Vector Field Visualization

Flow Line Computation

Courtesy of Prof. Han-Wei Shen
(CSE 5544, The Ohio State University)

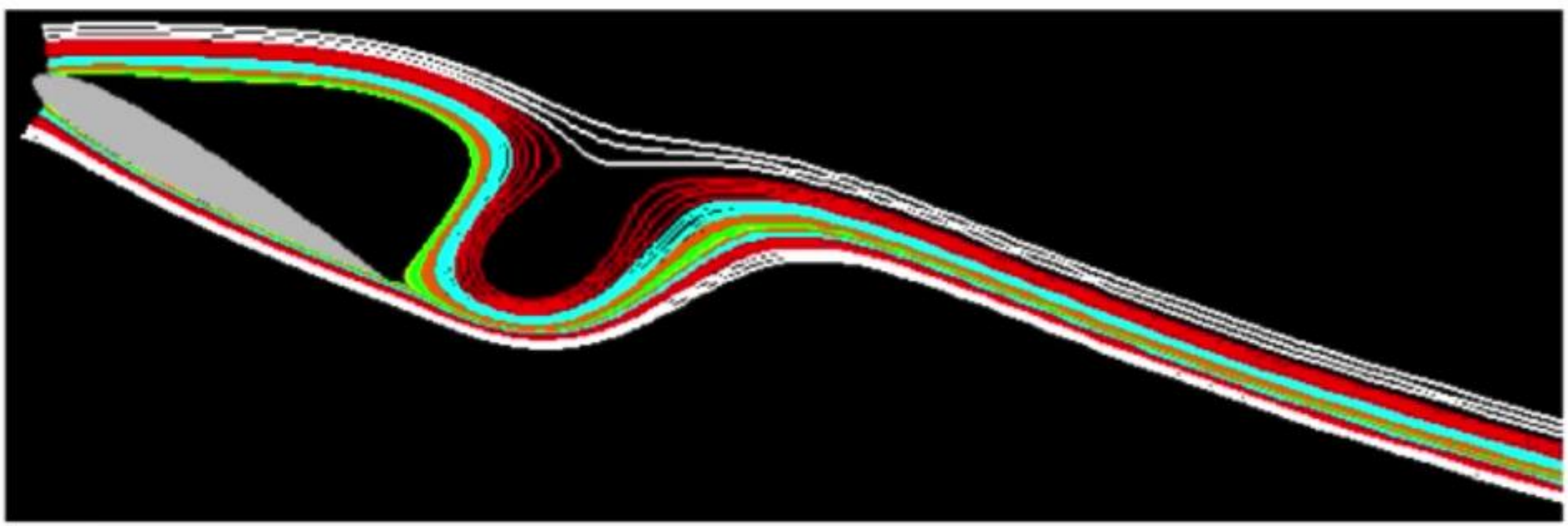
Flow Line Visualization

- One of the most fundamental flow visualization techniques
- Type of flow lines:
  - **Streamline:** a field line tangent to the velocity field at an instant in time
  - **Pathline:** the trajectory of a massless particle released from a seed point over a period of time
  - **Streakline:** a line joining the positions, at an instant in time, of particles released from a seed point
  - **Timeline:** a line connecting a row of particle that are released simultaneously
- In a steady flow filed, streamlines, pathlines, and streaklines are identical

Streamlines

- Streamline is a line that is tangential to the instantaneous velocity direction (velocity is a vector, and it has a magnitude and a direction)

- Release a particle into the flow and perform ***numerical integration*** to compute the path of the particle

