HipFT v1.0

An Open Source High-Performance Flux Transport Model

Ronald M. Caplan[1], Miko M. Stulajter[1], James A. Turtle[1], Jon A. Linker[1], Cooper Downs[1], Lisa A. Upton[2], Charles Nicholos Arge[3], Raphael Attie[3], Carl J. Henney[4], and Nikolai Pogorelov[5]

(1) Predictive Science Inc., (2) Southwest Research Institute, (3) NASA Goddard Space Flight Center, (4) Air Force Research Laboratory Kirkland AFB, (5) University of Alabama in Huntsville

Predictive Science Inc.

- Surface flux transport models
- Open Source Flux Transport (OFT)
- High Performance Flux Transport (HipFT)
- Flow and diffusion models
- Data Assimilation
- Multiple Realizations
- Numerical Methods
- Code Implementation
- Examples
- Availability

Surface Flux Transport Models

- SFT treats the solar surface radial magnetic field as a scalar quantity subject to 2D surface flows and processes
- Used to generate full-Sun maps, constrain dynamo models, study surface dynamics, solar cycle prediction, etc.
- Many models exist:
  ADAPT, LMSAL-ESFAM, AFT, etc.
- While some produce publicly available full-Sun maps, none are currently open-source or able to be run on-demand

- Part of the *"Improving Space Weather Predictions with Data-Driven Models of the Solar Atmosphere and Inner Heliosphere"* SWQU project
- Open source and extensible
- Three main components:

SWQU

Space Weather with Quantified Uncertainty

OFTpy: Aquire and prepare observational data

ConFlow: Generate supergranular convective flows

HipFT: Integrate the flux transport model

Implements advection, diffusion, data assimilation, and flux emergence over multiple realizations using high-accuracy numerical methods and CPU/GPU parallelism

$$\frac{\partial B_r}{\partial t} = -\nabla_s \cdot (B_r \mathbf{v}) + \nabla_s \cdot (\nu \nabla_s B_r) + S,$$

$$\nabla_s \cdot (B_r \mathbf{v}) = \frac{1}{R_\odot \sin\theta} \frac{\partial}{\partial \theta}(\sin\theta \, B_r \, v_\theta) + \frac{1}{R_\odot \sin\theta} \frac{\partial}{\partial \phi}(B_r \, v_\phi),$$

$$\nabla_s \cdot (\nu \nabla_s B_r) = \frac{1}{R_\odot^2 \sin\theta} \frac{\partial}{\partial \theta} \left(\nu(\theta, \phi, B_r) \sin\theta \frac{\partial B_r}{\partial \theta} \right) + \frac{1}{R_\odot^2 \sin^2\theta} \frac{\partial}{\partial \phi} \left(\nu(\theta, \phi, B_r) \frac{\partial B_r}{\partial \phi} \right),$$

HipFT Flow Models

- Differential rotation

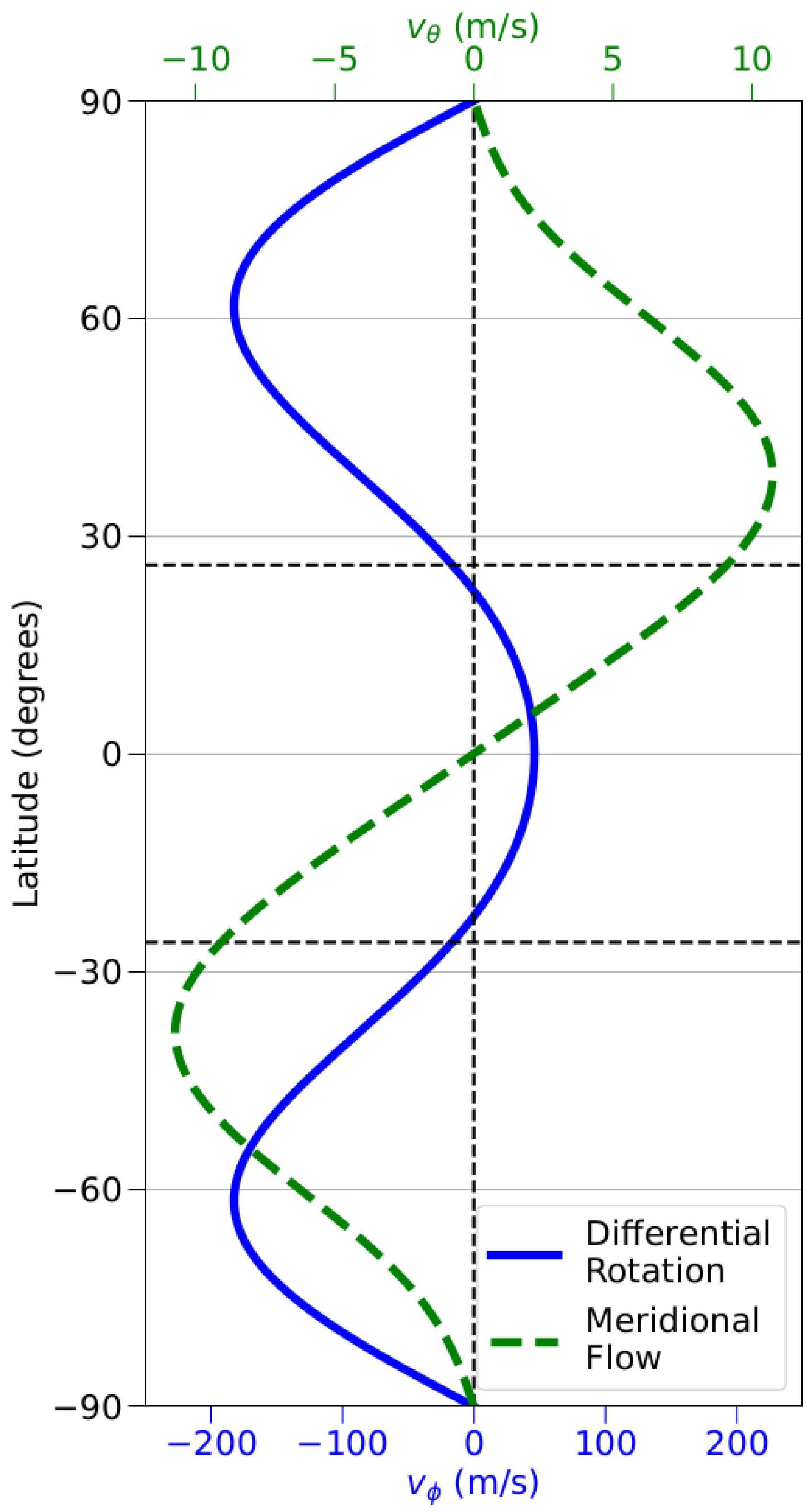
$$v_\phi(\theta) = \left[d_0 + d_2\ \cos^2(\theta) + d_4\ \cos^4(\theta)\right]\ \sin\ \theta,$$

- Meridianal flows

$$v_\theta(\theta) = -\left[m_1\ \cos\ \theta + m_3\ \cos^3\theta + m_5\ \cos^5\theta\right]\ \sin\ \theta,$$

- Velocity attenuation

$$v_{\theta/\phi} \rightarrow v_{\theta/\phi}\ \left[1.0 - \tanh\left(\frac{|B_r|}{B_0}\right)\right],$$

