Strengths and Weaknesses of Common Numerical Methods for Simulating Atmospheric Flows

Dale Durran
University of Washington

6 July 2011

Broad Categories

- Spatially discrete
  - Finite difference
  - Finite volume

Broad Categories

- Spatially discrete
  - Finite difference
  - Finite volume
- Series expansions
  - Orthogonal expansion functions
  - Finite elements

Broad Categories

- Spatially discrete
  - Finite difference
  - Finite volume
- Series expansions
  - Orthogonal expansion functions
  - Finite elements
- Hybrids
  - Spectral element and discontinuous Galerkin

Broad Categories

- Spatially discrete
  - Finite difference
  - Finite volume
- Series expansions
  - Orthogonal expansion functions
  - Finite elements
- Hybrids
  - Spectral element and discontinuous Galerkin
- Fluid dynamical viewpoint
  - Eulerian
  - Lagrangian
  - Semi-Lagrangian

Illustrative Equation

Transport in 2D nondivergent flow.

Advective form:

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} = 0.$$

Illustrative Equation

Transport in 2D nondivergent flow.

Advective form:

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} = 0.$$

Flux form, an equivalent expression obtained using continuity:

$$\frac{\partial \psi}{\partial t} + \frac{\partial u\psi}{\partial x} + \frac{\partial v\psi}{\partial y} = 0.$$

Illustrative Equation

Transport in 2D nondivergent flow.

Advective form:

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} = 0.$$

Flux form, an equivalent expression obtained using continuity:

$$\frac{\partial \psi}{\partial t} + \frac{\partial u\psi}{\partial x} + \frac{\partial v\psi}{\partial y} = 0.$$

- Flux form facilitates the construction of schemes with local (cell-wise) and global conservation.

Illustrative Equation

Transport in 2D nondivergent flow.

Advective form:

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} = 0.$$

Flux form, an equivalent expression obtained using continuity:

$$\frac{\partial \psi}{\partial t} + \frac{\partial u\psi}{\partial x} + \frac{\partial v\psi}{\partial y} = 0.$$

- Flux form facilitates the construction of schemes with local (cell-wise) and global conservation.
- Advective form helps preserve uniform ψ in non-trivial velocity fields

Finite difference notation

$$\delta_{nx} f(x) = \frac{f(x + n\Delta x/2) - f(x - n\Delta x/2)}{n\Delta x}$$

Finite difference notation

$$\delta_{nx} f(x) = \frac{f(x + n\Delta x/2) - f(x - n\Delta x/2)}{n\Delta x}$$

Example:

$$\delta_x f_j = \frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{\Delta x}$$

Finite difference notation

$$\delta_{nx} f(x) = \frac{f(x + n\Delta x/2) - f(x - n\Delta x/2)}{n\Delta x}$$

Example:

$$\delta_x f_j = \frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{\Delta x}$$

Local conservation:

$$\frac{d\phi_{i,j}}{dt} + \delta_x(u_{i,j}\phi_{i,j}) + \delta_y(v_{i,j}\phi_{i,j}) = 0$$

Finite-Difference Methods

- Unknowns are

Finite-Difference Methods

- Unknowns are values at *points* on a grid: $\phi_j^n \approx \psi(j\Delta x, n\Delta t)$

Finite-Difference Methods

- Unknowns are values at *points* on a grid: $\phi_j^n \approx \psi(j\Delta x, n\Delta t)$
- Can approximate flux or advective form

Finite-Difference Methods

- Unknowns are values at *points* on a grid: $\phi_j^n \approx \psi(j\Delta x, n\Delta t)$
- Can approximate flux or advective form
- Basic methods are quite simple:

$$\left(\frac{d\psi}{dx}\right)_j = \delta_{2x}\phi_j + O[(\Delta x)^2], \quad \left(\frac{d\psi}{dx}\right)_j = \frac{4}{3}\delta_{2x}\phi_j - \frac{1}{3}\delta_{4x}\phi_j + O[(\Delta x)^4]$$

Finite-Difference Methods

- Unknowns are values at *points* on a grid: $\phi_j^n \approx \psi(j\Delta x, n\Delta t)$
- Can approximate flux or advective form
- Basic methods are quite simple:

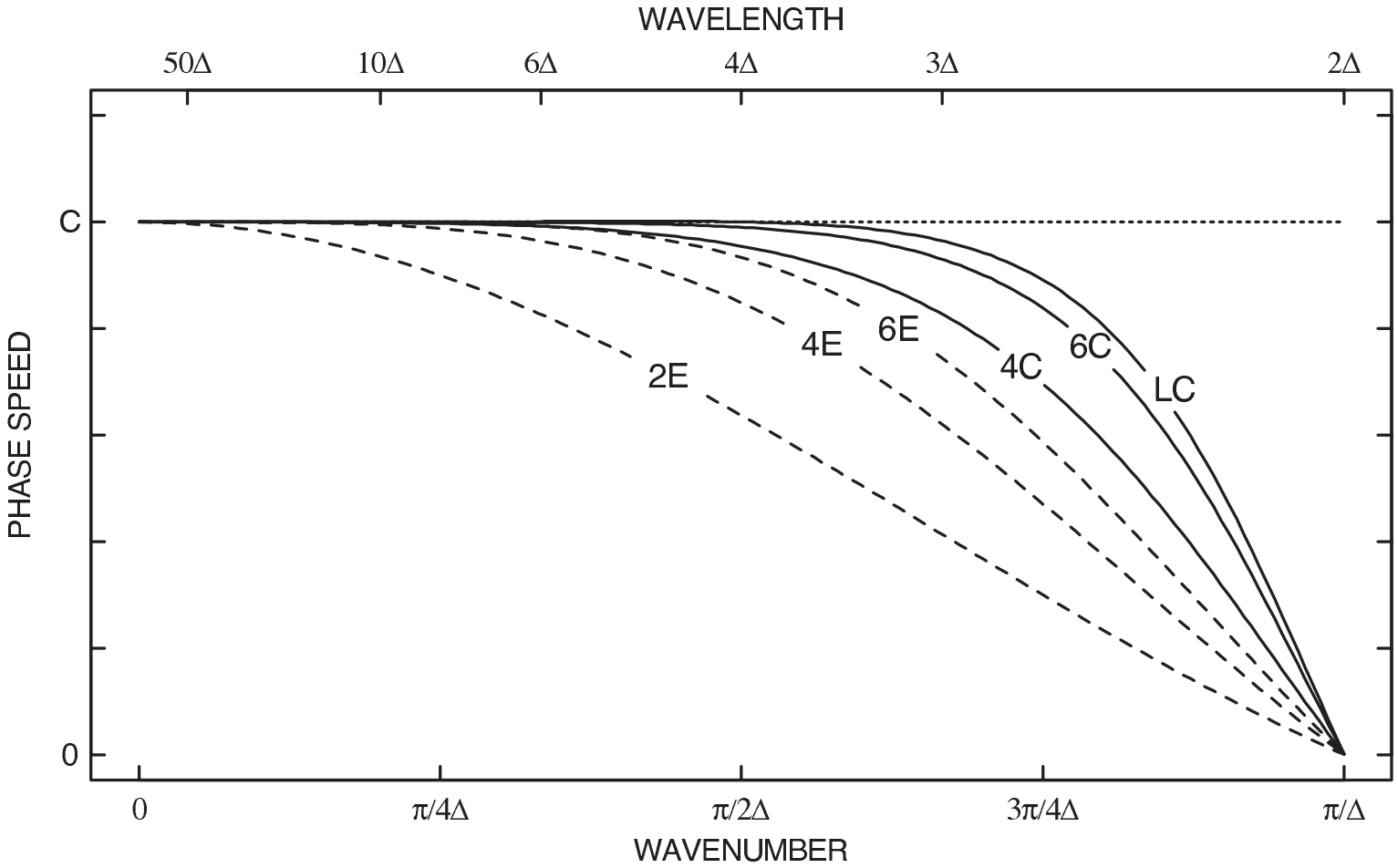
$$\left(\frac{d\psi}{dx}\right)_j = \delta_{2x}\phi_j + O[(\Delta x)^2], \quad \left(\frac{d\psi}{dx}\right)_j = \frac{4}{3}\delta_{2x}\phi_j - \frac{1}{3}\delta_{4x}\phi_j + O[(\Delta x)^4]$$

- More advanced approach: a 4th-order compact scheme

$$\frac{1}{24}\left[5\left(\frac{d\psi}{dx}\right)_{j+1} + 14\left(\frac{d\psi}{dx}\right)_j + 5\left(\frac{d\psi}{dx}\right)_{j-1}\right] = \frac{1}{12}\left(11\delta_{2x}\phi_j + \delta_{4x}\phi_j\right)$$

Phase Speed Error in 1D Advection

Semi-discrete approximation to $\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0.$



Utility of High Order

High order schemes converge more rapidly as the grid is refined when the solution is already almost correct.

Utility of High Order

High order schemes converge more rapidly as the grid is refined when the solution is already almost correct.

- They are essential for efficiency when really trying to converge.

Utility of High Order

High order schemes converge more rapidly as the grid is refined when the solution is already almost correct.

- They are essential for efficiency when really trying to converge.
- Convergence is never achieved in high Reynolds number atmospheric flow. (Why not?)

Utility of High Order

High order schemes converge more rapidly as the grid is refined when the solution is already almost correct.

- They are essential for efficiency when really trying to converge.
- Convergence is never achieved in high Reynolds number atmospheric flow. (Why not?)
  - *Exception:* all those complicated linear solutions for nontrivial basic states!

Utility of High Order

High order schemes converge more rapidly as the grid is refined when the solution is already almost correct.

- They are essential for efficiency when really trying to converge.
- Convergence is never achieved in high Reynolds number atmospheric flow. (Why not?)
  - *Exception:* all those complicated linear solutions for nontrivial basic states!
- In atmospheric applications, errors are generally dominated by the most poorly resolved scales.

Utility of High Order

High order schemes converge more rapidly as the grid is refined when the solution is already almost correct.

- They are essential for efficiency when really trying to converge.
- Convergence is never achieved in high Reynolds number atmospheric flow. (Why not?)
  - *Exception:* all those complicated linear solutions for nontrivial basic states!
- In atmospheric applications, errors are generally dominated by the most poorly resolved scales.
- High-order schemes *may* treat the marginally resolved scales better.

Utility of High Order II

Is it best to approximate all terms with differences having the same order of accuracy?

Utility of High Order II

Is it best to approximate all terms with differences having the same order of accuracy?

- Yes – if you are trying to achieve convergence.

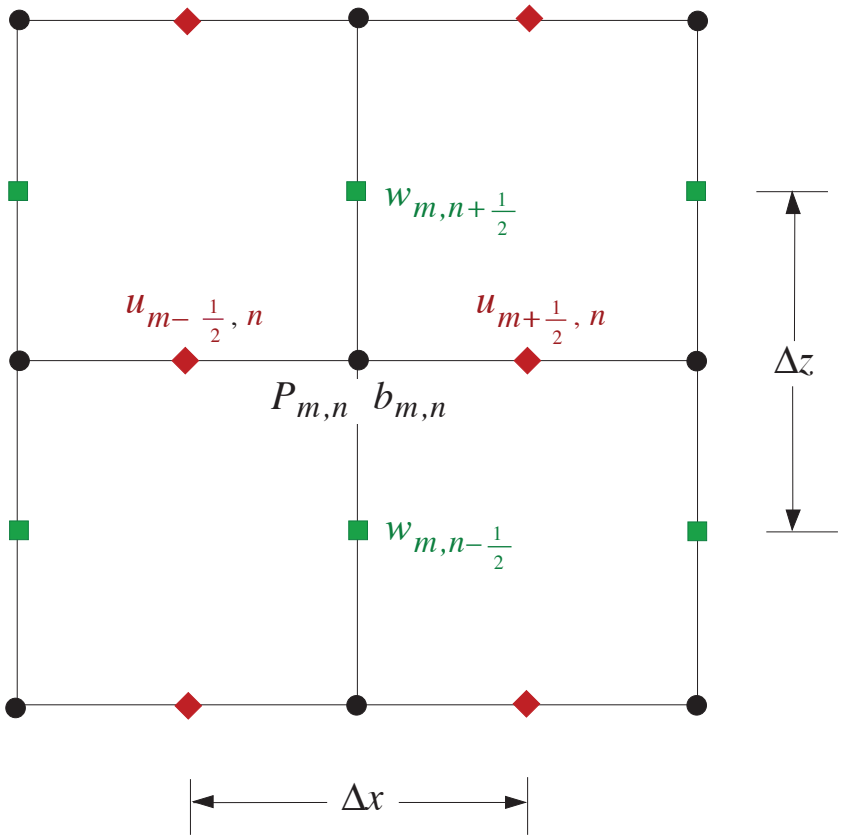
Utility of High Order II

Is it best to approximate all terms with differences having the same order of accuracy?

- Yes – if you are trying to achieve convergence.
- Not particularly if you are trying to improve the representation of poorly resolved scales.

