JAERI-M
8616

STABILITY ANALYSIS BY ERATO CODE

December 1979

Toshihide TSUNEMATSU, Tatsuoki TAKEDA,
Toshihiko MATSUURA*, Masafumi AZUMI,
Gen-ichi KURITA and Tomonori TAKIZUKA

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

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

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

Stability Analysis by ERATO Code

Toshihide TSUNEMATSU, Tatsuoki TAKEDA, Toshihiko MATSUURA,*
Masafumi AZUMI, Gen-ichi KURITA and Tomonori TAKIZUKA
Division of Thermonuclear Fusion Research
Tokai Research Establishment,
JAERI

(Received November 27, 1979)

Problems in MHD stability calculations by ERATO code are described; which concern convergence property of results, equilibrium codes, and machine optimization of ERATO code.

It is concluded that irregularity on a convergence curve is not dut to a fault of the ERATO code itself but due to inappropriate choice of the equilibrium calculation meshes. Also described are a code to calculate an equilibrium as a quasi-inverse problem and a code to calculate an equilibrium as a result of a transport process. Optimization of the code with respect to I/O operations reduced both CPU time and I/O time considerably. With the FACOM230-75 APU/CPU multiprocessor system, the performance is about 6 times as high as with the FACOM230-75 CPU, showing the effectiveness of a vector processing computer for the kind of MHD computations. This report is a summary of the material presented at the ERATO workshop 1979(ORNL), supplemented with some details.

Keywords: MHD Stability Analysis, MHD Equilibrium, ERATO Code, Tokamak Code, Computational Physics, Vector-processing Computer, Convergence, Optimization.

* On leave from Fujitsu Co. Ltd., Ohta-ku, Tokyo Japan

ERATOコードによる電磁流体力学的安定性解析

日本原子力研究所東海研究所核融合研究部
常松 俊秀・竹田 辰興・松浦 俊彦*・安積 正史
栗田 源一・滝塚 知典

（1979年11月27日受理）

本報告書は，ERATOコードを使ってMHD安定性の解析を行う際に生ずる諸問題およびERATOに関連する種々の計算コード，さらにERATOの計算機における最適化についてまとめたものである。

第1の問題については，特に，平衡計算の精度にかかわる諸問題についてまとめた。関連コードの章では，逆平衡計算コードと輸送計算から平衡を得るコードを中心に解説している。また，計算機における最適化では，FACOM230－75での入出力処理の最適化，75APU／CPUシステムでの高速化について実験した結果を示した。

* 外来研究員：富士通㈱

Contents

目　　次

1. Introduction

Since the ERATO code[1)] was adapted for the FACOM230-75 computer system of JAERI in March 1978, MHD stability of several tokamaks and problems on ERATO itself are being studied intensively. As well-known the ERATO code analyzes linearized ideal MHD stability by solving a large-scale general eigenvalue problem of matrices which are derived from Lagrangian of an axisymmetric toroidal plasma by using the finite element formulation. Some basic properties of the method were remarked and standard procedure to carry out a series of stability analyses was proposed by the original author of the code[1)]. There are, however, many problems to be solved if one is to carry out detailed analyses under various conditions.

Among the problems the most basic and important one is concerning convergence property of a solution. In some cases required number of computation meshes for beta optimization analyses is larger than that used previously by Gruber et al.[2)] Moreover, the convergence property of the stability calculation seems to depend on mesh number of equilibrium calculation.

Concerning choice of classes of equilibria problems on an interface between the equilibrium code and stability code are also important. Especially, analyses of stabilities of equilibria which are reached by tracing realistic transport process are desired. Stability criterion of a high toroidal mode number instability had better be checked in the interface routine as it is rather difficult to calculate this mode in the ERATO code.

Improvement of ERATO or extension of function of the code are interesting. Among them stability analyses of equilibria with external poloidal field coils are urgently necessary to investigate positional instability of the JT-60 tokamak. But this problem is outisde of the scope of this report.

From the viewpoint of cost of stability analyses, optimization of the code with respect to an individual computer system is necessary. As for this problem the following improvements or studies were carried out.

(1) Improvement of I/O routine (system program).

(2) Optimization of the code for FACOM230-75 APU (Array Processing Unit).

(3) Test runs of the code in various computer systems, (FACOM230-75,

230-75 APU, M-200; HITAC M-180, M-180 with IAP (Integrated Array Processor); IBM 360/168, CDC 7600; Data of the test runs will be reported elsewhere).

In section 2 the convergence property of the ERATO calculation is described with a recomendable procedure to determine a converged growth rate. Concerning the interface between the equilibrium and stability codes, the equilibrium code SELENE, high-n ballooning stability code BOREAS and 2D tokamak code APOLLO which gives ballooning stable equilibria are described in sections 3, 4, and 5, respectively. Section 6 describes several topics on the optimization of the ERATO code with respect to a computer system. Optimization of the I/O routine of the FACOM 230-75 computer and optimization of the ERATO code for the APU are main topics of this section. Section 7 is devoted for summary and discussions.

2. Convergence Study

2.1 Procedure of the convergence study

First we summarize the procedure of the convergence study briefly. As for the relation between N_ψ and N_χ, we chose same number both for N_ψ and for N_χ. In ref.1, $N_\chi = N_\psi + 1$ is recommended but in our calculations ($N_\psi = N_\chi$) cases gave better or almost same convergence properties as compared with ($N_\chi = N_\psi + 1$) cases. Anyway the differences were slight. Usually we started calculations from $N = N_\psi = N_\chi = 25$ case. And the next calculation is carried out by increasing the mesh number N by one and using the growth rate of the last step as the guess value of the next step. Finally growth rates for 14 different mesh numbers are computed (N = 25, 26, -----, 36, 37)(Fig.1). Necessary number of data points for a complete convergence curve seems to depend on classes of equilibria. Actually such a large number of data points as 14 is not always necessary. Often it is possible to save the cost of computation by reducing the number of data points if one knows well the convergence property of the solution for the class of equilibria. We analyzed the stability limit of the JT-60 plasma for $\beta_p \approx 1.8$ by calculating growth rates for only 5 points (N = 25,26,27,28,34) and obtained satisfactory results (Fig.2).

230-75 APU, M-200; HITAC M-180, M-180 with IAP (Integrated Array Processor); IBM 360/168, CDC 7600; Data of the test runs will be reported elsewhere).

In section 2 the convergence property of the ERATO calculation is described with a recomendable procedure to determine a converged growth rate. Concerning the interface between the equilibrium and stability codes, the equilibrium code SELENE, high-n ballooning stability code BOREAS and 2D tokamak code APOLLO which gives ballooning stable equilibria are described in sections 3, 4, and 5, respectively. Section 6 describes several topics on the optimization of the ERATO code with respect to a computer system. Optimization of the I/O routine of the FACOM 230-75 computer and optimization of the ERATO code for the APU are main topics of this section. Section 7 is devoted for summary and discussions.

2. Convergence Study

2.1 Procedure of the convergence study

First we summarize the procedure of the convergence study briefly. As for the relation between N_ψ and N_χ, we chose same number both for N_ψ and for N_χ. In ref.1, $N_\chi = N_\psi + 1$ is recommended but in our calculations ($N_\psi = N_\chi$) cases gave better or almost same convergence properties as compared with ($N_\chi = N_\psi + 1$) cases. Anyway the differences were slight. Usually we started calculations from $N = N_\psi = N_\chi = 25$ case. And the next calculation is carried out by increasing the mesh number N by one and using the growth rate of the last step as the guess value of the next step. Finally growth rates for 14 different mesh numbers are computed (N = 25, 26, -----, 36, 37)(Fig.1). Necessary number of data points for a complete convergence curve seems to depend on classes of equilibria. Actually such a large number of data points as 14 is not always necessary. Often it is possible to save the cost of computation by reducing the number of data points if one knows well the convergence property of the solution for the class of equilibria. We analyzed the stability limit of the JT-60 plasma for $\beta_p \approx 1.8$ by calculating growth rates for only 5 points (N = 25,26,27,28,34) and obtained satisfactory results (Fig.2).

2.2 Problems on the convergence study

During the convergence studies for several series of stability calculations the following problems were found.

(1) Slight convexity of the convergence curves

We plotted the data points in the $\gamma^2 - 1/N^2$ plane, where γ is the instability growth rate. If the convergence is quadratic, the curve is expected to be a straight line. However, the data points are not on the straight line but on a slightly convex curve in this plane (see Fig.1).

(2) Convergence properties of low-growth-rate mode

It is rather difficult to determine a converged value of a small growth rate and there are several problems concerning low-growth-rate modes.

i) Computed growth rates of a fixed boundary mode differ considerably among results for different size of fitting area in the vicinity of the magnetic axis.

ii) The convergence property of the fixed boundary mode is not quadratic. The data points scatter considerably around the quadratic convergence line (Fig.3).

iii) Even if it is small the growth rate of a free boundary mode does not depend on the size of fitting area for $R_{ext} \gtrsim 1.15$ and the convergence is quadratic (Fig.4).

iv) When the shell is located very close to the plasma surface ($R_{ext} \lesssim 1.15$) the convergence of the free boundary mode is not quadratic and the extrapolated value ($R_{ext} \rightarrow 1.0$) from these data does not coincide with that of the fixed boundary mode.

(3) Resonance-like phenomenon observed on the convergence curve

Even when the data points are almost on a quadratic convergence line, sometimes, there appears a resonance-like phenomenon and data point for a certain mesh number differs considerably from the value which is evaluated from the convergence curve datermined by the other data points (see Fig.5). This phenomenon, sometimes, occurs at several mesh numbers simultaneously and in this case the convergence curve looks very irregular.

2.3 Explanation of the phenomenon or solution of the problems

As for the phenomenon or problems described in the previous subsection the following explanations or solutions are possible.

(1) Slight convexity of the convergence curves

The convergence property of results is very sensitive to the mesh number of the equilibrium calculation. When one chooses different mesh number of the equilibrium calculation for every data point on the convergence curve according to the "$N_r = 6N_\psi$" law, one could not attain a quadratic convergence. Quadratic convergence is realized only when the mesh number of the equilibrium calculation is fixed constant and that of the ERATO calculation is varied ($N_r \geq 6N_\psi$). The change of the convergence property due to the change of equilibrium mesh number is clearly seen in Fig.6.

Necessary mesh numbers for the equilibrium and stability calculations are dependent on classes of equilibria and also on an instability mode number. In a certain case ($n = 1$, $\beta_p \approx 1$, $\beta = 1.59$ %, $q_0 = 1.0$, $q_a = 3.34$) calculations with large mesh numbers ($N = 40,45,50$) were carried out by using the FACOM M-200 computer system installed at IPP Nagoya. In this case mesh numbers for the equilibrium calculation are; $N_r = 400$, $N_z = 200$. The computed data points are very well on the quadratic convergence curve which is extrapolated from the data points with $N_\psi \leq 37$ (Fig.7). Therefore, it may be safe to say that by using a suffisiently large equilibrium mesh number and varying the stability mesh number for a fixed equilibrium meshes a quadratic convergence could be attained for low mode number instabilities.

(2) Convergence properties of low-growth-rate modes

Problems i)- ii) described in 2.3 seem to indicate that non-quadratic behavior of the convergence of the low-growth-rate mode is attributed to the computational error due to inaccurately represented equilibrium quantities, especially, those near the magnetic axis. It seems rather difficult to avoid these problems by increasing the accuracy of the ERATO calculation itself. Therefore, it is recommendable to investigate the lower-growth-rate mode by extrapolation with respect to some parameters, such as the position of the shell, and so on.

On the other hand the problem iv) is due to the inaccuracy of evaluation of Green function in the vacuum region. This may be solved by improving the evaluation procedure of vacuum contribution but it has not been tried yet.

(3) Resonance-like phenomena observed on the convergence curve

The cause of the phenomena has not been cleared. Features of the phenomena are,

i) This phenomenon occurs only at the limited data points ($N = 29,30,35,37$ in the case of $\beta_p \approx 1.0$).

ii) This phenomenon is more remarkable for an equilibrium with a flat current distribution.

iii) For different equilibrium meshes the growth rates differs each other at the resonance mesh number (Fig.5).

We conjectured that this phenomenon was caused by a certain mismatch of the equilibrium and stability mesh numbers and we neglected the data points where the resonance-like phenomena occured. By neglecting these data points the other data points are usually on a quadratic convergence curve.

3. SELENE — Equilibrium codes

It is necessary to calculate the MHD equilibrium for the stability analyses with high accuracy near a magnetic axis and in the outer region of a torus where magnetic surfaces are closely located, because conventional numerical codes for equilibrium calculation are formulated on the basis of square meshes in a r-z poloidal plane but the stability calculation in the ERATO code is carried out on square meshes in a ψ-χ plane. To avoid this discrepancy we developed a new equilibrium code in which the structure of meshes in a r-z plane are corrected during iterations so that the ψ value of a certain mesh is always same and the magnetic axis is always on a mesh point. The algorithm to solve the Grad-Shafranov equation is, in principle, the usual solution method of a partial differential equation by the finite element method. The complementary azimuthal function $\chi(r,z)$ to $\psi(r,z)$ is obtained by the usual contour integration along a constant-ψ line.

Up to now two kinds of fixed boundary versions which use above scheme are completed. One (SELENE20) solves the equation as a non-linear eigenvalue problem[3)] and the other (SELENE30) solves it under the flux-conserving (FCT) condition[4)]. A free boundary version which solves the equation on a conventional mesh structure (SELENE40) is also by improving the evaluation procedure of vacuum contribution but it has not been tried yet.

(3) Resonance-like phenomena observed on the convergence curve

The cause of the phenomena has not been cleared. Features of the phenomena are,

i) This phenomenon occurs only at the limited data points ($N = 29,30,35,37$ in the case of $\beta_p \approx 1.0$).

ii) This phenomenon is more remarkable for an equilibrium with a flat current distribution.

iii) For different equilibrium meshes the growth rates differs each other at the resonance mesh number (Fig.5).

We conjectured that this phenomenon was caused by a certain mismatch of the equilibrium and stability mesh numbers and we neglected the data points where the resonance-like phenomena occured. By neglecting these data points the other data points are usually on a quadratic convergence curve.

3. SELENE — Equilibrium codes

It is necessary to calculate the MHD equilibrium for the stability analyses with high accuracy near a magnetic axis and in the outer region of a torus where magnetic surfaces are closely located, because conventional numerical codes for equilibrium calculation are formulated on the basis of square meshes in a r-z poloidal plane but the stability calculation in the ERATO code is carried out on square meshes in a ψ-χ plane. To avoid this discrepancy we developed a new equilibrium code in which the structure of meshes in a r-z plane are corrected during iterations so that the ψ value of a certain mesh is always same and the magnetic axis is always on a mesh point. The algorithm to solve the Grad-Shafranov equation is, in principle, the usual solution method of a partial differential equation by the finite element method. The complementary azimuthal function $\chi(r,z)$ to $\psi(r,z)$ is obtained by the usual contour integration along a constant-ψ line.

Up to now two kinds of fixed boundary versions which use above scheme are completed. One (SELENE20) solves the equation as a non-linear eigenvalue problem[3] and the other (SELENE30) solves it under the flux-conserving (FCT) condition[4]. A free boundary version which solves the equation on a conventional mesh structure (SELENE40) is also used and completed. The interface between the SELENE code and the ERATO code has not been completed, but the smaller number of meshes than the conventional ones is expected to give the highly accurate numerical equilibrium.

