# FVNS.1 The Class of FVS$^h$-Implemented PPNS Algorithms

## Commercial CFD codes for INS employ FVS$^h$ PPNS approximations

e.g., Fluent, CFX, Star-CD, Flow 3D

## Theory requirement

D$M^h$: $\nabla \cdot \mathbf{u} = 0 \Rightarrow \left\| \nabla^h \cdot \mathbf{u}^h \right\| \le \varepsilon > 0$ iteratively

"famous" named algorithms with origination

MAC, SMAC – Los Alamos Nat. Lab
SIMPLE,- ER, -EC, -EST – Imperial College, UK
PISO – Imperial College, UK

each is an iterative strategy for approximating the theory requirement

1. measure D$M^h$ error $\nabla^h \cdot \mathbf{u}^h$ via a potential function $\phi^h$
2. employ $\phi^h$ to moderate D$\mathbf{P}^h$ error via
   - velocity corrections
   - pressure corrections
3. iterate D$\mathbf{P}^h$ with D$M^h$ by relaxing corrections (stability)
4. stop when $\mathrm{FVS}^h(\mathcal{L}(q^h)) \Rightarrow \left| \{\mathrm{RES}\}_{\max} \right| < \delta$

# FVNS.2 Generic FVS$^h$ Algorithm for PPNS Form for INS

## Reynolds transport theorem generates typical $\mathcal{L}(q)$, hence

$$\mathrm{D}(\cdot):\ \mathcal{L}(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial x_j}\left(f_j - f_j^{\,r}\right) - s = 0,\ \text{ on } \Omega \subset \Re^n$$

$$\mathrm{FVS}^h \equiv \int_\Omega \mathcal{L}(q^h)\,\mathrm{d}\tau \equiv \sum_{\Omega^h}\int_{\Omega_v}\left(\frac{\partial q}{\partial t} - \frac{\partial}{\partial x_j}\left(f_j - f_j^{\,v}\right) - s\right)\mathrm{d}\tau \equiv 0$$

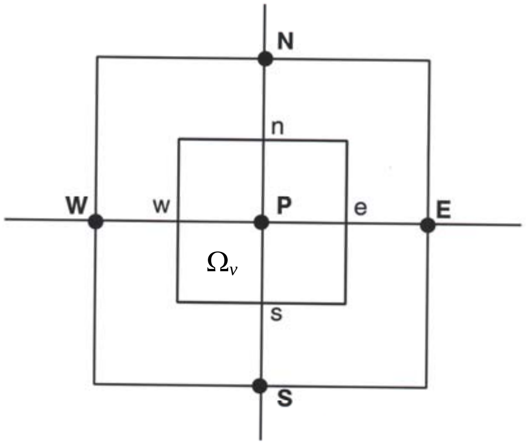
$$= \sum_{\Omega^h}\left[\int_{\Omega_v}\left(\frac{\partial q^h}{\partial t} - s\right)\mathrm{d}\tau - \oint_{\partial\Omega_v}\left(f_j - f_j^{\,v}\right)^h \cdot \hat{\mathbf{n}}_j\,\mathrm{d}\sigma\right] = 0$$

## Use of divergence theorem necessitates "staggered" meshing

generic FV notation :

$$\int_{\Omega_v}(\cdot)\,\mathrm{d}\tau \Rightarrow (\ \ )_{\mathrm{P}} \cdot h^n$$

$$\oint_{\partial\Omega_v}(\cdot)\cdot\hat{\mathbf{n}}\,\mathrm{d}\sigma \Rightarrow \sum_{\alpha=e,w,n,s}\Delta_\alpha(\cdot)(\pm)\,h^{n-1}$$



