Efficient numerical simulations for fluid dynamics across different scales

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

San Diego State University

December 6th, 2024

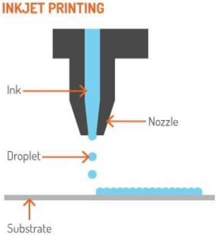
Overview

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

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 $\boldsymbol{\sigma}$
- creep:
  constant $\boldsymbol{\sigma} \Rightarrow$ increasing ϵ

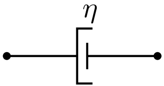
Mechanical system analogs

- Hookean: elastic solids

$\sigma_{ij} = 2G\epsilon_{ij}$
G shear elastic modulus

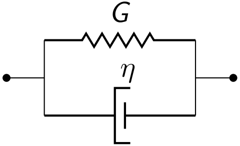
- Newtonian: viscous fluids

$\sigma_{ij} = 2\eta\dot{\epsilon}_{ij}$
η dynamic (shear) viscosity



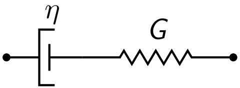
- Kelvin-Voigt: linear viscoelastic solids

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



- Maxwell: linear viscoelastic fluids

$\sigma_{ij} + \lambda_1 \partial_t \sigma_{ij} = 2\eta\dot{\epsilon}_{ij}$
λ_1 relaxation time, s. t. $\lambda_1 = \eta/G$.



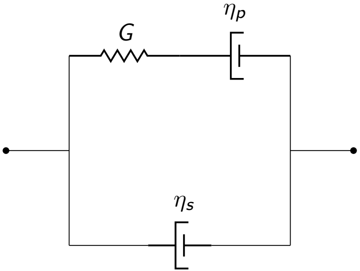
Mechanical system analog for Jeffreys

Jeffreys Model: linear viscoelastic fluids

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

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.



Governing equations

Conservation laws for incompressible fluids:

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

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

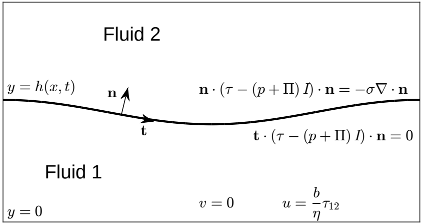
where, in $2D$, $\mathbf{u} = (u(x,y,t), v(x,y,t))$, is the vector velocity field, $\nabla = (\partial_x, \partial_y)$, p is the pressure, Π is the disjoining pressure due to the van-der-Waals interaction (attraction/repulsion) force, and $\mathbf{F_b} = (\rho g \sin\alpha, -\rho g \cos\alpha)$ body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

Jeffreys' model:

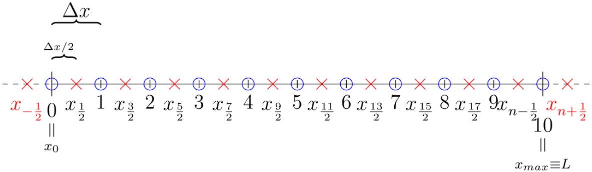
$$\boldsymbol{\sigma} + \lambda_1 \partial_t \boldsymbol{\sigma} = 2\eta(\dot{\boldsymbol{\epsilon}} + \lambda_2 \partial_t \dot{\boldsymbol{\epsilon}})$$

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



Dimensionless governing equations

Long-wave approximation in two spatial dimensions:

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

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

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

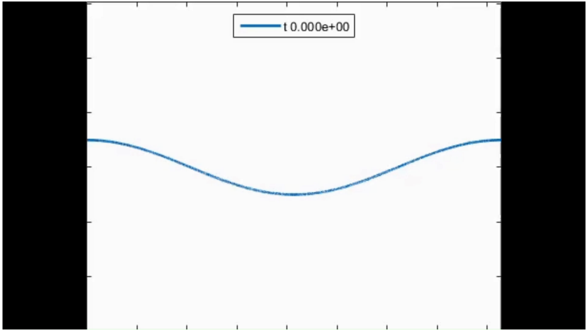
disjoining pressure: $\Pi(h) = \frac{\gamma(1-\cos\theta_e)}{Mh_\star}\left[\left(\frac{h_\star}{h}\right)^n - \left(\frac{h_\star}{h}\right)^m\right]$,

