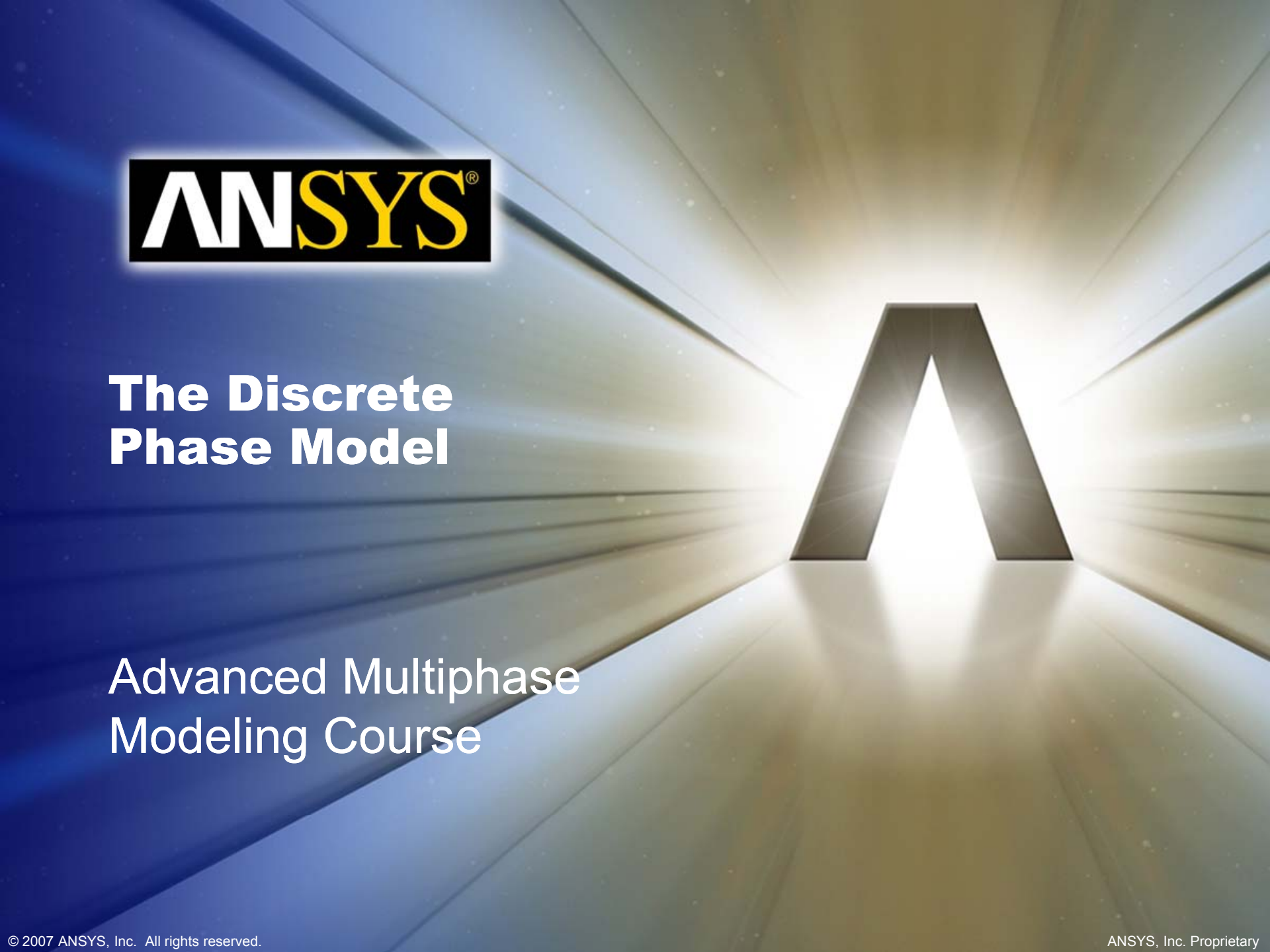


The Discrete Phase Model

Advanced Multiphase Modeling Course

Basic Concepts – Forces Acting on Particles

- Particle acceleration

$$m_p \frac{d\vec{u}_p}{dt} = \vec{F}_{\text{drag}} + \vec{F}_{\text{pressure}} + \vec{F}_{\text{virtual mass}} + \vec{F}_{\text{gravity}} + \vec{F}_{\text{other}}$$

- Drag force

$$\vec{F}_{\text{drag}} = A_p \frac{\rho}{2} C_D \left\|\vec{u} - \vec{u}_p\right\| \left(\vec{u} - \vec{u}_p\right) = m_p \frac{18}{24} \frac{\rho}{\rho_p} \frac{C_D}{d_p} \left\|\vec{u} - \vec{u}_p\right\| \left(\vec{u} - \vec{u}_p\right)$$

- Pressure force

$$\vec{F}_{\text{pressure}} = \int_{\text{surface}} -p\,\vec{n}\,dS = \int_{\text{volume}} -\nabla p\,dV = -\frac{m_p}{\rho_p}\nabla p$$

- Virtual mass force

$$\vec{F}_{\text{virtual mass}} = \frac{m_p}{2} \frac{\rho}{\rho_p} \frac{d\left(\vec{u} - \vec{u}_p\right)}{dt}$$

- Gravitation

$$\vec{F}_{\text{gravity}} = m_p \frac{\rho_p - \rho}{\rho_p} \vec{g}$$

Basic Concepts – Other Forces

- Forces in rotational reference frames
  - Includes forces on particles that arise due to rotation of the reference frame.
- Thermophoretic force
  - Small particles suspended in a gas that has a temperature gradient experience a force in the direction opposite to that of the gradient.
- Brownian force for sub-micron particles
  - The effects of Brownian motion can be optionally included in the additional force term.
- Saffman's lift force
  - Lift force due to fluid shear can also be included in the additional force term as an option.
- User-defined (other) forces

Coupling with Continuous Phase

- One-way coupling
  - Continuous phase variables may change along the particle trajectory.
  - Changes affect particles along the particle trajectory.
- DPM interpolates free-stream properties which are stored in a local data structure.

$$\vec{u}\left(\vec{x}_p\right) = \vec{u}_{\text{cell}} + \nabla\vec{u}_{\text{cell}}\left(\vec{x}_p - \vec{x}_{\text{cell}}\right)$$

$$T_\infty = T_{\text{cell}}$$

$$C_{i,\infty} = X_{i,\text{cell}}\frac{p_{\text{op}}}{RT_\infty} \qquad \theta_R = \left(\frac{G_{\text{cell}}}{4\sigma}\right)^{1/4}$$

Categorization – Eulerian/Lagrangian Model

- Hydrodynamics
  - Particle relaxation times — All
  - Change of diameter — Yes
  - Particle-wall collision — Yes
  - Particle-particle collision — Yes
  - Coalescence — Yes
- Chemics
  - Heat transfer — Yes
  - Mass transfer — Yes
  - Change in composition — Yes
- Restrictions
  - Volume fraction < 10%

DPM is the only model with discrete representation of all variables

Categorization – Eulerian/Lagrangian Model

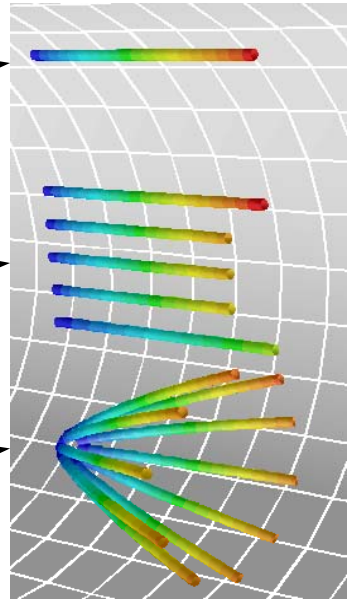
- Applications to be adressed
  - Particulate flows
    - Cyclones
    - Dryers
    - Pneumatic conveyors
    - Coal-fired furnaces
  - Droplet flows
    - Sprays
    - In-Cylinder flows
    - Combustors
    - Quenchers
  - Bubbly flows
    - Aerators
    - Bubble columns at low volume fractions

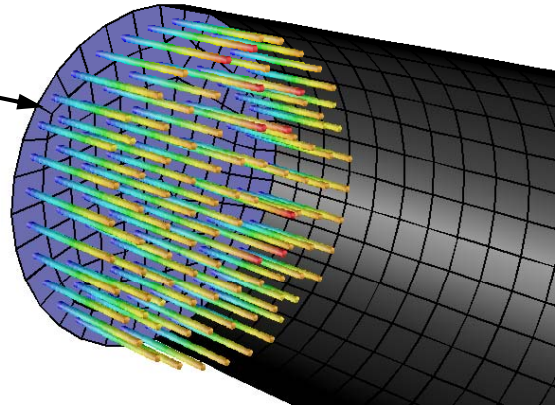
Injections

- Particles need initial information about
  - Location
  - Velocity
  - Temperature
  - Start time
  - Diameter
  - Composition
  - Flow rate
  - Stop time
- Injections are a means to provide this information.
- Two generic types available:
  - Direct specification of starting conditions.
  - Automated computation of starting conditions based on atomization sub models for injection device.

Injection Types – Direct Specification

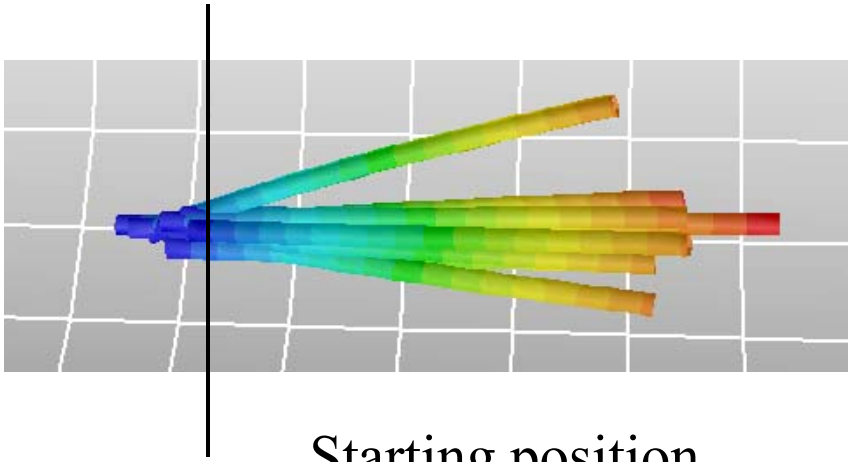
- Single
- Group
- Hollow cone
- Surface
- File





Injection Types – Direct Specification

- Solid cone is randomizing the particles to reduce grid dependency.
- A sufficient number of streams must be specified.
- The following particle properties are randomized:
  - Velocity
  - Direction
  - Location



Injection Types – Direct Specification

Injection type	Available size distributions	No. of particles computed
Single	No	1
Group	Linear, RR, RR-log	# of specified particle streams
Cone	No	# of specified particle streams
Solid-cone	No	# of specified particle streams
Surface	Uniform, RR, RR-log	(# of faces) × (# of diameters)

Injection Types – Automated Computation

- Initial information to be specified:
  - Location
  - Temperature
  - Start time
  - Composition
  - Flow rate
  - Stop time
  - Information about the injector device to apply primary breakup theories.
- Initial information, computed automatically
  - Size distribution
  - Velocity distribution
- Available auto-injection types
  - Plain orifice atomizer
  - Pressure-swirl atomizer
  - Air-blast atomizer
  - Flat fan atomizer
  - Effervescent atomizer

Details of each can be found in the Appendix

Particle Types

- Each injection needs one of the following particle types:

Type	Physical effects	Assumptions
Inert	Tracking, Heating, Cooling	None
Droplet	Vaporization, Boiling	Energy equation, at least two species
Combustion	Devolatization, heterogeneous reactions, heating	Energy equation, at least three species

Drag Coefficients

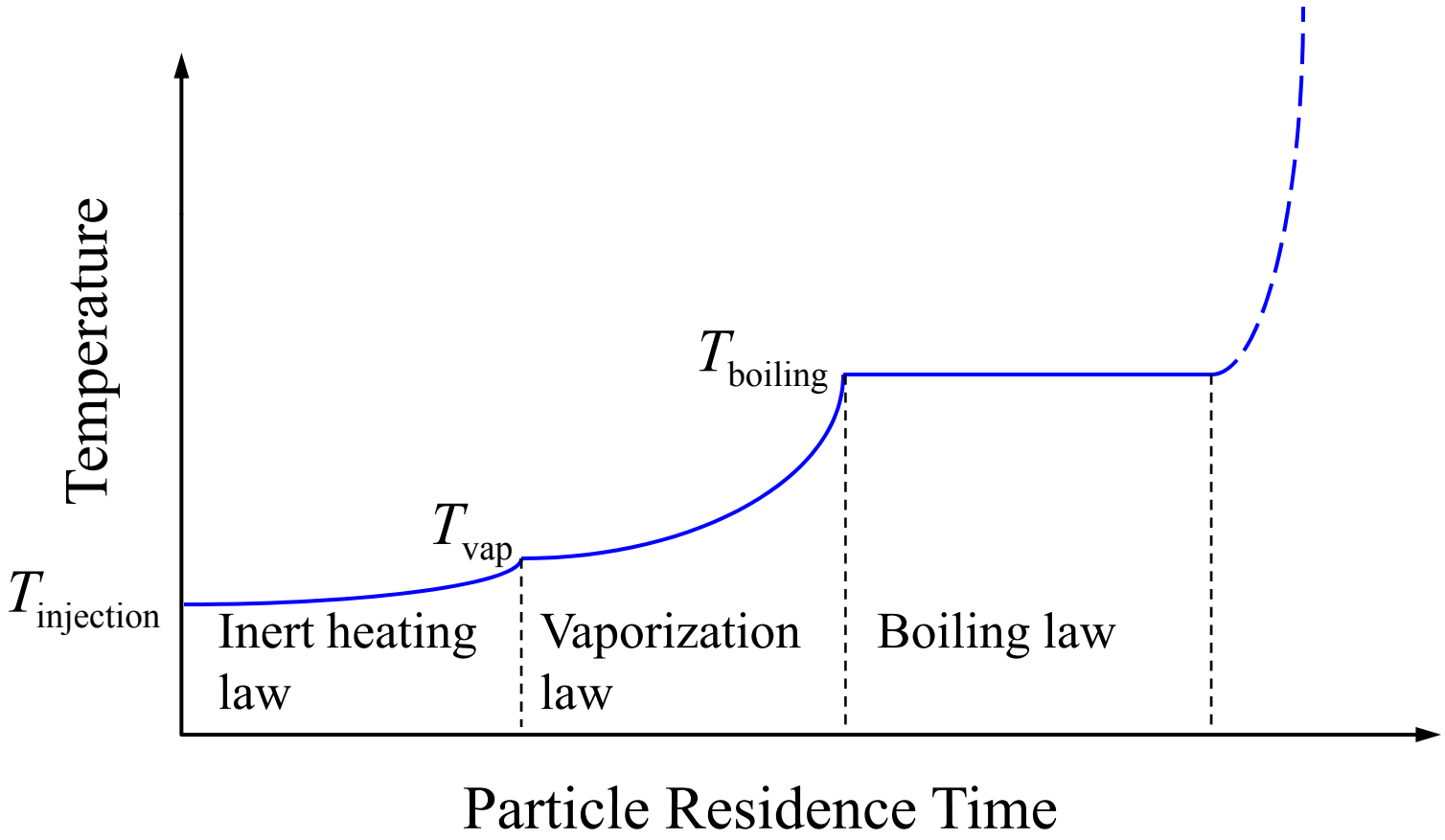
- Drag force in particle force balance

$$\vec{F}_{\text{drag}} = A_p \frac{\rho}{2} C_D \left\| \vec{u} - \vec{u}_p \right\| \left(\vec{u} - \vec{u}_p \right)$$

$$= m_p \frac{18}{24} \frac{\rho}{\rho_p} \frac{C_D}{d_p} \left\| \vec{u} - \vec{u}_p \right\| \left(\vec{u} - \vec{u}_p \right)$$

