Galaxy formation – Burns, Greif, Bromm, and Klessen –- Texas Advanced Computing Center-- www.tacc.utexas.edu/research/gallery

Atms 502, CSE 566

Numerical Fluid Dynamics

Thu., Mar. 7, 2019

ATMS 502
CSE 566

Thursday,
7 March 2019

Class #16

Plan for Today

- 1) Skamarock & Klemp
  - Global/local refinement
  - A full AMR method, explained
- 1) Space differencing
  - Differential-difference approach
  - Phase speed & group velocity
  - Impacts of higher order, added diffusion
- 2) Programming input/output
  - C and Fortran considerations
- 3) Program 4: *continued*
  - Nest placement / movement

Review from last class

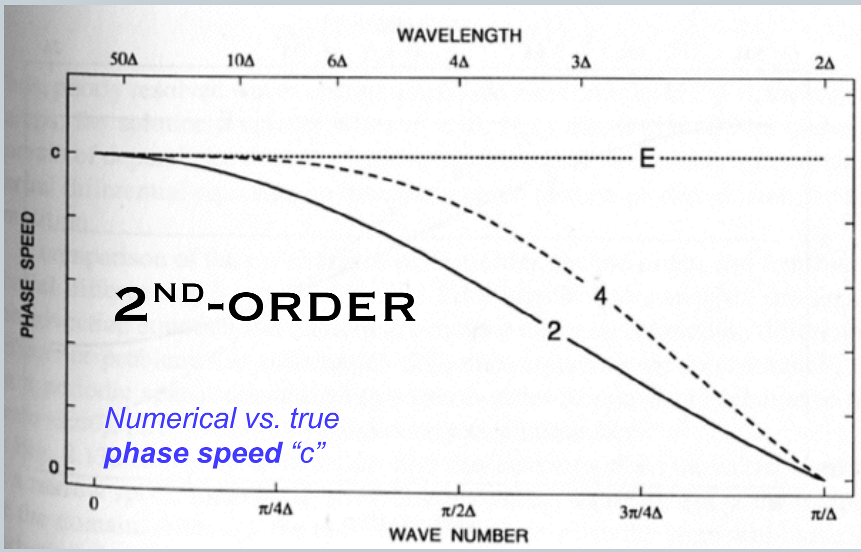
Review: Space Differencing

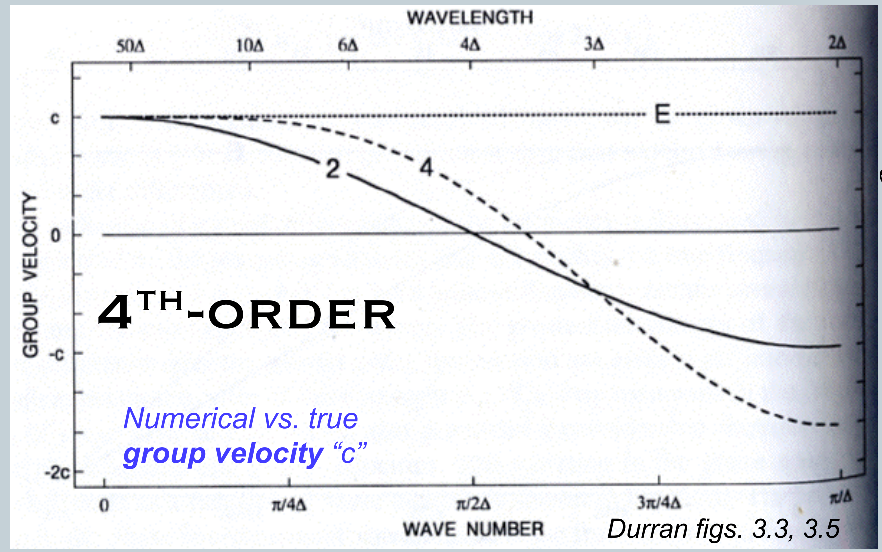
- Differential-difference eqn:
  - 2nd-order space: $\frac{du_j}{dt} + c\left(\frac{u_{j+1} - u_{j-1}}{2\Delta x}\right) = 0$ Substitute: $u_j(t) = e^{i(kj\Delta x - \omega_{2c} t)}$
  - Solve for *frequency* ω, which *may be complex;* here $\omega = \frac{c \sin\beta}{\Delta x}$

