Vector Field Visualisation

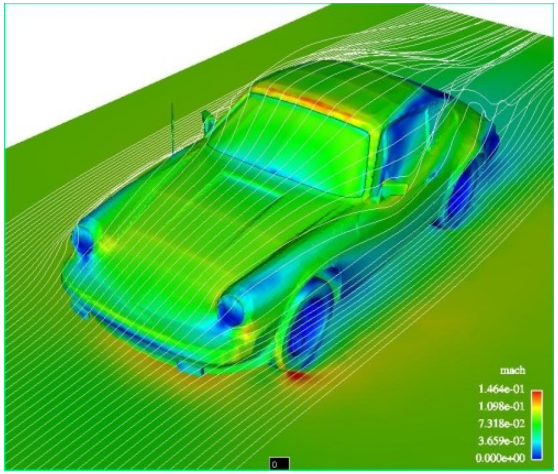
Visualisation – Lecture 12

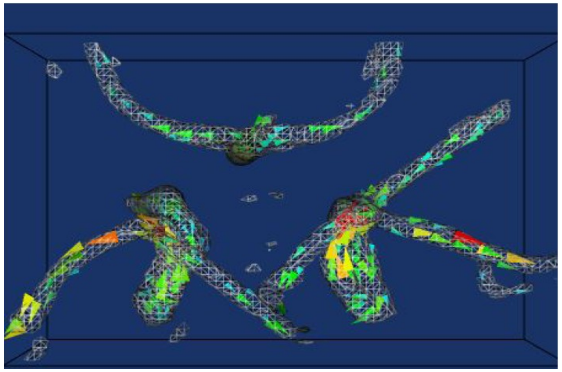
Taku Komura

Institute for Perception, Action & Behaviour
School of Informatics

Visualising Vectors

- Examples of vector data:
  - meteorological analyses / simulation
  - medical blood flow measurement
  - Computational simulation of flow over aircraft, ships, submarines etc.
  - visualisation of derivatives
    - not just of flow itself





- **Why is visualising these difficult ?**
  - **2 or 3 components** per data point, **temporal** aspects of vector flow, vector **density**

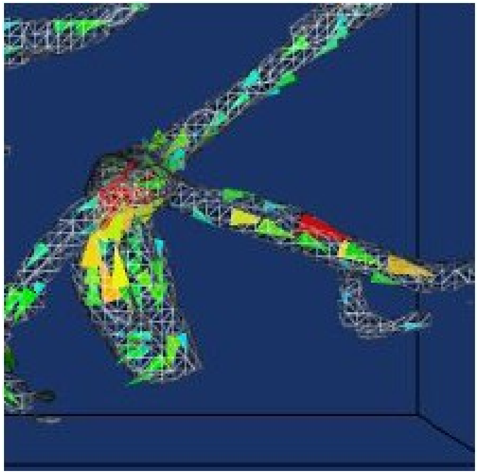
Insight in Vector Fields

- **Two properties** of vector fields to visualise :
  - **local view** of the flow
  - **global view** of the flow

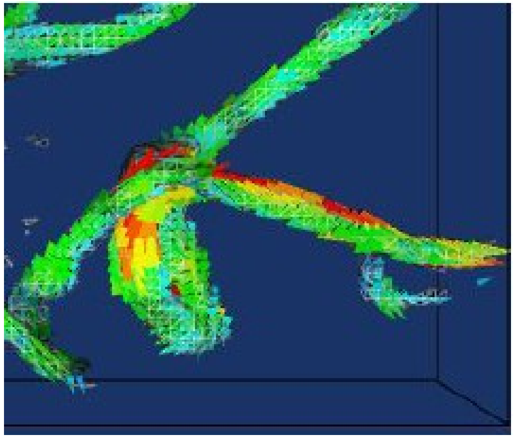
- e.g a meteorological wind forecast
  - **Local** : for given location, what is the current wind strength and direction
  - **Global**: a given location, where has the wind flow come from, and where will it go to.

Two Methods of Flow Visualisation

- Visualise Flow **wrt fixed point**
  - e.g. plot flow glyphs to show **local** direction and magnitude
  - **local view** of vector field

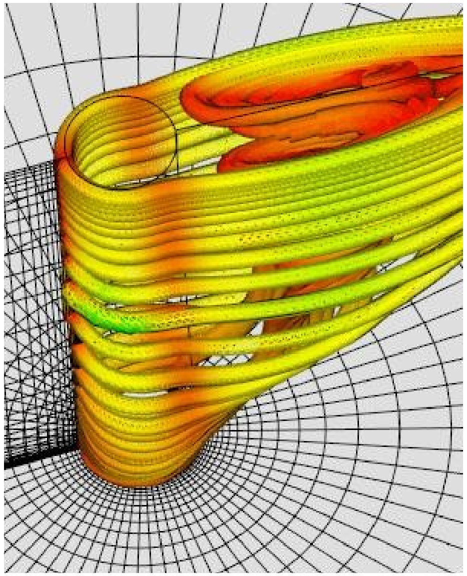


- Visualise flow as the **trajectory of a particles transported** by the flow
  - e.g plot particle traces, streamlines etc.
  - **global view** of vector field
  - require integrating the flow equation

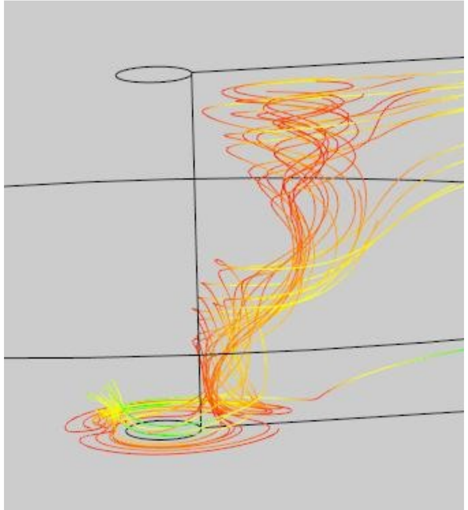


State of Flow : Steady / Unsteady

- **Steady flow**
  - remains **constant** over time
  - state of **equilibrium** or snapshot
  - use massless particle traces known as **streamlines**

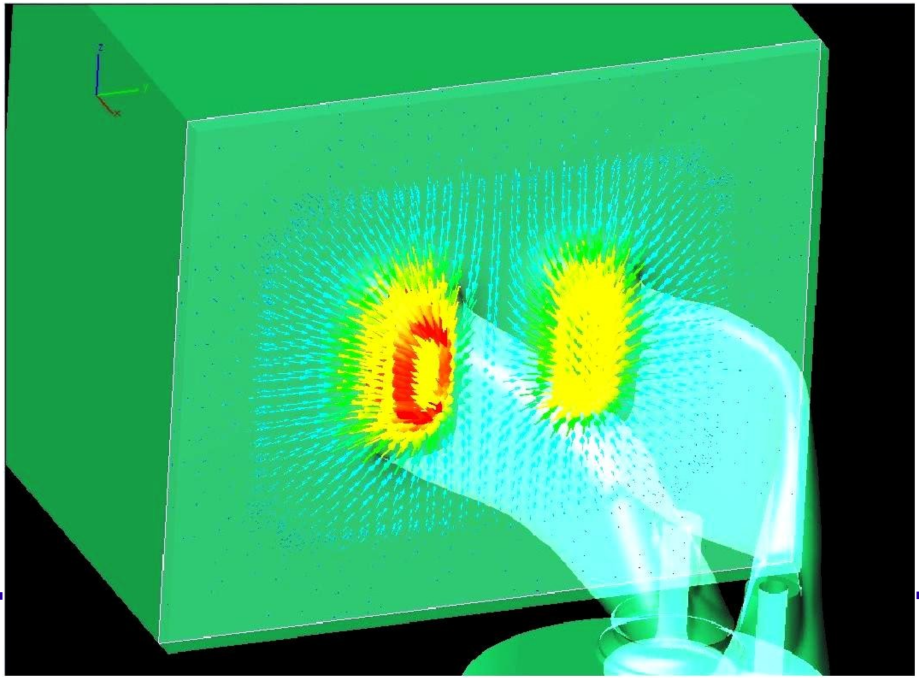


- **Unsteady flow**
  - **varies** with time
  - implications to tracing massless particles
  - particle traces known as **streaklines**
    - show little information about flow direction or magnitude



