The CRONOS suite of codes for integrated tokamak modelling

2010 Nucl. Fusion 50 043001

(http://iopscience.iop.org/0029-5515/50/4/043001)

IOP Publishing and International Atomic Energy Agency Nuclear Fusion

Nucl. Fusion **50** (2010) 043001 (25pp) doi:10.1088/0029-5515/50/4/043001

TOPICAL REVIEW

The CRONOS suite of codes for integrated tokamak modelling

J.F. Artaud[1], V. Basiuk[1], F. Imbeaux[1], M. Schneider[1], J. Garcia[1], G. Giruzzi[1], P. Huynh[1], T. Aniel[1], F. Albajar[2], J.M. Ané[1], A. Bécoulet[1], C. Bourdelle[1], A. Casati[1], L. Colas[1], J. Decker[1], R. Dumont[1], L.G. Eriksson[3], X. Garbet[1], R. Guirlet[1], P. Hertout[1], G.T. Hoang[1], W. Houlberg[4], G. Huysmans[1], E. Joffrin[1], S.H. Kim[5], F. Köchl[6], J. Lister[5], X. Litaudon[1], P. Maget[1], R. Masset[1], B. Pégourié[1], Y. Peysson[1], P. Thomas[2], E. Tsitrone[1] and F. Turco[1]

[1] CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France
[2] Fusion for Energy, Josep Pla 2, Barcelona, 08019, Spain
[3] European Commission, Research Directorate General, B-1049 Brussels, Belgium
[4] ITER Organization, 13108 Saint-Paul-Lez-Durance, France
[5] Ecole Polytechnique Fédérale de Lausanne (EPFL), Centre de Recherches en Physique des Plasmas, Association Euratom-Confédération Suisse, CH-1015 Lausanne, Switzerland
[6] Association EURATOM-ÖAW/ATI, Vienna, Austria

Received 5 October 2009, accepted for publication 18 February 2010
Published 16 March 2010
Online at stacks.iop.org/NF/50/043001

Abstract
CRONOS is a suite of numerical codes for the predictive/interpretative simulation of a full tokamak discharge. It integrates, in a modular structure, a 1D transport solver with general 2D magnetic equilibria, several heat, particle and impurities transport models, as well as heat, particle and momentum sources. This paper gives a first comprehensive description of the CRONOS suite: overall structure of the code, main available models, details on the simulation workflow and numerical implementation. Some examples of applications to the analysis of experimental discharges and the predictions of ITER scenarios are also given.

PACS numbers: 52.55.Fa, 52.65.-y, 28.52.Av, 28.52.-s, 52.55.-s, 52.55.Dy, 52.65.Kj, 52.30.-q, 52.35.Ra, 52.50.-b, 52.35.Hr, 52.55.Wq, 52.40.Mj, 52.55.Pi, 52.65.Cc, 52.65.Ff, 52.65.Pp, 25.60.Pj, 52.55.Rk, 52.25.Vy, 52.25.Ya, 52.30.Cv, 52.55.Tn, 52.25.Xz, 52.70.-m, 02.70.-c, 01.50.Lc, 01.30.L-

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Integrated modelling of burning plasmas is an essential step for the realization of the ITER programme. As for any nuclear reactor, the existence of a reliable simulator of the ITER operation is mandatory, both for nuclear safety reasons and for the optimization of the operation of such a costly device. A simulator will be a computer code capable of predicting the full time evolution of a tokamak discharge, having as input the same control and feedback parameters that are used to pilot the machine itself. Since ITER will be a nuclear reactor of a completely new type, such a simulator will include sophisticated physics models, and the experimental validation of such models in present and future tokamak devices will also be an essential requirement. The model development and validation for the various ingredients of a burning plasma discharge of course covers the full range of research subjects in fusion plasma theory. Assembling sophisticated fusion physics codes into a comprehensive tokamak discharge simulation requires a dedicated platform that handles the physics workflow and data management. Such an integration poses specific problems, both at the physics level (models compatibility, degree of approximation) and at the numerical level (algorithm optimization, parallelization, etc).

This has been the general motivation basis for the development of the CRONOS suite of codes. The core of

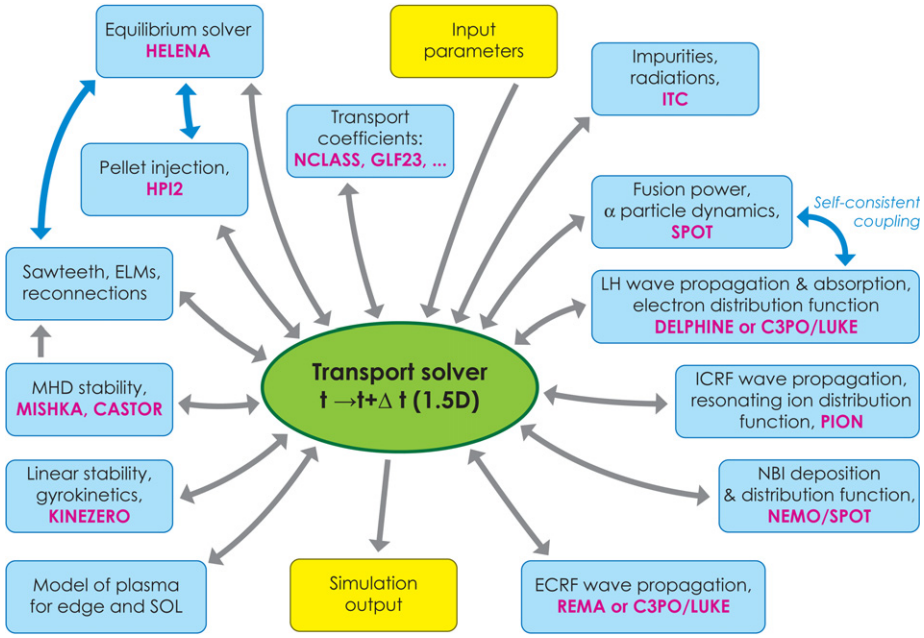

Figure 1. Global CRONOS workflow with the main sophisticated physics modules around the core transport equation solver.

CRONOS is a numerical code which solves the transport equations for various plasma fluid quantities, i.e. particle and impurities densities, ion and electron temperatures, current density, plasma momentum, etc. This is done in one dimension, i.e. the magnetic flux coordinate associated with the tokamak minor radius, which is the most relevant space coordinate after averaging over the various tokamak symmetries. CRONOS integrates, in a modular structure, general 2D magnetic equilibria, radiation and particle losses, several heat, particle and impurities transport models, as well as heat, particle and momentum source modules, associated, e.g., with neutral beams, radio-frequency waves, pellet ablation, etc. An essential ingredient for reactor simulations is of course the kinetic description of the alpha particle source generated by the fusion reactions, which is also provided by a module of CRONOS. Various feedback algorithms, the same as those used in the real-time control of a tokamak discharge, are built in since the CRONOS conception, with both the purpose of better comparison with the experimental realization of tokamak discharges and of testing new or complex feedback schemes before their implementation.

The validation of such a complex combination of codes requires sophisticated tools for detailed comparison with experimental results. These include direct access to the experimental databases of various tokamaks; the possibility of running the same code not only in the predictive mode but also in the so-called interpretative mode, i.e. using the experimental profiles of various fluid quantities to deduce the transport coefficients; sophisticated post-processing tools to evaluate, from the computed quantities, the predicted output of several diagnostics (these tools are often called 'synthesized diagnostics'). These functions are fully implemented in CRONOS and have been extensively used to validate the overall functioning of the code, as well as various specific modules, against experimental data produced on different machines, e.g., Tore Supra, JET, FTU, TCV, DIII-D. This validation gives confidence in the output of the code, which has already been used to predict relevant scenarios of ITER and DEMO.

A key feature of the CRONOS suite is that all these functionalities are embedded in a unique environment, controlled by a user-friendly graphical interface: access to experimental data for generating the input, multi-diagnostic profile fitting, visualization and edition of input data, parametrization and launching of the simulations, visualization of the results and comparison with experimental databases or other simulations. This makes CRONOS a quite effective and versatile platform capable of handling all main missions of integrated modelling: data validation, analysis of experiments, model validation, diagnostic studies, predictive simulations.

The CRONOS suite has been designed as an open source software, generally available to the fusion community for scientific purposes. Installed on the computers of several fusion research institutes in Europe, USA, India and China, it has been used to analyse experimental data from many tokamaks: Tore Supra, JET, FTU, TCV, DIII-D, HL-2A, JT60-U, ASDEX Upgrade. Scenario studies have also been carried out with CRONOS for devices in construction phase such as SST1, EAST and ITER, as well as DEMO for the longer term. Diagnostic or heating/current drive system optimization in relation to plasma scenario design can also be carried out, as has been done for the study of a possible lower hybrid current drive (LHCD) system for ASDEX Upgrade [1].

For each kind of physics problem, the CRONOS suite includes a series of available modules ranging from basic to state-of-the-art degree of physics sophistication (see figure 1). From the beginning of CRONOS development, the philosophy has been to include the most comprehensive physics solvers available. Among those, fast electron physics modules capable of treating LHCD–electron cyclotron current drive (ECCD) synergy or the interaction between LHCD and fast ions, the most sophisticated existing modules for synchrotron radiation

and an Orbit Following Monte Carlo code for the dynamics of NBI or fusion-born fast ions. CRONOS has also been developed with a particular care concerning the numerics of the coupled transport equations. Those are solved as a consistent set of finite difference equations, with a specific convergence loop on the transport coefficients non-linearities, using an automated adaptive time stepping. The particular issue of consistency between the poloidal flux transport equation (current diffusion) and the plasma equilibrium is addressed in CRONOS with a specific convergence loop carried out at each call of the equilibrium solver. These numerical features help to ensure that the code is correctly solving the coupled transport equations, in particular for transient phenomena.

Other integrated modelling codes, addressing similar physics to CRONOS, have been developed throughout the world, such as ASTRA, CORSICA, JETTO, TRANSP (this list is not exhaustive). Comparing point by point these codes would require a dedicated review work and is outside the scope of this paper. Broadly speaking, all these codes solve a similar set of core transport equations (with some differences such as normalized versus not normalized radial coordinate, single particle transport equation for the electrons or multiple ones for each individual ion species). They all include an equilibrium solver, although numerical details on how it is used differ (the convergence loop on the equilibrium and current diffusion in CRONOS is quite distinctive, see section 3.1). They all include a variety of source term and transport modules (see sections 5 and 2.8), with various degrees of sophistications among them. Nevertheless, one can here try to outline some of the most distinctive features. One of them is the fact that CRONOS has been written almost from scratch starting in 1999, with a modern framework and programming techniques (see section 11) while the other codes are older and include several legacy parts that may be more difficult to evolve. On the physics side, the strong point of CRONOS is to offer a quite comprehensive suite of state-of-the-art modules for all main heating and current drive schemes used on tokamaks (see section 5). Also, CRONOS includes an integrated graphical environment allowing setting up all simulation parameters, preparing input profiles from experimental data, post-processing (including synthesized diagnostics) and the specific visualization tools for the comparison of simulation results with experimental data. This feature makes it a versatile code for interpretation of experiments, diagnostic studies and data validation, as well as for predictive discharge simulation and model testing. Conversely, some of the codes quoted above are more specifically designed either for interpretative or predictive studies. Comparatively, one of the weak points of CRONOS is the interaction with the scrape-off layer and the wall which are treated presently in CRONOS with quite simplistic models, while core-edge integration is much more achieved in codes such as JETTO and CORSICA.

This paper gives a first comprehensive description of the CRONOS suite, including some examples of applications to (i) the analysis of experimental discharges and (ii) the predictions of ITER scenarios. The layout of the paper is the following. Section 2 presents the basic equations of the core solver and the transport models included in the code. Section 3 is devoted to the magnetic equilibrium modules, which define the general geometry of the system, adopted by all the modules. Section 4 describes the CRONOS workflow, i.e. the suite of actions carried out for solving the transport equations self-consistently with equilibrium and source terms. The source and loss terms are described in section 5: they include heating, current drive, particle and momentum sources, as well as radiation losses; these modules are the most complex elements of the suite of codes and those which use most of the computing time of a typical simulation. Section 6 accounts for the implementation in the code of transient events, such as, e.g., MHD instabilities or pellet ablation, which have a strong impact on the discharge evolution; however, their details are solved outside the transport equations. Feedback algorithms are described in section 7. Sections 8 and 9 describe the post-processing tools (including synthesized diagnostics) and the connection to experimental databases, respectively. Section 10 presents some examples of application of the code to the analysis of experimental discharges and predictions for ITER. Section 11 gives some technical information about the organization of the software, in particular the data model and graphical interface. Conclusions and prospects for further developments are presented in section 12.

2. Transport equations

The core of CRONOS consists of the transport equations for the poloidal flux, electron and ion heat, particles (electrons only up to now) and momentum. Two additional generic transport equations are already built in the code, so that the user can easily compute the transport of any additional quantity. The transport equations are solved for the core plasma only (from the magnetic axis up to the last closed flux surface). Detailed modelling of the plasma edge and plasma–wall interactions (scrape-off layer and beyond) is beyond the scope of CRONOS so far, although this topic is considered for future extensions of the code.

The transport equations deal with radial transport only. Therefore most variables in CRONOS are 1D radial profiles (time dependent), corresponding to the average of the physical quantities over an infinitesimal volume around a flux surface of given radial position. The radial grid is equally spaced in normalized toroidal flux coordinate $x = \rho/\rho_m$, with $\rho = \sqrt{\Phi/\pi B_0}$, where Φ is the toroidal magnetic flux, B_0 the vacuum magnetic field at a given major radius R_0 (usually taken at the centre of the vacuum vessel) and ρ_m is the value of ρ at the last closed flux surface. The normalized radial grid does not depend on time, but ρ_m is time dependent and calculated by the equilibrium solver (see section 4). The transport equations used in CRONOS are based on the derivation carried out in [2], which assumes that B_0 does not vary with time, which is almost always the case during tokamak discharges.

The equilibrium is calculated consistently with the evolution of the plasma profiles, thus the transport equations are solved with self-consistent equilibrium. In addition, the 1D quantities deduced from the transport equations can be mapped as a function of other coordinates (poloidal flux, 2D (R,Z) map).

2.1. Current diffusion equation

The current diffusion equation is solved in terms of the poloidal flux Ψ, as given below [2, 3]:

$$\left.\frac{\partial\Psi}{\partial t}\right|_x = \frac{\langle|\nabla\rho|^2/R^2\rangle}{\mu_0\sigma_\parallel\rho_m^2\langle 1/R^2\rangle}\frac{\partial^2\Psi}{\partial x^2} + \left\{\frac{\langle|\nabla\rho|^2/R^2\rangle}{\mu_0\sigma_\parallel\rho_m^2\langle 1/R^2\rangle}\frac{\partial}{\partial x}\right. \times\left[\ln\left(\frac{V'\langle|\nabla\rho|^2/R^2\rangle}{F}\right)\right] + \left.\frac{x}{\rho_m}\frac{\mathrm{d}\rho_m}{\mathrm{d}t}\right\}\frac{\partial\Psi}{\partial x} + \frac{B_0}{\sigma_\parallel F\langle 1/R^2\rangle}j_{\mathrm{ni}}, \quad (1)$$

where $(\partial\Psi/\partial t)|_x$ is the time derivative of the poloidal flux at a given radial position x, $\sigma_\parallel$ denotes the parallel conductivity, calculated by the neoclassical module (see section 5), F the diamagnetic function (see section 4), j_{ni} the current density driven by the non-inductive sources, R the major radius and μ_0 the magnetic permeability of free space ($4\pi\times10^{-7}$ in MKS units). The notation $\langle\ \rangle$ indicates a magnetic flux surface average, defined as the volume average of a quantity around a flux surface of radius ρ, i.e. in an elementary volume $\mathrm{d}V$ enclosed between two magnetic surfaces distant of $\mathrm{d}\rho$:

$$\langle\mathrm{A}\rangle = \frac{1}{\mathrm{d}V}\int_V A\,\mathrm{d}V = \frac{1}{\mathrm{d}V}\int_\rho^{\rho+\mathrm{d}\rho}\mathrm{d}\rho\int_0^{2\pi}\mathrm{d}\varphi\int_0^{2\pi}AJ\,\mathrm{d}\theta = 2\pi\frac{\mathrm{d}\rho}{\mathrm{d}V}\int_0^{2\pi}AJ\,\mathrm{d}\theta = \frac{2\pi}{V'}\int_0^{2\pi}AJ\,\mathrm{d}\theta, \quad (2)$$

where φ is the toroidal angle, θ the poloidal index, J the Jacobian of the coordinates transform from the usual (R,Z,φ) system to the flux surface related coordinate system (ρ,θ,φ), so that $\mathrm{d}V = \mathrm{d}R\cdot R\,\mathrm{d}\varphi\cdot\mathrm{d}Z = J\,\mathrm{d}\rho\,\mathrm{d}\theta\,\mathrm{d}\varphi$. J is calculated as

$$J = R\begin{vmatrix}\frac{\partial R}{\partial\rho} & \frac{\partial R}{\partial\theta}\\ \frac{\partial Z}{\partial\rho} & \frac{\partial Z}{\partial\theta}\end{vmatrix} = R\left(\frac{\partial R}{\partial\rho}\frac{\partial Z}{\partial\theta} - \frac{\partial Z}{\partial\rho}\frac{\partial R}{\partial\theta}\right). \quad (3)$$

The volume enclosed inside the magnetic surface of flux coordinate ρ is given by $V(\rho) = \iiint_{\mathrm{inside}\,\rho}\mathrm{d}R\,\mathrm{d}Z\cdot R\,\mathrm{d}\varphi = \int_0^\rho\mathrm{d}\rho\int_0^{2\pi}\mathrm{d}\varphi\int_0^{2\pi}J\mathrm{d}\theta = 2\pi\int_0^\rho\mathrm{d}\rho\int_0^{2\pi}J\,\mathrm{d}\theta$. The quantity V' appearing in equation (2) is defined as $V' = (\mathrm{d}V/\mathrm{d}\rho) = 2\pi\int_0^{2\pi}J\,\mathrm{d}\theta$. All the averaged quantities above, as well as those linked to the flux surface geometry, are computed by the equilibrium solver (see section 4).

