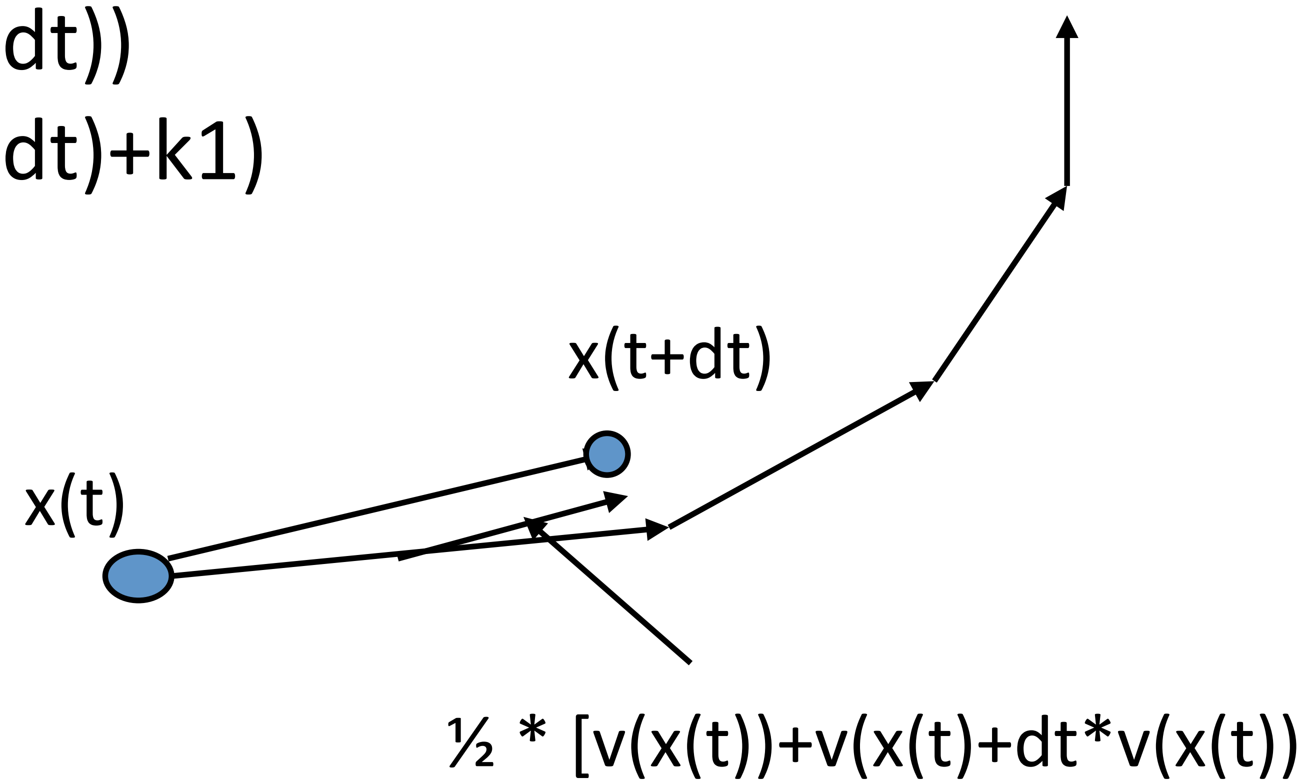
Numerical Integration (2)

Second Runge-Kutta Method

x(t) = x(t-dt) + ½ * (K1 + K2)

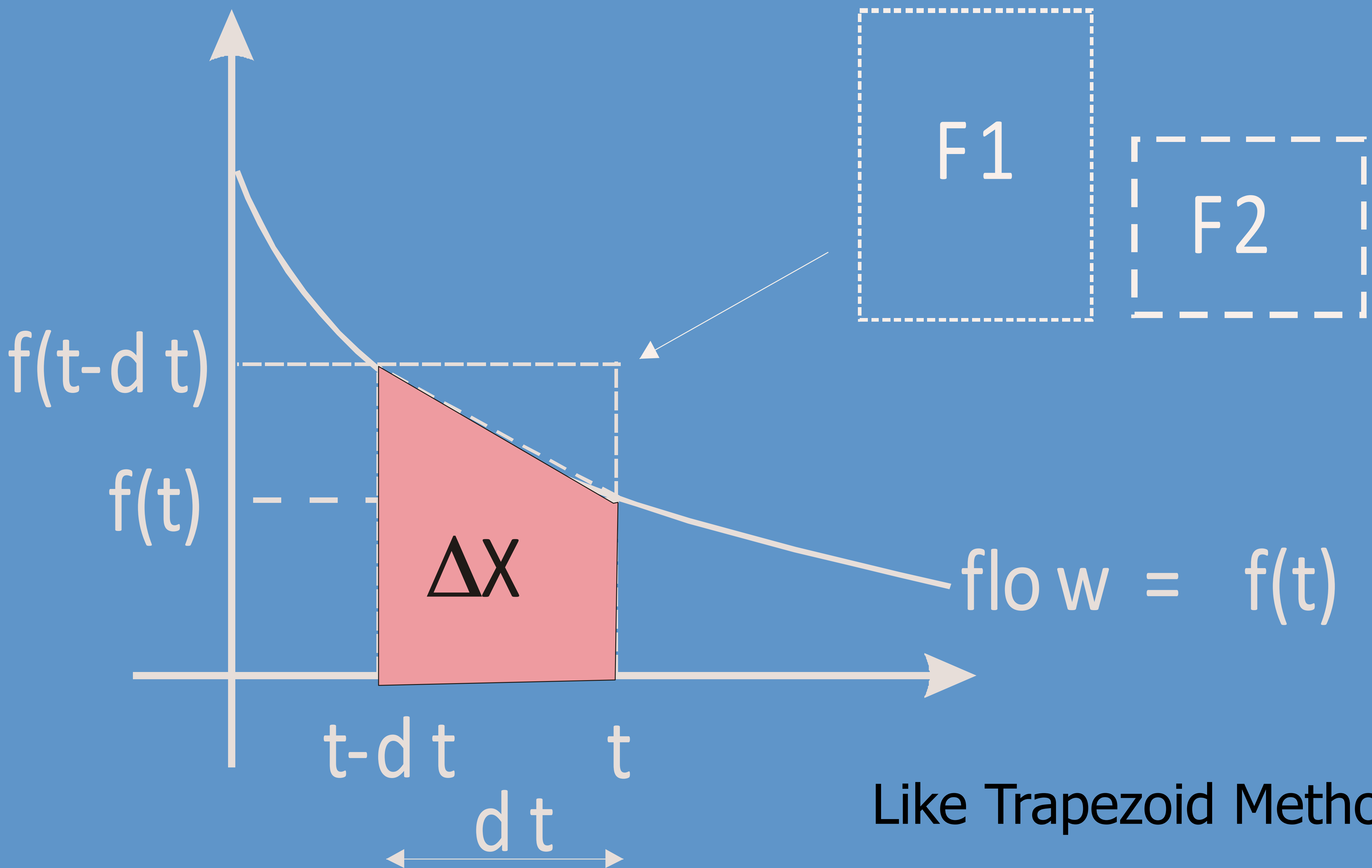
k1 = dt * v(x(t-dt))

k2 = dt * v(x(t-dt)+k1)



Runge-Kutta 2

Assume flow = f(t)



Like Trapezoid Method.

Numerical Integration (3)

Standard Method: Runge-Kutta fourth order

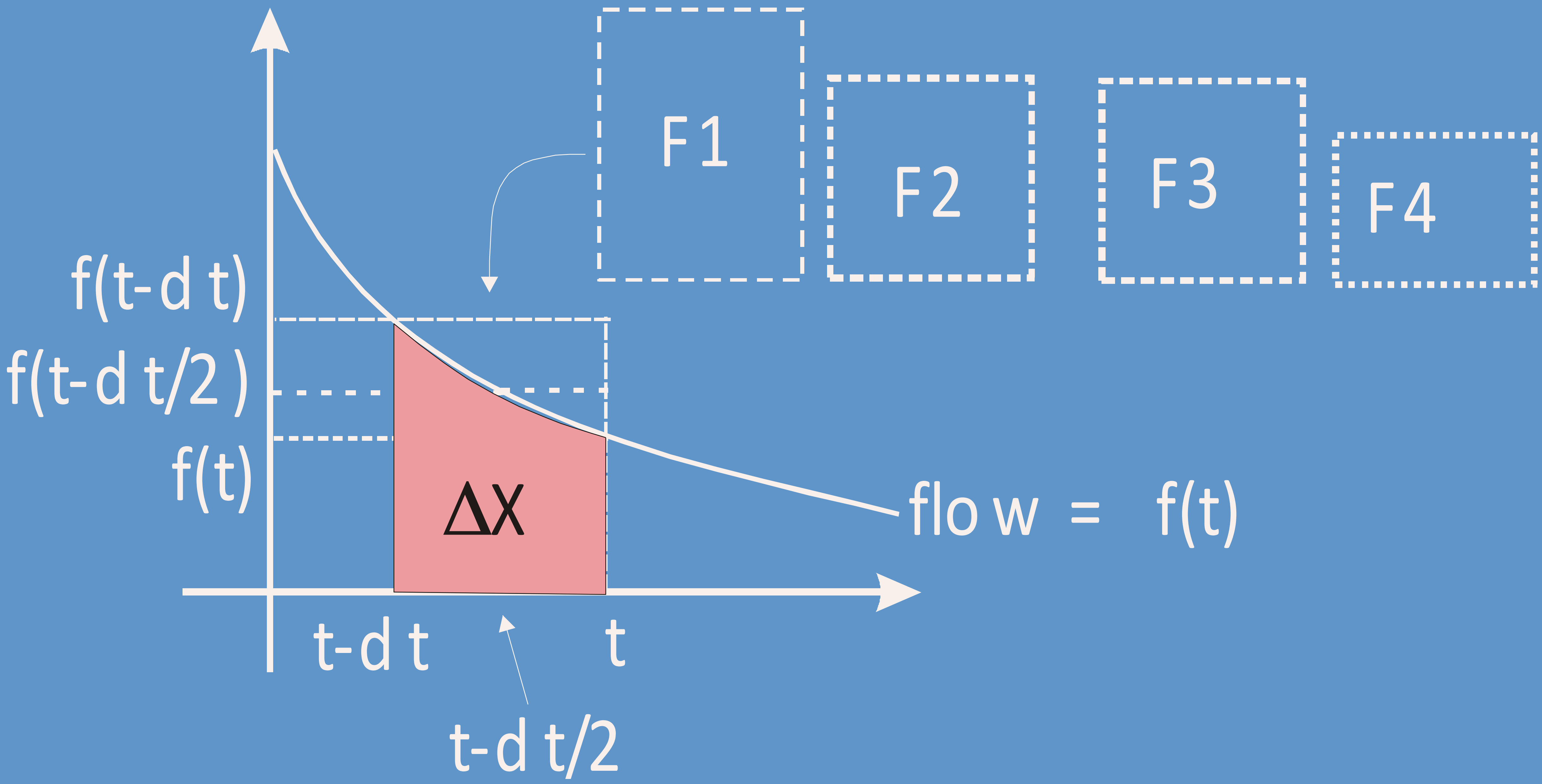
$$x(t) = x(t\text{-}dt) + 1/6\ (k_1 + 2k_2 + 2k_3 + k_4)$$

$k_1 = dt * v(t\text{-}dt)$; $k_2 = dt * v(x(t\text{-}dt) + k1/2)$

$k_3 = dt * v(x(t\text{-}dt) + k2/2)$; $k_4 = dt * v(x(t\text{-}dt) + k3)$

Runge-Kutta 4

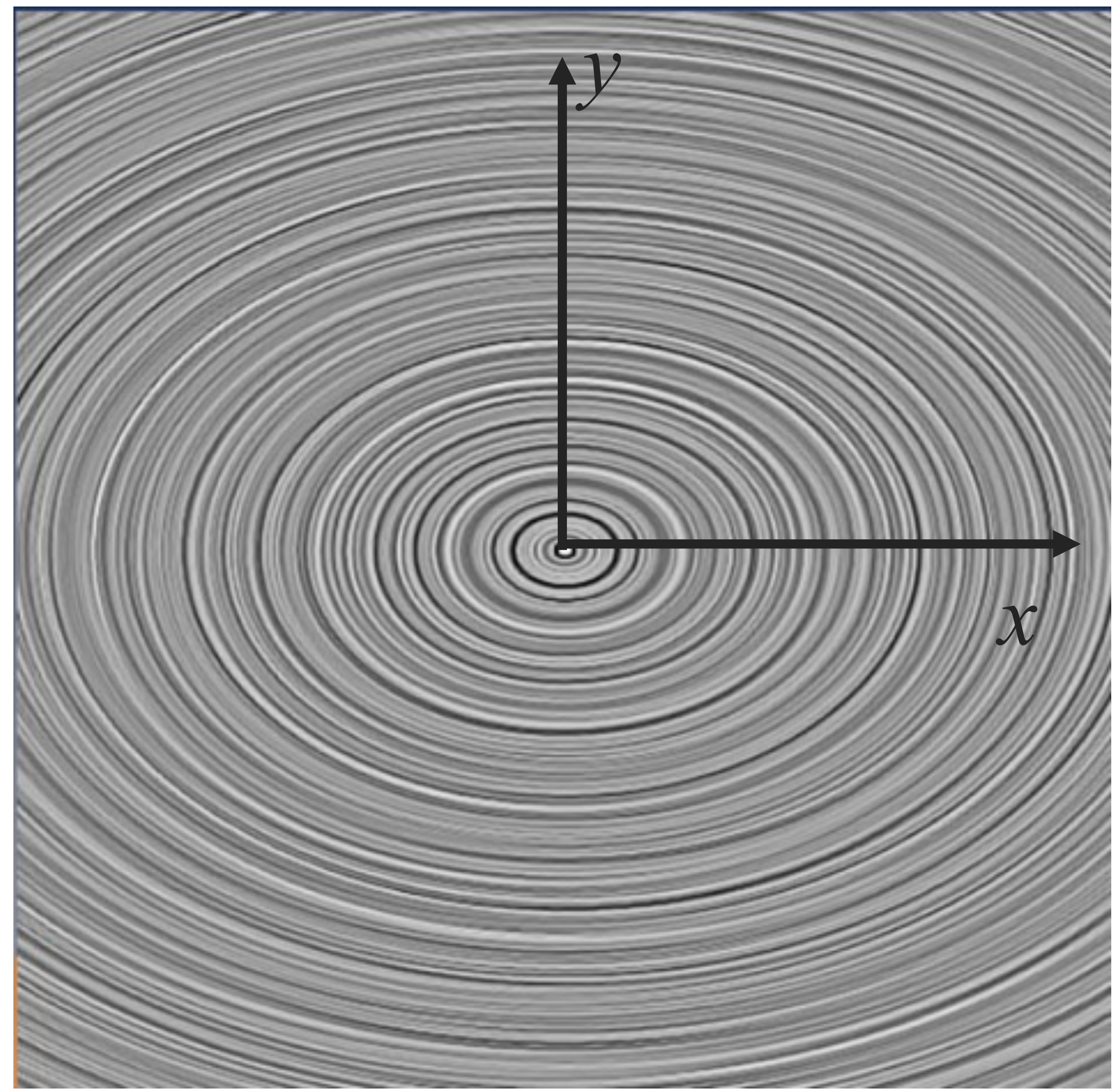
Assume flow = f(t)



- **Numerical integration of stream lines:**

- approximate streamline by polygon $\mathbf{x}_i$

- **Testing example:**
  - $\mathbf{v}(x,y) = (-y, x/2)^T$
  - exact solution: ellipses
  - starting integration from (0,-1)



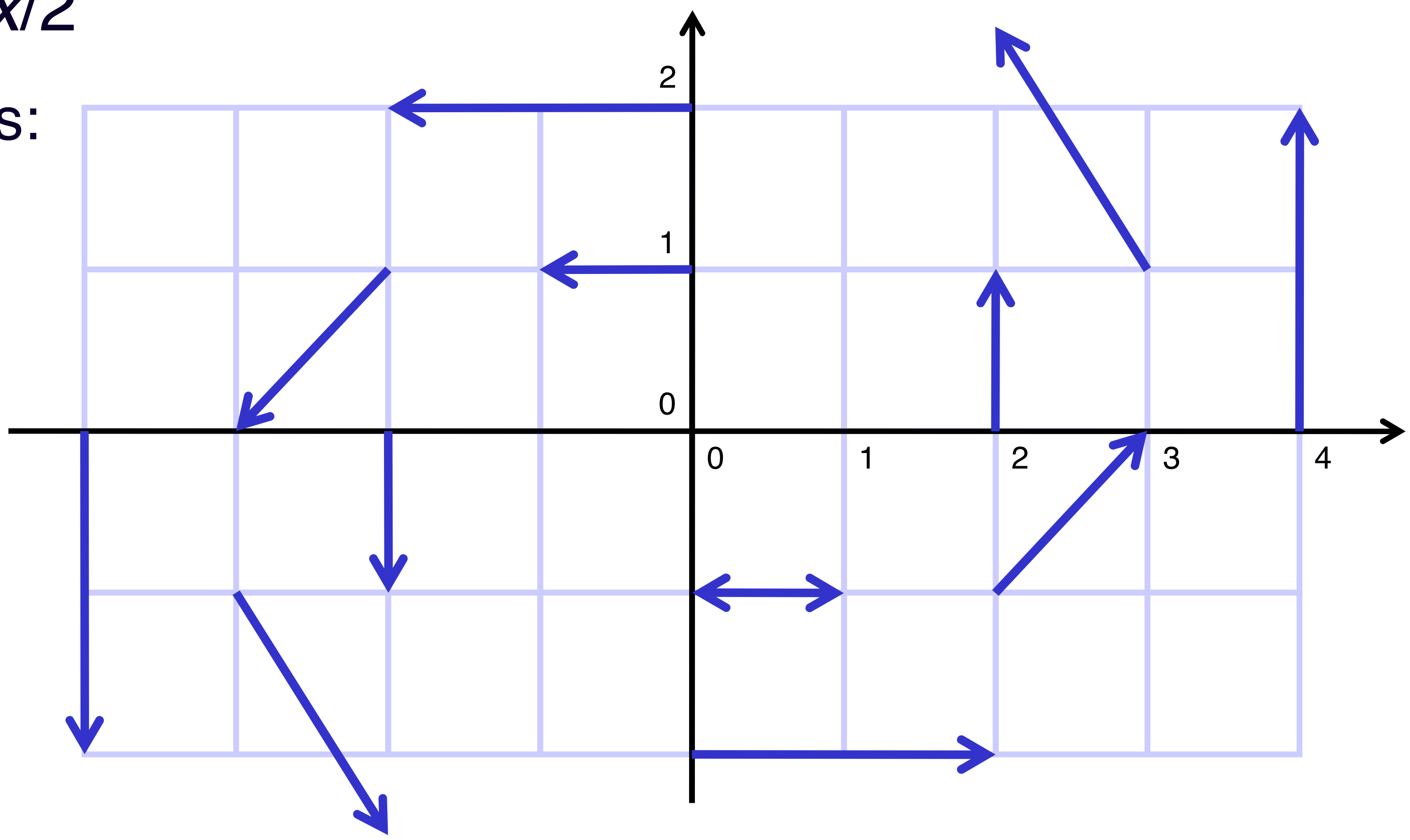
Euler Integration – Example

2D model data:

$v_x = \mathrm{d}x/\mathrm{d}t = -y$

$v_y = \mathrm{d}y/\mathrm{d}t = x/2$

Sample arrows:



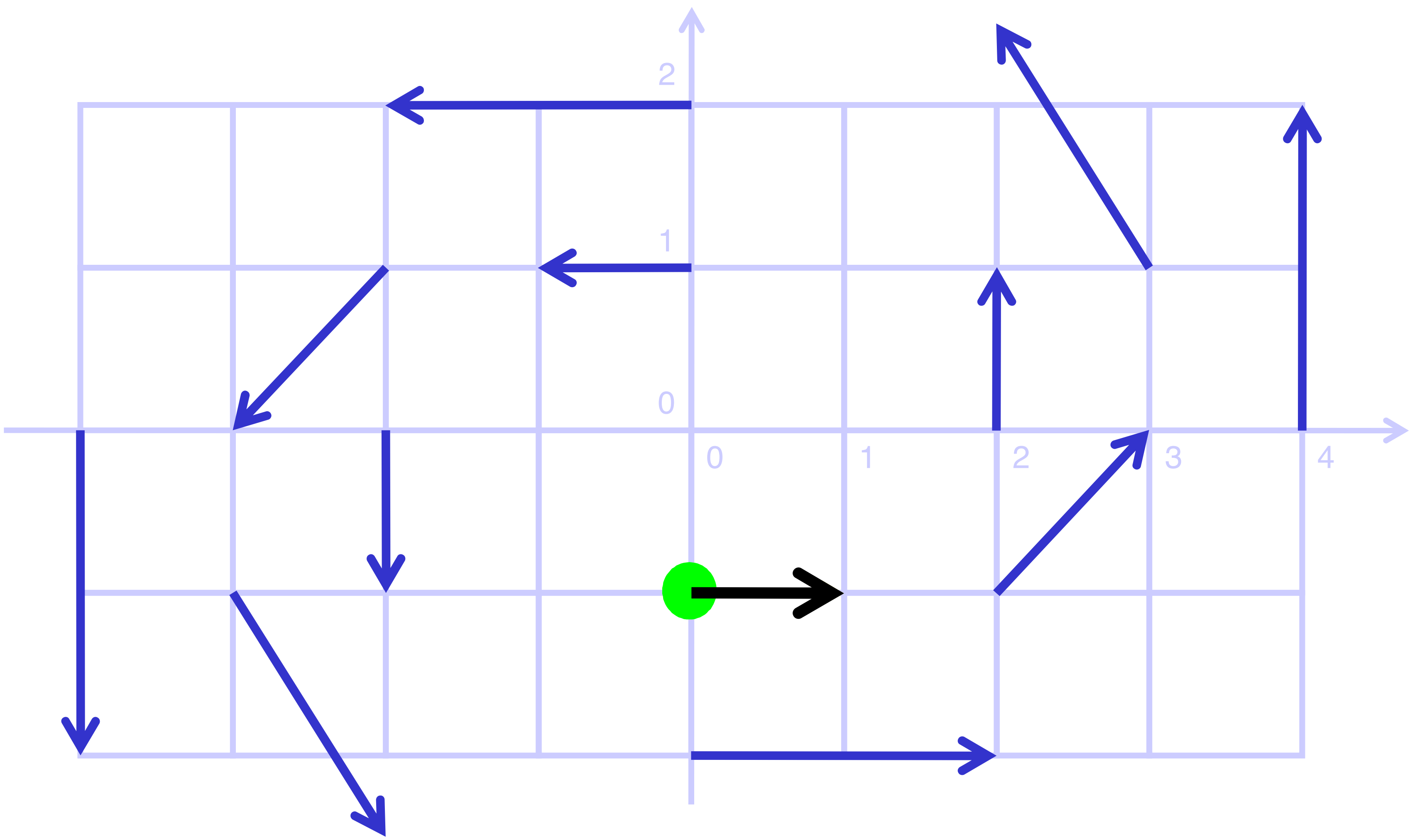
True solution: ellipses.

Euler Integration – Example

- Seed point $\mathbf{s}_0 = (0|-1)^T$;
current flow vector $\mathbf{v}(\mathbf{s}_0) = (1|0)^T$;
$\mathrm{d}t = ½$

$v_x = \mathrm{d}x/\mathrm{d}t = -y$

$v_y = dy/dt = x/2$

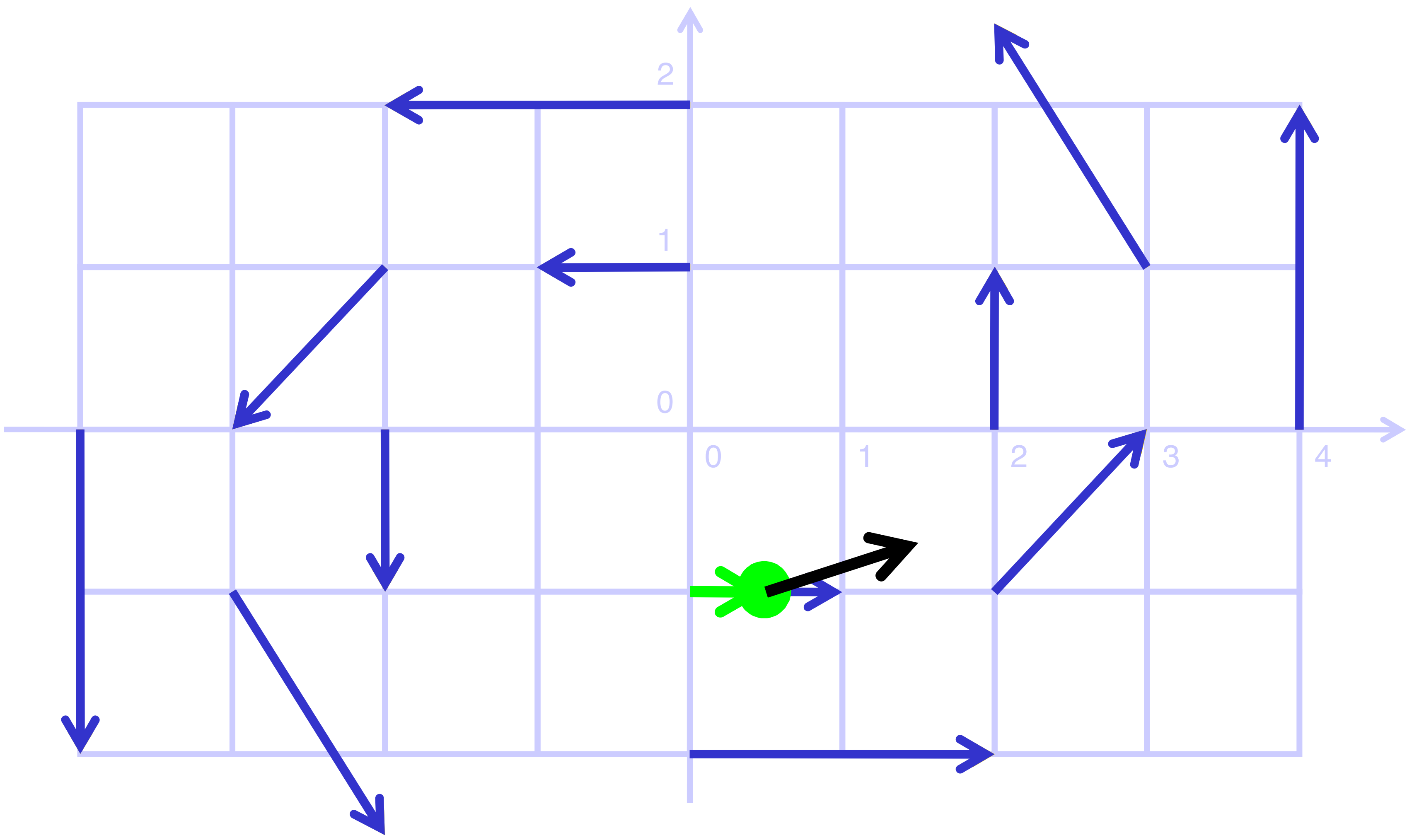


Euler Integration – Example

- New point $\mathbf{s}_1 = \mathbf{s}_0 + \mathbf{v}(\mathbf{s}_0) \cdot \mathrm{d}t = (1/2 | -1)^T$; current flow vector $\mathbf{v}(\mathbf{s}_1) = (1 | 1/4)^T$;

$v_x = \mathrm{d}x/\mathrm{d}t = -y$

$v_y = dy/dt = x/2$

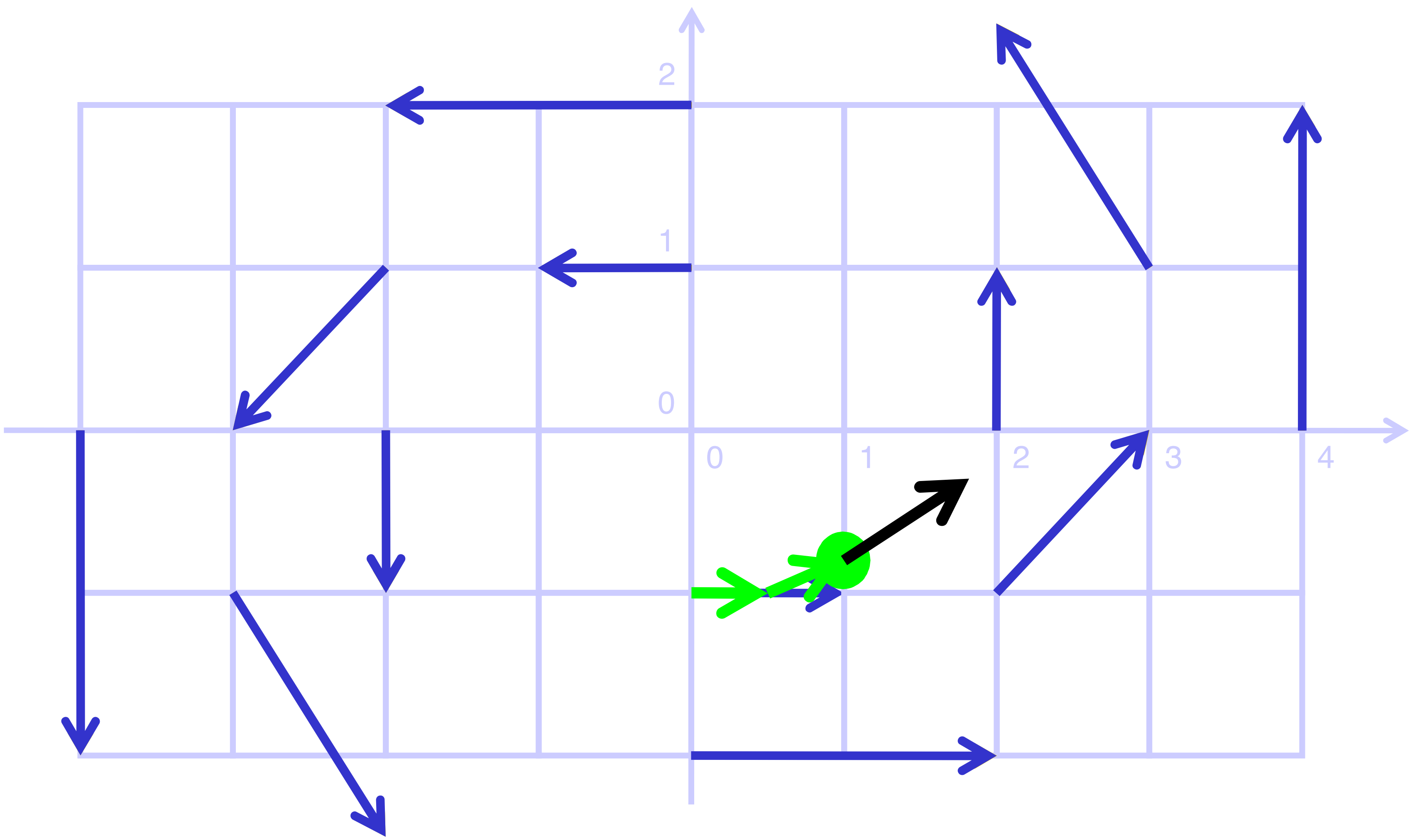


Euler Integration – Example

- New point $\mathbf{s}_2 = \mathbf{s}_1 + \mathbf{v}(\mathbf{s}_1) \cdot \mathrm{d}t = (1 \mid -7/8)^T$; current flow vector $\mathbf{v}(\mathbf{s}_2) = (7/8 \mid 1/2)^T$;

