JAERI-M
5697

NUMERICAL SIMULATION OF THE TOKAMAK PLASMA WITH A MOVABLE BOUNDARY

May 1974

Tatsuoki TAKEDA and Satoshi ITOH*

日本原子力研究所
Japan Atomic Energy Research Institute

この報告書は，日本原子力研究所が JAERI-M レポートとして，不定期に刊行している研究報告書です．入手，複製などのお問合わせは，日本原子力研究所技術情報部（茨城県那珂郡東海村）あて，お申しこしください．

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

JAERI-M 5697

Numerical Simulation of the Tokamak Plasma with a Movable Boundary

Tatsuoki TAKEDA and Satoshi ITOH*

Thermonuclear Fusion Lab., Tokai, JAERI

(Received April 23, 1974)

A numerical simulation of the tokamak plasma with the magnetohydrodynamic fluid model is described. The code has the following two applications: interpretation of the experiment of a tokamak plasma with a dynamic limiter, and comparison of the experimental with the theoretical results obtained with various transport theories. For the former application position of the boundary in the code is movable. For the latter main part of the code can be written automatically in a formula manipulation language (IBM-FORMAC). Examples of the numerical calculation are also given.

---

* Present Address: Institute of Plasma Physics, Nagoya University, Nagoya

JAERI－M 5697

可動リミターを備えたトカマク・プラズマの計算機シミュレイション

日本原子力研究所東海研究所核融合研究室

竹田辰興・伊藤智之*

（1974年4月23日受理）

流体モデルによるトカマク・プラズマの計算機シミュレイションについて述べられている。この計算機コードは主として，次に示す二つの目的の為に作成された。まず，第一に，可動リミターを備えたトカマク・プラズマの実験の解釈の為であり，次に，種々の輸送理論から得られる結果と実験データの比較の為である。第一の目的の為に，このコードでは，境界の位置を可動にしてある。第二の目的の為に，本コードの主要部分は，数式処理言語（IBM－FORMAC）を用いて，自動的に作成できるようになっている。数値計算の結果について，いくつかの例が示されている。

＊ 名古屋大学プラズマ研

目　次　な　し

1. Introduction

The main objective of the computer simulation of a tokamak plasma described in this article is twofold. Firstly, we aim at the comparison of the results obtained by the simulation and an experiment in which a dynamic limiter[1] is used. For this purpose the model for the code includes ionization and charge exchange processes and the position of the boundary for the numerical calculation is movable. Secondly, we aim at seeking the way how we can get agreement between various quantities obtained by the simulation and the experiment, especially, by taking various kinds of atomic processes into account. For this purpose we used an "Automatic Code Generator" to write the main part of the simulation code, in order to rewrite the code easily when we would like to modify the set of the differential equations.

As for the work in reference to the first objective, Matsuda wrote a computer code[2] and studied a plasma behaviour numerically. In the code, however, the plasma current was assumed to be constant and the source term of the particle diffusion was neglected. Therefore, it is not satisfactory to use the code for comparison of the results of the simulation with the experimental results obtained in JFT-2 tokamak. In extending the code to more general one we improved the code in the following two points. First, we chose the mesh point at the position of the dynamic limiter as the boundary of the calculation, though in Ref.(2), the mesh point of the boundary was fixed and anomaly factors were multiplied to the transport coefficients of the plasma in the shade of the dynamic limiter. By virtue of this improvement the code can be applied to the simulation where a plasma current changes violently. Second, we used the recursion formula to solve the resulting linear simultaneous equations in order to reduce the computational time and the size of the core area.

As for the second objective a lot of computer codes have been developed

by many authors to simulate the spatial and temporal evolution of plasma behaviour in a tokamak. For the simulation of the basic process of a plasma it is desirable to develop a computer code with a particle model. At present, however, no satisfactory computer code with a particle model is found to describe a plasma with high temperature and high density such as a tokamak plasma. Therefore, the above-mentioned codes and the one which will be described in this article are also fluid codes. In the field of tokamak experiments this kind of the computer code was first developed by Dnestrovskii et al.[3] and Luc et al.[4] to explain the experimental results of T-3 tokamak. Though the codes were written on a basis of a simple model, they succeeded in explaining the behaviour of the macroscopic quantities of a tokamak plasma qualitatively. Thereafter, stimulated by the success of them and, moreover, by the remarkably good experimental results obtained with the T-3 and ST tokamaks, efforts were made to explain, quantitatively, the experimental results by the computer simulation. For example, the codes by the originators[5,6] became more accurate and Düchs et al.[7] began to study the plasma behaviour by a new code taking account of more complicated formula for the transport processes[8]. Other authors such as Barnett et al. of Oak Ridge National Laboratory[9] also wrote simulation codes for the tokamaks of their laboratories. There remain, however, two important problems concerning the contradictions between the results of the experiments and the computer simulations. The first one is that there are anomalously large energy losses in the experimental results comparing with the results of the computer simulation when we look at the behaviour of the macroscopic quantities such as energy confinement time. The second problem is that the spatial distributions of the plasma quantities obtained experimentally seems different from those obtained by the computer simulation. They have close relations with each other. When we compare the results of the experiments with the computer simulations several points should be noted.

As for the computer simulation it is very important to note that in the simulation codes we use theoretically obtained transport coefficients which were derived under some assumptions and some approximations and are not necessarily confirmed by experimental results. Another important thing concerning the simulation is the fact that we do not know the initial condition of the system because in the initial phase of the tokamak discharge there occurs too complex processes to be predicted theoretically. As for the experimental results, the most important thing is that it is, in general, very difficult and, sometimes, impossible to measure the spatial distributions of plasma parameters. By this fact even a macroscopic quantity which does not depend on the spatial coordinates includes large measuring errors, if it depends, implicitly, on the spatial distributions of plasma parameters. Though many computer codes have been written with due regard to the above things, it is inevitable that there remain some ambiguities on the choice of the input parameters of the simulation code. On the basis of the present transport thories, usually three kinds of factors are introduced to explain the discrepancies between the results of the simulations and the experiments, that is,anomalies due to some unknown mechanisms which are not considered in the transport theories, the effects of the impurity ions with high atomic number and the effect of neutral particles. We wrote a computer code taking notice of the importance of the third one of the above factors and tried to compare the experimental results obtained by JFT-2(JAERI tokamak)[10,11] with the results of the simulation. Because one of our aims is to simulate the plasma behaviour when the dynamic limiter is driven to move, it is especially important to solve the diffusion equations with ionization term.

In section 2 the model of the calculation is described. Section 3 describes the method of the calculation briefly. Examples of the solutions are presented in section 4 and section 5 is devoted to the discussions on the problems of the code, though the detailed discussions on the comparison

between the results of the simulation and the experiment of the dynamic limiter are presented in a separate paper. For the convenience of the users of the computer code the manual of the code is included in Appendix.

2. Model of the Calculation

(2-1) Basic Equations

The plasma to be simulated is described by the particle and energy balance equations and Maxwell equation. Though we are interested in the behaviour of a toroidal plasma, we adopt a cylindrical plasma as the model of the calculation assuming that it is a good approximation to shift the toroidal effect to the transport coefficients. Recently theoretical works concerning the neoclassical transport processes are carried out intensively by many authors and the complicated and precise formula for the process is reported by Hinton et al.[12] We adopted, however, simplified transport formulas[7] in the code for the sake of the easiness of interpretation of the results of the simulation.

The set of the equations consists of four diffusion equations and an auxiliary equation which describes the relation between the poloidal magnetic field and the toroidal plasma current. The four diffusion equations are those for the electric current, the electron density, and the energies of the electronic and ionic components. By solving the equations we can get the spatial and temporal evolutions of five quantities, that is, the poloidal magnetic field(B_p), the electron density(n_e), the electron temperature(T_e), the ion temperature(T_i) and the toroidal plasma current(j_z). The equations are,

$$\frac{\partial}{\partial t}B_p - \frac{1}{\mu_0}\frac{\partial}{\partial x}\left(\frac{\eta}{x}\frac{\partial}{\partial x}xB_p\right) = 0 \qquad (1)$$

$$\frac{\partial}{\partial t}n_e - \frac{1}{x}\frac{\partial}{\partial x}\left(xD_p\frac{\partial}{\partial x}n_e\right) - \langle\sigma v\rangle_{ion}n_e n_n = 0 \qquad (2)$$

$$\frac{3}{2}\frac{\partial}{\partial t}T_e - \frac{1}{n_e}\frac{1}{x}\frac{\partial}{\partial x}\left(xK_e\frac{\partial}{\partial x}T_e\right) - \frac{1}{x}T_e\frac{\partial}{\partial x}\left(xD_{pn_e}\frac{1}{\partial x}\frac{\partial}{\partial x}n_e\right) - \frac{1}{\tau_{eq}}(T_i - T_e)$$

$$- \frac{k'}{n_e}\eta j^2 + \langle\sigma\upsilon Q\rangle_{ion}n_n + \langle\sigma\upsilon Q\rangle_{ex}n_i + \langle\sigma\upsilon Q\rangle_{br}n_i = 0, \qquad (3)$$

$$\frac{3}{2}\frac{\partial}{\partial t}T_i - \frac{1}{n_e}\frac{1}{x}\frac{\partial}{\partial x}\left(xK_i\frac{\partial}{\partial x}T_i\right) - \frac{1}{x}T_i\frac{\partial}{\partial x}\left(xD_{pn_i}\frac{1}{\partial x}\frac{\partial}{\partial x}n_i\right) - \frac{1}{\tau_{eq}}(T_e - T_i)$$

$$+ \langle\sigma\upsilon Q\rangle_{ch.ex}n_n = 0, \qquad (4)$$

$$j - \frac{1}{\mu_0}\frac{1}{x}\frac{\partial}{\partial x}(xB_p) = 0, \qquad (5)$$

where η is the plasma resistivity, τ_{eq} is the energy equipartition time between the electronic and ionic components, D_p is the diffusion coefficient, K_s(s=electron or ion) is the thermal conductivity, n_n is the neutral particle density, $\langle\sigma\upsilon\rangle_j$ and $\langle\sigma\upsilon Q\rangle_j$ are the reaction rate and energy loss rate for the process j (j= ionization, excitation, bremsstrahlung, and charge exchange processes) averaged over the Maxwellian distribution. The functional forms of the above quantities are summarized in Appendix. Though the different symbols are used for the electron density(n_e) and the ion density(n_i), as a matter of convenience, they have in fact the common value in the code and the effect of impurity ions with high atomic number is not included.

