CLM-R253

CLM-R253

UKAEA

Report

FETRAN
A FINITE ELEMENT TRANSPORT MODELLING PACKAGE

Lee-Hsiao
J. W. Eastwood

CULHAM LABORATORY
Abingdon, Oxfordshire
1985

C.13

Available from H. M. Stationery Office

Price £4.00

FETRAN
A FINITE ELEMENT TRANSPORT MODELLING PACKAGE

Lee-Hsiao* and J W Eastwood
Culham Laboratory, Abingdon, Oxfordshire, OX14 3DB.
(EURATOM/UKAEA Fusion Association)

Abstract

A new moving finite element formulation for equilibrium and transport modelling has been developed. This report describes the implementation of multifluid surface averaged transport using these moving finite elements. The program can be used for axisymmetric toroidal or cylindrical configurations of arbitrary cross section. Test results using INTOR parameters and geometry are presented.

* Permanent Address: Institute of Plasma Physics, PO No 26, He Fei, An Huei Province, China

July 1985

ISBN: 0-85311-137-5

1. INTRODUCTION

The finite element method [1,2] may be used to derive a consistent family of computer simulation models for plasma from Alfvenic to transport timescales. The algorithms obtained have optimal accuracy and are more flexible than their finite difference approximation counterpart. Figure 1 shows the family of models. The branch we are primarily concerned with in this report is the right hand side one which leads to a surface averaged transport model. Other branches of the family are described elsewhere [3,4,5,6]. The approach we have used is such that multiple magnetic axes and fully two dimensional transport, such as would be encountered in the outer plasma regions and scrape-off layer in devices with poloidal divertors could be handled, although this option has not been implemented in the present version of the code.

For the long timescales associated with transport processes implicit time integration using time dependent basis functions has been found advantageous. Time dependent basis functions are finite element analogues to moving mesh finite difference schemes [7,8,9]. They are used to obtain good resolution where it is required and in transport calculations allow flux surface aligned elements to be maintained.

The method of deriving the discrete equations used in the program FETRAN is summarised in figure 1. The MHD equations are split by using a fractional timestep. Physically this may be viewed as diffusion processes moving the plasma from equilibrium followed by advection to restore force balance, followed by more diffusion and so forth. Details of the equations and their splitting are given in the next section, followed in section 3 by an outline of the finite element discretisation. Sections 4 and 5 describe initial and boundary conditions and transport coefficients, respectively. The remainder of the paper gives a brief description of the code structure, implementation, and how the code is used.

2. THE PHYSICAL MODEL

The MHD model we assume is the charge neutral two-fluid approximation:

$$\frac{d\rho}{dt} + \rho \underset{\sim}{\nabla} \cdot \underset{\sim}{u} = S_m \tag{1}$$

$$\rho \frac{d\underset{\sim}{u}}{dt} + \underset{\sim}{\nabla} p - \underset{\sim}{j} \times \underset{\sim}{B} = 0 \tag{2}$$

$$\frac{nk_B}{\gamma - 1} \frac{dT_e}{dt} + nk_B T_e \underset{\sim}{\nabla} \cdot \underset{\sim}{u} = \underset{\sim}{\nabla} \cdot \underset{\approx}{\kappa}_e \cdot \underset{\sim}{\nabla} T_e + S_e \tag{3}$$

$$\frac{nk_B}{\gamma - 1} \frac{dT_i}{dt} + nk_B T_i \underset{\sim}{\nabla} \cdot \underset{\sim}{u} = \underset{\sim}{\nabla} \cdot \underset{\approx}{\kappa}_i \cdot \underset{\sim}{\nabla} T_i + S_i \tag{4}$$

$$\underset{\sim}{E} + \underset{\sim}{u} \times \underset{\sim}{B} = \underset{\approx}{\eta} \cdot \underset{\sim}{j} \tag{5}$$

where ρ is the mass density, $\underset{\sim}{u}$ is the fluid velocity, $p = nk_B(T_i + T_e)$ is pressure, $\underset{\sim}{j}$ is current density, $\underset{\sim}{B}$ is magnetic field, n is number density and T_i and T_e are respectively ion and electron temperature. The source terms S_i and S_e, conductivity tensors $\underset{\approx}{\kappa}_i$ and $\underset{\approx}{\kappa}_e$ and the resistivity tensor $\underset{\approx}{\eta}$ are assumed to be function of the fields and plasma state. Ampere's and Faraday's Laws close the set of equations:

$$\underset{\sim}{\nabla} \times \underset{\sim}{B} = \mu_0 \underset{\sim}{j} \qquad \underset{\sim}{\nabla} \cdot \underset{\sim}{B} = 0 \tag{6}$$

$$\underset{\sim}{\nabla} \times \underset{\sim}{E} = \frac{-\partial \underset{\sim}{B}}{\partial t} \tag{7}$$

Time splitting separates the advective and diffusive parts of equations (1)-(7). The equations may be formally written as

$$A \frac{\partial U}{\partial t} - L_1(U) = L_2(U) + S \tag{8}$$

where the operator L_1 contains advective terms and L_2 contains diffusive terms. Time splitting is achieved by defining subsidiary vectors V and W, where V describes the diffusion stage

$$A \frac{\partial V}{\partial t} = L_2(V) + S \tag{9}$$

and W describes the advection stage

$$A \frac{\partial W}{\partial t} = L_1(W) \tag{10}$$

Suppose that $U(t)$ is known, then a first order accurate approximation U^* to $U(t + \Delta t)$ may be constructed as follows:

$$V^* = U(t) + \int_t^{t+\Delta t} \dot{V}\, dt \tag{11}$$

$$U^* = V^* + \int_t^{t+\Delta t} \dot{W}\, dt \tag{12}$$

Similarly, a second order approximation could be constructed by symmetrising the splitting in time. Equation (9) is the set of equations whose solution is handled by FETRAN. The solution of the advective (ideal MHD) processes is handled by the program EQUSOL [4].

Expressing the component equations of Eq (9) explicitly gives

$$\frac{\partial n}{\partial t} = S_n \tag{13}$$

$$\frac{nk_B}{\gamma - 1} \frac{\partial T}{\partial t} = \underset{\sim}{\nabla} \cdot \underset{\approx}{K} \cdot \underset{\sim}{\nabla} T + S_T \tag{14}$$

$$\frac{\partial f}{\partial t} = -\mathrm{Re}_{\phi} \underset{\sim}{\nabla} \times \underset{\sim}{E} \tag{15}$$

$$\frac{\partial \psi}{\partial t} = -RE_{\phi} \tag{16}$$

Equation (14) represents the temperature equation for electron and ion species. Faraday's law (Eqs (15) and (16)) is written in terms of the poloidal flux, ϕ, and toroidal flux, f, where

$$\underset{\sim}{B} = \frac{f \underset{\sim}{e}_\phi}{R} + \nabla\phi \times \underset{\sim}{e}_\phi \qquad (17)$$

R is major radius and $\underset{\sim}{e}_\phi$ is the unit vector in the toroidal direction. Equations (13)-(16), with appropriate definitions for S_n and S_T, are the transport equations whose discrete approximations FETRAN solves.

3. FINITE ELEMENT DISCRETISATION

The differential equations, Eqs (13)-(16) are reduced to discrete algebraic equations by the method of weighted residuals. If we let ϕ_p be the node basis function for the assembled elements (ie, the same function that is used in EQUSOL [4]) then, provided that elements are flux surface aligned, we may construct surface basis functions Φ_s by summing over nodes $\{p\}$ belonging to surface s :

$$\Phi_s = \sum_{\{p\} \in s} \phi_p \qquad (18)$$

In the slow timescale limit, equilibriation of density and temperature on the flux surfaces may be approximated as an instantaneous process. This allows Eqs (13)-(16) to be projected onto basis functions $\{\Phi_s\}$ to yield a set of surface averaged discrete transport equations. If surface equilibrium were not assumed, then the choice of surface aligned elements would again be advantageous in that it decouples the disparate perpendicular and parallel transport scalelengths.