The non-inductive current density j_{ni} is the sum of the contributions of the various non-inductive current sources, as calculated by the external modules (see section 5). It includes the bootstrap current, the current driven by neutral beam injection (NBI), lower hybrid (LH) waves, electron cyclotron (EC) waves, ion cyclotron (IC) waves, as well as an optional generic current source 'EXT'.

$$j_{\mathrm{ni}} = j_{\mathrm{boot}} + j_{\mathrm{NBI}} + j_{\mathrm{LH}} + j_{\mathrm{IC}} + j_{\mathrm{EC}} + j_{\mathrm{EXT}}.$$

All these current densities are defined in terms of current parallel to the local magnetic field: $j = \langle\vec{J}\cdot\vec{B}\rangle/B_0$.

2.2. Electron heat transport equation

The code solves the equation describing the conservation of the thermal energy flowing through the electrons as derived for the neoclassical theory in [2]:

$$\frac{3}{2}\frac{\partial}{\partial t}(P_\mathrm{e}V'^{\frac{5}{3}}) + V'^{\frac{2}{3}}\frac{\partial}{\partial\rho}[V'\langle|\nabla\rho|^2\rangle(q_\mathrm{e}+\lambda T_\mathrm{e}\Gamma_\mathrm{e})] = V'^{\frac{5}{3}}Q_\mathrm{e}, \quad (4)$$

where P_e is the electron pressure, T_e the electron temperature, q_e the heat flux flowing through electrons and Γ_e the electron particle flux. Those fluxes are calculated by the transport modules as described below. The value of the coefficient λ is set by the user, being either $0, \frac{3}{2}$ or $\frac{5}{2}$ depending on the formulation of the transport model. Note that rigorously the value of λ must be $\frac{5}{2}$ in order to obtain the $\frac{5}{3}$ exponent for V' in the first term of the above equation, which leads to the correct expression for describing adiabatic compression $(\partial/\partial t)(P_\mathrm{e}V'^{\frac{5}{3}})$ [2]. Nevertheless, depending on the definition of the transport model output, it may be convenient to use other values for λ: models calculating directly the total energy flux (including the heat convected by the particle flux) should use $\lambda = 0$ in the expression above (and replace q_e by the total energy flux). This approximation is equivalent to overestimating the energy flux by a term $\frac{5}{2}P_\mathrm{e}\langle\vec{u}_\rho\cdot\vec{\nabla}\rho\rangle$, where $\vec{u}_\rho$ is the velocity of the flux surface in the laboratory frame. This term is negligible in most practical cases and is moreover ignored in the formulation of most turbulent transport models.

The source term Q_e (right-hand side of equation (4)) is the sum of the following contributions:

$$Q_\mathrm{e} = -Q_\mathrm{ei} + Q_\Omega + Q_\mathrm{neo} + Q_\mathrm{e,lh} + Q_\mathrm{e,nbi} + Q_\mathrm{e,icrh} + Q_\mathrm{e,ecrh} + Q_{\mathrm{e},n0} + Q_\mathrm{e,ext} - Q_\mathrm{rad} - Q_\mathrm{brem} - Q_\mathrm{cyclo} + Q_\mathrm{e,fus} + Q_\mathrm{e,rip},$$

respectively, electron–ion collisional energy transfer, ohmic, neoclassical contribution $Q_\mathrm{neo}\approx-(\Gamma_\mathrm{e}/n_\mathrm{e})(\partial P_\mathrm{i}/\partial\rho)$ [2], LH, NBI, ICRH, ECRH, charge exchange, optional additional source 'EXT', line radiation, bremsstrahlung, synchroton radiation, fusion reactions, energy losses induced by toroidal magnetic field ripple. The source terms are calculated by the external source modules (see section 5).

The electron heat flux can be calculated either as the sum of a diffusive and a convective terms:

$$q_\mathrm{e} = -\mathrm{K}_\mathrm{e}\frac{\partial T_\mathrm{e}}{\partial\rho} - P_\mathrm{e}V_\mathrm{e}^q$$

or as a full transport matrix formulation:

$$q_\mathrm{e} = -\mathrm{K}_\mathrm{e}\frac{\partial T_\mathrm{e}}{\partial\rho} - \mathrm{K}_\mathrm{e}^\mathrm{i}\frac{T_\mathrm{e}}{T_\mathrm{i}}\frac{\partial T_\mathrm{i}}{\partial\rho} - \mathrm{K}_\mathrm{e}^n\frac{T_\mathrm{e}}{n_\mathrm{e}}\frac{\partial n_\mathrm{e}}{\partial\rho} - \mathrm{K}_\mathrm{e}^E\frac{T_\mathrm{e}E_\parallel}{\mu_0 B_\mathrm{p}} - P_\mathrm{e}V_\mathrm{e}^q$$

or even

$$q_\mathrm{e} = -\mathrm{K}_\mathrm{e}^\mathrm{p}\frac{\partial P_\mathrm{e}}{\partial\rho} - \mathrm{K}_\mathrm{e}^\mathrm{i}\frac{T_\mathrm{e}}{T_\mathrm{i}}\frac{\partial T_\mathrm{i}}{\partial\rho} - \mathrm{K}_\mathrm{e}^n\frac{T_\mathrm{e}}{n_\mathrm{e}}\frac{\partial n_\mathrm{e}}{\partial\rho} - \mathrm{K}_\mathrm{e}^E\frac{T_\mathrm{e}E_\parallel}{\mu_0 B_\mathrm{p}} - P_\mathrm{e}V_\mathrm{e}^q.$$

Note that the convective term V_e^q is a pinch term (positive V_e^q means inward flux).

The electron–ion energy exchange is written, using the Coulomb logarithm Λ_ei and the energy confinement time τ_e, as [2]

$$\Lambda_\mathrm{ei} = 15.2 - \frac{1}{2}\log\left(\frac{n_\mathrm{e}}{10^{20}}\right) + \log\left(\frac{T_\mathrm{e}}{10^3}\right),$$

$$\tau_e = \frac{12\pi^{\frac{3}{2}}\varepsilon_0^2}{e^4}\sqrt{\frac{m_e}{2}}\frac{(eT_e)^{\frac{3}{2}}}{n_e\Lambda_{ei}},$$
$$Q_{e,i} = \frac{3}{2}\sum_j \frac{n_j Z_j^2}{A_j}\frac{e(T_e - T_i)}{(m_p/2m_e)\tau_e}$$

with temperature in eV and density in m^{-3}.

2.3. Ion heat transport equation

The ion heat transport equation has a formulation similar to the electron heat transport equation. The code solves the equation for the total (all species) thermal ion pressure:

$$\frac{3}{2}\frac{\partial}{\partial t}\left(P_i V'^{\frac{5}{3}}\right) + V'^{\frac{2}{3}}\frac{\partial}{\partial \rho}\left[V'\left\langle|\nabla\rho|^2\right\rangle(q_i + \lambda T_i\Gamma_i)\right] = V'^{\frac{5}{3}}Q_i, \tag{5}$$

where P_i is the sum of the pressure of all thermal ion species, T_i the ion temperature (all ion species are assumed to be at the same temperature), q_i the heat flux through all ions and Γ_i the total ion flux. Those fluxes are calculated by the transport modules as described below. The source term Q_i is the sum of the following contributions:

$$Q_i = Q_{ei} - Q_{neo} + Q_{i,lh} + Q_{i,nbi} + Q_{i,icrh} + Q_{i,ecrh} + Q_{i,n0} + Q_{i,ext} + Q_{i,fus} + Q_{i,rip}, \tag{6}$$

respectively, electron–ion collisional energy transfer, neoclassical contribution (opposite sign w.r.t. electron heat equation), LH, NBI, ICRH, ECRH, charge exchange, optional additional source 'EXT', fusion reactions, energy losses induced by toroidal magnetic field ripple. The source terms are calculated by the external source modules (see section 5).

The ion particle flux Γ_i is linked to Γ_e by means of equation (7), which guarantees electroneutrality in the assumption of diffusivity and convective velocity equal for all species (ions and electrons):

$$\Gamma_i = \alpha_e\Gamma_e - D_e n_e\frac{\partial\alpha_e}{\partial\rho},$$
$$\alpha_e = \frac{n_i}{n_e} = \frac{\sum_{j=\text{species}} n_j}{n_e}, \tag{7}$$

where n_i is the sum of the density of all ion species and D_e the electron diffusion coefficient (see below).

The ion heat flux can be calculated either as the sum of a diffusive and a convective terms:

$$q_i = -K_i\frac{\partial T_i}{\partial\rho} - P_i V_i^q$$

or as a full transport matrix formulation:

$$q_i = -K_i\frac{\partial T_i}{\partial\rho} - K_i^e\frac{T_i}{T_e}\frac{\partial T_e}{\partial\rho} - K_i^n\left(\frac{T_i}{n_e}\frac{\partial n_e}{\partial\rho} + \frac{T_i}{\alpha_e}\frac{\partial\alpha_e}{\partial\rho}\right) - K_i^E\frac{T_i E_\parallel}{\mu_0 B_p} - P_i V_i^q$$

or even

$$q_i = -K_i^p\frac{\partial P_i}{\partial\rho} - K_i^e\frac{T_i}{T_e}\frac{\partial T_e}{\partial\rho} - K_i^n\left(\frac{T_i}{n_e}\frac{\partial n_e}{\partial\rho} + \frac{T_i}{\alpha_e}\frac{\partial\alpha_e}{\partial\rho}\right) - K_i^E\frac{T_i E_\parallel}{\mu_0 B_p} - P_i V_i^q$$

Note that the convective term V_i^q is a pinch term (positive V_i^q means inward flux).

2.4. Particle transport equation

In its present status, CRONOS evolves only the electron density. The density of the various ion species is then calculated as prescribed by the 'impurity' module. Usually this module uses prescribed ratios between the various species, following the constraints of electroneutrality and Z_{eff}. More complex behaviour of some minority ions species can be introduced in the 'impurity' module; however, this is limited to cases where the dominant ion species has the same dynamics as the electron density. An important application is the description of helium ash in a burning plasma, which is handled by an independent transport equation directly in the 'impurity' module. Note that development is ongoing to allow solving transport equations for the density of the ion species (including various charge states), instead of the electron density, using an impurity transport module. Ambipolarity is assumed at all time scales considered by the code, which is consistent with the other approximations of the code. By default, the CRONOS data model usually contains five ion species, but this number can be freely adjusted by the user.

The electron transport equation solved by CRONOS is

$$\frac{\partial}{\partial t}\left(V' n_e\right) + \frac{\partial}{\partial\rho}\left(V'\left\langle|\nabla\rho|^2\right\rangle\Gamma_e\right) = V' S_{ne}, \tag{8}$$

where n_e is the electron density.

The particle source term results from the following contributions:

$$S_{ne} = S_{ne,neutrals} + S_{ne,nbi} + S_{ne,ext} + S_{ne,ripple},$$

respectively, ionization source from neutrals (wall recycling, gas puffing, etc), NBI, optional additional source 'EXT', particle losses induced by toroidal magnetic field ripple. Pellet injection is treated separately as an 'event' that induces direct modification of the density profile (without going through the transport equation (8), see section 6).

The electron flux can be calculated either as the sum of a diffusive and a convective terms:

$$\Gamma_e = -D_e\frac{\partial n_e}{\partial\rho} - n_e V_e^\Gamma$$

or as a full transport matrix formulation:

$$\Gamma_e = -D_e\frac{\partial n_e}{\partial\rho} - \frac{n_e D_e^e}{T_e}\frac{\partial T_e}{\partial\rho} - \frac{n_e D_e^i}{T_i}\frac{\partial T_i}{\partial\rho} - n_e D_e^E\frac{E_\parallel}{\mu_0 B_p} - n_e V_e^\Gamma$$

Note that the convective term V_e^Γ is a pinch term (positive V_e^Γ means inward flux).

2.5. Momentum transport equation

The momentum transport equation is written in terms of the total toroidal momentum $\Re = \sum_k m_k n_k\langle R V_{k,\varphi}\rangle$, where the sum is over all plasma species (ions and electrons), m_k is the mass of species k, n_k the density, R the major radius and $V_{k,\varphi}$ the toroidal velocity. The notation $\langle\,\rangle$ indicates a magnetic

flux surface average as defined by equation (2). The transport equation is written as [4, 5]

$$\frac{\partial \Re}{\partial t} = -\left\langle \sum_k R\hat{\varphi}\cdot\nabla\cdot\pi_k \right\rangle + \left\langle \sum_k \boldsymbol{j_k}\cdot\nabla\psi \right\rangle + \left\langle \sum_k R\boldsymbol{F_k} \right\rangle + \left\langle \sum_k R\boldsymbol{W_k} \right\rangle + \left\langle \sum_k R\boldsymbol{M_k} \right\rangle, \quad (9)$$

where π_k is the viscosity tensor, j_k the current density, ψ the poloidal flux and the three last terms of the right-hand side represent, respectively, the sources related to collisions, wave-induced quasi-linear diffusion and matter injection.

For the first term of the right-hand side (involving the viscosity tensor), by analogy with the heat transport equations, a diffusive–convective form is chosen as closure, as suggested by quasi-linear theory [6]:

$$\left\langle \sum_k R\hat{\varphi}\cdot\nabla\cdot\pi_k \right\rangle = \frac{1}{V'}\frac{\partial}{\partial\rho}\left(V'\left\langle|\nabla\rho|\right\rangle^2\Omega_\varphi\right) \quad (10)$$

with the toroidal momentum flux given as

$$\Omega_\varphi = -K_\Re\frac{\partial\Re}{\partial\rho} - V_\Re\Re, \quad (11)$$

where $K_\Re$ is the toroidal momentum diffusivity and $V_\Re$ the toroidal momentum pinch.

The second rhs term of equation (9) corresponds to the radial currents that may arise from the heating schemes that create fast trapped ions (such as NBI, ICRH and fusion reactions). Indeed, their wide orbits tend to create charge separation in the plasma, and therefore induce radial back-currents by ambipolarity. Radial currents can also arise from ripple losses. All these radial currents cross the poloidal magnetic field induce a Laplace force in the toroidal direction, i.e. a source of toroidal momentum. All these contributions are calculated by the source modules (see section 5).

The third rhs term is the toroidal momentum source that arises from collisions, which includes friction on the cold neutrals at the plasma edge, momentum transport by trapped fast ion orbit effects (NBI, ICRH, fusion reactions, MHD redistribution of fast ions) and direct momentum source from parallel momentum injection (NBI mostly). All these contributions are calculated by the source modules (see section 5).

The fourth rhs term is the toroidal momentum source that arises from wave-induced quasi-linear diffusion of electrons (ECCD, LHCD, fast wave current drive). Since the electrons are much lighter than ions, the contribution of this term is negligible with respect to the others. However, the code structure allows the source modules to calculate this contribution.

The fifth rhs term is the toroidal momentum source that arises from direct thermal particle injection, such as tangential pellet injection. No model is available for such a contribution yet.

Equation (9) allows us to calculate the evolution of the total toroidal momentum $\Re$ of the plasma. Starting from this, we can also derive the toroidal velocity of the individual plasma species, which is the actual measurable quantity (by charge exchange spectroscopy for instance). This is of course important in order to compare the predicted velocity with the experiment. The expression of the radial electric field is [5]

$$E_r = \frac{|\nabla\rho|}{e_k n_k}\frac{\partial p_k}{\partial\rho} - V_{k,\theta}B_\varphi + V_{k,\varphi}B_\theta, \quad (12)$$

where e_k is the electric charge (signed) of the species k.

The poloidal and toroidal velocity of a given species can be written as

$$V_{k,\theta} = u_k B_\theta,$$

$$V_{k,\varphi} = w_k R + u_k\frac{F}{R}, \quad (13)$$

where F is the diamagnetic function $F = RB_\varphi$, u_k and w_k are constant on a given flux surface [5]. Replacing the expressions of the poloidal and toroidal velocity in (12), using $B_\theta = (1/R)(\partial\psi/\partial\rho)|\nabla\rho|$ and averaging over the flux surface yields

$$E_r = \left[\frac{1}{n_k e_k}\frac{\partial P_k}{\partial\rho} + \frac{\left(\left\langle RV_{k,\varphi}\right\rangle - u_k F\right)}{\left\langle R^2\right\rangle}\frac{\partial\psi}{\partial\rho}\right]\left\langle|\nabla\rho|\right\rangle. \quad (14)$$

The u_k component (poloidal velocity) is assumed to be purely neoclassical and is computed by the neoclassical module NCLASS. The force balance equation solved by NCLASS for calculating u_k involves forces due to fast ions, which should be provided by the source modules.

