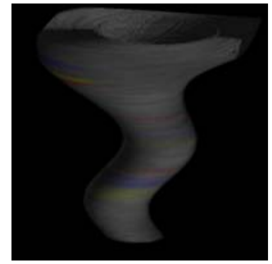
# Chapter 6 – Visualization Techniques for Vector Fields

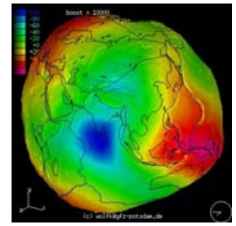




# 6.1 Introduction

- Vector fields are common in science and engineering:
  - Displacement fields in elasticity theory, velocity fields in computational fluid dynamics (CFD), force fields (e.g. gravitation), displacement fields
- In general vector fields have an orientation and are then termed *signed*, though *unsigned* vector fields, such as eigenvector fields, also exist.

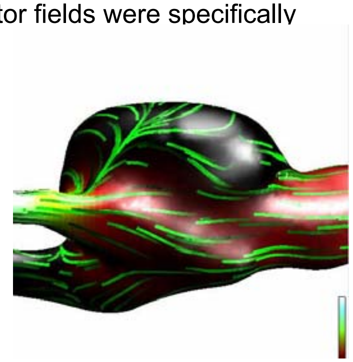






# Introduction (cont'd)

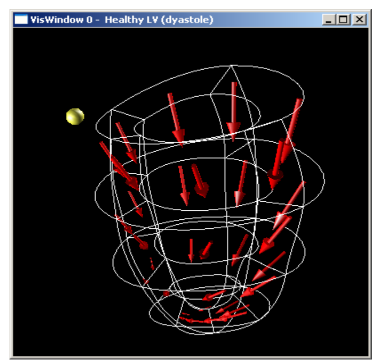
- Many visualisation techniques for vector fields were specifically developed for velocity fields.
  - *Steady flows* are constant over time.
  - *Unsteady flows* vary over time
    - Laminar flows are characterized by layers of fluid elements with similar velocities
    - Turbulent flows the velocities in neighbouring fluid elements vary randomly.





# 6.2 Vector Glyphs

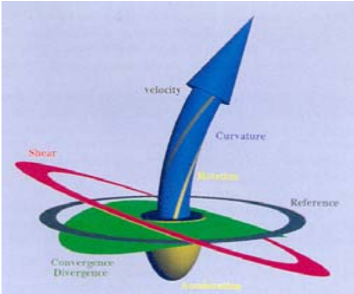
- Draw arrow or line segment in the direction of the vector with length equal to the vector magnitude.
  - Advantages:
    - Good perception of visualized data (use illuminated volumetric icons for 3D vector field visualization).
  - Disadvantages:
    - Not clear which data point vector represents
    - Leads to visual cluttering
    - Requires a lot of screen space
    - Easy to miss important features



## Vector Glyphs (cont'd)

- Flow field probe (de Leeuw & van Wijk)
  - Visualises additionally neighbourhood information derived from the local velocity gradient.

- The length, curvature and the candy stripes of the cylindrical shaft visualise magnitude, local streamline curvature and rotation of the flow field.
- The half ellipsoid at the bottom of the shaft encodes acceleration of velocities.
- The bending circular membrane describes convergence or divergence.
- The angle of the ring shaped surface with respect to a reference frame encodes shear.

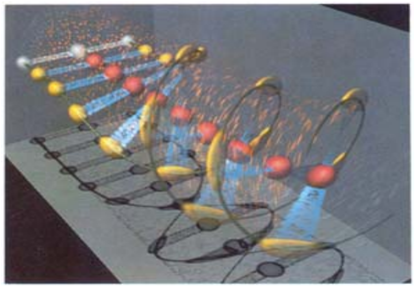



© 1994, Frits H. Post and Jarke J. van Wijk, Visual representation of vector fields: recent developments and research directions, in "Scientific Visualization: Advances and Challenges", Academic Press.



## 6.3 Particle Advection

- Distribute a set of particles over the domain and advect them with the vector field.
  - Flow direction and speed can be emphasized by blurring the particles.
  - Intuitive and easily understood for visualising fluid flows.
  - Well suited for turbulent flows where icons computed by integral curves and surfaces become highly irregular.
  - Lack of interactivity if the particle number is too high.
  - Difficulties in perceiving the 3D structure of the flow.