$v_x = \mathrm{d}x/\mathrm{d}t = -y$

$v_y = dy/dt = x/2$



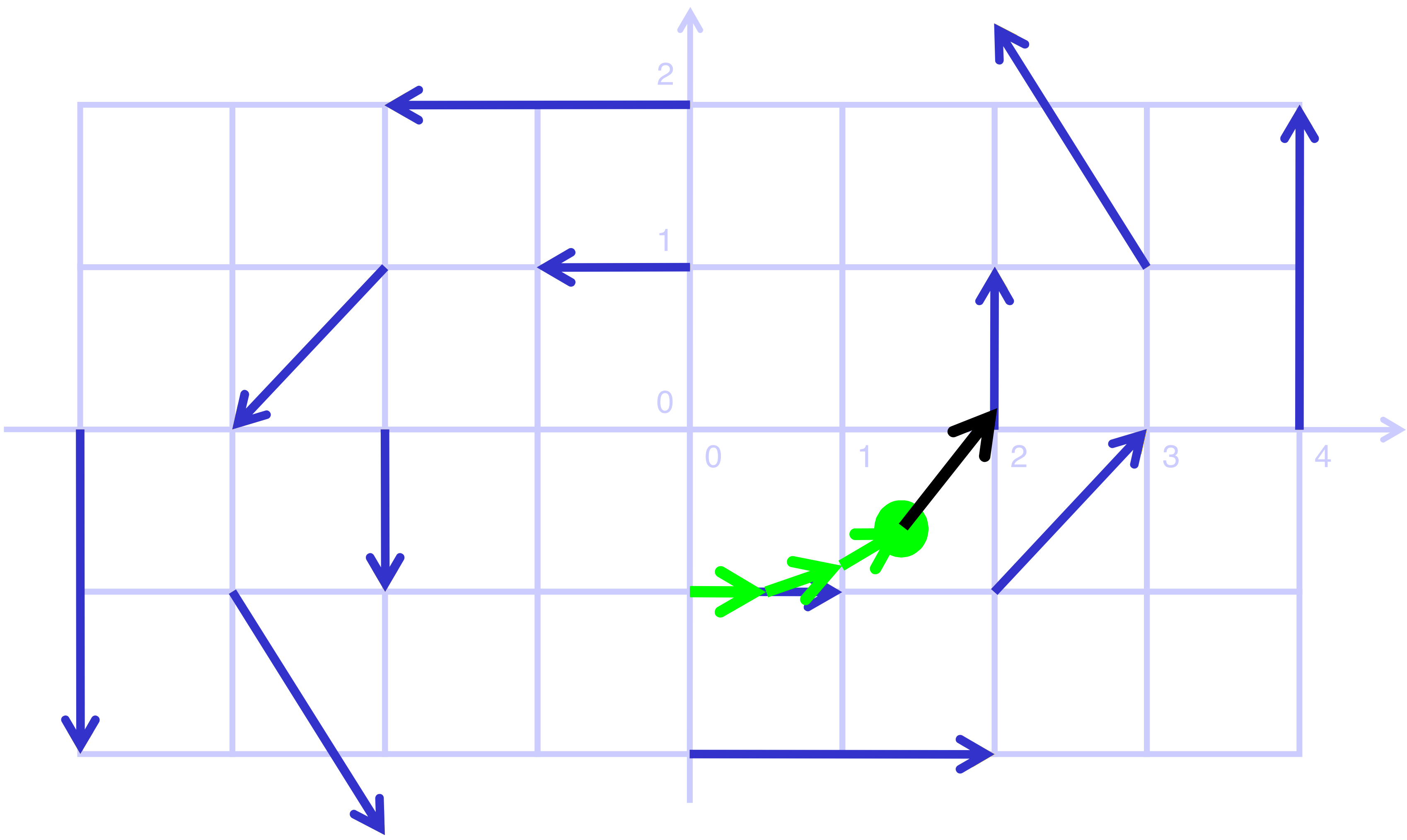
Euler Integration – Example

- $\mathbf{s}_3 = (23/16 \mid -5/8)^T \approx (1.44 \mid -0.63)^T;$

$\mathbf{v}(\mathbf{s}_3) = (5/8 \mid 23/32)^T \approx (0.63 \mid 0.72)^T;$

$v_x = \mathrm{d}x/\mathrm{d}t = -y$

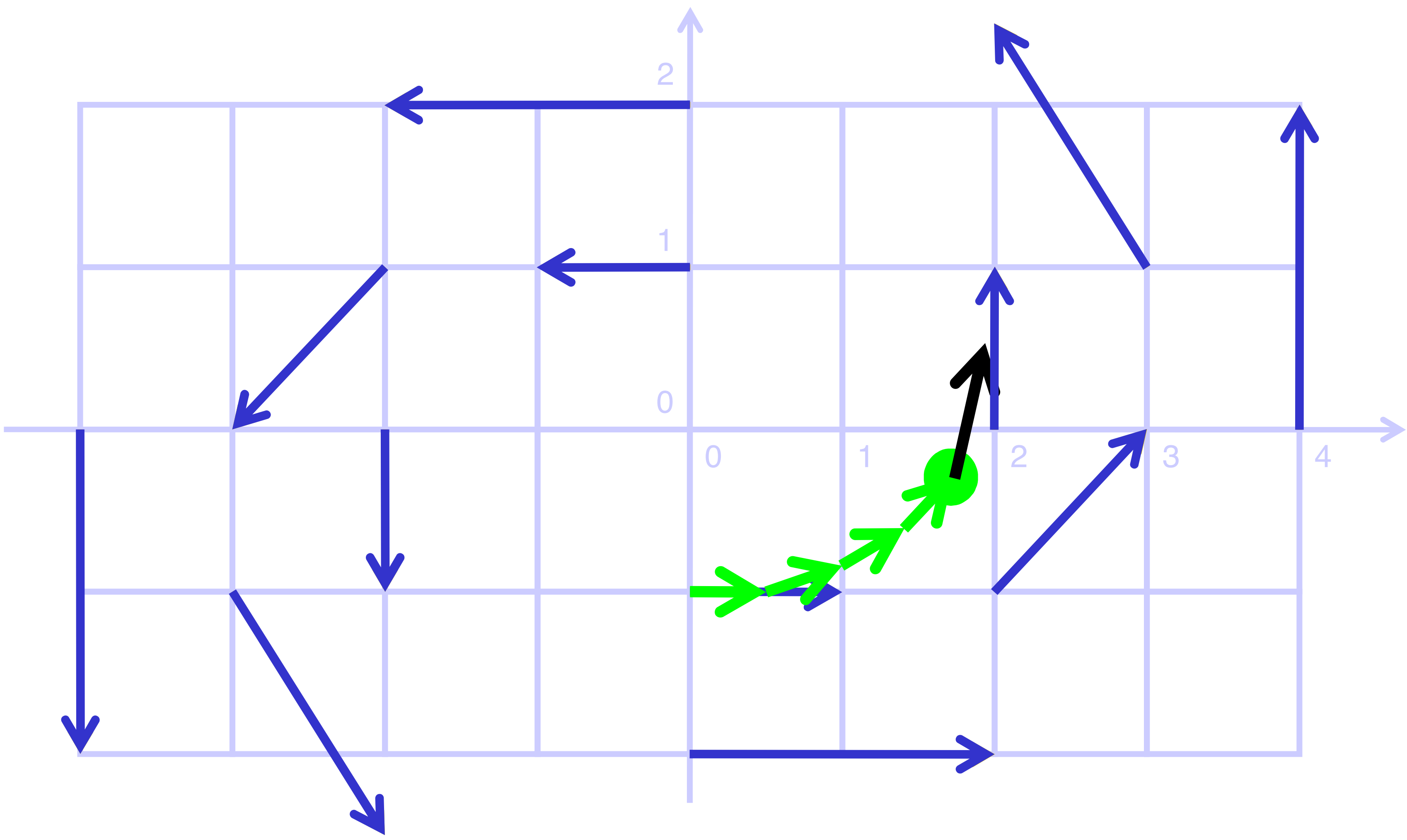
$v_y = dy/dt = x/2$



Euler Integration – Example

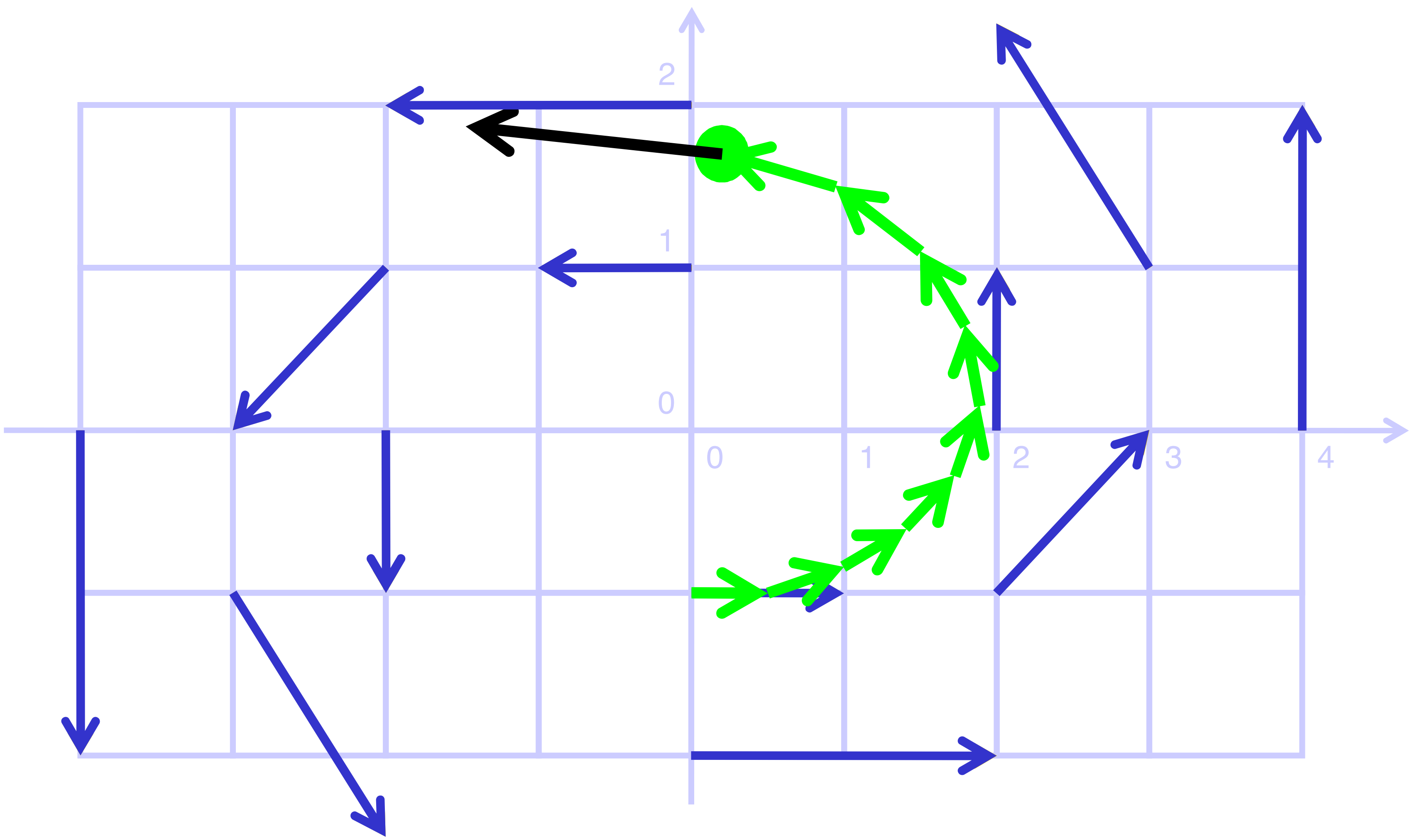
- $\mathbf{s}_4 \quad = (7/4 \mid -17/64)^T \approx (1.75 \mid -0.27)^T;$

$\mathbf{v}(\mathbf{s}_4) \quad = (17/64 \mid 7/8)^T \approx (0.27 \mid 0.88)^T;$



Euler Integration – Example

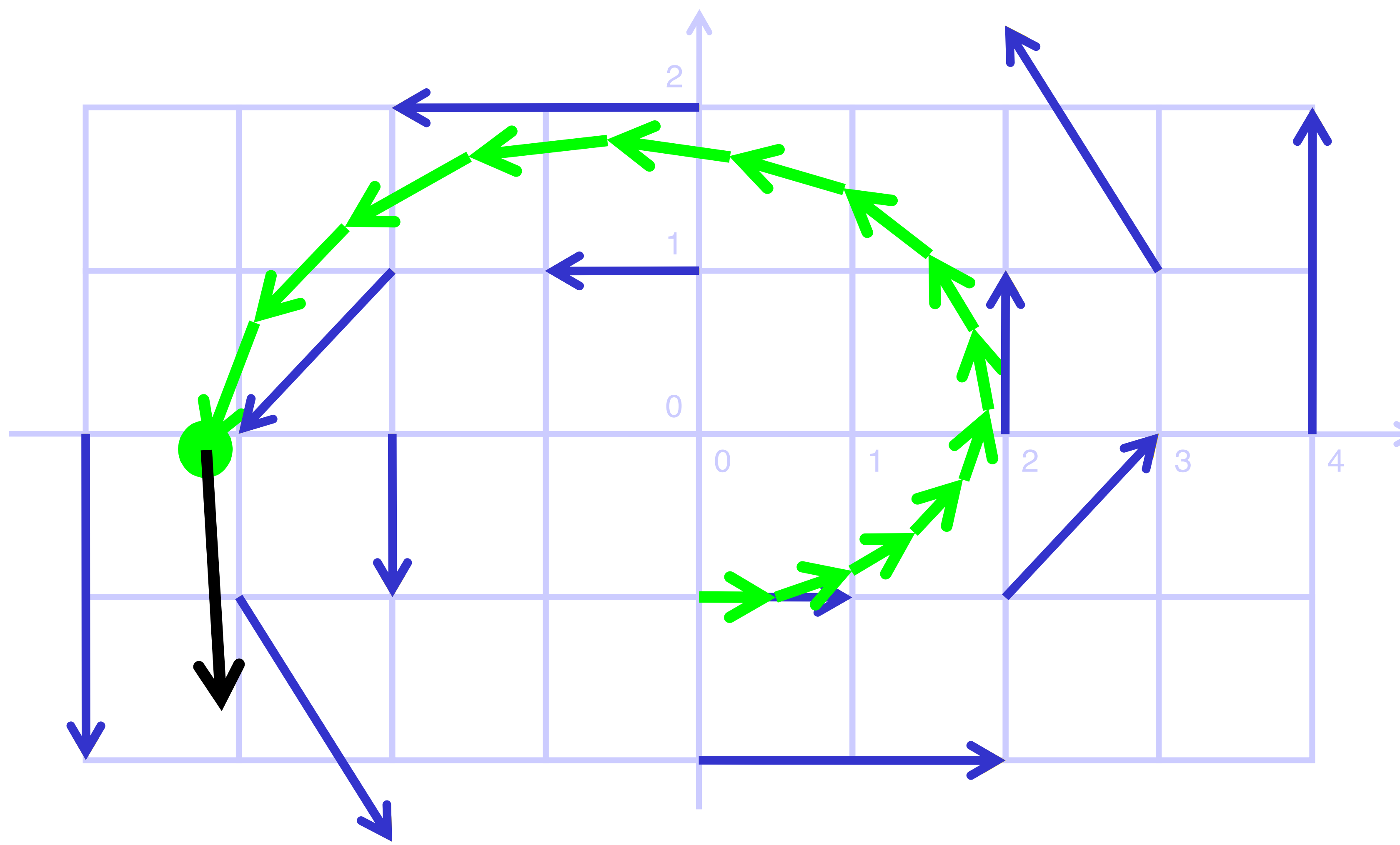
$\mathbf{s}_9 \approx (0.20 | 1.69)^T$;
$\mathbf{v}(\mathbf{s}_9) \approx (-1.69 | 0.10)^T$;



Euler Integration – Example

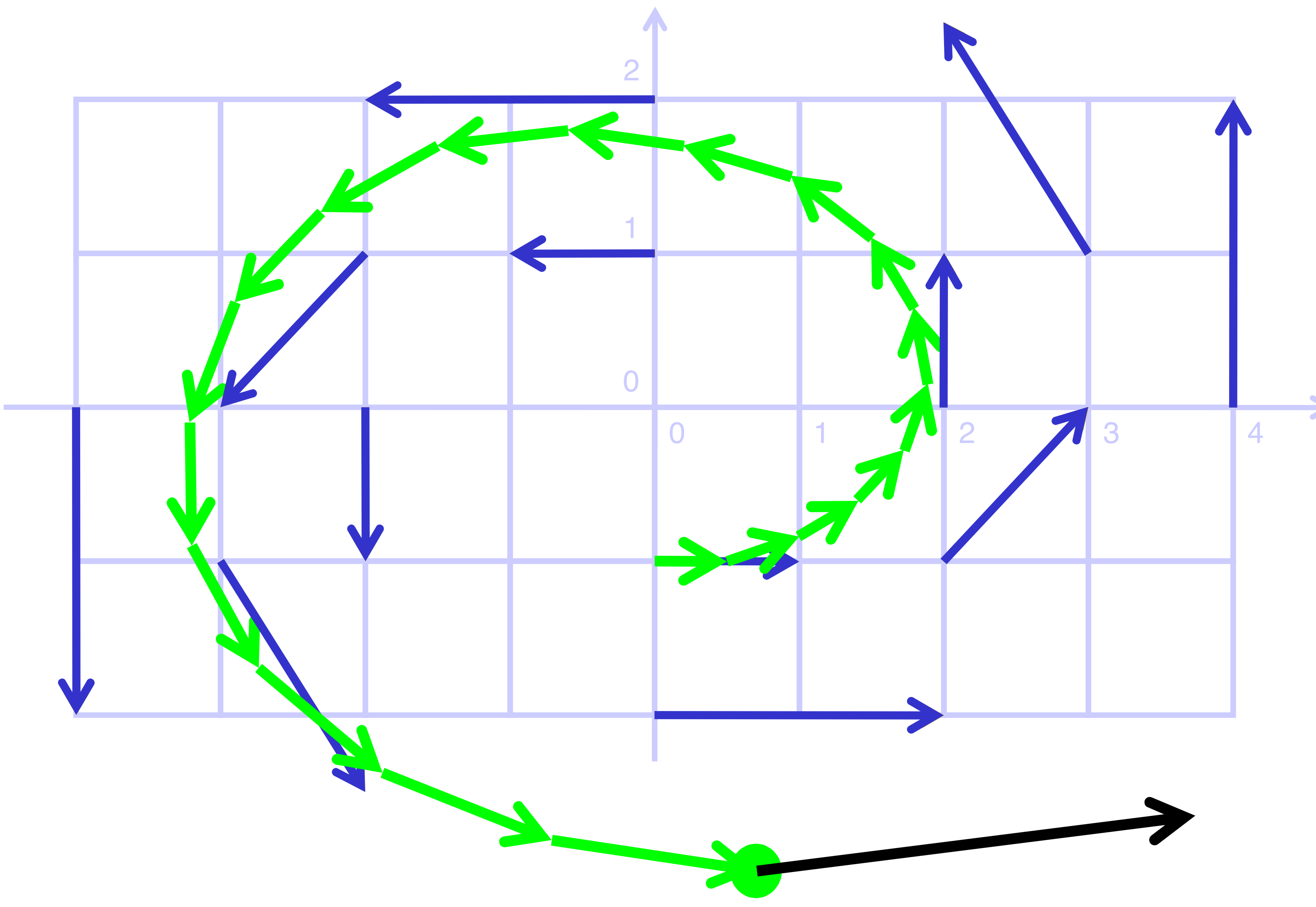
- $\mathbf{s}_{14} \approx (-3.22 | -0.10)^T$;

$\mathbf{v}(\mathbf{s}_{14}) \approx (0.10 | -1.61)^T$;



Euler Integration – Example

- $\mathbf{s}_{19} \approx (0.75 | -3.02)^T$; $\mathbf{v}(\mathbf{s}_{19}) \approx (3.02 | 0.37)^T$; clearly: large integration error, dt too large, 19 steps

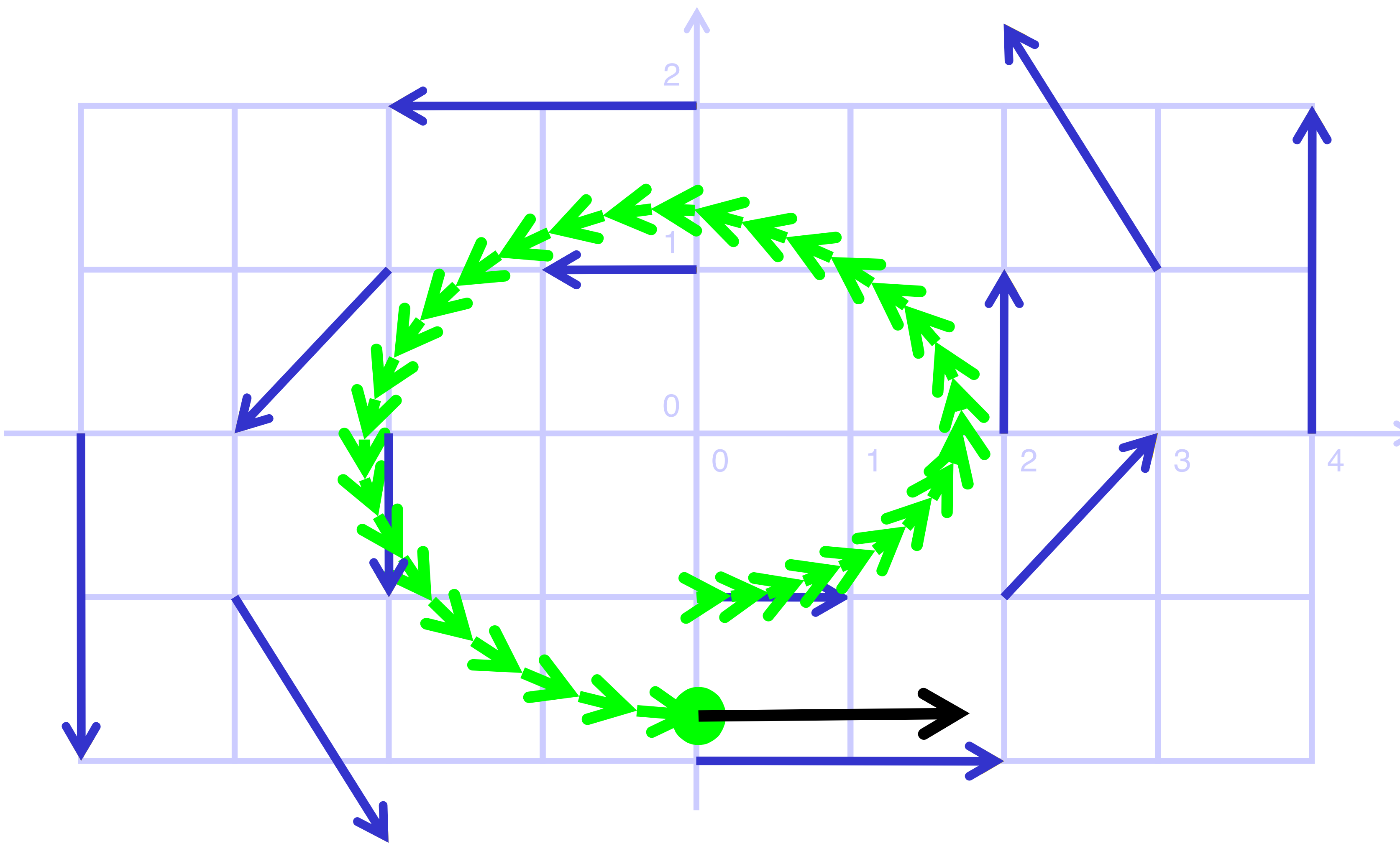


Euler Integration – Example

- d*t* smaller (1/4): more steps, more exact.

$\mathbf{s}_{36} \approx (0.04 \mid -1.74)^T$; $\mathbf{v}(\mathbf{s}_{36}) \approx (1.74 \mid 0.02)^T$;

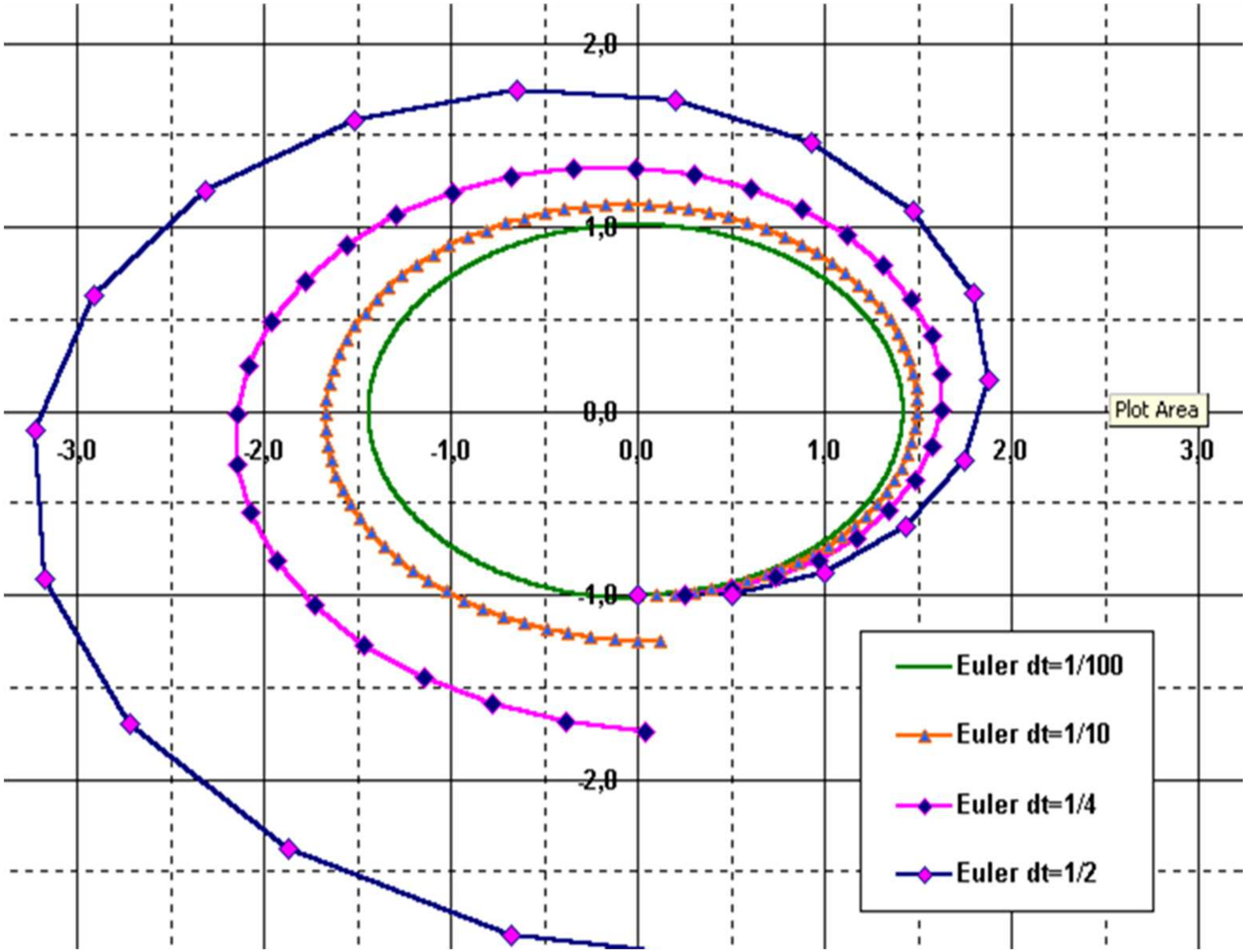
- 36 steps



Comparison Euler, Step Sizes

Euler

quality is proportional to d*t*

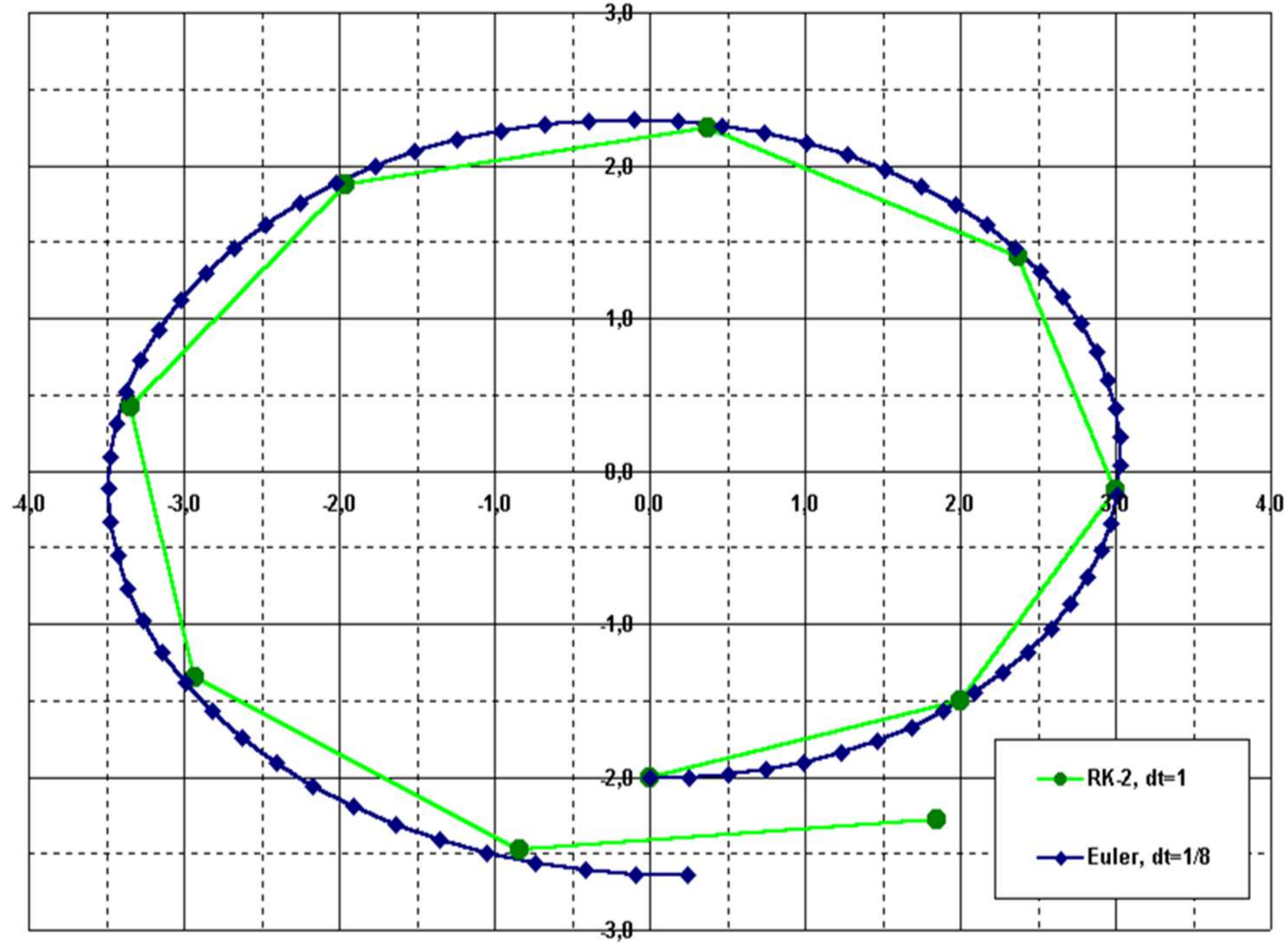


Euler Example – Error Table

dt	#steps	error
1/2	19	~200%
1/4	36	~75%
1/10	89	~25%
1/100	889	~2% ✓
1/1000	8889	~0.2%

RK-2 – A Quick Round

RK-2: even with dt = 1 (9 steps)
better
than Euler
with dt = 1/8
(72 steps)



RK-4 vs. Euler, RK-2

Even better: fourth order RK:

- four vectors **a**, **b**, **c**, **d**
- one step is a convex combination:
  $\mathbf{s}_{i+1} = \mathbf{s}_i + (\mathbf{a} + 2\cdot\mathbf{b} + 2\cdot\mathbf{c} + \mathbf{d})/6$
- vectors:

  $\mathbf{a} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i)$ … original vector

  $\mathbf{b} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i+\mathbf{a}/2)$ … RK-2 vector

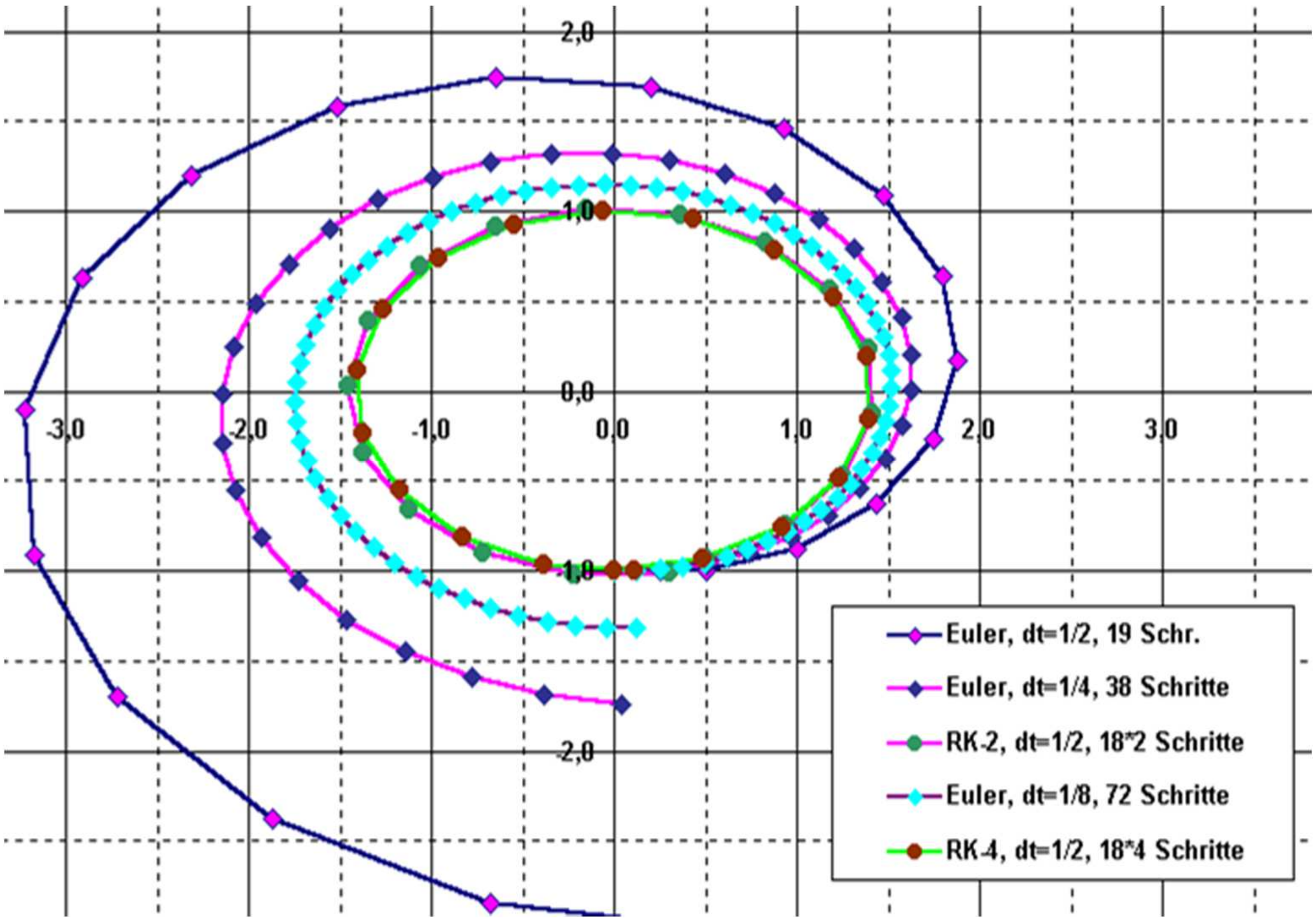
  $\mathbf{c} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i+\mathbf{b}/2)$ … use RK-2 …

  $\mathbf{d} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i+\mathbf{c})$ … and again

Euler vs. Runge-Kutta

RK-4: pays off only with complex flows

Here approx. like RK-2



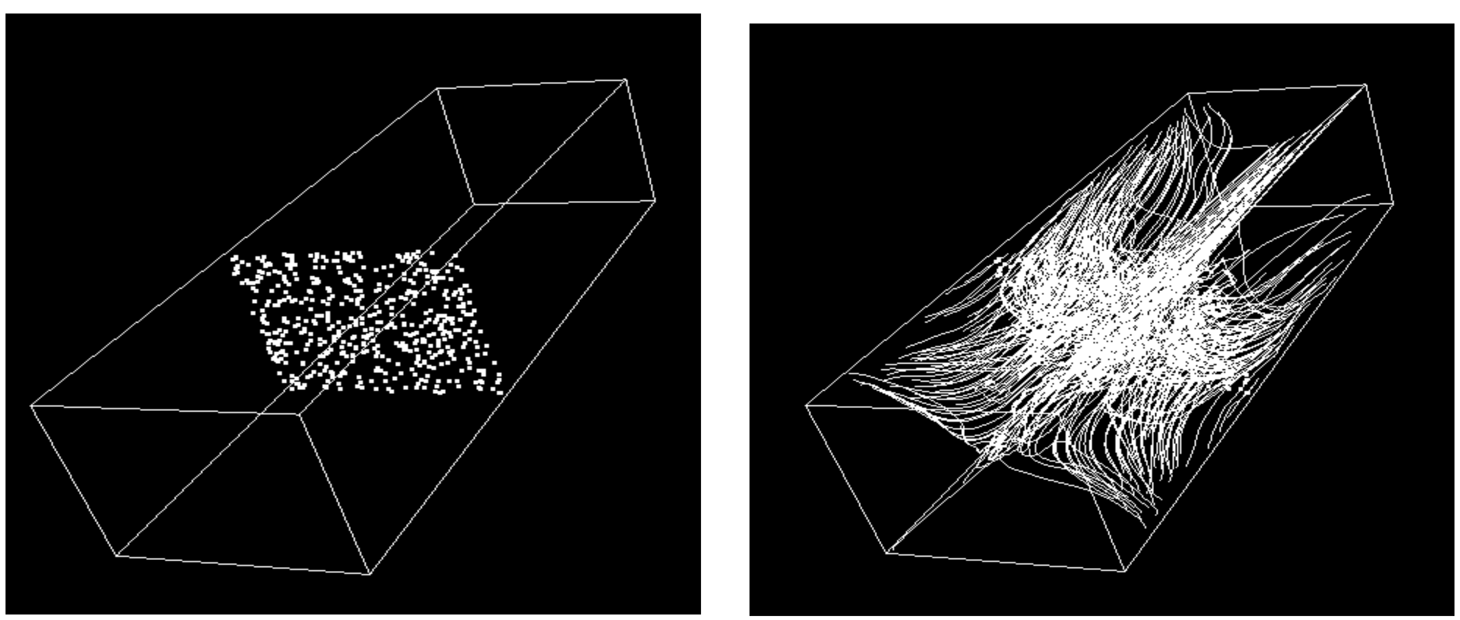
Integration, Conclusions

Summary:

- analytic determination of streamlines usually not possible
- hence: numerical integration
- various methods available
  (Euler, Runge-Kutta, etc.)
- Euler: simple, imprecise, esp. with small dt
- RK: more accurate in higher orders
- furthermore: adaptive methods, implicit methods, etc.

