MARBLE:

How to make an open science global magnetosphere code?

C. Bard, J. C. Dorelli, D. da Silva, G. Khazanov, D. Sur

DASH Conference October 11, 2023

MARBLE under construction

Kinetic MHD Magnetosphere
Leads: Dorelli, Bard

AGATE:
Finite-Volume Solver
Lead: Bard

*DISCO:
Guiding-Center Particle Pusher
Lead: da Silva

*STEELIE:
Ionospheric Conductances
Leads: Khazanov, Sur

Funded by LWS Grant thru 2026

*Disclaimer: All logos, names, and backronyms subject to change

2023: Year of Open Science

White House:
Office of Science and Technology

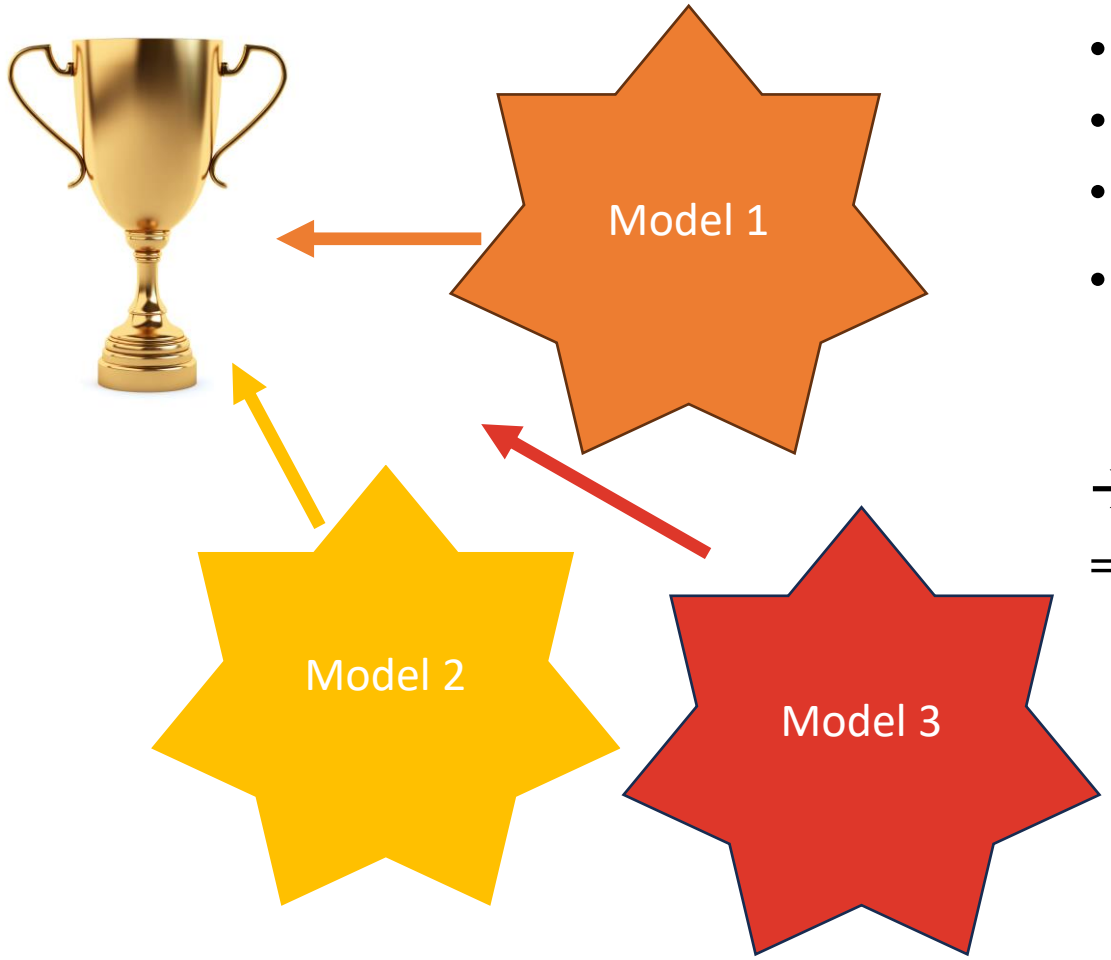
NASA:
Transform to Open Science

From *NASA SPD-41a*:

VII.
C. : "To achieve reproducibility, scientific software developed using SMD funding and used in support of a scientific, peer-reviewed publication **shall be released as open source software** no later than the publication date.

