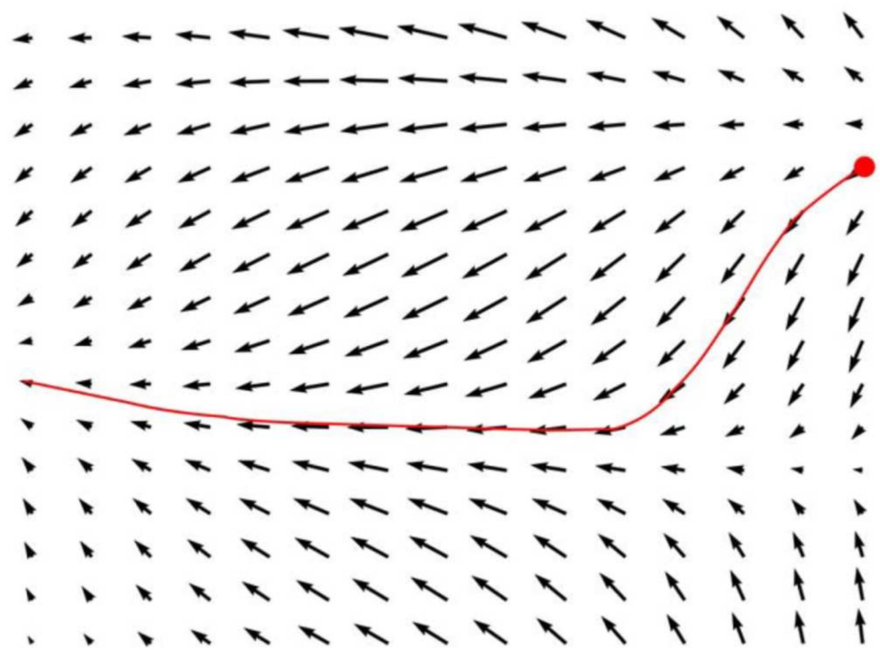
Vector Field Data: Introduction

What is a Vector Field?

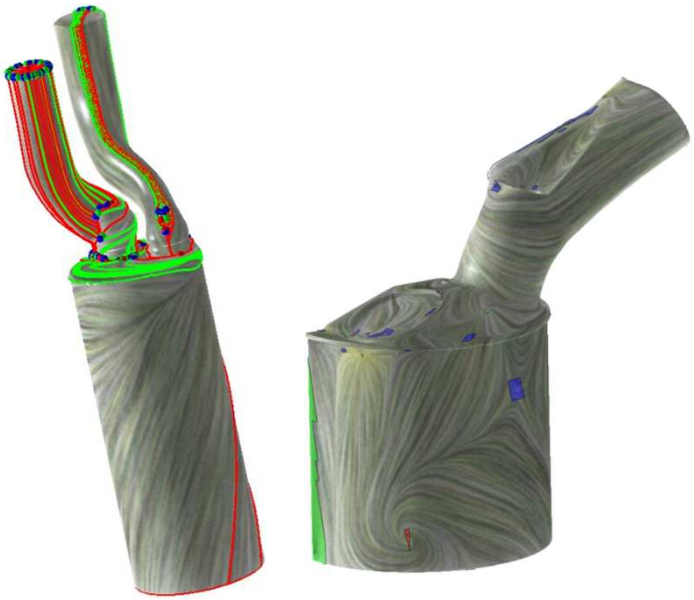


$$\frac{d\varphi(x)}{dt} = V(x)$$

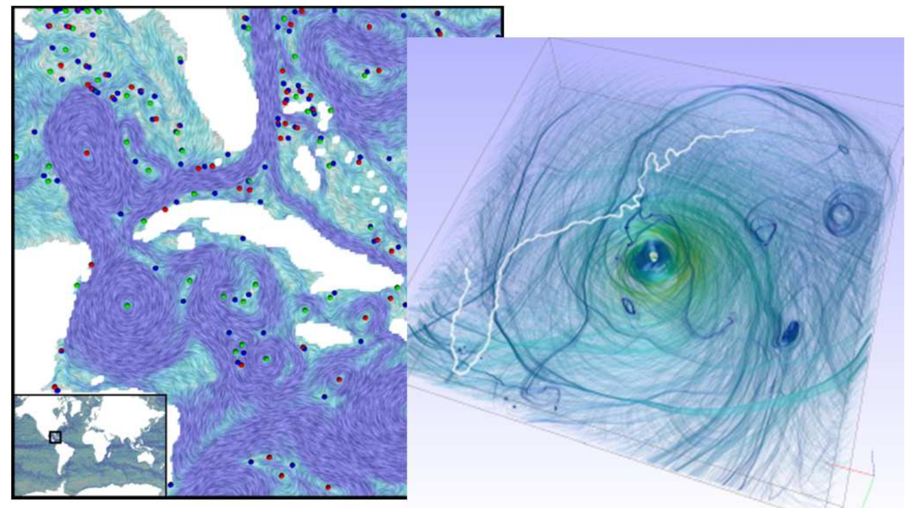
Its solution gives rise to a “flow”.

Why Is It Important?

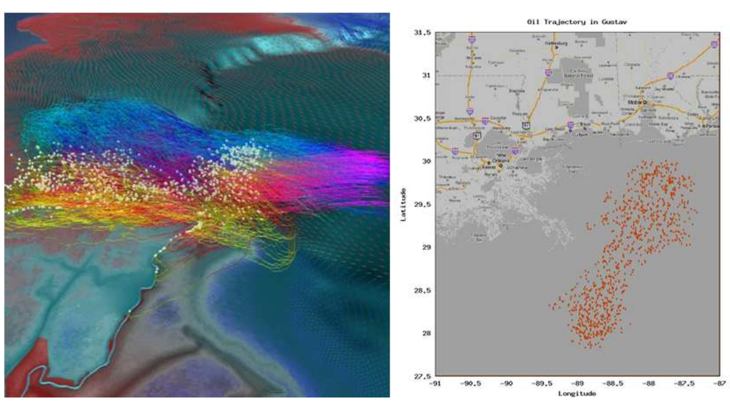
Vector Fields in Engineering and Science



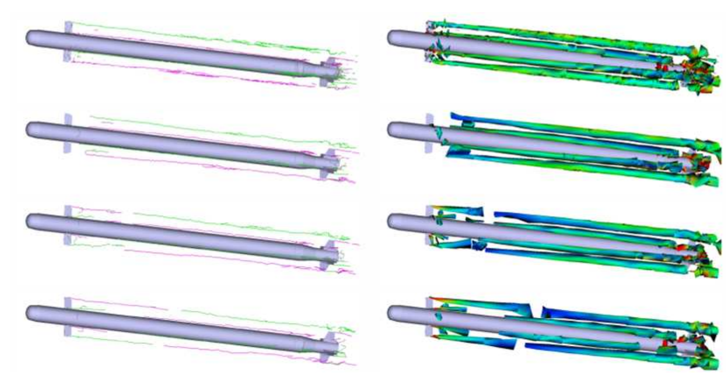
Automotive design
[**Chen** et al. TVCG07,TVCG08]



Weather study [Bhatia and Chen et al. TVCG11]

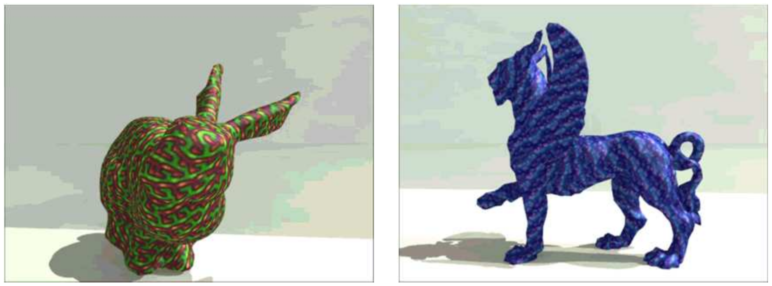


Oil spill trajectories [Tao et al. EMI2010]

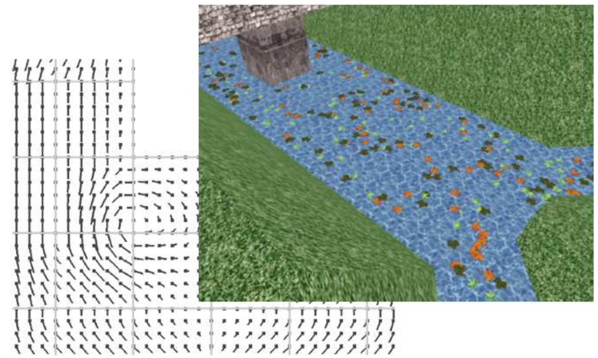


Aerodynamics around missiles [Kelly et al. Vis06]

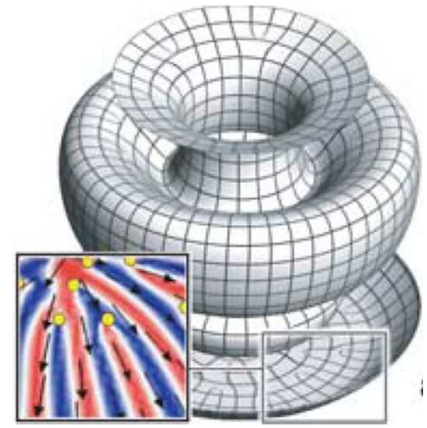
Vector Field Design in Computer Graphics



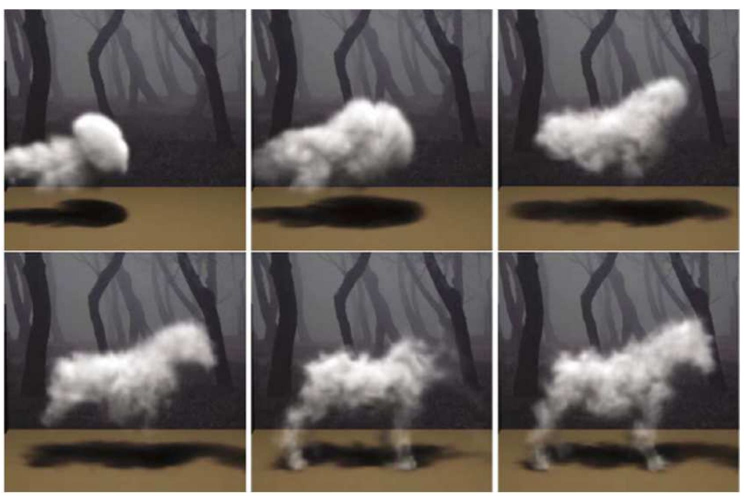
Texture Synthesis [**Chen** et al. TVCG11b]



River simulation [Chenney SCA2004]

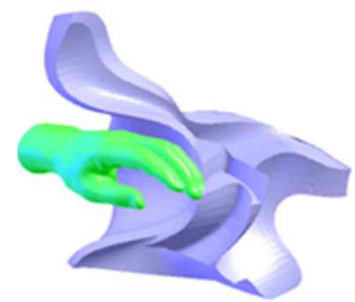


Parameterization
[Ray et al. TOG2006]



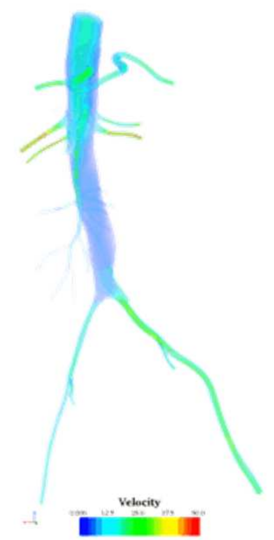
Smoke simulation [Shi and Yu TOG2005]

Painterly Rendering [Zhang et al. TOG2006]



Shape Deformation
[von Funck et al. 2006]

Flow Data

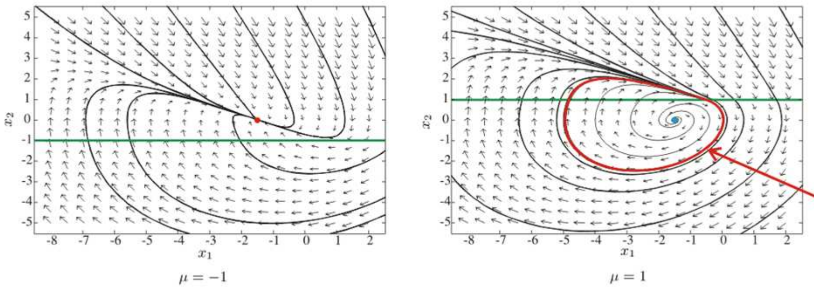


Source: simtk.org

Data sources:

- flow simulation:
  - airplane- / ship- / car-design
  - weather simulation (air-, sea-flows)
  - medicine (blood flows, etc.)
- flow measurement:
  - wind tunnels, water channels
  - optical measurement techniques
- flow models (analytic):
  - differential equation systems (dynamic systems)

Source: speedhunter.com



$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_2 - \tfrac{3}{2}|x_2 - \mu| - x_1$$

equilibrium: $x_1 = -\frac{3}{2}|\mu|, \quad x_2 = 0$

limit cycle (attracting)

Source: zfm.ethz.ch

Flow Data

Simulation:

- flow: estimate (partial) differential equation systems (i.e. a model)
- set of samples (n-dims. of data), e.g., given on a curvilinear grid
- most important primitive: tetrahedron and hexahedron (cell)
- could be adaptive grids

Analytic:

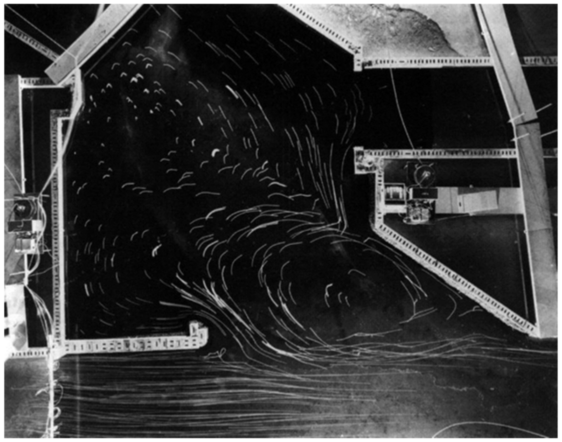
- flow: analytic formula, differential equation systems $d\boldsymbol{x}/dt$ (dynamical system)
- evaluated where ever needed

