Neural-network accelerated coupled core-pedestal simulations with self-consistent transport of impurities and compatible with ITER IMAS

O. Meneghini[1], G. Snoep[2], B.C. Lyons[1], J. McClenaghan[1], C.S. Imai[3], B. Grierson[4], S.P. Smith[1], G.M. Staebler[1], P.B. Snyder[1], J. Candy[1], E. Belli[1], L. Lao[1], J.M. Park[5], J. Citrin[6], T.L. Cordemiglia[7], A. Tema[8], S. Mordijck[9]

E-mail: meneghini@fusion.gat.com
[1] General Atomics, San Diego CA
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
[3] University California San Diego, San Diego CA
[4] Princeton Plasma Physics Laboratory, Princeton NY
[5] Oak Ridge National Laboratory, Oak Ridge TN
[6] Dutch Institute for Fundamental Energy Research, Eindhoven, The Netherlands
[7] Max Planck Institute for Plasma Physics Garching, Germany
[8] Ecole polytechnique federale de Lausanne, Switzerland
[9] College of William and Mary, Williamsburg, Virginia

PACS numbers: 52.65.-y, 52.55.Fa

Abstract. An integrated modeling workflow capable of finding the steady-state plasma solution with self-consistent core transport, pedestal structure, current profile, and plasma equilibrium physics has been developed and tested against a DIII-D discharge. Key features of the achieved core-pedestal coupled workflow are its ability to account for the transport of impurities in the plasma self-consistently, as well as its use of machine learning accelerated models for the pedestal structure and for the turbulent transport physics. Notably, the coupled workflow is implemented within the OMFIT framework, and makes use of the ITER integrated modeling and analysis suite (IMAS) data structure for exchanging data among the physics codes that are involved in the simulations. Such technical advance has been facilitated by the development of a new numerical library named OMAS.

Submitted to: *Nucl. Fusion*

1. Introduction

Self-consistently accounting for the interaction between the models that describe different physics of a thermonuclear plasma is essential to obtain more accurate fusion gain predictions, and rely less on the initial assumptions on which models are based. An integrated modeling workflow capable of finding the steady-state plasma solution with self-consistent core transport, pedestal structure, current profile, and plasma equilibrium physics has been developed, tested, and used to perform predictions for a 15 MA D-T ITER baseline scenario. Key features of the core-pedestal coupled workflow are its ability to self-consistently account for the transport of impurities in the plasma, as well as its use of machine learning accelerated models for the pedestal structure and the turbulent transport [1].

Notably, the coupled workflow is implemented as part of the STEP (stability transport equilibrium pedestal) module within the OMFIT framework [2], which makes use of the ITER integrated modeling and analysis suite (IMAS [3]) data structure for exchanging data among the physics codes that are involved in the simulations. Such technical advance has been facilitated by the development of a new numerical library named OMAS [4].

The self-consistent STEP workflow is discussed in Sec. 2, with the details of the underlying OMAS library reported in Appendix A. Section 3 explains how impurities are self-consistently coupled to the workflow, while Section 4 details how the neural-network accelerated transport models have been extended to include impurity particle transport and operate over a wide parameter space. Finally, Section 5 illustrates the application of the self-consistent workflow on DIII-D and ITER.

2. Stability Transport Equilibrium Pedestal (STEP) workflow

The advent of ITER has been a driving force towards the worldwide adoption of a unified standard for managing tokamak data. Specifically, the ITER Integrated Modeling and Analysis Suite (IMAS) [3] defines an ontology, a storage infrastructure, and an Application Programmer Interface (API) for interacting with ITER data. All ITER simulated and experimental data will be served to the ITER parties through IMAS. Scientific workflows that handle ITER data will therefore eventually need to become compatible with IMAS.

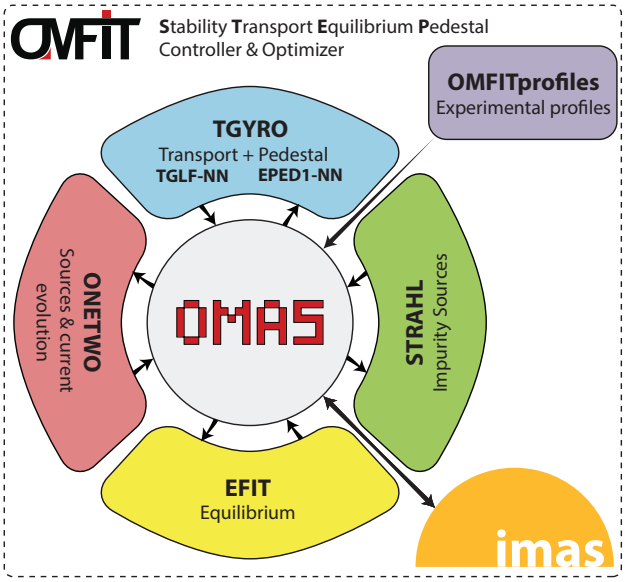

Figure 1. This figure illustrates the data flow for the OMFIT physics modules involved in the workflow presented in this paper. The STEP module in OMFIT leverages OMAS for centralized data communication among its physics modules. Each of these modules accept IDSs as inputs and produce IDSs as outputs. At any moment in the IDSs of the coupled simulation can be written from OMAS to a (possibly remote) IMAS database. The workflow is also initialized via IDSs, which can either be retrieved from an IMAS database (also remotely), or can be generated from other sources within OMFIT (see Appendix A).

The IMAS ontology, known as the ITER Physics Data Model (PDM), consists of over 50 ordered hierarchical Interface Data Structures (IDSs) organized by topical areas across the modeling (equilibrium, kinetic profiles, sources, etc.) and experimental domains (diagnostic, heating system, etc.). Each IDS is designed to store all the relevant information of the plasma or tokamak subsystems with which it is associated. For each quantity within an IDS, the PDM defines its units, coordinates, numerical type, and provides an overall description. The ITER strategy towards coupling physics codes and developing integrated modeling workflows is to rely on standardized IDSs to transfer data from one code component to another. Such perspective has motivated the development of an extensible integrated-modeling module within OMFIT, which leverages IDSs for

centralized data communication among its physics modules.

The new OMFIT physics module, named STEP (Stability Transport Equilibrium Pedestal), defines a series of "steps", each of which reads all necessary input information as IDSs, sets-up and executes a physics code, and then writes the code's output back as IDSs. The centralized nature of the data flow within STEP is schematically illustrated in Fig. 1. STEP can be initialized from data stored within any IMAS database, or any OMFIT module which produces IMAS compatible data, such as the experimental profiles fitting module OMFITprofiles [5]. In addition, the ability to start a STEP simulation from existing TRANSP and ONETWO simulations has been implemented. Finally, since ITER will store experimental data within IMAS using the same data representations as today's simulations, we expect that in the future the same approach could be used to perform interpretative analyses of ITER experimental runs.

From a technical perspective the interaction between the OMFIT framework and IMAS has been made possible by leveraging OMAS [4] (Ordered Multidimensional Array Structure), a numerical library designed to simplify the interface of Python codes with IMAS. At the heart of OMAS is the idea of providing a convenient API which can store data in a format that is compatible with the IMAS data model, but using standard storage systems (such as HDF5 or NetCDF) in addition to the one provided by IMAS itself. OMAS does not address the problem of the mapping of physics codes' I/O to the IMAS data model, which is instead done within the definition of the data classes of the OMFIT framework. This approach was used to adapt all of the physics codes used in this paper to exchange data via IDSs. Further details of the OMAS library and data mappings to/from IDSs can be found in Appendix A.

We note that the centralized data communication approach upon which the STEP module is developed is a significant departure from the original ad-hoc data exchange approach that has characterized OMFIT in the past. The new strategy hides the complexities of translating data from one format to another within the routines of the OMFIT classes, thus greatly simplifying the implementation of the physics workflow within STEP. New physics codes can be easily coupled to STEP, as long as their I/O has been mapped to the ITER PDM format using OMAS. Yet, the greatest advantage of adopting a centralized data structure is that this approach decouples the data flow and the execution workflow, meaning that the individual components can be executed in any order, allowing for the flexible design of workflows that suit different physics applications. These include open-loop self-consistent predictive workflows (as described in the following section), as well as closed-loop predictions, and multivariate constrained optimization and parametric scans.

A copy of the IDSs is retained in memory each time physics codes are executed within STEP, thus providing a history of how the data evolves throughout the workflow. Depending on the application, users can choose to save the IDSs from OMAS to a (possibly remote) IMAS database. The steps chosen may be for some of the steps of the simulation (most often only the final) or the whole history of the workflow. In the latter case, use of IMAS *occurrences* lends itself well for storing the evolution of the IDSs in the same machine and pulse IMAS database entry.

3. Self-consistent core-pedestal simulations with transport of impurities

Impurity transport is an important element in integrated modeling simulations because it influences both core performance, through dilution and radiation, and pedestal stability via its effect on the bootstrap current. Previous coupled core-pedestal simulations were shown to be able to reproduce experimental profiles [1, 6], but relied on prior knowledge of the plasma average charge $Z_{\rm eff}$ across the plasma.

