Representations and Applications of Tangential Vector Fields

Mirela Ben-Chen and Omri Azencot

Technion, Israel Institute of Technology

Funded by the European Union

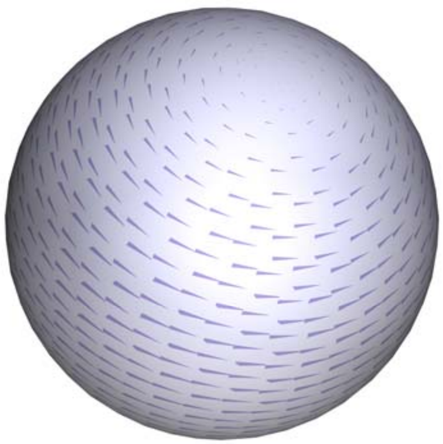
Vector Fields in the Wild

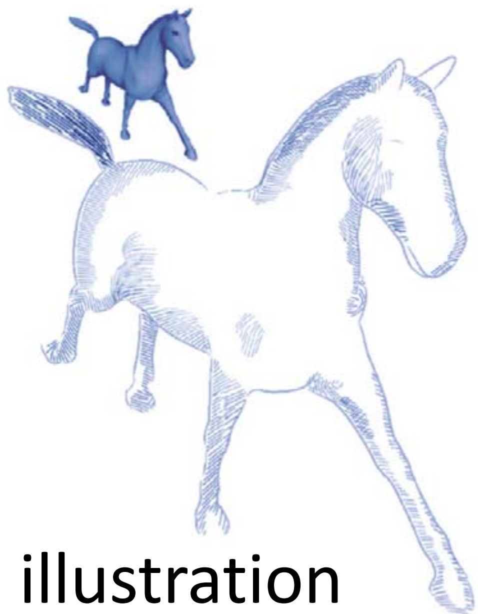
http://satview.bom.gov.au/

Vector Fields in the Wild

http://www.thetimes.co.uk/article/four-die-as-storms-lash-australia-s-east-coast-tx789nljt
http://www.taipeitimes.com/News/front/photo/2016/06/07/2008134158

Tangential Vector Fields



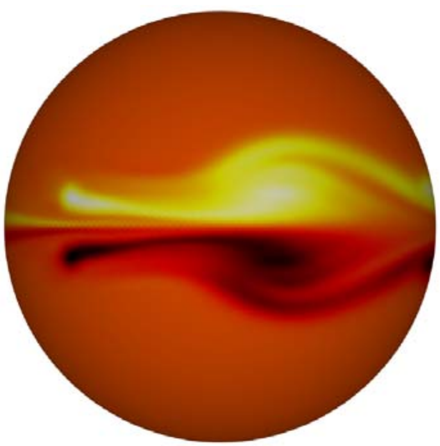


illustration

[Herzman et al.,2012]

visualization

[Diewald et al., 2000]

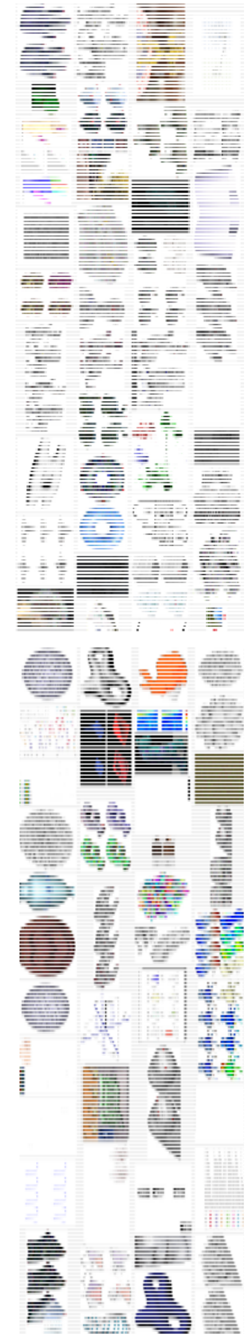


simulation

[Azencot et al., 2014]

design

[Fisher et al., 2007]



...

Outline

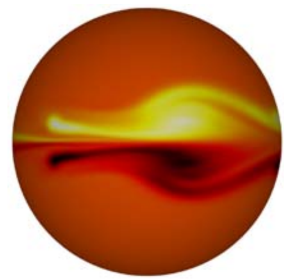
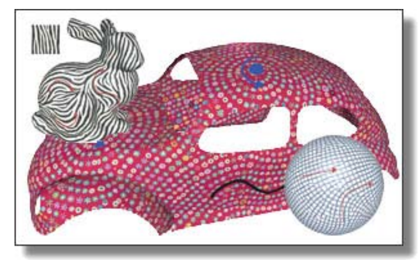
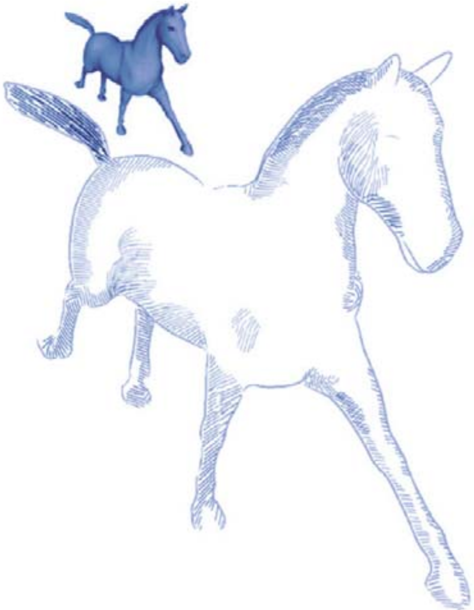
- Intro (done!)
-    

  illustration (miri) | visualization (omri) | simulation (omri) | design (miri)

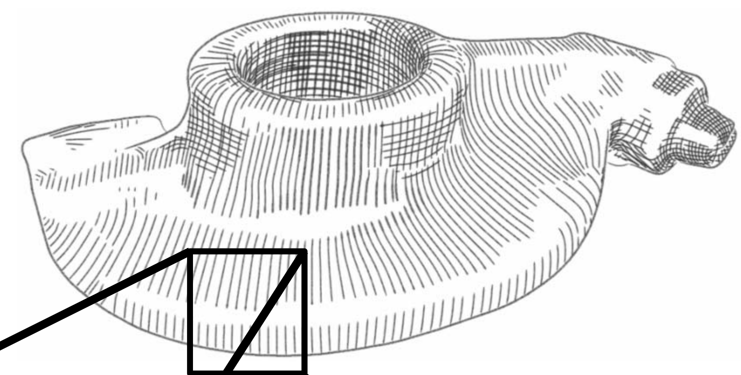
- Closing (miri)

Pen-and-ink Illustration



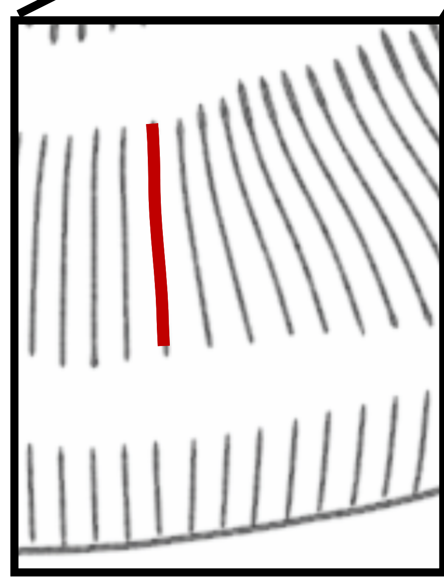
The Problem

input



output

“hatching”

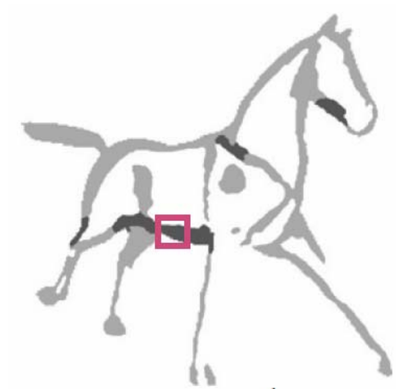


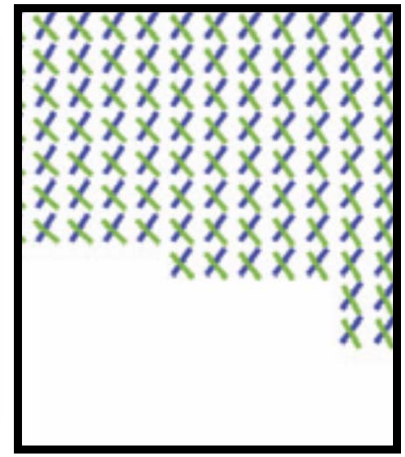
1. Representation?
2. Synthesis?
3. Rendering?

Images: [Kalogerakis et al.,2012]

Hatchings

- Representation
  - Sampled orientations
    - Image space (per pixel)
    - Object space (per face/vertex)

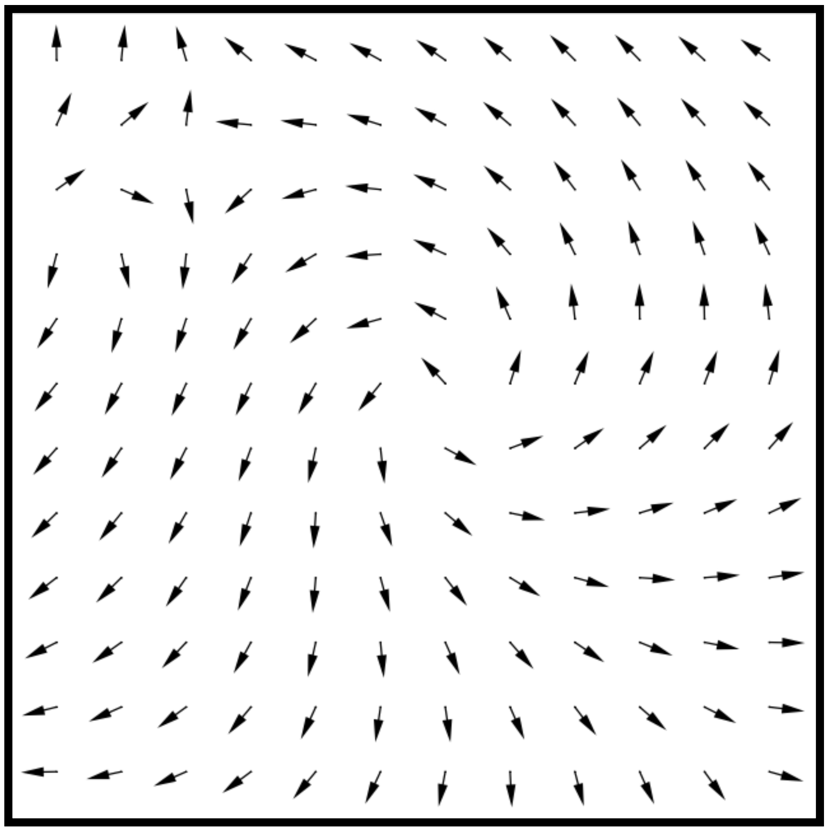




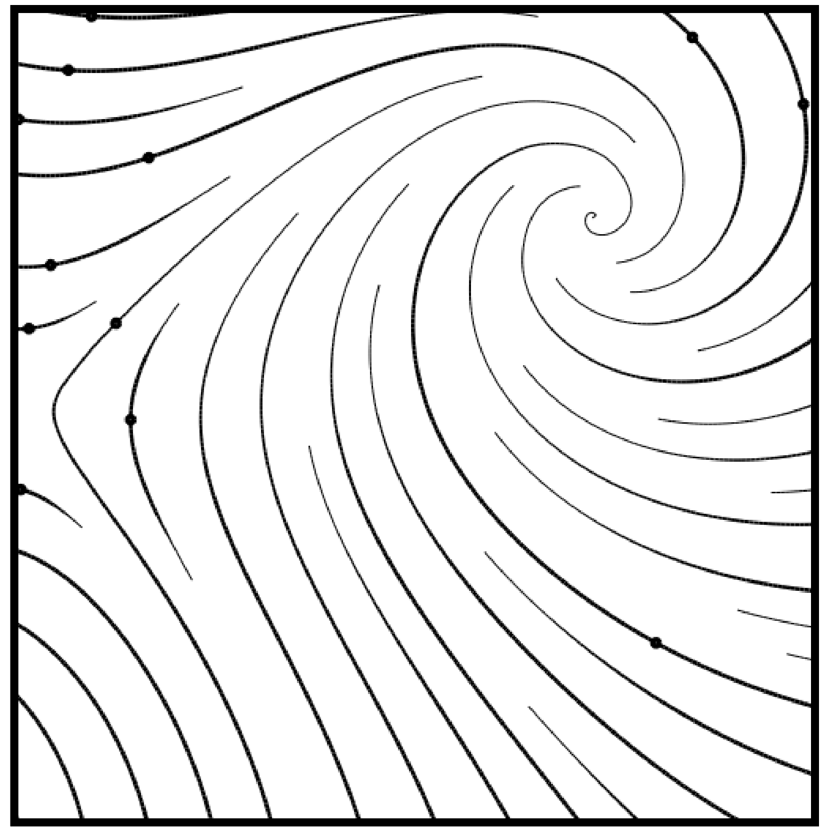
- Synthesis
  - Curvature directions [Hertzmann et al.,2000]
  - Data driven (artist + machine learning) [Kalogerakis et al.,2012]
- Rendering
  - Evenly spaced streamlines

Images: [Kalogerakis et al.,2012]

Evenly Spaced Streamlines



Input: vector per point



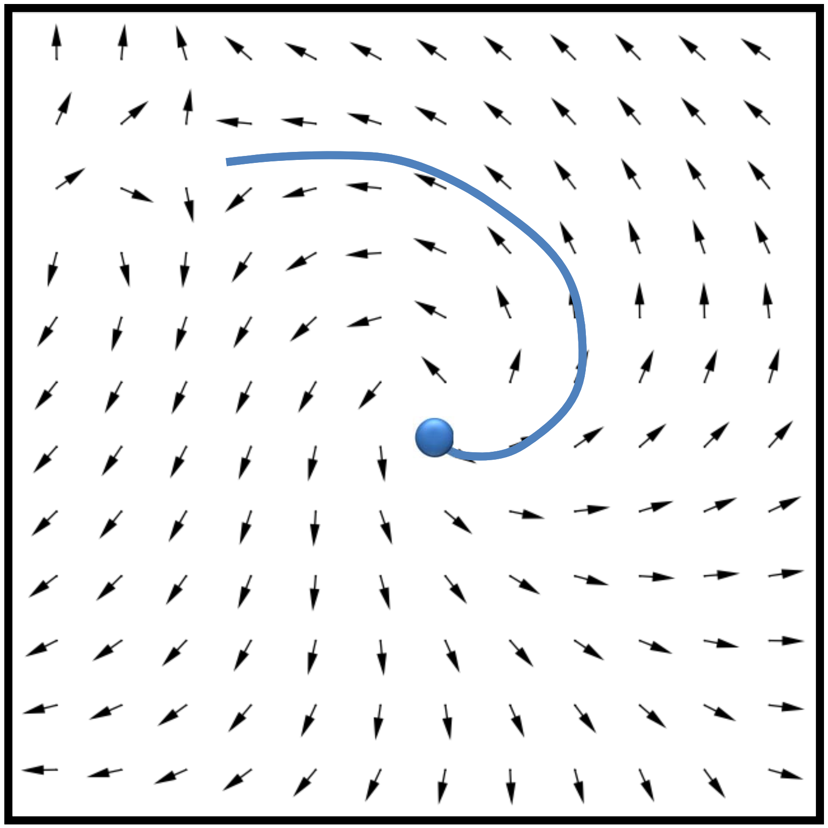
Output

Images: [Mebarki et al.,2005]

Streamlines

http://www.motionexposure.com/

Evenly Spaced Streamlines



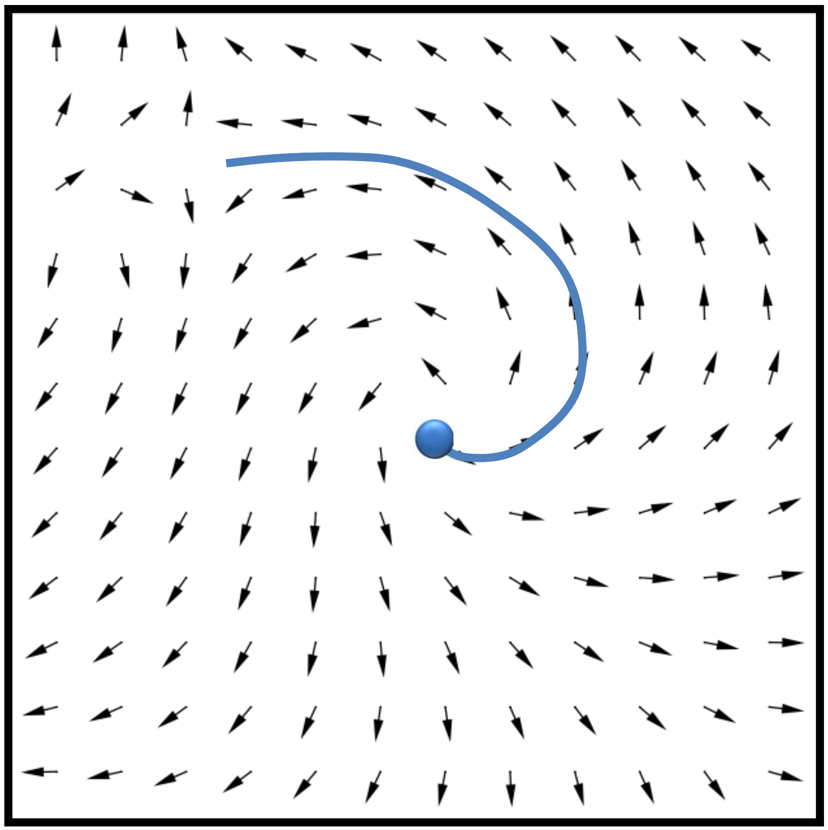
Input: vector per point

Algorithm:

1. Pick a *seed* point
   which?
2. Trace the *streamline*
   how?
3. Repeat
   until?

Images: [Mebarki et al.,2005]

Evenly Spaced Streamlines



Input: vector per point

Algorithm:

1. Pick a seed point
   Furthest from existing
2. Trace the *streamline*
   how?
3. Repeat
   Until space is covered

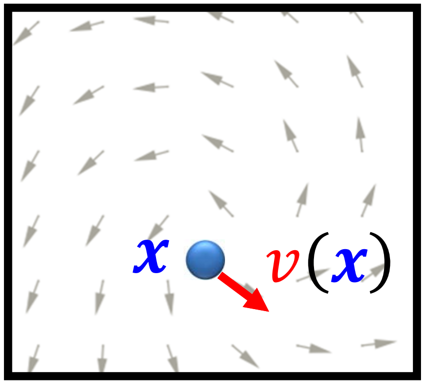
Images: [Mebarki et al.,2005]

Tracing Streamlines

The equation

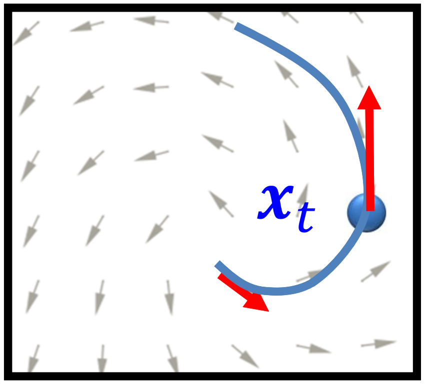
Vector field

$$v = v(\boldsymbol{x}),\ \boldsymbol{x} \in \mathbb{R}^2,\ v \in \mathbb{R}^2$$



Particle path $\boldsymbol{x}_t,\ t \in \mathbb{R}$

$$\frac{d\boldsymbol{x}}{dt} = v(\boldsymbol{x})$$



Tracing Streamlines

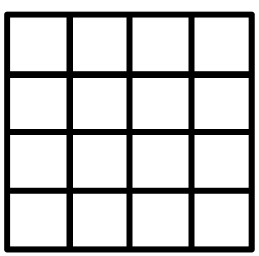
Discretization

$$\frac{d\boldsymbol{x}}{dt} = v(\boldsymbol{x})$$

Time

t $\quad$ $t + \delta t$

Space



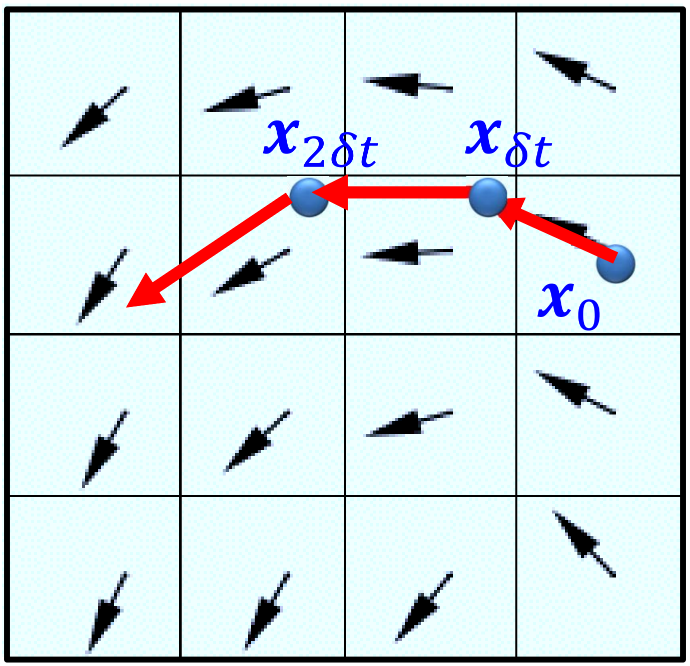
$$\frac{\boldsymbol{x}_{t+\delta t} - \boldsymbol{x}_t}{\delta t} \approx v(\boldsymbol{x}_t)$$