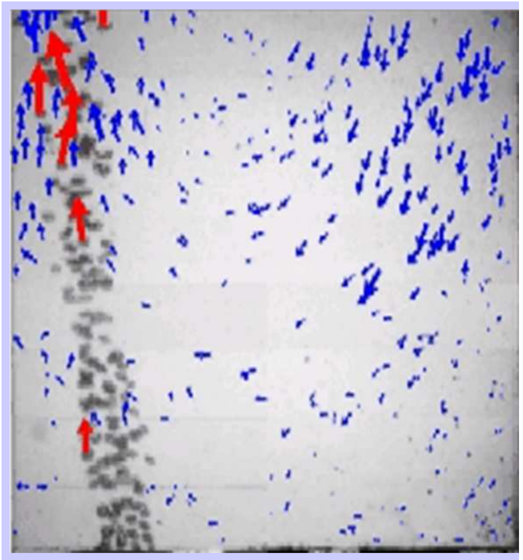
Measurement:

- vectors: taken from instruments, often computed on a uniform grid
- optical methods + image recognition, e.g.: PIV (particle image velocimetry)

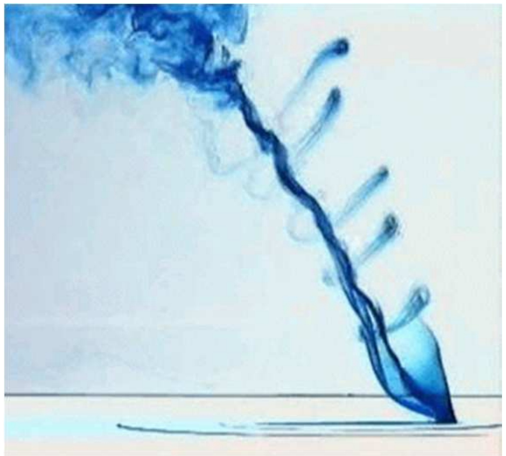
Flow Data Via Measurement

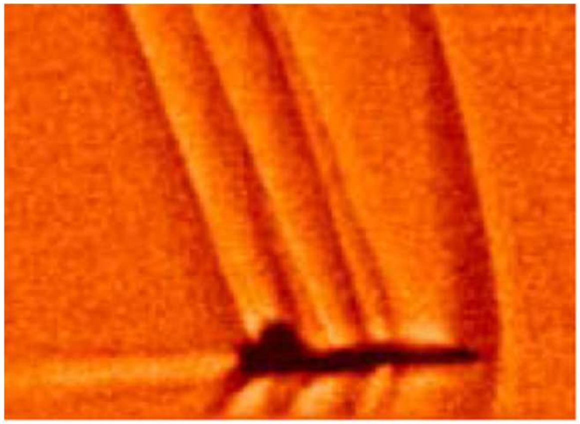
- Injection of dye, smoke, particles

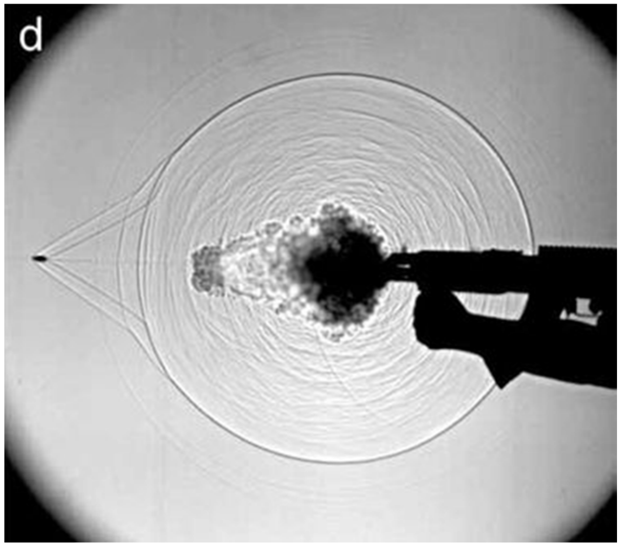




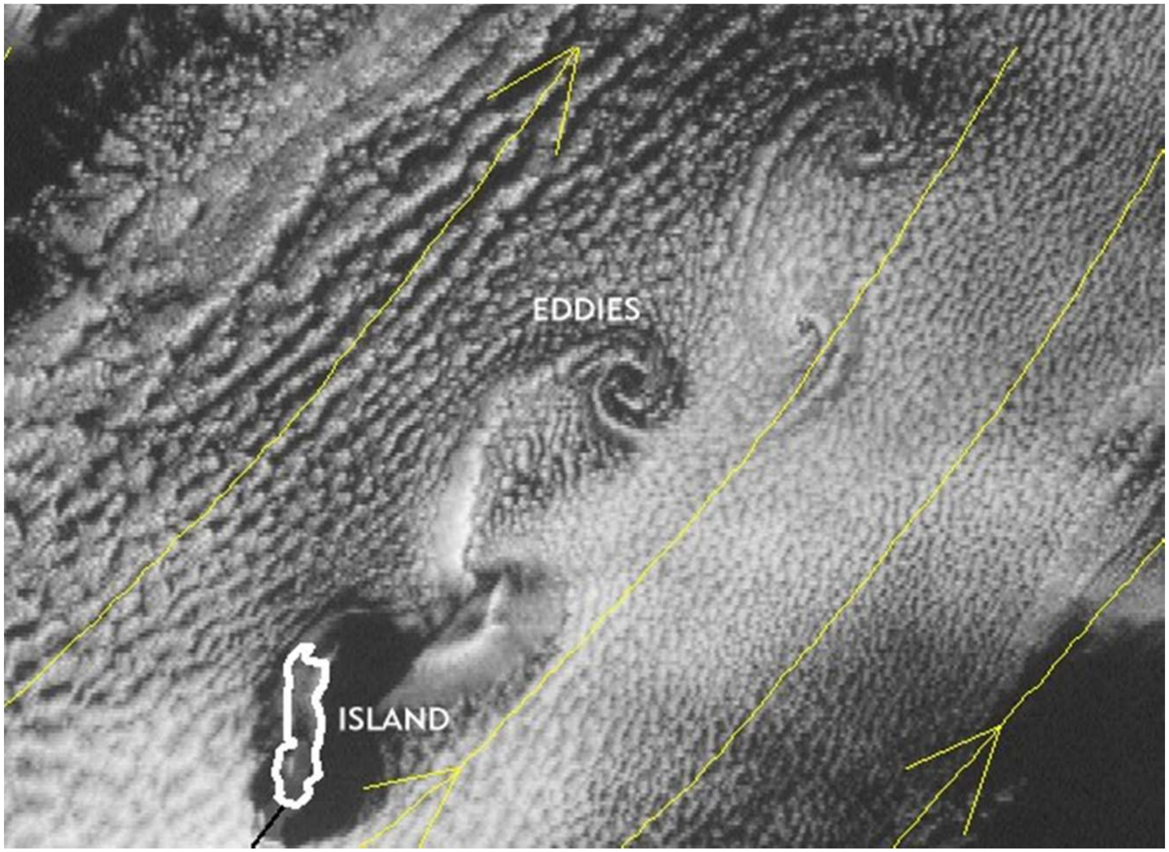
- Optical methods:
  - transparent object with complex distribution of light refraction index
- Streaks, shadows







Large Scale Dying

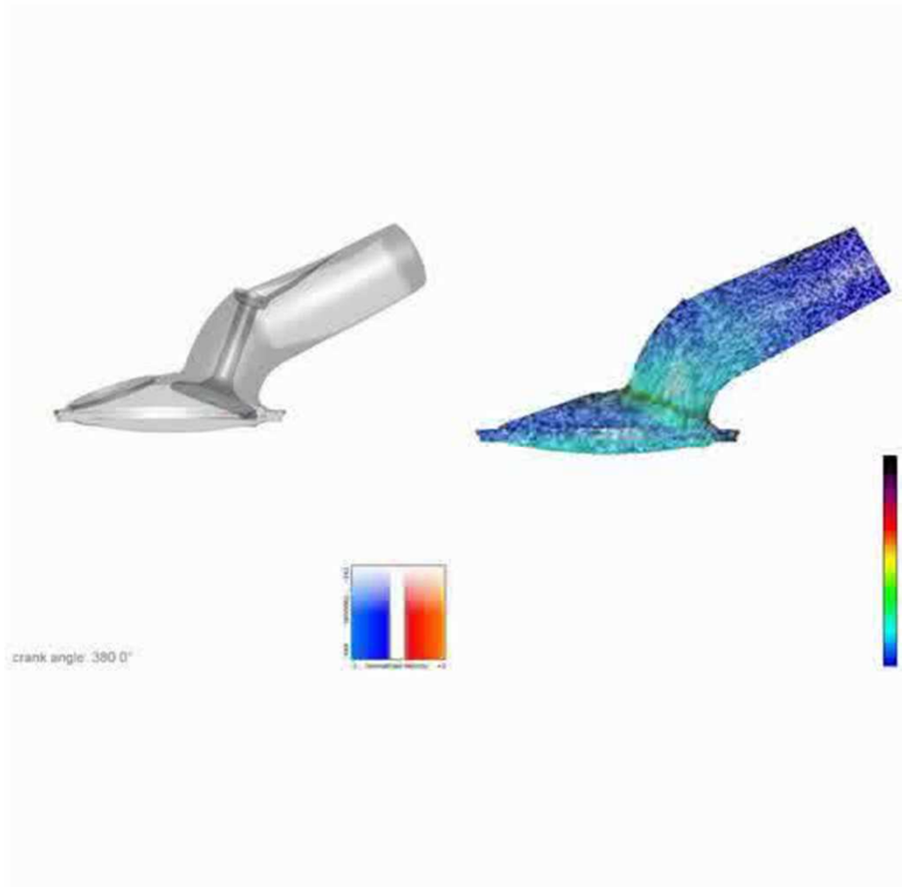


Source: weathergraphics.com

Source: ishtarsgate.com

Flow Data Via Simulations

- We often analyze **C**omputational **F**luid **D**ynamics (CFD) simulation data
- CFD is the discipline of predicting flow behavior, quantitatively
- data is (often) the result of a simulation of flow through or around an object of interest

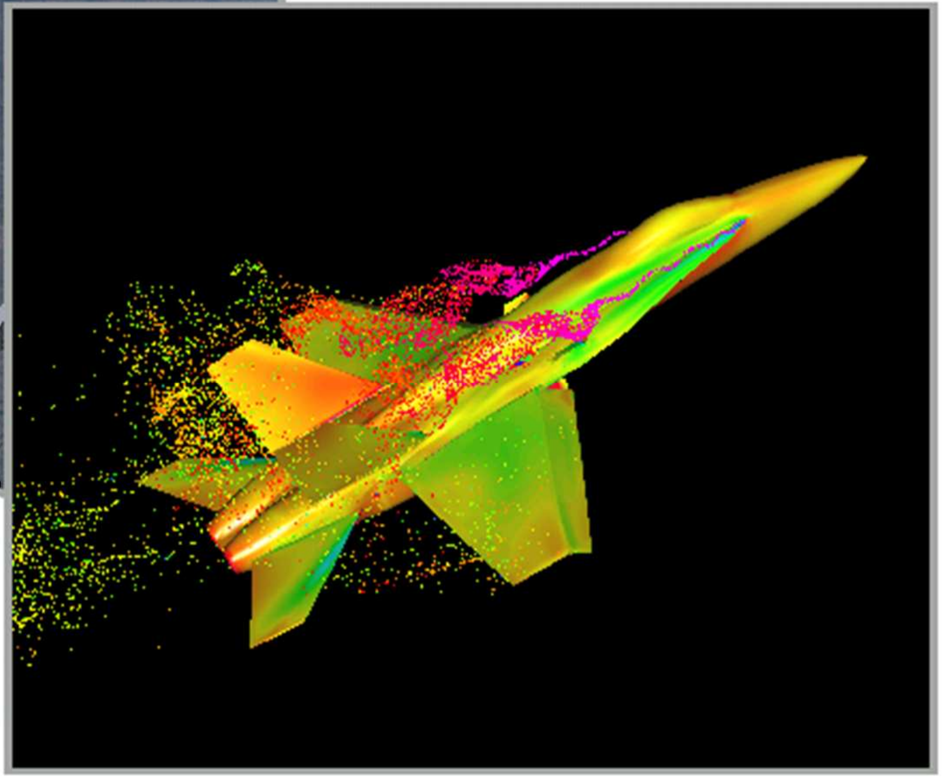


some **characteristics** of CFD data:

- large, often gigabytes
- Unsteady, i.e. time-dependent
- unstructured, adaptive resolution grids
- Smooth field

Comparison with Reality

Experiment

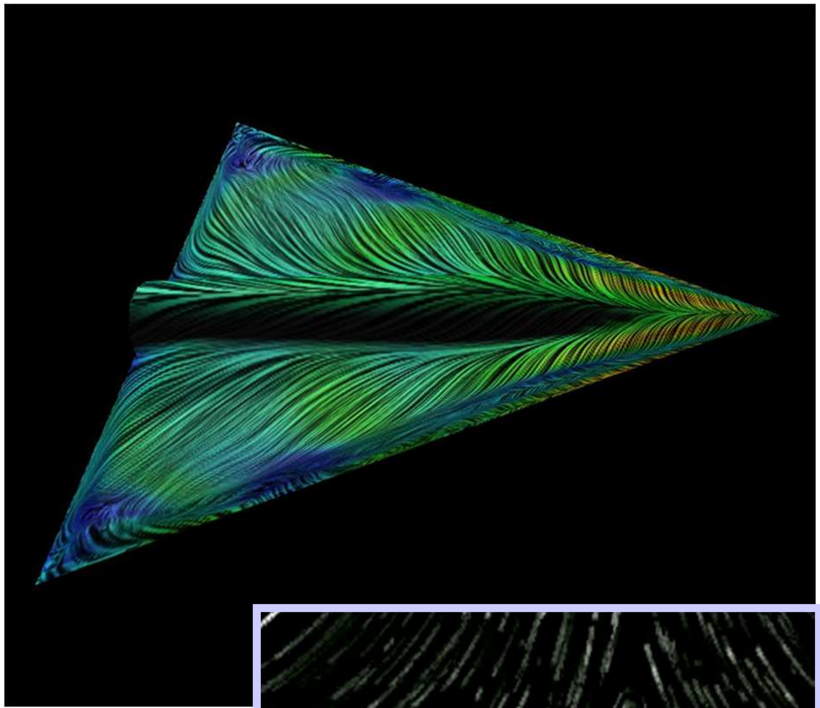


Simulation

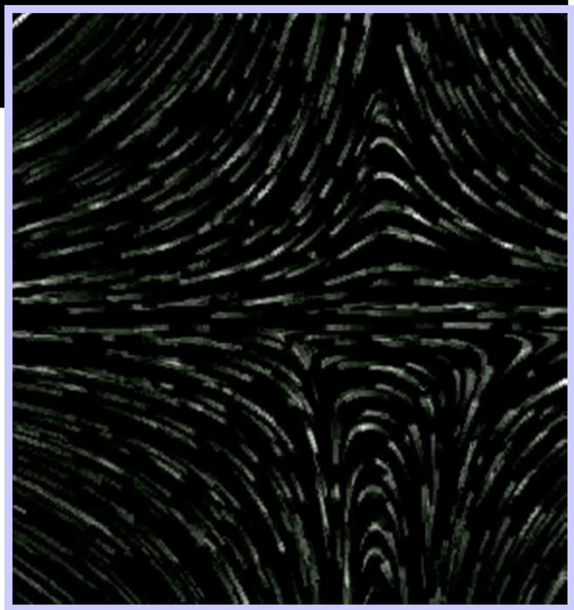
Dimensions: 2D vs. 2.5D/Surfaces vs. 3D

- 2D flow
  - $f\colon R^2 \to R^2$, i.e. $\vec{v} = (vx, vy)$ in a plane
  - analytic, flow layers (2D section through 3D)
- 2.5D, i.e. surface flow (embedded in 3D)
  - 3D flows ***around*** obstacles (i.e. on the surface of obstacles)
  - $f\colon M \to R^3$, i.e. $\vec{v} = (vx, vy, vz)$ confined on the tangent plane
  - locally 2D
  - Simulation, synthetic
- 3D flow visualization
  - $f\colon R^3 \to R^3$, i.e. $\vec{v} = (vx, vy, vz)$ in a volume
  - simulations, 3D models