Ohmic dissipation causes flux surfaces to move. If the moving flux surface aligned elements were chosen to be tied to given flux surfaces, computational failure would eventually occur as elements collapsed into magnetic nulls. A more suitable choice is to assume that elements move with some velocity $\underset{\sim}{u}$, where

$$\frac{d\Phi_s}{dt} = \frac{\partial}{\partial t}\Phi_s + \underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s = 0 \qquad (19)$$

Consider first the equation (13) for number density. Projecting it onto basis functions Φ_s gives

$$\int \Phi_s \dot{n} d\tau = \int \Phi_s S_n d\tau$$

$$= \frac{d}{dt}\int \Phi_s n d\tau + \int n\underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s d\tau$$

$$\therefore \quad \frac{d}{dt}\int \Phi_s n d\tau = -\int n\underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s d\tau + \int \Phi_s S_n d\tau \qquad (20)$$

Similarly

$$\frac{d}{dt}\int \Phi_s \frac{nk_B T}{\gamma - 1} d\tau = -\int\left(\frac{p\underset{\sim}{u}}{\gamma - 1} + \underset{\sim}{\kappa}\cdot\underset{\sim}{\nabla}T\right)\cdot\underset{\sim}{\nabla}\Phi_s d\tau + \int \Phi_s S_r' d\tau \qquad (21)$$

$$\frac{d}{dt}\int \Phi_s f \, d\tau = -\int f\underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s d\tau - \int\left(\underset{\sim}{E} \times \underset{\sim}{\nabla}\Phi_s R^2\right)\cdot\underset{\sim}{e}_\phi \frac{d\tau}{R} \qquad (22)$$

$$\frac{d}{dt}\int \Phi_s \psi \, d\tau = -\int \psi\underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s d\tau - \int \Phi_s R E_\phi d\tau \qquad (23)$$

For surface averaged transport, p, n, T, f are functions of ψ only. Making the particular choices of 'incompressible' basis functions (ie, the surface average of $\underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s$, $\langle \underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s \rangle$, is zero) leads to all terms in Eqs (20)-(23) containing $\underset{\sim}{u}$ being zero. Another consequence of $\langle \underset{\sim}{u}\cdot\underset{\sim}{\nabla}\Phi_s \rangle \equiv 0$ is that

$$\frac{d}{dt}\int \Phi_s \Phi_t d\tau = 0 \qquad (24)$$

So by selecting trial functions $p = \sum_r p_r \Phi_r$, $p_r = n_r T_r$, $f = \sum_r f_r \Phi_r$, etc, equations (20)-(23) reduce to

$$M_{sr}\Delta t \frac{dn_r}{dt} = \int \Phi_s S_n d\tau \qquad (25)$$

$$M_{sr}\Delta t \frac{d}{dt}\frac{n_r k_B T_r}{\gamma - 1} = -\int \kappa_\perp \underset{\sim}{\nabla} T \cdot \underset{\sim}{\nabla}\Phi_s \, d\tau + \int \Phi_s S_T' d\tau \tag{26}$$

$$M_{sr}\Delta t \frac{df_r}{dt} = -\int \left(\underset{\sim}{E} \times \underset{\sim}{\nabla}\Phi_s R^2\right) \cdot \underset{\sim}{e}_\phi \frac{d\tau}{R} \tag{27}$$

$$M_{sr}\Delta t \frac{d\psi_r}{dt} = -\int \Phi_s R E_\phi \, d\tau \tag{28}$$

where

$$M_{sr} = \int \frac{\Phi_s \Phi_r}{\Delta t} \, d\tau \tag{29}$$

$$\underset{\sim}{E} = \underset{\sim}{g} \cdot \left(\frac{1}{\mu_0 R} \underset{\sim}{\nabla} f \times \underset{\sim}{e}_\phi + j_\phi \underset{\sim}{e}_\phi\right) \tag{30}$$

The set of evolutionary ordinary differential equations, Eqs (23)-(28) are discretised in time using a fully implicit approximation, yielding a coupled set of tridiagonal matrix equations to be solved at each timestep. The fully implicit approximation replaces the equation of the form

$$\frac{dg}{dt} = h \tag{31}$$

by

$$\frac{g(t) - g(t - \Delta t)}{\Delta t} = h(t) \tag{32}$$

4. INITIAL AND BOUNDARY CONDITIONS

The program FETRAN may either be used independently, or in conjunction with the equilibrium solver package EQUSOL. In the former case, force balance is ignored and initial conditions, not necessarily satisfying $\underset{\sim}{\nabla} p = \underset{\sim}{j} \times \underset{\sim}{B}$, must be provided. If the equilibrium package is coupled to the transport program, then output from one provides input for the other, and vice-versa, thus giving a consistent surface averaged transport model.

For the test calculations described below, parabolic temperature and density profiles of the form

$$F(r) = F(o) + \big(F(o) - F(a)\big)\big(1 - (r/a)^2\big) \tag{33}$$

are assumed for initial conditions, where $F(o)$ and $F(a)$ are respectively values on axis and at the wall. The toroidal flux function f is initially set to $R_0 B_{\phi 0}$, where R_0 is the major radius of the axiysmmetric toroidal system and $B_{\phi 0}$ is the vacuum toroidal magnetic field.

A prescription for the poloidal flux, ψ, completes the specification of the initial conditions. This is most conveniently done by setting the safety factor q to, say, a parabolic form, then computing ψ from known values of f and q. The parameters in the specification of q are chosen to give the desired q_{axis} $(\gtrsim 1)$ and total plasma current. The relationship between f, q and ψ, when projected onto basis functions $\{\phi\}$ is

$$\int \frac{\phi f}{R^2}\, dv = 2\pi \int \phi q\, d\psi \tag{34}$$

where integrals are taken over the whole of the computational region. Taking ϕ to be linear basis functions gives the discrete equations which allow nodal values of ψ to be solved given f and q.

The finite element formalism has homogeneous Neumann boundary conditions as natural boundary conditions. The program also allows Dirichlet boundary conditions to be specified for n, T, ψ and f. Constant total plasma current conditions can be used on poloidal flux

$$\int \underset{\sim}{B} \cdot d\underset{\sim}{\ell} = \mu_0 I_p = \text{constant} \tag{35}$$

This leads to the gradient condition relating poloidal flux function values at the two outermost surfaces

$$\psi_s - \psi_{s-1} = \text{constant} \tag{36}$$

The constant on the rhs of Eq (36) is proportional to I_p, with the constant of proportionality being determined by the flux surface geometry.

Initial conditions are set in subroutine <1.6> INITAL (the numbering system follows the OLYMPUS convention [15]), and boundary conditions are applied by subroutine <2.50> XPTBC.

5. TRANSPORT COEFFICIENTS

The code FETRAN provides a device and geometry independent vehicle into which different transport models can be inserted, depending on the circumstances to be modelled. The subroutine <2.52> XPCOEF is presented with the current, field, and plasma parameters as input and returns transport coefficients.

At present, XPCOEF calls the classical transport package COEFFS [10]. COEFFS evaluates the Braginskii coefficients [11] and various plasma parameters. In addition, XPCOEF computes the INTOR transport model used for comparison with the 1-D transport model, HERMES. This takes perpendicular electron thermal conductivity, κ_e, a constant, and perpendicular ion thermal conductivity, κ_i, neoclassical:

$$\kappa_e = 5 \times 10^{19} \text{ (m sec)}^{-1} \tag{37}$$

$$\kappa_i = \frac{0.68}{1 + 0.36\nu_i^*} Z_{eff} \frac{n\rho_{i\theta}^2}{\tau_i} \varepsilon^{1/2} + Z_{eff} \frac{\rho_i^2 n}{\tau_i} \left(1 + 1.6q^2\right) \tag{38}$$

and Spitzer resistivity without a trapping correction. In Eq (38), $\varepsilon = r/R$, q is the safety factor and the remaining parameters follow the definitions given in [11]. For further details on Tokamak transport see references [11,12,13].

6. ELEMENT MATRICES

One advantage of the finite element method is that it allows the processes of discretisation, mesh addressing and matrix solving to be separated. Discretisation is independent from the method used to connect elements together. Equations relating nodal values are obtained by constructing matrices of coefficients for a single element, then mapping these 'element' matrices onto the global matrix for solution.

In this section, we shall explicitly give the element matrices arising from Eqs (25)-(32) in order to show additional assumptions we make. The most significant assumption is the lumping of the toroidal current in the flux equations. This approximation is used so that fully implicit equations for ψ and f are obtained without increasing the bandwidth of the resulting global matrix.

