Convection

Numerical Flow Simulation

Edouard Boujo

Fall 2024

École
polytechnique
fédérale
de Lausanne

Simple "model" equations

- General conservation equation:

$$\boxed{\frac{\partial(\rho\phi)}{\partial t}} + \boxed{div(\rho\phi\mathbf{u})} = \boxed{div(\Gamma\, grad(\phi))} + \boxed{S}$$

unsteadiness convection diffusion source

- Steady/unsteady diffusion (e.g. heat conduction):

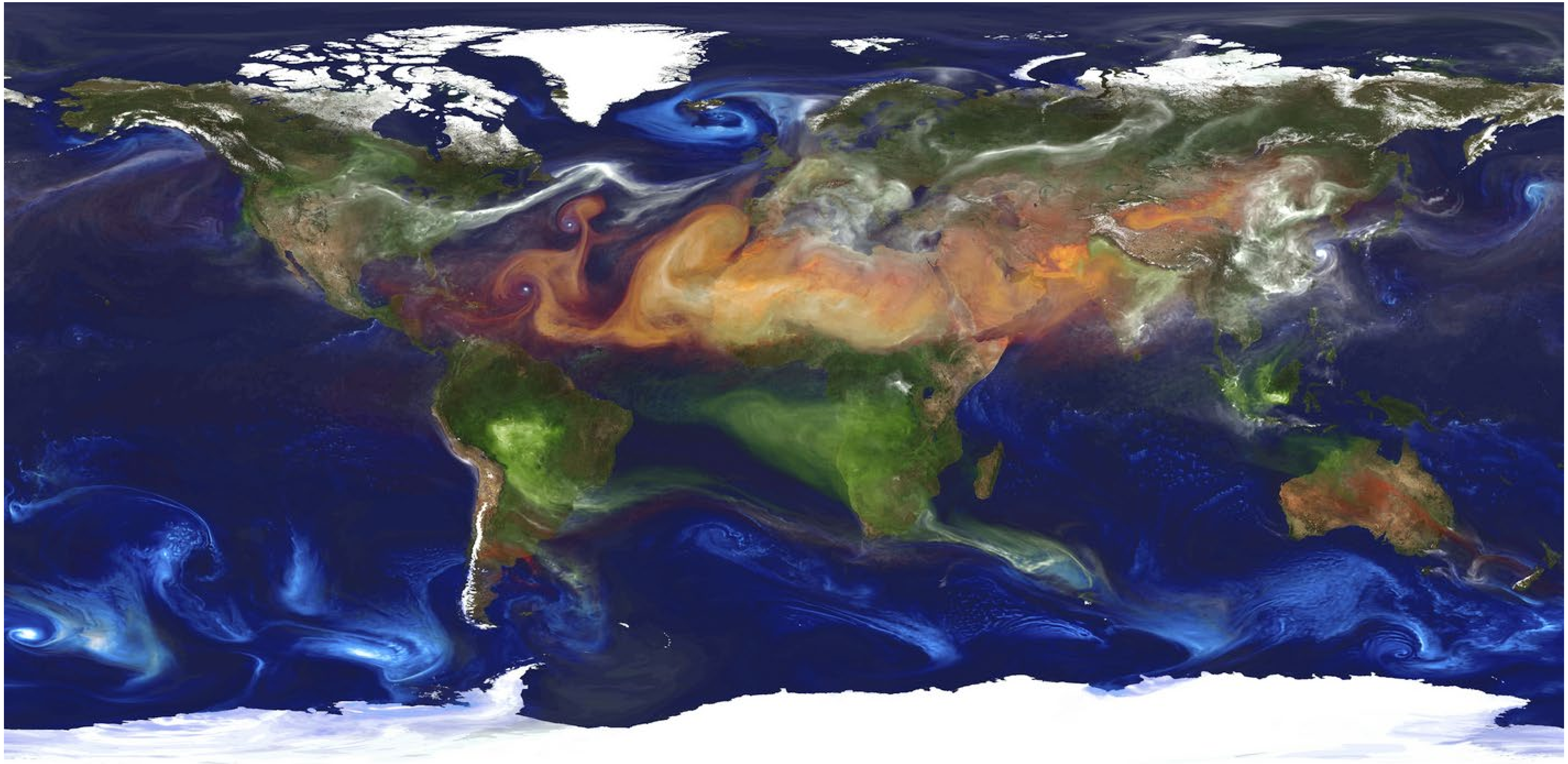
$$\boxed{div(\Gamma\, grad(\phi))} + S = 0$$

$$\boxed{\frac{\partial(\rho\phi)}{\partial t}} = div(\Gamma\, grad(\phi))$$

- Steady convection-diffusion (transport of a scalar, e.g. dye, salt, chemical species):

$$\boxed{div(\rho\phi\mathbf{u})} = div(\Gamma\, grad(\phi))$$

Convection-diffusion



Atmospheric simulation (GEOS-5 simulation, 10-km grid size), NASA Center for Climate Simulation at Goddard Space Flight Center. Dust (red) is lifted from the surface, sea salt (blue) swirls inside cyclones, smoke (green) rises from fires, and sulfate particles (white) stream from volcanoes and fossil fuel emissions.

1D steady convection-diffusion

$$\frac{d(\rho\phi u)}{dx} = \frac{d}{dx}\left(\Gamma \frac{d\phi}{dx}\right) \qquad \oint_A \rho\phi \mathbf{u}\cdot\mathbf{n}\, dA = \oint_A \Gamma\, grad(\phi)\cdot\mathbf{n}\, dA$$

- Assume the density and velocity are known, and satisfy continuity:

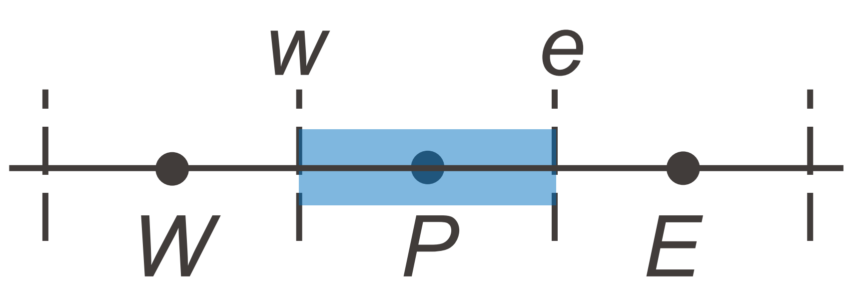
$$\frac{d(\rho u)}{dx} = 0 \quad \rightarrow \quad (\rho u)_e - (\rho u)_w = 0$$

$$F_e - F_w = 0$$

denoting $F = \rho u$ the convective mass flux

- Integration over CV:

$$F_e\phi_e - F_w\phi_w = \Gamma_e \left.\frac{d\phi}{dx}\right|_e - \Gamma_w \left.\frac{d\phi}{dx}\right|_w$$



- Use CD to discretize the diffusion term, as usual:

$$F_e\phi_e - F_w\phi_w = \Gamma_e \frac{\phi_E - \phi_P}{\delta x_{PE}} - \Gamma_w \frac{\phi_P - \phi_W}{\delta x_{WP}}$$

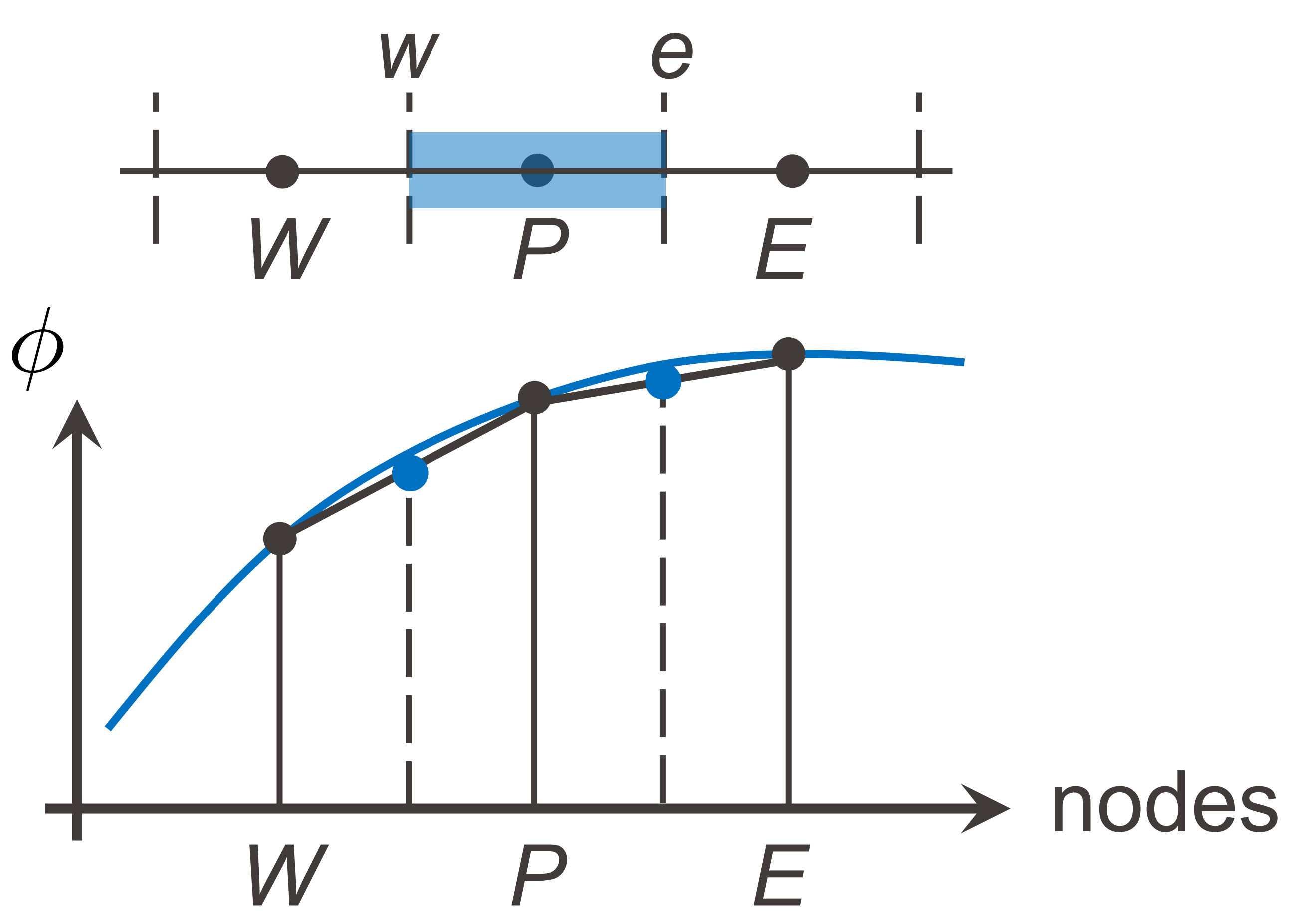
$$\boxed{F_e\phi_e - F_w\phi_w = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)}$$

denoting $D_e = \dfrac{\Gamma_e}{\delta x_{PE}}, \; D_w = \dfrac{\Gamma_w}{\delta x_{WP}}$

- What about the convection term? How to discretize the face values?

1D steady convection-diffusion: central differencing

$$F_e\phi_e - F_w\phi_w = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)$$



- Linear interpolation

On a uniform grid: $\phi_w \approx \dfrac{\phi_W + \phi_P}{2}$ $\phi_e \approx \dfrac{\phi_P + \phi_E}{2}$

- Governing equation becomes:

$$\frac{F_e}{2}(\phi_P + \phi_E) - \frac{F_w}{2}(\phi_W + \phi_P) = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)$$

- Algebraic equation: $a_P\phi_P = a_W\phi_W + a_E\phi_E$

$$a_W = D_w + \frac{F_w}{2}, \quad a_E = D_e - \frac{F_e}{2}, \quad a_P = \left(D_w + D_e - \frac{F_w}{2} + \frac{F_e}{2}\right) = a_W + a_E + (F_e - F_w)$$

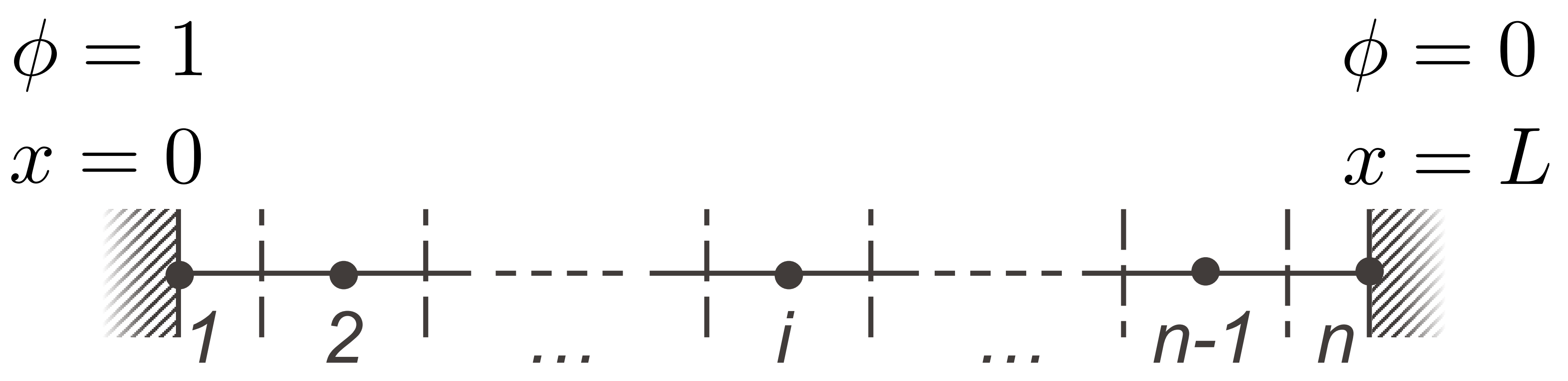
- Same form as steady diffusion eq.
(but additional terms for convection).

Note: by continuity, F_e=F_w

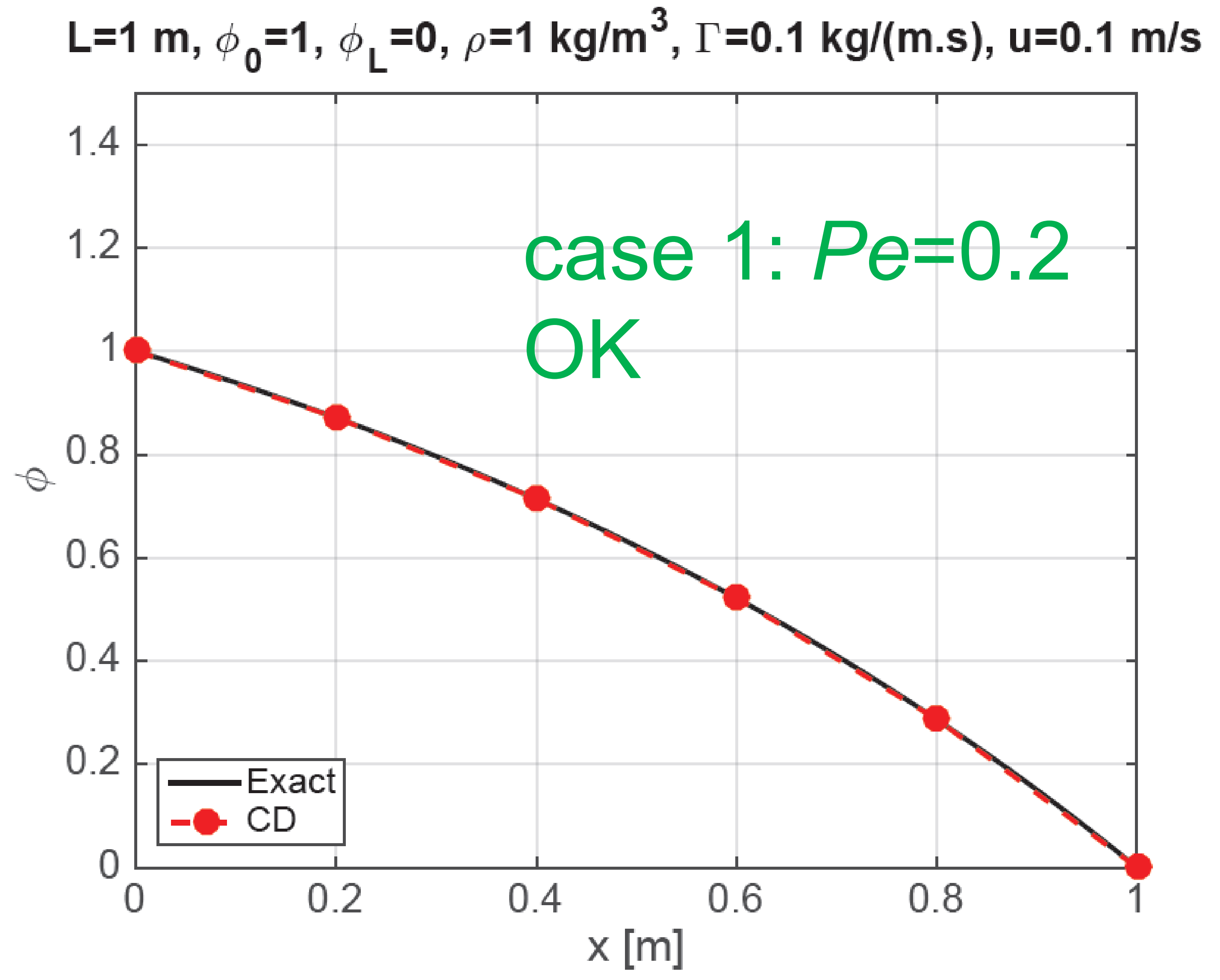
1D steady convection-diffusion: central differencing

- Example: domain $[0,1]$ m

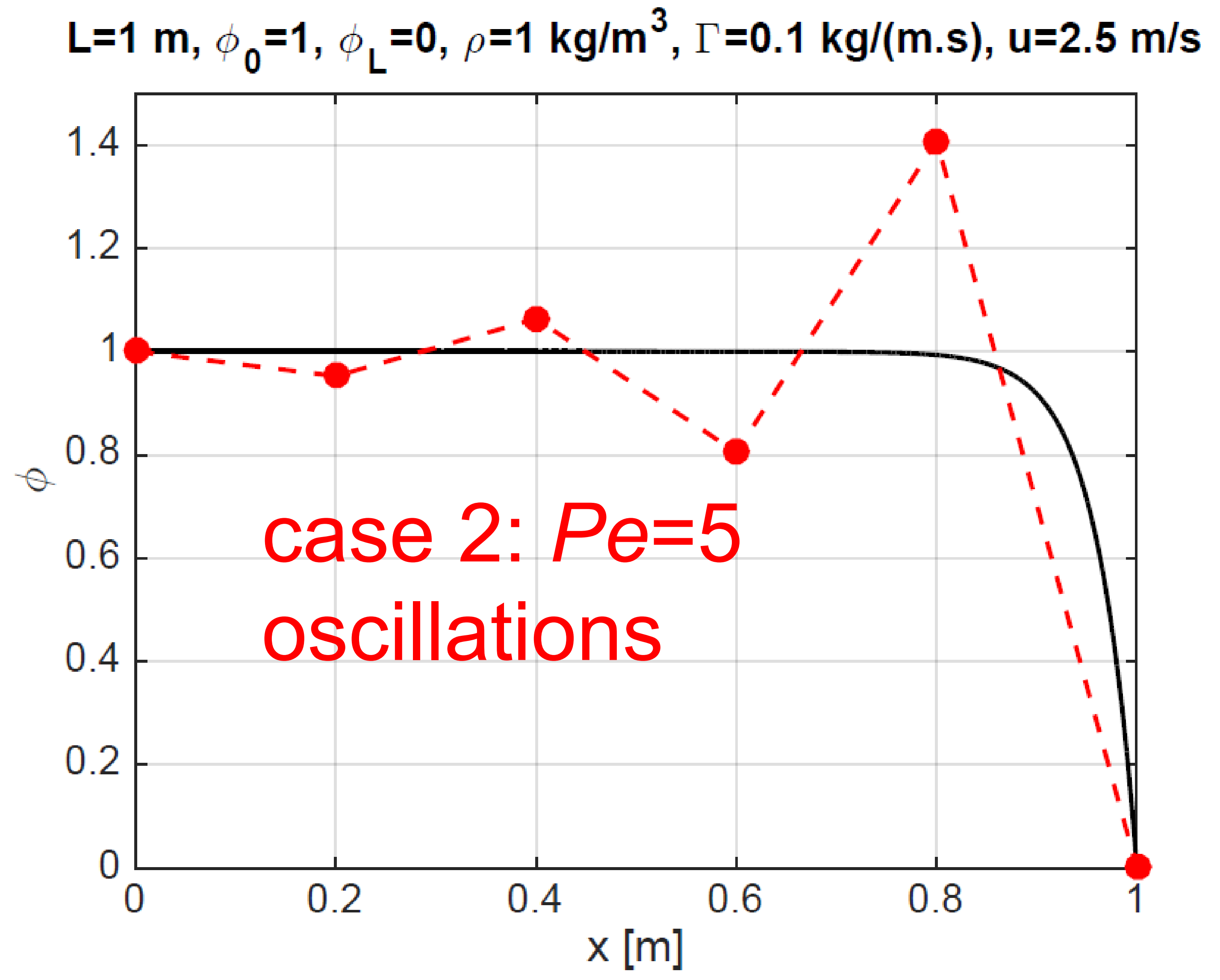
  $\rho = 1\ \mathrm{kg/m^3},\ \Gamma = 0.1\ \mathrm{kg/(m.s)}$