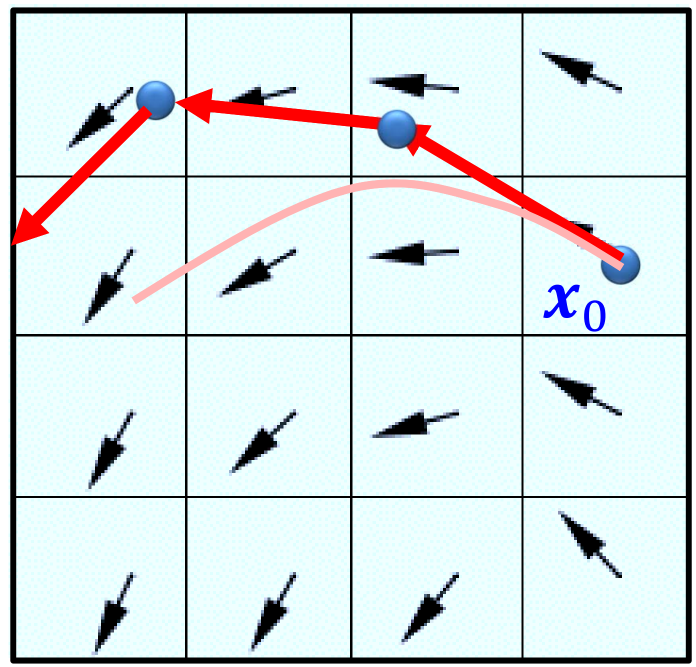
$$\frac{\boldsymbol{x}_{t+\delta t} - \boldsymbol{x}_t}{\delta t} \approx v(\boldsymbol{x}_t)$$

Tracing Streamlines

Euler Integration

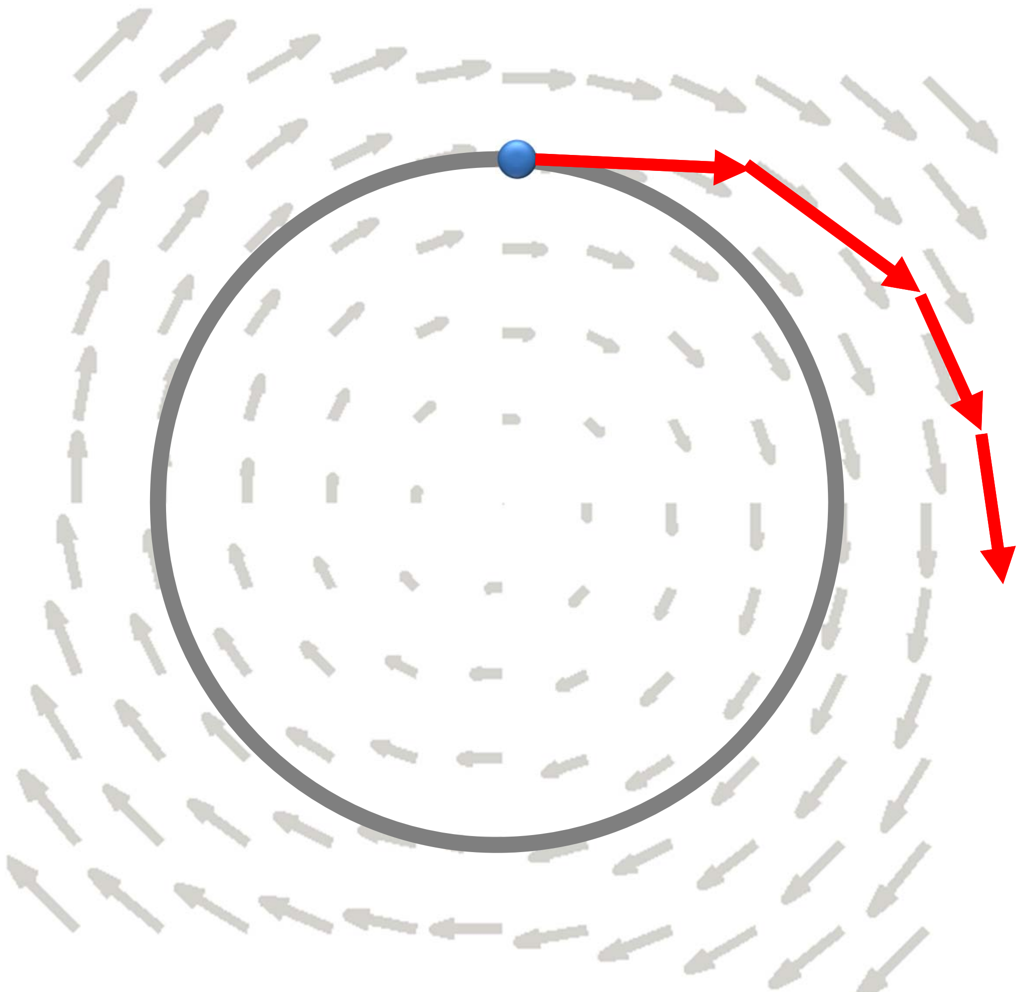
$$\boldsymbol{x}_{t+\delta t} = \boldsymbol{x}_t + \delta t \, v(\boldsymbol{x}_t)$$





Tracing Streamlines

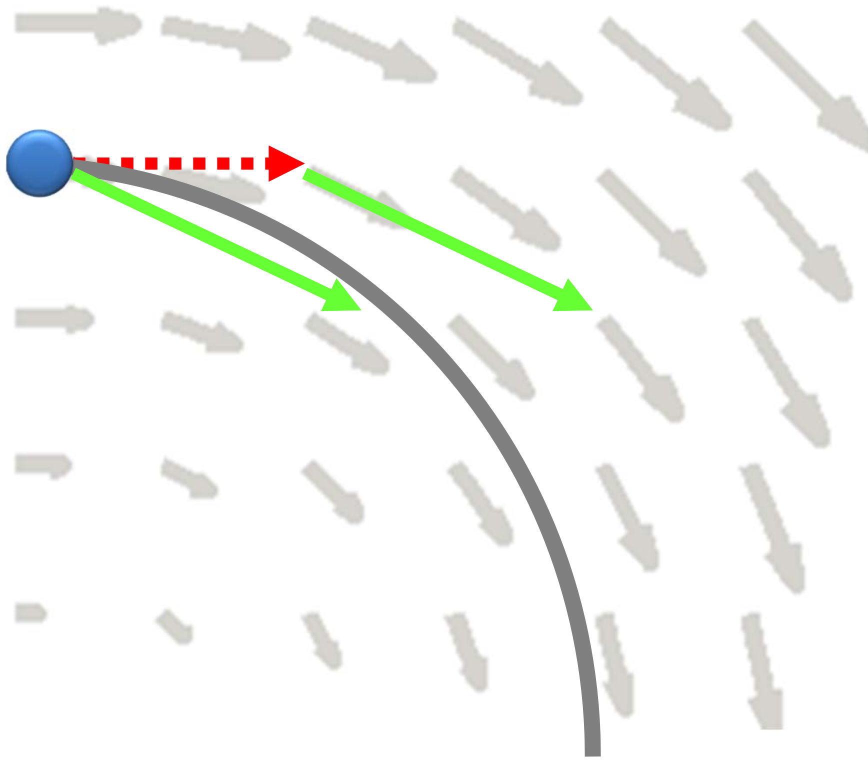
Euler Integration



$$x_{t+\delta t} = x_t + \delta t\, v(x_t)$$

Works well only
for very small steps

Tracing Streamlines
Runge-Kutta Integration

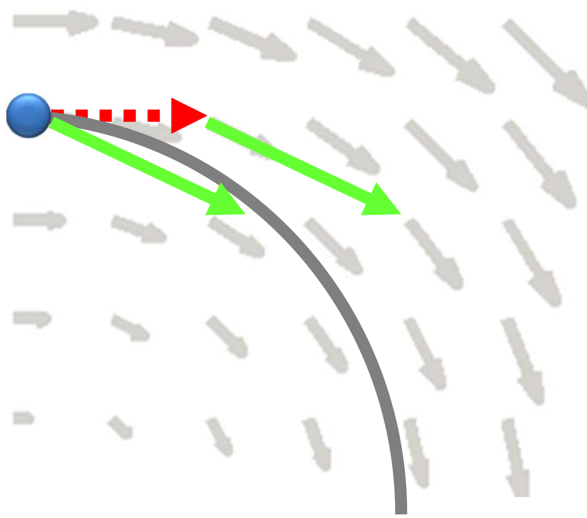


Idea: "cut" short the curve arc

1. Half an Euler
2. Evaluate v there
3. Use it at origin

Tracing Streamlines

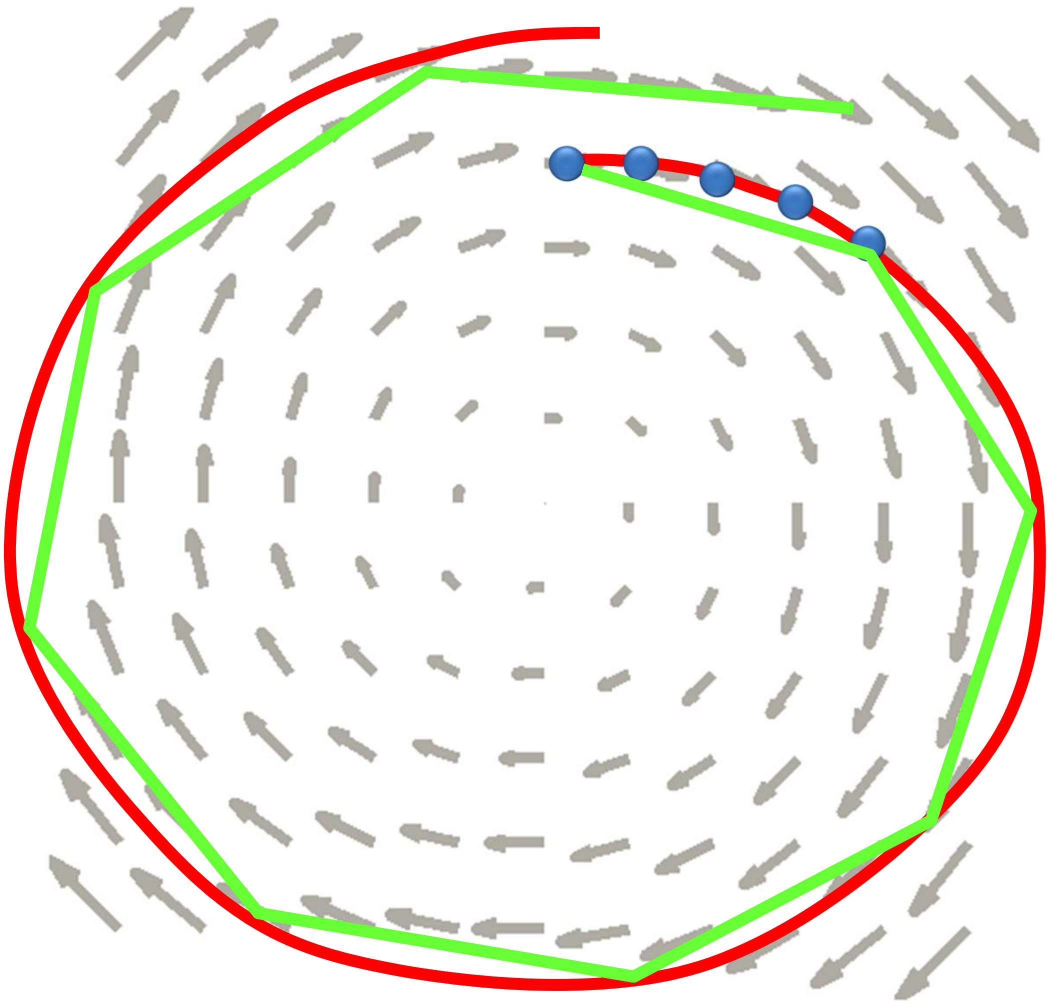
Runge-Kutta Integration



1. Half an Euler
2. Evaluate v there
3. Use it at origin

$$\boldsymbol{x}_{t+\delta t} = \boldsymbol{x}_t + \delta t\, v\left(\boldsymbol{x}_t + \frac{\delta t}{2} v(\boldsymbol{x}_t)\right)$$

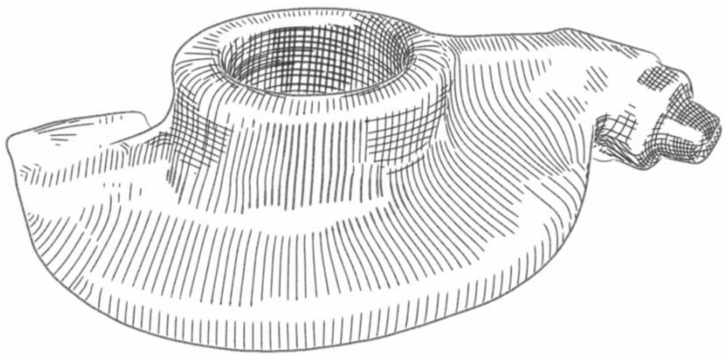
Tracing Streamlines
Runge-Kutta Integration



With **RK** can use **larger time steps** than **Euler** with similar accuracy

The Problem

input



output

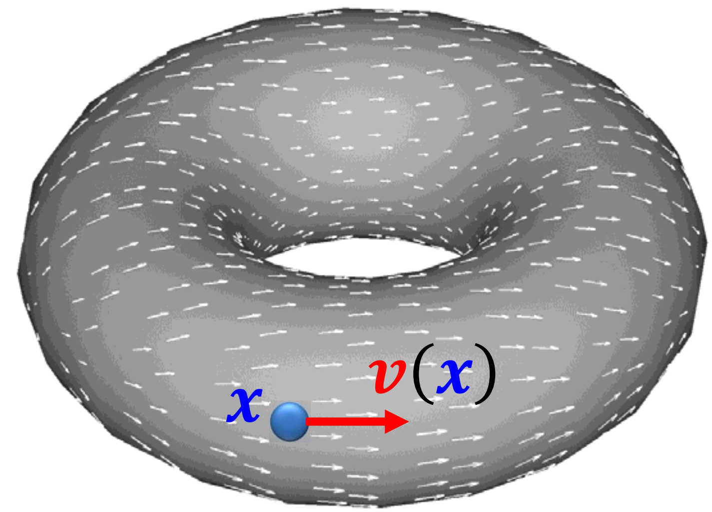
Tracing a streamline on a surface?

Tracing Streamlines on Surfaces

The equation

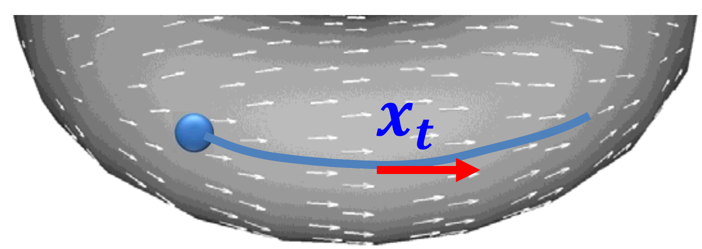
Vector field

$$v = v(\boldsymbol{x}),\ \boldsymbol{x} \in M,\ v \in T_x M$$



Particle path $\boldsymbol{x}_t,\ t \in \mathbb{R}$

$$\frac{d\boldsymbol{x}}{dt} = v(\boldsymbol{x})$$



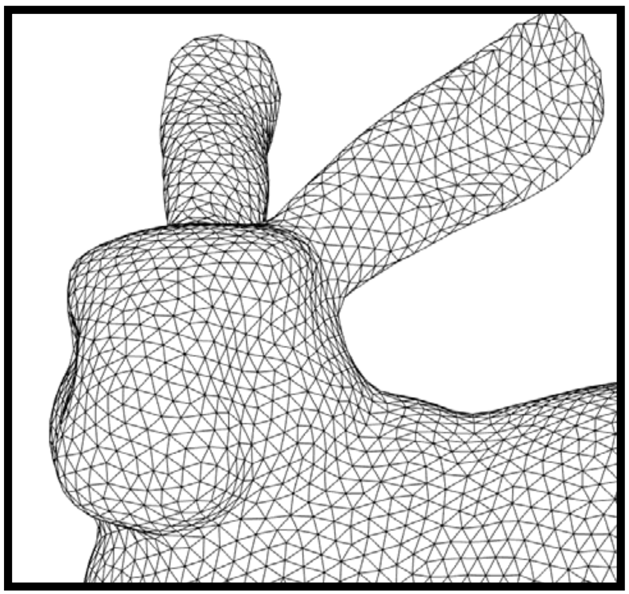
Tracing Streamlines on Surfaces

Euler Integration?

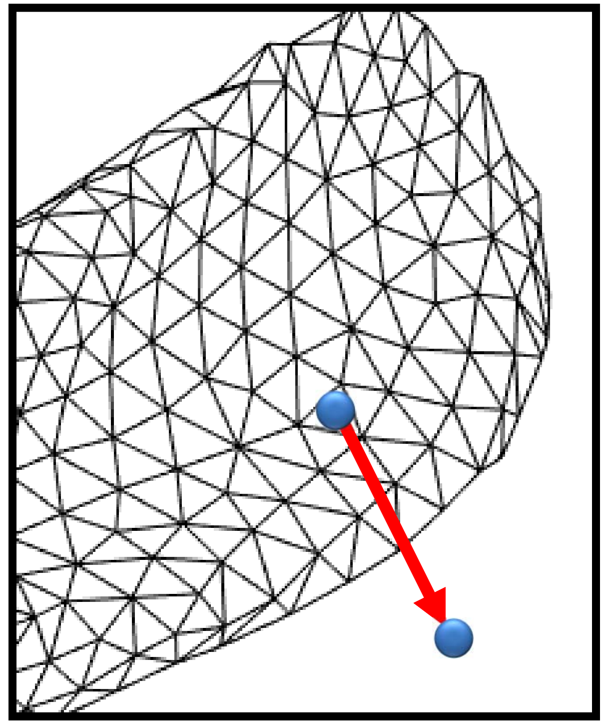
Time

t $t + \delta t$

Space



$$x_{t+\delta t} = x_t + \delta t \, v(x_t)$$



Not on the surface!

Tracing Streamlines on Surfaces

Euler Integration?

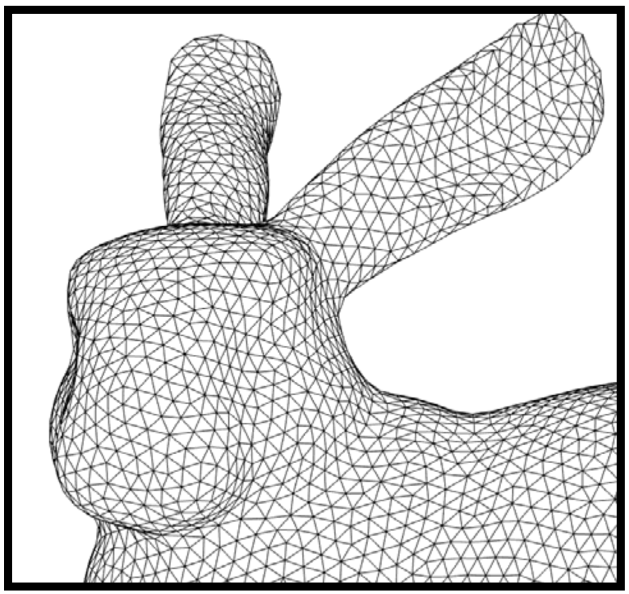
Time

$t \quad\quad t + \delta t$

$$\boldsymbol{x}_{t+\delta t} = \boldsymbol{x}_t + \delta t\, v(\boldsymbol{x}_t)$$

$$= \boldsymbol{line}(\boldsymbol{x}_t, \delta t, v(\boldsymbol{x}_t))$$

Space



End of line starting from $\boldsymbol{x}_t$
in the direction $v(x_t)$
of length δt

Tracing Streamlines on Surfaces

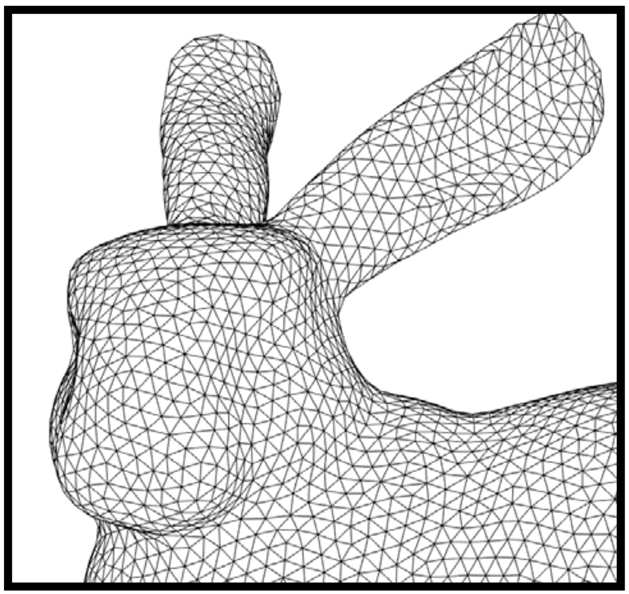
Euler Integration?