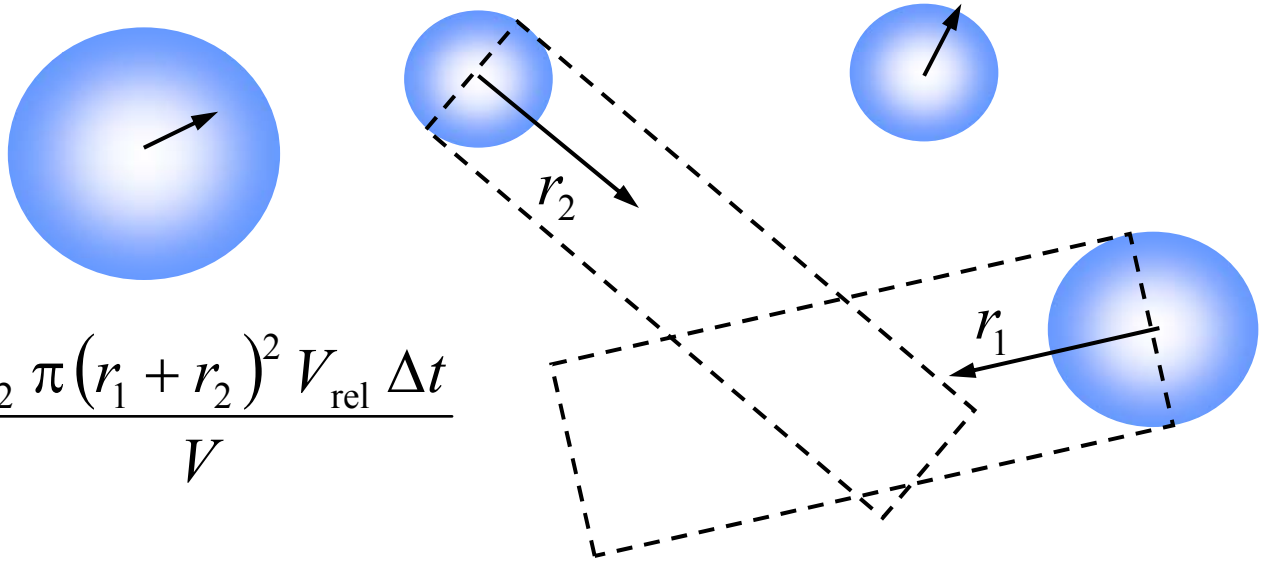
- Several choices
  - Morsi-Alexander (spherical particles)
  - High mach number effects
  - Haider-Levenspiel (non-spherical particles)
  - Stokes law with Cunningham correction (submicron particles)

Heat and Mass Transfer Models



Collision and Coalescence Model (O'Rourke)

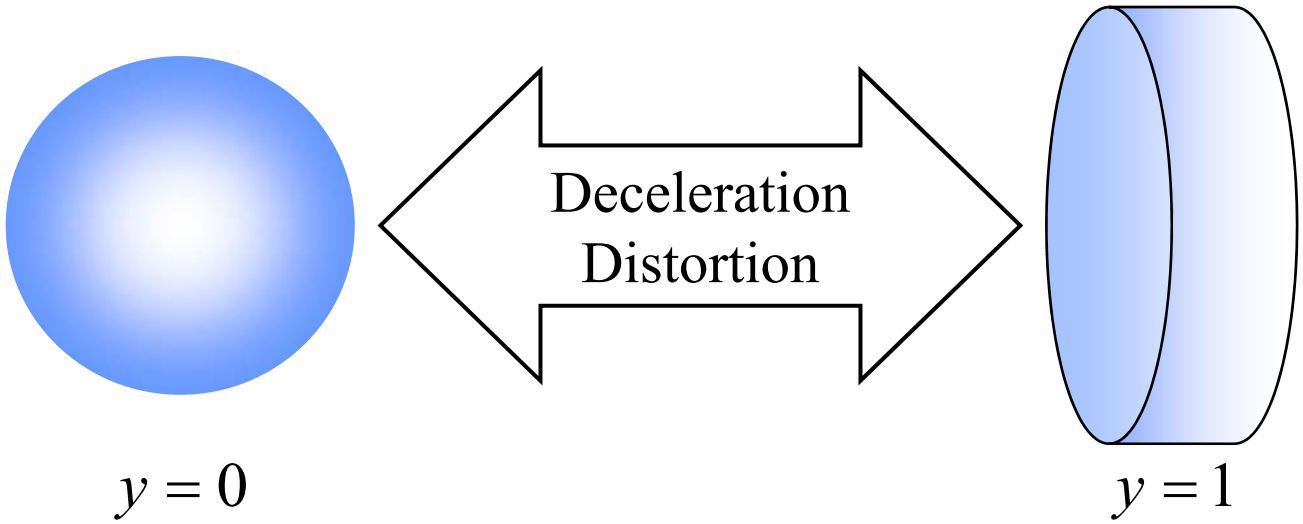
- Particles move around and may collide or may not.
- Probability depends on particle velocity and diameter.



$$\overline{n} = \frac{n_2 \, \pi \left(r_1 + r_2\right)^2 V_{\text{rel}} \, \Delta t}{V}$$

- Collision volume relative to the cell volume is a measure of the probability of collision.

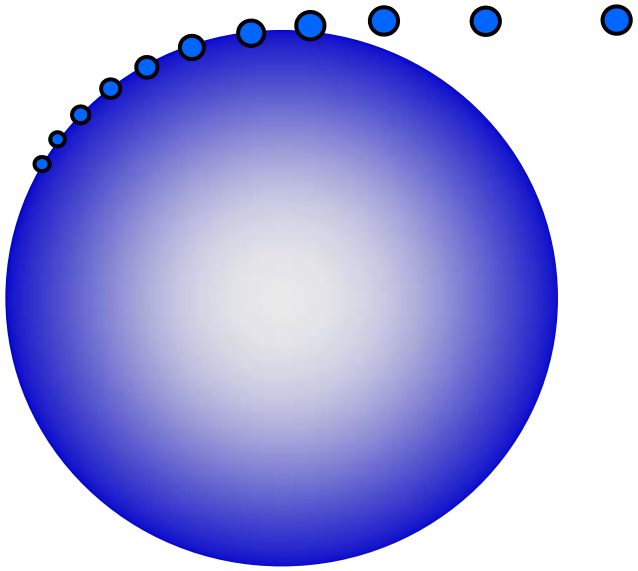
Secondary Atomization – Taylor Analogy Breakup



Spring-Mass System	Distorting and Oscillating Droplet
Restoring force of the spring	Surface tension forces
External force	Droplet drag force
Damping force	Droplet viscous forces

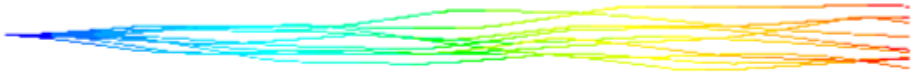
Secondary Atomization – The Wave Breakup Model

- Continuous shedding of child droplets due to growth of Kelvin-Helmholtz waves.
- Reitz derived from a jet stability analysis the maximum growth rate and corresponding wavelength.
- The size of the child droplets is proportional to the fastest growing wavelength.



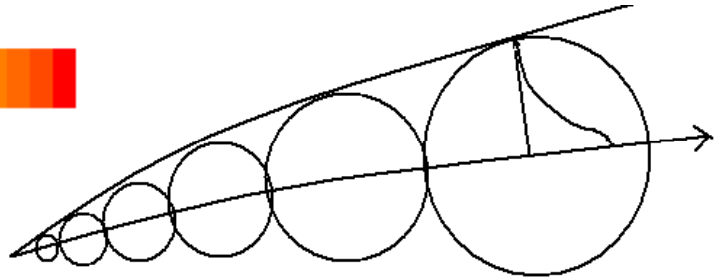
Turbulence Models

- When particles enter a turbulent eddy, they try to follow it for the time they are crossing the eddy.
- This effect leads to lateral dispersion which must be considered in modeling.
- Two approaches are available:
  - Discrete random walk model



$$u_i' = \varsigma\sqrt{\frac{2k}{3}}$$

  - Particle cloud model

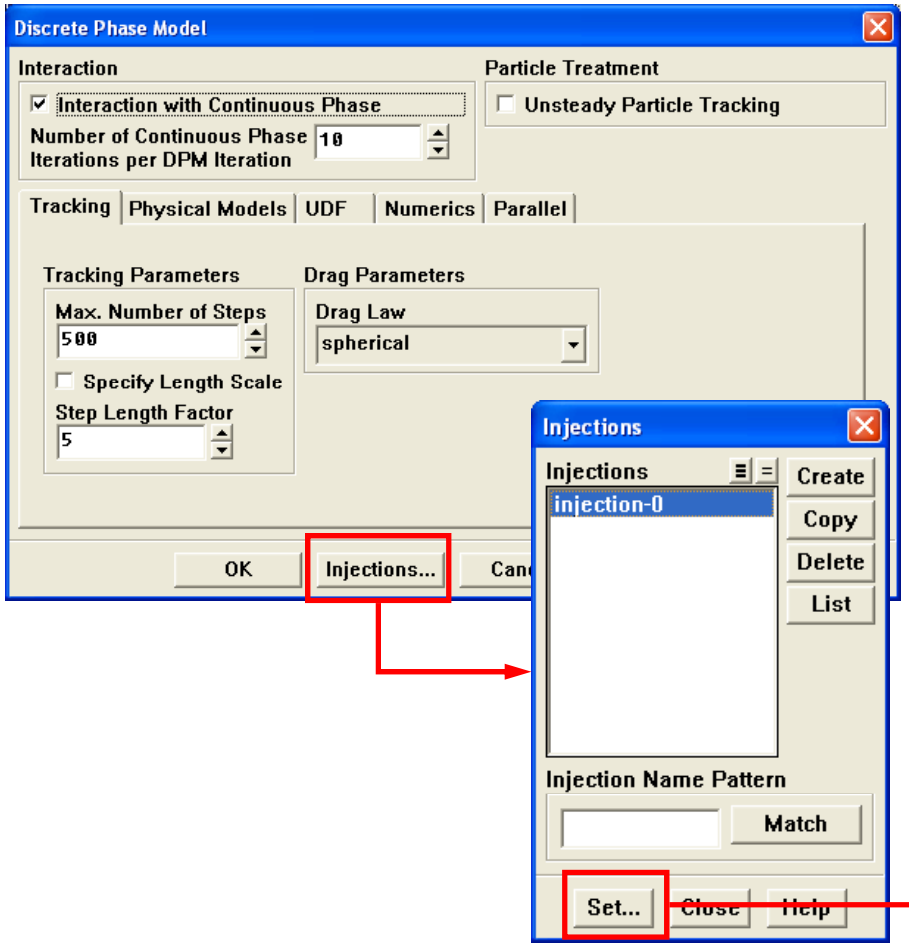


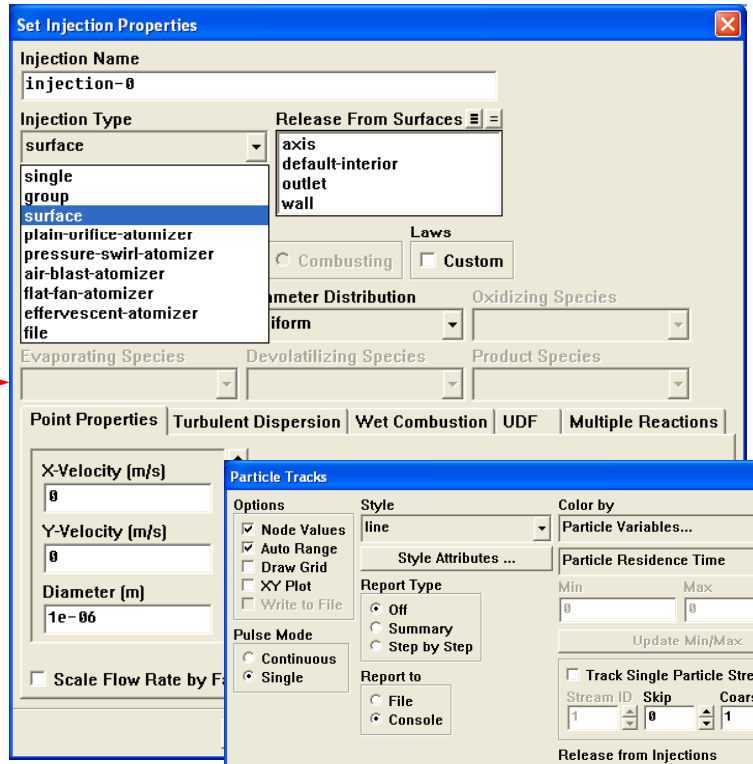
Discrete Random Walk vs. Cloud Tracking

- Discrete Random Walk (stochastic) tracking:
  - Accounts for local variations in flow properties.
  - Requires a large number of stochastic tries in order to achieve a statistically significant sampling (function of grid density).
  - Insufficient number of stochastic tries results in convergence problems due to non-smooth particle source term distributions.
  - Recommended for use in complex geometry
- Cloud tracking:
  - Local variations in flow properties get averaged inside the particle cloud.
  - Smooth distributions of particle coupling source terms.
  - Each diameter size requires its own cloud trajectory calculation.

