Modified-Truncation Finite Difference Schemes

Department Mathematics and Statistics
Arizona State University

Don Jones
dajones@math.asu.edu
Heat Equation

\[ \partial_t u = \lambda u_{xx} + f \]
Heat Equation

\[ \partial_t u = \lambda u_{xx} + f \]

Finite Difference scheme

\[ \partial_t U_i = \frac{\lambda}{\Delta x^2} (U_{i+1} + U_{i-1} - 2U_i) + f_i \]
Heat Equation

\[ \partial_t u = \lambda u_{xx} + f \]

- **Finite Difference scheme**

\[ \partial_t U_i = \frac{\lambda}{\Delta x^2} (U_{i+1} + U_{i-1} - 2U_i) + f_i \]

- **True nodal values**

\[ \partial_t u_i = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i \]

\[ + \lambda \frac{u_{xxxx}(t, i\Delta x)}{12} \Delta x^2 + O(\Delta x^4) \]
Goal

- Modify truncation error
Goal

- Modify truncation error
- Same time step
Goal

- Modify truncation error
- Same time step
- Same spatial stencil
Goal

- Modify truncation error
- Same time step
- Same spatial stencil
- Same properties
Goal

- Modify truncation error
- Same time step
- Same spatial stencil
- Same properties
- **Never** less accurate than original scheme
Related Ideas

- Compact Difference Schemes

\[
\frac{(u_{i+1} + u_{i-1} - 2u_i)}{\Delta x^2} = u_{xx} + \frac{\partial^4 u_i}{12 \Delta x^2} \Delta x^2 + \ldots
\]
Related Ideas

- Compact Difference Schemes

\[
\frac{(u_{i+1} + u_{i-1} - 2u_i)}{\Delta x^2} = u_{xx} + \frac{\partial^4_{xxxx}u_i}{12} \Delta x^2 + \ldots
\]

\[
\delta^2_x u_i = \partial^2_{xx} u_i + \frac{\partial^2_{xx} u_{i+1} + \partial^2_{xx} u_{i-1} - 2\partial^2_{xx} u_i}{12}
\]
Related Ideas

- Compact Difference Schemes

\[
\frac{u_{i+1} + u_{i-1} - 2u_i}{\Delta x^2} = u_{xx} + \frac{\partial^4_{xxxx}u_i}{12}\Delta x^2 + \ldots
\]

\[
\delta_x^2 u_i = \partial_{xx}^2 u_i + \frac{\partial_{xx}^2 u_{i+1} + \partial_{xx}^2 u_{i-1} - 2\partial_{xx}^2 u_i}{12}
\]

\[
\delta_x^2 u_i = \left(\frac{\partial_{xx}^2 u_{i+1} + 10\partial_{xx}^2 u_i + \partial_{xx}^2 u_{i-1}}{12}\right) + O(\Delta x^4)
\]
Related Ideas

- **Compact Difference Schemes**

\[
\frac{(u_{i+1} + u_{i-1} - 2u_i)}{\Delta x^2} = u_{xx} + \frac{\partial^4_{xxxx} u_i}{12} \Delta x^2 + \ldots
\]

\[
\delta^2_x u_i = \partial^2_{xx} u_i + \frac{\partial^2_{xx} u_{i+1} + \partial^2_{xx} u_{i-1} - 2\partial^2_{xx} u_i}{12}
\]

\[
\delta^2_x u_i = \frac{(\partial^2_{xx} u_{i+1} + 10\partial^2_{xx} u_i + \partial^2_{xx} u_{i-1})}{12} + O(\Delta x^4)
\]

- **Well-Balanced Schemes**
Big Idea \( u_t = \lambda u_{xx} + f \)

- True nodal values

\[
\partial_t u_i = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i \\
+ \lambda \frac{u_{xxxx}(t, i\Delta x)}{12} \Delta x^2 + \mathcal{O}(\Delta x^4)
\]
Big Idea $u_t = \lambda u_{xx} + f$

- True nodal values

$$\partial_t u_i = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i$$

$$+ \lambda \frac{u_{xxxx}(t, i\Delta x)}{12} \Delta x^2 + O(\Delta x^4)$$

- At steady state $-\lambda \frac{u_{xxxx}}{12} = \frac{f_{xx}}{12}$
**Big Idea** \( u_t = \lambda u_{xx} + f \)

