Real-time-capable prediction of temperature and density profiles in a tokamak using RAPTOR and a first-principle-based transport model

F. Felici[1*], J. Citrin[2], A. A. Teplukhina[3], J. Redondo[4], C. Bourdelle[4], F. Imbeaux[4], O. Sauter[3], JET contributors[5], and the EUROfusion MST1 team[6]

[1]Eindhoven University of Technology Department of Mechanical Engineering Control Systems Technology Group P.O. Box 513, 5600MB, Eindhoven, The Netherlands.
[2]DIFFER - Dutch Institute for Fundamental Energy Research, Eindhoven, The Netherlands
[3]SPC/EPFL, Lausanne, Switzerland
[4]CEA, IRFM, F-13108 Saint Paul Lez Durance, France
[5]See the author list of X. Litaudon et al 2017 *Nucl. Fusion* **57** 102001
[6]See the author list of H. Meyer et al 2017 *Nucl. Fusion* **57** 102014
* Present address: SPC/EPFL, Lausanne, Switzerland.

E-mail: *federico.felici@epfl.ch

Abstract. The RAPTOR code is a control-oriented core plasma profile simulator with various applications in control design and verification, discharge optimization and real-time plasma simulation. To date, RAPTOR was capable of simulating the evolution of poloidal flux and electron temperature using empirical transport models, and required the user to input assumptions on the other profiles and plasma parameters. We present an extension of the code to simulate the temperature evolution of both ions and electrons, as well as the particle density transport. A proof-of-principle neural-network emulation of the quasilinear gyrokinetic QuaLiKiz transport model is coupled to RAPTOR for the calculation of first-principle-based heat and particle turbulent transport. These extended capabilities are demonstrated in a simulation of a JET discharge. The multi-channel simulation requires ~ 0.2 second to simulate 1 second of a JET plasma, corresponding to ~ 20 energy confinement times, while predicting experimental profiles within the limits of the transport model. The transport model requires no external inputs except for the boundary condition at the top of the H-mode pedestal. This marks the first time that simultaneous, accurate predictions of T_e, T_i and n_e have been obtained using a first-principle-based transport code that can run in faster-than-real-time for present-day tokamaks.

19 April 2018

1. Introduction

For future, larger tokamaks, as well as for reactor applications, it will be increasingly important to thoroughly prepare and optimize discharges before they are executed. This requires simulation tools that can accurately and rapidly simulate the time-evolution of the plasma. In addition, testing, preparation and verification of plasma control systems benefit from tokamak simulation models that can execute in real-time [1]. For such purposes, it is beneficial to have a fast, *control-oriented* simulation code, capable of evolving the quantities of interest as rapidly as possible.

One control-oriented simulator for the core plasma transport evolution is the RAPTOR code. Presently, the code has two main applications. Used off-line, it acts as a fast transport simulator capable of simulating an entire plasma discharge in a few seconds on a single core of a standard PC. This makes it highly suited for discharge analysis and preparation, as well as automated optimization of plasma discharges [2], [3], [4], or for design of feedback controllers [5],

[6]. Other codes aimed at fast tokamak simulation for this application include the METIS code, part of the CRONOS suite [7] and the control-design-oriented transport simulator developed at Lehigh University [8].

Another important application of a control-oriented simulator, which is so far unique to RAPTOR, is its use as part of a tokamak plasma control system. In this application, it is used in conjunction with a state observer algorithm, specifically an Extended Kalman Filter [9], to merge diagnostic measurements with model-based prediction in real-time. Additionally, real-time predictions of the expected plasma evolution, also calculated in the control system, can be used for real-time monitoring of the plasma performance. In this function, RAPTOR has been implemented in the control systems of TCV [10], ASDEX-Upgrade [11] and RFX [12].

The version of RAPTOR used so far was capable of simulating the coupled evolution of poloidal magnetic flux (current diffusion) and electron temperature T_e, requiring user-defined assumptions on the other profiles. In this paper we present the extension of the code to allow simultaneous simulations of T_e, T_i, n_e and other particle density species, and demonstrate for the first time the capability to predict core kinetic profiles in real-time.

When predicting the evolution of temperature and particle density, the result strongly depends on the thermal conductivity and particle diffusion and pinch terms. Direct numerical simulation of turbulent fluxes within a nonlinear gyrokinetic framework (e.g. GENE [13] or GYRO [14]) shows increasingly routine agreement between predictions and experiments. However, the computational cost – $\sim 10^4$ CPUh for fluxes at a single radius for a single time point, at ion-scales only – is prohibitive for routine profile prediction applications.

The quasilinear approximation for transport prediction provides significant computational speed-up, while still reproducing nonlinear simulations over a wide range of parameter space. Examples are QuaLiKiz (see [15, 16], and references therein), and TGLF [17], [18]. These transport models are presently used routinely for scenario interpretation and prediction in integrated modelling. However, the computation cost is then still a few seconds for fluxes at a single radius, corresponding to $\sim$50CPUh per second of simulated plasma evolution for JET. While 6 orders of magnitude faster than nonlinear simulations, it is still too slow for real-time computation.

Therefore, earlier versions of RAPTOR used empirical models for the thermal conductivity to predict the temperature profile evolution. By empirically tuning the various parameters of the transport models for a given operational regime, experimental results could be adequately matched, for example simulating entire discharges for ASDEX-Upgrade and TCV [4]. For applications in real-time control design for a particular experiment, with a known plasma scenario, this was sufficient. However, if we wish to predict the core profiles in a wide range of the operational space, for several devices and for future tokamaks without having to re-tune such parameters, transport models with better predictive capabilities are required, while still being sufficiently fast to be used in control-oriented applications.

One way to achieve this is to use neural networks to emulate the results of first-principle-based turbulent transport predictions. Model output databases are compiled with inputs spanning a portion of the experimentally relevant parameter space. The neural network then learns the mapping from code inputs to outputs. The speed of the quasilinear models allows for the production of sufficiently large training sets for this application. A first demonstration of this approach was shown in [19], where QuaLiKiz was emulated within a restricted 4D input space, where each input was varied over an experimentally relevant range (corresponding to $\sim$ 10^5 points in total). While useful as a demonstration, the version of the transport model used in [19] assumed adiabatic electrons and returned only electron heat transport coefficients. In the present paper, we have extended this proof-of-principle neural network transport model to include kinetic electrons. This updated transport model – named QLKNN-4Dkin – now predicts both electron and ion heat fluxes, as well as electron particle diffusion and pinch terms. Impurity transport is not included in this proof-of-principle model. More recently, a neural network was used to emulate TGLF outputs, using higher input dimensionality (23D), but restricted to input parameter combinations encountered in a set of 24 discharges from the

DIII-D tokamak [20].

Another important bottleneck in integrated transport simulation is the self-consistent calculation of the various sources of particles, power and auxiliary current, which often take a significant fraction of the simulation time. When performing rapid simulations (for optimization or control purposes, as is often the case or RAPTOR), results from earlier runs of physics-based models can be directly used, or parametrized. Alternatively, real-time-capable source codes can be used, such as real-time TORBEAM for Electron Cyclotron Heating/Current drive [21, 22], and the recently developed RABBIT code [23] for Neutral Beam Injection. When it is not possible to make real-time capable physics models, a neural network emulation can also be carried out. In the work presented here, source profiles are directly taken from previous runs of more sophisticated codes.