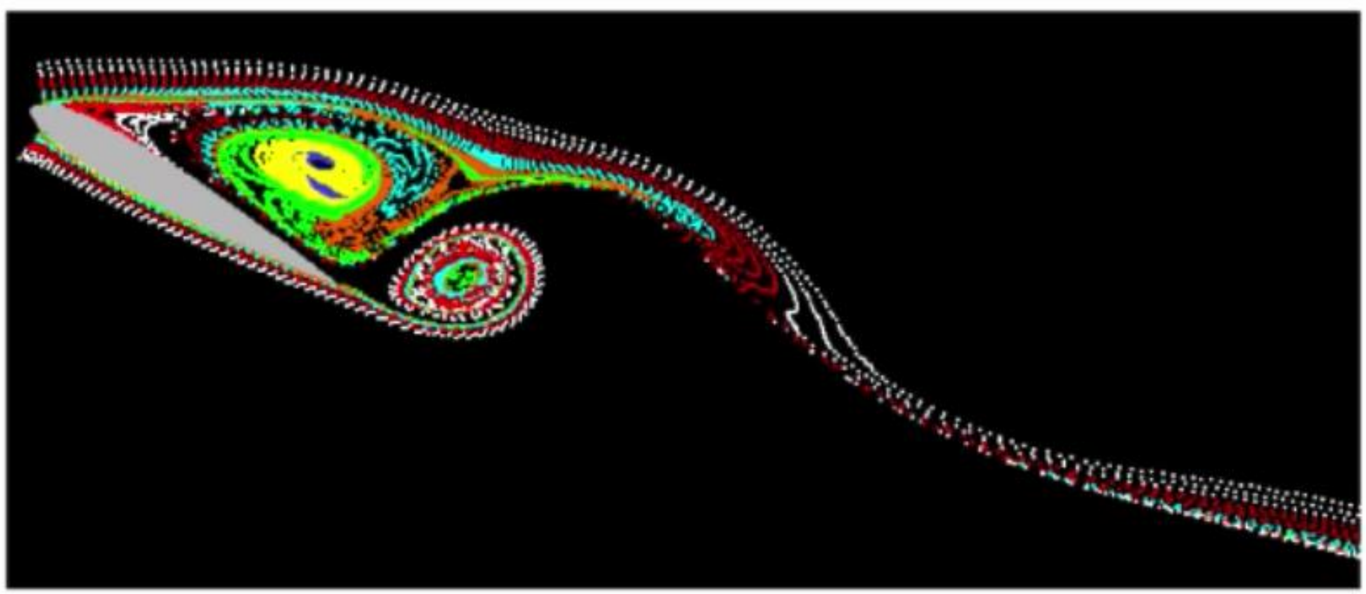


Pathlines

- A pathline shows the trajectory of a single particle released from a fixed location (seed point)
- Experimental method: marking a fluid particle and taking a time exposure photo of its motion will generate a pathline
- This is similar to what you see when you take a long-exposure photograph of car lights on a freeway at night