- **True nodal values**

\[
\partial_t u_i = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i \\
+ \lambda \frac{u_{xxxx}(t, i\Delta x)}{12} \Delta x^2 + O(\Delta x^4)
\]

- **At steady state** 

\(-\lambda \frac{u_{xxxx}}{12} = \frac{f_{xx}}{12}\)

- **The Modified Scheme**

\[
\partial_t u_i = \lambda \delta_x^2 u_i + f_i + \frac{f_{xx}}{12} \Delta x^2 \\
= \lambda \delta_x^2 u_i + \frac{f_{i+1} + 10f_i + f_{i-1}}{12}
\]
Burgers Equation

\[ u_t + uu_x = \lambda u_{xx} + f \]

Standard Scheme

\[ \partial_t u_i + \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x} = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i \]
Burgers Equation

\[ u_t + uu_x = \lambda u_{xx} + f \]

- Standard Scheme

\[ \partial_t u_i + \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x} = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i \]

- At steady state

\[ \frac{uu_{xxx}}{6} + \frac{u_xu_{xx}}{2} = \frac{f_{xx}}{6} \]
**Burgers Equation**

\[ u_t + uu_x = \lambda u_{xx} + f \]

- **Standard Scheme**

\[
\partial_t u_i + \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x} = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i
\]

- **At steady state**

\[
\frac{uu_{xxx}}{6} + \frac{u_xu_{xx}}{2} = \frac{f_{xx}}{6}
\]

- **The Modified Scheme (low Viscosity)**

\[
\partial_t u_i + \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x} = \\
+ \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i) + f_i + \frac{f_{xx}}{6} \Delta x^2
\]
Burgers Equation

\[ u_t + uu_x = \lambda u_{xx} + f \]

- Method Applies to Any Scheme...
  - Modified-Modified Scheme

\[
\partial_t u_i = \frac{\lambda}{\Delta x^2} (u_{i+1} + u_{i-1} - 2u_i)
- \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x}
+ f_i + \sum_{n=0}^{N-1} \frac{\partial_x^2 f}{(2n + 1)!} \Delta x^{2n}
\]
**Burgers Equation**

\[ u_t + uu_x = \lambda u_{xx} + f \]

- Method Applies to Any Scheme...Modified-Modified Scheme

\[
\frac{\partial_t u_i}{\partial t} = \frac{\lambda}{\Delta x^2}(u_{i+1} + u_{i-1} - 2u_i) - \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x} - f_i + \sum_{n=0}^{N-1} \frac{\partial_x^n f}{(2n + 1)!} \Delta x^{2n}
\]

- is order \( \Delta x^{2N} \) near steady state.
Numerical Results

- Exact Solution \[ u = \eta \sin(\pi x) + \epsilon \sin(\pi x) \left( x \cos(\pi t) + (1 - x) \sin(\pi t) \right) \]

- Left: \( \eta = .99, \; \epsilon = .01 \)  
- Right: \( \eta = .01, \; \epsilon = .99 \)
With Diffusion

\[ u_t + Cu_x = \lambda u_{xx} + f \]

- Standard Scheme

\[
\frac{du_i}{dt} + C \frac{u_{i+1} - u_{i-1}}{2\Delta x} = \lambda \delta_x^2 u_i + f_i
\]
With Diffusion

\[ u_t + C u_x = \lambda u_{xx} + f \]

- Standard Scheme

\[
\frac{d u_i}{d t} + C \frac{u_{i+1} - u_{i-1}}{2\Delta x} = \lambda \delta_x^2 u_i + f_i
\]

- At Steady state

\[
\lambda u_{xxxx} = C u_{xxx} - f_{xx} \quad u_{xxx} = \frac{1}{\lambda} (C u_{xx} - f_x)
\]
With Diffusion

\[ u_t + C u_x = \lambda u_{xx} + f \]

- **Standard Scheme**

\[
\frac{d u_i}{d t} + C \frac{u_{i+1} - u_{i-1}}{2\Delta x} = \lambda \delta^2_x u_i + f_i
\]

