ID: 1229

Development of Integrated Suite of Codes and Its Validation on KSTAR

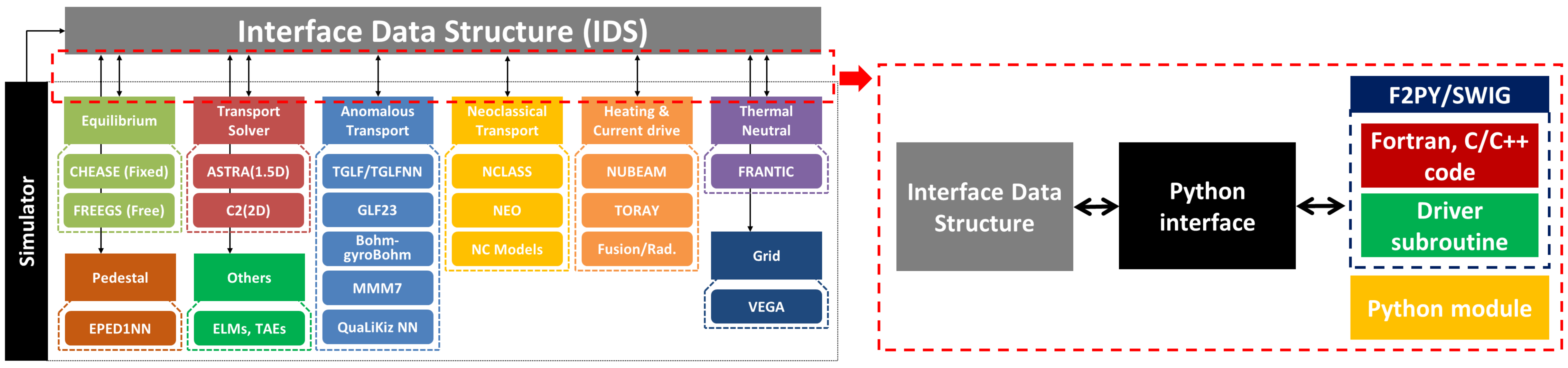
C.Y. Lee[1], J. Seo[1], S.J. Park[1], J.G. Lee[1], S.K. Kim[2], B. Kim[1], C.S. Byun[1], Y.S. Lee[1], J.W. Gwak[1], Yong-Su Na[1*], J.S. Kang[3], L. Jung[3], H.-S. Kim[3], and S.-H. Hong[3]

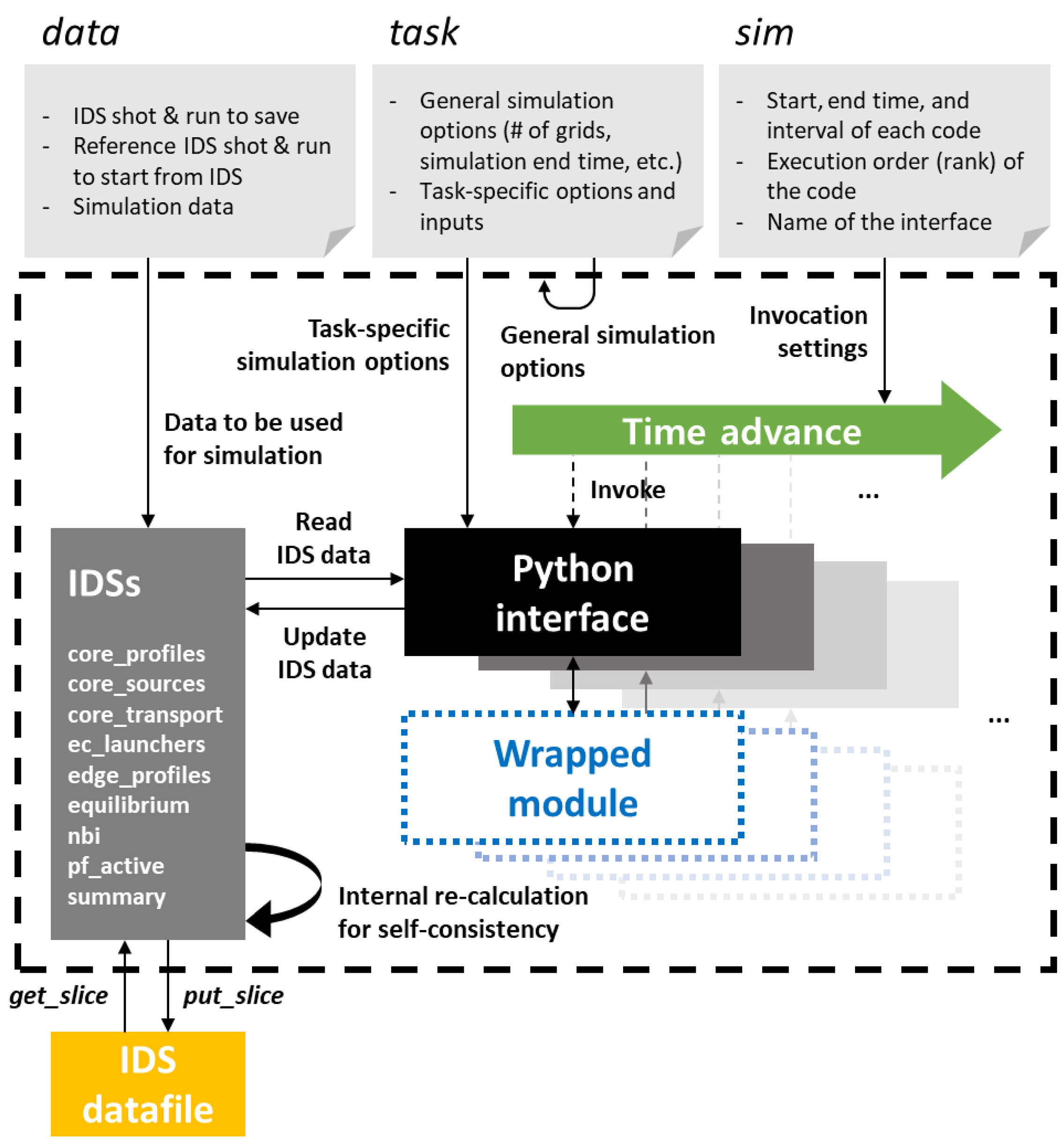
[1]Department of Nuclear Engineering, Seoul National University, Seoul, Korea, Republic Of
[2]Department of Mechanical and Aerospace Engineering, Princeton University and Princeton Plasma Physics Laboratory, Princeton, NJ, U.S.A
[3]Korea Institute of Fusion Energy, Daejeon, Korea, Republic Of
*Corresponding author's e-mail: ysna@snu.ac.kr