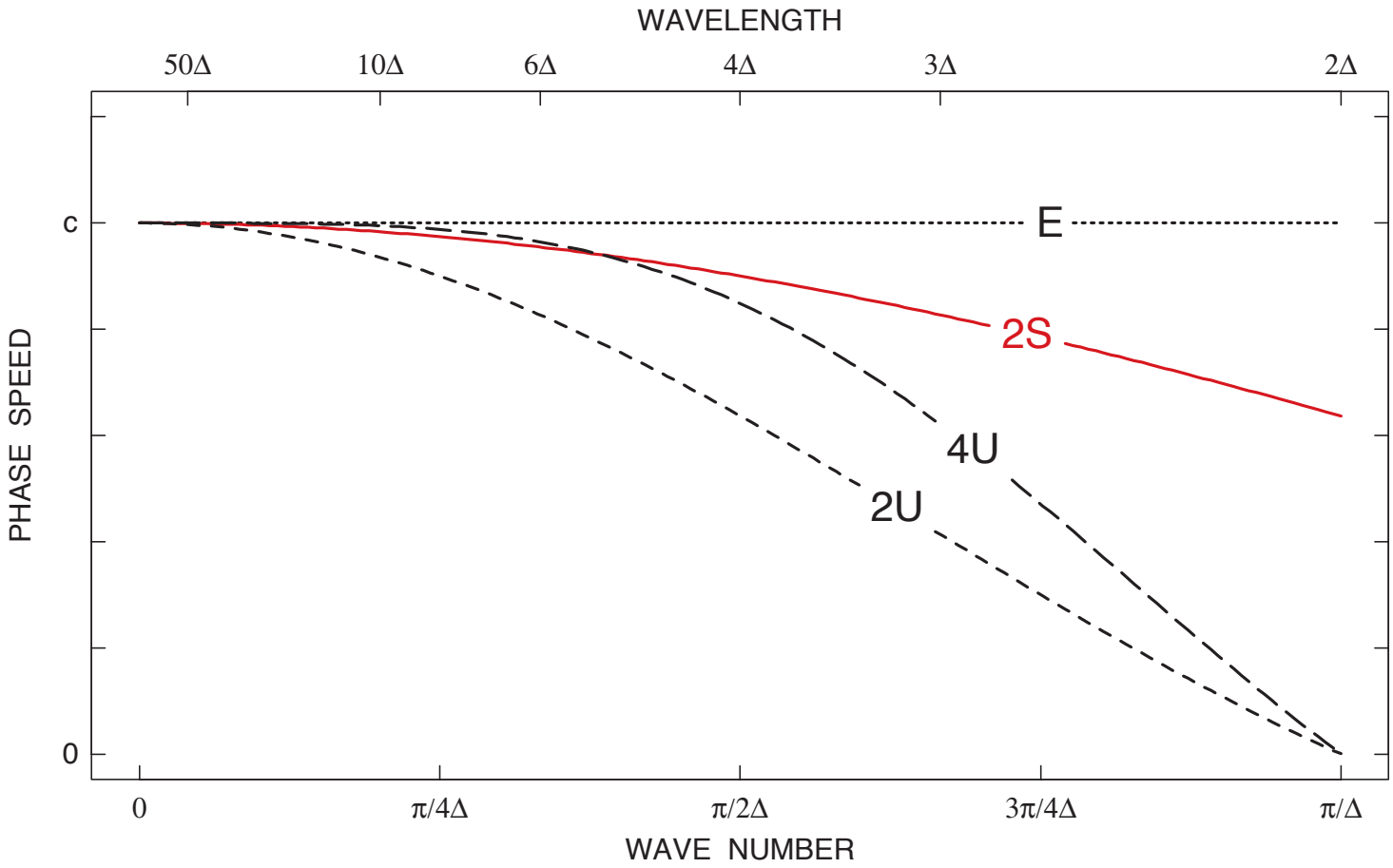
Example: Staggered Meshes

Arakawa C-grid



Example: Staggered Meshes II

Phase speeds using staggered (S) or unstaggered (U) 2nd- or 4th-order differences

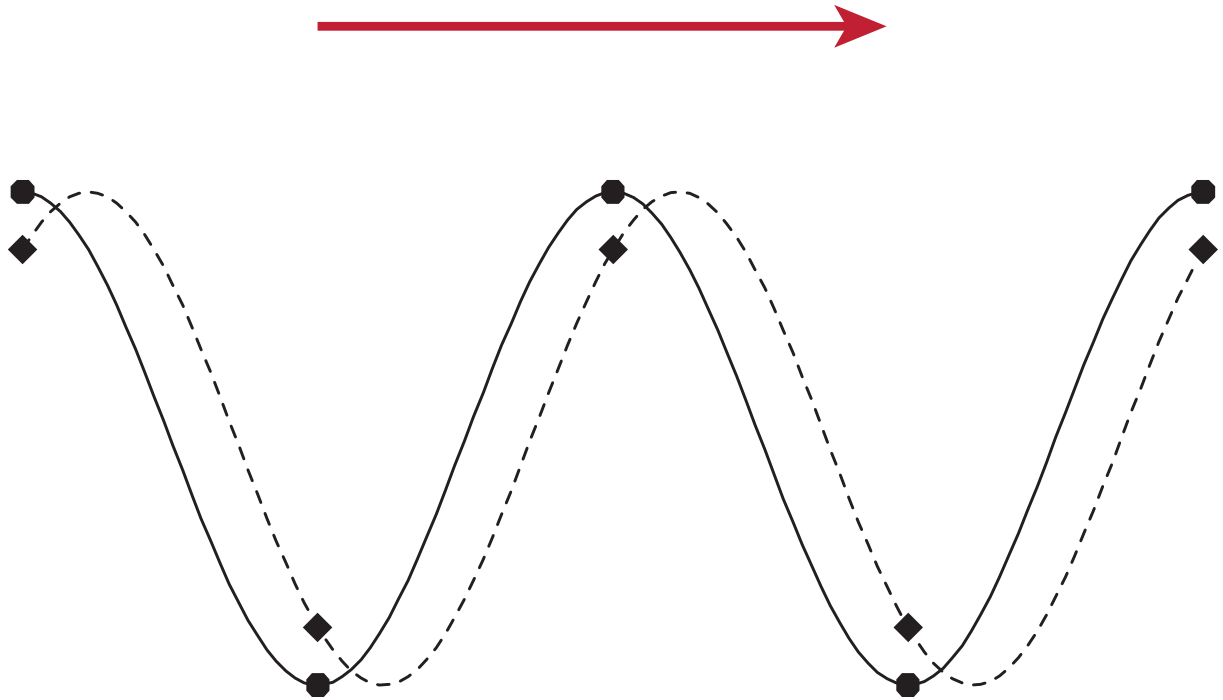


Trouble with $2\Delta x$-Waves

Finite-difference methods do not propagate $2\Delta x$-waves on an unstaggered mesh. Why not?

Trouble with $2\Delta x$-Waves

Finite-difference methods do not propagate $2\Delta x$-waves on an unstaggered mesh. Why not?

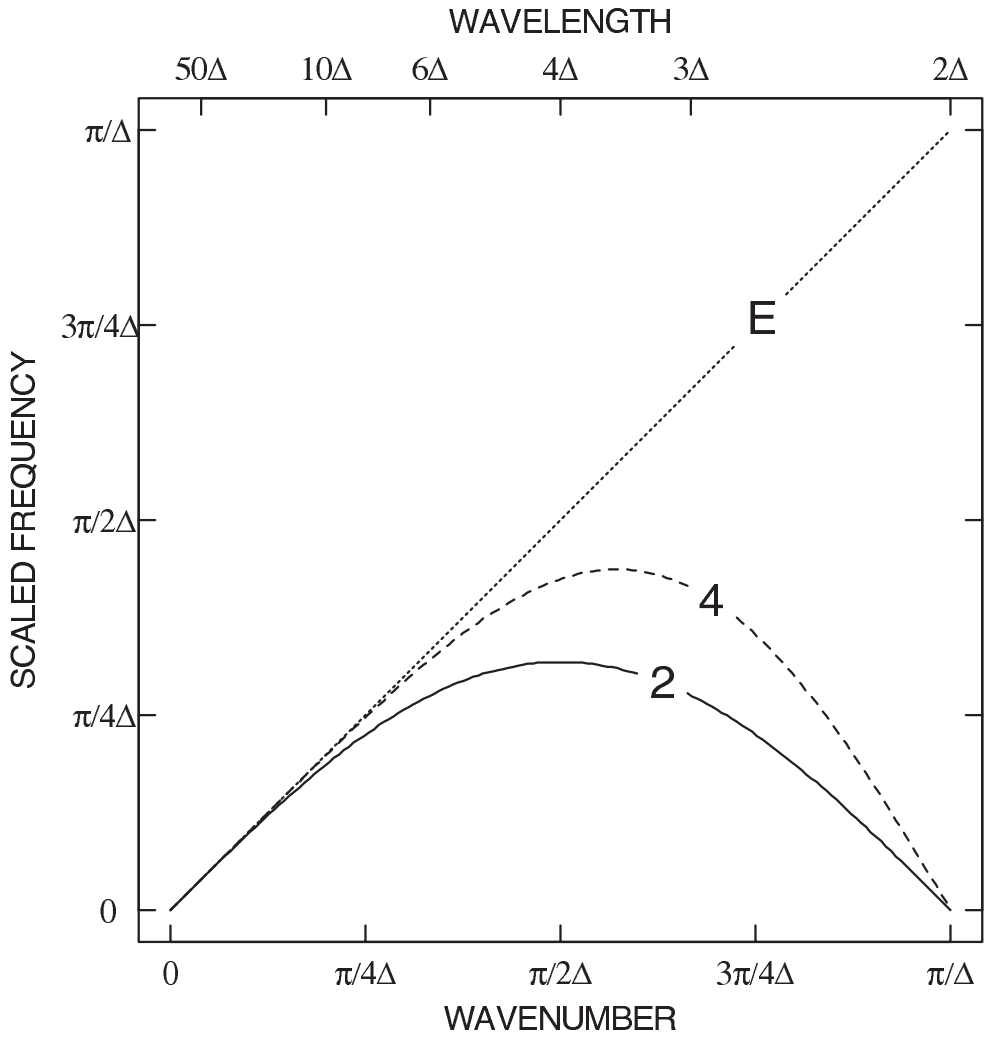


$2\Delta x$-Waves and Negative Group Velocities

Animation of $2\Delta x$-wide spike simulated by upstream and by explicit centered 2nd and 4th-order finite differences.

2Δx-Waves and Negative Group Velocities

Group velocity is $\frac{\partial(\text{frequency})}{\partial(\text{wavenumber})}$



Killing Off $2\Delta x$-Waves

- $2\Delta x$ waves are in serious error
- Very short waves can generate aliasing error

Killing Off $2\Delta x$-Waves

- $2\Delta x$ waves are in serious error
- Very short waves can generate aliasing error

Remove the very short waves via:

- A scale-selective global filter

Killing Off $2\Delta x$-Waves

- $2\Delta x$ waves are in serious error
- Very short waves can generate aliasing error

Remove the very short waves via:

- A scale-selective global filter
- (Nonlinear) local filters that become active only in regions with large-amplitude short-waves

Killing Off $2\Delta x$-Waves

- $2\Delta x$ waves are in serious error
- Very short waves can generate aliasing error

Remove the very short waves via:

- A scale-selective global filter
- (Nonlinear) local filters that become active only in regions with large-amplitude short-waves
  - e.g., WENO finite difference methods

Killing Off $2\Delta x$-Waves

- $2\Delta x$ waves are in serious error
- Very short waves can generate aliasing error

Remove the very short waves via:

- A scale-selective global filter
- (Nonlinear) local filters that become active only in regions with large-amplitude short-waves
  - e.g., WENO finite difference methods
  - Major focus of finite-volume methods

Finite Volume Methods

- Unknowns are *cell averages*:

$$\phi_j^n \approx \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \psi(x, n\Delta t)\, dx$$

Finite Volume Methods

- Unknowns are *cell averages*:

$$\phi_j^n \approx \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \psi(x, n\Delta t)\, dx$$

- *Essential* for the simulation of shocks
  - Weak solutions satisfy integral form of conservation law

Finite Volume Methods

- Unknowns are *cell averages*:

$$\phi_j^n \approx \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \psi(x, n\Delta t)\, dx$$

- *Essential* for the simulation of shocks
  - Weak solutions satisfy integral form of conservation law
- Approximations most naturally in flux form

Finite Volume Methods

- Unknowns are *cell averages*:

$$\phi_j^n \approx \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \psi(x, n\Delta t)\, dx$$

- *Essential* for the simulation of shocks
  - Weak solutions satisfy integral form of conservation law
- Approximations most naturally in flux form
- Fluxes at cell boundaries computed from sub-cell reconstructions

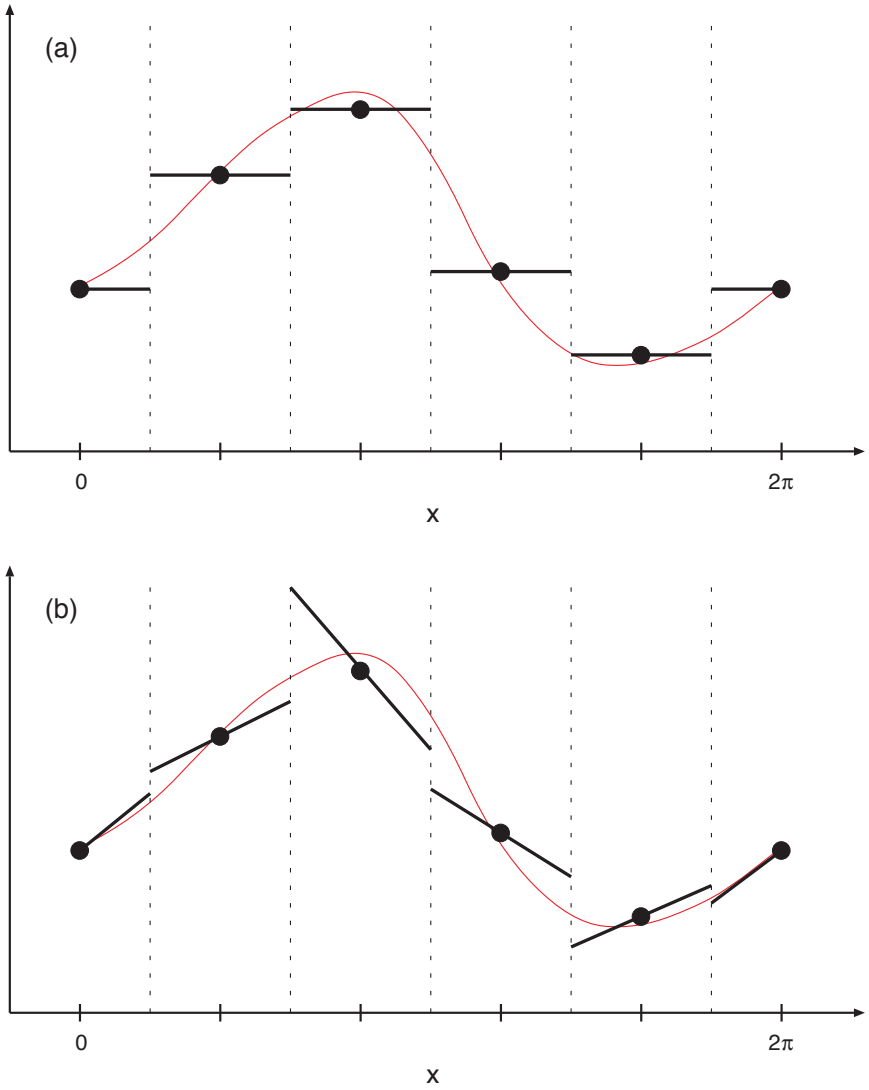
Finite Volume Methods

- Unknowns are *cell averages*:

$$\phi_j^n \approx \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} \psi(x, n\Delta t)\, dx$$

- *Essential* for the simulation of shocks
  - Weak solutions satisfy integral form of conservation law
- Approximations most naturally in flux form
- Fluxes at cell boundaries computed from sub-cell reconstructions
- *Leads directly to two-level forward-in-time schemes*

Sub-Cell Reconstructions



Avoiding overshoots and undershoots

When it it important?