3.1 Numerical method

The Grad-Shafranov equation which describes the axisymmetric toroidal equilibria is given by

$$\Delta^* \psi = -\mu_0 r j_\varphi(r,\psi) \quad , \tag{1}$$

$$\Delta^* \equiv r\frac{\partial}{\partial r}\frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2} \quad . \tag{2}$$

In the fixed boundary problem, the boundary conditions are

$$\psi = \psi_{axis} \text{ (a given constant) at a magnetic axis} \tag{3}$$

and

$$\psi = 0 \qquad \text{at a plasma surface} \quad . \tag{4}$$

For the finite element formulation of eq.(1), we adopt the functional L for the equation as follows.

$$L(\psi^{n+1}) = -\int_V r dr dz\{(\frac{1}{r}\frac{\partial\psi^{n+1}}{\partial r})^2 + (\frac{1}{r}\frac{\partial\psi^{n+1}}{\partial z})^2 - \frac{2}{r}\mu_0 j_\varphi(r,\psi^n)\psi^{n+1}\} \quad , \tag{5}$$

where the index n means the iteration step. By using an appropriate set of basis functions, the functional L (eq.(5)) is represented by M parameters (ψ_1^{n+1}, ----- ψ_M^{n+1}) where M is the number of the node points. From eq.(5) simultaneous linear equations with respect to ψ_i^{n+1} is immediately derived, that is,

$$A\psi^{n+1} = b^n \quad , \tag{6}$$

where ${}^t\psi^{n+1} = (\psi_1^{n+1}$, ----- $\psi_M^{n+1})$ and the matrix A is the finite element representation of the operator Δ^*.

In SELENE20 the current distribution is determined so that ψ^n may be $\psi_{axis} \leq \psi \leq 0$, that is,

$$j_\varphi(r,\psi^n) = \lambda^n \; (r \frac{dp}{d\psi} (\psi^n) + \frac{1}{\mu_0 r} \frac{1}{2} \frac{dT^2}{d\psi} (\psi^n)) \quad , \tag{7}$$

and

$$\lambda^n = - \frac{1}{\psi_0^n} \lambda^{n-1} \quad . \tag{8}$$

The parameter λ^n corresponds to the eigenvalue of the eq.(1). The function $p(\psi)$ and $T(\psi)$ are given a priori or by the results of the transport codes. In SELENE30 j_φ is determined under the flux-conserving condition where the profiles of $p(\psi)$ and the safety factor $(q(\psi))$ are given, that is,

$$j_\varphi(r,\psi^n) = r \frac{dp}{d\psi} (\psi^n) + \frac{1}{\mu_0 r} T^n \frac{dT^n}{d\psi} \quad , \tag{9}$$

$$T^n = 4\pi^2 q(\psi^n) \; (\frac{d\psi}{dv})^n \frac{1}{A^n} \quad , \tag{10}$$

and

$$A^n = (\oint \frac{1}{r} \frac{d\ell}{|\nabla\psi|} \; / \oint \frac{r \; d\ell}{|\nabla\psi|} \;)_{\psi=\psi^n} \quad , \tag{11}$$

where $\frac{d\psi}{dV}$ is obtained by solving the surface-avaraged Grad-Shafranov equation simultaneously,

$$\frac{d}{dV} \; [<| \frac{|\nabla V|}{r} |^2 >^n \frac{d\psi}{dV} \;] = - \frac{dp}{d\psi} - q \frac{d}{dV} \; [\; \frac{q}{A^n} \frac{d\psi}{dV} \;] \quad . \tag{12}$$

The boundary conditions (3) and (4) are also used for eq.(12) $(\psi(V{=}0) = \psi_{axis}$ and $\psi(V{=}V_{max}) = 0)$.

The overall iteration procedure is summarized as follows.

STEP1 : Prepare initial meshes

STEP2 : Calculate $j_\varphi(r,\psi^n)$

STEP3 : Solve eq.(6)

STEP4 : Move mesh points on the constant ψ lines

STEP5 : if $|r_{axis}^{n+1} - r_{axis}^{n}| > \varepsilon$ or $|\lambda^{n+1} - \lambda^{n}| > \varepsilon'$ (SELENE20), or $|r_{axis}^{n+1} - r_{axis}^{n}| > \varepsilon$ or $|\frac{d\psi^{n+1}}{dV} - \frac{d\psi^{n}}{dV}| > \varepsilon'$ (SELENE30), then return to STEP2

STEP6 : End of iterations and calculations of metrices

3.2 Numerical results

For an example of calculations, we show the finial meshes which represent the constant ψ lines in Fig.8. Figure 9 shows the convergence of the eigenvalue (λ) in SELENE20. From this figure we find the guadratic convergence of λ versus 1/N, where $N = N_\psi = N_\chi$, and N_ψ and N_χ are the number of the division in the radial and poloidal direction. For the check of the accuracy, we compared the magnetic shear ($S \equiv -\frac{d}{d\psi}[T\oint(1/r|\nabla\psi|)d\ell] / (\frac{dV}{d\psi})^3$) between the analytic ($A = \infty$, $E = 1$) and the computed ($A = 10,20,$ and 40, $E = 1$) cases for the parabolic current distribution (Fig.10), where A and E are the aspect ratio and the ellipticity of the poloidal cross-section. Both results coincide in very high accuracy. The CPU time required in FACOM 230-75 versus the number of the division ($N = N_\psi = N_\chi$) is shown in Fig.11. In this figure the expected CPU time in CDC 7600 is also shown.

4. BOREAS — the localized mode analysis code[5)]

The ERATO code is useful for the analyses of the low mode number stabilities (n = 1-5, n: the toroidal mode number). On the other hand, the code for the analyses of the high-n mode is complementally necessary. The criteria for the stability of the high-n mode are the Mercier criterion and one for the high-n ballooning mode. These criteria are checked in the BOREAS code. The ERATO1 module will be replaced by the linked module of the SELENE and BOREAS with the routines for the calculation of the metrices. All metrices used in the ERATO are calculated in the present version of the SELENE and the calculation of "the Array EQ" in the ERATO1 has not been completed yet.

4.1 Numerical method

For the analysis of the high-n ballooning mode, we use the energy integral given by Conor et al.[6)] By Fourier-transformation of the normal displacement $X = rB_p\xi_\psi$ the potential energy is written as

STEP3 : Solve eq.(6)

STEP4 : Move mesh points on the constant ψ lines

STEP5 : if $|r^{n+1}_{axis} - r^{n}_{axis}| > \varepsilon$ or $|\lambda^{n+1} - \lambda^{n}| > \varepsilon'$ (SELENE20), or $|r^{n+1}_{axis} - r^{n}_{axis}| > \varepsilon$ or $|\frac{d\psi^{n+1}}{dV} - \frac{d\psi^{n}}{dV}| > \varepsilon'$ (SELENE30), then return to STEP2

STEP6 : End of iterations and calculations of metrices

3.2 Numerical results

For an example of calculations, we show the finial meshes which represent the constant ψ lines in Fig.8. Figure 9 shows the convergence of the eigenvalue (λ) in SELENE20. From this figure we find the guadratic convergence of λ versus 1/N, where $N = N_\psi = N_\chi$, and N_ψ and N_χ are the number of the division in the radial and poloidal direction. For the check of the accuracy, we compared the magnetic shear ($S \equiv -\frac{d}{d\psi}\,[T\oint(1/r|\nabla\psi|)d\ell]\,/\,(\frac{dV}{d\psi}\,)^3$) between the analytic ($A = \infty$, $E = 1$) and the computed ($A = 10,20$, and 40, $E = 1$) cases for the parabolic current distribution (Fig.10), where A and E are the aspect ratio and the ellipticity of the poloidal cross-section. Both results coincide in very high accuracy. The CPU time required in FACOM 230-75 versus the number of the division ($N = N_\psi = N_\chi$) is shown in Fig.11. In this figure the expected CPU time in CDC 7600 is also shown.

4. BOREAS — the localized mode analysis code[5)]

The ERATO code is useful for the analyses of the low mode number stabilities (n = 1-5, n: the toroidal mode number). On the other hand, the code for the analyses of the high-n mode is complementally necessary. The criteria for the stability of the high-n mode are the Mercier criterion and one for the high-n ballooning mode. These criteria are checked in the BOREAS code. The ERATO1 module will be replaced by the linked module of the SELENE and BOREAS with the routines for the calculation of the metrices. All metrices used in the ERATO are calculated in the present version of the SELENE and the calculation of "the Array EQ" in the ERATO1 has not been completed yet.

4.1 Numerical method

For the analysis of the high-n ballooning mode, we use the energy integral given by Conor et al.[6)] By Fourier-transformation of the normal displacement $X = rB_p\xi_\psi$ the potential energy is written as follows.

$$W = \pi \int d\psi \; W(\psi) \quad , \tag{13}$$

$$W(\psi) = \int_{-\infty}^{\infty} d\eta \; \{f(\psi,\eta) \; \left|\frac{dX_0}{d\eta}\right|^2 + g(\psi,\eta)\left|X_0\right|^2\} \quad , \tag{14}$$

$$f(\psi,\eta) = \frac{1}{Jr^2B_p^2} + \frac{r^2B_p^2}{J^2B^2} \left(\int_{-\infty}^{\eta} \frac{\partial\nu}{\partial\psi} d\eta\right)^2 \quad , \tag{15}$$

$$g(\psi,\eta) = -\frac{2J}{B^2}\frac{dp}{d\psi}\left[\frac{\partial}{\partial\psi}\left(\mu_0 p + \frac{B^2}{2}\right) - \frac{T}{B^2}\frac{1}{J}\frac{\partial}{\partial\eta}\left(\frac{B^2}{2}\right)\int_{-\infty}^{\eta}\frac{\partial\nu}{\partial\psi}d\eta\right], \tag{16}$$

$$B^2 = B_p^2 + \frac{T^2}{r^2} \quad , \tag{17}$$

$$\nu = \frac{TJ}{r^2} \quad , \tag{18}$$

where J is the Jacobian and X_0 is the fourier coefficient of X. All quantities except $\int_{-\infty}^{\eta} \frac{\partial\nu}{\partial\psi} d\eta$ are periodic in 2π with respect to η.

By using the finite hybrid element method and expressing $\frac{dX_0}{d\eta}$ and X_0 as

$$\frac{dX_0}{d\eta} = \frac{X_{i+1} - X_i}{\eta_{i+1} - \eta_i} \quad , \tag{19}$$

and

$$X_0 = \frac{X_{i+1} + X_i}{2} \quad , \tag{20}$$

the functional $W(\psi)$ becomes the quadratic form

$$\hat{W} = {}^tX \; A \; X \quad , \tag{21}$$

$${}^tX = (X_1, \text{-----} \; X_N) \quad , \tag{22}$$

where X_i is the nodal value at a mesh point on a magnetic surface and N is the number of mesh points. The minimum value of $\hat{W}$ is given by the minimum eigenvalue (λ_{min}) of the matrix A. If λ_{min} is negative, the plasma is ballooning-unstable. The matrix A is a tridiagonal one so that λ_{min} is obtained very fast by using the usual bisection method.

4.2 Numerical results

The computed λ_{min} gives the pessimistic value as so in the ERATO code, but it remarkably does not seem to change the sign as $N \to \infty$. The convergence property is shown in Fig.12. Though this property has not been theoretically proved yet, we can determine whether the plasma is stable or unstable without extrapolation with respect to N, if our conjecture is correct. As an application of the BOREAS code, we tested several classes of equilibria by using the INTOR parameters (A = 4.2, E = 1.5),

Class 1:
$$p = \beta_J p_0 \{(\psi-\psi_b) - \frac{\alpha}{2}[(\psi-\psi_m)^2 - (\psi_b-\psi_m)^2] + \frac{\gamma}{L+1}[(\psi-\psi_m)^{L+1} - (\psi_b-\psi_m)^{L+1}]\} \quad , \quad L \geq 4 \tag{23}$$

$$\gamma = \frac{\alpha(\psi_m-\psi_b)-1}{(\psi_b-\psi_m)^L} \quad , \tag{24}$$

$$T\frac{dT}{d\psi} = R_0^2\ (1/\beta_J-1)\frac{dp}{d\psi} \quad , \tag{25}$$

Class 2:
$$p = \left(\frac{\psi-\psi_b}{\psi_m-\psi_b}\right)^{\alpha} \quad , \tag{27}$$

$$T\frac{dT}{d\psi} = R_0^2\ (1/\beta_J-1)\frac{dp}{d\psi} \quad , \tag{28}$$

Class 3:
$$p = \frac{\alpha_1}{2}\psi^2 - \frac{\alpha_3}{3}\psi^3 \quad , \tag{29}$$

$$T\frac{dT}{d\psi} = \alpha_3 R_0^2\psi^2 + \alpha_4\psi^4 \quad . \tag{30}$$

Results are summarized as the figure of the critical beta value (β_c) versus the poloidal beta value (β_p) in Fig.13.

5. APOLLO — 1-1/2 dimensional tokamak code

The plasma equilibrium depends on the plasma parameters which evolve according to the transport processes, and this evolution of the magnetic geometry affects the plasma transport. This process is especially important in a tokamak with finite beta value and/or non-circular cross-section. To solve this problem, the consistent treatment of the plasma transport and the MHD equilibrium is required[7),8)]. The APOLLO code has been developed to study this reciprocal relation between the plasma transport and the MHD equilibrium. One of the main aspects of this code is to give the realistic (or non-intuitive) equilibrium configuration for the MHD stability analysis.

The basic structure of the APOLLO code is the combination of the transport step and equilibrium step and these two steps are solved alternatively. In the transport step, plasma parameters are advanced in time for the fixed magnetic geometry. Using the fact that all transport processes along the magnetic field advance faster than those in the radial direction, the problem is reduced to the 1-dimensional problem. Choosing the toroidal flux function (χ) as the radial coordinate, the transport equations to be solved are

$$\frac{\partial}{\partial t}\,[n_e \frac{\partial v}{\partial \chi}] + \frac{\partial}{\partial \chi}\,[\frac{\partial v}{\partial \chi}\,\Gamma_e] = S \quad , \qquad (31)$$

$$\frac{3}{2}\frac{\partial}{\partial t}\,[p_e\,(\frac{\partial v}{\partial \chi})^{5/3}] + (\frac{\partial v}{\partial \chi})^{2/3}\,\frac{\partial}{\partial \chi}\,\{\frac{\partial v}{\partial \chi}\,[q_e + \frac{5}{2}\,T_e\Gamma_e]\} = Q_e \quad , \qquad (32)$$

$$\frac{3}{2}\frac{\partial}{\partial t}\,[p_i\,(\frac{\partial v}{\partial \chi})^{5/3}] + (\frac{\partial v}{\partial \chi})^{2/3}\,\frac{\partial}{\partial \chi}\,\{\frac{\partial v}{\partial \chi}\,[q_i + \frac{5}{2}\,T_i\Gamma_i]\} = Q_i \quad , \qquad (33)$$

$$\frac{\partial}{\partial t}\,B_{p0} = \frac{\partial}{\partial \chi}\,\{\frac{\eta_{/\!/}}{\mu_0 \langle r^{-2}\rangle^2\,(\frac{\partial v}{\partial \chi})^2}\,\frac{\partial}{\partial \chi}\,[\frac{\langle|\nabla v/r|^2\rangle}{\frac{\partial v}{\partial \chi}}\,B_{p0}]\} \quad , \qquad (34)$$

and

$$B_{p0} = \frac{\partial \psi}{\partial \chi} = \frac{1}{2\pi q} \quad , \qquad (35)$$

where $v(\chi)$ is the volume contained by the χ-surface, and the angular bracket < > denotes the surface avaraged quantity. S is the particle source and the Q's represent energy source or loss due to the ohmic heating, injection heating, radiation and charge exchange processes. The particle flux Γ_e and the energy flux $q_{e,i}$ depend on the transport model and on the geometric factors[9].

5.1 Numerical method

This system of transport equations is solved by the finite element method and the implicit time-integration scheme. The former makes it easy to introduce variable mesh without reducing the accuracy, while the latter guarantees the numerical stability.

In the equilibrium step, the Grad-Shafranov equation

$$\Delta^* \psi = - \mu_0 r^2 \frac{dp}{d\psi} - T \frac{dT}{d\psi} \tag{36}$$

is solved by using the SELENE30 under the adiabatic constraints, which require that the following quantities must be conserved: $\mu(\psi) = p(\psi)(\frac{\partial v}{\partial \psi})^{5/3}$, $q(\psi) = T(\psi) \frac{\partial v}{\partial \psi} \frac{<r^{-2}>}{(2\pi)^2}$ and ψ-values at the magnetic axis and the plasma surface. These constraints can be insured by solving the surface-avaraged equilibrium equation

$$\frac{d}{dv} [<|\frac{\nabla v}{r}|^2> \frac{d\psi}{dv}] = - \mu_0 \frac{d}{d\psi} \mu(\frac{\partial \psi}{\partial v})^{5/3} - q \frac{d}{dv} [\frac{q}{<r^{-2}>} \frac{d\psi}{dv}] \quad , \tag{37}$$

for given boundary values of ψ. The combined system of eqs.(36) and (37) gives the solution by the iteration. Since we have to calculate several tens or hundreds of equilibria in one transport simulation, the equilibrium solver with high speed and high accuracy is required. For this purpose, we developed two kinds of equilibrium modules. The one, SELENE30, solves the fixed boundary equilibrium by using the variable meshes. The other module, SELENE40, solves the free boundary equilibrium. The basic procedure of this module is the combination of the Buneman solver of the Poisson type equation with the method of the surface Green function. Both equilibrium module include the ballooning code (the BOREAS or the BALOON code) and automatically checks the ballooning stability of the high-n mode.