Streamline Placement
(in 2D)

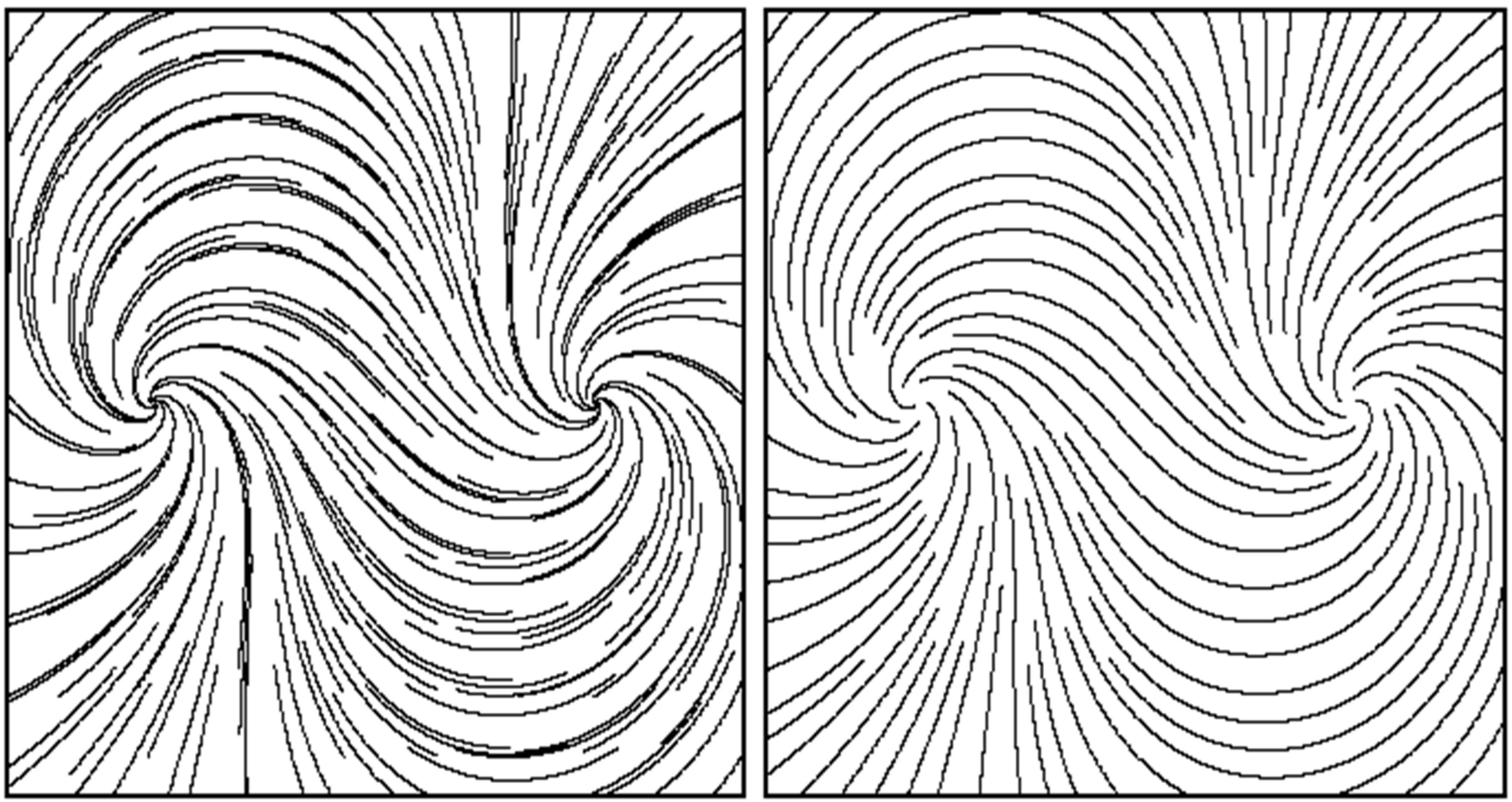
- **Seeding of integral lines:**
- which stream/path/streak/time lines to visualize?
- too few: important details get lost
- too many: overload, visual clutter
- simple approaches:
  - start on regular grid points
  - start randomly
- It has to be the right number at the right places!!!

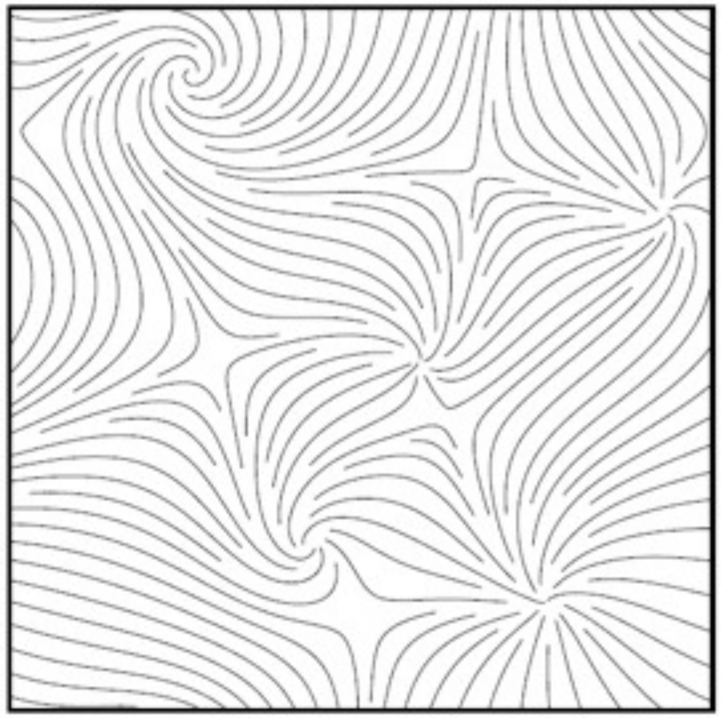


Problem: Choice of Seed Points

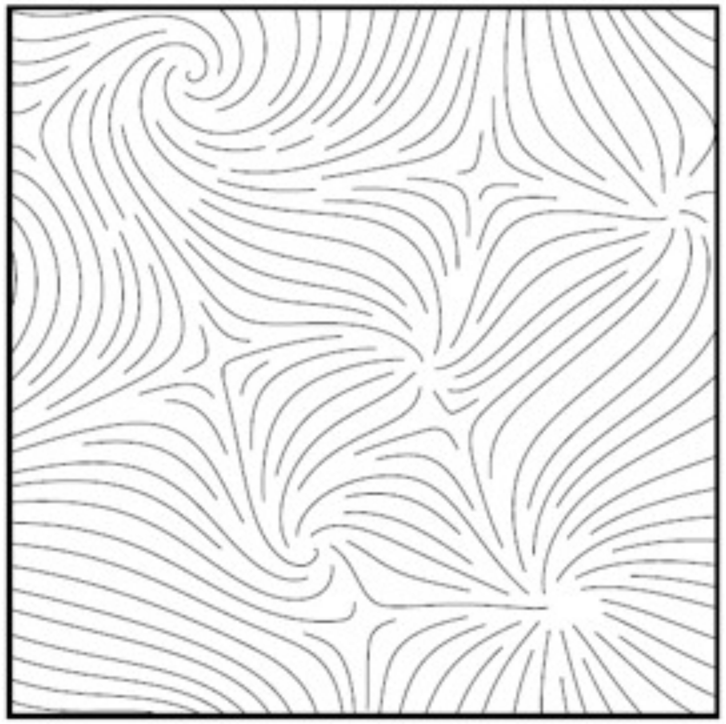
Streamline placement:

- If regular grid used: very irregular result



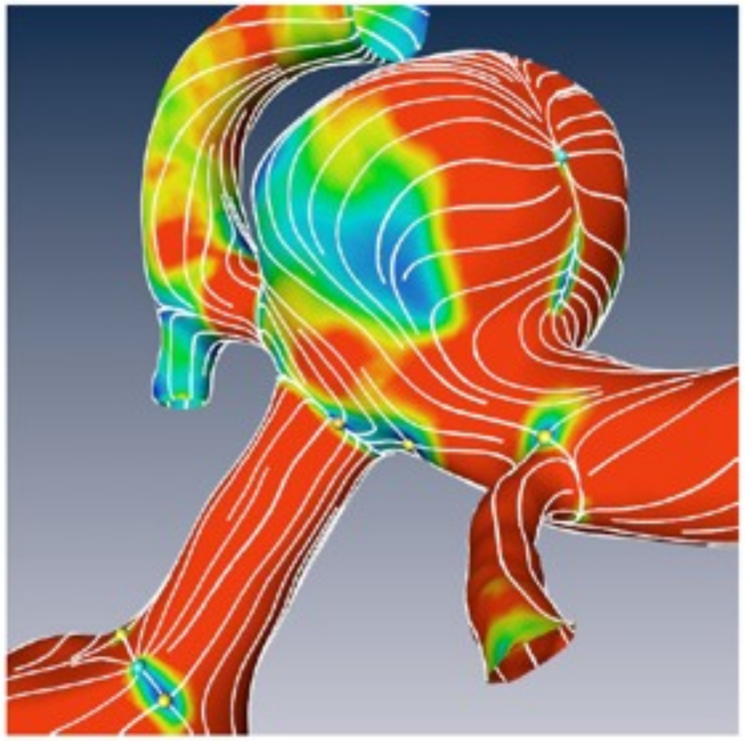


Turk and Banks, 1996



Jobard et al., 1997

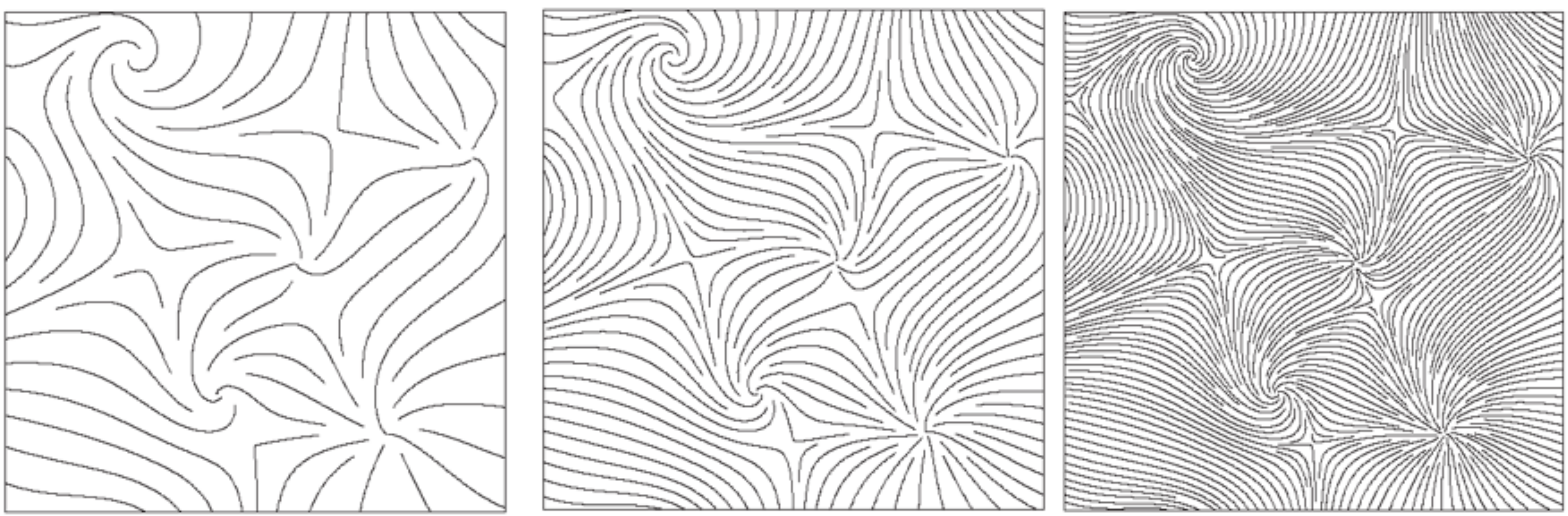
Mebarki et al., 2005



Rosanwo et al., 2009

Streamline seeding

- 2D: evenly spaced stream lines
- Turk/Banks 96:
  - Start with “streamlets” (very short stream lines)
  - Apply a series of energy-decreasing elementary operations: combine, delete, create, lengthen, shorten streamlets
  - Energy: difference between low-pass filtered version of current placements and uniform grey image



Main idea: the distribution of ink on the screen should be even [Turk and Bank 96]

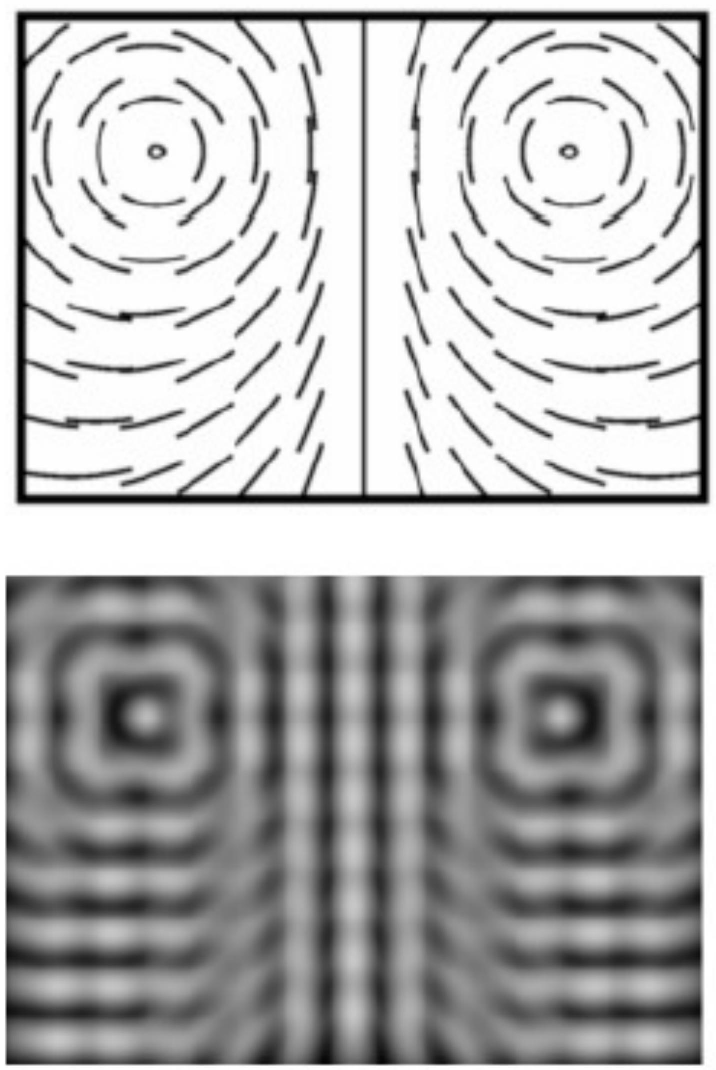


Figure 2: (a) Short streamlines with centers placed on a regular grid (top); (b) filtered version of same (bottom).

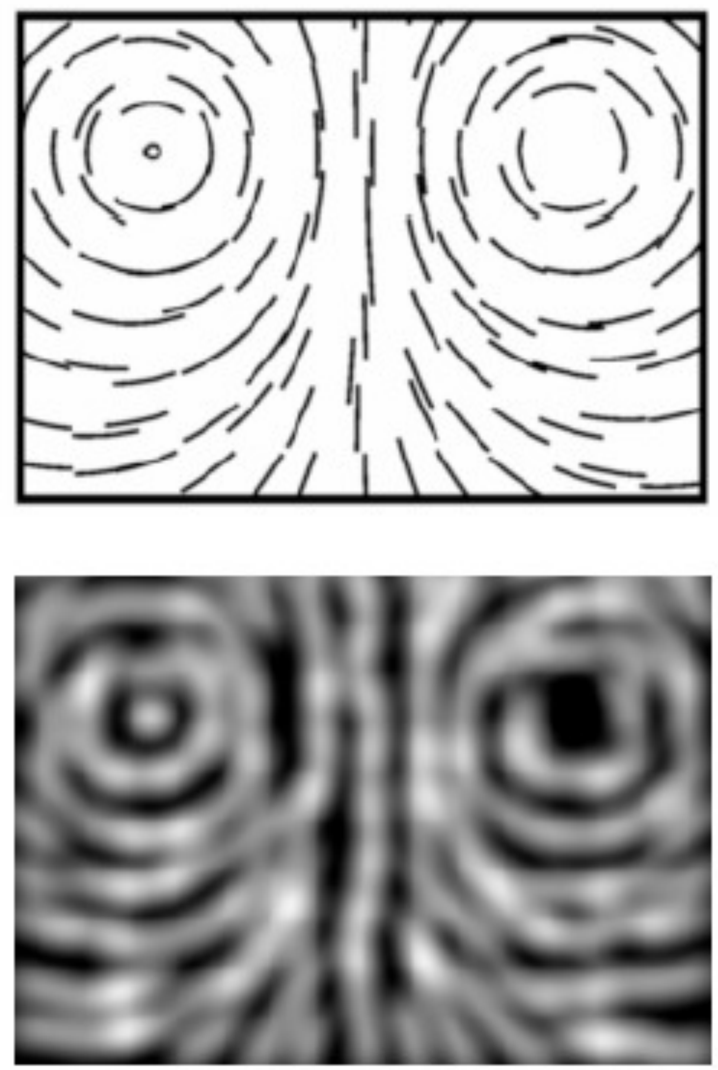


Figure 3: (a) Short streamlines with centers placed on a jittered grid (top); (b) filtered version showing bright and dark regions (bottom).

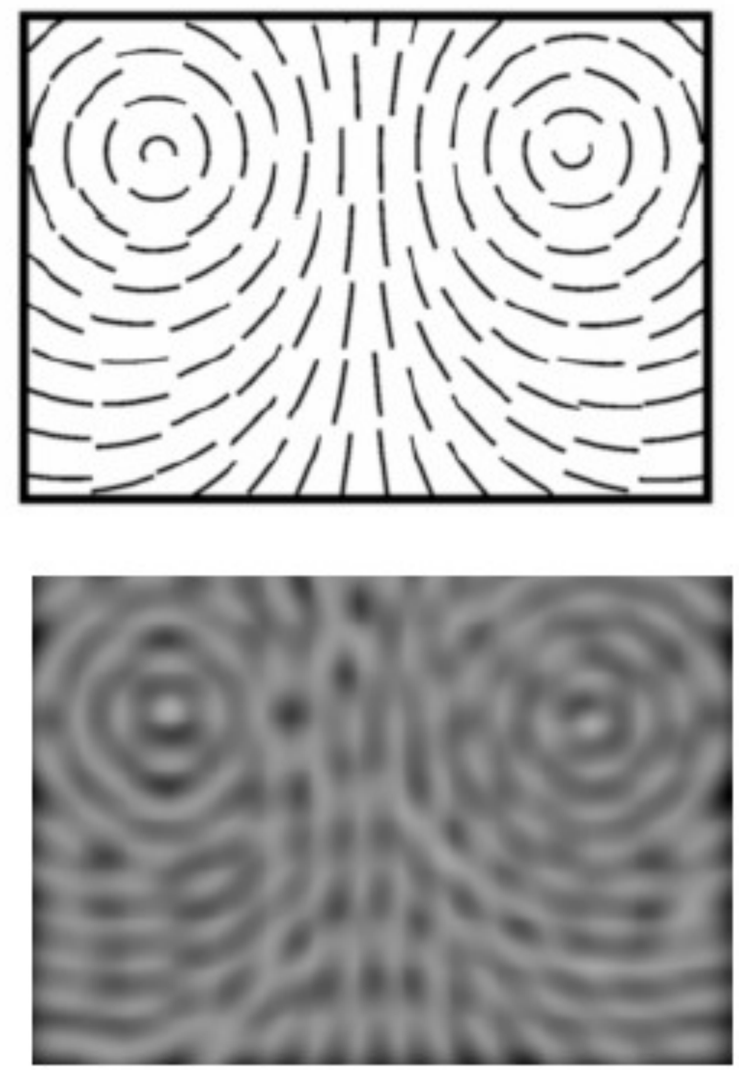
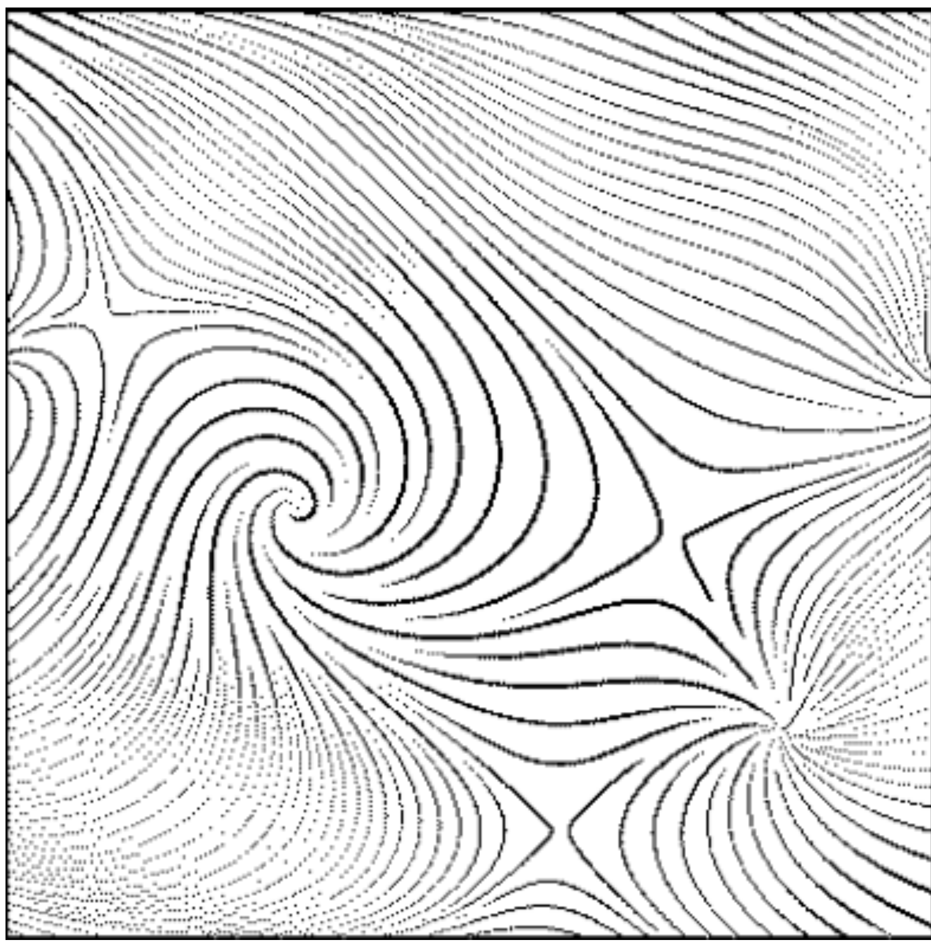


Figure 4: (a) Short streamlines placed by optimization (top); (b) filtered version showing fairly even gray value (bottom).

Results



Tapering at streamline ends

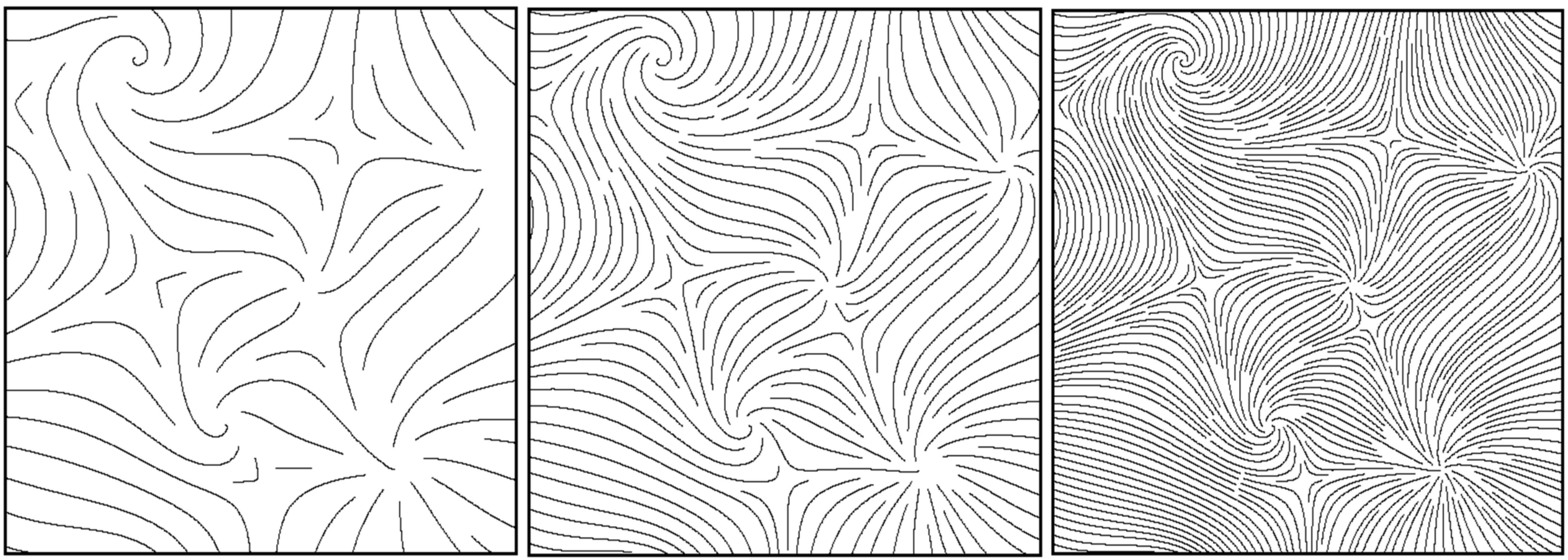
Optimized arrow plots

[Turk and Bank ’96]

Different Streamline Densities

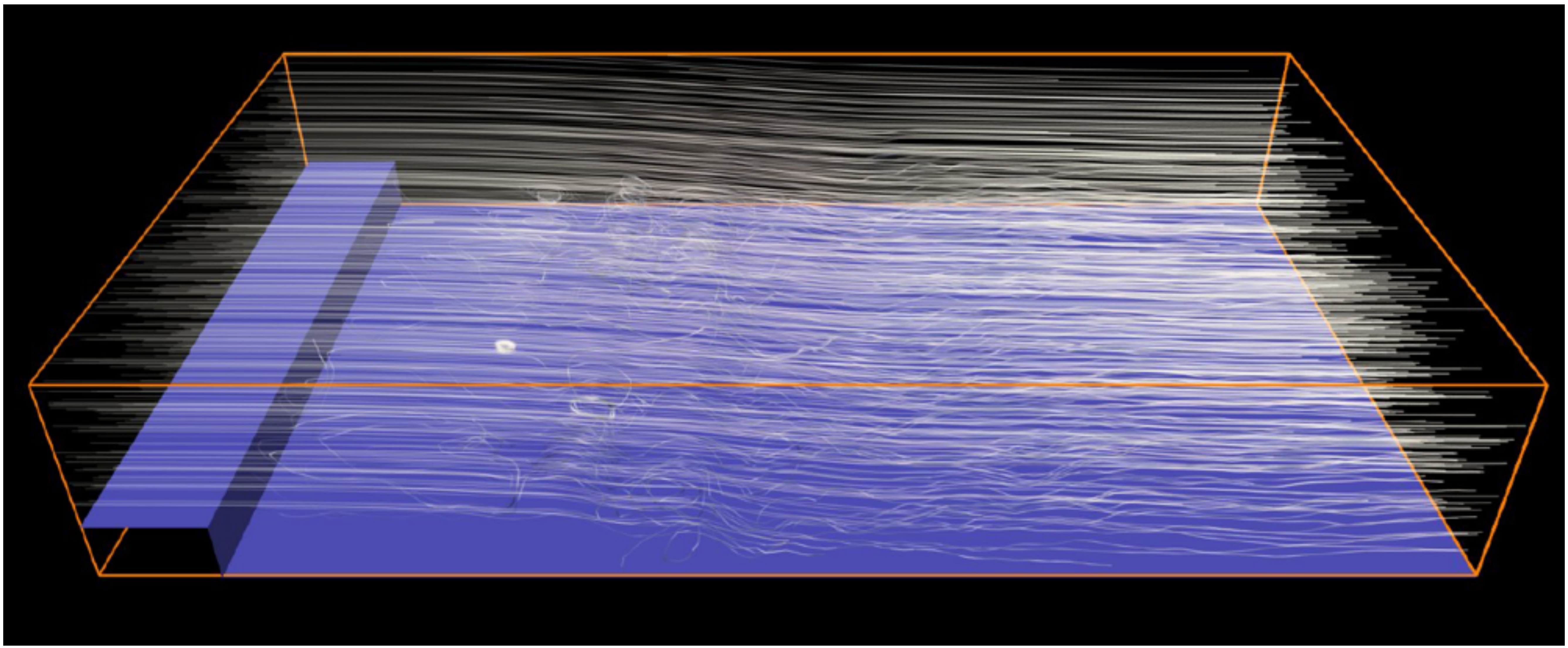
Variations of d_{sep} relative to image width:

6% 3% 1.5%



Streamline Seeding in 3D

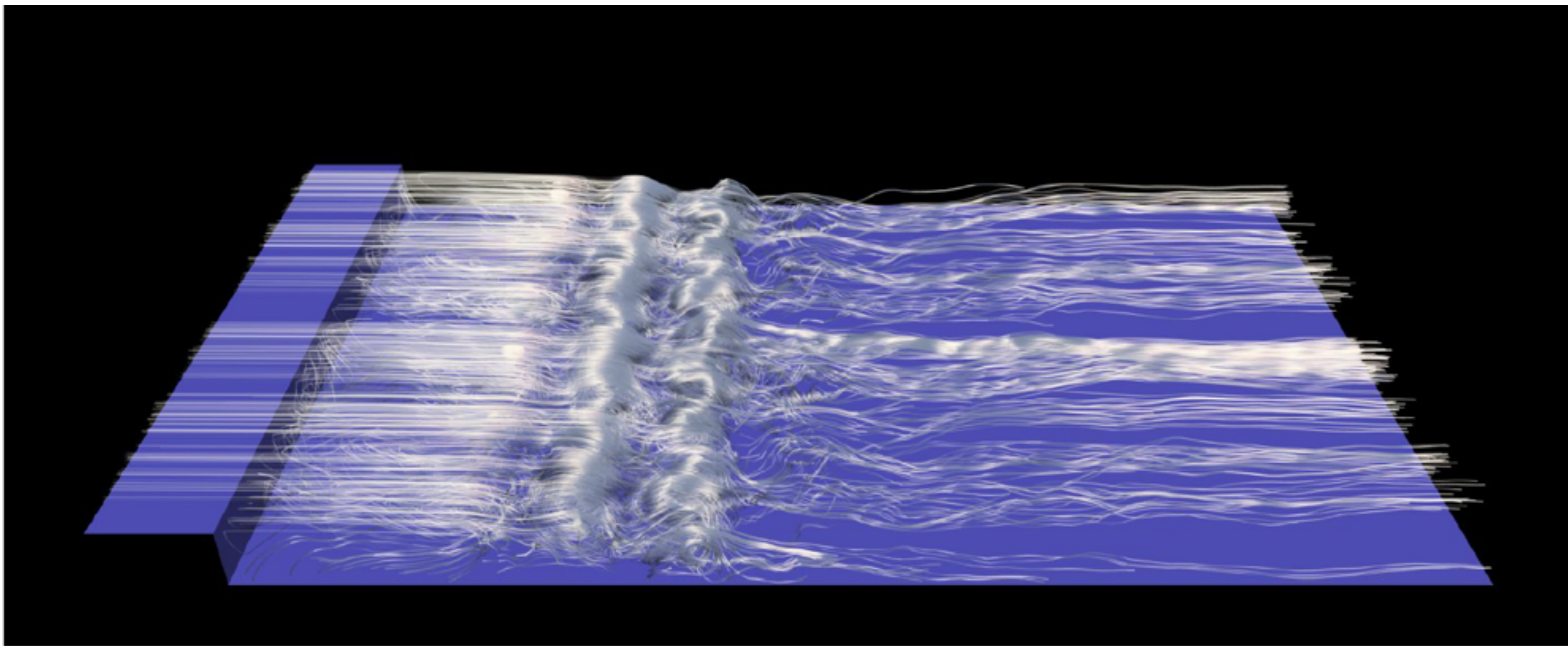
- Evenly-spaced does not make sense
- Start on uniform grid



Weinkauf 2003

Streamline Seeding in 3D

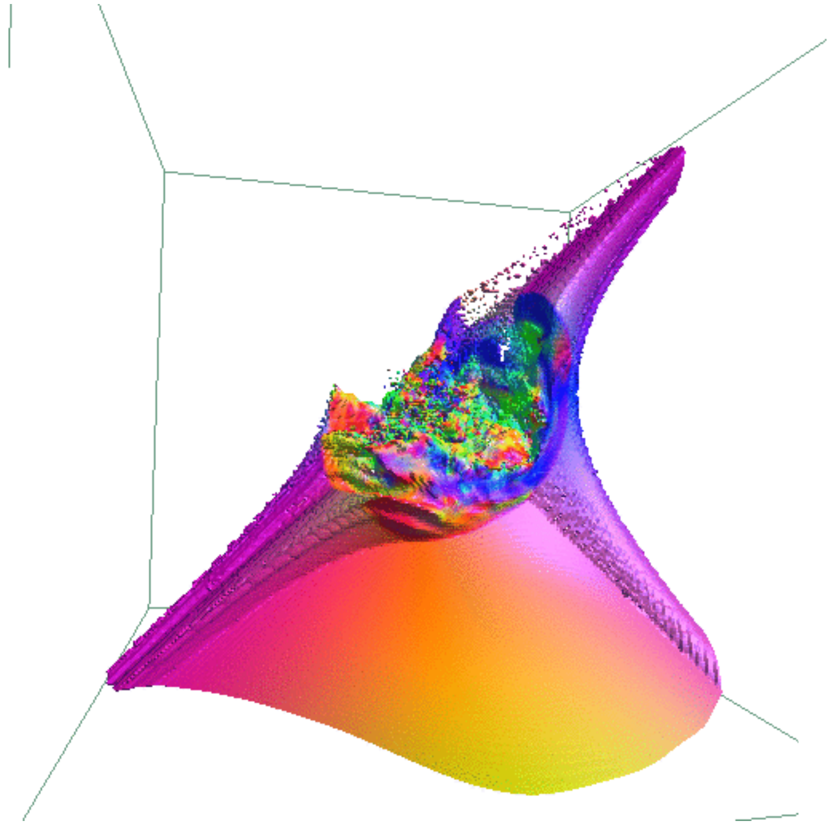
- Evenly-spaced does not make sense
- Start in regions of high vector field curvature (i.e., close to critical points):



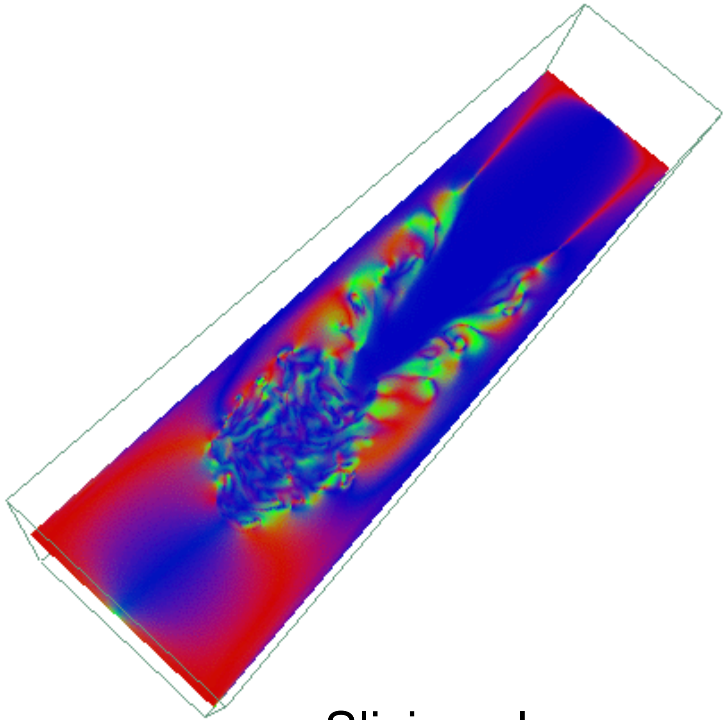
Weinkauf 2003

Seeds in Image Space

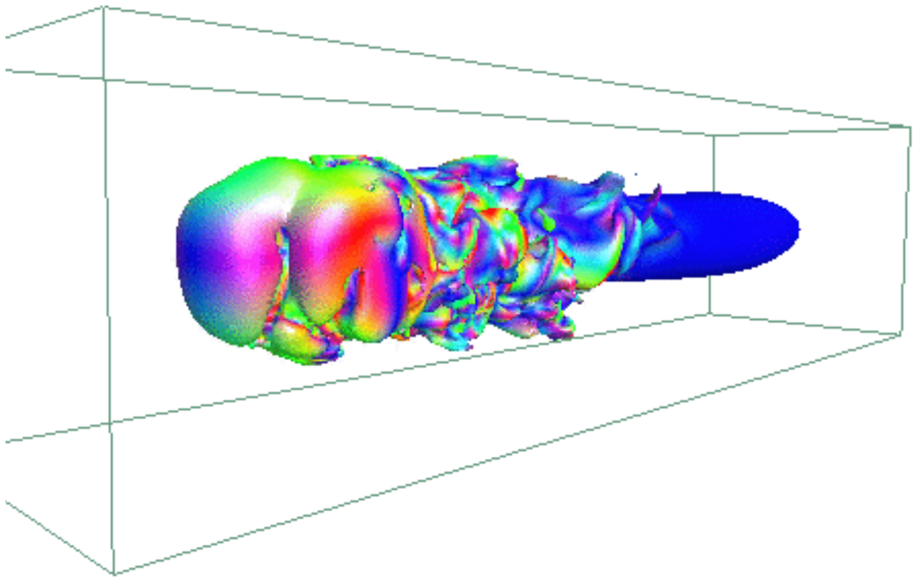
- Need to un-project the seeds back to 3D object space for streamline integration
- Utilize depth maps generated from other visualization techniques