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

Jeffreys' constitutive law:

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

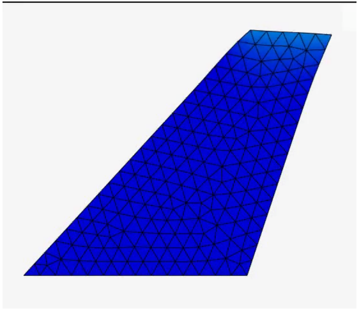
Dewetting film

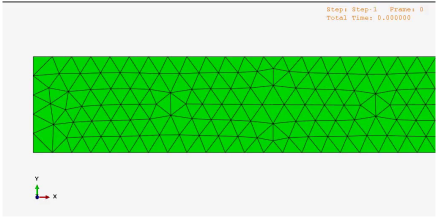


A viscoelastic dewetting film exhibits secondary satellite droplets in the dewetting region that viscous films do not exhibit.

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)

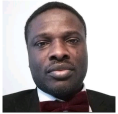
Publications

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)

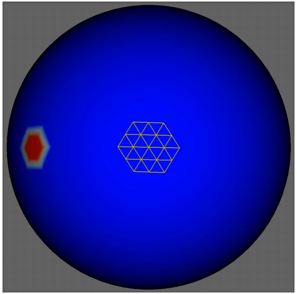
B. Adeyemi, P. Jadhawar, L. Akanji, **V. Barra**, *Effects of fluid–fluid interfacial properties on the dynamics of bounded viscoelastic thin liquid films*, J. Non-Newt. Fluid Mech., 309, 104893 (2022)

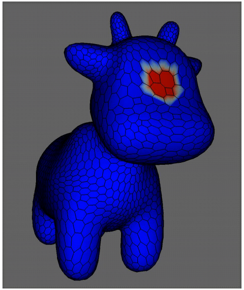


Other projects: Computer Graphics applications

PIXAR

Developed 2 proprietary libraries in C++ for viscous fluid simulations on curved surfaces

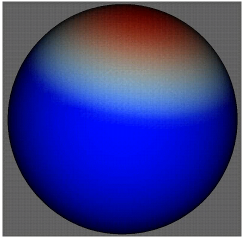


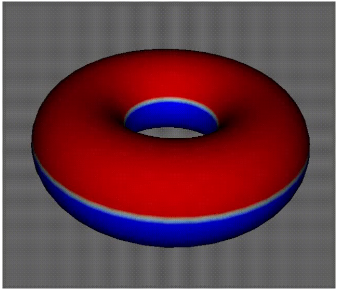


A vorticity-formulation 2D Navier-Stokes solver with fluid-structure interactions, using Discrete Exterior Calculus (DEC) in a finite volume discretization

Other projects (cont'ed)

A thin film (long-wave) solver on curved surfaces, with arbitrary topology and element shapes





Prototyped a plugin for the 3D graphics software package Houdini

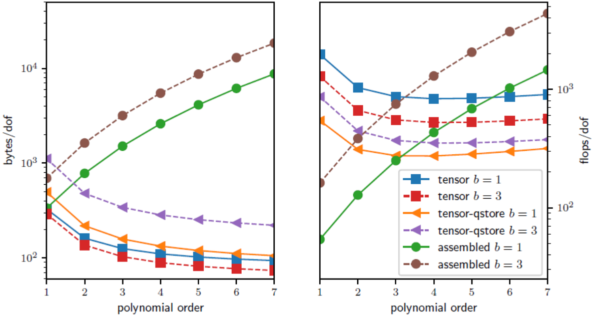
Overview

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

Postdoc project supervised by Jed Brown at University of Colorado Boulder

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

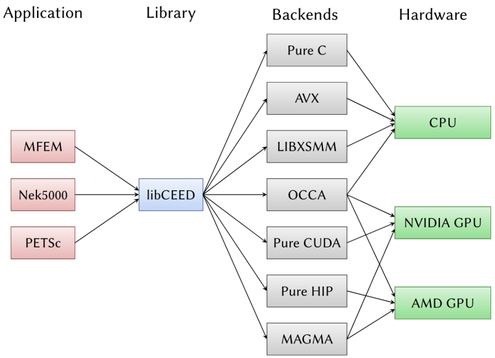


[Courtesy: Jed Brown]

