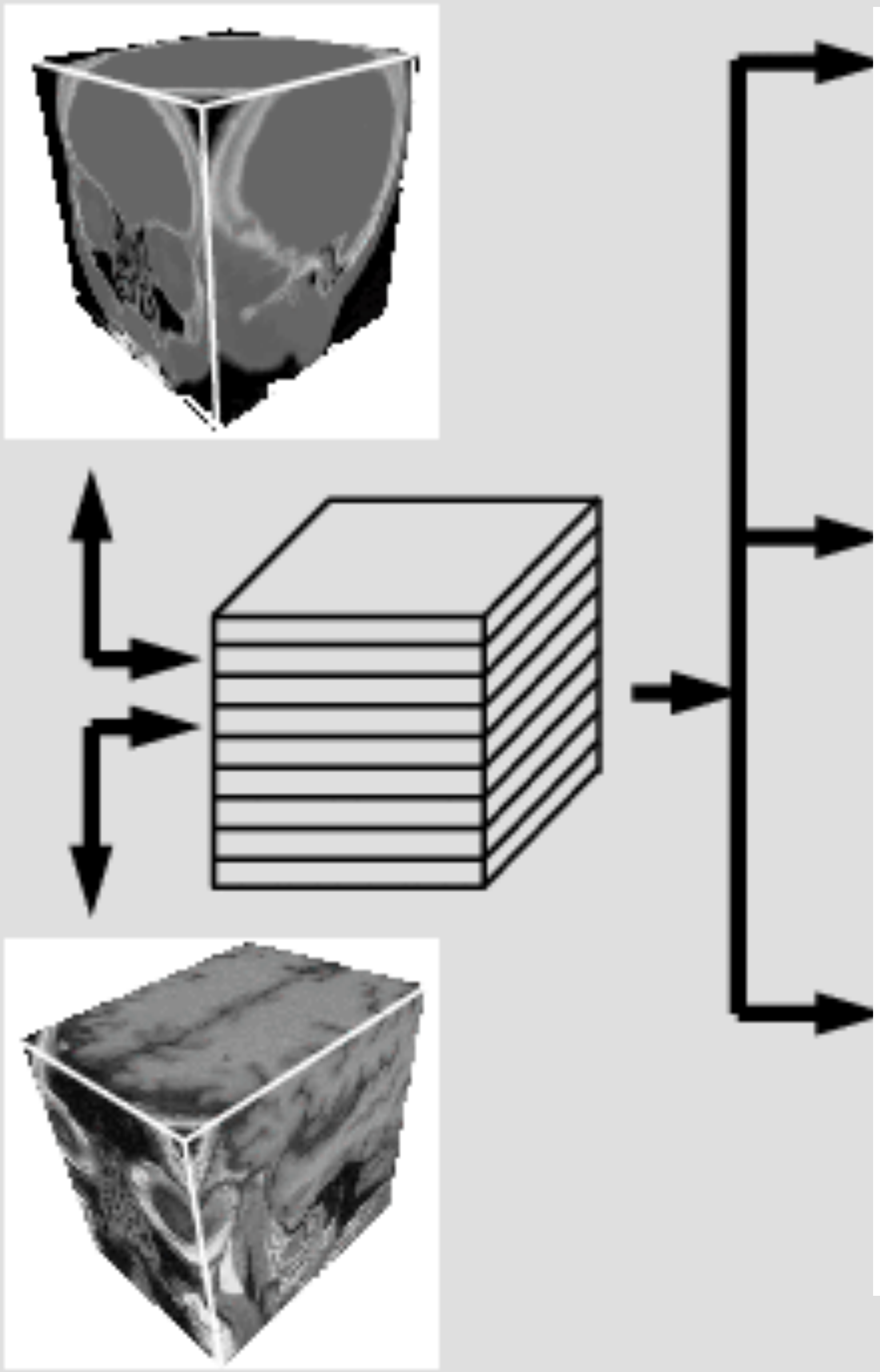
# Data Visualization (CSCI 627/490)

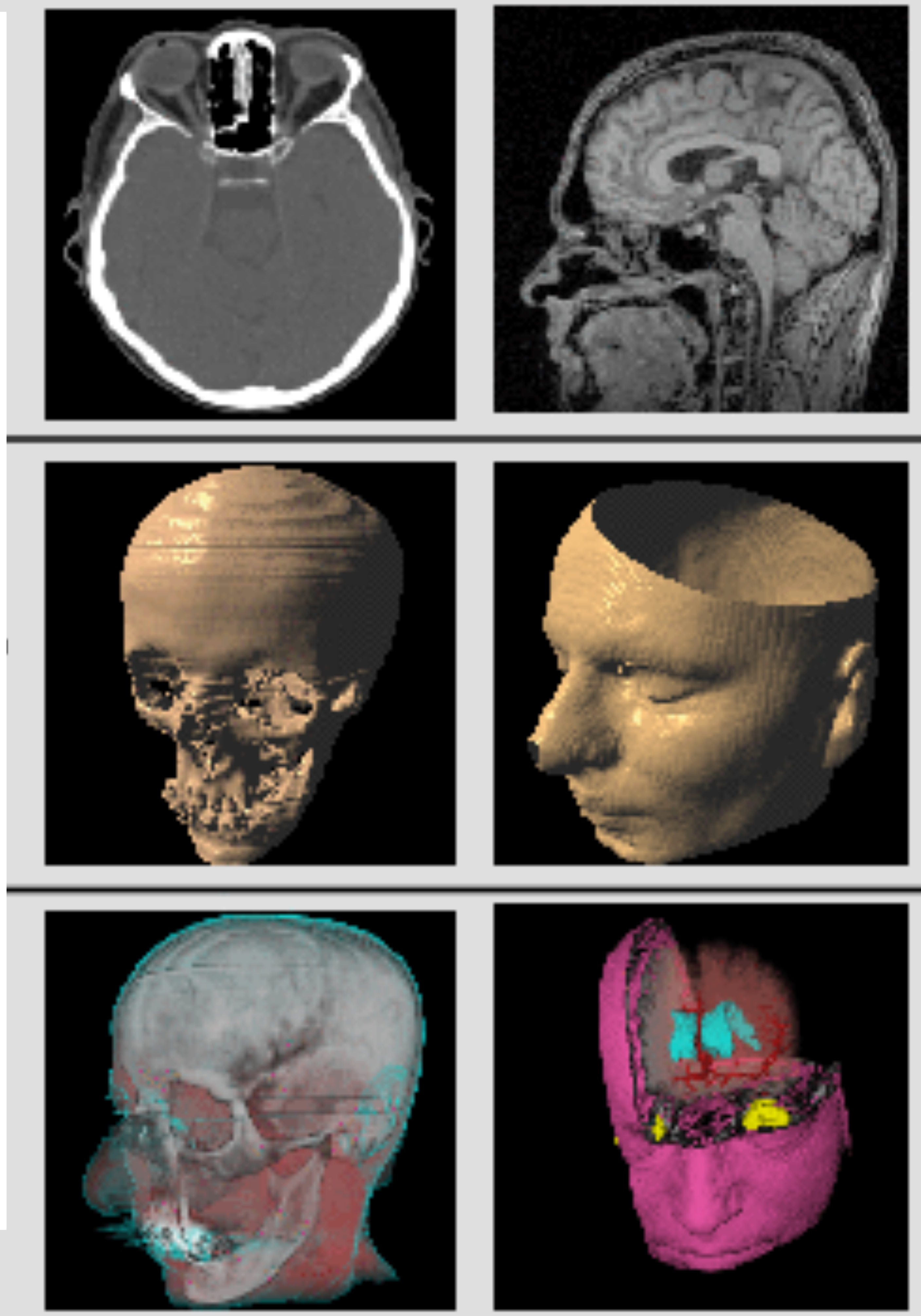
## Vector Field Visualization

Dr. David Koop

# Visualizing Volume (3D) Data

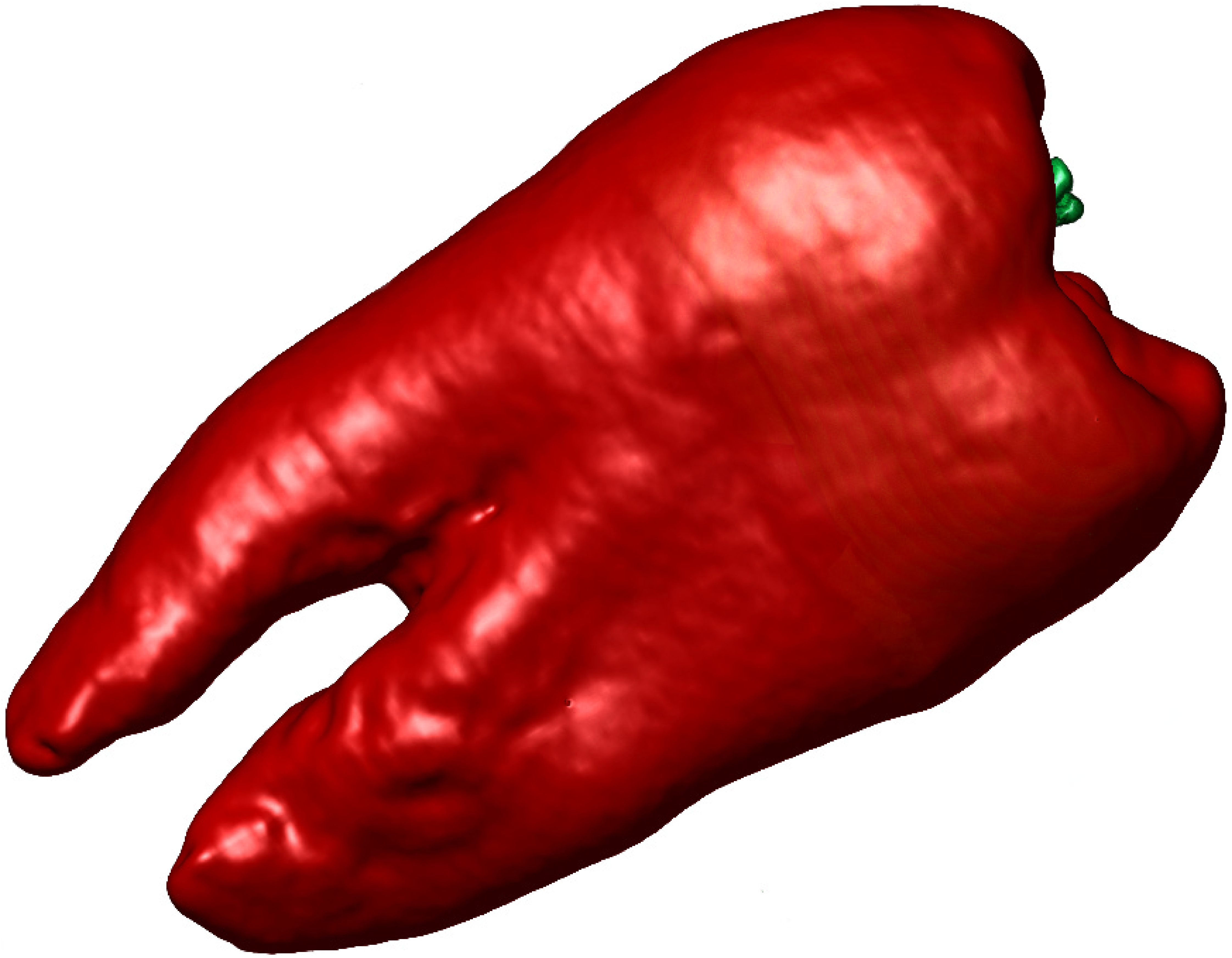


- 2D visualization slice images (or multi-planar reformating MPR)
- *Indirect* 3D visualization isosurfaces (or surface-shaded display SSD)
- *Direct* 3D visualization (direct volume rendering DVR)

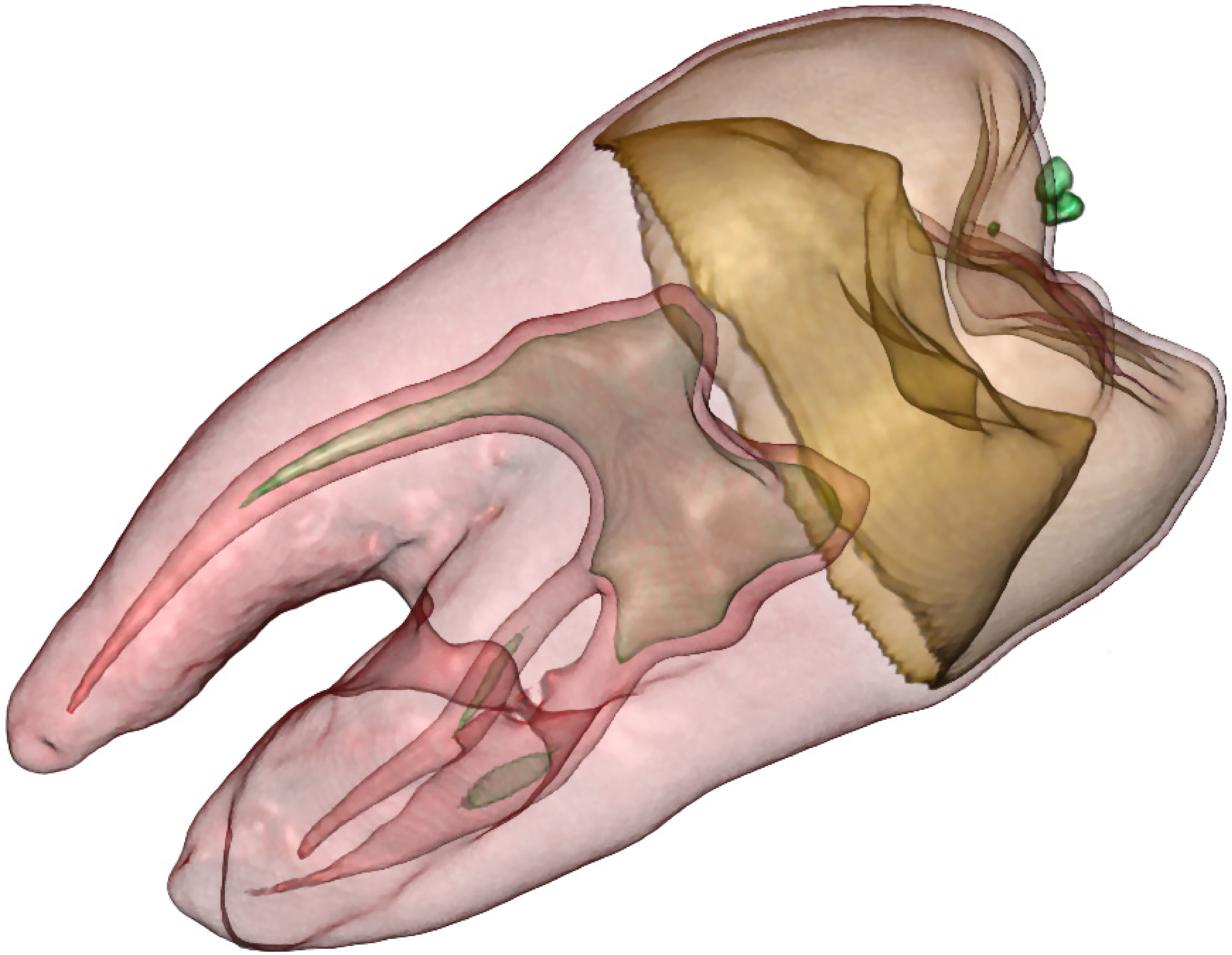


[© Weiskopf/Machiraju/Möller]

# Isosurfacing



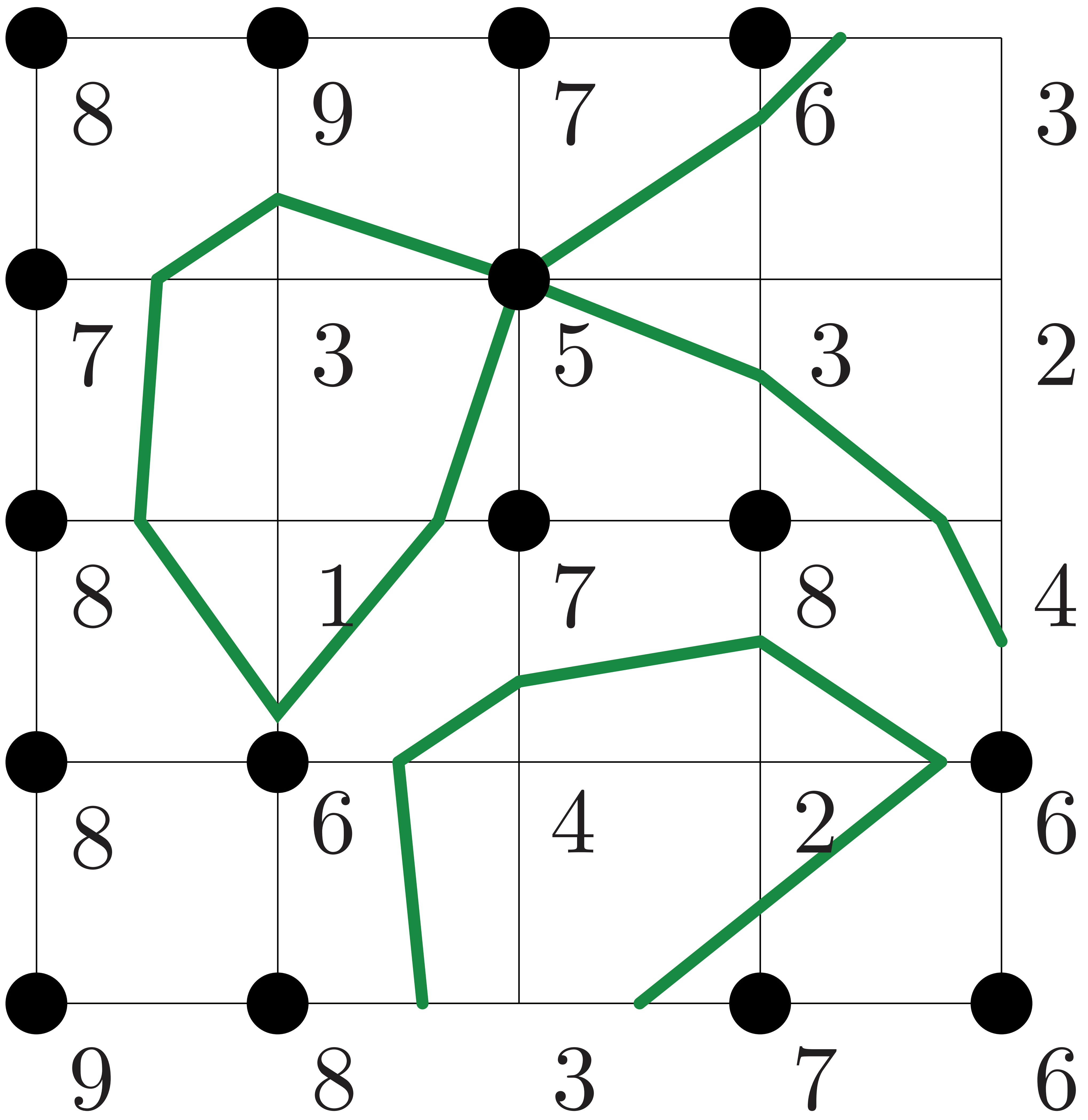
(a) An isosurfaced tooth.



(b) Multiple isosurfaces.

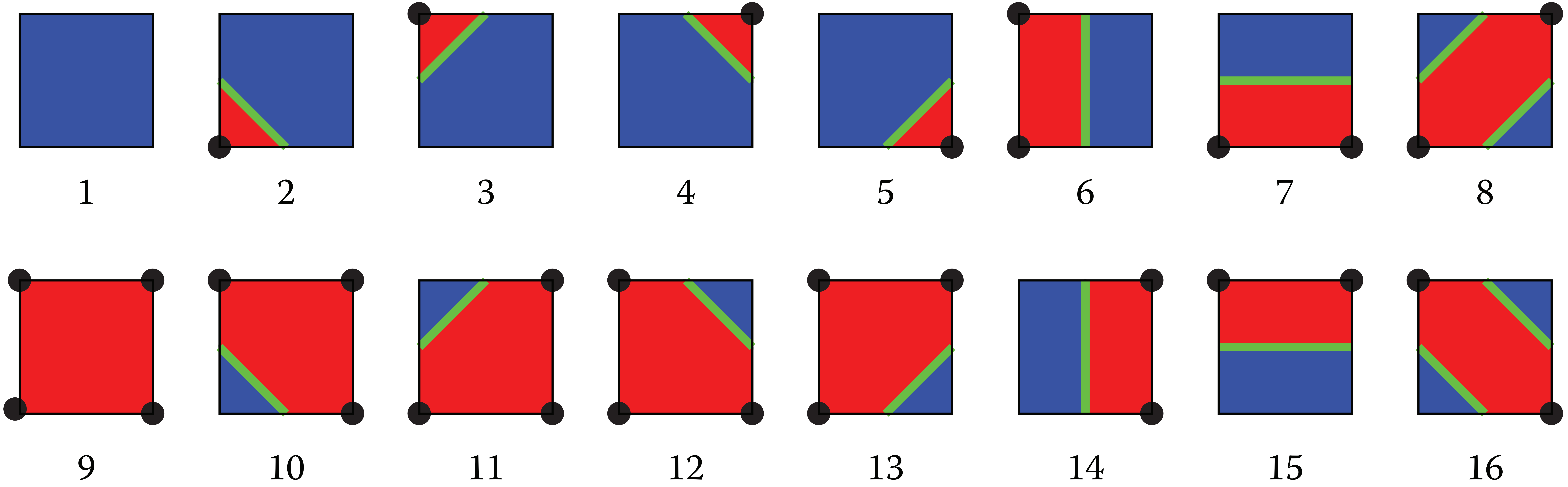
[J. Kniss, 2002]

# Generating Isolines




[R. Wenger, 2013]

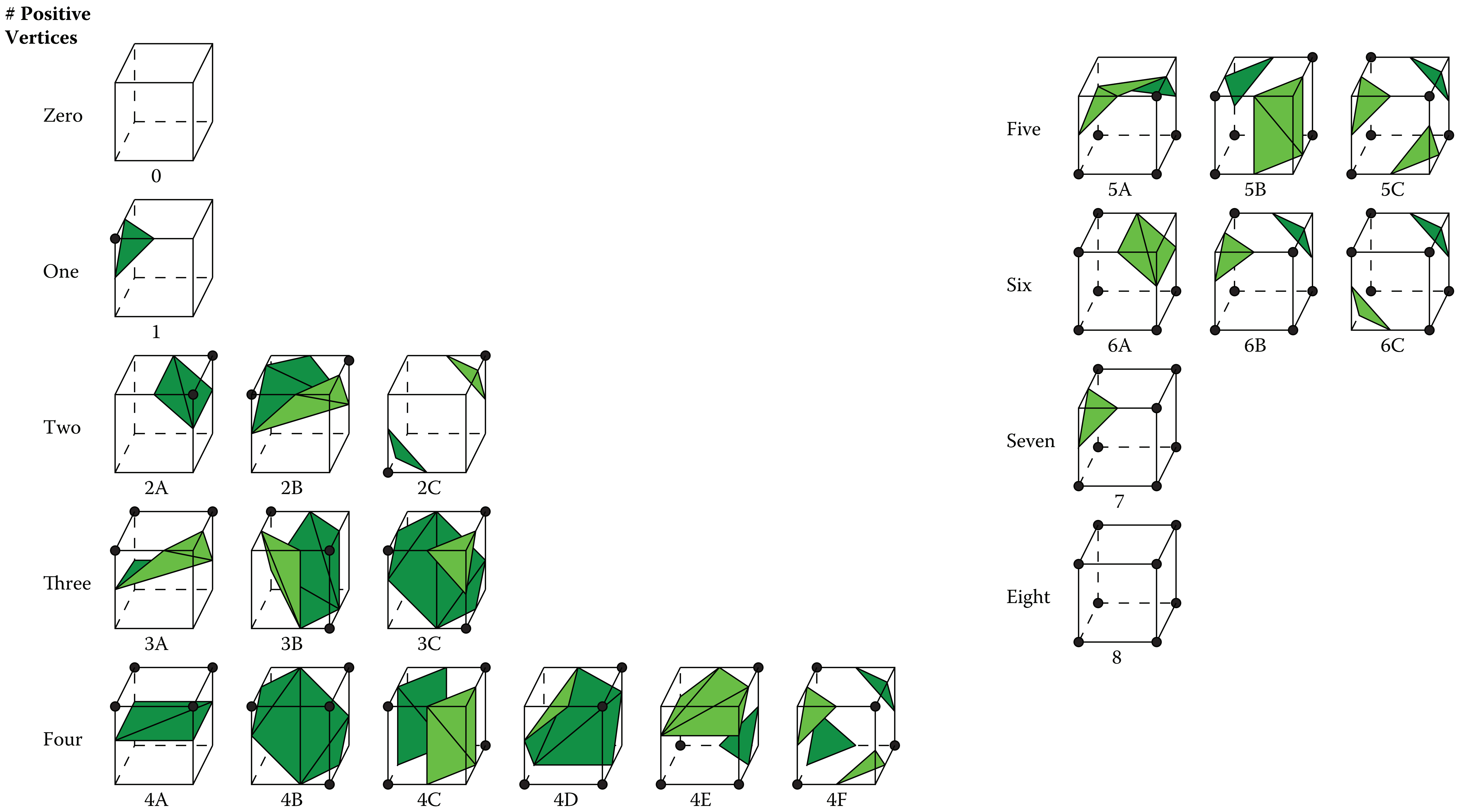
# Marching Squares




[R. Wenger, 2013]

# 3D: Marching Cubes

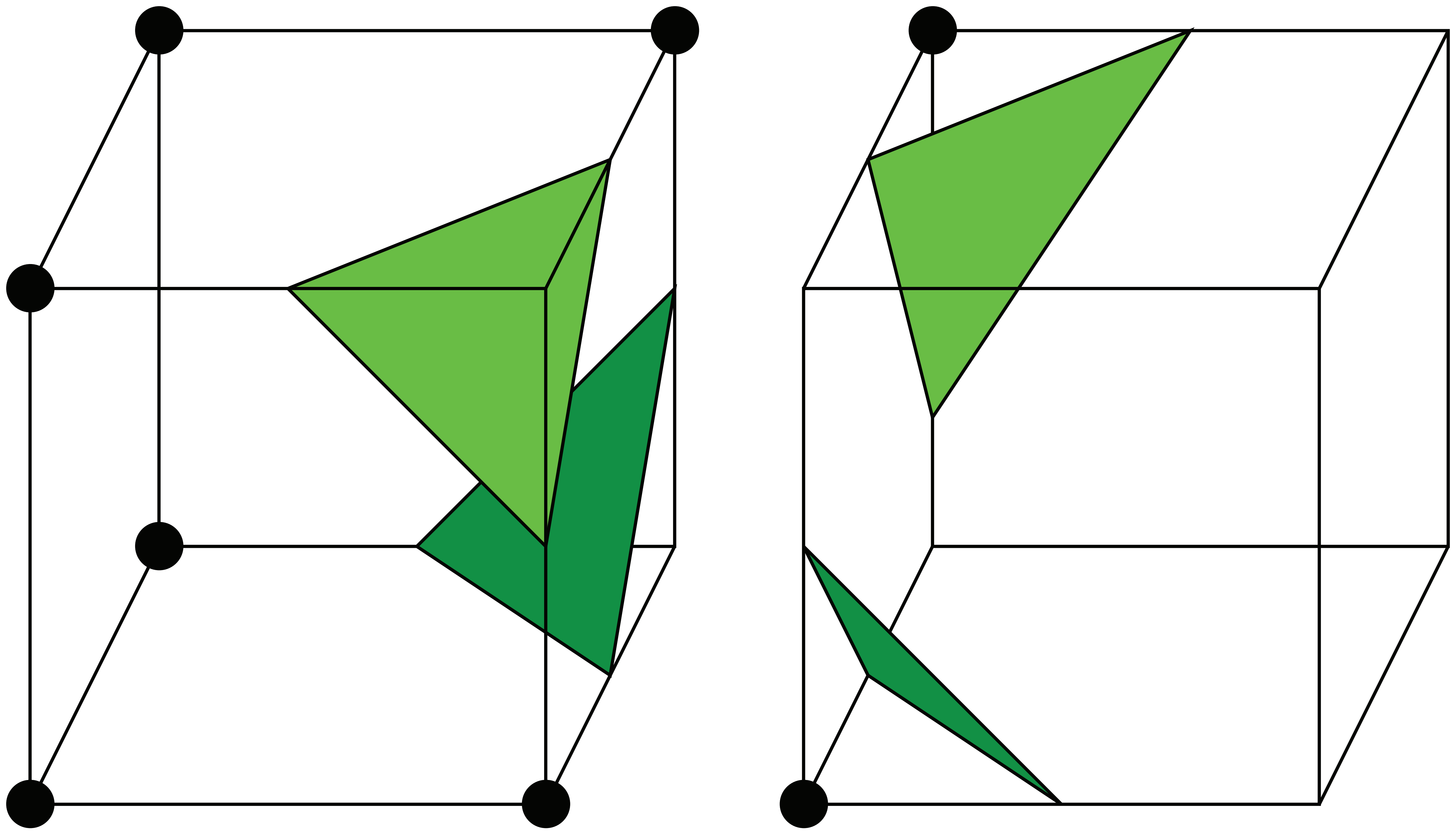
- Same idea, more cases [Lorensen and Cline, 1987]