Stream surface

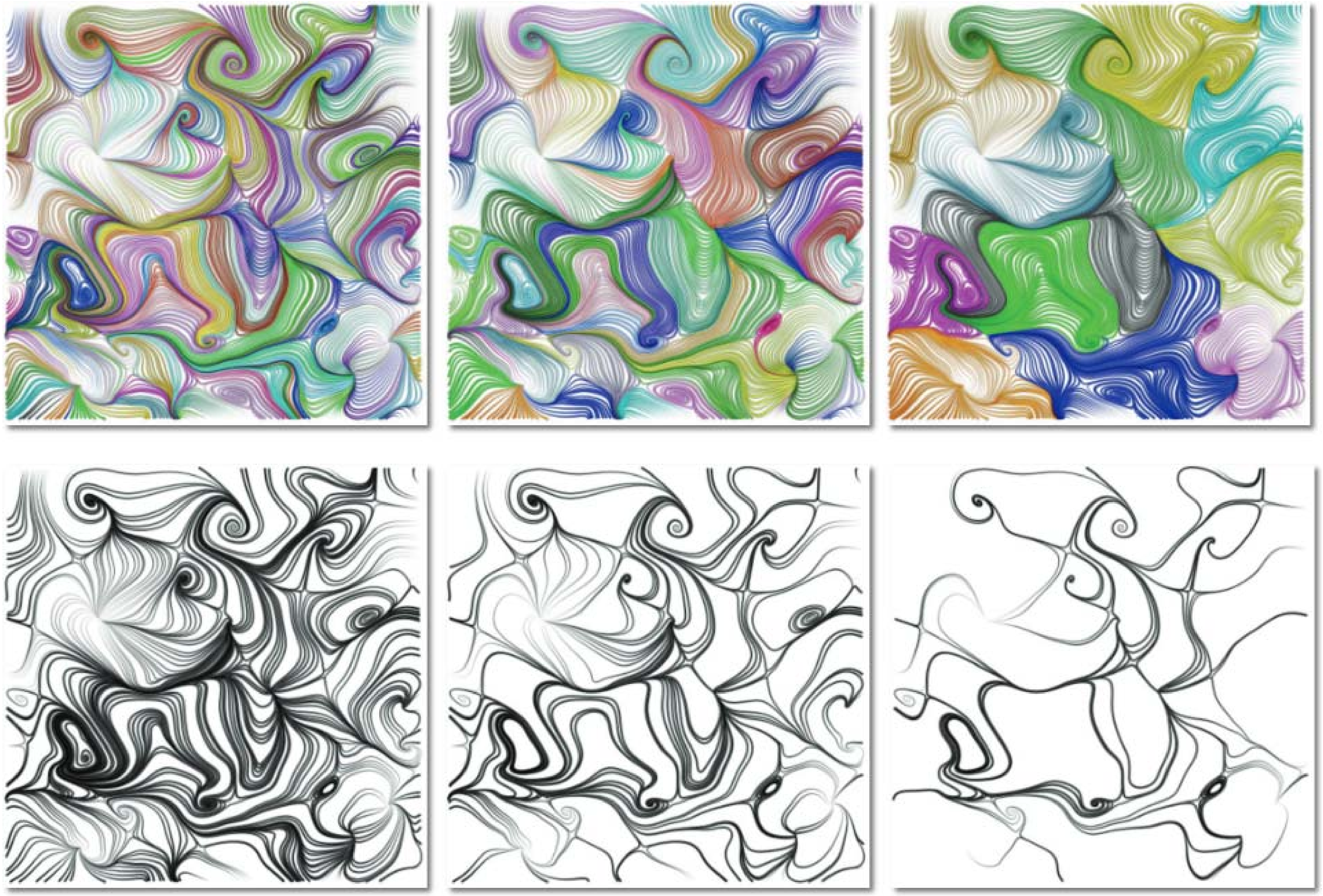


Slicing plane



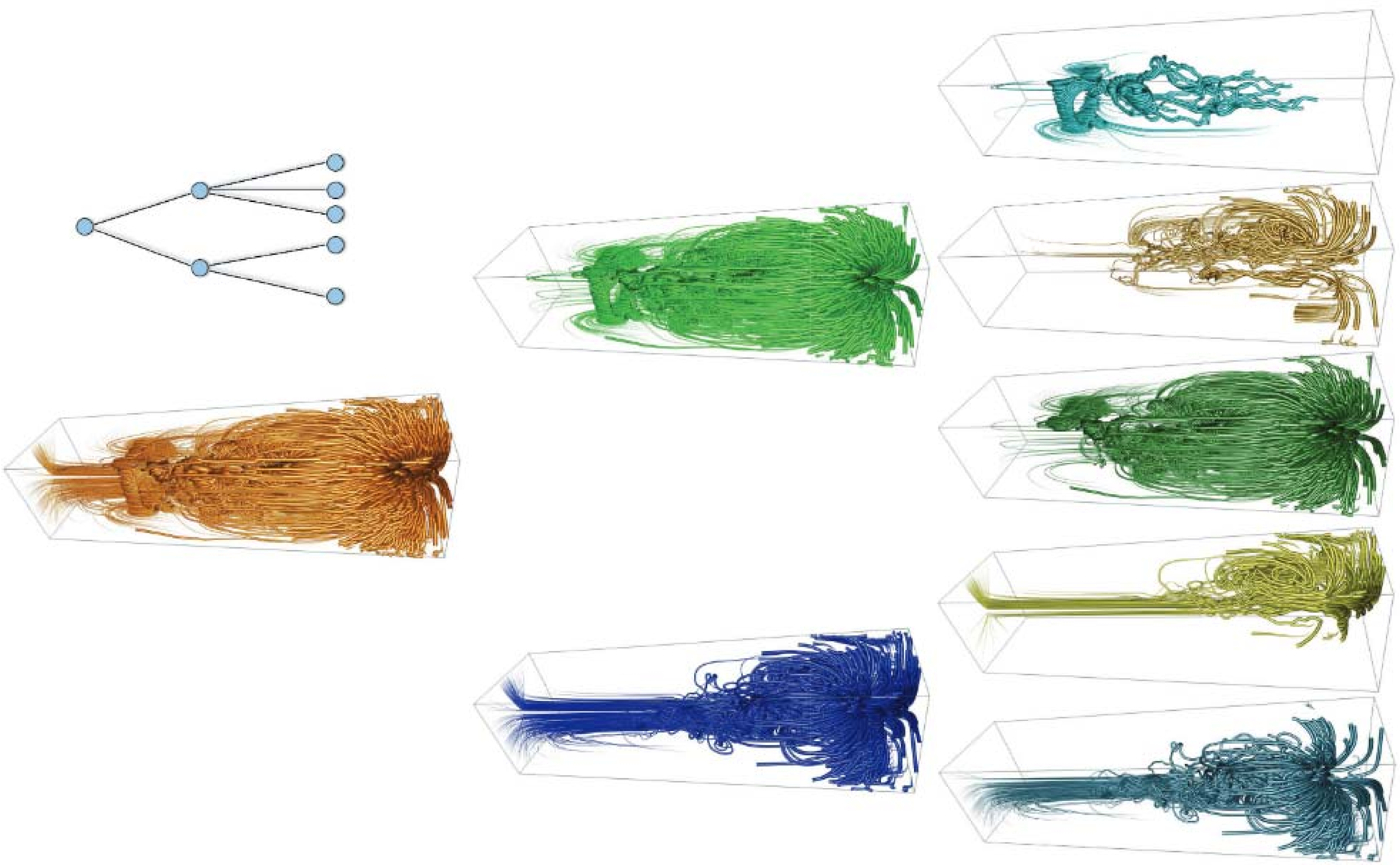
Isosurface

Streamline Bundling



[Yu et al. 2012]

Streamline Bundling



[Yu et al. 2012]

Opacity Optimization for 3D Line Fields

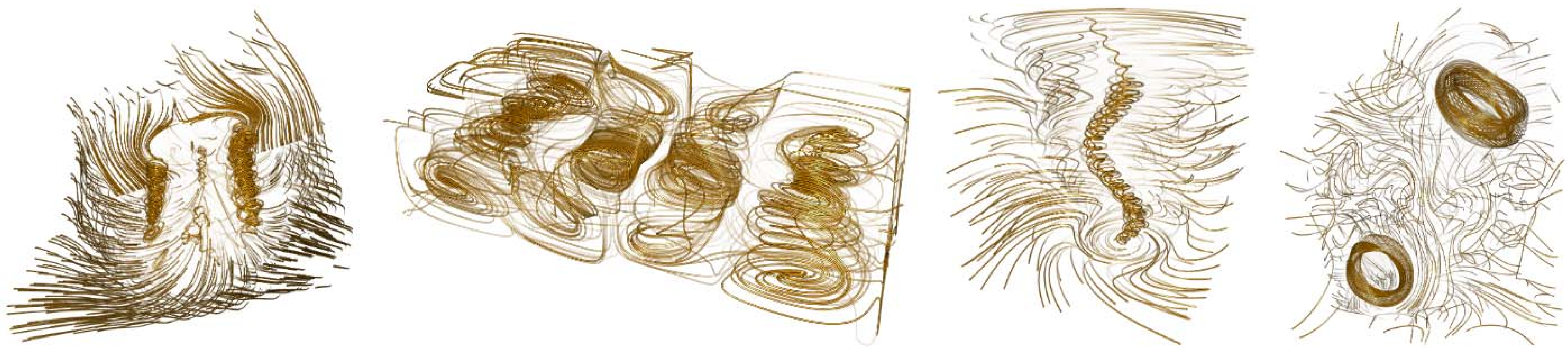


Figure 1: *Applications of our interactive, global line selection algorithm. Our bounded linear optimization for the opacities reveals user-defined important features, e.g., vortices in rotorcraft flow data, convection cells in heating processes (Rayleigh-Bénard cells), the vortex core of a tornado and field lines of decaying magnetic knots (from left to right).*

(a) Given is a set of polylines. (b) Discretize polylines into n segments (here: $n = 6$). (c) Compute per-segment opacity α_i by energy minimization. (d) Interpolate opacities between adjacent segments for final rendering.

Idea: make less important sections of streamlines transparent to fix occlusion, remove clutter.
SIGGRAPH 2013!

[Gunthe et al. 2013]

Illuminated Streamlines

Use lighting to improve spatial perception of lines in 3D.

This can to some extend reduce the 3D cluttering issue.

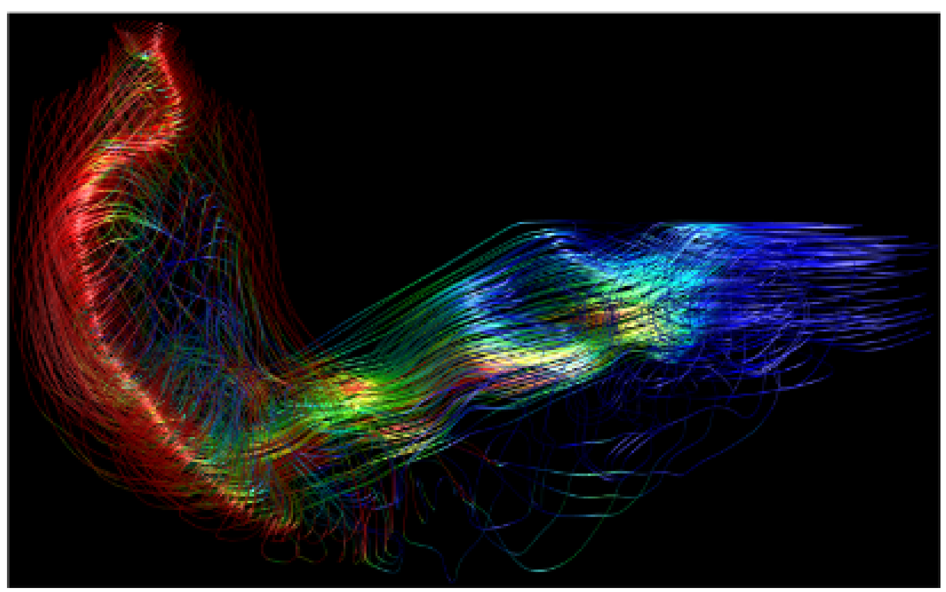
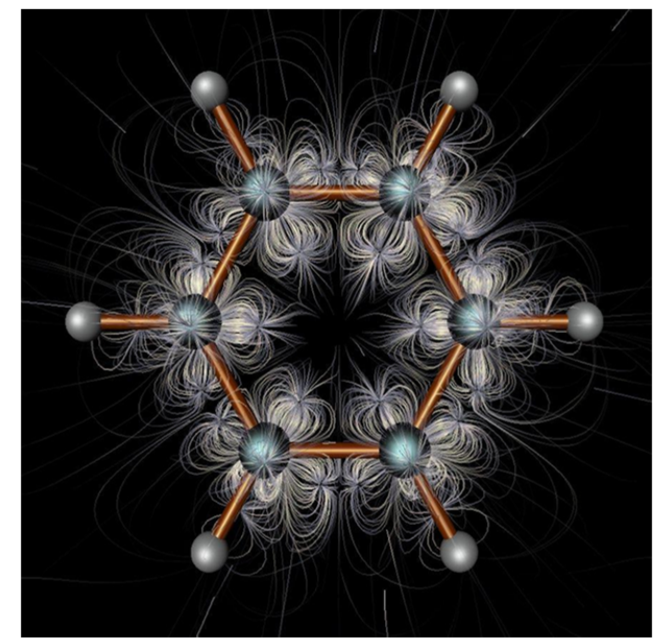


Figure 1: Flow in a Francis draft tube visualized by streamlines regularly seeded on a cone and colored by speed. Streamlines are illuminated based on cylinder averaging. In the vertical part of the tube, a vortex rope is visible.



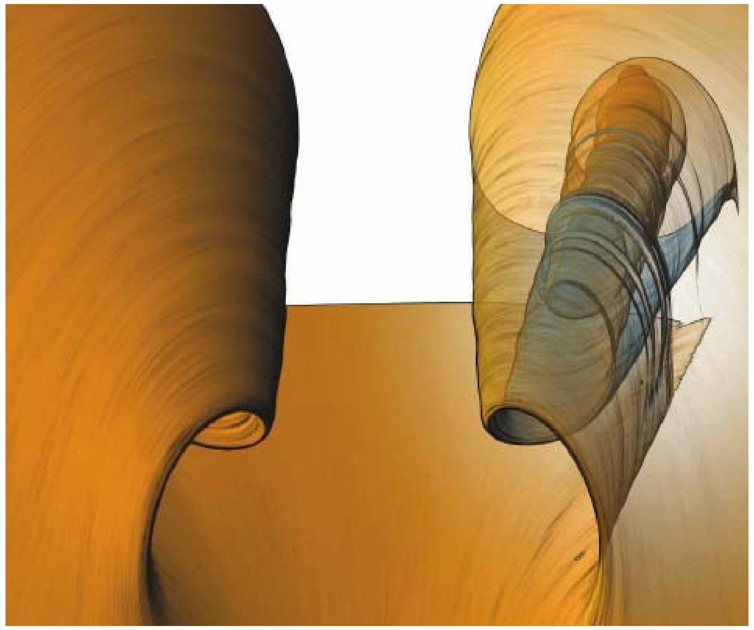
Open Source: http://www.scivis.ethz.ch/research/projects/illuminated_streamlines

[Zockler et al. 96, Mallo et al. 2005]

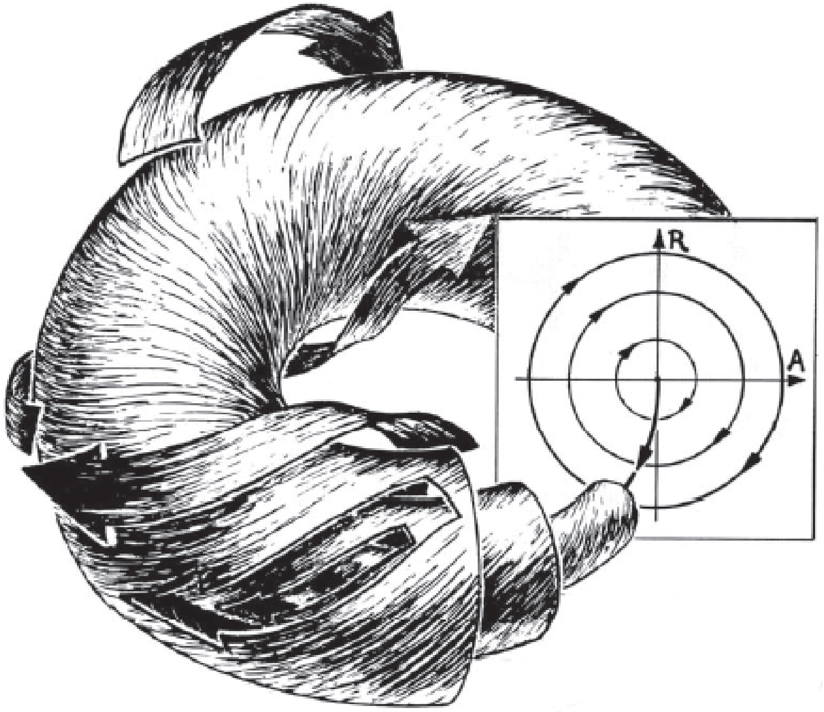
Rendering of stream surfaces

Illustrative visualization

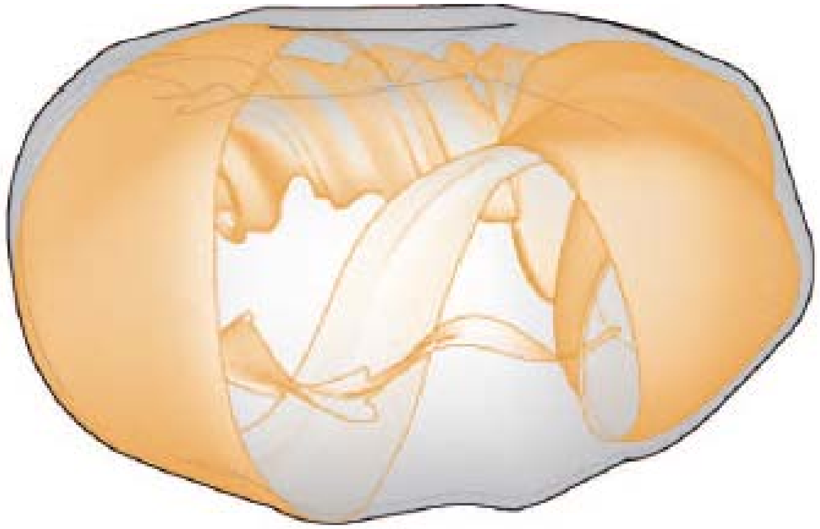
- Using transparency and surface features such as silhouette and feature curves.



[Hummel et al. 2010]

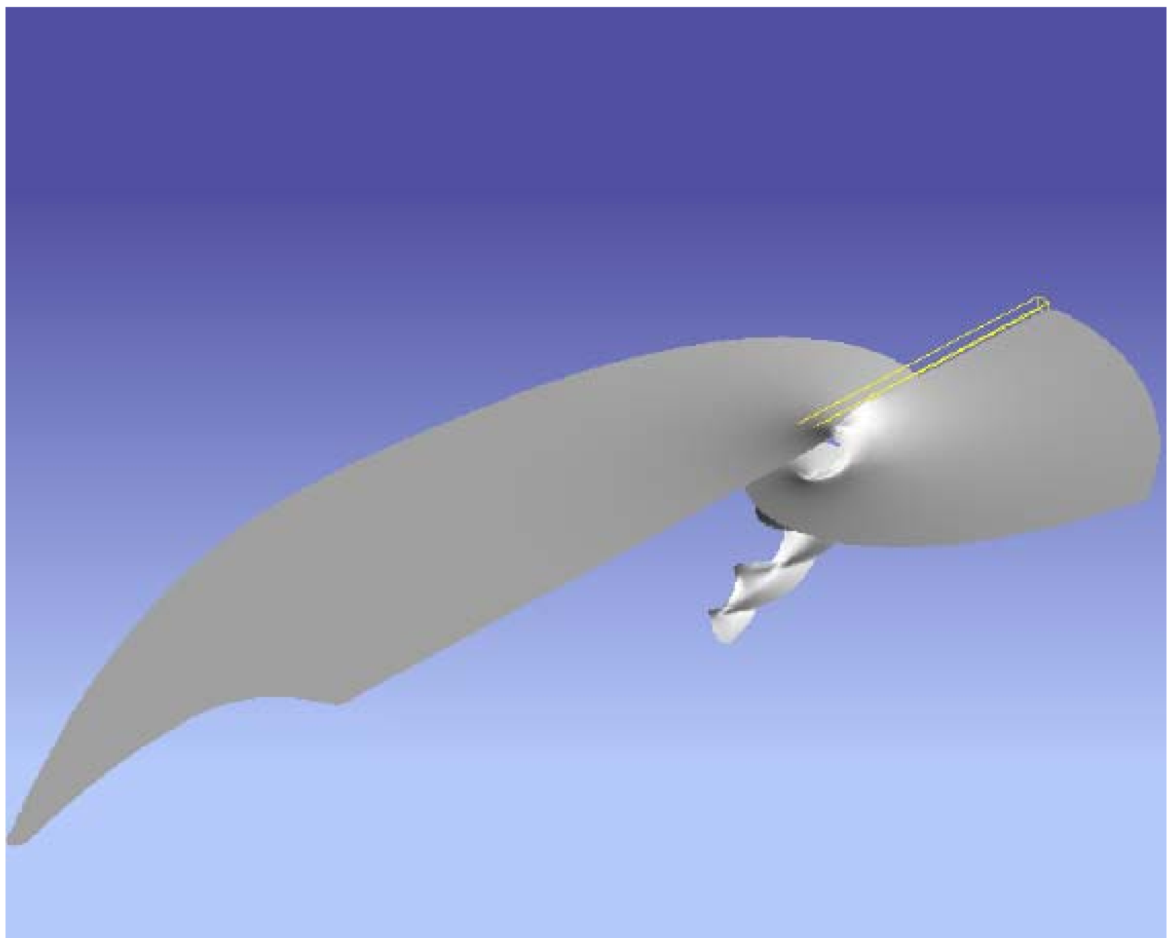


Abraham/Shaw's illustration, 1984

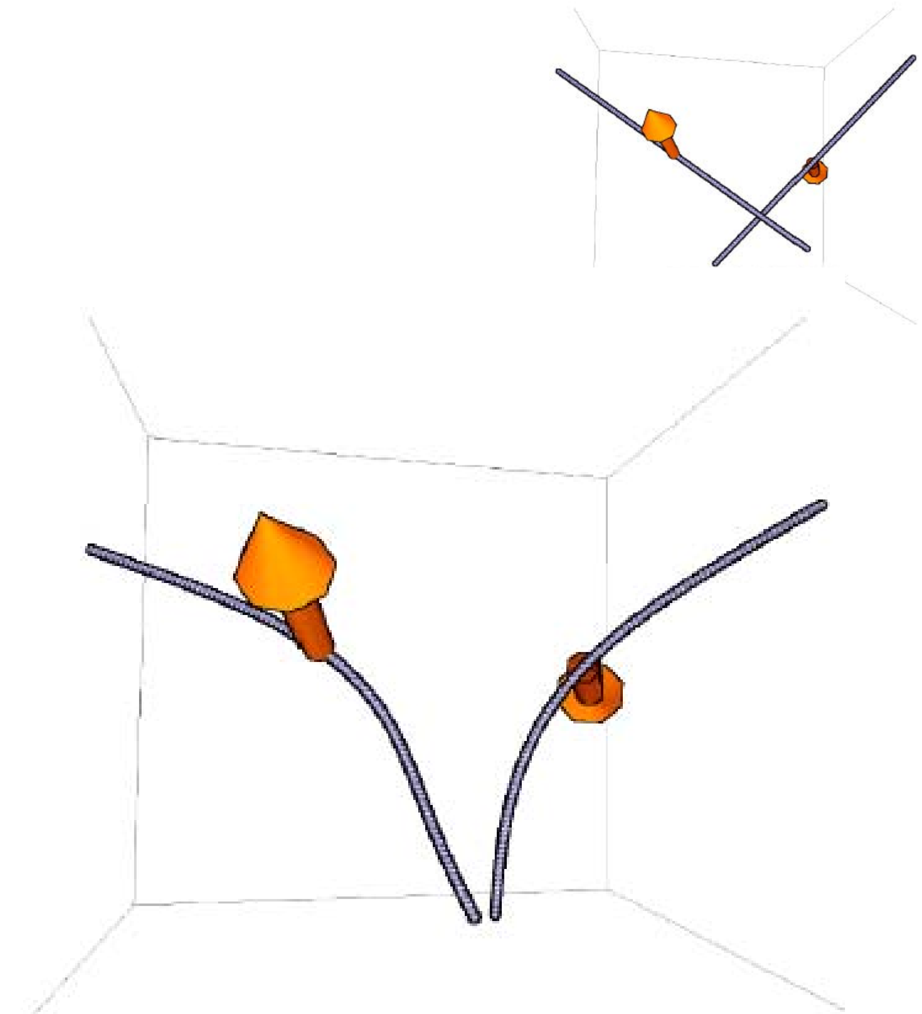


[Born et al. Vis2010]

Where to put seeds to start the integration?



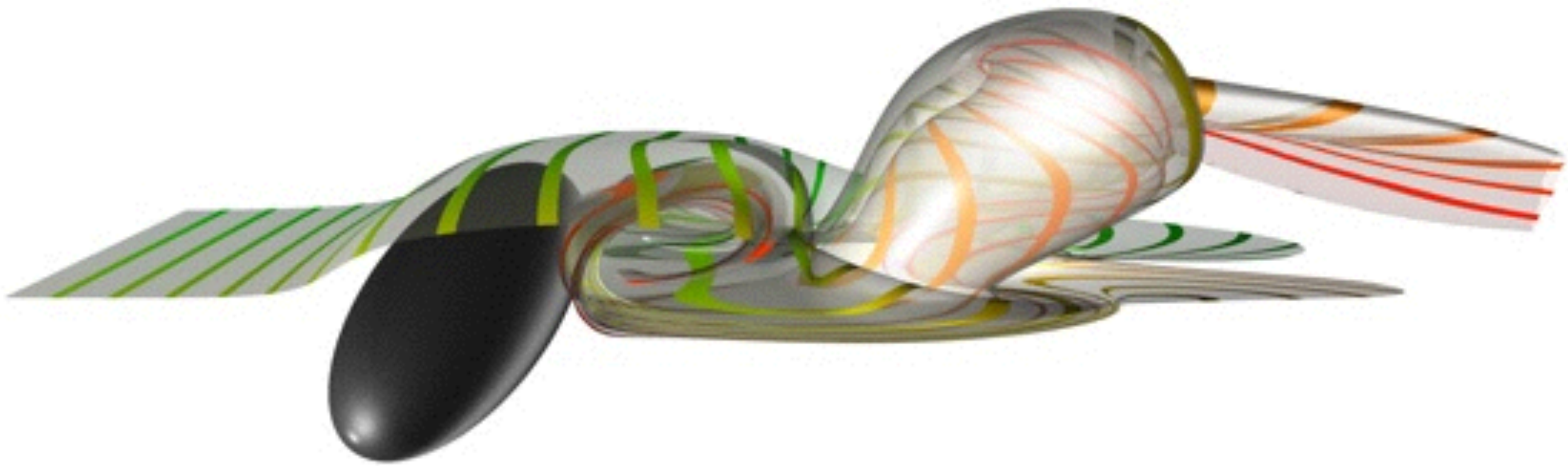
Seeding along a straight-line
Allow user exploration
[Weiskopf et al. 2007]



Seeding along the direction that is perpendicular to the flow leads to stream surface with large coverage
[Edmunds et al. EuroVis2012]

Time and streak surfaces

http://www.vacet.org/gallery/images_video/Krishnan_TimeStreakSurfaces.mp4



Hari Krishnan, Christoph Garth, Ken Joy. Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets. IEEE Visualization 2009.

Approaches to flow vis

- "How?"
  - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
  - **Texture-based (LIC, spot noise)**
  - Direct + geometry-based (hedehogs, glyphs)
  - Direct + heuristic (magnitude, Laplacian, FTLE)
  - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- "Where?"
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

Overview — Texture-Based Methods

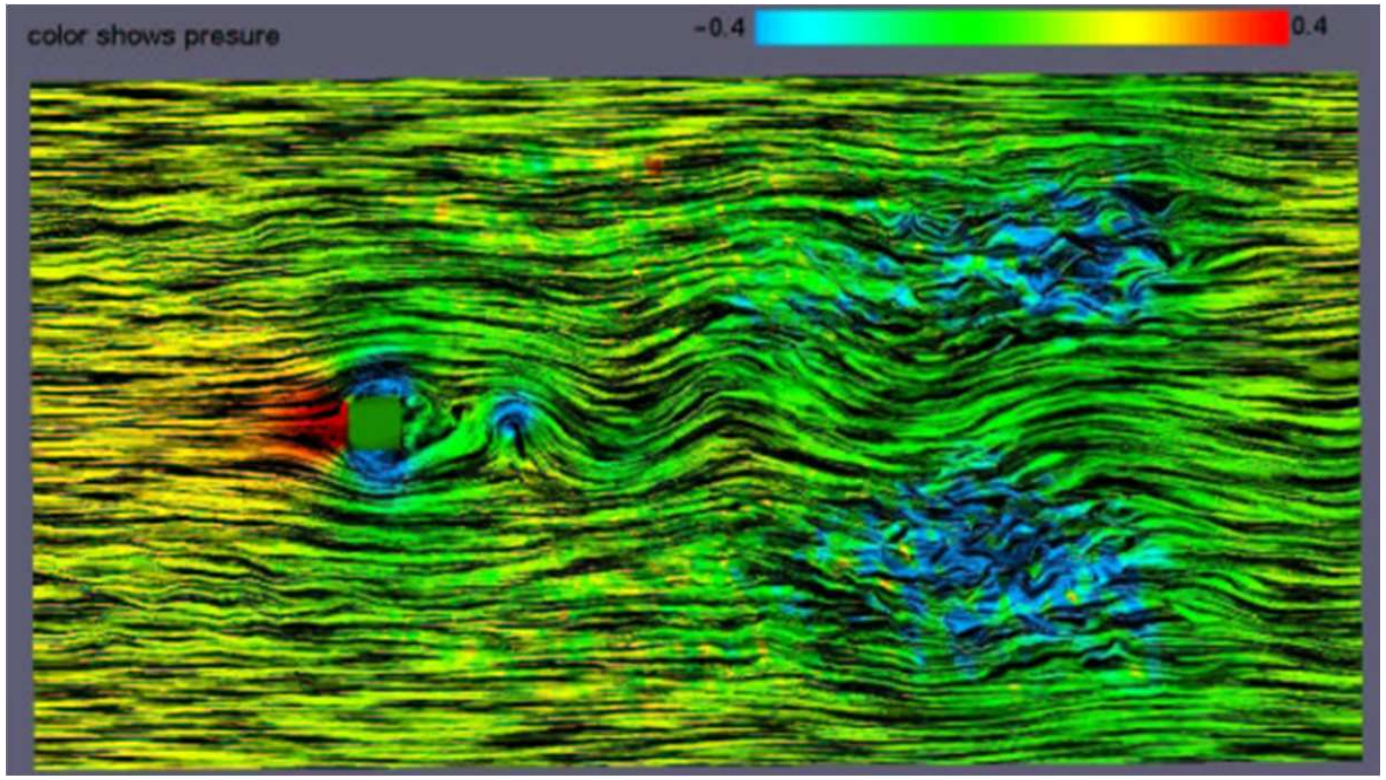
➢ Spot Noise

✧ One of the first texture-based techniques (*Van Wijk, Siggraph1991*).

✧ **Basic idea**: *distribute a set of intensity function, or spot, over the domain, that is wrapped by the flow over a small step*.

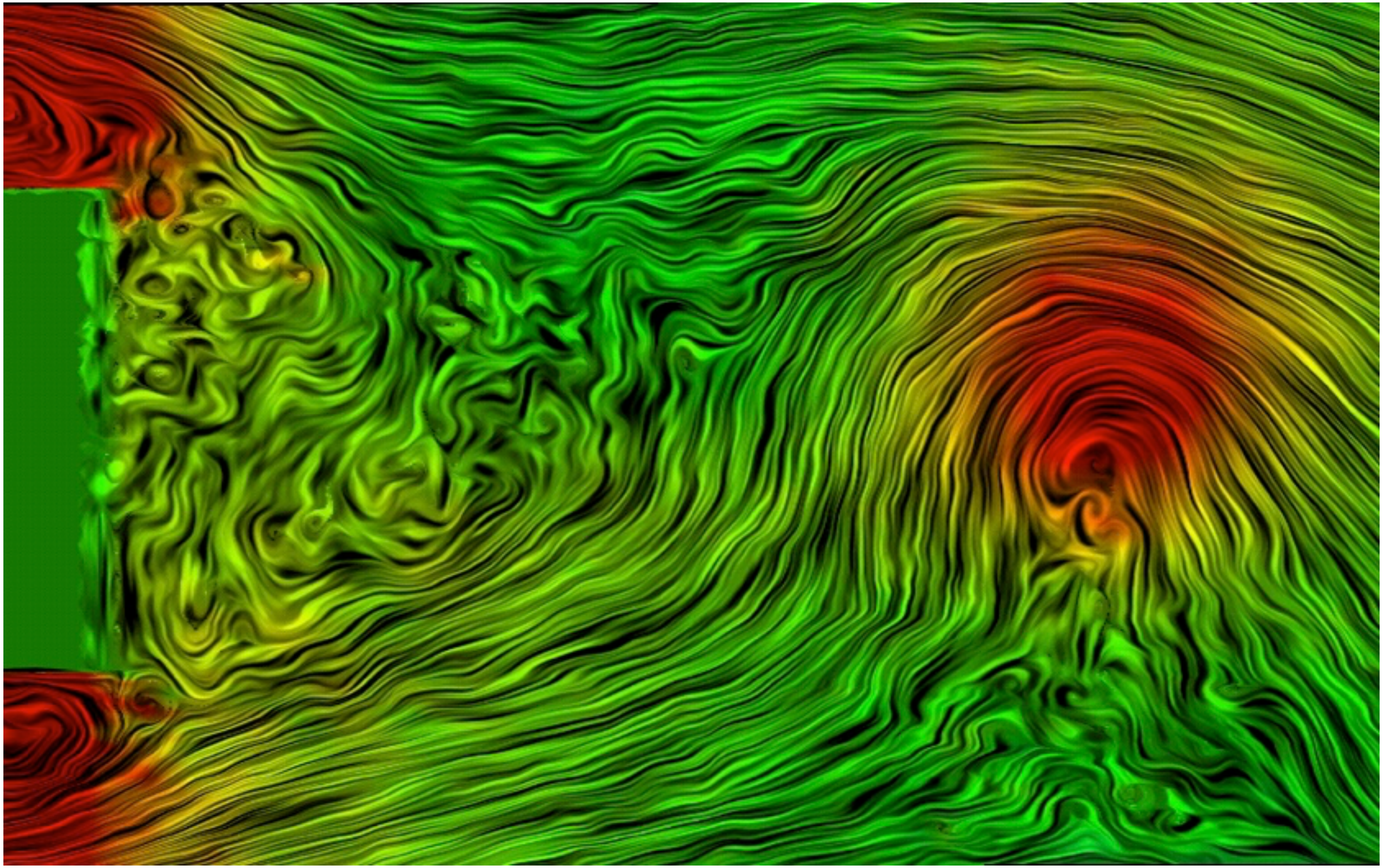
✧ **Pro**: mimic the smear effect of oil; encode magnitude; can be applied for both steady and unsteady flow.

