JAERI-M
87-062

VECTORIZATION OF MHD EQUILIBRIUM AND STABILITY CODES

April 1987

Toshiyuki NEMOTO* and Toshihide TSUNEMATSU

日　本　原　子　力　研　究　所
Japan Atomic Energy Research Institute

編集兼発行　日本原子力研究所
印　　　刷　いばらき印刷㈱

Vectorization of MHD Equilibrium and Stability Codes

Toshiyuki NEMOTO* and Toshihide TSUNEMATSU

Department of Thermonuclear Fusion Research
Naka Fusion Research Establishment
Japan Atomic Energy Research Institute
Naka-machi, Naka-gun, Ibaraki-ken

(Received March 20, 1987)

An MHD equilibrium code (SELENE) and a stability code (ERATO-J) are extensively used for the analysis of ideal MHD beta limit of a tokamak plasma. High efficiency is required for the analysis of experimental data and the design of the next step fusion experimental devices. In the report, the methods of vectorization are described as well as the basic equations and numerical methods. Vectorization reduces the comptational time to about a third through a quarter of the original version on Fujitsu VP-100.

Keyword: MHD, Beta Limit, SELENE Code, ERATO-J Code, Vectorization, VP-100, Stability

---

* on leave from Fujitsu Ltd.
Present Address: Kanazawa Computer Service, Tokai, Naka, Ibaraki, Japan

MHD平衡・安定性コードのベクトル化

日本原子力研究所那珂研究所核融合研究部
根本　俊行*・常松　俊秀

（1987年3月20日受理）

MHD平衡コード（SELENE）および安定性解析コード（ERATO－J）は，トカマクプラズマにおける理想MHDベータ限界の解析によく使用されており，実験データの解析および次期核融合実験装置の設計においては，大量の計算がなされるため，コードの高速化が必要とされている。このレポートでは，これらのコードの基礎方程式，数値解法およびベクトル化の手法について述べる。このコードのベクトル化版は，富士通VP－100において，オリジナル版の3～4倍の計算時間の高速化を達成した。

＊外来研究員（富士通）
那珂研究所：〒311－02　茨城県那珂郡那珂町大字向山801－1

Contents

目　　次

1. Introduction

One of the critical issues in a tokamak fusion research is to improve the beta value of a plasma, where the beta is the ratio of the volume-averaged plasma pressure to the magnetic pressure. The maximum value of the beta in a tokamak plasma is theoretically evaluated by an ideal MHD stability analysis and the results of the theoretical prediction agree with those of experiments. A lot of calculations have been carried out to assess the beta limit for the design of the next step experimental device {1}. There still remain the differences in the results given by different authors. In the international collaborations for the design of a specific fusion reactor, such as INTOR workshops {1}, it is necessary to clarify the cause of the differences for the assessment of the data base. The enhancement also is necessary to improve the design of fusion reactor{2}. In addition the stability calculation is used to analyze the experimental data {3}. For these investigation, high efficiency in CPU time and I/O time is required to carry out a lot of equilibrium and stability calculations. In this report, we describe the methods of the vectorization in SELENE and ERATO-J codes for the Fujitsu VP-100 computer as well as the basic equations and numerical methods.

2. Equilibrium Code SELENE

2.1 Basic Equations

In the axisymmetric toroidal system, the equilibrium magnetic field B and current J are written by the poloidal flux function $\psi(R,Z)$ in the cylindrical coordinates (R,Z,φ):

$$B = \nabla\varphi \times \nabla\psi + F\nabla\varphi \quad (1)$$

and

$$\mu_0 J = \Delta^*\psi\nabla\varphi + \nabla F\times\nabla\varphi \ , \quad (2)$$

1. Introduction

One of the critical issues in a tokamak fusion research is to improve the beta value of a plasma, where the beta is the ratio of the volume-averaged plasma pressure to the magnetic pressure. The maximum value of the beta in a tokamak plasma is theoretically evaluated by an ideal MHD stability analysis and the results of the theoretical prediction agree with those of experiments. A lot of calculations have been carried out to assess the beta limit for the design of the next step experimental device {1}. There still remain the differences in the results given by different authors. In the international collaborations for the design of a specific fusion reactor, such as INTOR workshops {1}, it is necessary to clarify the cause of the differences for the assessment of the data base. The enhancement also is necessary to improve the design of fusion reactor{2}. In addition the stability calculation is used to analyze the experimental data {3}. For these investigation, high efficiency in CPU time and I/O time is required to carry out a lot of equilibrium and stability calculations. In this report, we describe the methods of the vectorization in SELENE and ERATO-J codes for the Fujitsu VP-100 computer as well as the basic equations and numerical methods.

2. Equilibrium Code SELENE

2.1 Basic Equations

In the axisymmetric toroidal system, the equilibrium magnetic field $\boldsymbol{B}$ and current J are written by the poloidal flux function $\psi(R,Z)$ in the cylindrical coordinates (R,Z,φ):

$$\boldsymbol{B} = \nabla\varphi \times \nabla\psi + F\nabla\varphi \qquad (1)$$

and

$$\mu_0 J = \triangle^*\psi\nabla\varphi + \nabla F\times\nabla\varphi \ , \qquad (2)$$

where

$$\triangle^*\psi = R^2\nabla \cdot (\nabla\psi/R^2) = R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial\psi}{\partial R}\right) + \frac{\partial^2\psi}{\partial Z^2} \ . \tag{3}$$

The equation for MHD equilibria, $\nabla P=J\times B$, can be reduced to the Grad-Shafranov equation,

$$\triangle^*\psi = -R^2\mu_0\frac{dP}{d\psi} - \frac{1}{2}\frac{dF^2}{d\psi} = g(R,\psi) \quad \text{(in a plasma)} \tag{4a}$$

and

$$\triangle^*\psi = 0 \qquad \text{(in a vacuum)} \ , \tag{4b}$$

when the plasma pressure is isotropic and the function of ψ. The poloidal current function , F ($F=RB_t$, B_t :toroidal magnetic field) is also the function of ψ. The functions P and F are arbitrary in eq.(4). The time-evolution of these functions are determined by a transport process. For the MHD stability analysis, P and F are usually given by using a simple model.

The shape of a plasma surface is specified by the functions,

$$R = R_0 + a\cos(\theta+\delta'\sin\theta) \ , \tag{5a}$$

and

$$Z = \kappa a\sin\theta \ , \tag{5b}$$

where R_0, κ and a are the major radius of the plasma center, the ellipticity and the minor radius, respectively. The parameter, δ', specifies the triangularity. The solution of the equation , $\triangle^*\psi_v=0$, gives the poloidal magnetic flux supplied by external coils (vacuum field solution). The general solutions, { ψ_{vi} }, are used to control a plasma shape. The vacuum flux is expressed by a linear combination of the general solutions:

$$\psi_v = \sum_{i=1}^{M} C_i\psi_{vi} \ . \tag{6}$$

The coefficients, $\{C_i\}$, are determined so that the flux contour with $\psi+\psi_v=\psi_s$ may pass the specified points on the plasma surface given by eq.(5), (ψ: the solution of the Grad-Shafranov equation, ψ_s: flux at the plasma surface). The condition that the contour with $\psi+\psi_v=\psi_s$ passes the specified points is too stringent for the coil systems in the design of experimental devices. For this purpose, a least square error can be minimized :

$$E = \sum_i a_i \mid (\psi+\psi_v)_i - \psi_{si} \mid^2 + \sum_i b_i I_i^2 = \min \ : \ , \tag{7}$$

where $\{a_i\}$, $\{b_i\}$ and I_i are the weights and the currents in the external coils. The Grad-Shafranov equation (eq.(4)) is solved in the rectangular domain , R^*, in the (R,Z) space (Fig.1). The poloidal flux function, ψ, is arbitrary by a constant which is chosen as $\psi_s=0$ at the plasma surface. By using this condition and the Green's theorem, the poloidal flux produced by a plasma current in a vacuum region is given by

$$\psi_p(\boldsymbol{r}) = \oint_{\psi=0} G(\boldsymbol{r},\boldsymbol{r}')\, B_p(R',Z')dl' \tag{8}$$

where

$$B_p = \mid \nabla\psi \mid \ / R \quad , \tag{9}$$

$$G(\boldsymbol{r},\boldsymbol{r}') = -\frac{1}{2\pi} \sqrt{RR'} / k \cdot \{(2-k^2)K(k) - E(k)\}, \tag{10a}$$

and

$$k = \frac{4RR'}{(R+R')^2 + (Z-Z')^2}. \tag{10b}$$

The functions K(k) and E(k) are the first and second complete elliptic integral, respectively. The Grad-Shafranov equation is numerically solved by using iterative method. The methods are described in 2.2 The boundary condition for the n-th iteration is given on the rectangular boundary, ∂R^*, by using the solution at the (n-1)th step;

$$\psi^n(\partial R^*) = \psi_p^{n-1}(\partial R^*) + \psi_v^n(\partial R^*) \;, \tag{11}$$

where ψ_v^n is calculated by using the condition $\psi^{n-1}+\sum C_i\psi_{vi}=0$ at the specified points of the plasma surface. When the iteration converges, the solution in an unbounded domain is obtained.

2.2 Numerical Methods

2.2.1 Nonlinear Eigenvalue Problem

When the inhomogeneous term in eq.(4), g (R,ψ), is given as the function of a normalized flux, $\overline{\psi}=1-\psi/\psi_0$ ($\psi_s=0$ and ψ_0 : poloidal magnetic flux at the axis), the semi-linear equation can be solved by using the algorithm of the nonlinear eigenvalue problem :

$$\triangle^*\psi^n = \lambda^n f(R,\overline{\psi}^{n-1}) \text{ (in a plasma)} \tag{12a}$$

$$\triangle^*\psi^n = 0 \qquad \text{(in vacuum)} \tag{12b}$$

with the boundary condition described in §2.1 . Equation (12) can be solved numerically in a rectangular domain by using the double-cyclic reduction method {4}. The eigenvalue at the n-th step, λ^n, is determined by $\lambda^n=(\psi_0/\psi_0^{n-1})\lambda^{n-1}$. The iteration converges when $|\lambda^n-\lambda^{n-1}|/\lambda^n<\varepsilon_\lambda$. The value, ψ_0, is obtained by a constraint :

$$I_p = \int \lambda^n f(R,\overline{\psi}^n)dRdZ = \text{given value} \tag{13}$$

or

$$q_0 = \frac{F}{2\pi}\oint\frac{dl}{R|\nabla\psi|}\bigg|_{\overline{\psi}=0} = \text{given value} \;, \tag{14}$$

where I_p and q_0 denote the total plasma current and the safety factor at the magnetic axis, respectively. This algorithm is useful when P and F are given as the function of $\overline{\psi}$.