(2-2) Correction on the Reaction Rates

One of the important things of the study is to write a computer code which includes a source term in the diffusion equation of the particle density. It requires, however, a long computing time to solve the equations with the source term because the time constant of the process is very short

for the neutral particles to be ionized or charge-exchanged and it requires a very small time step to solve, numerically, the difference equation derived from the diffusion equation. Therefore we modified the rate coefficients for the ionization ($\langle\sigma v\rangle_{ion}$) and charge exchange ($\langle\sigma v\rangle_{ch.ex}$) processes after the following consideration. To simplify the discussion we take account of only the ionization process and consider the following equation for the temporal evolution of the neutral particle density (n_n),

$$\frac{d}{dt}n_n = - \langle\sigma v\rangle_{ion} n_e n_n + \Phi(t,x), \tag{6}$$

where it should be noted that the spatial distribution of the neutral particles is determined only by the ionization process and the neutral influx (Φ) whose functional form is given in advance, in other words, the diffusion of the neutral particles is not taken into account. Assuming that the time interval (Δt) is small and the rate coefficient and the influx do not change considerably during the time interval, we can solve Eq.(6) analytically and we get the increment of the neutral particle density for the time interval as,

$$\Delta n_n = n_n(t)\{\exp(-\langle\sigma v\rangle_{ion} n_e \Delta t) - 1\}$$

$$- \frac{\Phi}{\langle\sigma v\rangle_{ion} n_e}\{\exp(-\langle\sigma v\rangle_{ion} n_e \Delta t) - 1\}. \tag{7}$$

From the above equation we find that the following modified rate coefficient $\langle\sigma v\rangle^*_{ion}$ can be used instead of the rate coefficient $\langle\sigma v\rangle_{ion}$.

$$\langle\sigma v\rangle^*_{ion} = \frac{1}{n_e}\{(1 - e^{-\langle\sigma v\rangle_{ion} n_e \Delta t})(1 - \frac{\Phi}{\langle\sigma v\rangle_{ion} n_e n_n}) + \frac{\Phi \Delta t}{n_n}\}\frac{1}{\Delta t} \quad . \tag{8}$$

As the sink of the neutral particles due to the charge exchange process has the same functional dependence on n_n, the same kind of modification can be applied on the rate coefficient of the process and finally the following equations are obtained for the two processes.

$$\langle\sigma v\rangle^{*}_{ion} = \frac{\langle\sigma v\rangle_{ion}}{\langle\sigma v\rangle_{ion}+\alpha_c\langle\sigma v\rangle_{ch.ex}} \frac{1}{n_e}\{(1-e^{-(\langle\sigma v\rangle_{ion}+\alpha_c\langle\sigma v\rangle_{ch.ex})n_e\Delta t}) \times(1-\frac{\Phi}{(\langle\sigma v\rangle_{ion}+\alpha_c\langle\sigma v\rangle_{ch.ex})n_e n_n}) + \frac{1}{n_n}\Phi\,\Delta t\}\frac{1}{\Delta t}, \quad (9)$$

$$\langle\sigma v\rangle^{*}_{ch.ex} = \frac{\alpha_c\langle\sigma v\rangle_{ch.ex}}{\langle\sigma v\rangle_{ion}+\alpha_c\langle\sigma v\rangle_{ch.ex}} \frac{1}{n_e}\{(1-e^{-(\langle\sigma v\rangle_{ion}+\alpha_c\langle\sigma v\rangle_{ch.ex})n_e\Delta t}) \times(1-\frac{\Phi}{(\langle\sigma v\rangle_{ion}+\alpha_c\langle\sigma v\rangle_{ch.ex})n_e n_n}) + \frac{1}{n_n}\Phi\,\Delta t\}\frac{1}{\Delta t}, \quad (10)$$

where α_c is the ratio of the neutral particle loss after the charge exchange process.

(2-3) Influx of Neutral Particles

It is conjectured that the neutral particles play an important role in the energy loss process of the tokamak plasma. And without considering the effect of the ionization and charge exchange processes of the neutral particles we can not afford to discuss the behaviour of the plasma when the dynamic limiter is removed. It is, however, shown in the previous subsection that to solve the ionization process numerically is a time consuming task. Therefore, we decided to give a spatial and temporal dependence of the neutral particle influx externally. The influx is given as,

$$\Phi(t,x) = F(x)\phi(t), \quad (11)$$

where $\phi(t)$ is the neutral particle influx per unit volume (particles $m^{-3}sec^{-1}$) averaged over the plasma cross section, and $F(x)$ is the spatial distribution of the influx. The former is given numerically as input data for the code and the latter is given analytically as an exponentially decaying function of the spatial coordinate as,

$$F(x) = \frac{\hat{F}(x)}{2\frac{\lambda x_\ell}{x_p^2} - 2\frac{\lambda^2}{x_p^2}(1-e^{-x_\ell/\lambda}) + 1 - \frac{x_\ell^2}{x_p^2}}, \qquad (12)$$

$$F(x) = \exp\{-(x_\ell - x)/\lambda\} \qquad (13)$$

where λ is the penetration length of the neutral particles, x_ℓ and x_p are the radii of the limiter and the boundary of the calculation, respectively.

(2-4) Motion of the Plasma boundary

To simulate the plasma in a tokamak with a dynamic limiter, we introduce two kinds of boundaries for the calculation. One is a fixed boundary which defines the bounds of the region where mesh points are defined and the other is a movable boundary defined by the edge of the limiter. The positions of the fixed and the movable boundaries are denoted by x_p and x_ℓ, respectively. The region inside the fixed boundary is divided into $N_{max}-1$ meshes and for each time step the calculation is carried out for $[(N_{max}-1)\, x_\ell/x_p]+1$ mesh points (inside the movable boundary), where [A] denotes an integer n which satisfies the relation $n-1 < A \leq n$. The boundary conditions for the unknown variables are given at the inner boundary ($x=x_\ell$) and the values outside the inner boundary are not calculated in the code.

3. Method of Calculation

The implicit method[13,14,15] is used to derive the difference equations corresponding to the differential equations (Eqs.(1-5)). After deriving the difference equations we reduce the size of the matrix for the linear simultaneous equations by the recursion formula given by Richtmyer and Morton[16,17] The above process is carried out automatically by the "Automatic Code Generator" written in IBM-FORMAC language. The idea to write automatically a plasma simulation code of the kind by FORMAC was first published by Rosen and Okabayashi[18]. Before we wrote the simulation code of the article we also wrote the "Automatic Code Generator". Using the code generator we can easily modify the simulation code and write a new code from a set of differntial equations of different forms. As the detailed description of the method of the calculation especially stressed on the "Automatic Code Generator" is given in a separate paper, only a necessary description of the method of the calculation is presented briefly in this section.

(3-1) Formula to Derive a Difference Equation

To derive a differnce equation from a differntial equation a rule of replacement of a derivative into a difference of a variable is introduced. Spatial derivatives of a variable $\bar{A}$ at a mesh point (n) are replaced by differences of variables at mesh points (n-1, n and n+1) as,

$$\frac{\partial}{\partial x}\bar{A} = \frac{1}{2\Delta x}\{\bar{A}(n-1) - \bar{A}(n+1)\} , \tag{14}$$

$$\frac{\partial^2}{\partial x^2}\bar{A} = \frac{1}{(\Delta x)^2}\{\bar{A}(n-1) + \bar{A}(n+1) - 2\bar{A}(n)\} , \tag{15}$$

where Δx is the distance between the two adjacent spatial mesh points.

And the temporal derivative of a variable $\tilde{A}$ is given as,

$$\frac{\partial}{\partial t}\tilde{A} = \frac{1}{\Delta t}\{A(t+\Delta t) - A(t)\} , \tag{16}$$

where it should be noted that all the variables are estimated at time $t+\frac{1}{2}\Delta t$ The symbol $\tilde{A}$ is used in Eqs.(14-16) to stress that the value of A is estimated at time $t+\frac{1}{2}\Delta t$. It should be, also, noted that the number of the outermost mesh point ($x=x_p$; the surface of the cylinder) is 1 and that of the innermost one ($x=0$; the axis of the cylinder) is N_{max}.

Next we must introduce a rule for linearization of the equation consisting of M unknown variables because the temporal mesh point for the time $t+\frac{1}{2}\Delta t$ is a virtual one and we must reduce each term estimated at $t+\frac{1}{2}\Delta t$ to a linear combination of values at t and ones at $t+\Delta t$. For this purpose we consider an arbitrary nonlinear term consisting of a product of arbitrary combination of M variables, A_1, A_2,.....A_M, and we denote the nonlinear term by $f(A_1,A_2,...A_M)$. Then we linearize the nonlinear term as follows,

$$\tilde{f}(A_1,A_2,.....A_M) = f(A_1(t),A_2(t),.....A_M(t)) + \frac{1}{2}\Delta t \sum_{i=1}^{M} \frac{\partial f}{\partial A_i}\Big|_t \frac{\partial}{\partial t}\tilde{A}_i , \tag{17}$$

where it should be observed that $\tilde{f}$ is an approximate value of $f(t+\frac{1}{2}\Delta t)$. Moreover we pose the following condition,

$$(\widetilde{A_iA_j}) = \tilde{A}_i\tilde{A}_j. \tag{18}$$

By the above condition, nonlinear term of more general form, that is,

$f(A_1, A_2, \ldots A_M; \frac{\partial}{\partial t}A_1, \frac{\partial}{\partial t}A_2, \ldots \frac{\partial}{\partial t}A_M)$ can be also linearized and the operations of the linearization and the replacement of the differential operaor become commutable.

Then we get M linear relations for the variables at $t+\Delta t$ as,

$$\sum_{j=1}^{M} \{\alpha_{ij}(n)A_j(n+1)+\beta_{ij}(n)A_j(n)+\gamma_{ij}(n)A_j(n-1)\}+\delta_i(n) = 0, \quad (19)$$

$$(i = 1,2,\ldots M)$$

where i and j are the numbers of the differential equations and the variables, respectively, $\alpha_{ij}(n)$, $\beta_{ij}(n)$, $\gamma_{ij}(n)$ and $\delta_i(n)$ are the coefficients estimated at the spatial mesh point (n) which are the functions of the variables estimated at time t.

(3-2) Boundary Conditions

To solve the difference equation (Eq.(19)) two kinds of boundary conditions are imposed at the plasma surface and the plasma axis. At the plasma surface the boundary conditions are,

$$A_j(1) = a_j , \qquad (j = 1,2,\ldots M) \qquad (20)$$

where a_j's are given as input data and for simplicity it is assumed that the plasma surface coincides with the fixed boundary (n=1). On the other hand at the plasma axis ($n=N_{max}$) the boundary conditions are given according to the parity of the variables as,

$$A_j(N_{max}+1) = (-1)^{P_j} A_j(N_{max}-1), \qquad (21)$$

where $P_j = \begin{cases} 0, & \text{if } A_j \text{ is even at } n=N_{max}, \\ 1, & \text{if } A_j \text{ is odd at } n=N_{max}. \end{cases}$

$(j = 1,2,.....M)$

In our case it is evident that among the five variables, only the first one (B_p) has an odd parity.

(3-3) Recursion Formula

To solve the linear simultaneous equations (Eq.(19)), it is not an efficient way to calculate the matrix with $M(N_{max}+1) \times M(N_{max}+1)$ elements[17,18] Therefore we used the recursion formula of Richtmyer. The value of the variable A_j at (n+1)-th mesh point is represented by the linear combination of the values of A_k (k=1,2,...M) at n-th mesh point as,

$$A_j(n+1) = \sum_{k=1}^{M} E_{jk}(n)A_k(n) + F_j(n), \qquad (22)$$

where $E_{jk}(n)$ and $F_j(n)$ are the coupling constants independent of A_j's. Then we substitute the representation (Eq.(22)) into Eq.(19) and we obtain the following equation,

$$\sum_{j=1}^{M} \Big[\sum_{r=1}^{M} \sum_{m=1}^{M} \alpha_{ir}E_{rm}(n)E_{mj}(n-1) + \sum_{r=1}^{M} \beta_{ir}E_{rj}(n-1) + \gamma_{ij} \Big] A_j(n-1)$$

$$+ \Big[\sum_{r=1}^{M} \sum_{m=1}^{M} \alpha_{ir}E_{rm}(n)F_m(n-1) + \sum_{m=1}^{M} \beta_{im}F_m(n-1)$$

$$+ \delta_i(n) + \sum_{m=1}^{M} \alpha_{im}F_m(n) \Big] = 0. \qquad (i=1,2,...M) \qquad (23)$$

The above equation is an identical equation and it should hold for arbitrary

values of $A_j(n-1)$. Therefore, the coefficients of A_j's and the constant terms are identically zero as,

$$\sum_{r=1}^{M}\sum_{m=1}^{M}\alpha_{ir}E_{rm}(n)E_{mj}(n-1) + \sum_{r=1}^{M}\beta_{ir}E_{rj}(n-1) + \gamma_{ij} = 0, \quad (24)$$

$(i = 1,2,.....M \text{ and } j = 1,2,.....M)$

$$\sum_{r=1}^{M}\sum_{m=1}^{M}\alpha_{ir}E_{rm}(n)F_m(n-1) + \sum_{m=1}^{M}\beta_{im}F_m(n-1)$$

$$+ \delta_i(n) + \sum_{m=1}^{M}\alpha_{im}F_m(n) = 0. \quad (25)$$

$(i = 1,2,.....M)$

From these equations $E_{mj}(n-1)$ and $F_m(n-1)$ are derived if $E_{mj}(n)$ and $F_m(n)$ are known beforehand. Therefore, if the values of $E_{mj}(N_{max}-1)$ and $F_m(N_{max}-1)$ are given, the values of $E_{mj}(n)$ and $F_m(n)$ at $n = N_{max}-2, N_{max}-3,.....2,1$ are determined succesively using the above equations. On the other hand, $E_{mj}(N_{max}-1)$ and $F_m(N_{max}-1)$ are derived from the boundary conditions at the plasma axis. Substituting Eq.(22) into the boundary condition (Eq.(21)) we obtain the following equations,

$$\sum_{k=1}^{M}E_{jk}(N_{max})E_{kl}(N_{max}-1) = (-1)^{P_j}\delta_{jl}, \quad (26)$$

$$\sum_{k=1}^{M}E_{jk}(N_{max})F_k(N_{max}-1) + F_j(N_{max}) = 0, \quad (27)$$

where $\delta_{jl}=1$ for $j=1$ and $=-1$ for $j\neq 1$.