- Once we have the frequency:
  - Phase speed: $c_{2c} = \frac{\omega}{k} = \frac{c\sin\beta}{k\Delta x}$; for small kΔx, $c_{2c} \approx c\left(1 - \frac{\beta^2}{6}\right)$
    - perfect (=c) for infinite waves where kΔx=0
    - zero for 2Δx waves
  - Group velocity: $c_{g2c} = \frac{\partial\omega}{\partial k} = \frac{\partial}{\partial k}\left(\frac{c\sin\beta}{\Delta x}\right) = c(\cos\beta)$
    - perfect for infinite waves, = -c for 2Δx waves.

Review: 2nd vs. 4th order space diff.

- *Summary: 2nd-order centered space differencing*



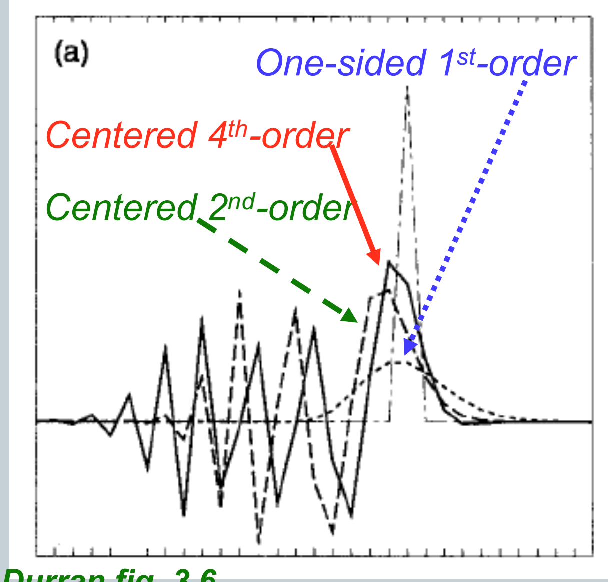


higher is better !

- Phase speed (left), group velocity (right)
- 4th better at "intermediate" wavelengths