2D/Surfaces/3D – Examples

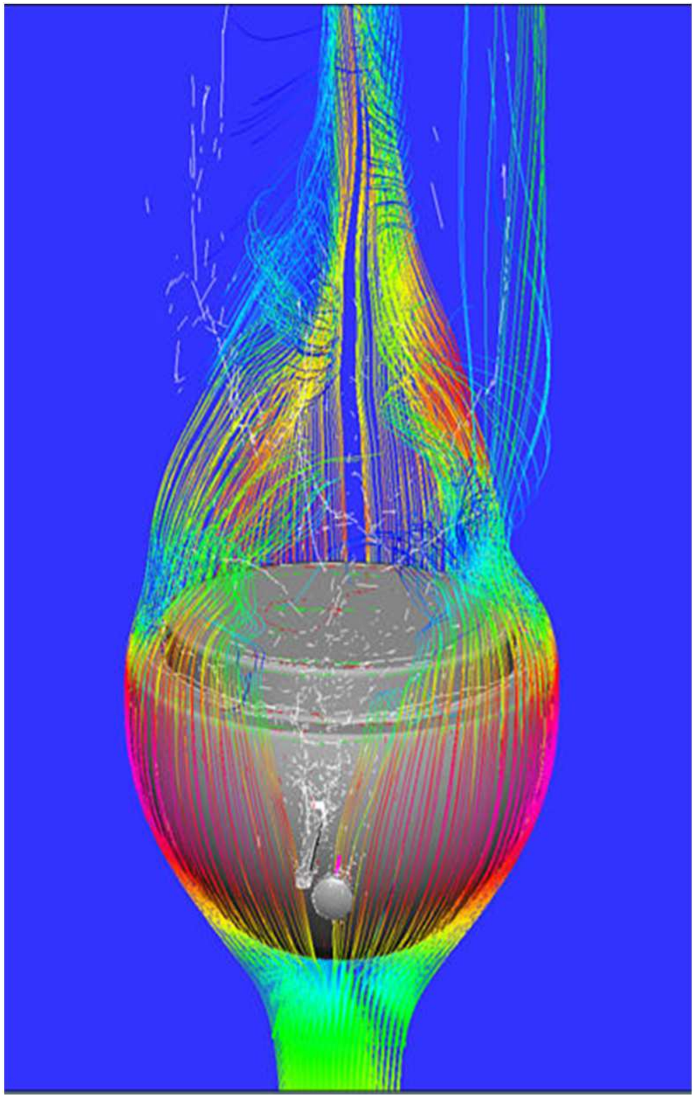


Surface



2D

3D



Steady vs. Time-dependent

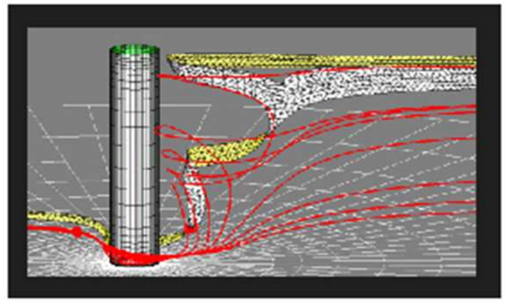
Steady (time-independent) flows:

- flow itself constant over time
- $\mathbf{v}(\mathbf{x})$, e.g., laminar flows
- well understood behaviors
- simpler case for visualization and analysis

Time-dependent (unsteady) flows:

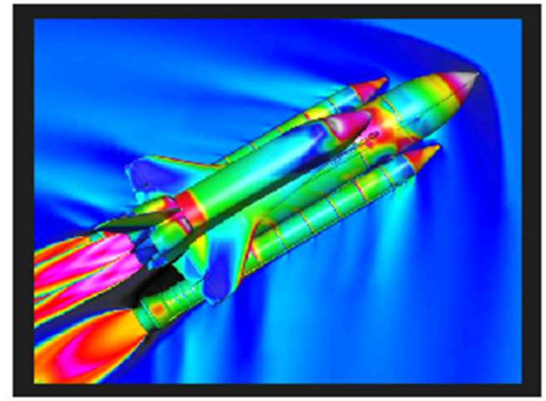
- flow itself changes over time
- $\mathbf{v}(\mathbf{x},t)$, e.g., combustion flow, turbulent flow
- more complex cases
- no uniform theory to characterize them yet!

Time-independent (steady) Data



Single Zone
100K Nodes
4 MB

(1985)



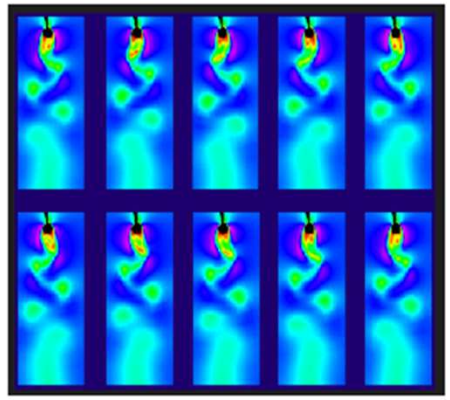
128 Zones
30M Nodes
1080 MB

(1996)

- Dataset sizes over years:

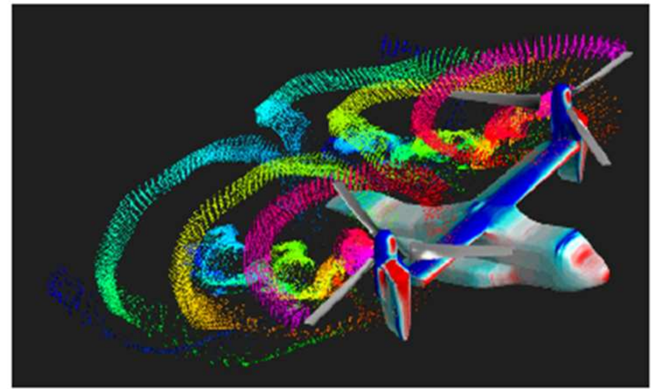
Data set name and year	Number of vertices	Size (MB)
McDonnell Douglas MD–80 '89	230,000	13
McDonnell Douglas F/A–18 '91	900,000	32
Space shuttle launch vehicle '90	1,000,000	34
Space shuttle launch vehicle '93	6,000,000	216
Space shuttle launch vehicle '96	30,000,000	1,080
Advanced subsonic transport '98	60,000,000	2,160
Army UH–60 Blackhawk '99	100,000,000	~4,000

Time-dependent (unsteady) Data



Single Zone
128K Nodes
1 GB

(1990)



25 Zones (9 Moving)
2.8M Nodes
300 GB

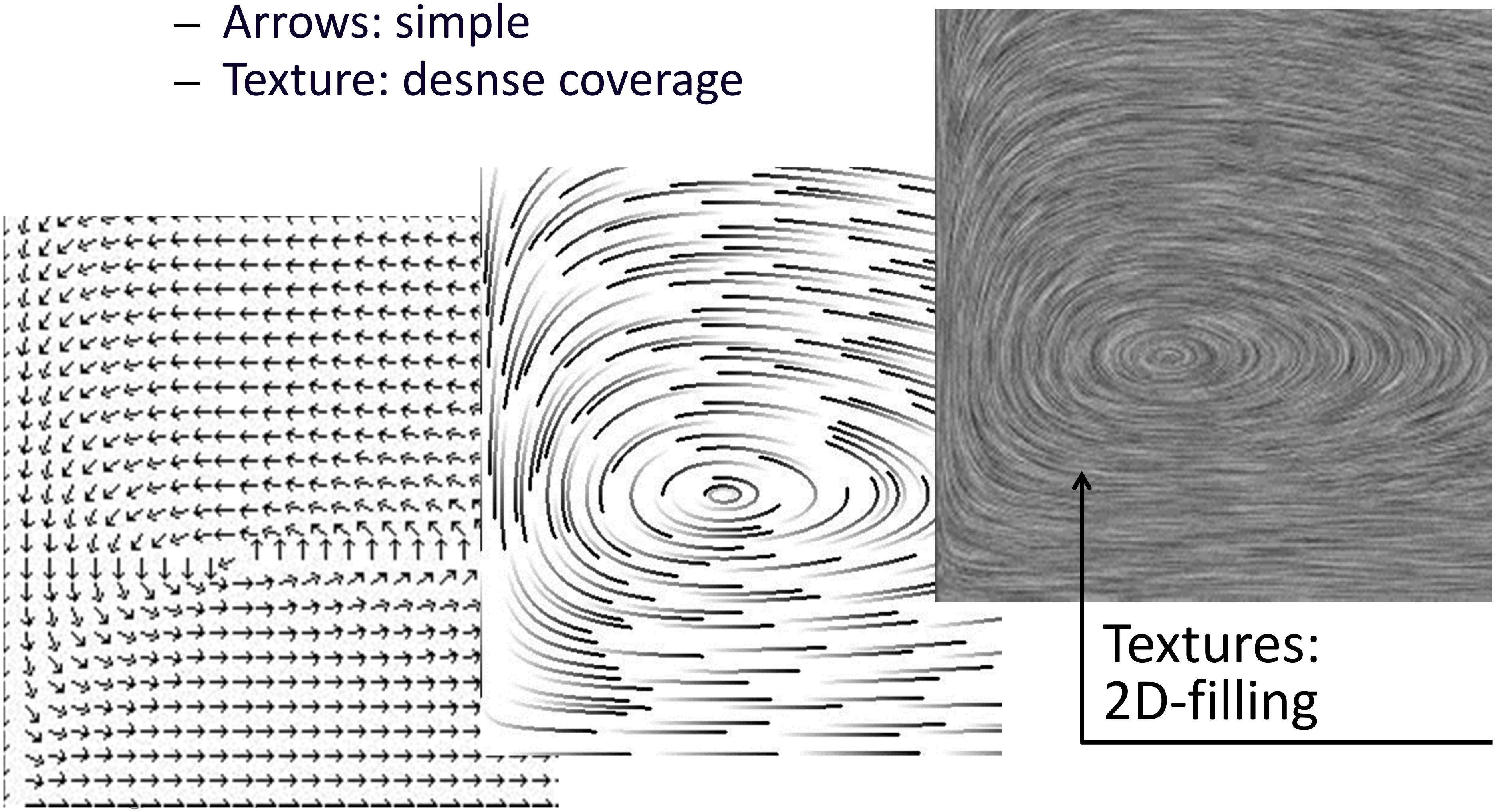
(1996)

- Dataset sizes over time:

Data set name and year	# vertices	# time steps	size (MB)
Tapered Cylinder ’90	131,000	400	1,050
McDonnell Douglas F/A–18 ’92	1,200,000	400	12,800
Descending Delta Wing ’93	900,000	1,800	64,800
Bell–Boeing V–22 tiltrotor ’93	1,300,000	1,450	140,000
Bell–Boeing V–22 tiltrotor ’98	10,000,000	1,450	600,000

Standard Visualization Techniques for Flow Data

- Arrows vs. Streamlines vs. Textures
  - Streamlines: selective
  - Arrows: simple
  - Texture: desnse coverage



Some Feature Geometry of Vector Fields

Some Feature Geometry of Vector Fields

Let us focus on steady flow at this moment

Streamlines – Theory

Correlations:

- flow data **V**: derivative information
  - $d\boldsymbol{x}/dt = \boldsymbol{v}(\boldsymbol{x})$;
    spatial points $\boldsymbol{x} \in R^n$, time $t \in R$, flow vectors $\boldsymbol{v} \in R^n$
- streamline $\boldsymbol{s}$: integration over time, also called trajectory, solution, integral curve
  - $\boldsymbol{s}(t) = \boldsymbol{s}_0 + \int_{0 \le u \le t} \boldsymbol{v}(\boldsymbol{s}(u))\, du$;
    seed point $\boldsymbol{s}_0$, integration variable u
- Property:
  - uniqueness
- difficulty: result **s** also in the integral→analytical solution usually impossible.