Multiplying (14) by $m_k n_k$ and summing (14) over the species yields

$$\left(\sum_k m_k n_k\right)E_r = \left[\sum_k \frac{m_k}{e_k}\frac{\partial P_k}{\partial\rho} + \frac{\Re}{\langle R^2\rangle}\frac{\partial\psi}{\partial\rho} - \sum_k m_k n_k u_k\frac{F}{\langle R^2\rangle}\frac{\partial\psi}{\partial\rho}\right]\langle|\nabla\rho|\rangle, \quad (15)$$

which allows, from the knowledge of $\Re$ and the u_k, to calculate E_r. Because several neoclassical quantities depend on E_r (such as the poloidal velocities u_k), a convergence loop is carried out to solve problem (15).

Finally, the knowledge of E_r allows us to calculate $\langle RV_{k,\varphi}\rangle$ from (14), then $V_{k,\varphi}$ from (13) and allows comparison with the experiment.

2.6. *Suprathermal particles*

CRONOS uses fluid transport equations which are derived for thermal plasma species. Nevertheless, fast particles are commonly present in tokamaks owing to external heating (ICRH, NBI, LHCD) and fusion reactions. In most discharges of present devices, the fast particle density fraction does not exceed a few percent, which allows neglecting it in the transport equations. Conversely, the fast particle pressure can reach larger levels (~30% in some present day NBI discharges), which can influence the plasma equilibrium as well as other parameters such as the MHD alpha. In CRONOS, fast particles are treated in detail in the source modules, although only information on the suprathermal pressure is provided to the main data structure.

The total pressure is calculated as:

$$P_{\rm tot} = P_{\rm e} + P_{\rm ion} + \sum_{\rm sources} P_{{\rm supra},\perp}$$

where $P_{{\rm supra},\perp}$ represents the perpendicular pressure due to suprathermal particles, as calculated by the various source modules. These modules also provide information on the parallel pressure, in view of future integration of equilibrium codes dealing with anisotropic pressure. At present, the equilibrium solver assumes isotropic pressure and takes into account the total perpendicular pressure $P_{\rm tot}$.

2.7. Additional transport equations

The structure of CRONOS already includes two additional transport equations identical to equations (7) and (8), that can be used by the user for any additional quantity. The structure of the sources and the transport matrix is the same as for the usual heat transport equations described above. A possible application of this is the implementation of a transport equation for the turbulence energy, coupled to the classical heat transport equations as proposed in [7].

2.8. Transport modules

In the CRONOS suite, the transport modules are specific modules that calculate the coefficients of the transport matrix. They all have the same input/output structure, beside the noticeable exception of the neoclassical module which calculates the neoclassical transport coefficients $K_i^{j,{\rm neo}}$. All coefficients K_i^j of the transport matrix are calculated as the sum of the neoclassical coefficient and of an anomalous transport coefficient:

$$K_i^j = K_i^{j,{\rm an}} + \mu_i^j K_i^{j,{\rm neo}}$$

where μ_i^j is a tunable multiplier in front of the neoclassical coefficient (it is 1 in theory and by default, however, it can be changed to allow some empirical tuning). The anomalous transport coefficients $K_i^{j,{\rm an}}$ are calculated by transport modules that can be chosen by the user. This one has also the possibility to set these anomalous transport coefficients to 0, or use prescribed values (time-dependent profiles) as given in the input simulation file.

In most common use, one transport module calculates all coefficients of the anomalous transport matrix. CRONOS also offers the possibility to execute sequentially up to five anomalous transport coefficient modules. This allows us to have modules defining only a part of the transport matrix, or to apply further processing to the transport coefficients calculated by a previous module. After executing all the requested transport coefficient modules, the output coefficient matrix is passed to the transport equation solver, which updates the profiles for the next time step. A convergence procedure is carried out to solve for the non-linearity of the transport coefficients, which is described in detail in section 4.

Several widely used transport modules are integrated in CRONOS already, such as the GLF23 [8, 9], Weiland [10, 11], Bohm–gyroBohm [12, 13] and empirical CGM models [14]. NCLASS is used for the calculation of neoclassical coefficients [15]. Thanks to the modularity of the code, the user can also design and easily couple his own anomalous transport module to CRONOS. The implementation of widely used transport modules such as NCLASS and GLF23 has been checked and successfully compared with other integrated modelling codes such as JETTO and ASTRA [16] and TSC [17].

2.9. Boundary conditions

The transport equations described above are second-order differential equations in space and therefore need two boundary conditions. The first one is given by $(\partial A/\partial\rho)|_{x=0} = 0$, as imposed by the toroidal geometry (where A is the transported quantity). The second boundary condition is given at $x = 1$ and may be of various types: one can prescribe either the value of $A|_{x=1}$ (e.g. the value of electron pressure or temperature for the electron heat transport equation) or the flux of the transported quantity (equivalent to prescribing the value of $(\partial A/\partial\rho)|_{x=1}$). The second boundary condition for the poloidal flux equation is peculiar. Physically in a tokamak, the value of the poloidal flux is imposed from the poloidal field coils. In most cases, a real-time control scheme is used to provide a given plasma current value. In CRONOS, the user can choose between three ways to prescribe the boundary condition: the total plasma current $I_{\rm p}$ or the loop voltage $V_{\rm loop}$ or the edge poloidal flux $\psi|_{x=1}$. The most common and convenient option is to prescribe directly the plasma current, which is equivalent to impose $(\partial\Psi/\partial\rho)|_{t,x=1}$ following the relation:

$$I_{\rm p} = -\frac{1}{2\pi\mu_0}\left[V'\left\langle\frac{|\nabla\rho|^2}{R^2}\right\rangle\frac{\partial\Psi}{\partial\rho}\bigg|_t\right]_{x=1}.$$

Imposing the loop voltage is equivalent to prescribing $(\partial\psi/\partial t)|_{x=1}$. However, this mode is numerically less stable than the others and requires a smaller solver time step. It is therefore not recommended. The last mode (imposing directly $\psi|_{x=1}$) is the closest to real tokamak operation and is often used in combination with a controller to guarantee e.g. a given value of the plasma current.

3. Equilibrium (fixed- and free-boundary solvers)

The plasma equilibrium is a key physical object since it is involved in the transport equations (geometrical elements) and in all the modules that need the topology of the magnetic field. The equilibrium is supposed to be axisymmetric (no dependence on the toroidal angle), thus the equilibrium data structure in CRONOS is bi-dimensional. The 2D equilibrium is used by most of the heating/current drive modules and for synthesized diagnostics.

The transport equations and equilibrium are coupled in the usual quasi-static approximation: the time scale for perpendicular force balance is assumed to be much smaller than the radial transport time scales. This allows calculating static equilibria using the Grad–Shafranov equation at several time slices when solving the time-dependent transport equations. The frequency of the equilibrium calculations is defined by the user and should be of the order of 1 ms for cases with fast equilibrium variations as current ramps to 10 ms for more stationary cases.

The effect of flows is not included in the equilibrium modules presently coupled to CRONOS.

The usual equilibrium calculation mode in CRONOS uses a fixed-boundary solver, the plasma boundary being prescribed. In that case, the equilibrium solver acts as a standard CRONOS module which updates the equilibrium data structure. An alternative option for the equilibrium is to use a free-boundary solver, which is interesting for plasma shape control studies. For this purpose CRONOS has been coupled through Simulink [18] to the non-linear free-boundary evolution code DINA-CH [19, 20].

3.1. Fixed-boundary mode

In this mode, the equilibrium module calculates the geometry of the magnetic surfaces, as well as the average of several useful quantities along these magnetic surfaces, using the current and pressure profiles that are given by the transport equations as input. The last closed flux surface is usually taken as boundary condition, and is prescribed (typically from magnetic measurements in interpretative simulations). The plasma boundary is described either as a series of (R, Z) points or as a series of moments. It can therefore be up/down asymmetric and deal with the X point geometry.

One of CRONOS distinctive features is the close link between the transport equations and an elaborated fixed-boundary equilibrium solver in the standard calculation mode. The transport equations provide the total pressure (see section 2.6) and poloidal flux profiles as a function of the normalized toroidal flux coordinate x. The equilibrium solver needs as input the pressure and mean current density profiles as a function of the *poloidal* flux. For consistency between the transport equations and the equilibrium, a convergence loop on the mean current density profile is carried out for every call of an equilibrium calculation. The mean current density profile is defined as

$$J_{\text{mean}} = \frac{\langle \vec{J} \bullet \nabla\varphi \rangle}{\langle 1/R \rangle} = \frac{\langle J_\varphi / R \rangle}{\langle 1/R \rangle} = -\frac{\dfrac{\partial}{\partial x}[V'(\partial\psi/\partial x)\langle|\vec{\nabla}\rho|^2/R^2\rangle]}{\rho_m^2 \mu_0 V' \langle 1/R \rangle}.$$

From this equation it is clear that J_{mean} depends on the metrics of the equilibrium, even for a given $\partial\psi/\partial x$. This explains the need for the convergence loop: the equilibrium solver is called using as input the output $J_{\text{mean}}(\psi)$ profile from the previous calculation until it converges (difference between two subsequent J_{mean} profiles below a given value). In this process, only the metrics that will be used for further time evolution of the transport equations are affected. The pressure $P(x)$ and poloidal flux $\psi(x)$ profiles considered in the transport equations are not modified.

The default equilibrium solver in CRONOS is the HELENA code [21]. After solving the Grad–Shafranov equation, the HELENA version implemented in CRONOS calculates and stores in the equilibrium data structure a number of variables which are subsequently used by many of the CRONOS modules (including the transport equation solver): flux-surface averages (1D, see definition in section 2.1), geometry and 2D coordinates of the magnetic surfaces, 2D map of the magnetic field components and some MHD stability criteria ($n = \infty$ ballooning limit, Mercier criterion).

A less sophisticated equilibrium module can be used for faster calculations, based on an approximate solution of the Grad–Shafranov equation [22–24].

3.2. Free-boundary mode

CRONOS can be used in conjunction with DINA-CH for carrying out transport simulations with a non-linear free-boundary evolution code [19]. The coupling between the codes is done via Simulink (see section 11.8). The DINA-CH code includes a free-boundary equilibrium solver and a transport equations solver [20]. The coupling scheme is as follows: in addition to calculating the free-boundary equilibrium, DINA also solves the current diffusion equation, while CRONOS evolves the transport equations for the other fields and calculates all the source and neoclassical terms. Input references such as the desired plasma current and boundary shape (expressed either by moments or by 'gaps' i.e. characteristic distances to the vessel) are provided to a controller that calculates the supply voltages. This controller is implemented directly under Simulink using a block diagram. The electric circuits are described and calculated inside DINA-CH. A common time clock is provided by Simulink. For a typical ITER current ramp simulation, CRONOS and DINA-CH communicate every 1 ms.

4. The CRONOS workflow

The CRONOS code features a graphical interface that allows generating an input dataset from experimental data, editing it, then running the transport simulation, storing and visualizing the results. In this section, we describe the series of calculations that are carried out during the transport simulation itself (see figure 2).

The internal computing scheme of CRONOS is tuned using switches that allow choosing (i) which transport equations are solved and (ii) how the various terms entering the transport equations (transport coefficients, sources, equilibrium, boundary conditions) are evaluated. For instance, any of the above terms may be either (i) set to 0, (ii) read from the input dataset or (iii) calculated by a physics module. In the latter case, the user defines the name of the module to be used, the value of its specific internal parameters and how often this module is called. Most of the switches for tuning the workflow are time dependent, thus allowing changing the workflow as a function of time. These switches define in a unique way the workflow that will be used for a simulation. They are included in the data model and are stored with the physical quantities in the CRONOS input and output files.

The data which are read from the input dataset are defined on the 'storage time base', which has usually much larger time steps than the internal time stepping used by the transport solver. Thus some interpolation has to be done in order to define the values of those quantities at all time slices addressed by the transport equation solver. Input time traces are linearly interpolated, while the other quantities are taken from the closest previous storage step.

The computation time depends of course strongly on the complexity of the source and transport modules, their

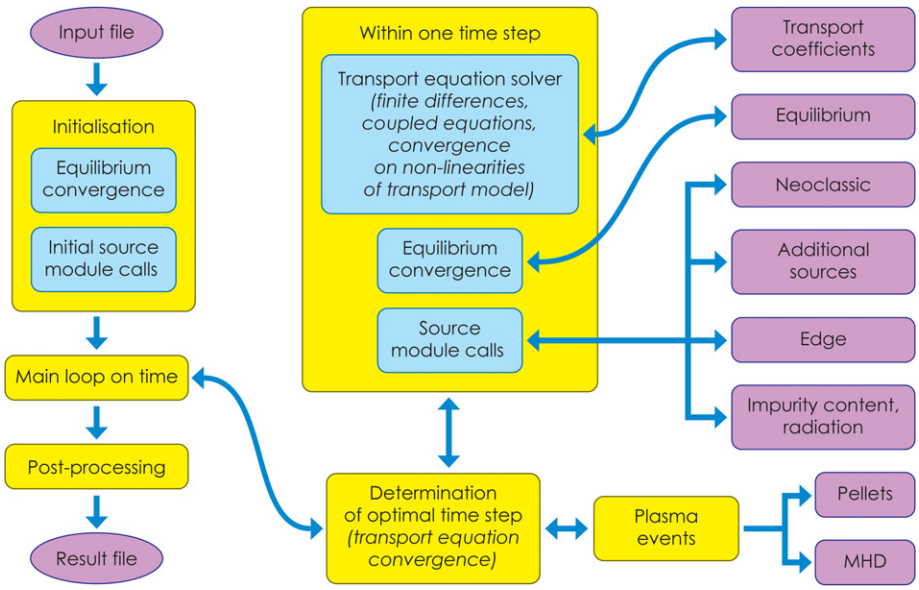

Figure 2. Overview of CRONOS workflow.

calling frequency, the equations solved and the numerical options used. To give some simple figures, a standard current diffusion simulation of a 30 s Tore Supra shot takes about 30 min on standard dual core PC with 2 Go of memory. Some sophisticated ITER simulations (current + heat transport with first principle based transport model), using ray tracing and Monte Carlo source modules, for 3000 s of discharge, take about one week on a small computer cluster with some source modules using up to 32 cores.

The CRONOS workflow starts with the initialization phase: the equilibrium is first computed, then the sources and transport coefficients, in such a way that a complete dataset is obtained for the first time slice. The first equilibrium is computed using the prescribed plasma boundary, the total pressure profile, plasma current and current density profile shape. A convergence is performed to determine a poloidal flux profile consistent with these constraints. This convergence process follows the same principle as the one done during the poloidal flux evolution (section 3.1), but modifies here the poloidal flux instead of the average current density (whose profile shape is prescribed).

After the initialization, the core of the CRONOS workflow is to loop on time solving the transport equations, i.e. computing the plasma state at time slice $k + 1$ from its state at time slice k. The internal time step used by the transport solver can either be prescribed by the user or automatically adjusted between two prescribed extrema. In the latter case, the automated adjustment of the time step is done depending on the convergence and the number of internal loops of the transport solver: if the transport solver does not converge on time slice $k + 1$, the result of the transport solver is discarded and the computation is tried again from time slice k with a smaller time step (generally the half of the first attempt). Conversely, if the convergence is successful, the result is stored and the calculation goes to the next step of the global loop on time. If the convergence was easy, the internal time step is increased for the next time slice calculation. This method has proven to be quite accurate, even when using stiff transport models (for the definition of stiffness, see [14]).

Changes in the plasma profiles that are calculated outside the transport equations (and thereafter named 'events', section 6) can interrupt the global loop on time. These are e.g. the injection of a pellet or the occurrence of an MHD crash (sawtooth, ELM). When such an event is going to happen, the transport solver time step is shorten in order to synchronize the solver time slice with the occurrence of the event.

For each step k of the global loop on time, a module named the 'one step solver' is called and carries out the following workflow:

(1) call the feedback control module that may update the simulation references (e.g. heating power, plasma current, pellet triggering, etc). Feedback controls are described in section 7.
(2) first estimate of fields variation (using adiabatic method, see below)
(3) transport solver: convergence loop to solve the set of coupled transport equations with consistency on fluxes (and optionally sources)
(4) equilibrium computation (with convergence on the current profile for consistency with the current diffusion). The common usage is not to do this step for each transport solver time step, a specific equilibrium calculation frequency being defined by the user (see section 3.1). For the highest precision, it is, however, possible to do it at each transport solver time step.
(5) calculation of source terms, ion species, MHD activity. Similarly to the equilibrium calculation, this step is usually not done for each transport solver time step. The user defines the time slices for the calculation of each source term independently.
    a. edge/SOL sources
    b. density of the various ion species and radiation losses (line, synchrotron, bremsstrahlung)

c. neoclassical quantities (resistivity, transport coefficients, etc)
d. magnetic ripple effects (modules available only for Tore Supra simulation)
e. auxiliary and fusion sources:
   i. ICRH/FWEH/FWCD
   ii. ECRH/ECCD
   iii. LHCD and (optional) common Fokker–Planck solver with ECRH
   iv NBI/NBCD
   v Fusion heating (DD and DT)
f. MHD stability calculations, which can be either trigger an event (modifies the plasma profiles outside the transport solver) or induce a 'continuous' effect (handled by the transport solver, e.g. increasing the transport coefficients, reducing current drive efficiency, etc)

Because the transport fluxes depend non-linearly on the plasma profiles, a particular attention must be dedicated to solve the transport equations properly. Indeed, a first evaluation of the transport fluxes yields a change in the plasma profiles, which induces in turn a change in the transport coefficients. This is particularly critical in the case of stiff transport models, for which small variation of the gradients of the input profiles induces a large variation of the output transport coefficients. A specific convergence procedure is used to solve the transport equations consistently, as detailed below (corresponds to steps 2 and 3 of the 'one step' workflow.