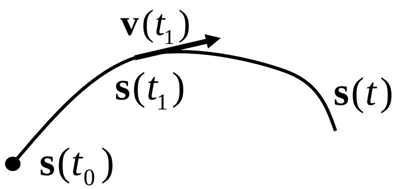
© 1994, Frits H. Post and Jarke J. van Wijk, Visual representation of vector fields: recent developments and research directions, in "Scientific Visualization: Advances and Challenges", Academic Press.



## 6.4 Streamlines

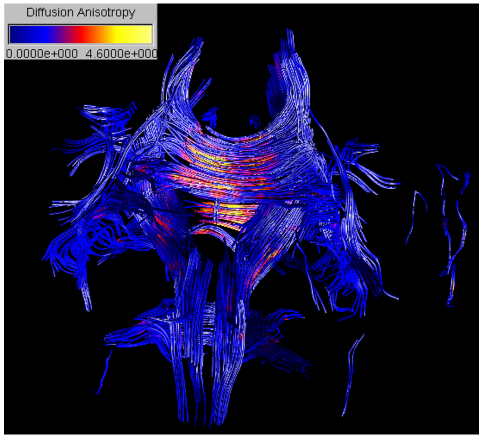
A streamline **s**($t$) for a vector field **v**($t$) is defined as the solution to the differential equation

$$\frac{\mathbf{s}(t)}{dt} = \mathbf{v}(\mathbf{s}(t)); \quad s(0) = x_0$$






## Streamlines (cont'd)




- Need shading and occlusion to better perceive the 3D geometry of streamlines
  - Fit thin tube around the lines

## Streamlines (cont'd)

- A simple algorithm to compute a streamline:
  - Approximate streamline by polyline $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\ldots$

    where $\mathbf{x}_i = \mathbf{s}(t_i)$

    $$t_0 = 0,\; t_{i+1} = t_i + \Delta t$$

    and $\mathbf{x}_i$ is computed by $\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t\; \mathbf{v}(\mathbf{x}_i)$



## Streamlines (cont'd)

The Problem …

- The above simple minded ODE solving method is called *Euler's Method*
- Errors accumulate steadily
- Can be unstable
  - e.g. imagine too long a time step
    - Vector at two time steps has opposite direction
    - Converging solution can blow up!
- Need to use very small step sizes to get tolerable results
  - Expensive and inefficient
- Need a better method to solve differential equations



## The mid-point method

- Can write $\mathbf{x}_{i+1}=\mathbf{s}(t + \Delta t)$ as a Taylor expansion

$$\mathbf{s}(t+\Delta t) = \mathbf{s}(t) + \Delta t\,\frac{d\mathbf{s}}{dt} + \frac{\Delta t^2}{2}\,\frac{d\mathbf{s}^2}{d^2 t} + \cdots$$

- Euler's method takes first two terms on RHS. Improve by taking more.
- If take three, get mid-point method:

$$\mathbf{s}(t+\Delta t) = \mathbf{s}(t) + \Delta t\left(f\left(\mathbf{s}(t) + \frac{\Delta t}{2} f(\mathbf{s}(t))\right)\right)$$

$$\text{where} \quad f(\mathbf{s}(t)) = \frac{d\mathbf{s}}{dt}$$

- In words:
  - compute Euler step to get first guess at **s**(t+ Δ*t*)
  - Determine mid point **s**(t)+**s**(t+ Δ*t*)
  - Evaluate d**s**/dt (i.e. the vector field) at this point
  - Use this value to compute a new step

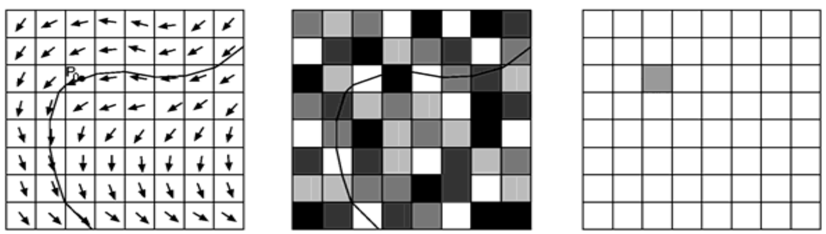


## Other issues

- Even mid-point method often not good enough
  - Use even higher-order methods, e.g. fourth-order Runge Kutta
- Need adaptive step sizes for best efficiency - use long time steps when things are moving slowly, short ones when changes are rapid.
- Test for vector field singularities
- Fundamental limitation of "explicit" ODE solvers:
  - Don't work well for "stiff" equations (common in computational fluid dynamics)
  - Better to use *implicit* methods