Streamlines – Computation

Basic approach:

- theory: $\boldsymbol{s}(t) = \boldsymbol{s}_0 + \int_{0 \le u \le t} \boldsymbol{v}(\boldsymbol{s}(u))\, du$
- practice: numerical integration
- idea:
  (very) locally, the solution is (approx.) linear
- Euler integration:
  follow the current flow vector $\boldsymbol{v}(\boldsymbol{s}_i)$ from the current streamline point $\mathbf{s}_i$
  for a very small time (dt) and therefore distance

Euler integration: $\boldsymbol{s}_{i+1} = \boldsymbol{s}_i + \boldsymbol{v}(\boldsymbol{s}_i) \cdot dt$,
integration of small steps (dt very small)

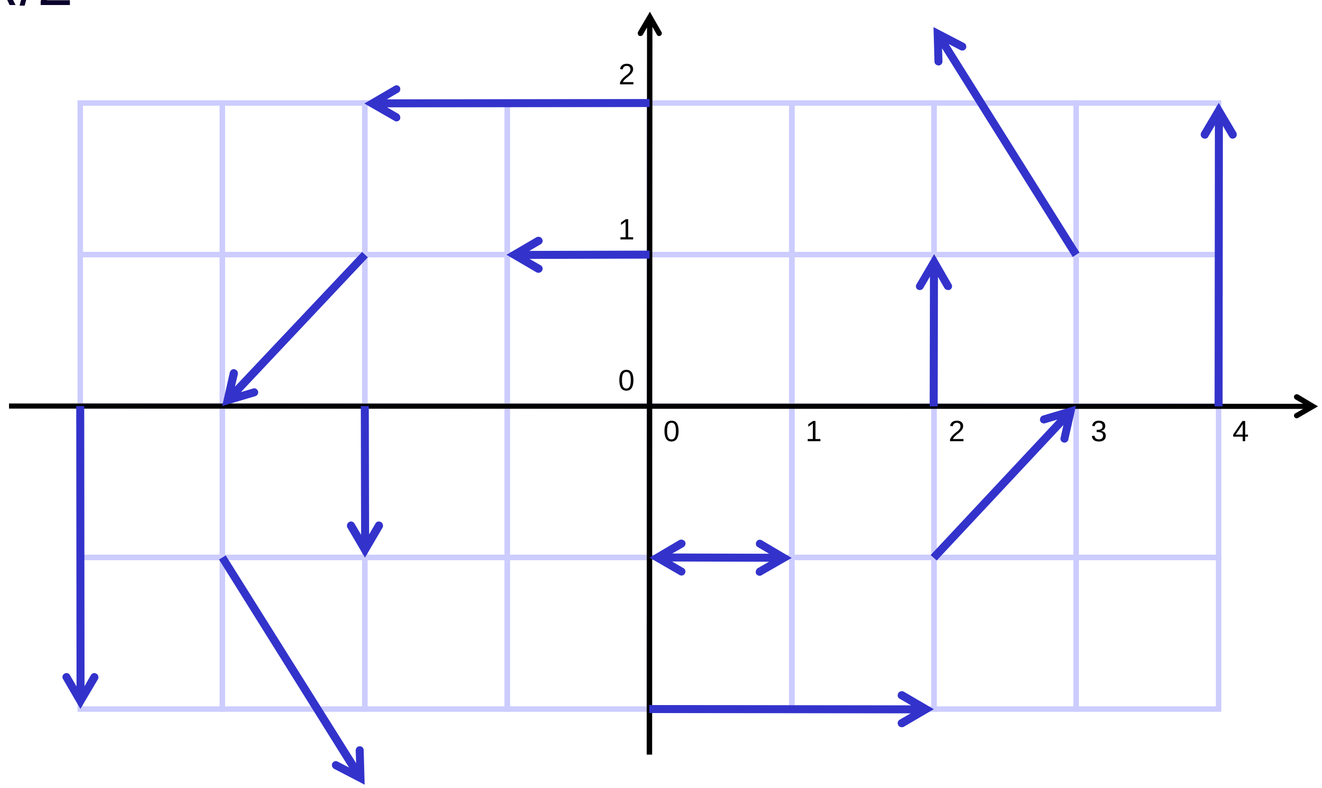
Euler Integration – Example

2D analytic field (no need of grid and interpolation):

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

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

Sample arrows:



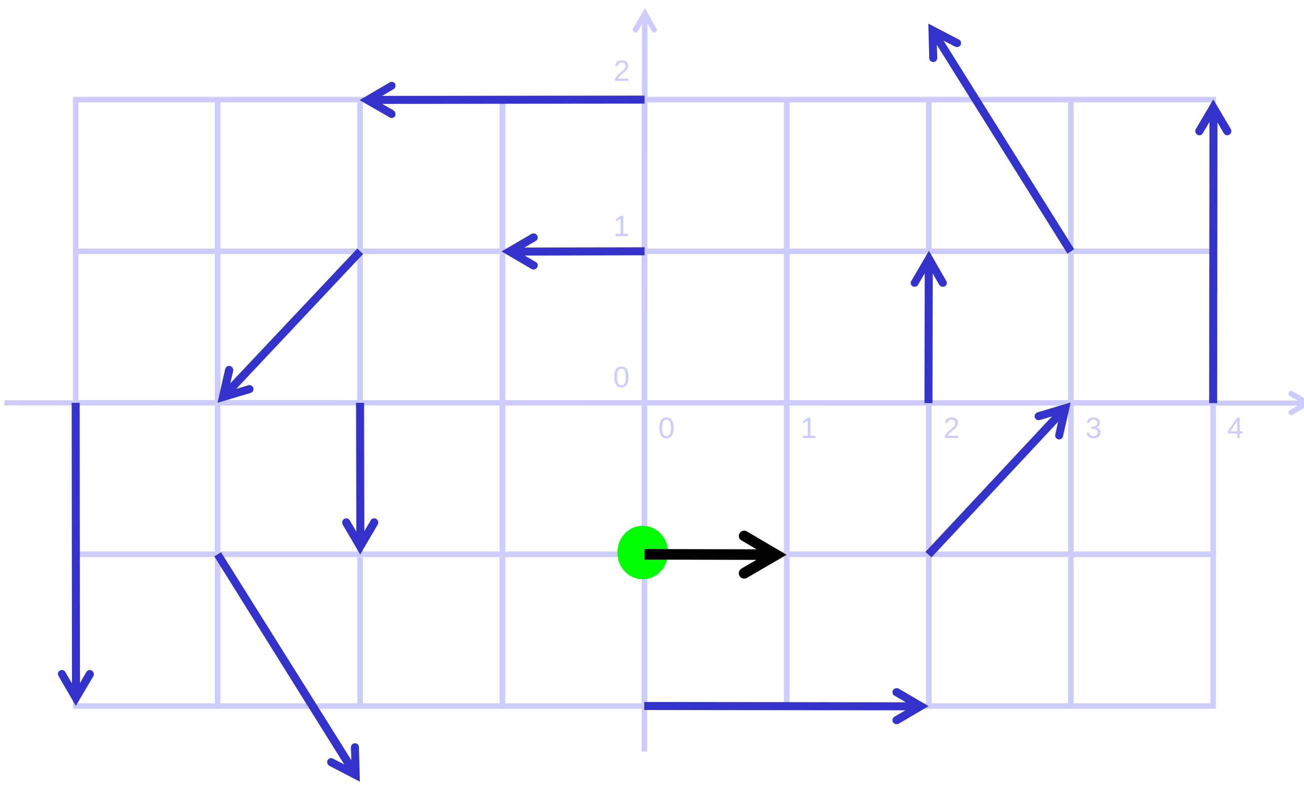
Ground truth flows form ellipses.

