WPJET1-PR(18) 19657

EG Highcock et al.

Updates to the Trinity first-principles transport solver and verification against JETTO

Preprint of Paper to be submitted for publication in
Computer Physics Communications

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Updates to the TRINITY first-principles transport solver and verification against JETTO

E. G. Highcock[a], M. Barnes[b], G. Szepesi[c], and JET Contributors[1]

[a] *Department of Physics, Chalmers University of Technology*
[b] *Rudolph Peierls Centre for Theoretical Physics, 1 Keble Road, Oxford, UK*
[c] *Culham Centre for Fusion Energy, Culham Science Centre, Abingdon, UK*

Abstract

The multiscale transport code TRINITY has been updated to evolve an arbitrary number of ion species as well as the profile of toroidal momentum. Additional options for calculating both neoclassical and anomalous transport have been included. A new file format and tool for facilitating data communication and comparison between different transport codes has been created, and used to provide exact validation between TRINITY and the transport code JETTO.

1. Introduction

The multiscale transport code TRINITY was created [1] to calculate the evolution of density and pressure in a toroidal magnetically confined fusion device using first-principles nonlinear simulations to obtain the fluxes of these quantities that result from turbulent transport. Its implicit algorithm made such an achievement possible by minimising the required number of such turbulence simulations. In this brief communication we report several enhancements to TRINITY which significantly extend its capability, and provide validation of its results by comparison with another well-known transport code, JETTO [2, 3]. The enhancements which are described in the subsequent sections are: the extension to allow an arbitrary number of ion species; the incorporation of the momentum transport equation, the incorporation of a full neoclassical solver (NEO [4]); the incorporation of the reduced transport

Email address: physics@edmundhighcock.com (E. G. Highcock)
[1] See the author list of "X. Litaudon et al 2017 Nucl. Fusion 57 102001"

model TGLF [5], allowing extraordinarily rapid generation of solutions with this model; the ability to find a steady-state solution at a single radius only, and the definition of a clean, well-documented file format, with an accompanying conversion tool, which has allowed rapid and easy sharing of data between TRINITY and other transport codes.

The study of turbulence in real magnetic fusion experiments has continued to demonstrate the importance of nonlinear turbulence effects [6, 7, 8]. While reduced, low-cost quasilinear models will continue to be important (and can furthermore be used in combination with large-database-driven interpolation, for example by neural networks [9] to make possible real-time control) the increasing availability of parallel computing resources, combined with the importance of such nonlinear effects (both known and unknown) mean that the use of TRINITY to make predictions of reactor performance will be of ever-increasing value to fusion programme.

2. Trinity equations and algorithm

The implicit algorithm remains substantially the same as that presented in Ref. [1], and thus we present it only briefly here, noting the addition of multiple species and momentum transport (which while given in the equations of Ref. [1] were neither implemented nor tested). We do, however, make a change in notation, as well as presenting additional details of the discretization and normalization which we hope will be of value to those working closely with the TRINITY code. A more casual reader may wish to skip the following three sub-sections.

2.1. Equations

As is usual for a magnetically confined fusion device, we assume that we are modelling a plasma that is confined by a series of nested toroidal magnetic flux surfaces, and that while the plasma is hot enough to be fully ionised, collisions are sufficient to engender a local Maxwellian distribution to first order, which can be described by local values of species densities, rigid body toroidal rotation and species temperatures. The principle behind the TRINITY system of equations is that of a separation of timescale between the evolution of mean, large-scale plasma state (which remains close to a local Maxwellian distribution), and the evolution of the small rapidly changing perturbations to the plasma distribution function, which are responsible for the majority of the fluxes of heat, momentum and particles which then

modify the mean quantities. It is further taken that any non-fluctuating, slowly evolving perturbations to the local Maxwellian (those responsible for neoclassical transport) need not be included in the calculation of the fluctuating perturbations. This system of equations, and the asymptotic ordering which justifies these assertions, is presented in Ref. [10]. Here it suffices to say that we are considering the following equations for the densities (n_s) and pressures (p_s) of each species, and the toroidal angular momentum density (L):

$$\frac{\partial n_s}{\partial t} + \frac{1}{V'}\frac{\partial}{\partial \rho}\left(\frac{V'}{\langle|\nabla\rho|\rangle_\rho}\langle\Gamma_{\rho,s}\rangle_\rho\right) = S_n, \tag{1}$$

$$\frac{\partial L}{\partial t} + \frac{1}{V'}\frac{\partial}{\partial \rho}\left(\frac{V'}{\langle|\nabla\rho|\rangle_\rho}\Pi\right) = S_L, \tag{2}$$

and

$$\frac{\partial p_s}{\partial t} + \frac{1}{V'}\frac{\partial}{\partial \rho}\left(\frac{V'}{\langle|\nabla\rho|\rangle_\rho}\langle Q_{\rho,s}\rangle_\rho\right) = S_p^T. \tag{3}$$

where t is time, $V' = dV/d\rho$ where V is the volume encompassed by the flux surface (and thus V' is related to the flux surface area), ρ is a dimensionless flux label which is 0 at the centre and 1 at the last closed flux surface, $\langle\rangle_\rho$ indicates a flux surface average, ψ_p is poloidal magnetic flux encompassed by the flux surface, $\Gamma_{\rho,s}$ is the particle flux for each species, S_n is the particle source density, Π is the total toroidal angular momentum flux, S_L is the toroidal torque density, $\langle Q_{\rho,s}\rangle_\rho$ is the flux surface averaged energy flux for each species and $S_{p,s}^T$ is the net energy source density for each species. Note that the total momentum flux Π includes the momentum borne by particles:

$$\Pi = \sum_s \left(\langle\pi_{\rho,s}\rangle_\rho + m_s\omega(\rho)\langle R^2\Gamma_{\rho,s}\rangle_\rho\right), \tag{4}$$

and the net heat source includes the effects of collisional species equilibration and turbulent heating:

$$S_{p,s}^T = \frac{3}{2}m_s\sum_u \nu_{s,u}^\varepsilon(T_u - T_s) + \langle H_s\rangle\rho + S_p. \tag{5}$$

In these equations $\pi_{\rho,s}$ is the turbulent toroidal angular momentum flux, m_s is the mass of species s, ω is the toroidal angular frequency, R is the

major radius (distance to the central axis), $\nu^{\varepsilon}_{s,u}$ is the collisional temperature equilibration frequency between species s and species u, T_s is the temperature of each species, H_s is the turbulent heating of species s and S_p is the external heat source density.