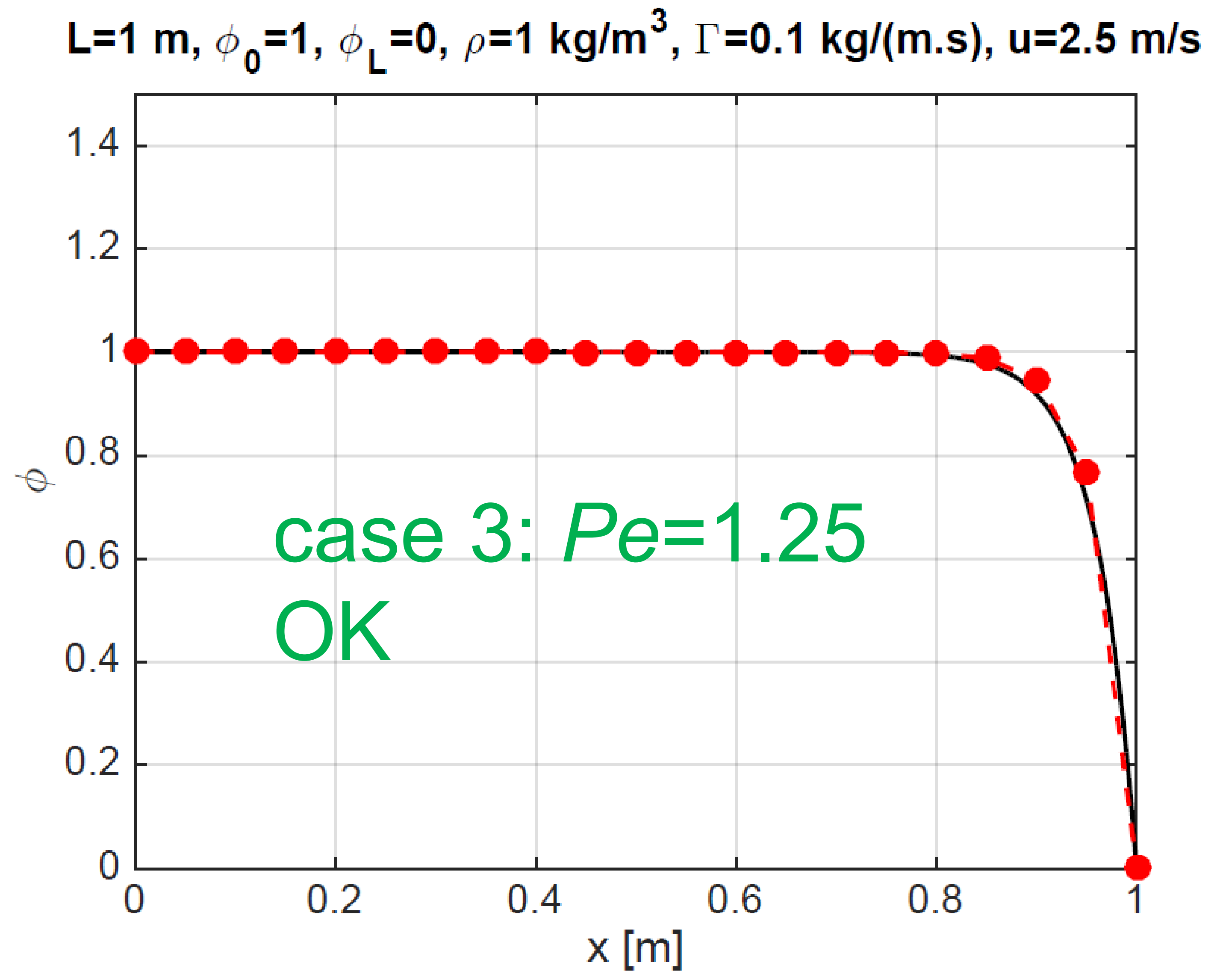
$u = 0.1\ \mathrm{m/s},\ n = 6$



$u = 2.5\ \mathrm{m/s},\ n = 6$



$u = 2.5\ \mathrm{m/s},\ n = 21$



Influence of the (numerical) Péclet number: $Pe = \dfrac{\rho u}{\Gamma/\delta x} = \dfrac{F}{D}$

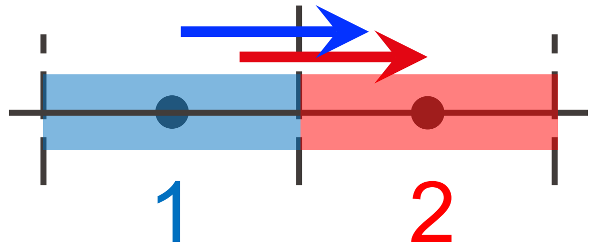
Properties of discretization schemes

- Why do we need to discuss that?
  - In theory, better **accuracy** when refining the mesh, for all discretization schemes.
  - In practice, can only use a finite number of CVs.
  - On a finite-size mesh, numerical results are physically **realistic** only when the scheme has some fundamental properties.

- In particular, a discretization scheme must be:
  1. Conservative
  2. Bounded
  3. Transportive

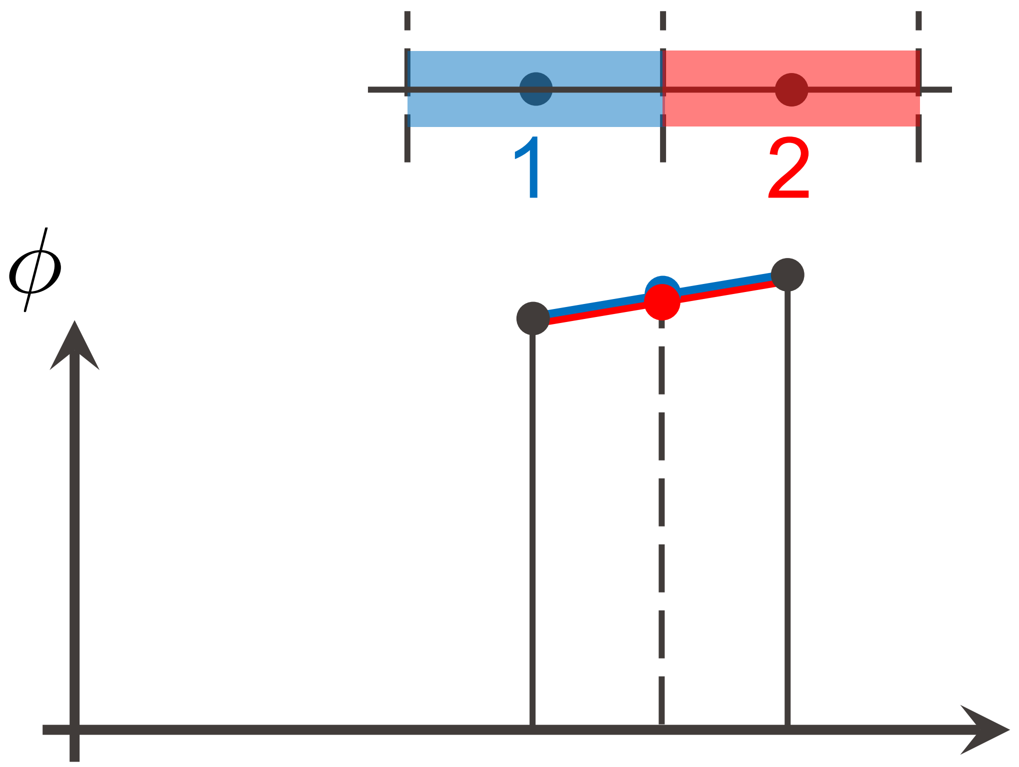
Properties of discretization schemes

1. Conservativeness
   - Local CV conservation ensures global conservation only if the flux leaving a CV across a face is equal to the flux entering the neighboring CV across the same face. (Relevant to both convective and diffusive terms.)

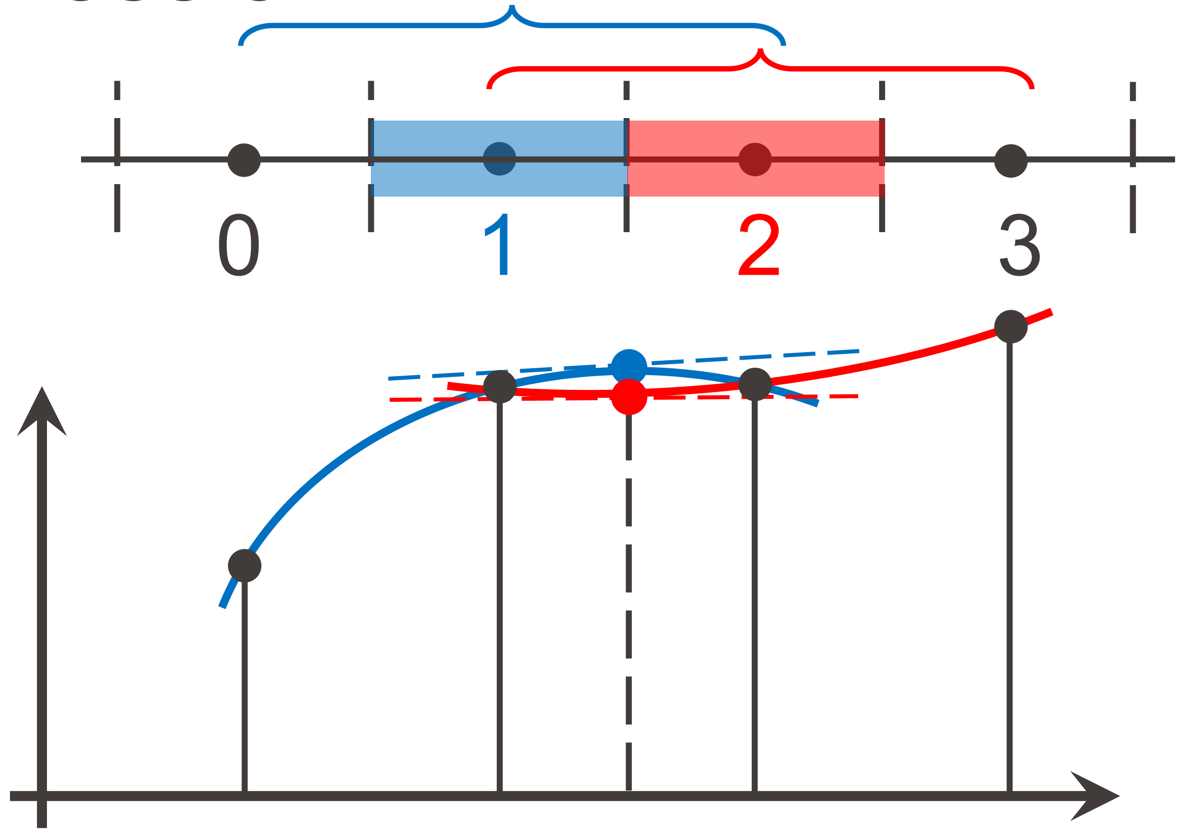


$$(\rho u \phi)_{e1} = (\rho u \phi)_{w2} \qquad \left(\Gamma \frac{\partial \phi}{\partial x}\right)_{e1} = \left(\Gamma \frac{\partial \phi}{\partial x}\right)_{w2}$$

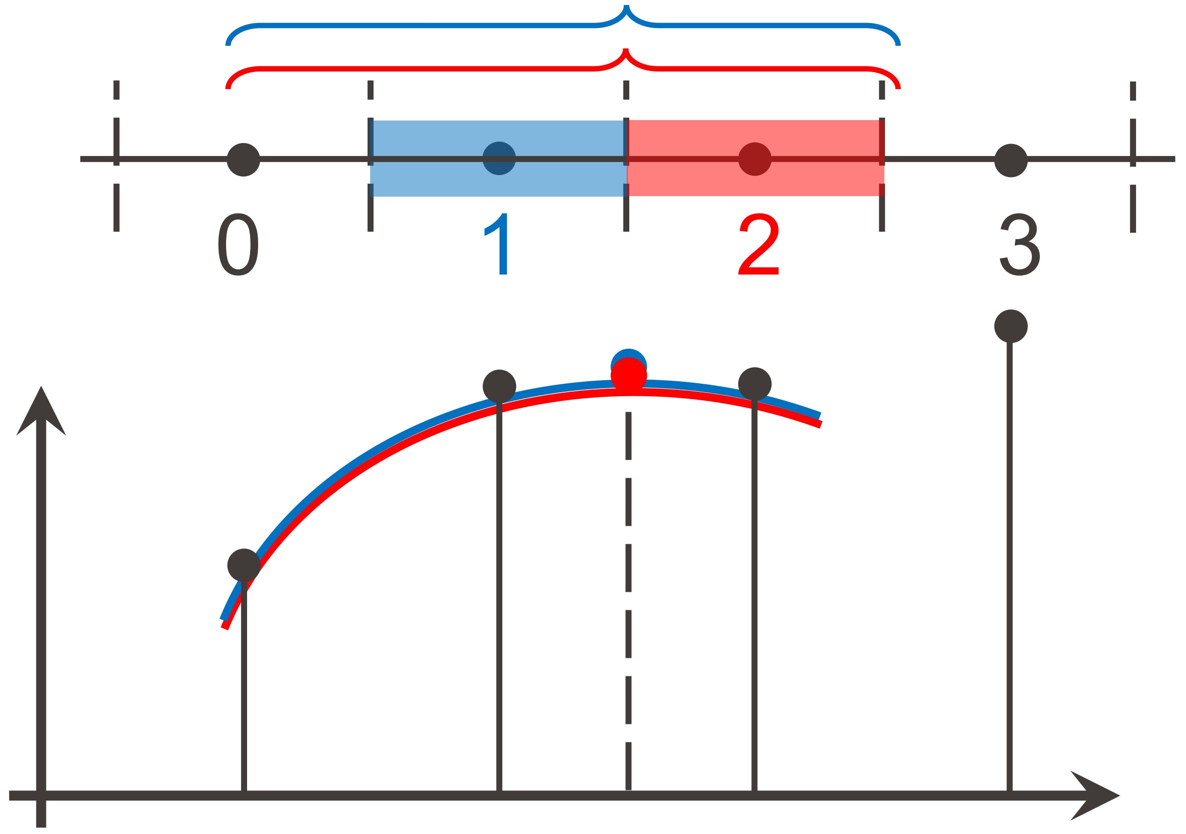
   - To achieve this, the flux must be represented in a **consistent** way in both CVs, i.e. with the same expression.



Linear across the face: conservative

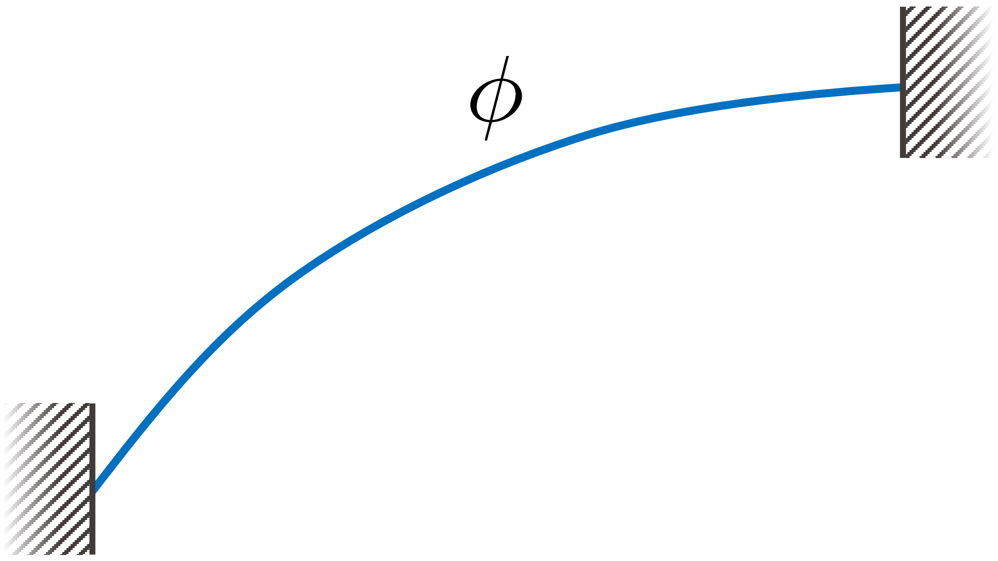


Quadratic defined with respect to CV: NOT conservative



Quadratic defined with respect to face: conservative

Properties of discretization schemes



2. Boundedness
   - In the absence of sources, internal values of ϕ must be **bounded** by the boundary values (no local min. or max.).
   - One way of achieving boundedness: if all coefficients of the algebraic equation
   $$a_P \phi_P = a_W \phi_W + a_E \phi_E + \ldots$$
   have the **same sign**. Physical interpretation: all else being equal, an increase in ϕ at one node should yield an increase of ϕ at neighboring nodes. With different signs, the solution may not converge or may contain unphysical oscillations.
   - One desirable feature for satisfying boundedness: if the matrix $\mathbf{A}$ of the linear system is strictly **diagonal dominant:**

   $$\begin{cases} |a_{i,i}| \geq \sum_{i \neq j} |a_{i,j}| & \text{for all rows } i, \\ |a_{i,i}| > \sum_{i \neq j} |a_{i,j}| & \text{for at least one row } i. \end{cases}$$

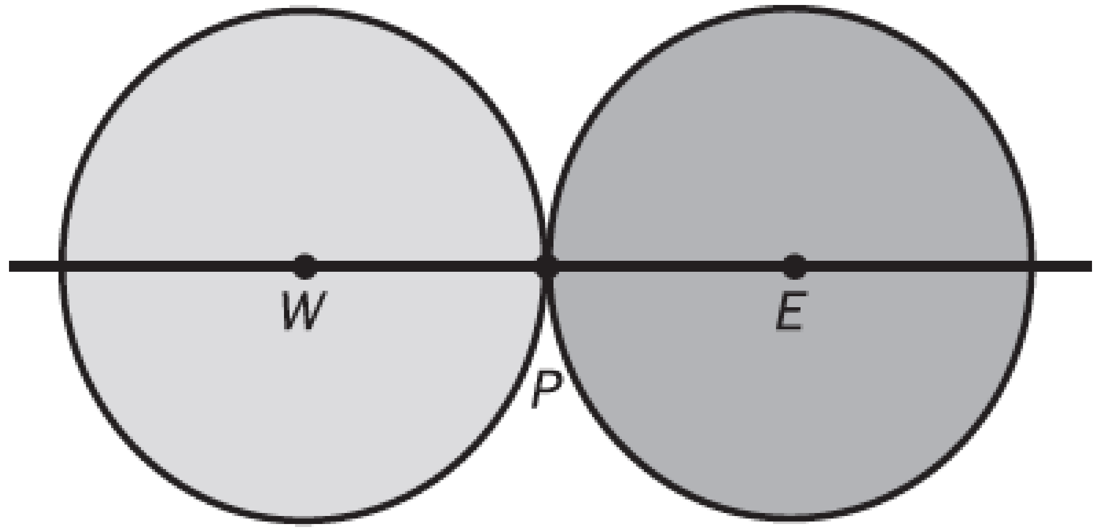
   Strict diagonal dominance is a sufficient condition for the **convergence** of iterative methods (Scarborough criterion).