5.2 Numerical results

Figure 14 shows an example of the results of analyses of ballooning-stable equilibria. This result is obtained by using the APOLLO30, which is the combination of the SELENE30, BOREAS and the simplified one-fluid transport module. Figure 15 show the result by using the APOLLO40, which simulates the process of the FCT heating.

5.3 Interface with ERATO

In the APOLLO code $p(\psi)$ and $q(\psi)$ (or $p(\chi)$ and $q(\chi)$ are calculated on the meshes $\psi = \psi_1$, -----, ψ_N. The number of the mesh points for the equilibrium calculation in the APOLLO is usually $N_\psi = N_\chi = 15$ (SELENE30) or $N_r = N_z = 64$ (SELENE40). This number is too small to insure the accuracy for the stability analysis using the ERATO code. The recalculation of the equilibrium is, therefore, necessary by using a large number of mesh points for the given $p(\psi)$ and $q(\psi)$ by the APOLLO.

The relation between the ERATO code and our SELENE and BOREAS code is summarized in Fig.16

6. Machine Optimization

Distinctions of the ERATO job are as follows;

i) A very large amount of I/O operations are required as compared with amount of other arithmetic operations performed on a CPU. Necessary I/O time is too large for a standard Fortran-supplied I/O routine of a general purpose computer to process a lot of large ERATO jobs smoothly during regular service of a computer center.

ii) Necessary CPU time rapidly becomes large if higher resolution of results and calculations of higher mode number instability are required.

In order to overcome the first difficulty the standard Fortran-supplied I/O routine was replaced by a specially developed "non-buffering I/O routine" written in Assembly language. As for the second point we optimized the ERATO code for a vector processing computer, FACOM 230-75 APU (F230-75APU hereafter: APU = Array Processing Unit) and run the code on the F230-75APU/CPU multiprocessor system[10).

In this section we describe the process how the ERATO code was optimized with respect to the I/O operations and the effectiveness of parallel processing for the ERATO calculations which was investigated by using the above F230-75APU/CPU multisystem.

one-fluid transport module. Figure 15 show the result by using the APOLLO40, which simulates the process of the FCT heating.

5.3 Interface with ERATO

In the APOLLO code $p(\psi)$ and $q(\psi)$ (or $p(\chi)$ and $q(\chi)$ are calculated on the meshes $\psi = \psi_1$, -----, ψ_N. The number of the mesh points for the equilibrium calculation in the APOLLO is usually $N_\psi = N_\chi = 15$ (SELENE30) or $N_r = N_z = 64$ (SELENE40). This number is too small to insure the accuracy for the stability analysis using the ERATO code. The recalculation of the equilibrium is, therefore, necessary by using a large number of mesh points for the given $p(\psi)$ and $q(\psi)$ by the APOLLO.

The relation between the ERATO code and our SELENE and BOREAS code is summarized in Fig.16

6. Machine Optimization

Distinctions of the ERATO job are as follows;

i) A very large amount of I/O operations are required as compared with amount of other arithmetic operations performed on a CPU. Necessary I/O time is too large for a standard Fortran-supplied I/O routine of a general purpose computer to process a lot of large ERATO jobs smoothly during regular service of a computer center.

ii) Necessary CPU time rapidly becomes large if higher resolution of results and calculations of higher mode number instability are required.

In order to overcome the first difficulty the standard Fortran-supplied I/O routine was replaced by a specially developed "non-buffering I/O routine" written in Assembly language. As for the second point we optimized the ERATO code for a vector processing computer, FACOM 230-75 APU (F230-75APU hereafter: APU = Array Processing Unit) and run the code on the F230-75APU/CPU multiprocessor system[10].

In this section we describe the process how the ERATO code was optimized with respect to the I/O operations and the effectiveness of parallel processing for the ERATO calculations which was investigated by using the above F230-75APU/CPU multisystem.

6.1 Optimization of ERATO with respect to I/O operations

The ERATO code was adapted for FACOM230-75 (F230-75 hereafter) computer system of JAERI from the code written for CDC6600 computer system at EPF-Lausanne. Characteristics of the F230-75 system, and differences between the F230-75 and CDC6600 computers are summarized in Appendix A with some remarks concerning adaptation of the code.

(1) Standard problems for test runs

Four kinds of matrices are prepared for test calculations. Parameters which characterize the matrices are summarized in Table 1. The matrices of the case 1 are those for a stability analysis of an analytically obtained equilibrium. This case is provided by ERATO as a standard test case (NTCASE = 1). The matrices of the cases 2,3 and 4 are those for a stability analysis of a numerically obtained equilibrium (NTCASE = 3). These three cases are different in the size of the matrices. In the I/O optimization only the cases 1 and 2 were tested. Table 2 summarizes CPU time and I/O time for the cases 1 and 2 before the optimization. From this table it is easily seen that most of CPU and I/O times are both spent in the ERATO4 module.

(2) Idle time of the I/O control

As shown in Table 2 the I/O time spent in the ERATO4 module for the case 2 calculation is about 453 sec. This value is fairly large as compared with the CPU time. In this test case matrix data of 122,081 KB are transferred between the main and disk memories and, consequently, transfer rate of 269 KB/sec is attained. This fact indicates that the I/O channel of the F230-75 system is very inefficiently*) used in the ERATO calculation and there is much idle time in the I/O control of the system because the maximum transfer rate of the disk memories (F479B2 and F1771D) is 806 KB/sec which is by far large compared with the actual transfer rate.

*) The inefficiency of the I/O channel of the F230-75 system for the ERATO4 calculation is attributed to a large number of BACKSPACE operation of the disk files. In order to perform backward substitution appearing in the inverse iteration loop, one should read one block of matrix data as one logical record from a sequential file, which required to perform a sequence of three operations, "BACKSPACE, READ, BACKSPACE". If the logical record size exceeds the size of a track, the standard Fortran I/O routine always checks if the data are transferred correctly, which increases the I/O time by a factor of about 3.0. Therefore, it is recommendable to avoid the BACKSPACE operations as far as possible.

(3) Optimization of the I/O process within the framework of a Fortran program

Because the length of the sequential data is fixed constant in the ERATO4 module, even in the Fortran framework it is possible to remove the BACKSPACE operations as follows,

i) Preparation of as many sequential files as the number of blocks of a matrix

Obviously the BACKSPACE operation is not necessary in this case, by reading data of different blocks from different files.

ii) Use of random access files

This method is also effective because the number of READ operations is by far larger than that of WRITE operations and necessary time for opening files is negligible.

(4) An Assembler I/O routine

As shown later the above methods are effective to reduce I/O time of the ERATO jobs. However, necessary time for calling the Fortran I/O routine cannot be neglected and there is room to reduce the I/O time much more by using a specially developed Assembler I/O routine. Because the Fortran I/O routine is general-purpose one it provides various kinds of error check routines. But we are concerned only with the case where a large number of matrix elements are transferred between the main memory and work files, and in this case a specially developed single-purpose I/O routine is helpful to reduce the I/O time. Actually "End of Cylinder" check which is issued at the changeover from a cylinder to another adds to the I/O time. By omitting this check the I/O time is reduced considerably. For this purpose the Fortran I/O routine was replaced by an Assembler (FASP language) I/O routine which has the following features.

i) This routine does not use the Fortran I/O buffer. Data are transferred between user specified arrays and disk memories directly.

ii) A head of a record is always written from the head of a cylinder. Length of command chaining should not exceed 19 tracks.

(5) Measurements of processing time

The results of the measurements are summarized in Table 3. By using the Assembler I/O routine both CPU time and I/O time are reduced.

Especially, considerable reduction of the I/O time is observed. For detailed information effects of the improvement on basic I/O operations are summarized in Appendix B.

6.2 Optimization of the code for FACOM230-75APU

F230-75APU is a high speed vector-processor developed after demand for high speed array manipulation. The F230-75APU is always operated as an additional processor to a F230-75 computer system. That is, the general purpose processor F230-75CPU and additional processor F230-75APU form an "asymmetric multiprocessor system".

In Appendix C a brief explanation of the F230-75APU is presented. The description of the software system for the APU-CPU multiprocessor system, especially, on the AP-Fortran and its optimization are presented in Appendix D[11,12].

(1) Automatic optimizations by the AP-Fortran and manual optimizations

In order to demonstrate the effectiveness of the parallel processing in the F230-75APU sufficiently, much more extensive optimizations of the code is generally required in addition to those by the AP-Fortran. Essential points required for the optimizations are as follows.

i) Rewriting of a Fortran source program to a vector-processor oriented form

First, the sequence of data should be carefully rearranged, which, sometimes, requires alteration of algorithms. In the ERATO case, the Gauss elimination method used for the matrix decomposition was replaced by the Choleski method (see Appendix E). After this change of algorithm slight reduction of CPU time was observed for the calculation by F230-75CPU. The effect of the change is by far large in the case of APU-CPU multiprocessor system.

Secondly, as in the APU-CPU multiprocessor system I/O channels are connected only to F230-75CPU the I/O operations interrupt a continuous flow of vector operations. Therefore it is very important to collect the I/O operation flows and vector processing flows separately in the program, though there is no room in the ERATO case. Because of a similar reason the scalar calculation flows and those for the vector calculations should be separated.

ii) Efficient use of vector registors (Vector temporary)

In F230-75APU there are 1792 W vector registors (Vector temporary). To use these vector registors very efficiently as working registors is the most essential point how to optimize the program for the F230-75APU.

Parts of the above optimizations are performed by the AP-Fortran but the level of the optimization is not, in general, satisfactory.

(2) Performance of F230-75APU for the ERATO job

The processing time of ERATO by the F230-75CPU and F230-75APU is summarized in Table 4 and Fig.17. The data of the table indicate that the overall performance ratio of F230-75APU to F230-75CPU for the ERATO4 module is 2 ~ 4 (OPT0) and 3.4 ~ 5.5 (OPT2). In Table 5 the performance ratios are presented for basic arithmetic operations separately.

6.3 Summary on the machine optimization

The main part of the ERATO code is an extremely I/O-dominant program but a general purpose computer system installed at a usual scientific computer center is not designed to handle such an extremely I/O-dominant job. Therefore, turn around time for such a job is considerably prolonged on a standard general purpose computer system. However, as far as the data format and amount of the transferred data are fixed as in the case of ERATO, one can write an efficient I/O routine in the EXCP level (most primitive I/O control level) by using the Assembler. We succeeded in reducing the I/O time of ERATO by the specially developed Assembler I/O routine. This fact indicates importance of analysis of task distribution for the concerned code and the effectiveness of the machine optimization for a large-scale general-purpose computer system.

Because of the regularity of its array data ERATO is one of the best codes that can run very efficiently on a vector processing computer. By using the F230-75APU/CPU asymmetric multiprocessor system performance ratio of F230-75APU up to 6 was proved as compared with the job optimized for the F230-75 CPU computer system. In summary in order to demonstrate the ability of a super high speed vector processing computer very effectively the most important points are, i) appropriate restructuring of the program, which may be possible only by profound understanding of used algorithm, and ii) efficient use of high-level machine-dependent optimization functions of the vector compiler, which was partly shown effective by the AP-Fortran of the F230-75APU.

By the study in this report we consider the effectiveness of the vector processing computer are proved in the field of MHD analyses and a larger vector processing computer with higher performance is desired in the near future.

7. Summary and Discussions

In this report we have described several topics on the ERATO code itself. Results of stability analyses by ERATO are described elsewhere. From the descriptions in the previous sections matters to be attended and problems to be solved are sammarized as follows,

i) Convergence study:

1 For n = 1, 2 mode calculation stability mesh number up to ~37 is usually necessary. Equilibrium mesh number should be fixed to attain a straight convergence.

2 To obtain a converged value of a low growth rate mode extrapolation from higher growth rates is efficient.

3 To neglect the resonance-like phenomenon it is recommended to calculate as many data points as possible.

ii) Interface between equilibrium and stability code:

1 As the convergence property of the code is influenced by choice of mesh number and accuracy of the equilibrium code, an equilibrium code with high accuracy is required. For this purpose we developed a equilibrium code SELENE. But the interface is not completed.

2 Auxiliary codes (BOREAS, APOLLO) for the equilibrium code to be used for the ERATO calculation are prepared. But the interfaces are not still completed.

iii) Optimization of the code with respect to computer systems

1 Optimization of I/O routine (system routine) was carried out. By this optimization both CPU time and I/O time are reduced considerably. The reason why a very large factor of improvement is attained is that the ERATO calculation is an I/O bound job and a general-purpose computer system is not in the standard option designed to deal with such a large-scale I/O bound job as the ERATO calculation.

2 Optimization of the ERATO code for a vector computer has been tried by using FACOM230-75APU. It was confirmed that factor of about 6 could be gained in CPU time by compared with

vector processing computer are proved in the field of MHD analyses and a larger vector processing computer with higher performance is desired in the near future.

7. Summary and Discussions

In this report we have described several topics on the ERATO code itself. Results of stability analyses by ERATO are described elsewhere. From the descriptions in the previous sections matters to be attended and problems to be solved are sammarized as follows,

i) Convergence study:

1 For n = 1, 2 mode calculation stability mesh number up to ~37 is usually necessary. Equilibrium mesh number should be fixed to attain a straight convergence.

2 To obtain a converged value of a low growth rate mode extrapolation from higher growth rates is efficient.

3 To neglect the resonance-like phenomenon it is recommended to calculate as many data points as possible.

ii) Interface between equilibrium and stability code:

1 As the convergence property of the code is influenced by choice of mesh number and accuracy of the equilibrium code, an equilibrium code with high accuracy is required. For this purpose we developed a equilibrium code SELENE. But the interface is not completed.

2 Auxiliary codes (BOREAS, APOLLO) for the equilibrium code to be used for the ERATO calculation are prepared. But the interfaces are not still completed.

iii) Optimization of the code with respect to computer systems

1 Optimization of I/O routine (system routine) was carried out. By this optimization both CPU time and I/O time are reduced considerably. The reason why a very large factor of improvement is attained is that the ERATO calculation is an I/O bound job and a general-purpose computer system is not in the standard option designed to deal with such a large-scale I/O bound job as the ERATO calculation.

2 Optimization of the ERATO code for a vector computer has been tried by using FACOM230-75APU. It was confirmed that factor of about 6 could be gained in CPU time by compared with

FACOM230-75CPU.

In summary it seems that almost all the computational problems concerning the ERATO calculation will be solved by a small progress of the computer technology. However, it seems necessary to investigate extensively the equilibrium codes and their interfaces to the stability code in order that the ERATO code may contribute more and more the tokamak research.

Acknowledgements

The authors would like to express their sincere thanks to Dr. Masatoshi Tanaka for helpful discussions. They are also very grateful to Dr. Ralf Gruber for his kind guidance on usage of ERATO and useful discussions. The authors also express their thanks to Mr. Naoyuki Saito of JAERI computer center for his fruitful comments on I/O optimization. Thanks are also due to Dr. Yoshio Tago of Fujitsu Co., Ltd. for his helpful suggestion on the usage of the computer system. They would like to acknowledge the continuing encouragements of Drs. Sigeru Mori and Yukio Obata.

FACOM230-75CPU.

In summary it seems that almost all the computational problems concerning the ERATO calculation will be solved by a small progress of the computer technology. However, it seems necessary to investigate extensively the equilibrium codes and their interfaces to the stability code in order that the ERATO code may contribute more and more the tokamak research.