From Eqs.(24 and 26) and Eqs.(25 and 27) the following two equations are derived, respectively,

$$\alpha_{ij}(-1)^{P_j} + \sum_{r=1}^{M} \beta_{ir} E_{rj}(N_{max}-1) + \gamma_{ij} = 0, \tag{28}$$

$$\sum_{m=1}^{M} \beta_{im} F_m(N_{max}-1) + \delta_i(N_{max}) = 0 . \tag{29}$$

By solving the above equations we can get the matrix $E_{mj}(N_{max}-1)$ and $F_m(N_{max}-1)$.

Thus instead of a set of linear simultaneous equations with $M(N_{max}+1)$ variables we obtain $(N_{max}+1)$ sets of linear simultaneous equations each with M variables. These equations are summarized as,

$$[C(n)]E_j(n-1) + G_j(n) = 0, \tag{30}$$

$$[C(n)]F(n-1) + D(n) = 0, \tag{31}$$

$$[\beta(N_{max})]E_j(N_{max}-1) + A_j(N_{max}) = 0, \tag{32}$$

$$[\beta(N_{max})]F(N_{max}-1) + D(N_{max}) = 0, \tag{33}$$

where

$${}^{t}E_j(n) = [E_{1j}(n), E_{2j}(n), \ldots . E_{Mj}(n)], \tag{34}$$

$${}^{t}F(n) = [F_1(n), F_2(n), \ldots . F_M(n)], \tag{35}$$

$${}^{t}A_j(n) = [\alpha_{1j}(n)(-1)^{P_j+\gamma_{1j}}(n), \ldots . \alpha_{Mj}(n)(-1)^{P_j+\gamma_{Mj}}(n) \tag{36}$$

$$ {}^{t}D_j(n) = [\delta_1(n), \delta_2(n), \ldots\ldots \delta_M(n)], \tag{37} $$

$$ {}^{t}G_j(n) = [\gamma_{1j}(n), \gamma_{2j}(n), \ldots\ldots \gamma_{Mj}(n)], \tag{38} $$

$$ [\alpha(n)] = \begin{Bmatrix} \alpha_{11}(n), \ldots\ldots\ldots \alpha_{1M}(n) \\ \ldots\ldots\ldots \\ \alpha_{M1}(n), \ldots\ldots\ldots \alpha_{MM}(n) \end{Bmatrix}, \tag{39} $$

$$ [\beta(n)] = \begin{Bmatrix} \beta_{11}(n), \ldots\ldots\ldots \beta_{1M}(n) \\ \ldots\ldots\ldots \\ \beta_{M1}(n), \ldots\ldots\ldots \beta_{MM}(n) \end{Bmatrix}, \tag{40} $$

$$ [C(n)] = \begin{Bmatrix} \sum_{r=1}^{M} \alpha_{1r}(n)E_{r1}(n)+\beta_{11}(n), \ldots\ldots & \sum_{r=1}^{M} \alpha_{1r}(n)E_{rM}(n)+\beta_{1M}(n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \sum_{r=1}^{M} \alpha_{Mr}(n)E_{r1}(n)+\beta_{M1}(n), \ldots\ldots & \sum_{r=1}^{M} \alpha_{Mr}(n)E_{rM}(n)+\beta_{MM}(n) \end{Bmatrix}, \tag{41} $$

for $n=1,2,\ldots\ldots N_{max}-1$, and $j=1,2,\ldots\ldots M$.

The output of the "Automatic Code Generator" is a deck of a FORTRAN cards of matrix elements corresponding to the coefficients of the above equations. We can get a simulation code by inserting the cards into a prepared set of a subprogram. In the code the quantities $A_j(n)$'s are derived for the boundary conditions (Eq.(20)), using Eq.(22)[19].

(3-4) Apparent Singularities of Diffusion Terms at the Origin

There appears apparent singularities at the plasma axis because of the factor of 1/x, if we replace simply the diffusion type equations by the difference equations. To avoid the singularities, we expand the terms by power series of x near the plasma axis and we get the following equations.

$$\lim_{x\to 0} \frac{\partial}{\partial x}\left[\frac{n}{x}\frac{\partial}{\partial x} x B_p\right] = 0, \tag{42}$$

$$\lim_{x\to 0} \frac{1}{x}\frac{\partial}{\partial x}\left[x D_p \frac{\partial}{\partial x} n_e\right] = 2D_p(0)\frac{\partial^2 n_e}{\partial x^2}\bigg|_{x=0}, \tag{43}$$

$$\lim_{x\to 0} \frac{1}{x} T_e \frac{\partial}{\partial x}\left[\frac{x D_p}{n_e}\frac{\partial}{\partial x} n_e\right] = 2\frac{T_e}{n_e} D_p(0)\frac{\partial^2 n_e}{\partial x^2}\bigg|_{x=0}. \tag{44}$$

The same kinds of equations hold also for the case of thermal diffusion.

(3-5) Determination of the Time Step Width

The time step width is automatically determined by examining the increments of the variables for each time step. The criteria for the determination of the step width are summarized in Table 1.

4. Examples of Solution

In this section we present two examples of the results of the numerical simulation. One is the simulation of the JFT-2 plasma with normal fixed limiter and the other is the simulation of the plasma which is obtained using the dynamic limiter. Both the results are preliminary and the more detailed descriptions of the results are given in a separate paper. It should be noted that in choosing the input parameters there remains a lot of degree of freedom because we can compare the distributions of the physical quantities only indirectly between the experimental results and the computational results.

(4-1) Simulation of the Plasma in JFT-2 with Normal Fixed Limiter

In this subsection one of the typical discharges of JFT-2[11] reproduced by the computer simulation is presented. The discharge conditions are as follows; P_f=1.6 x 10^{-4} Torr H_2, B_0=1.0 Wb/m^2, B_v=0.012 Wb/m^2 and the radius of the limiter is 0.25 m. The waveforms of the loop voltage and the plasma current obtained experimentally are shown in Fig.1. For the computer simulation the waveform of the plasma current was approximated by a piecewise linear function as shown in Fig.2. In the preliminary calculation only the effect of the neutral influx was especially noticed. Case 1 (Fig.3) is the resultof the simulation for the spatially averaged neutral influx of 3 x 10^{20} $m^{-3}sec^{-1}$ and case 2 (Fig.4) is that for the influx of 1 x 10^{20} $m^{-3}sec^{-1}$. From these figures it is found that the plasma parameter which is most sensitive to the magnitude of the neutral influx is the spatial distribution of the electron density. The loss flux and the increment of the electron density due to the ionization process is nearly balanced when the averaged neutral influx is 1 x 10^{20} $m^{-3}sec^{-1}$ and

in this case the stationary state of the density distribution is nearly attained.

The reason why the electron and ion temperatures are higher than those obtained experimentally[11] will be partly explained by the fact that the calculation is carried out for the effective charge of unity and the collison frequency falls in the banana region of the neoclassical transport theory.

(4-2) Simulation of the Plasma Obtained Using the Dynamic Limiter

The dynamic limiter which is installed in JFT-2 tokamak is a quickly removable limiter. It can be triggered to move at any time during the discharge. It is driven to move by a pair of air cylinders and the maximum velocity of the limiter is about 10 m/sec. The main purpose of the experiment of the dynamic limiter is to investigate whether the equilibrium and the stability of the plasma is sustained when the plasma is insulated from the external solid material (fixed) limiter or not[20,21]. The transport phenomena such as particle diffusion will be also investigated experimentally using the dynamic limiter. These experiments are, however, very difficult because available diagnostic methods are not sufficient to get a spatial distribution of the plasma parameters. Therefore we analyze the experimental data by comparing them with the results of the numerical simulation. In the following four examples of the results of the simulation (cases a, b, c, and d) are presented, which are all preliminary ones and show qualitatively the effect of the neutral particles and the magnitude of the transport coefficient. The case a is the reference case where anomalous transport coefficients are not taken into account, and the neutral particle influx is $3 \times 10^{20}\ m^{-3}sec^{-1}$. The case b shows the result for the condition that the neutral particle influx is reduced to $1 \times 10^{20} m^{-3}sec^{-1}$ when the dynamic limiter is removed. The cases c and d show the results for the condition

that the anomaly factors of 5 and 10 are multiplied to the neoclassical electron thermal conductivity of the plasma in case b. Figures 5,6 and 7 are the waveform of the plasma current, the position of the dynamic limiter and the mean neutral influx, respectively. Figures 8, 9, 10 and 11 are the distributions of the plasma parameters for the above four cases, respectively. The time behaviour of the averaged temperatures is given Fig.12. From these figures it can be generally said that the distributions of the plasma parameters change more slowly than the speed of the dynamic limiter when the neutral influx is 1×10^{20} $m^{-3}sec^{-1}$ and the anomaly factor of the transport coefficients is not taken into account. Comparing the density distribution of case a with that of case b we find that tha abrupt change of density distribution in case a after the dynamic limiter is removed is due to the ionization process of the neutrals. The fact that the distribution of the current density does not change considerably when the dynamic limiter is removed seems to contradict with the experimental result which shows that the hot region of the plasma moves with the edge of the dynamic limiter. So far as the distribution of the electron temperature is concerned, however, the experimental results are reproduced easily by choosing a large anomaly factor for the electron thermal conductivity (Figs. 10 and 11).

5. Discussions

In writing the computer code several simplifications and approximations are introduced and there remain some problems and restrictions for usage of the code.

Firstly the transport process in the code is based on the simplified neoclassical theory. The particle loss and the thermal loss are described only by the diffusion coefficient and the thermal conductivity. Therefore if the processes are well described by the neoclassical theory, the complete neoclassical process should be included in the code. It is, however, very easy to rewrite the code by using the "Automatic Code Generator".

Secondly the code is written on the basis of the two fluid model (electron and ion) and the neutral particle influx is given as an external function. Therefore the three fluid model should be adopted if the behaviour of the neutral particles is the important purpose of the calculation.
In the same way, many fluids model should be adopted if the behaviour of the impurity ions is analyzed by using the computer code.

Because of the above mentioned facts and some other reasons, the results of the computer calculation do not fully agree with the experimental results. The code is, however, very useful to analyze the separate phenomena in detail.

Acknowledgments

The authors wish to express their sincere thanks to Messers. T. Sugawara, N. Suzuki and K. Toi for the discussions on the experiments of the dynamic limiter. Thanks are also due to Mr. T. Tazima and Dr. M. Tanaka for the stimulating discussions on the simulation code. We are pleased to thank Dr. S. Mori for the continueing encouragement on the work.

Appendix A. Fundamental Quantities Used in the Simulation Code

Functional forms of the quantities which appear in the simulation code are summarized in the following. The temperature is expressed in eV and the other quantities are expressed in MKS unit.

