

From nanofilms and droplets to dycores and couplers: a numerical modeling journey

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

 valeriabarra.org

Scripps Institution of Oceanography, Climate and Atmosphere Seminar

April 29th, 2025



Overview

1 Introduction

2 Interfacial flows

3 libCEED

4 CliMA

ClimaCore.jl

Examples for Climate Applications

ClimaCoupler.jl

Coupler Overview

5 Recent and Future Projects

6 Conclusions

About me

- From Siena, Tuscany, Italy
- B.S. and M.Sc. in Mathematical Sciences at the University of Siena
- Exchange program + Ph.D. program in Applied Math at NJIT, Newark, NJ
- Postdoc at the University of Colorado at Boulder in the ECP CEED project
- Research Software Engineer (3+ years) at Caltech in the CLiMA project

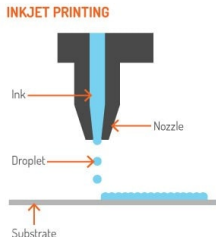
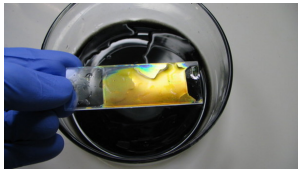
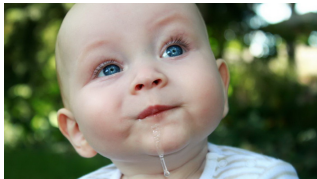


University of Colorado
Boulder

Caltech

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

Mechanical system analogs

- Hookean: elastic solids

$$\sigma_{ij} = 2G\epsilon_{ij}$$

G shear elastic modulus

- Newtonian: viscous fluids

$$\sigma_{ij} = 2\eta\dot{\epsilon}_{ij}$$

η dynamic (shear) viscosity

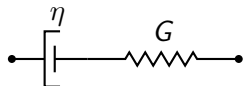
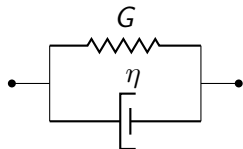
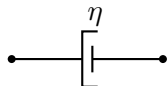
- Kelvin-Voigt: linear viscoelastic solids

$$\sigma_{ij} = 2G\epsilon_{ij} + 2\eta\dot{\epsilon}_{ij}$$

- Maxwell: linear viscoelastic fluids

$$\sigma_{ij} + \lambda_1 \partial_t \sigma_{ij} = 2\eta\dot{\epsilon}_{ij}$$

λ_1 relaxation time, s. t. $\lambda_1 = \eta/G$.



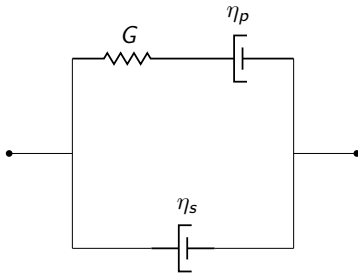
Mechanical system analog for Jeffreys

Jeffreys Model: linear viscoelastic fluids

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

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.



Governing equations

Conservation laws for incompressible fluids:

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

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

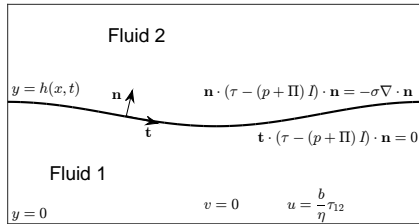
where, in $2D$, $\mathbf{u} = (u(x, y, t), v(x, y, t))$, is the vector velocity field, $\nabla = (\partial_x, \partial_y)$, p is the pressure, Π is the disjoining pressure due to the van-der-Waals interaction (attraction/repulsion) force, and $\mathbf{F}_b = (\rho g \sin \alpha, -\rho g \cos \alpha)$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

Jeffreys' model:

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

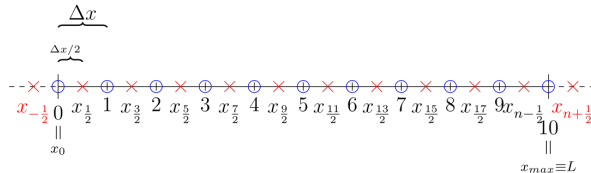
Schematic

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 $f(x, y, t) = y - h(x, t)$.