In this paper, we demonstrate the use of QLKNN-4Dkin in the extended RAPTOR-code to predict T_e, T_i, q and n_e simultaneously for a JET discharge. We obtain core profiles that agree with results from the CRONOS transport simulation suite [7] using the same transport model, confirming that the equations are solved correctly in RAPTOR. These profiles reasonably predict the experimental measurements. Remaining discrepancies can be understood from the limits of the transport model used. We stress that the transport model does not have any free parameters other than the boundary conditions set at the top of the pedestal. The RAPTOR simulations run faster than real-time on a standard laptop (2.6GHz Intel®core$^{\text{TM}}$ i5), taking approximately 1 second to simulate 5 seconds of JET discharge with time steps of 0.1s.

The remainder is organised as follows: section 2 details the multichannel transport equations solved within RAPTOR, and introduces the QLKNN-4Dkin transport model. Section 3 shows the results of the JET simulation and comparison with CRONOS. Conclusions and an outlook to future work is provided in section 4.

2. RAPTOR transport simulations with multiple channels

In this section we list the equations solved in RAPTOR as well as the main assumptions in the code. We also describe the QLKNN-4Dkin transport model in further detail.

2.1. Equations solved in RAPTOR

2.1.1. Basic assumptions and coordinate conventions The RAPTOR code assumes an axisymmetric tokamak equilibrium with right-handed (R, ϕ, Z) cylindrical coordinate system, poloidal flux defined as $\psi(R, Z) = \int_S \mathbf{B_p} \cdot d\mathbf{S_z}$, and I_p and B_0 always positive‡.

In all equations, the normalized square root of the toroidal flux $\hat{\rho} = \rho_{tor,N}$ is used as the spatial variable, defined as $\hat{\rho} = \sqrt{\Phi/\Phi_b}$ with $\Phi(\psi) = \int_{S_\psi} B_\phi dS_\phi$ the toroidal magnetic flux enclosed by a poloidal flux surface, and Φ_b denotes the toroidal flux enclosed by the last closed flux surface (LCFS). We write separate equations for ψ and generic equations for energy and particle transport for each species, as follows.

2.1.2. Flux diffusion equation This is the parabolic PDE used to evolve the distribution of poloidal magnetic flux $\psi(\rho, t)$ in time:

$$\sigma_\| \left(\left.\frac{\partial \psi}{\partial t}\right|_{\hat{\rho}} - \frac{\hat{\rho}\dot{\Phi}_b}{2\Phi_b}\frac{\partial \psi}{\partial \hat{\rho}} \right) = \frac{F^2}{16\pi^2\mu_0\Phi_b^2\hat{\rho}}\frac{\partial}{\partial \hat{\rho}}\left[\frac{g_2 g_3}{\hat{\rho}}\frac{\partial \psi}{\partial \hat{\rho}} \right] - \frac{B_0}{2\Phi_b\hat{\rho}}V_{\hat{\rho}}'(j_{bs} + j_{aux}) \quad (1)$$

$\sigma_\|$ is the neoclassical conductivity, for which we use the expression in the Sauter model [25, 26]. $j_{aux} = \langle \mathbf{j_{aux}} \cdot \mathbf{B} \rangle / B_0$ is the non-inductive current density driven by auxiliary systems. The bootstrap current density $j_{bs} = \langle \mathbf{j_{bs}} \cdot \mathbf{B} \rangle / B_0$ is also computed from the Sauter model. $F = RB_\phi$, $V_{\hat{\rho}}' = \frac{\partial V}{\partial \hat{\rho}}$, $g_2 = \langle |\nabla V^2|/R^2 \rangle$ and $g_3 = \langle 1/R^2 \rangle$ (where $\langle \cdot \rangle$ denote the flux-surface average), are terms which depend on the magnetic equilibrium.

‡ In the COCOS tokamak coordinate convention system [24], RAPTOR has COCOS=11, the same as the ITER convention and with $\sigma_{I_p} = \sigma_{B_\phi} = 1$.

The boundary condition at the magnetic axis $\hat{\rho} = 0$ is $\frac{\partial \psi}{\partial \hat{\rho}} = 0$ while at the plasma boundary $\hat{\rho} = 1$ we use the relation $\frac{g_2 g_3}{\hat{\rho}} \frac{\partial \psi}{\partial \hat{\rho}} = \frac{16\pi^3 \mu_0 \Phi_b}{F}$ to prescribe a Neumann boundary condition for $\frac{\partial \psi}{\partial \hat{\rho}}$ via the total plasma current I_p at the plasma edge.

2.1.3. Energy transport equations We write the energy transport equation, from [27], for a generic species s as:

$$\frac{3}{2}(V'_{\hat{\rho}})^{-5/3} \left(\left.\frac{\partial}{\partial t}\right|_{\hat{\rho}} - \frac{\dot{\Phi}_b}{2\Phi_b} \frac{\partial}{\partial \hat{\rho}} \hat{\rho} \right) +$$

$$\left[(V'_{\hat{\rho}})^{5/3} n_s T_s \right] + \frac{1}{V'_{\hat{\rho}}} \frac{\partial}{\partial \hat{\rho}} \left(-\frac{g_1}{V'_{\hat{\rho}}} n_s \chi_s \frac{\partial T_s}{\partial \hat{\rho}} + \frac{5}{2} T_s \Gamma_s g_0 \right) = P_s \qquad (2)$$

where $T_s(\rho, t)$ and $n_s(\rho, t)$ are the temperatures and densities of the considered species. χ_s is the thermal diffusivity and Γ_s is the convective flux (4). P_s is the sum of power density sources and sinks, which will be discussed later. $g_0 = \langle |\nabla V| \rangle$ and $g_1 = \langle |\nabla V|^2 \rangle$ are again flux-surface averaged terms that depend on the magnetic equilibrium geometry. In practice, (2) is re-written as an equation for $T_s(\rho, t)$ where $n_s(\rho, t)$ is considered an external parameter entering into the equation. To obtain a unique solution, the value of the plasma temperature is prescribed at the plasma boundary or at another radial location close to the edge. This is discussed in more detail in Appendix A.

2.1.4. Particle transport equations The particle transport equation, again following [27], is written for the species s as:

$$\frac{1}{V'_{\hat{\rho}}} \left(\left.\frac{\partial}{\partial t}\right|_{\hat{\rho}} - \frac{\dot{\Phi}_b}{2\Phi_b} \frac{\partial}{\partial \hat{\rho}} \hat{\rho} \right) \left[(V'_{\hat{\rho}}) n_s \right] + \frac{1}{V'_{\hat{\rho}}} \frac{\partial}{\partial \hat{\rho}} \Gamma_s = S_s \qquad (3)$$

with

$$\Gamma_s = -\frac{g_1}{V'_{\hat{\rho}}} D_s \frac{\partial n_s}{\partial \hat{\rho}} + g_0 V_s n_s \qquad (4)$$

where D_s ad V_s are the particle diffusivity and pinch terms (positive V_s corresponds to outward pinch) and S_s are (spatially distributed) sources of particles. Similarly to the energy transport equation, boundary conditions can be given at the plasma boundary or another radial location close to the edge (see Appendix A).

