

JAERI-M

9040

MHD安定性解析コード ERATO-J

—原研版ERATO利用手引—

1980年8月

田中 幸夫^{*}・松浦 俊彦^{*}・竹田 辰興・安積 正史
徳田 伸二・常松 俊秀・栗田 源一・滝塚 知典

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.

MHD 安定性解析コード ERATO - J
-原研版 ERATO 利用手引-

日本原子力研究所東海研究所核融合研究部

田中幸夫*・松浦俊彦*・竹田辰興
安積正史・徳田伸二・常松俊秀
栗田源一・滝塚知典

(1980年7月31日受理)

ERATO コードを用いてMHD 安定性計算を行う時に必要な計算機システム資源の量を評価した。本報告書には、これらのデータ及びF 版ERATO - J を用いて一連の計算を行う際の手順が記されている。F 版ERATO - J は、原研計算センターのFACOM M 200 計算機システム用のERATO コードの最初のもので、R. Gruber 等によって開発された原版ERATO から変換されたものである。この版のERATO には、いくつかの小さな変更が加えられている。これらのうち、出力データの収集と検索を容易にするためのDIARY プログラムは、大量のERATO 計算を効率よく行う上で非常に有用である。

* 外来研究員；富士通（株）

JAERI-M 9040

MHD Stability Analysis Code ERATO-J
— User's Manual of JAERI-version ERATO —

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

Division of Thermonuclear Fusion Research,
Tokai Research Establishment, JAERI

(Received July 31, 1980)

Necessary resources of a computer system for the MHD stability calculations by the ERATO are estimated. In this report, these data and concrete procedure to carry out a series of calculations by using the ERATO-J(F-version) code are described. The ERATO-J(F-version) is the first version of the ERATO code for the FACOM M200 computer system of JAERI computer center, which was adapted from the original ERATO code developed by R. Gruber et al. In this version several minor changes were introduced. Among them the DIARY program which facilitates acquisition and sorting of the output data is very useful to carry out a large amount of the ERATO calculations efficiently.

Key words: MHD Stability Analysis, Tokamak Plasma, Finite Element Method, ERATO Code

* On leave from Fujitsu Ltd.

目 次

1. はじめに	1
2. 必要なシステム資源の評価	2
2.1 概 要	2
2.2 ERATO 4 の処理の特徴	2
2.3 ERATO 4 のプログラム・サイズの概算	7
2.4 ディスク・ファイル・サイズの概算	7
2.5 ERATO 4 のCPU時間の予測	12
2.6 I/O アクセス回数の概算	12
2.7 I/O 時間の概算	18
2.8 経過時間の予測	18
2.9 不等メッシュ ($N_\phi \neq N_\chi$) の場合のシステム資源必要量	20
3. ERATO コード使用手順	25
3.1 概 要	25
3.2 新規プログラムの説明	26
3.2.1 DIARY プログラム	26
3.2.2 MTOUT および MTIN プログラム	34
3.2.3 ERATO 6 プログラム	44
3.3 ロード・モジュールの作成	45
3.4 ディスクと磁気テープの初期化	56
3.4.1 ディスクの初期化	56
3.4.2 磁気テープの初期化	57
3.5 ERATO 計算の実行	57
3.6 DIARY FILE の参照	64
3.6.1 PRINT 0 および PRINT 1 プログラム	64
3.6.2 CHANGE プログラム	66
3.6.3 ERTG プログラム	69
3.7 磁気テープに保管されている情報の利用	74
4. ま と め	85
謝 辞	86

Contents

1.	Introduction	1
2.	Estimation of necessary system resources	2
2.1	Introduction	2
2.2	Feature of the ERATO4 calculations	2
2.3	Program size of the ERATO4	7
2.4	Size of disk memory	7
2.5	CPU time for the ERATO4 calculations	12
2.6	Number of I/O accesses	12
2.7	I/O time for the ERATO4 calculations	18
2.8	Expected elapsed time	18
2.9	Necessary system resources for calculations with the unequal mesh numbers ($N_\psi=N_X$)	20
3.	Procedure to carry out a series of ERATO calculations	25
3.1	Introduction	25
3.2	New subroutines	26
3.2.1	DIARY program	26
3.2.2	MTOUT and MTIN program	34
3.2.3	ERATO6 program	44
3.3	Generation of load modules	45
3.4	Initialization of disks and magnetic tape	56
3.4.1	Initialization of disks	56
3.4.2	Initialization of a magnetic tape	57
3.5	JCL programs for ERATO calculations	57
3.6	Use of the DIARY FILE	64
3.6.1	PRINT0 and PRINT1 programs	64
3.6.2	CHANGE program	66
3.6.3	ERTG program	69
3.7	Use of data stored in the magnetic tape	74
4.	Summary	85
	Acknowledgments	86

1. はじめに

ERATO コードは、有限要素法を用いて、軸対称トーラス・プラズマの線形 MHD 安定性を解析する計算コードである。このコードは、最初 Gruber 等¹⁾ によって開発されたが、その有用性が認められて世界各地の研究機関で利用されるようになり、また改良や拡張も行われている。原研においても、昭和 53 年 3 月に、Gruber によって、このコードの導入が行われ、利用が可能になった。導入後、ERATO コードは JT - 60 の安定性解析²⁾、INTOR の安定性解析³⁾、高ベータ平衡の安定性解析⁴⁾ に用いられ、現在に至っている。これらの安定性解析を通じて、ERATO コードの解の収束性についての知見を得る一方、コードに含まれているいくつかの問題点を見出した。

第 1 の問題点は、解の収束曲線上にあらわれる小規模な不規則性や共鳴的現象および固有関数の不自然な振舞いの形で現われるもの⁵⁾ で原因は計算メッシュ上でのメトリック量の評価の不正確さに帰因するものと推定される。計算点を多くすれば精度の高い収束点を求めることができるが全体の解析の効率は著しく低下する。この問題を取り除くために、現在、その改良法について検討が進められている。

第 2 の問題点は、我々に利用可能な計算機システムでこのコードを実行する場合、ディスク入出力の速さに基づいて経過時間が極端に長くなるという事である。この問題に関しては、演算速度の高速化の問題とも併せて詳しく調べられた。⁶⁾

第 3 の問題点は、上記 2 つの問題点に比べるとかなり性格の異なるもので、一連の安定性解析を行う上で遭遇する実際的問題である。通常 ERATO コードの 1 ケースの計算量は必ずしも大きなものとは言えない（第 2 章参照）。実際的な計算の場合、FACOM M 200 によって、CPU 時間が数分ないし数 10 分、経過時間が数 10 分ないし数時間の範囲におさまるのが普通である。しかし、このコードの結果から物理的に意味のあるデータを得るためにメッシュ数に関する収束計算を行い、メッシュ数が無限大となる極限値を外挿する必要がある。この収束計算を行い、一点の意味のあるデータを求める為には、現在の ERATO で、5 ケースないし 10 ケースの計算が必要となる。この為、いくつかの物理パラメータについて MHD 平衡を計算し、それぞれの平衡についてトロイダル・モード数の違い、導体壁の位置の違い等について場合分けをして安定性解析を行おうとするならば、直ちに数 100 ケースから 1000 ケース程度の計算が必要になる。このように膨大な数の計算を間違いなく、効率的に行う為には、計算手順を十分明確にし、出力データの保管、編集等を容易に行えるようにコードを整備しておく必要がある。原著者による ERATO コードおよび利用手引書⁷⁾ においては、これらの点は、はぶかれており、むしろ、個々の利用者がそれぞれの計算機システムに応じて対策をたてるべきものと考えられる。原研計算センター用に改良された FACOM M-200 版 ERATO (ERATO-J (F 版)) では出力データの保管、編集等を容易にする目的で ERATO 5 を改良し、ERATO 6 を追加してある。

本報告書においては、ERATO-J (F 版) による計算を実際に、原研計算センターの FACOM M 200 計算機システムで実行し、安定性解析を行う上で必要な実用的な知識をまとめ

である。MHD 安定性の理論、ERATO で採用している数値解析法等については触れていない。これらについては別の文献に詳しい。また、現在改良中の ERATO 1, 2, 3, 4 についても別に述べる。第 2 章においては、必要なシステム資源の量についての評価を行い、第 3 章においては、新規に追加したプログラムの説明の他実際に計算を行う上で必要な手順や JCL プログラム等についての説明を行う。また、ERATO に関して予定されている今後の改良、拡張作業については第 4 章でまとめる。

2. 必要なシステム資源の評価

2.1 概 要

計算センターの与えられた計算機環境（運用条件）の下で、ERATO コードを効率良く実行処理するためには、まず、ERATO 計算で要求されるシステム資源の量を知る必要がある。計算機システムへの負荷（CPU 時間、メモリサイズ、ディスク・ファイル量等）の大きさの一例は第 2-1 表に示されている。ERATO コードの物理、数値解法、構成等については別の文献^{1) 7)} に詳しいので、本報告書では、各ジョブステップの内容、計算の流れについて、第 2-1 図に図示するだけに止める。ERATO ジョブは、大量のディスク I/O 処理を伴う大型ジョブであるために、多重プログラミング処理中に他のジョブと資源要求の競合を生じると、実行処理の経過時間が悪化し勝ち（長びく）である。従って、ERATO の実行にあたっては、システム資源要求量を必要最小限に見積るとともに、複数の ERATO ジョブ同志の資源競合を避けるために、ジョブ投入のスケジューリングを行うべきである。

ところで、ERATO ジョブ全体の必要資源量を支配しているのは、第 2-1 表からも分るとおり、ERATO 4 である。これは有限要素法に基づいて作られた大次元行列の一般固有値問題の解法ルーチンであり、その計算規模は有限要素法のメッシュ数、 N_ψ 、 N_χ 及び、ベクトル逆反復法の反復回数（itr）に依って決まっている。従って ERATO の計算実行に必要なシステム資源について評価するには、ERATO 4 の必要資源量を詳しく調べれば十分である。以下に、ERATO 4 の処理の特徴とプログラムサイズ、ディスクファイル量、CPU 時間、I/O 時間、I/O アクセス回数、及び経過時間の評価結果を与える。

2.2 ERATO 4 の処理の特徴

ERATO 4 は、一般固有値問題 ($Ax = \lambda Bx$) の最小固有値及び固有ベクトルを、ベクトル逆反復法⁸⁾ (Inverse - Vector - Iteration Method) に依り求める。行列のサイズとパターンを第 2-2 図に示す。A, B ともブロック対角の実対称行列であり、(B は正定値)、行列データ量が大きいために、1 ブロック毎にディスク・ファイルと主記憶との間で転送を行ながら計算処理が行われる。行列データの I/O 処理に注目したフローチャートを第 2-3 図に与える。反復ループ中の後退代入は、行列 $D^{-1}L^T$ のブロック番号について逆順のアクセスを必要とするた

である。MHD 安定性の理論、ERATO で採用している数値解析法等については触れていない。これらについては別の文献に詳しい。また、現在改良中の ERATO 1, 2, 3, 4 についても別に述べる。第 2 章においては、必要なシステム資源の量についての評価を行い、第 3 章においては、新規に追加したプログラムの説明の他実際に計算を行う上で必要な手順や JCL プログラム等についての説明を行う。また、ERATO に関して予定されている今後の改良、拡張作業については第 4 章でまとめる。

2. 必要なシステム資源の評価

2.1 概 要

計算センターの与えられた計算機環境（運用条件）の下で、ERATO コードを効率良く実行処理するためには、まず、ERATO 計算で要求されるシステム資源の量を知る必要がある。計算機システムへの負荷（CPU 時間、メモリサイズ、ディスク・ファイル量等）の大きさの一例は第 2-1 表に示されている。ERATO コードの物理、数値解法、構成等については別の文献^{1) 7)}に詳しいので、本報告書では、各ジョブステップの内容、計算の流れについて、第 2-1 図に図示するだけに止める。ERATO ジョブは、大量のディスク I/O 処理を伴う大型ジョブであるために、多重プログラミング処理中に他のジョブと資源要求の競合を生じると、実行処理の経過時間が悪化し勝ち（長びく）である。従って、ERATO の実行にあたっては、システム資源要求量を必要最小限に見積るとともに、複数の ERATO ジョブ同志の資源競合を避けるために、ジョブ投入のスケジューリングを行うべきである。

ところで、ERATO ジョブ全体の必要資源量を支配しているのは、第 2-1 表からも分るとおり、ERATO 4 である。これは有限要素法に基づいて作られた大次元行列の一般固有値問題の解法ルーチンであり、その計算規模は有限要素法のメッシュ数、 N_ψ 、 N_χ 及び、ベクトル逆反復法の反復回数（itr）に依って決まっている。従って ERATO の計算実行に必要なシステム資源について評価するには、ERATO 4 の必要資源量を詳しく調べれば十分である。以下に、ERATO 4 の処理の特徴とプログラムサイズ、ディスクファイル量、CPU 時間、I/O 時間、I/O アクセス回数、及び経過時間の評価結果を与える。

2.2 ERATO 4 の処理の特徴

ERATO 4 は、一般固有値問題 ($Ax = \lambda Bx$) の最小固有値及び固有ベクトルを、ベクトル逆反復法⁸⁾ (Inverse - Vector - Iteration Method) に依り求める。行列のサイズとパターンを第 2-2 図に示す。A, B ともブロック対角の実対称行列であり、(B は正定値)、行列データ量が大きいために、1 ブロック毎にディスク・ファイルと主記憶との間で転送を行いながら計算処理が行われる。行列データの I/O 処理に注目したフローチャートを第 2-3 図に与える。反復ループ中の後退代入は、行列 DL^T のブロック番号について逆順のアクセスを必要とするた

Table 2-1. Example of program size, CPU time, and elapsed time for an ERATO calculation.

(a) Example of an equilibrium calculation⁴⁾.

APLD128F is the initialization code. FCT128 and FCT256A are both the FCT equilibrium codes. The former is for the calculation with meshes 128×128 and the latter for the calculation with the meshes 256×256.

(b) Example of a stability calculation⁴⁾. In this case, size of the disk file is ~ 100 CYL \times 3 ~ 165 MB and number of I/O accesses is about 61000.

(a) Equilibrium Calculation

Job Step	Program Size	CPU Time	Elapsed Time
APLD128F	832 KB	103.94 sec	4 min
FCTD128	896	13.64	< 1
FCTD256A	960	72.41	2

(b) Stability Analysis

Job Step	Program Size	CPU Time	Elapsed Time
ERATO1	960 KB	11.24 sec	< 1 min
ERATO2	1088	74.18	2
ERATO3	2816	28.79	3
ERATO4	3328	1117.69	69
MTOUT	1152	0.36	1

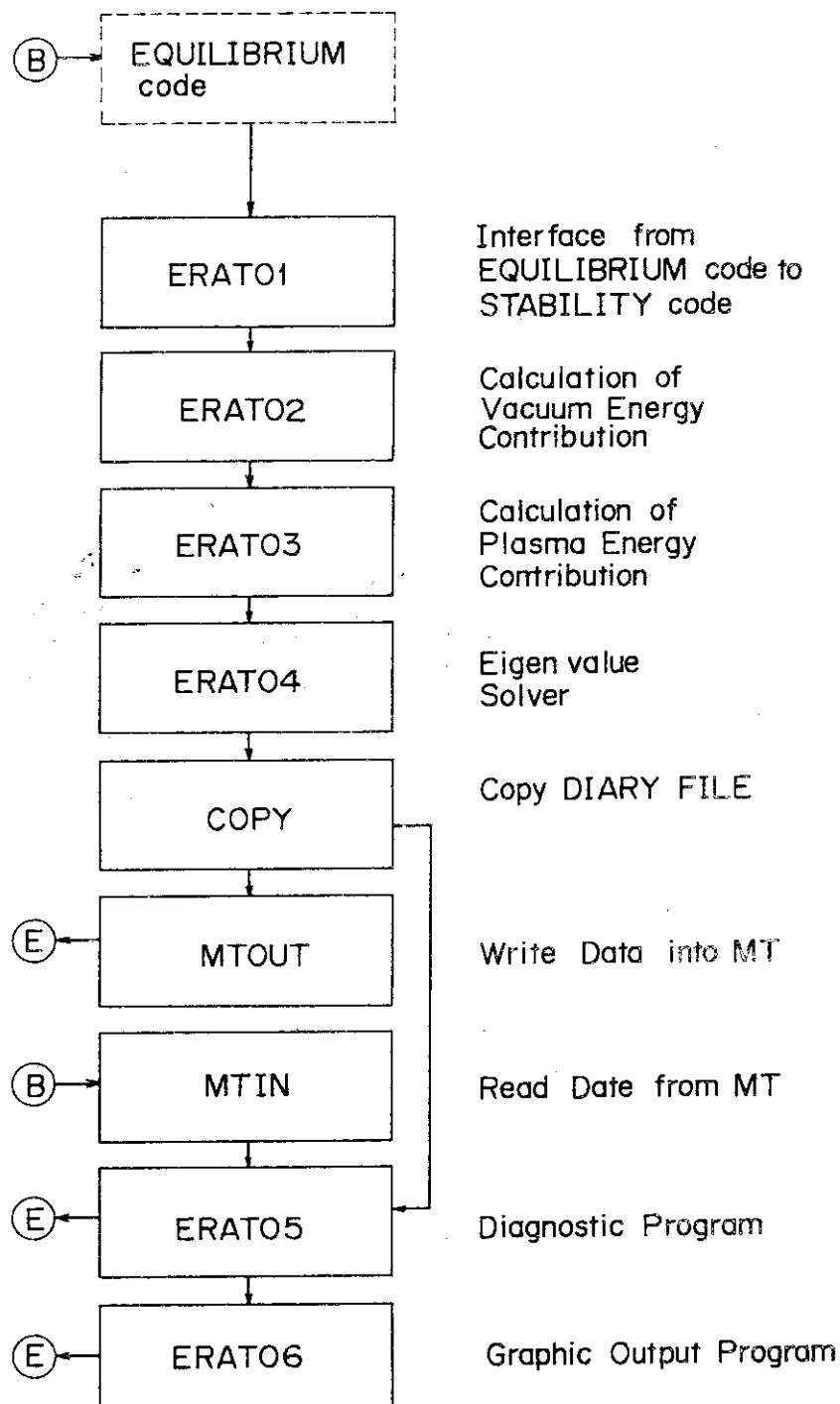


Fig.2-1 Flow chart of the ERATO job

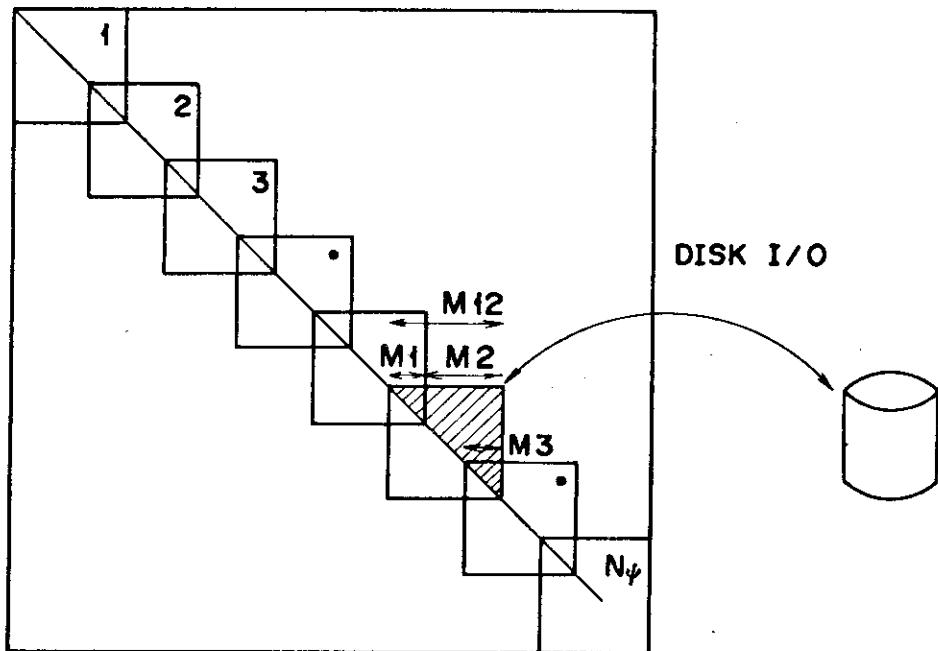


Fig.2-2 Pattern and size of the matrices A and B

Both the A and B matrices are real symmetric matrices.

Therefore, only the upper right half elements are stored.

N_{ψ} : number of blocks, $M_{12}(=NCOLMN)=8\times(N_{\chi}+1)$: length of a block,

$M_1, M_3=2\times(N_{\chi}+1)$ or 0 : number of overlapped columns.

