Stabilization, Robustness and Accuracy of Numerical Simulations of Fluids From Thin Films to Global Scales

Valeria Barra, Ph.D.
Assistant Professor,
CSRC + Math Department, SDSU
valeriabarra.org

The Third Joint SIAM/CAIMS Annual Meetings (AN25), Montréal, Québec, Canada

July 29th, 2025

Overview

SDSU

The need for control of numerical accuracy

It was the year 1986... when Roache, Ghia, and White wrote in the Journal of Fluids Engineering editorial:

Journal of Fluids Engineering **Editorial**

Editorial Policy Statement on the Control of Numerical Accuracy

A professional problem exists in the computational fluid dynamics community and also in the broader area of computational physics. Namely, there is a need for higher standards on the control of numerical accuracy.

The numerical fluid dynamics community is aware of this problem but, although individual researchers strive to control accuracy, the issue has not to our knowledge been addressed standards should be raised. Consequently, this journal hereby announces the following policy:

The Journal of Fluids Engineering will not accept for publication any paper reporting the numerical solution of a fluids engineering problem that fails to address the task of systematic truncation error testing and accuracy estimation.

"Whatever the authors use will be considered in the review process, but we must make it clear that *a single calculation in a fixed grid will not be acceptable*, since it is impossible to infer an accuracy estimate from such a calculation."

V & V

Verification and Validation in scientific computations:

- Verification: “Are we solving the problem *right*?”
- Validation: “Are we solving the *right* problem?”

Verification is a mathematical exercise. In principle, it can always be completed even when the analytical solution is now known (though one must always establish sufficient resolution).

Validation often involves comparing with observational/experimental data, it is always ongoing, subject to tolerances.

Consistency, Stability, and Convergence

- Consistency: describes how well the numerical scheme aproximates the PDE (if it is at least of order 1 $\implies$ it is consistent — The residual reduces under grid refinement).
- Stability: Numerical stability concerns how errors introduced during the execution of an algorithm affect the result. It is a property of an algorithm rather than the problem being solved [**Higham**]. This gets subtle for problems like incompressible materials or contact mechanics.
- Convergence: When the solution of the approximated equation approaches the actual solution of the continuous equation.

Lax equivalence Theorem:

Consistency + Convergence $\implies$ Stability

Consistency + Stability $\implies$ Convergence

Hence,

Consistency + Convergence $\iff$ Stability

All good, but in praxtice?

These are foundational theoretical tools, and can often be tested in practice, but

- it doesn't establish that the code works
- it's not possible to prove convergence for many real-world problems
- there are open research questions about whether numerous important problems are even well-posed

Empirical measurement of convergence

Convergence on a problem with an analytical solution:

- These can be great, but analytical solutions of nonlinear equations are extremely hard to find.
- Such solutions usually have many symmetries and rarely activate all terms.

Self-convergence:

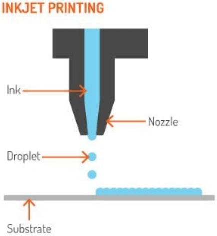
- Just set up a problem and solve it on a sequence of meshes, use a refined solution as reference (perhaps using Richardson extrapolation), then plot error.
- You've checked that the code convergences to *some* solution, not the *correct* solution. (You could have a factor of 2 typo, or more serious mistakes.)

Method of Manufactured Solutions (MMS):

- Errors that affect solution accuracy can easily be detected.
- There are some technical issues with singular or under-resolved problems (shocks, material discontinuities).

PhD research: Viscoelastic fluids

PhD in Applied Math from NJIT on numerical simulations of thin films (long-waves) of non-Newtonian viscoelastic fluids



Viscoelastic materials:

- hysteresis:
  loop in stress-strain rate curve
- stress relaxation:
  constant $\epsilon \Rightarrow$ decreasing σ
- creep:
  constant $\sigma \Rightarrow$ increasing ϵ

Governing equations

Conservation laws for incompressible fluids:

$$\rho\,(\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla(p+\Pi) + \nabla\cdot\boldsymbol{\sigma} + \mathbf{F_b}\ , \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0\,, \tag{2}$$

where, in $2D$, $\mathbf{u} = (u(x,y,t), v(x,y,t))$, is the vector velocity field, $\nabla = (\partial_x, \partial_y)$, p is the pressure, Π is the disjoining pressure due to the van-der-Waals interaction (attraction/repulsion) force, and $\mathbf{F_b} = (\rho g \sin\alpha, -\rho g\cos\alpha)$ body force.