After choosing for which species to solve the transport PDE above, densities of other species and the effective charge are constrained by the quasineutrality condition $n_e = \sum_j Z_j n_j$ (where Z_j is the charge of the jth ion species) and $n_e Z_{eff} = \sum_j Z_j^2 n_j$. If Z_{eff} is prescribed as input to the code then two particle densities can be determined. If Z_{eff} is not prescribed, then only one particle density can be constrained and the others have to be manually prescribed or solved for. Typically one solves for either electrons or main ions. Separate transport of impurity species can in principle be simulated as well but is outside the scope of the transport model applied in this work.

2.2. Treatment of magnetic geometry, sources, and MHD

2.2.1. Treatment of magnetic geometry The (possibly time-varying) terms in the equations that depend on the magnetic geometry $(g_0, g_1, g_2, g_3, F, V'_{\hat{\rho}})$ are assumed to be externally prescribed, as explained in detail in [4]. An external Grad-Shafranov code is used to compute the flux-surface averages for the required times and the results are passed as input to the code. Iterative coupling with a fixed-boundary equilibrium code has been achieved but this is not used in the present paper. All simulations shown in this paper are done with a fixed equilibrium geometry.

2.2.2. Sources and sinks Heating and current drive sources and sinks are either externally prescribed (as $\hat{\rho}$-dependent heat deposition/current density profiles from more sophisticated physics-based source simulation codes), or calculated using simple Gaussian profiles, prescribing deposition location and width, as in [2]. For real-time applications, results from real-time ray tracing calculations (e.g. TORBEAM [22]) can be used. Other sources and sinks are modeled as described in [2]. In the simulations shown in this paper, ohmic power, electron-ion equipartition and bremsstahlung are self-consistently calculated. NBI and ICRH power are taken from physics-based codes, as will be detailed later. Line-radiation is included based on experimentally-estimated profiles.

2.2.3. Sawteeth and NTMs RAPTOR includes modules for simulating sawtooth crashes and Neoclassical Tearing Modes. The sawtooth module is discussed in [12], and uses the Porcelli crash criterion for determining the onset of a sawtooth crash. A Kadomtsev full reconnection model is used for computing the new profiles of magnetic flux, temperature, and particle density following the sawtooth crash [28]. Neoclassical tearing modes are simulated self-consistently by using the Generalized Rutherford Equation to evolve the width of the island, while increasing the thermal transport coefficient in the region of the island to simulate the effect of enhanced transport across flux surfaces reconnected by the island [29], [30].

2.3. Transport coefficients for particles and energy

Solving (2) and (3) requires expressions for χ_s, D_s and V_s for all the species solved. In RAPTOR, several transport models can be chosen. Empirical transport models for the thermal diffusion can be used, such as Bohm-gyroBohm [31], as well as the empirical expression introduced in [2] to simulate improved confinement at low magnetic shear. More recently, a more advanced critical-gradient-based empirical model was introduced [4], which requires prescribing the H-factor and the line-averaged density of the discharge as a function of time. In the present work, we use a novel neural-network emulation of results of the QuaLiKiz quasilinear gyrokinetic transport model, which is described in more detail below.

2.3.1. Neural-network emulation of the quasilinear gyrokinetic transport model QuaLiKiz As described in the introduction, QLKNN-4Dkin is a neural network regression of the QuaLiKiz model in a restricted input dimensionality subspace using a multilayer perceptron network [32]. A first proof-of-principle version of this model was presented in [19], which used adiabatic electrons and only predicted electron heat conductivity. The version used for the present paper has been extended by including kinetic electrons and allows simultaneous evaluation of ion and electron heat conductivities, and electron particle diffusivity and pinch. The training set consists of a database of $\sim 10^5$ fluxes, generated using QuaLiKiz, perturbing 4 inputs (R/L_{T_i}, T_i/T_e, q, $\hat{s}$), covering experimentally relevant ranges. Here, $L_{T_i} \equiv -T_i/\nabla T_i$, q is the safety factor, and $\hat{s} = \frac{\hat{\rho}}{q}\frac{\partial q}{\partial \hat{\rho}}$ the magnetic shear. The gradient in the expression for L_{T_i} is defined with respect to the mid-plane averaged minor radius of the flux surface. The parameter ranges in the training set are identical to those in [19], and are repeated in Table 2. Also the training methodology was similar and for brevity not repeated here.

Only ITG modes were unstable throughout the parameter range of the training set, due to specific values of the input parameters that were kept fixed in the construction of the neural network trainings set. The instabilities found in the discharge simulated here are all in the ITG regime, as corroborated in [33] by an independent linear stability analysis for a similar discharge with nearly identical parameters. This increases the validity of this proof-of-principle version of the QLKNN transport model, when applied to this specific JET discharge. However for increased generality, work is in progress on neural-network regressions of QuaLiKiz including higher input dimensionality and cover ITG/TEM/ETG turbulence regimes [34, 35].

QLKNN4D-kin is 6 orders of magnitude faster than QuaLiKiz itself, predicting full profiles of transport coefficients on millisecond timescales, making it suitable for faster-than-realtime

tokamak simulation. One essential feature of this neural network-emulated transport model is that analytical derivatives can be obtained for the transport coefficients with respect to the plasma profiles, which allows for fully implicit solution of the transport equations (see Appendix A for details), with great advantages in numerical stability and speed.

3. Simulation results

3.1. Simulation of JET shot #73342 with ψ, T_e, T_i and n_e

3.1.1. Simulation setup To demonstrate the multi-channel simulations, we will predict the core profiles of JET discharge #73342 using RAPTOR and the QLKNN-4Dkin transport model. JET #73342 is a C-wall baseline discharge with $B/I_p = 2.7T/2.5MA$, 15MW of NBI heating and 2MW of ICRH power. The discharge has been analyzed in [33], and the reader is referred to that work for further details on the discharge. The results are compared to experimental measurements averaged over 0.5s in the middle of the flat-top phase of the discharge, from Charge Exchange (T_i) and High Resolution Thomson Scattering (T_e, n_e) as well as to simulations using CRONOS with the same QLKNN-4Dkin transport model.

The CRONOS simulation was run for 2.5 seconds with boundary conditions at the pedestal top taken from time-averages of the experimental measurements. It evolves the profiles together with a self-consistent equilibrium from HELENA [36] and uses NEMO-SPOT [37] to calculate the NBI power/ particle deposition, and driven current profiles. No line radiation, ICRH power or MHD effects were included in this simulation.

For the RAPTOR simulation, profiles of the power density to electrons and ions, the particle source profile, and the equilibrium, are taken directly from the final time of the CRONOS simulation, and kept constant. The profile of Z_{eff} was set to a constant value of 1.8 as in CRONOS, which was determined using Bremsstrahlung measurements. The ion density n_i and of one impurity species (assumed to be carbon) were scaled based on the evolving n_e profile, constrained by quasineutrality and Z_{eff}. The line-radiation profile was taken from bolometry data, and was observed to be dominated by edge radiation. ICRH profiles were calculated using the TORIC code [38] within TRANSP [39]. Sawtooth effects were modeled as described in Section 2.2.3, where the critical magnetic shear at $q = 1$ that triggers a sawtooth crash, was tuned to match the experimentallly observed sawtooth period of $\sim$ 10Hz in this phase of the discharge. All these physics effects were considered when comparing the RAPTOR simulations to the experimental data. However, for the benchmark with the CRONOS simulation, the effects of line radiation, ICRH and sawteeth were not included, matching the setup of this particular CRONOS run.