Number of vector components is $N=(3N_{\psi}+1)\times(N_{\chi}+1)\times 2$ and the size of a block is $NCOLMN\times(NCOLMN+1)/2$, (unit: DW=8 Byte).

Example $N_{\chi}=60, N_{\psi}=60$: $NCOLMN=488, M_1=M_3=122$ or 0, $N=22082$,

size of block= 954528 Byte, required memory size for a matrix =54.62 MB.

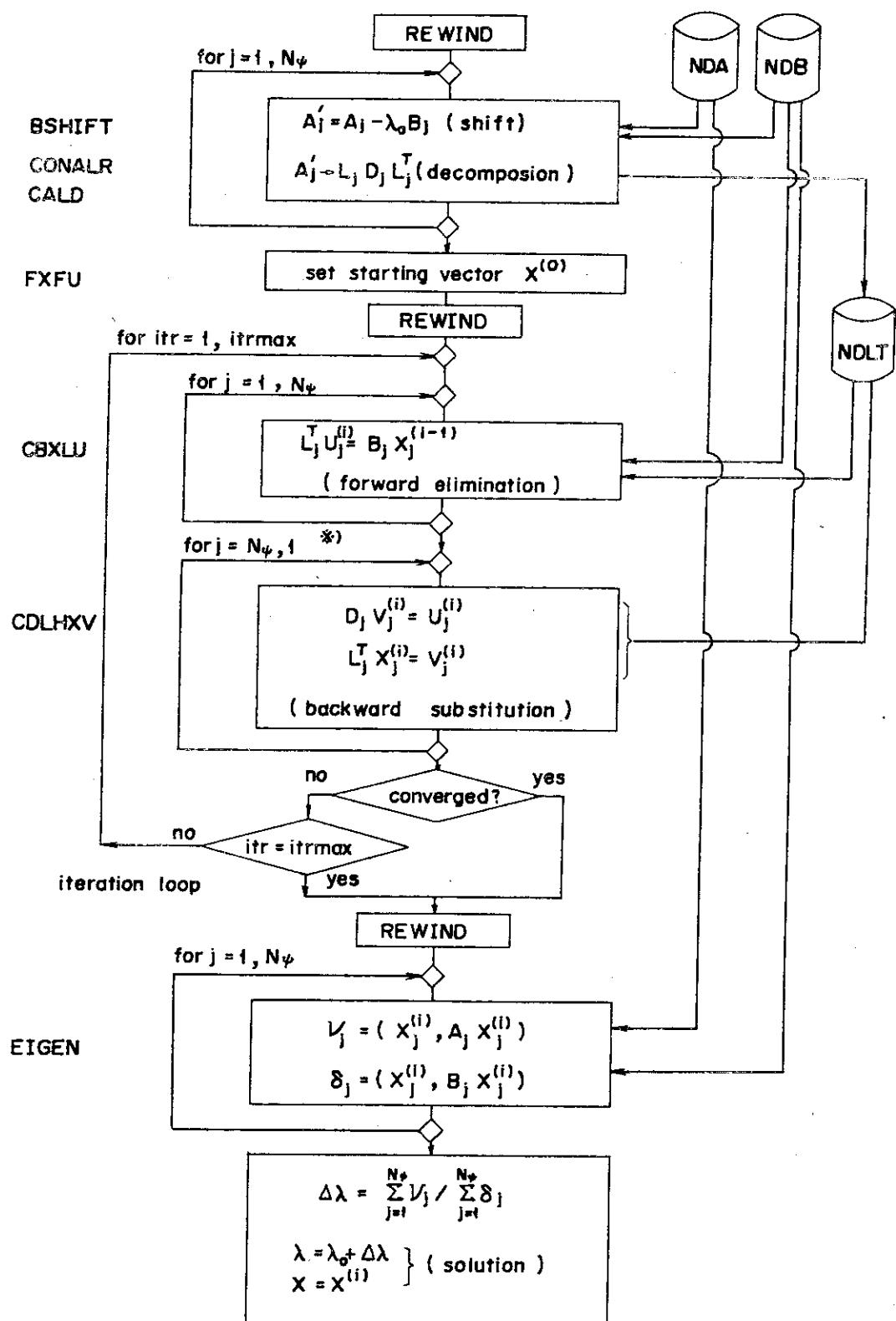


Fig. 2-3 Flow chart of the eigenvalue solver (ERATO4).

め、直接編成ファイルを用いている。（これは、標準フォートランー I/O ライブライの順編成ファイルを用いると、BACKSPACE 処理に伴うオーバーヘッドのために I/O 効率が悪化するのを避けるためである⁶⁾。）

2.3 ERATO4 のプログラム・サイズの概算

プログラムは 2 倍精度で書かれており、必要な主記憶容量は、

手続部分： 約 75 kW (= 307200 Byte)

共通領域

- 1. 行列 (A, B) : $4(N\chi + 1)[8(N\chi + 1) + 1]$ DW, (2-1)
- 2. ベクトル (XT, UT) : $2(3N\psi + 1)(N\chi + 1)$ DW, (2-2)
- 3. ベクトル (X, U, W, WK) : $8(N\chi + 1)$ DW, (2-3)

(単位 1 DW (Double Word) = 8 Byte)

の和として与えられる。具体的な、 $N\psi$, $N\chi$ の値に対するプログラムサイズを、第 2-2 表及び第 2-4 図に示す。

2.4 ディスク・ファイル・サイズの概算

ERATO では、ジョブステップ間のデータ受渡しは、すべてディスク・ファイルを経由して行う。また、ERATO 4 のジョブステップ内でのみ使用される作業用ファイルもある。ジョブステップと使用ファイル機番との関連表を第 2-3 表に示す。ERATO-J (F 版) の標準的使用法では、合計 15 本のファイルを必要とするが、そのうち、特に大容量のものは、行列データ、A, B, DL^T 用の 3 本である。行列 A, B は、ERATO 3 で作成されて、ERATO 4 へ渡されるものである。（ファイル機番 NDA, NDB）。行列 DL^T は、行列、 $A - \lambda_0 B$ (λ_0 は固有値のシフト量) をコレスキー分解した結果得られるもの ($A - \lambda_0 B = LDL^T$) で、ERATO 4 のジョブステップ内でのみ使用される。ベクトル逆反復法のループ中で、後退代入に伴うファイル操作を効率良く行うために、この行列だけは、直接編成ファイルに格納している。（ファイル機番 NDLT）。以上 3 つの行列は、すべて、第 2-2 図で示した同一の行列パターンで取り扱っている。ディスク・ファイル容量は以下の式で評価できる。

- 1. 行列 (A, B, DL^T) : 各 $4N\psi(N\chi + 1)[8(N\chi + 1) + 1]$ DW, (2-4)
- 2. 固有ベクトル (XT) : $2(3N\psi + 1)(N\chi + 1)$ DW, (2-5)
- 3. その他 : 少々

具体的な数値を、第 2-2 表及び第 2-5 図に示す。

これらの大容量ディスク・ファイルのスペース確保にあたっては、計算センターの運用条件が許す限り、ERATO ジョブが最終的に必要とするファイル・スペースを、一次量で確保し、（ファイルの利用率が高く、連続した未使用スペースが少ない場合には、大容量ファイルを一次量で確保することは困難であるため、一次量を小さく指定し、二次量の増分を繰り返し行って最終的必要量を確保することになるが、その場合には、論理的には同一のファイルが、物理的には細切

Table 2-2. Program size and required disk memory for ERATO calculations

 $p = 75KW = 34,400DW : 1 TRK = 19,064 Byte, 1 CYL = 30 TRK = 571,920 Byte$

	a) NCOJMN	b) N	c) Size of a BLOCK DW	Size of ERATO4 $4a+2b+2c+p \leq d$ DW	same (in MB)	Size of a MATRIX $c \times N\Psi \times 8 \leq e$ (in MB)	same M (in TRK)	Size of Scratch Files for 3 MATRICES (in CYL)
$N_\chi = N_\Psi =$	20	168	2562	14196	72588	0.55	2.17	120
	= 30	248	5642	30876	112428	0.86	7.07	389
	= 40	328	9922	53956	163356	1.28	16.47	906
	= 50	408	15402	83436	237708	1.81	31.83	1751
	= 60	488	22082	119316	323148	2.47	54.62	3005
	= 70	568	29962	161596	423788	3.23	86.30	4747
	= 80	648	39042	210276	539628	4.12	128.34	7060
	= 90	728	49322	265356	670668	5.12	182.21	10023
	= 100	808	60802	326836	816908	6.23	249.36	13716
	= 110	888	73482	394716	978348	7.46	331.26	18221
	= 120	968	87362	468996	1154988	8.81	429.38	23618
	= 130	1048	102442	549676	1346828	10.28	545.18	29987
	= 140	1128	118722	636756	1553868	11.86	680.13	37410
	= 150	1208	135900	730236	1775504	13.55	835.69	45966

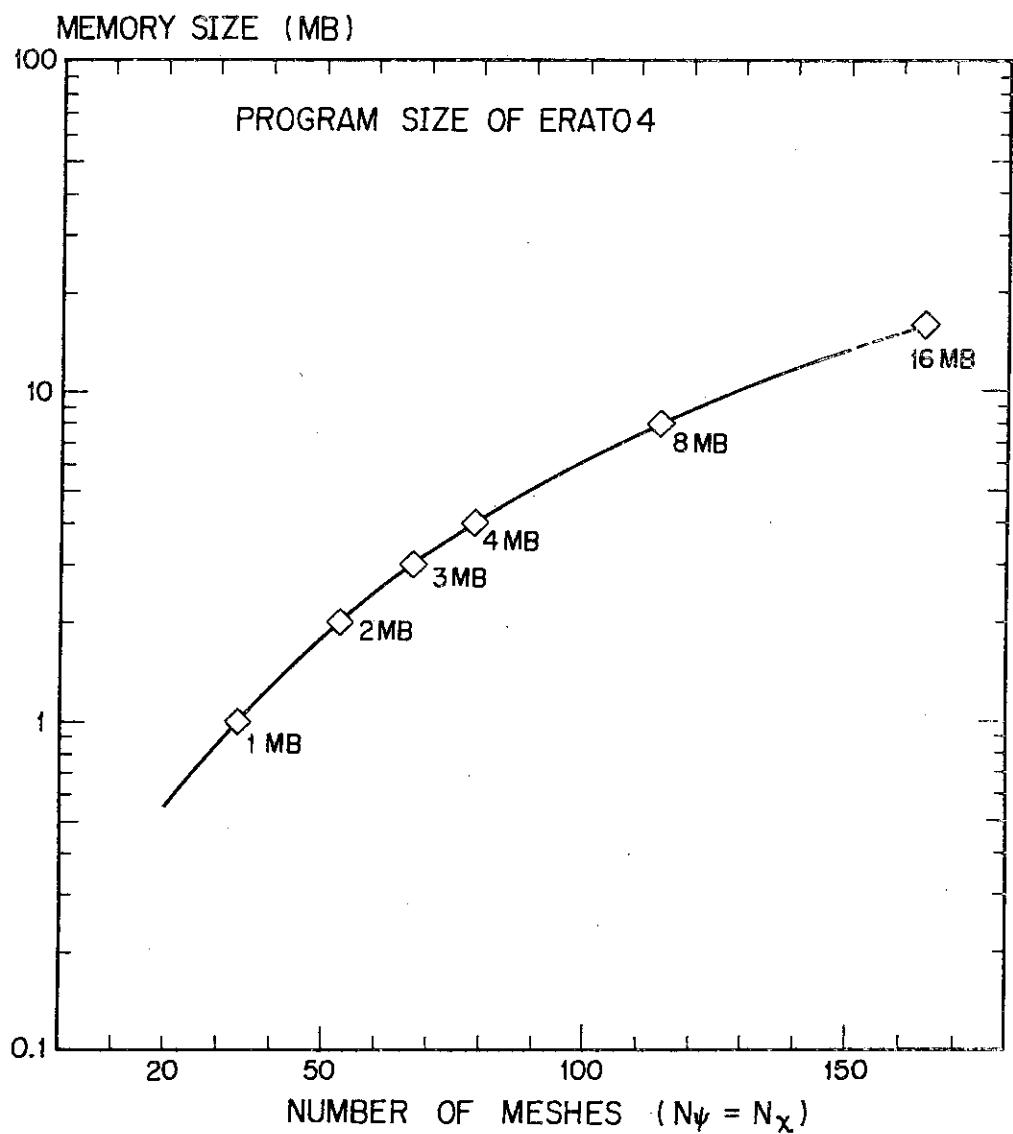


Fig.2-4 Program size of the ERATO4 as a function of the mesh number.

Table 2-3. Cross-reference table of job steps to I/O reference numbers

job steps I/O reference numbers	Content		
	ERATO6 (graphic)	ERATO5 (printing)	MTIN
FT01			Eigenvalue with input data (for convergence check)
FT04 (MEQ)	○	○	Mapped equilibrium
FT05 (NIN)	○	○	Read NAMELIST from card
FT06 (NPRNT)	○	○	Line printer output
FT07 (NDA)	○	○	Matrix A
FT08 (NSAVE)	○	○	NAMELIST
FT09 (NDB)	○	○	Matrix B
FT10 (NDS)	○	○	Eigenfunction
FT11 (NDORY)	○	○	Numerical equilibrium
FT16 (NDES)	○	○	Plotting quantities
FT17 (NVAC)	○	○	Surface equilibrium and vacuum matrix
FT22	○	○	Mode structure, q-profile etc.
FT33 (NDLT)	○	○	Matrix LDLT (direct access file)
FT50	○	○	Eigenvalue with input data
FT60	○	○	Preservation & retrieval of the final state of ERATO4 (via Magnetic Tape)
*	Equilibrium codes	**) Catalogue file	***) MT

*) Equilibrium codes **) Catalogue file ***) MT

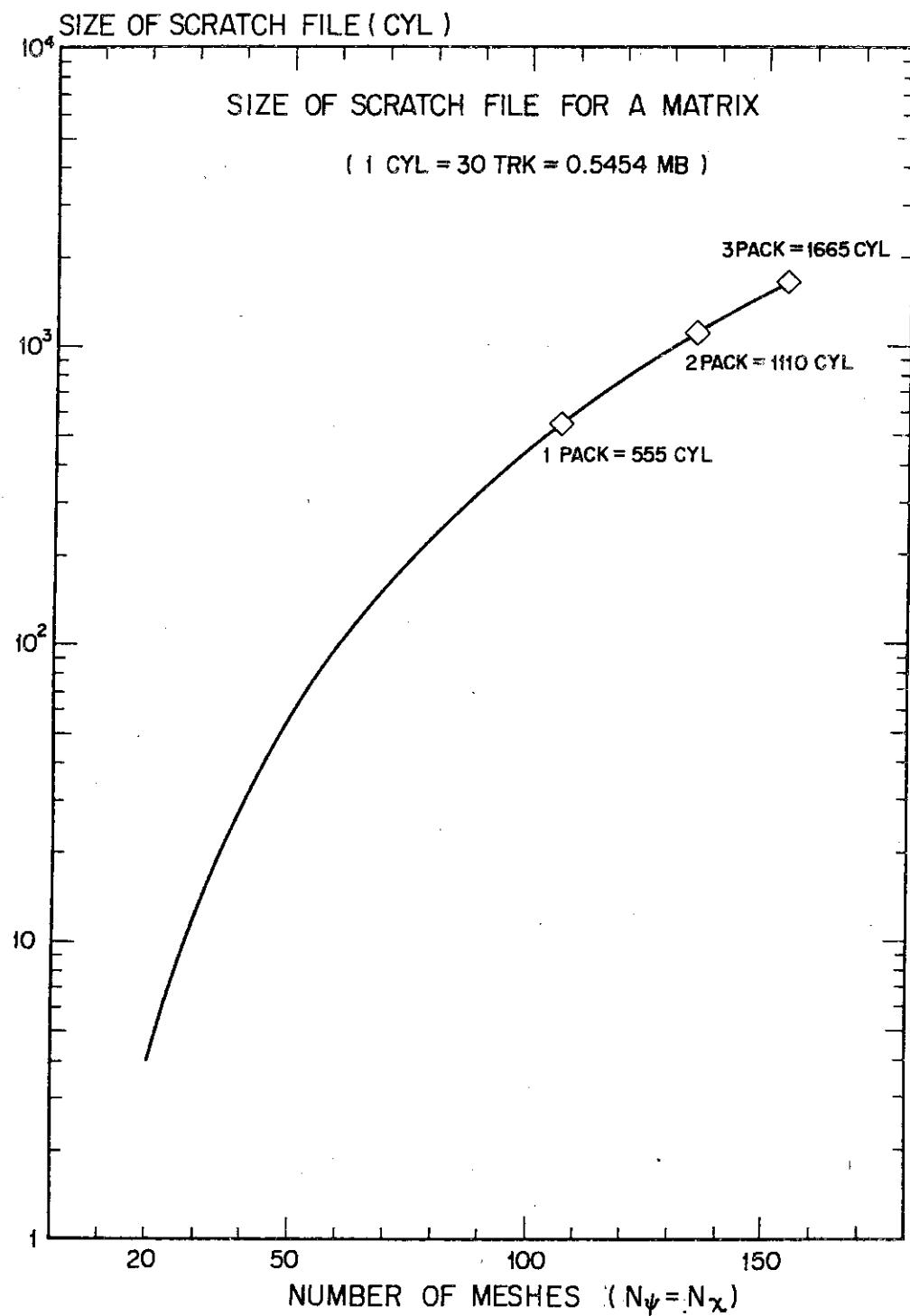


Fig.2-5 Necessary amount of the disk memory.

れの領域の集合になってしまう。) かつ、スペース要求をシリンド単位 (CYL 指定) で行なうことが、I/O 効率向上のために望ましい。理由は、回転式ディスク・ファイル装置では、ディスク・ヘッドの位置決め (シリンドの切替え) のための時間が無視できないので、このオーバーヘッドを減らすためには、連続領域から成るスペースを確保する方が良いからである。

なお、直接編成ファイルを使用する場合には、実際の入出力処理を行う前に、一度だけ、必要なデータセット領域の全体を初期化 (確保した領域に零を書く) しなければならず、この領域をあとで増分指定で拡張することは出来ない。この初期化は、DEFINE FILE 文の指示の内容に従って、最初に現われる直接 WRITE 文の実行の際に行われる。従って、取り扱うべきデータ総量の最大値に応じた指示を予め与えておく必要がある。ERATO 4 の場合、サブルーチン IODISK 4 中の DEFINE FILE 文、

DEFINEFILE NDLT (M, 4766, U, IX) (2-6)

において、メッシュ数 N_ψ , N_χ に応じて、M の値を以下のように設定すればよい。

$$M = \left[\frac{8N_\psi(N_\chi + 1)\{8(N_\chi + 1) + 1\}}{4766} \right] + 1 \quad (2-7)$$

Gauss

M はブロック化されたフォートラム記録の総個数を意味し、[] Gauss はガウスの記号である。また、(2-6) の表式中、4766 及び U は、1 フォートラム記録の最大記録帳 (ブロック長) が、1 語 = 4 バイトを単位として、4766 語 (= 19064 バイト = 1 トラック) であることを示している。従って、M は総データ量をトラック単位で表わしたものと考えてよい。第 2-2 表に M の値を示す。実際に M の値を変更するには、プログラムソースの修正が必要になるが、この作業は、EOS プリプロセッサより行えるので、コモン・ファイル中の M の値を修正すればよい (詳細は、3-2 節参照)

2.5 ERATO 4 の CPU 時間の予測

ERATO 4 の CPU 時間は以下の式で評価できる。

$$(CPU \text{ 時間}) = \alpha N_\psi (N_\chi + 1)^3 + \beta N_\psi (N_\chi + 1)^2 \times \text{itr} (\text{sec}), \quad (2-8)$$

ここで第 1 項は行列 $A - \lambda_0 B$ のコレスキー分解に対応し、第 2 項はベクトル逆反復法に対応する CPU 時間である。itr は収束計算が終了するまでの反復回数である。FACOM M 200 では、 $\alpha \cong 7.29 \times 10^{-5} \text{ sec}$, $\beta \cong 2.29 \times 10^{-6} \text{ sec}$ である。反復回数 $\text{itr} = 3, 5, 10, 15, 20$, 及び 30 に対する CPU 時間の計算値を第 2-4 表及び第 2-6 図に示す。

2.6 I/O アクセス回数の概算

主記憶装置と入出力装置との間のデータ転送の頻度は I/O アクセス回数として、ユーザに通知され、ジョブの I/O 負荷の程度を示す。原研計算センターでは、ジョブ優先順位及びジョブ