We consider first the toroidal flux equation (27). The basis functions $\{\Phi_s\}$ are generated by summing over all the basis functions $\{\phi_p;\ p\epsilon s\}$ of the triangular elements having nodes on flux surface s. In a similar manner, projections of equations onto the surface basis function, Φ_s, can be achieved by projecting onto single elements, then summing contributions of the subset of elements with nodes on surface s. Assume that a triangular element has nodes labelled anticlockwise from 1 to 3, where node 1 lies on surface s and nodes 2 and 3 lie on surface $t\ (= s \pm 1)$. Each term in Eq (27) will lead to four element matrix contributions to the global matrices. The mass matrix contributions are given by element integrals over the linear basis functions, ϕ_i,

$$\int \frac{\phi_i \phi_j}{\Delta t}\, dv \simeq \frac{V}{12\Delta t}\left(a + d_{ij}\right) \qquad (39)$$

where V is the element volume. These contribute to the surface average mass matrix (Eq (29)) elements (s,s), (s,t), (t,s), (t,t). Summing contributions from Eq (1) ((i,j) = (1,1) for (s,s), (i,j) = (1,2), (1,3) for (s,t), (i,j) = (2,1), (3,1) for (t,s) and (i,j) = (2,2), (2,3), (3,2), (3,3) for (t,t)) gives

$$\delta m = \begin{Bmatrix} \delta m_{ss} & \delta m_{st} \\ \delta m_{ts} & \delta m_{tt} \end{Bmatrix} = \begin{Bmatrix} 1 & 1 \\ 1 & 3 \end{Bmatrix} \frac{V}{6\Delta t} \tag{40}$$

Accumulation of the mass matrix is implemented in subroutine <2.41> MASMAT.

The right hand side of Eq (27) may be written as

$$- \int \underset{\sim}{\nabla}(\Phi_s R^2) \cdot \bar{\underset{\sim}{\eta}}_{pp} \cdot \frac{\underset{\sim}{\nabla} f}{\mu_0 R^2} \, d\tau - \int \underset{\sim}{\nabla}(\Phi_s R^2) \cdot \bar{\underset{\sim}{\eta}}_{p\phi} j_\phi \frac{d\tau}{R} \equiv -A^{11}_{st} f_t - A^{12}_{st} \psi_t \tag{41}$$

where

$$\bar{\underset{\sim}{\eta}}_{pp} = \eta_\perp \underset{\approx}{1} + (\eta_\parallel - \eta_\perp) \frac{\underset{\sim}{\nabla}\psi \underset{\sim}{\nabla}\psi}{f^2 + \left|\underset{\sim}{\nabla}\psi\right|^2} \tag{42}$$

$$\bar{\underset{\sim}{\eta}}_{p\phi} = (\eta_\parallel - \eta_\perp) \frac{f \underset{\sim}{\nabla}\psi}{f^2 + \left|\underset{\sim}{\nabla}\psi\right|^2} \tag{43}$$

$$\mu_0 j_\phi = - R \underset{\sim}{\nabla} \cdot \frac{1}{R} \underset{\sim}{\nabla}\psi \tag{44}$$

Evaluation of matrix element contributions to A^{11}_{st} is straightforward. For a single triangular element, the integral becomes

$$\int \underset{\sim}{\nabla}\phi_i \cdot \frac{\bar{\underset{\sim}{\eta}}_{pp}}{\mu_0} \cdot \underset{\sim}{\nabla}\phi_j d\tau + 2 \int \phi_i \, \underset{\sim}{\bar{R}} \cdot \bar{\underset{\sim}{\eta}}_{pp} \cdot \frac{\underset{\sim}{\nabla}\phi_j}{\mu_0} \frac{d\tau}{R} \tag{45}$$

Contributions are summed as described above for the mass matrix. We assume element dimensions are much smaller than R in evaluating the integrals to obtain

$$\delta A^{11} = \begin{Bmatrix} 1 & -1 \\ -1 & 1 \end{Bmatrix} \underset{\sim}{\lambda}_1 \cdot \langle \bar{\underset{\sim}{\eta}}_{pp} \rangle \cdot \underset{\sim}{\lambda}_1 \frac{V}{\mu_0} + \begin{Bmatrix} 1 & -1 \\ 2 & -2 \end{Bmatrix} \frac{\underset{\sim}{\bar{R}}}{3\langle R \rangle} \cdot \langle \bar{\underset{\sim}{\eta}}_{pp} \rangle \cdot \underset{\sim}{\lambda}_1 \frac{V}{\mu_0} \tag{46}$$

where $\langle\rangle$ denote element average and the vector $\underset{\sim}{\lambda}_1$ is given by

$$\underset{\sim}{\lambda}_1 = (\underset{\sim}{x}_3 - \underset{\sim}{x}_2) \times \underset{\sim}{e}_\phi \frac{\langle R \rangle}{2V} \tag{47}$$

and $\underset{\sim}{x}_i = (R_i, Z_i)$ is the position of element node i .

Lumping of the toroidal current is employed to force the matrix A^{12} into the same sparsity pattern at A^{11} . It involves approximating the left hand side of

$$\int \mu_0 \phi_i j_\phi \frac{d\tau}{R} = \int \underset{\sim}{\nabla}\phi_i \cdot \underset{\sim}{\nabla}\psi \frac{d\tau}{R^2} \tag{48}$$

by $$\left(\int \mu_0 \phi_i \frac{d\tau}{R}\right) j_i \tag{49}$$

where j_i is the value of toroidal current density, and similarly approximating the integral

$$\int \underset{\sim}{\nabla} (\phi_i R^2) \cdot \underset{\sim}{\bar{\eta}}_{p\phi} j_\phi \frac{d\tau}{R} \simeq \left[\int \underset{\sim}{\nabla} (\phi_i R^2) \cdot \underset{\sim}{\bar{\eta}}_{p\phi} \frac{d\tau}{R}\right] j_i \tag{50}$$

Combining these two approximations gives the contribution of node pair (i,j) to A^{12} as

$$\frac{\beta_i}{\alpha} \int \underset{\sim}{\nabla}\phi_i \cdot \underset{\sim}{\nabla}\phi_j \frac{d\tau}{R^2} \tag{51}$$

where

$$\beta_i = \int \underset{\sim}{\nabla} (\phi_i R^2) \cdot \underset{\sim}{\bar{\eta}}_{p\phi} \frac{d\tau}{R} \tag{52}$$

$$\alpha = \int \mu_0 \frac{\phi_i R^2}{R} = \frac{\mu_0 V}{3\langle R \rangle} \tag{53}$$

Thus

$$\delta A^{12} = \begin{Bmatrix} \beta_1 \underset{\sim}{\lambda}_1 & -\beta_1 \underset{\sim}{\lambda}_1 \\ \beta_2 \underset{\sim}{\lambda}_2 + \beta_3 \underset{\sim}{\lambda}_3 & -\beta_2 \underset{\sim}{\lambda}_2 + \beta_3 \underset{\sim}{\lambda}_3 \end{Bmatrix} \cdot \frac{3 \underset{\sim}{\lambda}_1}{\mu_0 \langle R \rangle} \tag{54}$$

and $\beta_i \simeq \left(\langle R\rangle \underset{\sim}{\lambda}_i + \frac{2\bar{\bar{R}}}{3}\right) \cdot \langle \bar{\bar{\eta}}_{p\phi} \rangle \; V$ (55)

Cyclically permuting indices in Eq (47) gives the definition of vectors $\underset{\sim}{\lambda}_1$. Elements of δA^{11} and δA^{12} are computed in sections 2.1 and 2.2 of subroutine <2.46> MATASS, respectively, whilst factors β_i are found in section 3 of <2.44> JTFACS.