Nondimensionalization

Scalings:

$$x = Lx^*, \quad (y, h, h_*, b) = H(y^*, h^*, h_*^*, b^*), \quad (p, \Pi) = P(p^*, \Pi^*), \\ u = Vu^*, \quad v = \varepsilon Vv^*, \quad (t, \lambda_1, \lambda_2) = T(t^*, \lambda_1^*, \lambda_2^*), \quad \gamma = \frac{V\eta}{\varepsilon^3}\gamma^*,$$

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

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:

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

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

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

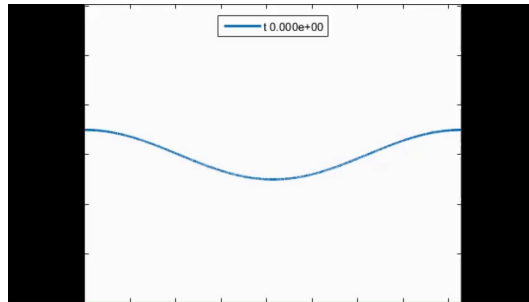
$$\text{disjoining pressure: } \Pi(h) = \frac{\gamma(1 - \cos \theta_e)}{M h_*} \left[\left(\frac{h_*}{h} \right)^n - \left(\frac{h_*}{h} \right)^m \right],$$

θ_e contact angle, $M = 0.5$, ($n = 3, m = 2$), h_* precursor film thickness.

Jeffreys' constitutive law:

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

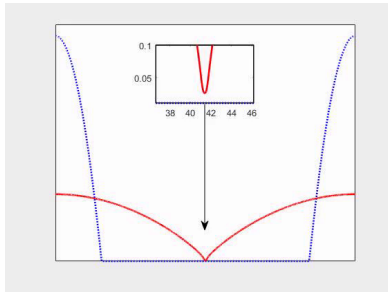
Dewetting film



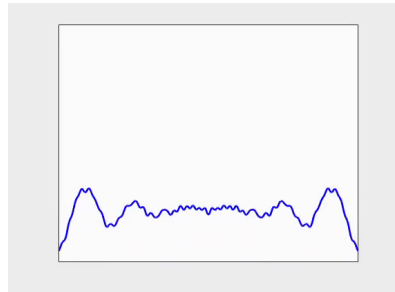
A viscoelastic dewetting film exhibits secondary satellite droplets in the dewetting region that viscous films do not exhibit.

Validation: Similar droplet formations observed in viscoelastic jets undergoing capillary thinning experimental in [Clasen2006].

Dewetting films with slippage and on an inverted plane

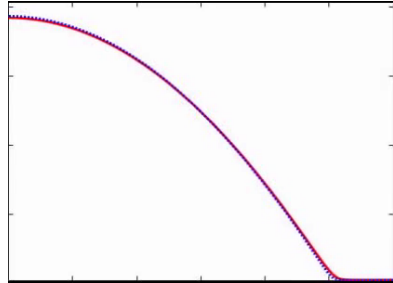
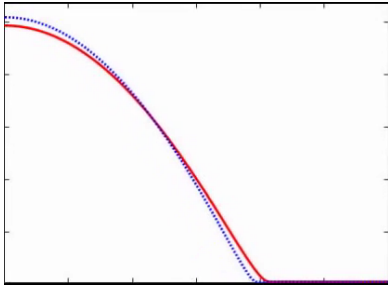


A higher slippage with the substrate suppresses formation of satellite droplets



Rayleigh-Taylor instabilities in the case of an inverted plane

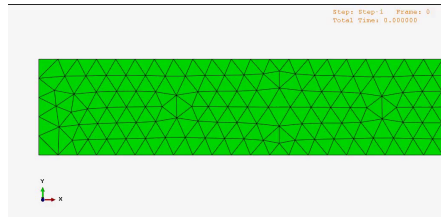
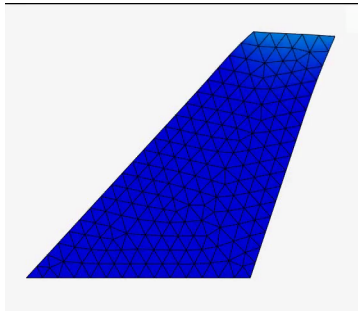
Spreading and receding drops



The **viscoelastic** drop spreads faster and recedes slower compared to the **Newtonian** one

Membranes

Shear and extensional free-boundary flows of viscoelastic membranes



Linear finite elements with plane stress formulation. Different constitutive models considered: elastic, viscous, viscoelastic (Maxwell)

Publications

V. Barra, S. Afkhami, L. Kondic, *Mathematical and numerical modeling of thin viscoelastic films of Jeffreys type subjected to the van der Waals and gravitational forces*, EPJE, 42, 1 – 14 (2019)

V. Barra, S. A. Chester, S. Afkhami, *Numerical Simulations of Nearly Incompressible Viscoelastic Membranes*, Computers & Fluids, 175, 36 – 47 (2018)

V. Barra, S. Afkhami, L. Kondic, *Interfacial dynamics of thin viscoelastic films and drops*, J. Non-Newt. Fluid Mech., 237, 26 – 38 (2016)