# 6.5 Line Integral Convolution

- Convolute noise texture with vector field
- Equivalent to averaging weighted pixel intensities along small streamlines





# Line Integral Convolution (cont'd)

- For any pixel *I(q,r)* of the input texture the centre $\mathbf{p}_0=(q+0.5,r+0.5)$ of it is used as the centre of a streamline which is advected forwards and backwards by a length *L*.
- The pixels intersected by the streamline in the forward direction have the indices $(\lfloor p_{i,x} \rfloor, \lfloor p_{i,y} \rfloor)$ where $\mathbf{p}_i = \mathbf{p}_{i-1} + \frac{\mathbf{v}(\mathbf{p}_{i-1})}{\|\mathbf{v}(\mathbf{p}_{i-1})\|}\Delta s_{i-1}$ and $\Delta s_{i-1}$ is the distance to the pixel boundary and $s_0 = 0,\ s_{i=1} = s_i + \Delta s_i$
- Pixels intersected in the backward direction are computed analogously and are indicated by negative indices.
- For each line segment $[s_i, s_{i+1}]$ of the streamline intersecting pixel $\mathbf{p}_i$ an exact integral of a convolution kernel *k(w)* is computed and used as weight in the LIC

$$h_i = \int_{s_i}^{s_i+\Delta s_i} k(w)\,dw$$



# Line Integral Convolution (cont'd)

- The output pixel O(q,r) is then given by

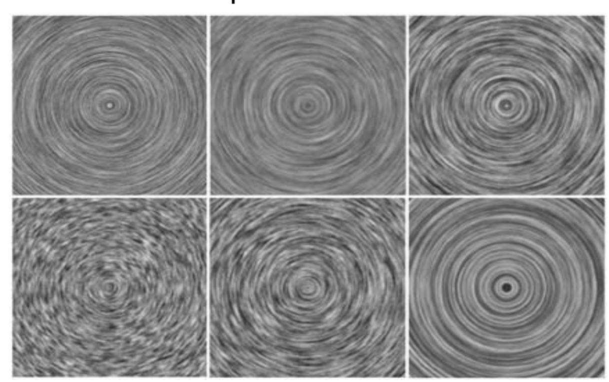
$$O(q,r) = \frac{\sum_{i=-l}^{l} I\left(\lfloor p_{i,x} \rfloor, \lfloor p_{i,y} \rfloor\right) h_i}{\sum_{i=-l}^{l} h_i}$$

- In the simplest case the convolution kernel is a box filter so that the output texture represents the weighted input texture along the streamline.
- Vector magnitude is represented either by using colour mapping or by varying the length *L* of the filter kernel.



# Line Integral Convolution (cont'd)

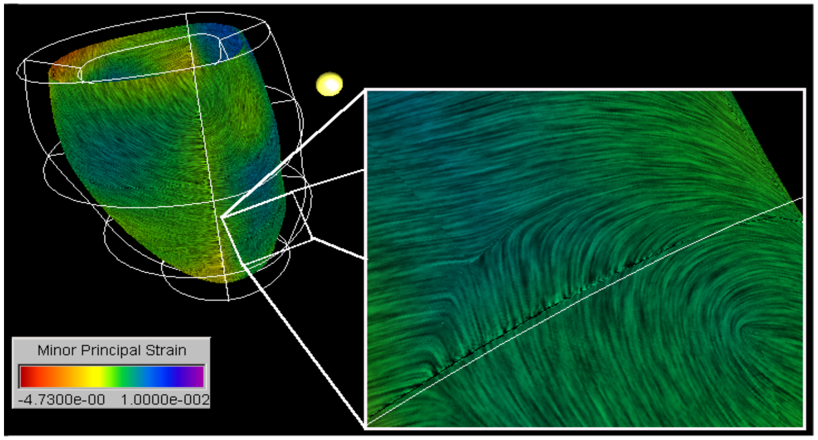
- Influence of parameters



Top row: LIC with a kernel length of 40. From left to right: using white noise, using low pass filtered white noise, using low pass filtered white noise and contrast stretching the output texture.