Vectors : local visualisation

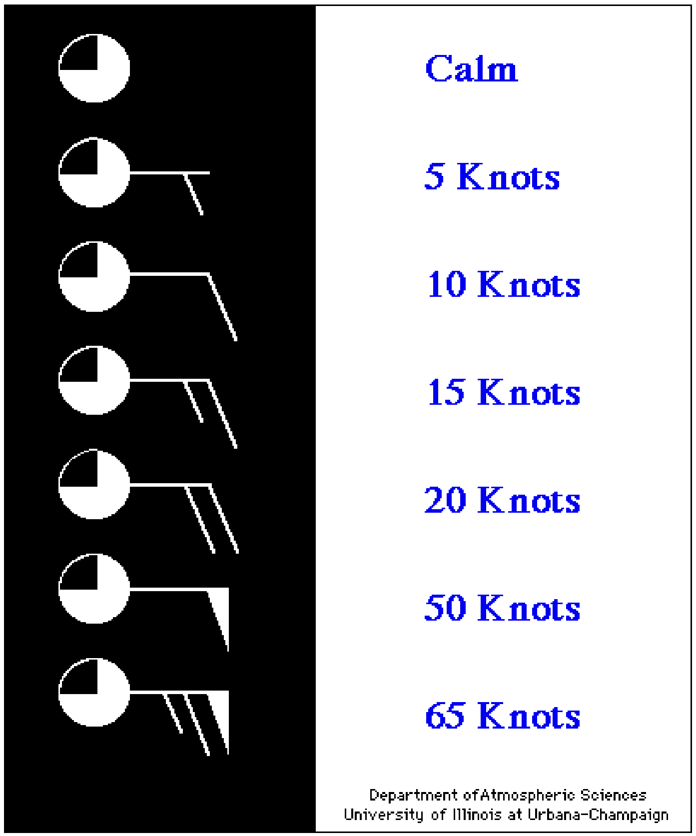
- Set of basic methods for showing **local view**:
  - oriented **lines**, **hedgehogs** & **glyphs**
  - **colour mapping** vector components *(lecture 5)*
  - **warping**
  - **animation**

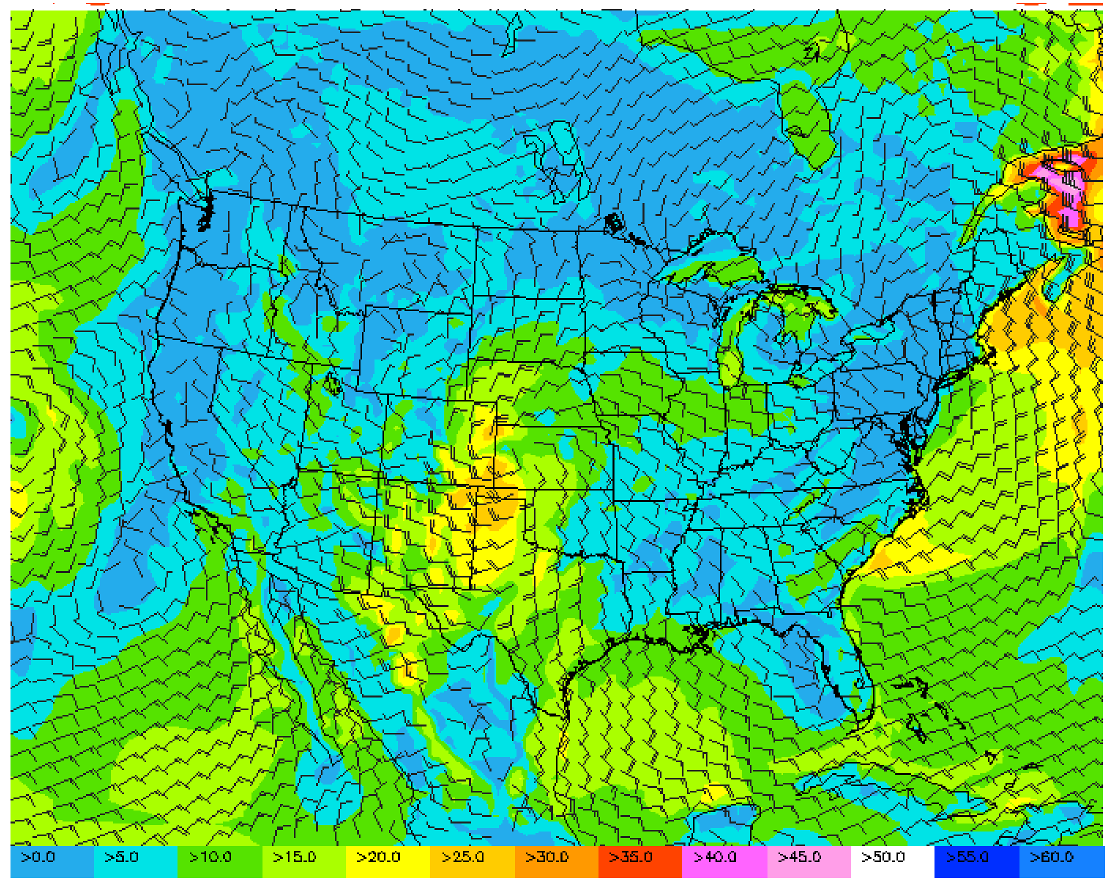


Local vector visualisation : lines

- **Draw line at data point indicating vector direction**
  - scale according to **magnitude**
  - **indicate direction** as vector orientation
- **problems**
  - non-uniform spacing
  - showing lower magnitude areas large dynamic range field
    - e.g. speed
- **Option** : use barbs to show speed
- Often referred to as **hedgehogs** !

Example : meteorology



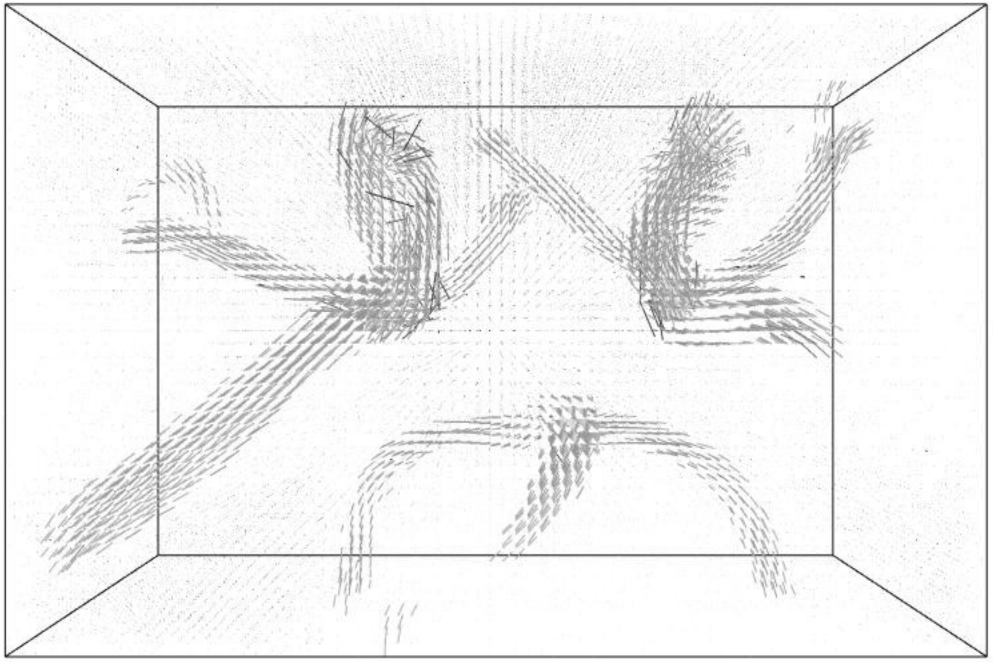


NOAA/FSL

Lines are drawn with constant length, *barbs* indicate wind speed. Also colour mapped scalar field of wind speed.

Example : lines in 3D

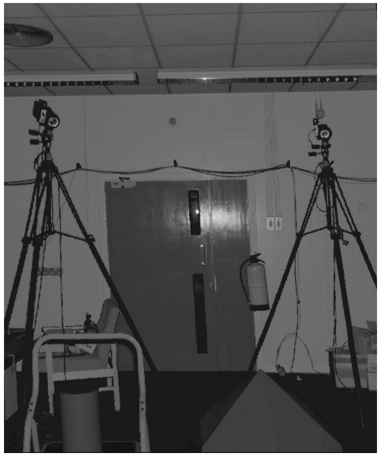
- **Problems** :
  - Difficult to understand position and orientation in projection to 2D image.
  - Clutter is also a problem.



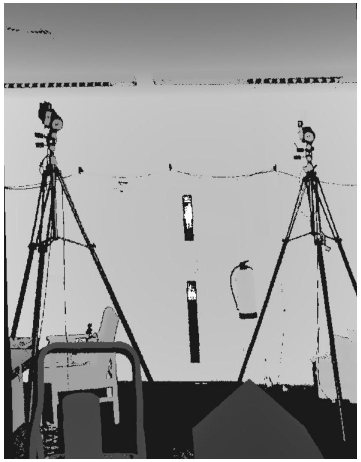
Local vector visualisation : colour map

Dense Normal Vectors in 3D capture of large scale environment

X component = Red.
Y component = Green
Z component = Blue.



