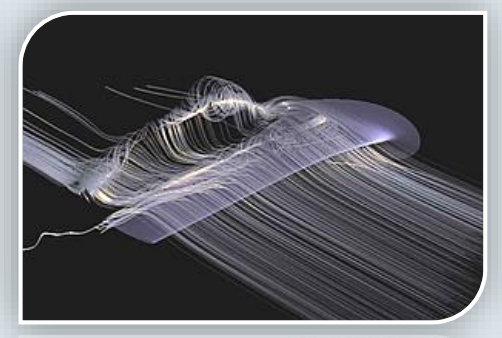
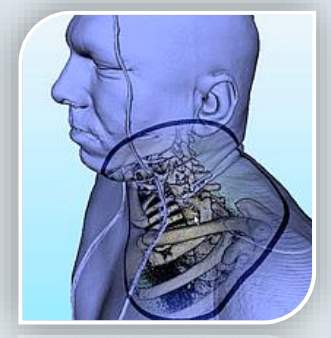
SIEMENS
Ingenuity for life


Visual Data Analytics
Flow Visualization I

Dr. Johannes Kehrer

Disclaimer

SIEMENS
Ingenuity for life

These lecture slides and materials are for personal use by students and are intended for educational purposes only. Distribution or reproduction of the material is prohibited.

The views and opinions expressed by any individual during the course of these lecturers are their own and do not necessarily represent the views, opinions, or positions of Siemens or the Technical University of Munich.

Although reasonable efforts have been made to ensure the information in these lecture slides is current and accurate, we do not assume any responsibility for the content of the posted material.

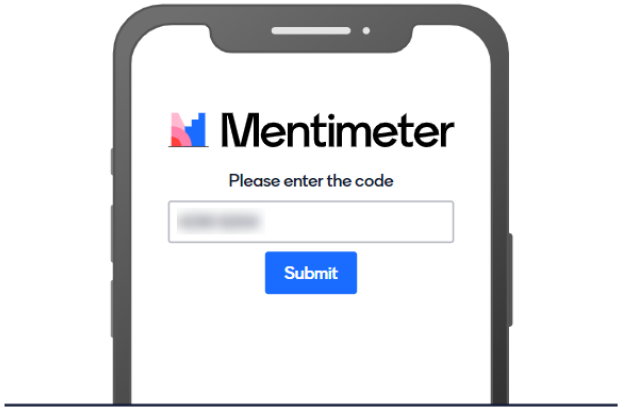
These slides also contain links to websites of third parties. As the content of these websites is not under our control, we cannot assume any liability for such external content.

Any comments/questions?

SIEMENS
Ingenuity for life

Go to

www.menti.com



Enter the code

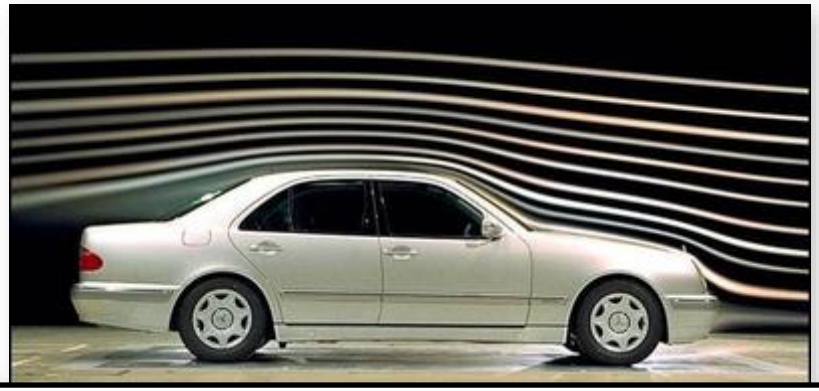
6248 9347

Or use QR code

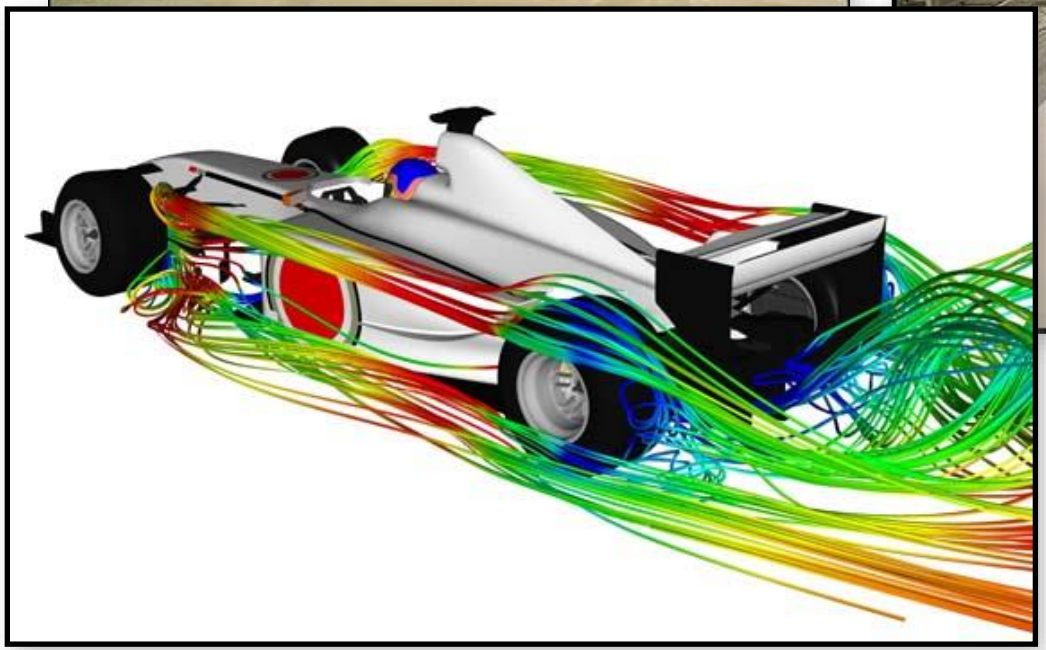
Flow visualization

SIEMENS
Ingenuity for life

Visualize vector-valued quantities like wind fields

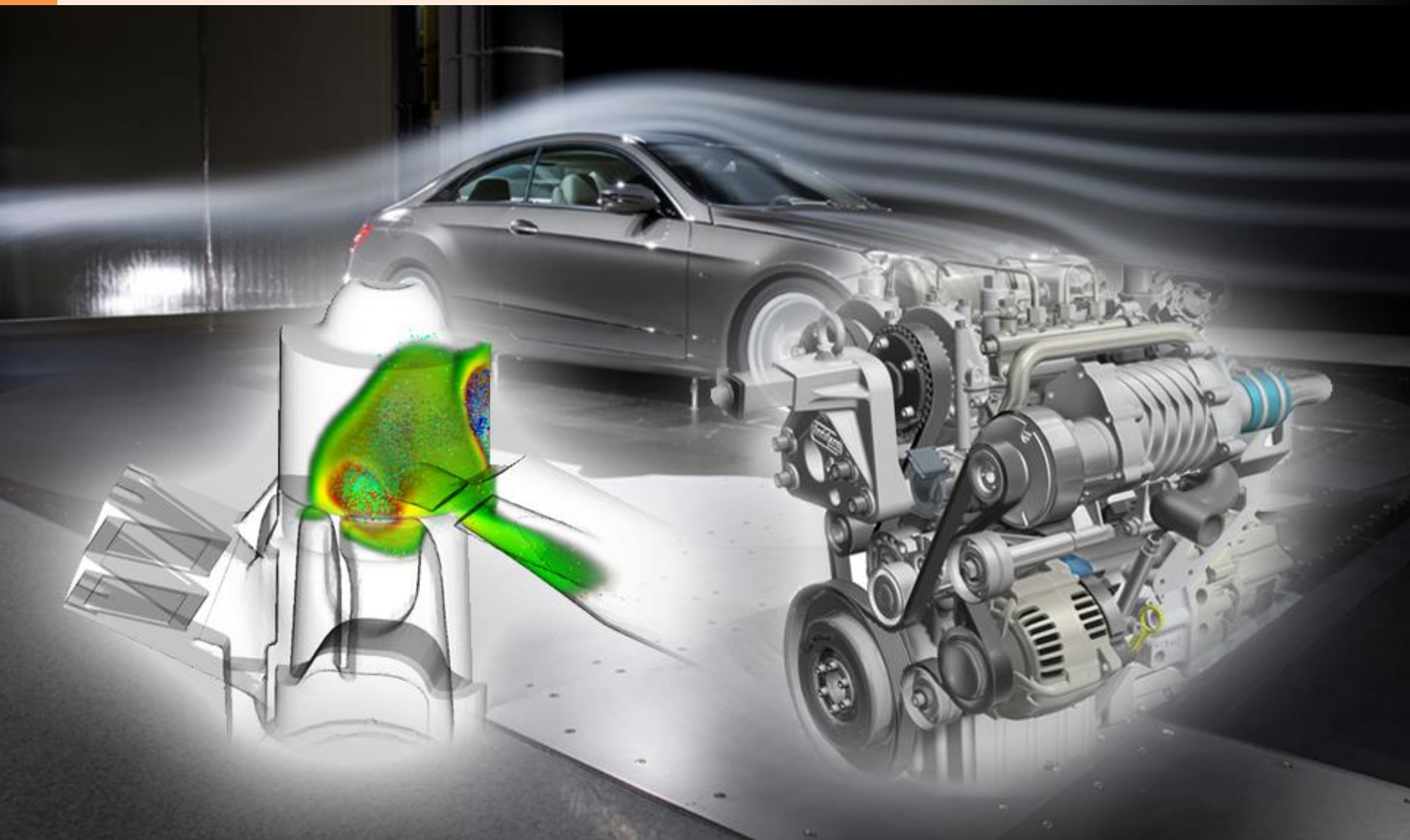


Real wind tunnel



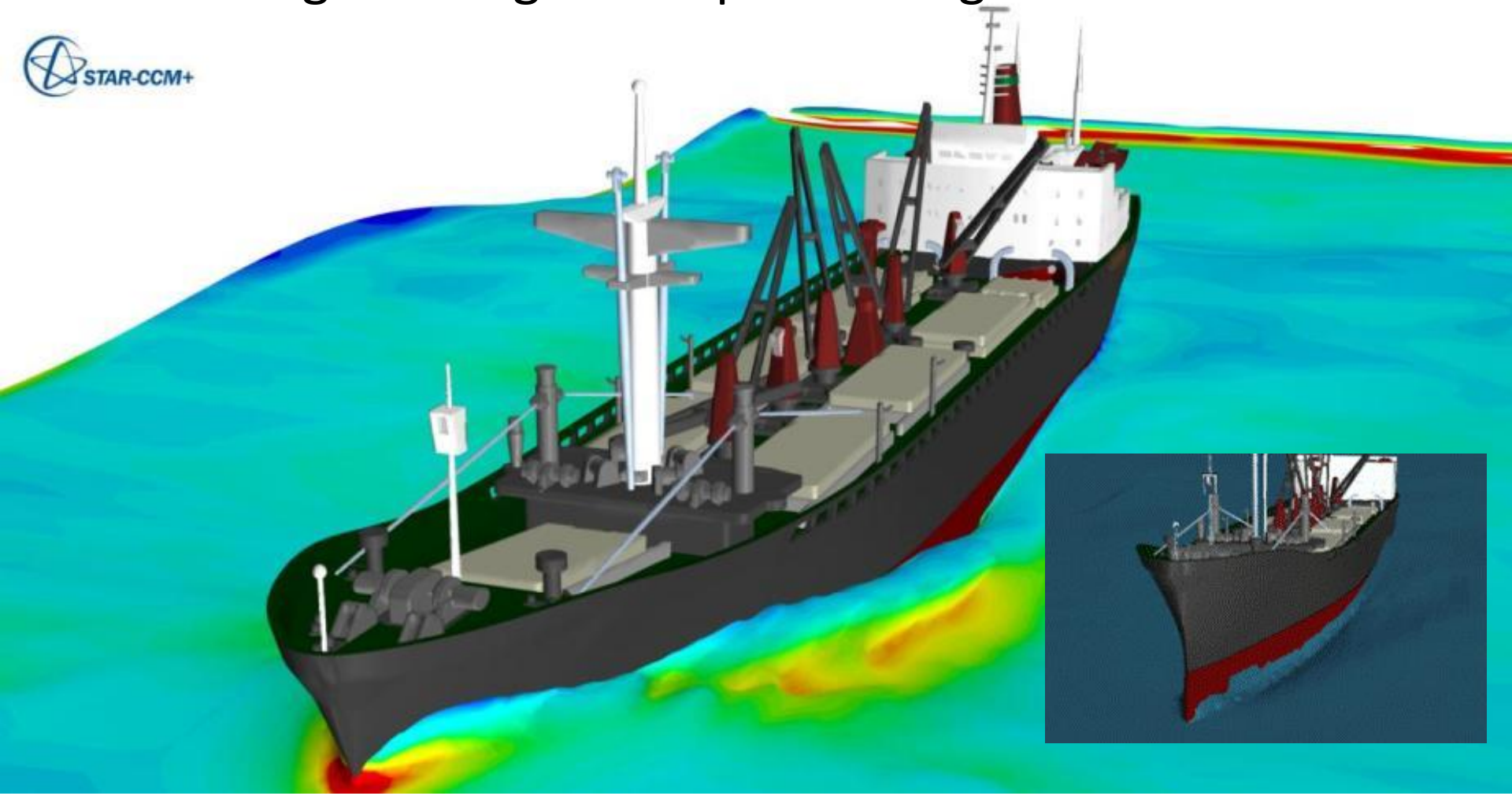
Car design: virtual wind tunnel

Automotive Engineering

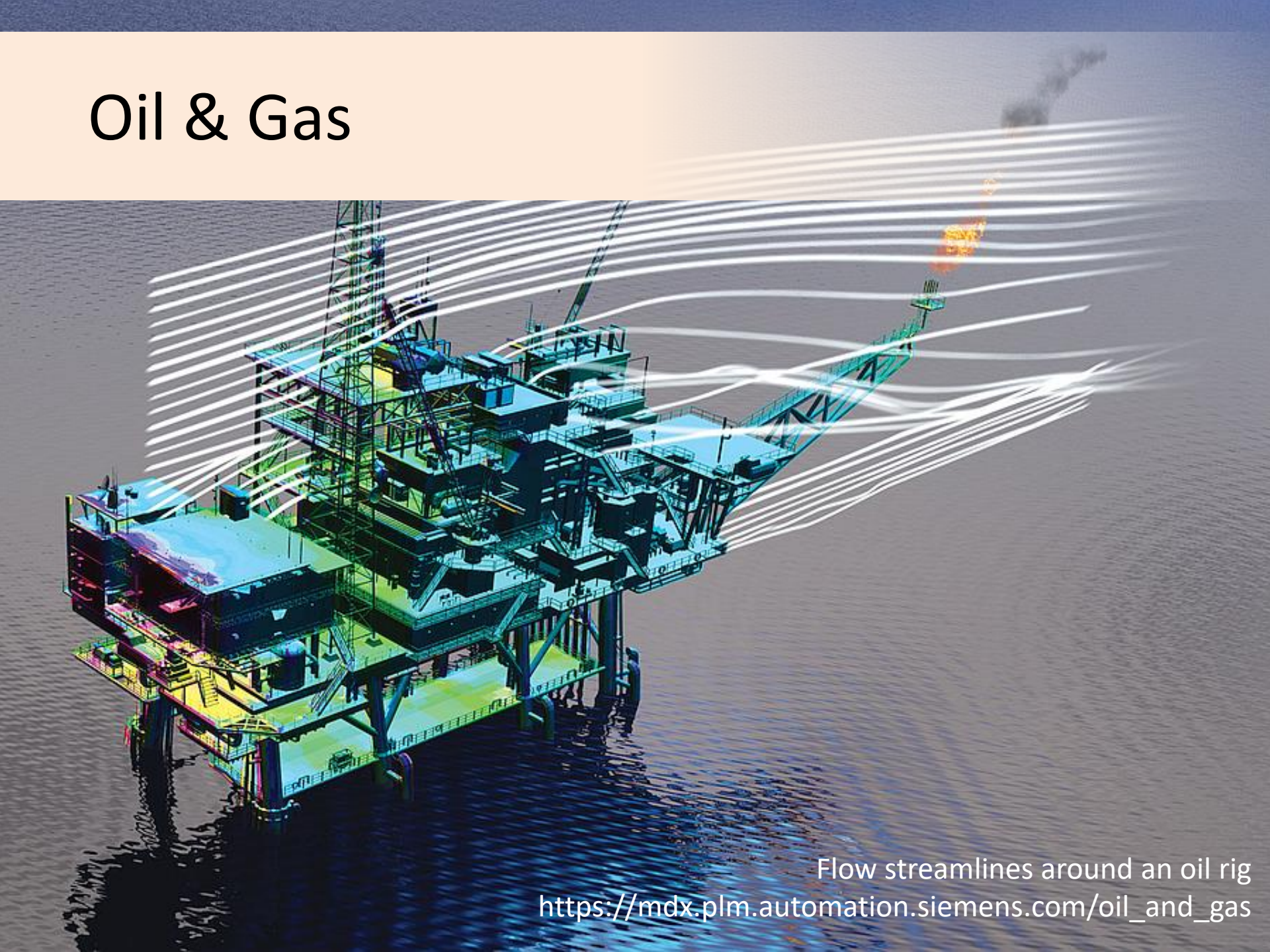


Ship design

- Simulates forces acting on everything from engine design to ship hull design

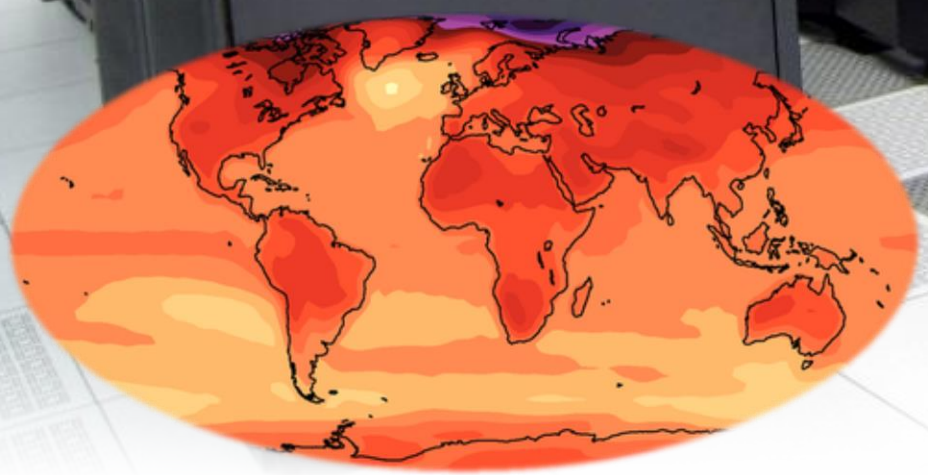


Oil & Gas



Flow streamlines around an oil rig
https://mdx.plm.automation.siemens.com/oil_and_gas

Weather & Climate Simulations



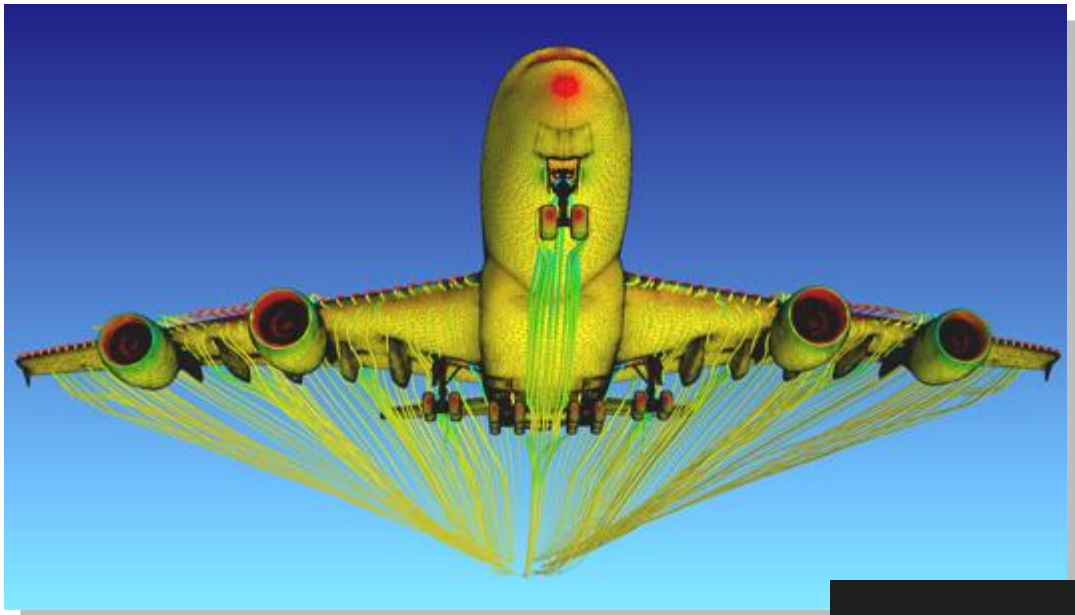
0 +1 +2 +3 +4 +5 +6 +7

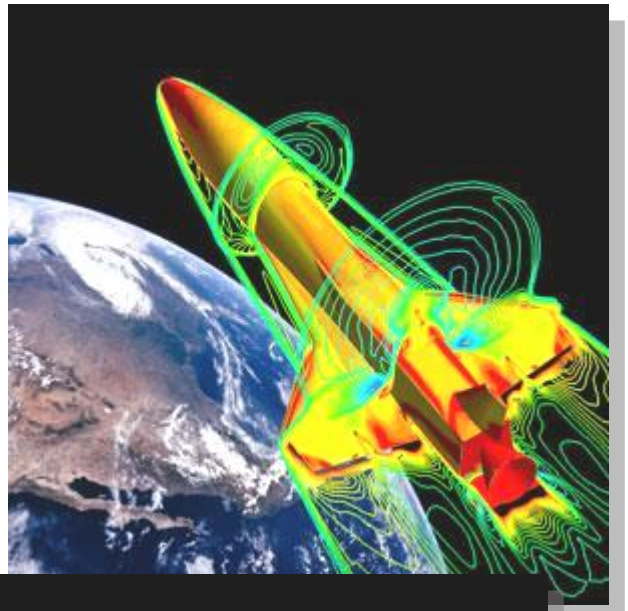
Changes in Temperature 2090-2099 (°C)

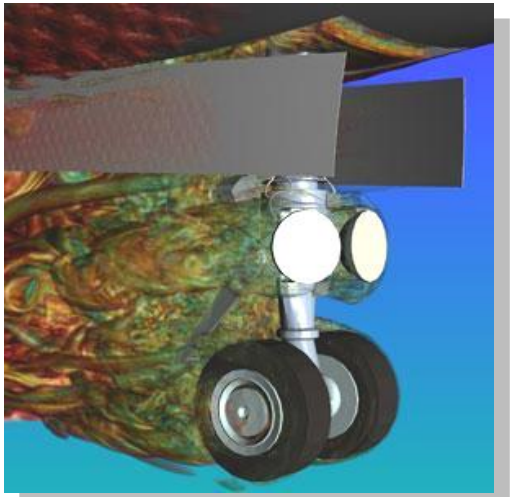
Hurricane Sandy

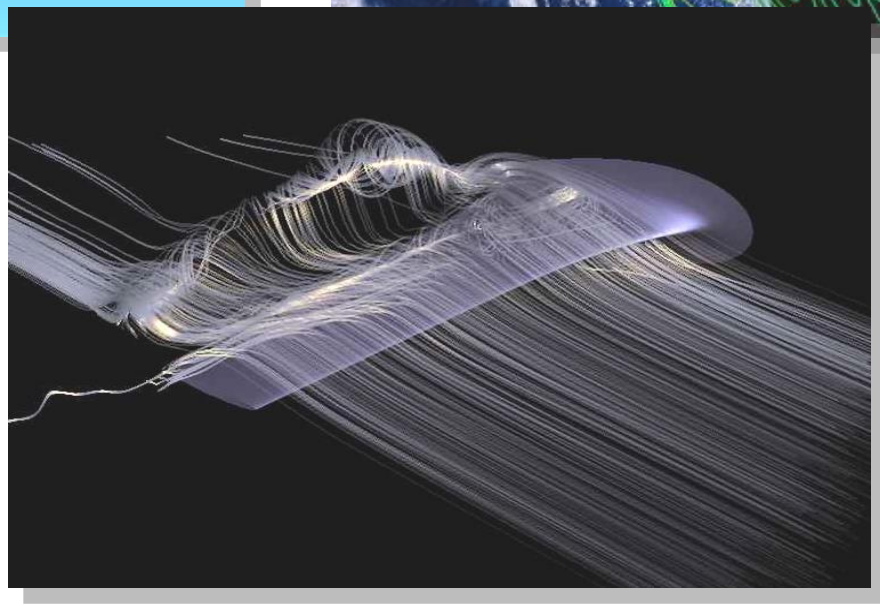
Aerospace design

SIEMENS
Ingenuity for life





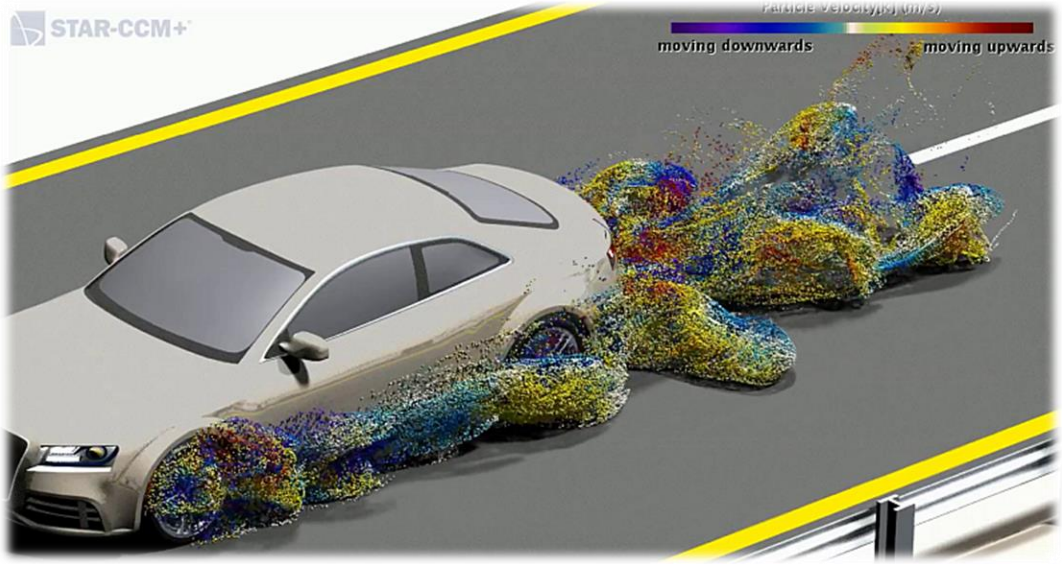


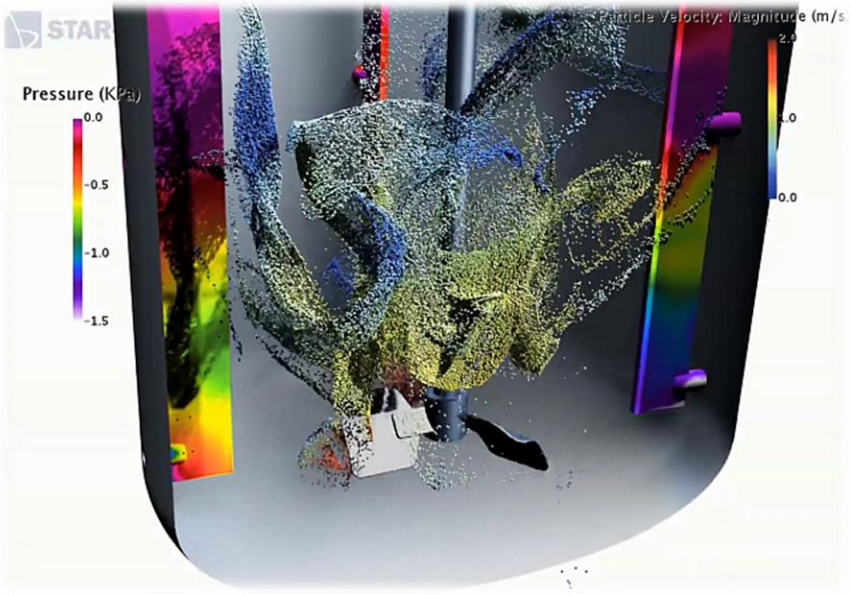


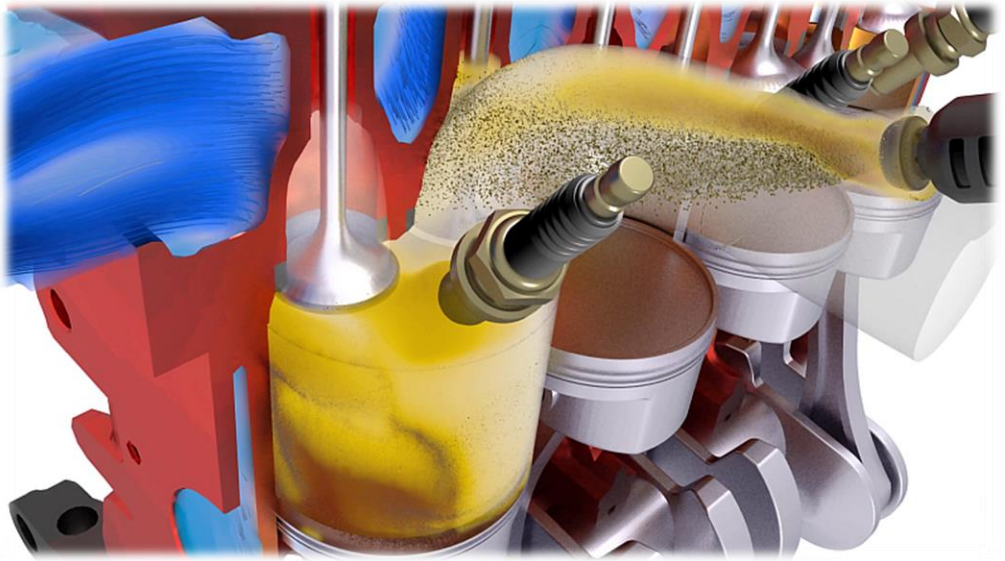
https://mdx.plm.automation.siemens.com/aerospace

Flow Visualization – Examples

SIEMENS
Ingenuity for life



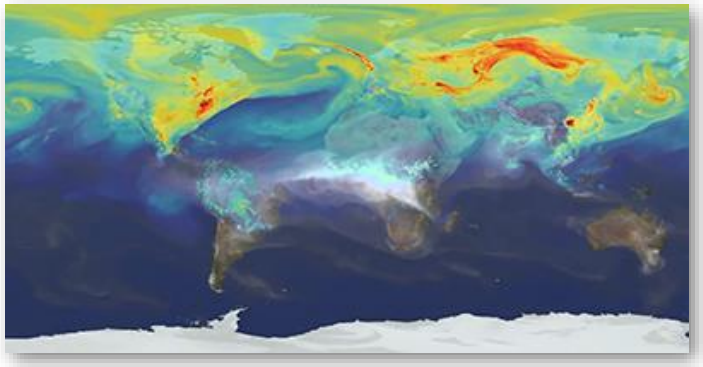


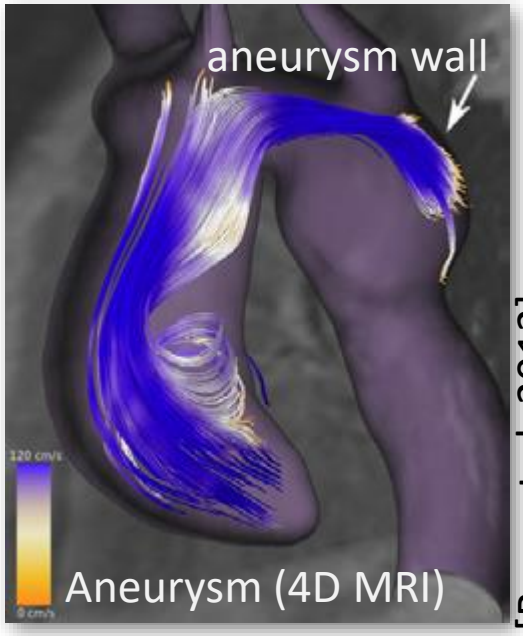


Simcenter STAR-CCM+
https://youtu.be/443kbDFPjUo

Flow visualization – data sources

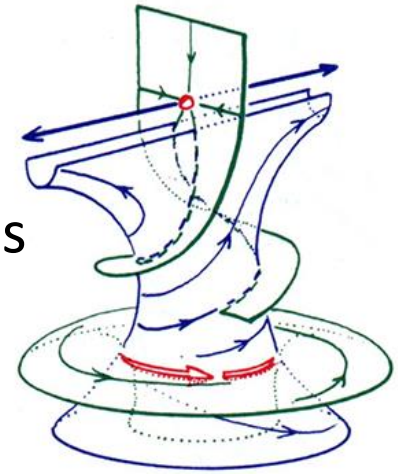
SIEMENS
Ingenuity for life

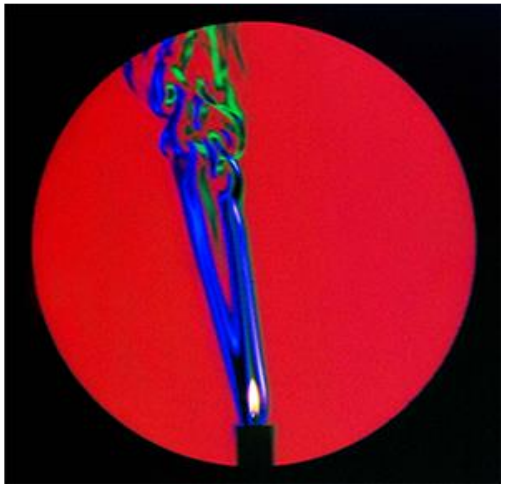




[Born et al. 2012]