Avoiding overshoots and undershoots

When it it important? *When simulating shocks.* Consistent monotone methods in flux form converge to the entropy consistent solution.

Avoiding overshoots and undershoots

When it it important? *When simulating shocks.* Consistent monotone methods in flux form converge to the entropy consistent solution.

$$\phi_j^{n+1} = H(\phi_{j-p}^n, \ldots, \phi_{j+q+1}^n),$$

is *monotone* if

$$\frac{\partial\, H(\phi_{j-p}, \ldots, \phi_{j+q+1})}{\partial \phi_i} \geq 0$$

for each integer i in the interval $[j-p,\ j+q+1]$.

Avoiding overshoots and undershoots

When it it important? *When simulating shocks.* Consistent monotone methods in flux form converge to the entropy consistent solution.

$$\phi_j^{n+1} = H(\phi_{j-p}^n, \ldots, \phi_{j+q+1}^n),$$

is *monotone* if

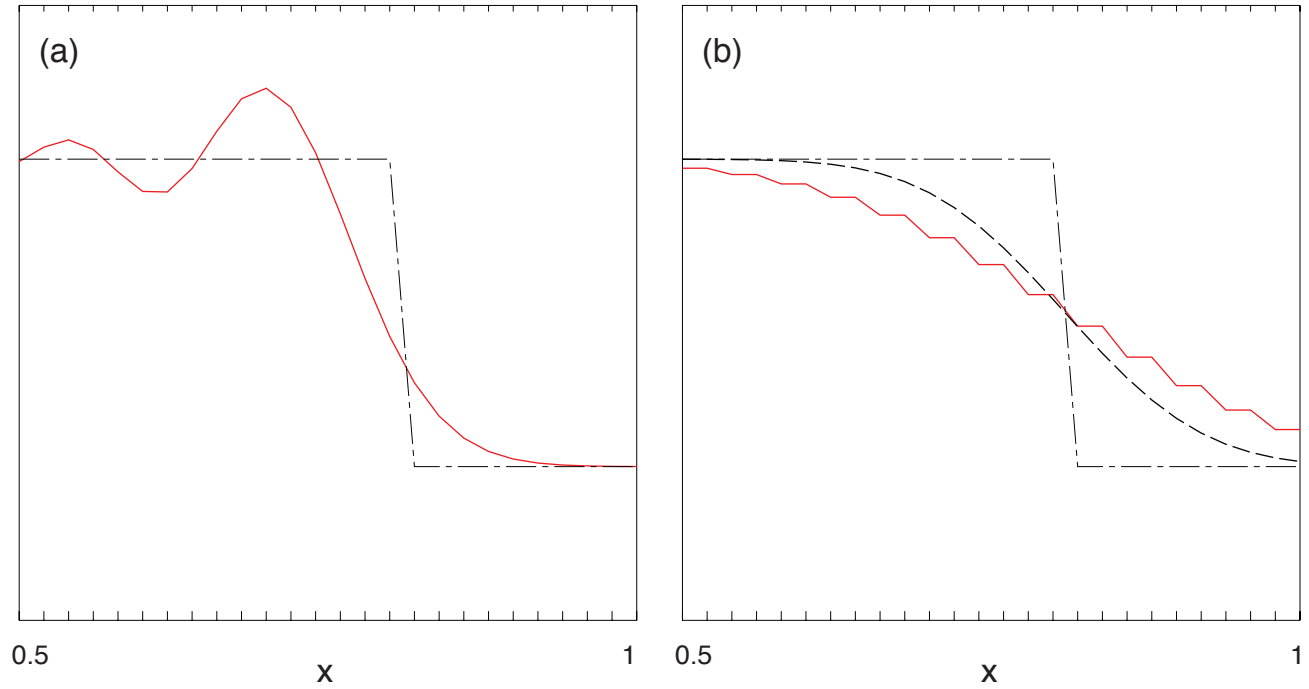
$$\frac{\partial\, H(\phi_{j-p}, \ldots, \phi_{j+q+1})}{\partial \phi_i} \geq 0$$

for each integer i in the interval $[j-p,\ j+q+1]$.

Monotone schemes

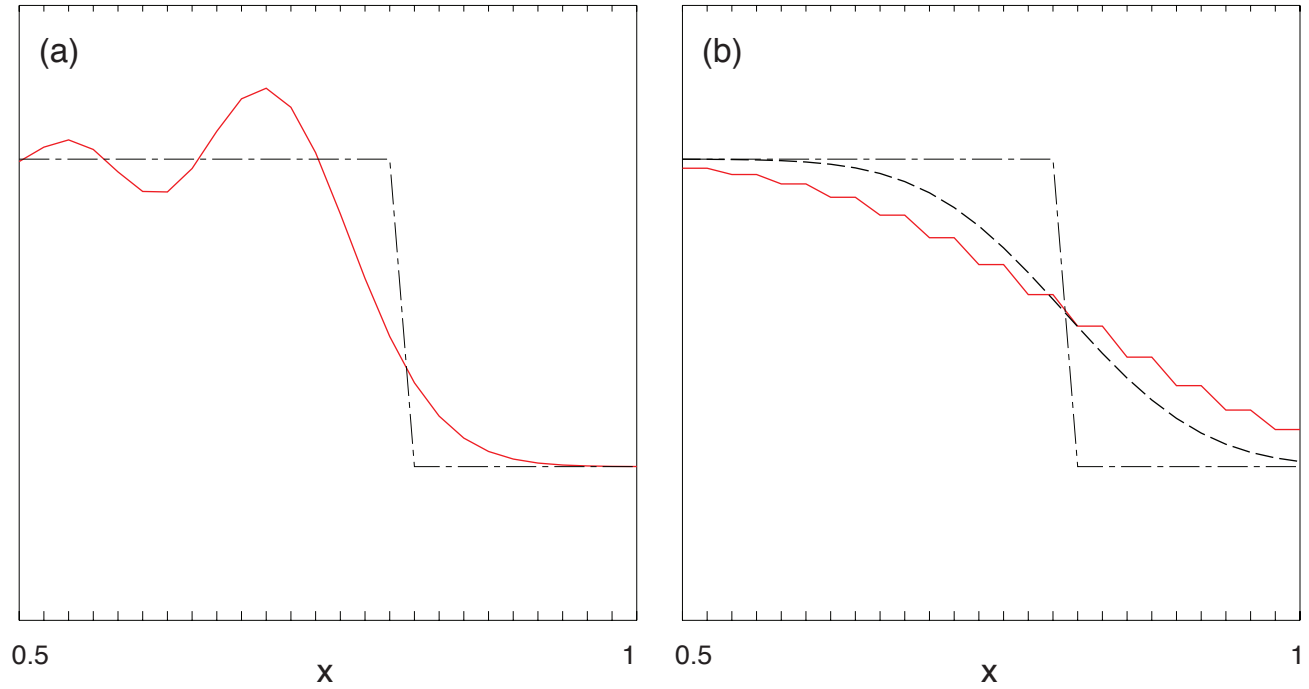
- Do not produce spurious ripples.
- Are first-order accurate.

Advection of a Step Function



- Left: 2nd-order centered with global 4th-derivative smoother
- Right: Upstream and Lax-Friedrichs (red) monotone methods

Advection of a Step Function



- Left: 2nd-order centered with global 4th-derivative smoother
- Right: Upstream and Lax-Friedrichs (red) monotone methods

Strategy: Scheme becomes monotone near discontinuities and is high-order elsewhere.

Avoiding overshoots and undershoots

When is it important in problems without shocks?

Avoiding overshoots and undershoots

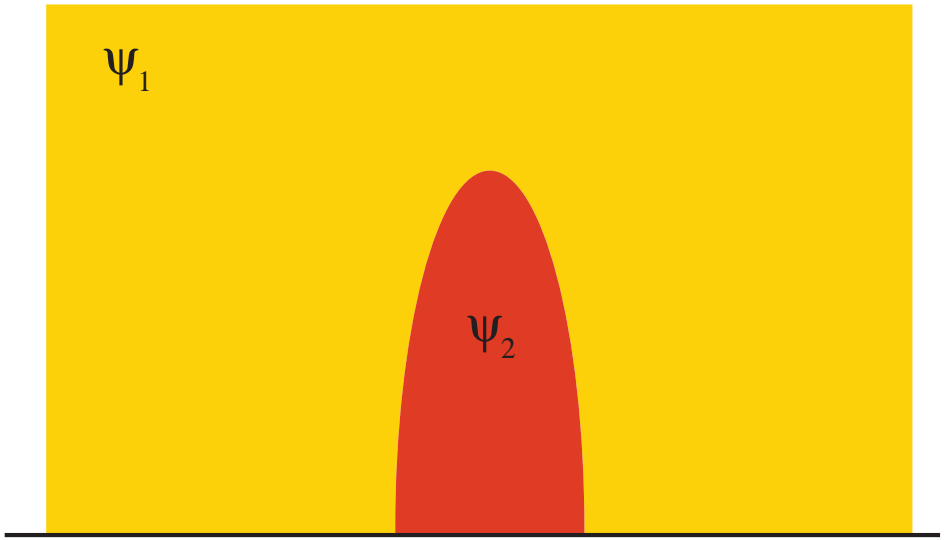
When is it important in problems without shocks? *Chemically reacting flow*

$$\begin{aligned} \frac{\partial \psi_1}{\partial t} + c\frac{\partial \psi_1}{\partial x} &= -r\psi_1\psi_2, \\ \frac{\partial \psi_2}{\partial t} + c\frac{\partial \psi_2}{\partial x} &= r\psi_1\psi_2. \end{aligned}$$

Avoiding overshoots and undershoots

When is it important in problems without shocks? *Chemically reacting flow*

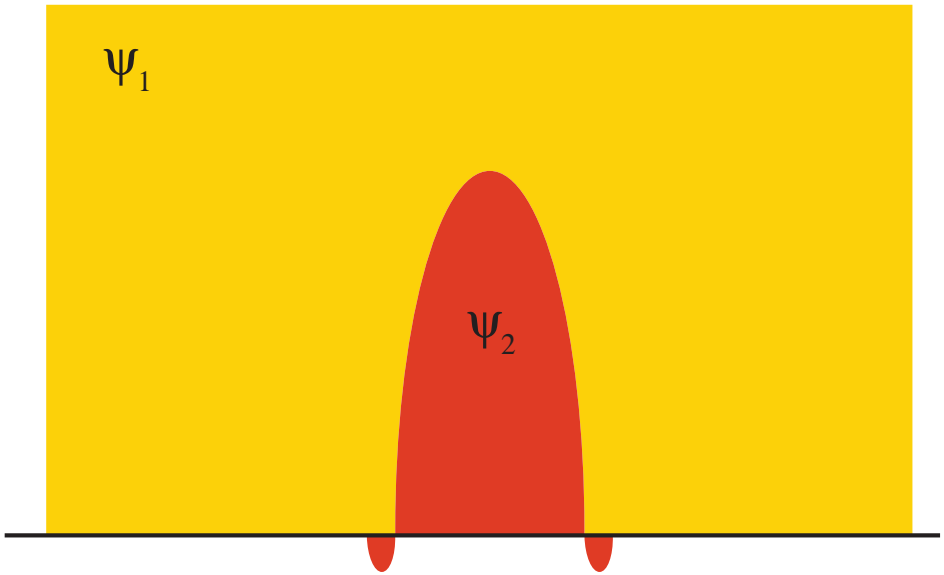
$$\frac{\partial \psi_1}{\partial t} + c\frac{\partial \psi_1}{\partial x} = -r\psi_1\psi_2,$$
$$\frac{\partial \psi_2}{\partial t} + c\frac{\partial \psi_2}{\partial x} = r\psi_1\psi_2.$$