D.: "At the end of the period of performance of a research award, scientific software developed as part of the award, **to extent practicable, shall be released as open source software** if allowed under existing laws and regulation."

… conflicts with Closed Development of Models



Current grant funding system for geospace models:

- Multi-year, decade-long development process
- Closed, secretive competition
- Difficult to use codes without consulting with dev team
- Difficult to add/modify code for one's own purposes (unless you're already a developer!)

→Not entirely compatible with open science principles!
= Findable, Accessible, Interoperable, Reusable

With MARBLE, we have a unique opportunity to build in open-science and open-source from ground up!

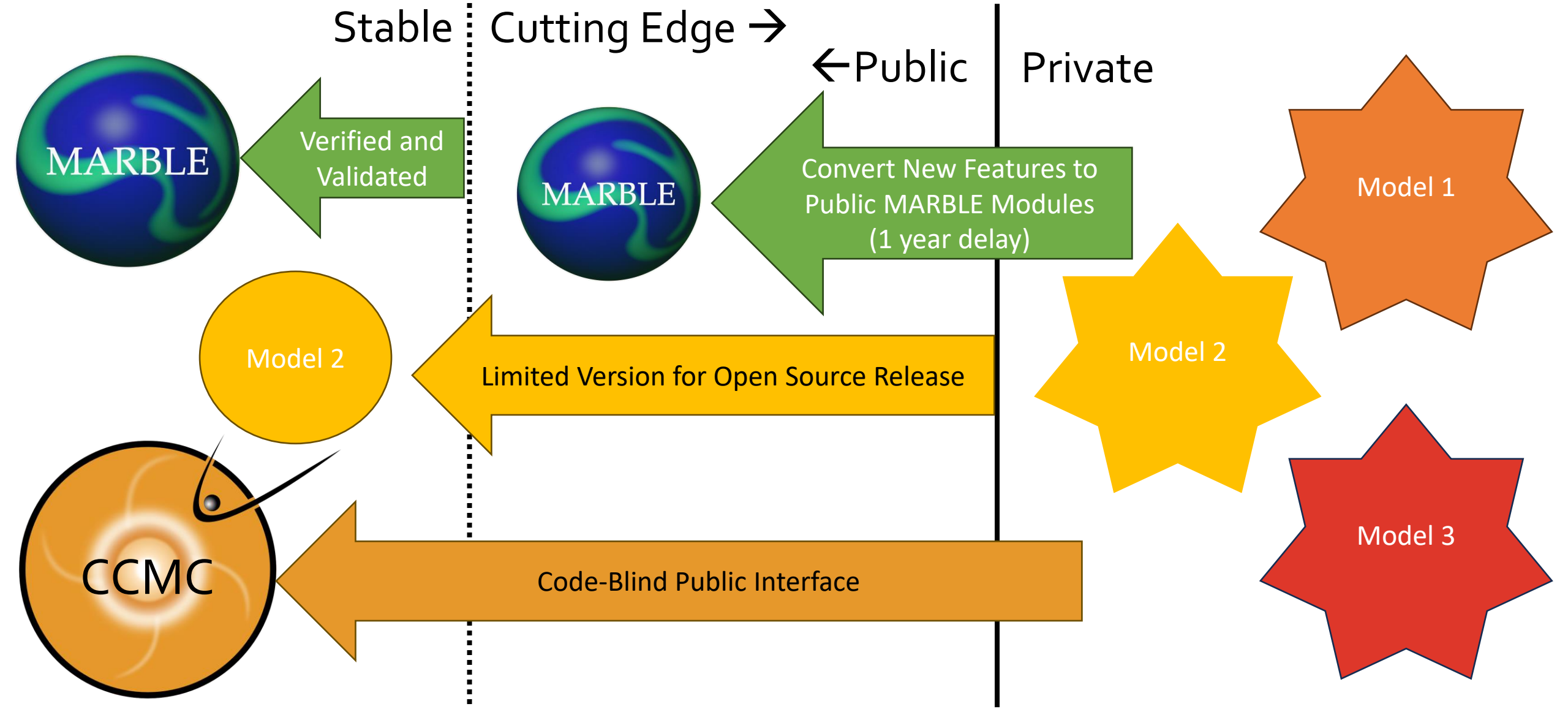
An open-science global magnetosphere code

