CLM - P 433

CLM - P 433

UKAEA RESEARCH GROUP

Preprint

ICARUS - A ONE-DIMENSIONAL PLASMA DIFFUSION CODE

M L WATKINS

M H HUGHES

K V ROBERTS

P M KEEPING

J KILLEEN

CULHAM LABORATORY
Abingdon Oxfordshire

1975

CLM-P433

ICARUS - A ONE-DIMENSIONAL PLASMA DIFFUSION CODE

M L Watkins, M H Hughes, K V Roberts, P M Keeping

UKAEA Culham Laboratory, Abingdon, Oxfordshire, England
(Euratom/UKAEA Fusion Association)

and

J Killeen

Lawrence Livermore Laboratory, University of California
Livermore, California, USA

(Submitted for publication in Methods in Computational Physics)

May 1975
hp

ICARUS - A One-dimensional Plasma Diffusion Code

ABSTRACT

The development of a physical model for one-dimensional diffusion in an axisymmetric, toroidal plasma containment device of the Tokamak type to one which simulates well the behaviour of an ST discharge, is monitored by the method of introduction of flux and source terms into the computer code, ICARUS. The basic model of neoclassical diffusion can be replaced readily by pseudo-classical or Pfirsch-Schlüter scaling laws, which can give results in good agreement with experiment provided that the presence of impurities and neutral particles is taken into account. The basic version of the code, ICARUS, employs simple models for the effects of impurities and neutral particles, but in accordance with the spirit of the OLYMPUS programming system, provides the interfaces necessary for easy development of the physical model.

I. Introduction

One-dimensional plasma diffusion codes (Dnestrovskii et al.,1970; Dory and Widner, 1970; Düchs, 1970; Mercier and Soubbarameyer, 1970; Düchs et al., 1971; Hogan et al.,1971; Keeping et al.,1972; Watkins et al., 1975) are used to study the temporal evolution of plasma and poloidal magnetic field in axisymmetric, toroidal, plasma containment devices of the Tokamak type (Artsimovich et al.,1969; Dimock et al.,1971; Bol et al., 1972; Drummond et al.,1973; Gibson et al.,1973; Itoh et al.,1973; Kelley et al.,1973; Rebut et al.,1973). The motion of the plasma is considered to be sufficiently slow for dynamical effects to be neglected, so that the configuration evolves through a series of quasi-equilibrium states satisfying the equation

$$\nabla p = \underline{J} \times \underline{B} \quad , \tag{1}$$

where p is the plasma pressure, $\underline{B}$ is the magnetic field and $\underline{J}$ is the current density*. The magnetic surfaces are surfaces of constant pressure and form a nested toroidal set centered on the magnetic axis. An appropriate set of coordinates for the problem is (r,θ,φ) where r and θ are polar coordinates referred to the magnetic axis of the torus which has major radius,R, and φ is the azimuthal angle measured along the magnetic axis (Fig.1). The intersections of the magnetic surfaces with a cross-sectional plane, φ = constant, are the field lines of the poloidal magnetic field, $\underline{B}_p = (B_r, B_\theta)$. If the minor radius of the torus is denoted by a and the inverse aspect ratio is defined to be $\epsilon(r=a) = a/R$, the limit of small inverse aspect ratio, $\epsilon(r=a) \to 0$, corresponds to the assumption of cylindrical symmetry. However, even for present-day experiments, usually $\epsilon(r=a) \geq 0.1$, so that toroidal effects are important.

Tokamak calculations of current interest are concerned with the

---

* The S.I. system of units is used throughout this article.

equilibrium, stability and evolution of a magnetically confined plasma. Given a two-dimensional equilibrium configuration which satisfies Eq.(1) at any given time, diffusion, heating and other entropy-generating processes will modify the plasma-field parameters and lead, by adiabatic re-adjustment to a new equilibrium configuration at a slightly later time. At each stage the configuration should be tested for stability.

At present, however, a complete two-dimensional calculation which would follow the time-evolution of the plasma-field configuration has not been undertaken. Although two-dimensional equilibrium calculations are considered routine (Feneberg and Lackner,1973; Von Hagenow and Lackner,1973) and magneto-hydrodynamic stability calculations, which were first developed in one dimension (Newcomb, 1960; Friedberg, 1970; Shafranov, 1970; Goedbloed and Hagebeuk, 1972), are now being extended to two dimensions (Bateman et al., 1974; Sykes and Wesson, 1974) with some promise of realistic three-dimensional calculations in the near future (Wooten et al.,1974), most of the evolutionary calculations for Tokamaks are still carried out in one-dimensional cylindrical geometry with the toroidal corrections modifying the transport coefficients. There are several reasons for retaining the one-dimensional model for the diffusion. First, it is not yet clear which two-dimensional set of equations should be used, since the timescale for establishing pressure and temperature equilibrium on a magnetic surface is some orders of magnitude shorter than that for transport across the surfaces. Almost certainly the standard two-dimensional, magnetohydrodynamic equations do not apply. The physical processes taking place in the Tokamak plasma are unclear, but are likely to be dominated by effects such as wall interaction, turbulent transport and the presence of impurities and neutral gas. Second, implicit, two-dimensional, numerical methods capable of handling efficiently the high Alfvén speed and long evolutionary timescale characteristic of Tokamaks are not yet available.

It seems preferable to establish a semi-quantitative agreement between computation and experiment in one-dimensional geometry before moving on to the more complex and expensive two-dimensional case. Nevertheless there exists some pioneering work on two-dimensional evolutionary calculations for Tokamaks (Winsor et al.,1970; Potter and Tuttle, 1973; Sharp and Taylor, 1973).

Since there are so many processes that may be important in Tokamaks, and since an adequate agreement between theory and experiment has not yet been achieved, it is expedient to plan any new, one-dimensional diffusion code as a versatile research apparatus into which new theoretical or empirical effects can be incorporated readily and assumptions or parameters changed with ease. The computer code, ICARUS, (Watkins et al.,1975) has been designed with due regard being paid to these features. The plasma-magnetic field configuration is described by a set of essentially conservative equations for the particle density, n, the electron and ion pressures, p_e, p_i, and the poloidal field component, B_θ. The toroidal magnetic field, B_φ, is considered to be sufficiently large to be unaffected by the plasma (low-β approximation, where β is the ratio of the thermal to the magnetic pressure) and only enters the calculation through its effect on the transport coefficients. The model is completed by the specification of the radial particle and heat fluxes and the toroidal electric field. Although the basic model employed in the code is that given by the neoclassical theory applicable to a collision-free plasma (Hazeltine et al.,1973), small changes in the input data for the code allow the use of other scaling laws such as those associated with collision-dominated plasmas (Pfirsch and Schlüter, 1962) or non-classical effects (Yoshikawa, 1970; Artsimovich, 1971).

The results of diffusion calculations are extremely dependent on the details of the ad hoc model, so it is necessary to publish together the model and the results if the literature is to maintain acceptably high standards. It is preferable that the code be available for use by other research groups so that the complex details of the physical and numerical models and the programming can be checked and the calculations confirmed and extended.

ICARUS uses the OLYMPUS programming system (Roberts, 1974; Christiansen and Roberts, 1974) which is intended to meet these requirements. Although it can be used for a wide variety of problems, OLYMPUS was originally designed for initial-value calculations of the form

$$\frac{\partial \underline{u}}{\partial t} + \underline{G}\ (\underline{u}) = \underline{0} \qquad , \qquad (2)$$

where $\underline{u}$ is the solution vector and $\underline{G}$ is a linear or non-linear operator. Many calculations in classical physics are of this kind including those for which one-dimensional Tokamak diffusion codes were developed. OLYMPUS codes are written in Standard Fortran and can be run on any computer system provided that the appropriate version of a Standard Control and Utility Package has been loaded first into the library. Packages have been developed for seven types of system so far and two have been published (Christiansen and Roberts, 1974; Hughes et al.,1975). Each code has the same underlying modular structure and is well documented so that a listing of the code is easily understood. Provision is made for ad hoc modifications for specific calculations without compromising the basic version. The first published example is the one-dimensional laser fusion code MEDUSA (Christiansen et al., 1974). This code has been used by a number of research groups who have checked and extended the test calculations and recommended a number of minor corrections (Christiansen, 1975).

The purpose of this article is to describe the basic physical model (Section II) and numerical scheme (Section III) used in ICARUS, together with the OLYMPUS programming technique (Section IV), and to indicate how these can be applied to a range of specific Tokamak calculations (Section V).

II. The Physical Model

A. INTRODUCTION

The computer code ICARUS (Watkins et al.,1975) provides the solution to a set of N partial differential equations which describe the temporal evolution of N dependent variables, $\underline{u}_n$, as functions of time (t) and a single spatial coordinate (r). The form of the equations to be solved is represented by the conservation equation in an infinitely long, cylindrically symmetric device

$$\frac{\partial \underline{u}_n}{\partial t}(r,t) + \frac{1}{r}\frac{\partial}{\partial r} r\underline{F}_n(r,t) = \underline{S}_n(r,t), \qquad n=1, \ldots N, \tag{3}$$

where $\underline{F}_n(r,t)$ and $\underline{S}_n(r,t)$ represent respectively the flux and source of $\underline{u}_n$ at each position (r,t). These functions have to be supplied explicitly in terms of $\underline{u}_n(r,t)$ and their spatial gradients. The particular applications of the code (Section V) involve the representation of $\underline{F}_n$ in the diffusion approximation so that, typically

$$\underline{F}_n \propto \frac{\partial \underline{u}_n}{\partial r} \quad . \tag{4}$$

When the equations represent predominantly diffusion processes, it is natural to solve these equations by an implicit technique (Section III).

B. THE NEOCLASSICAL TRANSPORT MODEL

The basic physical model employed in the computer code ICARUS is the neoclassical transport model of Hazeltine et al. (1973). The fluxes obtained represent averages over both the toroidal, φ, and poloidal, θ,

directions in an axisymmetric torus (Fig.1). The velocity moments of the Boltzmann equation (Chapman and Cowling, 1953) with a Fokker-Planck collision operator (Rutherford, 1970) give a series of conservation equations in which the dependent variables are functions of the single spatial coordinate, r, the distance measured from the geometrical minor axis of the torus.

The model describes a hot, axisymmetrically-confined, low β plasma consisting of hydrogen ions (mass, m_i) and electrons (mass, m_e). A strong magnetic field exists in the toroidal direction, and is specified for all time to be the Knorr field (Knorr, 1965)

$$B_\varphi = \frac{B_o}{1 - \epsilon \cos\theta} \quad , \quad \epsilon = \frac{r}{R}$$

B_o is a reference magnetic field, independent of position and time, and ϵ is the local inverse aspect ratio. An expansion of the Boltzmann equation to $O(\epsilon^{\frac{1}{2}})$ indicates the radial diffusion is ambipolar (Rosenbluth et al.,1972): the electron and ion particle fluxes, Γ_e and Γ_i, are equal

$$\Gamma_e = \Gamma_i = \Gamma$$

For scale lengths sufficiently large that the plasma may be considered to be electrically neutral, the electrons and ions are present everywhere in equal numbers and the temporal variation of the particle number density, n, is described by the equation of continuity

$$\frac{\partial n}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}\, r\Gamma = 0 \tag{5}$$

The electron and ion heat balance equations may be written in conservative form for the electron and ion energy densities, ϵ_e and ϵ_i

$$\frac{\partial}{\partial t}\,\epsilon_e + \frac{1}{r}\frac{\partial}{\partial r}\, rQ_e = E_\varphi J_\varphi - \left(\frac{\epsilon_e - \epsilon_i}{\tau_{eq}}\right) - \Gamma T_i \left(\frac{1}{n}\frac{\partial n}{\partial r} - \frac{0.172}{T_i}\,\frac{\partial T_i}{\partial r}\right), \tag{6}$$

$$\frac{\partial}{\partial t}\,\epsilon_i + \frac{1}{r}\frac{\partial}{\partial r}\, rQ_i = \left(\frac{\epsilon_e - \epsilon_i}{\tau_{eq}}\right) + \Gamma T_i \left(\frac{1}{n}\frac{\partial n}{\partial r} - \frac{0.172}{T_i}\,\frac{\partial T_i}{\partial r}\right) \tag{7}$$

The heat fluxes, Q_e and Q_i, represent both thermal and particle transport of energy. Energy input to the plasma is by virtue of the ohmic heating term, $\underline{E}.\underline{J}$, which involves only the toroidal electric field, E_φ, for the Tokamak ordering procedure in which the poloidal magnetic field, B_θ, is assumed to be small in comparison with the externally applied toroidal magnetic field, B_φ. The energy is transferred from electrons to ions by virtue of the equipartition term characterized by the time

$$\tau_{eq} = \frac{1}{2} \frac{m_i}{m_e} \tau_{ei} \quad ,$$

where τ_{ei} is the electron-ion collision time given by Braginskii (1965)

$$\tau_{ei} = \frac{3\, m_e^{\frac{1}{2}} (kT_e)^{\frac{3}{2}} (4\pi\epsilon_o)^2}{4\sqrt{2\pi}\, n\, e^4\, \log_e \Lambda} \quad .$$

(k is the Boltzmann constant; ϵ_o is the permittivity of free space; e is the charge on the electron; T_e is the electron temperature).

The Coulomb logarithmic function, $\log_e \Lambda$, is (Braginskii,1965)

$$\log_e \Lambda = \begin{cases} 30.3 - 1.15 \log_{10} n + 3.45 \log_{10} \left(\frac{kT_e}{e}\right), \frac{kT_e}{e} < 50 \text{ eV} \\ 32.2 - 1.15 \log_{10} n + 2.3 \log_{10} \left(\frac{kT_e}{e}\right), \frac{kT_e}{e} > 50 \text{ eV} \end{cases} .$$

The electron and ion energy equations are coupled also by the last term on the right-hand sides of Eqs.(6)-(7) representing ion motion along the magnetic field.

Equations (6)-(7) may be used to describe the temporal evolution of the electron and ion pressures, p_e and p_i, by means of equations of state of the form applicable to a perfect gas

$$\epsilon_e = \frac{p_e}{(\gamma-1)} = \frac{nk\, T_e}{(\gamma-1)} \quad , \quad \epsilon_i = \frac{p_i}{(\gamma-1)} = \frac{nk\, T_i}{(\gamma-1)} \quad , \qquad (8)$$

where γ is the ratio of the specific heats, assumed to be $5/3$, and T_i is the ion temperature.

The toroidal current density, J_φ, may be determined from Maxwell's form of Ampere's law

$$\mu_o J_\varphi = \frac{1}{r} \frac{\partial}{\partial r} r B_\theta \quad , \tag{9}$$

(μ_o is the permeability of free space)

with the poloidal magnetic field being given by Faraday's Law

$$\frac{\partial B_\theta}{\partial t} = \frac{\partial}{\partial r} E_\varphi \quad . \tag{10}$$

Consistent with the Tokamak ordering procedure it is valid to consider changes only in the poloidal magnetic field, B_θ.

The plasma is described by the three dependent variables, n, p_e, p_i, representing respectively the particle number density and electron and ion pressures. The conservation equations for these quantities are closed by the specification of the radial particle flux, the radial electron and ion heat fluxes and the toroidal electric field applicable to the diffusion of plasma in a torus (Galeev and Sagdeev, 1968; Kadomtsev and Pogutse, 1971). In such a system a charged particle may execute three types of nearly periodic motions. These are gyration about a magnetic field line, motion along the field line resulting in transit around the system or bouncing between mirror points, and drifting across the field lines in response to electric fields and the gradient and curvature of the magnetic field. The relation between the frequencies associated with these motions and the frequency for electron-ion collisions, $\nu_{ei} = \tau_{ei}^{-1}$, determines the exact form of the diffusion coefficient, which is given, in general, by an equation of the form

$$D \sim \nu \ell^2 \quad , \tag{11}$$

where ν is the effective collision frequency and ℓ is a characteristic step size for a random walk process. The relative forms of the diffusion coefficients in the high-frequency ("classical"), intermediate-frequency ("plateau") and low-frequency ("banana") regimes are summarized in Fig.2. The frequencies that separate these regimes are representative of the

bounce frequency, ν_b, of trapped particles and the transit frequency, ν_t, of untrapped particles.