Avoiding overshoots and undershoots

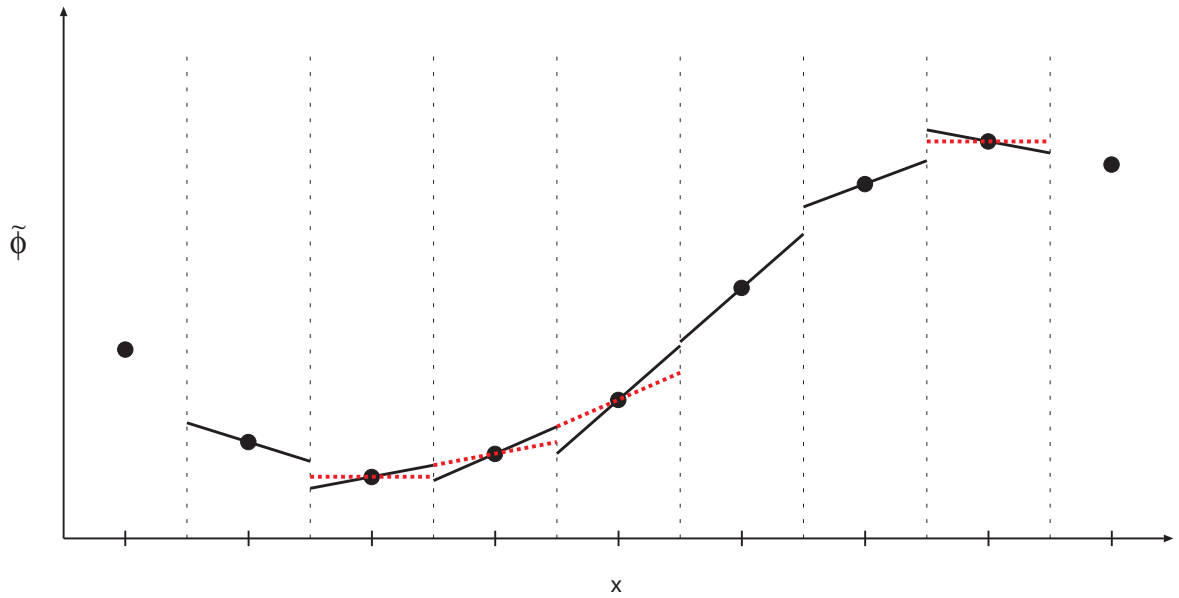
When is it important in problems without shocks? *Chemically reacting flow*

$$\frac{\partial \psi_1}{\partial t} + c\frac{\partial \psi_1}{\partial x} = -r\psi_1\psi_2,$$
$$\frac{\partial \psi_2}{\partial t} + c\frac{\partial \psi_2}{\partial x} = r\psi_1\psi_2.$$



Minmod limiter

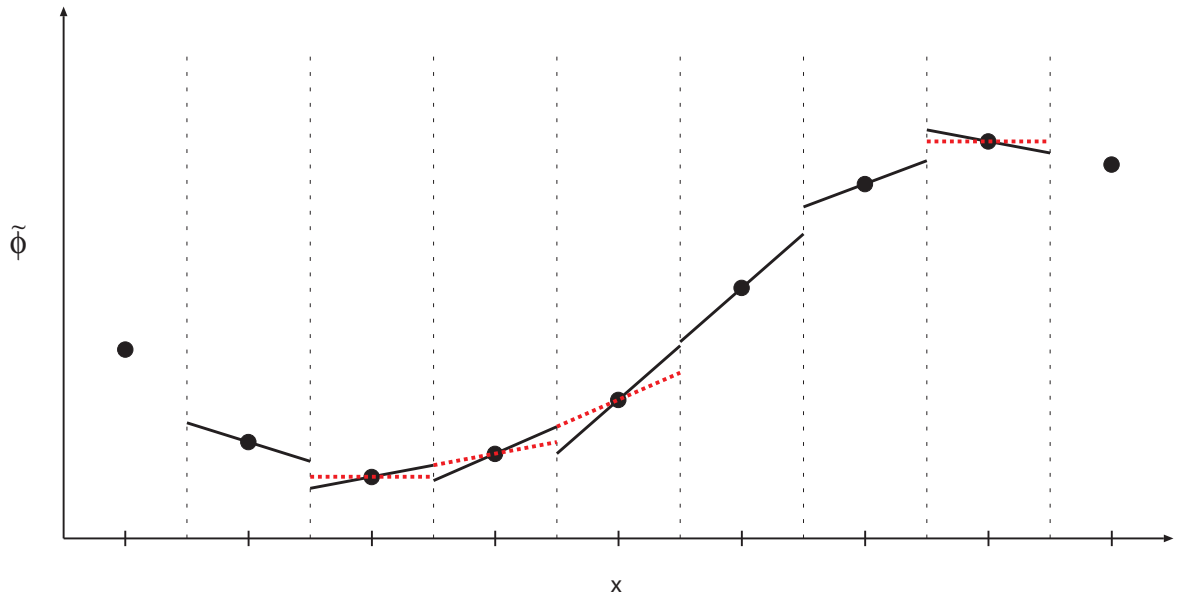
One local-smoothing strategy to prevent growth of ripples.



- If $\mathrm{sgn}(\phi_{j+1} - \phi_j) \neq \mathrm{sgn}(\phi_j - \phi_{j-1})$ slope in cell j is zero
- Otherwise | slope | is $\min(|\phi_{j+1} - \phi_j|, |\phi_j - \phi_{j-1}|)$

Minmod limiter

One local-smoothing strategy to prevent growth of ripples.

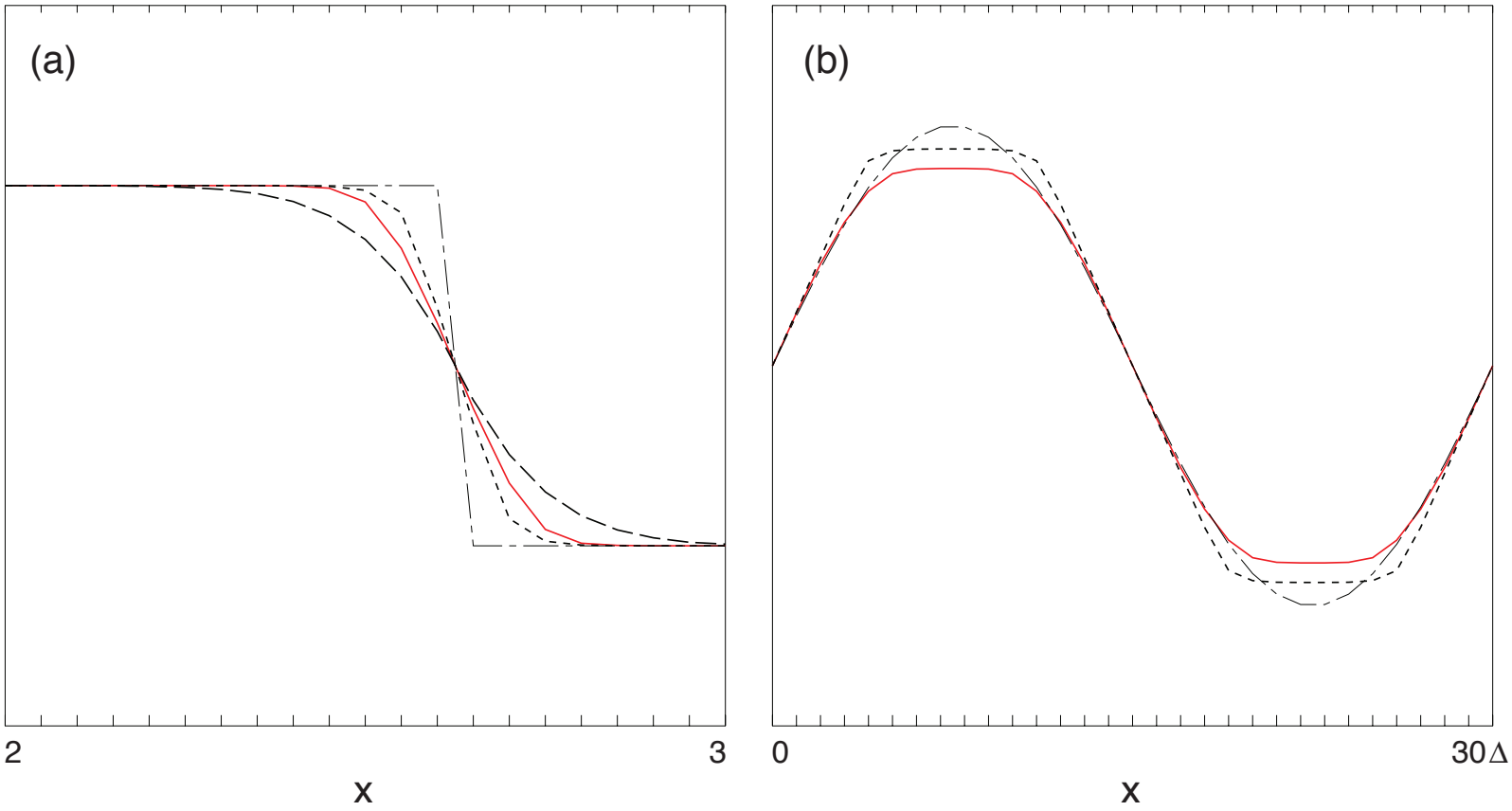


- If $\mathrm{sgn}(\phi_{j+1} - \phi_j) \neq \mathrm{sgn}(\phi_j - \phi_{j-1})$ slope in cell j is zero
- Otherwise | slope | is $\min(|\phi_{j+1} - \phi_j|, |\phi_j - \phi_{j-1}|)$

Scheme is TVD, but not monotone.

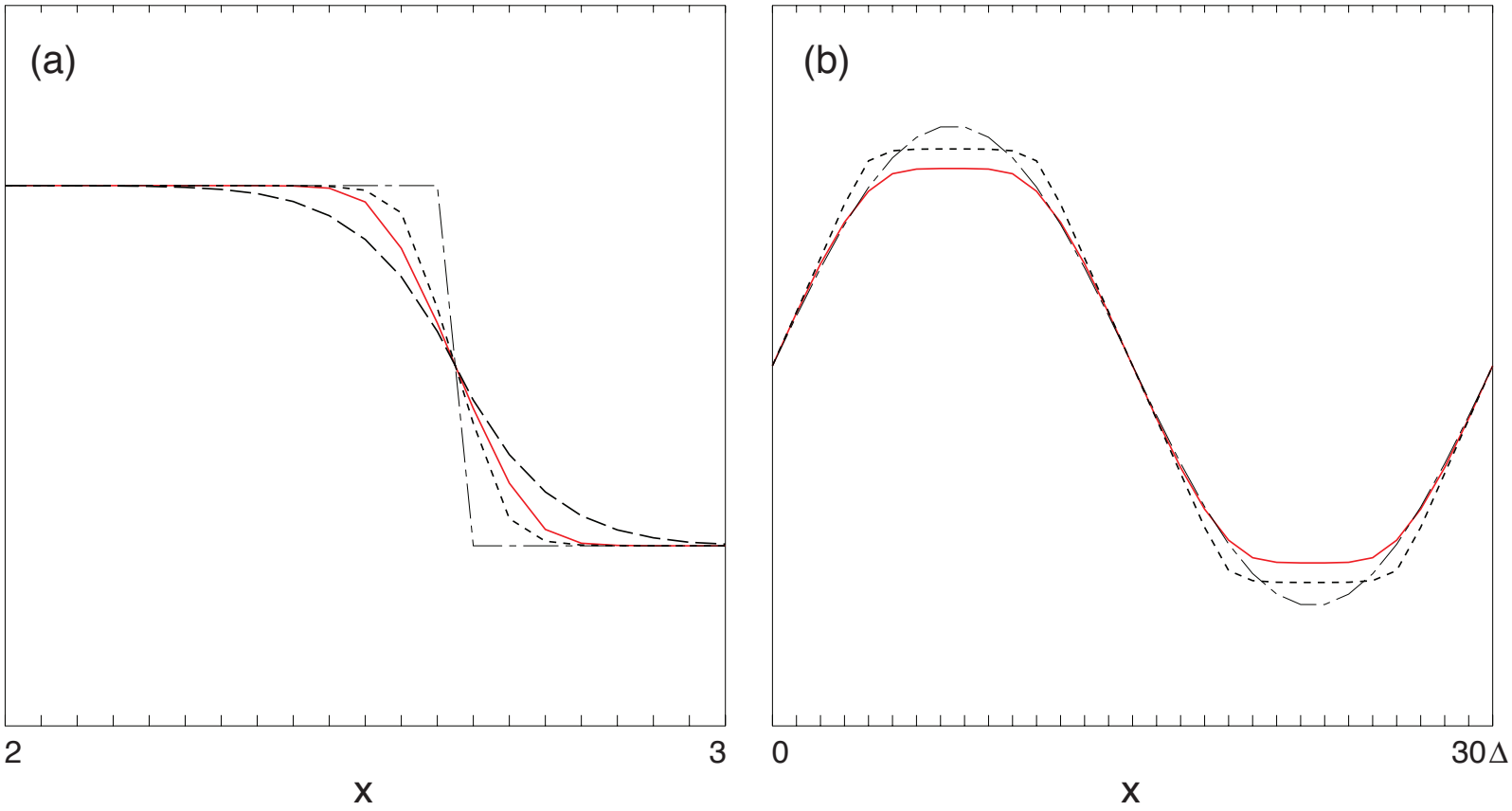
Limiters in Action

Minmod (long-dashed), MC (red), Superbee (short-dashed)



Limiters in Action

Minmod (long-dashed), MC (red), Superbee (short-dashed)



Nice job at jump, but messes up smooth extremum.

