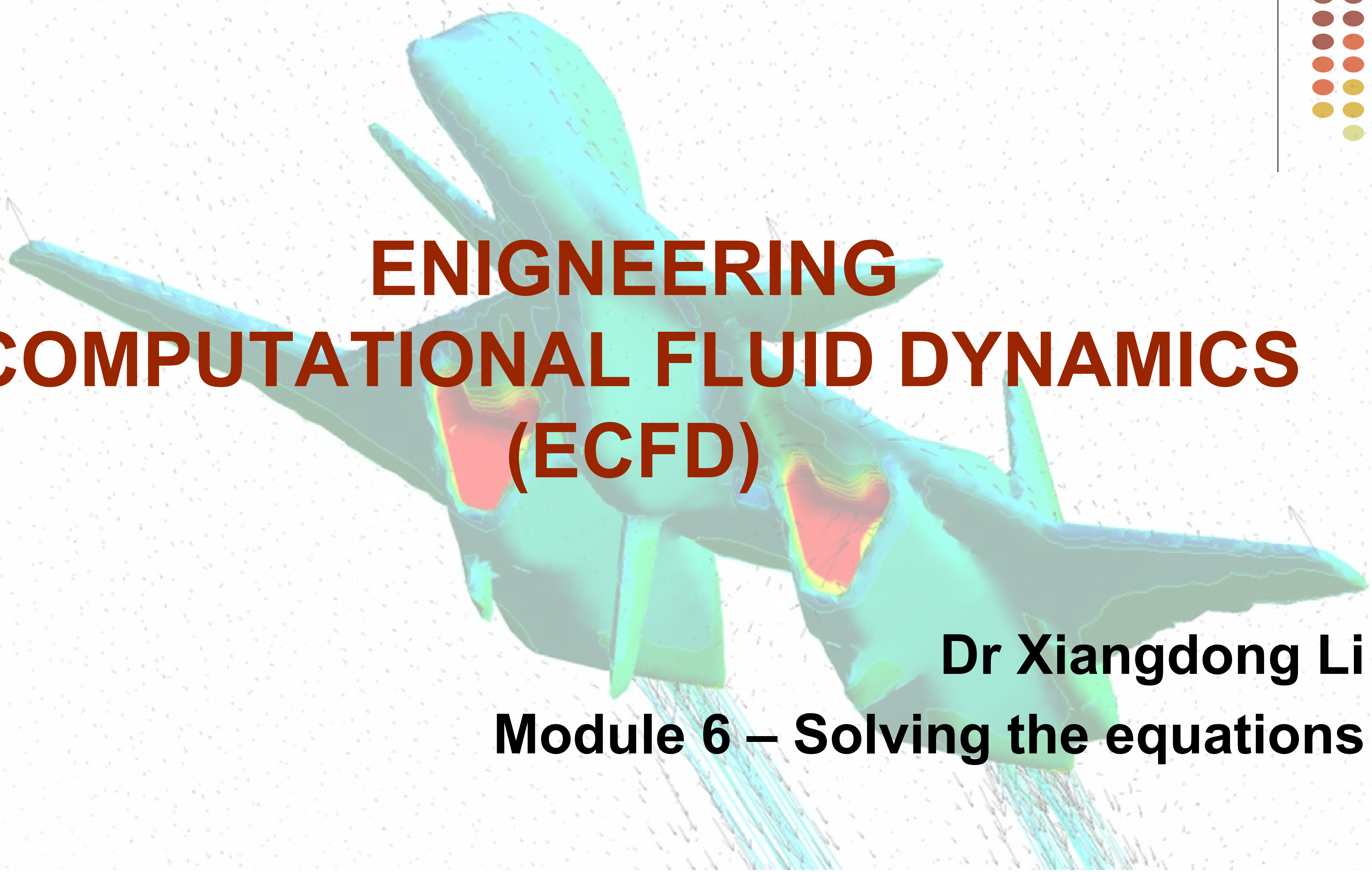
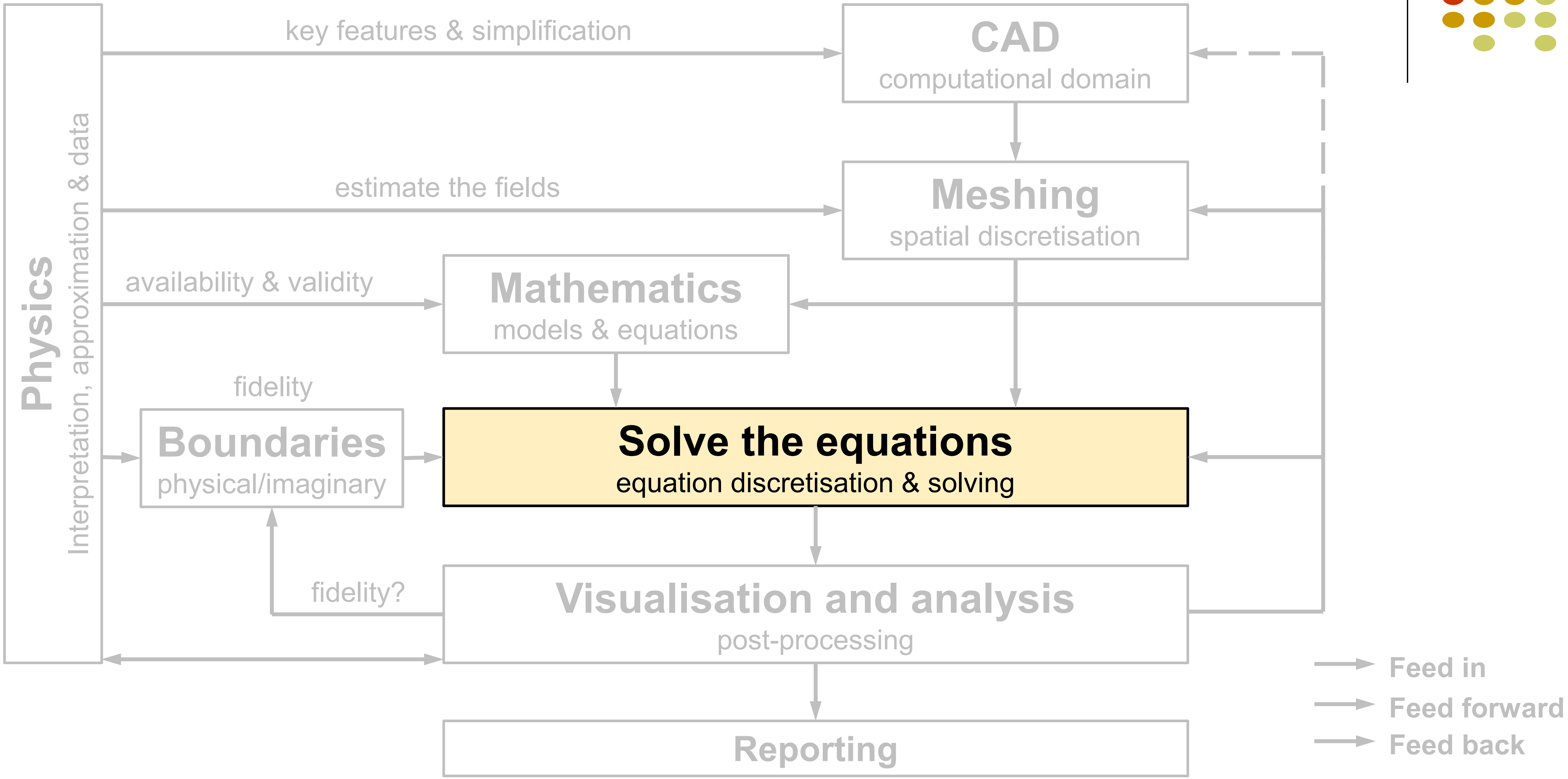


ENIGNEERING COMPUTATIONAL FLUID DYNAMICS (ECFD)

Dr Xiangdong Li

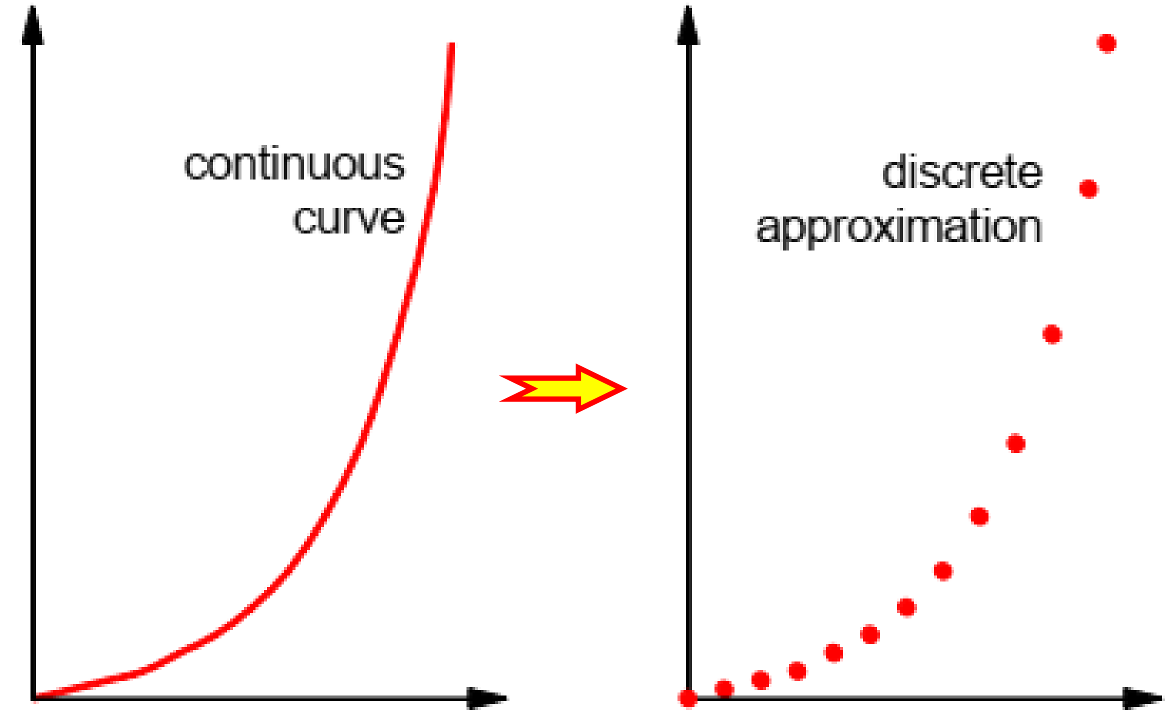
Module 6 – Solving the equations

CFD workflow



Discretisation

Field variables:



