A short description of AMNS

David Coster
2022-04

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.

Design Goals for AMNS

- From the ITM era:
  - AMNS data should be centralized and managed
    - Version control of data imported to the ITM-TF data base is mandatory.
    - The provenance of the data must be accurate and stored in the ITM database
    - For “production” runs with ITM-TF codes using AMNS data it is important that the data have been given a stamp of approval by an expert.

Design Goals for AMNS

- The data should be comprehensive, ubiquitous and easily used
  - This means identifying what data is needed
    - not always easy!
  - The AMNS data must be communicated to ITM-TF codes via a standardised interface
    - this should also ensure coherence between different ITM-TF codes needing the same type of data
  - All AMNS data used by codes should be available through the AMNS data interface
    - no back doors

Design Goals for AMNS access

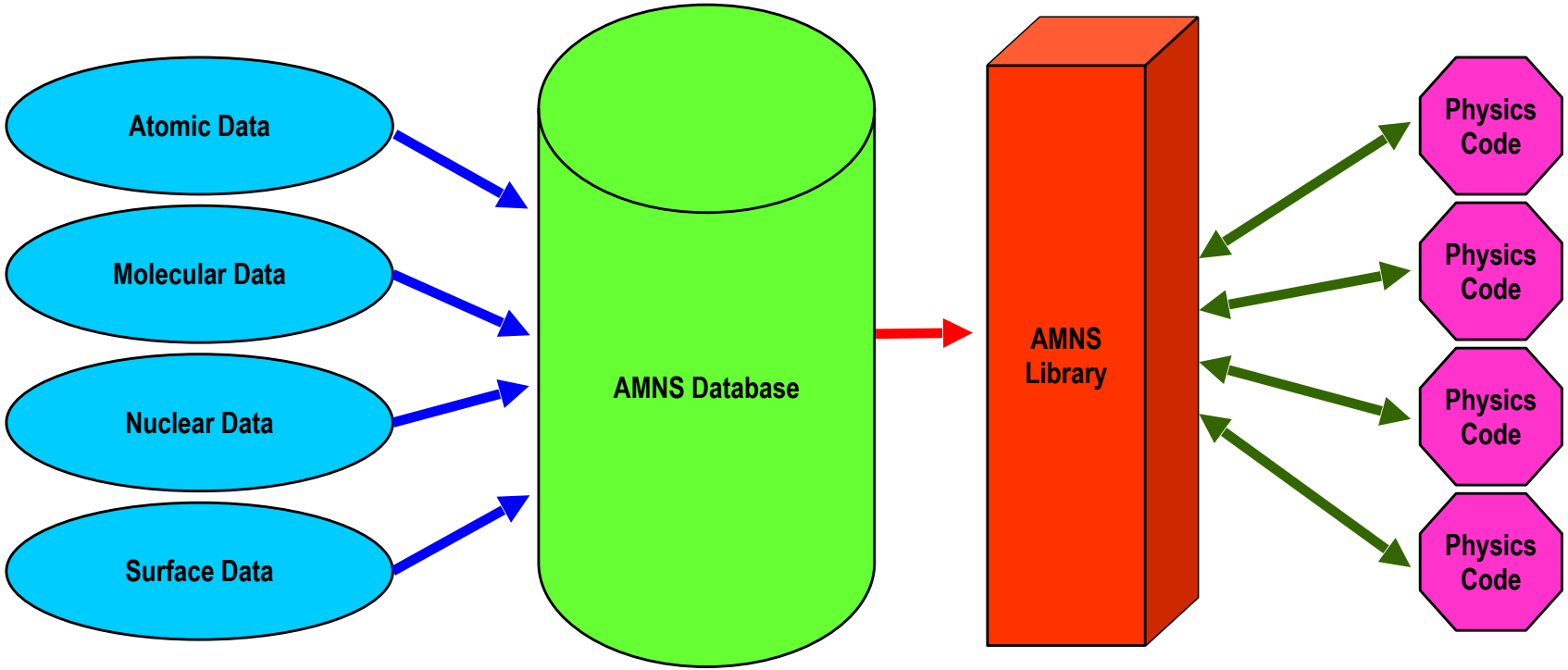
Physics code

- Access to AMNS data only via interface
  - initialization (2)
  - finalization (2)
  - querying parameters (2)
  - setting parameters (2)
  - **getting data** (1)
- Separation between use of the data and the implementation of the data
- Code author doesn't need to become an expert in AMNS
- Ensures compatibility between codes

