WAYS TO REPRESENT TERRAIN

Terrain following layers

Cut cells

Unstructured

Source: Smolarkiewicz & Szmelter
http://ral.ucar.edu/hap/events/orographic-precip/images/2wed/am/day2-Wed_am_3-Orogunmesh2.ppt
SLANTED CELLS

Before

$c_1$  $c_2$

$c_3$  $c_4$

$c_5$  $c_6$
SLANTED CELLS

Before

\[ c_1 \quad c_2 \]
\[ \quad \quad \quad \]
\[ c_3 \quad c_4 \]
\[ \quad \quad \quad \]
\[ c_5 \quad c_6 \]

After

\[ c_1 \quad c_2 \]
\[ \quad \quad \quad \]
\[ c_3 \quad c_4 \]
CUT CELLS

SLANTED CELLS

Source: Shaw & Weller 2016, MWR, dx.doi.org/10.1175/MWR-D-15-0226.1
SLANTED CELLS

• Easy to construct
• Avoid arbitrarily small cells
• Generalise to 3D with arbitrary horizontal meshes
CUBICFIT: AN ADVECTION SCHEME FOR STEEP SLOPES
CUBICFIT: AN ADVECTION SCHEME FOR STEEP SLOPES

- Finite volume
- Eulerian
- Multidimensional cubic approximation
- Method-of-lines with Runge-Kutta timestepping

- No flux correction
- Not monotonic
FINITE VOLUME DISCRETISATION

\[ \frac{\partial \phi}{\partial t} + \nabla \cdot (u \phi) = 0 \]
FINITE VOLUME DISCRETISATION

\[
\frac{\partial \phi}{\partial t} + \nabla \cdot (u \phi) = 0
\]

\[
\frac{\partial \phi}{\partial t} + \frac{1}{V} \sum_f \phi_f u_f \cdot S_f = 0
\]
HOW TO ESTIMATE $\Phi_F$?

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (u\phi) = 0$$

$$\frac{\partial \phi}{\partial t} + \frac{1}{V} \sum_f \phi_F u_f \cdot S_f = 0$$
UPWIND-BIASED STENCIL
STENCIL-LOCAL COORDINATES
LEAST SQUARES FIT

\[ \phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 x y^2 \]
Φ_F IS CHEAP TO COMPUTE

Φ_\(F = a_1 = \begin{bmatrix} w_1 & \phi_1 \\ w_2 & \phi_2 \\ \vdots & \vdots \\ w_{12} & \phi_{12} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{12} \end{bmatrix}

\phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 x y^2
ESTIMATING $\Phi_F$ NEAR BOUNDARIES
\[ \phi = a_1 + a_2 x + a_3 y \]
\[ \phi = a_1 + a_2 x + a_3 y + a_4 x^2 \]
ESTIMATING $\Phi_F$ NEAR BOUNDARIES

$\phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2$
ESTIMATING $\Phi_F$ NEAR BOUNDARIES

\[ \phi = a_1 + a_2 x + a_3 y + a_4 x y + a_5 y^2 + a_6 x y^2 \]
ESTIMATING \( \Phi_F \) NEAR BOUNDARIES

\[
\phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2
\]
ESTIMATING $\Phi_F$ NEAR BOUNDARIES

$$\phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2$$
ESTIMATING $\Phi_F$ NEAR BOUNDARIES

$$\phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2$$
What is the most suitable polynomial for a given distribution of points?
What is the highest degree polynomial that ensures numerically stable advection?
\[ \frac{\partial \phi_j^{(n)}}{\partial t} = -u \frac{\phi_R - \phi_L}{\Delta x} \]
VON NEUMANN STABILITY

\[
\begin{align*}
\frac{\partial \phi_j^{(n)}}{\partial t} &= -u \frac{\phi_R - \phi_L}{\Delta x} \\
\phi_L &= w_u \phi_{j-1} + w_d \phi_j \\
\phi_R &= w_u \phi_j + w_d \phi_{j+1}
\end{align*}
\]
VON NEUMANN STABILITY

\[ \phi_L = w_u \phi_{j-1} + w_d \phi_j \]
\[ \phi_R = w_u \phi_j + w_d \phi_{j+1} \]

- Assume perfect timestepping
VON NEUMANN STABILITY

- Assume perfect timestepping
- Assume wave-like solution $\phi_j^{(n)} = A^n e^{ijk\Delta x}$

$\phi_L = w_u \phi_{j-1} + w_d \phi_j$

$\phi_R = w_u \phi_j + w_d \phi_{j+1}$
• Assume perfect timestepping
• Assume wave-like solution $\phi_j^{(n)} = A^n e^{ijk\Delta x}$
• Introduce constraints:
  • $|A| \leq 1$
VON NEUMANN STABILITY

\[ \phi_L = w_u \phi_{j-1} + w_d \phi_j \]
\[ \phi_R = w_u \phi_j + w_d \phi_{j+1} \]

- Assume perfect timestepping
- Assume wave-like solution \( \phi_j^{(n)} = A^n e^{ijk\Delta x} \)
- Introduce constraints:
  - \(|A| \leq 1\)
  - \(\arg(A) < 0\) for \(\text{Co} > 0\)
Assume perfect timestepping