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], IPS [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.

Structure of TRIASSIC



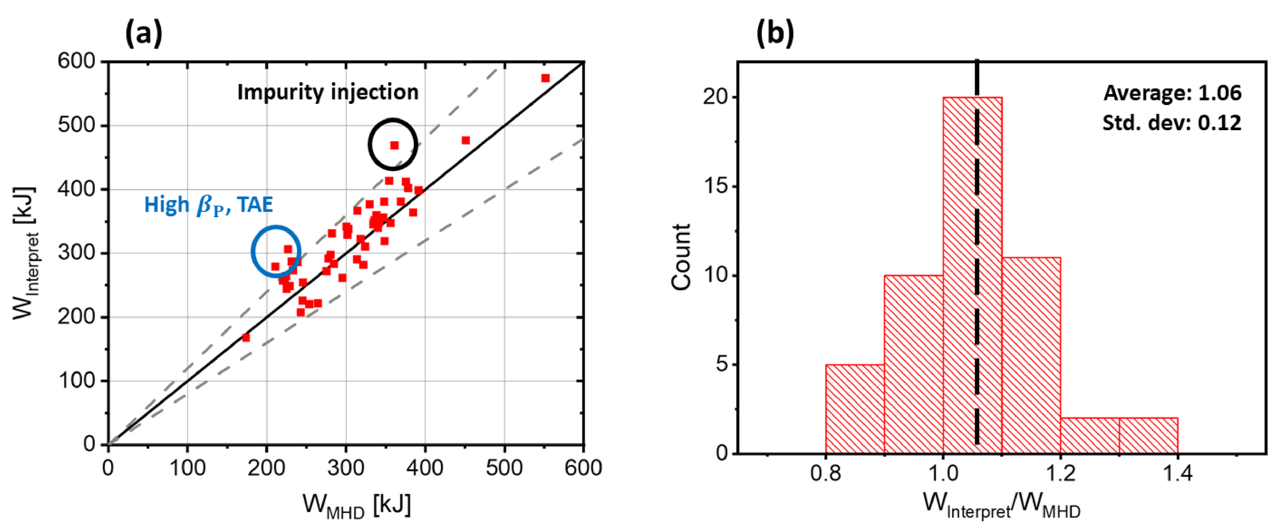


- Various models that can consider plasma equilibrium, transport, and H&CD are contained in TRIASSIC.
- TRIASSIC has a unique structure when compared with the pre-developed integrated suite of codes.
  - There is no interconnection between the models, and the models directly communicate with IDS through 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 module 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.