Strictly speaking, as laid out in Ref. [10], the quantities $\Gamma_{\rho,s}$, $\pi_{\rho,s}$ and $Q_{\rho,s}$ are the combination of neoclassical fluxes which result from solving the first-order (in $\rho^{\star}$, the ratio of the ion Larmor radius to the system size) drift kinetic equation, and the turbulent fluxes which are obtained by solving the second-order nonlinear gyrokinetic equation. Such is the case when TRINITY is using for example, the NEO code to calculate the neoclassical transport and GS2 to calculate the turbulent transport. However, TRINITY can in fact use a variety of different models to calculate these quantities, from simple formulas such as those in Refs. [11, 12], to reduced models such as TGLF.

There is a last important point to raise. Most flux codes return all fluxes as the amount of the relevant quantity passing through each unit of the physical surface area, A, of the flux surface, so that the total amount passing through the flux surface is this flux multiplied by A; we have defined $\Gamma_{\rho,s}$, $\pi_{\rho,s}$ and $Q_{\rho,s}$ likewise. However, it is very important to note that A is not equal to V', but to $V'/\left\langle|\nabla\rho|\right\rangle_{\rho}$, which accounts for the size of the infinitesimal increment in volume with respect to an infinitesimal increment in ρ, $\delta V = \int_A dA\,|\nabla\rho|\,d\rho$. This explains the factor of $\left\langle|\nabla\rho|\right\rangle_{\rho}$ in equations (1–3).

2.2. Normalisation

With some exceptions, TRINITY itself uses SI units (with some factors of 10 removed for convenience). This facilitates easy comparison with experimental data. Care must be taken, however, when treating $\Gamma_{\rho,s}$, $\pi_{\rho,s}$ and $Q_{\rho,s}$. These are typically calculated in a separate code or codes (which we will refer to as the flux codes) to which TRINITY is linked, and are functions, in general, of n_s, L and T_s, as well as other properties of the global equilibrium. Both groups of parameters, the input and the output, are normalised differently in most flux codes to the way they are normalised in TRINITY, and furthermore there are subtle but important differences between the way that normalisations are chosen in each different code, most notably the definition of the thermal velocity $v_{t,r}$ and the normalising magnetic field B_a.

With this in mind, we may summarize the TRINITY normalisation strategy as follows: divide all quantities into three groups: those within TRINITY, those which are inputs to flux codes, and those which are outputs. For the first group, normalise to SI, with some qualifications. For the second and

Quantity	Definition
T_0	1keV
n_0	$10^{20}\mathrm{m}^{-3}$
B_0	1T
m_p	mass of a proton
m_r	main ion atomic mass
γ	1 or 2, factor in thermal velocity
$v_{t,0}$	$\sqrt{\gamma T_0/m_r}$
Ω_0	$eB_0/m_r c$
ρ_0	$v_{t,0}/\Omega_0$
S_{n0}	$n_0(v_{t,0}/a_0)(a_0/\rho_0)^2$
T_r	main ion temperature
n_r	main ion density
B_r	$I/R_{\mathrm{centre},\rho}$
$v_{t,r}$	$\sqrt{\gamma T_r/m_r}$
Ω_r	$eB_r/m_r c$
ρ_r	$v_{t,r}/\Omega_r$
Γ_{gB}	$n_r v_{t,r}(\rho_r/a_0)^2$
Q_{gB}	$n_r T_r v_{t,r}/(\rho_r/a_0)^2$
π_{gB}	$n_r m_r a_0 v_{t,r}^2(\rho_r/a_0)^2$

Table 1: Normalising quantities in TRINITY. The upper half can loosely be characterised as TRINITY normalisations, the lower as flux normalisations (which are all functions of ρ). I/R is the toroidal magnetic field, and $R_{\mathrm{centre},\rho}$ centre of the flux surface with flux label ρ at the height of the magnetic axis.

third group, define a TRINITY normalisation which is close to the normalisations used by flux codes. When linking to the flux codes themselves, convert between this TRINITY normalisation and the actual normalisations used by the flux codes. Thus depending on the stage of the algorithm, a quantity such as the heat flux may pass through up to three normalisations. This complexity is regrettable but necessary.

Table 1 defines the normalising quantities used in TRINITY, and Table 2 defines some important normalised variables. Using these definitions, equations (1–3) can be rewritten as:

Quantity	Definition
t_N	$(v_{t,0}/a_0)(\rho/a_0)^2 t$
n_{sN}	n_s/n_0
L_N	$L/(n_0 m_p v_{t,0} a_0)$
p_{sN}	$p_s/T_0 n_0$
T_{rN}	T_r/T_0
n_{rN}	n_r/n_0
m_{rN}	m_r/m_p
$\Gamma_{\rho,sN}$	$\Gamma_{\rho,s}/\Gamma_{gB}$
S_{nN}	S_{nN}/S_{n0}

Table 2: Selected normalised quantities in TRINITY.

$$\frac{\partial n_{sN}}{\partial t_N} + \frac{\langle|\nabla_N \rho|\rangle_\rho}{A_N}\frac{\partial}{\partial \rho}\left(A_N \frac{n_{rN} T_{rN}^{3/2}}{B_a^2}\left\langle \Gamma_{\rho,sN}\right\rangle_\rho\right) = S_{nN}, \tag{6}$$

$$\frac{\partial L_N}{\partial t_N} + \frac{\langle|\nabla \rho|\rangle_\rho}{A_N}\frac{\partial}{\partial \rho}\left(A_N \frac{m_{rN} n_{rN} T_{rN}^{2}}{B_a^2}\Pi_N\right) = S_{LN}, \tag{7}$$

and

$$\frac{\partial p_{sN}}{\partial t_N} + \frac{\langle|\nabla \rho|\rangle_\rho}{A_N}\frac{\partial}{\partial \rho}\left(A_N \frac{n_{rN} T_{rN}^{5/2}}{B_a^2}\left\langle Q_{\rho,sN}\right\rangle_\rho\right) = S_{pN}^T. \tag{8}$$

2.3. Discretization

It can readily be seen that equations (6–8) may be written collectively in the following form:

$$\frac{\partial \mathbf{y}}{\partial t_N} + G\frac{\partial \mathbf{F}(\mathbf{y})}{\partial \rho} = \mathbf{S}, \tag{9}$$

where $\mathbf{y}^T = \{n_{sN}, L_N, p_{sN}\}$, and so on. All spatial quantities are discretized on a grid of N points in ρ, with the inner ρ point being a half grid space away from the magnetic axis. This of course means that $\mathbf{y}$ is in fact a vector of length $(2N_s + 1)\,N$ where N_s is the number of species.

