A Multi-Fluid Code for Simulation of the Edge Plasma in Tokamaks

B.J. Braams[1]

[1]FOM Instituut voor Plasmafysica, Postbus 1207, 3430 BE Nieuwegein, The Netherlands. Present address: Plasma Physics Laboratory, Princeton University.

Work performed under NET Contract Number 142/83–11/FU–NL/NET.

Abstract

A multi-fluid code for study of the two-dimensional structure of the tokamak edge plasma is presented. The code models an electrically neutral and current-free plasma containing several ion species and electrons. Each ion fluid is governed by a Navier-Stokes system of equations. Coupling between the plasma species occurs through ionization and recombination processes, interspecies friction, electric and thermal forces, and temperature equilibration. The code is specifically intended for study of the transport of helium and impurities through the high-density edge plasma envisaged for a tokamak reactor experiment.

1. Introduction

A detailed understanding of multi-species transport through the tokamak edge plasma is required in order to assess the pumping requirements for helium exhaust from a reactor experiment and to predict the penetration into the plasma core of impurities released from the wall and plates. To date, most of the work on modelling of helium and impurities in tokamaks has been concerned with the radial transport in the inner plasma. Issues of plasma–wall interaction and the associated impurity release can be modelled to some extent by using such radial transport codes, but only in so far as poloidal asymmetries and transport along magnetic field lines do not play a significant role. This excludes the study of transport through the edge plasma under conditions of high recycling on divertor plates or limiters. Multi-species plasma transport along the magnetic field in the tokamak boundary layer has been studied numerically by Neuhauser, Schneider et al. [1], and by Harbour and Morgan [2]. Their work demonstrated the importance of the longitudinal electric field and of thermal forces, which act differently on the various ionic species, for calculating the parallel transport of minority ions. However, they did not treat the radial transport.

The present paper describes a numerical model for a two-dimensional multi-fluid plasma. The code is specifically intended for study of the transport of helium and impurities through the high-density edge plasma envisaged for a tokamak reactor experiment. The equations used are of Navier-Stokes form for each ion fluid. The relative concentrations of the ion fluids are unrestricted. The ion fluids have a common temperature, which is distinct from the electron temperature. The electron density and velocity follow from the assumptions of charge neutrality and absence of electric current. Coupling between the plasma species occurs through ionization and recombination processes, interspecies friction, electric and thermal forces, and temperature equilibration. The numerical methods used here are well able to handle these coupling terms.

This code has been developed under a Euratom/NET contract. It supercedes the previous single-ion edge plasma code described in Refs. [3]–[7]. Both codes are described in Ref. [8]. This code and the previous one have been made available to the NET team through the group of M.F.A. Harrison at the UKAEA Culham Laboratory. The previous code has already been used extensively for modelling studies of the edge plasma in NET and INTOR; e.g. see Ref. [9]. The present code is organized in a very similar way, but it is more general in allowing several ionic species, and is somewhat more robust because of improvements in the numerical scheme. The multi-species capability has been implemented in such a way that there is no penalty when the code is used to

model a plasma that has just one ion species, and it is therefore recommended to use the present code in preference to the previous one in all cases.

Related two-dimensional edge plasma models have been developed at NRL [10], Princeton [11]–[13], Kurchatov [14]–[16], JAERI [17], [18], and JET [19], but all these models deal only with a single ion species. Although the need for a consistent two-dimensional transport model of a multi-species edge plasma has been felt for some years now, the code described here and in Ref. [8] seems to be the first actual implementation of such a model.

The outline of the report is as follows. Section 2 presents the mathematical model. In Sec. 3 the numerical approach is discussed, and in Sec. 4 two applications are treated. Conclusions are given in Sec. 5. Appendix A describes the organization of the code, and Appendix B describes in detail some routines that may have to be changed for different applications.

2. Mathematical Model

The purpose of this Section is to present the equations used to describe the two-dimensional multi-species edge plasma.

Outline. The present equations form a direct extension of the two-dimensional edge plasma model described in Refs. [3]–[7], which considers electrons and a single ionic species only. The equations are of Navier-Stokes form as regards the parallel flow and of a diffusive form in the radial direction. There are N ion fluids, which may have different velocities but have a common temperature. For each ion fluid α ($1 \leq \alpha \leq N$) there is a continuity equation governing the particle density n_α, a momentum equation governing the parallel velocity $u_{\parallel\alpha}$, and a diffusion equation governing the radial velocity v_α. The electron density and velocity follow from the assumptions of charge neutrality and absence of an electric current. The electron and ion temperatures T_e and T_i are governed by convection–conduction equations. The poloidal flow velocity u_θ is computed as $(B_\theta/B)u_{\parallel\alpha}$, therefore ignoring a diamagnetic contribution (B_θ and B are the poloidal and total field strengths, respectively).

Classical multi-species plasma transport theory [20]–[22] gives rise to a rather complicated system of force-friction relations, both for the parallel and for the radial transport. This theory has been used here as a guide to obtain a simplified set of equations for the parallel transport. These equations are consistent with the standard classical theory in the limit when one fluid is dominant and all others are trace impurities, and they remain mathematically sound also at finite relative concentrations. The radial transport is taken to be anomalous in our model. Interspecies friction and temperature equilibration are classical.

Our interest is restricted for the present to steady solutions of the equations. It is to be noted that, although the parallel transport coefficients are much larger than the perpendicular coefficients, also the parallel length scale is larger than the perpendicular scale, so the problem really is two-dimensional. The nonlinearity of the transport coefficients (e.g. heat conduction coefficients proportional to temperature to the power $5/2$, temperature equilibration coefficient proportional to temperature to the power $-3/2$) makes it difficult to predict whether any particular term will dominate any other, and even within one calculation the relative magnitude of the transport terms usually varies by several orders of magnitude over the geometrical domain. For cases of interest the flow velocity usually is subsonic, increasing to sonic velocity at an outflow boundary.

Equations. Specifically, we solve the following system of equations:

Continuity of species a $(1 \le a \le N)$:

$$\frac{\partial n_a}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x} n_a u_a\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y} n_a v_a\right) = S^a_n, \tag{1}$$

Momentum balance of species a $(1 \le a \le N)$:

$$\begin{aligned}
&\frac{\partial}{\partial t}(m_a n_a u_{ja}) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x} m_a n_a u_a u_{ja} - \frac{\sqrt{g}}{h_x^2}\eta_x^a \frac{\partial u_{ja}}{\partial x}\right) \\
&\quad + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y} m_a n_a v_a u_{ja} - \frac{\sqrt{g}}{h_y^2}\eta_y^a \frac{\partial u_{ja}}{\partial y}\right) \\
&\quad = \frac{B_\theta}{B}\frac{1}{h_x}\left[-\frac{\partial p_a}{\partial x} + \frac{Z_a n_a}{n_e}\frac{\partial p_e}{\partial x} + c_e\left(\frac{Z_a}{Z_{\mathrm{eff}}} - 1\right) Z_a n_a \frac{\partial T_e}{\partial x}\right. \\
&\qquad \left. + c_i\left(\frac{Z_a}{Z_{\mathrm{eff}}} - 1\right) Z_a n_a \frac{\partial T_i}{\partial x}\right] + \sum_{b=1}^{N} F_{ab} + S^a_{mu_j},
\end{aligned} \tag{2}$$

Diffusion of species a $(1 \le a \le N)$:

$$v_a = -\frac{D_n^a}{h_y}\frac{\partial}{\partial y}(\ln n_a) - \frac{D_p^a}{h_y}\frac{\partial}{\partial y}(\ln p_a), \tag{3}$$

Electron energy balance:

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\frac{3}{2} n_e T_e\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\frac{5}{2} n_e u_e T_e - \frac{\sqrt{g}}{h_x^2}\kappa_x^e \frac{\partial T_e}{\partial x}\right) \\
&\quad + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\frac{5}{2} n_e v_e T_e - \frac{\sqrt{g}}{h_y^2}\kappa_y^e \frac{\partial T_e}{\partial y}\right) \\
&\quad = \frac{u_e}{h_x}\frac{\partial p_e}{\partial x} + \frac{v_e}{h_y}\frac{\partial p_e}{\partial y} - k(T_e - T_i) + S^e_E,
\end{aligned} \tag{4}$$

Ion energy balance:

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\frac{3}{2} n_i T_i + \sum_a \frac{1}{2}\rho_a u_{ja}^2\right) \\
&\quad + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left[\frac{\sqrt{g}}{h_x}\left(\sum_a \frac{5}{2} n_a u_a T_i + \sum_a \frac{1}{2} m_a n_a u_a u_{ja}^2\right)\right. \\
&\qquad \left. - \frac{\sqrt{g}}{h_x^2}\left(\kappa_x^i \frac{\partial T_i}{\partial x} + \sum_a \frac{1}{2}\eta_x^a \frac{\partial u_{ja}^2}{\partial x}\right)\right] \\
&\quad + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left[\frac{\sqrt{g}}{h_y}\left(\sum_a \frac{5}{2} n_a v_a T_i + \sum_a \frac{1}{2} m_a n_a v_a u_{ja}^2\right)\right.
\end{aligned}$$

$$-\frac{\sqrt{g}}{h_y^2}\left(\kappa_y^i \frac{\partial T_i}{\partial y} + \sum_a \frac{1}{2}\eta_y^a \frac{\partial u_{\|a}^2}{\partial y}\right)\Bigg]$$

$$= -\frac{u_e}{h_x}\frac{\partial p_e}{\partial x} - \frac{v_e}{h_y}\frac{\partial p_e}{\partial y} + k(T_e - T_i) + S_E^i, \tag{5}$$

In these equations:

x, y = poloidal and radial coordinate,

$\sqrt{g}, h_x, h_y$ = metric coefficients,

B_θ, B = poloidal and total magnetic field,

Z_a, m_a = charge number and mass of an ion of species a,

$S_n^a, S_{mu_\|}^a$ = volume sources of ions and momentum for species a,

S_E^e, S_E^i = volume sources of electron and ion energy,

η_x^a, η_y^a = poloidal and radial viscosity coefficients for species a,

F_{ab} = friction force on ion species a due to species b,

c_e, c_i = coefficients in the thermal force for electrons and ions,

D_n^a, D_j^a = diffusion coefficients for species a,

$\kappa_x^{e,i}, \kappa_y^{e,i}$ = heat conduction coefficients,

k = energy equipartition coefficient.

Auxiliary quantities used here and elsewhere are: $n_i = \sum_a n_a$, $n_e = \sum_a Z_a n_a$, $\rho_a = m_a n_a$, $p_a = n_a T_i$, $p_e = n_e T_e$, $u_a = (B_\theta/B)u_{\|a}$, $u_e = (\sum_a Z_a n_a u_a)/n_e$, $v_e = (\sum_a Z_a n_a v_a)/n_e$, $Z_{\text{eff}} = (\sum_a Z_a^2 n_a)/(\sum_a Z_a n_a)$.

The coordinate system may be curvilinear although it must be orthogonal. By a proper choice of the local metric coefficients it is possible to represent the geometric effects associated with changes in the major radius and in the poloidal and toroidal field strength.

The source terms S_n^a, $S_{mu_\|}^a$, S_E^e, S_E^i in Eqs. (1)–(5) are associated with ionization, recombination and radiation, and have a complicated nonlinear and non-local dependence on the unknowns of the system. The model used to obtain the source terms is highly problem dependent.

Transport coefficients. The radial transport coefficients η_y^a, D_n^a, D_p^a, κ_y^e, and κ_y^i are anomalous and problem dependent. The poloidal transport coefficients are related to parallel coefficients according to $\eta_z^a = (B_\theta^2/B^2)\,\eta_\parallel^a$, $\kappa_z^e = (B_\theta^2/B^2)\,\kappa_\parallel^e$ and $\kappa_z^i = (B_\theta^2/B^2)\,\kappa_\parallel^i$. For the parallel electron heat conduction coefficient $\kappa_\parallel^e$ a flux-limited expression is employed. The coefficient is computed from the classical Spitzer–Härm coefficient $\kappa_{\parallel SH}^e$ according to the formula

$$\kappa_\parallel^e = \kappa_{\parallel SH}^e \left[1 + \left|\frac{q_{SH}}{q_{FL}}\right|^\gamma\right]^{-1/\gamma}, \tag{6}$$

where $q_{SH} = -\kappa_{\parallel SH}^e \partial T_e/\partial x$ is the classical conductive electron energy flux density, $q_{FL} = \alpha n_e T_e \sqrt{T_e/m_e}$ is the flux limit, and the parameters $\alpha = 0.12$ and $\gamma = 1$ were chosen in accordance with Refs. [23] and [24].

The equipartition coefficient k, parallel heat conduction coefficients $\kappa_{\parallel SH}^e$ and $\kappa_\parallel^i$, and the parallel viscosity coefficients $\eta_\parallel^a$ are computed using the formulae given in Ref. [22] for the case of a simple plasma (one ionic species), with the following replacements:

Equipartition coefficient:

Simple plasma: $k \propto Z_i^2 m_i^{-1} n_i n_e$

Multiple ion species: $k \propto \sum_a Z_a^2 m_a^{-1} n_a n_e$

Parallel Spitzer–Härm electron heat conduction coefficient:

Simple plasma: $\kappa_{\parallel SH}^e \propto Z_i^{-1}$

Multiple ion species: $\kappa_{\parallel SH}^e \propto \sum_a Z_a n_a / \sum_b Z_b^2 n_b$

Parallel ion heat conduction coefficient:

Simple plasma: $\kappa_\parallel^i \propto Z_i^{-4} m_i^{-1/2}$

Multiple ion species: $\kappa_\parallel^i \propto \sum_a Z_a^{-2} n_a / \sum_b Z_b^2 n_b \sqrt{2 m_a m_b/(m_a + m_b)}$

Parallel ion viscosity coefficient:

Simple plasma: $\eta_\parallel^i \propto Z_i^{-4} m_i^{1/2}$

Multiple ion species: $\eta_\parallel^a \propto Z_a^{-2} n_a / \sum_b Z_b^2 n_b \sqrt{2/(m_a + m_b)}$

These prescriptions were obtained by consideration of a simple random-walk model of the transport processes. Although this is a severe simplification of the complete multi-species transport theory, the expressions have the correct limit in the case when one species dominates, and they are considered adequate for the present application.