Assume wave-like solution $\phi_j^{(n)} = A^n e^{ijk \Delta x}$

Introduce constraints:

- $|A| \leq 1$
- $\arg(A) < 0$ for $C_0 > 0$
- No more damping than first-order upwind ($w_u = 1$, $w_d = 0$)

\[
\phi_L = w_u \phi_{j-1} + w_d \phi_j \\
\phi_R = w_u \phi_j + w_d \phi_{j+1}
\]
2-point approximation

\[ \phi_L = w_u \phi_{j-1} + w_d \phi_j \]
\[ \phi_R = w_u \phi_j + w_d \phi_{j+1} \]
VON NEUMANN STABILITY

2-point approximation

\[ u \]
\[ \phi_{j-1} \quad \phi_j \quad \phi_{j+1} \]
\[ \phi_L = w_u \phi_{j-1} + w_d \phi_j \]
\[ \phi_R = w_u \phi_j + w_d \phi_{j+1} \]

3-point approximation

\[ \phi_{j-2} \quad \phi_{j-1} \quad \phi_j \quad \phi_{j+1} \]
\[ \phi_L = w_{uu} \phi_{j-2} + w_u \phi_{j-1} + w_d \phi_j \]
\[ \phi_R = w_{uu} \phi_{j-1} + w_u \phi_j + w_d \phi_{j+1} \]

4-point approximation

\[ \phi_{j-3} \quad \phi_{j-2} \quad \phi_{j-1} \quad \phi_j \quad \phi_{j+1} \]
\[ \phi_L = w_{uuu} \phi_{j-3} + w_{uu} \phi_{j-2} + w_u \phi_{j-1} + w_d \phi_j \]
\[ \phi_R = w_{uuu} \phi_{j-2} + w_{uu} \phi_{j-1} + w_u \phi_j + w_d \phi_{j+1} \]
VON NEUMANN STABILITY

\[ 0.5 \leq w_u \leq 1 \]
\[ 0 \leq w_d \leq 0.5 \]
\[ w_u - w_d \geq \max_{\rho \in P}(|w_{\rho}|) \]

\[ \Phi_F = \begin{bmatrix} w_u \\ w_d \\ w_3 \\ \vdots \\ w_{12} \end{bmatrix} \cdot \begin{bmatrix} \Phi_u \\ \Phi_d \\ \Phi_3 \\ \vdots \\ \Phi_{12} \end{bmatrix} \]
POLYNOMIAL FIT ALGORITHM

1. Generate candidate polynomials
2. Test each candidate against von Neumann stability criteria
3. Choose the best candidate that satisfies the criteria

\[
\begin{align*}
\phi &= a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2 \\
\phi &= a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y \\
\phi &= a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 xy^2 \\
\phi &= a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^2 y + a_8 xy^2 \\
\vdots \\
\phi &= a_1 + a_2 x + a_3 y \\
\phi &= a_1 + a_2 x + a_3 x^2 \\
\phi &= a_1 + a_2 y + a_3 y^2 \\
\phi &= a_1 + a_2 x \\
\phi &= a_1 + a_2 y
\end{align*}
\]

\[0.5 \leq w_u \leq 1\]

\[0 \leq w_d \leq 0.5\]

\[w_u - w_d \geq \max_{p \in P} (|w_p|)\]
ESTIMATING $\Phi_F$ NEAR BOUNDARIES

\[ \phi = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2 \]
ESTIMATING $\Phi_F$ NEAR BOUNDARIES

$$\phi = a_1 + a_2 x + a_3 y + a_4 x y + a_5 x^2$$
NUMERICAL EXPERIMENTS

1. Schär horizontal advection over orography
2. “Slug” advection over orography
NUMERICAL EXPERIMENTS

1. Schär horizontal advection over orography
2. “Slug” advection over orography

Compare
- cubicFit
- linearUpwind
SCHÄR HORIZONTAL ADVECTION

Horizontal wind profile, surface terrain profile and initial tracer
Adapted from Schär et al. 2002, MWR
BASIC TERRAIN FOLLOWING

linearUpwind

cubicFit
CUT CELLS

linearUpwind

cubicFit
“SLUG” ADVECTION TEST
BASIC TERRAIN FOLLOWING

linearUpwind

cubicFit

T_diff

-5.000e-01  -0.25   0   0.25   5.000e-01
CUT CELLS

linearUpwind

cubicFit
SLANTED CELLS

linearUpwind

cubicFit
“SLUG” ADVECTION TIMESTEPS

![Chart showing timestep values for different scenarios: Basic Terrain Following, Slanted cells, Cut cells. The Basic Terrain Following scenario has the highest timestep value.]
MAXIMUM TIME STEPS

![Graph showing relationship between Δt and Δx for different cell types]

- Basic terrain following
- Slanted cells
- Cut cells
CONCLUSIONS

• cubicFit is cheap to compute (dot product of two vectors)
• cubicFit is suitable for many types of mesh
• Maximum timesteps on slanted cells scale predictably with mesh spacing

FUTURE WORK

Contact me: @hertzsprrrungr or js102@zepler.net
Slides and additional resources: goo.gl/jLR7vW