The expressions for the fluxes in the low frequency regime are given most concisely by Hazeltine et al. (1973). Corresponding to the forces, $A_m (m = 1, 2, 3)$ which result from gradients in the particle density, electron and ion temperatures and the electric field there exist three fluxes, $J_l (l = 1, 2, 3)$

$$A_1 = A1/p_e = p_e^{-1} \frac{\partial(p_e + p_i)}{\partial r} - \frac{5}{2} T_e^{-1} \frac{\partial T_e}{\partial r} + (y - \frac{5}{2}) T_e^{-1} \frac{\partial T_i}{\partial r}$$

$$A_2 = \frac{A2}{kT_e p_e} = k^{-1} T_e^{-2} \frac{\partial T_e}{\partial r}$$

$$A_3 = A3 = E_\varphi \qquad (12)$$

$$J_1 \equiv \Gamma$$

$$J_2 \equiv Q_e$$

$$J_3 \equiv (kT_e)^{-1}(J_\varphi - \sigma_s E_\varphi) \quad .$$

The fluxes and forces are related through the transport coefficients, L_{lm}

$$J_l = - \sum_{m=1}^{3} L_{lm} A_m \quad , \quad l = 1, 2, 3.$$

In addition, the total ion heat flux is given by

$$J_i = y \Gamma k T_i - L_i k^{-1} T_i^{-2} \frac{\partial T_i}{\partial r} \quad . \qquad (13)$$

The parameter, y, has been introduced so that A_2 is dependent only on the gradient of the electron temperature. To within a reasonable degree of accuracy the transport coefficients, L_{lm}, are given by the simple ansatz (Hazeltine et al.,1973)

$$L_{lm} = \alpha_{lm} \epsilon^{\frac{1}{2}} + \beta_{lm} \epsilon \quad . \qquad (14)$$

This gives exact results in the limits $\epsilon \to 0$, $\epsilon \to 1$. In the code ICARUS it has been found most useful to provide the forces, Am, (m = 1, 2, 3) and the corresponding transport coefficients, RL1m, (1, m = 1, 2, 3) in terms of the

classical diffusion coefficient in the poloidal magnetic field, D_θ^c, and the Spitzer conductivity, σ_s, defined according to

$$D_\theta^c = \rho_{e\theta}^2 \, \nu_{ei} \quad , \quad \sigma_s = ne^2/m_e \, \nu_{ei} \quad , \tag{15}$$

where the electron gyro radius, $\rho_{e\theta}$, in the poloidal magnetic field is given by

$$\rho_{e\theta}^2 = \frac{2m_e \, kT_e}{e^2 B_\theta^2} \quad . \tag{16}$$

The following definitions for the transport coefficients apply

$$\begin{aligned}
RL11 &= \frac{D_\theta^c}{kT_e} \left(CA11 \, \epsilon^{\frac{1}{2}} + CB11 \, \epsilon + CC11\right) \\
RL12 &= \frac{RL21}{kT_e} = \frac{D_\theta^c}{kT_e} \left(CA12 \, \epsilon^{\frac{1}{2}} + CB12 \, \epsilon + CC12\right) \\
RL13 &= -nRL31 = \frac{n}{B_\theta} \left(CA13 \, \epsilon^{\frac{1}{2}} + CB13 \, \epsilon + CC13\right) \\
RL22 &= D_\theta^c \left(CA22 \, \epsilon^{\frac{1}{2}} + CB22 \, \epsilon + CC22\right) \\
RL23 &= -p_e \, RL32 = \frac{p_e}{B_\theta} \left(CA23 \, \epsilon^{\frac{1}{2}} + CB23 \, \epsilon + CC23\right) \\
RL33 &= \sigma_s \left(CA33 \, \epsilon^{\frac{1}{2}} + CB33 \, \epsilon + CC33\right) \\
RLI &= \left(\frac{m_i \, T_e}{m_e \, T_i}\right)^{\frac{1}{2}} D_\theta^c \left(CAI \, \epsilon^{\frac{1}{2}} + CBI \, \epsilon + CCI\right) \\
y &= CAY \, \epsilon^{\frac{1}{2}} + CBY \, \epsilon + CCY \quad .
\end{aligned} \tag{17}$$

The numerical values of the constants (CA11, ...,CCY) employed in the definitions of the transport coefficients are summarized in Table I. The range of validity of the transport expressions may be extended to the intermediate frequency ("plateau") regime with the aid of "smoothing functions" (Table II). These functions effect a smooth transition of the diffusion coefficients as the collision frequency increases to a value

greater than the bounce frequency, ν_b (Fig.2).

The transport model applicable to low and intermediate values of the collision frequency forms the basic physical model employed in ICARUS.

C. BOUNDARY AND INITIAL CONDITIONS

The code employs a finite-difference mesh that is mapped onto a continuous domain (Section III) in such a way that the physical boundaries of a wall (or limiter), at radius r = a, and the minor axis at radius r = o, coincide with mesh-cell boundaries rather than mesh-cell centers at which the dependent variables are defined. The physical boundary conditions apply to the fluxes, Γ, J_φ, Q_e, Q_i. On the minor axis of the torus at radius, r = o, all radial components of vector quantities must be zero, with the result that the gradients of the dependent variables, n, p_e, p_i, must also be zero. On the other hand the toroidal current density, J_φ, should exist and can be related in general to the poloidal magnetic field, B_θ, at a position r_o (near r=o) by Ampere's Law applied to a cylinder about the axis

$$J_\varphi = \frac{2B_\theta}{\mu_o r_o} \quad .$$

The boundary conditions at the outer radius, r = a, are more difficult to treat. The simplest method is to employ the use of "pedestal" values for the dependent variables at the first mesh point with a radius, r > a : the boundary values of the dependent variables are set to a fixed fraction of the central values at time, t = o, by specifying the functional forms

$$f = f_o \left(1 - (1-P)x^l\right)^m \quad , \quad f \equiv \{n, T_e, T_i\} \quad ,$$

$$J_\varphi = J_o \left(1 - x^l\right)^m \quad , \quad x = r/a \quad .$$

The indices $\{l,m\}$ are preset to values applicable to parabolic distributions and the fractional pedestals, P, are set to 10%. These may be modified through data input, as also may the maximum values, f_o, and the

total toroidal current

$$I_\varphi = 2\pi \int_0^a J_\varphi \, r dr \quad .$$

The pedestal values are maintained at the first mesh point with $r > a$ for all time.

III. The Numerical Model

A. INTRODUCTION TO ONE-DIMENSIONAL FINITE-DIFFERENCE METHODS

The equations to be solved may be written concisely in the form of Eq.(2) with $\underline{u}(r,t)$ being the column vector $\{n, p_e, p_i, B_\theta\}$ and $\underline{G}$ being an operator dependent on $\underline{u}$ and its spatial derivatives. The problem formulated by Eq.(2) is an initial-value problem so that $\underline{u}$ is determined for all time given its initial-value, $\underline{u}(r,o)$, together with the specification at all times of the spatial boundary conditions.

In general, it is not possible to express the solutions to Eq.(2) in terms of known functions. Numerical methods rather than analytical methods have to be employed. The vector, $\underline{u}$, and the operator, $\underline{G}$, are defined on the mesh points of a grid (Fig.3) mapped onto a continuous domain bounded by the positions, $r = o$ and $r = a$. Temporal differencing assumes a discrete set of lattice points along the time axis.

The general difference analogue of Eq.(2) is obtained by integration

$$\underline{u}^{n+1} - \underline{u}^n = - \int_{t^n}^{t^{n+1}=t^n + \Delta t^n} \underline{G}\,(\underline{u}) dt \quad . \tag{18}$$

The solution to this equation may be approximated by replacing the right-hand side of Eq.(18) by some temporal average

$$\underline{u}^{n+1} - \underline{u}^n = - \left[\zeta \, \underline{G}(\underline{u}^{n+1}) + (1-\zeta) \, \underline{G}(\underline{u}^n) \right] \Delta t^n , \tag{19}$$

where

$$0 \leq \zeta \leq 1 \quad .$$

When $\underline{G}$ is of such a form that Eq.(2) is parabolic in (r,t), fully implicit methods of solution ($\zeta \equiv 1$) are particularly effective (Richtmyer and Morton, 1967).

B. THE METHOD OF SOLUTION EMPLOYED IN ICARUS

In the diffusion approximation the model equations which can be written to conform with Eq.(3) contain fluxes, $\underline{F}$, and source terms, $\underline{S}$, that can be functions of the dependent variable $\underline{u}$

$$\underline{F}(\underline{u}) = f_{\underline{u}}(\underline{u}) \frac{\partial}{\partial r} \underline{u} \quad ,$$

$$\underline{S}(\underline{u}) = s_{\underline{u}}(\underline{u}) \underline{u} \quad .$$

On the finite-difference mesh the values of $\underline{u}^{n+1}$ at the time, t^{n+1}, and positions, $r = r_{j-1}, r_j, r_{j+1}$, may be related by the matrix equation

$$\left[-\underline{\underline{A}}_j \cdot \underline{u}_{j+1} + \underline{\underline{B}}_j \cdot \underline{u}_j - \underline{\underline{C}}_j \cdot \underline{u}_{j-1} \right]^{n+1} = \underline{D}_j^{\,n} \quad , \qquad (20)$$

where $\underline{\underline{A}}$, $\underline{\underline{B}}$, and $\underline{\underline{C}}$ are 4×4 matrices and $\underline{D}$ is a 4-vector. Equation (20) is solved as a linear system by means of the algorithm given by Richtmyer and Morton (1967)

$$\underline{u}_j^{n+1} = \underline{\underline{G}}_j^{n+1} \cdot \underline{u}_{j+1}^{n+1} + \underline{H}_j^{n+1} \quad , \quad j = N-1, N-2 \ldots\ldots 2, \qquad (21)$$

where the matrix, $\underline{\underline{G}}$, and the vector, $\underline{H}$, are determined from the recurrence relations

$$\underline{\underline{G}}_j^{n+1} = \left(\underline{\underline{B}}_j^{n+1} - \underline{\underline{C}}_j^{n+1} \cdot \underline{\underline{G}}_{j-1}^{n+1} \right)^{-1} \cdot \underline{\underline{A}}_j^{n+1} \quad , \qquad (22)$$

$$\underline{H}_j^{n+1} = \left(\underline{\underline{B}}_j^{n+1} - \underline{\underline{C}}_j^{n+1} \cdot \underline{\underline{G}}_{j-1}^{n+1} \right) \cdot \left(\underline{D}_j^{n} + \underline{\underline{C}}_j^{n+1} \cdot \underline{H}_{j-1}^{n+1} \right) \quad . \qquad (23)$$

The calculation involves inverting a 4×4 matrix at each mesh point. Given the boundary conditions in finite-difference form this can be performed easily (Potter, 1973).

As a result of the non-linear nature of Eq.(20) with the coefficients of $\underline{u}^{n+1}$ being functions of $\underline{u}^{n+1}$ it is necessary to solve Eq.(20) iteratively in order to obtain accurate values for the coefficients. Initially, $\underline{\underline{A}}^n, \underline{\underline{B}}^n$ and $\underline{\underline{C}}^n$ are evaluated at time, t^n, and a trial solution, $\underline{u}^{(1)}$ is obtained. The iteration is repeated with $\underline{\underline{A}}^{(1)}, \underline{\underline{B}}^{(1)}, \underline{\underline{C}}^{(1)}$ to obtain $\underline{u}^{(2)}$ and is continued using $\underline{\underline{A}}^{(p-1)}$, $\underline{\underline{B}}^{(p-1)}$, $\underline{\underline{C}}^{(p-1)}$ to determine $\underline{u}^{(p)}$ until the solution approximates $\underline{u}^{n+1}$ to within the desired accuracy. Convergence of the iterative procedure is checked by imposing that the fractional change in $\underline{u}$ at the p^{th} and $(p-1)^{th}$ iteration be less than the specified convergence tolerance, δu :

$$\text{Max} \left\{ \frac{\left| \left| \underline{u}_j^{(p)} \right| - \left| \underline{u}_j^{(p-1)} \right| \right|}{\left| \underline{u}_j^{(p)} \right| + \left| \underline{u}_j^{(p-1)} \right|} \right\} \leq \delta u \quad , \text{ for all } \quad 1 < j < N \quad .$$

If the relative deviations are below specified values, convergence is assumed to be established. If convergence is not achieved within a fixed number of iterations the complete timestep is repeated with a reduced value for the timestep. At the completion of a timestep the conservation of particle number and total energy is checked. Since the total energy is not a variable that is involved in the Eqs.(5)-(10), this check provides a very sensitive guide to the behaviour of the numerical procedure.

The implicit method of solving parabolic equations guarantees a numerically stable solution (Richtmyer and Morton, 1967) and the choice of the timestep only serves to determine the accuracy. To monitor the temporal variation of $\underline{u}$, the timestep is restricted according to

$$\Delta t^n \leq a \min \left\{ \frac{\left| \underline{u}_j^{n+1} \right| + \left| \underline{u}_j^{n} \right|}{\left| \left| \underline{u}_j^{n+1} \right| - \left| \underline{u}_j^{n} \right| \right|} \right\} \Delta t^{n-1} \quad , \quad \text{for all} \quad 1 < j < N \quad .$$

The variation in the timestep is restricted by

$$\frac{1}{a_o} \leq \frac{\Delta t^n}{\Delta t^{n-1}} \leq a_o \quad ,$$

where a and a_o are constants $O(1)$.

IV. Programming Techniques

A. THE OLYMPUS SYSTEM

The purpose of the OLYMPUS programming system (Roberts, 1974 ; Christiansen and Roberts, 1974) is to establish a clear standard structure for Fortran programs of a similar kind. The first application is to the generalized initial-value problem (Eq.(2)) which predicts the evolutionary behaviour of a physical system. Initial-value problems are familiar in many areas of classical physics and the Tokamak diffusion calculation for which ICARUS has been designed provides a typical example. The concepts on which OLYMPUS is based are adapted from disciplines such as mathematics, theoretical physics, engineering and architecture and from the field of scientific publishing (Roberts, 1969 and 1971). It is intended that the programs together with their write-ups should be published in tested and refereed form in journals such as Computer Physics Communications and the associated CPC Program Library (Burke, 1970). This has the advantage of making the programs available for general use and criticism just like other scientific publications. Particular attention is paid therefore to methods of documentation and layout, the eventual aim being to reach a level of program intelligibility similar to that expected of a mathematical textbook (Roberts, 1969).

Standardization is practicable because all programs that solve problems of type (2) have many similar duties to perform, irrespective of the specific equations that are used. Once-and-for-all decisions can be made concerning questions such as overall program structure, control techniques, terminology, notation and layout, and a powerful library of general-purpose routines can be provided to deal with the frequently-encountered housekeeping duties. Automatic techniques for program generation also

become possible. All this can relieve the individual programmer of a considerable amount of effort provided that the system has been sufficiently developed and the programmer has understood how to use it. The programmer's work is made more accessible to other people since they know what to expect.

Similar ideas have been accepted in mathematics and theoretical physics for a long time and are known to be very powerful. Typical examples are the abstract formalisms of group theory and of Hamiltonian dynamics. One of the aims of the OLYMPUS system is to translate these ideas, so far as is possible, into programming practice. The Hamiltonian analogy is particularly useful since it specifically applies to equations of type (2).

Each OLYMPUS physics program consists of a set of subprograms written in Standard Fortran (National Computing Centre, 1970) and running under the supervision of a universal main program and an associated control routine, COTROL, which form part of a Standard Control and Utility Package (Christiansen and Roberts, 1974). This package is contained in a local system library and a version is provided for each different type of computer system on which OLYMPUS has been implemented (currently, OLYMPUS has been installed on seven types of computer system). The versions are written mostly in Fortran and are similar. However, they need to take account of system-dependent features such as word and byte length, channel conventions and supervisor calls. Segregation within the OLYMPUS package of system-dependent features makes it possible to transfer large programs from one type of computer system to another with only minor changes, provided that the recommended conventions have been followed.

The control routine COTROL described by Christiansen and Roberts (1974) expects the programmer to supply a set of primary subprograms with the names and functions defined in Table III: standardization of the nomenclature and overall structure is ensured. These primary subprograms can call secondary subprograms whose names and functions are chosen by the programmer to reflect the type of calculation performed (Table IV and Fig.4). Dummy versions of the primary subprograms are provided within the package as part of the OLYMPUS library and are useful during program development. Taken in their entirety they constitute a dummy program called CRONUS that has the same structure as other members of the OLYMPUS family but solves a null set of equations. An OLYMPUS program package contains also a number of utility and diagnostic routines in binary form that may be regarded as an extension of the library of Standard Fortran Functions (National Computer Centre, 1970), a set of standard COMMON blocks, a preprocessor program (where this is not already provided by the computer system itself (Hughes et al.,1975)), test programs and data, and documentation files.

B. PROGRAM STRUCTURE

Four aspects of program structure are taken into account in the design of an OLYMPUS program, namely architecture, dynamics, control and diagnostics.

It is often convenient to exploit a mechanical or electrical analogy in which the program (software) is compared to a real physical machine (hardware).

Architecture is a static concept and is concerned with the underlying plan on which the program is based. In Fortran this plan is the division of the program into subprograms and COMMON blocks, each of which has a

specific duty to perform. This corresponds to the design of a real machine as represented by a set of blueprints or a stationary model. The architecture of OLYMPUS programs is developed further in Section IV, C.

Dynamics is concerned with how the program actually works. It is useful to exploit the analogy with Hamiltonian dynamics in which there exists the concept of the state of a dynamical system changing with time according to certain laws; the state is represented by a set of coordinates and momenta ($\underline{q}$,$\underline{p}$) and the laws of motion by the Hamiltonian, H. In the case of a program it is necessary to have a clear understanding of the amount of information needed to determine the "state of the computation" in order that the program may be checked out thoroughly. The dynamics of OLYMPUS programs will not be discussed in this article, except to remark that it is simplified considerably by the choice of Standard Fortran (National Computer Centre, 1970) which excludes ENTRY and RETURN i statements, so making subroutine jumps easier to follow. The dynamics of programs written in extended versions of Fortran, in Algol 60 and 68, in PL/I and especially in assembly language can become progressively more difficult to understand unless some extra restrictions are accepted by the programmer.

Control is concerned with the operation of the program by the user, either for routine calculations or for those requiring ad hoc modifications. Three standard techniques which are used in OLYMPUS programs will be described in Section IV, E.

Diagnostics may be compared to the measurements or observations that are made on a physical mechanism or electrical apparatus in order to monitor its working and to check that it is performing correctly. Usually these measurements do not disturb the working of the system, and the same

criterion should be applied to the diagnostics in a computer program (although there will result a slight reduction in running speed and a small increase in the storage requirements in order to accommodate the diagnostic routines themselves). The diagnostic facilities of a program are required to monitor

(a) The working of the program

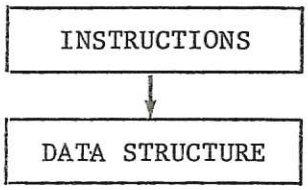
(b) The validity of the numerical scheme

(c) The behaviour of the physical model which the calculation is describing.