Jeffreys' model:

$$\boldsymbol{\sigma} + \lambda_1 \partial_t \boldsymbol{\sigma} = 2\eta(\dot{\boldsymbol{\epsilon}} + \lambda_2 \partial_t \dot{\boldsymbol{\epsilon}})$$

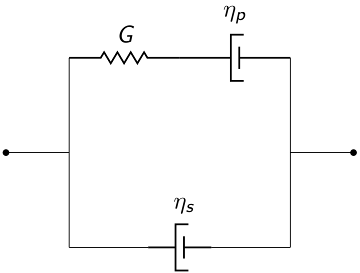
Jeffreys constitutive model

Jeffreys Model: linear viscoelastic fluids

$$\sigma_{ij} + \lambda_1 \partial_t \sigma_{ij} = 2\eta \left(\dot{\epsilon}_{ij} + \lambda_2 \partial_t \dot{\epsilon}_{ij} \right) \ ,$$

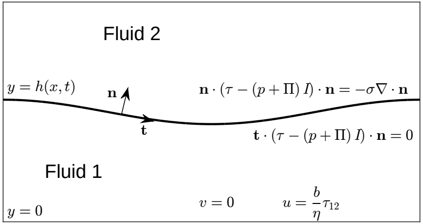
with $\lambda_2 = \lambda_1 \frac{\eta_s}{\eta_s + \eta_p}$, and $\eta = \eta_s + \eta_p \Rightarrow \lambda_1 \geq \lambda_2$. With η_s and η_p viscosity of Newtonian solvent and polymeric solute, respectively.

λ_1 relaxation time, λ_2 retardation time.



Schematic and Nondimensionalization

Setup and boundary conditions of the two-phase interfacial flow:



Schematic of the fluid interface and boundary conditions in the case in which $\mathbf{F_b} = 0$.
Kinematic BC: $Df/Dt = f_t + \mathbf{u} \cdot \nabla f = 0$, with $f(x, y, t) = y - h(x, t)$.

Scalings:

$$x = Lx^*\,,\ (y, h, h_\star, b) = H(y^*, h^*, h_\star^*, b^*)\,,$$

$$(p, \Pi) = P(p^*, \Pi^*)\,,\ u = Vu^*\,,\ v = \varepsilon V v^*\,,$$

$$(t, \lambda_1, \lambda_2) = T(t^*, \lambda_1^*, \lambda_2^*)\,,\ \gamma = \frac{V\eta}{\varepsilon^3}\gamma^*\,,$$

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} = \frac{\eta}{T} \begin{pmatrix} \sigma_{11}^* & \frac{\sigma_{12}^*}{\varepsilon} \\ \frac{\sigma_{21}^*}{\varepsilon} & \sigma_{22}^* \end{pmatrix}\,,$$

where $H/L = \varepsilon \ll 1$ is the small parameter. Pressure is scaled with $P = \eta/(T\varepsilon^2)$, and time with $T = L/V$.