Table 2-4. Summary of required system resources

		$\gamma = 2\gamma_0$					
		a)	b)	c)	d)	e)	g)
		CPU TIME for Cholesky decomp.	CPU TIME for One Inv. Itr.	EXCP Count for UNIT I/O	I/O TIME for UNIT I/O	I/O TIME (1.33+itr)xd (sec)	Elapsed TIME f + 2g (sec)
		$\alpha N_\Psi (N_\chi + 1)^3$	$\beta N_\Psi (N_\chi + 1)^2$	$(2.6 + \text{itr}) \times C$	$\gamma \cdot N_\Psi (N_\chi + 1)^2$	$(1.33 + \text{itr}) \times d$ (sec)	
$N_\Psi = N_\chi = 20$	itr= 3			0.203x10 ⁴	0.47x10 ²	1.1x10 ²	
	5	13.50 sec	1.23 sec	17.2 19.7 25.8 32.0	0.275 0.455 0.636	0.68 1.22 1.76	1.6
	10			361 times	10.79 sec	2.7	
	15			38.1	0.816	3.9	
	20			50.4	1.18	5.0	
$N_\Psi = N_\chi = 30$	3			77.3 85.3 105.5 126.6 145.8	0.661x10 ⁴ 0.897 1.49 2.08 2.67	0.152x10 ³ 0.22 0.40 0.58 0.75	0.38x10 ³ 0.53
	5			186.1	3.85	1.11	
	10	65.16	4.03				
	15						
	20						
$N_\Psi = N_\chi = 40$	3			229. 248. 295.	1.54x10 ⁴ 2.09 3.47	0.35x10 ³ 0.52 0.93	0.94x10 ³ 1.29
	5			342. 389. 483.	2752 4.84 6.22 8.97	82.25 1.34 1.75 2.58	2.16 3.03 3.90 5.64
	10	201.0	9.39				
	15						
	20						
$N_\Psi = N_\chi = 50$	3			538. 574. 665. 756.	0.298x10 ⁵ 0.405 0.671 0.937	0.69x10 ³ 1.00 1.80 2.59	0.19x10 ⁴ 0.26 0.43 0.59
	5			847. 1029.	1.21 1.74	3.39 4.98	0.76 1.10
	10	483.6	18.2				
	15						
	20						
	30						

	a) CPU TIME for Cholesky decompo.	b) CPU TIME for One Inv. Itr. $\beta N_\Psi (N_\chi + 1)^2$	c) EXCP Count for UNIT I/O $(2.6 + \text{itr}) \times C$	d) I/O TIME for UNIT I/O $\gamma \cdot N_\Psi (N_\chi + 1)^2$	e) Elapsed TIME $(1.33 + \text{itr}) \times d$ (sec)
$N_\Psi = N_\chi = 60$	itr= 3				
	5	993.0	1.09x10 ³	0.51x10 ⁵	1.18x10 ³
	10		1.15	0.70	1.72
	15		1.31	1.15	3.09
	20		1.46	1.61	4.45
	30		1.62	2.07	5.82
			1.93	2.98	8.55
					1.90
					0.34x10 ⁴
					0.46
$= 70$	3		1.98x10 ³	0.81x10 ⁵	0.19x10 ⁴
	5		2.07	1.10	0.27
	10	1827.	2.32	1.82	0.49
	15		2.57	2.54	0.71
	20		2.81	3.27	0.92
	30		3.31	4.71	1.35
					1.67
					2.12
					3.04
					0.57x10 ⁴
$= 80$	3		3.32x10 ³	1.21x10 ⁵	0.28x10 ⁴
	5		3.47	1.63	0.41
	10	3100.	3.83	2.1481	0.73
	15		4.20		1.05
	20		4.57		1.37
	30		5.30	7.00	2.01
					1.16
					1.84
					2.51
					3.19
$= 90$	3		5.26x10 ³	1.71x10 ⁵	0.39x10 ⁴
	5		5.47	2.32	0.58
	10	4945.	5.99	3.85	1.03
	15		6.51	5.37	1.49
	20		7.03	6.90	1.94
	30		8.07	9.95	2.86
					4.55
					6.52
					0.31x10 ⁴
					1.70

		a)	b)	c)	d)	e)	f)	Elapsed TIME
		CPU TIME for Cholesky decom.	CPU TIME for One Inv. Itr.	EXCP Count for UNIT I/O	I/O TIME for UNIT I/O	I/O TIME (1.33+itr)×d (sec)	I/O TIME (1.33+itr)×d (sec)	f + 2g (sec)
	$\alpha N_{\psi} (N_{\chi}+1)^3$	$\beta N_{\psi} (N_{\chi}+1)^2$	$\epsilon \cdot N_{\psi} (N_{\chi}+1)^2$	$(2.6+itr) \times C$	$\gamma \cdot N_{\psi} (N_{\chi}+1)^2$			
$N_{\psi}=N_{\chi}$ =100	itr= 3							
	5	0.794×10 ⁴	0.794×10 ⁶	0.24×10 ⁶	0.54×10 ⁴	0.54×10 ⁴	1.87×10 ⁴	
	10	7512 sec	0.823	0.32	0.79	0.79	2.40	
	15		0.894	0.53	1.41	1.41	3.71	
	20		0.965	0.74	2.04	2.04	5.03	
	30		1.04	0.95	2.66	2.66	6.35	
$=110$	3	1.15×10 ⁴	0.31×10 ⁶	0.31×10 ⁶	0.71×10 ⁴	0.71×10 ⁴	0.26×10 ⁵	
	5	1.19	0.42	0.42	1.04	1.04	0.33	
	10	10970	1.29	0.70	1.87	1.87	0.50	
	15		1.38	0.98	1.658	1.658	2.70	
	20		1.48	1.26	3.53	3.53	0.68	
	30		1.66	1.81	5.19	5.19	0.85	
$=120$	3	1.62×10 ⁴	0.40×10 ⁶	0.40×10 ⁶	0.92×10 ⁴	0.92×10 ⁴	0.35×10 ⁵	
	5	1.67	0.55	0.55	1.35	1.35	0.44	
	10	15500	1.80	0.91	2.43	2.43	0.67	
	15		1.92	1.27	3.50	3.50	0.89	
	20		2.04	1.63	4.58	4.58	1.12	
	30		2.29	2.35	6.73	6.73	1.57	
$=130$	3	2.22×10 ⁴	0.51×10 ⁶	0.51×10 ⁶	1.17×10 ⁴	1.17×10 ⁴	0.46×10 ⁵	
	5	2.29	0.70	0.70	1.72	1.72	0.57	
	10	21310	2.44	1.15	3.08	3.08	0.86	
	15		2.60	1.61	4.45	4.45	1.15	
	20		2.75	2.06	5.81	5.81	1.44	
	30		3.07	2.98	8.54	8.54	2.02	

a) CPU TIME for Cholesky decompos.		b) CPU TIME for One Inv. Itr. $\beta N_\Psi (N_\chi + 1)^2$	c) EXCP Count for UNIT I/O $\epsilon \cdot N_\Psi (N_\chi + 1)^2$	d) I/O TIME for UNIT I/O $(2.6 + \text{itr}) \times C$ $\gamma \cdot N_\Psi (N_\chi + 1)^2$	e) I/O TIME $(1.33 + \text{itr}) \times d$ (sec)	f) Elapsed TIME $f + 2g$ (sec)
$N_\Psi = N_\chi$	$= 3$	2.98×10^4	0.64×10^6	0.146×10^5	0.215	0.59×10^5
	$= 5$	3.06	0.87		0.385	0.74
	$= 10$	3.25	1.44		0.56	1.10
	$= 15$	3.44	2.01		0.73	1.46
	$= 20$	3.64	2.58			1.82
	$= 30$	4.03	3.72			2.54
$N_\Psi = N_\chi$	$= 140$					
	$= 3$	3.91×10^4	0.79×10^6	0.180×10^5	0.264	0.93
	$= 5$	4.00	1.07			
	$= 10$	4.24	1.77			
	$= 15$	4.48	139971	4184	0.47	1.37
	$= 20$	4.72	2.47		0.68	1.81
	$= 30$	5.20	3.17		0.89	2.26
					1.31	3.14
$N_\Psi = N_\chi$	$= 150$					
	$= 3$					
	$= 5$					
	$= 10$					
	$= 15$					
	$= 20$					
	$= 30$					

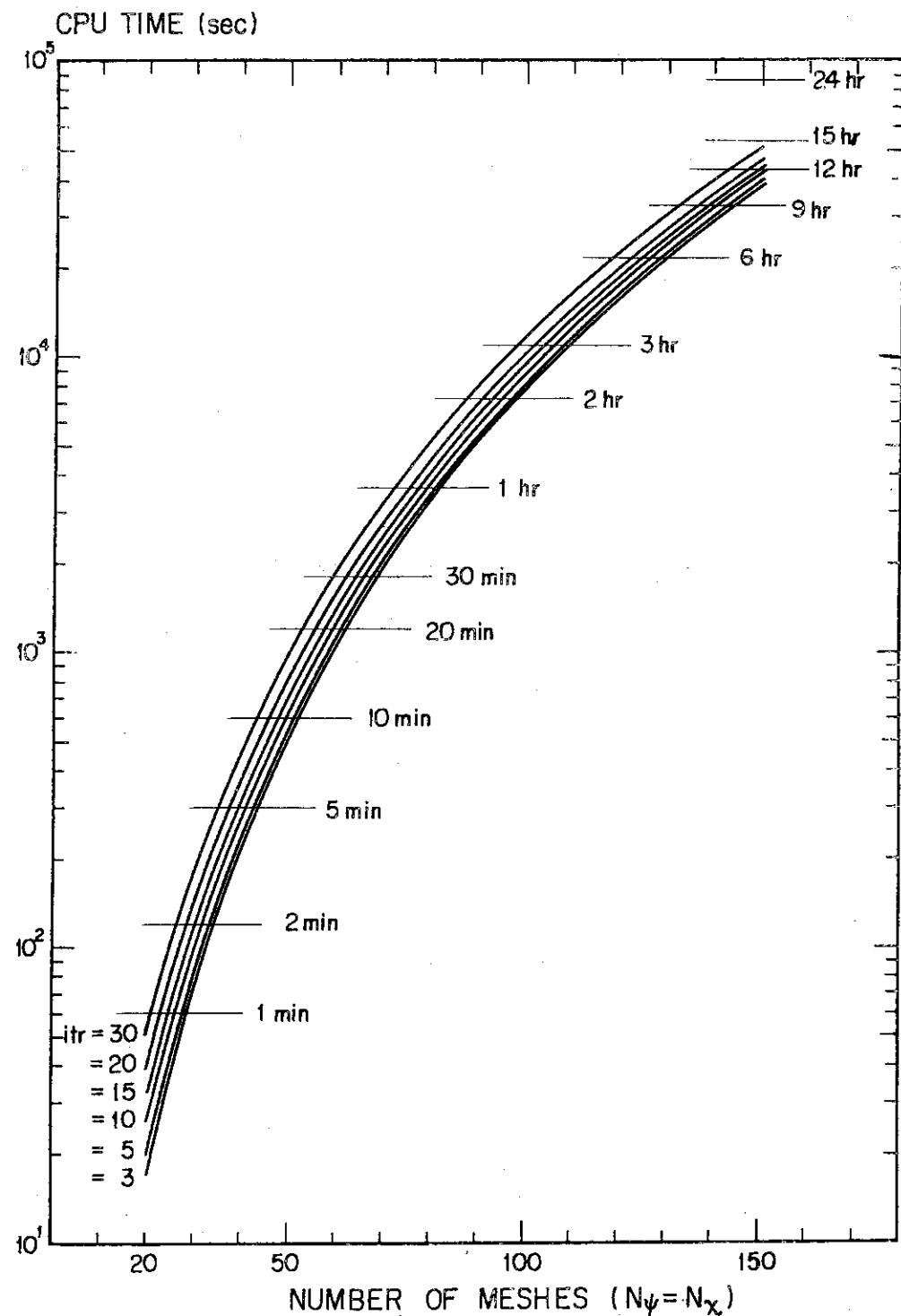


Fig. 2-6 CPU time for ERATO calculation.

打ち切りのパラメータの一つとして採用している。I/O アクセス回数は、EXCP 発行回数（ブロック化されたデータを単位とする転送回数）に換算してカウントしているため、大量データの転送には、使用装置に許される最大のブロック化因子を用いるべきである。ERATO の I/O アクセス回数の主要部は、次式で評価できる。

$$I/O \text{ アクセス回数} \cong \epsilon N_\psi (N_\chi + 1) \times (itr + 2.6) \quad (2-9)$$

この表式は、ERATO 3における行列 A, B のディスクファイルへの書き出し、及び ERATO 4における直接アクセスファイルの初期化に伴う I/O アクセス回数を含んでいる。行列データの転送に関しては、最大のブロック化因子（19064 Byte = 1 TRK）を使用するものとすれば $\epsilon \cong 4.09 \times 10^{-2}$ である。（2-9）式による I/O アクセス回数の概算値を第 2-4 表及び第 2-7 図に示す。

2.7 I/O 時間の概算

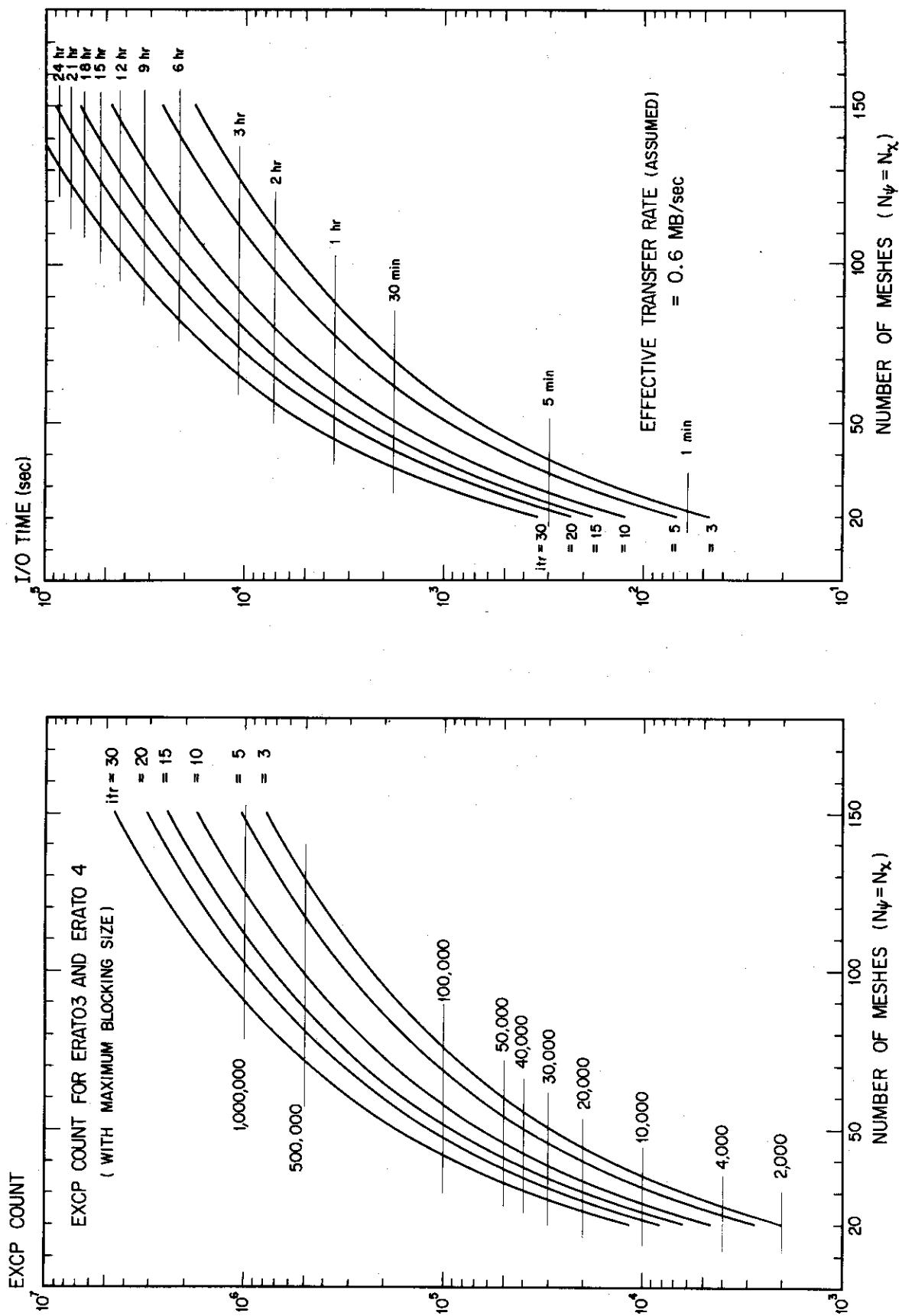
ERATO コード実行上の大量データの I/O 処理に伴い、データ転送中、このジョブが入出力チャンネル装置の一部を占有することになる。I/O 時間は、このチャンネル占有時間を示すものであり、原研の M 200 システムでは、直接ユーザに通知されることは無いが、ジョブの I/O 負荷の特徴を示す一つの指標である。ERATO 4 の I/O 時間は次式で与えられる。

$$I/O \text{ 時間} \cong r N_\psi (N_\chi + 1)^2 \times (itr + 1.33), \quad (2-10)$$

ここで、 r はディスク装置の転送速度で決まる因子である。ディスク装置の転送速度の最大値（ハードウェア性能）は、1.2 MB/sec であり、もし、ユーザがその性能を利用し切れば、 $r = r_0 \cong 6.114 \times 10^{-4} \text{ sec}$ に近いものになると思われるが、そのためには、ユーザが新たに EXCP マクロ命令を用いた I/O プログラムを作成する手間が要るため、この種の最適化は M 200 については未だ行っていない。（旧 F 230-75 システムでは、このような専用 I/O プログラムを作成して、良い結果が得られている。⁶⁾）一般的のユーザプログラムで用いるフォートラン - I/O ライブライアリを介した入出力処理は、使用方法が簡便であるが、I/O ライブライアリのオーバーヘッドが大きいため実効的な転送速度は、ハードウェア性能の約半分程度と思われる。従って実際に即した I/O 時間を概算するため、 $r = 2r_0 \cong 1.223 \times 10^{-5} \text{ sec}$ （実効転送速度 0.6 MB/sec）と仮定した場合の I/O 時間を、第 2-4 表及び第 2-8 図に示す。

2.8 経過時間の予測

ERATO ジョブは計算機システム資源の要求が巨大なため、他のジョブとシステム資源利用上の競合を生じやすく、計算センター運用条件下の多重プログラミング処理システムでの経過時間を予測することは極めて困難である。システム資源の競合（優先権が低いための待ち時間も含む）に由来する不測のオーバーヘッドが無視できれば、経過時間、CPU 時間と I/O 時間の関係は次式で与えられる。



$$(経過時間) = (CPU時間) \cong (I/O時間) \cong r N \psi (N \chi + 1)^2 (itr + 1.33), \quad (2-11)$$

$(r = 2r_0)$

実際のオーバーヘッドの効果を見るために、第2-9図に

$\{(経過時間) - (CPU時間)\} / (itr + 1.33)$ という量の実測値と、 $r N \psi (N \chi + 1)^2$ の曲線 ($r = r_0, 2r_0, 4r_0$ の場合) を示す。原研計算センター運用条件下では、多重プログラミングに由来するオーバーヘッドの効果は、経験的に $r = 4r_0$ として経過時間の増加に換算できる。実測値のバラツキ具合は競合ジョブとの干渉の結果と考えられる。上記経験則 ($r = 4r_0$) に基づいて計算した経過時間の予測を第2-4表及び第2-10図に示す。第2-9図における実測値のバラツキから明らかな様に、ここで示した経過時間の予測値は、一つの目安を与えるに過ぎない。

2.9 不等メッシュ ($N_\psi \neq N_\chi$) の場合のシステム資源必要量

前述のシステム資源評価の表およびグラフでは、メッシュ数はすべて、 $N_\psi = N_\chi$ の場合について示してある。しかし、解析の都合上、一方のメッシュに関する精度を相対的に高める必要が生じ得る。ここでは特に、 $N_\chi = \text{一定}$ 、 $N_\psi \neq N_\chi$ の場合のシステム資源評価の一例を示す。
 $N_\chi = 40, 50, 60, 70$ の場合について、 $itr = 3, 5, 10$ の場合について評価を行い、プログラムサイズ、ディスクファイルサイズ、CPU時間、I/Oアクセス回数、I/O時間の評価結果、及び経過時間の予測値を、それぞれ、第2-11、2-12、2-13、2-14、2-15 及び 2-16 図に示す。なお、評価方法は、(2-1)～(2-11)式に従った。

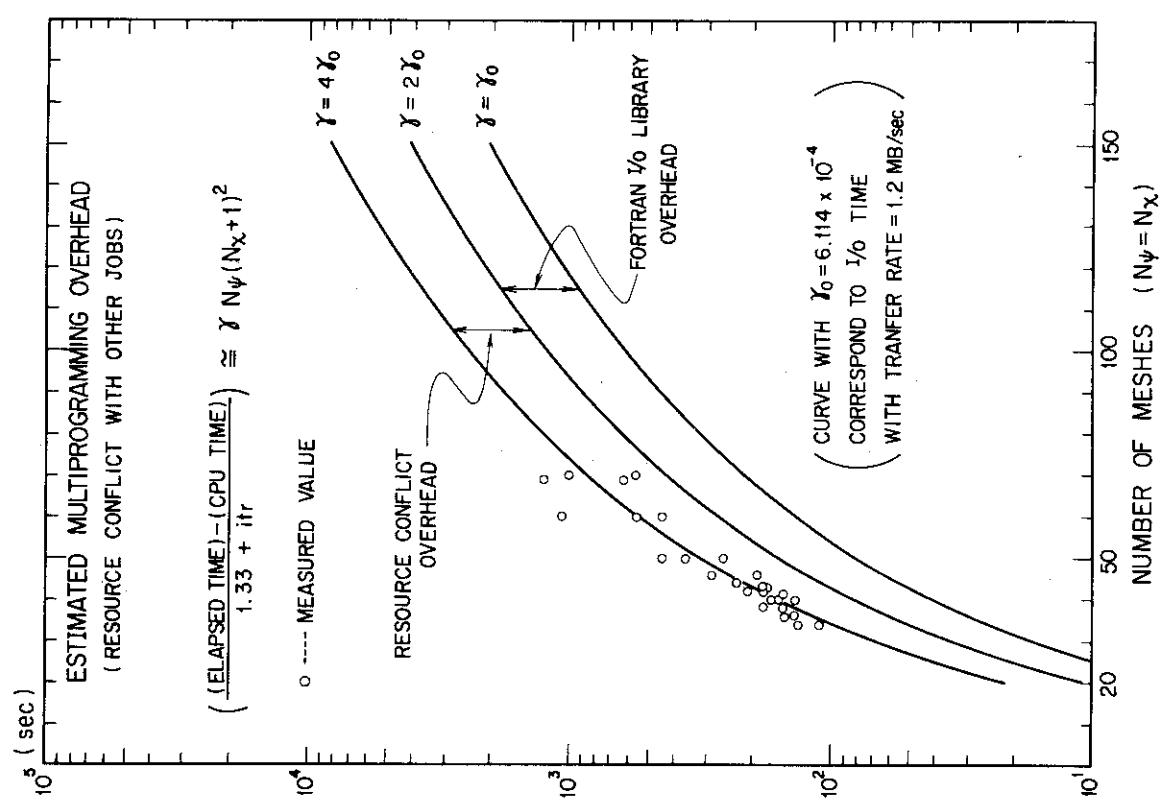


Fig. 2-9 Estimated multiprogramming overhead.