AMNS implementation

- Only accessed by a set of defined calls
- Implementation by AMNS experts
- Different versions can be supported
- Different implementations possible
  - Analytic formulae
  - Table lookup
- "Old" versions should always be recoverable (even if wrong)
- Should become easier to implement "new" data

Design Goals for AMNS

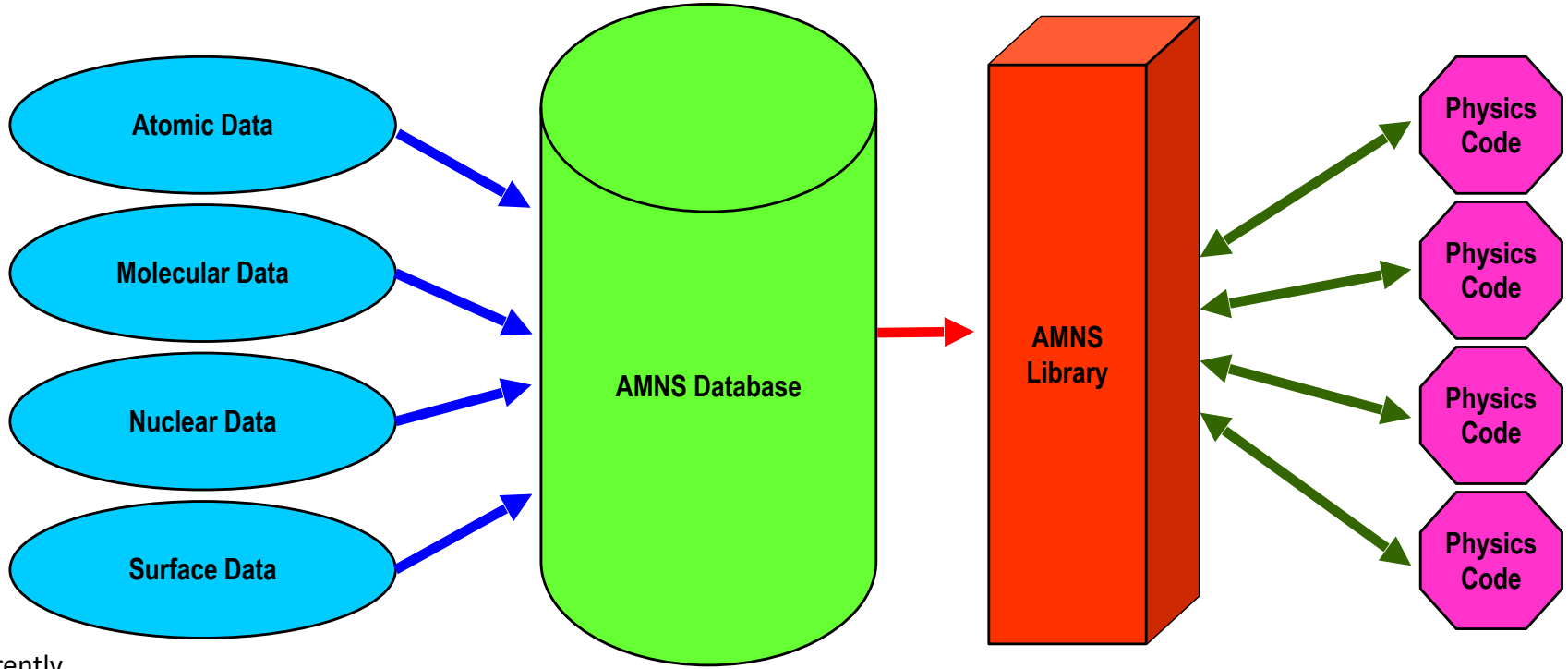


- Data selection
- Data import

- Data storage
- Data transport

- User interface design
- Supporting multiple languages

Design Goals for AMNS



Currently

- ADAS ADF11
- Nuclear cross section and rates

Previous (CPO) version also included

- Surface reflection and sputtering data
- Elastic collisions

- Initially used CPOs
- Now using IDSs

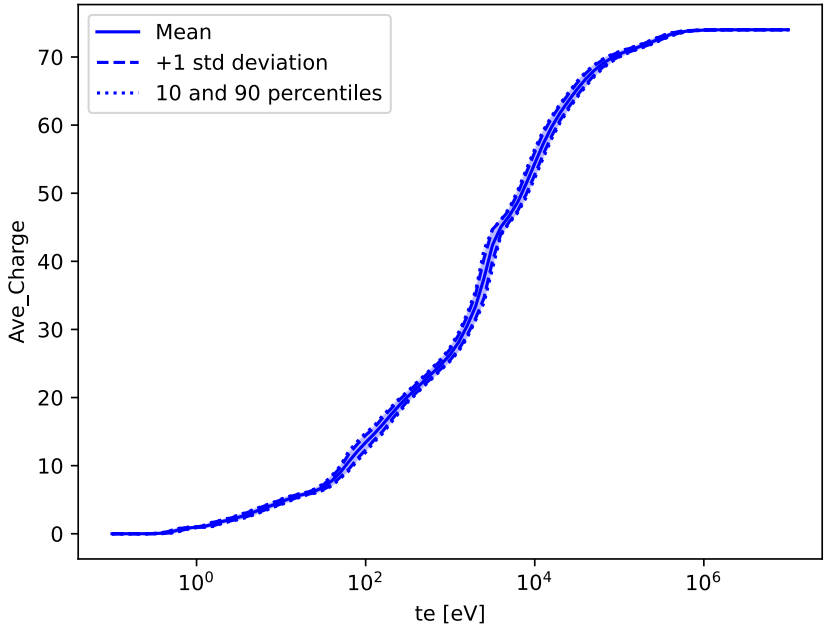
- Library written in Fortran
- Interfaces to Fortran, Python, C, MATLAB, Java

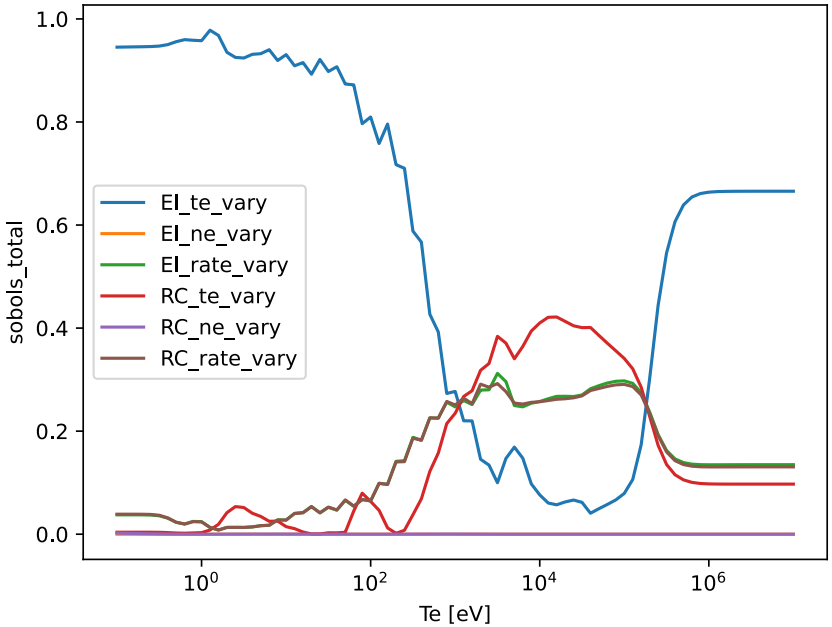
Some areas where more work is needed

- The fact that the IDSs (data dictionary and access layer) are still closed source limits the appeal outside the ITER community
  - Would like to see DD, AL and AMNS open sourced
    - And easier to install
  - Could also imagine an implementation not using IDS's for the backend
- Data ingestion is done by a driver written in Fortran (historical reasons)
  - Might make more sense to have this in Python (good ADAS bindings plus easier access to other formats for data input)
- Data storage is still in locally stored IDS's
  - Having a global IDS infrastructure would be useful
    - UDA?
    - CernVM-FS?
- Should have a push in the community to bring in more data (ADF15, beam stopping, more nuclear data, reintroduction of surface data, molecular data)
  - Then have a push in the code community to use the data

Application to UQ for W rates

- Average charge and total Sobols: EI_te_vary dominant over most of the range except for an important Te range where RC_te_vary takes over …





Usage of N data from AMNS

- Use AMNS nuclear rates to look at the density of various components as a function of temperature at 1000 seconds starting with 5×10^{19} m^{-3} each of D and T