✧ **Con**: tricky to implement; low quality; computationally expensive.



[De Leeuw and Van Liere]

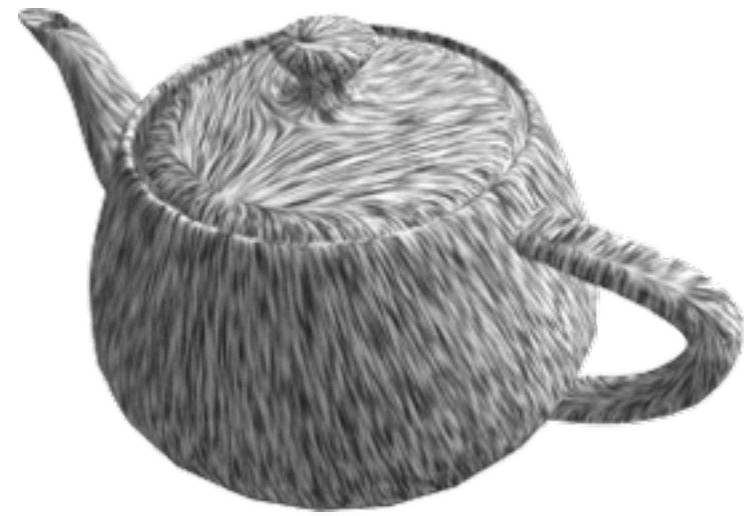
Spot noise



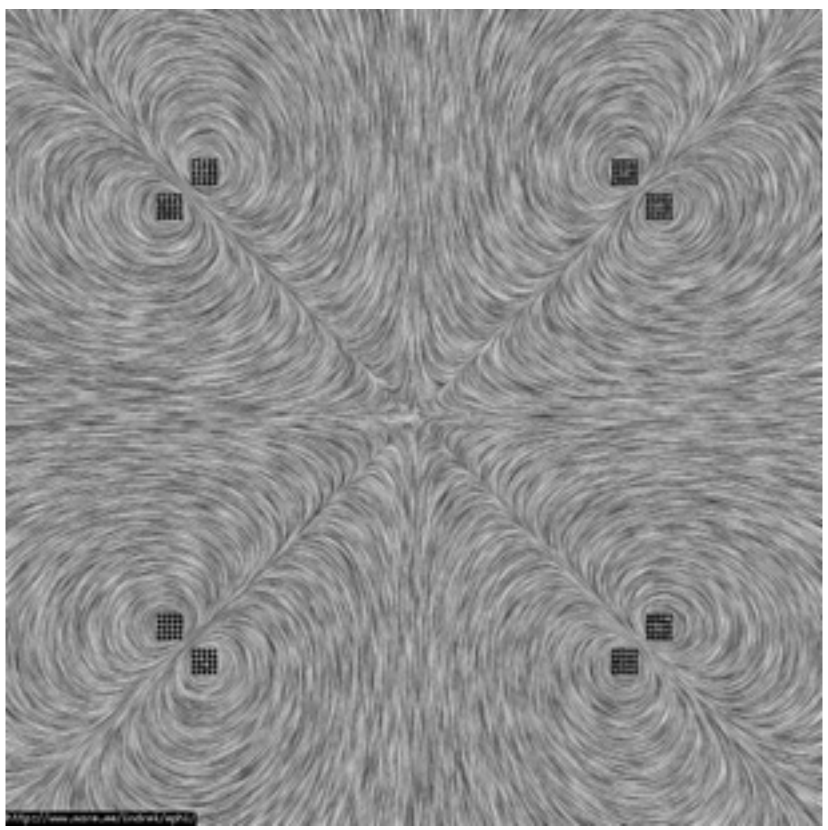
- Image: Wim de Leeuw. http://homepages.cwi.nl/~robertl/movies/flow1.mpg

www.sci.utah.edu

LIC – Line Integral Convolution

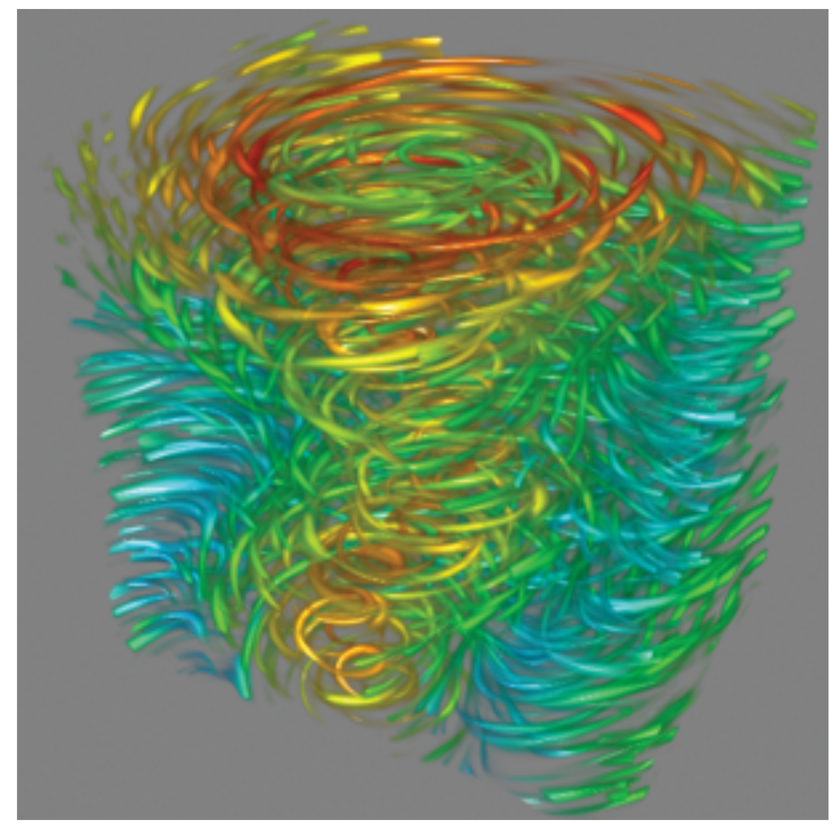


- (Cabral/Leedom, Siggraph 1993)
- A global method to visualize vector fields



2D vector field

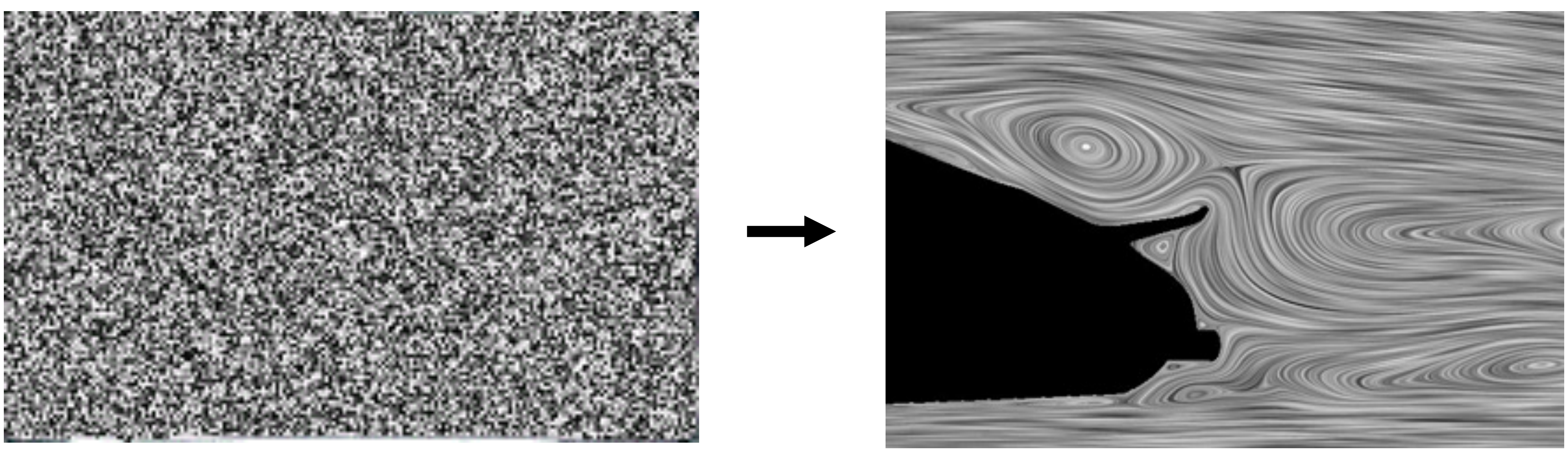
vector field on surface
(often called 2.5D)



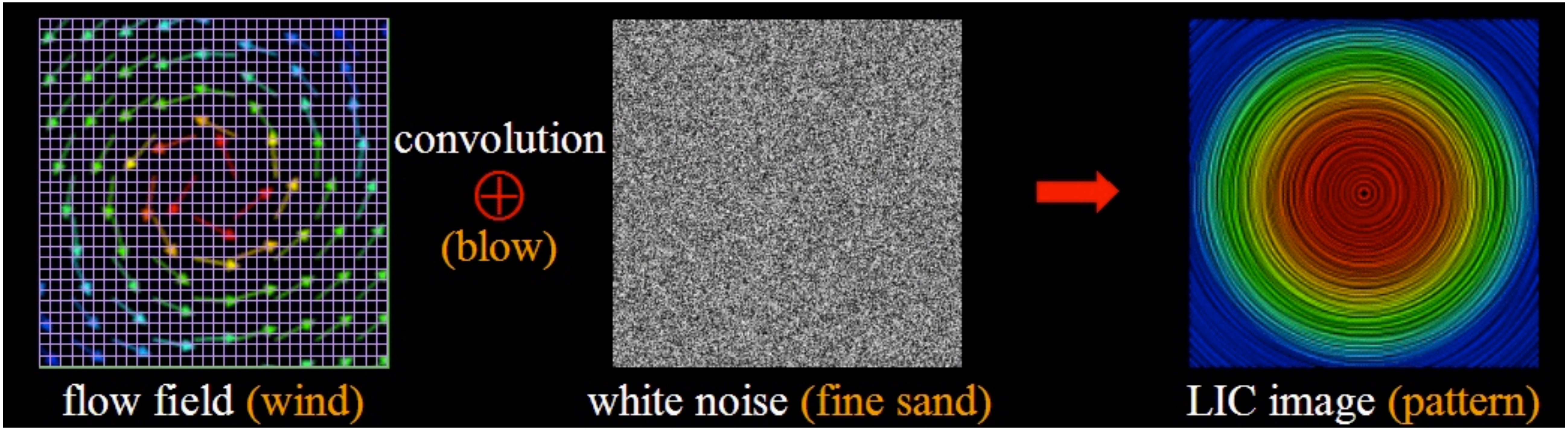
3D vector field

Idea of LIC

- **Global** visualization technique; not only one particle path
- Start with a **random texture**
- **Smear** out this texture along the path lines in a vector field, results in
  - **Low correlation** of intensity values **between** neighboring lines,
  - But **high correlation along** them

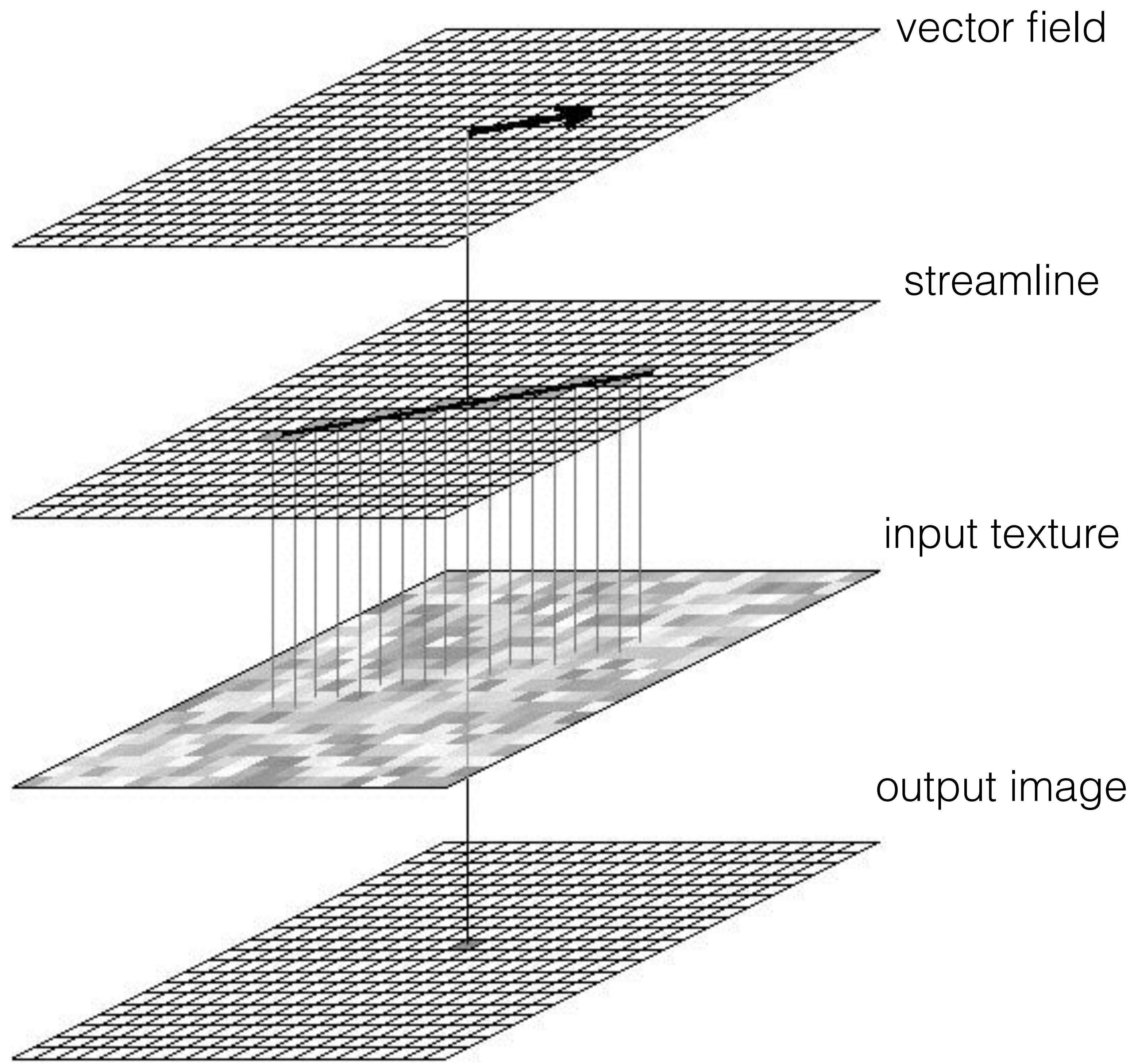


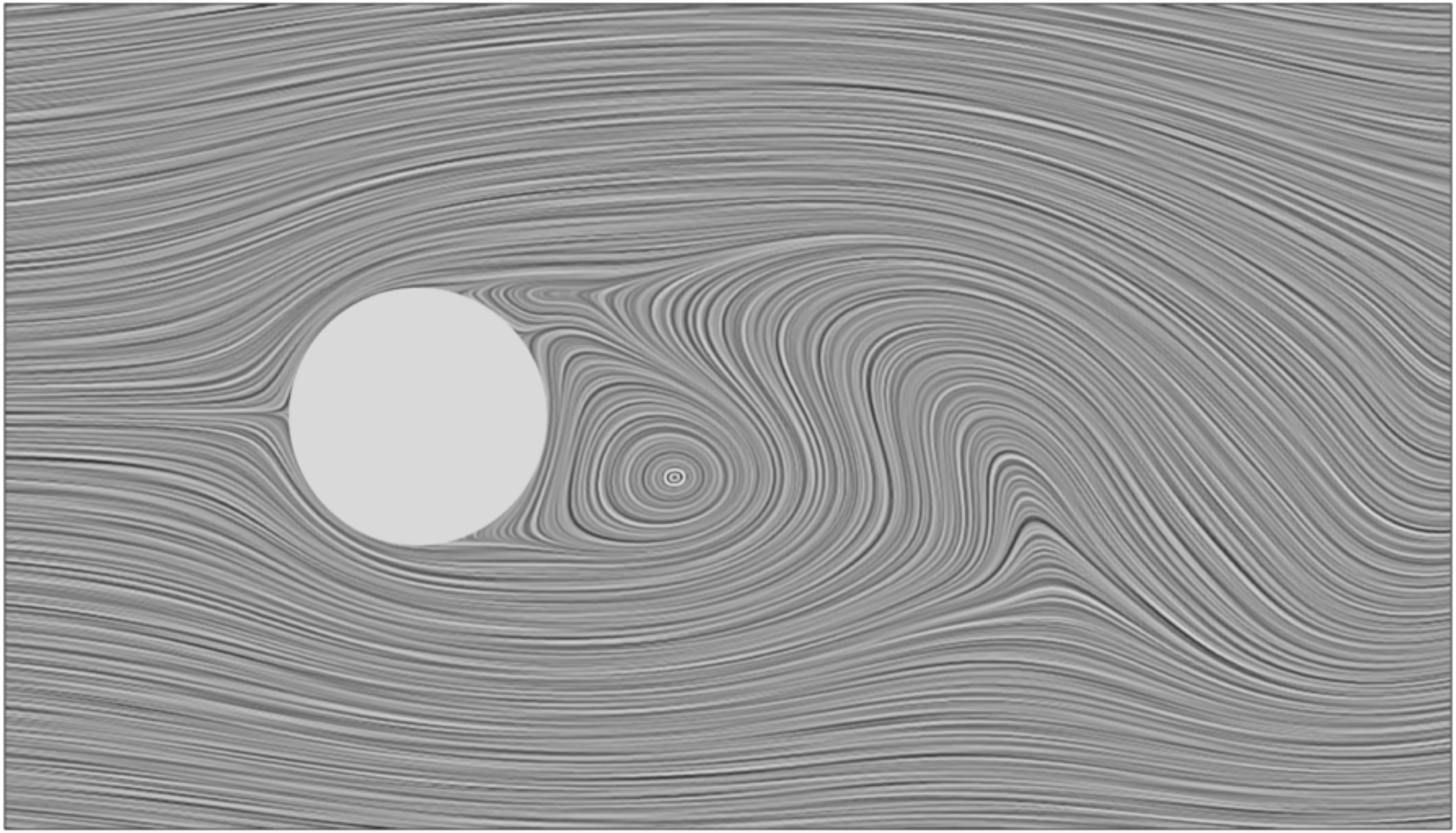
Idea of LIC


convolution
(blow)
flow field (wind)
white noise (fine sand)
LIC image (pattern)

Algorithm for 2D LIC

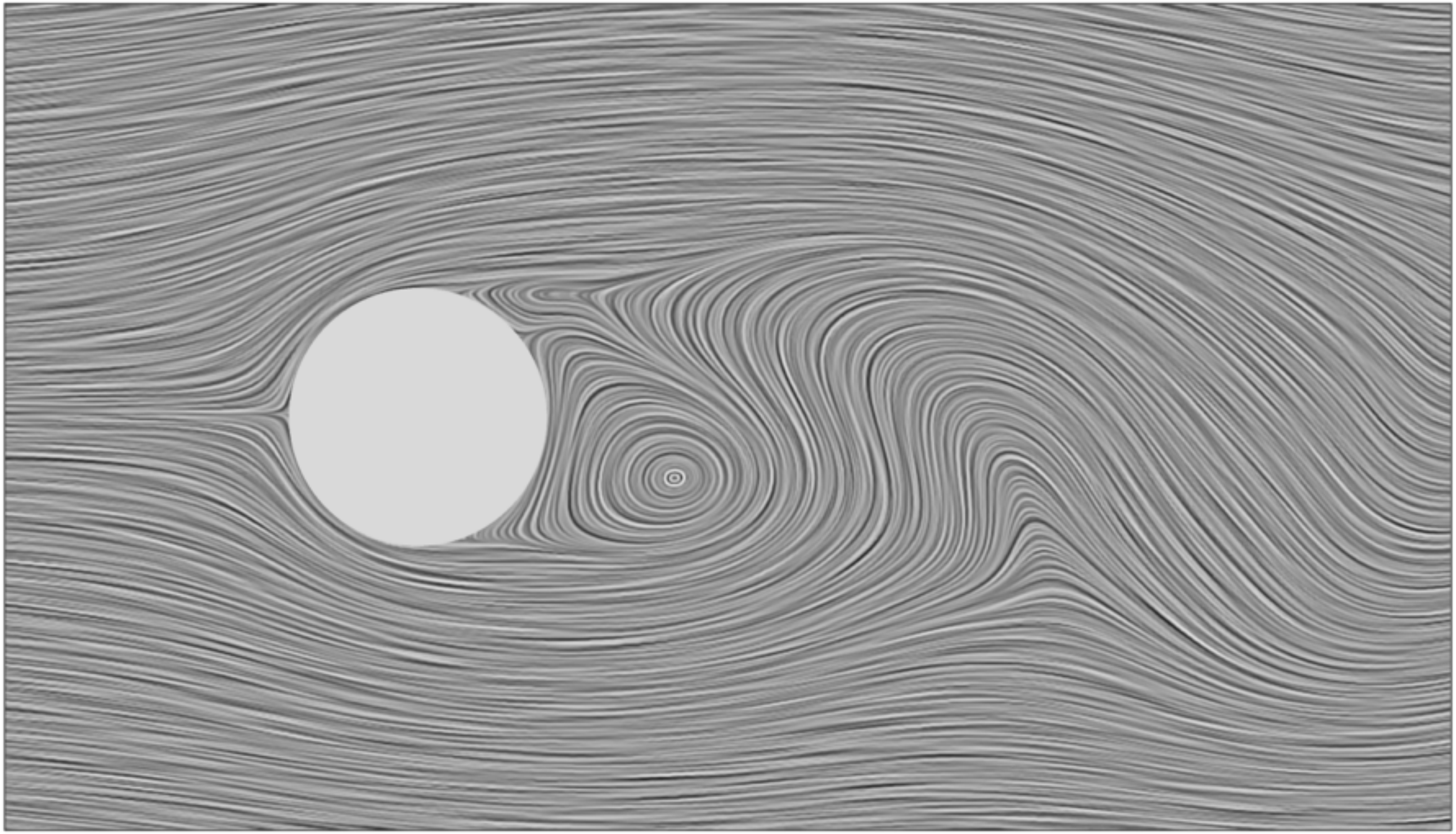
- Convolve a random texture along the streamlines





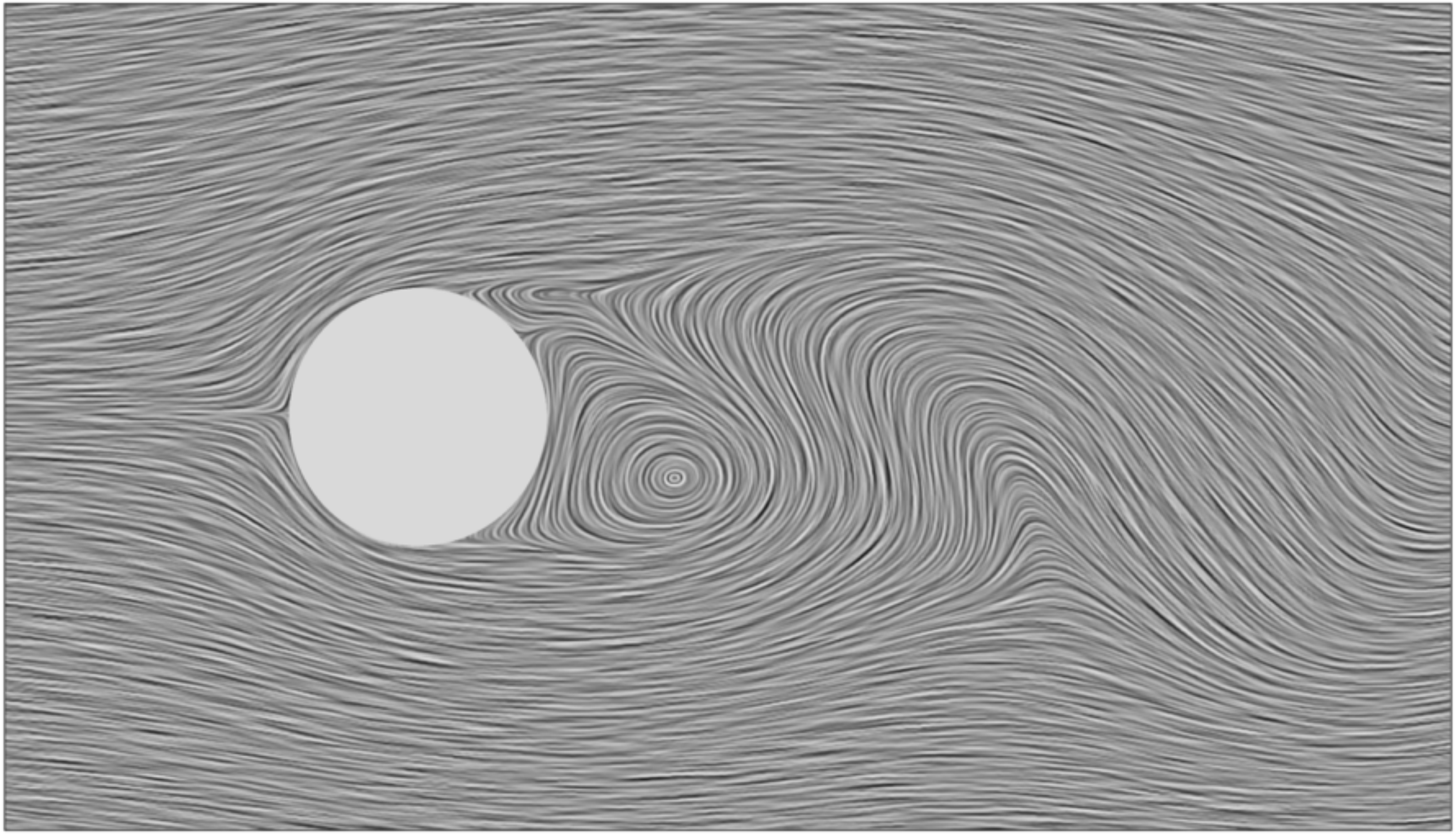
Filter length influences the quality of LIC images
filter length = 100

2D flow behind a cylinder



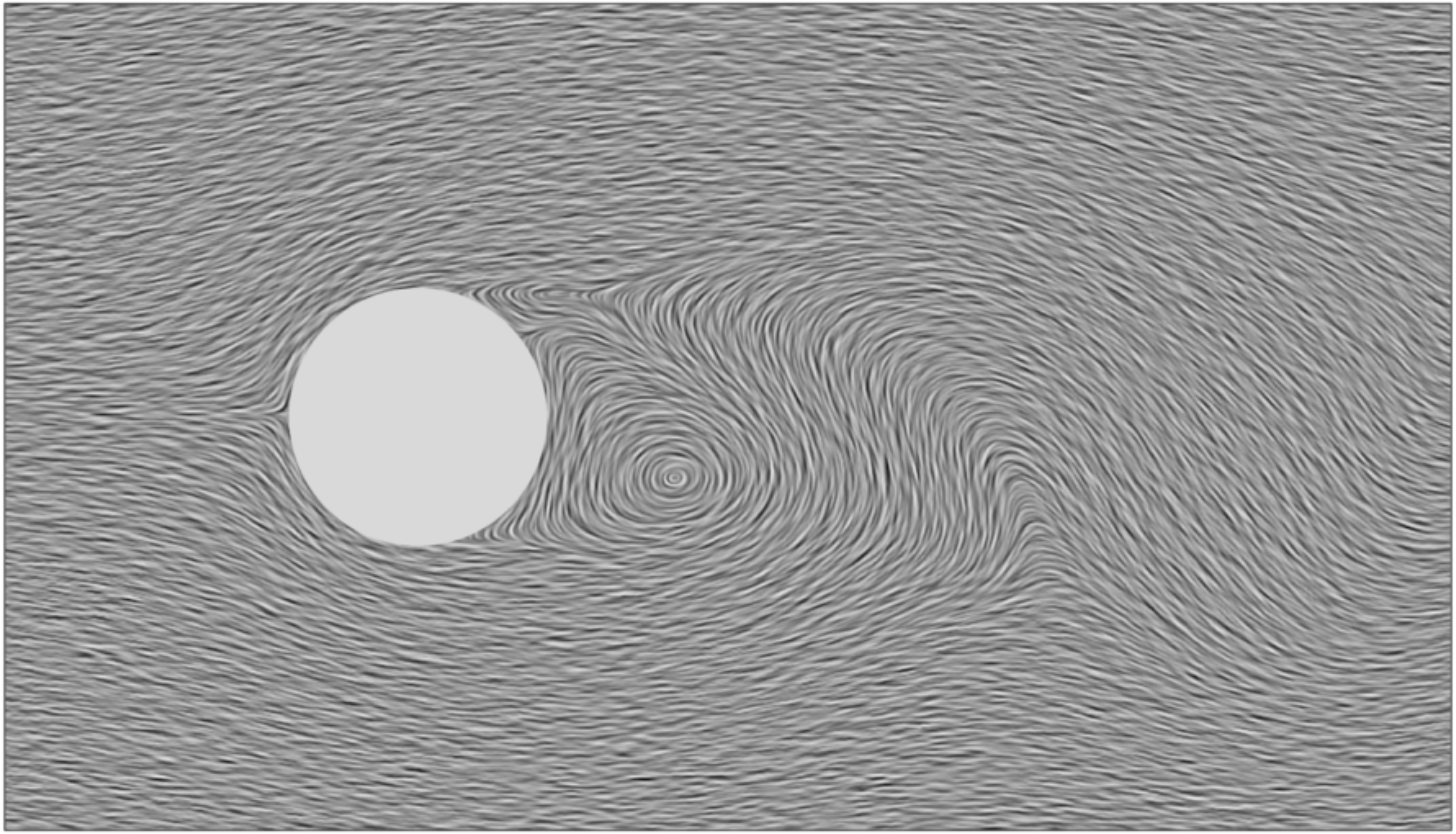
Filter length influences the quality of LIC images
filter length = 50

2D flow behind a cylinder



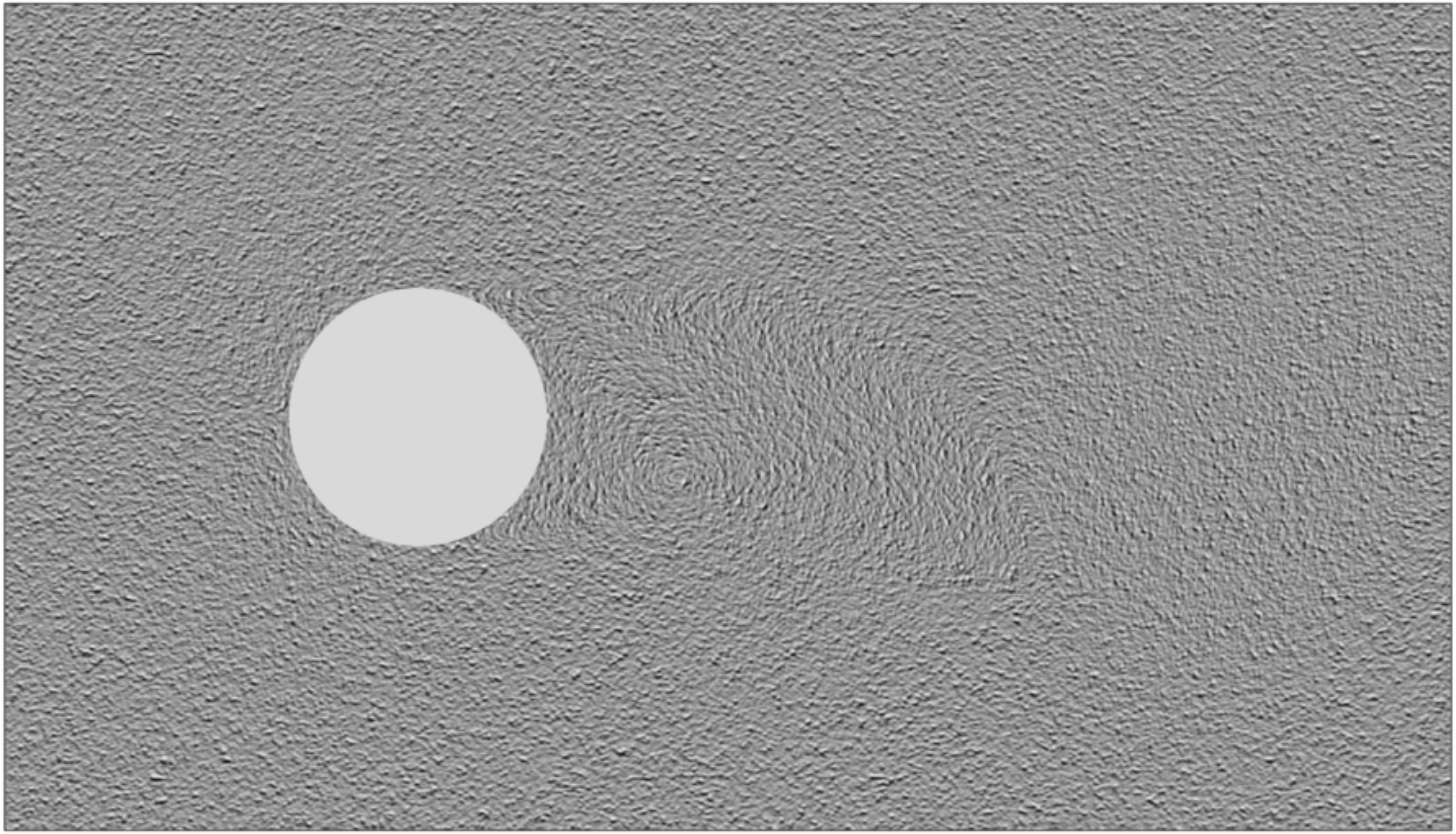
Filter length influences the quality of LIC images
filter length = 25