Properties of discretization schemes

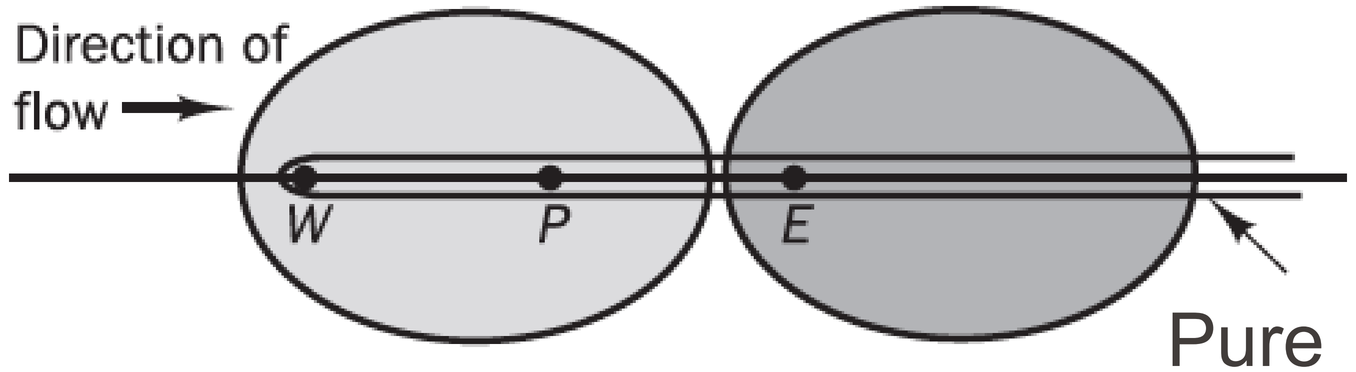
3. Transportiveness
   - With convection, the influence of neighboring nodes is **not symmetric**.
   - Characterized by the Péclet number:

$$Pe = \frac{\text{convective flux}}{\text{diffusive flux}} = \frac{\rho u \phi}{\Gamma\, grad(\phi)} = \frac{\rho u}{\Gamma / L}$$

   - In the limit of pure convection: $\phi_P = \phi_W$.
     Upstream influence (*W*) only, no downstream influence (*E*).
   - Numerical schemes should account for the asymmetry, the flow direction, and the relative strengths of convection/diffusion (*Pe*).



(a) Pure diffusion $Pe = 0$



(b) Convection + diffusion $Pe > 0$

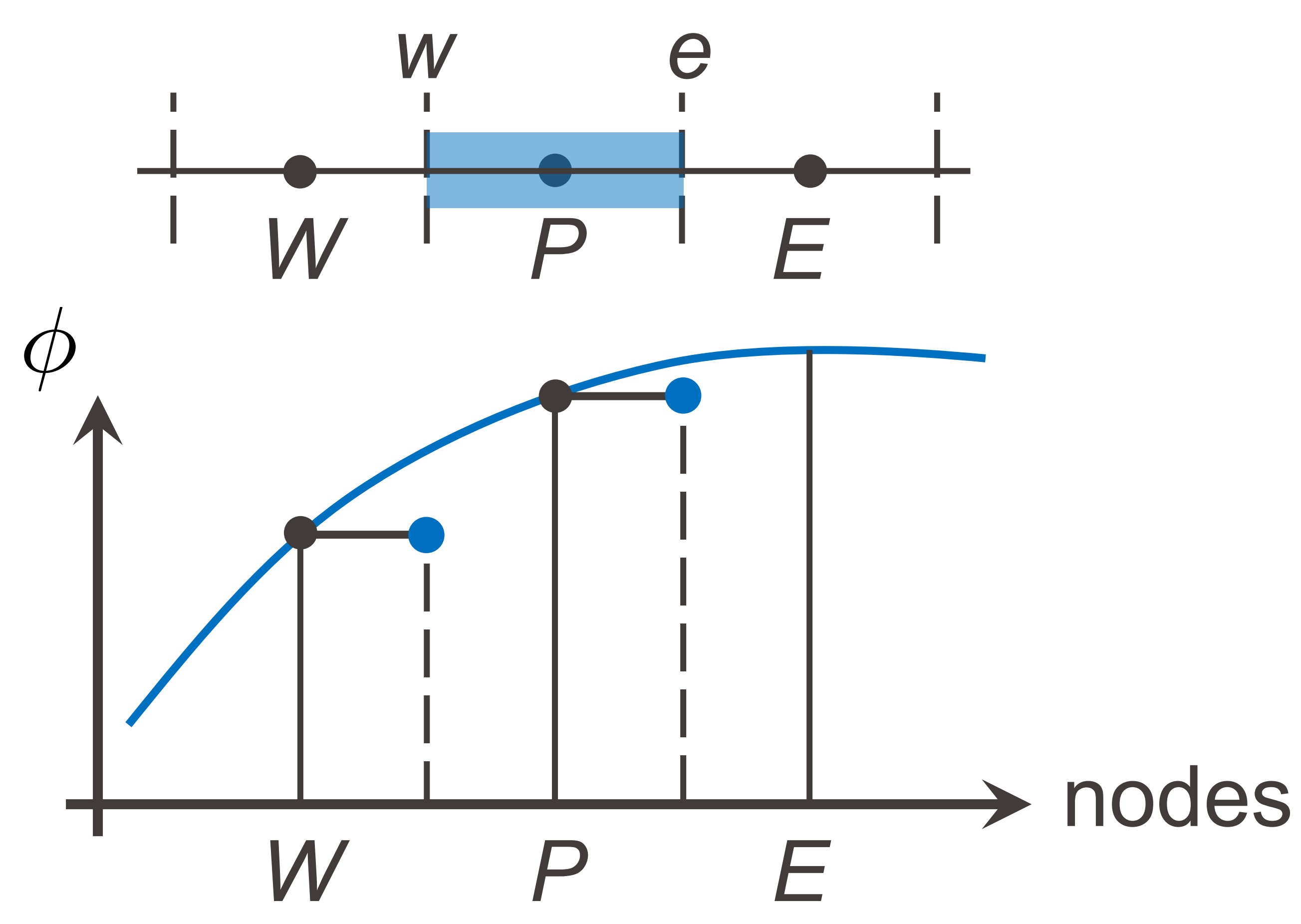
Pure convection $Pe \to \infty$

Steady convection-diffusion: central differencing CD

- Bounded?
  - Satisfies the Scarborough criterion…
  - … but coefficients can become negative: $a_E = D_e - \frac{F_e}{2} < 0 \quad \text{if} \quad Pe > 2$
- Transportive?
  - Not transportive: symmetric, too much weight on *E*.
- Conservative?
  - Yes
- Accuracy: 2nd-order

Steady convection-diffusion: upwind differencing UD

$$F_e\phi_e - F_w\phi_w = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)$$



- Constant interpolation, taking into account the flow direction (use upstream node):

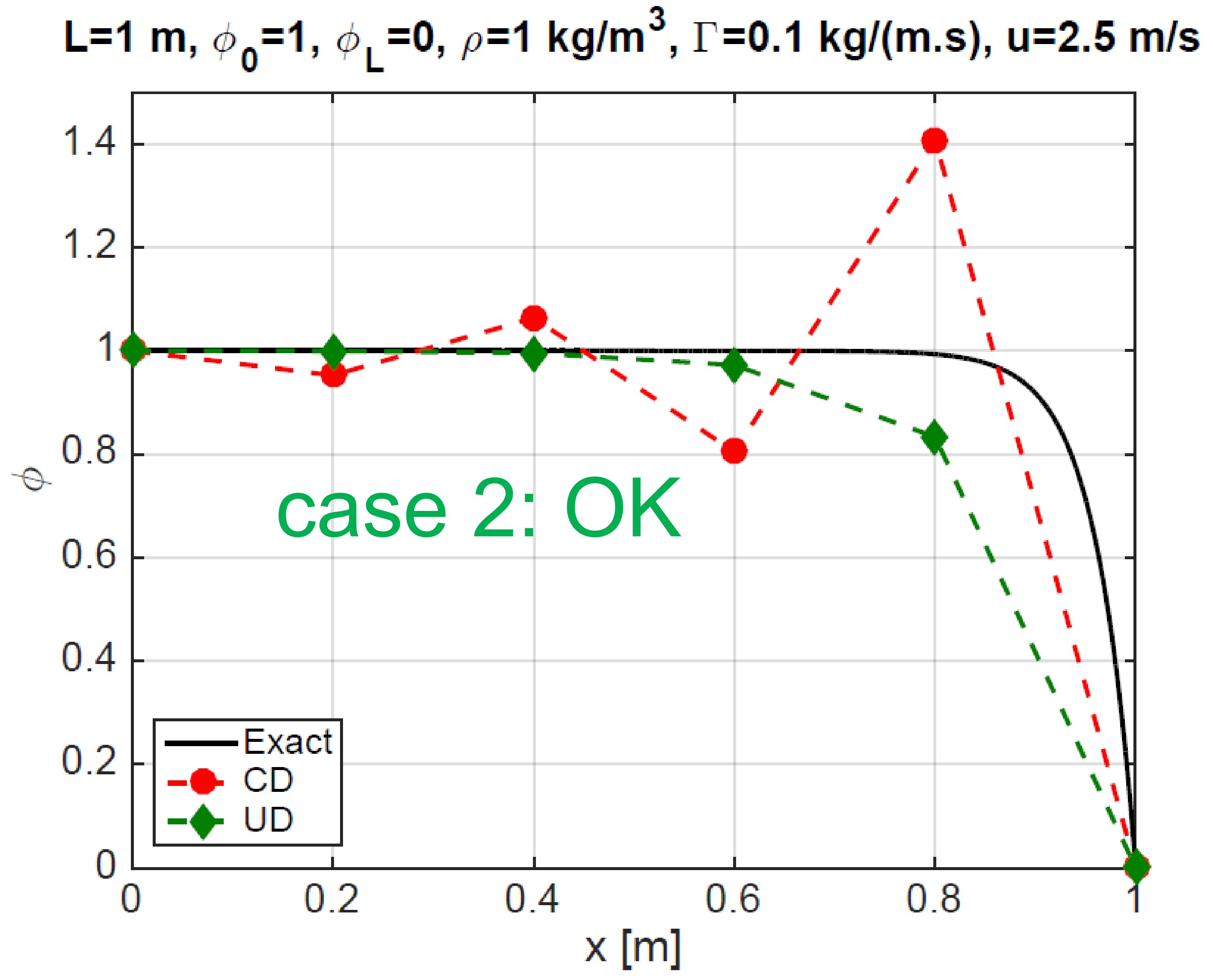
$$\text{if } F_w, F_e > 0: \ \phi_w \approx \phi_W, \ \phi_e \approx \phi_P$$

$$\text{if } F_w, F_e < 0: \ \phi_w \approx \phi_P, \ \phi_e \approx \phi_E$$

- Governing equation becomes:

$$\text{if } F_w, F_e > 0: \quad (D_w + D_e + F_e)\phi_P = (D_w + F_w)\phi_W + D_e\phi_E$$

$$\text{if } F_w, F_e < 0: \quad (D_w + D_e - F_w)\phi_P = D_w\phi_W + (D_e - F_e)\phi_E$$



- General expression: $a_P\phi_P = a_W\phi_W + a_E\phi_E$

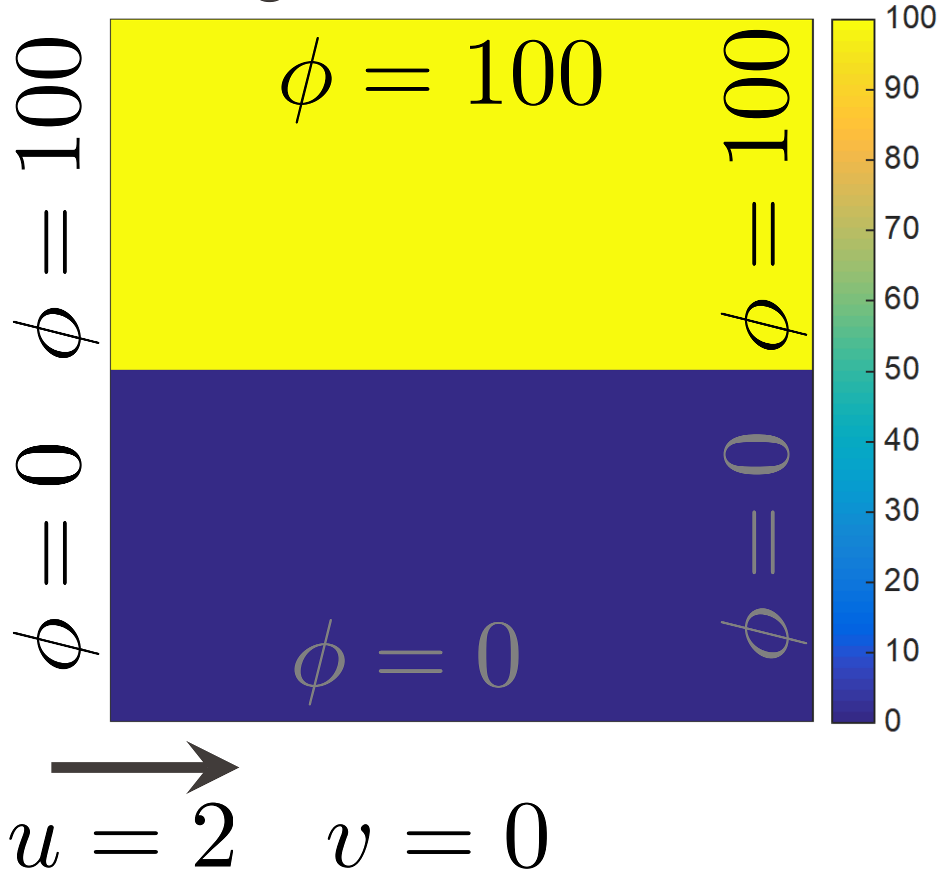
$$a_W = D_w + \max(0, F_w)$$
$$a_E = D_e + \max(0, -F_e)$$
$$a_P = a_W + a_E + (F_e - F_w)$$

Coefficients always positive.

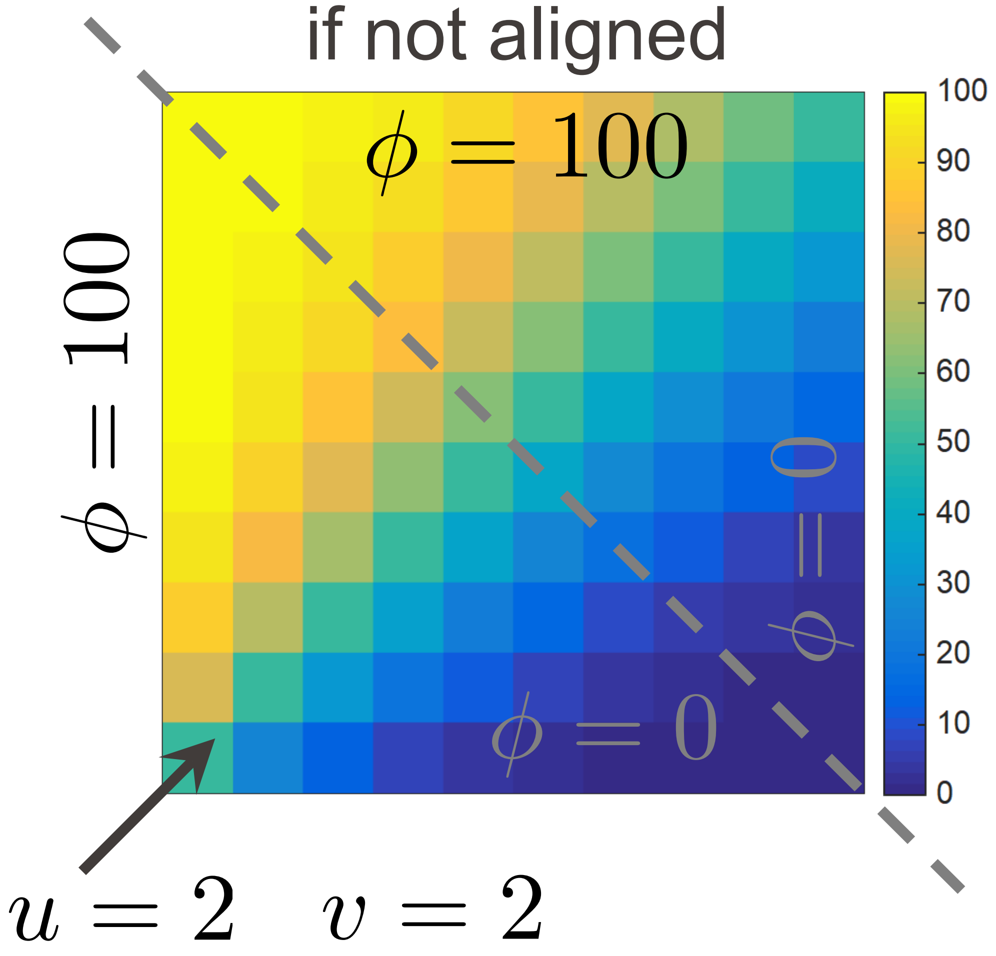
Steady convection-diffusion: upwind differencing UD

- Drawback: numerical diffusion ("false diffusion") when flow not aligned
- Example: 2D pure convection ($\Gamma = 0$), prescribed velocity field

