Some Interesting Results from Element-based Galerkin Nonhydrostatic Atmospheric Models

ECMWF Annual Seminar, September 17, 2020

Frank Giraldo

Department of Applied Mathematics

Naval Postgraduate School

Monterey CA 93943

fxgirald@nps.edu

http://frankgiraldo.wix.com/mysite

Acknowledgements

- **GNuMe, NUMA, NUMO Model**
  - Funded by ONR, DARPA, AFOSR, NSF, DOE,
  - Collaborators: Felipe Alves, Sohail Reddy, and FXG (NPS); Jim Kelly (NRL-DC); Yassine Tissaoui and Simone Marras (NJIT)
  - See https://frankgiraldo.wixsite.com/mysite/numa
- **CLIMA-atmos Model**
  - Funded by NSF and private donors
  - Collaborators: Jeremy Kozdon, Lucas Wilcox, Maciej Waruszewski, and Thomas Gibson, and FXG (NPS numerics group); Toby Bischoff, Simon Byrne, Akshay Sridhar, Charlie Kawczynski, Lenka Novak, Zhaoyi Shen, Jia He, Kiran Pamnany, and Tapio Schneider (Caltech); Yassine Tissaoui and Simone Marras (NJIT); Valentin Churavy (MIT)
  - Open Source/Open Development
  - See https://github.com/CliMA/ClimateMachine.jl
  - A variety of kernels (1d, 2d, 3d for shallow water, Euler, Navier-Stokes) can be found here: https://github.com/fxgiraldo/Canary.jl/tree/compute-kernels/examples

Talk Summary

- Governing Equations
- Underlying Numerics
- Results with NUMA
- Results with CLIMA
- Summary

NUMA Governing Equations

- NUMA High-Altitude
  - No assumption on gas constants - can be spatially and temporally varying.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{u}\right) = 0$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla P + \nabla \phi + \vec{\Omega} \times \vec{u} = 0$$

$$\frac{\partial e}{\partial t} + \vec{u} \cdot \nabla e + (\gamma - 1) e \nabla \cdot \vec{u} = 0$$

$$P = (\gamma - 1)\rho e$$

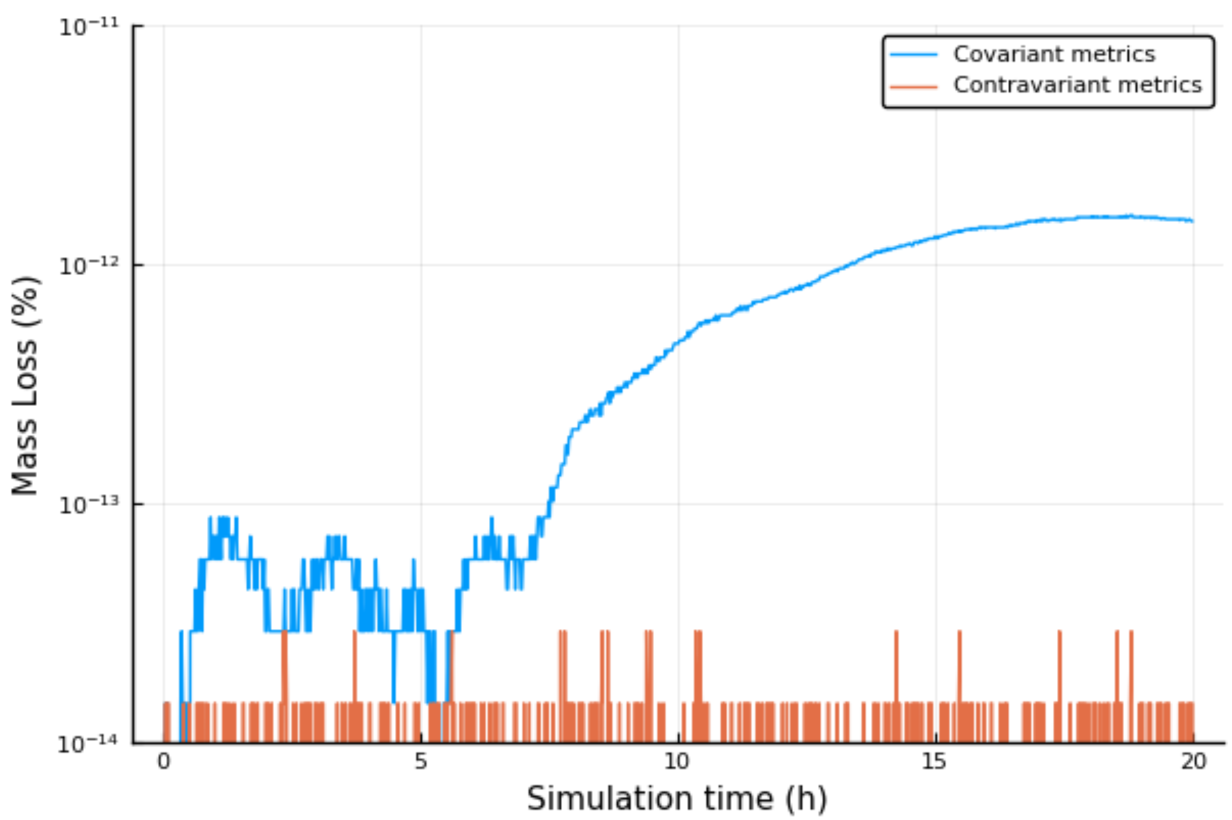
- CLIMA

$$\frac{\partial \vec{q}}{\partial t} + \nabla \cdot \vec{F} = S\left(\vec{q}\right) \qquad \vec{q} = \left(\rho, \vec{U}^T, E\right)^T$$

$$\vec{F} = \begin{pmatrix} \vec{U} \\ \dfrac{\vec{U} \times \vec{U}}{\rho} + P\vec{I}_3 \\ \dfrac{(E+P)\vec{U}}{\rho} \end{pmatrix} \qquad S\left(\vec{q}\right) = \begin{pmatrix} \vec{0} \\ -\rho \nabla \phi - \vec{\Omega} \times \vec{U} \\ \vec{0} \end{pmatrix} \qquad P = (\gamma - 1)\rho \left(c_v T - \frac{1}{2}\vec{u} \cdot \vec{u} - \varphi\right)$$

Governing Equations: Contravariant Form

- For Conservation Law $$\frac{\partial \vec{q}}{\partial t} + \nabla \cdot \vec{F} = \vec{0}$$
- with Cartesian coordinates $$\nabla = \frac{\partial}{\partial x}\vec{e}_x + \frac{\partial}{\partial y}\vec{e}_y + \frac{\partial}{\partial z}\vec{e}_z$$
- The contravariant form is $$\frac{\partial \vec{q}}{\partial t} + \frac{1}{J^{(e)}}\sum_{i=1}^{3}\frac{\partial}{\partial \xi_i}\left(J^{(e)}\vec{F}^{\xi_i}\right) = \vec{0}$$
- with contravariant vectors $$\vec{F}^{\xi_i} = \vec{F} \cdot \nabla \xi_i$$



Mass Conservation

Governing Equations: Contravariant Form

- For contravariant form
$$\frac{\partial \vec{q}}{\partial t} + \frac{1}{J^{(e)}} \sum_{i=1}^{3} \frac{\partial}{\partial \xi_i} \left(J^{(e)} \vec{F}^{\xi_i} \right) = \vec{0}$$

- We can easily decompose along grid directions as follows
$$\frac{\partial \vec{q}}{\partial t} + \frac{1}{J^{(e)}} \frac{\partial}{\partial \zeta} \left(J^{(e)} \vec{F}^{\zeta} \right) + R_H = \vec{0}$$

- with the horizontal component
$$R_H = \frac{1}{J^{(e)}} \sum_{i=1}^{2} \frac{\partial}{\partial \xi_i} \left(J^{(e)} \vec{F}^{\xi_i} \right)$$

This allows for a straightforward implementation of HEVI schemes regardless of the underlying coordinate system (e.g., Cartesian). This way we can build one code that handles both flow on a sphere (GCM) and in a box (mesoscale and LES).

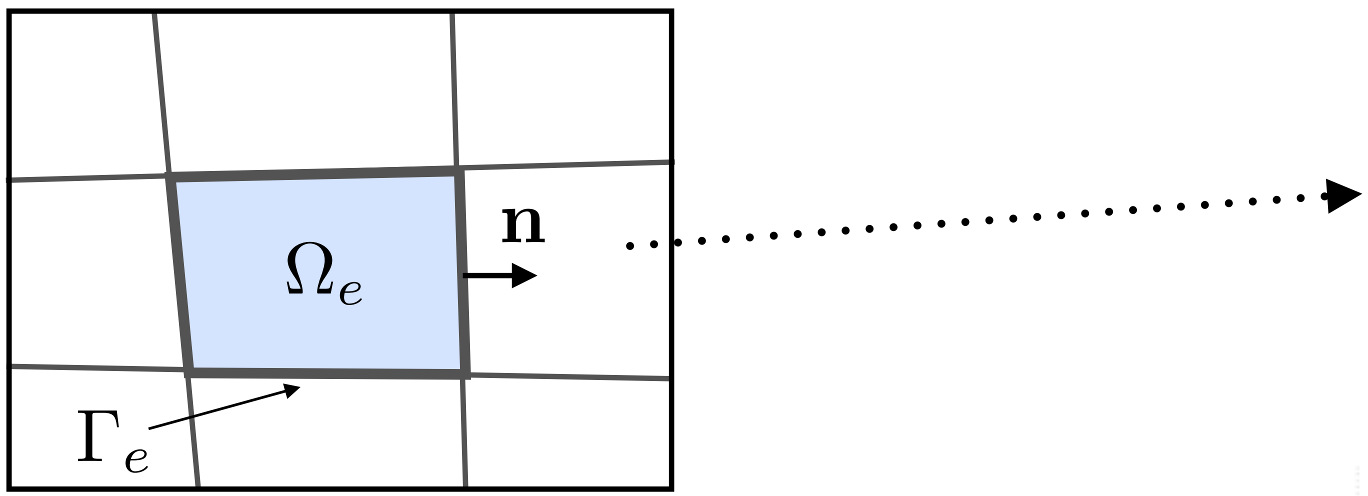
Talk Summary

- Governing Equations
- Underlying Numerics
- Results with NUMA
- Results with CLIMA
- Summary