B. Adeyemi, P. Jadhawar, L. Akanji, **V. Barra**, *Effects of fluid–fluid interfacial properties on the dynamics of bounded viscoelastic thin liquid films*, J. Non-Newt. Fluid Mech., 309, 104893 (2022)



Overview

1 Introduction

2 Interfacial flows

3 **libCEED**

4 CliMA

ClimaCore.jl

Examples for Climate Applications

ClimaCoupler.jl

Coupler Overview

5 Recent and Future Projects

6 Conclusions

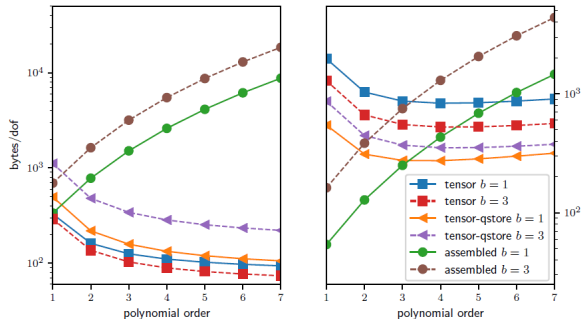
Fast algebra for high-order element-based discretizations: libCEED

Postdoc project supervised by Jed Brown at



libCEED Overview

- High-order methods have been considered too expensive for decades because relied on sparse matrices assembly, which results in $O(p^d)$ storage and $O(p^{2d})$ compute per degree of freedom (DoF) in d dimensions, for basis polynomial order p
- On the other hand, optimized spectral element implementations can achieve $O(1)$ storage and $O(p)$ compute per DoF



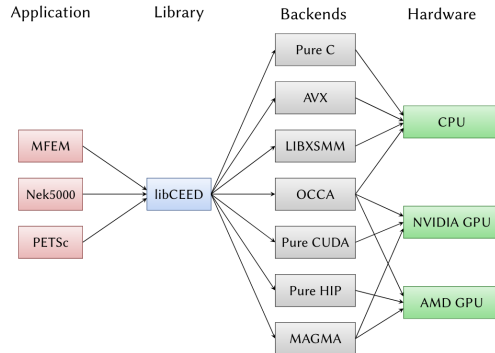
[Courtesy: Jed Brown]



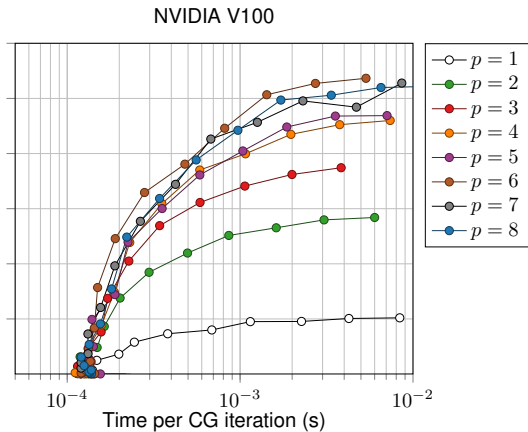
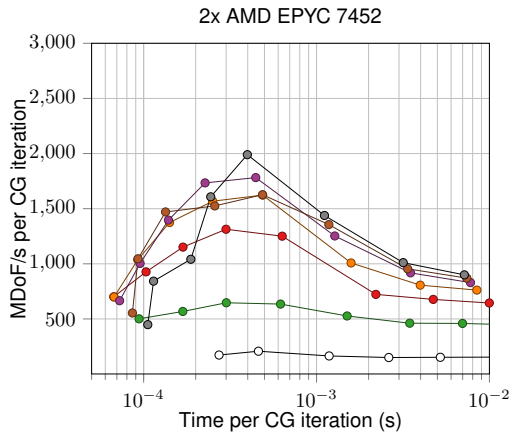
[<https://github.com/CEED/libCEED/>]

libCEED: the Code for Efficient Extensible Discretization

- libCEED uses a matrix-free operator description, based on a purely algebraic interface, where user only specifies the action of weak form operators
- Primary target: high-order finite/spectral element methods (FEM/SEM) exploiting tensor-product structure
- Open-source (BSD-2 license) C library with Fortran, Python, Julia and Rust interfaces
- libCEED is light-weight and performance-portable via run-time selection of specialized implementations (backends) optimized for CPUs and GPUs

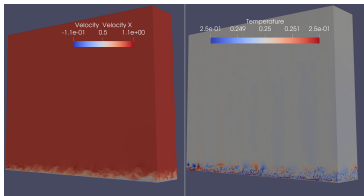


Performance

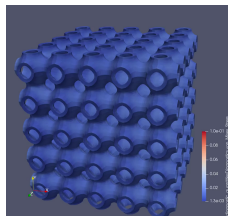


Application examples

Integration of libCEED with the Portable, Extensible Toolkit for Scientific Computation (PETSc) for examples in fluids and solid mechanics

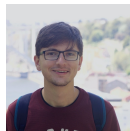


DNS of a flat plate synthetic turbulence generator.