Euler Integration – Example

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

$v_x = dx/dt = -y$

$v_y = dy/dt = x/2$

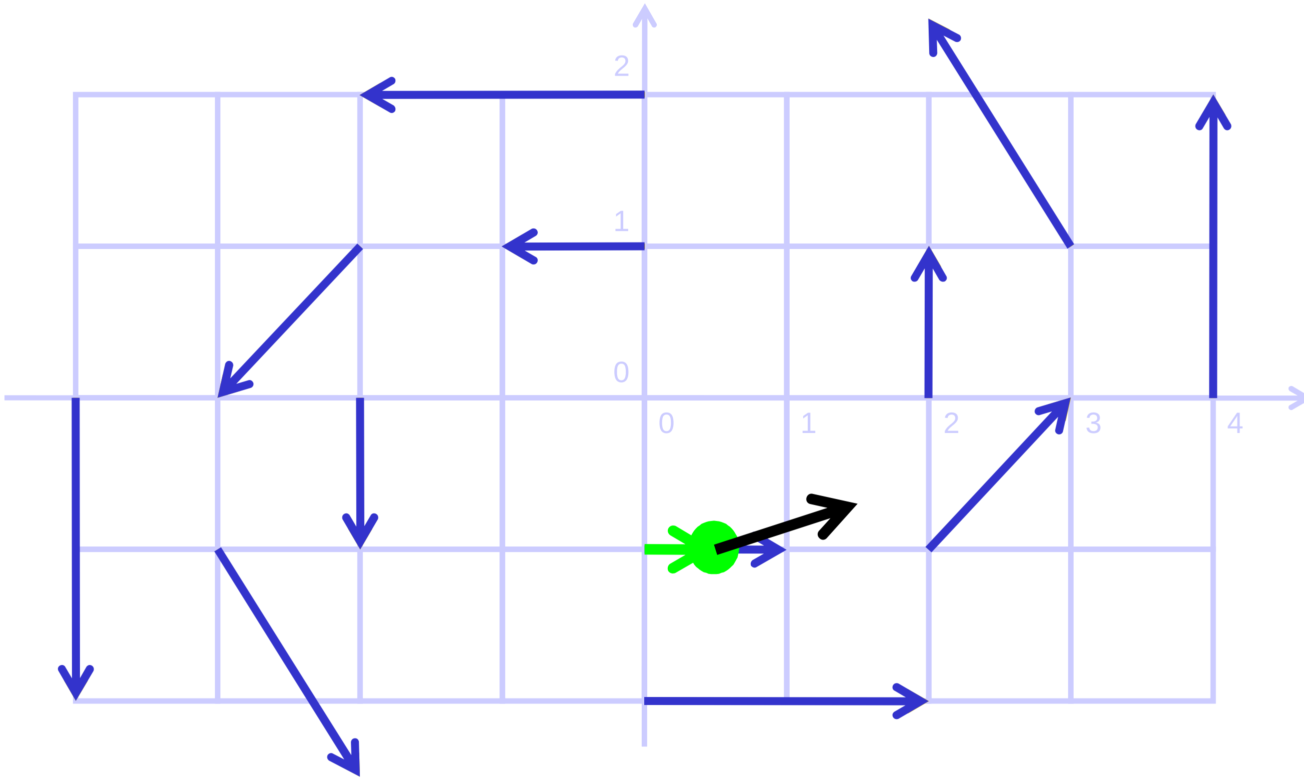


Euler Integration – Example

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

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

$v_y = dy/dt = x/2$

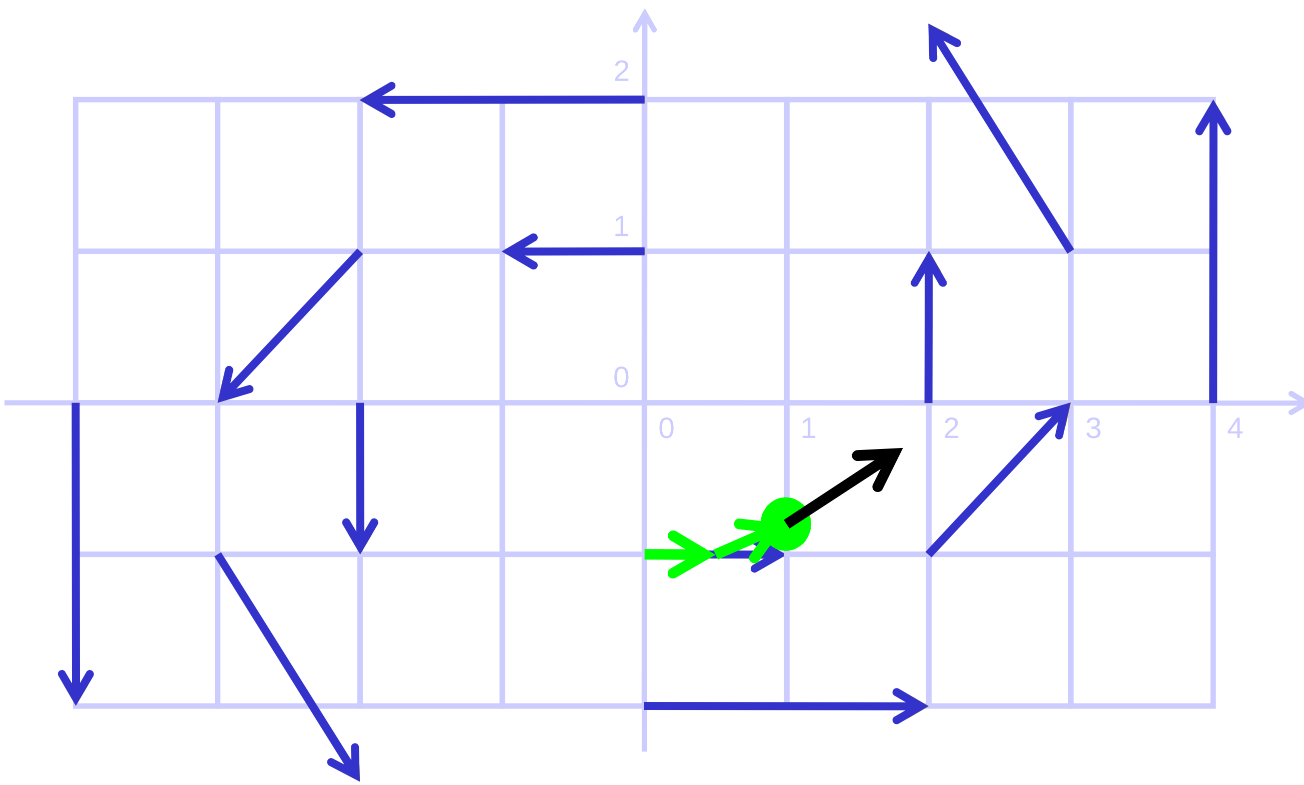


Euler Integration – Example

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

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

$v_y = dy/dt = x/2$



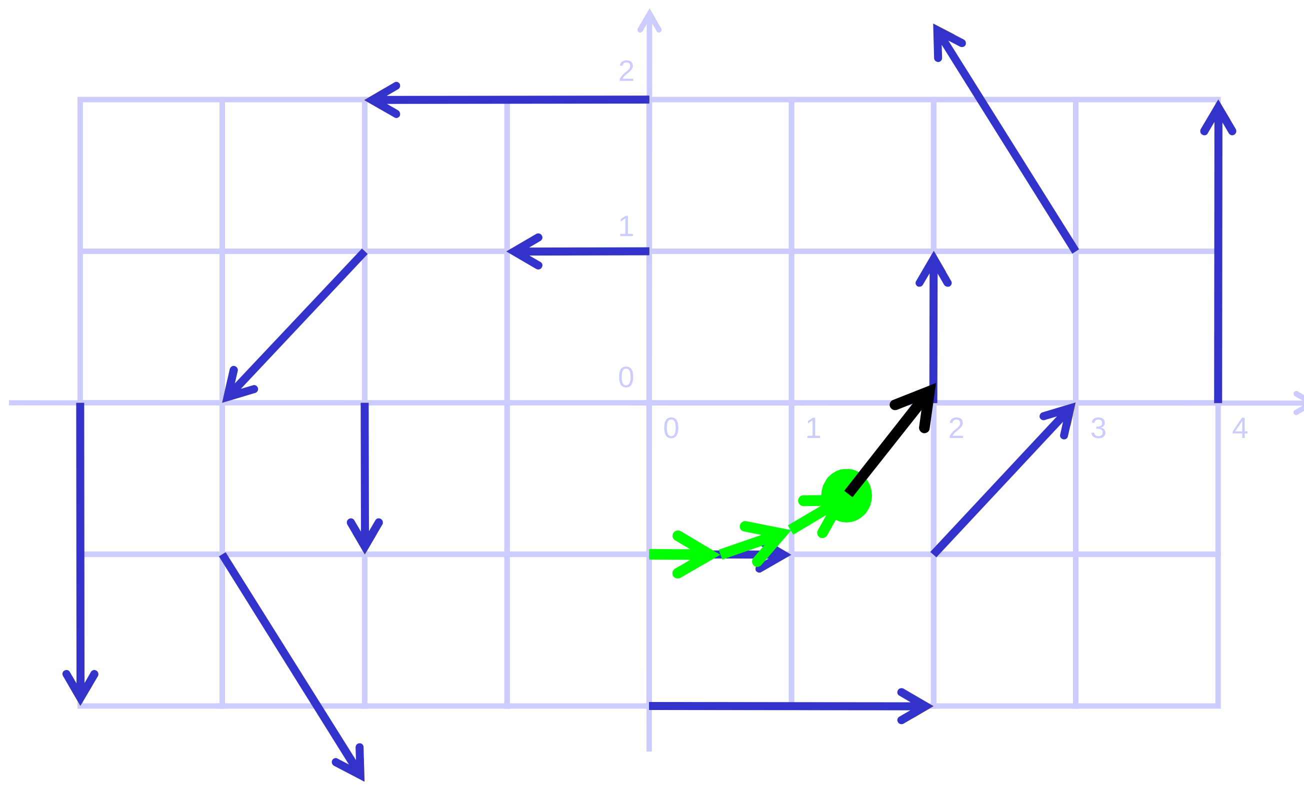
Euler Integration – Example

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

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

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

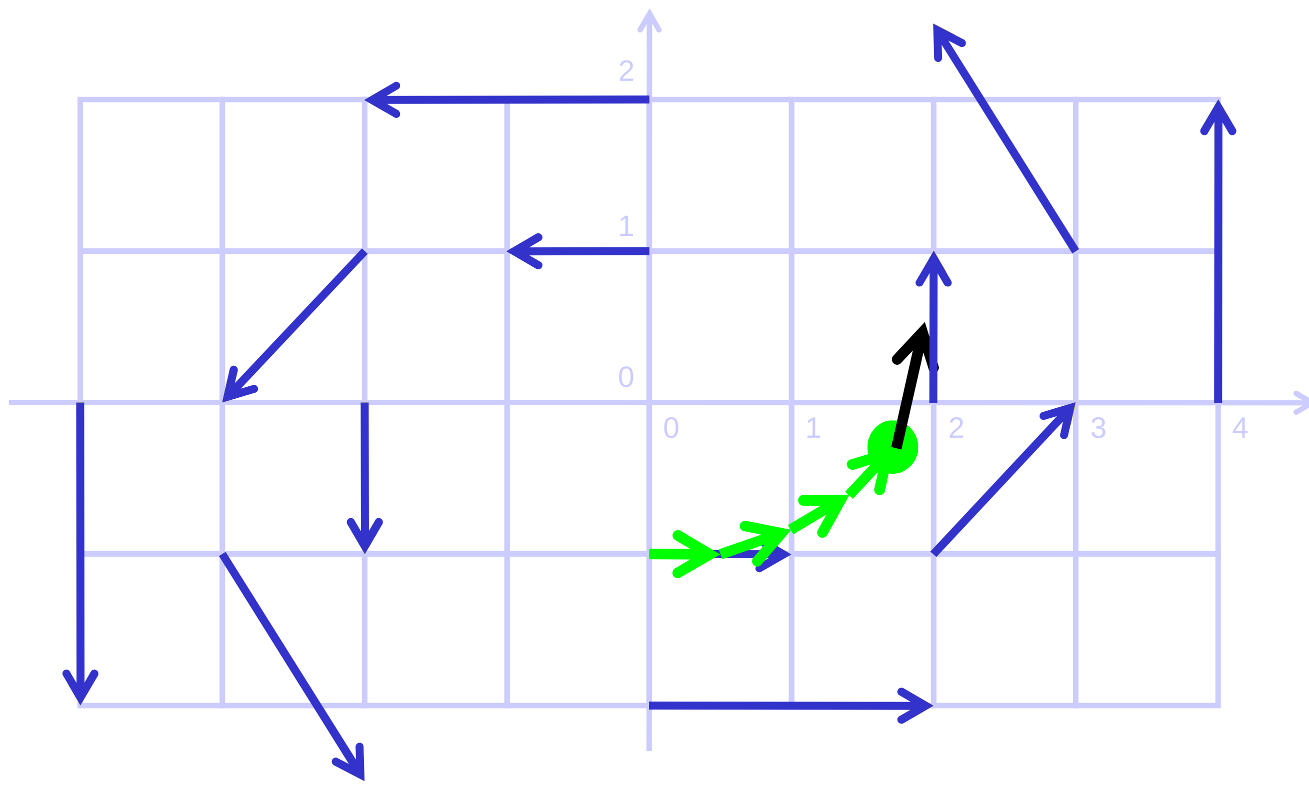
$v_y = dy/dt = x/2$



Euler Integration – Example

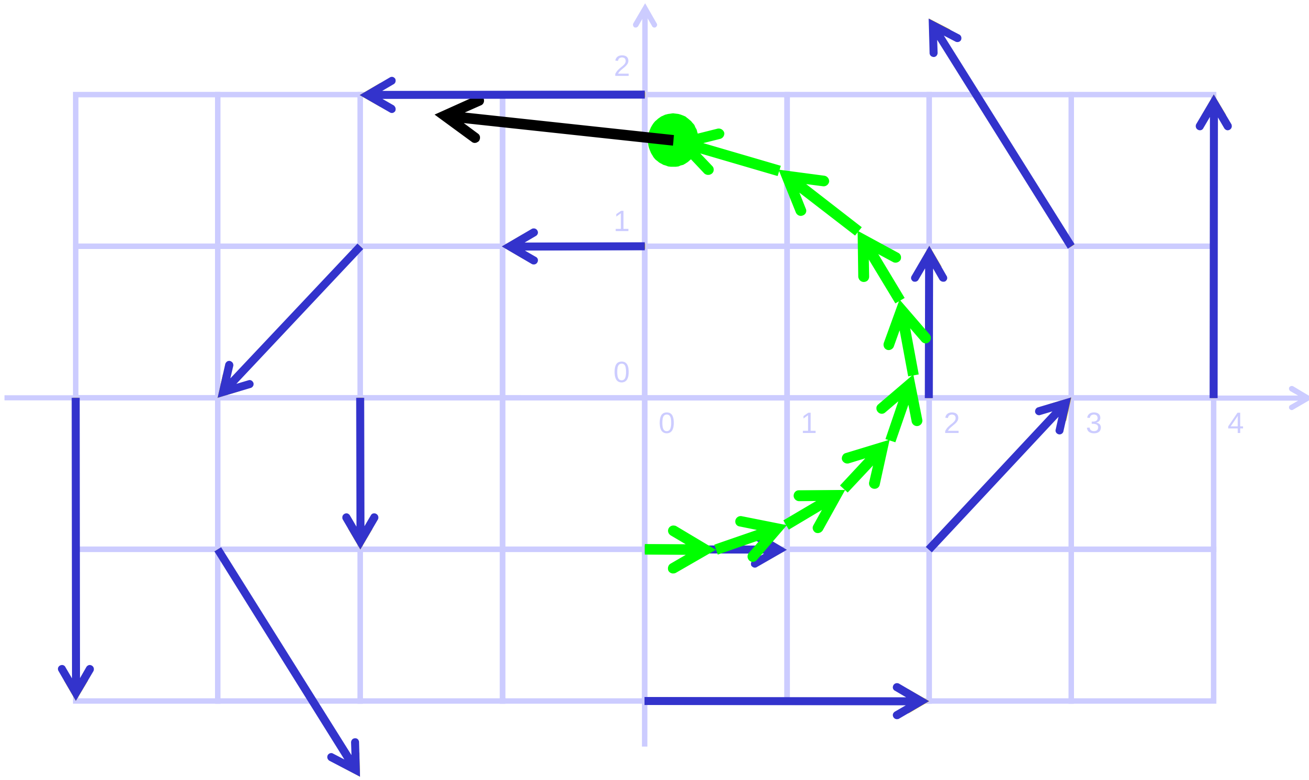
- $\mathbf{s}_4 \quad = (7/4\,|\,-17/64)^T \approx (1.75\,|\,-0.27)^T;$

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



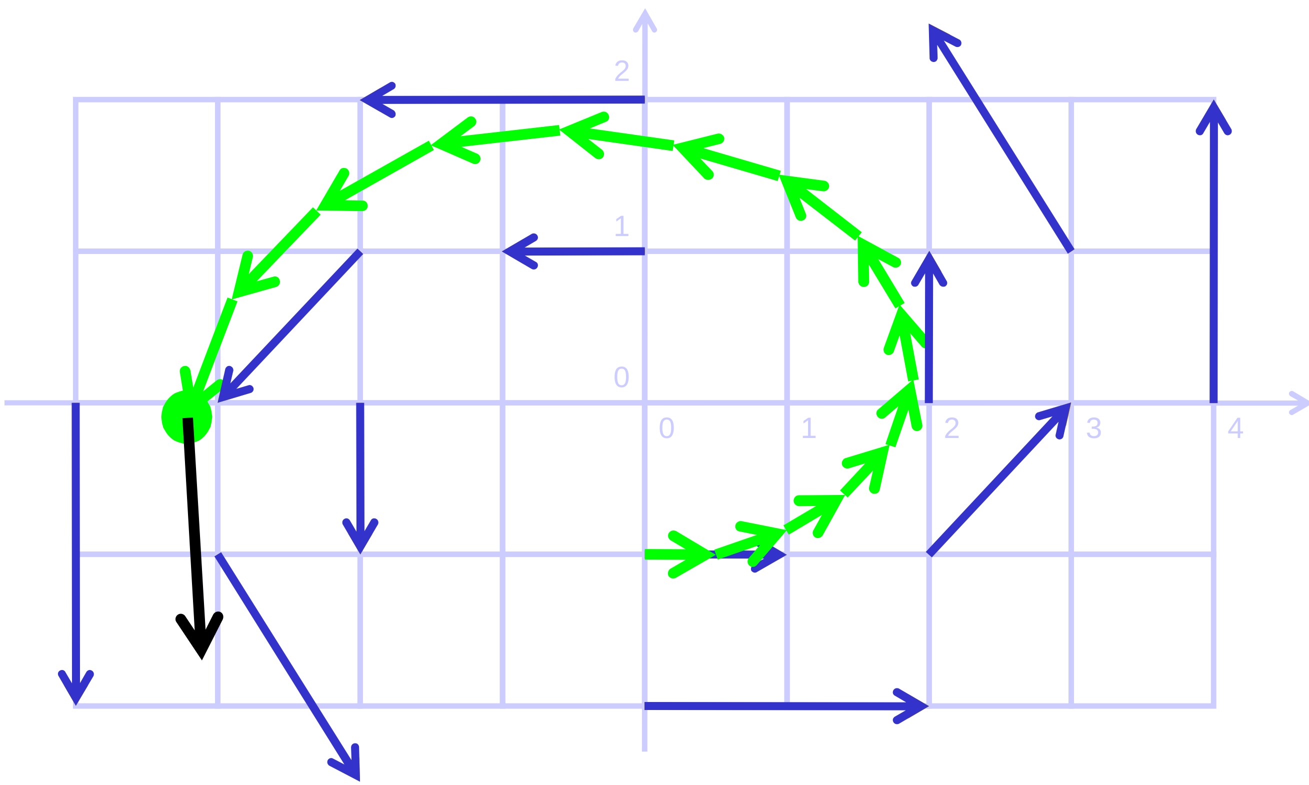
Euler Integration – Example

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



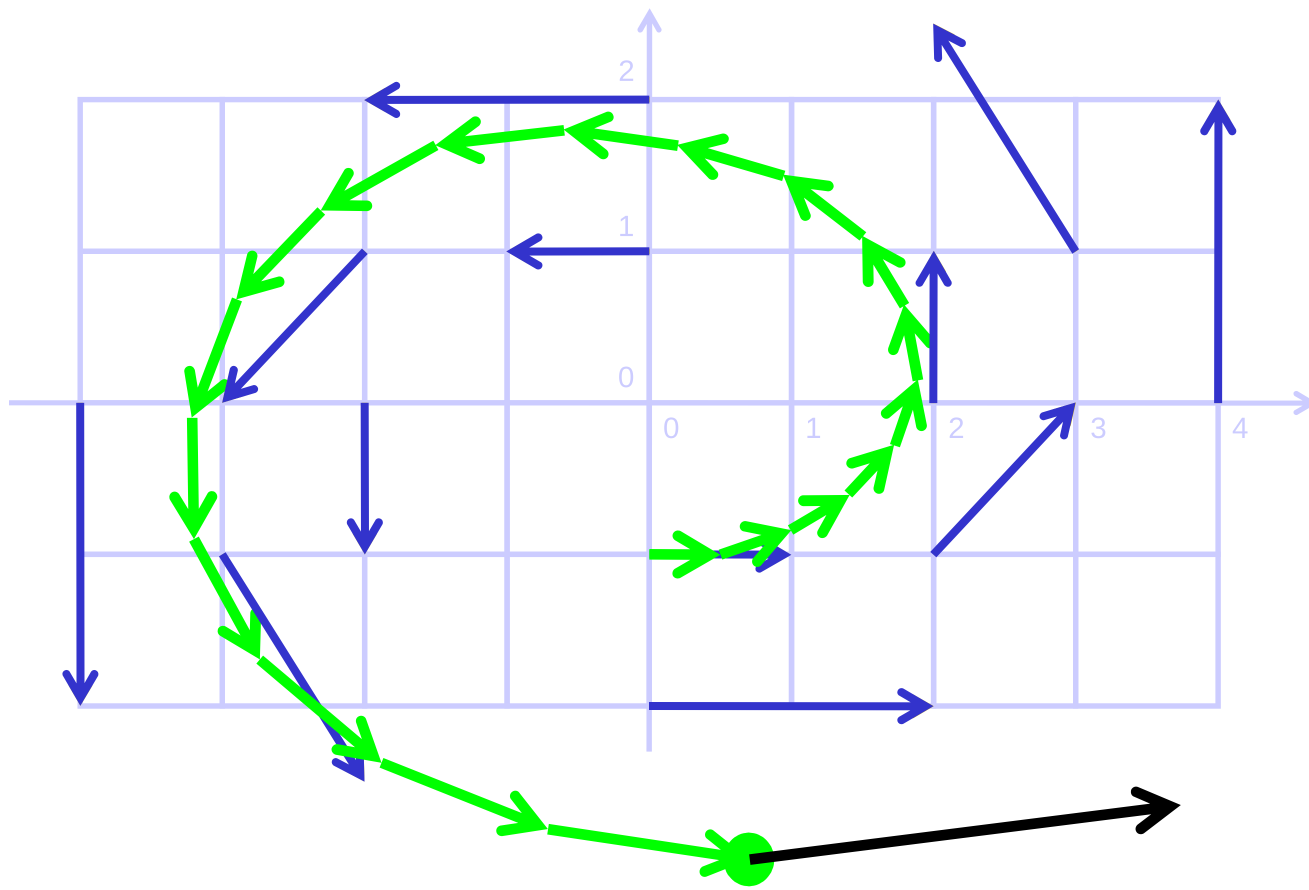
Euler Integration – Example

$\mathbf{s}_{14} \approx (-3.22|-0.10)^T;$
$\mathbf{v}(\mathbf{s}_{14}) \approx (0.10|-1.61)^T;$