Time

t $\qquad$ $t + \delta t$

$$\boldsymbol{x}_{t+\delta t} = \boldsymbol{x}_t + \delta t\, v(\boldsymbol{x}_t)$$

Space



$$= \boldsymbol{g}(\boldsymbol{x}_t, \delta t, v(\boldsymbol{x}_t))$$

End of **geodesic** starting from $\boldsymbol{x}_t$
in the direction $v(x_t)$
of length δt

Tracing a Geodesic

On a *smooth surface* the initial value problem:

1. Start from a point $\boldsymbol{x}$ $\qquad g(0) = \boldsymbol{x}$
2. In the direction v $\qquad \frac{dg}{dt}(0) = v$
3. Continue "straight" $\qquad \frac{d^2g}{dt^2}(t)^{tangent} = 0$

has a *unique* solution $g(t)$

[Polthier et al.,1998]

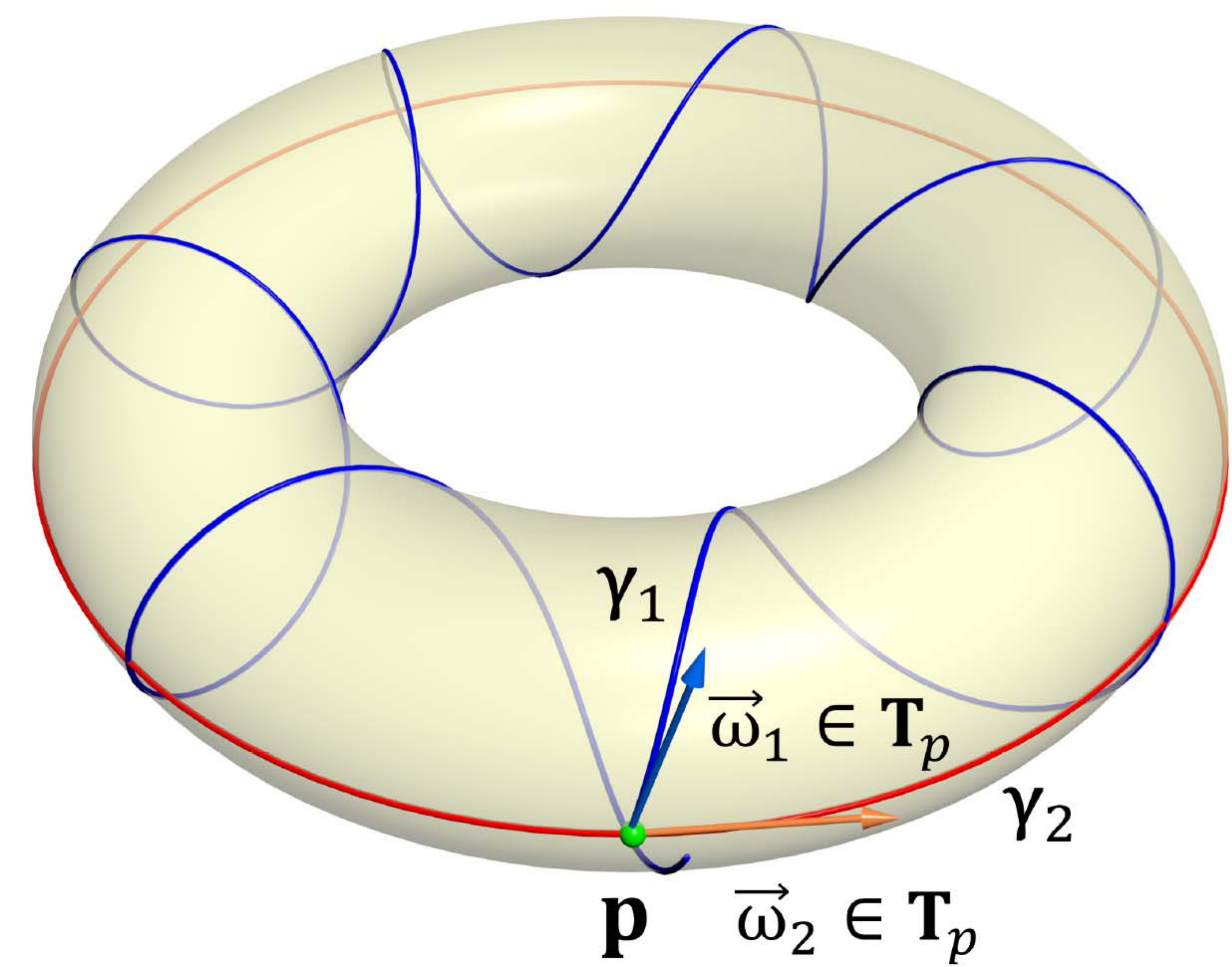
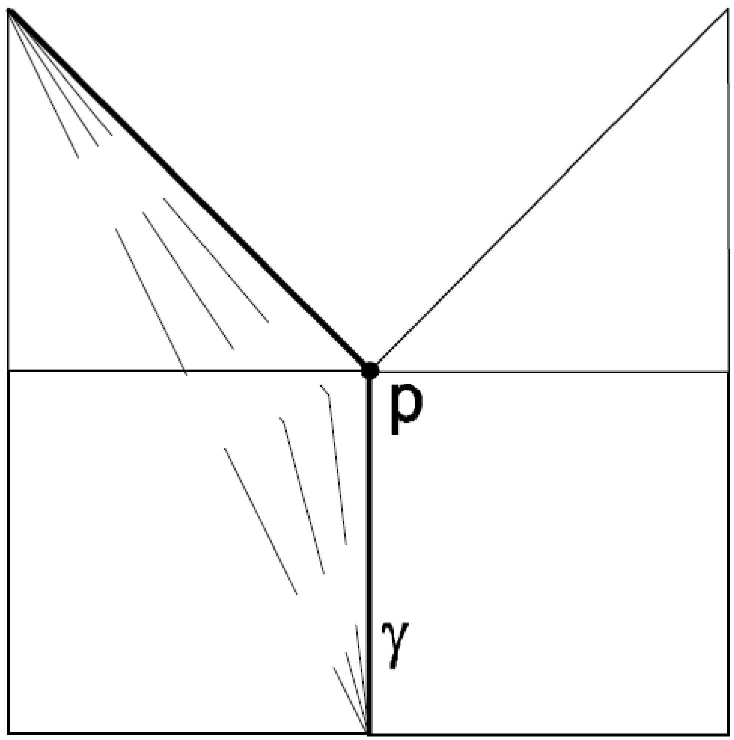
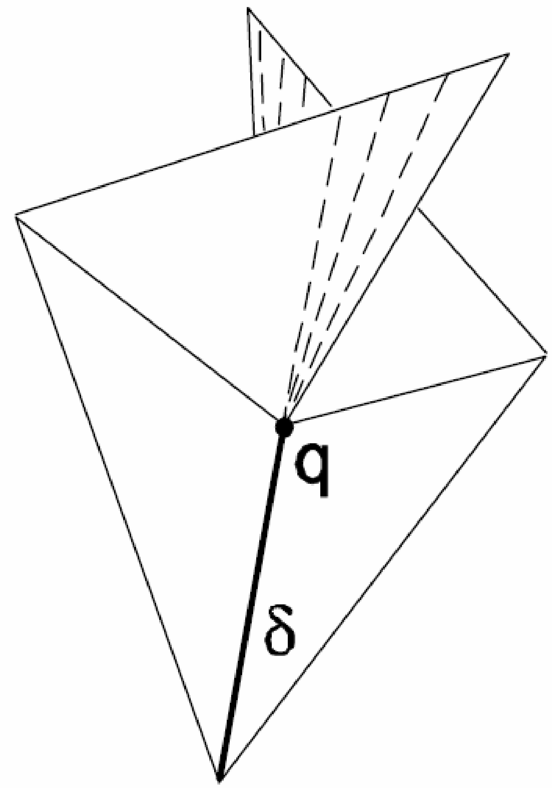


image from [Cheng et al.,2015]

Tracing a Shortest Geodesic
on a Triangle Mesh



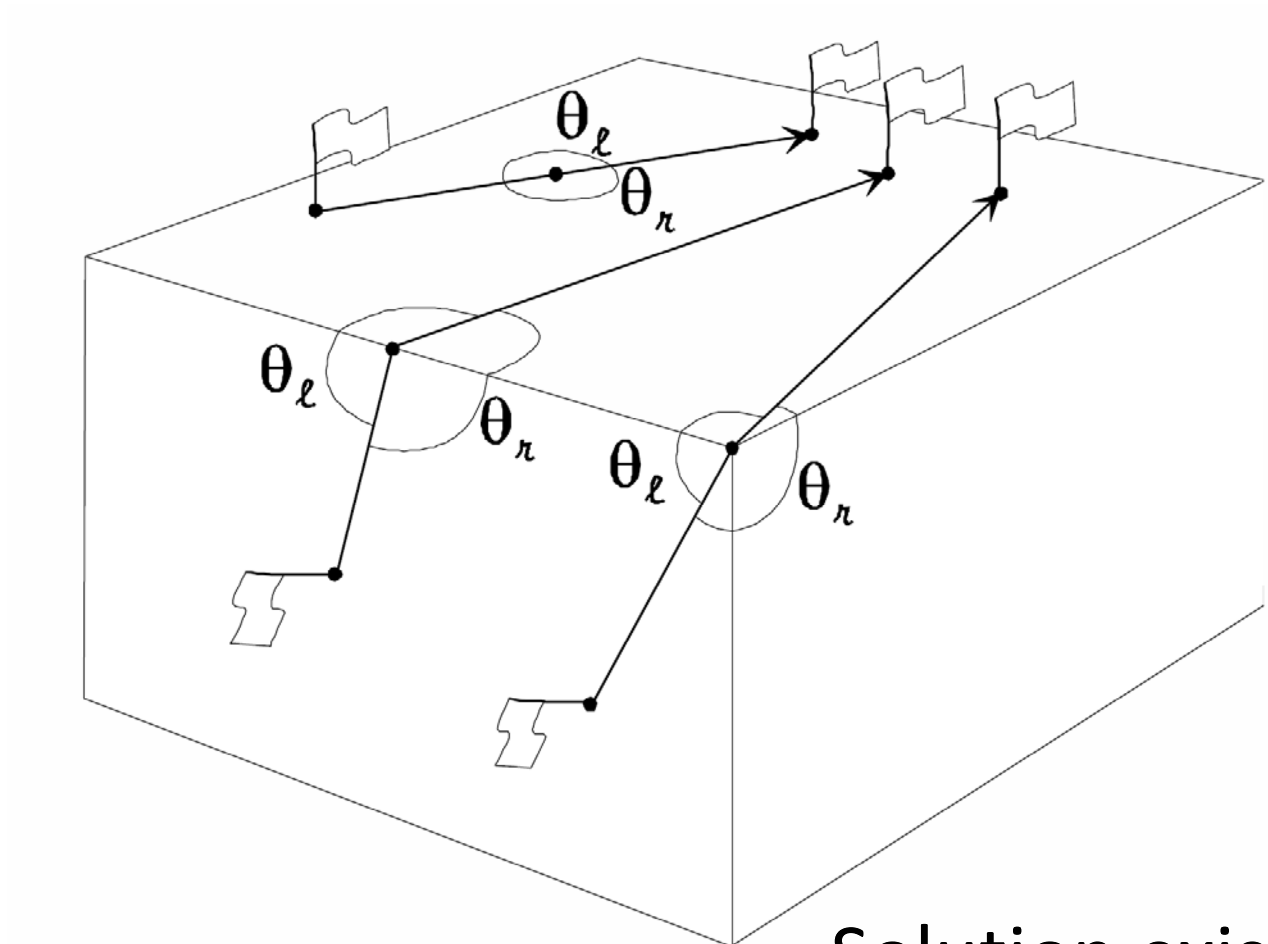
No solution
spherical vertex



Multiple solutions
hyperbolic vertex

[Polthier et al.,1998]

Tracing a Straightest Geodesic
on a Triangle Mesh



$$\theta_l = \theta_r$$

Solution exists and is unique
from any point $\boldsymbol{x}$ and direction v

[Polthier et al.,1998]

Tracing Streamlines on Surfaces

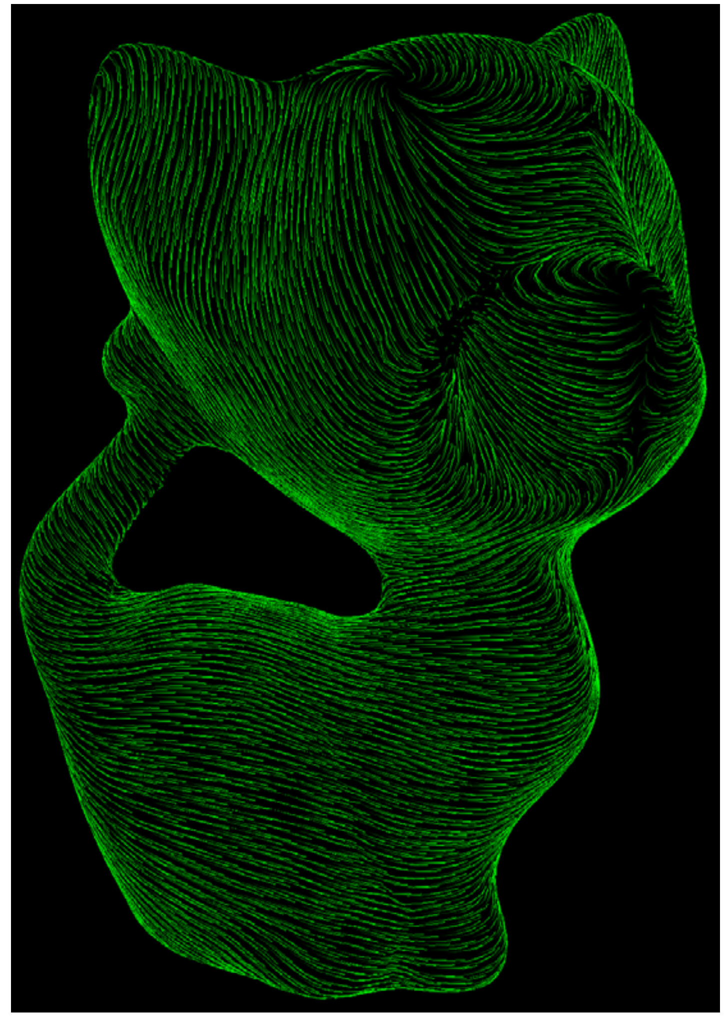
Geodesic Euler Integration

$$\boldsymbol{x}_{t+\delta t} = \boldsymbol{g}(\boldsymbol{x}_t, \delta t, \boldsymbol{v}(\boldsymbol{x}_t))$$

where g is the straightest geodesic
from $\boldsymbol{x}_t$
in the direction v
and length δt

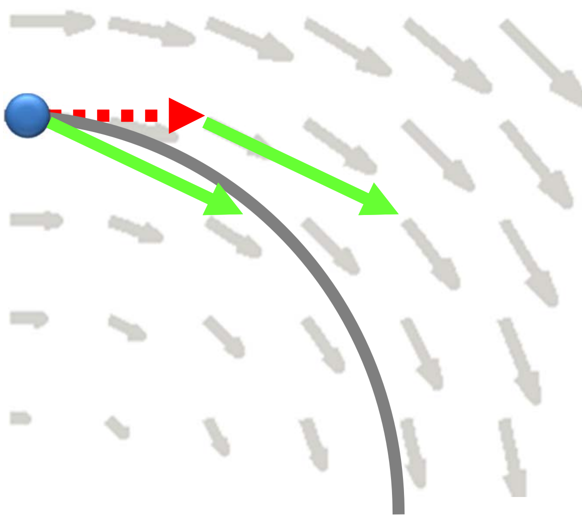
Example

- Missing:
  - Smart seed points sampling
  - Streamline proximity for stopping



Tracing Streamlines on Surfaces

Geodesic Runge-Kutta Integration



$$\boldsymbol{x}_{t+\delta t} = \boldsymbol{x}_t + \delta t\, v\left(\boldsymbol{x}_t + \frac{\delta t}{2} v(\boldsymbol{x}_t)\right)$$

$$\boldsymbol{x}_{t+\delta t} = g\left(\boldsymbol{x}_t, \delta t, v\left(g\left(\boldsymbol{x}_t, \frac{\delta t}{2}, v(\boldsymbol{x}_t)\right)\right)\right)$$

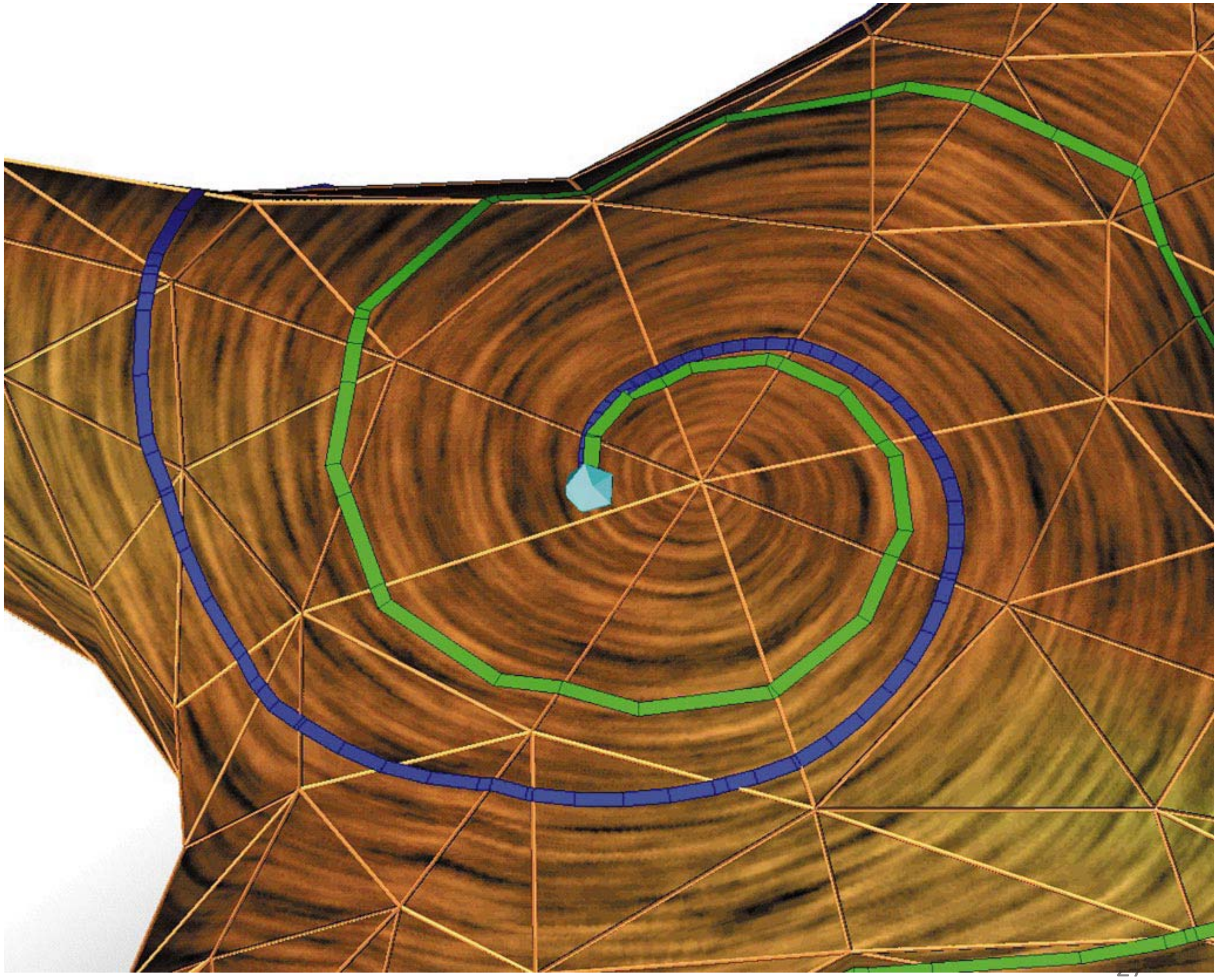
Parallel transport

[Polthier et al.,1998]

Example

Euler

RK4



[Polthier et al.,1998]

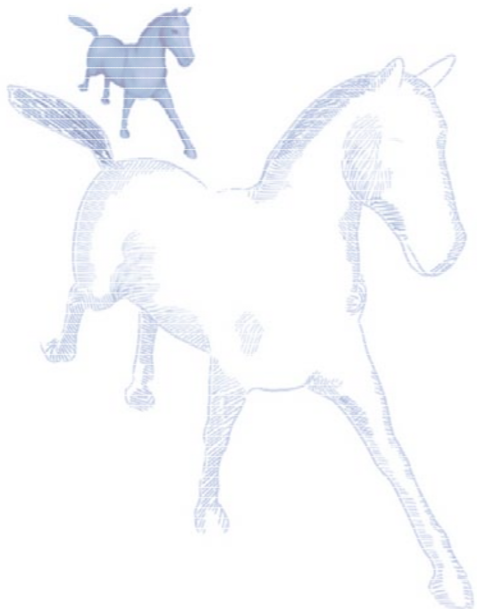
Recap

- Discrete vector field: sampled orientations + length
- Can render by tracing intelligently chosen streamlines
- On a surface, take care when going straight

References

- “Solving the initial value problem of discrete geodesics”, Cheng et al., 2015
- “Illustrating smooth surfaces**”,** Hertzmann et al., 2000
- “Learning Hatching for Pen-and-Ink Illustration of Surfaces”, Kalogerakis et al., 2012
- “Farthest Point Seeding for Efcient Placement of Streamlines**”,** Mebarki et al., 2005
- “Straightest geodesics on polyhedral surfaces”, Polthier et al., 1998

Pen-and-ink Illustration
and streamline tracing



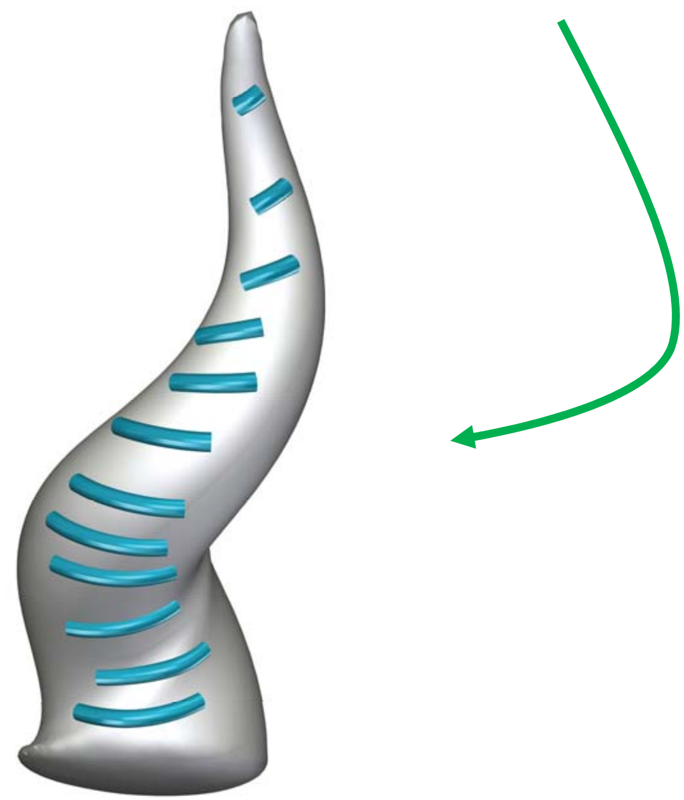
Visualization

Application:
Vector Field Visualization

VF Visualization

How to visualize motion?