A compressed Schwarz periodic minimal surface.
Applications: additive manufacturing, soft robotics

In preparation: J. Brown, **V. Barra**, N. Beams, L. Ghaffari, M. Knepley, W. Moses, R. Shakeri, K. Stengel, J. Thompson, J. Zhang, *Performance-Portable Solid Mechanics via Matrix-Free p -Multigrid*, in preparation
ArXiv: arXiv:2204.01722v3



Publications

A. Abdelfattah, **V. Barra**, N. Beams et al., *GPU algorithms for Efficient Exascale Discretizations*, Parallel Computing, 108, 102841 (2021)

T. Kolev et al., *Efficient exascale discretizations: High-order finite element methods*, Int J. High. Perform. Comput. Appl., 6, 527-552 (2021)

J. Brown, A. Abdelfattah, **V. Barra** et al., *libCEED: Fast algebra for high-order element-based discretizations*, JOSS 63, 2945 (2021)

V. Barra, J. Brown, J. Thompson, Y. Dudouit, *High-performance operator evaluations with ease of use: libCEED's Python interface*, Proceedings of the 19th Python in Science Conference 85 - 90 (2020)

Overview

1 Introduction

2 Interfacial flows

3 libCEED

4 **CliMA**

ClimaCore.jl

Examples for Climate Applications

ClimaCoupler.jl

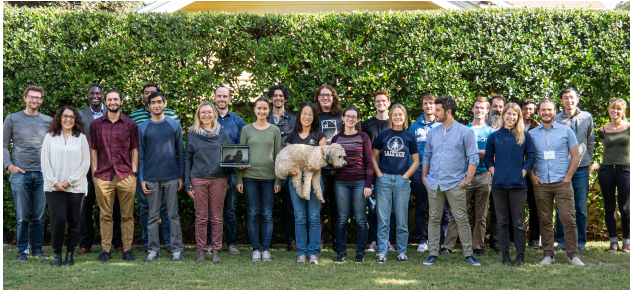
Coupler Overview

5 Recent and Future Projects

6 Conclusions

About CliMA

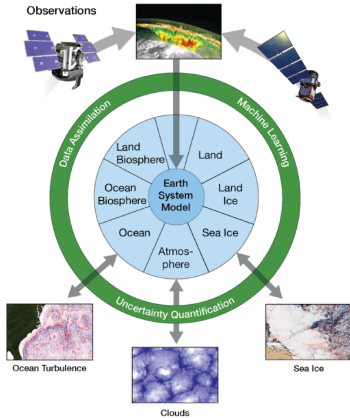
The Climate Modeling Alliance (CliMA) is a coalition of scientists, engineers, and applied mathematicians from **Caltech**, **MIT**, and the **NASA Jet Propulsion Laboratory**, who is building the first Earth System Model (ESM) in the Julia programming language that automatically learns from diverse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



Thanks to all team members: Tapio Schneider¹ (PI), Paul Ullrich², Oswald Knoth³, Simon Byrne¹, Jake Bolewski¹, Charles Kawczynski¹, Sriharsha Kandala¹, Gabriele Bozzola¹, Zhaoyi Shen¹, Jia He¹, Kiran Pamnany¹, Ben Mackay¹, Akshay Sridhar¹, Dennis Yatunin¹, Lenka Novak¹, Toby Bischoff¹, Daniel (Zhengyu)Huang¹, Andre Souza⁴, Yair Cohen¹

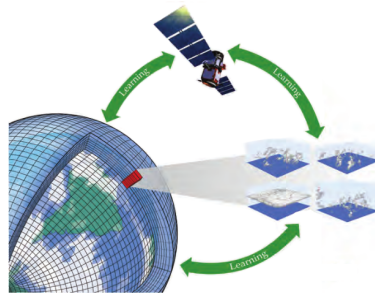
1: Caltech, 2: UC Davis, 3: TROPOS, 4: MIT

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.