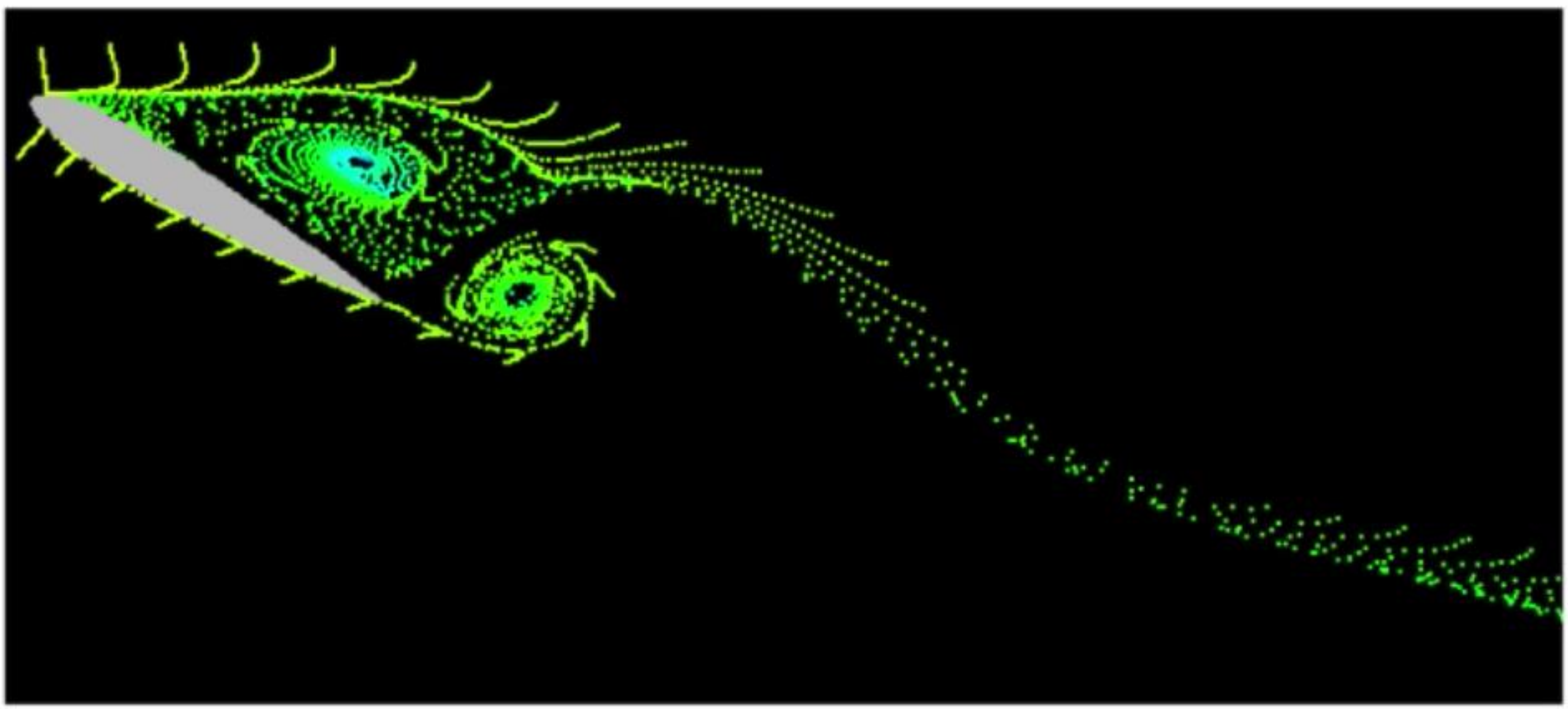
Streaklines

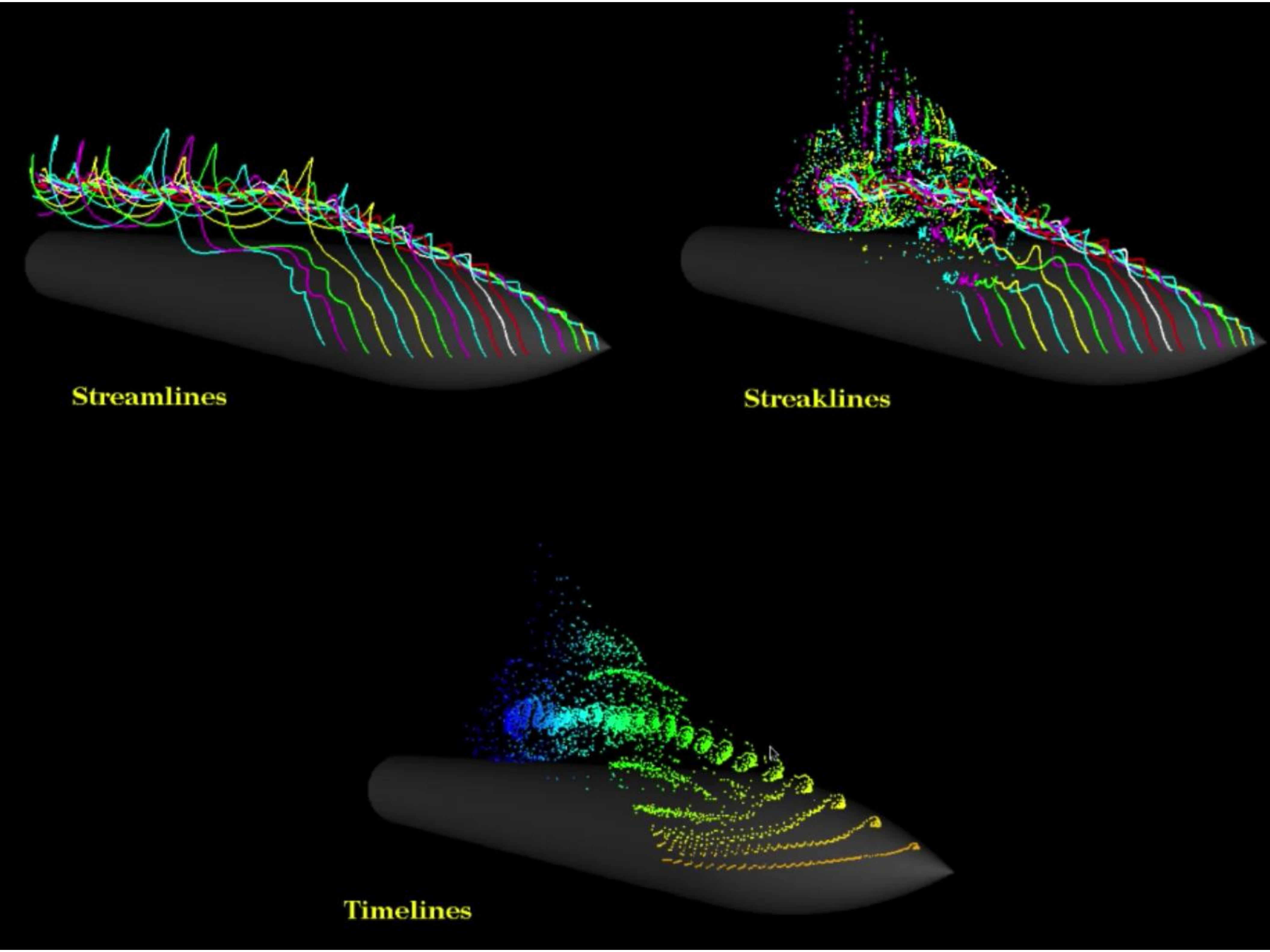
- Streakline is a line joining the positions, at an instant in time, of all particles that were previously released from a fixed location (seed point)
- Continuously inject particles into the flow at each time step and track the paths of the particles
- In a steady flow field, streamlines, pathline, and streaklines are identical. However, they can be very different in unsteady flows