The friction force F_{ab} (proportional to $u_{ib} - u_{ia}$) is taken from Ref. [22]. The term $\mathbf{u} \cdot \nabla p_e$ in the energy equations represents work done by the electric field.

Boundary conditions. Counting derivatives, one sees that a total of $3N + 4$ conditions is required on the boundaries perpendicular to the x-coordinate, and that $4N + 4$ conditions are required on boundaries perpendicular to the y-coordinate. For the N momentum equations (2) and for each of the two energy equations (4) and (5), conditions must be given on each segment of the boundary. For the combination of the continuity equation (1) and the diffusion equation (3) for each species, one condition must be specified on the boundaries perpendicular to the x-coordinate, and two conditions on the boundaries perpendicular to the y-coordinate.

The boundary conditions appropriate to the two energy equations may specify either the energy fluxes or the temperatures, or more generally they may specify the energy fluxes as function of the temperatures and density. For the momentum equations we usually impose a sonic flow condition on one boundary segment perpendicular to the x-coordinate, and zero flow or zero shear elsewhere. For each combination of continuity equation and diffusion equation, on one face perpendicular to the x-direction and on both faces perpendicular to the y-direction, either a density, or a particle flux, or some relation between the two may be specified.

3. Numerical Solution

This Section gives a description of the discretization and solution methods employed for the system of equations (1)–(5).

Outline. The present code follows closely the approach that has been established in the author's earlier edge plasma modelling work and that that is described in Refs. [4] and [8]. Both codes are based on a finite-volume discretization of the conservation equations on a topologically rectangular mesh, using methods that were largely developed by D.B. Spalding and co-workers [25], [26]. The discrete coefficients depend continuously on the local cell Péclet number, and give central differencing and pure convective upwind differencing in the limits of small and large Péclet number, respectively. A nonlinear modification enhances stability in the presence of strong gradients. The discretization is fully implicit in time. A distributive relaxation method, leading to an elliptic equation, is employed to obtain the pressure correction at each iteration. The discretized equations are solved with the aid of Stone's "Strongly Implicit Procedure" [27], which is based on incomplete L*U decomposition.

In comparison with the previous code the principal difficulty in the numerical treatment of the system (1)–(5) lies in the strong frictional coupling between the ion flow velocities. A less severe problem is the possibility of strong coupling between the densities of ions of neighbouring charge states in situations where both ionization and recombination are important.

Spatial discretization. This subsection describes in general terms the finite-volume discretization scheme that is in employed in the code. The description follows Patankar [25]. We refer to a convection–conduction equation written in conservation form:

$$\nabla \cdot (\rho \mathbf{u} \phi - \Gamma \cdot \nabla \phi) = S. \tag{7}$$

It is assumed that Γ is diagonal when the equation is expanded on coordinates, so that

$$\nabla \cdot (\rho \mathbf{u} \phi - \Gamma \cdot \nabla \phi) \equiv \frac{1}{\sqrt{g}} \sum_{\alpha} \frac{\partial}{\partial x_\alpha} \left[\frac{\sqrt{g}}{h_\alpha} \left(\rho u_\alpha \phi - \frac{\gamma_\alpha}{h_\alpha} \frac{\partial \phi}{\partial x_\alpha} \right) \right]. \tag{8}$$

Let us consider the three-dimensional case. The region is divided into rectangular cells (control volumes), with ϕ discretized at cell centers and u at cell faces. For each interior mesh point P the differential equation (7) may be integrated over the control volume surrounding P. Let the neighbours of P be denoted by E, W, N, S, T, B (east, west,

north, south, top, bottom), and let the corresponding cell faces be denoted by subscripts e, w, n, s, t, b. The volume integral can be expressed as a sum of six surface integrals:

$$\iiint S\,dV = J_e - J_w + J_n - J_s + J_t - J_b,$$

where, e.g., for the 'east' face:

$$J_e = \iint \left(\rho u_1 \phi - \frac{\gamma_1}{h_1}\frac{\partial \phi}{\partial x_1}\right) h_2 h_3\, dx_2 dx_3 \tag{9}$$

In order to arrive at a discretization scheme of seven-point molecule form (five-point form in two dimensions) the expression (9) is to be approximated by

$$J_e \simeq a\phi_P - \beta\phi_E \tag{10}$$

and similarly for the other fluxes. The coefficients α and β will depend on approximations F_e and D_e to the strength of flow through the surface and the conductance between the mesh points: $F_e \simeq (\rho u_1)_e A_e$ and $D_e \simeq (\gamma_1)_e A_e/d_e$, in which A_e is the area of the 'east' cell face and d_e is the distance between the points P and E.

Two discretization schemes used often are the central difference scheme, which employs

$$\alpha = F_e/2 + D_e, \qquad \beta = -F_e/2 + D_e$$

and the upwind scheme, for which

$$\alpha = \max(F_e, 0) + D_e, \qquad \beta = \max(-F_e, 0) + D_e$$

As is well known, the central difference scheme is second-order accurate but unstable at high cell Péclet number, $P_e = F_e/D_e$, whereas the upwind scheme is always stable but is only first-order accurate.

Through consideration of the exact solution to the one-dimensional convection–conduction equation with constant coefficients Patankar [25] is led to define two intermediate schemes, both of which approximate central differencing at low cell Péclet number and upwind differencing for zero diffusion at high cell P. The general form of the coefficients in these and other schemes is

$$\alpha = \max(F_e, 0) + D'_e, \qquad \beta = \max(-F_e, 0) + D'_e, \tag{11}$$

where D'_e is an approximation to $D_e|P_e|/(\exp(|P_e|) - 1)$. For the piecewise linear scheme $D'_e = \max(0, D_e - |F_e|/2)$, and for the power law scheme $D'_e = D_e \max(0, (1 - |P_e|/10)^5)$.

These schemes reduce to upwind differencing for zero conduction at $|P| \geq 2$ and at $|P| \geq 10$, respectively. In practice there is little difference between the results obtained by using the power law scheme or the piecewise linear scheme. All the calculations reported in Refs. [3]–[7] were done using the power law scheme.

In the present code we employ for the temperature equations a modification of the piecewise linear scheme that is more robust in cases when strong gradients are present. We now use Eq. (11) together with the assignment

$$D'_e = \max\left(0,\, D_e - |F_e|\frac{\min(\phi_P, \phi_E)}{\phi_P + \phi_E}\right),$$

where ϕ is the strictly positive quantity that is being discretized. In comparison with the piecewise linear scheme this discretization enhances the strength of conduction whenever the values of ϕ vary widely over one mesh spacing. The modification helps to avoid convergence to a solution in which the temperature is negative at some point in the grid.

The continuity equation is purely convective. In order to evaluate, e.g., the mass flux through the 'east' cell face, it is natural to employ the assignment $F_e = \rho_e u_e A_e$, where $\rho_e = (\rho_P + \rho_E)/2$. (Remember that ρ is discretized on cell centers and u on cell faces.) That is in fact what was done for all the calculations reported in Refs. [3]–[7]. At present, however, we employ

$$\rho_e = \begin{cases} (\rho_P + \rho_E)/2, & \text{if } u_e(\rho_P - \rho_E) \geq 0 \\ 2\rho_P\rho_E/(\rho_P + \rho_E), & \text{if } u_e(\rho_P - \rho_E) < 0 \end{cases} \qquad (12)$$

The mass flux $F_e = \rho_e u_e A_e$ computed by using Eq. (12) is a continuous function of ρ_P, ρ_E, and u_e. Furthermore, $(\rho_P + \rho_E)/2 \geq 2\rho_P\rho_E/(\rho_P + \rho_E)$, with equality holding only if $\rho_P = \rho_E$. In comparison with the standard assignment, $\rho_e = (\rho_P + \rho_E)/2$, the mass flux is reduced when the flow is from a cell of lower ρ into a cell of higher ρ. This helps to keep ρ positive. The procedure bears some relation to upwind differencing, but it is significant only for large differences in ρ between neighbouring cells and maintains second-order accuracy.

Pressure correction procedure. The need for a special solution procedure for the continuity equations may be seen most clearly by considering the system of equations that governs the incompressible flow of a simple fluid. If one would consider the momentum balance equation to govern the velocity field and the energy equation to govern the temperature, then the continuity equation would have to govern the pressure. But the pressure does not even appear in that equation.

For compressible flow the pressure is a derived quantity, and the density is one of the primary variables. The continuity equation then appears suitable in principle for relaxation of the density field, but severe problems appear for low Mach number flows, when the fluid is effectively incompressible. The traditional recommendation is to employ an explicit discretization in time for the continuity equation, irrespective of the treatment of the other equations in the system. The timestep used for the continuity equation is then governed by the CFL condition based on the velocity of sound.

In our code a method due to Patankar and Spalding is employed. Their approach is to satisfy the continuity equation through simultaneous local changes to the density, pressure and velocity fields. We will present the method here with reference to a single, steady-state equation of the form $\partial(nu)/\partial x + \partial(nv)/\partial y = S_n$, noting that it is equally well applicable to an implicit treatment of a time-dependent equation. At each iteration on the continuity equation the following coupled adjustments are made:

$$\begin{cases} p \leftarrow p + \xi \\ n \leftarrow n + \kappa\xi \\ u \leftarrow u - c_x \dfrac{\partial \xi}{\partial x} \\ v \leftarrow v - c_y \dfrac{\partial \xi}{\partial y} \end{cases} \tag{13}$$

Inserting these changes into the continuity equation and retaining only the terms that are linear in ξ, one sees that ξ is to be obtained as solution to a standard convection–conduction equation:

$$\frac{\partial}{\partial x}\left(\kappa u\xi - nc_x \frac{\partial \xi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa v\xi - nc_y \frac{\partial \xi}{\partial y}\right) = r \tag{14}$$

where r is the residual before relaxation, $r = S_n - \partial(nu)/\partial x - \partial(nv)/\partial y$. Thus, through the relations (13) a relaxation procedure that is suitable for Eq. (14) is turned into a relaxation procedure for the continuity equation (1). The local coefficients κ, c_x and c_y

are chosen in order to minimize the damage that the replacements (14) do to the other equations of the system, i.e. the equation of state and the two equations that govern the velocities.

In order not to upset the equation of state the assignment $\kappa = (\partial n/\partial p)_T$ is appropriate. The choice of the coefficients c_x and c_y is less straightforward and requires consideration of the discretized momentum equation. The discrete equation for the x-velocity u has the form

$$A \cdot u = S_{mu} - \frac{\delta p}{\delta x}.$$

We assume that A is a diagonally dominant operator of five-point form, and that any term that does not fit into A has been moved into the right hand side. The prescription of Patankar and Spalding for the coefficient c_x is now to set $c_x = 1/\alpha$ at each point, where α is the diagonal coefficient in the matrix A. In this way the effects of the two adjustments $u \leftarrow u - c_x \partial\xi/\partial x$ and $p \leftarrow p + \xi$ partially cancel out in the x-component of the momentum equation. With the usual fluid flow problems the prescription for c_y is similar to that for c_x, but in the system (1)–(5) the y-velocity is governed by a diffusion equation instead of by a momentum balance equation. In order to let the two adjustments $v \leftarrow v - c_y \partial\xi/\partial y$ and $p \leftarrow p + \xi$ cancel approximately in the diffusion equation an assignment of the form $c_y = \kappa D/n$ is indicated.

Overall iteration procedure. In order to obtain a steady solution to the discretized system of equations (1)–(5), a procedure is employed in which each equation is relaxed in turn in a cyclic order until convergence is achieved. Time-stepping is employed, but only to obtain some under-relaxation; the discretization is fully implicit and within any single timestep the equations are not relaxed to convergence. Each cycle consists of the following actions:

1. The source terms S^a_n, S^a_{mu}, S^e_E and S^i_E are computed.
2. The momentum balance equations (2) are relaxed by changes to the field u_{1a} for each species a.
3. The total momentum equation is relaxed through identical changes in the velocities of all species.
4. The fields v_a are adjusted to satisfy the diffusion equations (3).
5. The continuity equations (1) are relaxed through simultaneous changes to n_a, u_{1a}, and v_a, for each species a.

6. The electron and ion energy equations (4) and (5) are relaxed separately by changes to the fields T_e and T_i, respectively.

7. The total energy equation (4) + (5) is relaxed by identical changes to T_e and T_i.

8. The continuity equations are relaxed once more, as in step 2.

Relaxation of each of the five-point equations is done by means of one or two iterations of the Strongly Implicit Procedure of Stone [27], as implemented in the NAG library code D03UAF [28]. The residuals of all equations are monitored in order to decide whether a converged solution has been achieved.

A special complication in the system (1)–(5) is the presence of two energy equations which can be strongly coupled over at least part of the domain through the term $k(T_e - T_i)$. Relaxing these equations separately will then lead to very slow convergence. (Analogous problems occur in the modelling of chemically reacting flow.) As seen above, this problem is dealt with by relaxing not just the separate energy equations but also the total energy equation. The total energy equation is relaxed through identical changes to T_e and T_i, and in this process the energy coupling term can be ignored.

For the relaxation of the individual momentum equations in step 2 the friction terms are treated implicitly. For the additional relaxation sweep in step 3 the inter-species friction can be ignored, since it contributes no net momentum source. For each individual species the coefficient c_x needed for the distributive relaxation of the continuity equation is the sum of two terms: one term corresponding to the momentum relaxation for that species individually, and one term corresponding to the total momentum relaxation.

Ionization has been treated implicitly in the calculations that are shown in Sec. 4, but it could have been treated explicitly or one can alternate between an explicit and an implicit treatment as suggested by Lackner et al. [29].

The procedure described above successfully solves the strongly coupled system of equations for the multi-fluid plasma. Evidently an alternative approach would be to rely on a locally one-dimensional splitting method, where the basic component of the iterative process is the solution of the coupled system of equations involving all unknowns along one poloidal or radial coordinate line. However, a comparison between the two approaches has not been made for the present system of equations.

4. Example Calculations

Calculations made with the previous single-fluid edge plasma code have been presented in Refs. [3]–[7], and in several contributions to the INTOR workshop (see also Refs. [8] and [9]). In those references the emphasis has been on parametric studies, involving variation of the assumptions made about the core plasma density and about the energy transport coefficients. The principal concern was with the peak temperature near material surfaces and the power load on these surfaces as function of the free parameters in the model.