ClimateMachine.jl: a first codebase

- Supports only Discontinuous Galerkin (DG) discretization. Same in each direction (horizontal/vertical) but allows different polynomial order. No staggered grids supported
- Can prescribe PDEs only in conservation form $\partial_t \mathbf{Q} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = \mathbf{S}(\mathbf{Q})$
- Operator volume/face kernels written in KernelAbstractions.jl (a unified programming model, similar to OpenCL/SYCL, which allows for single-source code for CPUs & GPUs, but primarily “a GPU code which runs on CPUs”)
- Overlaps computation & communication: distributed via MPI.jl.
- Efficient code, but somewhat inflexible API

A. Sridhar et al., *Large-eddy simulations with ClimateMachine v0.2.0: a new open-source code for atmospheric simulations on GPUs and CPUs*, Geoscientific Model Development, 15, 6259–6284 (2022)

A. Souza, J. He, T. Bischoff, M. Waruszewski, L. Novak, **V. Barra**, T. Gibson, A. Sridhar, S. Kandala, S. Byrne, L. Wilcox, J. Kozdon, F. Giraldo, O. Knuth, J. Marshall, R. Ferrari, T. Schneider, *The Flux-Differencing Discontinuous Galerkin Method Applied to an Idealized Fully Compressible Nonhydrostatic Dry Atmosphere*, JAMES, e2022MS003527, (2023)

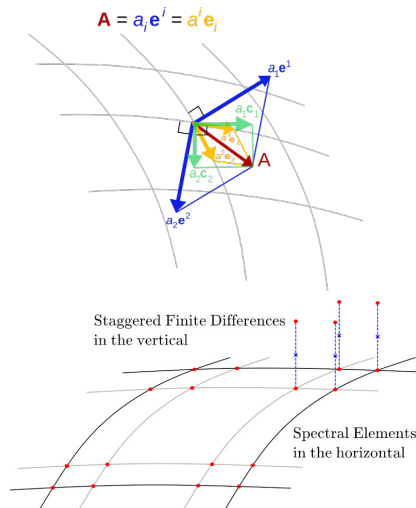
ClimaCore.jl



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).



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

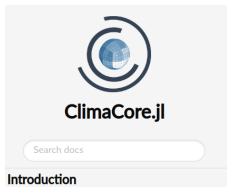
The dycore serves both the Land and Atmos model.
ClimaAtmos is the atmosphere component with:

- 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



Operators

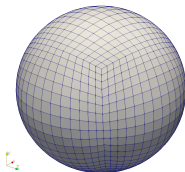
[Edit on GitHub](#) 

Operators

Operators can compute spatial derivative operations.

- for performance reasons, we need to be able to "fuse" multiple operators and function applications

- 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:

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

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

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

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

$$\frac{\partial h}{\partial t} + \frac{1}{J} \frac{\partial}{\partial \xi^j} (h J u^j) = 0 \quad (4a)$$

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

```

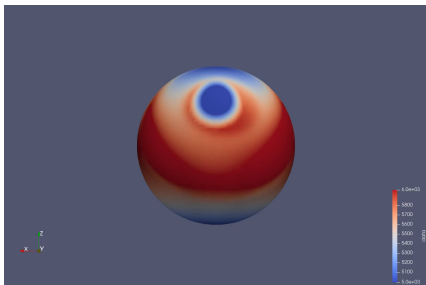
∇ = Operators.Gradient()
∇· = Operators.WeakDivergence()
curl = Operators.Curl()

@. 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

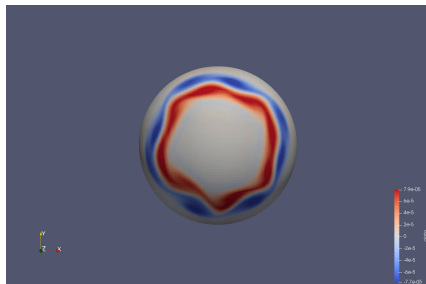
```

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

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \rho \mathbf{u}, \quad (5a)$$

$$\frac{\partial Q}{\partial t} = -\nabla \cdot Q \mathbf{u}, \quad (5b)$$

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) # continuity equation
@. dydt.pq = - ∇·(y.pq * u) # advection of tracer equation