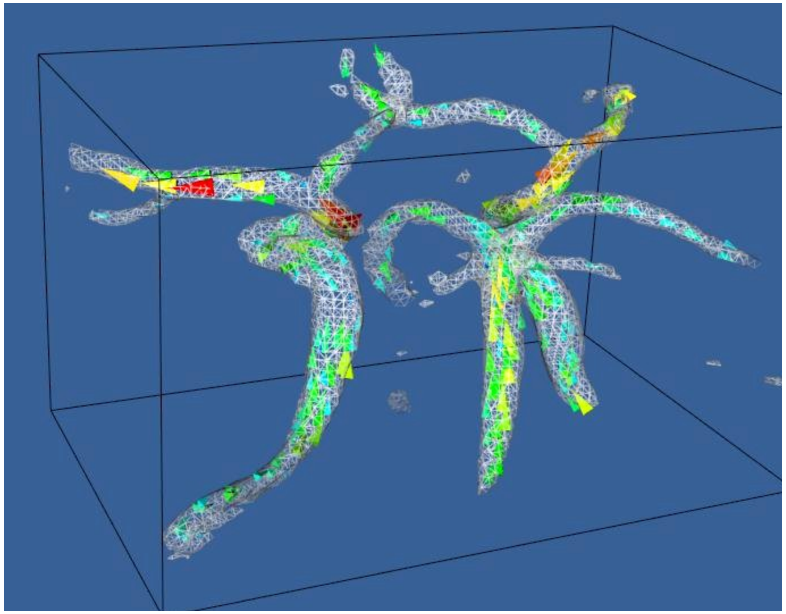
Returned laser power



Distance to object
(darker is closer – black is no data)

Local vector visualisation : Glyphs

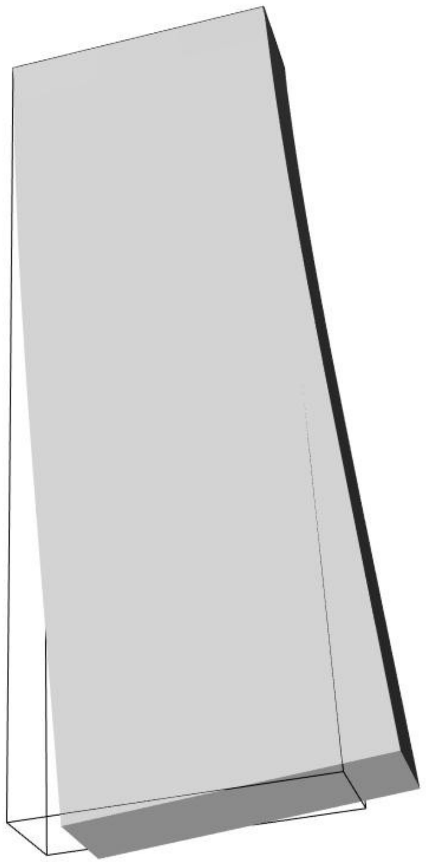
- **2D or 3D objects**
  - inserted at data point, oriented with vector flow
- **problem** : scaling
  - scaling glyph results in non-linear change in appearance
    - surface area changes with square of size
- **problem** : clutter



- e.g. blood flow (reduced data)
  - colourmap shows magnitude in addition to glyph scale

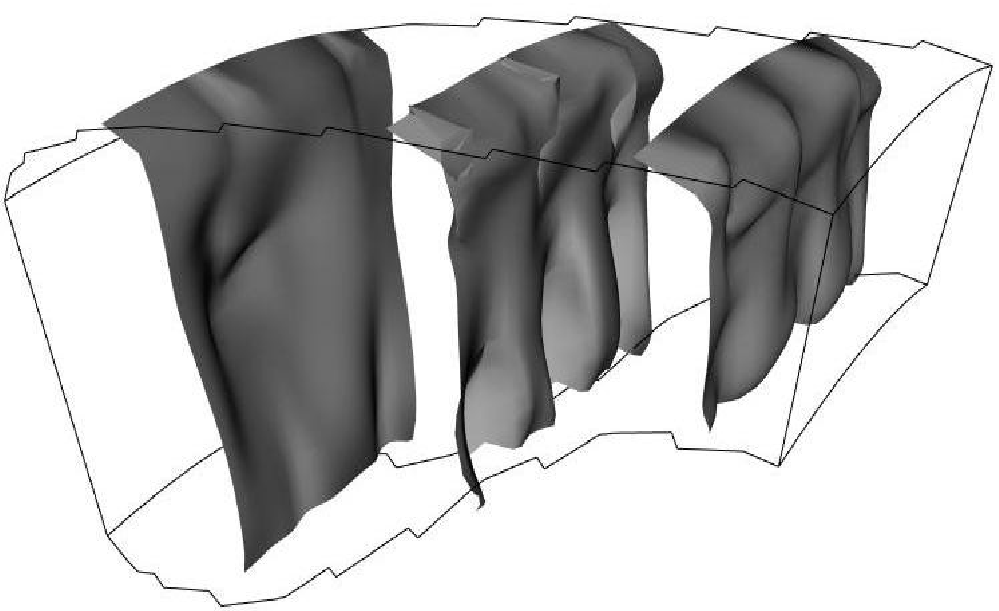
Local vector visualisation : Warping

- **deform geometry** according to the vector field
  - vector fields often associated with motion, or displacement.
  - e.g vibration of a beam.



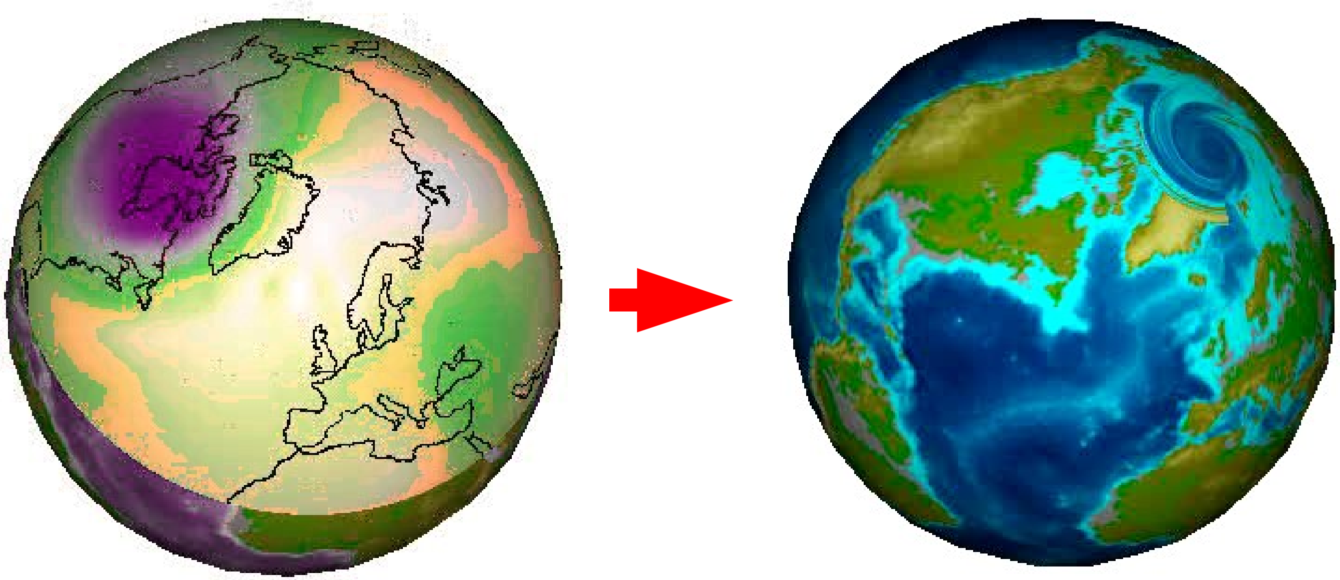
Example : warping

- Insert slice planes into the data volume
- Displace surface according to flow momentum
  - take care with scaling to **avoid excessive geometric distortion**
    - surfaces may intersect, or even turn inside-out



Local vector visualisation : animation

- **Animation** to enhance lines or glyphs
  - **improved clarity** of magnitude and/or **direction**
    - draw lines or glyphs & **animate over time**
  - **removes ambiguity** in line or glyph direction
  - also move glyphs along a streamline to visualise transport