- Flow simulation
  - Design of ships, cars, airplanes, …
  - Weather simulations (e.g., atmospheric flow)
  - Medical blood flows
- Measurements
  - Wind tunnel
  - Schlieren imaging
- Modeling
  - Differential equations systems (dynamical systems)





Color schlieren of burning candle

Vector field visualization

SIEMENS
Ingenuity for life

- Main application of flow visualization
  - Motion of fluids (gases, liquids)
  - Geometric boundary conditions
  - Velocity/flow field $\boldsymbol{v}(\boldsymbol{x}, t)$
  - Conservation of mass, energy, and momentum
  - Navier-Stokes equations
  - Computational fluid dynamics (CFD)

Vector field visualization

SIEMENS
Ingenuity for life

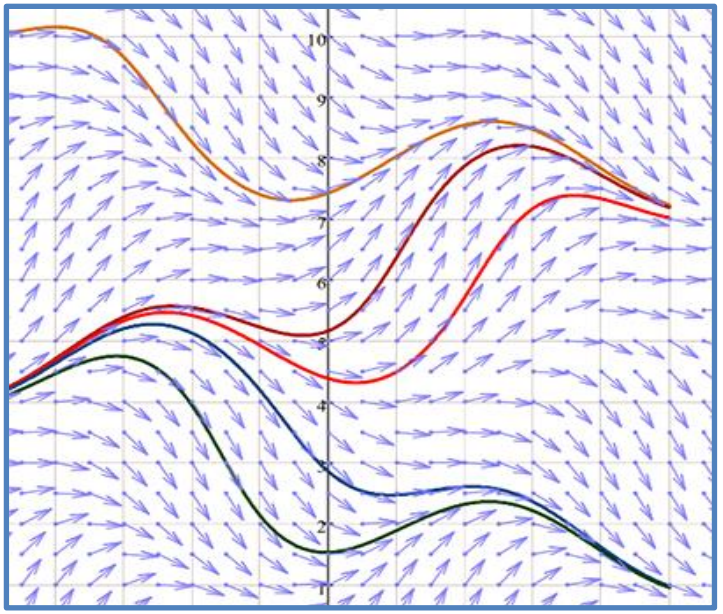
- Flow visualization – classification
  - Dimension (2D or 3D)
  - Time-dependency: steady vs. time-varying flows
  - Direct vs. indirect flow visualization

- In most cases, numerical methods required for flow visualization

Vector field visualization

SIEMENS
Ingenuity for life

- Vector data representing direction & magnitude
  - Given by an n-tupel $(f_1, \dots, f_n)$ with $f_k = f_k(x_1, \dots, x_n)$, $n \geq 2$ and $1 \leq k \leq n$
  - Typically 2D $(n = k = 2)$ or 3D $(n = k = 3)$

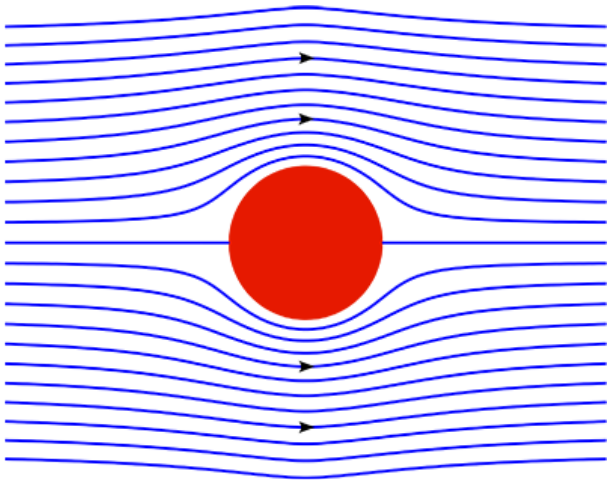


- Example
  - 2D vector field where every sample represents a 2D vector (u, v) with $u = f(x, y)$ and $v = g(x, y)$

Steady vs. time-dependent flows

SIEMENS
Ingenuity for life

- Steady (time-independent) flow
  - Flow static over time
  - $\boldsymbol{v}(\boldsymbol{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, e.g., laminar flows
  - Simpler interrelationships

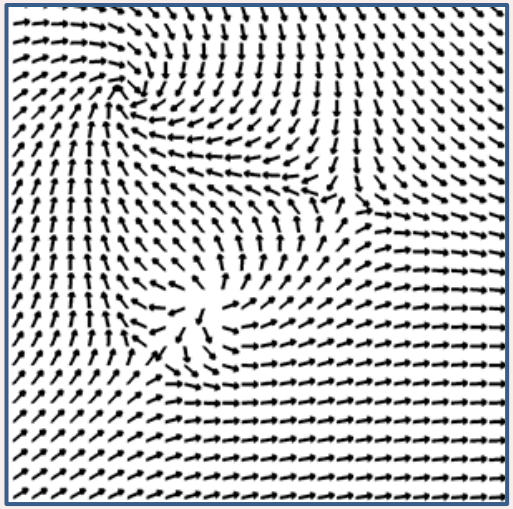


- Time-varying (unsteady) flow
  - Flow changes over time
  - $\boldsymbol{v}(\boldsymbol{x}, t) : \mathbb{R}^n \times \mathbb{R}^1 \rightarrow \mathbb{R}^n$, e.g., turbulent flows
  - More complex interrelationships

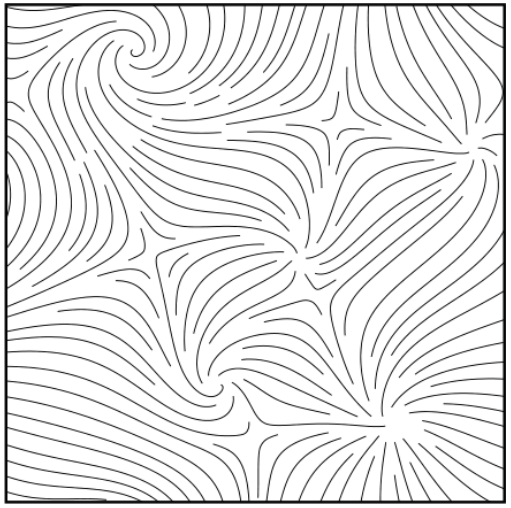
Turbulence from an airplane wing

Flow visualization – Approaches

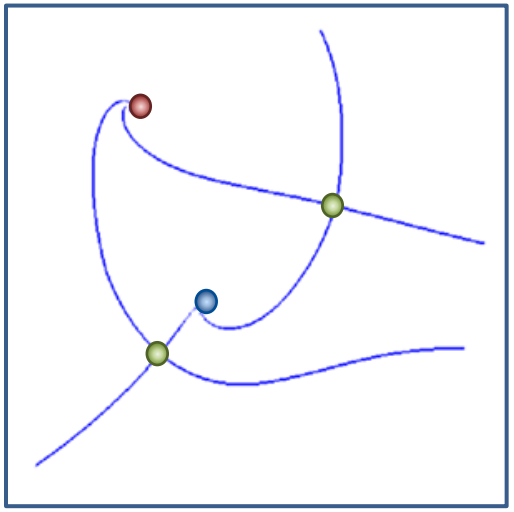
SIEMENS
Ingenuity for life



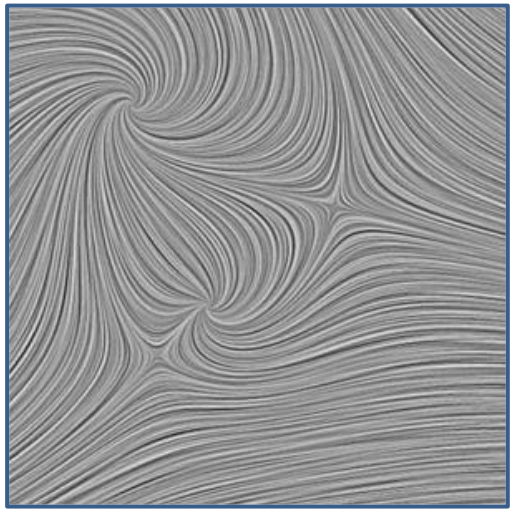
Direct flow visualization
(arrows, color coding, …)



Geometric flow visualization
(stream lines/surfaces, …)



Sparse (feature-based) vis.

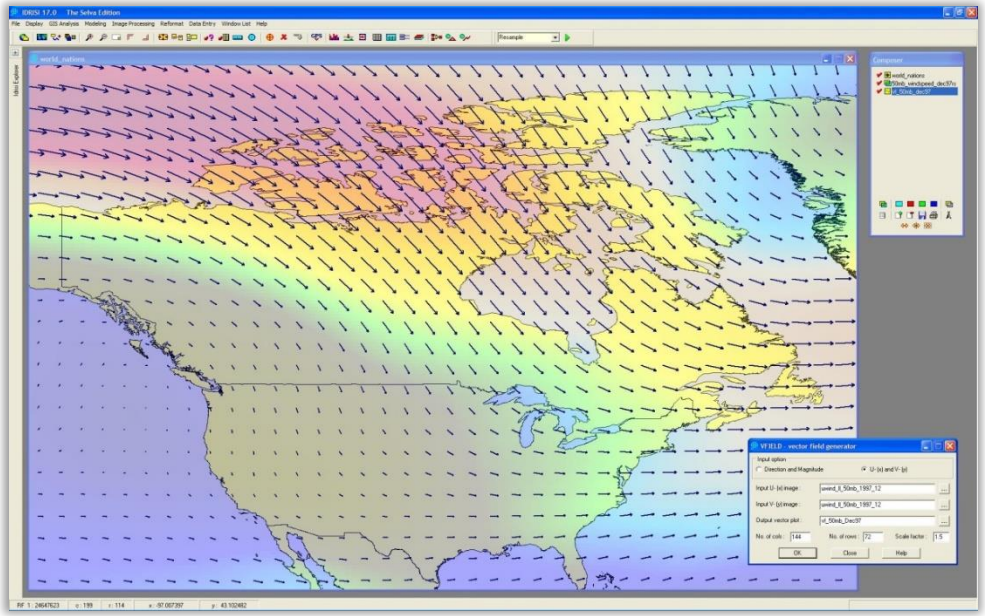


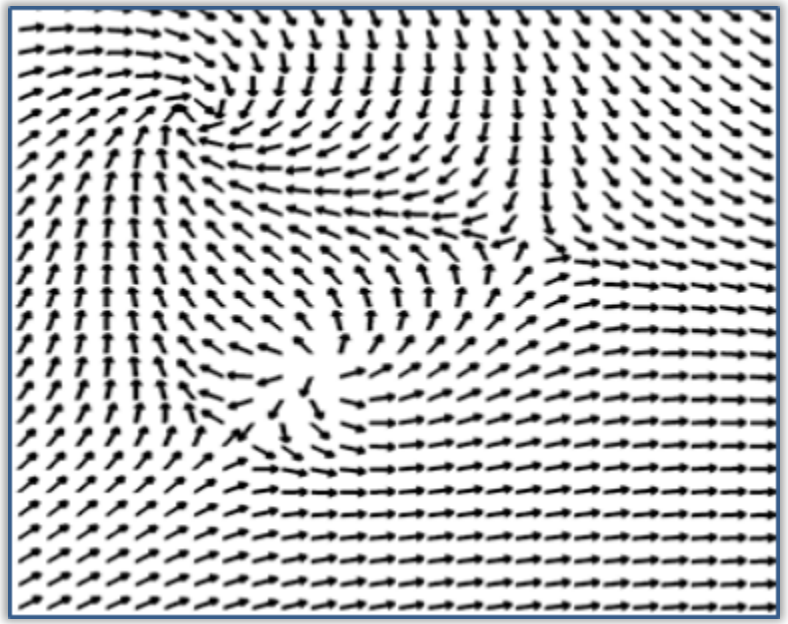
Dense (texture-based) vis.

Flow visualization – Approaches

SIEMENS
Ingenuity for life

- Direct flow visualization
  - Color coding, arrow plots, glyphs
  - Gives overview on current flow state
  - Visualization of vectors





Vector and scalar functions

SIEMENS
Ingenuity for life

- Scalar function $\rho(\boldsymbol{x}, t)$
  - Gradient

$$\nabla\rho(\boldsymbol{x}, t) = \begin{pmatrix} \frac{\partial}{\partial x}\rho(\boldsymbol{x}, t) \\ \frac{\partial}{\partial y}\rho(\boldsymbol{x}, t) \\ \frac{\partial}{\partial z}\rho(\boldsymbol{x}, t) \end{pmatrix}$$

$\boldsymbol{x}$ … position
t … time
∇ … vector operator

Vector of partial derivatives

- Gradient takes a scalar field and gives us a vector field
- Gradient points into direction of max. change of $\rho(\boldsymbol{x}, t)$
- Gradient magnitude gives the slope along this direction

Vector and scalar functions

SIEMENS
Ingenuity for life

- Vector function $\boldsymbol{v}(\boldsymbol{x}, t)$
  - Jacobian matrix ("Gradient tensor")

$$\mathbf{J} = \nabla \boldsymbol{v}(\boldsymbol{x}, t) = \begin{pmatrix} \frac{\partial}{\partial x}\boldsymbol{v}_x & \frac{\partial}{\partial y}\boldsymbol{v}_x & \frac{\partial}{\partial z}\boldsymbol{v}_x \\ \frac{\partial}{\partial x}\boldsymbol{v}_y & \frac{\partial}{\partial y}\boldsymbol{v}_y & \frac{\partial}{\partial z}\boldsymbol{v}_y \\ \frac{\partial}{\partial x}\boldsymbol{v}_z & \frac{\partial}{\partial y}\boldsymbol{v}_z & \frac{\partial}{\partial z}\boldsymbol{v}_z \end{pmatrix}$$

Matrix of first-order partial derivatives of vector field $\boldsymbol{v}(\boldsymbol{x}, t)$

Vector and scalar functions

SIEMENS
Ingenuity for life

- Vector function $\boldsymbol{v}(\boldsymbol{x}, t)$
  - Jacobian matrix ("Gradient tensor")

$$\mathbf{J} = \nabla \boldsymbol{v}(\boldsymbol{x}, t) = \begin{pmatrix} \frac{\partial}{\partial x}\boldsymbol{v}_x & \frac{\partial}{\partial y}\boldsymbol{v}_x & \frac{\partial}{\partial z}\boldsymbol{v}_x \\ \frac{\partial}{\partial x}\boldsymbol{v}_y & \frac{\partial}{\partial y}\boldsymbol{v}_y & \frac{\partial}{\partial z}\boldsymbol{v}_y \\ \frac{\partial}{\partial x}\boldsymbol{v}_z & \frac{\partial}{\partial y}\boldsymbol{v}_z & \frac{\partial}{\partial z}\boldsymbol{v}_z \end{pmatrix}$$

Matrix of first-order partial derivatives of vector field $\boldsymbol{v}(\boldsymbol{x}, t)$

  - Divergence (dot product of vector operator)

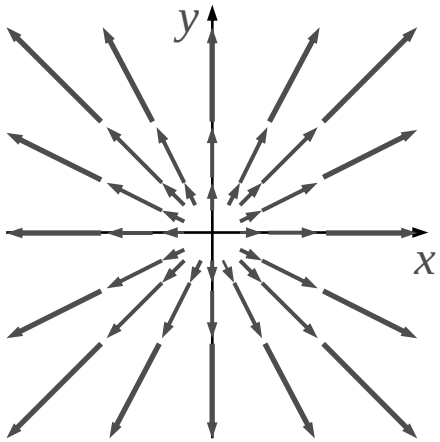
$$div\ \boldsymbol{v}(\boldsymbol{x}, t) = \nabla \cdot \boldsymbol{v}(\boldsymbol{x}, t) = \frac{\partial}{\partial x}\boldsymbol{v}_x(\boldsymbol{x}, t) + \frac{\partial}{\partial y}\boldsymbol{v}_y(\boldsymbol{x}, t) + \frac{\partial}{\partial z}\boldsymbol{v}_z(\boldsymbol{x}, t)$$

Outflow per unit volume (rate at which density exits a region)

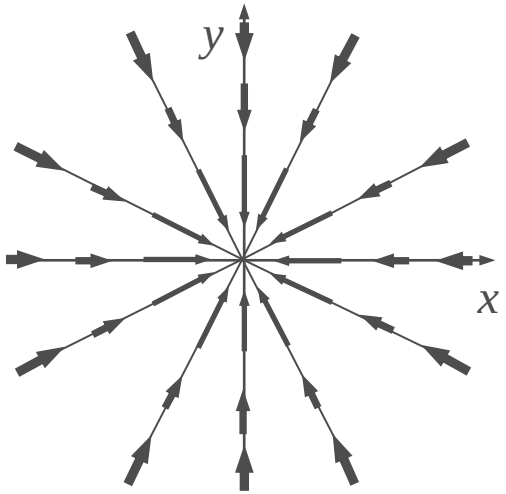
Vector and scalar functions

SIEMENS
Ingenuity for life

- Properties of divergence
  - Describes flow into/out of a region
  - $div\ \boldsymbol{v}$ is a scalar field
  - $div\ \boldsymbol{v}(\boldsymbol{x}_0) > 0$: $\boldsymbol{v}$ has a source in $\boldsymbol{x}_0$
  - $div\ \boldsymbol{v}(\boldsymbol{x}_0) < 0$: $\boldsymbol{v}$ has a sink in $\boldsymbol{x}_0$
  - $div\ \boldsymbol{v}(\boldsymbol{x}_0) = 0$: $\boldsymbol{v}$ is source-free in $\boldsymbol{x}_0$



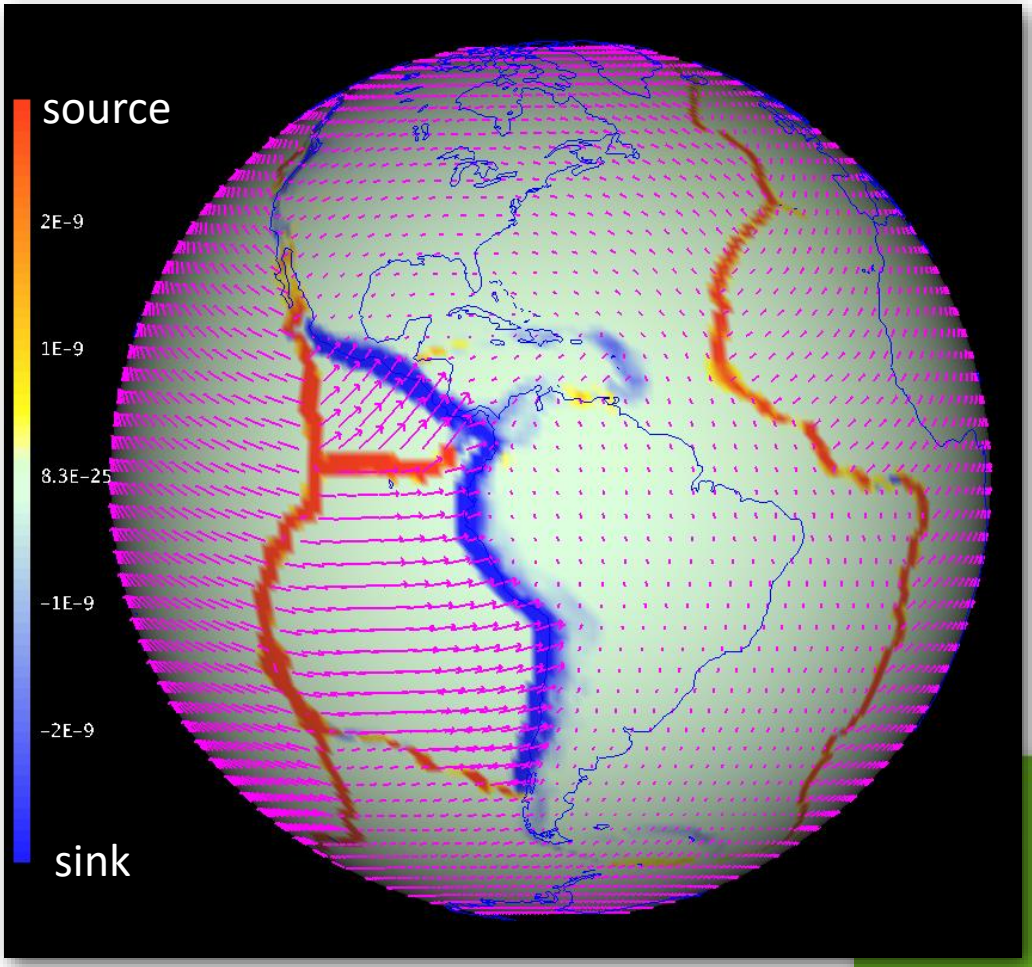
source



sink

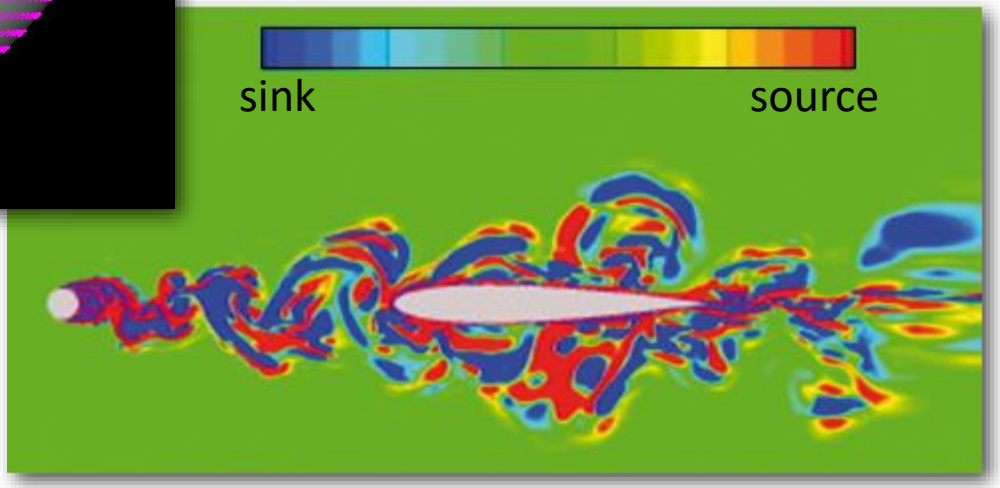
Vector and scalar functions

SIEMENS
Ingenuity for life



Tectonic plate motion

Color coding of divergence



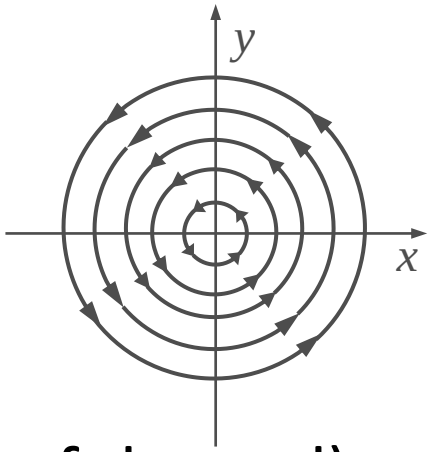
Vector and scalar functions

SIEMENS
Ingenuity for life

- Curl / vorticity

$$curl\ \boldsymbol{v}(\boldsymbol{x},t) = \nabla \times \boldsymbol{v}(\boldsymbol{x},t) = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \times \begin{pmatrix} \boldsymbol{v}_x \\ \boldsymbol{v}_y \\ \boldsymbol{v}_z \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial y}\boldsymbol{v}_z - \frac{\partial}{\partial z}\boldsymbol{v}_y \\ \frac{\partial}{\partial z}\boldsymbol{v}_x - \frac{\partial}{\partial x}\boldsymbol{v}_z \\ \frac{\partial}{\partial x}\boldsymbol{v}_y - \frac{\partial}{\partial y}\boldsymbol{v}_x \end{pmatrix}$$

  - Curl is a vector-valued quantity
  - Describes vortex characteristics in flow

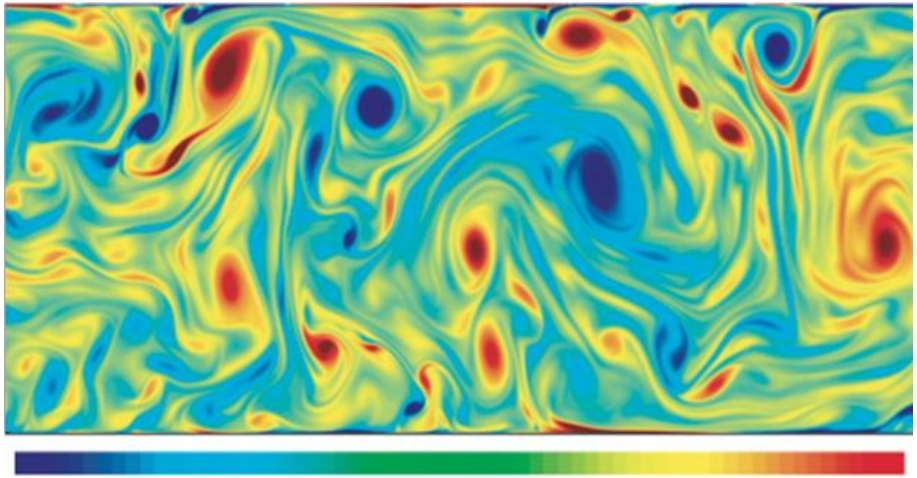


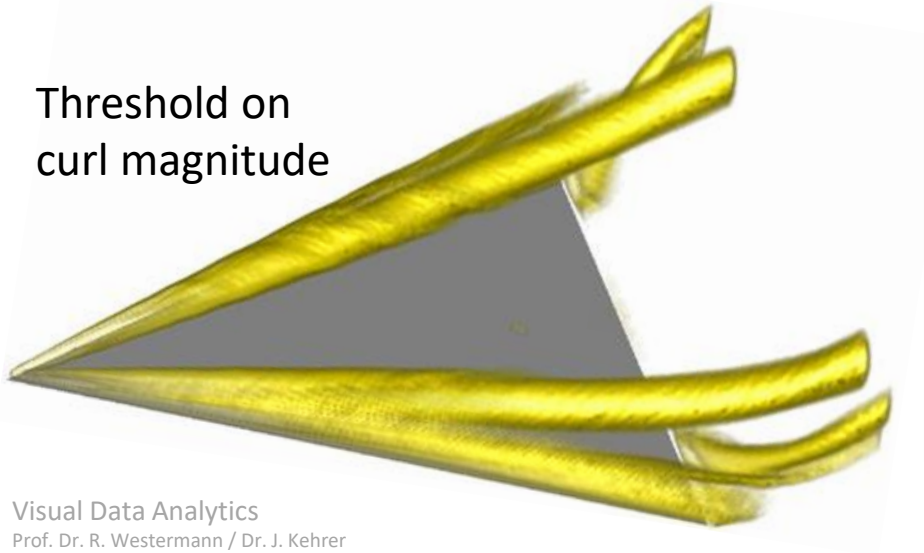
  - A measure of how fast the flow rotates (magnitude of the curl), and
  - … around which axis it rotates (direction of the curl)