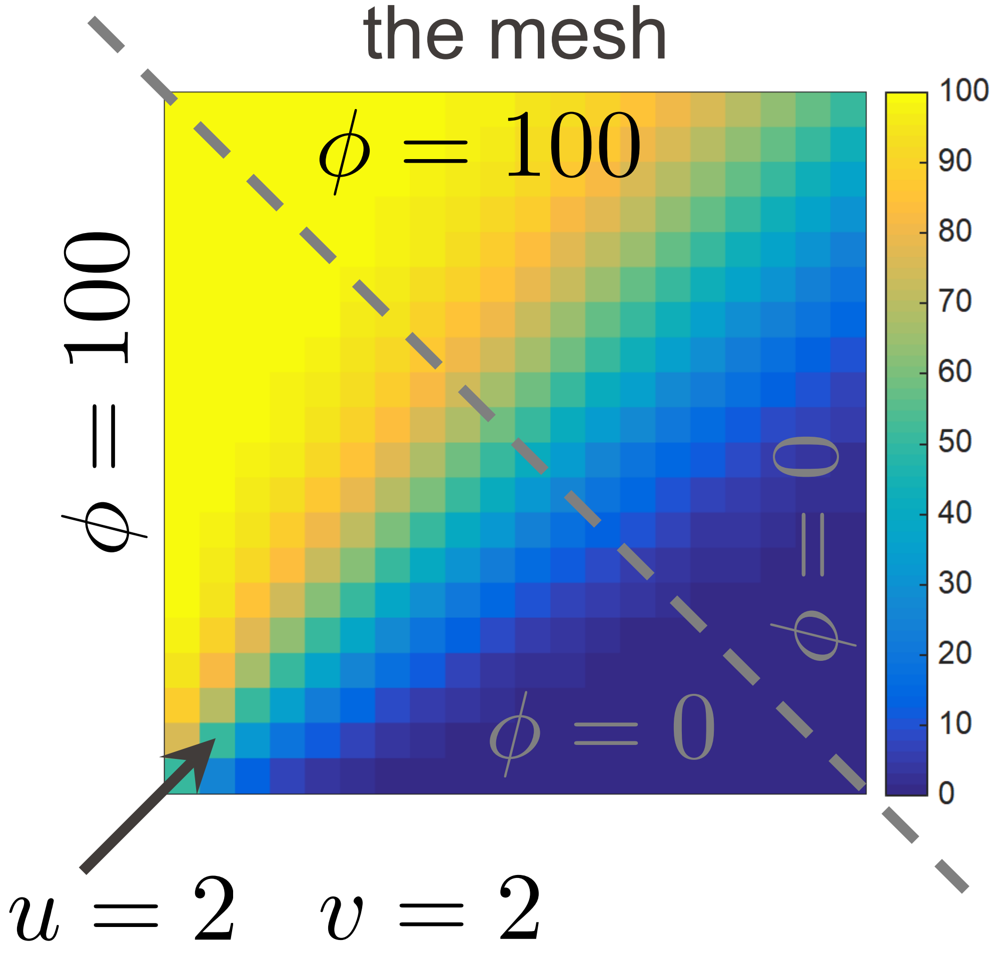
OK when flow aligned with mesh

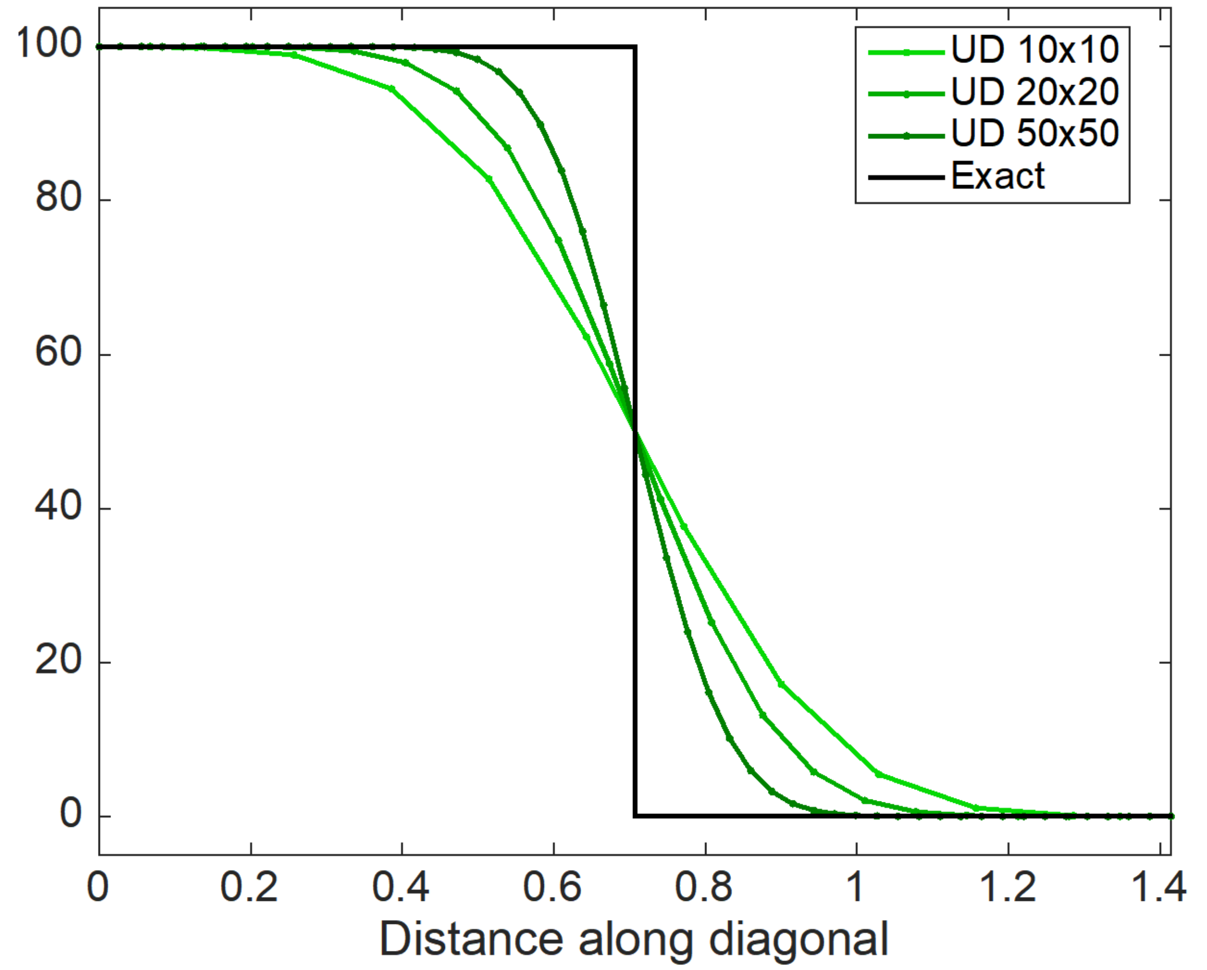


Numerical diffusion if not aligned



Better when refining the mesh





Can be understood: $0 = F_e\phi_e - F_w\phi_w + F_n\phi_n - F_s\phi_s \approx F_e\phi_P - F_w\phi_W + F_n\phi_P - F_s\phi_S$

$$F_e = F_w > 0,\ F_n = F_s = 0$$
$$\rightarrow \quad \phi_P = \phi_W$$

$$F_e = F_w = F_n = F_s$$
$$\rightarrow \quad \phi_P = (\phi_W + \phi_S)/2$$

- Need higher-order scheme and/or finer mesh.

Steady convection-diffusion: upwind differencing UD

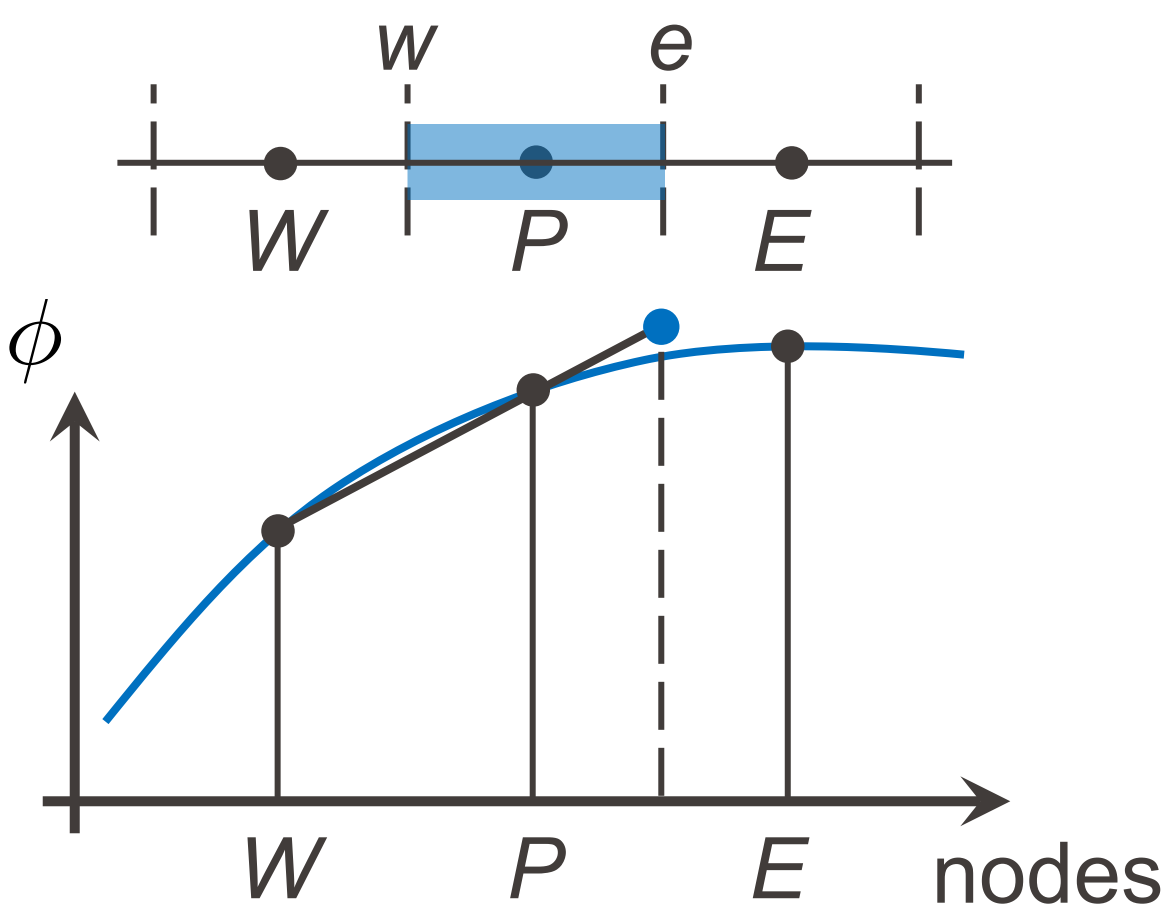
- Bounded?
  - Satisfies the Scarborough criterion
  - Coefficients always positive

- Transportive?
  - Yes by construction

- Conservative?
  - Yes

- Accuracy: 1st-order, numerical diffusion

Steady convection-diffusion: LUD (SOU)



- "Linear upwind differencing" (or "second-order upwind")
- 2nd-order extension of UD: linear interpolation, taking into account the flow direction (use 2 upstream nodes)

$$\text{if } F_w, F_e > 0: \quad \phi_e \approx \phi_P + \left.\frac{\partial \phi}{\partial x}\right|_P \delta x_{Pe} \approx \phi_P + \frac{\phi_P - \phi_W}{\delta x_{WP}} \delta x_{Pe}$$

On a uniform grid: $\phi_w \approx \dfrac{3\phi_W - \phi_{WW}}{2}$ $\qquad \phi_e \approx \dfrac{3\phi_P - \phi_W}{2}$

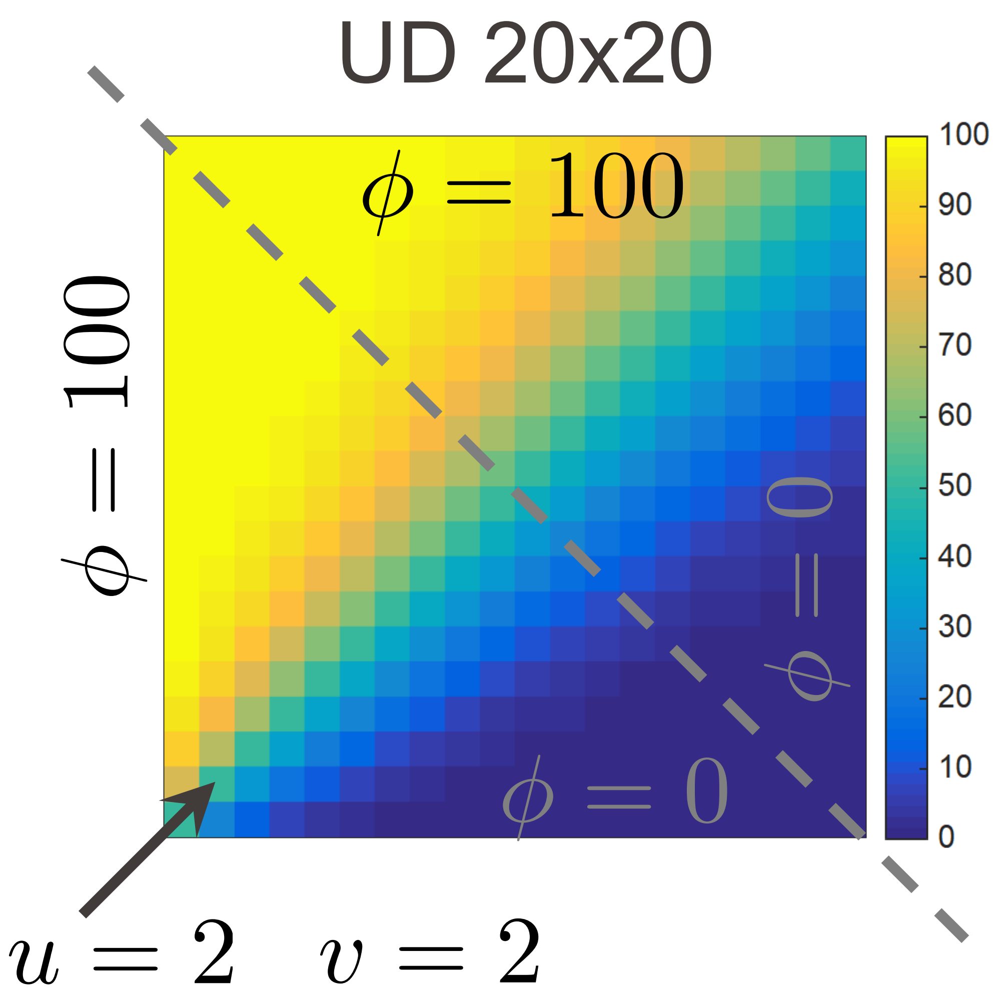
- Governing equation becomes: $a_P \phi_P = a_{WW} \phi_{WW} + a_W \phi_W + a_E \phi_E$

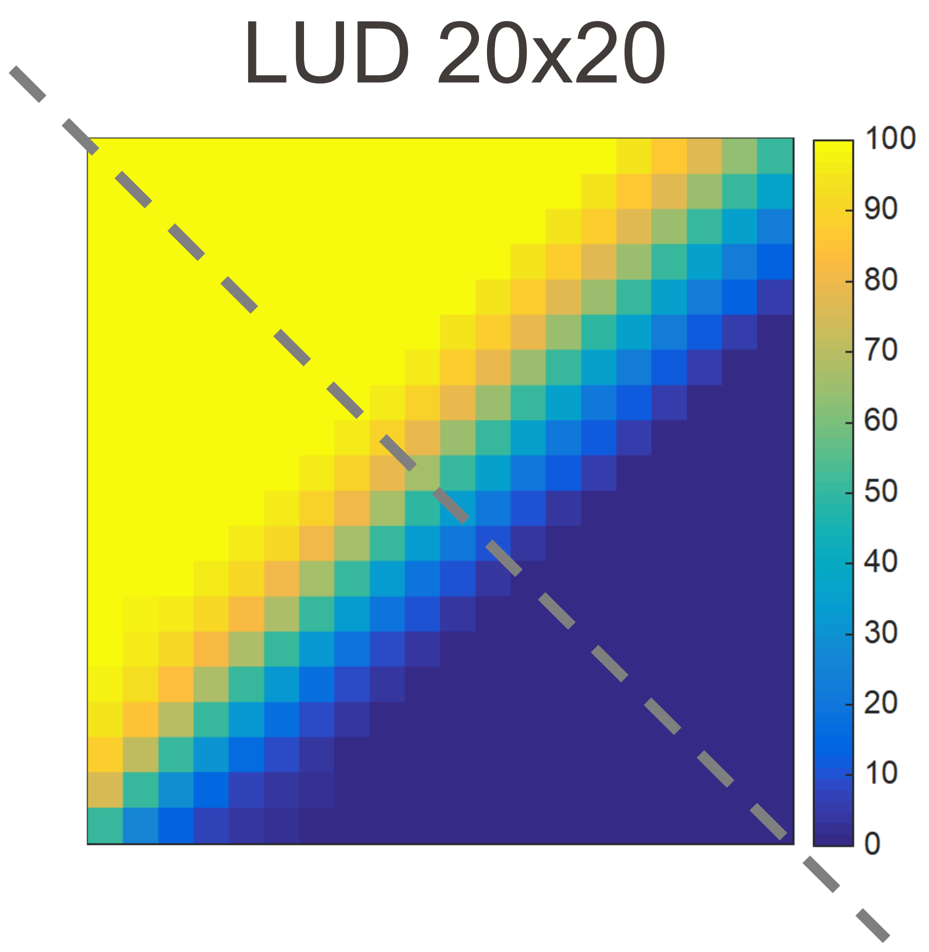
$$a_{WW} = -\frac{F_w}{2}$$
$$a_W = D_w + \frac{F_e}{2} + \frac{3F_w}{2}$$
$$a_E = D_e$$
$$a_P = a_{WW} + a_W + a_E + (F_e - F_w)$$

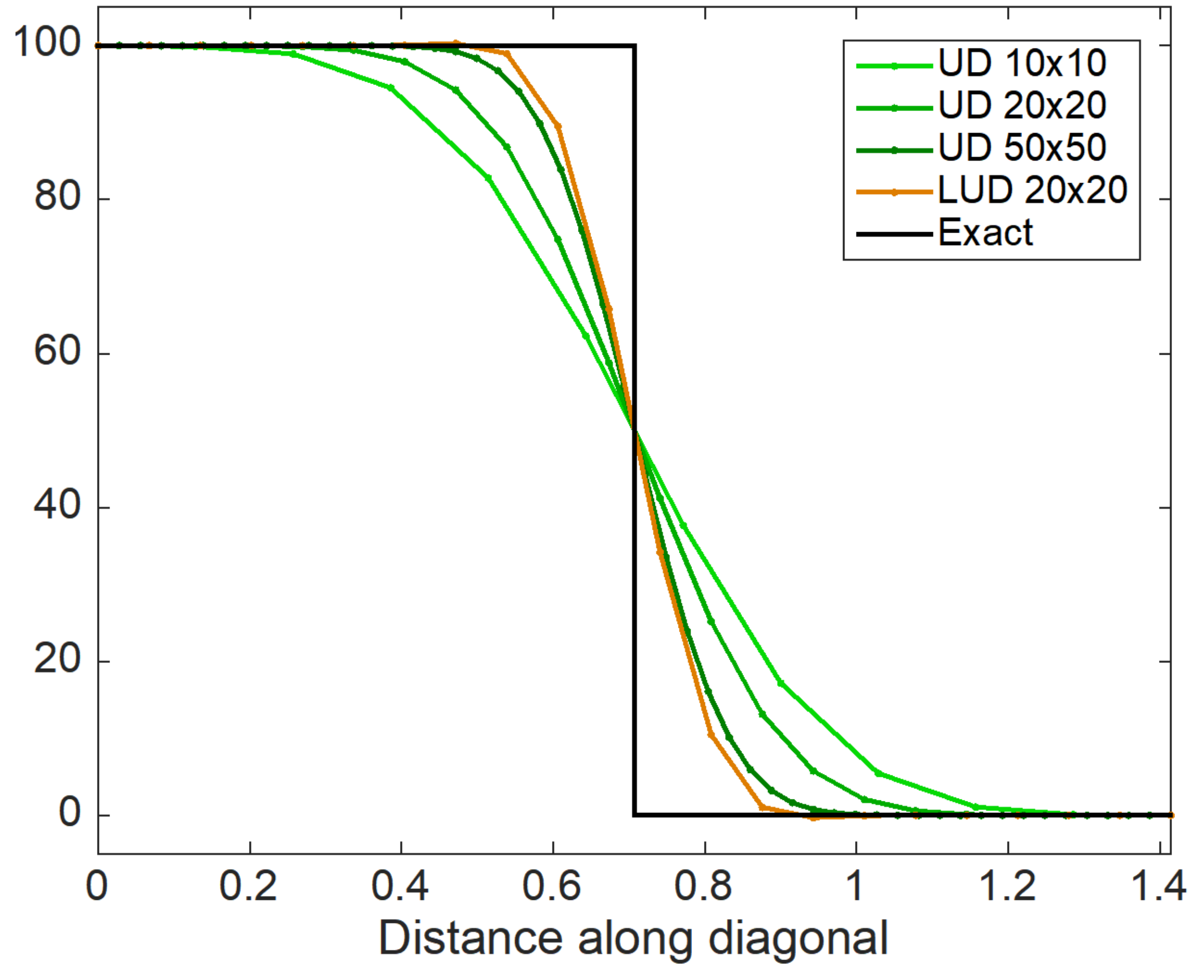
One negative coefficient.

Steady convection-diffusion: LUD (SOU)

- LUD (2nd-order) more accurate but less stable.







In this example, LUD 20x20 better than UD 50x50

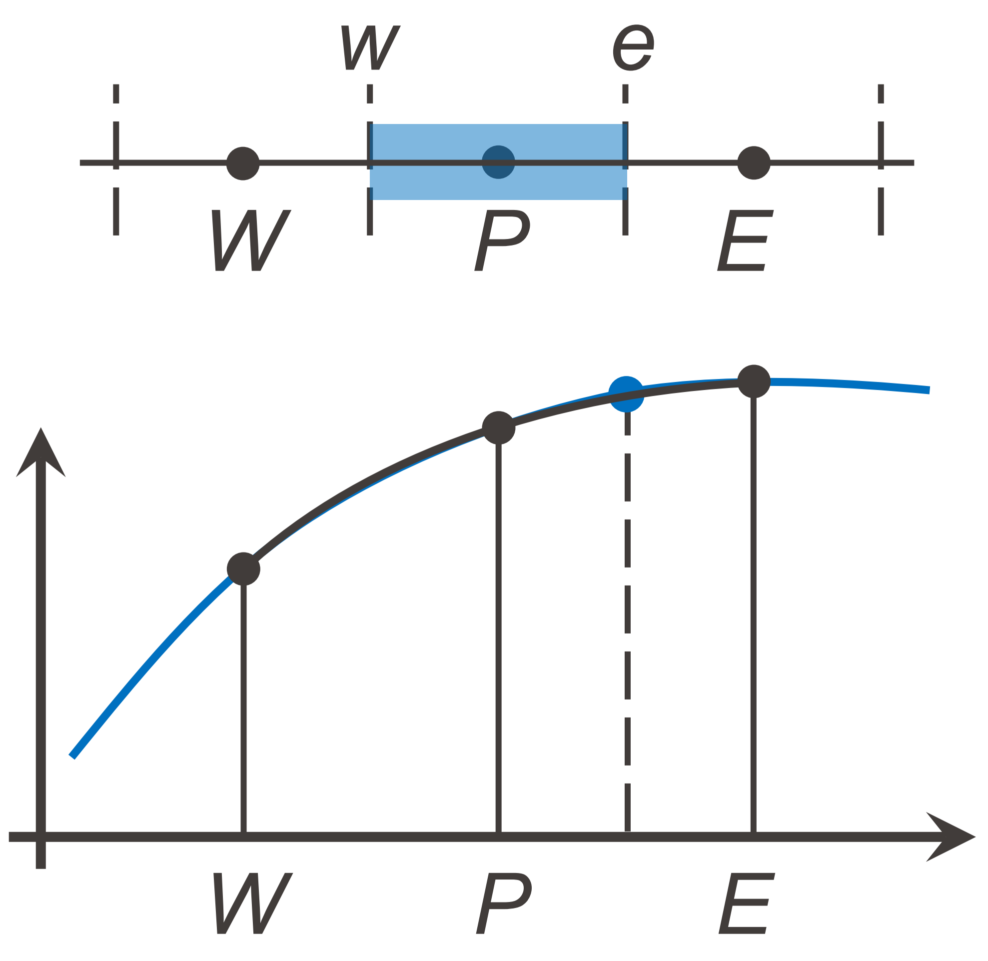
- Note: for nonlinear problems, 1st-order schemes (like UD) are useful to get a first solution, that can then be used as initial guess for a second calculation with a higher-order scheme (like LUD). This two-step procedure is generally faster than starting directly with a higher-order calculation from a poor initial guess.

