TSVV3 Regular meeting

# Toolbox for uploading simulation data to IMAS

**Krzysztof Gałązka**

Instytut Fizyki Plazmy i Laserowej Mikrosyntezy,
ul. Hery 23, 01-497 Warszawa

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.

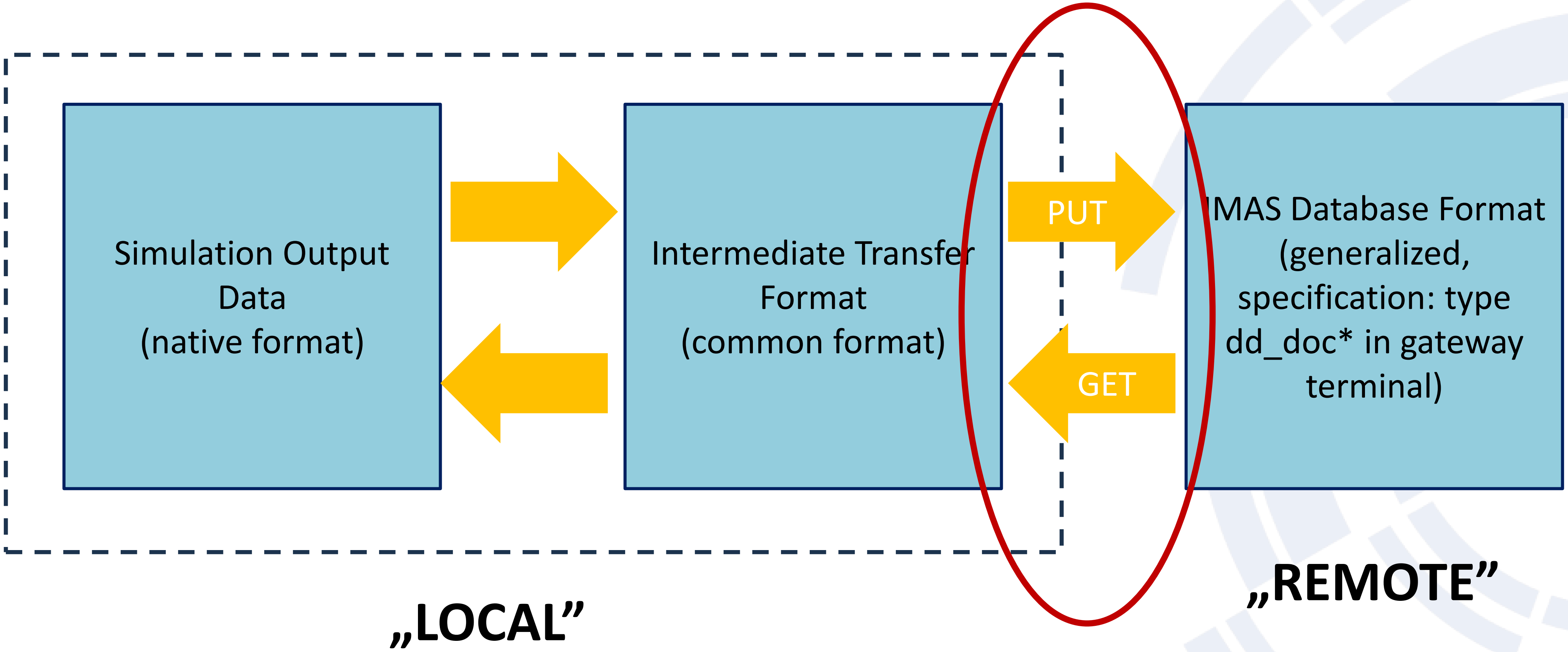
# Outline

- Aim of IMASification work
- General scheme
- Exemplary implementation

# IMASification effort

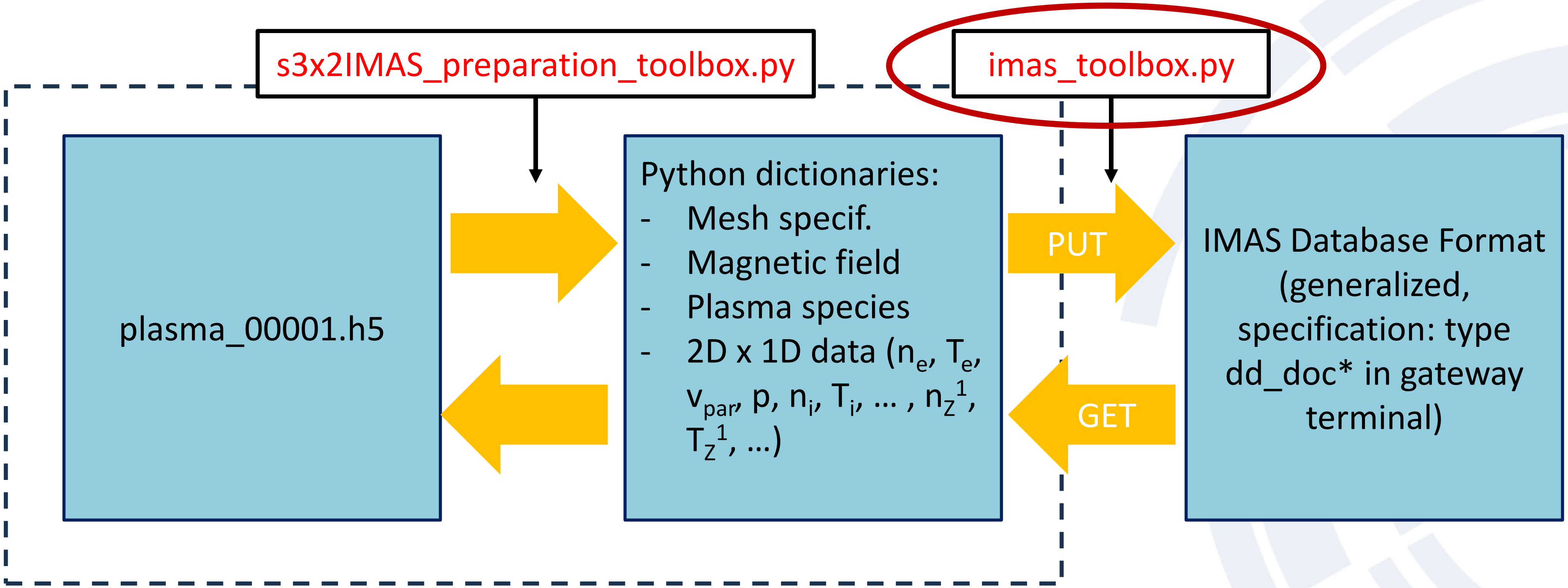
- Standarize the format of output data
  - From experiment
  - From simulation
  - Synthetic diagnostic
  - …
- Allow data interchange between different programs (interface)
- Create a database avaliable to the community

# General scheme


Simulation Output Data
(native format)
Intermediate Transfer Format
(common format)
PUT
GET
IMAS Database Format (generalized, specification: type dd_doc* in gateway terminal)
„REMOTE"
„LOCAL"

* Brings you to: /gw/swimas/core/IMAS/3.31.0/AL/4.8.7/share/doc/imas/html_documentation.html

# General scheme – SOLEDGE3X example



s3x2IMAS_preparation_toolbox.py

imas_toolbox.py

plasma_00001.h5

Python dictionaries:

- Mesh specif.
- Magnetic field
- Plasma species
- 2D x 1D data ($n_e$, $T_e$, $v_{par}$, p, $n_i$, $T_i$, … , $n_Z^1$, $T_Z^1$, …)

IMAS Database Format (generalized, specification: type dd_doc* in gateway terminal)

**equilibrium.py**

**edge_profiles.py**

* Brings you to: /gw/swimas/core/IMAS/3.31.0/AL/4.8.7/share/doc/imas/html_documentation.html

# Intermediate Transfer Format specification (1)

## Dictionaries needed to transfer the mesh and equilibrium data

IMPORTANT:
currently the same 2D poloidal mesh is assumed for all toroidal slices

'griddata' dictionary:

Information on the grid. Generally unstructured.