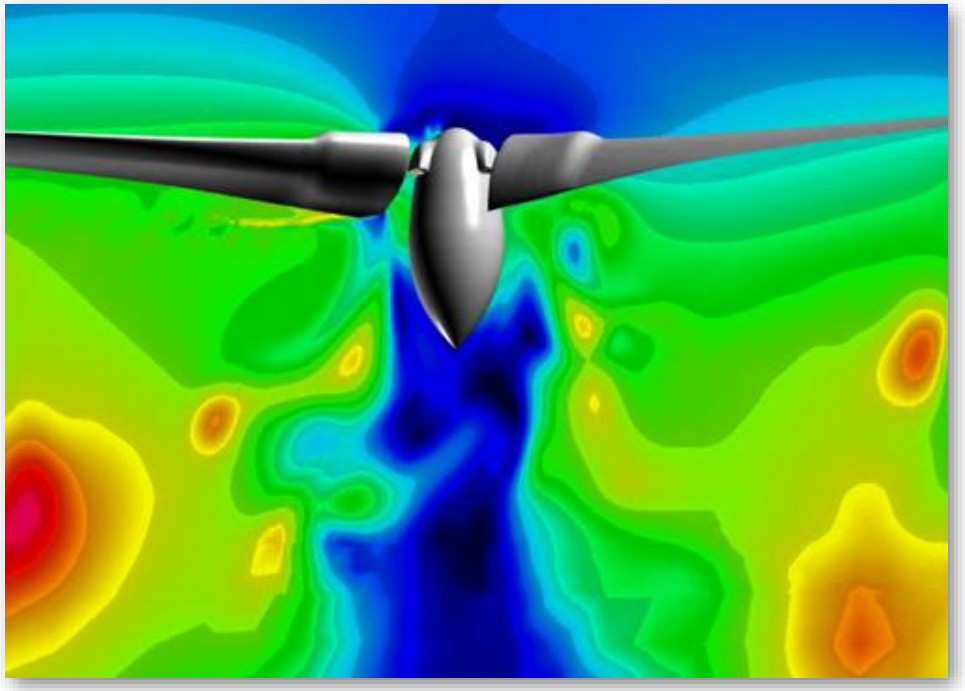
Vector and scalar functions

SIEMENS
Ingenuity for life

- Color coding of curl direction/magnitude





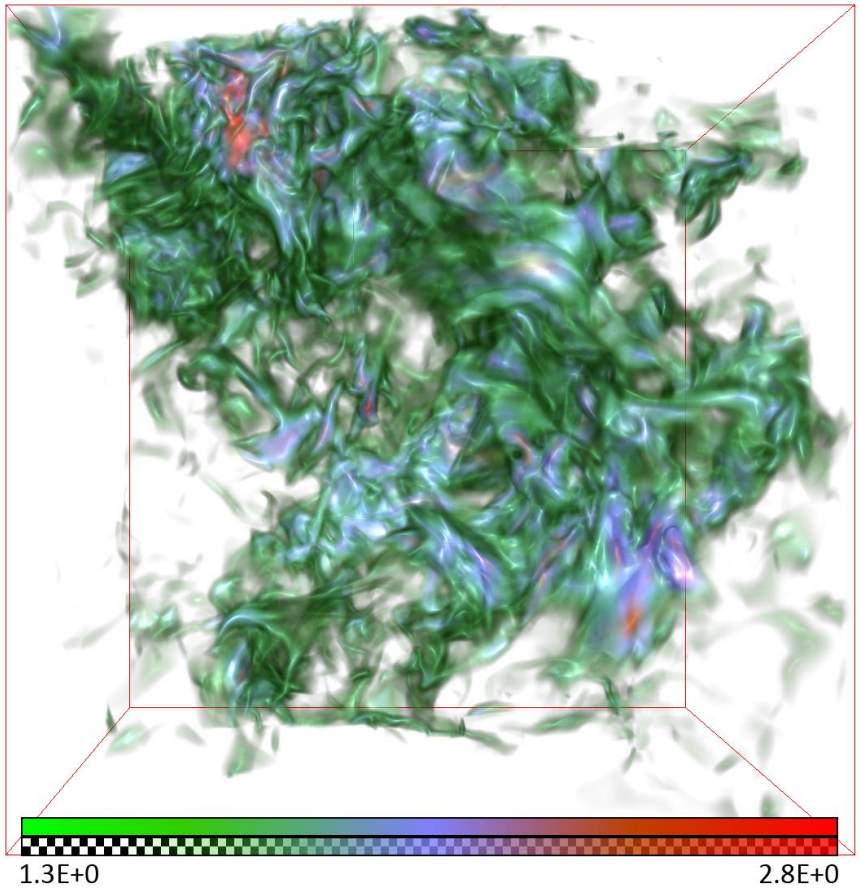


Curl magnitude

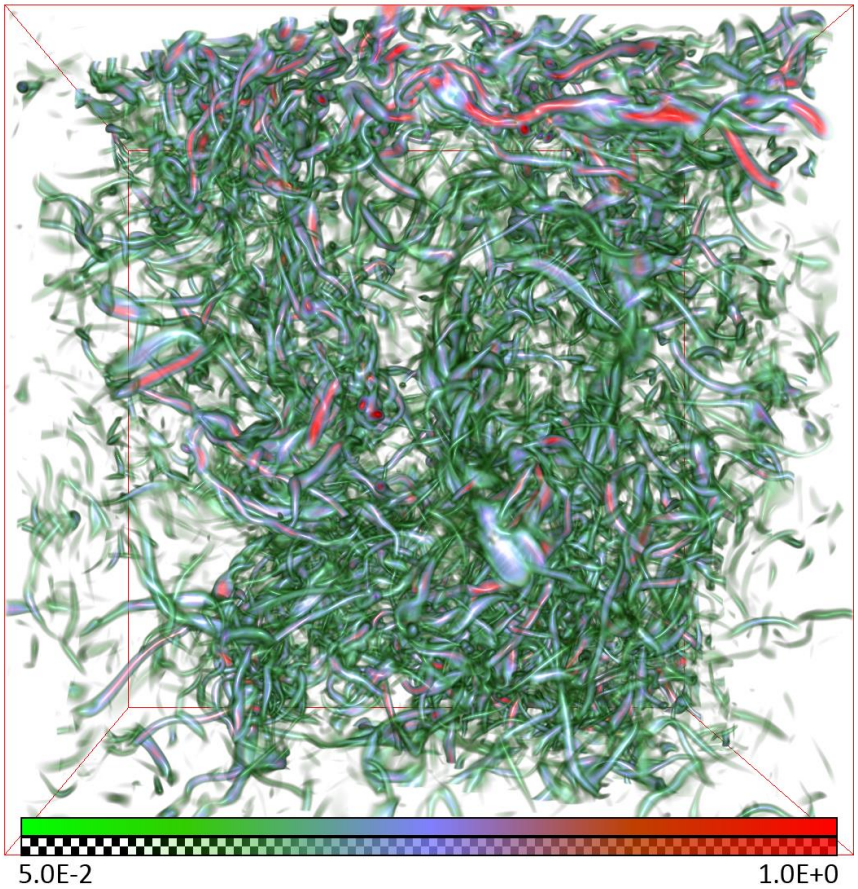
Vector and scalar functions

SIEMENS
Ingenuity for life

- Direct volume rendering (DVR) of turbulent flow



DVR of velocity magnitude

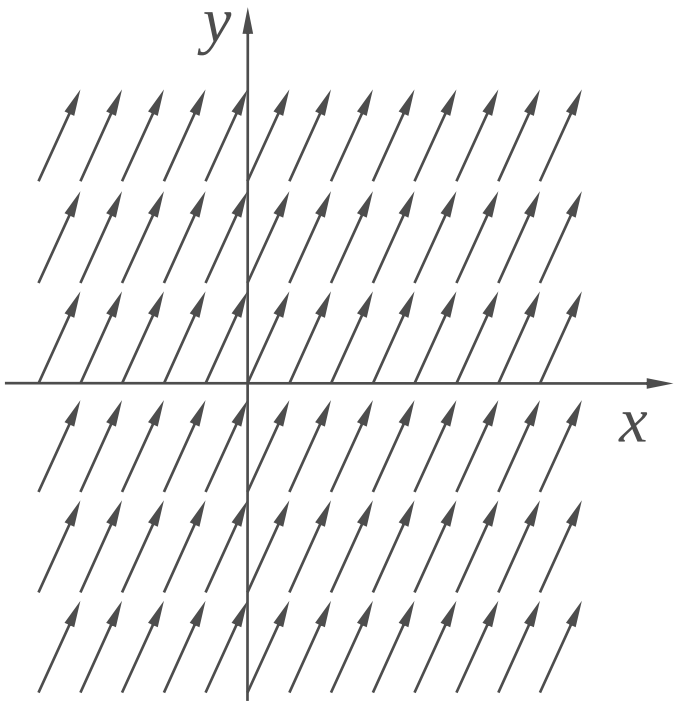


DVR of curl magnitude

Vector and scalar functions

SIEMENS
Ingenuity for life

- Example



A constant 2D vector function

$$H_1(x,y,z) = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

$$J_{H_1}(x,y,z) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

No velocity change in any direction

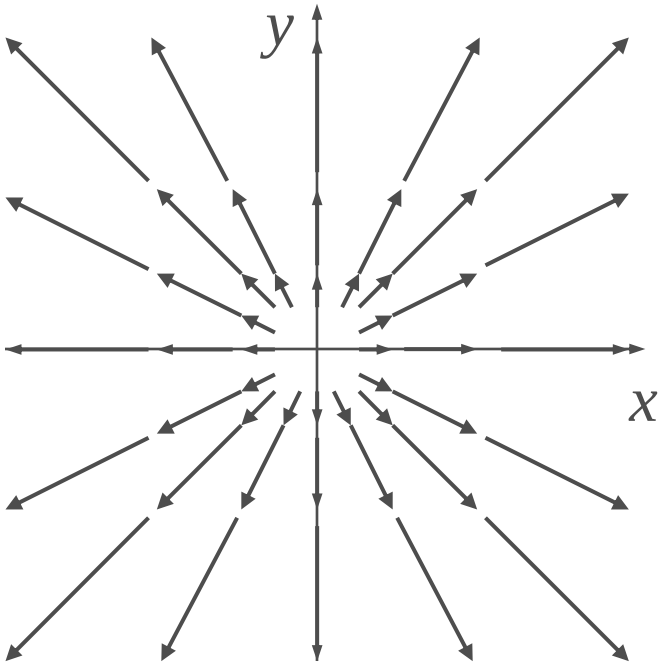
$$div(H_1)(x,y,z) = 0$$

$$curl(H_1)(x,y,z) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Vector and scalar functions

SIEMENS
Ingenuity for life

- Example



The 2D identity vector function

$$\text{Identity}: H_2(x,y,z) = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Velocity increases with distance

$$J_{H_2}(x,y,z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

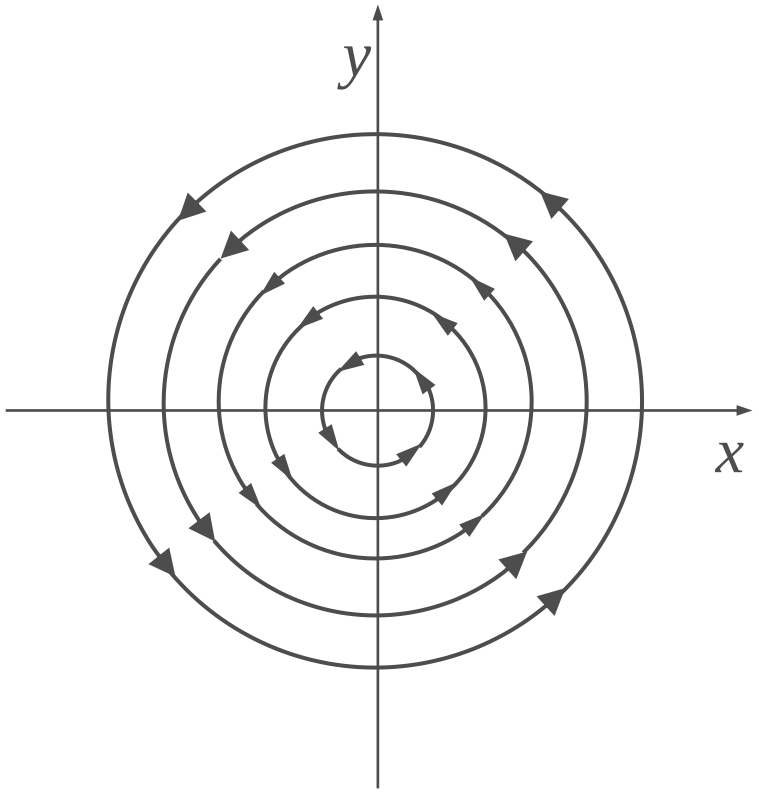
$$div(H_2)(x,y,z) = 3$$

$$curl(H_2)(x,y,z) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Vector and scalar functions

SIEMENS
Ingenuity for life

- Example



A 2D curl field

$$\text{Curl field}: H_3(x,y,z) = \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}$$

$$J_{H_3}(x,y,z) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

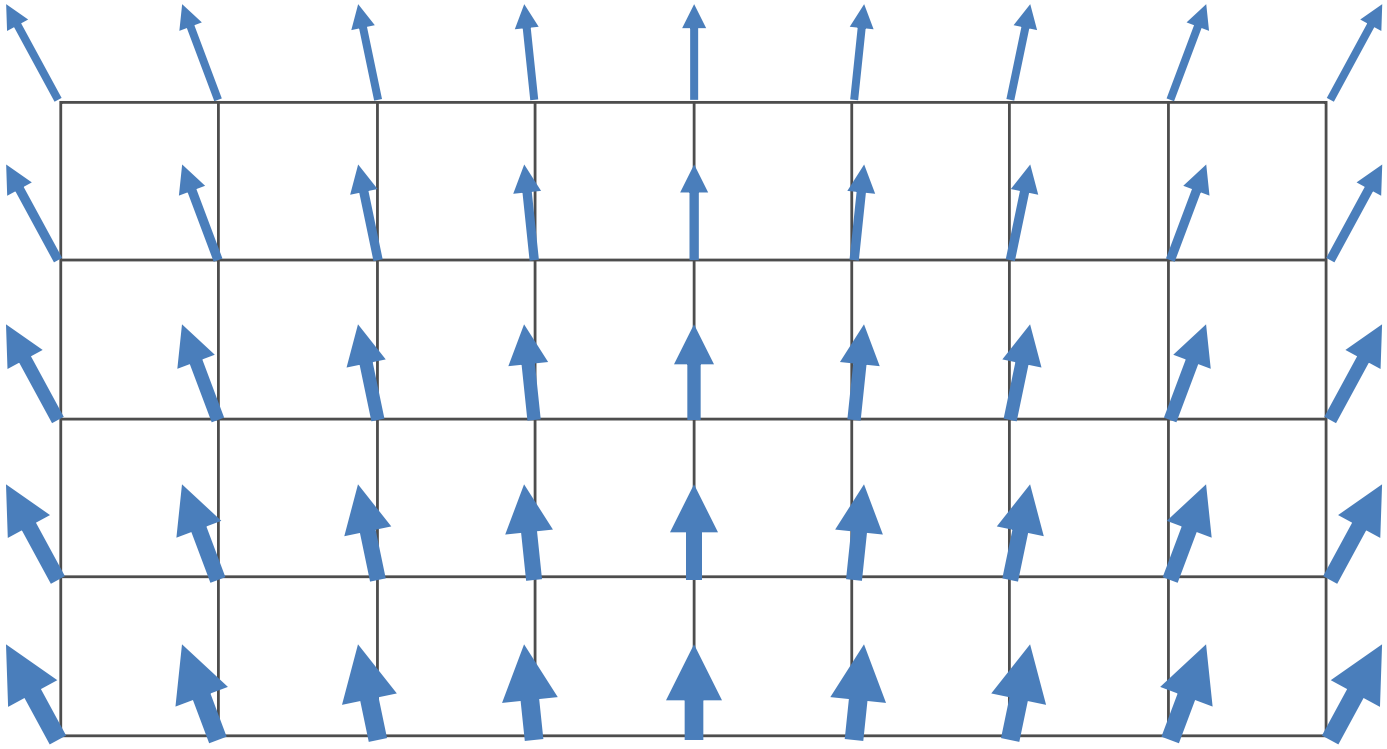
$$div(H_3)(x,y,z) = 0$$

$$curl(H_3)(x,y,z) = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

Flow visualization

SIEMENS
Ingenuity for life

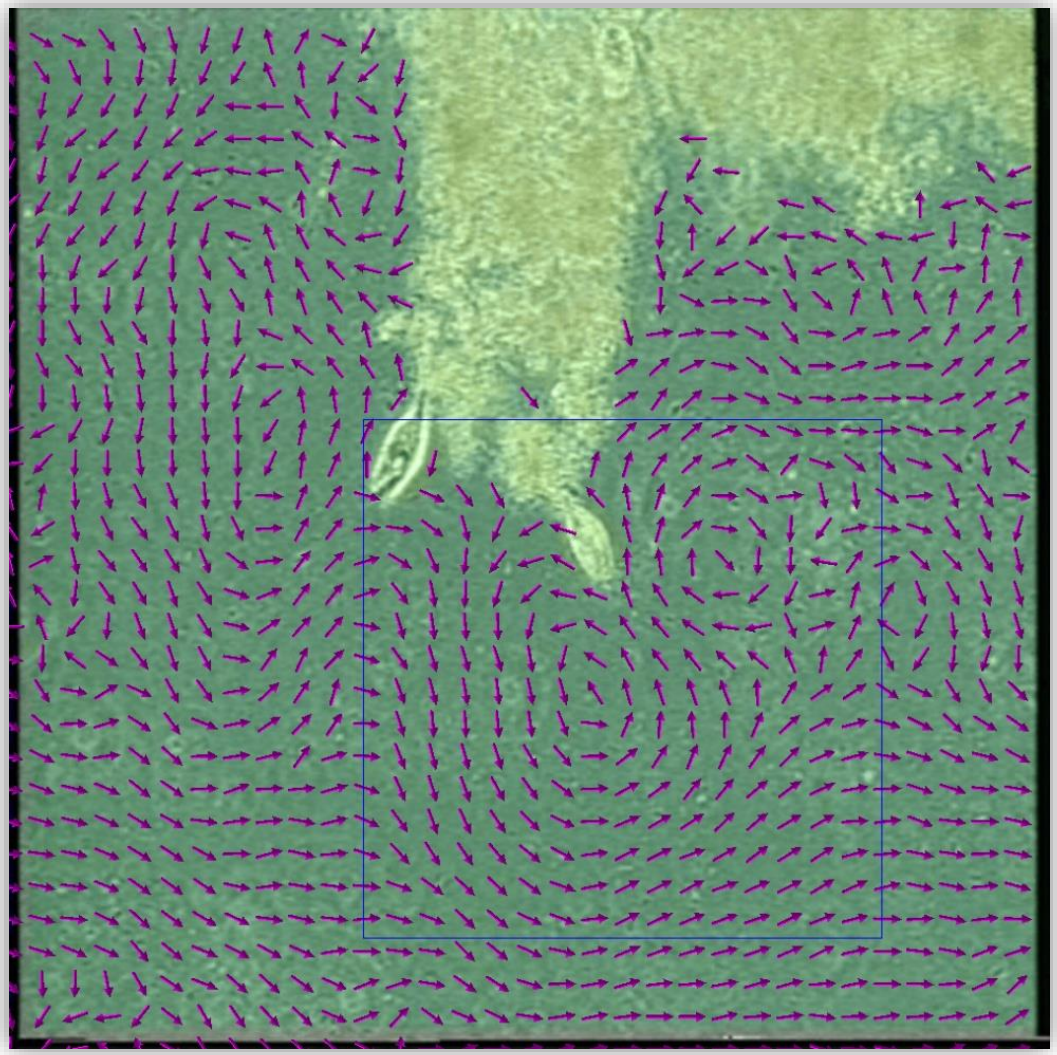
- Glyphs
  - Visualize local features of the vector field
  - Map vector or curl to arrow glyphs



Flow visualization with arrows

SIEMENS
Ingenuity for life

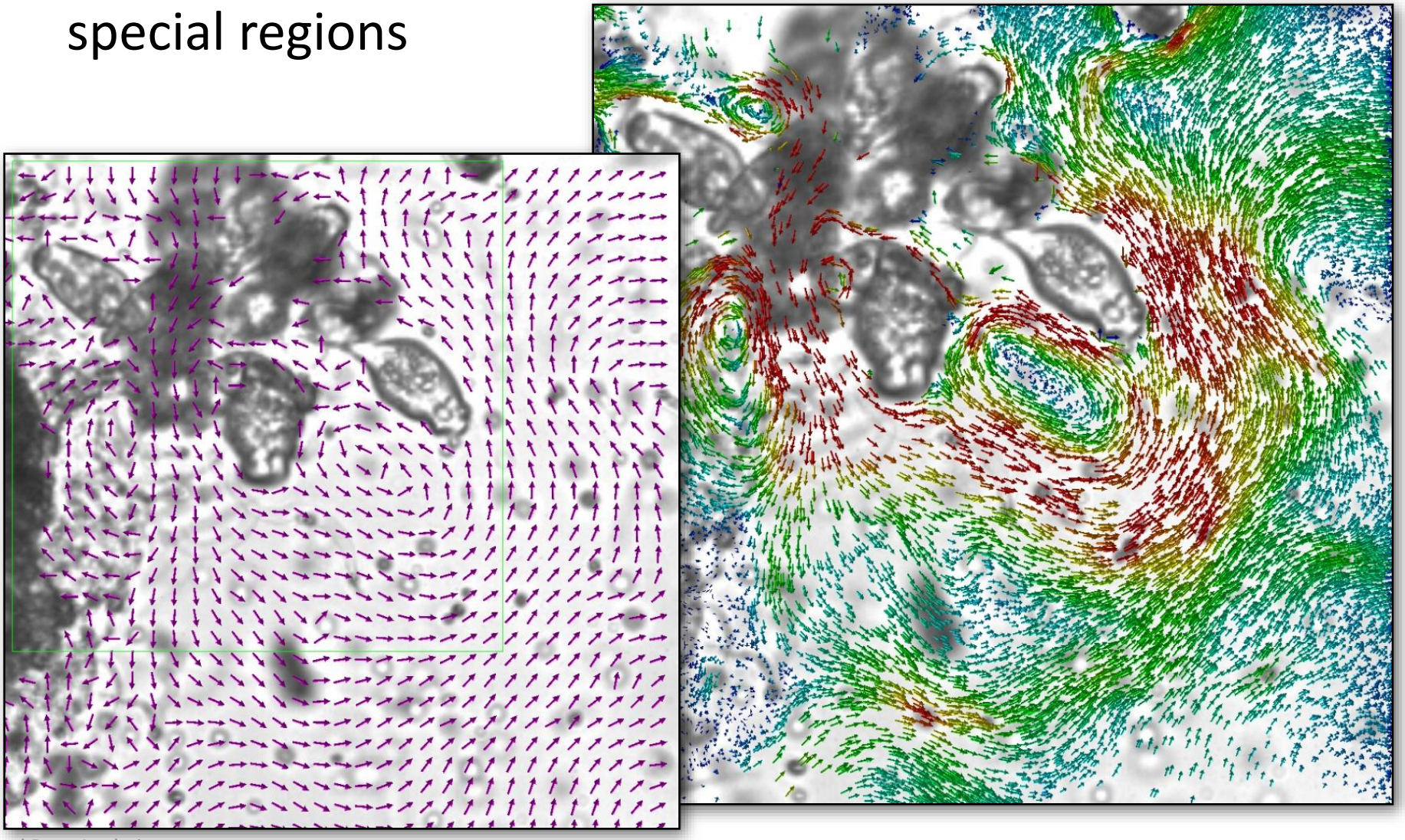
- Vector per grid point pointing into the flow direction



Flow visualization with arrows

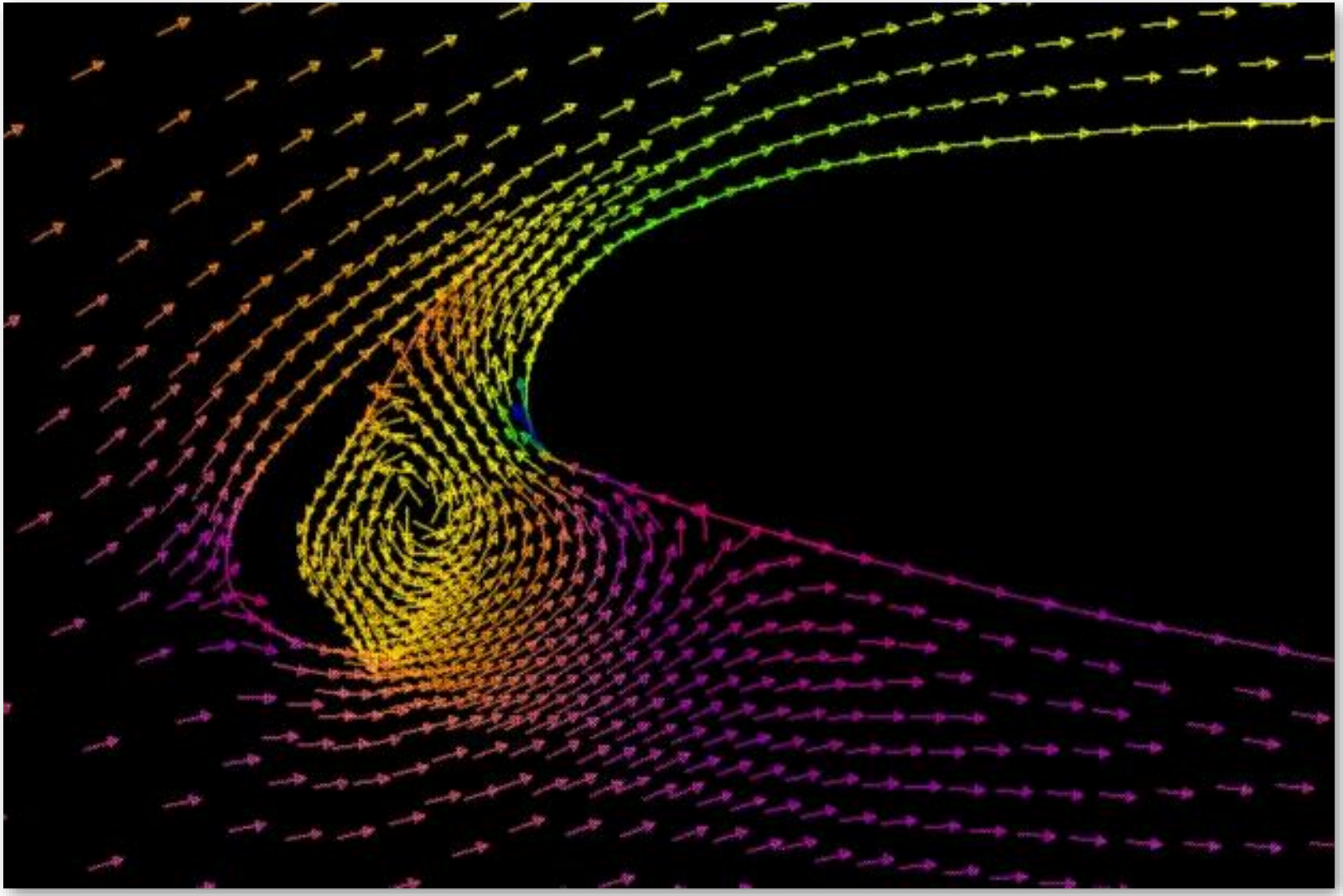
SIEMENS
Ingenuity for life

- Use arrow length and/or color to highlight special regions



Flow visualization with arrows

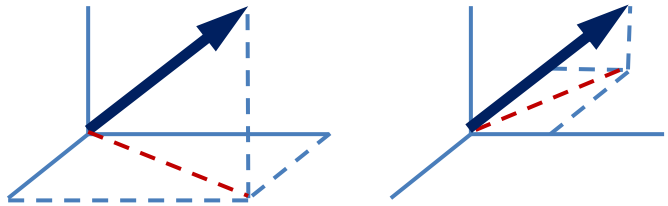
SIEMENS
Ingenuity for life



Arrows in 3D

SIEMENS
Ingenuity for life

- Advantages
  - Simple
  - 3D effects
- Disadvantages
  - Ambiguity
  - Difficult spatial perception (1D-objects in 3D)
  - Inherent occlusion effects
  - Poor results if magnitude of velocity varies significantly and changes rapidly

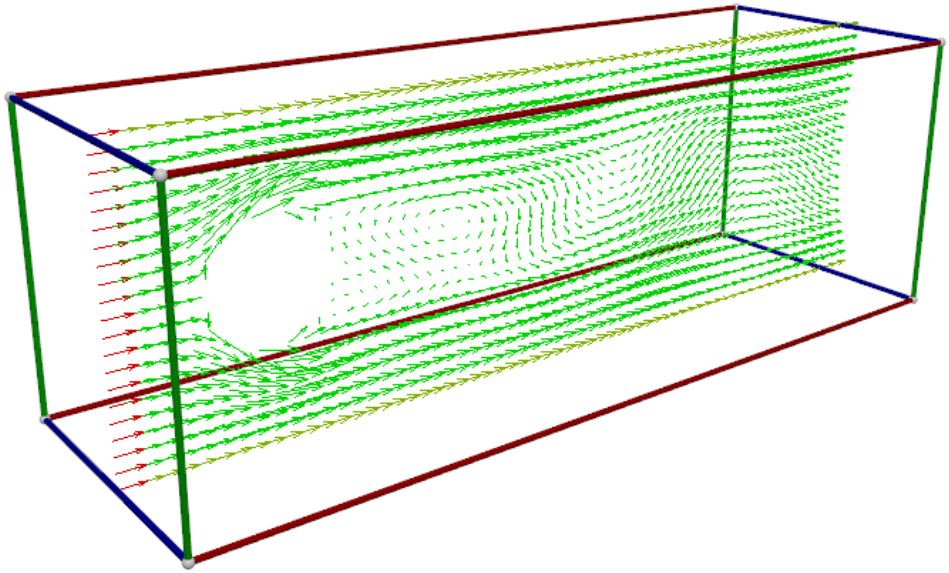


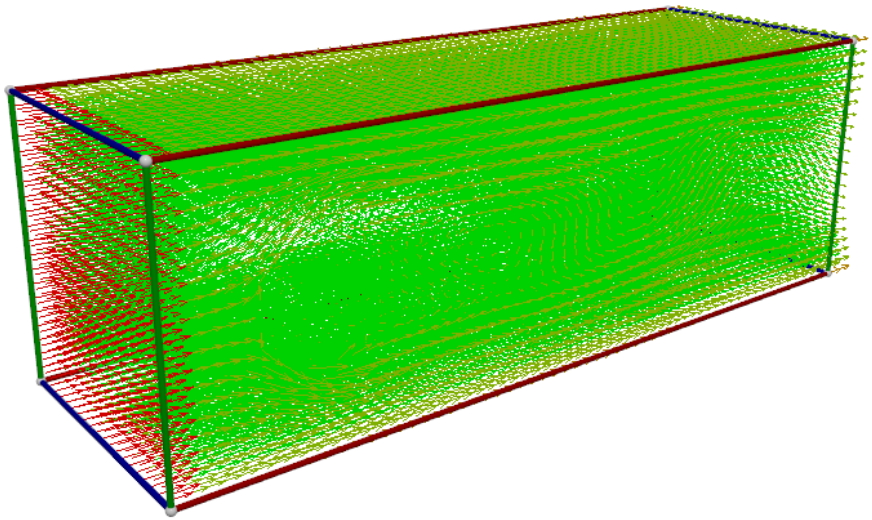
Use 3D arrows of constant length and color code magnitude

Arrows in 3D

SIEMENS
Ingenuity for life

- Compromise
  - Arrows only in slices

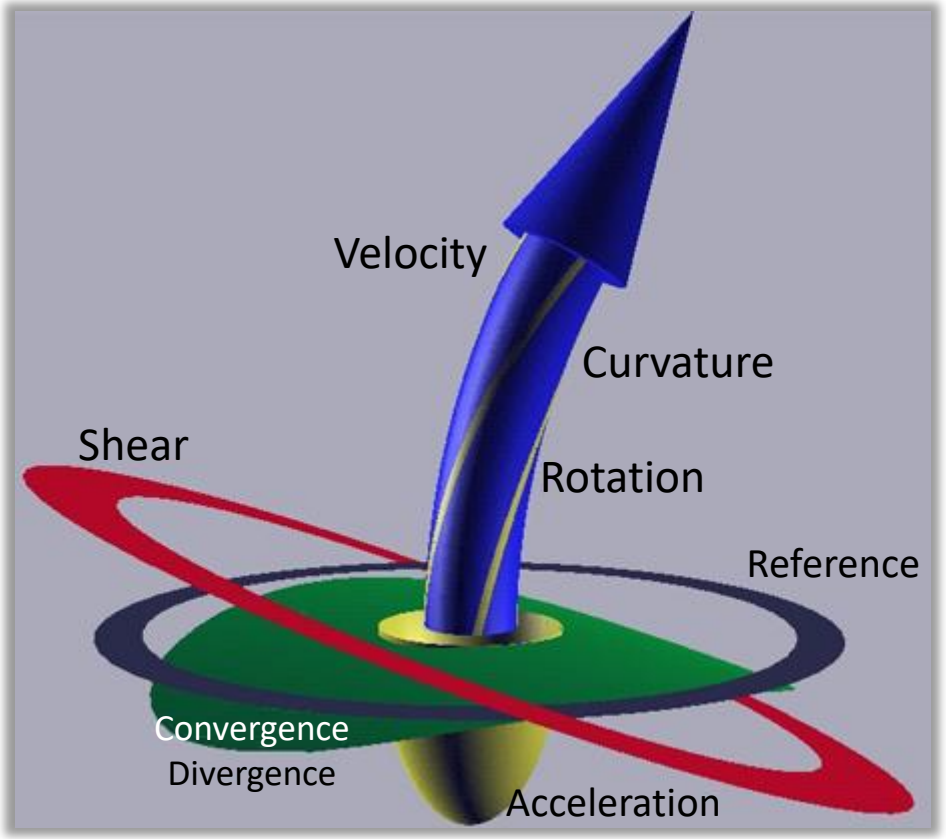




Flow visualization with glyphs

SIEMENS
Ingenuity for life

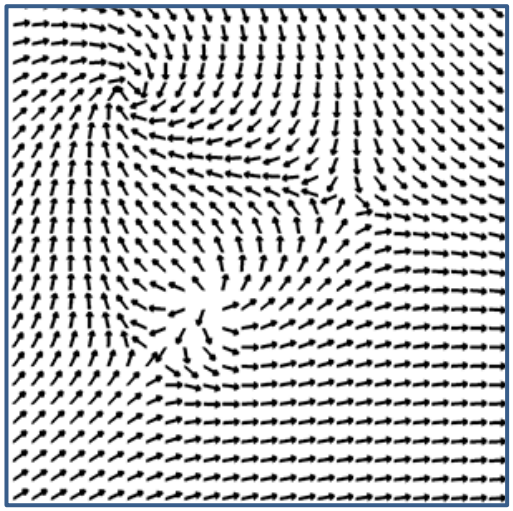
- Glyphs
  - Can visualize more features of vector field



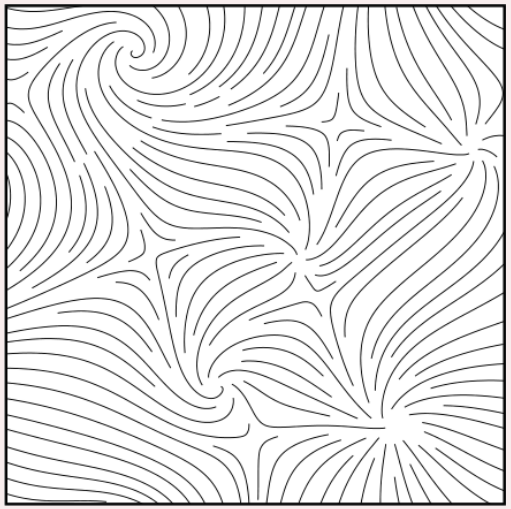
[de Leeuw and van Wijk 93]

Flow visualization – Approaches

SIEMENS
Ingenuity for life

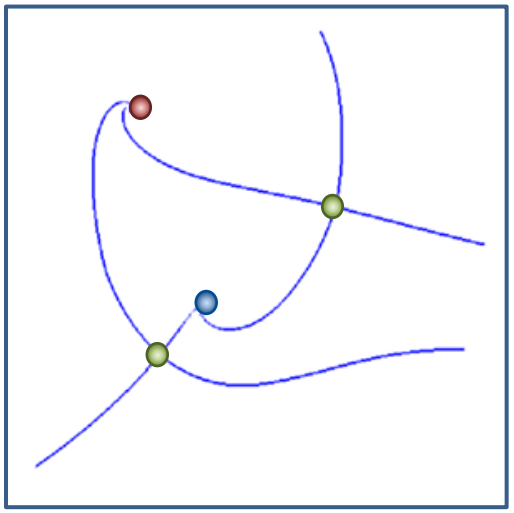


Direct flow visualization
(arrows, color coding, …)

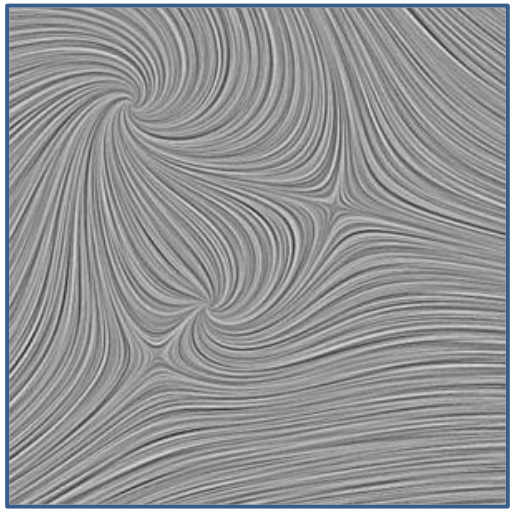


Geometric flow visualization

- Use intermediate representation (vector-field integration over time)
- Visualization of temporal evolution
- Stream lines, path lines, streak lines



Sparse (feature-based) vis.