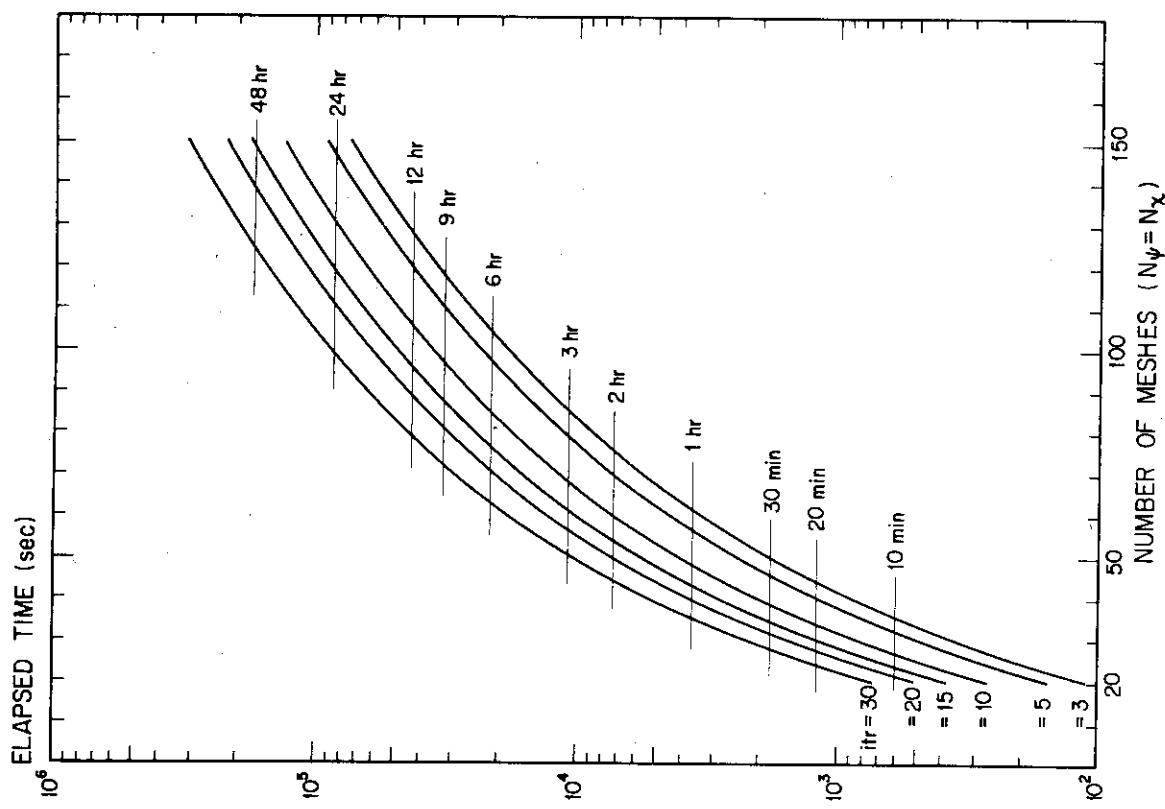


Fig. 2-10 Estimated elapsed time.

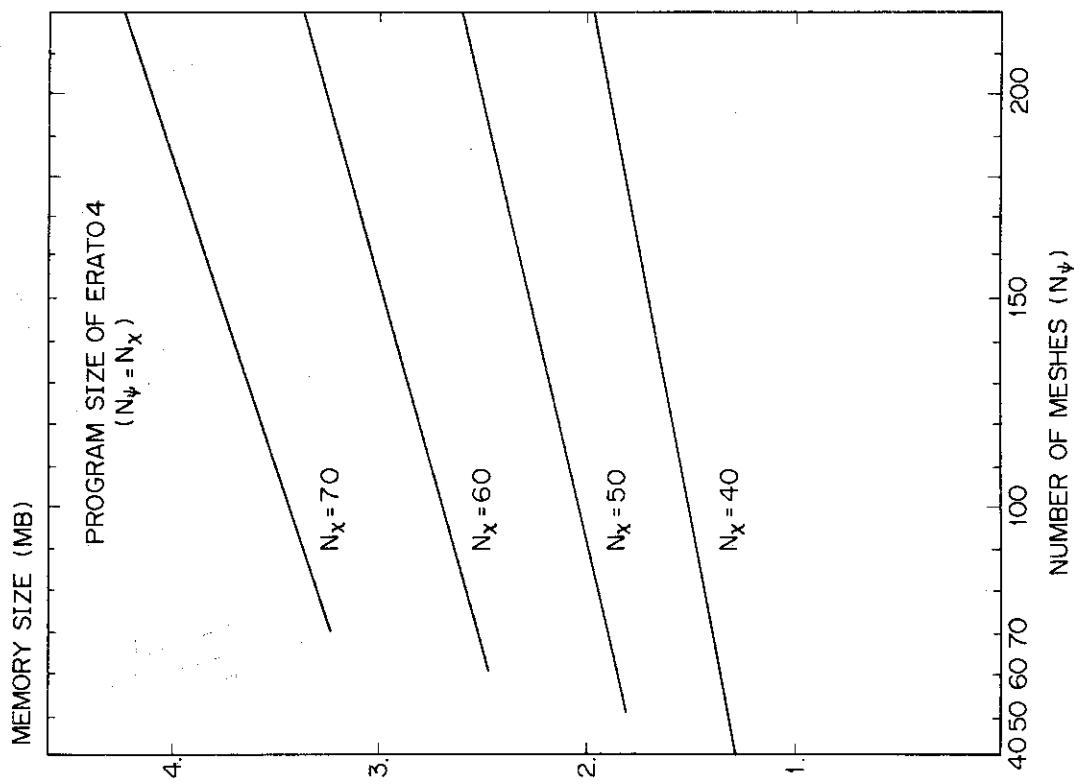
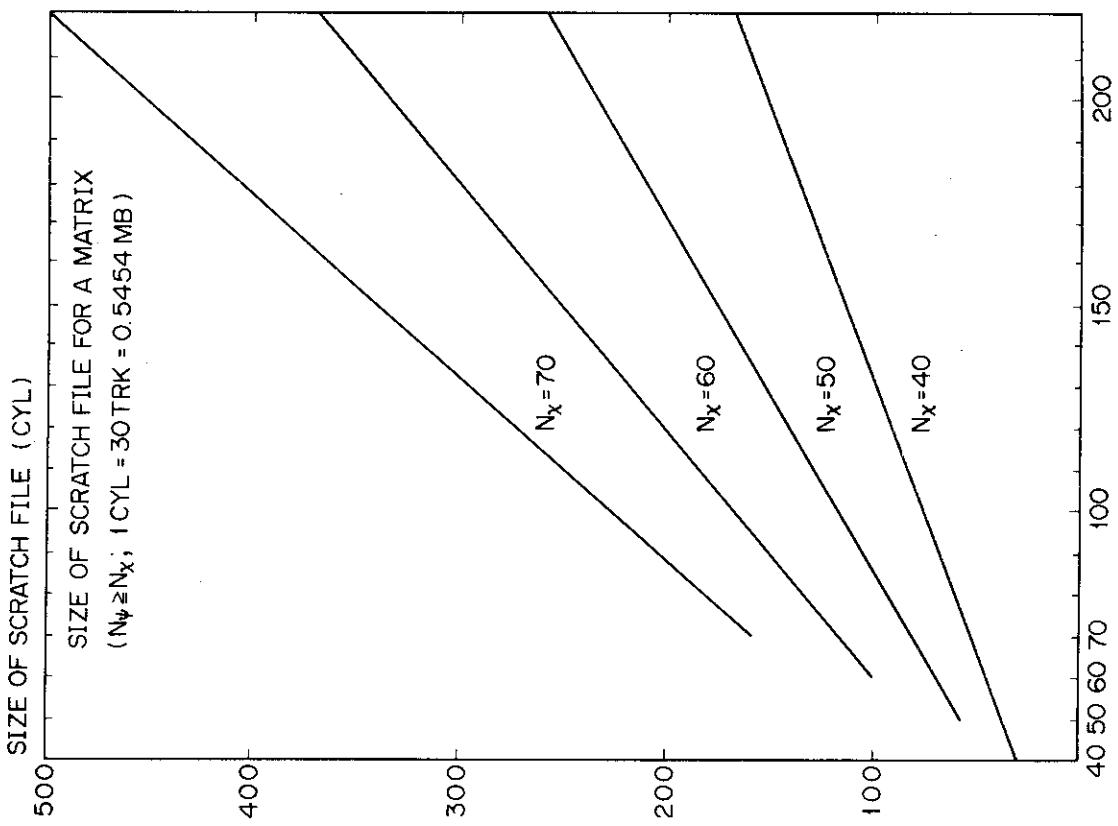


Fig. 2-11 Program size for the calculations with unequal mesh numbers,

Fig. 2-12 Necessary disk size for the calculations with unequal mesh numbers.

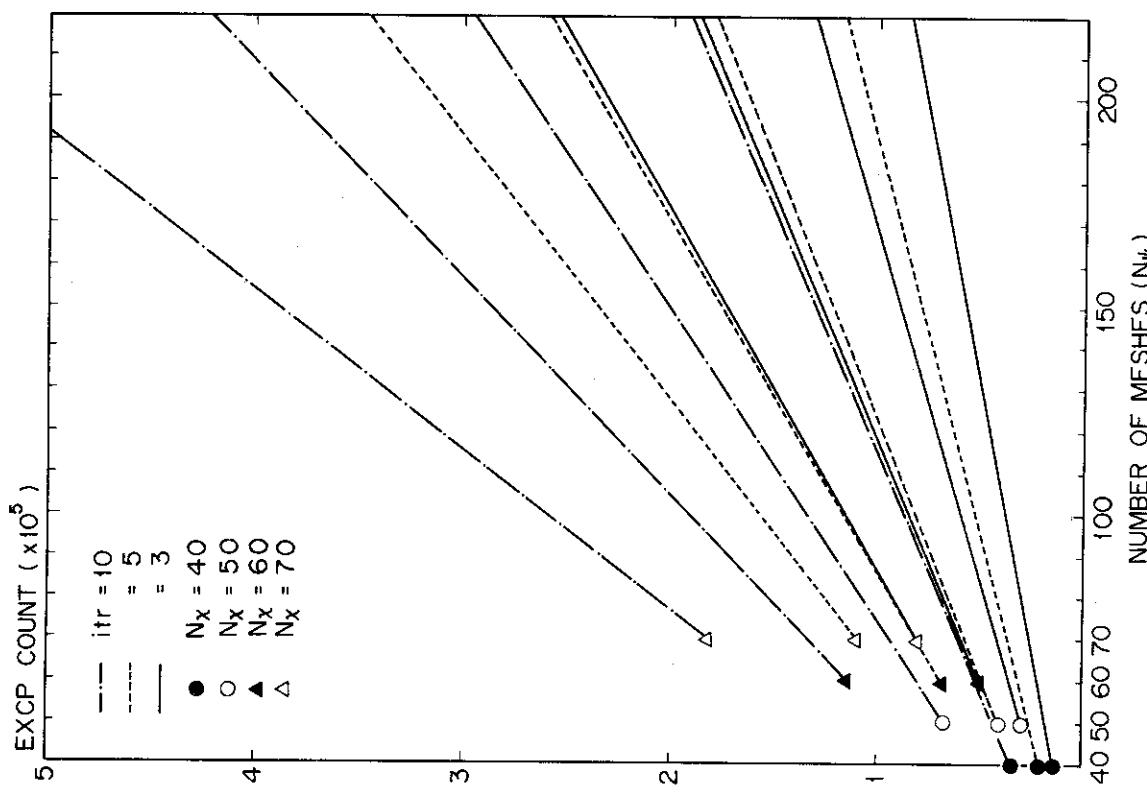


Fig. 2-14 I/O access times for the calculations with unequal mesh numbers.

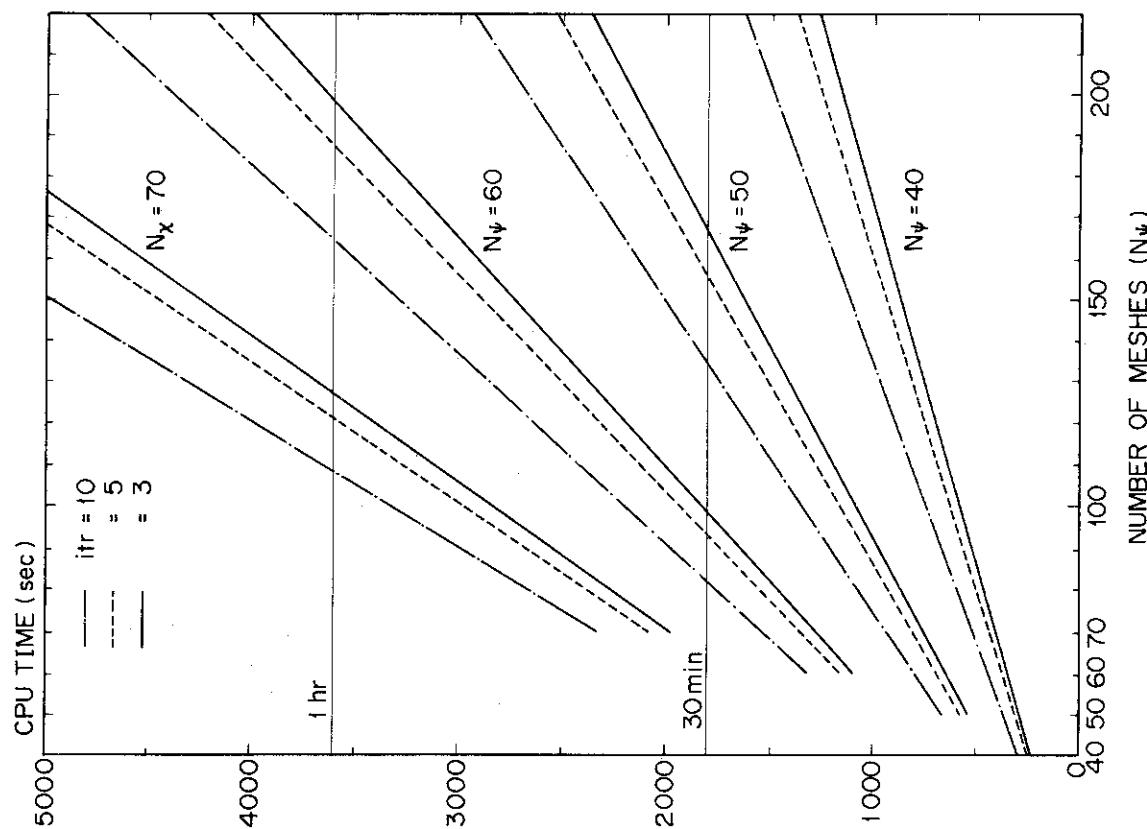


Fig. 2-13 CPU time for the calculations with unequal mesh numbers.

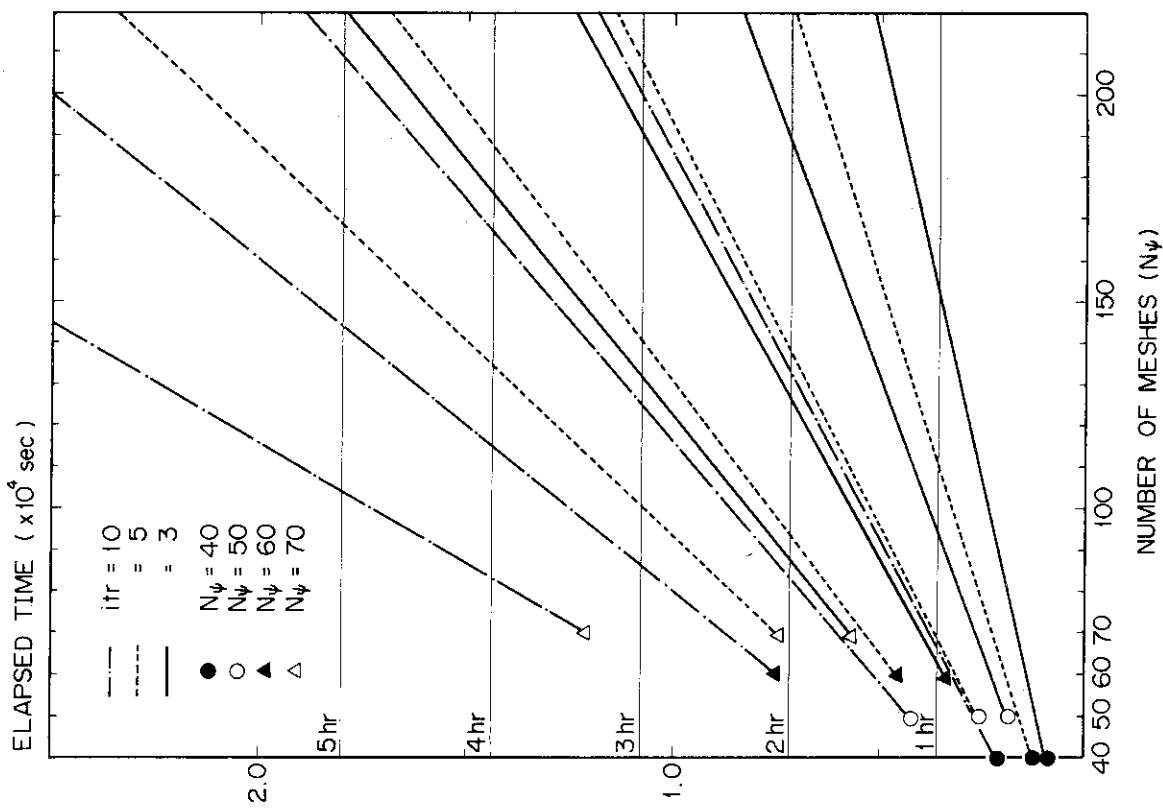


Fig. 2-16 Elapsed time for the calculations with unequal mesh numbers.