Element-based Galerkin Methods*

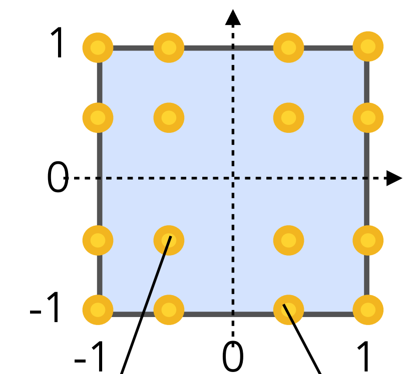
Domain decomposition

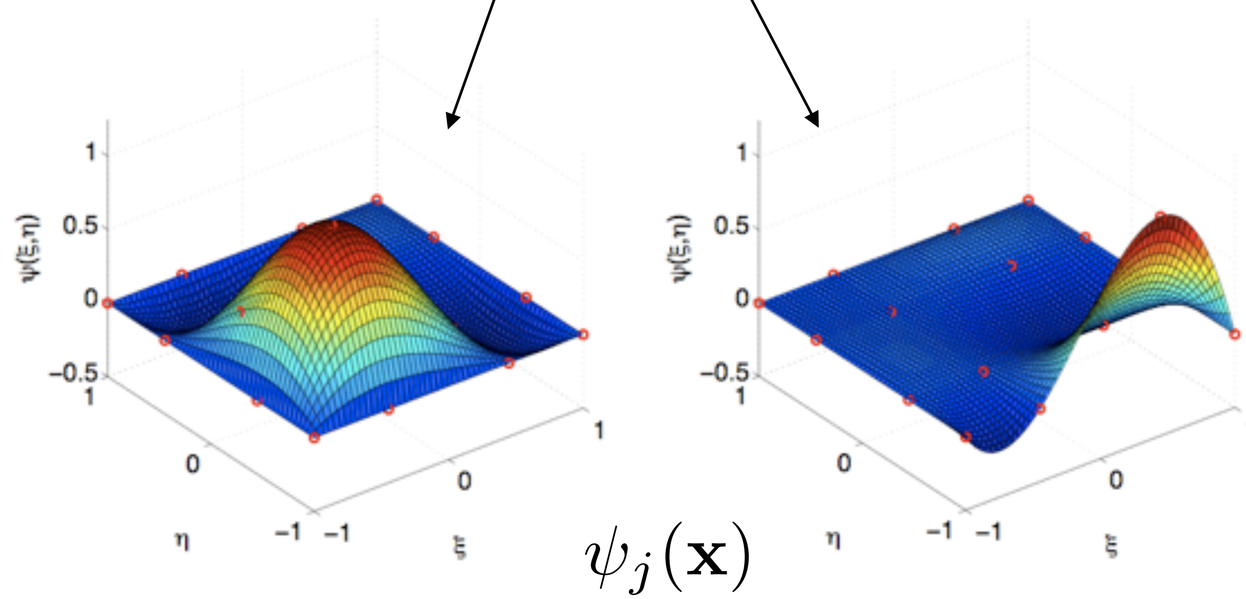
$$\Omega = \bigcup_{e=1}^{N_e} \Omega_e$$



Reference element

Legendre-Gauss-Lobatto points





Basis functions - Lagrange polynomials

Approximate local solution as:

$$\mathbf{q}_N^{(e)}(\mathbf{x},t) = \sum_{j=1}^{M} \psi_j(\mathbf{x})\mathbf{q}_j^{(e)}(t)$$

* F.X. Giraldo. An Introduction to Element-Based Galerkin Methods on Tensor-Product Bases: Analysis, Algorithms, and Applications, Springer Texts in Computational Science and Engineering, Series Volume 24, ISBN: 978-3-030-55068-4 (2020)

Continuous/discontinuous Galerkin methods

Governing equation: $$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = S(\mathbf{q})$$

Approximate the **global** solution as: $$\mathbf{q}_N(\mathbf{x},t) = \sum_{i=1}^{M_N} \psi_i(\mathbf{x}) \mathbf{q}_i(t)$$

$$\mathbf{F}_N = \mathbf{F}(\mathbf{q}_N)$$
$$S_N = S(\mathbf{q}_N)$$

Define residual: $$R(\mathbf{q}_N) \equiv \frac{\partial \mathbf{q}_N}{\partial t} + \nabla \cdot \mathbf{F}_N - S_N = \epsilon$$

Problem statement:

Find $\mathbf{q}_N \in \mathcal{S} \quad \forall \psi \in \mathcal{S}$ (test function)

$$\begin{cases} \mathcal{S}_{CG} = \{\psi \in H^1(\Omega) : \psi \in P_N(\Omega_e)\ \forall \Omega_e\} \\ \mathcal{S}_{DG} = \{\psi \in L^2(\Omega) : \psi \in P_N(\Omega_e)\ \forall \Omega_e\} \end{cases}$$

such that $$\int_{\Omega_e} \psi_i R(\mathbf{q}_N) \mathrm{d}\Omega_e = 0$$

Continuous/discontinuous Galerkin methods

Integral form:

$$\int_{\Omega_e} \psi_i R(\mathbf{q}_N^{(e)}) \mathrm{d}\Omega_e = 0$$

$$\int_{\Omega_e} \psi_i \frac{\partial \mathbf{q}_N^{(e)}}{\partial t} \mathrm{d}\Omega_e + \int_{\Omega_e} \psi_i \nabla \cdot \mathbf{F}_N^{(e)} \mathrm{d}\Omega_e - \int_{\Omega_e} \psi_i S_N^{(e)} \mathrm{d}\Omega_e = 0$$

Integration by parts: $\int_{\Omega_e} \psi_i \nabla \cdot \mathbf{F}_N^{(e)} \mathrm{d}\Omega_e = \int_{\Omega_e} \nabla \cdot (\psi_i \mathbf{F}_N^{(e)}) \mathrm{d}\Gamma_e - \int_{\Omega_e} \nabla \psi_i \cdot \mathbf{F}_N^{(e)} \mathrm{d}\Omega_e = \int_{\Gamma_e} \mathbf{n} \cdot \psi_i \mathbf{F}_N^{(e)} \mathrm{d}\Gamma_e - \int_{\Omega_e} \nabla \psi_i \cdot \mathbf{F}_N^{(e)} \mathrm{d}\Omega_e$

$$\int_{\Omega_e} \psi_i \frac{\partial \mathbf{q}_N^{(e)}}{\partial t} \mathrm{d}\Omega_e + \underbrace{\int_{\Gamma_e} \mathbf{n} \cdot \psi_i \mathbf{F}_N^{(e)} \mathrm{d}\Gamma_e}_{\text{face integral}} \underbrace{- \int_{\Omega_e} \nabla \psi_i \cdot \mathbf{F}_N^{(e)} \mathrm{d}\Omega_e - \int_{\Omega_e} \psi_i S_N^{(e)} \mathrm{d}\Omega_e}_{\text{volume integrals}} = 0$$

Matrix form:

$$M_{ij}^{(e)} \frac{\mathrm{d}\mathbf{q}_j^{(e)}}{\mathrm{d}t} + \sum_{f=1}^{N_{faces}} \left(\mathbf{M}_{ij}^{(e,f)}\right)^T \mathbf{F}_j^{(e,f,*)} - \left(\tilde{\mathbf{D}}_{ij}^{(e)}\right)^T \mathbf{F}_{ij}^{(e)} - S_i^{(e)} = 0$$

$M_{ij}^{(e)} = \int_{\Omega_e} \psi_i \psi_j \mathrm{d}\Omega_e$ — mass matrix

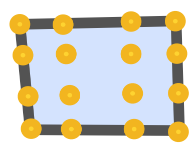
$\mathbf{M}_{ij}^{(e,f)} = \int_{\Gamma_e} \psi_i \psi_j \mathbf{n}^{(e,f)} \mathrm{d}\Gamma_e$ — face mass matrix

$\tilde{\mathbf{D}}_{ij}^{(e)} = \int_{\Omega_e} \nabla \psi_i \psi_j \mathrm{d}\Omega_e$ — differentiation matrix

Unified CG/DG methods

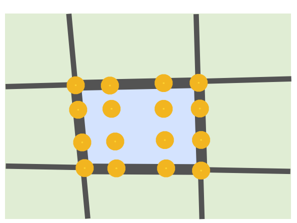
$$\int_{\Omega_e} \psi_i \frac{\partial \mathbf{q}_N^{(e)}}{\partial t} \mathrm{d}\Omega_e + \int_{\Gamma_e} \mathbf{n} \cdot \psi_i \mathbf{F}_N^{(e)} \mathrm{d}\Gamma_e - \int_{\Omega_e} \nabla\psi_i \cdot \mathbf{F}_N^{(e)} \mathrm{d}\Omega_e - \int_{\Omega_e} \psi_i S_N^{(e)} \mathrm{d}\Omega_e = 0$$

1. Evaluate “volume” integrals on element interiors



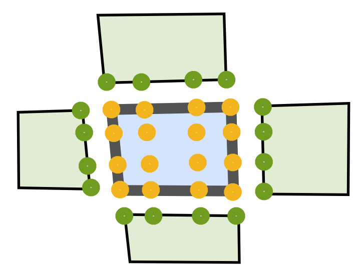
$$R^{(e)} := \int_{\Omega_e} \nabla\psi_i \cdot \mathbf{F}_N^{(e)} \mathrm{d}\Omega_e + \int_{\Omega_e} \psi_i S_N^{(e)} \mathrm{d}\Omega_e$$

2. CG: direct stiffness summation



$$R := \widehat{R^{(e)}}$$

2. DG: evaluate flux integrals



$$R := R^{(e)} - \int_{\Gamma_e} \mathbf{n} \cdot \nabla\psi_i \mathbf{F}_N^{(e)} \mathrm{d}\Gamma_e$$

3. Divide by mass matrix and time step

$$\frac{\mathrm{d}\mathbf{q}_i}{\mathrm{d}t} = M_{ij}^{-1} R_j$$

- NUMA carries three types of storage: CGc, CGd, and DG.
- CLIMA only carries DG.