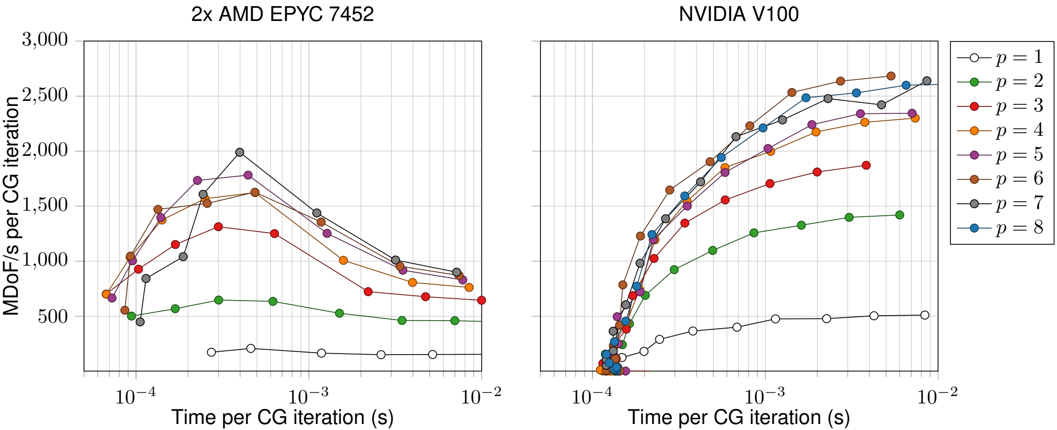
[https://github.com/CEED/libCEED/]

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

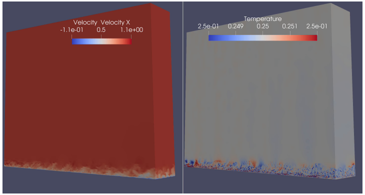


Performance

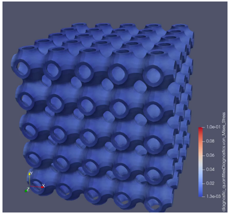


Application examples

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)

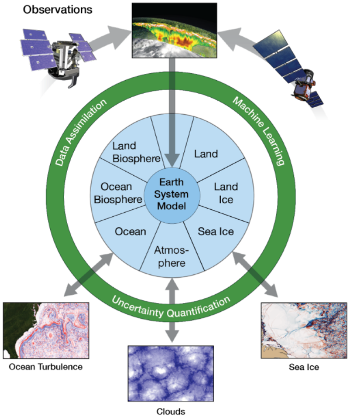
V. Barra, J. Brown, J. Thompson, Y. Dudouit, *High-performance operator evaluations with ease of use: libCEED's Python interface*, Proceedings of the 19th Python in Science Conference 85 - 90 (2020)

Overview

About CliMA

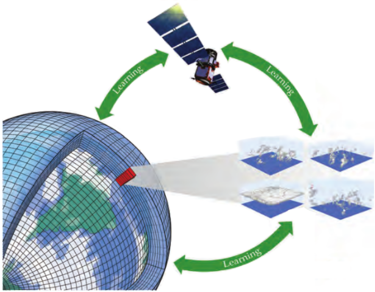
The Climate Modeling Alliance (CliMA) is a coalition of scientists, engineers, and applied mathematicians from **Caltech**, **MIT**, and the **NASA Jet Propulsion Laboratory**, who is building the first Earth System Model (ESM) in the Julia programming language that automatically learns from diverse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

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 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.

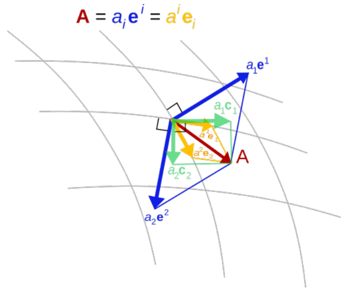
- For the atmosphere model, support both Large-Eddy Simulation (LES) and General Circulation Model (GCM) configurations (i.e., Cartesian and spherical geometries).
- Allow specification of any governing equations and boundary conditions by composing operators.
- Support non-uniform unstructured meshes.