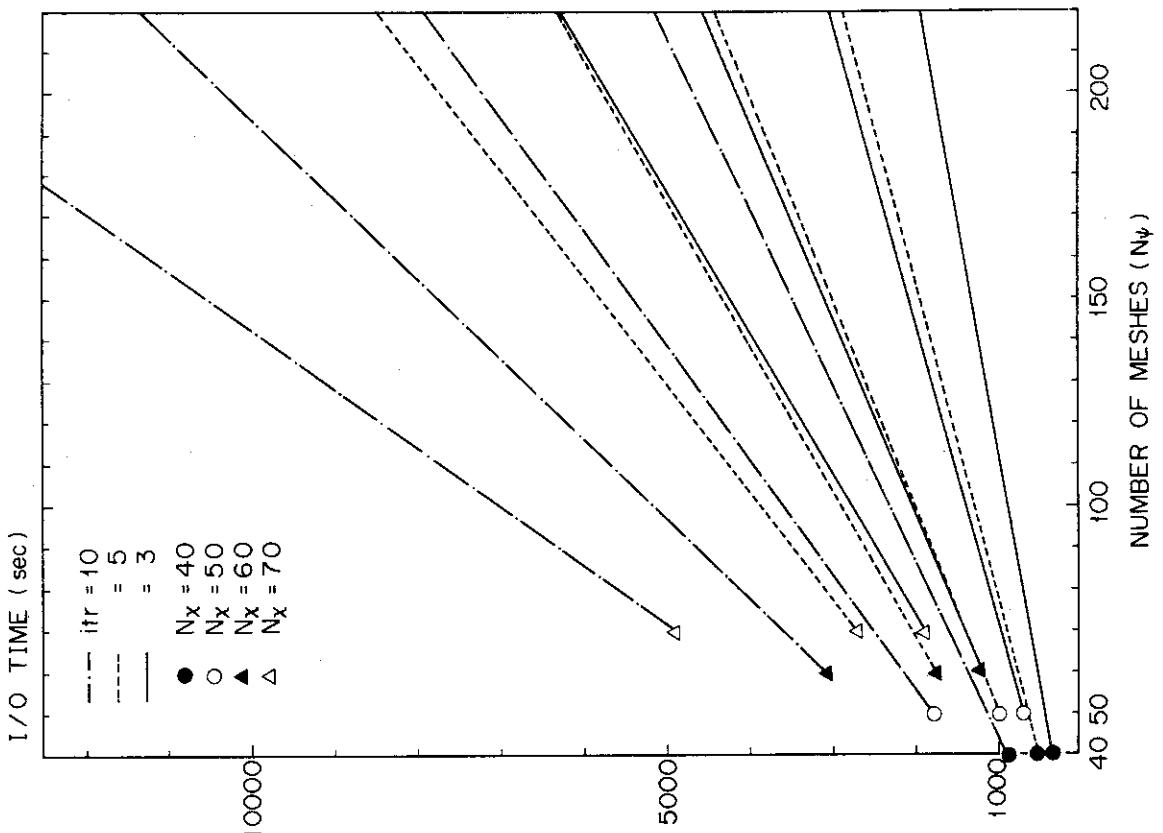


Fig. 2-15 I/O time for the calculations with unequal mesh numbers.

3. ERATO コード使用手順

3.1 概要

前章までに述べてきたことから理解されるように、ERATO コードによる計算は、大量のシステム資源を使用しつつ何回も計算を行う必要があるという特徴をもつ。この為、一連の計算を行って物理的な意味のある結果を得たい場合には次のような事に注意しなければならない。

- (1) 通常、最終的結果としては、計算メッシュ数に関する収束計算を行い、無限小メッシュ巾の答を外挿によって求めなければならない。従って、メッシュ数の異なる多数の計算が必要であるが、システム資源の節約の観点からはなるべくプログラム・サイズを必要最小限の大きさにすることが望ましい。この為には、メッシュ数に応じて、プログラムのロード・モジュールを変更して計算しなければならない。
- (2) 多数の計算を、長期間にわたって実行する必要上、ジョブの入力が常に間違いなく行われるようになっていなければならない。
- (3) 1回の計算で得られるのは、不安定性成長率に相当する1つの数値(行列の最小固有地)であるが、全体の計算ケース数が膨大であるため関連データの収集を自動的に行い、入力データ、計算条件に従ってこの数値を分類し、希望の分類に従って過去に計算で得られたデータを見やすい形式で表示できることが望ましい。

ERATO-J (F版) では、次のような方法で、これらの要求にこたえている。

(i) EOS システムの利用

EOS⁹⁾ は、配列サイズの指定を変数や数式の形で行える機能を持ったプリプロセッサである。ERATO-J (F版) は、EOS 用に書き直してあるので、計算可能なメッシュ・サイズの最大値、NP (N_y) および、NC (N_x) を入力データとして与える事により、全ての配列の大きさが計算され確保される。この為、メッシュ・サイズを変更する度に、必要な配列サイズを計算しなおしてフォートラン・ソース・プログラムに手を入れる必要はない。

(ii) ジョブ投入手順の簡略化

いくつかの場合を考慮して、JCL プログラムを固定し、端末装置を使って入力データ等の変更を行いジョブの投入が行えるように手順を簡略化してある。

(iii) DIARY, MTOUT, MTIN の利用

DIARY プログラムは、ERATO 計算の主要な結果と入力データ、計算条件等を保存し、検索、図形表示をする為のプログラムである。また、MTOUT, MTIN プログラムは、ERATO 計算の途中結果を全て磁気テープ (MT) 上に保存し、必要に応じて参照する為のプログラムである。これらのプログラムを利用することによって、大量のライン・プリンター出力用紙を保存しておく必要から解放され、これらの中から必要なデータを探し出す手間を無くすことができる。

一連のERATO計算を行って、物理問題の答を得ようとする場合、標準的には、DIARY FILE, MT の準備を行い、ロード・モジュールを作成し、計算を実行した後、DIARY FILE 参照プログラムを利用して結果を整理するという手順が必要である（第3-1図参照）。

以下の節では、関連するプログラムの簡単な説明とともに、上記手順の各段階について説明する。なお、本章に關係あるディスク上のファイル一覧表およびこれらを格納してある磁気テープ（Volume 通番：ERATOF）の記録データを第3-1表にまとめておく。

3.2 新規プログラムの説明

ERATO-J (F版) では、概要の項で述べた要求を満たす為、ERATO 4, ERATO 5 の出力処理部に若干の改良を加え、更に関連した新プログラムを付け加えてある。主要な改良点は、次の3項目に要約できる。

- (1) ERATO 4 の出力データおよび入力データ等の記録をとり、データを整理する為のプログラム群 DIARY の追加。
- (2) 行列要素を除く全ての中間結果を MT に保存し再現する為の MTOUT, MTIN プログラムの追加。
- (3) プラズマ変位をフーリエ展開して表示したり、位置、運動エネルギーの空間分布を図形表示する為の ERATO 5 の改造と ERATO 6 の追加。

以下、これらの改造について簡単に記す。

3.2.1 DIARY プログラム

ERATO-J (F版) では計算結果の主要部と入力データ、計算条件等を保存するプログラム DIARY をそなえている。このプログラムはサブルーチン形式で ERATO 4 のサブルーチン VEKIT から呼ばれ主要計算結果を DIARY FILE に書き込む機能を持つ。書き込まれたデータは必要に応じて指定したものだけを出力することが可能である。

(1) DIARY プログラム

DIARY プログラムによって DIARY FILE に書き込まれる情報は第3-2表の通りである。ここで "Serial number" は DIARY プログラムが呼ばれるごとに順番につけられるもので計算結果を検索、図形表示するときの識別番号である。

"Check flag" については、CHANGE プログラムの項で説明する。

第3-2図に DIARY プログラムのソース・プログラム・リストを示す。

(2) INITIAL プログラム

DIARY FILE を使用して一連の計算結果を蓄積するには、その前に FILE 領域を確保して FILE を初期化せねばならない。この目的の為に使われるものが INITIAL プログラムである。第3-3図には、このプログラムのソース・プログラム・リストを示す。

(3) CHANGE プログラム

CHANGE プログラムは DIARY FILE の "Check flag" を書き換える為のプログラムである。DIARY プログラムを使用して計算結果を DIARY FILE に書き込むとき、

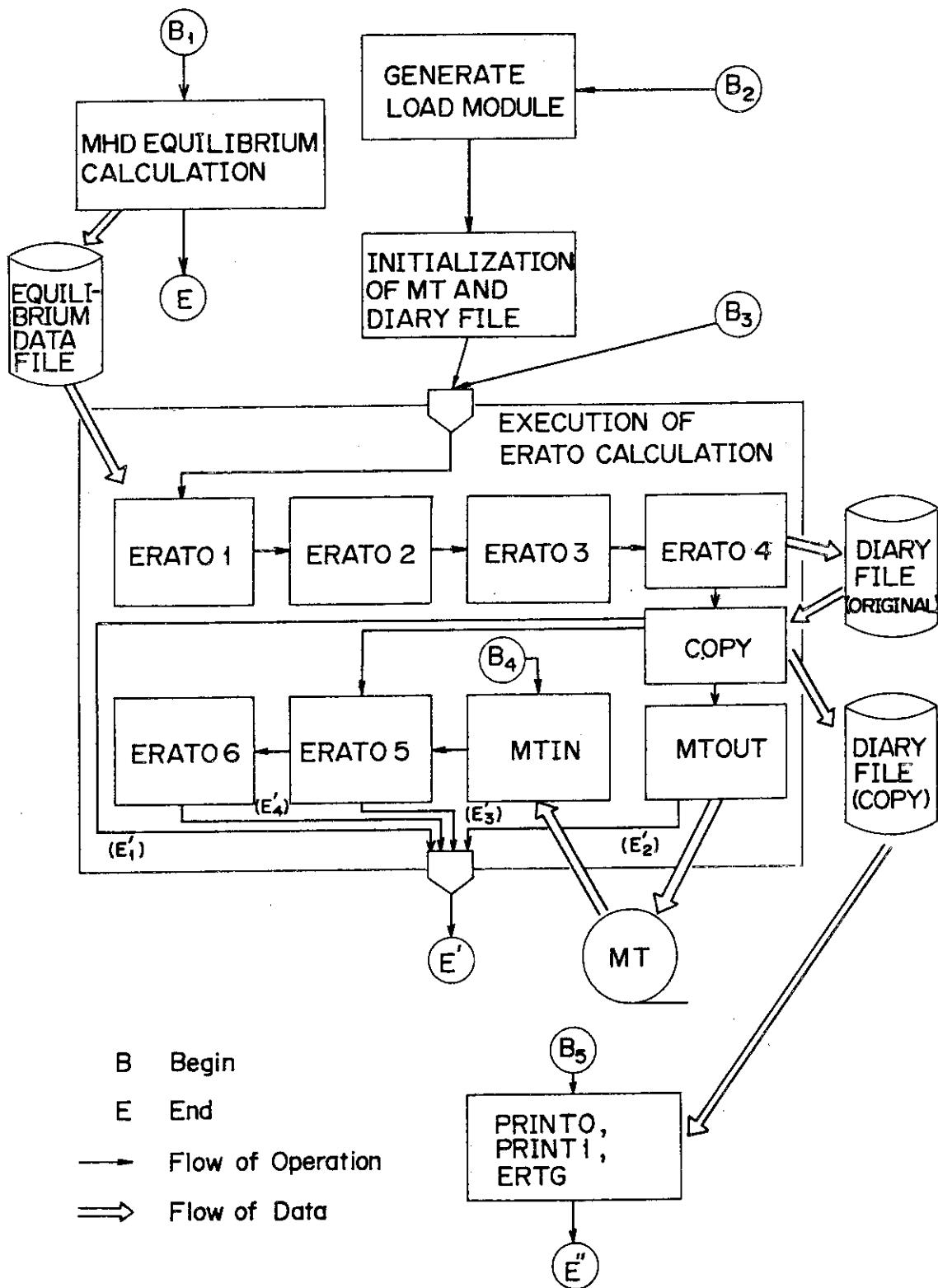


Fig.3-1 Standard procedure of the ERATO calculation.

Table 3-1. List of program files in the disks (a) and magnetic tape(b).

(a) List of program files in disks

Name of file	Kind of file	Content
J3331.ERTOF.FORT	GEM	COMMON file (EOS source)
J3331.ERT1FS.FORT	GEM	ERATO1 (ERATO-J F-v., EOS source)
J3331.ERT2F.FORT	GEM	ERATO2 (ERATO-J F-v., EOS source)
J3331.ERT3F.FORT	GEM	ERATO3 (ERATO-J F-v., EOS source)
J3331.ERT4F.FORT	GEM	ERATO4 (ERATO-J F-v., EOS source)
J3331.ERT5F.FORT	GEM	ERATO5 (ERATO-J F-v., EOS source)
J3331.ERT6F.FORT	GEM	ERATO6 (ERATO-J F-v.)
J3331.MTOUT	GEM	MTOUT (EOS source)
J3331.MTIN	GEM	MTIN (EOS source)
J3331.E1D50FS5.LOAD	PO	ERATO1; $N_{\psi}, N_{\chi} < 50$; 512×512 Equilibrium
J3331.E2D50F.LOAD	PO	ERATO2; $N_{\psi}, N_{\chi} < 50$
J3331.E3D50F.LOAD	PO	ERATO3; $N_{\psi}, N_{\chi} < 50$
J3331.E4D50F.LOAD	PO	ERATO4; $N_{\psi}, N_{\chi} < 50$
J3331.E5D70F.LOAD	PO	ERATO5; $N_{\psi}, N_{\chi} < 70$
J3331.E6.LOAD	PO	ERATO6
J3331.MTOUTD70.LOAD	PO	MTOUT; $N_{\psi}, N_{\chi} < 70$
J3331.MTIND70.LOAD	PO	MTIN; $N_{\psi}, N_{\chi} < 70$
J3331.EJCL50G.CNTL	PS	JCL program (Guess value)
J3331.EJCL50L.CNTL	PS	JCL program (Large job)
J3331.EJCL50M.CNTL	PS	JCL program (MT job)
J3331.EJCL50W.CNTL	PS	JCL program (Whole program job)
J3331.INITIAL.FORT	GEM	INITIAL
J3331.CHANGE.FORT	GEM	CHANGE
J3331.PRINT.FORT	GEM	PRINT (PRINT0, PRINT1)
J3331.ERTG.FORT	GEM	ERTG
J3331.EMTDUMPO.FORT	GEM	EMTDUMPO (EOS source)
J3331.INITIAL.LOAD	PO	INITIAL
J3331.CHANGE.LOAD	PO	CHANGE
J3331.PRINT0.LOAD	PO	PRINT0
J3331.PRINT1.LOAD	PO	PRINT1
J3331.ERTG.LOAD	PO	ERTG
J3331.EMTDUMPO.LOAD	PO	EMTDUMPO
J3331.GTPARM.LOAD	PO	GET PARAM
J3331.EMTDUMPO.CNTL	PS	JCL program for EMTDUMPO
J3331.EMTDUMP1.CNTL	PS	JCL program for EMTDUMP1
J3331.EMTDUMP2.CNTL	PS	JCL program for EMTDUMP2
J3331.TSS.CLIST	PO	Command procedure
J3331.JCL.CNTL	PO	JCL programs

JAERI-M 9040

SLS
S T A N D A R D L A B E L M T (ERATO) DATE(1980-07-30) TIME=09:15:16

FILE NO.	DATA-SET-NAME	BLKSIZE	LRECL	BLOCKS	RECFM	CREATION	DENSITY	VOL	JOB/STEP-NAME	UTILITY	MEMO --
1.	J2331.ERTOF.FORT	17256	0	18	U	1980/07/24 (THU)	1600	ERATO	F9C7918.../GEMBKUP	?	
2.	J3331.ERTIFS.FORT	1736	0	161	U	1980/07/24 (THU)	1600	ERATO	F907922 /GEMAKUP	?	
3.	J3331.ERT2F.FORT	1736	0	79	U	1980/07/24 (THU)	1600	ERATO	F907925 /GEMBKUP	?	
4.	J3331.ERT2F.FORT	1736	0	92	U	1980/07/24 (THU)	1600	ERATO	F907930 /GEMBKUP	?	
5.	J3331.ERT4F.FORT	17256	0	79	U	1980/07/24 (THU)	1600	ERATO	F907932 /GEMBKUP	?	
6.	J3331.ERTSF.FORT	1736	0	86	U	1980/07/24 (THU)	1600	ERATO	F907934 /GEMBKUP	?	
7.	J3331.ERTSF.FORT	17256	0	35	U	1980/07/24 (THU)	1600	ERATO	F907936 /GEMBKUP	?	
8.	J3331.ERTMIN.FORT	1736	0	15	U	1980/07/24 (THU)	1600	ERATO	F907938 /GEMBKUP	?	
9.	J3331.ERTMOUT.FORT	1736	0	11	U	1980/07/24 (THU)	1600	ERATO	F907940 /GEMBKUP	?	
10.	J3331.EJCL506	3120	60	2	FB	1980/07/29 (TUE)	1600	ERATO	F907958 /	?	
11.	J3331.EJCL50L	3120	60	3	FB	1980/07/29 (TUE)	1600	ERATO	F907958 /	?	
12.	J3331.EJCL50M	3120	60	3	FB	1980/07/29 (TUE)	1600	ERATO	F907958 /	?	
13.	J3331.EJCL50W	3120	60	3	FB	1980/07/29 (TUE)	1600	ERATO	F907958 /	?	
14.	J9079.DIARYINO	6212	6208	15	YES	1980/07/29 (TUE)	1600	ERATO	F907958 /	?	
15.	J3331.EMTDUMP0	3120	80	1	FB	1980/07/29 (TUE)	1600	ERATO	F907958 /	?	
16.	J3331.EMTDUMP1	3120	80	1	FE	1980/07/29 (TUE)	1600	ERATO	F907961 /	?	
17.	J3331.EMTDUMP2	3120	80	2	FE	1980/07/29 (TUE)	1600	ERATO	F907961 /	?	
18.	J3331.INITIAL.FORT	17256	0	6	U	1980/07/29 (TUE)	1600	ERATO	F907964 /GEMBKUP	?	
19.	J3331.CHANGE.FORT	1736	0	15	U	1980/07/29 (TUE)	1600	ERATO	F907964 /GEMBKUP	?	
20.	J3331.EMTDUMP0.FORT	1736	0	11	U	1980/07/29 (TUE)	1600	ERATO	F907964 /GEMBKUP	?	
21.	J3331.PRINT.FORT	17256	0	38	U	1980/07/29 (TUE)	1600	ERATO	F907964 /GEMBKUP	?	
22.	J3331.ERTG.FORT	17256	0	54	U	1980/07/29 (TUE)	1600	ERATO	F907967 /GEMBKUP	?	
23.	J3331.JCL.CNTL	1736	0	57	U	1980/07/29 (TUE)	1600	ERATO	F907967 /GEMBKUP	?	
24.	J3331.GTPARM.LOAD	19089	19085	5	VS	1980/07/29 (TUE)	1600	ERATO	F907970 /JSE COPY	?	