# FVNS.3 FVS$^h$ CFD Algorithm Notation, Fluxes

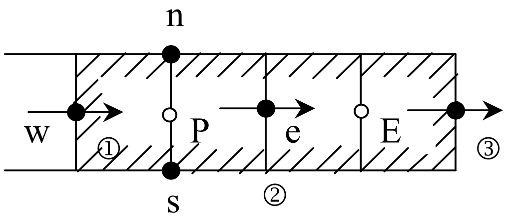
## SIMPLE – type FV notation for $\mathcal{L}(q^h)$, $n$ = 2: (Patankar, 1980, Ch.5)

$$\mathrm{FVS}^h\left(\mathcal{L}(q^h)\right) \Rightarrow a_P Q_P = a_E Q_E - a_W Q_W + a_N Q_N - a_S Q_S + S_P$$
$$= \sum_{B=1}^{4} a_B Q_B + S_P$$

## FVS$^h$ for DP$^h$ mixes capital and lower case node evaluations

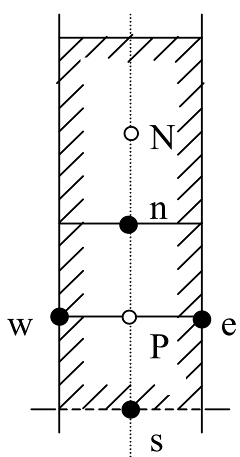
for $\mathrm{DP}_x$ : $\hat{\mathbf{i}}$ – flux for P evaluated at edge e, $n = 2$

$$\mathrm{FVS}^h(\mathcal{L}(u^h)) \Rightarrow a_e u_e = \sum_{b=1}^{3} a_b u_b + (p_P - p_E)\ell_e + s_P$$




for $\mathrm{DP}_y$ : $\hat{\mathbf{j}}$ – flux for P evaluated at edge n, $n = 2$

$$\mathrm{FVS}^h(\mathcal{L}(v^h)) \Rightarrow a_n v_n = \sum_{b=1}^{3} a_b v_b + (p_P - p_N)\ell_n + s_P$$



# FVNS.4 FVS$^h$ Pressure-Velocity Corrections

**FVS$^h$ solution for $(u, v, w)^h$ does not satisfy** $\mathrm{D}M^h \Rightarrow \left\| \nabla^h \cdot \mathbf{u}^h \right\| < \varepsilon$

predictor-corrector strategy:

$$\mathbf{u}^{n+1} = \mathbf{u}^h + \mathbf{u}'$$
$$p^{n+1} = p^h + p'$$

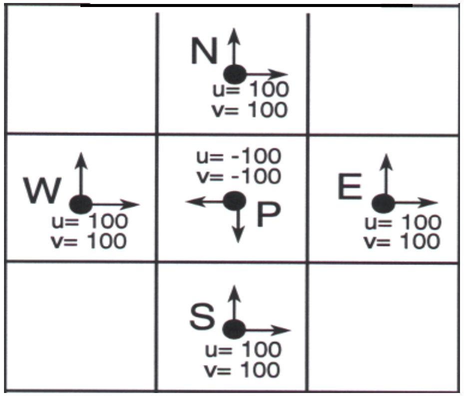
**DP$^h$ SIMPLE-type correction procedure**

$$\text{solve}: \ \nabla^2 p' = \nabla \cdot \mathbf{u}^h \quad \text{, for BCs} \ \ \hat{\mathbf{n}} \cdot \nabla \phi = 0$$
$$\text{update pressure}: \ p^{n+1} \equiv p^h + p'$$
$$\text{correct velocity}: \ \mathbf{u}^{n+1} = \mathbf{u}^h + \nabla p'$$
$$\text{under-relax } \mathbf{u}', p', \text{resolve FVS}^h(\mathcal{L}(\mathbf{u}^h)) \text{ until } \mathbf{u}', p' \text{ small enough}$$

**Central differences can produce spurious** $\mathrm{D}M^h$

example: $\nabla^h \cdot \mathbf{u}_\mathrm{p} = 0$ for $2\Delta x$ error mode

leads to upwinding, QUICK differencing
dissipate dispersion error



# FVNS.5 FVS$^h$ SIMPLE-Type PPNS Implementations

## Recall PPNS analytical theory ingredients

$$\mathrm{D}M^h:\ \mathbf{u}^{n+1} - \mathbf{u}^h = -\nabla\phi$$

$$\text{divergence}:\ \mathcal{L}(\phi) = -\nabla^2\phi + \nabla\cdot\mathbf{u}^h = 0$$