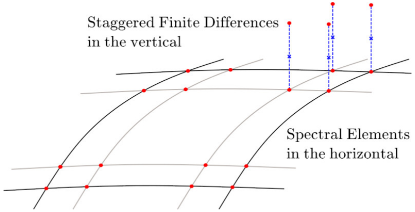
ClimaCore.jl

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

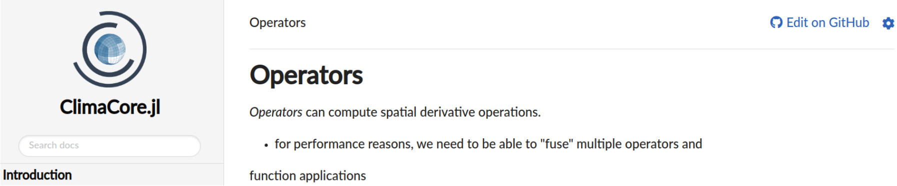
A library (suite of tools) for constructing flexible space discretizations.

- 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).



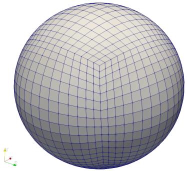


Some personal contributions



I have worked in adding support for

- 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:

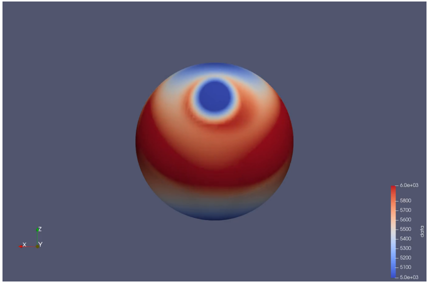
$$\frac{\partial h}{\partial t} + \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(hJu^j\right) = 0 \tag{4a}$$

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

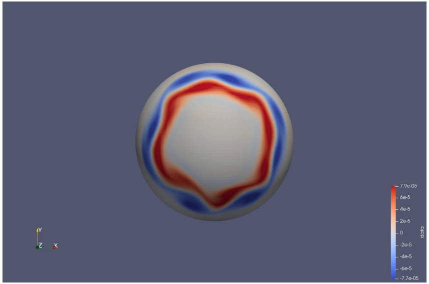
```
∇ = Operators.Gradient()
∇⋅ = Operators.WeakDivergence()
curl = Operators.Curl()

@. 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.

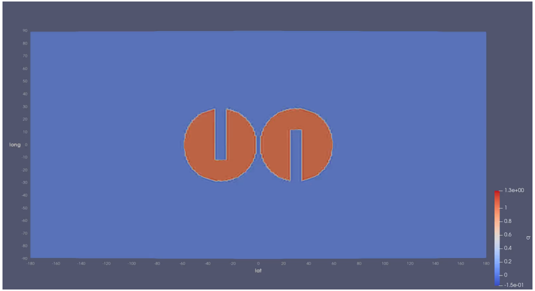
```
∇⋅ = Operators.WeakDivergence()

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

Quasimonotone flux limiters

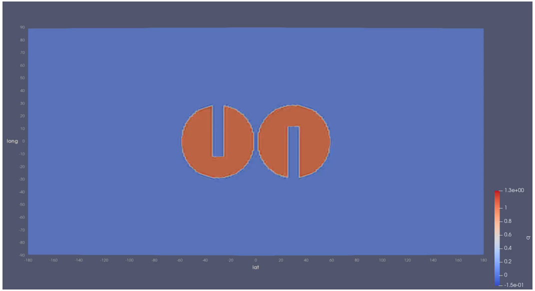
- Traditional SEM advection operator is oscillatory but due to its mimetic properties it is locally conservative and has a monotone property with respect to element averages
- We use a class of optimization-based locally conservative quasimonotone (monotone with respect to the spectral element nodal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



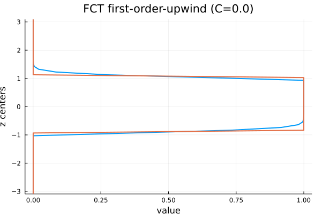
No limiter.