Discrete Phase Model (DPM) Setup

Define → Models → Discrete Phase…

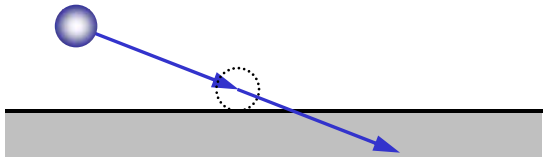




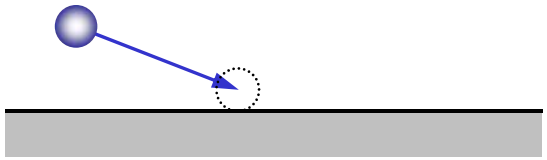
Display → Particle Tracks…

DPM Boundary Conditions

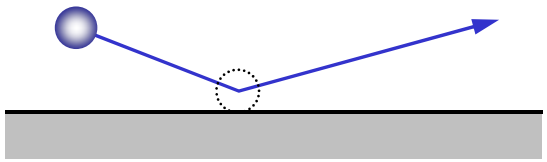
- Boundary Conditions for DPM must be set.
  - Escape

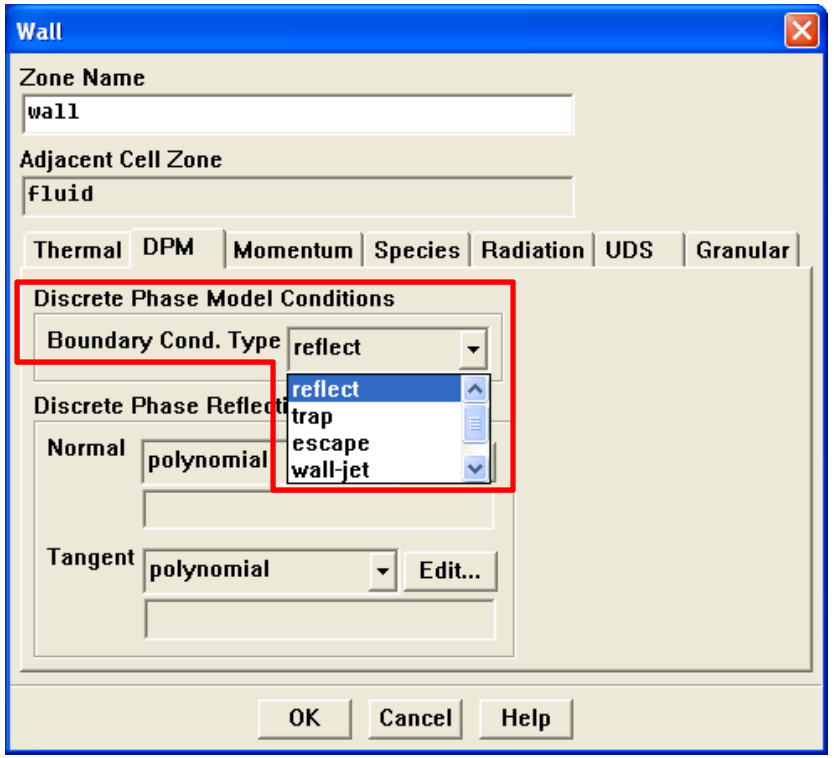


  - Trap

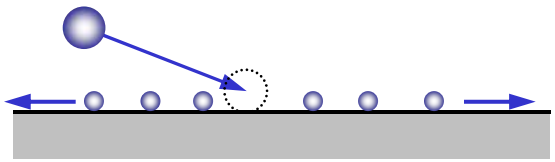


  - Reflect





  - Wall Jet

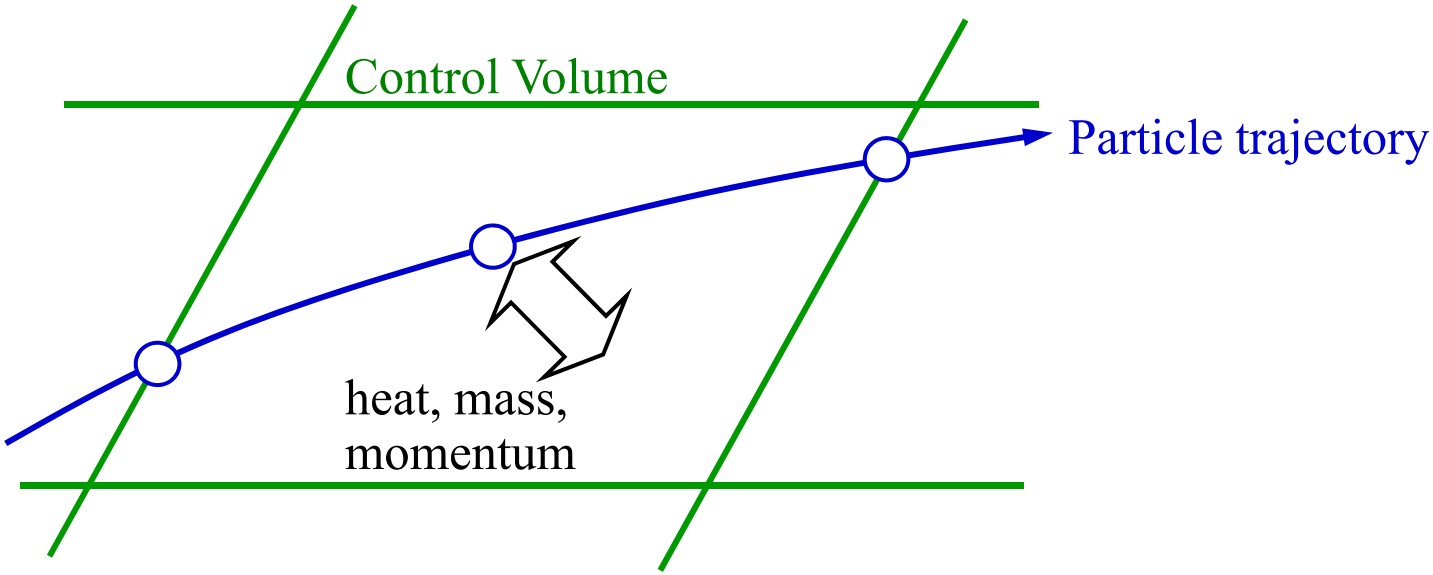


Coupling

- The discrete particle model has its own equations apart from the FLUENT solver.
- The particle equations and their sub models have their own numerical algorithms with their specific convergence behavior and settings.
- A coupled flow simulation will never converge if the particle algorithms are failing.
- In addition some sub models show stochastic behavior. Therefore take care for a smooth distribution of source terms.

Coupling

- Two-way coupling:
  - Transfer of momentum, energy and mass from continuous phase to particles is considered by means of source terms.



Coupling

- Procedure
  - Particles are computed including the source terms every N^{th} flow iteration.
  - For the next N flow iterations the particle source terms are kept constant and considered in the flow equations.
  - Flow solution changes due to particles.
  - Repeat until convergence is achieved.

- Don’t forget to switch off all Check Convergence buttons in the Residual Monitor panel!

Coupling Procedure – Steady State

- Lagrangian particles always are transient.
- Procedure to do steady state computation:
  - Freeze continuous phase fields
  - Put particle at injection point
  - Compute time steps based on local cell velocity
  - Integrate trajectory until
    - Maximum number of time steps is reached
    - OR Particle leaves the domain
    - OR Particle is collected by a wall
    - OR Particle evaporates, combusts, etc.
  - Step to next particle
  - Continue with continuous phase

Coupling Procedure – Transient

- Lagrangian particles and continuous flow fields must be coincident in time in order for the solution to make sense.
- Procedure for transient computation:
  - Freeze continuous phase fields
  - Take particle at its current location
  - Compute particle time steps based on local cell velocity
  - Integrate trajectory until
    - flow time step is reached, or
    - particle leaves the domain, or
    - particle is collected by a wall
  - Store particle at new location and step to next particle
  - Continue with continuous phase time step

Coupling – Underrelaxation

- Additional source terms appear in discretized flow equations.

$$a_p \phi_p + \sum_{nb} a_{nb} \phi_{nb} = b_p + S_{\text{DPM}}$$

- Affects solution process and convergence behavior.
- Source terms can be under-relaxed directly.
- Energy and species under-relaxation factors need to be reduced in order to converge.

Coupling – Transient

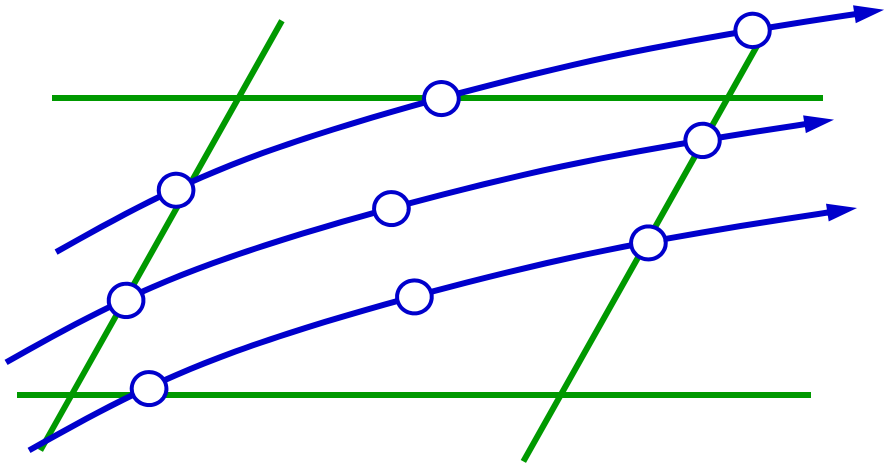
- The DPM source terms are updated only every particle iteration. If the underrelaxation factor is smaller than one, take care that sufficient particle iterations are performed within the time step in order to achieve the full source terms.
- You must use a factor of 1, when doing only one particle computation per time step.
- Reduce the Number of Continuous Phase Iterations per DPM Iteration if you need to lower the DPM under-relaxation factor.
- Use smaller time steps if it does not converge.

Solution Strategies for Steady Flows

- Two strategies possible:
  - Closer coupling between dispersed and continuous flow:
    - Increase underrelaxation factor for Discrete Phase
    - Decrease number of continuous phase calculations between trajectory calculations (< 3).
    - Lower underrelaxation factors for continuous phase.
  - Decoupling of dispersed and continuous flow:
    - Lower underrelaxation factor for Discrete Phase.
    - Increase number of continuous phase calculations between trajectory calculations (> 15)
- Smooth out particle source terms.
  - Increase number of stochastic particle trajectories.

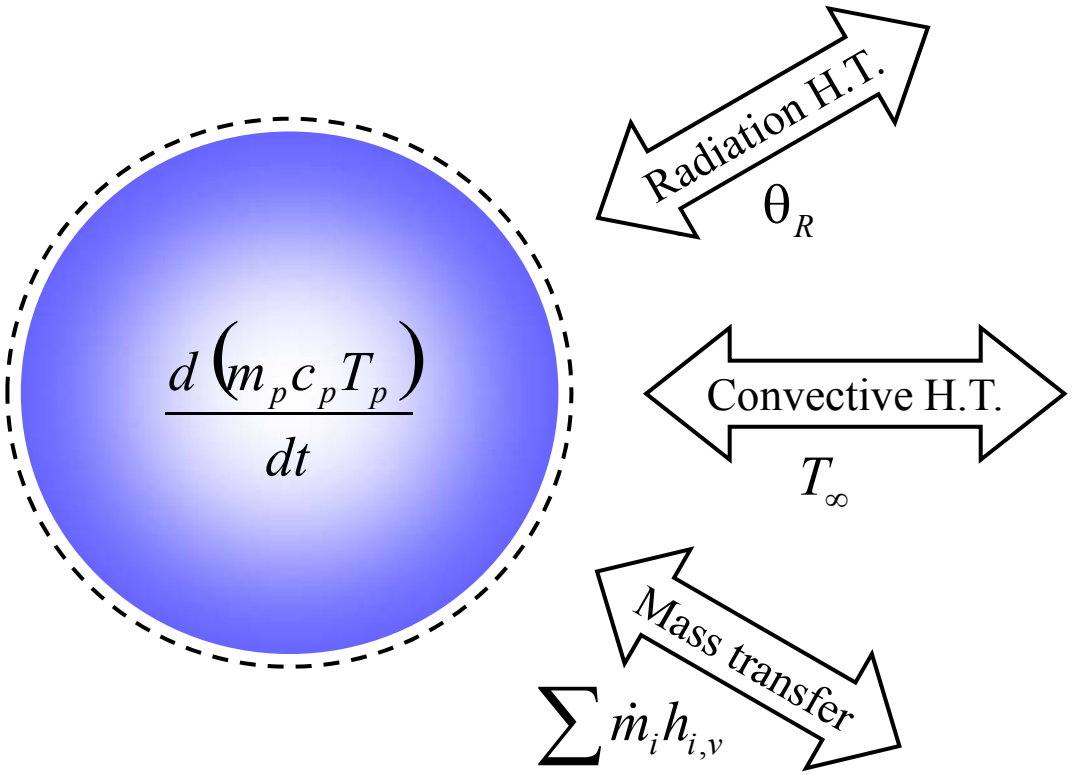
Solution Strategies

- During the tracking procedure every particle is seeing a "fresh" cell and makes no notice of particles having already visited it and marked with their source terms.
- This leads to overprediction of the source terms and ultimately poor convergence.
- Occurs mainly during evaporation, combustion, radiation.
- You can solve such cases mostly transient.