2.2.2 Flux Conserving Tokamak (FCT) Algorithm

The Grad-Shafranov equation can be solved by specifying the profiles of the adiabatic invariant, $\mu(\bar{\psi})$, and the safety factor, $q(\bar{\psi})$, instead of $P(\bar{\psi})$ and $F(\bar{\psi})$:

$$\mu(\bar{\psi}) = P(\bar{\psi}) \left(\frac{dV}{d\chi}\right)^{\Gamma} , \tag{15}$$

and

$$q(\bar{\psi}) = \frac{1}{4\pi^2}\frac{d\chi}{d\psi} = \frac{F}{2\pi}\oint\frac{dl}{R^2 B_p} , \tag{16}$$

where χ, V and Γ are the toroidal magnetic flux, the volume surrounded by a magnetic surface and the specific heat ratio (Γ=5/3). This model describes a non-dissipative transport system and is called "Flux Conserving Tokamak (FCT)" model {5}. By substituting eq.(15) into the right hand side of the Grad-Shafranov equation (eq.(4)), we have

$$\frac{1}{R^2}\Delta^*\psi = -\mu_0 \frac{dV}{d\psi}\frac{d}{dV}\mu\left(4\pi^2 q\frac{d\psi}{dV}\right)^{\Gamma} - \frac{1}{R^2} F \frac{dF}{d\psi} . \tag{17}$$

This equation is the combination of an elliptic partial differential equation (PDE) and an ordinary differential equation (ODE). Equation (17) can be solved iteratively by using the Grad-Shafranov equation and the averaged equation on a magnetic surface {6}:

$$\frac{d}{dV}\left(\langle B^2{}_p\rangle \frac{dV}{d\psi}\right) = -\mu_0 \frac{dP}{d\psi} - \langle R^{-2}\rangle F \frac{dF}{d\psi}, \tag{18}$$

where

$$\langle X\rangle = \lim_{\Delta V\to 0}\int_{\Delta V} X d^3x \Big/ \int_{\Delta V} d^3x = 2\pi \frac{d\psi}{dV}\oint \frac{X dl}{B_p} . \tag{19}$$

By using eqs. (15) and (16), eq.(18) is written as

$$\frac{1}{F}\frac{dF}{d\psi} = -D , \tag{20a}$$

and

$$\frac{d\chi}{dV} = F\langle R^{-2}\rangle , \tag{20b}$$

where

$$D = \frac{\nu\langle R^{-2}\rangle(dK/d\psi) + \mu_0 F^{\Gamma-2}(d(\langle R^{-2}\rangle^{\Gamma}\mu)/d\psi)}{\langle R^{-2}\rangle + \nu K\langle R^{-2}\rangle + \mu_0\Gamma\langle R^{-2}\rangle F^{\Gamma-2}} , \tag{21}$$

$$K = \nu\langle R^{-2}\rangle\langle B^2_p\rangle \cdot 2\pi \oint\frac{dl}{B_p} , \tag{22}$$

and

$$\nu = \frac{1}{4\pi^2 q} . \tag{23}$$

The boundary condition of eq. (20) is given by

$$\chi\ (\bar{\psi}=0) = \chi(V=0) = 0 \tag{24a}$$

and

$$\chi\ (\bar{\psi}=1) = \chi\ (V=V_s) = 4\pi^2 \int_{\psi_0}^{0} q(\bar{\psi})d\psi . \tag{24b}$$

The nonliner equation can be solved iteratively :

$$F^n = C\ \exp(-\int_{\psi_0}^{\psi} D(F^{n-1})d\psi) , \tag{25}$$

and

$$\chi^n = \int_0^V F^n\langle R^{-2}\rangle dV. \tag{26}$$

The constant C is determined by the boundary condition (24b). The iteration converges when

$$|(d\chi^n/dV - d\chi^{n-1}/dV)/(d\chi^n/dV)| < \varepsilon_\chi . \tag{27}$$

The averaged quantities on a magnetic surface, $\langle X\rangle$, are obtained by solving the Grad-Shafranov equation (PDE) and the right hand side of PDE is obtained by using

$$F\frac{dF}{d\psi} = -F^2 D \tag{28}$$

and

$$\frac{dp}{d\psi} = \frac{d}{d\psi}\left(\mu\left(\frac{d\chi}{dV}\right)^{\Gamma}\right) \quad . \tag{29}$$

The ODE determines $F(\psi)=RB_t$ and the toroidal magnetic field at the plasma surface, F_s, changes from the specified value (the value of the vacuum toroidal field) due to the change in the pressure. To avoid the jump of the toroidal magnetic field, the adjustment of the plasma surface is necessary such that

$$E_F(\delta r) = | (F^l(\bar{\psi}=1) - F_s)/F_s | < \varepsilon_F \ . \tag{30}$$

Due to the modification of the plasma surface, the vacuum magnetic field to control the plasma surface also should be corrected . The alternative iteration of PDE and ODE converges when

$$E_M = \max\{ | (\psi^l(\bar{\psi})-\psi^{l-1}(\bar{\psi}))/\psi^l(\bar{\psi}))| , | (V^l(\bar{\psi})-V^{l-1}(\bar{\psi}))/V^l(\bar{\psi})| \ ,$$

$$| (\frac{dP^l}{d\psi}-\frac{dP^{l-1}}{d\psi})/\frac{dP^l}{d\psi} | , | (\frac{dF^l}{d\psi}-\frac{dF^{l-1}}{d\psi})/\frac{dF^l}{d\psi} | \} < \varepsilon_M \quad , \tag{31}$$

where l denotes the step of the iteration.

2.3 Critical Pressure to the Ballooning Modes and Local Interchange Mode

For a given $P(\bar{\psi})$ and $q(\bar{\psi})$, the stability of the ballooning mode and the local interchange mode are investigated. The equation of the high mode number stability is given at a magnetic surface by {7},

$$\frac{d}{dy} f(y) \frac{dG}{dy} + h(y)G = \omega^2 k(y)G \ , \tag{32}$$

where

$$f(y) = \frac{1}{\sqrt{g}\,|\nabla\psi|^2}\{1 + (\frac{|\nabla\psi|^2}{B}\frac{\partial z}{\partial\psi_\perp})^2\} \ , \tag{33}$$

$$h(y) = \frac{\sqrt{g}}{B^2}\mu_0\frac{dP}{d\psi}\frac{\partial}{\partial\psi_\perp}(2\mu_0 P + B^2) - \frac{F}{B^4}\mu_0\frac{dP}{d\psi}\frac{\partial z}{\partial\psi_\perp}\frac{\partial B^2}{\partial y} \ , \tag{34}$$

$$k(y) = \frac{1}{|\nabla\psi|^2}\{1 + (\frac{|\nabla\psi|^2}{B}\frac{\partial z}{\partial\psi_\perp})^2\}, \tag{35}$$

$$z(y) = \int_{y_0}^{y}\frac{\sqrt{g}F}{R^2}\,dy , \tag{36}$$

$$B^2 = (F^2 + |\nabla\psi|^2)/R^2 , \tag{37}$$

$$\frac{\partial}{\partial\psi_\perp} = \frac{\nabla\psi\cdot\nabla}{|\nabla\psi|^2} \tag{38}$$

and $\sqrt{g}$ is the Jacobian. The boundary condition of eq.(32) is given by

$$G(y=-\infty) = G(y=+\infty) = 0 . \tag{39}$$

When $\omega^2<0$, a ballooning mode is unstable at a magnetic surface. The marginal pressure, $dP^\infty/d\psi$, is obtained as the "eigenvalue" by solving the equation with $\omega^2=0$. The alternative iteration of the Grad-Shafranov equation and the ballooning equation with $\omega^2=0$ gives the critical pressure (the beta limit) for a given $q(\bar{\psi})$.

The asymptotic solution of eq.(32) is given by {7}

$$G(|y|\to\infty) \sim (\frac{\partial z}{\partial\psi_\perp})^\alpha , \tag{40}$$

where

$$\alpha = -\frac{1}{2} \pm\sqrt{1/4 - D} , \tag{41}$$

$$D = \frac{\mu_0(dP/d\psi)}{(4\pi^2 dq/d\psi)}\{(F^2Q_2+4\pi^2\frac{q}{F})(\mu_0\frac{dP}{d\psi}Q_3-\frac{d^2V}{d\psi^2}) + 4\pi^2\frac{dq}{d\psi}Q_1-\mu_0\frac{dP}{d\psi}F^2Q_1^2\} . \tag{42}$$

$$Q_1 = \frac{dV}{d\psi}\langle R^{-2}B_p^{-2}\rangle = 2\pi\oint\frac{dl}{R^2B_p^3} , \tag{43}$$

$$Q_2 = \frac{dV}{d\psi}\langle R^{-4}B_p^{-2}\rangle = 2\pi\oint\frac{dl}{R^4B_p^3} , \tag{44}$$

and

$$Q_3 = \frac{dV}{d\psi}\langle B_p^{-2}\rangle = 2\pi\oint \frac{dl}{B_p{}^3} \ . \tag{45}$$

The condition of a non-oscillatory solution is $D<1/4$ which is the stability criterion for the local interchange mode (the Mercier criterion (8)) :

$$M = M_s + M_w + M_p > 0 \ , \tag{46}$$

where

$$M_s = \frac{1}{4}(4\pi^2\frac{dq}{d\psi})^2 = C_1 \ , \tag{47}$$

$$M_w = -\mu_0\frac{dP}{d\psi}C_2 = -\mu_0\frac{dP}{d\psi}\{4\pi^2\frac{dq}{d\psi}Q_1 - \frac{d^2V}{d\psi^2}(F^2Q_2 + 4\pi^2\frac{q}{F})\} \ , \tag{48}$$

and

$$M_p = -(\mu_0\frac{dP}{d\psi})^2C_3 = -(\mu_0\frac{dP}{d\psi})^2\{F^2(Q_2Q_3-Q_1{}^2) + 4\pi^2\frac{q}{F}Q_3\} \ . \tag{49}$$

The ballooning equation with $\omega^2=0$ is solved in a bounded domain of y, $[0,2\pi N]$, assuming $y_0=0$ for a up-and-down symmetric case , where N is the numbers of turns in the integration of the equation. The marginal equation is solved numerically by using the Runge Kutta Method or the matrix method with the boundary conditions

$$G(0) = \text{finite} \ , \tag{50}$$

and

$$G(2\pi N) = 0 \ . \tag{51}$$

When the Mercier criterion is violated, the marginal equation has the oscillatory solution and the boundary condition (51) can not be used. In this case, the marginal pressure $dP^{\infty}/d\psi$ is obtained by using the criterion of the local interchange mode:

$$\mu_0 dP^{\infty}/d\psi = -(C_2 + \sqrt{C_2{}^2+4C_1C_3})/(2C_3) \ . \tag{52}$$

2.4 Structure of Code

Figure 2 shows the brief sequence of SELENE code. In STEQU, the initial equilibrium to increase the beta value is obtained by using the nonlinear eigenvalue problem for a given $P(\psi)$ and $F(\psi)$ in eq.(4a). Equation (12a) is solved by using the double-cyclic reduction method in EQPDE. The right hand side of eq.(12a) is calculated in EQRCU. In these procedures, subroutines, EQBND and EQADJ, are called to adjust the vacuum magnetic field or coil current so that the plasma surface may pass through the specified points given by eqs.(5a) and (5b). The beta value is increased by fixing $q(\psi)$ obtained in STEQU (FCT processes). The function, $F(\psi)$, is calculated by solving an ordinary differential equation eq.(18) in EQODE. The averaged quantities on a magnetic surface are obtained in EQLIN. The critical pressure to the ballooning modes is evaluated in BLPDS by solving the eignvalue equation, eq.(32), for $dP^{\infty}/d\psi$ with $\omega^2=0$. By using $dP^{\infty}/d\psi$ and $q(\psi)$, the next step of equilibrium is obtained.

2.5 Vectorization of SELENE Code

The computational cost of the original version is evaluated by using a software, FORTUNE, which is offered by Fujitsu Ltd. Table 1 shows the result of the cost evaluation. Most expensive routines are BLPDS, FLUX, EQPDE, and EQLIN.