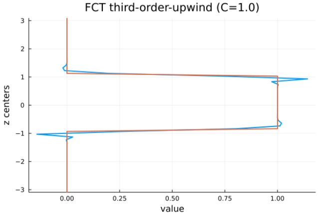
With limiter.

Flux-Corrected Transport

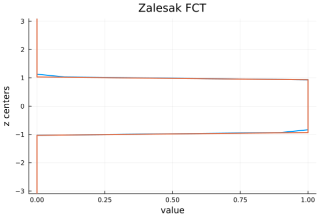
For the advection operator discretized by Finite Differences, the Flux-corrected transport (FCT) approximates with a high-order scheme in regions where the solution is smooth, and low-order monotone scheme where the solution is poorly resolved or discontinuous [**Zalesak1979**].



Dissipative and very dispersive



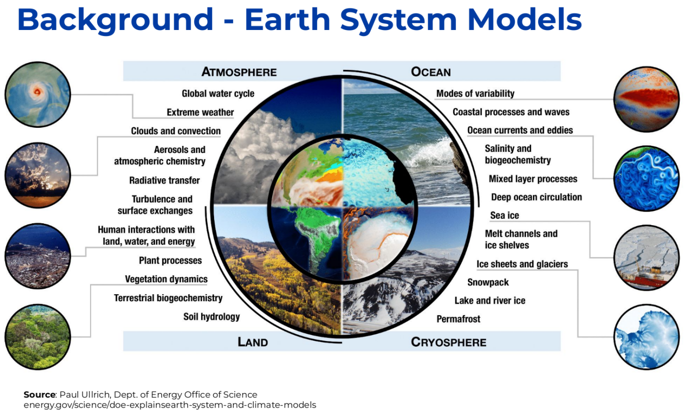
Less dissipative and less dispersive (but over/under-shoots)

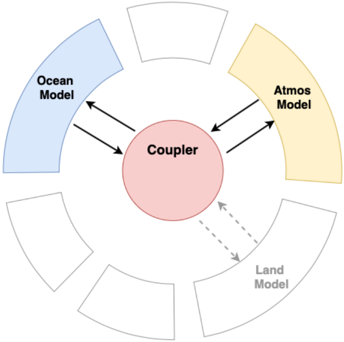


No over/under-shoots and limited dispersion.

D. Yatunin, S. Byrne, . . . , **V. Barra**, O. Knoth, P. Ullrich, T. Schneider, *The CliMA atmosphere dynamical core: Concepts, numerics, and scaling*, in preparation

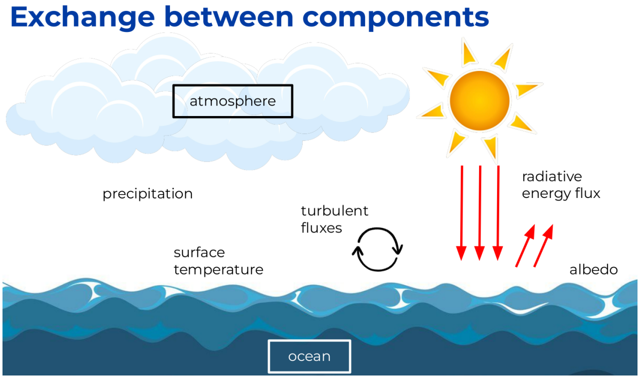
ESM





[Source: ClimaCoupler.jl docs]

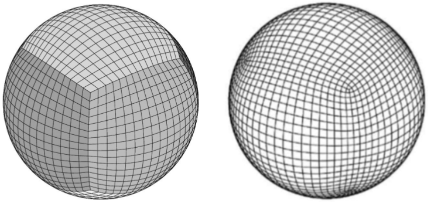
Exchange quantities between components



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

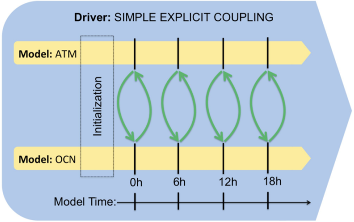
Regridding/Remapping and Time-stepping

Cubed sphere types:



equiangular cubed sphere — Ullrich 2014

conformal cubed sphere — Rančić et al. 1996

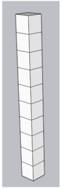


→ Sequential or concurrent

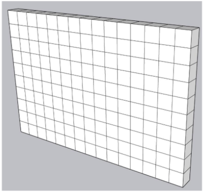
```
"""
    step_model_sims!(model_sims, t)

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
```