Talk Summary

- Governing Equations
- Underlying Numerics
- Results with NUMA
- Results with CLIMA
- Summary

NUMA Update

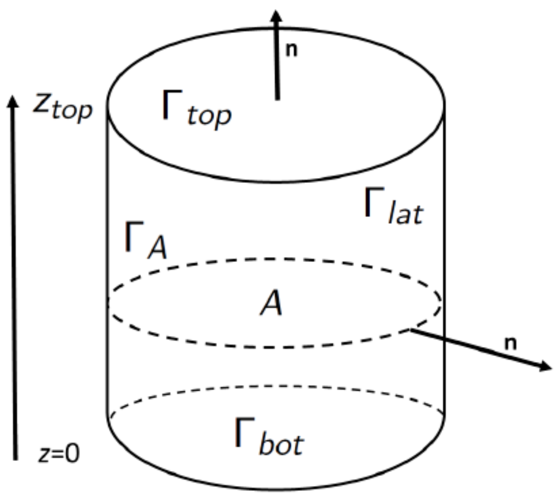
- NUMA-G is on course to go operational inside of U.S. Navy's NEPTUNE system
- NUMA-HA* and NUMA-LES are our current focus:
  - global model requirements are different from LES model requirements (e.g., time-integration strategy), as are stratospheric vs thermospheric model requirements (e.g., time-integration).
  - Many tools currently in NUMA can be leveraged to answer interesting questions. E.g., (all results shown with equation set 4NC)
    1. What do we need to change in earth weather model for use in high-altitude simulations (~500 km)? (Result 1).
    2. What is the role of high-order accuracy (Taylor-Green Vortex) (Result 2).
    3. How does positivity-preservation behave with AMR EBG methods (Result 3)

* NUMA-HA is a prototype for SEPHIR: Space Environment Prediction with High Resolution, a DARPA-funded project.

Result 1: NUMA-HA and Mass Permeable BC*

- Pressure-level hydrostatic models use an elastic BC which allows "breathing", so we want the mass permeable BC to mimic these hydrostatic BCs in a nonhydrostatic model. In contrast, rigid BCs and/or sponges do not allow expansion/contraction.

- Design a mass permeable BC that allows the atmosphere to expand/contract (i.e. "breathe') during heating/cooling.



$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0$$

Let $M(z_0, z_1) = A \int_{z_0}^{z_1} \rho dz$

Integrate mass

$$\frac{\partial M(z_s, z_t)}{\partial t} + \int_\Gamma \vec{n} \cdot (\rho \vec{u})\, d\Gamma = 0$$

$$\int_\Gamma \vec{n} \cdot (\rho \vec{u})\, d\Gamma = \int_{\Gamma_{bot}} \vec{n} \cdot (\rho \vec{u})\, d\Gamma + \int_{\Gamma_{top}} \vec{n} \cdot (\rho \vec{u})\, d\Gamma + \int_{\Gamma_{lat}} \vec{n} \cdot (\rho \vec{u})\, d\Gamma$$

Assume hydrostatic columns

$$M(z_s, z_t) = M(z_s, \infty) - M(z_t, \infty) = \frac{A}{g}(p_s - p_t)$$

Mass becomes

$$\frac{A}{g}\left(\frac{\partial p_s}{\partial t} - \frac{\partial p_t}{\partial t}\right) + 0 + A\rho_t w_t + A \int_{z_s}^{z_t} \nabla \cdot (\rho \vec{u}_H)\, dz = 0$$

wtop

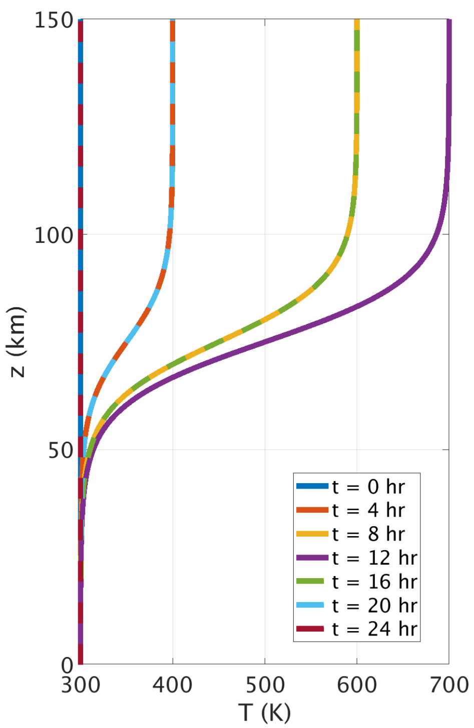
$$w_t = -\frac{1}{\rho_t g}\left(\frac{\partial p_s}{\partial t} - \frac{\partial p_t}{\partial t}\right) - \frac{1}{\rho_t} \int_{z_s}^{z_t} \nabla \cdot (\rho \vec{u}_H)\, dz$$

* Courtesy of Jim Kelly (NRL-DC) and Sohail Reddy (NPS)

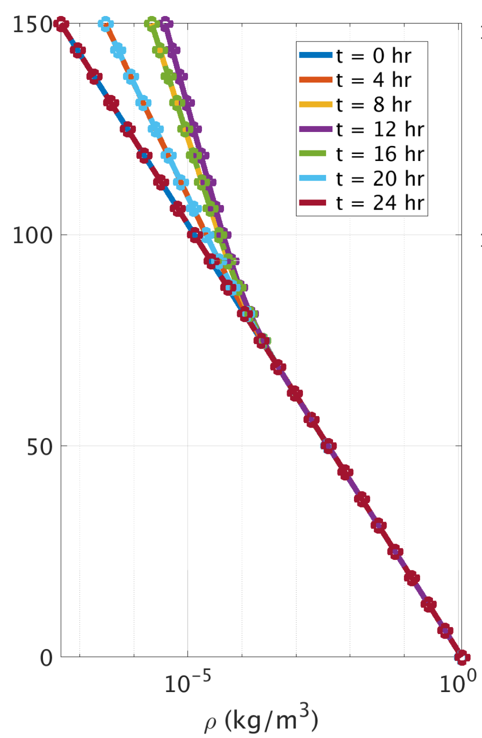
Result 1: NUMA-HA and Mass Permeable BC

Experiment setup: NUMA is run in 1D Column model,
Vertical grid: 60 spectral elements with order p=4 w/ ztop = 150km
Initially balanced isothermal atmosphere (T_0=300 K) is heated to a thermosphere-like atmosphere above ~75km

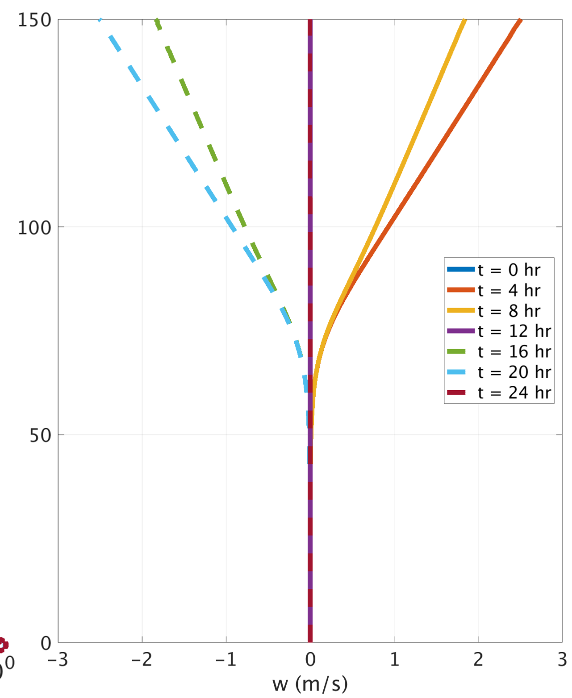
1D: $w_t = T_t(t)\int_{z_s}^{z_t} \frac{1}{T^2(y,t)}\frac{\partial T(y,t)}{\partial t}dy$ $\quad T(z,t) = a(t)T_0 + (1-a(t))T_{ref}(z)$ $\quad a(t) = \frac{1}{2}(1+\cos(2\pi\alpha t))$