[R. Wenger, 2013]

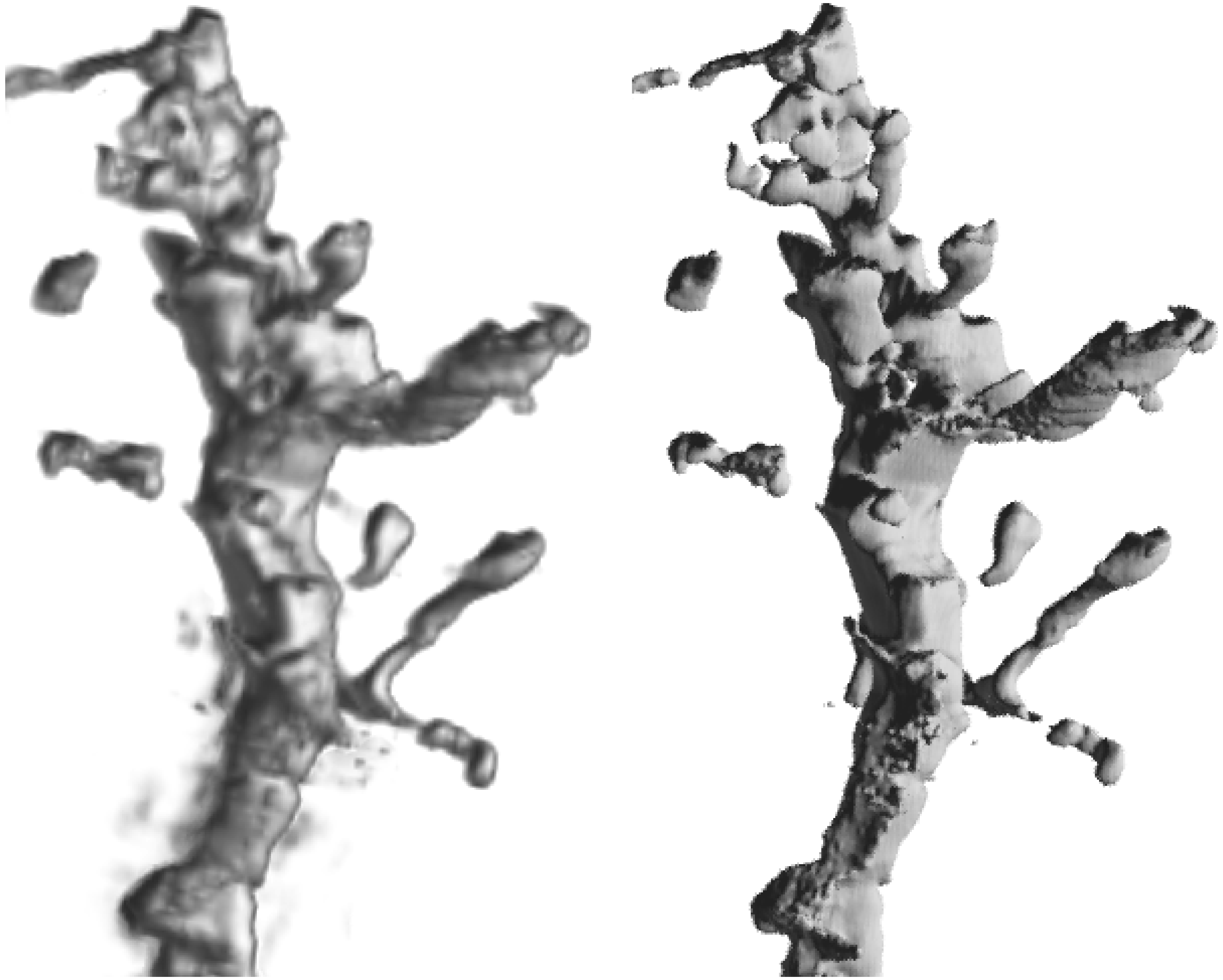
# Incompatible Choices

- If we have ambiguous cases where we choose differently for each cell, the surfaces will not match up correctly—there are holes
- Fix with the **asymptotic decider** [Nielson and Hamann,1991]



[R. Wenger, 2013]

# Volume Rendering vs. Isosurfacing

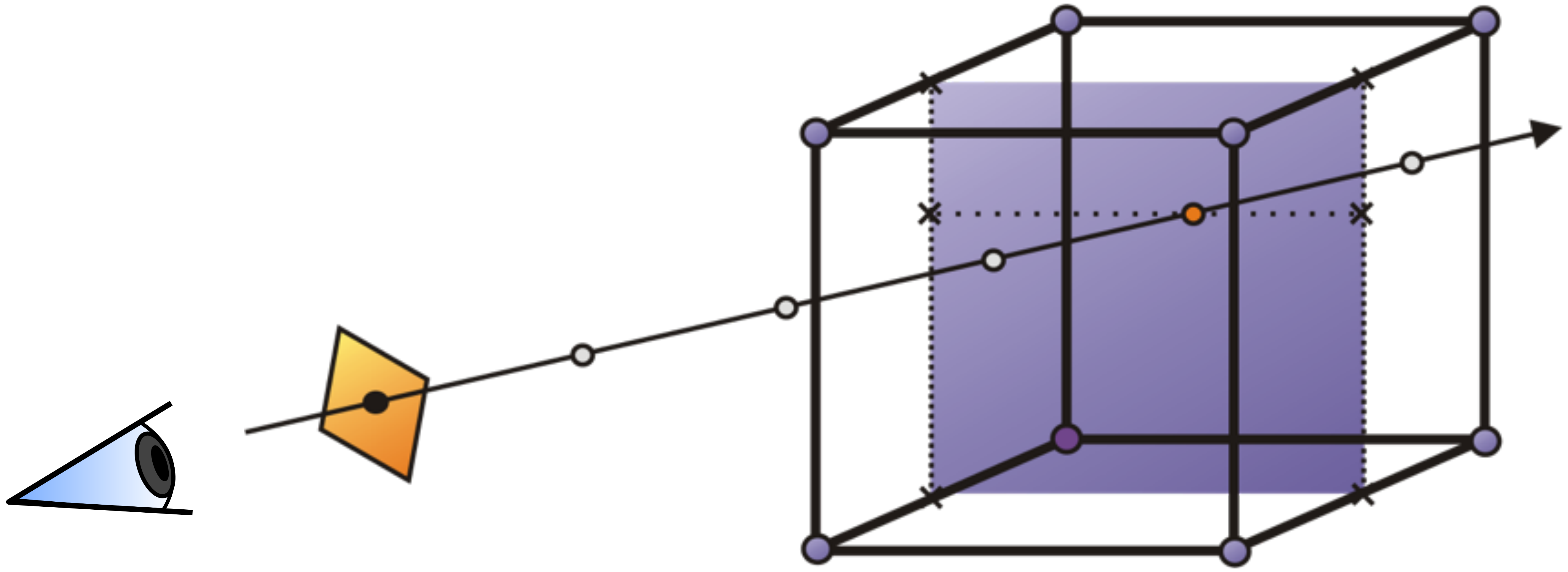


(a) Direct volume rendered (b) Isosurface rendered

[Kindlmann, 1998]

# How? Volume Ray Casting

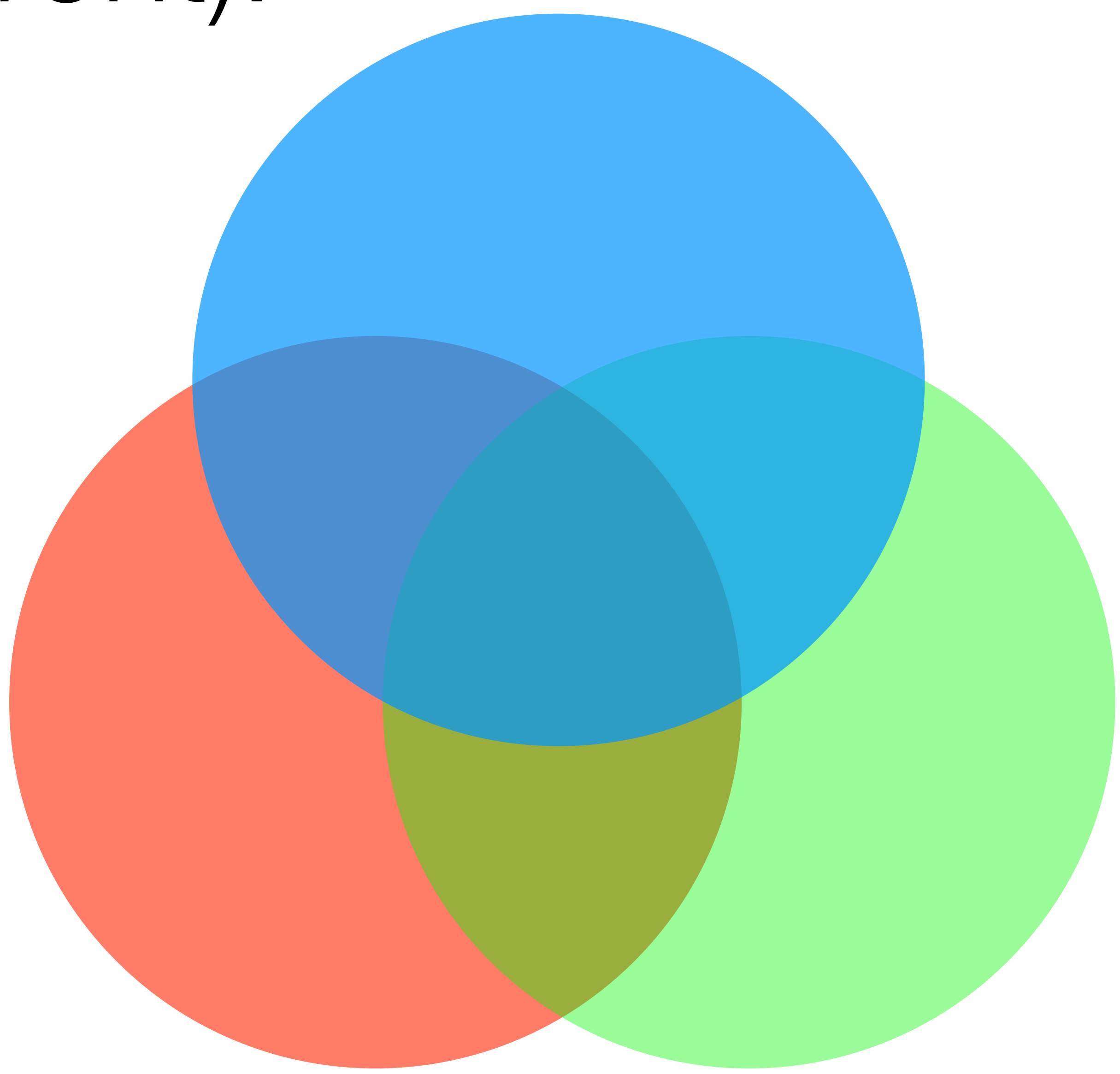
- Approximate volume rendering integral: light absorption & emission
- Sample at regular intervals along each ray
- Trilinear interpolation: linear interpolation along each axes (x,y,z)



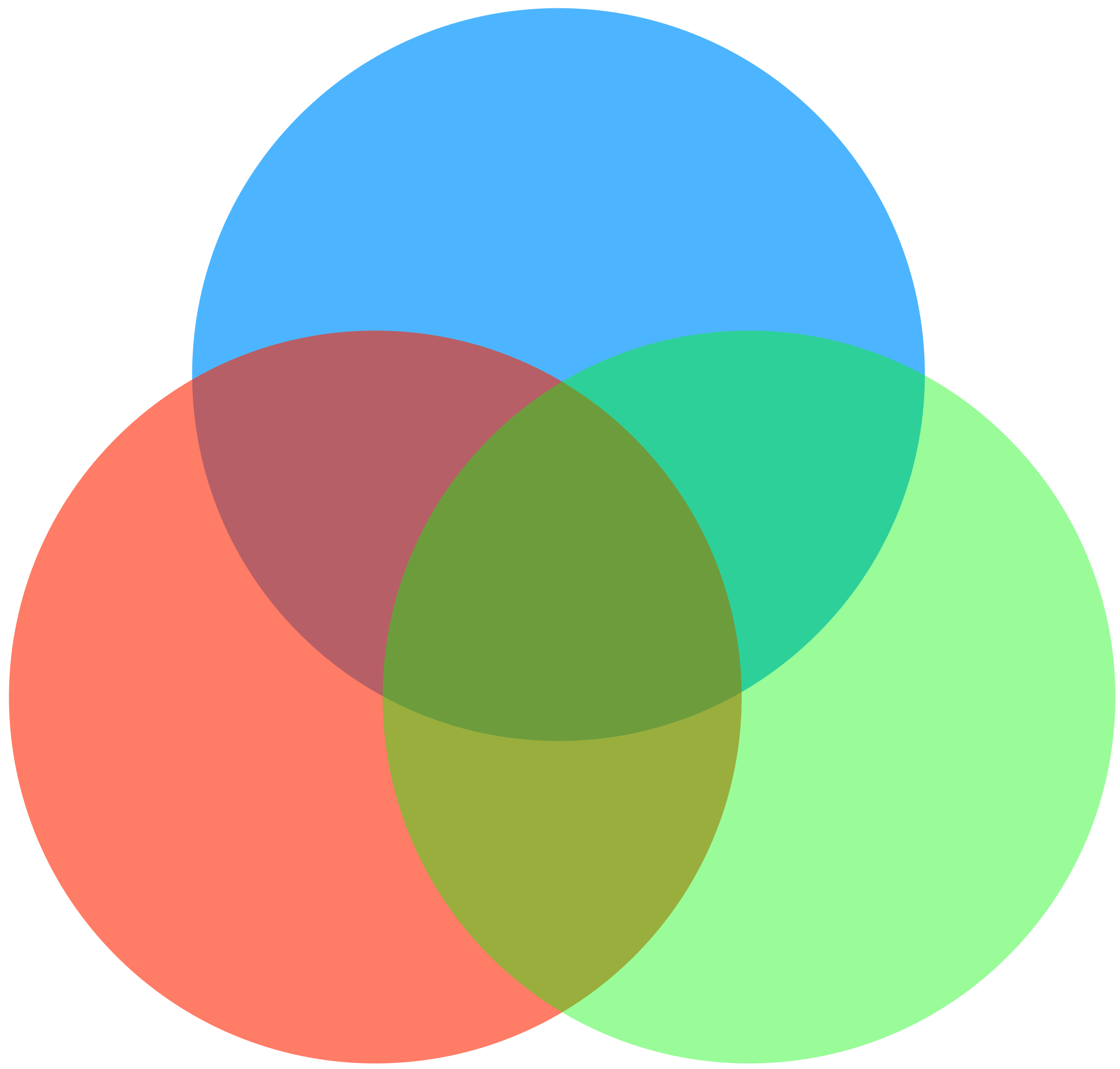
- Not the only possibility, also "object order" techniques like splatting or texture-based and combinations like shear-warp

# Accumulation

- If we're not just calculating a single number (max, average) or a position (first), how do we determine the accumulation?
- Assume each value has an associated color (c) and opacity (α)
- Over operator (back-to-front):
  - $c = \alpha_f \cdot c_f + (1-\alpha_f) \cdot \alpha_b \cdot c_b$
  - $\alpha = \alpha_f + (1-\alpha_f) \cdot \alpha_b$
- Order is important!



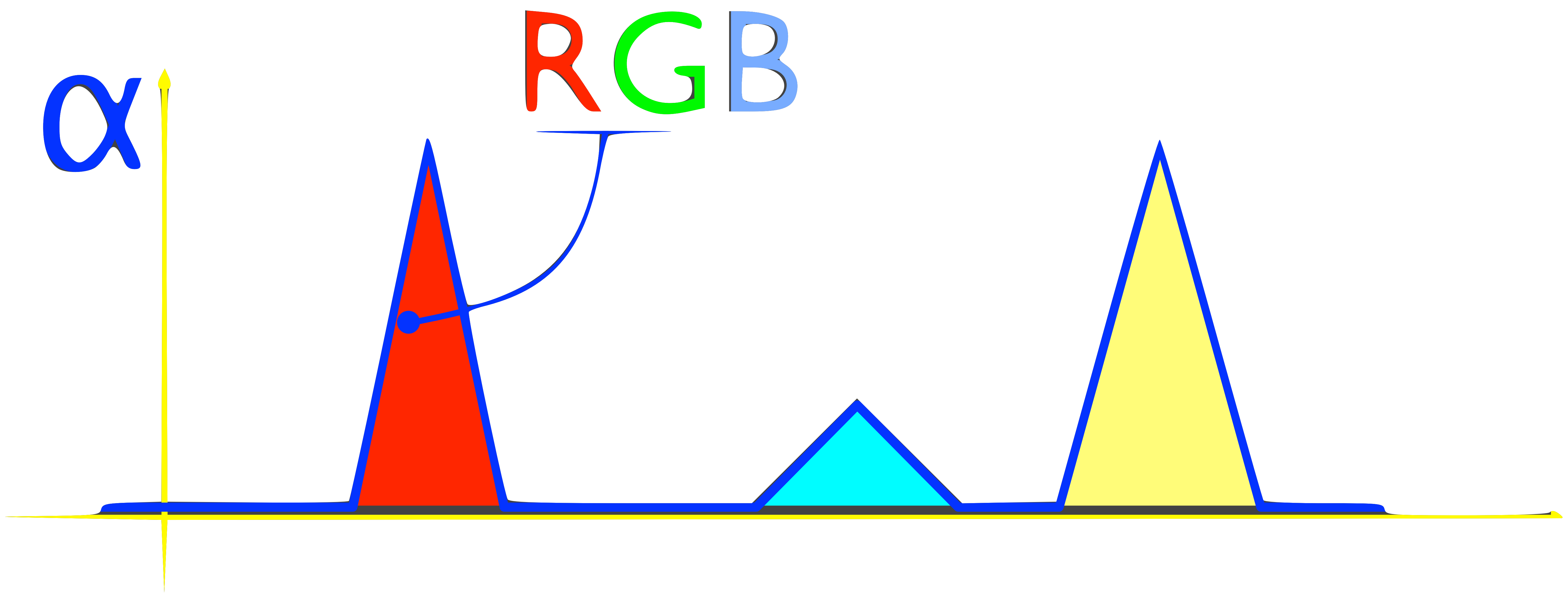
Blue Last



Blue First

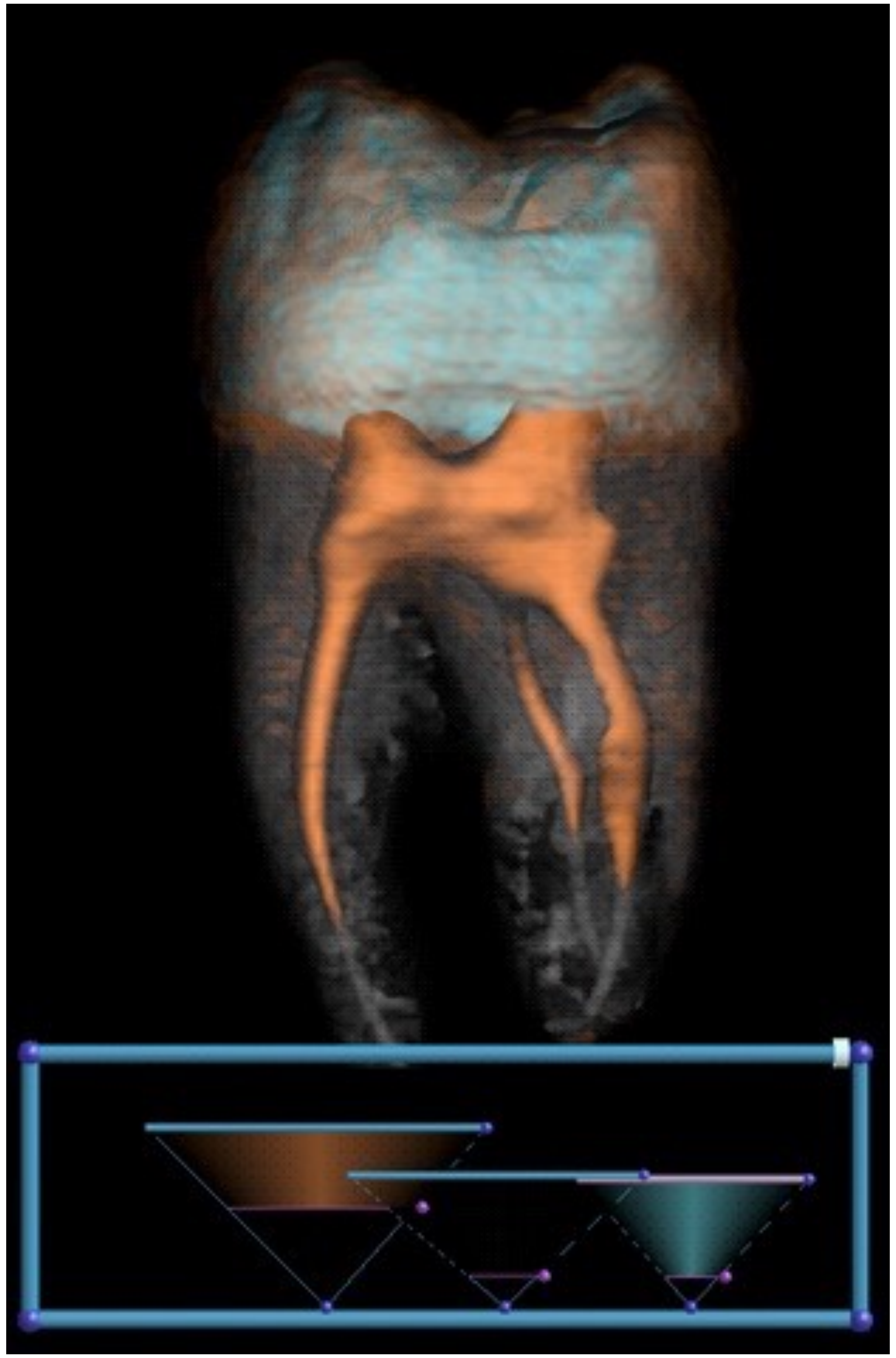
# Transfer Functions

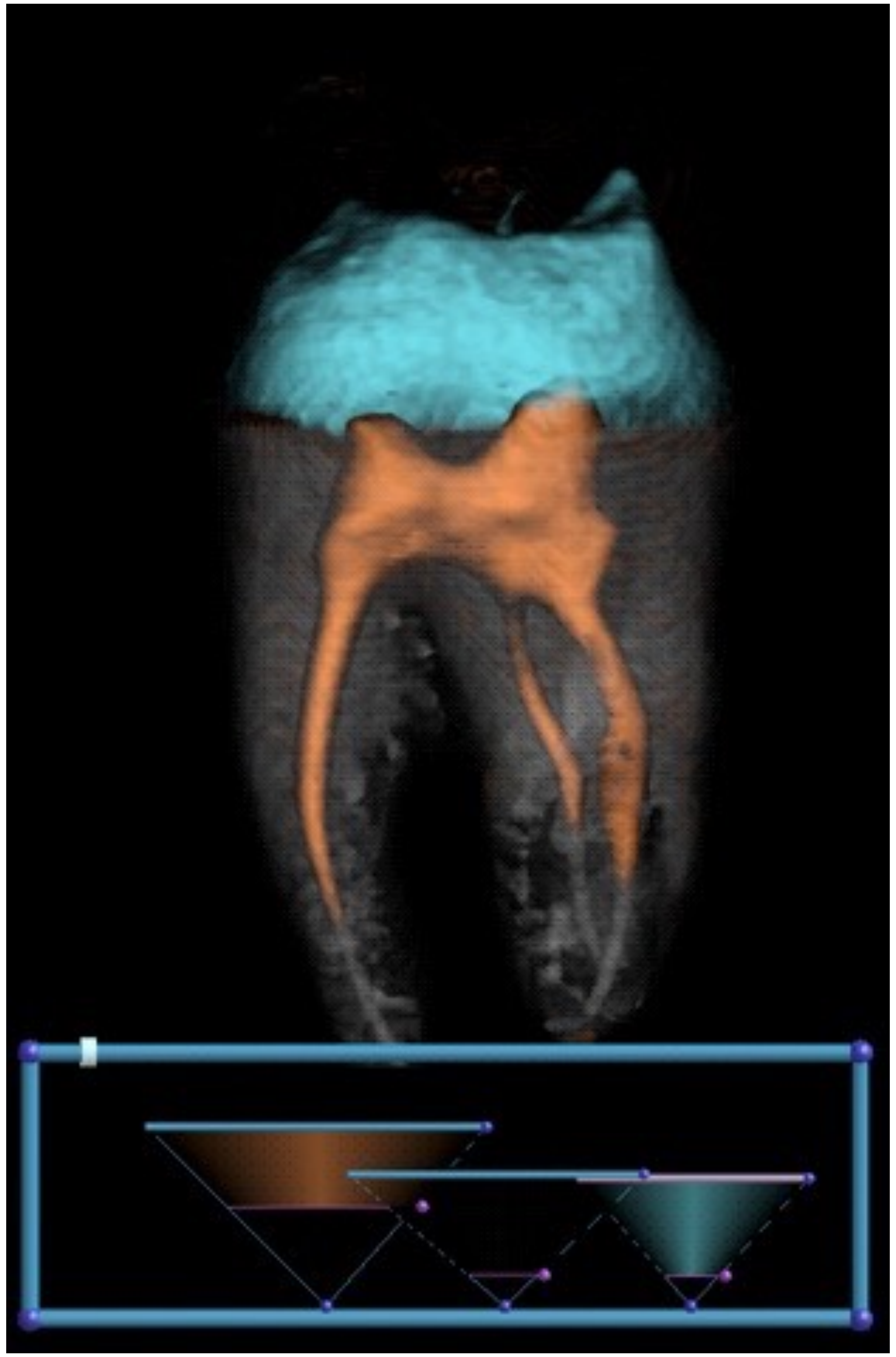
- Where do the colors and opacities come from?
- Idea is that each voxel emits/absorbs light based on its scalar value
- …but users get to choose how that happens
- x-axis: color region definitions, y-axis: opacity




[Kindlmann]

# Multidimensional Transfer Functions





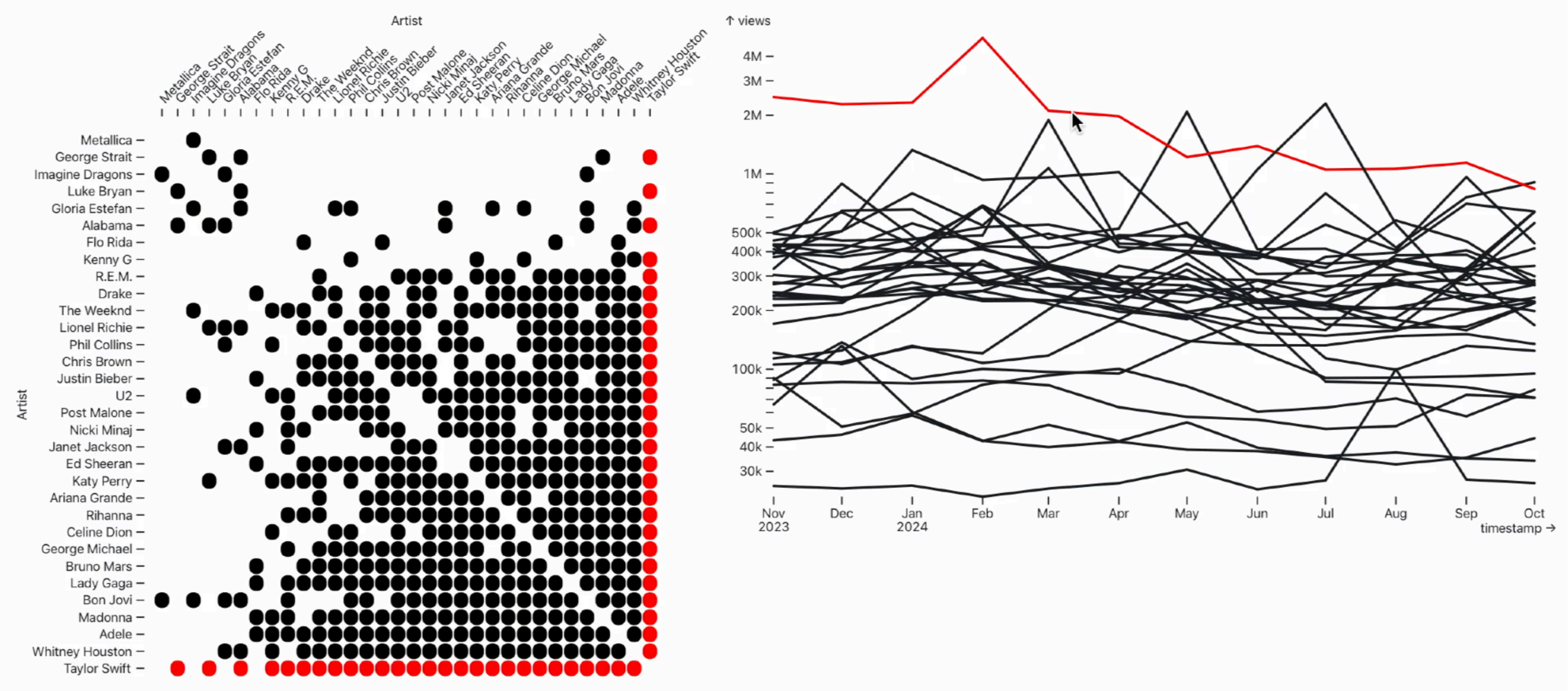
[J. Kniss]