A subset of flow lines



Taken from: Ben-Chen et al. SGP'10

Heat Equation

Distribution of heat is described by the heat equation

$$\partial_t f = \Delta f$$

Change in temperature

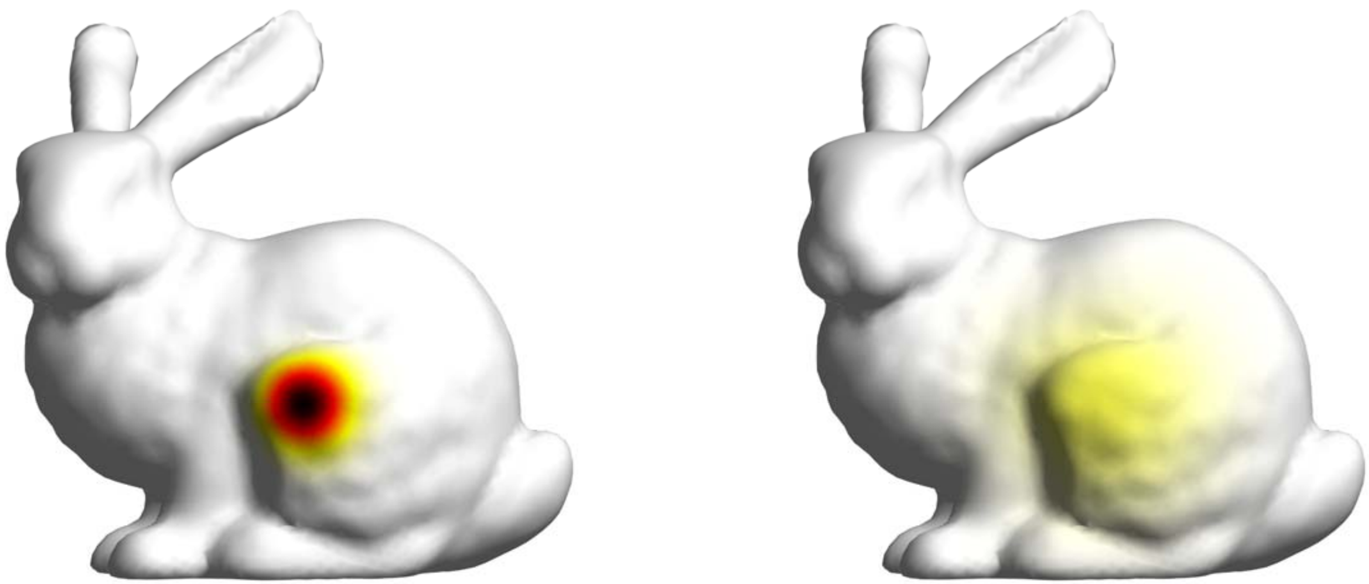
Temperature

Laplace—Beltrami Operator

Heat Equation

Distribution of heat is described by the heat equation

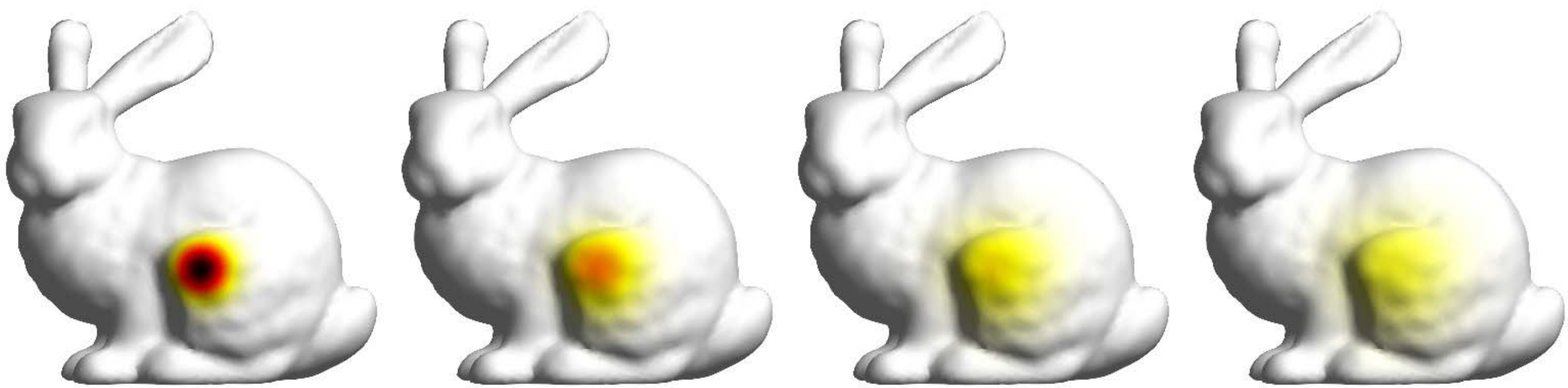
$$\partial_t f = \Delta f$$



$\partial_t f = \Delta f$

Heat Equation

Plot a sequence of images!

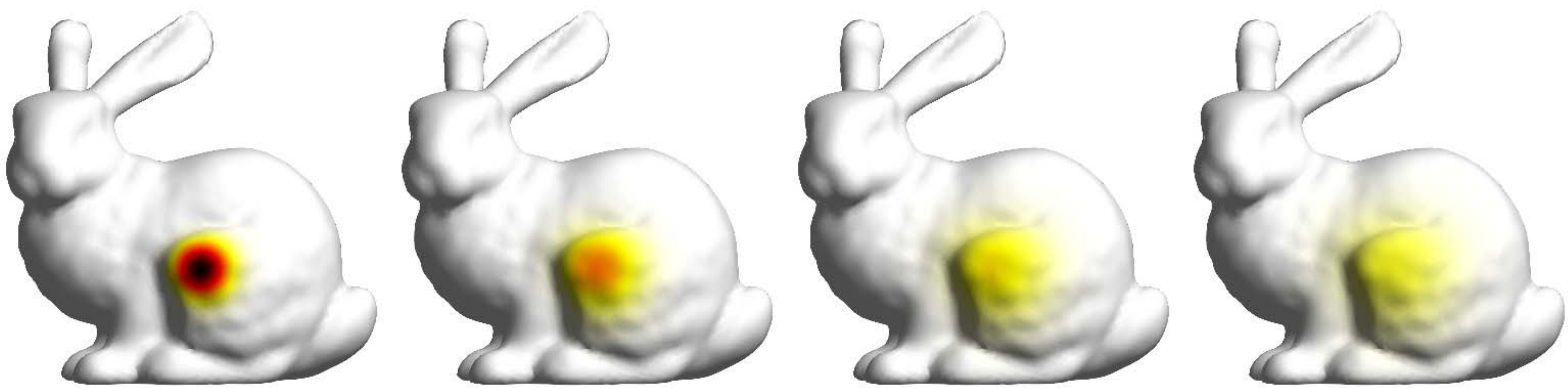


$\partial_t f = \Delta f$

Heat Equation

Plot a sequence of images!

Isotropic motion



VF Visualization

Shortcomings of the former approach:

- **uniform** propagation of information

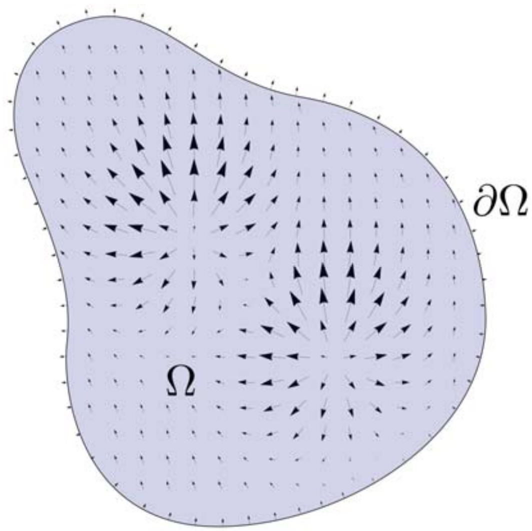
- need a **sequence** of images

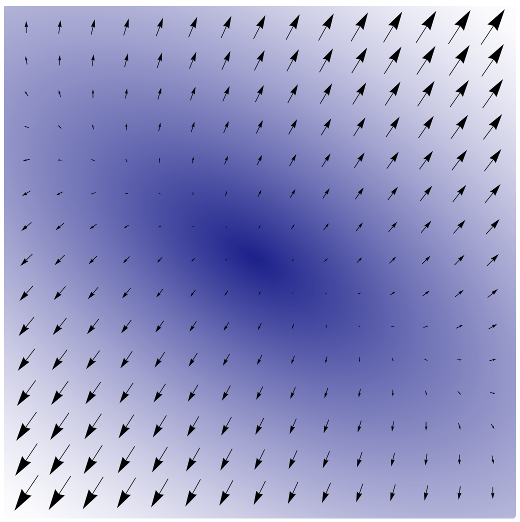
$\partial_t f = \Delta f$

VF Visualization

Or alternatively

$$\partial_t f = \mathrm{div}(\mathrm{grad}\, f)$$



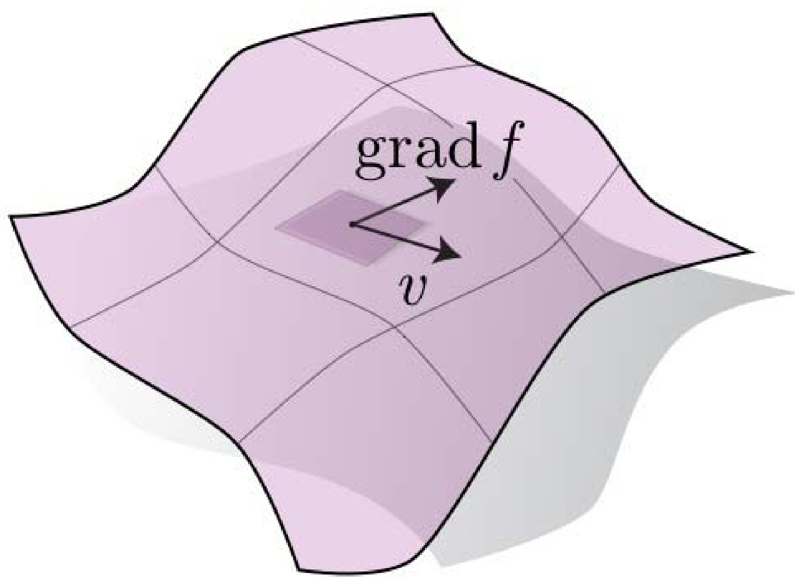


http://brickisland.net/cs177fa12/

VF Visualization

Project onto a given vector field v

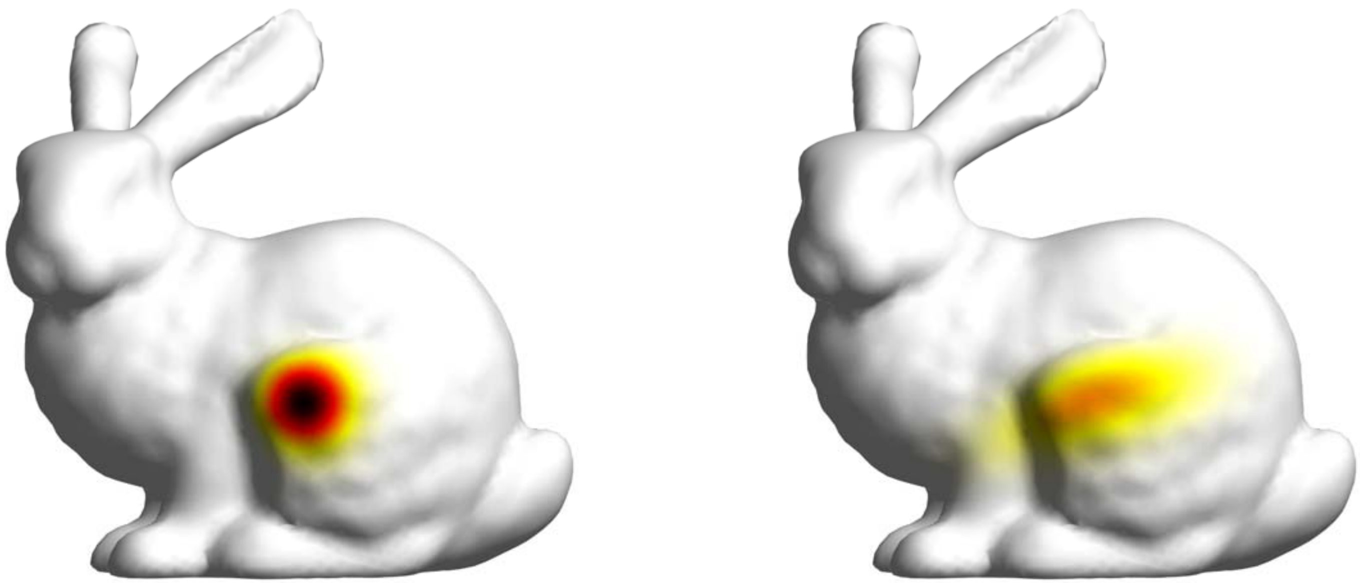
$$\partial_t f = \mathrm{div}(\square v \cdot \mathrm{grad}\, f)$$



Anisotropic Diffusion

Project onto a given vector field v

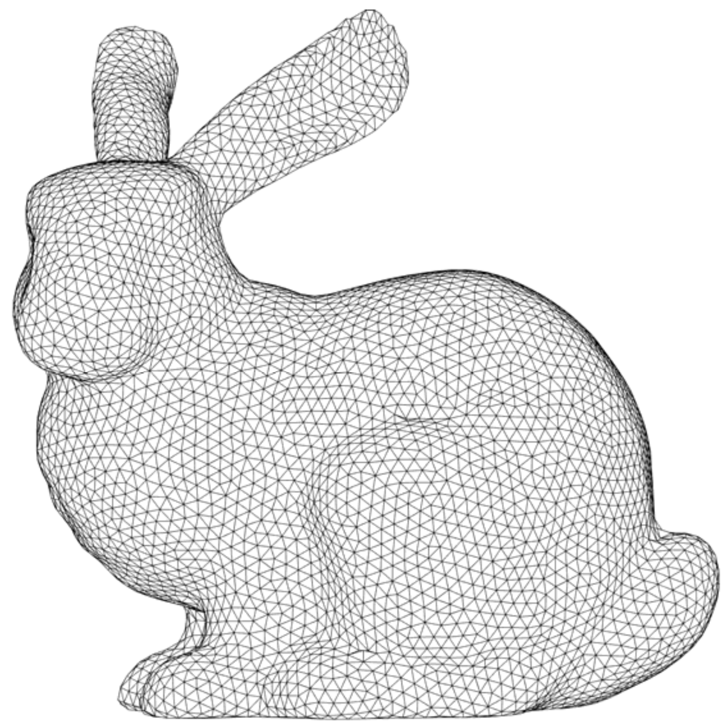
$$\partial_t f = \operatorname{div}(vv^T \operatorname{grad} f)$$



Discretization

Time

Space

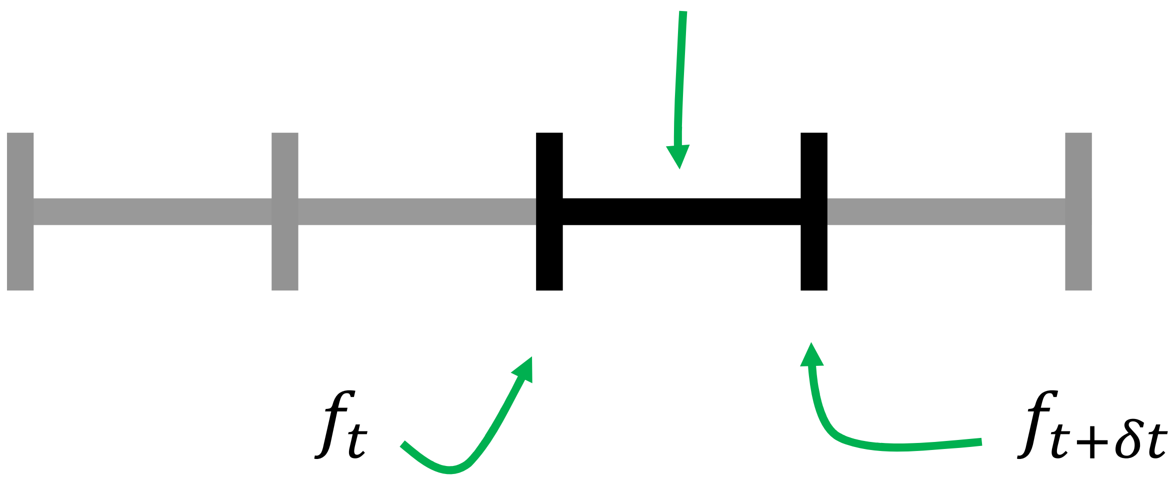


$\partial_t f = \mathrm{div}(A_v \,\mathrm{grad}\, f)$

Temporal Discretization

(Straight)Forward finite differences

$$\partial_t f \approx \frac{f_{t+\delta t} - f_t}{\delta t}$$



$f_{t+\delta t} = f_t + \delta t \partial_t f$

Temporal Discretization

Forward finite differences

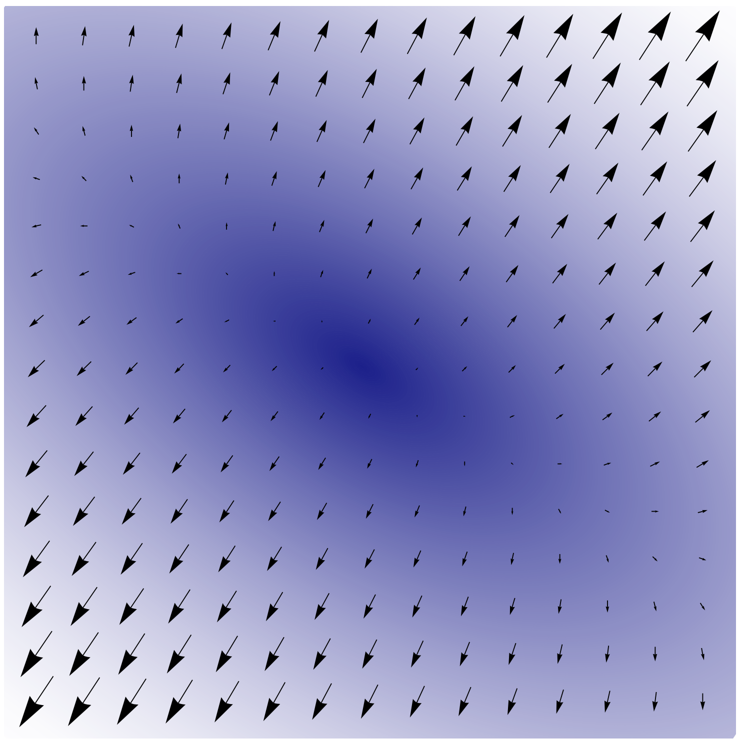
$$f_{t+\delta t} = f_t + \delta t \operatorname{div}(A_v \operatorname{grad} f_t)$$

Also known as
Explicit integration

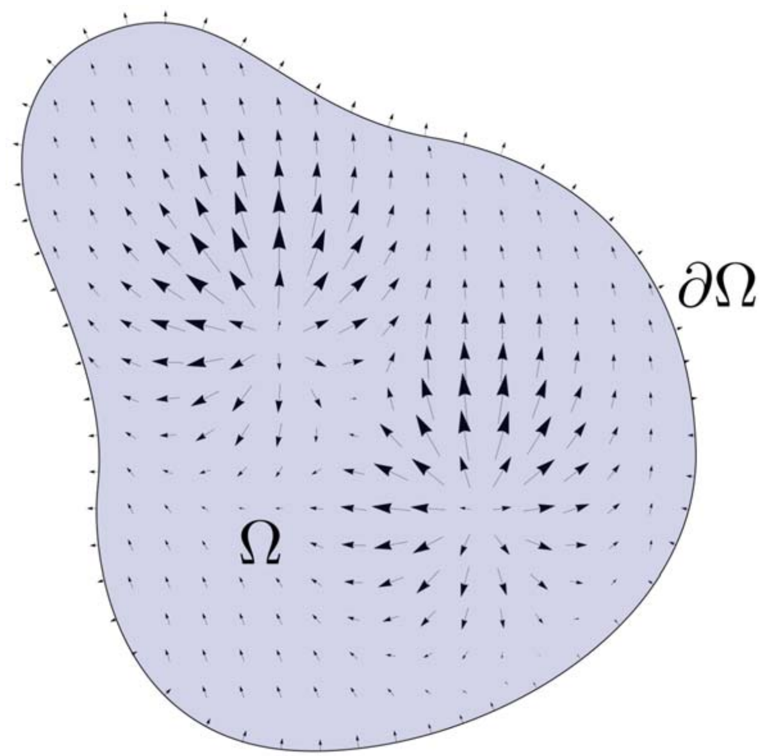
$\partial_t f = \mathrm{div}(A_v \,\mathrm{grad}\, f)$

