Plasma Convection in ITER Integrated Modelling with the Code TOKES

I.S. Landman, G. Janeschitz

Forschungszentrum Karlsruhe, IHM, FUSION, P.O. Box 3640, 76021 Karlsruhe, Germany

Abstract.

A two-dimensional simulation approach with arbitrary behaviour of magnetic flux coordinates uniformly covering the whole vessel developed for the code TOKES is applied to simulate plasma convection. The technique is based on a graph structure automatically generated by the code.

Integrated computer simulations of the future tokamak ITER should necessarily account for major consequences of unstable edge localized modes (ELMs). At each ELM, enhanced transport of deuterium-tritium plasma develops in the core and the pedestal, and along and across unclosed field lines ending at the walls. The resulting high energy loads lead to a contamination of the confined plasma by eroded wall materials beryllium, carbon and tungsten. Due to the influence of applied poloidal field coils, there are several regions in the vessel of monotonic decrease and increase of poloidal magnetic flux w.

To simulate those features the integrated two-dimensional tokamak code TOKES was designed in which the toroidal configuration of plasma and magnetic field are represented by dynamically calculated multi-mapping magnetic surfaces, and with the relations between the neighbouring regions of monotonic behaviour of w being described by ordering the mapped surfaces in a graph structure [1]. The code simulates with large time steps Δt multi-species MHD plasma diffusion and neutral rays from the wall penetrating the scrape-off layer (SOL) and the confinement region [2]. This work describes a special convection algorithm adjusted to the TOKES structures that was developed for mutual penetration of different species, plasma thermal transport and the transport of excited states of multi-charged tungsten ions.

In TOKES the different transport processes are split into independent algorithms. To simulate the convection, a split off hyperbolic equation for a function S is solved:

$$\frac{\partial}{\partial t}(GS) + \frac{\partial}{\partial w}(VS) = Q - PS \qquad (1)$$

The function $S(t,w) > 0$ is a number density or a thermal energy density, $G > 0$ the plasma volume per unit interval of the magnetic flux w, V the convection rate of substance across the surfaces, $Q \geq 0$ the substance source, and $P \geq 0$ the sink rate inversely proportional to the local lifetime τ : $P = G/\tau$. In the confinement region τ is infinite. The charged particles in

unclosed surfaces have diverse lifetimes, for instance for the trapped ions in the SOL τ is the time of scattering into the "loss cone".

To start with, a region of monotonic change of w is considered. The rates $V \neq 0$ are given at the numerical magnetic surfaces w_x ($w_{x+1} - w_x = \Delta w > 0$). In explicit numerical schemes Δt is limited ($0 < |V|\Delta t < G\Delta w$) otherwise the calculations become unstable [3]; in particular, S can get negative. The reason is that if V is too large the flow VS from a numerical layer (w_x, w_{x+1}) can empty it within one time step. However, usually some streams through many layers are formed in the substance, which can be used for improving stability.

To avoid $S < 0$, in case of $V > 0$ the following implicit scheme is used:

$$\frac{G_x S_x - \hat{G}_x \hat{S}_x}{\Delta t} + \frac{S_x V_{x+1} - S_{x-1} V_x}{\Delta w} = \hat{Q}_x - P_x S_x \tag{2}$$

The symbol "^" indicates the values before the time step, and S, G, Q and P are layer quantities. The solution follows with iterations starting from some x:

$$S_x = \frac{\hat{g}_x \hat{S}_x + S_{x-1} \upsilon_x + \hat{Q}_x}{g_x + \upsilon_{x+1} + P_x} \tag{3}$$

Here $g = G/\Delta t$ and $\upsilon = V/\Delta w$. A stream with $V < 0$ can be considered similarly.

A region with varying V may include several segments margined by zeroed V:
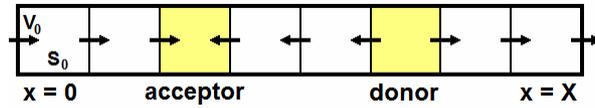
Let's consider a layer x with substance flow out of its borders: $\upsilon_x \leq 0$ and $\upsilon_{x+1} \geq 0$. The implicit scheme for this layer that provides $S > 0$ reads:

$$g_x S_x - \hat{g}_x \hat{S}_x + S_x \upsilon_{x+1} + S_x |\upsilon_x| = \hat{Q}_x - P_x S_x \tag{4}$$

Thus we don't need the values of S from the neighbour layers and obtain:

$$S_x = \frac{\hat{g}_x \hat{S}_x + \hat{Q}_x}{g_x + \upsilon_{x+1} + |\upsilon_x| + P_x} \tag{5}$$

Those "donor" layers allow for the iterations Eq.(3) in both directions over the adjoined segments of constant sign of υ. Calculating all streams we finally get the inflow into the "acceptors", that are the layers with inflow from both sides. For some acceptor x we have the known values $\upsilon_x \geq 0$, $\upsilon_{x+1} \leq 0$, $S_{x-1} > 0$ and $S_{x+1} > 0$. Finally, $S_x > 0$ of acceptor is obtained:

$$S_x = \frac{\hat{g}_x \hat{S}_x + S_{x+1} |\upsilon_{x+1}| + S_{x-1} \upsilon_x + \hat{Q}_x}{g_x + P_x} \tag{6}$$

As to the region edges, only the inflow must be given, and the outflow is calculated.
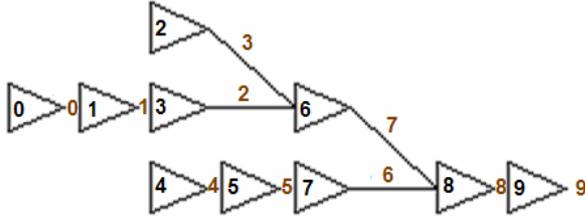
Fig. 1: A small graph. The triangles are layer images, with layer indices inside. The integers between the triangles are the indices of the convection rates V.