This module

- How are pressure and density calculated in CFD codes?
- How are the partial differential equations solved? – the finite volume method (FVM)
- Strategies for coping with stability issues
- Convergence

EQUATION DISCRETISATION

In Module 1

- L. Euler (1707-1783), the Euler equation
- C.L. Navier (1785-1836) and G.G. Stokes (1819-1903), the Navier-Stokes (N-S) equations – $US 1 million prize offered by the Clay Mathematics Institute for an analytical solution
- Lewis F. Richardson (FDM, 1911), A. Thom (1933), …
- Harlow and Fromm, 1965, “computer experiments” – **the birth of CFD**
- Los Alamos Lab, 1960’s, PIC algorithm, … Digital Computers were used to solve the N-S equations, successfully.
- McDonald (1971) and MacCormack and Paullay (1972), the finite volume method
- S.V. Patanka & D.B. Spalding, 1970 – 1980’s, the k-ε model, SIMPLE algorithm, up-wind differencing, …
- CHM Ltd. (UK), 1981, Phoenix – the world’s first general-purpose commercial CFD package
- Exponential growth since 1980’s: Fluent, CFX, OpenFoam, SimScale, …

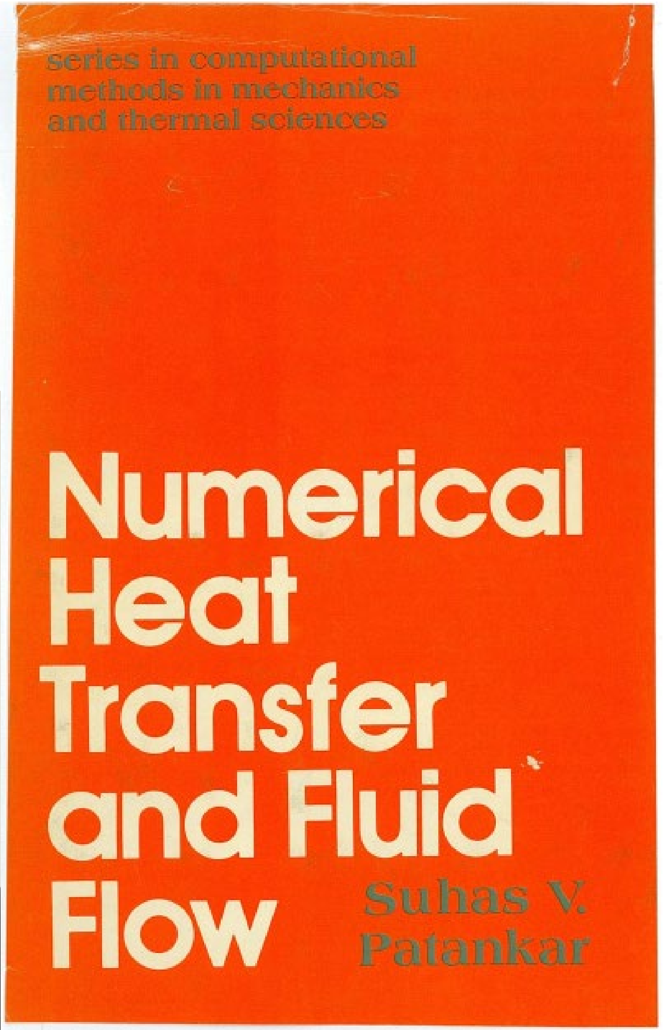
Manual Fluid Dynamics

Computational Fluid Dynamics

In Module 1 – Discrete numerical methods

- ❖ Boundary element method
- ❖ Spectral methods
- ❖ Finite difference method
- ❖ Finite volume method
- ❖ Finite element method
- ❖ Vorticity based methods
- ❖ And more

Published in 1980

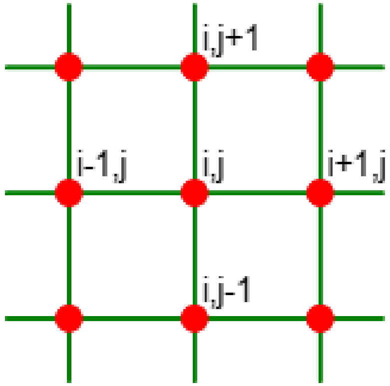


S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flow. Int J Heat Mass Trans, 15, 1972: 1787-1806

FDM vs FVM

- **Finite-difference:**
  - discretise differential equations

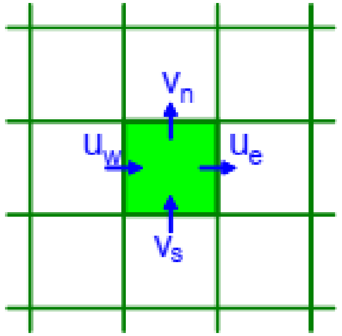
$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y}$$



Considering the change at a point

- **Finite-volume:**
  - discretise control-volume equations

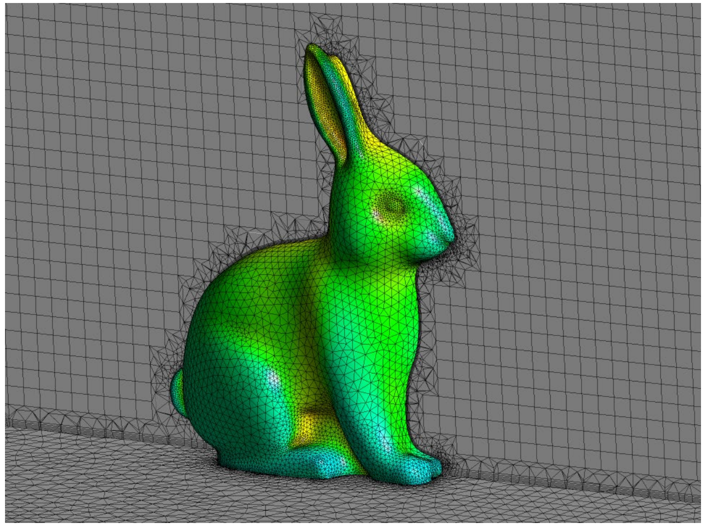
$$0 = net\ mass\ outflow = (\rho u A)_e - (\rho u A)_w + (\rho v A)_n - (\rho v A)_s$$



Considering the change in a volume

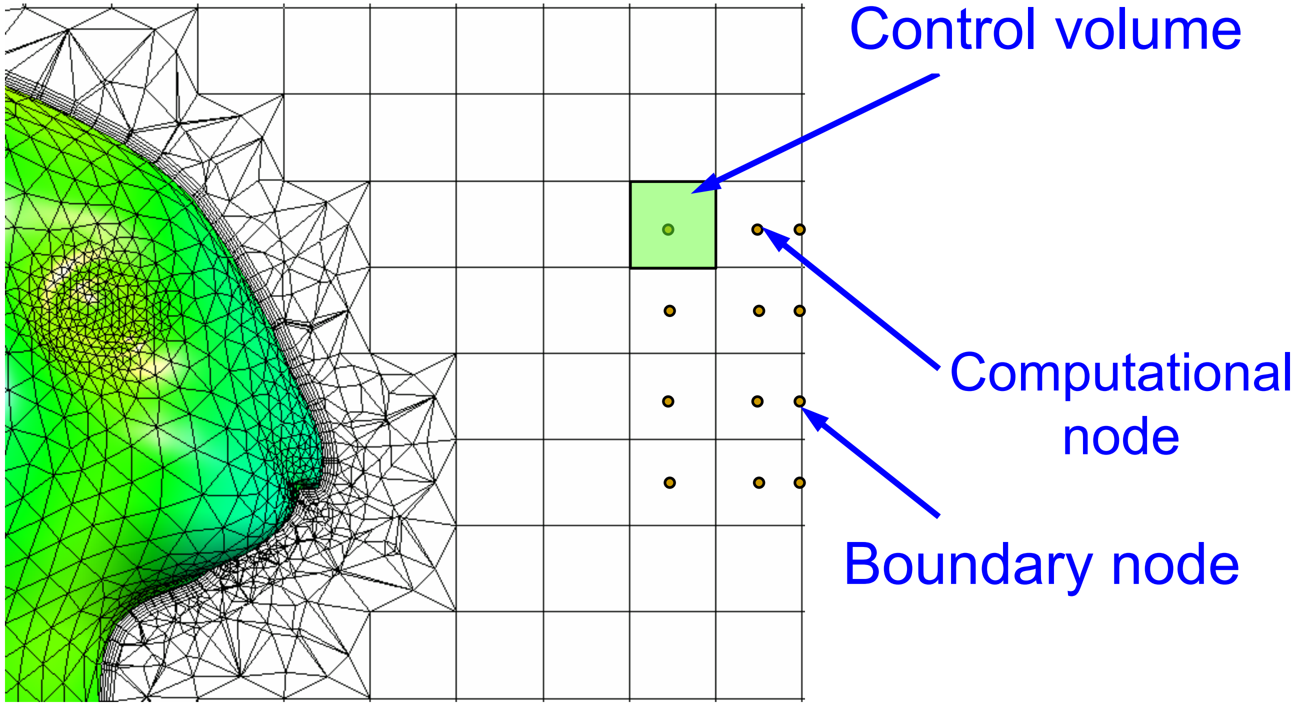
The finite volume method (FVM)