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

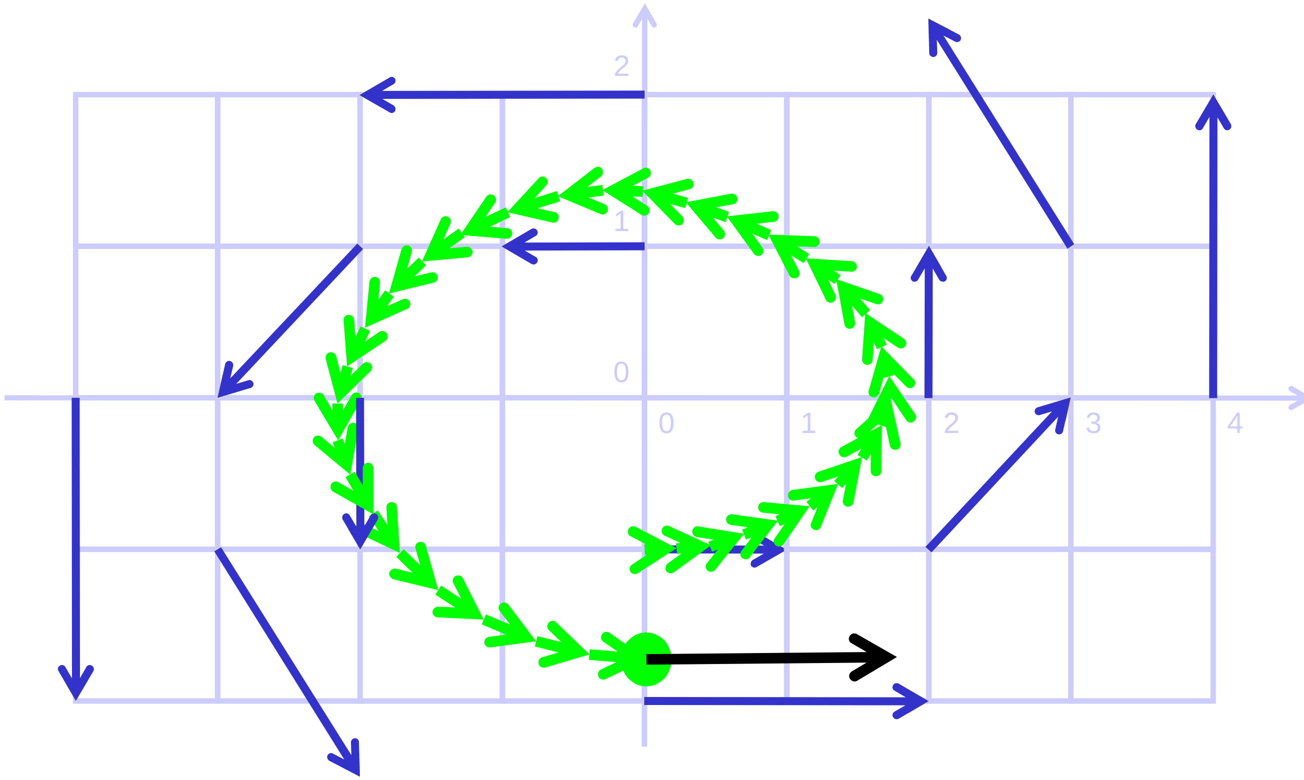


Euler Integration – Example

- dt smaller (1/4): more steps, more exact.

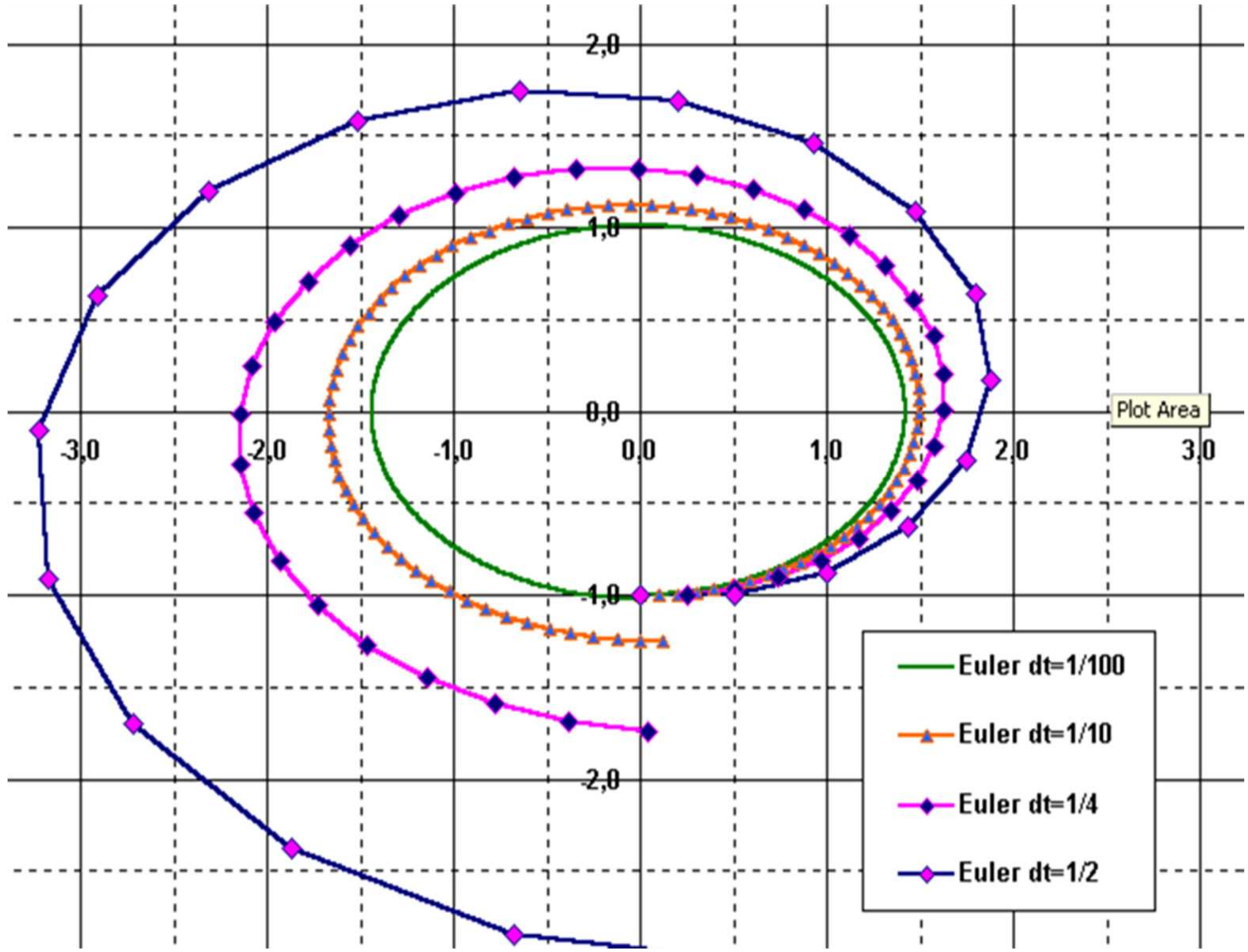
$\mathbf{s}_{36} \approx (0.04 | -1.74)^T$; $\mathbf{v}(\mathbf{s}_{36}) \approx (1.74 | 0.02)^T$;

- 36 steps



Comparison Euler, Step Sizes

Euler
quality is proportional to dt

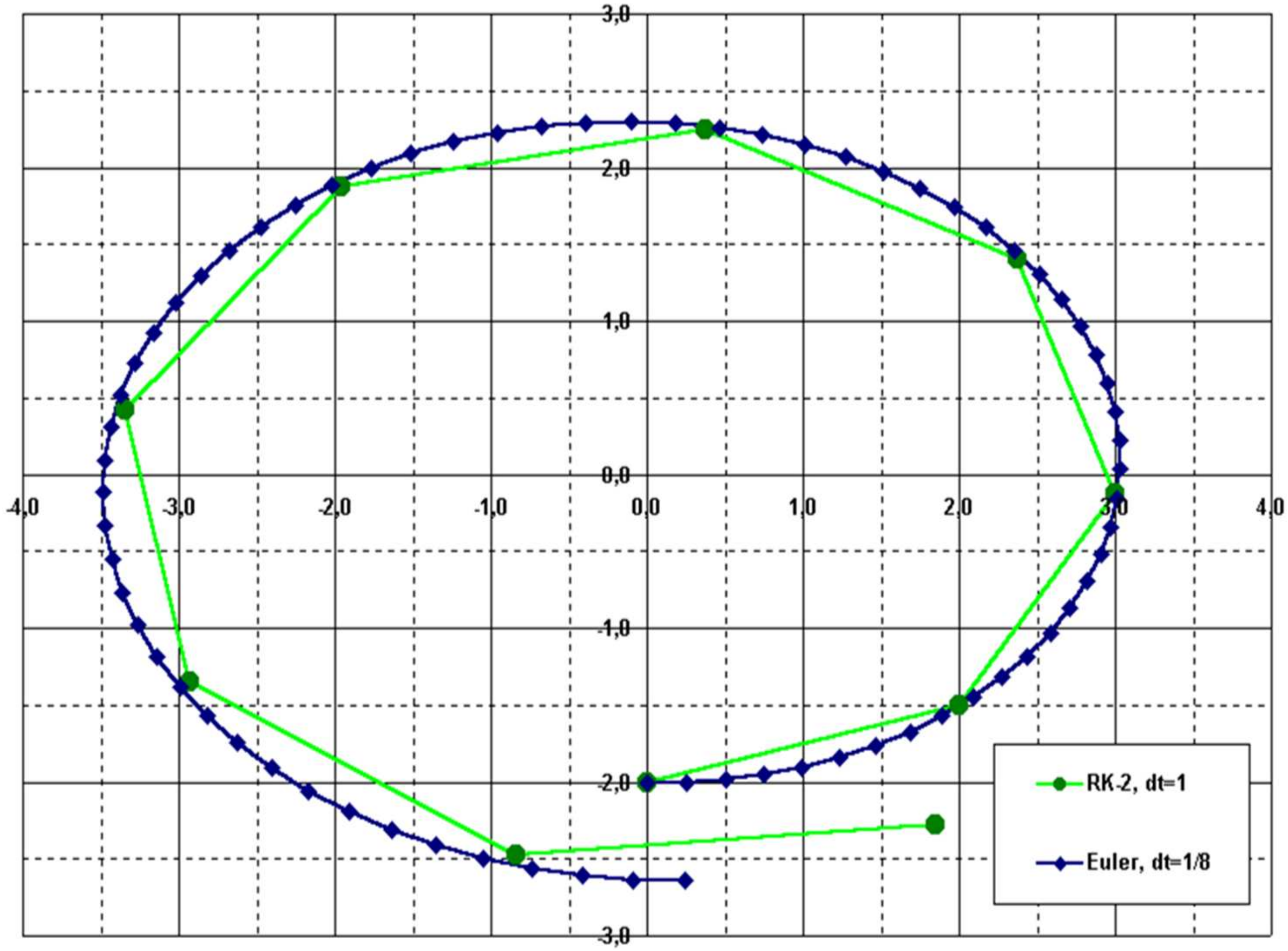


Euler Example – Error Table

d*t*	#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 d*t* = 1 (9 steps)
better
than Euler
with d*t* = 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