Acknowledgements

The authors would like to express their sincere thanks to Dr. Masatoshi Tanaka for helpful discussions. They are also very grateful to Dr. Ralf Gruber for his kind guidance on usage of ERATO and useful discussions. The authors also express their thanks to Mr. Naoyuki Saito of JAERI computer center for his fruitful comments on I/O optimization. Thanks are also due to Dr. Yoshio Tago of Fujitsu Co., Ltd. for his helpful suggestion on the usage of the computer system. They would like to acknowledge the continuing encouragements of Drs. Sigeru Mori and Yukio Obata.

References

1) R. Gruber, to be published in CPC.

2) R. Gruber et al., "Dependence of Ideal MHD Beta Limits on Current Density and Pressure Profiles", Seventh International Conf. on Plasma Physics and Controlled Nuclear Fusion Research (Innsbruck, Austria 23-30 August 1978).

3) T. Takeda and T. Tsunematsu, "A numerical code SELENE to calculate axisymmetric toroidal equilibria", JAERI-M 8042 (1979).

4) J.F. Clarke, "High beta flux-conserving tokamaks", ORNL/TM-5429 (1976).

5) T. Takeda and T. Tsunematsu, "The high-n ballooning stability code BOREAS", to appear in JAERI-M.

6) J.M. Connor, R.J. Hastie, and J.B. Taylor, Phys. Rev. Lett. 6 (1978) 396.

7) H. Grad, "Survey of 1-1/2 D transport codes", COO-3077-154, MF-93 (1978).

8) F.J. Helton, R.L. Miller, and J.M. Rawls, J. Comp. Phys., 24 (1977) 117.

9) J.T. Hogan, "The accessibility of high β tokamak states", ORNL/TM-6049 (1978).

10) O. Miwa, N. Kume, K. Uchida, S. Suzuki, Y. Tanakura, and F. Isobe, "FACOM 230-75 Array Processor System", FUJITSU, 29 (1978) 93.

11) FACOM 230-75 M VII AP-FORTRAN Manual, 75FP-0370-1 (1977).

12) FACOM 230-75 M VII AP-FORTRAN User's Guide, 75FP-0500-1 (1978).

Table 1 Data concerning ERATO4

CASE	MESH NUMBER		Matrix Size	Size of a Block	No. of Iterations
	N_ψ	N_χ			
1	14	15	1376	128	6
2	24	25	3796	208	7
3	37	37	8512	304	3
4	49	49	14800	400	1

Table 2 Execution time of ERATO on F230-75 CPU system (OS-MONITOR VII, FORTRAN-H OPT2)

CASE 1. $N_\psi = 14$, $N_\chi = 15$, No. of Iteration = 6, NTCASE = 1

Job Step	CPU Time	I/O Time*
ERATO1	2.552	5.196
ERATO2	0.043	0.090
ERATO3	6.211	6.444
ERATO4	34.630	86.471
ERATO5	0.906	3.543

CASE 2. $N_\psi = 24$, $N_\chi = 25$, No. of Iteration = 7, NTCASE = 3

Job Step	CPU Time	I/O Time*
ERATO1	6.384	17.608
ERATO2	31.883	0.448
ERATO3	25.306	25.050
ERATO4	182.090	453.492
ERATO5	2.456	9.962

*) "I/O time" of F230-75 system is the accumulation of time intervals between the instant when "Start I/O" command is issued and that when "Channel end" command is executed. Hereafter we use "I/O time" as a characteristic quantity to indicate the I/O performance of each program in a multiprogramming computer system.

Table 3 Execution time of ERATO I/O procedure

(a) Sequential file (FORTRAN I/O library, the original program)

(b) Distributed sequential files (FORTRAN I/O library)

(c) Direct access file (FORTRAN I/O library)

(d) Distributed sequential files (Assembler I/O routine)

CASE 1.

	I/O in ERATO3		I/O in ERATO4	
	CPU Time	I/O Time	CPU Time	I/O Time
(a)	0.322	6.444	3.232	86.471
(b)	0.322	6.444	3.301	65.785
(c)	0.322	6.444	3.253	75.595
(d)	0.058	6.697	0.141	42.631

CASE 2.

	I/O in ERATO3		I/O in ERATO4	
	CPU Time	I/O Time	CPU Time	I/O Time
(a)	1.224	25.050	16.070	453.492
(b)	1.224	25.050	15.571	318.150
(c)	1.224	25.050	15.588	307.021
(d)	0.074	26.658	0.394	194.776

Table 4 Execution Time of ERATO4 on FACOM230-75APU/CPU system

CASE 1. No. of iteration = 6, I/O Time = 42.631 sec.
Task Switching = 538 Times

Compiler	CPU Time (sec)
FORTRAN-H(OPT2)	19.761
AP-FORTRAN(OPTO)	9.564 (9.389 on AP Task) (0.175 on CP Task)
AP-FORTRAN(OPT2)	5.885 (5.711 on AP Task) (0.174 on CP Task)

CASE 2. No. of iteration = 7, I/O Time = 162.981 sec.
Task Switching = 1030 Times

Compiler	CPU Time (sec)
FORTRAN-H(OPT2)	126.542
AP-FORTRAN(OPTO)	43.391 (42.965 on AP Task) (0.426 on CP Task)
AP-FORTRAN(OPT2)	27.938 (27.513 on AP Task) (0.425 on CP Task)

CASE 3. No. of iteration = 3, I/O Time = 208.531 sec.
Task Switching = 835 Times

Compiler	CPU Time (sec)
FORTRAN-H(OPT2)	476.499
AP-FORTRAN(OPTO)	135.108 (134.386 on AP Task) (0.722 on CP Task)
AP-FORTRAN(OPT2)	90.435 (89.714 on AP Task) (0.721 on CP Task)

CASE 4. No. of iteration = 1, I/O Time = 267.465 sec.
Task Switching = 605 Times

Compiler	CPU Time (sec)
FORTRAN-H(OPT2)	1402.960
AP-FORTRAN(OPTO)	351.327 (350.512 on AP Task) (0.815 on CP Task)
AP-FORTRAN(OPT2)	253.297 (252.421 on AP Task) (0.876 on CP Task)

Table 5 Execution time of Choleski-decomposition and Inverse-iteration

CASE 1

	CPU Time (sec)		Ratio
	FORTRAN-H(OPT2)	AP-FORTRAN(OPT2)	
Choleski-decomposition	13.833	3.343	4.14
One Inverse-iteration	0.912	0.237	3.85

CASE 2

	CPU Time (sec)		Ratio
	FORTRAN-H(OPT2)	AP-FORTRAN(OPT2)	
Choleski-decomposition	97.400	17.340	5.62
One Inverse-iteration	4.018	0.831	4.84

CASE 3

	CPU Time (sec)		Ratio
	FORTRAN-H(OPT2)	AP-FORTRAN(OPT2)	
Choleski-decomposition	449.631	68.103	6.60
One Inverse-iteration	12.790	2.392	5.35

CASE 4

	CPU Time (sec)		Ratio
	FORTRAN-H(OPT2)	AP-FORTRAN(OPT2)	
Choleski-decomposition	1339.842	214.096	6.26
One Inverse-iteration	29.013	5.116	5.67

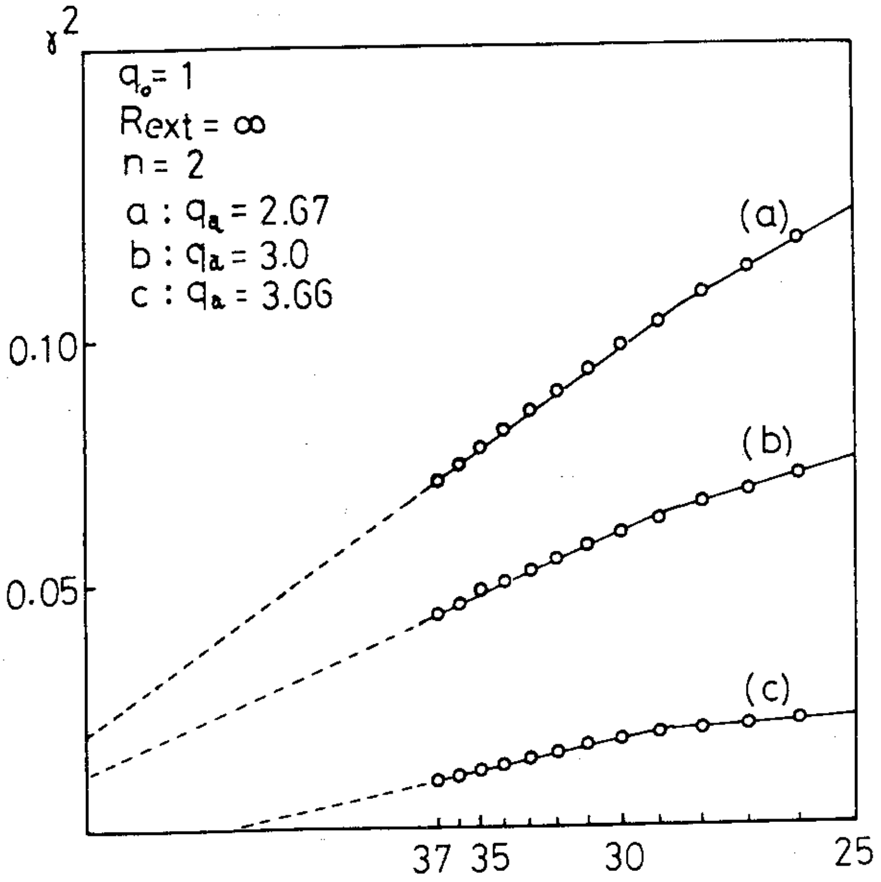


Fig. 1 Example of convergence curve

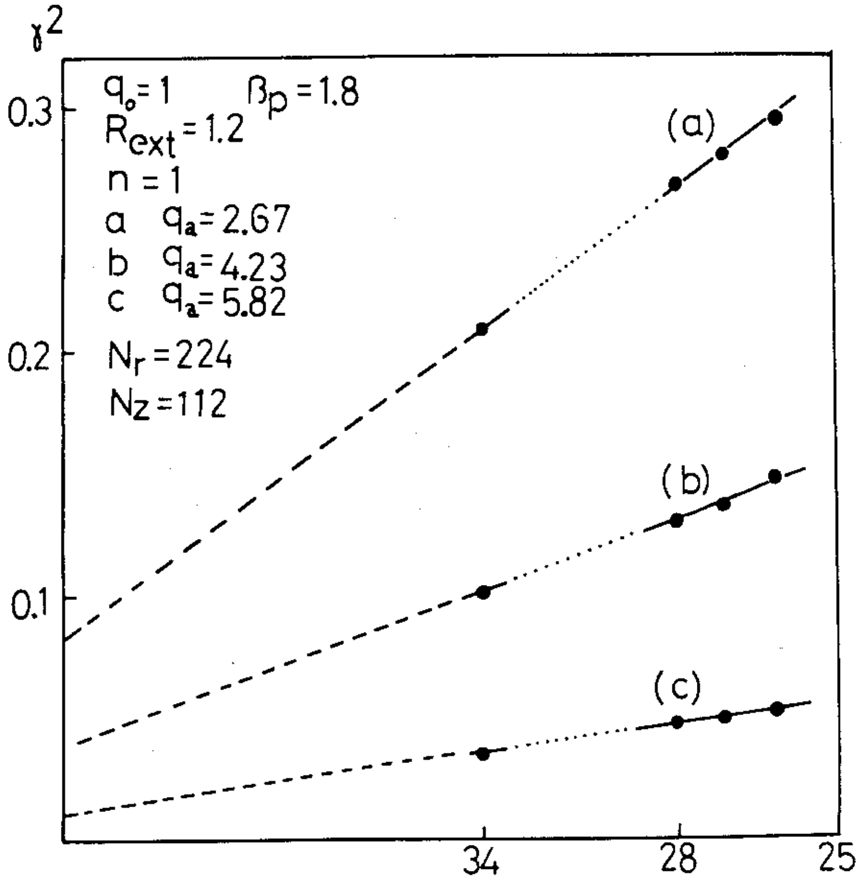


Fig. 2 Convergence curve on which the points N = 29–33 are omitted

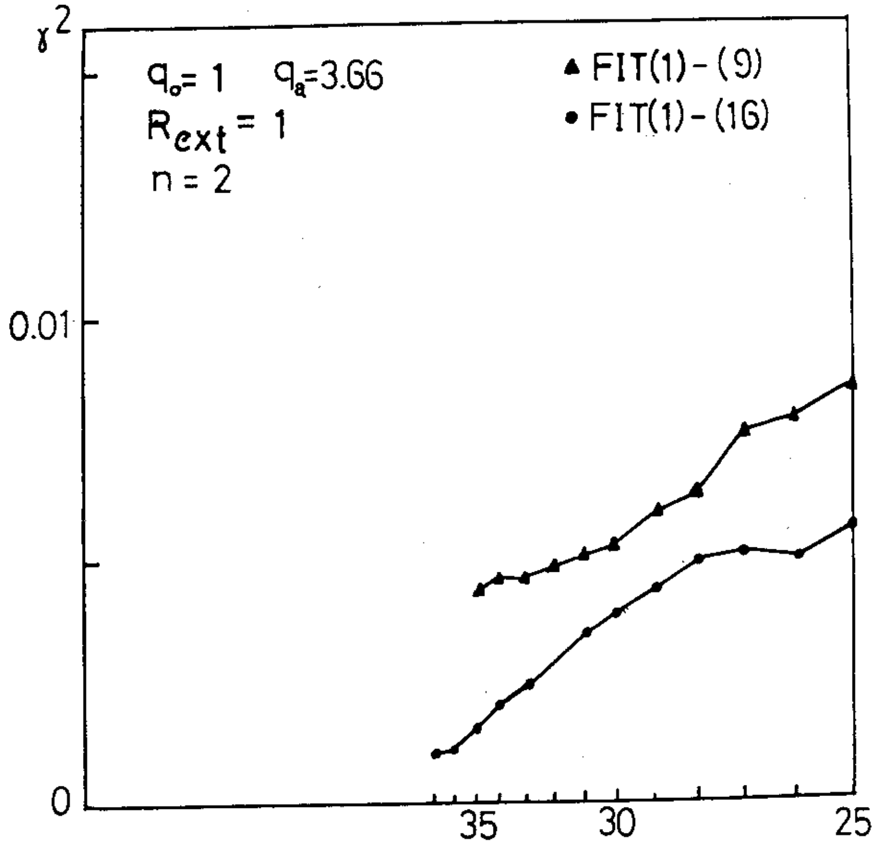


Fig. 3 The convergence curve of the fixed boundary mode.