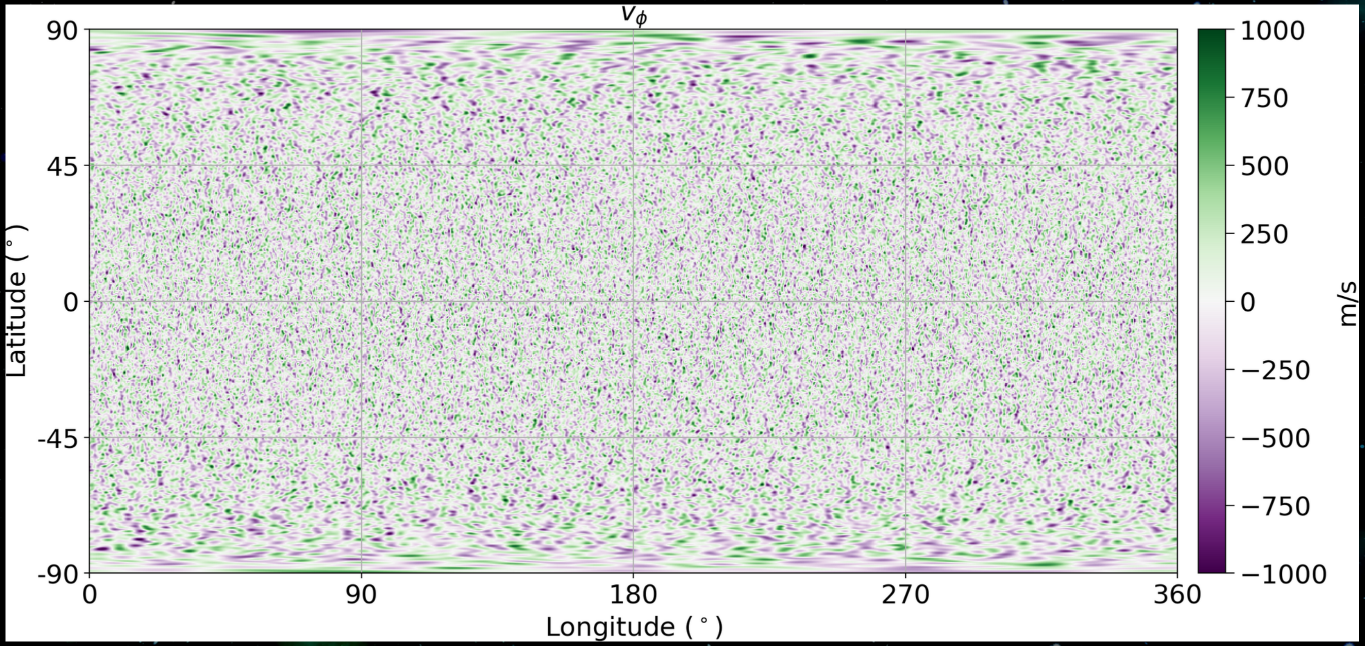


HipFT Flow Models

- Diffusion in SFT models used as a proxy for the flux cancellation caused by granular and super-granular motions
- However, there are advantages to directly modeling these flows
- The default HipFT resolution of 1024x512 is high enough to resolve most of the super-granular scale sizes
- ConFlow generates a sequence of flow data encompassing random motions and super-granulation
- HipFT reads in the files and drives the FT with the flows
- Some diffusion is still necessary to represent flux cancellation at smaller scales

ConFlow

[Hathaway et. al. (2010,2015)]



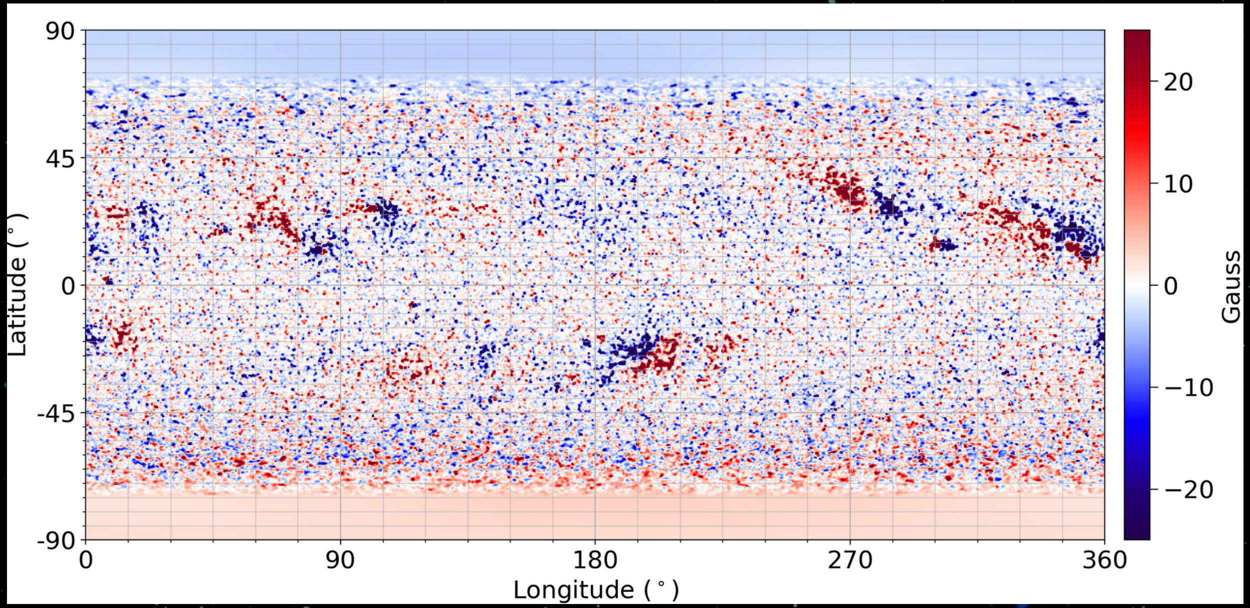
HipFT Diffusion Models

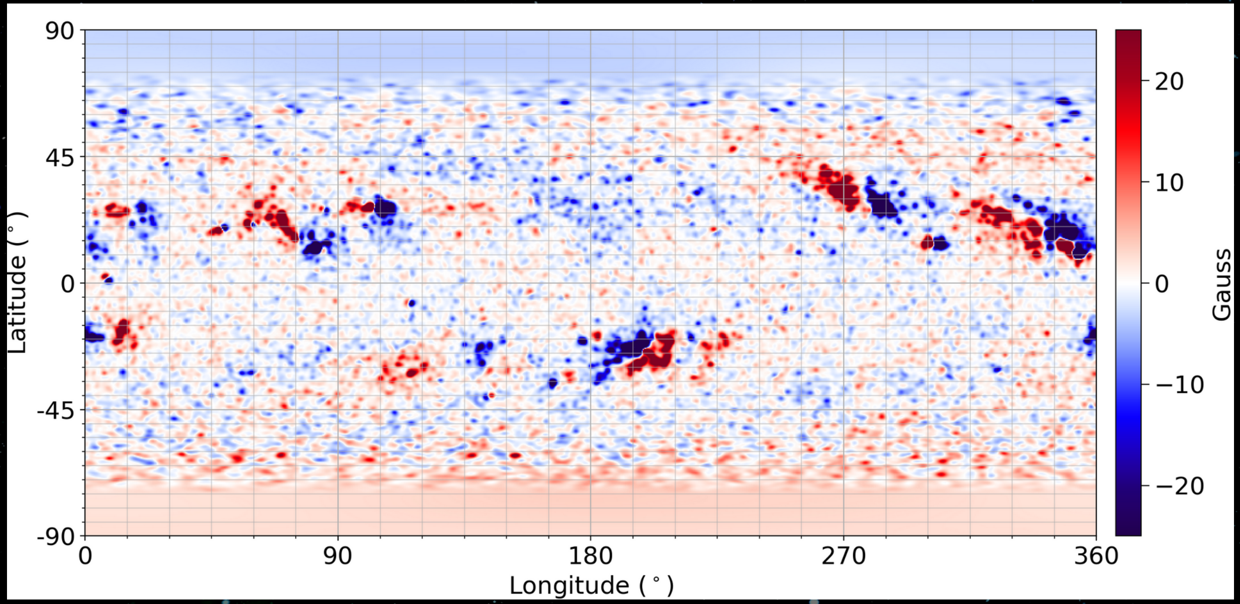
- Diffusion coefficient can be constant, or a user-defined spatially varying file