$$\text{BCs}:\ \mathcal{L}(\phi) = \hat{\mathbf{n}}\cdot\nabla\phi = 0,\ \ \phi_{\text{out}} = 0$$

$$\mathrm{D}\mathbf{P}^h\ \text{pressure}:\ \ \mathrm{SUM}\Phi_{n+1}^{p+1} = \Phi_n + (\theta\Delta t)^{-1}\sum_{\alpha=0}^{p}\phi_{n+1}^{\alpha+1}$$

$$\text{genuine pressure}:\ \mathcal{L}(p) = -\mathrm{Eu}\nabla^2 p - s(\nabla\mathbf{uu},\Theta) = 0$$

$$\ell(p) = \hat{\mathbf{n}}\cdot\nabla p + f(\mathrm{Re},\nabla^2\mathbf{u}\cdot\hat{\mathbf{n}})$$

$$\text{accuracy assessment}:\ \left|\phi^{p+1}\right|_E << \varepsilon,\ \ \left|\mathrm{SUM}\Phi\right|_E < \varepsilon\ \ \textit{and}\ \ \left|p^h\right|_E > \varepsilon$$

## SIMPLE implementations readily interpreted within the theory

genune pressure *never* determined from $\mathcal{L}(p)$

$\Rightarrow$ SUM$\Phi$ is interpreted as pressure

correction pressure is identical to $\phi^p$

velocity correction utilizes D$M^h$ definition

$\Rightarrow \mathbf{u}' \cong \nabla^h\phi^h$

# FVNS.6 $FVS^h$ SIMPLE-Type PPNS Implementations, Con'd

## SIMPLE (Spalding, et al, 1968)

1. guess $p_{n+1}^0$
2. solve d) $\Rightarrow \mathbf{u}^h$
3. solve a) $\Rightarrow p'$
4. relaxing $p'$, update b) $\Rightarrow p_{n+1}^1$
5. evaluate c) $\Rightarrow \mathbf{u}'$
6. repeat 1 – 5 until d) $|\mathrm{RES}|_{max} < \delta$
7. advance index $n$

## PPNS theory ingredients

a) $\mathcal{L}(\phi) = -\nabla^2\phi + \nabla\cdot\mathbf{u}^h = 0$

b) $\mathrm{SUM}\,\Phi_{n+1}^p = \Phi_n + \Delta t^{-1}\sum_{\alpha}^{p}\phi^{\alpha}$

c) $-\nabla\phi^{\mathrm{p}} = \mathbf{u}^{n+1} - \mathbf{u}^h$

d) $\mathrm{D}\mathbf{P}^h \Rightarrow \mathrm{FVS}^h(\mathcal{L}(\mathbf{u}^h)) = \{\mathrm{RES}\}_{n+1}$

## SIMPLER (Patankar, 1981)

1. guess $\mathbf{u}_{n+1}^0$
2. for $p_{n+1}^{'} \equiv 0$, solve d) $\Rightarrow \mathbf{u}_{n+1}^1$
3. solve a) $\Rightarrow p'$
4. update b) $\Rightarrow p_{n+1}^1$
5. solve d) $\Rightarrow u_{n+1}^2$
6. solve a) $\Rightarrow p'$
7. evaluate c) $\Rightarrow \mathbf{u}''$
8. repeat 1 – 7 until d) $|\mathrm{RES}|_{\max} < \delta$
9. advance index $n$

## Comments on performance

0. SIMPLE $\Rightarrow$ Semi-Implicit Method for Pressure-Linked Equations
1. all employ linear stationary iteration e.g., Gauss-Seidel
2. SIMPLE slow to converge for $p$ hence SIMPLER
3. SIMPLEC (Rathby, et. al., 1984) SIMPLE iteration with refined velocity update c) $\Rightarrow \mathbf{u}''$