Review: Accuracy and Order

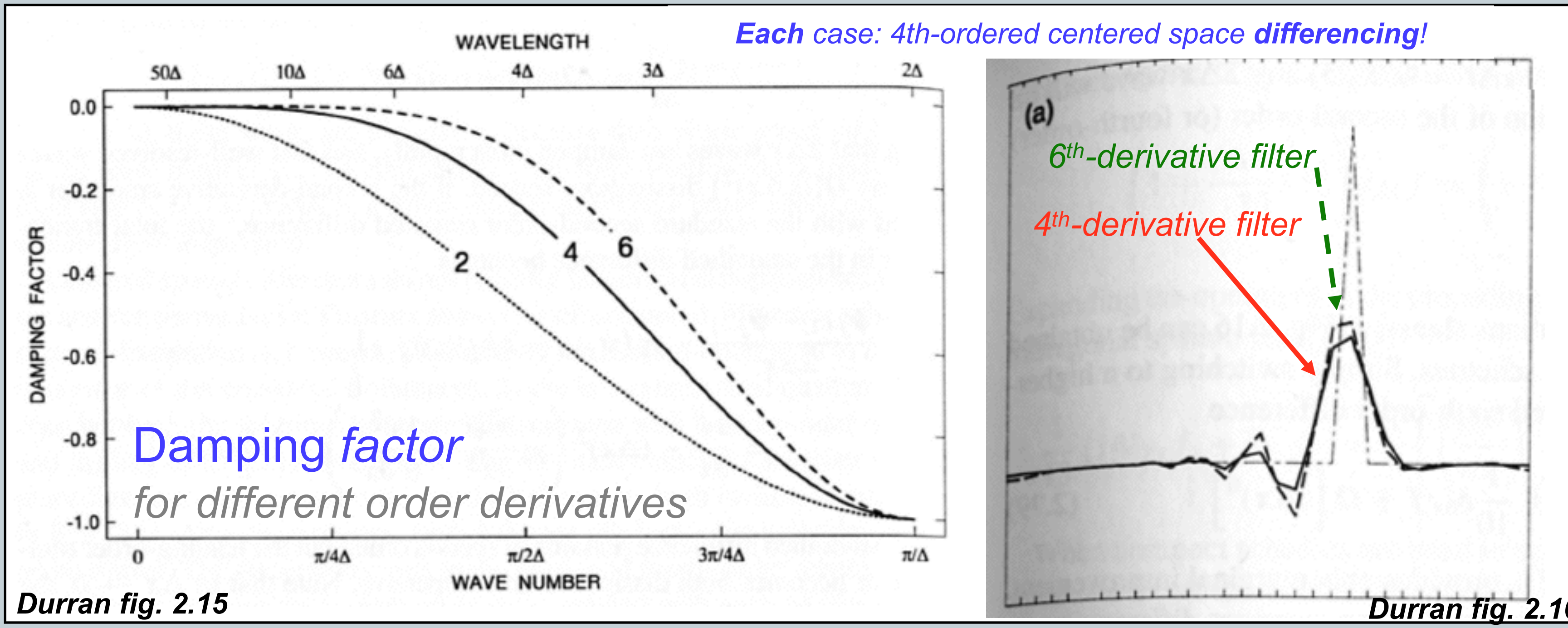
- "**Centered** [even-ordered] **schemes** preserve the amplitude of each individual Fourier component, [but] the various *components propagate at different speeds [and the result is ugly]*
- "Switching to a higher-order scheme does **not** improve the performance of finite-difference methods *when they are used to model poorly resolved features*



Durran fig. 3.6

Review: Explicit Artificial Dissipation

- *Added explicit damping can be beneficial*



Durran fig. 2.15

Durran fig. 2.16

- Higher order *derivatives [in damping]* damp intermediate wavelengths *less*.
- Added damping can reduce the dispersion errors.

Skamarock & Klemp

ADAPTIVE MESH REFINEMENT PAPER

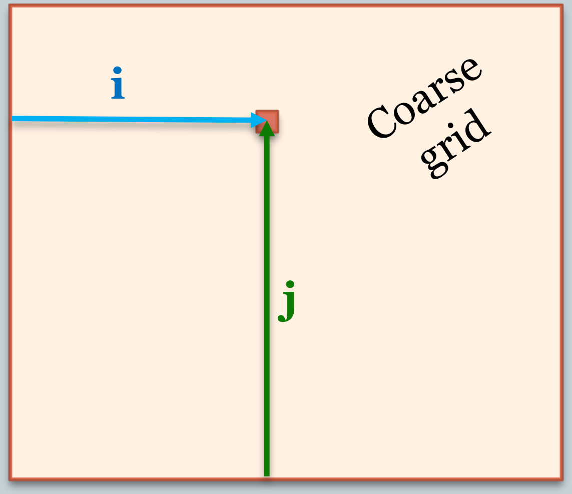
SEE NOTES FROM LAST CLASS!