Equation (28) yields the equation for advancing poloidal flux values. Substituting for E_ϕ gives

$$M_{sr}\Delta t \frac{d\psi_r}{dt} = - \int \Phi_s \, \bar{\bar{\eta}}_{p\phi} \cdot \frac{\nabla f}{\mu_0} \, d\tau - \int \Phi_s R \eta_{\phi\phi} j_\phi \, d\tau$$
$$= - A^{21}_{st} f_t - A^{22}_{st} \psi_t \qquad (56)$$

Element contributions to A^{21} and A^{22} are found in exactly the same manner as shown above for A^{11} and A^{12} . Again, lumping is used for j_ϕ and R is replaced by its element average $\langle R \rangle$ to give:

$$\delta A^{21} = \begin{Bmatrix} 1 & -1 \\ 2 & -2 \end{Bmatrix} \frac{\langle \bar{\bar{\eta}}_{p\phi} \rangle \cdot \underset{\sim}{\lambda}_1 V}{3\mu_0} \qquad (57)$$

$$\delta A^{22} = \begin{Bmatrix} \gamma_1 \underset{\sim}{\lambda}_1 & -\eta_1 \underset{\sim}{\lambda}_1 \\ \gamma_2 \underset{\sim}{\lambda}_2 + \gamma_3 \underset{\sim}{\lambda}_3 & -\gamma_2 \underset{\sim}{\lambda}_2 - \gamma_3 \underset{\sim}{\lambda}_3 \end{Bmatrix} \cdot \frac{3\underset{\sim}{\lambda}_1}{\mu_0 \langle R \rangle}$$
$$= \begin{Bmatrix} 1 & -1 \\ -1 & 1 \end{Bmatrix} \frac{\langle \eta_{\phi\phi} \rangle V \left| \lambda_1 \right|^2}{\mu_0} \qquad (58)$$

where

$$\gamma_i = \langle R \rangle \langle \eta_{\phi\phi} \rangle \frac{V}{3}$$

δA^{21} and δA^{22} are computed, respectively in Sections 3.1 and 3.2 of <2.46> MATASS.

If we include ohmic heating, heat exchange and Bremsstrahlung terms, the electron temperature equation becomes

$$M_{sr}\Delta t \frac{d}{dt}\frac{(nk_B T_e)_r}{\gamma - 1} = -\left(\int \kappa_{\perp e} \underset{\sim}{\nabla} T_e \cdot \nabla \Phi_s \, d\tau + \int \Phi_s C_{ex} T_e \, d\tau\right)$$

$$+ \int \Phi_s C_{ex} T_i \, d\tau + \int \Phi_s \left(\underset{\sim}{j} \cdot \underset{\sim}{\eta} \cdot \underset{\sim}{j} + C_R Z n^2 T_e^{1/2}\right) d\tau$$

$$= - A^{33}_{sr} T_{er} - A^{34}_{sr} T_{ir} + S^3_r \qquad (59)$$

giving

$$\delta A^{33} = \begin{Bmatrix} 1 & -1 \\ -1 & 1 \end{Bmatrix} \langle \kappa_e \rangle \left| \underset{\sim}{\lambda}_1 \right|^2 V - \delta A^{34} \qquad (60)$$

and

$$\delta A^{34} = - \begin{Bmatrix} 3 & 2 \\ 2 & 3 \end{Bmatrix} \frac{C_{exs}}{30} V - \begin{Bmatrix} 2 & 3 \\ 3 & 12 \end{Bmatrix} \frac{C_{ext}}{30} V \qquad (61)$$

C_{ex} and C_r are exchange energy and radiation energy coefficients, respectively C_{exs} and C_{ext} are values of the exchange energy coefficients on surfaces s and t .

The last two terms in Eq (59) are treated explicitly. The ohmic term is

$$\int \phi_i \underset{\sim}{j} \cdot \underset{\sim}{\eta} \cdot \underset{\sim}{j} d\tau = \int \phi_i \underset{\sim}{\nabla} f \cdot \bar{\underset{\sim}{\eta}}_{pp} \cdot \underset{\sim}{\nabla} f \frac{d\tau}{(\mu_0 R)^2} + 2\int \phi_i \cdot \frac{\underset{\sim}{\nabla} f}{\mu_0 R} \cdot \bar{\underset{\sim}{\eta}}_{p\phi} j_\phi \, d\tau$$

$$+ \int \phi_i \eta_{\phi\phi} j_\phi^2 d\tau \qquad (62)$$

$$\simeq \underset{\sim}{\lambda}_1 \cdot \langle \bar{\underset{\sim}{\eta}}_{pp} \rangle \cdot \underset{\sim}{\lambda}_1 \left(\frac{\Delta f}{\mu_0 \langle R \rangle}\right)^2 + 2\underset{\sim}{\lambda}_1 \cdot \langle \bar{\underset{\sim}{\eta}}_{p\phi} \rangle \frac{V}{\mu_0 \langle R \rangle} \Delta f \, j_i + \langle \eta_{\phi\phi} \rangle \frac{V j_i^2}{3}$$

if lumping is used for j_ϕ .

The electron temperature equation is coded in Section 4 of <2.46> MATASS. Section 5 of that subroutine contains the corresponding terms for the ion equation.

7. MATRIX ASSEMBLY AND SOLUTION

The matrix assembly scheme used exploits the fact that for a single magnetic axis configuration, the surface averaged transport equations reduce to a set of block tridiagonal equations. The element matrix computations described in the previous section, and implemented in subroutine <2.46> of the program, would be unchanged for multiple axis configurations - the only changes needed would be in the routines <2.48> ADDMEL and <2.49> ADDVEL that <2.46> calls to add the element matrix and vector contributions to the global matrix and vector, respectively.

Each triangular finite element has vertices (1,2,3) which correspond to surfaces (s,t,t) where $t = s \pm 1$, and contributes to block matrix elements (s,s) , (s,t) , (t,s) , (t,t) . The block matrix elements are of order MEQ × MEQ with entries for each variable $(f,\psi,T_e,T_i,\ldots)$ in each equation. The base address for each block element is given by

$$I(s_i,s_j) = MEQ^2(2s_i + s_j - 3) \tag{63}$$

where s_i and s_j take values s or t . Contributions from equation NEQ to coefficients for variable NVAR are stored in a linear array at location

$$I(s,t) + (NVAR - 1)MEQ + NEQ \tag{64}$$

Thus, element matrices δA^{ij} described in the previous section are added into global matrix B at locations

$$\begin{Bmatrix} I(s,s) & I(s,t) \\ I(t,s) & I(t,t) \end{Bmatrix} + \begin{Bmatrix} 1 & 1 \\ 1 & 1 \end{Bmatrix} (i + (j - 1)\ MEQ) \tag{65}$$

This addition is performed by <2.48> ADDMEL.

Source terms are similarly stored in a linear array. The value of variable NVAR on surface s is stored in vector element

$$(NVAR - 1)\ NSMAX + s \qquad (66)$$

where NSMAX is some integer greater than or equal to the number of flux surfaces. The solution vector u has elements stored in the same order.

Once matrix assembly is completed, then elements in the global matrix can be adjusted to take into account imposed boundary conditions (cf subroutine <2.50> XPTBC) giving the equation

$$Bu = d \qquad (67)$$

to be solved. In FETRAN, the solution vector u is found using the block tridiagonal matrix solver BLKSLV [14]. This approach is somewhat inefficient as it does not exploit the sparsity of the block submatrices, but is sufficient to demonstrate the viability of the formulation.

8. PROGRAM STRUCTURE

The subroutines of FETRAN have been written following the OLYMPUS [15] conventions for notation, layout and documentation. The principal subroutines have the standard OLYMPUS functions. Table 1 gives a flow diagram of the program. Subroutine <1.9> SETMSH1 and the routines it calls belong to the triangular finite element mesh initialisation package ELSET [15]. <E1> COEFFS and <A7>-<A10> are respectively classical transport coefficient [13] and block tridiagonal solver packages [14]. The block data module (used by COEFFS) loads fundamental constants in SI units into the common block COMFUN. Values of constants used are those given in [16].

Subroutine <2.40> XPORT controls the solution of the transport equations. The physics is contained in subroutines <2.52>, <2.44>, <2.46> and <2.50>. <2.52> XPCOEFF computes transport coefficient, <2.44> JTFACS computes terms associated with the lumped j_ϕ approximation. <2.46> MATASS computes and assembles element matrix contributions as described in section 6, and <2.50> XPTBC applies the boundary conditions. The remaining Class 2 subroutines are concerned with moving data to and from the block solver and evaluating errors. An index of subprograms is given in Table 2.