- The most common in commercial CFD packages (www.bakker.org)
  - The FVM – 80%
  - The finite element method – 15%
  - Other (including the FDM) – 5%
- First documented by Evans and Harlow (1957) at Los Alamos
- Robust when variables may not be continuously differentiable
- Advantage in memory use and speed for very large problems
- Not limited to cell shape
- False diffusion possible

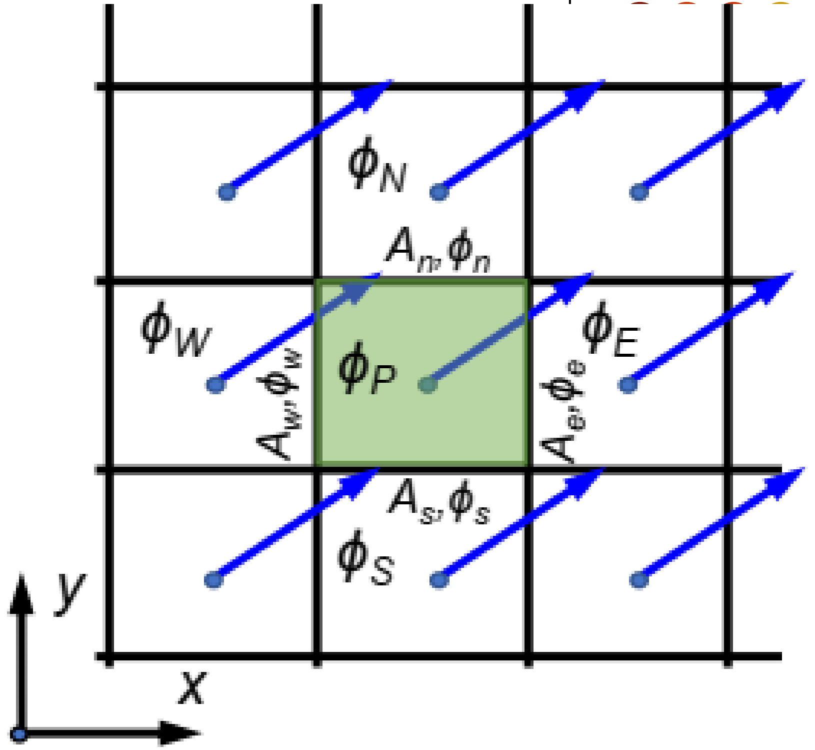


The FVM

- Discretise the domain into control volumes (cells/mesh elements)
- Integrate the differential equations over the control volume and its faces
  - Cells and nodes
- Values at control volume faces are need to evaluate the derivatives – interpolation.
- Results in a set of linear algebraic equations, one for each control volume



The FVM – Equation discretisation



- For a conserved variable ϕ, the partial differential conservation equation over the control volume P can be discretised to

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \left(\vec{U} \phi \right) = \nabla \cdot \left(\Gamma \nabla \phi \right) + S_\phi$$

$$A_e u_e \phi_e - A_w u_w \phi_w + A_n v_n \phi_n - A_s v_s \phi_s =$$
$$\Gamma A_e \left(\phi_E - \phi_P \right) / \delta x_e - \Gamma A_w \left(\phi_P - \phi_W \right) / \delta x_e$$
$$+ \Gamma A_n \left(\phi_N - \phi_P \right) / \delta y_n - \Gamma A_s \left(\phi_P - \phi_S \right) / \delta y_s + S_P$$

$$a_P \phi_P = \sum a_{nb} \phi_{nb} + b$$

- Then the value of ϕ at the cell faces will be found through **interpolation**.

Interpolation

- ❖ First-order upwind scheme
- ❖ Central differencing scheme
- ❖ Power-law scheme
- ❖ Second-order upwind scheme
- ❖ QUICK scheme

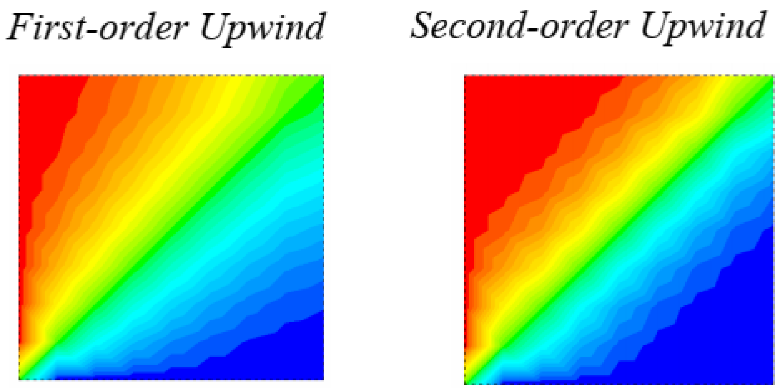
$$\phi_e = \phi_P + \frac{d\phi}{dx}\left(x_e - x_P\right) + \frac{d^2\phi}{dx^2}\frac{\left(x_e - x_P\right)}{2!} + \ldots + \frac{d^n\phi}{dx^n}\frac{\left(x_e - x_P\right)}{n!} + \ldots$$

- First-order upwind: 1st term in RHS
- Central and 2nd-order: 2 first terms in RHS
- QUICK: 3 first terms in RHS

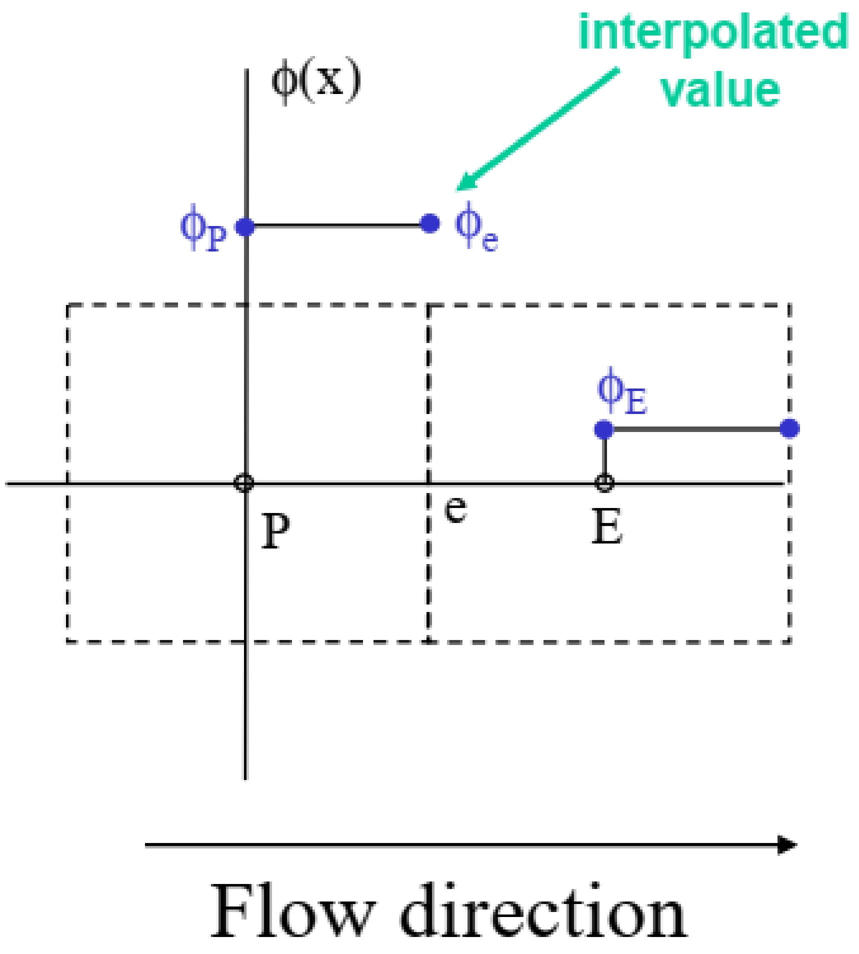
Interpolation – 1st-order upwind

❖ First-order upwind scheme

- Very stable
- Best scheme to start with
- Very diffusive: gradients can be smeared out – numerically induced diffusion, also called "**false diffusion**"