- **At Steady state**

\[
\lambda u_{xxxx} = C u_{xxx} - f_{xx} \quad u_{xxx} = \frac{1}{\lambda} (C u_{xx} - f_x)
\]

- **Modified Scheme**

\[
\frac{d a_i}{d t} + C \frac{a_{i+1} - a_{i-1}}{2\Delta x} = \left( \lambda + \frac{C^2 \Delta x^2}{12\lambda} \right) \delta^2_x a_i - \left( \frac{\Delta x}{12\lambda} \right) C(f_{i+1} - f_{i-1})
\]

\[
+ \frac{f_{i+1} + 10f_i + f_{i-1}}{12}
\]
Linear Advection

Sign Preserving for all parameter values
With Diffusion

\[ u_t + uu_x = \lambda u_{xx} + f \]

**Standard Scheme**

\[
\frac{du_i}{dt} + \frac{u_{i+1}^2 - u_{i-1}^2 + u_i (u_{i+1} - u_{i-1})}{6\Delta x} = \lambda \delta_x^2 u_i + f_i
\]
With Diffusion
\[ u_t + uu_x = \lambda u_{xx} + f \]

- **Standard Scheme**

\[
\frac{du_i}{dt} + \frac{u_{i+1}^2 - u_{i-1}^2 + u_i(u_{i+1} - u_{i-1})}{6\Delta x} = \lambda \delta_x^2 u_i + f_i
\]

- **Modified Scheme**

\[
\frac{da_i}{dt} + \frac{a_{i+1}^2 - a_{i-1}^2 + a_i(a_{i+1} - a_{i-1})}{6\Delta x} = \delta_x^+ (\lambda_i^e \delta_x^- a_i) - \left( \frac{\Delta x}{48\lambda} \right) (a_{i+1} f_{i+1} - a_{i-1} f_{i-1}) + \frac{f_{i+1} + 10f_i + f_{i-1}}{12}
\]

where

\[
\lambda_i^e := \lambda \left( 1 + \frac{\Delta x^2}{12\lambda^2} \left( \frac{a_i + a_{i-1}}{2} \right)^2 \right)
\]
Random Initial Data

![Graphs showing wave number and energy distributions for Standard and Enslaved methods with different grid sizes.]
Linear Advection

- Constant Velocity

\[ u_t + C \cdot \nabla u = \nu \nabla^2 u + f \]
Linear Advection

- **Constant Velocity**

\[ u_t + C \cdot \nabla u = \nu \nabla^2 u + f \]

- **Standard Scheme**

\[ \frac{du_{i,j}}{dt} + (C_1, C_2) \cdot (\delta_x u, \delta_y u) = \nu (\delta_x^2 + \delta_y^2) u_{i,j} + f_{i,j} \]
Modified Linear Advection

\[
\frac{du_{i,j}}{dt} + (C_1, C_2) \cdot (\delta_x \overline{u}, \delta_y \overline{u}) \\
= \nu (\delta^2_x u_{i,j} + \delta^2_y u_{i,j}) + \frac{\Delta x^2}{12 \nu} (C_1^2 u_{xx} + 2C_1 C_2 u_{xy} + C_2^2 u_{yy}) \\
+ f_{i,j} + \frac{\Delta x^2}{12} \nabla^2 f - \frac{\Delta x^2}{12 \nu} (C \cdot \nabla f)
\]

- \(\overline{u}_{i,j} = \frac{1}{6} (u_{i+1,j} + 4u_{i,j} + u_{i-1,j})\)

- Sign Preserving for all parameter values
Advection Diffusion

Standard Scheme

Modified Scheme
Shallow Water Equations

\[ \begin{align*}
    u_t + (u \cdot \nabla)u + f_\beta k \times u &= \nu D - g \nabla h + F \\
h_t + \nabla \cdot (hu) &= 0
\end{align*} \]
Shallow Water Equations

\[ \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} + f_\beta \mathbf{k} \times \mathbf{u} = \nu D - g \nabla h + \mathbf{F} \]

\[ h_t + \nabla \cdot (h \mathbf{u}) = 0 \]