# Newer Technology

- Intel OSPRay
- https://www.ospray.org/gallery.html

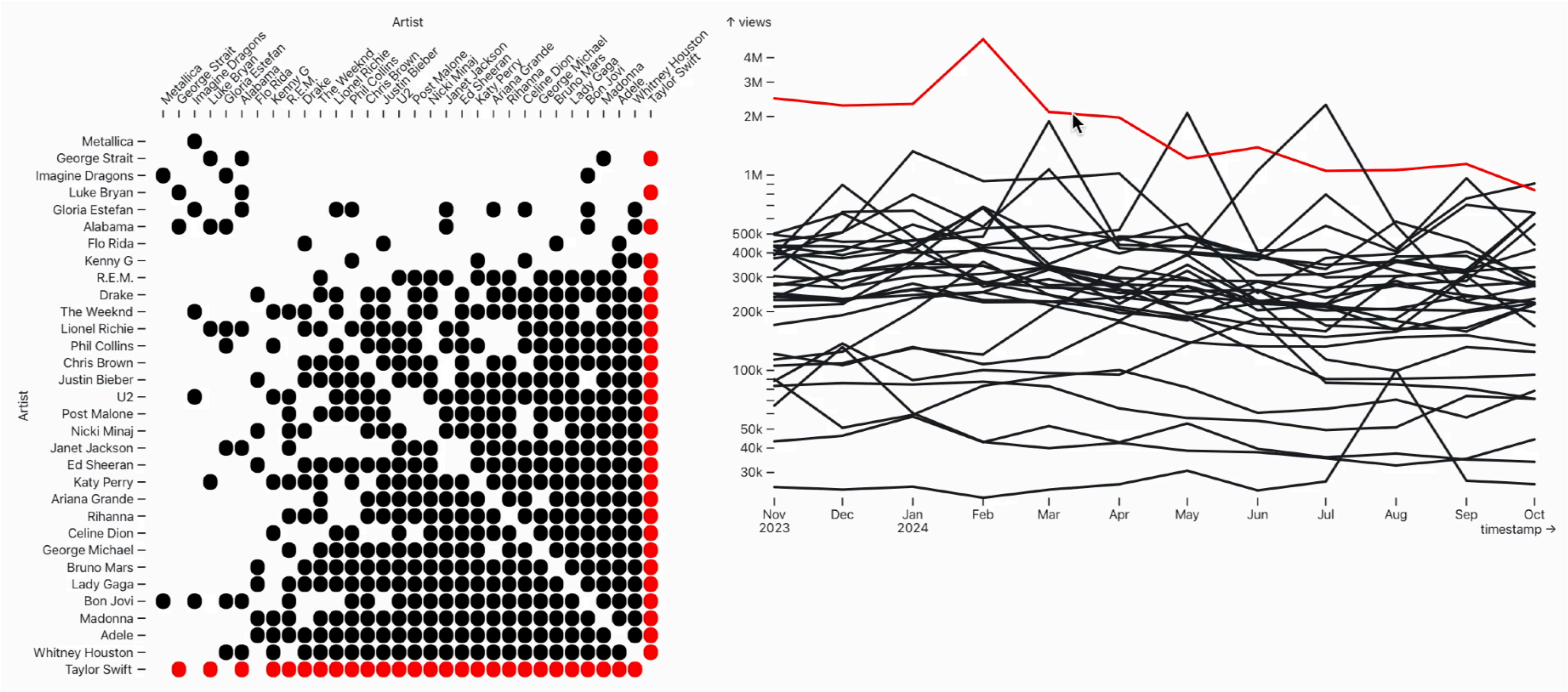
# Assignment 5

- Adjacency Matrix
- Line Graph
- Linked Highlighting



# Assignment 5

- Adjacency Matrix
- Line Graph
- Linked Highlighting



# Projects

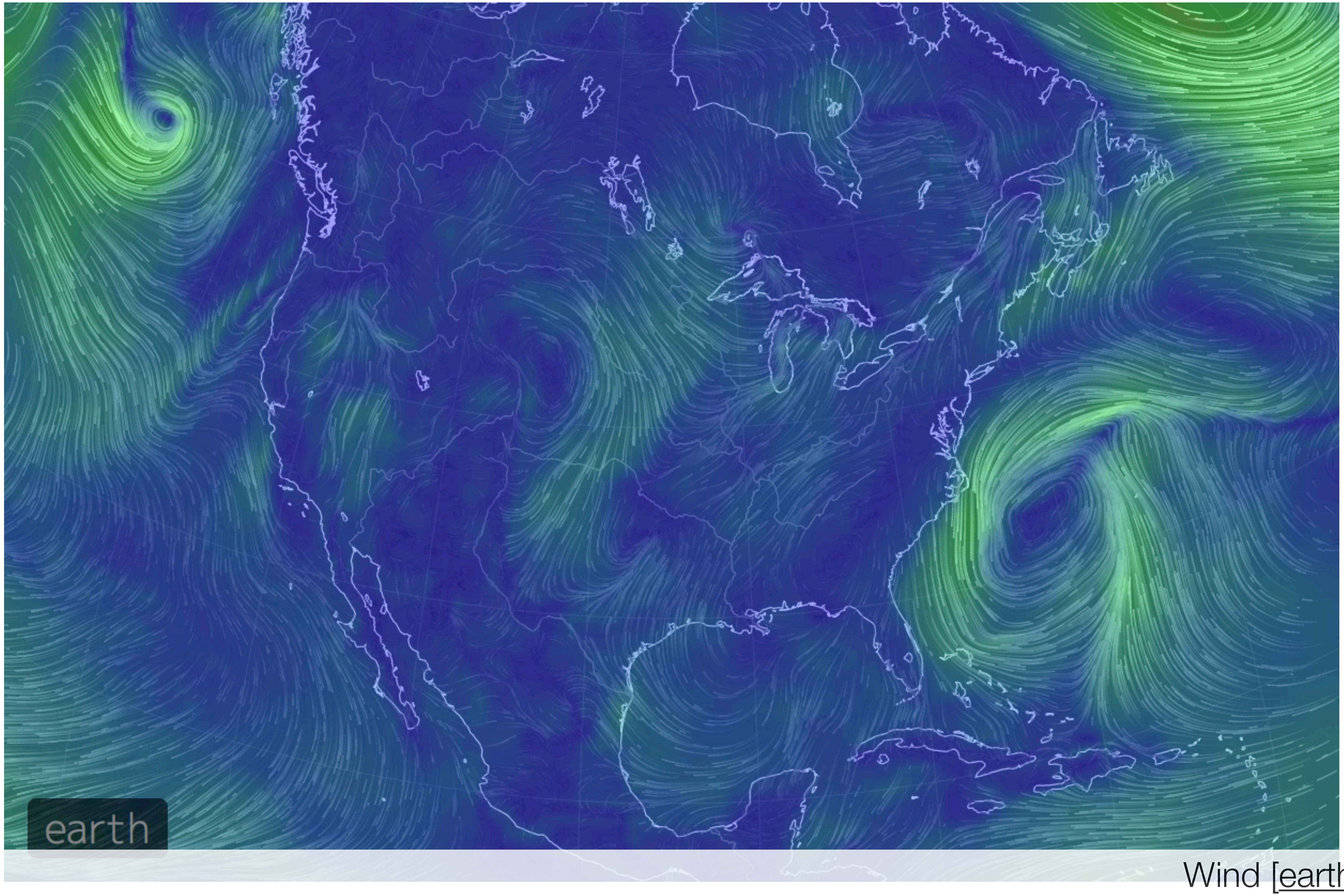
- Keep working on implementation
- Be creative
- Think about interaction
- Presentations on the last two days of class (Dec. 2 & Dec. 4)
  - Submit current visualization code (or a link) to Blackboard
  - Presentation preferences (Monday or Wednesday)
  - Upload link / full code to Blackboard beforehand in case of technical issues
- Can keep working on final project & report until end of semester

# Final Exam

- December 9, 2024, **12:00-1:50pm**
- Covers all topics but emphasizes second half of the course
- Similar format as Midterm (multiple choice, free response)
- 627 Students will have a extra questions related to the research papers

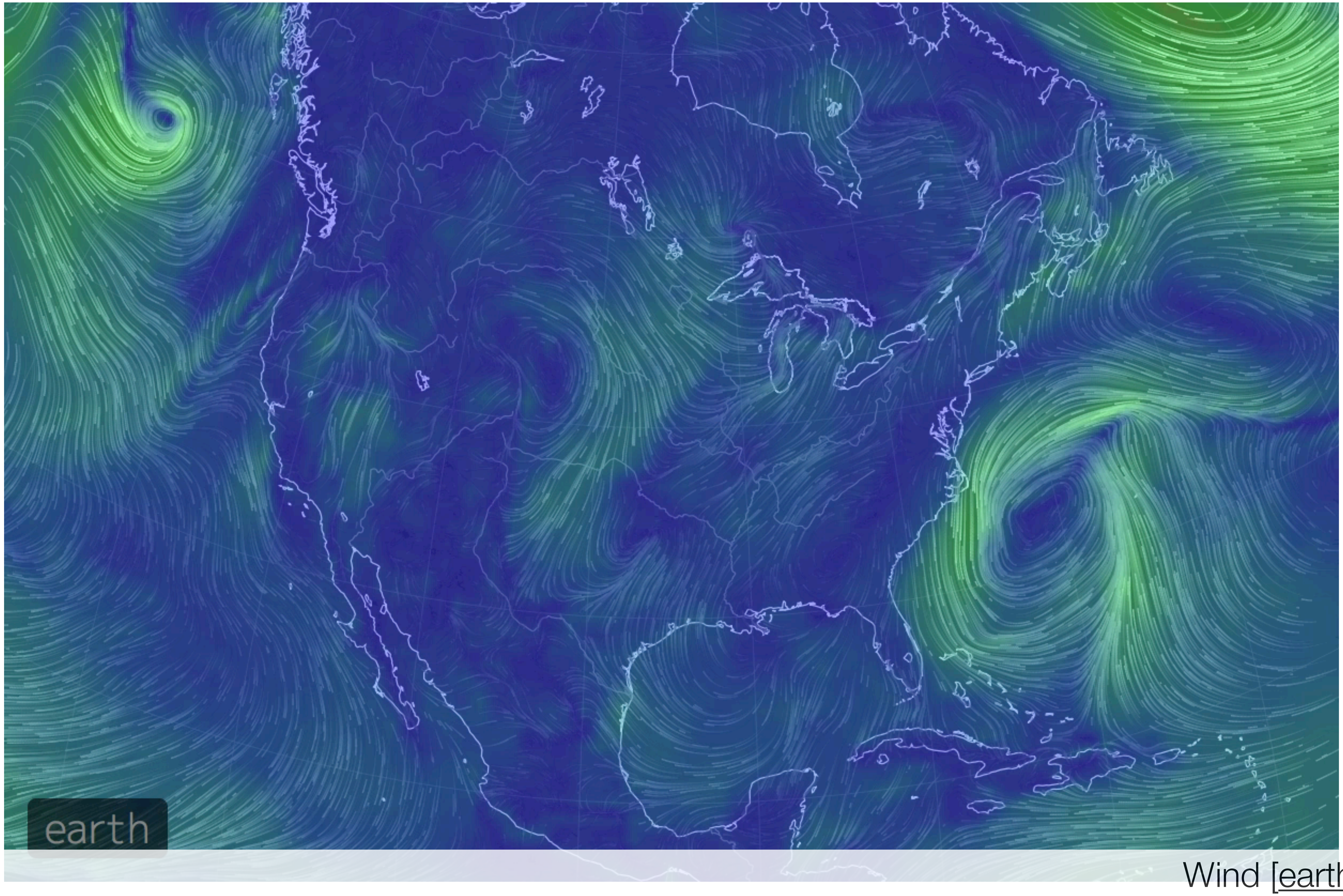
# Vector Field Visualization

# Examples of Vector Fields



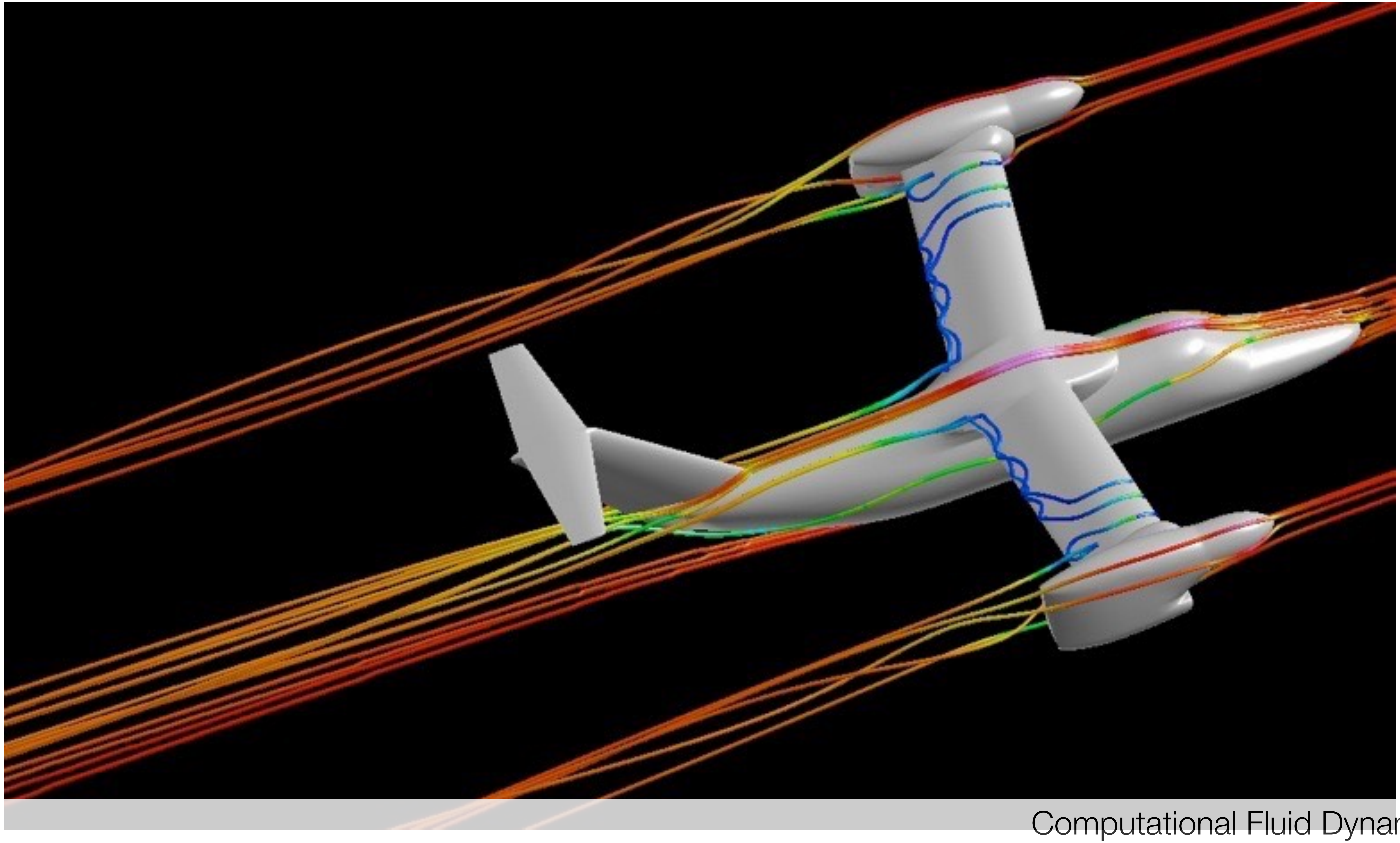

Wind [earth.nullschool.net, 2014]

# Examples of Vector Fields



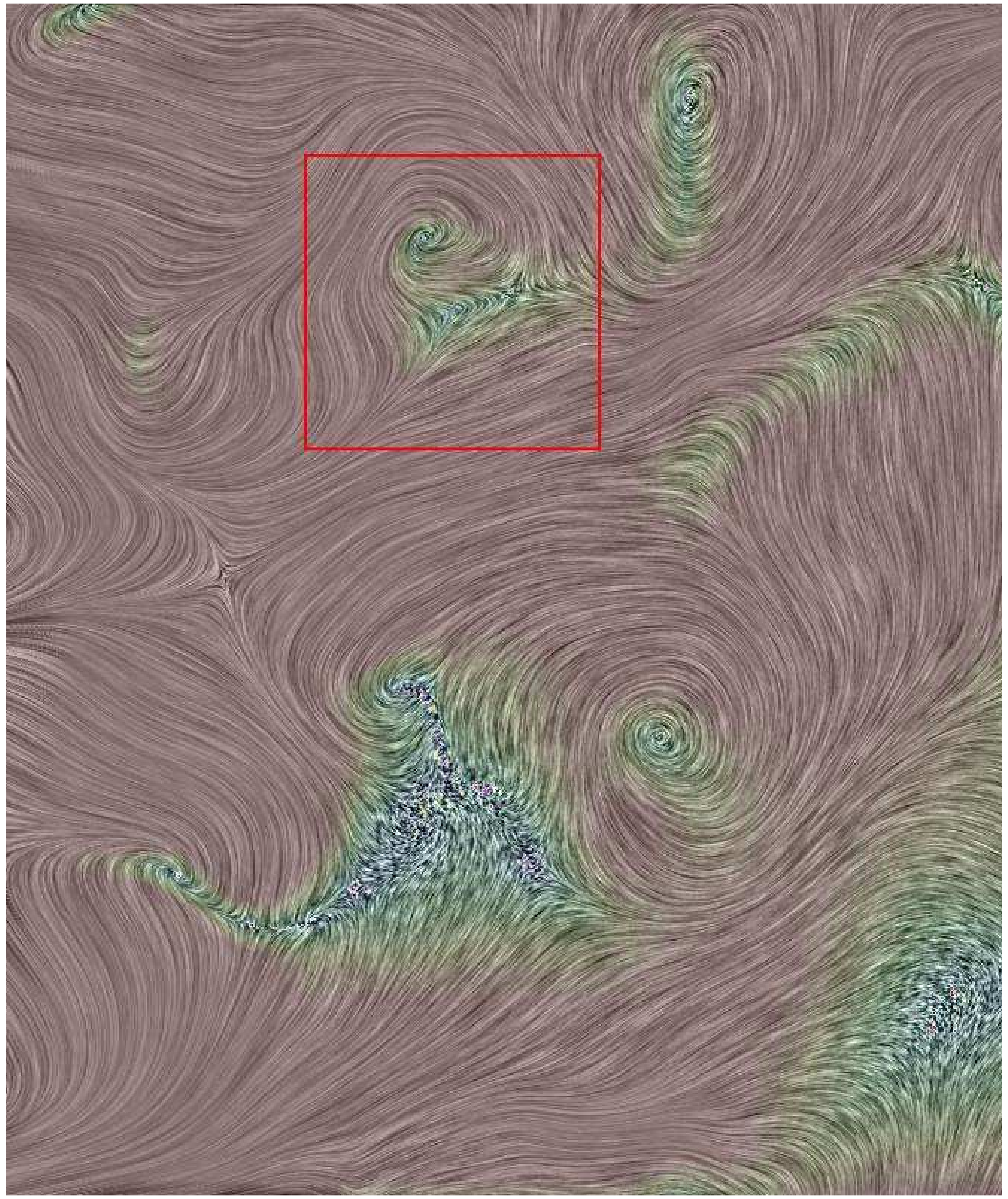

Wind [earth.nullschool.net, 2014]

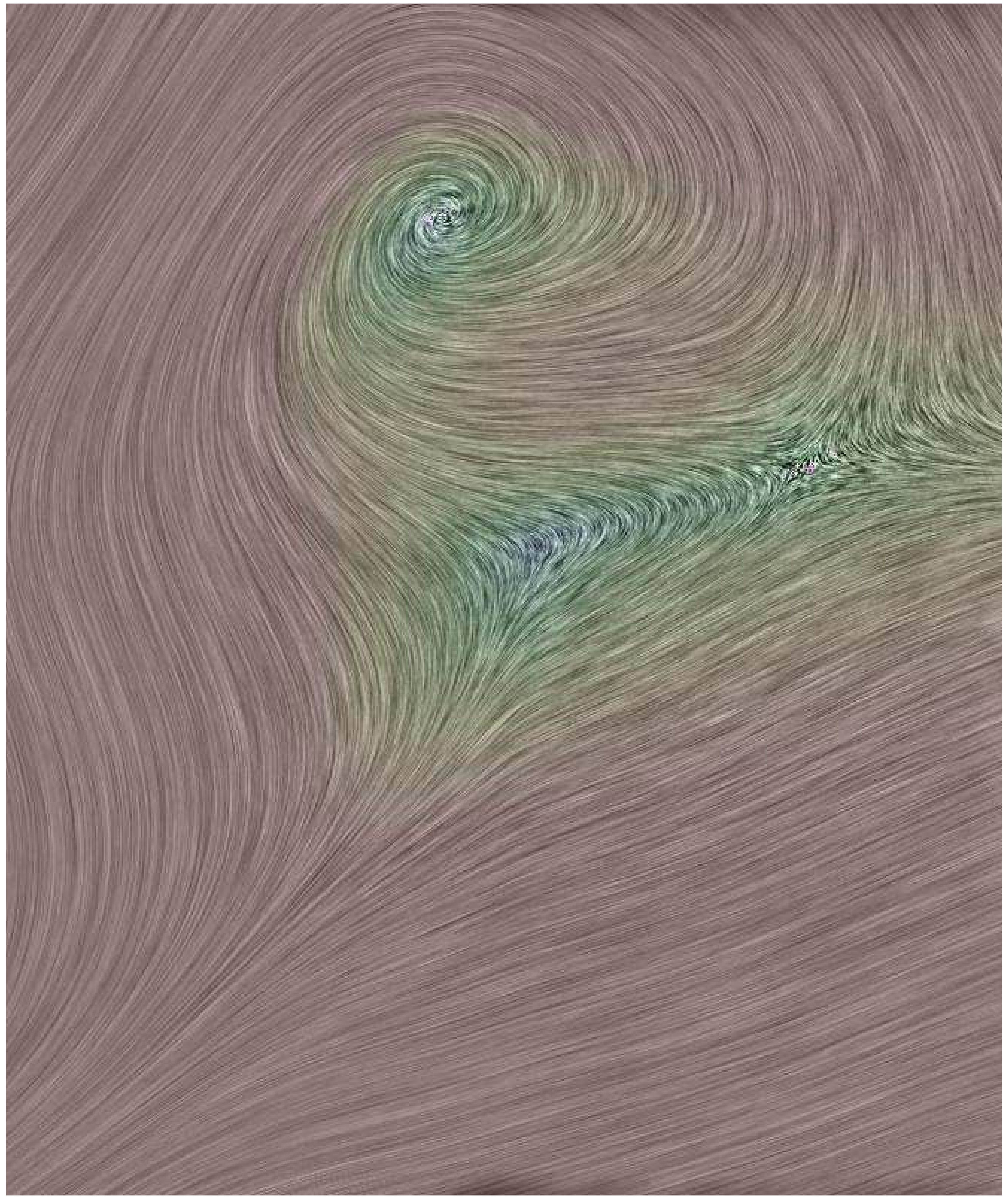
# Examples of Vector Fields



Computational Fluid Dynamics [newmerical]

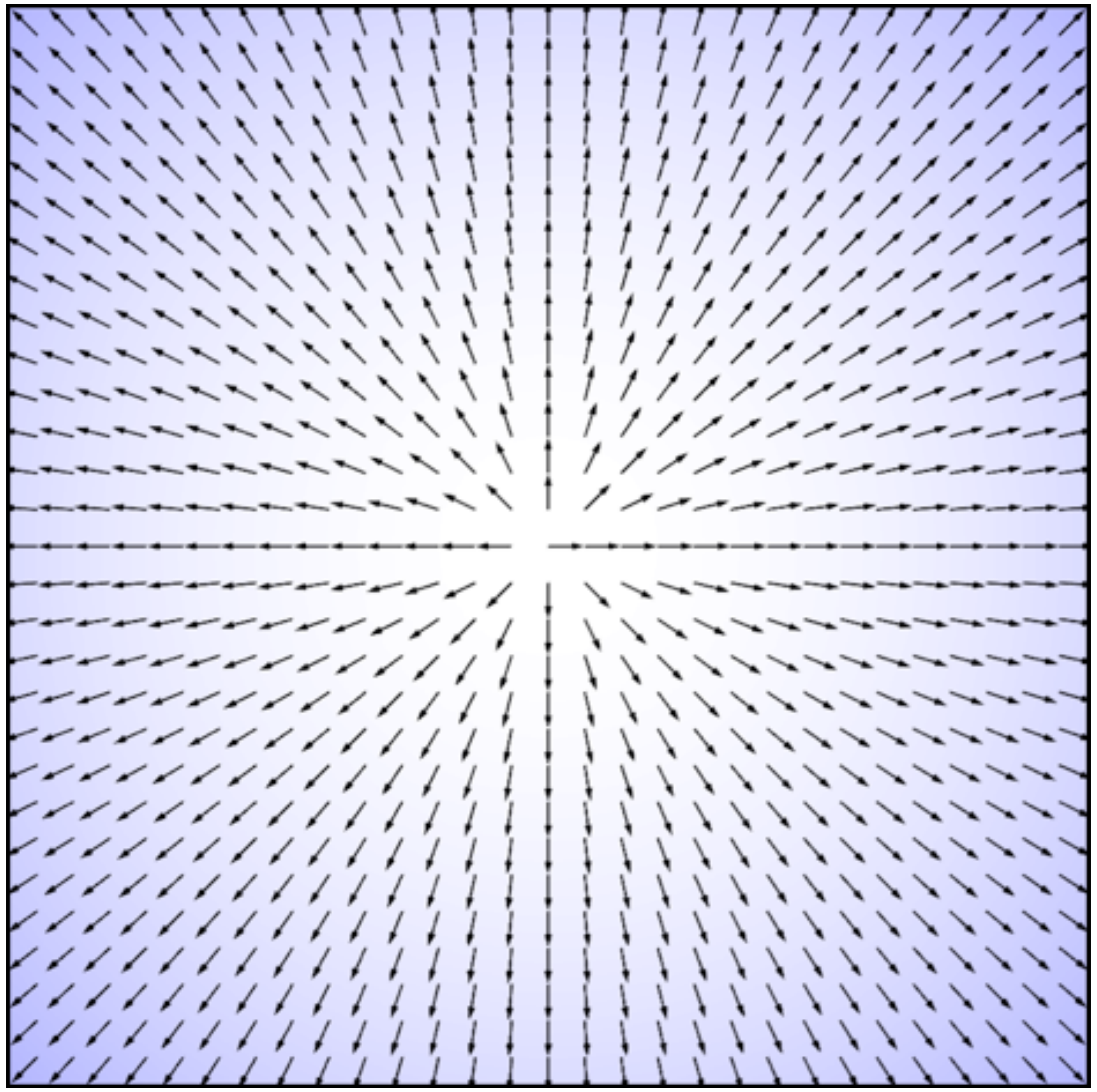
# Examples of Vector Fields

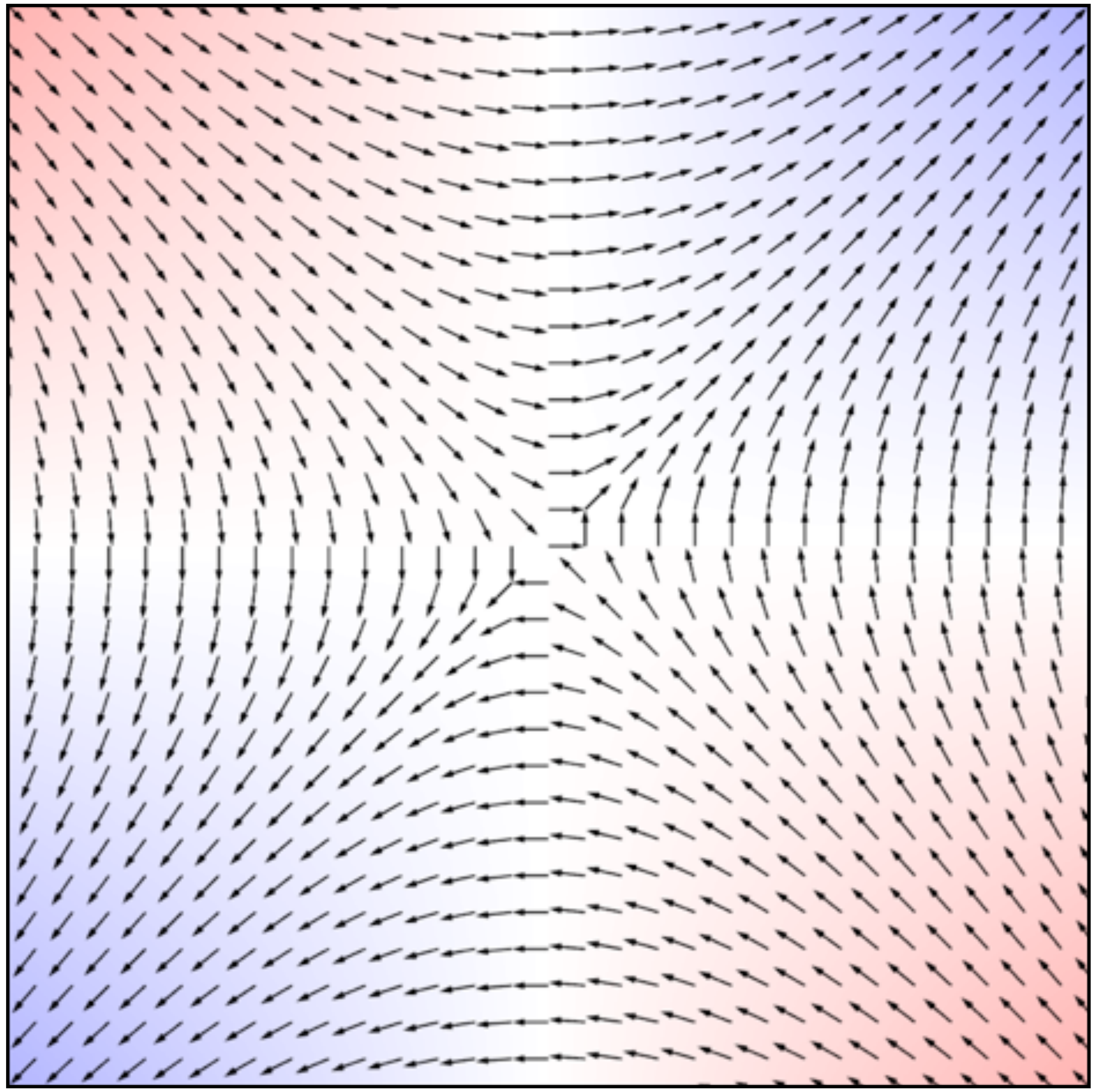




Earthquake Ground Surface Movement [H. Yu et. al., SC2004]