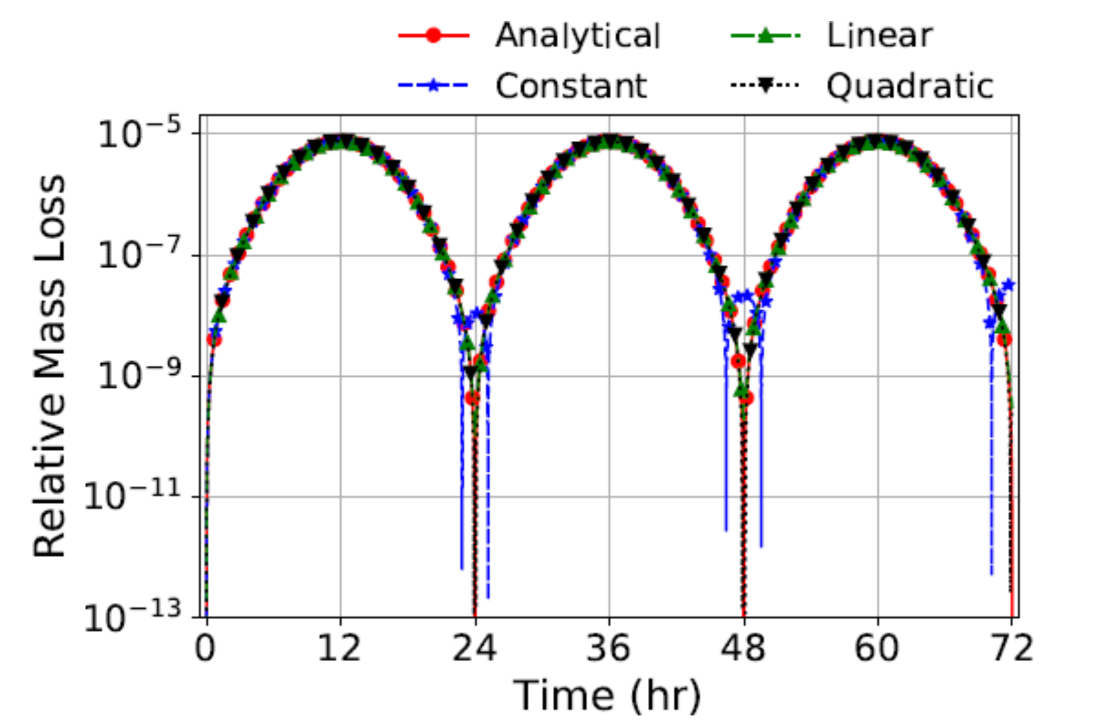
temperature



density



velocity

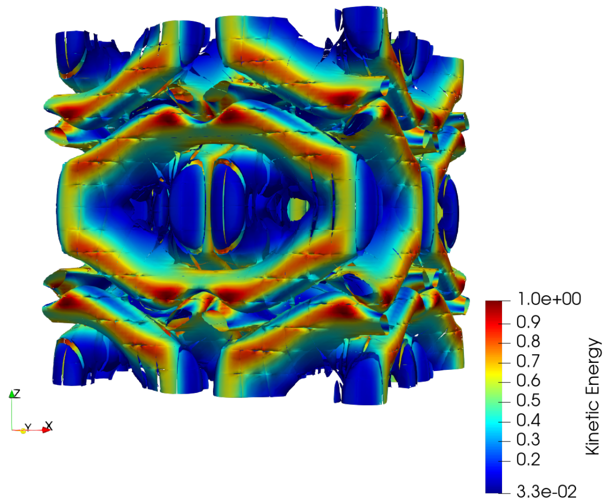


mass loss with extrapolated rho bc

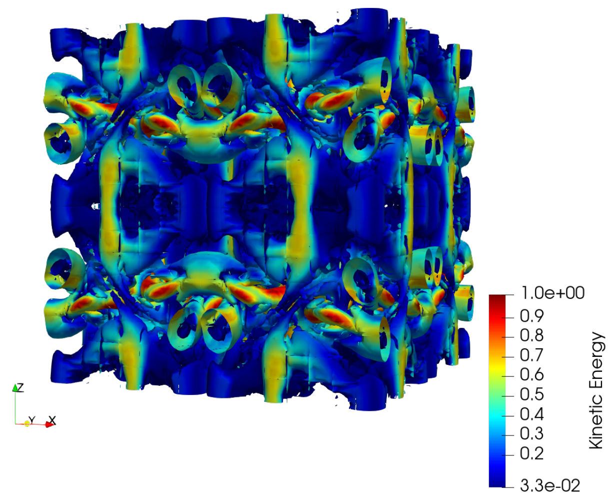
First 12 hours: heating, so top BC is an outlet BC. Only the vertical velocity is specified.
Hours 12-24: cooling, so top BC is an inlet BC. Temperature, density, and vertical velocity are all specified; density must be extrapolated from the interior.
NUMA's mass loss/gain is compared to the analytical mass loss. Quadratic extrapolation provides the most accurate mass loss/gain.

Result 2: NUMA-LES Taylor-Green Vortex*

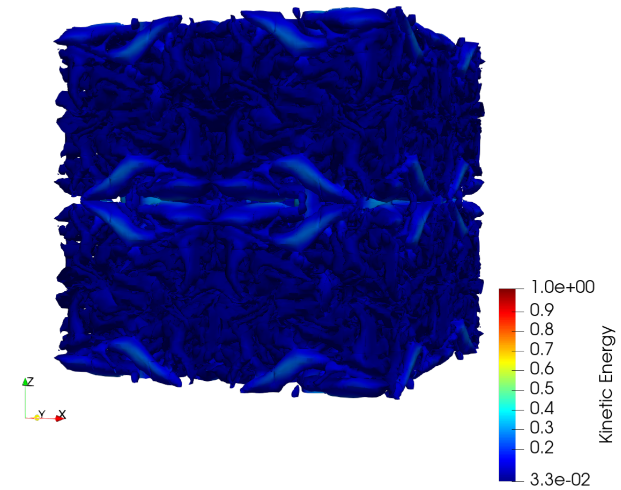
t* = 4



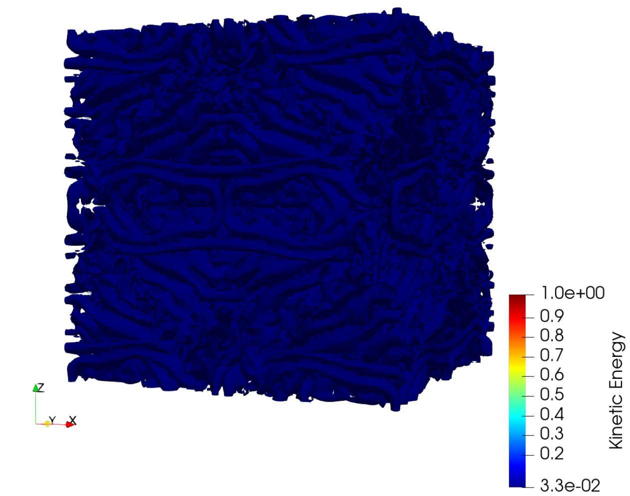
t* = 8

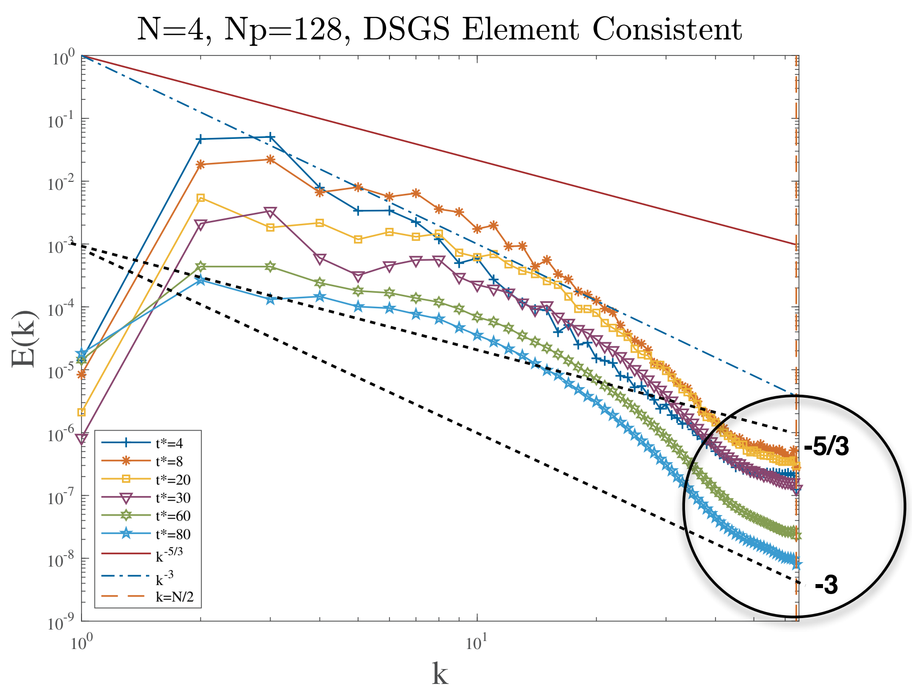


t* = 20

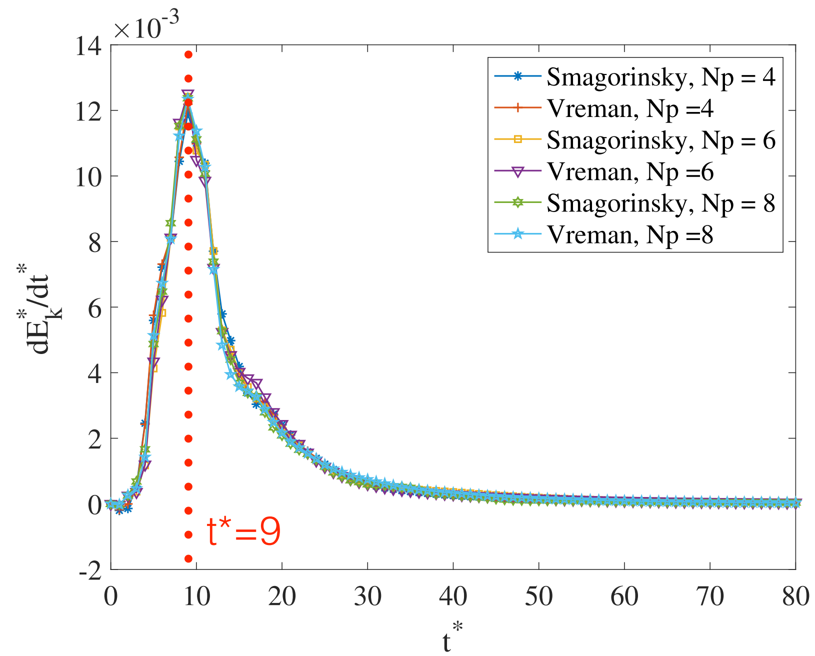


t* = 60

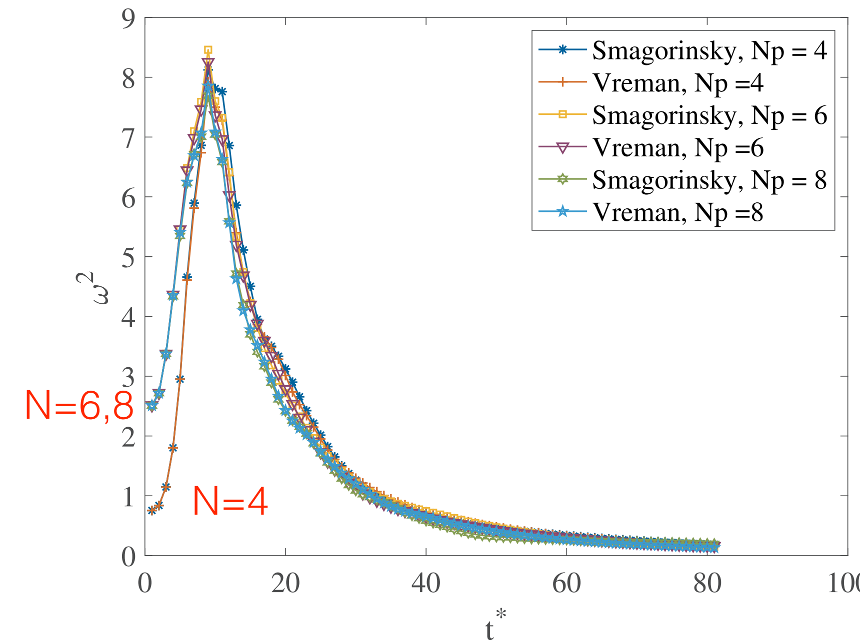




Taylor-Green vortex in a triply periodic box. Simulations. N=4 and Np=128 points in each of the 3 directions.