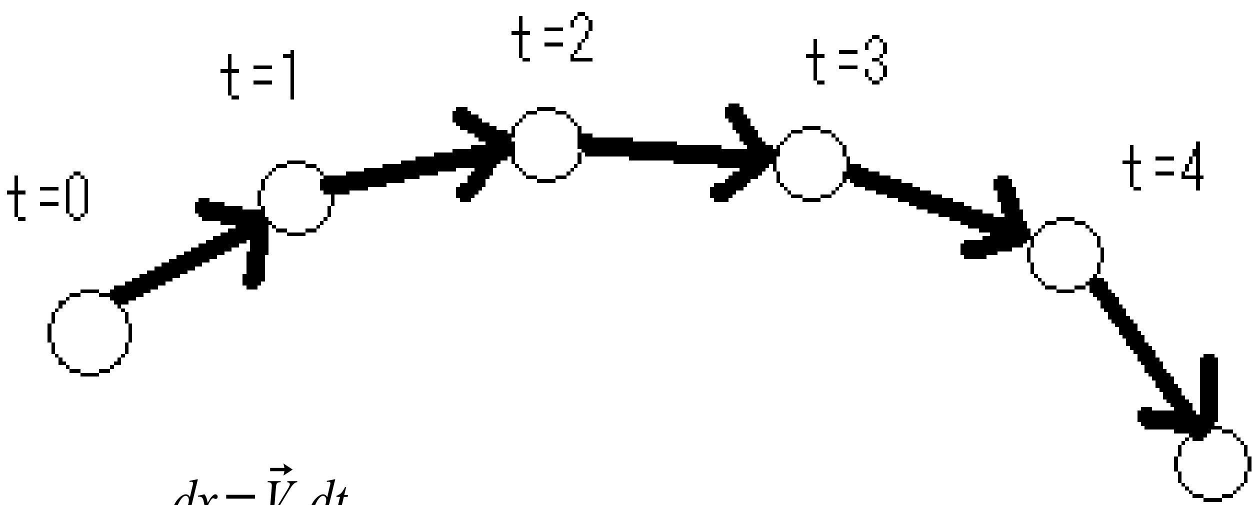
Vector Field Visualisation : local & global

- **Vector Fields** specify flows through the field
  - **aim** : visualise flow in field

- Two properties of vector fields to visualise
  - **local** view
    - with **respect to a fixed point**
    - e.g. glyphs, lines, warping, displacement etc.
  - **global** view
    - **trajectory of particles transported by vector field**

Particle trace

- **Particle trace :** the path over time of a massless fluid particle transported by the vector field
- The particle's velocity is always determined by the vector field



$$dx = \vec{V}\,dt$$

Express in integral form :

$$\vec{x}(t) = \int_t \vec{V}\,dt$$

Solve using numerical integration methods .

Stream & Streak lines : the difference

- **Streakline :** the set particle traces **at a particular time** that have previously passed through a specific point **(snapshot)**
  - **Path of the particles that were released from a point x0 at times t0< s < t**

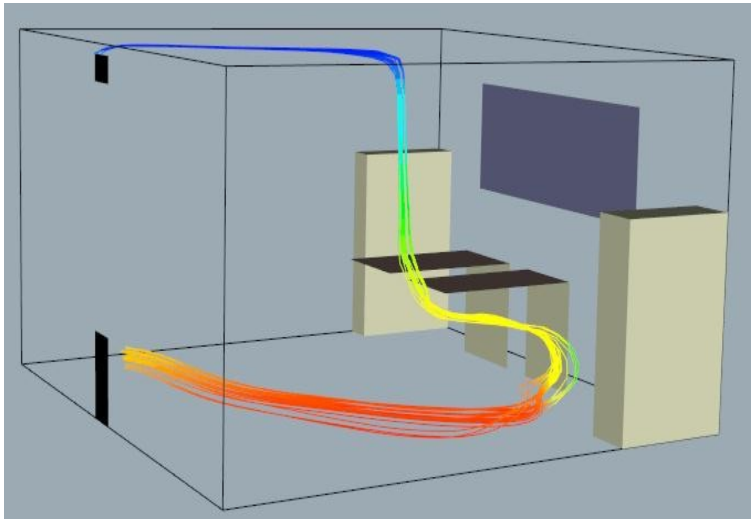
- **Streamline :** integral curves along a curve s satisfying:

$$\boxed{\begin{array}{l} s = \int_s \vec{V}\, ds, \;\; with \;\; s = s(x, \bar{t}) \\ at\ a\ fixed\ time\ \bar{t} \end{array}}$$

  - **Integral in the vector field while keeping the time constant**

Streamlines

- **Always** tangent to the vector field
  - Fluid **do not cross streamline**
  - streamlines **technically not particle traces**



- For **steady flows**
  - streamlines == streaklines
    - **2 are equivalent**
- For **unsteady flow**
  - Every streamline only exists at one moment in time
  - Always changing its shape

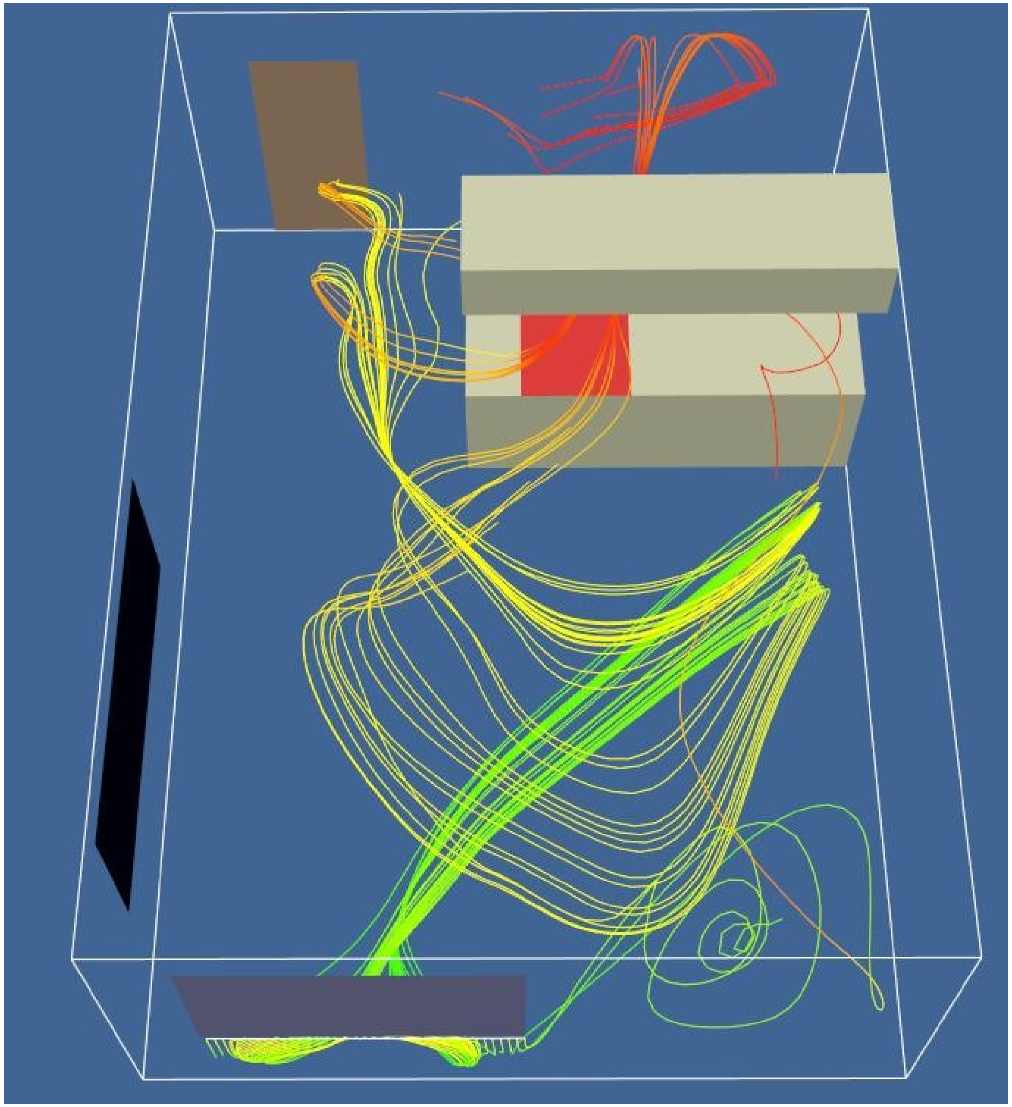
Example : convection streaklines

Ventilation simulation of a kitchen.
- Steady state or equilibrium.

Thirty **streaklines** initiated under a window.

Colour mapped (lecture 5) by air pressure (with is scalar).

Note the warm air convected by the hot stove.