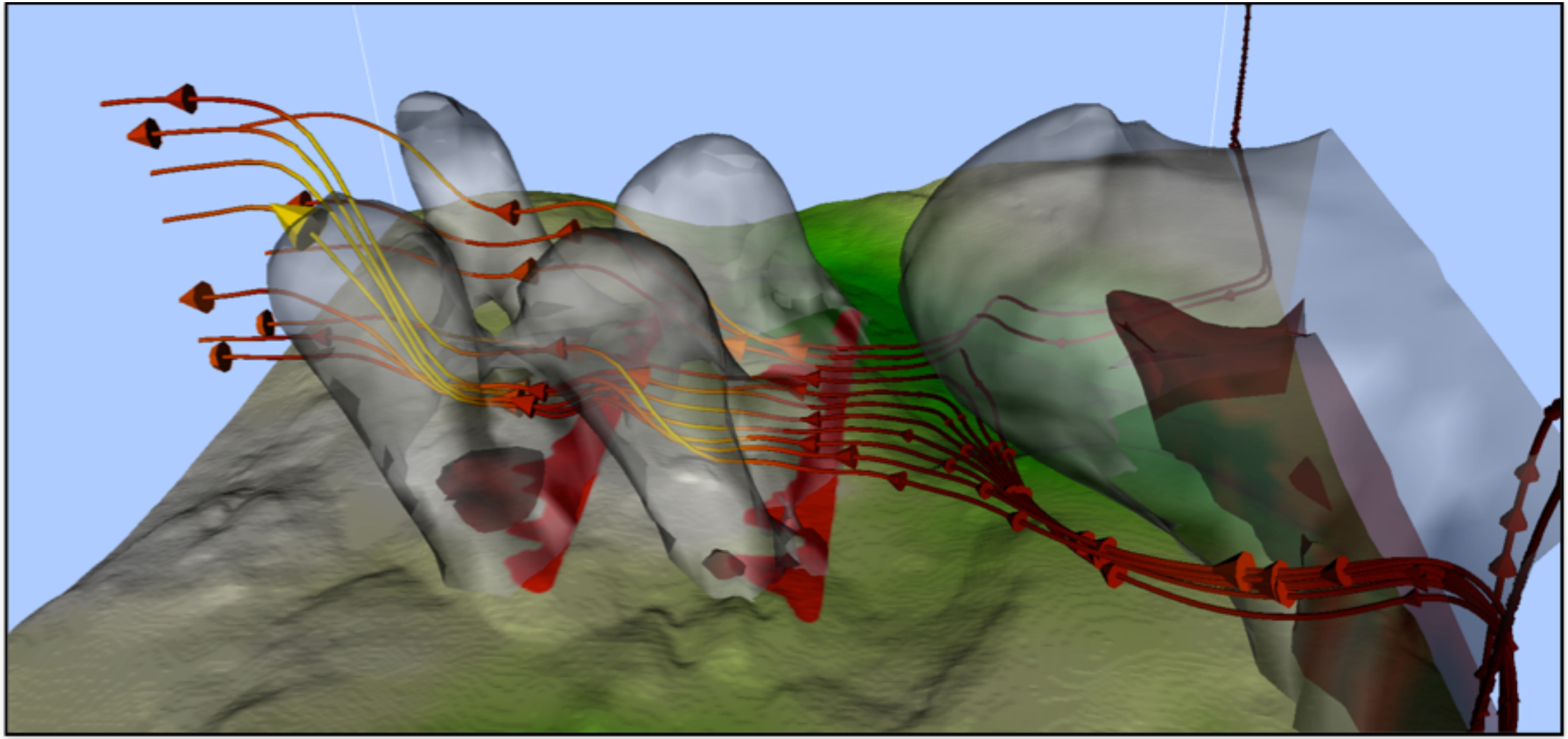
# Examples of Vector Fields





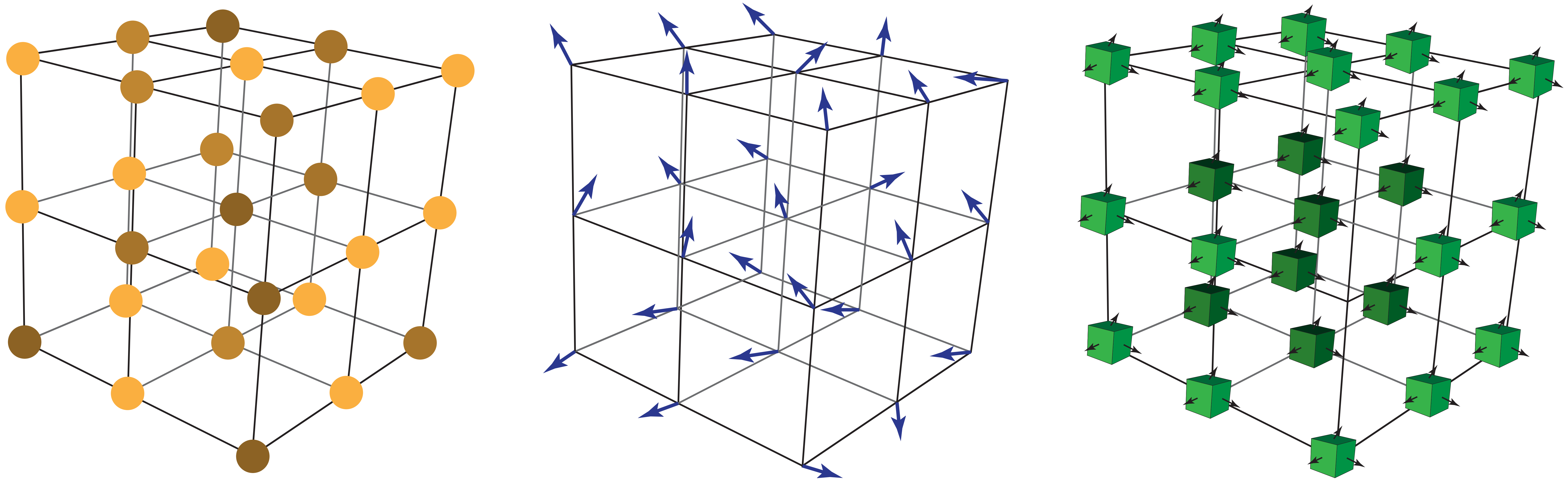
Gradient Vector Fields

# Examples of Vector Fields



Wildfire Modeling [E. Anderson]

# Fields in Visualization



| Scalar Fields | Vector Fields | Tensor Fields |
|---|---|---|
| (Order-0 Tensor Fields) | (Order-1 Tensor Fields) | (Order-2+) |

Each point in space has an associated...

| $s_0$ | $\begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}$ | $\begin{bmatrix} \sigma_{00} & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11} & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22} \end{bmatrix}$ |
|---|---|---|
| Scalar | Vector | Tensor |

# Visualizing Vector Fields

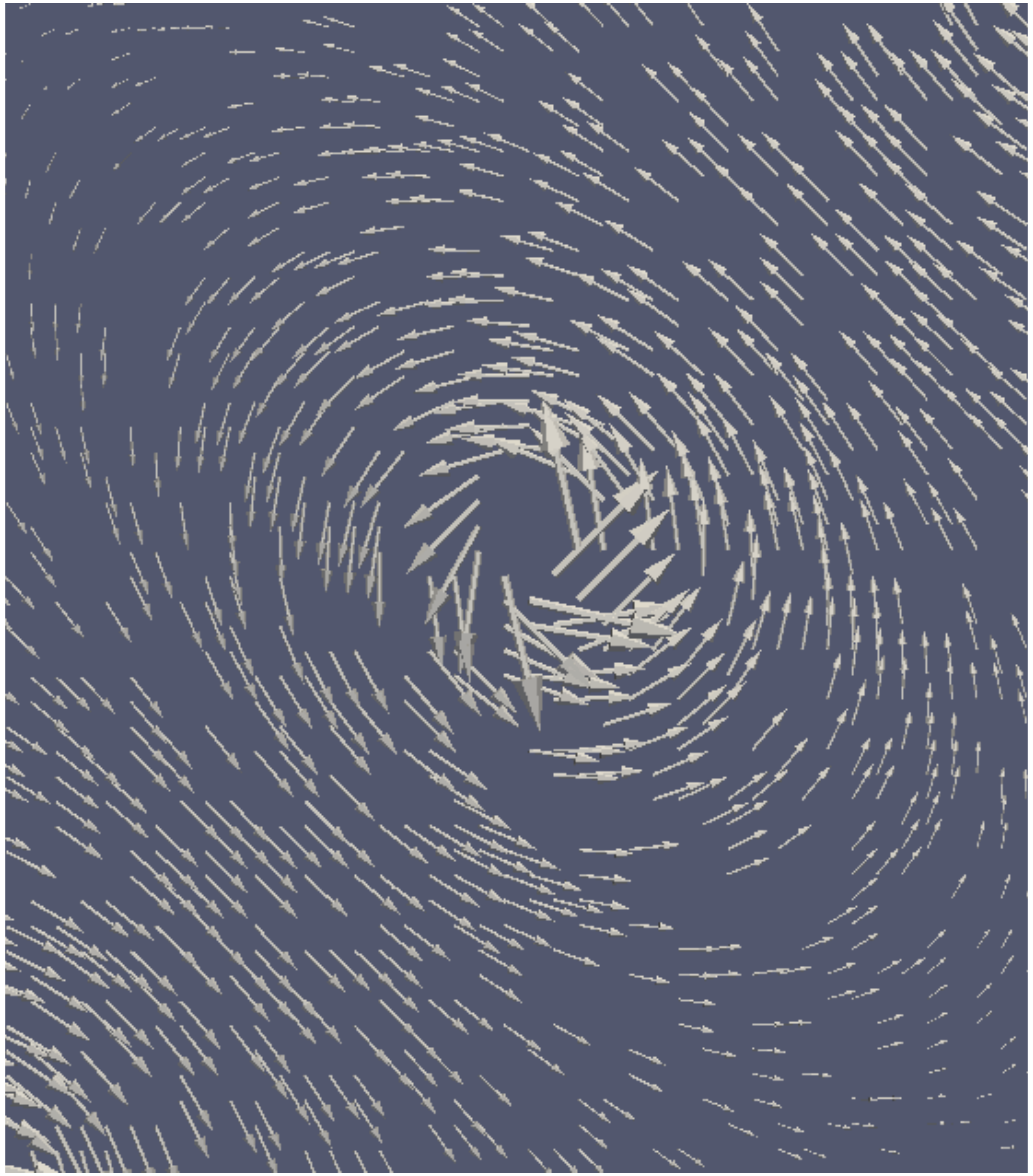
- Direct: Glyphs, Render statistics as scalars
- Geometry: Streamlines and variants
- Textures: Line Integral Convolution (LIC)
- Topology: Extract relevant features and draw them

# Glyphs

- Represent each vector with a symbol
- Hedgehogs are primitive glyphs (glyph is a line)
- ParaView Example

# Glyphs

- Represent each vector with a symbol
- Hedgehogs are primitive glyphs (glyph is a line)
- Glyphs that show direction and/or magnitude can convey more information
- If we have a separate scalar value, how might we encode that?
- Clutter issues

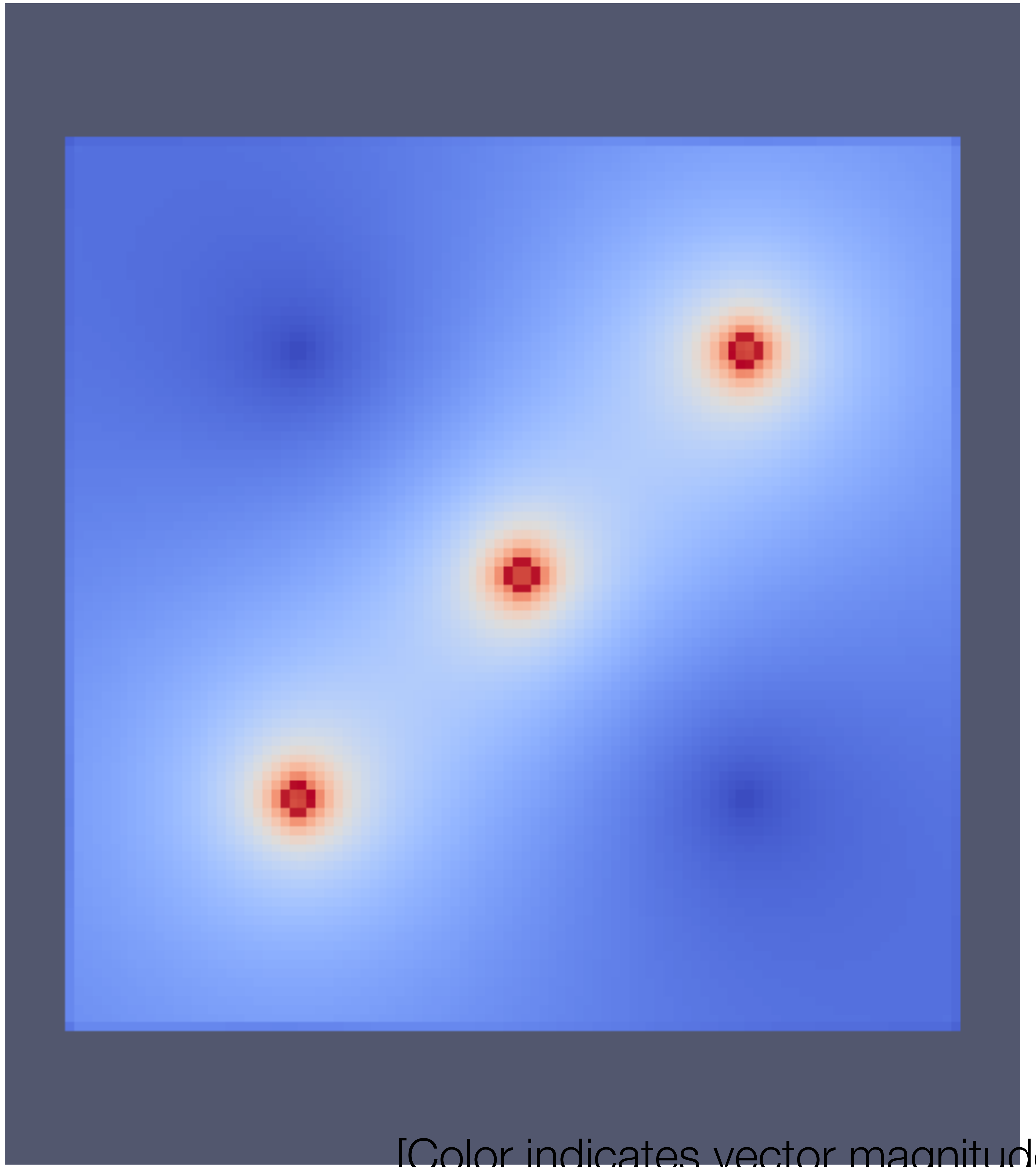


# Glyphs

- For vector fields, can encode direction, magnitude, scalar value
- Good:
  - Show precise local measures
  - Can encode scalar information as color
- Bad:
  - Possible sampling issues
  - Clutter (Occlusion): Can remove some points to help
  - Clutter is worse in higher dimensions

# Rendering Vector Field Statistics as Scalars

- Many statistics we can compute for vector fields:
  - Magnitude
  - Vorticity
  - Curvature
- These are scalars, can color with our scalar field visualization techniques (e.g. volume rendering)



[Color indicates vector magnitude]

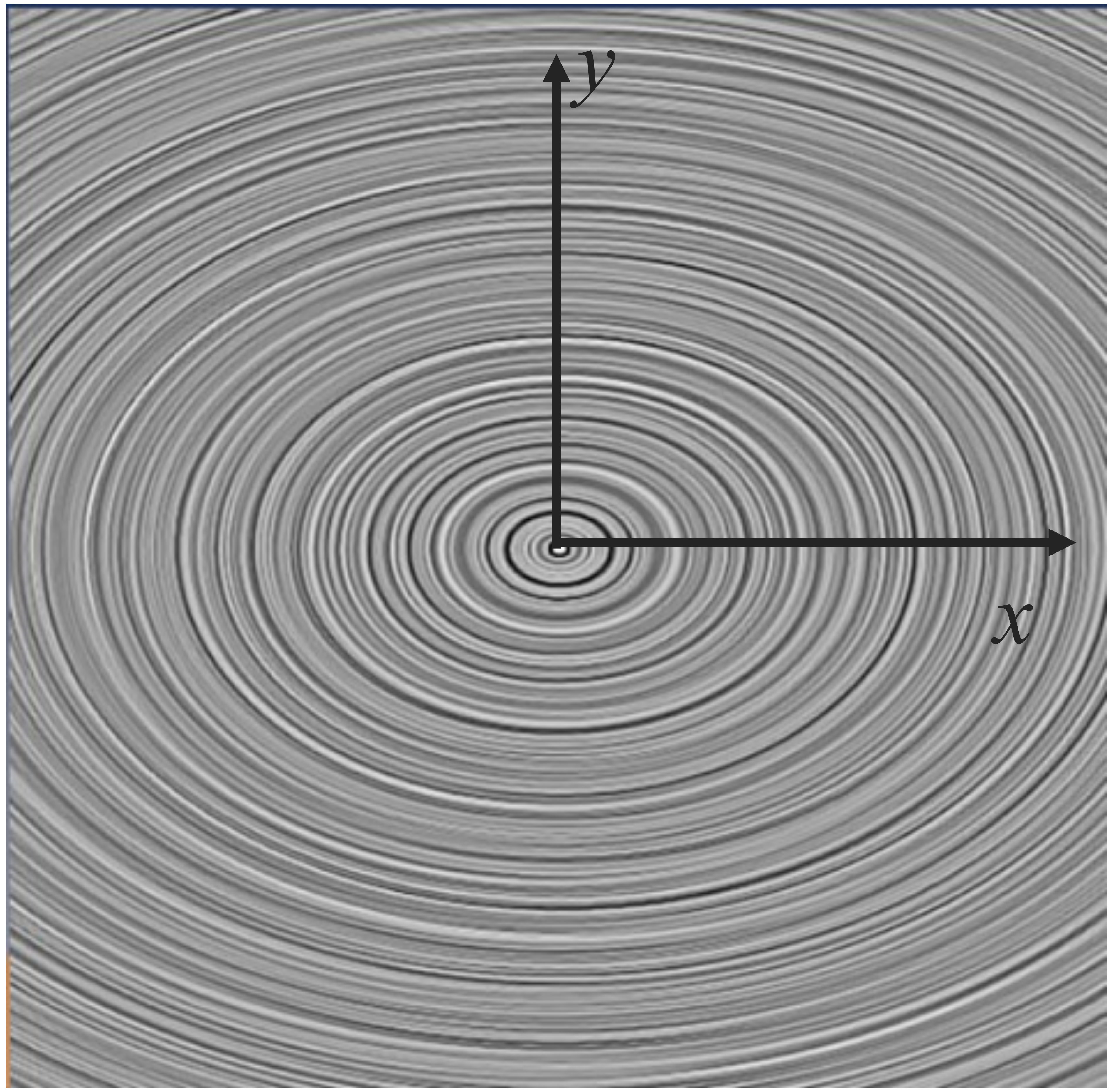
# Streamlines & Variants

- Trace a line along the direction of the vectors
- Streamlines are always tangent to the vector field
- Basic Particle Tracing:

1. Set a starting point (seed)
2. Take a step in the direction of the vector at that point
3. Adjust direction based on the vector where you are now
4. Go to Step 2 and Repeat

# Example

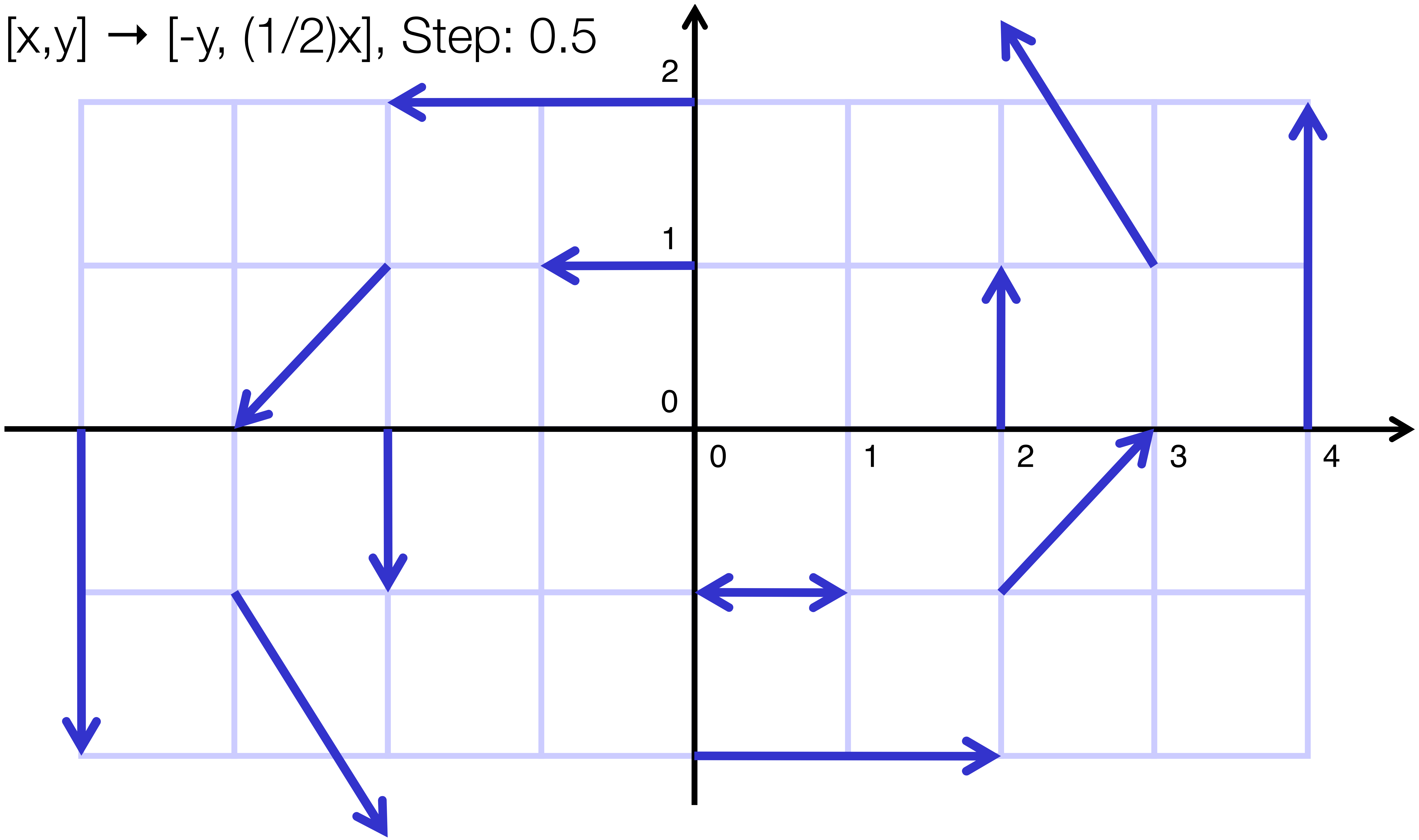
- Elliptical path
- Suppose we have the actual equation
- Given point (x,y), the vector is at that point is $[v_x, v_y]$ where
  - $v_x = -y$
  - $v_y = (1/2)x$
- Want a streamline starting at (0,-1)




[LIC (not streamlines!) via Levine]

# Some Glyphs

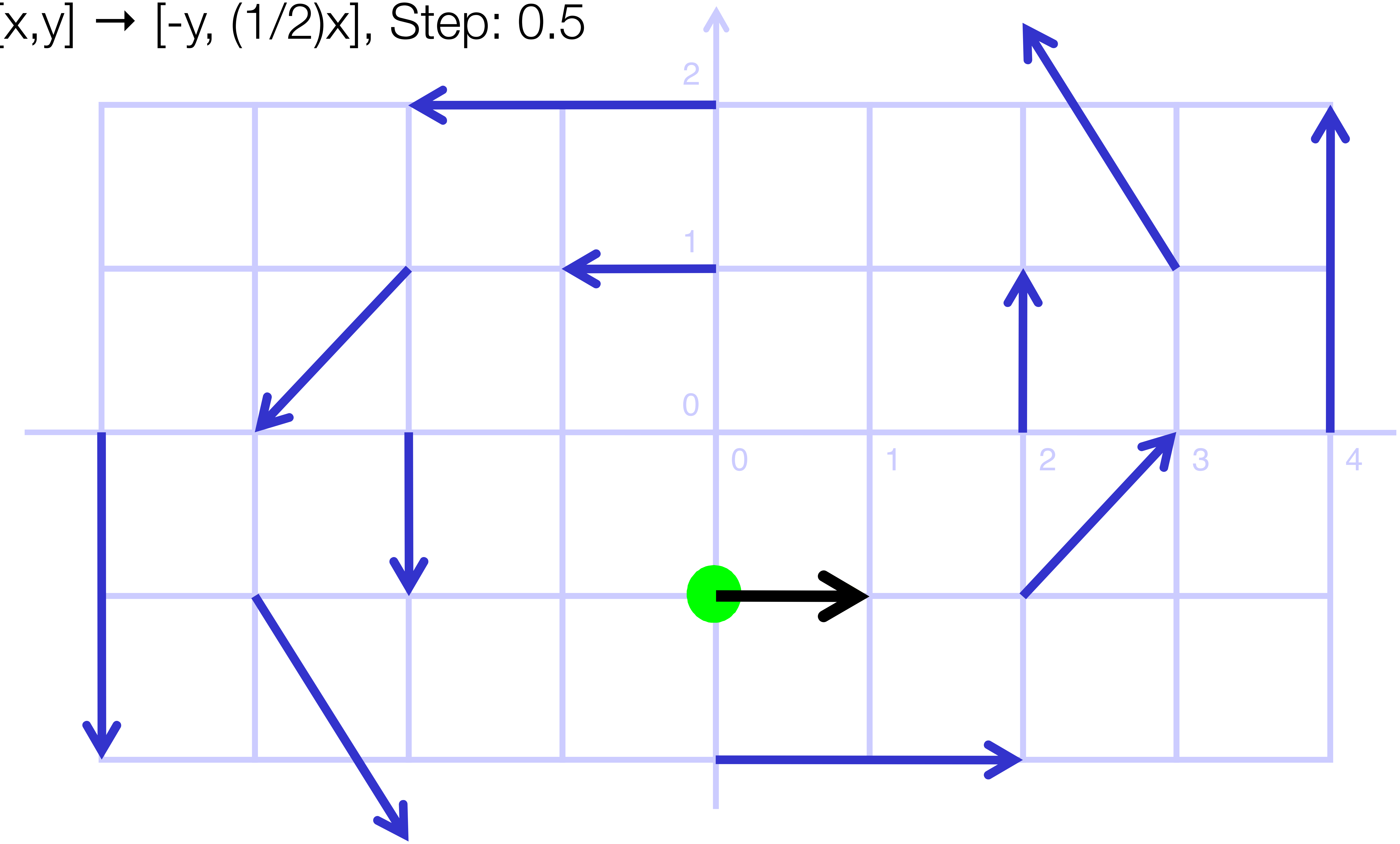
[x,y] → [-y, (1/2)x], Step: 0.5




[via Levine]