Change in kinetic energy peaks at t*=9 for all simulations

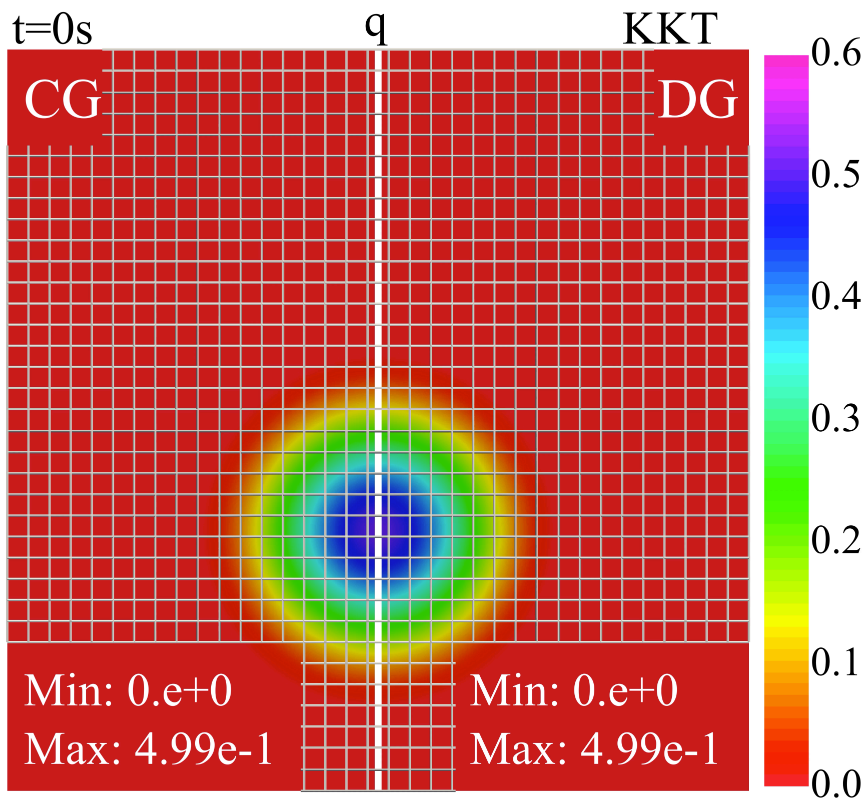


Enstrophy for N=4 is lower at early times. For N=6,8 the values are similar.

Result 3: Positivity-Preserving Schemes*

Test Problem: Rising Thermal Bubble
Discretization: CG+No AMR & DG+No AMR
Positivity Preserving Limiter: KKT

Unphysical minima NOT observed with AMR



KKT (Karush-Kuhn-Tucker) Formulation

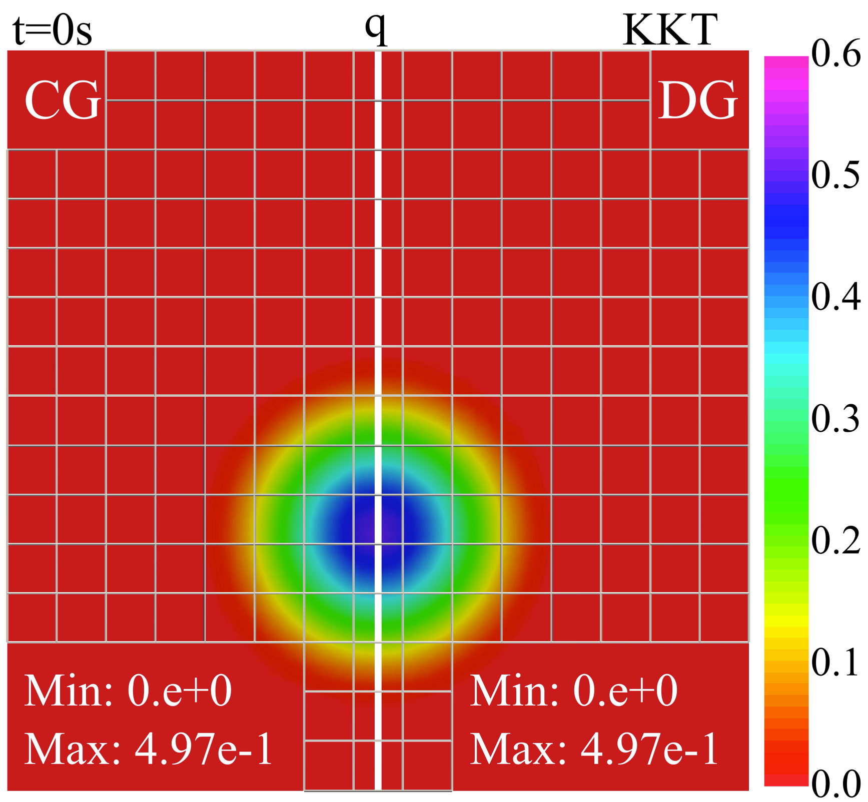
The modified solution q^* should preserve:

1) Accuracy: $\min ||q - q^*||^2$

2) Positivity: $q^* \geq 0$

3) Conservation: $\bar{q} = \bar{q}^*$

Unphysical minima ARE observed with AMR



The optimization problem solved using semi-smooth Newton method

Talk Summary

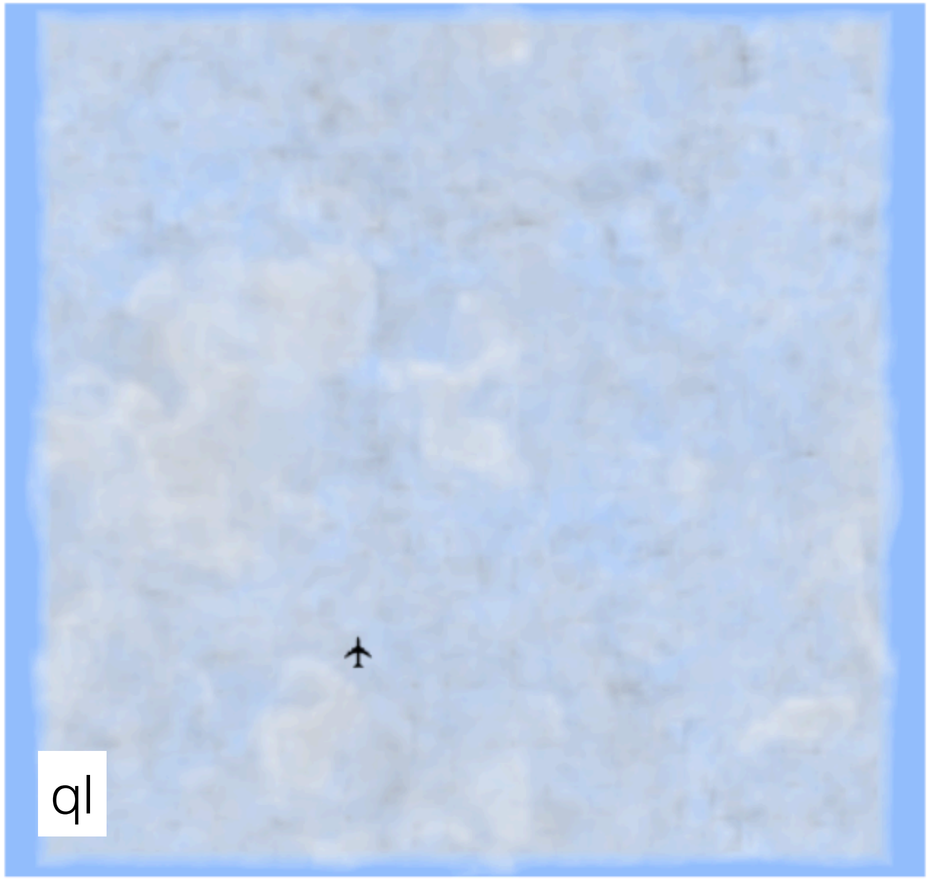
- Governing Equations
- Underlying Numerics
- Results with NUMA
- Results with CLIMA
- Summary

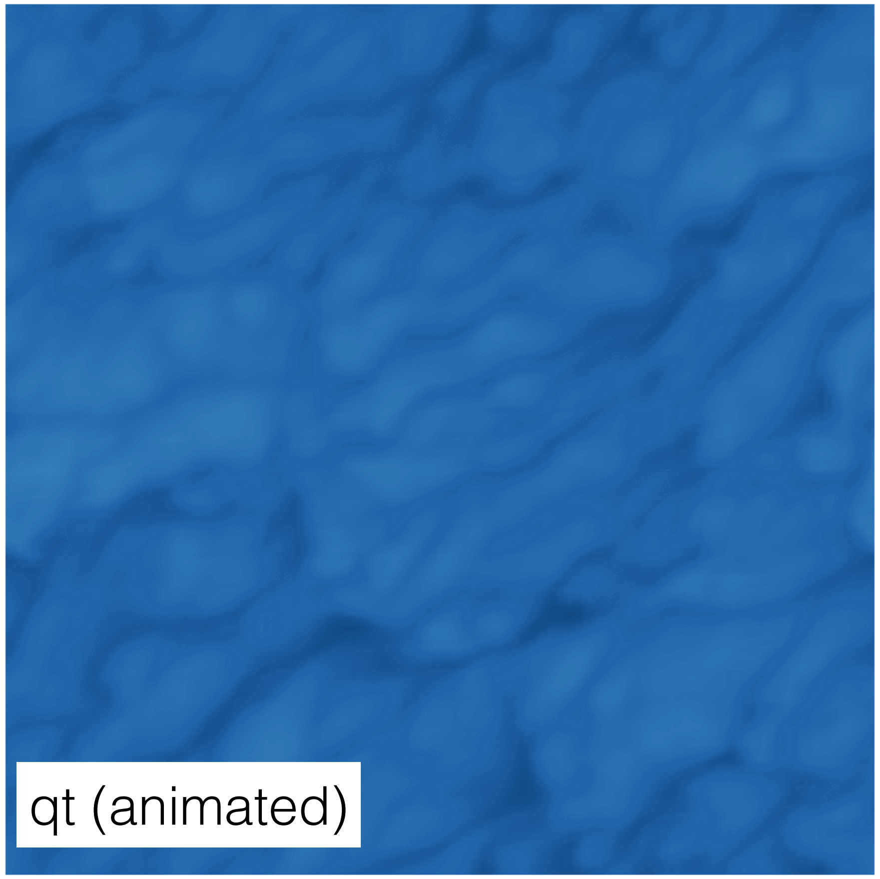
CLIMA Update

- CLIMA is a collection of atmospheric, ocean, ice, and land models for climate studies.
- NPS has developed CLIMA-atmos and works with MIT on CLIMA-ocean. Two groups at MIT involved: one works on the ocean model and the other works on Julia-related issues. Caltech works on parameterizations, data assimilation, and oversees the project. JPL works on the land model.
- CLIMA uses DG throughout and is written in Julia. It has been written in a platform portable way (using something we call KernelAbstractions.jl) which allows running on CPUs and GPUs.
- We began work in 2018
- CLIMA Results:
  - CLIMA-atmos has been run in LES mode for DYCOMS-II (RF01) (Result 1).
  - CLIMA-atmos has been run in global mode using the Baroclinic Instability and Held-Suarez test cases in dry mode (Result 2). Moist versions of these tests are under way.
  - Simulations run on Google Cloud platform (GCP) (Result 3).
  - Performance on GPUs (Result 4).