Showing motion over time

- A scaled, oriented **line is an approximation to a particle's motion in the flow field**

$$\text{If velocity } V = \frac{dx}{dt}$$

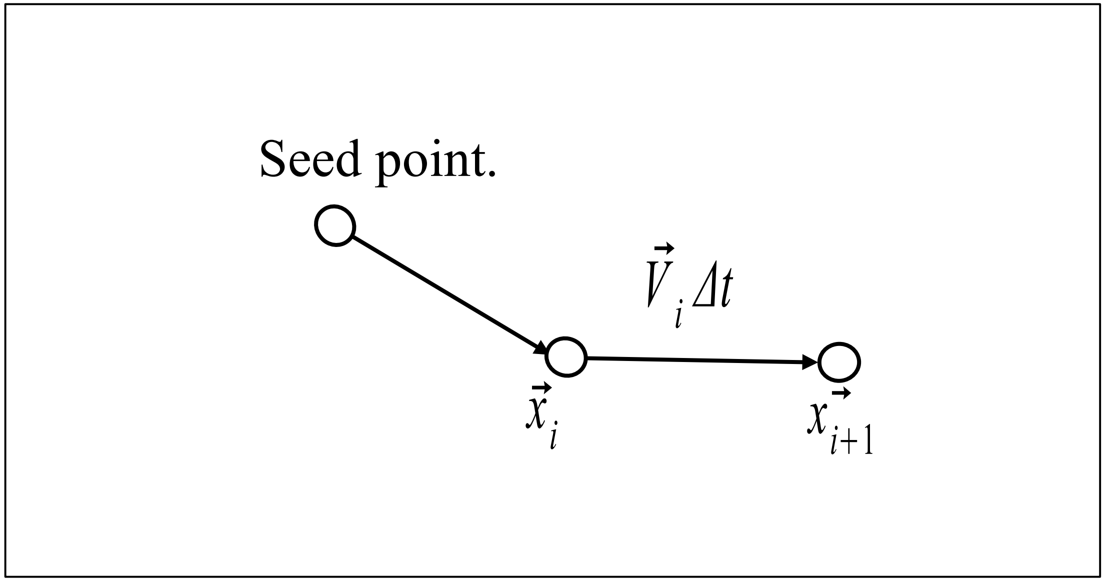
$$\text{Displacement of a point is } dx = \vec{V}\,dt$$

- Need to integrate in order to draw streamlines / streaklines

Numerical Integration

- **Numerical Integration** : beyond scope course
  - Accuracy depends on step size – *dt*
  - Results require careful examination
- **But ...** What do we mean by error in the context of visualisation ?
  - At least to make it appear nice
    - Should avoid the path to diverge!
    - It is numerically and visually bad!

Numerical Integration : Euler's Method



$$\vec{x}(t) = \int_t \vec{V}\, dt$$

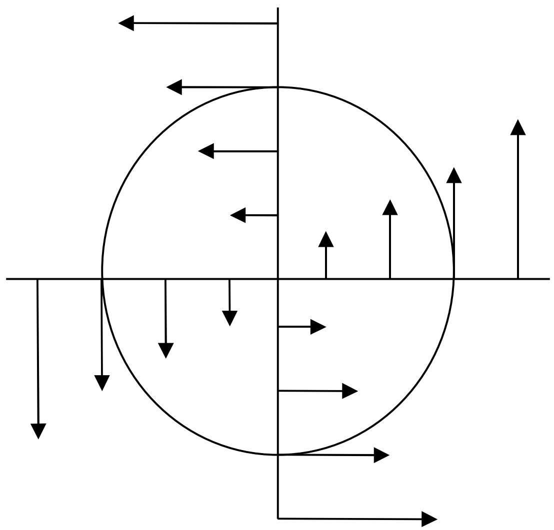
Euler's method :

$$\vec{x}_{i+1} = \vec{x}_i + \vec{V}_i \Delta t$$

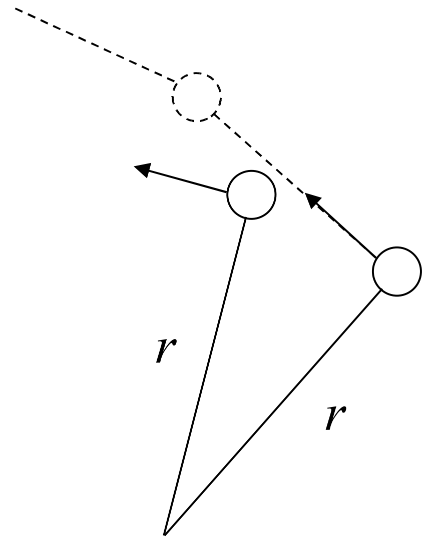
New position $\vec{x}_{i+1} =$ *old position,* $\vec{x}_i$ *plus instantaneous velocity times incremental time step*

Numerical Error is $O(\Delta t^2)$

Problem with Euler's method



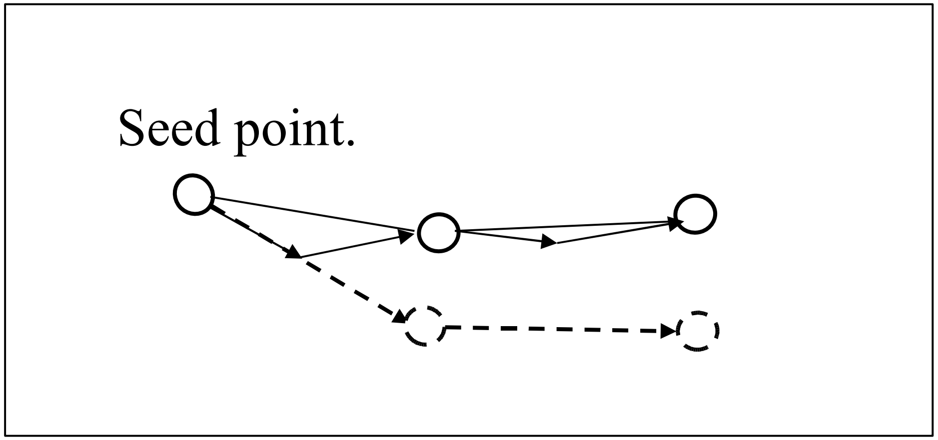
Rotational flow field.



With Euler’s method, integrated flow occurs in a spiral.

- With a rotational flow field – Euler's method wrongly diverges due to error

Runge-Kutta method, 2nd Order



Euler's method : $\vec{x}_{i+1} = \vec{x}_i + \vec{V}_i \Delta t$

Runge-Kutta method :

$$\vec{x}_{i+1} = \vec{x}_i + \frac{\Delta t}{2}\left(\vec{V}_i + \vec{V}_{i+1}\right)$$

$\vec{V}_{i+1}$ *is calculated using Euler's method .*

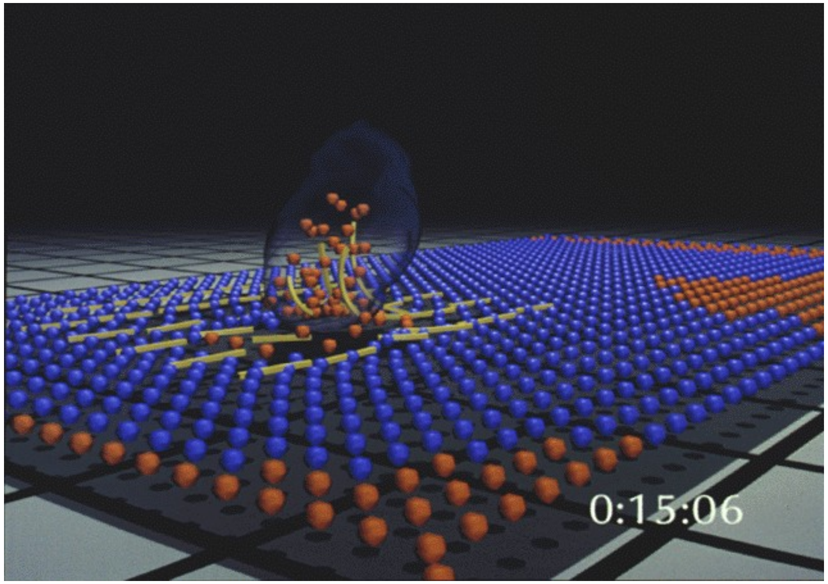
Error is $O(\Delta t^3)$

(assuming $0 < \Delta t < 1$)