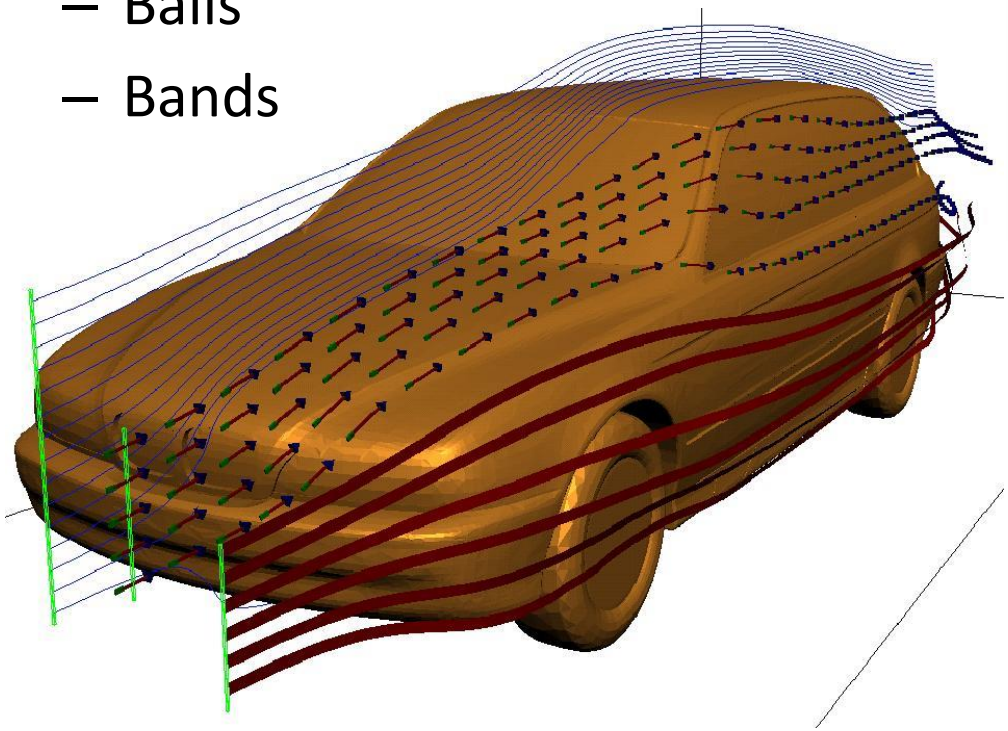


Dense (texture-based) vis.

Characteristic Lines

SIEMENS
Ingenuity for life

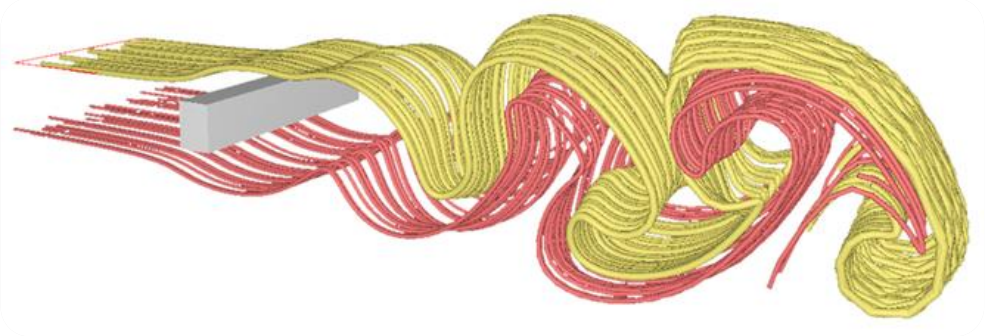
- Basic idea: trace particles along characteristic trajectories
- Map trajectories to
  - Particles
  - Lines
  - Balls
  - Bands



Characteristic Lines

SIEMENS
Ingenuity for life

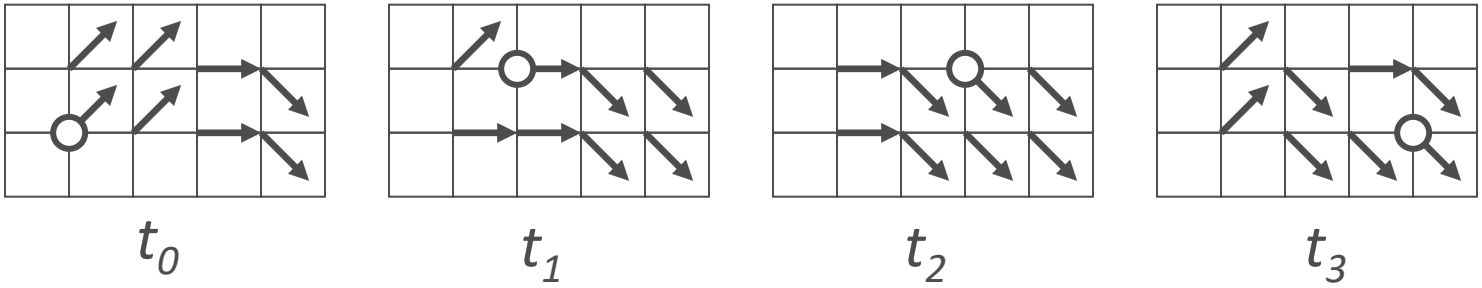
- Types of characteristic lines
  - Stream lines: trajectories of massless particles in a "frozen" (steady) vector field
  - Path lines: trajectories of massless particles in (unsteady/time-varying) flow
  - Streak lines: trace of dye that is continuously released into (unsteady/time-varying) flow at a fixed position

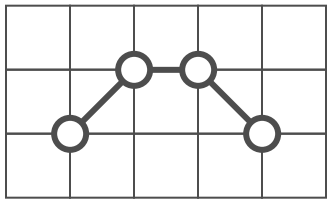


Characteristic Lines

SIEMENS
Ingenuity for life

- Path lines
  - Follow one particle through time and space



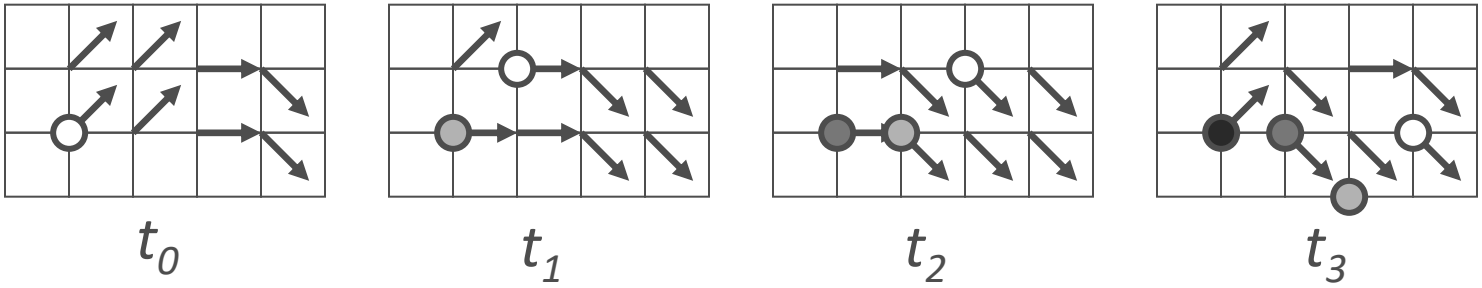


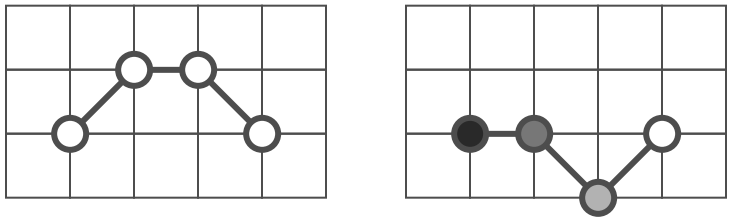
path line

Characteristic Lines

SIEMENS
Ingenuity for life

- Streak lines
  - Connect all particles that started at the same seed point
    - A new particle is continuously injected at the same seed point
    - All existing particles are advected & connected (from youngest to oldest)



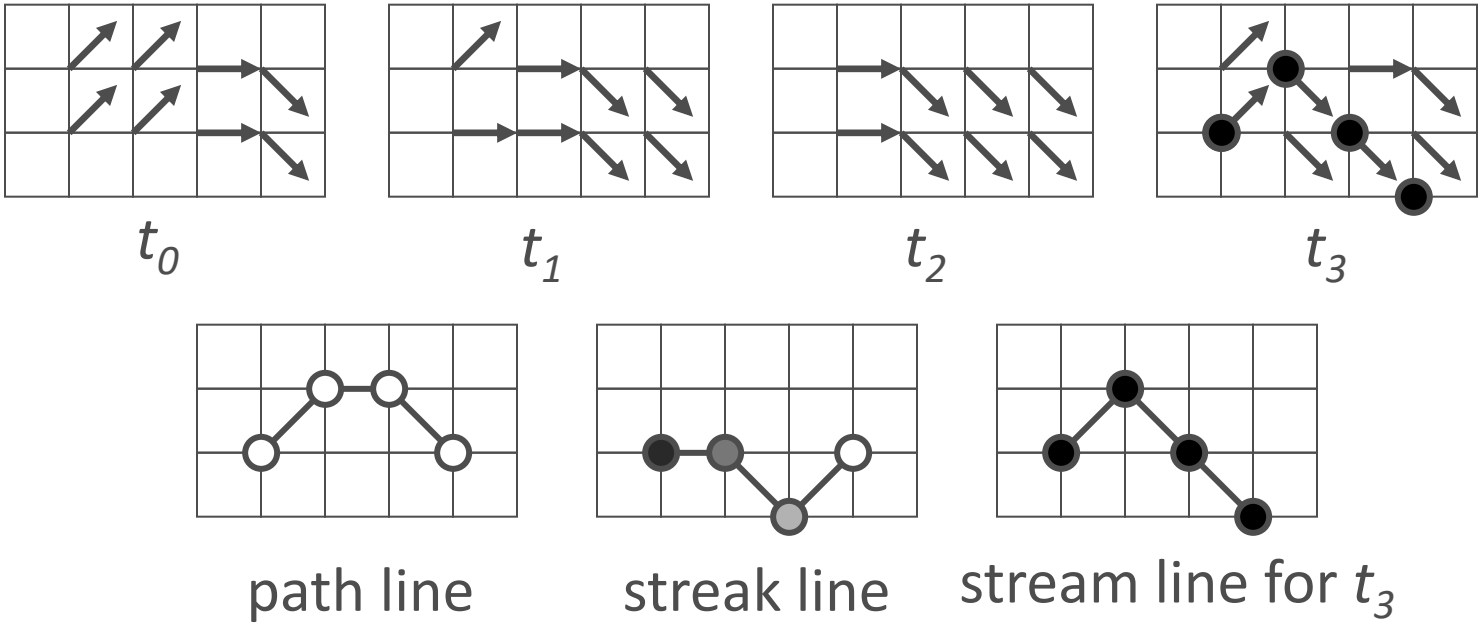


path line streak line

Characteristic Lines

SIEMENS
Ingenuity for life

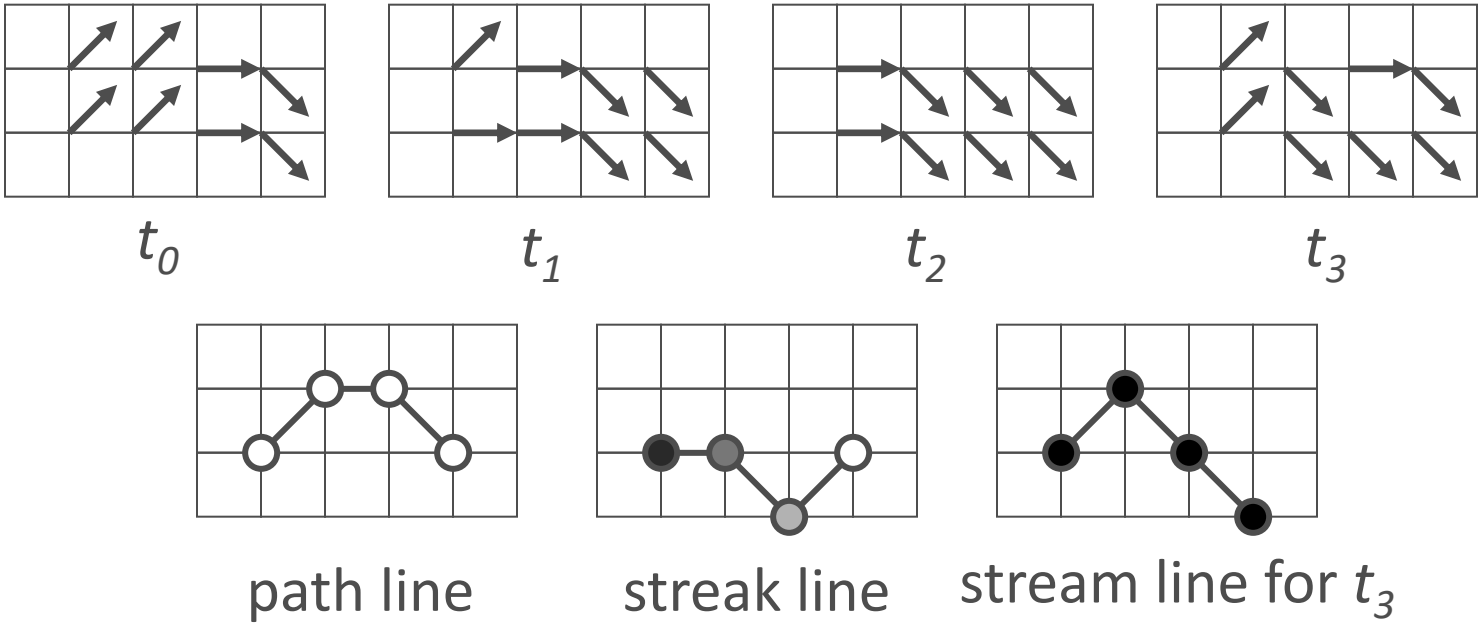
- Stream lines
  - Trajectories of massless particles at one time step



Characteristic Lines

SIEMENS
Ingenuity for life

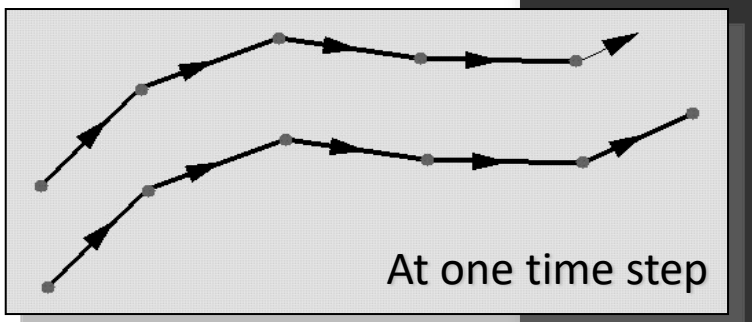
- Comparison of path lines, streak lines, and stream lines
  - Identical for steady flows



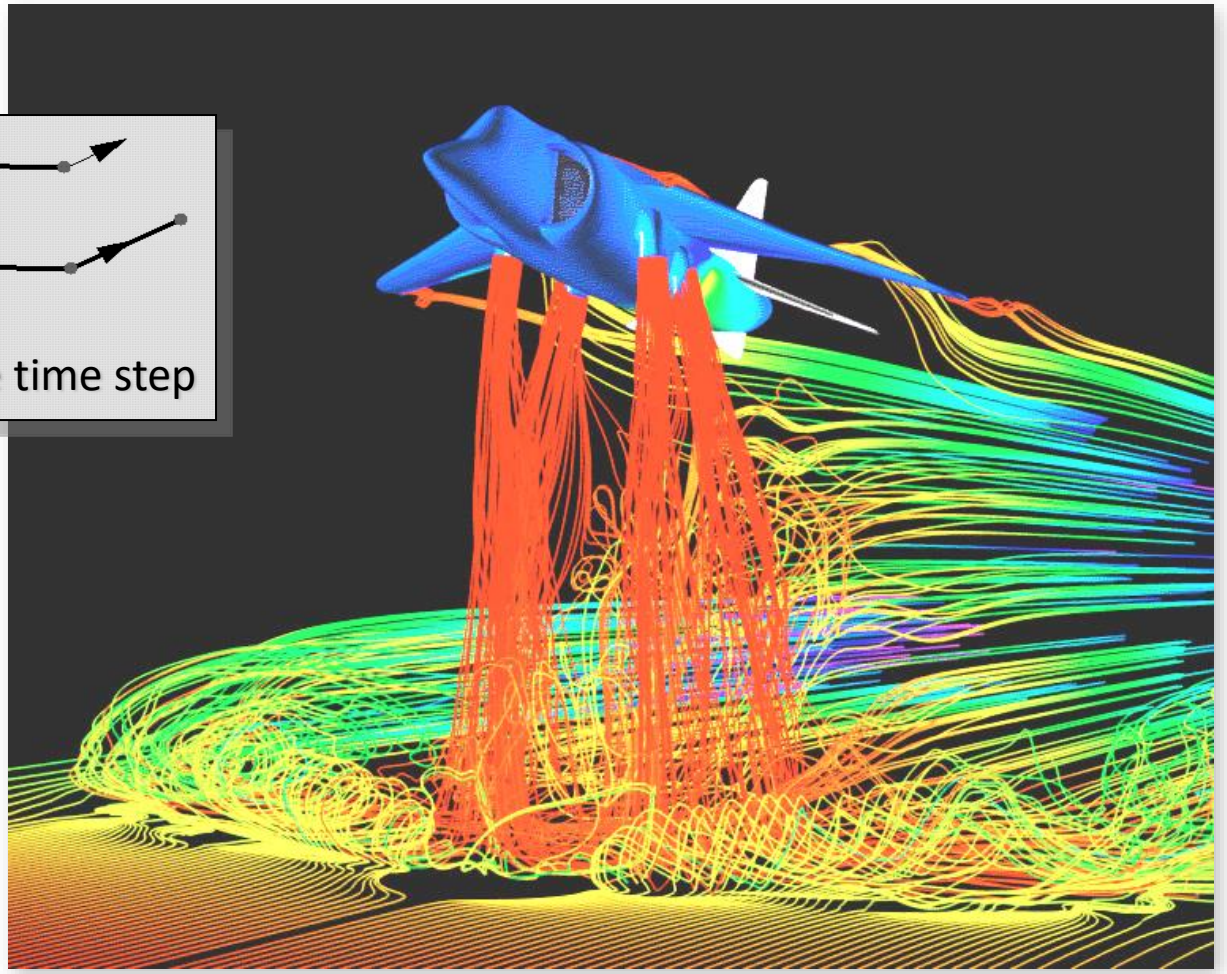
Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

- Stream lines



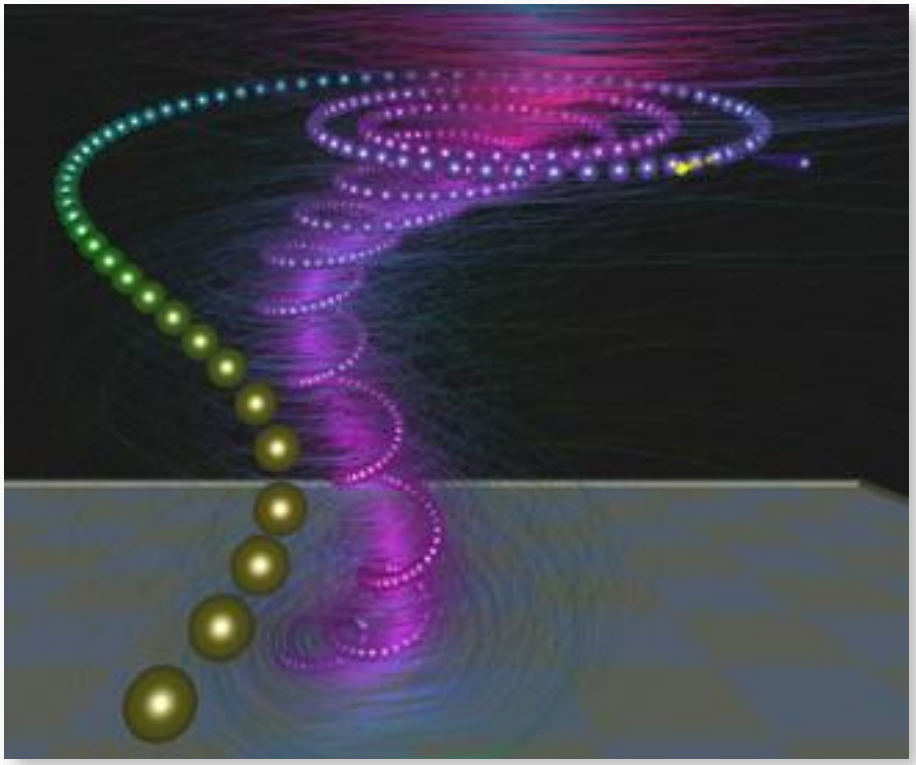
Simple colored stream lines



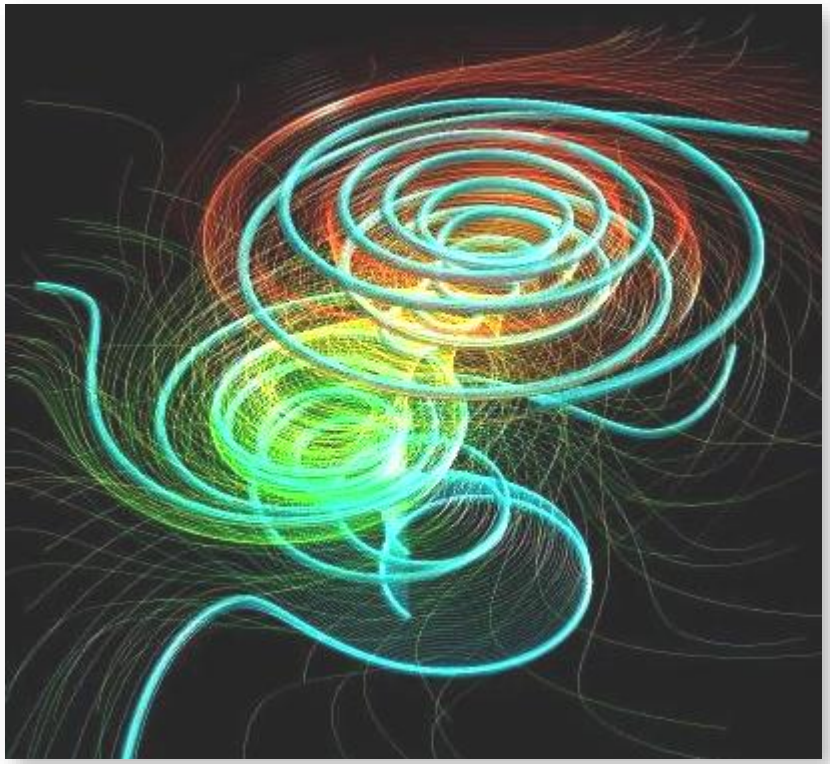
Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

- Stream lines



Stream balls & illuminated stream lines



Stream tubes

Particle tracing, bir madde parçacığının bir akışkan içinde nasıl hareket ettiğini takip etmek için kullanılan bir yöntemdir. Bu yöntem, bir parçacığın başlangıç koordinatlarını belirleyerek, parçacığın türevlerini hesaplayarak ve bu türevleri zamana göre çözerek parçacığın yolunu tahmin etmeye çalışır. Bu yöntem, hareket eden parçacıkların gelecekteki hareketlerini tahmin etmek için kullanılabilir.

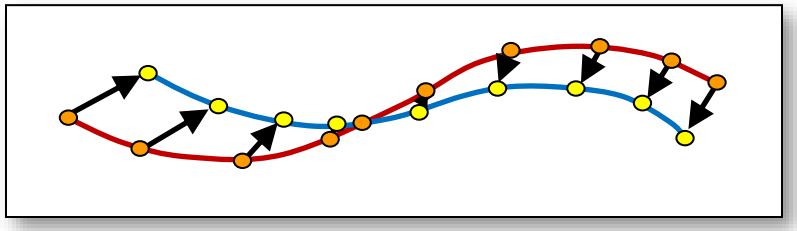
Streamlines, bir akışkan içinde hava veya su gibi maddelerin hareketini göstermek için kullanılan bir yöntemdir. Streamlines, bir akışkanın hızının yönünü gösterir ve bir akışkanın hareketini anlamaya yardımcı olur. Streamlines, bir akışkanın hareketi hakkında bilgi verirken, particle tracing ise bir madde parçacığının hareketi hakkında bilgi verir.

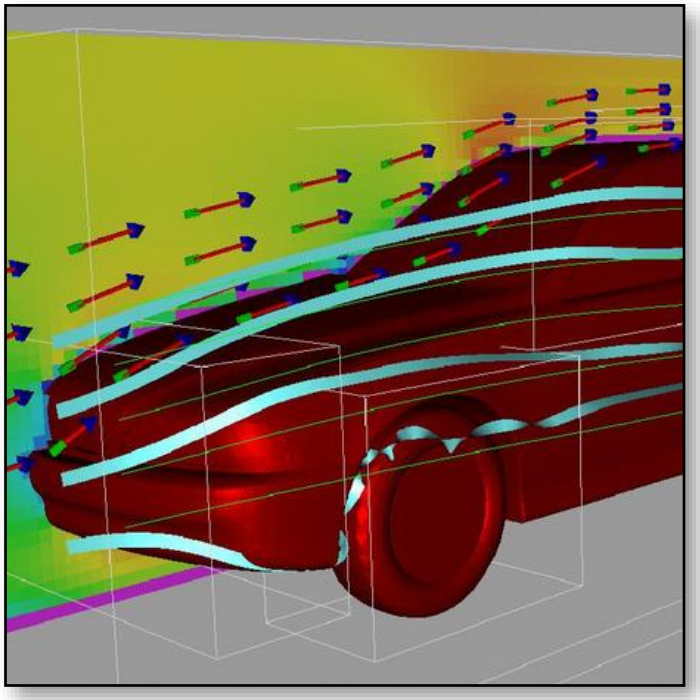
Farklı olarak, streamlines akışkanın hareketini gösterirken, particle tracing ise bir parçacığın hareketini takip eder. Ayrıca, particle tracing yalnızca bir parçacığın hareketini takip ederken, streamlines tüm akışkanın hareketini gösterir.

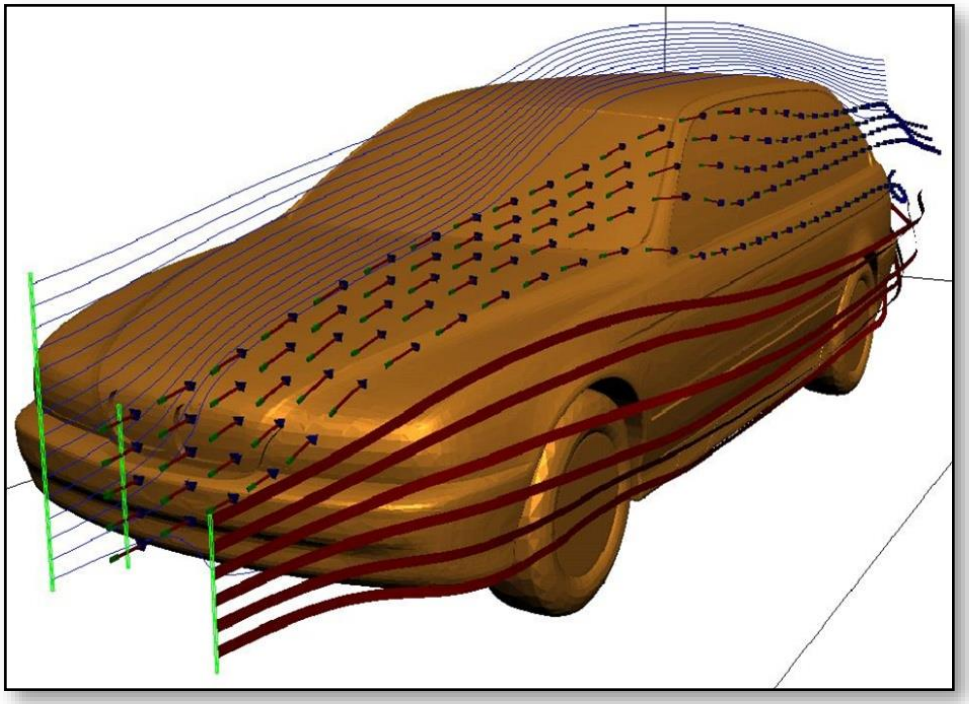
Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

- Stream ribbons (flow oriented)
  - We liked to see places where the flow twists (vortices)
  - Trace two close-by particles
    (keep distance constant)
  - Or rotate band according to curl



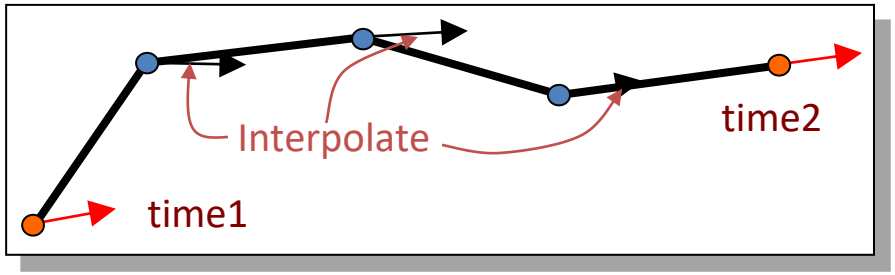


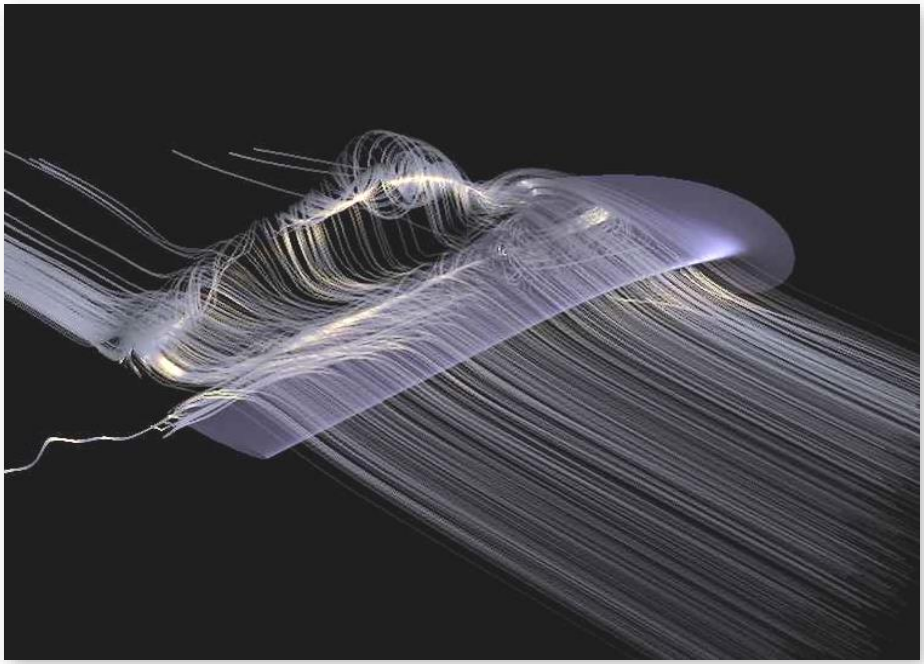


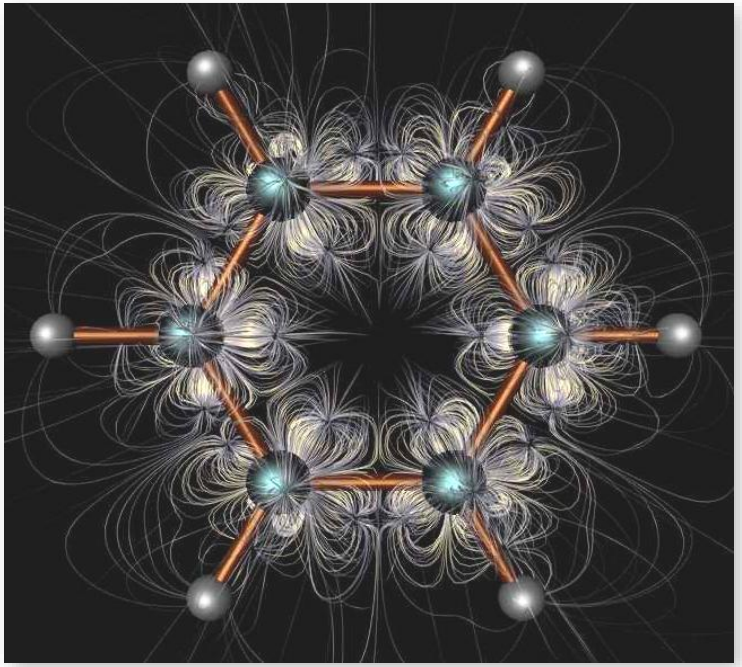
Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