2D flow behind a cylinder



Filter length influences the quality of LIC images
filter length = 10

2D flow behind a cylinder

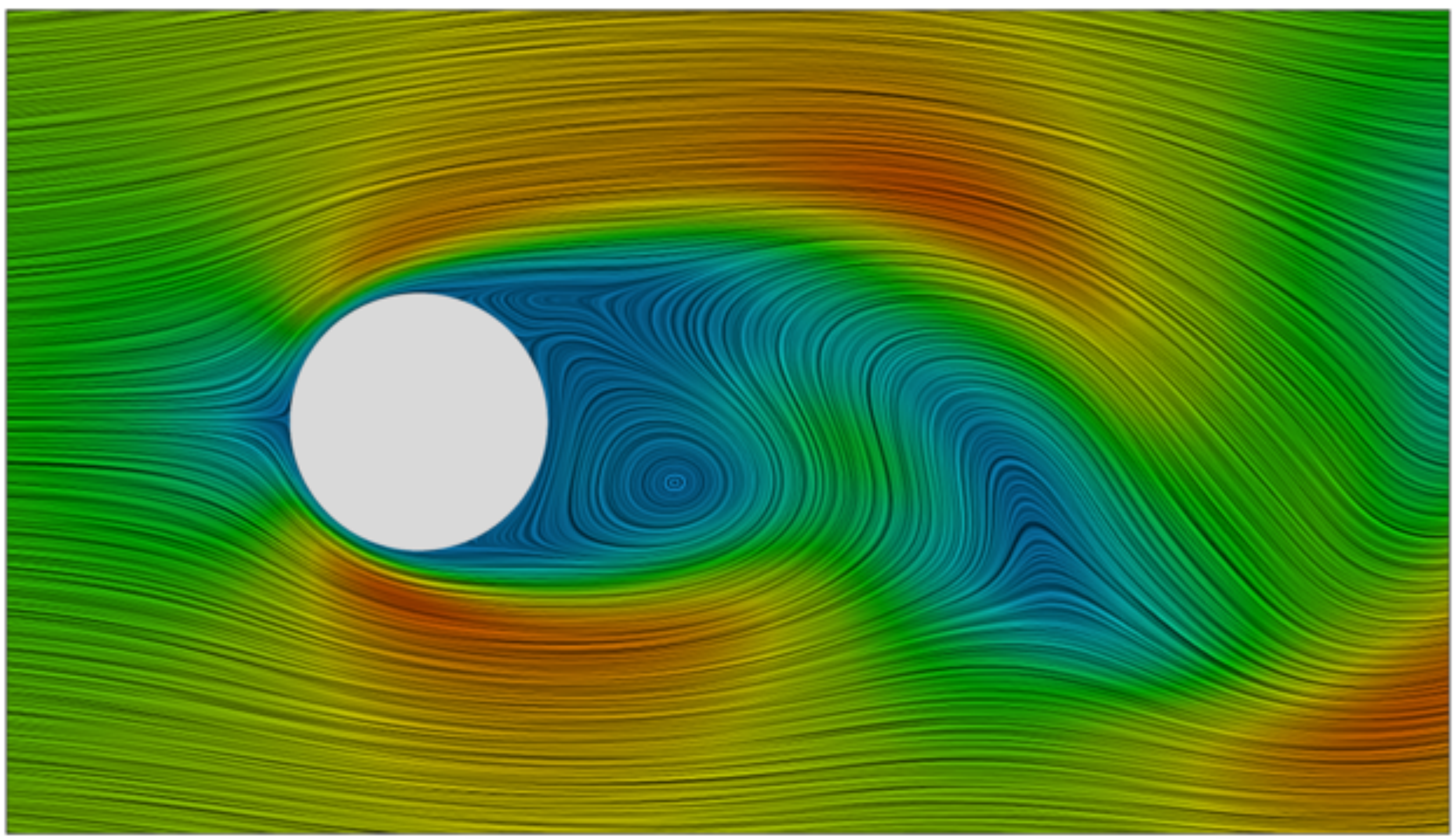


Filter length influences the quality of LIC images
filter length = 1

2D flow behind a cylinder

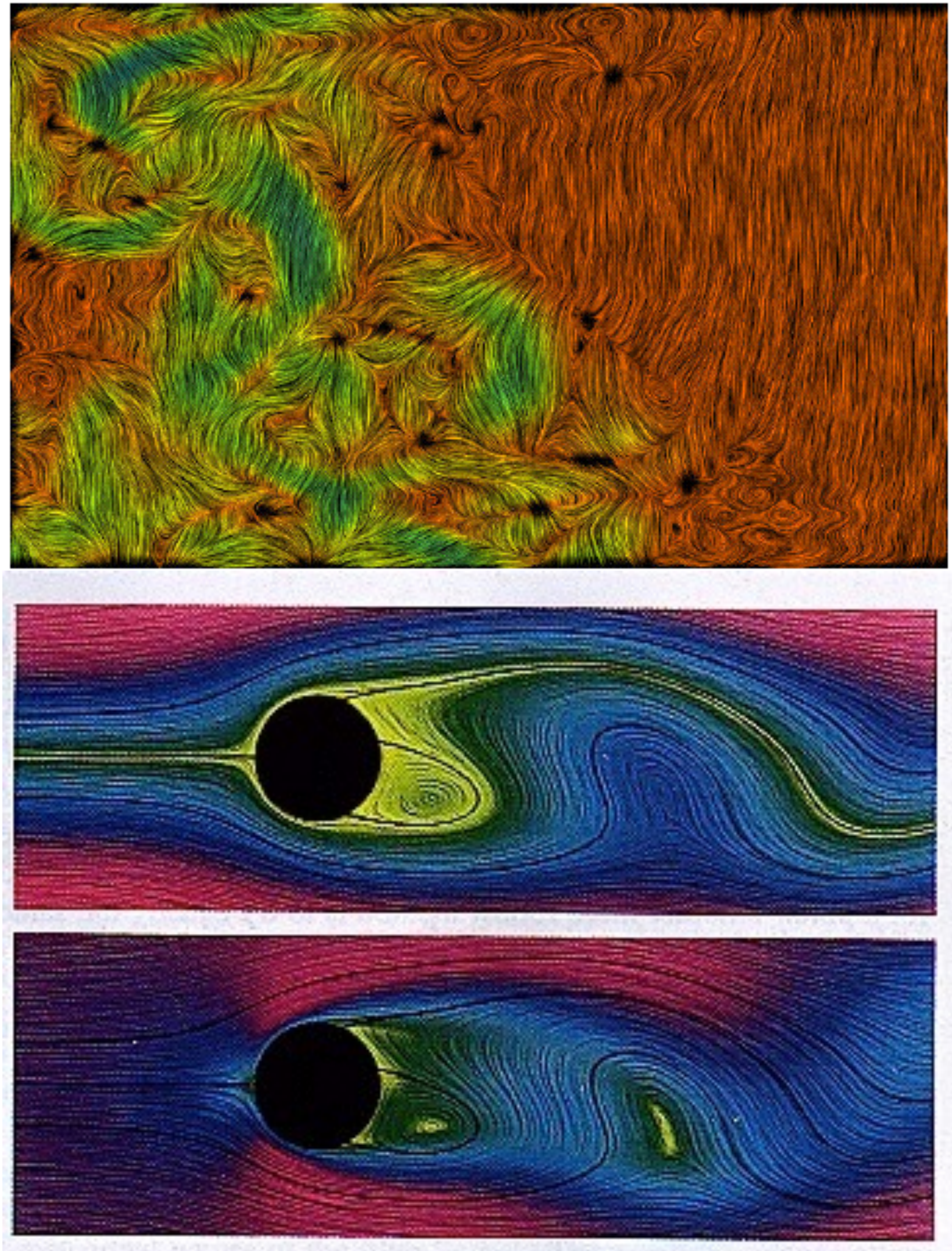
LIC Color Coding

- Usually, LIC does not use the color channel
  - Could use color to encode scalar quantities



Velocity magnitude encoded using color

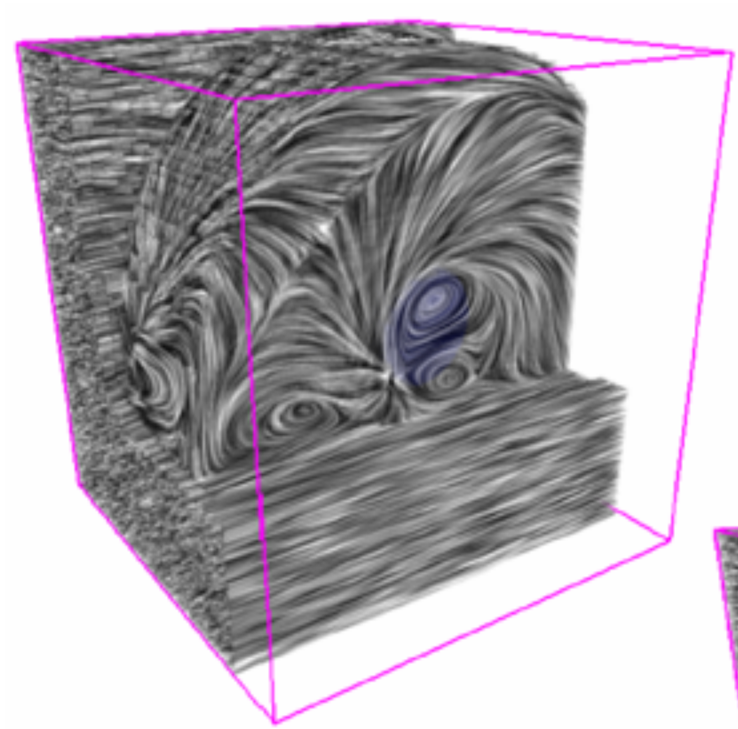
2D flow behind a cylinder

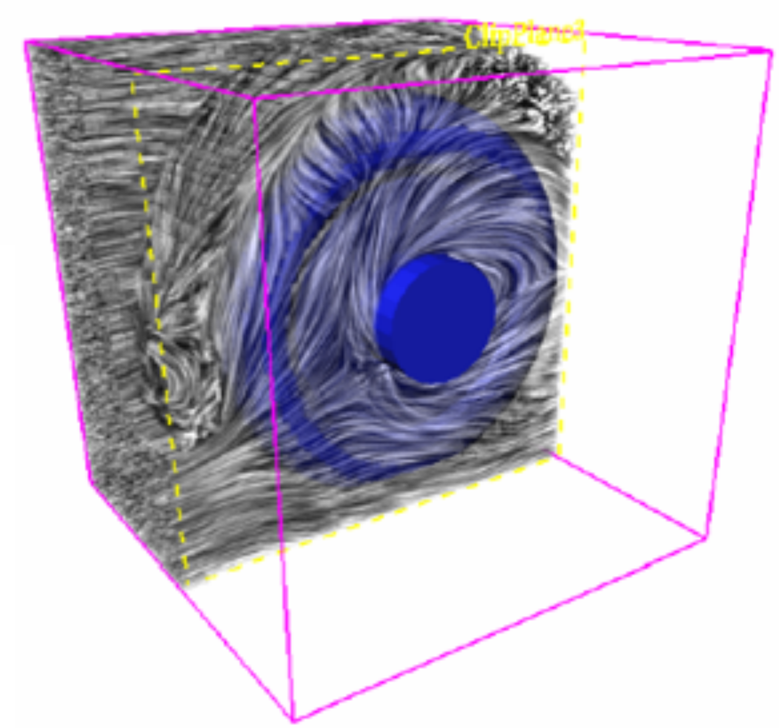


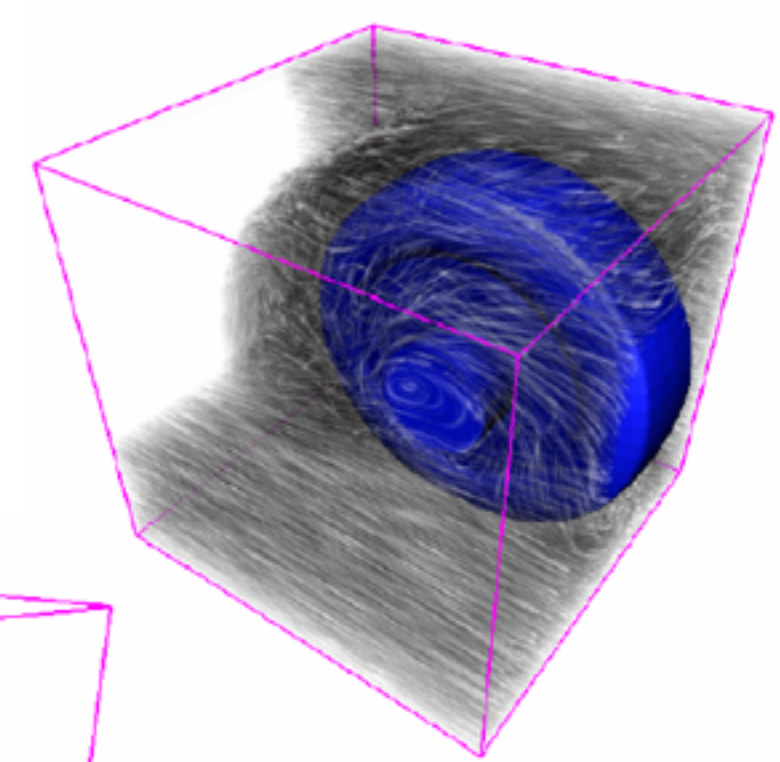
LIC and color coding of velocity magnitude

LIC for 3D Flows

- LIC concept easily extendable to 3D
- Problem: rendering!







3D LIC can only reveal interesting structures if some data is discarded.

Images courtesy of [Rezk-Salama et al. 99]

Overview — Texture-Based Methods

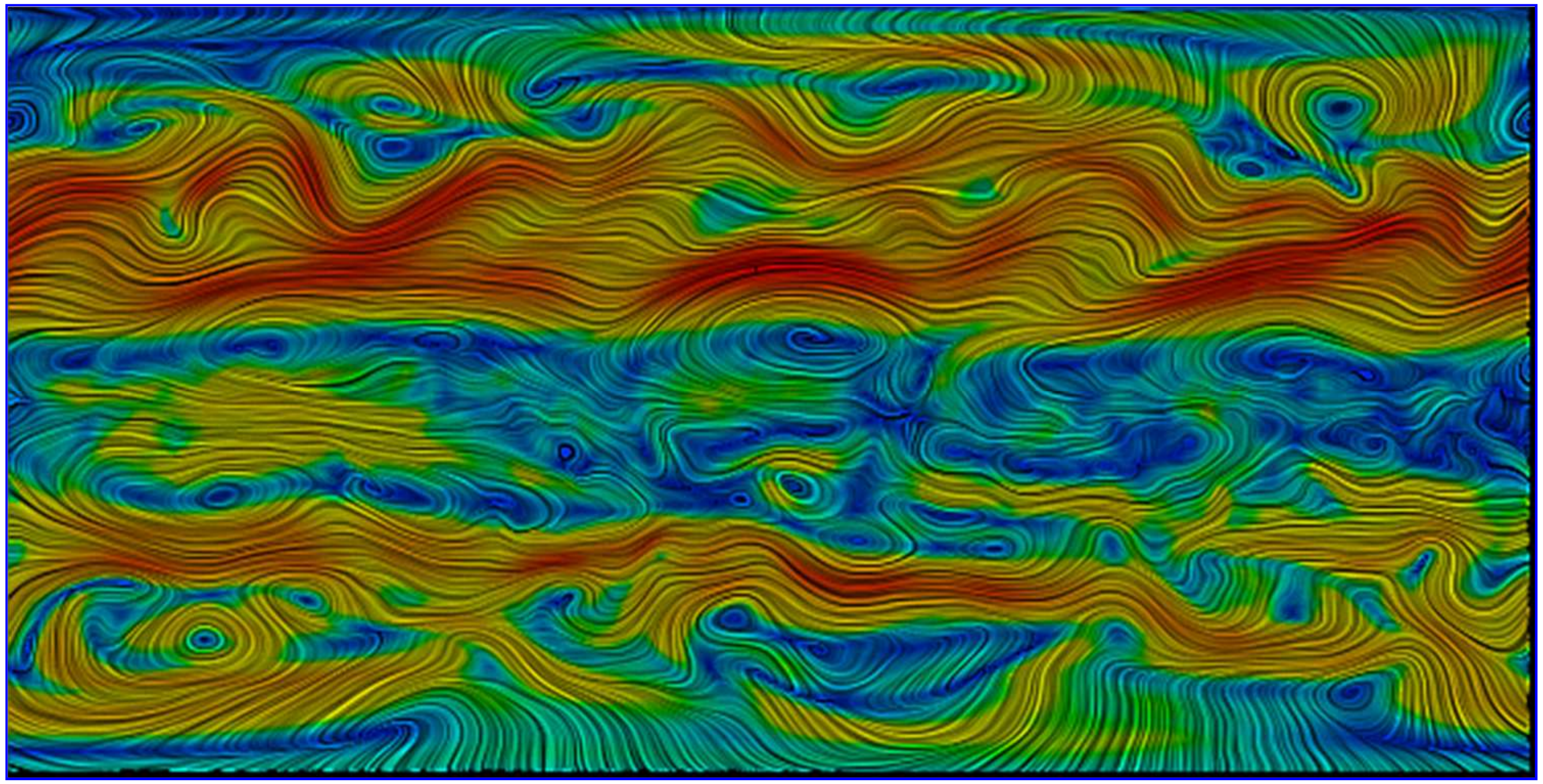
➢ Unsteady Flow LIC (UFLIC)

✧ The first texture-based **unsteady** flow visualization method (by *Han-Wei Shen and David Kao, IEEE Visualization 97 & IEEE TVCG 98*).

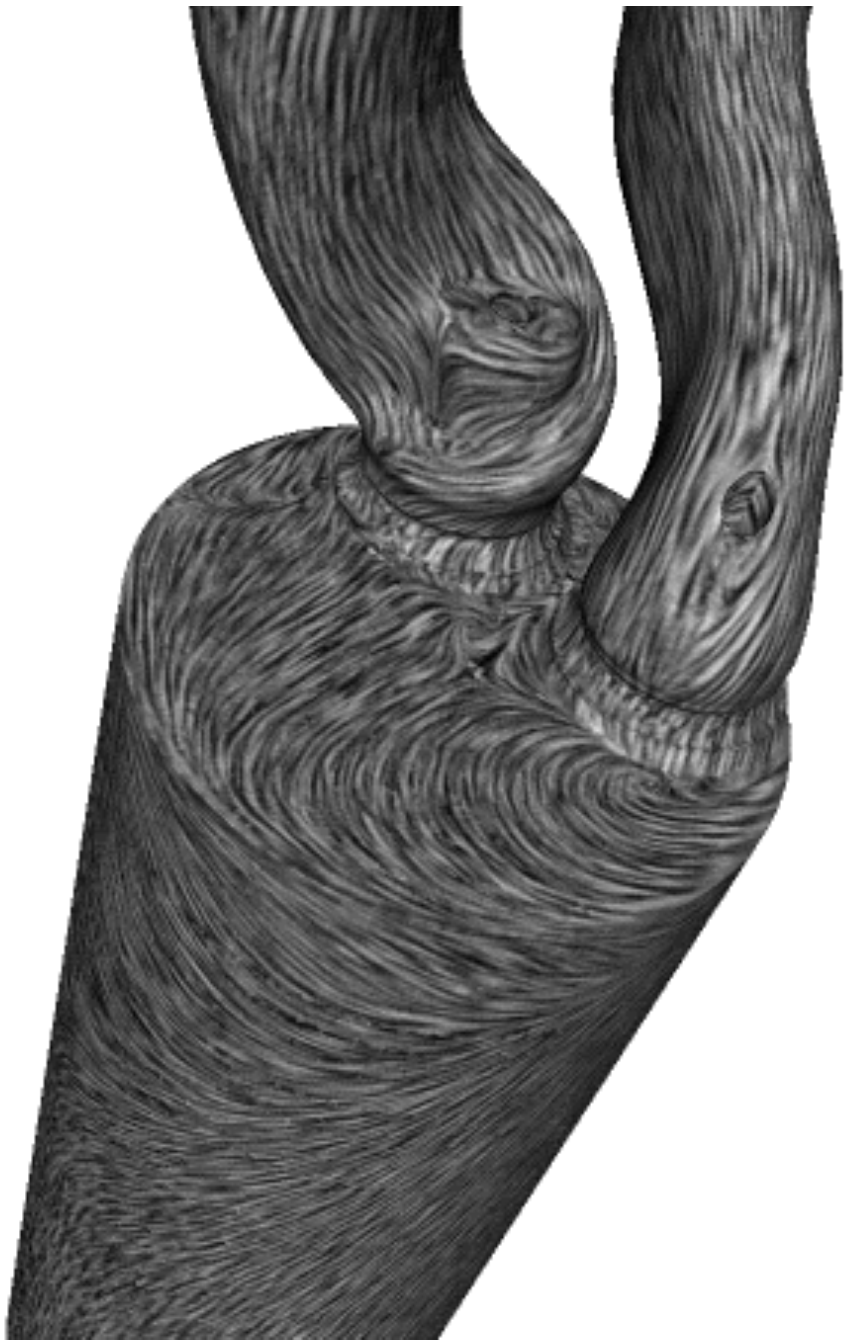
✧ **Basic idea:** Time-accurately scatters particle values of *successively fed-forward textures* along **pathlines** *over several time steps* to convey the footprint / contribution that a particle leaves at downstream locations as the flow runs forward.

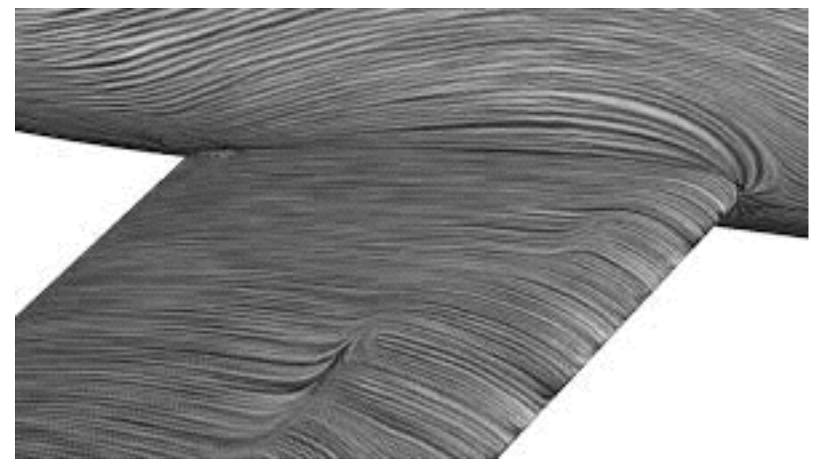
✧ **Pro**: High temporal coherence & high spatial coherence & hardware-independent.

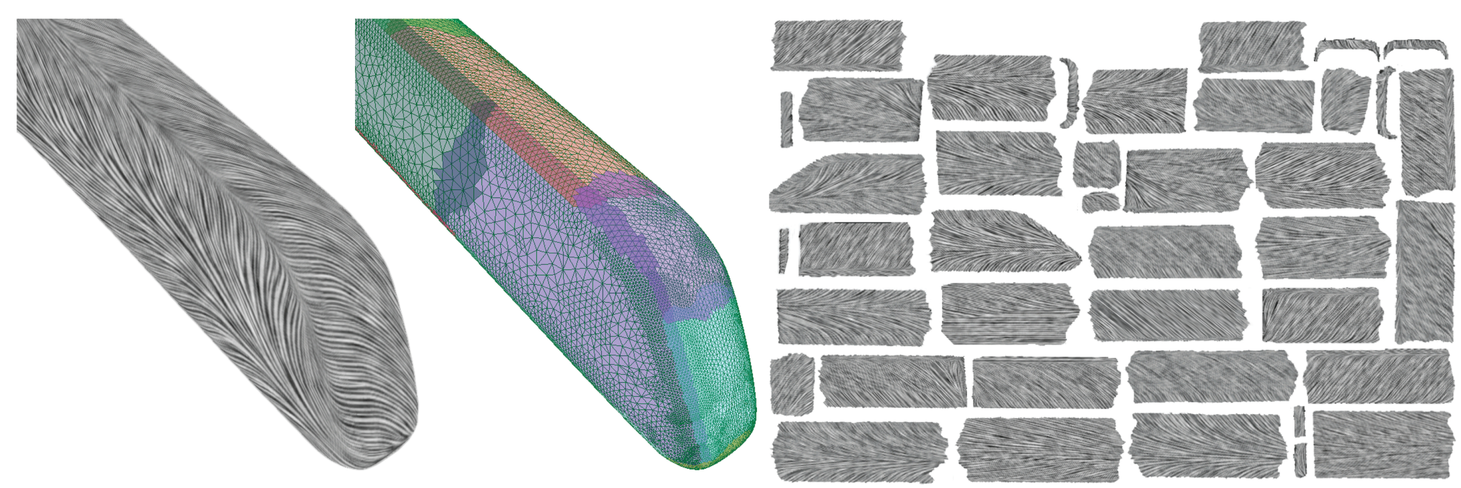
✧ **Con**: Low computational performance due to *multi-step* (≈ 100) *pathline integration.*



GPU-accelerated UFLIC on arbitrary surfaces





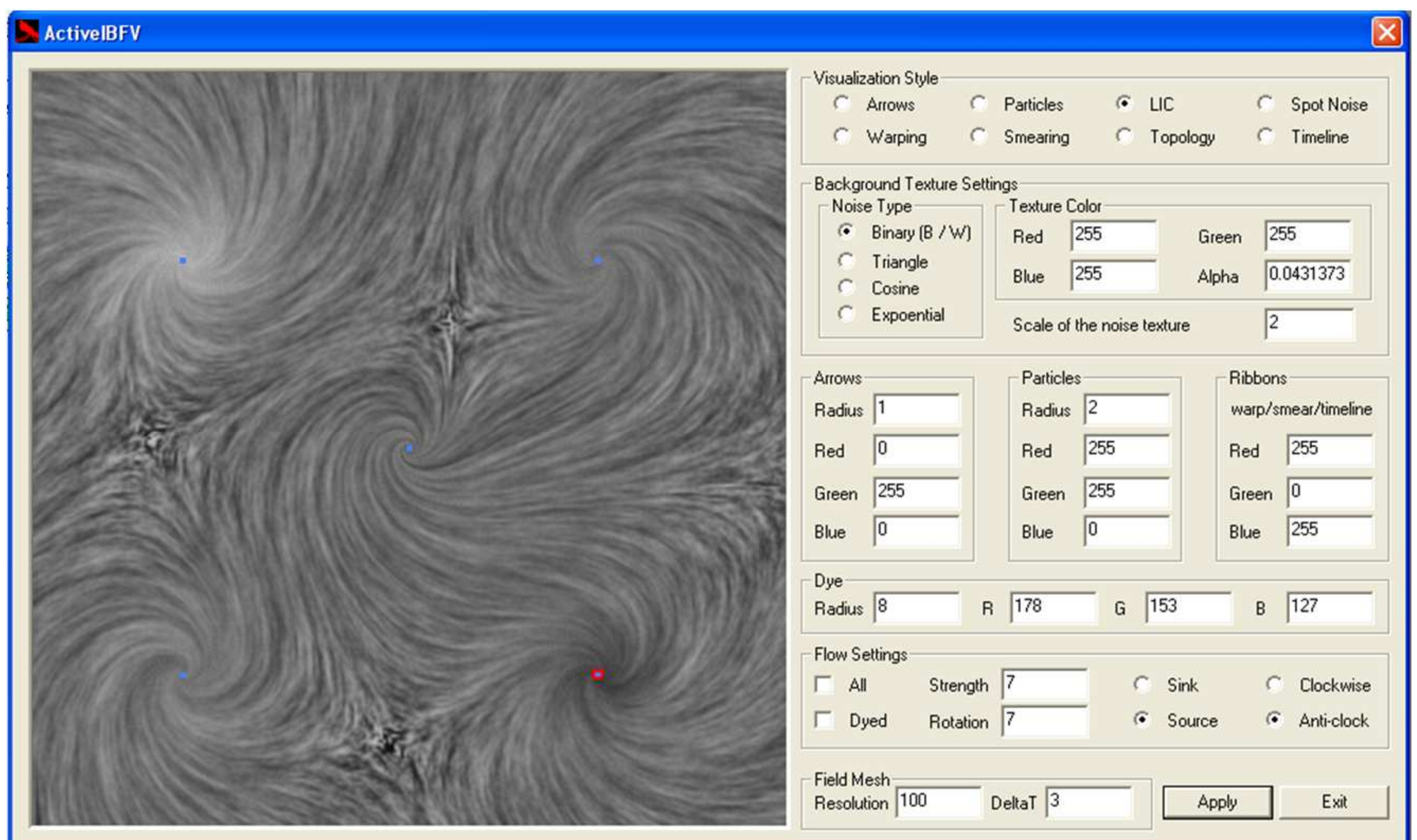


Flow Charts: Visualization of Vector Fields on Arbitrary Surfaces. G Li, X Tricoche, D Weiskopf, C Hansen. IEEE TVCG 2008.

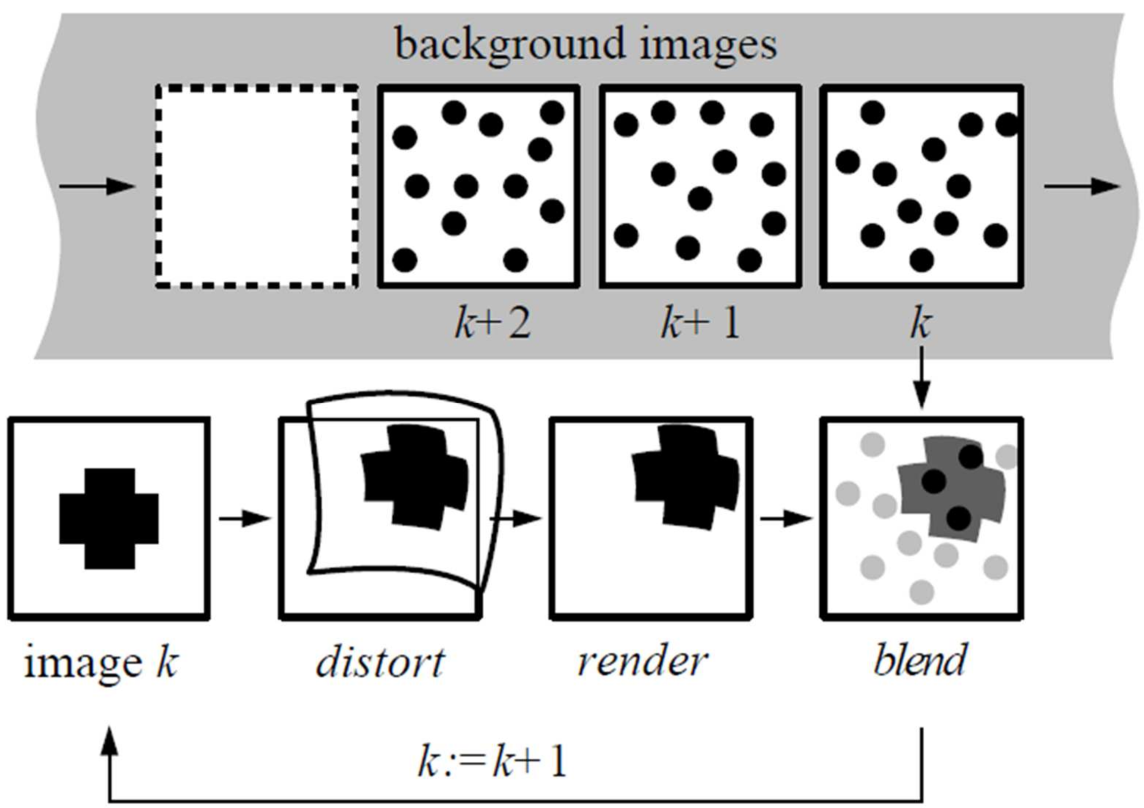
Overview — Texture-Based Methods

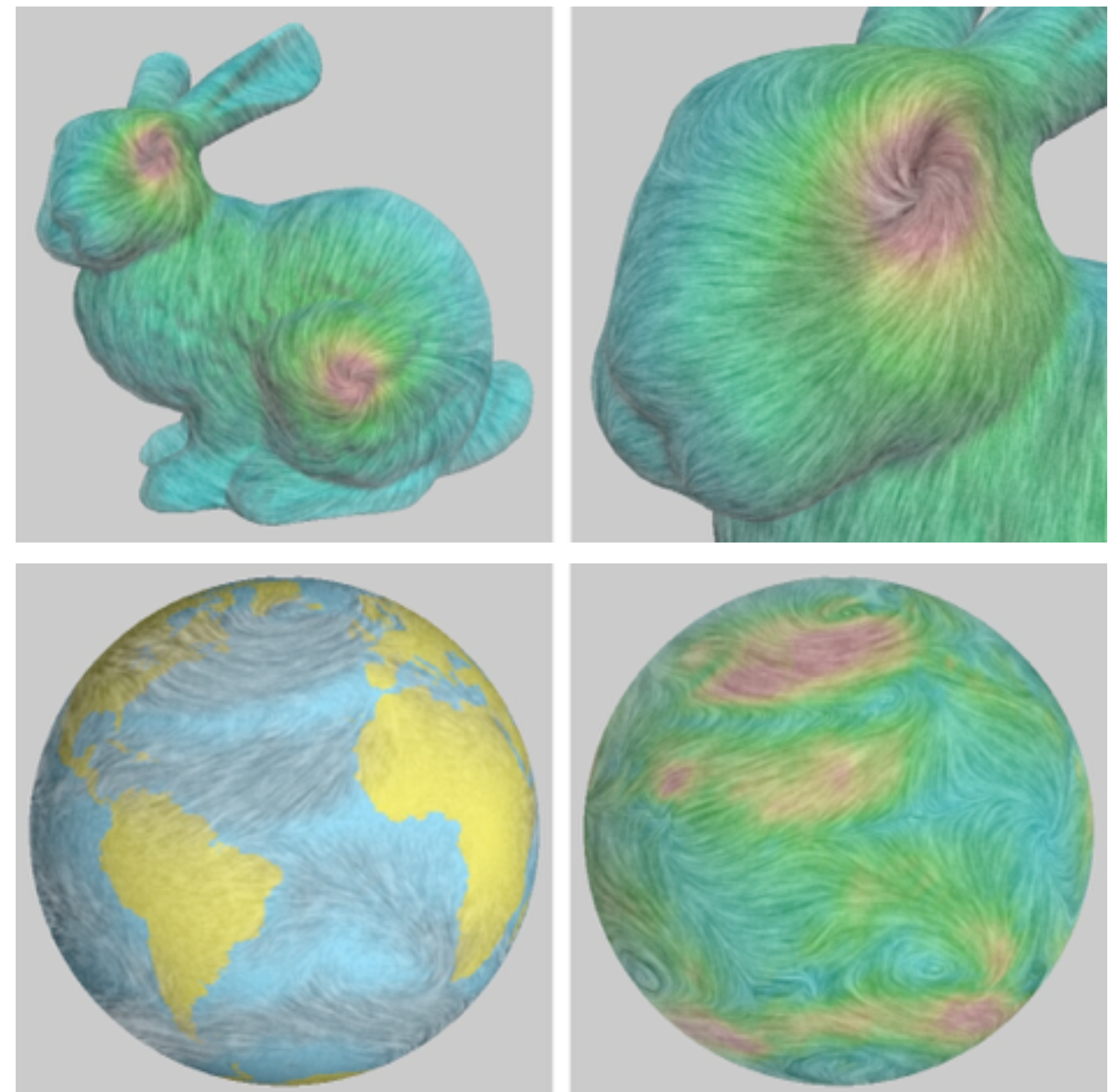
➢ Image-Based Flow Visualization (IBFV)

✧ One of the most versatile and the easiest-to-implement hardware-based methods (by *Jarke J. van Wijk, SIGGRAPH02*).
✧ **Basic idea:** Designs a sequence of *temporally-spatially low-pass filtered* noise textures and cyclically blends them with an iteratively advected (using *forward single-step pathline integration*) image (which is initially a BLACK rectangle).
✧ **Pro:** Interactive frame rates and easy simulation of many visualization techniques.
✧ **Con:** Good temporal coherence and insufficient spatial coherence (*noisy or blurred*).