The initial conditions for the RAPTOR simulation were set as follows: For the q profile, we choose the profile at the end of the CRONOS simulation. For T_e, T_i, n_e, we choose profiles 20% higher than the final profiles of the CRONOS simulation, to allow for discernible evolution of the profiles towards stationary state. The simulation is run for 2 JET seconds. Since this is larger than the energy confinement time (< 0.3s) and particle confinement time (< 1.0s), the particle and energy balances are close to their equilibrium. The current distribution time is, however, longer than the simulation time. If left to evolve further, the q profile would change, affecting the transport fluxes via the dependency on q and s. However, in the interest of comparing the results to the CRONOS results with a given q profile, we choose to stop the simulation before the q profile has evolved significantly. The obtained profiles are therefore not the true steady-state profiles, but the profiles corresponding to the q profile given as input.

This is illustrated by the time evolution of the on-axis T_e, T_i, and n_e in the RAPTOR simulation shown in figure 1. The values converge to their quasi-stationary values. In each plot, two time scales of profile evolution are evident. The fast (for T and n) timescale, and a slower evolution associated with the resistive timescale due to the dependence of transport on the q-profile. The n evolution timescale is slower by a factor ~ 5 with respect to the temperature evolution timescale.
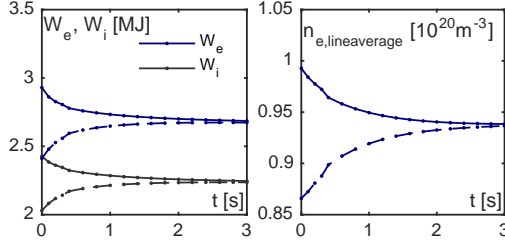
Figure 1. Time-convergence of electron and ion thermal energy W_e, W_i and line integrated density during the RAPTOR simulation. The simulations are initialized with profiles 20% higher/lower than the final profiles of the CRONOS simulation
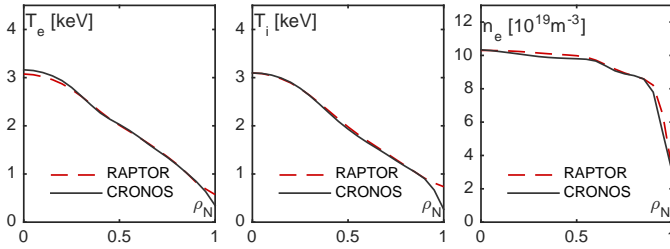

Figure 2. Comparison of RAPTOR profiles using QLKNN-4Dkin transport model to CRONOS simulations using the same transport model. Agreement is excellent except for the region outside $\rho_N > 0.85$, owing to a different treatment of boundary conditions

3.1.2. Benchmark between RAPTOR and CRONOS   The RAPTOR simulation of JET 73342 is shown in figure 2, showing the quasi-stationary state profiles of T_e, T_i and n_e following 2s of profile evolution. The comparison between RAPTOR and CRONOS is excellent, with only minor deviations, attributed to the different methods for smoothing of the transport coefficients employed in RAPTOR and CRONOS. This verifies the implementation of the ion temperature and density transport equations in RAPTOR, as well as the coupling of QLKNN4D-kin.

The boundary condition in the simulations was set at $\hat{\rho} = 0.85$, inside the (ELM-averaged) pedestal. The values of the profiles outside this position are of minor importance for core profile predictions, as the q profile is not significantly affected since we simulate over times shorter than the current diffusion time. The mismatch of the $\hat{\rho} = 1$ boundary condition (due to a lack of consistency in the χ values in that range) is hence not of importance to the comparisons.

Since the simulations shown in Figure 2 did not include a sawtooth model, and since local gyrokinetic turbulence calculations tend to predict stability in the inner radii ($\hat{\rho} < 0.2$), which would lead to significant local peaking, the transport coefficients in the core region were adjusted to yield flat core profiles. This was carried out by imposing minimum values of the transport coefficients in the simulation. In this case $D_{e,min} = 0.1\text{m}^2/\text{s}$, $\chi_{e,min} = 0.15\text{m}^2/\text{s}$, and $\chi_{i,min} = 0.2\text{m}^2/\text{s}$. Also, when $D_e = D_{e,min}$, then $V_e = 0$. Such a prescription effectively acts as a proxy for the effect of sawteeth.

3.1.3. Comparison between RAPTOR and experimental results   In figure 3, a comparison is shown between RAPTOR simulations and experimental measurements. In this case, the effects of line radiation, IC heating and sawteeth are simulated. Including these effects causes an increase in the region $\rho_N < 0.3$ of only $\sim 10\%$ in the sawtooth-averaged profiles. This relatively small difference means that we can employ prescribed minimum transport coefficients in the region affected by sawteeth as a proxy for their impact, allowing larger time steps and faster simulations. We also show the effect of perturbing the boundary condition at $\rho_N = 0.85$ by 10%.
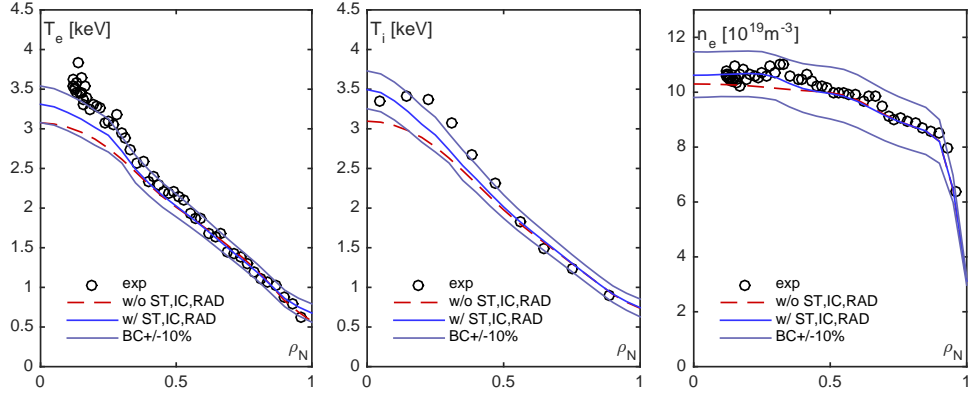
Figure 3. Comparison of RAPTOR profiles using QLKNN-4Dkin transport model and experimental data for JET shot 73342. Red-dashed curves show the results for the case without sawteeth, ICRH heating and line radiation, corresponding to the case shown in Figure 2. Blue-solid curves show the results after adding these effects. For the latter case, the boundary conditions at $\hat{\rho} = 0.85$ are also perturbed by $\pm 10\%$ for sensitivity analysis. Experimental statistical errors are estimated at $\pm 10\%$.