Step 2: first evaluation of fields variation. A first guess of the field variation from time t to time $t + \mathrm{d}t$ is obtained by one of the three methods available. The first one and most used just takes into account the adiabatic evolution under variation of the plasma geometry. The second one uses a fast evaluation in explicit mode of the transport equations without consistency nor internal convergence loop. The third is a mix of the two first methods. The use of the adiabatic evolution method increases the solver efficiency during transient phases. The adiabatic evolution is computed by prescribing a null partial time derivative for the left-hand side terms of the transport equations (1), (4), (5), (8)) which gives at the first order:

$$\Psi\left(t+\mathrm{d}t, x\right) = \Psi\left(t, x\right),$$

$$P_{\mathrm{e}}\left(t+\mathrm{d}t, x\right) = P_{\mathrm{e}}\left(t, x\right)\left(1-\frac{5}{3}\frac{\left(V'\left(t+\mathrm{d}t, x\right)-V'\left(t, x\right)\right)}{V'\left(t, x\right)}\right),$$

$$P_{\mathrm{i}}\left(t+\mathrm{d}t, x\right) = P_{\mathrm{i}}\left(t, x\right)\left(1-\frac{5}{3}\frac{\left(V'\left(t+\mathrm{d}t, x\right)-V'\left(t, x\right)\right)}{V'\left(t, x\right)}\right),$$

$$n_{\mathrm{e}}\left(t+\mathrm{d}t, x\right) = n_{\mathrm{e}}\left(t, x\right)\left(1-\frac{\left(V'\left(t+\mathrm{d}t, x\right)-V'\left(t, x\right)\right)}{V'\left(t, x\right)}\right).$$

Step 3: starting from this first estimate of the fields at $t + \mathrm{d}t$, the code starts solving the coupled set of transport equations. A convergence loop is used to enforce the consistency between the resulting fields and the fluxes. The transport coefficients, the transported quantities and the fluxes are evaluated self-consistently during this loop. As an option (option 5 in the list below), it is possible to evaluate also all source terms at each iteration of this loop, which is extremely costly in computation time but may be useful in the case of strongly varying source terms[7] The convergence loop is as follows:

(1) computation of edge/SOL sources and of field value at the last closed flux surface (LCFS, boundary condition).
(2) computation of the density of the various ion species and radiation losses (line, synchrotron, bremsstrahlung)
(3) computation of neoclassical quantities (resistivity, transport coefficients, etc). In the 'fast = 2' mode (faster calculation for current diffusion only simulations), this step is done just once.
(4) computation of magnetic ripple effects (if necessary, e.g. Tore Supra simulation)
(5) computation of auxiliary and fusion sources (optional)
(6) computation of anomalous transport coefficients
(7) stabilization of coefficients value between two calls (stabilized value = $\alpha \times$ old value + $(1-\alpha)\times$ new value, where α is a number between 0 and 1 and this value changes depending on the number of iterations in the convergence loop).
(8) build the transport matrix (containing the transport coefficients for the set of coupled transport equations).
(9) application of edge (boundary) conditions
(10) resolution of the transport equation for one internal time step. This uses a finite second-order difference scheme in mixed implicit/explicit mode (Crank–Nicholson like). This scheme is automatically turned into pure implicit mode in the case of slow convergence (when the solver time step has decreased down to its prescribed minimal value). The solver processes the complete set of coupled transport equations by inverting the whole transport coefficient matrix.
(11) update of the various transported fields with the result of the previous step.
(12) correction of possible unphysical results in the values of the fields (e.g. negative values forbidden, maintaining the values below a maximum and above a minimum).
(13) specific modifications of the poloidal flux by MHD (mean effect, applied instantaneously to the poloidal flux profile): (i) avoiding negative current values in current holes [25]; (ii) simulation of the mean sawteeth effect on the profiles (optional).
(14) time derivatives computation (causal on 2 points and regularized on 3 points)
(15) computation of equilibrium (optional)
(16) flux calculations (of the transported quantities)
(17) convergence condition evaluation: check that the difference on the transported quantities between the previous and the present iteration is small enough.
(18) loop break conditions: if the solver has converged (step 17) or if the maximum number of iterations has been reached or if numerical problems are detected (oscillations, exponential growth of the difference), this ends the convergence loop.

[7] With the option that evaluates all source terms at each step of the transport solver convergence, the simulation of the transient from fully ohmic plasma to a zero loop voltage plasma with a sophisticated source solver (C3PO/LUKE for LHCD, see section 5) takes 4 h for 1 ms of plasma (on a 4-cores computer).

Table 1. Main source modules available in CRONOS and the equations they contribute to (marked by an X).

Module name	Source Type	Electron/ion heat equation	Poloidal field equation	Toroidal momentum equation	Particles equation
PION	ICRF	X	X		
ABSOR	ICRF	X	X		
REMA	ECRF	X	X		
LUKE/C3PO	LHCD, ECRF	X	X		
DELPHINE	LHCD	X	X		
NEMO, SINBAD, SPOT	NBI	X	X	X	X
SPOT	Fusion	X	X	X	X
HPI2	Pellet	X			X
EXACTEC, CYTRAN	Cyclotron radiation source	X			
JONASSLICE	Neutrals	X			X

5. Source modules

5.1. General description

The source terms of the transport equations (section 3) can be either prescribed as time-dependent radial profiles or calculated by a 'source module'. Several source modules are embedded in CRONOS, from simple ones used for fast calculations to the most accurate and complex models. For each type of source (NBI, ICRF, LH, ECRF, pellets, fusion reactions of thermal ions, ripple effects, ionization from wall recycling/gas puffing, radiative processes), the user selects a module that calculates the associated source terms in the various transport equations (electron heat, ion heat, particles, poloidal flux, toroidal momentum), see table 1.

The source modules also calculate (when relevant) the neutron rate due to fusion reactions (either from thermal or superthermal ions), as well as the fast particle pressure (parallel and perpendicular to the magnetic field) due to the heating scheme. Those quantities are not directly involved in the transport equations, although the fast particle pressure contributes to the total pressure and thus enters the equilibrium calculation as an input. They are, however, very useful quantities for comparison of the simulation with an experiment and data consistency applications (neutron production, stored energy, etc).

The source modules used in CRONOS can be fairly sophisticated and may require a significant computation time. By default, they are run at every storage time slice of the simulation, but the user can choose to call them less often to save CPU time. The user then selects the time slices at which the source module should be run (or the time step between two runs), this is the 'complex' calculation mode, see section 4. Between two source module calculations, the user can specify to keep the value of the source term as in the previous calculation. Alternatively, he can use a simple interpolation rule, such as updating the source terms proportionally to the injected power, which allows taking into account in a simplified way variations of the plasma or control parameters without doing a full calculation.

Note also that most of the sophisticated source modules in CRONOS are 2D models (ρ, θ) or even 3D since some of them can account for the effect of the magnetic field ripple. They use the 2D geometry calculated by the equilibrium solver as input (see section 3). They produce the 1D transport equation source terms by surface averaging of their 2D output quantities.

In addition to sophisticated first-principle source models, the CRONOS suite features also simpler modules for fast calculation. The simplest of them prescribe a Gaussian source term with a given centre and width. The CRONOS suite features also a comprehensive set of 'intermediate complexity' modules, often using scaling expressions, which allow a coarse but very fast estimation of the source terms.

5.2. Brief description of the main source modules included in CRONOS

We describe here the most commonly used modules in CRONOS when a sophisticated evaluation of the source terms is needed. One of the strength of the CRONOS suite is that sophisticated first-principle modules are available for the calculation of most of the source terms that are involved in tokamak transport processes.

5.2.1. ICRF minority heating. The ion cyclotron radio-frequency heating (ICRH) in the minority heating scheme is usually simulated by the PION module. PION [26, 27] is a time dependent code which calculates the ICRF power deposition and the velocity distribution function(s) of the resonating ions. At the beginning of each time step the power deposition is calculated. The output is then used to calculate the time evolution of the distribution function(s) with a Fokker–Planck model. The output from the Fokker–Planck calculation at the end of the time step is used in the power deposition at the next time step, and the whole procedure is repeated until the end of the calculation.

In the power deposition calculation the launched wave is Fourier decomposed in the toroidal direction. The power deposition is then calculated for each toroidal mode number according to the model described in [28, 29]. This model has been partly obtained by analysing results from the full wave code LION [30]. Since the absorption strength depends on the distribution function of the resonating ions, the dielectric tensor components used in the power deposition calculation are updated, using results from the Fokker–Planck calculation, at the beginning of each time step according to the procedure described in [27]. The distribution function(s) of the resonating ions are calculated with a time dependent 1D Fokker–Planck equation. Effects due to finite orbit widths are taken into account by assuming that the fast ions have turning points close to the cyclotron resonance (i.e. where $\omega = n\omega_{ci}$) and then averaging the collision coefficients over the resulting

orbits; calculated quantities such as the energy density of the resonating ions and their collisional power transfer to the background ions and electrons are also redistributed along the orbits. Furthermore, the averaged square parallel velocity, which is used to determine the Doppler broadening of the cyclotron resonance in the power deposition calculation, is obtained from an *ad hoc* formula given in [26].

5.2.2. Fast wave electron heating (FWEH). The ABSOR module [31] simulates the FWEH for various antenna geometries (TS, JET, ITER) using a 1D fluid model accounting for the single pass scheme, where the wave power is shared between each toroidal number, assuming multiple reflexions at the plasma cuts (at high and low field side). ABSOR is not coupled with a Fokker–Planck calculation for minority schemes. It provides the heat power deposition but not the current drive. ABSOR is also able to simulate the absorption of ICRH power by minority species, although it was originally dedicated to the simulation of electron heating. In this case, the power is considered as completely absorbed at the resonance layer (considering a Gaussian deposition profile for the residual power with a width equal to the Doppler shift at the resonance layer), making the treatment by multiple reflexions unnecessary.

5.2.3. Electron cyclotron radio-frequency heating (ECRH). The ECRH propagation and absorption are simulated with the REMA module [32]. This is a ray-tracing code optimized for ECRH. REMA uses a linearized theory based on a perturbation method, assuming that the wave damping depends weakly on the shape of the electron perpendicular distribution function (i.e. the absorption is computed assuming a Maxwellian distribution function). The electron cyclotron (EC) propagation is computed via the ray-tracing method, computing ray trajectories in the toroidal geometry. Its absorption is derived from a fully relativistic dielectric tensor accounting for hot plasma effects. The EC heating source and current drive are analytically calculated; the driven current is deduced from a linear formula [33]. Based on linear theory, REMA does not account for synergy effects with lower hybrid (LH) waves or for strong quasi-linear effects. However, a synergy enhancement factor of the EC driven current can be introduced as an input parameter, as defined in [34].

The C3PO/LUKE code can also be used to simulate ECRH/ECCD, as well as synergistic effects with other electron waves such as LHCD (see paragraph below).

5.2.4. Lower hybrid RF waves. Two sophisticated modules for calculating the LH deposition and current drive are coupled to CRONOS: its historical solver DELPHINE and more recently C3PO/LUKE. Both codes are based on a ray tracing for the wave propagation and a Fokker–Planck solver for the absorption of the waves by the electrons.

In DELPHINE [35], the ray tracing uses the exact numerical equilibrium as computed by the CRONOS equilibrium module. The ray tracing is made in (R, Z, φ) coordinates and can include the effect of magnetic ripple by a perturbative approach. It is coupled to a relativistic Fokker–Planck solver (2D in momentum space) for the absorption on the electrons [36]. DELPHINE can also calculate the absorption of LH waves by fast ions (in particular fast fusion-born alpha particles), either using a prescribed fast ion distribution function [37] or by a full coupling to the Monte Carlo fast ion distribution solver SPOT [38].

The more recently developed module C3PO/LUKE [39, 40] has been designed to treat any RF electron waves (LH, EC, electron Bernstein waves). It allows evaluating possible synergistic effects in a consistent way, e.g. between the LH and EC waves. Both the ray-tracing C3PO and the 3D bounce-averaged relativistic Fokker–Planck solver LUKE are considering arbitrary axisymmetric magnetic equilibrium, using for this purpose the curvilinear coordinate system (psi, theta, phi), where psi is the poloidal magnetic flux. In ray-tracing calculations, the cold dielectric tensor is used, but warm and relativistic formulations may also be employed. For fast and very accurate evaluations of all derivatives in the ray tracing, the magnetic equilibrium is expressed in a vectorial form using a cubic spline technique combined with a standard Fourier transform. Once calculated, data are transfered to the Fokker–Planck solver and the quasi-linear absorption of the RF waves is determined by solving self-consistently the power transport equation along the rays with the build-up of the 2D fast electron distribution function in momentum space [41]. Since LUKE is a fully implicit 3D solver, 2D momentum and 1D radial dynamics are considered simultaneously in the calculations, which considerably reduces the computational effort, the steady-state solution being obtained after few matrix inversions. It is also possible to evaluate self-consistently the quasi-linear absorption in the presence of a possible anomalous radial transport, a unique feature of this code. The code LUKE is characterized by several important numerical improvements which turn out to be critical for the robustness of the current drive simulations. In particular, the new determination of the cross-derivatives which incorporate naturally the internal boundary conditions has led to use routinely high values of the quasi-linear diffusion coefficient without onset on numerical instabilities. The kinetic solver LUKE may also used for evaluating the runaways losses, including the effect of avalanches during a disruption phase when a large Ohmic electric field occurs. It is also able to calculate the fast electron losses in the magnetic ripple using a simplified model [42]. Finally, the output of the code can be used for determining the fast electron bremsstrahlung in the hard x-ray energy range with the synthesized diagnostic module R5-X2, an important feature for comparison with the experiment [43].

5.2.5. Neutral beam injection (NBI). Two main NBI modules are available in CRONOS, namely SINBAD [44] and an upgraded version of it, NEMO. Both can be used either with simplified solvers for the fast ion distribution function or the more accurate Orbit Following Monte Carlo code SPOT.

The first part of the SINBAD package simulates the injection, deposition and attenuation of the neutral beam inside the plasma. It uses the narrow-beam model, i.e. including an analytical calculation of the beam spatial distribution and a description of the 2D intensity distribution of the beam and of the particle trajectory (neglecting the diffusivity). In SINBAD, the ion source can be considered as point-like, the beam divergence is described by a Gaussian function, and the beam geometry is chosen by the user. Regarding the calculation

of the ion trajectory and slowing-down, SINBAD can run with an internal relaxation model, but this is rather inaccurate. However, it can instead be coupled with a 2D Fokker–Planck calculation. This second option is the one chosen in CRONOS. The usual Fokker–Planck calculation carried out in CRONOS is done by the NR-FPS code, consisting of a simple model using Stix collision equations and resolving the ion distribution function by expanding it in the eigenfunctions of the pitch angle scattering via Legendre polynomials [45]. SINBAD's main asset is its rapidity (much faster than a Monte Carlo calculation) and accuracy. However, compared with Monte Carlo codes, it has the disadvantage of being unable to model NBI transient regimes, and it does not account for fast ion orbit losses or magnetic field ripple.

This issue has recently been solved by coupling SINBAD with the SPOT Monte Carlo code, described in the next sub-section, thus providing CRONOS with the capability of taking into account accurately finite orbit width effects, magnetic field ripple and the effect of anomalous turbulent transport on the fast ion distribution.

Another module for NBI injection has recently been developed. This new code, called NEMO, is based on the same principles and approximations as SINBAD. It has actually been completely rewritten in Fortran 90 from the SINBAD original version, in order to take advantage of the Fortran 90 new features, allowing for easier exportability (via dynamic allocations and the use of structures/modules). This version has largely been lightened since most of the SINBAD original capabilities are useless in the context of CRONOS (e.g. the internal equilibrium calculation and the correction of orbit width effects have been removed). NEMO has been coupled with the fast Fokker calculation extracted from the METIS 0D suite, in order to get fast approximated calculations of the fast ion propagation and distribution, and with the SPOT Monte Carlo code for more accurate calculations. The geometry of ITER and JET NBI injectors are presently available in NEMO. The implementation of other tokamaks NBI injector geometry is planned to be added in the near future.

5.2.6. Sources associated with fast ions. The plasma fast ions, i.e. fusion-born alpha particles and fast ions coming from NBI, are simulated by the SPOT module [46]. This is a guiding centre Orbit Following Monte Carlo code, valid for arbitrary axisymmetric equilibrium. It provides all fast ion characteristics, such as their heat deposition on the bulk plasma and the current and pressure they induce. However, owing to its Monte Carlo nature, SPOT is quite time consuming when run on a single processor and has therefore been parallelized.

SPOT includes operators describing the collisions of fast ions with the bulk plasma (both thermal electrons and ions), based on the Stix formalism [47] including slowing-down, pitch angle scattering and energy diffusion processes. Another operator allows for self-consistent simulations of the interaction between fast ions and LH waves, describing the wave-induced transport in both velocity and real space [48]. Other operators are included in order to describe the effect of magnetic field ripple (stochastic diffusion and ripple well losses are described using a simplified 2D model) and of fast ion anomalous transport (i.e. induced by interaction with plasma instabilities). The latter is a radial transport operator simulating an *ad hoc* diffusion and convection of fast ions in order to quantify them by direct comparison with the experiment [49].

5.2.7. Cyclotron radiation. The cyclotron radiation source can be calculated by two different modules in CRONOS: CYTRAN or EXACTEC.