The OLYMPUS system per se deals only with (a).

C. ARCHITECTURE

Every OLYMPUS program has a similar architecture which consists of two main parts



The instructions are represented by the Fortran subprograms which are currently organized into seven classes

0	Control	4	Epilogue
1	Prologue	5	Diagnostics
2	Calculation	U	Utilities
3	Output		

Each subprogram is assigned a decimal number ⟨m.n⟩ as indicated in Table III. The control class 0 is common to all programs as also is the utility class U and most of the diagnostic class 5. The subprograms of classes 1-4, which represent respectively the initialization, calculation, output and termination operations of the code, will differ from one program to another

although the names of the primary subprograms must always be present since these are called by the permanent control subprogram ⟨0.3⟩ COTROL. As a real program is developed, working subprograms replace the dummy versions otherwise loaded from a library. In this way a program is constructed in a standard and methodical way enabling sections of the program to be tested independently.

The data structure is represented by labelled COMMON blocks which are organized into five groups

1. General OLYMPUS data
2. Physical problem
3. Numerical scheme
4. Housekeeping
5. Input, output and diagnostics.

These groups are subdivided into decimally numbered blocks [Cr.s] as indicated in Table V. Blank COMMON is not normally used but has the label [C9.0]. It is useful to visualize the COMMON blocks as operands and the subprograms as operators which act on the data and change the values of the variables and arrays in the same way as the Hamiltonian, H, acts on the coordinates and momenta ($\underline{q}$,$\underline{p}$) in dynamics. In order to take advantage of this simple picture all information relevant to the "state of the calculation" is stored in COMMON rather than in local storage, and is transmitted via COMMON rather than through subprogram argument lists. This is easier to understand and has the advantage that the diagnostic routines are able to examine COMMON and provide a "snapshot" of the current state. Alphanumeric indices and subprogram maps are used to make the architecture as clear as possible.

Precautions are taken to avoid the duplication of non-local names by the use of different initial letters to denote variables and arrays of different status (COMMON, local, loop index, formal parameter). Only one copy of each COMMON block is used throughout the program. This is inserted automatically by means of a file substitution facility. A preprocessor control statement such as

// SUBSTITUTE COMTOK

for the ICL 4/70 or IBM 360/370 computers results in the named file being inserted in place of the control card. Corresponding facilities are built into the CDC and Univac systems.

The COMMON blocks in Group 1 are intended to be standard library versions which are available to all programs. So far two have been published, namely [C1.1] COMBAS which contains basic system parameters and [C1.9] COMDDP which contains parameters used during the development and checking of a program. Other blocks will contain fundamental physical constants, character codes and other general-purpose information.

The physics of the step-by-step calculation is controlled always by ⟨2.1⟩ SUBROUTINE STEPON, which in turn calls other Class 2 subprograms to perform the actual calculation as illustrated in Fig.4 and Table IV. There is considerable advantage in separating this part from the rest of the program, which is concerned mainly with practical matters (although ⟨1.6⟩ INITAL sets up the physical initial conditions).

D. INITIALIZATION

Experience indicates that in order to make a program easy to develop, to understand and to use, it is important to initialize the data structure in a logical way. Although programmers often employ a sequence of data cards which may have quite complex and varied formats and are not necessarily all read at the same position in the program, it is possible

to use in most cases, a much simpler scheme which requires only a single READ statement

```
READ(NREAD,NEWRUN)                    .                    (24)
```

Statement (24) is the same for all OLYMPUS programs: NREAD is the input channel number, and NEWRUN is the name of a NAMELIST containing a list of all variables and arrays whose default values may require to be changed.

Data initialization is the main purpose of the Prologue (Class 1) which functions as follows: variables and arrays are cleared to zero in

⟨1.2⟩ CLEAR prior to a sequence of three subroutines being entered

⟨1.3⟩ PRESET

⟨1.4⟩ DATA

⟨1.5⟩ AUXVAL .

PRESET assigns suitable default values to all those variables which can be independently set. For clarity this is performed block-by-block with the names arranged in alphanumeric order. DATA is called in order to change any of these default values, and finally AUXVAL computes the values of auxiliary variables which depend on the input data, such as the determination of

```
NM1 = N-1                    ,
```

where N is the number of mesh points. This procedure is organized block-by-block where practicable.

NAMELIST data input has the effect of overwriting the values of variables and array elements that are referenced by name and leaving all others unchanged. This facility is flexible, concise and symbolic so that it is well-adapted to on-line working. The input file can be kept quite short if appropriate default values are chosen, as illustrated in Table VI.

⟨1.6⟩ INITAL is intended to provide the physical initial conditions for the calculation. Frequently this is performed best by calling function

subprograms such as FRHO (density), FTE (electron temperature) which contain generalized functions whose parameters are acceptable in the NAMELIST NEWRUN. ⟨1.8⟩ START performs any housekeeping duties necessary before the run can begin.

E. CONTROL

Three standard methods are used to control an OLYMPUS calculation

1. NAMELIST data
2. Modification of the initial functions
3. The EXPERT facility.

Method 1 has been explained in Section IV,D, while Method 2 simply requires that a new load module be generated with a different choice of one or more of the initial functions FRHO, FTE etc. Usually this means copying the source file and changing a few well-identified statements using an editor facility. Method 3 is more powerful, and its purpose is to enable extensive ad hoc modifications to be made to an existing source program without compromising the original version, which may have been carefully tested and which may be in current use by other people.

At appropriate points throughout an OLYMPUS program calls are inserted of the form

```
CALL EXPERT(ICLASS,ISUB,IPOINT)  .                    (25)
```

The formal parameters, ICLASS and ISUB, are local variables defining the decimal code number ⟨ICLASS.ISUB⟩ of the subprogram from which the call is made, and IPOINT is an integer which specifies the precise point within that subprogram. These calls are not placed of course inside inner loops.

Subroutine ⟨0.4⟩ EXPERT(KCLASS,KSUB,KPOINT) is contained in skeleton form in the OLYMPUS library and is used to control the diagnostics, but the user is free to provide his own version which can be used to insert ad hoc sections of coding. If an insertion is to be made at the point (2.13.3)

(and possibly also at other points) a convenient technique is to employ the statements

```
      ICODE=10000*KCLASS+100*KSUB+KPOINT
      .....
      IF(ICODE.EQ.21303)GO TO 21303
      .....
      RETURN
21303 CONTINUE
      [Additional coding for point (2,13,3)]
      RETURN
      .....
      END
```

⟨0.4⟩ EXPERT has access to all COMMON variables and arrays (provided that the corresponding // SUBSTITUTE statements are included) and it can call subprograms itself, if necessary. However, it must not call any subprograms that already contain statements of the form (25). This difficulty can be avoided by including a switch of the form

```
IF(NLEXPT) CALL EXPERT (ICLASS,ISUB,IPOINT)      .      (26)
```

This is turned off as soon as EXPERT is entered and restored to its original setting as soon as the return is made.

Provided that EXPERT is arranged in a logical way, a listing of this file together with that of any additional subprograms constitutes a generalized form of data input and sufficiently defines the ad hoc modifications that have been made.

A further technique that can be used is to maintain a master version of EXPERT on-line in a file called MASTER, which also contains all the control statements needed to compile EXPERT, to link-edit or compose the new version of the program and to run the job. The line numbers in this

file are coordinated with the decimal numbering scheme used by OLYMPUS so that it is easy to incorporate changes without making mistakes. The ICL 4/70 Multijob Editor uses a command of the form

```
EDIT B/C ,,,,A
```

in order to edit corrections contained in file A into file B to produce a new file C. The files A and B and the line numbers are left unchanged. In order to insert ad hoc modifications and to run a new case commands of the following form need to be typed

```
[Edit file MOD2]
EDIT MASTER/CASE2,,,, MOD2
REMJOB CASE2(S)
```

The job will be queued in the appropriate remote batch stream and executed. Alternatively it could be executed on-line.

F. DIAGNOSTICS

Facilities are provided by OLYMPUS for tracing the flow of a program, for switching-off selected subprograms, for printing messages, and for displaying the contents either of selected COMMON blocks or of individual variables or arrays. Diagnostic output is arranged in convenient form and can largely be controlled by switches set in NAMELIST data input. For example, the input line

```
NLREPT=T,NPDUMP=20701,NVDUMP=100,
```

would cause all COMMON variables (but not arrays) to be displayed at the point (2,7,1), arranged decimally by blocks and alphanumerically within the blocks in the form

```
⟨name⟩ = ⟨value⟩
```

This output forms a properly organized index to the current state of the calculation. Other switch settings can be used to select only certain of

the blocks, or to display the arrays.

If the local system supports the facility, certain classes of error cause the supervisor to re-enter the program at a point where the Fortran statements

```
CALL CLIST(0,0)
CALL ARRAYS(0,0)
```

are used to generate an index for all COMMON variables and arrays. This is preferable to the octal or hexadecimal dump that is provided by most computer systems.

These facilities are useful during program check-out: the switches can be visualized as part of an "engineer's test panel" and calls of the form (26) as the "cables" which connect the panel to the program under test. The OLYMPUS utility routines are also useful during ad hoc modifications, since, for example, the statement

```
CALL RARRAY(8HBTHETA  ,BTHETA,MAXMSH)
```

in EXPERT will output the array containing the values of the poloidal magnetic field, B_θ, without the need to construct FORMAT statements in the usual way.

G. DOCUMENTATION

OLYMPUS programs together with their documentation are arranged in a standard format as illustrated in Fig.5. This format is based on that of a mathematical textbook and is intended to be as easy to follow as is possible. The subprograms are divided into decimally numbered sections and subsections each of which has an appropriate heading. Indentation is used to improve the layout. In Fig. 5 the comments have been printed in lower case on a General Electric Termi Net 300 to make them stand out from the statements. It is planned to publish a number of programs in this form for training purposes.

V. Applications

A. INTRODUCTION

Recent publications (Dimock et al.,1973; Hinnov et al.,1973; Berry et al.,1974) on the behaviour of Tokamak plasmas in a variety of devices have provided a detailed description of the temporal behaviour of the discharge. In particular, detailed measurements of not only the ohmic heating current and voltage, but also the radial profiles of the electron temperature and density have been obtained.

The predictions of neoclassical theory can be examined and are found to be incorrect. It is therefore of paramount importance to be able to omit selectively parts of the model, and to replace complete sections of the model. The development of the physical model to one which simulates well the behaviour of an ST discharge (Dimock et al.,1973) is monitored by the method of introduction of additional flux and source terms into the code ICARUS. Physical effects that may be present in the next generation of Tokamaks can be incorporated readily into the code.

B. PRESENT GENERATION TOKAMAKS

1. The results of neoclassical simulations

The various codes which solve the full set of neoclassical equations give essentially the same results. In Fig.6 the radial profiles of particle density, n, electron and ion temperatures, T_e and T_i, as predicted by the Düchs code (Hinton et al., 1972) and ICARUS (Watkins et al., 1975) are plotted for conditions applicable to a low density discharge in a device such as ST at a time of 60 ms.

The codes predict narrow, almost linear, density profiles coupled with much broader thermal distributions. The density distribution results from the dominance in the neoclassical equations of the Ware pinch effect

(Ware,1970), which depletes of plasma the outer regions of the discharge while the density near the axis increases. Since the temperature gradients exist only near the edge where the density becomes low, the thermal losses are also low.

However, the numerical predictions for existing Tokamaks are not in accordance with experimental results. For conditions applicable to a high density discharge in a device such as ST (Dimock et al.,1973) the experimental profiles at 35 ms are very different from those predicted numerically (Fig.7). As a result the energy balance is found experimentally to be dominated by electron thermal conduction and recycling of particles rather than by ion thermal conduction and radiation as is predicted numerically. The experimental replacement times for the particle density, τ_p , electron and ion energies, τ_{Ee}, τ_{Ei}, and total energy, τ_E', are at least an order of magnitude smaller than those obtained numerically (Table VII).

A further deficiency of neoclassical theory is its inability to describe the current rise phase in Tokamaks. In this case, if the transport were purely neoclassical the skin current would persist for extremely long times (Düchs et al.,1971). This is not observed experimentally although the ST discharge (Dimock et al.,1973) appears to exhibit a mild skin effect which is probably not negligible for the purposes of a detailed analysis of the energy balance, but which reaches a saturation level. It is found (Rosenbluth and Kaufman, 1958) that in a vacuum region the magnetohydrodynamic equations break down as a result of the joule heating term (which is independent of number density) being shared between fewer and fewer particles with the result that a singularity in the electron temperature can be produced. In the case of the neoclassical equations, Furth et al.(1970) have indicated that the skin effect during the rise of the plasma current could

result in the excitation by surface heating of an unstable thermal mode which will then augment and perpetuate the skin effect.

To circumvent this difficulty, the computer codes usually run with the current initially distributed throughout the plasma volume and with the total current maintained constant for all time. Nevertheless, skin currents can result if too low a value for the density pedestal is maintained (Fig.8).

2. The Development of the Physical Model

The model of a Tokamak plasma as formulated by the neoclassical theory gives the discharge a temporal behaviour which relates very little to an experimental discharge. It is necessary to be able to omit or replace easily sections of the model. This implies that the solution be determined in such a way that the physical model is separated from the numerical model in the sense that changes in the physical model can be undertaken readily and will be transmitted automatically in the correct form to the numerical model. An interface may be constructed which will transform the flux, $\underline{F}$, and source, $\underline{S}$, terms supplied by a user as functions of the dependent variables and of their spatial gradients into the block-tridiagonal, matrix representation of the finite-difference equations.

The fluxes and source terms are determined respectively as coefficients, $FX\alpha\beta$ and $SC\alpha\beta$ so that Eq.(3) is assumed to be of the form

$$\frac{\partial \alpha}{\partial t} + \frac{1}{r}\,\frac{\partial}{\partial r}\left(r \sum_{\beta} FX\alpha\beta\,\frac{\partial \beta}{\partial r}\right) = \sum_{\beta} \beta SC\alpha\beta \quad , \quad \alpha,\beta \equiv \{n, p_e, p_i, B_\theta\} \; . \qquad (27)$$

In general, for a system of N-equations in N-dependent variables it would be most advantageous to define the coefficients as two-dimensional arrays $FX(\alpha,\beta)$, $SC(\alpha,\beta)$ so that looping over the indices 1-N may be used to introduce even further independence.

3. Additional Flux Terms

a. Pseudo-classical diffusion. The pseudo-classical scaling laws (Yoshikawa, 1970; Artsimovich, 1971) may be incorporated into the code ICARUS in a number of different ways which highlight the versatility of the OLYMPUS system. The definitions for the fluxes of particle density, electron and ion thermal energy densities and current density are defined according to

$$
\begin{aligned}
\Gamma &= nV_r = -K\, D_\theta^c \left(1 + \frac{T_i}{T_e}\right) \frac{\partial n}{\partial r} \\
\tilde{Q}_e &= -K\, D_\theta^c \left(1 + \frac{T_i}{T_e}\right) \left(\frac{\partial p_e}{\partial r} - kT_e \frac{\partial n}{\partial r}\right) \\
\tilde{Q}_i &= -\left(\frac{m_i}{m_e}\frac{T_e}{T_i}\right)^{\frac{1}{2}} D_\theta^c \left(\frac{\partial p_i}{\partial r} - kT_i \frac{\partial n}{\partial r}\right)(CAI\ \epsilon^{\frac{1}{2}} + CBI\ \epsilon + CCI) \\
Q_e &= \tilde{Q}_e + \tfrac{3}{2}\,\Gamma\, kT_e \\
Q_i &= \tilde{Q}_i + \tfrac{3}{2}\,\Gamma\, kT_i \\
J_\varphi &= K^{-1}\,\sigma_s\,(CA33\,\epsilon^{\frac{1}{2}} + CB33\,\epsilon + CC33)E_\varphi \quad ,
\end{aligned}
\tag{28}
$$

where (Spitzer, 1962)

$$
K \sim Z_{eff}\left\{\frac{0.457}{1.077 + Z_{eff}} + 0.29\right\} . \tag{29}
$$

The expressions for the fluxes incorporate no cross-terms so that Γ is dependent on $\partial n/\partial r$ only and $\tilde{Q}$ is dependent on $\partial T/\partial r$ only. In particular the trapped particle pinch terms in the neoclassical expressions are not present. Since the ion transport is found experimentally to be predominantly neoclassical, these terms are retained. The effect of a relatively small population of heavy impurity ions (such as iron or tungsten) is to dominate the effective ionic charge, Z_{eff}, of a hydrogen plasma and so modify the resistivity and the other transport coefficients in accordance with Eqs.(28)-(29) (Spitzer, 1962).

For a set of fluxes as given by these equations, the coding for the neoclassical fluxes contained in the subroutine ⟨2.6⟩ FLUXES (Fig.4) may be completely replaced by a new version. This would be the procedure if the neoclassical version were now to be disregarded and the code were to be based completely on the pseudo-classical model. The ad hoc inclusion of the pseudo-classical model could be achieved by a suitable call to the ⟨0.4⟩ EXPERT routine after the determination of the neoclassical fluxes. The appropriate section of the user-supplied EXPERT would over-write or append the fluxes with those appropriate to the pseudo-classical model.