In this Section the results of two specific applications of the multi-fluid code are presented. One application is representative for the ASDEX divertor and scrape-off plasma, and the other applies to the limiter edge of the conceptual TFCX experiment. The ASDEX simulation pertains to a two-ion-species plasma containing similar amounts of hydrogen (protons) and deuterium. The TFCX simulation is concerned with helium transport and models three ion fluids: a hydrogen fluid with mass number 2.5 (deuterium–tritium), a He^{1+} fluid, and a He^{2+} fluid.

The ASDEX simulation. The purpose of this simulation is to demonstrate the ability of the code to handle a situation in which there exist large concentrations of more than one ionic species—a case for which some numerical methods would fail to converge because of the strong frictional coupling between the ion fluids. In order for the simulation to have some physical interest as well it was decided to model an edge plasma containing similar amounts of hydrogen and deuterium.

Figure 1 shows a poloidal cross-section of the ASDEX experiment, with the domain of the calculation indicated. This region is mapped to the rectangular mesh shown in Fig. 2. The size of the region is 1.0 m × 0.04 m and the mesh contains 32×24 cells, which are strongly concentrated in front of the divertor target plate. The metric coefficients $\sqrt{g}$, h_z and h_y were taken as constants in this calculation, and B_ϑ/B was assumed to have the constant value 0.06.

The two ion species were $^1H^{1+}$ and $^2H^{1+}$. For the parallel transport coefficients $\eta_\parallel^a$, $\kappa_\parallel^e$, $\kappa_\parallel^i$, and the equipartition coefficient k we took the classical values as described in Sec. 2, with a flux limit on $\kappa_\parallel^e$ according to Eq. (6). The radial transport coefficients were assigned the following anomalous values: for both species ($a = 1, 2$) $D_n^a = 2\ m^2/s$, $D_p^a = 0$, and $\eta_y^a/m_a n_a = 0.2\ m^2/s$, while furthermore $\kappa_y^e/n_e = 4\ m^2/s$ and $\kappa_y^i/n_i = 0.2\ m^2/s$. These radial transport coefficients are near the upper end of the range found empirically for the ASDEX edge plasma.

IPP3- LKN 301-84

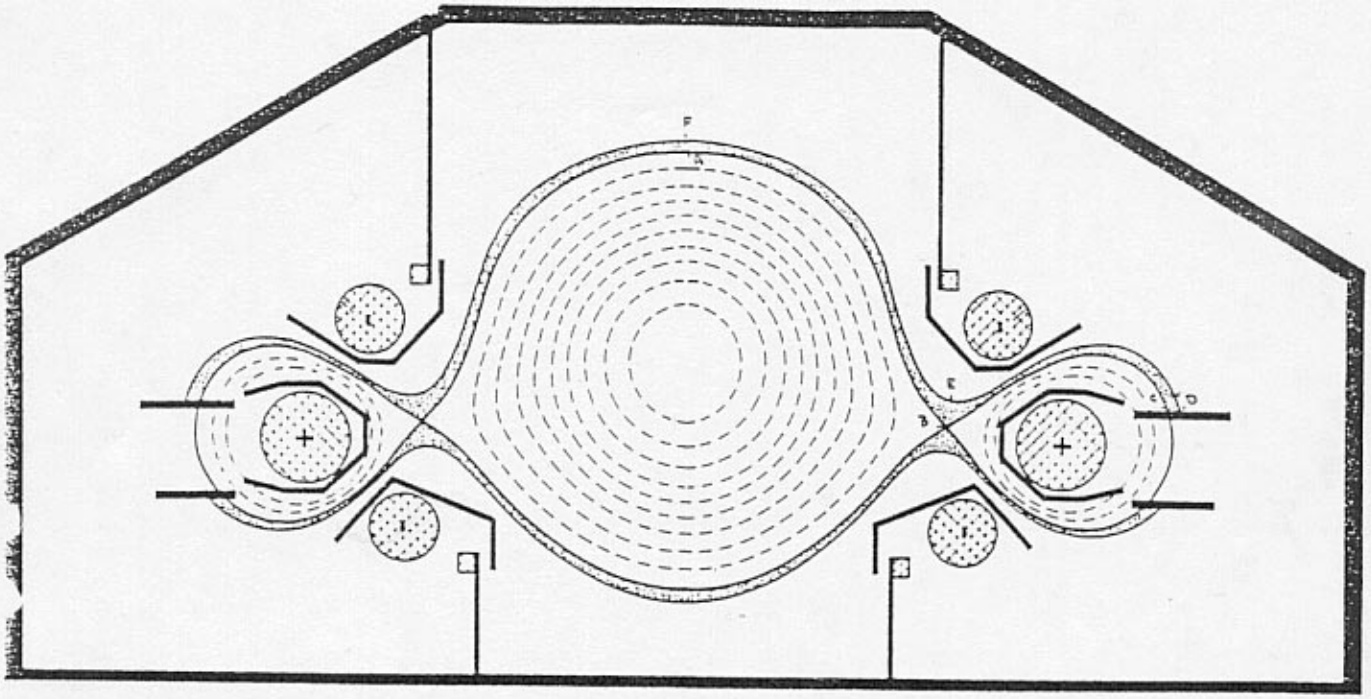


Fig. 1. Geometry of the ASDEX scrape-off layer and divertor.

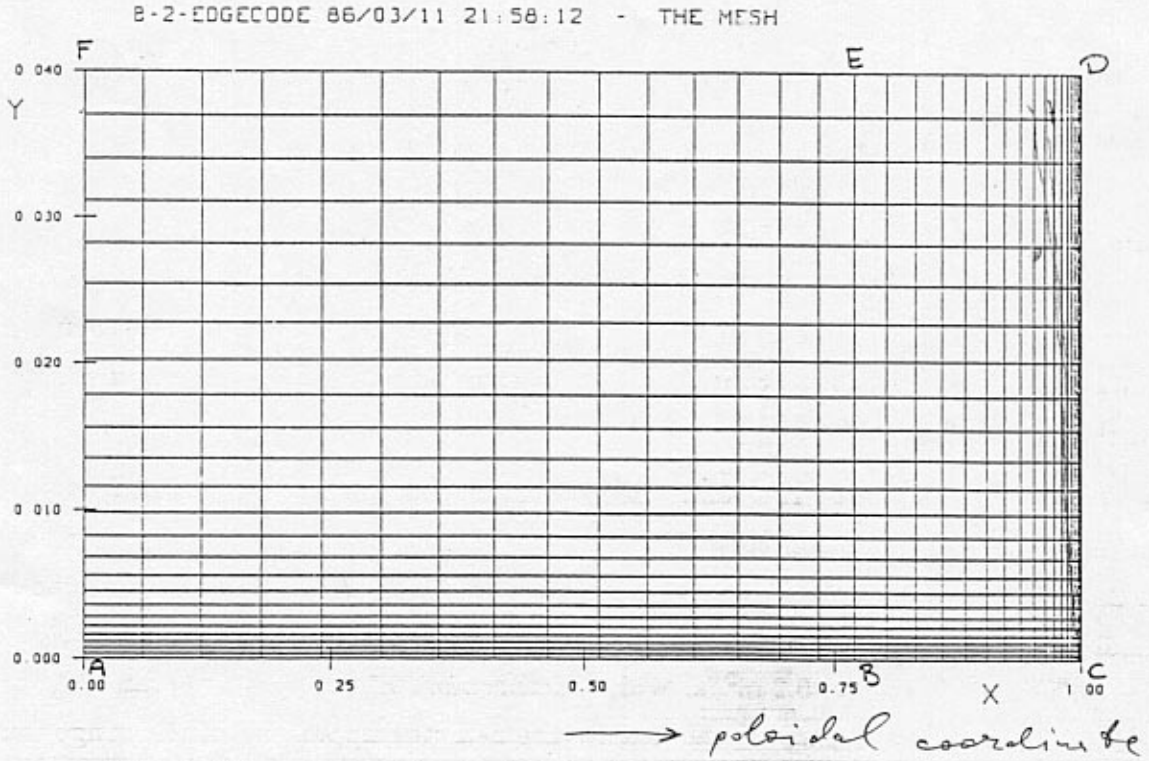


Fig. 2. The computational mesh for the ASDEX study.

The boundary conditions were chosen as follows:

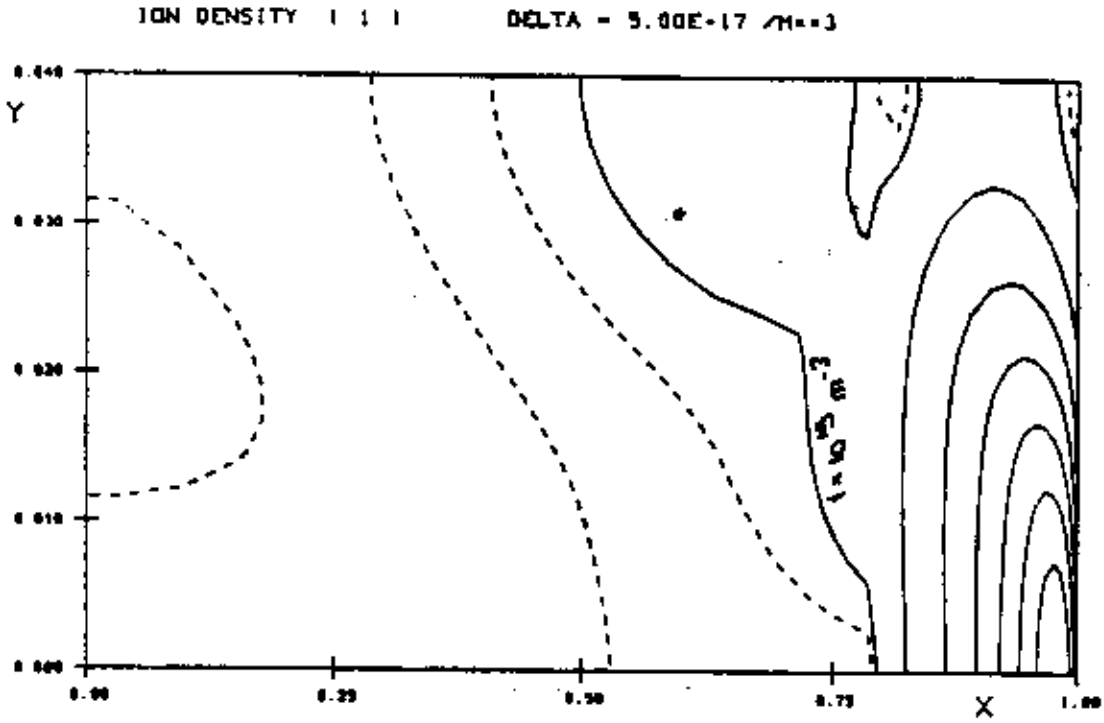
- On the interface with the main plasma, the ion densities, parallel flow velocities and two temperatures were prescribed: $n_1 = 0.9 \times 10^{19}$ m^{-3}, $n_2 = 0.9 \times 10^{19}$ m^{-3}, $u_{\parallel 1} = 0$, $u_{\parallel 2} = 0$, $T_e = 80$ eV and $T_i = 80$ eV.
- On the outer wall we prescribed zero transverse particle flux, zero shear, and pedestal temperatures $T_e = 2$ eV and $T_i = 2$ eV.
- On the upstream boundary the following symmetry conditions were specified: $\partial n_a/\partial x = 0$, $u_{\parallel a} = 0$ $(a = 1, 2)$, $\partial T_e/\partial x = 0$ and $\partial T_i/\partial x = 0$.
- On the divertor plate we specified flow at the common sound velocity for both species: $u_{\parallel 1} = u_{\parallel 2} = \sqrt{p/\rho}$. We also specified the energy fluxes $Q_e = \delta_e n_e u_e T_e$ and $Q_i = \sum_a (\delta_i n_a u_a T_i + \frac{1}{2} m_a n_a u_a u_{\parallel a}^2)$, in which $\delta_e = 4.0$ and $\delta_i = 2.5$.

To obtain the volume sources S_n^a, $S_{mu_{\parallel a}}$, S_E^e and S_E^i we relied on a simple model for the hydrogen recycling. The flux of plasma ions impinging on the divertor plate was assumed to be completely converted into a flux of neutrals flowing along the poloidal coordinate and away from the plate at a velocity $u_a = \sqrt{(10\text{ eV})/m_a}$. (Recycling coefficient unity.) These neutrals were ionized according to an approximation to the collisional radiative cross-section, which was taken as $\langle \sigma v \rangle = 3 \times 10^{-14} \times \alpha^2/(3 + \alpha^2)$ m^3/s, in which $\alpha = T_e/(10\text{ eV})$. These approximations are based on data in Ref. [30]. With each ionization event, whether involving D–T or He, was associated an electron energy loss of 25 eV (representing both the ionization energy and the radiation in multistep processes) and an ion energy gain of 5 eV (representing the kinetic energy of the neutral particle). The momentum source from ionization processes was ignored in view of the small pitch angle of the field lines.

The results of this calculation are shown in the contour plots, Figs. 3–6.

Figure 3 shows the density fields n_1 and n_2 (hydrogen and deuterium, respectively). The hydrogen and deuterium density profiles remain virtually identical through most of the scrape-off layer. In the recycling zone in front of the target plate the density rise for deuterium takes place in a narrower region and reaches a higher value than that for hydrogen. The reason is that the recycling deuterium neutrals are slower than the hydrogen neutrals, by a factor that is the square root of the mass ratio, and therefore they become ionized closer to the target plate.