# FVNS.7 FVS$^h$ PISO PPNS Implementation, Genuine Pressure

## PISO $\Rightarrow$ Pressure-Implicit with Splitting of Operators (Issa,1984)

### A non-iterative predictor-corrector

1. using $p_n$ solve d) $\Rightarrow \mathbf{u}^1_{n+1}$
2. solve a) $\Rightarrow p'$
3. evaluate b) $\Rightarrow p^1_{n+1}$
4. evaluate c) $\Rightarrow \mathbf{u}'$
5. solve e) $\Rightarrow p^2_{n+1}$
6. solve d) $\Rightarrow \mathbf{u}^2_{n+1}$
7. advance index $n$

### PPNS theory ingredients

a) $\mathcal{L}(\phi) = -\nabla^2\phi + \nabla\cdot\mathbf{u}^h = 0$

b) $\mathrm{SUM}\Phi^p_{n+1} = \Phi_n + \Delta t^{-1}\sum_{\alpha}^{p}\phi^\alpha$

c) $-\nabla\phi^p = \mathbf{u}^{n+1} - \mathbf{u}^h$

d) $\mathrm{D}\mathbf{P}^h \Rightarrow \mathrm{FVS}^h(\mathcal{L}(\mathbf{u}^h)) = \{\mathrm{RES}\}_{n+1}$

e) $\nabla\cdot\mathrm{D}\mathbf{P}^h \Rightarrow \mathrm{FVS}^h(\mathcal{L}(p^h, \mathrm{Re}))$

### Comments on PISO

1. $\nabla\cdot\mathrm{D}\mathbf{P}^h \Rightarrow \mathcal{L}(p)$ with BCs $\hat{\mathbf{n}}\cdot\nabla p = 0$

   requires 3$^{\text{rd}}$ derivatives of $\mathbf{u}^h$ in $\{\mathrm{RES}\}$!

2. accepts step 6 solution as converged

   $\Rightarrow \left|\phi^h\right|_E < ?$

# FVNS.8 FVS$^h$ for PPNS System, Stability, Numerical Diffusion

## FVS$^h$ commercial CFD codes based on PPNS employ numerical diffusion

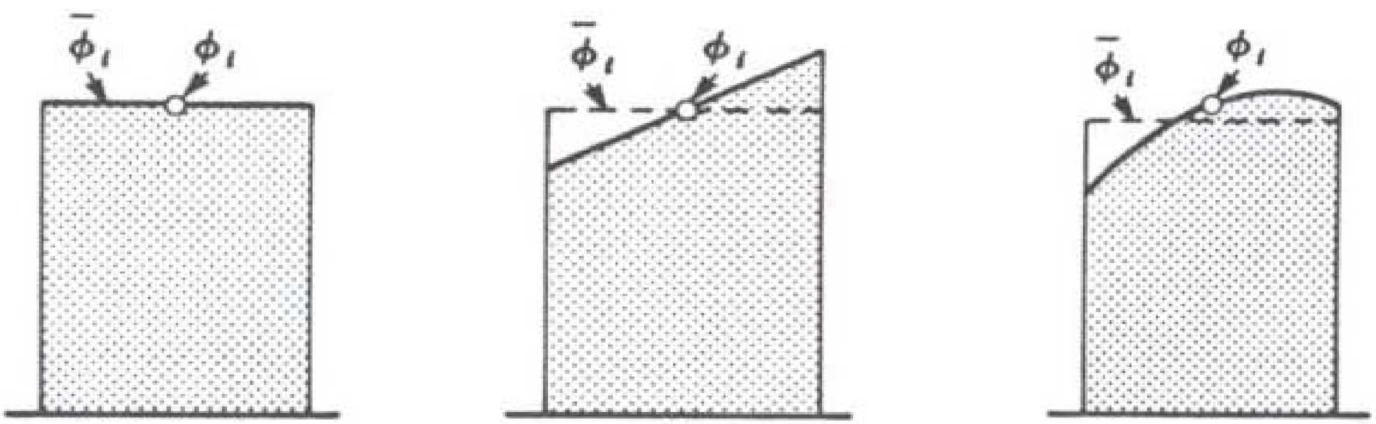
options for altering symmetric evaluation of $\int_{\partial\Omega_v} (u_j u_i)\hat{\mathbf{n}}_j d\sigma$