The agreement with experimental data is quantified in Table 1, using a standard deviation figure of merit [40], [41], where, for a quantity f:

$$\sigma = \sqrt{\int_{\hat{\rho}_{in}}^{\hat{\rho}_{BC}} dx \left(f_{sim} - f_{exp}\right)^2} / \sqrt{\int_{\hat{\rho}_{in}}^{\hat{\rho}_{BC}} dx f_{exp}^2} \tag{5}$$

Where f_{sim} is the simulated quantity, and f_{exp} is the measured quantity. We have evaluated the figure of merit with $\hat{\rho}_{min} = 0.2$, to avoid the sawtoothing region. We have included a propagation of the estimated boundary condition uncertainty in the comparison. The predicted and simulated profiles generally agree within $\sim 10\%$, on the order of the measured statistical uncertainties. However, there is a slight systematic underprediction of the temperature gradients in the inner half-radius.

Table 1. Standard deviation figures of merit for the RAPTOR-QLKNN4D-kin JET simulations shown in figure 3

Case	σ_{T_i}	σ_{T_e}	σ_{n_e}
Nominal	10.0%	6.9%	2.7%
BC reduced by 10%	13.1%	10.6%	11.0%
BC increased by 10%	4.0%	6.5%	7.1%

3.2. Comparison of full QualiKiz vs QLKNN-4Dkin for simulation of JET #73342

The RAPTOR+QLKNN-4Dkin predictions in Section 3.1 show good correspondence with the experimental measurements, in spite of the restricted dimensionality of the QuaLiKiz training set used for the neural network fitting. To further understand the sources of discrepancy, we performed a comparison between QLKNN-4Dkin, where many dimensionless quantites are assumed constant, and full QuaLiKiz. For simplicity, we restrict the analysis to a standalone transport coefficient comparison at two radial locations, at $\hat{\rho} = 0.8$ where the gradient agreement is excellent, and at $\hat{\rho} = 0.5$ where the ion and electron temperature gradients were underpredicted by QLKNN-4Dkin.

The dimensionless parameter inputs are shown in Table 2. The first row corresponds to the dimensionless parameters employed in the QuaLiKiz-4Dkin neural network training set.

The values for $R/L_{T_e}, R/L_{n_e}, \epsilon, \alpha, \nu^*$ and Z_{eff} are shown, which are kept fixed in the training database. For the other values (R/L_{T_i}, T_i/T_e, q, $\hat{s}$), which were varied in the training set, we display their ranges. QuaLiKiz is an electrostatic code, hence β is not one of the dimensionless parameters. The second and third row correspond to the full dimensionless parameter set at $\hat{\rho} = 0.5$ and $\hat{\rho} = 0.8$ respectively, taken from the last time point in the nominal RAPTOR simulation.

The comparison between QLKNN-4Dkin and QuaLiKiz standalone calculations is shown in figure 4. The left panel corresponds to calculation at $\hat{\rho} = 0.5$, and the right panel to $\hat{\rho} = 0.8$. The values of the transport coefficients that are computed by the transport model (χ_e, χ_i, D_e, RV_e) are shown for various calculations. All transport coefficients are in SI units, rescaled using the local plasma parameters at each radius from the QuaLiKiz-computed GyroBohm-normalized coefficients.

The set labelled "NN4D" corresponds to the neural network transport model prediction. The set labelled "QLK4D" corresponds to the QuaLiKiz calculated transport coefficients, where the fixed variables from the neural network training are imposed, and $R/L_{Ti}, T_i/T_e, \hat{s}, q$ are all included with their nominal values. The comparison of NN4D and QLK4D indicates the quality of the NN regression of QuaLiKiz. It is evident that the regression provides a good match, particularly at $\hat{\rho} = 0.8$, with some mismatch at $\hat{\rho} = 0.5$ From additional analysis, it appears that these discrepancies are resolved with only a 5% reduction in the input R/L_{Ti}, due to the transport stiffness.

The set labelled "QLK nominal" corresponds to including all parameters taken from the nominal simulation profiles in the full QuaLiKiz calculation. This serves to analyze the impact of including the actual R/L_{Te}, R/L_n, ϵ, α, ν^* and Z_{eff} on the resulting transport coefficients. For the $\hat{\rho} = 0.5$ case, the unstable modes are stabilized (hence transport coefficients are zero). At $\hat{\rho} = 0.8$, only a moderate increase in transport is predicted. This is surprising considering the significant differences in the parameters compared to the fixed training set values, particularly for α, ν^* and R/L_{Te}. However, the destabilizing impact (e.g. of higher ϵ and α) and stabilizing impact (e.g. of lower ν^* and higher Z_{eff}) of the various parameters in this specific regime, tend to cancel each other out, coincidentally. Through an increase of R/L_{T_i} by 15% at $\hat{\rho} = 0.5$, and a reduction by only 7% at $\hat{\rho} = 0.8$, the original heat transport coefficients are recovered, as shown in the rightmost transport coefficient set. This suggests that the underprediction of the temperature gradients at mid-radius would be partially rectified by including all nominal parameters. The only discrepancy is in the particle convection term RV_e, which reverses sign when including all nominal parameters, possibly indicating that the nominal QuaLiKiz prediction would lead to flatter density. However these effects are included in the NN regression of QLK under development [34, 35].

Furthermore, this analysis has confirmed that ITG modes are dominant in this regime, confirming the validity of QLKNN4D-kin for this specific case. In wider parameter regimes, where more mode classes may be unstable, a more general neural network emulation of QuaLiKiz will be used. Indeed, a 10D neural network emulation of QuaLiKiz, including R/L_{Te}, R/L_n, ϵ, ν^*, Z_{eff} and rotation, is currently under construction [34]. Additionally, recent versions of QuaLiKiz [42, 16] do include rotation and will be used to train future versions of the neural network emulations.

3.3. NBI input power scan including core-pedestal coupling

To conclude this section, we present demonstrative time-dependent simulations of an NBI power ramp based on the discharge presented in section 3.1.2 (without sawteeth). Starting from the final profiles of that simulation, a 5-second simulation is carried out with two steps of NBI power. These steps in power are simulated by scaling the NBI heat and particle sources, and driven current profile, by ×1.5 (between 1s and 3s) and ×2 (from 3s until 5s) respectively. This is not an entirely accurate representation of the true effect of increasing NBI power, which would need to consider a self-consistent evaluation of the ionization and fast ion slowing-down physics with respect to the evolving plasma parameters.

Table 2. JET 733424 dimensionless parameters at 2 radial locations at $\hat{\rho} = 0.5$ and $\hat{\rho} = 0.8$, taken from the stationary profiles of the RAPTOR simulation displayed in figure 3. $\hat{\rho}$ is the normalized toroidal flux coordinate. ν^* is the normalized electron collisionality: $\nu^* \equiv \nu_{ei} \frac{qR}{\epsilon^{1.5} v_{te}}$, with $\epsilon = a/R$, $v_{te} \equiv \sqrt{T_e/m_e}$ and $\alpha \equiv q^2 \sum_j \beta_j \left(R/L_{nj} + R/L_{Tj} \right)$. The dimensionless parameters are compared to the values used in the QuaLiKiz runs applied for the 4D neural network (NN) training set (1st row). The first four quantities, which can vary in the NN inputs, are marked as [min–max] indicating the minimum/maximum values in the database used to train the neural network. The ν^* in the first row corresponds to the small variation due to the varying q in the training set, whereas the underlying ν_{ei} remains constant – in any case ν^* remains in the near-collisionless regime. No rotation was included in the simulations considered here.