The main transport controlling routine, XPORT contains an adaptive timestep scheme. At each step the max norm

$$\varepsilon = \max \left| \frac{u^{n+1} - u^{n}}{u^{n+1} + u^{n}} \right| \qquad (68)$$

over all members of the solution vector is computed. The timestep is adjusted according to the prescription

```
IF    ε ≤ ε0
THEN  DT := DT*DTFAC
ELSE  return to old timelevel,
      DT := DT/DTFAC
FI
```

Here DTFAC is a factor larger than unity. Usually DTFAC = 1.2 and $\varepsilon_0 = 5\%$ prove satisfactory. In addition, it is possible to iterate the solution at each timestep, thus allowing the transport coefficients and source terms to be computed implicitly. Usually such iteration is not necessary, although parameter NXPTIT is provided to allow iteration to be performed. In runs tried so far, setting NXPTIT = 4 guarantees an error less than 10^{-3}.

9. TEST RUN

Test runs have been performed using INTOR transport parameters (cf section 5) and geometry. In the large aspect ratio approximation and when force balance is ignored, the code gives results in good agreement with those obtained using the one dimensional code HERMES [17] running under the same conditions. For 1 1/2-D transport, integration of the transport equations is interleaved with equilibrium solutions to restore force balance.

Table 3 shows input data for a sample run using INTOR transport, but without the equilibrium being recomputed. Figure 2 shows the initial 2-D element triangulation generated by the ELSET routines [5] for NCASE = 2, NSECT = 3 and NSURF = 11. (Figure 3 shows the element displacements as a result of solving for pressure balance - for further details on solving force balance see [4].) Figure 4 shows (a) initial electron temperature, T_e, and poloidal flux, ψ, and (b) T_e and ψ after 6.8 seconds. Figure 5 shows the time evolution of the axial electron temperature, and figure 6 shows the final electron temperature profile with and without equilibrium computation. For the case with equilibrium solution, EQUSOL was called every 10 transport steps. A sample of output for the test case input of Table 3 is given in Table 4.

10. FINAL REMARKS

The work reported here successfully demonstrates the finite element treatment of transport in plasmas. The finite element approach allows the physics, the addressing and storage, and the matrix solutions to be separated. This makes the part of the program that a user needs to be aware of smaller. The result is that programs become easier to adapt. New applications should require relatively little development work.

The number of equations in the transport model can be easily extended. For example, to add a density equation to the set (for $f,\psi,T_e T_i$) solved in FETRAN required (i) MEQ to be increased to 5 to deal with addressing and matrix solution, (ii) computation of element matrices

to be added to <2.46> MATASS and copying of results from and to the density variable ELDEN added to <2.42> OLDVEC and <2.51> NEWVEC. Adding new terms to existing equations is equally straightforward, involving only integrations over a single triangular element to find the form of the element matrices to be coded into <2.46> MATASS. Questions of complicated curvilinear coordinates systems and metrics do not arise, as geometrical features are taken care of by element assembly.

Any plasma cross section shape can be dealt with simply by appropriately connecting triangular elements. Examples of this may be seen in [4]. Multiple axis systems do not change discretisation over triangular elements, so to adapt the model to multiple axis systems only requires the matrix solver BLKSLV to be replaced by a more general one, and addressing used by <2.48> ADDMEL to be altered to match the replacement matrix solver.

The implementation of the finite element described here does not compare favourably with existing 1 1/2-D finite difference models in terms of computational speed. This is because we aimed to demonstrate first that the finite element method would work by using the simplest code structure and existing modules not tailored to the present application. The inferior computational speed of FETRAN arises because (i) the known sparsity of block submatrices fed to BLKSLV is not exploited, (ii) many spurious calculations are performed by calling COEFFS at each node on each flux surface, and (iii) geometric factors, which change only when the equilibrium solver is called, are recomputed at every iteration of each transport timestep. For example, the matrix term

$$\int \kappa \underset{\sim}{\nabla}\Phi_s \cdot \underset{\sim}{\nabla}\Phi_t$$

in Eq (59) is computed by summing the contributions of all triangular elements with nodes on surfaces s and t (= s ± 1) :

$$\delta \int \kappa \underset{\sim}{\nabla}\Phi_s \cdot \underset{\sim}{\nabla}\Phi_t = \begin{Bmatrix} 1 & -1 \\ -1 & 1 \end{Bmatrix} \sum_{i \varepsilon E_{st}} \left(\left| \underset{\sim}{\lambda}_1 \right|^2 V_i \right) \langle \kappa \rangle_i \qquad (69)$$

In Eq (69), the sum i is over the set E_{st} of triangular element lying between surface s and t. E_{st} comprises the subsets E^+ and E^- of elements with two nodes on surfaces s and t, respectively. Thus we may rewrite Eq (69) as

$$\delta\int\kappa\underset{\sim}{\nabla}\Phi_s\cdot\underset{\sim}{\nabla}\Phi_t = \begin{Bmatrix} 1 & -1 \\ -1 & 1 \end{Bmatrix} (\sigma_s\kappa_s + \sigma_t\kappa_t) \tag{70}$$

where

$$\sigma_s = \left(\frac{2}{3}\sum_{i\in E^-} + \frac{1}{3}\sum_{i\in E^+}\right)\left(\left|\underset{\sim}{\lambda}_1\right|^2 V\right)_i \tag{71}$$

$$\sigma_t = \left(\frac{1}{3}\sum_{i\in E^-} + \frac{2}{3}\sum_{i\in E^+}\right)\left(\left|\underset{\sim}{\lambda}_1\right|^2 V\right)_i \tag{72}$$

and κ_s, κ_t, are values of κ on surfaces s and t, respectively. If geometrical surface factors σ_s and σ_t are computed only when the geometry changes, then the computations of matrix elements is reduced from a two-dimensional (Eq (69)) to a one dimensional (Eq (70)) summation, with obvious speed gains. Similar arguments hold for other terms. The cost benefits of these reordering of computations dictate that the use of macroelements (unions of triangular elements on surfaces) should be implemented to separate geometrical and surface averaged quantities.

ACKNOWLEDGEMENTS

We thank Dr M H Hughes for providing HERMES test data and advice on the INTOR parameters and runs.

REFERENCES

[1] G Strang and G J Fix, "An Analysis of the Finite Element Method", Prentice-hall, NJ, (1973)

[2] T J Chung, "Finite Element Analysis in Fluid Dynamics", McGraw-Hill, NY, (1978)

[3] J W Eastwood, "Two Dimension Euler-Lagrange Finite Element Plasma Simulation", CPN/81/1 (1981)

[4] J W Eastwood and Lee-Hsiao, "EQUSOL - a finite element package for MHD equilibrium in arbitrary geometry", TPN in preparation (1984)

[5] J W Eastwood, "ELSET- Triangular mesh finite element initialisation", CPN(80)7 (1980)

[6] J W Eastwood, "The finite element solution to the hyperbolic problem", in preparation (1984)

[7] J U Brackbill, Meth Comput Phys 16, 1-41 (1976)

[8] M J Fritts and J P Boris, J Computational Phys 31, 173-215 (1979)

[9] D V Anderson, J Computational Phys 17, 246-275 (1975)

[10] J P Christiansen, M L Watkins and K V Roberts, "Plasma Parameters and MHD Transport Coefficients", CPN 5/76 (1976)

[11] S I Braginskii, "Transport Processes in a Plasma", Rev: Plasma Phys 1, 205-311 (1965)

[12] J T Hogan, "Multifluid Tokamak Transport Models", Meth Comput Phys 16, 131-165 (1976)

[13] D F Düchs, D E Post, P H Rutherford, Nuclear Fusion 17, 3 (1977)

[14] M H Hughes and P M Keeping, "BLKSV - a block tridiagonal solver", private communication (1983)

[15] K V Roberts, Computer Phys Commun 7, 237 (1974)

[16] B N Taylor et al, Rev Mod Phys 41, 375-496 (1969)

[17] M H Hughes, "HERMES - A 1-D Tokamak Transport Program", private communication (1983)