- direct upwinding $\Rightarrow$ TS accuracy is $O(h)$
- hybrid upwind $\Rightarrow O(h \leftrightarrow h^2)$
- quadratic upwind $\Rightarrow O(h^3)$

these typically employ $\theta = 1$ equivalent time – differencing

- time – accurate requires $\theta = 0.5$

## Formulation options based on $\Omega_v$ interpolation (B. Leonard, *Adv. N.H.T.*,1997)




constant linear quadratic

# FVNS.9 Convective Flux Differencing for FVS$^h$ PPNS

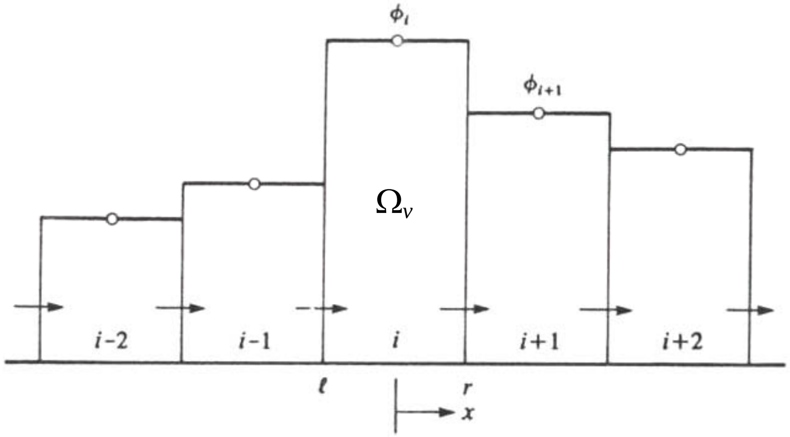
## First-order upwind:

$$\text{on } \Omega_v: \quad \phi = \text{constant}$$

$$\oint_{\partial\Omega_v} \mathbf{u}\phi \cdot \hat{\mathbf{n}} d\sigma = (u_r\phi_r - u_\ell\phi_\ell)$$

$$\Rightarrow \phi_r(\text{1UP}) = \phi_i,\ u_r > 0$$

$$= \phi_{i+1}, u_r < 0$$

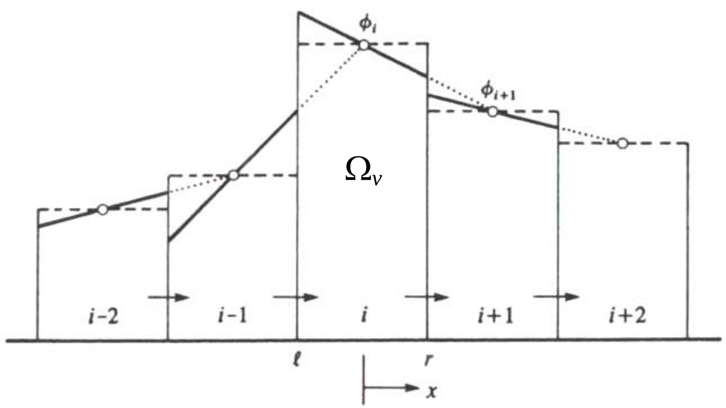



## Second-order central:

$$\text{on } \Omega_v: \quad \phi(x) = \phi_i + (\phi_{i+1} - \phi_i)x/h,\ u_r > 0$$

$$\int_{\partial\Omega_r} \mathbf{u}\phi \cdot \hat{\mathbf{i}} = \phi_r(\text{2CE}) = \phi(x = h/2 - \varepsilon)$$