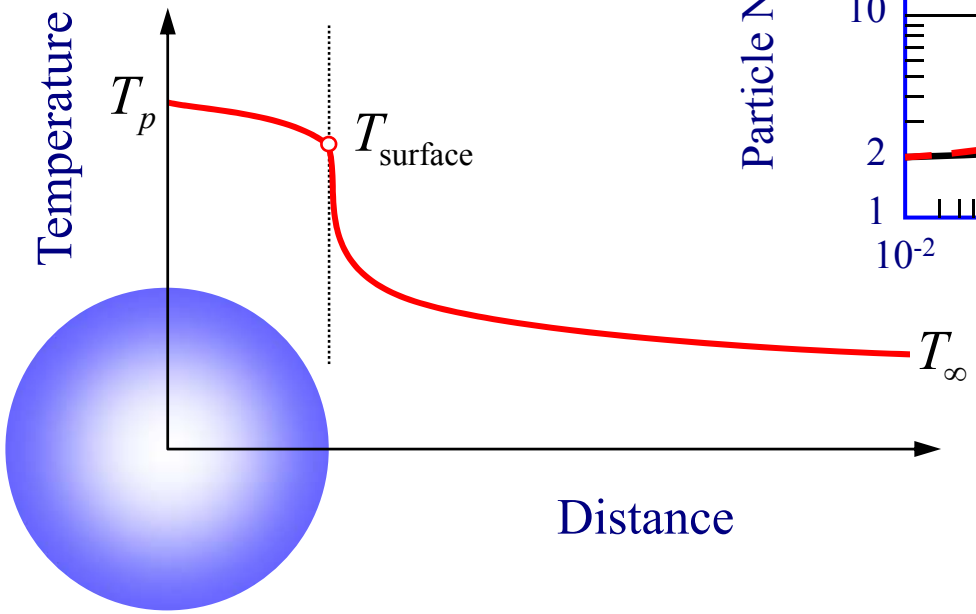
Appendix

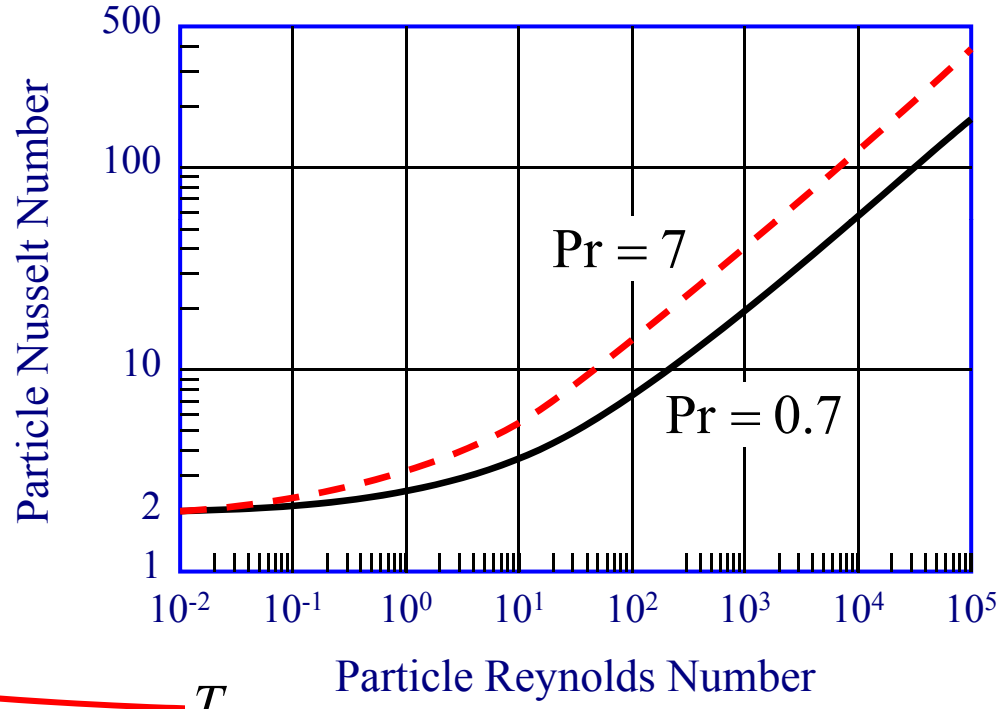
Basic Concepts – Heat and Mass Transfer



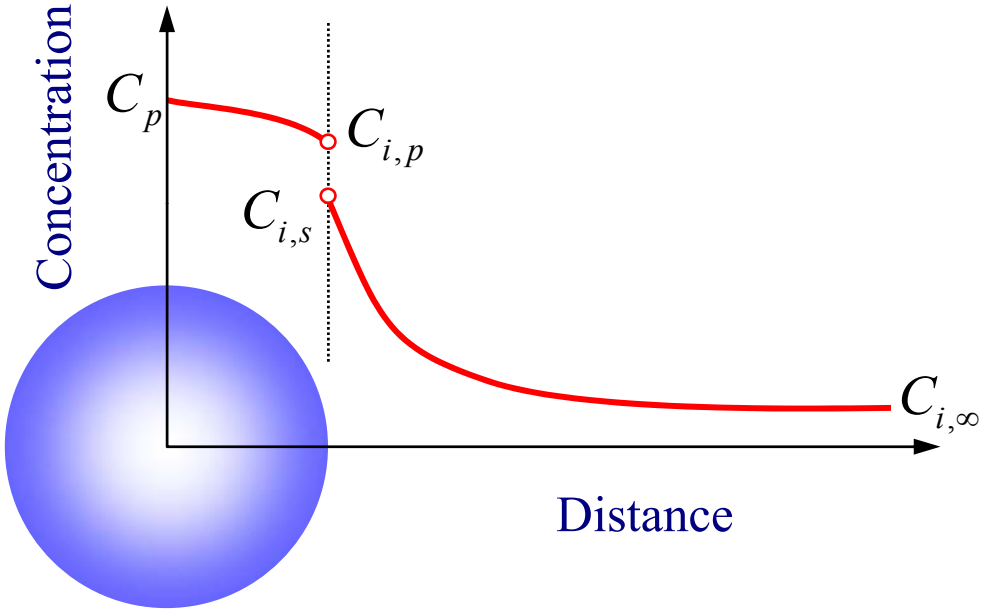
$$\frac{\partial\left(m_p C_p T_p\right)}{\partial t} = \pi d_p^2 h\left(T_\infty - T_p\right) + \sum \dot{m}_i h_{i,v} + \pi d_p^2 \varepsilon_p \sigma\left(\theta_R^4 - T_p^4\right)$$

Basic Concepts – Heat Transfer





Basic Concepts – Mass Transfer



$$\frac{dm_p}{dt} = \sum \dot{m}_i = -\pi d_p^2 \sum M_{w,i} k_c \left(C_{i,s} - C_{i,\infty}\right)$$

$$C_{i,s} = \frac{p_{\text{sat}}(T_p)}{R\,T_p} \qquad C_{i,\infty} = X_i \frac{p_{\text{op}}}{R\,T_\infty}$$

Basic Equations of Mass Transfer

Effect on the Energy Equation

$$\frac{d\left(m_p c_p T_p\right)}{dt} = \pi d_p^2 h\left(T_\infty - T_p\right) + \sum \dot{m}_i h_{i,v} + \pi d_p^2 \varepsilon_p \sigma\left(\theta_R^4 - T_p^4\right)$$

$$\frac{d\left(m_p C_p T_p\right)}{dt} = m_p C_p \frac{dT_p}{dt} + m_p T_p \frac{dC_p}{dt} + \underbrace{C_p T_p}_{h_p} \frac{dm_p}{dt}$$

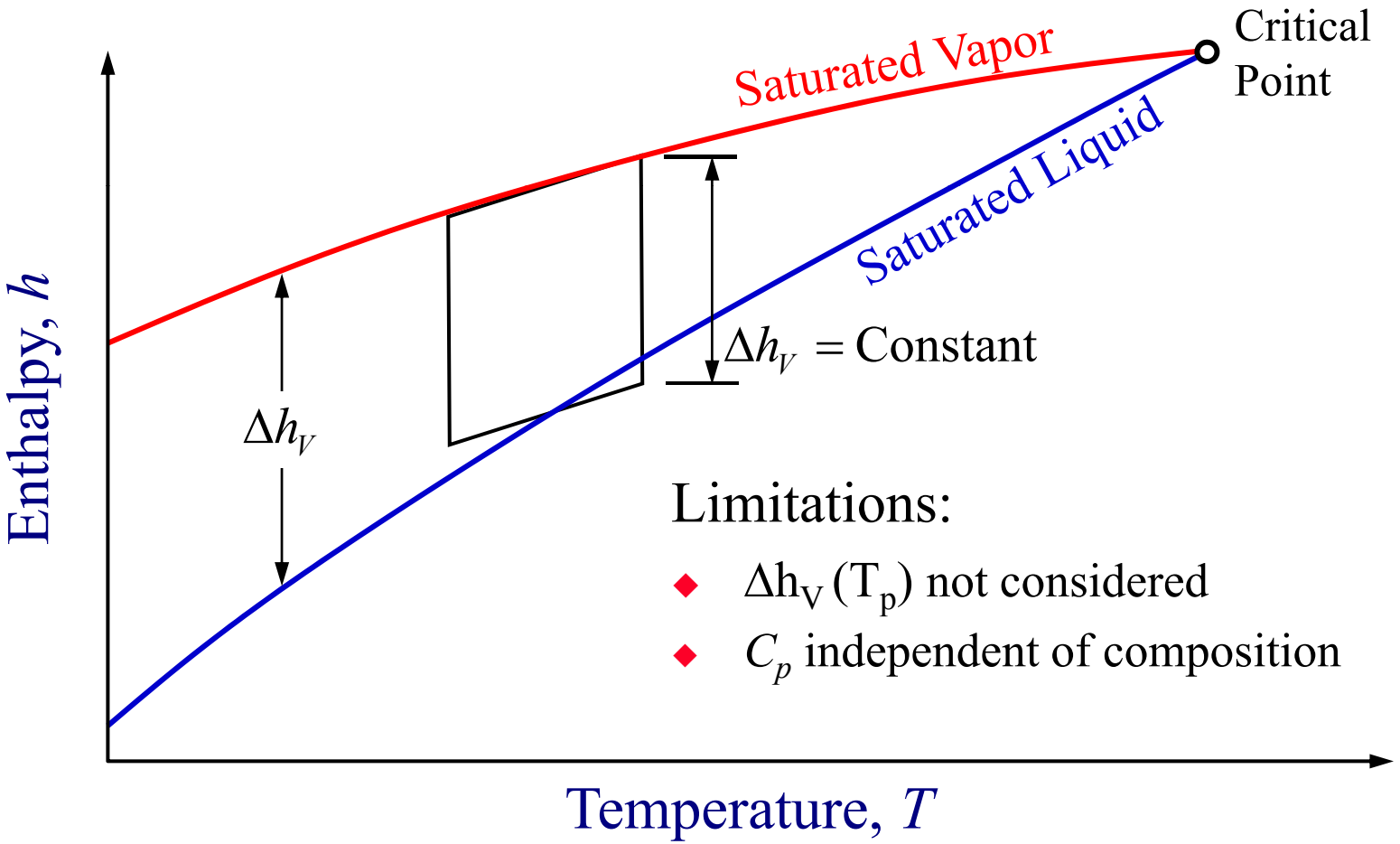
$\frac{dC_p}{dt} \rightarrow 0$

$$m_p C_p \frac{dT_p}{dt} = \pi d_p^2 h\left(T_\infty - T_p\right) + \sum \dot{m}_i \left(h_{i,v} - h_p\right) + \pi d_p^2 \varepsilon_p \sigma\left(\theta_R^4 - T_p^4\right)$$

$\left(h_{i,v} - h_p\right) \rightarrow \Delta h_{v,p}$

Basic Equations of Mass Transfer

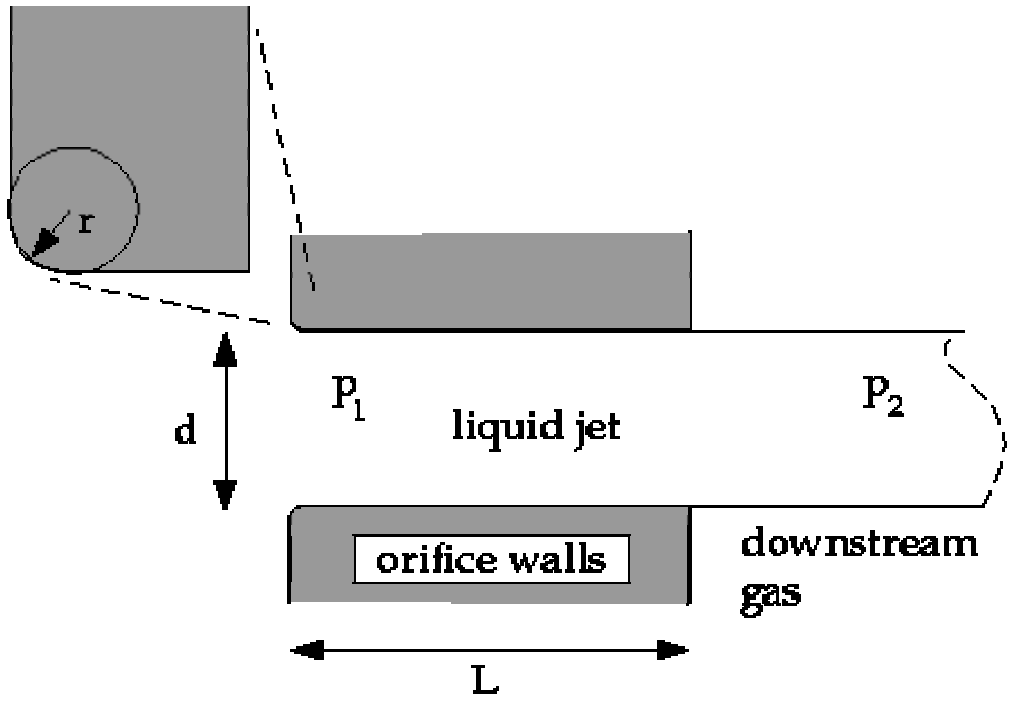
Latent Heat of Vaporization



Limitations:

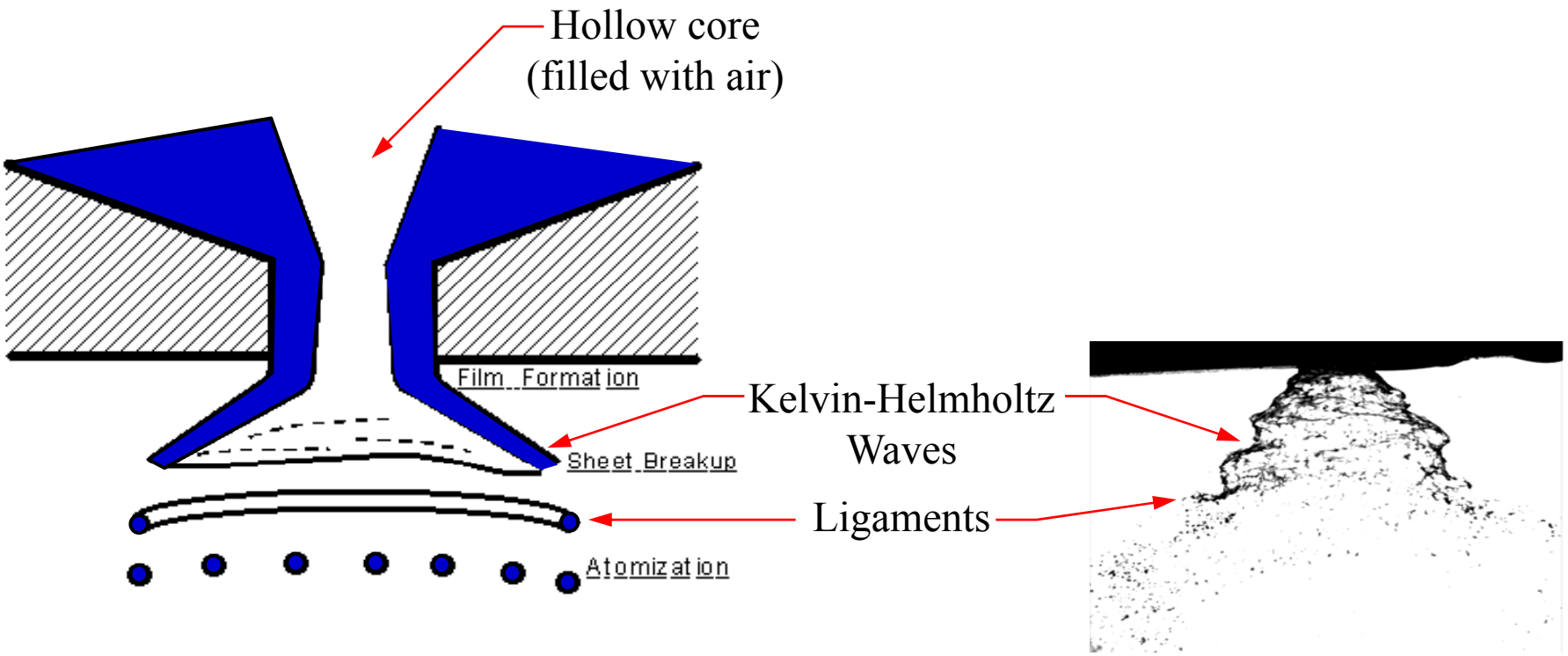
- $\Delta h_V (T_p)$ not considered
- C_p independent of composition