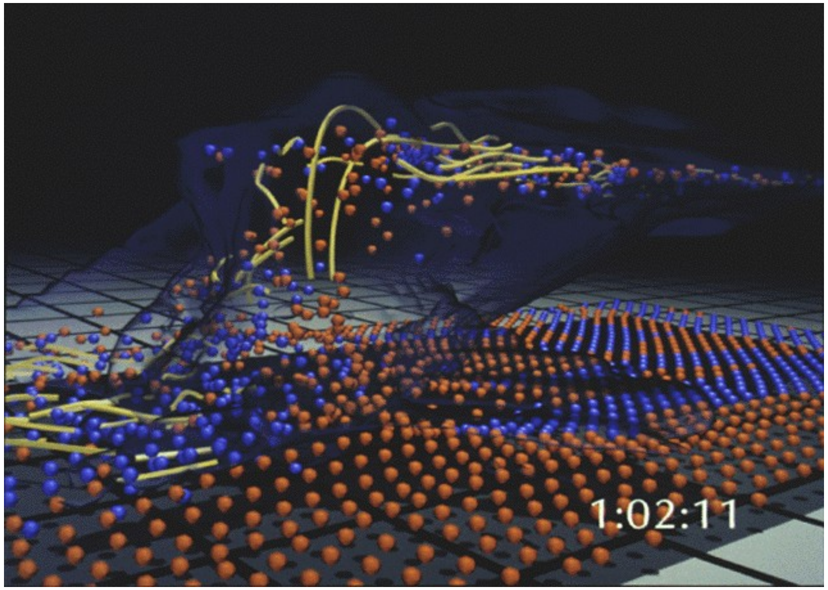
2nd Order Runge-Kutta

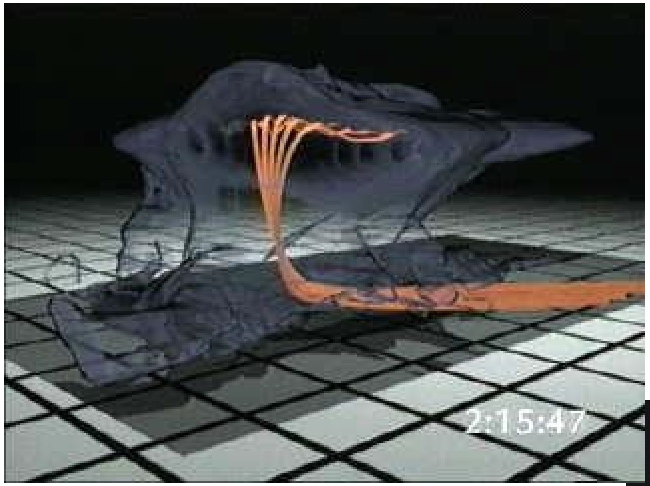
- Improves accuracy, but more expensive
  - additional function evaluation
  - *N.B.* $0 < \Delta t < 1$
- Larger time-step for same error
- 4th Order Runge-Kutta also popular for integration
- Best method depends on data and interpretation

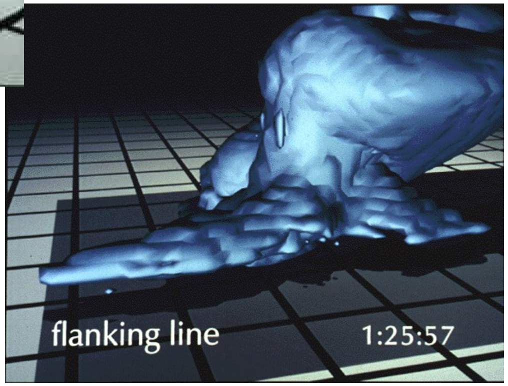
Example : thunderstorm simulation



- Massless particles are introduced in a regular grid
  - Orange indicates ascending
  - Blue indicates descending