Order of Accuracy

Scheme	*Estimated Order of Accuracy*
Upstream	0.9
Minmod Limiter	1.6
Superbee Limiter	1.6
Zalesak FCT	1.7
MC Limiter	1.9
Lax–Wendroff	2.0

Table: Empirically determined order of accuracy for constant-wind-speed advection of a sine wave

Order of Accuracy

Scheme	*Estimated Order of Accuracy*
Upstream	0.9
Minmod Limiter	1.6
Superbee Limiter	1.6
Zalesak FCT	1.7
MC Limiter	1.9
Lax–Wendroff	2.0

Table: Empirically determined order of accuracy for constant-wind-speed advection of a sine wave

Avoid limiting smooth extrema!

Orthogonal Expansion Functions

- Unknowns are coefficients of expansion functions $a_k(t)$

$$\phi(x,t) = \sum_{k=1}^{N} a_k(t)\varphi_k(x)$$

Orthogonal Expansion Functions

- Unknowns are coefficients of expansion functions $a_k(t)$

$$\phi(x,t) = \sum_{k=1}^{N} a_k(t)\varphi_k(x)$$

- Evolution equations for a_k obtained via
  - Galerkin requirement that residual be orthogonal to all $\varphi_k(x)$
    - “Spectral"
    - Global conservation of ϕ and ϕ^2

Orthogonal Expansion Functions

- Unknowns are coefficients of expansion functions $a_k(t)$

$$\phi(x,t) = \sum_{k=1}^{N} a_k(t)\varphi_k(x)$$

- Evolution equations for a_k obtained via
  - Galerkin requirement that residual be orthogonal to all $\varphi_k(x)$
    - “Spectral"
    - Global conservation of ϕ and ϕ^2
  - Collocation requirement that sets the residual to zero at a set of grid points
    - “Pseudo-spectral"
    - 50% faster than spectral
    - Lose conservation

Orthogonal Expansion Functions – II

Using orthogonal expansion functions

- Decouples the evolution equations for the a_k (good for explicit time differencing).

Orthogonal Expansion Functions – II

Using orthogonal expansion functions

- Decouples the evolution equations for the a_k (good for explicit time differencing).
- Computation of the forcing for all N of the a_k involves $O(N^2)$ operations (Fourier methods can be $O(N \log N)$).

Orthogonal Expansion Functions – II

Using orthogonal expansion functions

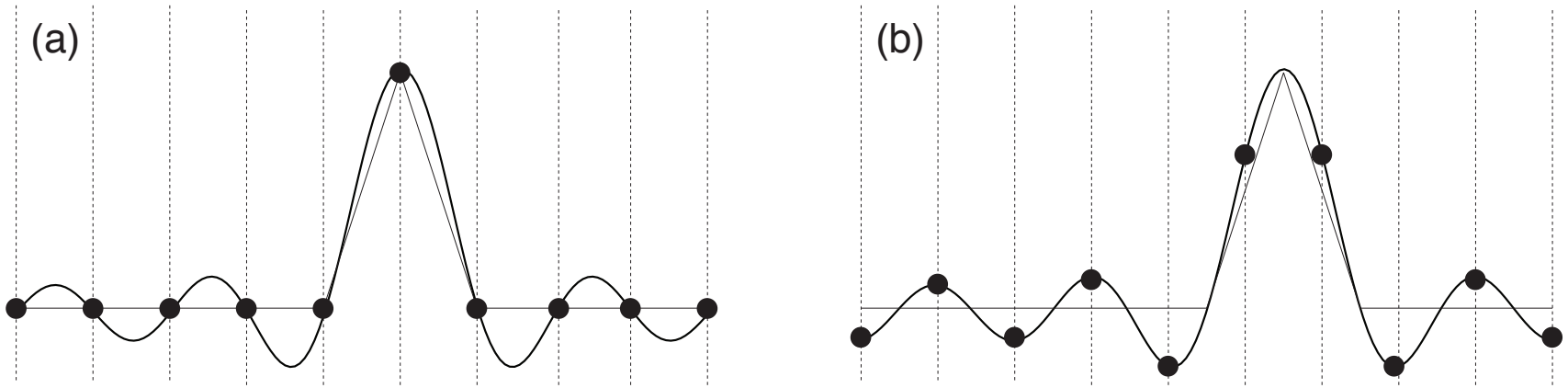
- Decouples the evolution equations for the a_k (good for explicit time differencing).
- Computation of the forcing for all N of the a_k involves $O(N^2)$ operations (Fourier methods can be $O(N \log N)$).
- “Spectral accuracy" when approximating smooth functions — error goes to zero faster than any finite power of the effective grid spacing.

Orthogonal Expansion Functions – II

Using orthogonal expansion functions

- Decouples the evolution equations for the a_k (good for explicit time differencing).
- Computation of the forcing for all N of the a_k involves $O(N^2)$ operations (Fourier methods can be $O(N \log N)$).
- “Spectral accuracy" when approximating smooth functions — error goes to zero faster than any finite power of the effective grid spacing.
- Not conducive to preserving positivity or treating steep gradients

Global Overshoots and Undershoots with Poor Resolution



Global Spectral Model

- Expansion functions are spherical harmonics.
- Avoids problems with short time steps at poles.

Global Spectral Model

- Expansion functions are spherical harmonics.
- Avoids problems with short time steps at poles.
- Efficient semi-implicit approximation of the pressure gradient terms.

Global Spectral Model

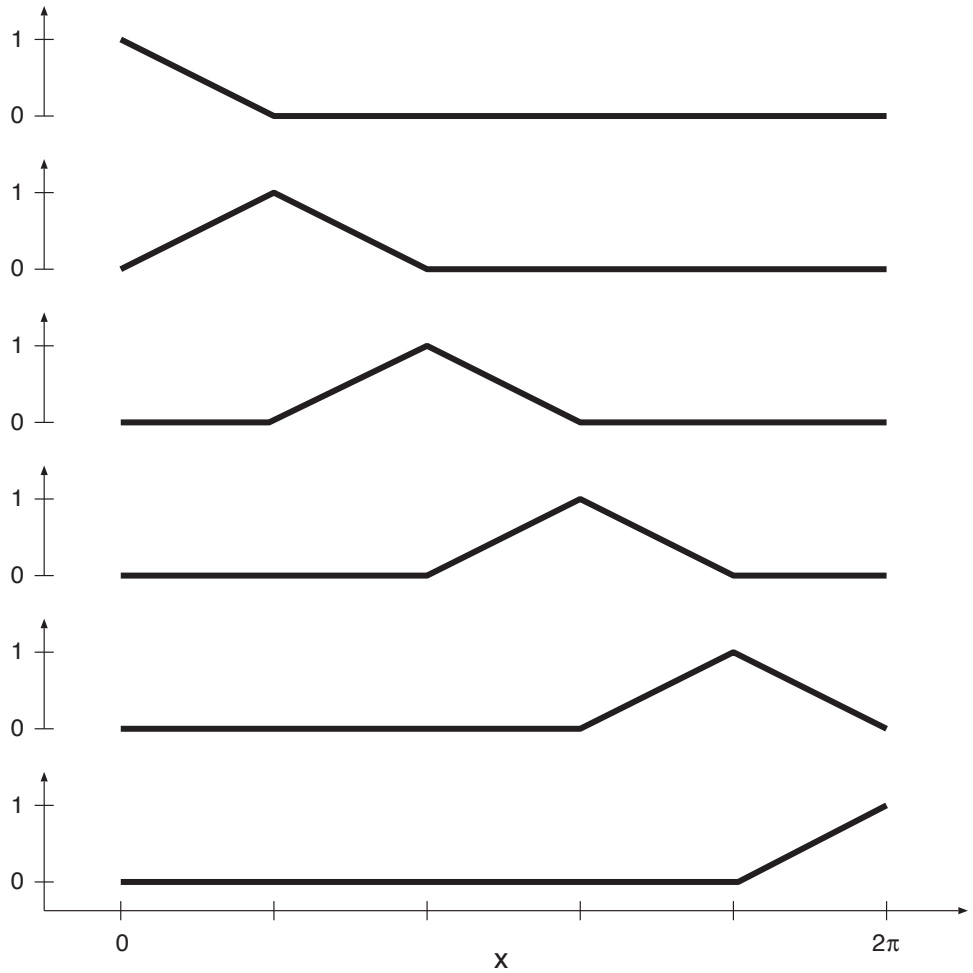
- Expansion functions are spherical harmonics.
- Avoids problems with short time steps at poles.
- Efficient semi-implicit approximation of the pressure gradient terms.
- Common choice for the dynamical variables in global hydrostatic models.

Global Spectral Model

- Expansion functions are spherical harmonics.
- Avoids problems with short time steps at poles.
- Efficient semi-implicit approximation of the pressure gradient terms.
- Common choice for the dynamical variables in global hydrostatic models.
- Moisture variables often finite volume or finite difference.

Finite Element Methods

Piecewise linear elements (chapeau functions)



Finite Element Methods

Expansion functions have compact support

- Evaluation of the forcing for all N of the a_k involves only $O(N)$ operations.

Finite Element Methods

Expansion functions have compact support

- Evaluation of the forcing for all N of the a_k involves only $O(N)$ operations.
- The evolution equations for the a_k are coupled
  - Implicit solve every time step
  - Very inefficient for wave-propagation problems

Finite Element Methods

Expansion functions have compact support

- Evaluation of the forcing for all N of the a_k involves only $O(N)$ operations.
- The evolution equations for the a_k are coupled
  - Implicit solve every time step
  - Very inefficient for wave-propagation problems
- Implicit coupling in quadratic or cubic finite element methods makes them completely impractical for the simulation of waves.

Finite Element Methods

Expansion functions have compact support

- Evaluation of the forcing for all N of the a_k involves only $O(N)$ operations.
- The evolution equations for the a_k are coupled
  - Implicit solve every time step
  - Very inefficient for wave-propagation problems
- Implicit coupling in quadratic or cubic finite element methods makes them completely impractical for the simulation of waves.
- Higher-order finite element methods are the nevertheless the standard approach for solving may elliptic problems.

Finite Element versus Compact Differencing in 1D

1D advection equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0$$

Finite Element versus Compact Differencing in 1D

1D advection equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0$$

Piecewise linear finite-elements:

$$\frac{d}{dt}\left(\frac{a_{j+1} + 4a_j + a_{j-1}}{6}\right) + c\left(\frac{a_{j+1} - a_{j-1}}{2\Delta x}\right) = 0$$

Finite Element versus Compact Differencing in 1D

1D advection equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0$$

Piecewise linear finite-elements:

$$\frac{d}{dt}\left(\frac{a_{j+1} + 4a_j + a_{j-1}}{6}\right) + c\left(\frac{a_{j+1} - a_{j-1}}{2\Delta x}\right) = 0$$

4th-order compact scheme:

$$\frac{d\phi_j}{dt} + c\left(\frac{d\psi}{dx}\right)_j = 0$$

$$\frac{1}{6}\left[\left(\frac{d\psi}{dx}\right)_{j+1} + 4\left(\frac{d\psi}{dx}\right)_j + \left(\frac{d\psi}{dx}\right)_{j-1}\right] = \frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x}$$

Finite Element versus Compact Differencing in 1D

Schemes are identical!

Finite Element versus Compact Differencing in 2D

4th-order compact scheme:

$$\frac{d\phi_{i,j}}{dt} + u\left(\frac{d\psi}{dx}\right)_{i,j} + v\left(\frac{d\psi}{dy}\right)_{i,j} = 0$$

$$\frac{1}{6}\left[\left(\frac{d\psi}{dx}\right)_{i+1,j} + 4\left(\frac{d\psi}{dx}\right)_{i,j} + \left(\frac{d\psi}{dx}\right)_{i-1,j}\right] = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}$$

$$\frac{1}{6}\left[\left(\frac{d\psi}{dy}\right)_{i,j+1} + 4\left(\frac{d\psi}{dy}\right)_{i,j} + \left(\frac{d\psi}{dy}\right)_{i,j-1}\right] = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y}$$

Finite Element versus Compact Differencing in 2D

4th-order compact scheme:

$$\frac{d\phi_{i,j}}{dt} + u\left(\frac{d\psi}{dx}\right)_{i,j} + v\left(\frac{d\psi}{dy}\right)_{i,j} = 0$$

$$\frac{1}{6}\left[\left(\frac{d\psi}{dx}\right)_{i+1,j} + 4\left(\frac{d\psi}{dx}\right)_{i,j} + \left(\frac{d\psi}{dx}\right)_{i-1,j}\right] = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}$$

$$\frac{1}{6}\left[\left(\frac{d\psi}{dy}\right)_{i,j+1} + 4\left(\frac{d\psi}{dy}\right)_{i,j} + \left(\frac{d\psi}{dy}\right)_{i,j-1}\right] = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y}$$

Finite-elements: huge implicit mess (element couples with itself and 8 neighbors).

The Challenge

How do we

- Extend finite-volume methods to high order (beyond piecewise parabolic or cubic)?

The Challenge

How do we

- Extend finite-volume methods to high order (beyond piecewise parabolic or cubic)?
- Extend finite-element methods to high order while avoiding implicit coupling (and other bad behaviors)?

The Challenge

How do we

- Extend finite-volume methods to high order (beyond piecewise parabolic or cubic)?
- Extend finite-element methods to high order while avoiding implicit coupling (and other bad behaviors)?
- Avoid the global coupling and $O(N^2)$ operation counts to update a set of expansion coefficients in spectral or pseudo-spectral methods?