[1] Plasma resistivity (η)

$$\eta = 5.98 \times 10^{-5} \ln\Lambda \, T_e^{-3/2} \ , \tag{A-1}$$

where

$$\ln\Lambda = \begin{cases} 23.4 - 1.15 \log_{10}(n_e \times 10^{-6}) + 3.45 \log_{10} T_e, & \text{for } T_e < 50 \text{ eV}, \\ 25.3 - 1.15 \log_{10}(n_e \times 10^{-6}) + 2.30 \log_{10} T_e, & \text{for } T_e > 50 \text{ eV}. \end{cases} \tag{A-2}$$

[2] Energy equipartition time (τ_{eq})

$$\tau_{eq} = 3.30 \times 10^{14} \, T_e^{3/2} / (n_e \ln\Lambda). \tag{A-3}$$

[3] Particle diffusion coefficient and thermal conductivity (D_p and K_s)

$$D_p = \begin{cases} \beta_{pb} \gamma_p \varepsilon^{-3/2} q^2 \nu_e \rho_e^2 \ , & \text{for } \nu_e < \nu_{1e}, \\ \beta_{pp} \gamma_p R^{-1} q \, v_{Te} \rho_e^2 \ , & \text{for } \nu_{1e} < \nu_e < \nu_{2e} \\ \beta_{pc} \gamma_p \, q^2 \nu_e \rho_e^2 \ , & \text{for } \nu_{2e} < \nu_e \end{cases} \tag{A-4}$$

and

$$K_s = \begin{cases} \beta_{kbs}\gamma_{ks}\varepsilon^{-3/2}q^2\nu_s\rho_s^2 n_e , & \text{for } \nu_s < \nu_{1s} \\ \beta_{kps}\gamma_{ks}R^{-1}q\, v_{Ts}\rho_s^2\, n_e , & \text{for } \nu_{1s} < \nu_s < \nu_{2s} \\ \beta_{kcs}\gamma_{ks}\, q^2\nu_s\rho_s^2 n_e , & \text{for } \nu_{2s} < \nu_s \end{cases} \quad \text{(A-5)}$$

where $\varepsilon = x/R$, $q = xB_o/(RB_p)$,

$$\rho_s = \begin{cases} 3.37 \times 10^{-6}\, B_o^{-1}\, T_e^{1/2}, & \text{for } s = e \\ 1.45 \times 10^{-4}\, B_o^{-1}\, T_i^{1/2}, & \text{for } s = i \end{cases} \quad \text{(A-6)}$$

$$\nu_s = \begin{cases} 1.00 \times 10^{-12}\, \ln\Lambda\, n_e\, T_e^{-3/2}, & \text{for } s = e \\ 7.15 \times 10^{-14}\, \ln\Lambda\, n_e\, T_i^{-3/2}, & \text{for } s = i \end{cases} \quad \text{(A-7)}$$

$$\nu_{1s} = \varepsilon^{3/2} R^{-1} q^{-1} v_{Ts}, \quad \text{for } s = e, i \quad \text{(A-8)}$$

$$\nu_{2s} = R^{-1} q^{-1} v_{Ts}, \quad \text{for } s = e, i \quad \text{(A-9)}$$

$$v_{Ts} = \begin{cases} 5.95 \times 10^5\, T_e^{1/2}, & \text{for } s = e \\ 1.36 \times 10^4\, T_i^{1/2}, & \text{for } s = i \end{cases} \quad \text{(A-10)}$$

β_{pb}, β_{pp}, β_{pc}, β_{kbe}, β_{kpe}, β_{kce}, β_{kbi}, β_{kpi}, β_{kci} are the numerical constants and the following values are adopted for them throughout the calculation.

$\beta_{pb} = 1.60$, $\beta_{pp} = 1.25$, $\beta_{pc} = 1.00$,

$\beta_{kbe} = 1.81$, $\beta_{kpe} = 3.00$, $\beta_{kce} = 2.33$,

$\beta_{kbi} = 0.48$, $\beta_{kpi} = 3.00$, $\beta_{kci} = 0.71$,

γ_p, γ_e and γ_i are anomaly factors which are defined by a subprogram.

[4] Ionization rate coefficient[22)]

$$\langle\sigma v\rangle_{ion} = 4.44 \times 10^{-12}\ b\ h(y)\ T_e^{-3/2}, \qquad \text{(A-11)}$$

where $h(y) = -\ e\ Ei(-y)/y$,

$y = \chi/T_e$,

$Ei(-y) = -\int_y^{\infty} z^{-1}\ e^{-z}\ dz$,

$\chi = 13.595$ eV,

$b = 0.2$.

[5] Excitation loss[22)]

$$\langle\sigma v Q\rangle_{ex}\ n_n = 0.59 \times 10^{-11}\ S(y_{ex})\ T_e^{-1/2}\ n_n n_e/(n_n+n_e)\ , \qquad \text{(A-12)}$$

where $S(y_{ex}) = -\ Ei(-y_{ex})$,

$y_{ex} = \chi_{ex}/T_e$, $\chi_{ex} = 10$.

[6] Charge exchange loss[23)]

$$\langle \sigma v Q \rangle_{ch.ex} n_n = 1.57 \times 10^{-16} \{101.1 - 28.4 \log_{10} T_i + 2 (\log_{10} T_i)^2\} n_n T_i^{3/2} . \quad \text{(A-13)}$$

[7] Bremsstrahlung loss

$$\langle \sigma v Q \rangle_b n_i = 1.05 \times 10^{-19} n_i T_e^{1/2} . \quad \text{(A-14)}$$

Appendix B. Flow chart of the code

The code consists of a main routine (MAIN), 10 principal subroutines and many other subprograms used for determining the coefficients of the difference equations. Figure 13 shows the flow chart of the main part of the code. Table 2 is the list of the subprograms which are not shown explicitly in Fig. 13 Functional form of the transport coefficients and their anomaly factors are changeable by replacing the corresponding subprograms in the list. For the convenience of the replacement of subprograms, the list of the COMMON variables are also presented in Table 3.

Appendix C. Input card format

The input card format and an example of the input data are shown in Tables 4 and 5, respectively. The meaning of the symbols in Table 4 is summarized in Table 6.

References

1) M. Ohta et al.; JAERI-M 5433 (1973), in Japanese.

2) S. Matsuda; JAERI-M 5035 (1972), in Japanese.

3) Y.N. Dnestrovskii, D.P. Kostomarov and N.L. Pavlova; Intern. Symp. on Closed Confinement Systems, Dubna (1969).

4) H. Luc, C. Mercier and Soubbaramayer; Intern. Symp. on Closed Confinement Systems, Dubna (1969).

5) Y.N. Dnestrovskii, D.P. Kostomarov and N.L. Pavlova; Fourth European Conf. on Controlled Fusion and Plasma Physics, Rome (1970) p.17.

6) C. Mercier and Soubbaramayer; Fourth European Conf. on Controlled Fusion and Plasma Physics, Rome (1970) p.16.

7) D.F. Düchs, H.P. Furth and P.H. Rutherford; 4th Intern. Conf. on Controlled Fusion, Madison, vol.I (1971) p.369.

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

9) C.F. Barnett et al.; 4th Intern. Conf. on Controlled Fusion, Madison, vol.I (1971) p.347.

10) S. Itoh et al.; The Third Intern. Conf. on Toroidal Plasma Confinement, (1973) B-4.

11) S. Itoh et al.; JAERI-M 5385 (1973).

12) F.L. Hinton and M.N. Rosenbluth; Phys. Fluids 16 (1973) 836.

13) I. Kawakami; IPPJ-DT-23 (1970).

14) J.P. Boujot, C. Mercier and Soubbaramayer; Computer Physics Communication 4 (1972) 89.

15) T. Tazima, T. Takeda and S. Itoh; JAERI-M 4941 (1972), in Japanese.

16) R.D. Richtmyer and K.W. Morton; in "Difference Methods for Initial-Value Problems" 2nd edition, New York, Interscience, 1967, p.198.

17) D.F. Düchs; NRL Report 7340 (1972).

18) B. Rosen and M. Okabayashi; Nucl. Fusion 13 (1973) 3.

19) Generally speaking, the boundary condition at the plasma surface is given as,

$$\xi_j A_j(1) + \eta_j \frac{1}{\Delta x}[A_j(1) - A_j(2)] = a_j,$$

where ξ_j, η_j, and a_j are given as input data. In this case $A_j(1)$'s are obtained by solving the following linear simultaneous equations,

$$\sum_{k=1}^{M} [\delta_{kj}(\xi_k + \frac{\eta_k}{\Delta x}) - \frac{\eta_j}{\Delta x} E_{jk}(1)]A_k(1) = a_j + \frac{\eta_j}{\Delta x} F_j(1).$$

20) S. Itoh et al.; submitted to Nucl. Fusion.

21) T. Sugawara et al.; submitted to Nucl. Fusion.

22) R.F. Post; Plasma Physics 3 (1961) 273.

23) N. Fujisawa and T. Sugawara; private communication.

Table 1

Table of multiplier (FACT*) to the time step width (DT).

The time step width (DT)is changed to a new value (FACT * DT) by examining the values of the variables (AN and XNN) according to the table.

* Underlined words are symbols used in the FORTRAN program.

	$\exists_{AN < 0}$ or $\exists_{XNN < 0}$	$\exists_{\lvert AN-AO \rvert > \theta_1 * AO}$ and $\forall_{XNN > 0}$	$\forall_{\lvert AN-AO \rvert < \theta_2 * AO}$ and $\forall_{\lvert \Delta XNN \rvert < \omega_2 * XNNO}$ **
$\Delta t < \Delta t_{min}$ (DTMN)	1	1	2
$\Delta t_{min} < \Delta t < \Delta t_{ch}$ (DTCH)	1/2	1	2
$\Delta t_{ch} < \Delta t < \Delta t_{max}$ (DTMX)	1/2	1/2	2
$\Delta t_{max} < \Delta t$	1/2	1/2	1

** XNNO is the old value of XNN

θ_1 = AOMXU, θ_2 = AOMXL, ω_2 = RMX.

Table 2

List of subprograms

NAME	S/F*	ARGUMENTS	FUNCTION
XL	F	TIME, C	Calculate x_1.
OUTET	F	TIME, N, C, AO, X	Dummy.
OUTDP	F	TIME, N, C, AO, X	Dummy.
OUTKE	F	TIME, N, C, AO, X	Dummy.
OUTKI	F	TIME, N, C, AO, X	Dummy.
ETA	F	TIME, N, C, AO, X	Calculate η.
DETA	F	TIME, N, C, AO, X	Calculate η'.
CETAP	F	N, C, AO	Subprograms for ETA and DETA.
DCETAP	F	N, C, AO	
RLN	F	N, AO	Calculate $\ln\Lambda$.
DRLN	F	N, AO	Calculate $(\ln\Lambda)'$.
DP	F	TIME, N, C, AO, X	Calculate D_p.
DDP	F	TIME, N, C, AO, X	Calculate $(D_p)'$.
DPXEQ0	F	N, C, AO, X	Calculate $D_p(N=N_{max})$.
CDP1	F	N, C, EPS15	Subprograms for DP, DDP and DPXEQ0.
DCDP1	F		
CDP2	F		
DCDP2	F		
CDP3	F		
DCDP3	F		
XKE	F	TIME, N, C, AO, X	Calculate K_e.
DXKE	F	TIME, N, C, AO, X	Calculate $(K_e)'$.
XKEX0	F	N, C, AO, X	Calculate $K_e(N=N_{max})$.
CKE1	F	N, C, EPS15	Subprograms for XKE, DXKE and XKEX0.
DCKE1	F		
CKE2	F		
DCKE2	F		
CKE3	F		
DCKE3	F		
XKI	F	TIME, N, C, AO, X	Calculate K_i.
DXKI	F	TIME, N, C, AO, X	Calculate $(K_i)'$.
XKIX0	F	N, C, AO, X	Calculate $K_i(N=N_{max})$.
CKI1	F	N, C, EPS15	Subprograms for XKI, DXKI and XKIX0.
DCKI1	F		
CKI2	F		
DCKI2	F		
CKI3	F		
DCKI3	F		

NAME	S/F	ARGUMENTS	FUNCTION
CRHOE	F	C	Calculate ρ_e.
CNUE	F	N, C, AO	Calculate ν_e.
DCNUE	F	N, C, AO	Calculate $(\nu_e)'$.
CRHOI	F	C	Calculate ρ_i.
CNUI	F	N, C, AO	Calculate ν_i.
DCNUI	F	N, C, AO	Calculate $(\nu_i)'$.
SGVION	F	N, C, AO, TIME, XNN	Calculate $\langle\sigma v\rangle_{ion}$.
CSVION	F	N, C, AO, TIME, XNN	Subprogram for SGVION.
EI	F	Y	Exponential integral.
DEXPI	S	ANS, X, IERR	Subprogram for EI.
TAUEQ	F	N, C, AO	Calculate $1/\tau_{eq}$.
CTEQ	F	N, C, AO	Subprogram for TAUEQ.
QION	F	N, C, AO	Calculate Q_{ion}.
SVGEX	F	N, AO, XNN	Calculate $\langle\sigma v Q\rangle_{ex}$.
EXCITE	F	N, AO	Subprogram for SVGEX.