Figures 4 and 5 show the electron temperature T_e and the ion temperature T_i. It can be seen that the two temperatures are not the same. In the hotter region of the scrape-off



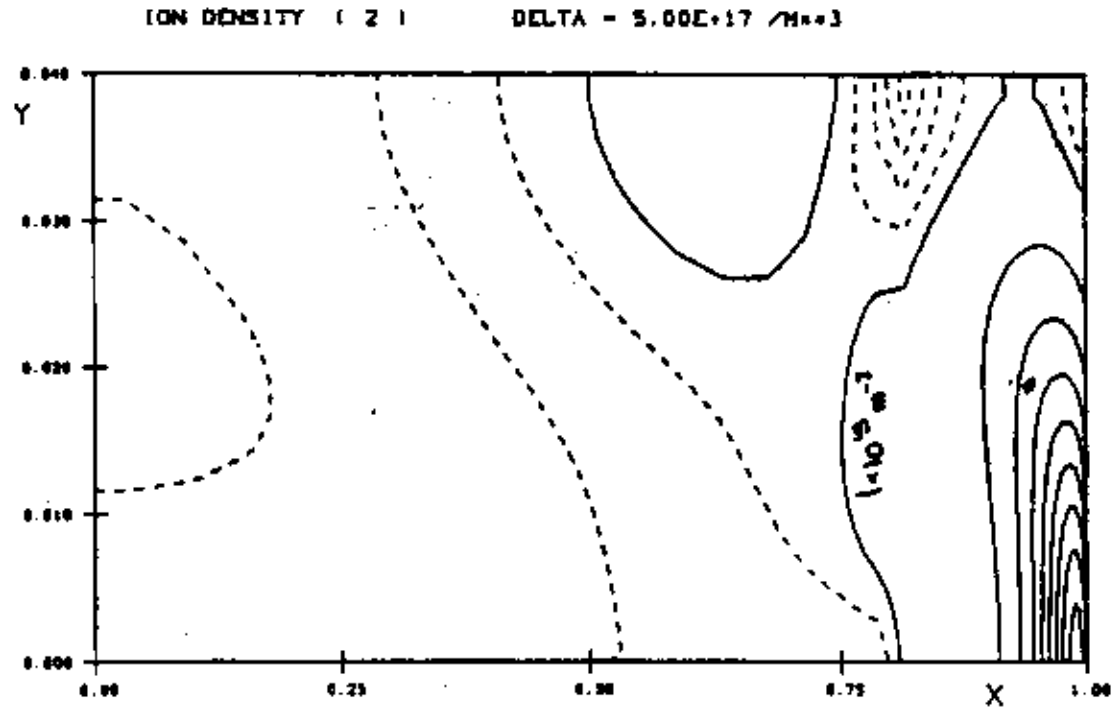


Fig. 3. Contour plots of the density of the two ion species in the ASDEX scrapeoff model. The increment between the dashed contours is 5×10^{17} m^{-3}; between the solid contours it is 2.5×10^{18} m^{-3}. Fig. 3a (upper): hydrogen (^{1}H) density. Fig. 3b (lower): deuterium density.

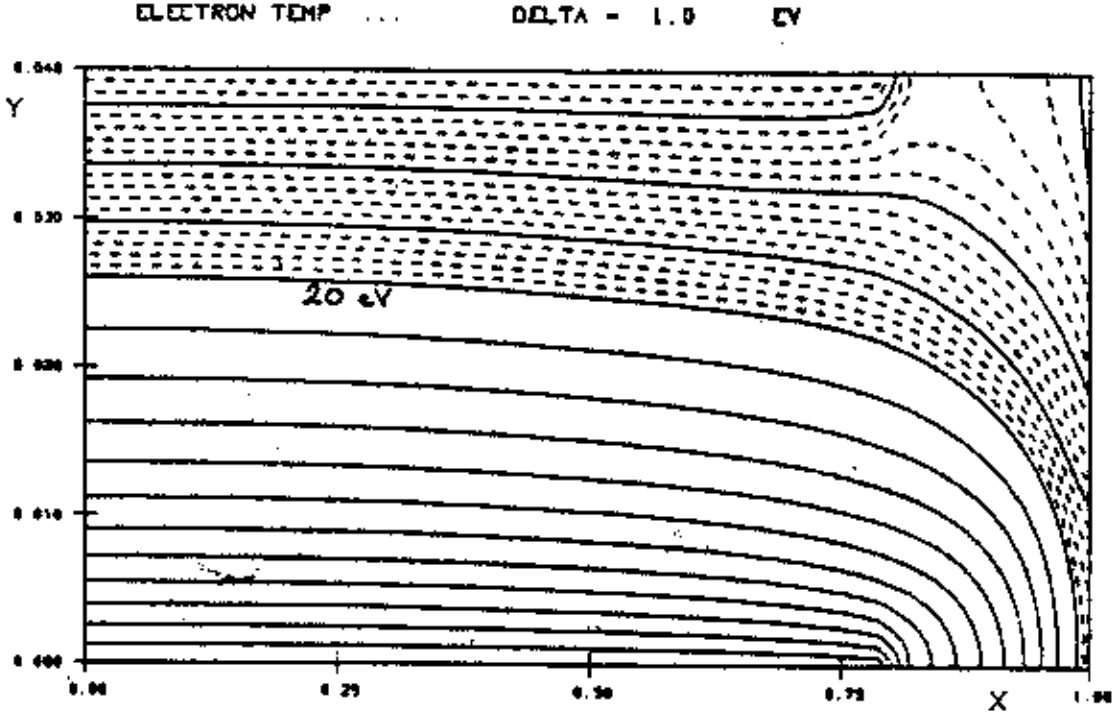


Fig. 4. Contour plot of the electron temperature. The increment between the dotted contours is 1 eV; between the solid contours it is 5 eV.

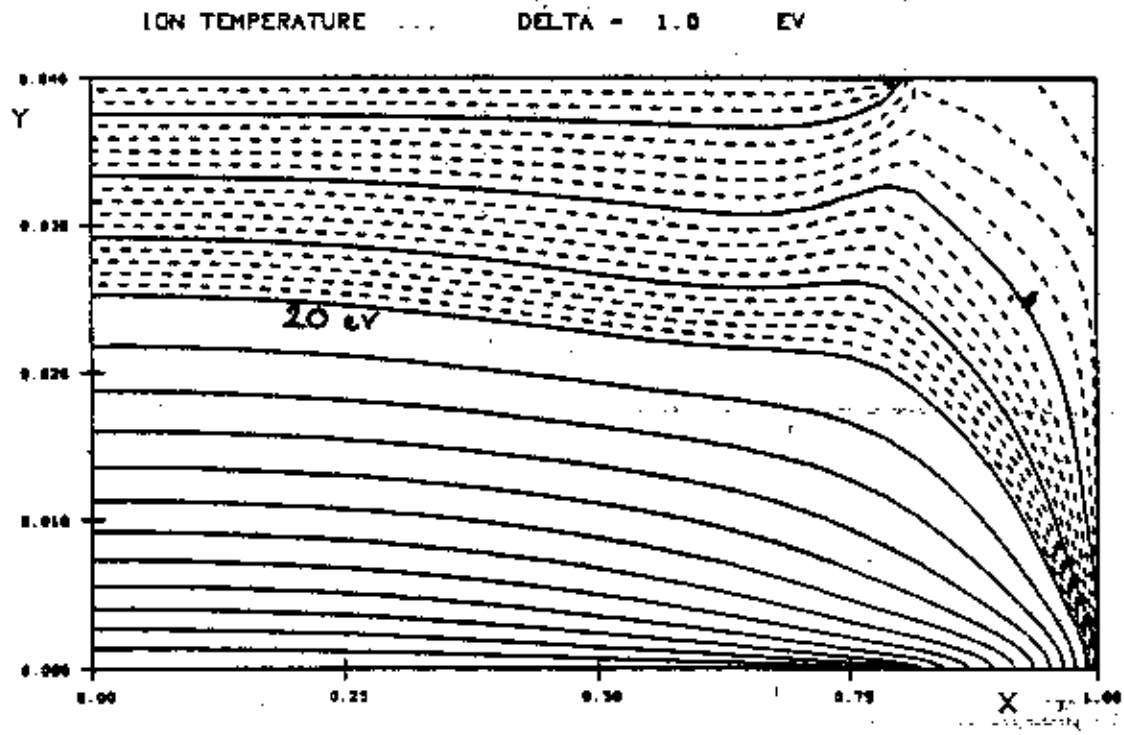


Fig. 5. Contour plot of the ion temperature. The increment between the dotted contours is 1 eV; between the solid contours it is 5 eV.

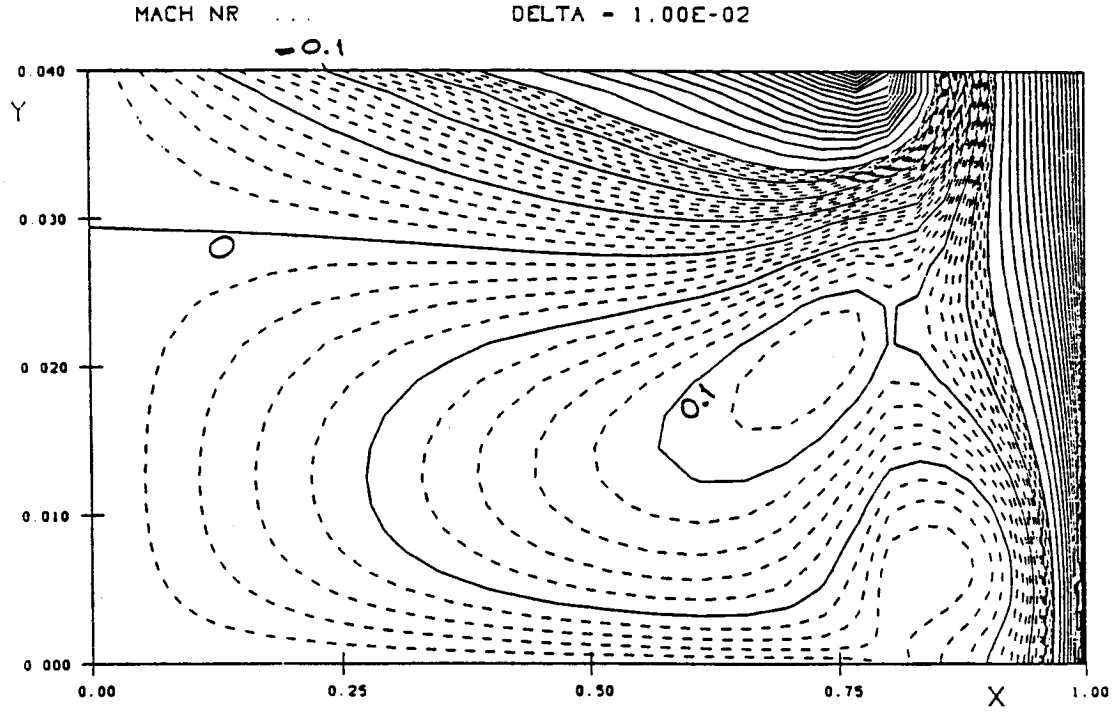


Fig. 6. Contour plot of the Mach number of the parallel flow. The increment between the dotted contours is 0.01; between the solid contours it is 0.05.

layer, outside the recycling zone, the strong parallel conductivity causes the temperatures to be nearly constant along the magnetic field. This effect is more pronounced for the electron temperature than for the ion temperature.

Figure 6 shows the Mach number of the parallel flow, computed as $u_{\|e}/\sqrt{p/\rho}$. (This is the parallel electron velocity normalized to the sound velocity of the fluid as a whole). The flow is subsonic everywhere, but approaches sonic velocity at the divertor plate. Notice that there exists a large region of recirculating flow, driven by the very localized recycling process.

The TFCX simulation. The immediate aim of the present plasma code is to aid in the study of helium transport in the edge plasma of a fusion reactor. The ratio between the concentration of helium near the limiter or divertor plate and the concentration in the main plasma is of interest for reactor design, since it determines the amount of gas that must be pumped in order to keep the central helium density at an acceptably low level. This factor of helium enrichment or dilution depends in a complicated way on the plasma flow in the scrape-off layer and on the helium recycling process, and cannot be

assessed in a realistic way except through detailed numerical simulation. As an example we present here a calculation for the edge plasma in the conceptual TFCX experiment. This calculation is a multi-species version of work done in collaboration with C.E. Singer and presented in Ref. [7].

Figure 7 shows a poloidal cross-section of the TFCX design, with the domain of the calculation indicated. This region is mapped to the rectangular mesh shown in Fig. 8. The size of the region is 4.0 m × 0.2 m, and it is divided into 32 × 24 cells. As in the ASDEX example, the metric coefficients $\sqrt{g}$, h_x and h_y are constants, and in this calculation $B_\theta/B = 0.2$.

The three ion species were H^{1+} at mass number 2.5 (representing D–T), $^4He^{1+}$ and $^4He^{2+}$. For the parallel transport coefficients and the equipartition coefficient we took again the classical values, modified by the flux limit on $\kappa_\parallel^e$. The anomalous values for the radial transport coefficients were the same as those used for the ASDEX study.

The boundary conditions were the following:

- On the interface with the main plasma, the densities of D–T and He^{2+}, the radial flux of He^{1+}, the parallel flow velocities of D–T and He^{2+}, the radial momentum flux associated with He^{1+}, and two temperatures were prescribed: $n_1 = 7 \times 10^{19}$ m^{-3}, $n_2 v_2 = 0$, $n_3 = 5 \times 10^{18}$ m^{-3}, $u_{\parallel 1} = 0$, $m_2 n_2 u_{\parallel 2} = 0$, $u_{\parallel 3} = 0$, $T_e = 150$ eV, and $T_i = 150$ eV.
- On the outer wall we prescribed zero transverse particle flux, zero shear, and pedestal temperatures $T_e = 2$ eV and $T_i = 2$ eV.
- On the midplane and on the downstream boundary inward from the limiter, symmetry conditions were specified: $\partial n_a/\partial x = 0$, $u_{\parallel a} = 0$ (for $a = 1, 2, 3$), $\partial T_e/\partial x = 0$, and $\partial T_i/\partial x = 0$.
- On the limiter plate we specified sonic flow at the common sound velocity: $u_{\parallel a} = \sqrt{p/\rho}$ (for $a = 1, 2, 3$); and energy fluxes $Q_e = \delta_e n_e u_e T_e$ and $Q_i = \sum_a (\delta_i n_a u_a T_i + \frac{1}{2} m_a n_a u_a u_{\parallel a}^2)$, with $\delta_e = 4.0$ and $\delta_i = 2.5$.

The deuterium–tritium recycling model used to calculate the volume sources was similar to the one used for the ASDEX example, but it proceeded in two stages. The recycled neutral flux travels first along the poloidal coordinate for a distance calculated from the local ionization rate coefficient, and then it travels radially inward for again such a distance. In this way we take into account to some extent the effect of the inclination of the limiter surface, which causes the neutral flux coming from the limiter to move preferentially inwards. The recycling coefficient was set to 0.98.