One Solution

Use *h*-*p* methods – break domain into *h* elements and represent the solution within each element by orthogonal polynomials of maximum order *p*.

One Solution

Use *h*-*p* methods – break domain into *h* elements and represent the solution within each element by orthogonal polynomials of maximum order *p*.

- Spectral element methods – solution is continuous at cell boundaries (good for approximating 2nd-order derivatives).

One Solution

Use *h*-*p* methods – break domain into *h* elements and represent the solution within each element by orthogonal polynomials of maximum order *p*.

- Spectral element methods – solution is continuous at cell boundaries (good for approximating 2nd-order derivatives).
- Discontinuous Galerkin methods – solution is discontinuous across cell boundaries (localizes communication between cells).

Discontinuous Galerkin Method

Enforce Galerkin criterion locally, within each element.

Discontinuous Galerkin Method

Enforce Galerkin criterion locally, within each element.

Polynomial structure within each element is either

- *Modal* variant: Legendre polynomials
  - Orthogonal on $[-1, 1]$ with weight function unity

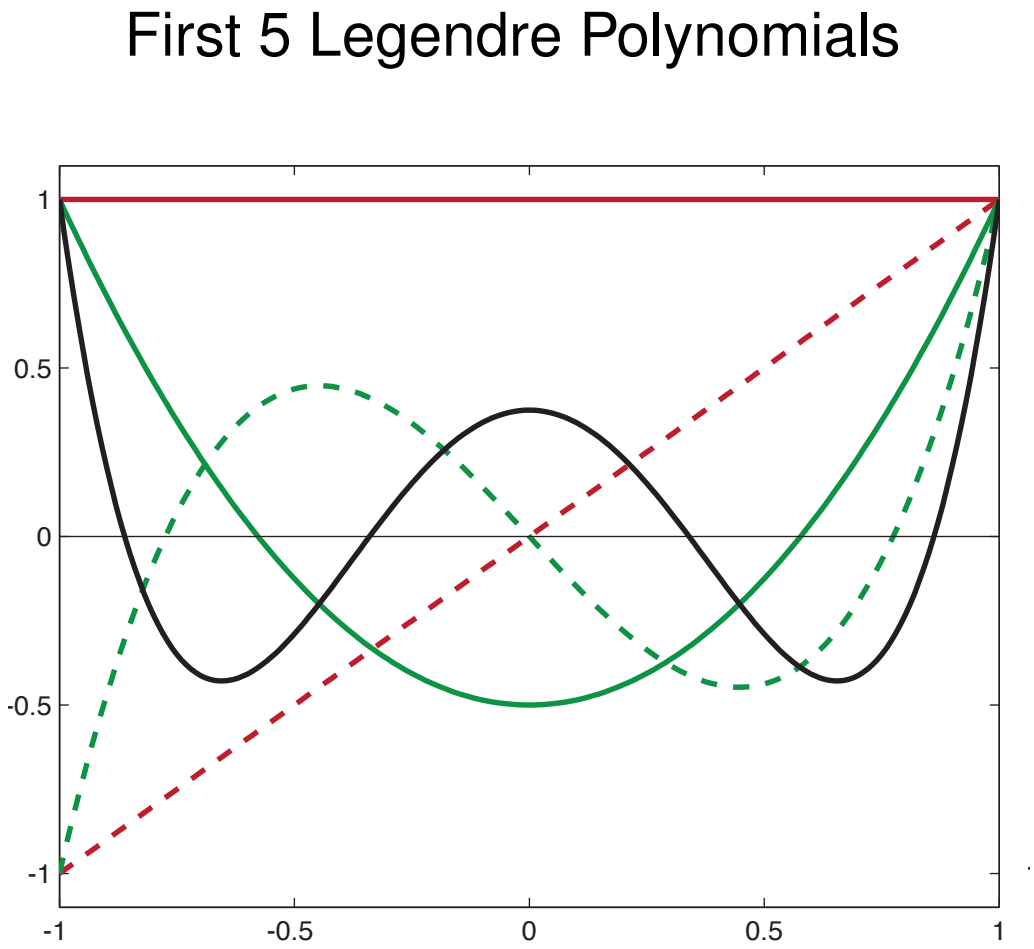
Discontinuous Galerkin Method

Enforce Galerkin criterion locally, within each element.

Polynomial structure within each element is either

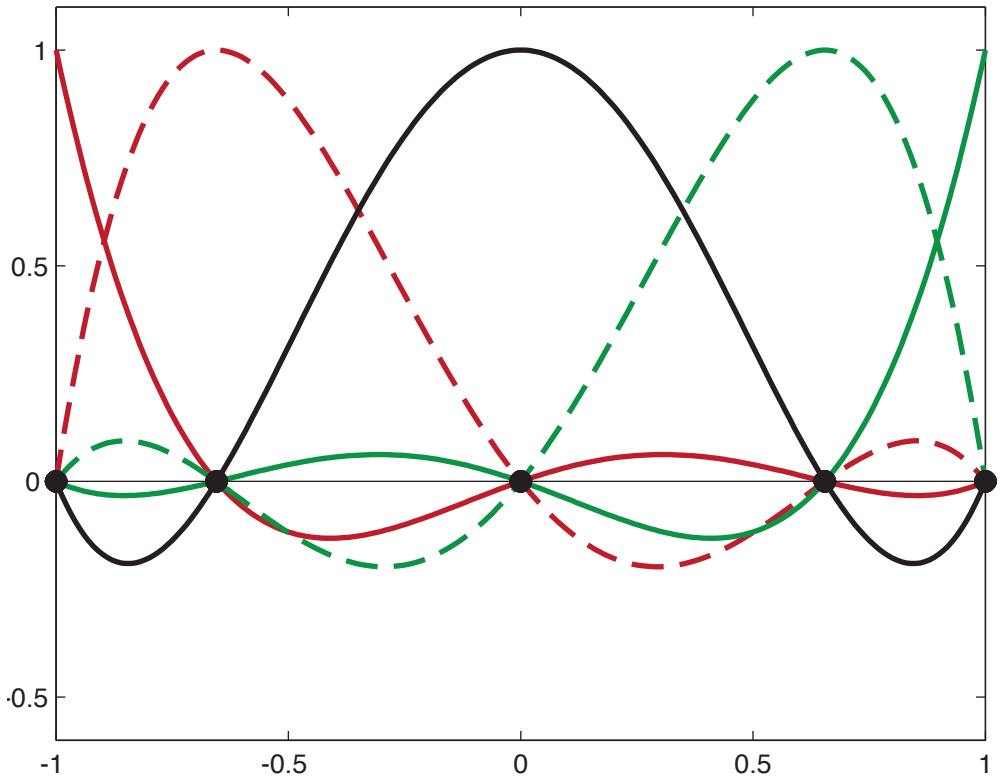
- *Modal* variant: Legendre polynomials
  - Orthogonal on $[-1, 1]$ with weight function unity
- *Nodal* variant: Lagrange polynomials interpolating the Gauss-Legendre-Lobatto (GLL) points
  - Most accurate node placement for quadrature when there is a node at each end of the interval
  - Polynomials are not truly orthogonal
  - Discrete integrals are orthogonal due to the approximate quadrature on the GLL nodes

Modal DG



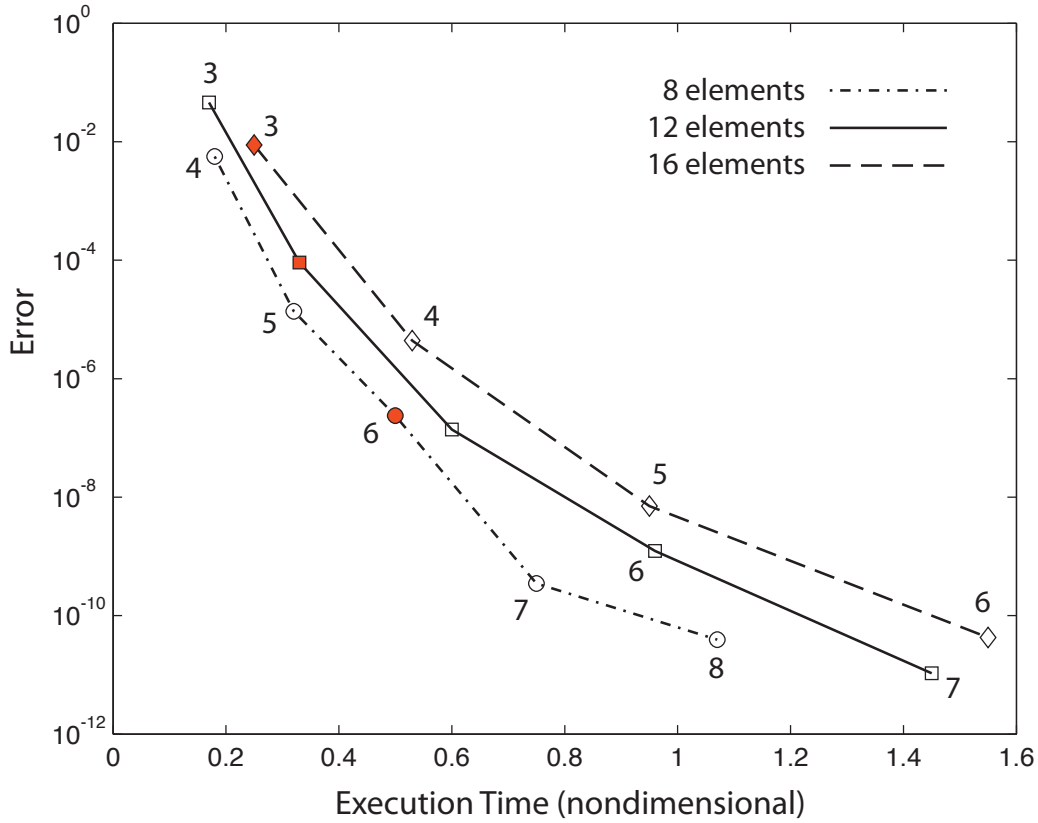
Nodal DG

Lagrange Polynomials Interpolating 5 GLL Nodes



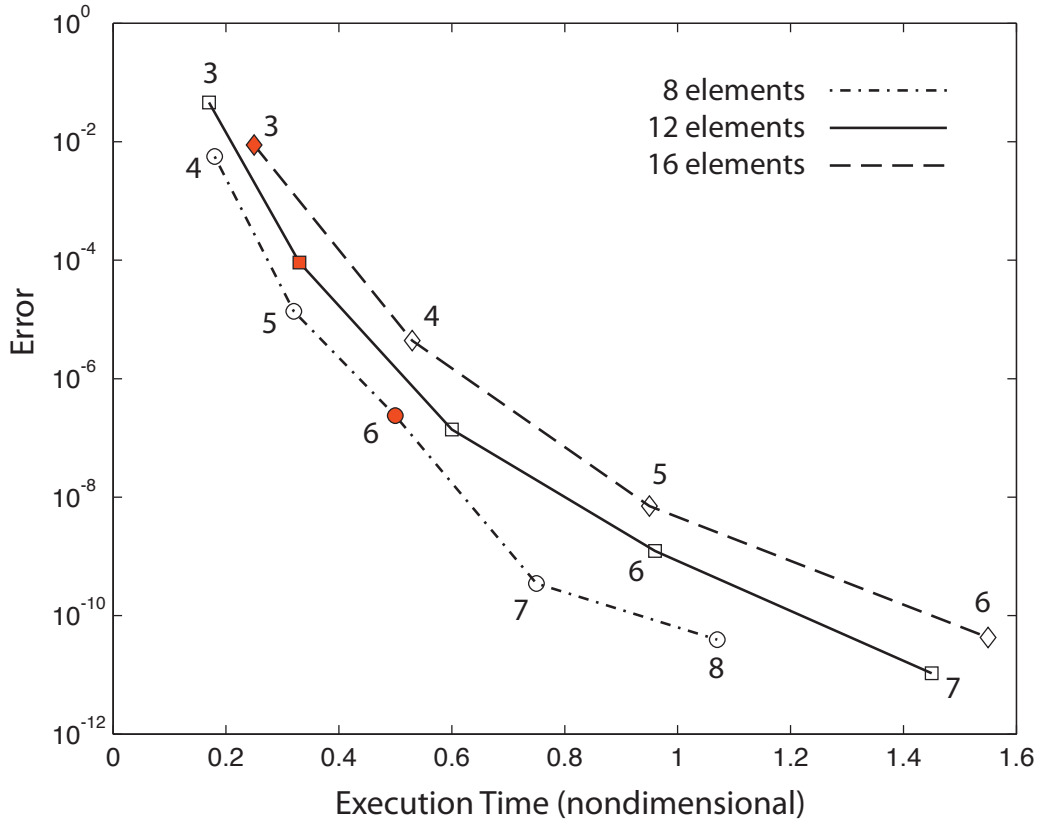
Convergence as a Function of Resolution

Nodal DG errors as a function of execution time



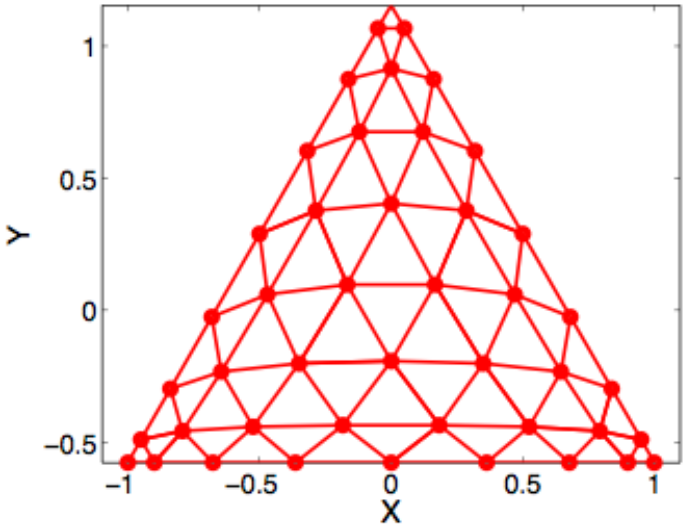
Convergence as a Function of Resolution

Nodal DG errors as a function of execution time



Most efficient way to reduce error is to increase the polynomial order.

