# Flow Visualization

Used for this lecture:

Visualisierung, Schumann/Müller, 2000

Flow Visualization, Gröller/Hauser (pdf on Web)

Interactive Data Visualization, Ward, Grinstein, Keim

and more.



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization
and Image Processtegrationg
Prof. Dr. Gitta Domik

# Representation – Flow Visualization

## Goals of this chapter:

- introduce visualization techniques for *flow data*
  - *Direct flow vis with arrows and glyphs;*
  - *Indirect flow vis with streamlines; streaklines; pathlines.*
  - *local versus global techniques*
  - *Line Integral Convolution (LIC)*
- understand different sources of flow data
- Compute with numerical methods for approximation

# Flow Data: Structure and Def

- Data on structured grid in 2 or 3 dimensions
- Always vector data arrays
  - but can also contain other – e.g. scalar data
- Direct relation between array element and structured grid
- According to Brodlie: $E_n^{Vk}$
  - with n = 2 or 3 (dimension of structured grid)
  - k= 1 or 2 or 3 (dimensionality of each vector)
  - Typically $E_3^{V_3}$ or $E_2^{V_2}$

## 2D versus Surfaces versus 3D

- 2d flow viz
  - 2d x 2d flow (another way of representing flow data dimensions)
  - models, slice flows (2d out of 3d)
- Visualization of surface flows
  - 3d flows around „obstacles“
  - boundary flows on surfaces (2d)
- 3d flow viz
  - 3d x 3d flows
  - Simulations, 3d-models

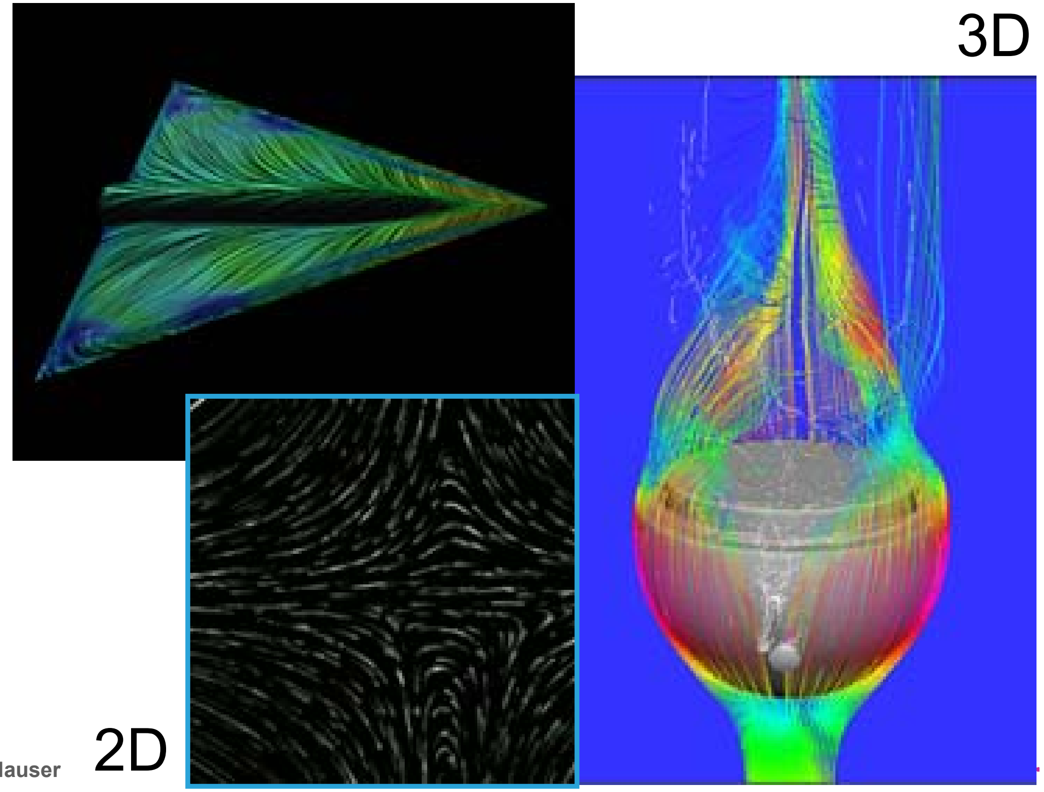
# Examples

## 2D/Surfaces/3D – Examples

TU VIENNA

3D

Surface



2D

Eduard Gröller, Helwig Hauser

TU Vienna

# Different Data Sources for Flow Viz

Where do the data come from:

- Flow simulation:
  - Airplane - / Ship - / Car Design
  - Weather simulation (air-, sea flows)
  - Medicine (e.g. blood flow)
- Flow measurements:
  - Wind tunnel, fluid tunnel
- Flow models:
  - Differential equations systems (dynamic systems)

## User Goals for Flow Viz

- Overview versus Detail (with context)

Examples TU Vienna

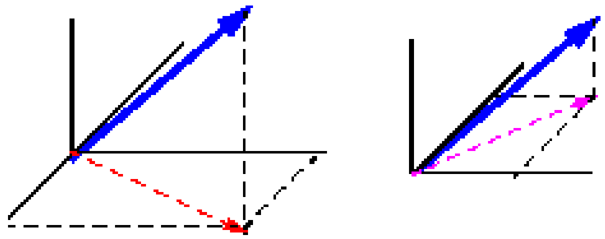
# Direct Flow Viz with Arrows and Glyphs

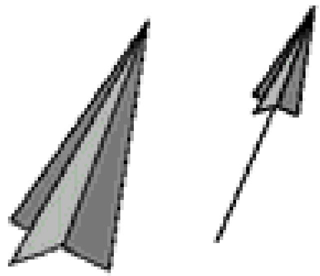
- **Representative data characteristics**
  - vector fields
- **Technique**
  - use arrow as glyph, vary following attributes of arrow depending on variables: direction/length/width/reflection properties of shaft, type/color of arrow head
- **Special note on effectiveness**
  - avoid cluttering by reducing amount of data to display
  - additional problems in 3-d through directional ambiguity

**Reference(s)** [POS94]

# Problems: Arrows in 3d

- Ambiguity
- Perspective Shortening
- 1d objects in 3d:
  - difficult spatial perception
- Visual clutter
- Improvement
- 3d arrows/glyphs
  - help to a certain extent

