Scalable and Efficient Multi-Moment ADER Methods For Atmospheric Dynamical Cores

Matthew R. Norman
Scientific Computing Group
Oak Ridge National Laboratory

PDEs On The Sphere 2012
Computing Has Changed And Is Changing

• Power consumption now the most constraining aspect of HPC
• Better hardware power efficiency makes your life harder
  – More raw computing, relatively less fast cache available
  – Data movement extremely slow
• Significant computation is required to justify a data access
  – Efficient data accesses, do as much as you can with each access
  – Avoid / sparsen data movement even at the cost of more computation
Computing Has Changed And Is Changing

• Power consumption now the most constraining aspect of HPC
• Better hardware power efficiency makes your life harder
  – More raw computing, relatively less fast cache available
  – Data movement extremely slow
• Significant computation is required to justify a data access
  – Efficient data accesses, do as much as you can with each access
  – Avoid / sparsen data movement even at the cost of more computation
• BUT, traditional efficiency constraints still matter
  – Small time step = More MPI communication + More wall time
  – Cost per time step: Core-hour allocations should be reasonable
  – Fastest path to an unusable solution does not give good science
  – Inappropriate algorithms achieving peak flops are still inappropriate
Some Options For A Numerical Method

- Moments: Define the moments to evolve
- Grid: Arrange where those moments live
- Recovery: Coordinate moments to describe fluid state
- Evolution: Formulate constraints to evolve moments
- Approximation: Practically satisfy the evolution constraints
- Time Integrator: Step forward to predict future moments
- Limiting: Control oscillations in the solution
Some Options For A Numerical Method

- Moments: Define the moments to evolve
- Grid: Arrange where those moments live
- Recovery: Coordinate moments to describe fluid state
- Evolution: Formulate constraints to evolve moments
- Approximation: Practically satisfy the evolution constraints
- Time Integrator: Step forward to predict future moments
- Limiting: Control oscillations in the solution
What Does “Multi-Moment” Look Like?

- Elements / control volumes with multiple discrete data

Single Moment - Stencil

Multi-Moment - Local
What Does “Multi-Moment” Look Like?

- Elements / control volumes with multiple discrete data

**Single Moment - Stencil**

All moments contribute to the recovery

Only **one** moment uses the recovery to evolve

**Multi-Moment - Local**

All moments contribute to the recovery

**All** moments use the recovery to evolve.
A Few Types Of “Multi-Moment”

Nodal Finite-Volume

\[ \sum \frac{a_{i,j}}{|\Omega|} \int_{\Omega} \frac{\partial^{i+j} U}{\partial x^i \partial y^j} d\Omega \]

Modal Finite-Volume

Nodal Basis

Modal Basis
Evolution Constraints For Moments

• Evolution: What you do to the PDEs

• Examples:
  – Do nothing (strong): Finite-Difference
  – Integrate spatially (weak): Finite-Volume
  – Apply test function & integrate: Variational / Galerkin
  – Differentiate & integrate spatially: Multi-Moment Finite-Volume
  – Apply a transform: Global Spectral
  – Change of reference frame: [Semi - ] Lagrangian

• Evolution largely informs time step & accuracy
**ADER: Arbitrary-order DERivative Riemann**

- **Input:** spatial derivatives  → **Output:** space-time derivatives

- Fully-discrete: Only one communication stage per time step
- Full multi-dimensional, non-linear coupling over a time step
- Arbitrarily high-order-accurate, fully couples all terms involved

**Exploring A Scalable Time Stepping Option**
**Exploring A Scalable Time Stepping Option**

- **ADER: Arbitrary-order DErivative Riemann**
  - **Input:** spatial derivatives  
  - **Output:** space-time derivatives
  - Fully-discrete: Only one communication stage per time step
  - Full multi-dimensional, non-linear coupling over a time step
  - Arbitrarily high-order-accurate, fully couples all terms involved

- **Example (Ye Olde Fashioned Way)**
  
  \[
  \frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U)
  \]
• ADER: Arbitrary-order DErivative Riemann  
  \textbf{Input: spatial derivatives}  \rightarrow \textbf{Output: space-time derivatives}  
  – Fully-discrete: Only one communication stage per time step  
  – Full multi-dimensional, non-linear coupling over a time step  
  – Arbitrarily high-order-accurate, fully couples all terms involved  

• Example (Ye Olde Fashioned Way)  

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U) 
\]

\[
\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial U} \frac{\partial U}{\partial x} + S(U) 
\]
Exploring A Scalable Time Stepping Option

- ADER: Arbitrary-order DErivative Riemann
  Input: spatial derivatives  Output: space-time derivatives
  - Fully-discrete: Only one communication stage per time step
  - Full multi-dimensional, non-linear coupling over a time step
  - Arbitrarily high-order-accurate, fully couples all terms involved

