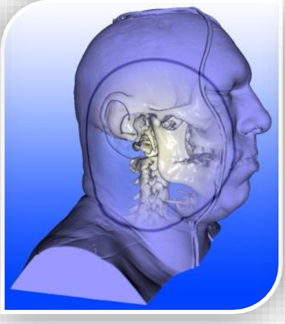
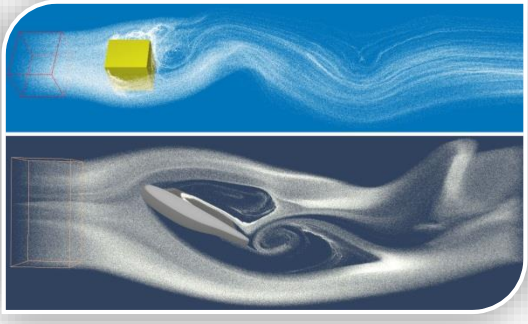
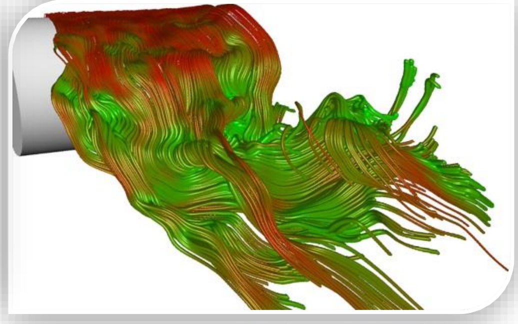
Master Practical Course

Interactive Visual Data Analysis

tum.3D

computer graphics & visualization

Schedule

- Regular meetings:
  - Jan 8 (today) – Ass. 10: Streak lines and tube rendering
  - Jan 15 – Ass. 11: Paper: Smoke surfaces
  - Jan 22 – Ass. 12: ???
  - Jan 29 – Deadline Ass. 12, no new assignment
  - Feb 5 (last week of semester) – Nothing

- Feb 4, 4pm – Demo Day
  http://www.in.tum.de/demoday

This Week

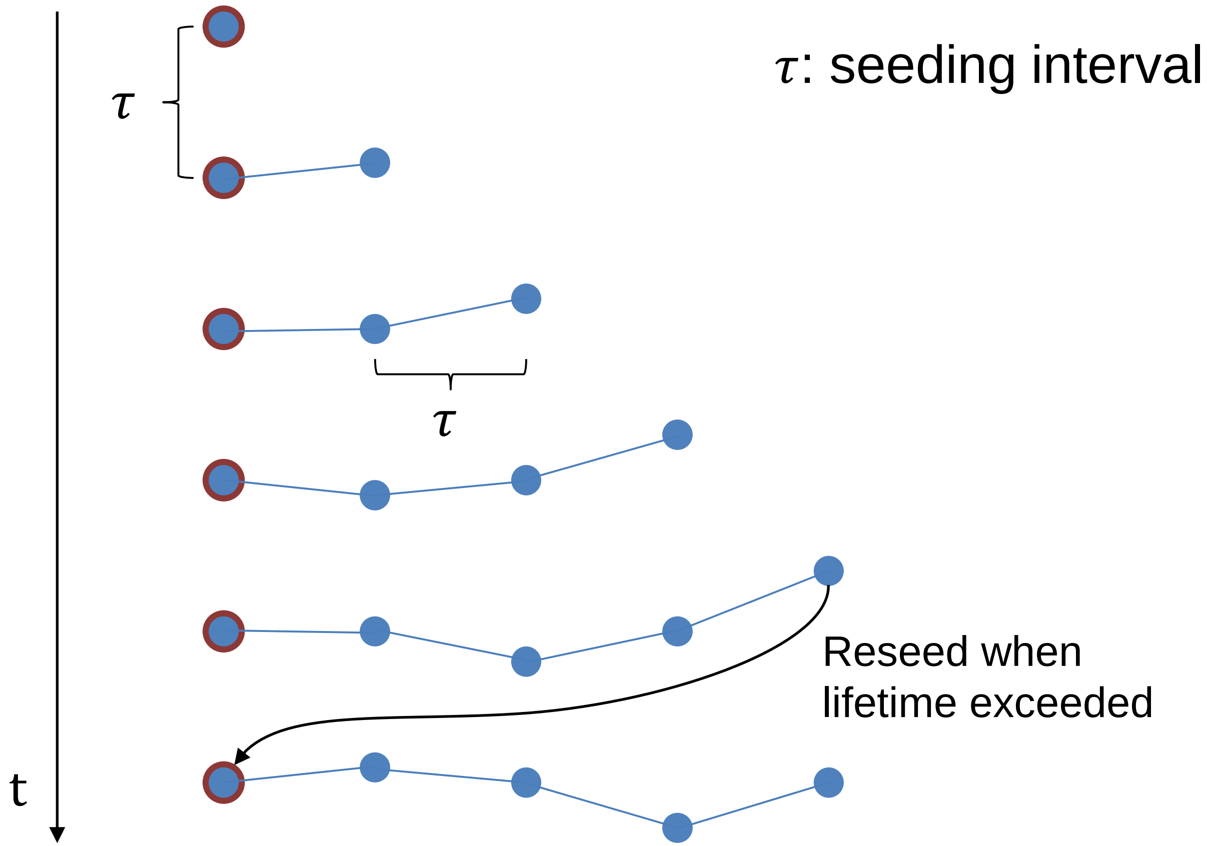
- Assignment 10:
  - Streak lines
  - Stream-/Streak-line rendering with tubes