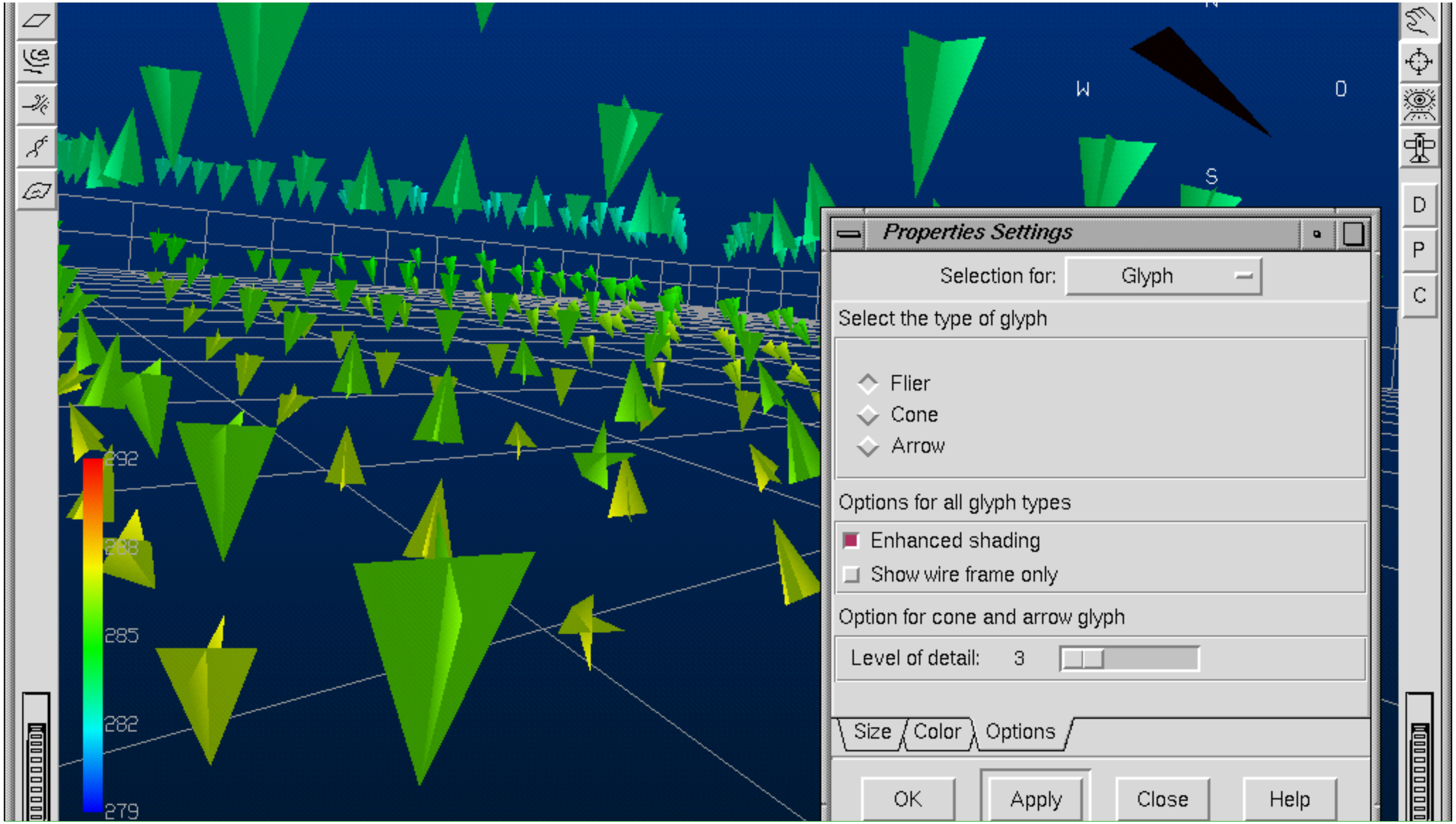




# Flow Viz more examples

- Project on environmental data of City of Hagen
  - wind directions+humidity+temperature ($V_3$ + 2 variables over x,y,z)
  - DEM+thematic map (buildings)+land use map (3 variables over x,y)
  - temporal
- Goal was to suggest location to build high-riser

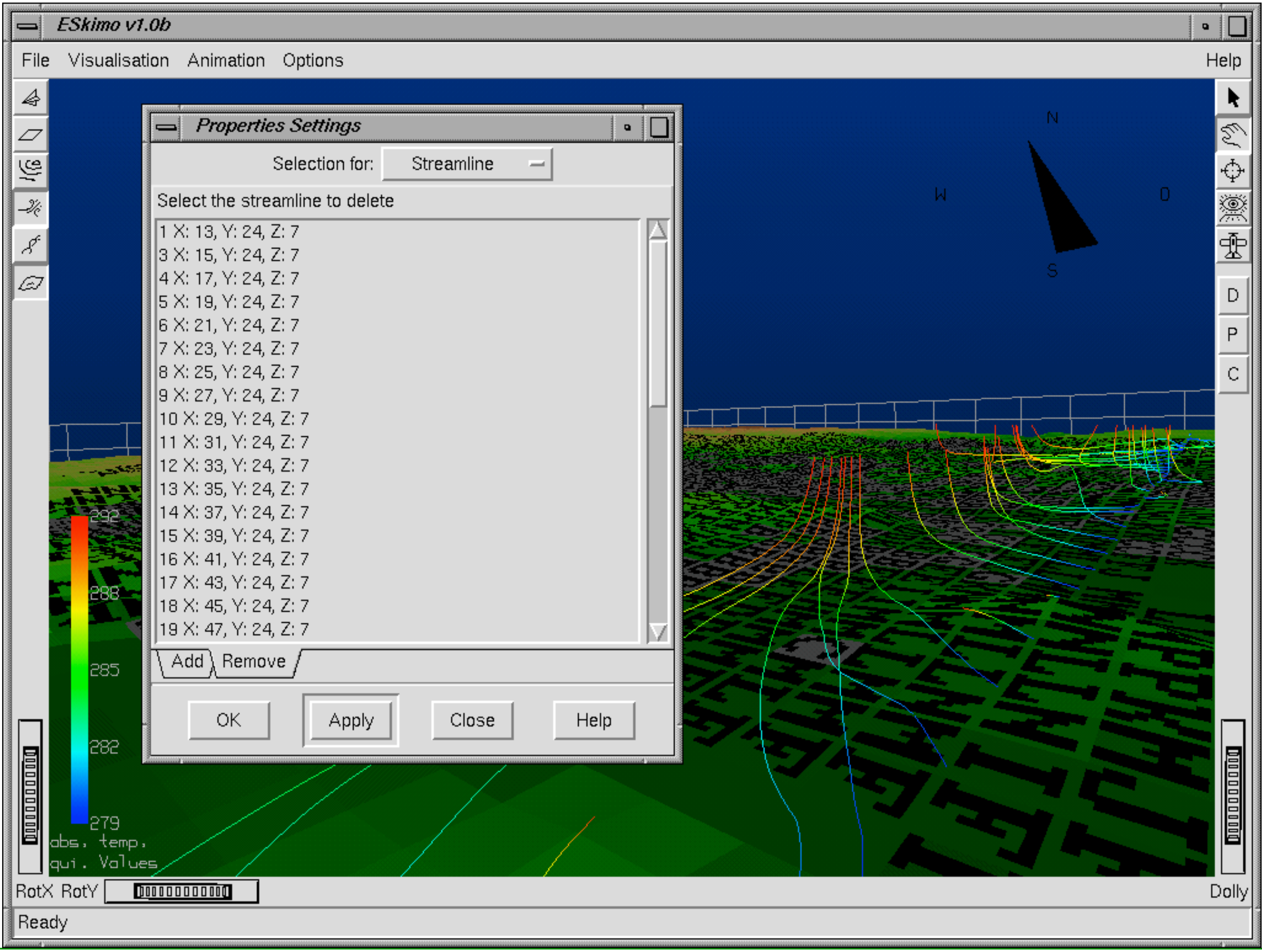
# Glyph based Vis of City of Hagen






UNIVERSITY OF PADERBORN
Computer Graphics, Visualization and Image Processtegrationg
Prof. Dr. Gitta Domik