Injection Types – Plain Orifice Atomizer



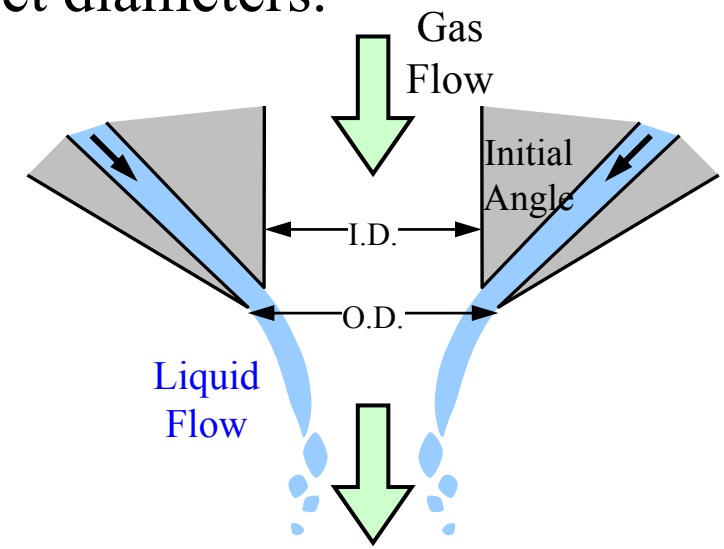
Injection Types – Pressure-Swirl Atomizer

- Liquid is accelerated in swirl chamber.



Injection Types – Air-Blast Atomizer

- Variation of pressure-swirl atomizer.
- Additional air is directed through the nozzle, leading to smaller droplet diameters.

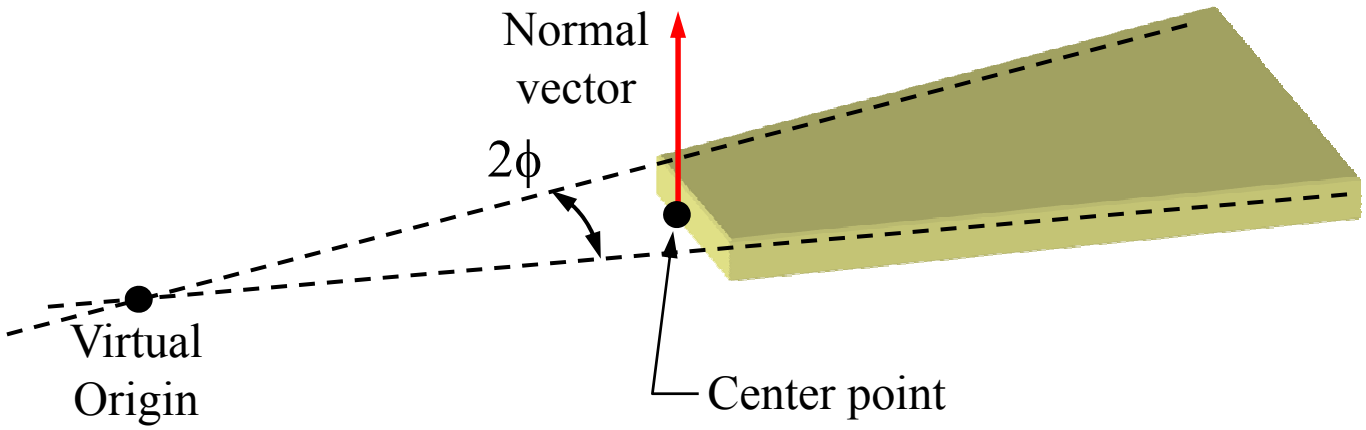


Air-Blast Atomizer

- Note: Gas flow is NOT setup for you automatically.

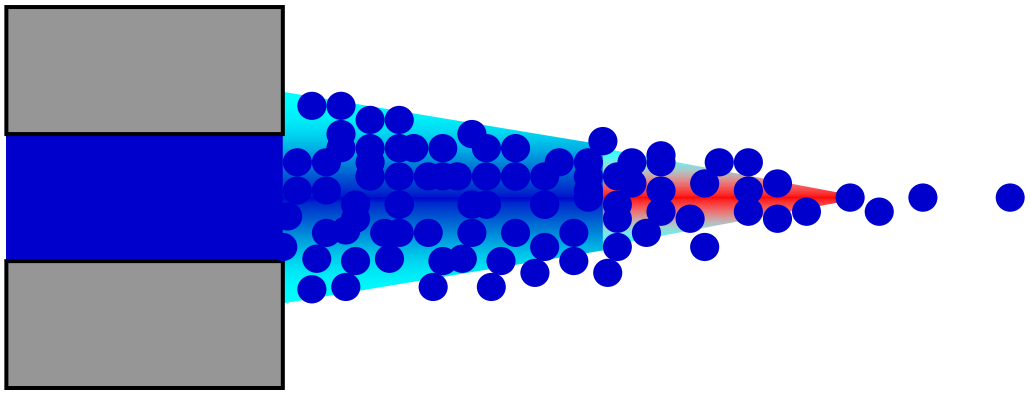
Injection Types – Flat-Fan Atomizer

- Effects similar to pressure-swirl atomizer.
- Liquid enters as a flat sheet.
- Sheet breakup is taken from pressure-swirl atomizer.

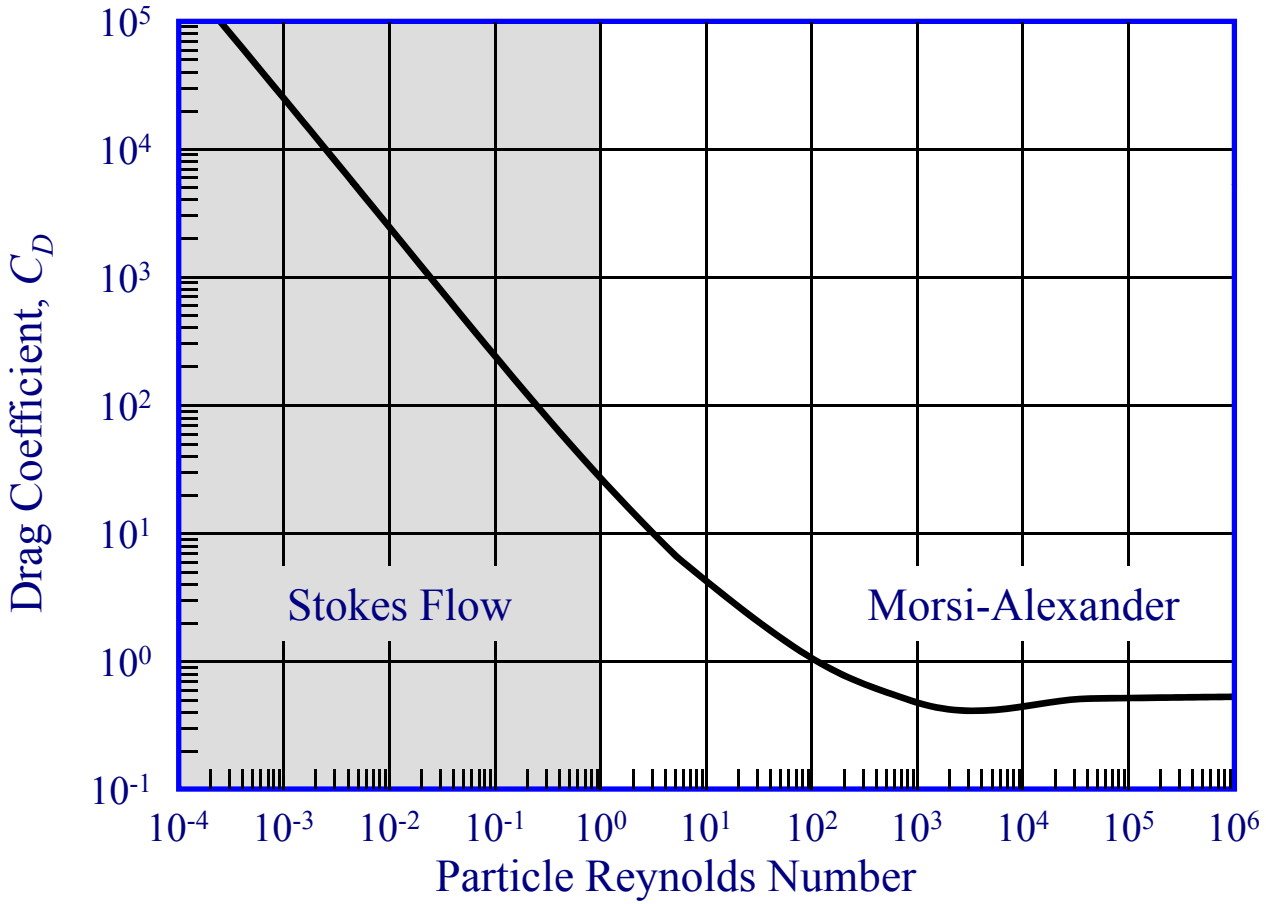


Injection Types – Effervescent Atomizer

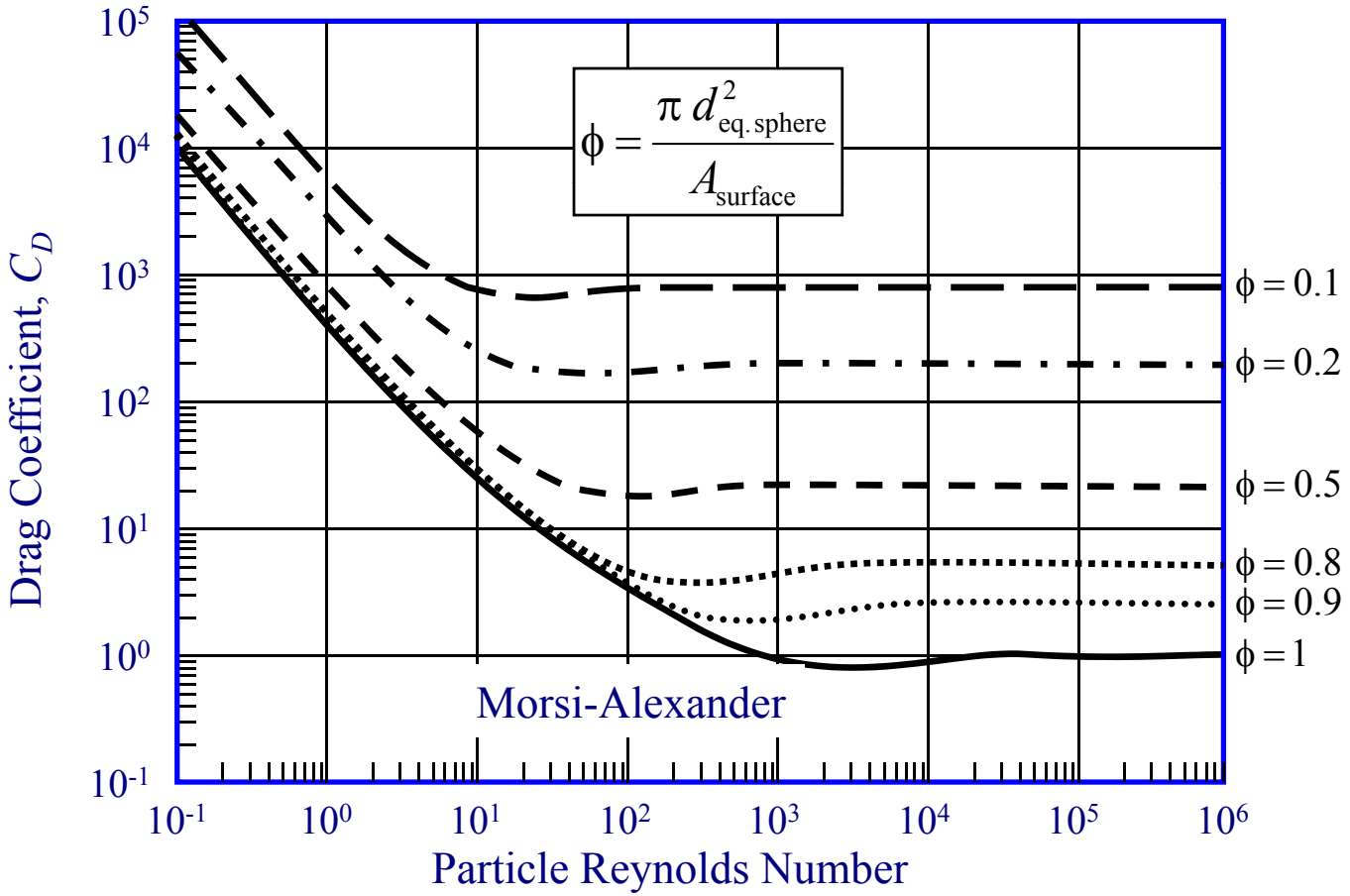
- Superheated or very hot liquid is discharged.
- Liquid is evaporating rapidly when leaving nozzle.
- A dense liquid core surrounded by a shroud of smaller droplets.