- **node_p** - 2D array of size Nnodes x 2 providing the R coordinate and the Z coordinate of each node in the poloidal plane
- **cell_p** - 2D array of size Ncells x 4 providing in direct order the indices of the nodes acting as vertices for the quadrangular cells in the poloidal plane
- **node_t** - 1D array of size Nphi-1 specifying the toroidal angle in radians of the toroidal direction coordinate for each poloidal plane.

'mag_field' dictionary:

The components of magnetic field ordered by toroidal slices. In each slice the points are ordered first by radial coordinate then by poloidal.

- **Br** - 2D array of size Nnodes x (Nphi-1) radial component of the magnetic field
- **Bz** - 2D array of size Nnodes x (Nphi-1) z component of the magnetic field
- **Bphi** - 2D array of size Nnodes x (Nphi-1) toroidal component of the magnetic field

# Intermediate Transfer format specification

## Dictionaries needed to transfer 2D (or 3D) physical quantities data: densities, energies …

'atdata' dictionary, keys:

Lists of species present in the plasma. Convention: electrons as index 0, deuterium as 1 and all charge stages of impurities following

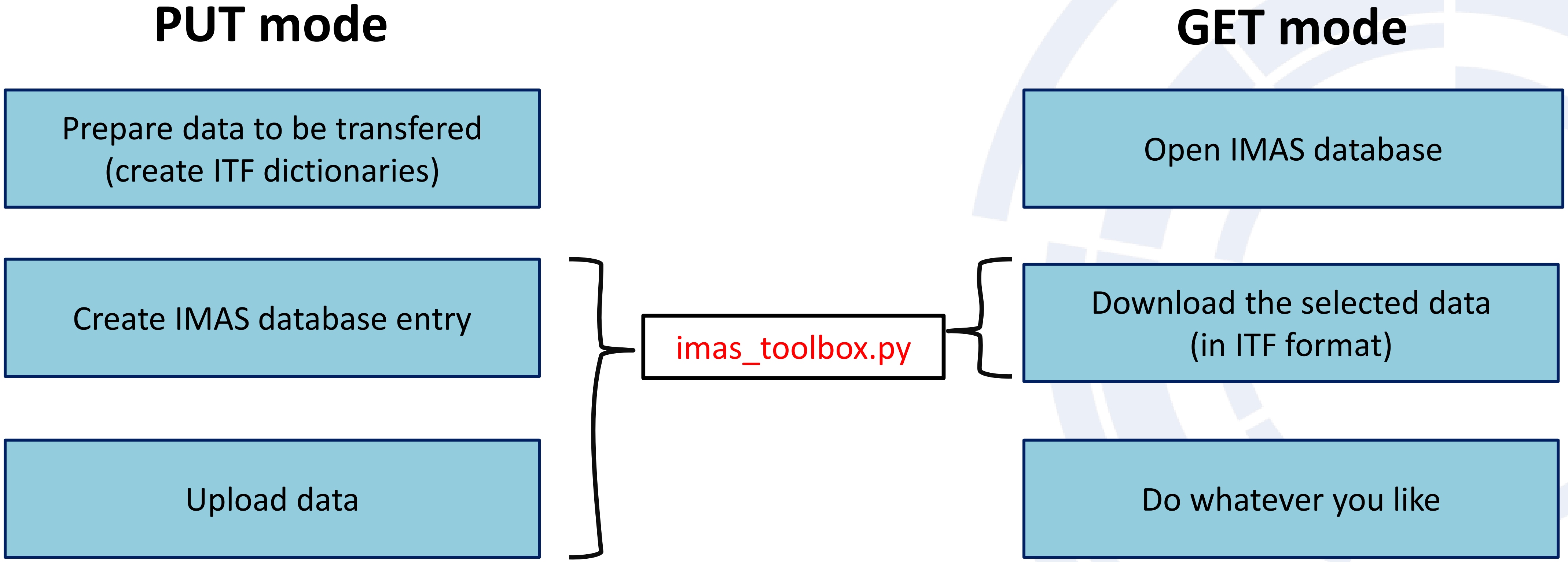
- **ions** - 1D array of all ions with different charge states (labels - chemical symbols)
- **masses** - 1D array with unit masses for each ion charge state (in atomic mass units)
- **charges** - 1D array of different ion charge states
- **Numsp** - 0D, number of species, including electrons and deuterons, each charge state separately