Characteristic Lines

- Types of characteristic lines in a vector field:
  - Path lines: trajectories of massless particles in the (unsteady) flow
  - Stream lines: trajectories of massless particles in a “frozen” (steady) vector field
  - Streak lines: trace of dye that is released into the (unsteady) flow at a fixed position

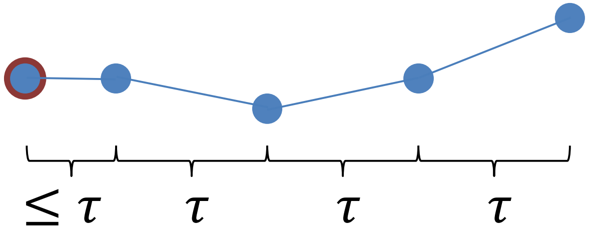
Streak Line Generation

- Repeatedly seed particles from a fixed position in space
- Render as connected line



Streak Line Seeding

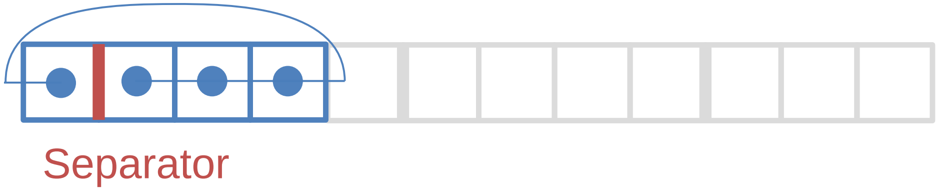
- User defined, fixed seeding interval τ
- Seeding strategy challenge: $\tau \neq elapsedTime$,
  i.e. regular seeding interval vs. irregular frame times
  - Streak line should always be „connected“ to seed point
  - First point can be less than τ away from seed point
  - May have to seed != 1 new particles per frame!



Implementation

- Create buffer for n*m particles/vertices (n lines, m vertices each)
- Streak line lifetime $T = \tau \cdot (m - 1)$
- Looped storage, separator index marks line begin/end