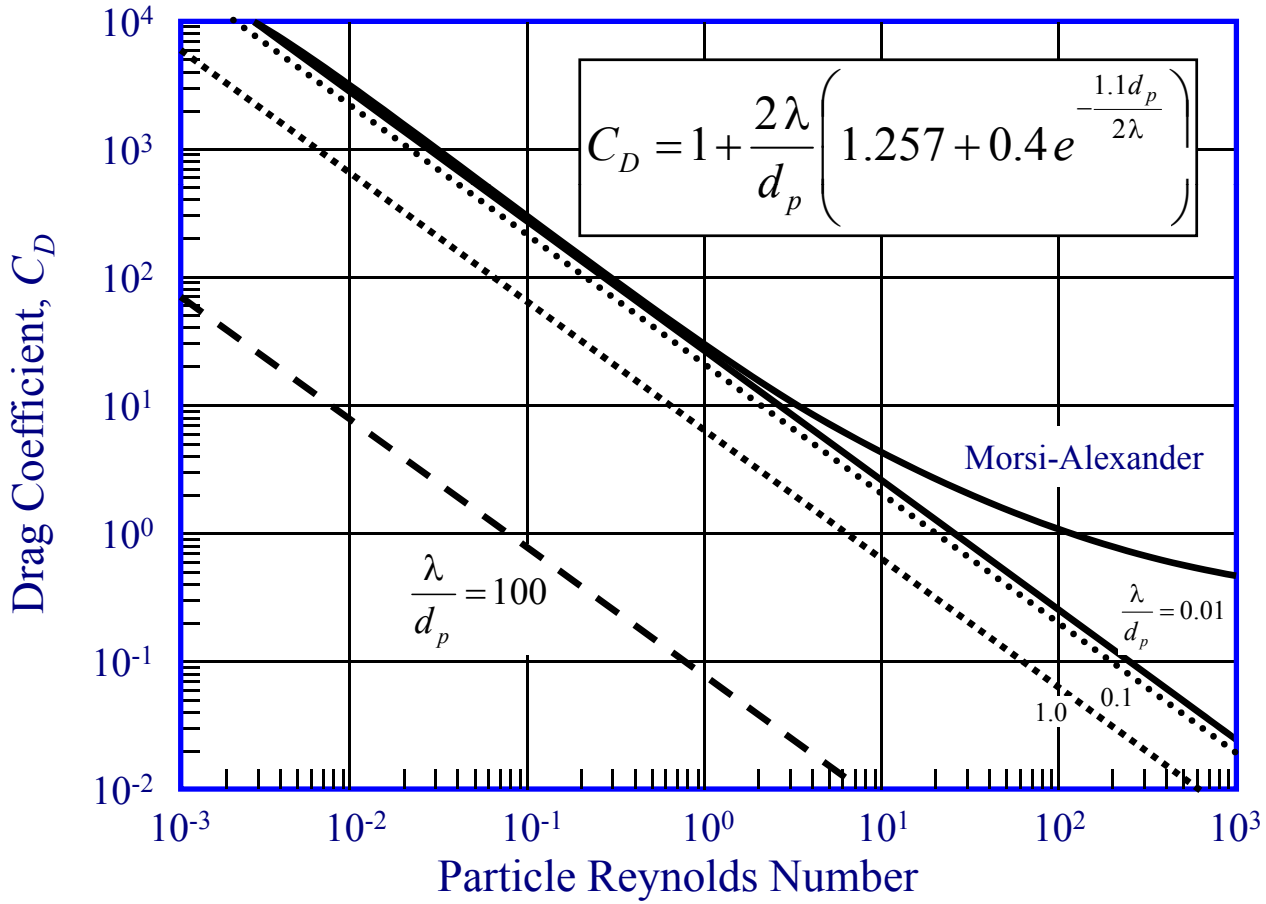
Drag Coefficients for Spherical Particles



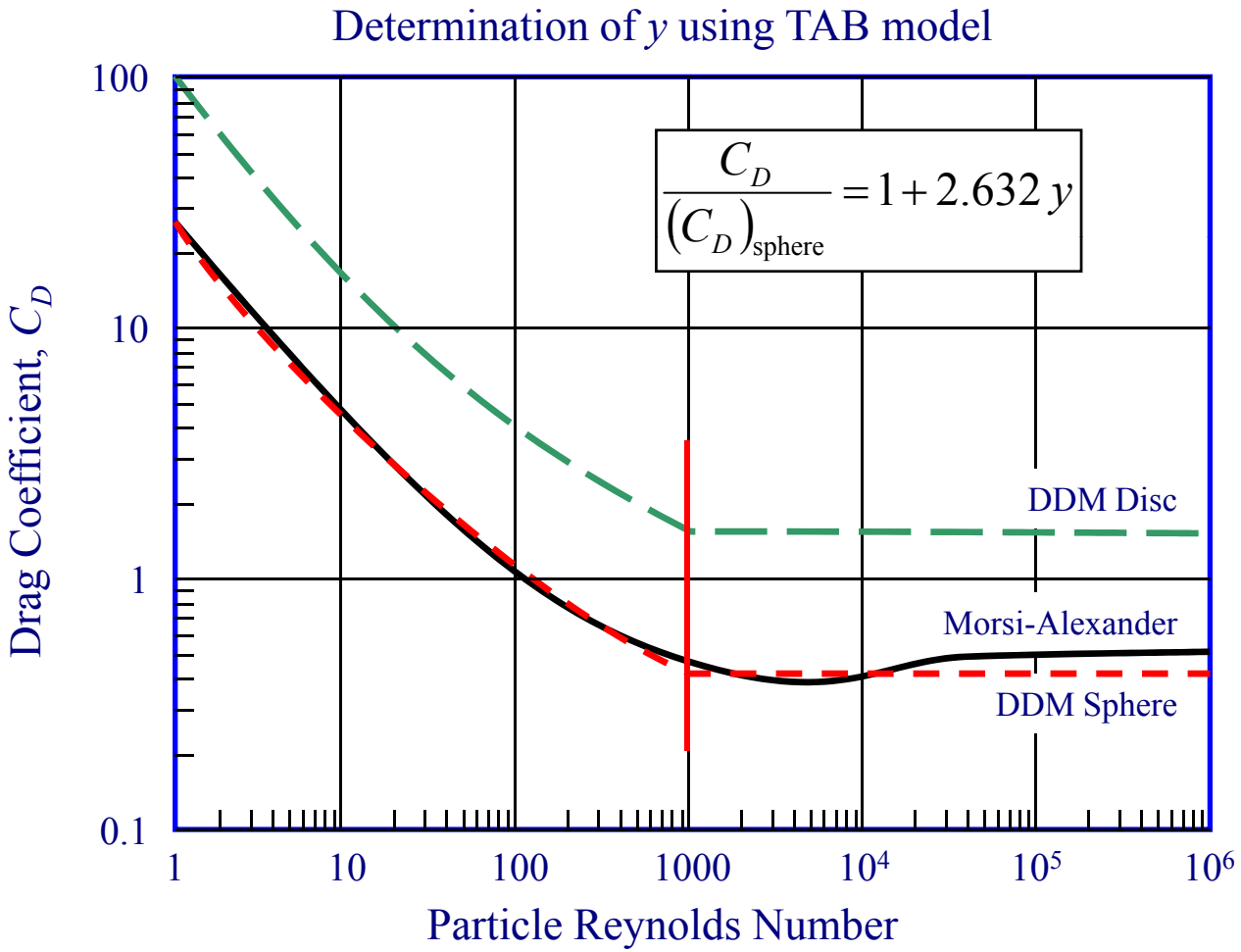
Drag Coefficients for Non-Spherical Particles



Drag Coefficients: Sub-Micron Particles

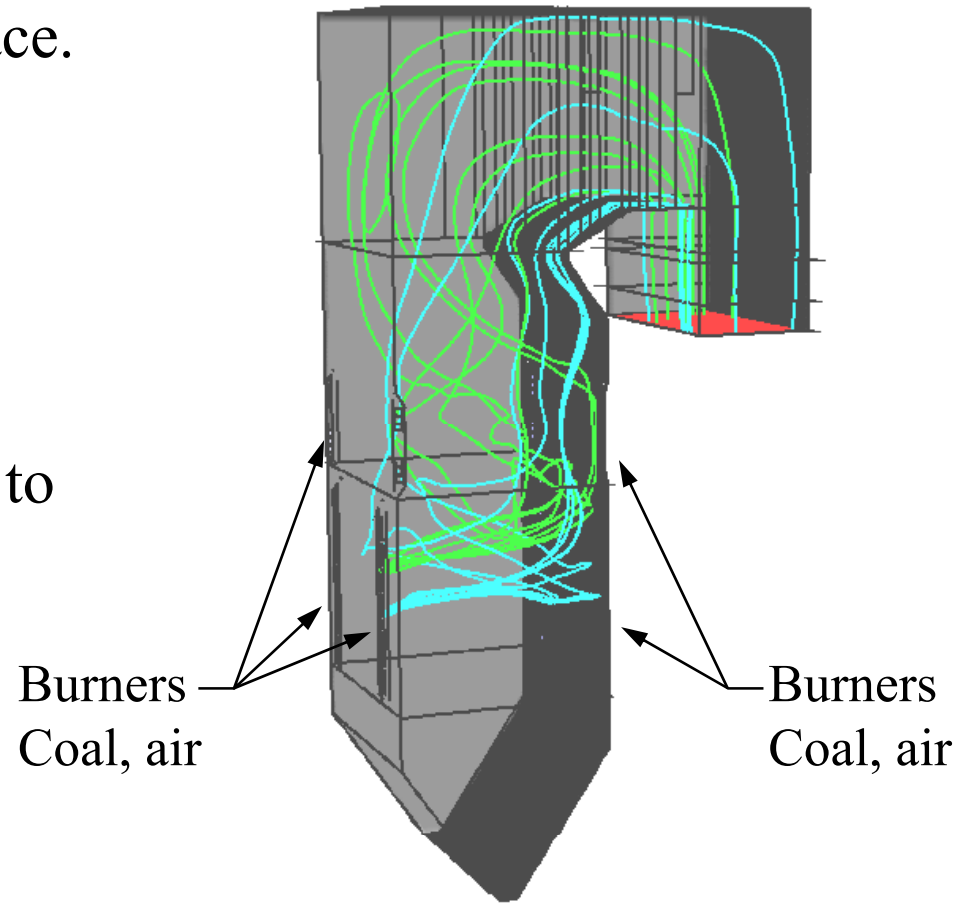


The Dynamic Drag Model



Numerics – A Real-World Problem

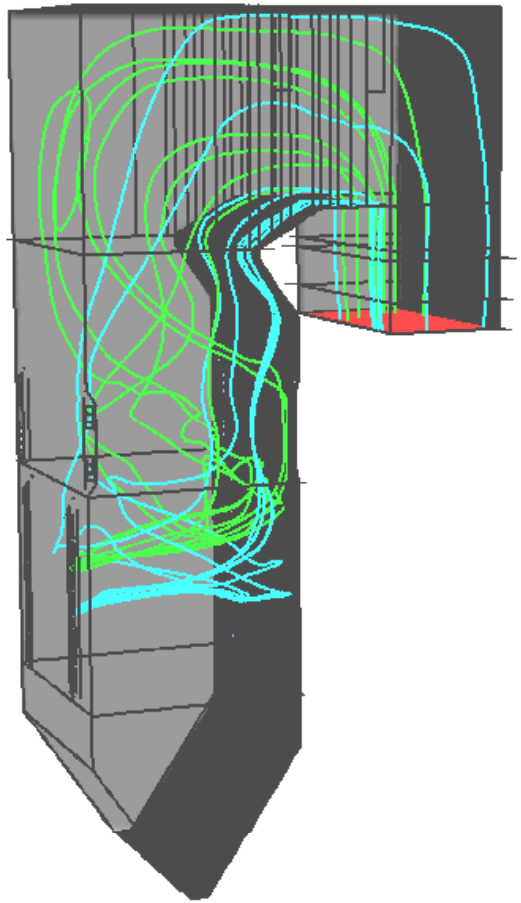
- 50 m high cross-fired furnace.
- 600,000 hex cells
- Coal combustion using Euler/Lagrange model
- 24 burners
- Around 20,000 trajectories to simulate coal particles



Numerics – A Real-World Problem

- Effectiveness of different schemes.
  - Constant integration step length

Euler implicit	98.7%
Exponential analytic	101.4%
Trapezoidal	100%
Runge-Kutta	505%

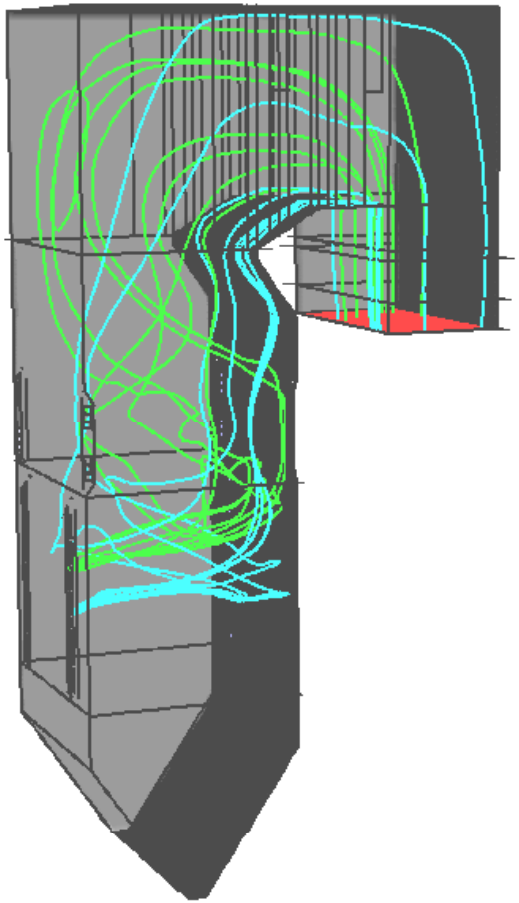


Numerics – A Real-World Problem

- Error based adaption including automated tracking scheme selection

Euler implicit	26.7%
Exponential analytic	24.4%
Trapezoidal	44.4%
Runge-Kutta	36.0%

- Acceleration up to a factor of 4



Numerics – Particle Trajectory Equations

- Particle force balance

$$m_p \frac{d\vec{u}_p}{dt} = \vec{F}_{\text{drag}} + \vec{F}_{\text{pressure}} + \vec{F}_{\text{virtual mass}} + \vec{F}_{\text{gravity}} + \vec{F}_{\text{other}}$$

- Particle acceleration

$$\frac{d\vec{u}_p}{dt} = \vec{a} + \beta\left(\vec{u}_f - \vec{u}_p\right)$$

- Particle location

$$\frac{d\vec{x}_p}{dt} = \vec{u}_p$$

Numerics – Particle Trajectory Equations

- Implicit discretization of particle velocity equation

$$\frac{d\vec{u}_p}{dt} \approx \frac{\vec{u}_p^{n+1} - \vec{u}_p^{n}}{\Delta t} = \beta\left(\vec{u}_f^{n} - \vec{u}_p^{n+1}\right) + \vec{a}$$

gives velocity at new location

$$\vec{u}_p^{n+1} = \frac{\vec{u}_p^{n} + \Delta t\left(\vec{a} + \beta\,\vec{u}_f^{n}\right)}{1 + \beta\,\Delta t}$$

- The scheme outlined here is the Euler implicit scheme.