(i) BLPDS

The subroutine, BLPDS, solves the critical pressure due to the ballooning modes by using the Runge-Kutta integration and the shooting method. In the original version, the integration is carried out on each magnetic surface and we have no vectorized procedure in this routine. Figure 3 shows the source program for the shooting method. When the shooting is unsuccessful, a jumping out of the DO loop occurs. The eigenvalue, FAC, is obtained by using the bisection method. When the solution has a zero point, the pressure gradient is reduced in the block of the statement number 40. If the solution tends to diverge, we can increase the pressure gradient in the block of the statement number 30. For the vectorization of the integration and the bisection method, we solve the equations simultaneously on magnetic surfaces. The vectorized version of source program is shown in Fig.4. As the shooting is not successful on every magnetic surface we use a list vector to specify the equations to be solved (ISN54,ISN153-155 in Fig.4). If a solution is out of a certain range ($G<0$ or $G>10$), the number of the magnetic surface is eliminated from the list vector. When the initial value of the eigenvalue is not good approximation, the shooting fails of success on many surfaces and the vector length becomes short in the DO loop (ISN53). The computational cost, NL, in the integration is shown in Fig.5 as the function of the vector length, where N and L are the steps of integration along a magnetic surface and the vector length, respectively. The computational cost for $L>7$ is larger than that for $L\leq7$, where $L=7$ is the break-even vector length between scalar and vector calculation on VP-100. The efficiency, α=(vector processing speed)/(scalar processing speed), can be expected to be, $\alpha\sim2$, if we use scalar calculations for $L<7$. We specify the scalar calculation for the short vector case by using *VOCL LOOP, SCALAR.

(ii) FLUX

In this subroutine, the poloidal flux at the boundary of the rectangular domain given by a plasma current is calculated by using eq.(8). In the original version, the poloidal flux at a specified point is obtained in the function subroutine (Fig.6). We vectorize this procedure by introducing DO loop for the points on the rectangular boundary in Fig.1 The vectorized subroutine is shown in Fig.7.

(iii) EQLIN

In this subroutine, the averaged quantities on a magnetic surface are calculated. The crossing points between a magnetic surface and the rectangular meshes change on each surface. The points increases as a magnetic surface becomes close to a plasma surface. The integrations along magnetic surfaces can be vectorized by using the method of the list vector.

(iv) EQPDE

This subroutine is already vectorized in the original version. In a special case which never occur in this code, several statements become recursive. The special case is omitted by using *VOCL LOOP, NOVREC. The vector length changes from NR to 2 in the double cyclic reduction method, where NR is the mesh numbers in the R direction and is usually taken NR=129 or 257. Other algorithm, e.g. FACR method {4} should be used to avoid the reduction of the vector length.

In the SELENE Code, four types of vectors appears:

Type A : Vector length is long, $L \leqq 50$, and main procedures are consist of simple arithmetics.

Type B : Vector length changes from a long one to a short one.

Type C : Short vectors are included

Type D : Vector length is long but IF statements are included.

The efficiency is defined by {9},

$$P=1/\left(1-\sum_i v_i+\sum_i \frac{v_i}{\alpha_i}\right) \quad , \tag{53}$$

where v_i =cost×vectorization rate and is given in Table 1 and α_i is the ratio of the vector processing speed and the scalar processing speed. We assume the values of { α_i } as in Table 2 for each type of vector length. In SELENE, the predicted value of P is about 3. When the mesh points are $NR \times NZ$ = 129×65 , the computational times are shown in Table 3 for the original version, the vectorized version with the scalar computation and the vector calculation. The vectorized version takes more computational times than the original version, when the computation is carried out in scalar. This is mainly due to the list vectors in

BLPDS and EQLIN. The guess value of P agrees with the ratio of the scalar and the vector processings for the vectorized version.

3. Stability Code ERATO-J

3.1 Basic Equation

The stability of the ideal MHD modes is studies by minimizing a Lagrangean {10},

$$L = W_P + W_V - \omega^2 W_K \ , \tag{54}$$

$$W_p = \frac{1}{2}\int_P d^3x \, [\, |\boldsymbol{Q} + (\boldsymbol{n}\cdot\xi)(\boldsymbol{J}_0\times\boldsymbol{n})|^2 + \Gamma P_0 |\nabla\cdot\xi|^2 - 2|\boldsymbol{n}\cdot\xi|^2(\boldsymbol{J}_0\times\boldsymbol{n})\cdot(\boldsymbol{B}_0\cdot\nabla)\boldsymbol{n}] \ , \boldsymbol{Q} = \nabla\times(\xi\times\boldsymbol{B}_0) \tag{55}$$

$$W_V = \frac{1}{2}\int_V d^3x \, |\nabla\times\boldsymbol{A}|^2 \ , \tag{56}$$

and

$$W_K = \frac{1}{2}\int_P d^3x \rho_0 |\xi|^2 \ . \tag{57}$$

Here ξ is the displacement of the fluid element, n is the unit vector normal to the equilibrium magnetic surface ($\boldsymbol{n}=\nabla\psi/|\nabla\psi|$), and ρ_0 is the mass density. The quantities with a subscript 0 denote ones in an equilibrium. The perturbation of the vacuum energy in eq.(73) is given by using the vector potential, $\boldsymbol{A}$, and the boundary conditions for ξ and $\boldsymbol{A}$ are given by {10}

$$\boldsymbol{n}\times\boldsymbol{A} = -(\boldsymbol{n}\cdot\xi)B_0 \quad \text{at the plasma surface} \ , \tag{58}$$

and

$$\boldsymbol{n}\times\boldsymbol{A} = 0 \qquad \text{at the conducting shell or infinity.} \tag{59}$$

BLPDS and EQLIN. The guess value of P agrees with the ratio of the scalar and the vector processings for the vectorized version.

3. Stability Code ERATO-J

3.1 Basic Equation

The stability of the ideal MHD modes is studies by minimizing a Lagrangean {10},

$$L = W_P + W_V - \omega^2 W_K \ , \tag{54}$$

$$W_p = \frac{1}{2}\int_P d^3x\,[\,|\boldsymbol{Q} + (\boldsymbol{n}\cdot\xi)(\boldsymbol{J}_0\times\boldsymbol{n})|^2 + \Gamma P_0|\nabla\cdot\xi|^2 - 2|\boldsymbol{n}\cdot\xi|^2(\boldsymbol{J}_0\times\boldsymbol{n})\cdot(\boldsymbol{B}_0\cdot\nabla)\boldsymbol{n}] \ , \boldsymbol{Q} = \nabla\times(\xi\times\boldsymbol{B}_0) \tag{55}$$

$$W_V = \frac{1}{2}\int_V d^3x\,|\nabla\times\boldsymbol{A}|^2 \ , \tag{56}$$

and

$$W_K = \frac{1}{2}\int_P d^3x\rho_0|\xi|^2 \ . \tag{57}$$

Here ξ is the displacement of the fluid element, $\boldsymbol{n}$ is the unit vector normal to the equilibrium magnetic surface ($\boldsymbol{n}=\nabla\psi/|\nabla\psi|$), and ρ_0 is the mass density. The quantities with a subscript 0 denote ones in an equilibrium. The perturbation of the vacuum energy in eq.(73) is given by using the vector potential, $\boldsymbol{A}$, and the boundary conditions for ξ and $\boldsymbol{A}$ are given by {10}

$$\boldsymbol{n}\times\boldsymbol{A} = -(\boldsymbol{n}\cdot\xi)\boldsymbol{B}_0 \quad \text{at the plasma surface ,} \tag{58}$$

and

$$\boldsymbol{n}\times\boldsymbol{A} = 0 \qquad \text{at the conducting shell or infinity.} \tag{59}$$

The weakly unstable MHD modes localize near the rational surface where $q(\psi)$ takes a rational number. For the accurate calculation of the eigenvalue, ω^2 , and the eigenvector, it is necessary to use a flux surface coordinate, (ψ,χ,φ), where χ is the azimuthal coordinate. In the axisymmetric system, the equilibrium quantities are independent of φ and the Largangean can be written in the form of the single summation with respect to the toroidal mode number, n,

$$L = \sum_n L_n \ , \tag{60}$$

and

$$\xi(\psi,\chi,\Phi) = \sum_n \xi_n(\psi,\chi)e^{in\varphi} \ . \tag{61}$$

The Fourier-component, $\xi_n(\psi,\chi)$, is written in the contravariant form :

$$\xi_n = R^2X(\nabla\chi\times\nabla\varphi) + R^2V\nabla\varphi\times\nabla\psi + R^2YB_0 \ . \tag{62}$$

3.2 Numerical Methods

The details of the numerical methods of the stability code, ERATO, is described in Ref.{11}. Here, The most important procedures in the ERATO-J code are described, i.e. the mapping from the (R,Z,φ) coordinate to the flux coordinate, (ψ,χ,φ) and the eigenvalue solver.

The azimuthal coordinate, χ, is defined by

$$\chi = \int_0^l \frac{dl}{\sqrt{g}B_p} \ \text{with} \ 2\pi = \oint_\psi \frac{dl}{\sqrt{g}B_p} \ , \tag{63}$$

where $\sqrt{g}$ is the Jacobian of the flux coordinate system. One of the typical coordinate systems is given by

$$\sqrt{g} = \frac{qR^2}{F} \ . \tag{64}$$

In this coordinate system, the angle between the toroidal and poloidal magnetic field lines is constant on a magnetic surface:

$$\frac{B^{\varphi}}{B^{\chi}} = \frac{\sqrt{g}F}{R^2} = q(\psi) \ , \tag{65}$$

where B^{φ} and B^{χ} are the contravariant components of the magnetic field. This coordinate system is called "a natural coordinate system".

For the mapping, the trace of the magnetic surface and the numerical derivatives with the high accuracy are inevitable. In the ERATO-J code, the 3rd order or the 5th order spline interpolation is used in the (R,Z) space. The magnetic surface is traced by solving the equation of the magnetic field line:

$$\frac{dR}{dl} = \frac{1}{|\nabla\psi|}\frac{\partial\psi}{\partial Z} \ , \ \frac{dZ}{dl} = -\frac{1}{|\nabla\psi|}\frac{\partial\psi}{\partial R} \ , \tag{66}$$

where dl is the element of the arc length along the magnetic surface. The differential equations (66) are solved by using the 4th order Runge-Kutta method. Along the magnetic surface, the derivatives of $\psi(R,Z)$ are calculated by using the two dimentional spline function.

Discretization of L_n in the (ψ, χ) plane and the variation with respect to lead to the generalized eigenvalue problem {11}

$$Ax = \omega^2 Bx \tag{67}$$

, where A is a symmetric matrix and B is a positive symmetric matrix. The minimum negative eigenvalue gives the growth rate, $\Gamma=\sqrt{-\omega^2}$. The eigenvalues are classified into four classes, the fast wave modes, the Alfvén wave modes, the slow wave modes and the unstable modes. There appear the continuum spectra in the Alfvén and slow wave modes. In Fig.8 the schematic distribution of the eigenvalues is shown. The unstable modes are located below the origin of the continuum spectra. The matrices A and B have the structure of a block diagonal and each block is consist of a sparse submatrix with the band width of 7 (Fig.9). The overlapped block corresponds to the radial component X. This structure of the reflection of the ideal MHD approximation, which contains the radial derivatives only in the radial component. The size of the block is $8N_\chi + 8$ and the matrices A and B are consist of N_ψ

blocks, where N_ψ and N_χ are the numbers of the radial and the azimuthal meshes, respectively. In usual calculation, the meshes of N_ψ=N_χ=100 are used and the size of the matrices A and B becomes $6N_\psi(N_\chi+1)$ = 60600 with the band width of 808. Taking account of the structure of the spectrum and the sparseness of the matrices, we use the inverse iteration method with the shift of the origin to solve the eq.(67) :

Step1 $\tilde{A}x=(A-\omega_0^2B)x=(\omega^2-\omega_0^2)Bx$ (68)

Step2 Initial vector x_0

Step3 Solution of $\tilde{A}x^{k+1}=Bx^k$

Step4 Normalization to $x^{k+1}Bx^{k+1}=1$

Step5 If $\max_j|x_j^{k+1}-x_j^k|>\varepsilon$ then go to step3

Step6 $\omega^2=\omega_0^2+(x^{k+1}Ax^{K+1})/(x^{k+1}Bx^{k+1})$.