# 5. Representation – Flow Visualization



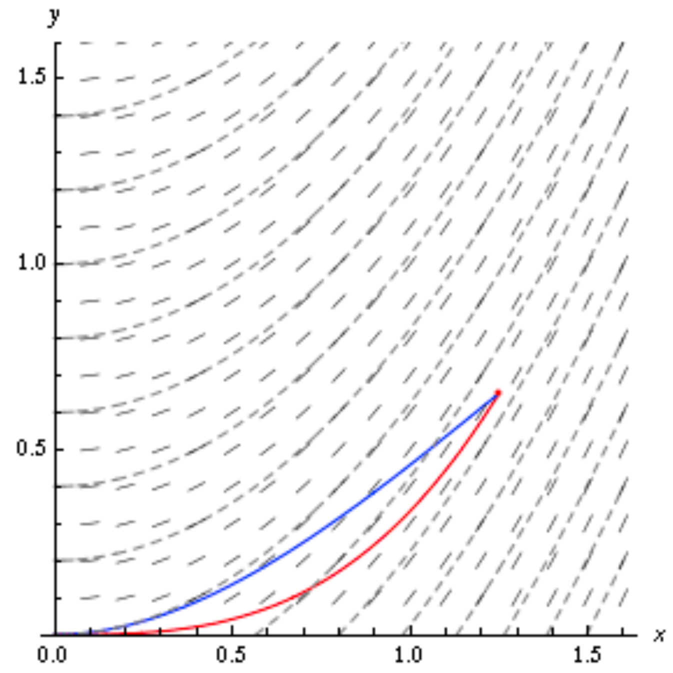



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization
and Image Processtegrationg
Prof. Dr. Gitta Domik

# Indirect Flow Viz: Pathlines, streamlines, streaklines

- **Representative data characteristics**
  - vector fields; algorithms discussed for regular grids only!
- **Techniques [POS94], [HEL94]** (Algorithms typically work on grid-sampled data!)
- **Streamlines:** velocity vectors of flow are tangent to streamlines and show the direction a fluid element will travel at any point in time
- **Streaklines**: dye steadily injected into the fluid at a fixed point extends along a streakline.
- **Pathlines** are the trajectories that individual fluid particles follow (recording of the path a fluid element in the flow takes over a certain period).
- stream ribbons: surface between two adjacent stream lines
- stream surfaces: surface defined by set of adjacent stream lines
- **Special note on interaction**
  - special interaction tools (DataGlove) and methods (gesturing) possible
  - changes in time usually depicted as motion

# Pathlines and Streamlines over 2d Vector Field

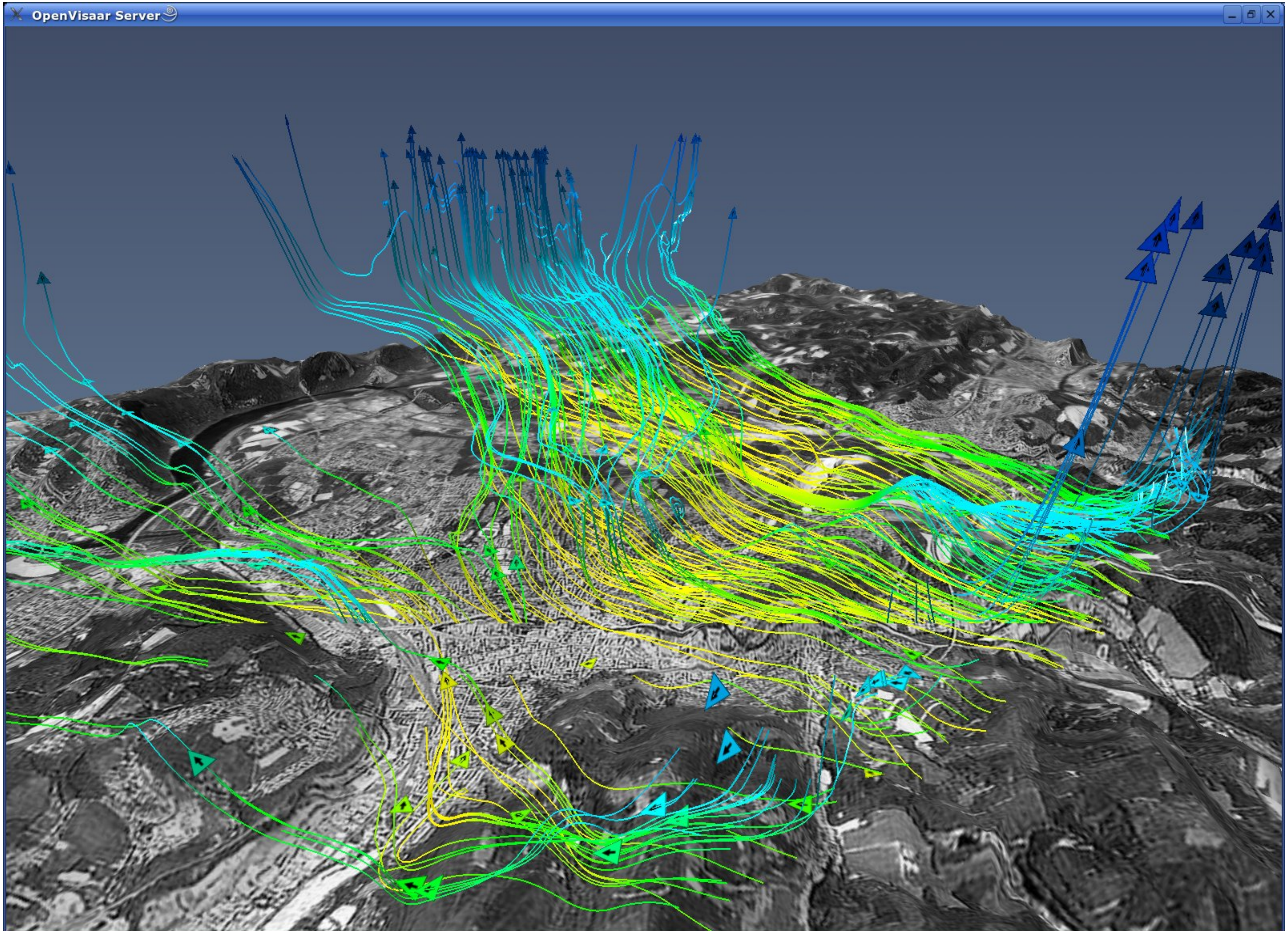



Dashed streamlines
Red pathline
Blue Streakline

http://en.wikipedia.org/wiki/File:Streaklines_and_pathlines_animation_(low).gif

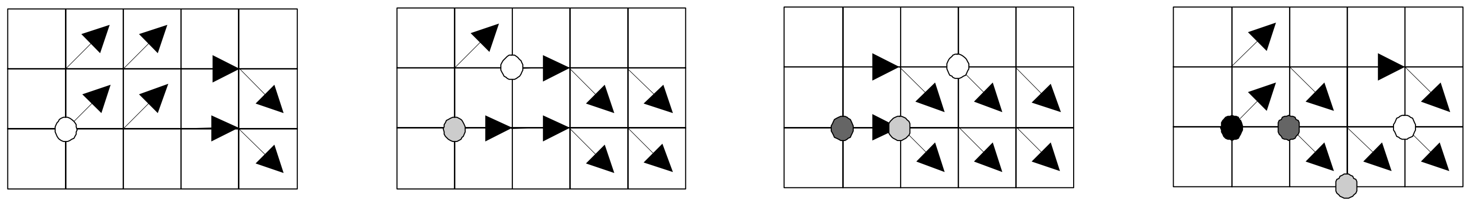


UNIVERSITY OF PADERBORN
Computer Graphics, Visualization
and Image Processtegrationg
Prof. Dr. Gitta Domik