However, for the particular pseudo-classical model given by Eqs.(28)-(29) examination of the equations indicates that they form a subset of the basic model, provided that the numerical coefficients are chosen correctly. In Table VI(a) the changes necessary to run the code in the pseudo-classical mode are listed in the NAMELIST NEWRUN that may be read in as input data.

The NAMELIST NEWRUN indicates additionally how this facility enables certain parts of the physical model to be temporarily removed in order to estimate the effect of these terms. For example, the setting of CEOB equal to zero enables all trapped-particle pinch terms to be omitted from the calculation. The selective omission of terms in an equation facilitates both the debugging of the code during development work and the understanding of the model equation during production work.

b. Pfirsch-Schlüter Diffusion. The model for plasma diffusion in the highly collisional regime ($\nu_{ei} \gg \nu_t$) (Rosenbluth and Kaufman, 1958; Taylor,1961) and modified by Pfirsch and Schlüter (1962) to take into account the effects of toroidal geometry is known to give diffusion losses which are too small in comparison with those observed in present-day Tokamaks. Nevertheless, Mercier and Soubbaramayer (1974) have found that the Pfirsch-Schlüter form of the diffusion coefficient modified by a large

multiplicative factor, $\lambda \gg 1$, is sufficient to simulate Tokamak behaviour in TFR (Rebut et al.,1973) and ST (Dimock et al.,1971) experiments. The modified Pfirsch-Schlüter diffusion may be represented by the following definition of the diffusion coefficient

$$D_{MPS} = \lambda(1 + 1.6q^2)\nu_{ei}\,\rho_{e\varphi}^{\,2} \quad , \qquad (30)$$

where q and $\rho_{e\varphi}$ are respectively the Tokamak safety factor and the electron gyro radius in the toroidal magnetic field

$$q = \frac{rB_\varphi}{RB_\theta} \quad , \quad \rho_{e\,\varphi}^{\,2} = \frac{2m_e kT_e}{e^2 B_\varphi^{\,2}} \quad .$$

The code ICARUS may be used to investigate the behaviour of a collision-free plasma in which the electron diffusion is determined by the modified Pfirsch-Schlüter diffusion coefficient and the ion thermal transport is determined by that given by the neoclassical model. The changes necessary to run the code in this mode are given in the NAMELIST reproduced in Table VI(b).

However, the use of neither a pseudo-classical nor a modified Pfirsch-Schlüter model is sufficient to be able to describe the "bell-shaped" electron temperature profile observed in the ST experimental discharge (Fig.7). Although the replacement times are reduced as a result of the enhanced diffusion processes (Table VII), the diffusion alone is unable to induce central peaking of the electron temperature, even though the Pfirsch-Schlüter diffusion coefficient peaks outwardly. It is necessary to invoke additional mechanisms to reproduce the experimental profiles.

4. Additional Source Terms

The addition of source terms to those already incorporated in the basic model may be accomplished in the first instance by the use of the EXPERT facility which will contain coding that modifies the arrays that contain the relevant source terms. Subsequent to the successful inclusion of these

terms, the coding in EXPERT may be transferred to a single, self-contained subroutine which is concerned exclusively with a particular aspect of the physical model. For example, it is reasonable to define in a single subroutine the radiation losses associated with bremsstrahlung and cyclotron processes (Fig.5). These are purely source terms with power losses given respectively by Rose and Clark (1961) and Rosenbluth (1970)

$$P_{bremsstrahlung} = \sqrt{\tfrac{3}{2}} \frac{Z_{eff}^2 e^6}{24\pi} \frac{\mu_o^3 c^3}{m_e h} n_e n_i \sqrt{\frac{8k T_e}{\pi m_e}} = CBREM\, Z_{eff}\, n_e^2\, T_{e(ev)}^{\frac{1}{2}} \qquad (31)$$

$$P_{cyclotron} = 0.16 \times 10^{16} \times \frac{B_\varphi^3}{R} T_{e(ev)} \sum_{i=1}^{2} \sum_{k=1}^{\infty} k^3 \frac{\Lambda_k^i}{1 + \Lambda_k^i} \qquad (32)$$

where

$$\Lambda_k^i = 5.0\ 10^{-16} \frac{nR}{B_\varphi} \left(10^{-6}\ T_{e(ev)}\right)^{k-1} f_k^{(i)}$$

$$f_k^{(1)} = (ke)^{k-2},\quad f_k^{(2)} = \frac{f_k^{(1)}}{2k+1} \quad .$$

(h is Planck's constant and c is the velocity of light. In the last expression above, e is the base for natural logarithms, and <u>not</u> the charge on the electron).

The coefficient CBREM is a function only of the fundamental constants. During the initialization procedure it is cleared to zero in subroutine ⟨1.2⟩ CLEAR , defined equal to unity in subroutine ⟨1.3⟩ PRESET and calculated as the product of CBREM and the fundamental constants in subroutine ⟨1.5⟩ AUXVAL. Accordingly, the magnitude of the bremsstrahlung losses may be modified in subroutine ⟨1.4⟩ DATA by defining the value of CBREM different from unity. In particular, setting CBREM equal to zero allows the removal of bremsstrahlung radiation effects.

The complete omission of all radiation effects contained in subroutine ⟨2.10⟩ RAD may be achieved by setting in the NAMELIST NEWRUN the logical array element

NLOMT2(10) = T

A logical IF statement at the beginning of each CLASS 1, CLASS 2 and CLASS 3 subroutine can allow the return to the calling subroutine before any calculation is performed. Any physical effect that is incorporated entirely within one subroutine can be suppressed by means of this facility.

Although radiation cooling is a process that will tend to make the Tokamak discharge thermally unstable (Furth et al.,1970) its effect is to reduce the central temperature of the discharge and encourage outward peaking of the electron temperature (Fig.8). If a cooling mechanism is required to induce central peaking of the electron temperature then it is necessary that this mechanism be effective in the outer, low-density regions of the discharge. Experimentally, it is observed that although there is a large outflow of particles to the limiter of the Tokamak, recycling of particles occurs to such an extent that the total number of particles contained in the torus remains approximately constant. Penetration into the plasma of these recycled particles occurs with the result that they are ionized and afford a source to the particle distribution and a sink to the electron energy distribution.

Detailed calculations of the effect of a neutral gas component have been performed by Düchs et al. (1972), Girard et al. (1972) and Hogan and Dory (1972), and take into account both the ionization of the neutral population and the production of hot neutrals as a result of charge exchange with the background plasma ion population. The source terms which enter the particle density equation (5) and electron energy equation (6) as a result of the ionization of the neutral species may be written

$$\frac{\delta n}{\delta t} = n n_n \langle \sigma v \rangle_I$$
$$\frac{\delta p_e}{\delta t} = - p_e \, n_n \, \langle \sigma v \rangle_I \left(\frac{E_I}{k T_e} \right) \quad , \tag{33}$$

where the interaction of the background plasma population, n, with the neutral species, n_n, is determined by the cross-section, $\langle \sigma v \rangle_I$, for ionization, together with an average energy loss of the electrons per ionizing collision, E_I . To indicate the predominant effect of the neutral gas, namely the depression of the electron temperature on the outside of the discharge and the elevation of the electron temperature in the central region of the discharge, a very simple model for the neutrals will be used for the purposes of this article. The source terms in Eq.(33) are replaced by an heuristic model which maintains constant the total particle content of the torus

$$\frac{\delta n}{\delta t} = \left[\Gamma (r = a) \, \frac{(m + 2)}{a} \right] \left(\frac{r}{a} \right)^m$$
$$\frac{\delta p_e}{\delta t} = - \left[\Gamma (r = a) \, \frac{(m + 2)}{a} \right] E_I \left(\frac{r}{a} \right)^m \quad . \tag{34}$$

The penetration of the neutral species may be achieved by setting m to an appropriate value. In ST the neutrals penetrate well into the plasma (Dimock et al.,1973) so it is reasonable to choose m = 2, which corresponds to a distribution that is parabolic in r. Although both the pseudo-classical and modified Pfirsch-Schlüter diffusion models give electron temperature profiles that have an inflexion point (Figs.9 and 10), the experimental values for the particle density and temperatures are approximated more accurately by the modified Pfirsch-Schlüter scaling law (Table VII) as a result of the greater outflow of particles which induces more peaking of the electron temperature on axis.

C. NEXT GENERATION TOKAMAKS

Extrapolations of the results of both neoclassical and pseudo-classical theory to the regime applicable to the next generation of larger Tokamaks

give optimistic predictions. In both cases conditions are favorable for thermonuclear burning to take place without the aid of additional heating mechanisms and provided that the mean particle density is less than 5×10^{19} m^{-3} and the toroidal current is greater than 3 MA (Fig.11). On the other hand the Pfirsch-Schlüter model, which gave the best simulation of the ST device, produces less favorable results when applied to a large Tokamak. Shallow penetration (m = 6) of a large influx of "neutrals" cools the plasma considerably near radius, r = a, with the result that the mean temperature is low (Table VIII).

The neoclassical model has the advantage that an isolated pinch is formed as a result of the Ware pinch effect. The input energy is dissipated almost entirely by bremsstrahlung radiation losses so that the predicted energy replacement time is comparatively long (Table VIII). However, the neoclassical solution is thermally unstable (Düchs et al., 1971) with off-axis peaking of the temperatures occurring to different extents depending on the initial conditions. Although the replacement times predicted by the pseudo-classical diffusion model are less than those corresponding to the neoclassical formulation, the solution is independent of the initial conditions and greater central and mean temperatures are obtained as a result of channelling of the toroidal current along the minor axis.

Within the context of the simple model for the neutral recycling, the pseudo-classical model produces the most optimistic results. Nevertheless it is anticipated that additional heating mechanisms will be necessary in order to ensure thermonuclear burning in future Tokamaks. The injection of a neutral beam of particles into a Tokamak (Kelley et al.,1972; Stewart et al.,1973; Aldcroft et al.,1973; Dei-Cas et al.,1973; McNally,1973; Cordey et al.,1974; Rome et al.,1974) in order to elevate the temperature

above that set by ohmic heating alone is an illustration of an experimental technique for which theoretical progress is very rapid (Sweetman, 1973; Callen et al., 1974; Connor and Cordey, 1974; Cordey, 1974; Cordey and Core, 1974; Rome et al., 1974).

The incorporation into a diffusion code of the effects of a beam of fast neutral particles may be performed in a sequence of separate operations which illustrate the versatility of the OLYMPUS system.

Provided that the energies of the fast neutrals are sufficiently high

$$E_f > 2 \times 10^4 A_f \quad \text{eV}$$

(where A_f is the atomic mass of the fast neutrals), the number $n_f(r,t)drdt$ of fast neutrals that are deposited in the radial interval $r \rightarrow r+dr$ and the time interval $t \rightarrow t+dt$ is given by Riviere (1971)

$$n_f(r,t) = \left[\frac{I_f}{q_f L}\, n_e(r)\right]\left[\exp\left\{\int_0^{x=r} \frac{n_e}{L}(x)dx\right\} + \exp\left\{-\int_0^{x=r} \frac{n_e}{L}(x)dx\right\}\right]\exp\left\{-\int_0^{x=a} \frac{n_e}{L}(x)dx\right\} \tag{35}$$

where I_f and q_f are respectively the neutral beam equivalent current and charge, and n_e is the electron number density.

The function, L, defined according to

$$L = \frac{L_f \sin\delta}{Z_{eff}(r)}$$

is a measure of the penetration of the beam and is affected by the effective charge, Z_{eff}, the angle, δ, at which the beam is inclined to the magnetic field, and the penetration "length", L_f, which for sufficiently high energies may be approximated (Sweetman, 1973) by

$$L_f = 5.5 \times 10^{14} \frac{E_f}{A_f} \qquad [m^{-2}] \ .$$

The determination of $n_f(r,t)$ forms a well-defined unit that is not affected by the calculations that follow. It is performed therefore in a single subroutine ⟨2.12⟩ INJECT which, during the initial stages of development, called ⟨0.4⟩ EXPERT in order to calculate the birth-profile, S(r,t), of electrons resulting from the ionization of the neutral beam. This involves distributing the beam around the toroidal and poloidal azimuths. A simplified model applicable to a finite pencil beam of neutrals, has been incorporated into the code ICARUS. This model defines

$$S(r,t) = \frac{n_f(r,t)}{4\pi rR} \tag{36}$$

where R is the major radius of the torus. The use of the EXPERT facility allows the model for the calculation of the birth profile to be subsequently developed - even by physicists who have no other connection with the rest of the code.

The distribution of injected power to the electrons and ions can be determined analytically (Stix, 1972; Sweetman, 1973). The model of Sweetman (1973) is incorporated at present into ICARUS with the electrons and ions receiving fractions f_e and f_i of the total injected power

$$\begin{aligned} f_i &= \left[1 + 0.34X^{\frac{1}{2}}\right]^{-1}, \qquad f_e = 1 - f_i, \\ X &= \left[\frac{A_i^{2/3}}{14.6\, A_f}\right]\frac{E_f}{T_{e(ev)}}, \end{aligned} \tag{37}$$

where A_i and A_f are respectively the atomic masses of the plasma ions and fast neutrals.

The effect on the temperature distributions of a beam of injected hot neutrals is dependent on the penetration of the beam (Fig.12). For injection at high energy ($E_f = 8 \times 10^4$ eV, $I_f = 10$ A) there is deep penetration of the beam with the result that the central temperature is strongly increased.

This can result in a tendency for the plasma to be thermally unstable when $q(r = o)$ becomes less than unity. On the other hand, for injection at low energy ($E_f = 2 \times 10^4$ eV, $I_f = 40$ A) the energy is deposited in the outer, low density regions with the result that the elevations of temperature and particle concentration increase the flow of particles and energy out of the discharge.

VI. Summary

Using the Tokamak code ICARUS as an example, this article has described the construction of a computer model which simulates the physical behaviour of a plasma discharge. The ICARUS code has been designed to be a versatile research apparatus which is able to examine various theoretical assumptions that may be represented by sets of coupled one-dimensional diffusion equations. By using the programming conventions and library facilities provided by the OLYMPUS system, it is possible to insert into the basic version of the code a selection of flux and source terms which are additional to those contained in the neoclassical diffusion model. Particular features of the code not only can be tested independently but also can be replaced easily, so that different physical assumptions, numerical techniques and input/output facilities can readily be used.

The predictions of neoclassical diffusion in an axisymmetric, toroidal containment device have been compared with the results of experiment, with the predictions of a pseudo-classical model, and with those of the Pfirsch-Schlüter model modified by a numerical factor chosen to make the associated diffusion coefficient comparable with that for pseudo-classical diffusion at radius $r \approx 0.5\,a$. Within the context of an heuristic model for neutral recycling, which serves to maintain constant the total particle content and to provide an energy sink for the electrons, the modified Pfirsch-Schlüter

diffusion provides the best description of a small Tokamak discharge. On the other hand the pseudo-classical model provides the most optimistic predictions for the behaviour of a future-generation large Tokamak.

In extrapolating from one regime to another it is of paramount importance both to have confidence in the behaviour of the computer code and to have the theoretical assumptions clearly stated. To this end the OLYMPUS system imposes rigorous standards of structure and documentation which enable the physics, numerical techniques and programming methods to be understood and the code to be run by other workers on their own computers.

Acknowledgements

The authors are glad to acknowledge discussions with Dr J A Reynolds and Dr T E Stringer, and would like to thank Dr D F Düchs for making versions of his code available to them for Tokamak calculations at Culham.

TABLE I. The numerical values of the constants in the neoclassical transport coefficients for a collision-free plasma

CA11 = 1.12	CB11 = - 0.62	CC11 = 0.0
CA12 = 1.27	CB12 = - 0.77	CC12 = 0.0
CA13 = 2.44	CB13 = - 1.44	CC13 = 0.0
CA22 = 2.64	CB22 = - 0.93	CC22 = 0.0
CA23 = 4.35	CB23 = - 1.85	CC23 = 0.0
CA33 = - 1.95	CB33 = 0.95	CC33 = 1.0
CAI = 0.48	CBI = 0.23	CCI = 0.0
CAY = 0.0	CBY = 1.17	CCY = 1.33

TABLE II. The "smoothing functions" used to extend the range of validity of the neoclassical transport model to more collisional plasmas.

SMTH11	$1.0/(1.0 + 1.76\ \nu^*_{ei})$
SMTH12	$1.0/(1.0 + 0.66\ \nu^*_{ei})$
SMTH13	$1.0/(1.0 + 0.85\ \nu^*_{ei})$
SMTH22	$1.0/(1.0 + 0.35\ \nu^*_{ei})$
SMTH23	$1.0/(1.0 + 0.40\ \nu^*_{ei})$
SMTH33	$1.0/(1.0 + 1.00\ \nu^*_{ei})$
SMTHI	$1.0/(1.0 + 0.36\ \nu^*_{ii})$

"Smoothing functions", SMTH$\alpha\beta$, act to transform $(CA\alpha\beta\epsilon^{\frac{1}{2}} + CB\alpha\beta\ \epsilon + CC\alpha\beta)$ into $(CA\alpha\beta\ \epsilon^{\frac{1}{2}} + CB\alpha\beta\ \epsilon) \times SMTH\alpha\beta + CC\alpha\beta$. The normalized collision frequencies, ν^*_{ei} and ν^*_{ii}, are defined as

$$\nu^*_{ei} = \frac{\nu_{ei}}{\nu_b} \quad , \quad \nu^*_{ii} = \frac{\nu_{ii}}{\nu_b} \quad ,$$

where ν_b is the bounce frequency of trapped particles.

TABLE III. The names, identification numbers and titles of the primary subroutines called by the main control subroutine ⟨0.3⟩ COTROL.