$$\nu(\theta, \phi)$$

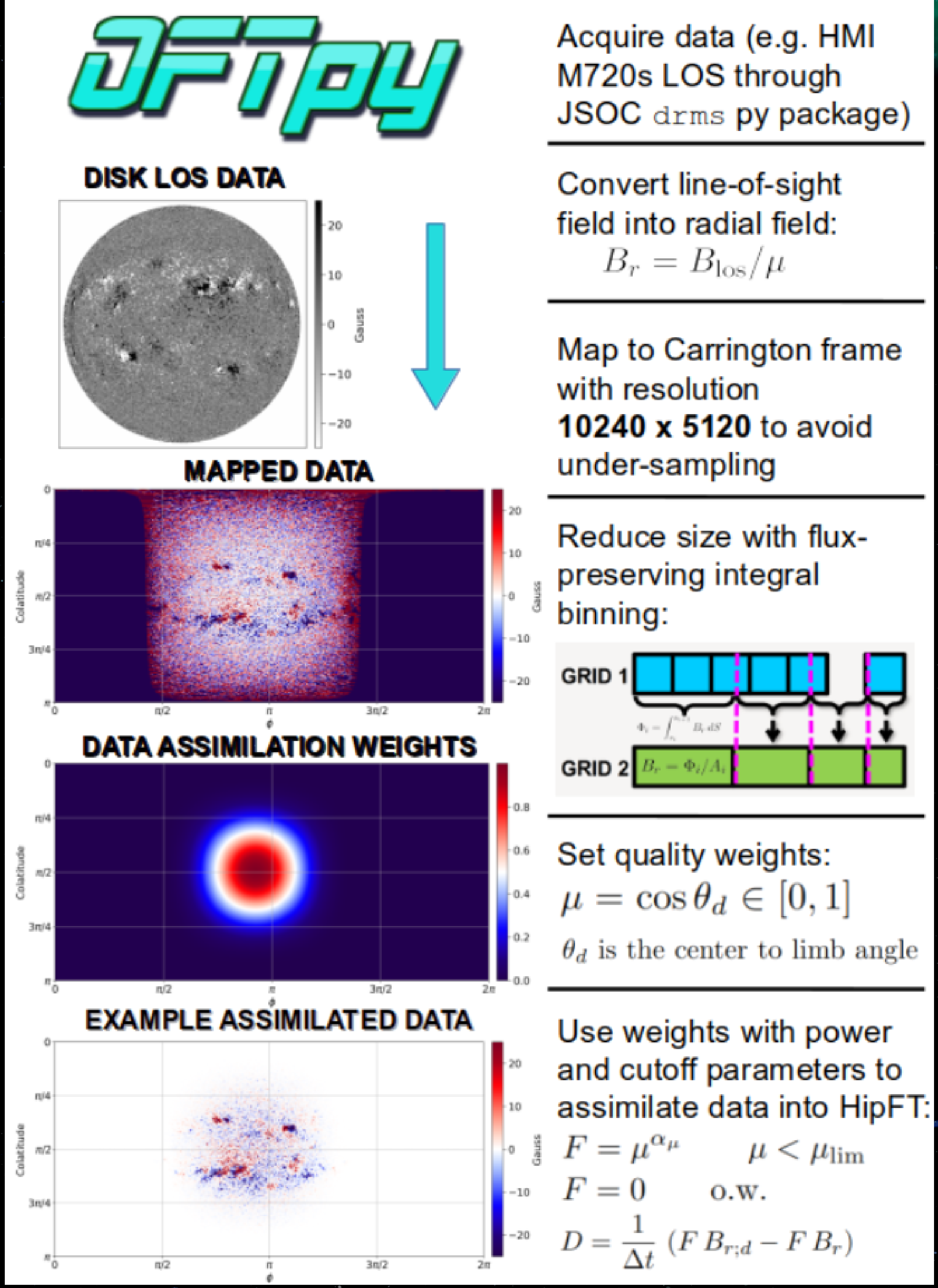
- HipFT can be used as a magnetogram smoother, in which case one can select a grid-based diffusion coefficient

$$\nu_g = \alpha_\nu \left[(\Delta\theta)^2 + (\Delta\phi \sin\theta)^2 \right],$$





Data Assimilation



- Data assimilation uses the output data from OFTpy
- A default weighting function is included in the data cube, applied as:

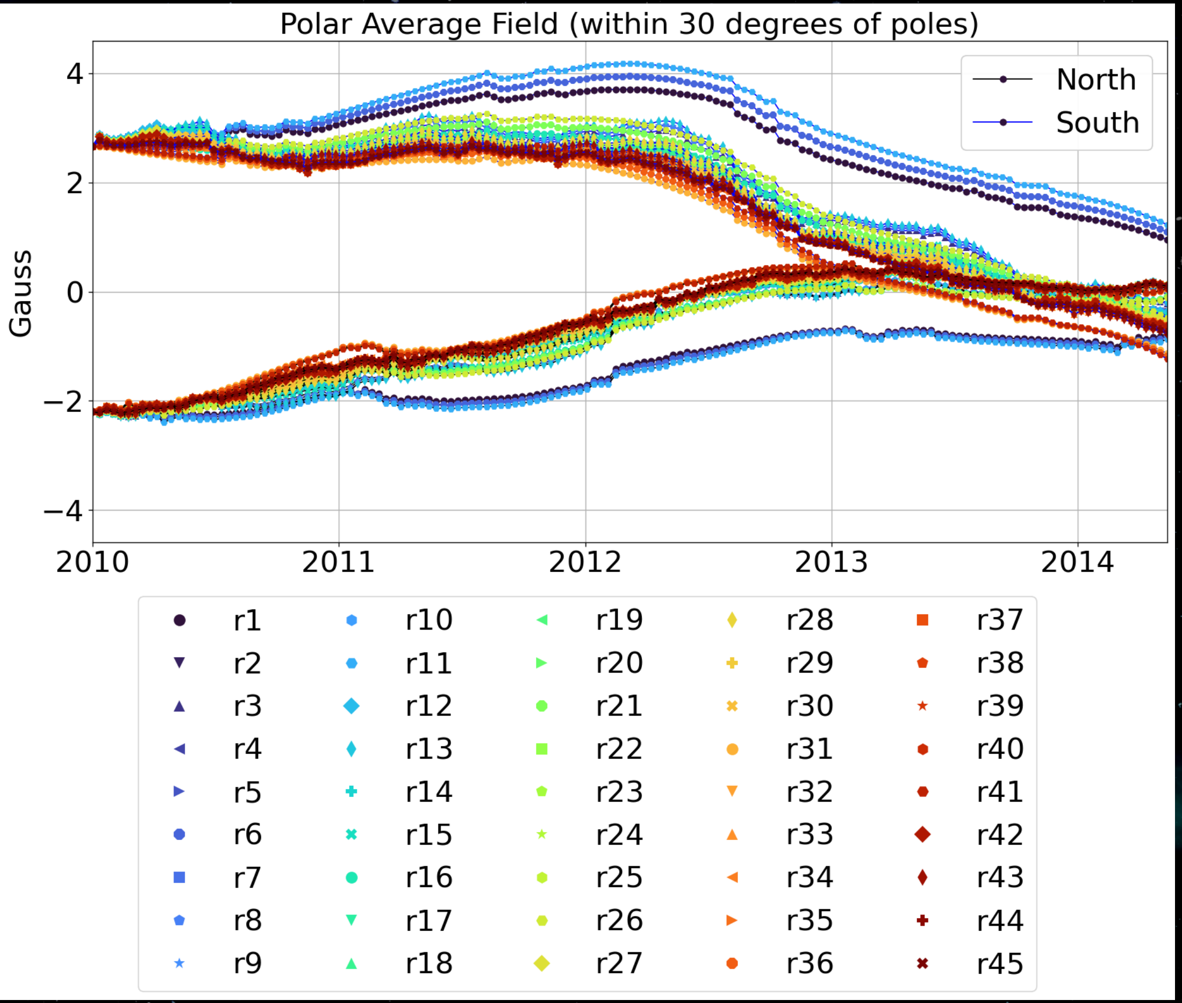
$$B_r \rightarrow F\,B_{r;d} + (1-F)\,B_r\,,$$