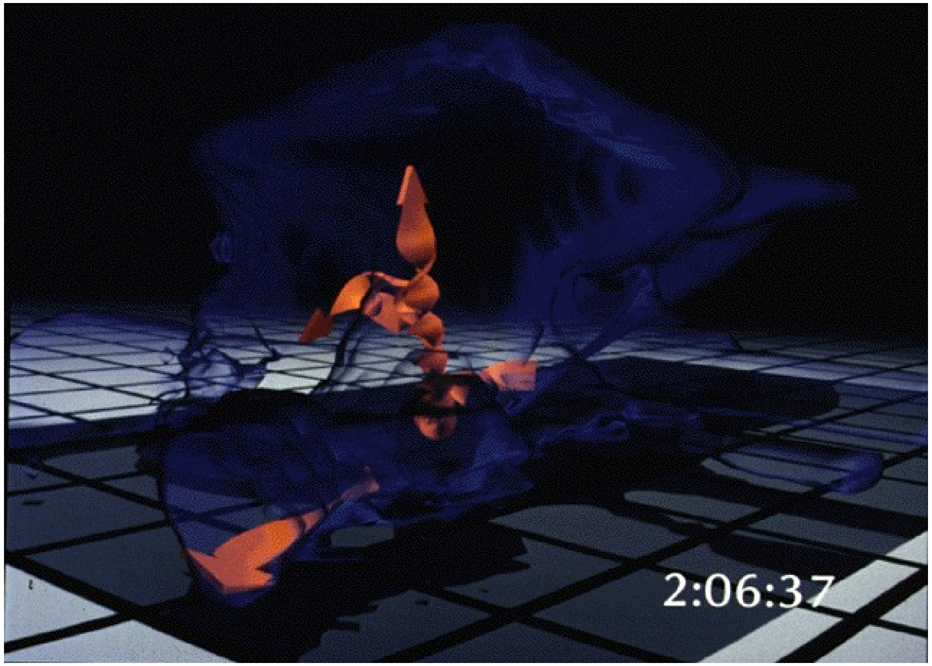




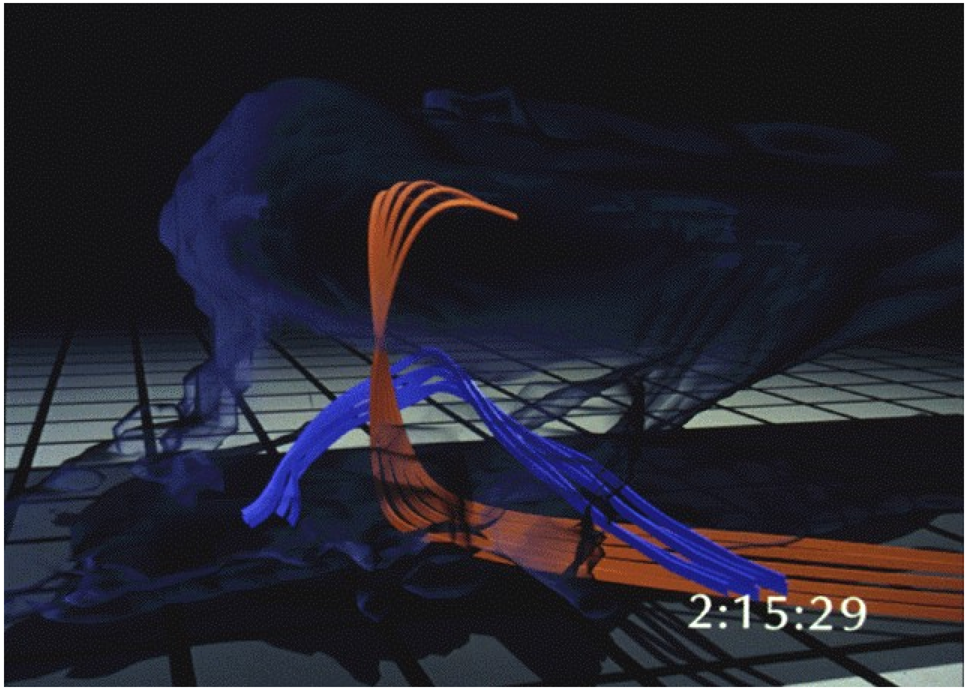


http://www.mediaport.net/CP/CyberScience/BDD/fich_050.en.html

Example : thunderstorm simulation

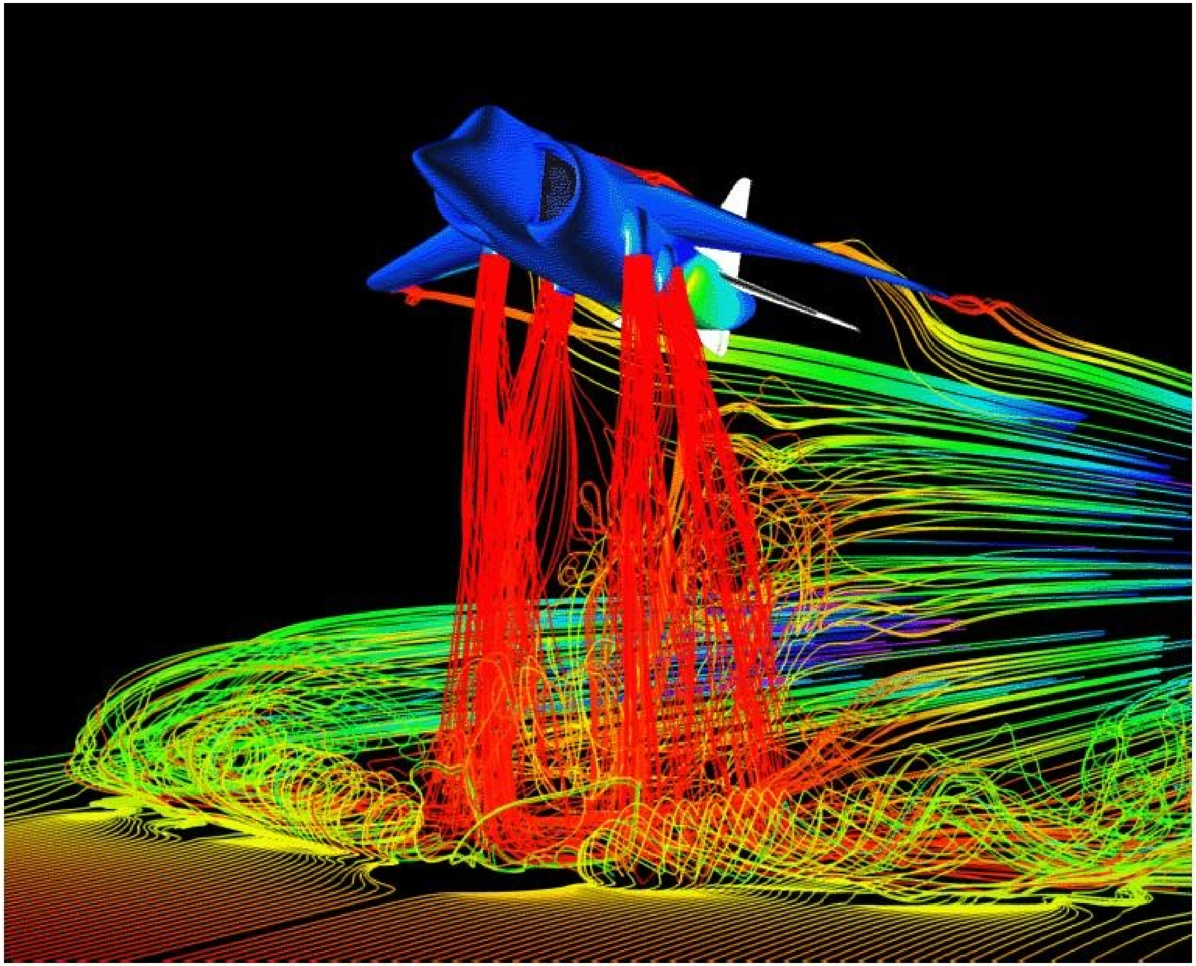


- **Streamers indicate air movement**, colours are used as before.
  - Rotation of air is shown by a ribbon.



http://www.mediaport.net/CP/CyberScience/BDD/fich_050.en.html

Initialisation of Streamlines



NASA Ames, FAST system

- Streamlines usually initialised along a curve, or ***rake***
- Often initialised at a source (e.g. engine thrust)
- Results can vary depending on placement of rake

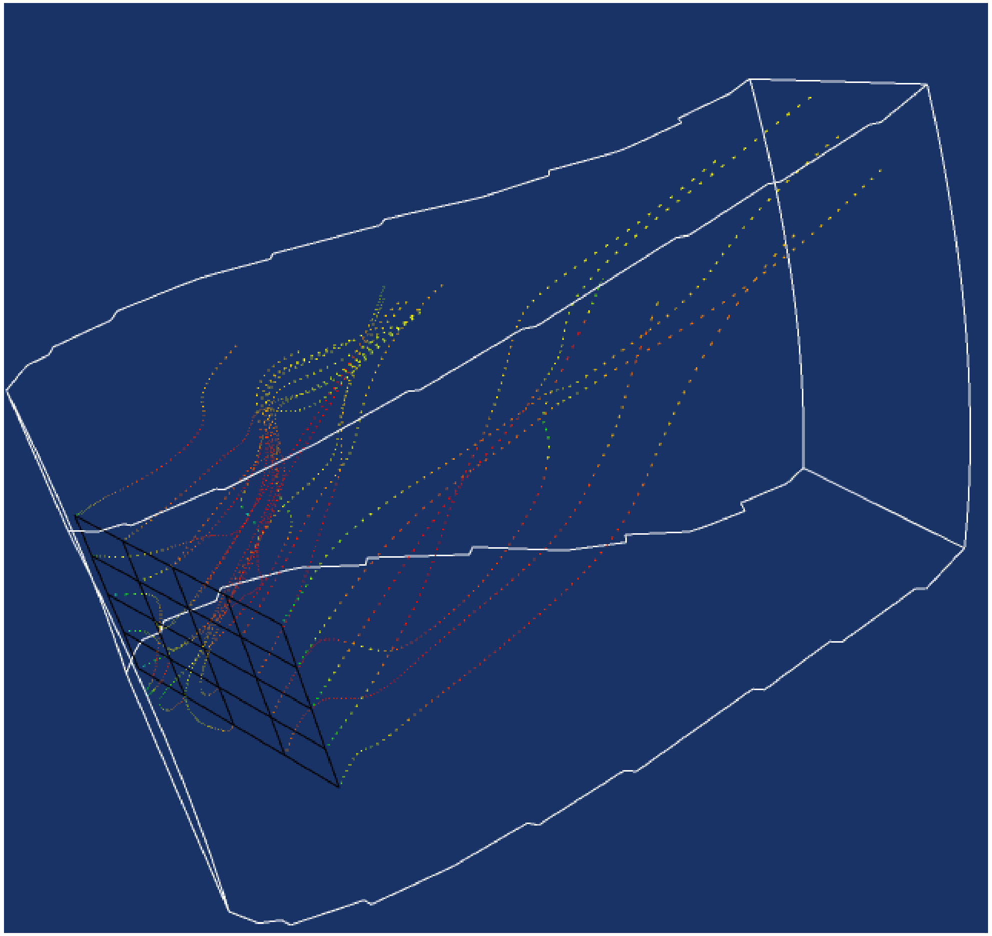
Lines & Points

- **visualise particle trace with points**
  - show all points simultaneously (like time-lapse photograph)
  - or animate the points over time (for trajectory trace)
  - can connect the points with lines
- **colour mapping** to show speed, or use **dashes**, with length proportional to speed
- Use **ribbons or tubes** to show other properties (next lecture)

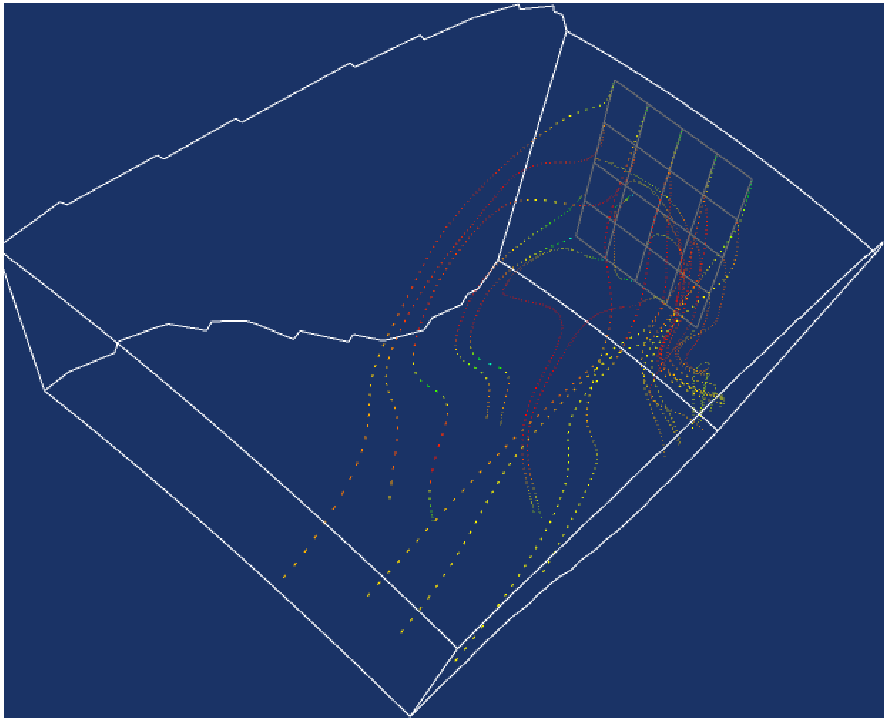
VTK : Streamlines

- `vtkStreamer`
  - base class
  - performs numerical integration to generate particle paths
- `vtkStreamLine`
  - derived class
  - produces connect stream lines from integration results
- `vtkDashedStreamLine / vtkStreamPoints`
  - derived classes

