Efficient and accurate solvers for nonhydrostatic simulations of the atmosphere: the interaction of high-order IMEX methods and customized algebraic solvers

Daniel Reynolds[1], *David Gardner*[2], *Chris Vogl*[2], Andrew Steyer[3], Paul Ullrich[4] & Carol Woodward[2]

reynolds@smu.edu, gardner48@llnl.gov, vogl2@llnl.gov

asteyer@sandia.gov, paullrich@ucdavis.edu, cswoodward@llnl.gov

[1]Department of Mathematics, Southern Methodist University
[2]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
[3]Sandia National Laboratories
[4]Department of Land, Air and Water Resources, University of California at Davis

SIAM Conference on Mathematical and Computational Issues in the Geosciences
Houston, Texas
13 March 2019

Outline

Outline

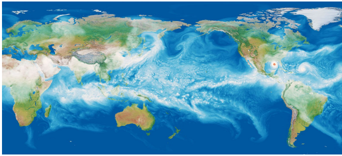
Energy Exascale Earth System Model (E3SM)

Motivation: 2013 DOE report on energy sector factors

- trends in air and water temperatures
- water availability
- storms and heavy precipitation
- coastal flooding and sea-level rise

Mission (`https://e3sm.org/about/vision-and-mission`)

- develop ensemble strategies for uncertainty quantification
- bridge the gap in scales and processes in existing E3SM ventures
- **integrate ECP advances to push model resolution capability**



`https://e3sm.org`

SMU

UCDAVIS

Non-hydrostatic Atmospheric Models

- Increased computational power is pushing climate model resolutions beyond the hydrostatic limit.
- Non-hydrostatic models consider the compressible Navier Stokes equations, that support acoustic (sound) waves.
- Acoustic waves have a negligible effect on climate.

- Acoustic waves travel much faster than convection (343 m/s vs 100 m/s horizontal and 1 m/s vertical).
- To overcome this stiffness, non-hydrostatic models utilize split-explicit, implicit-explicit, or fully implicit time integration.

Non-hydrostatic Formulation (Tempest)

Tempest is an experimental "dycore" used for method development; it considers 5 governing [hyperbolic] equations in an arbitrary coordinate system:

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\frac{1}{J}\frac{\partial}{\partial \alpha}\left(J\rho u^{\alpha}\right) - \frac{1}{J}\frac{\partial}{\partial \beta}\left(J\rho u^{\beta}\right) - \frac{1}{J}\frac{\partial}{\partial \xi}\left(J\rho u^{\xi}\right) \\
\frac{\partial u_\alpha}{\partial t} &= -\frac{\partial}{\partial \alpha}\left(K+\Phi\right) - \theta\frac{\partial \Pi}{\partial \alpha} + \left(\eta \times \mathbf{u}\right)_\alpha \\
\frac{\partial u_\beta}{\partial t} &= -\frac{\partial}{\partial \beta}\left(K+\Phi\right) - \theta\frac{\partial \Pi}{\partial \beta} + \left(\eta \times \mathbf{u}\right)_\beta \\
\left(\frac{\partial r}{\partial \xi}\right)\frac{\partial w}{\partial t} &= -\frac{\partial}{\partial \xi}\left(K+\Phi\right) - \theta\frac{\partial \Pi}{\partial \xi} + u^{\alpha}\frac{\partial u_\alpha}{\partial \xi} + u^{\beta}\frac{\partial u_\beta}{\partial \xi} - u^{\alpha}\frac{\partial u_\xi}{\partial \alpha} - u^{\beta}\frac{\partial u_\xi}{\partial \beta} \\
\frac{\partial \theta}{\partial t} &= -u^{\alpha}\frac{\partial \theta}{\partial \alpha} - u^{\beta}\frac{\partial \theta}{\partial \beta} - u^{\xi}\frac{\partial \theta}{\partial \xi},
\end{aligned}
$$

where ρ is the density, (u_α, u_β) are the horizontal velocity, w is the vertical velocity, and θ is the potential temperature.

Key: horizontal propagation and vertical propagation.