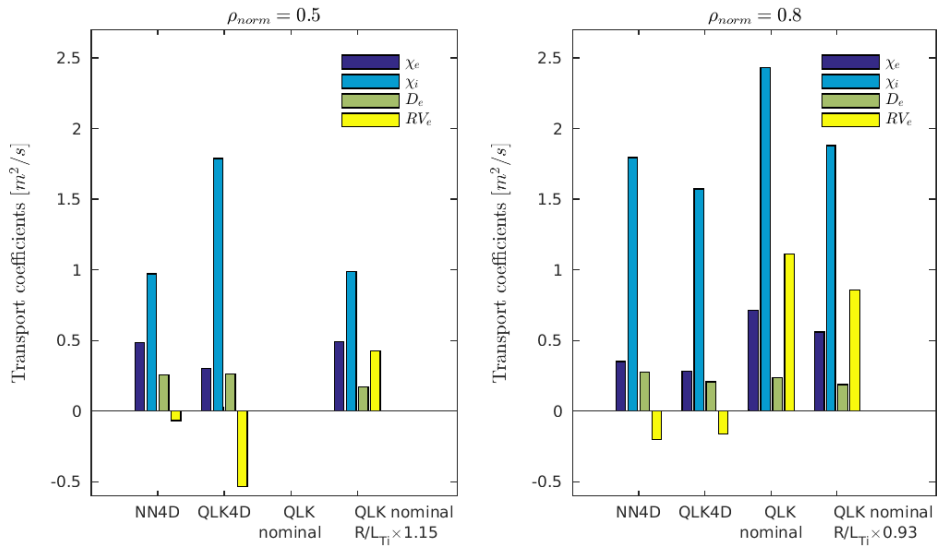| Data source	R/L_{Ti}	T_i/T_e	$\hat{s}$	q	R/L_{Te}	R/L_n	ϵ	α	ν^*	Z_{eff}
QLKNN4D-kin	[2–12]	[0.3–3]	[0.1–3]	[1–5]	6	2	0.167	0	[1.3×10^{-3}–6×10^{-3}]	1.0
JET 73342 simulation $\hat{\rho} = 0.5$	0.4	0.97	0.9	5.0	3.9	1.5	0.172	0.19	0.05	1.8
JET 73342 simulaiton $\hat{\rho} = 0.8$	1.5	0.96	1.7	8.5	10.8	2.6	0.275	0.71	0.17	1.8

Figure 4. Comparison of transport coefficients between QLKNN4D-kin (NN4D) and full QuaLiKiz with the same fixed parameters as used for training the network (QLK4D), full QuaLiKiz with input parameters taken from the experiment, albeit without rotation (QLK nominal), and full QuaLiKiz with manually scaled R/LT_i to match experimental heat fluxes.

In order to simulate the expected increase in pedestal pressure with increasing power, a simple model for the boundary condition is introduced that couples the pedestal top temperature to the core profiles via proportionality to the Shafranov shift parameter: $\Delta_s = \beta_p + l_i/2$. Pedestal confinement is known to improve with Shafranov shift, through an increased critical threshold of peeling-ballooning modes [43]. This simple, ad-hoc model is written (both for ion and electron temperature) as

$$T_{e,i}(\rho_{ped}) = c_{T_b}(\Delta_s - \Delta_s^0) + T_{b0}. \tag{6}$$

The values of Δ_s^0, T_{eb0} are taken from the initial condition of the simulation. The free parameter $c_{T_{eb}}$ is chosen to maintain the H_{98} factor roughly constant during the power scan.

Results of the simulation, which took only a few seconds to execute on single 2.6GHz Intel®core$^{\text{TM}}$ i5 processor, are shown in figure 5. On the top panels, the time-history of the central temperatures, electron density and pressure are shown. Clearly, two time scales can be seen owing to the (fast) confinement times and (slow) current redistribution times. Kinetic and
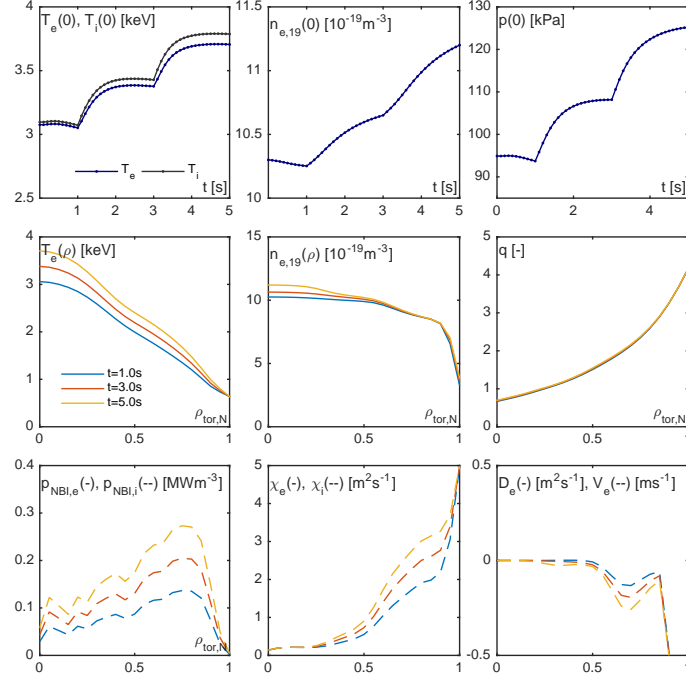
Figure 5. Time-dependent simulation using RAPTOR with QLKNN-4Dkin and a core-pedestal coupling model. Two steps in NBI power are simulated, resulting in different temperature, density and conductivity profiles. Time-traces of temperatures, electron density and pressure are shown in the top panels. Profiles of electron temperature, density, and safety factor q are shown in the middle panels. T_i is omitted since its evolution strongly resembles that of T_e. Sources, thermal conductivities and particle transport coefficients are presented in the bottom panels. The various colours correspond to the profiles at $t = 1$s, $t = 3$s and $t = 5$s.

source profiles at the initial, middle and final time of the simulation are shown in the middle and lower panels.

The changes in the pedestal boundary condition at $\rho_{ped} = 0.85$ are clearly visible as contributing to the increase in temperature. We emphasize that the variation in pedestal boundary condition is due to the response of the first-principle-based core transport model to the power increment. Also, the values in the center region ($\hat{\rho} < 0.2$) are sensitive to the prescribed minimum values of the transport coefficients that are used in regions where ITG modes are stable. The q changes are rather small owing to the long global current redistribution time for JET discharges at these temperatures (typically $\sim$ 10s). The values of H_{98} are $(0.796, 0.7560.755)$ at $t = [1, 3, 5]$ respectively.

Note that while it is known that in reality H_{98} tends to increase with power at JET [44], (owing an the overly pessimistic $P^{-0.7}$ term in the H_{98} scaling), the purpose of this model is to demonstrate the core-pedestal coupling in these time-dependent simulations. In the future, this capability can be used to include a more physics-based pedestal model into these simulations, for example as in [20].