We can hold the sparseness to solve the linear simultaneous equation, eq.(68), by using Scott's algorithm {12}. The combination of the submatrices corresponding to the V and Y components leads to the following linear simultaneous equations in a block:

$$A_1Z_1 + A_2Z_2 + A_3Z_3 = U_1 \quad (69)$$

$$\text{<previous block>} + A_2^TZ_1 + A_4Z_2 + A_5Z_3 = U_2 \quad (70)$$

$$A_3^TZ_1 + A_5^TZ_2 + A_6Z_3 + \text{<next block>} = U_3, \quad (71)$$

where $Z_1=(Y, V)$, $Z_2 = X_1$ and $Z_3 = X_2$. Equation (69) is separated from the previous and the next blocks and Z_1 is expressed by $Z_1 = A_1^{-1}(U_1 - A_2Z_2 - A_3Z_3)$. Substitution of Z_1 to eqs.(70) and (71) gives 2×2 block simultaneous equations :

$$\text{<previous block>} + \hat{A}_4Z_2 + \hat{A}_5Z_3 = \hat{U}_2 \quad (72)$$

$$\hat{A}_5^TZ_2 + \hat{A}_6Z_3 + \text{<next block>} = \hat{U}_3 , \quad (73)$$

where

$$\hat{A}_4 = A_4 - A_2{}^T A_1{}^{-1} A_2,\ \hat{U}_2 = U_2 - A_2{}^T A_1{}^{-1} U_1, \tag{74}$$

$$\hat{A}_5 = A_5 - A_2{}^T A_1{}^{-1} A_3, \tag{75}$$

$$\hat{A}_6 = A_6 - A_3{}^T A_1{}^{-1} A_3,\ \hat{U}_3 = U_3 - A_3{}^T A_1{}^{-1} U_1. \tag{76}$$

Elimination of Z_2 in eqs.(73) and (74) gives the last overlapping block

$$Z_2 = \hat{A}_4{}^{-1}\ (\hat{U}_2 - \hat{A}_5 Z_3)\ , \tag{77}$$

$$\tilde{A}_6 Z_3 + \langle\text{next block}\rangle \qquad = \tilde{U}_3, \tag{78}$$

$$\tilde{A}_6 = \hat{A}_6 - \hat{A}_5{}^T \hat{A}_4{}^{-1} \hat{A}_5, \tag{79}$$

$$\tilde{U}_3 = \hat{U}_3 - \hat{A}_5{}^T \hat{A}_4{}^{-1} \hat{U}_2. \tag{80}$$

The subblock $\tilde{A}_6$ becomes the new overlapping block in the next block. The inversion of a matrix is expressed by the LU decomposition and the sparseness of the matrix can be held. In this algorithm only the overlapping block becomes a dense matrix. The solution of the linear simultaneous equation is obtained by using a forward and a backward substitutions.

Forward substitution :

$$\hat{U}_2 = U_2 - A_2{}^T A_1{}^{-1} U_1, \tag{81}$$

$$\hat{U}_3 = U_3 - A_3{}^T A_1{}^{-1} U_1, \tag{82}$$

$$\hat{U}_3 = \hat{U}_3 - (A_5 - A_3{}^T A_1{}^{-1} A_2) \hat{A}_4{}^{-1} \hat{U}_2\ , \tag{83}$$

Replacing U_2 of the next block by $\hat{U}_3$. (84)

Backward substitution :

$$Z_3 = \hat{A}_6^{-1}\tilde{U}_3 \tag{85}$$

$$Z_2 = \hat{A}_4^{-1}(\hat{U}_2 - (A_5 - A_2^T A_1^{-1} A_3) Z_3) , \tag{86}$$

$$Z_1 = A_1^{-1}(U_1 - A_2 Z_2 - A_3 Z_3) , \tag{87}$$

Replacing Z_3 of the previous block by Z_1 . (88)

3.3 Structure of ERATO-J Code

The ERATO-J code is consist of four modules, i.e. ERATOS, ERATO2, ERATO4 and ERATO5. In ERATOS, the main procedures are the mapping of geometrical quantities from the rectangular meshes to the ψ and χ meshes and the construction of matrices A and B. ERATO2 solves eq.(56) to obtain the perturbation of the magnetic field in the vacuum region. In ERATO4, the eigenvalue problem (67) is solved. This module takes more than 80% of the computational time and the vectorization of this module increases the efficiency of the ERATO-J code. ERATO5 is used for the summary and the graphic plot of the results. In Fig.10 the brief flow of ERATO4 is shown. Each block corresponds to each step in §2.3. In each block, several subroutines are called. The tree structure is shown in Fig.11.

3.4 Vectorization of ERATO4

In the inverse iteration method, three kinds of vector calculations appear :

(i) SAXPY : $\boldsymbol{y} = \boldsymbol{y} + a\boldsymbol{x}$ (a is scalar) ,
(ii) SDOT : $S = \sum x_i y_i$,
(iii) SXYPZ : $Z_i = Z_i + x_i y_i$.

The cost of these calculations are about 92% of the whole arithmetics. The original version of ERATO4 was developed by Scott and Gruber {12} for CRAY-1 computer. The names of the arithmetics are those of mathematical subroutines in CRAY FORTRAN. For other computers than CRAY, the subroutines written in FORTRAN are prepared. In Fig.12 those subroutines are shown. In VP-100, the subroutines SAXPY and SXYPZ cannot be vectroized because the compiler assumes the cases of NY=0 and NZ=0. The cases never occur in ERATO and we can force the vectorization by using *VOCL LOOP, NOVREC. As these subroutines are very short, we expand the procedures to the upper level of routine where the subroutines are called. The expansion reduces the CPU time by 18% in scalar calculations. Table 4 shows the cost each routine, C, the relative rate of the vectorization in a routine, V, and the rate of the vectorzation, v=CV, for the decomposition of a matrix $\tilde{A}$. The types of the vector and typical vector length are also shown. The forward and backward substitution use about 7% cost. The rate of the vectorization is summarized for the type of the vector in Table 5. Assuming the efficiency parameter, α, as shown in Table 5, we predict the total efficiency $P\sim 3$ through 4. The dependency of P on meshes obtained in VP-100 is shown in Fig.13.

4 Summary and Discussions

We described the methods of vectorization in the equilibrium code SELENE and the stability code ERATO-J. In SELENE code, we use the list vector to vectorize the integrations of the ballooning equation on magnetic surfaces. However, in BLPDS, the integration with the vector length of less than 7 takes a third through a half of the computational time. This is one of the reasons that the enhancement of the efficiency is limited by 2.5 through 3. In the ERATO-J code, only ERATO4 was vectorized. The vectorization has been already done and the main effort was made for the analysis of the cost and the type of the vector. Due to the vectorization of ERATO4, the relative computational for CRAY-1 computer. The names of the arithmetics are those of mathematical subroutines in CRAY FORTRAN. For other computers than CRAY, the subroutines written in FORTRAN are prepared. In Fig.12 those subroutines are shown. In VP-100, the subroutines SAXPY and SXYPZ cannot be vectroized because the compiler assumes the cases of NY=0 and NZ=0. The cases never occur in ERATO and we can force the vectorization by using *VOCL LOOP, NOVREC. As these subroutines are very short, we expand the procedures to the upper level of routine where the subroutines are called. The expansion reduces the CPU time by 18% in scalar calculations. Table 4 shows the cost each routine, C, the relative rate of the vectorization in a routine, V, and the rate of the vectorzation, $v=CV$, for the decomposition of a matrix $\tilde{A}$. The types of the vector and typical vector length are also shown. The forward and backward substitution use about 7% cost. The rate of the vectorization is summarized for the type of the vector in Table 5. Assuming the efficiency parameter, α, as shown in Table 5, we predict the total efficiency $P\sim3$ through 4. The dependency of P on meshes obtained in VP-100 is shown in Fig.13.

4 Summary and Discussions

We described the methods of vectorization in the equilibrium code SELENE and the stability code ERATO-J. In SELENE code, we use the list vector to vectorize the integrations of the ballooning equation on magnetic surfaces. However, in BLPDS, the integration with the vector length of less than 7 takes a third through a half of the computational time. This is one of the reasons that the enhancement of the efficiency is limited by 2.5 through 3. In the ERATO-J code, only ERATO4 was vectorized. The vectorization has been already done and the main effort was made for the analysis of the cost and the type of the vector. Due to the vectorization of ERATO4, the relative computational time in ERATOS increases. The vectorization of the procedures for the mapping described in §3.2 is required.

Acknowledgments

The authors wish to express their sincere thanks to Drs. M. Azumi and S. Seki of the Depertment of Large Tokamak Reserch and Dr. S. Tokuda of Plasma Theory Laboratory, for their fruitful discussions and comments on the equilibrium and stability analysis. They also thanks Dr. T. Matsuura of FACOM-HITAC Ltd. and Dr. M. Makino of JAERI computer center on leave from Fujitsu Ltd. for their valuable suggestions on the vectorization in VP-100. One of the authors, T. Nemoto, acknowledges Dr. T. Takeda, the manager of Plasma Theory Laboratory for the opportunity to study plasma physics and computational methods in MHD analysis.

References

{1} INTOR Phase Two A Part II (IAEA, Vienna) (1986) 453.

{2} T. Tsunematsu et al., "Second Stability Access in Tokamak Plasmas", IAEA-CN-47/E-I-2-1 (1986).
S. Seki et al., to be published in Nucl Fusion.

{3} T. Ozeki et al., "Kink Instability of the Divertor Configuration in JT-60", JAERI-M 87-004 (1987).

{4} R. W. Hockney, *Method in Computational Physics* Vol.9 (Academic Press, New York) (1970) 135.

{5} J. F. Clarke and D. J. Sigmar, Phys. Rev. Lett. 38 (1977) 70.

{6} H. Grad et al., Proc. Nat. Acad. Sci. (USA) 72 (1975) 3789.

{7} J. W. Connor et al., Proc. Roy. Soc.(London) A365 (1979) 1.

{8} C. Mercier, Nucl. Fusion 1 (1960) 47.
Y. -K. M Peng et al., Phys. Fluids 21 (1978)

{9} T. Matsuura et al. Comput. Phys. Commun. 26 (1982) 377

{10} I. B. Bernstein et al., Proc. Roy. Soc. (London) 244 (1958) 17.

{11} R. Gruber et al., Comput. Phys. Commun. 21 (1981) 323.

{12} D. S. Scott and R. Gruber, "Implementing Sparce Matrix Technique in the ERATO Code", Lausanne Report, LRP 181/81 (1981).