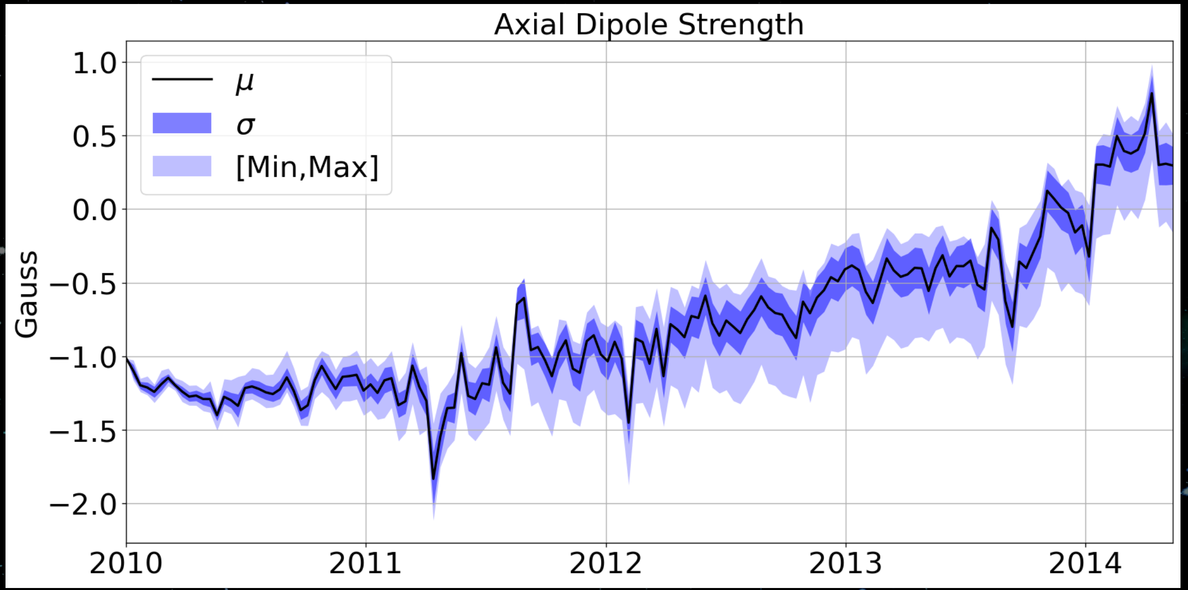
- The center-to-limb distance is also provided, which can be used to generate a user-defined custom weight profile:

$$F = \mu^{\alpha_\mu} \qquad \mu < \mu_{\rm lim}\;\&\;|\theta_1| < \theta_{1,\rm lim}\,, \qquad F = 0 \;\; \text{o.w.},$$