'data' dictionary

Selected data from the simulation output.

- **data** - 2D array of size Nnodes x (Nphi-1) with selected data, organized by toroidal cross-sections and then 1D vectors for all data points. In the 1D vectors the points are ordered first by radial coordinate then by poloidal. Generally unstructured.
- **time** - 0D integer, the time index of output file
- **spec** - 0D integer, index of species of interest, corresponds to the index in atdata, 0: electrons, 1: D+1 ions, ... (secondary importance)
- **field** - string, selected field of interest name, currently accepted: 'n', 'T', 'v_par', 'v_rad', 'p’.
- **charge** - 0D double, charge state, the same as in atdata['charges']
- **mass** - 0D double, mass, the same as in atdata['masses']
- **element_symb** – 0D string, name of the element, the same as in atdata['ions']
- **Npts_pol** – 0D integer, number of points in a single toroidal plane (secondary importance - size of data in the poloidal plane ‚direction’)
- **Npts_tor** – 0D integer, number of toroidal planes (secondary importance - size of data in the toroidal direction)

# Exemplary flow of data upload/download



## PUT mode

Prepare data to be transfered (create ITF dictionaries)

Create IMAS database entry

Upload data

imas_toolbox.py

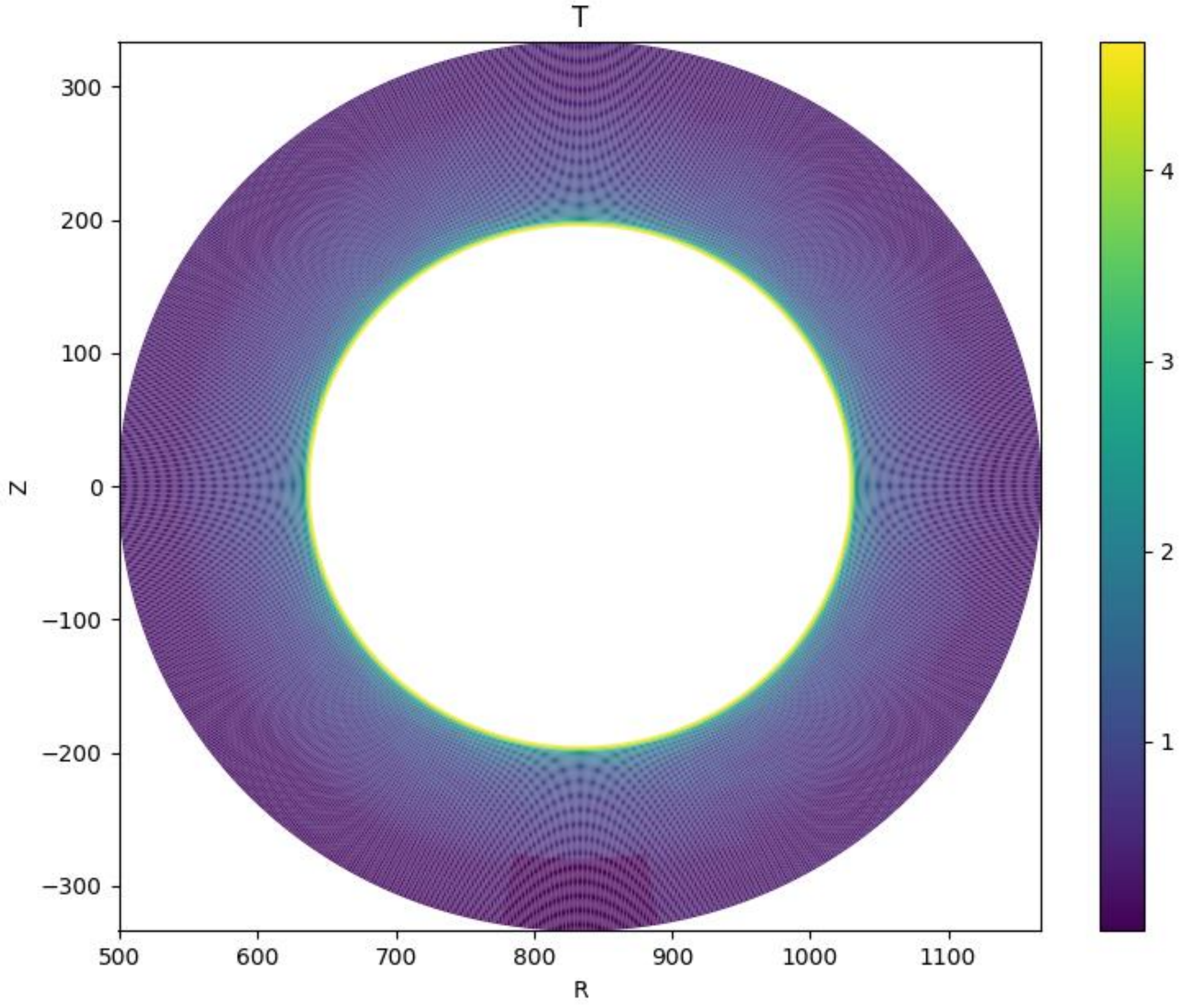
## GET mode

Open IMAS database

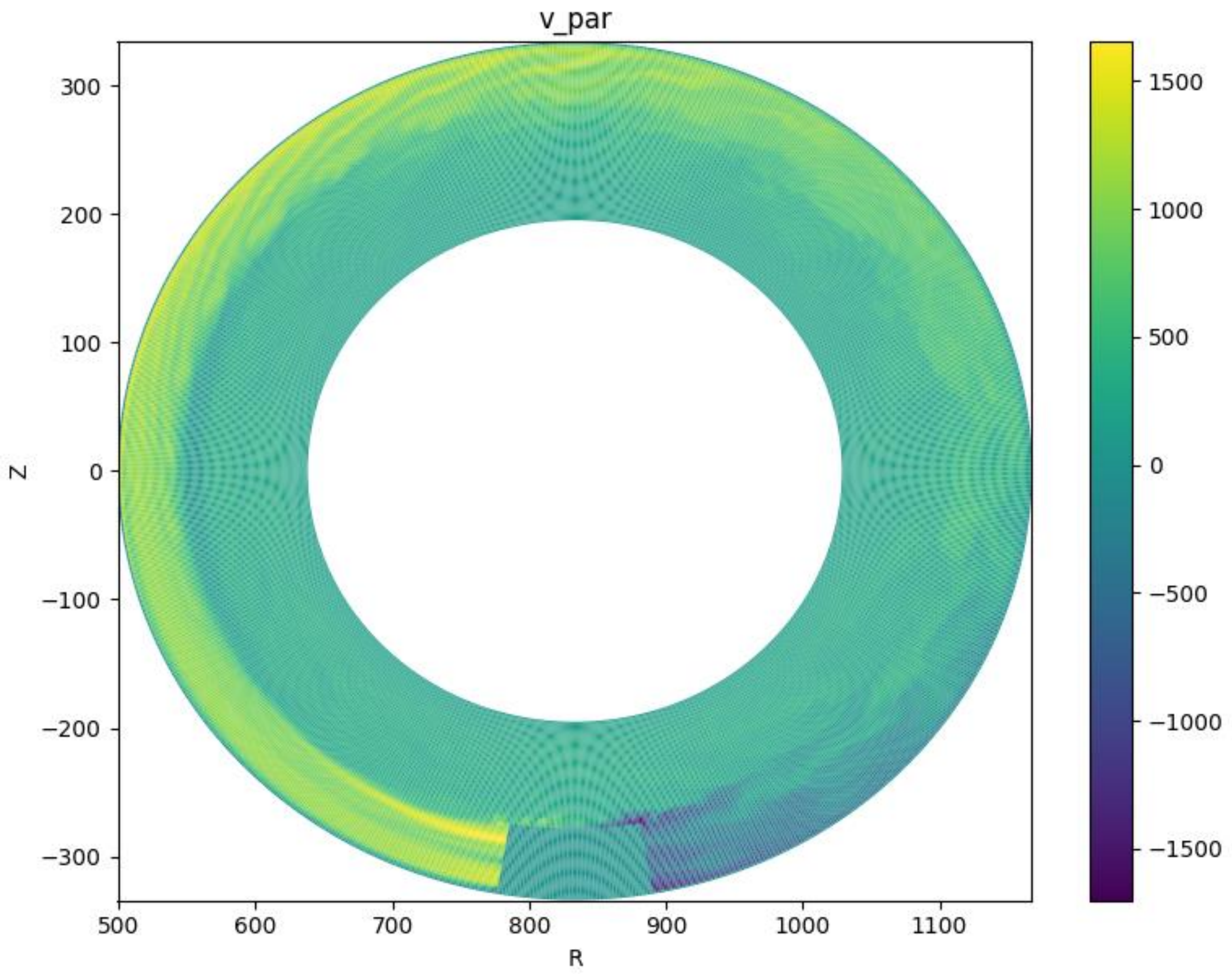
Download the selected data (in ITF format)