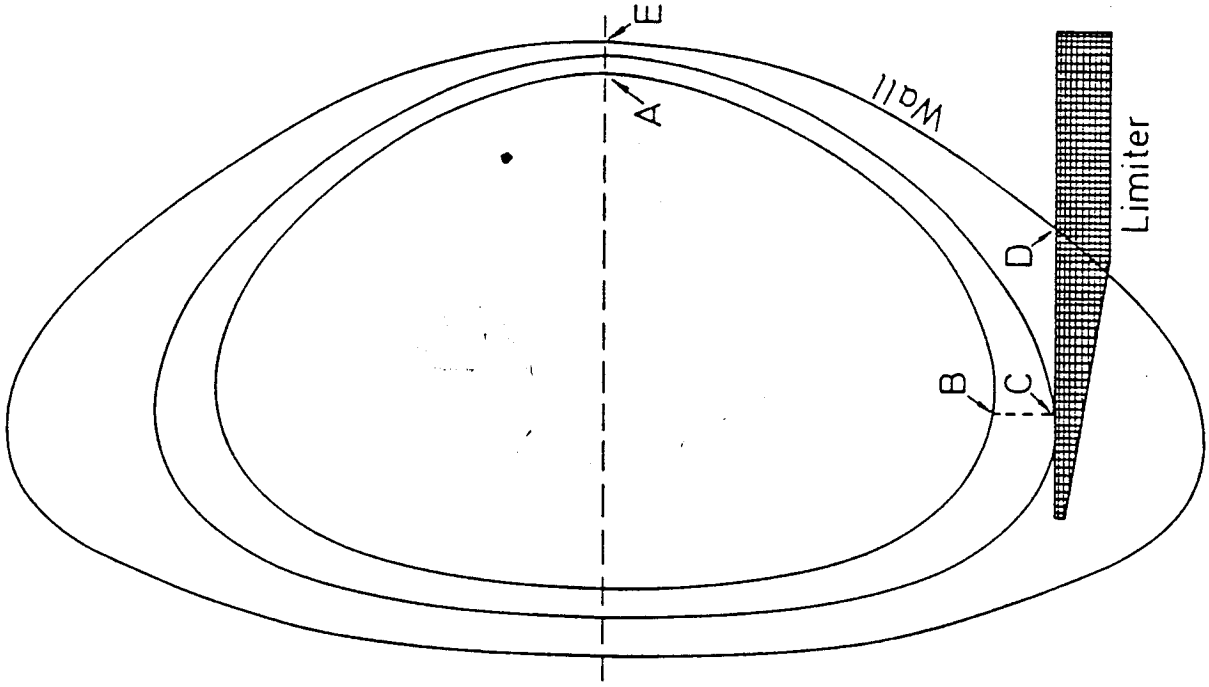


Fig. 7. Geometry of the TFCX edge plasma.

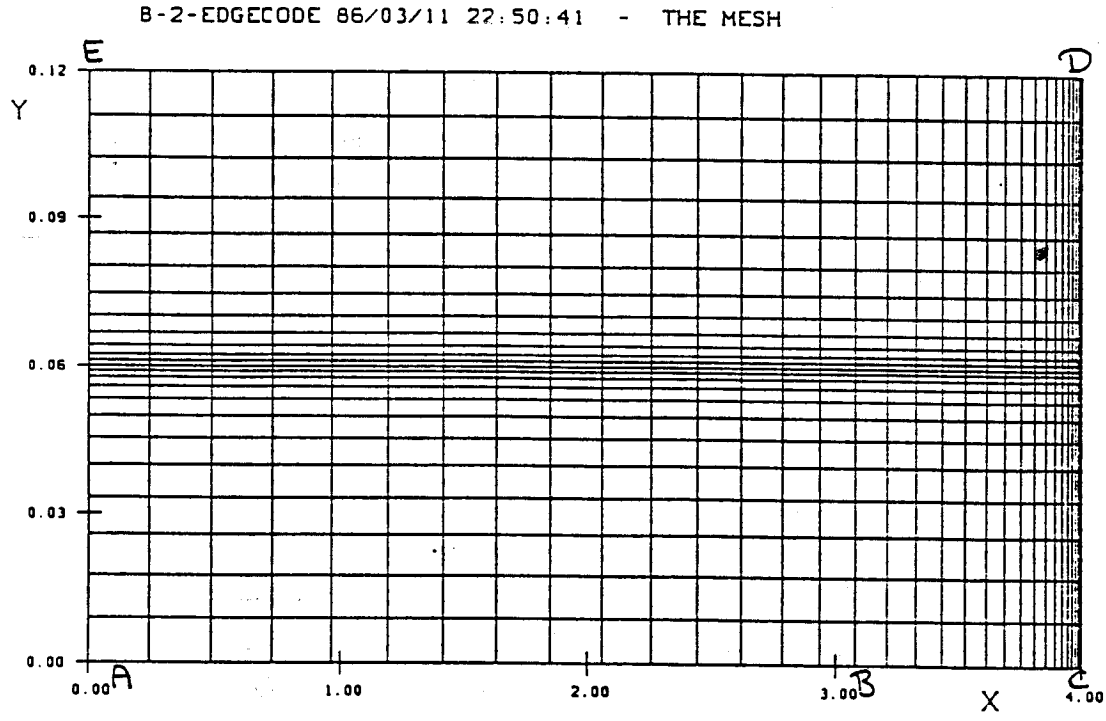


Fig. 8. The computational mesh for the TFCX study.

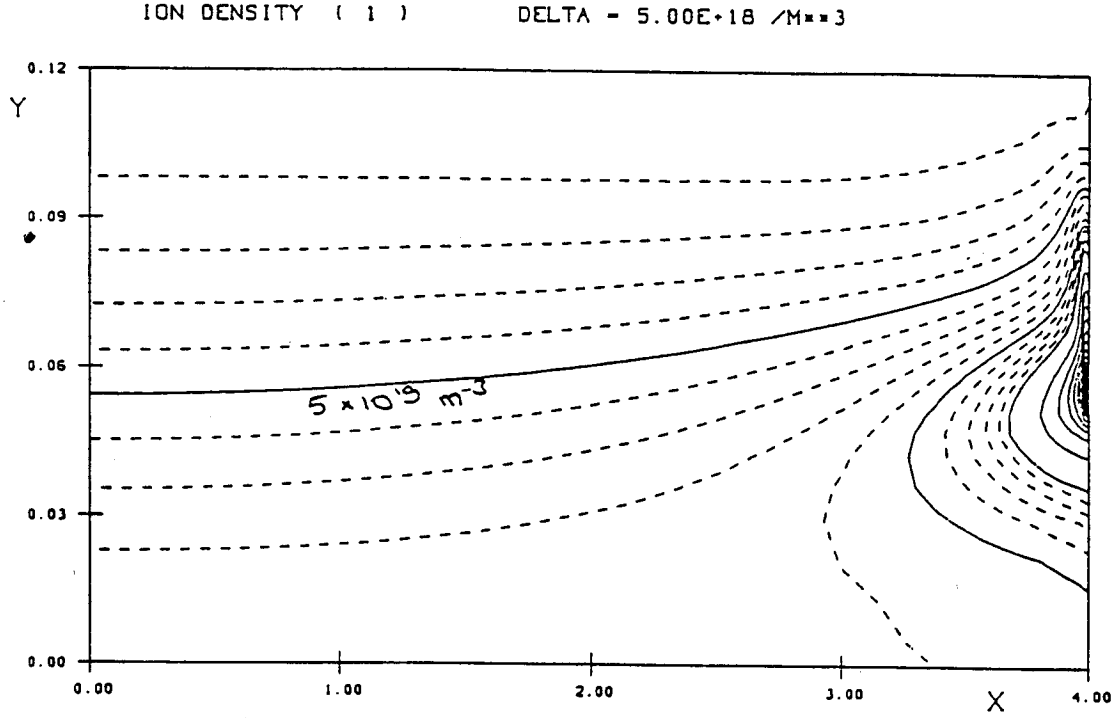


Fig. 9. Contour plot of the density of the D–T fluid in the TFCX scrapeoff layer. The increment between the dashed contours is 5×10^{18} m^{-3}; between the solid contours it is 2.5×10^{19} m^{-3}.

The helium recycling model has the same structure as that for the D–T fluid, but of course both He^{1+} and He^{2+} recycle into He^{1+}. The collisional radiative cross-section for ionization of neutral He was approximated by the expression $\langle \sigma v \rangle = 2.5 \times 10^{-14} \times \alpha^2/(20+\alpha^2)$ m^3/s, in which $\alpha = T_e/(10\ \text{eV})$. The cross-section for ionization of He^{1+} to He^{2+} was approximated by $\langle \sigma v \rangle = 4 \times 10^{-15} \times \alpha^2/(50+\alpha^2)$ m^3/s, and the cross-section for ionization of neutral hydrogen by $\langle \sigma v \rangle = 3 \times 10^{-14} \times \alpha^2/(3+\alpha^2)$ m^3/s. These approximations are again based on Ref. [30]. With each ionization event, whether involving D–T or He, was associated an electron energy loss of 25 eV and an ion energy gain of 5 eV.

The outcome of this calculation is shown in Figs. 9–14.

The deuterium–tritium ion density is shown in Fig. 9. Figures 10 and 11 show the density of He^{1+} and He^{2+}, respectively. It is seen that singly charged helium exists in significant concentrations only close to the limiter plate. In comparing Figs. 9–11 it is notable that the density rise of He^{2+} takes place over a much wider region than that of either D–T or He^{1+}. This is a combined effect of the flow in the plasma and of the relatively low cross-section for ionization to He^{2+}.

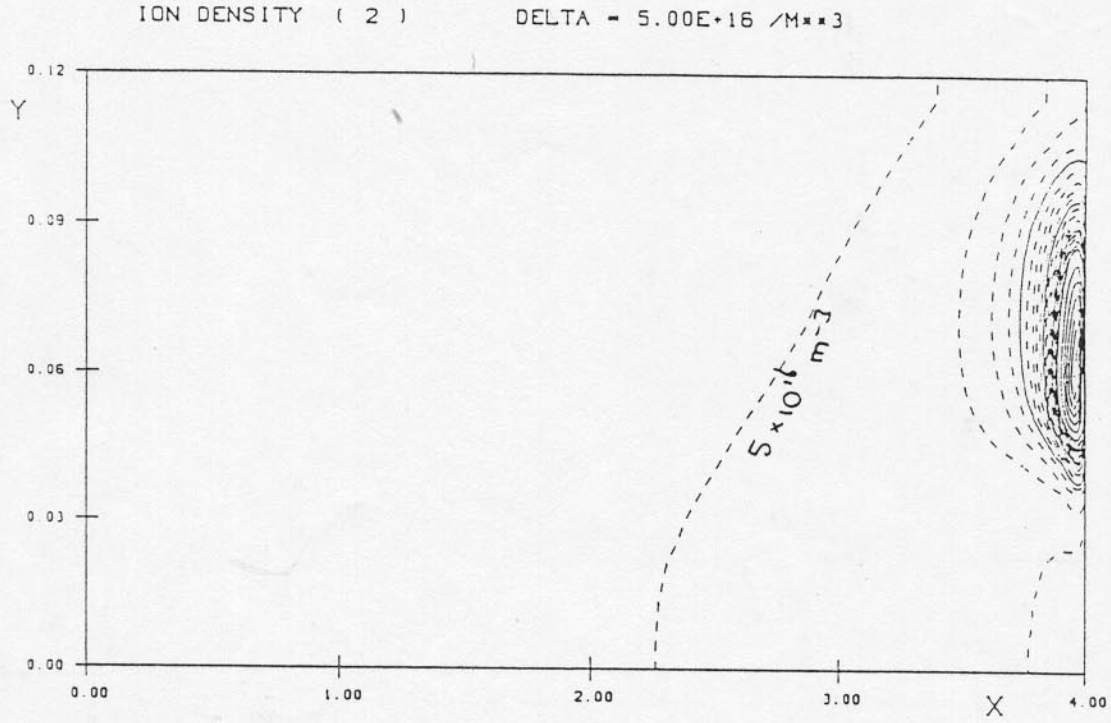


Fig. 10. Contour plot of the density of He^{1+} in the TFCX scrapeoff model. The increment between the dashed contours is 5×10^{16} m^{-3}; between the solid contours it is 2.5×10^{17} m^{-3}.

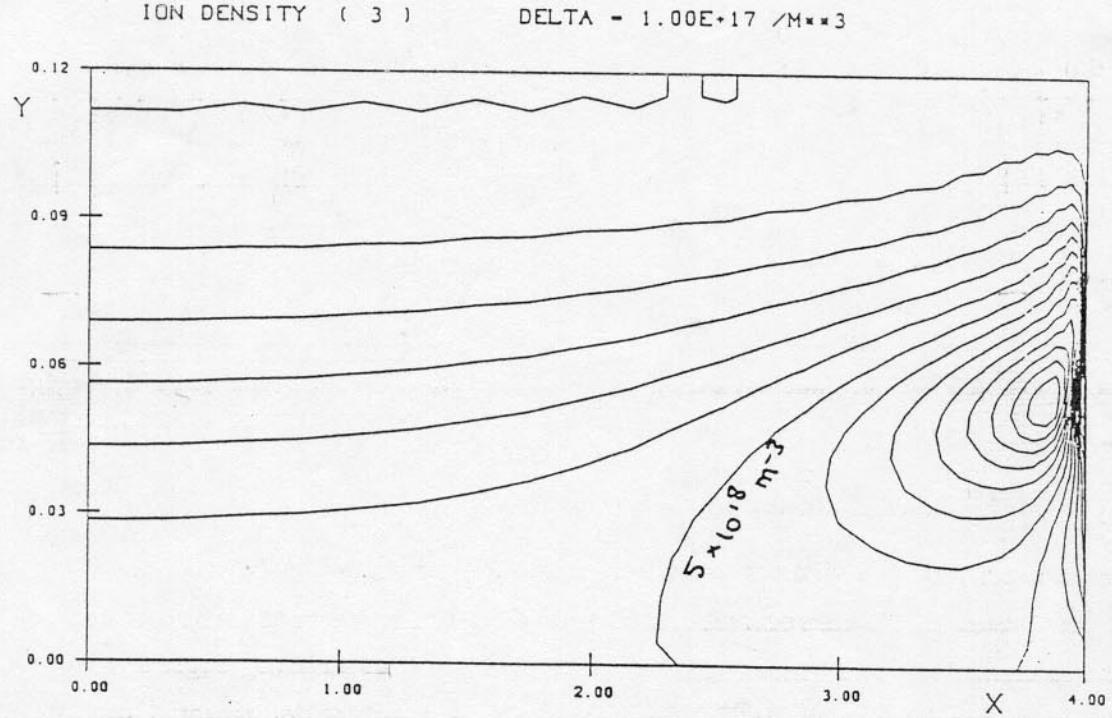


Fig. 11. Contour plot of the density of He^{2+} in the TFCX scrapeoff model. The increment between the solid contours is 5×10^{17} m^{-3}.