Non-hydrostatic Formulation (HOMME-NH)

HOMME-NH will be the "production" dycore in E3SM v2 responsible for global atmospheric flow (again, 5 hyperbolic equations):

$$\frac{\partial}{\partial t}\left(\frac{\partial \pi}{\partial \eta}\right) = -\nabla_\eta \cdot \left(\frac{\partial \pi}{\partial \eta}\mathbf{u}\right) - \frac{\partial}{\partial \eta}\left(\pi \frac{\mathrm{d}\eta}{\mathrm{d}t}\right)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\nabla_\eta \times \mathbf{u} + 2\boldsymbol{\Omega}) \times \mathbf{u} - \frac{1}{2}\nabla_\eta(\mathbf{u}\cdot\mathbf{u}) - \frac{\mathrm{d}\eta}{\mathrm{d}t}\frac{\partial \mathbf{u}}{\partial \eta} - \frac{1}{\rho}\nabla_\eta p$$

$$\frac{\partial w}{\partial t} = -\mathbf{u}\cdot\nabla_\eta w - \frac{\mathrm{d}\eta}{\mathrm{d}t}\frac{\partial w}{\partial \eta} - g(1-\mu), \quad \mu = \left(\frac{\partial p}{\partial \eta}\right) / \left(\frac{\partial \pi}{\partial \eta}\right),$$

$$\frac{\partial \Theta}{\partial t} = -\nabla_\eta \cdot (\Theta \mathbf{u}) - \frac{\partial}{\partial \eta}\left(\Theta \frac{\mathrm{d}\eta}{\mathrm{d}t}\right), \quad \Theta = \frac{\partial \pi}{\partial \eta}\theta,$$

$$\frac{\partial \phi}{\partial t} = -\mathbf{u}\cdot\nabla_\eta \phi - \frac{\mathrm{d}\eta}{\mathrm{d}t}\frac{\partial \phi}{\partial \eta} + gw,$$

where π is hydrostatic pressure, η is vertical coordinate, $\mathbf{u}$ and w are horizontal and vertical velocities, θ is potential temperature, and ϕ is geopotential.

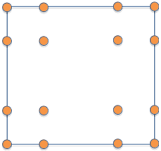
Key: hydrostatic model and nonhydrostatic terms.

Outline

Spatial Semi-discretization

Both codes use spectral elements to discretize horizontally

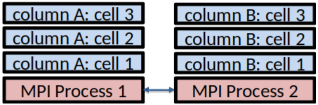
- Lagrange polynomials basis $\{\varphi_j\}$ over GLL points
- Inner product is defined by GLL quadrature:
  $\langle \varphi_j \, , \, \varphi_k \rangle = \int \varphi_j \, \varphi_k \, \mathrm{d}\mathbf{x} = w_j \, \delta_{jk}$



Vertical discretizations differ; both utilize shallow cells (100:1 aspect ratio):

- Tempest uses an up-to-$\mathcal{O}\left(\Delta\xi^5\right)$ staggered nodal finite element method
- HOMME-NH uses $\mathcal{O}\left(\Delta\eta^2\right)$ mimetic finite differences

2D parallel domain decomposition stores entire vertical column(s) on a single MPI task.



In both models, Jacobians of discretized RHS have purely imaginary eigenvalues.

Both models explicitly apply "hyperviscosity" between time steps to stabilize discretization [Ullrich et al., JCP, 2018].

Additive Runge–Kutta (ARK) Methods [Ascher et al. 1997; Araújo et al. 1997; . . .]

Both codes use "ARKode" for time integration, that supports up to two split components: *explicit* and *implicit*,

$$\dot{y} = f^E(t,y) + f^I(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0,$$

- $f^E(t,y)$ contains the explicit terms,
- $f^I(t,y)$ contains the implicit terms.

Combine two s-stage RK methods; denoting $t^*_{n,j} = t_n + c^*_j \Delta t_n$ and $\Delta t_n = t_{n+1} - t_n$:

$$z_i = y_n + \Delta t_n \sum_{j=1}^{i-1} A^E_{i,j} f^E(t^E_{n,j}, z_j) + \Delta t_n \sum_{j=1}^{i} A^I_{i,j} f^I(t^I_{n,j}, z_j), \quad i = 1, \ldots, s,$$

$$y_{n+1} = y_n + \Delta t_n \sum_{j=1}^{s} \left[b^E_j f^E(t^E_{n,j}, z_j) + b^I_j f^I(t^I_{n,j}, z_j) \right]$$

Solving each stage z_i, $i = 1, \dots, s$

Each stage is implicitly defined via a root-finding problem:

$$0 = G_i(z)$$

$$= \left[z - \Delta t_n A^I_{i,i} f^I(t^I_{n,i}, z)\right] - \left[y_n + \Delta t_n \sum_{j=1}^{i-1} \left(A^E_{i,j} f^E(t^E_{n,j}, z_j) + A^I_{i,j} f^I(t^I_{n,j}, z_j)\right)\right]$$

- if $f^I(t, y)$ is *linear* in y then we must solve a linear system for each z_i,
- else G_i is nonlinear, requiring an iterative solver – ARKode options relevant to this work:
  - *Newton*: inexact or *standard* (depends on linear solver),
    - Scaled, preconditioned, GMRES; "matrix-free" available.
    - *user-supplied* to exploit linear system structure
  - user-supplied (new feature).