# Streamlines (Step 1)

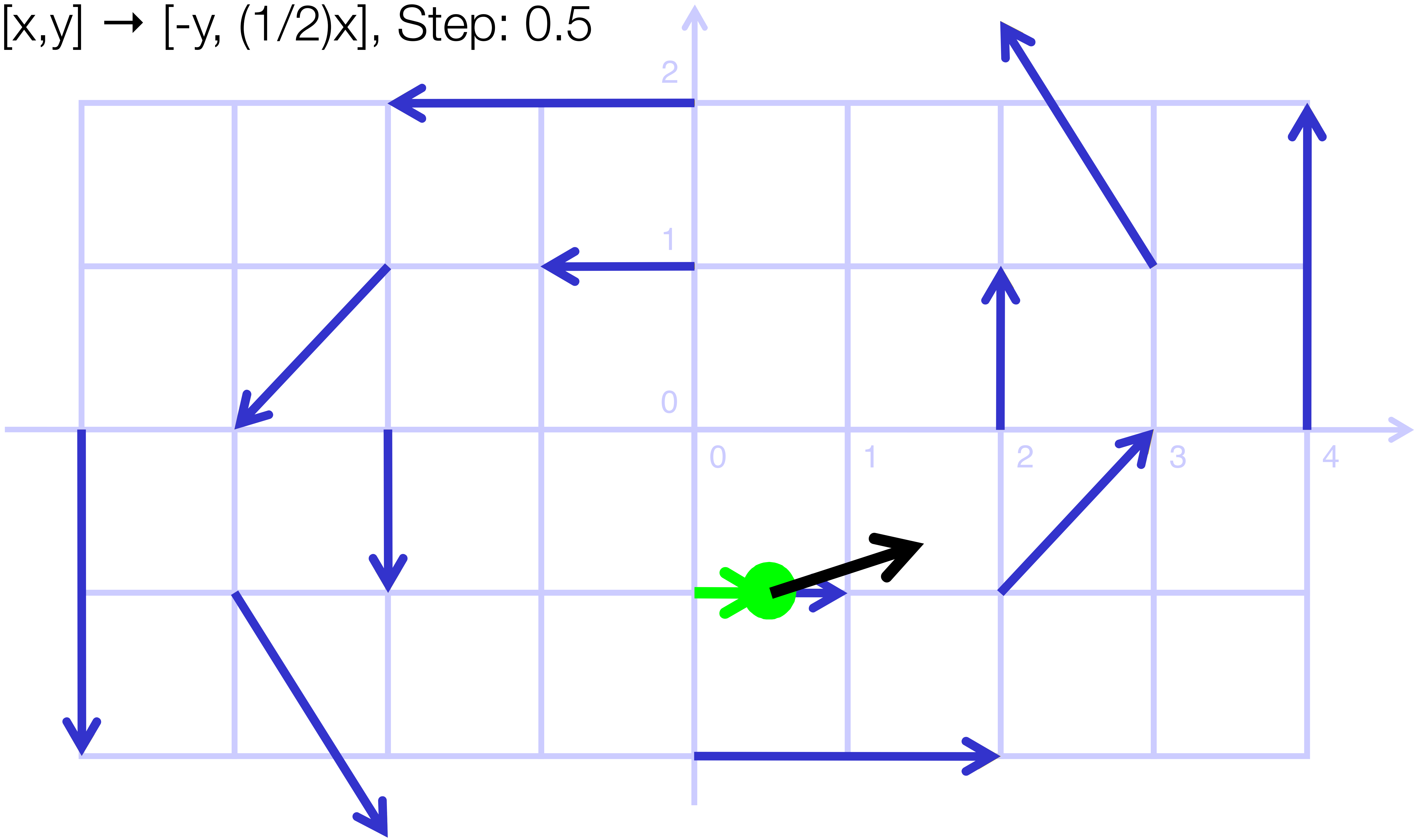
[x,y] → [-y, (1/2)x], Step: 0.5




[via Levine]

# Streamlines (Step 2)

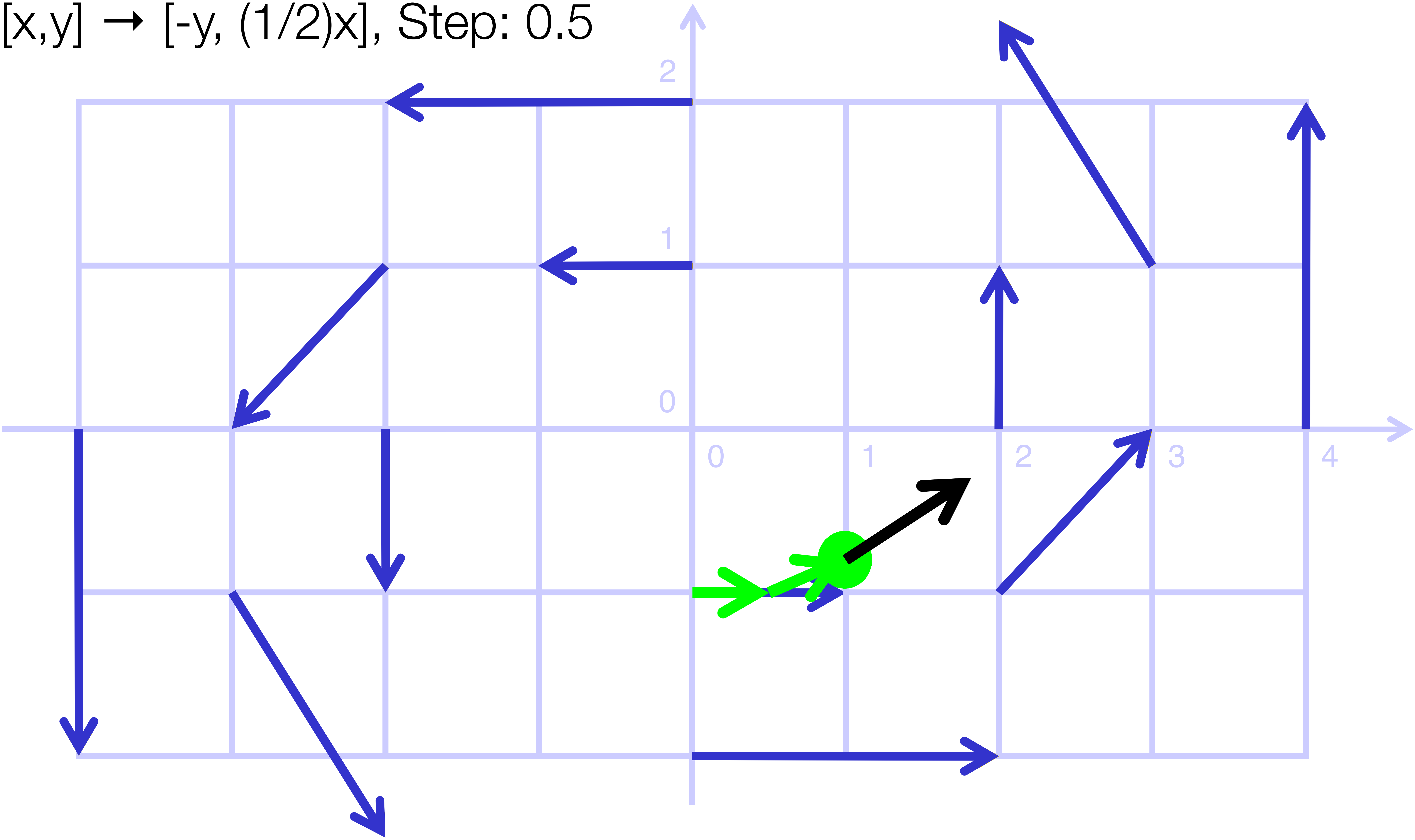
[x,y] → [-y, (1/2)x], Step: 0.5




[via Levine]

# Streamlines (Step 3)

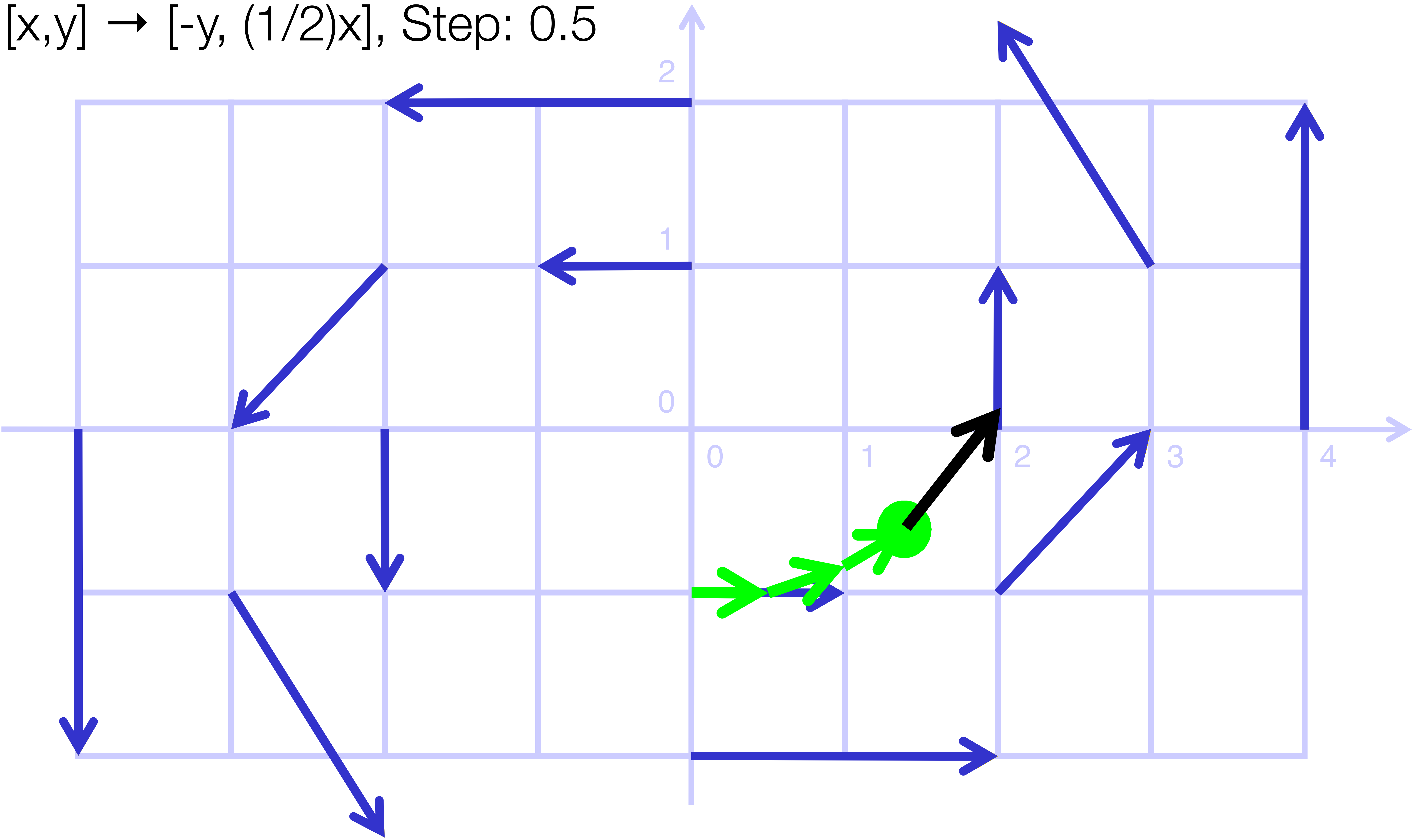
[x,y] → [-y, (1/2)x], Step: 0.5




[via Levine]

# Streamlines (Step 4)

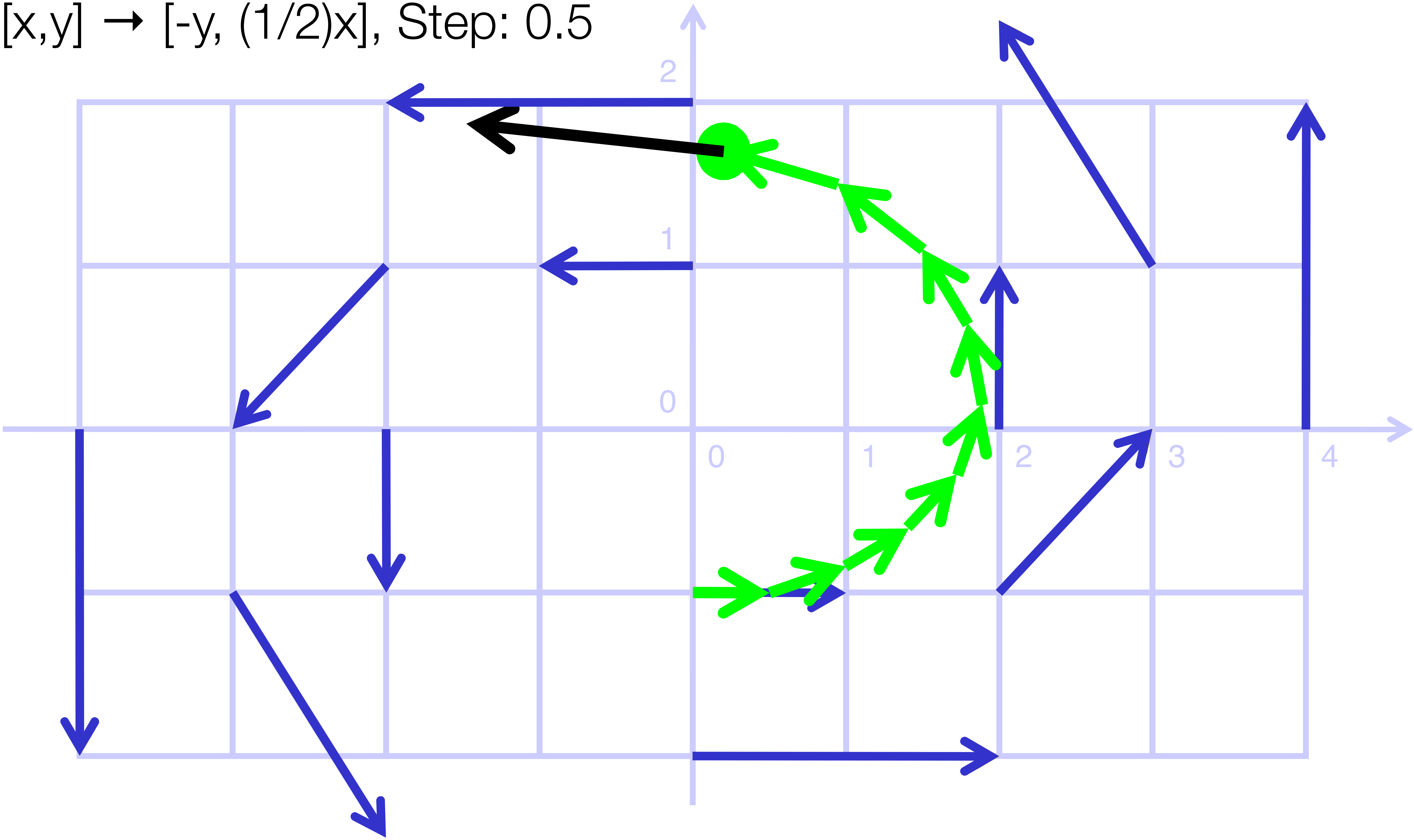
[x,y] → [-y, (1/2)x], Step: 0.5




[via Levine]

# Streamlines (Step 10)

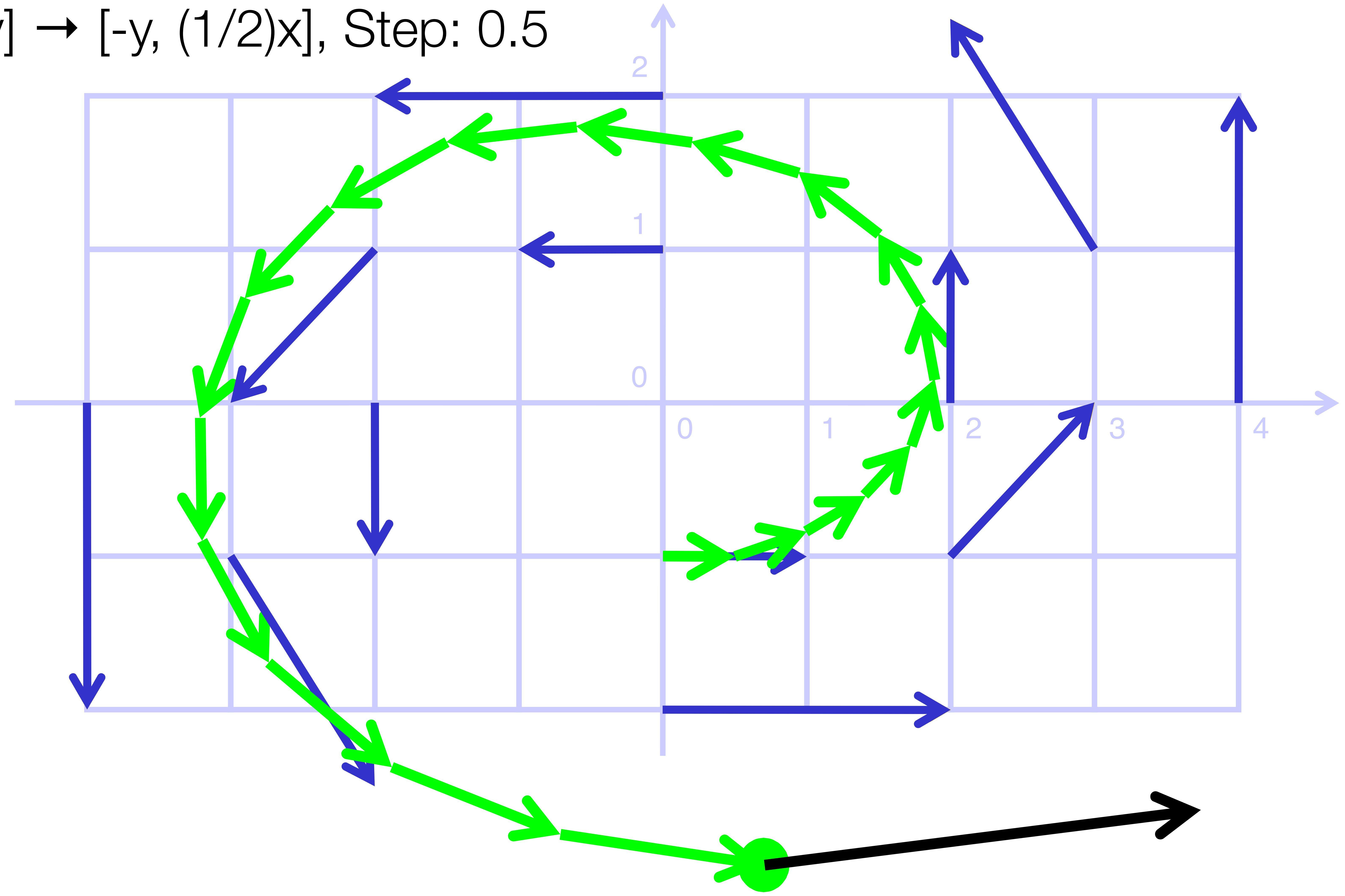
[x,y] → [-y, (1/2)x], Step: 0.5




[via Levine]

# Streamlines (Step 19)

[x,y] → [-y, (1/2)x], Step: 0.5




[via Levine]

# Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?

# Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?
  - Choice of step size is important

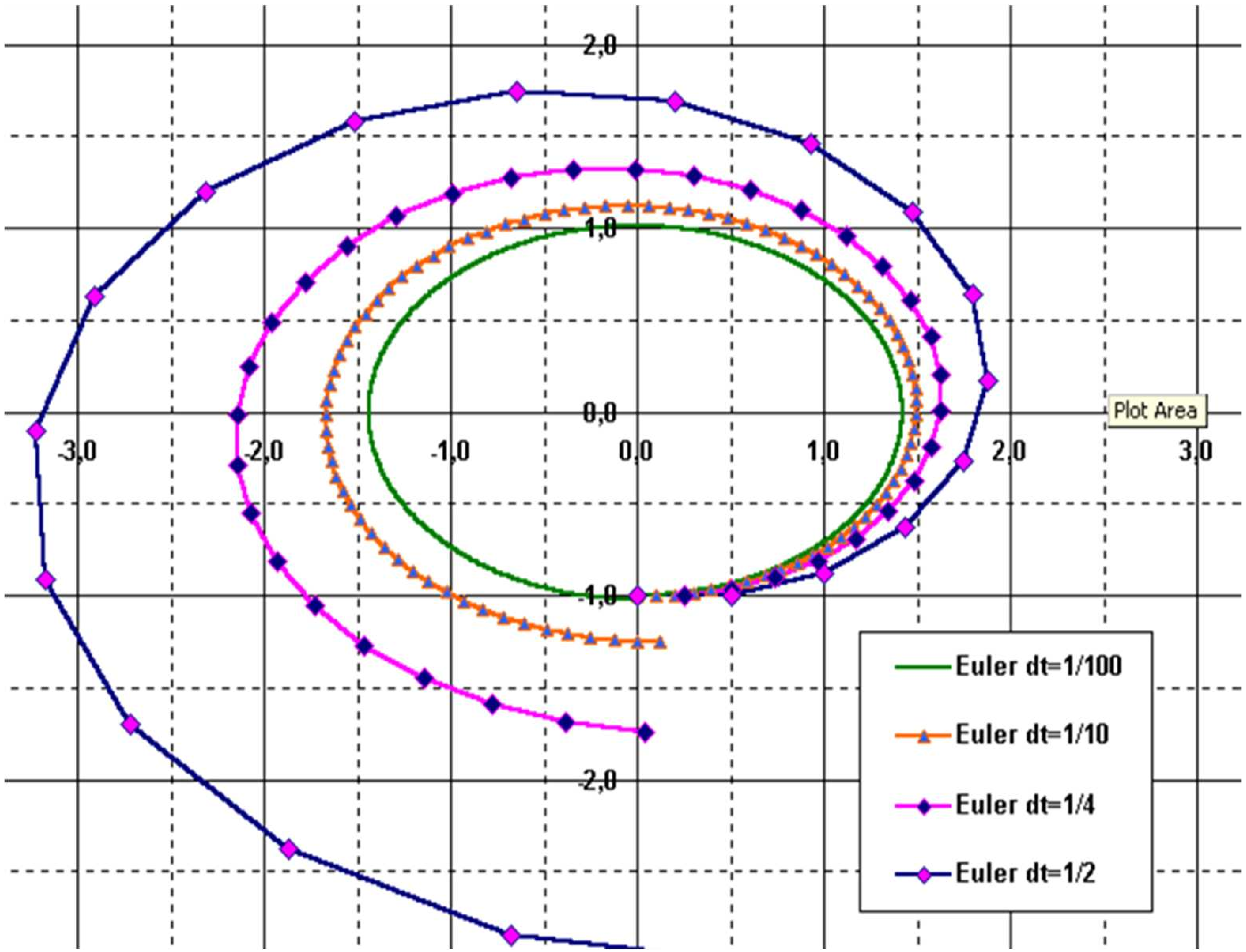
# Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?
  - Choice of step size is important
  - Choice of seed points are important

# Euler Method

- Seeking to approximate integration of the velocity over time
- Euler method is the starting point for approximating this
- Problems?
  - Choice of step size is important
  - Choice of seed points are important
- Also remember that we have a field—we don't have measurements at every point (interpolation)

# Euler Quality by Step Size




[via Levine]

# Numerical Integration

- How do we generate accurate streamlines?
- Solving an ordinary differential equation

$$\frac{dL}{dt} = v(L(t)) \qquad L(0) = L_0$$

where $L$ is the streamline, $v$ is the vector field, and $t$ is “time”

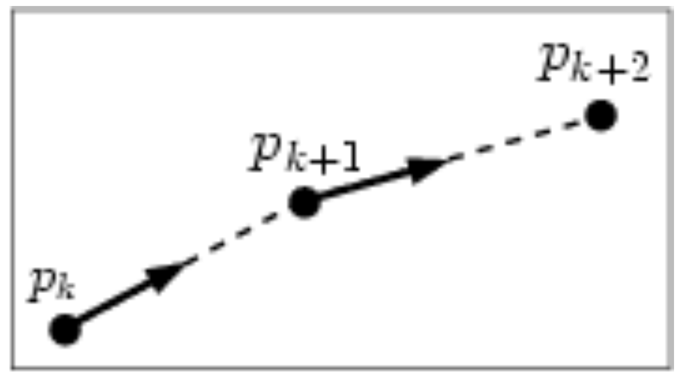
- Solution:

$$L(t + \Delta t) = L(t) + \boxed{\int_t^{t+\Delta t} v(L(t))dt}$$

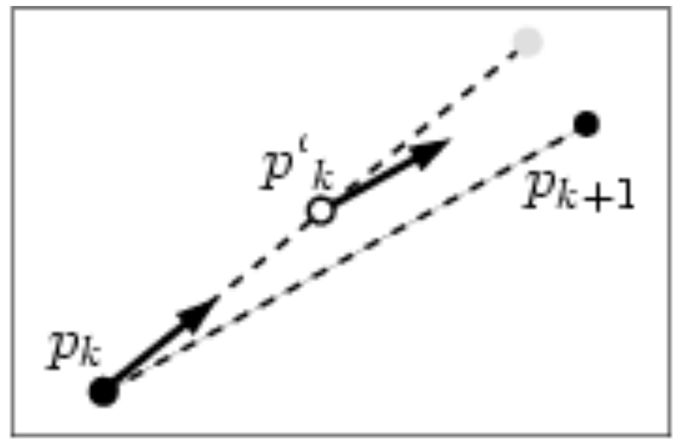
# Higher-order methods

$$\int_t^{t+\Delta t} v(L(t))dt$$

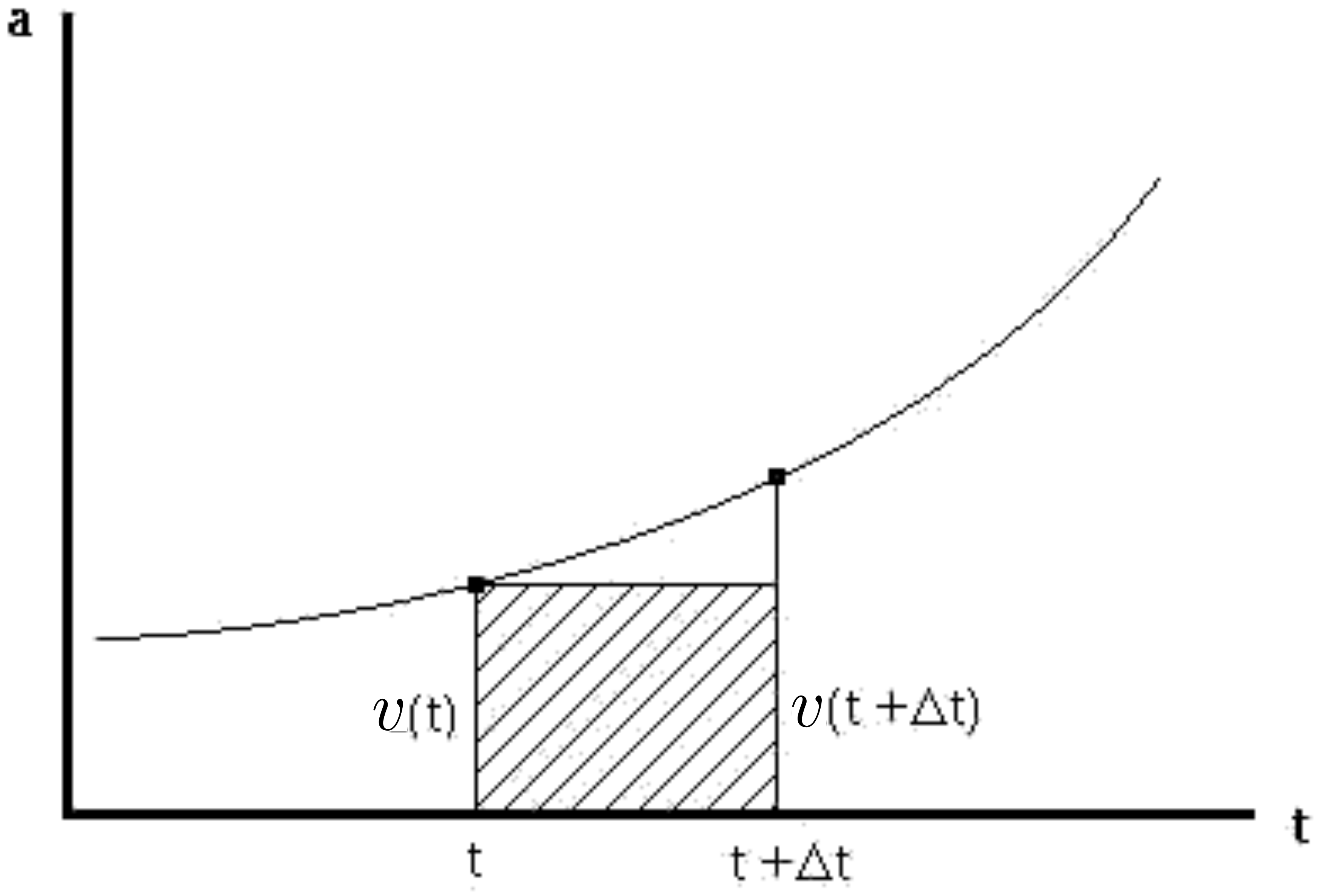
- Euler method (use single sample)