- Example (Ye Olde Fashioned Way)

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U)
\]
\[
\frac{\partial^2 U}{\partial x \partial t} = -\frac{\partial}{\partial x} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial x} S(U)
\]
Exploring A Scalable Time Stepping Option

- ADER: Arbitrary-order DDerivative Riemann
  - Fully-discrete: Only one communication stage per time step
  - Full multi-dimensional, non-linear coupling over a time step
  - Arbitrarily high-order-accurate, fully couples all terms involved

Example (Ye Olde Fashioned Way)

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U)
\]

\[
\frac{\partial^2 U}{\partial x \partial t} = -\frac{\partial}{\partial x} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial x} S(U)
\]

\[
\frac{\partial^2 U}{\partial t^2} = -\frac{\partial}{\partial t} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial t} S(U)
\]
• ADER: Arbitrary-order D Errivative Riemann
  Input: spatial derivatives  \rightarrow \text{Output: space-time derivatives}
  – Fully-discrete: Only one communication stage per time step
  – Full multi-dimensional, non-linear coupling over a time step
  – Arbitrarily high-order-accurate, fully couples all terms involved
• Example (Ye Olde Fashioned Way)

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U)
\]

\[
\frac{\partial^2 U}{\partial x \partial t} = -\frac{\partial}{\partial x} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial x} S(U)
\]

\[
\frac{\partial^2 U}{\partial t^2} = -\frac{\partial}{\partial t} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial t} S(U)
\]

\[
\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial U} \frac{\partial U}{\partial x} + S(U)
\]

\[
\frac{\partial^2 U}{\partial x \partial t} = -\frac{\partial F}{\partial U} \frac{\partial^2 U}{\partial x^2} - \frac{\partial^2 F}{\partial U^2} \left( \frac{\partial U}{\partial x} \right)^2 + \frac{\partial S}{\partial U} \frac{\partial U}{\partial x}
\]

\[
\frac{\partial^2 U}{\partial t^2} = -\frac{\partial F}{\partial U} \frac{\partial^2 U}{\partial x \partial t} - \frac{\partial^2 F}{\partial U^2} \left( \frac{\partial U}{\partial t} \right)^2 + \frac{\partial S}{\partial U} \frac{\partial U}{\partial t}
\]
**Exploring A Scalable Time Stepping Option**

- **ADER: Arbitary-order DERivative Riemann**
  - **Input:** spatial derivatives  
  - **Output:** space-time derivatives
  - Fully-discrete: Only one communication stage per time step
  - Full multi-dimensional, non-linear coupling over a time step
  - Arbitrarily high-order-accurate, fully couples all terms involved

- **Example (Ye Olde Fashioned Way)**

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U)
\]

\[
\frac{\partial^2 U}{\partial x \partial t} = - \frac{\partial}{\partial x} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial x} S(U)
\]

\[
\frac{\partial^2 U}{\partial t^2} = - \frac{\partial}{\partial t} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial t} S(U)
\]

\[
\frac{\partial U}{\partial t} = - \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} + S(U)
\]

\[
\frac{\partial^2 U}{\partial x \partial t} = - \frac{\partial F}{\partial U} \frac{\partial^2 U}{\partial U^2} \frac{\partial U}{\partial x} - \frac{\partial^2 F}{\partial U^2} \left( \frac{\partial U}{\partial x} \right)^2 + \frac{\partial S}{\partial U} \frac{\partial U}{\partial x}
\]

\[
\frac{\partial^2 U}{\partial t^2} = - \frac{\partial F}{\partial U} \frac{\partial^2 U}{\partial U^2} \frac{\partial U}{\partial t} - \frac{\partial^2 F}{\partial U^2} \left( \frac{\partial U}{\partial t} \right)^2 + \frac{\partial S}{\partial U} \frac{\partial U}{\partial t}
\]
Exploring A Scalable Time Stepping Option

• ADER: Arbitrary-order DERivative Riemann
  Input: spatial derivatives → Output: space-time derivatives
  – Fully-discrete: Only one communication stage per time step
  – Full multi-dimensional, non-linear coupling over a time step
  – Arbitrarily high-order-accurate, fully couples all terms involved

• Example (Ye Olde Fashioned Way)

\[
\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U)
\]

\[
\frac{\partial^2 U}{\partial x \partial t} = -\frac{\partial}{\partial x} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial x} S(U)
\]

\[
\frac{\partial^2 U}{\partial t^2} = -\frac{\partial}{\partial t} \left( \frac{\partial F}{\partial U} \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial t} S(U)
\]

• Use space-time derivatives however you want
Exploring A Scalable Time Stepping Option