- Path lines
  - Follow one particle through time and space
  - Illuminated 3D curves improves 3D perception



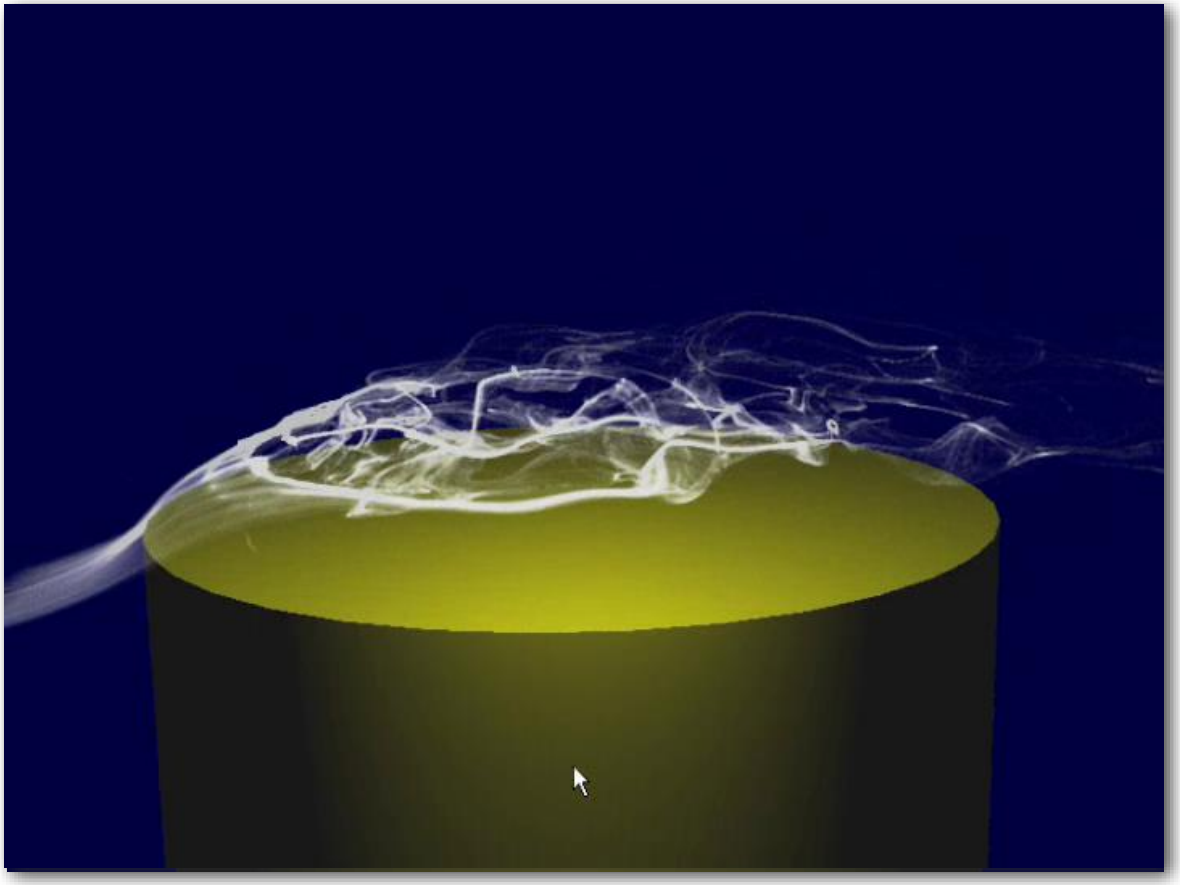




Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

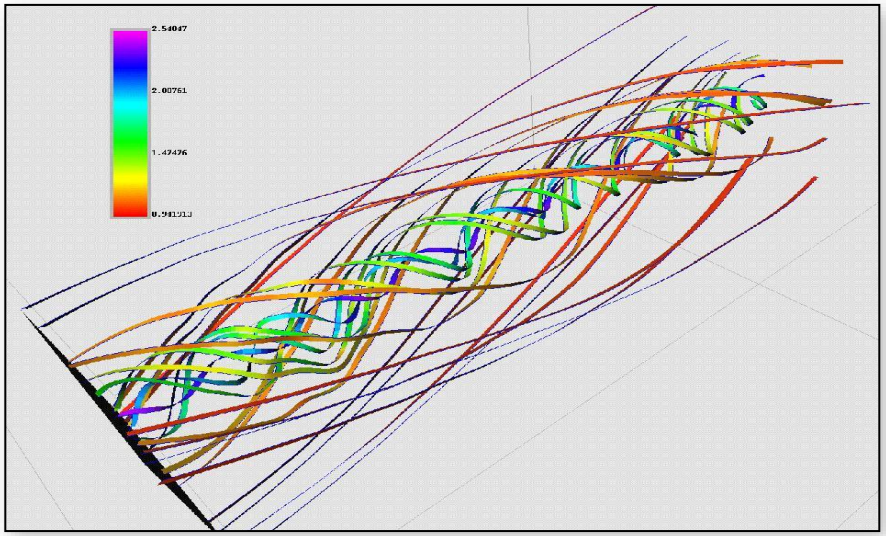
- Particle visualization in unsteady flow

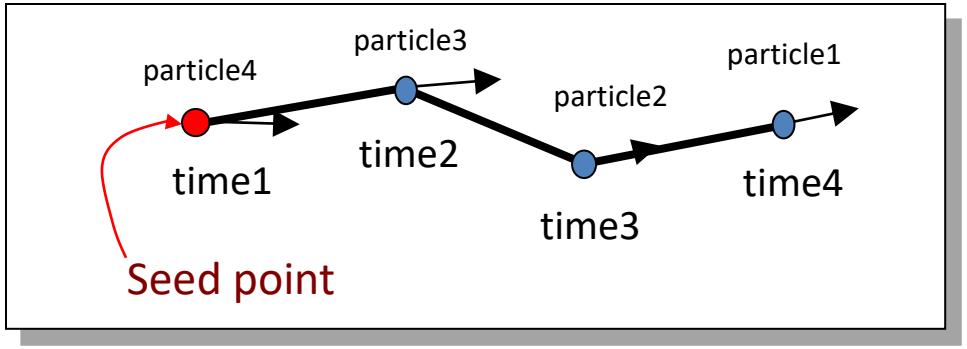


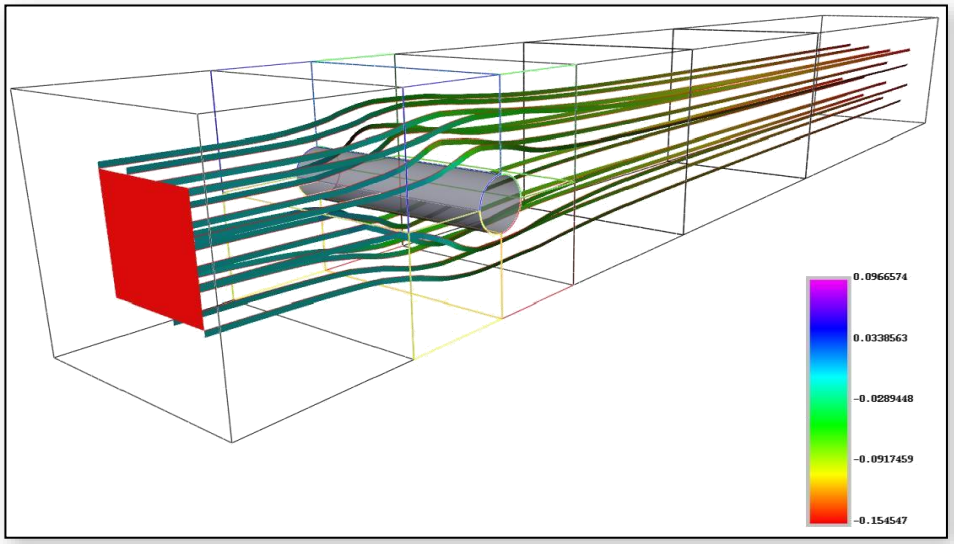
Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

- Streak lines
  - A new particle is continuously injected at the same seed point
  - Existing particles are advected & connected (from youngest to oldest)



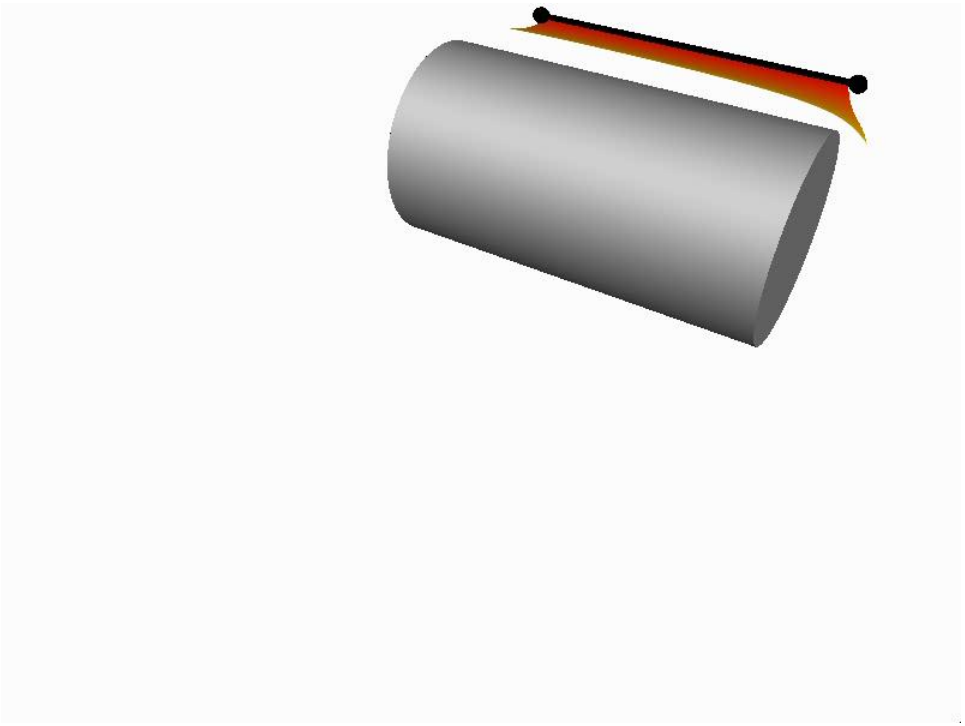




Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

- Streak surface
  - Simultaneously release particles along a seeding structure (line) and connect all them to form a surface

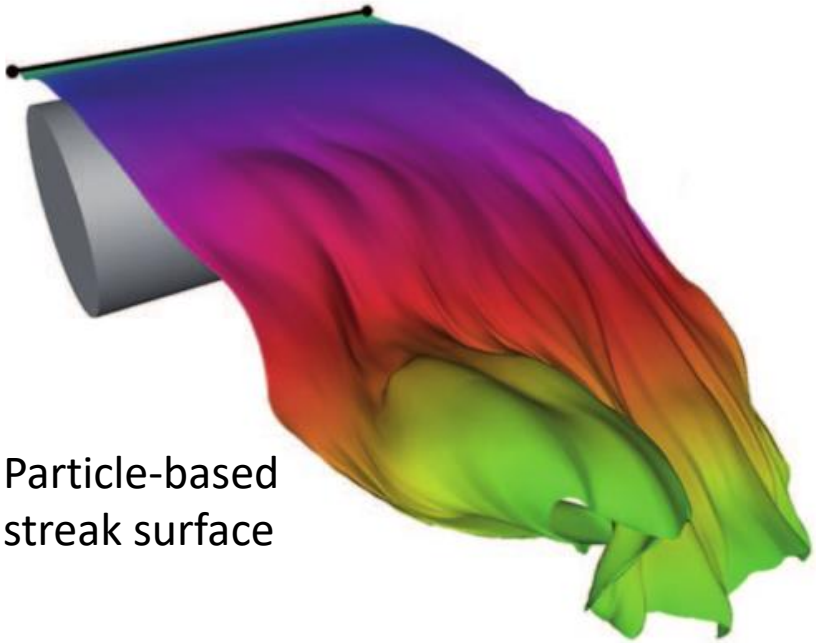


[Bürger et al. 09]

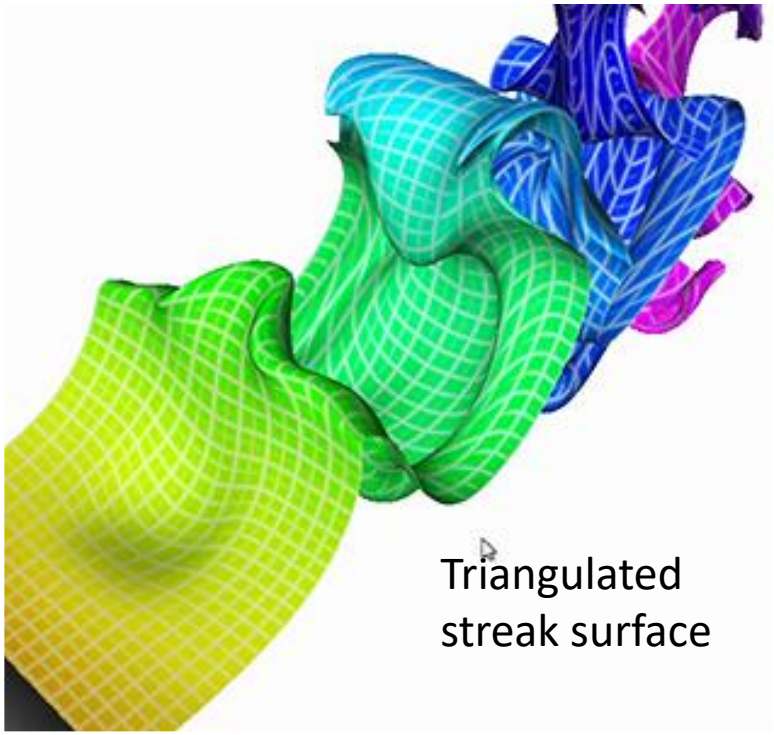
Mapping Based on Particle Tracing

SIEMENS
Ingenuity for life

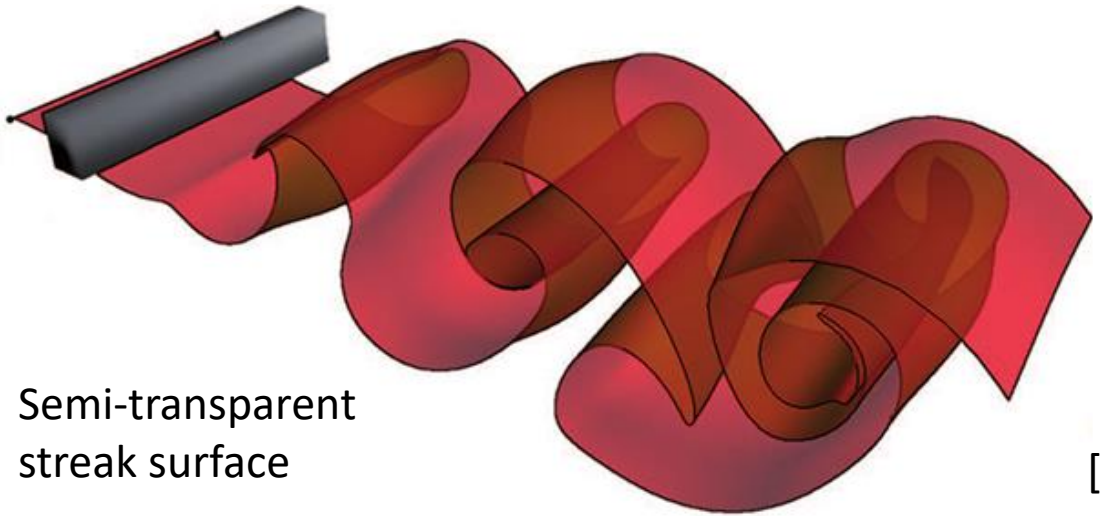
- Streak surface



Particle-based streak surface



Triangulated streak surface



Semi-transparent streak surface

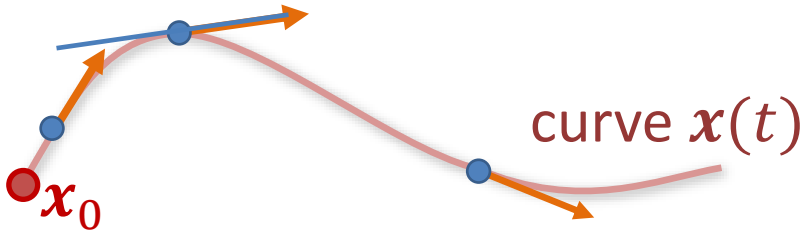
[Bürger et al. 09]

Characteristic Lines

SIEMENS
Ingenuity for life

- Characteristic lines are tangential to the flow
  - Means that line tangent (1st derivative) = vector field direction

$$\frac{\partial \boldsymbol{x}(t)}{\partial t} = \boldsymbol{v}(\boldsymbol{x}(t), t)$$



  - Characteristic lines do not intersect!
  - They are solutions to the initial value problem of an ordinary differential equation (ODE)

$$\boldsymbol{x}(0) = \boldsymbol{x}_0\,, \qquad \frac{\partial \boldsymbol{x}(t)}{\partial t} = \boldsymbol{v}(\boldsymbol{x}(t), t)$$

Initial value (seed point $\boldsymbol{x}_0$)

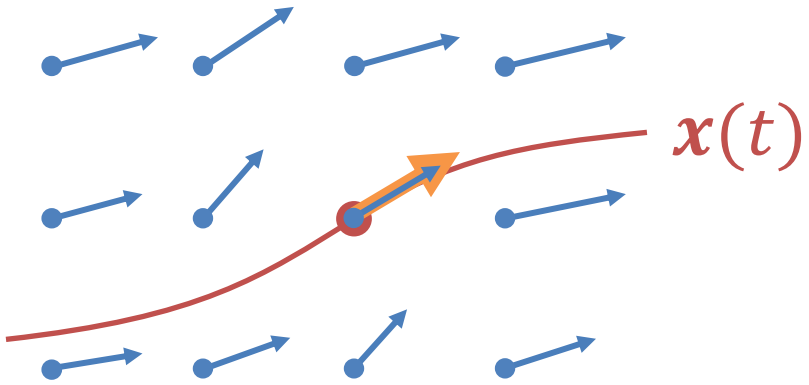
Ordinary differential equation (ODE)

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Particle tracing problem – how to solve the differential equation:

$$\boldsymbol{x}(0) = \boldsymbol{x}_0\,, \qquad \dot{\boldsymbol{x}}(t) = \frac{\partial \boldsymbol{x}(t)}{\partial t} = \boldsymbol{v}(\boldsymbol{x}(t), t)$$



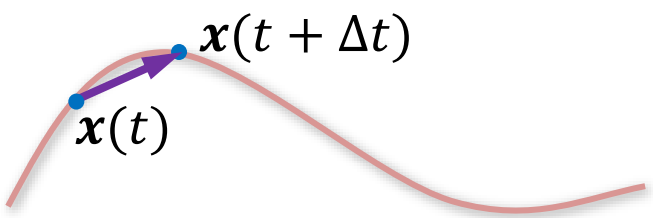
- What kind of numerical solver?

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

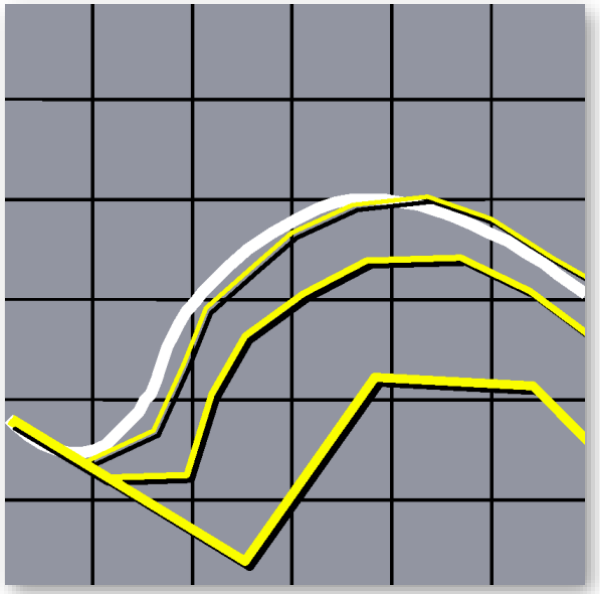
- Rewrite ODE in generic form
- Initial value problem for: $\dot{\boldsymbol{x}}(t) = \boldsymbol{v}(\boldsymbol{x}(t), t)$
- Simple approach: Euler method

$$\frac{\boldsymbol{x}(t + \Delta t) - \boldsymbol{x}(t)}{\Delta t} \approx \boldsymbol{v}(\boldsymbol{x}(t), t)$$



$$\Rightarrow \boldsymbol{x}(t + \Delta t) \approx \boldsymbol{x}(t) + \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}(t), t)$$

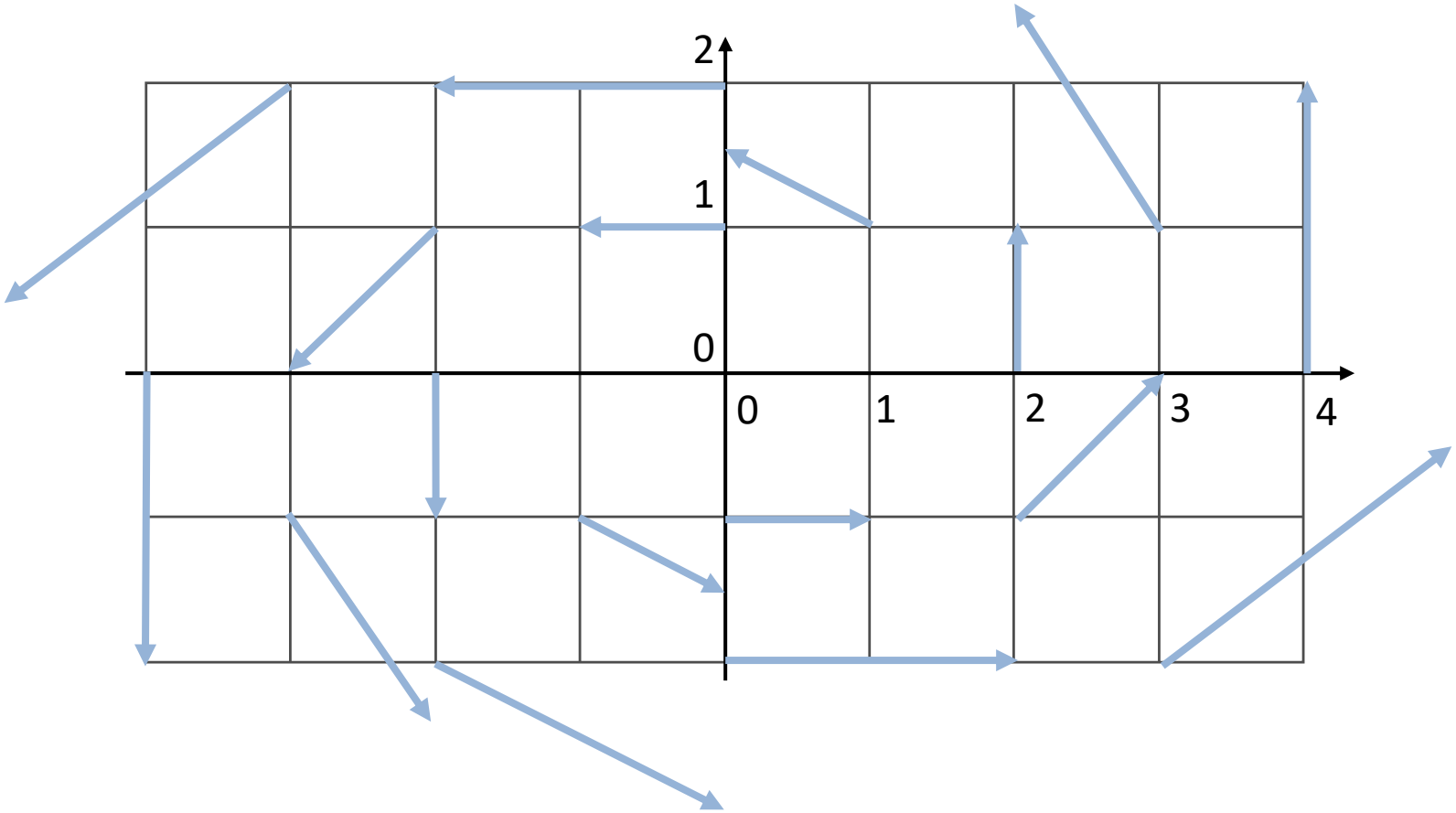
- First order method
- Higher accuracy with smaller step size



Numerical Integration of ODEs

SIEMENS
Ingenuity for life

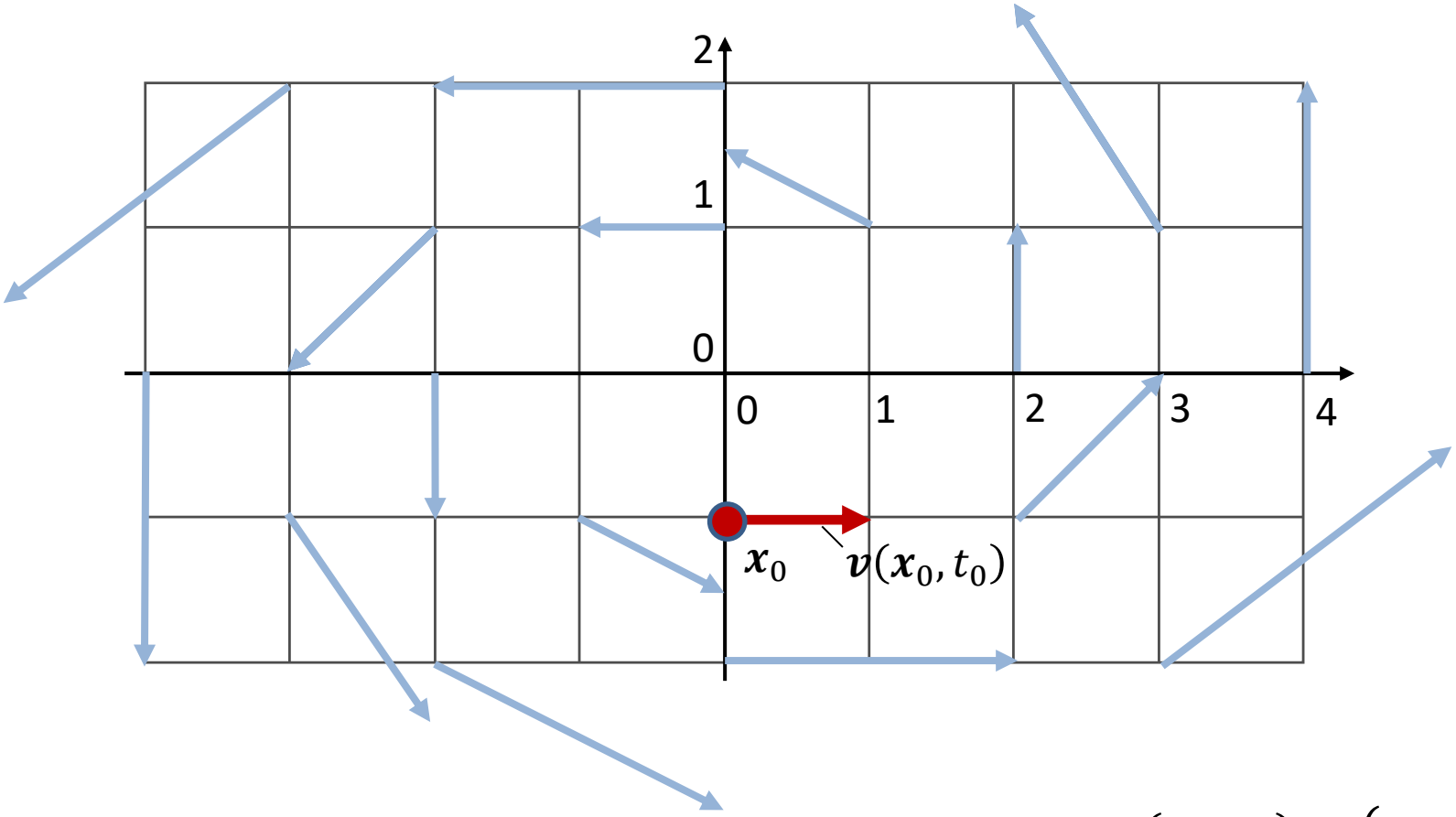
- Example: 2D vector field $\boldsymbol{v}(x, y, t) = (-y, {}^{x}/_{2})$
  - Sample arrows shown
  - True solution are ellipses



Numerical Integration of ODEs

SIEMENS
Ingenuity for life

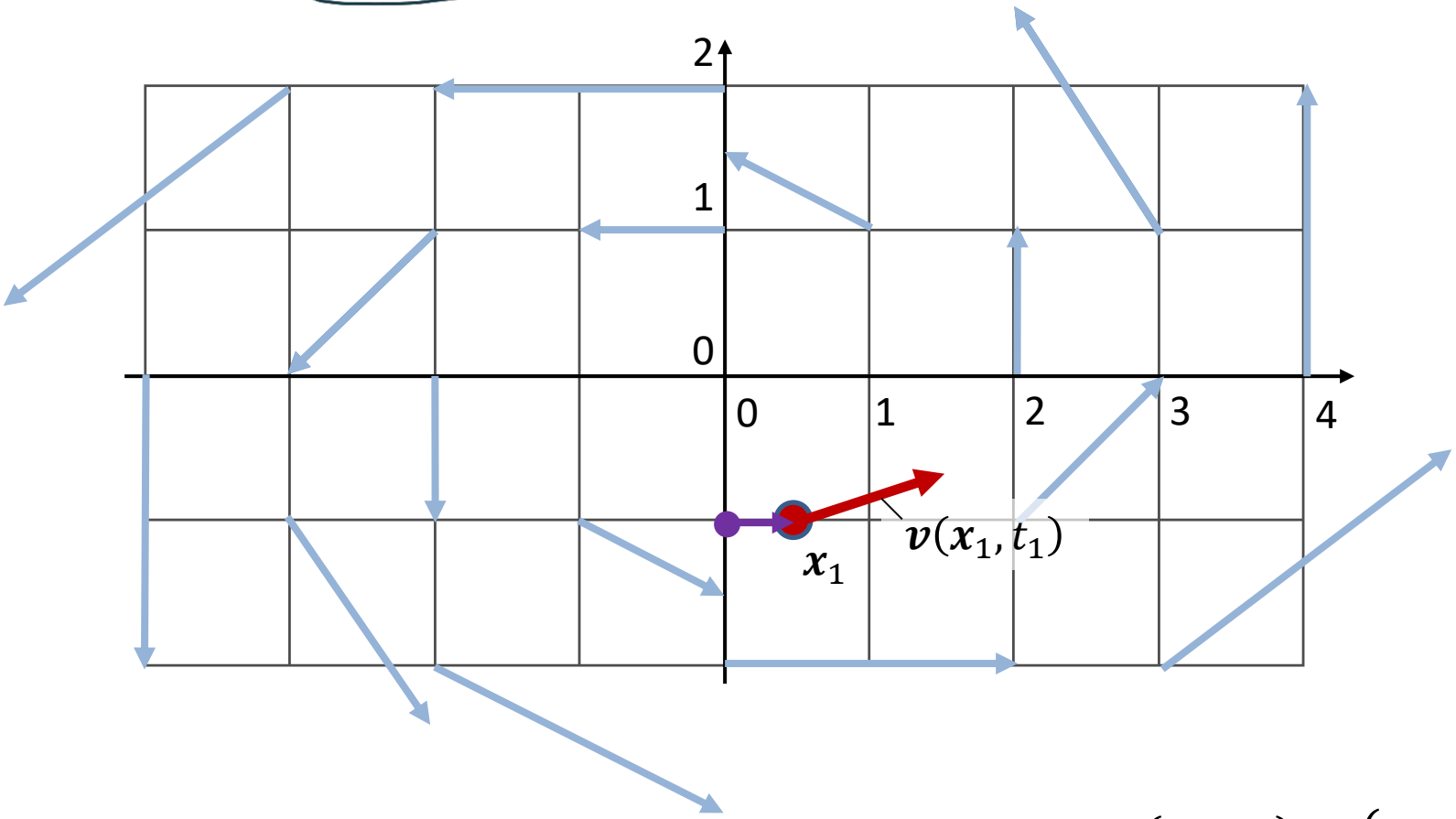
- Example: Euler Integration ($\Delta t = 0.5$)
  - Seed point: $\boldsymbol{x}_0 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$, current vector $\boldsymbol{v}(\boldsymbol{x}_0, t_0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$



$$\boldsymbol{v}(x, y, t) = \left(-y, {}^{x}/_{2}\right)$$

Numerical Integration of ODEs

- Example: Euler Integration ($\Delta t = 0.5$)
  - New point: $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}_0, t_0) = \begin{pmatrix} 0.5 \\ -1 \end{pmatrix}$, $\boldsymbol{v}(\boldsymbol{x}_1, t_1) = \begin{pmatrix} 1 \\ 0.25 \end{pmatrix}$