Customized Vectors and Splittings

Unlike some frameworks that force data structures or algorithms on users, SUNDIALS/ARKode easily leverages problem-specific implementations.

Vectors: both Tempest and HOMME-NH pre-allocate all vectors objects to be used throughout a simulation at initialization.

- "Taught" ARKode how to perform vector arithmetic directly on these structures.
- Requested a sufficient number of preallocated vectors for ARKode temporaries.
- Implemented a system for ARKode to "check out" and "check in" these temporary vectors, in lieu of standard allocation/deallocation.

IMEX/HEVI Splittings:

- Repurposed existing physics routines to provide the IMEX splitting(s) f^E and f^I.
- Tempest also included preprocessor directives to explore alternate splittings by moving terms between f^E and f^I.

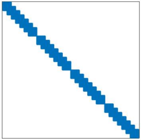
Customized Algebraic Solvers

Since HEVI splittings only include implicit coupling within each vertical column (decoupled from other columns), *each MPI task is decoupled*.

When beginning this research, ARKode did not yet support custom nonlinear solvers, so we focused on the linear solve (plan to upgrade these soon).

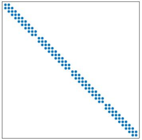
Tempest:

- Block-banded systems on each MPI task
- Direct factorization/solve: `DGBTRF` and `DGBTRS`
- For non-HEVI splittings, this preconditions GMRES



HOMME-NH:

- Tridiagonal systems for w on each MPI task
- Direct factorization/solve: `DGTTRF` and `DGTTRS`
- Post-process result for ϕ update



Unlike most IMEX methods, at large scales the implicit portions of these models are nearly "free" (cost is dominated by explicit RHS MPI communication).

Customized Norms and Tolerances

Accuracy/efficiency for both models hinged on the choice of norm and tolerances for nonlinear and linear solvers.

ARKode utilizes a *weighted root-mean-squared norm* for error-like quantities:

$$\|v\| = \left(\frac{1}{N} \sum_{i=1}^{N} w_i^2 v_i^2 \right)^{1/2}, \qquad w_i = \frac{1}{\varepsilon_r |y_i| + \varepsilon_a}$$

where $y \in \mathbb{R}^N$ is the previous time-step solution, $\varepsilon_r \in \mathbb{R}$ and $\varepsilon_a \in \mathbb{R}^N$.

Newton iterations cease when estimated nonlinear residual norm < 0.1.

GMRES iterations stop when linear residual norm < 0.0005.

Tempest: $\varepsilon_r = \varepsilon_a = 10^{-4}$, based on trial/error in comparisons against native solvers.

HOMME-NH: $\varepsilon_r = 10^{-6}$, and $\varepsilon_a = \begin{cases} 10^{-5}, & \text{for } \mathbf{u} \text{ and } w \text{ components} \\ 0.1, & \text{for } \phi \text{ components,} \\ 1, & \text{for } \Theta \text{ components,} \\ 10^{-6}, & \text{for } \frac{\partial \pi}{\partial \eta} \text{ components} \end{cases}$

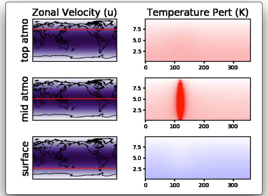
Outline

Test problems (from 2012 Dynamical Core Model Intercomparison Project)

Both codes used two "standard" climate test problems:

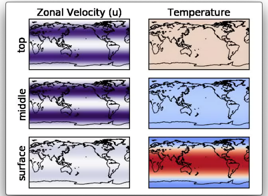
Inertia Gravity Wave

- Small temperature perturbation from equilibrium on "small Earth" (1/125) – gravity wave propagates around the globe.
- Insignificant nonlinear effects; no need for hyperviscosity or vertical remap
- Used to assess temporal convergence and benefit of higher-order methods



Baroclinic Instability

- Small zonal velocity perturbation from equilibrium sets off instability.
- More significant nonlinear effects; stabilization required (reduces overall accuracy to $\mathcal{O}(\Delta t)$).
- Measure surface pressure as proxy for overall error.