CYTRAN simulates the cyclotron radiation transport using a fast approximation making it suitable for the integration into a general transport code. The main assumptions for this approximation are the isotropy of the radiation field and the simplified treatment of the plasma either as a weak EC wave absorber or as optically thick (at low frequencies). It accounts for the non-local nature of the radiative process due to both re-absorption, reflectivity of the walls and polarization scrambling [50]. This method is adequate for non-homogeneous plasmas (e.g. with non-uniform spatial distribution of the EC radiative power density), contrary to global models. In CYTRAN, the EC absorption is calculated with approximate expressions for the absorption coefficients.

The EXACTEC module calculates the cyclotron radiation source and transport more accurately than CYTRAN, but requires more CPU time as well [51]. It is based on the exact solution of the radiative transfer equation for cylindrical plasma geometry with circular cross-section and specular wall reflection. It was upgraded for non-homogeneous magnetic field and elongated plasmas when coupled to CRONOS. It accounts for the non-local characteristics of this radiation, i.e. emission and re-absorption by the plasma, and includes wall reflections. The absorption coefficient calculation is done using simple but accurate relativistic formulae [52, 53].

5.2.8. Edge source. This module evaluates the sources arising from the ionization of edge neutrals (recycling + gas puffing). The main physics module for the edge source in CRONOS is JONASSLICE, originally a part of EDCOLL [54] which calculates the heat, matter and rotation sources of electrons due to the ionization of the cold neutral species coming from gas puff and wall recycling. Both inward and outward neutral fluxes are calculated. The radial profiles of the different neutral species, sources and sinks in front of each plasma facing component (PFC) are modelled from a set of 1D-slab calculations. These calculations can be repeated at different positions to model the 3D extent of the PFCs. Impurities are accounted for using the dilution of deuterium n_D/n_e and the perturbation of neutral pressure in the pumping system. The calculation is presently available only for a slab geometry, which limits its validity to situations where the neutral penetration is much smaller than the tokamak minor radius, which is the case in almost all present day experiments.

5.2.9. Pellet injection. Pellet injection is treated as an 'event' (see section 6), i.e. the pellet module provides CRONOS directly with the final state of the plasma profiles taking into account ablation, drift and homogenization. The pellet module in CRONOS is the HPI2 model, which is valid for any magnetic and plasma configuration, any hydrogen isotope pellets and any injection position. It computes the pellet ablation taking into account thermal ions and electrons and the suprathermal ions

generated by the plasma heating systems [55]. The drift model is based on the compensation of the cloud polarization by parallel currents [56, 57]. Additional effects as the pre-cooling of the plasma by the previously deposited material drifting in front of the pellet in the case of HFS injection and/or the pellet acceleration in the direction of the tokamak major radius due to the inhomogeneous shielding and the associated rocket effect [58] can be included optionally.

6. Instantaneous events: MHD activity, pellets

In CRONOS, pellet injection and MHD reconnection (such as a sawtooth or an ELM crash) are not described as time-dependent phenomena, since they induce fast variations of the profiles on time scales much shorter than the general transport processes. Therefore, they are treated as events which 'instantaneously' modify the plasma profiles outside the transport equations. Specific modules calculate the effects of pellet injection and MHD reconnection on the plasma profiles.

MHD reconnection is treated by two kinds of modules: first, a 'trigger' module checks at all time steps the stability condition of the MHD modes. When a mode is found to be unstable, the 'trigger' module flags the occurrence of an MHD event and a 'reconnection' module is called to compute the change in the plasma profiles. The new plasma state is added as a new storage time slice in the output data set and the global loop on time of the transport equations restarts from this state. There are three main sawtooth trigger modules in CRONOS, based, respectively, on (i) a list of prescribed time slices, (ii) the Porcelli trigger model [59] and (iii) a linear MHD stability calculation using the CASTOR code [60]. In the sawtooth reconnection module, standard models are implemented: the Kadomtsev model for full reconnection [61] or the Porcelli model for partial reconnection [58]. The final state is computed using MHD invariants and conserving main physics quantities (matter, energy, etc). An example of simulation with sawteeth in CRONOS and the full physics description can be found in [62].

The same organization is used for the pellet injection, but in this case the trigger is prescribed as a reference signal, either preset during the preparation of the CRONOS simulation or set on the fly by a feedback control scheme. Density control via pellet injection with adaptive frequency can thus be simulated. Details on the pellet module are given in section 5.2.9.

Note that the modules computing 'instantaneous' modifications of the plasma profiles can, if the physics process requires it, process the various ion species in independent ways, with the only constraint that the final plasma state calculated by the module has to fulfil electroneutrality.

7. Feedback control

An important feature of CRONOS is its ability to simulate feedback control experiments. In addition to the built-in CRONOS feedback control features, which are described in this section, CRONOS can also be used within Simulink, a Matlab toolbox dedicated to control (see section 11.8 for Simulink). CRONOS has a specific class of modules for feedback control. The user can choose the name of the module and the time at which it must be called (as for any source or transport coefficient module). The feedback control module acts by modifying the part of the data structure containing the tokamak reference values: power feeding the heating systems, gas puff rate, plasma current (or time derivative of the poloidal flux at the plasma boundary), target values for the plasma state (temperature at a given radius, minimum of the safety factor, etc). As input, the feedback control module receives a large part of the CRONOS data structure in order to know about the plasma state and to apply the corresponding feedback response. It receives also the reference values originally prescribed (feed-forward values) and their values at the previous call of the feedback module. In addition, a memory-persistent data structure is associated with the feedback module in order to perform e.g. integral or derivative controls. All this information allows a wide range of feedback controls to be simulated by CRONOS.

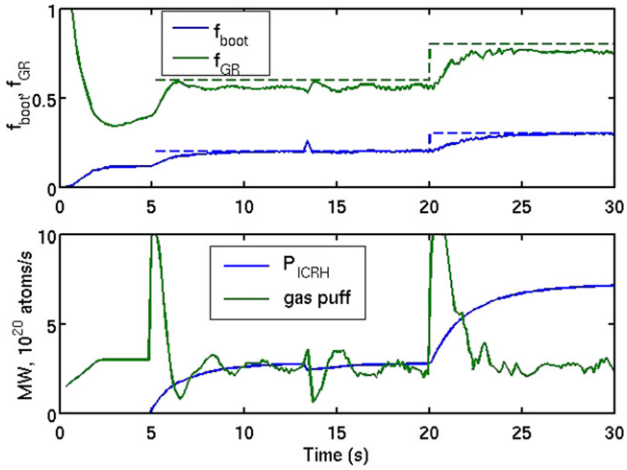

Figure 3. Current, heat and particle transport simulation with double PID feedback control of (i) the Greenwald density fraction via gas puffing and (ii) the bootstrap fraction via the ICRH power. Top: time traces of Greenwald density f_{GR} (green, solid) and its reference (green, dashed), bootstrap fraction f_{boot} (blue, solid) and its reference (blue, dashed). Bottom: time traces of the ICRH power (blue) and gas puff (green). The control starts at time $t = 5$ s. $5 < t < 20$ s: feedback control with reference $f_{boot} = 20\%$, $f_{GR} = 60\%$. $t > 20$ s: feedback control with reference $f_{boot} = 30\%$, $f_{GR} = 80\%$. When the reference f_{GR} is reached steadily, the gas injection becomes stabilized around the pumping speed (2.5×10^{20} atoms s^{-1}).

Some classical control schemes are provided in the CRONOS suite, such as proportional integral derivative (PID) controllers. This has been used for Tore Supra simulations in which the LH power has been controlled to sustain a given target plasma current at zero loop voltage [63]. Proportional controllers have been used to simulate DIII-D current profile control experiments [64]. Figure 3 shows an example of a double simultaneous proportional control of the Greenwald density and bootstrap current fractions, in a Tore Supra like discharge. More original control techniques such as the 'search, optimization and control' (SOC) have also been implemented in CRONOS and applied to Tore Supra current profile control experiments [65]. The same control technique, extended to five actuators and including MHD stability criteria (e.g. $\beta_N < 4l_i$ for instance), has been applied to design studies for the ITER steady-state scenario, as shown in figure 4. In this example, the control scheme aims at maximizing the

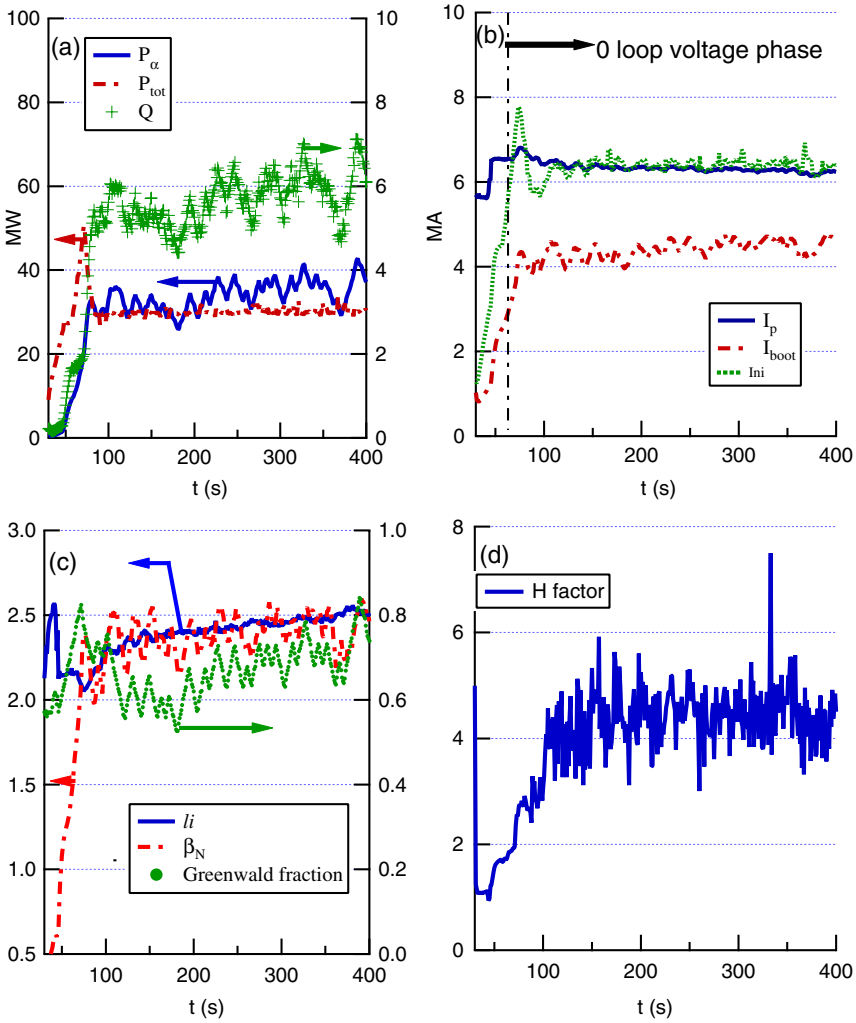

Figure 4. Simulation of a steady-state scenario for ITER using the SOC algorithm: the control scheme aims at maximizing the fusion Q ratio with four actuators (NBI and ICRH power, plasma current, electron density). (*a*) α heating power P_α, total injected power $P_{\rm tot}$ and fusion Q ratio (*b*) plasma current $I_{\rm p}$, non-inductive current $I_{\rm ni}$ and bootstrap current $I_{\rm boot}$ (*c*) $\beta_{\rm N}$, 4 $\times l_i$ and Greenwald fraction (*d*) H_{98} factor.

fusion Q ratio with four actuators (NBI and ICRH power, plasma current, electron density). Current and heat transport are simulated, using a scaling-based heat transport model with transport reduction based on magnetic shear to mimic the formation of an internal transport barrier. In the SOC technique, each actuator is optimized individually one after the other during the discharge evolution. The optimization procedure is the following: the value of the actuator is modified, then after some time the result on the target (Q factor) is measured. Depending on whether Q has increased or decreased, the actuator is changed in the same (respectively opposite) direction, until Q does not improve further. Then the algorithm applies the same procedure to the next actuator. The plasma current is left floating in the simulation, by imposing constant poloidal flux at the plasma boundary starting from $t = 60\,$s. In addition, the SOC optimization procedure is constrained by the ideal no-wall MHD limit $\beta_{\rm N} < 4l_{\rm i}$ (constraint on the total injected power).

As for the other types of modules (sources, transport coefficients, etc), new feedback controller modules can easily be designed and coupled to CRONOS by the user. New plasma target variables can also be added in the data model by the user and the code can be dynamically updated to handle them. All this allows a complete flexibility in the design of new feedback controllers.

8. Post-processing: synthesized diagnostics, stability analysis

Post-processing calculations can be made once the simulation has run. These calculations do not interact with the simulation itself but merely add new information based on its result. They include (i) calculation of diagnostic signals, as would be measured by a diagnostic viewing the plasma corresponding to the simulation result (synthesized diagnostic) and (ii) calculations that are done on selected time slices and do not need to be done during the run (e.g. checking *a posteriori* the MHD linear stability of a given time slice of the simulation result).

8.1. Synthesized diagnostics

The comparison of local measurements with the results of a simulation is rather straightforward: a map of the plasma

geometry is usually provided independently of the simulation, which allows us to convert the (R,Z) position of local measurements into one of the flux coordinates. Then the radial profiles predicted by the simulation can be directly compared with the experimental points. However, a direct comparison is not possible in several cases:

- diagnostics that integrate a quantity along a line-of-sight, such as interferometry or polarimetry
- diagnostics such as motional Stark exchange (MSE) spectroscopy that measure a typically 2D quantity (the orientation of the local magnetic field), which is not a direct result of the 1D transport equations
- measurements of complex quantities corresponding, for instance, to integrals of non-thermal distribution functions (such as the radiation emitted by a population of fast particles)

In these cases, some post-processing tools are needed to reconstruct the signal that would be measured by a diagnostic viewing the plasma corresponding to the simulation result. An essential element of this reconstruction is the 2D equilibrium that is solved during the simulation, which will link in a consistent way the simulation results to the diagnostic geometry. CRONOS includes such post-processing tools for interferometry, polarimetry and MSE spectroscopy (see example in section 10.1). The latter application is quite useful since usually MSE measurements are exploited only via an equilibrium identification that includes them as a constraint. In most cases, the physicist compares the q-profile resulting from this equilibrium reconstruction to the one of his simulation, without knowing precisely what is really constrained by the MSE measurement in the q-profile he looks at. Having a post-processing tool that reconstructs the MSE angles from the simulation results allows a direct comparison with the experimental data and a better understanding of the influence of the various physics parameters on the measurement.

Since CRONOS includes codes that model the distribution function of fast particles, post-processing tools can be coupled in order to reconstruct the signal of the related diagnostic. This has been done, for instance, for the interpretation of gamma ray spectroscopy measurements at JET, which aim at measuring the dynamics of fast alpha particles [46].

Finally, the equilibrium calculated by CRONOS can also be used to check *a posteriori* that the initial mapping of local measurements used to produce the input profiles is consistent. Indeed the CRONOS calculation takes into account much more information on the current and pressure profiles than what is usually available to the initial equilibrium identification code. The consistency of the mapping of local measurements can thus be checked, and can be improved if needed by rerunning the simulation after mapping the measurement on the equilibrium of the first simulation.

The availability of synthesized diagnostic modules contributes to make the CRONOS suite a very efficient and user-friendly tool for data validation and diagnostic design studies (see section 10.1).

8.2. Post-simulation analysis

Specific calculations can be done on selected time slices of the simulation results. CRONOS includes the MISHKA module for the calculation of linear MHD stability [66]. Note that this module can also be used during the simulation to trigger MHD events which have an impact on the evolution of the physics quantities.

Those post-processing tools are set up and run through the CRONOS interface, and their results are written in a specific post-processing data structure. Those results can be displayed using various tools included in the interface.

9. Link to tokamak databases

9.1. Reading input data from experimental databases

When doing simulations based on existing experiments, several plasma measurements are used to prepare the input simulation file. The access to experimental databases is handled by tools included in the CRONOS suite and usable through the CRONOS graphical user interface (GUI). The database access is, however, decoupled from the simulation itself: it is done entirely before the run. After database access, any signal can be edited and modified using the CRONOS GUI. The next step is to save the input simulation file on disk, which is then used to run the simulation.

The access to experimental databases is done via a tokamak dependent routine which maps the database variables onto the CRONOS data structure. For most of the tokamaks already coupled to CRONOS (JET, DIII-D, TCV, FTU, ITPA Profile Database [67]) database access is done via MDS+ [68]. Tore Supra data access is done using the local TSLib library. The same data access routines also tune default simulation parameters for each machine, usually to run a current diffusion simulation with consistent calculation of the source terms.

A key element in experimental data access is that measurements must be mapped onto an equilibrium, resulting in 1D profiles on a toroidal flux coordinate grid in order to be used as input by CRONOS. In some cases, this mapping is done prior to the use of CRONOS and the 1D profiles are already stored in the database (Tore Supra, DIII-D, TCV, ITPA Profile Database). When these mapped profiles are not present in the database (JET, FTU), machine-dependent tools are provided in the CRONOS suite to do the mapping and the fitting of the experimental data, reading equilibrium and diagnostic measurements directly in the database.

In the case of the ITPA Profile Database [67], CRONOS reads also the source terms from the database. This information is usually not present in the individual tokamak experimental databases, therefore it has to be calculated by the CRONOS modules. Thus, when coupling CRONOS to a new tokamak device, one must not only adapt the database access routine but also provide the various modules with the hardware description (heating systems, diagnostics, etc). During the simulation, the modules receive the information of the machine name so that they can use the relevant internal machine description.