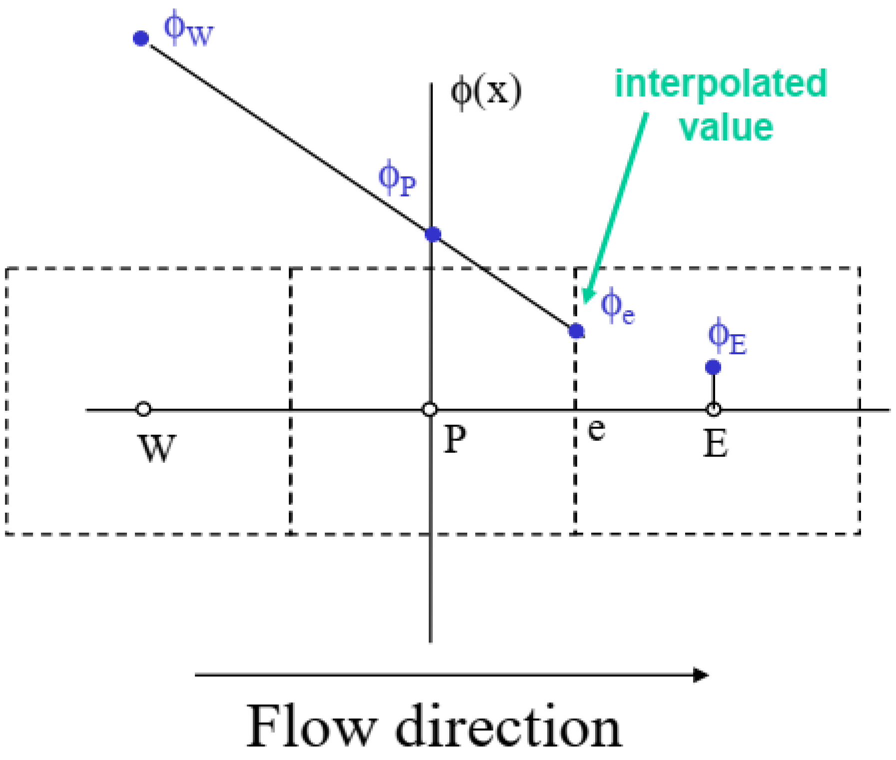
False diffusion



Interpolation – 2nd-order upwind

❖ Second-order upwind scheme

- Better than 1st-order upwind
- Sometimes demands limit

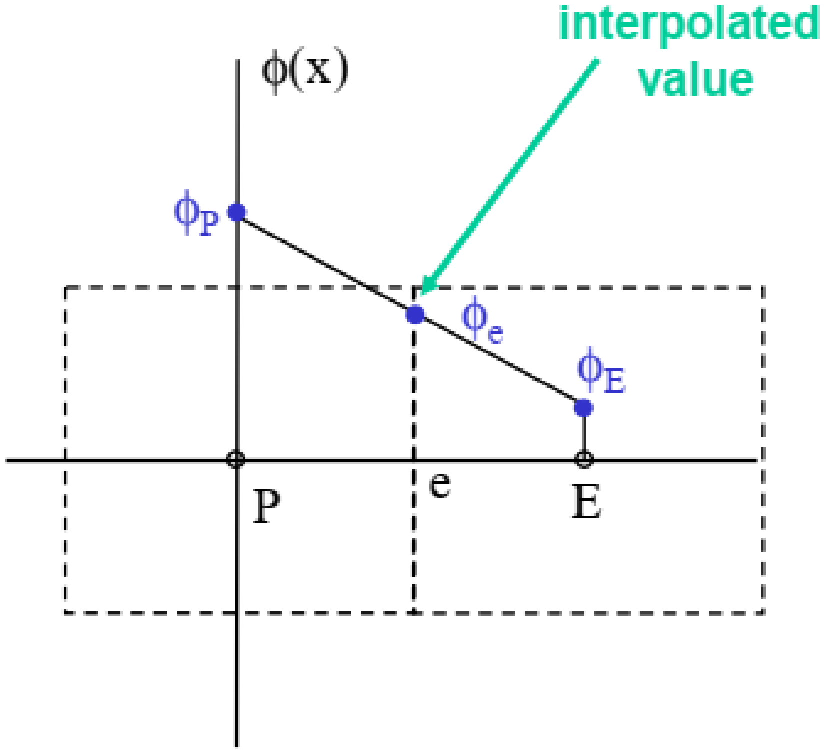


Interpolation – central differencing

❖ Central differencing scheme

- More accurate than 1st-order upwind scheme
- Leads to oscillation when the Peclet number (Pe) is larger than 2 (often switch to 1st-order upwind scheme when Pe > 2)
- Pe is the ratio between convective and diffusive transport

$$Pe = \frac{\rho u L}{D}$$

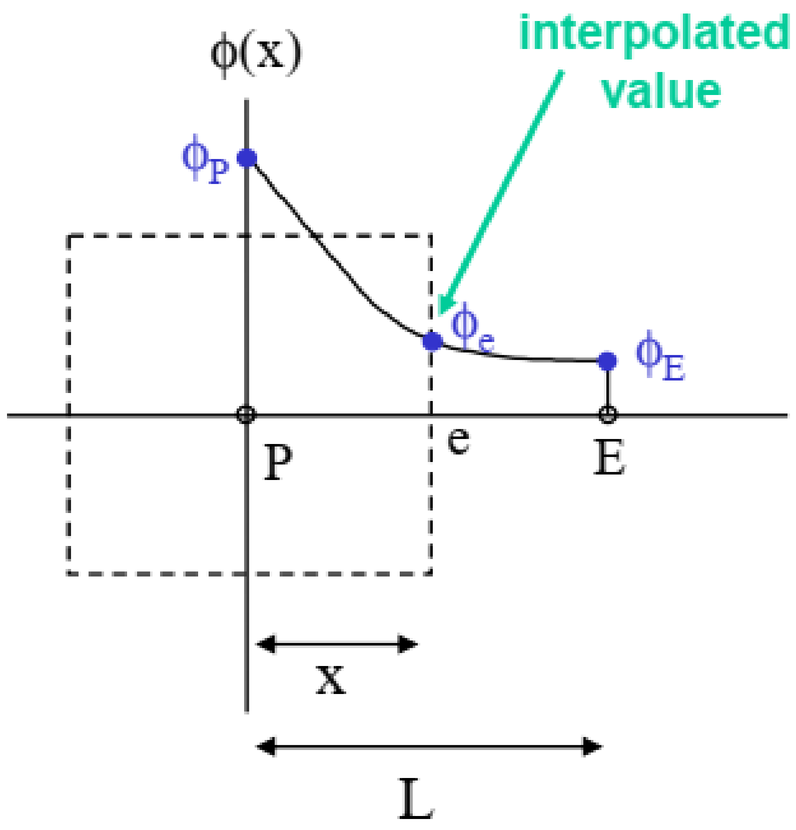


Interpolation – power law

❖ Power-law scheme

- A theoretical solution
- Switch to 1st-order upwind scheme when Pe > 10

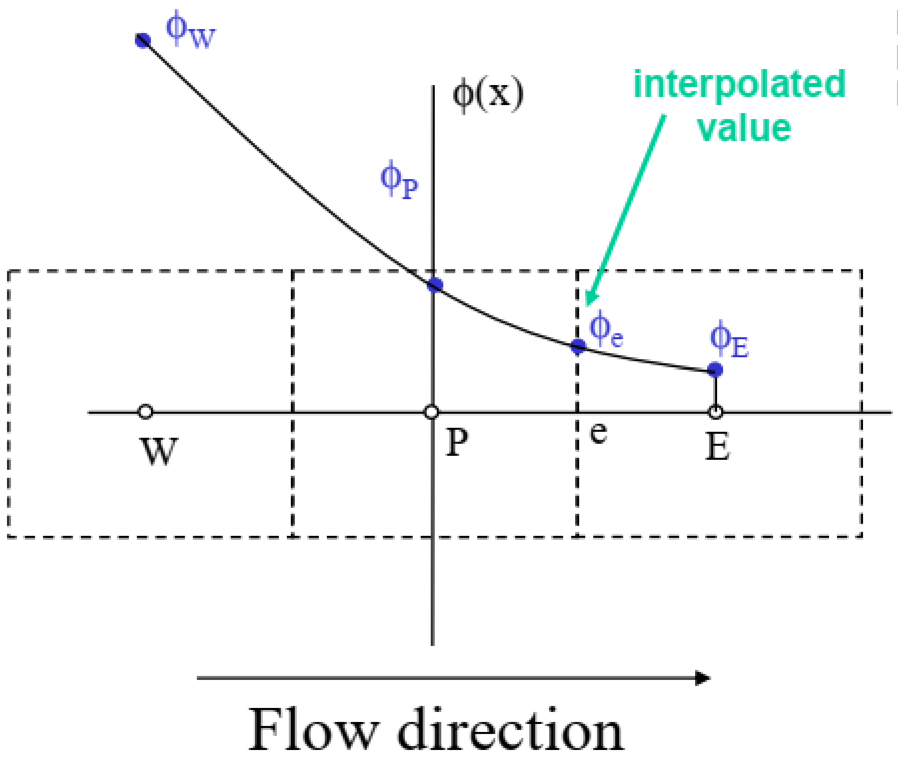
$$\phi_e = \phi_P - \frac{(1-0.1Pe)^5}{Pe}(\phi_E - \phi_P)$$