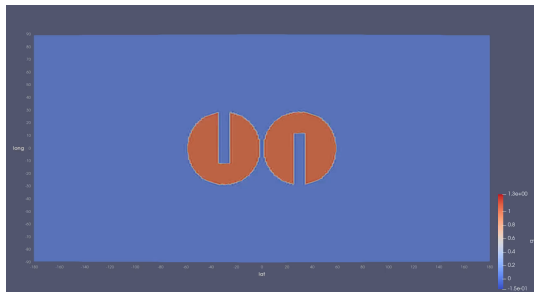
```

Quasimonotone flux limiters

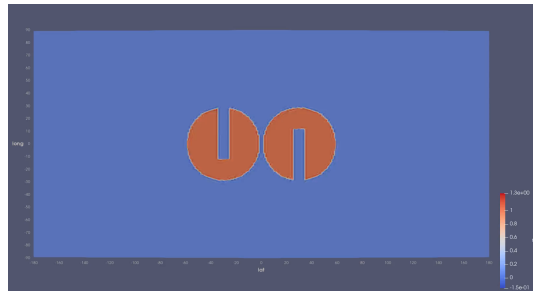
- Traditional SEM advection operator is oscillatory but due to its mimetic properties it is locally conservative and has a monotone property with respect to element averages
- We use a class of optimization-based locally conservative quasimonotone (monotone with respect to the spectral element nodal 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.)



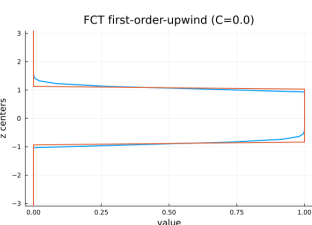
No limiter.



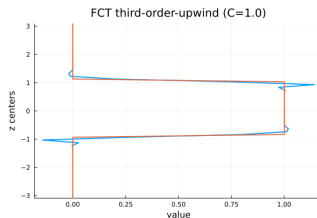
With limiter.

Flux-Corrected Transport

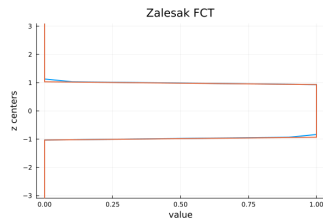
For the advection operator discretized by Finite Differences, the Flux-corrected transport (FCT) approximates with a high-order scheme in regions where the solution is smooth, and low-order monotone scheme where the solution is poorly resolved or discontinuous [Zalesak1979].



Dissipative and very dispersive



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

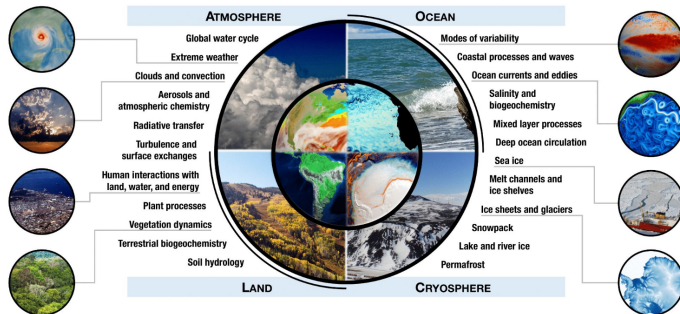


No over/under-shoots and limited dispersion.

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

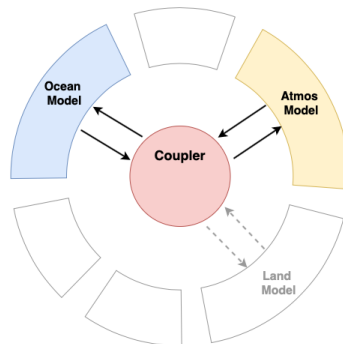
ESM

Background - Earth System Models



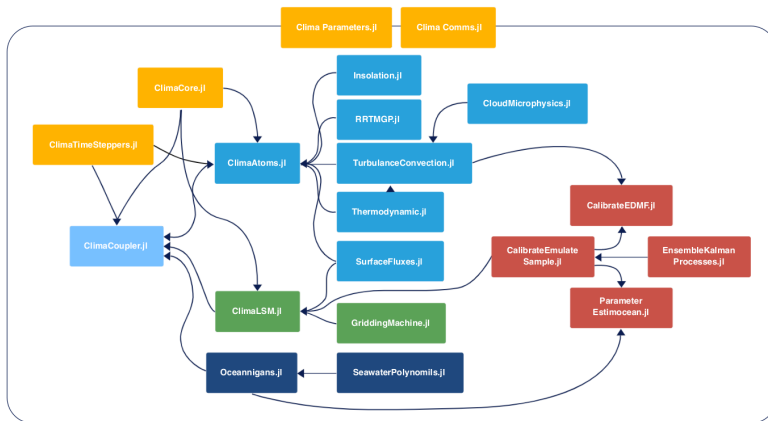
Source: Paul Ullrich, Dept. of Energy Office of Science
energy.gov/science/doe-explainearth-system-and-climate-models

- CliMA's ocean model: Oceananigans.jl
- CliMA's atmos model: ClimaAtmos.jl



[Source: ClimaCoupler.jl docs]

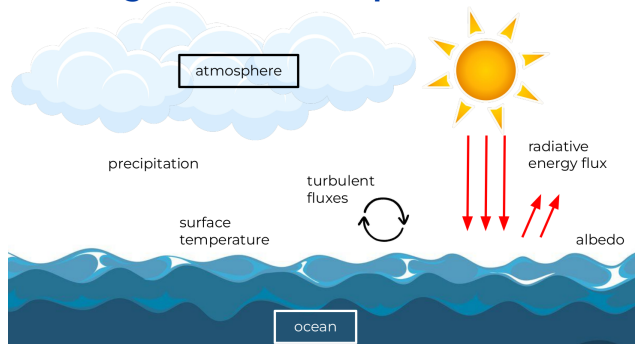
ClimaCoupler.jl in the CliMA ESM ecosystem



[Source: courtesy of Lenka Novak (CliMA, Caltech)]

Exchange quantities between components

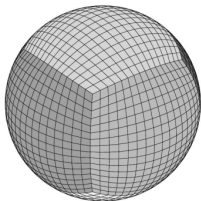
Exchange between components



[Source: courtesy of Julia Sloan (CliMA, Caltech)]

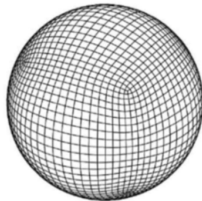
Regridding/Remapping and Time-stepping

Cubed sphere types:



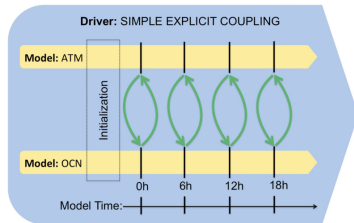
equiangular
cubed sphere

Ullrich 2014



conformal
cubed sphere

Rančić et al. 1996



→ Sequential or concurrent