Steady convection-diffusion: LUD (SOU)

- Bounded?
  - Satisfies the Scarborough criterion
  - One negative coefficient
- Transportive?
  - Yes by construction
- Conservative?
  - Yes
- Accuracy: 2nd-order

Steady convection-diffusion: QUICK

- "Quadratic upstream interpolation for convective kinetics"
- Quadratic interpolation, taking into account the flow direction (use 2 upstream / 1 downstream nodes):



On a uniform grid:

$$\text{if } F_w,\, F_e > 0: \quad \phi_w = \frac{-\phi_{WW} + 6\phi_W + 3\phi_P}{8} \qquad \phi_e = \frac{-\phi_W + 6\phi_P + 3\phi_E}{8}$$

$$\text{if } F_w,\, F_e < 0: \quad \phi_w = \frac{3\phi_W + 6\phi_P - \phi_E}{8} \qquad \phi_e = \frac{3\phi_P + 6\phi_E - \phi_{EE}}{8}$$

- Governing equation becomes for $F_w,\, F_e > 0:$

$$\frac{F_e}{8}(-\phi_W+6\phi_P+3\phi_E)-\frac{F_w}{8}(-\phi_{WW}+6\phi_W+3\phi_P) = D_e(\phi_E-\phi_P)-D_w(\phi_P-\phi_W)$$

Steady convection-diffusion: QUICK

- Governing equation becomes for $F_w,\ F_e > 0:$

$$\frac{F_e}{8}(-\phi_W+6\phi_P+3\phi_E)-\frac{F_w}{8}(-\phi_{WW}+6\phi_W+3\phi_P) = D_e(\phi_E-\phi_P)-D_w(\phi_P-\phi_W)$$

- General expression: $a_P\phi_P = a_{WW}\phi_{WW} + a_W\phi_W + a_E\phi_E + a_{EE}\phi_{EE}$

$$a_{WW} = -\max\left(0, \frac{F_w}{8}\right)$$

$$a_W = D_w + \max\left(0, \frac{6F_w}{8}\right) + \max\left(0, \frac{F_e}{8}\right) - \max\left(0, -\frac{3F_w}{8}\right)$$

$$a_E = D_e - \max\left(0, \frac{3F_e}{8}\right) + \max\left(0, -\frac{6F_e}{8}\right) + \max\left(0, -\frac{F_w}{8}\right)$$

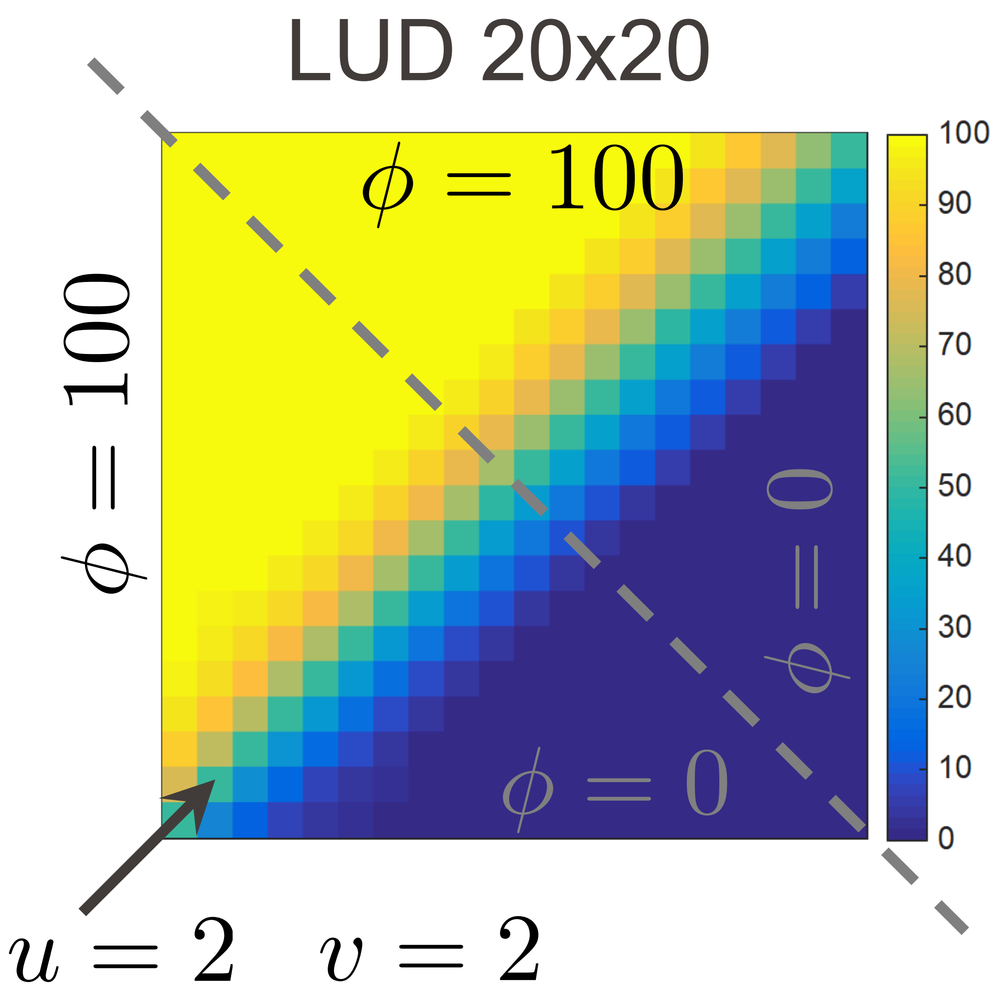
$$a_{EE} = -\max\left(0, -\frac{F_e}{8}\right)$$

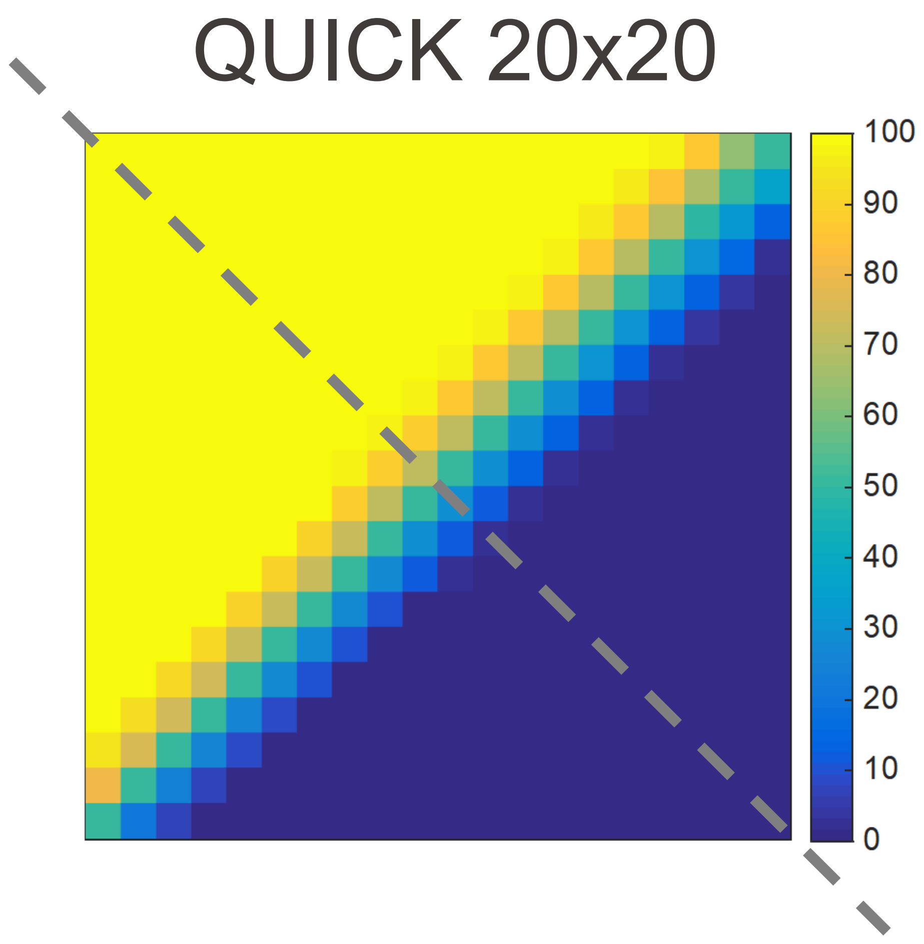
$$a_P = a_{WW} + a_W + a_E + a_{EE} + (F_e - F_w)$$

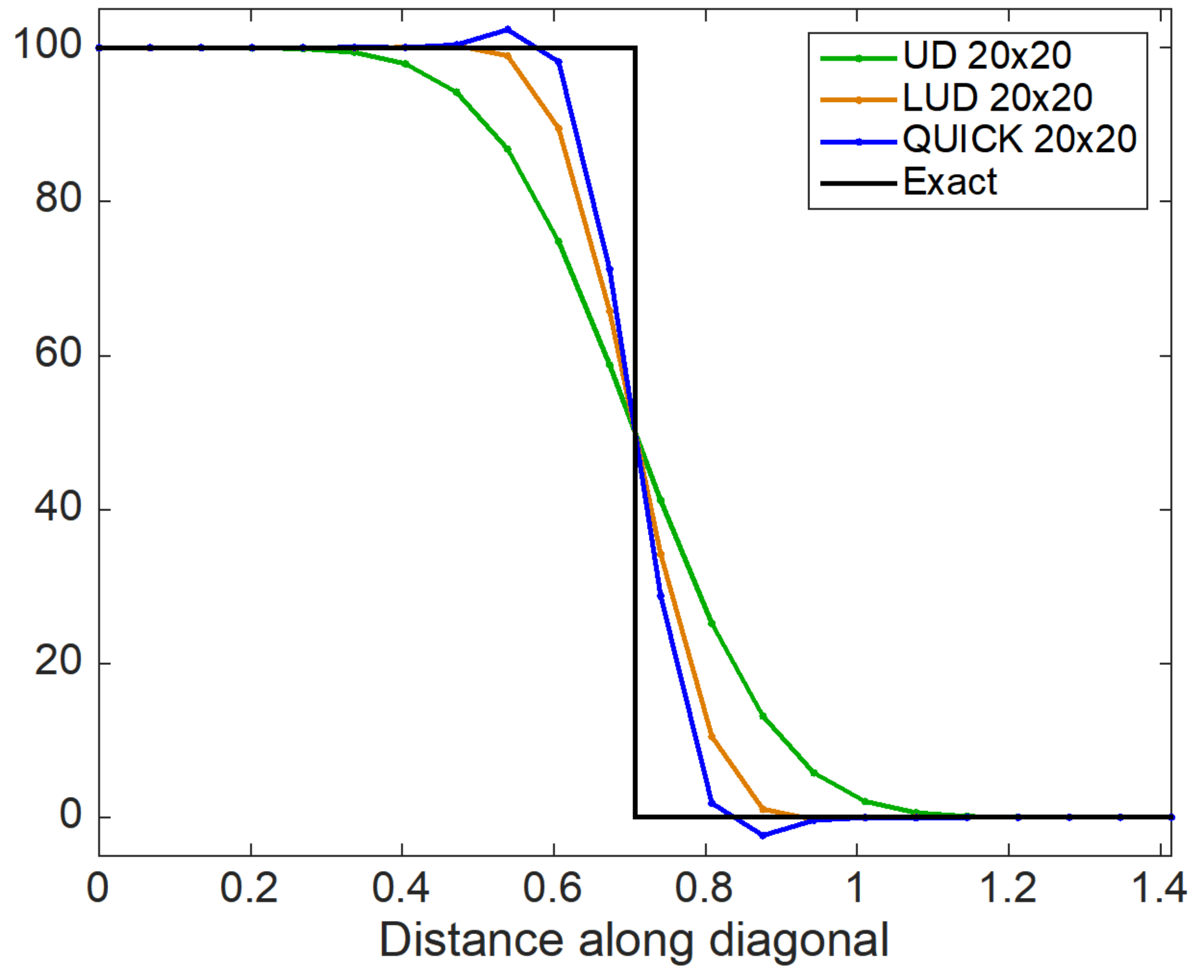
Coefficients can be negative.

Steady convection-diffusion: QUICK

- QUICK is more accurate and has less numerical diffusion, but may produce minor under/overshoots.







Steady convection-diffusion: QUICK

- Bounded?
  - Satisfies the Scarborough criterion
  - Coefficients can be negative
- Transportive?
  - Yes by construction
- Conservative?
  - Yes
- Accuracy: 3rd-order, but may give rise to slight under/overshoots

Deferred correction

- Procedure to facilitate convergence despite negative coefficients: **“deferred correction”.**
- Place troublesome coefficients in the source term, to retain positivity:

Example for $F_w,\ F_e > 0$: rewrite

$$\phi_w = \frac{-\phi_{WW} + 6\phi_W + 3\phi_P}{8} \qquad \phi_e = \frac{-\phi_W + 6\phi_P + 3\phi_E}{8}$$

as

$$\phi_w = \phi_W + \frac{-\phi^*_{WW} - 2\phi^*_W + 3\phi^*_P}{8} \qquad \phi_e = \phi_P + \frac{-\phi^*_W - 2\phi^*_P + 3\phi^*_E}{8}$$

- Governing equation becomes:

$$F_e\left(\phi_P + \frac{-\phi^*_W - 2\phi^*_P + 3\phi^*_E}{8}\right) - F_w\left(\phi_W + \frac{-\phi^*_{WW} - 2\phi^*_W + 3\phi^*_P}{8}\right) = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)$$

Only positive coefficients:

$$\boxed{(F_e + D_e + D_w)}\phi_P = \boxed{D_e}\phi_E + \boxed{(D_w + F_w)}\phi_W + \boxed{S_{dc}}$$

$$\boxed{a_P}\phi_P = \boxed{a_E}\phi_E + \boxed{a_W}\phi_W + \boxed{S_{dc}}$$

Deferred correction source term:

$$S_{dc} = -\frac{F_e}{8}(-\phi^*_W - 2\phi^*_P + 3\phi^*_E) + \frac{F_w}{8}(-\phi^*_{WW} - 2\phi^*_W + 3\phi^*_P)$$

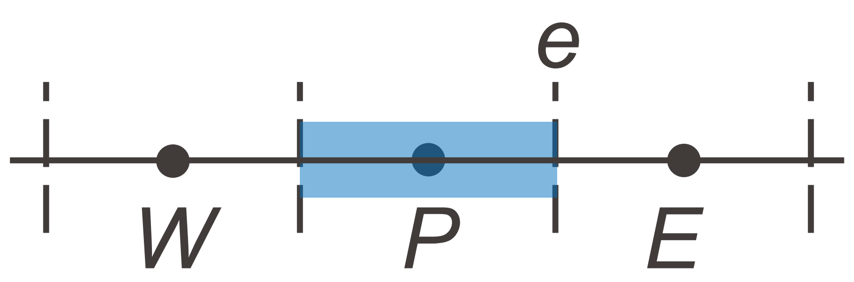
- Use guess values for starred quantities, and iterate until convergence.

Summary of discretization schemes (convective term)

- UD: bounded, transportive, but 1st-order (numerical diffusion)
- CD: 2nd-order, but not transportive and can be unbounded (oscillations)
- LUD, QUICK: 2nd/3rd-order, transportive, but can be unbounded (oscillations)

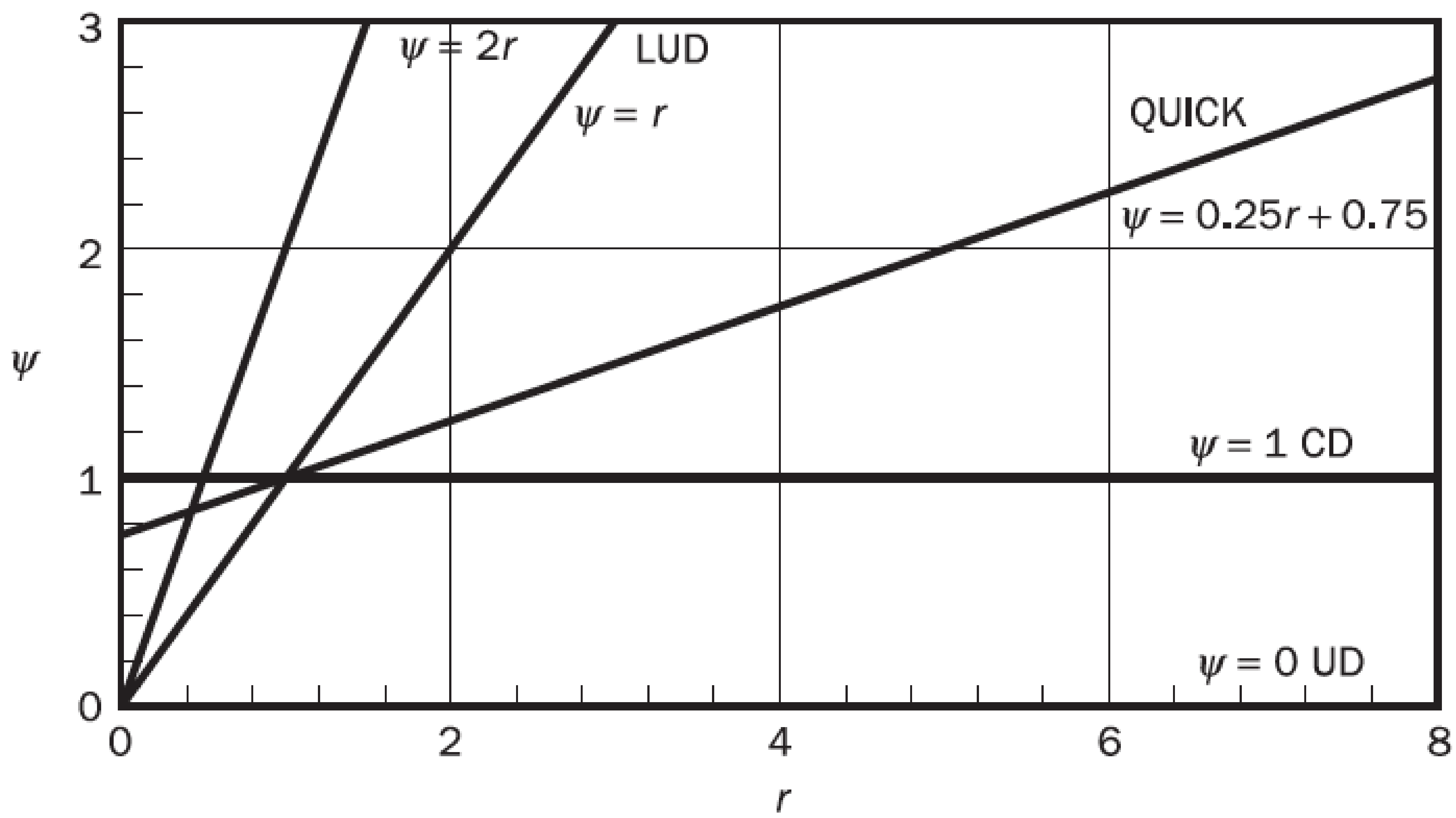
- Would like higher order without oscillations.