Numerics – Particle Trajectory Equations

- We now know the new particle velocity.
- Discretization of particle location equation yields

$$\frac{d\vec{x}_p}{dt} \approx \frac{\vec{x}_p^{n+1} - \vec{x}_p^{n}}{\Delta t} = \frac{\vec{u}_p^{n+1} + \vec{u}_p^{n}}{2}$$

or,

$$\vec{x}_p^{n+1} = \vec{x}_p^{n} + \frac{\left(\vec{u}_p^{n+1} + \vec{u}_p^{n}\right)\Delta t}{2}$$

Numerics – Particle Trajectory Equations

- Applying the idea of using averaged information also when discretizing the particle velocity equation, implicit trapezoidal discretization yields

$$\frac{d\vec{u}_p}{dt} = a + \beta\left(\vec{u}_f - \vec{u}_p\right)$$

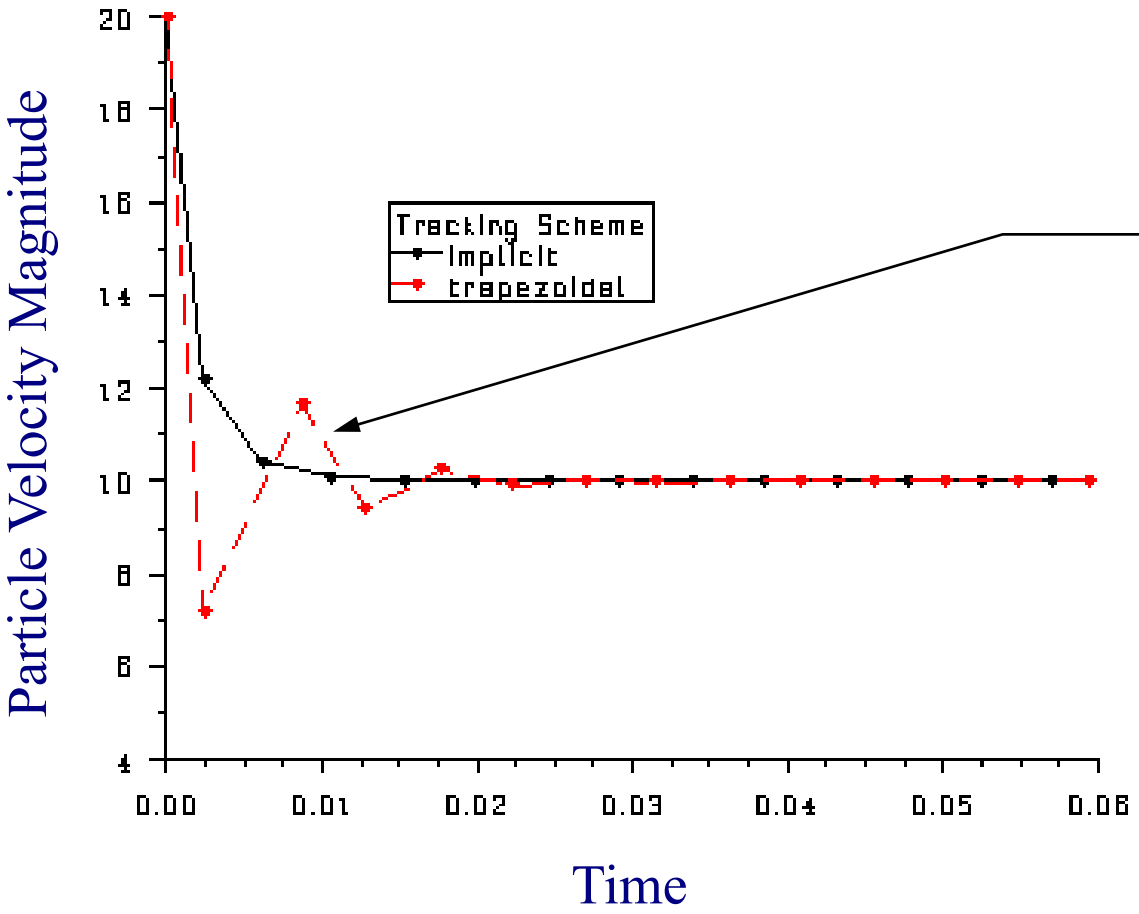
which gives

$$\frac{d\vec{u}_p}{dt} \approx \frac{\vec{u}_p^{\,n+1} - \vec{u}_p^{\,n}}{\Delta t} = \vec{a} + \beta\left(\frac{\vec{u}_f^{\,n+1} + \vec{u}_f^{\,n}}{2} - \frac{\vec{u}_p^{\,n+1} - \vec{u}_p^{\,n}}{2}\right)$$

or,

$$\vec{u}_p^{\,n+1} = \frac{\vec{u}_p^{\,n}\left(1 - \frac{\beta\,\Delta t}{2}\right) + \Delta t\left[\vec{a} + \beta\left(\vec{u}_f^{\,n} + \frac{\Delta t}{2}\vec{u}_p^{\,n}\nabla\vec{u}_f^{\,n}\right)\right]}{1 + \frac{\beta\,\Delta t}{2}}$$

Numerics – Particle Trajectory Equations



Trapezoidal scheme tends to produce overshoots and undershoots for small particles

Numerics – Particle Trajectory Equations

- Stability analysis of simplified equations
  - Euler implicit $\vec{u}_p^{\,n+1} = \dfrac{\vec{u}_p^{\,n}}{1+\beta\,\Delta t}$ — Unconditionally stable
  - Euler explicit $\vec{u}_p^{\,n+1} = \vec{u}_p^{\,n}\left(1-\beta\,\Delta t\right)$ $\quad 1-\beta\,\Delta t > 0 \Rightarrow \boxed{\Delta t < \dfrac{1}{\beta}}$
  - Euler trapezoidal $\vec{u}_p^{\,n+1} = \vec{u}_p^{\,n}\left(\dfrac{1-\dfrac{\beta\,\Delta t}{2}}{1+\dfrac{\beta\,\Delta t}{2}}\right)$ $\quad 1-\dfrac{\beta\,\Delta t}{2} > 0 \Rightarrow \boxed{\Delta t < \dfrac{1}{2\beta}}$

Numerics – Particle Trajectory Equations

- Exponential analytical solution
  - Particle velocity equation

$$\vec{u}_p^{\,n+1} = \vec{u}_f^{\,n} + \left(\vec{u}_p^{\,n} - \vec{u}_f^{\,n}\right)e^{-\beta\Delta t} + \frac{\vec{a}}{\beta}\left(1 - e^{-\beta\Delta t}\right)$$

  - Particle location equation

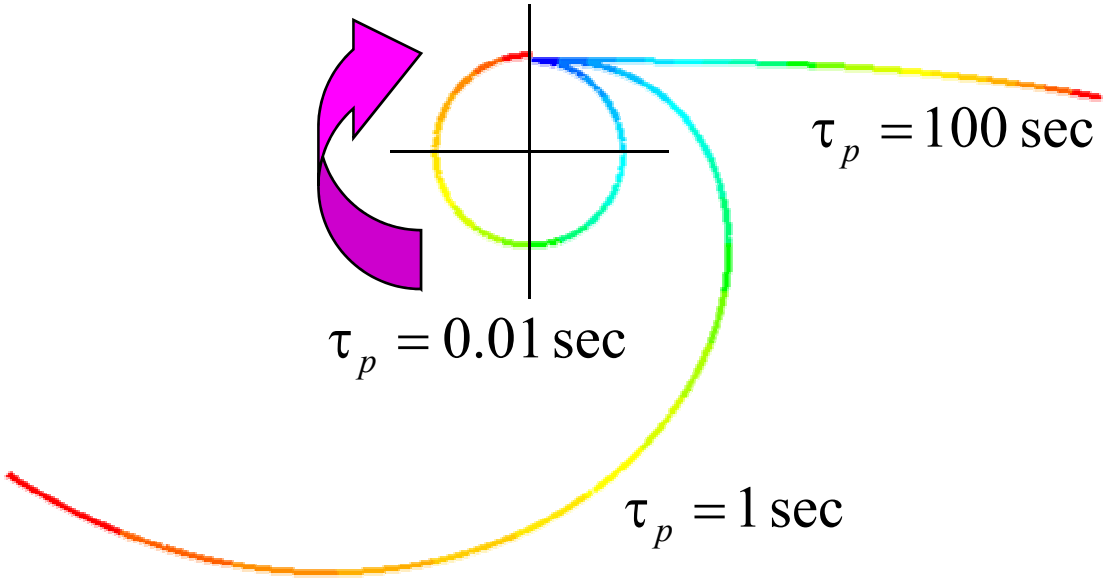
$$\vec{x}_p^{\,n+1} = \vec{x}_f^{\,n} + \vec{u}_f^{\,n}\,\Delta t + \frac{1}{\beta}\left(\vec{u}_p^{\,n} - \vec{u}_f^{\,n}\right)\left(1 - e^{-\beta\Delta t}\right)$$

$$+ \frac{\vec{a}}{\beta}\left[\Delta t + \frac{1}{\beta}\left(\vec{u}_p^{\,n} - \vec{u}_f^{\,n}\right)\left(1 - e^{-\beta\Delta t}\right)\right]$$

