

Progress on the KINDNES and SPECTRE projects

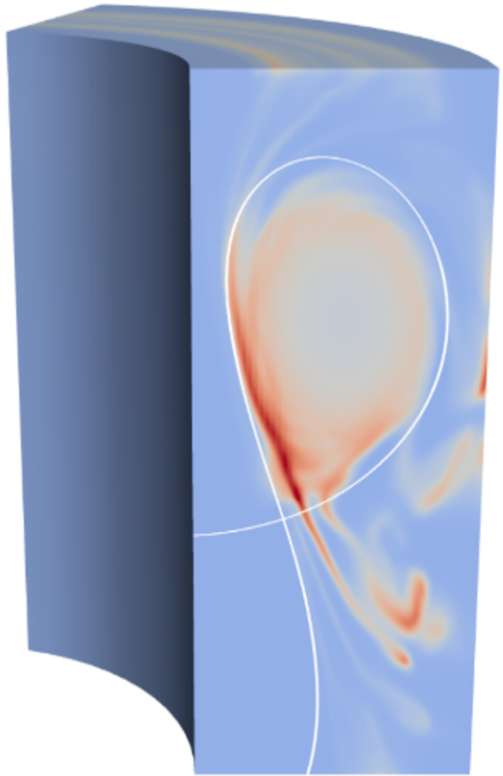
E. Lanti

4th Annual Meeting of EUROfusion HPC ACHs
Tue. 27th of Octobre 2026, Garching

KINetic Deterministic NEutral Solver (KINDNES)

D. Mancini, M. Guido, B. De Lucca, G. Drago, P. Ricci

- Plasma simulations need a self-consistent neutral model
- First neutral model implemented in GBS in 2015 for limited scenarios [C. Wersal et al, NF 2015]
- More than 10 years of development:
  - Used for realistic sources in turbulent simulations and GPI synthetic diagnostics [C. Wersal et al, NF 2017]
  - Extended to diverted scenarios [D. Mancini et al, NF 2021]
  - Multiple neutral species [A. Coroado et al, NF 2022]
  - Qualitative agreement of density shoulder formation, detachment and turbulence diagnostics results [D. Mancini et al, PPCF 2026]
- Kinetic treatment without Monte Carlo noise, *e.g.* EIRENE



The goal of this project is to extract GBS' neutral module into the KINDNES library

Creating KINDNES out of GBS

- Choose the minimum subset of neutral features
  - Features added back afterwards
- Clear inventory of data, flows, and dependencies
- Chose a clear API / boundaries between plasma/neutral worlds
  - Think general, not GBS-centered
  - Minimize plasma code decisions into KINDNES

Creating KINDNES out of GBS

- Choose the minimum subset of neutral features
  - Features added back afterwards
- Clear inventory of data, flows, and dependencies
- Chose a clear API / boundaries between plasma/neutral worlds
  - Think general, not GBS-centered
  - Minimize plasma code decisions into KINDNES

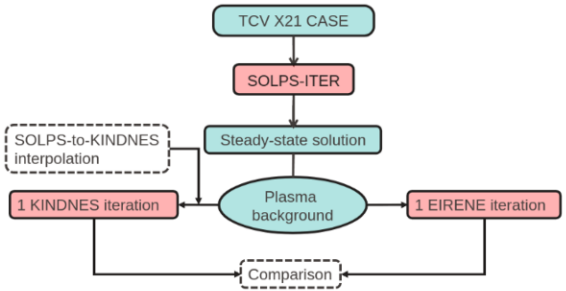
- Initial goal: remove dependencies on plasma parts until standalone
  - Use of temporary middle interface mapping old/new data formats
- Polish the interface and workflow, add back initially dropped features