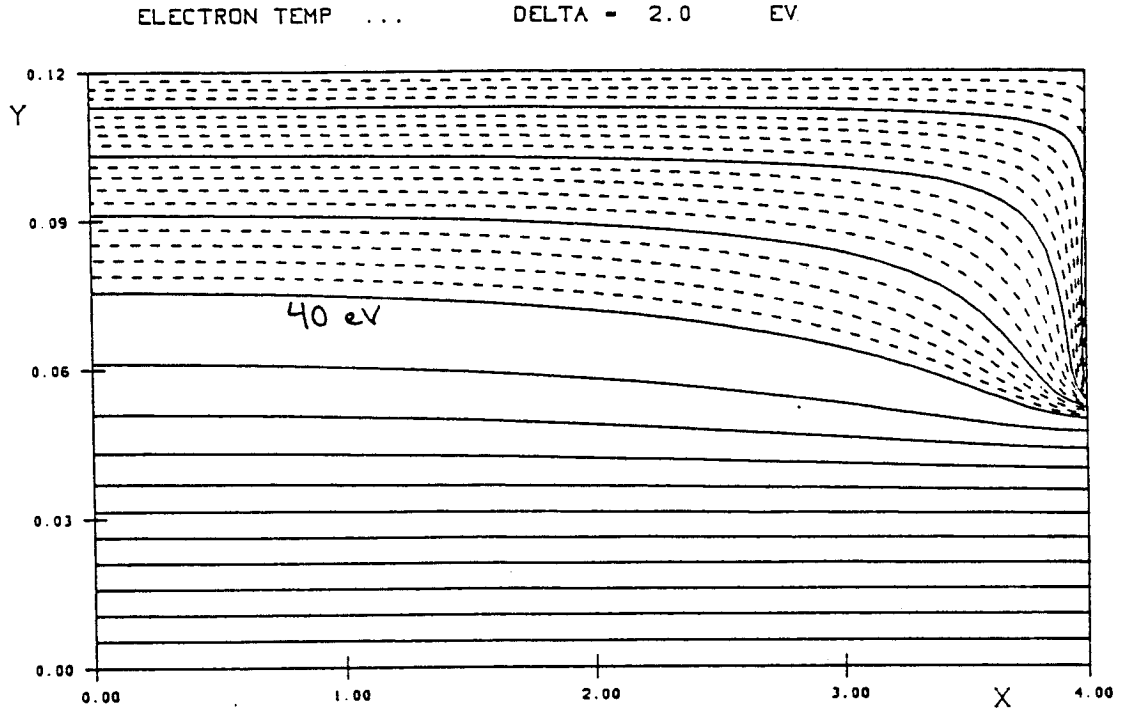


Fig. 12. Contour plot of the electron temperature. The increment between the dotted contours is 2 eV; between the solid contours it is 10 eV.

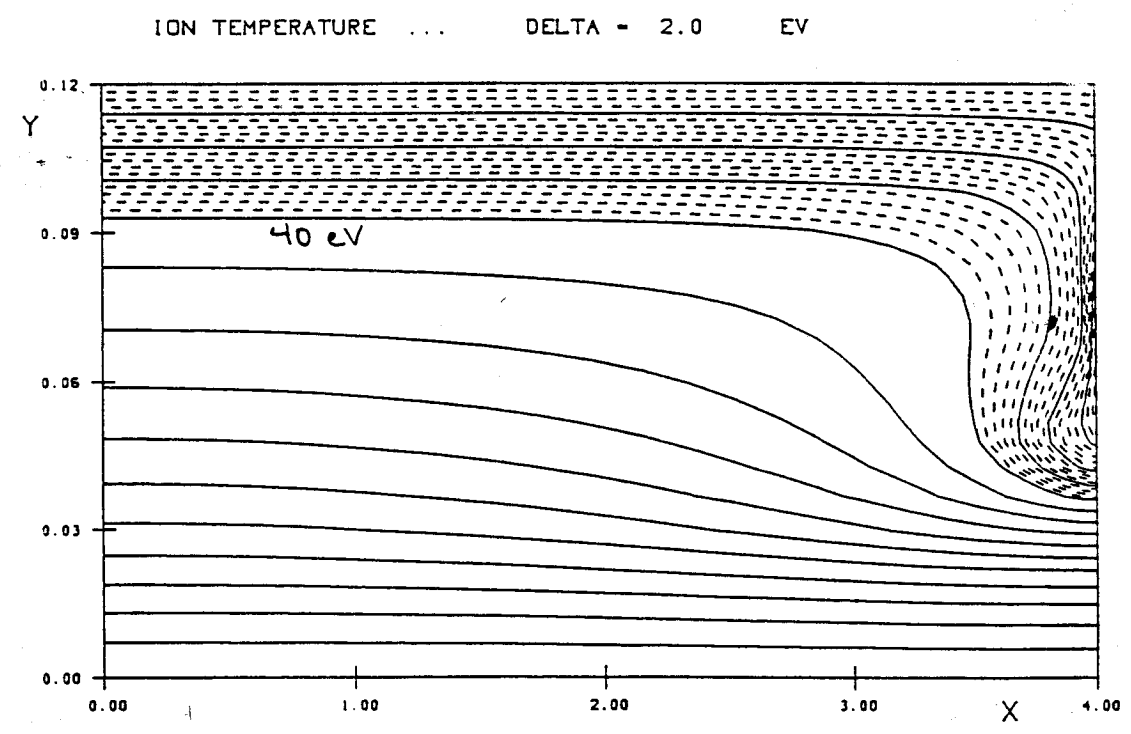


Fig. 13. Contour plot of the ion temperature. The increment between the dotted contours is 2 eV; between the solid contours it is 10 eV.

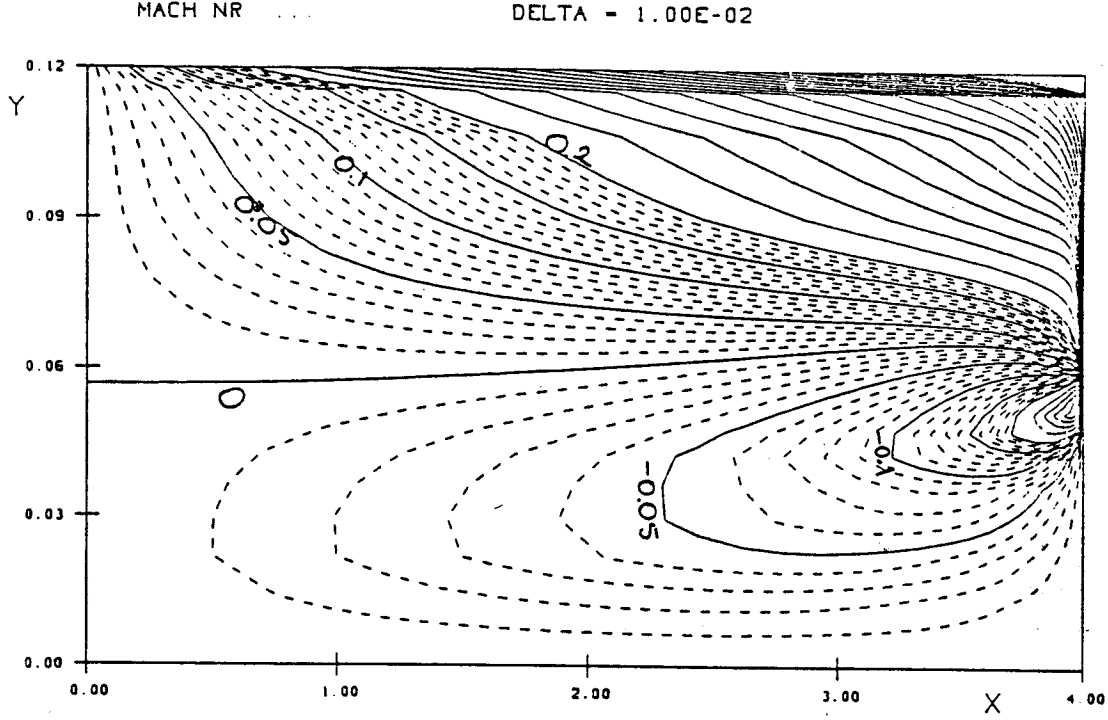


Fig. 14. Contour plot of the Mach number of the parallel flow. The increment between the dotted contours is 0.01; between the solid contours it is 0.05.

In the present calculation the prescribed ratio of He to D–T on the outer boundary of the main plasma was 7.1%, while for the ratio of the flux of He (both 1+ and 2+) to D–T on the target plate we find a value of only 2.6%. This represents a significant and unfavourable reduction of the helium concentration in the region where pumping has to take place. Much more detailed recycling models than the present one are required, however, in order to predict accurately the pumping performance of any configuration. The present result must therefore be taken as indicative only.

Figures 12 and 13 show the electron and ion temperatures, respectively. The dominating effect of parallel electron heat conduction is again clearly visible. A large electron temperature gradient along the field occurs only in the high-recycling zone.

Figure 14 shows the Mach number of the parallel flow, computed as $u_{\parallel e}/\sqrt{p/\rho}$. Notice again how the localized recycling process creates a recirculating flow throughout a substantial part of the scrape-off layer. The direction of the flow is away from the limiter in the vicinity of the separatrix. This causes a concern that impurities released from the limiter tip will be carried into the main plasma.

5. Conclusions

The modelling of helium pumping and impurity penetration requires accurate and effective models both for transport in a multi-species plasma as well as for plasma-wall interaction, and the results of such calculations may be quite sensitive to details of the recycling model employed. A code such as presented here is an indispensable tool in such studies. However, an improved treatment of neutral particles, whether based on a diffusion model or on Monte Carlo calculations, is very desirable, and more effort in that direction is required. Another area where more effort is desirable is to include diamagnetic transport terms in the equations

The calculation times required for the present code are comparatively modest (e.g. less than 3 minutes of Cray-1 time for each of the calculations presented above), so that an extension of the simulation to more than three fluids is feasible from a computational viewpoint. Application of the code to the study of transport of light impurities should, therefore, pose no special problems. Modelling of heavy impurities will be feasible provided that one restricts consideration to low charge states, which is not unreasonable for the cold scrape-off layer.

Acknowledgements

I am grateful to Prof. F. Engelmann, Dr. K. Lackner and Dr. J. Neuhauser for valuable discussions and encouragement. During the development of the earlier single-fluid code I have benefited in addition from the advice of Drs. M.F.A. Harrison, P.J. Harbour, E.S. Hotston, J.G. Morgan, and W. Schneider. I am grateful to all members of Theory Division III at the Max-Planck-Institut für Plasmaphysik, Garching, where this work was performed, for their hospitality.

This work was performed under NET Contract Number 142/83-11/FU-NL/NET between Euratom and the "Stichting voor Fundamenteel Onderzoek der Materie" (FOM), and was supported by ZWO and Euratom. The author's stay at Garching was supported by the Euratom Mobility Fund.

Appendix A. Code Organization

This Appendix is addressed to the reader who has access to the code and wishes to use it. Much effort has been made to make the code transparent, and the Appendix and the code listing may be studied concurrently.

Principal routines. The more important routines in the code and their mutual relationships are summarized here.

b2main – main program. Calls *b2init*, *b2ctrl*, *b2ceps*, *b2eval*, *b2moni*. Defines the size of the grid and the number of species, declares the principal program variables, initializes the system (routine *b2init*), and performs an iterative process to solve the equations. Each iteration involves calls to *b2ctrl*, *b2ceps*, *b2eval*, and *b2moni*.

b2init – subroutine. Called by *b2main*, calls *phygeo*. Defines the geometry (subroutine *phygeo*) and initializes the state of the plasma.

b2ctrl – subroutine. Called by *b2main* on each iteration. Sets certain physical and numerical parameters for one iteration. This routine is described in more detail in Appendix B.

b2ceps – subroutine. Called by *b2main* on each iteration. Calls *eqstat*, *physrc*, *phyvis*, *phydif*, *phythc*, *phyeqp*, *srcmod*, *mombal*, *parbal*, *enebal*. Performs one iteration of a relaxation process for the solution of the complete system of equations. The relaxation process is described in Section 3 of this report.

b2eval – subroutine. Called by *b2main* on each iteration. Evaluates certain auxiliary functionals of the estimated solution (mainly one-dimensional projections of the two-dimensional fields).

b2moni – subroutine. Called by *b2main* on each iteration. Provides monitoring output on each iteration, and final output at the conclusion.

phygeo – subroutine. Called by *b2init*. Defines the geometry and the grid. This routine is described in more detail in Appendix B.

eqstat – subroutine. Called by *b2ceps*. Computes the total pressure and the electron density. These are auxiliary quantities, not governed by differential equations.

physrc – subroutine. Called by *b2ceps*. Computes linearized expressions for external sources of ion number (each species), momentum (each species), ion energy, and electron energy. This routine is described in more detail in Appendix B.

phyvis – subroutine. Called by *b2ceps*. Computes the viscosity coefficients for each species.

phydif – subroutine. Called by *b2ceps*. Computes the diffusion velocity field for each species.

phythc – subroutine. Called by *b2ceps*. Computes the poloidal and radial thermal conductivities for electrons and ions.

phyeqp – subroutine. Called by *b2ceps*. Computes the coefficient of electron–ion temperature equilibration energy transfer.

srcmod – subroutine. Called by *b2ceps*. Computes additional contributions to the source terms: (a) contributions associated with the implicit time discretization; (b) contributions in the momentum and energy equations due to the electron pressure gradient, the thermal force, and ion–ion friction; (c) numerical terms that vanish in steady state, but that serve to stabilize the iteration.

mombal – subroutine. Called by *b2ceps*. Evaluates the residual and performs relaxation for the momentum balance equations (called once for all species).

parbal – subroutine. Called by *b2ceps*. Evaluates the residual and performs relaxation for the continuity equations (called separately for each species).

enebal – subroutine. Called by *b2ceps*. Evaluates the residual and performs relaxation for the two energy equations.