```

===
step_model_sims!(model_sims, t)

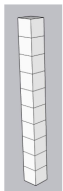
Iterates 'step!' over all component model simulations saved in `cs.model_sims`.
===
function step_model_sims!(model_sims, t)
    for sim in model_sims
        step!(sim, t)
    end
end
end

```

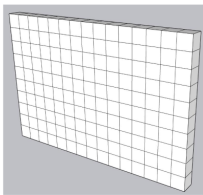
Process-based Hierarchy: e.g., Geometry Hierarchies

The process-based hierarchy spectrum allows simplification or isolation of an individual process within the model, for in-depth investigations into model's sensitivity to various design choices.

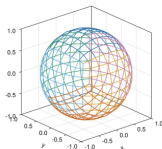
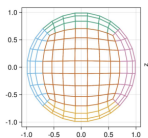
Domain visualizations



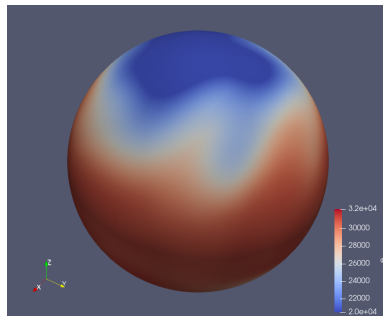
1D column



2D plane



3D cubed sphere

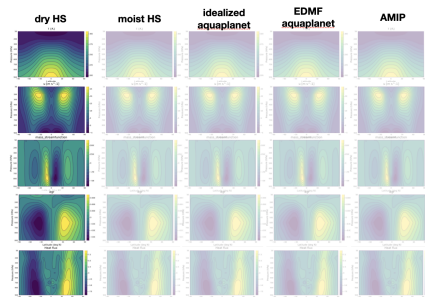
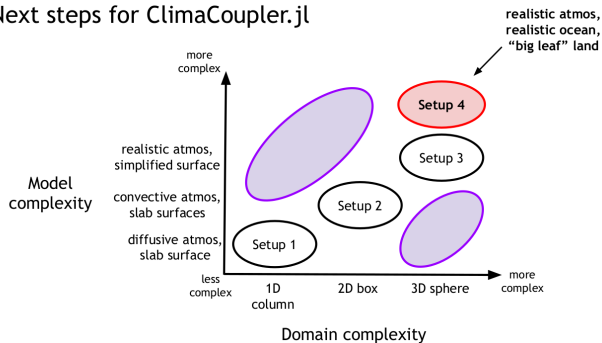


Held-Suarez 180-days simulation.

Generality-based Hierarchy: Model Hierarchies

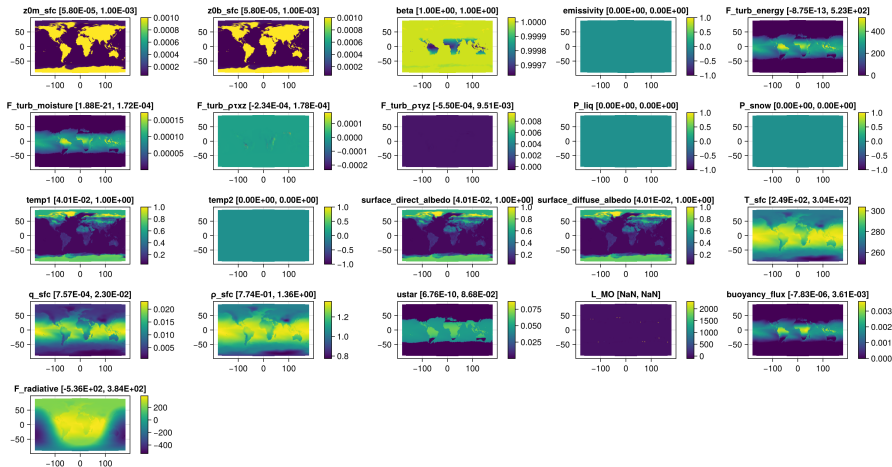
The generality-based hierarchy spectrum tests the universality of atmospheric behaviour in a set of simple to complex models, each of which represents Earth-like atmospheres in its own right.

Next steps for ClimaCoupler.jl



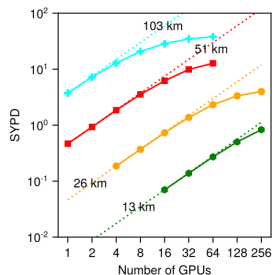
Preliminary climates of the five constituents of the generality-based hierarchy.

Quality control, testing, V & V: Continuous Integration

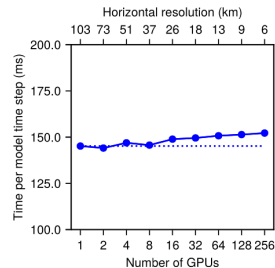
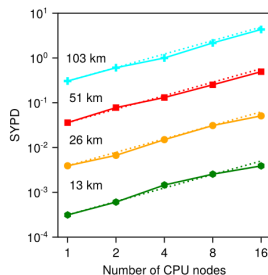


Continuous integration test, for AMIP with target resolution (~ 26 km in the horizontal & 43 vertical levels), topography and diagnostic EDMF.

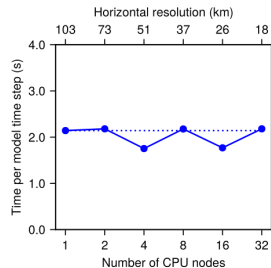
Scaling studies for the Atmospheric component



Strong scaling study