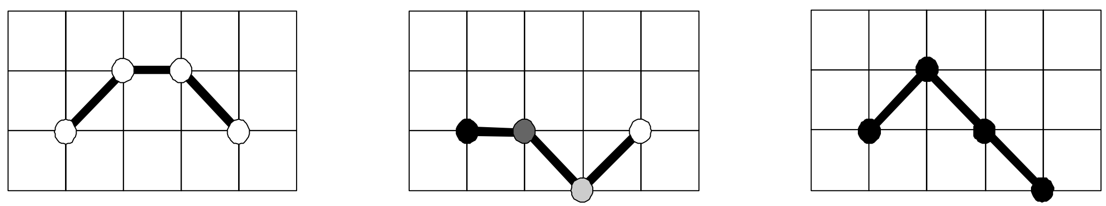

OpenVisaar Server

# Pathlines, Streamlines, Streaklines

(Algorithms typically work on grid-sampled data!)



$t_0$ $t_1$ $t_2$ $t_3$



Pathline $t_0$ to $t_3$ Streakline at $t_3$ Streamline at $t_3$

Release of white particle at $t_0$, light grey at $t_1$, dark grey at $t_2$, black at $t_3$.



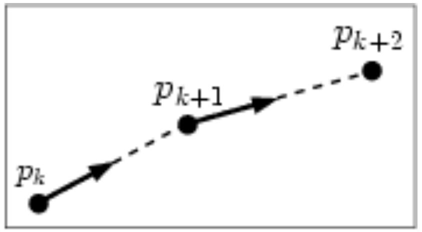
UNIVERSITY OF PADERBORN
Computer Graphics, Visualization and Image Processtegrationg
Prof. Dr. Gitta Domik

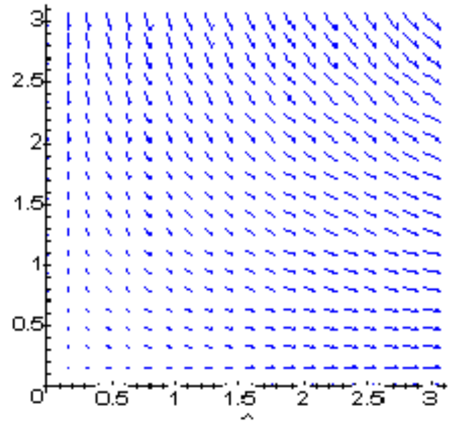
# Next Assignment: Redo Assignment 1 ...

- ... this time use STREAMLINES
- How to do this with the given vector field?
- Could you do streaklines or pathlines with the given data?

# Streamlines in more detail

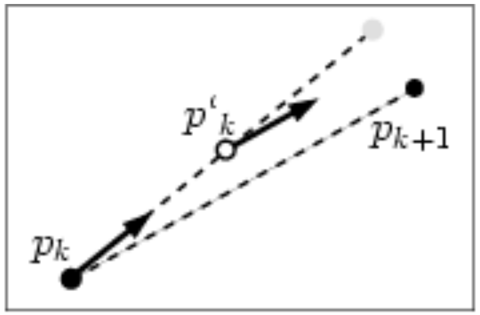
- Given a field of vectors v
- Each v is a tangent to the streamline at that point
- Therefore: streamline is integral of vector field
- But vectors only at discrete locations availabe
- Do numerical approximations
  - e.g. Euler integration
    $p_{k+1} = p_k + hv(p_k)$
    *h is integration step*




http://www.math.ucla.edu/~ronmiech/Calculus_
Problems/32B/chap14/section1/896d25/896_25.html

- E.g. second order Runge-Kutta integration
  $p'_k \quad = p_k + 1/2\, hv(p_k)$
- $p_{k+1} \quad = p_k + hv(p'_k)$




http://www.cgal.org/Manual/latest/doc_html/cgal_manual/
Stream_lines_2/Chapter_main.html

Example: TU Vienna, Eduard Gröller Helwig Hauser

## Integration of Streamlines

Numerical Integration

TU Vienna, Eduard Gröller Helwig Hauser

## Streamlines – Practice

- Basic approach:
  - theory: $\mathbf{s}(t) = \mathbf{s}_0 + \int_{0\le u\le t} \mathbf{v}(\mathbf{s}(u))\,\mathrm{d}u$
  - practice: numerical integration
  - idea:
    (very) locally, the solution is (approx.) linear
  - Euler integration:
    follow the current flow vector $\mathbf{v}(\mathbf{s}_i)$ from the current streamline point $\mathbf{s}_i$ for a very small time (d$t$) and therefore distance
  - Euler integration: $\mathbf{s}_{i+1} = \mathbf{s}_i + \mathrm{d}t \cdot \mathbf{v}(\mathbf{s}_i)$,
    integration of small steps (d$t$ very small)



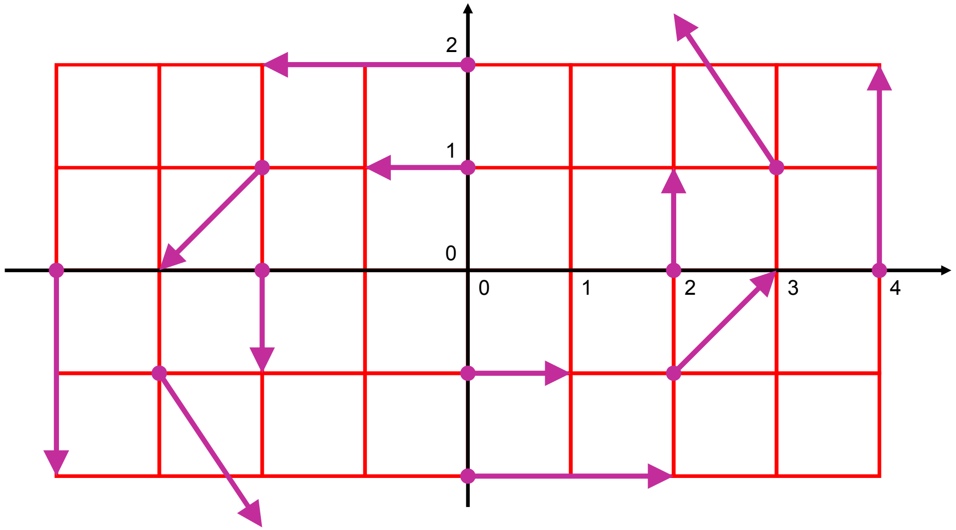
TU Vienna, Eduard Gröller Helwig Hauser



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization and Image Processtegrationg
Prof. Dr. Gitta Domik

## Euler Integration – Example

- 2D model data: $v_x = dx/dt = -y$
  $v_y = dy/dt = x/2$
- Sample arrows:
- True solution: ellipses!