The equation is solved implicitly, as will be described below. The time derivative is discretized as follows:

$$\frac{\partial \mathbf{y}}{\partial t_N} = \Delta^0 \mathbf{y}^{m+1} + \Delta^1 \mathbf{y}^m + \Delta^2 \mathbf{y}^{m-1} \tag{10}$$

where $\Delta^{0,1,2}$ may be chosen to implement a second-order backwards difference scheme:

$$\Delta^2 = \frac{(\Delta t)^{m-1} + 2(\Delta t)^m}{(\Delta t)^m \left[(\Delta t)^m + (\Delta t)^{m-1}\right]} \tag{11}$$

$$\Delta^1 = \frac{(\Delta t)^m + (\Delta t)^{m-1}}{(\Delta t)^m (\Delta t)^{m-1}} \tag{12}$$

$$\Delta^0 = \frac{(\Delta t)^m}{(\Delta t)^{m-1} \left[(\Delta t)^m + (\Delta t)^{m-1}\right]} \tag{13}$$

or a simple single step scheme:

$$\Delta^0 = (\Delta t_N)^{-1}, \ \Delta^1 = -(\Delta t_N)^{-1}, \ \Delta^2 = 0 \tag{14}$$

The derivative of $\mathbf{F}$ with respect to ρ is discretized using a centred scheme, with $\mathbf{F}$ evaluated half way between ρ grid points, which is second-order accurate and conservative, and can be adjusted between fully implicit and fully explicit using the parameter α:

$$\frac{\partial \mathbf{F}}{\partial \rho} = \alpha \frac{\mathbf{F}_+(\mathbf{y}^{m+1}) - \mathbf{F}_-(\mathbf{y}^{m+1})}{\Delta \rho} + (1-\alpha) \frac{\mathbf{F}_+(\mathbf{y}^m) - \mathbf{F}_-(\mathbf{y}^m)}{\Delta \rho} \tag{15}$$

Note that in general $\mathbf{S}$ is a function of time and must be treated implicitly as well, but since at the present day TRINITY is primarily used to seek steady-state solutions with constant sources, we will treat $\mathbf{S}$ as a constant in this exposition.

Lastly, we will in fact make the assumption that although the functional $\mathbf{F}$ may depend on $\mathbf{y}$ in its entirety, it is principally dependent on the local value of $\mathbf{y}$ and its first derivative. These first derivatives are calculated using a five-point centered scheme.

2.4. Implicit algorithm

The scheme outlined above requires knowledge of $\mathbf{F}(\mathbf{y}^{m+1})$. This is a nonlinear relationship and thus we solve for $\mathbf{y}^{m+1}$ using Newton's method. Consider a sequence of iterations $\mathbf{y}^{m+1,0} \ldots \mathbf{y}^{m+1,i+1}$ where $\mathbf{y}^{m+1,0} = \mathbf{y}^m$ and $\mathbf{y}^{m+1,i+1} = \mathbf{y}^{m+1} + O(\varepsilon_I)$, where ε_I is an adjustable error threshold. Using the discretizations (10) and (15) we may rewrite (9) as simply

$$\mathbf{R}\left(\mathbf{y}^{m+1}, \mathbf{y}^m\right) = 0 \tag{16}$$

meaning that our Newton iteration can be written as follows:

$$\mathbf{y}^{m+1,i+1} = \mathbf{y}^{m+1,i} - \underline{\underline{J}}^{-1} \cdot \mathbf{R}\left(\mathbf{y}^{m+1,i}, \mathbf{y}^{m}\right), \tag{17}$$

where the Jacobian matrix

$$\underline{\underline{J}} = \frac{\partial \mathbf{R}}{\partial \mathbf{y}}. \tag{18}$$

In general, of course, it is numerically more efficient not to evaluate $\underline{\underline{J}}^{-1}$ but to solve the linear equation

$$\underline{\underline{J}} \cdot \left(\mathbf{y}^{m+1,i+1} - \mathbf{y}^{m+1,i}\right) = \mathbf{R}\left(\mathbf{y}^{m+1,i}, \mathbf{y}^{m}\right). \tag{19}$$

Calculation of $\underline{\underline{J}}$ requires evaluation of derivatives such as $\partial \mathbf{F}/\partial \mathbf{y}$. These are made tractable by the local assumption, that is, that if $\mathbf{F}_\rho$ denotes all elements of $\mathbf{F}$ at a given ρ, and $\mathbf{y}_\rho$ denotes all elements of $\mathbf{y}$ at that ρ, $\mathbf{F}_\rho$ depends only on the value of $\mathbf{y}_\rho$ and $\partial \mathbf{y}_\rho/\partial\rho$, and therefore that

$$\frac{\partial \mathbf{F}_\rho}{\partial \mathbf{y}} \simeq \frac{\partial \mathbf{F}_\rho}{\partial \mathbf{y}_\rho} + \frac{\partial \mathbf{F}_\rho}{\partial\left(\partial \mathbf{y}_\rho/\partial\rho\right)} \frac{\partial \partial \mathbf{y}_\rho/d\rho}{\partial \mathbf{y}_\rho}. \tag{20}$$

The evaluation of the two derivatives of $\mathbf{F}_\rho$ on the RHS of this equation is effectively achieved by calculating the values of all fluxes once with $\mathbf{y}_\rho = \mathbf{y}_\rho^{m+1,i}$, and then once with every element of $\mathbf{y}_\rho$ perturbed, and then once with every value of $\partial \mathbf{y}_\rho/d\rho$ perturbed, a total of $4N_s + 1$ evaluations of the fluxes per ρ grid point if all densities and temperatures, and the flow are being evolved (remembering that one density profile is fixed as we assume quasineutrality). This is of, course, the expensive part of the calculation, but the unconditionally stable implicit algorithm allows large time steps, justifying the cost of calculating the Jacobian.

2.5. Extension to multiple species

The TRINITY algorithm now supports an arbitrary number of ion species. TRINITY treats all ion species uniformly, and makes no distinction between, for example, impurities and main ions. This technical choice allows the incorporation without any further modification of an arbitrary number of distinct ion species. However, it does not preclude the possibility of using different transport models and sources for, say, fusion alpha particles and deuterium ions. Nor does it preclude an approach taken, for example, by T3Core [13, 14], and treat different components of a species (for example,

alpha particles and helium ash, and even alpha particles of different energies) separately. The design of TRINITY allows great flexibility when choosing which profiles to evolve, and choices can be made such as keeping all ion temperatures the same while evolving their densities and so on.

