SWiG: Open-source Empirical Solar Wind Generator

github.com/predsci/swig

Ronald M. Caplan,
Jon A. Linker, Cooper Downs, Pete Riley, and Miko M. Stulajter

Predictive Science Inc.

SWiG

NASA NSF

DASH

SWiG, an Open-source Empirical Solar Wind Generator

- SWiG is an open-source empirical solar wind generator based on a PFSS+CS model of the coronal magnetic field, and includes both WSA and DCHB models
- Many WSA models exist and some are available for runs-on-request at CCMC
- As part of an SWQU project, lower heliospheric solar wind boundary conditions were needed, and the project had an open-source requirement
- We therefore extracted/adapted/updated the techniques/codes used in the CORHEL model suite into a new open-source code package called SWiG

SWQU

Space Weather with Quantified Uncertainty

`swig.py` =

`cor_pfss_cs_pot3d.py` — POT3D
Computes PFSS+CS model

`mag_trace_analysis.py` — MAFFL
Performs field-line tracings

`eswim.py`
Calculates solar wind model

PFSS+CS Model for Coronal Magnetic Field

Maxwell equation: $\nabla \times \vec{B} = \mu_0 \vec{J}$

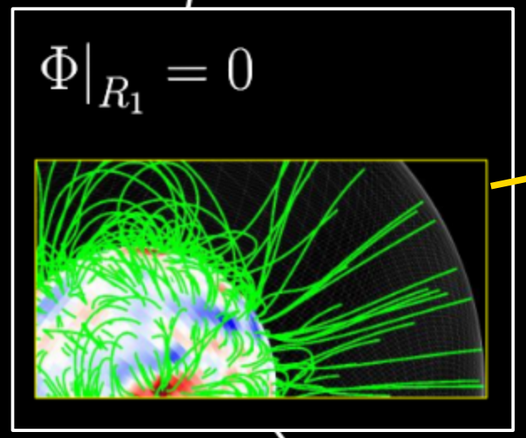
Assume no current: $\vec{J} \rightarrow \nabla \times \vec{B} = 0$