Table 1: Flow diagram of Program FETRAN

```
BLOCK DATA
MASTER 0.0
BASIC  0.1
MODIFY 0.2
COTROL 0.3 ---LABRUN 1.1
           ---CLEAR  1.2
           ---PRESET 1.3
           ---DATA   1.4
           ---AUXVAL 1.5 ---SETMSH 1.9  ---SETCS1 1.11 ---TESVOL 1.10
                                                       ---BCTOBV 1.19

           ---INITAL 1.6T
           ---START  1.8
           ---STEPON 2.1T ---XPORT 2.40X---MASMAT 2.41
                                        ---OLDVEC 2.42  ---ADDVEC 2.43
                                        ---XPCOEF 2.52X ---COEFFS E1
                                        ---JTFACS 2.44

                                        ---MASSTM 2.45
                                        ---MATASS 2.46  ---MBASAD 2.47
                                                        ---ADDMEL 2.48
                                                        ---ADDVEL 2.49

                                         ---XPTBC 2.50  ---VFIXBC 2.54

                                         ---BLKSLV A7   ---LUF A8
                                                        ---FANDB A9
                                                        ---FANDBV A10

                                        ---NEWVEC 2.51 ---XPCOPY 2.53

                                        ---WRITE  3.2
                                        ---GRAPH  3.3
           ---OUTPUT 3.1
           ---TESEND 4.1
           ---ENDRUN 4.2
```

Table 2: Index of subprograms

Table 2 : Index of subprograms

	PROLOGUE CLASS 1	
SETMSH	*ELSET*CONTROL MESH INITIALISATION	1.9
SETCS1	*ELSET*HEXAGONAL MESH IN HALF-CIRCLE	1.11
TESVOL	*ELSET*CHECK FOR POSITIVE VOLUMES	1.10
BCTOBV	*ELSET*BOUNDARY CELL TO VERTEX POINTERS	1.19
	CALCULATION CLASS 2	
XPORT	ADVANCE TRANSPORT EQUATIONS BY ONE STEP	2.40
MASMAT	COMPUTE TIME DERIV 'MASS' MATRIX	2.41
OLDVEC	FIND OLD TIMELEVEL PART OF RHS	2.42
ADDVEC	ADD CONTRIBUTION TO VECTOR FOR BLOCKSOLVER	2.43
JTFACS	FIND NODE QUANTITIES FOR J-TOROIDAL CALC	2.44
MASSTX	LOAD MASS MATRIX TERMS INTO BLOCK MATRIX	2.45
MATASS	ASSEMBLE BLOCK MATRIX BY MESH F.E.	2.46
MBASAD	COMPUTE BLOCK MATRIX BASE ADDRESSES	2.47
ADDMEL	LOAD F.E. CONTRIBUTION INTO BLOCK MATRIX	2.48
ADDVEC	LOAD F.E.CONTRIBUTION INTO RHS VECTOR	2.49
XPTBC	APPLY B.C.TO BLOCK MATRIX AND RHS VECTOR	2.50
NEWVEC	UPDATE STATE AND TEST FOR CONVERGENCE	2.51
XPCOEF	SUPPLY TRANSPORT COEFFICIENT	2.52
XPCOPY	COPY RESULTS FROM U TO PV,EVALUTE ERROR	2.53
VFIXBC	DIRICHLET B.C.	2.54
	OUTPUT CLASS 3	
WRITE	LINEPRINTER OUTPUT	3.2
GRAPH	GRAPHICAL OUTPUT	3.3
	UTILITIES CLASS E.A	
BDATA	*COEF1*BLOCK DATA	E.1
COEFFS	*COEF1*EVALUTE BRAGINSKII TRANSPORT COEFFS	E.1
BLKSLV	*BLKSLV*SOLVE BLOCK TRIDIAGONAL EQUATIONS	A.7
LUF	*BLKSLV*FORM LU FACTORS	A.8
FANDB	*BLKSLV*FORWARD AND BACKWARD SUBSTITUTION	A.9
FANDBV	*BLKSLV*VECTOR FORWARD AND BACKWARD SUBST	A.10

Table 3: Test input data

```
FETRAN  INTOR Transport Run
Hexagonal mesh in upper half plane
No equilibrium solution case
H.LEE  MAY 18 1984
 &NEWRUN
  RMAJOR=4.5D0,
  RMINOR=1.5D0,
  BTOR0=5.0D0,
  TORCUR=4.0D6,
  DEN0=2.0D20,
  TELC0=200.0D0,
  DT=0.1D-3,
  DTFAC=1.2D0,
  EPSLO5=0.5D-1,
  NTIMES=60,
  NSTART=2,
  NSPRIN=10,
  NRUN = 60,
  NCASE = 2,
  NSECT = 3,
  NSURF = 11,
  NLNOTT = .FALSE.,
  NSMAX = 11,
  NXPTIT = 4,
  NLXPRT = .FALSE.,
  MEQ = 4,
  MAXEQN = 9,
  MAXU= 468,
&END
```

```
RADIATION LOSS
   1 0.1122D 05   2 0.1119D 05   3 0.1013D 05   4 0.8742D 04   5 0.7185D 04   6 0.5556D 04   7 0.3955D 04   8 0.2498D 04
   9 0.1456D 04  10 0.5355D 03  11 0.3023D 02
SAFETY FACTOR
   1 0.9937D 00   2 0.9937D 00   3 0.1216D 01   4 0.1354D 01   5 0.1543D 01   6 0.1774D 01   7 0.2045D 01   8 0.2364D 01
   9 0.2707D 01  10 0.2968D 01  11 0.3572D 01
 BZ
   1 0.2251D 02   2 0.2251D 02   3 0.2251D 02   4 0.2251D 02   5 0.2250D 02   6 0.2250D 02   7 0.2250D 02   8 0.2250D 02
   9 0.2251D 02  10 0.2250D 02  11 0.2250D 02
 BFLUX
   1 0.2167D-01   2 0.6851D-01   3 0.2072D 00   4 0.4156D 00   5 0.6725D 00   6 0.9612D 00   7 0.1269D 01   8 0.1586D 01
   9 0.1908D 01  10 0.2244D 01  11 0.2559D 01
 TELEC
   1 0.2756D 03   2 0.2840D 03   3 0.2600D 03   4 0.2344D 03   5 0.2104D 03   6 0.1872D 03   7 0.1639D 03   8 0.1399D 03
   9 0.1434D 03  10 0.1163D 03  11 0.2000D 02
 TIONS
   1 0.2737D 03   2 0.2821D 03   3 0.2586D 03   4 0.2334D 03   5 0.2097D 03   6 0.1867D 03   7 0.1635D 03   8 0.1395D 03
   9 0.1422D 03  10 0.1158D 03  11 0.2000D 02
ZEPSLO=0.3181D-01  JEPS= 42


AT TIME=0.1867D-01  RUN TIMES= 21
   1 0.2251D 02   2 0.2251D 02   3 0.2251D 02   4 0.2251D 02   5 0.2250D 02   6 0.2250D 02   7 0.2250D 02   8 0.2250D 02
   9 0.2251D 02  10 0.2250D 02  11 0.2250D 02  12 0.2167D-01  13 0.6851D-01  14 0.2072D 00  15 0.4156D 00  16 0.6725D 00
  17 0.9612D 00  18 0.1269D 01  19 0.1586D 01  20 0.1908D 01  21 0.2244D 01  22 0.2559D 01  23 0.2756D 03  24 0.2840D 03
  25 0.2600D 03  26 0.2344D 03  27 0.2104D 03  28 0.1872D 03  29 0.1639D 03  30 0.1399D 03  31 0.1434D 03  32 0.1163D 03
  33 0.2000D 02  34 0.2737D 03  35 0.2821D 03  36 0.2586D 03  37 0.2334D 03  38 0.2097D 03  39 0.1867D 03  40 0.1635D 03
  41 0.1395D 03  42 0.1422D 03  43 0.1158D 03  44 0.2000D 02
ZEPSLO=0.3208D-01  JEPS= 42


AT TIME=0.2250D-01  RUN TIMES= 22
   1 0.2251D 02   2 0.2251D 02   3 0.2251D 02   4 0.2251D 02   5 0.2251D 02   6 0.2251D 02   7 0.2250D 02   8 0.2250D 02
   9 0.2251D 02  10 0.2250D 02  11 0.2250D 02  12 0.2541D-01  13 0.7246D-01  14 0.2109D 00  15 0.4191D 00  16 0.6758D 00
  17 0.9642D 00  18 0.1272D 01  19 0.1588D 01  20 0.1911D 01  21 0.2248D 01  22 0.2564D 01  23 0.2898D 03  24 0.2961D 03
  25 0.2707D 03  26 0.2428D 03  27 0.2171D 03  28 0.1929D 03  29 0.1685D 03  30 0.1459D 03  31 0.1529D 03  32 0.1192D 03
  33 0.2000D 02  34 0.2879D 03  35 0.2942D 03  36 0.2693D 03  37 0.2418D 03  38 0.2163D 03  39 0.1924D 03  40 0.1681D 03
  41 0.1454D 03  42 0.1516D 03  43 0.1188D 03  44 0.2000D 02
ZEPSLO=0.3167D-01  JEPS= 42
```