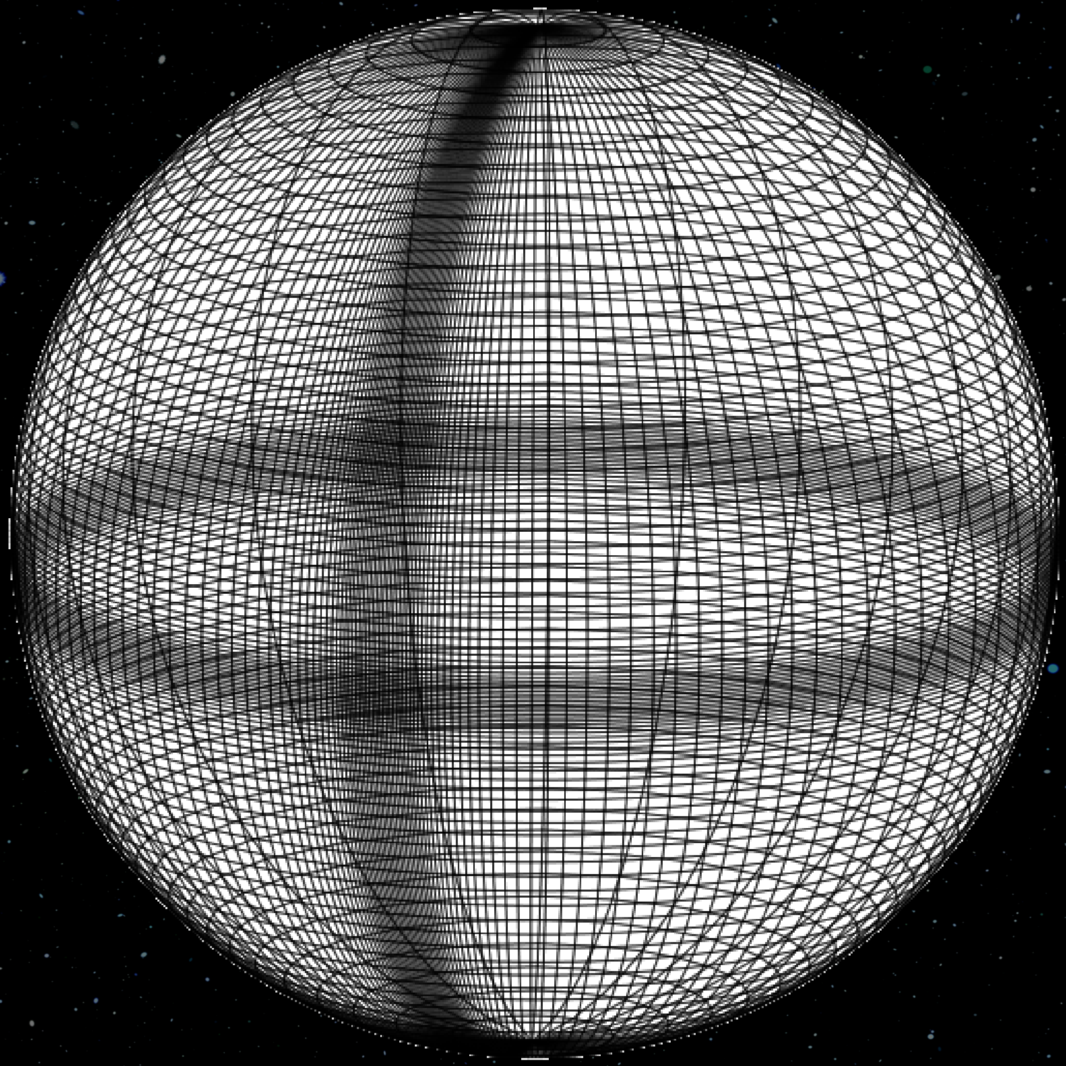
Multiple Realizations

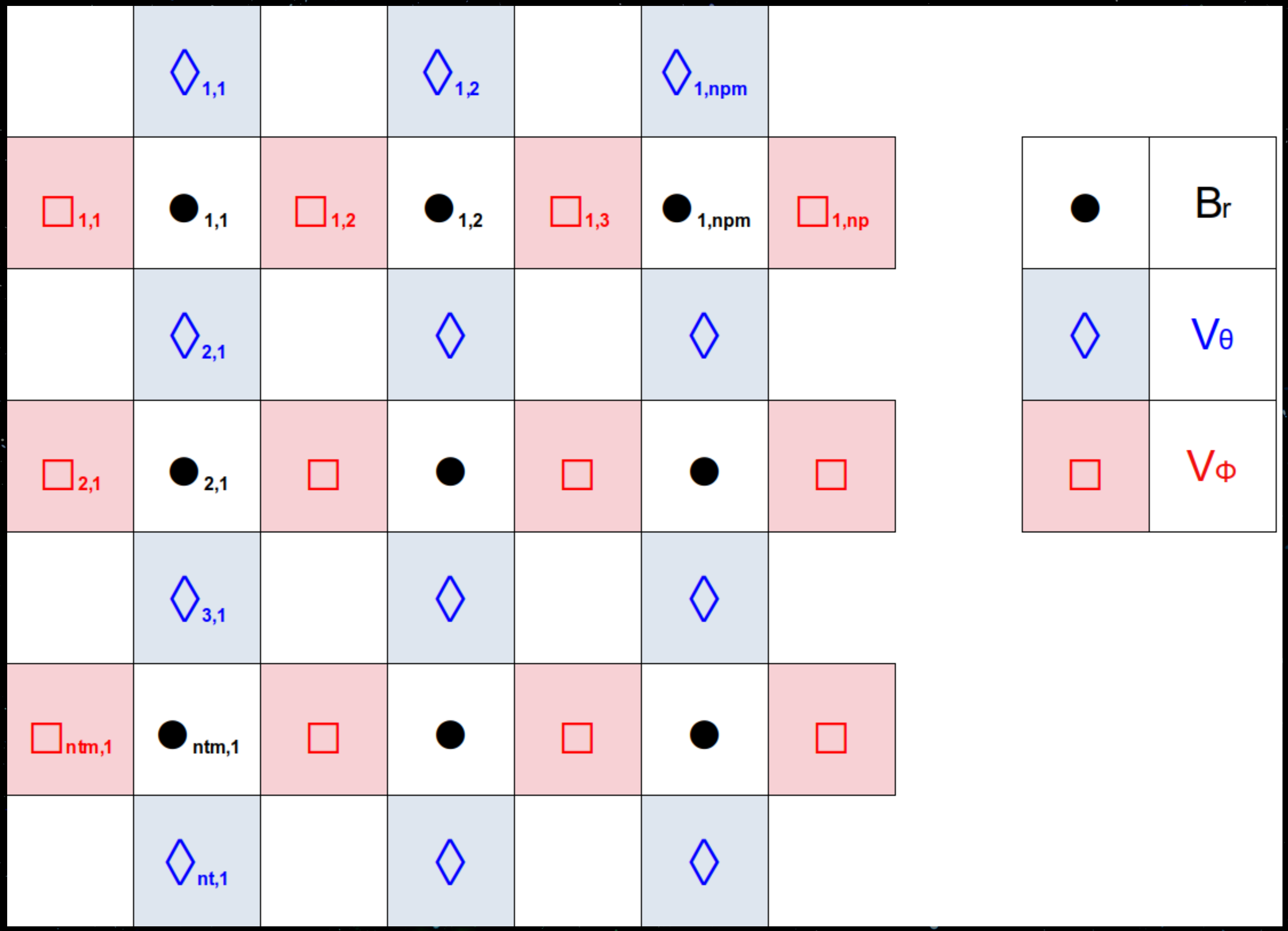
- Can run multiple realizations simultaneously across many model parameters
- Current cross-realization parameters include diffusion rate, flow profile coefficients, flow attenuation levels, and data assimilation options
- Post processing python scripts are included to analyze results





Non-uniform, logically-rectangular spherical surface staggered grid

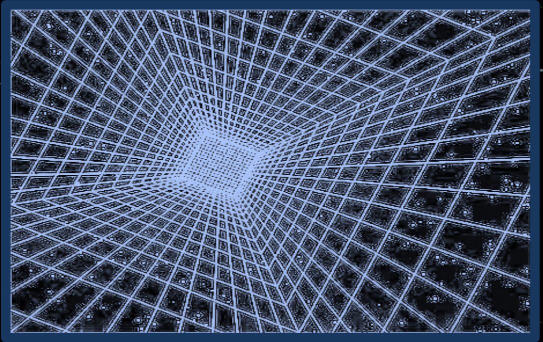




HipFT Numerical Methods

ADVECTION: 3rd-order SSPRK(4,3)

DIFFUSION: 2nd-order Runge-Kutta-Gegenbauer Super Time-Stepping



ADVECTION: 3rd-order WENO3

DIFFUSION: 2nd-order Central Finite Difference

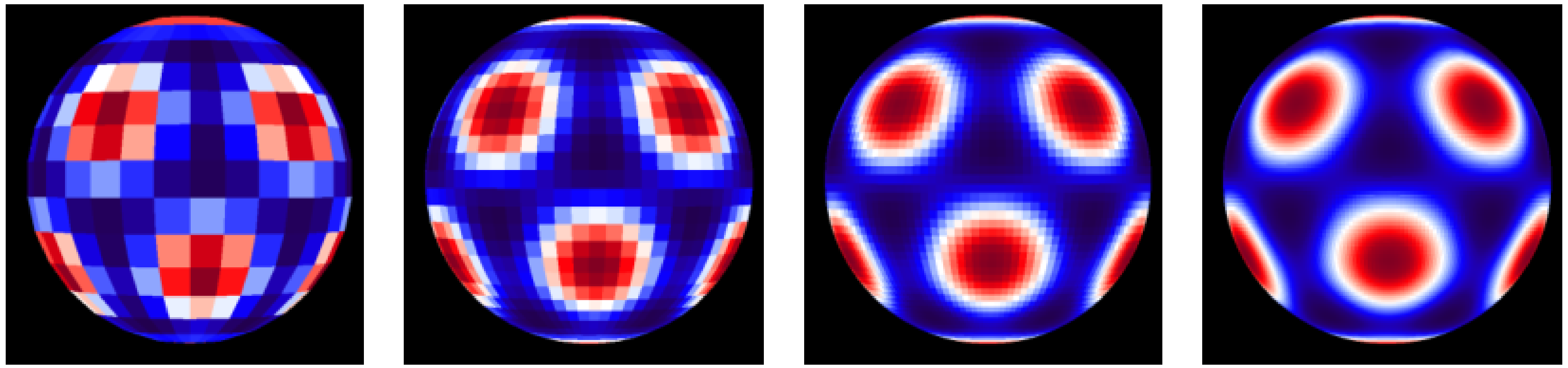
Validation:

$$v_\phi = \Omega \sin\theta$$

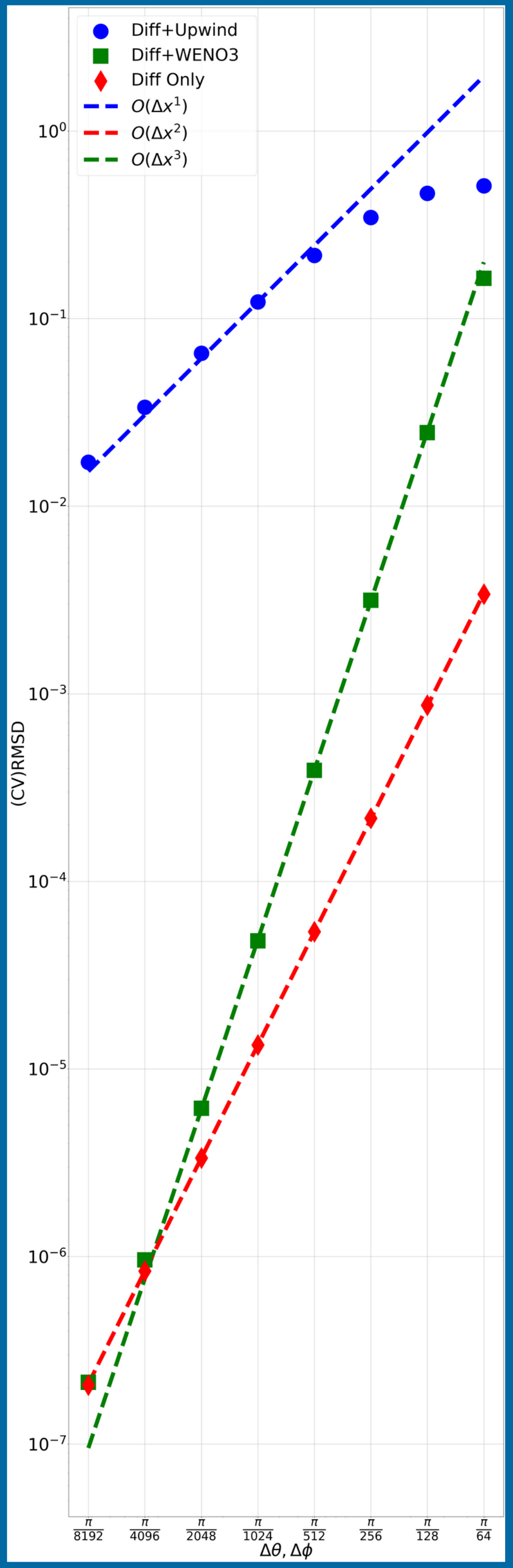
$$\Omega = 1.8076\ldots\ \mathrm{km/s}$$

$$\nu = 500\ \mathrm{km^2/s}$$

$$u(\theta,\phi,t) = 1000\, e^{-42\,\nu\, t}\left(Y_6^0(\theta,\phi) + \sqrt{\frac{14}{11}}\, Y_6^5(\theta,\phi)\right)$$



$\Delta\theta, \Delta\phi = \pi/16$ | $\Delta\theta, \Delta\phi = \pi/32$ | $\Delta\theta, \Delta\phi = \pi/64$ | $\Delta\theta, \Delta\phi = \pi/128$



HipFT Code Implementation

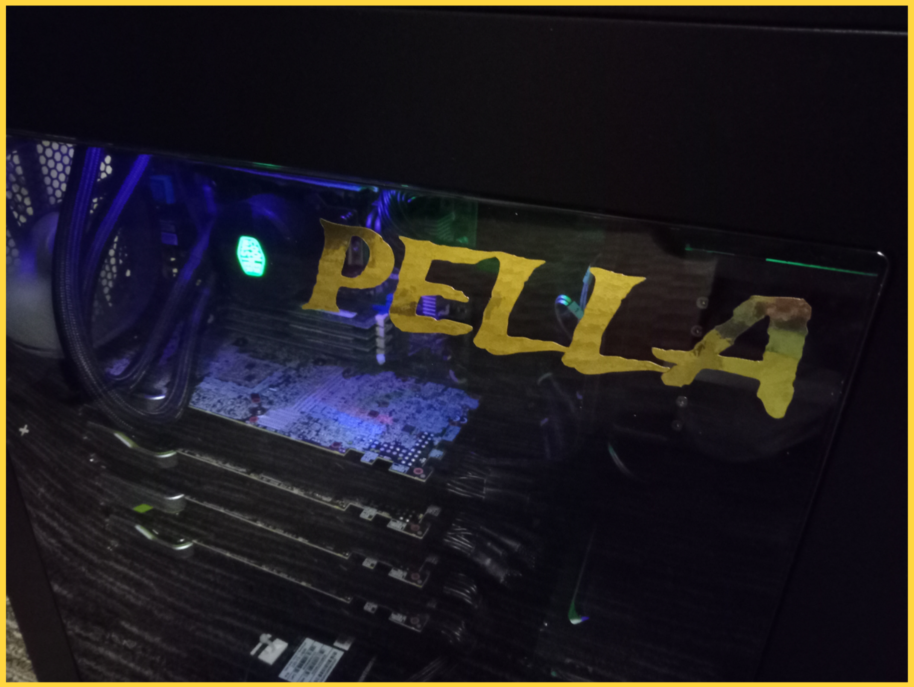
- Written in Fortran 2023
- Parallelized for multi-core CPU and GPUs with Fortran standard's `do concurrent` and OpenMP Target for CPU-GPU data movement

```
do concurrent (i=1:N,j=1:M)
  Computation
enddo
```

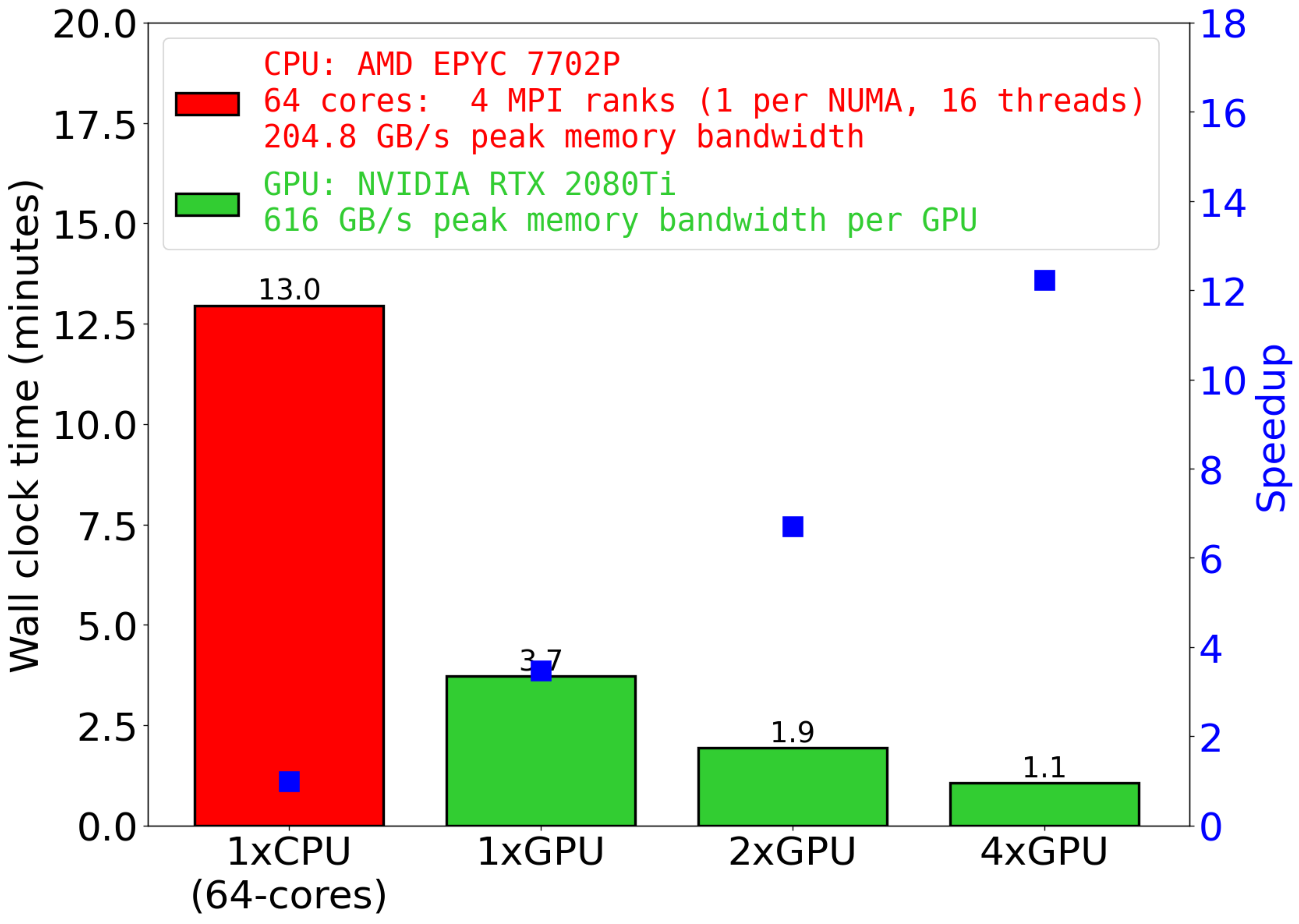
```
!$omp target enter data map(to:a)
!$omp target enter data map(alloc:b)
  Computation
!$omp target exit data map(from:a)
!$omp target exit data map(release:b)
```