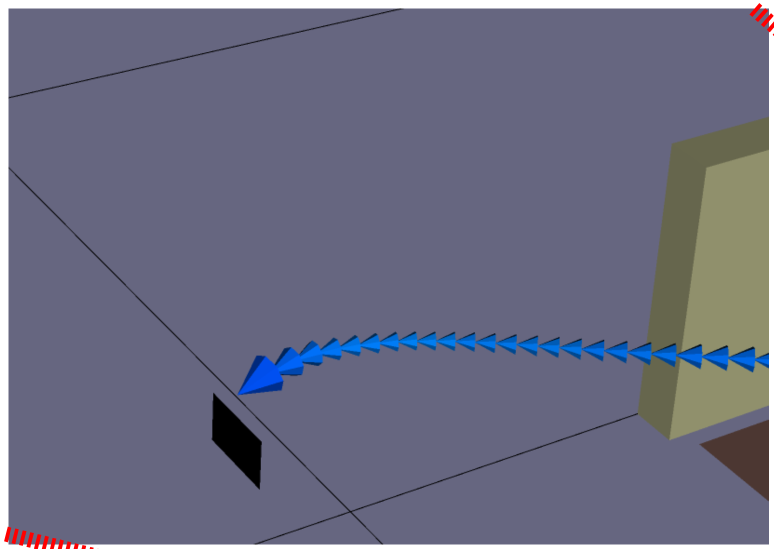
VTK : Stream Points



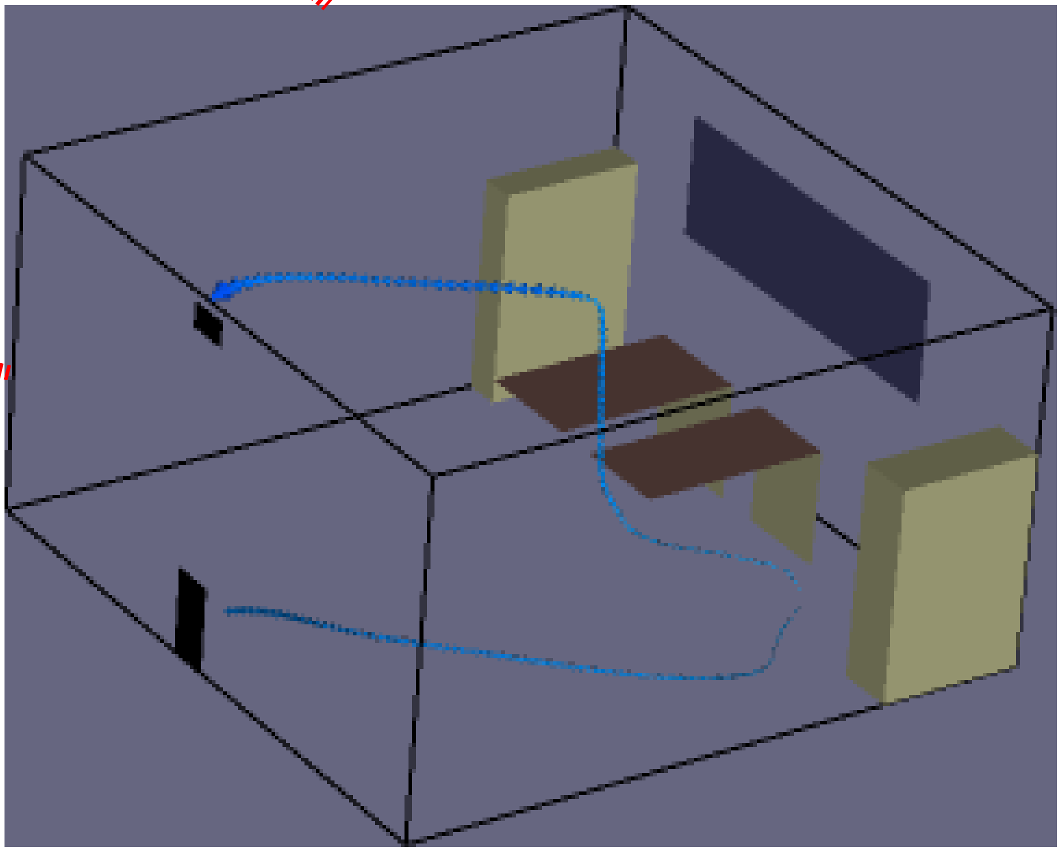
Stream Points (points along stream line at given separation)
N.B. rake = 2D grid



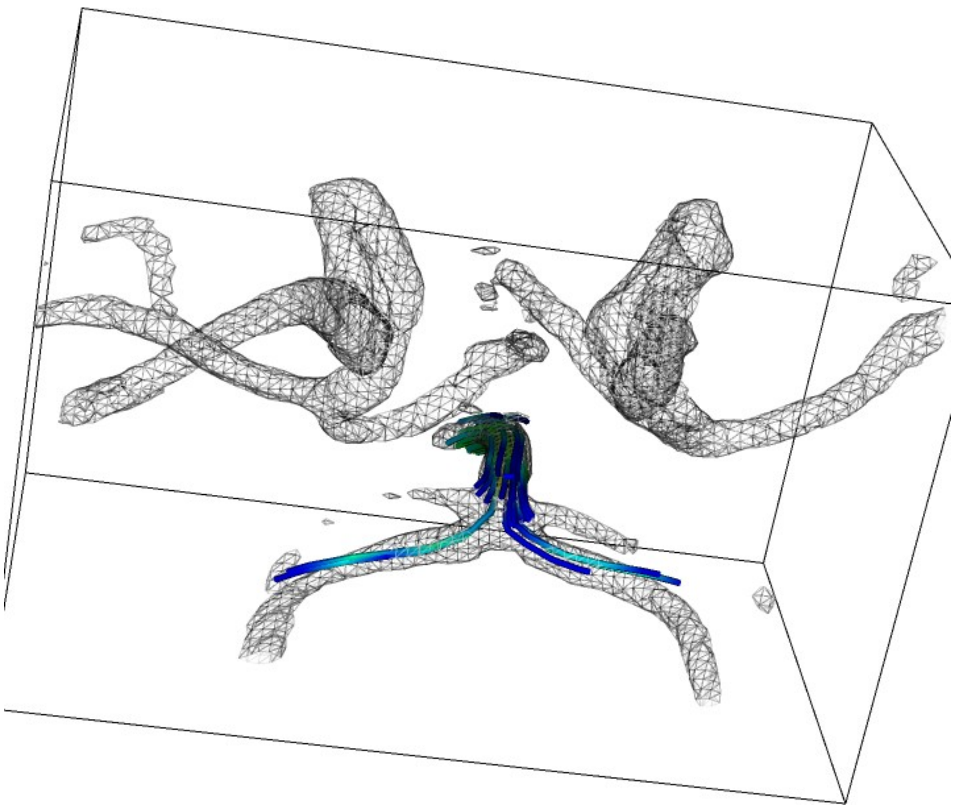
VTK : Dashed Stream Line



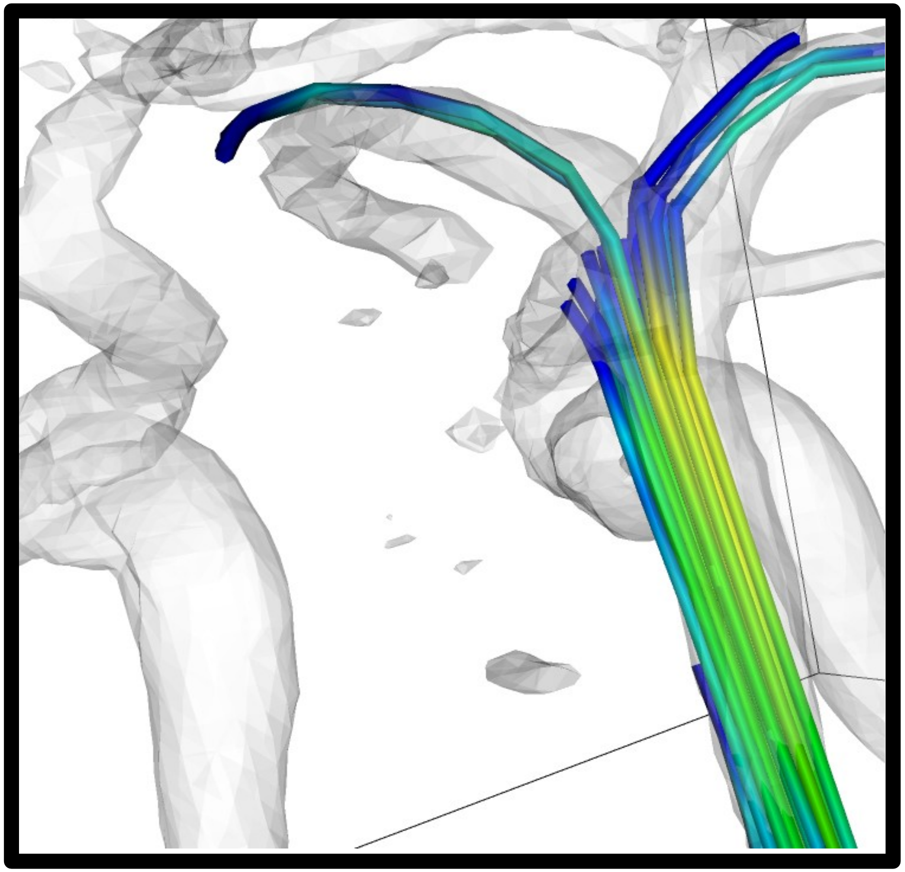
- Uses `vtkGylph3D` to specify shape to use for dash



Example : carotid artery



- Visualisation using VTK streamlines
  - solves clarity issues of glyphs



`streamV.tcl`

Vector Visualisation Summary

- **Vector visualisation:**
  - **local** view / **global** view
  - **steady** / **unsteady** flow
- Local Vector Visualisation:
  - **lines**, **hedgehogs** & **glyphs**
  - **colour mapping, warping & animation**
- **Global View of Vector Fields**
  - visualising **transport**
  - requires **numerical integration**
    - Euler's method
    - Runge-Kutta
  - **stream** {lines | points | glyphs }