Dimensionless governing equations

Long-wave approximation in two spatial dimensions:

$$(1+\lambda_2\partial_t)h_t + \frac{\partial}{\partial x}\left\{(\lambda_2-\lambda_1)\left(\frac{h^2}{2}Q - hR\right)h_t \right.$$
$$\left. + \left[(1+\lambda_1\partial_t)\frac{h^3}{3} + (1+\lambda_2\partial_t)bh^2\right]\frac{\partial}{\partial x}\left(\frac{\partial^2 h}{\partial x^2}+\Pi(h)\right)\right\} = 0\,,$$

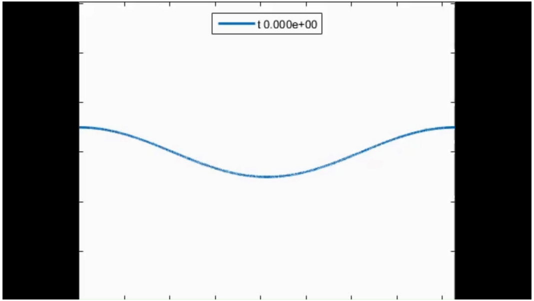
$$Q + \lambda_2 Q_t = -\tfrac{\partial}{\partial x}\left(\tfrac{\partial^2 h}{\partial x^2}+\Pi(h)\right),$$
$$R + \lambda_2 R_t = -h\tfrac{\partial}{\partial x}\left(\tfrac{\partial^2 h}{\partial x^2}+\Pi(h)\right).$$

disjoining pressure: $\Pi(h) = \frac{\gamma(1-cos\theta_e)}{Mh_\star}\left[\left(\frac{h_\star}{h}\right)^n - \left(\frac{h_\star}{h}\right)^m\right]$,
θ_e contact angle, $M=0.5$, $(n=3, m=2)$, $h_\star$ precursor film thickness.

Jeffreys' constitutive law:

$$\boldsymbol{\sigma} + \lambda_1\partial_t\boldsymbol{\sigma} = 2\eta(\dot{\epsilon} + \lambda_2\partial_t\dot{\epsilon})$$

Dewetting film



A viscoelastic dewetting film exhibits secondary satellite droplets in the dewetting region that viscous films do not exhibit.

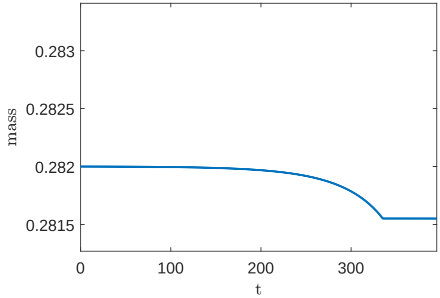
Verification and Consistency: persistent under mesh refinement.
Validation: Similar droplet formations observed in viscoelastic jets undergoing capillary thinning experimental in [**Clasen2006**].

Numerical Instabilities:

For high Weissenberg number (Wi), $Wi = \frac{\text{elastic forces}}{\text{viscous forces}}$, which indicates the degree of anisotropy or orientation generated by the deformation (suitable for shear or elongation movemenets), the problem becomes stiffer and numerical instabilities may arise.

Typical symptom: simulation crashes.

Closer look / Investigation: loosing mass! (even only 1‰.)

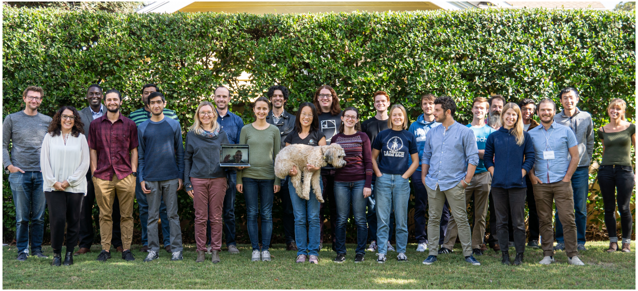


Solution:
adaptive time stepping.

Overview

About CliMA

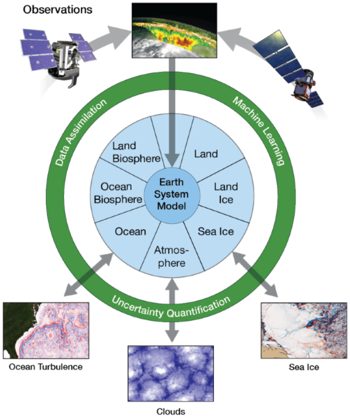
The Climate Modeling Alliance (CliMA) is a coalition of scientists, engineers, and applied mathematicians from **Caltech**, **MIT**, and the **NASA Jet Propulsion Laboratory**, who is building the first Earth System Model (ESM) in the Julia programming language that automatically learns from diverse data sources to produce more accurate climate predictions with quantified uncertainties.