Finally, as a demonstration of the numerical stability of the code, we repeat the simulations shown in figure 5, which were done with time steps of 0.1s, with time steps of 0.25s and 0.5s respectively. As can be witnessed from figure 6, the results on long timescales are practically identical, while on time scales shorter than the confinement time, taking a too long time step results in some inaccuracy during the initial response to the power steps. Still, when only
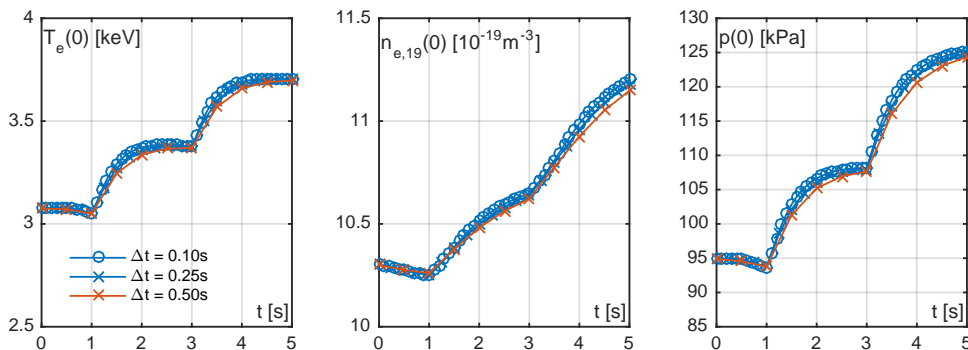
Figure 6. Comparison of time-dependent simulations as shown in Figure 5, executed with different simulation time steps. Owing to the numerical stability of the fully implicit time scheme used by RAPTOR, the simulation is numerically stable even for very large time steps up to 0.5s (which is longer than a JET confinement time). For such large time steps, the transient phase following the power steps can not be captured accurately, while the simulations converge to similar profiles on longer time scales.

long-term time evolution is important, taking long time steps allows even faster simulations to be carried out.

4. Conclusion and outlook

In this paper, we have shown the extension of the RAPTOR control-oriented transport simulator to provide, for the first time, real-time capable simulations of core T_e, T_i, n_e and q profiles in a tokamak.

These simulations presented here used the QLKNN-4Dkin transport model, providing first-principle predictions of the core profiles. By prescribing only the boundary conditions at the top of the H-mode pedestal, good agreement with JET experimental data was found in the region $\hat{\rho} < 0.85$. Within $\hat{\rho} = 0.5$, slightly lower core temperatures were obtained with respect to diagnostic measurements, which could be explained by the restricted input dimensionality of the present version of the QLKNN transport model. Work is in progress to extend the QuaLiKiz database to higher input dimensionality (10D), covering ITG, TEM, and ETG turbulence in a wider range of experimentally relevant input parameters, and to train new neural network transport models using these databases [34]. Furthermore, a JET-specific database spanning 14 QuaLiKiz input dimensions is being constructed, with data points clustered around regions of the parameter space that appear in experimental profiles based on 2000 discharges [35]. A multi-machine extension thereof is also planned.

While here the new capabilities of RAPTOR to solve several transport equations have been illustrated using the QLKNN-4D transport model, other transport models can be used, e.g. Bohm-gyroBohm [31] or the model used in [4].

Further work on improving and extending the RAPTOR code for predictive simulation purposes will focus on coupling to free- and fixed-boundary Grad-Shafranov equilibrium solvers, as well as on fast models to estimate the sources and sinks of power and particles (including radiation). Furthermore, use of core-pedestal models is envisaged, for example using a neural network regression of pedestal heights and widths derived from the EPED model [45], as in [20]. Coupling to the edge would require validated, reduced models of the plasma in that region (e.g. 2-point models) or neural network regressions of more complex, higher dimensional models.

Finally, real-time applications of the new capabilities of RAPTOR are envisaged, for example for real-time monitoring of the discharge evolution with respect to increasingly accurate first-principle-based transport model predictions, allowing a control system to take

appropriate action if discrepancies are detected. The addition of particle density transport equations enables, in particular, estimation of approaching density limits and prediction of impurity accumulation. Also, automated off-line tokamak scenario optimization as in [2], [3], [4] will benefit from more enhanced predictive capability of the transport simulator.

Acknowledgements

Appendix A. Details of the numerical implementation

Details of the numerical implementation for the original RAPTOR code were given in [2] and [4]. The details of the implementation are updated here for completeness.

First, the PDE to be solved for a dependent variable $y(\hat{\rho}, t)$ is written in the generic form

$$m_y \frac{\partial y}{\partial t} = \frac{\partial}{\partial \hat{\rho}}(a_y y) + \frac{\partial}{\partial \hat{\rho}} d_y \frac{\partial y}{\partial \hat{\rho}} + h_y T_e + s_y \tag{A.1}$$

where m_y, a_y, d_y, h_y, s_y may also depend on (ρ, t).

Then, the dependent variable is expressed as a sum of n_{sp} third-order B-spline basis functions: $y(\hat{\rho}, t) = \sum_i^{n_{sp}} \Lambda_\alpha(\rho)\hat{y}_\alpha$ with (scalar) basis function coefficients $\hat{y}_\alpha$. The set of basis functions is chosen so as to identically satisfy the boundary condition at $\rho = 0$. The resulting equation is multiplied by a set of trial functions Λ_β, chosen equal to Λ_α. The result is integrated over the domain $\hat{\rho} = [0, 1]$, where integration by parts is used to remove second-order derivative terms, at the expense of introducing additional boundary terms.

This yields a set of nonlinear ordinary differential equations, written in matrix-vector form as:

$$\mathbf{M}_y \dot{\hat{\mathbf{y}}} = (-\mathbf{A}_y - \mathbf{D}_y + \mathbf{H}_y)\hat{\mathbf{y}} + \mathbf{l} + \mathbf{s} \tag{A.2}$$

where the α, β elements of the matrices are:

$$\mathbf{M}_{y,(\alpha,\beta)} = \int_0^1 m_y \Lambda_\beta \Lambda_\alpha \mathrm{d}\hat{\rho} \tag{A.3}$$

$$\mathbf{A}_{y,(\alpha,\beta)} = \int_0^1 a_y \frac{\partial \Lambda_\beta}{\partial \hat{\rho}} \Lambda_\alpha \mathrm{d}\hat{\rho} \tag{A.4}$$

$$\mathbf{D}_{y,(\alpha,\beta)} = \int_0^1 d_y \frac{\partial \Lambda_\beta}{\partial \hat{\rho}} \frac{\partial \Lambda_\alpha}{\partial \hat{\rho}} \mathrm{d}\hat{\rho} \tag{A.5}$$

$$\mathbf{H}_{y,(\alpha,\beta)} = \int_0^1 h_y \Lambda_\beta \Lambda_\alpha \mathrm{d}\hat{\rho} \tag{A.6}$$

Furthermore

$$\mathbf{s}_{y,\beta} = \int_0^1 \Lambda_\beta s_y \mathrm{d}\hat{\rho} \tag{A.7}$$

$$\mathbf{l}_{y,\beta} = \left[\left(a_y y + d_y \frac{\partial y}{\partial \hat{\rho}} \right) \Lambda_\beta \right]_{\hat{\rho}=1} \quad \text{if } \beta = n_{sp}, \quad 0 \text{ otherwise} \tag{A.8}$$

Note that there are many nonlinearities in (A.2) due to the possible dependence of m_y, a_y, d_y, h_y, s_y on all the profiles that are being solved for.