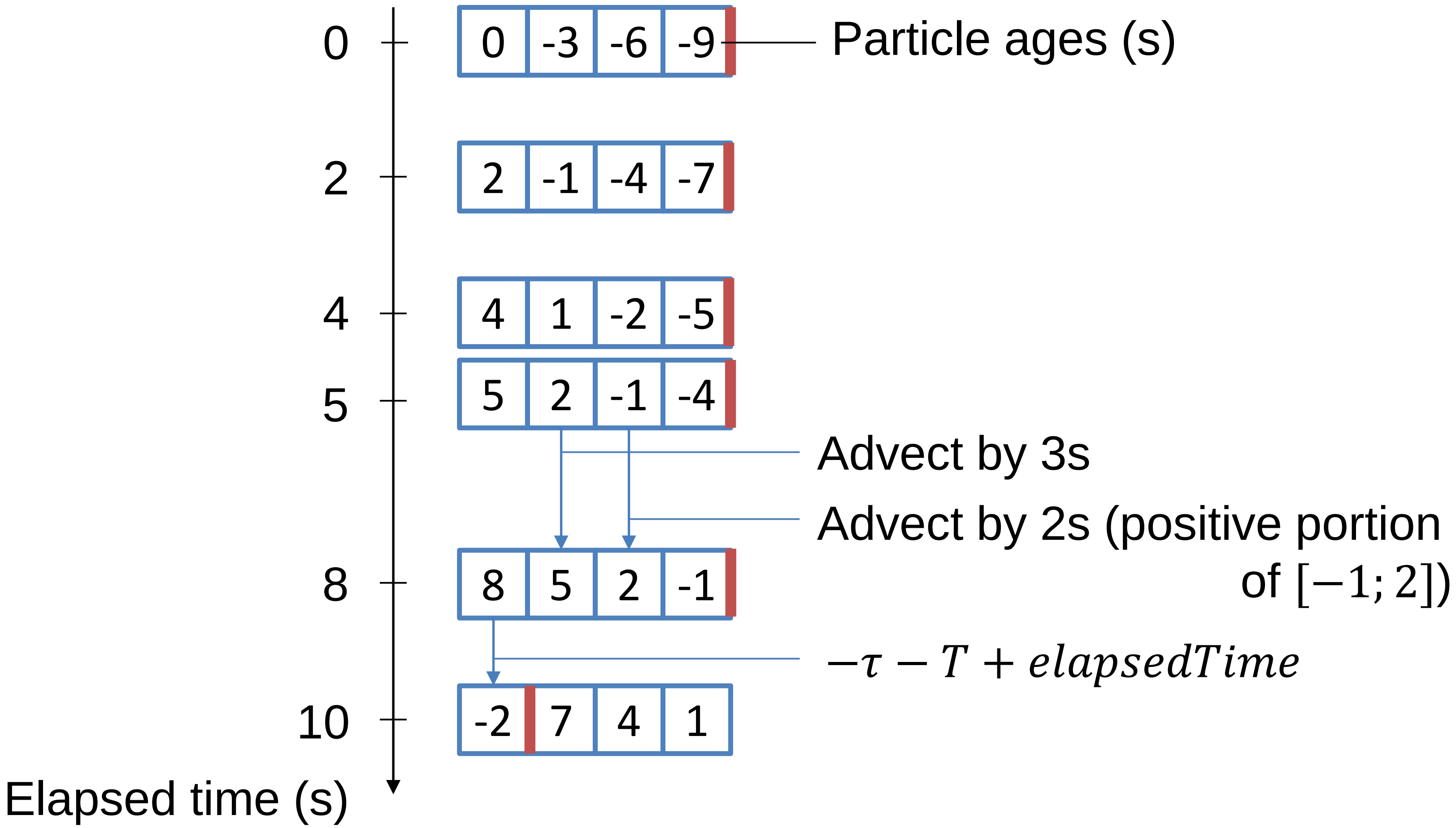
  → Moved everytime a particle is reseeded



- Particle age $\in [-T; T]$
  - $age < 0$: stationary particle at seeding point
  - Only advect the „positive portion“ of the timestep