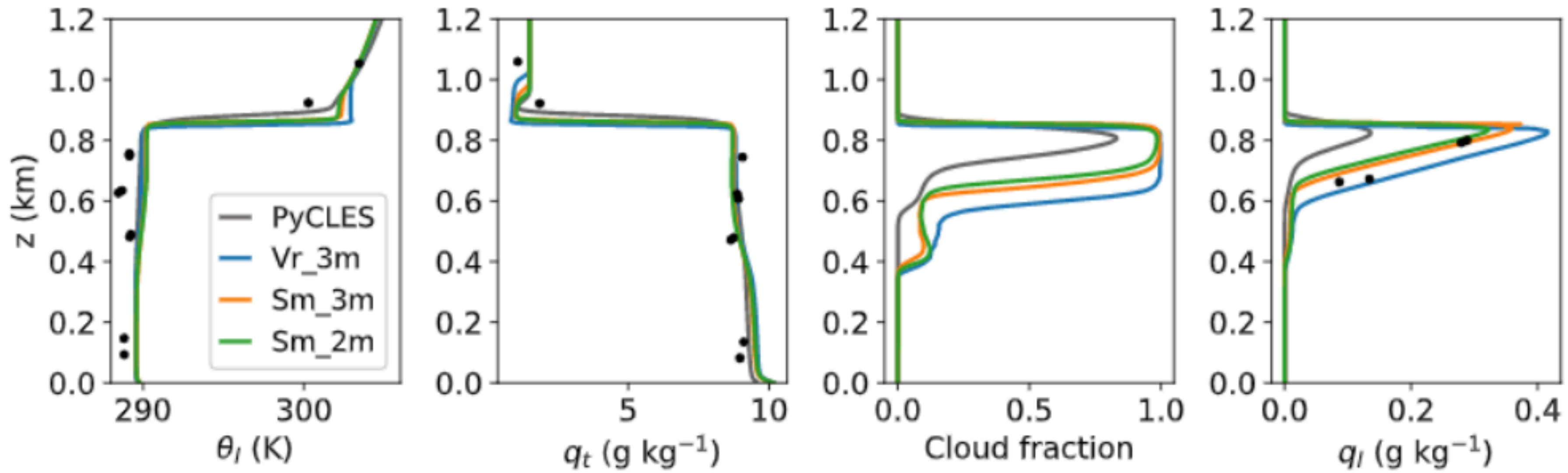
Result 1: CLIMA-LES DYCOMS-II (RF01)

Instantaneous volume rendering of ql at t= 4 hours



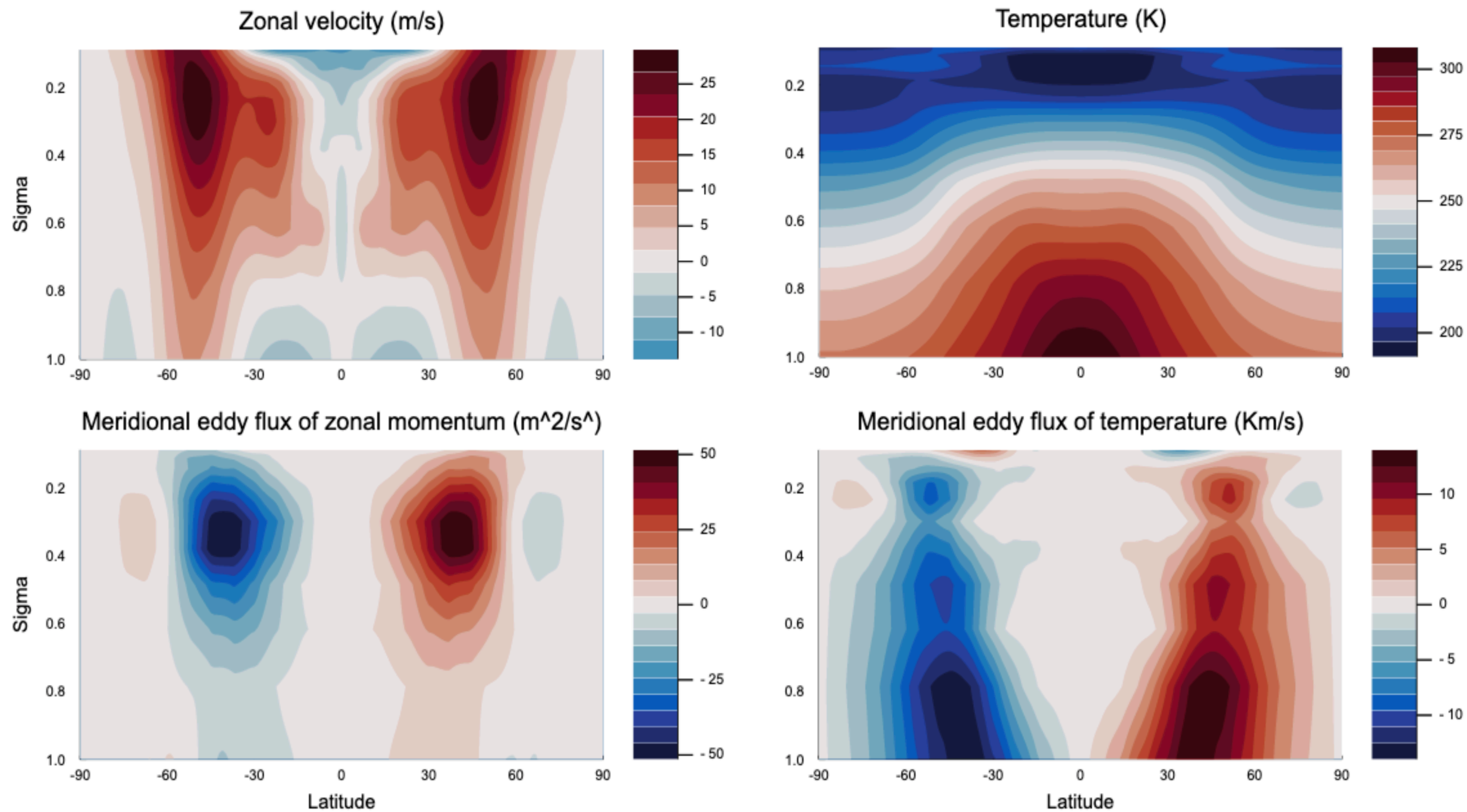


4 hr average statistics of theta, qt, cloud fraction and ql



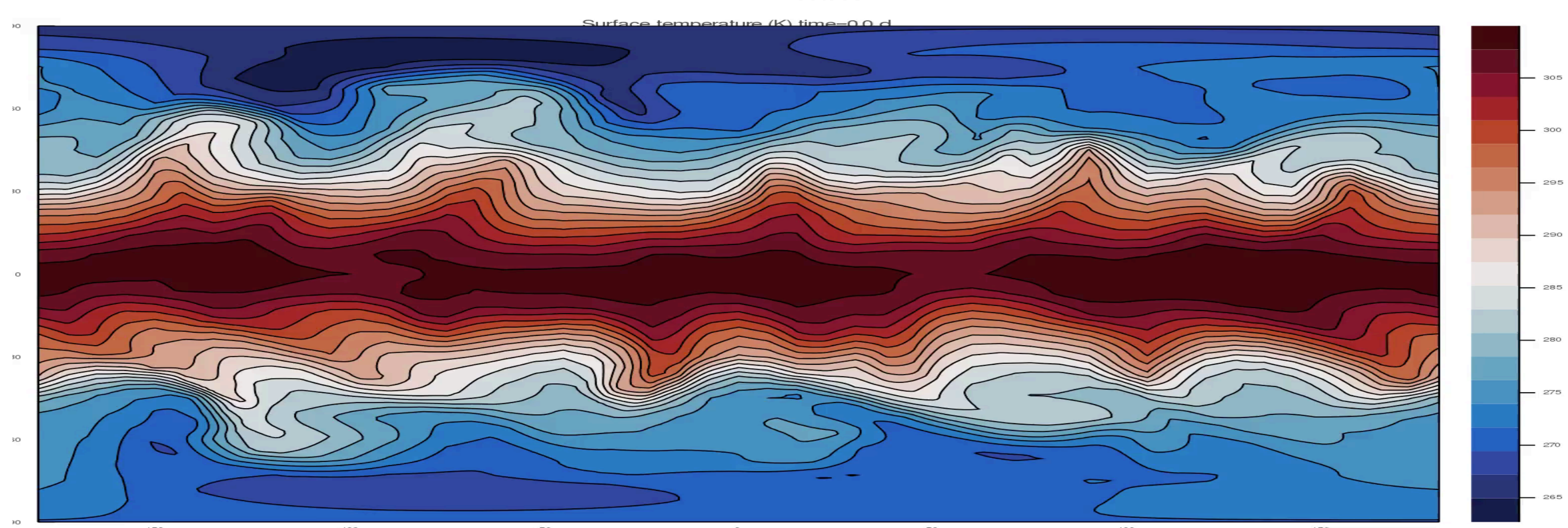
* Courtesy of Yassine Tissaoui and Simone Marras (NJIT), Akshay Sridhar, Charlie Kawczynski, and Zhaoyi Shen (Caltech), and NPS numerics group.