Validation on KSTAR

- For the validation of core modeling in TRIASSIC, 50 stationary time slices from 30 different KSTAR discharges from the 2015 to 2018 campaign were prepared.
- The validation of interpretive simulation was done by comparing the calculated energy ($W_{Interpret}$) with EFIT stored energy (W_{MHD}).
- The validation of predictive simulation was done by comparing the predicted density ($n_{el,\ Predict}$) and energy ($W_{Predict}$) with experiments.
- The effective charge Z_{eff} was assumed to be equal to 1.9 (identical with $n_C = 0.03 * n_e$).

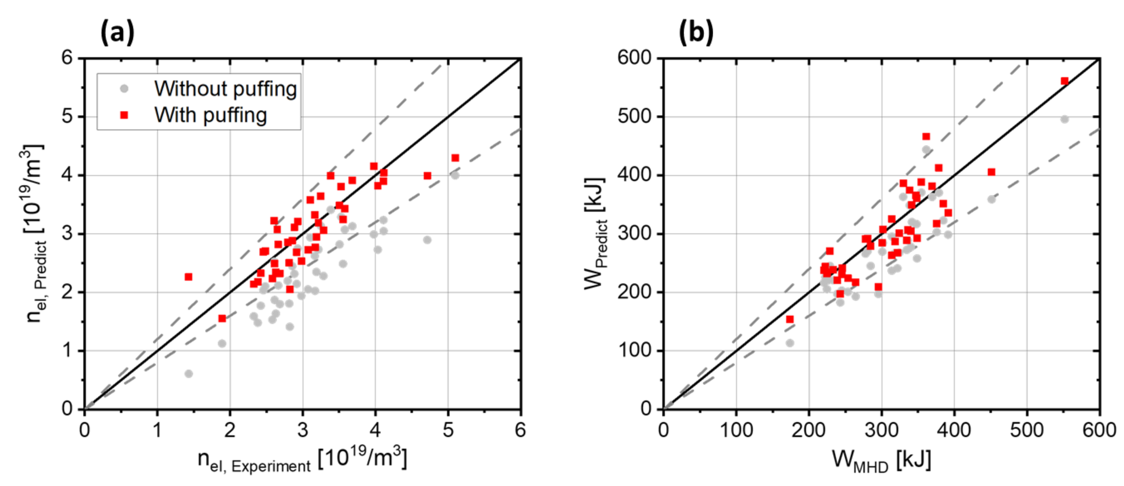
Interpretive Simulations

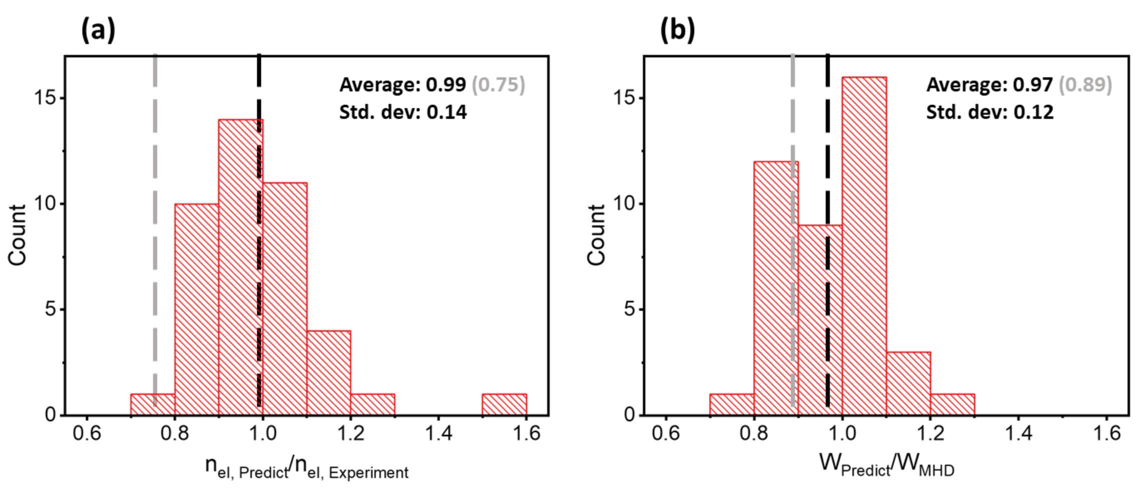


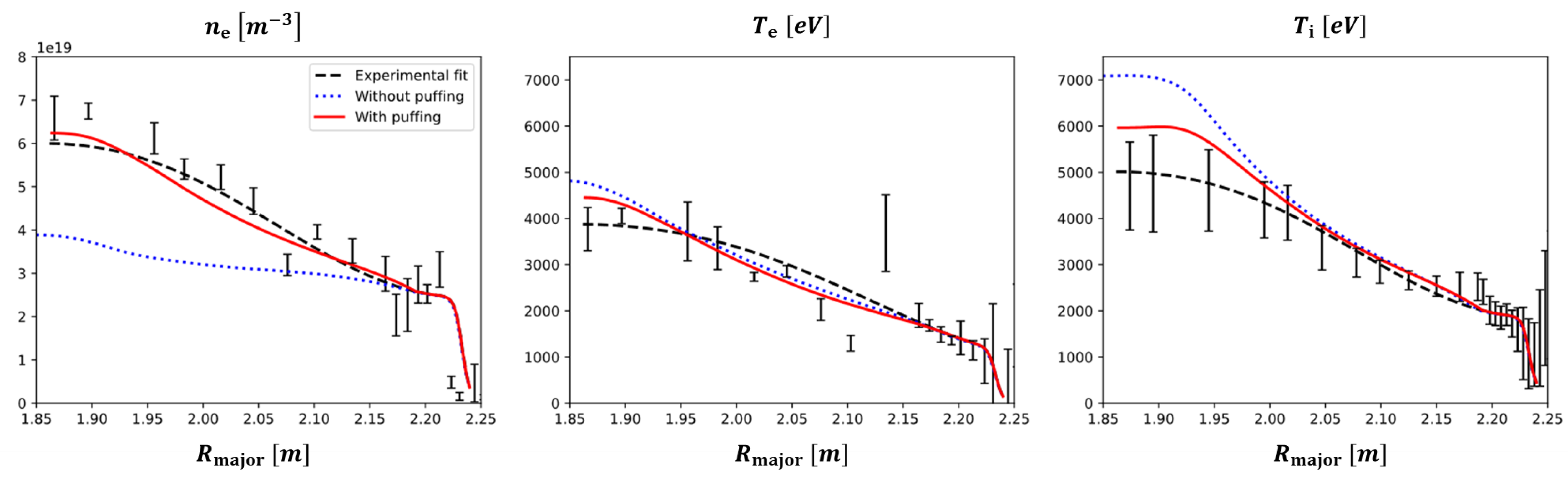
- The equilibrium (CHEASE) and NBI (NUBEAM) components were used for the interpretive simulations.
- $W_{Interpret}$ well lies on the y = x line with W_{MHD} on the x-axis (a), and the value of $W_{Interpret}/W_{MHD}$ does not deviate much from 1.0, as its average is 1.06 and its standard deviation is 0.12 (b).
- Overestimation was found for high β_P discharges (blue circle).
  - Those high β_P discharges showed the TAE activity due to an absence of EC wave heating/current drive.
  - Lack of Alfvén eigenmode driven energetic particle transport.
- And for the Argon impurity injection experiment (black circle).
  - Overestimation of ion population (due to low Z_{eff} assumption) caused by the high-Z impurity injection might be the reason.
- The RMP-induced fast ion loss effect is not observed.