Timelines

- Timeline is a line connecting a row of particles that released simultaneously
- Timelines are generated by injecting rows of particles at some fixed time interval




Streamlines
Streaklines
Timelines

S10-01

Computing Flow Line by Particle Tracing

- The path of a massless particle at position p at time t can be described by the following ordinary differential equation:

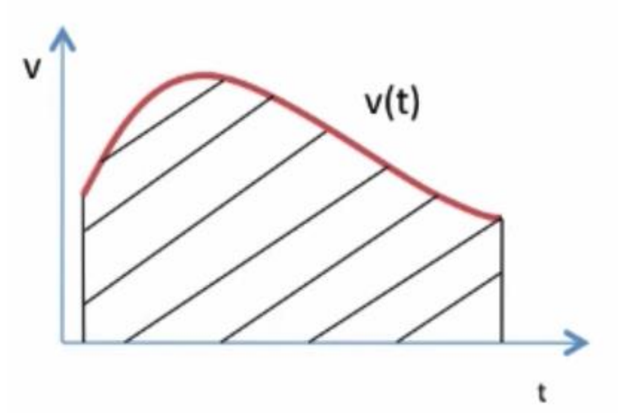
$$\underbrace{\frac{d\mathbf{p}}{dt} = v(\mathbf{p}(t))}_{\boxed{\text{steady flow}}} \quad \text{or} \quad \underbrace{\frac{d\mathbf{p}}{dt} = v(\mathbf{p}(t), t)}_{\boxed{\text{unsteady flow}}}$$

- And the positions of the particle can be computed by

$$p(t + \Delta t) = p(t) + \underbrace{\int_{t}^{t+\Delta t} v(p(t),t)dt}_{\boxed{\text{Solved by numerical integration}}}$$