To self-consistently account for the effect of impurities, the tightly-coupled TGYRO [7] and EPED1-NN core-pedestal workflow described in Ref. [1] has been iteratively coupled to the 1D impurity transport code STRAHL [8]. Here EPED1-NN is a neural-network regression trained to reproduce the pedestal structure from the EPED1 model [9]. In this scheme NEO [10] and TGLF [11] (or its neural network accelerated counterpart) provide the transport fluxes that are used to calculate the diffusion D and pinch v profiles that are input to STRAHL, which is mainly responsible for calculation of the core impurity source (ie. how much carbon VI enters the pedestal) and the evaluation of the overall impurity radiated power. Figure 3 illustrates the density profiles of the different ionization stages of carbon as calculated by STRAHL for a typical DIII-D plasma. The resulting core-pedestal profiles (now with consistent impurities) are finally iterated with the calculation of the plasma equilibrium (with the EFIT code [12]) and sources (with the ONETWO code [13]). Figure 2 summarizes the execution workflow, highlighting its three nested self-consistency loops.

As per Ref. [1] the core-pedestal coupling loop is achieved by tightly integrating EPED1-NN within the TGYRO transport solver. According to this scheme, the pedestal solution (that is the boundary condition for the core solution) is updated after each step of

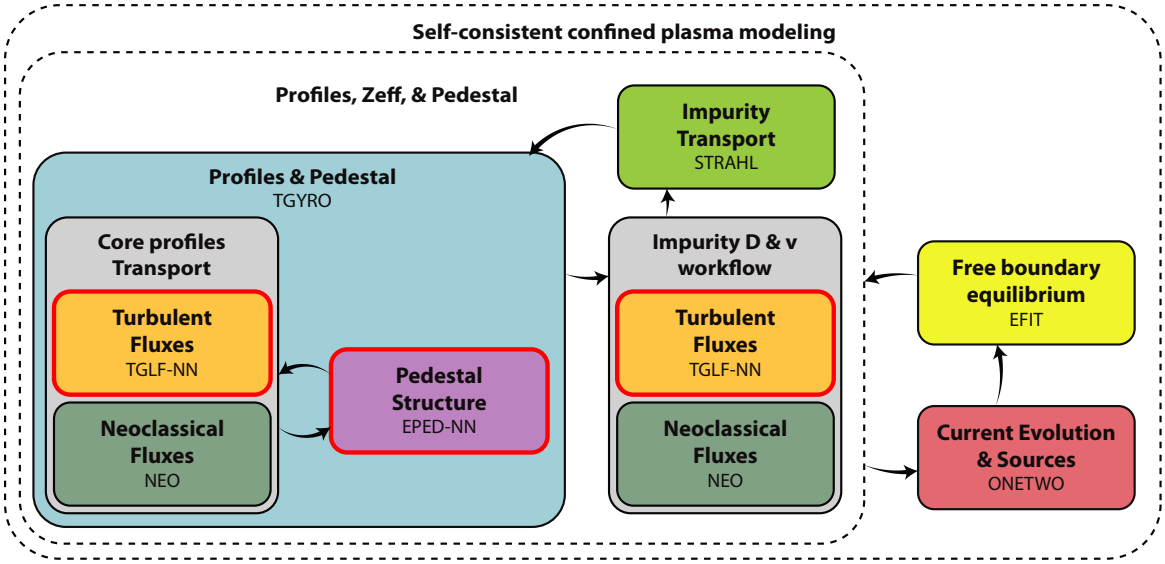

Figure 2. The stationary-state core-pedestal coupled workflow described in [1] has been extended to account for the effect of impurities self-consistently. The second iteration loop couples the core-profiles calculation to be consistent with the impurities model, before the sources and equilibrium are updated. This approach is advantageous under the assumption that sources and equilibrium are less sensitive to the details of the core profiles, but if necessary, the workflow can be reconfigured for arbitrary iteration among the core-profiles, impurities, sources, and equilibrium calculations.

the TGYRO flux matching calculation. As a result, the core-profiles and pedestal structure that is output by TGYRO are now self consistent with one another, before being iterated with the impurities, sources, and equilibrium calculations. The second iteration loop of Fig. 2 allows for all core profiles – including the impurities – to be consistent with one another, before the sources and equilibrium are updated. This approach is advantageous under the assumption that sources and equilibrium are less sensitive to the details of the core profiles. If for any reason this assumption is not satisfied, the STEP workflow allows for easy reconfiguration of the iteration loops among the core-profiles, impurities, sources, and equilibrium calculations.

Because the inward flow of neutrals at the last closed flux surface is not known, we set its value in STRAHL either to maintain the number of impurity particles in the plasma equal to a constant since the plasma is assumed to be in steady-state, or to match a predefined plasma effective charge at one point in the plasma (typically at the top of the pedestal $Z_{\mathrm{eff,ped}}$). While the integrated (or local) impurity density is prescribed, the shape of the impurity density profile from the magnetic axis to the separatrix is predicted based on the first-principles NEO and TGLF transport models. In the future, this information could be provided by a Scrape-Off-Layer (SOL) model, and the core-pedestal-SOL solution iterated to convergence.

The impurity particle transport model in STRAHL is based on the ansatz that the radial particle flux density can be split into a diffusive and a convective part:

$$\Gamma_I = -D\,\frac{\partial n_I}{\partial r} + v\,n_I \tag{1}$$

where Γ_I is the impurity particle flux of each charge state I of a given impurity, n_I is its density, and D and v are the flux surface averaged diffusion and convection (drift velocity) coefficients, respectively.

The NEO and TGLF neoclassical and turbulent transport models do not use this assumption and output directly the transport fluxes. For the purpose of using these particle transport models in STRAHL, we compute the particle diffusion and convection coefficients by linearizing the neoclassical and turbulent fluxes in their density dependence. We thus evaluate the fluxes Γ_1 and Γ_2 for two values of the impurity density scale-length $L_1 = -n_1\,/\,n_1'$ and $L_2 = L_1 + \Delta L$ such that

$$D = \frac{\Gamma_1\,n_2 - \Gamma_2\,n_1}{n_2'\,n_1 - n_1'\,n_2} \tag{2}$$

$$v = \frac{\Gamma_2\,n_1' - \Gamma_1\,n_2'}{n_2'\,n_1 - n_1'\,n_2} \tag{3}$$

where the prime superscript indicates radial derivatives.

Such procedure is valid insofar as D and v are weakly varying over the ΔL, which can be ensured by either setting an appropriately small value for ΔL, or by calculating them for a trace amount of the impurity of interest that is added to the NEO and TGLF simulations. With concentrations several orders of magnitude lower than the intrinsic impurity species,

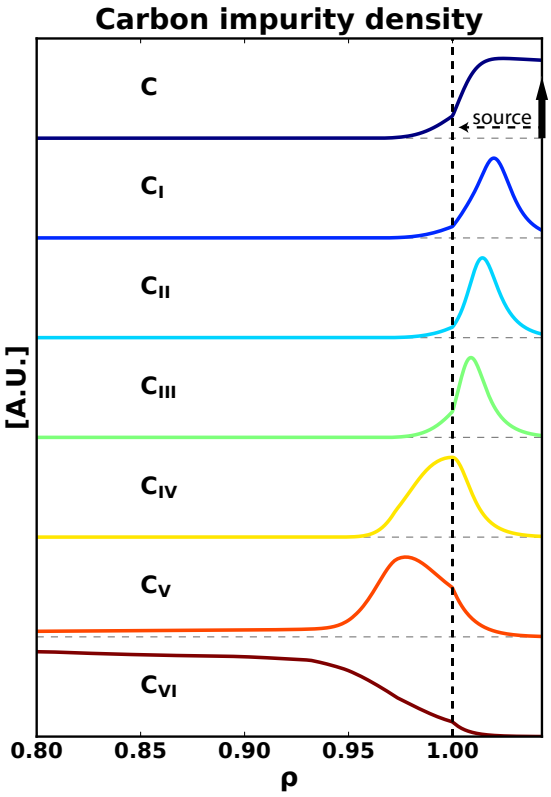

Figure 3. Sample edge density profiles of the different ionization stages of carbon as calculated by STRAHL for a DIII-D plasma. In the simulations a source of neutral carbon atoms is placed outside of the plasma separatrix. The impurity source is scaled to maintain the number of impurity particles in the plasma equal to a constant, or to match a predefined plasma effective charge at one point in the plasma.

variations of the trace impurity scale-length does not affect the transport. Although simpler, the former method is only valid if the fluxes are a continuous function of the impurity scale-length, while the latter method is more general and robust to models (such as TGLF) which may present discontinuities in the output fluxes.

We point out that the evaluation of the NEO and TGLF fluxes done as part of the impurity D and v calculation were done by running TGYRO, which in this case is not used as a transport solver but rather as a tool to conveniently evaluate the transport fluxes from on-axis to the separatrix in parallel.

The TGLF transport model is generally not applicable in the axis and edge regions of the plasma. Within TGYRO this issue is elegantly resolved by integrating the flux-matching scale-lengths from the pedestal boundary condition (provided by EPED1-NN) to the axis, assuming that the profiles' inverse scale-length is piece-wise linear and continuous, and is zero on axis [6, 7]. However, the same approach cannot be applied to STRAHL, which instead requires prescribed diffusion and convection coefficients at every point of its computational grid.