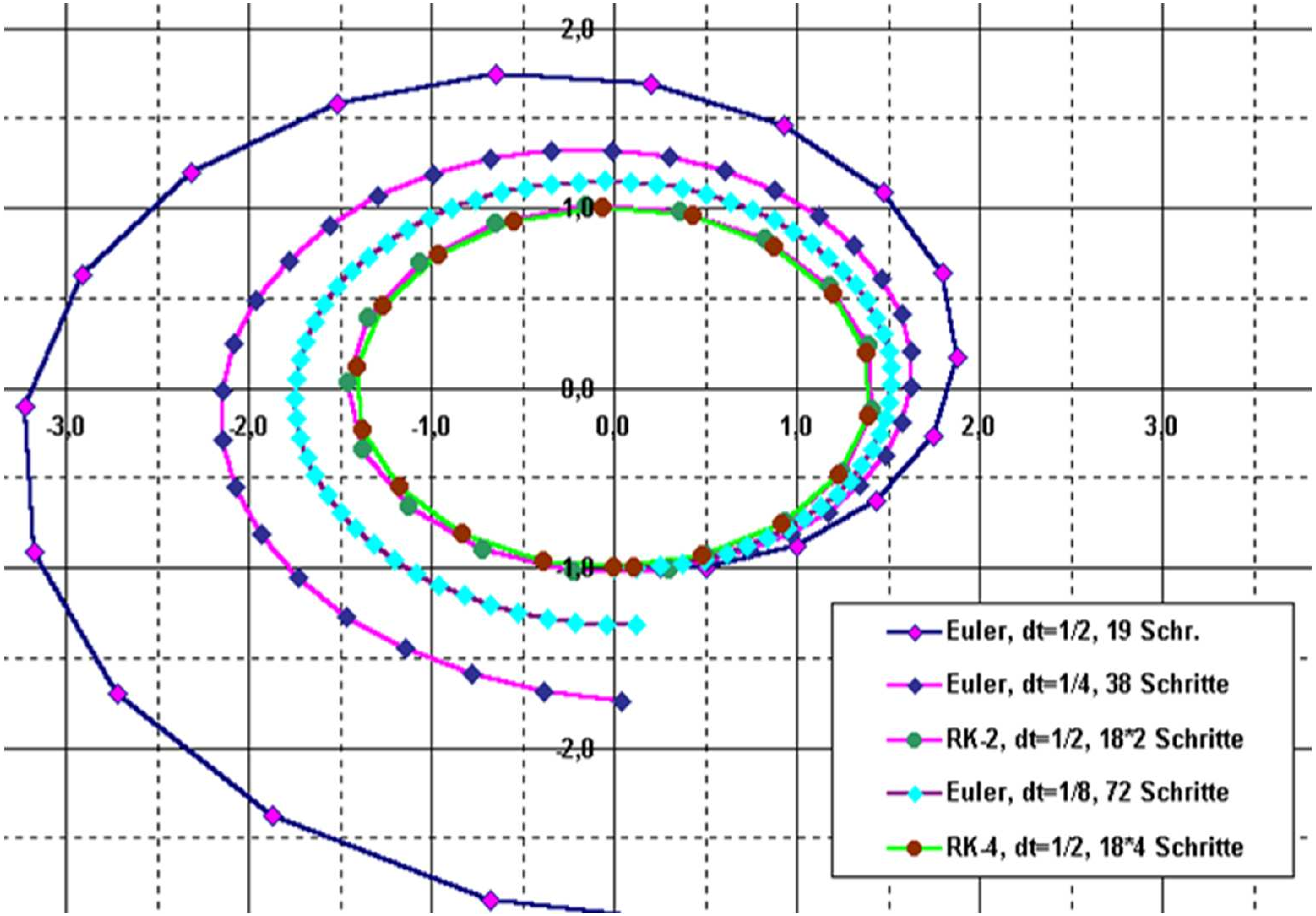

- Higher-order methods (Runge-Kutta) (use more samples)




[A. Mebarki]



# Higher-Order Comparison

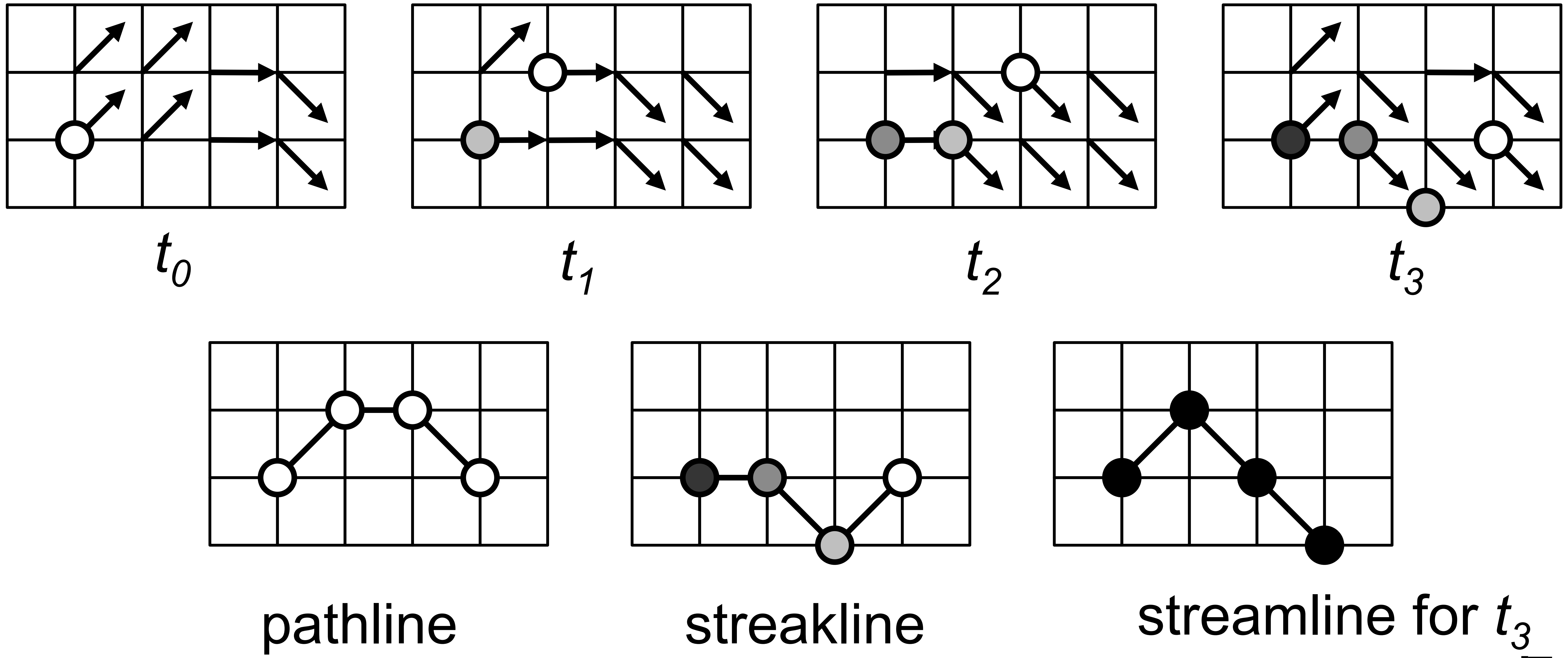



[via Levine]

# ParaView Examples

# Streamlines & Variants

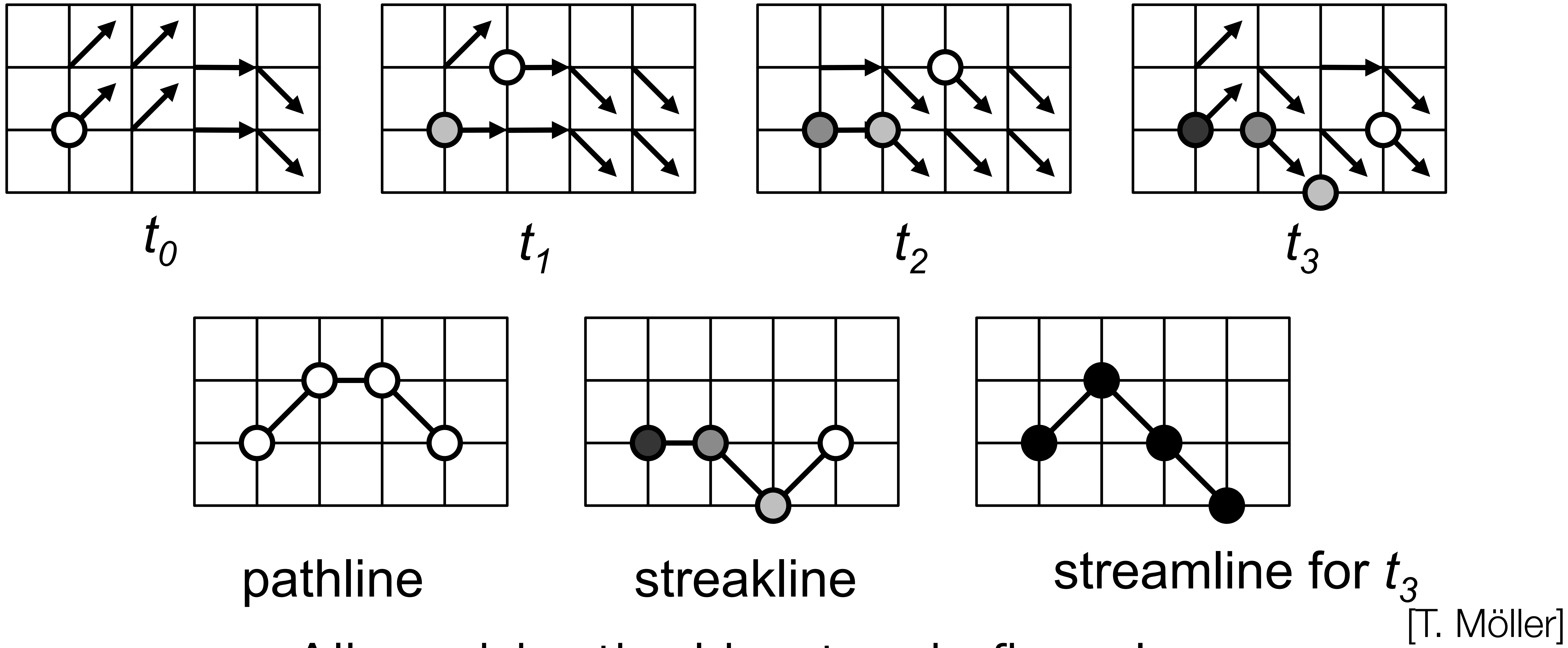
- Steady vs. **Unsteady** flows
  - In unsteady flows, the vector field **changes** over time
- Variants: **Pathlines** and **Streaklines**




[T. Möller]

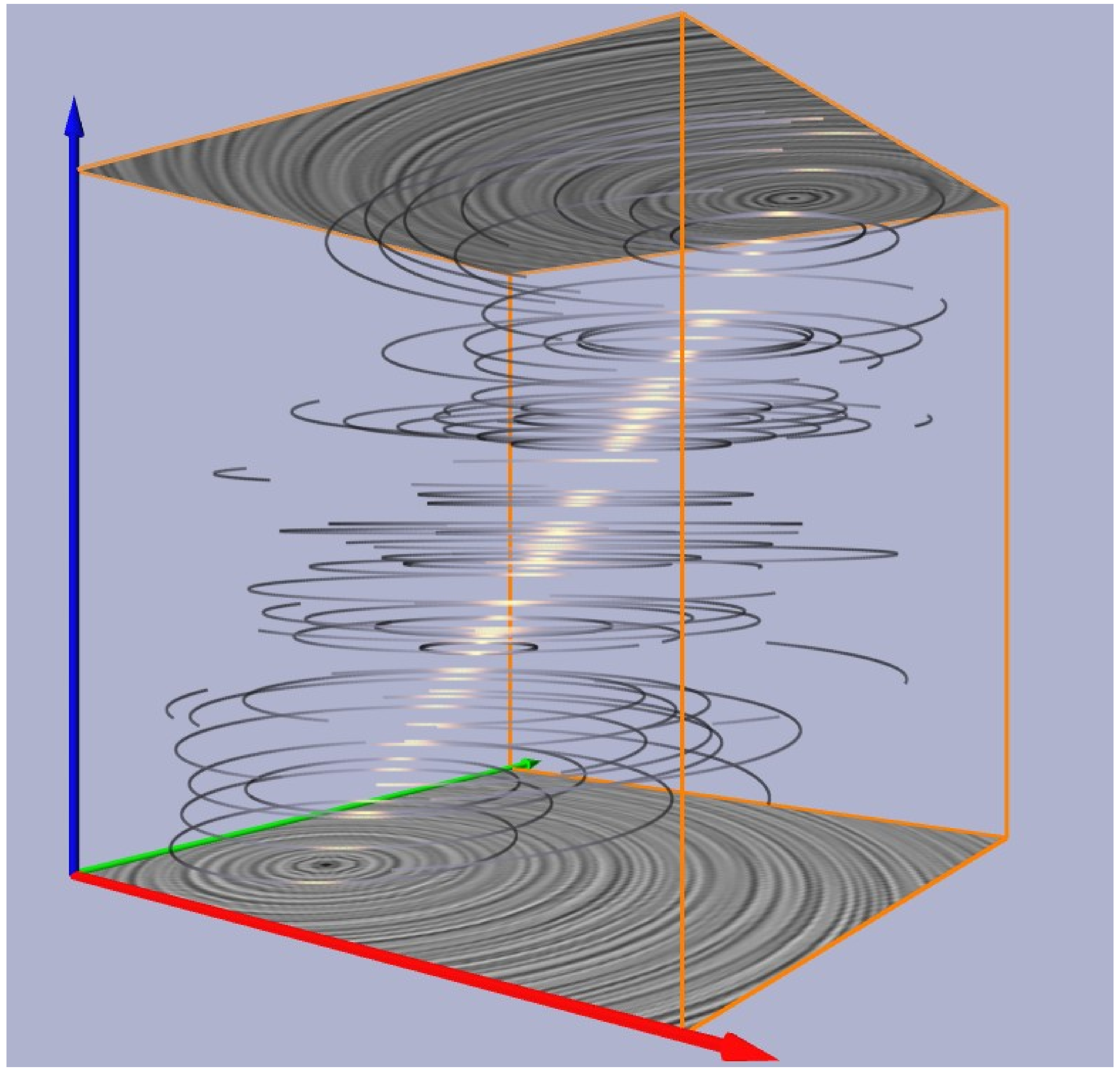
# Streamlines & Variants

- Steady vs. **Unsteady** flows
  - In unsteady flows, the vector field **changes** over time
- Variants: **Pathlines** and **Streaklines**

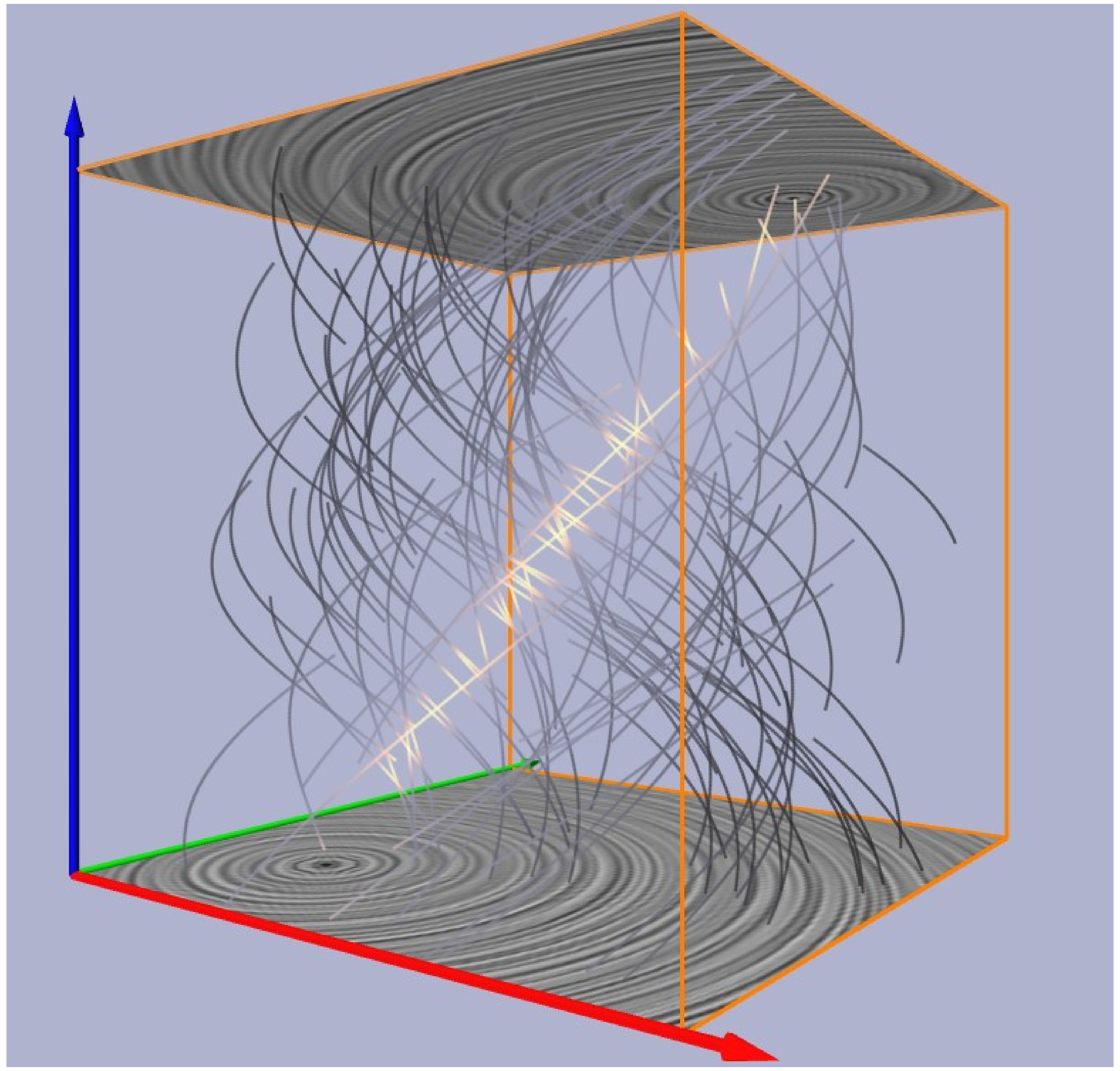



All are identical in steady flows!

# Streamlines vs. Pathlines



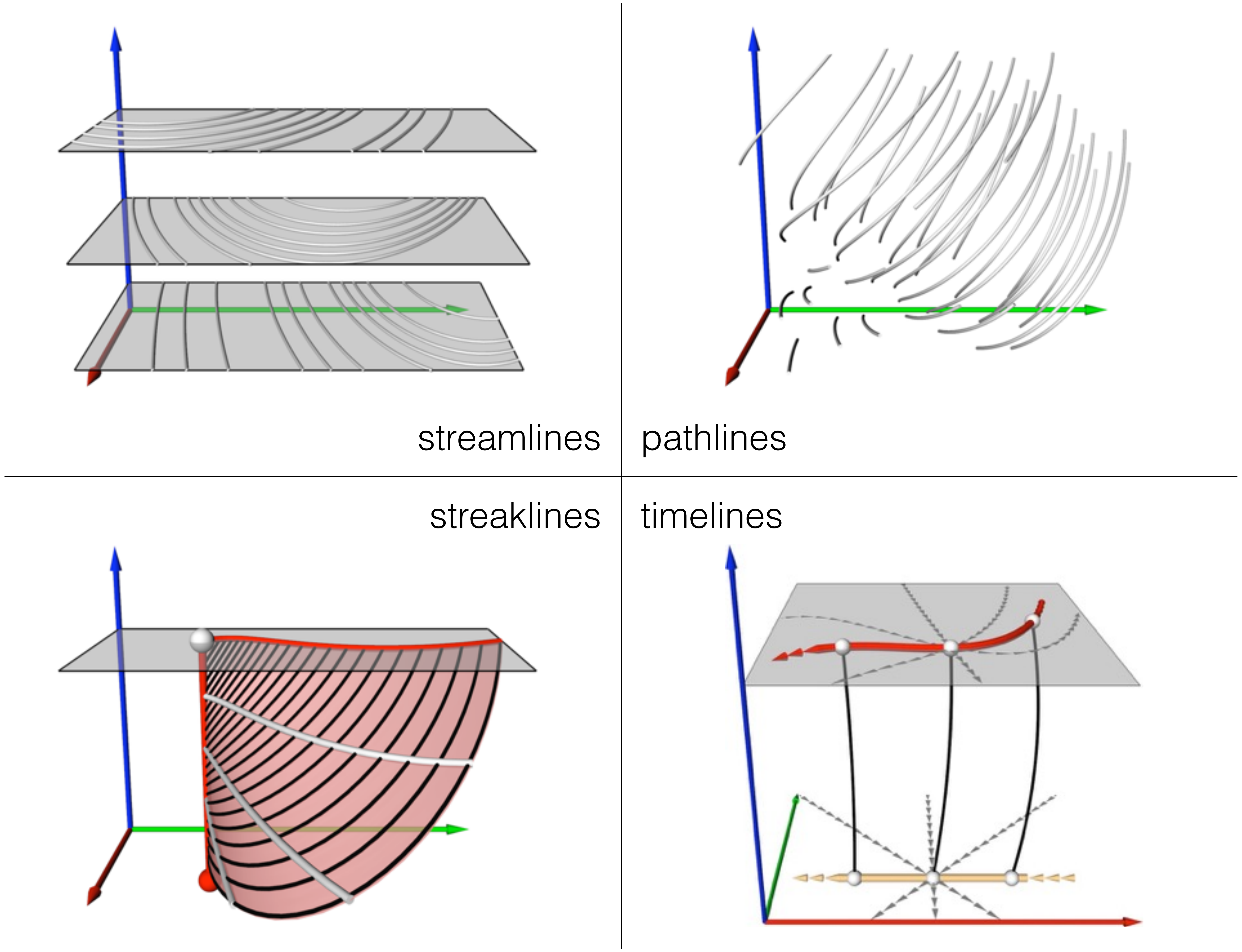
Streamlines



Pathlines

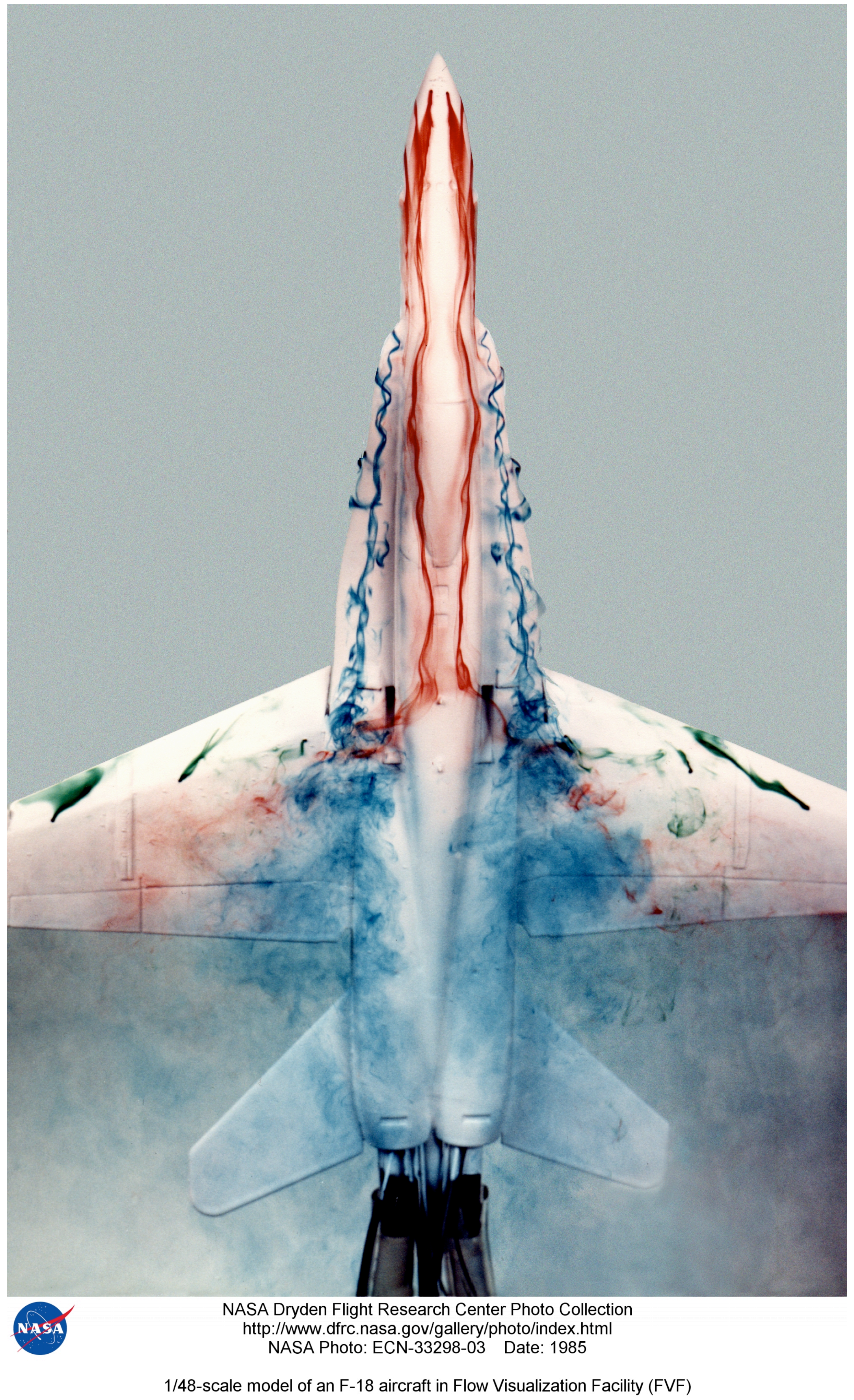
[Weinkauf & Theisel, 2010]

# Streaklines and timelines

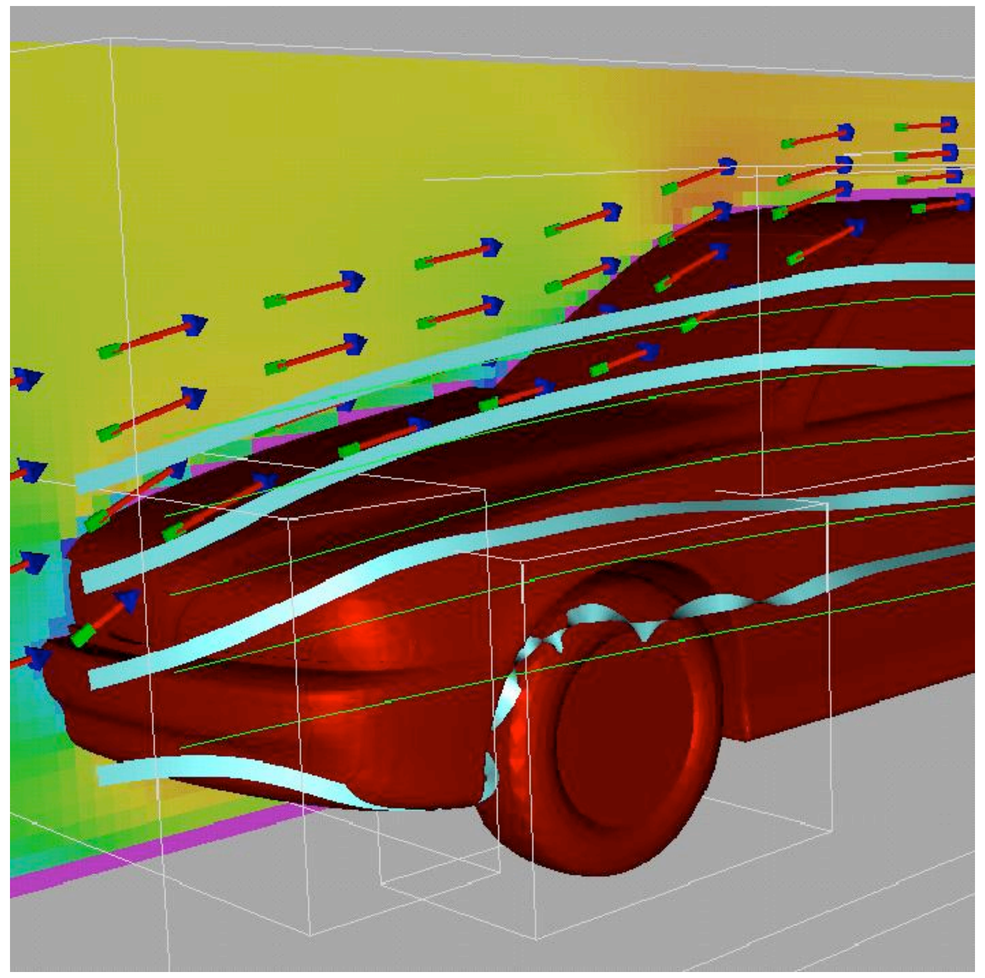



[via Levine]