Bottom row: kernel length of 10,20, and 160. All images are contrast stretched and use low-pass filtered white noise

## Line Integral Convolution (cont'd)






## 6.6 Vector Field Topology

- A vector field **v**(**x**) can be characterized by considering its *critical points* which are points with zero vector magnitude.
- Critical points are the only points where streamlines are non-parallel and therefore indicate important flow features.
- A critical point $\mathbf{x}_0$ can be classified by considering the eigenvalues of the Jacobian

$$\mathbf{J}_{\mathbf{v}}(\mathbf{x}_0) = \left( \frac{\partial v_i}{\partial x_j} \right) \Bigg|_{\mathbf{x}_0}$$

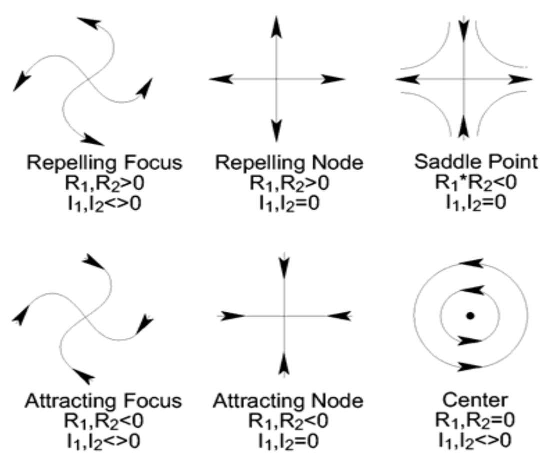


## Vector Field Topology (cont'd)

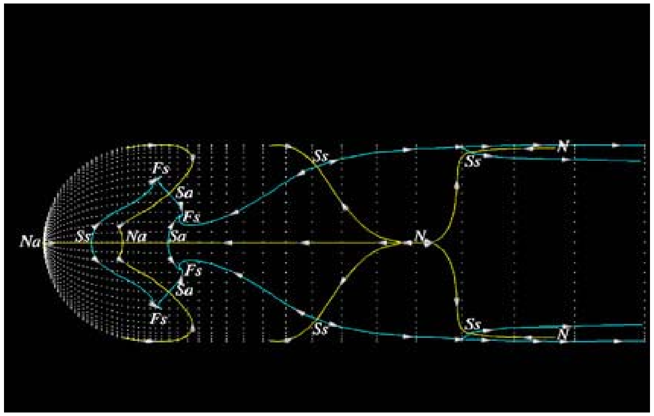
- The type of a critical point indicates the flow pattern in its immediate neighbourhood.
- In two dimensions the Jacobian of a vector field is a 2x2 matrix and therefore has two eigenvalues with real components $R_1$ and $R_2$ and imaginary components $I_1$ and $I_2$.
- The type of a critical point and hence the local flow topology depends on the signs of these components.
  - Real components greater or smaller than zero represent repelling or attracting flow features, respectively.
  - Non-zero imaginary components symbolise circular flows.



## Vector Field Topology (cont'd)



# Vector Field Topology (cont'd)




The vector field topology is obtained by connecting critical points by special streamlines





# 6.7 References


- L. Rosenblum et al., Scientific Visualization - Advances and Challenges, Academic Press, 1994.
- Burkhard Wünsche, *Scientific Visualization*, chapter 4, In "A Toolkit for the Visualization of Tensor Fields in Biomedical Finite Element Models", PhD Thesis, 2003.
- Brian Cabral and Leith (Casey) Leedom, Imaging Vector Fields Using Line Integral Convolution", Computer Graphics (SIGGRAPH '93 Proceedings), vol. 26, pages 263-272, August 1993.
- Willem C. de Leeuw and J. J. van Wijk, A Probe for Local Flow Field Visualization, Proceedings of IEEE Visualization '93, pages 39-45, 1993.
- J. J. van Wijk, Rendering Surface Particles, Proceedings of IEEE Visualization '92, pages 54-61, 1992.
- James L. Helman and Lambertus Hesselink, Visualizing Vector Field Topology in Fluid Flows, IEEE Computer Graphics & Applications, vol. 11, no. 3, pages 36-46, May 1991.