Figure 4 illustrates the D and v terms as calculated by NEO and TGLF, as well as the values that are input into the STRAHL simulation. TGLF is generally applicable in the core region of the plasma ($0.2 < \rho < 0.8$), and the D and v coefficients are simply calculated from the sum of the neoclassical and turbulent flux contributions.

Near the axis ($0. < \rho < 0.2$), the source of carbon is negligible, and as such, at equilibrium the carbon flux should also be zero. Under these conditions, Eq. (1) becomes $v/D = -n'/n = 1/L$. In this region the diffusion coefficient is held fixed at its value at the axis-core interface ($\rho = 0.2$), while the convection term is calculated such that the impurity scale-length goes linearly to zero on axis. This approach is consistent with the treatment of the density profiles in this region by TGYRO.

In the edge region ($\rho > 0.8$) the impurity particle source from the wall can be significant (see Fig. 3) and the impurity profile cannot be evaluated analytically. Furthermore, even if we know that the steep density gradients in this region are associated with large values of the v/D ratio, the solution (specifically the $Z_{\rm eff}$ profile) is sensitive to the alignment of the electron and impurity density pedestal profiles. Although EPED provides the pressure height and width, the precise details of the temperature and density profiles in the pedestal depend sensitively on particle sources and inter-ELM transport mechanisms, both of which are out of the scope of this work. Attempts to model the D and v profiles in this region from first principles resulted in values of $Z_{\rm eff}$ with non-physical large excursions near the edge. Based on this observation, the problem was re-cast to find the edge D and v profiles such that the $Z_{\rm eff}$ is flat in this region. For this purpose we decided to maintain the D profile at its value at the core-edge interface ($\rho = 0.8$), and iteratively modify the v profile to achieve that goal starting from an initial guess that assumes zero sources. This approach results in a robust and physical solution that is consistent with experimental profile reconstructions that are typically inferred from electron and carbon density measurements on DIII-D under stationary conditions.

4. Extension of TGLF-NN and EPED1-NN neural-network accelerated models

Neural-network based models have been shown to accelerate two of the most critical aspects for functional whole device modeling applications: 1) the H-mode plasma pedestal pressure structure, with the EPED1-

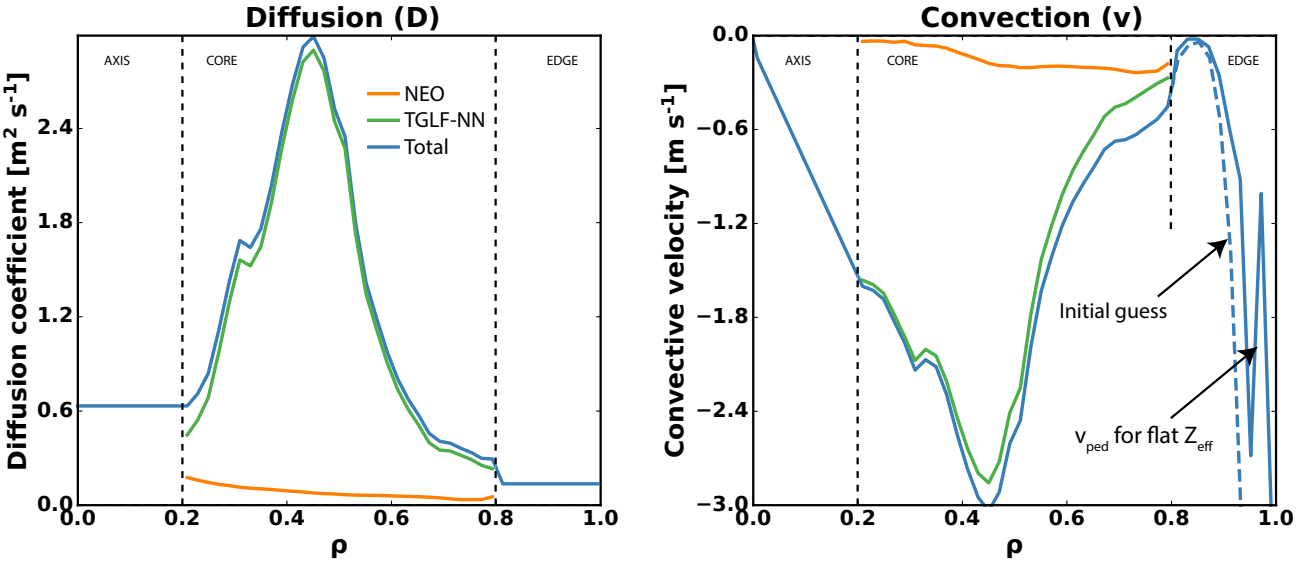

Figure 4. Carbon diffusivity (a) and convection (b) terms as a function of normalized toroidal flux ρ, as calculated by NEO and TGLF. The total diffusivity and convection values (sum of the neoclassical and turbulent transport contribution) are input to STRAHL.

NN model; and 2) the prediction of the turbulent transport fluxes in the core of the plasma, with the TGLF-NN model.

The TGLF-NN model originally described in Ref. [1] was trained on a reduced set of DIII-D discharges, and assumed a deuterium-carbon plasma to predict the electron and ion energy fluxes, the electron particle flux, and total ion momentum flux. In this work we extended TGLF-NN by adding the prediction of the particle fluxes for all ion species to the outputs of the NN (as is required for any impurity transport simulations) and we significantly expanded the training database to cover a much wider DIII-D operation range. In addition, a new database with 50/50 deuterium-tritium mix (main ion), and helium ash and neon (impurity species) was built to model ITER in the nuclear phase.

The TGLF-NN training database was built by gathering data from TGLF simulations run (in parallel) via the TGYRO transport manager. The initial condition of each TGYRO run was based on TRANSP [14] simulation time-slices that were randomly drawn from a database of over 3000 existing DIII-D and ITER TRANSP runs. This approach was chosen because the TRANSP data is readily available, and the database naturally covers a wide range of plasma regimes of physical interest. The same technique can be used to create a TGLF training database for other devices for which TRANSP simulations are available.

To sample the region of the TGLF input parameters space that is in the vicinity of the nominal conditions, the kinetic profiles and the Miller equilibrium coefficients that are input to TGYRO were randomly perturbed from the original values as defined in the TRANSP database. Each TGYRO run was set to sample the plasma profiles between $0.2 < \rho < 0.9$, packing more grid points where the inverse scale-lengths of the kinetic profiles were higher. The resulting TGLF-NN training database have been made publicly available [15].

The implementation of the TGLF-NN model itself has also been revised and rewritten leveraging Google's Tensorflow [16] state-of-the-art machine learning library. As in the previous implementation, the TGLF-NN prediction is the weighted average of multiple NNs outputs (ensemble learning), but now the topology of each of these NNs is built based on the structure learned by the trees in a random forest (deep jointly-informed neural networks DJINN algorithm [17]). The DJINN mapping "warm starts" the neural network training process by initializing the network in a state that performs similarly to the decision tree, and naturally initializes each NN with a diverse topology that improves the ability of the ensemble to generalize [18]. Figure 5 illustrates how the TGLF-NN model includes the ensemble of DJINN in a single Tensorflow graph. The study presented in this paper uses 10 DJINN networks, of 2 hidden layers each.

To give more relative importance to points in the database that are near the critical gradient, the definition of the error used for the back-propagation training of the TGLF-NN model was non-linearly transformed by an hyperbolic inverse sine function. Specifically, the cost c is defined to be $c = 1 - \sum_s R_s^2$,

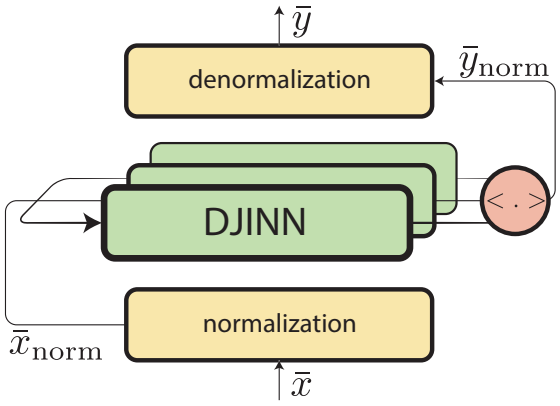

Figure 5. High level view of the TGLF-NN model. The normalization and denormalization operation, as well as the averaging of the output of the DJINN ensemble are all contained within a single Tensorflow graph.

where R_s^2 is the *coefficient of determination* (in other words the cost is the residual sum of squares over the total sum of squares) of each of the model outputs, marked with the subscript s:

$$c = 1 - \sum_s R_s^2 = \sum_s \frac{\sum_i (y'_{is} - t'_{is})^2}{\sum_i (t'_{is} - \bar{t}'_s)^2} \tag{4}$$

where y is the model prediction, t is the training data, $\bar{t}$ is average value of the training data, and i is the index over the samples in the database. The prime superscript here represents the following non-linear transformation:

$$y'_s = \text{arcsinh}\left(\frac{y_s}{\sigma_s}\,\alpha\right) \tag{5}$$

which assigns higher weighing to small values of the fluxes (ie. closer to the critical gradient) for higher values of the parameter α. Expressing the error in terms of R^2 is advantageous in that the coefficient of determination is bounded between 0 and 1, thus giving the same relative importance in the optimization process to each of the non-linearly transformed model outputs. Once trained, TGLF-NN can be used in a transport solver (such as TGYRO).