IBFV: Image-Based Flow Visualization (Advect Dye in Image-Space)

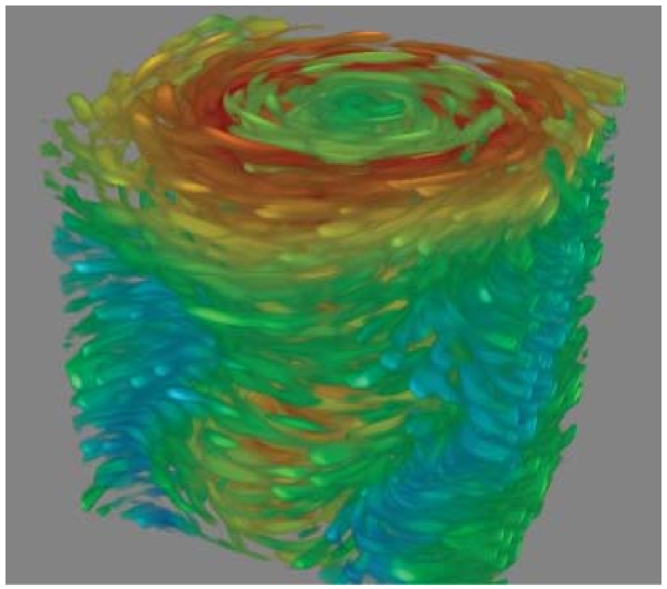




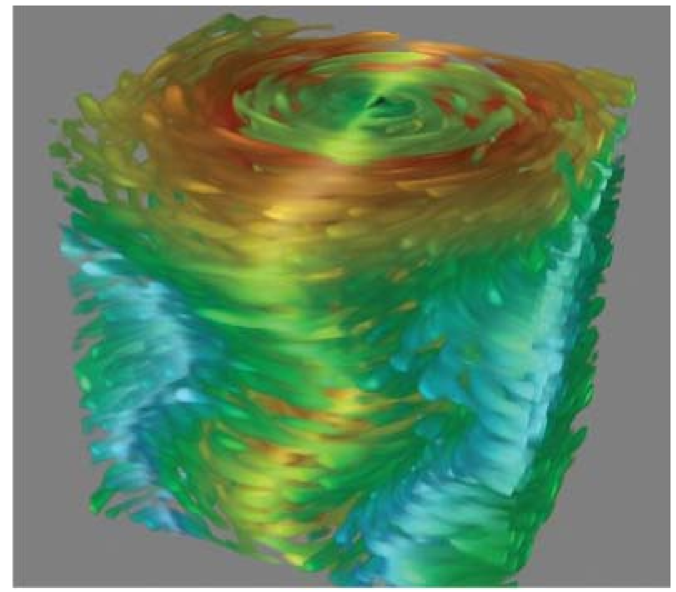
http://www.win.tue.nl/~vanwijk/ibfv/

http://www.win.tue.nl/~vanwijk/ibfvs/

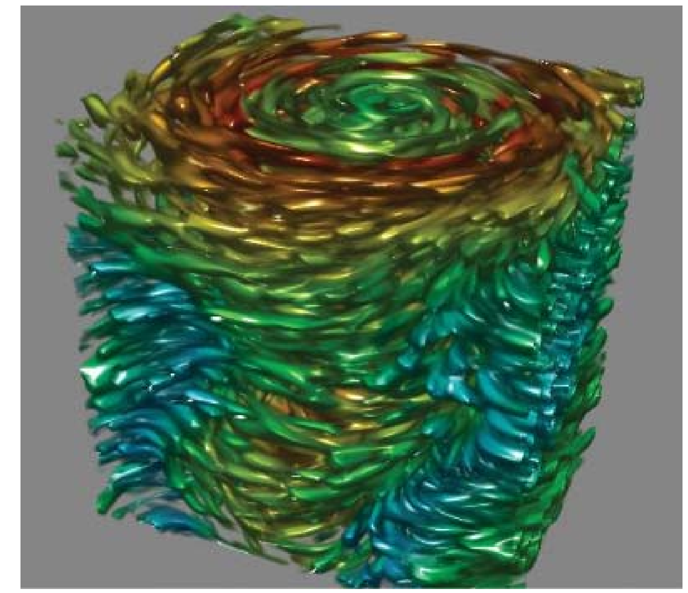
Recent Advances in 3D Texture-based Method



without illumination

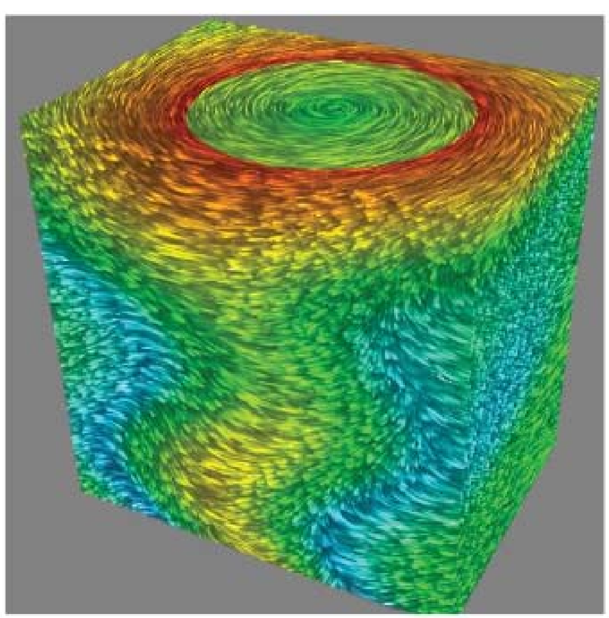


with illumination

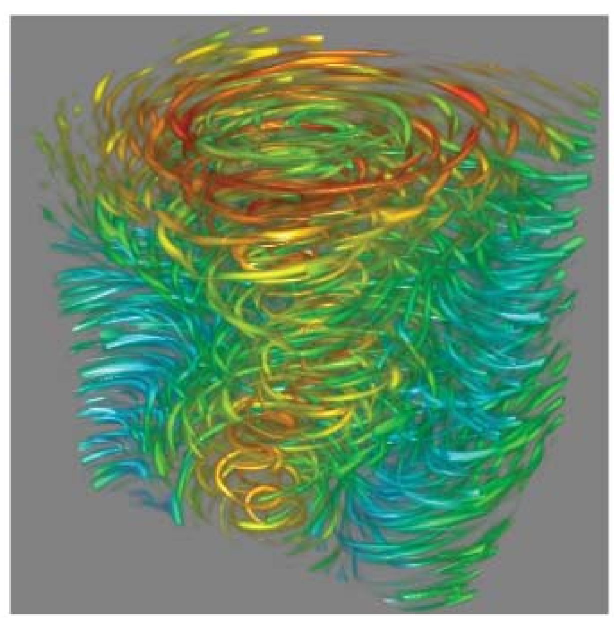


Gradient-based illumination

Codimension-2 illumination

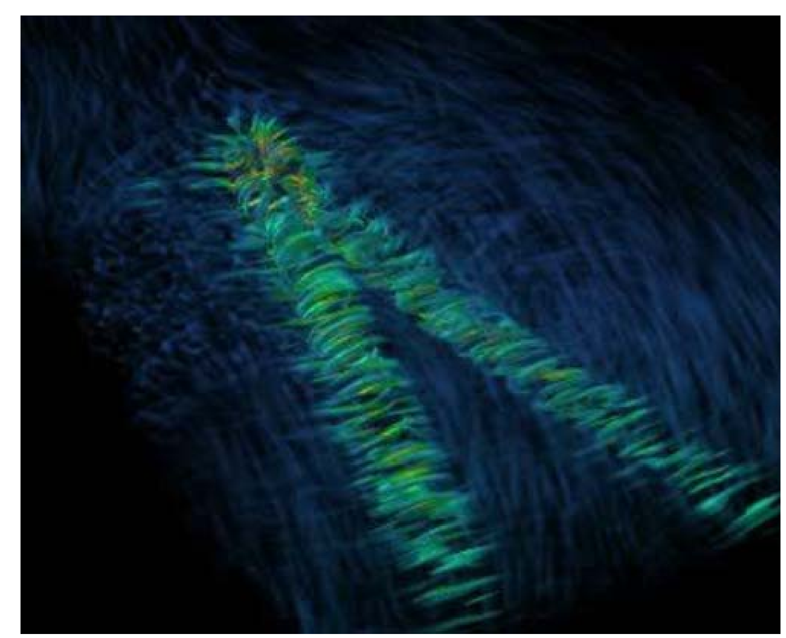


Dense (white noise)



Sparse noise

Different seeding strategies



Feature enhancement

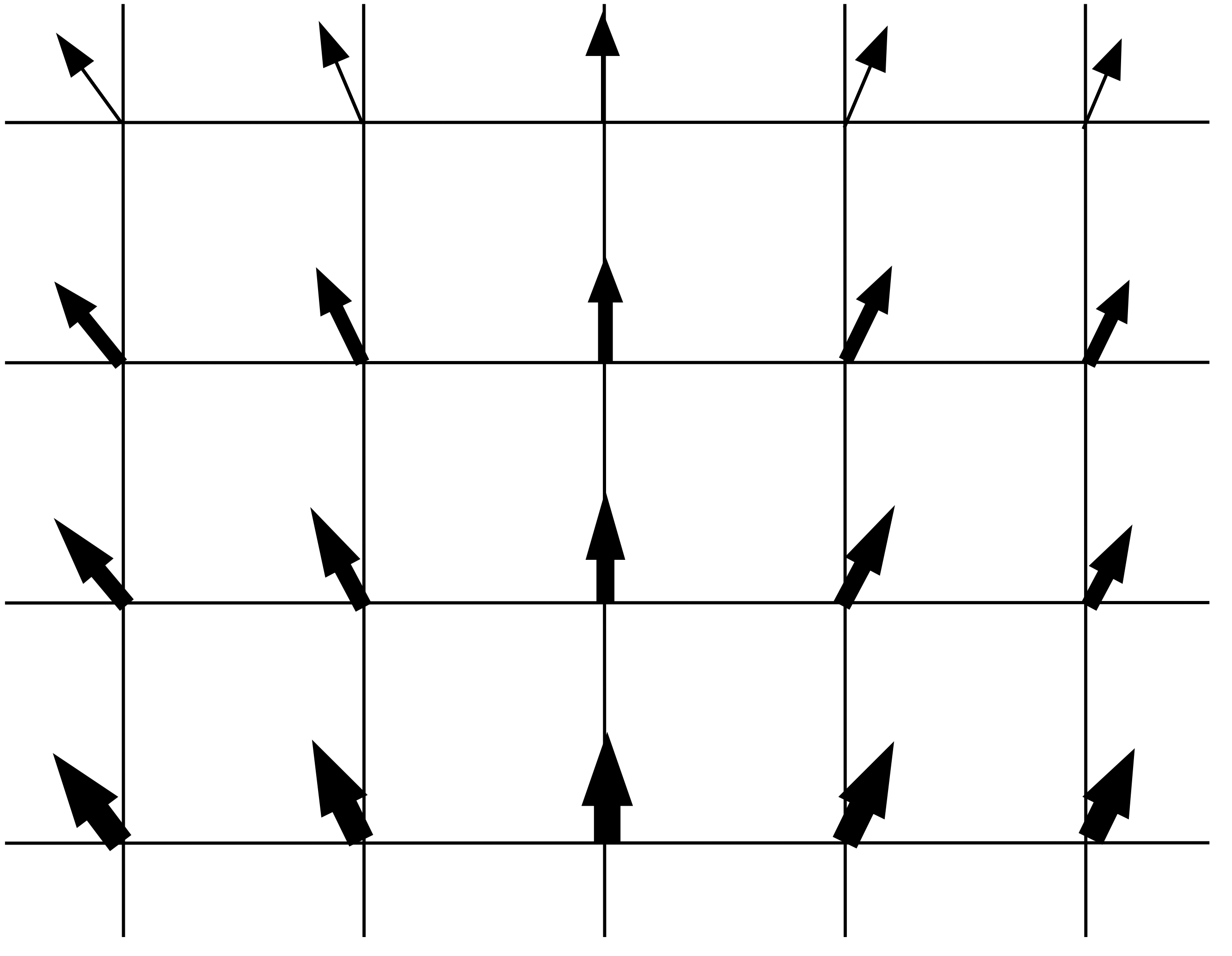
[Falk and Weikopf 2008]

Approaches to flow vis

- "How?"
  - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
  - Texture-based (LIC, spot noise)
  - **Direct + geometry-based (hedehogs, glyphs)**
  - Direct + heuristic (magnitude, Laplacian, FTLE)
  - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- "Where?"
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

- **Arrow plots:**
- also called hedgehog plots
- represent velocity as arrows at regular locations, e.g., place arrows at grid points
- ➔ overloading possible
- arrows: (scaled) unit length or encode magnitude
- well-established for 2D

- Arrows visualize
  - Direction of vector field
  - Orientation
  - Magnitude:
    - Length of arrows
    - Color coding



- [Kirby et al 99]: multiple values of 2d flow data by layering concept related to painting process of artists

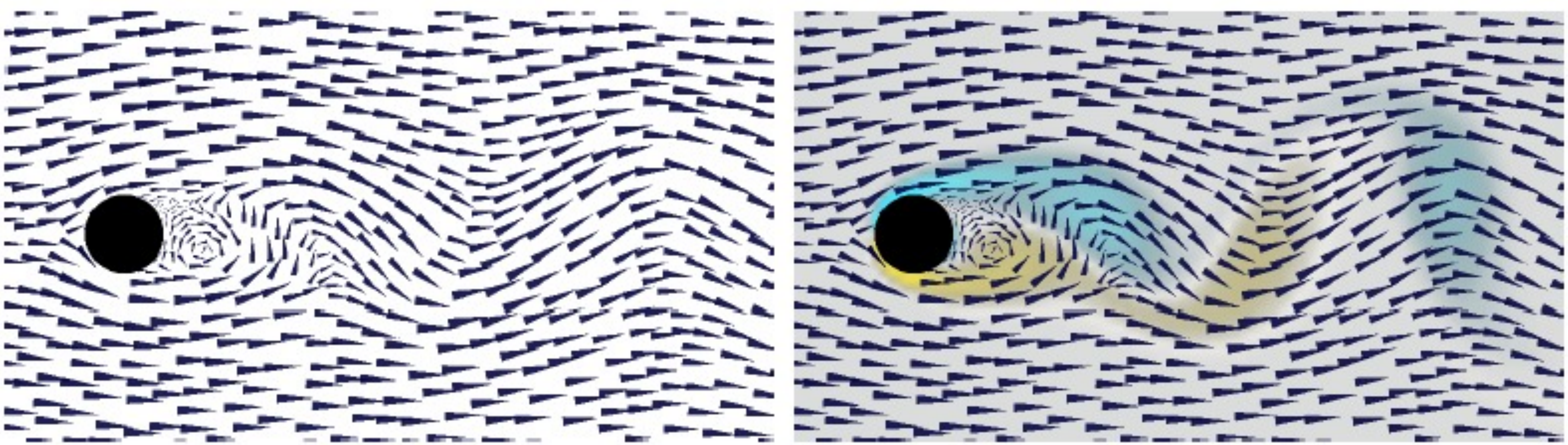
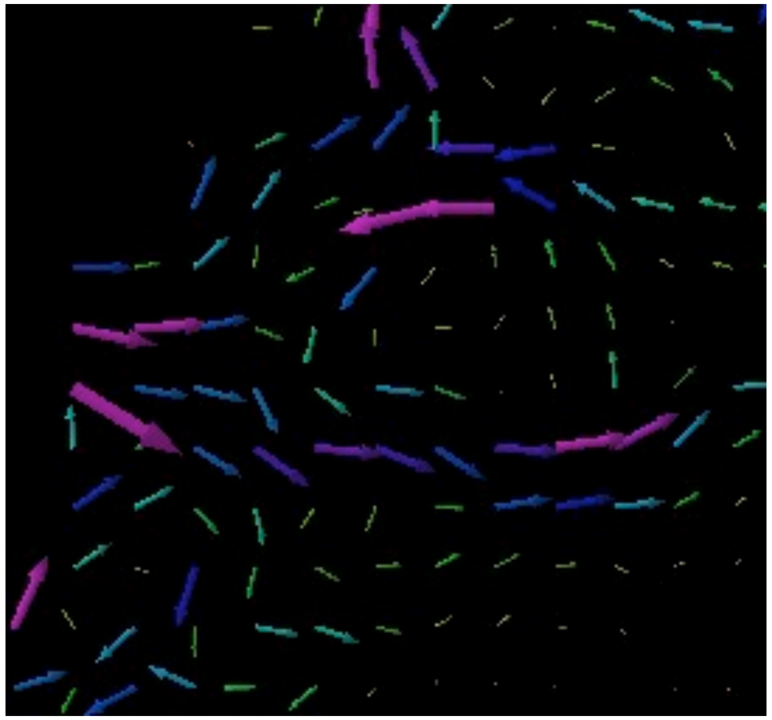
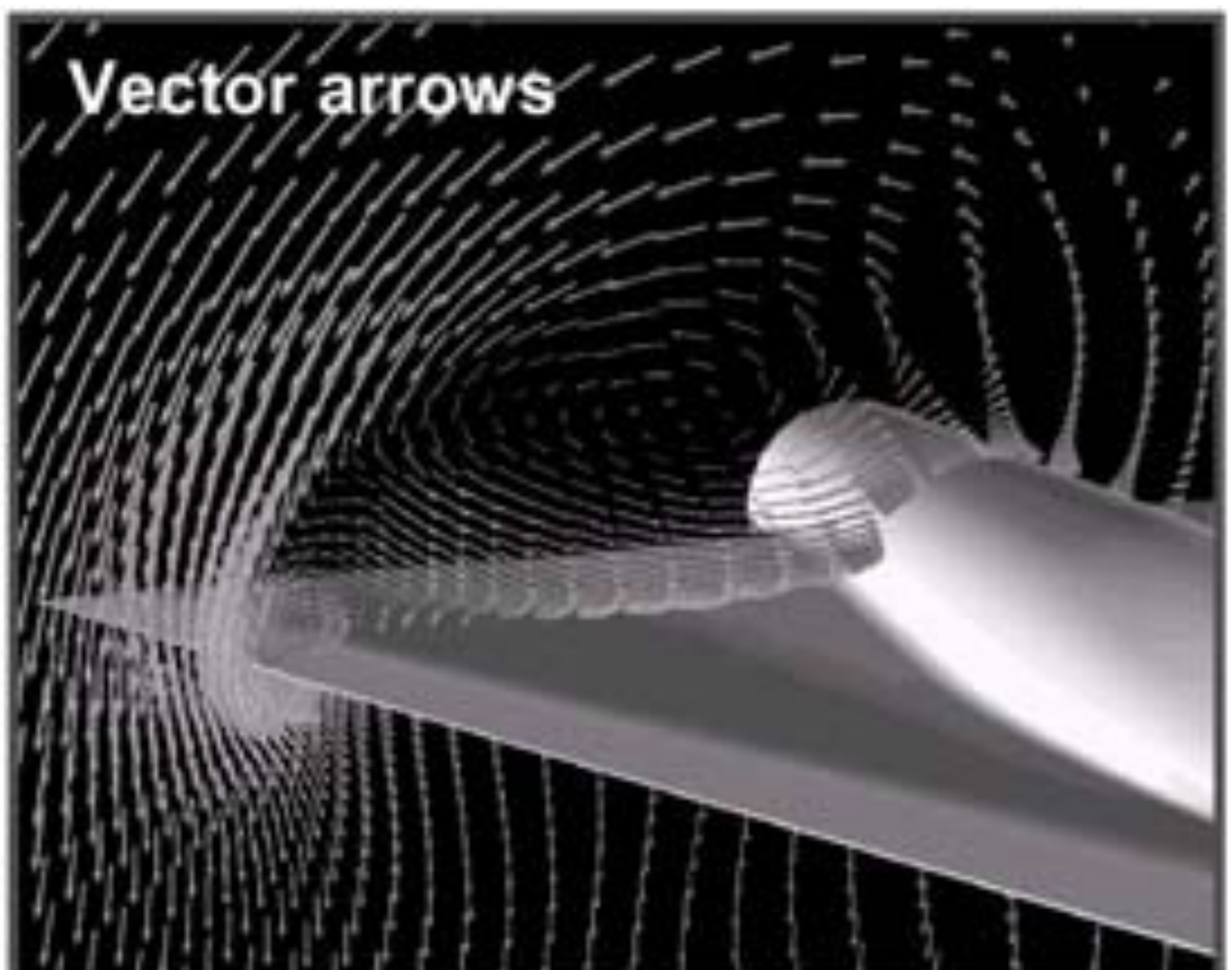
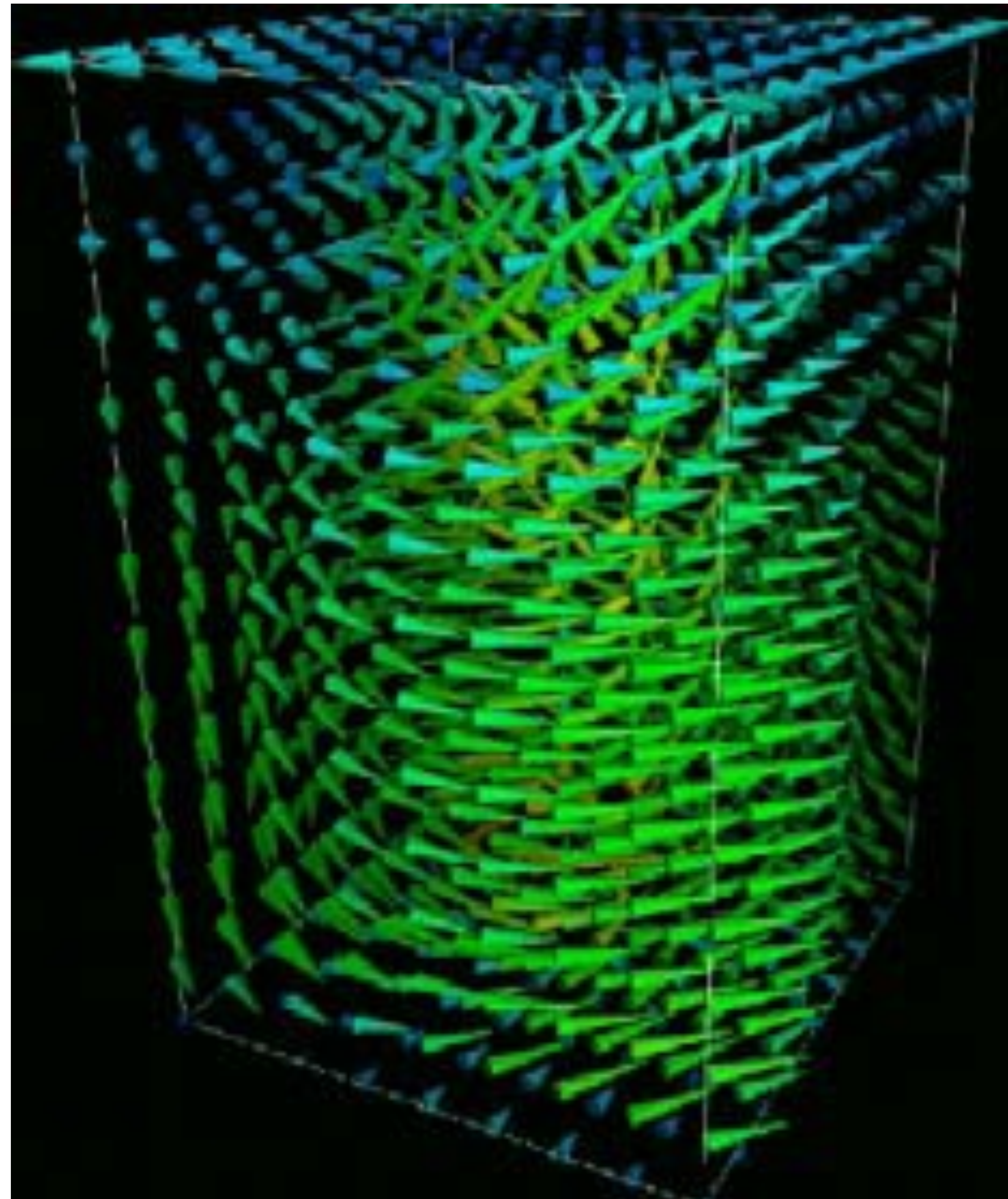


Figure 1: Typical visualization methods for 2D flow past a cylinder at Reynolds number 100. On the left, we show only the velocity field. On the right, we simultaneously show velocity and vorticity. Vorticity represents the rotational component of the flow. Clockwise vorticity is blue, counterclockwise yellow.

Arrows in 3D







- Advantages and disadvantages of glyphs and arrows:
  - + Simple
  - + 3D effects
  - - Inherent occlusion effects
  - - Poor results if magnitude of velocity changes rapidly
    (Use arrows of constant length and color code magnitude)

Approaches to flow vis

- “How?”
  - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
  - Texture-based (LIC, spot noise)
  - Direct + geometry-based (hedehogs, glyphs)
  - **Direct + heuristic (magnitude, Laplacian, FTLE)**
  - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- “Where?”
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

• Volume illustration for flow visualization [Svakine et al 05]

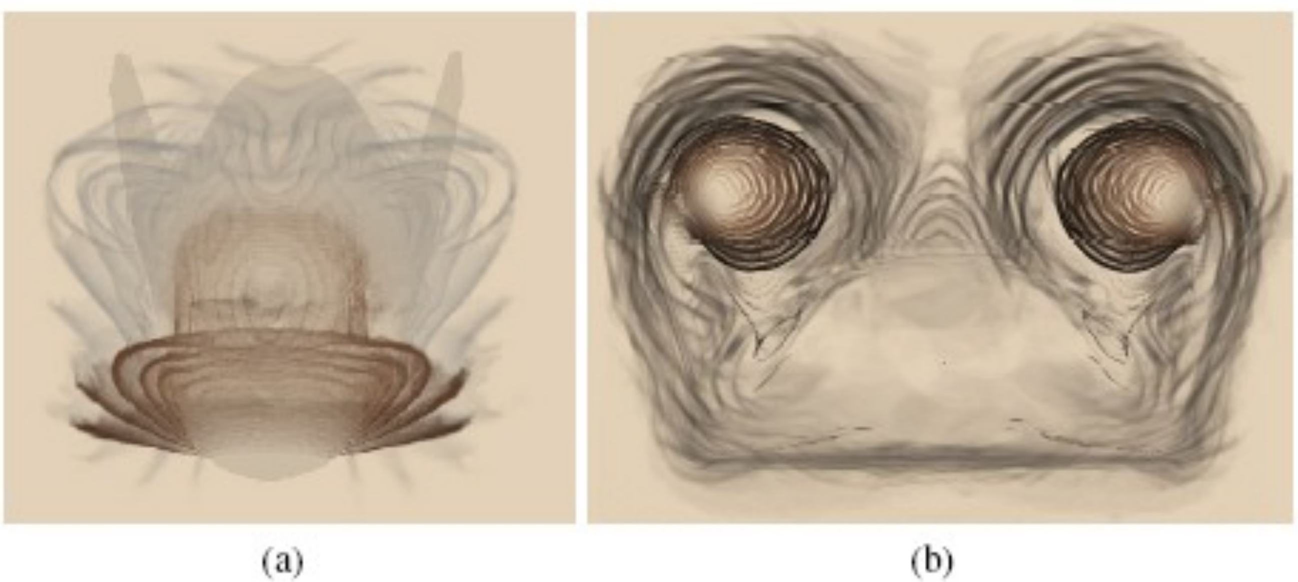


Figure 3: Volume illustrations of flow around the X38 spacecraft. (a) is an illustration of density flow and shock around the bow, while (b) highlights the vortices created above the fins of the spacecraft.

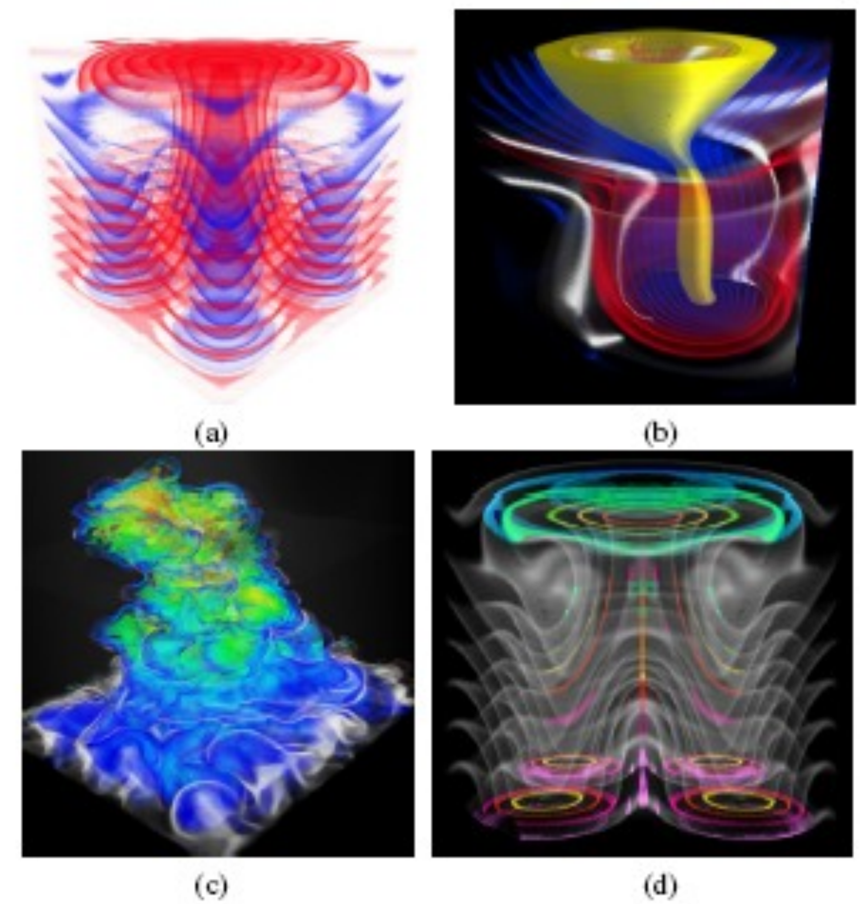
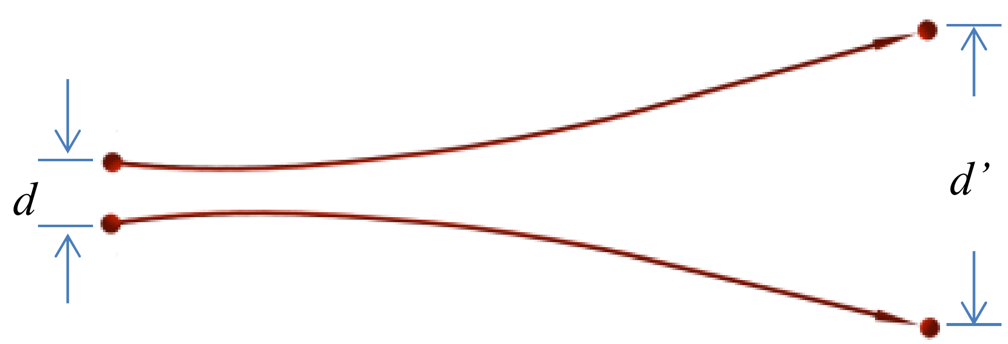
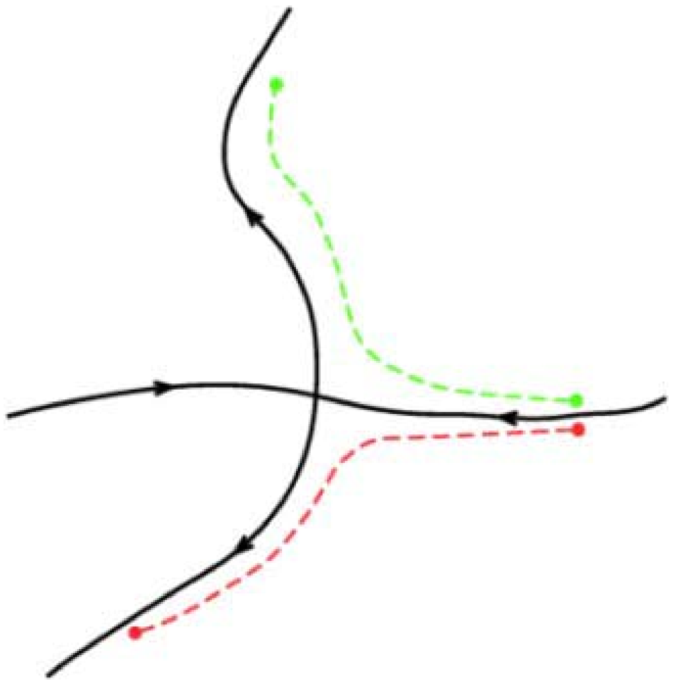


Figure 6: Use of two-dimensional transfer function with the Laplacian operator and other flow quantities. (a) shows heat inflow (red) and outflow (blue). (b) shows all values of the Laplacian of velocity magnitude in the tornado dataset. (c) visualizes the cloud TKE using the Laplacian to highlight boundaries (white) and velocity for silhouetting. (d) highlights emerging flow structures in the convection dataset using banding of the second derivative magnitude of the temperature field.