Table 1 Cost and relative rate of the vectorization in each routine

	Original Ver.		Vectorized Ver.	
	Cost (%)	V-rel (%)	Cost (%)	V-rel (%)
BLPDS	31.3	0.0	33.9	84.58
FLUX	19.2	0.0	18.0	99.93
EQPDE	18.4	30.69	17.3	97.18
EQLIN	14.0	0.0	15.0	95.42
EQRBP	5.0	99.94	4.7	99.94
EQADJ	2.4	99.94	1.8	99.94
EQRCU	2.1	89.89	1.9	93.39

Table 2 Types of vectors and efficiency parameter, α

Routine	v (%)	type	α
BLPDS	0.287	C	2
FLUX	0.180	D	10
EQPDE	0.168	B	10
EQLIN	0.143	A	15
others	0.080	-	15
Total	0.85		

Table 3 CPU time and relative efficiency

	Original(Scalar)	Vectorized(Scalar)	Vectorized(Vector)
Time	251.63 sec	317.33 sec	106.58 sec
Ratio	1	1.26	0.42

Table 4 Cost and vectorization rates, V-ral and v in FACMAT

	cost	V-rel	v	type	L
FACMAT					
ALBCON	0.4	0.98	0.4	B	N→1
LBDDSL	14.1	0.74	10.4	C	7
UBDSOL	8.5	0.67	5.7	C	7
ODTMLT	13.4	1.0	13.4	A	N
CALD	4.8	0.8	3.8	B	N→1
CACA2	17.9	1.0	17.9	B	N→1
LTRDSL	19.9	0.97	19.3	B	N→1
UTRSOL	12.6	0.97	12.2	B	N→1
Total	91.6	－	83.1	－	－

Table 5 Types of vectors and efficiency parameter α. In v, the procedures for the forward and backward substitutions are included

Type	v	α
A	17.9%	15
B	55.7%	10
C	17.8%	1 ~2
D	0.3%	10

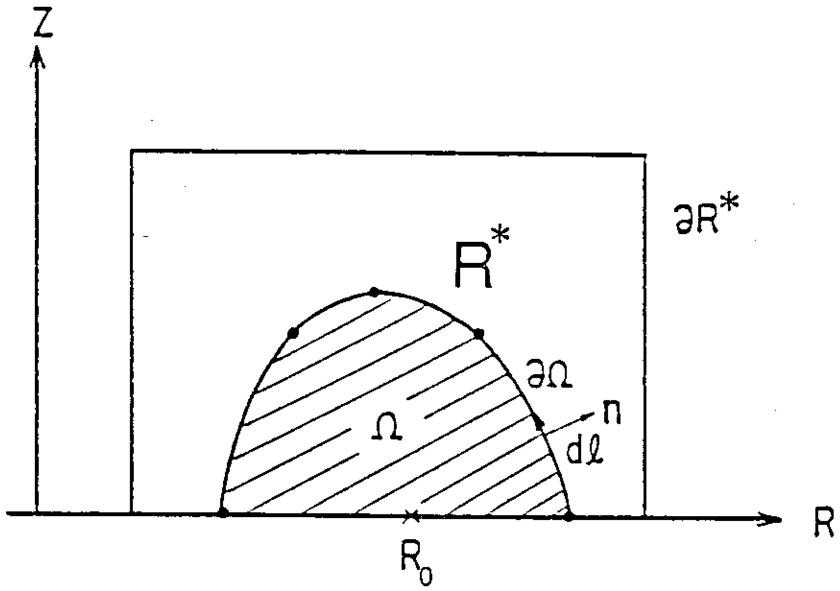

Fig. 1 Rectangular domain for Grad-Shafranov equation.
Boundary condition is given on ∂R^*

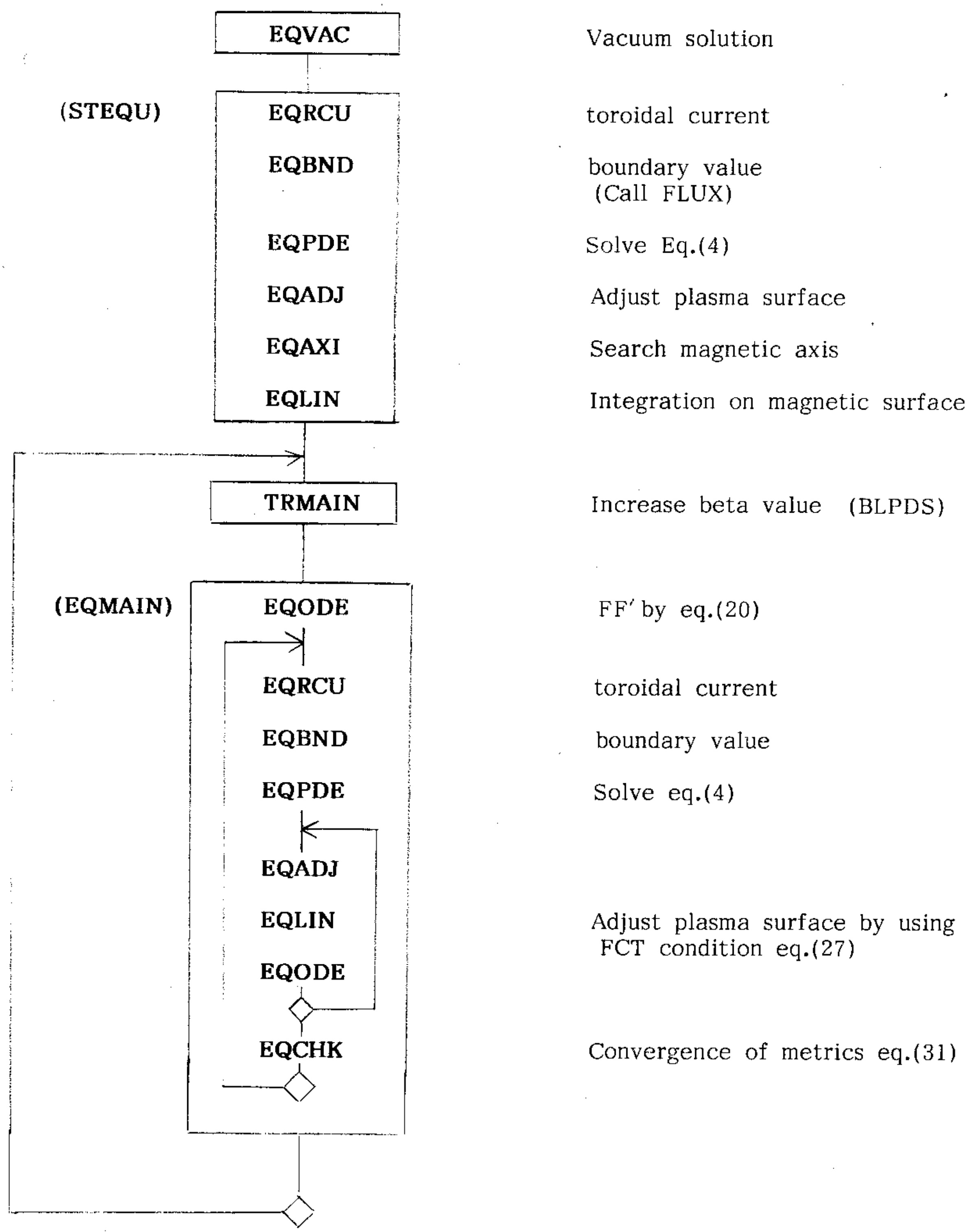

Fig.2 Flow of SELENE Code

```
C==========================================================================00010000
       SUBROUTINE BLPDS(NB)                                              ISN=0001
       DPR=DPBL(NB)                                                      ISN=0015
       IU=0                                                              ISN=0016
       IL=0                                                              ISN=0017
       FAC=PBL(NB)                                                       ISN=0018
    10 CONTINUE                                                          ISN=0019
       IS=1                                                              ISN=0020
       JS=1                                                              ISN=0021
       S=0.                                                              ISN=0022
       CZ=1./COE1(1)                                                     ISN=0023
       CY=-FAC*COE5(1)                                                   ISN=0024
       Z=1.                                                              ISN=0025
       Y=0.                                                              ISN=0026
1------DO 20 J=2,KSMAX                                                   ISN=0027
1      ZO=Z                                                              ISN=0028
1      YO=Y                                                              ISN=0029
1      CZO=CZ                                                            ISN=0030
1      CYO=CY                                                            ISN=0031
1      IS=IS+1                                                           ISN=0032
1      JS=JS+1                                                           ISN=0033
1      IF(IS.GT.ISMAX)IS=2                                               ISN=0034
1      S=S+DSS                                                           ISN=0035
1      PHI=COE7(IS)+AVCOE7*S                                             ISN=0036
1      PHI2=PHI*PHI                                                      ISN=0037
1      CZ=1./(COE1(IS)+COE2(IS)*PHI2)                                    ISN=0038
1      CY=FAC*(-COE5(IS)+COE6(IS)*PHI)                                   ISN=0039
1      DSH=0.5*DSS                                                       ISN=0040
1      DDS=DSH*DSH                                                       ISN=0041
1      D=1.-DDS*CZ*CY                                                    ISN=0042
1      Z=((1.+DDS*CZ*CYO)*ZO+DSH*(CZ+CZO)*YO)/D                          ISN=0043
1      Y=((1.+DDS*CY*CZO)*YO+DSH*(CY+CYO)*ZO)/D                          ISN=0044
1      IF(Z.LT. 0.)GOTO 40                                               ISN=0045
1      IF(Z.GT.10.)GOTO 30                                               ISN=0046
+---20 CONTINUE                                                          ISN=0047
    30 FU=FAC                                                            ISN=0048
       IU=1                                                              ISN=0049
       IF(IL.NE.0)GOTO 50                                                ISN=0050
       FAC=FAC+0.5                                                       ISN=0051
       IF(FAC.GT.100.)GOTO 60                                            ISN=0052
       GOTO 10                                                           ISN=0053
    40 FL=FAC                                                            ISN=0054
       IL=1                                                              ISN=0055
       IF(IU.NE.0)GOTO 50                                                ISN=0056
       FAC=FAC-0.1                                                       ISN=0057
       IF(FAC.LE.0.1)GOTO 60                                             ISN=0058
       GOTO 10                                                           ISN=0059
    50 ER=DABS((FL-FU)/(FL+FU))                                          ISN=0060
       FAC=0.5*(FL+FU)                                                   ISN=0061
       IF(ER.GT.EGBL)GOTO 10                                             ISN=0062
    60 PBL(NB)=FAC                                                       ISN=0063
       RETURN                                                            ISN=0064
       END                                                               ISN=0065
```