Principal variables. The following Fortran variables (parameters, scalars, and arrays) occur in many routines throughout the code.

nx, *ny* – integer. The number of interior cells along the x- and y-axis, respectively. With boundary cells included there are $(nx+2)*(ny+2)$ cells in the mesh, indexed from $(0{:}nx+1, 0{:}ny+1)$. *nx* and *ny* are initialized in *b2main* and are never changed.

nxd, *nyd* – integer constant. Used instead of *nx* and *ny* in array declarations.

nfld – integer constant. Number of ion species.

pi – real constant. The mathematical constant π.

me, *mp* – real constant. The electron and proton mass, respectively, expressed in kilograms.

ev, *qe* – real constant. The magnitude of 1 eV in Joules and of the elementary charge in Coulombs. (The numerical values are equal.)

mi, *zi* – (1:*nfld*) real array. The ion mass in kilograms and the ion charge number, respectively, for each species. Initialized in *b2main* and never changed.

vol, *gx*, *gy*, *sx*, *sy*, *rr* – (0:*nxd*+1, 0:*nyd*+1) real array. These arrays describe the local metric properties of the grid. If (ix, iy) is in (0:*nx*+1, 0:*ny*+1) then $vol(ix, iy)$ is the volume of the (ix, iy) cell, $gx(ix, iy)$ is the inverse of the x-diameter, $gy(ix, iy)$ is the inverse of the y-diameter, and $rr(ix, iy)$ is the ratio B_θ/B. If (ix, iy) is in (0:*nx*, 0:*ny*+1) then $sx(ix, iy)$ is the area of the face between cells (ix, iy) and $(ix+1, iy)$. If (ix, iy) is in (0:*nx*+1, 0:*ny*) then $sy(ix, iy)$ is the area of the face between cells (ix, iy) and $(ix, iy+1)$. The grid metric is initialized in *phygeo* and is never changed.

ni – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The number density for each species. Initialized in *b2init* and updated under control of *b2ceps*.

uu, *vv*, *up* – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The poloidal velocity, radial velocity, and parallel velocity, respectively, for each species. At present the only difference between *uu* and *up* is a factor B_θ/B. If a diamagnetic velocity is included in the model the distinction is more significant. These fields are initialized in *b2init* and updated under control of *b2ceps*.

te, *ti* – (0:*nxd*+1, 0:*nyd*+1) real array. The electron- and ion-temperatures, respectively, expressed in Joules. Initialized in *b2init* and updated under control of *b2ceps*.

fnix, *fniy* – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The ion flux in the x- and y-direction, respectively, for each species. Initialized in *b2init* and updated under control of *b2ceps*.

feex, *feey* – (0:*nxd*+1, 0:*nyd*+1) real array. The electron energy flux in the x- and y-direction, respectively. Initialized in *b2init* and updated under control of *b2ceps*.

feix, *feiy* – (0:*nxd*+1, 0:*nyd*+1) real array. The ion energy flux in the x- and y-direction, respectively. Initialized in *b2init* and updated under control of *b2ceps*.

visx, *visy* – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The ion viscosity coefficients for the x- and y-direction, respectively, for each species. Computed in *phyvis*.

hcxe, *hcye* – (0:*nxd*+1, 0:*nyd*+1) real array. The electron heat conduction coefficients for the x- and y-direction, respectively. Computed in *phythc*.

hcxi, *hcyi* – (0:*nxd*+1, 0:*nyd*+1) real array. The ion heat conduction coefficients for the x- and y-direction, respectively. Computed in *phythc*.

eqp – (0:*nxd*+1, 0:*nyd*+1) real array. The electron–ion energy equipartition coefficient. Computed in *phyeqp*.

snic, *sniv* – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The coefficients in the ion source linearization for each species. Computed in *physrc*; additional terms included in *srcmod*.

smoc, *smov* – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*+1) real array. The coefficients in the momentum source linearization for each species as well as for the sum of all species. Computed in *physrc*; additional terms included in *srcmod*.

seec, *seev* – (0:*nxd*+1, 0:*nyd*+1) real array. The coefficients in the electron energy source linearization. Computed in *physrc*; additional terms included in *srcmod*.

seic, *seiv* – (0:*nxd*+1, 0:*nyd*+1) real array. The coefficients in the ion energy source linearization. Computed in *physrc*; additional terms included in *srcmod*.

resco – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The residual of the continuity equation, for each species. Computed in *parbal*.

resmo – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*+1) real array. The residual of the momentum balance equation, for each species separately and for all species together. Computed in *mombal*.

resdi – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The residual of the diffusion equation, for each species. Computed in *b2ceps* from the result of *phydif*.

resee, *resei* – (0:*nxd*+1, 0:*nyd*+1) real array. The residual of the electron energy and the ion energy equation, respectively. Computed in *enebal*.

corni – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The latest correction made to the ion density, for each species. Computed in *parbal*.

corup – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*+1) real array. The latest correction made to each ion parallel velocity field separately and to the bulk ion velocity. Computed in *mombal*.

corvv – (0:*nxd*+1, 0:*nyd*+1, 1:*nfld*) real array. The latest correction made to each ion diffusion velocity field. Computed in *b2ceps* from the result of *phydif*.

corte, *corti* – (0:*nxd*+1, 0:*nyd*+1) real array. The latest corrections made to the electron- and ion-temperatures, respectively. Computed in *enebal*.

Appendix B. Problem-Dependent Routines

The routines *phygeo* and *physrc* are properly considered as part of the problem specification. The routine *b2ctrl* specifies numerical control parameters for each iteration and may also be used to specify a physical continuation procedure. These three routines are likely to vary between applications, and are therefore described in more detail here.

Routine *phygeo*. The purpose of this routine is to specify the geometry and the grid metric. The routine is called once in the initialization phase of the computation.

The heading for routine *phygeo* is as follows:

```
      subroutine phygeo (nx, ny, nxd, nyd, xx, yy,
     . .    vol, gx, gy, sx, sy, rr)

*  -- input arguments --
      integer nx, ny, nxd, nyd

*  -- output arguments --
      real  xx(0:nxd+1,1:*), yy(0:nyd+1,1:*),
     .      vol(0:nxd+1,0:nyd+1), gx(0:nxd+1,0:nyd+1),
     .      gy(0:nxd+1,0:nyd+1), sx(0:nxd+1,0:nyd+1),
     .      sy(0:nxd+1,0:nyd+1), rr(0:nxd+1,0:nyd+1)
```

The geometry is topologically rectangular. One must choose a mapping of the physical region in the poloidal plane to a rectangle in the x-y plane, with the flux surfaces corresponding to lines of constant y. Therefore, x is a poloidal coordinate and y is a radial coordinate.

The input arguments nx and ny specify the size of the grid. The grid contains nx interior cells along the x-coordinate and ny interior cells along the y-coordinate. The total number of cells is $(nx+2)*(ny+2)$; the outer rim of cells is used to impose the boundary conditions of the problem. The cells are indexed over $(0{:}nx+1, 0{:}ny+1)$. nx and ny are both greater than 0.

The input arguments nxd and nyd are used instead of nx and ny in the array declarations. Therefore, $nxd \geq nx$ and $nyd \geq ny$. Elements outside the $(0{:}nx+1, 0{:}ny+1)$ subrange of any $(0{:}nxd+1, 0{:}nyd+1)$ array should never be referenced.

The output arguments $xx(0{:}nx+1, 1{:}2)$ and $yy(0{:}ny+1, 1{:}2)$ specify the coordinates of the mesh. For (ix, iy) in $(0{:}nx+1, 0{:}ny+1)$ the 'center' of the (ix, iy) cell has coordinates $x = xx(ix, 1)$ and $y = yy(iy, 1)$. For (ix, iy) in $(0{:}nx, 0{:}ny+1)$ the face between cells (ix, iy) and $(ix+1, iy)$ has x-coordinate $xx(ix, 2)$. For (ix, iy) in $(0{:}nx+1, 0{:}ny)$ the face between cells (ix, iy) and $(ix, iy+1)$ has y-coordinate $yy(iy, 2)$.

The output arguments *vol*, *gx*, *gy*, *sx*, *sy*, *rr* specify metric properties of the grid. If (ix, iy) is in $(0{:}nx{+}1, 0{:}ny{+}1)$ then $vol(ix, iy)$ is the volume of the (ix, iy) cell, $gx(ix, iy)$ is the inverse of the x-diameter, $gy(ix, iy)$ is the inverse of the y-diameter, and $rr(ix, iy)$ is the local ratio B_θ/B. If (ix, iy) is in $(0{:}nx, 0{:}ny{+}1)$ then $sx(ix, iy)$ is the area of the face between cells (ix, iy) and $(ix{+}1, iy)$. If (ix, iy) is in $(0{:}nx{+}1, 0{:}ny)$ then $sy(ix, iy)$ is the area of the face between cells (ix, iy) and $(ix, iy{+}1)$.

The base unit for the grid metric is the meter. A common factor in *vol*, *sx*, and *sy* will scale out of all differential equations. For definiteness, however, it is advisable to compute these quantities as though each cell extends all the way around the toroidal circumference of the torus (the ignorable coordinate). In that case, integrated quantities correspond to the total content (of particles, energy, or whatever) in the torus.

For a large aspect ratio torus, *gx* and *sy* are almost independent of the y-coordinate, and their variation along the x-coordinate is associated with non-uniformity of the mesh in that direction. The variation of *gy* and *sx* is associated both with nonuniformity of the mesh in the y-direction and with variation of the distance between flux surfaces along the x-coordinate.

For accuracy it is desirable that the grid metric parameters for the interior cells are smoothly varying over space. This is achieved by constructing the grid via a smooth map of a uniform grid. The outer rim of cells, used only to impose the boundary conditions, may be chosen to be very thin. In spite of the terminology, it is not required that the cell 'centers' be located exactly midway between the cell faces.

Routine *physrc*. The purpose of this routine is to compute the external sources of particles (each ion species), momentum (each ion species), electron energy, and ion energy. These are the terms S_n^a, $S_{mu_\parallel}^a$, S_E^e, and S_E^i in Eqs. (1)–(5). The routine is called once every iteration. The sources may be arbitrary nonlinear and nonlocal functions of the state of the plasma, although the code is not guaranteed to converge for any choice of source term. Sources in the outer rim of cells serve to impose the boundary conditions.

Routine *physrc* is to compute the coefficients of a local linear expression that approximates the source term in the vicinity of the present state of the plasma. Each ion number source is linearized with respect to the corresponding ion density, each momentum source with respect to the corresponding parallel velocity, the electron energy source with respect to the electron temperature, and the ion energy source with respect to the ion temperature.

The heading for routine *physrc* is as follows:

```
      subroutine physrc (nx, ny, nxd, nyd, nfld,
     .      mi, zi, iter, ff, xx, yy,
     .      vol, gx, gy, sx, sy, rr,
     .      ni, uu, vv, te, ti, pr, ne, phi, up,
     .      snic, sniv, smoc, smov, seec, seev, seic, seiv, w)

*  -- input arguments --
      integer nx, ny, nxd, nyd, nfld, iter
      real  mi(1:nfld), zi(1:nfld),
     .      ff(1:*), xx(0:nxd+1,1:*), yy(0:nyd+1,1:*),
     .      vol(0:nxd+1,0:nyd+1),
     .      gx(0:nxd+1,0:nyd+1), gy(0:nxd+1,0:nyd+1),
     .      sx(0:nxd+1,0:nyd+1), sy(0:nxd+1,0:nyd+1),
     .      rr(0:nxd+1,0:nyd+1),
     .      ni(0:nxd+1,0:nyd+1,1:nfld),
     .      uu(0:nxd+1,0:nyd+1,1:nfld),
     .      vv(0:nxd+1,0:nyd+1,1:nfld),
     .      te(0:nxd+1,0:nyd+1), ti(0:nxd+1,0:nyd+1),
     .      pr(0:nxd+1,0:nyd+1), ne(0:nxd+1,0:nyd+1),
     .      phi(0:nxd+1,0:nyd+1),
     .      up(0:nxd+1,0:nyd+1,1:nfld)

*  -- output arguments --
      real  snic(0:nxd+1,0:nyd+1,1:nfld),
     .      sniv(0:nxd+1,0:nyd+1,1:nfld),
     .      smoc(0:nxd+1,0:nyd+1,1:nfld),
     .      smov(0:nxd+1,0:nyd+1,1:nfld),
     .      seec(0:nxd+1,0:nyd+1), seev(0:nxd+1,0:nyd+1),
     .      seic(0:nxd+1,0:nyd+1), seiv(0:nxd+1,0:nyd+1)

*  -- workspace arguments --
      real w(0:nxd+1,0:nyd+1,1:nfld)
```

The input arguments *nx*, *ny*, *nxd*, and *nyd* have the same meaning as in routine *phygeo*.

The input argument *nfld* gives the number of ion species. Because the content of routine *physrc* is problem-dependent, it may well be written in such a way as to tolerate just one specific value of *nfld*.

The input arguments *mi*(1:*nfld*) and *zi*(1:*nfld*) specify the mass of each ion species and its charge number, respectively. The mass is expressed in kilograms.

The input argument *iter* is an iteration counter. In many cases it may be ignored by *physrc*. It may be used, for instance, when one wishes to alternate between iterations the implicit/explicit treatment of ionization and recombination processes.

The input argument $ff(1{:}*)$ holds certain global parameters of the plasma, computed in *b2eval* (quo vide). It may often be ignored by *physrc*, but is available if needed.

The input arguments $xx(0{:}nx{+}1, 1{:}2)$ and $yy(0{:}ny{+}1, 1{:}2)$ are as computed in *phygeo*. The input arguments $xx(0{:}nx{+}1, 3{:}*)$ and $yy(0{:}ny{+}1, 3{:}*)$ hold certain one-dimensional reductions of the two-dimensional plasma profiles, again computed in *b2eval*. These can often be ignored by *physrc*.

The input arguments *vol*, *gx*, *gy*, *sx*, *sy*, and *rr* are as computed in *phygeo*.

The input argument $ni(0{:}nx{+}1, 0{:}ny{+}1, 1{:}nfld)$ specifies the ion density, expressed in m^{-3}, in each cell for each species.