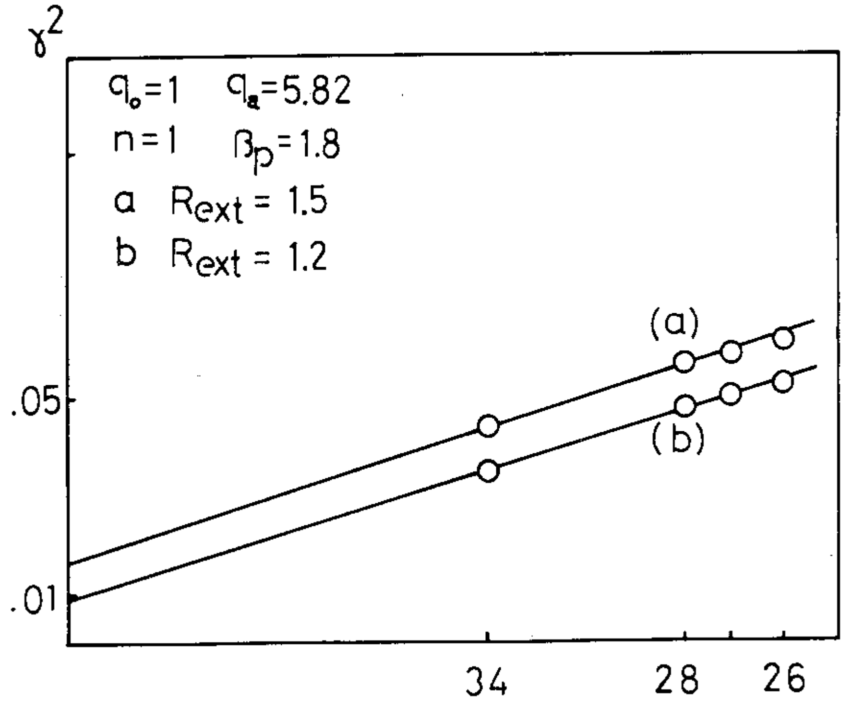


Fig. 4 The convergence curve of the free boundary mode with small growth rate

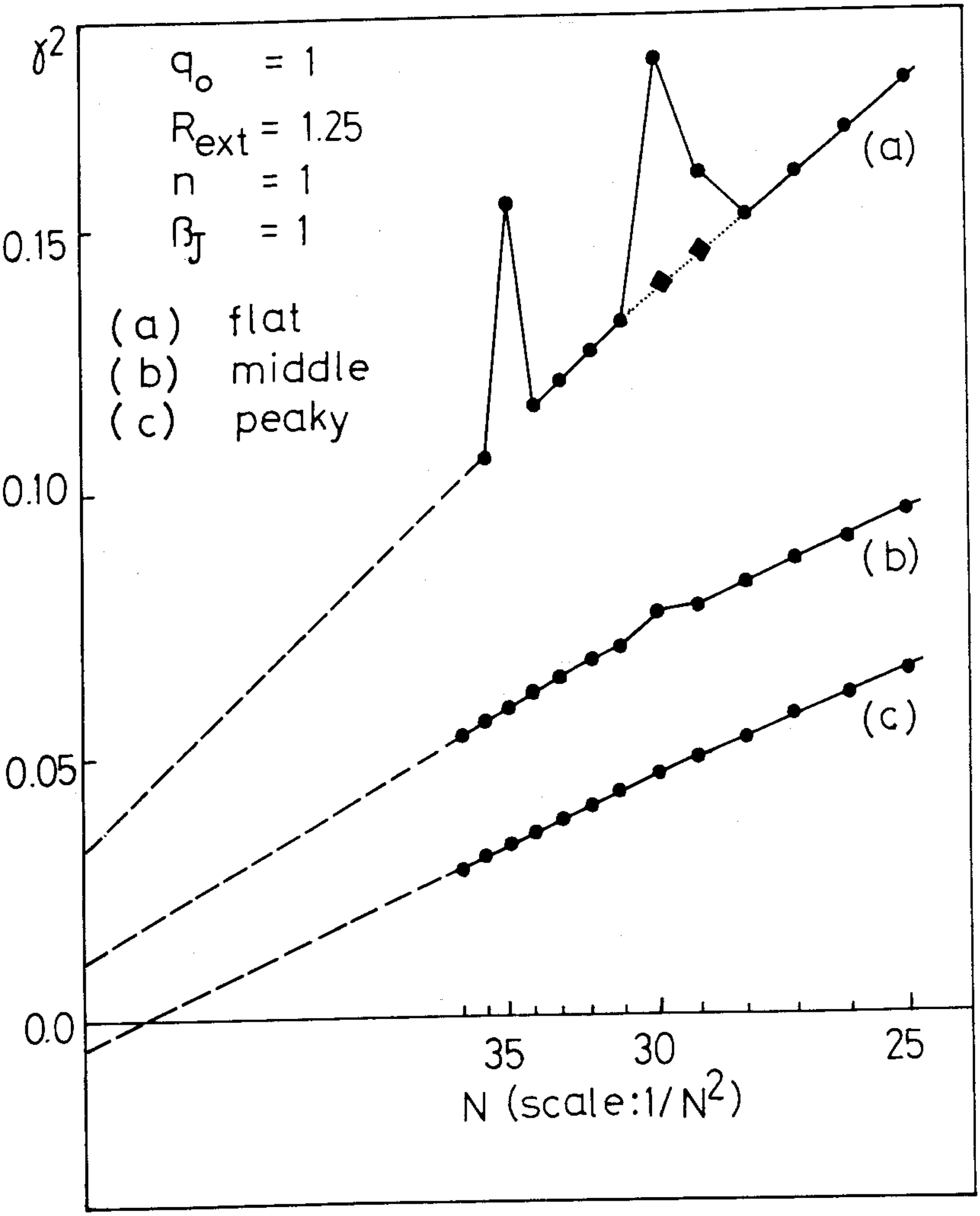


Fig. 5 Example of resonance-like phenomenon
The number of the equilibrium meshes is $N_r = 224$ and $N_z = 112$.
The mark ◆ shows the growth rate in the case of $N_r = 223$ and $N_z = 111$

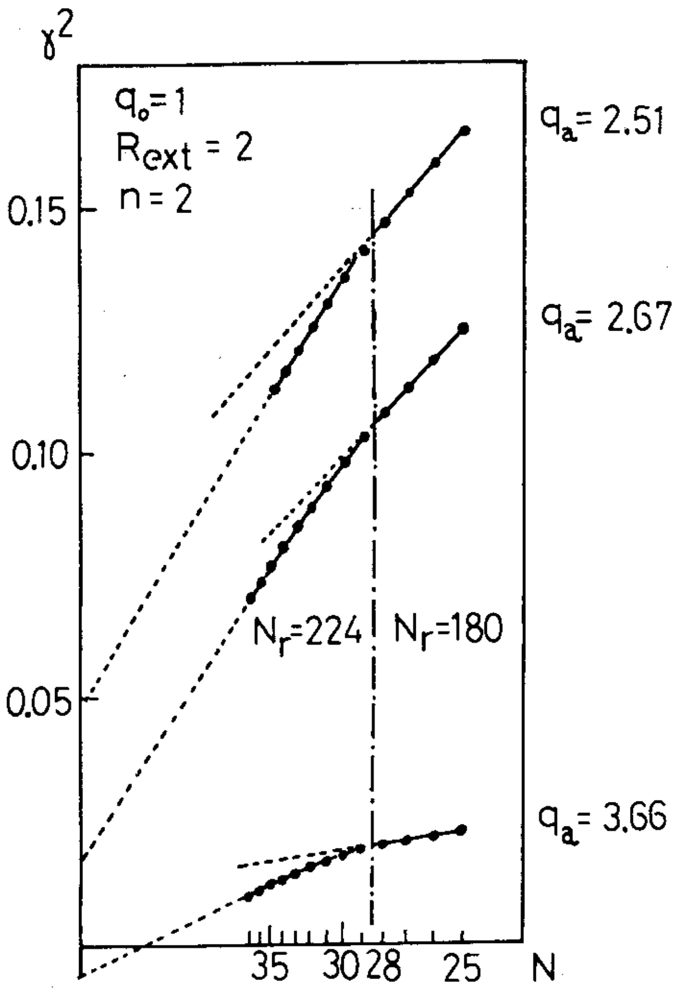


Fig. 6 Change of the convergence property due to the change of equilibrium mesh numbers

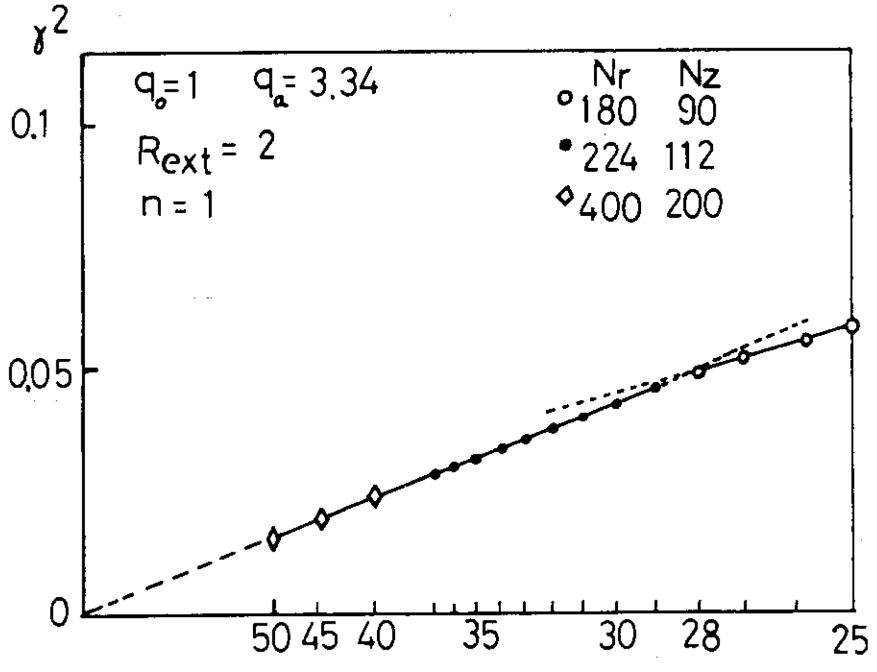


Fig. 7 Results of calculation with large mesh numbers.

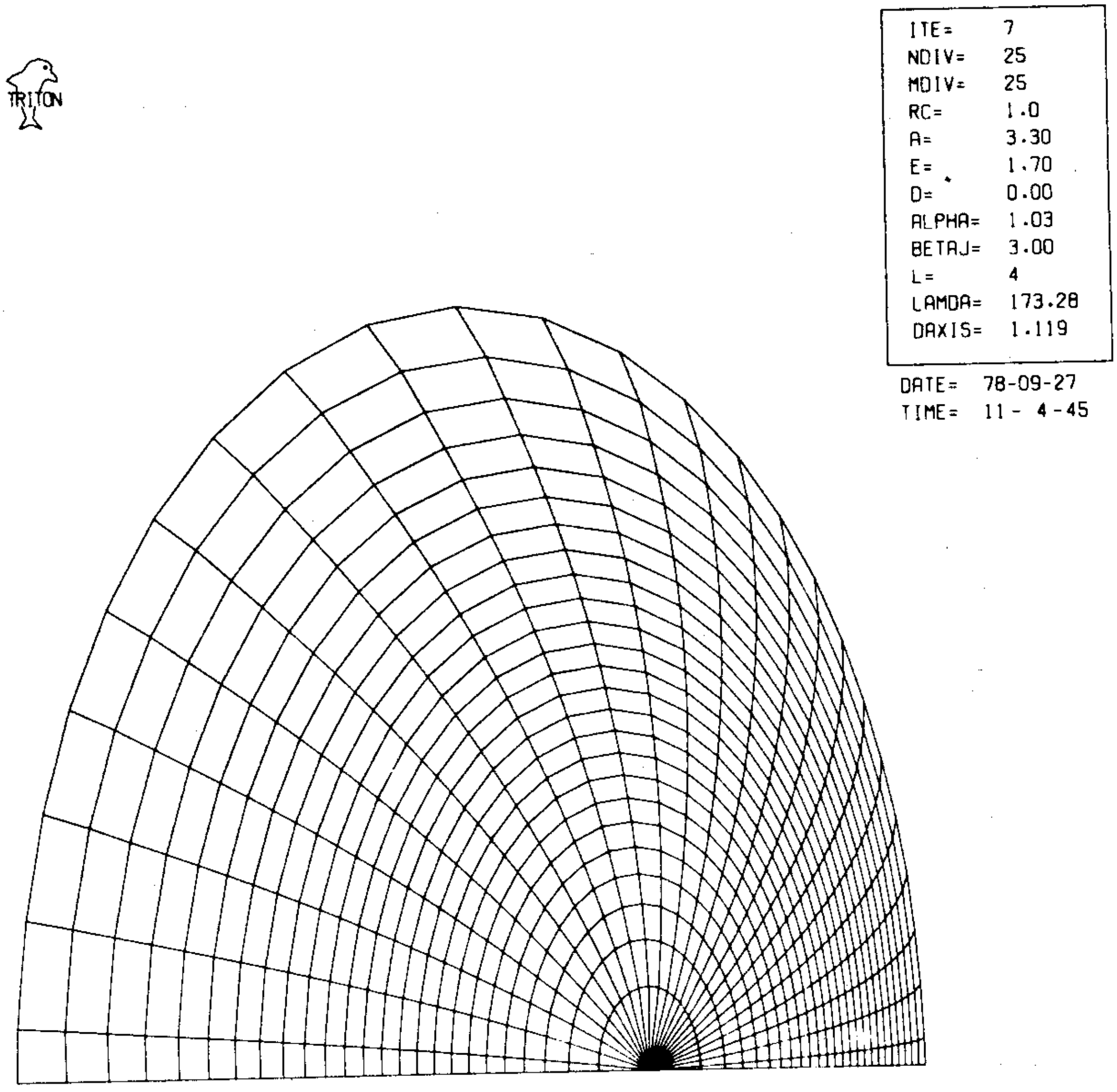


Fig. 8 The final meshes of the SELENE20, which represent the constant ψ lines.

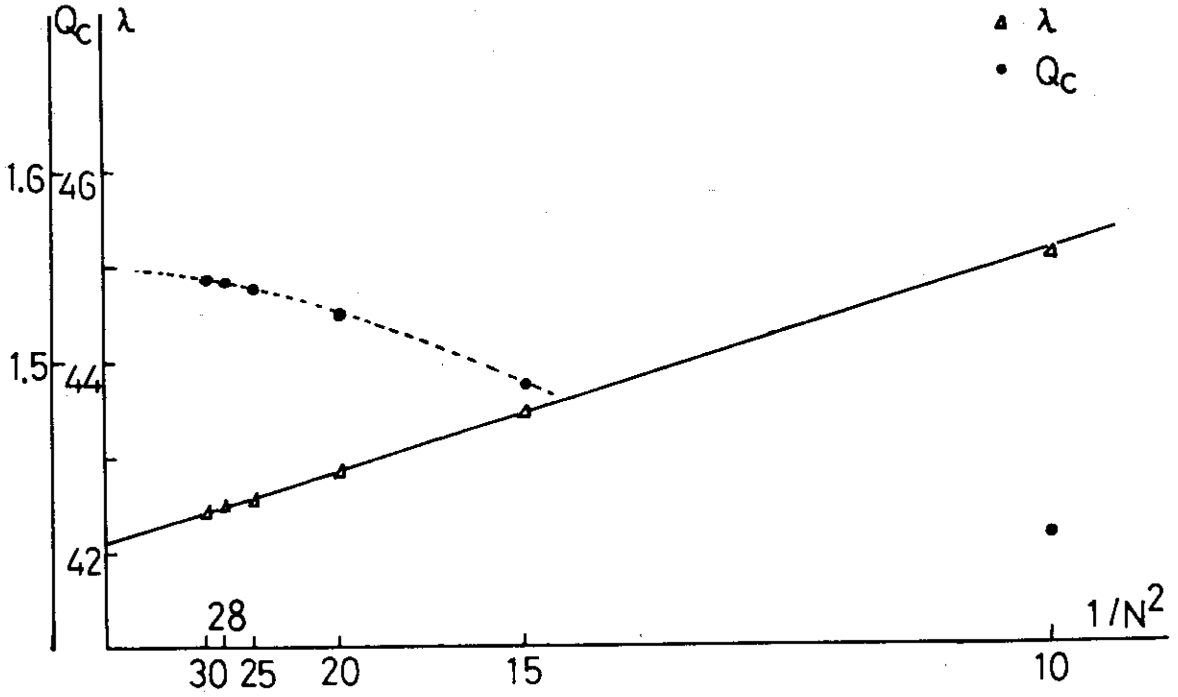


Fig. 9 The convergence of the eigenvalue λ versus the number of the division ($N = N_\psi = N_\chi$).