Weak scaling study



Recent and Future Projects

1 Introduction

2 Interfacial flows

3 libCEED

4 CliMA

ClimaCore.jl

Examples for Climate Applications

ClimaCoupler.jl

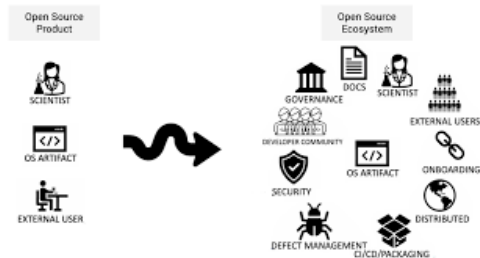
Coupler Overview

5 Recent and Future Projects

6 Conclusions

Open-source software ecosystems

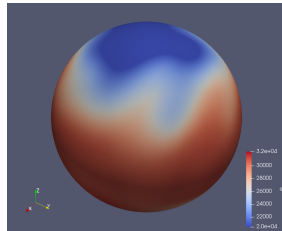
- Recent NSF POSE award to transition the Mimetic Operators Library Enhanced (MOLE) library into an open-source ecosystem.
- Need to apply best practices in community scientific software
 - Add documentation, tutorials, enhance examples
 - Quality of software: best software design patterns and engineering principles, test-driven design, reproducibility, code coverage, unit testing, Continuous Integration (CI) / Continuous Deployment (CD), portability



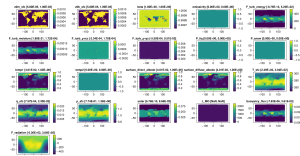
Conclusions and Future Directions

- Introduced libCEED for high-performance operator evaluations on heterogeneous architectures
- 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
 - Introduced ClimaCoupler.jl for flexible coupled model hierarchies

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] 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
- [2] 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.
- [3] 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.
- [4] Oksana Guba, Mark Taylor, Amik St-Cyr, *Optimization-based limiters for the spectral element method*, Journal of Computational Physics, 267, 176–195, 2014.
- [5] 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.
- [6] Steven T Zalesak, *Fully multidimensional flux-corrected transport algorithms for fluids*, Journal of computational physics, 31(3), 335–362, 1979.