Table 4: Section of the ouptut from run using data of Table 3

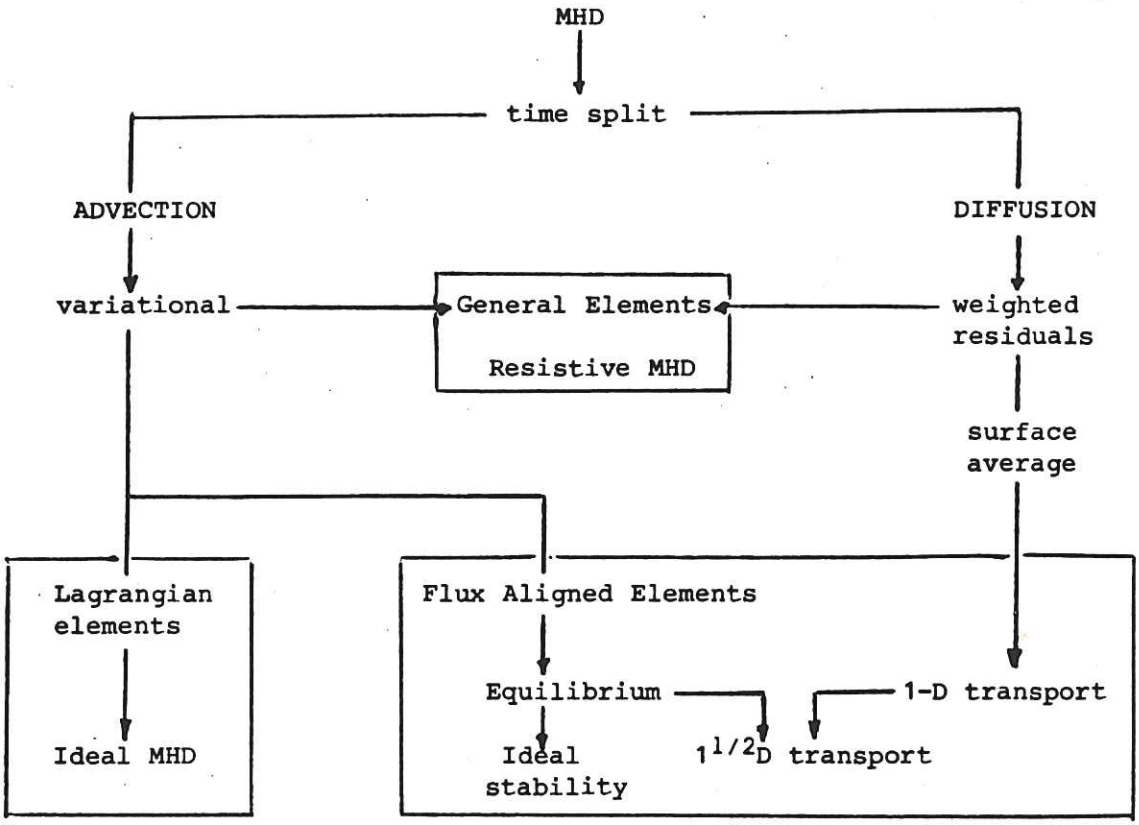

Fig. 1 The family of related finite element MHD models.

CLM-R253

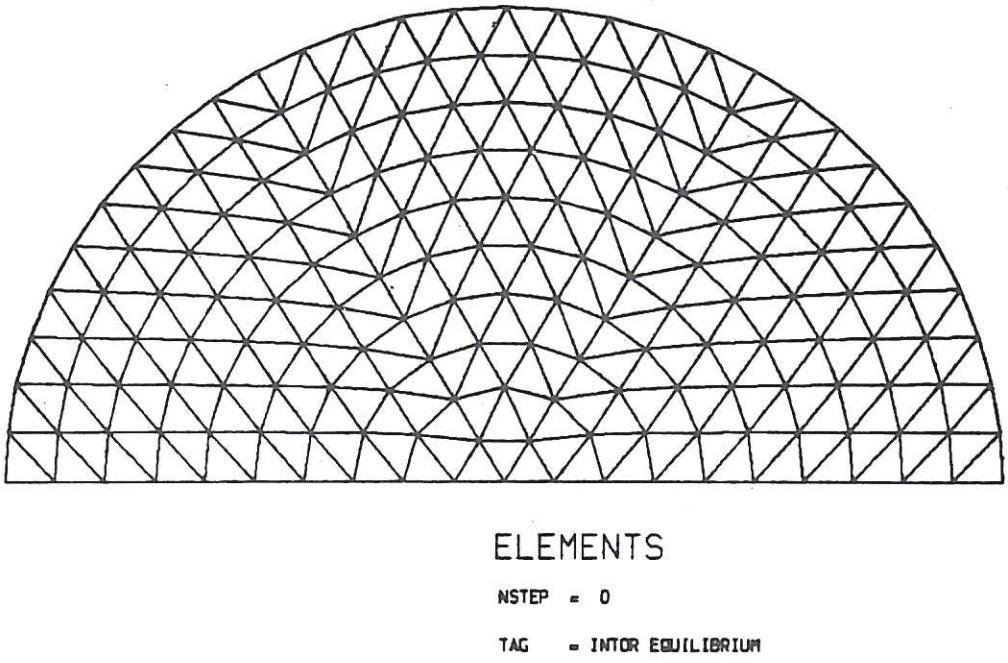

Fig. 2 Initial 2-D element triangulation.

CLM-R253

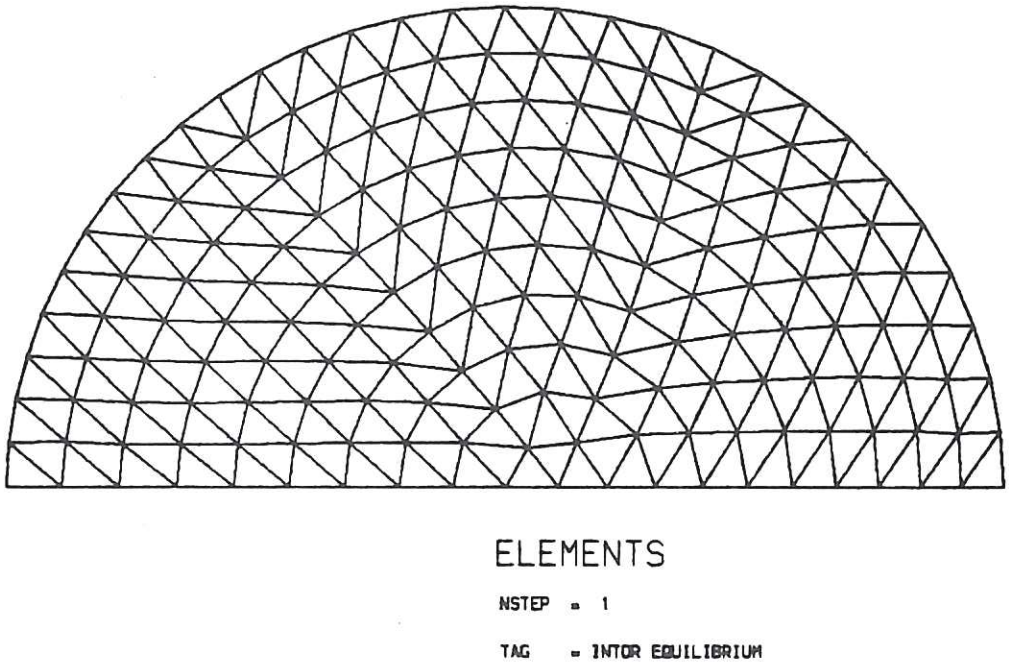

Fig. 3 Element triangulation after $\nabla p = j \times B$ is satisfied by element displacements.