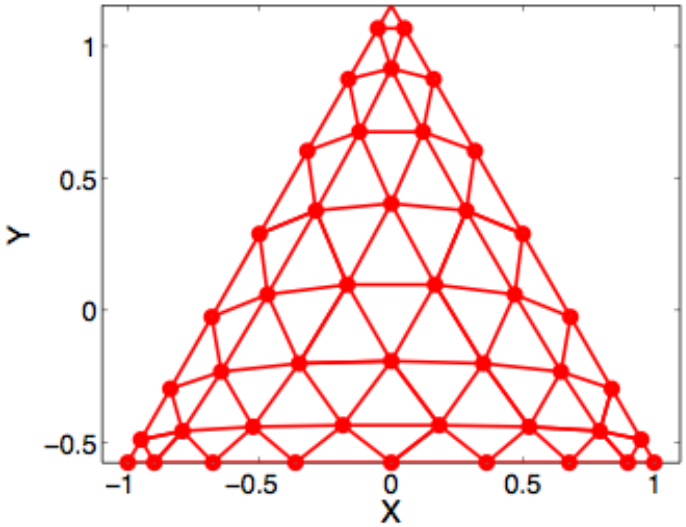
DG Considerations



Advantages:

- Suitable for massively parallel computing

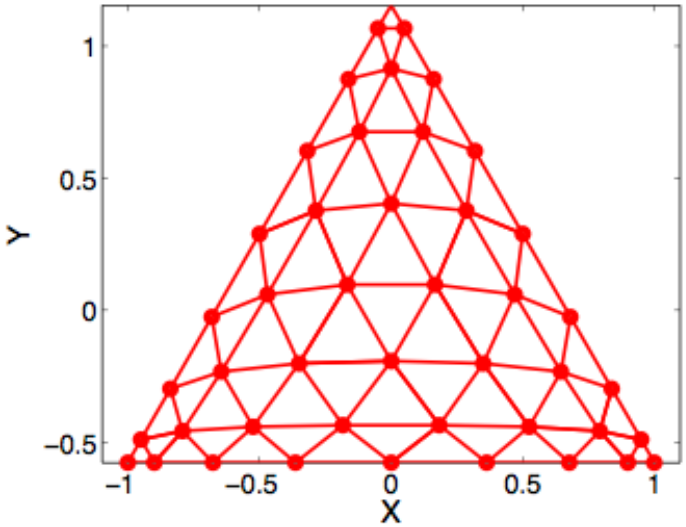
DG Considerations



Advantages:

- Suitable for massively parallel computing
  - Communicates via fluxes with nearest neighbors only

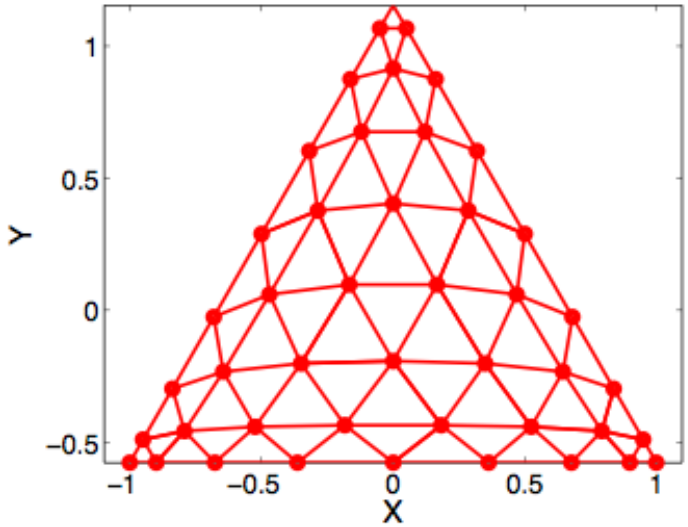
DG Considerations



Advantages:

- Suitable for massively parallel computing
  - Communicates via fluxes with nearest neighbors only
  - For high order polynomials, solution relatively insensitive to flux formulation

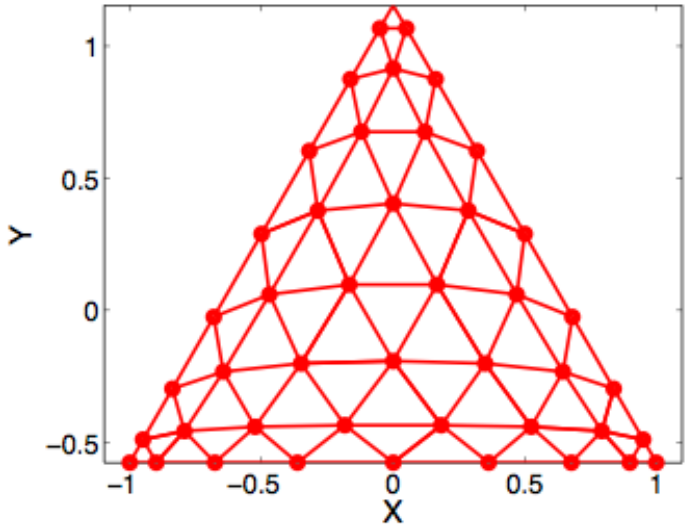
DG Considerations



Advantages:

- Suitable for massively parallel computing
  - Communicates via fluxes with nearest neighbors only
  - For high order polynomials, solution relatively insensitive to flux formulation
  - Lots of work to do within each element

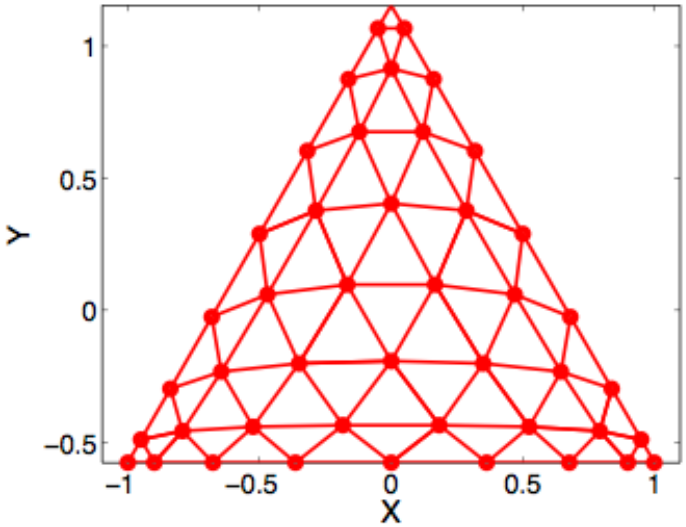
DG Considerations



Advantages:

- Suitable for massively parallel computing
  - Communicates via fluxes with nearest neighbors only
  - For high order polynomials, solution relatively insensitive to flux formulation
  - Lots of work to do within each element
- Rapid convergence for smooth solutions

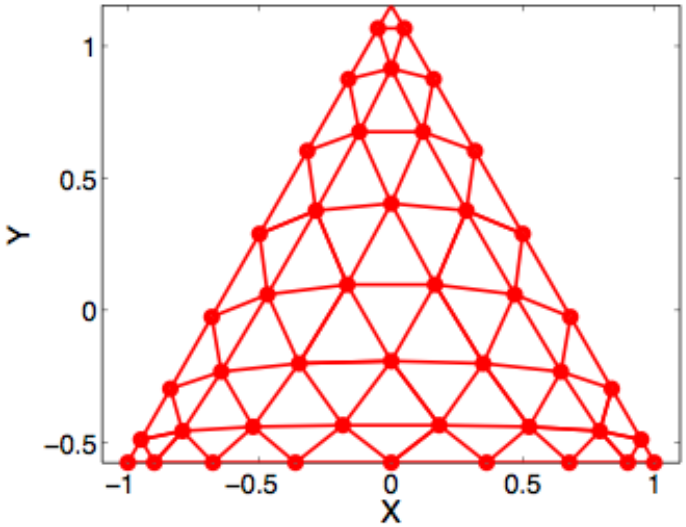
DG Considerations II



Disadvantages:

- Requires very short time step
  - Grid spacing reduced toward element boundaries

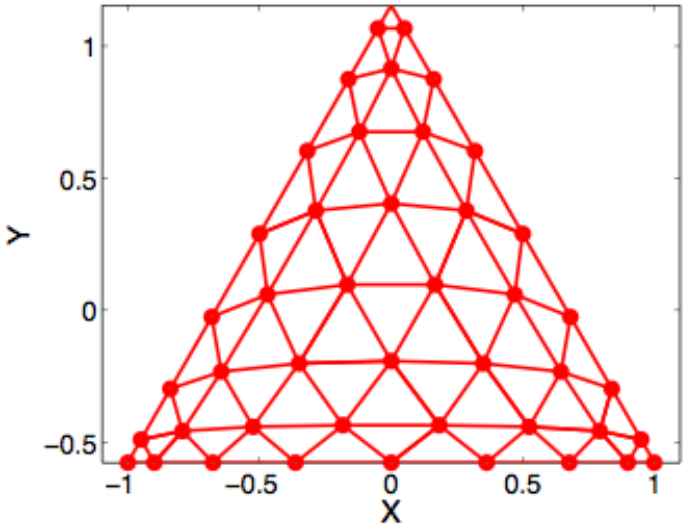
DG Considerations II



Disadvantages:

- Requires very short time step
  - Grid spacing reduced toward element boundaries
- Discontinuities
  - Can be accommodated across element boundaries by limiting the fluxes

DG Considerations II



Disadvantages:

- Requires very short time step
  - Grid spacing reduced toward element boundaries
- Discontinuities
  - Can be accommodated across element boundaries by limiting the fluxes
  - Cannot naturally be accommodated within each element

Eulerian

Accelerations evaluated at fixed points.

Eulerian

Accelerations evaluated at fixed points.

- + Easily adapted to many grid structures

Eulerian

Accelerations evaluated at fixed points.

- + Easily adapted to many grid structures
- - Need to evaluate nonlinear advection terms ($u\partial v/\partial x$).

Eulerian

Accelerations evaluated at fixed points.

- + Easily adapted to many grid structures
- - Need to evaluate nonlinear advection terms ($u\partial v/\partial x$).
- - Impossible to eliminate numerical diffusion in high Reynolds number flow

Lagrangian

Accelerations evaluated along fluid parcel trajectories

Lagrangian

Accelerations evaluated along fluid parcel trajectories

- + Can completely eliminate numerical diffusion

Lagrangian

Accelerations evaluated along fluid parcel trajectories

- + Can completely eliminate numerical diffusion
- + No nonlinear advection terms to evaluate

Lagrangian

Accelerations evaluated along fluid parcel trajectories

- + Can completely eliminate numerical diffusion
- + No nonlinear advection terms to evaluate
- + Some processes (like advection) Doppler shift to lower frequencies (allowing larger time steps).

Lagrangian

Accelerations evaluated along fluid parcel trajectories

- + Can completely eliminate numerical diffusion
- + No nonlinear advection terms to evaluate
- + Some processes (like advection) Doppler shift to lower frequencies (allowing larger time steps).
- - Trajectory for each fluid parcel must be evaluated

Lagrangian

Accelerations evaluated along fluid parcel trajectories

- + Can completely eliminate numerical diffusion
- + No nonlinear advection terms to evaluate
- + Some processes (like advection) Doppler shift to lower frequencies (allowing larger time steps).
- - Trajectory for each fluid parcel must be evaluated
- - Fluid parcels tend to become unevenly distributed.

Lagrangian

Accelerations evaluated along fluid parcel trajectories

- + Can completely eliminate numerical diffusion
- + No nonlinear advection terms to evaluate
- + Some processes (like advection) Doppler shift to lower frequencies (allowing larger time steps).
- - Trajectory for each fluid parcel must be evaluated
- - Fluid parcels tend to become unevenly distributed.
- - Awkward to compute gradients of field variables ($\partial p/\partial x$)

Semi-Lagrangian

Fluid parcels arrive at every node on the specified mesh at the *end* of each time step.

Semi-Lagrangian

Fluid parcels arrive at every node on the specified mesh at the *end* of each time step.

Semi-Lagrangian approximation to the advection equation is

$$\frac{\phi(x_j, t^{n+1}) - \phi(\tilde{x}_j^n, t^n)}{\Delta t} = 0,$$

where $\tilde{x}_j^n$ denotes the departure point of a trajectory originating at time t^n and arriving at (x_j, t^{n+1}).