Thanks to all team members: Tapio Schneider[1] (PI), Paul Ullrich[2], Oswald Knoth[3], Simon Byrne[1], Jake Bolewski[1], Charles Kawczynski[1], Sriharsha Kandala[1], Gabriele Bozzola[1], Zhaoyi Shen[1], Jia He[1], Kiran Pamnany[1], Ben Mackay[1], Akshay Sridhar[1], Dennis Yatunin[1], Lenka Novak[1], Toby Bischoff[1], Daniel (Zhengyu)Huang[1], Andre Souza[4], Yair Cohen[1]

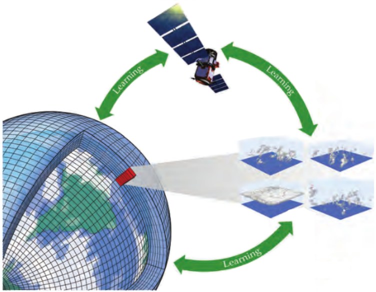
1: Caltech, 2: UC Davis, 3: TROPOS, 4: MIT

Goals



[Source: courtesy of Tapio Schneider (Caltech)]

- The Earth System Model (ESM) will be grounded in physics (using sub-grid scale, cloud-resolving modeling) and designed for automated calibration of parameters using machine learning.
- High-resolution Large-Eddy Simulations (LES) are used to inform parametrizations of the global circulation model (GCM), which in turn, can be used for large-scale forcings to force the LES.



[Source: Physics Today - June 2021, pg. 44-51]

Technical and Scientific Aims

- Support parallel computing on CPUs and GPUs using a common open-source code base written in the high-level, dynamic Julia programming language (familiar syntax, similar to Python and Matlab).
- Julia has an interactive REPL, is Just-In-Time (JIT) compiled (triggered by first evaluation of function). Allows polymorphism via multiple dispatch (at compile or run time).
- Can write generic code, compiler will specialize on types of calling arguments, e.g., `f(x::AbstractArray)` where `AbstractArray` can be `Array` of `Float32`, `Float64` or a `CuArray`.
- Be accessible and extensible by a mixture of users.

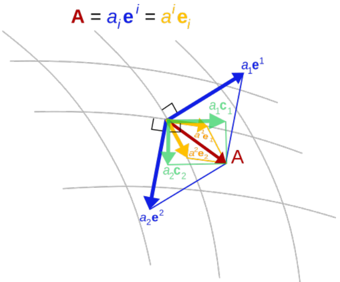
- For the atmosphere model, support both Large-Eddy Simulation (LES) and General Circulation Model (GCM) configurations (i.e., Cartesian and spherical geometries).
- Allow specification of any governing equations and boundary conditions by composing operators.
- Support non-uniform unstructured meshes.

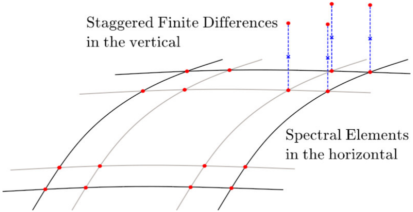
ClimaCore.jl

ClimaCore.jl — the new dynamical core (*dycore*).

A library (suite of tools) for constructing flexible space discretizations.

- Geometry:
    - Supports different geometries (Cartesian & spherical).
    - Supports covariant/contravariant vector representation for curvilinear, non-orthogonal systems and Cartesian vectors for Euclidean spaces.
- Space Discretizations:
    - Horizontal: Support both Continuous Galerkin (CG) and Discontinuous Galerkin (DG).
    - Vertical: staggered Finite Differences (FD).