A very small part of the capability of treating multiple species is demonstrated in Figure 1. We consider and compare the results of four different cases. All four cases use magnetic geometry, density (n), heat source (S) and initial temperature data taken from the tokamak profile database [15] for JET shot #42982, chosen because it was considered in the original paper on TRINITY [1]. All four cases evolve the electron temperature (not shown) and the temperatures of one or both of the ion temperatures, hold the density fixed and neglect ion species other than deuterium and tritium. In all four cases, the anomalous heat transport is calculated using TGLF ([5]; see below), with neoclassical heat transport estimated using analytic formulas [11]. The simulations are run until steady state, that is where the heat transport matches the integrated heat source. The distinctions between the four cases A–D are as follows.

In case A, tritium is neglected, the deuterium density is set equal to the electron density, and the heat source S_D is set equal to the total ion heat source S_i.

In case B tritium is included, but its temperature T_T is not evolved and is set equal to the deuterium temperature T_D. The density deuterium is now equal to what provided by the database, and the deuterium heat source is reduced accordingly so that $S_D = S_i n_D/n_e$ where n_e and n_D are the densities of the electrons and deuterium respectively. As a result of this, one would expect that final deuterium temperature might be very close to that of case A, and this can be seen from the top panel of Figure 1. The differences between them result from the fact that the TGLF calculation now incorporates three species, and the inclusion of tritium modifies the predicted heat fluxes for deuterium to a small degree.

In case C the tritium temperature T_T is now evolved, but the tritium heat flux Q_T is artificially set equal to $Q_D S_T/S_D$ where Q_D is the deuterium heat flux calculated by TGLF and S_T is the tritium heat source, equal to $S_i n_T/n_e$. As the collisional equilibrium of temperature is included in TRINITY, the net result of this artificial tritium heat flux should be that the steady-state temperatures of case C should be exactly those of case B, and this is borne out by top two panels of Figure 1.

In the 'full' case, case D, the tritium temperature is evolved with Q_T now

being obtained from TGLF. It can be seen that final values of both T_D and T_T are higher than all previous cases. This can be explained with reference to Figure 2, where it can be seen that initially, Q_T is smaller with respect to the tritium heat source S_T^{int} than Q_D is with respect to S_D^{int}. Roughly speaking, we would therefore expect the tritium temperature to climb higher than T_D; with temperature equilibration keeping them close together and opposing this separation we end up with both temperatures somewhat higher than in case C. The differences in the heat fluxes will be a result of the different local density gradients of the two species as well as the greater mass of the tritium.

2.6. Momentum transport

The first-principles evolution of flow is both very important and very challenging. It is very important because flow-gradient regulation of turbulence is thought to play an important role in many advanced modes of tokamak operation, whether in internal transport barriers [16, 17, 18], hybrid discharges ([19, 20, 21], with some qualifications for JET, see [22]), or in the pedestal (see e.g. [23]). Leaving aside the pedestal, where many challenges remain for first-principles simulations in general [24], first-principles evolution of flow in the core is challenging because of the high demands that including flow effects places on resolution, and because of additional phenomena such as subcritical turbulence [7, 25, 26].

Thus, a full first-principles simulation including flow evolution has yet to be achieved. Nonetheless, it is important to have this capability within TRINITY as the resources for such studies are likely to become available in the near future. As of the publication of this paper, full self-consistent flow evolution (including the ability to handle the case of low flow and intrinsic rotation) is available within TRINITY.

2.7. Single radius mode

We conclude this discussion of the current state of TRINITY by describing its "single radius mode" capability. Owing to the vast expense of a TRINITY simulation when run using a full non-linear turbulence model, it is desirable to find alternative approaches which nonetheless can yield insight. With this end, TRINITY has been extended with a new mode of operation in which a non-linear turbulence model is used at a single radial location. At other radial locations, a simpler model may be used, or alternatively, the fluxes may be set equal to power balance, maintaining the local temperature gradient.

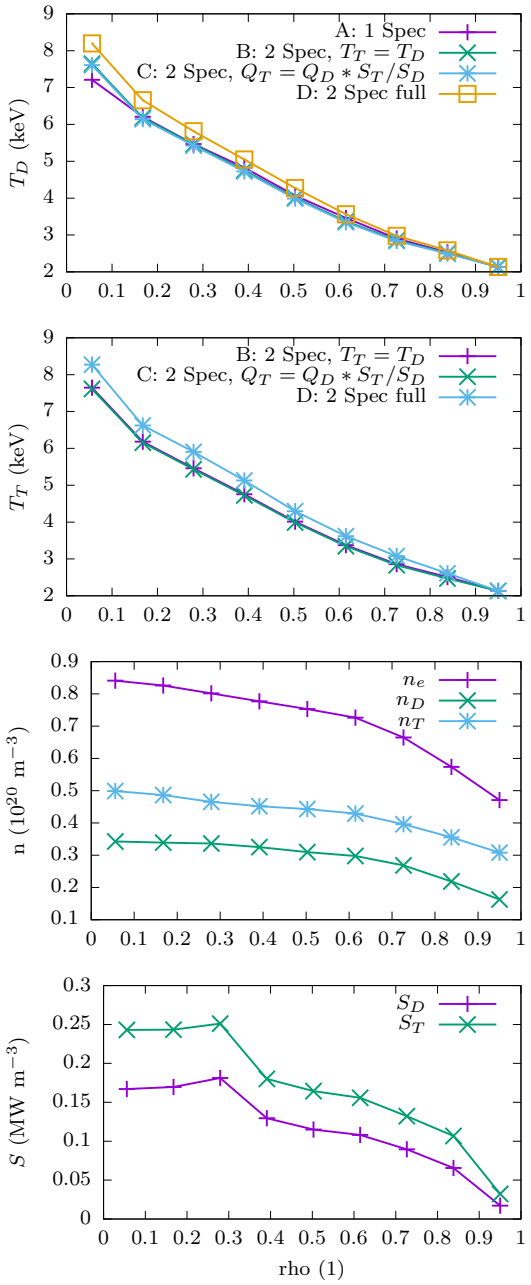

Figure 1: Demonstration of multi-species TRINITY comparing four cases A–D as described in the text. From top to bottom, the panels display: final temperature of deuterium for all four cases; final temperature of tritium for the three cases B–D where it is included; density of all species for the cases B–D where tritium is included; volumetric heat source for deuterium and tritium for cases C–D where the tritium temperature was evolved.