- **ADER: Arbitrary-order DDerivative Riemann**
  
  **Input:** spatial derivatives  ➔ **Output:** space-time derivatives
  - Fully-discrete: Only one communication stage per time step
  - Full multi-dimensional, non-linear coupling over a time step
  - Arbitrarily high-order-accurate, fully couples all terms involved

- **Example (Ye Olde Fashioned Way)**

\[
\begin{align*}
  \frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} &= 0 \\
  \frac{\partial^2 U}{\partial x \partial t} &= -\frac{\partial}{\partial x} \left( \frac{\partial F(U)}{\partial x} \right) + \frac{\partial}{\partial x} S(U) \\
  \frac{\partial^2 U}{\partial t^2} &= -\frac{\partial}{\partial t} \left( \frac{\partial F(U)}{\partial x} \right) + \frac{\partial}{\partial t} S(U)
\end{align*}
\]

Typically done in Maple / Mathematica

Not the most efficient way to do this

- Use space-time derivatives however you want
Exploring A Scalable Time Stepping Option

• A Better ADER Procedure (Norman & Finkel 2012, JCP)
  – Differential Transforms: Transforms PDE into a recurrence relation
  \[
  \frac{\partial q(x,t)}{\partial t} = -\frac{1}{2} \frac{\partial q(x,t)^2}{\partial x}
  \]
  \[
  Q(k,h+1) = -\frac{1}{2} \frac{k+1}{h+1} \sum_{r=0}^{k+1} \sum_{s=0}^{h} Q(r,s)Q(k+1-r,h-s)
  \]
  – Result is Taylor expansion
  – Taylor expansion closes any spatial operator or mesh (very flexible)
  – Quadrature-free possible: All PDE terms expanded as polynomials
  – AMR local time stepping simplified (Yulong Xing, ORNL)
  – Limiting the input spatial derivatives also limits the temporal evolution
    • But does not (yet?) guarantee maintained monotonicity
Some interesting time stepping properties base on evolution

- “p” is the number of moments per cell / element
- Nodal SSP-RK (fixed) Galerkin: \[ \Delta t \propto \Delta x^{-1} p^{-1.7} \]
Exploring Scalable Evolution Operators

• Some interesting time stepping properties base on evolution
  – “p” is the number of moments per cell / element
  – Nodal SSP-RK (fixed) Galerkin: \( \Delta t \propto \Delta x^{-1} p^{-1.7} \)
  – ADER Spectral Volume: Depends on “\( \mu \)” \( \Delta t \propto \Delta x^{-1} p^{-(1+0.3\mu^{1.25})} \)
  • Larger \( \mu \) means more grid clustering near edges
  • Smaller \( \mu \) also means more oscillatory reconstructions
  • Literature standard is \( \mu=1.6 \): \( \Delta t \propto \Delta x^{-1} p^{-1.5} \)
  • Yet \( \mu=1.0 \) is also well bounded: \( \Delta t \propto \Delta x^{-1} p^{-1.3} \)
Exploring Scalable Evolution Operators

- Some interesting time stepping properties base on evolution
  - “p” is the number of moments per cell / element
  - Nodal SSP-RK (fixed) Galerkin: \( \Delta t \propto \Delta x^{-1} p^{-1.7} \)
  - ADER Spectral Volume: Depends on “\( \mu \)” \( \Delta t \propto \Delta x^{-1} p^{-\left(1+0.3\mu^{1.25}\right)} \)
    - Larger \( \mu \) means more grid clustering near edges
    - Smaller \( \mu \) also means more oscillatory reconstructions
    - Literature standard is \( \mu=1.6 \): \( \Delta t \propto \Delta x^{-1} p^{-1.5} \)
    - Yet \( \mu=1.0 \) is also well bounded: \( \Delta t \propto \Delta x^{-1} p^{-1.3} \)
  - MM-FV: \( \Delta t \propto \Delta x^{-1} p^{0} \)
    - No time step reduction with p-refinement
Some interesting time stepping properties base on evolution

- “p” is the number of moments per cell / element
- Nodal SSP-RK (fixed) Galerkin: \( \Delta t \propto \Delta x^{-1} p^{-1.7} \)
- ADER Spectral Volume: Depends on “\( \mu \)” \( \Delta t \propto \Delta x^{-1} p^{-(1+0.3\mu^{1.25})} \)
  - Larger \( \mu \) means more grid clustering near edges
  - Smaller \( \mu \) also means more oscillatory reconstructions
  - Literature standard is \( \mu=1.6 \): \( \Delta t \propto \Delta x^{-1} p^{-1.5} \)
  - Yet \( \mu=1.0 \) is also well bounded: \( \Delta t \propto \Delta x^{-1} p^{-1.3} \)
- MM-FV: \( \Delta t \propto \Delta x^{-1} p^0 \)
  - No time step reduction with \( p \)-refinement
Some interesting time stepping properties base on evolution