1. Findable + Accessible
   - Open-source license on public repository
2. Interoperable
   - Needs to work with other community codes/libraries/data analysis pipelines
3. (Re)usable
   - Needs to be fast/efficient, depending on problem
   - Verification and Validation
   - Easy to set up run configurations

A Community Code can also be used for:

- Teaching
- Quick Analysis/Experimentation
- Prototype Development
- Ingestion of Advanced Features

Balancing Closed Development with Open Science



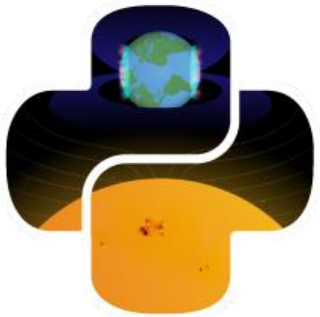
How do you make an open-science global magnetosphere code?

1. Make it easy to use
2. Make it work well
3. Make it easy to modify and append
4. ???
5. Science!

Step 1: Ease of Use

Marble will be written entirely in Python.

- Simple installation via pip and requirements.txt
  - No messing around with Makefiles/Cmake/Intel OMP etc.
- More rapid, easier development compared to C/C++/Fortran
- Cross-compatibility with PyHC ecosystem



sunpy

plasmapy

(and all other Python libraries, e.g. matplotlib)

Step 2: Speed

Although native Python is slow, there are several accelerator libraries:

CPUs/multi-threading

MPI4Py

Graphics Processing Units

PyCUDA

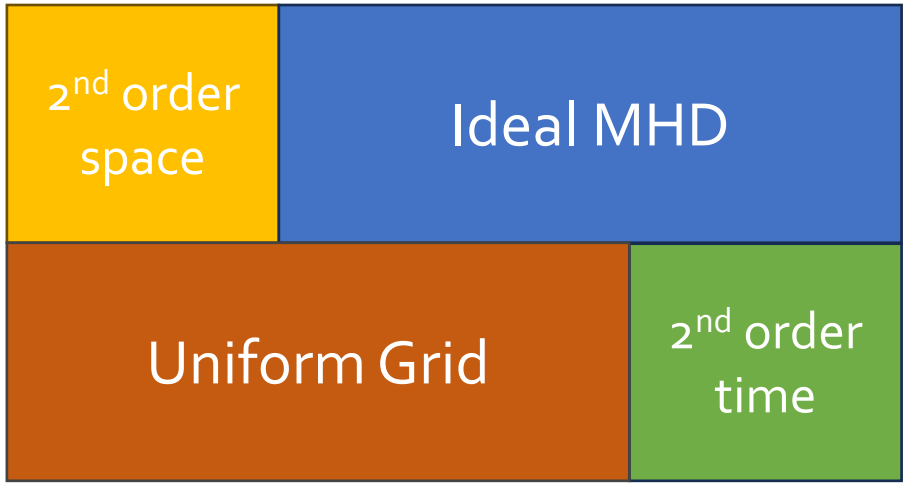
CUDA Python

Computers of the future will likely utilize “heterogeneous architectures”: run on CPUs and GPUs simultaneously!

Step 3: Modular Development

Sub-modules provide balance between:

Flexibility



Simplicity

A second-order scheme that solves the ideal MHD equations on a uniform grid

Modules also allow for ease of adding new features, e.g.:

Kinetic MHD

Adaptive Mesh Grid

Step 4: ???

Other challenges in community code model for global codes:

- How do we balance "needing to publish" with "openly sharing"?
  - Need to encourage community to value software development efforts like they do paper publications
- How will it be funded long-term?
  - E.g. MARBLE project ends in 2026; will it continue with volunteer work?
  - Infrastructure funding?

Step 5: Science!

A Python community global magnetosphere code will enhance science return!

- Teaching/Training:
  - simulation codes and algorithms
- Flexibility
  - Applications to many use cases
- Innovation/Assimilation:
  - Easier to prototype and rewrite in Python
  - Push new features back to community repo
- Ease of use/co-existing with PyHC libraries
- Reproducibility