- Observation: higher-order schemes can be written as an extension of UD:



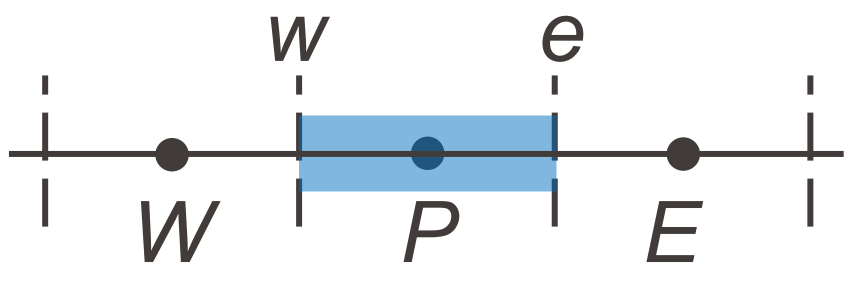
  - UD: $\phi_e = \boxed{\phi_P}$ $\qquad \psi(r) = 0$
  - CD: $\phi_e = \dfrac{\phi_P + \phi_E}{2} = \boxed{\phi_P} + \boxed{\dfrac{1}{2}(\phi_E - \phi_P)}$ $\qquad \psi(r) = 1$
  - LUD: $\phi_e = \phi_P + \dfrac{1}{2}(\phi_P - \phi_W) = \boxed{\phi_P} + \boxed{\dfrac{1}{2}\left(\dfrac{\phi_P - \phi_W}{\phi_E - \phi_P}\right)(\phi_E - \phi_P)}$ $\qquad \psi(r) = r$
  - QUICK: $\phi_e = \phi_P + \dfrac{1}{8}(-\phi_W - 2\phi_P + 3\phi_E) = \boxed{\phi_P} + \boxed{\dfrac{1}{2}\left[\dfrac{1}{4}\left(3 + \dfrac{\phi_P - \phi_W}{\phi_E - \phi_P}\right)\right](\phi_E - \phi_P)}$ $\qquad \psi(r) = \dfrac{3+r}{4}$
  - General form: $\phi_e = \boxed{\phi_P} + \boxed{\dfrac{\psi(r)}{2}(\phi_E - \phi_P)}$ where $r = \dfrac{\phi_P - \phi_W}{\phi_E - \phi_P}$

Summary of discretization schemes (convective term)



- UD: $\phi_e = \boxed{\phi_P}$ — $\psi(r) = 0$
- CD: $\phi_e = \dfrac{\phi_P + \phi_E}{2} = \boxed{\phi_P} + \boxed{\dfrac{1}{2}(\phi_E - \phi_P)}$ — $\psi(r) = 1$
- LUD: $\phi_e = \phi_P + \dfrac{1}{2}(\phi_P - \phi_W) = \boxed{\phi_P} + \boxed{\dfrac{1}{2}\left(\dfrac{\phi_P - \phi_W}{\phi_E - \phi_P}\right)(\phi_E - \phi_P)}$ — $\psi(r) = r$
- QUICK: $\phi_e = \phi_P + \dfrac{1}{8}(-\phi_W - 2\phi_P + 3\phi_E) = \boxed{\phi_P} + \boxed{\dfrac{1}{2}\left[\dfrac{1}{4}\left(3 + \dfrac{\phi_P - \phi_W}{\phi_E - \phi_P}\right)\right](\phi_E - \phi_P)}$ — $\psi(r) = \dfrac{3 + r}{4}$
- General form: $\phi_e = \boxed{\phi_P} + \boxed{\dfrac{\psi(r)}{2}(\phi_E - \phi_P)}$ where $r = \dfrac{\phi_P - \phi_W}{\phi_E - \phi_P}$

TVD schemes



- General form: upwind flux + correction flux (related to gradient $\left.\frac{\partial \phi}{\partial x}\right|_e \approx \frac{\phi_E - \phi_P}{\Delta x}$)

$$\phi_e = \boxed{\phi_P} + \boxed{\frac{\psi(r)}{2}(\phi_E - \phi_P)} \qquad r = \frac{\phi_P - \phi_W}{\phi_E - \phi_P}$$

ratio of upstream to downstream gradients

- Can also be seen as a linear combination of UD and CD:

$$\phi_e = (1 - \psi(r))\boxed{\phi_e^{UD}} + \psi(r)\boxed{\phi_e^{CD}} = (1 - \psi(r))\boxed{\phi_P} + \psi(r)\boxed{\left(\frac{\phi_P + \phi_E}{2}\right)}$$

- Idea of “total-variation diminishing” **(TVD)** schemes: evaluate the convective flux with the above form, choosing a suitable $\psi(r)$ to achieve higher-order accuracy without introducing new extrema.

TVD schemes

- Criteria for boundedness:

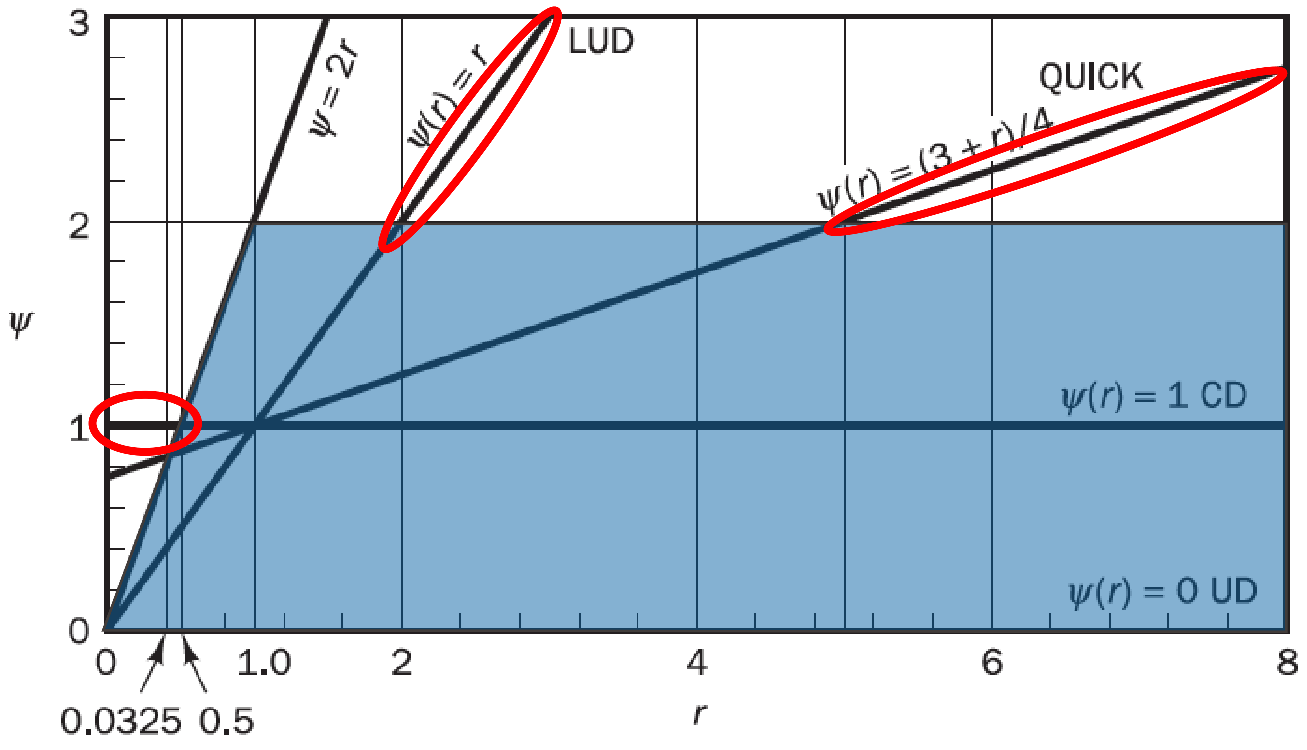
$$\psi(r) \le 2r \text{ for } 0 < r < 1$$
$$\psi(r) \le 2 \text{ for } 1 \le r$$

$\psi(r)$ is called “flux limiter” since it limits the higher-order correction flux:

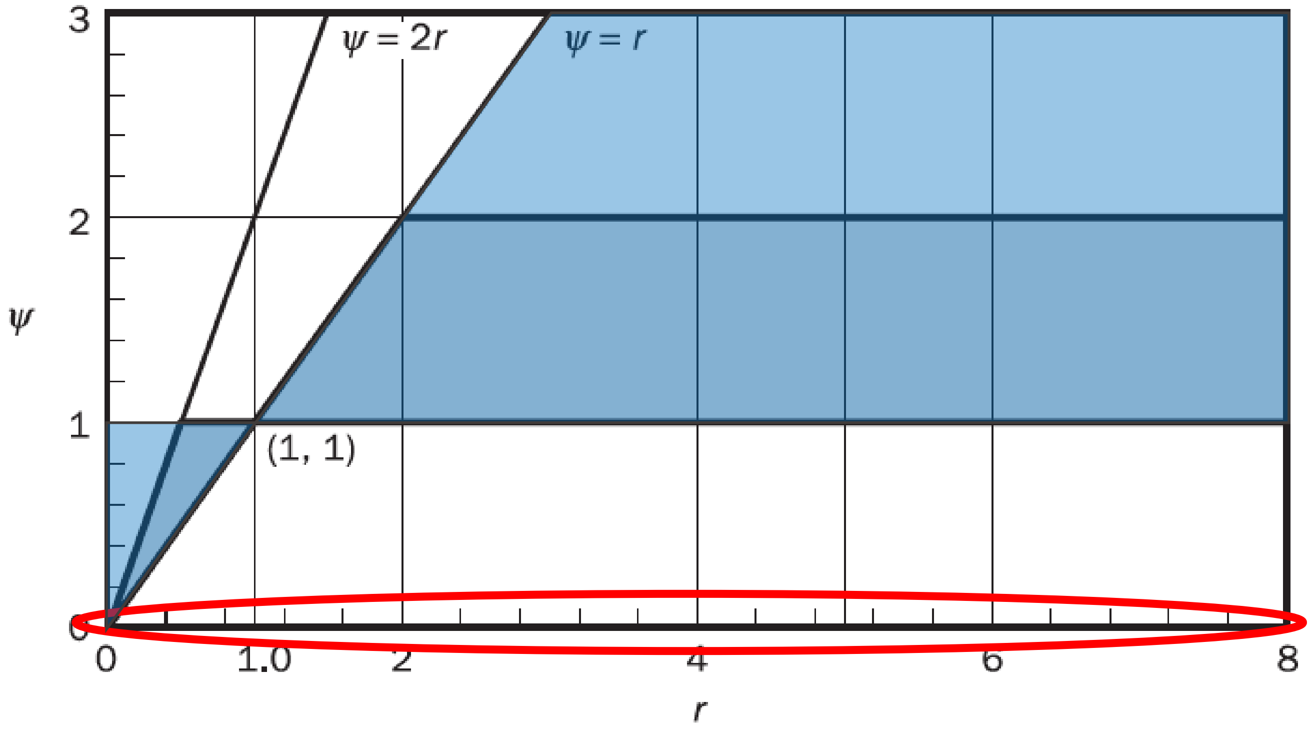
$$\phi_e = \phi_P + \boxed{\frac{\psi(r)}{2}}(\phi_E - \phi_P)$$

- Criteria for 2nd-order accuracy:

$$\psi(1) = 1$$
$$r \le \psi(r) \le 1 \text{ for } 0 < r < 1$$
$$1 \le \psi(r) \le r \text{ for } 1 \le r$$



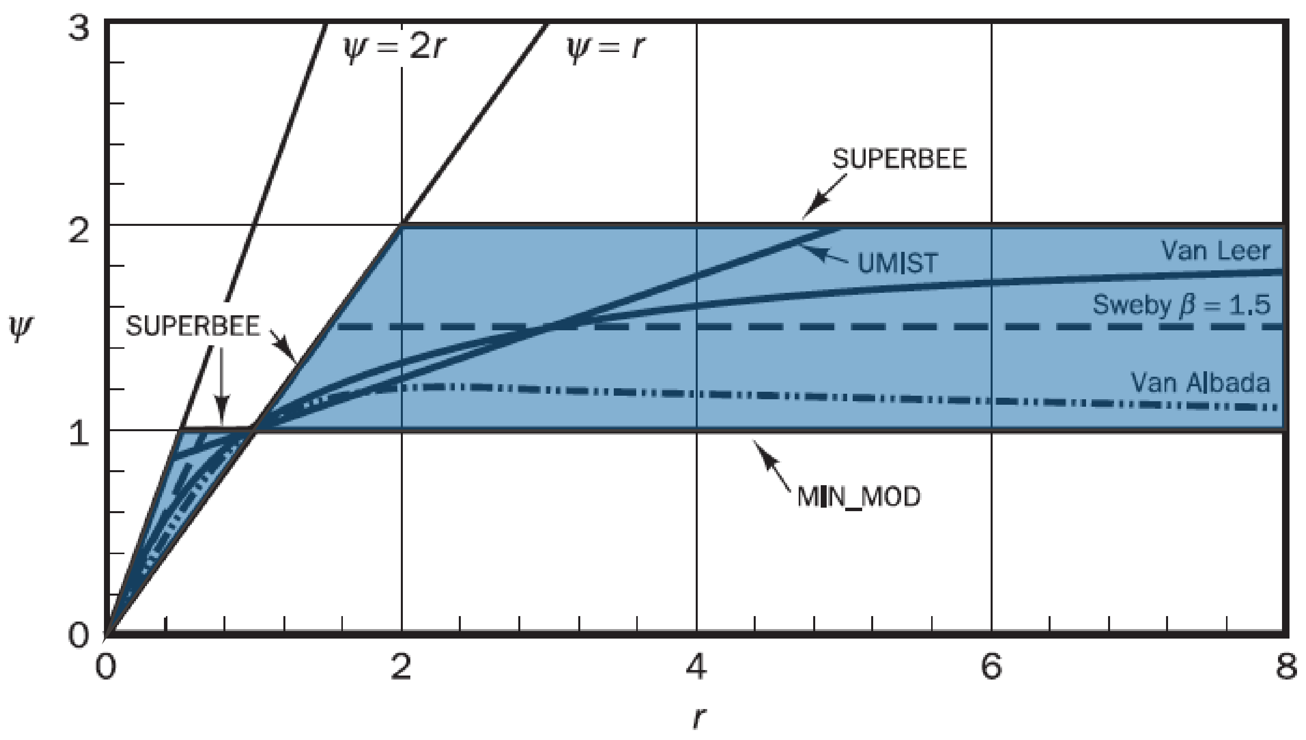
Check: CD, LUD and QUICK are not bounded for some values of *r*.



Check: UD is not 2nd-order accurate.

TVD schemes

- Some examples:



Name	*Limiter function ψ(r)*	*Source*
Van Leer	$\dfrac{r+\lvert r\rvert}{1+r}$	Van Leer (1974)
Van Albada	$\dfrac{r+r^2}{1+r^2}$	Van Albada *et al.* (1982)
Min-Mod	$\psi(r)=\begin{cases}\min(r,1) & \text{if } r>0\\ 0 & \text{if } r\le 0\end{cases}$	Roe (1985)
SUPERBEE	$\max[0, \min(2r, 1), \min(r, 2)]$	Roe (1985)
Sweby	$\max[0, \min(\beta r, 1), \min(r, \beta)]$	Sweby (1984)
QUICK	$\max[0, \min(2r, (3+r)/4, 2)]$	Leonard (1988)
UMIST	$\max[0, \min(2r, (1+3r)/4, (3+r)/4, 2)]$	Lien and Leschziner (1993)

- Governing equation becomes for $F_w, F_e > 0$:

$$r_e = \frac{\phi_P - \phi_W}{\phi_E - \phi_P} \quad r_w = \frac{\phi_W - \phi_{WW}}{\phi_P - \phi_W}$$

$$F_e\left(\phi_P + \frac{\psi(r_e)}{2}(\phi_E - \phi_P)\right) - F_w\left(\phi_W + \frac{\psi(r_w)}{2}(\phi_P - \phi_W)\right) = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)$$

- With deferred correction approach:

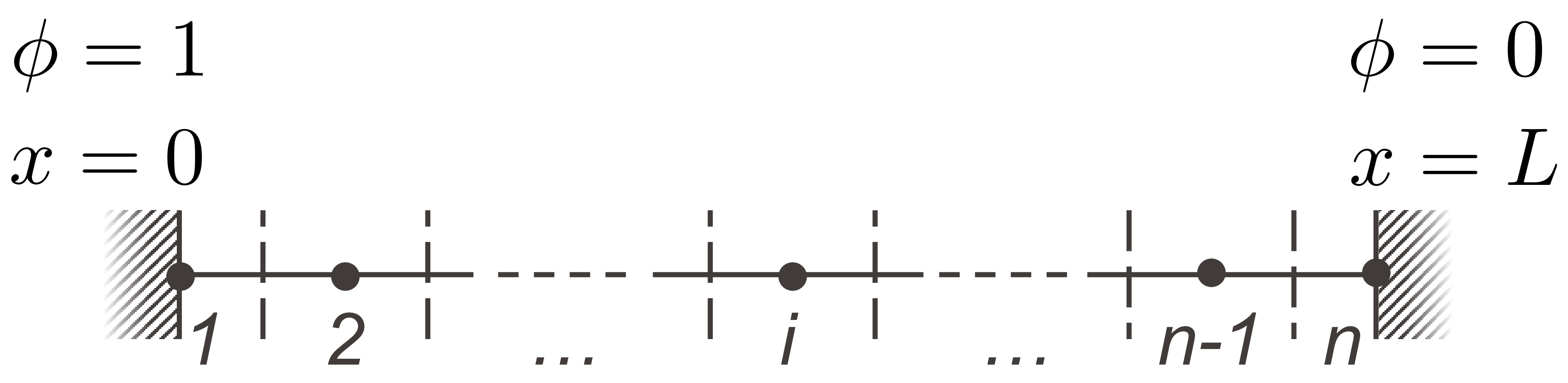
$$(D_e + D_w + F_e)\,\phi_P = D_e\phi_E + (D_w + F_w)\phi_W + \left[F_w\frac{\psi(r_w^*)}{2}(\phi_P^* - \phi_W^*) - F_e\frac{\psi(r_e^*)}{2}(\phi_E^* - \phi_P^*)\right]$$