Program 4

NEST PLACEMENT ALGORITHM

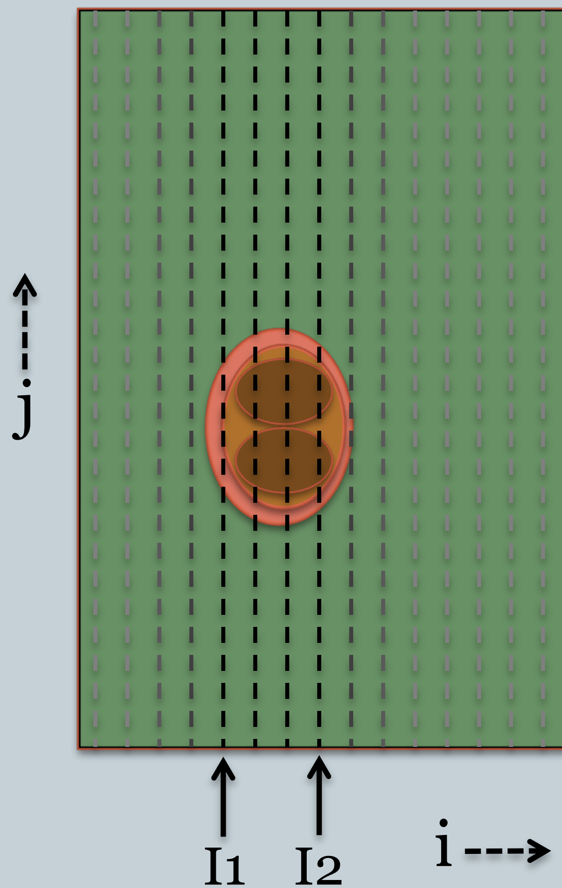
Nest location

- About that nest location -
  - This is what my code sequence looks like, in the routine in which I compute the truncation error.
    - loop: all **i** = 3:nx-2
      - loop: all **j** = 3:ny-2
        - xtrunc = (expression in X direction: i-2, i-1, i, i+1, i+2)
        - ytrunc = (expression in Y direction: j-2, j-1, j, j+1, j+2)
        - trunc_error(i,j) = max(abs(xtrunc) , abs(ytrunc))
      - end loop **j**
    - end loop **i**
  - Now I work with trunc_error() array - get max value, find truncation error "edges", average to find nest **center**.



Determining the nest location (2)

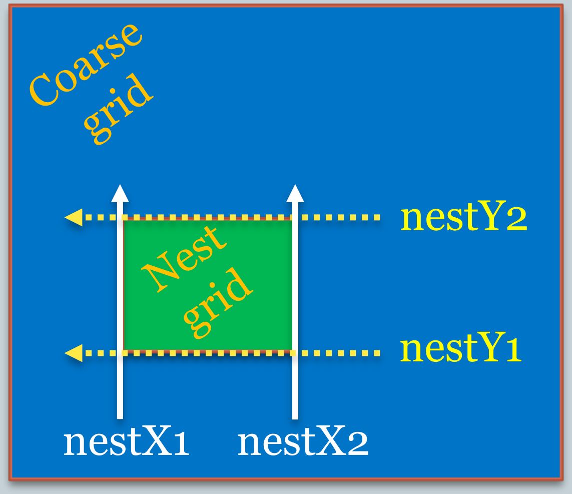
Cartoon of T.E. on coarse grid.



- Example of finding left, right edges of T.E. region
  - For each *i* column, left to right … check TE(i,j) for all j rows – determine max value
  - Find first and last column (i) for which max ≥ threshold.
- Do same for top/bottom.
- Average I1,I2,J1,J2 for nest *center*. Nest *edges* depend on nest *size*.

Program 4 questions (1)

- Handling the nest
  - four variables define *edges* of the nest
    - these variables are integers holding the *coarse grid coordinate* of the nest (discuss)
  - when *placing the nest* for the first time, *setting boundary values* for the nest, *doing feedback …*
    - then these four are the variables *dointerp* uses.
    - the other variables *nestX1old, nestX2old, nestY1old…* are ignored.
  - when *moving* the nest:
    - nestX1, nestX2, nestY1, nestY2 = location *where nest is to be moved*
    - nestX1old, nestX2old, etc … contain *old (current) nest location*



Program 4 Steps, part 1

1. You need rotational flow I.C. code (from program 2)
2. Copy my program 4 files on Stampede
3. Try interpolation code (F90 or C)
4. "Run" code with a passive "nest"