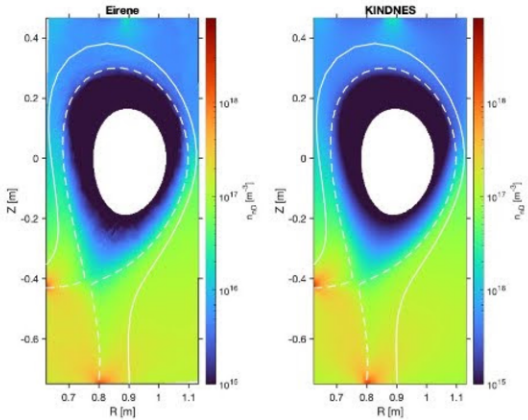
Testing KINDNES

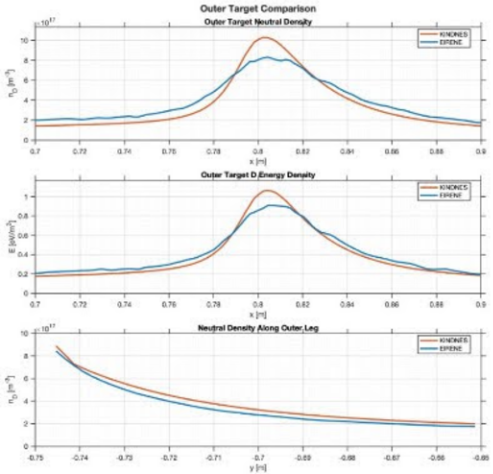
- Comparison between original GBS and GBS + KINDNES
  - In particular, given the same input plasma moments, same output neutral moments

Testing KINDNES

- Comparison between original GBS and GBS + KINDNES
  - In particular, given the same input plasma moments, same output neutral moments
- Link KINDNES to other codes
  - SOLPS + {KINDNES, EIRENE}
  - Good agreement between two codes







From dense to data sparse

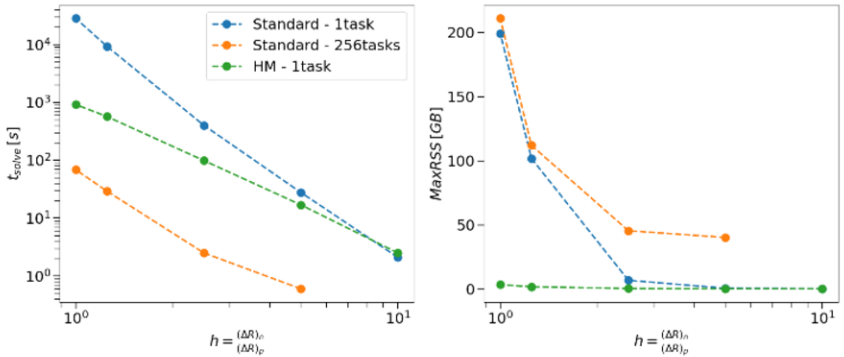
M. Guido

- Neutral solved using method of characteristics
- System size prop. to $(N_x \times N_y)^2$

From dense to data sparse

M. Guido

- Neutral solved using method of characteristics
- System size prop. to $(N_x \times N_y)^2$
- Use low-rank approximation and data-sparse representation (H-matrices): N log(N)



- Currently limited to a serial implementation

Conclusions and outlooks

- GBS' neutral module has been extracted into the KINDNES library
- Coupling tests with other plasma codes
- Successfully tested against original GBS and SOLPS+EIRENE
- Implementation of a data-sparse approximation solving the memory requirements

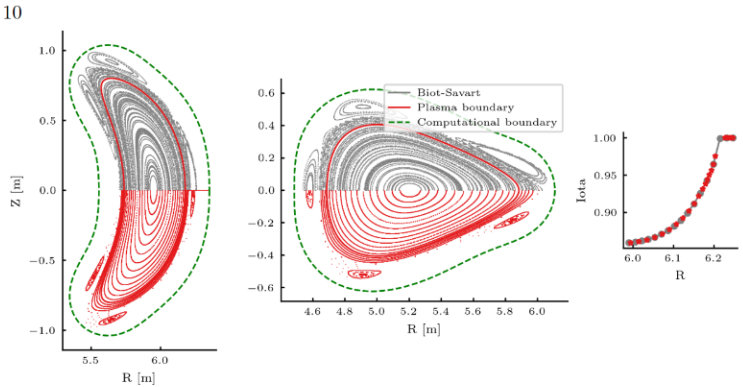
- Parallelization of the H-matrices approach
- Refactoring/optimization of the code to make it competitive with, *e.g.* EIRENE