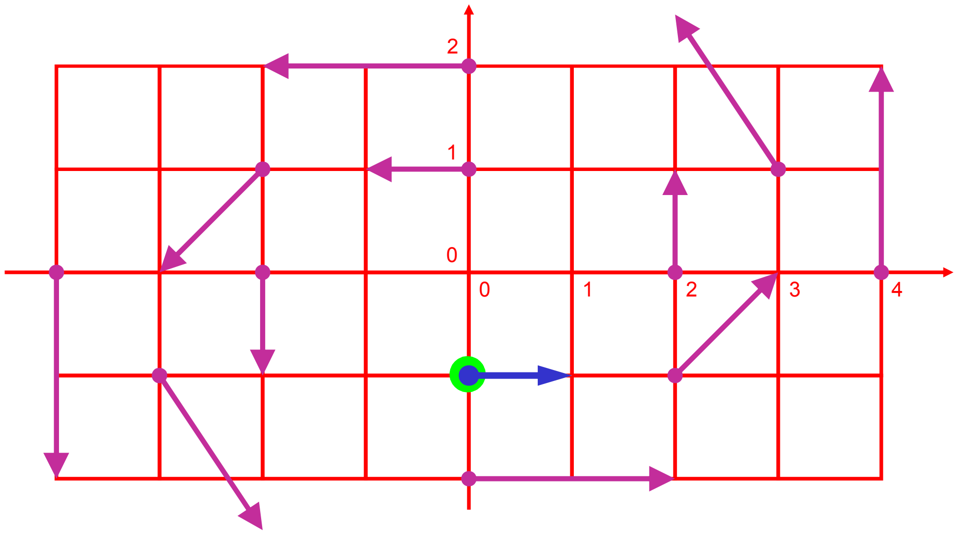

TU Vienna, Eduard Gröller Helwig Hauser



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization
and Image Processtegrationg
Prof. Dr. Gitta Domik

## Euler Integration – Example

- Seed point $\mathbf{s}_0 = (0|-1)^T$;
  current flow vector $\mathbf{v}(\mathbf{s}_0) = (1|0)^T$;
  $dt = 1/2$





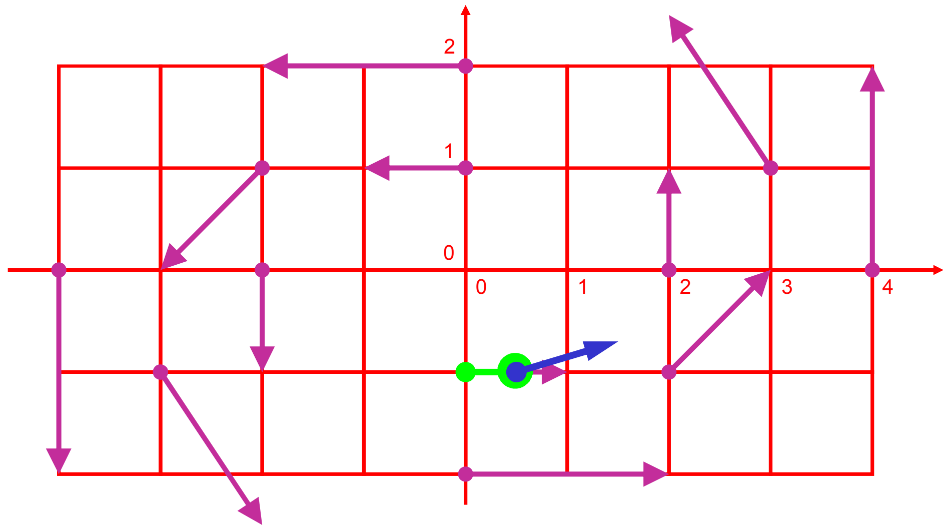

TU Vienna, Eduard Gröller Helwig Hauser



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization and Image Processtegrationg
Prof. Dr. Gitta Domik

## Euler Integration – Example

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





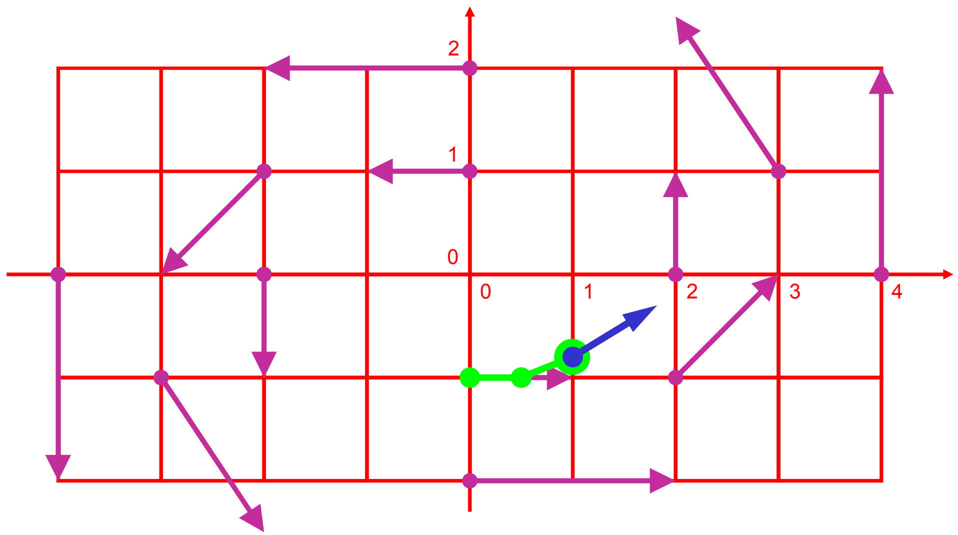

TU Vienna, Eduard Gröller Helwig Hauser



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization and Image Processtegrationg
Prof. Dr. Gitta Domik