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

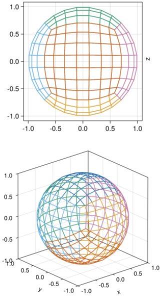
Domain visualizations



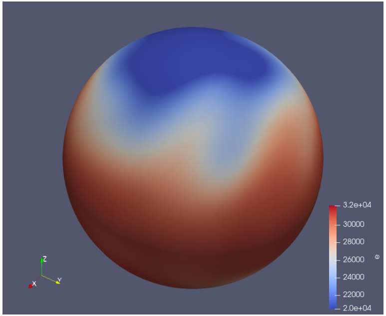
1D column



2D plane



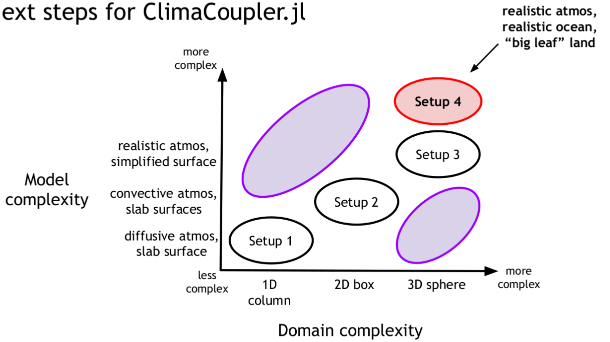
3D cubed sphere

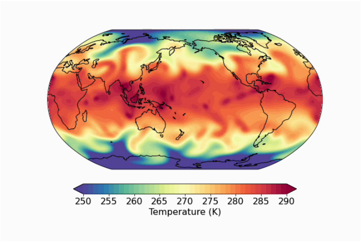


Held-Suarez 180-days simulation.

Generality-based Hierarchy: Model Hierarchies

Next steps for ClimaCoupler.jl





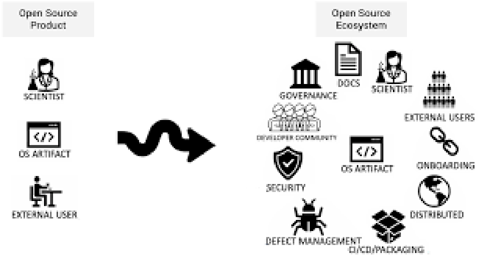
Preliminary AMIP (w/o EDMF and topography) simulation—temperature.

L. Novak, . . . , **V. Barra**, T. Schneider, *ClimaEarth: Earth System Model Hierarchies*, in preparation

Recent and Future Projects

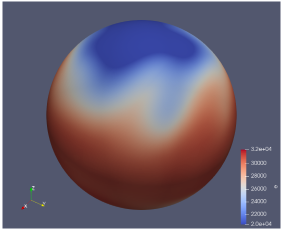
Open-source software ecosystems

- 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



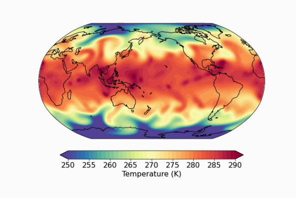
Conclusions and Future Directions

- Introduced libCEED for high-performance operator evaluations on heterogenous 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



[Held-Suarez 180-day simulation.]

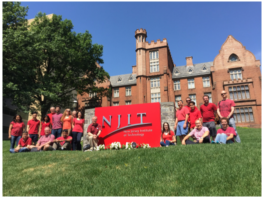
Future Directions:
Entropy-stable methods for combustion applications with Yiyue.



[AMIP (w/o EDMF and topography) simulation.]

Mentoring and Engagement

- Mentoring experience (GSMM Camp)
- Continue efforts in Diversity, Equity, Inclusion and Belonging
- Advocacy for (invisible/visible) disabilities in STEM, women in STEM, international students and scholars, upward mobility, systemic exclusion of underrepresented groups
- Active in industrial workshops (MPI)
- Students success and professional development: professional associations, e.g., US-RSE, WHPC, SIAM, etc



Thank you!

References

[1] 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.

[2] 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.

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

[4] 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.

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