The EPED1-NN model, a neural-network regression to a database of EPED1 simulations, described in [1] has also been updated to provide the pedestal height and width for the H-mode, metastable, and super H-mode solutions of three different diamagnetic stabilization models of the peeling ballooning instabilities in EPED1.

A full treatment of diamagnetic stabilization of ideal MHD with short to intermediate wavelength instabilities requires a detailed kinetic or two-fluid formalism. Instead, simple rules can be used to estimate the impact of diamagnetic stabilization on the pedestal instabilities. In the EPED1 model, the calculated ideal MHD growth rate γ can be compared to half the ion diamagnetic frequency $\omega_\star/2$ [19]:

$$\gamma/(\omega^\star/2) > 1 \quad , \quad \text{(H rule)}$$

however this rule is known to be too strong at high densities. Under these conditions the growth rate is best compared to 0.03 ω_A [20]:

$$\gamma/\omega_A > 0.03 \quad , \quad \text{(G rule)}$$

where ω_A is the Alfvén frequency, the logic of this being that typical flow rates in the edge are around .01 to .02 ω_A and the growth rate has to exceed this value to be significant. A hybrid approach that attempts to take both rules into account is $GH > 0.03$:

$$\gamma^2/(\omega_A\omega^\star/2) > 0.03 \quad , \quad \text{(GH rule)}$$

The updated EPED1-NN has been re-implemented in Tensorflow and makes use of DJINN ensembles.

5. Core-pedestal prediction with self-consistent transport of impurities

As an example, we report the application of the core-pedestal workflow with self-consistent transport of impurities to DIII-D H-mode discharge #168830 at 3500 ms. This is a purely NBI heated discharge, and is well representative of high performance steady-state conditions for intrinsic impurity transport studies on DIII-D. In these simulations, the total carbon impurity content is an input to the simulations. The `equilibrium`, `wall`, `core profiles`, and `core sources` IDSs for this simulation were generated starting from an existing TRANSP interpretative simulation.

Figure 6 illustrates the solution of the STEP workflow when convergence is achieved for each of the three nested loops of Fig. 2. The experimentally measured profiles are given by the dashed black lines, with gray uncertainty bands. The other profiles are predicted iterating between TGYRO w/ EPED1-NN (Blue), [TGYRO w/ EPED1-NN] + STRAHL (orange), and [[TGYRO w/ EPED1-NN] + STRAHL] + EFIT + ONETWO (green). Each simulation was run until local changes in the kinetic profiles at the end of the TGYRO iteration were less than 5%. Some discrepancies between simulations and measurements can be observed. Most notably, the EPED1-NN model is over-predicting the experimental pedestal pressure. In turn, through the stiff core transport, this results in an over-prediction of the temperatures in the outer half of the core plasma region. The core particle transport (TGLF + NEO) is under predicted, which results in a higher electron density on axis. The flattening of the experimental profiles in the core region

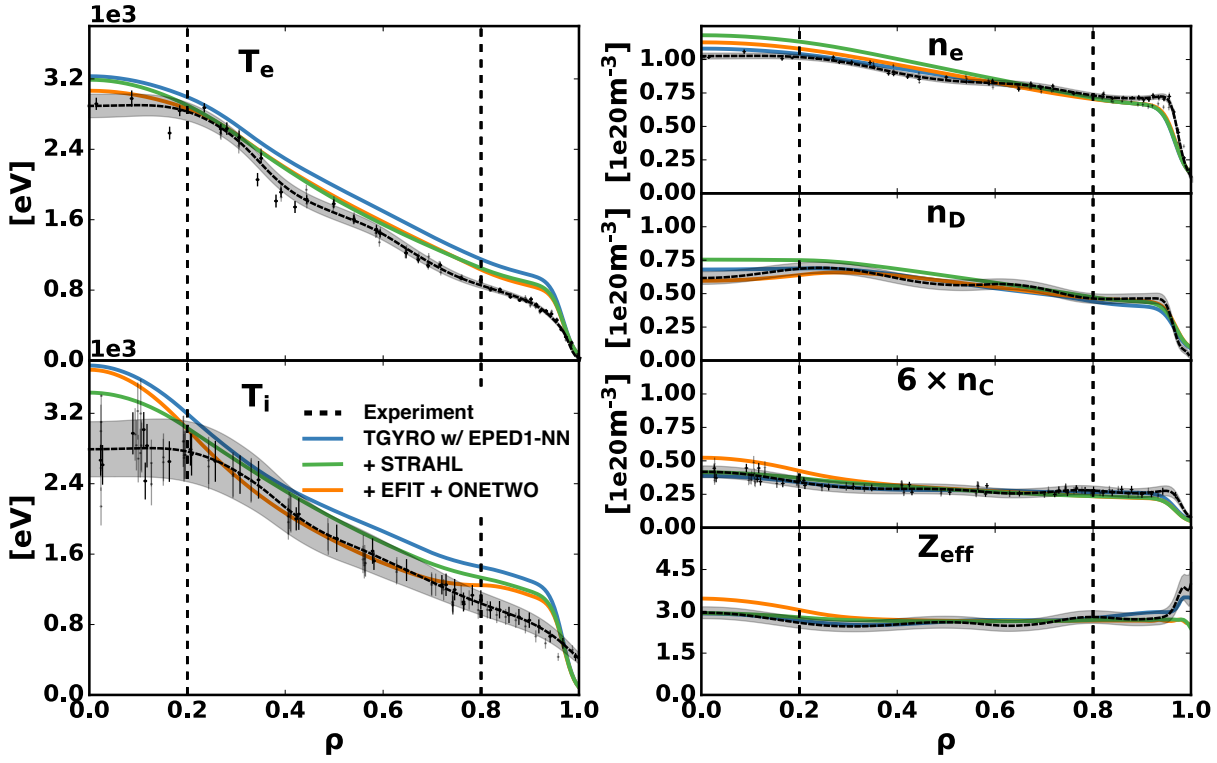

Figure 6. Solution of the STEP workflow for prediction of DIII-D shot #168830 at 3500 ms as convergence is achieved at each of the three nested loops of Fig. 2: iterating between [TGYRO w/ EPED1-NN] (Blue), [[TGYRO w/ EPED1-NN] + STRAHL] (orange), and [[[TGYRO w/ EPED1-NN] + STRAHL] + EFIT + ONETWO] (green).

(especially visible in the ion temperature profile) are associated with the plasma saw-teething, which is not part of our model. Importantly, as we enable different parts of the workflow, and make the simulation more self-consistent, the overall solution does not change (significantly), and all three cases well reproduce the carbon impurity density that is measured in the experiments.

Figure 7 illustrates the results of a sensitivity study with respect to the impurity content in the plasma, which is varied to be 0.5 and 1.5 times the nominal experimental value. In these simulations the deuterium density (main ion) was updated to retain quasi-neutrality, keeping the initial condition of the electron density profile for all cases. Interestingly, at higher $Z_{\rm eff}$ a peaking of the carbon density is observed, accompanied by flattening of the electron density profile. With increasing impurity content the line and the bremsstrahlung radiation increase proportionally to the impurity content. The pedestal pressure is seen to decrease with lower impurity, as expected from the dependency of the edge bootstrap current on $Z_{\rm eff,ped}$.

A similar sensitivity study was carried out to predict the stationary profiles of an ITER baseline scenario discharge. The input parameters to the workflow were taken from the ITER baseline case studied in Ref. [21] (see Fig. 6 of Ref. [21]), and only the pedestal and core profiles (including the impurities) were iterated to consistency, while keeping the magnetic equilibrium and sources constant. Also, to allow use of the TGLF-NN model described in Sec. 4, the plasma composition was modified to be a 50/50 deuterium-tritium mix, helium ash, and neon as an impurity species. Also in this case, the `equilibrium`, `wall`, `core profiles`, and `core sources` IDSs for this simulation were generated starting from an existing TRANSP simulation.

Four variations of the same STEP simulations were run, each with an effective pedestal ion charge $Z_{\rm eff,ped}$ of 1.5, 2.0, 2.5, 3.0 as boundary condition. We emphasize that $Z_{\rm eff}$ is only prescribed at one location in the pedestal region, while the $Z_{\rm eff}$ profile itself is self-consistently calculated from the STEP impurity sources and transport calculation. All simulations have a pedestal electron density of $n_{\rm e,ped}[10^{20}/m^3] = 0.8$. The kinetic profiles of the $Z_{\rm eff,ped}$ scan are shown in Fig. 8 with the relevant scalar quantities plotted in Fig. 9.