Other information read from the experimental databases are:

- Plasma boundary information (either as a list of R,Z points or moments)
- Initial current and pressure profiles for the initialization of the equilibrium, initial profiles for the transport equations
- Reference values (heating power, plasma current, etc)

- Boundary condition for the transport equations are extracted from the related profiles (using their value at the plasma boundary)

9.2. Writing output data to fusion databases

The CRONOS output is an instance of the CRONOS data structure in Matlab format (see section 11). Although it is very convenient to use directly this output with the visualization tools of the CRONOS suite, the main results of the simulation can also be written to databases in order to allow a broader access to them. Routines exist that extract the most useful data from the CRONOS dataset and map them to the database format. These routines are provided:

- For the JET database: within the JET project of Integration of Transport and MHD codes, an interface has been developed to write CRONOS results in the same format as other transport codes (JETTO, ASTRA) in the JET database.
- For the ITPA Profile database: CRONOS can write its results in the ITPA Profile Database (0D, 1D, and 2D variables using the most recent variable definitions of the PDB). As an application, ITER simulations have been already submitted to the ITPA Profile DB [67].

10. Some applications of the CRONOS suite

The versatility of the CRONOS suite makes it suitable for a variety of applications in tokamak modelling. Owing to its close interfaces to experimental data and synthesized diagnostics, CRONOS is well suited for interpretative analysis and data validation purposes. On the other hand, CRONOS variety of modules and flexibility make it a useful tool for predictive simulations. Thus the CRONOS suite combines the two aspects (data validation/interpretative modelling and predictive simulations) at an equally high level, while most of the other existing transport codes are specialized in only one of these aspects.

This section gives a few typical examples of the various kinds of simulations that can be carried out from interpretative analysis of existing experiments to fully predictive reactor simulations.

10.1. Current diffusion and data validation

Simulation of current diffusion using prescribed density and temperatures is the simplest way of running the code and is routinely used for experimental analysis. It provides a prediction for the q-profile evolution, which is a quantity usually difficult to obtain with good accuracy from measurements. The non-inductive current drive sources can also be calculated consistently with the experimental temperature and density profiles. The results of the simulation are validated by comparison with global parameters such as loop voltage, internal inductance, as well as more complex quantities such as polarimetry and MSE angles, MHD phenomena. Figure 5 shows an example of such a comparison for a JET discharge, a quite good agreement is found between current diffusion with CRONOS and experimental measurements of the current profile [69]. Current diffusion simulations can also be the starting point for refined MHD analysis: as an example, CRONOS has been used for comprehensive studies of the MHD activity in the non-inductive regimes of Tore Supra [70–74]. Studies of turbulent transport on Tore Supra routinely use the current profile calculated by CRONOS as input [75, 76]. Current diffusion simulations are also quite useful for data validation since most available core plasma measurements (energy content, effective charge, kinetic profiles) are put together in such simulations, which is essential for checking the consistency of the dataset. This mode gives also access to the interpretative diffusion coefficients (heat, particle, etc), provided the related sources are being calculated by the code (see e.g. analysis of the electron heat diffusivity during giant oscillations [77]).

10.2. Predictive heat transport

This is the most commonly used mode of the code, where poloidal flux, electron and ion heat transport equations are solved together. The main purpose of this mode is to check the validity of transport models against an experiment. In [78], this is done in a sort of global way, i.e. the prediction of the electron temperature profiles is checked not only with respect to local measurements but also with respect to global and current profile measurements (data consistency) such as loop voltage, thermal energy content Faraday rotation angles (see figure 6). In [79], CRONOS is used both to investigate the properties of current diffusion and to demonstrate the existence of an internal transport barrier on the electron temperature linked to negative magnetic shear (see figure 7).

As an original example of predictive heat and current transport simulation, CRONOS has been used to simulate the dynamics of the electron temperature oscillation regime (O-regime) which has been discovered on Tore Supra. Specific modules have been designed for this purpose for transport coefficient and lower hybrid source terms. The implementation of these customized modules in the code has been straightforward thanks to the high modularity of CRONOS. This procedure allowed testing various assumptions and models on the nature of the processes underlying the O-regime. The up to now most successful simulation shown in [80], obtained by introducing strong magnetic shear dependence in the electron heat transport coefficient, has reproduced many features of the experimental q-profile dynamics that have been more recently diagnosed by the observation of MHD activity [81]. We show here an example of O-regime simulation carried out with an alternative transport model, featuring a reduction in the electron heat transport coefficient in a band around a given q-value [82]. This is intended to mimic the reduction in heat transport around a low order rational q-surface, although in this simulation we did not try to fit the oscillations to a particular q-value. The LH-driven current density and power deposition profiles are taken from the typical hard x-ray emission measured in Tore Supra steady-state discharges, i.e. slightly hollow with a maximum at $\rho = 0.2$. Outside $\rho = 0.3$, these profiles are kept constant throughout the simulation. Inside $\rho = 0.3$, the LH-driven current density j_{LH} is modulated proportionally to both the central temperature and current density ($j_{\mathrm{LH}} \propto T_{\mathrm{e0}} \cdot j_0$),

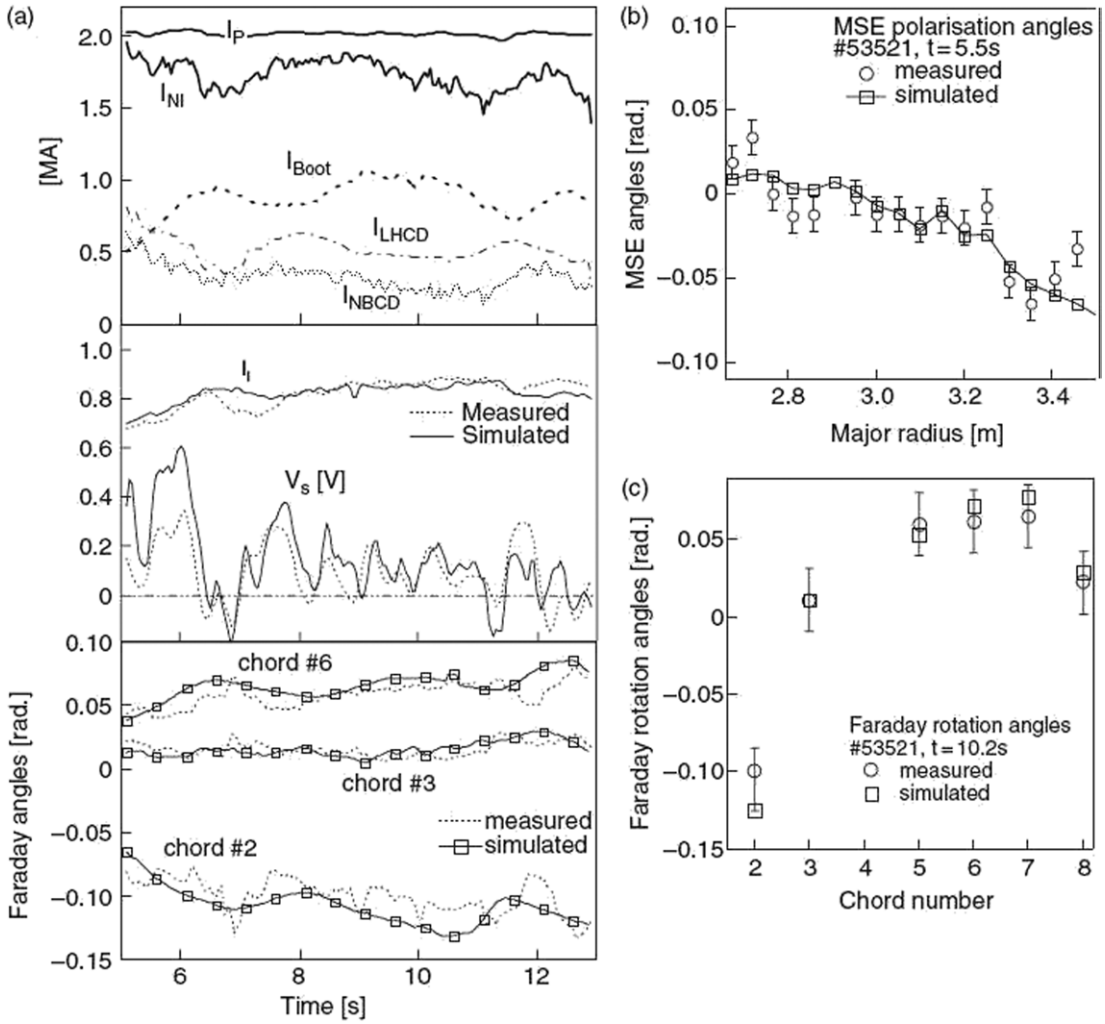

Figure 5. An example of current diffusion simulation of a JET discharge with CRONOS, with comparison to measurements [69].

and the LH power deposition p_{LH} to the central temperature only, in order to introduce a predator–prey type coupling mechanism. Temperature oscillations are hence obtained due to the following mechanism: as q goes up, the radial width of the good confinement zone increases, because the zero shear region lies below the enhanced confinement q sideband when the simulation starts. As a consequence, T_{e0} grows. Then, this increases j_{LH}, the current density also starts to increase and q decreases. Conversely, this reduces the width of the good confinement region, which causes T_{e0} to decrease, j_{LH} does the same and finally q will increase, starting the whole cycle again. In our simulation, we obtain periodic T_e oscillations with frequency 11 Hz and amplitude 130 eV, which is of the same order of magnitude as in the experiments (figure 8). However, there is no sign of propagation of the temperature modulation, since the profiles move as a whole inside $\rho = 0.3$. For the same reason, the magnetic shear is frozen. Therefore, the change in diffusivity is really due to the variation of the distance of the minimum q to a given (rational) value and reflects the topology aspects of the rational q theory fairly well [83]. Note also that this model provides periodic oscillations only if the minimum q is below the low order rational. This feature can be used as a test of the model relevance on experiment.

Another interesting application that highlights the tight coupling between the transport equations and the equilibrium solver in CRONOS has been reported in [84]. The simulations were done for the analysis of ICRH modulation in ITB experiments at JET. The motivation for this simulation was to check whether variations of the Shafranov shift of the plasma equilibrium, in phase with the heating power modulation, could induce apparent spurious effects on the amplitude of the temperature modulation as detected by the electron cyclotron emission diagnostic at fixed major radius. Indeed radial displacement of the plasma equilibrium may be wrongly interpreted as a temperature variation when measuring at fixed major radius, this effect being amplified at the position of the ITB where the temperature gradient is high. In order to quantify this effect, the equilibrium was calculated at a high rate in order to simulate with enough resolution the Shafranov shift modulation. The associated apparent temperature modulation amplitude could be calculated and led to the conclusion that the T_e amplitude peak detected at the position of the ITB was not due to this spurious effect.

Several more classical predictive heat transport simulations with CRONOS can be found in the literature, aiming at testing the accuracy of some transport models against

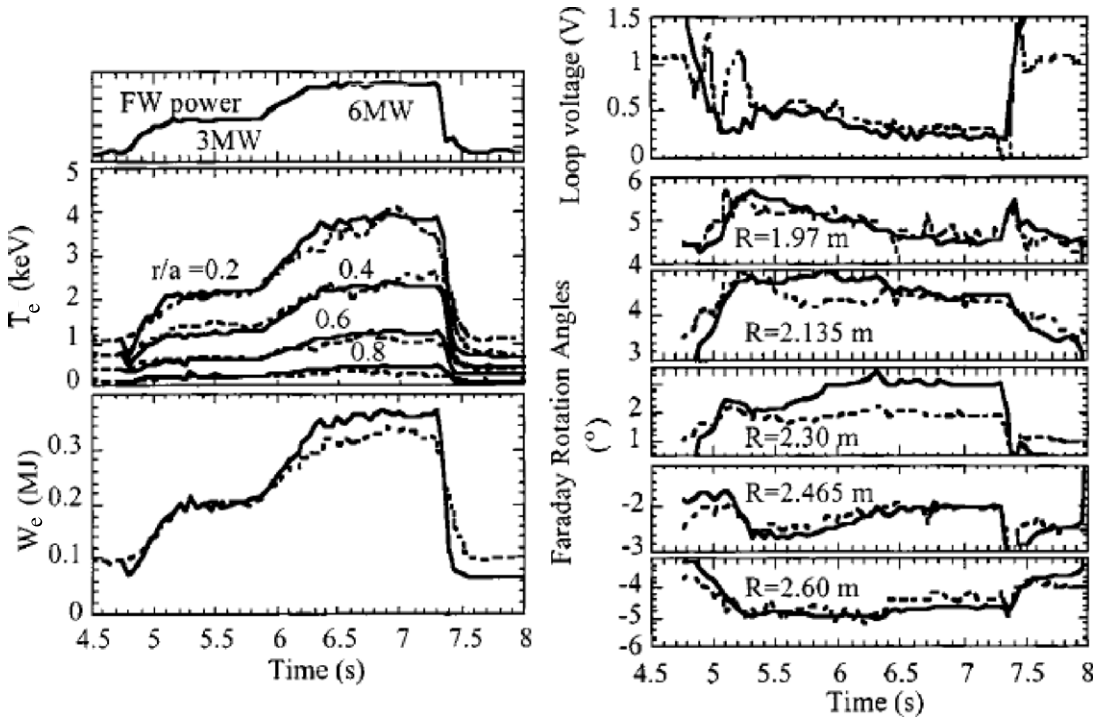

Figure 6. Predictive simulation of an existing Tore Supra FWEH discharge #18368, $I_p = 0.6$ MA, $B = 2.2$ T, $n_e(0) = 4 \times 10^{19}$ m^{-3}, with the fast wave power rising from 3 to 6 MW. Time evolution of electron temperature (T_e) at various radii, electron energy content (W_e), loop voltage and Faraday rotation angles. Solid lines: CRONOS simulations, predictive on current and electron temperature; dashed lines: experimental measurements. To be reprinted from figure 10 of [78].

experimental data in scenarios with internal transport barriers [85] and in hybrid regimes [86, 87]. Recently, CRONOS has been used intensively to benchmark heat transport models for current ramp-up phase against experimental data from JET, Tore Supra and ASDEX Upgrade and draw projections to ITER [16].

10.3. Integrated ITER or DEMO simulation

The CRONOS code has been used intensively during the past years for the design of ITER scenarios. Part of this activity has been conducted in an international context (e.g. the European Task Force on Integrated Tokamak Modelling, see [16], or the International Tokamak Physics Activities, see [17, 86, 88]). Owing to its large choice of sophisticated heating and current drive modules, the analysis has focused up to now on Advanced Tokamak Scenarios. Early examples of such simulations can be found in [88], showing the difficulty to sustain an ITB in the steady-state scenario. More recently, an original steady-state scenario has been proposed with different parameters than the ITER reference one, with no usage of NBI which central current drive is in this particular case found detrimental to the ITB sustainment. Localized ECCD is applied at mid-radius to form and sustain the ITB [89]. This original solution has also been extended to the steady-state operation of DEMO [90].

Coupled to the DINA-CH free-boundary equilibrium evolution code, the CRONOS code has been used to optimize the ITER current ramp-up with the injection of 20 MW LHCD [91]. The target q-profile could be tailored by means of LHCD (see figure 9) without exceeding the coil current and voltage limits in this self-consistent free-boundary equilibrium + current and heat transport simulation. By reducing the internal inductance, LHCD also makes the vertical position stabilization much easier (figure 10).

The ITER hybrid scenario has also been investigated, in conjunction with the analysis of present hybrid discharges in various experiments [86]. Benchmarking with other integrated modelling codes has been carried out based on hybrid scenario cases [17]. More recently, we have investigated the impact of the heating mix on the current profile, with the aim of (i) slowing down the occurrence of the $q = 1$ surface in the plasma and (ii) since it has not been possible to avoid it completely on a long time scale, reduce its radius in order to minimize the size of the sawteeth. This aspect is a key for the MHD stability of the hybrid scenario, allowing the avoidance of sawtooth-triggered deleterious neoclassical tearing mode. For this purpose, two different hybrid scenarios ($I_p = 12$ MA) have been studied: one with 20 MW of ICRH and 33 MW on NBI and another one in which the NBI system has been replaced by an ECRH/ECCD system with the same input power. The CRONOS modules used are PION for the ICRH, NEMO for NBI and REMA for ECRH/ECCD. In the case of cyclotron radiation, the EXACTEC code has been used since the electron temperatures expected in this type of scenario are high (>20 keV). The heat transport model applied inside the H mode pedestal is GLF23. However, a fixed level of diffusivity $\chi_e = \chi_i = 1\,\mathrm{m}^2\,\mathrm{s}^{-1}$ is imposed inside $\rho = 0.3$, since in this scenario GLF23 predicts no turbulent transport in this region which would trigger a strongly positive feedback loop with fusion power and alpha heating. The pedestal features are fixed to a height of 5 keV and a normalized width of 0.08. A fixed shape flat density profile is assumed.