NAME	NUMBER	TITLE
LABRUN	1.1	Label the run
CLEAR	1.2	Clear variables and arrays
PRESET	1.3	Set default values
DATA	1.4	Define data specific to run
AUXVAL	1.5	Set auxiliary values
INITAL	1.6	Define physical initial conditions
RESUME	1.7	Resume run from previous record
START	1.8	Start the calculation
STEPON	2.1	Step on the calculation
OUTPUT	3.1	Control the output
TESEND	4.1	Test for completion of run
ENDRUN	4.2	Terminate the run
EXPERT	0.4	Modify standard operation of program

TABLE IV. The names, identification numbers and titles of the Class 2 subroutines which perform the diffusion calculation in the computer code, ICARUS.

NAME	NUMBER	TITLE
STEPON	2.1	Step on the calculation
BLKSET	2.2	Set up the matrix form of the difference equations
BNDRYO	2.3	Set up the boundary conditions at radius, r = o
BNDRYN	2.4	Set up the boundary conditions at radius, r = a
PHYSIN	2.5	Organize the determination of the transport model
TRANS	2.6	Evaluate the neoclassical transport coefficients
SOURCE	2.7	Evaluate the neoclassical source terms
RIPPLE	2.8	Evaluate the transport resulting from modulations in the toroidal field
TRAP	2.9	Evaluate the transport resulting from trapped particle instabilities
RAD	2.10	Evaluate the radiation losses
ALPHA	2.11	Evaluate the effect of alpha particles
INJECT	2.12	Evaluate the effect of the injection of a beam of neutrals
BLKSLV	2.13	Solve the block tridiagonal system of linear equations
LUF	2.14	Form the Lower and Upper factors
FANDB	2.15	Forward and Backward substitution
FANDBV	2.16	Vector Foward and Backward substitution
CHECK	2.17	Check the accuracy of the solution
RESET	2.18	Reset variables so that the timestep may be repeated
ADCOMP	2.19	Evaluate the effect of adiabatic compression
STORE	2.20	Determine the variables to be stored periodically in time

TABLE V. The labelled common blocks which define the "state of the calculation" in the computer code, ICARUS.

NAME	NUMBER	TITLE
COMBAS	1.1	Basic system parameters
COMFUN	1.4	Fundamental constants
COMDDP	1.9	Development and diagnostic parameters
COMPHY	2.1	Physical quantities
COMCNS	2.2	Physical constants
COMTOK	2.3	Machine parameters
COMFLX	2.4	Flux variables
COMSRC	2.5	Source variables
COMESH	3.1	Mesh variables
COMSTO	3.2	Temporally accumulated variables
COMXSN	3.3	Averages over the minor cross-section
COMADM	3.4	Administrative variables
COMCON	4.1	Control variables
COMPLT	5.1	Graph plotting variables

TABLE VI. The namelist NEWRUN used as input data for the computer code, ICARUS, in order to model

(a) pseudo-classical diffusion in a small Tokamak

(b) modified Pfirsch-Schlüter diffusion in a small Tokamak

(c) neoclassical diffusion in a large Tokamak

(a)
SMALL TOKAMAK RUN
75/01/01
PSEUDO-CLASSICAL DIFFUSION

```
NEWRUN
NLOMT2(12)= T,
CEOB      = 0.0,
XN        = 1.0,
CB11      = 0.0,
CB12      = 0.0,
CB13      = 0.0,
CB22      = 0.0,
CB23      = 0.0,
CBI       = 0.0,
CA11      = 0.0,
CA12      = 0.0,
CA22      = 0.0,
CAY       = 0.0,
CBY       = 0.0,
CC11      = 1.0,
CC12      = 1.5,
CC22      = 3.25,
CCY       = 1.5,
PIQ(12)   = 2.0,
PIQ(13)   = 1.0,
PIQ(14)   = 40.0,

END
RESET
END
```

(b)
SMALL TOKAMAK RUN
75/01/01
MODIFIED PFIRSCH-SCHLÜTER DIFFUSION

```
& NEWRUN
  NLOMT2(12)= T,
  CEOB      = 0.0,
  XE        = 200.0,
  XN        = 0.0,
  CB11      = 0.0,
  CB12      = 0.0,
  CB13      = 0.0,
  CB22      = 0.0,
  CB23      = 0.0,
  CBI       = 0.0,
  CA11      = 0.0,
  CA12      = 0.0,
  CA22      = 0.0,
  CAY       = 0.0,
  CBY       = 0.0,
  CC11      = 1.0,
  CC12      = 1.5,
  CC22      = 3.25,
  CCY       = 1.5,
  PIQ(12)   = 2.0,
  PIQ(13)   = 1.0,
  PIQ(14)   = 40.0,

& END
& RESET
& END
```

(c)
LARGE TOKAMAK RUN
75/01/01
NEOCLASSICAL DIFFUSION

```
& NEWRUN
  RMAJOR = 2.93,
  RMINOR = 1.28,
  RPLSMA = 1.28,
  CURENT = 3.0E+06,
  DENMAX = 9.1E+19,
  TEMAX  = 1000.0,
  TIMAX  = 1000.0,
  NLOMT2(12) = T,
  CB11   = 0.0,
  CB12   = 0.0,
  CB13   = 0.0,
  CB22   = 0.0,
  CB23   = 0.0,
  CBI    = 0.0,
& END
& RESET
& END
```

TABLE VII. General characteristics of a high density discharge in a small Tokamak

($R = 1.09m$, $a = 0.14m$, $\langle n \rangle^{*} \sim 1.7 \times 10^{19} m^{-3}$, $I_{\varphi} = 60$ kA, $B_{\varphi} = 3.7T$).

Scaling law	Central values					Mean values *					Replacement times (ms)†			
	n ($\times 10^{19} m^{-3}$)	T_e (keV)	T_i (keV)	E_Z (Vm^{-1})	q	$\langle n \rangle$ ($\times 10^{19} m^{-3}$)	$\langle T_e \rangle$ (keV)	$\langle T_i \rangle$ (keV)	$\beta_{\theta e}$	$\beta_{\theta i}$	τ_P	τ_E'	τ_{Ee}	τ_{Ei}
ST Experiment (35 ms)	2.7	2.2	0.46	0.36	0.6	1.8	0.76	-	0.75	-	16	13	19	-
Neoclassical Theory (34 ms)	3.5	0.67	0.51	0.08	3.2	1.65	0.58	0.45	0.52	0.40	1506	106	604	98
Pseudo-classical Theory (33 ms)	2.0	1.03	0.46	0.22	2.0	1.65	0.65	0.35	0.58	0.31	23	20	55	23
Pfirsch-Schlüter x 200 (34 ms)	2.6	2.6	0.50	0.20	0.58	1.65	0.62	0.28	0.56	0.25	6	11	44	22

$$* \langle n \rangle = \frac{2}{a^2} \int_0^a n \, r \, dr \qquad \langle T \rangle = \frac{\int_0^a n k T \, r \, dr}{\int_0^a n \, r \, dr}$$

$$\beta_\theta = \frac{16\pi^2 \int_0^a n k T \, r \, dr}{\mu_0 I^2}$$

$$\dagger \; \tau_P = \int_0^a \frac{n \, r \, dr}{[r \Gamma]_{r=a}} \qquad \tau_E' = \int_0^a \frac{nk(T_e + T_i) r \, dr}{\frac{(\gamma-1)}{\mu_0} [E_\varphi \beta_\theta]_{r=a}}$$

$$\tau_{Ee} = \int_0^a \frac{nk \, T_e \, r \, dr}{(\gamma-1)[rQ_e]_{r=a}} \qquad \tau_{Ei} = \int_0^a \frac{nk \, T_i \, r \, dr}{(\gamma-1)[rQ_i]_{r=a}}$$

TABLE VIII. General characteristics of a discharge in a large Tokamak

($R = 2.93$m, $a = 1.28$ m, $\langle n \rangle^{*} = 5.0 \times 10^{19}$ m^{-3}, $I_\varphi = 3$MA, $B_\varphi = 3$T).

Scaling law		Central values					Mean values*					Replacement times† (s)			
		n ($\times 10^{19}$ m^{-3})	T_e (keV)	T_i (keV)	E_Z (Vm^{-1})	q	$\langle n \rangle$ ($\times 10^{19}$ m^{-3})	$\langle T_e \rangle$ (keV)	$\langle T_i \rangle$ (keV)	$\beta_{\theta e}$	$\beta_{\theta i}$	τ_P	τ_E'	τ_{Ee}	τ_{Ei}
Neoclassical Theory with T_e max set initially	100 eV (5.5s)	22.1	1.9	1.9	0.014	1.28	5.1	2.0	2.0	0.19	0.19	-942	16	7349	500
	4000eV (5.4s)	21.5	2.1	2.1	0.013	1.17	5.2	2.2	2.2	0.21	0.21	-325	11	-5479	447
Pseudo-classical Theory (4.6s)		8.1	3.1	3.1	0.012	0.73	5.0	2.6	2.6	0.24	0.24	147	6	399	31
Hirsch-Schlüter x200 (4.0s)		6.0	3.6	3.6	0.011	0.65	5.0	1.3	1.3	0.12	0.12	0.2	1	5	7

* $\langle n \rangle = \dfrac{2}{a^2}\int_0^a n\,r\,dr \qquad \langle T \rangle = \dfrac{\int_0^a n k T\,r\,dr}{\int_0^a n\,r\,dr}$

$\beta_\theta = \dfrac{16\pi^2 \int_0^a n k T\,r\,dr}{\mu_0 I^2}$

† $\tau_P = \dfrac{\int_0^a n\,r\,dr}{[r\Gamma]_{r=a}}$

$\tau_{Ee} = \dfrac{\int_0^a nk\,T_e\,r\,dr}{(\gamma-1)[rQ_e]_{r=a}}$

$\tau_E' = \dfrac{\int_0^a nk(T_e + T_i)r\,dr}{\dfrac{(\gamma-1)}{\mu_0}[E_\varphi B_\theta]_{r=a}}$

$\tau_{Ei} = \dfrac{\int_0^a nk\,T_i\,r\,dr}{(\gamma-1)[rQ_i]_{r=a}}$

References

Aldcroft, D., Burcham, J., Cole, H.C., Cowlin, M., and Sheffield, J. (1973). Nucl.Fus. 13, 393.

Artsimovich, L.A. (1971). Sov.Phys.- JETP Lett.13, 70.

Artsimovich, L.A., Anashin, A.M., Gorbunov, E.P., Ivanov, D.P., Petrov, M.P., and Strelkov, V.S. (1969). Sov.Phys.- JETP Lett.10, 82.

Bateman, G., Schneider, W., and Grossmann, W. (1974). Nucl.Fus. 14, 669.

Berry, L.A., Clarke, J.F., and Hogan, J.T. (1974). Phys.Rev.Lett. 32, 362.

Bol, K., Ellis, R.A., Eubank, H., Furth, H.P., Jacobsen, R.A., Johnson, L.C., Mazzucato, E., Stodiek, W., and Tolnas, E.L. (1972). Phys.Rev.Lett. 29, 1495.

Braginskii, S.I. (1965). In "Reviews of Plasma Physics" (M.A.Leontovich,ed.), Vol.I, pp. 205-311. Consultants Bureau, New York.

Burke, V.M. (1970). Computer Phys.Commun. 1, 473.

Callen, J.D., Colchin, R.J., Fowler, R.H., McAlees, D.G., and Rome, J.A. (1974). Proc.5th Int.Conf.on Plasma Physics and Controlled Nuclear Fusion Research, Tokyo, IAEA-CN-33/A16-3.

Chapman, S., and Cowling, T.G. (1953). "The Mathematical Theory of Non-Uniform Gases." Cambridge University Press, New York.

Christiansen, J.P., and Roberts, K.V. (1974). Computer Phys.Commun. 7, 245.

Christiansen, J.P., Ashby, D.E.T.F., and Roberts, K.V. (1974). Computer Phys.Commun. 7, 271.

Christiansen, J.P. (1975). To be submitted for publication in Computer Physics Communications.

Connor, J.W., and Cordey, J.G. (1974). Nucl.Fus. 14, 185.

Cordey, J.G. (1974). Proc.5th Int.Conf. on Plasma Physics and Controlled Nuclear Fusion Research, Tokyo, IAEA-CN 33/A16-1.

Cordey, J.G., and Core, W.G.F. (1974). Phys.Fluids 17, 1626.

Cordey, J.G., Hugill, J., Paul, J.W.M., Sheffield, J., Speth, E., Stott,P.E., and Tereshin, V.I. (1974). Nucl.Fus. 14, 441.

Dei-Cas, R., De Sacy, S., Druaux, J., Marty, D., and Rebut, P.H. (1973). Proc.3rd Int.Symp.on Toroidal Plasma Confinement, Garching, E9.

Dimock, D., Eckhartt, D., Eubank, H., Hinnov, E., Johnson, L.C., Meservey, E., Tolnas, E., and Grove, D.J. (1971). Proc.4th Int.Conf.on Plasma Physics and Controlled Nuclear Fusion Research, Madison 1, 451.

Dimock, D.L., Eubank, H.P., Hinnov, E., Johnson, L.C., and Meservey, E.B. (1973). Nucl.Fus. 13, 271.

Dnestrovskii, Y.N., Kostomarov, D.P., and Pavlova, N.L. (1970). Proc.4th European Conf. on Control Fusion and Plasma Physics, Rome 1, 17.

Dory, R.A., and Widner, M.M. (1970). Bull.Am.Phys.Soc. Ser.II, No.11, 1418.

Drummond, W.E., Nielsen, P., Phillips, P., Medley, S., Jancarik, J., and Bengston, R. (1973). Proc.3rd Int.Symp. on Toroidal Plasma Confinement, Garching, B6.

Düchs, D.F. (1970). Bull.Am.Phys.Soc. Ser.II, No.11, 1488.

Düchs, D.F., Furth, H.P., and Rutherford, P.H. (1971). Proc.4th Int.Conf.on Plasma Physics and Controlled Nuclear Fusion Research, Madison 1, 369.

Düchs, D.F., Furth, H.P., and Rutherford, P.H. (1972). Proc.5th European Conf. on Controlled Fusion and Plasma Physics, Grenoble 1, 14.

Feneberg, W., and Lackner, K. (1973). Nucl.Fus. 13, 549.

Friedberg, J.P. (1970). Phys.Fluids 13, 1812.

Furth, H.P., Rosenbluth, M.N., Rutherford, P.H., and Stodiek, W. (1970). Phys.Fluids 13, 3020.

Galeev, A.A., and Sagdeev, R.Z. (1968). Sov.Phys.- JETP 26, 233.

Gibson, A., Bickerton, R.J., Cole, H.C., Haegi, M., Hugill, J., Paul, J.W.M., Reynolds, P., Sheffield, J., Speth, E., and Stott, P.E. (1973). Proc.3rd Int.Symp. on Toroidal Plasma Confinement, Garching, B16-I.

Girard, J.P., Khelladi, M., and Marty, D. (1972). Proc.5th European Conf. on Controlled Fusion and Plasma Physics, Grenoble 1, 105.

Goedbloed, J.P., and Hagebeuk, H.J.L. (1972). Phys.Fluids 15, 1090.

Hazeltine, R.D., Hinton, F.L., and Rosenbluth, M.N. (1973). Phys.Fluids 16, 1645.

Hinnov, E., Dimock, D.L., Johnson, L.C. and Meservey, E.B. (1973). Proc.3rd Int.Symp.on Toroidal Plasma Confinement, Garching, B13.

Hinton, F.L., Wiley, J.C., Düchs, D.F., Furth, H.P., and Rutherford, P.H. (1972). Phys.Rev.Lett. 29, 698.

Hogan, J.T., and Dory, R.A. (1972). Proc.5th European Conf. on Controlled Fusion and Plasma Physics, Grenoble 1, 40.

Hogan, J.T., Widner, M.M., and Dory, R.A. (1971). Phys.Rev.Lett. 36A,3, 217.

Hughes, M.H., Roberts, K.V., and Roberts, P.D. (1975). Computer Phys.Comm. 9, 51.

Itoh, S., Fujisawa, N., Funahashi, A., Kunieda, S., Takeda, T., Matoba, T., Kasai, S., Sugawara, T., Toi, K., Suzuki,N., Maeno, M., Inoue, K., Ohta, M., Matsuda, S., Ohga, T., Arai, T., Yokokura, K., and Mori, S. (1973). Proc.3rd Int.Symp. on Toroidal Plasma Confinement, Garching, B4.

Kadomtsev, B.B.,and Pogutse, O.P. (1971). Nucl.Fus. 11, 67.

Keeping, P.M., Grimm, R.C., and Killeen, J. (1972). Proc.5th European Conf. on Controlled Fusion and Plasma Physics, Grenoble 1, 38.

Kelley, G.G., Morgan, O.B., Stewart, L.D., and Stirling, W.L. (1972). Nucl.Fus. 12, 169.

Kelley, G.G., Barnett, C.F., Berry, L.A., Bush, C.E., Callen, J.D., Clarke, J.F., Colchin, R.J., Dory, R.A., England, A.C., Hogan, J.T., McNally, J.R., Murakami, M., Neidigh, R.V., Roberts, M., Rome, J.A., and Wing, W.R. (1973). Proc.3rd Int.Symp. on Toroidal Plasma Confinement, Garching, B3-I.

Knorr, G. (1965). Phys.Fluids 8, 1334.

McNally, J.R. (1973). Report ORNL TM 4363, Oak Ridge.

Mercier, C., and Soubbaramayer. (1970). Proc.4th European Conf. on Controlled Fusion and Plasma Physics, Rome 1, 16.