Fig.3 Original version of BLPDS

```
              C======================================================================00020000
                    SUBROUTINE BLPDS(NB)                                             ISN=0001
              C     V-LENGTH                                                         00162010
                    LENGT=7                                                          ISN=0023
              C                                                                      00164010
                    IVMAX = NB                                                       ISN=0024
1------------V------DO 100 I=1, IVMAX                                                ISN=0025
1            V         IVL(I) = I                                                    ISN=0026
1            V         FAC(I) = PBL(I)                                               ISN=0027
1            V         IU(I) = 0                                                     ISN=0028
1            V         IL(I) = 0                                                     ISN=0029
+------------V--100 CONTINUE                                                         ISN=0030

              C                                                                      00240003
1------------V-2500 DO 200 K=2, IVMAX                                                ISN=0031
1            V         ID=IVL(K)                                                     ISN=0032
1            V         CZ(ID) = 1./COE1(1,ID)                                        ISN=0033
1            V         CY(ID)=-FAC(ID)*COE5(1,ID)                                    ISN=0034
1            V         Z(ID)=1.0                                                     ISN=0035
1            V         Y(ID)=0.                                                      ISN=0036
+------------V--200 CONTINUE                                                         ISN=0037

                    IV¥MAX=IVMAX                                                     ISN=0038
1------------V------DO 250 I3=2, IV¥MAX                                              ISN=0039
1            V         IVL¥(I3)=IVL(I3)                                              ISN=0040
+------------V--250 CONTINUE                                                         ISN=0041

                    LC=1                                                             ISN=0042
              C                                                                      00370003
1------------V------DO 350 I3=2,IV¥MAX                                               ISN=0043
1            V         ID=IVL¥(I3)                                                   ISN=0044
1            V         IS(ID) =1                                                     ISN=0045
1            V         S(ID)=0                                                       ISN=0046
+------------V--350 CONTINUE                                                         ISN=0047

              C                                                                      00430008
1-------------------DO 3000 J=2, KSMAX                                               ISN=0048
1                      LC1=1                                                         ISN=0049
1                      LCA=1                                                         ISN=0050
1                      LCB=1                                                         ISN=0051
1 2-----------------IF(IV¥MAX .GE. LENGT) THEN      ------- judement of vector length ISN=0052
1 2           *VOCL LOOP,NOVREC                                                      00490003
1 2 3--------V------DO 1000 I=2, IV¥MAX                                              ISN=0053
1 2 3        V         ID=IVL¥(I)                                                    ISN=0054
1 2 3         C                                                                      00520003
1 2 3        V         IS(ID)=IS(ID)+1                                               ISN=0055
1 2 3        V         IF(IS(ID).GT.ISMAX) IS(ID)=2                                  ISN=0056
1 2 3        V         ZO=Z(ID)                                                      ISN=0057
1 2 3        V         YO=Y(ID)                                                      ISN=0058
1 2 3        V         CZO=CZ(ID)                                                    ISN=0059
1 2 3        V         CYO=CY(ID)                                                    ISN=0060
1 2 3        V         S(ID) = S(ID)+DSS1(ID)                                        ISN=0061
1 2 3        V         PHI=COE7(IS(ID),ID)+AVCOE7(ID)*S(ID)                          ISN=0062
1 2 3        V         PHI2=PHI*PHI                                                  ISN=0063
1 2 3        V         CZ(ID)=1./(COE1(IS(ID),ID)+COE2(IS(ID),ID)*PHI2)              ISN=0064
1 2 3        V         CY(ID)=FAC(ID)*(-COE5(IS(ID),ID)+COE6(IS(ID),ID)*PHI)         ISN=0065
1 2 3        V         DSH=0.5*DSS1(ID)                                              ISN=0066
1 2 3        V         DDS=DSH*DSH                                                   ISN=0067
1 2 3        V         D=1.-DDS*CZ(ID)*CY(ID)                                        ISN=0068
1 2 3        V          Z(ID)=((1.+DDS*CZ(ID)*CYO)*ZO+                               ISN=0069
1 2 3        V     R           DSH*(CZ(ID)+CZO)*YO)/D                                00680003
1 2 3        V          Y(ID)=((1.+DDS*CY(ID)*CZO)*YO+                               ISN=0070
1 2 3        V     R           DSH*(CY(ID)+CYO)*ZO)/D                                00700003
```

Fig.4 Vectorized version of BLPDS

```
1 2 3 4------V-----------IF(Z(ID).LT.0.)  THEN         -------- for unstable case          ISN=0071
1 2 3 4       V              LCA=LCA+1                                                     ISN=0072
1 2 3 4       V              LL1(LCA)=ID                                                   ISN=0073
1 2 3 +------V-----------ELSE                                                              ISN=0074
.1 2 3 4 5----V-------------IF(Z(ID).GT.10.0 .OR. J .EQ. KSMAX) THEN                       ISN=0075
1 2 3 4 5     V                LCB=LCB+1              ------- for stable case              ISN=0076
1 2 3 4 5     V                LL2(LCB)=ID                                                 ISN=0077
1 2 3 4 +----V-------------ELSE                                                            ISN=0078
1 2 3 4 5 6--V---------------IF(J .LT. KSMAX) THEN    ---- for futher steps                ISN=0079
1 2 3 4 5 6   V                  LC1 = LC1+1                (see DO 1050)                  ISN=0080
1 2 3 4 5 6   V                  L2(LC1)=ID                                                ISN=0081
1 2 3 4 5 +--V---------------END IF                                                        ISN=0082
1 2 3 4 +----V-------------END IF                                                          ISN=0083
1 2 3 +------V-----------END IF                                                            ISN=0084
1 2 +--------V-1000 CONTINUE                                                               ISN=0085
1 2
1 2
1 +------------------ELSE                                                                  ISN=0086
1 2           *VOCL LOOP,SCALAR                                                            00870003
1 2 3---------------DO 1001 I=2, IV¥MAX                                                    ISN=0087
1 2 3                   ID=IVL¥(I)                                                         ISN=0088
1 2 3         C                                                                            00900003
1 2 3                   IS(ID)=IS(ID)+1                                                    ISN=0089
1 2 3                   IF(IS(ID).GT.ISMAX) IS(ID)=2                                       ISN=0090
1 2 3                   Z0=Z(ID)                                                           ISN=0091
1 2 3                   Y0=Y(ID)                                                           ISN=0092
1 2 3                   CZ0=CZ(ID)                                                         ISN=0093
1 2 3                   CY0=CY(ID)                                                         ISN=0094
1 2 3                   S(ID) = S(ID)+DSS1(ID)                                             ISN=0095
1 2 3                   PHI=COE7(IS(ID),ID)+AVCOE7(ID)*S(ID)                               ISN=0096
1 2 3                   PHI2=PHI*PHI                                                       ISN=0097
1 2 3                   CZ(ID)=1./(COE1(IS(ID),ID)+COE2(IS(ID),ID)*PHI2)                   ISN=0098
1 2 3                   CY(ID)=FAC(ID)*(-COE5(IS(ID),ID)+COE6(IS(ID),ID)*PHI)              ISN=0099
1 2 3                   DSH=0.5*DSS1(ID)                                                   ISN=0100
1 2 3                   DDS=DSH*DSH                                                        ISN=0101
1 2 3                   D=1.-DDS*CZ(ID)*CY(ID)                                             ISN=0102
1 2 3                    Z(ID)=((1.+DDS*CZ(ID)*CY0)*Z0+                                    ISN=0103
1 2 3             R               DSH*(CZ(ID)+CZ0)*Y0)/D                                   01060003
1 2 3                    Y(ID)=((1.+DDS*CY(ID)*CZ0)*Y0+                                    ISN=0104
1 2 3             R               DSH*(CY(ID)+CY0)*Z0)/D                                   01080003
1 2 3 4-----------------IF(Z(ID).LT.0.)  THEN                                              ISN=0105
1 2 3 4                    LCA=LCA+1                                                       ISN=0106
1 2 3 4                    LL1(LCA)=ID                                                     ISN=0107
1 2 3 +-----------------ELSE                                                               ISN=0108
1 2 3 4 5-----------------IF(Z(ID).GT.10.0 .OR. J.EQ.KSMAX) THEN                           ISN=0109
1 2 3 4 5                    LCB=LCB+1                                                     ISN=0110
1 2 3 4 5                    LL2(LCB)=ID                                                   ISN=0111
1 2 3 4 +-----------------ELSE                                                             ISN=0112
1 2 3 4 5 6-----------------IF(J .LT. KSMAX) THEN                                          ISN=0113
1 2 3 4 5 6                    LC1 = LC1+1                                                 ISN=0114
1 2 3 4 5 6                    L2(LC1)=ID                                                  ISN=0115
1 2 3 4 5 +-----------------END IF                                                         ISN=0116
1 2 3 4 +-----------------END IF                                                           ISN=0117
1 2 3 +-----------------END IF                                                             ISN=0118
1 2 +----------1001 CONTINUE                                                               ISN=0119
1 2
1 2
1 +-----------------END IF                                                                 ISN=0120
1             C                                                                            01250003
1             *VOCL LOOP,SCALAR                                                            01260003
1 2-------------------DO 1008 I=2,LCA                                                      ISN=0121
1 2                     ID=LL1(I)                                                          ISN=0122
1 2                     FL(ID)=FAC(ID)                                                     ISN=0123
1 2                     IL(ID)=1                                                           ISN=0124
1 2                     IF(IU(ID).NE.0) GO TO 50                                           ISN=0125
1 2                     FAC(ID)=FAC(ID)-0.1                                                ISN=0126
1 2                     IF(FAC(ID).LE.0.1) GO TO 1008                                      ISN=0127
```

```
1 2                    GO TO 110                                                  ISN=0128
1 2          50        ER=DABS((FL(ID)-FU(ID))/(FL(ID)+FU(ID)))                   ISN=0129
1 2                    FAC(ID)=0.5*(FL(ID)+FU(ID))                                ISN=0130
1 2                    IF(ER.GT.EGBL) GO TO 110                                   ISN=0131
1 2                    GO TO 1008                                                 ISN=0132
1 2         110        LC=LC+1                                                    ISN=0133

1 2                    L1(LC)=ID                                                  ISN=0134
1 +------------1008   CONTINUE                                                    ISN=0135
1
1
1            C                                                                    01420003
1            *VOCL LOOP,SCALAR                                                    01430003
1 2------------------DO 1011 I=2,LCB                                              ISN=0136
1 2                     ID=LL2(I)                                                 ISN=0137
1 2                    FU(ID)=FAC(ID)                                             ISN=0138
1 2                    IU(ID)=1                                                   ISN=0139
1 2                    IF(IL(ID).NE.0) GO TO 51                                   ISN=0140
1 2                    FAC(ID)=FAC(ID)+0.5                                        ISN=0141
1 2                    IF(FAC(ID).GT.100.) GO TO 1011                             ISN=0142
1 2                    GO TO 111                                                  ISN=0143
1 2          51        ER=DABS((FL(ID)-FU(ID))/(FL(ID)+FU(ID)))                   ISN=0144
1 2                    FAC(ID)=0.5*(FL(ID)+FU(ID))                                ISN=0145
1 2                    IF(ER.GT.EGBL) GO TO 111                                   ISN=0146
1 2                    GO TO 1011                                                 ISN=0147
1 2         111        LC=LC+1                                                    ISN=0148
1 2                    L1(LC)=ID                                                  ISN=0149
1 +------------1011   CONTINUE                                                    ISN=0150
1
1
1            C                                                                    01590009
1                      IF(LC1 .EQ. 1) GO TO 3500                                  ISN=0151
1                        IV¥MAX=LC1                                               ISN=0152
1 2----------V------------DO 1050 I2=2, LC1      ------ construction of list vector   ISN=0153
1 2          V              IVL¥(I2)=L2(I2)                                       ISN=0154
1 +----------V-1050        CONTINUE                                               ISN=0155
1
1
+-------------3000 CONTINUE                                                       ISN=0156


             C                                                                    01930003
1-------------3500 IF(LC.GT.1) THEN                                               ISN=0157
1 2----------V--------DO 700 I=2, LC                                              ISN=0158
1 2          V            IVL(I) = L1(I)                                          ISN=0159
1 +----------V--700    CONTINUE                                                   ISN=0160
1
1
1                     IVMAX= LC                                                   ISN=0161
1                     GO TO 2500                                                  ISN=0162
+------------------END IF                                                         ISN=0163
             C                                                                    02010003
1------------V------DO 800 I=2, NB                                                ISN=0164
1            V        PBL(I) = FAC(I)                                             ISN=0165
1            V        IF(DGBL.LE.O.AND.PBL(I).LT.1) GBL(I)=1.                     ISN=0166
+------------V--800 CONTINUE                                                      ISN=0167

                                                                                  02060003
             C                                                                    02070003
                   RETURN                                                         ISN=0168
                   END                                                            ISN=0169
```