Interpolation – QUICK

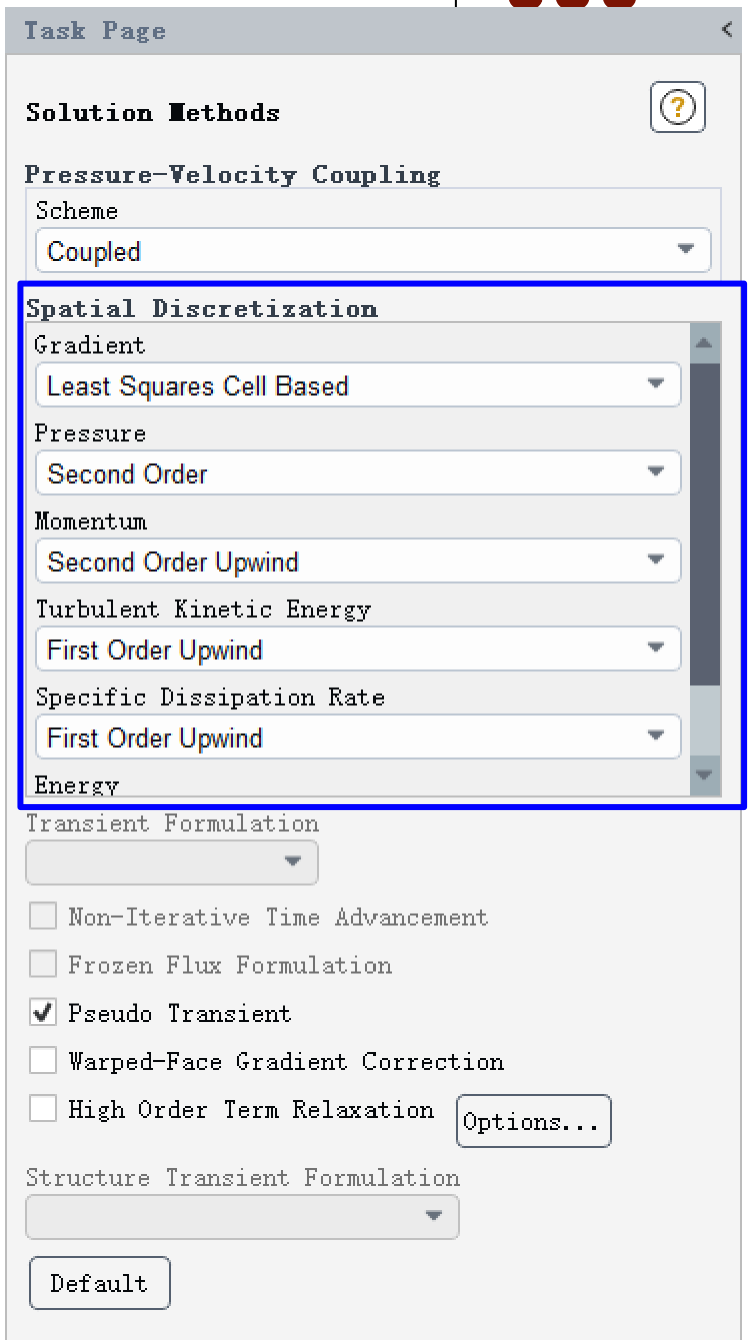
❖ QUICK scheme

- Quadratic Upwind Interpolation for Convective Kinetics
- Very accurate
- But can lead to stability problems in high-gradient regions



Interpolation - recommendations

- Higher-order schemes are more accurate, but less stable
- Start with 1st-order upwind scheme then switch over to 2nd-order upwind scheme
- Central differencing scheme should only be used for transient LES where mesh is fine enough to ensure $Pe<1$
- Power law and QUICK schemes are only used for especially suitable problems



PRESSURE, VELOCITY AND DENSITY

The issue of pressure

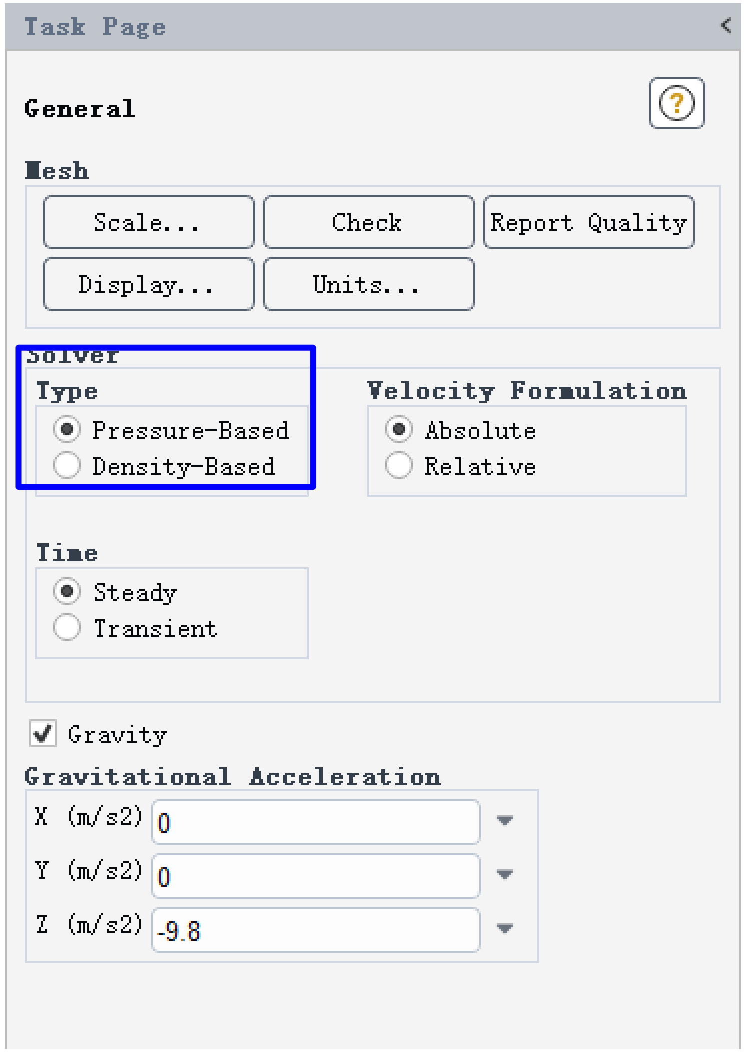
- Think about the continuity equation and momentum equation: we have the momentum equations for *u*, *v*, *w*, and we have the continuity equation for constraining the mass conservation; however, we don't have an equation for *p*.

$$\frac{\partial \rho}{\partial t} + \vec{U} \cdot \nabla \rho = 0$$

$$\frac{\partial}{\partial t}\left(\rho \vec{U}\right) + \nabla \cdot \left(\rho \vec{U}\vec{U}\right) = -\nabla p + \nabla \cdot \left(\mu\left(\nabla \vec{U} + \left(\nabla \vec{U}\right)^T\right)\right) - \nabla \cdot \left(\frac{2}{3}\mu\left(\nabla \cdot \vec{U}\right)\right) I + S_M$$

- What's the solution?

Solutions in Fluent



- Pressure-based solver (historically for low-speed incompressible flows): The pressure field is extracted by deriving and solving **a pressure (correction) equation** from the continuity and momentum equations.
- Density-based solver (historically for high-speed compressible flows): The continuity equation is used to obtain the density field while the pressure field is determined form the equation of state.
- ATTN: now the both solvers can be applied to most cases although the density-based solver has advantage for compressible flows.