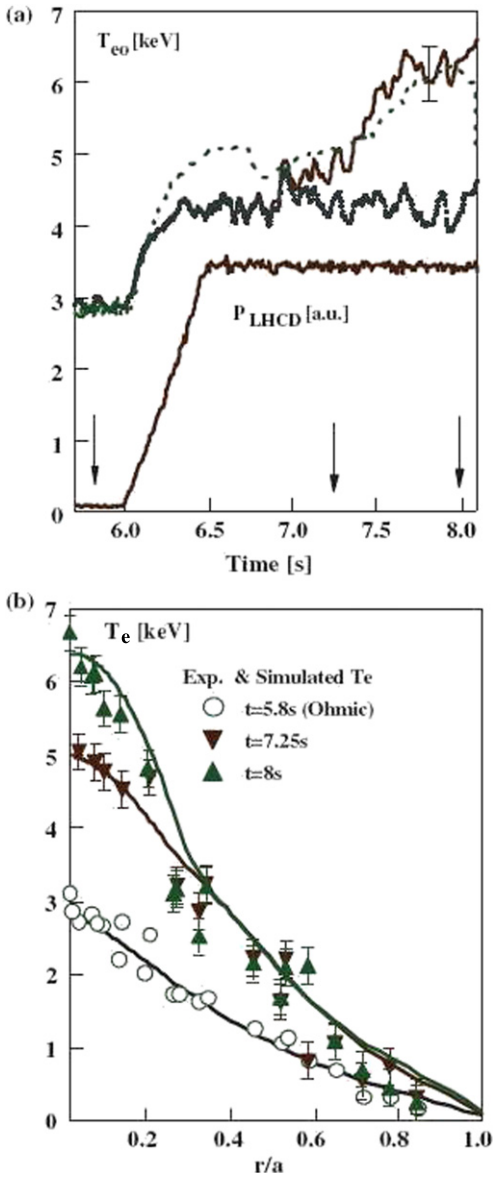

Figure 7. Predictive heat transport analysis of a Tore Supra discharge in which lower hybrid power is used to reach full non-inductive current drive. The Bohm–gyroBohm transport model is used. (*a*) Experimental (dashed line) and simulated (full line) time evolution of T_{e0}. The dotted line corresponds to a simulation performed without including the magnetic shear dependence in the transport model. (*b*) Experimental (markers) and simulated T_e profiles (full lines) [79].

The ECRH/ECCD system configuration, ray propagation and power deposition obtained with REMA are shown in figure 11. The ITER equatorial launchers are used with a wave toroidal angle propagation of $\phi_{tor} = 38^\circ$ for the three launchers. With this configuration, the power deposition obtained is well located between $\rho = 0.35$ and $\rho = 0.38$.

In figure 12, the distribution of currents at $t = 1200$ s, the q-profile for several time slices and the final temperature profiles are shown for both heating mixes. The current density profile is highly influenced by the current drive deposition from both NBI and ECCD. Conversely to the broad profile of NBI driven current, the ECCD deposition is highly localized around $\rho = 0.38$. Therefore, although the total ECCD current (1.1 MA) is half than that obtained with the NBI system (2.0 MA) the final q-profile is similar or even better since the region with $q < 1$ is smaller with ECCD. Regarding the temperatures, the change from NBI to a pure electron heating has an impact, mainly, for the ions, T_i dropping from 23 to 20.5 keV.

Since the possible operation space of ITER is extremely vast, the perspectives for ITER scenario studies with the CRONOS code are to progressively build a database of possible scenarios. The objective is (i) to study alternatives to the reference scenarios, in particular for Advanced Tokamak regimes with high non-inductive current fraction (ii) to include more and more complexity in the physics assumptions used.

11. Technical software aspects

11.1. Programming languages and general features

CRONOS is a modular and open source code running on Unix-like systems (mainly Linux). Its distribution is only constrained by the signature of a free of charge license agreement, which primarily prevents the multiplication of versions. CRONOS modules can be developed in MATLAB, C or C++ and Fortran 77 or 95 (other languages can be easily used). CRONOS is built around MATLAB, a numerical computing environment and programming language created by The MathWorks company. MATLAB allows easy matrix manipulation, visualization of functions and data, implementation of algorithms, creation of user interfaces and interfacing with programs in other languages. The logic underlying this choice for the CRONOS architecture is to benefit of the Matlab environment for data manipulation, matrix calculations (including the transport equation solver) and graphical interfaces, while most of the time consuming physics modules (equilibrium, source terms, etc) are written in C or Fortran. MATLAB also provides a high level of interactivity: simulation data can be manipulated and visualized directly by writing instructions to the Matlab command line. Another powerful feature is that the simulation can be run interactively, offering the possibility of real-time visualization of the results and interactive debugging. However, running the simulations is most frequently done in batch mode, since sophisticated Integrated Modelling requires usually a significant computation time. The CRONOS application is controllable and split in 'front end' (mainly the GUI) and 'back end' (mainly the computing code). The Fortran and C modules are compiled using a 'makefile' utility. The 'makefile' script uses a system-dependent configuration file which describes the compilers and environment that must be used.

CRONOS uses the MATLAB mexfile application program interface (API) or a dedicated generic API using files to call external modules written in various languages (C and C++, Fortran 77 and 95). Several physics modules in Fortran make use of scientific libraries such as NAG [http://www.nag.co.uk/], LAPACK and MUMS. The access to external data from various tokamaks is made through the TSLib library for Tore Supra and through MDS+ library for the others (JET, FTU, TCV, DIII-D, etc). Some special features of MATLAB are mandatory: the 'Signal Toolbox' is used in many part of CRONOS and SIMULINK is used for some feedback control applications (see section 11.8).

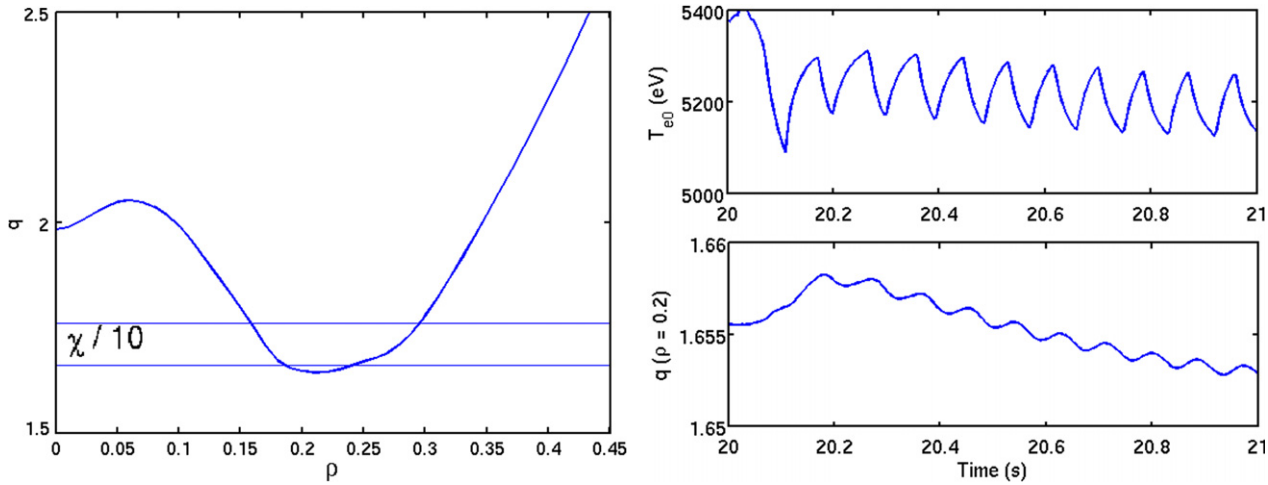

Figure 8. Predictive simulation of the O-regime involving $j_{LH} \propto T_{e0} \cdot j_0$ and a small q-region of enhanced confinement where the electron heat diffusivity χ is divided by 10. Left: zoom on the q-profile, the q-region with reduced χ is located inside the two horizontal thin lines. Right: time traces of central electron temperature T_{e0} (top) and safety factor q at $\rho = 0.2$ (bottom).

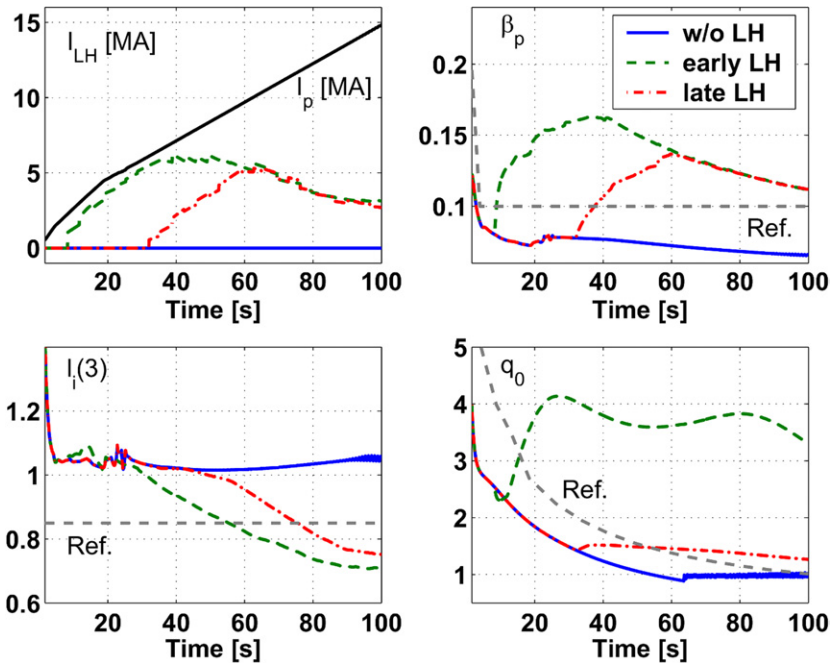

Figure 9. Coupled CRONOS/DINA-CH simulation of ITER current ramp-up assisted by LHCD. Time traces of LH-driven currents, poloidal plasma betas, internal plasma inductances and central safety factors. Three cases, without/early/late application of LHCD are compared. Reprinted from figure 2 of [91].

Thanks to these features, the CRONOS suite is easily portable on any Unix/Linux system and has been already installed on several fusion laboratories computers in Europe, USA, China and India. A website dedicated to the CRONOS collaborators (https://wikicronos.partenaires.cea.fr, password protected) gathers the CRONOS User's guide, technical notes, bug tracking system and a simulation database, thus offering a rich collaborative environment.

11.2. Data model and data files

The CRONOS data model describes all the data that are used and exchanged between CRONOS and its modules. This description includes the data type (numeric, string, etc), dimension (scalar, vector, matrix, multi-dimension array) and its definition. The data are made of Matlab structures. They are distributed in three main structures: one containing all the time dependent variables (including physical quantities and also time-dependent calculation flags, reference signals, etc), another contains variables that are not time dependent (e.g. module-specific parameters), while a third structure contains the data related to post-processing (diagnostic signal reconstruction, MHD or micro-turbulence stability analysis). These main structures are sub-divided into various sub-structures corresponding to physical objects which are calculated by certain types of codes. As an example, the first main structure (time-dependent variables) contains an equilibrium, a transport coefficient and source terms

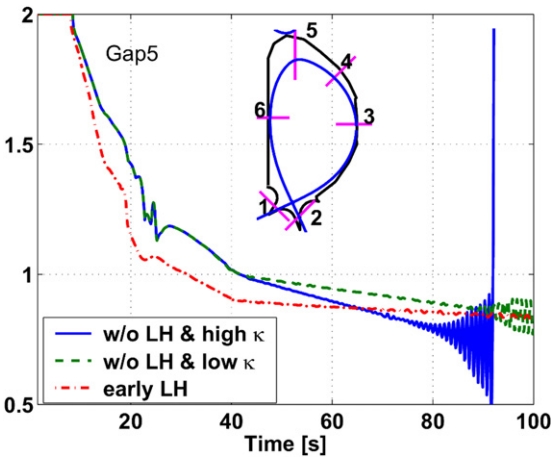

Figure 10. Coupled CRONOS/DINA-CH simulation of ITER current ramp-up assisted by LHCD. Time traces of gap5 indicated at the top of the vacuum vessel in the nested small figure. Reprinted from figure 8 of [91].

sub-structures (this list is not exhaustive). This is important for the communication between the core of the code and its modules: the input and output data structure of the modules depend only on the type of the module (e.g. equilibrium code or source term module). In other words, the I/O data structure is specific to the physics problem addressed by the module, not to the particular code used for the calculation. Having this standardized I/O data structure for a given physical problem is a key aspect of the modularity of the code, since any particular module can be used provided it adheres to this object- and physics-oriented I/O structure.

A CRONOS simulation file contains all the physical data (either input or output of the calculation) and also all parameters that are used for the run (calculation modes, module-specific parameters, reference values, etc). It is an instance of the complete data model. There is no explicit difference between data 'taken from experiment' and calculated results: the 'input' simulation file contains all input physical quantities and calculation parameters. Depending on these parameters, some physical quantities are predicted during the run, while the others remain unchanged. The 'output' simulation file contains exactly the same data structure as the input file (i.e. physical data + calculation parameters), with the physical data resulting from the calculation. The data are stored as MATLAB files.

All time dependent data are given on a prescribed and unique time base vector. The time base can be irregular (the time step can change from one time slice to the other). It corresponds of course only to a 'storage' time base for the input and output data, since the calculations usually use much smaller time steps. Note that the input and output data are stored on the same time base, with the exception that time steps can be added to the output data to describe specific events (MHD reconnection, pellet injection, etc).

The data can be loaded and edited in the Matlab workspace, either directly using Matlab instructions or through the GUI. Although CRONOS data are basically Matlab structures, one can use CRONOS without knowing anything of the Matlab programming language, thanks to the GUI.

11.3. Graphical user interface

The CRONOS GUI is built around a form generator and a few specific windows that allow editing wave forms, 2D data, to define specific time intervals and to edit internal time dependent configuration switches. All variables of the data model (i.e. both physical data and simulation parameters) can be edited by the GUI. The GUI is based on the Matlab GUI tools.

The CRONOS GUI allows us to read data from tokamak databases, handle CRONOS files, modify data, update data version, extend data set in time, open standard graphics, launch wallet programs (that made simple post-processing computation) and run, restart or post-process simulation (in interactive or in batch mode). The GUI includes a generic data viewer that is built dynamically from the data model. Pre-defined visualization programs allow an easy comparison of the simulation results with experimental data or between two CRONOS simulations.

Another powerful feature of the CRONOS GUI is the parametrization of the modules: for each physics problem (equilibrium, transport coefficient, etc) the user selects the module he wants to use. The window allowing tuning the module-specific parameters is then created dynamically from their declaration (written at the beginning of the module source code). Thus, when coupling a new module to CRONOS, one has simply to declare its specific parameters in a standard way and the associated GUI is generated dynamically from this information.

11.4. Dynamically generated code

The part of the code that manipulates data (like getting/putting a single time slice from/to the complete data set) is dynamically built from the data model. Thus, changes in the data structure are automatically reported in the code. As an example, a user can easily add a new variable to the data model by writing its one-line description in the data model file.

The code generator also makes data documentation and code automatic documentation in HTML for each Matlab 'mfile'. It also tests the syntax of the 'mfiles' (syntax error or inconsistencies or suspicious structure).

11.5. Storing results and debugging

Once the main loop (time evolution of the transport equations) reaches a storage time slice, the whole CRONOS data set for this time slice is stored on disk. These temporary files are then gathered at the end of the simulation to form the output file. The advantage of this system is to allow recovering results in the case of a crash during the simulation. The simulation can be then restarted from the last stored time slice. It is also possible to save the workspace of specific modules on disk in order to investigate the details of their calculation. For debugging purposes, it is possible to select some of the CRONOS variables to be stored in a file at each internal time step of the transport solver.

The Matlab environment of CRONOS allows tracking errors easily and offers a quite convenient interactive debug mode (in the Matlab-written modules only). The Matlab profiler can also be activated from CRONOS to perform a time consumption analysis and code optimization.

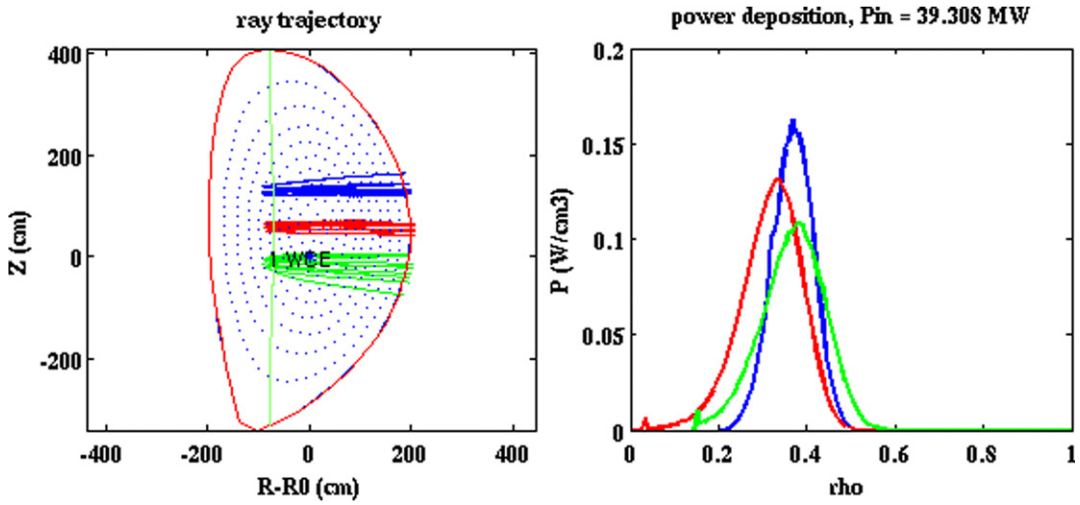

Figure 11. Ray-tracing simulation with REMA for the hybrid regime in ITER. The three equatorial launchers are used with 11 MW each. Left: ray propagation (one colour per launcher). Right: power deposition.