NAME	S/F	ARGUMENTS	FUNCTION
SVCHEX	F	N, C, AO, TIME, XNN	Calculate $\langle\sigma v Q\rangle_{ch.ex}$.
CSVCHX	F	N, C, AO, TIME, XNN	Subprograms for SVCHEX.
QSVCHX	F	N, C, AO, TIME, XNN	
SGBRM	F	N, C, AO	Calculate $\langle\sigma v Q\rangle_{br}$.
RCOF	F	N, C, AO, TIME, XNN	Subprogram for SGVION and SVCHEX.
PHI	F	N, C, AO, TIME, XNN	Calculate $\Phi(t,x)$.
PHIT	F	TIME	Calculate $\phi(t)$.
GAMM	F	N	Anomaly factors.
GETA	F		
DGETA	F		
GD	F		
DGD	F		
GKE	F		
DGKE	F		
GKI	F		
DGKI	F		

* S: SUBROUTINE subprogram,

F: DOUBLE PRECISION FUNCTION subprogram.

Table 3

List of variables

(1) COMMON variables

LABEL	SYMBOL	MEANING
INP	MMAX	Number of unknown variables.
	NMAX	Number of spatial mesh points.
	NCST	Number of floating constants.
	ICST	Number of integer constants.
	DX	Δx
	DT	Δt
	PDT	$(\Delta t)_{output}$
	CTIME	t_{end}
	NCHECK(10)	Check parameter.
	I5	= 5.
	I6	= 6.
	I7	= 5.
	I8	= 6.
DIM	N0 - N22	Size of arrays.
INQ	PDT0	PDT < PDT0
	PDT1	PDT = PDT2, for t - t_{start} > PDT1.
	PDT2	
STR	STIME	t_{start}
ACC	AOMXL	See Table 1.
	RMX	
	AOMXU	
	DTCH	
	DTMX	
	DTMN	
PCH	PTIME	t_{punch}
SHD	NSHD	= 0
	DXS	= 0
BOU	NSET(8)	Number of time points when the boundary values are given.
	BOUN(8,2,500)	Boundary conditions. BOUN(I,1,J): time, BOUN(I,2,J): boundary values, I = 1: I_p in kA, I = 2: n_e, I = 3: T_e, I = 4: T_i, I = 5: x_ℓ, I = 6: $\phi(t)$.
ANM	GKET	= 1.0
	GKDP	= 1.0
	GKKE	= 1.0
	GKKI	= 1.0
XXX	AXX(5,100)	= A0, see (2).

(2) Arguments of the subprograms

SYMBOL		MEANING/VALUE
TIME		t
N		Number of the mesh point.
X		Radial position of the mesh point.
EPS15		$(x/R)^{3/2}$
AO(1,J)		$B_p(J)$, (old value*).
AO(2,J)		$n_e(J)$
AO(3,J)		$T_e(J)$
AO(4,J)		$T_i(J)$
AO(5,J)		$j_z(J)$
		(*AN's are new values corresponding to AO's)
C(1)	$1/\mu_o$	$= 7.96177 \times 10^5$
C(2)	$x_j(t = t_{start})$	
C(3)	α_c	
C(4)		dummy.
C(5)	x_p	
C(6)	Z	$= 1$
C(7)	C_1	$= 0.58$
C(8)	λ	
C(9)	β_{pb}	
C(10)	β_{pp}	
C(11)	β_{pc}	
C(12)	R	
C(13)	B_o	
C(14)	b	$= 0.2$
C(15)	e	$= 2.71828$
C(16)	A	$= 1.0$
C(17)	χ	$= 13.595$
C(18)	β_{keb}	
C(19)	β_{kep}	
C(20)	β_{kec}	
C(21)	β_{kib}	
C(22)	β_{kip}	
C(23)	β_{kic}	

Table 4. Input card format

CARD NO. 1: Title card
* TITLE(J), J=1,20
* 20A4

CARD NO. 2: Punch control
* PCH
* E12.5

CARD NO. 3: Time mesh control
* AOMXL, RMX, AOMXU, DTCH, DTMX, DTMN
* 6E12.5

CARD NO. 4: Integer constants
* MMAX, NMAX
* 2I3

CARD NO. 5: Geometry
* XLO, XP, R, BO
* 4E12.5

CARD NO. 6: Particle diffusion
* BPB, BPP, BPC
* 3E12.5

CARD NO. 7: Electron thermal conductivity
* BKEB, BKEP, BKEC
* 3E12.5

CARD NO. 8: Ion thermal conductivity
* BKIB, BKIP, BKIC
* 3E12.5

CARD NO. 9: Neutral particles
* PLNG, RCHX, VNTR, ITMX
* 3E12.5, I12

CARD NO.10: Time step
* DX, DT, PDT, STIME, CTIME
* 5E12.5

CARD NO.11: Print control
* PDTO, PDT1, PDT2
* 3E12.5

CARD NO.12: Check parameter
* NCHECK(J), J=1,10
* 10I3

CARD NO.13: No. of points for boundary condition
* NSET(J), J=1,6
* 6I2

CARD NO.14: Boundary conditions
* BOUN(1,1,I), I=1,NSET(1)
  BOUN(1,2,I), I=1,NSET(1)
  BOUN(2,1,I), I=1,NSET(2)
  BOUN(2,2,I), I=1,NSET(2)
  BOUN(3,1,I), I=1,NSET(3)
  BOUN(3,2,I), I=1,NSET(3)
  BOUN(4,1,I), I=1,NSET(4)
  BOUN(4,2,I), I=1,NSET(4)
  BOUN(5,1,I), I=1,NSET(5)
  BOUN(5,2,I), I=1,NSET(5)
  BOUN(6,1,I), I=1,NSET(6)
  BOUN(6,2,I), I=1,NSET(6)
* 6E12.5

CARD NO.15: Anomalous transport
* GGG(J), J=1,18
* 6E12.5

CARD NO.16: Type of initial values
* ITYPE
* I1

CARD NO.17: Initial values for ITYPE=1
* AO(1,J), J=1,NMAX
  AO(2,J), J=1,NMAX
  AO(3,J), J=1,NMAX
  AO(4,J), J=1,NMAX
  AO(5,J), J=1,NMAX
  XNN(J), J=1,NMAX
* 6E12.5

CARD NO.18: Initial values for ITYPE=2+
* AA(2), A1(2), A2(2), SS(2), TT(2)
  AA(3), A1(3), A2(3), SS(3), TT(3)
  AA(4), A1(4), A2(4), SS(4), TT(4)
  AA(5), A1(5), A2(5), SS(5), TT(5)
  AA(6), A1(6), A2(6), SS(6), TT(6)
* 5E12.5

+ Initial distribution of a plasma parameter Y

$$Y = AA + A1 * [\, 1 - (x/x_1)^{SS}] + A2 * [\, 1 - (x/x_1)^2]\, x^{TT}$$

Table 5

```
$DATA
JFT-2 DLX (NORMAL,MOVE) NO.67940 PLATEAU, INFL.=V-TYPE,3.0E+21,ANOM=5
0.250
0.05        0.10        0.10            1.0E-040.003            1.0E-14
  5 26
0.12        0.25        0.9         1.0
1.60        1.25        1.00
1.81        1.66        3.73
0.68        1.33        1.60
0.03        1.00        1.00                20
0.01        0.0001      0.001       0.01        0.145
0.003       0.02        0.005
  0  0  0  0  0  0  0  0  0  0
15 2 6 6 9 5
0.0         0.008       0.01        0.02        0.027       0.03
0.04        0.06        0.07        0.08        0.10        0.12
0.14        0.148       0.15
0.0         0.0         6.8         20.2        26.3        25.0
27.7        30.4        30.4        29.7        26.3        20.3
13.5        7.5         0.0
0.0         0.200
     5.0E+17     5.0E+17
0.0         0.01        0.02        0.04        0.10        0.20
2.0         2.0         10.0        40.0        40.0        10.0
0.0         0.01        0.02        0.04        0.10        0.20
1.0         1.0         5.0         20.0        20.0        5.0
0.0         0.06        0.065       0.07        0.075       0.08
0.085       0.09        0.16
0.12        0.12        0.133       0.160       0.195       0.230
0.240       0.245       0.250
0.0         0.06        0.09        0.12        0.18
    1.00E+21    2.00E+20    2.00E+19    1.00E+20    0.30E+20

0.01        1.0         0.01        1.0         0.01        1.0
0.0         0.0         4.0         0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0         0.0
2
     5.0E+17     1.0E+190.0          2.0         2.0
2.0         5.0         0.0         2.0         2.0
1.0         2.5         0.0         2.0         2.0
0.0         277057.0    0.0         2.0         2.0
     1.0E+13     1.0E+150.0          2.0         2.0
****
$JEND
```

Table 6. Meaning of the symbols in Table 4*

SYMBOL	MEANING
TITLE(J)	Title of the calculation (within 80 characters).
MMAX	Number of unknown variables (=5).
NMAX	Number of spatial mesh points.
XLO	Initial position of the limiter (m).
XP	Position of the outermost mesh point (m).
R	Major radius of the plasma (m).
BO	Toroidal magnetic field (Wb/m^2).
BPB	β_{pb}
BPP	β_{pp}
BPP	β_{pc}
BKEB	β_{keb}
BKEP	β_{kep}
BKEC	β_{kec}
BKIB	β_{kib}
BKIP	β_{kip}
BKIC	β_{kic}
PLNG	Penetration length of the neutrals (initial guess).
RCHX	Loss rate of the charge exchanged neutrals.
VNTR	Temperature of the cold neutrals.
ITMX	Maximum number of iteration for determination of the penetration of the cold neutrals.

* See also Tables 1 and 3.

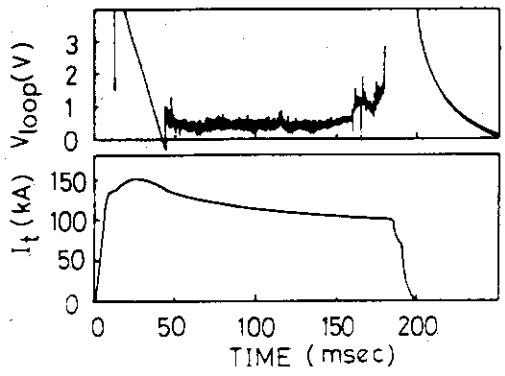


Fig. 1 The waveforms of the plasma current and the loop voltage of the plasma obtained using the normal fixed limiter (500 mmϕ).