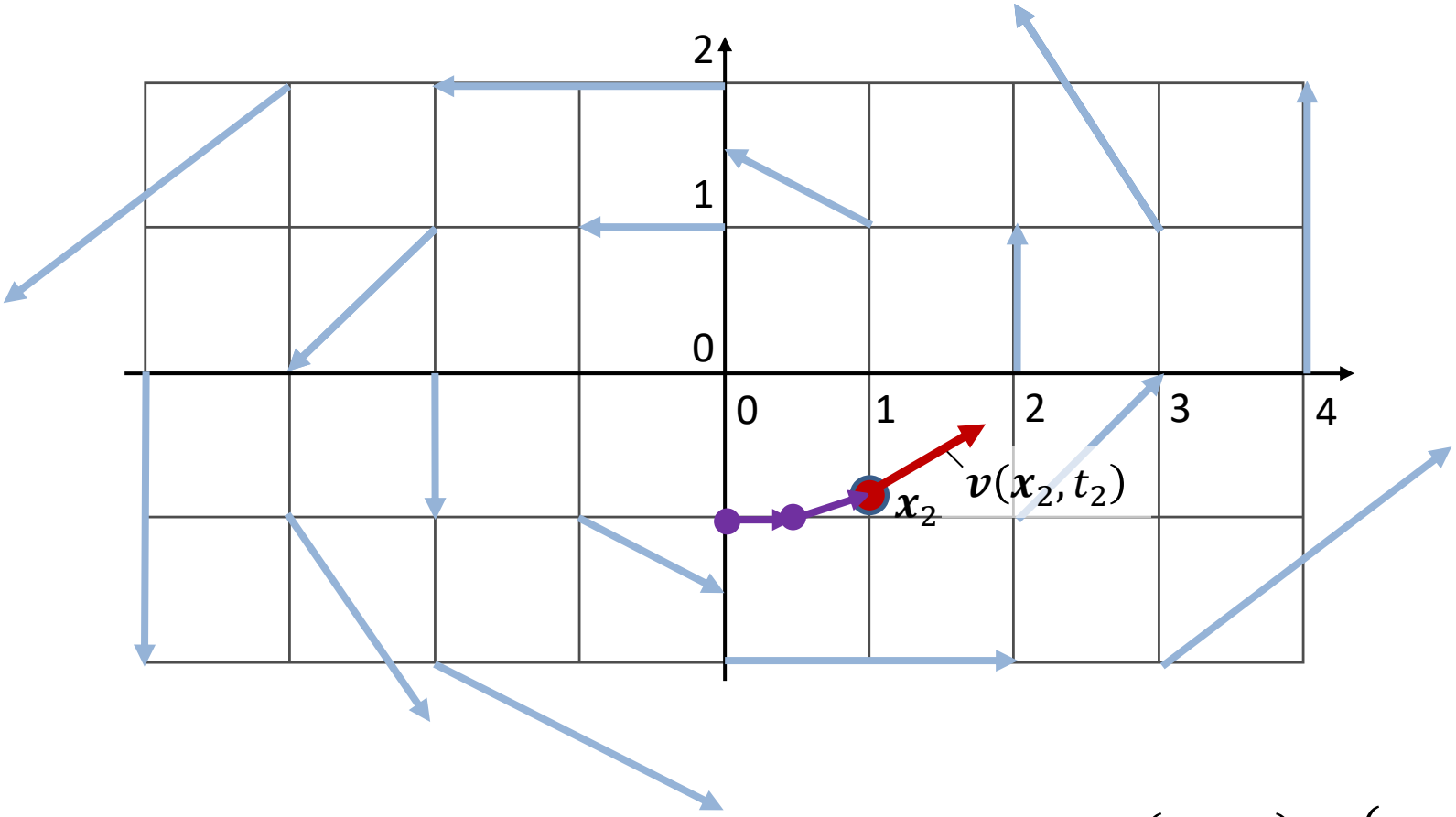


$$\boldsymbol{v}(x, y, t) = \left(-y, {}^{x}/_{2}\right)$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Example: Euler Integration ($\Delta t = 0.5$)
  - New point: $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}_1, t_1) \approx \begin{pmatrix} 1 \\ -0.88 \end{pmatrix}$, $\boldsymbol{v}(\boldsymbol{x}_2, t_2) \approx \begin{pmatrix} 0.88 \\ 0.5 \end{pmatrix}$

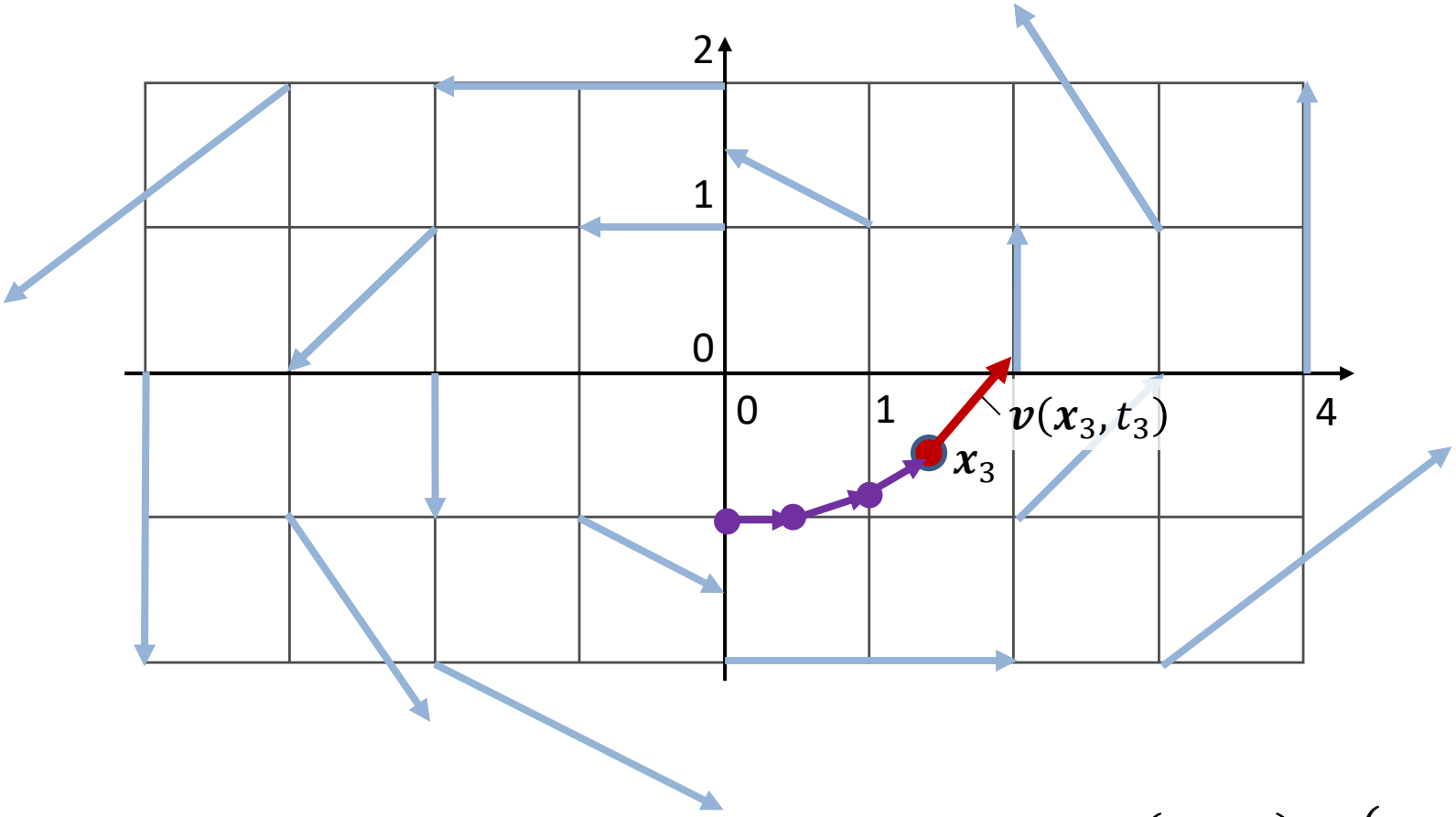


$$\boldsymbol{v}(x, y, t) = \left(-y, {}^{x}/_{2}\right)$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Example: Euler Integration ($\Delta t = 0.5$)
  - New point: $\boldsymbol{x}_3 = \boldsymbol{x}_2 + \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}_2, t_2) \approx \begin{pmatrix} 1.44 \\ -0.63 \end{pmatrix}$, $\boldsymbol{v}(\boldsymbol{x}_3, t_3) \approx \begin{pmatrix} 0.63 \\ 0.72 \end{pmatrix}$

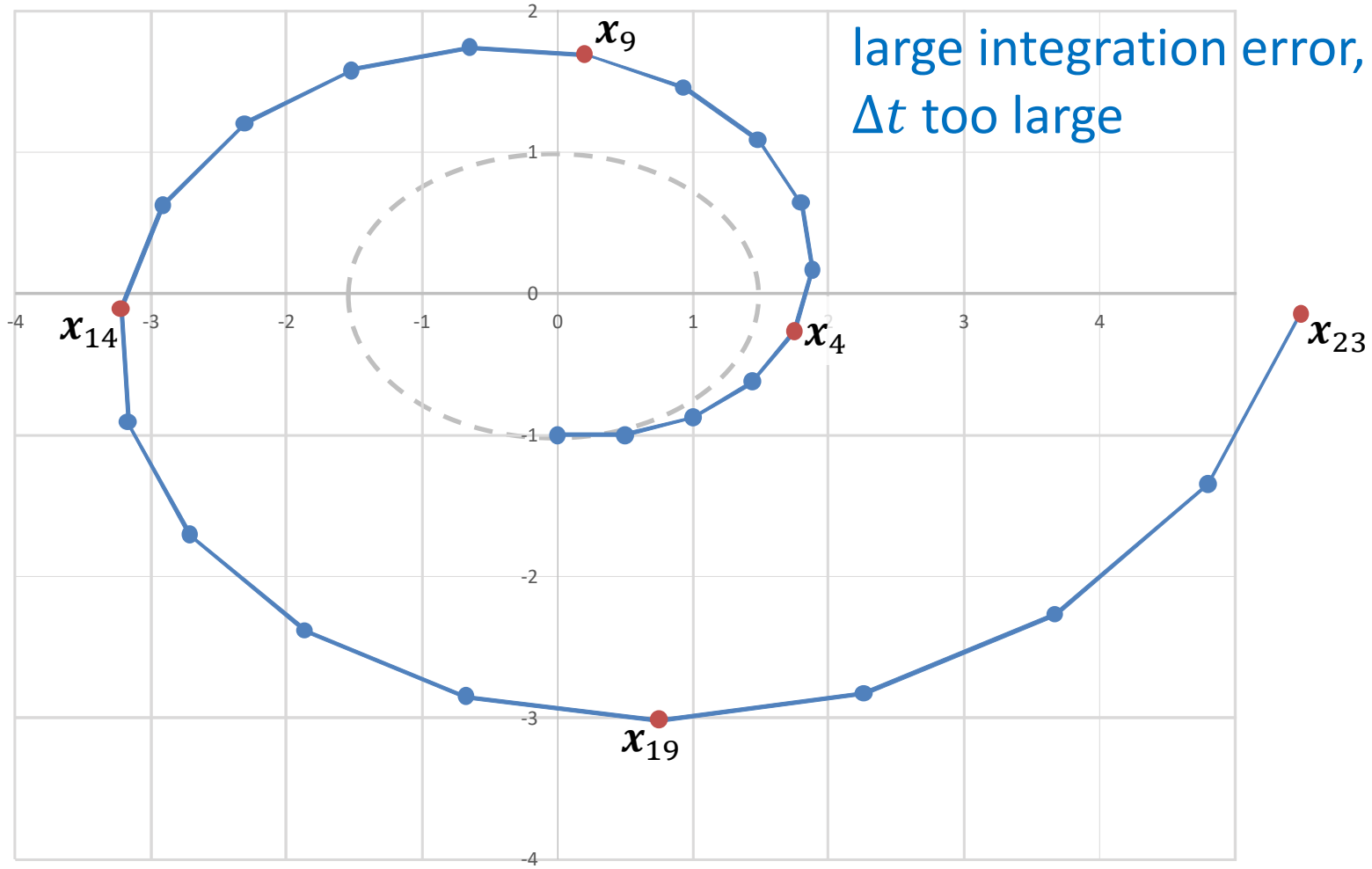


$$\boldsymbol{v}(x, y, t) = \left(-y, {}^{x}/_{2}\right)$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

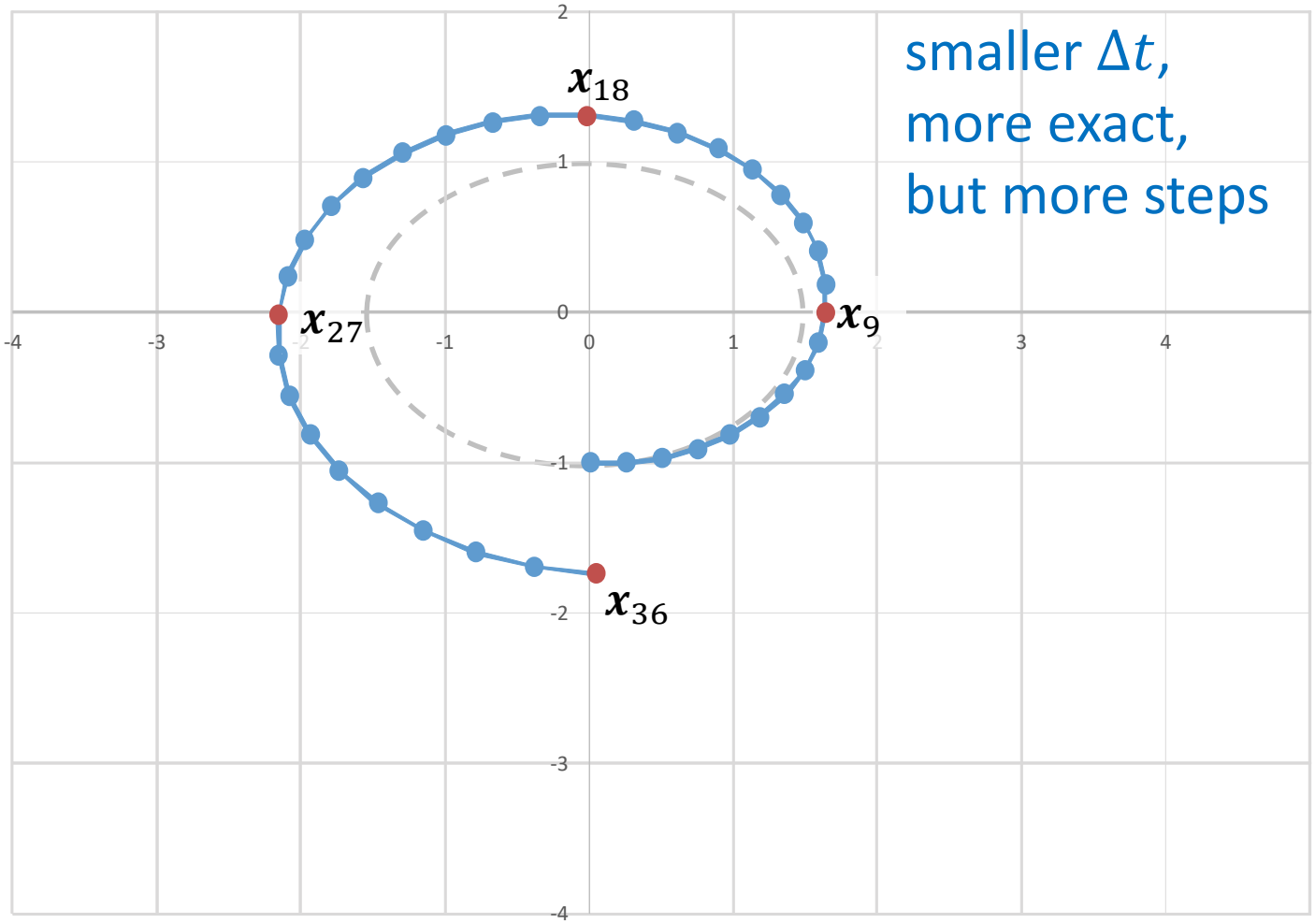
- Euler Integration after 23 steps ($\Delta t = 0.5$)



Numerical Integration of ODEs

SIEMENS
Ingenuity for life

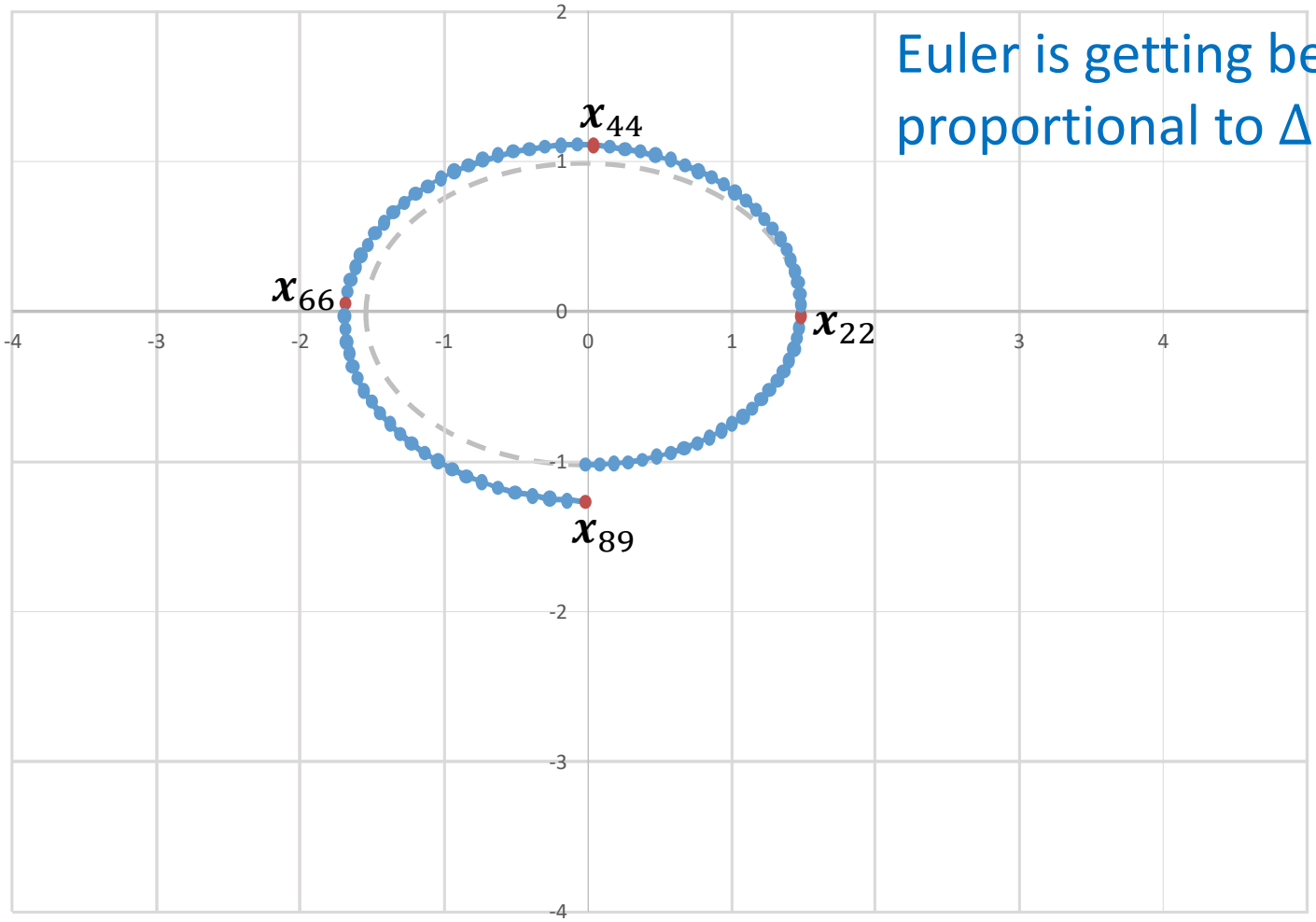
- Euler Integration after 36 steps ($\Delta t = 0.25$)



Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Euler Integration after 89 steps ($\Delta t = 0.1$)

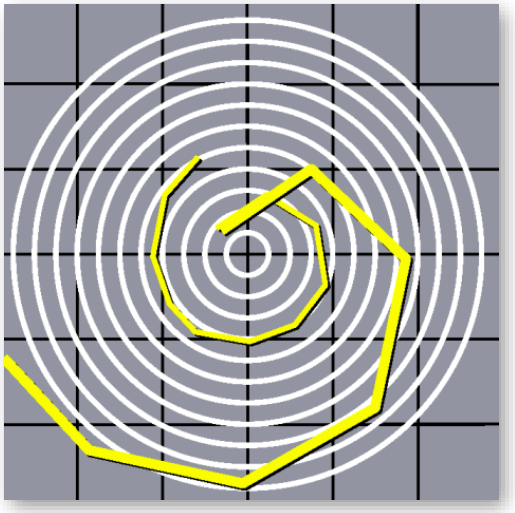


Euler is getting better proportional to Δt

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

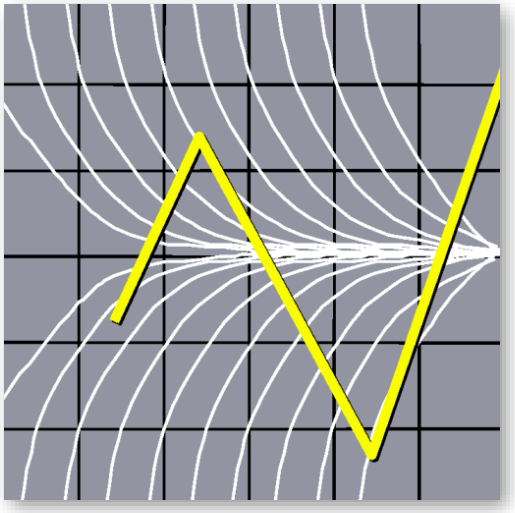
- Problem of Euler method
  - Inaccurate
  - Unstable



- Example:

$$\dot{\boldsymbol{x}}(t) = -k\boldsymbol{x}(t)$$
$$\boldsymbol{x}(t) = e^{-kt}$$

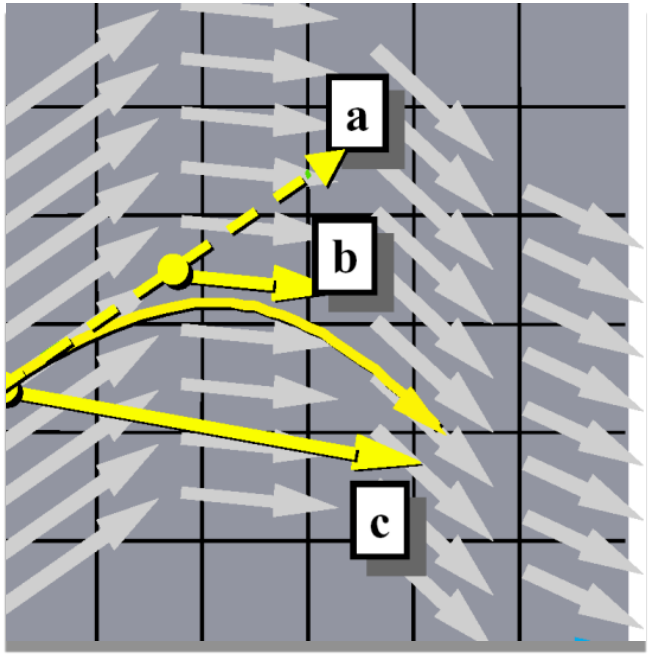
divergence for $\Delta t > 2/k$



Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Midpoint method (Runge-Kutta 2nd order)
  a. Euler step
  $$\Delta \boldsymbol{x} = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}, t)$$
  b. Evaluation of $\boldsymbol{v}$ at midpoint
  $$\boldsymbol{v}_{\text{mid}} = \boldsymbol{v}(\boldsymbol{x} + \frac{\Delta \boldsymbol{x}}{2}, t + \frac{\Delta t}{2})$$
  c. Complete step with vector at midpoint
  $$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + \Delta t \cdot \boldsymbol{v}_{\text{mid}}$$

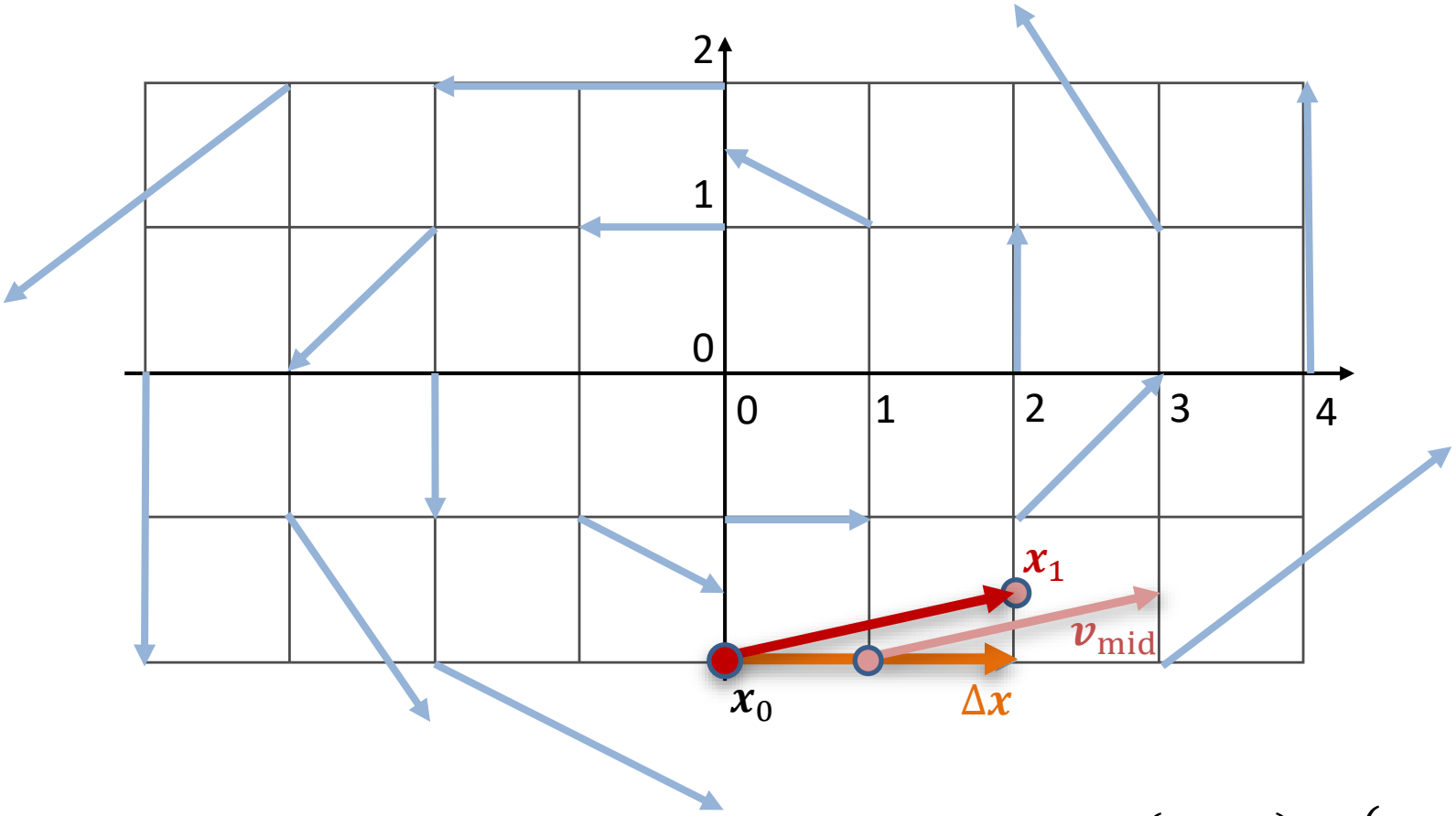


More accurate results with midpoint method

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Example: Midpoint method ($\Delta t = 1$)
  - Seed point: $\boldsymbol{x}_0 = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$, $\Delta \boldsymbol{x} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$, $\boldsymbol{v}_{\text{mid}} = \boldsymbol{v}\left(\boldsymbol{x}_0 + \frac{\Delta \boldsymbol{x}}{2}, t_0 + \frac{\Delta t}{2}\right) = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}$

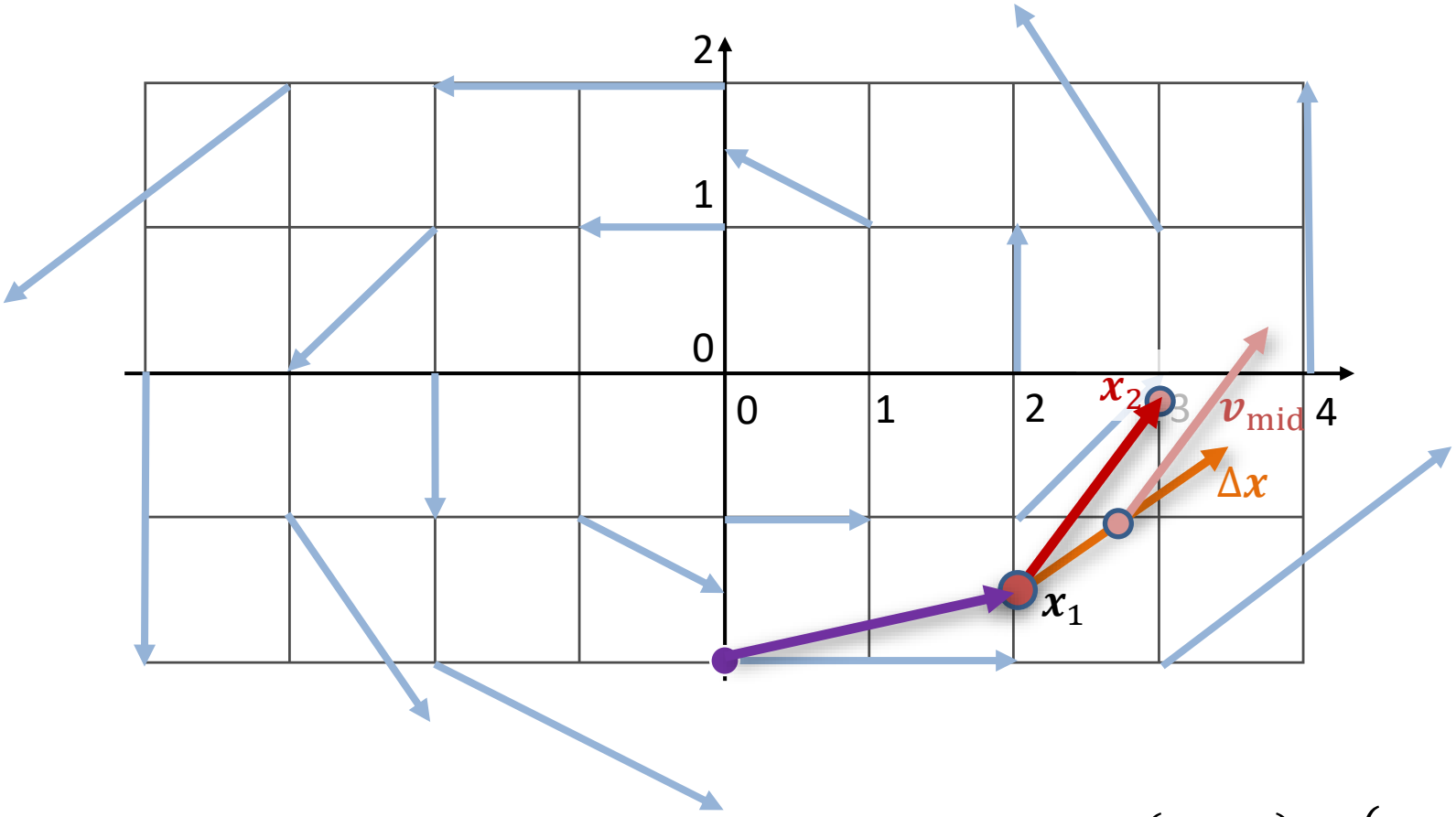


$$\boldsymbol{v}(x, y, t) = \left(-y, {}^{x}/_{2}\right)$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Example: Midpoint method ($\Delta t = 1$)
  - New point: $\boldsymbol{x}_1 = \begin{pmatrix} 2 \\ -1.5 \end{pmatrix}$, $\Delta \boldsymbol{x} = \begin{pmatrix} 1.5 \\ 1 \end{pmatrix}$, $\boldsymbol{v}_{\text{mid}} = \boldsymbol{v}\left(\boldsymbol{x}_1 + \frac{\Delta \boldsymbol{x}}{2}, t_1 + \frac{\Delta t}{2}\right) \approx \begin{pmatrix} 1 \\ 1.4 \end{pmatrix}$