Simply use cone center for nest location; then interpolate coarse > nest

5. Develop truncation error code
   (find x, y truncation error, store 2d array of max(abs(xerror,yerror))…
6. Determine nest location
   get x,y error bounds; average = nest center; determine X1,X2,Y1,Y2

Program 4 Steps, part 2

7. Alter code to *place* nest at start (n=1); *move* nest afterwards (at time step n=5,10,…)
8. Boundary conditions for nest
   (Easy: Call dointerp with flag to just set nest BCs)
9. Alter integrate routine for nest
   (This is just a matter of what points are updated – 2:nx-1 on nest, etc)
10. Evolve the nest
    No feedback! Just get B.C's from coarse grid, integrate/update nest
11. Add feedback.
    By now you know your nest is 'ok' – do this step last.

Program 4 variables

- **You need to add:**
  - Extra storage (s1, s2 for nested grid)
  - Nest-specific parameters
    - Time step for nest
    - Grid spacing for nest
  - New variables – ***read these in***
    - Grid refinement ratio *(an integer. time & space!)*
    - Current nest location info *(first, last grid points in X,Y)*
    - Nest *update frequency (an integer; 1 = every step)*
    - Feedback *option*
      - *on or off.*

Program 4 Code restructuring

- **<u>Changes to problem 3 layout:</u>**
  - Just *one* advection method *(Lax-W)*
  - Advection routine called for both grids
    - Boundary condition will differ between grids
  - Additional time loop for nested grid
    - Boundary conditions *from coarse grid*
    - Many *nested grid steps* for each coarse grid step
  - Feedback code
  - Error calculations

Finite volume method; van Leer

<u>Overview:</u>

- van Leer published five papers between 1973-79
- J. Comp. Phys., vol. 23, 276-299 (1977):

  "Towards the ultimate conservative difference scheme: IV. A new approach to numerical convection"

<u>Handout:</u>

- Hourdin and Armengaud, 1999: Use of finite volume methods for atmospheric advection of trace species
  - flux forms
  - monotonicity
  - "approximating the sub-grid-scale distribution by a polynomial" *(HA99 p. 823)*

van Leer (1977)

- This all applies to the flux form of our equations e.g.

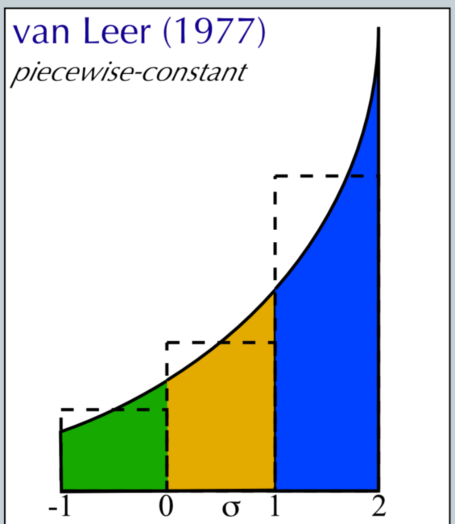
$$\frac{\partial q}{\partial t} = -u\frac{\partial q}{\partial x} - w\frac{\partial q}{\partial z} \quad \Rightarrow$$

$$\frac{\partial q}{\partial t} = -\frac{\partial}{\partial x}(uq) - \frac{\partial}{\partial z}(wq) + q\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)$$

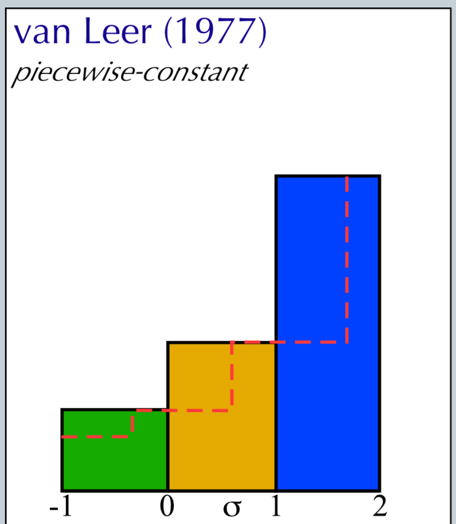
- Conceptually:
  - grid point values are averages within a grid zone
  - local functions describe field changes within the zone
    - piecewise *constant*, piecewise *linear*, piecewise *parabolic*
  - we integrate under local functions at time t
    that will be in grid zone of interest [0,Δx] *at t+Δt*.