Spatial Discretization

Gradient Operator



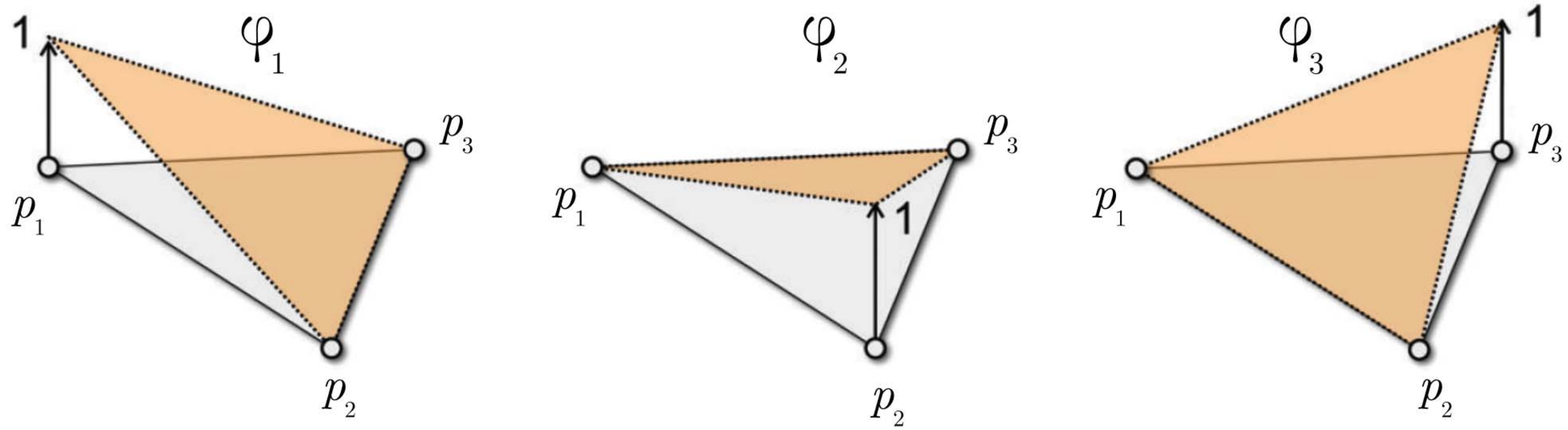
Divergence Operator



Discrete PL Functions

Functions are linearly interpolated over triangles

$$f(p) = \sum_{i=1}^{3} f_i \varphi_i(p)$$

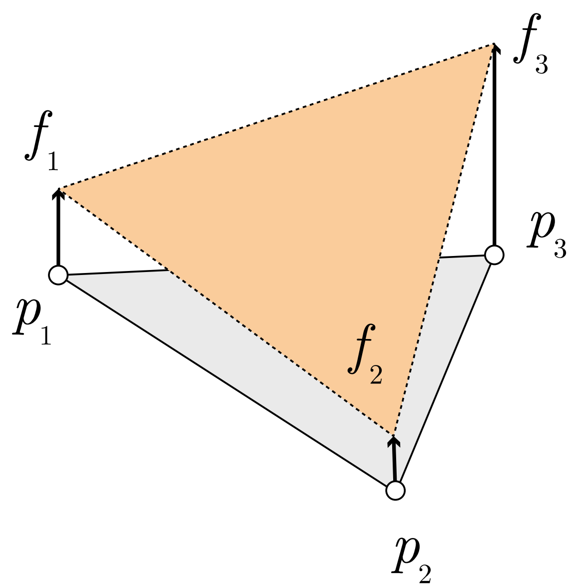


http://www.pmp-book.org/

Discrete PL Functions

The values at the vertices are given by $\{f_i\}$

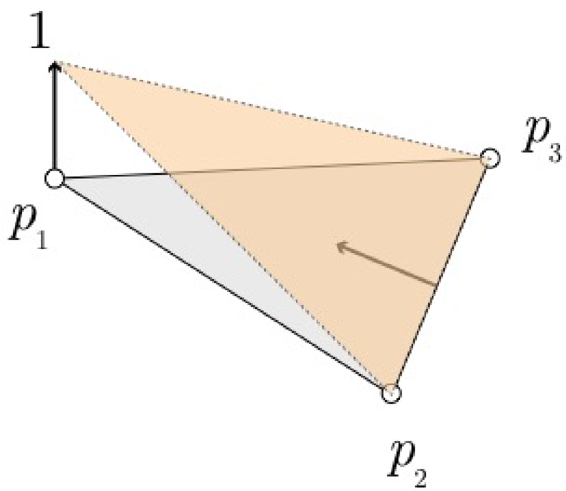
$$f(p) = \sum_{i=1}^{3} f_i \varphi_i(p)$$



Discrete Gradient

Only compute the gradients of basis functions

$$\text{grad}\, f(p) = \sum_{i=1}^{3} f_i \,\text{grad}\, \varphi_i(p)$$



Discrete Gradient

It is easy to show that

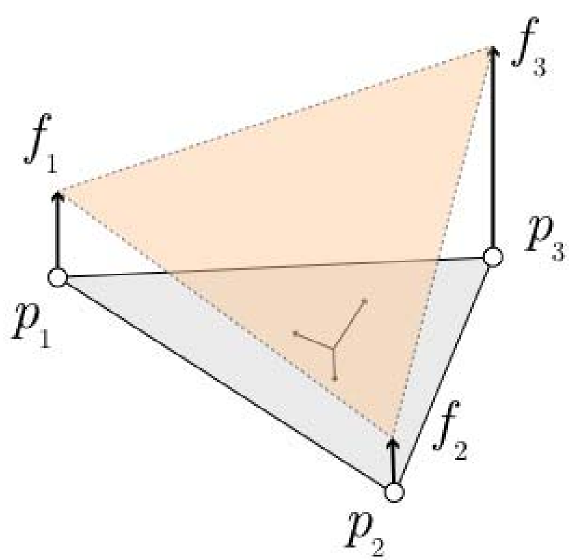
$$\text{grad }\ \varphi_1\Big|_j = \frac{\mathcal{J}(p_3 - p_2)}{2A_j}$$

where $\mathcal{J}$ rotates by $\pi/2$ in the tangent plane

Discrete Gradient

Finally, we obtain

$$\text{grad}\ f\Big|_j = \frac{1}{2A_j}\sum_{i=1}^{3} f_i\, \mathcal{J} e_i$$



Discrete Divergence

The following holds for any f and v

$$\int f \cdot \operatorname{div} v \, \mathrm{d}a + \int \operatorname{grad} f \cdot v \, \mathrm{d}a = 0$$

Integration by parts

$$\int f \cdot \text{div}\, v\, \text{d}a + \int \text{grad}\, f \cdot v\, \text{d}a = 0$$

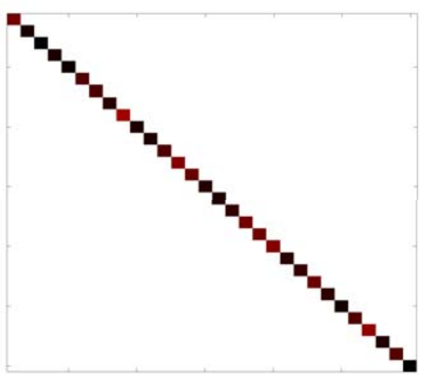
Discrete Divergence

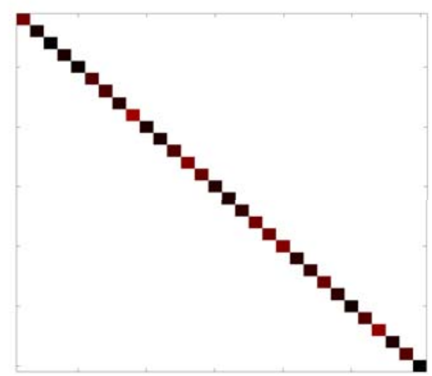
The discrete analog

$$f^T A_{\mathcal{V}}\, \text{div}\, v + f^T \text{grad}^T A_{\mathcal{F}} v = 0$$

Vertex areas

Triangle areas





Discrete Divergence

For operators, we obtain

$$A_{\mathcal{V}}\,\mathrm{div} \; + \mathrm{grad}^T A_{\mathcal{F}} = 0$$

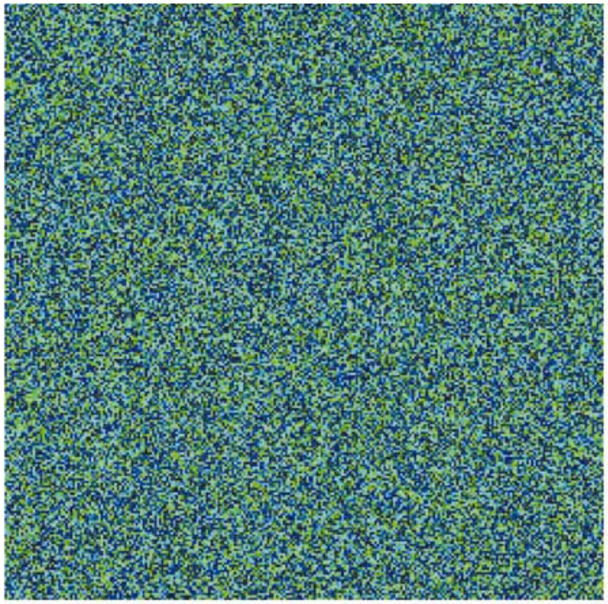
Thus,

$$\mathrm{div} = -A_{\mathcal{V}}^{-1}\,\mathrm{grad}^T A_{\mathcal{F}}$$

VF Visualization

Instead of a sequence of images, plot a single image!

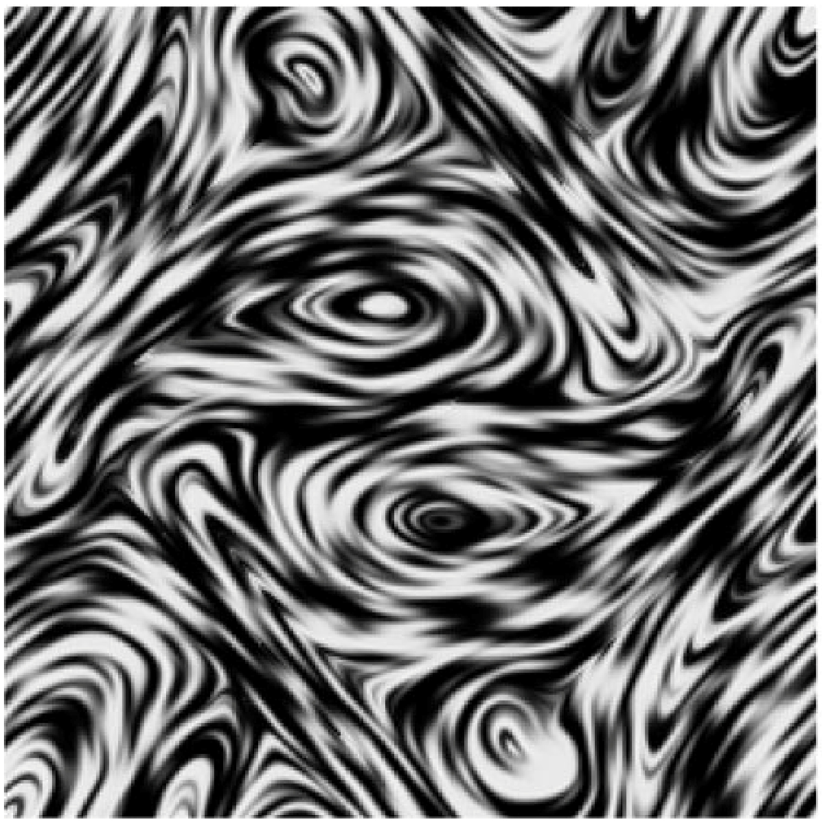
The solution: use initial random noise

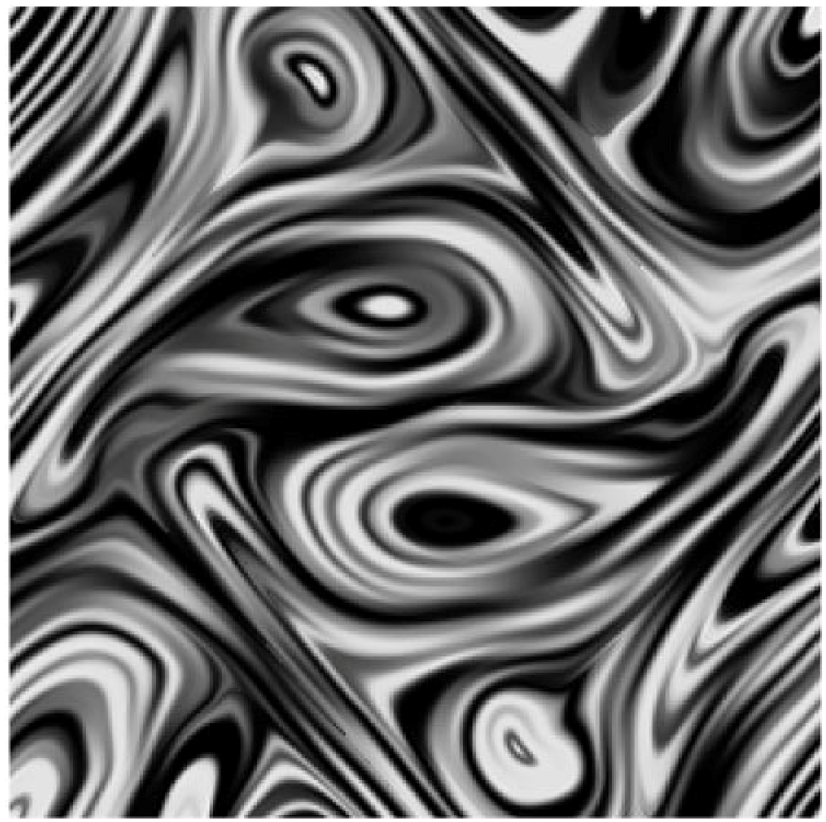


Taken from: Diewald et al. TVCG'00

Results

Results





Taken from: Diewald et al. TVCG'00

Results

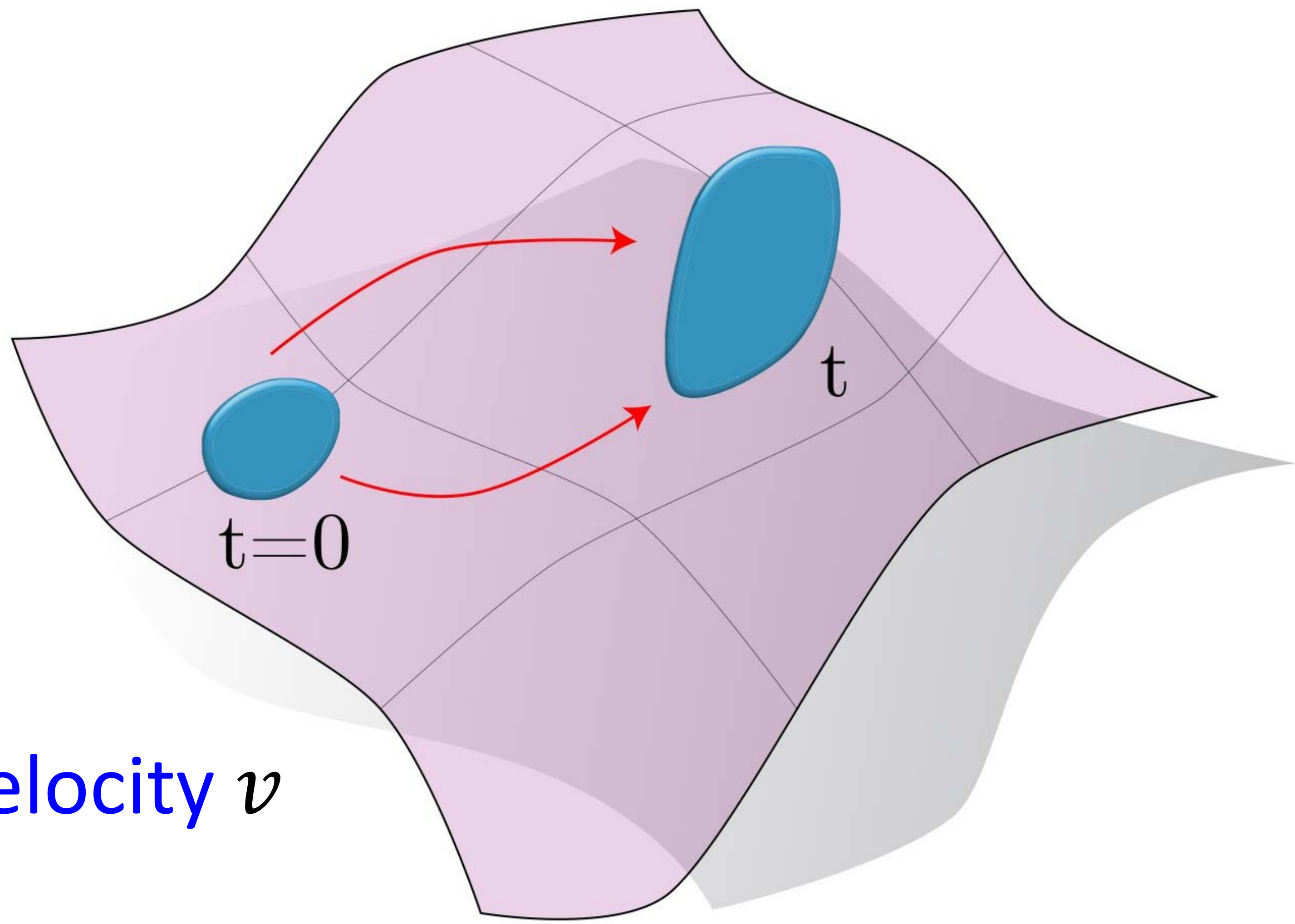
Taken from: Diewald et al. TVCG'00

Results

Taken from: Diewald et al. TVCG'00

Application:
Fluid Simulation

Fluid Mechanics



velocity v

Fluid Mechanics

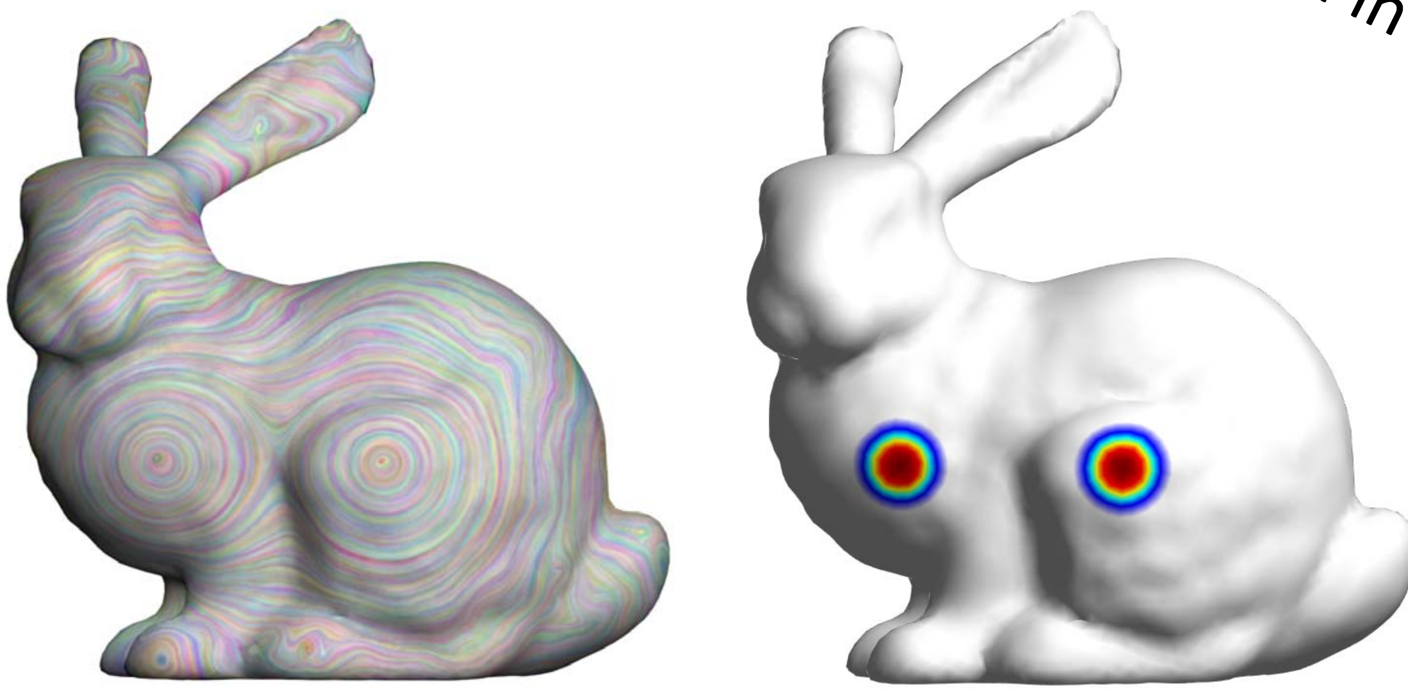
The vorticity describes the local spinning

$$\omega = \mathrm{curl}\, v$$

2D Fluid Mechanics

$$\omega = \operatorname{curl} v$$

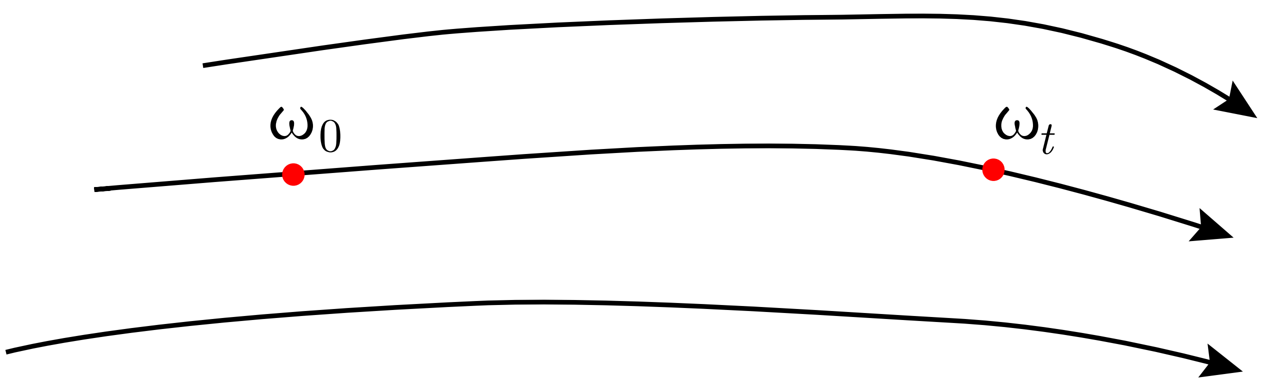
A scalar function in 2D



Momentum Equation

Vorticity is transported along the flow lines

$$\partial_t \omega = v \cdot \operatorname{grad} \omega$$
$$\omega = \operatorname{curl} v$$



$$\partial_t \omega \approx (\omega_{t+\delta t} - \omega_t)/\delta t$$

Explicit Integration

$$\omega_{t+\delta t} = \omega_{t} + \delta t\, v_{t} \cdot \operatorname{grad} \omega_{t}$$

Conditionally stable at best!