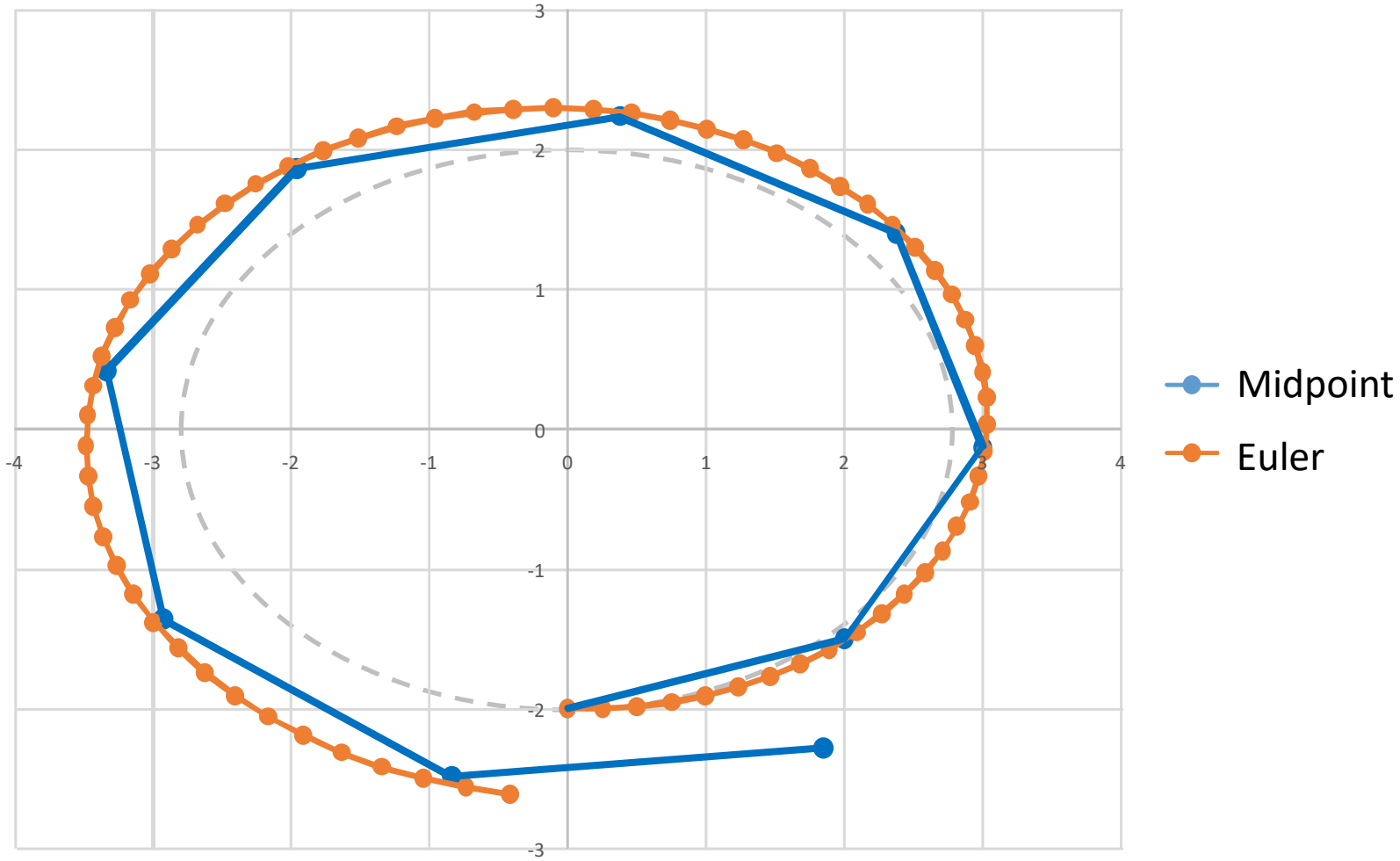


$$\boldsymbol{v}(x, y, t) = \left(-y, {}^{x}/_{2}\right)$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Comparison: Midpoint method even with $\Delta t = 1$ (9 steps)
  better than Euler with $\Delta t = 1/8$ (71 steps)



Numerical Integration of ODEs

SIEMENS
Ingenuity for life

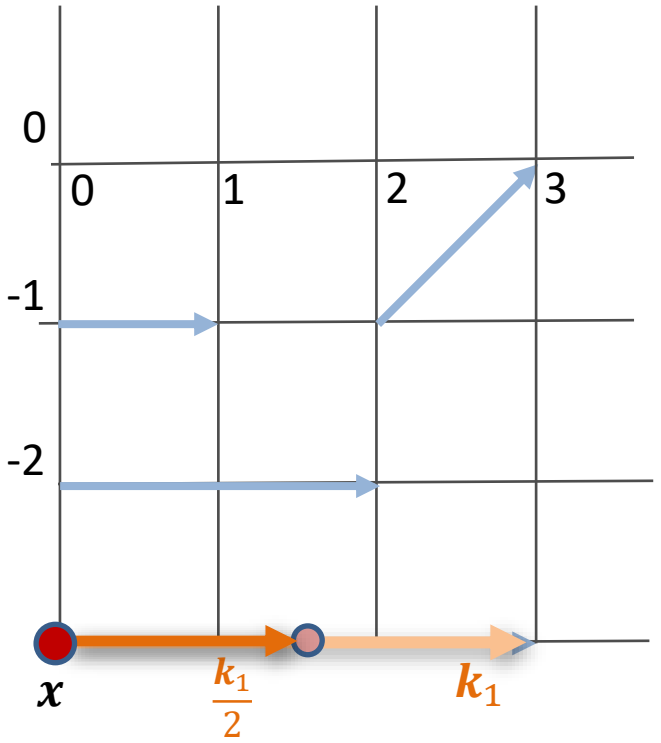
- Runge-Kutta of 4th order ($\Delta t = 1$)

$$\boldsymbol{k_1} = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}, t)$$

$$\boldsymbol{k_2} = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k_1}}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k_3} = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k_2}}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k_4} = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x} + \boldsymbol{k_3}, t + \Delta t)$$



$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x} + \frac{\boldsymbol{k_1}}{6} + \frac{\boldsymbol{k_2}}{3} + \frac{\boldsymbol{k_3}}{3} + \frac{\boldsymbol{k_4}}{6}$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

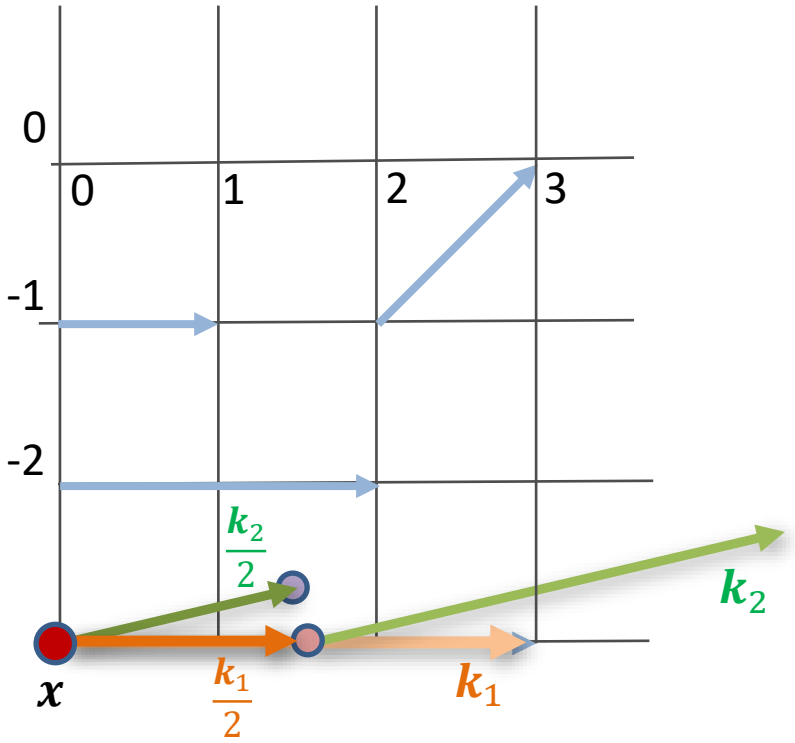
- Runge-Kutta of 4th order ($\Delta t = 1$)

$$\boldsymbol{k}_1 = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}, t)$$

$$\boldsymbol{k}_2 = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k}_1}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k}_3 = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k}_2}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k}_4 = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x} + \boldsymbol{k}_3, t + \Delta t)$$



$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x} + \frac{\boldsymbol{k}_1}{6} + \frac{\boldsymbol{k}_2}{3} + \frac{\boldsymbol{k}_3}{3} + \frac{\boldsymbol{k}_4}{6}$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

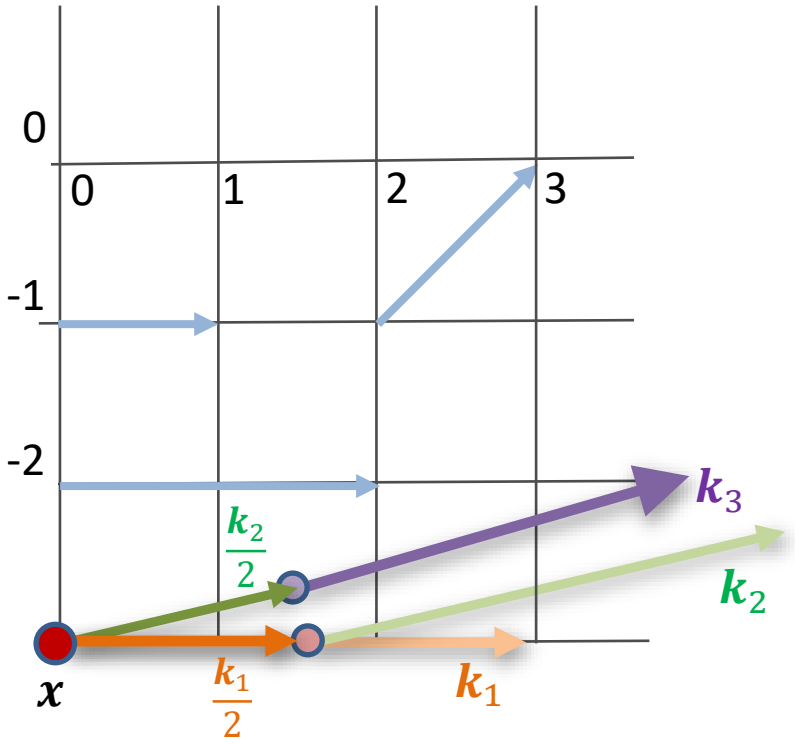
- Runge-Kutta of 4th order ($\Delta t = 1$)

$$\boldsymbol{k}_1 = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}, t)$$

$$\boldsymbol{k}_2 = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k}_1}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k}_3 = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k}_2}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k}_4 = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x} + \boldsymbol{k}_3, t + \Delta t)$$



$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x} + \frac{\boldsymbol{k}_1}{6} + \frac{\boldsymbol{k}_2}{3} + \frac{\boldsymbol{k}_3}{3} + \frac{\boldsymbol{k}_4}{6}$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

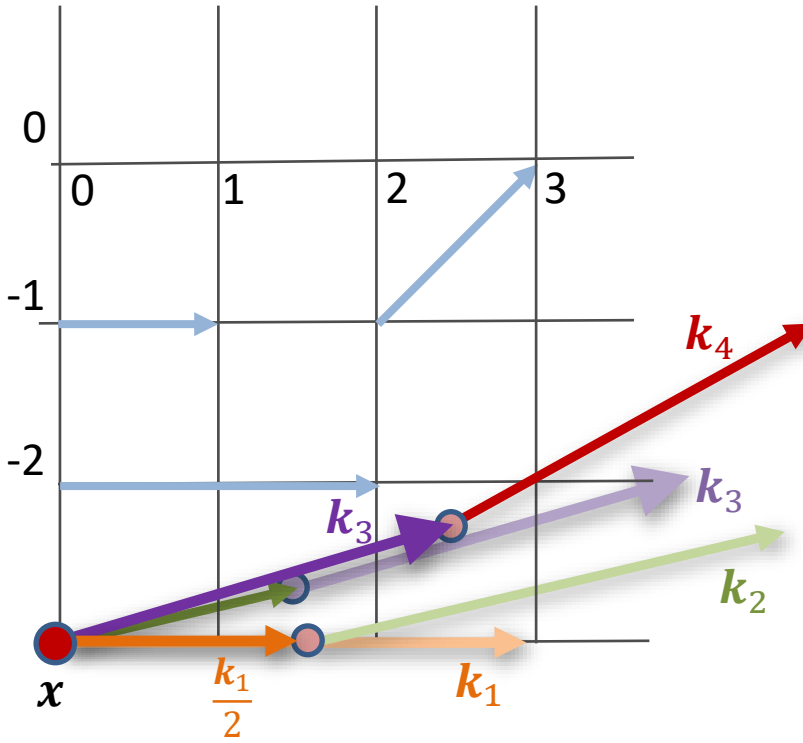
- Runge-Kutta of 4th order ($\Delta t = 1$)

$$\boldsymbol{k}_1 = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}, t)$$

$$\boldsymbol{k}_2 = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k}_1}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k}_3 = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k}_2}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k}_4 = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x} + \boldsymbol{k}_3, t + \Delta t)$$



$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x} + \frac{\boldsymbol{k}_1}{6} + \frac{\boldsymbol{k}_2}{3} + \frac{\boldsymbol{k}_3}{3} + \frac{\boldsymbol{k}_4}{6}$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

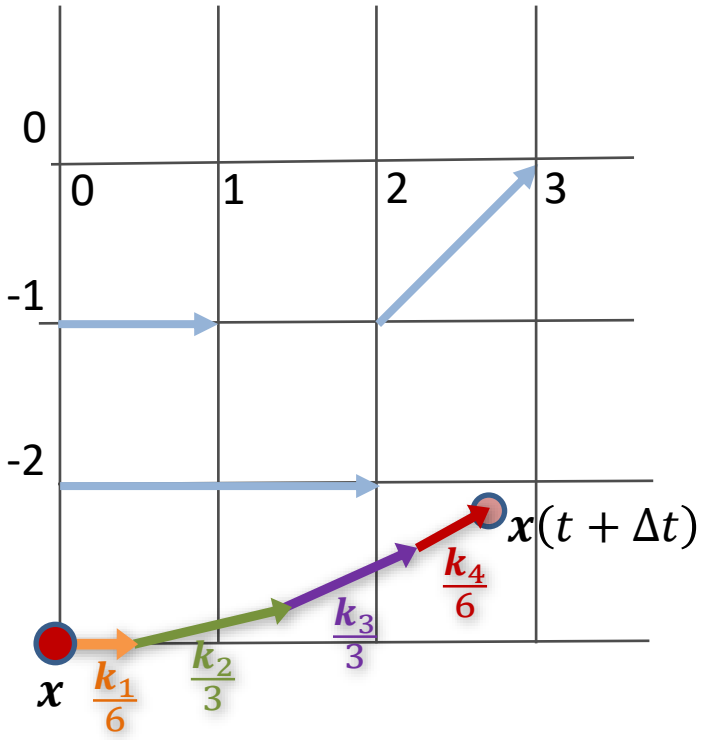
- Runge-Kutta of 4th order ($\Delta t = 1$)

$$\boldsymbol{k_1} = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x}, t)$$

$$\boldsymbol{k_2} = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k_1}}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k_3} = \Delta t \cdot \boldsymbol{v}\left(\boldsymbol{x} + \frac{\boldsymbol{k_2}}{2}, t + \frac{\Delta t}{2}\right)$$

$$\boldsymbol{k_4} = \Delta t \cdot \boldsymbol{v}(\boldsymbol{x} + \boldsymbol{k_3}, t + \Delta t)$$

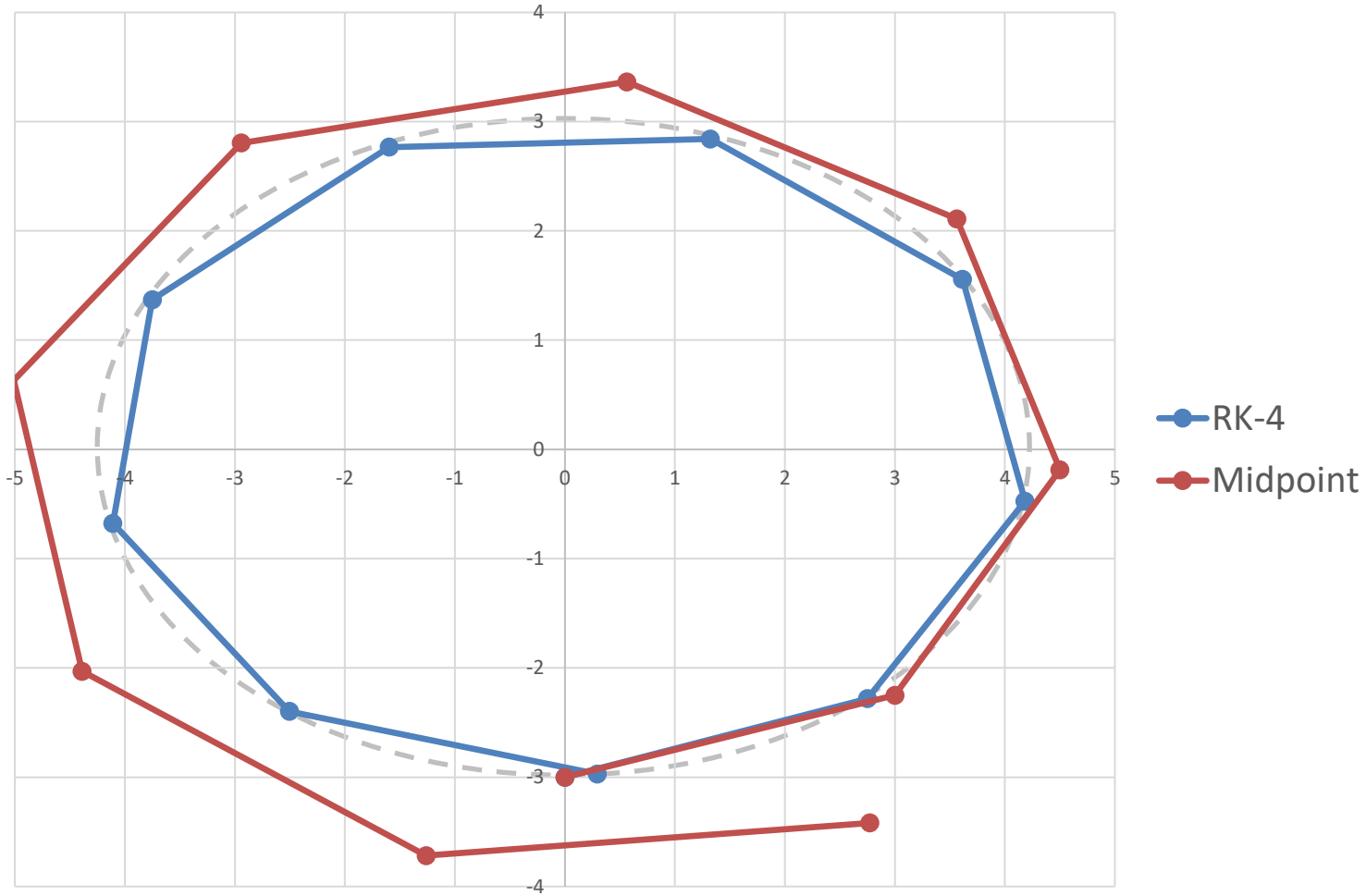


$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x} + \frac{\boldsymbol{k_1}}{6} + \frac{\boldsymbol{k_2}}{3} + \frac{\boldsymbol{k_3}}{3} + \frac{\boldsymbol{k_4}}{6}$$

Numerical Integration of ODEs

SIEMENS
Ingenuity for life

- Comparison: Runge-Kutta 4th order vs. midpoint method
  after 9 steps ($\Delta t = 1$)



Numerical Integration of ODEs

SIEMENS
Ingenuity for life

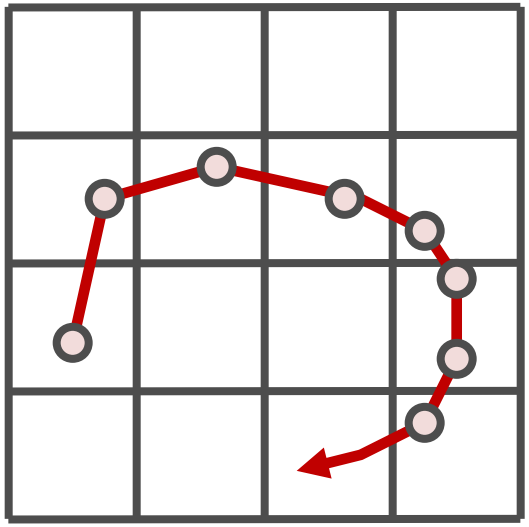
- Summary
  - Analytic determination of streamlines usually not possible
  - Hence, numerical integration
  - Several methods available
    - Euler: simple, imprecise, especially with larger Δt
    - Runge-Kutta: more accurate in higher orders,
      pays off with complex flows

Particle Tracing on Grids

SIEMENS
Ingenuity for life

- Vector field given on a grid
  - Solve $\boldsymbol{L}(0) = \boldsymbol{x}_0, \quad \frac{d\boldsymbol{L}(u)}{du} = \boldsymbol{v}(\boldsymbol{L}(u), t)$

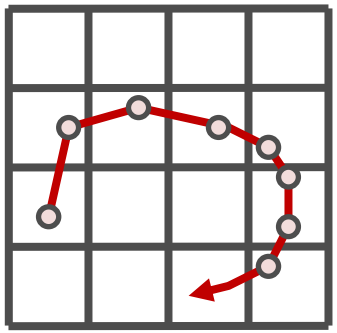
    for the path line
  - Incremental integration
  - Discretized path of the particle



Particle Tracing on Grids

SIEMENS
Ingenuity for life

- Most simple case: Cartesian grid
- Basic algorithm:



```
Select start point (seed point)
Find cell that contains start point  // point location
While (particle in domain) do
    Interpolate vector field at      // interpolation
      current position
    Integrate to new position        // integration
    Find new cell                    // point location
    Draw line segment between latest
      particle positions
Endwhile
```

Particle Tracing on Grids

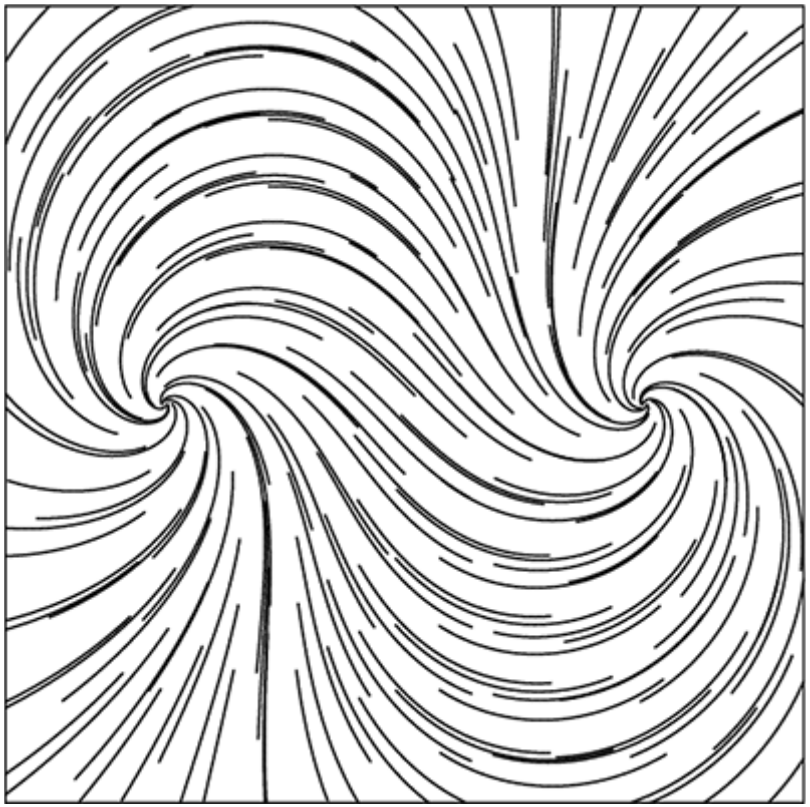
SIEMENS
Ingenuity for life

- Point location (cell search) on Cartesian grids
  - Indices of cell directly from position (x, y, z)
  - For example: $i_x = \lfloor (x - x_0) / \Delta x \rfloor$
  - Simple and fast
- Interpolation on Cartesian grids
  - Bilinear (in 2D) or trilinear (in 3D) interpolation
  - Required to compute the vector field (= velocity) inside a cell
  - Component-wise interpolation

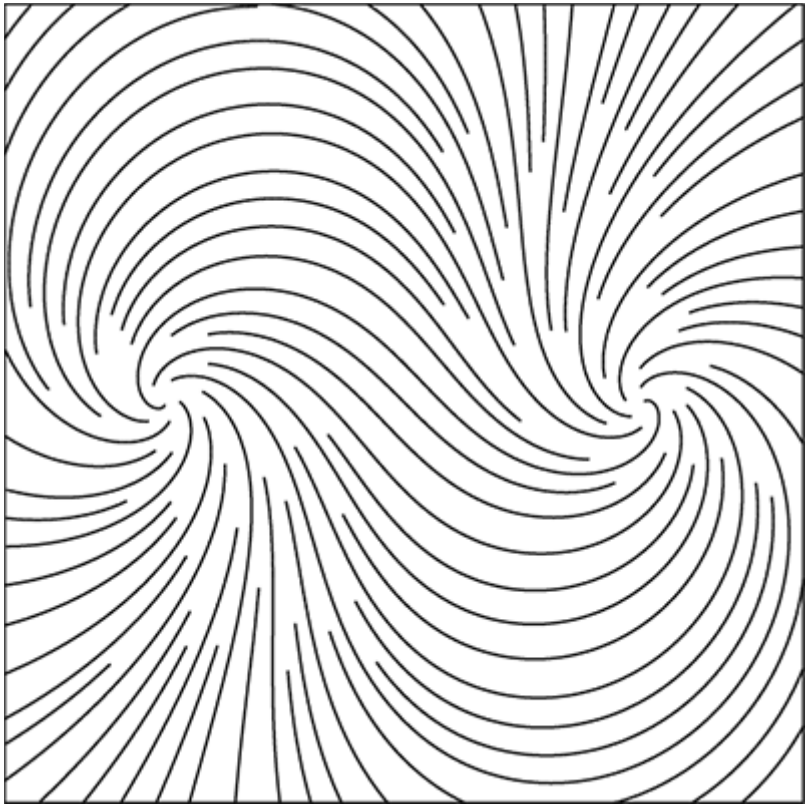
Stream line placement in 2D

SIEMENS
Ingenuity for life

- Stream line placement
  - Irregular results when using regular grid



Stream line placement from regular grid



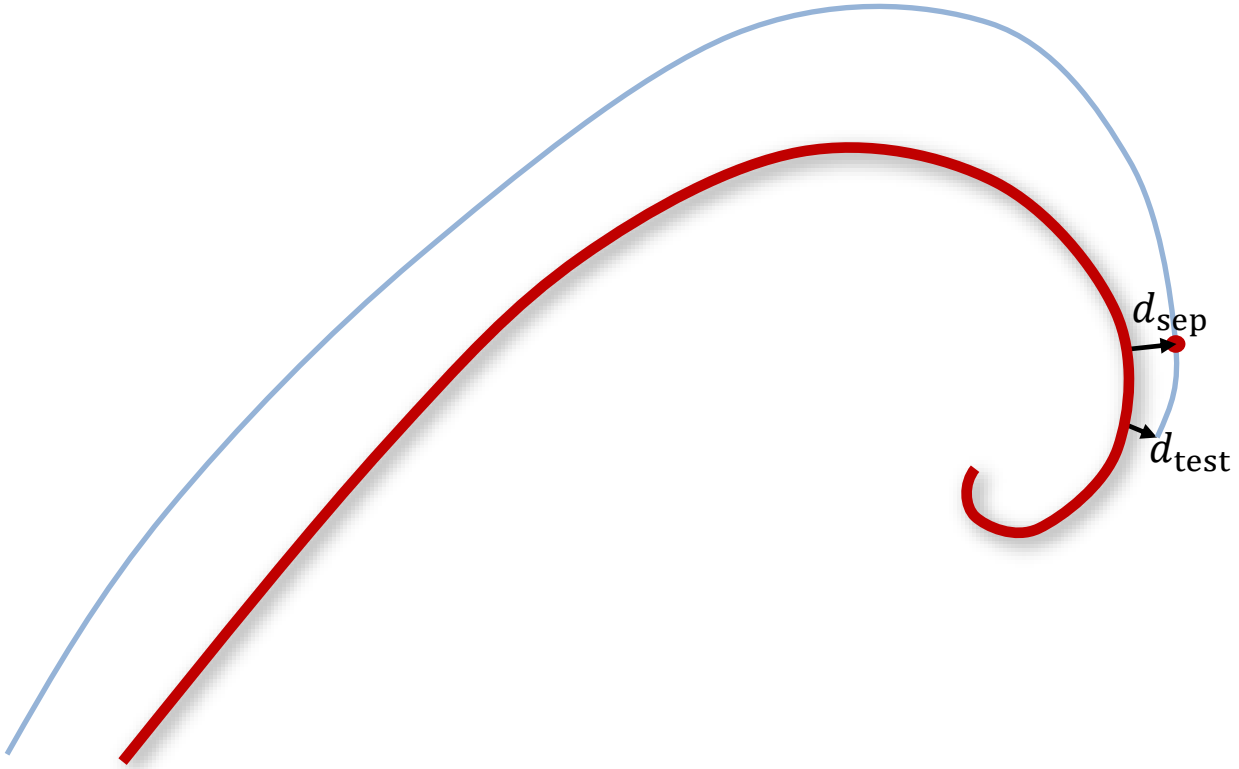
Evenly-spaced stream lines

[Jobard & Lefer 97]

Stream line placement in 2D

SIEMENS
Ingenuity for life

- Evenly-spaced streamlines
- Idea: stream lines should not get too close to each other
  - Choose seed point with distance d_{sep} from existing stream line
  - Forward- and backward-integration until distance d_{test}

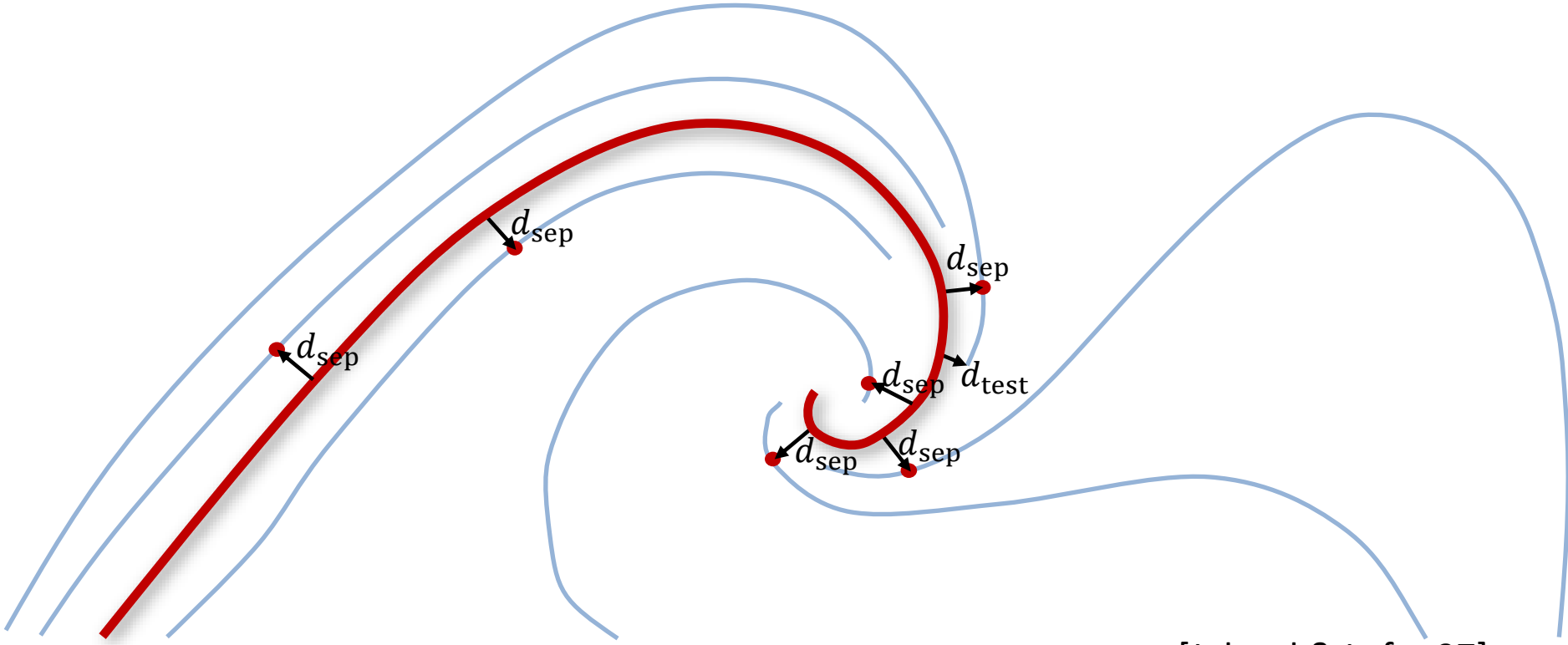


[Jobard & Lefer 97]