Standard Scheme - Centered, B Grid, - Leap Frog

\[ \frac{\mathbf{u}^{n+1}_{i,j} - \mathbf{u}^{n-1}_{i,j}}{2\Delta t} = \text{Rhs}^n_{i,j} \]
Meandering Jet
Time Dependence

Maximum of each term in SWE.
Truncation Error

\[
\frac{u_{i,j}(u_{i+1,j} - u_{i-1,j})}{2\Delta x} = uu_x + uu_{xxx} \frac{\Delta x^2}{6} + \ldots
\]
Truncation Error

\[ \frac{u_{i,j}(u_{i+1,j} - u_{i-1,j})}{2\Delta x} = uu_x + uu_{xxx} \frac{\Delta x^2}{6} + \ldots \]

At steady state

\[ \nabla^2(uu_x + vu_y) = \nabla^2(fv - gh_x + Fu) \]
Truncation Error

\[ \frac{u_{i,j}(u_{i+1,j} - u_{i-1,j})}{2\Delta x} = uu_x + uu_{xxx} \frac{\Delta x^2}{6} + \ldots \]

At steady state

\[ \nabla^2 (uu_x + vu_y) = \nabla^2 (fv - gh_x + Fu) \]

or

\[ \frac{1}{6} (uu_{xxx} + vu_{yyy} + gh_{xxx}) = \frac{1}{6} (uu_{xyy} + vu_{xxy} + gh_{xyy}) + \text{LOTS} \]
Kinetic Energy

- Standard Scheme $\Delta x=40\text{km}$
- Modified Scheme $\Delta x=40\text{km}$
- Standard Scheme $\Delta x=13.3\text{km}$

Density

- $\Delta x=40\text{km}$
- $\Delta x=20\text{km}$
- $\Delta x=13.3\text{km}$
Histograms

- $\Delta x = 40 \text{ km}$
- $\Delta x = 20 \text{ km}$
- $\Delta x = 13.3 \text{ km}$
EOFs/PODs

EOF 2, 40km Resolution

EOF 2, 40km Resolution, Modified Scheme

EOF 2, 20km Resolution

EOF 2, 13.3km Resolution
Navier Stokes

- The equations

\[
\frac{\partial u}{\partial t} + (u^2)_x + (vu)_y = \nu \nabla^2 u - p_x + f_u
\]

\[
\frac{\partial v}{\partial t} + (uv)_x + (v^2)_y = \nu \nabla^2 v - p_y + f_v
\]

\[
u_x + \nu_y = 0
\]
Navier Stokes

- The equations

\[
\frac{\partial u}{\partial t} + (u^2)_x + (vu)_y = \nu \nabla^2 u - p_x + f_u \\
\frac{\partial v}{\partial t} + (uv)_x + (v^2)_y = \nu \nabla^2 v - p_y + f_v \\
u_x + v_y = 0
\]

- Harlow, Welch (1965)

\[
\frac{u_{i,j}^{n+1} - u_{i,j}}{\Delta t} = \rho hu_{i,j} - \frac{p_{i+1,j} - p_{i,j}}{\Delta x} \\
\frac{v_{i,j}^{n+1} - v_{i,j}}{\Delta t} = \rho hv_{i,j} - \frac{p_{i,j+1} - p_{i,j}}{\Delta x}
\]
Navier Stokes

The equations

\[
\frac{\partial u}{\partial t} + (u^2)_x + (vu)_y = \nu \nabla^2 u - p_x + f_u
\]

\[
\frac{\partial v}{\partial t} + (uv)_x + (v^2)_y = \nu \nabla^2 v - p_y + f_v
\]

\[
u_x + v_y = 0
\]

Harlow, Welch (1965)

\[
\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \rho u_{i,j} - \frac{p_{i+1,j} - p_{i,j}}{\Delta x}
\]

\[
\frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} = \rho v_{i,j} - \frac{p_{i,j+1} - p_{i,j}}{\Delta x}
\]

Discrete Divergence

\[
0 = D_{i,j}^{n+1} = \frac{D_{i,j}}{\Delta t} + \frac{\rho u_{i,j} - \rho u_{i-1,j}}{\Delta x} + \frac{\rho v_{i,j} - \rho v_{i,j-1}}{\Delta y} - (\delta^2_x + \delta^2_y)p_{i,j}
\]
Modified Scheme