Solution form: $\vec{B} = \nabla\Phi$

Divergence-free condition: $\nabla \cdot \vec{B} = 0$

$$\nabla^2\Phi = 0$$

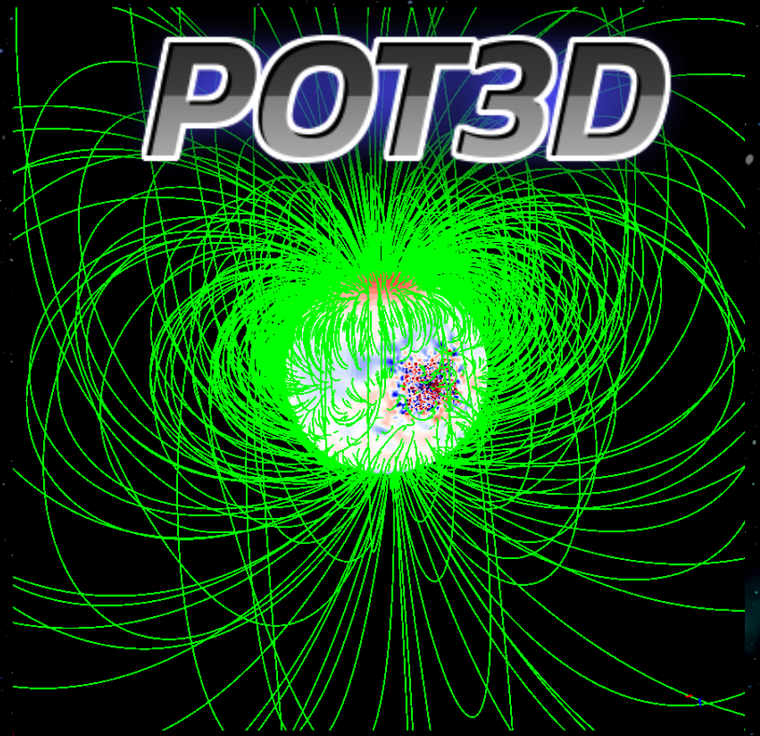
Laplace equation

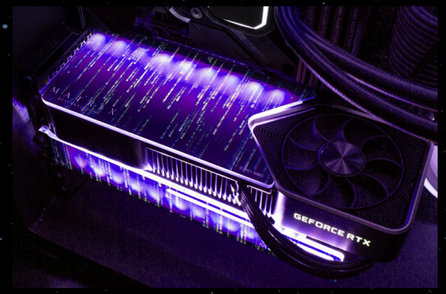


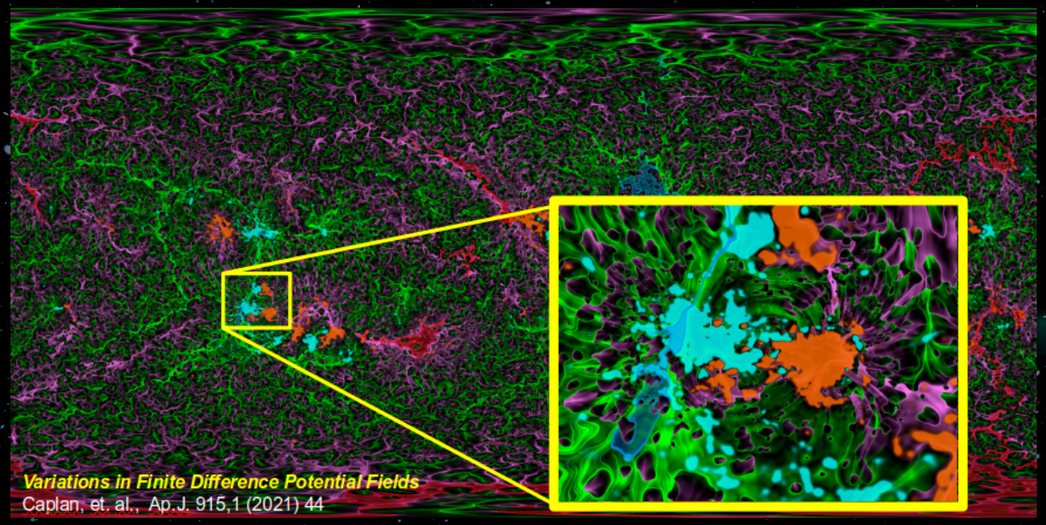
Observed Surface

$$\left.\frac{\partial\Phi}{\partial r}\right|_{R_\odot} = \left. B_r \right|_{R_\odot}$$

- POT3D is a code that computes potential field approximations of the solar coronal magnetic field using observations of the solar surface magnetic field as a boundary condition
- The code is parallelized for use on CPUs and GPUs using MPI+OpenMP and Fortran Standard Parallelism (do concurrent)
- The HDF5 file format is used for input/output