$$= \frac{1}{2}(\phi_{i+1} - \phi_i),\ u_r > 0$$



# FVNS.10 Convective Flux Differencing Methods for FVS$^h$ PPNS

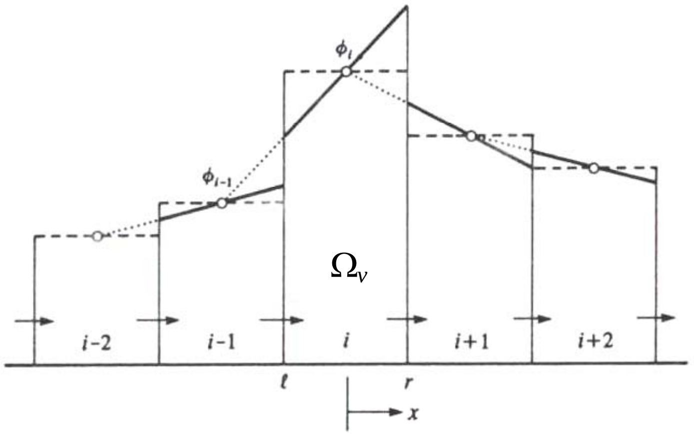
## Second-order upwind:

$$\text{on } \Omega_v: \quad \phi(x) = \phi_i + (\phi_i - \phi_{i-1})x/h, \;\; u_r > 0$$

$$\int_{\partial\Omega_r} \mathbf{u}\phi \cdot \hat{\mathbf{i}} = \phi_r(2\text{UP}) = \phi(x = h/2 - \varepsilon)$$

$$= \frac{3}{2}\phi_i - \phi_{i-1}, \quad u_r > 0$$

$$= \frac{3}{2}\phi_{i+1} - \phi_{i+2}, \;\; u_r < 0$$



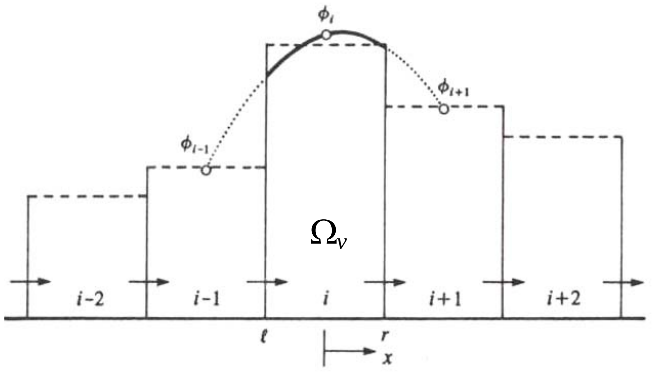

## Third-order upwind (QUICK):

$$\text{on } \Omega_v: \quad \phi(x) = \phi_i + \frac{1}{2}(\phi_{i+1} - \phi_{i-1})x/h + \frac{1}{2}(\phi_{i+1} - 2\phi_i + \phi_{i-1})\,(x/h)^2$$

$$\int_{\partial\Omega_r} \mathbf{u}\phi \cdot \hat{\mathbf{i}} \; = \phi_r(3\text{QK}) = \phi(x = h/2 - \varepsilon)$$

$$= (6\phi_i + 3\phi_{i+1} - \phi_{i-1})/8, \;\; u_r > 0$$

$$= (3\phi_i + 6\phi_{i+1} - \phi_{i+2})/8, \;\; u_r < 0$$



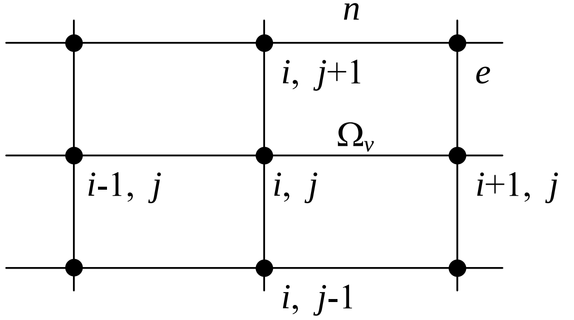
# FVNS.11 QUICK Differencing for Convection, *n* = 1, 2

## QUICK upwind differencing in curvature factor form, *n* = 1

$$\phi_r(3QU) = \frac{1}{2}(\phi_{i+1} + \phi_i) - \frac{1}{8}(\phi_{i+1} - 2\phi_i + \phi_{i-1}),\ u_r > 0$$
$$= \frac{1}{2}(\phi_{i+1} + \phi_i) - \frac{1}{8}(\phi_{i+2} - 2\phi_{i+1} + \phi_i),\ u_r < 0$$
$$= \text{average}\,\phi\Big|_{\partial\Omega_r} + c\delta^2\phi\Big|_{\Omega_v}$$