(Semi-)Implicit Integration

$$\omega_{t+\delta t} = \omega_t + \delta t \; v_t \cdot \operatorname{grad} \omega_{t+\delta t}$$

(Semi-)Implicit Integration

$$\omega_{t+\delta t} = \omega_{t} + \delta t \, v_{t} \cdot \mathrm{grad}\, \omega_{t+\delta t}$$

$$\Downarrow$$

$$(I - \delta t \, v_{t} \cdot \mathrm{grad})\omega_{t+\delta t} = \omega_{t}$$

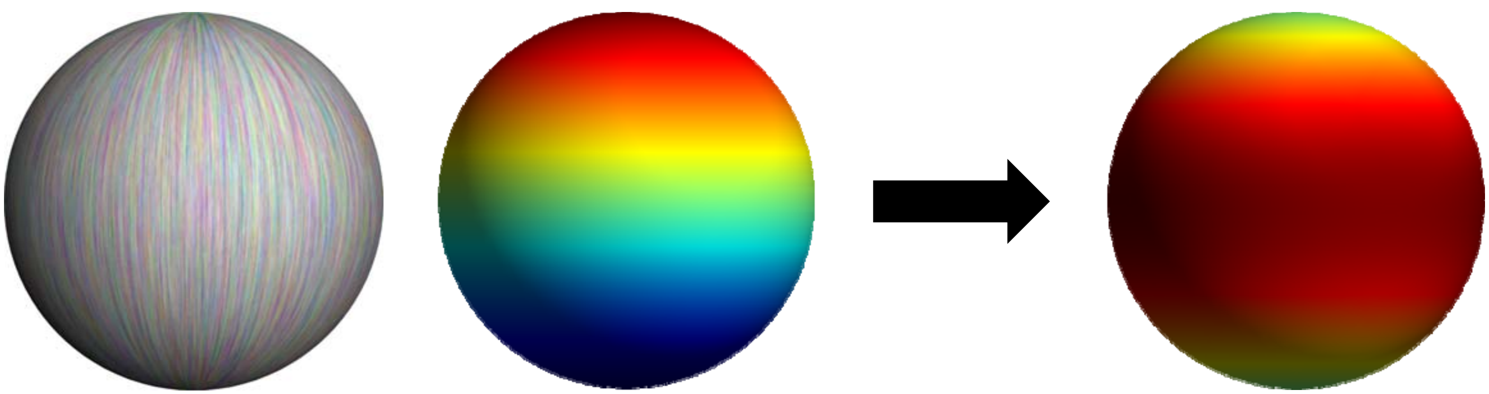
$(I - \delta t \, v_t \cdot \text{grad})\omega_{t+\delta t} = \omega_t$

VFs as Operators

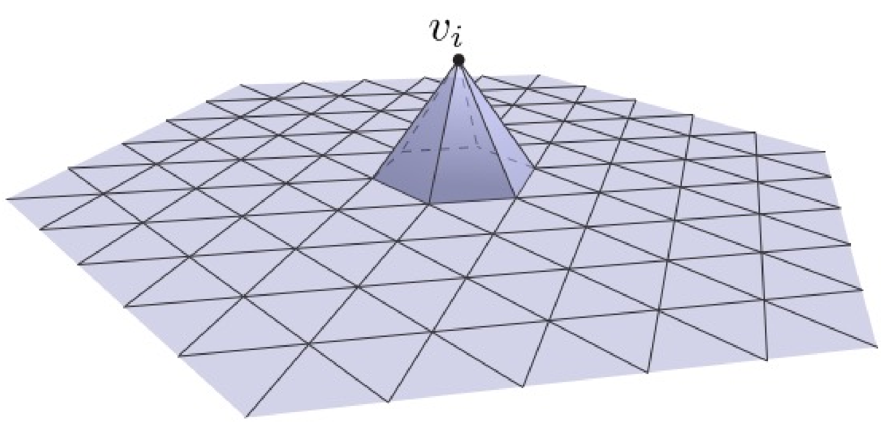
Vector fields are associated to functional operators

$$D_v = v \cdot \text{grad}$$

D_v sends functions to their directional derivatives

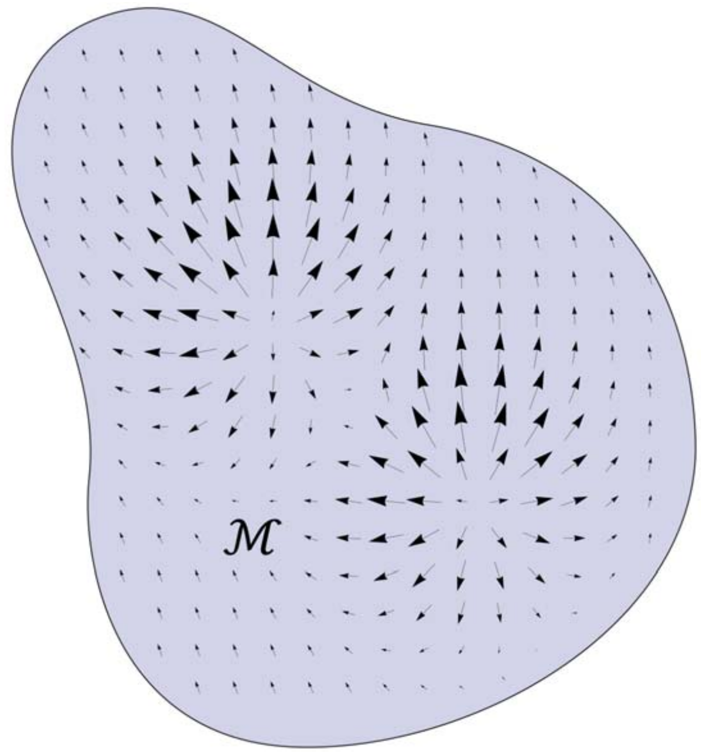


Spatial Discretization



Piecewise-linear functions

$f: \mathcal{V} \rightarrow \mathbb{R}$

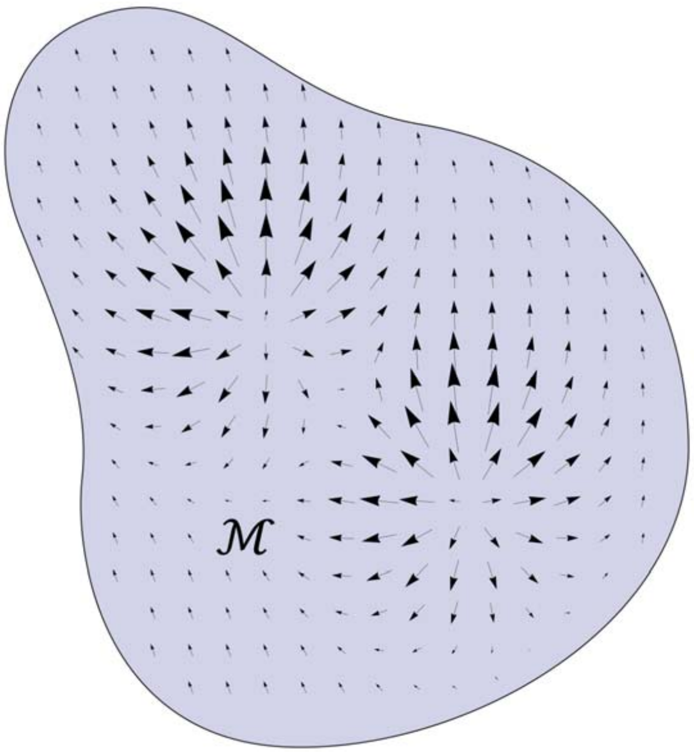


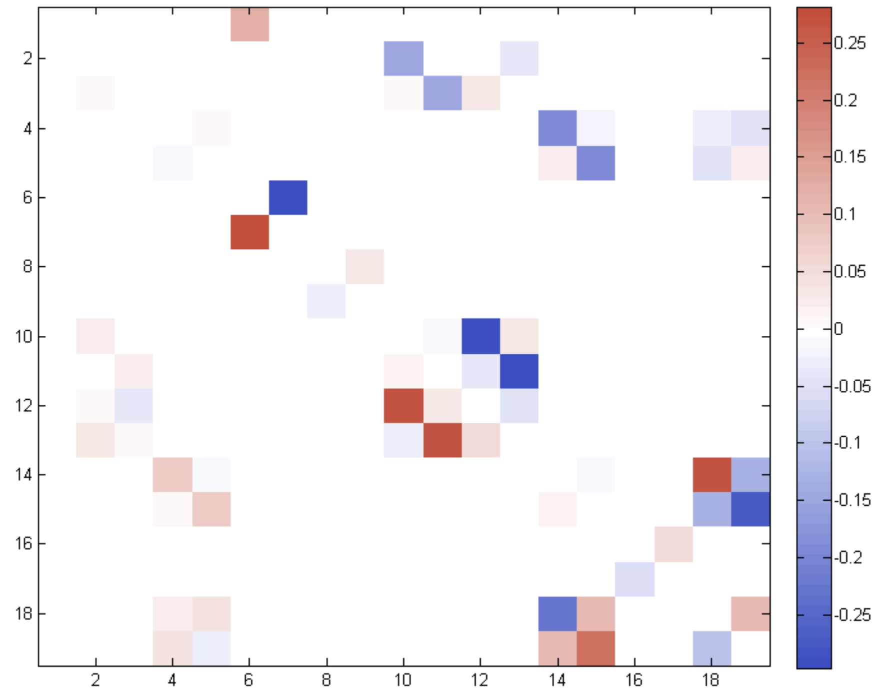
Piecewise-constant vector fields

$v: \mathcal{F} \rightarrow \mathbb{R}^3$

http://brickisland.net/cs177fa12/

Vector Fields as Operators





$v: \mathcal{F} \to \mathbb{R}^3$

$D_v: \mathbb{R}^n \to \mathbb{R}^n$

Azencot et al. SGP`13

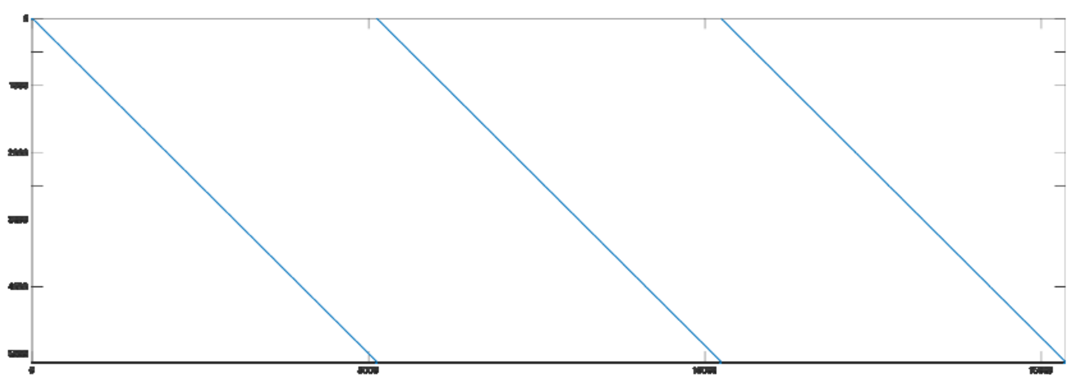
$D_v(f) = v \cdot \text{grad } f$

Derivation Operators

$D_v : \mathbb{R}^n \to \mathbb{R}^n$ is constructed by

$$D_v = \mathrm{I}_{\mathcal{V}}^{\mathcal{F}}[v]\text{grad}$$

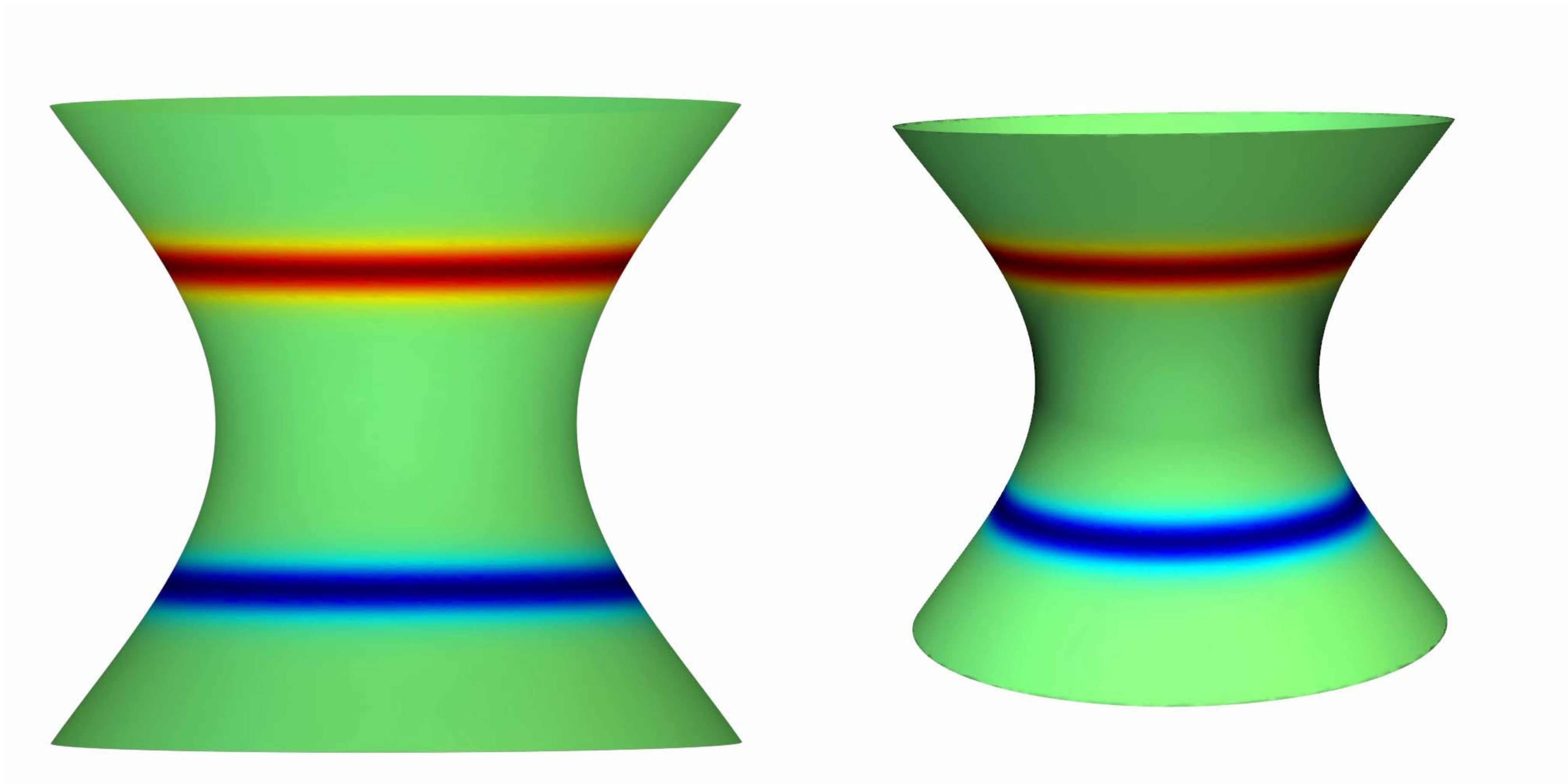
where $\mathrm{I}_{\mathcal{V}}^{\mathcal{F}}$ interpolates from faces to vertices and



$[v] \in \mathbb{R}^{m \times 3m}$

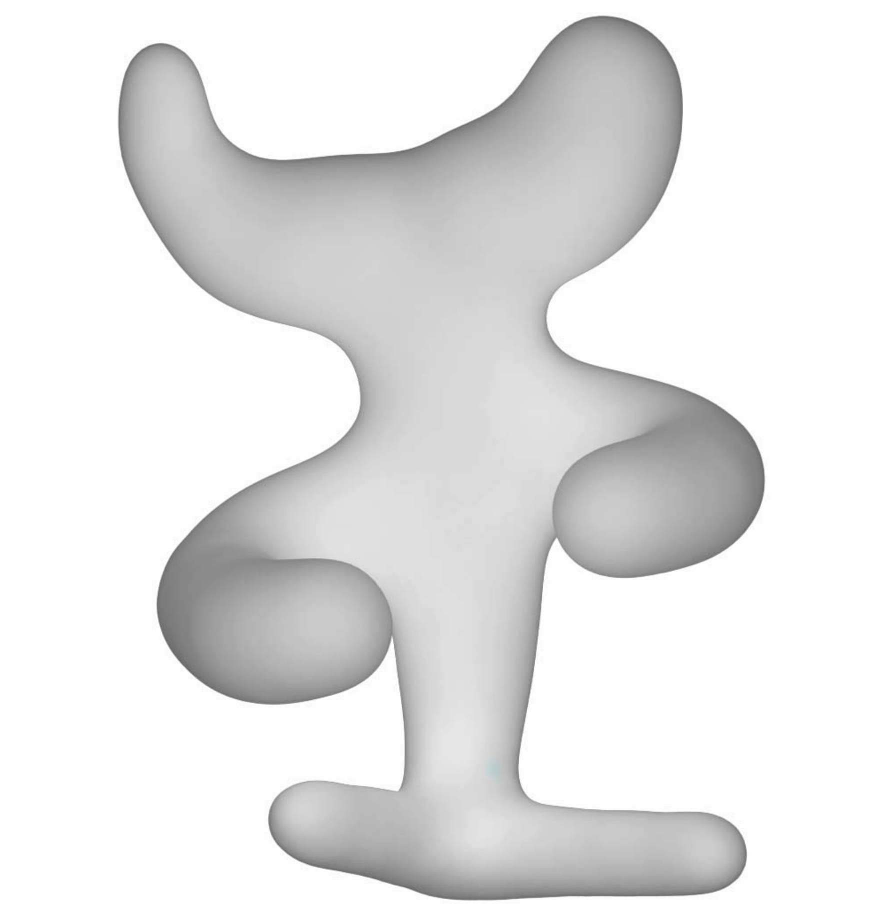
Results

Results



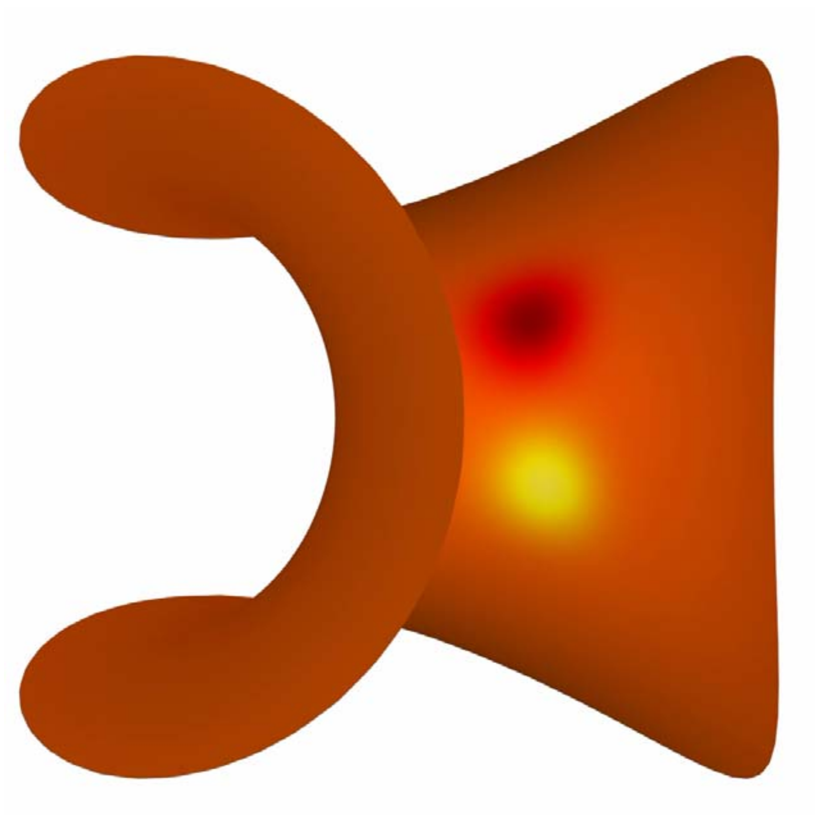
Azencot et al. SGP`14

Results



Azencot et al. SGP`14

Results



Azencot et al. SGP`14

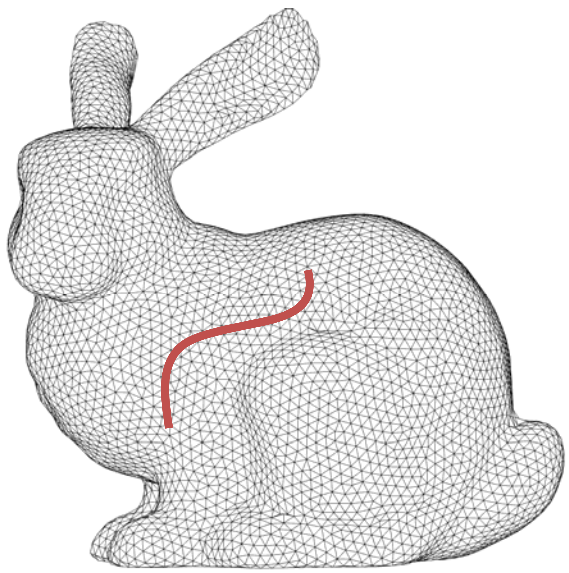
References

- “Anisotropic Diffusion in Vector Field Visualization on Euclidean Domains and Surfaces”, Diewald et al., TVCG 2000
- “An Operator Approach to Tangent Vector Field Processing”, Azencot et al., SGP 2013
- “Functional Fluids on Surfaces”, Azencot et al., SGP 2014

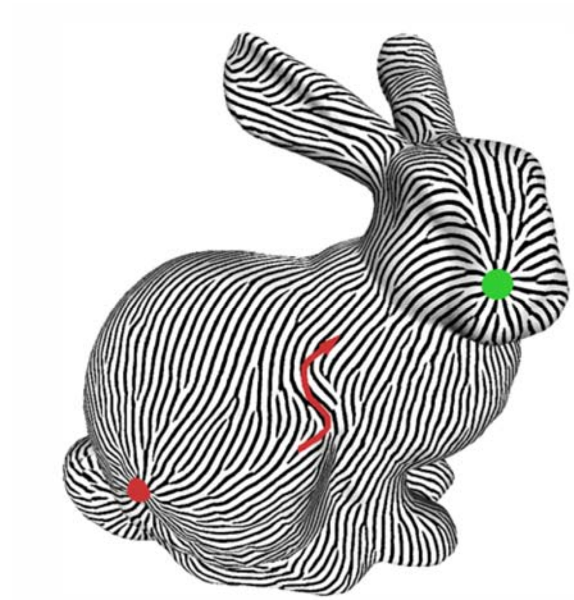
Design

[Fisher et al., 2007]