Upstream – piecewise *constant*

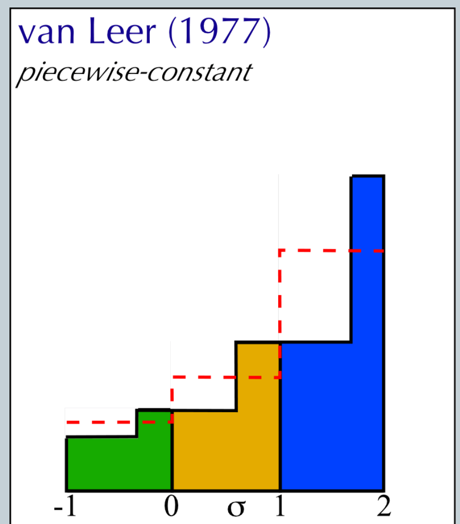
- Step 1 –
- identify grid zone averages…
- X coordinate is Courant number σ
- Grid box runs from [0-to-1]•Δx



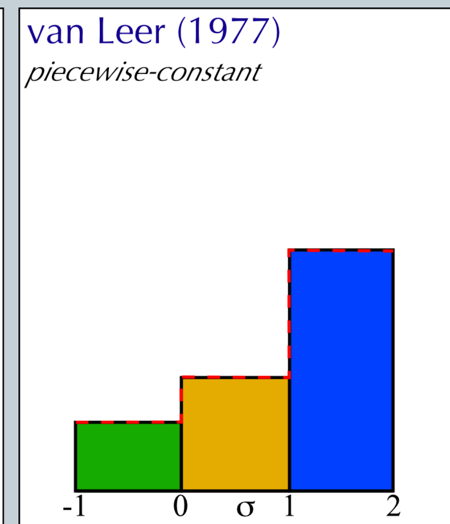
- Step 2 –
- look at distribution **before** and after advection takes place



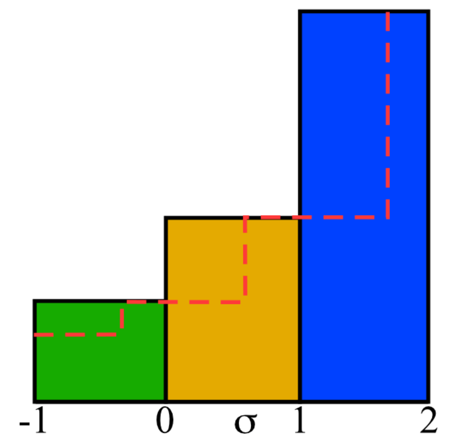
- Step 3 -
- Compute new grid zone averages.



- Step 4 -
- The averages *are* the function here (for piecewise constant)
- These are the initial values for the next time step.



van Leer (1977)



- New value from integrating under piecewise constant function at time *t* that *will be in the grid zone [0,Δx] at t+Δt.*

$$q^{n+1} \equiv \overline{q}^{1/2} = \int_0^{1-\sigma} q_{1/2}\,dx + \int_{-\sigma}^{0} q_{-1/2}\,dx \qquad \sigma = \frac{u\Delta t}{\Delta x}$$

Grid-point value f(j) represents the average of the function over the grid cell (see Durran, § 1.3.1, p. 27)