Result 2: CLIMA-GCM Held Suarez



1200 days at 150kmx750m with P=4

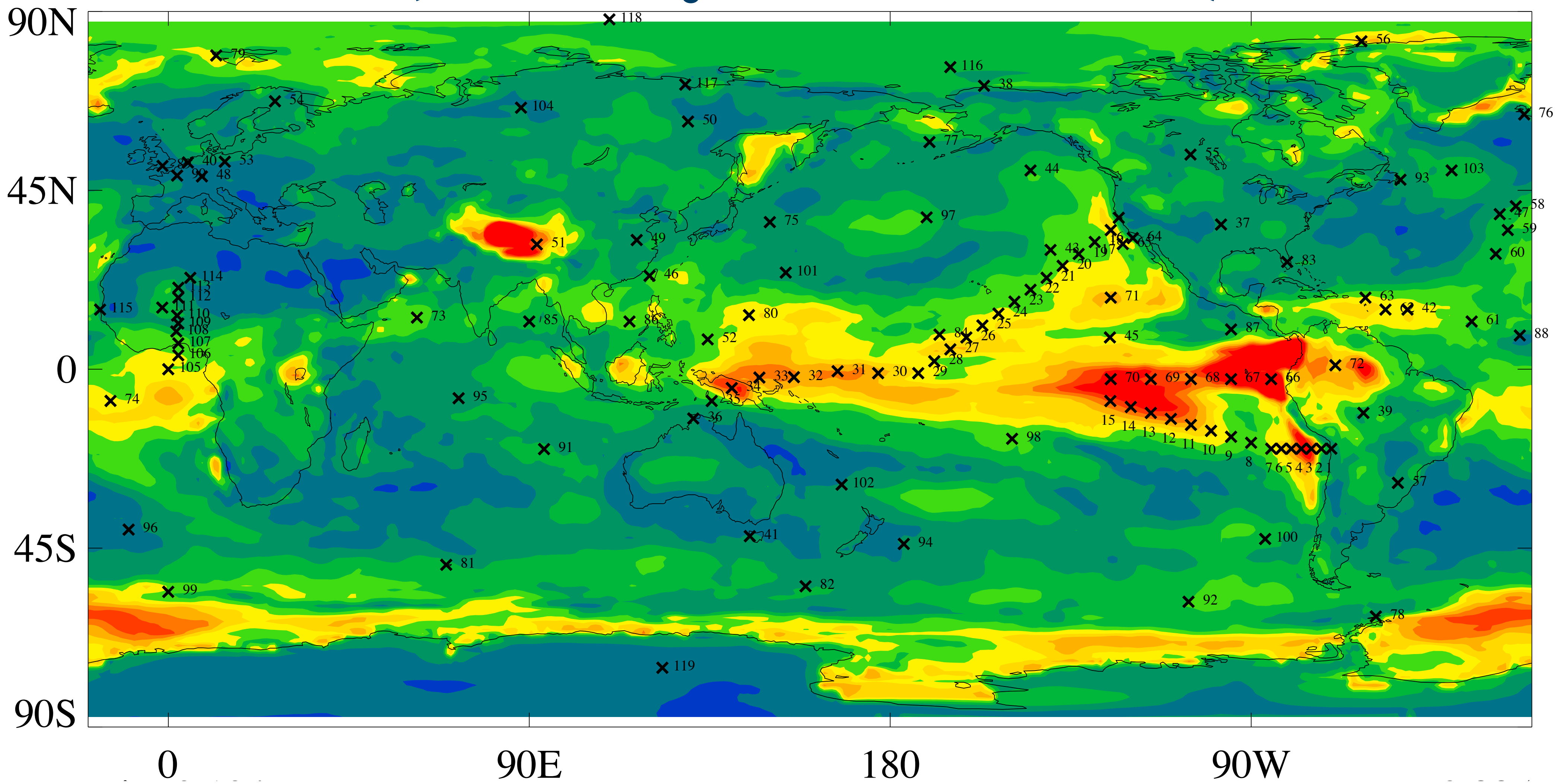
Lon-Lat View of Surface Temperature for 200 days



* Courtesy of Maciej Waruszewski and Thomas Gibson (NPS), Toby Bischoff, Lenka Novak, Zhaoyi Shen, and Jia He (Caltech), and NPS numerics group.

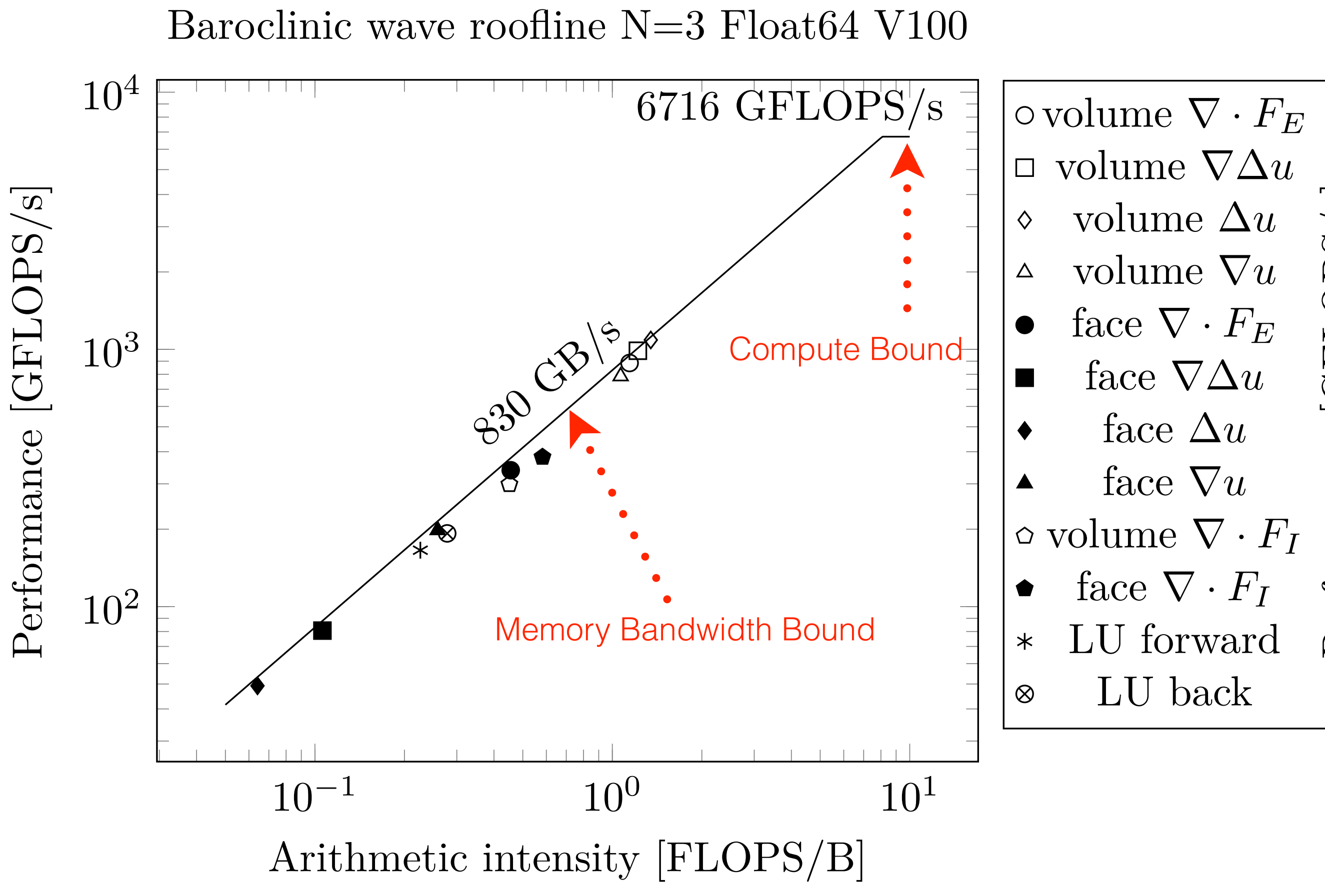
Result 3: GCM-LES Google Cloud Runs

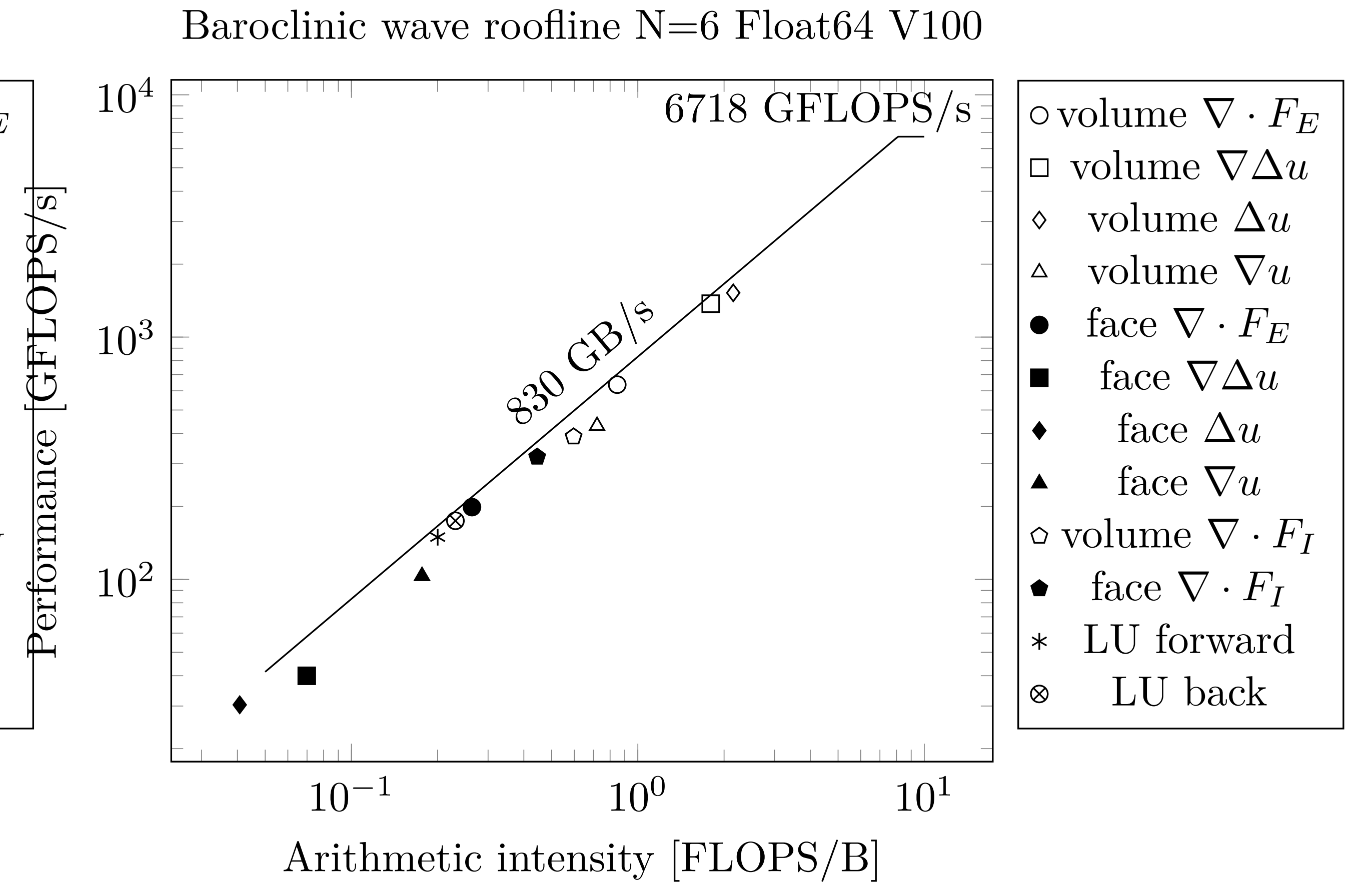
(thanks to Google Cloud Platform for access)