SDSU

CliMA Dycore key points

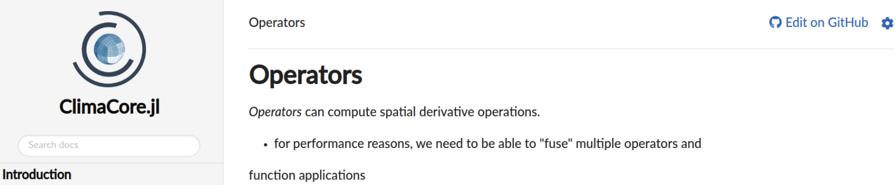
- A non-hydrostatic dynamical core with consistent moist thermodynamics and total energy as prognostic variable
- The model uses a hybrid spectral element/finite difference discretization that exactly conserves mass, total energy, and water
- Excellent CPU/GPU scaling makes the model suitable for cloud computing

The dycore serves both the Land and Atmos model.
ClimaAtmos is the atmosphere component with:

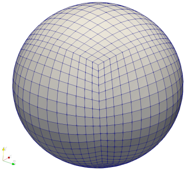
- a new prognostic EDMF
- calibrated parameters from observations and high resolution models with uncertainty estimates
- library of pluggable radiation schemes (gray and RRTMGP), turbulent surface fluxes (bulk and Monin Obukhov with customizable functions), microphysics schemes (0 to 3 moment), vertical transport terms (diffusion and EDMF)

[A preview of the dycore paper on the ESS open archive.]

Some personal contributions



- Geometry and Topology modules
- Grid generation: Different "cubed-sphere" meshes (Equiangular, Equidistant, Conformal)
- High-order differential operators and flux limiters
- Unit tests, integration tests and examples
- Docs, tutorials, CliMAWorkshops (https://github.com/CliMA/ClimaWorkshops)



Examples: Shallow-water equations

The shallow water equations
(in vector-invariant form) on a rotating sphere:

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\boldsymbol{u}) = 0 \tag{3a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla(\Phi + \tfrac{1}{2}\|\boldsymbol{u}\|^2) = (\boldsymbol{u} \times (f + \nabla \times \boldsymbol{u})) \tag{3b}$$

where f is the Coriolis term and $\Phi = g(h + h_s)$.

Written in terms of a curvilinear, non-orthogonal basis:

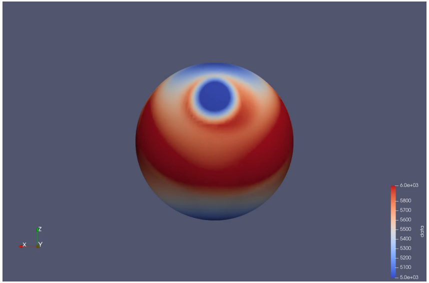
$$\frac{\partial h}{\partial t} + \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(hJu^j\right) = 0 \tag{4a}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial \xi^i}(\Phi + \tfrac{1}{2}\|\boldsymbol{u}\|^2) = E_{ijk}u^j(f^k + \omega^k) \tag{4b}$$

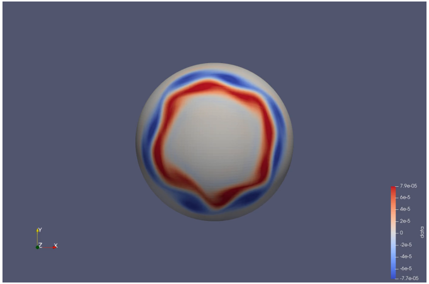
```julia
∇ = Operators.Gradient()
∇⋅ = Operators.WeakDivergence()
curl = Operators.Curl()

@. begin
    dydt.h = - ∇⋅(y.h * y.u)
    dydt.u = -∇(g * (y.h + h_s) - norm(y.u)^2 / 2) + y.u × (f + curl(y.u))
end
```

Shallow-water equation Test Cases

ClimaCore.jl/examples/sphere/shallow_water.jl



Shallow-water equations suite, Test Case 5 [**Williamson1992**].
Zonal flow over an isolated mountain.



Shallow-water equations suite, barotropic instability test case [**Galewsky2004**]. Zonal jet with compact support at mid-latitude. A small height disturbance is then added, which causes the jet to become unstable and collapse into a highly vortical structure.