- Parallelized for multiple multi-CPU/GPU nodes across realizations with MPI

HipFT Code Performance



In-house workstation:
EPYC 7702P 64-core CPU
Four RTX 2080Ti GPUs

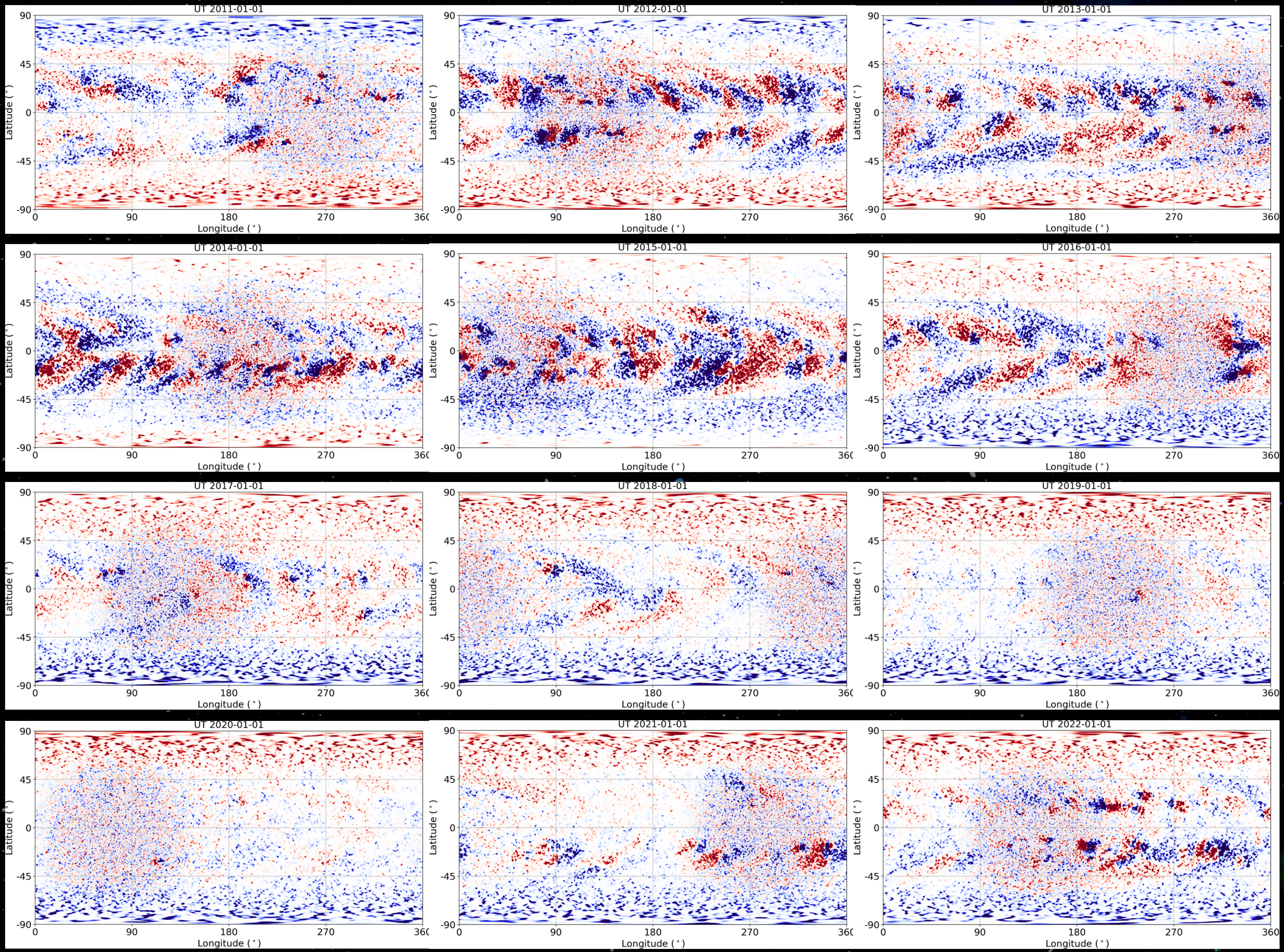


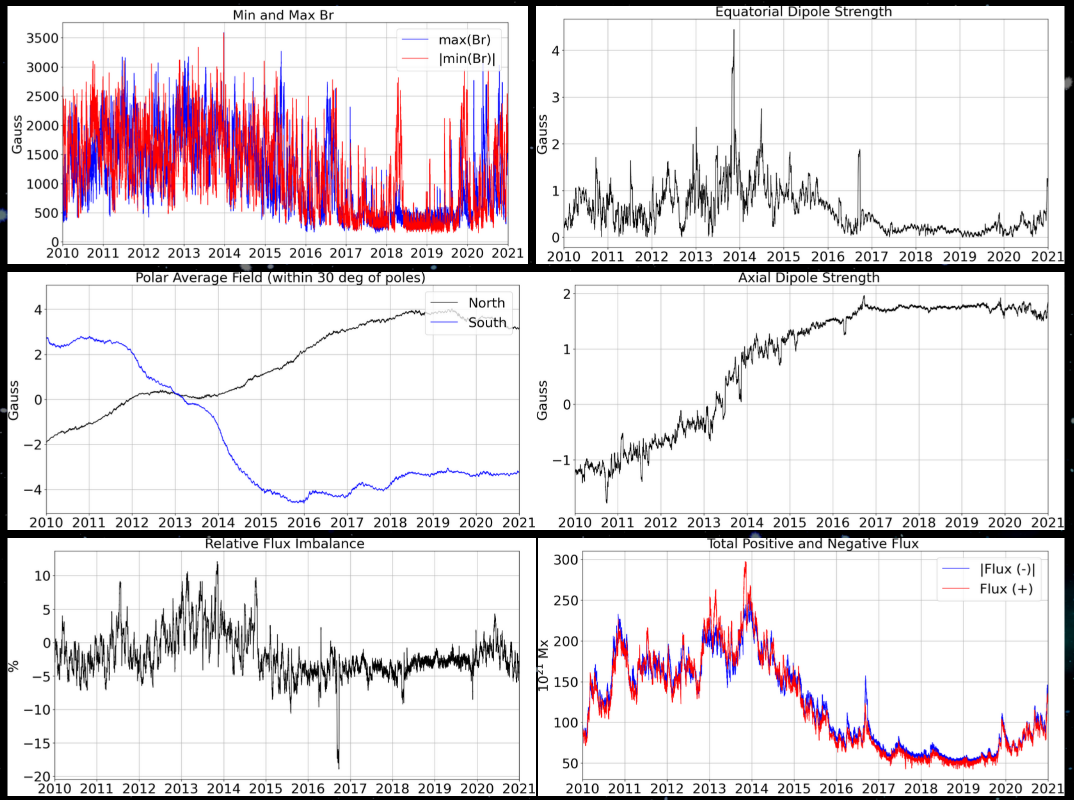
Test: 28-day run at 1024x512 with analytic flow models and diffusion. Eight realizations spanning various levels of diffusion and flow attenuation