## Euler Integration – Example

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

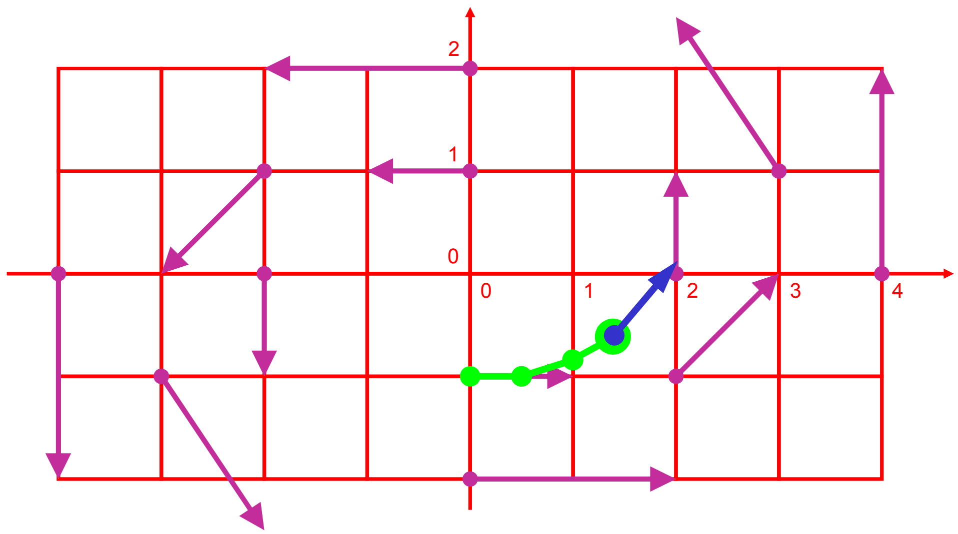





TU Vienna, Eduard Gröller Helwig Hauser

## Euler Integration – Example

- $\mathbf{s}_3 \quad = (23/16 \,|\, -5/8)^T \quad \approx (1.44 \,|\, -0.63)^T;$
  $\mathbf{v}(\mathbf{s}_3) \quad = (5/8 \,|\, 23/32)^T \quad \approx (0.63 \,|\, 0.72)^T;$

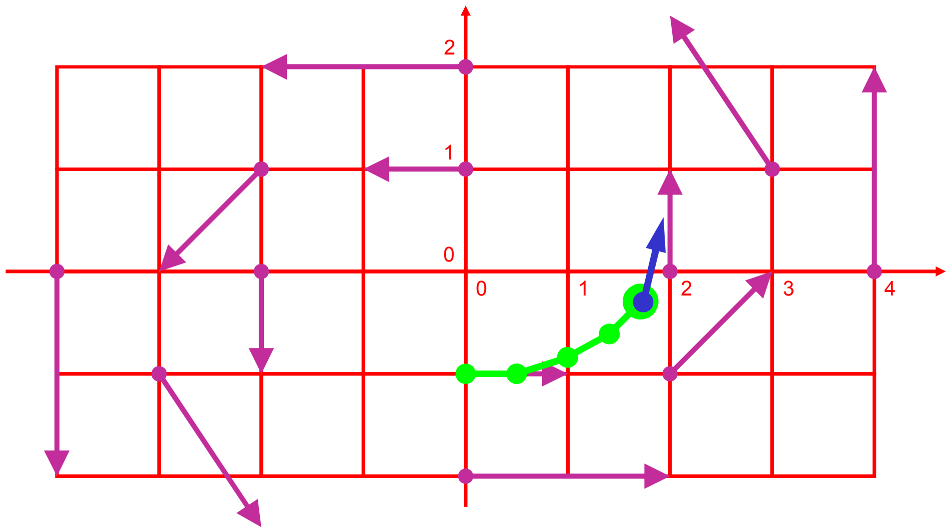



Helwig Hauser 39

TU Vienna, Eduard Gröller Helwig Hauser

## Euler Integration – Example

- $\mathbf{s}_4 \quad = (7/4\,|\,-17/64\,)^T \quad \approx (1.75\,|\,-0.27\,)^T;$
  $\mathbf{v}(\mathbf{s}_4) \quad = (\,17/64\,|\,7/8\,)^T \quad \approx (\,0.27\,|\,0.88\,)^T;$