The Problem



Input

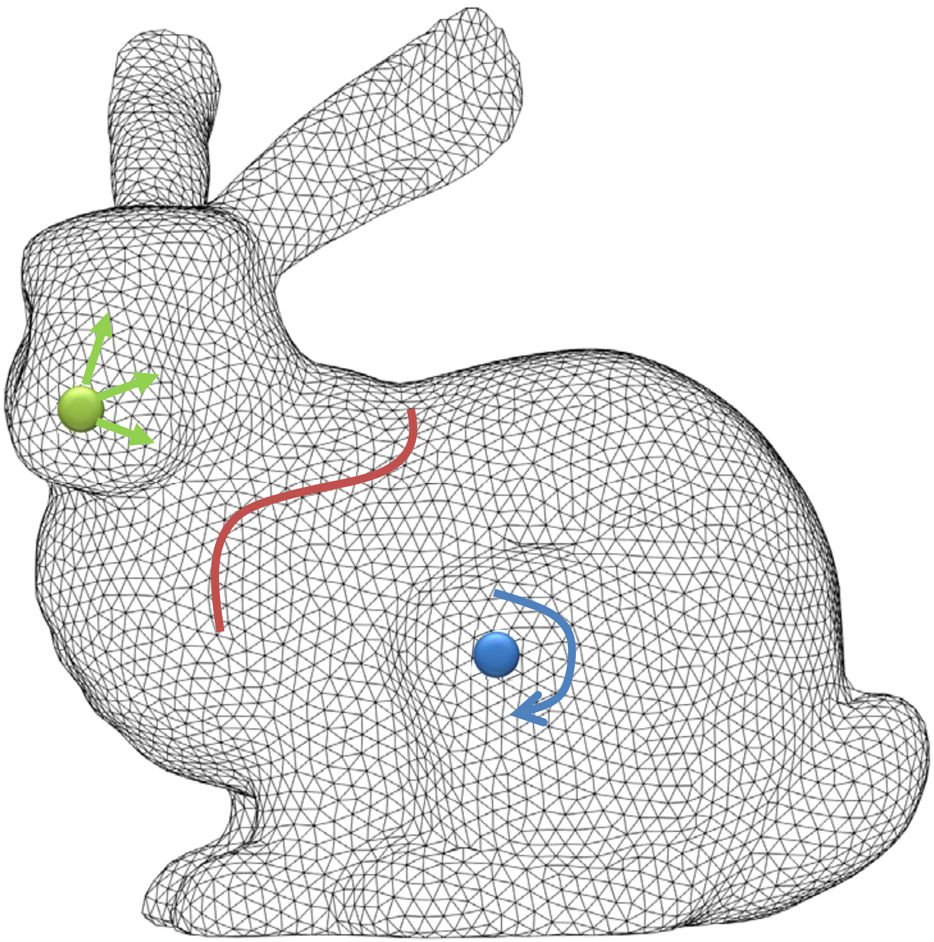


Output

- Goals
  - Intuitive modeling
  - Fast

[Fisher et al., 2007]

Design Metaphors



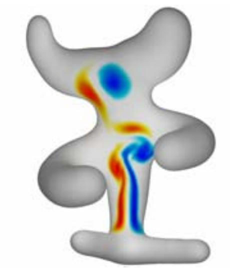
streamline curves

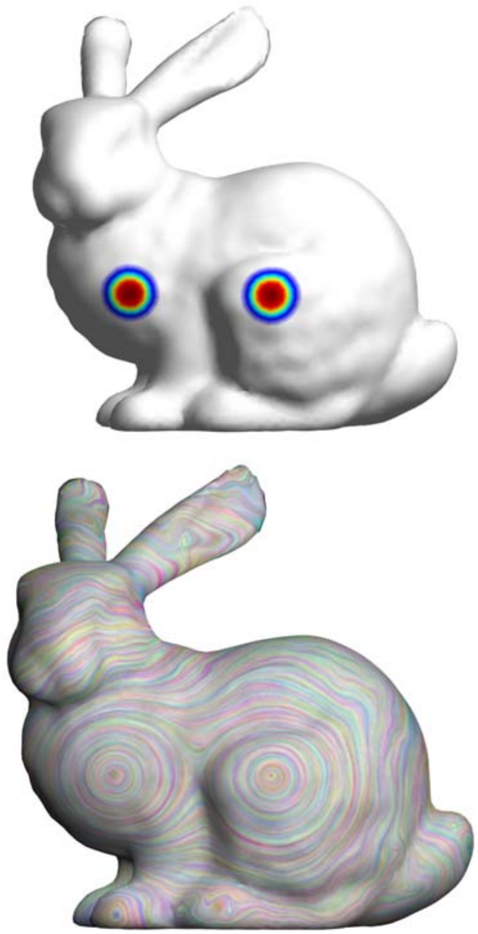
vortices

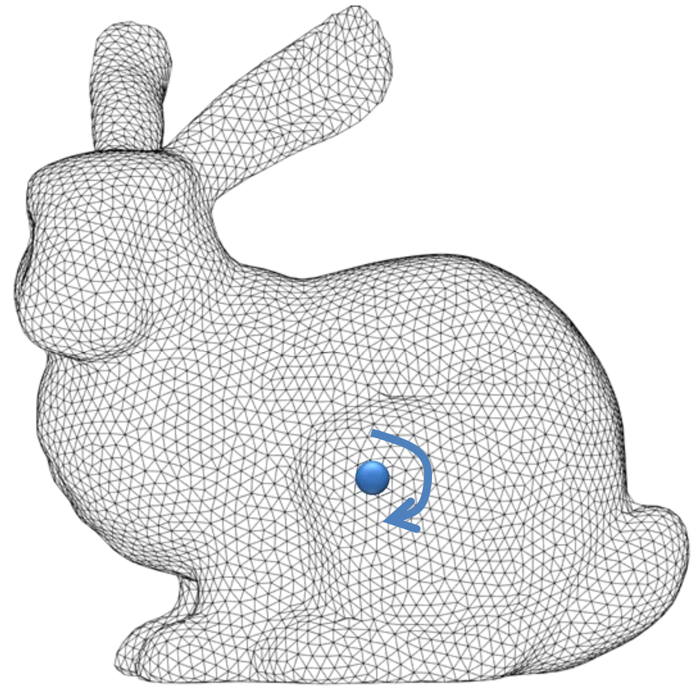
sources / sinks

optimization problem?

Specifying Vortices

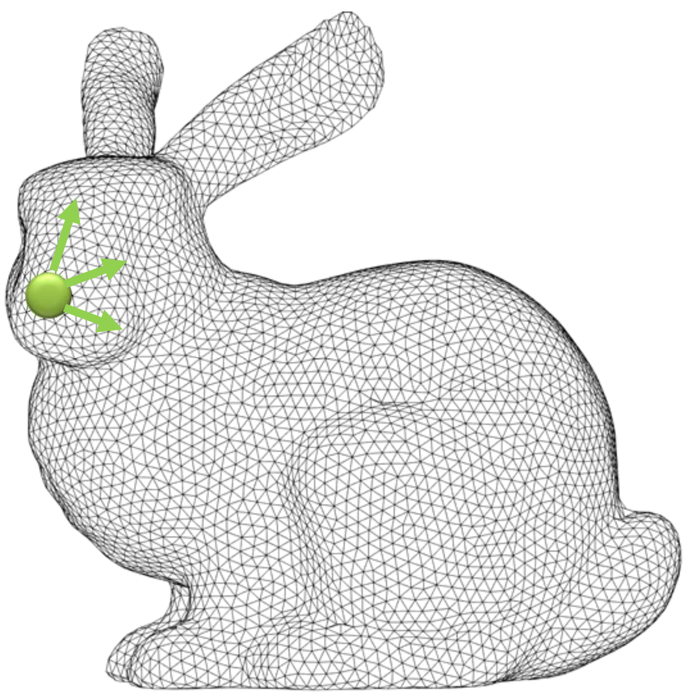




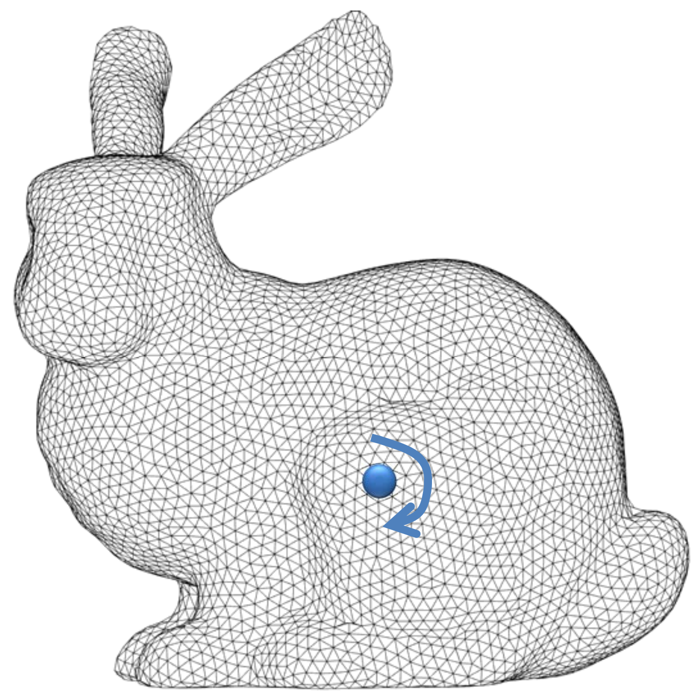


$$\omega = \text{curl}\ v$$

Specifying Sources/Sinks



$\xi = \text{div}\, v$



$\omega = \text{curl}\, v$

Hodge Decomposition Theorem

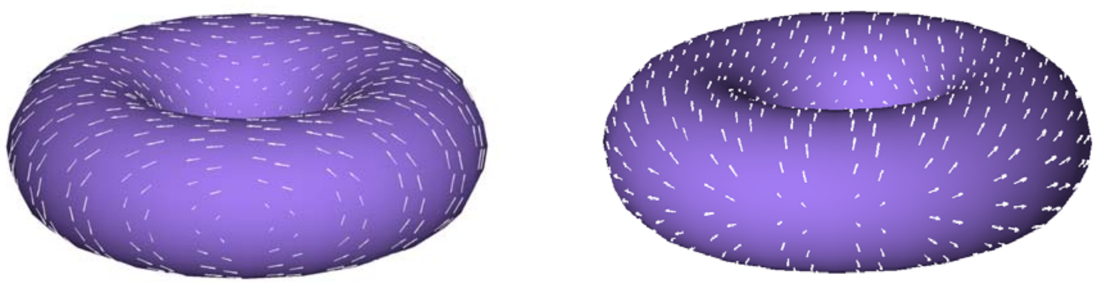
$$\xi = \mathrm{div}\, v \qquad \omega = \mathrm{curl}\, v$$

$\Delta^{-1}\omega$

Poisson equation

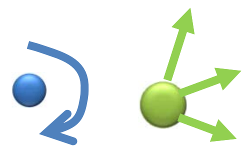
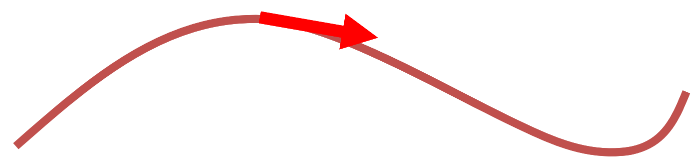
$\Delta^{-1}\xi$

Poisson equation



Vector Field Design

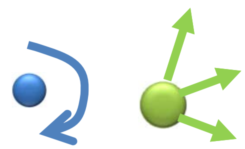
The Constraints

1. Specify **singularities** 
   Use Hodge
2. Specify pointwise **vector constraints** 
   Set variables
3. Specify a **streamline** 
   Sample tangent to curve, set variables

What are the *variables*?

Vector Field Design

The Variables

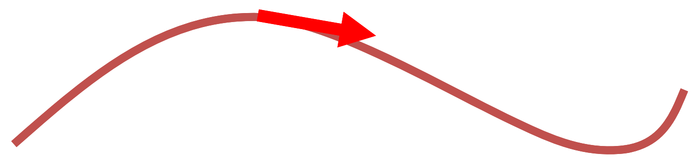
1. Specify **singularities** 

   Use Hodge

Scalar functions

2. Specify pointwise **vector constraints** 

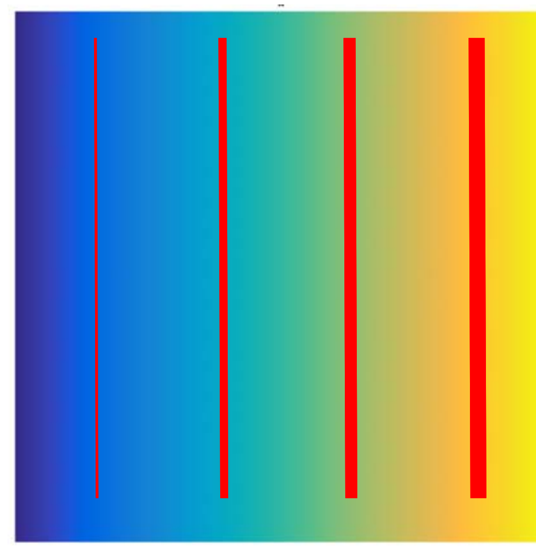
   Set variables

3. Specify a **streamline** 

   Sample tangent to curve, set variables

Vector fields as scalar functions?

Covectors



vector

$v \in \mathbb{R}^2$

covector

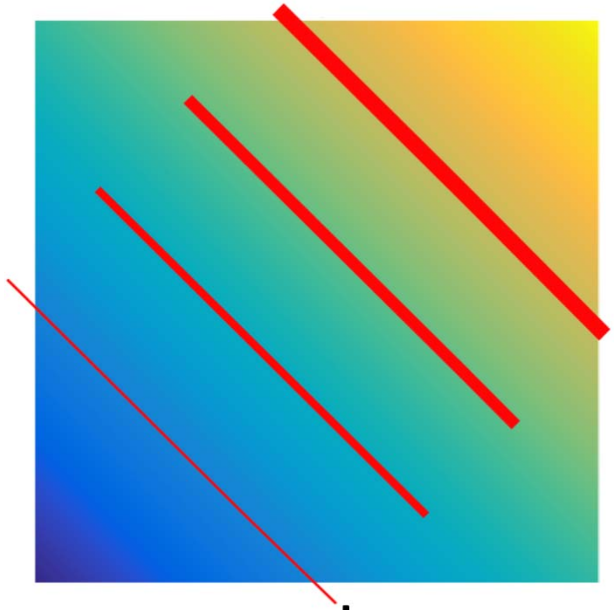
$\alpha_v : \mathbb{R}^2 \to \mathbb{R}$

$\alpha_v(u) = \langle v, u \rangle$

Covectors

vector

$v \in \mathbb{R}^2$



covector

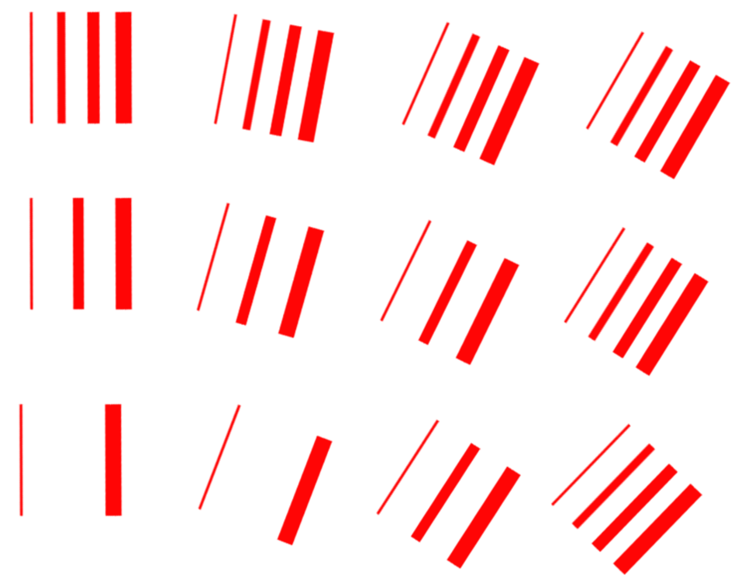
$\alpha_v : \mathbb{R}^2 \to \mathbb{R}$

$\alpha_v(u) = \langle v, u \rangle$

1-forms

Vector field

$v: \mathbb{R}^2 \to \mathbb{R}^2$

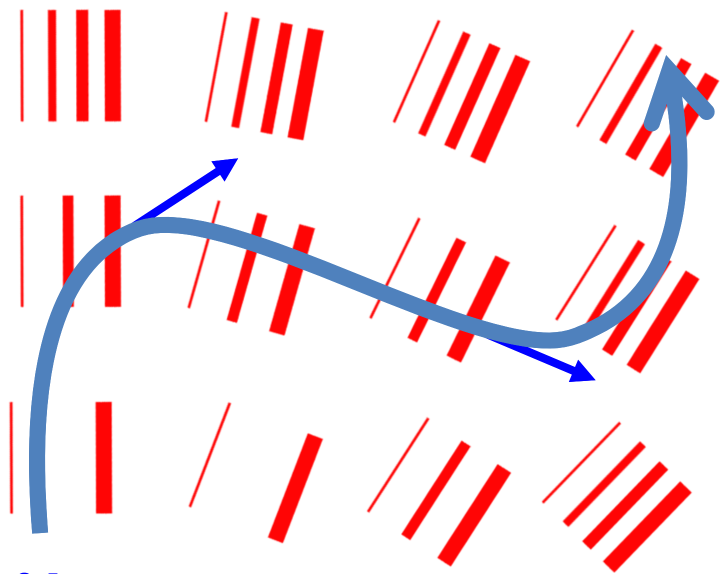


1-form

$\alpha_v: \mathbb{R}^2 \to (\mathbb{R}^2 \to \mathbb{R})$

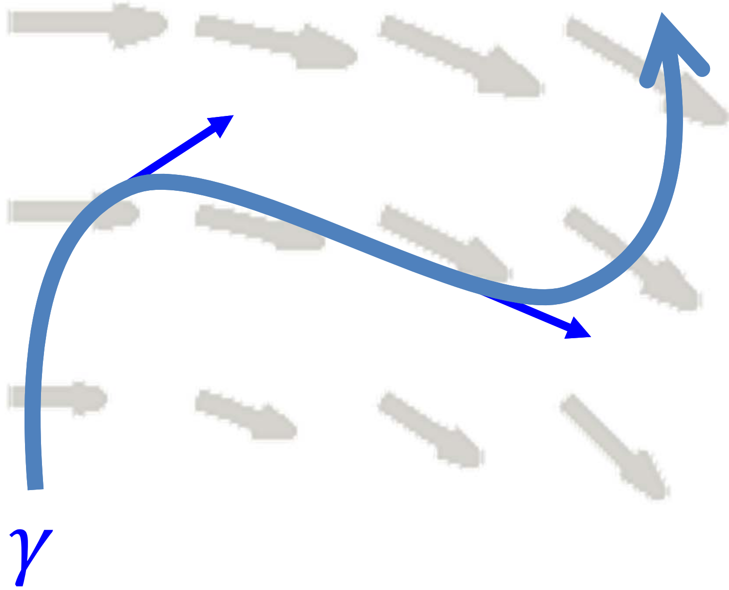
$(\alpha_v(\boldsymbol{x}))(u) = \langle v(\boldsymbol{x}), u \rangle$

Integrated 1-forms



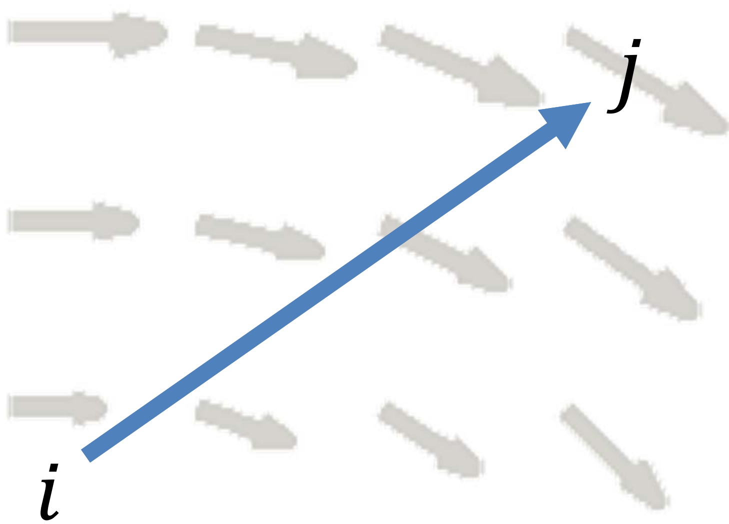
$$\int_{\gamma} \alpha_{v}\left(\frac{d\gamma}{dt}\right) = c_{\gamma}^{v}$$

Integrated 1-forms



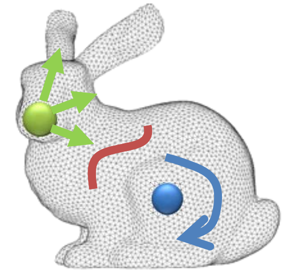
$$\int_{\gamma} \langle v, \frac{d\gamma}{dt} \rangle = c_{\gamma}^{v}$$

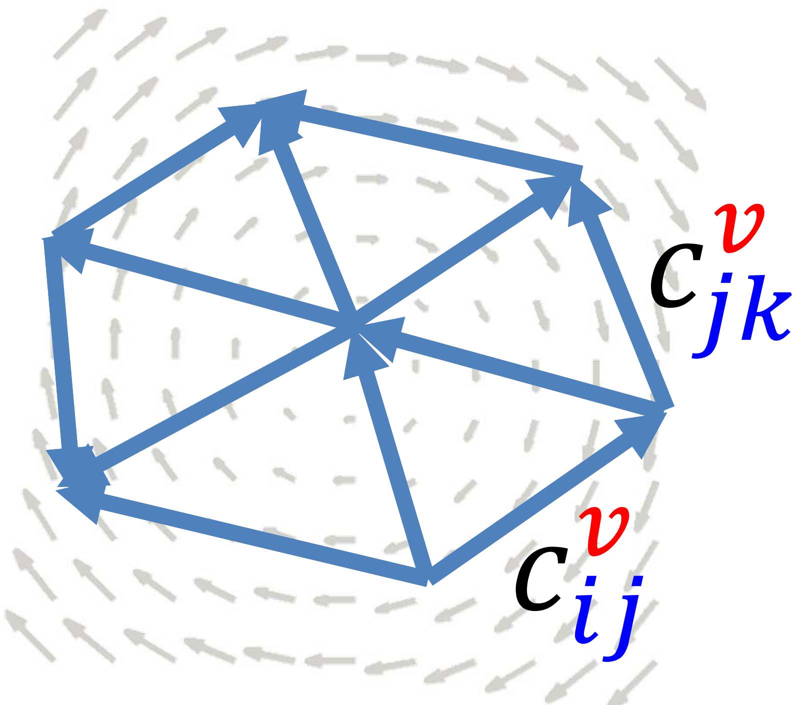
Discrete 1-forms



$$\int_{e_{ij}} \langle v, \widehat{e_{ij}} \rangle = c_{ij}^{v}$$

Discrete 1-forms





Variables:

- 1 number per edge

Constraints:

- Curl?
- Divergence?
- Pointwise?
- Streamline?