Semi-Lagrangian II

For constant $U > 0$

$$\tilde{x}_j^n = x_j - U\Delta t.$$

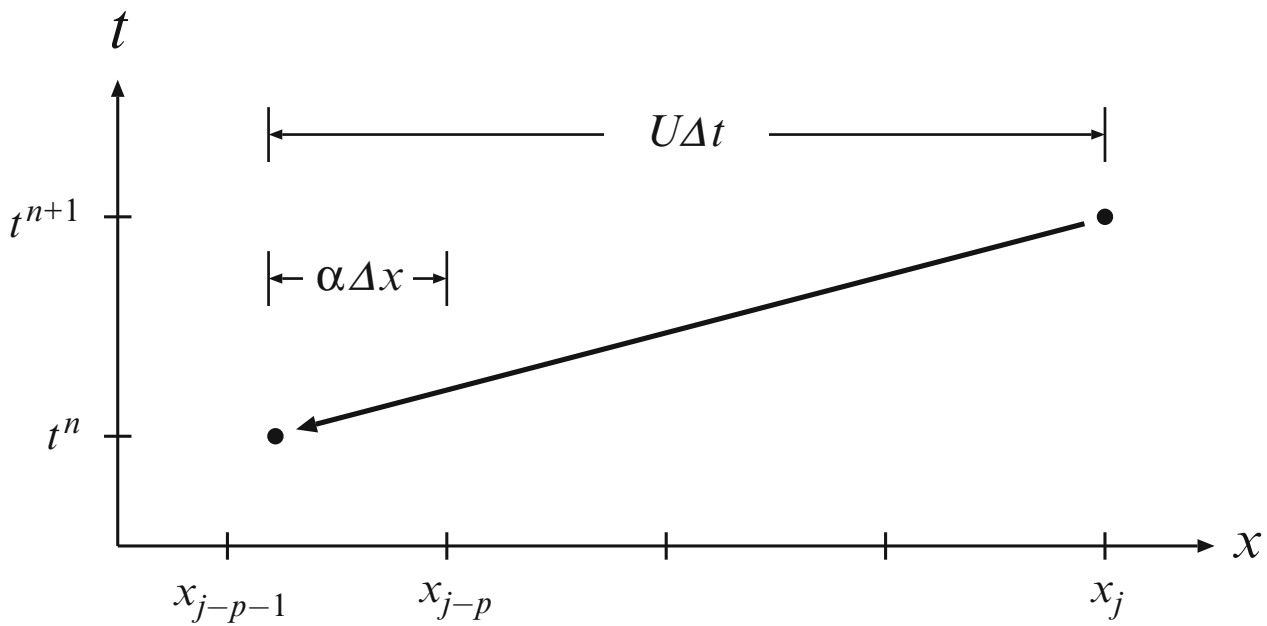
Semi-Lagrangian II

For constant $U > 0$

$$\tilde{x}_j^n = x_j - U\Delta t.$$

Let p be the integer part of $U\Delta t/\Delta x$, then

$$x_{j-p-1} \leq \tilde{x}_j^n < x_{j-p}$$



Semi-Lagrangian III

Interpolate the grid-point values to $\tilde{x}_j^n$.

Semi-Lagrangian III

Interpolate the grid-point values to $\tilde{x}_j^n$.

If $U\Delta t/\Delta x \leq 1$

- Linear interpolation gives upstream scheme
- Quadratic interpolation gives Lax-Wendroff

Semi-Lagrangian III

Interpolate the grid-point values to $\tilde{x}_j^n$.

If $U\Delta t/\Delta x \leq 1$

- Linear interpolation gives upstream scheme
- Quadratic interpolation gives Lax-Wendroff

Interpolation introduces numerical diffusion

- Use at cubic interpolation (cheat in 2 or 3D)
- Take the largest possible time step. What limits Δt?

Semi-Lagrangian III

Interpolate the grid-point values to $\tilde{x}_j^n$.

If $U\Delta t/\Delta x \leq 1$

- Linear interpolation gives upstream scheme
- Quadratic interpolation gives Lax-Wendroff

Interpolation introduces numerical diffusion

- Use at cubic interpolation (cheat in 2 or 3D)
- Take the largest possible time step. What limits Δt?
  - Accuracy of the trajectory calculation

Semi-Lagrangian III

Interpolate the grid-point values to $\tilde{x}_j^n$.

If $U\Delta t/\Delta x \leq 1$

- Linear interpolation gives upstream scheme
- Quadratic interpolation gives Lax-Wendroff

Interpolation introduces numerical diffusion

- Use at cubic interpolation (cheat in 2 or 3D)
- Take the largest possible time step. What limits Δt?
  - Accuracy of the trajectory calculation
  - Frequency of the forcing in the Lagrangian frame

CFL Condition

Time step can be very large when simulating advection of a passive scalar (e.g., $U\Delta t/\Delta x = 4$). Do SL methods avoid the CFL condition?

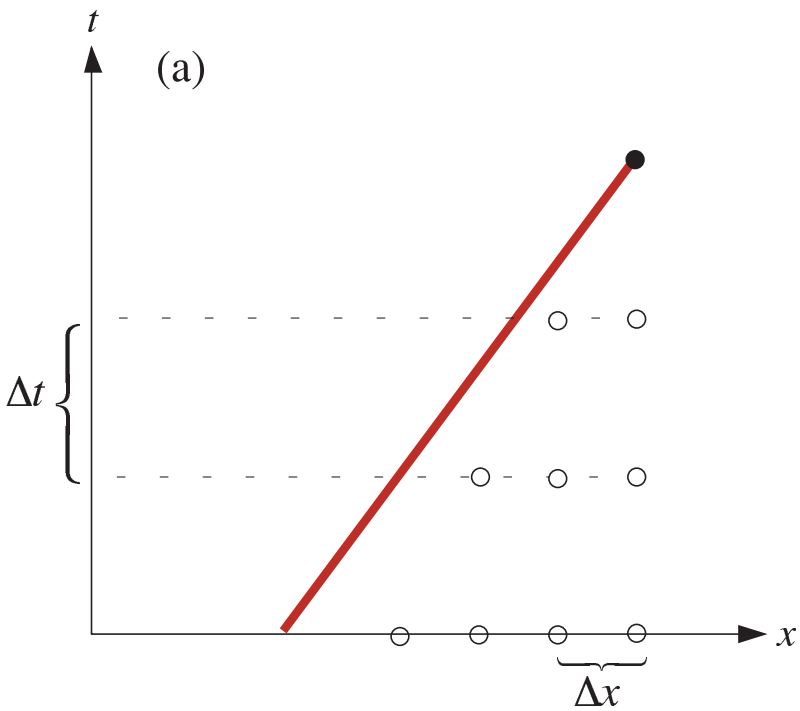
CFL Condition

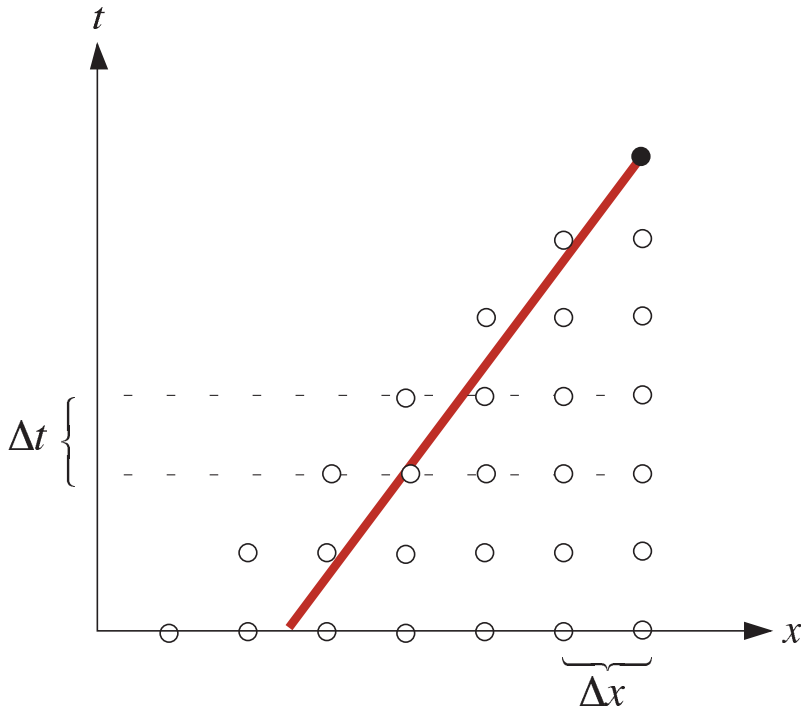
Time step can be very large when simulating advection of a passive scalar (e.g., $U\Delta t/\Delta x = 4$). Do SL methods avoid the CFL condition?

Courant-Freidrichs-Levy Condition: the numerical domain of dependence must include the domain of dependence of the true solution. (Necessary but not sufficient for stability.)

Eulerian CFL

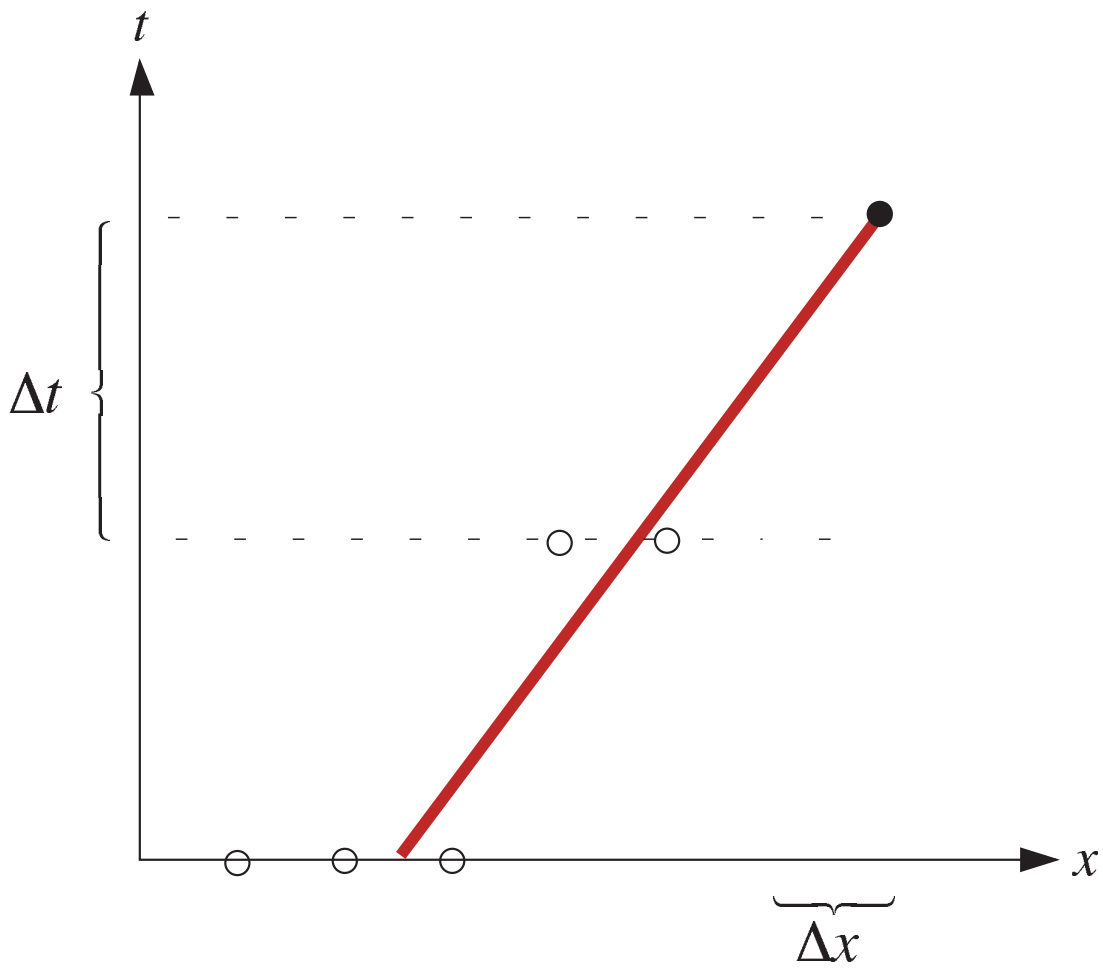
Upstream differencing, constant-windspeed advection





Semi-Lagrangian CFL

Linear interpolation to departure point, constant-windspeed advection



Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

- SL very effective for dealing with the pole problem in global models.

Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

- SL very effective for dealing with the pole problem in global models.
- Large-scale dynamics mostly advection of PV.

Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

- SL very effective for dealing with the pole problem in global models.
- Large-scale dynamics mostly advection of PV.
  - Trajectories *smooth* compared to advected field.

Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

- SL very effective for dealing with the pole problem in global models.
- Large-scale dynamics mostly advection of PV.
  - Trajectories *smooth* compared to advected field.
- Typical CFL number of 4 in global models is determined by jet-stream max.

Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

- SL very effective for dealing with the pole problem in global models.
- Large-scale dynamics mostly advection of PV.
  - Trajectories *smooth* compared to advected field.
- Typical CFL number of 4 in global models is determined by jet-stream max.
  - Max ≈ 160 m s^{-1}; most winds < 40 m s^{-1}, so CFL over most of the domain < 1

Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

- SL very effective for dealing with the pole problem in global models.
- Large-scale dynamics mostly advection of PV.
  - Trajectories *smooth* compared to advected field.
- Typical CFL number of 4 in global models is determined by jet-stream max.
  - Max ≈ 160 m s^{-1}; most winds < 40 m s^{-1}, so CFL over most of the domain < 1
- Vertical advection often associated with limiting CFL in simulations of convection.

Eulerian vs Semi-Lagrangian

Semi-Lagrangian widely used for global-scale models, seldom used on the mesoscale.

- SL very effective for dealing with the pole problem in global models.
- Large-scale dynamics mostly advection of PV.
  - Trajectories *smooth* compared to advected field.
- Typical CFL number of 4 in global models is determined by jet-stream max.
  - Max ≈ 160 m s^{-1}; most winds < 40 m s^{-1}, so CFL over most of the domain < 1
- Vertical advection often associated with limiting CFL in simulations of convection.
  - Trajectory morphology *not* simpler than the advected fields.

Reference

Durran, D.R., 2010: *Numerical Methods for Fluid Dynamics: With Applications to Geophysics. 2nd Ed.* Springer.