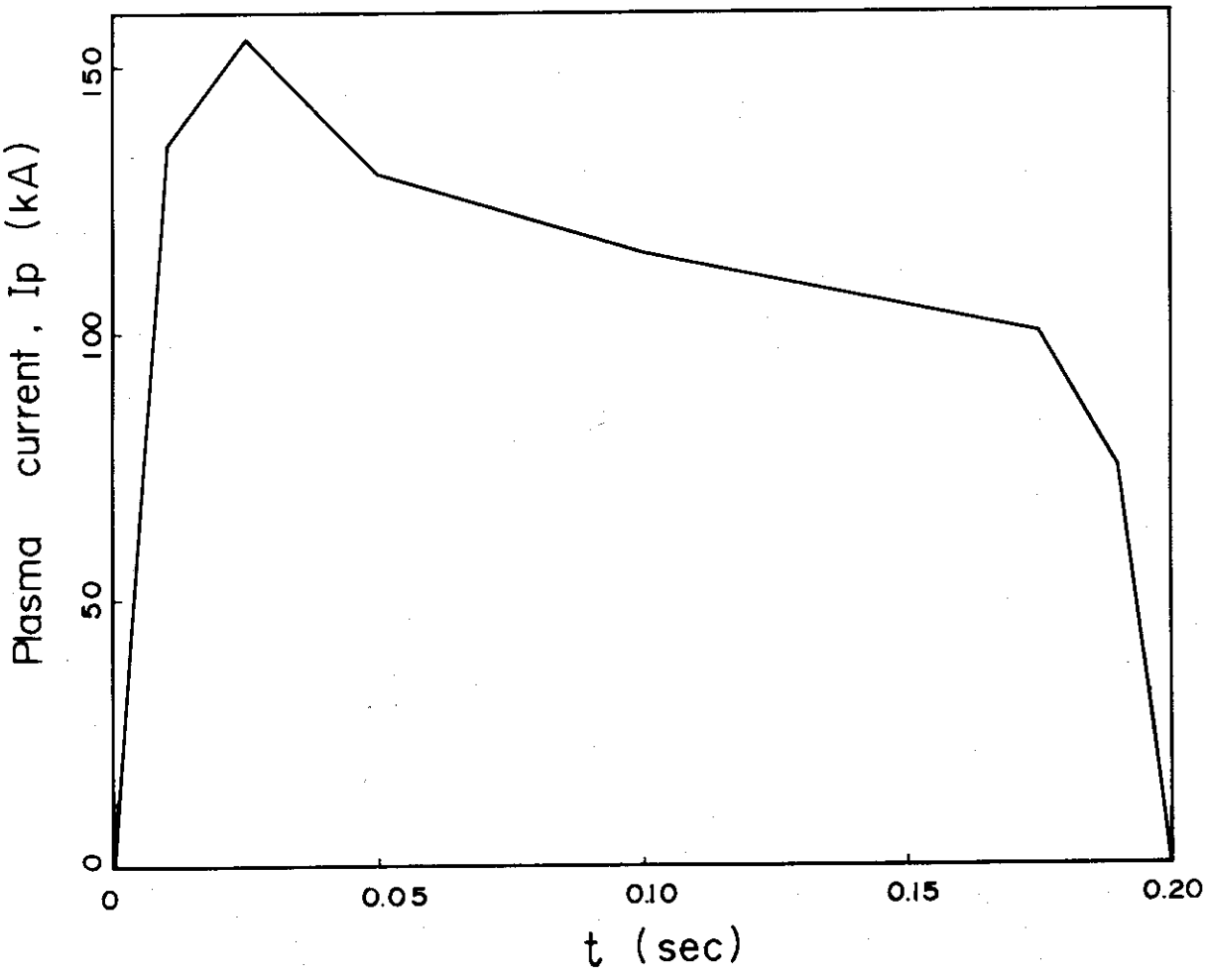


Fig. 2 The waveform of the plasma current expressed by a piecewise linear function for the input data of the code.

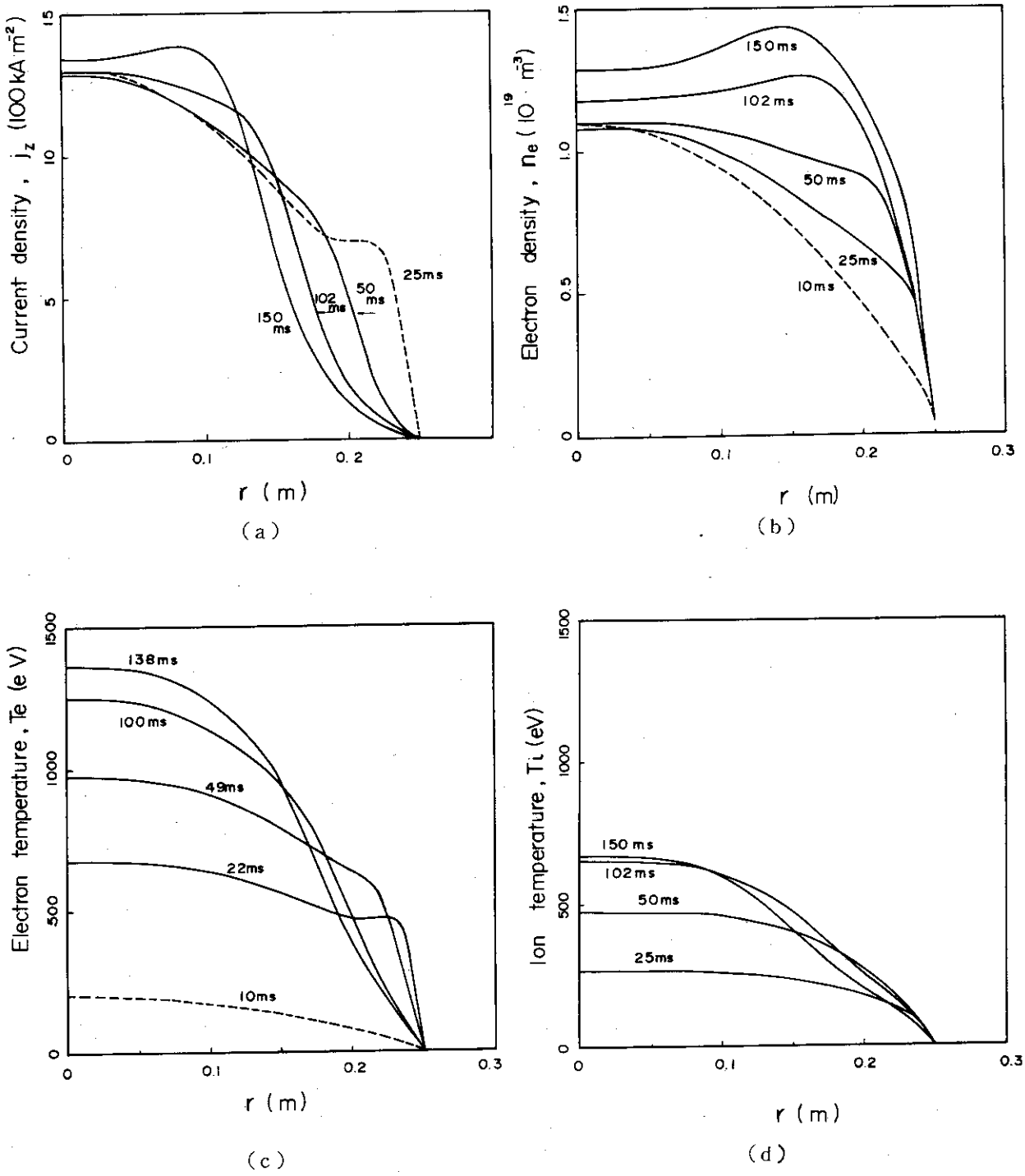


Fig. 3 The result of the computer simulation for the plasma current of Fig. 2 and the neutral particle influx averaged over the plasma cross section of 3×10^{20} m^{-3} sec^{-1} (case 1). Figures 3(a),(b),(c) and (d) show the plasma current density, electron density, electron temperature and ion temperature, respectively.

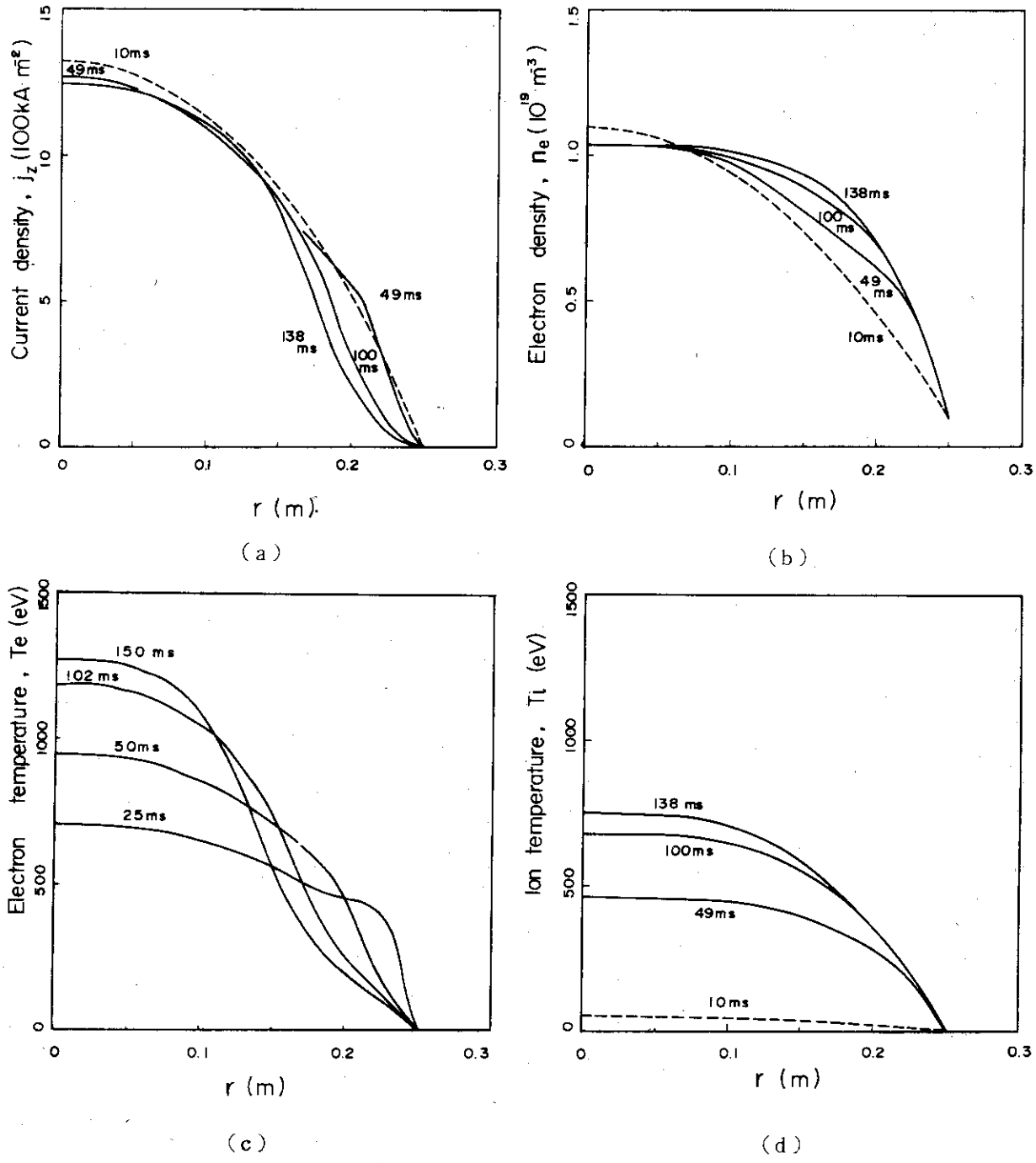


Fig. 4 The result of the computer simulation for the plasma current of Fig. 2 and the neutral particle influx averaged over the plasma cross section of 1×10^{20} m^{-3} sec^{-1} (case 2). Figures 4(a),(b),(c) and (d) show the plasma current density, electron density, electron temperature and ion temperature, respectively.