- **Goal**: Improve EDMF parameterizations for GCM by running targeted LES simulations
- **GCM**: HADGEM2A (AMIP archives) used as proxy. 115 data sites (x's in plot).
  - Prescribed surface fluxes, saturation adjustment for moist thermodynamics
- **LES:** 6 hour sim time in a 2km x 2km x 4km domain [7.2km x 7.2km x 4km for main dataset]
  - 75m x 75m x 20m equivalent DG resolution P=4
  - 60 simulations, each simulation uses 8 V100 Nvidia GPUs.
  - Currently, no microphysics.
  - Goal will be to do order of 100 to 1000 s with microphysics so expect to use O(20K) GPUs (few simulations for some sites)

Result 4: Roofline Model*





- **Goal**: Check the performance of the model on a single Nvidia V100 GPU card using DP.
- **Test Case**: Baroclinic Instability on the Sphere. We test all operators including the DG volume and face integrals as well as the first order (DG on LGL points with Rusanov flux) and second order operators (LDG) with 1D-IMEX (ARK2) method.
- **Results:** As we saw with NUMA on Titan (OLCF), face kernels are the least performant while the volume kernels are the most performant (Abdi et al IJHPCA 2017 and 2019).

*Courtesy of Maciek Waruszewski, Lucas Wilcox, Jeremy Kozdon, and Thomas Gibson (NPS); Valentin Churavy (MIT).

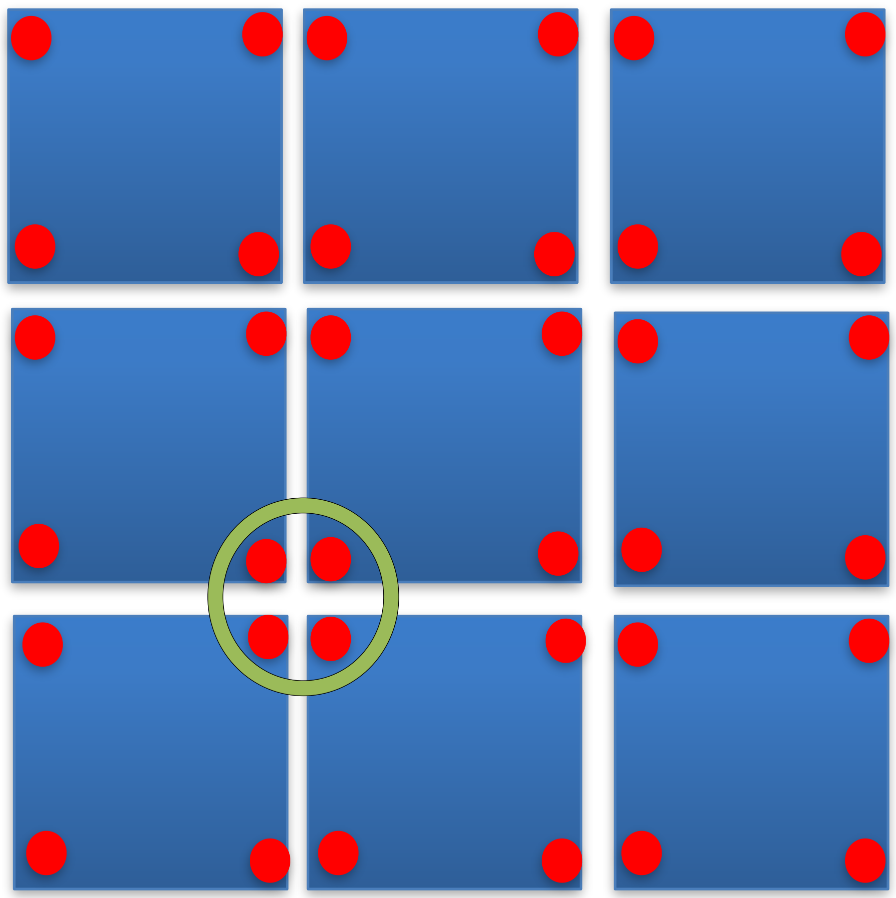
Talk Summary

- Governing Equations
- Underlying Numerics
- Results with NUMA
- Results with CLIMA
- Summary

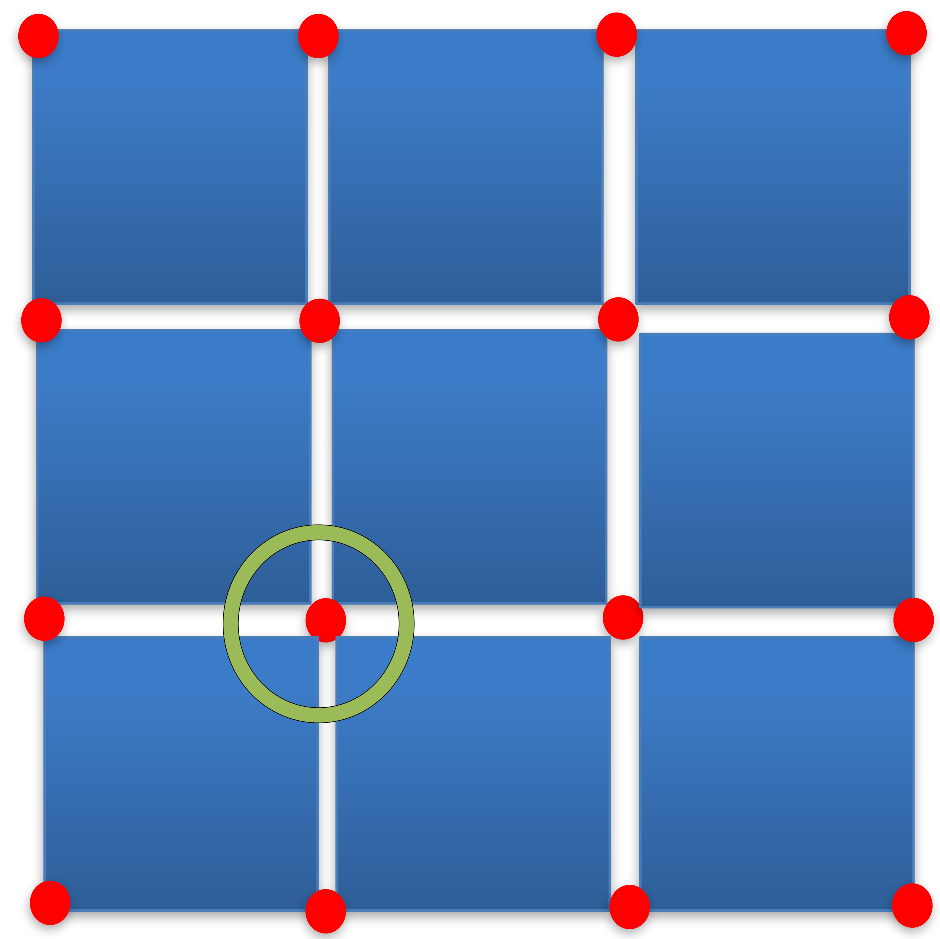
Summary

- **Contravariant Form**
  - Allows taking advantage of special directions in the problem (e.g., decomposing the horizontal and vertical domains in global simulations, and applying boundary conditions), and makes it easier to prove conservation
- **NUMA**
  - NUMA-HA "breathing atmosphere" boundary condition works in 1D; 3D simulations are in progress
  - NUMA-LES simulations show that 6th order may be the maximum order worth considering
  - NUMA-LES with AMR shows that positivity-preserving schemes need special handling
  - Derivation of Schur complement (SC) for DG allows us to pursue similar preconditioning strategies for both CG and DG. We are currently extending the SC idea to the HEVI time-integrators (reduced the 5Nx5N nonlinear equations to 3Nx3N)
- **CLIMA**
  - CLIMA v0.1 was released in June 2020. This version was used in the Google Cloud LES simulations (DYCOMS-II) and in the GCM simulations for the baroclinic instability and Held-Suarez test cases (dry)
  - CLIMA-atmos has been run with moist physics and radiation for LES simulations. GCM simulations of this type are underway
  - Three projects currently underway will allow us to complete the modeling infrastructure: (1) EDMF schemes, (2) nonlinear HEVI time-integrators, and (3) completion of the CLIMA-ocean global model (current model runs in flow-in-a-box mode only)

Unified CG/DG methods



Discontinuous Storage (CGd,DG)



Continuous Storage (CGc)

- NUMA carries three numerical methods: CGc, CGd, and DG.
- Discontinuous Storage requires more memory but solution vectors are easily stored in contiguous memory (direct addressing). Great for vectorization and AMR.
- Continuous Storage requires less memory but requires indirect memory addressing which may result in more cache misses.