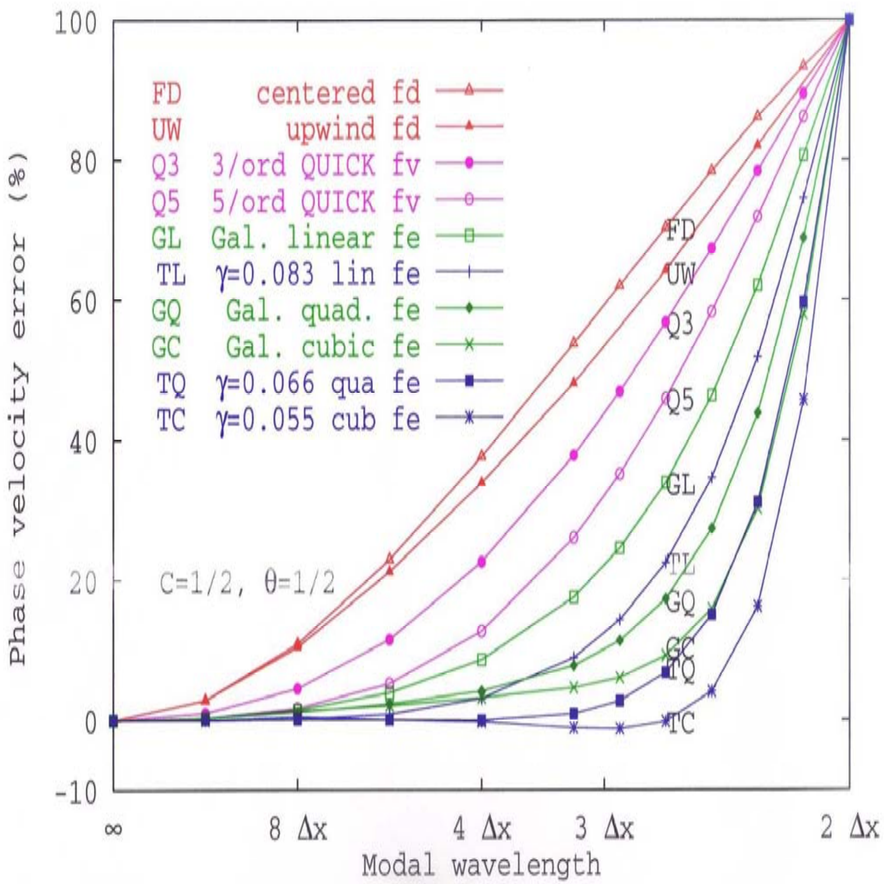
## Multi-dimensional QUICK differencing for convection

$$\phi_{east}(3QU) = \frac{1}{2}(\phi_{i+1,j} + \phi_{i,j})$$
$$-\frac{1}{8}(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})$$
$$+\frac{1}{24}(\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}),\ u_{east} > 0$$
$$= \text{average}\,\phi\Big|_{\partial\Omega_v} + c_1\delta_x^2\phi\Big|_{\Omega_v} - c_2\delta_y^2\phi\Big|_{\Omega_v}$$