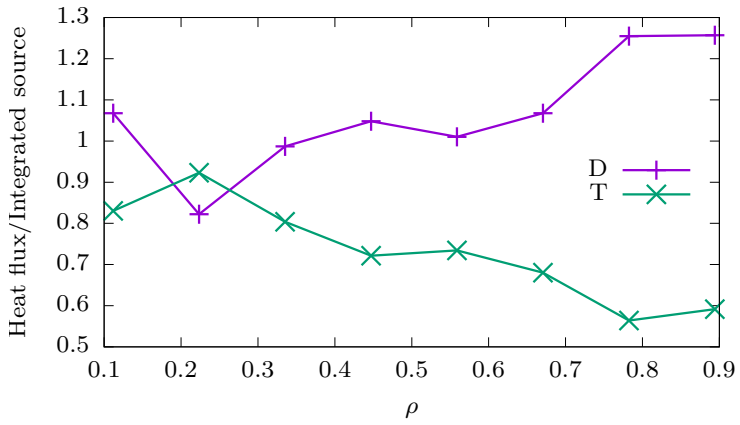

Figure 2: The initial ratio of the calculated heat fluxes for deuterium and tritium to their integrated sources for case D of Figure 1. It can be seen that this ratio is initially lower for tritium than deuterium and this is why the final temperatures are higher in the multi-species case.

This "single radius mode" is a useful tool for examining the effect of the plasma configuration on the turbulence at a local point. Rather than reading off local parameters such as the temperature gradient and then running a local flux calculation, one may use the powerful TRINITY algorithm to find local parameters such that the turbulence matches the steady-state conditions, a generalisation of techniques already in use when, for example, stiff transport means that uncertainty in the experimental measurement of the temperature gradient means that sensitivity scans must be carried out. In other words, "single radius mode" provides a fast automation of the manual sensitivity scans carried out in, for example, Refs. [8, 27]. Lastly, this "single radius mode" may be used in combination with "flux driver mode", whereby TRINITY runs a single local simulation given the initial parameters provided (i.e. without involving any profiles). This allows the rapid running of a local flux simulation based on data provided in any of the file formats supported by TRINITY (see Table 3).

3. Validation of Trinity by comparison with JETTO

3.1. Incorporation of TGLF and NEO

TGLF and NEO, both available as part of a suite of tools from General Atomics, [5, 4], have emerged as a widely-used extensively calibrated reduced

Format	Description
TRINITY NetCDF	A flat portable well-documented binary file type which can store time-dependent profile data and time-dependent equilibrium data. Can be easily accessed by any application using the universally available NetCDF library.
Tokamak Profile Database files	A set of 3 text files which contain time-dependent (and independent) profile data. The file format used by the Tokamak Profile Database [15] (`tokamak-profiledb.ccfe.ac.uk`) which provides a public repository of data from well-diagnosed discharges in several machines.
EXPRO	Files supported by the EXPRO library, including output of the ONETWO transport code.
CHEASE [28] `ogyropsi` file	Text file output by the CHEASE code (generated by setting `NIDEAL=9` within CHEASE)
TRANSP [29] NetCDF file	NetCDF file output by the TRANSP code containing time-dependent profile data.

Table 3: File formats which can be read by TRINITY. A separate tool is distributed with TRINITY which can convert all of these file formats to the TRINITY NetCDF format.

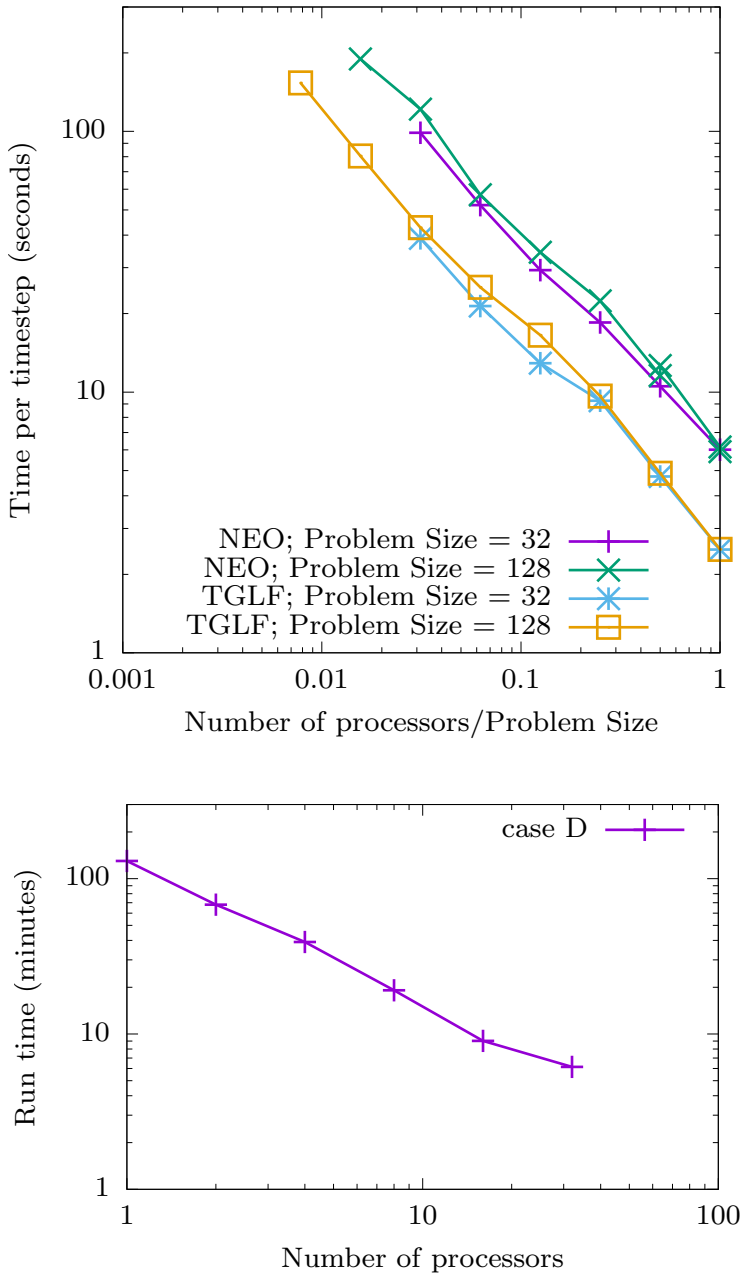

Figure 3: *Top.* The near-perfect strong and weak scaling of TRINITY using the NEO and TGLF flux options. The problem size is related to the size of the Jacobian $\underline{\underline{J}}$, and is equal to the number of cell centres multiplied by one more than the number of evolved profiles. *Bottom* The actual run time for case D of Figure 1, showing that a full Trinity/TGLF run can be accomplished in just over six minutes.

turbulence flux model, and a comprehensive solver for the neoclassical drift kinetic equation respectively. The incorporation of NEO into TRINITY enables state-of-the-art first-principles treatment of neoclassical transport as well as turbulent. The inclusion of TGLF is both important for validation of TRINITY, as we shall discuss, and also demonstrates the power of the TRINITY algorithm to allow as-rapid-as-possible solution of transport equations for a given flux model, and to move away from purely a tool for expensive first-principles modelling but also towards use as a general purpose rapid transport solver.