- Piecewise *constant* in each zone, so:

$$\begin{aligned}\overline{q}^{1/2} &= \overline{q}_{1/2}(1-\sigma) + \overline{q}_{-1/2}\sigma \\ &= \overline{q}_{1/2} - \sigma\left(\overline{q}_{1/2} - \overline{q}_{-1/2}\right)\end{aligned} \quad \longleftrightarrow \quad \overline{q}^{1/2} = \overline{q}_{1/2} - \left[u\overline{q}_{1/2} - u\overline{q}_{-1/2}\right]\frac{\Delta t}{\Delta x}$$

van Leer (1977)

- van Leer notation ...

$$\bar{q}^{1/2} = \bar{q}_{1/2} - \sigma\left(\bar{q}_{1/2} - \bar{q}_{-1/2}\right)$$
$$= \bar{q}_{1/2} - \left[\sigma\bar{q}_{1/2} - \sigma\bar{q}_{-1/2}\right]$$
$$= \bar{q}_{1/2} - \left[Flux_{1/2} - Flux_{-1/2}\right]$$

- Fluxes ...

$$Flux(i) \equiv Flux_{-1/2}$$
$$= \sigma q_{-1/2} = \sigma q(i-1)$$

  - note Δt is already included in (is part of) the fluxes.

- Coding:

$$q^{n+1} = q^n - \left(Flux_{1/2} - Flux_{-1/2}\right) + q_i^n \Delta t \delta_x u$$
$$= q^n - \left[(\sigma q_i) - (\sigma q_{i-1})\right] + q_i^n \frac{\Delta t}{\Delta x}\left(u_{i+1} - u_i\right)$$
$$= q^n - \sigma\left(q_i - q_{i-1}\right) \quad if \ \mathrm{u} = \mathrm{constant}$$

Greater accuracy: *Piecewise linear*

- Our general update formula:

$$q^{n+1} \equiv \overline{q}^{1/2} = \int_0^{1-\sigma} q_{1/2}\,dx + \int_{-\sigma}^{0} q_{-1/2}\,dx \quad \sigma = \frac{u\Delta t}{\Delta x}$$

  - Piecewise *constant*

$$q(x,t_0) = \overline{q}_{1/2}$$

  - Piecewise *linear*

$$q(x,t_0) = \overline{q}_{1/2} + \overline{\Delta}_{1/2} q\left(x - \frac{1}{2}\right)$$

  - Includes:
    - Zone average $\overline{q}_{1/2} = \int_0^1 q(x,t^0)\,dx$
    - Average slope $\overline{\Delta}_{1/2} q \equiv \left(\overline{\partial q / \partial x}\right)_{1/2}$

Constant – why?

Piecewise linear

- Van Leer piecewise linear: *Scheme I*

$$\overline{q}^{1/2} = \overline{q}_{1/2} - \sigma\left(\overline{q}_{1/2} - \overline{q}_{-1/2}\right) - \frac{\sigma}{2}\left(1-\sigma\right)\left(\overline{\Delta}_{1/2}q - \overline{\Delta}_{-1/2}q\right)$$

Piecewise linear & beyond

- Van Leer piecewise linear: *Scheme I*

$$\overline{q}^{1/2} = \overline{q}_{1/2} - \sigma\left(\overline{q}_{1/2} - \overline{q}_{-1/2}\right) - \frac{\sigma}{2}(1-\sigma)\left(\overline{\Delta}_{1/2}q - \overline{\Delta}_{-1/2}q\right)$$

LIKE PIECEWISE-CONSTANT: FLUXES FROM ZONE AVERAGES.

SLOPES IN ZONES.
- CAN BE EVALUATED MANY WAYS.
SCHEME 1: CENTERED DIFFERENCES

$$\overline{q}_{1/2} = \int_0^1 q(x,t^0)\,dx$$

$$\overline{\Delta}_{1/2}q = \frac{1}{2}\left(\overline{q}_{3/2} - \overline{q}_{-1/2}\right)$$

- How could we improve our method?
  - 1. ________________________________
  - 2. ________________________________