Mercier, C., and Soubbaramayer. (1974). Private communication.

National Computing Centre Limited. (1970). "Standard Fortran Programming Manual." The National Computing Centre Limited, Manchester, England.

Newcomb, W.A. (1960). Ann.Phys.(NY) 10, 232.

Rebut, P.H., Bariaud, A., Breton, C., Bussac, J.P., Crenn, J.P., Dei-Cas, R., Michelis, C.De., Delmas, M., Ginot, P., Girard, J.P., Gourdon, C., Hennion, F., Huguet, M., Launois, D., Lecoustey, P., Marty, D., Mattioli, M., Mercier, C., Morriette, P., Platz, P., Plinate, P., Sledziewski, Z., Smeulders, P., Soubbaramayer., Tachon, J., Torossian, A., and Ya'akobi, B. (1973). Proc.6th European Conf. on Controlled Fusion and Plasma Physics, Moscow 2, 20.

Richtmyer, R.D., and Morton, K.W. (1967). "Difference Methods for Initial-Value Problems." Wiley, New York.

Riviere, A.C. (1971). Nucl.Fus. 11, 363.

Roberts, K.V. (1969). Computer Phys.Commun. 1, 1.

Roberts, K.V. (1971). Computer Phys.Commun. 2, 385.

Roberts, K.V. (1974). Computer Phys.Commun. 7, 237.

Rome, J.A., Callen, J.D., and Clarke, J.F. (1974). Nucl.Fus. 14, 141.

Rose, D.J., and Clark, M. (1961). "Plasmas and Controlled Fusion." MIT Press/Wiley, New York.

Rosenbluth, M.N. (1970). Nucl.Fus. 10, 340.

Rosenbluth, M.N., and Kaufman, A.N. (1958). Phys.Rev. 109, 1.

Rosenbluth, M.N., Hazeltine, R.D., and Hinton, F.L. (1972). Phys.Fluids 15, 116.

Rutherford, P.H. (1970). Phys.Fluids 13, 482.

Pfirsch, D., and Schlüter, A. (1962). Report MPI/PA/7/62, Max-Planck Institute for Physics and Astrophysics, Munich.

Potter, D.E. (1973). "Computational Physics." Wiley, New York.

Potter, D.E., and Tuttle, G.H. (1973). Proc.6th European Conf. on Controlled Fusion and Plasma Physics, Moscow 1, 217.

Shafranov, V.D. (1970). Sov.Phys.- Tech.Phys. 15, 175.

Sharp, W., and Taylor, J.C. (1973). Proc.6th European Conf. on Controlled Fusion and Plasma Physics, Moscow 1, 45.

Spitzer, L. (1962). "Physics of Fully Ionized Gases." Wiley/Interscience, New York.

Stewart, L.D., Callen, J.D., Clarke, J.F., Davies, R.C., Dory, R.A., Hogan, J.T., Jernigan, T.C., Morgan, O.B., Rome, J.A., and Stirling, W.L. (1973). Proc.3rd Int.Symp. on Toroidal Plasma Confinement, Garching, E8.

Stix, T.H. (1972). Plasma Phys. 14, 367.

Sweetman, D.R. (1973). Nucl.Fus. 13, 157.

Sykes, A., and Wesson, J.A. (1974). Nucl.Fus. 14, 645.

Taylor, J.B. (1961). Phys.Fluids 4, 1142.

Von Hagenow, K., and Lackner, K. (1973). Proc.3rd Int.Symp. on Toroidal Plasma Confinement, Garching, F7.

Ware, A.A. (1970). Phys.Rev.Lett. 25, 916.

Watkins, M.L., Hughes, M.H., Keeping, P.M., and Killeen, J. To be submitted for publication in Computer Physics Communications.

Winsor, N.K., Johnson, J.L., and Dawson, J.M. (1970). J.Comp.Phys. 6, 430.

Wooten, J., Hicks, H.R., Bateman, G., and Dory, R.A. (1974). Report ORNL TM4784, Oak Ridge.

Yoshikawa, S. (1970). Phys.Rev.Lett. 25, 353.

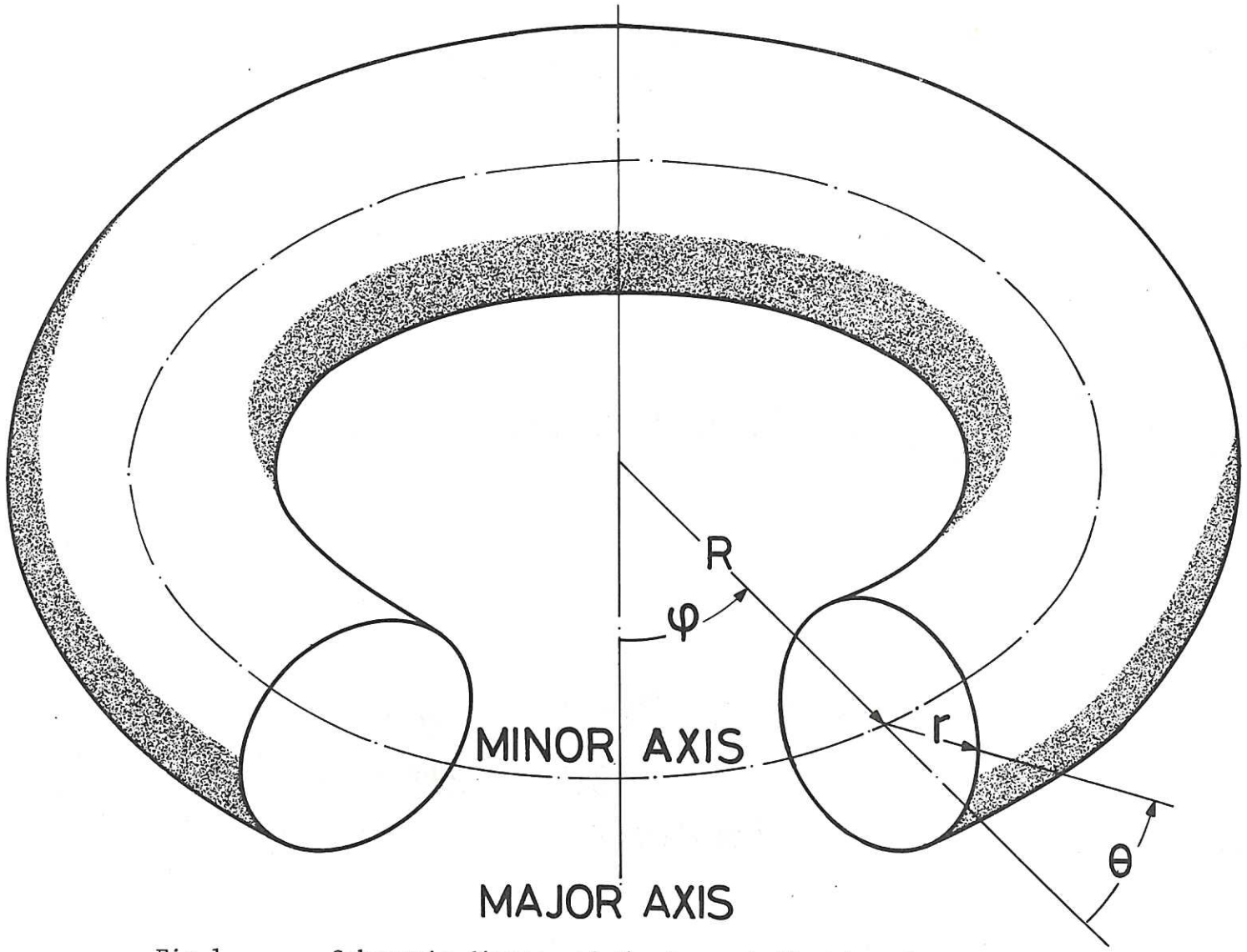


Fig.1 Schematic diagram of the torus indicating the coordinate system (r,θ,φ).

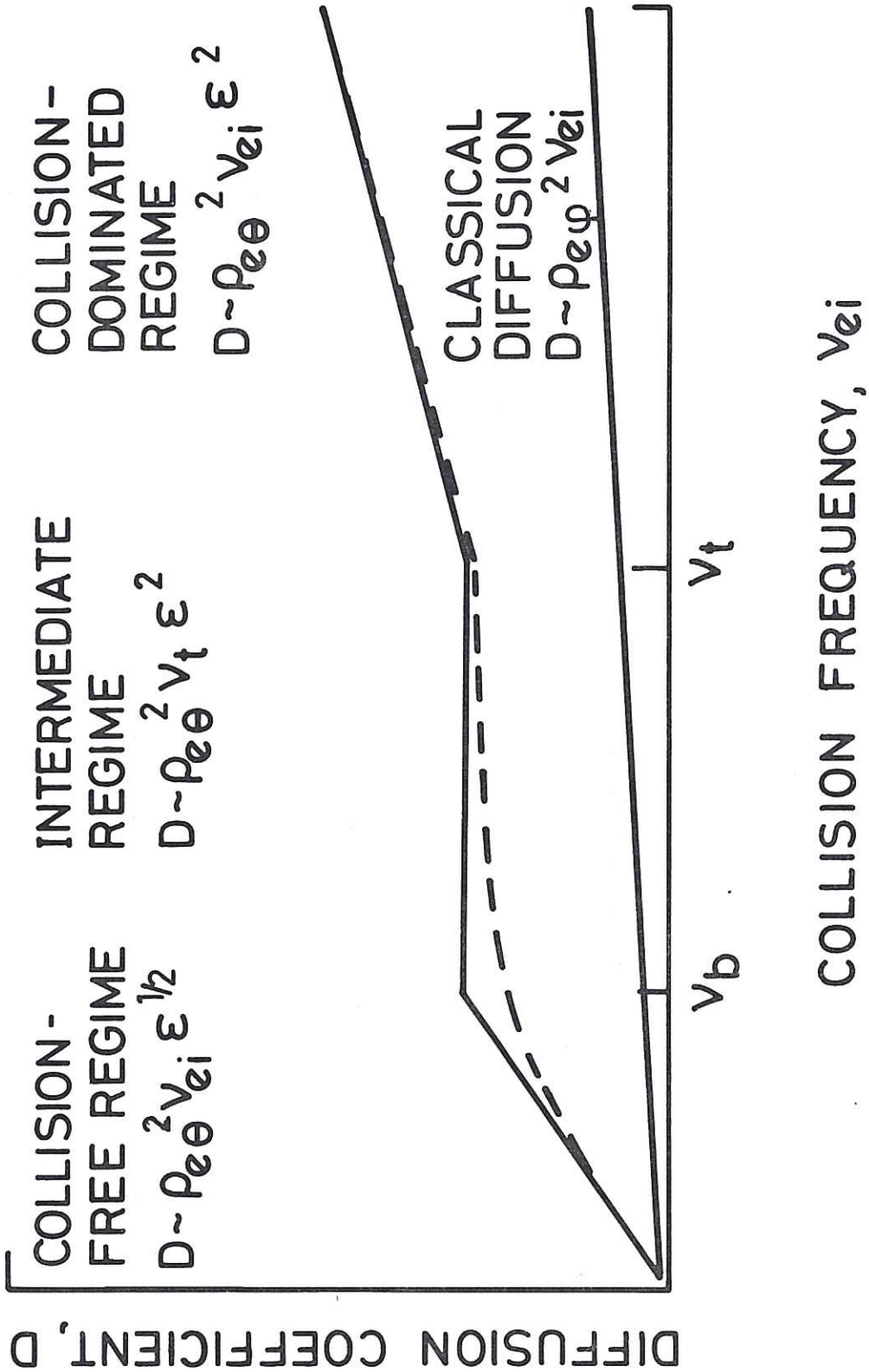


Fig.2 Schematic representation of the neoclassical diffusion coefficient, D, as a function of the electron-ion collision frequency, ν_{ei}, in the collision-free ($\nu_{ei} < \nu_b$), intermediate ($\nu_b < \nu_{ei} < \nu_t$), and collision-dominated ($\nu_{ei} > \nu_t$) regimes. The "smoothed" approximation for the diffusion coefficient used in the computer code, ICARUS, is also indicated together with the coefficient for classical diffusion in a cylinder.

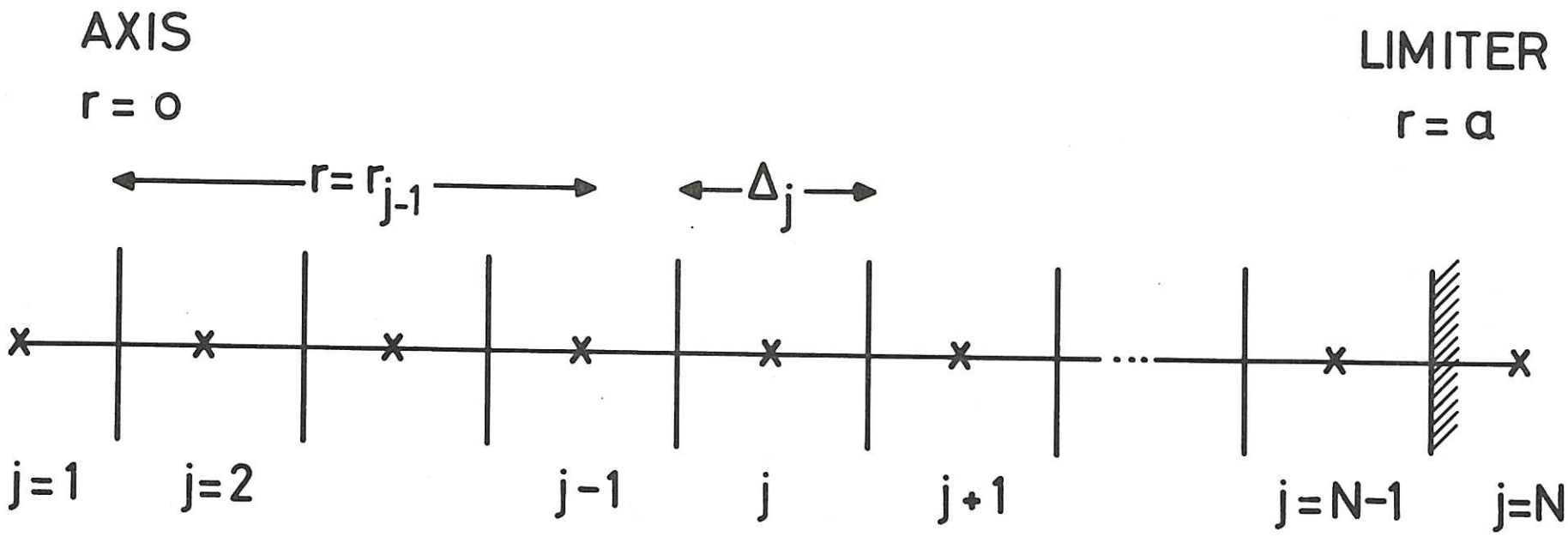


Fig.3 The finite-difference mesh.