Examples: Advection (transport) problems

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \rho \boldsymbol{u}, \tag{5a}$$

$$\frac{\partial Q}{\partial t} = -\nabla \cdot Q \boldsymbol{u}, \tag{5b}$$

Transport of a passive tracer, with $Q = \rho q$, where q denotes tracer concentration (i.e., mixing ratio or mass of tracer per mass of dry air, in dry problems, or mass of tracer per mass of moist air, in moist problems) per unit mass, and ρ fluid density.

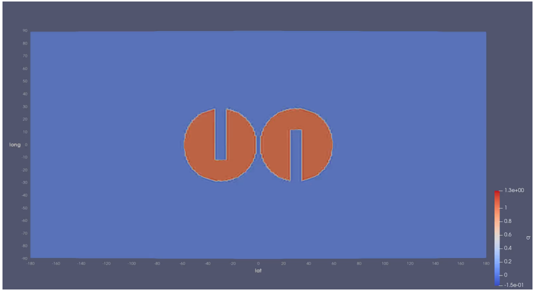
```
∇· = Operators.WeakDivergence()

@. dydt.ρ =  - ∇·(y.ρ * u)  # contintuity equation
@. dydt.ρq = - ∇·(y.ρq * u) # advection of tracer equation
```

Quasimonotone flux limiters

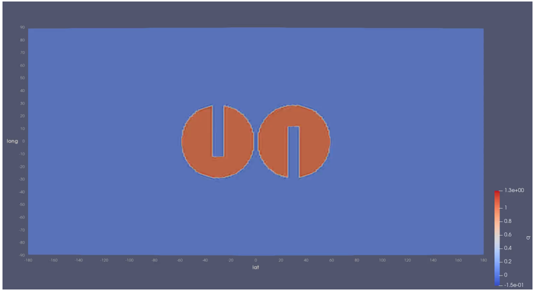
- Traditional SEM advection operator is oscillatory but due to its mimetic properties it is locally conservative and has a monotone property with respect to element averages
- We use a class of optimization-based locally conservative quasimonotone (monotone with respect to the spectral element nodal values) limiters that prevent all overshoots and undershoots at the element level [**GubaOpt2014**]
- It also maintains quasimonotonicity even with the addition of a dissipation term such as viscosity or hyperviscosity
- The only additional interelement communication introduced is in determining the suitable minimum and maximum constraints

Flux limiter test case: slotted cylinders on a 2D sphere

$p = 6$, $ne = 20 \times 20 \times 6$ (effective resolution 0.75° at equator.)



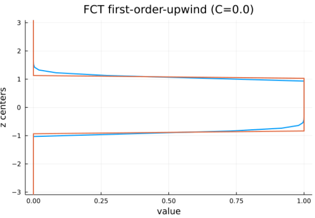
No limiter.



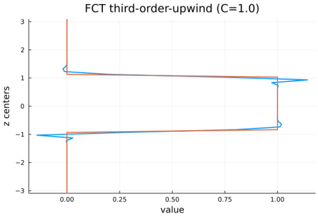
With limiter.

Flux-Corrected Transport

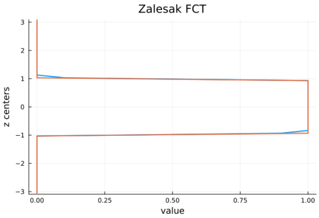
For the advection operator discretized by Finite Differences, the Flux-corrected transport (FCT) approximates with a high-order scheme in regions where the solution is smooth, and low-order monotone scheme where the solution is poorly resolved or discontinuous [**Zalesak1979**].