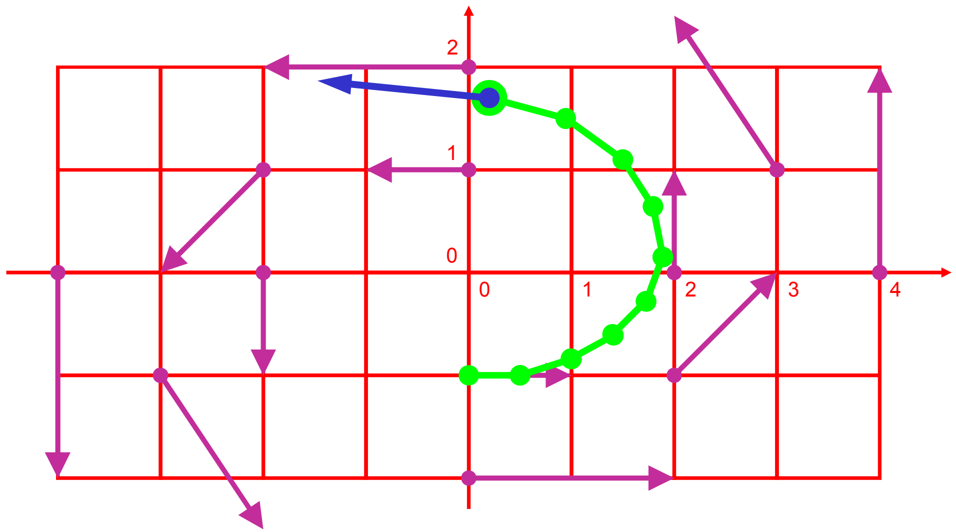



Helwig Hauser 40

TU Vienna, Eduard Gröller Helwig Hauser

## Euler Integration – Example

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

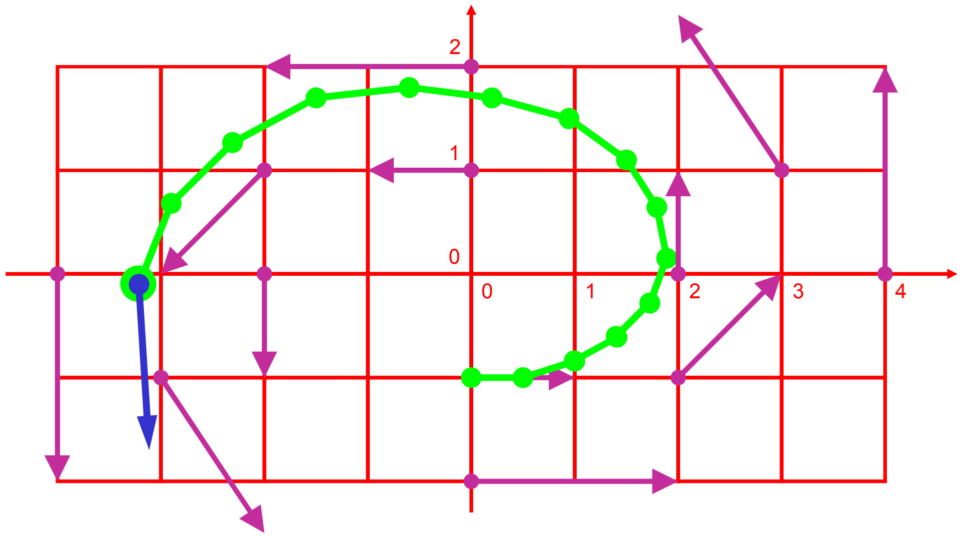





TU Vienna, Eduard Gröller Helwig Hauser

## Euler Integration – Example

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





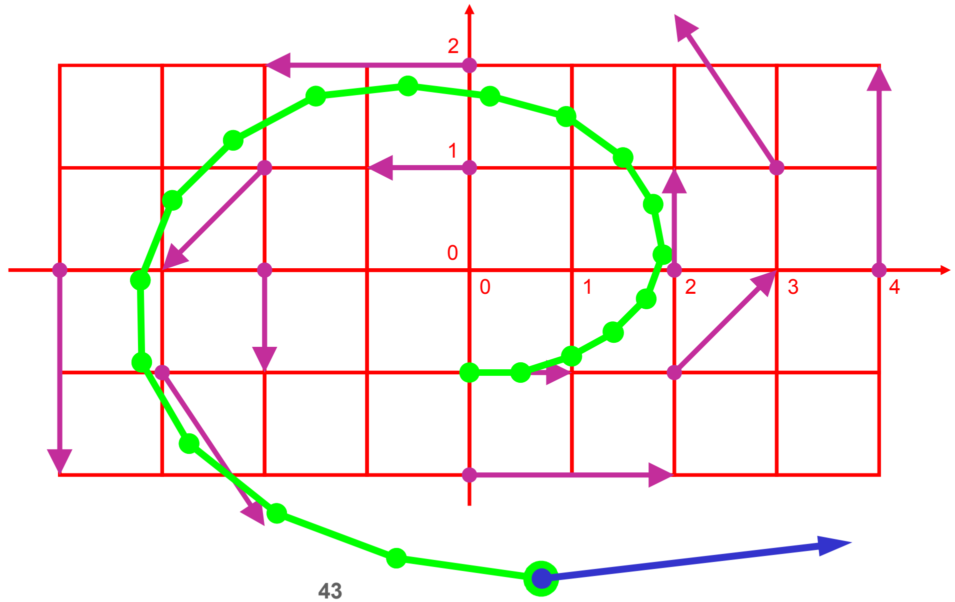

TU Vienna, Eduard Gröller Helwig Hauser



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization and Image Processtegrationg
Prof. Dr. Gitta Domik

# 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, d$t$ too large!
  19 steps

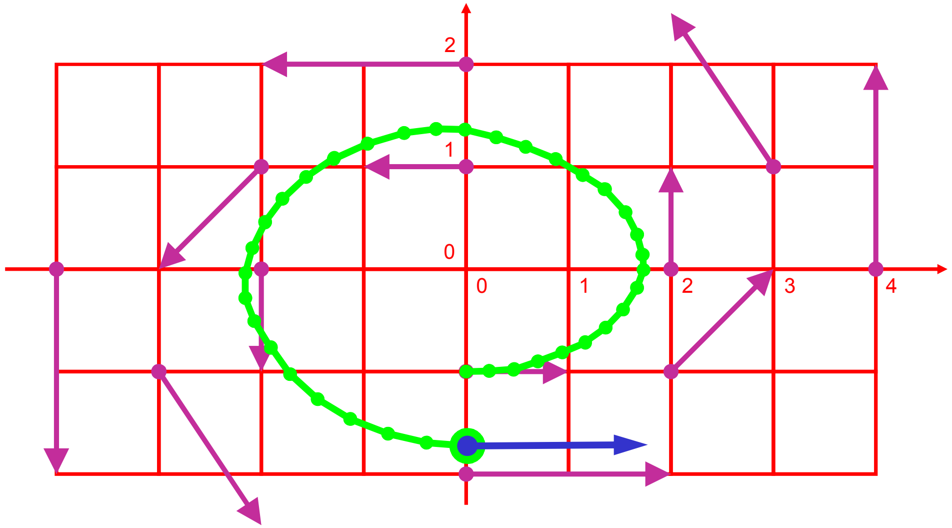



Helwig Hauser 43

TU Vienna, Eduard Gröller Helwig Hauser

# Euler Integration – Example

- d*t* 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




Helwig Hauser 44

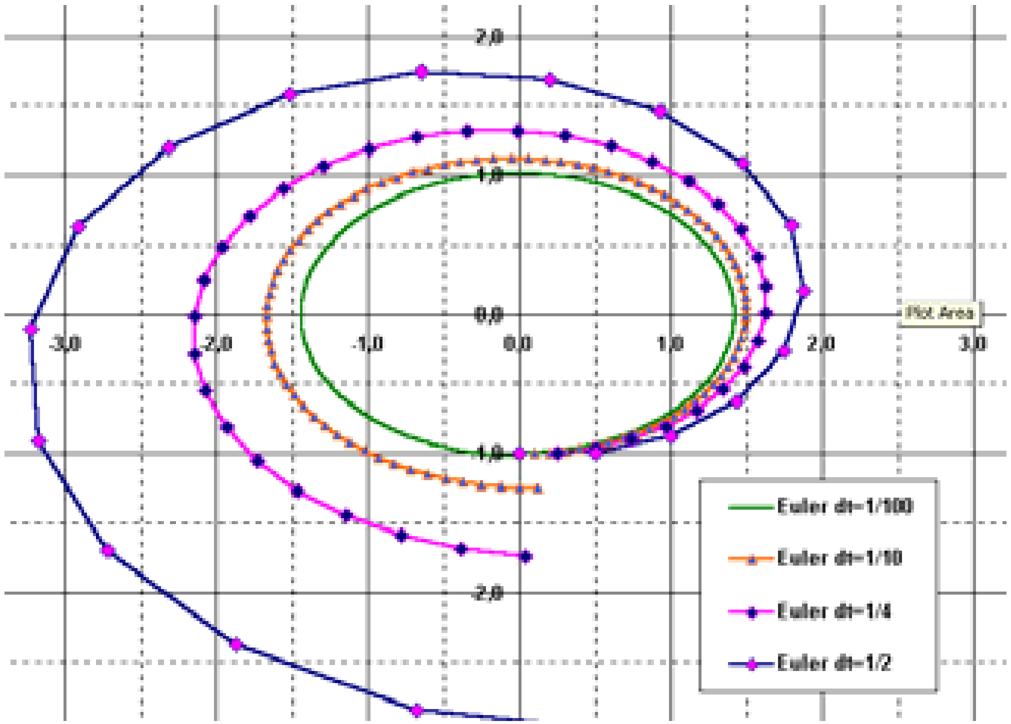
TU Vienna, Eduard Gröller Helwig Hauser



UNIVERSITY OF PADERBORN
Computer Graphics, Visualization
and Image Processtegrationg
Prof. Dr. Gitta Domik