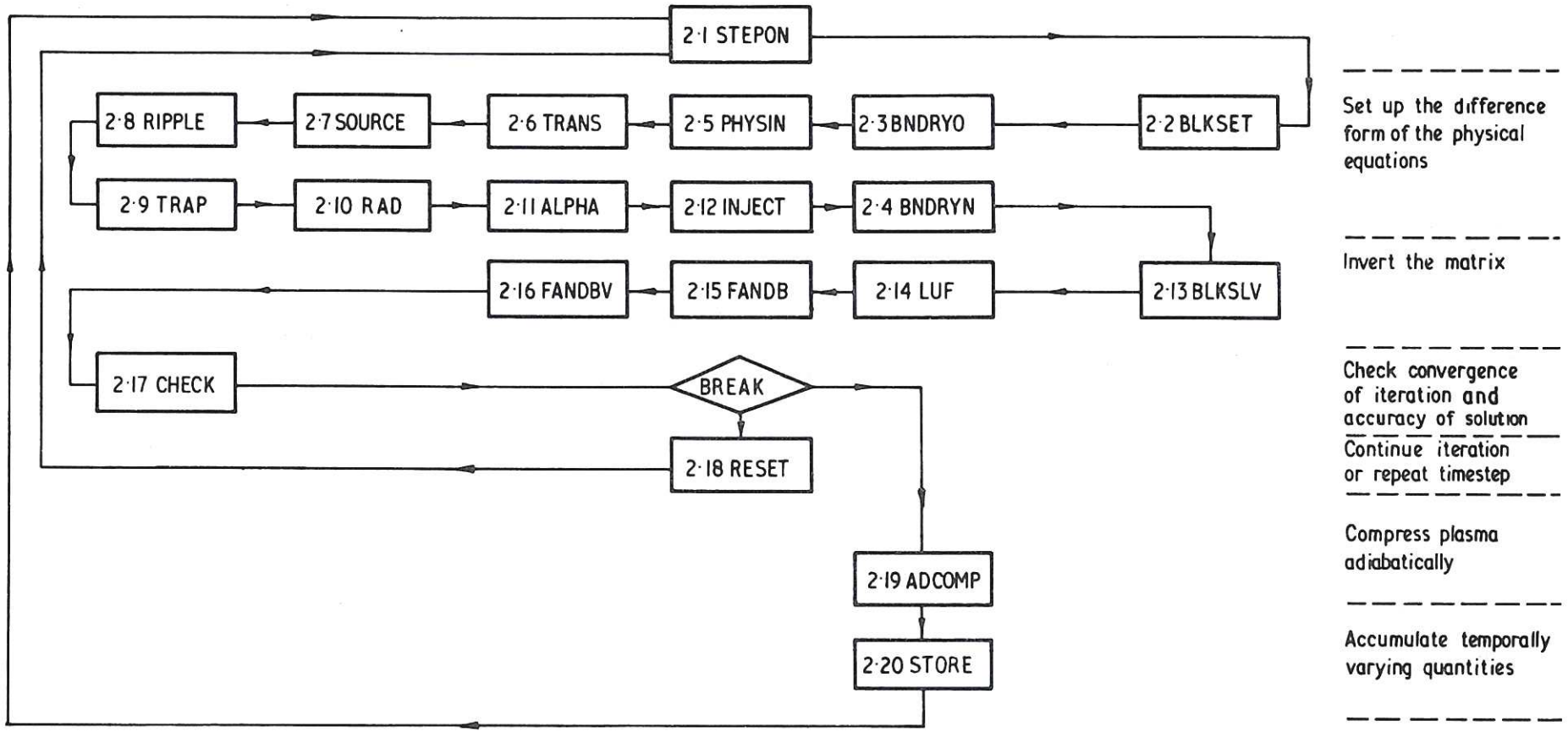


Fig.4 Flow diagram for the calculation subroutines (Class 2).

```
C
      SUBROUTINE RAD (K, KNIT)
C
C 2.10 Evaluate the radiation losses
C
// SUBSTITUTE COMFUN
// SUBSTITUTE COMDDP
// SUBSTITUTE COMCNS
// SUBSTITUTE COMTOK
// SUBSTITUTE COMSRC
// SUBSTITUTE COMESH
// SUBSTITUTE COMXSN
// SUBSTITUTE COMADM
C-----------------------------------------------------------------------
      DIMENSION ZFIN(5,2)
C
      DATA ICLASS, ISUB/2, 10/
C
      DATA ZFIN/0.38,1.0,8.2,120.0,2500.0,0.13,0.2,1.2,13.0,220.0/
C-----------------------------------------------------------------------
      IF (NLOMT2(ISUB)) RETURN
C-----------------------------------------------------------------------
CL              1.        Prologue
C
C     Initialize storage of energy changes at point *1*
      IF (K .EQ. 1) GO TO 500
C
C     Dependent variables at point *k*
      ZN = DEN(K)
      ZPE = PE(K)
C
      ZEFF = EFFZ(K)
C
C     Evaluate the independent variable
      ZR = R(K)
      ZDR = DELTAR(K)
      ZFAC = ZN * ZR * ZDR
C
C     Calculate the electron temperature in electron volts
      ZTE = (ZPE / ZN) / CHARGE
      ZRTTE = SQRT (ZTE)
C
C     Calculate the electron temperature in joules
      ZTE = ZTE * CHARGE
C
C-----------------------------------------------------------------------
CL              2.        Bremsstrahlung radiation loss
C
      ZBREM = CBREM * ZEFF * ZRTTE * ZN
C
C-----------------------------------------------------------------------
CL              3.        Cyclotron radiation loss
C
      ZCYC1 = CYC1 * ZN
C
C     Initialize accumulator
      ZSUM=0.0
C     Scan over *i*
      DO 302 J1 = 1, 2
C     Scan over harmonics
      DO 301 J2 = 1, 5
C     Rosenbluth zlambda function
      ZLIN = ZCYC1 * ((0.625E+13 * ZTE) ** (J2-1)) * ZFIN(J2,J1)
C     Contribution of this harmonic
      ZHARM=(J2*J2*J2)*ZLIN/(1.0+ZLIN)
C     Accumulate cyclotron loss
301   ZSUM = ZSUM + ZHARM
302   CONTINUE
C
C     Cyclotron power loss
      ZCYC = CYC2 * ZTE *ZSUM * (1.0 - CYCFAC) / ZN
C
C-----------------------------------------------------------------------
CL              4.        Source terms
C
CL                  4.1     Electron pressure equation
      SCPEN = SCPEN - (ZBREM + ZCYC) * CGM1
C
C-----------------------------------------------------------------------
CL              5.        Store quantities for diagnostic purposes
C
C     Return if not called during the first iteration
500   IF (KNIT .NE. 0) RETURN
C
C     Initialise for energy changes this timestep
      IF (K .NE. 1) GO TO 501
      XBREM = 0.0
      XCYC = 0.0
      RETURN
501   CONTINUE
C
C     Determine the energy changes this timestep
C     Bremsstrahlung radiation loss
      XBREM = XBREM + ZBREM * ZFAC
C
C     Cyclotron radiation loss
      XCYC = XCYC + ZCYC * ZFAC
C
C     Determine the energy changes since time, t=0
      IF (K .NE. NM1) RETURN
      XBREMT = XBREMT + XBREM * DT
      XCYCT = XCYCT + XCYC * DT
C
      RETURN
      END
```

Fig.5 Listing of subroutine ⟨2.10⟩ RAD in which are calculated the source terms that result from bremsstrahlung and cyclotron radiation losses.

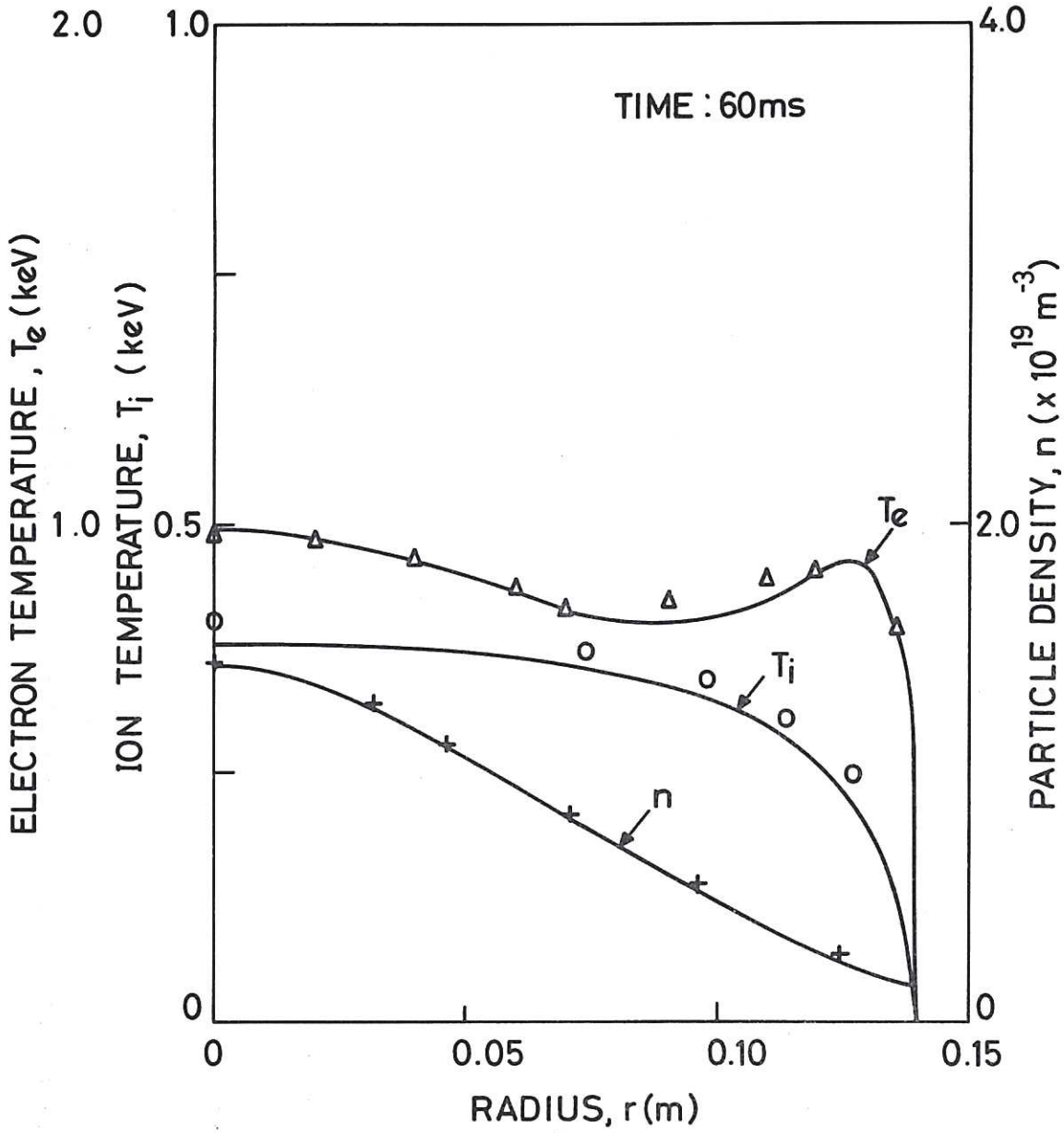


Fig.6 Comparison of the radial profiles of particle density, n, electron temperature, T_e, and ion temperature, T_i, for conditions applicable to a low density discharge in a Small Tokamak ($R = 1.09m$, $a = 0.14m$, $\langle n \rangle = 6 \times 10^{18} m^{-3}$; $I_\varphi = 40$ kA, $B_\varphi = 3T$) at 60 ms as given by the code, ICARUS (—), and that of Düchs (+, 0, Δ), using the neoclassical diffusion model.

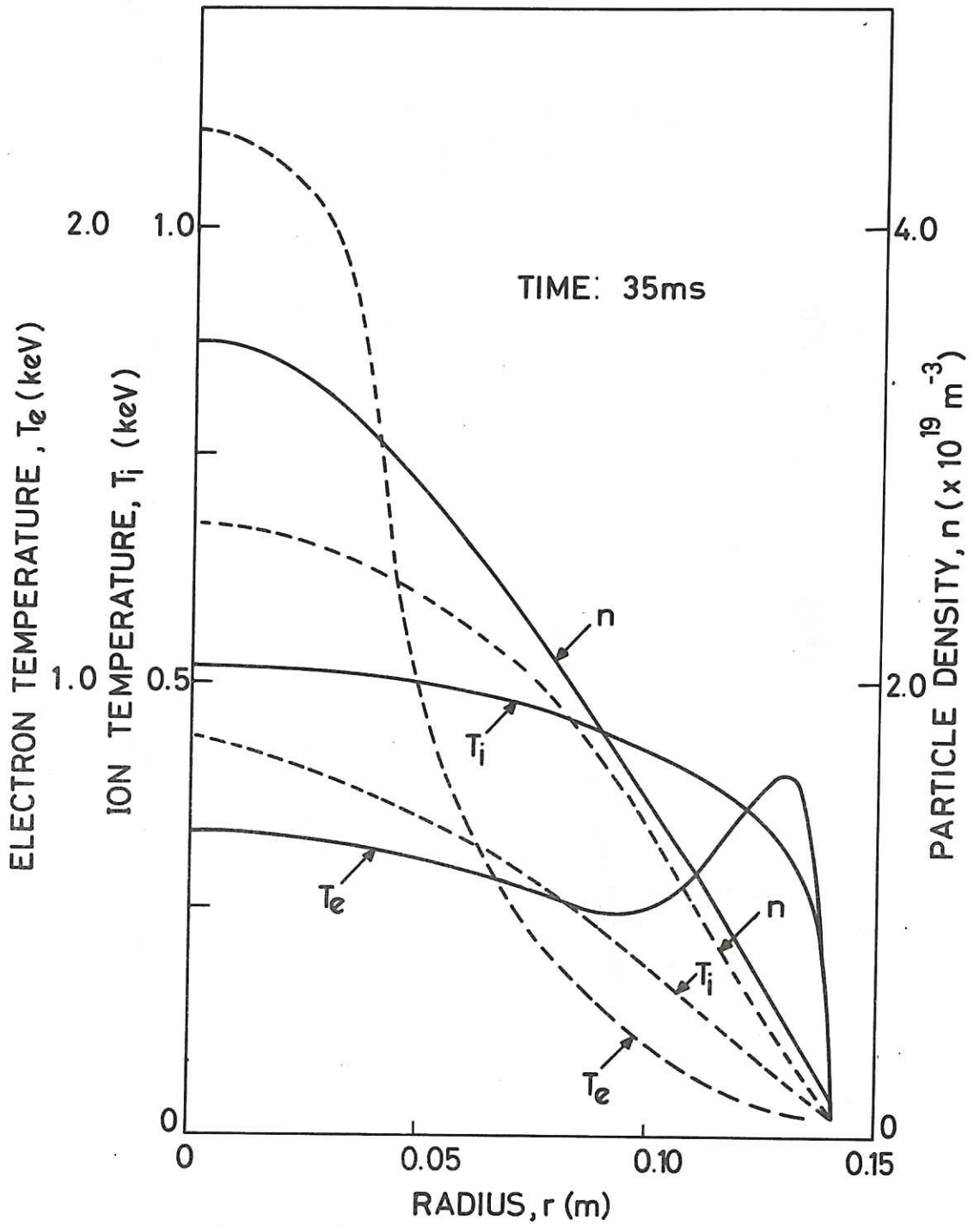


Fig.7 Comparison of the radial profiles of particle density, n, electron temperature, T_e, ion temperature, T_i, for conditions applicable to a high density discharge in a Small Tokamak ($R = 1.09$m, $a = 0.14$m, $\langle n \rangle = 1.65 \times 10^{19}$ m^{-3}, $I_\varphi = 60$ kA, $B_\varphi = 3.7$ T) given at 35 ms by the code, ICARUS, using the neoclassical diffusion model (—), and the laboratory experiment (Dimock et al., 1973) (----).

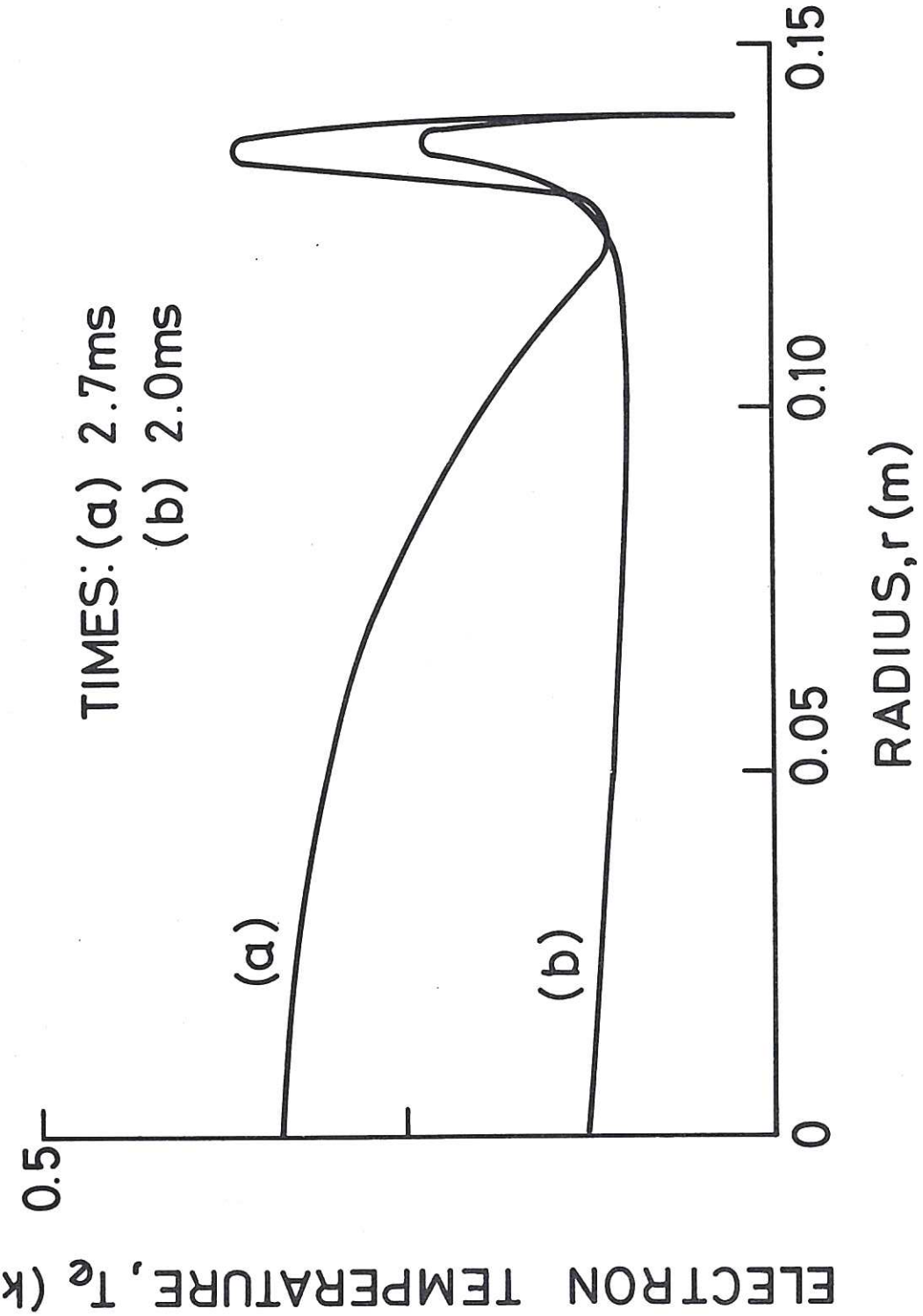


Fig.8 The radial profiles of the electron temperature, T_e, for conditions applicable to a high density discharge in a Small Tokamak ($R = 1.09$m, $a = 0.14$m, $\langle n \rangle = 1.65 \times 10^{19}$ m^{-3}, $I_\varphi = 60$ kA, $B_\varphi = 3.7$ T), given by the code, ICARUS, to indicate the neoclassical "skin-effect" when (a) a 5% density pedestal is maintained at $r = a$ for all time, and (b) the bremsstrahlung radiation losses are enhanced by a factor of 10000.0.