Dissipative and very dispersive



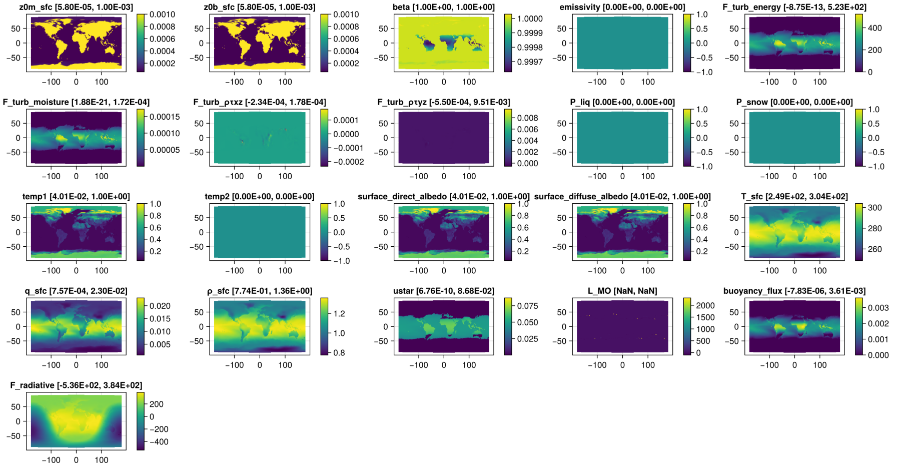
Less dissipative and less dispersive (but over/under-shoots)



No over/under-shoots and limited dispersion.

D. Yatunin, S. Byrne, . . . , **V. Barra**, O. Knoth, P. Ullrich, T. Schneider, *The Climate Modeling Alliance Atmosphere Dynamical Core: Concepts, Numerics, and Scaling*, in review for JAMES (2025)

Quality control, testing, V & V: Continuous Integration of the coupler

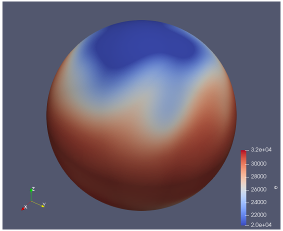


Continuous integration test, for AMIP with target resolution ($\sim$ 26 km in the horizontal & 43 vertical levels), topography and diagnostic EDMF.

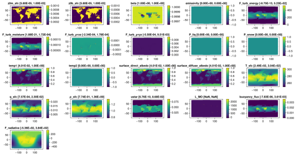
Conclusions and Future Directions

- Introduced numerical analysis key concepts and typical challenges
- Introduced the CliMA Earth System Model (ESM):
  - Introduced ClimaCore.jl, the new open-source dycore for the atmosphere and land components of the ESM, entirely written in the Julia dynamic language
  - We showed examples of applications for atmospheric flows and flux limiters to overcome oscillation challenges for the high-order SEM advection operator

Future Directions:
Stabilization of high-order methods for various CFD applications with Yiyue, first-year PhD student.



[Held-Suarez 180-day simulation.]



[AMIP w/ topography and diagnostic EDMF.]

References

[1] Nick Higham, *What is Numerical Stability?*, `https://nhigham.com/2020/08/04/what-is-numerical-stability/`

[2] Jie Li, Marco A. Fontelos, *Drop dynamics on the beads-on-string structure for viscoelastic jets: A numerical study*, Physics of Fluids, 15, 922–937, 2003

[3] David L. Williamson, John B. Drake, James J. Hack, Rüdiger Jakob, Paul N. Swarztrauber, *A standard test set for numerical approximations to the shallow water equations in spherical geometry*, Journal of Computational Physics, 102(1), 211-224, 1992.

[4] Joseph Galewsky, Richard K. Scott, Lorenzo M. Polvani, *An initial-value problem for testing numerical models of the global shallow-water equations*, Tellus A: Dynamic Meteorology and Oceanography, 56(5), 429-440, 2004.

[5] Oksana Guba, Mark Taylor, Amik St-Cyr, *Optimization-based limiters for the spectral element method*, ournal of Computational Physics, 267, 176-195, 2014.

[6] Jay P Boris, David L Book, *Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works*, Journal of Computational Physics, 11(1), 38-69, 1973.

[7] Steven T Zalesak, *Fully multidimensional flux-corrected transport algorithms for fluids*, Journal of computational physics, 31(3), 335–362, 1979.