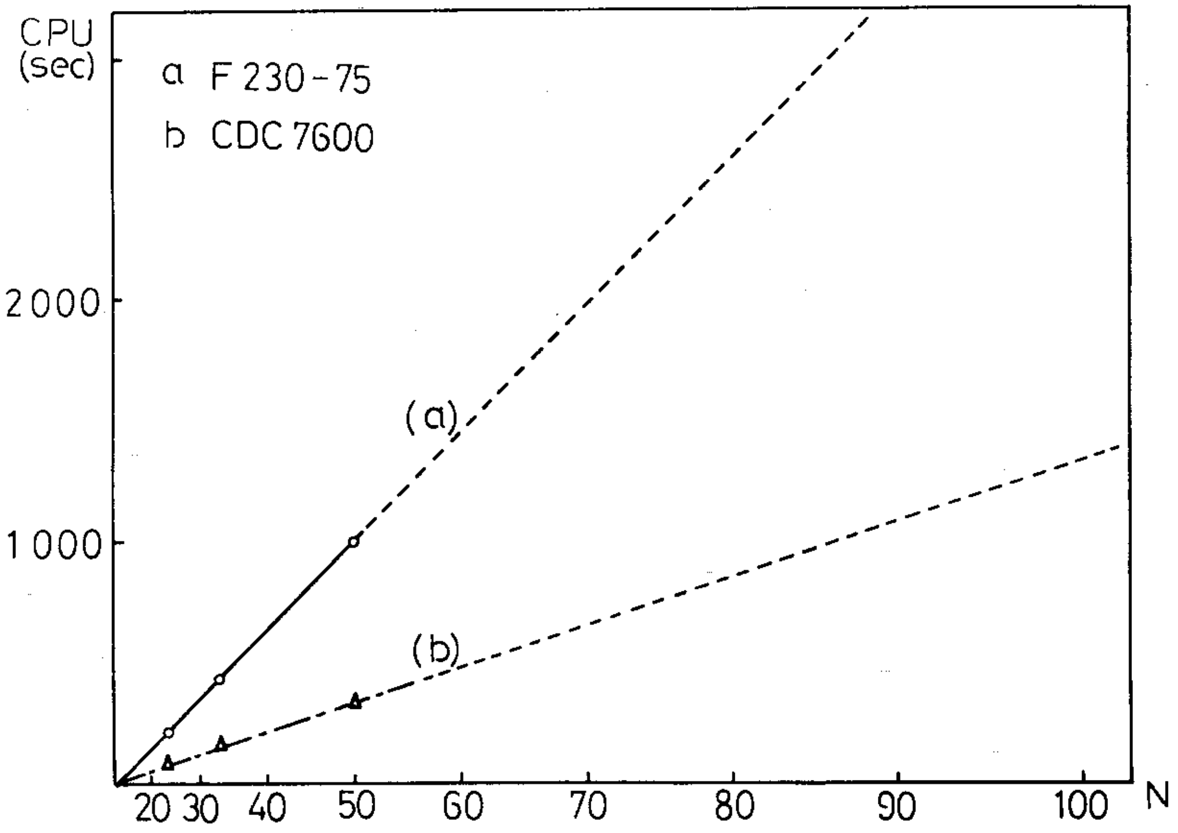


Fig.11 CPU time required in the FACOM230-75 versus the number of the division ($N = N_\psi = N_\chi$) (solid line) and the expected CPU time in the CDC7600 or FACOM M-200 (dotted solid line). The dotted lines denote the extrapolated ones.

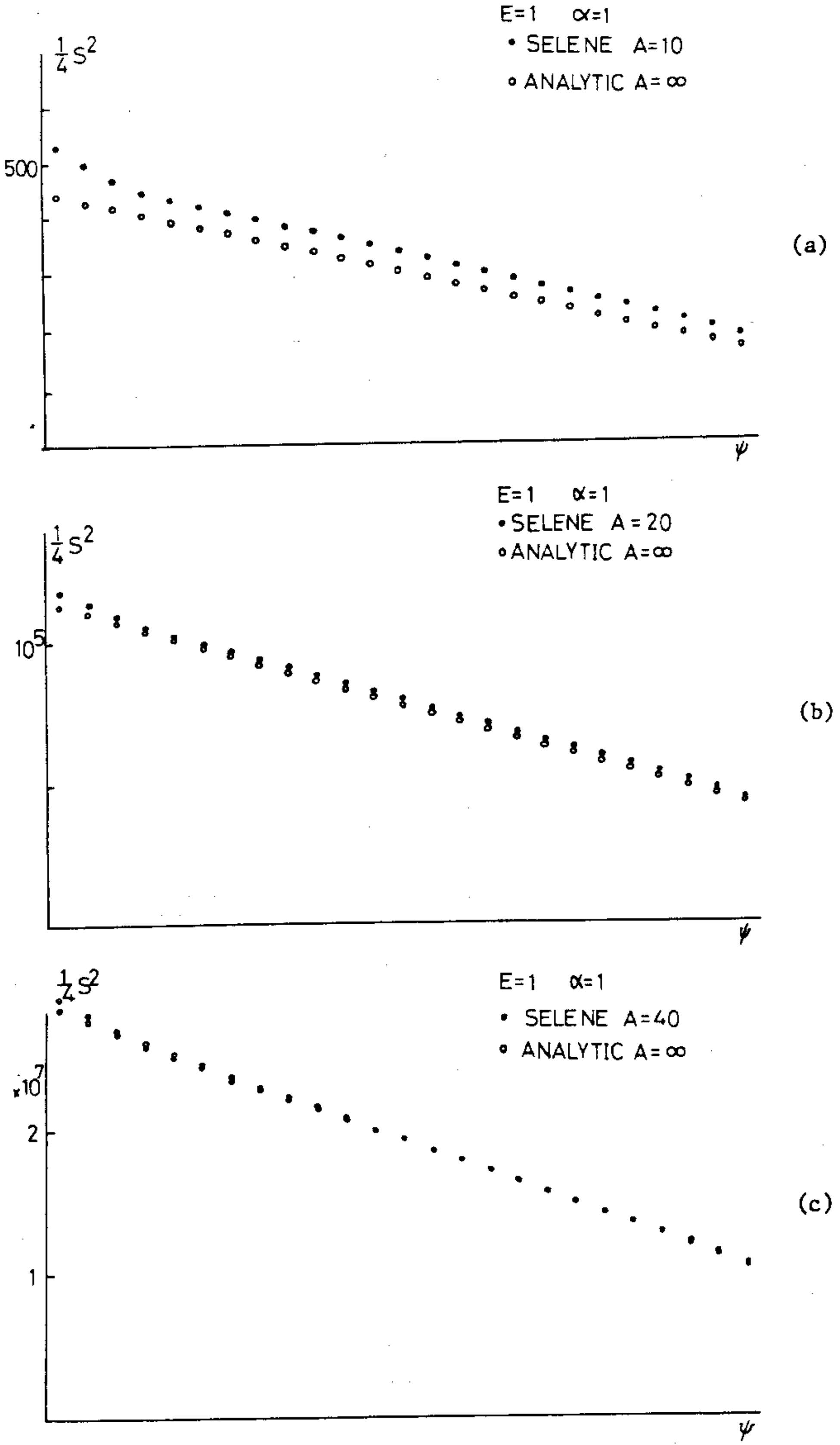


Fig.10 Magnetic shears of large aspect ratio plasmas (numerical) and a cylindrical plasmas (analytic).

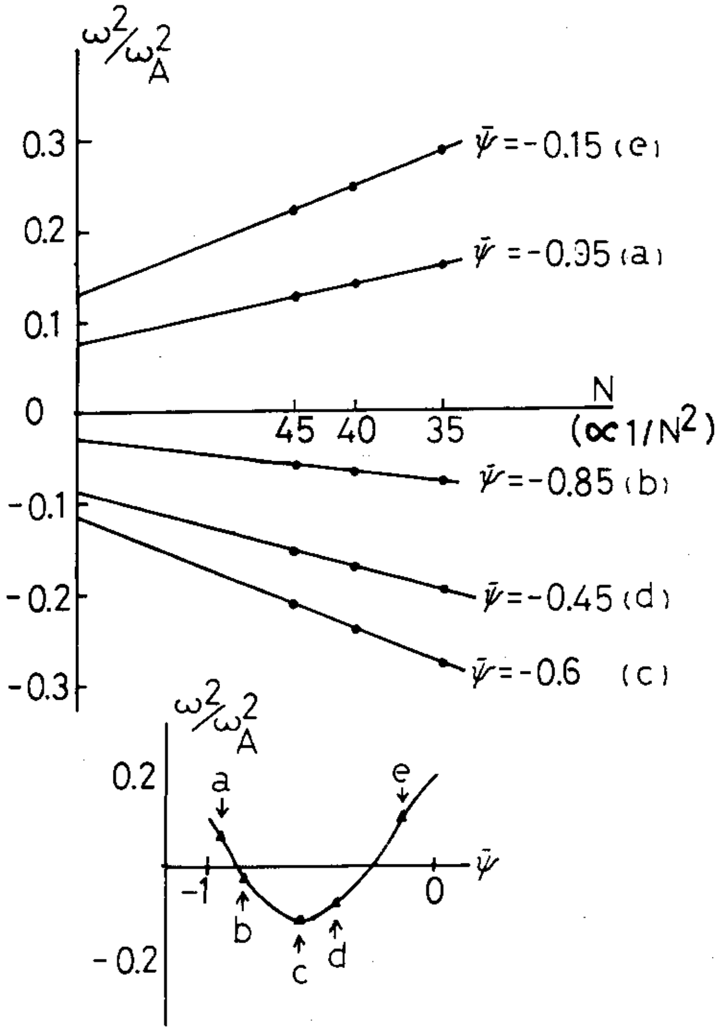


Fig.12 The convergence properties of the minimum eigenvalue in the BOREAS code.

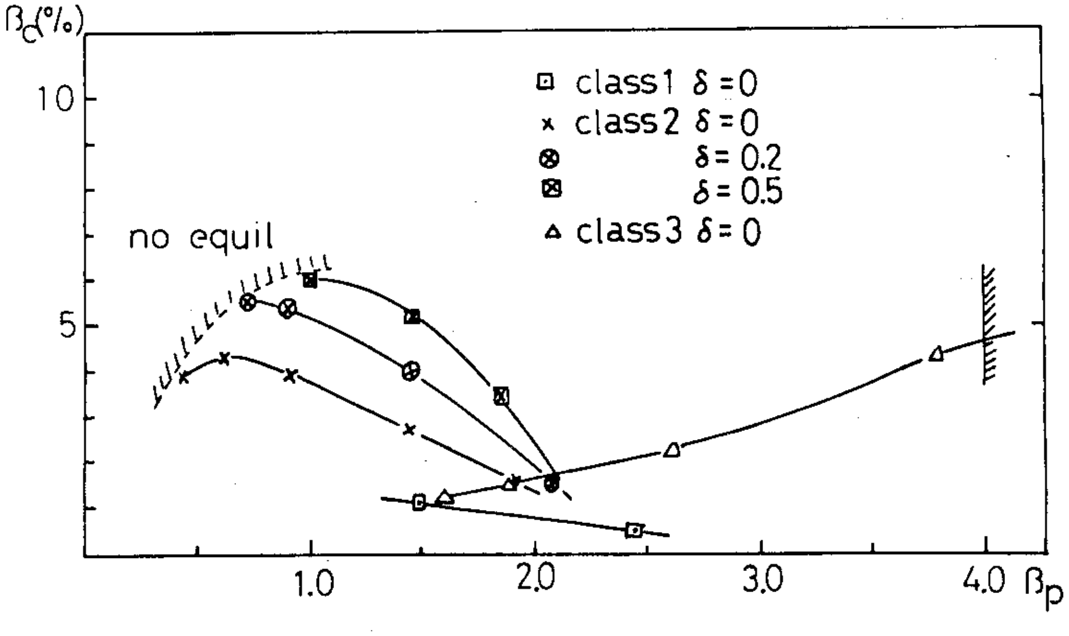


Fig.13 Critical beta values versus β_p. δ denotes the triangularity.

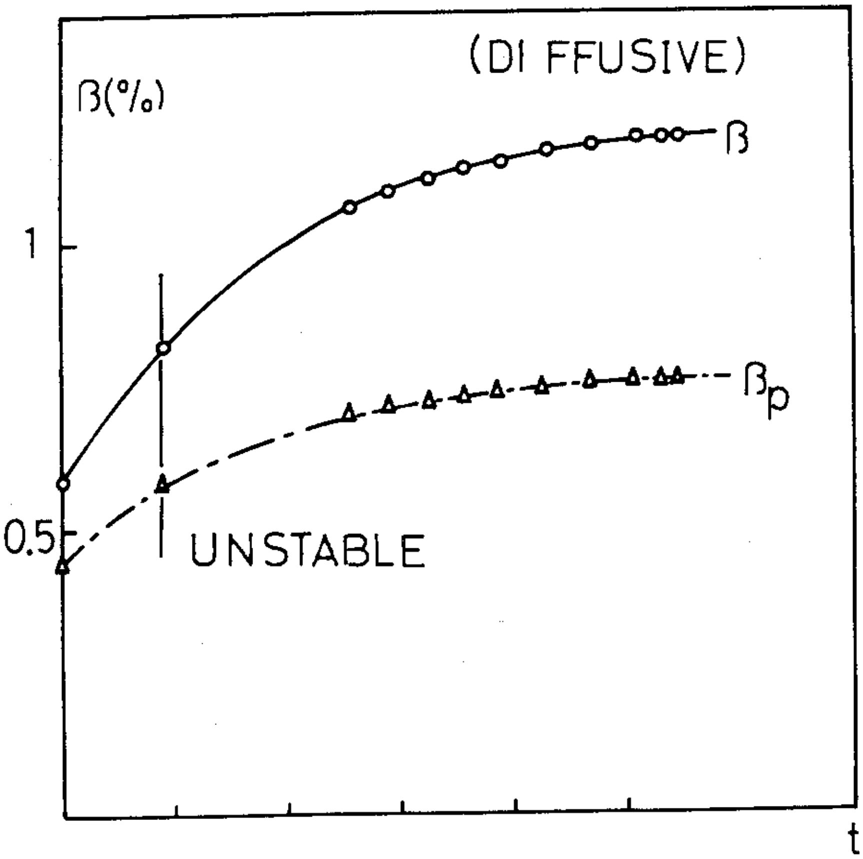


Fig.14 Time evolution of beta value obtained by using the APOLLO30 (including the simplified diffusion processes).

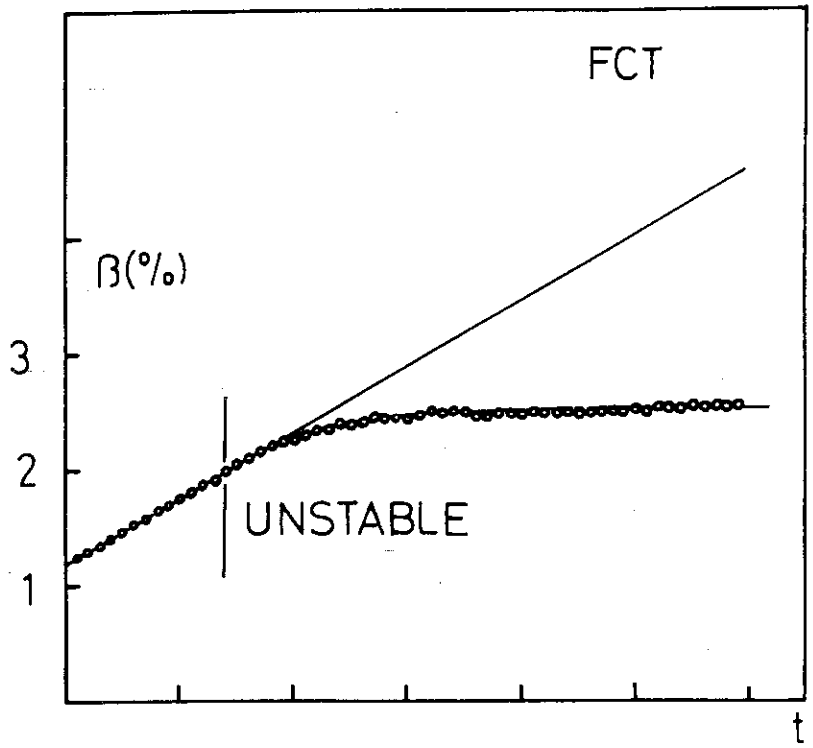


Fig.15 Time evolution of beta value obtained by using the APOLLO40 (FCT heating).

