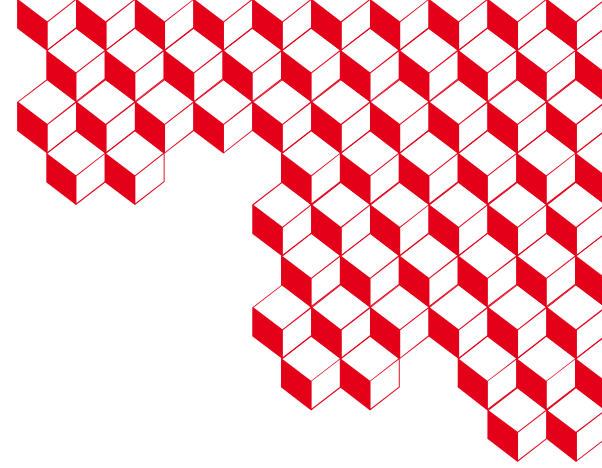




irfm



STYX 2.0, an improved plasma solver-EIRENE interface

TSVV5 EIRENE-NGM Code Camp 2024
20-11-2024

*Nicolas RIVALS (CEA IRFM), Yannick MARANDET (AMU
CNRS), Patrick TAMAIN (CEA IRFM), Hugo BUFFERAND
(CEA IRFM)*



Caveat – Not a release presentation, work on-going: « progress report »

Initial situation: STYX 1.0 interface SOLEDGE-EIRENE interface

- **Goal: Interface code** to manage all **setup** (input file generation and other parameters) and back & forth **communications** with SOLEDGE2D plasma solver and **EIRENE**
- **“STYX” developed initially for SOLEDGE2D** code and then TOKAMK3X (legacy codes, deprecated, both superseded by SOLEDGE3X code)
- Incremental improvements up to 2020
- Many diverse features implemented over the years (hybrid, sheath database, etc.)
- Modified for SOLEDGE3X from 2020- on

→ **Work-horse of SOLEDGE-EIRENE simulations for ~10 years**

- **Slow built up of *technical debt***...
- No modularity, intertwined with both EIRENE and plasma solver → very complex to modify
- Frozen EIRENE version (Dec. 2020, manual updated a few with lot of effort)
- Objectives for new projects with SOLEDGE3X from different parties WEST, ITER, linear machines, etc.: (dynamic wall models H-D, D-T simulations, hybrid neutrals, 3D) **would require unreasonable efforts and expertise** to implement due to technical debt...

→ **Develop new interface leveraging current knowledge, modern, modular, and as user/developer friendly as possible:**

→ **Project STYX 2.0**

Project started in 2023

- Code like this sounds simple...
- But actually **is not at all** (a lot of tasks needs to be done “under the hood”)
- Currently **~10k lines of code**, over a year (though not full time)
- Some “project management” done:
 - **Kick off with scoping workshop in Marseille 2023:**
 - What is / what is not (target/vision)
 - Feature full listing
 - Prioritization
 - Definition of minimal set for v1.



STYX 2.0 SCOPING

Objectives of STYX 2.0:

1. **IS:** an interface between generic plasma solvers and (specifically) EIRENE, but fully independent of EIRENE
2. **IS NOT:**
 - i. an integrated modelling interface for other matters than plasma-surface interactions
 - ii. a hacked version of EIRENE

Features prioritization:

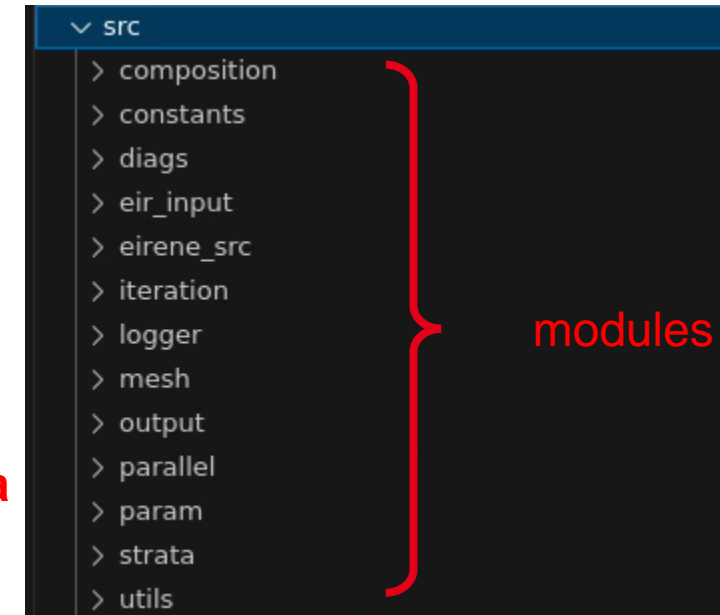
Priority 1 (Minimal set of features)