The interfaces to both codes allows the greatest possible scaling, from use on a single core up to the maximum possible parallelisation: in the case of NEO, the number of flux calculations required per time step across all radii, and in the case of TGLF, to this number multiplied by the internal scaling potential of TGLF, which is the number of evolved modes. The strong and weak scaling with both NEO and TGLF is demonstrated in the top panel of Figure 3. We can see from the bottom panel that a full predictive simulation using TGLF, specifically case D from above, can be completed in only six minutes at maximum parallelisation.

3.2. The TRINITY *NetCDF file format and* PYTHON *tool*

In order to facilitate the comparison between TRINITY and JETTO it was necessary to find means of efficient communication of data between the two. Many integrated modelling tools are installed on only a few systems and are tightly integrated with specific repositories of data, most notably those hosted on the JET systems. By contrast, TRINITY is very portable and of necessity must be run on whichever system can provide sufficient resources. In theory, it would be possible for TRINITY to use the remote access protocols to retrieve data on the fly from some data repository, for example the JET PPFs. In practice, it is much easier and more robust not to rely on such protocols during an expensive TRINITY run, and instead to transmit data in a flexible, self-contained, portable, well-documented file format. Several file formats are in common use for sharing data, for example the outputs of the TRANSP [29] and EFIT [30] codes, as well as file formats supported by the EXPRO library (available as part of the TGLF and GYRO [31] library). However, none of them satisfy concurrently the multiple requirements of clear and unambiguous documentation, the ability to store time-dependent data, storage in a compact binary form, et cetera.

Therefore a new NetCDF-based file format was created, allowing the storage of a large variety of time-dependent data on a flux surface grid, as well as data concerning the magnetic equilibrium. This file format is accompanied by very comprehensive documentation which leaves no ambiguity as to the definitions of quantities and normalisations. As it is hoped that this file format will prove useful to others, the standard is released separately under a creative Commons licence, and can be downloaded and freely shared.

The new file format is already accompanied by a PYTHON tool that can convert the outputs of the JETTO code, as stored in the JET PPFs, into this file format. The tool may readily be extended to work with other integrated modelling tools which can deposit data in the JET PPFs. It is of course this tool that was used in the benchmark subsequently described.

Lastly, a separate tool is provided with the TRINITY distribution which can take any file format which can be read by the EXPRO library and convert it into the TRINITY NetCDF file format.

3.3. Benchmark Description and Results

The benchmark selected for a comparison between TRINITY and JETTO, discharge number 75225, was chosen because of its use in a comprehensive previous study [32]. It is a carbon-wall hybrid discharge from the JET tokamak, the physics of which have been of considerable interest [6, 33]. In this work, however, we are not concerned with this physics: we wish merely to use the benchmark to provide a physically relevant case for comparing the two codes.

In order to construct the benchmark, the JETTO code is used, in interpretative mode, to reconstruct the temperatures, densities, sources and magnetic geometry at a particular time (46 seconds) in the discharge. These are then used as input and both JETTO and TRINITY are run predictively. The temperatures and densities are evolved until the system has reached a steady state, that is, until the fluxes predicted match the integrated sources. Note that as this benchmark does not include the effects of rotation, it is not expected that the predicted profiles will be close to those actually seen in the experiment.

As the codes use different finite difference stencils in radius, agreement is only expected for sufficiently high radial resolution. In this case, a radial resolution of 40 was sufficient to get excellent agreement.