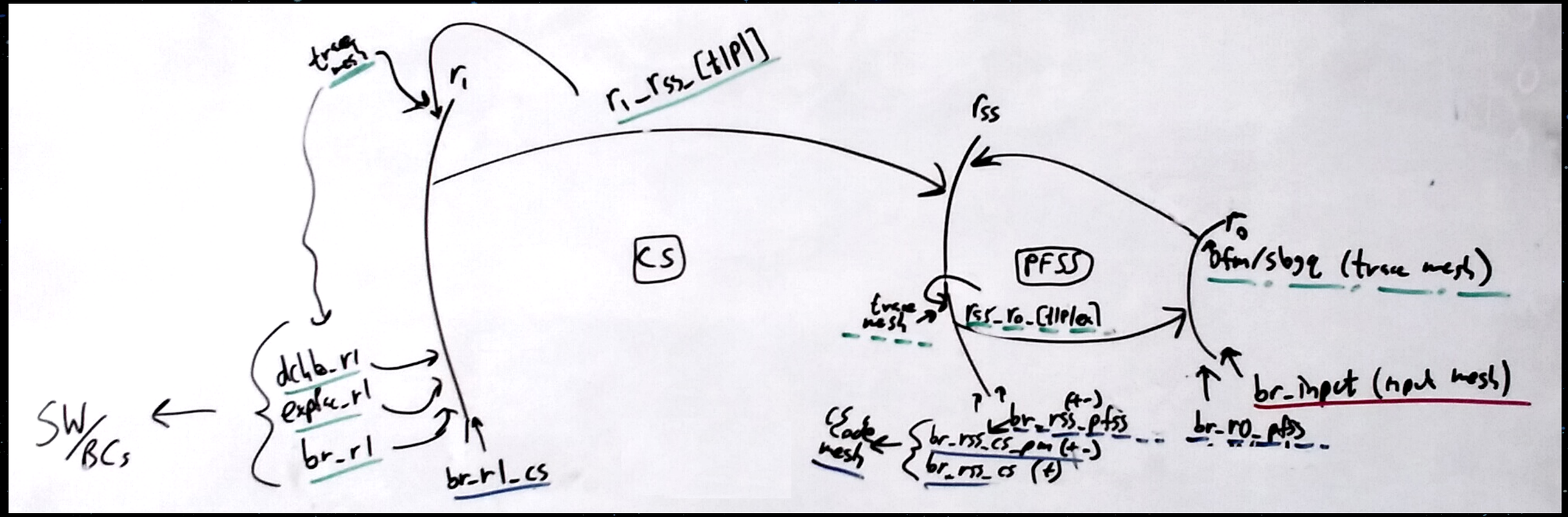


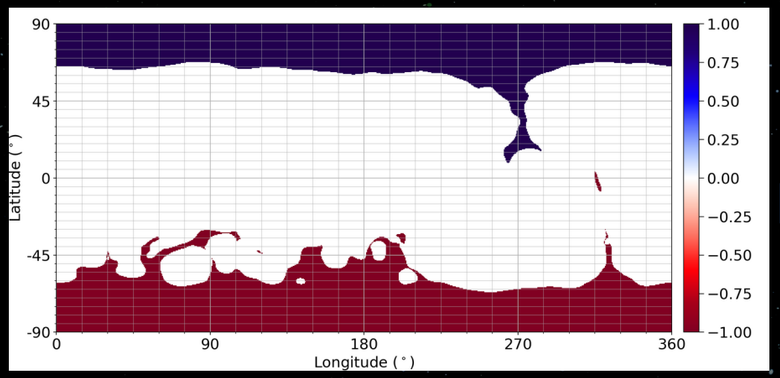


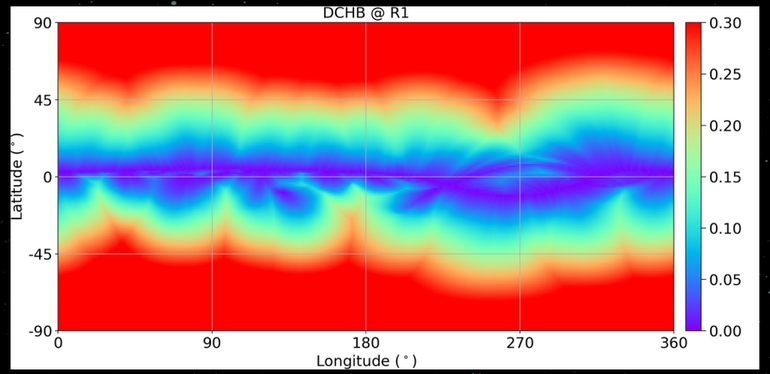
github.com/predsci/pot3d

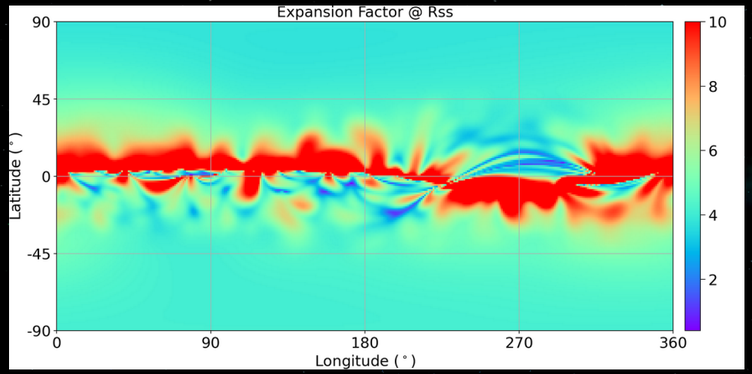
DCHB and Expansion Factor

The empirical solar wind models in SWiG require the distance to coronal hole (open field) boundaries (DCHB) and/or the expansion factor (from R0 to Rss)