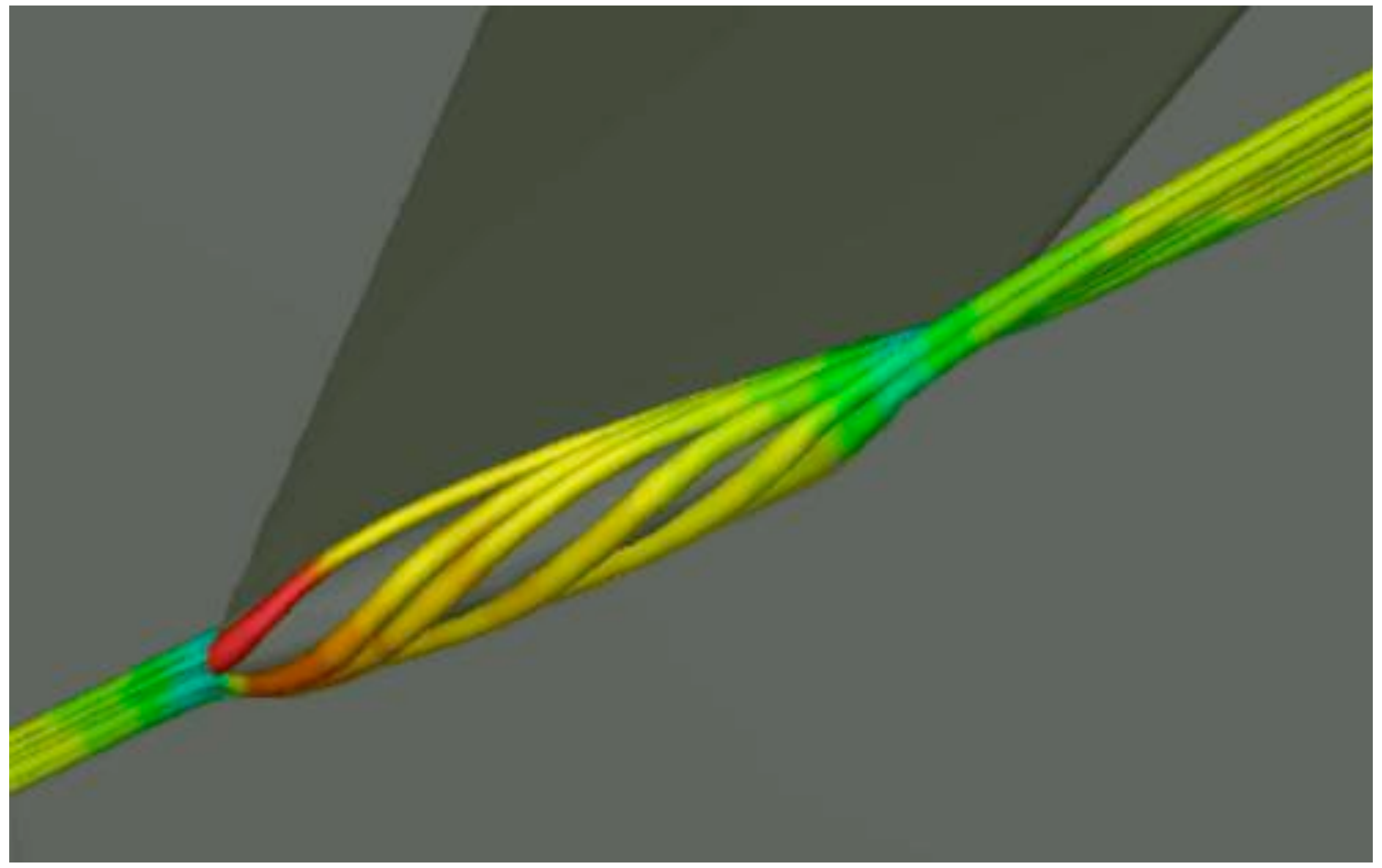
# Streamline Variants




Streaklines [NASA]

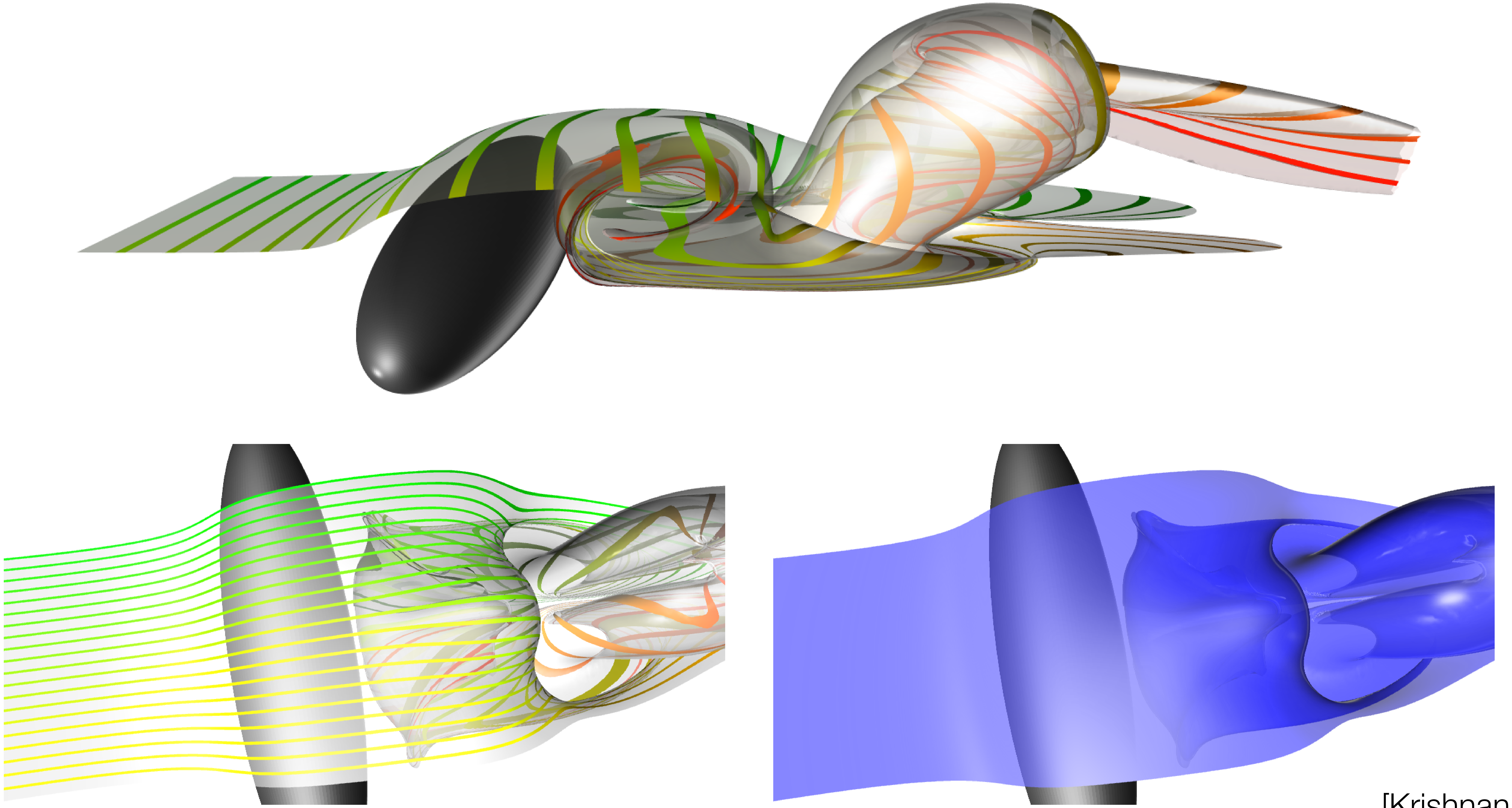


Stream Ribbons [Weiskopf/Machiraju/Möller]



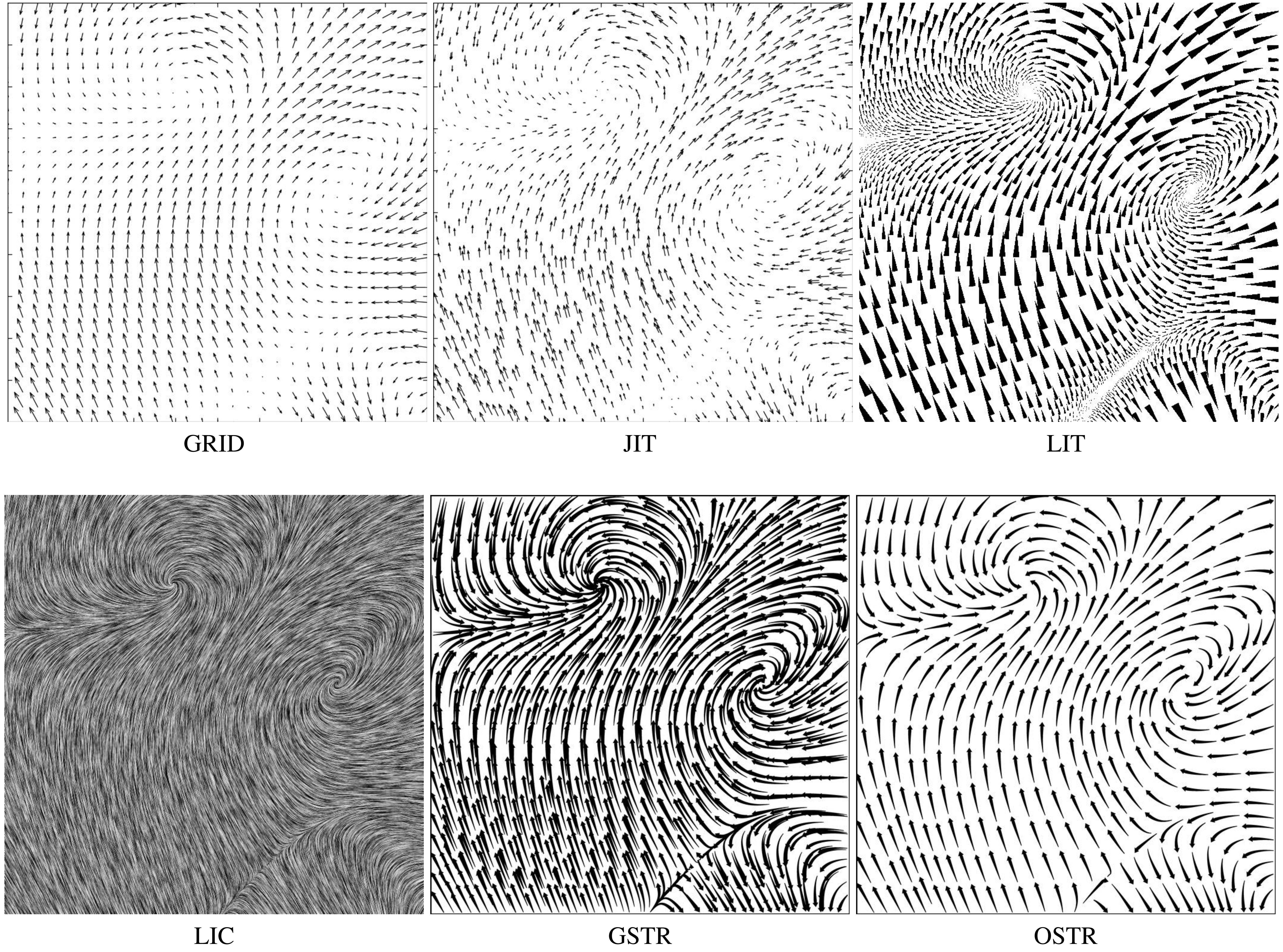
Stream Tubes [Weiskopf/Machiraju/Möller]

# Streak Surfaces



[Krishnan et al., 2009]

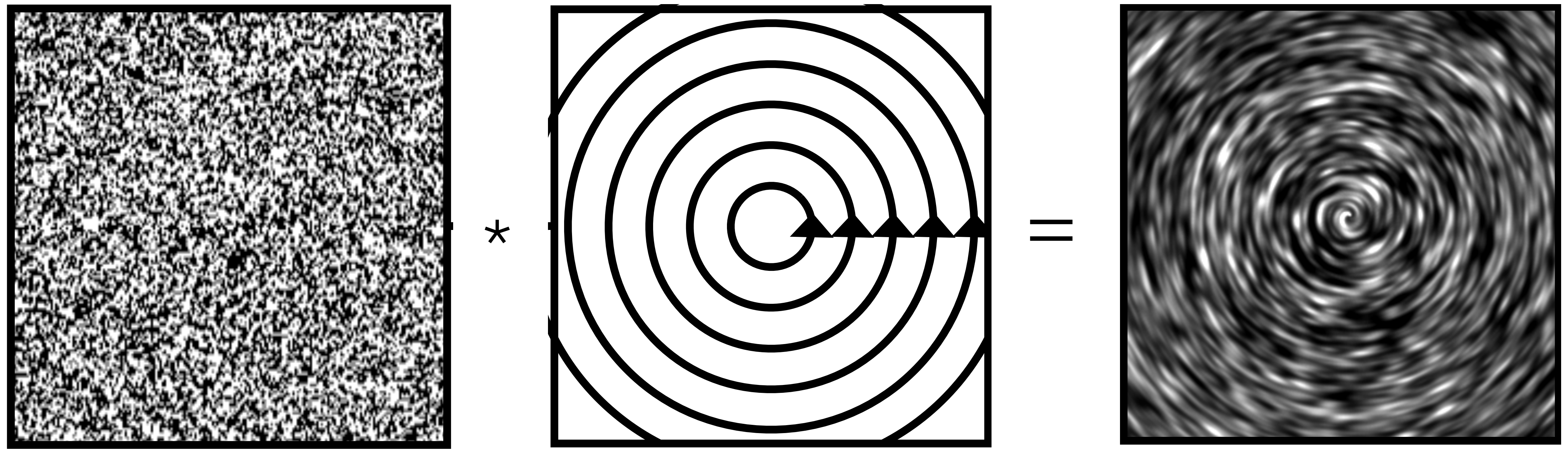
# 2D Vector Field Visualization Techniques




[Laidlaw et al., 2005]

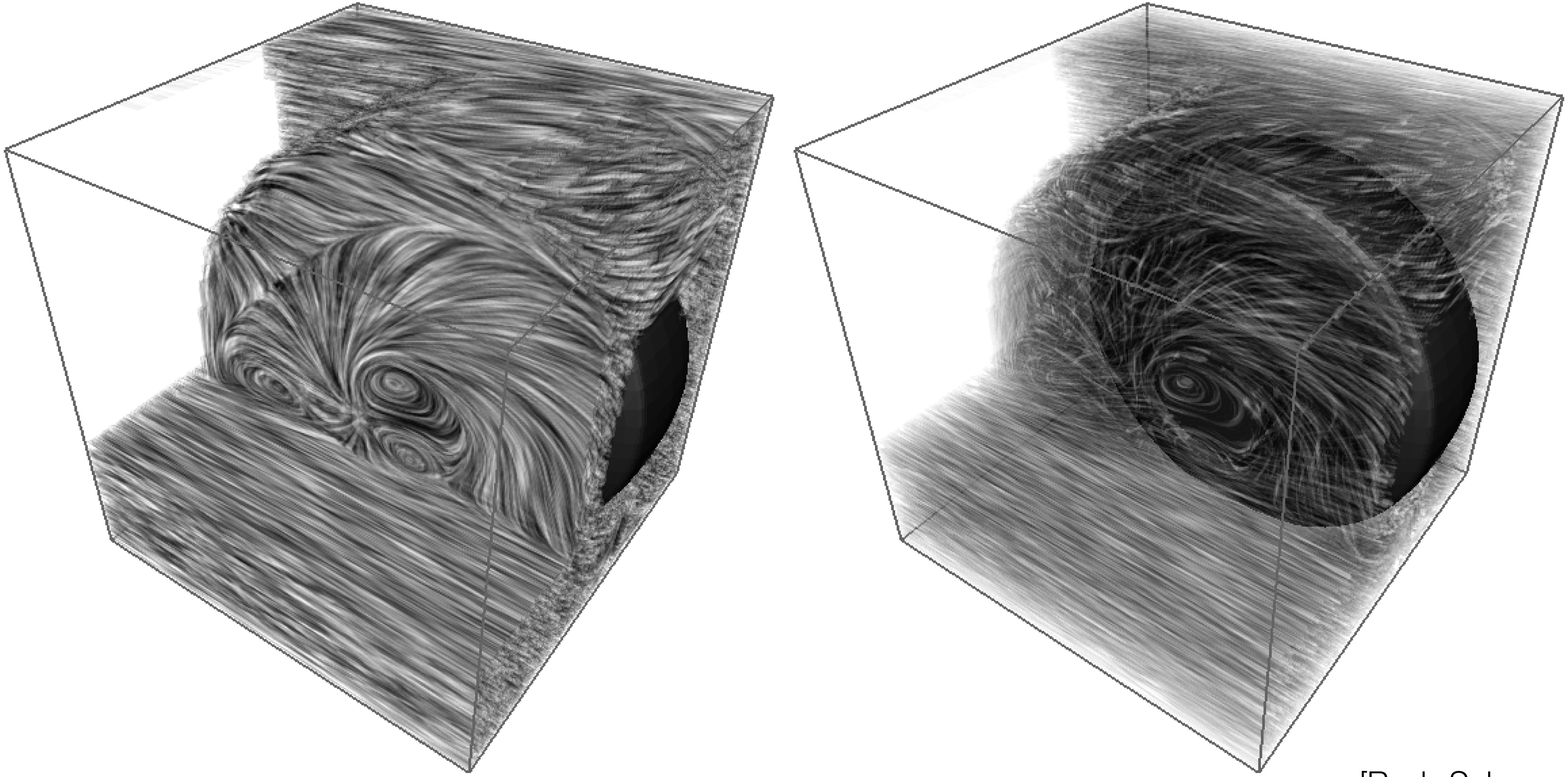
# Line Integral Convolution

- Goal: provide a global view of a steady vector field while avoiding issues with clutter, seeds, etc.
- Remember convolution?
- Start with random noise texture
- Smear according to the vector field
- Need structured data




[Weiskopf/Machiraju/Möller]

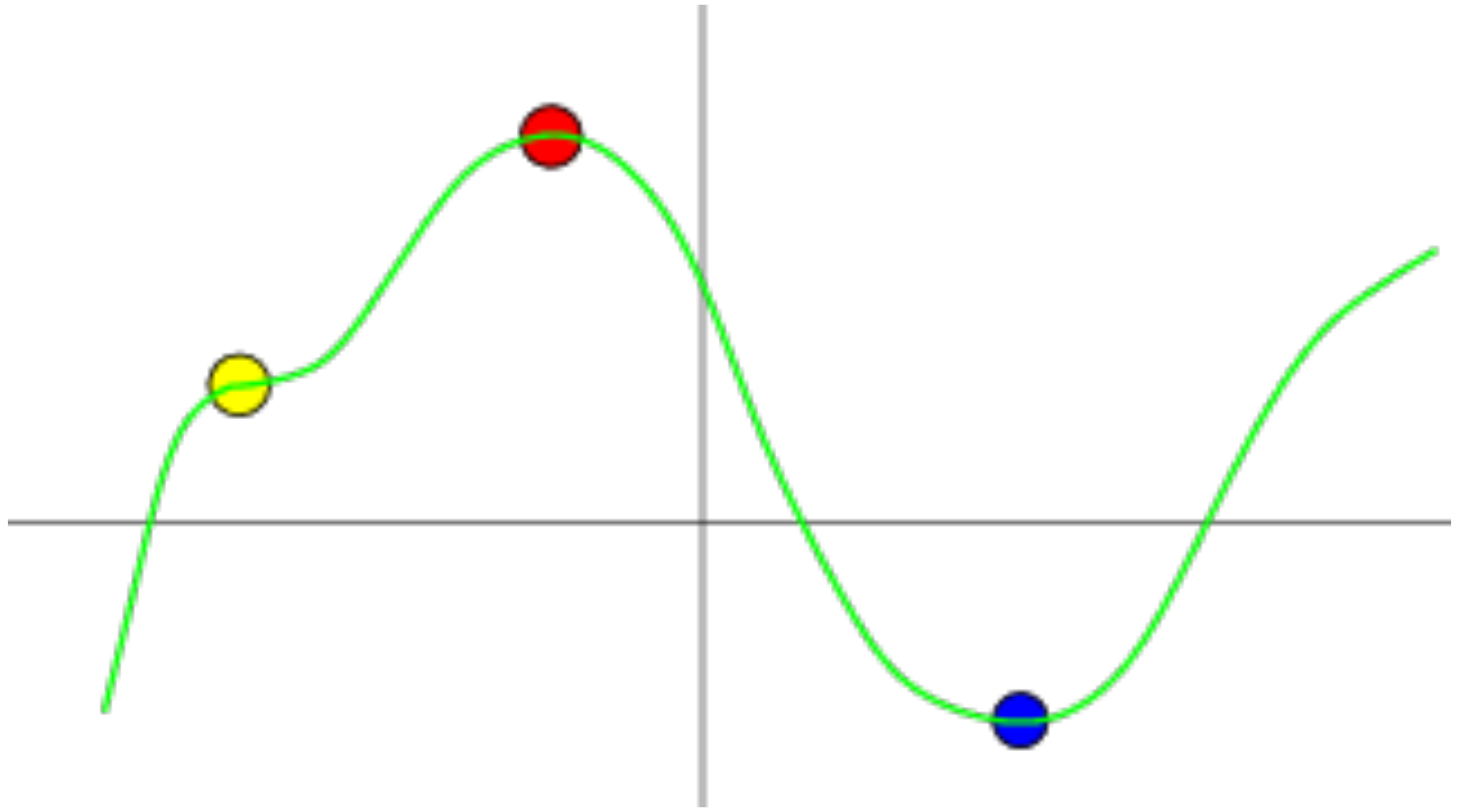
# 3D LIC



[Rezk-Salama et al., 1999]

# Critical Points

- Remember finding min/max for functions?
- Want to understand the general structure of a field, not the exact values
- Find critical points, understand there is a general trend in between
- How?
  - Derivative for functions
  - For fields…gradients

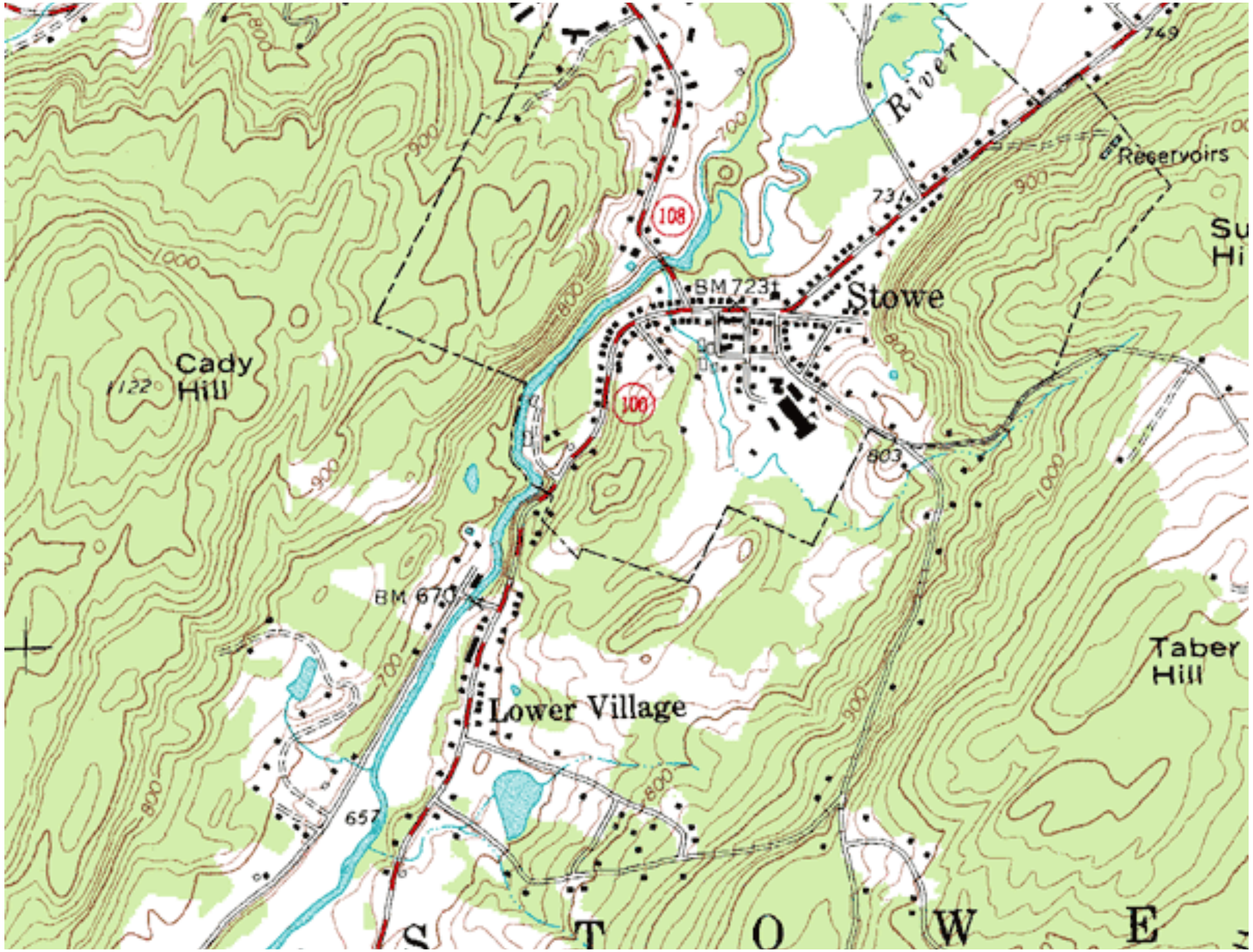


[DQ Nykamp, MathInsight]