Although the dilution of the D-T ions in the core raises the core plasma temperature as $Z_{\rm eff}$ is increased (an effect also observed in Ref. [22] for reactor simulations) this positive effect is offset by a decrease in

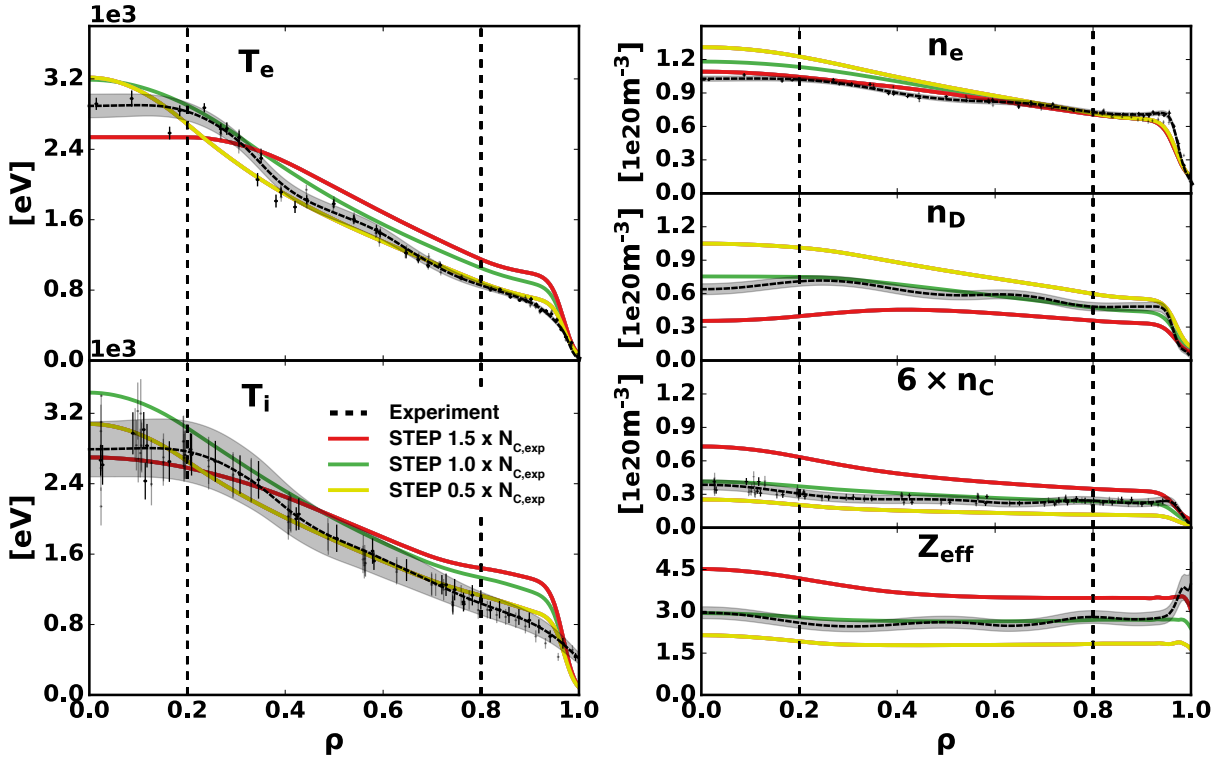

Figure 7. Solution of the STEP workflow for prediction of DIII-D shot #168830 at 3500 ms for varying levels of total carbon impurity content in the plasma (from 0.5 (yellow) to 1.5 (red) times the nominal experimental value (green)).

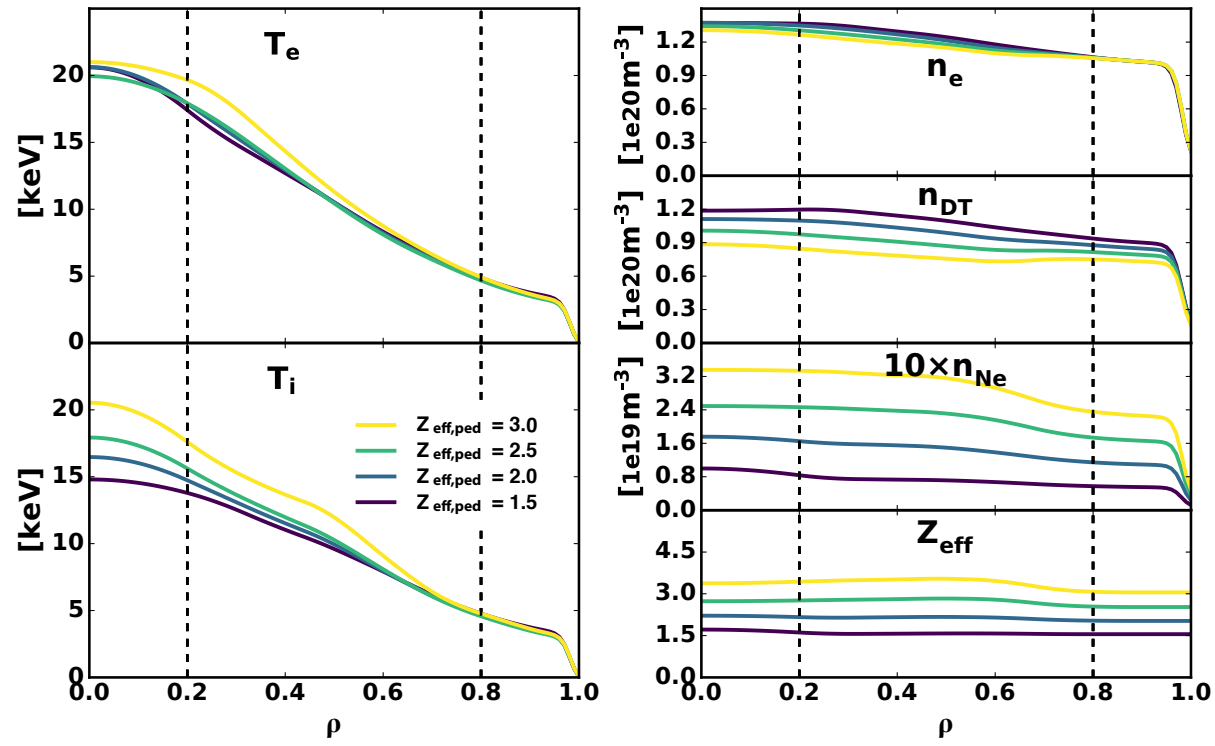

Figure 8. Profiles of the simulated ITER baseline scenarios with effective pedestal ion charge $Z_{\mathrm{eff,ped}}$ ranging from 1.5 to 3.0

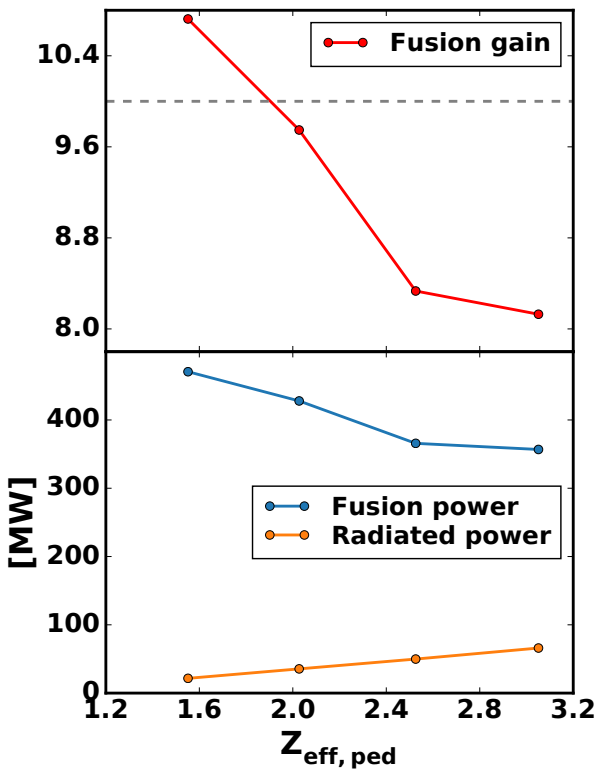

Figure 9. Dependency of the fusion gain, and fusion and radiated powers for the modeled ITER baseline scenario as function of the pedestal $Z_{\mathrm{eff,ped}}$

the available D-T fuel, a decrease in pedestal pressure (of up to 15%), and increased radiation (of 22, 35, 50, and 65 MW for the four simulations). The resulting fusion power for the four simulations were 470, 428, 465, and 356 MW (which for 44 MW of input power correspond to a fusion gain of 10.7, 9.7, 8.3, and 8.1 respectively).

6. Summary

An integrated modeling workflow capable of finding the steady-state solution with self-consistent core transport, pedestal structure, current profile, and plasma equilibrium physics has been developed and tested against a DIII-D discharge, and used to perform predictions for a 15 MA D-T ITER baseline scenario. Key features of the core-pedestal coupled workflow are its ability to account for the transport of impurities in the plasma self-consistently, as well as its use of machine-learning-accelerated models for the pedestal structure and for the turbulent transport physics. Notably, the coupled workflow is implemented within the OMFIT framework, and makes use of the ITER integrated modeling and analysis suite (IMAS) data structure for exchanging data among the physics codes that are involved in the simulations. Such technical advance has been facilitated by the development of a new numerical library named OMAS.

7. Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Awards DE-SC0017992 (AToM), DE-FG02-95ER54309 (GA Theory), DE-FC02-06ER54873 (ESL), and DE-FC02-04ER54698 (DIII-D). This research used resources of the National Energy Research Scientific Computing Center (NERSC), a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

8. Disclaimer

Appendix A. Interfacing the OMFIT framework with the ITER IMAS data infrastructure via OMAS

Appendix A.1. Interfacing physics codes to IMAS

OMFIT [23] is a framework that has been widely used within the fusion community for performing experimental analyses and integrated simulations, across a wide variety of physics domains [2]. A distinguishing feature of OMFIT is its ability to perform integrated modeling simulations using the native files of the physics codes used in its workflows. This is accomplished by being able to read and write a wide range of scientific data formats, some specific

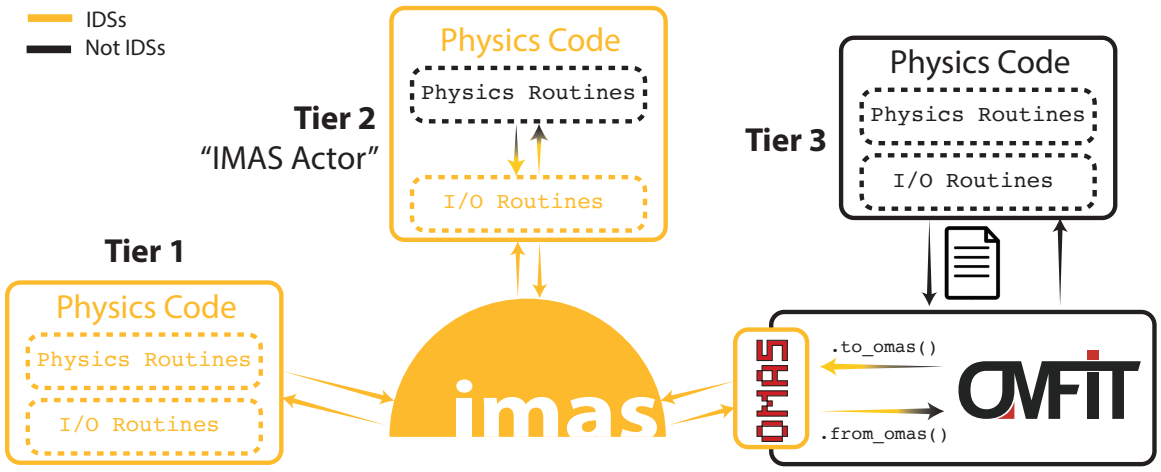

Figure A1. The level of integration of physics codes into IMAS can be categorized into three tiers. Tier 1 are physics codes that natively use IMAS for their internal calculations; Tier 2 physics codes only use IMAS as part of their I/O routines; Tier 3 make use of data mappers that are external to the physics codes themselves to interface with IMAS. OMAS enables OMFIT integration of physics codes with IMAS, independently of their tier. All physics codes used in this paper were adapted to exchange data via IDSs via tier 3 integration.

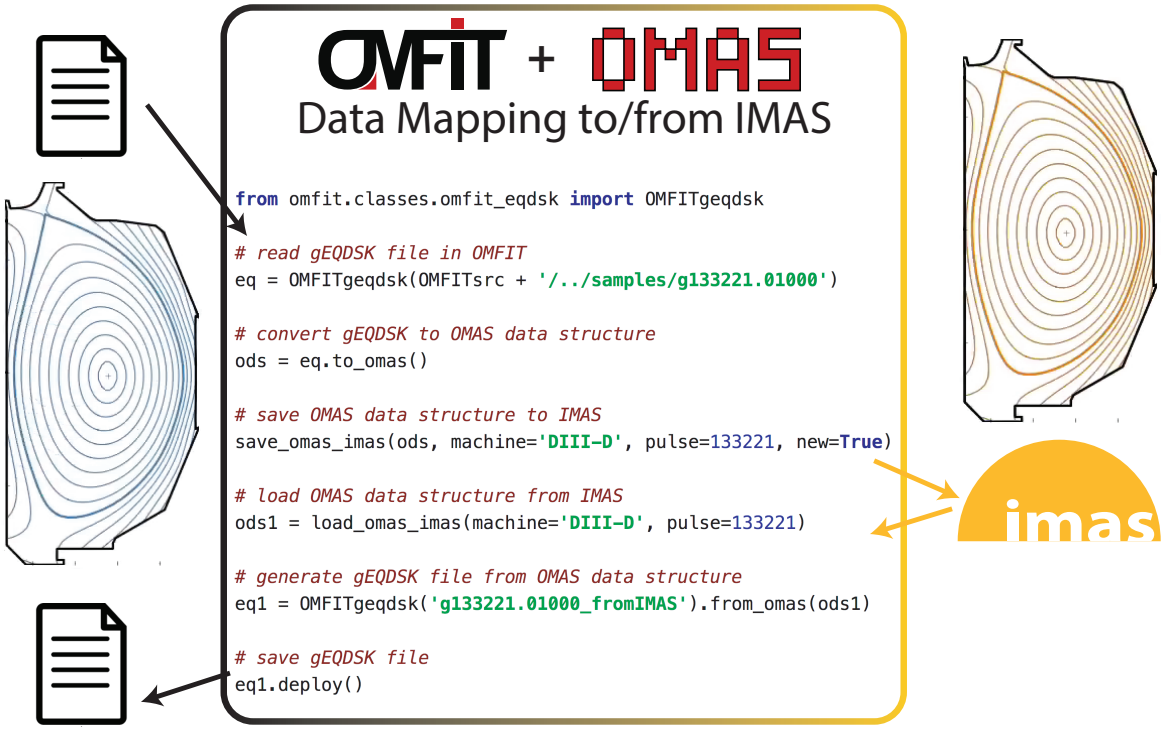

Figure A2. An example showcasing how OMFIT classes make use of OMAS to interface with data stored in IMAS. Here: (1) an EFIT equilibrium file (so called `g-file`) is parsed in OMFIT with the `OMFITgeqdsk` class; (2) the equilibrium object is converted to an OMAS data structure (ODS); (3) the ODS is then saved to IMAS; (4) a new ODS is loaded from IMAS, and (5) a new EFIT equilibrium object is generated from it; finally (6) the new EFIT equilibrium object is saved to file. We note that in this example, any code that manipulates `g-files` can make use of this infrastructure to read/write data from/to IMAS.

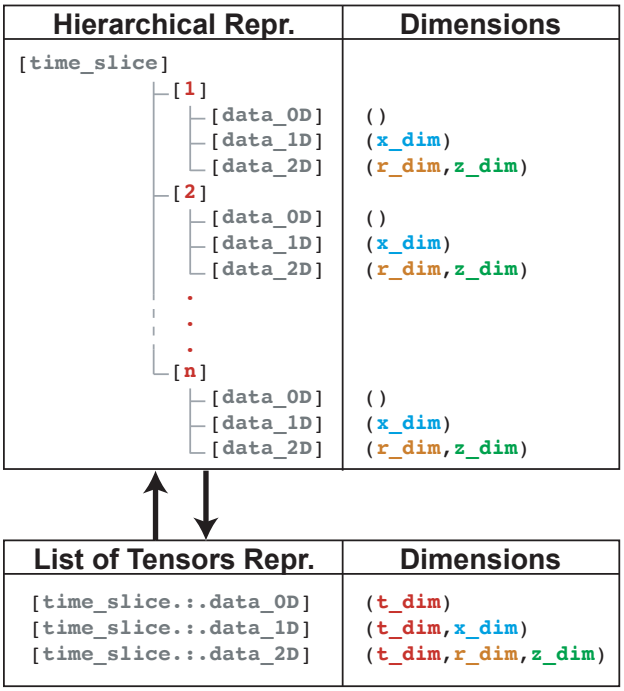

Figure A3. So long as the grid sizes are homogeneous across arrays of structures, the data that is contained in the hierarchical structure can be represented as a series of tensors of higher rank. In this example, a hypothetical IDS contains a series of time-slices each having a scalar, as a one dimensional array of size `x_dim`, and a two dimensional array with shape `r_dim`, `z_dim`. The same data of this IDS can be represented a list of multidimensional arrays, each element being a tensor extended in a time dimension with size `t_dim` with respect to the original IDS. This approach is generic and extensible to nested hierarchical lists of structures.

to the fusion community. Internally OMFIT organizes the data in a free-form hierarchical data structure (named the *OMFIT tree*) which provides a consistent way to access and to manipulate such a collection of heterogeneous objects, independent of their type and origin. At the expense of having to handle more heterogeneous data, this approach has the notable advantage of enabling integrated simulations while leaving physics codes untouched.

For all intents and purposes IMAS data within OMFIT is yet another data format. As such, the adoption of IMAS within OMFIT does not break the original paradigm of the framework, and working with IMAS data can be done in coexistence with the existing legacy data formats. The ability to manipulate IMAS data within OMFIT is enabled by the development of a new numerical library named OMAS [4].