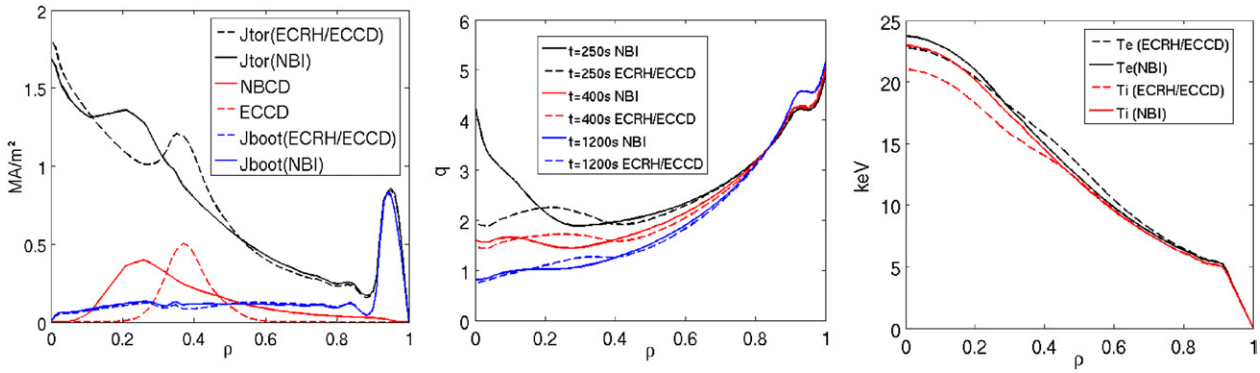

Figure 12. Total current (J_{tor}), neutral beam current drive (NBCD), electron cyclotron current drive (ECCD) and bootstrap current (J_{boot}) density profiles (left) q-profile for several times (centre). Electron (T_e) and ions (T_i) temperature profiles (right).

11.6. Coupling to external modules

The codes around the core transport solver that calculate the various terms involved in the transport equations (source term, geometry, transport coefficients, MHD impact on transport, etc) are called external modules. All these terms are calculated in a modularized way, i.e. the name of the module can be chosen by the user, who then tunes the model-specific parameters from a dynamically generated GUI. For doing this, the modules must start by a short standardized interface in Matlab language which defines:

(1) the list of module-specific parameters and their types (dynamically generated)
(2) the standard input/output format for an external module of a given type (e.g. source term).

The data transfer between the CRONOS workspace and the module can be carried out either using direct memory access or by writing temporary files on disk. The latter option uses a CRONOS toolbox which allows easy coupling with any external code.

The CRONOS data model contains the information on the characteristics of the modules which are used in a simulation: their name, the values of specific parameters (whose list is dynamically generated thanks to their declaration in the interface) and the list of time slices at which they should be called.

11.7. Test and certification tools—code version handling

The CRONOS source code is handled via a CVS server. This allows managing versions and subversions in a collaborative development environment.

The CRONOS suite includes tools that allow testing every module and the main CRONOS functions. The objective is to detect and avoid code regression as changes are made to the source code. It is also useful for checking the code after installing it to a new platform. The tests consist of compiling, then running parts of the code or even a full simulation and comparing the results with the expected ones on validated test cases. All this, including the analysis of the differences in the results, is done automatically by running a script from Matlab. Three levels of tests are available: individual modules, parts of the CRONOS workflow (initialization, advancing one time step in the transport solver, call to source modules) and full simulation.

The test scripts themselves are generated with a user-friendly interface from the results of an existing CRONOS simulation.

In order to release a new official version of the CRONOS suite, a complete set of test is run on various hardware and software architectures. The results must agree with the reference ones (within a tolerance of the order of the numerical accuracy of the computing platform).

11.8. CRONOS and Simulink

The CRONOS suite is linked to Simulink, an extension of the Matlab package dedicated to the simulation of feedback controllers. Although CRONOS can simulate feedback controls internally (see section 7), the coupling to Simulink allows a more general handling and implementation of feedback controls. On the Simulink side, the whole CRONOS suite is seen as an S-function that can import any CRONOS data in Simulink. On the CRONOS side, Simulink appears simply as a controller module which updates the reference time traces (e.g. heating power, gas puffing, etc) during the simulation. An important application of Simulink in CRONOS is the coupling with the DINA-CH code to carry out transport simulations with free-boundary equilibrium (see section 3.2).

12. Conclusions

The CRONOS suite has been developed since 1999 with the objective of achieving an integrated tokamak simulator with maximum modularity, flexibility in the type of simulations that can be carried out and user-friendliness with a powerful and extensive graphical interface. Original features like the automatic generation of graphical interface for new modules, the possibility for any user to add new variables and to couple new modules in a straightforward way are key elements which help in achieving these objectives.

During the past 10 years, the CRONOS code has been validated against both experiments and other codes of the same kind. It is now a mature code, able to cope with a variety of integrated simulations for present and future tokamak devices.

The versatility and adaptability of the code makes it a powerful core for further integration of more physics fields. As an example, a coupling to an impurity transport solver (dealing with multiple charge states of the ion species) is being developed and should be available in the near future. In addition, more sophisticated modules such as an ICRF full wave code [92], additional free-boundary equilibrium solvers and a quasi-linear transport model [93, 94] are being developed and integrated to CRONOS. In the longer term, more core-edge integration will also be required. Active development of the CRONOS suite is still going on and will continue in the coming years.

Although CRONOS is already a complete integrated modelling framework by itself, collaborations have been developed to link it with other integrated modelling projects. One of them is the Integration of Transport and MHD codes at JET. This project offers a common framework (named JAMS) for integrating various codes on the JET computer system, with a common interface to input and output data and core parametrization. The CRONOS suite has been adapted to work under this framework and can read the input data from it, write its results in the JET database using the common JAMS format, the simulations parameters being read from the JAMS graphical interface. Some common modules that are developed under this project (such as a common transport model access layer) can also be used by CRONOS and shared with other transport codes such as JETTO and ASTRA. The main advantage of the common framework is to provide users at JET with several modelling tools that can be handled in a very similar way. The communication between these modelling tools is also easier. Moreover it becomes much more convenient to compare modules in different integrated modelling suites since the input data come from the same environment. The success of the integration of CRONOS to the JAMS project illustrates the code adaptability and portability and makes it available to all JET users.

The CRONOS development effort is also beneficial to another major project, the European Integrated Tokamak Modelling Task Force [95]. This project has the quite ambitious goal of integrating all participating European modelling codes into a common framework, allowing extensive coupling and data sharing between modules. In addition to several important applications such as code benchmarking (versus other codes and experiments) and tackling new physics issues thanks to the coupling of sophisticated theoretical models, the ultimate aim of this project is to build a complete tokamak simulator. The contribution of CRONOS to this arising European effort will consist of not only providing its original physics models but also sharing the 10 years experience in code coupling, numerical techniques, code validation and benchmarking that has resulted from this endeavour.

Acknowledgments

This work, supported by the European Communities under the contract of Association between EURATOM and CEA, was carried out within the framework of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

The views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect the views of Fusion for Energy. Neither Fusion for Energy nor any person acting on behalf of Fusion for Energy is responsible for the use which might be made of the information in this publication.

References

[1] Gruber O. 2009 *Fusion Eng. Des.* **84** 170
[2] Hinton F.L. and Hazeltine R.D. 1976 *Rev. Mod. Phys.* **48** 239
[3] Blum J. 1989 *Numerical Simulation and Optimal Control in Plasma Physics* (Paris: Wiley/Gauthier Villars)
[4] Rozhansky V. and Tendler M. 1996 Plasma rotation in tokamaks *Rev. Plasma Phys.* **19** 147–249
[5] Helander P. and Sigmar D.J. 2002 *Collisional Transport in Magnetized Plasmas* (*Cambridge Monographs on Plasma Physics*) (Cambridge University Press) pp 246–8
[6] Becker G. 1990 *Nucl. Fusion* **30** 1610
[7] Sarazin Y., Garbet X., Ghendrih P. and Benkadda S. 2000 *Phys. Plasmas* **7** 1085
[8] Waltz R.E. *et al* 1997 *Phys. Plasmas* **4** 2482
[9] Kinsey J.E., Staebler G.M. and Waltz R.E. 2005 *Phys. Plasmas* **12** 052503

[10] Nordman H., Weiland J. and Jarmen A. 1990 *Nucl. Fusion* **30** 983
[11] Strand P. *et al* 1998 *Nucl. Fusion* **38** 545
[12] Erba M. *et al* 1997 *Plasma Phys. Control. Fusion* **39** 261
[13] Erba M., Aniel T., Basiuk V., Becoulet A. and Litaudon X. 1998 *Nucl. Fusion* **38** 1013
[14] Garbet X. *et al* 2004 *Plasma Phys. Control. Fusion* **46** 1351 with Addendum: 2005 *Plasma Phys. Control. Fusion* **47** 957
[15] Houlberg W.A. *et al* 1997 *Phys. Plasmas* **4** 3230
[16] Parail V. *et al* 2009 *Nucl. Fusion* **49** 075030
[17] Kessel C.E. *et al* 2007 *Nucl. Fusion* **47** 1274
[18] http://www.mathworks.com/products/simulink/
[19] Kim S.H., Artaud J.F., Basiuk V., Dokouka V., Khayrutdinov R.R., Lister J.B. and Lukash V.E. 2007 *Proc. 34th EPS Conf. on Plasma Physics and Controlled Fusion (Warsaw, Poland)* vol 31F (ECA) p P-5.142
[20] Favez J.-Y., Khayrutdinov R.R., Lister J.B. and Lukash V.E. 2002 *Plasma Phys. Control. Fusion* **44** 171
[21] Huysmans G.T.A., Goedbloed J.P. and Kerner W. 1991 *CP90 Conf. on Computational Physics, Amsterdam, The Netherlands* ed A. Tenner (Singapore: Word Scientific) p 371
[22] De Blank H.J. 2006 *Trans. Fusion Sci. Technol.* **49** 111–7
[23] De Blank H.J. Plasma equilibrium in tokamaks, lecture note @ www.rijnh.nl
[24] Lao L.L. *et al* 1981 *Phys. Fluids* **24** 1436
[25] Huysmans G. 2001 *Phys. Rev. Lett.* **87** 245002
[26] Eriksson L.-G., Hellsten T. and Willén U. 1993 *Nucl. Fusion* **33** 1037
[27] Eriksson L.-G. and Hellsten T. 1995 *Phys. Scr.* **55** 70
[28] Hellsten T. and Villard L. 1988 *Nucl. Fusion* **28** 285
[29] Hellsten T. and Eriksson L.-G. 1989 *Nucl. Fusion* **29** 2165
[30] Villard L. *et al* 1986 *Comput. Phys. Rep.* **4** 95
[31] Zerlauth P. *et al* 1997 *EURATOM-CEA Report EUR-CEA-FC*-1278
[32] Krivenski *et al* 1985 *Nucl. Fusion* **25** 127
[33] Cohen R.H. 1987 *Phys. Fluids* **30** 2442
[34] Dumont R. *et al* 2004 *Phys. Plasmas* **11** 3449
[35] Imbeaux F. and Peysson Y. 2005 *Plasma Phys. Control. Fusion* **47** 2041
[36] Peysson Y. and Shoucri M. 1998 *Comput. Phys. Commun.* **109** 55
[37] Imbeaux F. Peysson Y and Eriksson L.G. 2003 *Proc. 15th Top. Conf. on RF power in Plasmas (Moran, WY)* ed C. Forest (New York: AIP) p 271
[38] Schneider M., Eriksson L.-G., Imbeaux F. and Artaud J.F. 2009 *Nucl. Fusion* **49** 125005
[39] Peysson Y. and Decker J. 2008 *EURATOM-CEA Report EUR-CEA-FC*-1738
[40] Decker J. and Peysson Y. 2004 *EURATOM-CEA Report EUR-CEA-FC*-1736
[41] Peysson Y. and Decker J. 2007 *AIP Conf. Proc.* **933** 293
[42] Ju M. *et al* 2002 *Phys. Plasmas* **9** 4615
[43] Peysson Y. and Decker J. 2008 *Phys. Plasmas* **15** 092509
[44] Feng Y. *et al* 1995 *Comput. Phys. Commun.* **88** 161
[45] Wolle B. *et al* 1994 *Plasma Phys. Control. Fusion* **36** 1051
[46] Schneider M., Eriksson L.G., Basiuk V. and Imbeaux F. 2005 *Plasma Phys. Control. Fusion* **47** 2087
[47] Stix T.H. 1975 *Nucl. Fusion* **15** 737
[48] Eriksson L.-G. and Schneider M. 2005 *Phys. Plasmas* **12** 072524
[49] Schneider M., Eriksson L.G., Basiuk V. and Imbeaux F. 2006 Interaction between fusion-born alpha particles and lower hybrid waves including magnetic field ripple and anomalous transport effects in ITER *Proc. 33rd EPS Conf. on Plasma Physics (Roma, Italy)* vol 30I (ECA) p P-1.118
[50] Tamor S. 1981 *Report SAI-0.23-81-189-LJ/LAPS*-72 (La Jolla, CA: Science Applications)
[51] Albajar F. *et al* 2002 *Nucl. Fusion* **42** 670
[52] Bertelli N. *et al* 2005 *Phys. Lett.* A **347** 114
[53] Bertelli N. *et al* 2006 *Phys. Lett.* A **350** 431
[54] Azeroual A. *et al* 2000 *Nucl. Fusion* **40** 1651
[55] Pégourié B. *et al* 2005 *Plasma Phys. Control. Fusion* **47** 17
[56] Pégourié B., Waller V., Nehme H., Garzotti L. and Géraud A. 2007 *Nucl. Fusion* **47** 44
[57] Commaux N. *et al* 2010 *Nucl. Fusion* **50** 025011
[58] Kamelander G. *et al* 2008 *Proc. 35th EPS Conf. on Plasma Physics and Controlled Fusion (Hersonissos, Greece)* vol 32 (ECA) p P-4.104
[59] Porcelli F. *et al* 1996 *Plasma Phys. Control. Fusion* **38** 2163
[60] Kerner W. *et al* 1998 *J. Comput. Phys.* **142** 271
[61] Kadomtsev B.B. 1975 *Fiz. Plasmy* **1** 710
1975 *Sov. J. Plasma. Phys.* **1** 389
[62] Maget P. *et al* 2005 *Plasma Phys. Control. Fusion* **47** 357
[63] Basiuk V. *et al* 2003 *Nucl. Fusion* **43** 822
[64] Ferron J. *et al* 2006 *Proc. 21st Int. Conf. on Fusion Energy 2006 (Chengdu, China, 2006)* (Vienna: IAEA) CD-ROM file EX/P1-4 and http://www-naweb.iaea.org/napc/physics/FEC/FEC2006/html/node59.htm#15442
[65] Barana O. *et al* 2007 *Fusion Eng. Des.* **82** 1023
[66] Mikhailovskii A.B., Huysmans G.T.A., Kerner W. and Sharapov S.E. 1997 *Plasma Phys. Rep.* **23** 844
[67] Roach C.M. *et al* 2008 *Nucl. Fusion* **48** 125001
[68] MDSPlus home page at http://www.mdsplus.org
[69] Litaudon X. *et al* 2002 *Plasma Phys. Control. Fusion* **44** 1057
[70] Maget P. *et al* 2004 *Nucl. Fusion* **44** 443
[71] Maget P. *et al* 2005 *Nucl. Fusion* **45** 69
[72] Maget P. *et al* 2007 *Nucl. Fusion* **47** 233
[73] Maget P. *et al* 2007 *Phys. Plasmas* **14** 052509
[74] Maget P. *et al* 2009 *Plasma Phys. Control. Fusion* **51** 065005
[75] Parisot T. *et al* 2008 *Plasma Phys. Control. Fusion* **50** 055010
[76] Guirlet R. *et al* 2009 *Nucl. Fusion* **49** 055007
[77] Imbeaux F. *et al* 2006 *Phys. Rev. Lett.* **96** 045004
[78] Hoang G.T. *et al* 2003 *Phys. Plasmas* **10** 405
[79] Litaudon X. *et al* 2001 *Plasma Phys. Control. Fusion* **43** 677
[80] Giruzzi G. *et al* 2003 *Phys. Rev. Lett.* **91** 135001
[81] Maget P. *et al* 2006 *Nucl. Fusion* **46** 797
[82] Imbeaux F. *et al* 2004 *Proc. 20th Int. Conf. on Fusion Energy 2004 (Vilamoura, Portugal, 2004)* (Vienna: IAEA) CD-ROM file EX/P6-16 and http://www-naweb.iaea.org/napc/physics/fec/fec2004/datasets/index.html
[83] Garbet X. *et al* 2001 *Phys. Plasmas* **4** 1792
[84] Marinoni A. *et al* 2006 *Plasma Phys. Control. Fusion* **48** 1469
[85] Tala T. *et al* 2006 *Nucl. Fusion* **46** 548
[86] Imbeaux F. *et al* 2005 *Plasma Phys. Control. Fusion* **47** B179
[87] Kinsey J.E. *et al* 2005 *Nucl. Fusion* **45** 450
[88] Houlberg W.A. *et al* 2005 *Nucl. Fusion* **45** 1309
[89] Garcia J. *et al* 2008 *Phys. Rev. Lett.* **100** 255004
[90] Garcia J. *et al* 2008 *Nucl. Fusion* **48** 075007
[91] Kim S.H. *et al* 2009 *Plasma Phys. Control. Fusion* **51** 065020
[92] Dumont R.J. 2009 *Nucl. Fusion* **49** 075033
[93] Bourdelle C. *et al* 2007 *Phys. Plasmas* **14** 112501
[94] Casati A. *et al* 2009 *Nucl. Fusion* **49** 085012
[95] Bécoulet A. *et al* 2007 *Comput. Phys. Commun.* **177** 55