The pressure-based solver

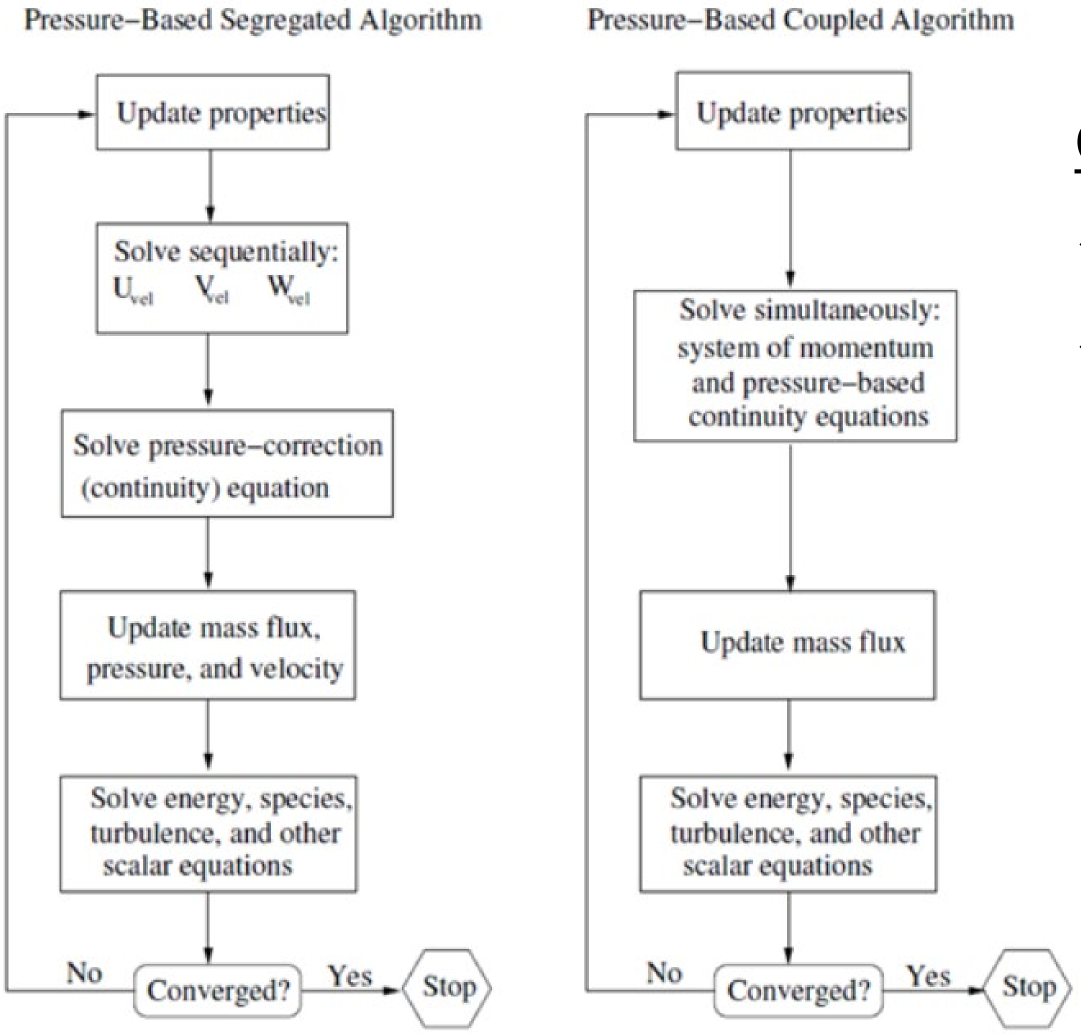
Segregated algorithm

- SIMPLE
- SIMPLEC
- PISO

❖ Memory-efficient

❖ Faster per iteration

❖ Slow convergence



Ansys Fluent User's Guide 2020R1

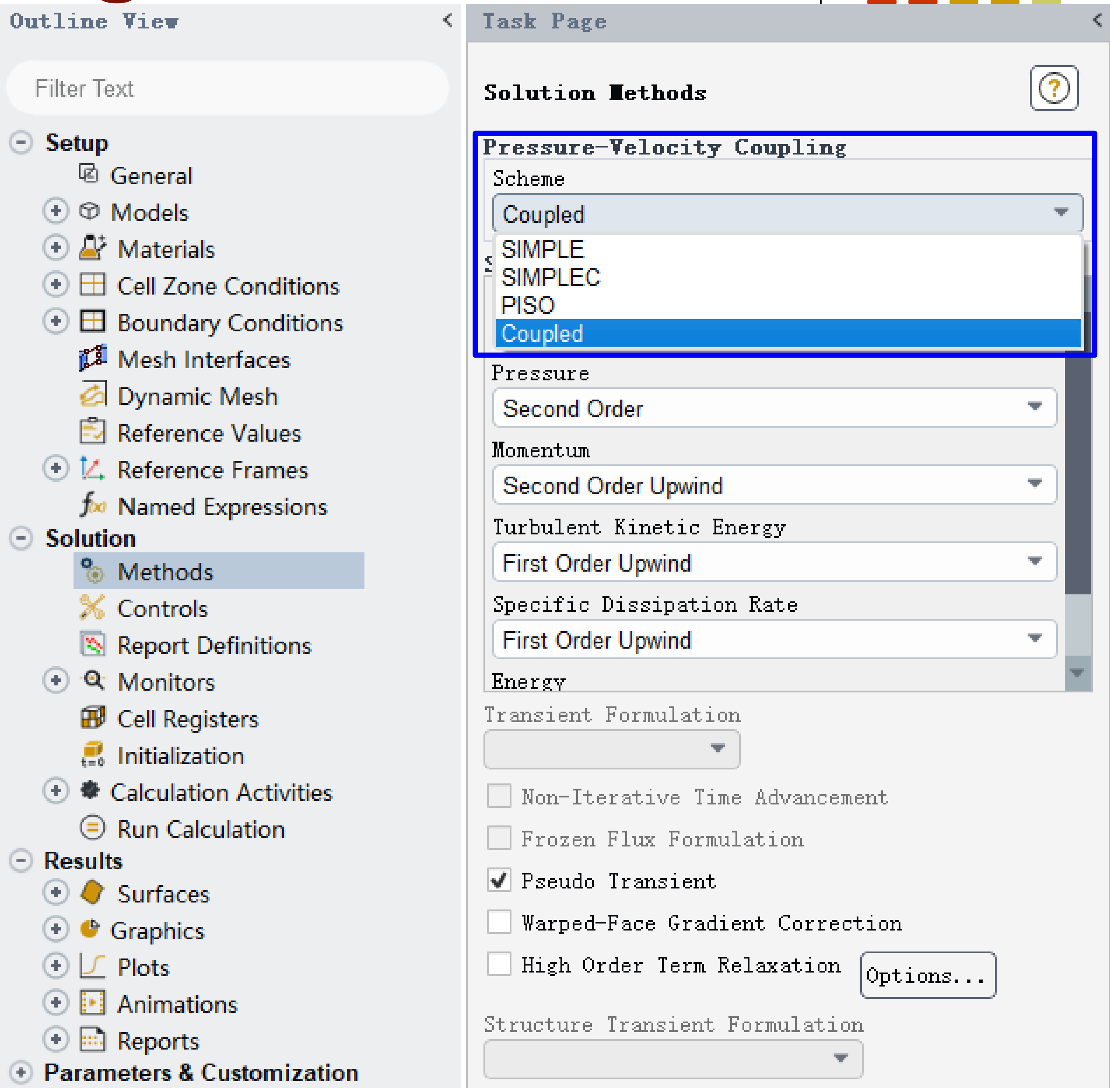
Coupled algorithm

❖ Memory demanding

❖ Fast convergence

Velocity-pressure coupling in Fluent

- SIMPLE – The default scheme, robust
- SIMPLEC – Allows faster convergence than SIMPLE for simple problems, all under-relaxation
- PISO – Useful for unsteady flows or for mesh containing cells with higher than average skewness
- FSM – For unsteady flows only
- Coupled – Recommended for steady-state flows though robust for transient flows



Operating/reference pressure

- For low speed flows, the pressure drop is small compared to the static pressure (e.g., 1 Pa / 101325 Pa) – round-off error can affect the solution.
- By defining an operating/reference pressure p_0, we can only focus on the gauge pressure – higher resolution
- Gauge pressure ($p_g = p - p_0$) is used in the momentum equation

$$\frac{\partial}{\partial t}\left(\rho\vec{U}\right)+\nabla\cdot\left(\rho\vec{U}\vec{U}\right)=-\nabla p+\nabla\cdot\left(\mu\left(\nabla\vec{U}+\left(\nabla\vec{U}\right)^{T}\right)\right)-\nabla\cdot\left(\frac{2}{3}\mu\left(\nabla\cdot\vec{U}\right)\right)I+S_M$$

$$\Downarrow$$

$$\frac{\partial}{\partial t}\left(\rho\vec{U}\right)+\nabla\cdot\left(\rho\vec{U}\vec{U}\right)=-\nabla p_g+\nabla\cdot\left(\mu\left(\nabla\vec{U}+\left(\nabla\vec{U}\right)^{T}\right)\right)-\nabla\cdot\left(\frac{2}{3}\mu\left(\nabla\cdot\vec{U}\right)\right)I+S_M$$

Density – pressure and density

- For incompressible or mildly compressible flows, we can define a reference pressure $p_0 >> p_g$, the density only depends on the temperature (incompressible idea gas law)

$$\rho = \frac{p_0 + p_g}{\left(R_U / M\right) T} \approx \frac{p_0}{\left(R_U / M\right) T}$$

- For compressible flows, the gauge pressure is the absolute pressure (reference pressure p_0 is 0) and the density is a function of both pressure and temperature

$$\rho = \frac{p_0 + p_g}{\left(R_U / M\right) T}$$

Density – Reference density

- At low flow speeds, if the buoyance force is large compared to others, it can make the solution unstable.

$$\frac{\partial}{\partial t}\left(\rho\vec{U}\right)+\nabla\cdot\left(\rho\vec{U}\vec{U}\right)=-\nabla p+\nabla\cdot\left(\mu\left(\nabla\vec{U}+\left(\nabla\vec{U}\right)^{T}\right)\right)-\nabla\cdot\left(\frac{2}{3}\mu\left(\nabla\cdot\vec{U}\right)\right)I+\boxed{\rho g}$$

- A reference density is introduced to alleviate the buoyance force and improve the solution stability

$$\frac{\partial}{\partial t}\left(\rho\vec{U}\right)+\nabla\cdot\left(\rho\vec{U}\vec{U}\right)=-\nabla p+\nabla\cdot\left(\mu\left(\nabla\vec{U}+\left(\nabla\vec{U}\right)^{T}\right)\right)-\nabla\cdot\left(\frac{2}{3}\mu\left(\nabla\cdot\vec{U}\right)\right)I+\boxed{\left(\rho-\rho_{0}\right)g}$$

- Reference density ρ_0 is usually set as the background density, but some CFD codes calculate it automatically as the average value of all mesh elements.

Density – the Boussinesq approx.