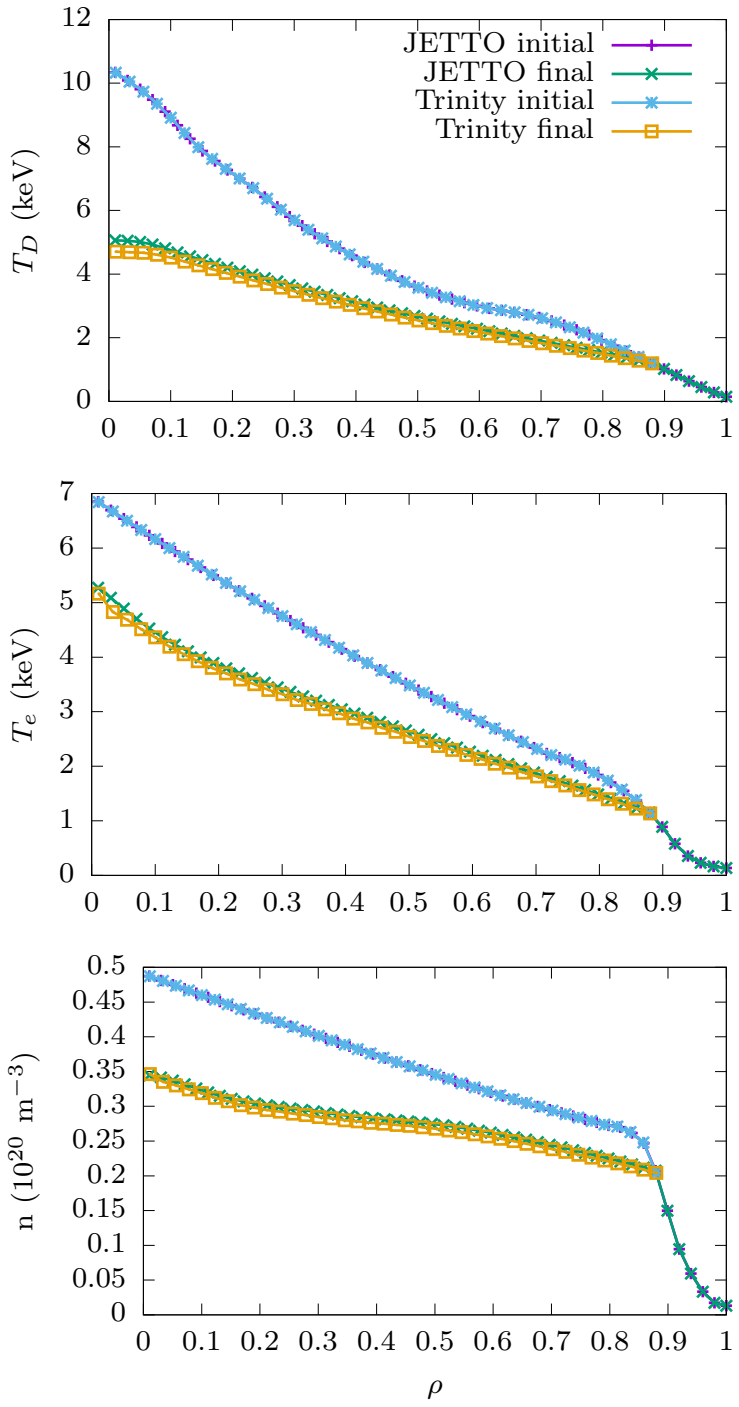

Figure 4: A comparison between TRINITY and JETTO using TGLF for the calculation of the anomalous transport.

4. Summary

In this paper we have described the current capability of the TRINITY transport code, and in particular reported extensions to the breadth of physics that can be studied, to the range of flux models that can be used, and the addition of new technical capabilities which allow easy sharing of data between TRINITY and other transport codes. We have verified that, using the same flux model in the same circumstances, excellent agreement can be achieved between TRINITY and the JETTO transport code, providing faith both in the underlying equations and their implementation within TRINITY. We have demonstrated that the implicit algorithm of TRINITY, which makes transport solutions with non-linear turbulence models possible, makes the generation of transport solutions with reduced models extremely fast. We hope that all these developments, and our reporting of them here, will add to the usefulness of TRINITY to the community.

These developments are all available in TRINITY version 1.0, which is freely available and widely portable. In future versions we will focus on implementing a suitable model of edge transport, as well as first principles models of sources of heat, particles and momentum.

5. Acknowledgements

The authors of this paper are indebted to T. Fülöp, G. Hammett, J. Ball, F. van Wyk, F. Parra, C. Roach and I. Pusztai for helpful discussions, ideas and support. This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014–2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work has also been supported by the Framework grant for Strategic Energy Research (Dnr. 2014-5392) from Vetenskapsrådet. Simulations carried out using the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk) were provided by the Plasma HEC Consortium (EP/L000237/1) and the Collaborative Computational Project in Plasma Physics funded by UK EPSRC (EP/M022463/1).

References

[1] M. Barnes, I. G. Abel, W. Dorland, T. Gorler, G. W. Hammett, and F. Jenko. Direct multiscale coupling of a transport code to gyrokinetic turbulence codes. *Physics of Plasmas*, 17(5):056109, 2010.

[2] G Cenacchi and A Taroni. JETTO: A Free-Boundary Plasma Transport Code. *JET-IR (88)*, 3(2), 1988.

[3] Michele Romanelli, Gerard Corrigan, Vassili Parail, Sven Wiesen, Roberto Ambrosino, Paula Da Silva Aresta Belo, Luca Garzotti, Derek Harting, Florian Köchl, Tuomas Koskela, Laura Lauro-Taroni, Chiara Marchetto, Massimiliano Mattei, Elina Militello-Asp, Maria Filomena Ferreira Nave, Stanislas Pamela, Antti Salmi, Pär Strand, and Gabor Szepesi. JINTRAC: A system of codes for integrated simulation of Tokamak scenarios. *Plasma and Fusion Research*, 9(SPECIALISSUE.2):1–4, 2014.

[4] E A Belli and J Candy. Kinetic calculation of neoclassical transport including self-consistent electron and impurity dynamics. *Plasma Physics and Controlled Fusion*, 50(9):095010, sep 2008.

[5] G. M. Staebler, J. E. Kinsey, and R. E. Waltz. A theory-based transport model with comprehensive physics. *Physics of Plasmas*, 14(5), 2007.

[6] J Citrin, J. Garcia, T. Görler, F. Jenko, P. Mantica, D. Told, C. Bourdelle, D R Hatch, G M D Hogeweij, T. Johnson, M. J. Pueschel, and M Schneider. Electromagnetic stabilization of tokamak microturbulence in a high- β regime. *Plasma Physics and Controlled Fusion*, 57(1):014032, 2015.

[7] F. van Wyk, E. G. Highcock, A. A. Schekochihin, C. M. Roach, A. R. Field, and W. Dorland. Transition to subcritical turbulence in a tokamak plasma. *Journal of Plasma Physics*, 82(6), 2016.

[8] F. van Wyk, E. G. Highcock, A. R. Field, C. M. Roach, A. A. Schekochihin, F. I. Parra, and W. Dorland. Ion-scale turbulence in MAST: anomalous transport, subcritical transitions, and comparison to BES measurements. *Plasma Physics and Controlled Fusion*, 59(11):114003, nov 2017.

[9] J. Citrin, S. Breton, F. Felici, F. Imbeaux, T. Aniel, J. F. Artaud, B. Baiocchi, C. Bourdelle, Y Camenen, and J Garcia. Real-time capable first principle based modelling of tokamak turbulent transport. *Nuclear Fusion*, 55(9):92001, 2015.

[10] I. G. Abel, G. G. Plunk, E. Wang, M. Barnes, S. C. Cowley, W. Dorland, and A. A. Schekochihin. Multiscale gyrokinetics for rotating tokamak plasmas: fluctuations, transport and energy flows. *Reports on progress in physics. Physical Society (Great Britain)*, 76(11):116201, oct 2013.

[11] C. S. Chang and F. L. Hinton. Effect of impurity particles on the finite-aspect ratio neoclassical ion thermal conductivity in a tokamak. *Physics of Fluids*, 25(1493):3314, 1982.

[12] M. Kotschenreuther, W. Dorland, M. A. Beer, and G. W. Hammett. Quantitative predictions of tokamak energy confinement from first-principles simulations with kinetic effects. *Physics of Plasmas*, 2(6):2381, jun 1995.

[13] George J. Wilkie, Ian G. Abel, Matt Landreman, and William Dorland. Transport and deceleration of fusion products in microturbulence. *Physics of Plasmas*, 23(6):1–6, 2016.

[14] G J Wilkie, I Pusztai, I Abel, W Dorland, and T Fülöp. Global anomalous transport of ICRH- and NBI-heated fast ions. *Plasma Physics and Controlled Fusion*, 59(4):044007, apr 2017.