(b) List of Program files in the magnetic tape ERATO

Table 3-2 Parameters in DIARY FILE

Parameters	Meaning	Printed by	
		PRINT0	PRINT1
NO	Serial number	o	o
NCHEK	Check flag	o	o
ITIME	Time	x	o
IDATE	Date	x	o
DSN	Names of equilibrium file, and load modules of ERATOL, 2, 3, 4	x	o
NPSI	Radial mesh number (N_ψ)	o	o
NCHI	Poloidal mesh number (N_χ)	o	o
WNTORE	Toroidal mode number	o	o
REXT	Position of conducting shell (b/a)	o	o
AL(1)	Guess value of eigenvalue	o	o
AL(2)	Eigenvalue by normalization	o	o
AL(3)	Eigenvalue by Reyleigh quotient	o	o
NEG	Number of eigenvalues less than AL(1)	o	o
NCONV	Number of converged components of an eigenvector	x	o
NCOMP	Number of components of an eigenvector	x	o
NIT	Number of executed iterations	o	o
NITMAX	Maximum number of iterations	x	o
NDIVSI	Type of mesh divisions	o	o

FACOM USIV/F4 GEM VOLVOSE DATE 80.07.16 TIME 10.32.00 LIB=J9079.ERT4F.FORT

-NO.	-MODULE	-NAME	-DIARY	-LEVEL	-BLOCKS	-5-----6-----7-R-----8
-NO.9				7	3	DATE 80.02.22 TIME 22.24.48
			SUBROUTINE DIARY(AL,DSN)		00000100	
			IMPLICIT REAL*8(A-H,O-Z)		00000200	
			LOGICAL*1 DSN(80),IDSN(80)		00000300	
			*CALL COMIVI		00000400	
			*CALL COMPLT		00000410	
			*CALL NEWRUN		00000500	
			LOGICAL NLREG		00000600	
			DIMENSION		00000700	
			AL(3),ZAL(3),IDATE(2)		00000800	
C					00000900	
C			READ GLD DATASET		00000950	
C					00001000	
			REWIND 1		00001010	
			REWIND-50		00001020	
			DO 110 J=1,1000		00001100	
			READ(1,END=999)		00001200	
			NU,NCHEK,ITIME,IDATE,		00001300	
			IDSN,		00001400	
			INPSI,INCHI,ZNTURE,ZREXI,		00001500	
			ZAL,IINEG,INCUNV,INCUMP,INIT,NITMAX,NDIVSI		00001600	
			IF(NCHEK.EQ.0) GO TO 120		00001700	
110			CONTINUE		00001800	
			GO TO 998		00001900	
C					00002000	
C			READ TIME,DATE AND NAMELIST		00002050	
C					00002100	
120			CONTINUE		00002200	
			CALL TIME(ITIME)		00002300	
			CALL DATE(IDATE)		00002400	
C			NSAVE = 8		00002500	
			REWIND NSAVE		00002600	
			READ(NSAVE,NEWRUN)		00002700	
C					00002800	
C			WRITE RECORDS		00002900	
C					00003000	
			BACK SPACE 1		00003100	
			NCHEK=1		00003200	
			NCOMP=NL		00003300	
			WRITE(1)		00003400	
			NU,NCHEK,ITIME,IDATE,		00003500	
			DSN,		00003600	
			NPSI,NCHI,WNTURE,REXI,		00003700	
			AL,NEG,NCUNV,NCUMP,INIT,NITMAX,NDIVSI		00003800	
			WRITE(50)		00003900	
			NU,NCHEK,ITIME,IDATE,		00003910	
			DSN,		00003920	
			NPSI,NCHI,WNTURE,REXI,		00003930	
			AL,NEG,NCUNV,NCUMP,INIT,NITMAX,NDIVSI		00003940	
C					00003950	
C			WRITE ZERO DATA		00004000	
C					00004010	
					00004020	

(続く)

Fig. 3-2 Source program list of DIARY (EOS source).

(続き)

NCHEK	= 0	00004030
ITIME	= 0	00004040
IDATE(1)	= 0	00004050
IDATE(2)	= 0	00004060
NPS1	= 0	00004070
NCHI	= 0	00004080
XNTORE	= 0.0	00004090
REXT	= 0.0	00004100
AL(1)	= 0.0	00004110
AL(2)	= 0.0	00004120
AL(3)	= 0.0	00004130
NEG	= 0	00004140
NCUNV	= 0	00004150
NCUMP	= 0	00004160
NIT	= 0	00004170
NITMAX	= 0	00004180
NLIVSI	= 0	00004190
DU 210 J=1,80		00004200
DSN(J)=* *		00004210
210 CONTINUE		00004220
NEND=1000-NU		00004230
		00004240
DU 220 J=1,NEND		00004250
NU=NU+1		00004260
WRITE(1)		00004270
NU,NCHEK,ITIME,DATE,		00004280
DSN,		00004290
NPS1,NCHI,XNTORE,REXT,		00004300
AL,NEG,NCUNV,NCUMP,NIT,NITMAX,NLIVSI		00004310
220 CONTINUE		00004320
RETURN		00004330
C-----		00004340
C-----		00004350
999 WRITE(6,600)		00004360
RETURN		00004370
998 WRITE(6,601)		00004380
RETURN		00004390
600 FORMAT(IH0,'END OF FILE FT01FC01 AT DIARY')		00004400
601 FORMAT(IH0,'FILE SPACE OVER AT DIARY')		00004410
END		00004420

FACOM OSIV/F4 GEM, V02L08E DATE 80.07.15 TIME 18.31.15 LIB=J9079.INITIAL.FORT

-----1-----2-----3-----4-----5-----6-----7-R-----8
 NO.1 MODULE NAME INITIAL BLOCKS 2
 LEVEL 10 DATE 80.02.09 TIME 09.48.01

```

IMPLICIT REAL*8(I,A-H,C-Z)          00100
DIMENSION                         00000200
  • AL(3),IDATE(2)                 00200
  • LOGICAL*1 DSN(85)               00400
C-----                                00000500
C           SET INITIAL VALUE        00000600
C                                     00000700
C
  NC    = 0                           00000800
  NCHEK = 0                           00000900
  ITIME = 0                           01000
  IDATE(1) = 0                        01100
  IDATE(2) = 0                        01200
  NPSI  = 0                           00001200
  NCHI  = 0                           00001400
  WNTCRE= 0.0                         00001500
  REAT  = 0.0                         00001600
  AL(1) = 0.0                         00001700
  AL(2) = 0.0                         00001800
  AL(3) = 0.0                         00001900
  NEG   = 0                           00002000
  NCENV = 0                           00002100
  NCMP  = 0                           00002200
  NIT   = 0                           00002300
  NITMAX= 0                           00002400
  N     = 0                           00002500
  DO 110 J=1,80                      00002600
  •      DSN(J)=*                   02700
  110  CONTINUE                      00002800
C-----                                00002900
C           WRITE RECORDS          00003000
C                                     00003100
C
  DO 210 J=1,1000                    00003200
  •      NC=NC+1                     00003300
  •      WRITE(1),                  00003400
  •      NC,NCHEK,ITIME,IDATE,
  •      CSN,
  •      NPSI,NCHI,WNTCRE,REAT,
  •      AL,NEG,NCENV,NCMP,NIT,NITMAX,N
  210 CONTINUE                      00003500
  •      WRITE(6,800)                04000
  600  FORMAT(1HC,' ----- INITIALIZATION NORMAL ENDS -----')
  STOP
  END

```

Fig.3-3 Source program list of INITIAL.

"Check flag" の値は常に1がセットされている。しかし DIARY FILE に記録された計算結果が常に意味があるものとは限らない。たとえば入力データの誤りや固有値が逆べき乗法計算で十分収束していない場合などが考えられる。このようなデータに対して "Check flag" を1以外の数にすることによって後に DIARY FILE を参照する場合に他のデータと区別することができる。例えば固有値図形表示プログラム ERTG は、"Check flag" が1以外の固有値は表示しないように作成されているので ERTG を使って描いた収束曲線からメッシュ数無限大の成長率を推定するときのむだな混乱をさけることができる。

第3-4図には CHANGE プログラムのソース・プログラム・リストを示す。

(4) PRINT 0 プログラム

DIARY FILE の参照用プログラムで、計算結果の主要部の中で特に重要なデータについてのみ作表する。作表の対象となるパラメータは、第3-2表に示すとおりである。PRINT 0 のソース・プログラム・リストは第3-5図のとおりである。

(5) PRINT 1 プログラム

DIARY FILE の参照用プログラムで、計算結果の主要部のデータを全て作表する。作表の対象になるパラメータは第3-2表のとおりである。PRINT 1 のソース・プログラム・リストは第3-6図に示す。

(6) ERTG プログラム

DIARY FILE の参照用プログラムで、固有値を $(\text{メッシュ数})^{-2}$ の関数として図形表示するプログラムである。第3-7図にソース・プログラム・リストを示す。

3.2.2 MTOUT および MTIN プログラム

ERATO 計算を実行して最終的に得たい情報は、MHD 不安定性の成長率、即ち、行列の固有値であるから、通常は、DIARY FILE に格納されているデータを利用するだけで十分である。不稳定性モードのパターンや運動エネルギー、位置エネルギーの空間分布等のデータを常に出力すると出力リストの量が膨大となり、無駄であるばかりか検索にも手間がかかり非能率的でさえある。更に、最終結果に問題点が見出され、途中で使ったメトリック量等を参照したい場合があるが、これら中間データを常に出力するとその出力リストの量は極端に増加する。一方、前章の記述からもわかるように、大メッシュ数計算ではシステム資源量が大きい事を考えると、後に問題点を見出した時にこのような計算を2度にわたって実行することはきわめて無駄である。この為、ERATO-J(F版)では、ERATO 4 実行後、中間結果も含めた全ての数値を磁気テープ上に保管し、後にいつでも利用が可能であるようにしてある。ERATO 4 終了後に磁気テープに書き込む為のプログラムが MTOUT で、磁気テープから読み出す為のプログラムが MTIN である。MTOUT や MTIN を使用せず従来のジョブ・ステップの順序で計算が実行できることはもちろんである(第3-1図参照)。

第3-8、3-9図に、MTOUT、MTIN のソース・プログラム・リストを示す。

FACOM OSIV/F4 GEM VOLCOSE DATE 80.07.16 TIME 11.52.40 L18=J9079.FPRINT.FCRT

NO.1	MODULE NAME CHANGE	BLOCKS	6	7-R-----8	
	LEVEL	DATE	80.02.20	TIME	11.26.26
<hr/>					
	IMPLICIT REAL*8(A-H,O-Z)			00000100	
	DIMENSION			00000200	
.	AL(3),IDATE(2)			00000400	
	LOGICAL*1 DSN(80)			00000450	
.	IICH=0			00000452	
	WRITE(6,620)			00000453	
.	READ(5,*) IING			00000455	
	WRITE(6,600)			00000500	
.	DO 110 J=1,1000			00000600	
	READ(1,END=999)			00000900	
.	NO,NCHEK,ITIME,IDATE,			00001100	
.	DSN,			00001200	
.	NPSI,NCHI,WNTGRE,REXT,			00001300	
.	AL,NEG,NCCNV,NCOMP,NIT,NITMAX,N			00001400	
	IF(IICH.EQ.10) GO TO 888			00001402	
	IF(NG.LT.IING) GO TO 888			00001403	
.	IF(NCHEK.EQ.0) GO TO 888			00001405	
	WRITE(6,610) NO,NPSI,NCHI,AL(2),AL(3),NIT,NEG,AL(1),			00001800	
.	REXT,WNTGRE,(DSN(J0),JD=7,14)			00001840	
	WRITE(6,630)			00001841	
	READ(5,*) IICH			00001842	
	IF(IICH.NE.1) GO TO 888			00001843	
	READ(5,500) NCHEK			00001845	
888	WRITE(2)			00001855	
.	NC,NCHEK,ITIME,IDATE,			00001860	
.	DSN,			00001870	
.	NPSI,NCHI,WNTGRE,REXT,			00001880	
.	AL,NEG,NCCNV,NCOMP,NIT,NITMAX,N			00001890	
110	CONTINUE			00001900	
	WRITE(6,640)			00001901	
	STOP			00001904	
600	FORMAT(1H1,'NO',1X,'NPSI',1X,'NCHI',2X,'NORMALIZATION',2X, .RAYLEIGH',4X'NIT',1X'NEG',3X'GUESS',10X'REXT',6X'N',8X'EQUAL')			00001905	
620	FORMAT(1H0,'INPUT UPDATE FIRST NUMBER')			00001906	
630	FORMAT(1H ,5X, '0 --- NO UPDATE 1 --- UPDATE 10 --- END OF UPDATE')			00001907	
610	FORMAT(1H ,13,2X,2(1Z,2X),2(1X,E13.5),2(2X,1Z),1X,E13.5,2X ,F7.3,2X,F5.1,2X,EA1)			00001910	
640	FORMAT(1H , 'END OF UPDATE')			00001915	
500	FORMAT(15)			00001916	
999	CONTINUE			00001918	
	WRITE(6,900)			00001920	
	STOP			00001922	
900	FORMAT(1H0,10X,' END OF FILE --- FT01F001')			00001923	
	END			00001924	
				00001930	

Fig.3-4 Source program list of CHANGE.

FACOM LS1V/P4 GEM V02L08 DATE 80.07.15 TIME 18.32.35 LIB=J9079.PRINT.FORT

NC#3	MODULE NAME	PRINT0	BLOCKS	6	-----	7-R-----8
	LEVEL	41	DATE	80.02.20	TIME	11.26.26

```

IMPLICIT REAL*8(A-H,O-Z)          00000100
DIMENSION                         00000200
AL(2),IDATE(2)                   00000300
LOGICAL LSBN(60)                 00000400
NAMELIST /FRI/ WN, REXT, NC    00000410
WN = 0.0                          00000500
REXT = 0.0                         00000600
ND = 0                            00000700
READ(5,FRI)                      00000800
WRITE(e,800)                      00000900
DO 110 J=1,1000                  00001000
  READ(1,END=999)                 00001100
  INE,NCREK,ITIME,IDATE,        00001200
  LSBN,                           00001300
  NCHI,NCHKE,ZREXT,              00001400
  AL,NEG,NCNV,NCEMP,NIT,NITMAX,NLIVSI   00001500
  IF(NCHER.EQ.0) GO TO 999         00001600
  IF(WN .NE. 0.0.AND.WNTER.EQ.WN ) GO TO 110
  IF(REXT.NE.0.0.AND.ZREXT .NE.REXT) GO TO 110
  IF(INC.LT.ND)                   GO TO 110
  WRITE(e,810) INE,NCHI,AL(2),AL(2),NIT,NEG,AL(1),
  ZREXT,WNTCKE,NCREK,NLIVSI,(LSN(JD),JD=1,14)
  IF(INC/5*5.EQ.1) WRITE(e,700)      00002100
110 CONTINUE                      00002200
600 FORMAT(IH1,'NC',1A'PLT',1A'NCHI',2A'NORMALIZATION',4A,
     ,1A'RAYLEIGH',4A'NIT',1A'NEG',5A'Guess',5A'ReXT',5A'N',2A'NCHEK',
     ,1A'NLIVSI',4A'EGL')           00002300
610 FORMAT(IH1,I3,2E2(1A,2A),2(1A,E13.5),2(2A,1A),1A,E13.5,2A
     ,F7.3,2A,F5.1,2A,12,2A,12,2A,14A1) 00002400
620 FORMAT(IH1)                     00002500
630 FORMAT(IH1)                     00002600
640 FORMAT(IH1)                     00002700
650 FORMAT(IH1)                     00002710
660 FORMAT(IH1)                     00002800
670 FORMAT(IH1)                     00002900

```

Fig.3-5 Source program list of PRINT0.

FACCM OSIV/F4 OEM VOZLOSE DATE 80.07.18 TIME 16.43.19 L16=J9079.PRINT.FORT

-----*----1----*----2----*----3----*----4----*----5----*----6----*----7-R----*----8
 NO.4 MODULE NAME PRINT1 BLOCKS 7
 LEVEL 47 DATE 80.02.20 TIME 11.26.28

```

IMPLICIT REAL*8(A-H,O-Z)          00000010
DIMENSION                         00000020
AL(3),IDATE(2),NTIME(2)          00000030
LOGICAL*1 DSN(80)                00000040
NAMELIST /PRI/ WN, REXT, NC      00000050
WN = 0.0                          00000060
REXT = 0.0                         00000070
NC = 0                            00000080
READ(5,PRI)
WRITE(6,600)
DO 110 J=1,1000
  READ(1,END=999)
  INC,NCHEK,ITIME,IDATE,
  DSN,
  NPSI,NCHI,WNTRE,ZREXT,
  AL,NEG,NCENV,NCLMP,NIT,NITMAX,NDIVSI
  IF(NCHEK.EQ.0) GO TO 999
  IF(WN .NE. 0.0.AND.WNTRE.NE.WN ) GO TO 110
  IF(REXT.NE.0.0.AND.ZREXT .NE.REXT) GO TO 110
  IF(INC.LT.NC) GO TO 110
C
  IS = ITIME/1000                 00000220
  IM = IS / 60                     00000230
  IS = IS - 60*IM                00000240
  IH = IM / 60                   00000250
  IM = IM - 60*IH                00000260
  ENCODE(8,777,NTIME) IH,IM,IS   00000270
C
  777    FORMAT(12,':',12,':',12)
  WRITE(5,610) INC,NPSI,NCHI,AL(2),AL(2),NIT,NITMAX,NEG,AL(1),
  ZREXT,WNTRE,NCHEK,NDIVSI,NCENV,NCLMP,NTIME,IDATE,
  (DSN(JD),JD=1,80)
  IF(INC/5*5.EQ.INC) WRITE(6,700)
110    CONTINUE
500    FORMAT(1H1,'NC',1X'NPSI',1X'NCHI',2X'NORMALIZATION',4X,
  'RAYLEIGH',4X'NIT',1X'MAX',1X'NEG',5A'GUESS',
  6X'REXT',7X'N',2X'NCHEK',
  1X'SI',2X'NCENV',2X'NCLMP',5X'TIME',6X'DATE')
610    FORMAT(1H ,13,2X2(12,2X),2(1X,E13.5),2(2X,12),1X,E13.5,2X
  ,F7.3,2X,F5.1,2X,2(12,2X),2(16,1X),2X,
  ,2(2A4,2X)/5X,'DSN = ',00A1/)
700    FORMAT(1H0)
999    STOP
END

```

Fig.3-6 Source program list of PRINT1.

FACOM DSIV/F4 GEM V02L08E DATE 80.07.16 TIME 13.33.24 LIE=J9079.ERTG.FORT

NO.8	MODULE NAME MAIN	BLOCKS	13	7-R	8
LEVEL	69	DATE	80.03.05	TIME	09.51.51

```

IMPLICIT REAL*8(A-H,O-Z)          00000010
DIMENSION AL(3),IDATE(2)          00000020
LOGICAL NLEND .1L               00000030
LOGICAL*1 DSN(80),CLDDSN(80,11),ZDSN(80) 00000040
INTEGER ISYM(11)                 00000050
DATA CLDDSN /8E0*1H /           00000060
DATA ISYM/4,3,11,12,0,1,2,5,7,9,10/ 00000070
NAMELIST /COM/
NLEND,REXT,WN,XMIN,XMAX,YMIN,YMAX 00000080
XF,YF,X0,Y0,IS,NO,DSN            00000090
REAL*4 X(500), Y(500)             00000100
REAL*4 XX,YY                      00000110
C-----00000120
C-----00000130
C-----00000140
C-----00000150
CALL ARGUS(3)                   00000160
XF = 330.0                       00000170
YF = 240.0                       00000180
X0 =-15.0                         00000190
Y0 = 0.0                          00000200
XMIN = 0.0                         00000210
XMAX = 30.0                        00000220
YMIN =0.0                          00000230
YMAX = 1.0                          00000240
IS = 4                            00000250
WN = 0.0                           00000260
REXT=-1.0                         00000270
NLEND=.FALSE.                     00000280
90 CONTINUE                        00000290
IDATA = 0                          00000300
DO 80 J=1,80                      00000310
80 DSN(J) = ? ?                  00000320
READ(5,COM)                        00000330
ZMAX = 1.0/(XMAX**2)              00000340
IF(NLEND) GO TO 999               00000350
C-----00000360
C-----00000370
C-----00000380
REWIND 1                           00000390
CALL DSNUM(ASN,NDSN)              00000400
IF(REXT.NE.-1.0.AND.REXT.LE.1.0) REXT=1.0 00000410
DO 210 J=1,500                    00000420
    READ(1,END=220)                00000430
    IN0,NCHEK,ITIME>IDATE,        00000440
    ZDSN,                          00000450
    NPS1,NCHI,WNTORE,ZREXT,       00000460
    AL,NEG,NCCNV,NCDMP,NIT,NITMAX,N 00000470
    IF(ZREXT.LE.1.0) ZREXT=1.0    00000480
    IF(NCHEK.EQ.0) GO TO 220     00000490
    IF(IN0.LT.NO) GO TO 210      00000500
    IF(NCHEK.NE.1) GO TO 210      00000510
    IF(WN .NE. 0.0.AND.WNTORE.NE.WN ) GO TO 210 00000520

```

(続く)

Fig.3-7 Source program list of ERTG.