- Incompressible flow with small temperature variation, where the density is correlated to the thermal expansion coefficient and temperature rise relative to a reference temperature

$$\rho - \rho_0 = -\beta \rho_0 \left(T - T_0 \right)$$

- No reference density is needed

UNDER-RELAXATION

Oscillation and divergence are long-unsolved problems of CFD

Common methods for under-relaxation

- Under-relaxation factors (Explicit)
- Pseudo transient under-relaxation (Implicit)
- Courant number for transient flows

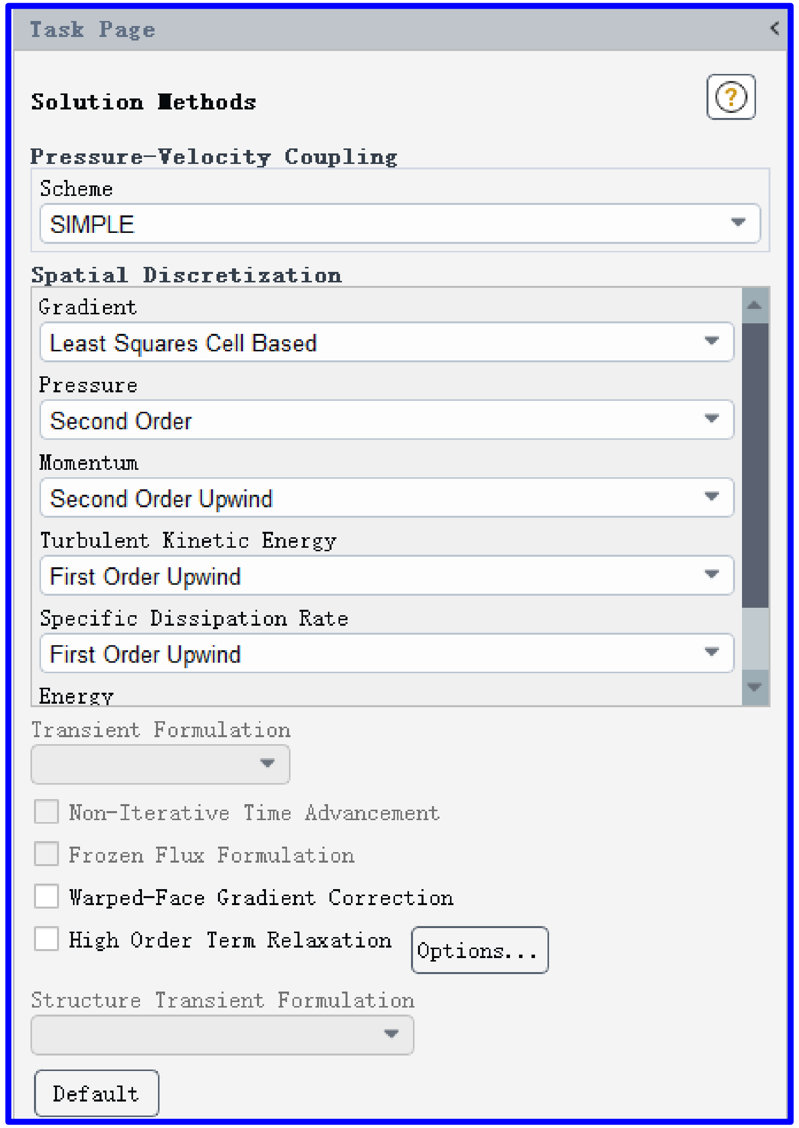
Relaxation

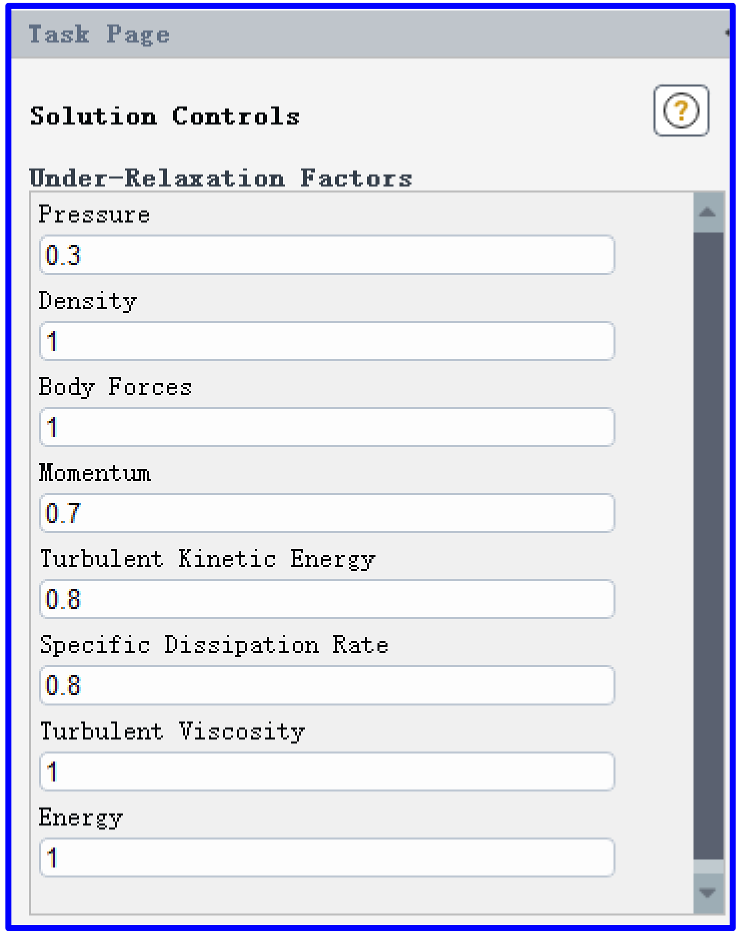
- The equations are solved iteratively: value of ϕ at cell P is recalculated from the value of last iteration.
- It is common to apply a relaxation fact U

$$\phi_P^{new,used} = \phi_P^{old} + U\left(\phi_P^{new,predicted} - \phi_P^{old}\right)$$

- U<1 is underrelaxation. This slows down convergence but increases stability.
- U=1, no relaxation. The predicted value will be used.
- U>1, overrelaxation. This accelerate convergence but decreases stability.

Under-relaxation factors





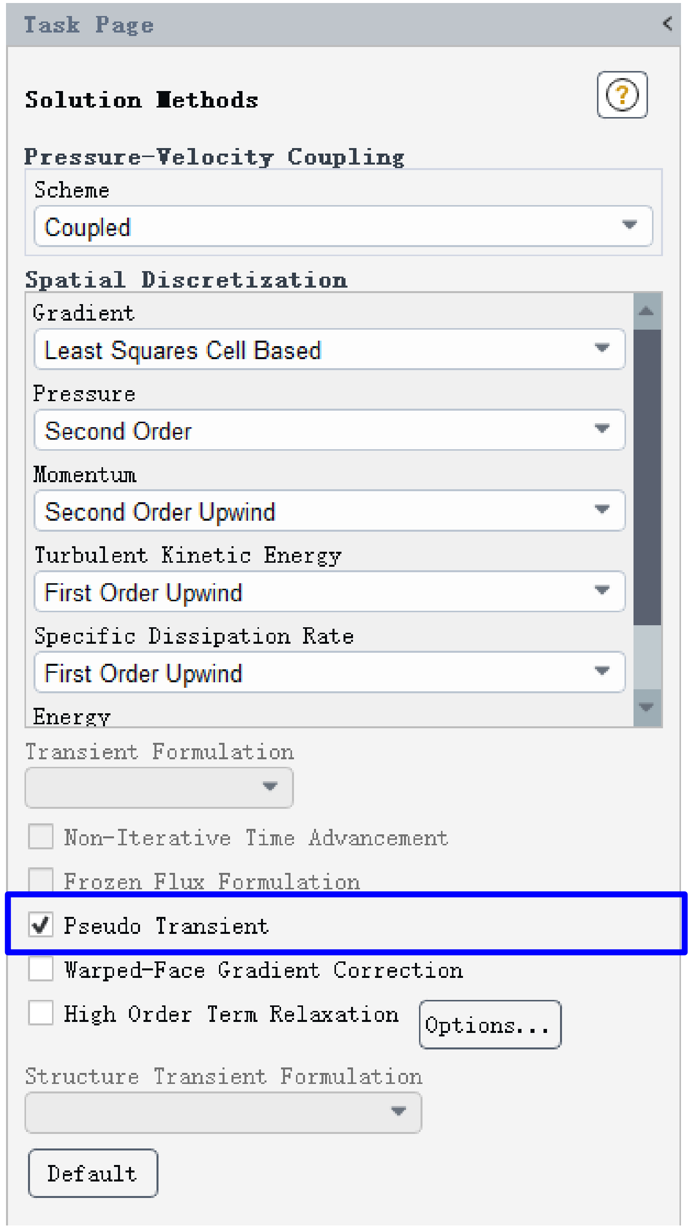
Fluent-default under-relaxation factors for the segregated algorithms

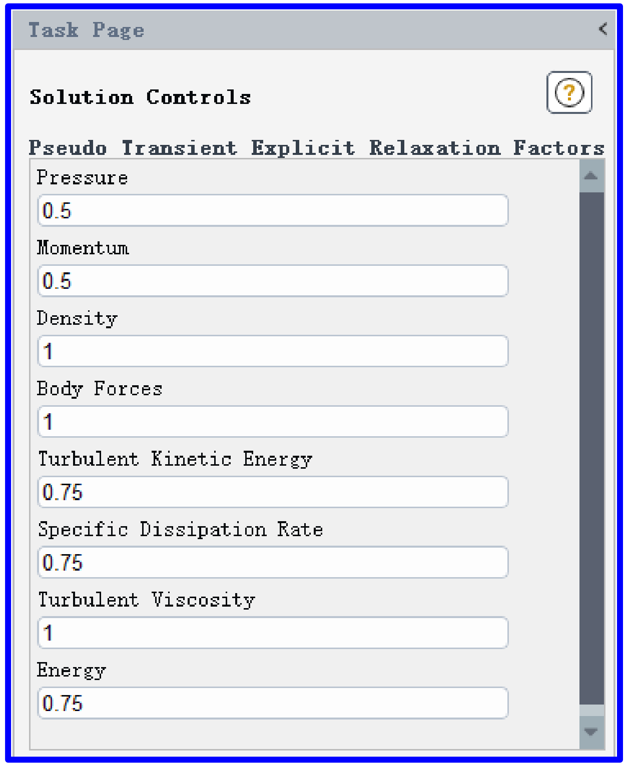
- Underrelaxation factors are to supress oscillations
- Small underrelaxation significantly slows down convergence and may cause “false convergence”.
- Always use large possible underrelaxations
- Try to stay with default underrelaxations
- Underrelax the pressure and momentum when necessary