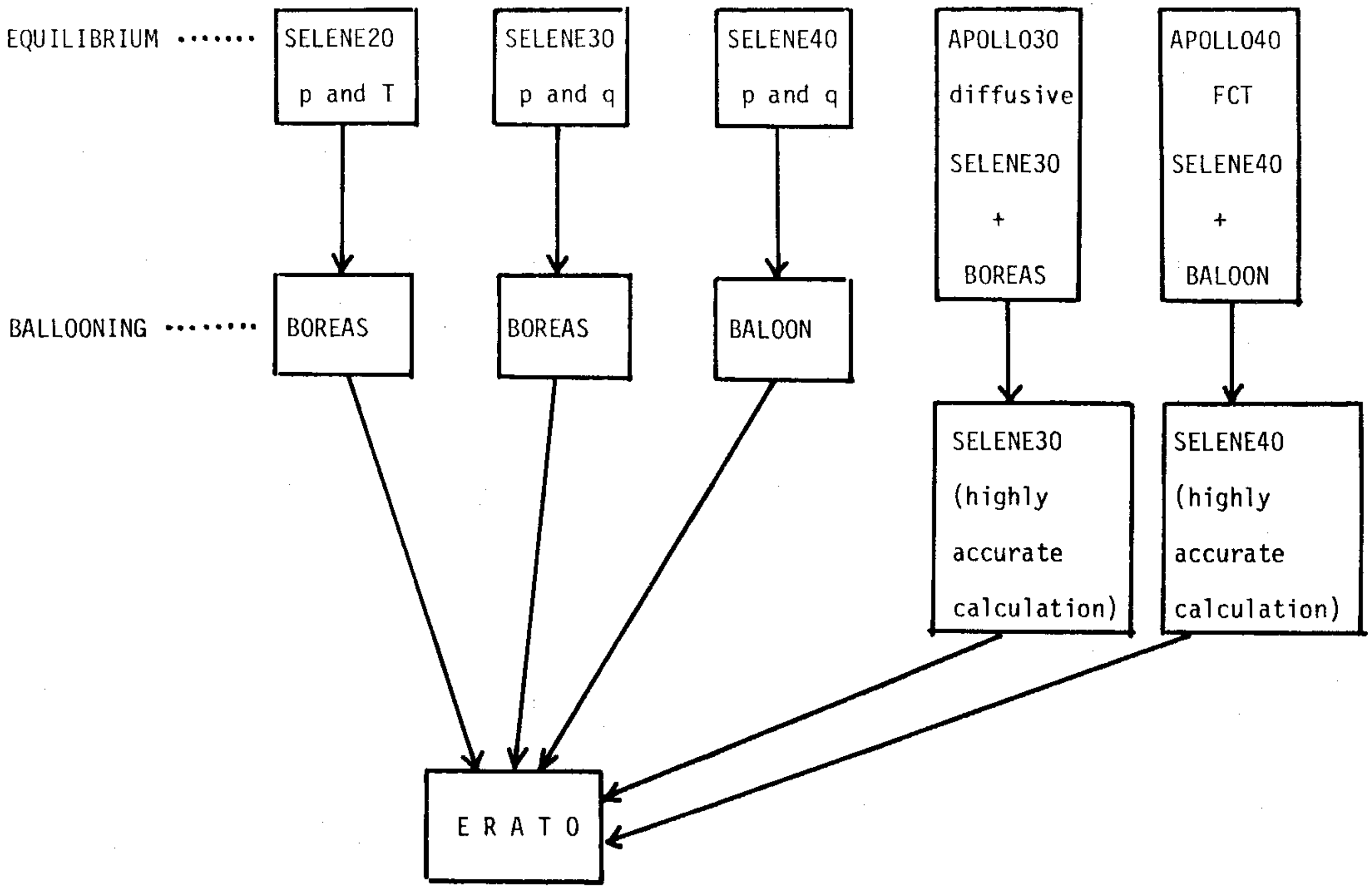


Fig.16 The relation between the SELENE, BOREAS and APOLLO and the ERATO.

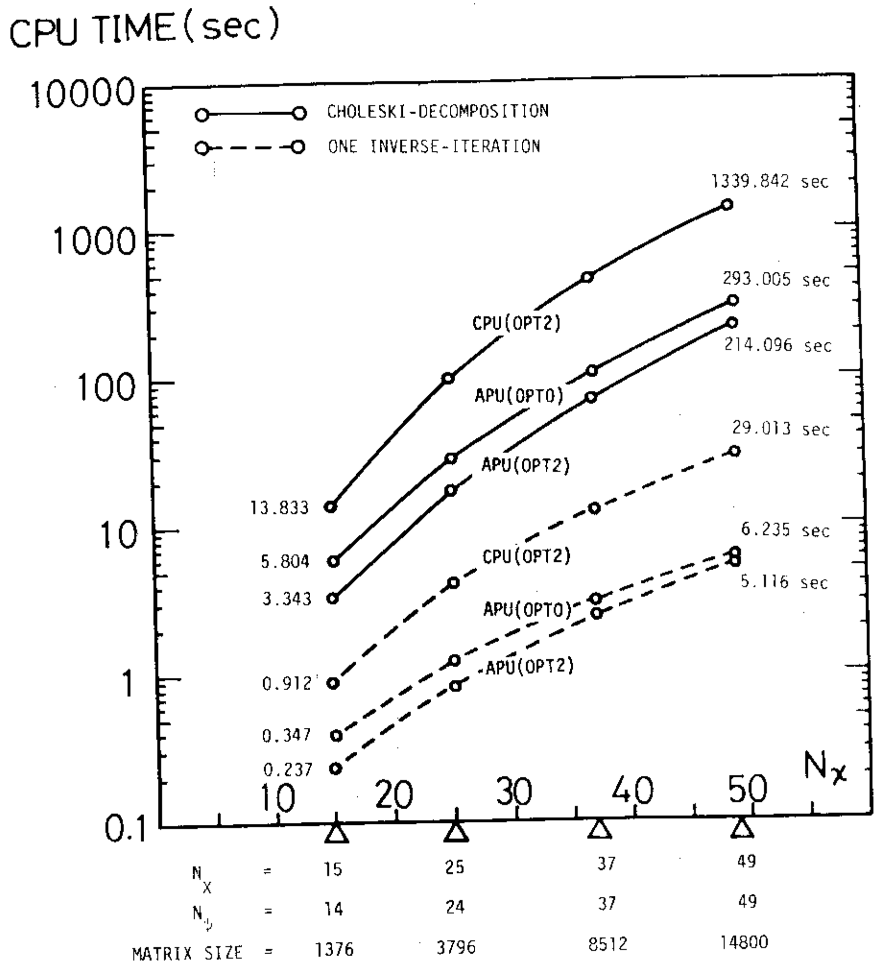


Fig.17(a) Execution time(CPU time) of Choleski-decomposition and One Inverse-iteration on FACOM 230-75 APU/CPU and 75CPU.

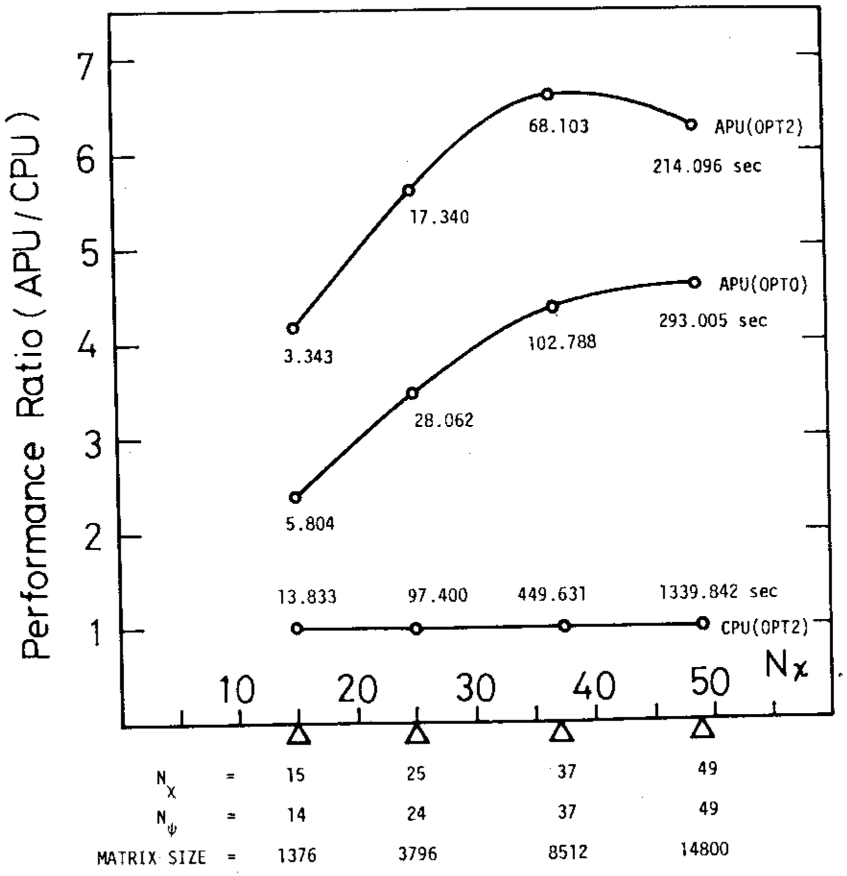


Fig.17(b) Performance ratio of APU/CPU for Choleski-decomposition.

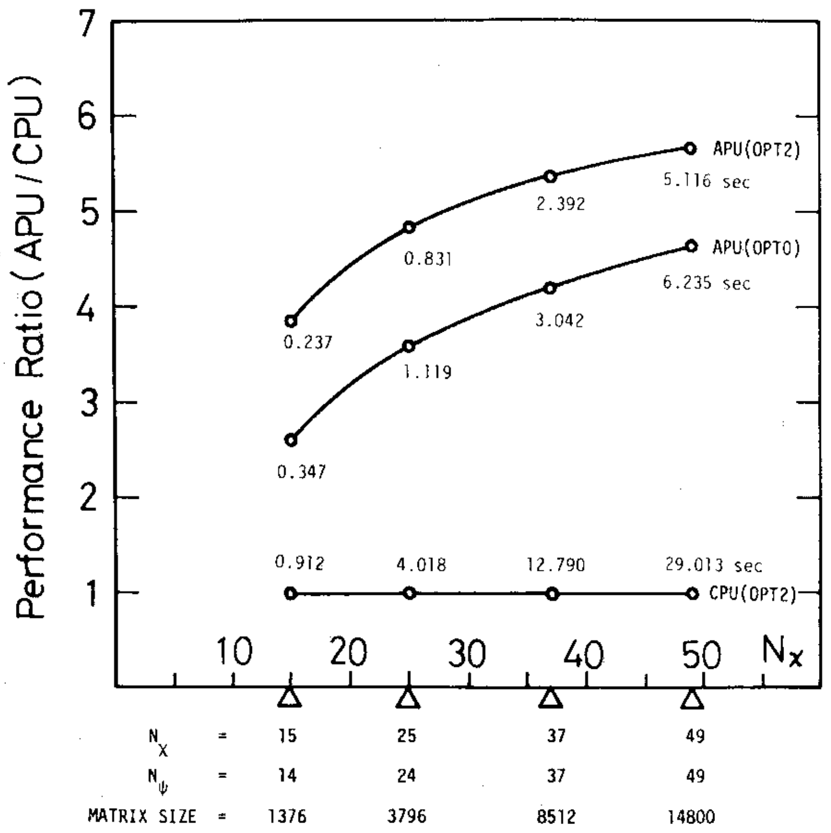


Fig.17(c) Performance ratio of APU/CPU for one Inverse-iteration.

Appendix A FACOM230-75 System and Adaptation of ERATO from CDC6600

A.1 The characteristics and performance of FACOM230-75 computer system are as follows,

CPU cycle time : 90 ns

Buffer memory	size : 2 or 4 kw access time : 90 ns/2 word
Core memory	size : 64 kw (min.)/1024 kw (max.) memory cycle time : 1 μs/2 word

Bit pattern : 40 bit (36 data bit + 4 flag bit)

Operation time

Fixed point	addition-subtraction	108 ns
	multiplication	450 ns
	division	2250 ns
Floating point	addition-subtraction	360 ns
	multiplication	540 ns
	division	1350 ns
Gibson mix.		267 ns

Operating System : MONITOR VII

Program language used for ERATO adaptation : FACOM FORTRAN IV

Compiler : FACOM FORTRAN-H

System configuration of FACOM230-75 in JAERI Computing Center is shown in Fig.A1.

A.2 Adaptation of ERATO from CDC6600 to FACOM230-75

In order to adapt ERATO from CDC6600 to FACOM230-75, the following alterations are necessary.

(1) Explicit initialization of local and common variables to zero.

(2) Matching of numbers of actual and dummy arguments of subroutines.

(3) Alteration of floating variables, constants and built-in functions to double precision.

(4) Adjustments of addressing shifts in equivalenced real and integer arrays caused by the alteration (3).

(5) Alterations concerning character handling in LABRUN (LABEL1 ~ LABEL8).

Appendix B Timing of Basic I/O Execution Processes (WRITE, READ, BACKSPACE)

Execution time of basic I/O processes is measured by using the following simplified I/O programs. Total amount of transferred data is 36.36 MB.

(a) Sequential File (FORTRAN I/O library)

```
      DIMENSION A(161600)
      DO 10 I = 1,50
      WRITE(1) A
   10 CONTINUE
      REWIND 1
      DO 20 I = 1,50
      READ(1) A
   20 CONTINUE
      DO 30 I = 1,50
      BACKSPACE
      READ(1) A
      BACKSPACE
   30 CONTINUE
```

		CPU Time	E Time*
WRITE	× 50	4.869	93.820
READ	× 50	4.378	96.328
BACKSPACE READ BACKSPACE	× 50	5.338	286.201

(b) Direct Access File (FORTRAN I/O library)

```
      DIMENSION A(161600)
      DEFINEFILE 1 (700,2894,U,IX)
      WRITE (1▽IX) DUMMY
      IX = 1
      DO 10 I = 1,50
      WRITE (1▽IX) A
   10 CONTINUE
      IX = 1
      DO 20 I = 1,50
      READ (1▽IX) A
   20 CONTINUE
```

		CPU Time	E Time*
File Open (Format WRITE)		0.562	95.660
WRITE	× 50	14.133	95.007
READ	× 50	3.639	95.672

(c) Sequential File (Assembler I/O Routine Programmed on EXCP-level)

```
      DIMENSION A(161600)
      DO 10 I = 1,50
      CALL ERTOUT (1,A,161600)
   10 CONTINUE
      CALL ERTRWD(1)
      DO 20 I = 1,50
      CALL ERTIN (1,A,161600)
   20 CONTINUE
```

		CPU Time	E Time*
WRITE	× 50	0.270	96.042
READ	× 50	0.052	54.773

* Elapsed Time

Appendix C FACOM230-75APU

The FACOM230-75 Array Processing Unit (APU) is a special purpose high-speed scientific computer with parallel processing architecture. It can perform up to 22 MFLOPS (Million Floating Operations per Second) for addition, inner product and summation. The ratio of gross processing speed to FACOM230-75 CPU are 5 to 10 times in average and 30 times at its maximum. The FACOM230-75APU employs the same device technology as the FACOM230-75CPU.

Characteristics of the FACOM230-75APU/CPU multiprocessor systems are as follows,

1) The CPU and APU share the main memory and form an asymmetric multiprocessor system.
2) The APU uses vector instructions to perform high-speed parallel processing of fully pipelined data access and arithmetic operations.
3) Various kinds of functions are made available by changing a descriptor of a vector instruction.
4) High speed data access is made possible by use of the 1,792 words vector register.
5) The APU also has high-speed scalar operation capabilities provided by the 2 kW buffer memory and 256 general registers.

The characteristics and performance of FACOM230-75APU are summarized in Table C1. System configuration of FACOM230-75APU/CPU in FUJITSU Computing Center (at Kawasaki, Japan) is shown in Fig.C1.

Performance ratio of APU/CPU for elementary operations in ERATO4 is shown in Fig.C2.

Appendix D Software systems of FACOM230-75APU

In this Appendix we describe APU MONITOR, AP-FORTRAN and APTRAN.

(1) The APU MONITOR is a system program belonging to the MONITOR VII. The APU MONITOR controls APU task independently from CPU unless communications between APU and CPU occur. Multiprogramming of APU jobs is made available by the APU MONITOR.

(2) The AP-FORTRAN provides extended FORTRAN functions to facilitate the vector operations. Examples of vector operations are shown in Table D1. Optimizations by the AP-FORTRAN are also shown in Table D2.