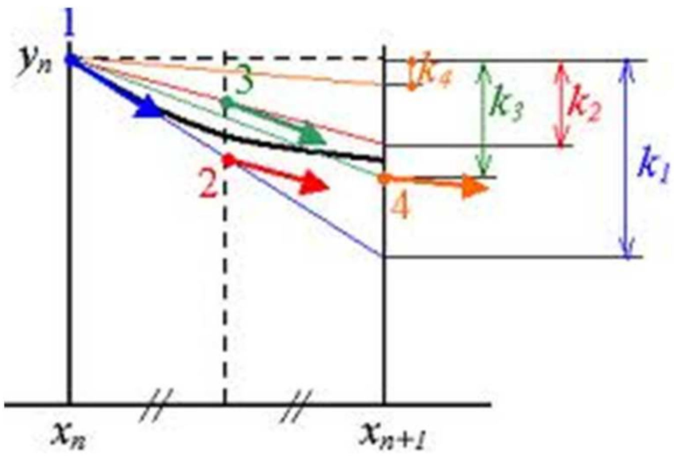
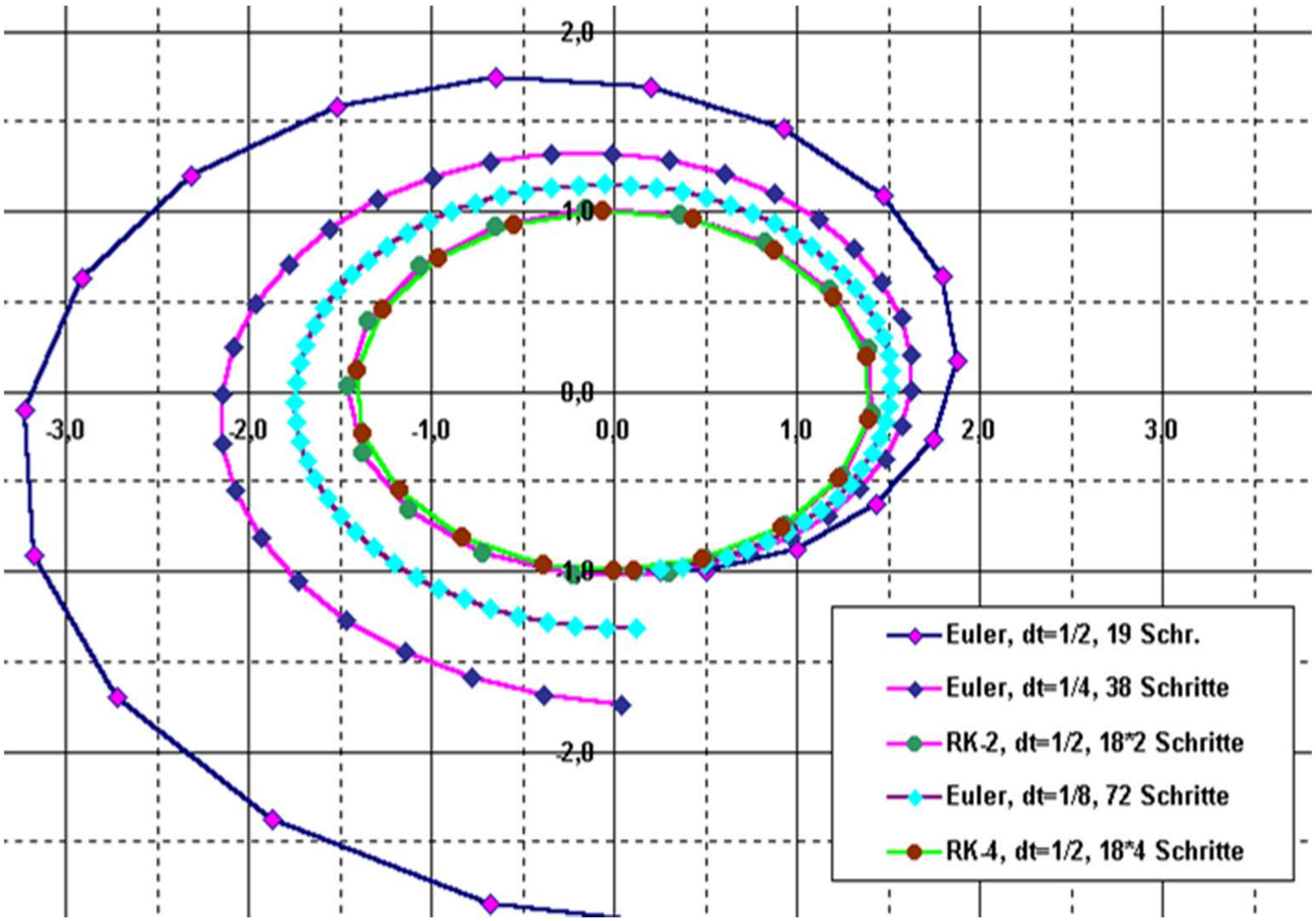


Image source: http://cinet.chim.pagesperso-orange.fr/ex_sa/int_num.html

Euler vs. Runge-Kutta

RK-4: pays off only with complex flows

Here
approx.
like
RK-2



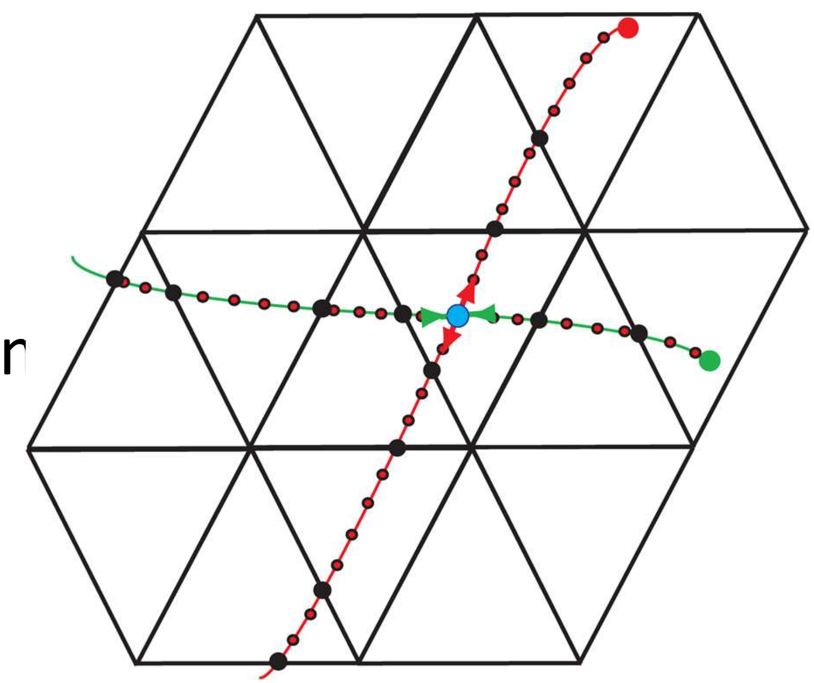
Integration Schemes

Summary:

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

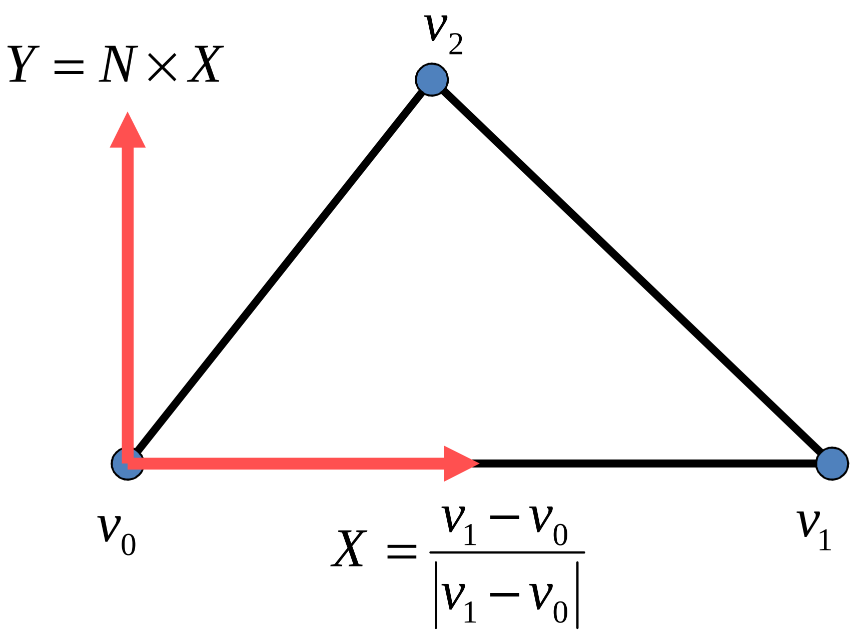
Streamline Tracing Under Discrete Samples

- Important components
  - Interpolation
  - Local frame
  - Interior verification
  - Neighborhood information



Local Frame

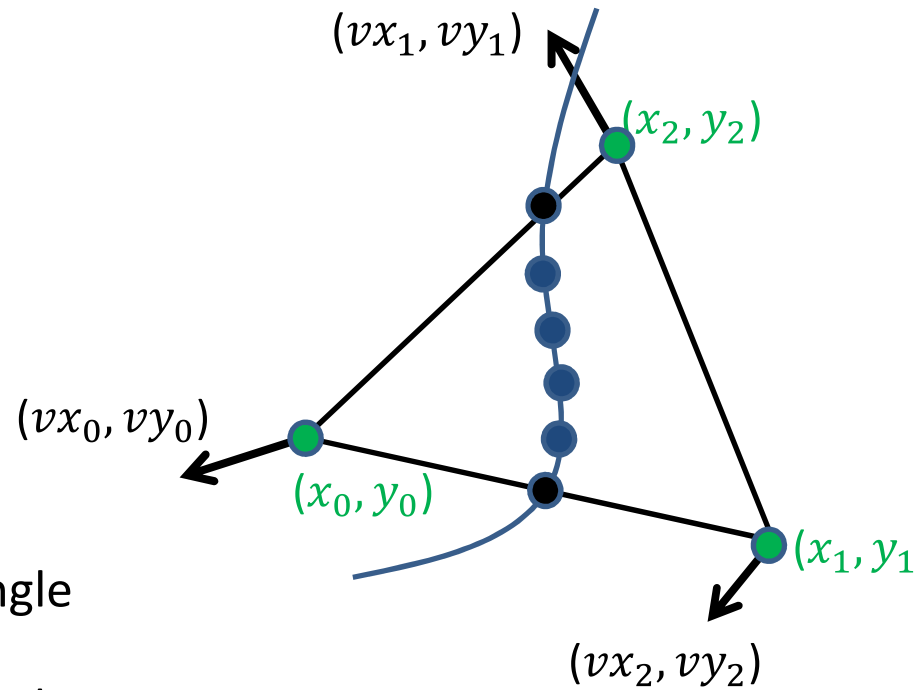
- Triangle



Streamline Tracing Under Discrete Samples

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax + by + c \\ dx + ey + f \end{pmatrix}$$



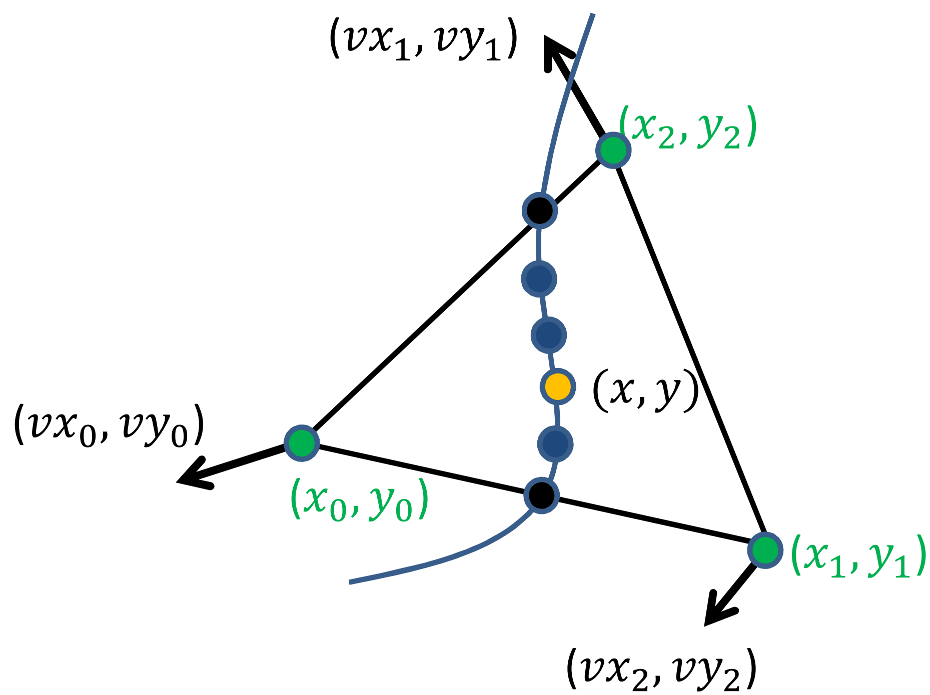
Trace streamline locally within each triangle

Need to explicitly handle the transition between different triangles

Streamline Tracing Under Discrete Samples

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax + by + c \\ dx + ey + f \end{pmatrix}$$

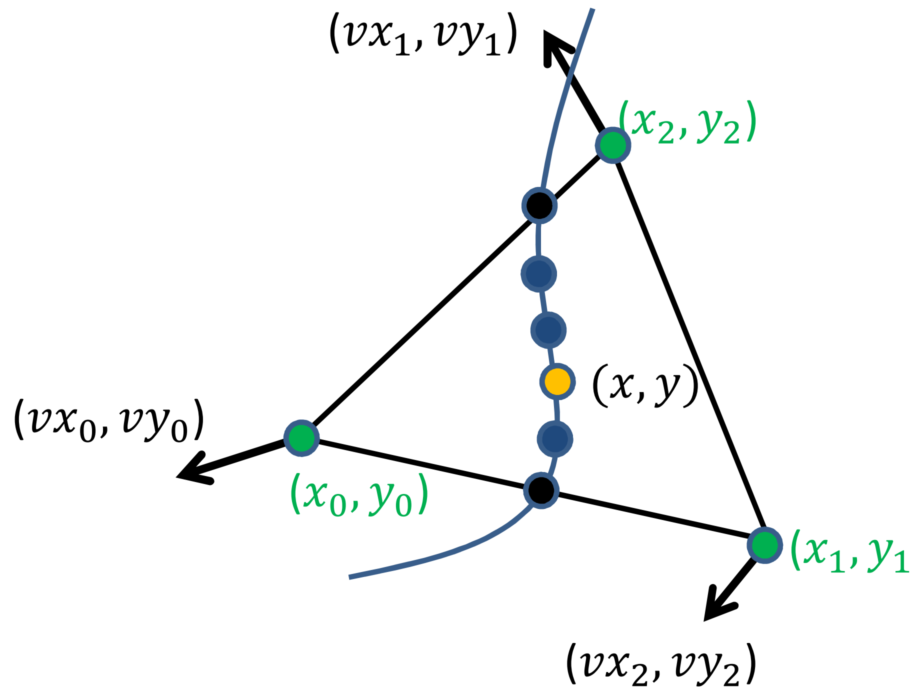


What is the vector value at (x, y)?

Streamline Tracing Under Discrete Samples

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \binom{v_x}{v_y} = \binom{ax+by+c}{dx+ey+f}$$



What is the vector value at (x, y)?

Interpolating the vector values at three vertices

Compute Barycentric Coordinates

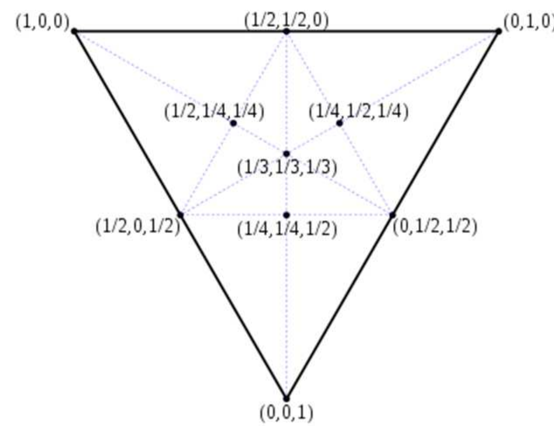
$$f_{ab}(x,y) = (y_a - y_b)x + (x_b - x_a)y - x_a y_b + x_b y_a$$

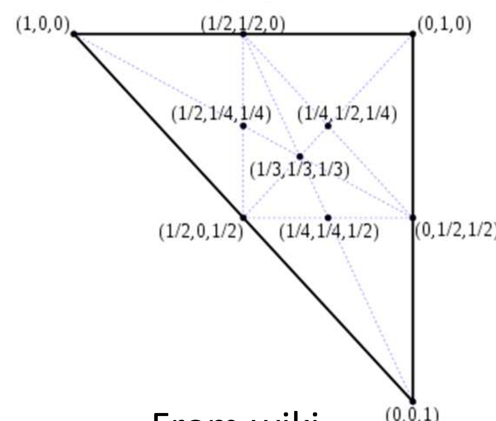
For $a, b \in \{0,1,2\}$, i.e. the index of the vertices of a triangle.