- “p” is the number of moments per cell / element
- Nodal SSP-RK (fixed) Galerkin: \( \Delta t \propto \Delta x^{-1} p^{-1.7} \)
- ADER Spectral Volume: Depends on “μ” \( \Delta t \propto \Delta x^{-1} p^{-\left(1+0.3\mu^{1.25}\right)} \)
  - Larger μ means more grid clustering near edges
  - Smaller μ also means more oscillatory reconstructions
  - Literature standard is μ=1.6: \( \Delta t \propto \Delta x^{-1} p^{-1.5} \)
  - Yet μ=1.0 is also well bounded: \( \Delta t \propto \Delta x^{-1} p^{-1.3} \)
- MM-FV: \( \Delta t \propto \Delta x^{-1} p^0 \)
  - No time step reduction with p-refinement
- Each p-converges exponentially but not at the same rate
Multi-Moment Finite-Volume Methods

- Store mean state vector derivatives
  \[ \frac{1}{|\Omega|} \int_{\Omega} \frac{\partial i U}{\partial x^i} d\Omega \quad i = 1, \ldots, N - 1 \]

- Evolve with FV treatment of PDE derivatives
  \[
  \frac{\partial}{\partial t} \frac{\partial i U}{\partial x^i} + \frac{1}{\Delta x} \left( \frac{\partial i F(U)}{\partial x^i} \bigg|_R - \frac{\partial i F(U)}{\partial x^i} \bigg|_L \right) = \int_{\Omega} \frac{\partial i S(U)}{\partial x^i} d\Omega \quad i = 1, \ldots, N - 1
  \]

- Do ADER procedure at time step beginning & cell center
  - State, flux & source components are all polynomial expansions

- Sample, integrate, & differentiate PDE terms to close scheme
  - These are all dot products and matrix-vector products

- Use flux-vector version of f-waves Riemann solver for fluxes
  - Allows direct use of time-averaged interface fluxes
  - Only one Riemann solve for all flux derivatives per time step
Accuracy vs Runtime (1-D SW Model)

L1 Relative Error vs Serial Runtime During p-Refinement

Closer to point of origin (lower left) is better

- On my mac
- Intel 12 compiler –fast
- Double precision
- Used PAPI hooks
- Tried to optimize each method
MPI Communication Comparison

Stages of Communication per Length of Simulation
Using Linear Analysis Maximum Stable Time Step, Normalized by ADER+MM-FV
MPI Communication Comparison

Stages of Communication per Length of Simulation
Using Linear Analysis Maximum Stable Time Step, Normalized by ADER+MM-FV

- ADER + DG-N
- ADER + SV1.0
- SSP-RK4 + DG-N
- ADER + MM-FV

Normalized stages of communication:
- 4
- 6
- 8
- 12
- 16
MPI Communication Comparison

Stages of Communication per Length of Simulation
Using Linear Analysis Maximum Stable Time Step, Normalized by ADER+MM-FV

In fairness, MM-FV communicates more data per stage than nodal DG
Still, latency is not negligible
ADER+MM-FV Limiting: [H]WENO or FCT

- Shallow-Water shock solution (Hermite WENO plotted below)
- Flux-Corrected Transport (FCT) is also an option
ADER+MM-FV Limiting: [H]WENO or FCT

- Shallow-Water shock solution (Hermite WENO plotted below)
- Flux-Corrected Transport (FCT) is also an option
Conclusions

• Unexplored moments and evolution formulations may be better suited for current and future massively parallel computers.

• ADER is a non-linearly coupled, highly accurate way to cluster computation and avoid communication – now much cheaper.

• Spectral Volume can use a larger time step than Galerkin depending upon mesh clustering near element boundaries.

• Multi-Moment Finite-Volume could prove very useful in communication avoidance, especially at high-order.

• We need to look at limiting more carefully to limit very high-order while preserving as much variation as possible.

• Computational profile in multiple spatial dimensions should be interesting to see.
Questions?
**Time Integration Options**

- Use the time derivative and evolution to get future moments
- Semi-discrete options (ODE solvers)
  - Multi-step: Use past time levels to predict future ones
  - Multi-stage: Advance forward iteratively in stages
- Fully-discrete options (Integrate directly in time)
  - Semi-Lagrangian: Trajectories to track over time
  - In-Place: Get temporal variation from spatial variation without tracking
- Time-implicit options
  - Future determined by the future, throw it all on LHS, linear solves
  - Must be preconditioned, need large problem size, \( \Delta t \) still not constant