Do whatever you like

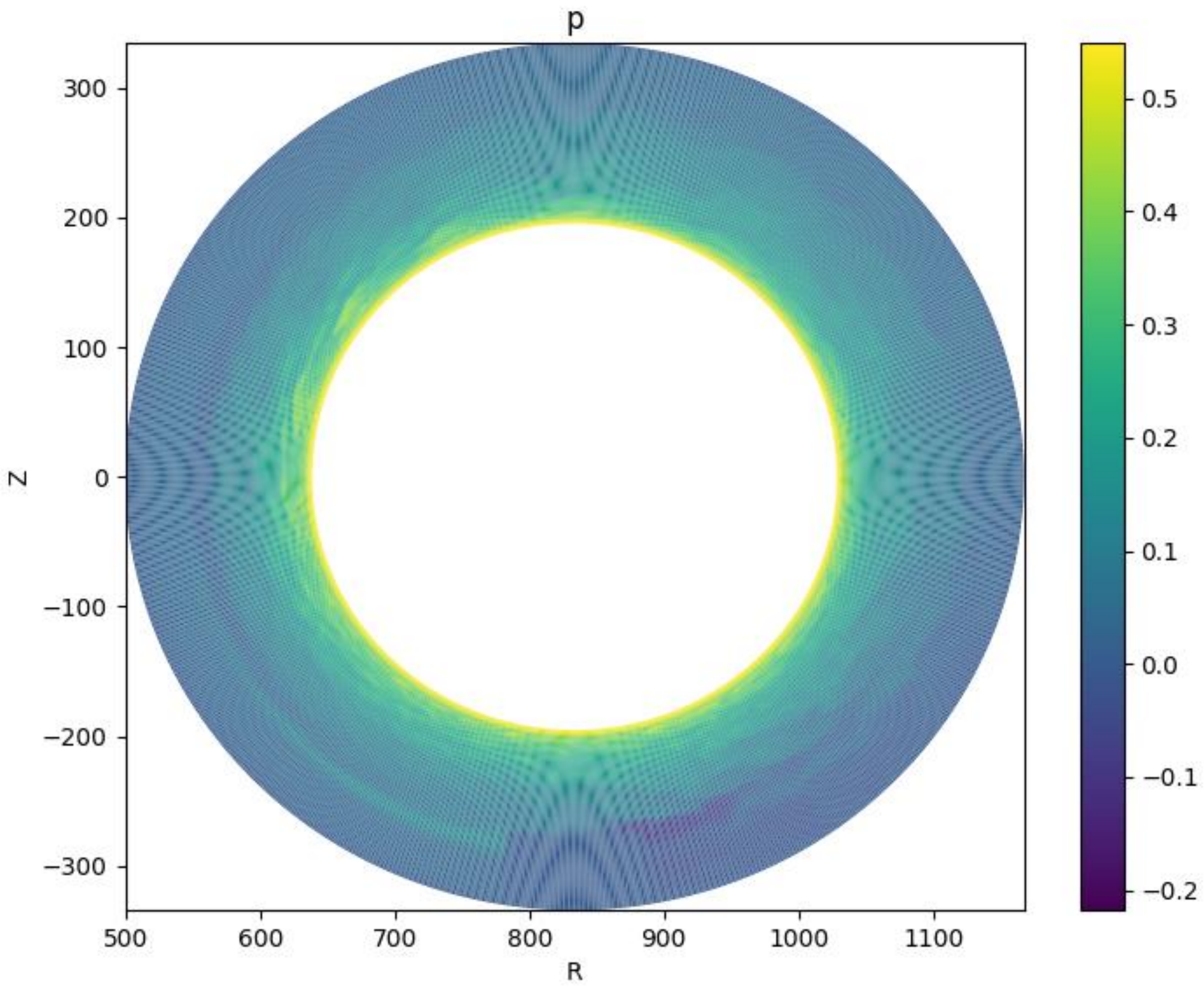
# Data uploaded/downloaded example



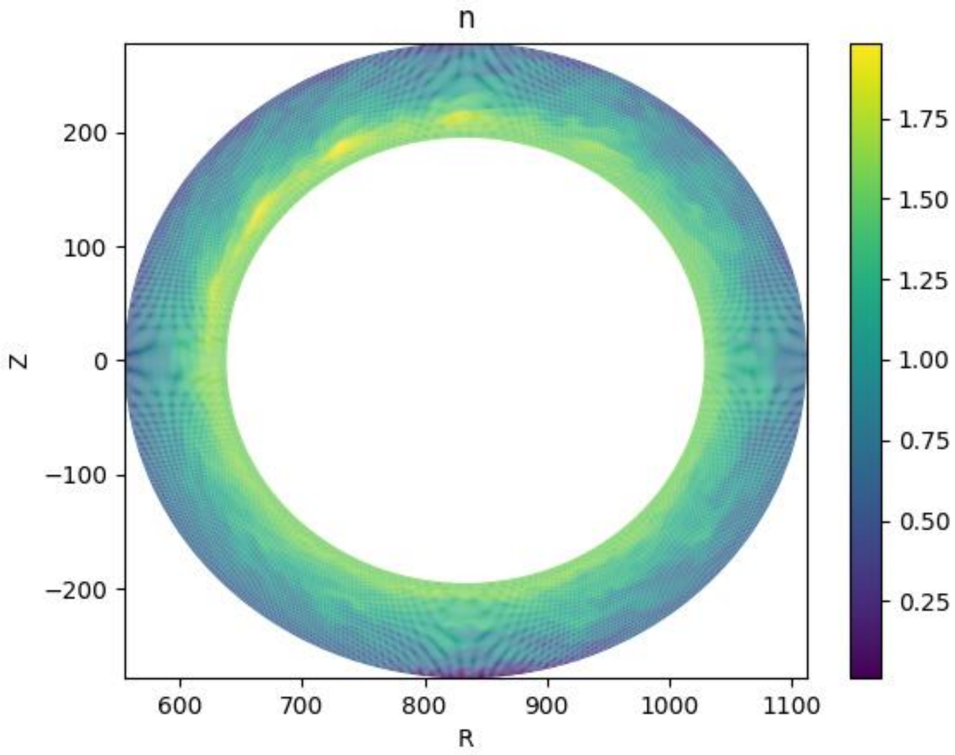