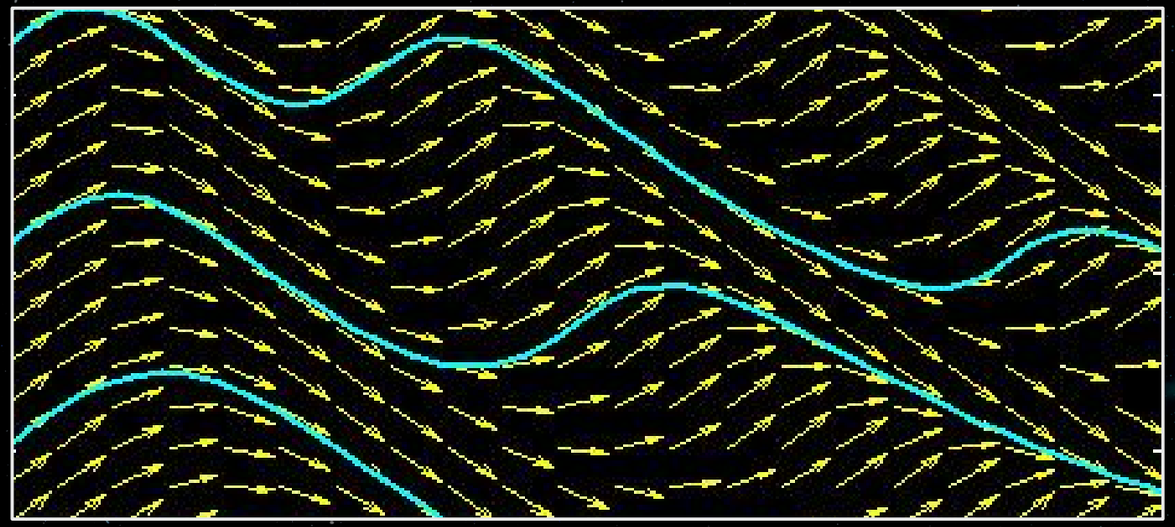






MAPFL

- Important to have accurate field line tracing
- MapFL is a Fortran code that traces field lines through a 3D field defined on a non-uniform spherical gird
- MapFL uses an adaptive tracing step size with a 2nd-order predictor-corrector scheme



Parallelized across multiple CPU threads using OpenMP

github.com/predsci/mapfl

Empirical Solar Wind Models (ESWiM)

- With DCHB and Expansion Factor, compute Vr
- Python script **`eswim.py`** does this using 3 models
- With Vr, set empirical density and temperature: $\rho = \rho_{\text{fast}} \left(\frac{\max(V_r)}{V_r} \right)^2 \quad t = t_{\text{fast}} \left(\frac{V_r}{\max(V_r)} \right)^2$

`wsa:`

$$V_r = V_{\text{slow}} + \frac{V_{\text{fast}}}{(1 + f_{\text{exp}})^{C_1}} \left(1 - C_2 \exp \left[- \left(\frac{\Theta_B^{\text{deg}}}{C_3} \right)^{C_4} \right] \right)$$