# Topology

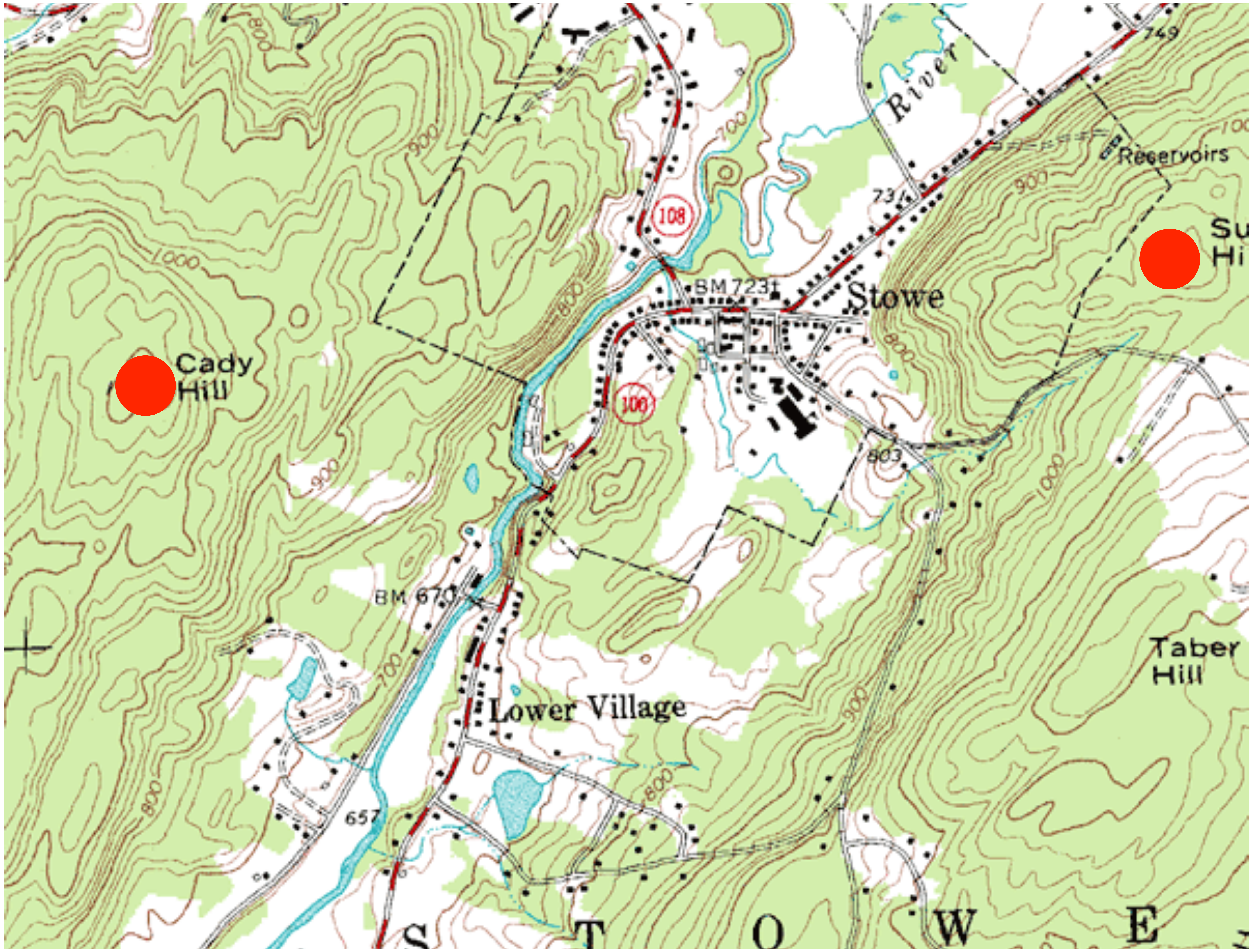
- The general shape of data
- Visualizations that can be "stretched" to resemble each other are topologically equivalent
- Technically, continuous transformations don't change anything
- Connect critical points to obtain a general picture of the data
- Can talk about topology in both scalar and vector fields

# 2D Scalar Field Topology




[Wikipedia]

# 2D Scalar Field Topology

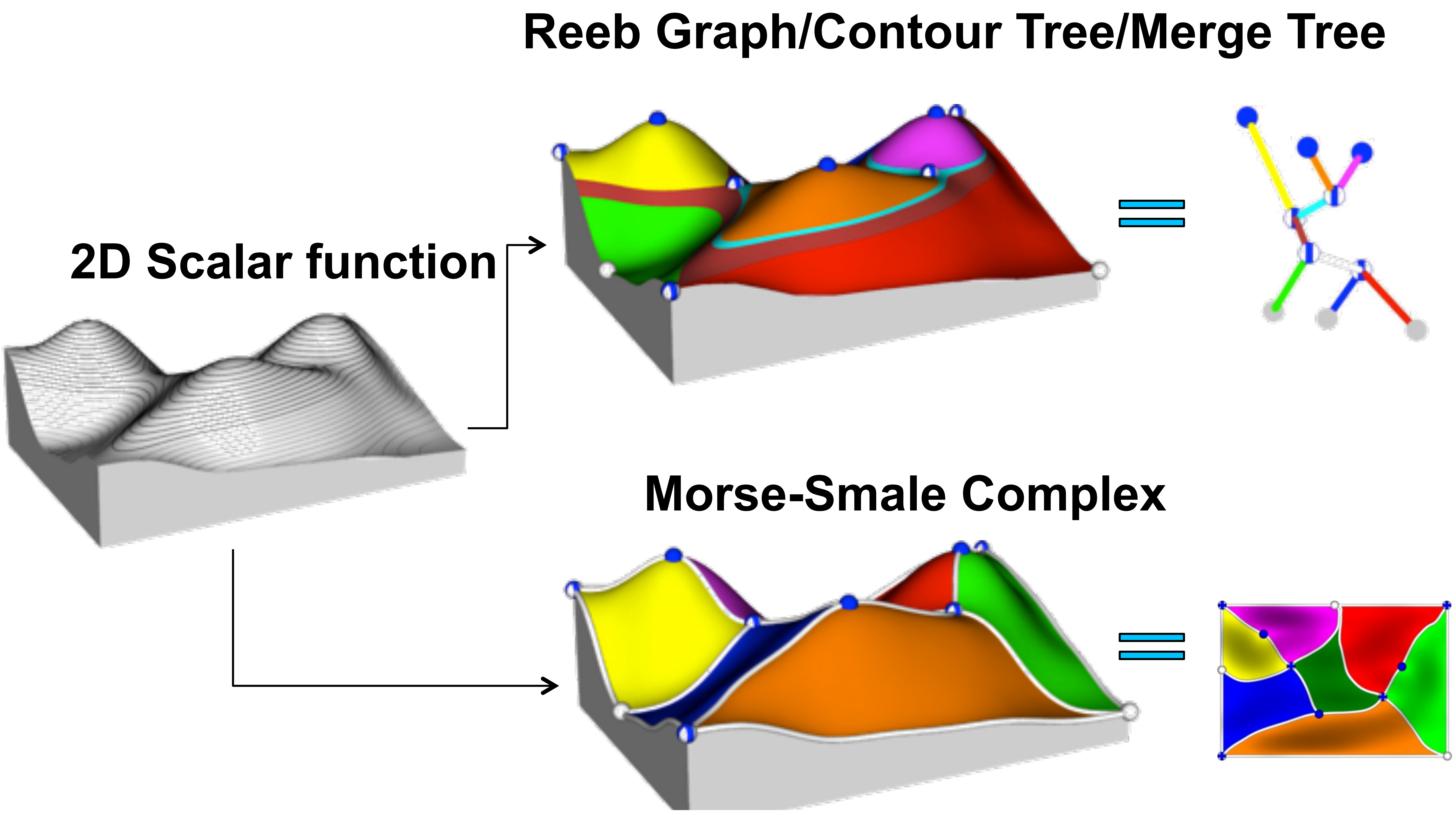



[Wikipedia]

# Scalar Field Topology

- Examine the gradient (changes between points on the grid) of the scalar field
- Where the gradient is zero, we have critical points (max, min, saddle)
- Can build Reeb Graph, Contour Tree, or Morse-Smale Complex from this information to show the topology (with some reasonable assumptions about how the scalar field looks)

# Scalar Field Topology




[via Levine]

# Vector Field Topology

- Instead of “guessing” correct seed points for streamlines to understand the field, try to identify structure (topology) of the field

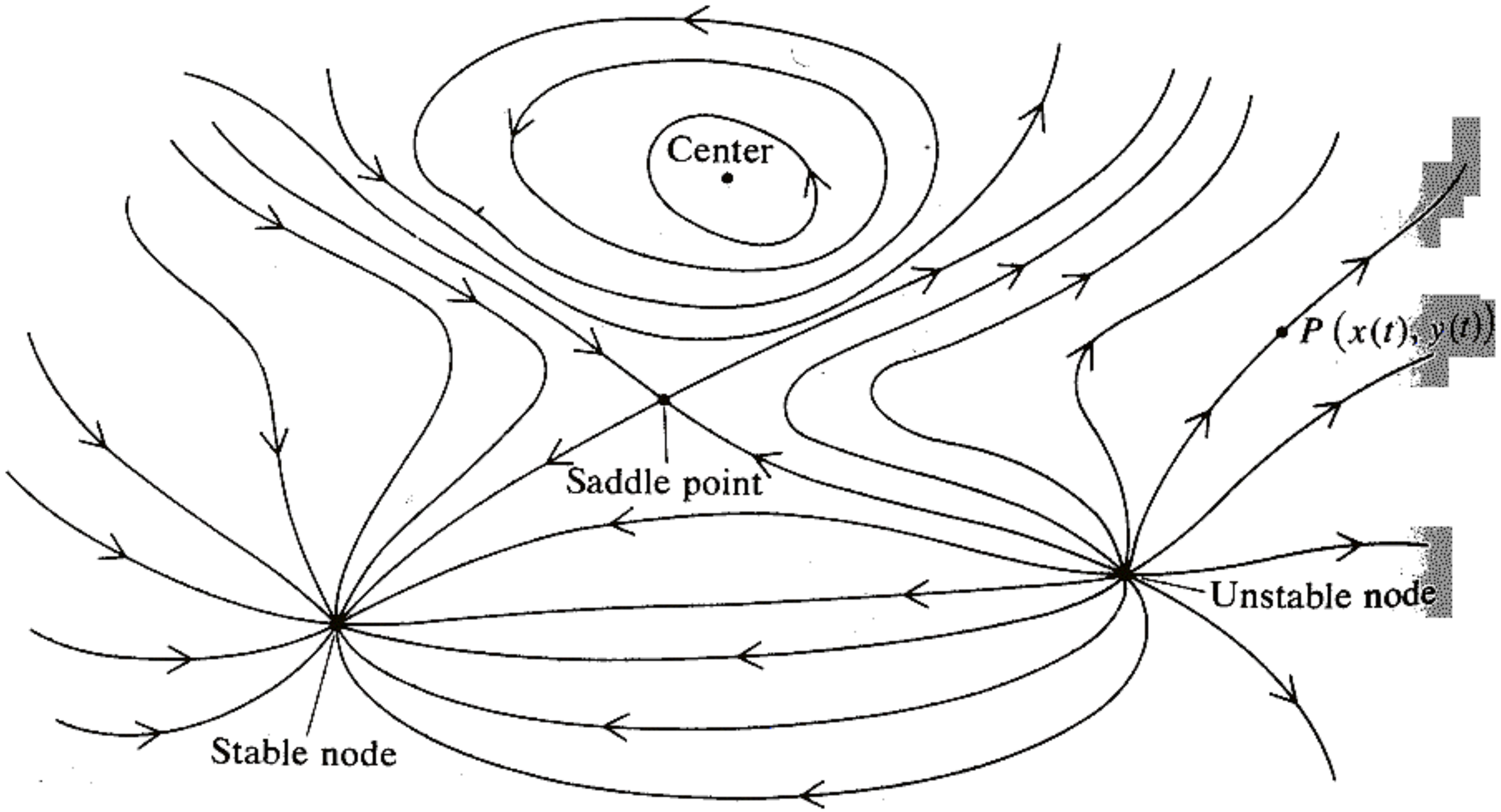
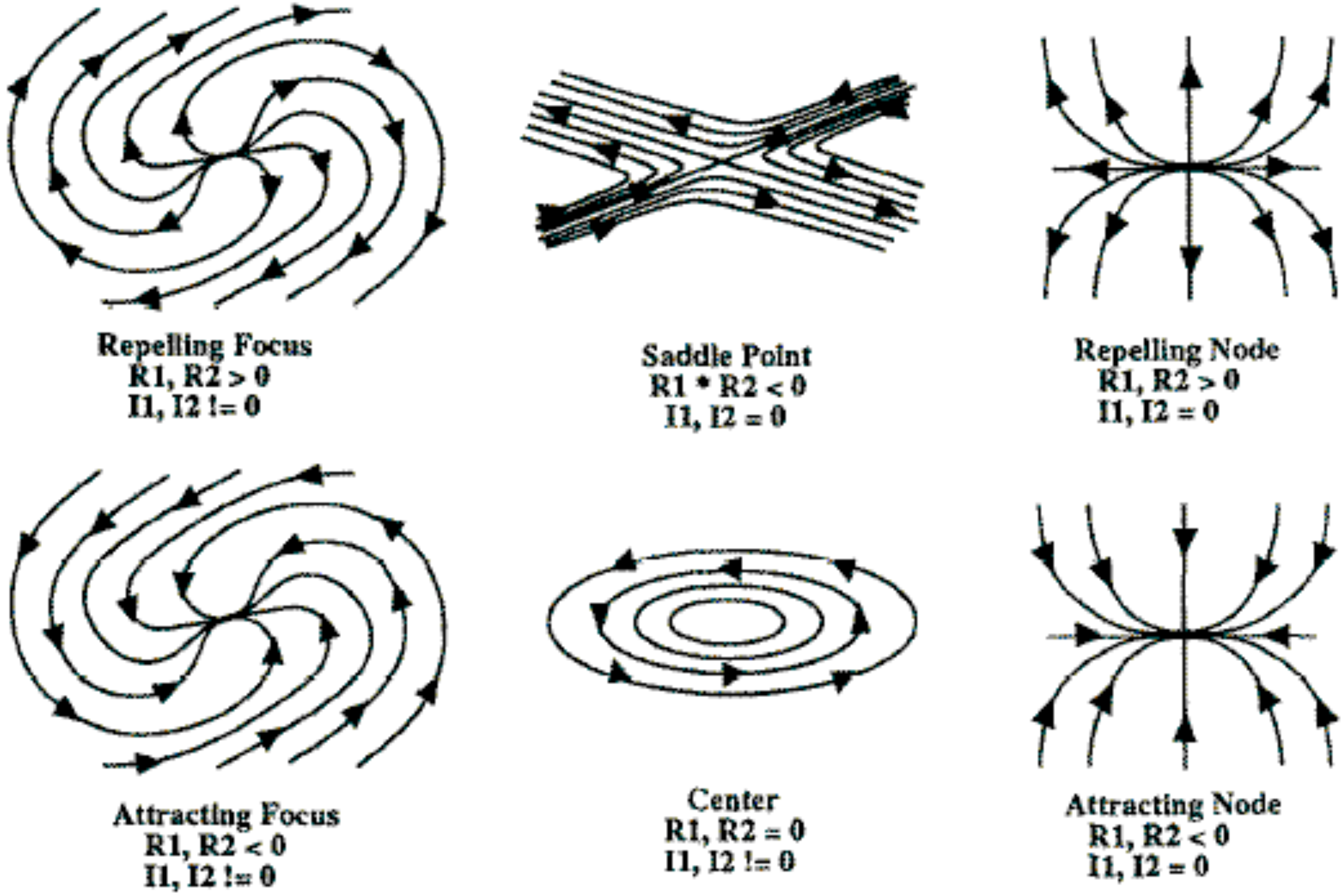



***Figure 7.1*** A phase portrait.

[M. Henle]

# Critical Points

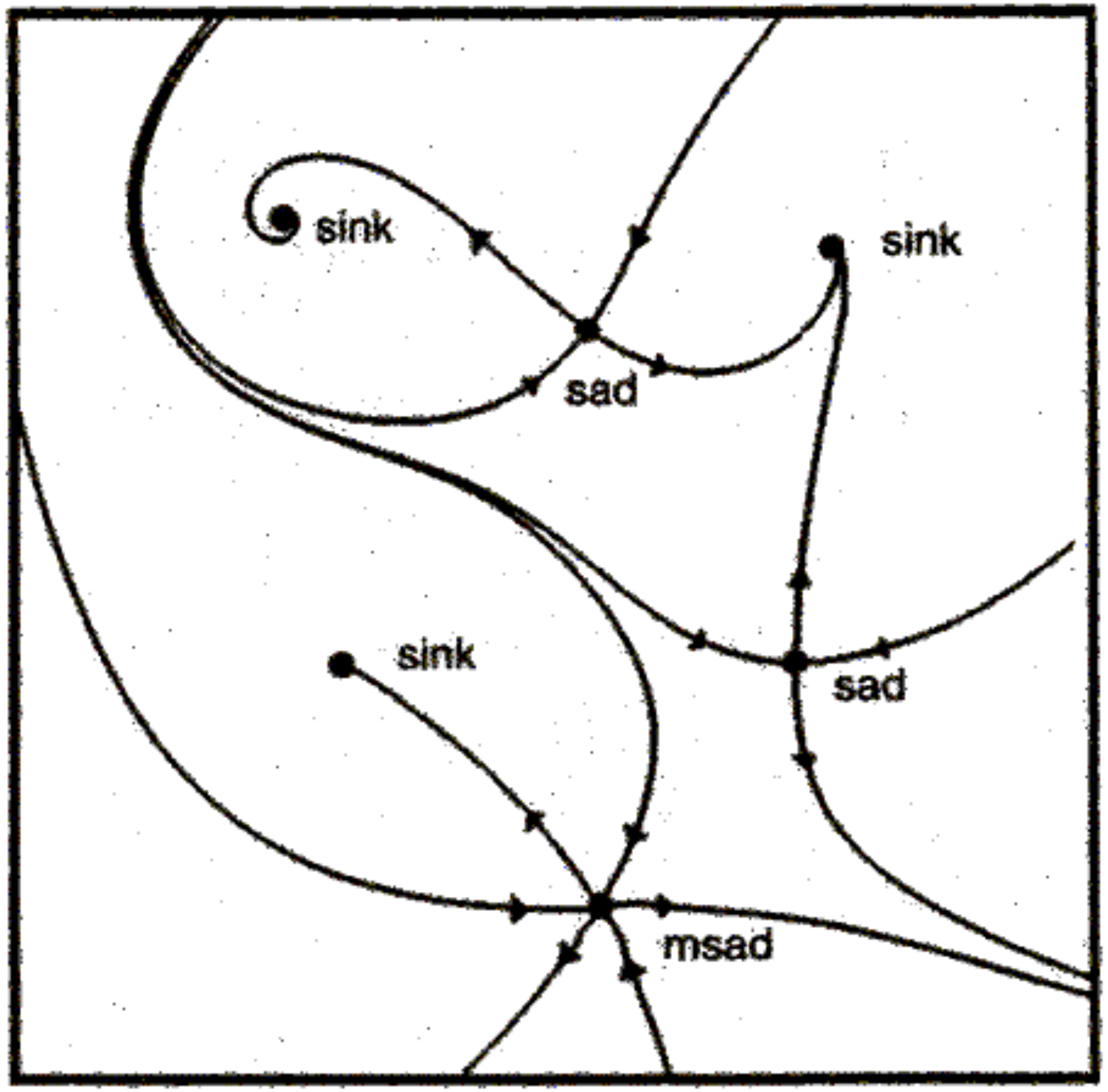


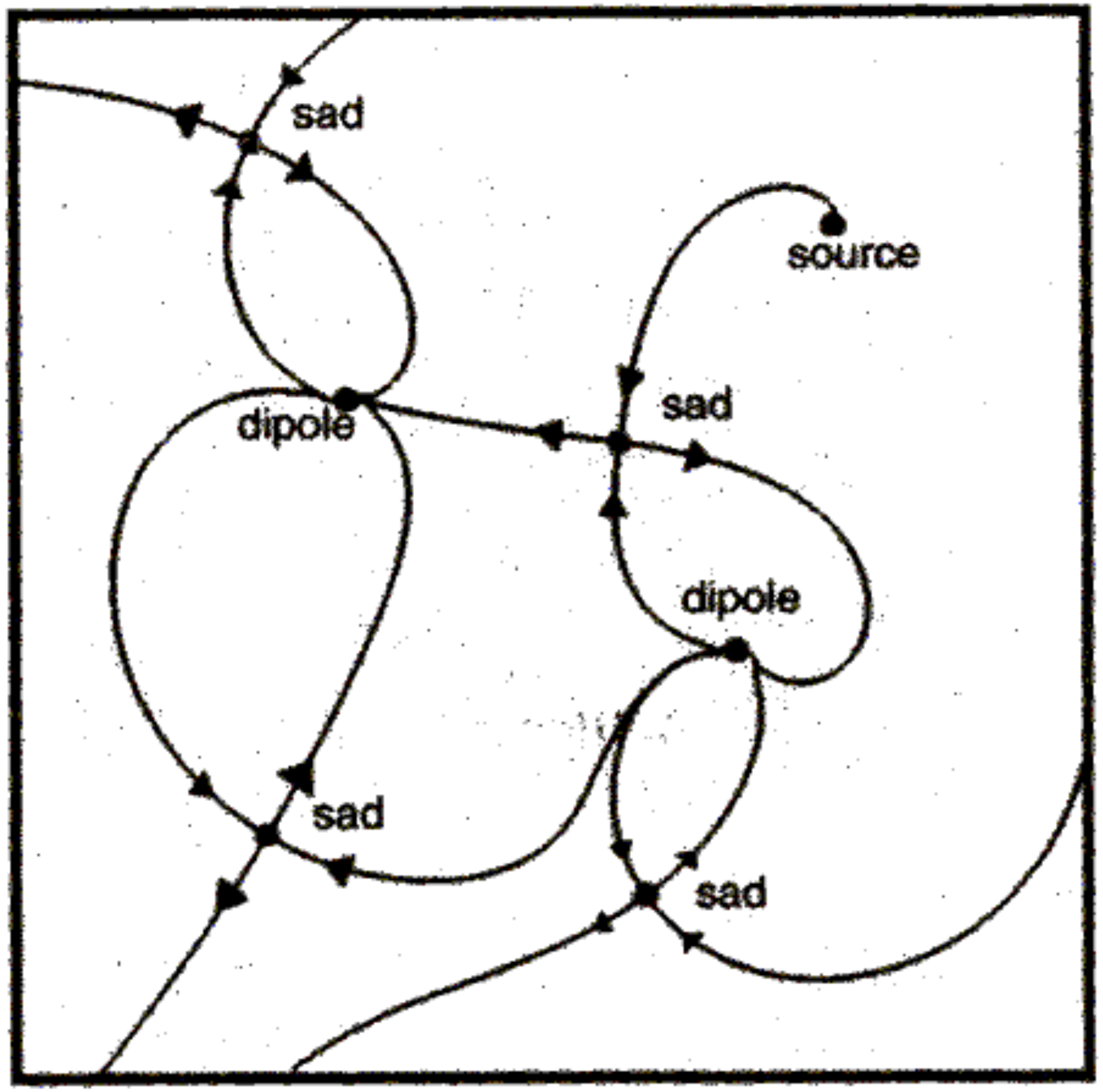

[Helman & Hesselink]

# Critical Points

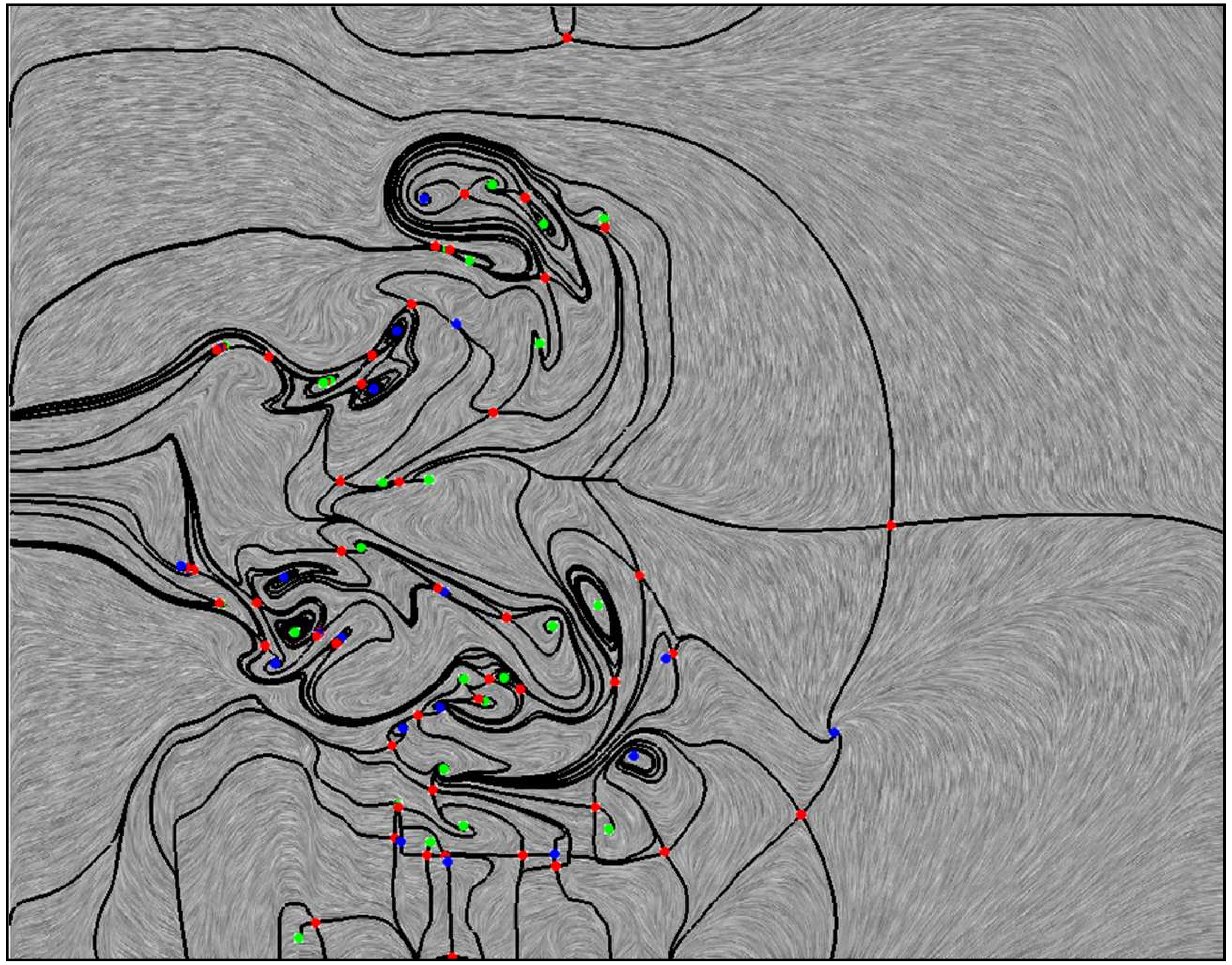
- Critical Points
  - Find where the vector field vanishes (the zero vector or undefined)
  - Attracting Nodes (Sinks), Repelling Nodes (Sources), Attracting Foci, Repelling Foci, Saddles, Centers
- How to find such points?
  - Can use a similar idea to Marching Cubes
  - Use the eigenvalues of the Jacobian matrix to classify

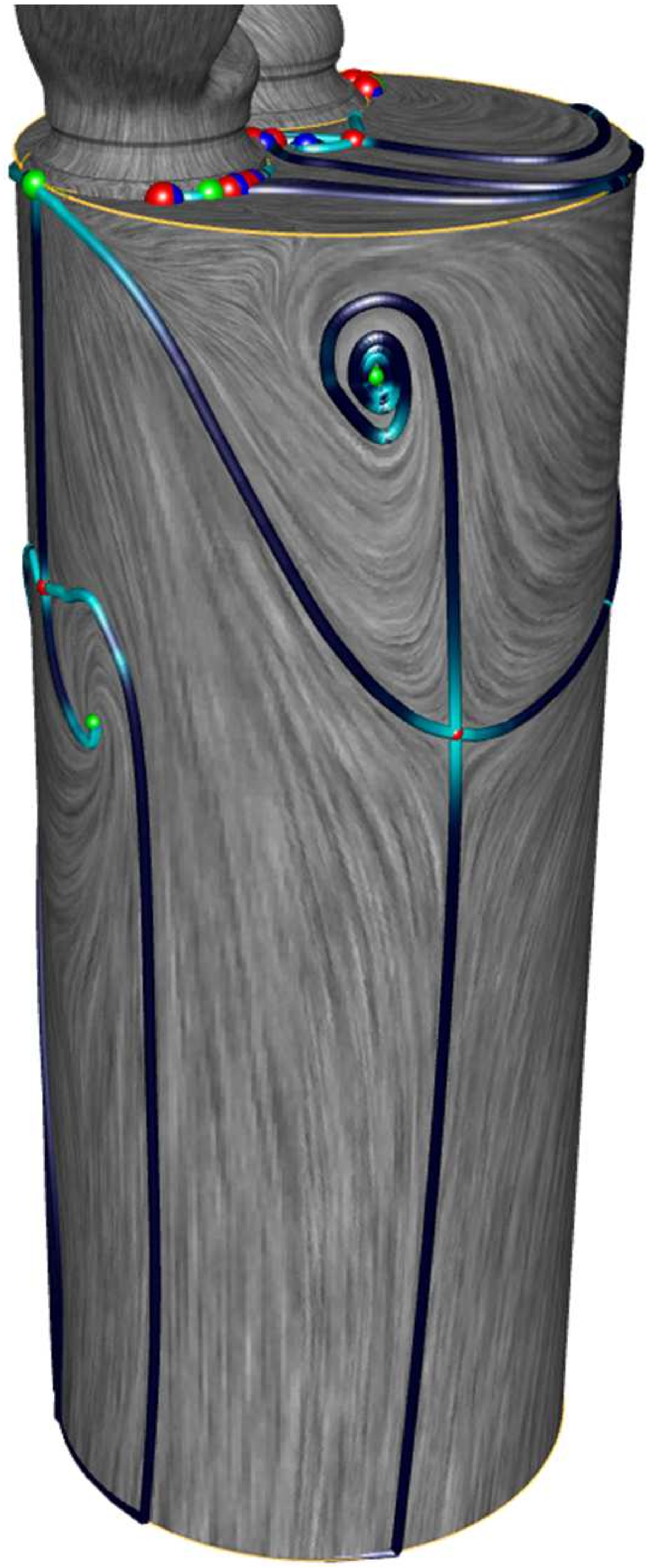
# Topological Skeleton



# More Examples





[Levine]