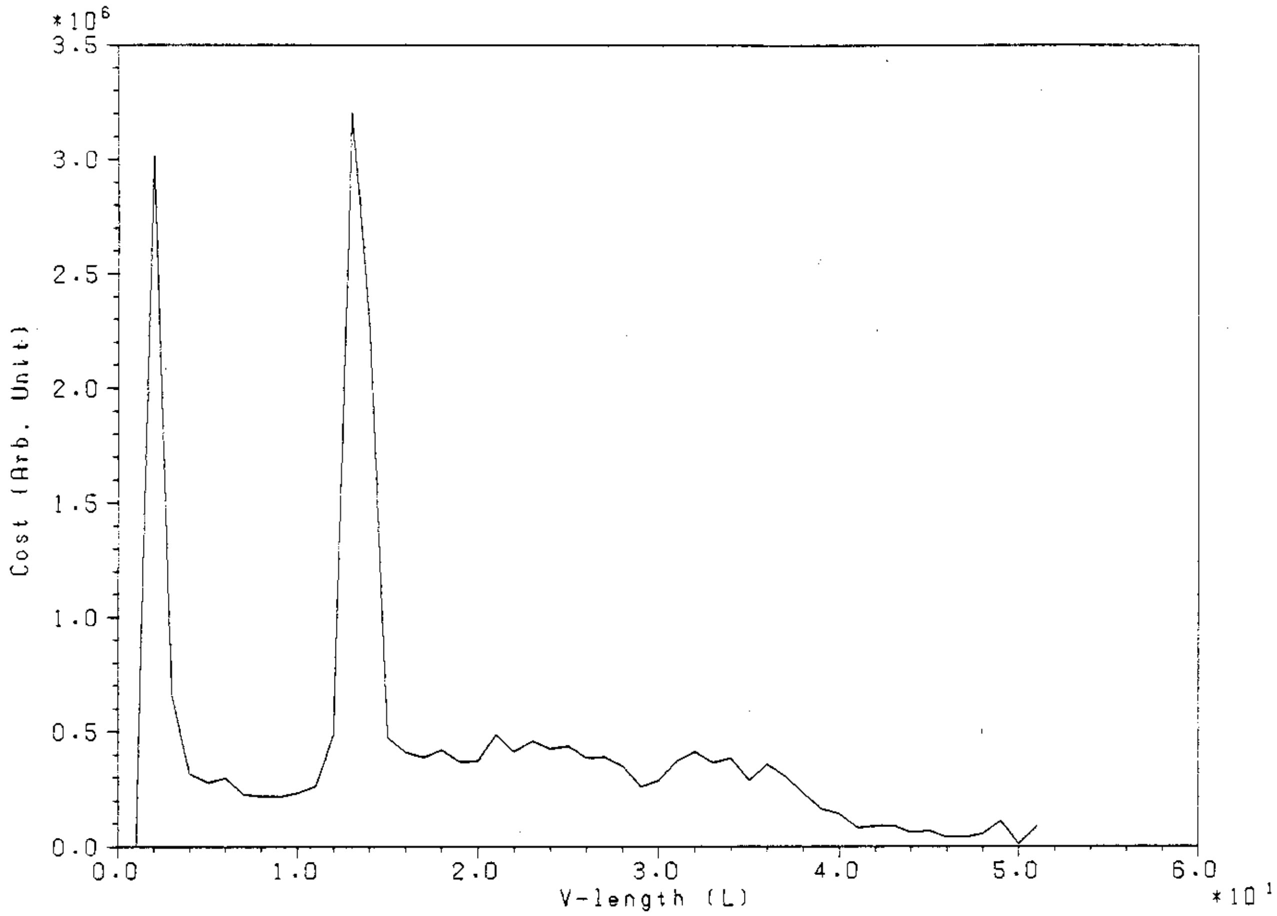

Fig.5 Computational cost (arb. unit) vs. vector length in BLPDS

```
              C========================================================================00020000
                     FUNCTION FLUX(R,Z,RO,ZO,CO,N)                                     ISN=0001
                     FLUX=0.                                                           ISN=0005
1------------S------DO 10 M=1,N                                                        ISN=0006
1            S       X=R*RO(M)                                                         ISN=0007
1            S       XX=4.*X/((R+RO(M))**2+(Z-ZO(M))**2)                               ISN=0008
1            S       I=1.-DTAB*DLOG(1.-XX)                                             ISN=0009
1            S       IF(I.GT.NTAB)GOTO 20                                              ISN=0010
1            S       D=(XX-XTAB(I))/(XTAB(I+1)-XTAB(I))                                ISN=0011
1            S       FLUX=FLUX-CO(M)*DSQRT(X)*(FTAB(I)+D*(FTAB(I+1)-FTAB(I)))          ISN=0012
1            S       GOTO 10                                                           ISN=0013
1               20   CONTINUE                                                          ISN=0014
1            S       FLUX=FLUX-CO(M)*DSQRT(X)*FLUXO(XX)                                ISN=0015
+---------------10   CONTINUE                                                          ISN=0016

                     RETURN                                                            ISN=0017
                     END                                                               ISN=0018
```

Fig.6 Original version of FLUX

```
             C=================================================================00020000
                   SUBROUTINE FLUX(R,Z,RO,ZO,CO,N,IM,NR,IS,IE)                    ISN=0001
                   IMM=IM                                                         ISN=0030
                   NRR=NR                                                         ISN=0031
1------------V------DO 600 K=IS,IE                                                ISN=0032
1            V      IMM=IMM+NRR                                                   ISN=0033
1            V      PSI(IMM)=0.DO                                                 ISN=0034
+------------V--600 CONTINUE                                                      ISN=0035


             CC                                                                   00520026
1------------S------DO 10 K=IS,IE                                                 ISN=0036
1                   IMM = IM + NRR*(K-IS+1)                                       ISN=0037
1            S      IFLAG=0                                                       ISN=0038
1            *VOCL LOOP,NOVREC                                                    00560027
1 2----------V------DO 100 M=1,N                                                  ISN=0039
1 2          V      X(M)=R(K)*RO(M)                                               ISN=0040
1 2          V      XX(M)=4.*X(M)/((R(K)+RO(M))**2+(Z(K)-ZO(M))**2)               ISN=0041
1 2          V      I(M)=1.-DTAB*DLOG(1.-XX(M))                                   ISN=0042
1 2          V      IF(I(M).GT.NTAB) IFLAG=1                                      ISN=0043
1 +----------V-100  CONTINUE                                                      ISN=0044
1
1
1            C                                                                    00630026
1 2----------S------IF(IFLAG.EQ.1) THEN                                           ISN=0045
1 2          *VOCL LOOP,SCALAR                                                    00650026
1 2 3-----------------DO 101 M=1,N                                                ISN=0046
1 2 3 4-----------------IF(I(M).GT.NTAB) THEN                                     ISN=0047
1 2 3 4                    XXF=1.-XX(M)                                           ISN=0048
1 2 3 4                    XL=DLOG(1./XXF)                                        ISN=0049
1 2 3 4                    XK=AO+XXF*(A1+XXF*A2)+(BO+XXF*(B1+XXF*B2))*XL          ISN=0050
1 2 3 4                    XE=COF+XXF*(C1+XXF*C2)+     XXF*(D1+XXF*D2) *XL        ISN=0051
1 2 3 4                    FLUXOF=((1.-XX(M)/2.)*XK-XE)/(PI*DSQRT(XX(M)))         ISN=0052
1 2 3 4      C++           PSI(IMM)=PSI(IMM)-CO(M)*DSQRT(X(M))*FLUXO(XX(M))       00730029
1 2 3 4                    PSI(IMM)=PSI(IMM)-CO(M)*DSQRT(X(M))*FLUXOF             ISN=0053
1 2 3 +-----------------END IF                                                    ISN=0054
1 2 +----------101      CONTINUE                                                  ISN=0055
1 2
1 2
1 2          *VOCL LOOP,NOVREC                                                    00760132
1 2 3--------V--------DO 102 M=1,N                                                ISN=0056
1 2 3 4------V----------IF(I(M).LE.NTAB) THEN                                     ISN=0057
1 2 3 4      V             D=(XX(M)-XTAB(I(M)))/(XTAB(I(M)+1)-XTAB(I(M)))         ISN=0058
1 2 3 4      V             PSI(IMM)=PSI(IMM)-CO(M)*DSQRT(X(M))                    ISN=0059
1 2 3 4      V     R                 *(FTAB(I(M))+D*(FTAB(I(M)+1)-FTAB(I(M))))    00760632
1 2 3 +------V----------END IF                                                    ISN=0060
1 2 +--------V-102  CONTINUE                                                      ISN=0061
1 2
1 2
1 +----------S------ELSE                                                          ISN=0062
1 2          *VOCL LOOP,NOVREC                                                    00761028
1 2 3--------V--------DO 103 M=1,N                                                ISN=0063
1 2 3        V             D=(XX(M)-XTAB(I(M)))/(XTAB(I(M)+1)-XTAB(I(M)))         ISN=0064
1 2 3        V             PSI(IMM)=PSI(IMM)-CO(M)*DSQRT(X(M))                    ISN=0065
1 2 3        V     R                 *(FTAB(I(M))+D*(FTAB(I(M)+1)-FTAB(I(M))))    00766028
1 2 +--------V-103  CONTINUE                                                      ISN=0066
1 2
1 2
1 +----------S------END IF                                                        ISN=0067
1            C                                                                    00780026
+------------S--10  CONTINUE                                                      ISN=0068

                   RETURN                                                         ISN=0069
                   END                                                            ISN=0070
```

Fig.7 Vectorized version of FLUX

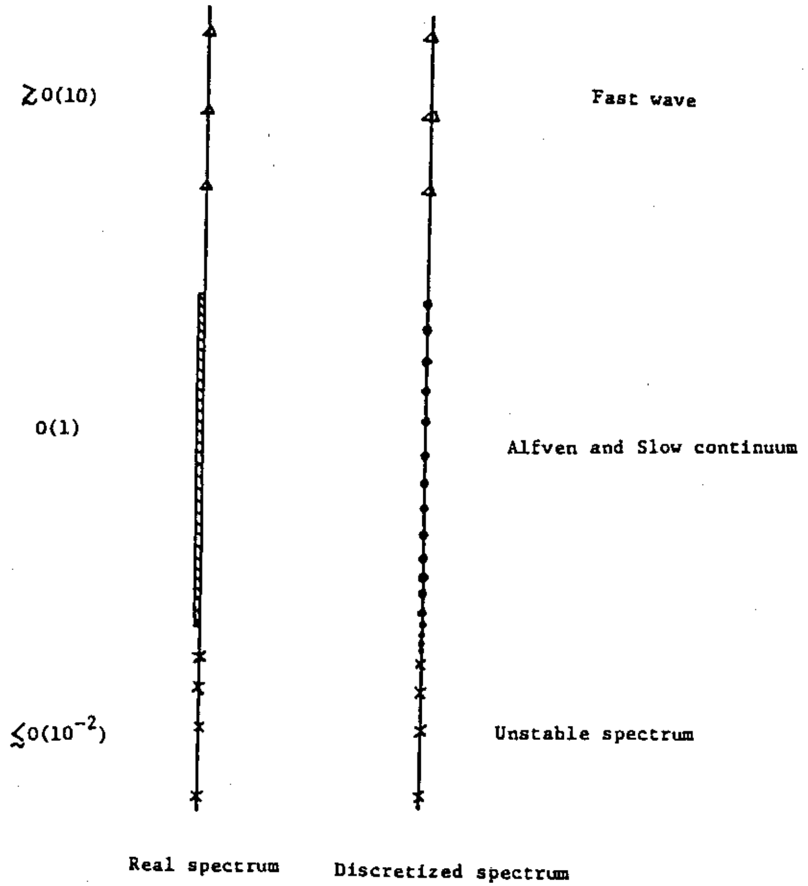

Fig. 8 Schematic diagram of the MHD spectrum and the discretized one. The spectrum is normalized by $B_0/(R_0\sqrt{\mu_0\rho_0})$.