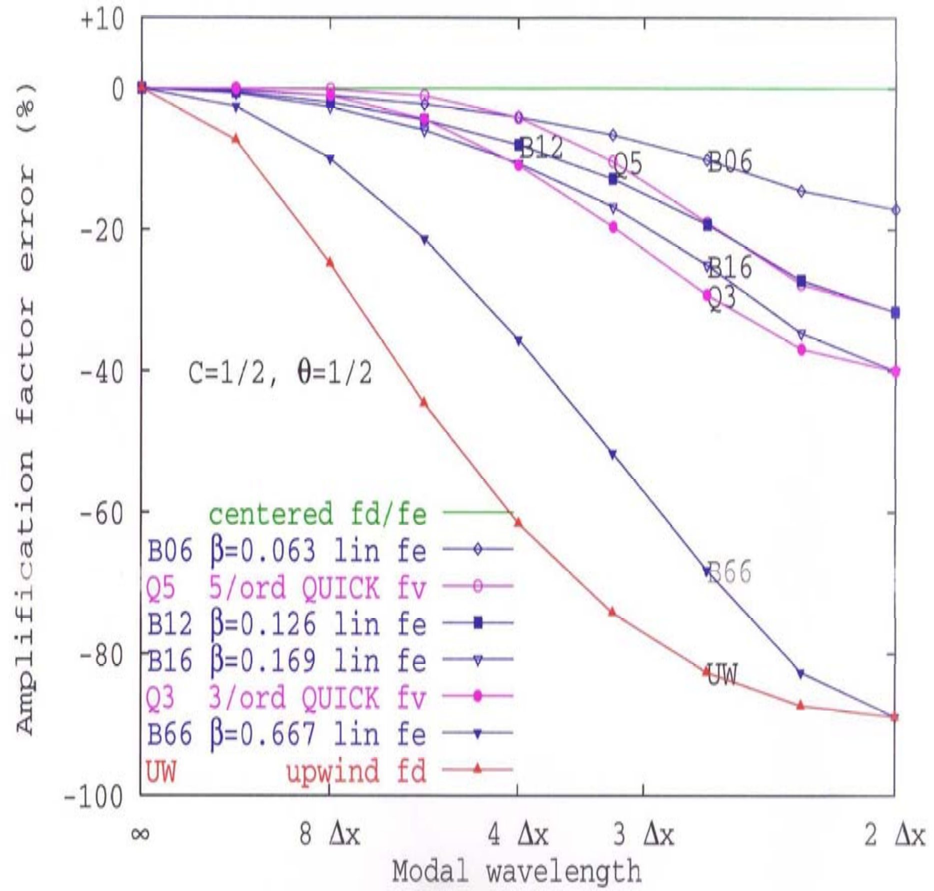


# FVNS.12 GWS$^h$-TWS$^h$-FVS$^h$ Spectral Resolution Comparisons

## Phase velocity error




## Artificial diffusion error



# FVNS.13 Summary FVS$^h$ - Implemental PPNS Algorithms

## PPNS analytical theory ingredients are independent of implementation

$$\text{D}M^h: \ \mathbf{u}^{n+1} - \mathbf{u}^h = -\nabla\phi$$

$$\text{divergence}: \ \mathcal{L}(\phi) = -\nabla^2\phi + \nabla\cdot\mathbf{u}^h = 0$$

$$\text{BCs}: \ \mathcal{L}(\phi) = \hat{\mathbf{n}}\cdot\nabla\phi = 0, \ \phi_{\text{out}} = 0$$

$$\text{D}\mathbf{P}^h \ \text{pressure}: \ \text{SUM}\,\Phi^{p+1}_{n+1} = \Phi_n + (\theta\Delta t)^{-1}\sum_{\alpha=0}^{p}\phi^{\alpha+1}_{n+1}$$

$$\text{genuine pressure}: \ \mathcal{L}(p) = -\text{Eu}\nabla^2 p - s(\nabla\mathbf{uu},\Theta) = 0$$

$$\ell(p) = \hat{\mathbf{n}}\cdot\nabla p + f(\text{Re}, \nabla^2\mathbf{u}\cdot\hat{\mathbf{n}})$$

$$\text{accuracy assessment}: \ \left|\phi^{p+1}\right|_E << \varepsilon, \ \left|\text{SUM}\,\Phi\right|_E < \varepsilon \ \ \textit{and} \ \ \left|p^h\right|_E > \varepsilon$$

## FVS$^h$ implementations constitute substantial simplifications

genune pressure *never* determined from $\mathcal{L}(p) + \ell(p)$

employstationary linear iteration predictor−corrector procedures

$\nabla^h\phi^h \Rightarrow$ velocity correction

$\phi^h \Rightarrow$ pressure correction

SUMΦ ⇒ interpreted as genuine pressure

convergence estimate based on $\max\left|\{\text{RES}\}^p\right|$

$\left|\phi^h\right|_E$ is never evaluated

artificial diffusion permeates implementations