Finite-Time Lyapunov Exponent

- Some observation
  - Observe particle trajectories
  - Measure the **divergence** between trajectories, i.e. how much flow stretch





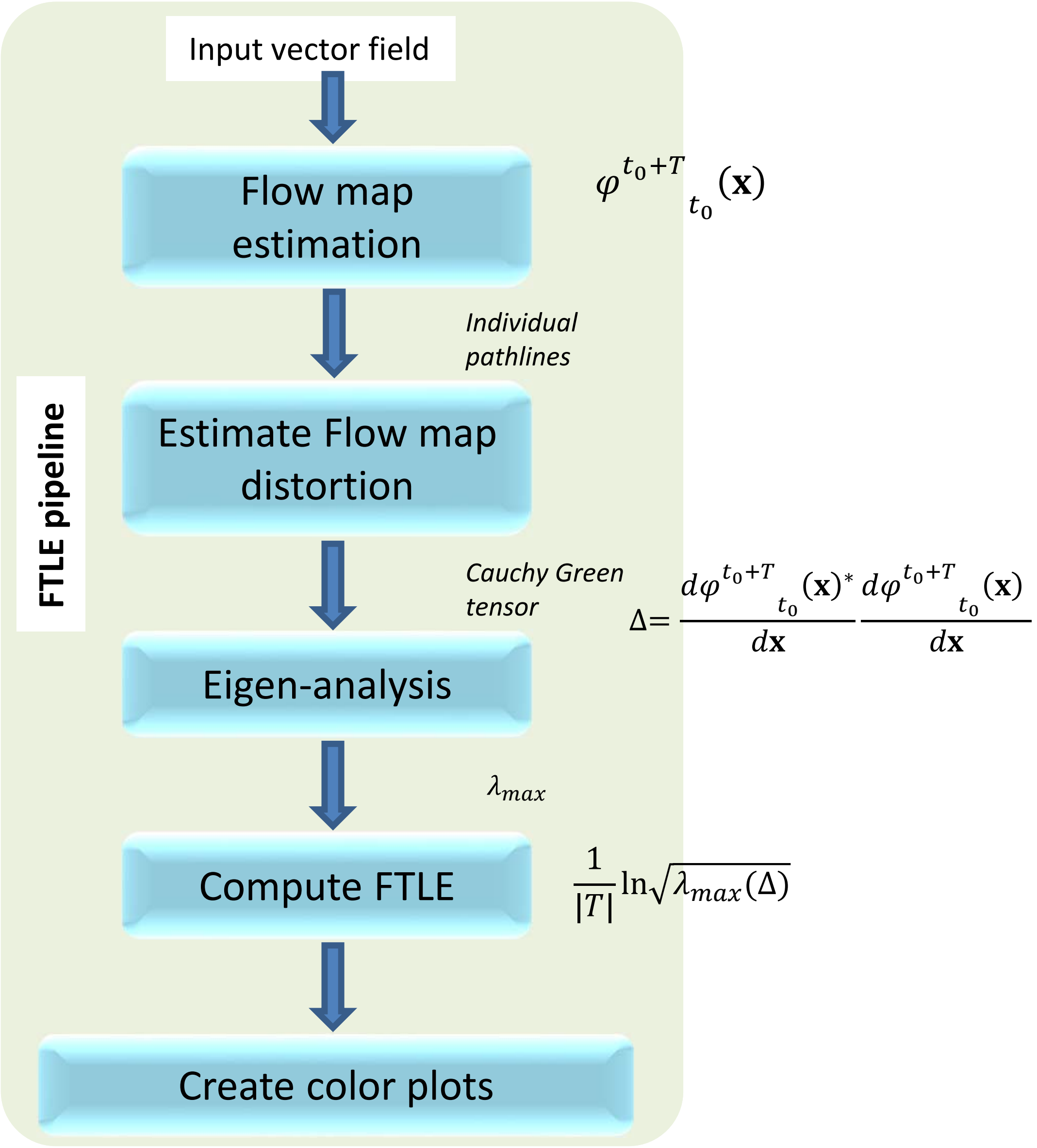
[Shadden]

Finite-Time Lyapunov Exponent

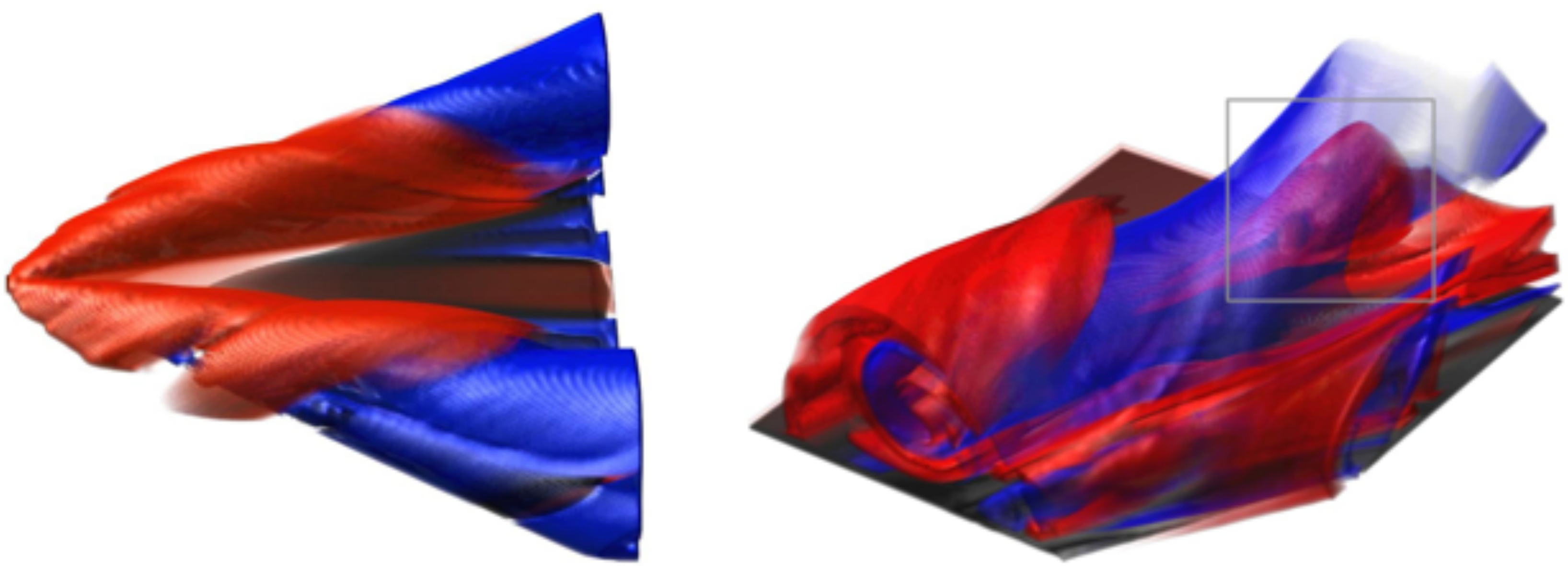
- Description
  - Lyapunov exponents describe rate of separation or stretching of two infinitesimally close points over time in a dynamical system
  - FTLE refers to the largest Lyapunov exponent for only a **limited time** and is measured **locally**
  - *Largest exponent is governing the behavior of the system, smaller ones can be neglected*
  - Ridge lines of FTLE correspond to **"Lagrangian Coherent Structures" (LCS)**
  - **i.e., sources and sinks**

Finite-Time Lyapunov Exponent

A computation framework

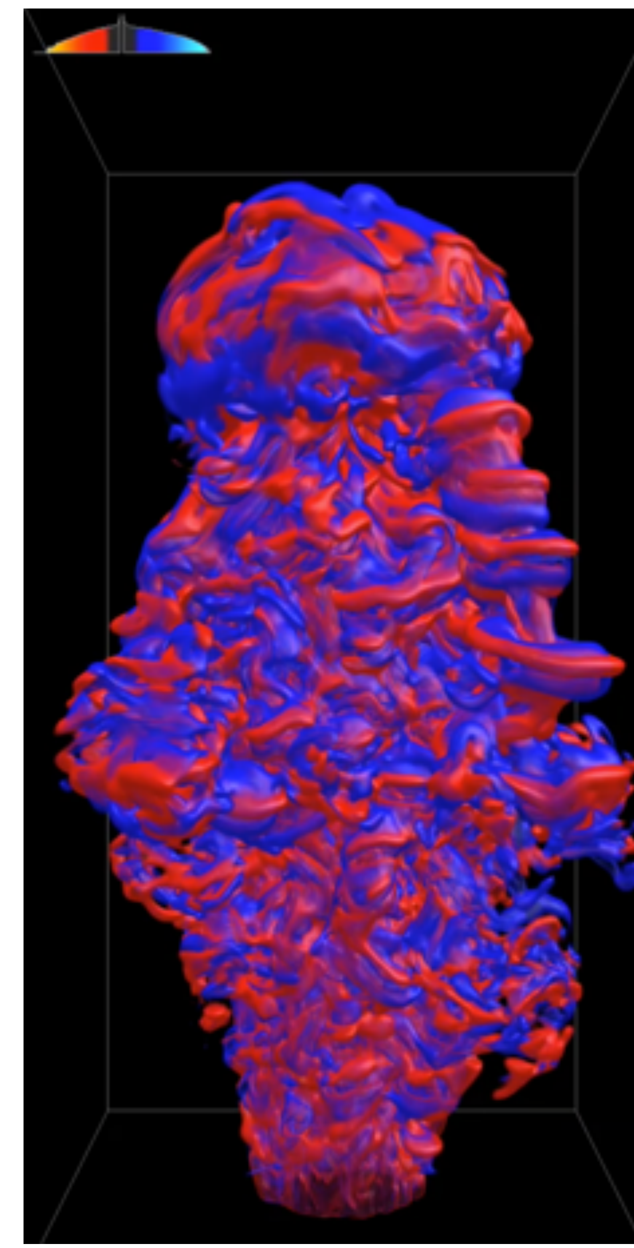


FTLE volumes - sources and sinks



Efficient Computation and Visualization of Coherent Structures in Fluid Flow Applications

C Garth, F Gerhardt, X Tricoche, H Hagen. IEEE Visualization 2007.



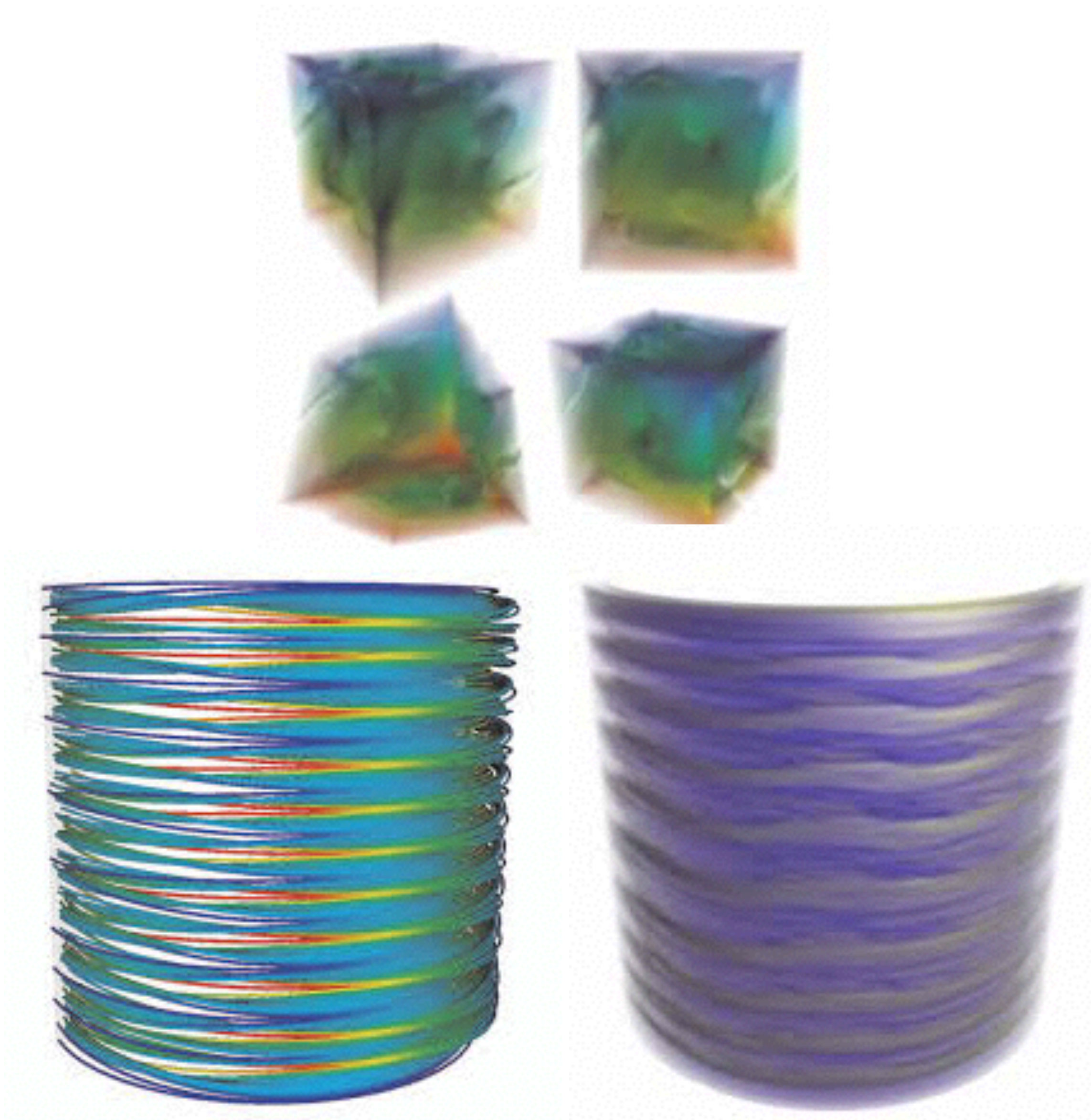
http://www.vacet.org/gallery/images_video/jet4-ftle-0.012.mp4

www.sci.utah.edu

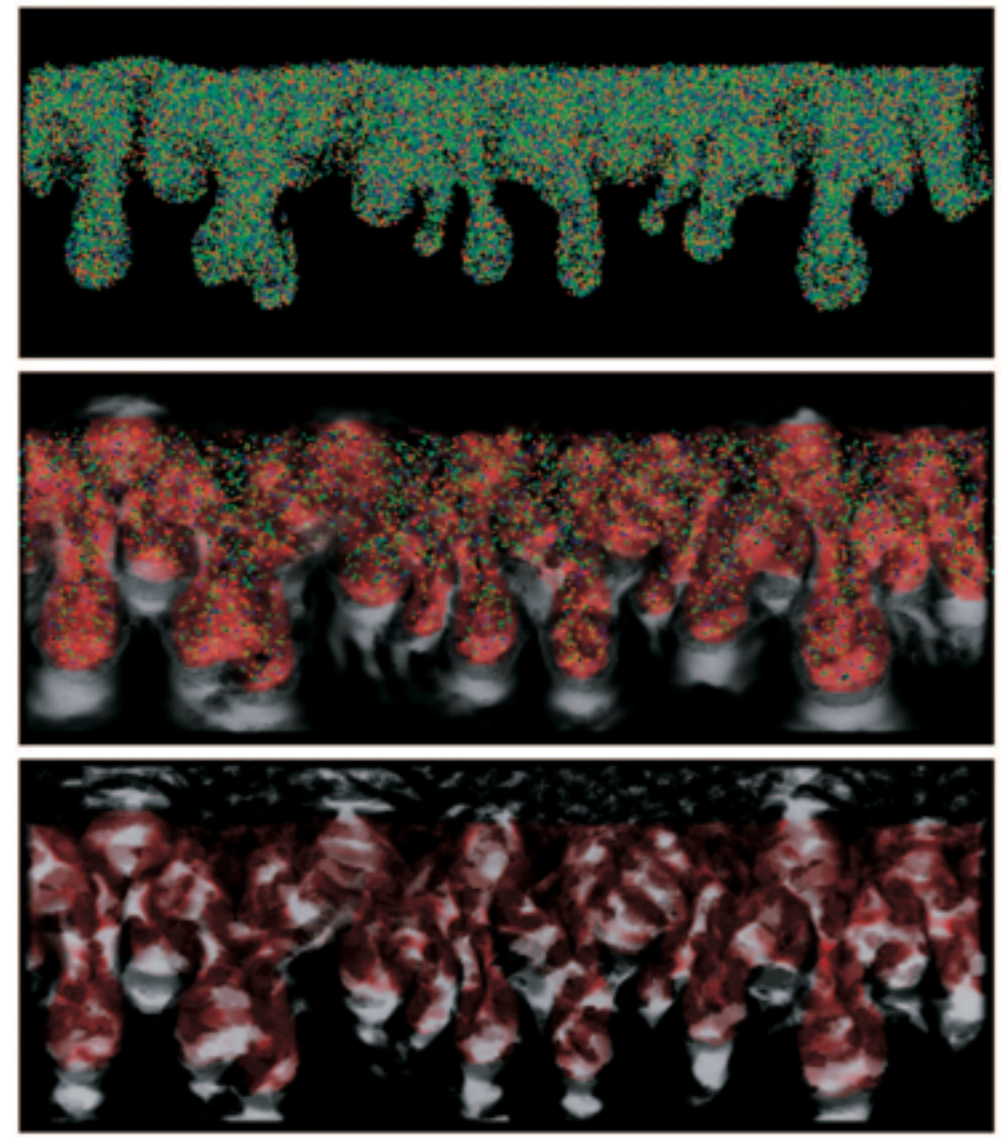
Approaches to flow vis

- "How?"
  - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
  - Texture-based (LIC, spot noise)
  - Direct geometry-based (hedehogs, glyphs)
  - Direct heuristic (magnitude, Laplacian, FTLE)
  - **Physically-based (Schlieren imaging, virtual rheoscopic fluids)**
- "Where?"
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

Virtual Rheoscopic Fluids



Barth et al. Virtual Rheoscopic Fluids for Flow Visualization, IEEE Vis 2007

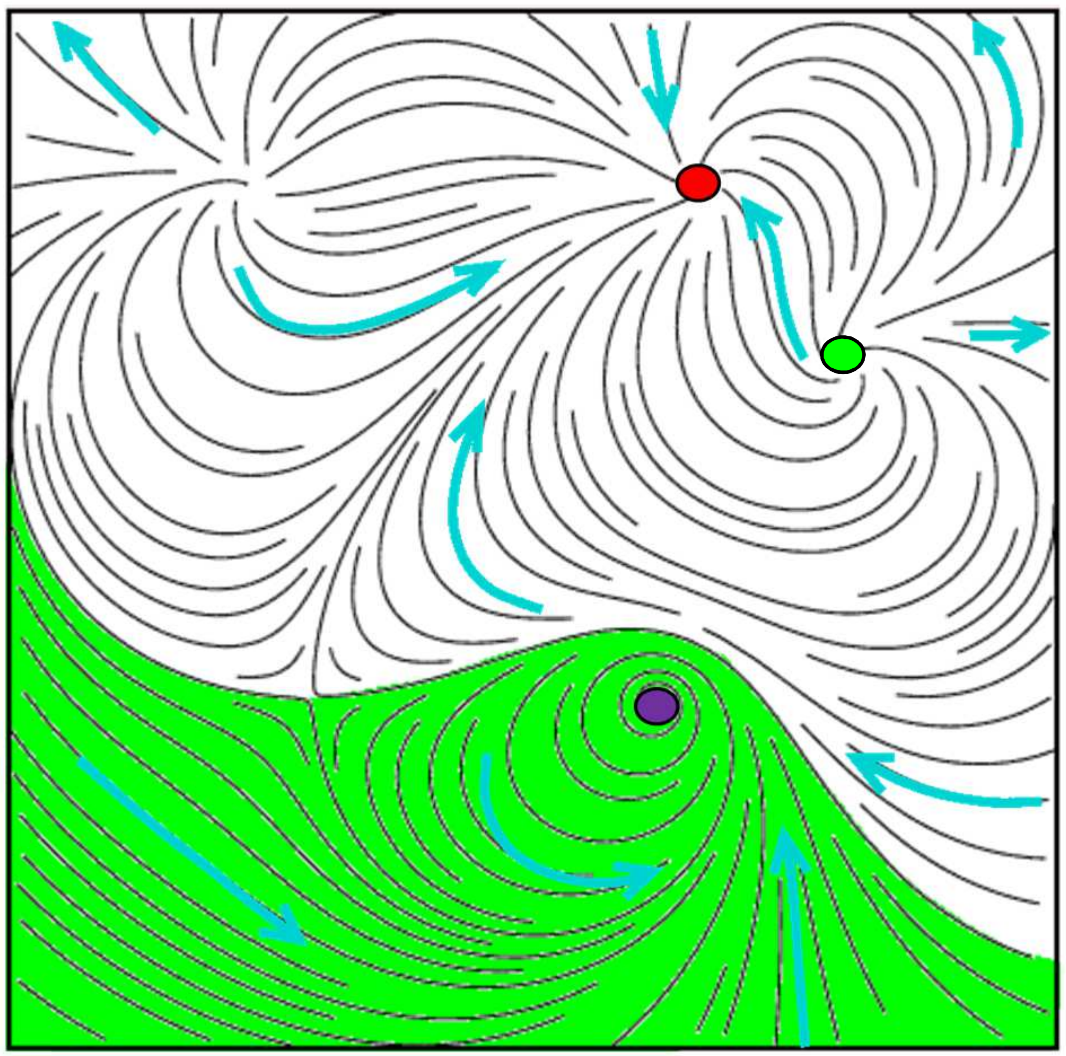


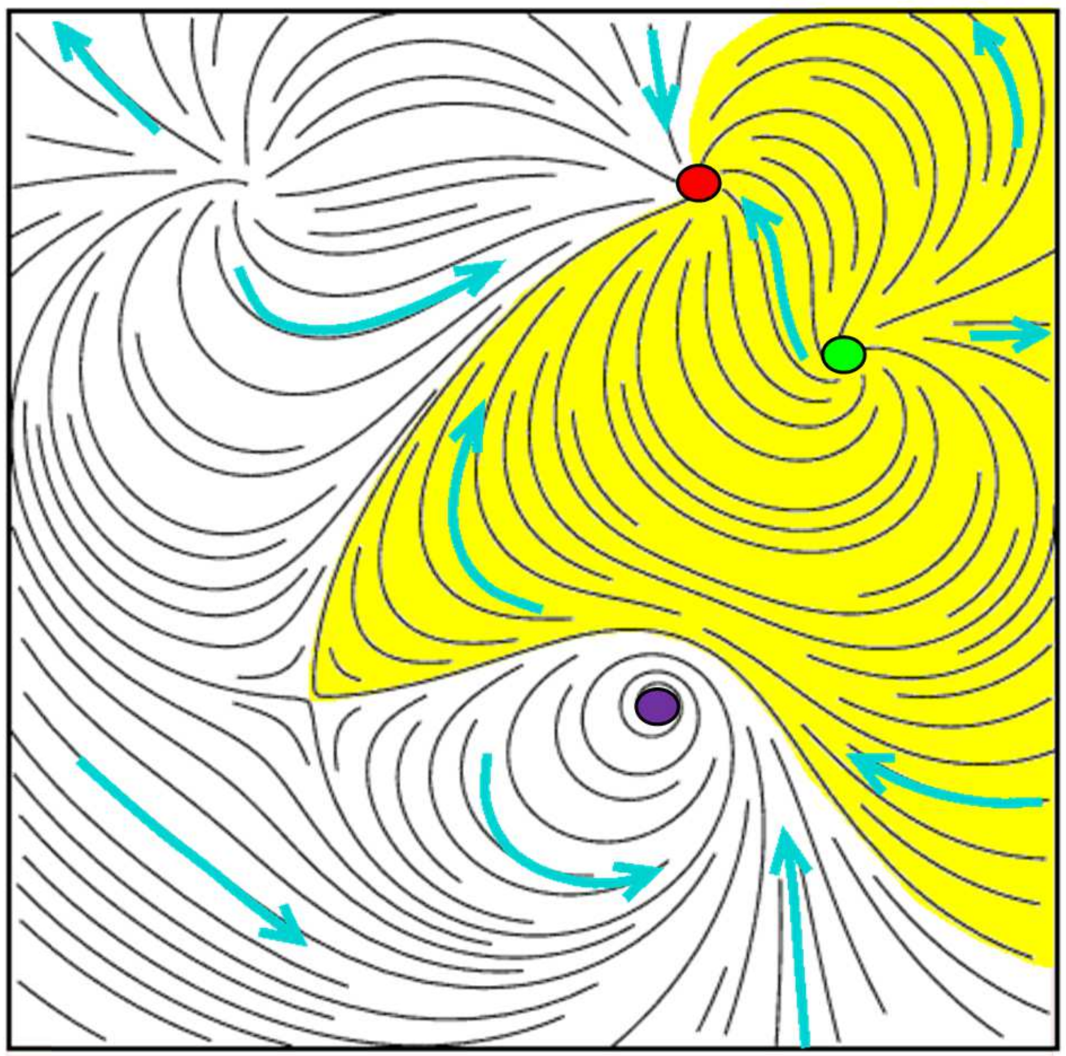
Hecht et al. Virtual Rheoscopic Fluids, IEEE Vis 2008

- Simulates the orientation of virtual microscope gold plate particles swimming in the vector field.
- Determine rheoscopic particle orientation via eigenvalues of the Jacobian (gradient tensor)

Vector field topology preview

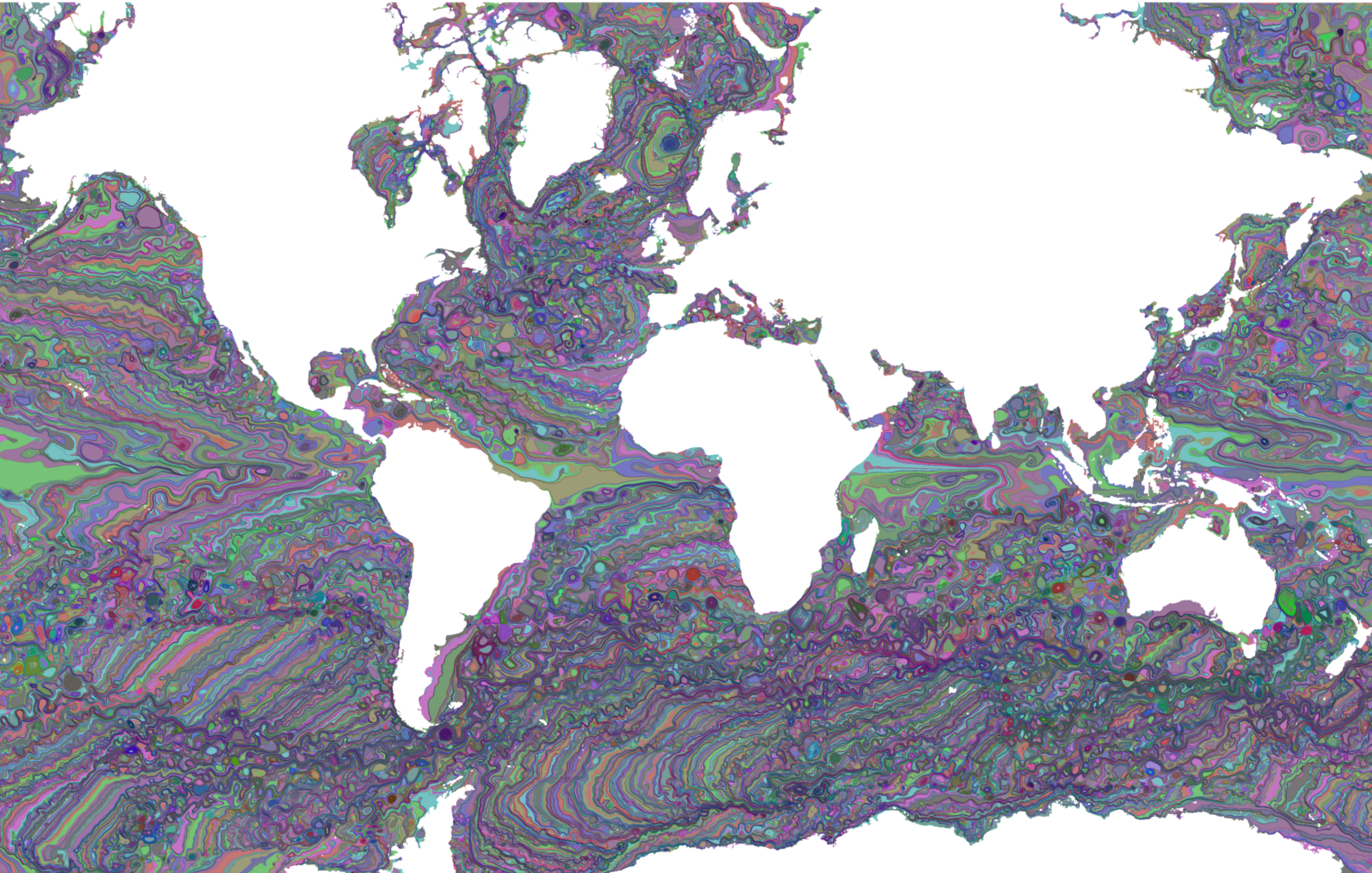
Repellor and Attractor Manifolds





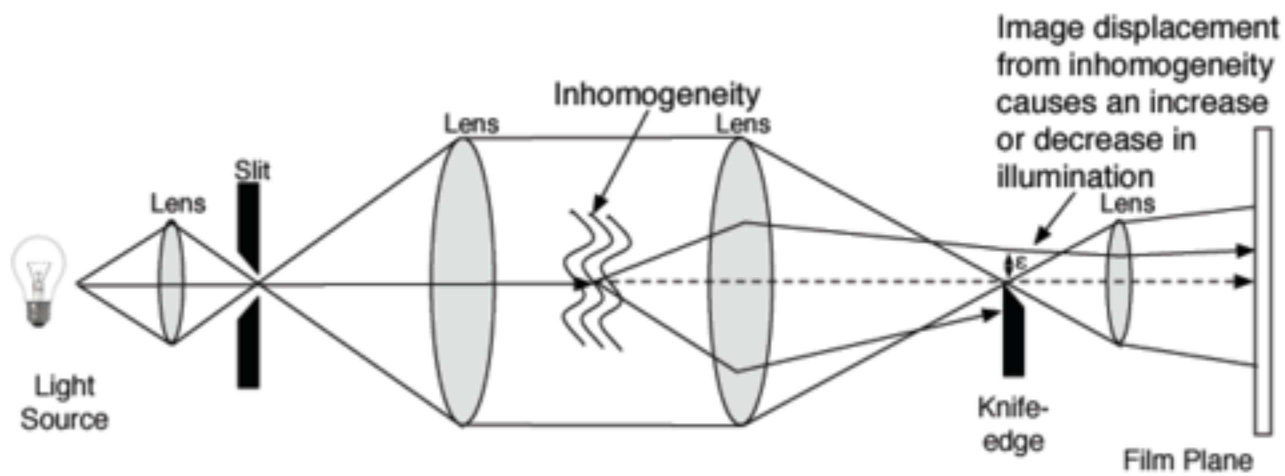
Vector field topology preview Levine, Jadhav, Bhatia, Pascucci, Bremer, CGF 31(3) 2012

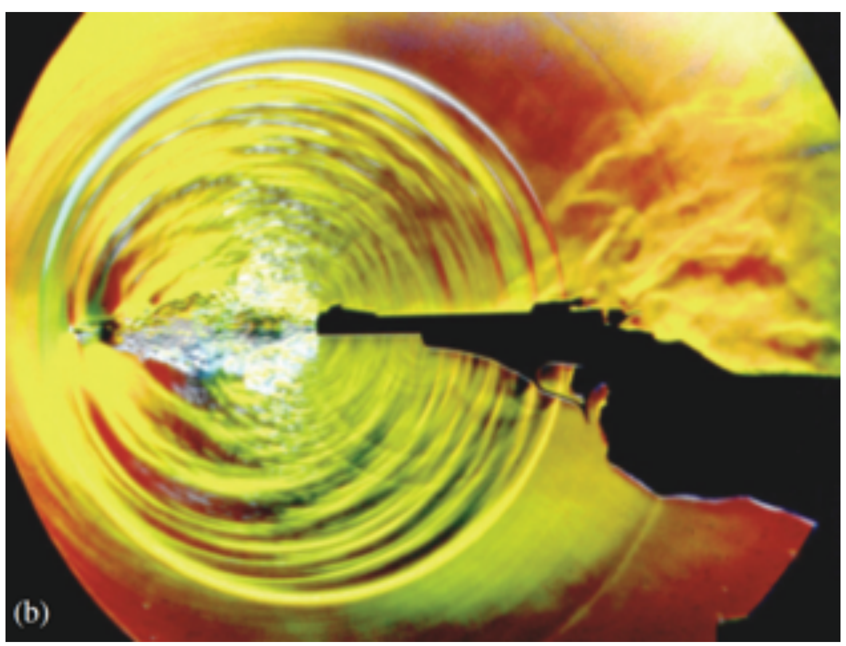
Stable Manifolds of Ocean Data

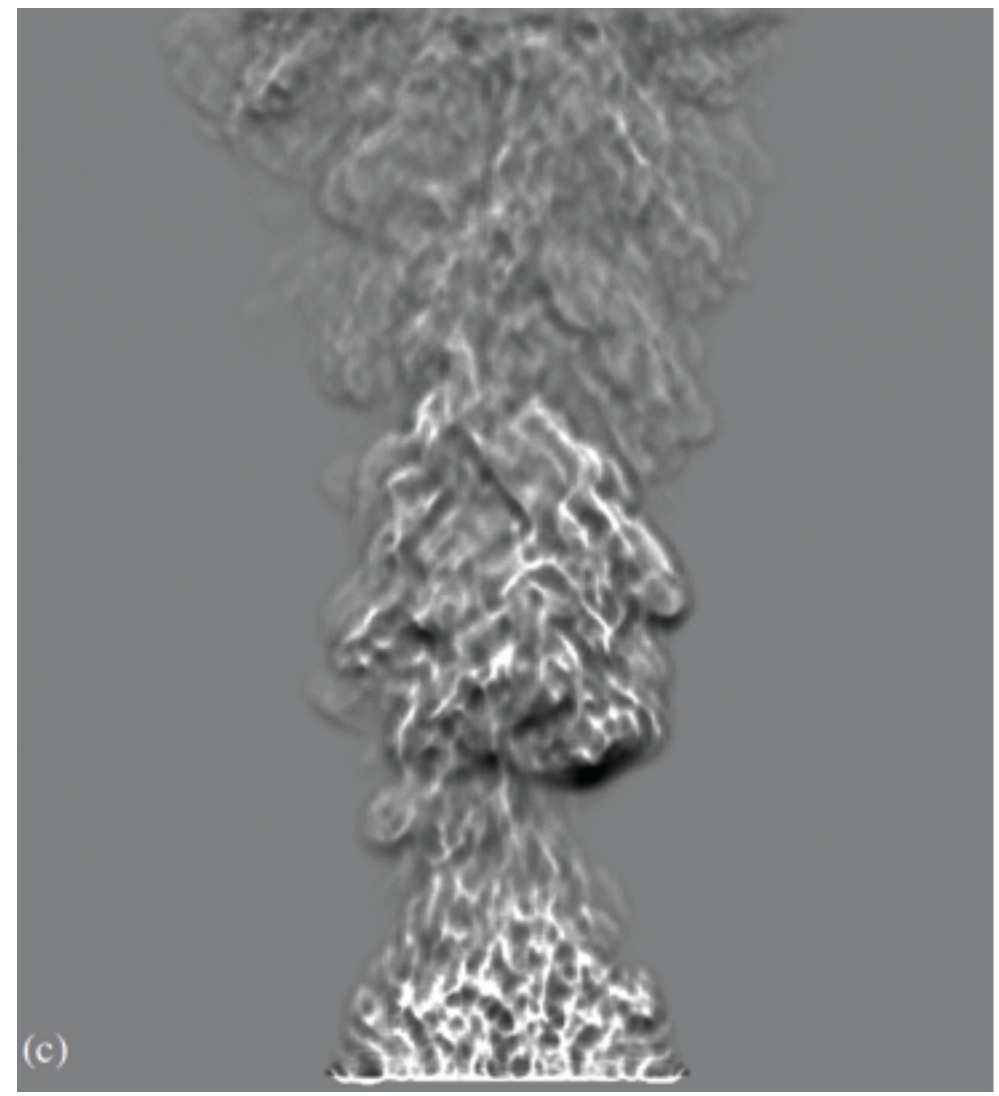


Non-vector field flow visualization

Schlieren imaging



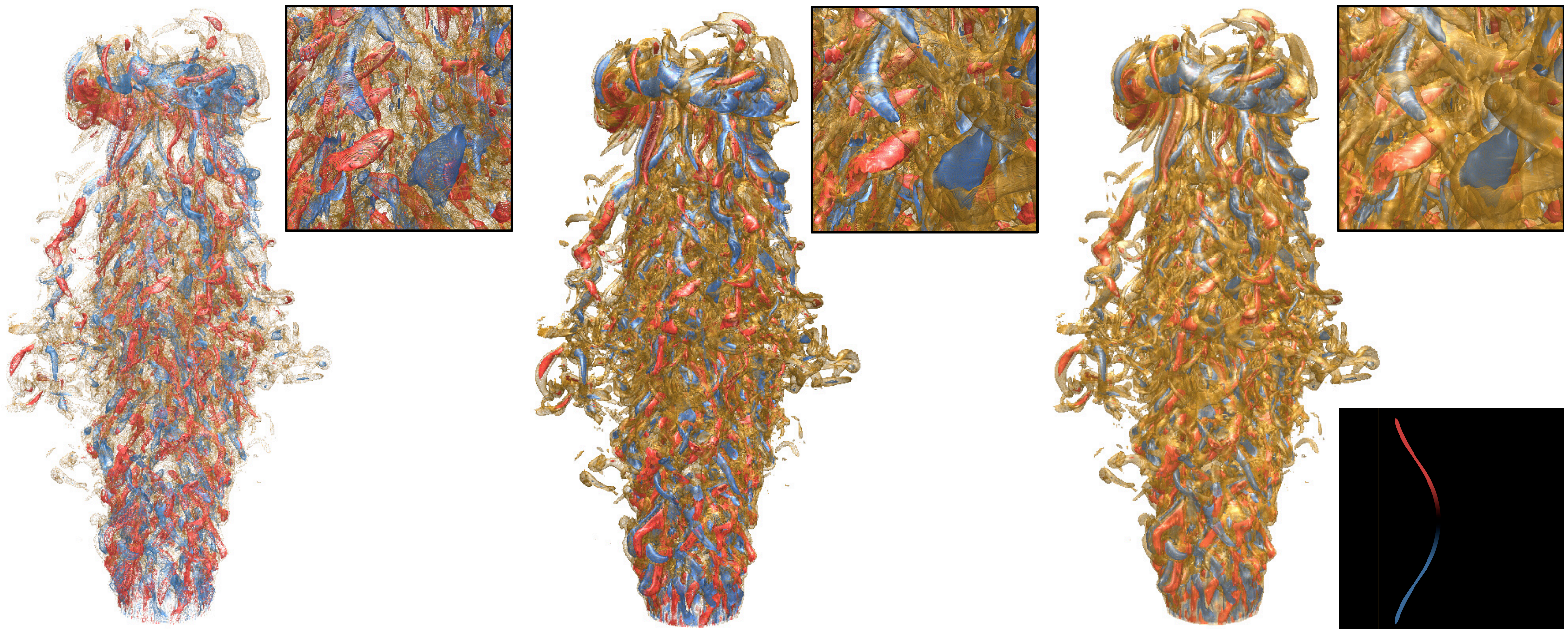




- Not *really* vector field visualization… but can show similar effects
- Uses precomputed index of refraction, and physically-based light transport (path tracing) to illustrate flow
- Brownlee et al. Physically-Based Interactive Schlieren Flow Visualization. IEEE Pacific Visualization 2010.

www.sci.utah.edu

Using multi-field volume rendering to visualize FTLE-like flow



(a) postclassification $\Delta t = 2$, 5fps (b) postclassification $\Delta t = 0.5$, 0.36fps (c) peak finding (ray march, chord) $\Delta t = 2$, 4.3fps

Classifying vorticity and normalized helicity, compare with FTLE computation.

Kotava et al. Volume Rendering with Multidimensional Peak Finding. IEEE Pacific Vis 2012

www.sci.utah.edu