MAX-PLANCK-INSTITUT
FÜR PLASMAPHYSIK

Overview of the ATEP workflow integration - TSVV-H

Virgil – Alin Popa

Collaborators: Ph. Lauber, T. Sterkl, T. Hayward-Schneider

EUROfusion

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Objectives

- Implement a MUSCLE3 compliant workflow that encapsulates the ATEP physics
- Workflow needs to run standalone and inside a larger transport code
- Incorporate various physics to be investigated after each step of the workflow
- Validate ATEP on several scenarios

Overview

- **What is done:**
  - General layout/implementation for the ATEP workflow
  - Moved all the actors (HELENA, LIGKA, HAGIS, Orbit FInder and ATEP) to MUSCLE3
  - Saved results after each actor/time-step
  - Take decisions on what parts of the workflow to run in real time

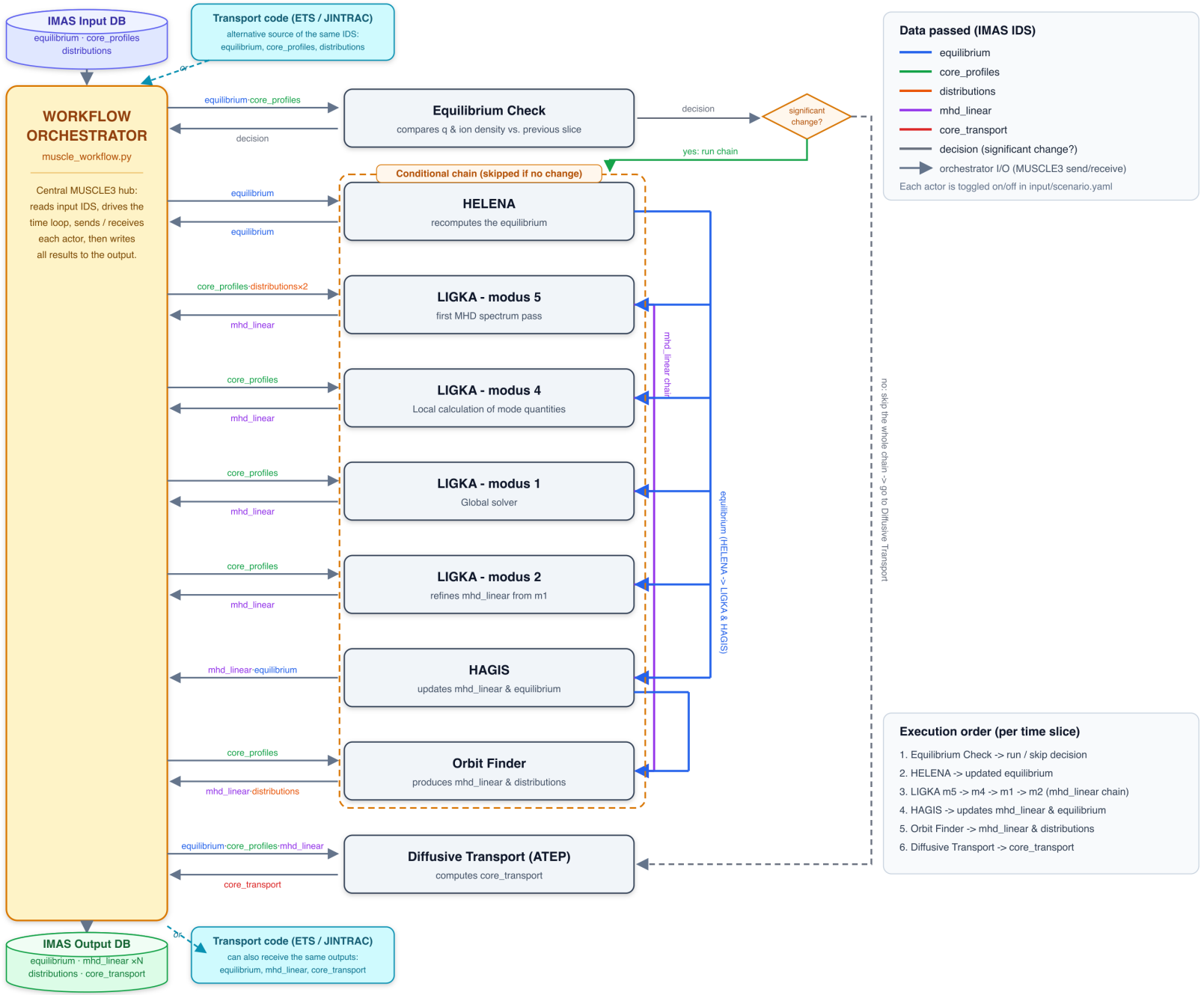
- **What is ongoing:**
  - Porting of ATEP-3D code to Python (fine-tuning ongoing)
  - Integrate ATEP in ETS/HFPS
  - Validation of the ATEP code

Hierarcy of models

- Models depend on eachother, as each one improves/adds on the results given by the previous one.
- From the analytical solver in Ligka_M5 to the refined global solver Ligka_M2
- Hagis_1 convers HELENA equilibrium and mhd_linear given by LIGKA to boozer coordinate system
- Orbit finder is used to calculate neoclassical and diffusion coefficients.
- This allows for easier debugging and saving of the results at each actor/time-step for validation.