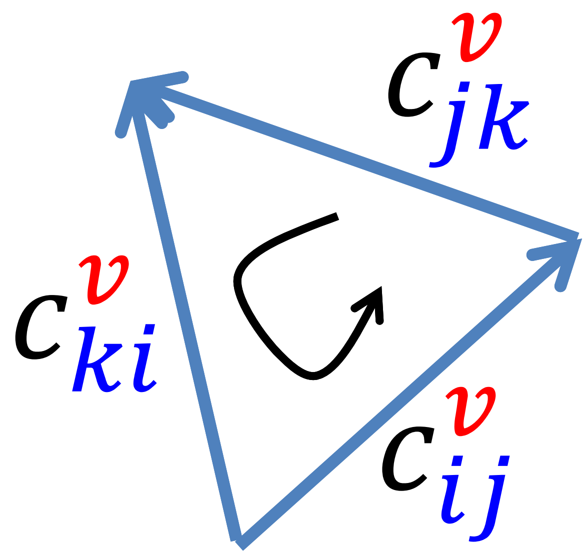
Discrete 1-forms

Curl

Stokes Theorem:

$$\int_{area} curl\ v = \int_{bdry} \langle v, tangent \rangle$$

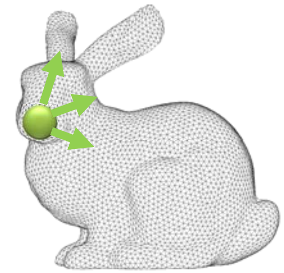
$$\int_{\Delta} curl\ v = \int_{edges} \alpha_v$$



$$\int_{\Delta_{ijk}} curl\ v = +c_{ij} + c_{jk} - c_{ki}$$

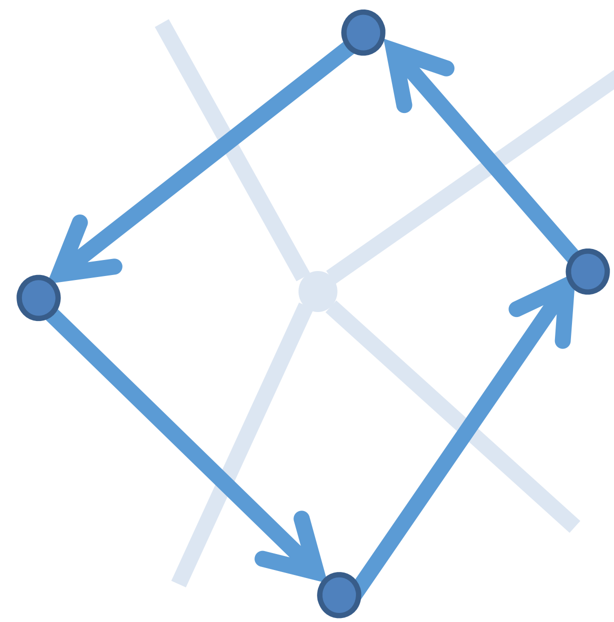
Discrete 1-forms

Divergence



Divergence Theorem:

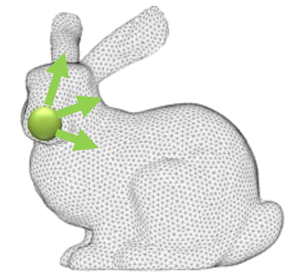
$$\int_{area} div\, v = \int_{bdry} \langle v, normal \rangle$$

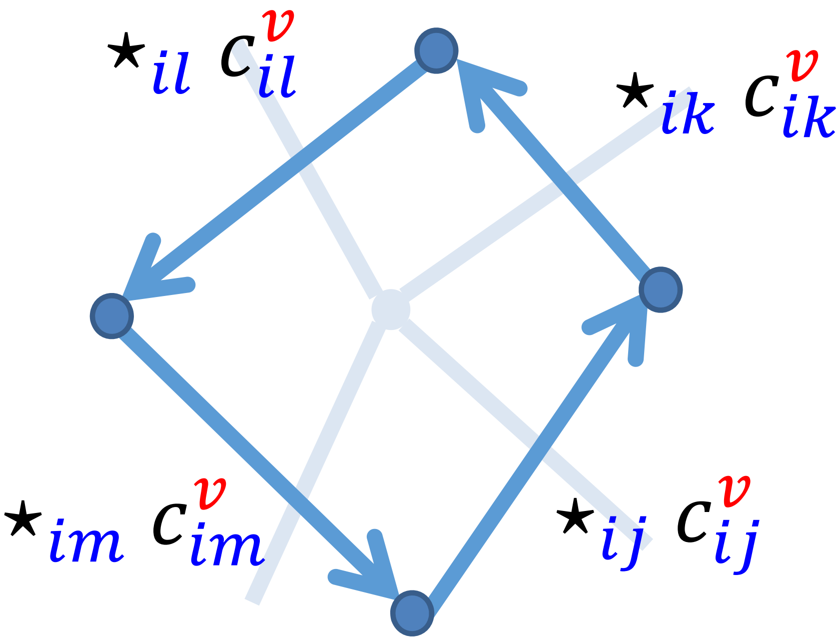


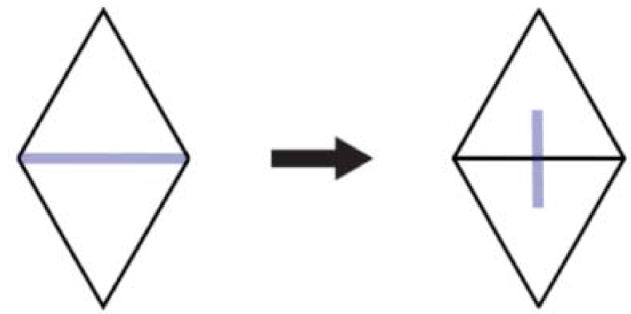
$$\int_{\times} div\, v = \int_{edges} \alpha_{Jv}$$

Discrete 1-forms

Divergence





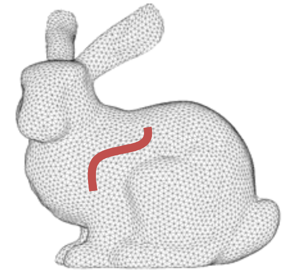


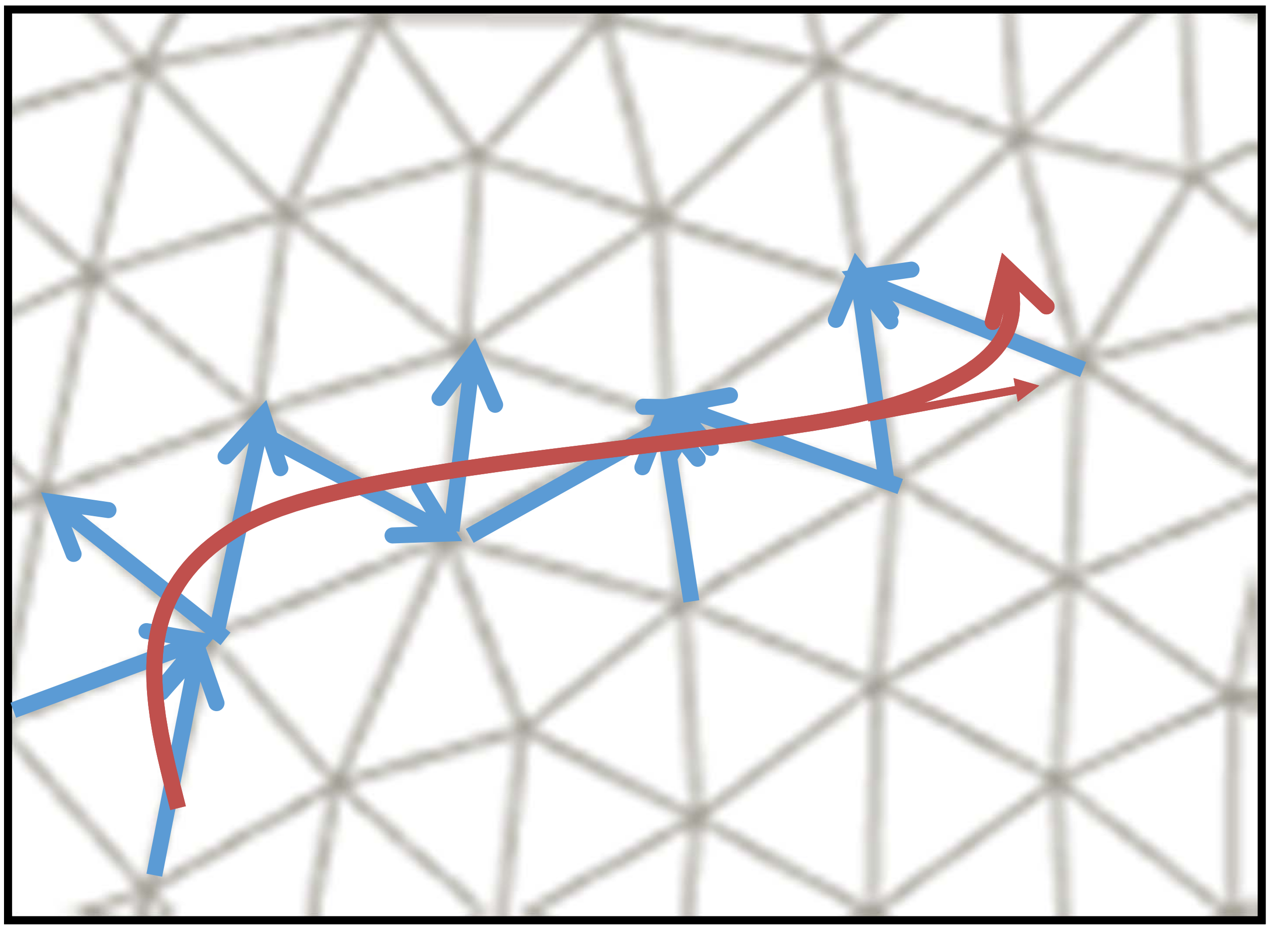
$$(\star_1)_{ee} = |e^*|/|e|$$

$$\int_{\times_i} div\ v = +\star\, c_{ij} +\star\, c_{ik} +\star\, c_{il} +\star\, c_{im}$$

Discrete 1-form

Streamline

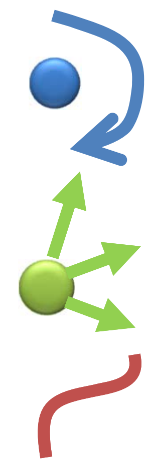




$$c_{ij} = \langle e_{ij}, \frac{d\gamma}{dt} \rangle$$

The Linear System

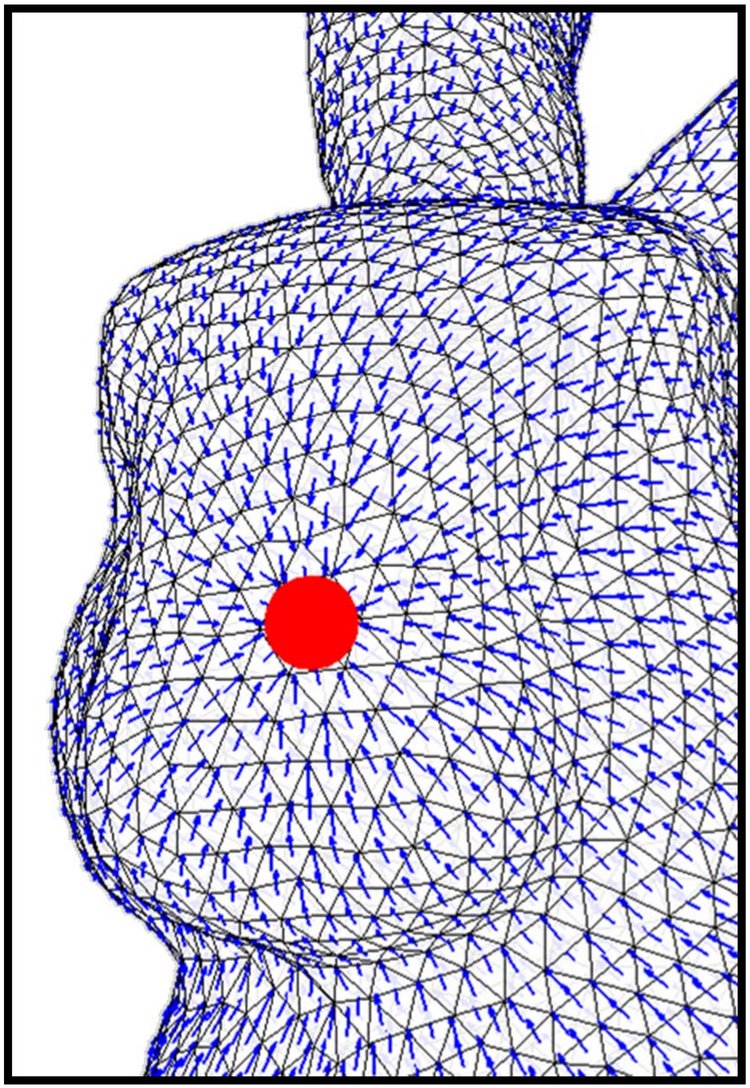
$$\begin{pmatrix} d \\ \delta \\ Z \end{pmatrix} c_e = \begin{pmatrix} r_t \\ s_v \\ c_z \end{pmatrix}$$

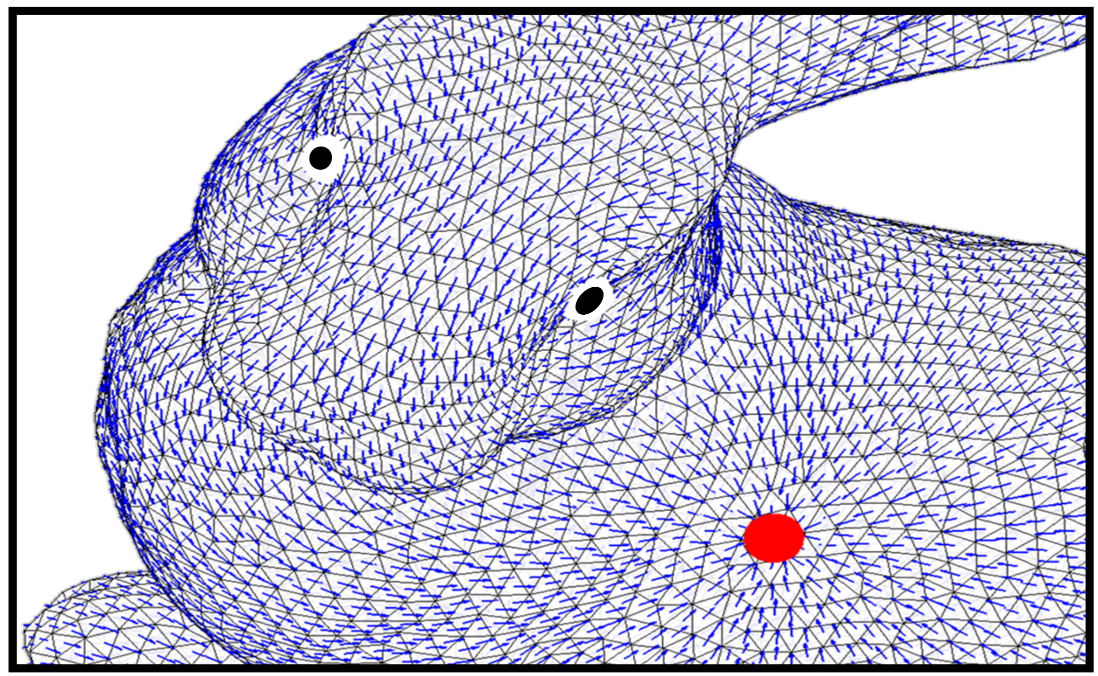


Solve as least squares system

Reconstruct vector field

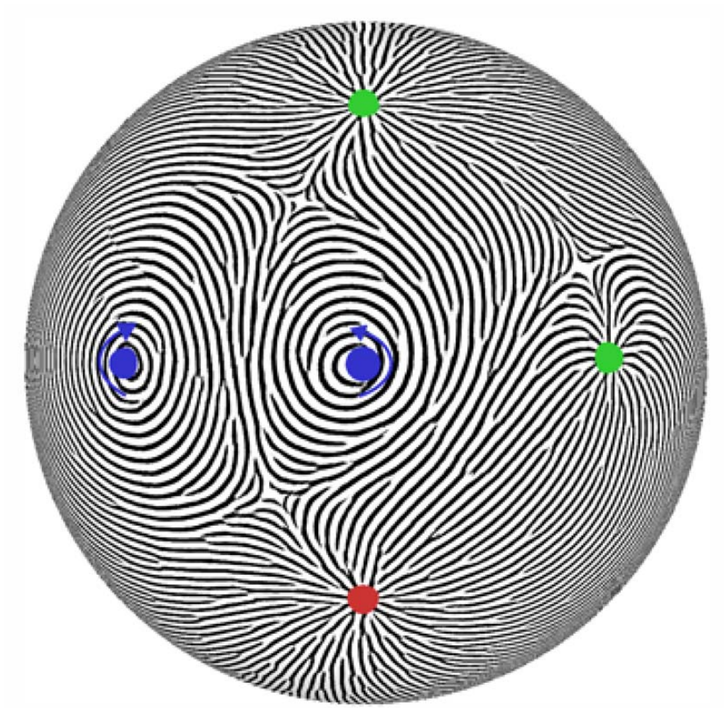
Examples





Images: Vardan Papyan

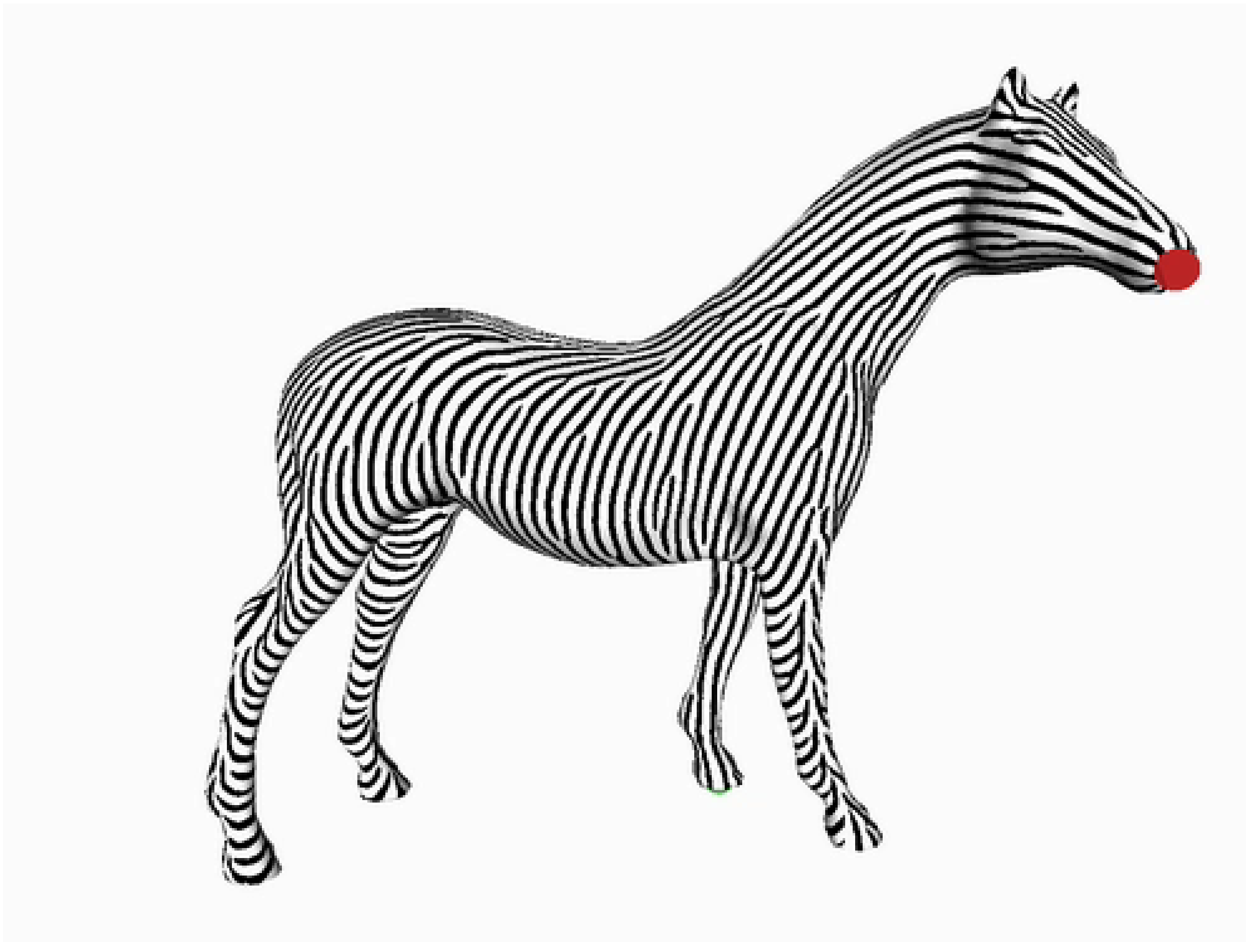
Applied to Texture Synthesis



Guaranteed:

No extra singularities

[Fisher et al., 2007]



[Fisher et al., 2007]

Applied to Texture Synthesis

[Fisher et al., 2007]

Recap

- Design vector fields by specifying divergence, curl and pointwise constraints
- Variables are scalar function on edges
- Solve with weighted least squares

References

- "Design of tangent vector fields", Fisher et al., SIGGRAPH 2007

Design

Closing

Closing Remarks

- Tangent vector fields pose many challenges
  - Convenient representation
  - Efficient optimization

- Every representation has its advantages and disadvantages

- Many open research questions
  - Await you!

Thank You!