1. Input:
 - i. JSON input file for STYX
 - ii. write JSON EIRENE input file
 - iii. Plasma conditions & fluxes preprocessor checker
2. Geometry:
 - i. Extruded triangles
 - ii. Independent mesh from plasma solver
 - iii. Mapping tables

STYX 2.0: A generic <plasma solver>-EIRENE interface code

Main features:

- **Modular:**
 - **STYX library** (→ depending on EIRENE library) simply called by plasma solver
 - STYX repository **only includes a link to EIRENE repo** (“submodule”)
- JSON input (also EIRENE JSON input) → GUI module planned
- **Plasma solvers’ only difficult job is mapping/interpolation of plasma mesh to EIRENE mesh** (background plasma-solver → EIRENE, source EIRENE → plasma solver)
- → **Mesh for EIRENE fully independent from the plasma solver** (reduced memory footprint). Plasma solver just has to write the mapping routine.
- Library: Much improved program flow: **initialization phase THEN run phase**, all passed **in-memory**, no file reading, even for the mesh (and as few as possible user ifX_cop calls)



Input file generation less error prone, more flexible

Quality of life improvements: **remove as much choices/free parameters as possible** (where there “reasonably” shouldn’t be):

- Easy/flexible species setup (only requires chemical formula “spec = array[A,Z,mult]”)
- Template-based system (JSON edits, no recompilation required):
 - For reactions
 - For primary sources
 - For surfaces



```
▼ styx_input_files
  ▼ coupled_codes_templates / SOLEDGE3X
    ▼ reactions_templates
      {} H_advanced.json
      {} impurities.json
      {} neutral_neutral_collision_template.json
    ▼ source_stratum_templates
      {} gaspuff_stratum_template.json
      {} recombination_stratum_template.json
      {} recycling_stratum_template.json
    ▼ surface_templates
      {} additional_surface_template.json
      {} grid_surface_template.json
      {} surface_model_template.json
      {} eirene.input.json
      {} TRIM_proj_surf_binding_energy.json
      {} EIRENE_coupling.json
```

```
"JFEX2MX": 0,
"FP1": [ 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0],
"FP2": [ 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0, 0.0E+0],
"REACTION MAPPING": {
  "ISCD1M": 0,
  "IESTMM": 0,
  "EELECM": -2.5E+1,
  "EELEC_REAC_ID": "None",
  "EBULKM": 0.0E+0,
  "ESCD1M": 1.0E+1,
  "EDPOTM": 0.0E+0
},
"ASSOCIATED REACTIONS": [
  {"TRACKED_PARTICLE": "H2", "BULK_PARTICLE": "e^1-", "PRODUCT_1": "H", "PRODUCT_2": "H^1+", "FREAC": 0.0},
  {"TRACKED_PARTICLE": "D2", "BULK_PARTICLE": "e^1-", "PRODUCT_1": "D", "PRODUCT_2": "D^1+", "FREAC": 0.0},
  {"TRACKED_PARTICLE": "T2", "BULK_PARTICLE": "e^1-", "PRODUCT_1": "T", "PRODUCT_2": "T^1+", "FREAC": 0.0}
],
},
```

**EIRENE
reaction format
« extension »**

Outputs

- To plasma solver, with unit choice (SI, or CGS)
- Save to own HDF5 file, or to any existing HDF5 group identifier
- Some plotting routines
- Split by stratum
- Automated checks (particle balance)

Current status & future steps



Current:

- **Close to minimum viable**, should be in first half of 2025. Minimal: no NN collision, no time-dependent, basic short cycling
- **Dummy plasma-solver code working** (in STYX repo)
- Current implementation with SOLEDGE3X on-going
- Merge request on EIRENE side (few changes, mainly exits points to EIRENE_COUPLE)

Future:

- Once first coupling with SOLEDGE3X OK:
 - Add NN-collisions, time-dependent, and rescaled short-cycling (particle balance conservation)
 - Then go to hybrid models, source linearization, H-D-T mixes, interfaces for dynamic wall models...
 - GUI
 - OpenMP tests
 - Test with other plasma solvers



“ Thank you for your attention