- Runge-Kutta scheme

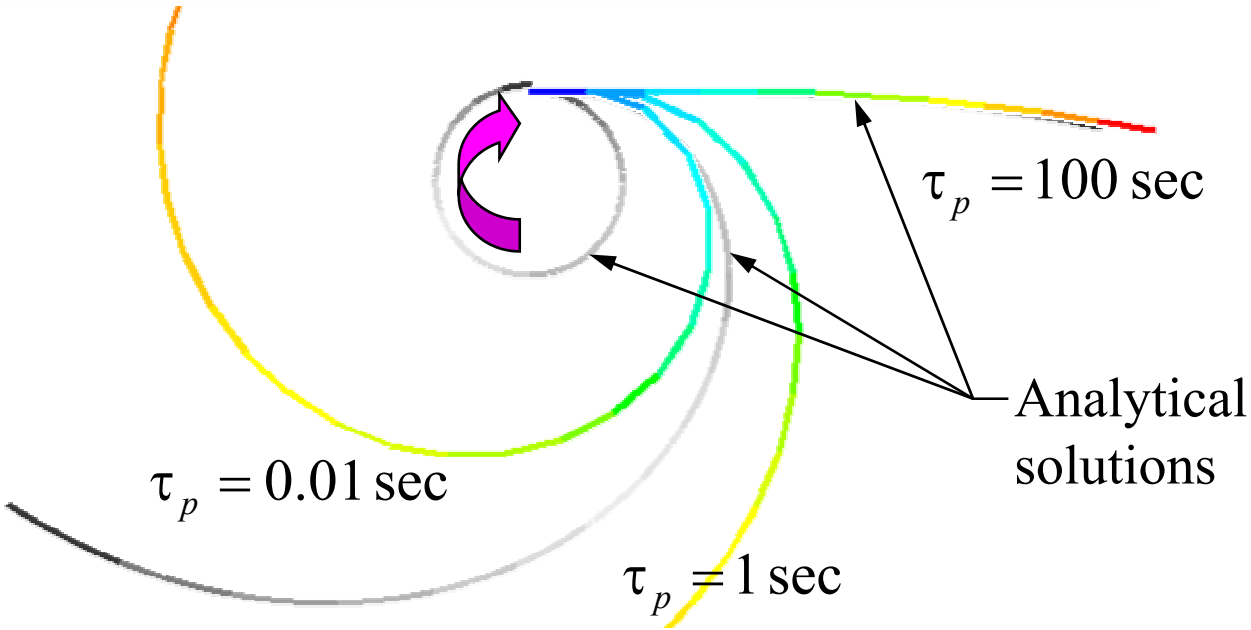
Numerics – Particle Tracking Schemes

- Euler implicit
- Euler trapezoidal
- Exponential analytic
- Runge-Kutta



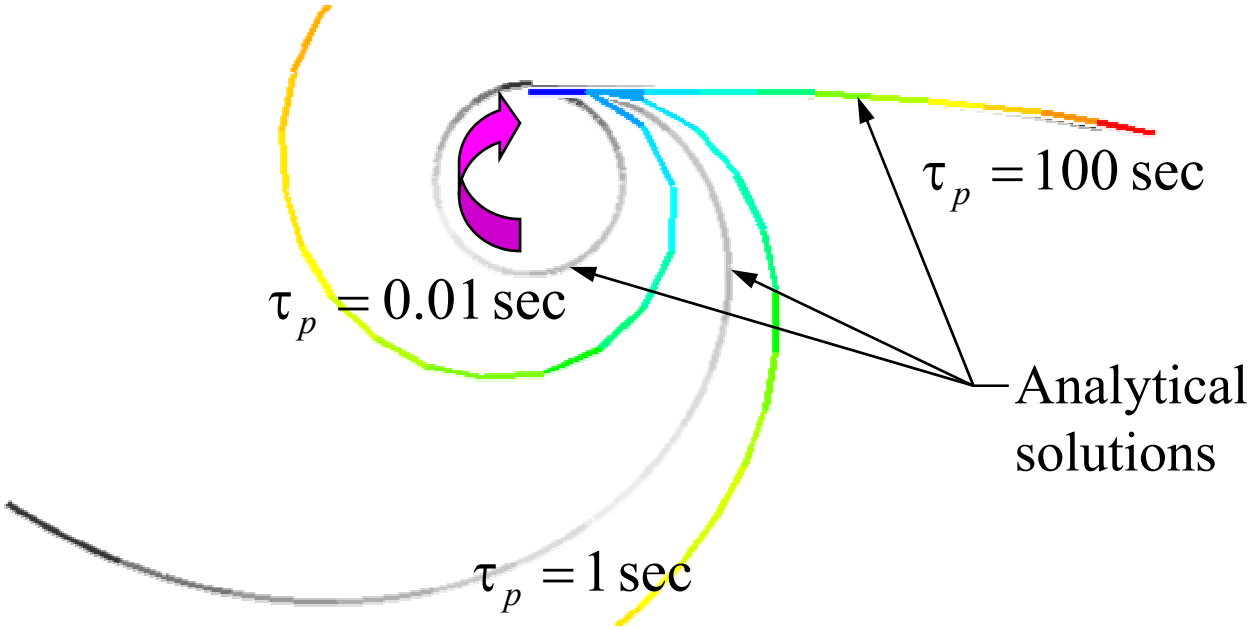
Numerics – Effect of Tracking Scheme

- Euler Implicit



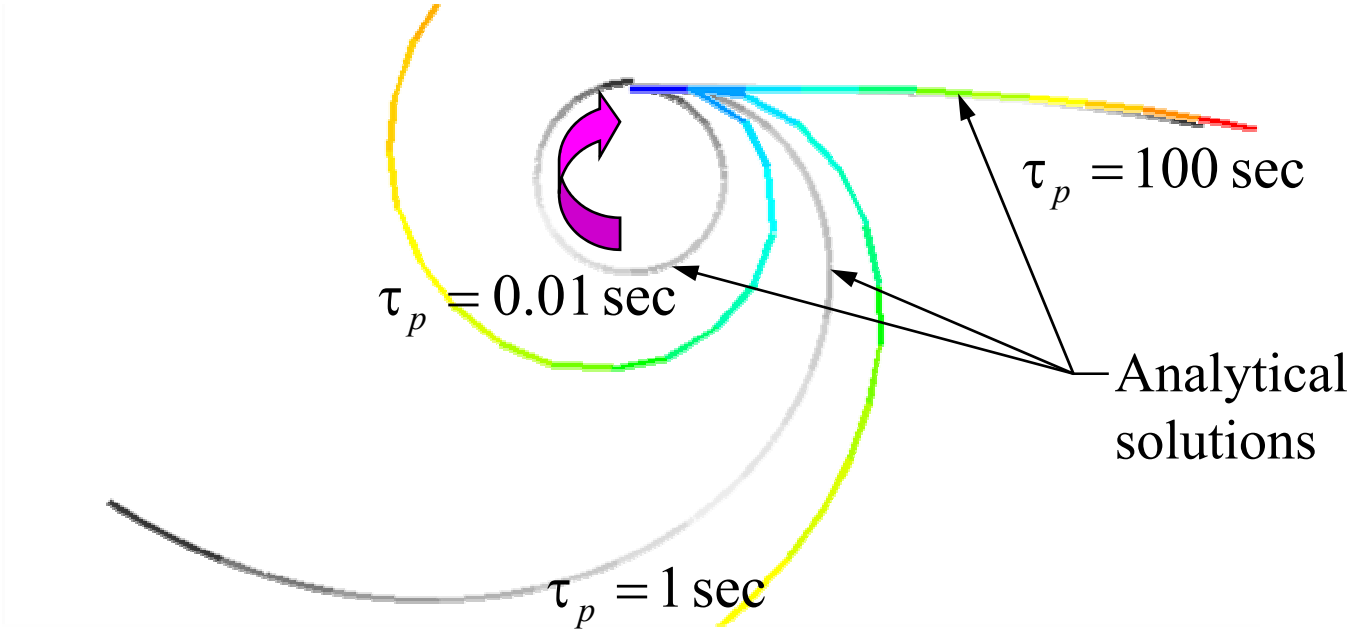
Numerics – Effect of Tracking Scheme

- Exponential Analytic



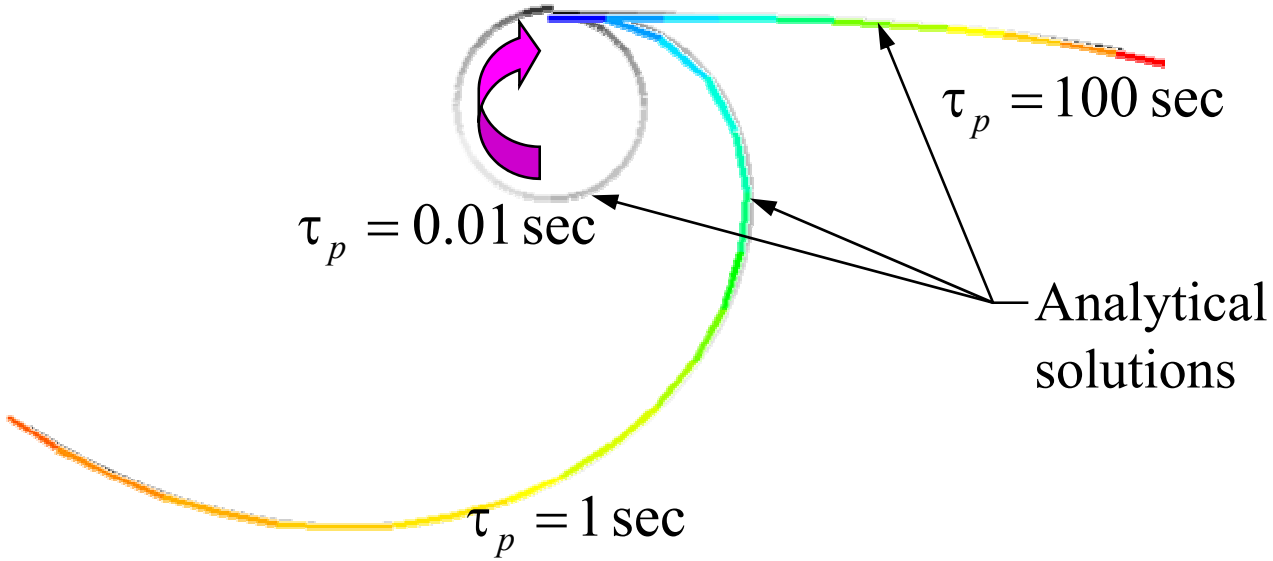
Numerics – Effect of Tracking Scheme

- Euler Trapezoidal



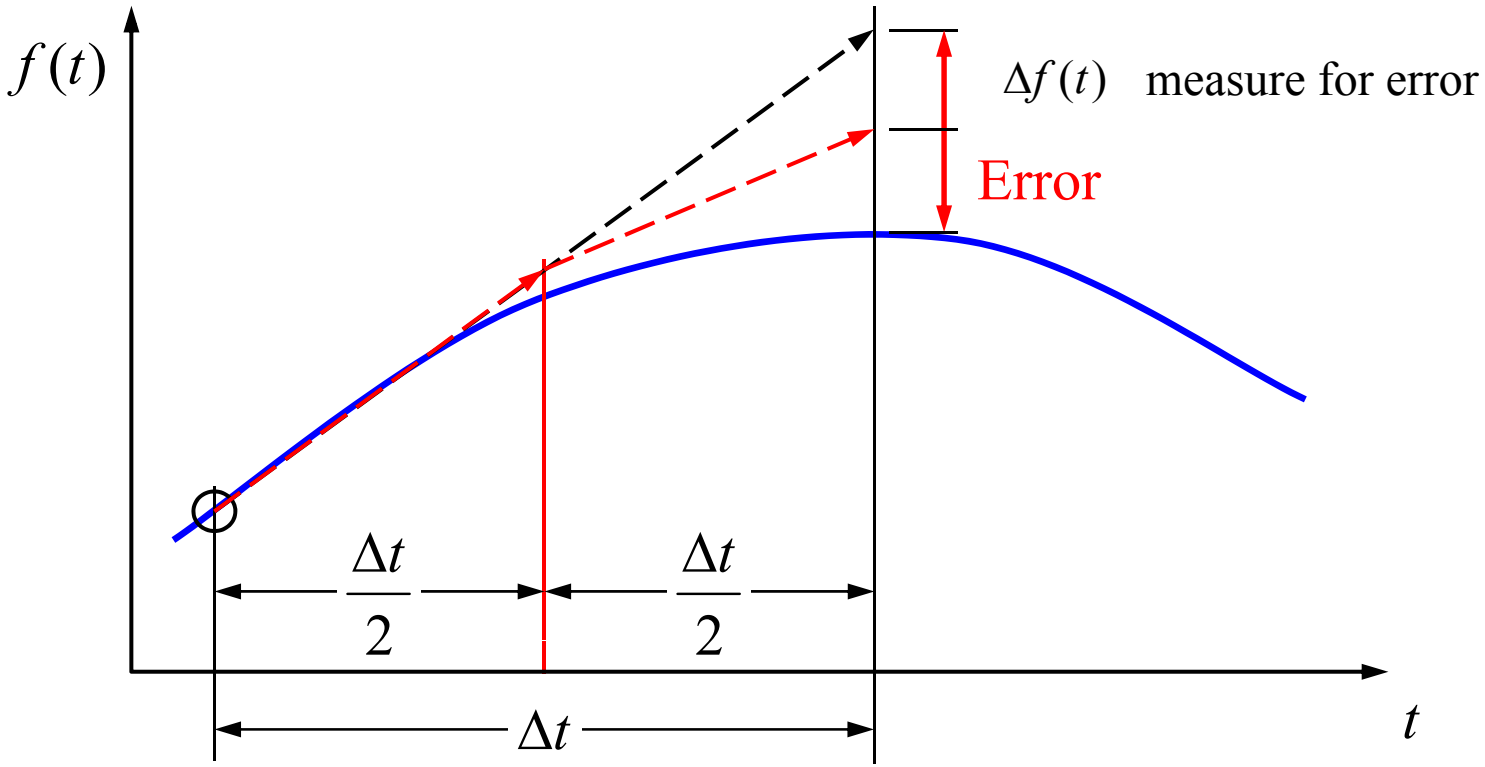
Numerics – Effect of Tracking Scheme

- Runge-Kutta



Numerics – How to Increase Accuracy

- Decrease the time step until the desired accuracy is achieved.



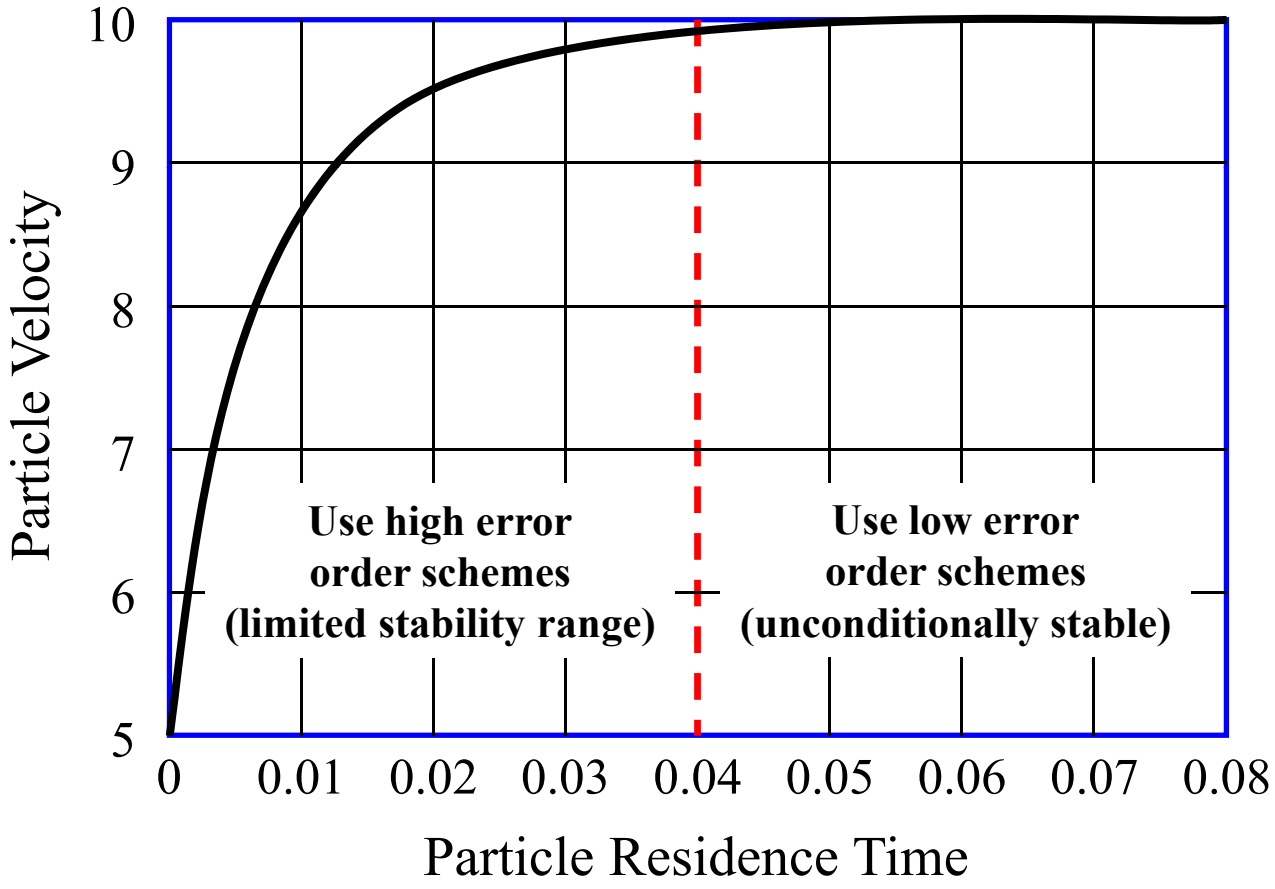
Numerics – How to Increase Accuracy

- Euler implicit scheme
  - Needs many refinement steps to achieve accuracy.
  - The reason for this is the scheme's low order of truncation error.
- Euler trapezoidal scheme
  - Converges quickly to a given accuracy due to order $O(\Delta t^2)$.
  - Shows limited stability range.
- Exponential analytical scheme
  - Same as Euler trapezoidal but without step size limitation.
- Runge-Kutta scheme
  - Embedded step size control.
  - Limitation on step size.

Numerics – How to Further Improve

- Automated tracking scheme selection
  - Stability intervals exist for Euler trapezoidal and Runge-Kutta schemes.
  - Euler implicit and exponential analytic are unconditionally stable.
  - Selection strategy
    - Use low order schemes where possible and if chosen error allows.
    - Use higher-order schemes when the integration step lies within the stability range of the scheme.
  - Available in conjunction with error-based integration step control.

Numerics – How to Further Improve



Numerics – Energy and Mass Equation

- Energy equation

$$m_p C_p \frac{dT_p}{dt} = \pi d_p^2 h \left(T_\infty - T_p\right) + \sum \dot{m}_i \Delta h_v + \pi d_p^2 \varepsilon_p \sigma \left(\theta_R^4 - T_p^4\right)$$

- Mass equation

$$\frac{dm_p}{dt} = \sum \dot{m}_i = -\pi d_p^2 \sum M_{w,i} k_c \left(\frac{p_{\text{sat}}\left(T_p\right)}{RT_p} - C_{i,\infty} \right)$$

- Two schemes
- Analytical integration in subsequent manner
- Coupled heat mass solution