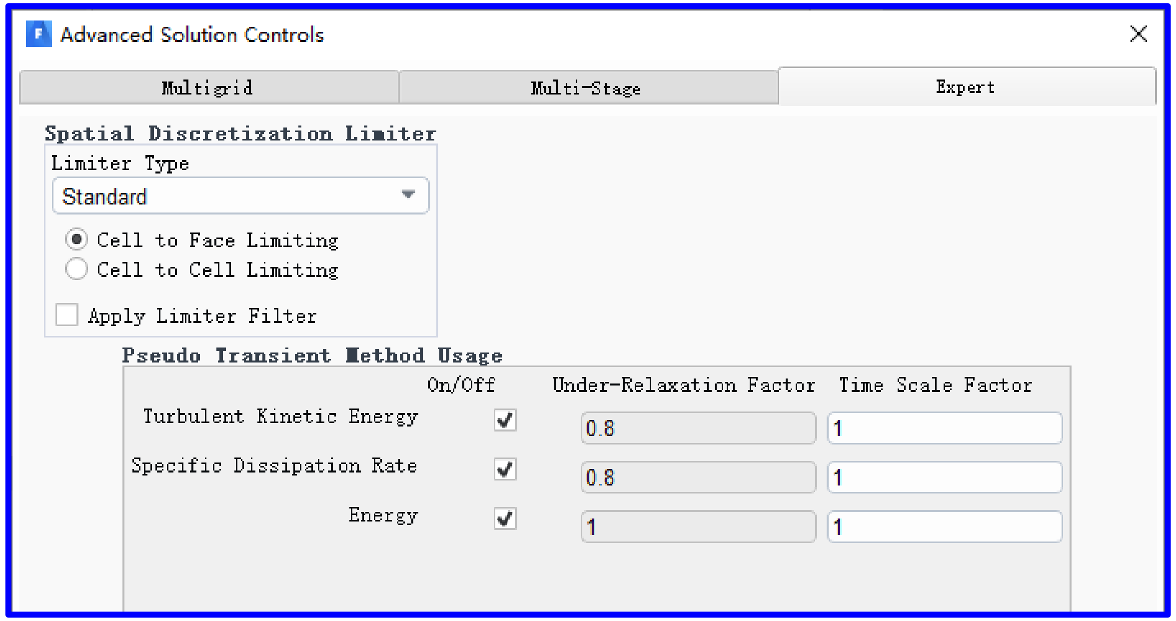
Pseudo-transient under-relaxation

- The under-relaxation is controlled through the pseudo time step size (Δt)
- Auto pseudo time step is used in Fluent

$$\rho_p \Delta V \frac{\phi_p - \phi_p^{old}}{\Delta t} + a_p \phi_p = \sum_{nb} a_{nb} \phi_{nb} + b$$







CONVERGENCE CONTROL

Residuals

Different package use different residuals

- Absolute residual $$R_P = \left| a_P\phi_P - \sum a_{nb}\phi_{nb} - b \right|$$
- Scaled residual $$R_{P,scaled} = \frac{\left| a_P\phi_P - \sum a_{nb}\phi_{nb} - b \right|}{\left| a_P\phi_P \right|}$$
- Normalised residual $$R_{P,scaled} = \frac{\left| a_P\phi_P - \sum a_{nb}\phi_{nb} - b \right|}{\left| a_{\max}\phi_{\max} \right|}$$
- Overall residual of ϕ $$R^{\phi} = \frac{\sum \left| a_P\phi_P - \sum a_{nb}\phi_{nb} - b \right|}{\sum \left| a_P\phi_P \right|}$$

Convergence

- Solution does not change with additional iterations
- Targeted scaled residual 1.0E-3 ~ 1.0E-4
- Monitor residuals

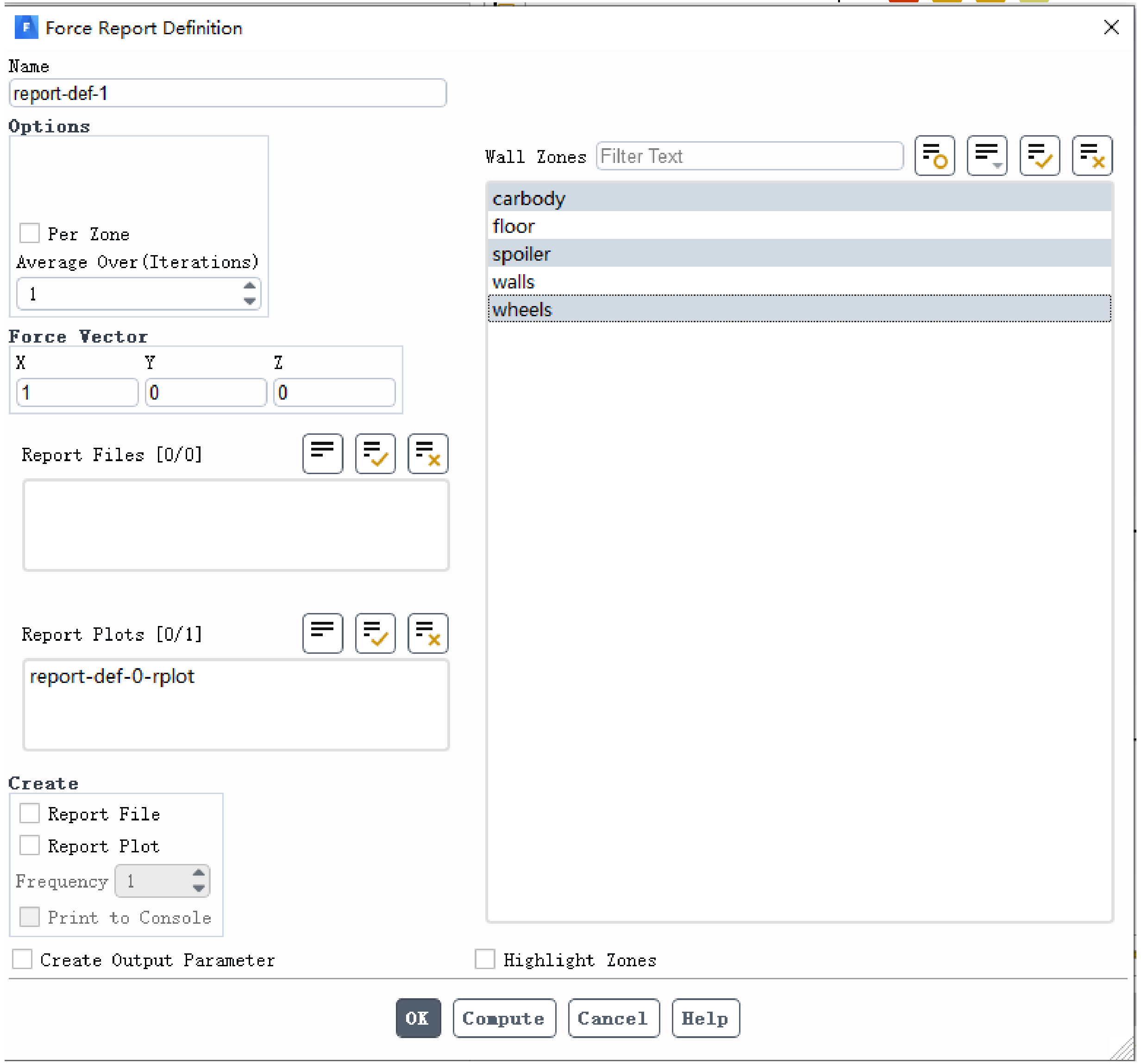
Monitor the residuals

Residuals

- More equations, higher residuals
- If the residuals have met the specified criteria but are still decreasing, the solution may not yet be converged.
- If all the residuals never meet the criteria but no longer decreasing, the solution is converged.
- Low residuals do not always mean a correct solution and high residuals do not automatically mean a wrong solution.
- High residuals are generally with higher-order discretisation – this does not mean 1st-order interpolation is better.

Monitor the fields and variables

- ❖ Monitor the flow/temperature fields during the run
- ❖ Monitor physically meaningful variables: force, etc
- ❖ **Flat and stable**



KEY TAKEWAYS

- ❖ Integral conservation equation applied to control volumes
- ❖ Variables at cell faces are found through interpolation
- ❖ Mesh must be refined to reduce “smearing” and false diffusion
- ❖ In incompressible flows, the density can be calculated using the equation of state or Boussinesq approximation
- ❖ SIMPLE is an iterative algorithm to couple the pressure and velocity
- ❖ Underrelaxation is good for convergence, but can cause “false convergence”
- ❖ Different ways to monitor solution: residual, variable plot, flat and stable

Q&A TIME