[15] C.M. Roach, M. Walters, R.V. Budny, F. Imbeaux, T.W. Fredian, M. Greenwald, J.A. Stillerman, D.A. Alexander, J. Carlsson, J.R. Cary, F. Ryter, J. Stober, P. Gohil, C. Greenfield, M. Murakami, G. Bracco, B. Esposito, M. Romanelli, V. Parail, P. Stubberfield, I. Voitsekhovitch, C. Brickley, A.R. Field, Y. Sakamoto, T. Fujita, T. Fukuda, N. Hayashi, G.M.D Hogeweij, A. Chudnovskiy, N.A. Kinerva, C.E. Kessel, T. Aniel, G.T. Hoang, J. Ongena, E.J. Doyle, W.A. Houlberg, and A.R. Polevoi. The 2008 Public Release of the International Multi-tokamak Confinement Profile Database. *Nuclear Fusion*, 48(12):125001, dec 2008.

[16] K H Burrell. Effects of ExB velocity shear and magnetic shear on turbulence and transport in magnetic confinement devices. *Physics of Plasmas*, 4(5):1499–1518, 1997.

[17] J.W Connor, T Fukuda, X Garbet, C Gormezano, V Mukhovatov, M Wakatani, the Itb Database Group, the Itpa Topical Group on Transport, and Internal Barrier Physics. A review of internal transport barrier physics for steady-state operation of tokamaks. *Nuclear Fusion*, 44(4):R1–R49, 2004.

[18] P. C. De Vries, E. Joffrin, M. Brix, C. D. Challis, K. Crombé, B. Esposito, N. C. Hawkes, C. Giroud, J. Hobirk, J. Lönnroth, P. Mantica, D. Strintzi, T. Tala, and I. Voitsekhovitch. Internal transport barrier dynamics with plasma rotation in JET. *Nuclear Fusion*, 49(7), 2009.

[19] P.A. Politzer, C.C. Petty, R.J. Jayakumar, T.C. Luce, M.R. Wade, J.C. DeBoo, J.R. Ferron, P. Gohil, C.T. Holcomb, A.W. Hyatt, J. E. Kinsey, R.J. La Haye, M.A. Makowski, and T.W. Petrie. Influence of toroidal rotation on transport and stability in hybrid scenario plasmas in DIII-D. *Nuclear Fusion*, 48(7):075001, 2008.

[20] E. Joffrin, C. Challis, J. Citrin, J. Garcia, J. Hobirk, I. Jenkins, J. Lonnroth, D. C. Mcdonald, P. Maget, P Mantica, M Beurskens, M Brix, P Buratti, F Cristianti, L Frassinetti, C Giroud, F Iimbeaux, M Piovesan, and F Rimini. High Confinement Hybrid Scenario in JET and its Significance for ITER. In *Proceedings of the 23rd IAEA Fusion Energy Conference, Daejon, Republic of Korea*, 2010.

[21] P. Mantica, C. Angioni, C. Challis, G. Colyer, L. Frassinetti, N. Hawkes, T. Johnson, M. Tsalas, P. C. Devries, J. Weiland, B. Baiocchi, M. N A Beurskens, A. C A Figueiredo, C. Giroud, J. Hobirk, E. Joffrin, E. Lerche, V. Naulin, A. G. Peeters, A. Salmi, C. Sozzi, D. Strintzi, G. M. Staebler, T. Tala, D. Van Eester, and T. Versloot. A key to improved ion core confinement in the JET tokamak: Ion stiffness mitigation due to combined plasma rotation and low magnetic shear. *Physical Review Letters*, 107(13):3–7, 2011.

[22] J. Citrin, F. Jenko, P. Mantica, D. Told, C. Bourdelle, R. Dumont, J. Garcia, J.W. Haverkort, G.M.D. Hogeweij, T. Johnson, and M.J. Pueschel. Ion temperature profile stiffness: non-linear gyrokinetic simulations and comparison with experiment. *Nuclear Fusion*, 54(2):023008, feb 2014.

[23] T. Pütterich, E. Wolfrum, R. Dux, and C. F. Maggi. Evidence for strong inversed shear of toroidal rotation at the edge-transport barrier in the ASDEX upgrade. *Physical Review Letters*, 102(2):1–4, 2009.

[24] Ian Abel and Axel Hallenbert. Multiscale Modelling for Tokamak Pedestals. *Submitted to JPP*, 2017.

[25] E. G. Highcock, M. Barnes, G. Colyer, J. Citrin, D. Dickinson, N. Mandell, F. van Wyk, C. M. Roach, A. Schekochihin, and W. Dorland. Trinity Multiscale Transport Code Development for Experimental Comparison. In *Bulletin of the American Physical Society*, 2014.

[26] M. Barnes, F. I. Parra, E. G. Highcock, A. Schekochihin, S. C. Cowley, and C. M. Roach. Turbulent Transport in Tokamak Plasmas with Rotational Shear. *Physical Review Letters*, 106(17):1–4, apr 2011.

[27] N. T. Howard, C. Holland, A. E. White, M. Greenwald, and J. Candy. Synergistic cross-scale coupling of turbulence in a tokamak plasma. *Physics of Plasmas*, 21(11), 2014.

[28] H. Lütjens, A. Bondeson, and O. Sauter. The CHEASE code for toroidal MHD equilibria. *Computer physics communications*, 97:219–260, 1996.

[29] PPPL. TRANSP Home Page. http://transpweb.pppl.gov/home.

[30] L.L. L Lao, H. St. John, R.D. Stambaugh, A.G. Kellman, and W. Pfeiffer. Reconstruction of current profile parameters and plasma shapes in tokamaks. *Nuclear Fusion*, 25(11):1611–1622, nov 1985.

[31] J. Candy and R.E. Waltz. An Eulerian gyrokinetic-Maxwell solver. *Journal of Computational Physics*, 186(2):545–581, apr 2003.

[32] R Bravenec, J Citrin, J Candy, P Mantica, and T Görler. Benchmarking the GENE and GYRO codes through the relative roles of electromagnetic and E B stabilization in JET high-performance discharges. *Plasma Physics and Controlled Fusion*, 58(12):125018, dec 2016.

[33] J. Garcia, C. Challis, J. Citrin, H. Doerk, G. Giruzzi, T. Görler, F. Jenko, and P. Maget. Key impact of finite-beta and fast ions in core and edge tokamak regions for the transition to advanced scenarios. *Nuclear Fusion*, 55(5):053007, 2015.