# Data uploaded/downloaded example

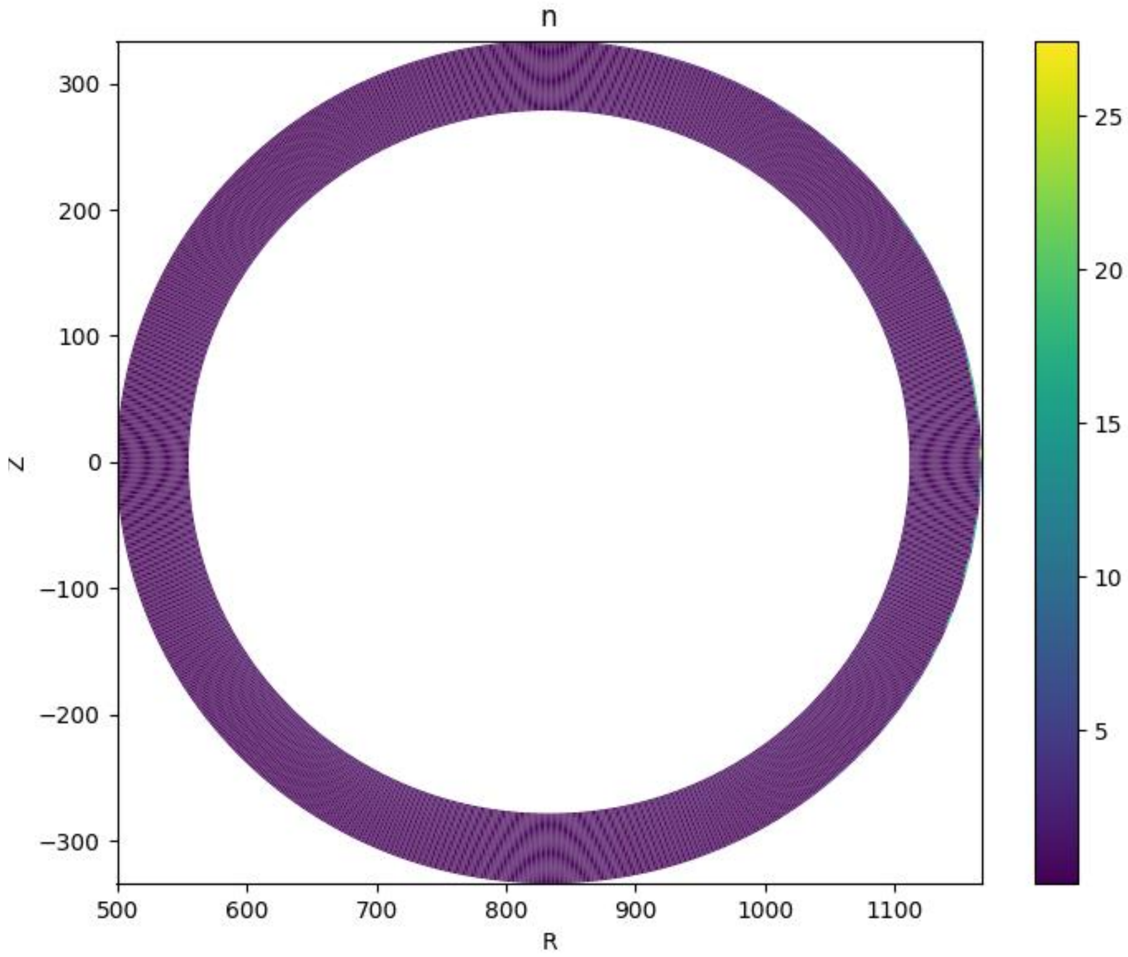


# Data uploaded/downloaded example



# Data uploaded/downloaded example





# Where to find it

Acces by Gateway account/password.

- https://gitlab.eufus.psnc.pl/ach/imas-interfaces/soledge3x

# Thank you for your attention