Introduction

- A tokamak plasma is a such complex that none of the single plasma analysis codes can fully describe the evolution.
- The integrated modeling approach is an appropriate way to investigate these complex non-linear phenomena self-consistently, helping us understand the physics behind them.
- There has been vigorous effort to improve the integrated modeling approach, such as TRANSP [1], JINTRAC [2], ISIS [3], ETS [4], and STEP [5].

We introduce a newly developed Python framework coined as TRIASSIC (Tokamak Reactor Integrated Automated Suite for Simulation and Computation) which uses IMAS/IDS [6] as its internal storage for its fully modular approach.

Validation on KSTAR

- For the validation of core modeling in TRIASSIC, 30 stationary time slices from 30 different KSTAR discharges from the 2015 to 2018 campaign were prepared.
- The validation of interpretative simulation was done by comparing the calculated energy ($W_{\text{predicted}}$) with EFIT stored energy ($W_{\text{EFIT}}$).
- The validation of predictive simulation was done by comparing the predicted density ($n_e\text{ (predicted)}$) and energy ($W_{\text{predicted}}$) with experiments.
- The effective charge $Z_{\text{eff}}$ was assumed to be equal to 1.9 (identical with $n_e\text{ (predicted)}$).

Predictive Simulations

- The equilibrium (CHEASE), 1D transport solver (ASTRA), NB/EC (NUBEAM, DIOKAY), neoclassical/analomalous transport (NCLASS, TGLF), and cold neutral (FRANTIC) components were used for predictive simulation.
- A significant understimation of the density level was found when the wall recycling was not considered.
- The effect was considered by assuming a constant influx ($6\times10^{22}\text{m}^{-2}\text{s}^{-1}$) of cold neutrals.
- The $n_e$ and energy was accurately predicted when the puffing was considered, and its average values were 0.99 and 0.97 with standard deviations of 0.14 and 0.12.
- The overestimation of $n_e$ was found for low-prefil discharge > limitation of constant puffing rate modeling.
- The overestimation of energy was found for impurity injection discharge (due to low $Z_{\text{eff}}$ assumption).

Structure of TRIASSIC

- Various models that can consider plasma equilibrium, transport, and H&CD are contained in TRIASSIC.
- TRIASSIC has a unique structure when compared with the pre-developed integrated suite of codes.
- There is no interconnection between the models, and the models directly communicate with IDS through Python interface.
- Every component is being modularized with minimal functionality and limited task.
- The exploitation of existing plasma analysis codes written in Fortran or C/C++ was done by F2PY/SWIG wrapper generator.
- The wrapping was done with an additional driver function or subroutine which properly executes the code.
- The code can be dynamically loaded in TRIASSIC, and the calculation routine can be invoked as if it is a Python function.
- TRIASSIC orchestrates the execution of each component with time advance.
  - data contains IDS shot and run information for save/load, and the simulation data if required.
  - task contains simulation options for each components.
  - sim contains the invocation settings for all the components.

Summary

- The TRIASSIC code, which is the integrated suite of codes written in Python, has been developed for analyses of tokamak plasmas.
- Exploiting the IMAS/IDS generic data structure enabled a fully modular approach without any interconnection between the components.
- TRIASSIC was validated on KSTAR by comparing the interpretatively calculated total plasma energy with the experiment and by comparing the prediction results with the experimental density and energy.

References