$$V_r = \min \left[V_r, V_{\text{max}} \right]$$

`wsa2:`

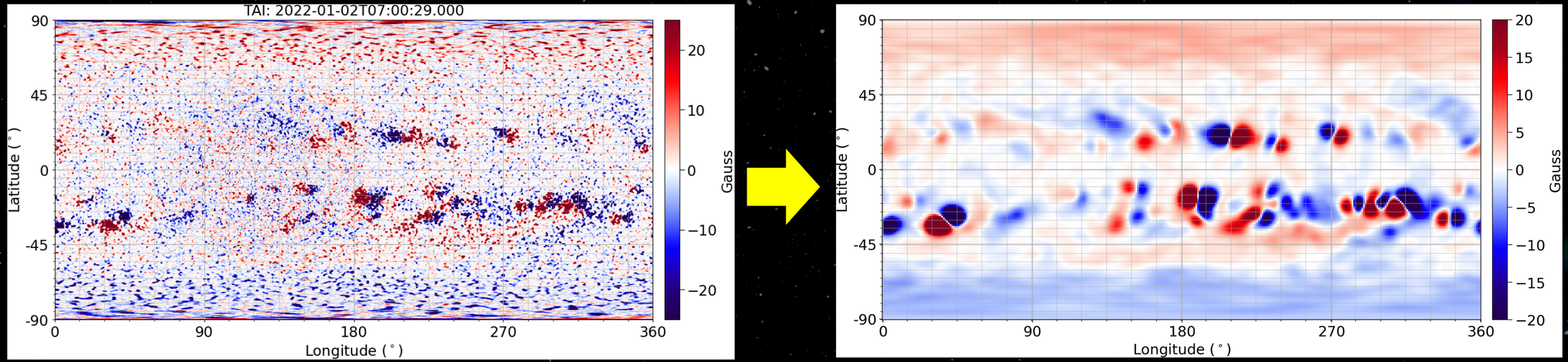
$$V = V_{\text{slow}} + \frac{V_{\text{fast}}}{(1 + f_{\text{exp}})^{C_1}} \left(1 - C_2 \exp \left[- \left(\frac{\Theta_B^{\text{deg}}}{C_3} \right)^{C_4} \right] \right)^{C_5}$$

`psi(DCHB):`

$$V_r = V_{\text{slow}} + \frac{V_{\text{fast}} - V_{\text{slow}}}{2} \left(1 + \tanh \left[\frac{\Theta_B^{\text{rad}} - C_\epsilon}{C_w} \right] \right)$$

Running SWiG: Process the Map!

- PREREQUISITE! Make sure to process the input map so structures are resolved for the resolution of the map
  (if map is under-resolved, field line tracing may fail)
- An installation of our public Open-source Flux Transport (OFT) (**`github.com/predsci/oft`**) includes map processing capabilities with the **`psi_map_prep.py`** script



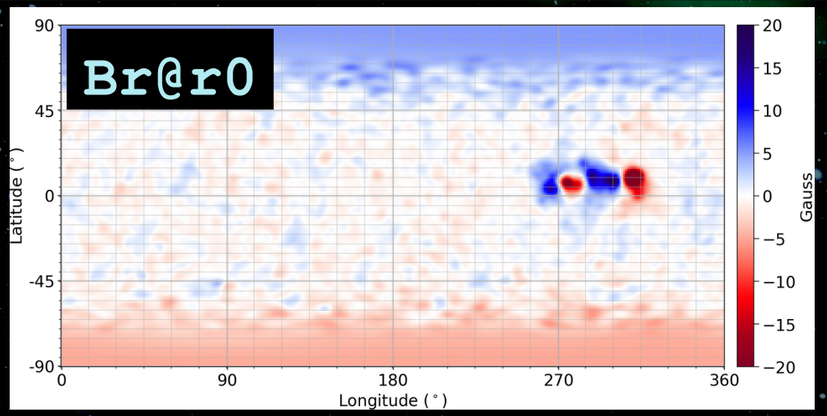
```
swig.py [-h] [-oidx OIDX] [-rnum RNUM] [-rundir RUNDIR] [-np NP]
        [-sw_model SW_MODEL] [-sw_model_params SW_MODEL_PARAMS]
        [-rss RSS] [-r1 R1] [-r0_trace R0_TRACE] [-noplot]
        input_map
```

```
swig_run_multiple_maps.py [-h] [-outdir OUTDIR]
                          [-swig_path SWIG_PATH] [-np NP]
                          [-sw_model SW_MODEL]
                          [-sw_model_params SW_MODEL_PARAMS]
                          [-rss RSS] [-r1 R1]
                          [-r0_trace R0_TRACE] [-noplot]
                          input_directory
```