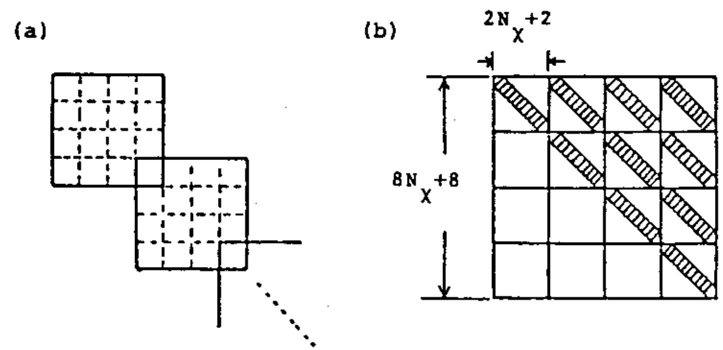

Fig. 9 Structures of (a) the matrix and (b) the block. The overlapping part in (a) corresponds to the variable X. Each subblock in (b) consists of the band matrices of width 7.

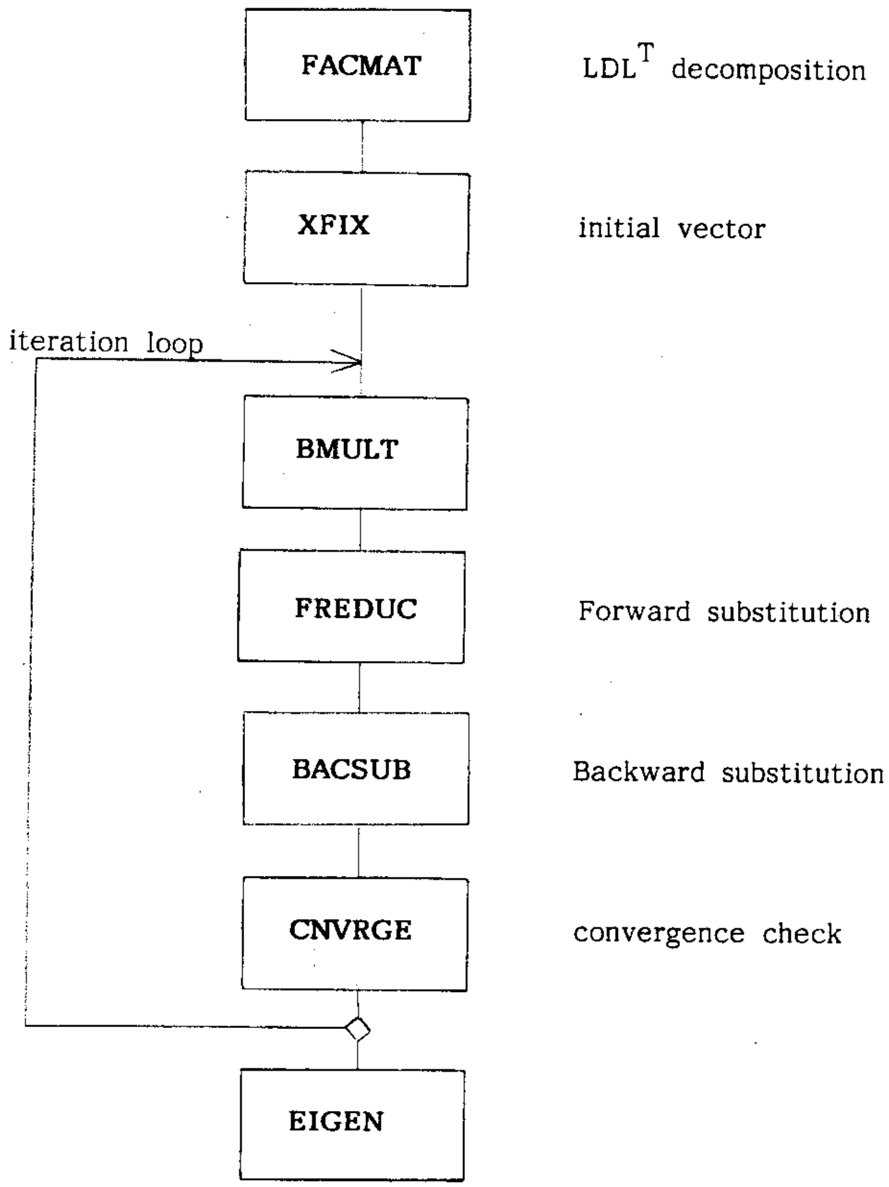

Fig.10 Flow of ERATO4

```
MAIN  -----EIGVAL-----SET3
                 +--VEKIT -----INFORM----*TIME
                           I         +-*CLOCKM
                           I         +-*DATE
                           +--IODSK4
                           +--FACMAT-----RDMAT -----IODSK4
                           I         +--FACBND----*DABS
                           I         +--FIXSQ
                           I         +--ALBCON-----CONCOL----*MAXO
                           I         I         I         +-*MINO
                           I         I         +--LBDDSL----*MINO
                           I         I         +--UBDSOL----*MINO
                           I         I         +--ODTMLT
                           I         +--CALD  ----*DABS
                           I         +--CACA2 -----LTRDSL
                           I         I         +--UTRSOL
                           I         +--PUTMAT
                           I         +--IODSK4
                           +--XFIX  -----BACSUB-----GETMAT
                           I         I         +--UTRSOL
                           I         I         +--OFDMLT
                           I         I         +--LBDDSL----*MINO
                           I         I         +--UBDSOL----*MINO
                           I         I         +--ODTMLT
                           I         I         +--OFD2MT
                           I         I         +--LTRDSL
                           I         +-*DSQRT
                           I         +--CNVRGE----*DABS
                           +--BMULT -----BBMULT-----IODSK4
                           I                   +--DBKMLT
                           I                   +--BLKMLT
                           I                   +--BKTMLT
                           +--FREDUC-----GETMAT
                           I         +--LBDDSL----*MINO
                           I         +--UBDSOL----*MINO
                           I         +--ODTMLT
                           I         +--LTRDSL
                           I         +--UTRSOL
                           I         +--OD2TMT
                           I         +--OFDMLT
                           +--BACSUB-----GETMAT
                           I         +--UTRSOL
                           I         +--OFDMLT
                           I         +--LBDDSL----*MINO
                           I         +--UBDSOL----*MINO
                           I         +--ODTMLT
                           I         +--OFD2MT
                           I         +--LTRDSL
                           +-*DSQRT
                           +--CNVRGE----*DABS
                           +--EIGEN -----BBMULT-----IODSK4
                                     I         +--DBKMLT
                                     I         +--BLKMLT
                                     I         +--BKTMLT
                                     +--IODSK4
```

Fig.11　Tree struncture of ERATO4

(a)

```
              SUBROUTINE SAXPY(N, A, X, NX, Y, NY)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSION X(1), Y(1)
        C
        C THIS SUBROUTINE COMPUTES Y = Y + A*X.
        C
              IF (A.EQ.0.0D0 .OR. N.LE.0) RETURN
              Y(1) = Y(1) + A*X(1)
              IF (N.EQ.1) RETURN
              NM1 = N - 1
1------------S-------DO 10 I=1,NM1
1            S            Y(I*NY+1) = Y(I*NY+1) + A*X(I*NX+1)
+----------------10 CONTINUE

              RETURN
              END
```

(b)

```
              FUNCTION SDOT(N, X, NX, Y, NY)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSION X(1), Y(1)
        C THIS FUNCTION COMPUTES THE INNER PRODUCT OF X AND Y.
        C
              SDOT = 0.0D0
              IF (N.LE.0) RETURN
              SDOT = X(1)*Y(1)
              IF (N.EQ.1) RETURN
              NM1 = N - 1
1------------V-------DO 10 I=1,NM1
1            V            SDOT = SDOT + X(I*NX+1)*Y(I*NY+1)
+------------V---10 CONTINUE

              RETURN
              END
```

(c)

```
              SUBROUTINE SXYPZ(N, X, NX, Y, NY, Z, NZ)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSION X(1), Y(1), Z(1)
        C
        C THIS SUBROUTINE RETURNS Z = Z + X*Y  (ELEMENTWISE).
        C
              IF (N.LE.0) RETURN
              Z(1) = Z(1) + X(1)*Y(1)
              IF (N.EQ.1) RETURN
              NM1 = N - 1
1------------S-------DO 10 I=1,NM1
1            S            Z(I*NZ+1) = Z(I*NZ+1) + X(I*NX+1)*Y(I*NY+1)
+----------------10 CONTINUE

              RETURN
              END
```

Fig.12 Subroutines for vector arithmetics (a) SAXPY, (b) SDOT and (c) SXYPZ.

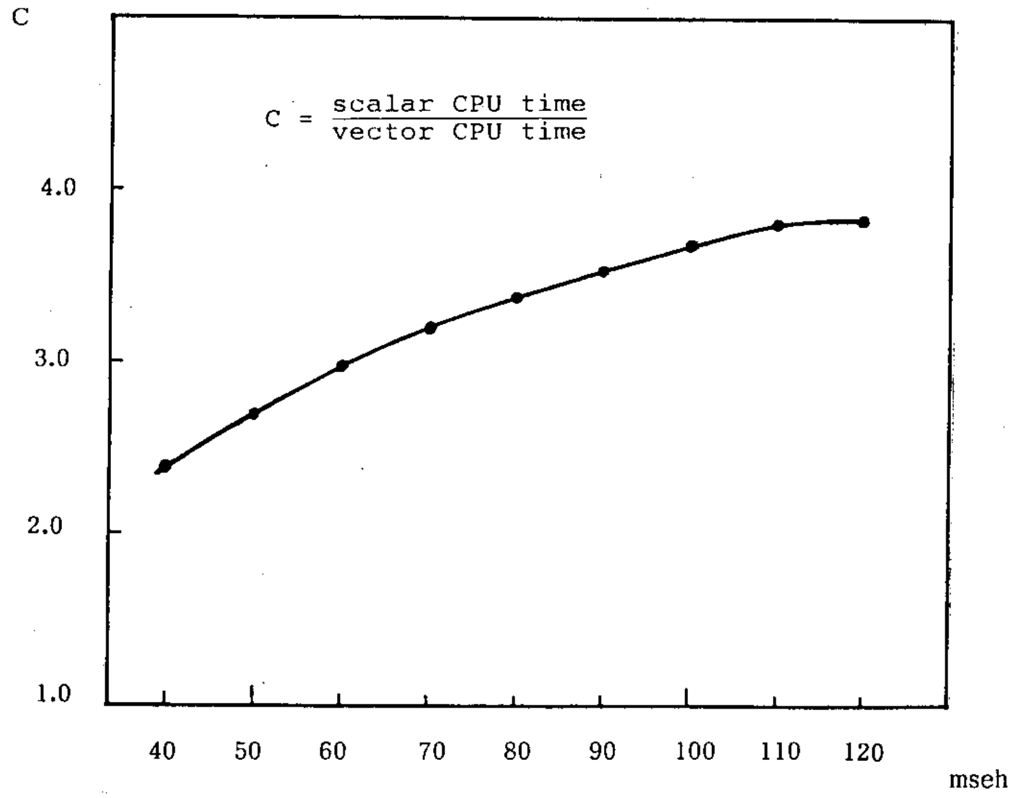

Fig.13 The ratio of CPU time in scalar and vector calculation vs. mesh numbers for ERATO4