(続き)

```

1F(REST.NE.-1.0.AND.ZREXT .NE.REXT) GO TO 210      00000530
DO 230 I=1,NDSN                                     00000540
IF(DSN(I).NE.1 .AND.ZDSN(I).NE.DSN(I))GO TO 210      00000550
230 CONTINUE                                         00000560
IDATA = IDATA + 1                                  00000570
X(IDATA) = 1.0/FLOAT(NPSI**Z)                      00000580
Y(IDATA) = -AL(Z)                                 00000590
1F(IDATA.GT.500) GO TO 999                         00000600
210 CONTINUE                                         00000610
220 CONTINUE                                         00000620
CALL STCR1D("ERATO",IDATA,1,X,1,Y,1DATA)          00000630
C-----                                             00000640
C DRAW GRAPH                                       00000650
C
CALL ARGUS(888)                                    00000670
CALL FRAME("ERATO",1,-6,1)                         00000680
CALL DATAAPT("ERATO",-3)                           00000690
CALL SCALE1("ERATO",1S,-IS,XF,YF,XMIN,ZMAX,YMIN,YMAX) 00000700
CALL CURVE ("ERATO",0)                            00000710
CALL ORIGIN("ERATO",X0,Y0)                         00000720
CALL ONEDIM("ERATO",0)                            00000730
C-----                                             00000740
C
C DRAW MESH NUMBER                                00000750
C
DO 280 J=20,80                                     00000760
ZX=1.0/FLOAT(J*J)                                 00000770
CALL CANSYM(ZX,-3.0,3.0,12,0.0,-1)                00000780
CALL CANSYM(ZX,243.0,3.0,13,180.0,-1)              00000790
280 CONTINUE                                         00000800
DO 290 J=20,80,10...                               00000810
ZJ=J
ZX=1.0/FLOAT(J*J)                                 00000820
CALL CANSYM(ZX,-3.0,3.0,2,0.0,-1)                 00000830
CALL CANSYM(ZX,243.0,3.0,2,180.0,-1)               00000840
CALL CANNUM(ZX-0.00001,-3.0,3.0,ZJ,0.0,-1)        00000850
290 CONTINUE                                         00000860
C-----                                             00000870
C DRAW SYMBOL                                      00000880
C
C INITIAL CLEAR                                   00000890
C
DO 300 J=1,260                                     00000900
OLDDSN(J,1)=1                                     00000910
300 CONTINUE                                         00000920
C-----                                             00000930
C READ FILE                                       00000940
C
ICLD = 0                                           00000950
REWIND 1                                         00000960
DO 440 J=1,1000                                    00000970
READ(1,END=450)                                     00000980
IN0,NCHEK,ITIME,1DATE,                            00000990
ZDSN,                                              00001000
NPSI,NCHI,WNTORE,ZREXT,                           00001010
AL,NEG,NCONV,NCOMP,NIT,NITMAX,N                  00001020

```

(続く)

(続き)

	IF(NCHEK.EQ.0)	GO TO 450	00001100
	IF(NCHEK.NE.1)	GO TO 440	00001110
	IF(WN_.NE._0.0.AND.WNTCKE.NE.WN_)	GO TO 440	00001120
	IF(REAL.NE.-1.0.AND.ZREXT_.NE.REXT)	GO TO 440	00001130
	IF(INO.LT.NO)	GO TO 440	00001140
	DO 470 I=1,NUSN		00001150
	IF(ZDSN(1).NE.' '.AND.ZDSN(1).NE.CSN(1))GO TO 440		00001160
470	CONTINUE		00001170
	XX = 1.0/FLAGAT(NPSI**2)		00001180
	YY = -AL(2)		00001190
	DO 410 JJ=1,ICLD		00001200
	11=JJ		00001210
	CALL CHDSN(ZDSN,OLDDSN(1,JJ),1L)		00001220
	IF(1L)	GO TO 430	00001230
C	SAVE DATA SET NAME		00001240
C			00001250
C			00001260
410	CONTINUE		00001270
	ICLD = ICLD + 1		00001280
	IF(ICLD.GT.11)	GO TO 999	00001290
	II = ICLD		00001300
	CALL CDFYDS(ZDSN,OLDDSN(1,ICLD))		00001310
430	CALL TANSYM(XX,YY,3.0,ISYM(II),0.0,-1)		00001320
440	CONTINUE		00001330
450	CONTINUE		00001340
	ZX=1.0		00001350
	ZY=230.0		00001360
	ZDY=5.0		00001370
	DO 460 JD=1,ICLD		00001380
	CALL SYMBOL(ZX,ZY,3.0,OLDDSN(7,JD),0.0,e)		00001390
	CALL SYMBOL(ZX+29.0,ZY+1.8,3.0,ISYM(JD),0.0,-1)		00001400
	ZY = ZY - ZDY		00001410
460	CONTINUE		00001420
	CALL PLCT(0.0,0.0,444)		00001430
	CALL PLCT(0.0,0.0,777)		00001440
	CALL PLCT(0.0,0.0,666)		00001450
	CALL DELETE('ERATO')		00001460
	CALL TANAKA(0.0,0.1,ZAX,ZSFX,ZAY,ZSFY,0)		00001470
	GO TO 90		00001480
999	CONTINUE		00001490
	STOP		00001500
	END		00001510

FACOM CSIV/F4 GEM V02LC8E DATE 80.07.18 TIME 18.29.27 LIB=J9079.ERT41.FORT

```

-----1-----2-----3-----4-----5-----6-----7-R-----8
NO.1 MODULE NAME AMAIN          BLOCKS   3
          LEVEL     8      DATE 80.04.28 TIME 15.19.19

*DECK FMAIN
  IMPLICIT REAL*8(A-H,O-Z)          00000100
*CALL COMBL5                         00000150
*CALL COMEGU                         00000200
*CALL NEWRUN                         00000250
                                      00000300
                                      00000350
  LOGICAL NRREG                      00000400
  INTEGER IDATE(2)                   00000500
  LOGICAL*1 DSN(80)                  00000600
  REAL*8 AL(3)                       00000700
C-----                                     00000800
  NDES = 16                           00000900
  NSAVE = 8                           00001000
C-----                                     00001100
C----- EIGEN VALUE INFORMATION        00001200
C-----                                     00001300
  READ(50)
  •  NG,NCHEK,ITIME,IDATE,DSN,NPSI,NCHI,WNTORE,REXT,AL,NEG,NCCNV, 00001400
  •  NCOMP,NIT,NITMAX,NDIVSI           00001500
  •  WRITE(60)
  •  NG,NCHEK,ITIME,IDATE,DSN,NPSI,NCHI,WNTORE,REXT,AL,NEG,NCCNV, 00001600
  •  NCOMP,NIT,NITMAX,NDIVSI           00001700
  •  00001800
C-----                                     00001900
C----- NAMELIST (NSAVE=8)             00002000
C-----                                     00002100
  READ (NSAVE,NEWRUN)                 00002200
  WRITE(60 ,NEWRUN)                  00002300
C-----                                     00002400
C----- EIGEN VECTOR (NDS=9)          00002500
C-----                                     00002600
  NDX = 2*(NCHI+1)*(3*NPSI+1)       00002700
  READ(NDS) (X(I),I=1,NDX)          00002800
  WRITE(60 ) (X(I),I=1,NDX)         00002900
C-----                                     00003000
C----- MAPPE'D EQUILIBRIUM          00003100
C-----                                     00003200
  DO 110 JPSI= 1,NPSI               00003300
    READ (MEQ) ((EQ(I,J),I=1,18),J=1,NCHI) 00003400
    WRITE(60 ) ((EQ(I,J),I=1,18),J=1,NCHI) 00003500
  110  CONTINUE                      00003600
C-----                                     00003700
C----- PLOTTING QUANTITIES          00003800
C-----                                     00003900
  DO 210 JPSI=1,NPSI+1              00004000
    READ (NDES) (CR (JPSI,J),J=1,NCHI) 00004100
    READ (NDES) (CZ (JPSI,J),J=1,NCHI) 00004200
    READ (NDES) (CNR(JPSI,J),J=1,NCHI) 00004300
    READ (NDES) (CNZ(JPSI,J),J=1,NCHI) 00004400
    WRITE(60 ) (CR (JPSI,J),J=1,NCHI) 00004500
    WRITE(60 ) (CZ (JPSI,J),J=1,NCHI) 00004600
    WRITE(60 ) (CNR(JPSI,J),J=1,NCHI) 00004700
    WRITE(60 ) (CNZ(JPSI,J),J=1,NCHI) 00004800
  210  CONTINUE                      00004900
  STOP                            00005000
  END                            00005100

```

Fig. 3-8 Source program list of MTOUT (EOS source).

FACOM DSIV/F4 GEM VC2L0BE DATE 80.07.18 TIME 18.29.06 LIS=J9079.ERTMTIN.FORT
 -----1-----2-----3-----4-----5-----6-----7-----8
 NO.1 MODULE NAME AMAIN BLOCKS 3
 LEVEL 12 DATE 80.04.28 TIME 15.19.19

```
*DECK FMAIN
  IMPLICIT REAL*8(A-H,D-Z)          00000100
*CALL COMBL5                         00000200
*CALL COMEQU                         00000300
*CALL NEWRUN                         00000400
  LOGICAL NLREQ                      00000500
  INTEGER IDATE=2                     00000600
  LOGICAL*1 DSN(80)                   00000700
  REAL*8 AL(3)                       00000800
  00000900
C-----00001000
  NDES = 16                          00001100
  NSAVE = 8                           00001200
  READ(5,*),INU                      00001300
C-----00001400
C----- EIGEN VALUE INFORMATION      00001500
C-----00001600
  50  CONTINUE                        00001700
  READ(60)                            00001800
  .  NO,NCHK,ITIME,ILATE,DSN,NPSI,NCHI,WNTORE,REXT,AL,NEG,NCONV, 00001900
  .  NCMP,NIT,NITMAX,NDIVSI          00002000
C-----00002400
C----- NAMELIST (NSAVE=8)            00002500
C-----00002600
  READ(60,*,NEWRUN)                  00002700
C-----00002800
  IF(1NG.NE.NO)                      00002900
C-----00003000
  WRITE(50)
  .  NO,NCHK,ITIME,ILATE,DSN,NPSI,NCHI,WNTURE,REXT,AL,NEG,NCONV, 00003100
  .  NCMP,NIT,NITMAX,NDIVSI          00003200
C-----00003300
  WRITE(NSAVE,NEWRUN)                00003400
C-----00003500
C----- EIGEN VECTOR (NDS=9)          00003600
C-----00003700
  NDX = 2*(NCHI+1)*(3*NPSI+1)       00003800
  READ(60)(A(I),I=1,NDX)             00003900
  WRITE(NDS)(A(I),I=1,NDX)           00004000
C-----00004100
C----- MAPPED EQUILIBRIUM           00004200
C-----00004300
  DO 110 JPSI= 1,NPSI               00004400
  READ(60)((EQ(I,J),I=1,18),J=1,NCHI) 00004500
  WRITE(MEQ)((EQ(I,J),I=1,18),J=1,NCHI) 00004600
  110  CONTINUE                      00004700
C-----00004800
C----- PLOTTING QUANTITIES          00004900
C-----00005000
  DO 210 JPSI=1,NPSI+1              00005100
  READ(60)(CR(JPSI,J),J=1,NCHI)     00005200
  READ(60)(CZ(JPSI,J),J=1,NCHI)     00005300
  READ(60)(CNR(JPSI,J),J=1,NCHI)    00005400
```

(続く)

Fig. 3-9 Source program list of MTIN (EOS source).

(続き)

```

      READ(60) (CNZ(JPSI,J),J=1,NCHI)          00005500
      WRITE(NDES) (CR (JPSI,J),J=1,NCHI)        00005600
      WRITE(NDES) (CZ (JPSI,J),J=1,NCHI)        00005700
      WRITE(NDES) (CNR(JPSI,J),J=1,NCHI)        00005800
      WRITE(NDES) (CNZ(JPSI,J),J=1,NCHI)        00005900
  210   CONTINUE                                00006000
C----- 00006200
C           WRITE INFORMATION                  00006300
C                                               00006400
C           WRITE(6,600) NO                     00006500
  600   FORMAT(1H1,'NO=',I5)                   00006600
      GO TO 410                                00006610
C----- 00006700
C           SKIP DATA                         00006800
C                                               00006900
  310   CONTINUE                                00007000
      READ(60)                                 00007100
      DO 320 JPSI=1,NPSI                      00007200
      READ(60)
  320   CONTINUE                                00007300
      DO 330 JPSI=1,NPSI+1                    00007400
      READ(60)
      READ(60)
      READ(60)
      READ(60)
  330   CONTINUE                                00007500
      GO TO 50                                 00007600
C----- 00007700
C           STOP                            00007800
C                                               00007900
  410   CONTINUE                                00008000
      STOP                                     00008100
      END                                      00008200
C----- 00008300
C----- 00008400
      00008500
      00008600
      00008700

```

3.2.3 ERATO 6 プログラム

ERATO 6 プログラムは、出力専用プログラムであって、本来の ERATO の構造によれば、ERATO 5 の一部に属すべきものであるが、新規プログラムである事を明確にしておく為、ERATO-J(F版)では、別プログラムとして構成されている。なお、これに関連して、ERATO 5においても若干の改造が加えられている。

ERATO 5 の改造か所および ERATO 6 の内容は下に示すとおりである。

(1) ポロイダル・モード構造の表示

プラズマ変位のポロイダル・モード構造を見る為に ψ 方向変位 (ψ) をポロイダル座標 χ に関するフーリエ展開 (ERATO 5) し、これを $\sqrt{\psi}$ の関数として図形表示 (ERATO 6) する。なお、図形表示されるポロイダル・モード数 (m) は、次の範囲にある。

$$1 \leq m \leq [n q_s]_{\text{Gauss}} + 1, \quad (3-1)$$

ここで、 $[]_{\text{Gauss}}$ は、ガウス記号、 n 、 q_s はトロイダル・モード数および表面での安全係数である。

(2) 位置エネルギー、運動エネルギーの表示

磁気面 ($\sqrt{\psi}$) の関数として、位置エネルギー (W) と運動エネルギー (K)，および、それぞれに含まれる各種の寄与を図形表示する。即ち、

$$W = W_1 + W_2 + W_3 + W_4 + W_5, \quad (3-2)$$

$$K = K_1 + K_2 + K_3, \quad (3-3)$$

ここで

$$W_1 = \frac{1}{2} \int \frac{dS d\chi}{S} \cdot \frac{2\psi T^3}{q R^4 B_p^2} |F(X)|^2, \quad (3-4)$$

$$W_2 = \frac{1}{2} \int \frac{dS d\chi}{S} \cdot \frac{T^3}{2q \psi_s} \left| \frac{\partial X}{\partial S} + \frac{\partial V}{\partial \chi} \right|^2, \quad (3-5)$$

$$\begin{aligned} W_3 = & \frac{1}{2} \int \frac{dS d\chi}{S} \cdot \frac{q T R^2 B_p^2}{2\psi} \left| S \frac{\partial V}{\partial \varphi} - \frac{S}{T} \frac{\partial}{\partial S} \left(\frac{T}{q} X \right) \right. \\ & \left. + \frac{2\psi J_T}{q R B_p^2} X - \frac{\vec{\nabla} \psi \cdot \vec{\nabla} V}{|\vec{\nabla} \psi|^2} \psi F(X) \right|^2, \end{aligned} \quad (3-6)$$

$$W_4 = \frac{1}{2} \int \frac{dS d\chi}{S} \frac{r_p T R^2}{2\psi_s q} \left| \frac{1}{R^2} \frac{\partial}{\partial S} (R^2 \chi) + \frac{1}{R^2} \frac{\partial}{\partial \chi} (R^2 V) + \frac{2}{R^2} F(R^2 Y) \right|^2, \quad (3-7)$$

$$W_5 = - \frac{1}{2} \int \frac{dS d\chi}{S} \frac{4\psi T R^2}{q} \left\{ \left(\frac{J_T}{R B_p} \right)^2 - \frac{J_T}{R B_p} \frac{\vec{\nabla} \psi \cdot \vec{\nabla} B_p}{|\vec{\nabla} \psi|^2} \right\}$$

$$+ \frac{T}{R^3} \frac{dT}{d\psi} \frac{\vec{\nabla}\psi \cdot \vec{\nabla}R}{|\nabla\psi|^2} \} \cdot X^2 , \quad (3-8)$$

$$K_1 = \frac{1}{2} \int \frac{\rho dS d\chi}{S} \cdot \frac{2\psi T}{q B_p^2} |X|^2 , \quad (3-9)$$

$$K_2 = \frac{1}{2} \int \frac{\rho dS d\chi}{S} \frac{2T R^4}{q \psi_s} |Y|^2 , \quad (3-10)$$

$$K_3 = \frac{1}{2} \int \frac{\rho dS d\chi}{S} \frac{q R^6 B_p^2}{2 \psi_s T} |V + \frac{2}{q} Y - \frac{\vec{\nabla}\psi \cdot \vec{\nabla}\chi}{|\nabla\psi|^2} \psi \frac{2}{S} X|^2 , \quad (3-11)$$

記号は参考文献 7) に従う。W₁, W₂, W₃ は、それぞれ、径方向、トロイダル方向及びポロイダル方向の磁場運動に伴うエネルギーを、又、W₄ は、断熱圧縮に伴なう内部エネルギー、W₅ は、プラズマ電流及び圧力勾配による不安定エネルギーを示している。また、K₁, K₂, K₃ の物理的意味は、それぞれ、径方向、トロイダル方向及びポロイダル方向の運動エネルギーである。

なお、W₁, W₂, W₃, W₄, W₅, W は、それぞれ、プログラム中では記号 RW1, RW2, RW3, RW4, RW5 及び RWVAL で表示されており、グラフの上では△, +, ×, ◇, ▲, および ○ で表示されている。また、K₁, K₂, K₃, K は、それぞれ、プログラム中では記号 RK1, RK2, RK3, 及び RKVAL で表示されており、グラフ上では△, +, × および ○ で表示されている。

(3) プラズマ変位のポロイダル座標依存性

位置エネルギーおよび運動エネルギーの局所的な寄与が最大となる磁気面上で変位成分 ψ の χ 依存性を示した図で、位置エネルギー最大面に関する変位を記号 XWVAL △ で、運動エネルギー最大面に関する変位を記号 XKVAL ○ で示してある。

このプログラムのソース・リストは第 3-10 図に示す。

3.3 ロード・モジュールの作成

一般にフォートランで書かれた数値計算コードではメッシュ等の最大値が固定されており、この制限を超えて計算を実行するときはプログラムの書き換えが必要となる。この作業は単調であり、また誤りを起こしやすい。この作業を自動的に行うために TRITON¹⁰⁾ システムではメッシュ・サイズの配列等を書き換えるプログラム EOS を備えている。ERATO-J (F 版) も EOS プログラムを使用して、Nψ, Nχ の大きさに対応する ERATO ロード・モジュールをソース・プログラムの書き換えなしに作成できるように整備されていることは第 3.1 節で述べたとおりである。

FACOM OSIV/F4 GEM V02L08E DATE 80.07.25 TIME 16.24.27 LIB=J9079.EARG.FORT

NO.3	MODULE NAME	EIGVEC	BLOCKS	9
	LEVEL	14	DATE	80.04.11 TIME 17.10.43

```

C          00000100
C          * ARGUS PROGRAM FOR ERATO *
C          00000200
C          00000300
C          00000400
C          REAL*8 DNTORE,DRKMAX,DRWMAX,DGXMAX,DGXMAX
C          00000500
C          REAL*8 DVCS (101),DVQ (101),DFXR (51,101)
C          00000600
C          REAL*8 DVCHI (101),DRKVAL(101),DRWVAL(101)
C          00000700
C          REAL*8 DXKVAL(101),DXWVAL(101)
C          00000710
C          REAL*8 DRK1(101),DRK2(101),DRK3(101)
C          00000720
C          REAL*8 DRW1(101),DRW2(101),DRW3(101),DRW4(101),DRW5(101)
C          00000730
C          REAL*8 WNTORE,REXT,AL(3)
C          00000740
C          LOGICAL*1 DSN(80)
C          00000750
C          INTEGER*4 IDATE(2)
C          00000800
C          DIMENSION VCS (101),VQ (101),FXR (51,101)
C          00000900
C          DIMENSION VCHI (101),RKVAL(101),RWVAL(101)
C          00001000
C          DIMENSION XKVAL(101),XWVAL(101)
C          00001100
C          DIMENSION RK1(101),RK2(101),RK3(101)
C          00001110
C          DIMENSION RW1(101),RW2(101),RW3(101),RW4(101),RW5(101)
C          00001120
C          DIMENSION X (101),Y(101),ALP(101)
C          00001200
C          DIMENSION ICM(10)
C          00001300
C          DIMENSION INT(50)
C          00001400
C          DIMENSION Z(101)
C          00001500
C          DIMENSION IL(50)
C          00001600
C          DIMENSION IDT(50)
C          00001700
C          DIMENSION INO(2 )
C          00001710
C          00001800
C          ****
C          READ DATA FROM FILE #22
C          00001810
C          ****
C          00001820
C          ****
C          00001830
C          ****
C          00001840
C          ****
C          IXX=22
C          00001900
C          ****
C          00002000
C          ****
C          00002100
C          ****
C          00002200
C          READ (IXX) NPSI,NCHI,DNTORE,MMAX,
C          (DVCS(I),I=1,NPSI),(DVQ(I),I=1,NPSI),
C          00002300
C          ((DFXR(J,I),I=1,NPSI),J=1,MMAX)
C          00002400
C          ****
C          READ (IXX) DRKMAX,DRWMAX,DGXMAX,DGXMAX,
C          00002500
C          (DVCHI (I),I=1,NCHI),
C          00002600
C          (DRKVAL(I),I=1,NPSI),
C          00002700
C          (DRWVAL(I),I=1,NPSI),
C          00002800
C          (DRK1(I),I=1,NPSI),(DRK2(I),I=1,NPSI),
C          00002810
C          (DRK3(I),I=1,NPSI),(DRW1(I),I=1,NPSI),
C          00002820
C          (DRW2(I),I=1,NPSI),(DRW3(I),I=1,NPSI),
C          00002830
C          (DRW4(I),I=1,NPSI),(DRW5(I),I=1,NPSI),
C          00002840
C          (DXKVAL(I),I=1,NCHI),
C          00002900
C          (DXWVAL(I),I=1,NCHI)
C          ****
C          READ(50) NO,NCHEK,ITIME,IDATE,DSN,NPSI,NCHI,WNTORE,REXT,
C          00003001
C          AL,NEG,NCONV,NCOMP,NIT,NITMAX,N
C          00003002
C          ****
C          00003010
C          ****
C          FROM REAL*8 TO REAL*4
C          00003100
C          ****
C          00003200
C          ****
C          00003300