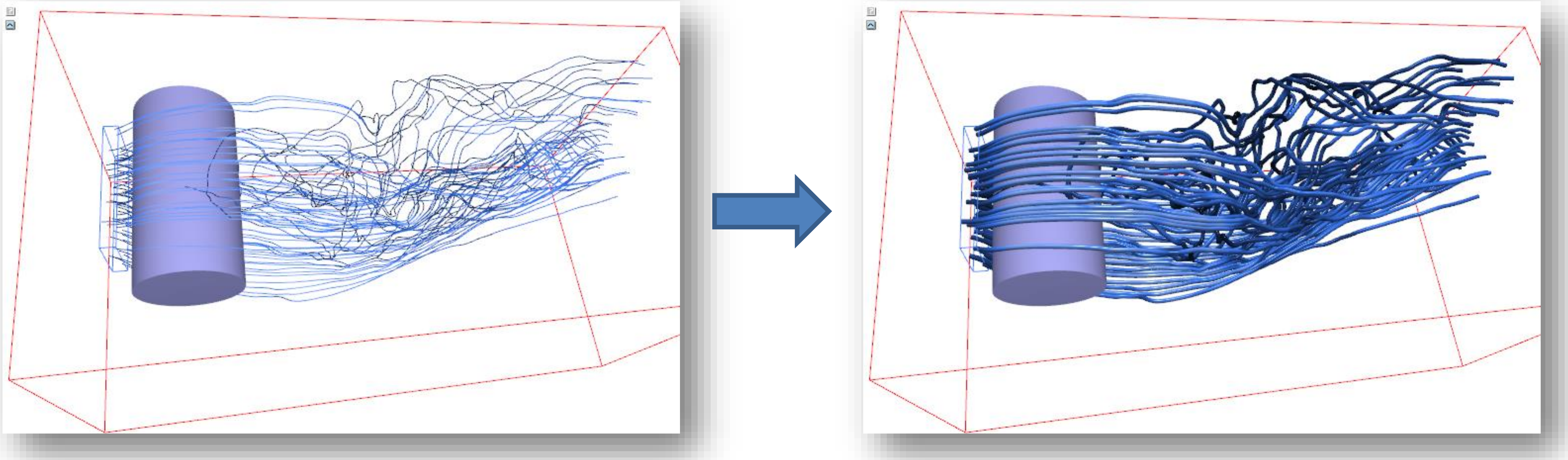
Implementation

- Example: $n = 1$ lines, $m = 4$ vertices, $\tau = 3$s, $T = 9$s



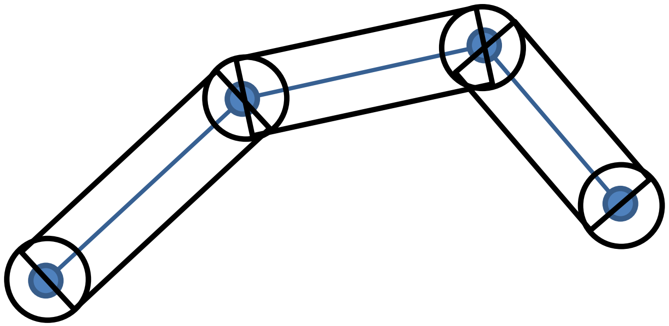
Rule: Separator is always right of the „youngest particle“

Line Rendering: Tubes



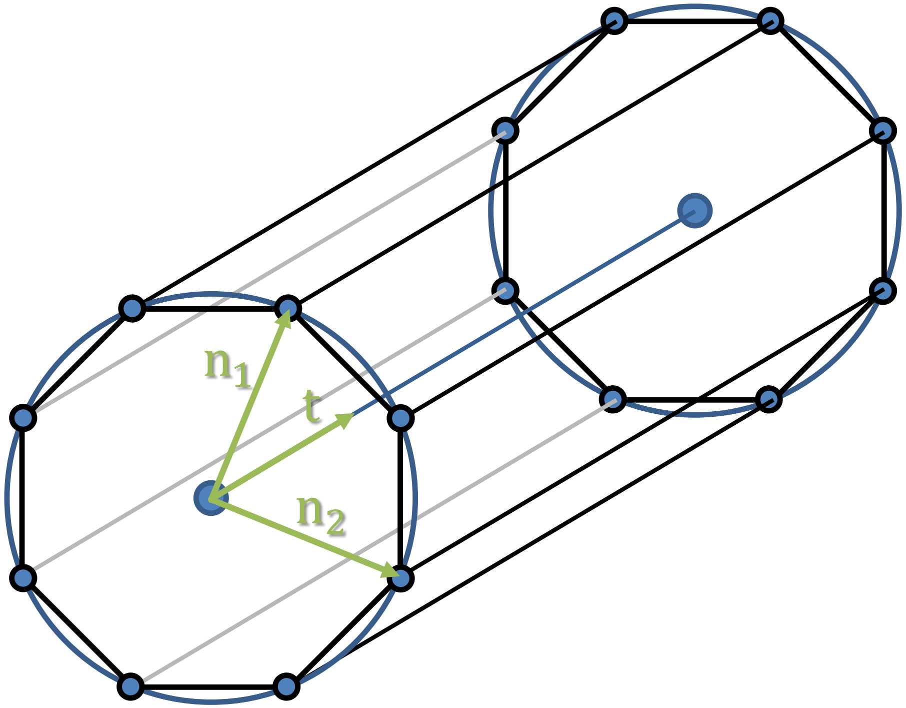
- Rendering options?
  - Analytic Cylinders
  - Extrude in Geometry Shader