Appendix A.2. Ordered Multidimensional Arrays Structure

The Ordered Multidimensional Arrays Structure (OMAS) library allows Python software to take advantage of the IMAS ontology without requiring the IMAS library to be installed. Such functionality is achieved by organizing the data in memory as hierarchical sets of OMAS Data Storage (ODS) objects, whose structure exactly reproduces that of the ITER physics data model (PDM). OMAS is aware of the ITER PDM, and it is by enforcing strict adherence to this ontology that it guarantees that at any moment the data can be automatically converted to IMAS and back (in which case both OMAS and IMAS need to be installed on the same system). In addition to IMAS, OMAS supports saving/loading data in other formats, such as Python pickle, NetCDF, HDF5, and JSON files, as well as the document-oriented NoSQL database MongoDB. Support for data storage solutions other than IMAS allows users to store data that is in compliance with the ITER PDM ontology without requiring IMAS to be installed. At any moment the data stored in these different formats can be moved to where IMAS is installed, read by OMAS, and stored there in IMAS. It is using this philosophy that OMFIT provides seamless remote data access to IMAS data.

OMAS adopts an API which is familiar to Python developers. Specifically, ODS objects extend the most fundamental Python classes (dictionaries and lists) with a rich set of functionality that allows them to perform automatic coordinate convention (COCOS [24]) transformations, grid interpolation, and units conversions for the data that is either input or requested from the data structure. In addition, ODSs have a growing set of functions that allow calculating sets of physical quantities that can be derived from more fundamental ones. Finally, OMAS is open-source, lightweight, and Python 3 compatible. These qualities make the library easy to maintain, expand, and install on any computer system, whether it is an institutional supercomputer or a personal laptop.

The whole advantage of defining a standard data ontology is to ease code coupling. However, it was a choice of the IMAS conceptual design not to define which entries in the data dictionary are mandatory and which are optional. Similarly, the standardization of the computational grids on which data is defined is also not pertinent to the data schema. On the one hand such design choice provides developers with great flexibility as to what fields of IDSs their codes will output, but on the other hand it burdens them with making provisions for input data that other codes may not have made available, or input data that is stored with different grids. In other words, how an IDS that is input to a code is populated depends on what physics codes were executed prior to it. As an example, let us consider an equilibrium solver that needs the pressure information to run. Depending on the how the input IDS was prepared, such data may

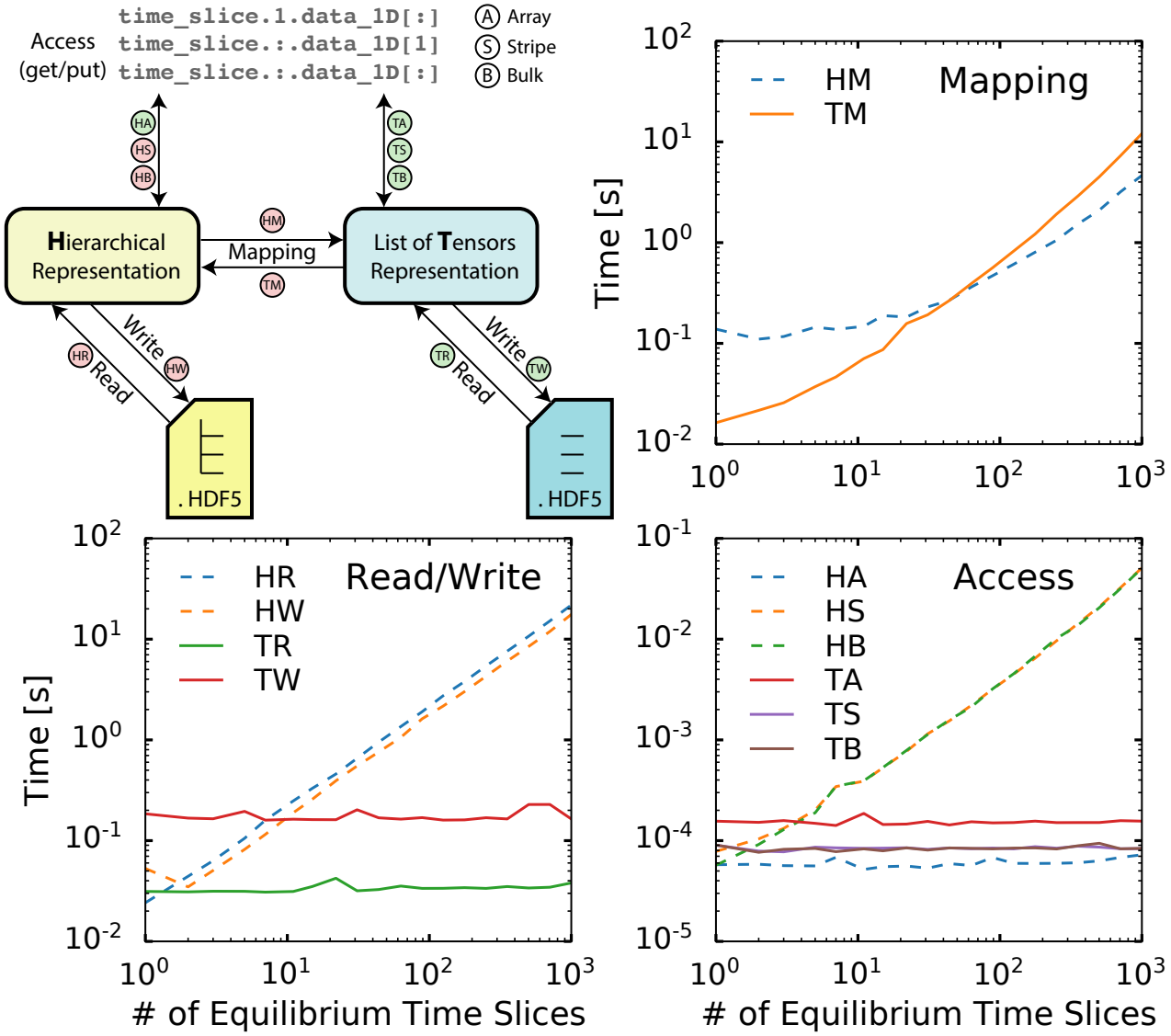

Figure A4. Performance study performed within OMAS for handling a sample equilibrium IDS with an increasing number of time-slices in either hierarchical or tensor representation format. The extent of this study is summarized in the upper-left diagram, and covers mapping between representations, read/write to HDF5 file, and access to the data. Most operations stemming from the hierarchical representation of the data scale linearly with the number of time-slices in the sample IDS (red markers in the diagram), whereas operations that make only use of the tensor representation show little to no dependency on the dataset size (green markers in the diagram).

be present as an equilibrium pressure constraint, or by summing the core profiles electron and ion pressure data (again including the fast ion contribution), or by summing the product of the densities and temperatures for the electron and all ion species (again including the fast ion contribution). As another example, let us consiter a physics code that accepts the equilibrium IDS as input. Internally such code would have to handle each of the possible grid definitions on which the equilibrium information could have been defined‡.

These complexities become ever more apparent when different elements of the integrated-modeling workflow are developed by different scientists, each asserting their own best judgment where the IMAS flexibility allows. One possibility is to let individual developers handle such complexities whenever an IDS is input to their software. Yet, an alternative approach is to rely on a library (such as OMAS) that provides the ability to automatically interpolate data on different grids, and self-consistently calculate derived quantities.

‡ There are 33 possible equilibrium grid definitions as per `equilibrium.time_slice[:].profiles_2d[:].grid_type.index` as well as the option to specify the geometry via the General Grid Description (GGD) [25] convention.

The whole advantage of defining a standard data ontology, as in the ITER PDM, is to ease code couplings, especially for codes that are not written in the same language. We classify the level at which physics codes are integrated within IMAS in three tiers, as graphically summarized in Fig. A1. Codes that natively use IDSs internally as well as for their I/O, belong to the first tier. These are physics codes that have been designed from the very beginning to work with the IMAS API, and as such depend on IMAS to run. For example, workflows that are used to carry out physics simulations in the Kepler [26] workflow manager are tier one. The second integration tier is reserved for codes that can use IMAS for their I/O. Most legacy physics codes that have been adapted to work with Kepler belong to this category. Typically such codes continue to support their original I/O functionality in addition to IMAS. We note that IDS compatible physics codes can also be executed from other programming languages besides Kepler, Python being one of them. We thus borrow from the nomenclature used in Kepler (which refers to these codes as *Kepler actors*) to broadly refer to IDS compatible physics codes as *IMAS actors*. The third tier is used to indicate physics codes that interface to IMAS by means of mapping routines that translate the content of their legacy file formats to IMAS. In this case the internal routines of the physics codes are not altered. Whether the adaptation of legacy codes to IMAS is done by acting on the codes' original I/O files or their I/O routines is a technical choice. Either way, the fundamental requirement of the ITER PDM that all data is stored and exchanged through the hierarchical data structure of IDSs is preserved.