CLM-R253

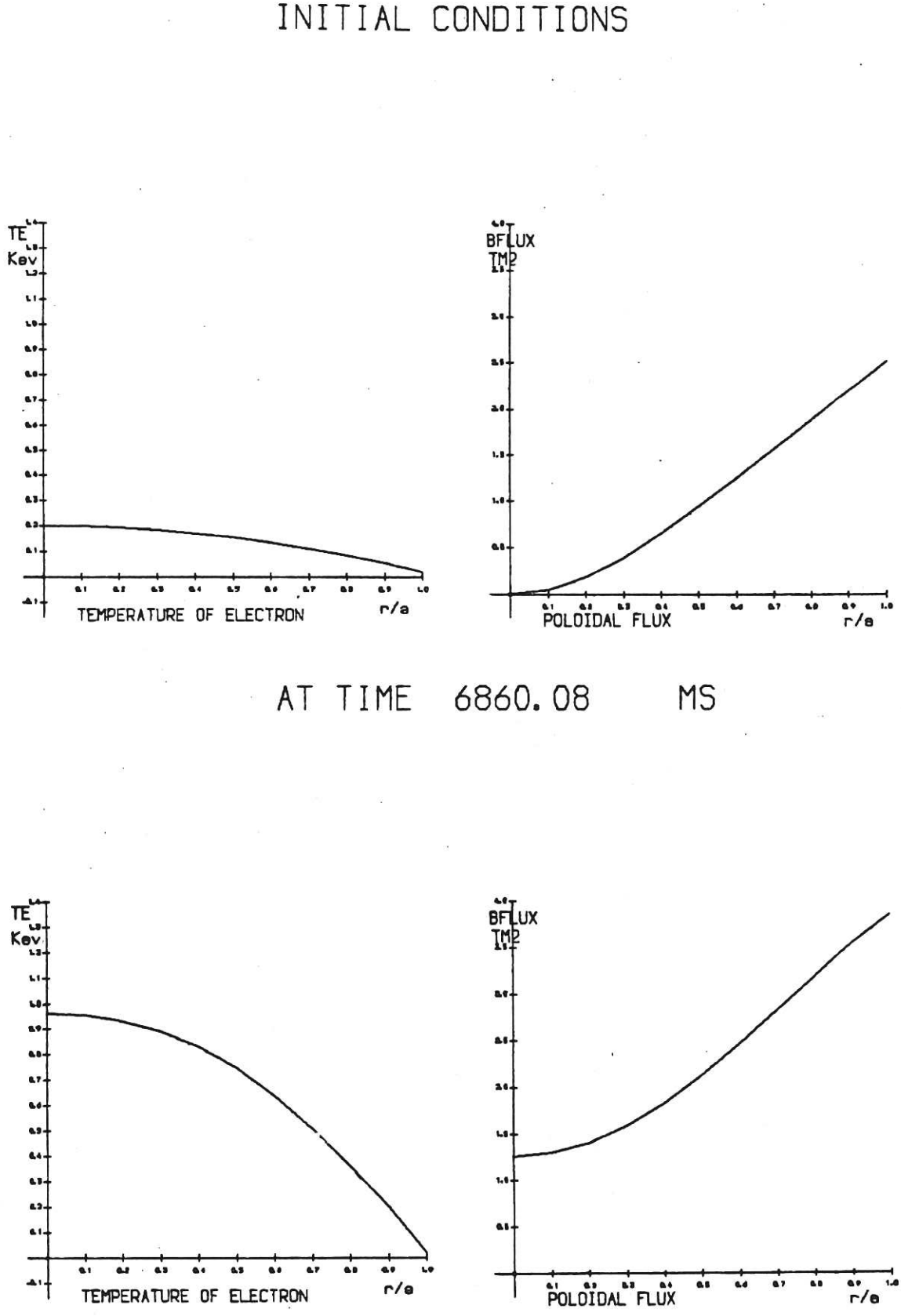

Fig. 4 a) Initial T_e and ψ profile.
b) Final T_e and ψ profile.

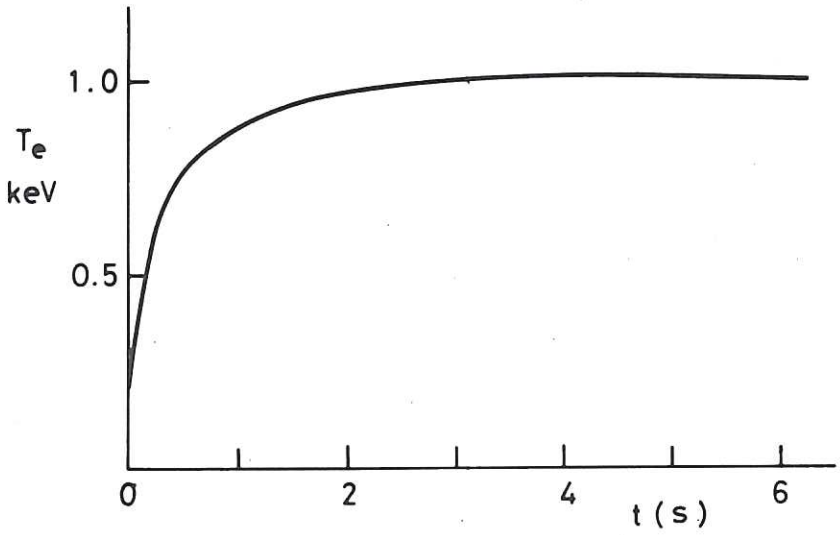

Fig. 5 Time variation of the axial equilibrium solution.

CLM-R253

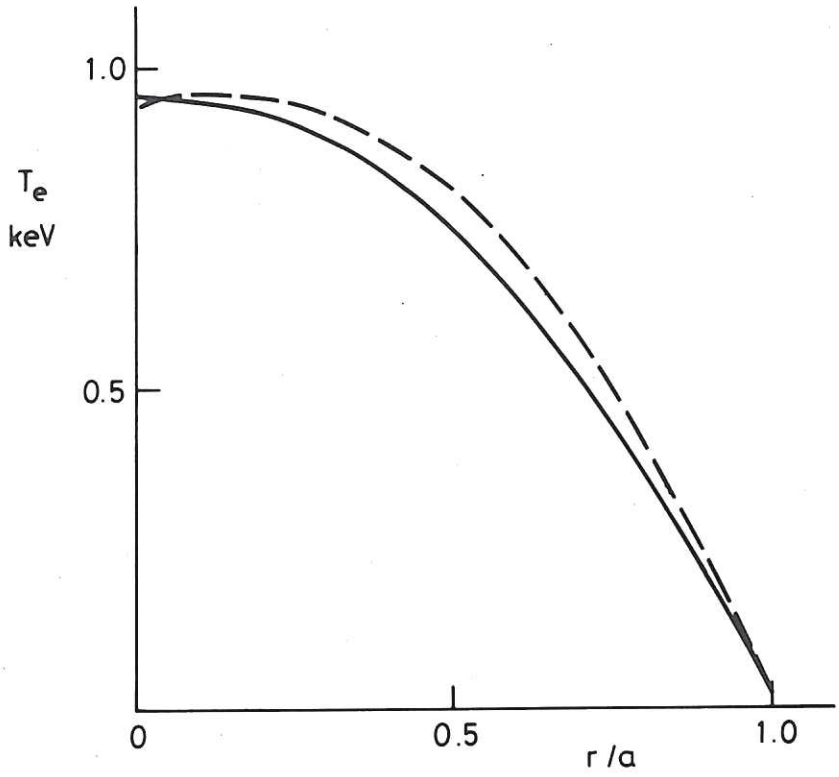

Fig. 6 Final T_e profile with (solid line) and without (broken line) equilibrium solution.

CLM-R253

Available from
HER MAJESTY'S STATIONERY OFFICE
49 High Holborn, London, WC1V 6HB
(Personal callers only)
P. O. Box 276, London, SE1 9NH
(Trade orders by post)
13a Castle Street, Edinburgh, EH2 3AR
41 The Hayes, Cardiff, CF1 1JW
Princess Street, Manchester, M60 8AS
Southey House, Wine Street, Bristol, BS1 2BQ
258 Broad Street, Birmingham, B1 2HE
80 Chichester Street, Belfast, BT1 4JY

PRINTED IN ENGLAND