# Comparison Euler, Step Sizes

Euler
is getting
better
propor-
tionally
to d*t*




TU Vienna, Eduard Gröller Helwig Hauser

## 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% | ✓ |



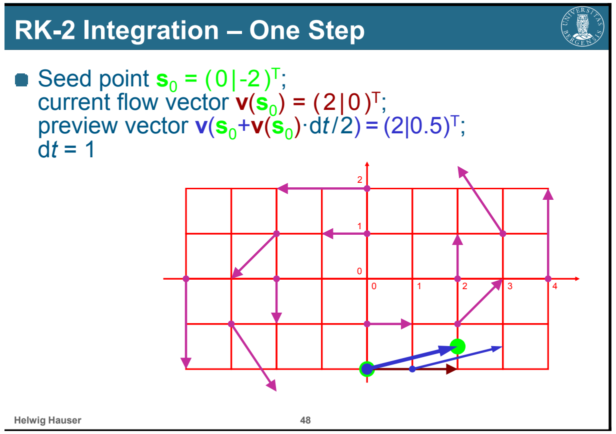
TU Vienna, Eduard Gröller Helwig Hauser

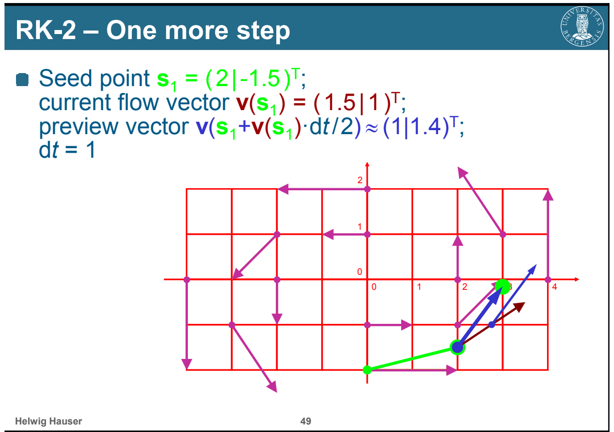


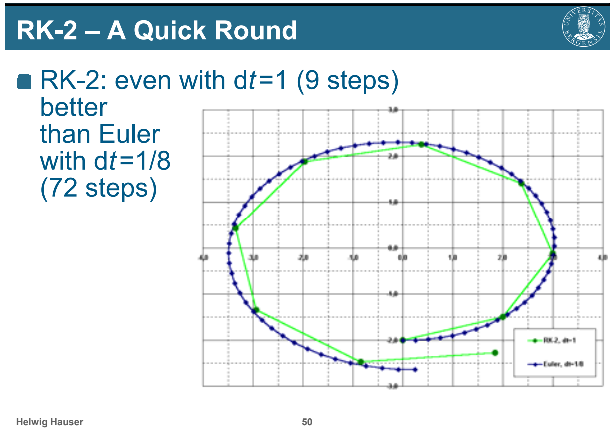
UNIVERSITY OF PADERBORN
Computer Graphics, Visualization
and Image Processtegrationg
Prof. Dr. Gitta Domik

# Runge-Kutta 2nd order

- $P_{i+1} = P_i + h\, v(K)$; with $K = P_i + ½\, h\, v(P_i)$




TU Vienna, Eduard Gröller Helwig Hauser

# Runge-Kutta 4th Order

$$P_{i+1} = P_i + h \cdot \left( \frac{1}{6} K_1 + \frac{1}{3} K_2 + \frac{1}{3} K_3 + \frac{1}{6} K_4 \right)$$

$$K_1 = v\left(P_i\right)$$

$$K_2 = v\left(P_i + \frac{h}{2} K_1\right)$$

$$K_3 = v\left(P_i + \frac{h}{2} K_2\right)$$

$$K_4 = v\left(P_i + \frac{h}{2} K_3\right)$$

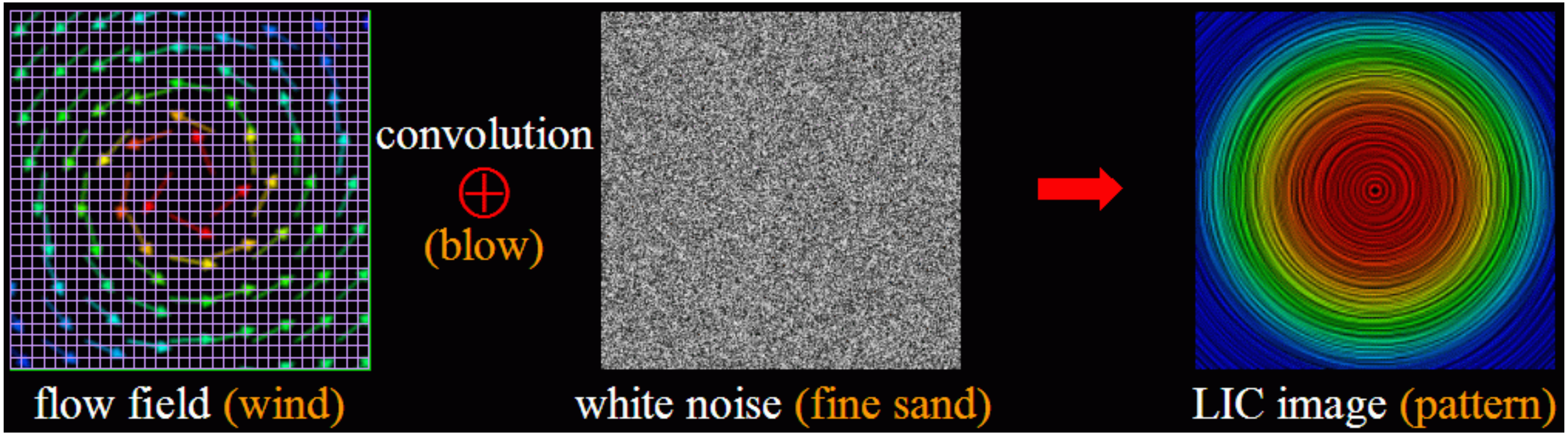
with h = integration step

# Next Assignment: Redo Assignment 1 ...

- Use 4th order Runge-Kutta to compute streamlines
- Use sensible seeds

# Local Integral Convolution (LIC)

- Goal: General overview of flow
- Use textures/pattern to show visual correlation between flow and pattern



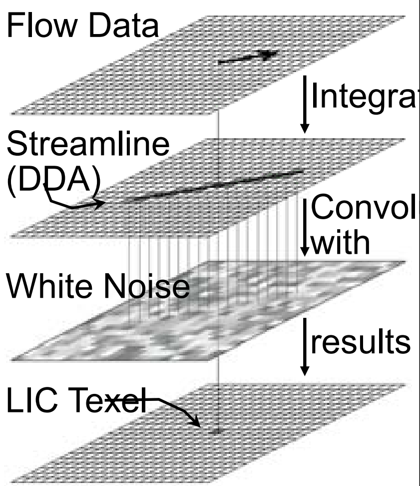

„Convolve“ the left two images to receive the right image
From www.zhanpingliu.org/

# Best references for LIC

**Imaging Vector Fields Using Line Integral** Convolution
Brian Cabral, Leith (Casey) Leedom
Lawrence Livermore National Laboratory

**Fast and Resolution Independent Line Integral Convolution**
Detlev Stalling and Hans-Christian Hege
Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

- Calculation of a texture value:
  - look at streamline through point
  - filter white noise along streamline




Helwig Hauser 76

# Examples: Ribbons, Surfaces

**[HEL90]**

# Example: ribbon around critical point

[HEL90]

# Local versus Global View

- Depicting local vector: arrows, glyphs; watch for critical points
- Depicting global view: pathlines, streamlines, LIC: where does the flow come from and where will it go? Give overview of flow.

# Summary

- 2 d, 3d, and in between
- Steady vs time dependent flows
- Direct flow viz: arrows, glyphs
- Indirect flow: pathlines, streamlines, streaklines, LIC
- Local versus global
- Euler integration
- Runge-Kutta integration