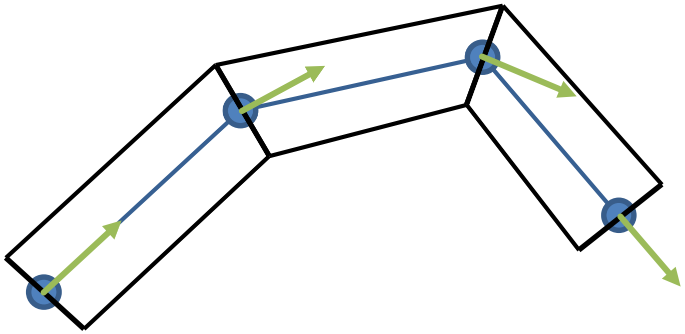
Analytic Tubes



- 1 cylinder per line segment
  - Proxy geometry: e.g. 3 front faces of oriented bounding box
- 1 sphere per vertex to close gaps
- Ray-cylinder intersection: Google ☺

GS Tubes

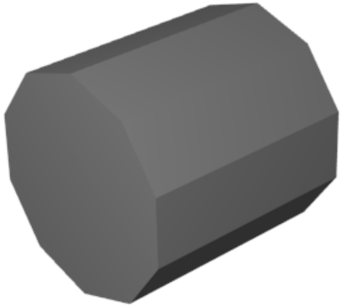




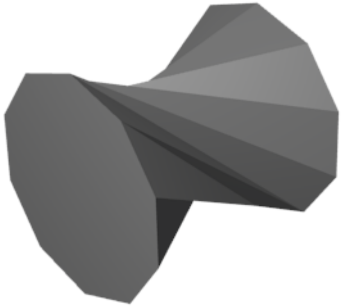
- GS (per line primitive):
  - Per vertex: generate n (e.g. 8) points on a circle
  - Connect corresponding points using quads (2 triangles)
  - Align circles with **per-vertex** tangent to avoid gaps

GS Tubes: Orientation

- Orientation doesn‘t matter
- … but needs to be consistent!



vs

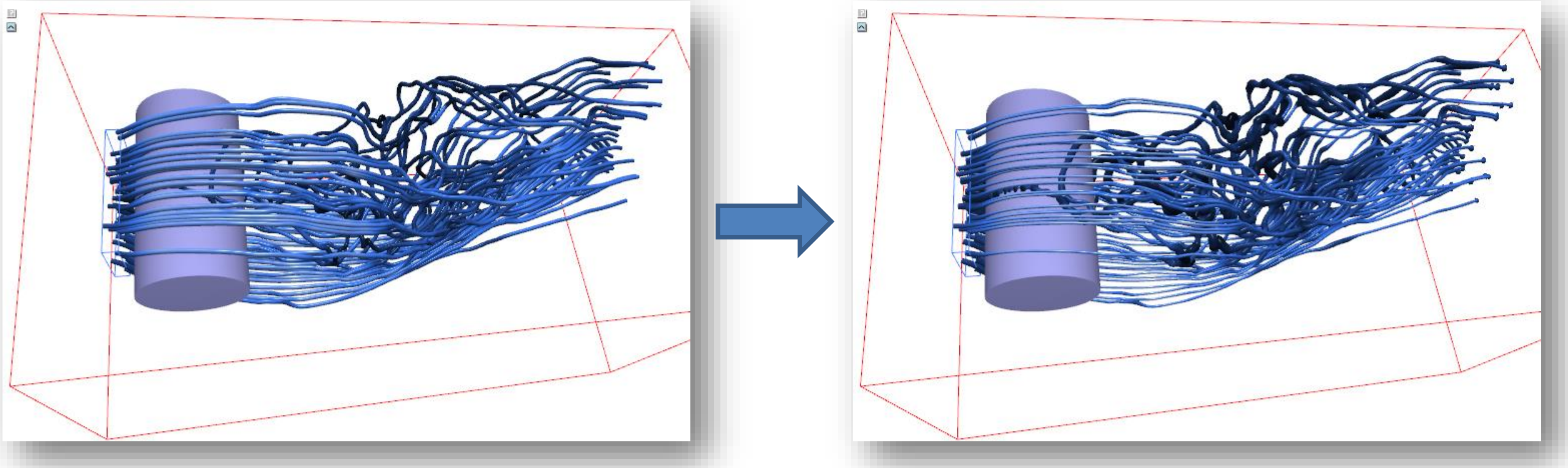


- Over the whole line, not only per segment!

- Track normal vector (“right” or „up“)
  - Initialize arbitrarily (orthogonal to line direction)
  - Re-orthogonalize wrt. tangent after each step (2 cross products)

Optional: Adapt Tube Thickness

- Try to keep volume constant („bubble gum“)
  - Adapt radius at each **vertex**
  - Clamp max radius!



- For analytic rendering: cylinders become cones!

Q&A

Questions ?