The truncation error

\[ T_u = \left( \frac{(u_x^2)x}{12} + \frac{(uu_{xx})x}{3} + \frac{(uv_{xx})y}{8} + \frac{p_{xx}}{24} \right) \Delta x^2 \]

\[ + \left( \frac{(u_{yy}v)y}{6} + \frac{(u_yv_y)y}{12} + \frac{(uv_{yy})y}{24} \right) \Delta y^2 \]
Modified Scheme

The truncation error

\[ T_u = \left( \frac{(u_x^2)_x}{12} + \frac{(uu_{xx})_x}{3} + \frac{(uv_{xx})_y}{8} + \frac{p_{xxx}}{24} \right) \Delta x^2 + \left( \frac{(u_{yy}v)_y}{6} + \frac{(u_{yv})_y}{12} + \frac{(uv_{yy})_y}{24} \right) \Delta y^2 \]

At Steady State

\[ \frac{(u_x^2)_x}{12} + \frac{p_{xxx}}{24} = -\frac{(uu_{xx})_x}{12} - \frac{1}{24} \left( u_{xx}v + 2u_x v_x + uv_{xx} \right)_y + \frac{f_{xx}}{24} \]
Modified Scheme

- The truncation error

\[ T_u = \left( \frac{(u_x^2)x}{12} + \frac{(u u_{xx})x}{3} + \frac{(u v_{xx})y}{8} + \frac{p_{xxx}}{24} \right) \Delta x^2 + \left( \frac{(u y_{yy}v)_y}{6} + \frac{(u y v_y)_y}{12} + \frac{(u v y y)_y}{24} \right) \Delta y^2 \]

- At Steady State

\[ \frac{(u_x^2)x}{12} + \frac{p_{xxx}}{24} = -\frac{(u u_{xx})x}{12} - \frac{1}{24} \left( u_{xx}v + 2u_x v_x + u v_{xx} \right)_y + \frac{f_{xx}^u}{24} \]

- Possion Solver Unchanged!
Test Case

\[ u(x, y, t) = -\eta \pi x \sin^2(\pi x) \sin(2\pi y) - \epsilon \pi \cos(\omega t) \sin^2(\pi x) \sin(2\pi y) \]

\[ v(x, y, t) = \eta \sin^2(\pi y) (\sin^2(\pi x) + x\pi \sin(2\pi x)) + \epsilon \pi \cos(\omega t) \sin^2(\pi y) \sin(2\pi x) \]

\[ p(x, y, t) = \cos(\pi xy) \]

Truncation Error Vs Number of Cells
Left: $\nu = .00025$, $\Delta x = \Delta y = 1/50$, $\omega = \pi$, $\eta = .5$, $\epsilon = .5$

Right: $\nu = .005$, $\Delta x = \Delta y = 1/80$, $\omega = \pi$, $\eta = .01$, $\epsilon = .99$
Sea Breeze Problem

\[ u_t + uu_x + wu_z - fv = -p_x + \nu(u_{xx} + u_{zz}) \]
\[ v_t + uv_x + wv_z + fu = \nu(v_{xx} + v_{zz}) \]
\[ w_t + uw_x + ww_z = -p_z + \nu(w_{xx} + w_{zz}) + g\beta\theta \]
\[ \theta_t + u\theta_x + w\theta_z = \lambda(\theta_{xx} + \theta_{zz}) \]
\[ u_x + w_z = 0 \]
Sea Breeze Problem
Wind Averages

Average wind speed for $0 \leq z \leq 320m$ at the shore for various resolutions.
Wind Averages

Average wind speed on right half of the domain:
\[ 0 \leq x \leq 25\text{km}, \quad 0 \leq z \leq 4000\text{m} \]
Conclusions

- Simple Idea - Not trivial to implement
Conclusions

- Simple Idea - Not trivial to implement
- Highly specific to model and differencing
Conclusions

- Simple Idea - Not trivial to implement
- Highly specific to model and differencing
- Effective way to increase the order if above met