The SPECTRE code

E. Balkovic, J. Loizu

- Built on top of SPEC to compute 3D magnetic equilibria with general topology
- Uses parts of the SPEC core written in Fortran
- Adds improvement through a Python interface, *e.g.*
  - New force formulation
  - Stable trust-region-based least squares minimization

- Last year we left off with the code being about to be released



SPECTRE: the open source release

- There was a lot of excitement around the release of SPECTRE
- We decided to polish the software to have the maximize user retention
  - Clean the (Python) code
  - Write extensive documentation, tutorials and code examples
    - The entry barrier should be as low as possible
  - Contributing guidelines
  - CI to automate code quality checks, tests, and documentation updates

Working with private companies

- There were discussions already end of 2025 for a potential partnership with a private company
- Took months before SPC/EPFL and the company landed on an agreement
  - EPFL’s Technology Transfer Office not up to date with software

Working with private companies

- There were discussions already end of 2025 for a potential partnership with a private company
- Took months before SPC/EPFL and the company landed on an agreement
  - EPFL's Technology Transfer Office not up to date with software

- Compatibility of the dependency licenses
- Example of FFTW:
  - GPL license version >= 2
  - As such, it "contaminates" any derivative work with GPL license
  - According to Free Software Foundation: a program linked or that links with a GPL library is a derivative work
  - Write a generic interface and use other compatible libraries for distribution
  - Especially important for the next PiPy release

The elephant in the room: AI

- Around one month after the open source release we got the first AI contribution
- We immediately decided to define clear guidelines on AI use
  - Made a survey of different projects to see how they deal with AI

The elephant in the room: AI

- Around one month after the open source release we got the first AI contribution
- We immediately decided to define clear guidelines on AI use
  - Made a survey of different projects to see how they deal with AI

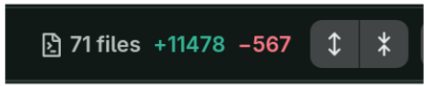
- The (maybe naive) rules:
  - Declare AI use: which AI and for what
  - The contributor is responsible of the changes, not the AI
  - Code must be read, understood, and tested by the contributor
  - We do not accept 100% AI and/or automatic contributions.

The elephant in the room: AI

- Around one month after the open source release we got the first AI contribution
- We immediately decided to define clear guidelines on AI use
  - Made a survey of different projects to see how they deal with AI

- The (maybe naive) rules:
  - Declare AI use: which AI and for what
  - The contributor is responsible of the changes, not the AI
  - Code must be read, understood, and tested by the contributor
  - We do not accept 100% AI and/or automatic contributions.

The next days...

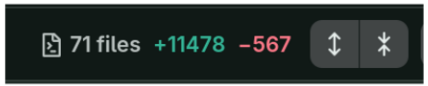


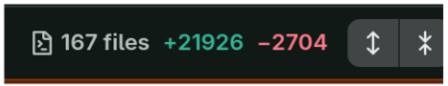
The elephant in the room: AI

- Around one month after the open source release we got the first AI contribution
- We immediately decided to define clear guidelines on AI use
  - Made a survey of different projects to see how they deal with AI

- The (maybe naive) rules:
  - Declare AI use: which AI and for what
  - The contributor is responsible of the changes, not the AI
  - Code must be read, understood, and tested by the contributor
  - We do not accept 100% AI and/or automatic contributions.

The next days...





Conclusions and future work

- SPECTRE is now open source!
- We gathered a small community of regular contributors
  - We need human resources to maintain the code
- How to deal with the AI shift?

- From user feedback, performance is now the clear bottleneck
  - Limiting compute-heavy parts already identified and need optimization