The input argument $uu(0{:}nx, 0{:}ny{+}1, 1{:}nfld)$ specifies the ion flow velocity in the x-direction, expressed in $\mathrm{m\,s}^{-1}$, on each cell x-face for each species.

The input argument $vv(0{:}nx{+}1, 0{:}ny, 1{:}nfld)$ specifies the ion flow velocity in the y-direction, expressed in $\mathrm{m\,s}^{-1}$, on each cell y-face for each species.

The input arguments $te(0{:}nx{+}1, 0{:}ny{+}1)$ and $ti(0{:}nx{+}1, 0{:}ny{+}1)$ specify the electron temperature and the ion temperature, respectively, in each cell. Temperatures are expressed in Joules.

The input argument $pr(0{:}nx{+}1, 0{:}ny{+}1)$ specifies the total pressure, in $\mathrm{J\,m}^{-3}$, in each cell.

The input argument $ne(0{:}nx{+}1, 0{:}ny{+}1)$ specifies the electron density, in m^{-3}, in each cell.

The input argument $phi(0{:}nx{+}1, 0{:}ny{+}1)$ should be ignored; it represents the intention to include at some future time a proper treatment of the electric potential.

The input argument $up(0{:}nx, 0{:}ny{+}1, 1{:}nfld)$ specifies the parallel ion flow velocity, expressed in $\mathrm{m\,s}^{-1}$, on each cell x-face for each species.

The output arguments $snic(0{:}nx{+}1, 0{:}ny{+}1, 1{:}nfld)$ and $sniv(0{:}nx{+}1, 0{:}ny{+}1, 1{:}nfld)$ specify a linearization for the particle source, S^a_n, integrated over each cell, for each species. That is, for (ix, iy) in $(0{:}nx{+}1, 0{:}ny{+}1)$ and is in $(1{:}nfld)$, the expression $snic(ix, iy, is){+}sniv(ix, iy, is){*}ni(ix, iy, is)$ should approximate the particle source for species is in the (ix, iy) cell. Units are s^{-1}.

The output arguments $smoc(0{:}nx, 0{:}ny{+}1, 1{:}nfld)$ and $smov(0{:}nx, 0{:}ny{+}1, 1{:}nfld)$ specify a linearization for the momentum source, $S^a_{mu_\parallel}$, integrated over each x-staggered cell, for each species. For (ix, iy) in $(0{:}nx, 0{:}ny{+}1)$ and is in $(1{:}nfld)$, the expression $smoc(ix, iy, is){+}smov(ix, iy, is){*}up(ix, iy, is)$ should approximate the momentum

source for species is in the staggered cell that surrounds the (ix, iy) x-face. Units are $\mathrm{kg\,m\,s^{-2}}$.

The output arguments $seec(0{:}nx{+}1, 0{:}ny{+}1)$ and $seev(0{:}nx{+}1, 0{:}ny{+}1)$ specify a linearization for the electron energy source, S_E^e, integrated over each cell. For (ix, iy) in $(0{:}nx{+}1, 0{:}ny{+}1)$, the expression $seec(ix, iy){+}seev(ix, iy){*}te(ix, iy)$ should approximate the electron energy source in the (ix, iy) cell. Units are $\mathrm{J\,s^{-1}}$.

The output arguments $seic(0{:}nx{+}1, 0{:}ny{+}1)$ and $seiv(0{:}nx{+}1, 0{:}ny{+}1)$ specify a linearization for the ion energy source, S_E^i, integrated over each cell. For (ix, iy) in $(0{:}nx{+}1, 0{:}ny{+}1)$, the expression $seic(ix, iy){+}seiv(ix, iy){*}ti(ix, iy)$ should approximate the ion energy source in the (ix, iy) cell. Units are $\mathrm{J\,s^{-1}}$.

A suitable choice of the source term linearization allows to impose either Dirichlet or Neumann or mixed boundary conditions, and also to specify internal boundary conditions. The book by Patankar [25] contains worthwhile advice on these matters.

Routine *b2ctrl*. The purpose of this routine is to set some physical and numerical parameters that control the iteration. The routine is called once every iteration.

The heading for routine *b2ctrl* is as follows:

```
      subroutine b2ctrl (nx, ny, nxd, nyd, nfld,
     .     mi, zi, iter, ff, xx, yy,
     .     lastit, rxf, rxd, freq, style)

*  -- input arguments --
      integer nx, ny, nxd, nyd, nfld, iter
      real  mi(1:nfld), zi(1:nfld),
     .     ff(1:*), xx(0:nxd+1,1:*), yy(0:nyd+1,1:*)

*  -- output arguments --
      logical lastit
      real  rxf, rxd, freq
      integer style(1:2)
```

The input arguments *nx*, *ny*, *nxd*, *nyd*, *nfld*, *mi*(1:*nfld*), *zi*(1:*nfld*), *iter*, *ff*(1:*), *xx*(0:*nx*+1, 1:*), and *yy*(0:*ny*+1, 1:*) all have the same meaning as in routine *physrc*.

The output argument *lastit* specifies whether the present iteration is to be the last one (*.true.* if yes). For a fixed iteration process, *lastit* depends only on the value of *iter*. Alternatively *lastit* can be made to depend on a norm of the residuals, computed in *b2eval* (quo vide) and stored in *ff*.

The output argument *rxf* specifies a relaxation factor for the present iteration. The computed corrections to the state of the plasma are all multiplied by *rxf* before being

added to the old values. A good value for *rxf* is to be found by experimentation; it will be in the range 0 to 1.

The output argument *rxd* provides another way to stabilize the iteration process, as explained here. Each partial differential equation from Eqs. (1)–(5) is discretized in five-point form, typically with a positive diagonal and negative off-diagonal terms. Before computing the incomplete L*U decomposition of the matrix, each diagonal element d will be replaced by the maximum of d and of $-rxd$ times the sum of the off-diagonal elements in the corresponding row. If $rxd \geq 1$ then this procedure makes the matrix diagonally dominant by rows. (A suitable source term linearization already guarantees that the matrix is diagonally dominant by columns). Experience shows that $rxd = 1$ is a good choice.

The output argument *freq* corresponds to the inverse timestep (in s^{-1}), and provides a third way to stabilize the iteration. A good value is to be found by experimentation.

The output argument *style*(1:2) specifies the level of output that is to be produced on the present iteration by routine *b2moni*. *style*(1) controls printed output, and *style*(2) controls graphical output. A negative value implies no output will be produced. A value 0 is suitable for the initial call. *style*(1) = 1 causes one line of output to be produced and is useful for monitoring convergence. *style*(1) = 2 produces more elaborate output, while *style*(1) = 3 is most elaborate and is suitable on the final iteration. *style*(2) > 0 produces graphical output, and is suitable on the final iteration. If an error occurs in the present iteration, then the assigned value of *style* will be disregarded in *b2moni*, and error output will be produced.

Besides defining the output arguments *lastit*, *rxf*, *rxd*, *freq*, and *style*(1:2), the routine *b2ctrl* may define or update certain physical parameters in common blocks; e.g. variables that control a continuation procedure. Such a common block may be shared by the routine *physrc*, or by other physics routines. The way this is done is problem dependent. For the calculations shown earlier in this paper, the outflow Mach number was used as a continuation parameter. This number was updated in *b2ctrl* and used in *physrc*.

References

1. J. Neuhauser, W. Schneider, R. Wunderlich, K. Lackner and R. Behringer, *Modelling of the Impurity Pumping by a Tokamak Scrape-Off Layer*, J. Nucl. Mater. **121** (1984), 194–198.
2. P.J. Harbour and J.G. Morgan, *The Transport of Impurity Ions in a Scrape-Off Plasma* (11th European Conference on Controlled Fusion and Plasma Physics, Aachen, 1983), Europhysics Conference Abstracts **7D-II** (1983), 427–430.
3. B.J. Braams, *Numerical Studies of the Two-Dimensional Scrapeoff Plasma* (11th European Conference on Controlled Fusion and Plasma Physics, Aachen, 1983), Europhysics Conference Abstracts **7D–II** (1983), 431–434.
4. B.J. Braams, *Modelling of a Transport Problem in Plasma Physics* in "Topics in Applied Numerical Analysis", Vol. 1, CWI Syllabus No. 4 (J.G. Verwer, Ed.), Centrum voor Wiskunde en Informatica, Amsterdam (1984), pp. 149–164.
5. B.J. Braams, P.J. Harbour, M.F.A. Harrison, E.S. Hotston and J.G. Morgan, *Modelling of the Boundary Plasma of Large Tokamaks*, J. Nucl. Mater. **121** (1984), 75–81.
6. B.J. Braams, M.F.A. Harrison, E.S. Hotston and J.G. Morgan, *Application of Two-Dimensional Modelling to the Scrape-off and the Single-Null Poloidal Divertor Plasma of INTOR*, in "Plasma Physics and Controlled Nuclear Fusion Research 1984" (Proc. 10th Int. Conf. Plasma Physics and Controlled Fusion Research, London, 12–19 September, 1984), Vol. 2, IAEA, Vienna, 1985; pp. 125–129.
7. B.J. Braams and C.E. Singer, *Low Temperature Plasma near a Tokamak Reactor Limiter*, Fusion Technol. **9** (1986), 320–327.
8. B.J. Braams, "Computational Studies in Tokamak Equilibrium and Transport", thesis, University of Utrecht, the Netherlands (June 1986).
9. M.F.A. Harrison, E.S. Hotston, J.G. Morgan and J.P. Maddison, *Plasma Edge Physics for NET/INTOR*, NET Report, 1986.
10. M.H. Emery and N.K. Winsor, *A Fully Two-Dimensional Equilibrium and Transport Model of the Poloidal Divertor*, Report NRL–MR–4498, Naval Research Laboratory, Washington, 1981.
11. M. Petravic, D. Heifetz, S. Heifetz and D. Post, *Recent Progress in the Tokamak Edge Modelling*, J. Nucl. Mater. **128** (1984), 91–99.
12. M. Petravic, D. Heifetz, G. Kuo-Petravic and D. Post, *Intor Divertor in a Realistic 2-D Geometry*, J. Nucl. Mater. **128** (1984), 111–113.
13. M. Petravic, D. Heifetz and D. Post, *Modelling Analysis of Tokamaks with Divertors and Pumped Limiters*, in "Plasma Physics and Controlled Nuclear Fusion Research 1984" (Proc. 10th Int. Conf. Plasma Physics and Controlled Fusion Research, London, 12–19 September, 1984), Vol. 2, IAEA, Vienna, 1985; pp. 103–112.
14. Yu.L. Igitkhanov, A.S. Kukushkin, A.Yu. Pigarov and V.I. Pistunovich, *Two-Dimensional Calculations of Near-Wall Plasma Dynamics for a Tokamak with a Poloidal Divertor* (11th European Conference on Controlled Fusion and Plasma Physics, Aachen, 1983), Europhysics Conference Abstracts **7D–II** (1983), 397–400.
15. Yu.L. Igitkhanov, A.S. Kukushkin, A.Yu. Pigarov, V.I. Pistunovich and V.A. Pozharov, *Advanced Two-Dimensional Simulation of Edge Plasma in a Poloidal Divertor*, in "Plasma Physics and Controlled Nuclear Fusion Research 1984" (Proc. 10th Int. Conf. Plasma Physics and Controlled Fusion Research, London, 12–19 September, 1984), Vol. 2, IAEA, Vienna, 1985; pp. 113–123.

16. Yu.L. Igitkhanov, A.S. Kukushkin, V.I. Pistunovich and V.A. Pozharov, *Improved Hydrodynamic Simulation of a Divertor Plasma under Weak Collisions* (12th European Conference on Controlled Fusion and Plasma Physics, Budapest, 1985), Europhysics Conference Abstracts **9F–II** (1985), 488–491.
17. S. Saito, M. Sugihara and N. Fujisawa, *Self-Consistent Numerical Analyses for Scrape-Off Plasmas and Neutral Particles in a FER Divertor Chamber*, J. Nucl. Mater. **121** (1984), 199–204.
18. M. Sugihara, S. Saito, S. Hitoki and N. Fujisawa, *Numerical Analyses of Plasma and Neutral Particle Behavior and Design Criteria for Poloidal Divertor in Fusion Experimental Reactor*, J. Nucl. Mater. **128** (1984), 114–117.
19. R. Simonini, W. Feneberg and A. Taroni, *Two Dimensional Analytic and Numerical Models of the Scrape-Off Layer of Toroidal Limiters* (12th European Conference on Controlled Fusion and Plasma Physics, Budapest, 1985), Europhysics Conference Abstracts **9F–II** (1985), 484–487.
20. S.I. Braginskii, *Transport Processes in a Plasma*, in "Reviews of Plasma Physics", Vol. 1 (M.A. Leontovich, Ed.), Consultants Bureau, New York, 1965; pp. 205–311.
21. F.L. Hinton, *Collisional Transport in Plasma*, in "Handbook of Plasma Physics", Vol. 1 "Basic Plasma Physics I" (A.A. Galeev and R.N. Sudan, Eds.), North Holland. Amsterdam, 1983; pp. 147–197.
22. D.L. Book, "NRL Plasma Formulary", Naval Research Laboratory, Washington, 1983.
23. J.F. Luciani, P. Mora and J. Virmont, *Nonlocal Heat Transport Due to Steep Temperature Gradients*, Phys. Rev. Lett. **51** (1983), 1664–1667.
24. J. Neuhauser, private communication, 1985.
25. S.V. Patankar, "Numerical Heat Transfer and Fluid Flow", Hemisphere, New York, 1980.
26. S.V. Patankar and D.B. Spalding, *A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows*, Int. J. Heat Mass Transfer **15** (1972), 1787–1806.
27. H.J. Stone, *Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations*, SIAM J. Numer. Anal. **5** (1968), 530–558.
28. Numerical Algorithms Group, "NAG Fortran Manual for Mark 10", NAG, Oxford, 1983.
29. K. Lackner, K. Behringer, W. Engelhardt and R. Wunderlich, *An Algorithm for Impurity Diffusion under Finite Reaction Rates*, Z. f. Naturf. **37a** (1982), 931.
30. K.L. Bell, H.B. Gilbody, J.G. Hughes, A.E. Kingston and F.J. Smith, *Recommended Data on the Electron Impact Ionization of Light Atoms and Ions*, J. Phys. Chem. Ref. Data **12** (1983), 891–916.