(3) The APTRAN is a debugging utility which translates a source program written in the AP-FORTRAN to that written in a standard FORTRAN.

Appendix E Optimization of the Subroutine CALD

In the subroutine CALD, matrix decomposition algorithm is altered from Gaussian elimination method to Choleski method. The essentials of these two methods are shown as follows,

(1) Decomposition by Gaussian elimination

$$L^{-1}A = DU \qquad \begin{cases} u_{ii} = 1 \\ u_{ij} = 0 \quad \text{for} \quad i > j \end{cases}$$

```
for  i = 1, n-1
    for j = i+1, n
        for  k = i+1, n
            $a_{kj}^{(new)} = a_{kj} - (a_{ki}/a_{ii})\cdot a_{ij}$
    $d_{ii} = a_{ii}^{(new)}$
    $u_{ij} = a_{ij}^{(new)} / d_{ii}$
```

(2) Modified Choleski-decomposition (Crout-method for Symmetric Matrix)

$$A = LDL^T \qquad \begin{cases} \ell_{ii} = 1 \\ \ell_{ij} = 0 \quad \text{for} \quad i < j \end{cases}$$

```
for  i = 1, n
    $\ell_{ii} = 1$
    $d_{ii} = a_{ii} - \sum_{k=1}^{i-1} \ell_{ik}^2 d_{kk}$
    for j = i+1, n
        $\ell_{ji} = (a_{ij} - \sum_{k=1}^{i-1} \ell_{ik} \ell_{jk} d_{kk}) / d_{ii}$
```

Execution times for matrix decomposition on CPU task are improved as

	(Gauss)		(Choleski)
case 1 :	18.916	→	15.032 (sec)
case 2 :	133.617	→	98.911

Table C1. The Characteristics and Performance of FACOM230-75APU

Machine Cycle (Basic clock for pipeline) :			90 ns
Elements	Highspeed CML (Current Mode Logic)		3 ns
	NTL (Non Threshold Logic)		1 ns
	IC memory	2 kinds 1.	14 ns
		2.	35 ns
Buffers	Instruction buffer		8 word
	Data buffer		256 kw × 3
	Buffer memory		2 kw
Registers	Data register		256
	Vector register		1,792 word
Operation pipeline	3 series	1. Addition pipeline	
		2. Multiplication pipeline	
		3. Logical operation pipeline	
Vector operation time (Floating point, single precision)	Addition	22	MFLOPS
	Multiplication	11	MFLOPS
	Division	1.2	MFLOPS
	Inner Product	22	MFLOPS
	Summation	22	MFLOPS

Table D1 Examples of vector operations

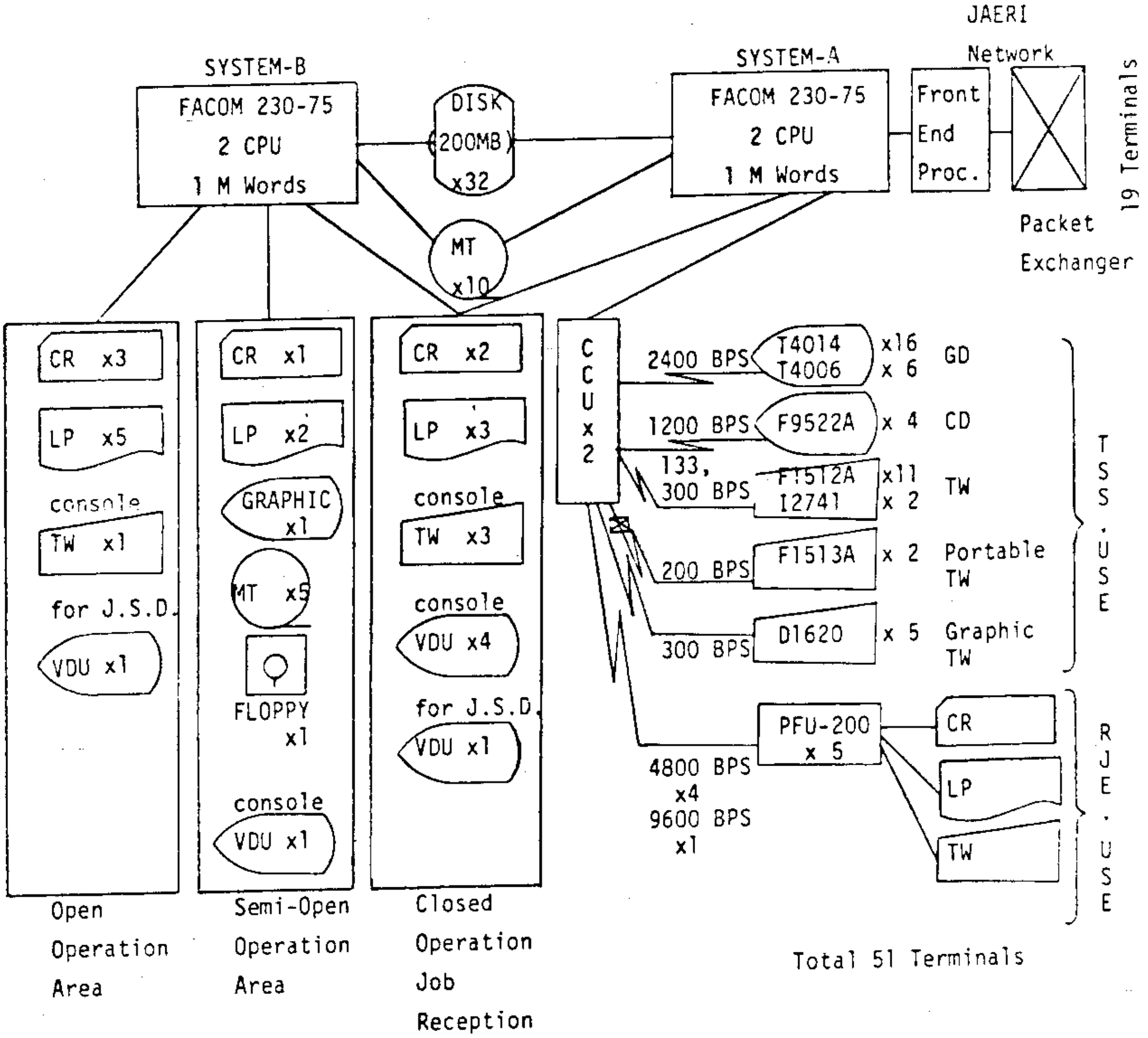
Instruction	Calling Procedure	Definition
Summation	S = VSUM(B(*))	$S = \sum_{i=1}^{m} b_i$
Inner Product	S = IPD(B(*),C(*))	$S = \sum_{i=1}^{m} (b_i,c_i)$
Average	A(*) = AVRG(B(*),C(*))	$a_i = \frac{1}{2} (b_i+c_i)$
Comvolution Product	A(*) = CVML(B(*),C(*))	$a = \sum_{j=1}^{m} (b_j,c_{i+j-1})$
Maximum	S = FMXV(B(*))	$S = \max(b_i)$ $1 \leqq i \leqq n$

Table D2 Optimizations by the AP-FORTRAN

Function of Optimization	Level	
	OPT0	OPT2
(a) Machine independent optimizations		
Adjustment of types of constants in a mixed operation	Yes	Yes
Inline expansion of a statement function	Yes	Yes
Alteration of powers to multiplication	Yes	Yes
Alteration of devision to multiplication	No	Yes
Optimization of a logical IF statement	Yes	Yes
Optimization of arrays in I/O statement	Yes	Yes
Common expression elimination	No	Yes
Invariant instruction movement	No	Yes
Induction variable optimization	No	Yes
Register saving optimization	No	Yes
(b) Machine dependent optimizations		
Descriptor optimization	No	Yes
Instruction scheduling	No	Yes
APU/CPU communication optimization	No	Yes
Others	Yes	Yes

Nov. 1979

Description of the Computing Facilities
of Japan Atomic Energy Research Institute



Comment: J.S.D. ..... Job Status Display

Fig.A1 System configuration of FACOM 230-75 in JAERI Computer Center.

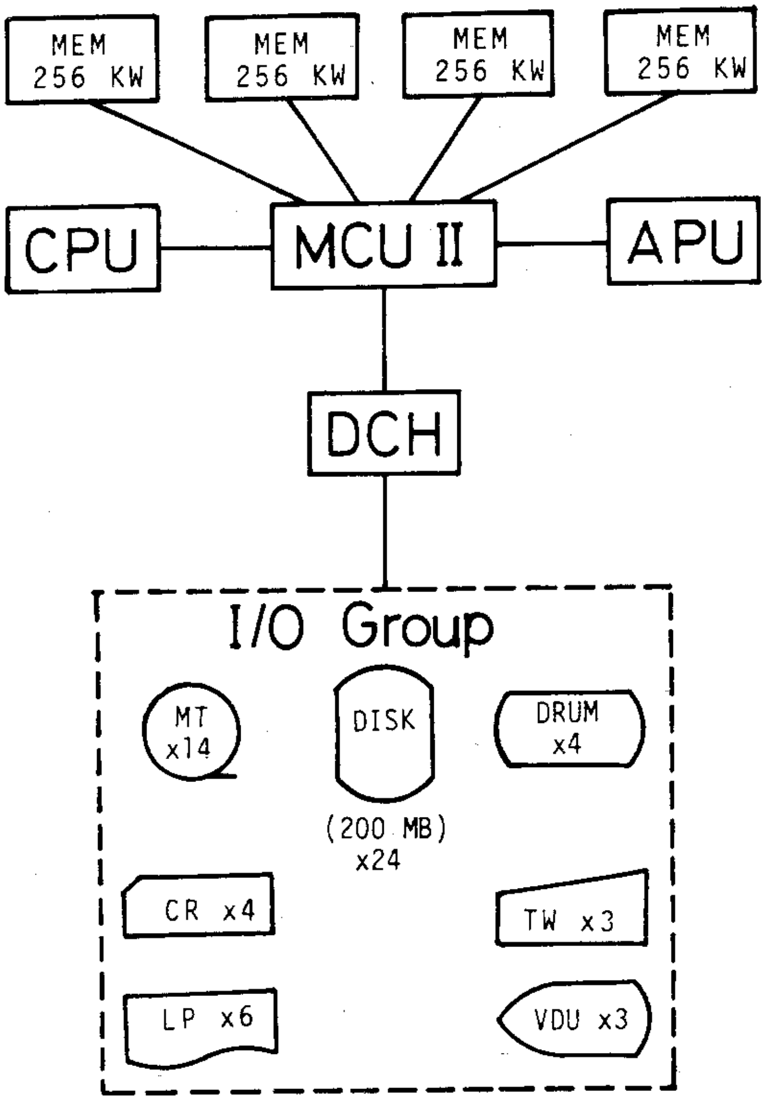
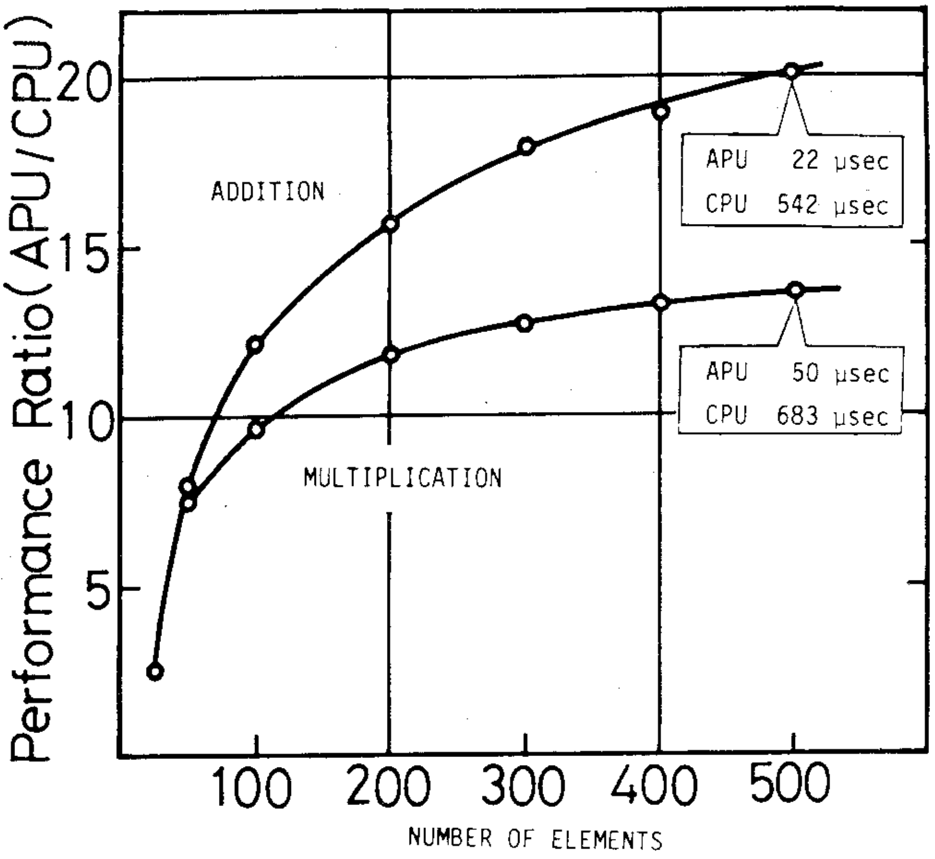
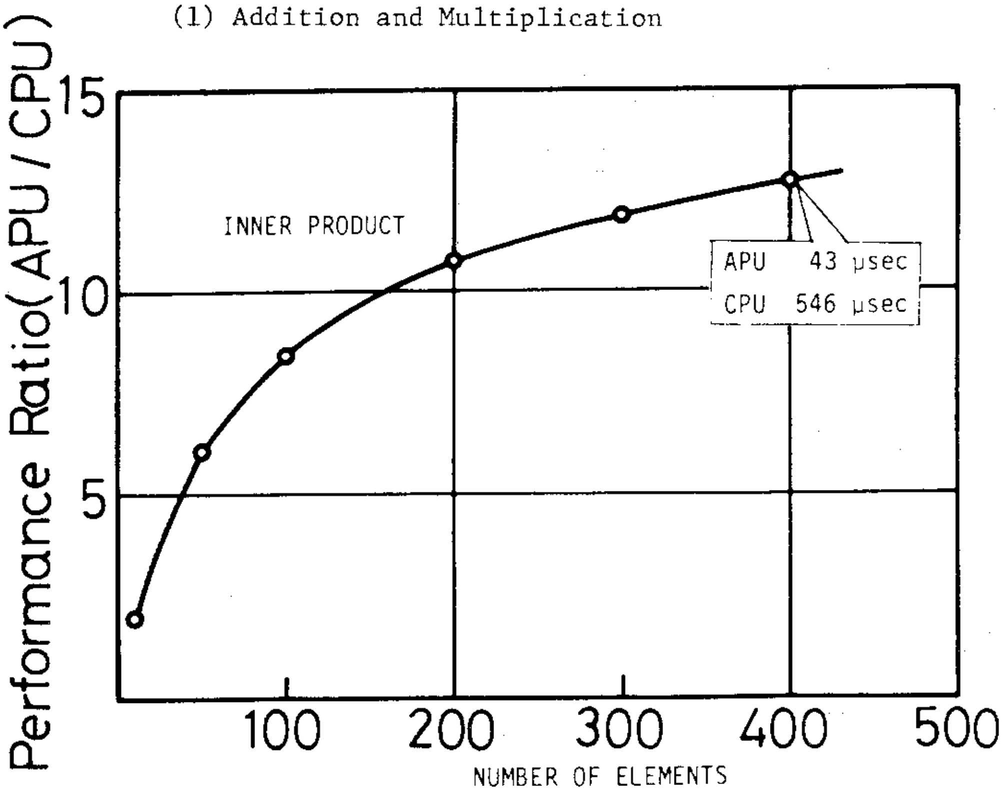


Fig.C1 System configuration of FACOM 230-75 APU/CPU in the FUJITSU Computing Center (at Kawasaki, Kanagawa-ken).



(1) Addition and Multiplication



(2) Inner Product

Fig.C2 Performance ratio of APU/CPU for elementary operations in ERATO4.