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + S_{dc}$$

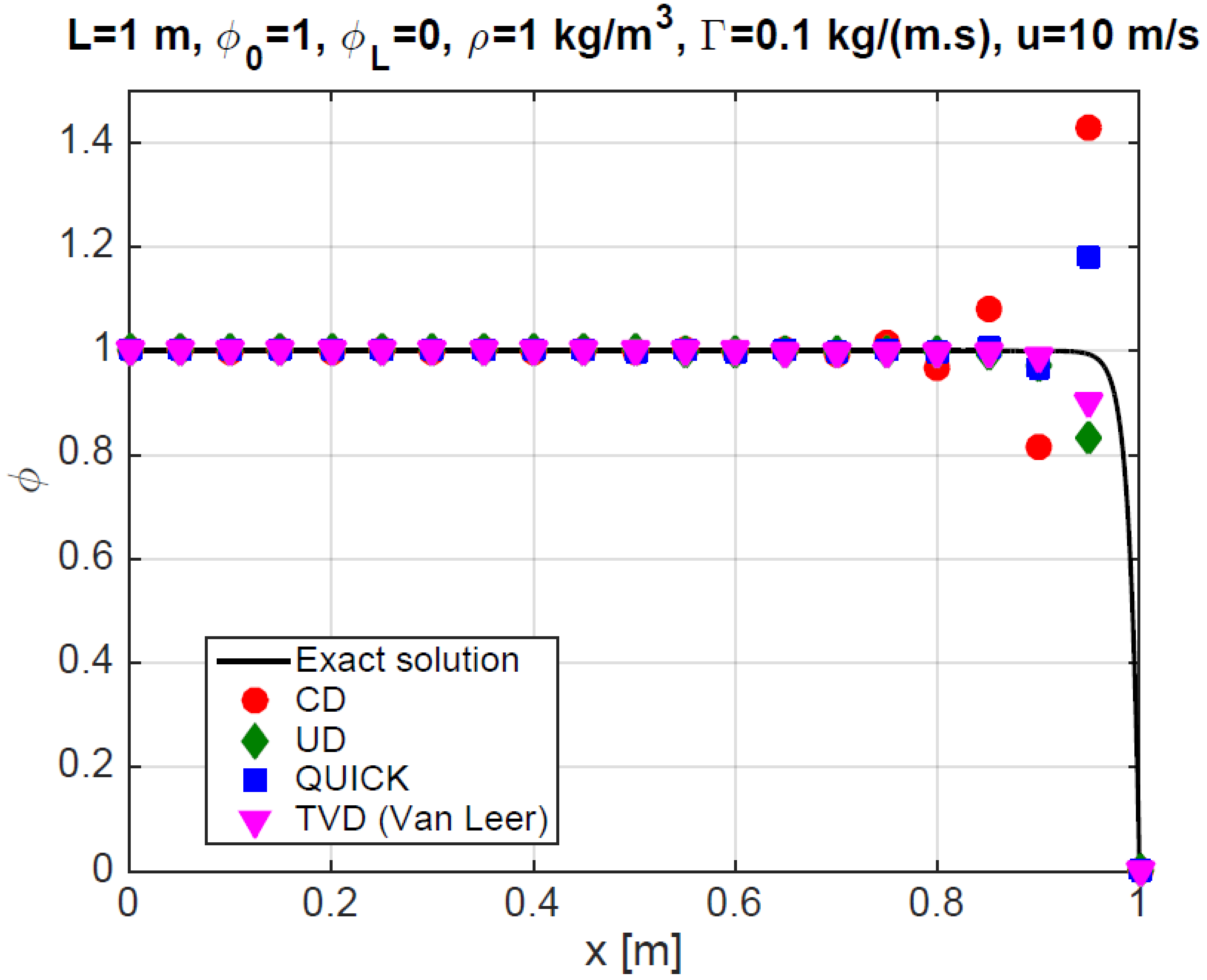
TVD schemes

- 1D example: domain $[0, 1]$ m

$$\rho = 1\ \mathrm{kg/m^3},\ \Gamma = 0.1\ \mathrm{kg/(m.s)}$$



$$u = 10\ \mathrm{m/s},\ n = 21$$

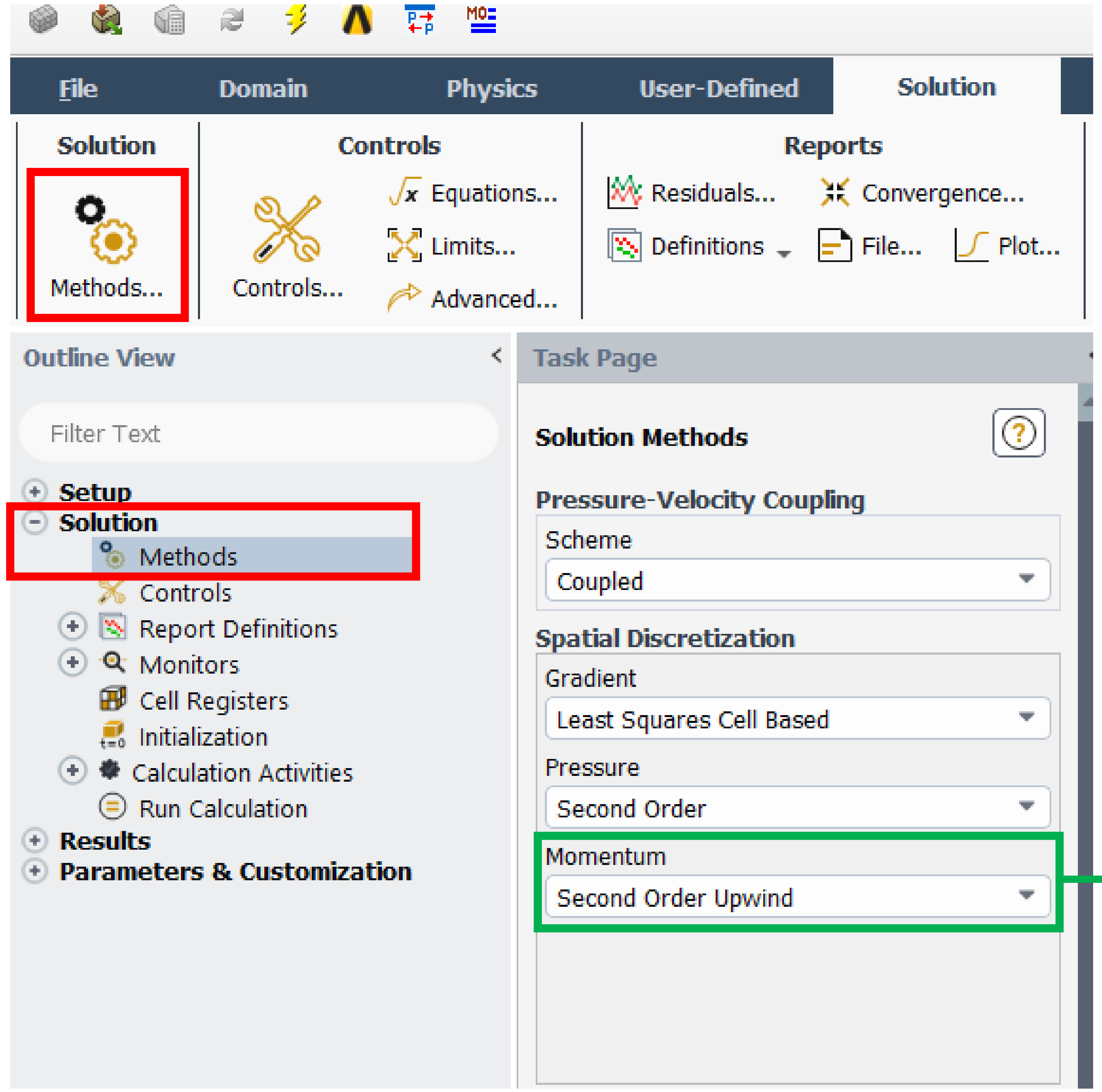


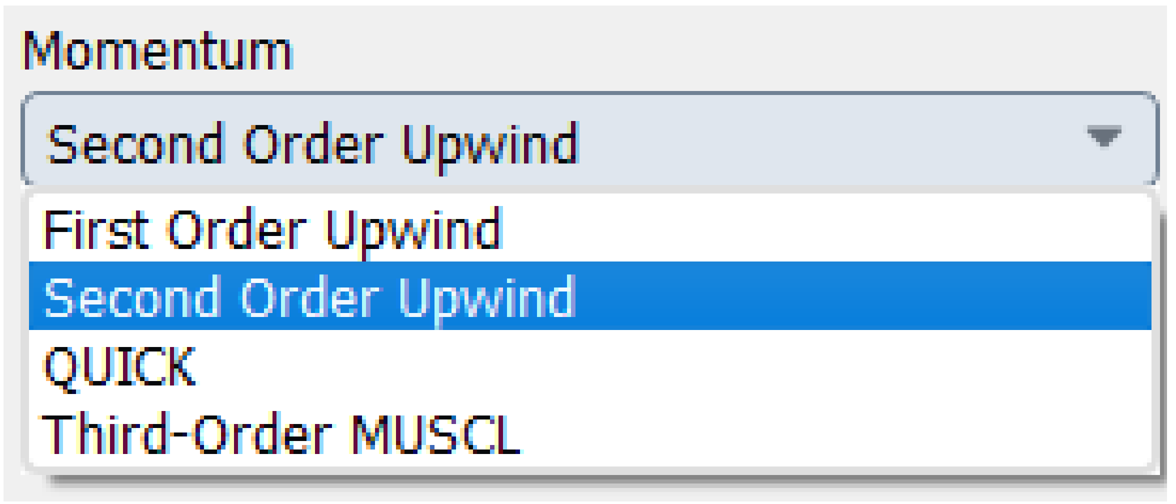
Summary and guidelines

- **CD** can be used for the **diffusion** term, but rarely for the convection term.
- For the **convection** term, **UD** most stable, but large numerical diffusion. Can be used to get an initial solution, before switching to a higher-order scheme.
- **LUD (SOU)** and **QUICK** more accurate.
- **TVD** schemes avoid oscillations.

Appendix: Fluent specifics – Discretization schemes

- Discretization schemes available: UD, LUD (SOU), QUICK and MUSCL.

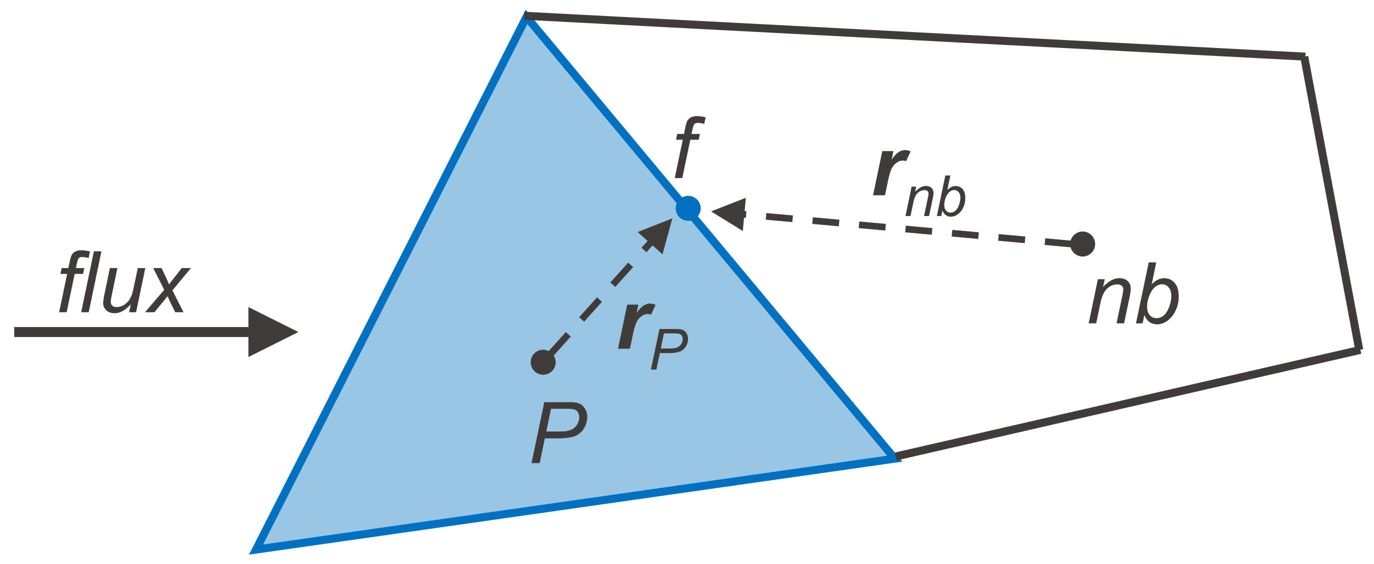




Appendix: Steady convection-diffusion: MUSCL

- Fluent has a 3rd-order, "QUICK-like" scheme for **unstructured** meshes (where it may not always be possible to uniquely identify "upstream" and "downstream" nodes): **MUSCL** ("monotone upstream-centered scheme for conservation laws").

- Blends the two 2nd-order schemes CD and LUD:



$$\phi_f = \theta \phi_f^{CD} + (1-\theta)\phi_f^{LUD}$$

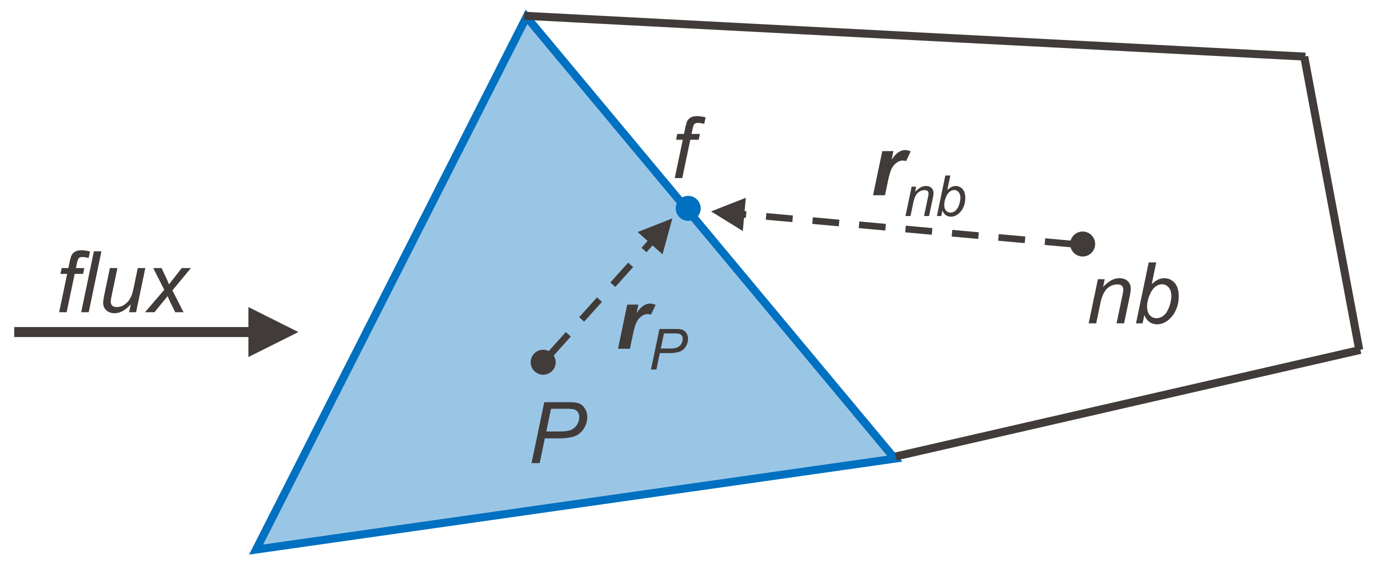
$$\phi_f^{LUD} = \phi_P + \nabla\phi_P \cdot \mathbf{r}_P \quad \text{(if } P \text{ upstream)}$$

$$\phi_f^{CD} = \frac{1}{2}\left[(\phi_P + \nabla\phi_P \cdot \mathbf{r}_P) + (\phi_{nb} + \nabla\phi_{nb} \cdot \mathbf{r}_{nb})\right]$$

Appendix: Steady convection-diffusion: MUSCL

- Fluent has a 3rd-order, "QUICK-like" scheme for **unstructured** meshes (where it may not always be possible to uniquely identify "upstream" and "downstream" nodes): **MUSCL** ("monotone upstream-centered scheme for conservation laws").

- Blends the two 2nd-order schemes CD and LUD:



$$\phi_f = \theta\phi_f^{CD} + (1-\theta)\phi_f^{LUD}$$

$$\phi_f^{LUD} = \phi_P + \boxed{\nabla\phi_P} \cdot \mathbf{r}_P \qquad \text{(if } P \text{ upstream)}$$

$$\phi_f^{CD} = \frac{1}{2}\left[\left(\phi_P + \boxed{\nabla\phi_P} \cdot \mathbf{r}_P\right) + \left(\phi_{nb} + \boxed{\nabla\phi_{nb}} \cdot \mathbf{r}_{nb}\right)\right]$$

- Note: on unstructured meshes, all schemes except UD require evaluating the gradient in the CV from neighboring nodal values (→ additional cost).

Appendix: Fluent specifics – Gradient reconstruction

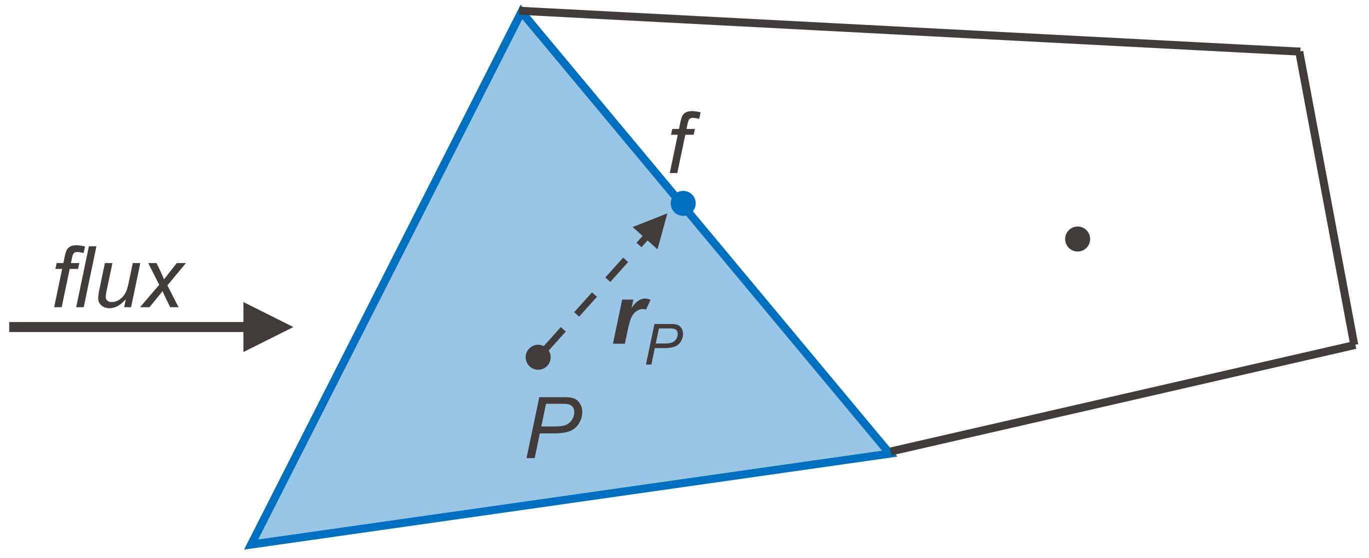
- Nodal gradient reconstruction:
  - We saw (week 2) that for the **diffusive term** we need to evaluate the **gradient on faces:**

$$\int_V div(\Gamma\, grad(\phi))\, dV = \oint_{A_i} \Gamma\, \boxed{grad(\phi)} \cdot \mathbf{n}\, dA$$

  - For the **convective term**, it seems we just need the solution on faces, but if we want better accuracy than 1st-order UD, we actually need to evaluate the **gradient at nodes** (slide 24):

$$\int_V div(\rho\phi\mathbf{u})\, dV = \oint_{A_i} \rho\underline{\phi}\mathbf{u} \cdot \mathbf{n}\, dA$$

For ex. (LUD): $\underline{\phi_f^{LUD}} = \phi_P + \boxed{\nabla\phi_P} \cdot \mathbf{r}_P$



- Two methods to compute the nodal gradient:
  1. “Green-Gauss”
  2. Least squares

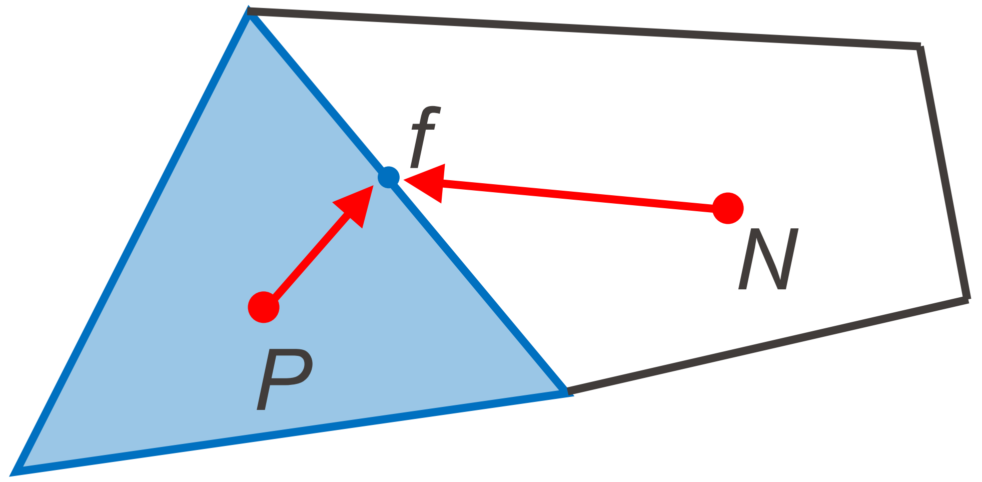
Appendix: Fluent specifics – Gradient reconstruction

1. From the **divergence theorem**: $$\int_V \nabla\phi\, dV = \int_V div(\phi\mathbf{I})\, dV = \oint_A \phi\mathbf{n}\, dA$$