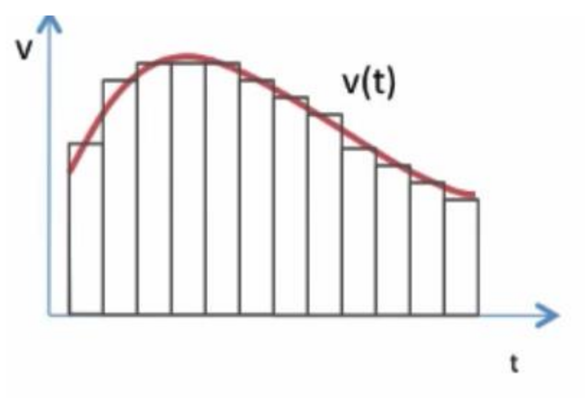
Numerical Integration

- Discrete Approximation of the continuous integration result



- Calculate area under curve for curve y(t)
    - Trapezoidal approximation

- Error related to the step size used

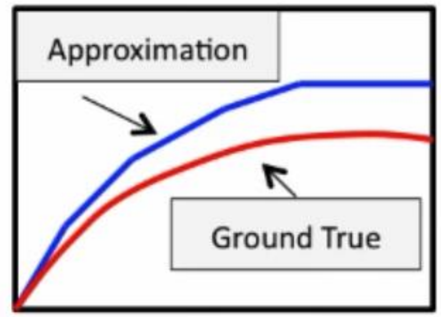


- Below we describe a few popular methods of numerical integration for particle tracing

Euler’s Method

- Simple but lower accuracy

$$\boxed{\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{v}(\boldsymbol{p}_k) \times \Delta t}$$



- This equation can be derived from Taylor series expansion where y is one component of the P position, and y’ is the one component of the velocity, h is the step size (Δt)

$$y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{1}{2}h^2 y''(t_0) + \dots$$
$$\Rightarrow y(t_0 + h) = y(t_0) + hy'(t_0) + O(h^2)$$

S10-02

2nd Order Runge-Kutta (RK2)

- Improved accuracy, more commonly used

$$\boxed{\begin{aligned} &\boldsymbol{p}^* = \boldsymbol{p}_k + \boldsymbol{v}(\boldsymbol{p}_k)\,\Delta t \\ &\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + (\boldsymbol{v}(\boldsymbol{p}_k) + \boldsymbol{v}(\boldsymbol{p}^*)\,)\ x\ \Delta t/2 \end{aligned}}$$

- Can also be derived from taylor series (next slides)

2^{nd} Order Runge-Kutta (RK2)

- Improved accuracy from Euler for ODE

$$y'(t) = f(t, y(t))$$

- Differentiating the above equation

$$y''(t) = f_t(t, y(t)) + f_y(t, y(t))y'(t) = f_t(t, y(t)) + f_y(t, y(t))f(t, y(t))$$

- From Taylor series expansion:

$$y(t+h) = y(t) + hy'(t) + \frac{1}{2}h^2 y''(t) + O(h^3)$$

$$= y(t) + hf(t,y) + \frac{1}{2}h^2[f_t(t,y) + f_y(t,y)f(t,y)]$$

$$= y(t) + \frac{1}{2}hf(t,y) + \frac{1}{2}h[f(t,y) + hf_t(t,y) + hf(t,y)f_y(t,y)]$$

4^{th} Order Runge-Kutta (RK4)

- Better accuracy, recommended

$$\boldsymbol{a} = 2\Delta t\, \boldsymbol{v}(\boldsymbol{p}_k),$$

$$\boldsymbol{b} = \mathbf{2}\Delta t\, \boldsymbol{v}(\boldsymbol{p}_k + \boldsymbol{a}/2),$$

$$\boldsymbol{c} = 2\Delta t\, \boldsymbol{v}(\boldsymbol{p}_k + \boldsymbol{b}/2)$$

$$\boldsymbol{d} = 2\Delta t\, \boldsymbol{v}(\boldsymbol{p}_k + \boldsymbol{c}/2),$$

$$\boxed{\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + (\boldsymbol{a} + 2\boldsymbol{b} + 2\boldsymbol{c} + \boldsymbol{d})/6}$$

Put it All Together
Particle Tracing Algorithm

1. Specify a seed position p(0), t = 0
2. Perform cell search to locate the cell that contains the p(t)
3. Interpolate the velocity field to determine the velocity at p(t)
4. Advance the particle from p(t) to p(t+Δt) using a numerical integration method
5. Repeat from step 2 until the particle moves a certain distance or goes out of bound

Notes on Particle Tracing

- The accuracy of particle tracing depends highly on the step size and the integration method
- Flow solvers are often second-order accurate in time, so particle tracing method should be at least third order or higher
- The velocity data need to be interpolated between two consecutive time steps, which can introduce errors too