OFT Example Production Run

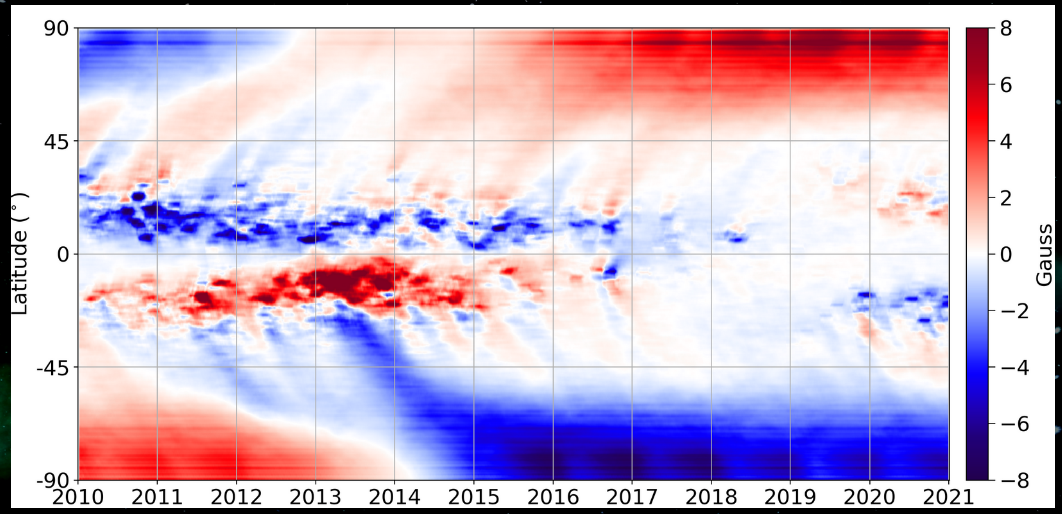
Initial map from AFT model, HMI data assimilation (1-hour cadence), ConFlow (1CR) and analytic flows with 500G attenuation, diffusion of 175 km²/s

Runtime on an NVIDIA RTX 2080Ti GPU: 13:00:00



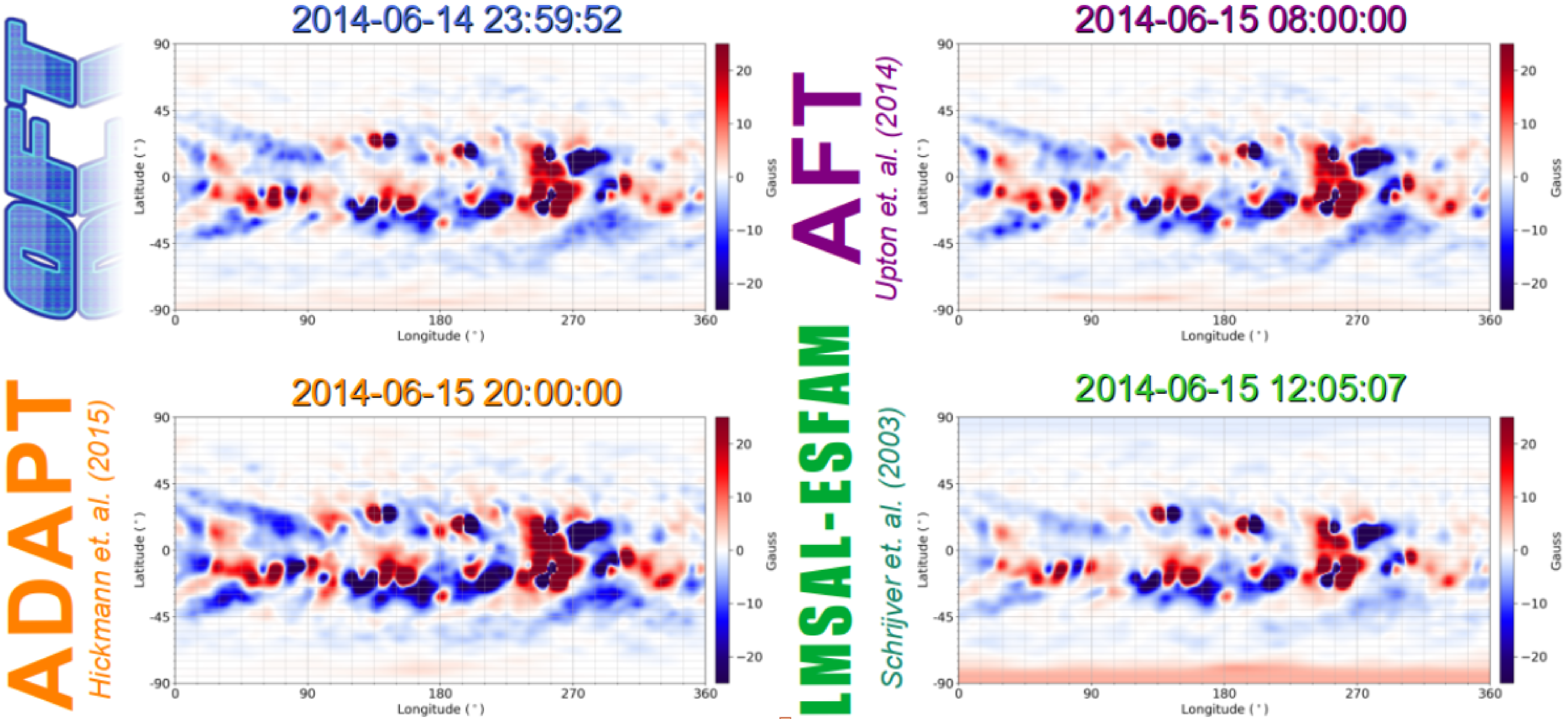


Butterfly Diagram (1CR average)



Comparison of OFT to other SFT models

- Maps from FT models are processed by interpolating to 300x150 resolution, flux balancing, and smoothing
- Note some models apply scaling factors to the HMI data



HipFT Availability

github.com/
predsci/hipft

HipFT Installation


```
> git clone https://github.com/predsci/HipFT.git
```

```
> cd HipFT
```

```
> cp build_examples/build_<CLOSEST>.sh build_local.sh
```

```
Edit build_local.sh to reflect local system/compiler
```

```
> ./build_local.sh
```

```
> cd testsuite; ./run_test_suite.sh
```

Data set for production level example run:

https://zenodo.org/records/10271121

Questions?

Full description of OFT (and HipFT) will be given in a series of papers (in preparation)

More features being added to HipFT:

→ Random flux emergence, source terms

→ Quality of life updates + processing