Stream line placement in 2D

SIEMENS
Ingenuity for life

- Evenly-spaced streamlines
- Idea: stream lines should not get too close to each other
  - Choose seed point with distance d_{sep} from existing stream line
  - Forward- and backward-integration until distance d_{test}



[Jobard & Lefer 97]

Stream line placement in 2D

SIEMENS
Ingenuity for life

Algorithm:

Compute initial stream line and put into the queue

Initial stream line becomes current stream line

While not finished do:

- Try: get new seed point with distance d_{sep} from current stream line
- If successful then compute new stream line and put into queue
- Else If no more stream lines in queue then exit loop
- Else next stream line in queue becomes current streamline

EndWhile

[Jobard & Lefer 97]

Stream line placement in 2D

SIEMENS
Ingenuity for life

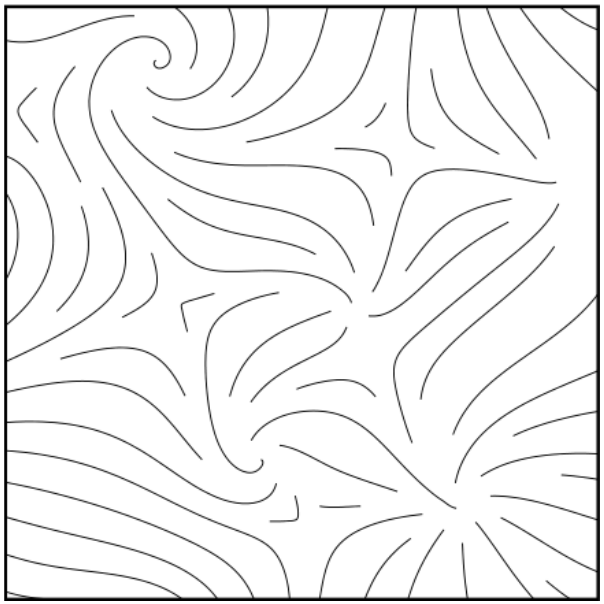
- When to stop stream line integration
  - When distance to neighboring stream line $\leq d_{\text{test}}$
  - When stream line leaves flow domain
  - When stream line runs into fixed point $(\boldsymbol{v}(\boldsymbol{x}^*) = 0)$
  - When stream line gets too close to itself
  - After a certain number of maximal steps

[Jobard & Lefer 97]

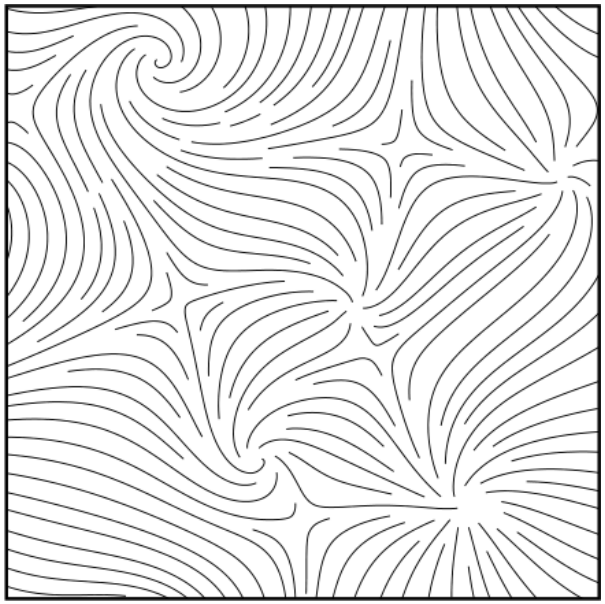
Stream line placement in 2D

SIEMENS
Ingenuity for life

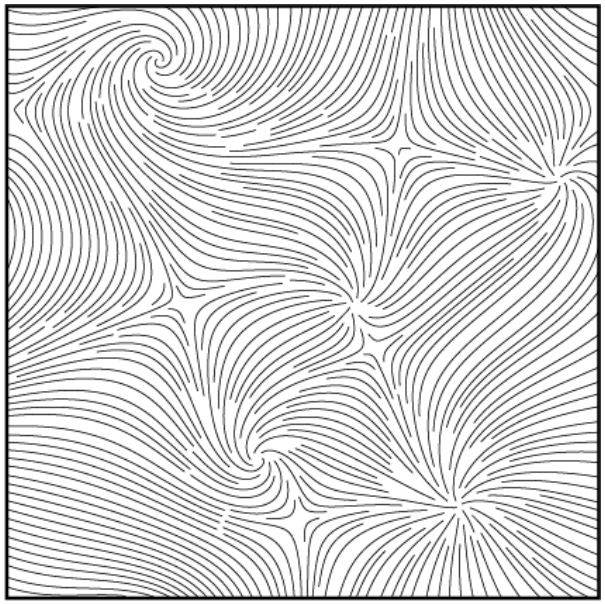
- Variations of d_{sep} in relation to image width



6%



3%



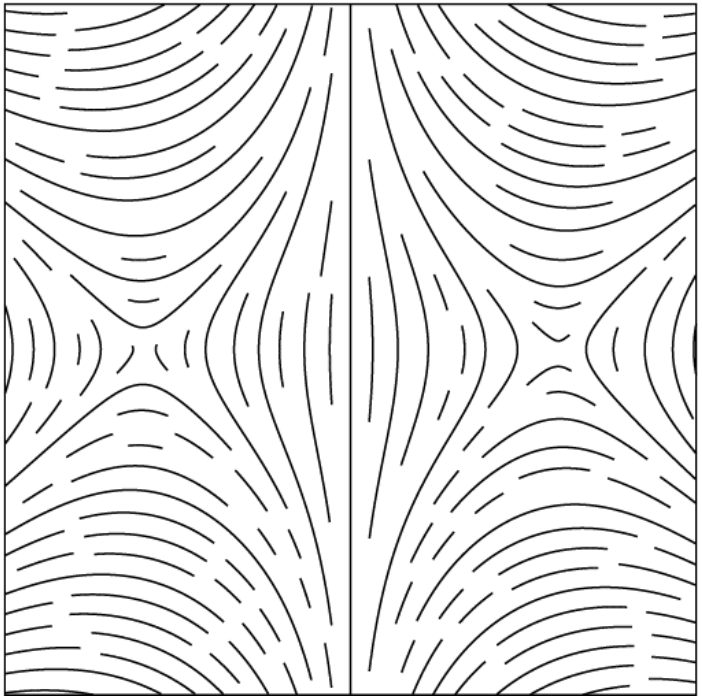
1.5%

[Jobard & Lefer 97]

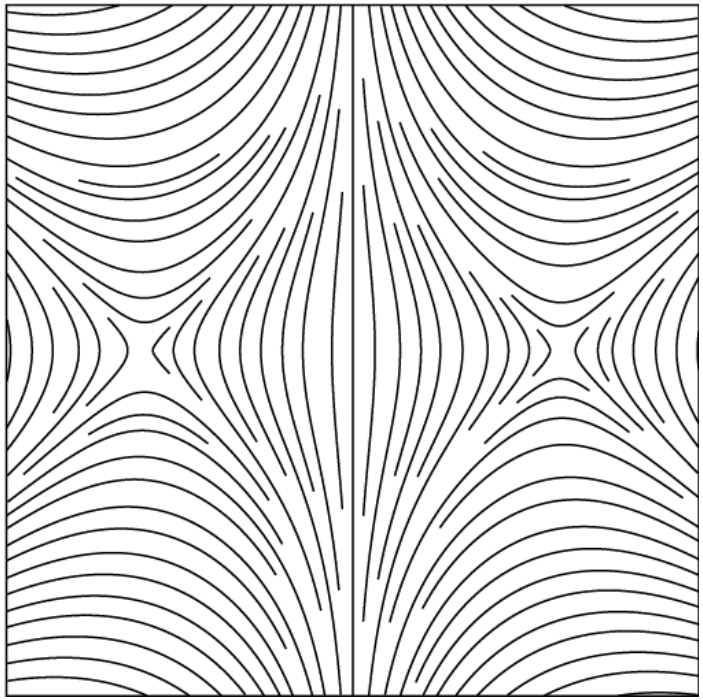
Stream line placement in 2D

SIEMENS
Ingenuity for life

- Variations of d_{test}



$d_{\text{test}} = 0.9\ d_{\text{sep}}$



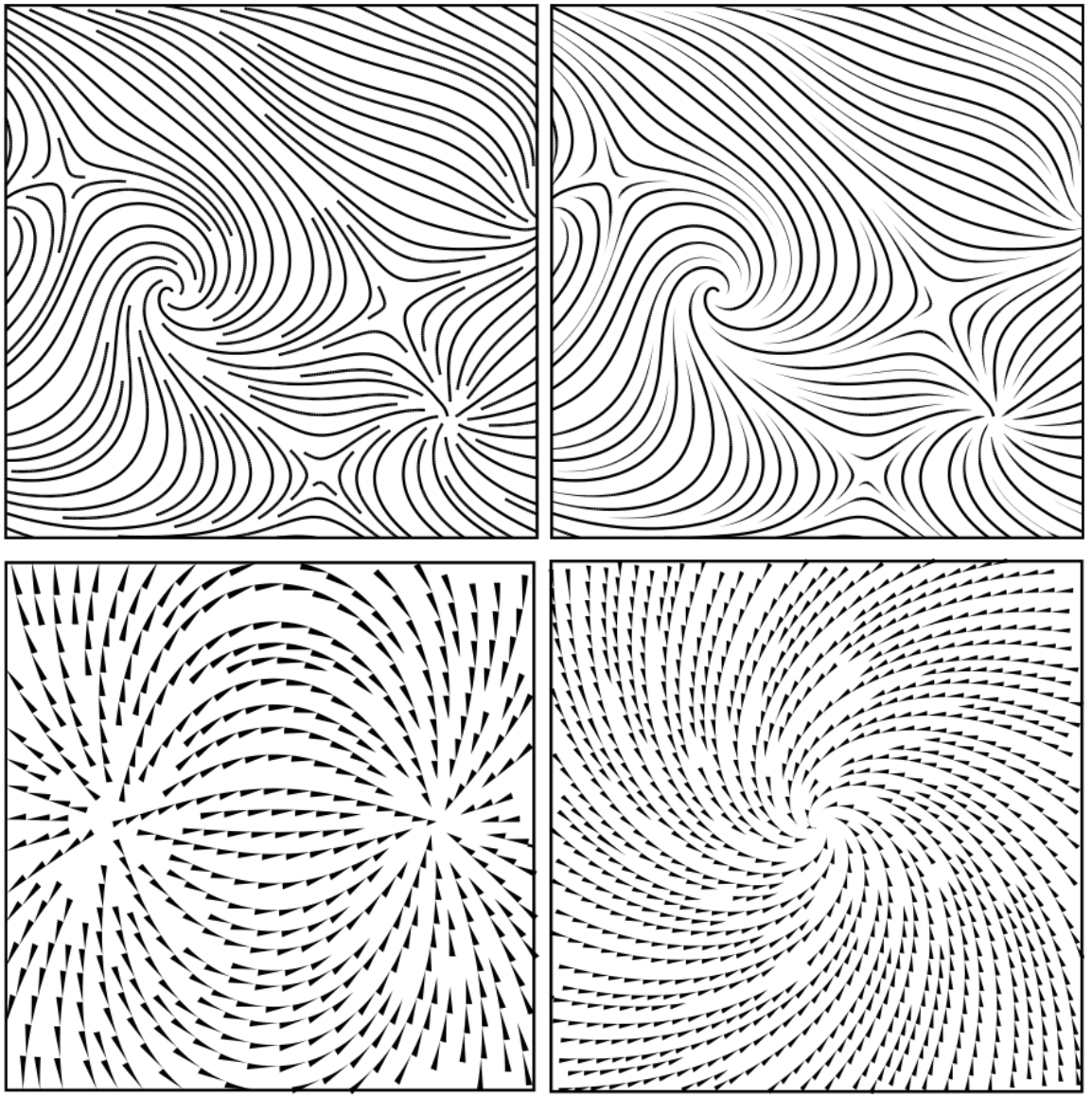
$d_{\text{test}} = 0.5\ d_{\text{sep}}$

[Jobard & Lefer 97]

Stream line placement in 2D

SIEMENS
Ingenuity for life

- Change thickness in relation to distance d
  - If $d \geq d_{\text{sep}}$: 1.0
  - If $d < d_{\text{sep}}$: $\frac{d - d_{\text{test}}}{d_{\text{sep}} - d_{\text{test}}}$
- Directional glyphs



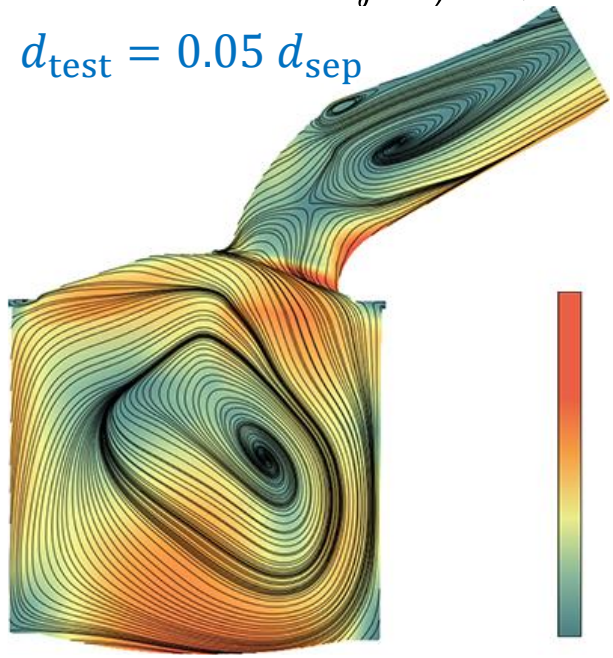
[Jobard & Lefer 97]

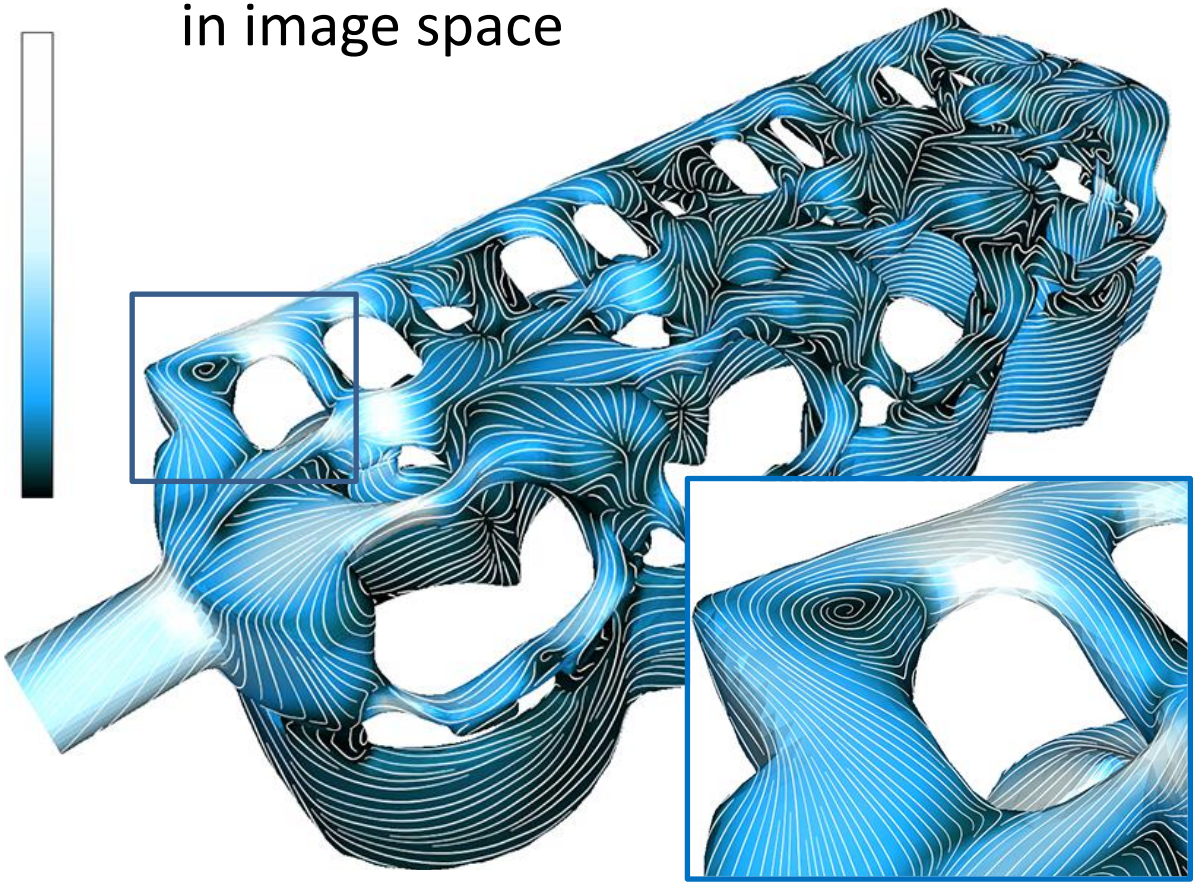
Stream line placement on surfaces

SIEMENS
Ingenuity for life

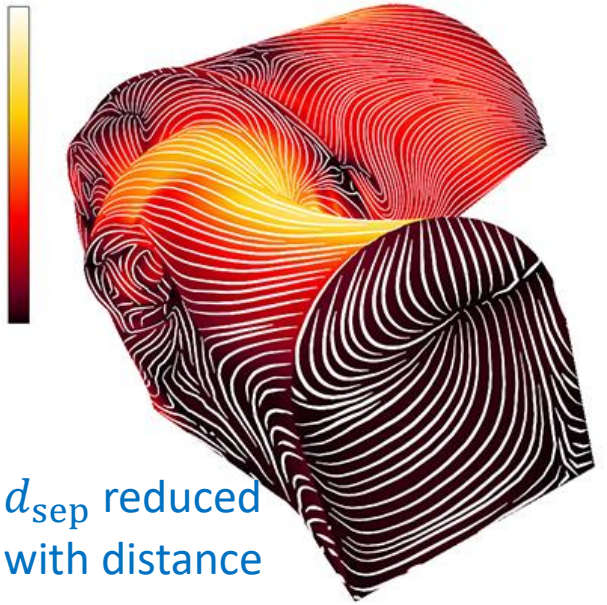
- Image-space technique
  - Vector field is first projected to 2D image
  - Seeding and integration happen in image space

$d_{\text{test}} = 0.05\, d_{\text{sep}}$





[Spencer et al. 09]



d_{sep} reduced with distance

Stream line placement on surfaces

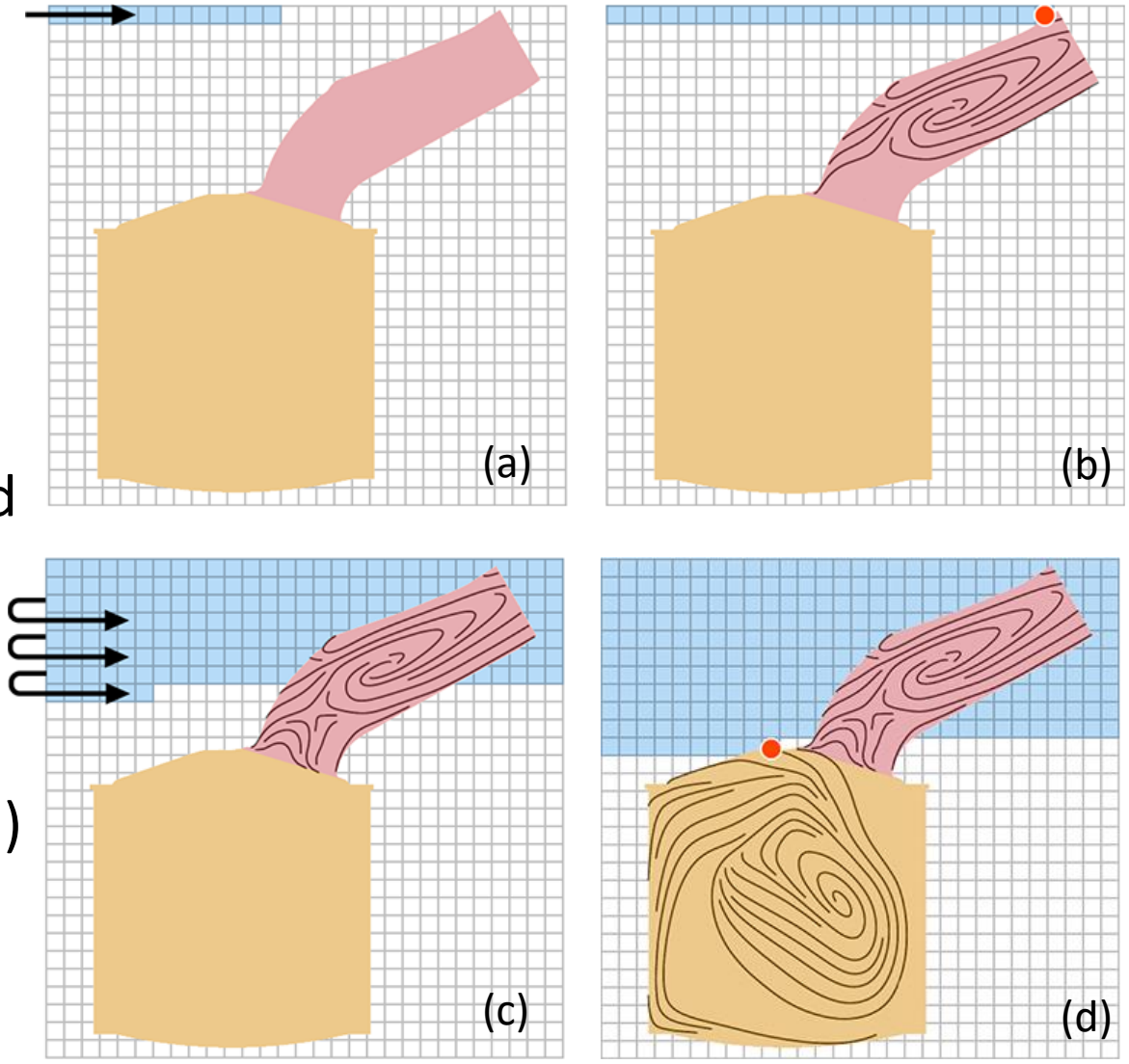
SIEMENS
Ingenuity for life

- Vector field is first projected to 2D image

a) 2D image is scanned at intervals d_{sep}

b) Seedpoint is found & stream lines are traced in that region

c) Scanning continues until seedpoint in new region is found (d)



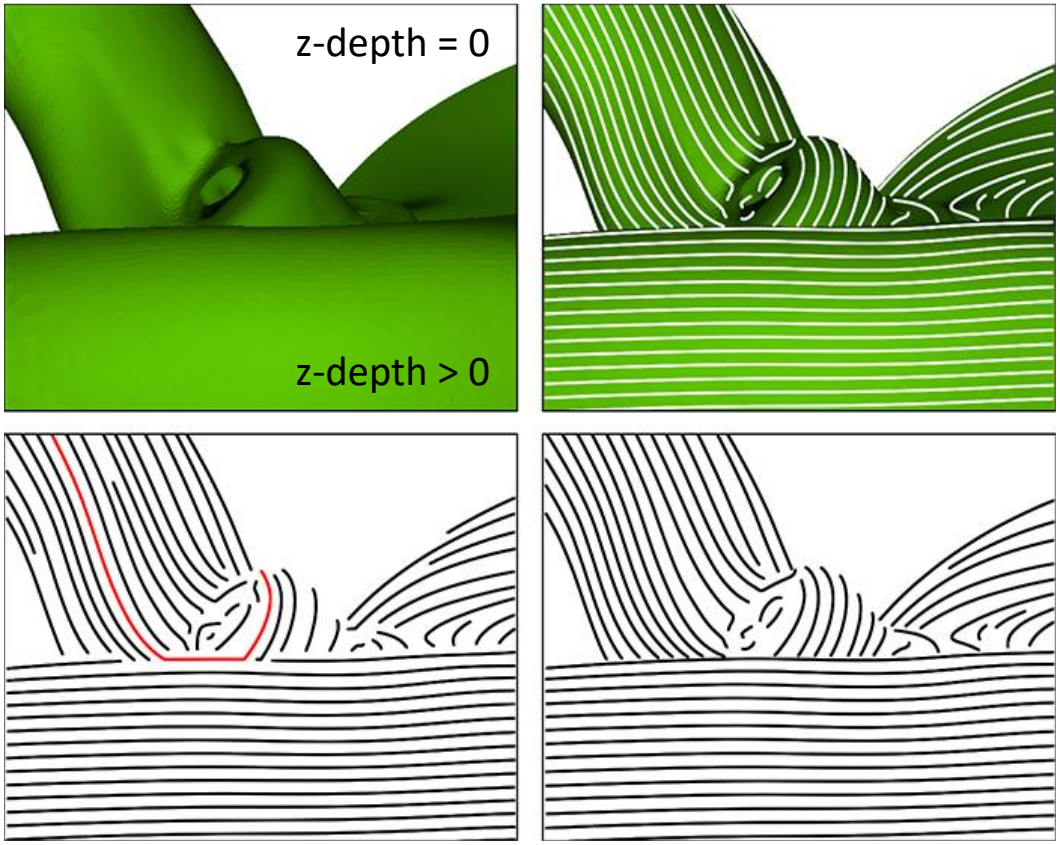
[Spencer et al. 09]

Stream line placement on surfaces

SIEMENS
Ingenuity for life

- Discontinuity detection
  - Stop stream line integration when z-depth drops to zero (edge of model)

  … or when z-depth changes too abruptly (edge of overlapping regions)



Without detection: stream lines run off edge of surface (red)

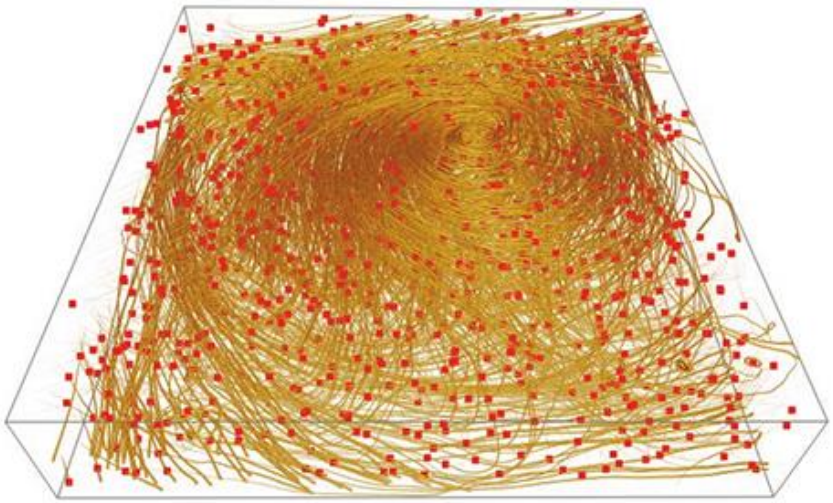
With detection: underlying surface is better reflected

[Spencer et al. 09]

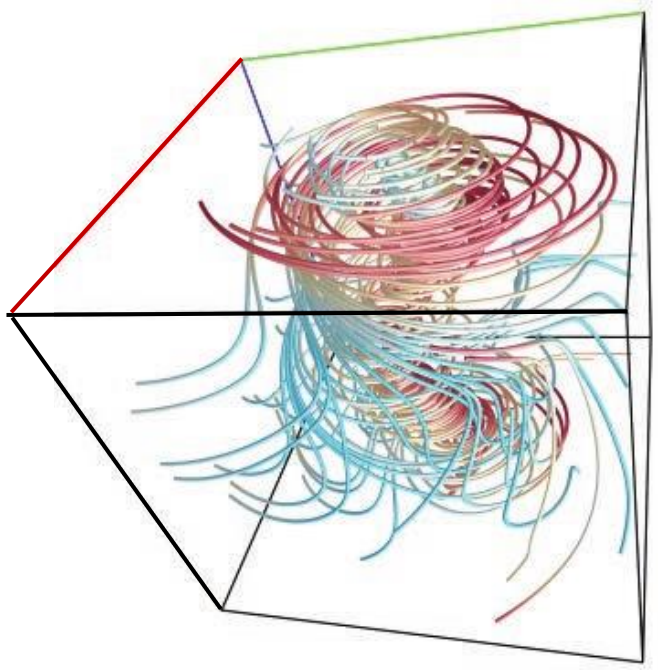
Stream line placement in 3D

SIEMENS
Ingenuity for life

- Placement of seeds directly affects the visual quality
  - Too many: scene cluttering (occlusion)
  - Too little: no patterns forming
- Seeding has to be at the right place and in the right amount!



A bad seeding example



[Tao et al. 13]

Finding best stream lines and best view point from many candidates

Stream line placement

SIEMENS
Ingenuity for life

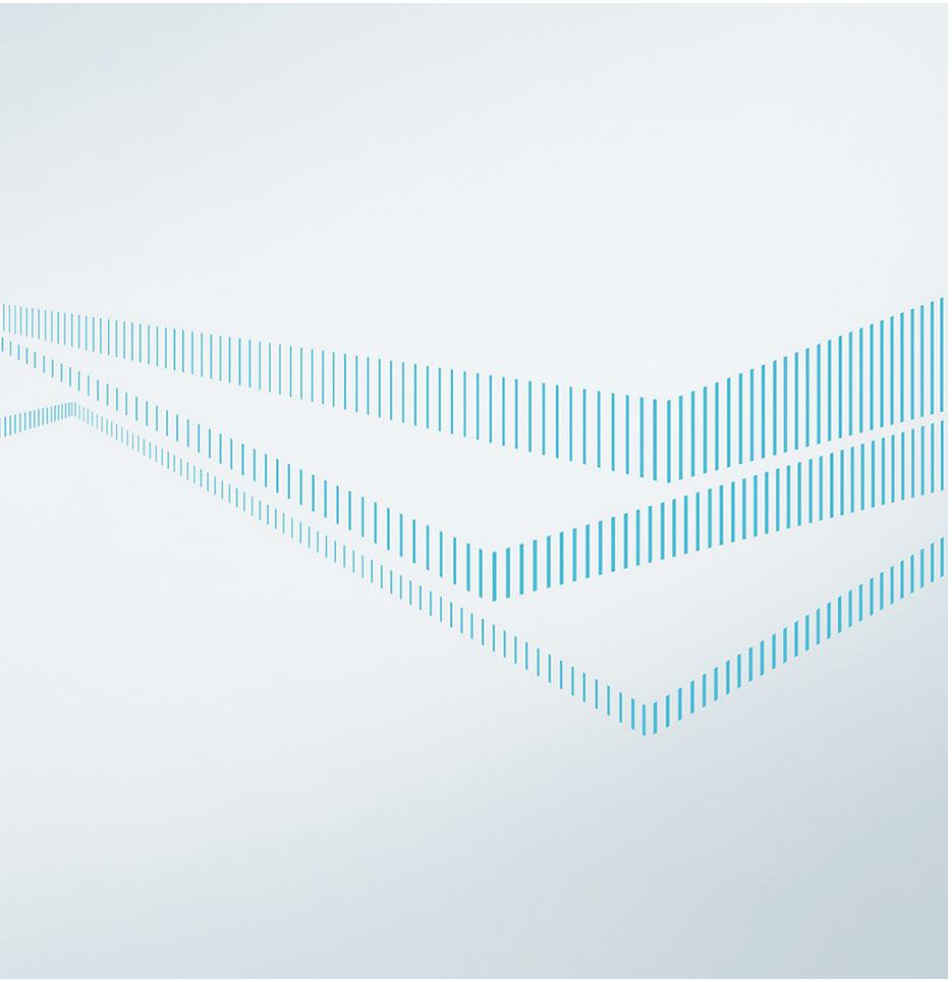
- Summary
  - Irregular results when using regular grid or random seeding
  - Stream lines should not get too close to each other
  - Seeding/placement techniques typically limed to 2D or surfaces and stationary vector fields
  - Not many methods for 3D and/or time-varying flows yet

Acknowledgements

- Helwig Hauser
- Andrea Brambilla
- Daniel Weiskopf
- Ronald Peikert
- Christoph Garth
- Alexandru C. Telea
- Many more

Contact information

SIEMENS
Ingenuity for life



Dr. Johannes Kehrer

Siemens AG
Technology
T DAI HCA-DE
Otto-Hahn-Ring 6
81739 München, Deutschland

E-mail:
kehrer.johannes@siemens.com

Internet
siemens.com/innovation

Intranet
intranet.ct.siemens.com