Tempest Experiments [Gardner et al., *GMD*, 2018]

We investigated three complementary aspects of integration in Tempest:

1. Alternate IMEX splittings:
   - HEVI; HEVI-DT (explicit vertical continuity and thermo.)
   - IMEX-D (HEVI + implicit ρ); IMEX-DTE (IMEX-D + implicit θ)
2. Alternate Nonlinear solver options:
   - Full Newton solve
   - Linearly-implicit (only performs a single Newton iteration) – "standard practice" in the field, but may not resolve nonlinearity
3. Performance of various Additive Runge–Kutta methods (21 total):
   - 5 from Ascher, Ruuth & Spiteri (1997) – $\mathcal{O}(\Delta t^2) \to \mathcal{O}(\Delta t^3)$
   - 3 from Kennedy & Carpenter (2003) – $\mathcal{O}(\Delta t^3) \to \mathcal{O}(\Delta t^5)$
   - 1 from Giraldo et al. (2013) – $\mathcal{O}(\Delta t^2)$
   - 4 SSP from Pareschi & Russo (2005) – $\mathcal{O}(\Delta t^2) \to \mathcal{O}(\Delta t^3)$
   - 6 SSP from Higueras et al. (2006, 2009, 2014) – $\mathcal{O}(\Delta t^2) \to \mathcal{O}(\Delta t^3)$
   - 2 from Conde et al. (2017) – $\mathcal{O}(\Delta t^3)$

Tempest Results [Gardner et al., *GMD*, 2018]

HEVI/IMEX splittings:

- Stability $\propto$ implicitness: IMEX-DTE $>$ IMEX-D $>$ HEVI $>$ HEVI-DT.
- Horizontally implicit terms increase runtime by 25% $\rightarrow$ 60%.
- Resulting IMEX-D cost between HEVI and HEVI-DT; HEVI was best overall.

Linearly-implicit (LI) solver vs Newton (N) solver:

- LI solve sufficient for gravity wave (accurate to discretization error).
- LI solve insufficient to resolve nonlinearity in baroclinic test at desired step sizes (nonphysical w); $2 \rightarrow 4$ N iterations required to recover nonlinear effects.

ARK methods:

- No benefit of high over low order (4-5 vs 2-3) at desired step sizes.
- Most SSP methods show nonphysical vertical velocities for HEVI splittings; others unstable except at small Δt (likely due to instability along imaginary axis).
- Best overall stability/accuracy from $\mathcal{O}(\Delta t^3)$ methods by Ascher, Ruuth & Spiteri (1997) and Kennedy & Carpenter (2003).

HOMME-NH Experiments [Vogl et al., *in prep*]

Resolved to use a HEVI formulation and a full Newton solve, we then focused on stability, accuracy and novel ARK methods:

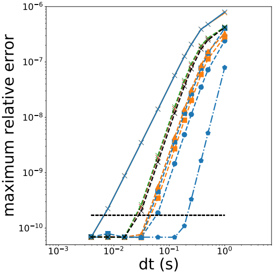
1. Effects of post-processing applied to time-step solutions:
   - hyperviscosity
   - vertical remap (dynamic adjustment to vertical coordinate η)
2. Performance of various ARK methods (22 total), and one ERK method:
   - 5 from Ascher, Ruuth & Spiteri (1997) – $\mathcal{O}(\Delta t^2) \to \mathcal{O}(\Delta t^3)$
   - 2 from Kennedy & Carpenter (2003) – $\mathcal{O}(\Delta t^3) \to \mathcal{O}(\Delta t^4)$
   - 2 from Conde et al. (2017) – $\mathcal{O}(\Delta t^3)$
   - 12 from Steyer et al. (*in prep*) – $\mathcal{O}(\Delta t^2) \to \mathcal{O}(\Delta t^3)$
   - 1 by R., from Vogl et al. (*in prep*) – $\mathcal{O}(\Delta t^3)$
   - 1 ERK method from Guerra & Ullrich (2016) – $\mathcal{O}(\Delta t^3)$

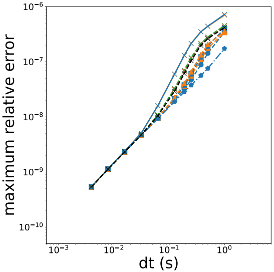
   Methods from Steyer, R., and Guerra & Ullrich are optimized to maximize linear stability along imaginary axis.

HOMME-NH Results (hyperviscosity & vertical remap) [Vogl et al., *in prep*]

Maximum relative θ error vs Δt for gravity wave test (top/bottom show without/with post-processing, resp.):

- All ARK methods converge at analytical order to reference accuracy when post-processing is disabled.
- All methods reduce to first order when post-processing is enabled.
- Unfortunately, production runs rapidly go unstable without stabilization.
- However, higher-order methods still show improved error at larger Δt (desired production step sizes).