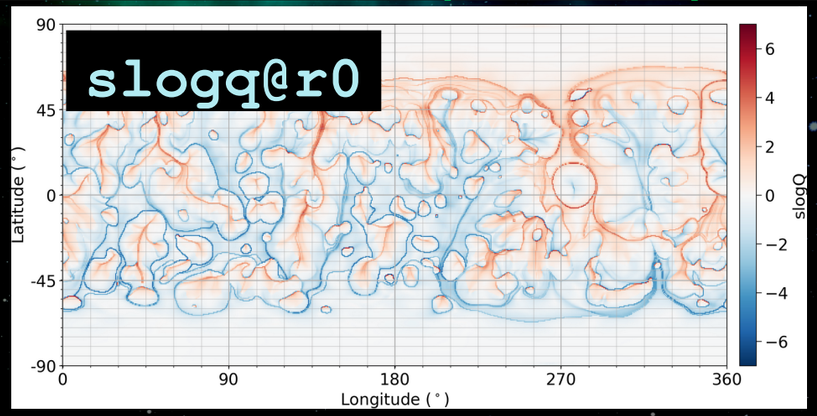
More options/features can be added (pull requests welcome!)

SWiG: Example Outputs

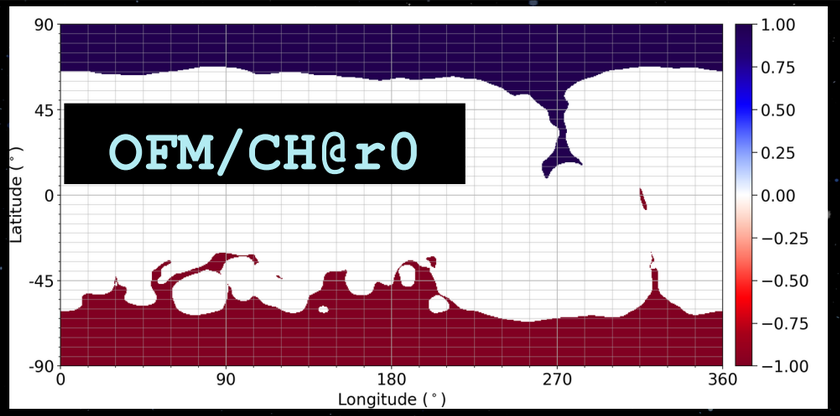
Br@r0



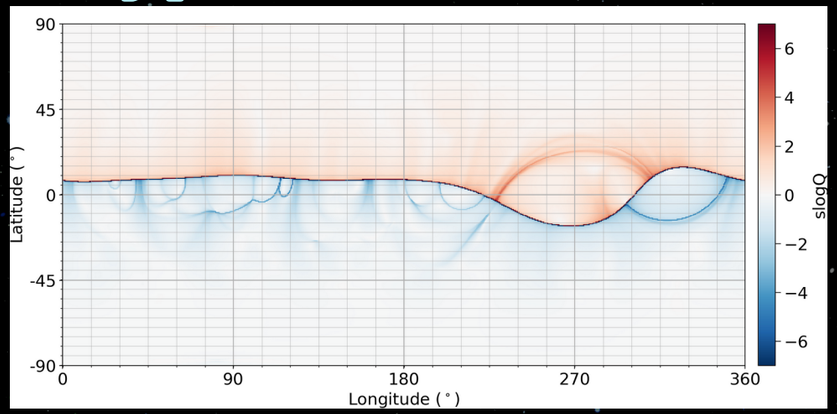
slogq@r0



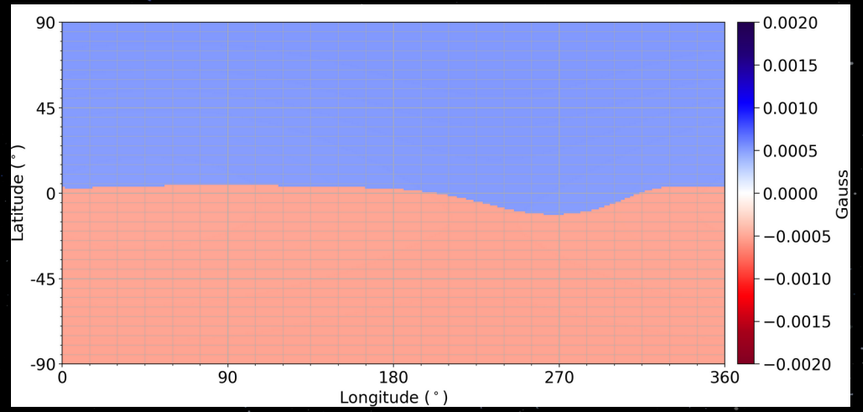
OFM/CH@r0



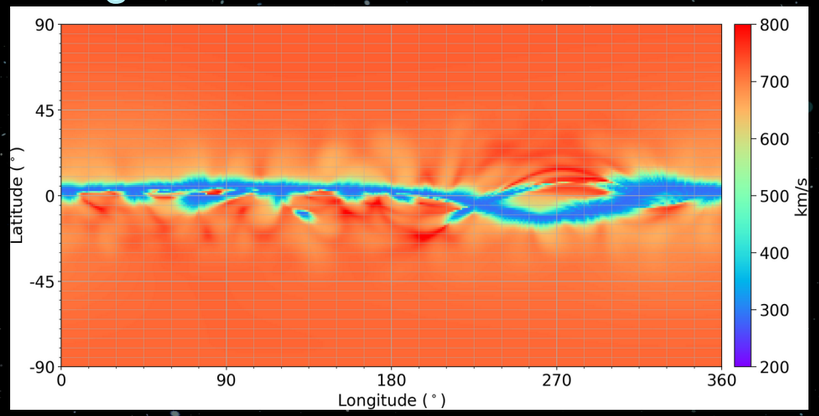
slogq@rss



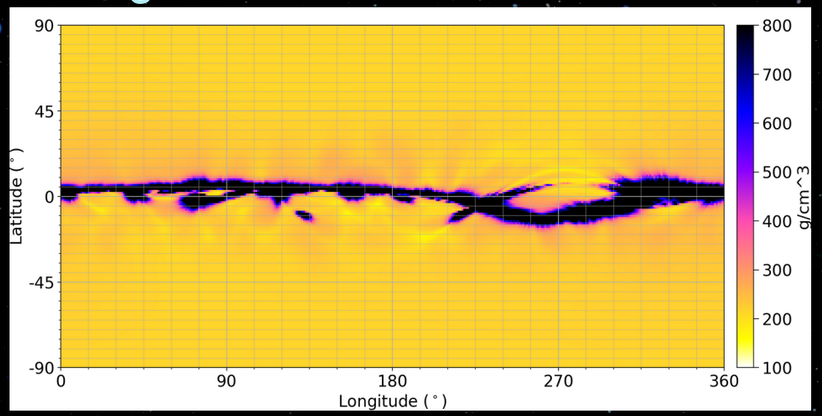
Br@r1



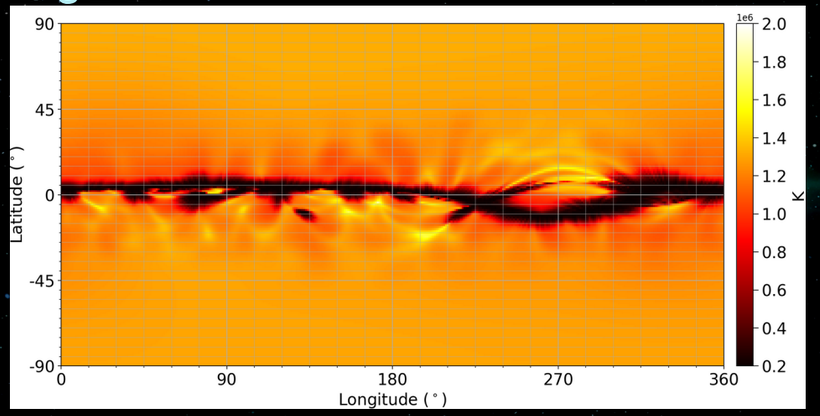
Vr@r1



rho@r1



t@r1



Summary

SWiG

github.com/
predsci/swig

SWiG is an open-source empirical solar wind generator based on a PFSS+CS model of the coronal magnetic field, and includes both WSA and DCHB models

Requires:
- Python 3 with some required packages
- Fortran Compiler
- HDF5 Library
- MPI Library

Works on Linux, Mac, and Windows (with WSL)
Can use NVIDIA GPUs for potential field calculations
On a Laptop with a GeForce 4070 GPU, SWiG can run with 1-degree resolution in ~100 seconds