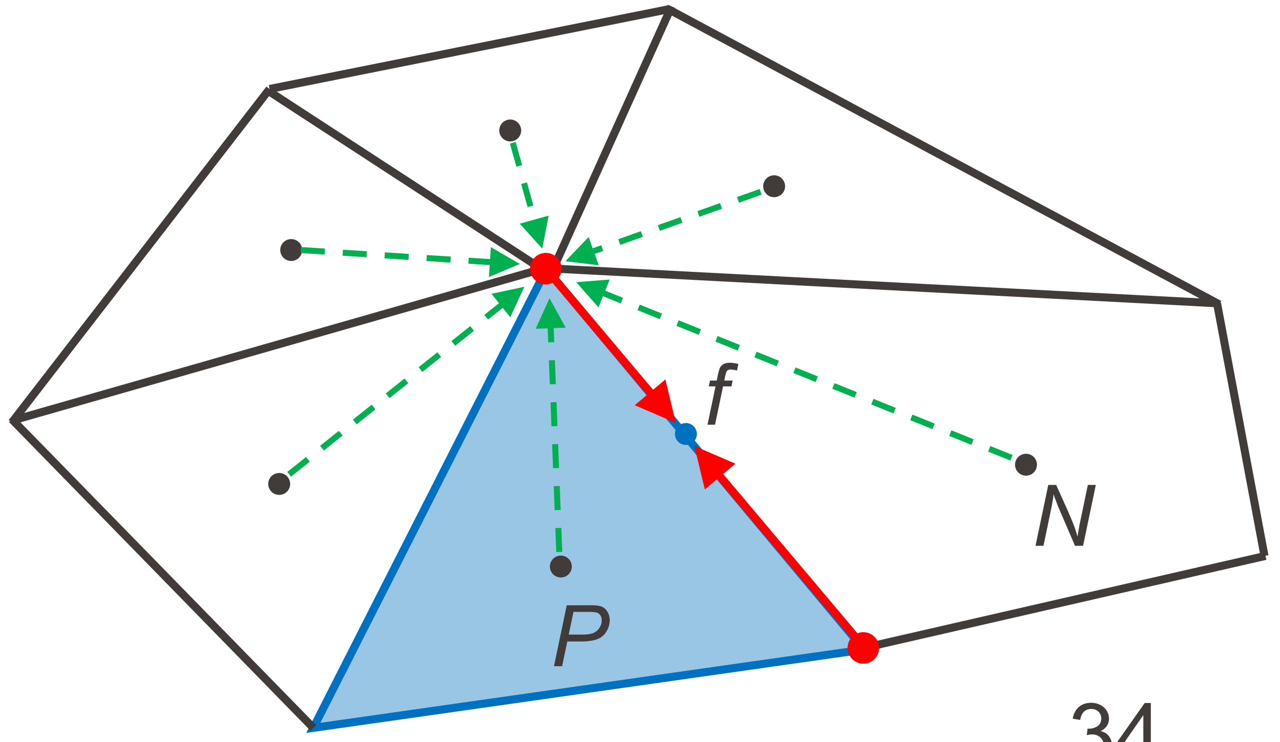
So, since $\int_V \nabla\phi\, dV \approx \nabla\phi_P\, V$, the **nodal gradient** can be approximated as: $$\nabla\phi_P \approx \frac{1}{V}\sum_f \phi_f \mathbf{n}_f A_f$$

Here, each **face value** ϕ_f can be evaluated as:

- the mean of the 2 neighboring cell values ("Green-Gauss, cell based"), $$\phi_f \approx \frac{1}{2}(\phi_P + \phi_N)$$



- or the mean of the N_v face vertex values ("Green-Gauss, node based"), each of them evaluated as a weighted average of all surrounding cell values.
  More accurate, but more expensive. $$\phi_f \approx \frac{1}{N_v}\sum_{v=1}^{N_v} \phi_v$$

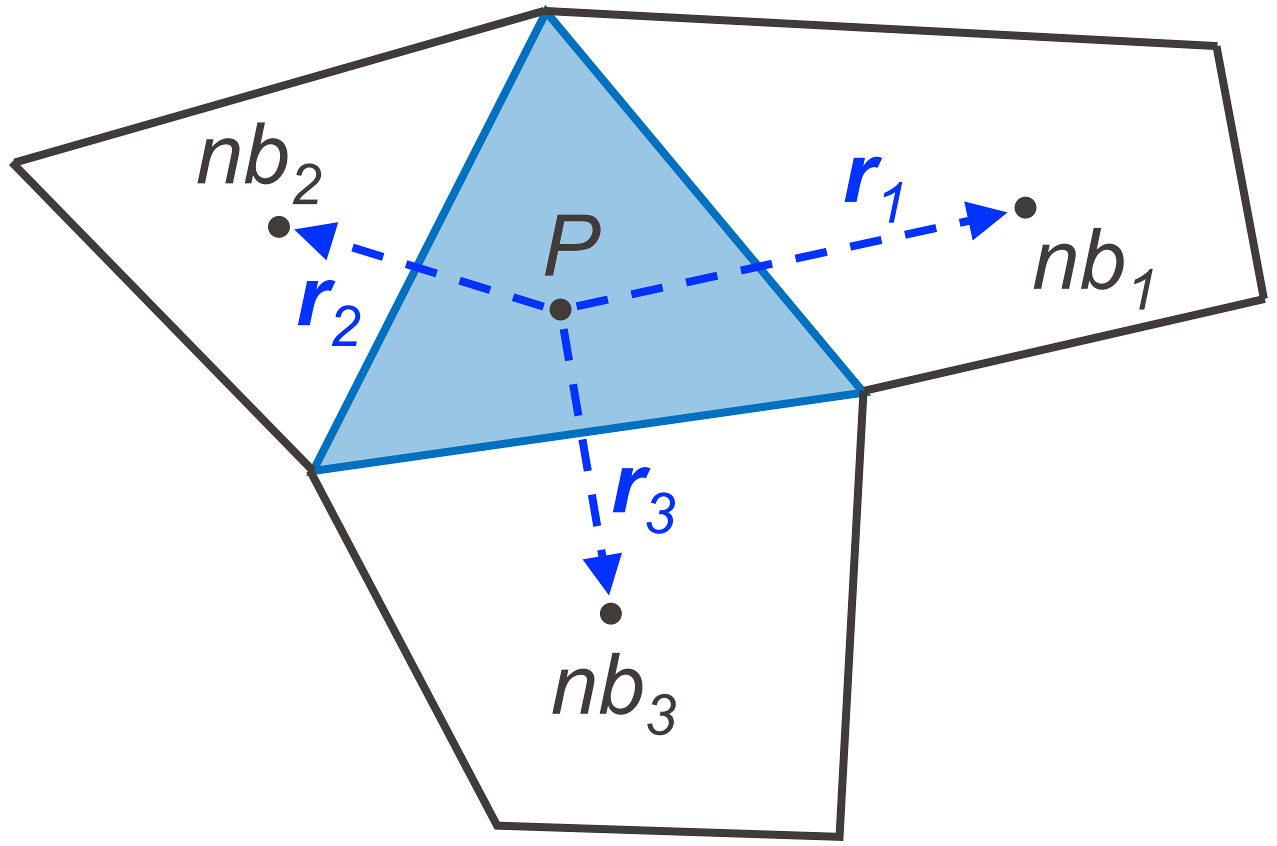


Numerical Flow Simulation

Appendix: Fluent specifics – Gradient reconstruction

2. Least squares: in each neighboring node nb_i, write the solution as a Taylor expansion about *P:*

$$\begin{cases} \phi_{nb_1} = \phi_P + \nabla\phi_P \cdot \mathbf{r}_1 \\ \phi_{nb_2} = \phi_P + \nabla\phi_P \cdot \mathbf{r}_2 \\ \qquad \dots \end{cases}$$



This is a small (N_{nb} x 3) overdetermined linear system to be solved for the gradient:

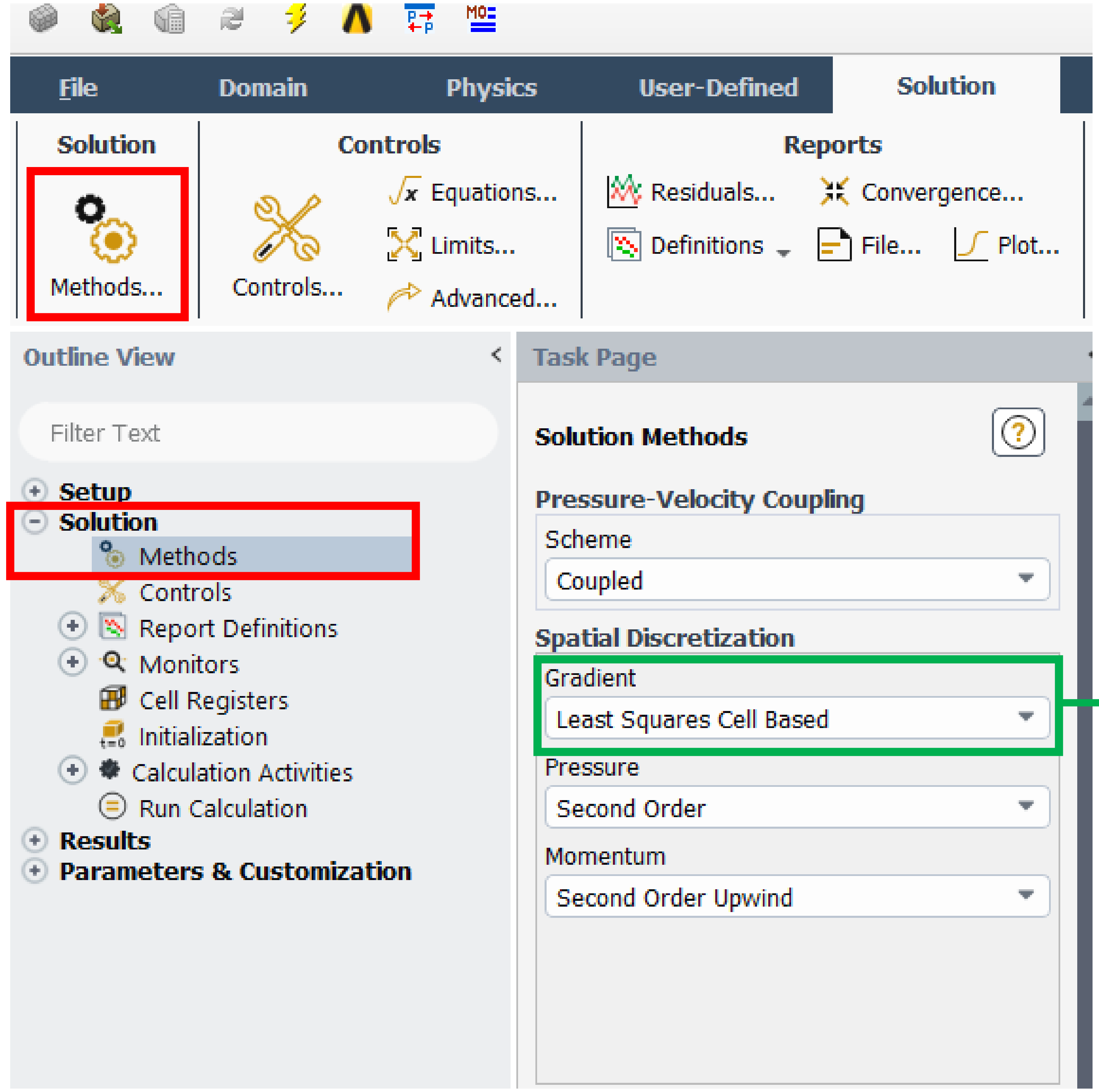
$$\mathbf{J}\,\nabla\phi_P = \boldsymbol{\phi_{nb}} - \boldsymbol{\phi_P}$$

$$\begin{bmatrix} r_{1,x} & r_{1,y} & r_{1,z} \\ r_{2,x} & r_{2,y} & r_{2,z} \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} \nabla\phi_{P,x} \\ \nabla\phi_{P,y} \\ \nabla\phi_{P,z} \end{pmatrix} = \begin{pmatrix} \phi_{nb_1} - \phi_P \\ \phi_{nb_2} - \phi_P \\ \vdots \end{pmatrix}$$

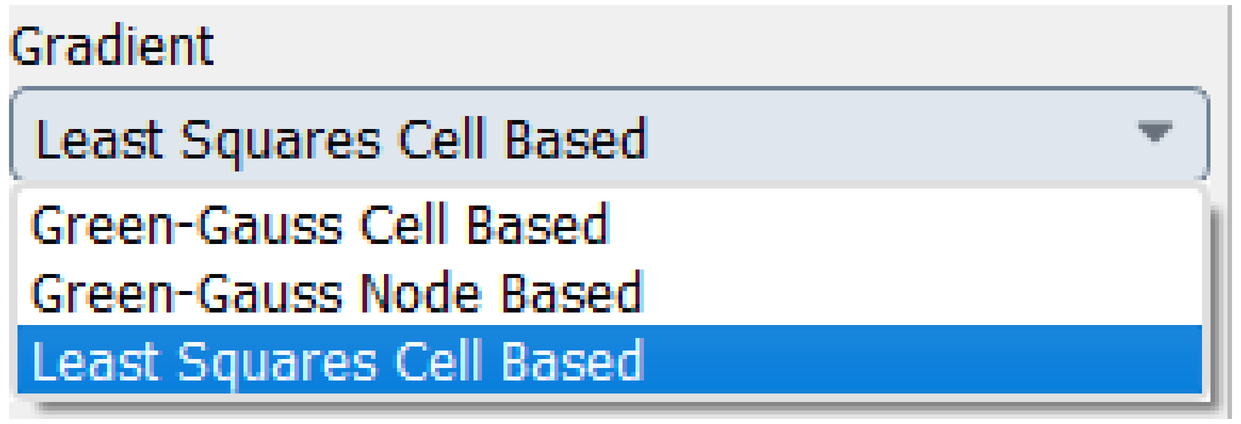
As accurate as the “Green-Gauss node based” reconstruction, but faster. Default setting in Fluent.

Appendix: Fluent specifics – Gradient reconstruction

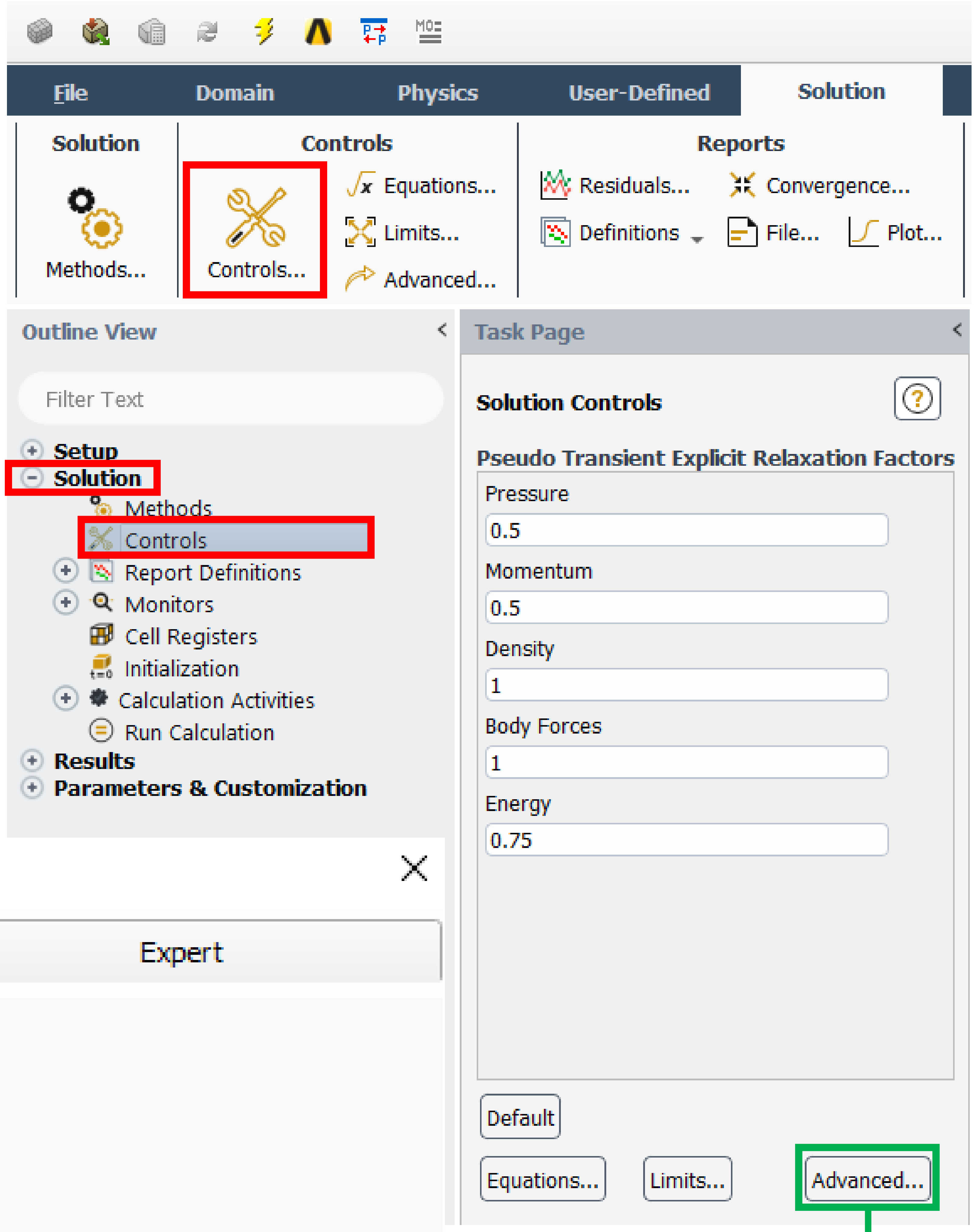
- Nodal gradient reconstruction:



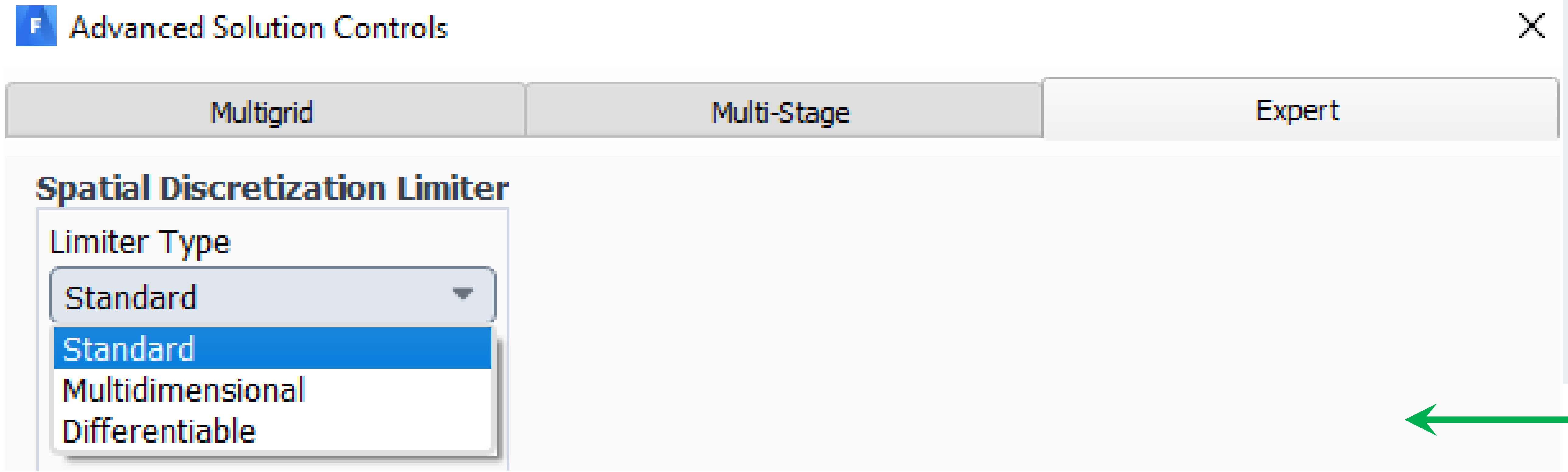
- Green-Gauss, cell based,
- Green-Gauss, node based,
- Least squares (default)



Appendix: Fluent specifics – TVD



- TVD: can choose among
  - standard Min-Mod (default; limits gradient in all directions),
  - multi-dimensional Min-Mod (limits normal gradient only → less dissipative),
  - differentiable limiter (helps to avoid residual convergence “stall”).



- … but you don’t really need to modify that.