HOMME-NH Results (ARK methods) [Vogl et al., *in prep*]

Measured the maximum relative surface pressure error, with accuracy tolerance set by comparison against "trusted" results. Two questions:

A. What is the max. Δt for each method to obtain solution within tolerance?

B. Which methods can run at the desired hydrostatic time step $\Delta t = 300$s? By how much do those methods exceed tolerance?

Question A on left, B on right (only best shown). *WT* is wall-clock (hr), *Exc* is exceedance, *IMKG* (Steyer et al.), *ARK* (Kennedy & Carpenter), *DBM* (Vogl et al.):

Method	Δt	$\frac{\Delta t}{f^I}, \frac{\Delta t}{f^E}$	WT
IMKG243b	270	90,68	0.9
DBM453	270	68,54	1.1
ARK324	240	80,60	1.0
IMKG242b	240	120,60	0.9
ARK436	216	43,36	1.6

Method	Δt	$\frac{\Delta t}{f^I}, \frac{\Delta t}{f^E}$	WT	Exc.
DBM453	300	75,60	1.0	3%
IMKG252b	300	150,60	0.8	61%
IMKG254$\frac{a}{b}$	300	75,60	1.0	61%
IMKG243b	270	90,68	0.9	55%
IMKG343a	270	90,68	0.9	55%

Methods tuned for stability on imaginary axis far outperform existing methods.

SMU

Outline

Conclusions & Future Work

In summary:

- Modular structure of ARKode/SUNDIALS allows use of problem-specific data structures and solvers within high-order time integration.
- Trivial to explore different ARK methods (just supply pairs of Butcher tables).
- Simplified exploration of IMEX splittings and nonlinear solvers – for 'optimal' efficiency these should be modified together, but JFNK can 'clean up' initially.
- Exploration can identify critical features for newly-derived methods.
- Top ARK methods are newly-developed versions by Steyer and R. – IMKG for raw speed; but DBM overall (stability & accuracy, without significant speed penalty)

Next steps:

- Reconstruct ARKode interfaces in Tempest and HOMME-NH to allow customized nonlinear solves (eliminate MPI communication in implicit solves).
- Investigate hyperviscosity/remap *within* stage solves to retain high order.
- Temporal adaptivity (current lack of a sufficient 'ecosystem' of embedded ARK methods). Invent new "IMKG" and "DBM" methods with embeddings.

Thanks & Acknowledgements

Collaborators:

- Jorge Guerra [UC Davis]
- Mark Taylor [SNL]
- Oksana Guba [SNL]

Grant/Computing Support:

- DOE BER, SciDAC & ECP Programs
- SMU Center for Scientific Computation

Software:

- ARKode – http://faculty.smu.edu/reynolds/arkode
- SUNDIALS – https://computation.llnl.gov/casc/sundials
- Tempest – http://github.com/paullric/tempestmodel
- E3SM (HOMME-NH) – https://github.com/E3SM-Project/E3SM

Support for this work was provided by the Department of Energy, Office of Science. SMU portion funded under Lawrence Livermore National Laboratory subcontracts B617813, B621892, B622784 and B626484.

References

- Ullrich, *Geosci. Model Dev.*, 7, 2014.
- Taylor & Steyer, *in prep.*, 2019.
- Guerra & Ullrich, *Geosci. Model Dev.*, 9, 2016.
- Ascher, Ruuth & Spiteri, *Appl. Numer. Math.*, 25, 1997.
- Araújo, Murua & Sanz-Serna, *SIAM J. Numer. Anal.*, 34, 1997.
- Ullrich, R., Guerra & Taylor, *J. Sci. Comput.*, 375, 2018.
- Gardner, Guerra, Hamon, R. Ullrich & Woodward, *Geosci. Model Dev.*, 11, 2018.
- Kennedy & Carpenter, *Appl. Numer. Math.*, 44, 2003.
- Giraldo, Kelly & Constantinescu, *SIAM J. Sci. Comput.*, 35, 2013.
- Pareschi & Russo, *J. Sci. Comput.*, 25, 2005.
- Higueras, *SIAM J. Numer. Anal.*, 44, 2006.
- Higueras, *J. Sci. Comput.*, 39, 2009.
- Higueras Happenhofer, Koch & Kupka, *J. Comput. Appl. Math.*, 272, 2014.
- Conde, Gottlieb, Grant & Shadid, *J. Sci. Comput.*, 73, 2017.
- Vogl, Steyer, R., Ullrich & Woodward, *in prep.*, 2019.
- Steyer, Vogl, Taylor & Guba, *in prep.*, 2019.