We then have,

$$\alpha = \frac{f_{12}(x,y)}{f_{12}(x_0,y_0)} \qquad \beta = \frac{f_{20}(x,y)}{f_{20}(x_1,y_1)} \qquad \gamma = \frac{f_{01}(x,y)}{f_{01}(x_2,y_2)}$$

- (α, β, γ): barycentric coordinates of a point (x, y)
- $\alpha + \beta + \gamma = 1$
- Only if $0 < \alpha, \beta, \gamma < 1$, (x, y) is inside the triangle



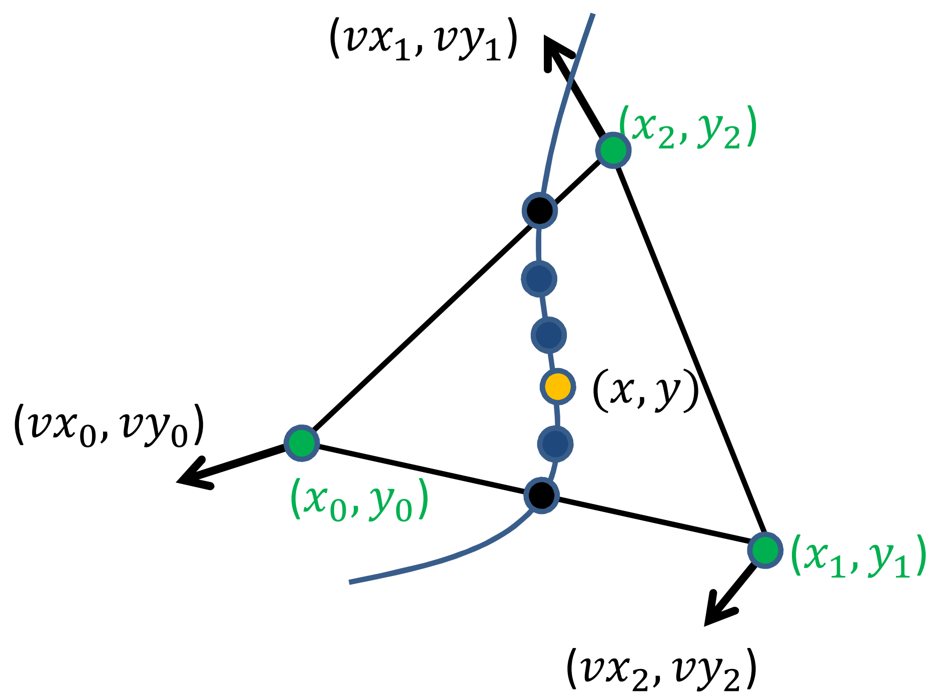


From wiki

Interpolating Vector Values

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax + by + c \\ dx + ey + f \end{pmatrix}$$



What is the vector value at (x, y)?

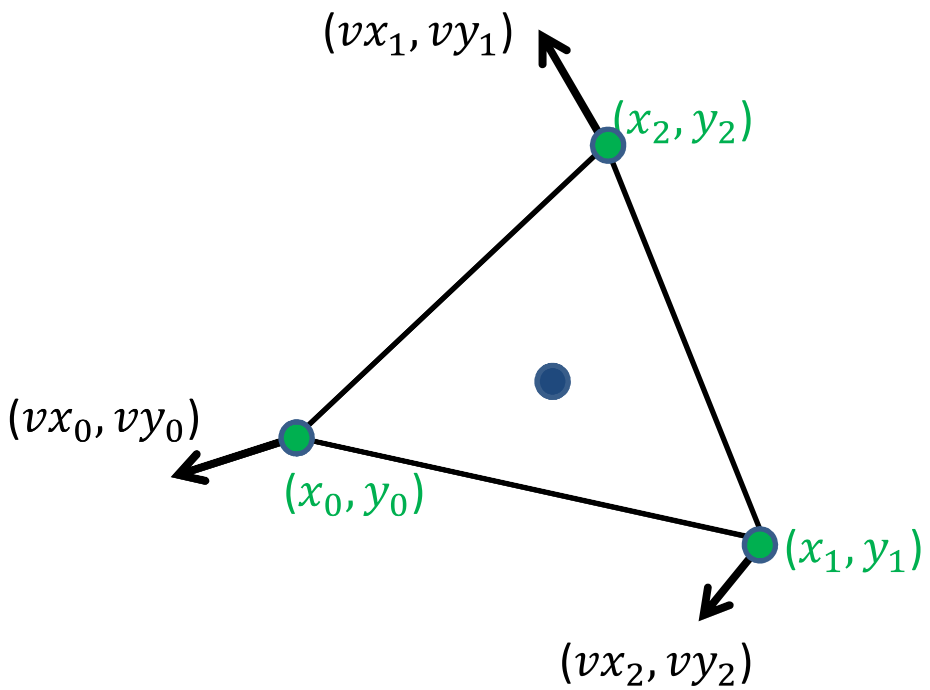
Interpolating the vector values at three vertices

$$\begin{pmatrix} vx \\ vy \end{pmatrix} = \alpha \begin{pmatrix} vx_0 \\ vy_0 \end{pmatrix} + \beta \begin{pmatrix} vx_1 \\ vy_1 \end{pmatrix} + \gamma \begin{pmatrix} vx_2 \\ vy_2 \end{pmatrix}$$

Interpolating Vector Values

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax + by + c \\ dx + ey + f \end{pmatrix}$$



We have

$$\begin{pmatrix} vx_0 \\ vy_0 \end{pmatrix} = \begin{pmatrix} ax_0 + by_0 + c \\ dx_0 + ey_0 + f \end{pmatrix}$$

$$\begin{pmatrix} vx_1 \\ vy_1 \end{pmatrix} = \begin{pmatrix} ax_1 + by_1 + c \\ dx_1 + ey_1 + f \end{pmatrix}$$

$$\begin{pmatrix} vx_2 \\ vy_2 \end{pmatrix} = \begin{pmatrix} ax_2 + by_2 + c \\ dx_2 + ey_2 + f \end{pmatrix}$$

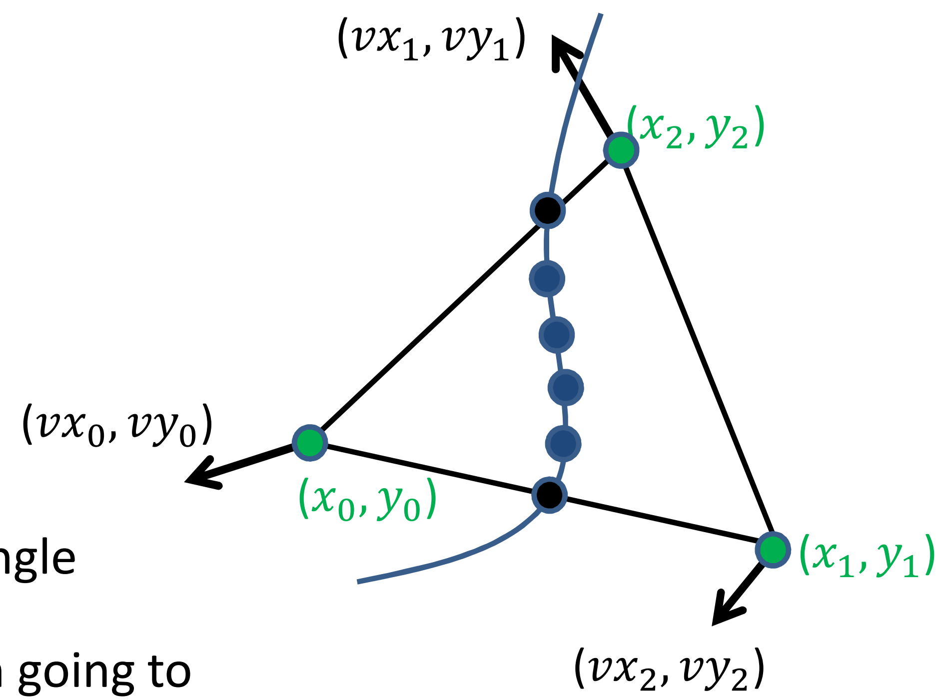
Solve for a linear system to get the coefficients

Given the linear form above, you can compute the vector value at any point within the triangle.

Streamline Tracing Under Discrete Samples

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax + by + c \\ dx + ey + f \end{pmatrix}$$



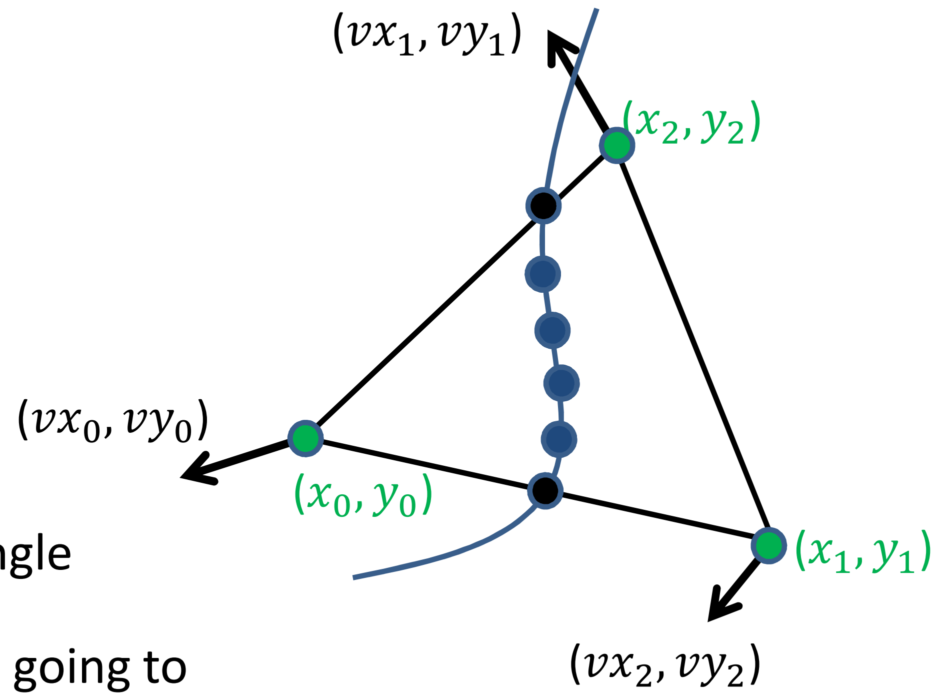
Trace streamline locally within each triangle

How can I determine which triangle I am going to enter?

Streamline Tracing Under Discrete Samples

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax+by+c \\ dx+ey+f \end{pmatrix}$$



Trace streamline locally within each triangle

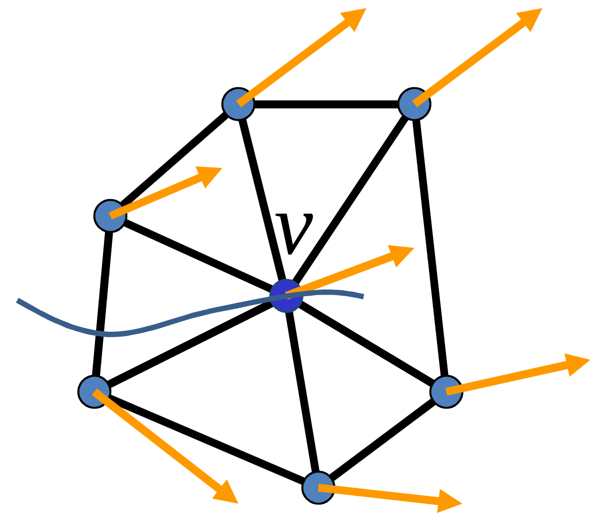
How can I determine which triangle I am going to enter?

Using the barycentric coordinate and the help with corner table.
Recall: only if $0 < \alpha, \beta, \gamma < 1$, (x, y) is inside the triangle

Streamline Tracing Under Discrete Samples

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x, y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax + by + c \\ dx + ey + f \end{pmatrix}$$



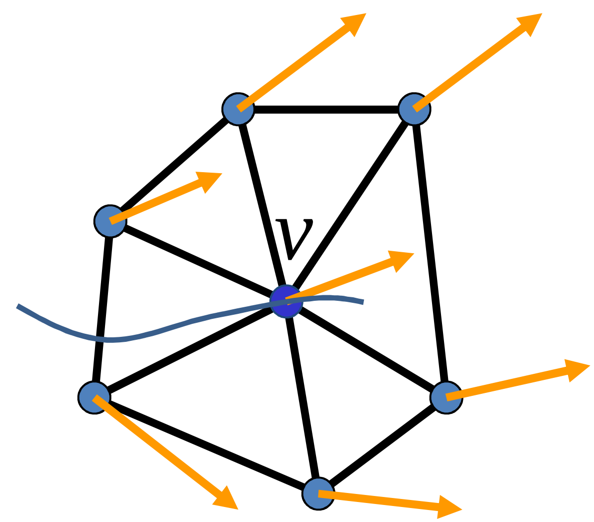
Trace streamline locally within each triangle

What if the streamline passes through a vertex?

Streamline Tracing Under Discrete Samples

Assume a piecewise linear vector field

$$\vec{\boldsymbol{V}}(x,y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} ax + by + c \\ dx + ey + f \end{pmatrix}$$



Trace streamline locally within each triangle

What if the streamline passes through a vertex?

Use sorted corners around v to determine which one contains the vector defined at v

Summary of The Algorithm

```
Input: seed (x,y) and triangle T
Output: a point list P that forms the streamline

for (i=0; i<max_triangles; i++)
{
     if (T < 0) return;  // we reach a boundary
     //compute streamline within current triangle T
      convert (x, y) to local coordinates;
      for (step = 0;  step < max_steps; step++)
      {    advance to next position using (Euler|RK2|RK4) integrator
           if we reach a fixed point, return;
           if the next position is still inside T
                  store this new position to point list P and continue
           else
                  determine next triangle it will enter, let T be the next triangle
       }
 }
```

Note that this framework can be extended to surface flow tracing with additional care

Acknowledgment

Thanks for the materials

- Prof. Robert S. Laramee, Swansea University, UK