In this form, the matrix $(-\mathbf{A}_y - \mathbf{D}_y + \mathbf{H}_y)$ does not have full rank, reflecting the fact that we need to impose a boundary condition to solve the PDE. To this end, the last equation of (A.2) is replaced by an equation imposing the boundary condition, written as $\sum_\alpha \Lambda_\alpha(\hat{\rho}_{BC}) = y_{BC}$.

In this way, it is possible to impose time-varying boundary conditions at a time-varying radial location, which is useful when switching from L-mode (where one might prescribes the boundary condition at $\hat{\rho} = 1$) to H-mode (where one might prescribe the boundary condition inside the pedestal top (e.g. $\hat{\rho} = 0.85$)). It is also possible to set y_{BC} as being a function of the plasma profiles, in order to implement core-pedestal coupling.

The user can specify for which channels to solve the PDE. For each of these we obtain a system of ODEs for the vector $\mathbf{y}_j \ \forall j = \{1, \ldots, N\}$, where N is the number of PDEs to be solved. By combining these into a single 'state' vector $x = [\mathbf{y}_1^T, \ldots, \mathbf{y}_N^T]^T$, the system of all the ODEs can be combined into a single equation of the form:

$$0 = \tilde{f}(\dot{x}(t), x(t), z(t)) \tag{A.9}$$

where we have collected all external time-dependent quantities that influence the PDE evolution (sources, transport coefficients, etc) into the variable $z(t)$.

As explained in more detail in [2], this continuous-time nonlinear ODE is discretized in time by a backward Euler method: $\dot{x} = (x_{k+1} - x_k)/\Delta t$, $x = x_{k+1}$. This yields the nonlinear equation:

$$0 = f(x_{k+1}, x_k, z_k) \tag{A.10}$$

With given x_k (from the previous time step) and z_k, this can be solved for x_{k+1}. The solution is sought by Newton-Rapson iterations, where the required Jacobian $\frac{\partial f}{\partial x_{k+1}}$ is calculated analytically at each iteration. While it is cumbersome to implement these analytical Jacobians for the many nonlinear expressions that enter into tokamak transport coefficients and sources, this method of solution allows large time-steps to be taken (even exceeding the energy confinement time).

RAPTOR is entirely implemented in MATLAB. For real-time applications, the code is included in a Simulink® model and C-code is automatically generated and compiled for the target real-time system [46]. For stand-alone use, MATLAB code can also be automatically converted to C-code and compiled, also providing a significant speed-up for the purpose of optimization or systematic scans which may require many runs. The latter option was used to determine the execution times cited in this paper.

References

[1] Humphreys, D. *et al.* 2015 *Physics of Plasmas* **22**
[2] Felici, F. *et al.* 2012 *Plasma Physics and Controlled Fusion* **54** 025002
[3] van Dongen, J. *et al.* 2014 *Plasma Physics and Controlled Fusion* **56** 125008
[4] Teplukhina, A.A. *et al.* 2017 *Plasma Physics and Controlled Fusion* **59** 124004
[5] Vu, N.M.T. *et al.* 2016 *Control Engineering Practice* **54** 34
[6] Maljaars, E. *et al.* 2017 *Nuclear Fusion* **57** 126063
[7] Artaud, J.F. *et al.* 2010 *Nuclear Fusion* **50** 43001
[8] Ou, Y. *et al.* 2007 *Fusion Engineering and Design* **82** 1153
[9] Felici, F. *et al.* 2014 in *2014 American Control Conference* IEEE, Portland, Oregon ISBN 978-1-4799-3274-0 4816–4823
[10] Felici, F. *et al.* 2011 *Nuclear Fusion* **51** 083052
[11] Felici, F. *et al.* 2016 in *26th IAEA Fusion Energy Conference, Kyoto, Japan* EX/P8–33
[12] Piron, C. *et al.* 2016 in *Proceedings of the 29th Symposium On Fusion Technology*
[13] Jenko, F. *et al.* 2000 *Physics of Plasmas* **7** 1904
[14] Candy, J. *et al.* 2003 *Journal of Computational Physics* **186** 545
[15] Bourdelle, C. *et al.* 2016 *Plasma Physics and Controlled Fusion* **58** 014036
[16] Citrin, J. *et al.* 2017 *Plasma Physics and Controlled Fusion*
[17] Staebler, G.M. *et al.* 2013 *Physical Review Letters* **110** 055003
[18] Staebler, G.M. *et al.* 2007 *Physics of Plasmas (1994-present)* **14**
[19] Citrin, J. *et al.* 2015 *Nuclear Fusion* **55** 092001
[20] Meneghini, O. *et al.* 2017 *Nuclear Fusion* **57** 086034
[21] Poli, E. *et al.* 2001 *Computer Physics Communications* **136** 90
[22] Reich, M. *et al.* 2015 *Fusion Engineering and Design* **100** 73
[23] Weiland, M. 2017 in *IAEA Technical Meeting on Energetic Particles in Magnetic Confinement Systems* to be published
[24] Sauter, O. *et al.* 2013 *Computer Physics Communications* **184** 293

[25] Sauter, O. *et al.* 1999 *Physics of Plasmas* **6** 2834
[26] Sauter, O. *et al.* 2002 *Physics of Plasmas* **9** 5140
[27] Hinton, F.L. *et al.* 1976 *Rev. Mod. Phys.* **48** 239
[28] Porcelli, F. *et al.* 1996 *Plasma Physics and Controlled Fusion* **38** 2163
[29] Turri, G. *et al.* 2008 in *Proceedings of the 22nd IAEA Fusion Energy Conference* Geneva EX/P3–6
[30] Basiuk, V. *et al.* 2017 *Plasma Physics and Controlled Fusion*
[31] Erba, M. *et al.* 1998 *Nuclear Fusion* **38** 1013
[32] Haykin, S. 1999 *Neural Networks: A Comprehensive Foundation, second edition* Prentice Hall, Upper Saddle River, New Jersey second edition
[33] Baiocchi, B. *et al.* 2015 *Plasma Physics and Controlled Fusion* **57** 035003
[34] van de Plassche, K. *et al.* 2017 in *44th EPS Conference on Plasma Physics, 26-30 June, Belfast, UK* P2.182
[35] Ho, A. *et al.* 2017 in *44th EPS Conference on Plasma Physics, 26-30 June, Belfast, UK* P5.173
[36] Huysmans, G. *et al.* 1991 in *CP90 Europhys. Conf. on Computational Physics* Amsterdam, NL 371
[37] Schneider, M. *et al.* 2011 *Nuclear Fusion* **51** 63019
[38] Brambilla, M. 2002 *Plasma Physics and Controlled Fusion* **44** 2423
[39] Budny, R.V. *et al.* 2008 *Nuclear Fusion* **48** 75005
[40] ITER Physics Basis Editors et al. 1999 *Nuclear Fusion* **39** 2175
[41] Holland, C. 2016 *Physics of Plasmas* **23** 060901
[42] Cottier, P. *et al.* 2014 *Plasma Physics and Controlled Fusion* **56** 015011
[43] Garcia, J. *et al.* 2015 *Nuclear Fusion* **55** 053007
[44] Challis, C. *et al.* 2015 *Nuclear Fusion* **55** 053031
[45] Snyder, P.B. *et al.* 2012 *Physics of Plasmas* **19** 56115
[46] Felici, F. *et al.* 2014 in *41st EPS Conference on Plasma Physics* Berlin, Germany