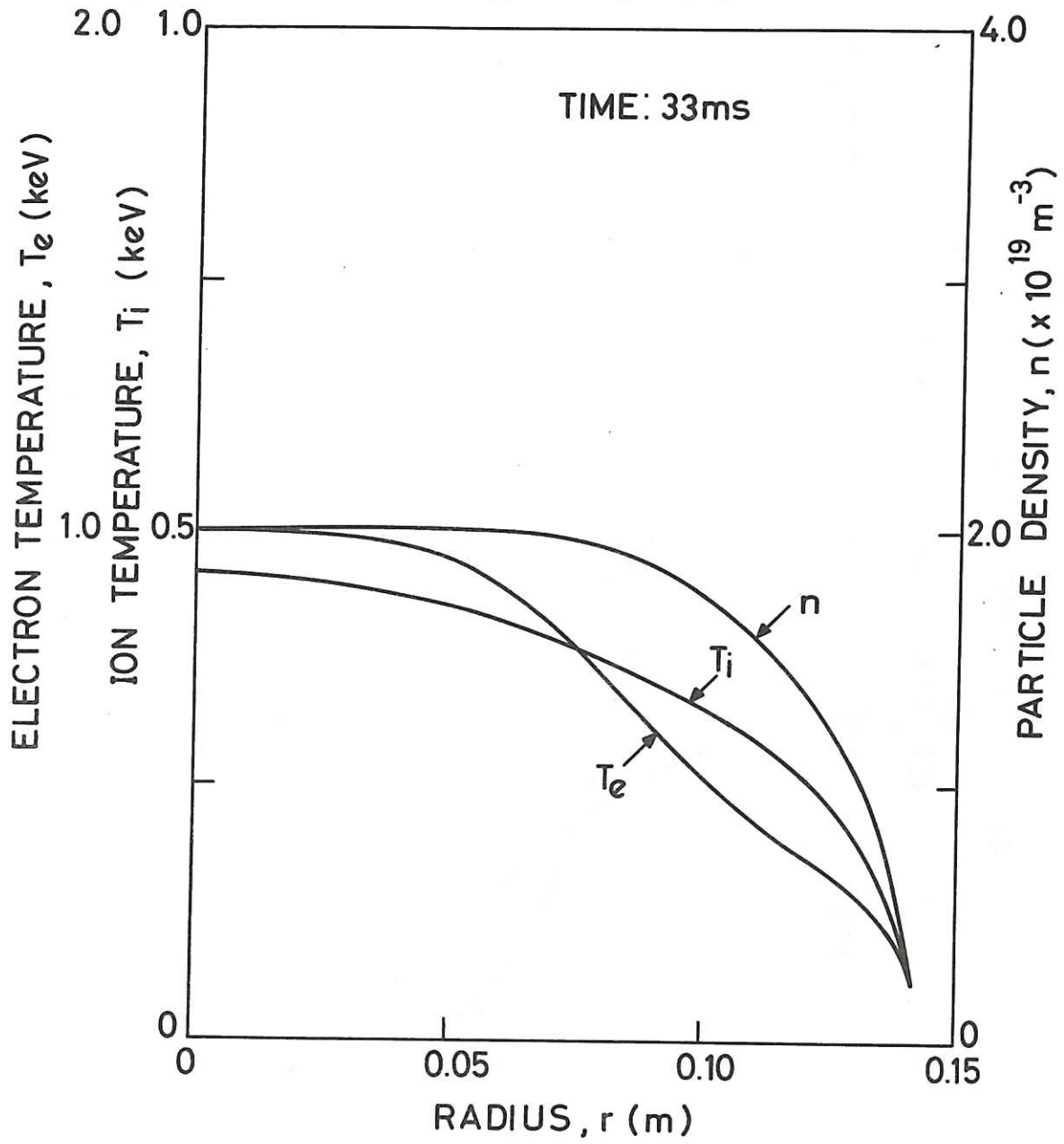


Fig.9 The radial profiles of particle density, n, electron temperature, T_e, ion temperature, T_i, for conditions applicable to a high density discharge in a Small Tokamak (R = 1.09m, a = 0.14m, $\langle n \rangle = 1.65 \times 10^{19} m^{-3}$, I_φ = 60 kA, B_φ = 3.7 T) given at 33 ms by the code, ICARUS, using the pseudo-classical diffusion model and a "neutral species" represented by m = 2 and E_I = 40 eV.

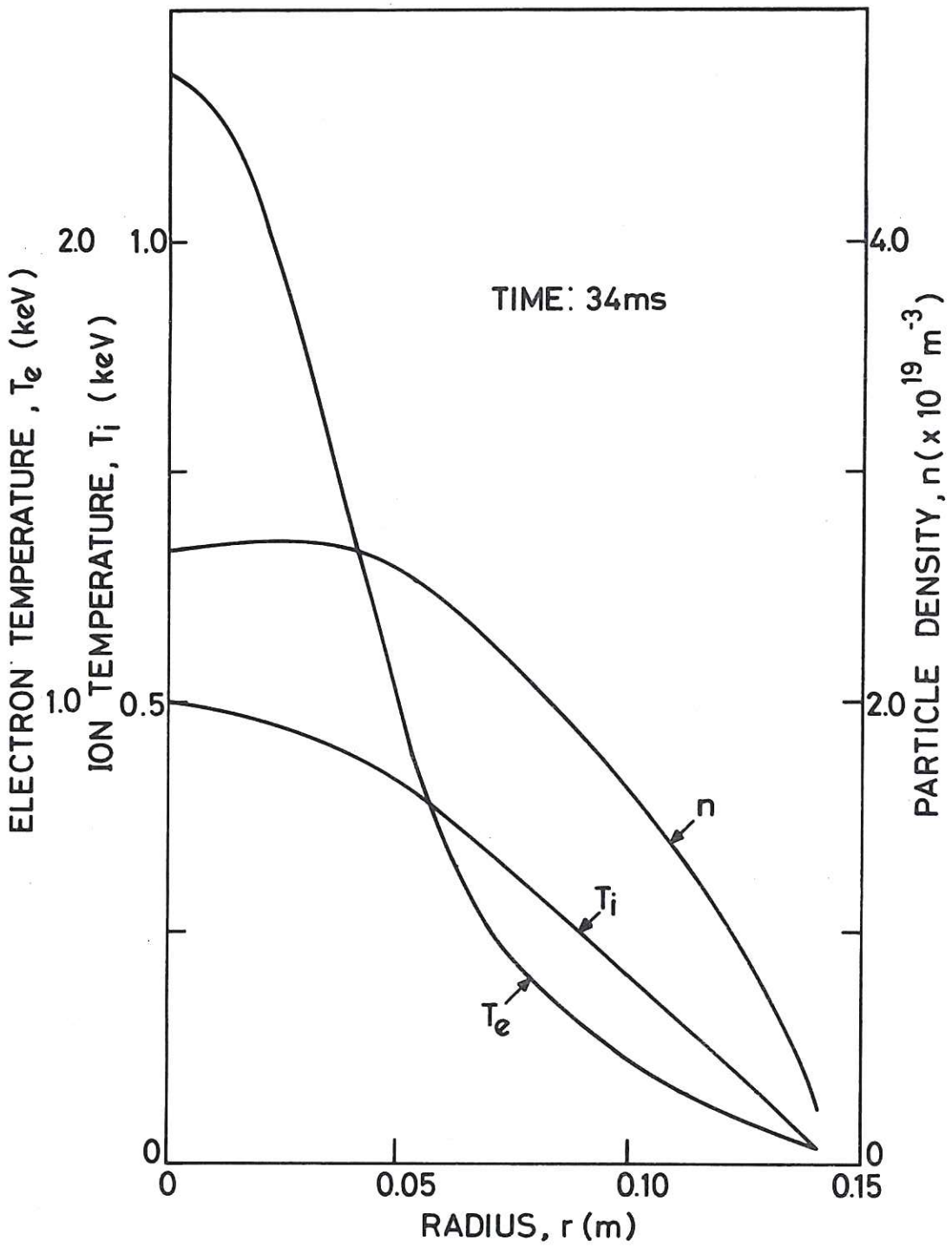


Fig.10 The radial profiles of particle density, n, electron temperature, T_e, ion temperature, T_i, for conditions applicable to a high density discharge in a Small Tokamak ($R = 1.09$ m, $a = 0.14$m, $\langle n \rangle = 1.65 \times 10^{19} m^{-3}$, $I_\varphi = 60$ kA, $B_\varphi = 3.7$ T) given at 34 ms by the code, ICARUS, using the Pfirsch-Schlüter diffusion model with $\lambda = 200.0$ and a "neutral species" represented by $m = 2$ and $E_I = 40$ eV.

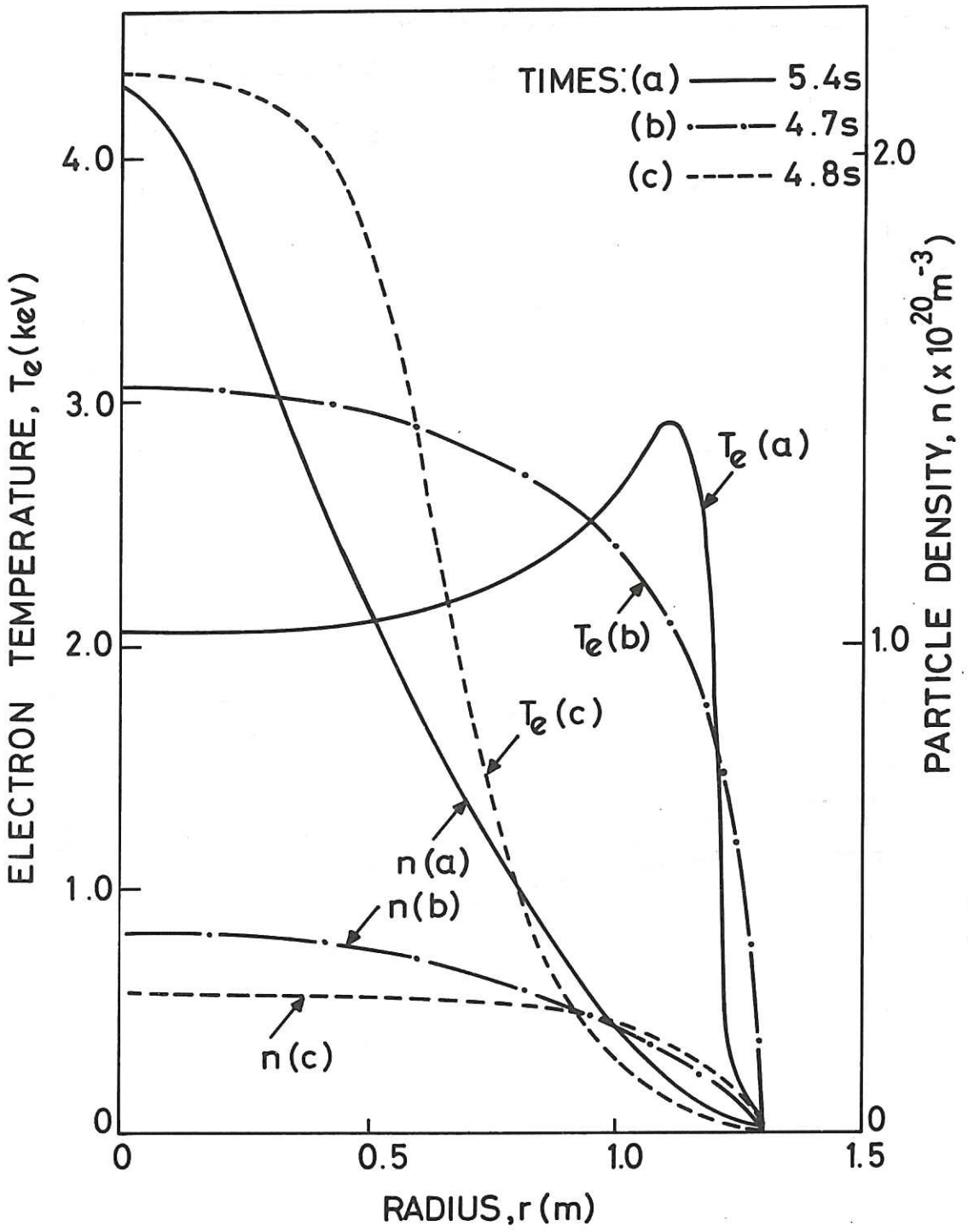


Fig.11 The radial profiles of number density, n, and electron temperature, T_e, for a large Tokamak (R = 2.82m, a = 1.28m, $\langle n \rangle = 5.0 \times 10^{19}$, I_φ = 3MA, B_φ = 3 T). The results are obtained by the code, ICARUS, using

(a) the neoclassical model (—)

(b) the pseudo-classical model (—·—)

(c) the Pfirsch-Schlüter model, with λ = 200.0 (----).

In (b) and (c) the particle content is maintained constant by a "neutral species" represented by m = 2 and E_I = 40 eV.

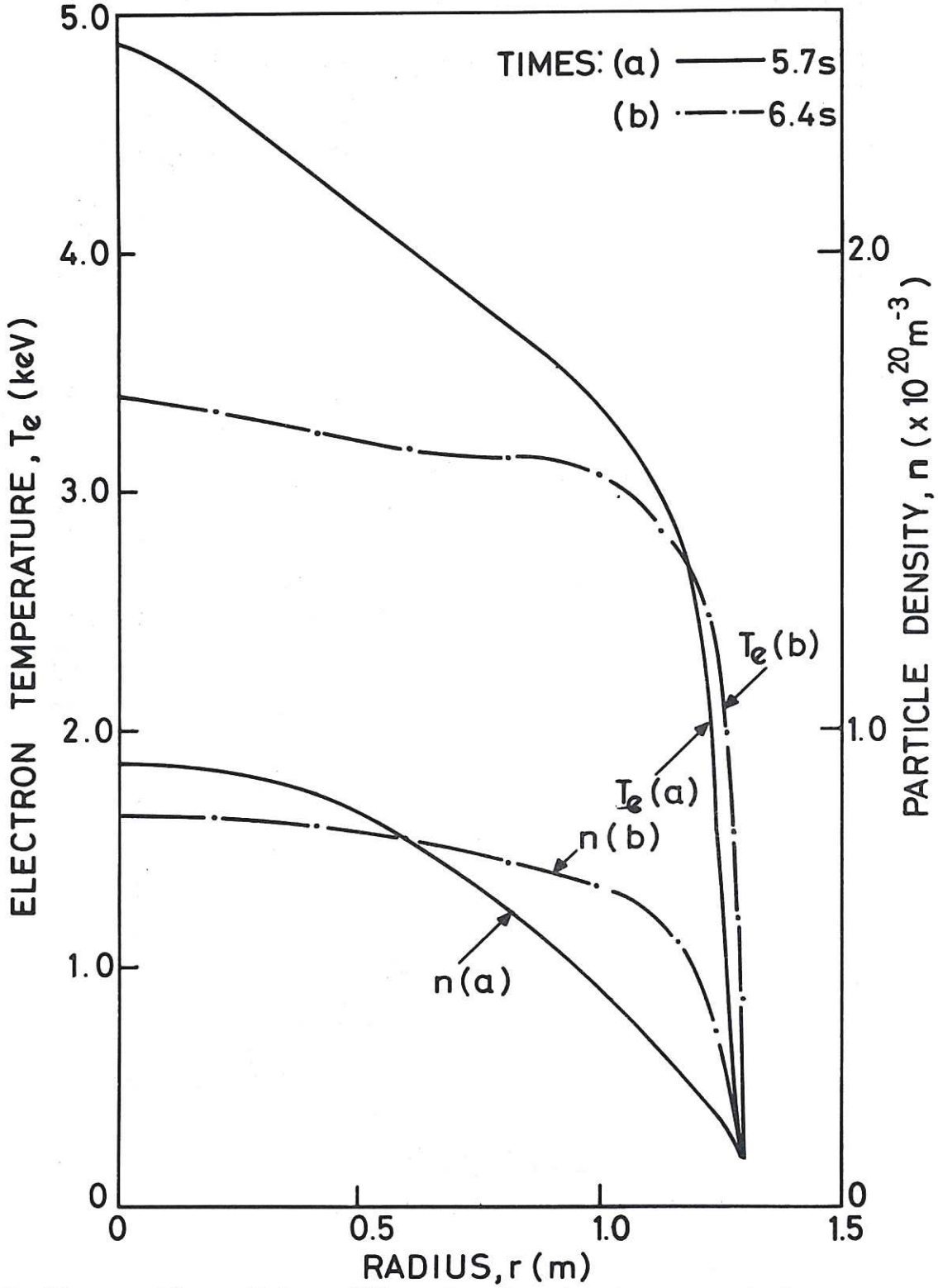


Fig.12 The radial profiles of number density, n, and electron temperature, T_e, for a large Tokamak (R = 2.82m, a = 1.28m, $\langle n \rangle = 5.0 \times 10^{19}$ m^{-3}, I_φ = 3MA, B_φ = 3 T). The results are obtained by the code, ICARUS, using the pseudo-classical diffusion model in which the particle content is maintained constant by a "neutral species" represented by m = 2 and E_I = 40 eV, and increased by the injection of a beam of fast neutrals at (a) high energies (E_f = 80 keV, I_f = 10A, ——), (b) low energies (E_f = 20 keV, I_f = 40 A, —·—).