ATEP Workflow - MUSCLE3 Data Flow

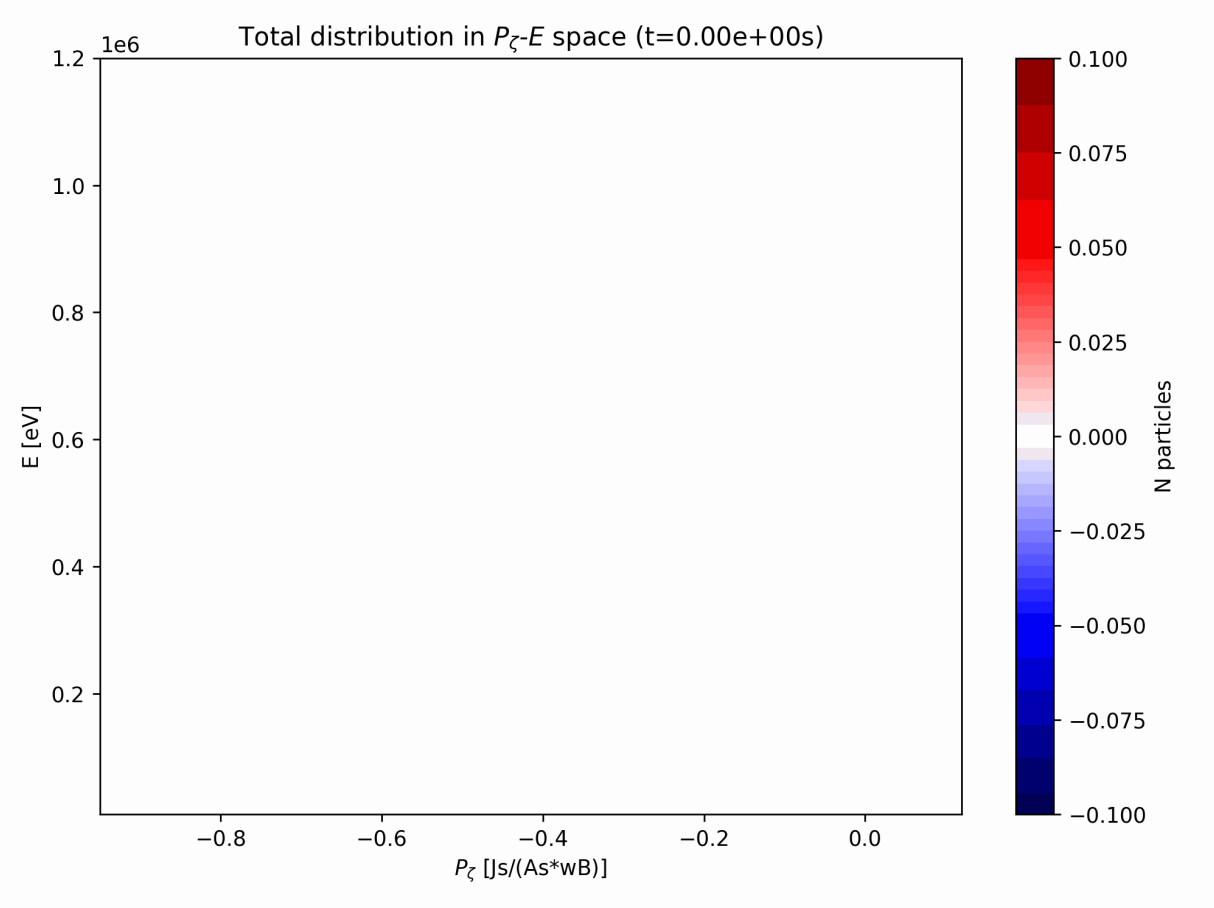
How IMAS IDS data is passed between the orchestrator and the physics actors (defined in coupling.ymmsl)