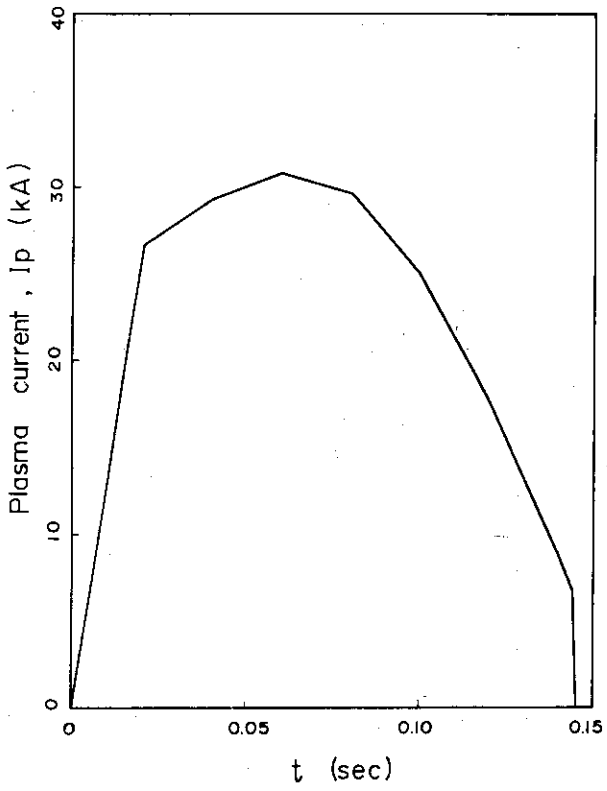


Fig. 5 The plasma current for the simulation of the plasma obtained by using the dynamic limiter.

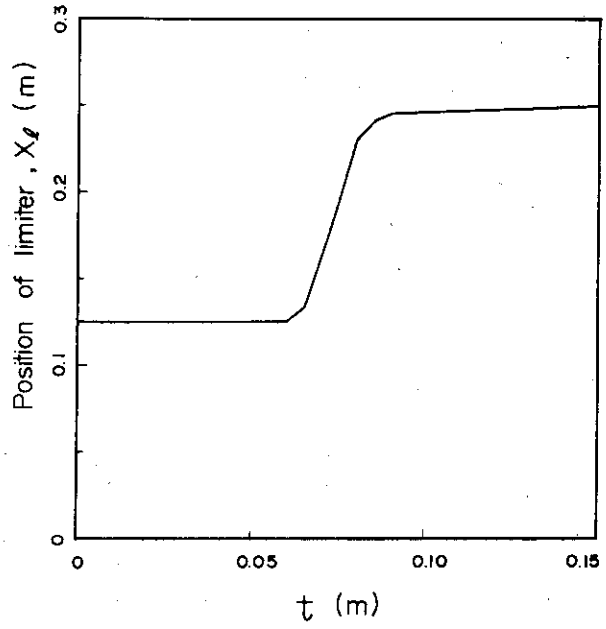


Fig. 6 The position of the dynamic limiter.

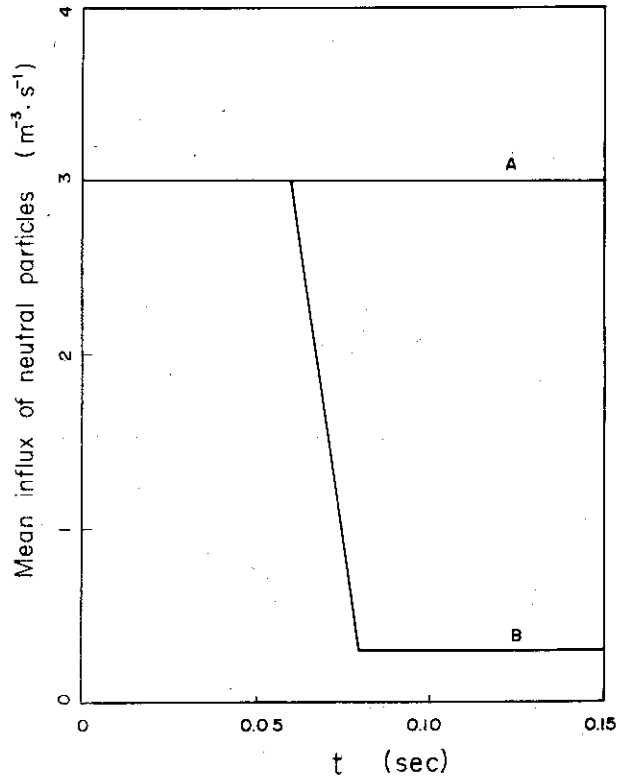


Fig. 7

The assumed time behaviour of the neutral particle influx averaged over the plasma cross section. The curve A is used for the simulation of Fig. 8 and the curve B is used for the simulation of Figs. 9, 10 and 11.

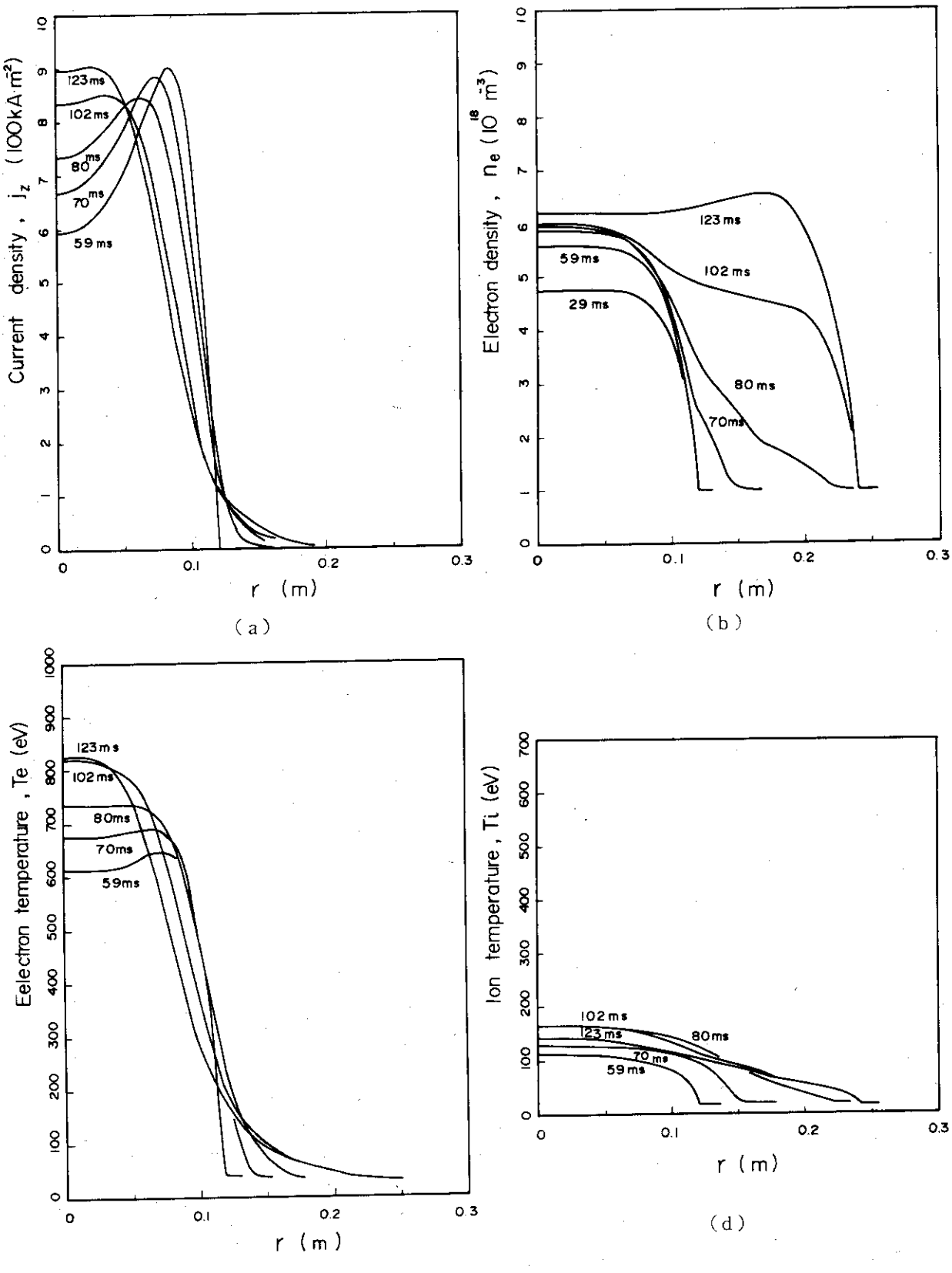


Fig. 8 The distribution of the plasma parameters (a : plasma current density, b : electron density, c : electron temperature and d : ion temperature) for case a.

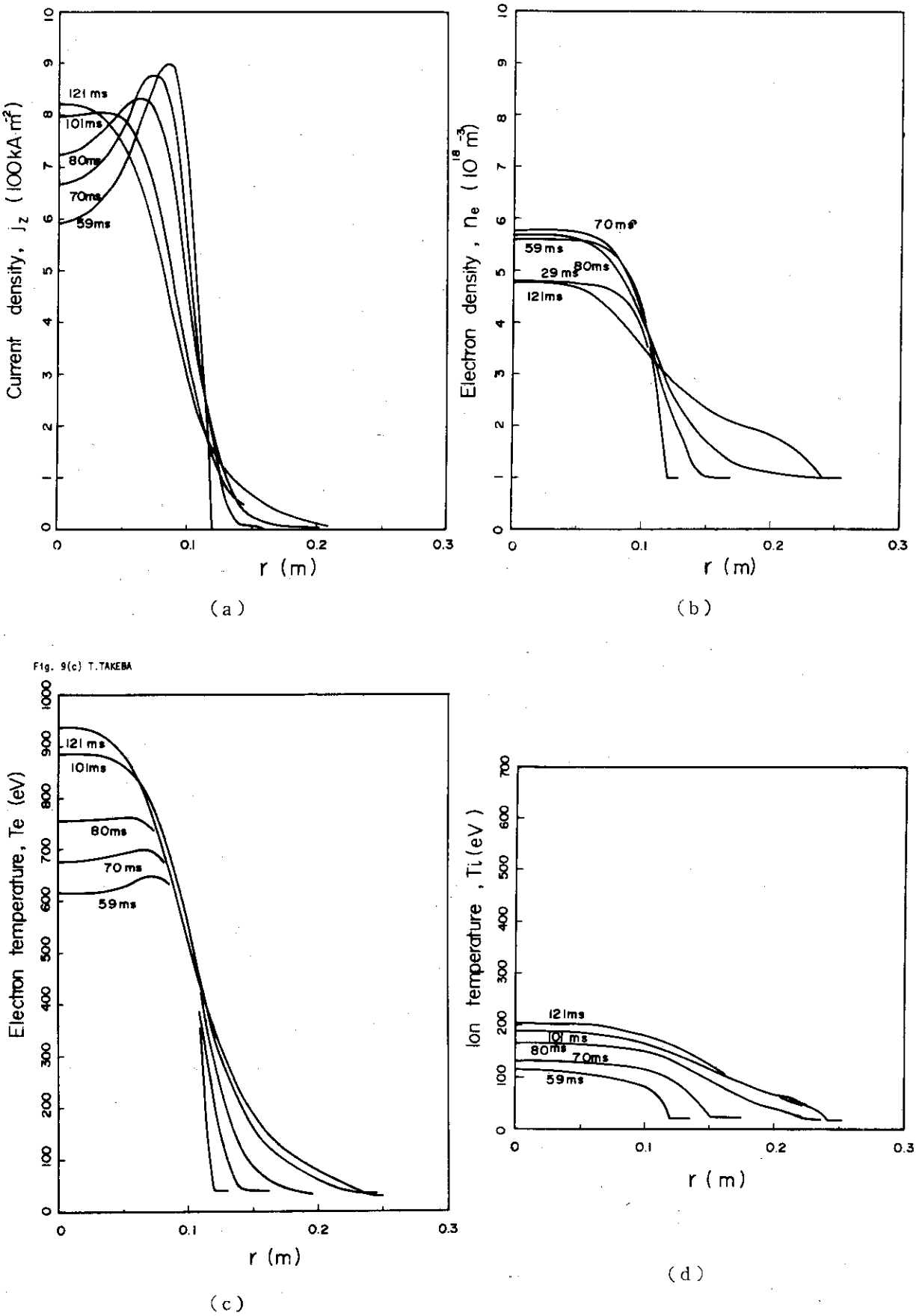


Fig. 9 The distribution of the plasma parameters (a : plasma current density, b : electron density, c : electron temperature and d : ion temperature) for case b.