OMAS enables integration of physics codes with IMAS, independently of their tier. For example, OMAS can be used to setup the data necessary for the execution of tier one and two codes in IMAS, and gather and postprocess their data. This approach has been used to seamlessly execute within the OMFIT framework existing IMAS Python actors and Kepler workflows. Such development allows physicists to simultaneously leverage the convenience of the OMFIT environment with the large set of Python IMAS actors and Kepler workflows that have being developed by EUROfusion and the ITER organization. For example, OMFIT and OMAS are being used to facilitate the execution of the European Transport Solver (ETS) [27]. In this scheme, OMFIT provides ETS with a user interface that allows (remotely) writing the starting experimental data (of JET, MAST, DIII-D, KSTAR, and other devices) into IMAS (via OMAS), to execute the ETS Kepler workflow, and to retrieve the simulation results from IMAS (via OMAS) for postprocessing [28].

Concerning the integration with tier three codes, the OMAS library itself does not address the problem of mapping the physics codes I/O to the IMAS data model. Such mappings must be defined in third party Python codes and frameworks, as done for example with the data classes of the OMFIT framework. To this end, the OMFIT data classes are being instrumented with two methods `.to_omas()` and `.from_omas()` that translate data from the native OMFIT format to IDSs and back, as for example illustrated in Fig. A2. An important benefit of instrumenting the OMFIT classes is that by making the format conversion at the files level there is no additional burden on the developers of the original physics codes to adapt their I/O scheme to work with IMAS. Furthermore, existing physics codes that generate the same file formats can share the same translators. The OMFIT mappings and OMAS compatibility with IMAS are tested whenever IDSs are stored/retrieved from a (possibly remote) IMAS database.

Appendix A.3. Handling large datasets in IMAS

The IMAS data structure is based on a hierarchical organization that limits the efficiency with which large datasets can be manipulated. A commonly used approach in computer science for high-performance numerical calculations is to use multidimensional arrays (ie. tensors) where possible. To ease this limitation, we propose use of a transformation that casts the data that is contained in the IMAS hierarchical structure as a list of tensors, by taking advantage of the homogeneity of grid sizes that is commonly found across arrays of structures. Such transformation is illustrated in Fig. A3 for a hypothetical IDS.

The ability to transform data from one representation to the other was implemented in OMAS, as well as the ability to seamlessly use either representation for storing data both in memory and on file. We note that the OMAS implementation of the tensor representation is generic and can handle nested hierarchical lists of structures. Also, we can automatically determine which data can be collected across the hierarchical structure, which cannot, and seamlessly handle both at the same time. Benchmarks show that storing data in this form can be several orders of magnitude faster than done previously, even for datasets of modest size, as illustrated in Fig. A4.

The favorable scaling that is observed when representing IMAS data as a list of tensors makes a strong case for adopting this convention. Implementing the same system as part of the IMAS backend storage of data and in memory representation would likely greatly benefit IMAS performance in many applications. Being able to directly access the IMAS

data as tensors would simplify integration with a broad range of tools and numerical libraries that are commonly used across many fields of science.

We point out that the results that we present for the equilibrium IDS are universal (scaling with the number of elements in an array of structure), but they can be more or less serious depending on the size of such arrays, and whether these are nested within each other. In fact, each level in the hierarchy raises the power of the scaling, which can pose a challenge considering that the some IDSs have up to six nested arrays of structures. Many of the deepest trees are associated with modeling IDS, especially when the General Grid Description (GGD) [25] hierarchical structure is used. Acccess to the experimental IDSs tends to be more efficient, as they tend to have shallower arrays of structures with few elements (in these IDSs the time dependent data is stored as tensors, and the arrays of structures are used to organized different data channels).

Finally, we consider IDS *occurrences*, which allow different versions of an IDS to exist within a given IMAS database entry. These occurrences can be used for all sorts of different applications (e.g. store the same physical quantities calculated with different methods, or at different stages in a iterative workflow). However, they operate as an additional level in the IMAS hierarchy, and their use can thus further exacerbate the inefficiencies described before. One particular application that is not suited for relying on occurrences is to store multiple realizations of signals from a distribution function of uncertain quantities. This sort of application often requires tens, if not hundreds, or even thousands of realizations for a given signal. Besides the fact that at present there is a preset maximum number of occurrences of a given IDS (e.g. only 3 for the equilibrium IDS), the computational cost of accessing a large number of occurrences would be impractical. Not only would the access time once again scale linearly with the number of occurrences, but also the time required for accessing each occurrence is orders of magnitude worse than the one required to access individual structures in an array of structures. By contrast, the simple addition of a dimension to the tensor representation could enable efficient storage of a large number of realizations, and thus open the door to uncertainty quantification workflows and Bayesian integrated data analyses within IMAS.

2. References

[1] O. Meneghini, S. Smith et al., Self-consistent core-pedestal transport simulations with neural network accelerated models, Nucl. Fusion, **57**, 0866034 (2017)

[2] O. Meneghini, S. Smith et al., Integrated modeling applications for tokamak experiments with OMFIT, Nucl. Fusion, **55**, 083008 (2015)

[3] F. Imbeaux, S. Pinches et al., Design and first applications of the ITER integrated modelling & analysis suite, Nucl. Fusion, **55**, 123006 (2015)

[4] OMAS, *Ordered Multidimensional Array Structures, http://gafusion.github.io/omas*

[5] N. Logan, B. A. Grierson et al., OMFIT tokamak profile data fitting and physics analysis, Fusion Sci. Tech., **74**, 125 (2018)

[6] O. Meneghini, P. B. Snyder et al., Integrated fusion simulation with self-consistent core-pedestal coupling, Phys. Plasmas, **23**, 042507 (2016)

[7] J. Candy, C. Holland et al., Tokamak profile prediction using direct gyrokinetic and neoclassical simulation, Phys. Plasmas, **16**, 060704 (2009)

[8] R. Dux and A. Peeters, Neoclassical impurity transport in the core of an ignited tokamak plasma, Nuclear fusion, **40**, 1721 (2000)

[9] P. Snyder, R. Groebner et al., Development and validation of a predictive model for the pedestal height), Phys. Plasmas, **16**, 056118 (2009)

[10] E. Belli and J. Candy, Kinetic Calculation of Neoclassical Transport Including Self-Consistent Electron and Impurity Dynamics, Plasma Phys. Control. Fusion, **50**, 095010 (2008)

[11] G. Staebler, J. Kinsey et al., A theory-based transport model with comprehensive physics, Phys. Plasmas, **14**, 055909 (2007)

[12] L. Lao, J. Ferron et al., Equilibrium analysis of current profiles in tokamaks, Nucl. Fusion, **30**, 1035 (1990)

[13] H. St John, T. Taylor et al., "Transport simulation of negative magnetic shear discharges", in *Presented at the 15th International Conference on Plasma Physics and Controlled Nuclear Fusion Research, Seville, Spain, 26 Sep.-1 Oct. 1994*, volume 1 (1994)

[14] R. Hawryluk, "An Empirical Approach to Tokamak Transport", in *Physics of Plasmas Close to Thermonuclear Conditions, Volume 1*, volume 1, 19 (1981)

[15] O. Meneghini, "TGLF-NN training dataset for deuterium-carbon species", (2019), URL `https://doi.org/10.5281/zenodo.3274234`

[16] M. Abadi, A. Agarwal et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", (2015), URL `http://tensorflow.org/`, software available from tensorflow.org

[17] K. D. Humbird, J. L. Peterson et al., Deep neural network initialization with decision trees, IEEE transactions on neural networks and learning systems (2018)

[18] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning", in *Advances in neural information processing systems*, 231–238 (1995)

[19] P. Snyder, H. Wilson et al., Edge localized modes and the pedestal: A model based on coupled peeling-ballooning modes, Phys. Plasmas, **9**, 2037 (2002)

[20] P. Snyder, R. Groebner et al., A first-principles predictive model of the pedestal height and width: development, testing and ITER optimization with the EPED model, Nucl. Fusion, **51**, 103016 (2011)

[21] R. Budny, Comparisons of predicted plasma performance in ITER H-mode plasmas with various mixes of external heating, Nucl. Fusion, **49**, 085008 (2009)

[22] N. Shi, V. S. Chan et al., Study of impurity effects on CFETR steady-state scenario by self-consistent integrated modeling, Nucl. Fusion, **57**, 126046 (2017)

[23] OMFIT, *One Modeling Framework for integrated Tasks, https://omfit.io*

[24] O. Sauter and S. Y. Medvedev, Tokamak coordinate conventions: COCOS, Computer Physics Communications, **184**, 293 (2013)

[25] D. Penko, "Visualisation of Fusion Data-Structures for Scrape-Off Layer Plasma Simulations", Technical Report

ITER_D_VL75XE, ITER (2017)
[26] Kepler, *Kepler,* `https://kepler-project.org/`
[27] D. P. Coster, V. Basiuk et al., The european transport solver, IEEE Transactions on Plasma Science, **38**, 2085 (2010)
[28] O. Meneghini, S. Smith et al., "Supporting IMAS actors and the European Transport Solver in OMFIT", in *Bulletin of the American Physical Society*, volume 64, American Physical Society (2019)