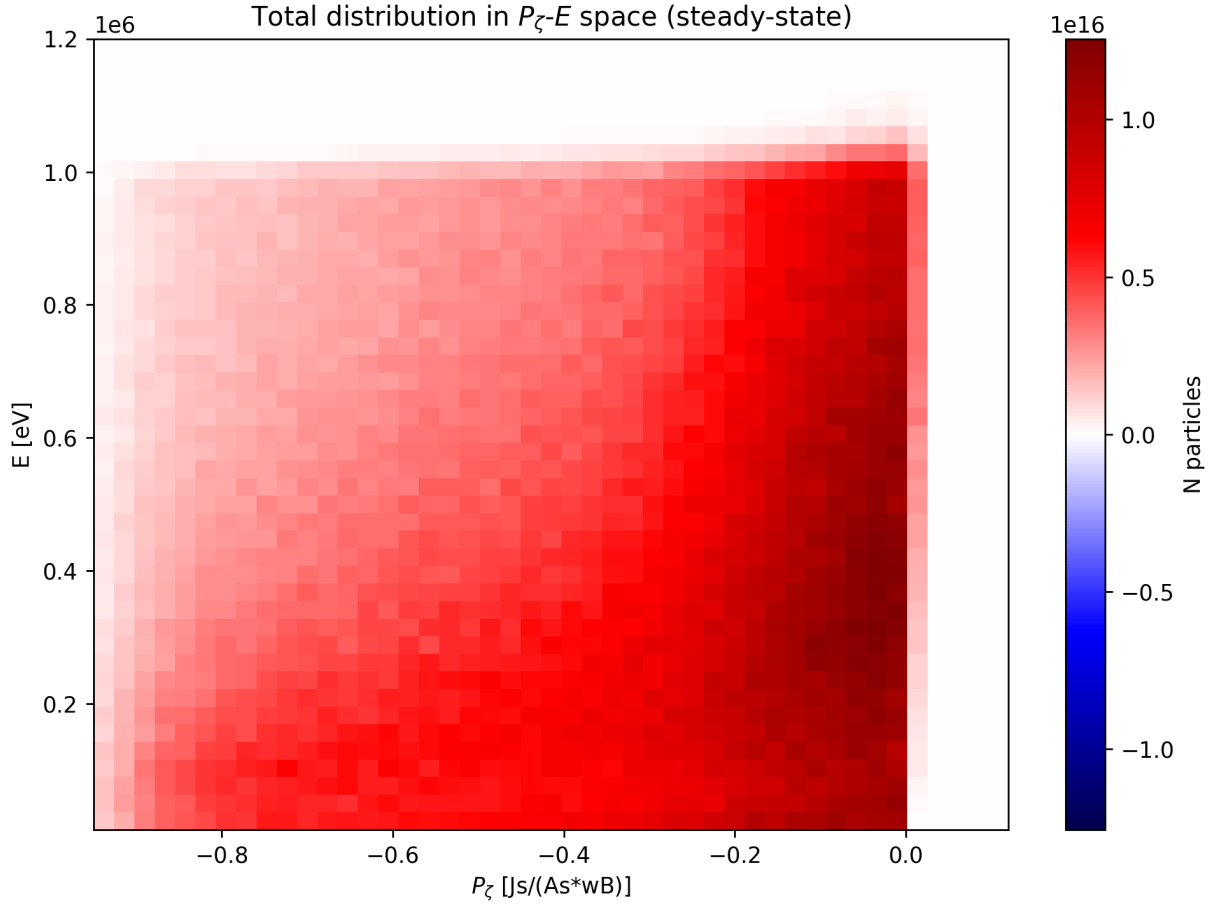


- Starting with all the parts of the workflow enabled, one by one these can be removed and the coupling.ymmsl file will be adjusted automatically.

ATEP - Python implementation - ongoing

Tristan Sterkl - IPP





- Off by a factor of 3, investigation ongoing.
- Checking: input files, normalizations, mapping of IDS data (markers) to solver
- ASCOT/BBNBI IDSs sources given by Chiara De Piccoli

Connecting to the transport code (inputs set, outputs…)

- **Various ways to tackle:**
  - Via core_sources IDS: — give source term for ion density equation and source term for the ion energy transport equation. (source(i1)/profiles_1d(itime)/ion(i2)/energy(:), source(i1)/profiles_1d(itime)/ion(i2)/particles(:))
  - via core_transport IDS: — effective diffusivity (ion density eq) — model(i1)/profiles_1d(itime)/ion(i2)/particles/d(:)) or effective diffusivity (ion energy equation) — model(i1)/profiles_1d(itime)/ion(i2)/energy/d(:)
  - Anything else? Providing the F_EP directly? Are tools available for projection to 1D space?

Discussion

- Integration assistance in ETS/HFPS - need of a simple test case so the ATEP implementation can be tested inside a minimal transport run (preferable ITPA ITER shot 53299)?

- What does a transport code need from us? To be confirmed: heat deposition profile or particle/heat diffusion coefficient?