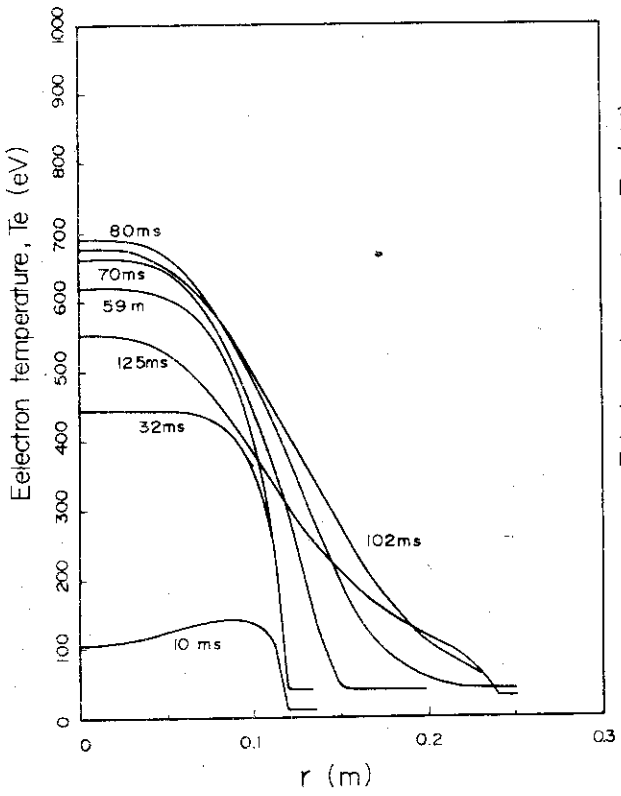


Fig. 10 The distribution of the electron temperature for case c.

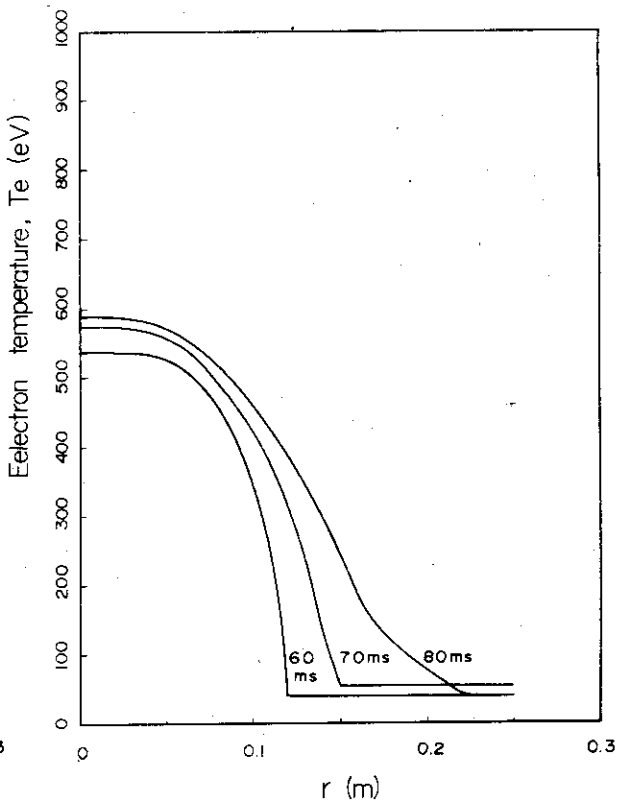


Fig. 11 The distribution of the electron temperature for case d.

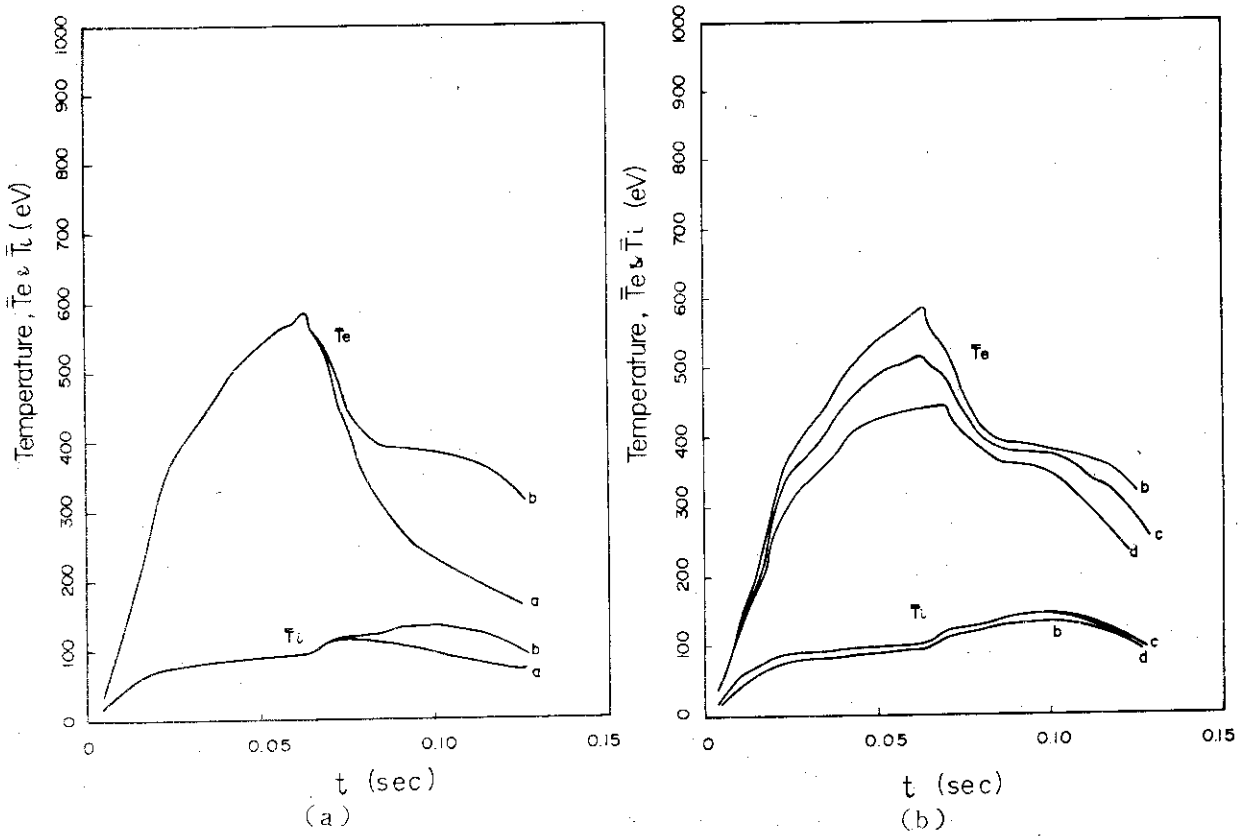
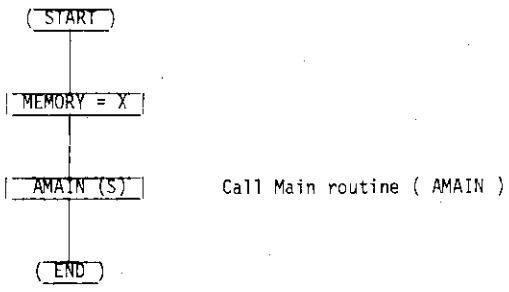


Fig. 12 The time behaviour of the averaged electron temperature for cases a and b (Fig. 12(a)) and for cases b, c and d (Fig. 12(b)).

[1] Dummy Main Routine (MAIN)

* Save memory area.



[2] Main Routine (AMAIN)

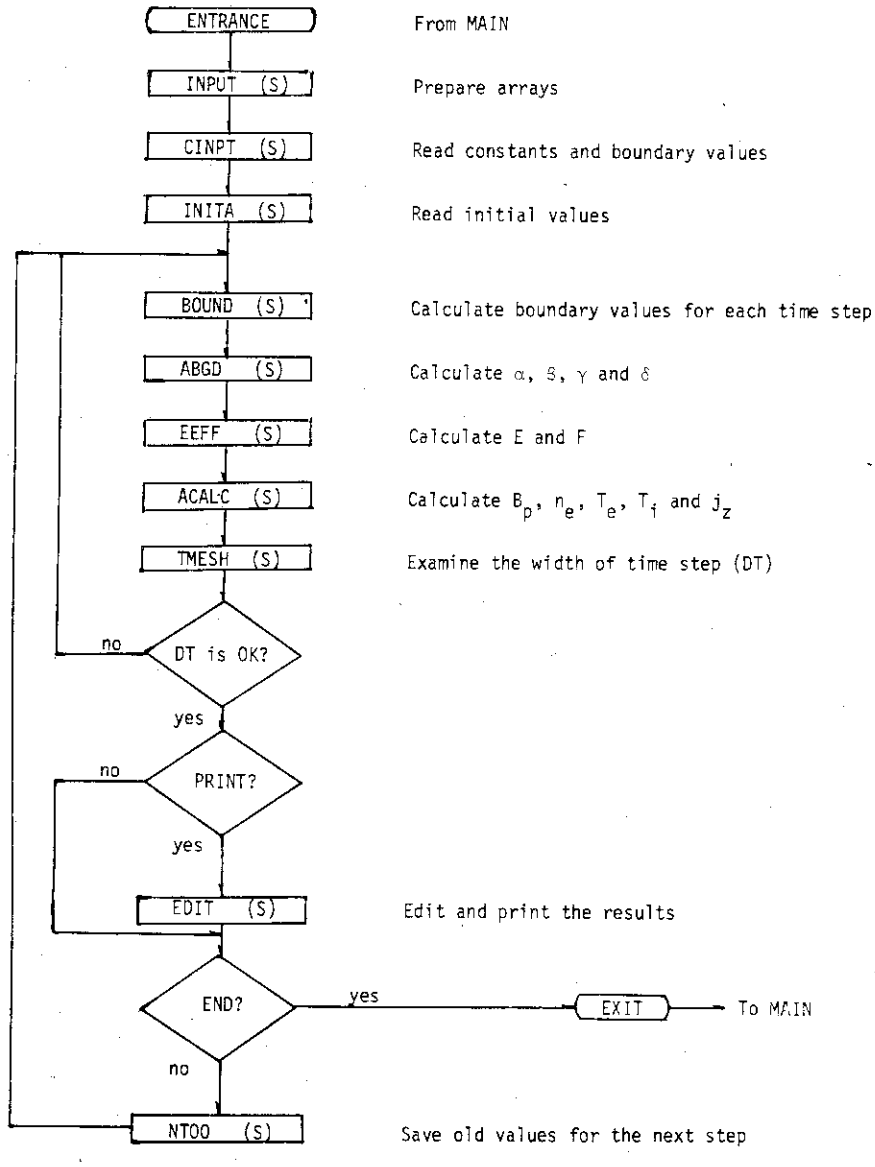


Fig. 13 Flow chart of the code.