In a graph hierarchy the solution to Eq.(1) is constructed as follows. Fig. 1 shows a simplified graph (a detailed explanation on the graphs see in [1]). Let's start from the layer 9 (the graph origin) and come throughout the chain of single "sons" (for instance layer 8 is single "son" of layer 9). Those chains represent the regions of monotonic behaviour of *w*. We come to either the sonless layer, which reduces the problem to the already solved one using Eqs.(2) to (6), or to a multiple-son layer. In Fig. 1 we meet the latter case. The sons of the layer 8 can all be either the acceptors or other layers that need the inflow from the layer 8. That also would solve the problem for the chain, because the information from the sons is not required. If some son (e.g. layer 7) is a donor for the layer 8, the same procedure as for the origin layer is repeated for each such son in advance and the inflow from such a son becomes available, which thus also solves the chain problem. Eventually, after sequential excluding the solved out chains the entire convection problem can be solved.

The algorithm implemented in TOKES doesn't distinguish among the sonless-, singe-son- and multiple-son layers. They are incorporated in the calculations in a common way. The positive flow is assumed from the sons to the "dad" layer. Eq.(1) for a layer that has several (or no one) sons (subscript *s*) and one "dad" layer (subscript *d*) takes the form:

$$ gS - \hat{g}\hat{S} + \upsilon_d^+ S + \upsilon_d^- S_d - \sum_s \left(\upsilon_s^+ S_s + \upsilon_s^- S \right) = Q - PS \tag{7} $$

Here $\upsilon^+ = \max(\upsilon, 0)$ and $\upsilon^- = \min(\upsilon, 0)$. Thus in the pair terms of Eq.(7) one of two terms always equals zero. To solve the problem the code starts from the origin layer. Its dad absents: $\upsilon_d^+ = \upsilon_d^- = 0$ but the code interprets the dad as an existing acceptor. Then the code checks which interface with the sons has negative υ_s and which not. If some of the sons have $\upsilon_s > 0$, the code assumes the corresponding values of S_s known and obtains:

$$ S = \frac{\hat{g}\hat{S} + \left| \upsilon_d^- \right| S_d + \sum_s \upsilon_s^+ S_s + Q}{g + \upsilon_d^+ + \sum_s \left| \upsilon_s^- \right| + P} \tag{8} $$

The sum in the numerator includes only positive υ_s and the sum in the denominator only negative υ_s. Then the calculation is repeated quite similarly for the sons of $\upsilon_s \leq 0$. To obtain

the S_s of $\upsilon_s > 0$ the code solves initially Eq.(7) for the corresponding son-layers, which is performed as a recursive process. At this stage S is not needed, because those sons are the donors of their dad. Therefore Eqs.(7) and (8) are solely used for those sons being sure that the numerator term with S_d doesn't contribute.

The new convection algorithm of TOKES was tested by comparing its results with that of a standard algorithm for solving the linear equations Eq.(7) for a density function S connected according to the graph of Fig. 1, and the obtained results have coincided. As practical examples, Figs. 2 and 3 demonstrate the mutual penetration of DT plasma and carbon impurity as well as the plasma temperature at some time moment of ITER discharge. In those calculations (with $\Delta t = 10^{-2}$ s) the convection algorithm works together with the diffusion algorithm described in Ref. [1]. The carbon is produced at the divertor surface due to sputtering under the impacts of the hot ions lost from the confinement region.
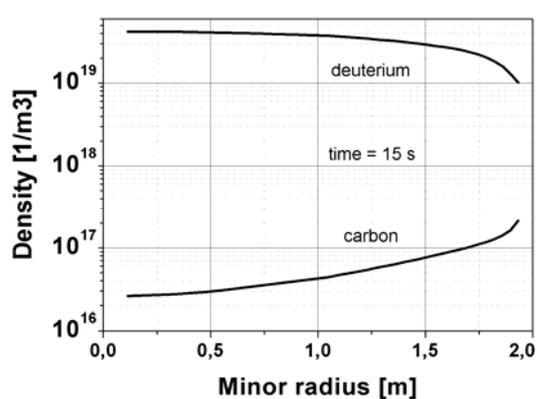
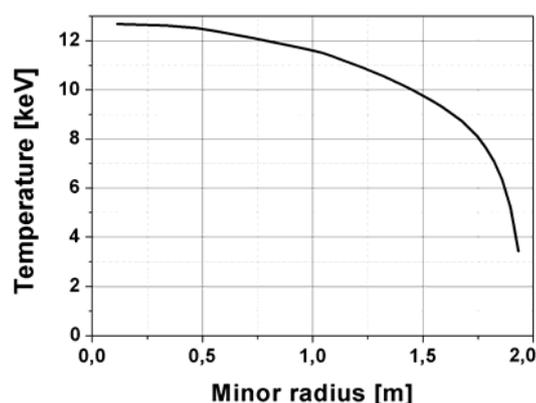*Fig. 2 Confined plasma densities (tritium density almost coincides with deuterium's)*

Fig. 3 Confined plasma temperature (the temperatures of all species are almost equal)

References: 1. I.S. Landman, G. Janeschitz, Proc. 33rd EPS Conf. on Plasma Physics, Roma, Italy, June 19-23, 2006, ECA Vol. 301, P5.165. 2. I.S. Landman, G. Janeschitz, J. Nucl. Mater. 363-365, (2007) 1061. 3. W.H. Press et al., "Numerical recipes in FORTRAN", 2nd edition, Cambridge University press, 1992, p. 820.