  - Might because the database used in this study does not include the discharge with a strong core field penetration.

Predictive Simulations

- The equilibrium (CHEASE), 1.5D transport solver (ASTRA), NBI/EC (NUBEAM, TORAY), neoclassical/anomalous transport (NCLASS, TGLF), and cold neutral (FRANTIC) components were used for predictive simulation.
- A significant underestimation of the density level was found when the wall recycling was not considered.
  - The effect was considered by assuming a constant influx ($6\times10^{20}\ m^{-2}s^{-1}$) of cold neutrals.
- The n_{el} 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_{el} was found for low-prefill discharge → limitation of constant puffing rate modeling.
- The overestimation of energy was found for impurity injection discharge (due to low Z_{eff} assumption).







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 interpretively calculated total plasma energy with the experiment and by comparing the prediction results with the experimental density and energy.

References

[1] HAWRYLUK, R.J., Physics of Plasmas Close to Thermonuclear Conditions, Vol. 1, (1981) 19–46
[2] ROMANELLI, M. et al., Plasma Fusion Res. 9 (2014) 3403023
[3] ELWASIF, W.R. et al., 18th Euromicro Int. Conf. on Parallel, Distributed and Network-Based Processing, (2010) 419–427
[4] COSTER, D.P. et al., IEEE Trans. Plasma Sci. 38 9 (2010) 2085.
[5] MENEGHINI, O. et al., Phys. Plasmas 23 4 (2016).
[6] IMBEAUX, F. et al., Nucl. Fusion 55 12 (2015).

PLARE

SEOUL NATIONAL UNIVERSITY