```

(続く)

Fig. 3-10 Source program list of ERATO6.

```

C 00003310
C * CHECK DATA * 00003320
  ITEST=0 00003330
C 00003340
  ANTORE=ONTORE 00003400
  INTORE=ANTORE 00003500
  CALL IBTOD(IN,INTORE) 00003600
  CALL IBTOD(IP,NPSI ) 00003700
  CALL IBTOD(IC,NCHI ) 00003800
  CALL IBTOD(IM,MMAX ) 00003900
  CALL IBTOD(INO(2),NO) 00003901
  RKMAX =DRKMAX 00004000
  RWMAX =DRWMAX 00004100
  XKMAX =DXKMAX 00004200
  XWMAX =DXWMAX 00004300
C 00004400
  DO 1000 I=1,NPSI 00004500
  VCS(I)=DVCS(I) 00004600
  VQ (I)=DVQ (I) 00004700
1000  CONTINUE 00004800
  DO 1100 J=1,MMAX 00004900
  DO 1050 I=1,NPSI 00005000
  FXR(J,I)=DFXR(J,I) 00005100
1050  CONTINUE 00005200
1100  CONTINUE 00005300
  DO 1200 I=1,NPSI 00005400
  RKVAL(I)=DRKVAL(I) 00005500
  RWVAL(I)=DRWVAL(I) 00005600
  RK1(I)=DRK1(I) 00005610
  RK2(I)=DRK2(I) 00005620
  RK3(I)=DRK3(I) 00005630
  RW1(I)=DRW1(I) 00005640
  RW2(I)=DRW2(I) 00005650
  RW3(I)=DRW3(I) 00005660
  RW4(I)=DRW4(I) 00005670
  RW5(I)=DRW5(I) 00005680
1200  CONTINUE 00005700
  DO 1300 I=1,NCHI 00005800
  VCHI (I)=DVCHI(I) 00005900
  XKVAL(I)=DXKVAL(I) 00006000
  XWVAL(I)=DXWVAL(I) 00006100
1300  CONTINUE 00006200
C 00006300
C ***** NORMALIZATION OF FOURIER COMPONENTS ***** 00006400
C ***** ***** ***** ***** ***** ***** ***** ***** 00006500
C 00006510
  FXRMAX=0.0 00006600
  DO 2100 J=1,MMAX 00006700
  DO 2000 I=1,NPSI 00006800
  IF(FXRMAX.GE.ABS(FXR(J,I))) GO TO 2000 00006900
  FXRMAX=ABS(FXR(J,I)) 00007000
  FXRM=FXR(J,I) 00007100
2000  CONTINUE 00007200
2100  CONTINUE 00007300
  DO 2300 I=1,NPSI 00007400
  DO 2200 J=1,MMAX 00007500

```

(続く)

```

      FXR(J,I)=FXR(J,I)/FXRM          00007600
2200  CONTINUE                      00007700
2300  CONTINUE                      00007800
C                                         00007810
C   ****
C   RESONANCE SURFACES               00007820
C   ****
C
C   IA=IABS(INTORE)                 00007830
C   NSRF=3.5*IA-IA+1                00007840
C                                         00007850
C
C   *****                         00007860
C   *****                         00007870
C                                         00007900
C   *****                         00007910
C   *****                         00007920
C   ** ARGUS PROCESSING **          00008000
C   *****                         00008100
C   *****                         00008110
C                                         00008120
C
C   CALL ARGUS(0)                   00008200
C
C   *****                         00008300
C   Q-PROFILE : GRAPHIC DATA SET   00008400
C   *****                         00009400
C                                         00009500
C                                         00009510
C
NCRV=1+NSRF                         00009600
DO 2306 I=1,NPSI                     00009700
X(I+2)=VCS(I)                       00009800
Y(I+2)= VQ(I)                       00009900
2306  CONTINUE                      00010000
X(1)=0.0005                          00010100
X(2)=0.0005                          00010200
Y(1)=VQ(1)                           00010300
Y(2)=VQ(1)                           00010400
X(NPSI+3)=0.9995                    00010500
X(NPSI+4)=0.9995                    00010600
Y(NPSI+3)=(Y(NPSI+2)-Y(NPSI+1))*(X(NPSI+3)-X(NPSI+2))/ 00010700
*                               (X(NPSI+2)-X(NPSI+1))+Y(NPSI+2) 00010800
Y(NPSI+4)=3.0                         00010900
C                                         00010910
C   CALL STOR1D('Q-PROFIL',NPSI+4,-NCRV,X,ALP,Y,NPSI+4) 00011000
C   * CHECK DATA *
IF(ITEST.EQ.0) GO TO 2309          00011010
WRITE(6,2308) NPSI,NCHI,MMAX,IA     00011020
WRITE(6,* ) X                         00011030
WRITE(6,* ) Y                         00011040
2308 FORMAT(1H0,2X,4I4,2X/1H , ' Q-PROFILE')
2309  CONTINUE                      00011060
C                                         00011070
C                                         00011080
IA3=3.5*IA                          00011100
DO 2400 J=IA,IA3                     00011200
DO 2310 I=2,NPSI+3                  00011300
AAJ=J                                00011400
AAI=IA                               00011500
Z(I)=AAJ/AAI                          00011600
2310  CONTINUE                      00011700
Z(1)=Y(1)                            00011800
Z(NPSI+4)=Y(NPSI+3)                  00011900
C                                         00011910

```

(続く)

```

CALL STOR1D('Q-PROFIL',NPSI+4,-NCRV,X,ALP,Z,NPSI+4)          00012000
C
C * CHECK DATA *
IF(ITEST.EQ.0) GO TO 2400                                     00012010
WRITE(6,2320)
WRITE(6,*) Z
2320 FORMAT(1H0,2X,'Q-PROFILE(Z)')
C
2400 CONTINUE
C
C *****
C FOURIER-EIGEN-VECTOR : GRAPHIC DATA SET                  00012020
C *****
C
NX=NPSI+1                                         00012030
X(1)=0.0005                                       00012040
Y(1)=0.0005                                       00012050
DO 2600 J=1,MMAX                                     00012060
DO 2500 I=1,NPSI                                    00012070
X(I+1)=VCS(I)                                      00012100
Y(I+1)=FXR(J,I)                                    00012200
2500 CONTINUE                                         00012300
CALL STOR1D('FEIGVEC ',NX,-MMAX,X,ALP,Y,NX)           00012400
2600 CONTINUE                                         00012500
C
C *****
C ENERGY VS. S : GRAPHIC DATA SET                      00012600
C *****
C
CALL STOR1D('K-ENERGY',NPSI,-4,VCS,ALP,RKVAL,NPSI)    00012700
CALL STOR1D('K-ENERGY',NPSI,-4,VCS,ALP,RK1 ,NPSI)    00012800
CALL STOR1D('K-ENERGY',NPSI,-4,VCS,ALP,RK2 ,NPSI)    00012900
CALL STOR1D('K-ENERGY',NPSI,-4,VCS,ALP,RK3 ,NPSI)    00013000
CALL STOR1D('P-ENERGY',NPSI,-6,VCS,ALP,RWVAL,NPSI)   00013100
CALL STOR1D('P-ENERGY',NPSI,-6,VCS,ALP,RW1 ,NPSI)   00013200
CALL STOR1D('P-ENERGY',NPSI,-6,VCS,ALP,RW2 ,NPSI)   00013300
CALL STOR1D('P-ENERGY',NPSI,-6,VCS,ALP,RW3 ,NPSI)   00013400
CALL STOR1D('P-ENERGY',NPSI,-6,VCS,ALP,RW4 ,NPSI)   00013500
CALL STOR1D('P-ENERGY',NPSI,-6,VCS,ALP,RW5 ,NPSI)   00013600
C
C *****
C * CHECK DATA *
IF(ITEST.EQ.0) GO TO 2700                                     00019600
WRITE(6,2610)
WRITE(6,*) VCS
WRITE(6,*) RKVAL
WRITE(6,*) RK1
WRITE(6,*) RK2
WRITE(6,*) RK3
2610 FORMAT(1H0,2X,'K-ENERGY(VCS,RKVAL,RK1,RK2,RK3)')
WRITE(6,2630)
WRITE(6,*) RWVAL
WRITE(6,*) RW1
WRITE(6,*) RW2
WRITE(6,*) RW3
WRITE(6,*) RW4

```

(続く)

```

      WRITE(6,*) RW5          00019936
2630  FORMAT(1H0,2X,'P-ENERGY(RWVAL,RW1,RW2,RW3,RW4)') 00019937
2700  CONTINUE            00019938
C
C *****DISPLACEMENT : GRAPHIC DATA SET*****           00020100
C
C *****CONTINUATION*****                                00020200
C
C
C     CALL STOR1D('DISPLACE',NCHI,-2,VCHI,ALP,XKVAL,NCHI) 00020200
C     CALL STOR1D('DISPLACE',NCHI,-2,VCHI,ALP,XWVAL,NCHI) 00020310
C
C * CHECK DATA *
C     IF(ITEST.EQ.0) GO TO 2800                         00020400
C     WRITE(6,2750)
C     WRITE(6,*) VCHI                                     00020500
C     WRITE(6,*) XKVAL                                    00020510
C     WRITE(6,*) XWVAL                                    00020520
2750  FORMAT(1H0,2X,'DISPLACEMENT(XKVAL,XWVAL)')        00020530
2800  CONTINUE            00020540
C
C *****END OF GDS PROCESSING***                         00020550
C *****CONTINUATION*****                                00020560
C
C
C *** CONTROL DATA SET ***
C
C *****CONTINUATION*****                                00020570
C
C
C     DO 2910 I=1,50                                     00020580
C     INT(I)=3                                         00020590
2910  CONTINUE            00020600
C     CALL APPROX('FEIGVEC ',INT)                      00020610
C     CALL APPROX('DISPLACE',INT)                      00020700
C     IDT(1)=1                                         00020800
C     IDT(2)=2                                         00020810
C     IDT(3)=3                                         00020820
C     IDT(4)=4                                         00020900
C     IDT(5)=5                                         00021000
C     IDT(6)=6                                         00021010
C     CALL DATAPT('K-ENERGY',IDT)                     00021100
C     CALL DATAPT('P-ENERGY',IDT)                     00021200
C     CALL DATAPT('DISPLACE',IDT)                     00021300
C     DO 2920 I=1,50                                     00021310
C     IDT(I)=0                                         00021320
2920  CONTINUE            00021330
C     CALL DATAPT('Q-PROFIL',IDT)                     00021340
C     CALL DATAPT('FEIGVEC ',IDT)                     00021350
C     CALL FRAME('Q-PROFIL',1,1,1)                   00021360
C     CALL GRID ('Q-PROFIL',1,4,0)                   00021370
C     CALL FRAME('FEIGVEC ',1,1,1)                   00021380
C     CALL FRAME('K-ENERGY',1,1,1)                   00021390
C     CALL FRAME('P-ENERGY',1,1,1)                   00021400
C     CALL FRAME('DISPLACE',1,1,1)                   00021410
C     DO 2920 I=1,50                                     00021420
C     IDT(I)=0                                         00021430
C     CALL DATAPT('Q-PROFIL',IDT)                     00021530
C     CALL DATAPT('FEIGVEC ',IDT)                     00021630
C     CALL FRAME('Q-PROFIL',1,1,1)                   00021730
C     CALL GRID ('Q-PROFIL',1,4,0)                   00022500
C     CALL FRAME('FEIGVEC ',1,1,1)                   00022600
C     CALL FRAME('K-ENERGY',1,1,1)                   00022700
C     CALL FRAME('P-ENERGY',1,1,1)                   00022800
C     CALL FRAME('DISPLACE',1,1,1)                   00022810
C     CALL GRID ('FEIGVEC ',1,4,6)                   00022900
C     CALL GRID ('K-ENERGY',1,4,4)                   00023000
C     CALL GRID ('P-ENERGY',1,4,4)                   00023100
C     CALL GRID ('DISPLACE',1,4,4)                   00023110
C
C
C

```

(続く)

```

CALL SCALE1('Q-PROFIL',1,-1,200.0,150.0,0.0,0.0,0.0,0.0) 00023300
CALL SCALE1('FEIGVEC ',1,-1,200.0,150.0,0.0,0.0,0.0,0.0) 00023400
CALL SCALE1('K-ENERGY',1,-1,200.0,150.0,0.0,0.0,0.0,0.0) 00023500
CALL SCALE1('P-ENERGY',1,-1,200.0,150.0,0.0,0.0,0.0,0.0) 00023600
CALL SCALE1('DISPLACE',1,-1,200.0,150.0,0.0,0.0,0.0,0.0) 00023700
IL(1)=1 00024400
DO 3100 I=2,50 00024500
IL(I)=4 00024600
3100 CONTINUE 00024700
CALL CURVE('Q-PROFIL',IL) 00024800
CALL CURVE('K-ENERGY',IL) 00024810
CALL CURVE('P-ENERGY',IL) 00024820
C 00024900
CALL COMENT('Q-PROFIL','X ',1,'(PSI/PSI-S)**0.5', 00025700
* 16,0.0,0.0,0.0,0.0) 00025800
CALL COMENT('FEIGVEC ','X ',1,'(PSI/PSI-S)**0.5', 00025900
* 16,0.0,0.0,0.0,0.0) 00026000
CALL COMENT('K-ENERGY','X ',1,'(PSI/PSI-S)**0.5', 00026700
* 16,0.0,0.0,0.0,0.0) 00026800
CALL COMENT('P-ENERGY','X ',1,'(PSI/PSI-S)**0.5', 00026810
* 16,0.0,0.0,0.0,0.0) 00026820
CALL COMENT('DISPLACE','X ',1,'CHI',3,0.0,0.0,0.0,0.0) 00026900
C 00027000
CALL COMENT('Q-PROFIL','Y1 ',2,'SAFETY FACTOR',13,0.0,0.0,0.0, 00027100
* 0.0) 00027200
CALL COMENT('FEIGVEC ','Y1 ',3,'FOURIER-EIG-VECTOR',18, 00027300
* 0.0,0.0,0.0,0.0) 00027400
CALL COMENT('K-ENERGY','Y1 ',2,'K(S)',4,0.0,0.0,0.0,0.0) 00028100
CALL COMENT('P-ENERGY','Y1 ',2,'W(S)',4,0.0,0.0,0.0,0.0) 00028200
CALL COMENT('DISPLACE','Y1 ',2,'X ON MAX K AND W SURF',21, 00028300
* 0.0,0.0,0.0,0.0) 00028400
ICM(1)='N' 00028700
ICM(2)=IN 00028800
CALL COMENT('FEIGVEC ',,'4,ICM,8,50.0,170.0,0.0,3.0) 00028900
ICM(1)='NPSI' 00029000
ICM(2)=IP 00029100
CALL COMENT('FEIGVEC ',,'5,ICM,8,50.0,165.0,0.0,3.0) 00029200
ICM(1)='NCHI' 00029300
ICM(2)=IC 00029400
CALL COMENT('FEIGVEC ',,'6,ICM,8,50.0,160.0,0.0,3.0) 00029500
ICM(1)='MMAX' 00029600
ICM(2)=IM 00029700
CALL COMENT('FEIGVEC ',,'7,ICM,8,50.0,155.0,0.0,3.0) 00029800
INO(1)='NO =' 00029810
CALL COMENT('Q-PROFIL',,'3,INO,8,-30.0,170.0,0.0,7.0) 00029820
CALL COMENT('FEIGVEC ',,'8,INO,8,-30.0,170.0,0.0,7.0) 00029830
CALL COMENT('K-ENERGY',,'3,INO,8,-30.0,170.0,0.0,7.0) 00029840
CALL COMENT('P-ENERGY',,'3,INO,8,-30.0,170.0,0.0,7.0) 00029850
CALL COMENT('DISPLACE',,'3,INO,8,-30.0,170.0,0.0,7.0) 00029860
C 00029900
C **** END OF CONTROL DATA SET *** 00030000
C **** END OF CONTROL DATA SET *** 00030100
C **** END OF CONTROL DATA SET *** 00030200
C **** END OF CONTROL DATA SET *** 00030210
C **** END OF CONTROL DATA SET *** 00030220
C *** DISPLAY SUBROUTINES *** 00030300
C **** END OF CONTROL DATA SET *** 00030400

```

(続く)

(続き)

C	CALL ONEDIM('Q-PROFIL',0)	00030410
	CALL ONEDIM('FEIGVEC ',0)	00030500
-----	CALL ONEDIM('K-ENERGY',0)	00030600
	CALL ONEDIM('P-ENERGY',0)	00030900
	CALL ONEDIM('DISPLACE',0)	00030910
C	CALL ARGUS(999)	00031000
	STOP	00031100
	END	00031200
		00031300
		00031400

EOS を使う場合、配列の大きさ等を変数としてプログラムを書き、このプログラムを入力データとしてプリプロセッサー EOS を実行することによってこの変数を実際の数値に置き換えたフォートラン・プログラムが生成される。このときの EOS プリプロセッサーの入力となるプログラムを“EOS ソース・プログラム”と呼ぶ。一般には、EOS ソース・プログラムの最初には配列の大きさを実際に決める文 (SIZE 文) とそれに続く COMMON 部分が存在するのが普通であるが、ERATO-J(F 版)の場合には、SIZE 文と COMMON 文を含む COMMON ファイル一式と、各ソース・プログラム (ERATO 1, 2, 3, 4, 5, 6 および MTOUT, MTIN) の 9 ファイルで完全な一組のプログラムを構成している (第 3-1 表参照)。これは、ERATO-J (F 版) では、ERATO 1～ERATO 6 のそれぞれのプログラムは、それぞれ一つのジョブ・ステップに対応していて、ロード・モジュールは別々に生成されるという事情があるにもかかわらず、各プログラム内の COMMON 文は共通しているからである。

第 3-11 図には、EOS プリプロセッサーを使用して、ロード・モジュールを作るまでの作業手順を、また、第 3-12 図には、この作業を行う為の JCL プログラムを示した。この JCL プログラムは、GEM ファイルの J 3331.JCL.CNTL (モジュール名 : PCL) に格納されている (第 3-1 表参照)。第 3-11 図に示してあるのは、COMMON ファイル

J 3331. ERT0F FORT (A # CM XX EL) と

EOS ソース・プログラム J 3331. ERT1 FS . FORT から、

ロード・モジュール J 3331. E 1 D 50 FS 5. LOAD が作成される例である。言うまでもなく、実際にはこれらファイル名は、場合に応じて必要なものに変えられねばならない。この JCL プログラムを用いて、ロード・モジュールを作成する際注意しなければならないことは、ロード・モジュール・ファイルは、短期保存ファイル領域 (1ヶ月保存) に作成されるようになっている事である。これは、ディスク資源が十分でないような条件下でもファイル確保が常に可能であるようにする為である。長期にわたって、ロード・モジュール・ファイルを保存する場合は、このようにしてできたロード・モジュール・ファイルを長期保存ファイルにコピーしなければならない。

以下に、ロード・モジュール作成までの手順を整理しておく。

- (1) COMMON ファイル (J 3331. ERT0F FORT) のモジュール (A # CM XX EL)において N P (N_P) , NC (N_C) , M の数値を書き替える。ここで、M は DEFINE FILE 文のレコード数である (第 2-2 表参照)。
- (2) JCL プログラム (J 3331.JCL.CNTL 中のモジュール PCL) において、EOS ソース・プログラム名、ロード・モジュール名等を書き替えて SUBMIT する。
- (3) 必要に応じて長期保存ファイルを確保して、ロード・モジュールをコピーする。この作業の具体的手順は、次のとおりである。

```
ALLOC F (F1) DA (XXX. LOAD) SPACE (20 5) T
      NEW DIR (1) UNIT (USER) CA
```

```
COPY △△△・LOAD XXX. LOAD
```

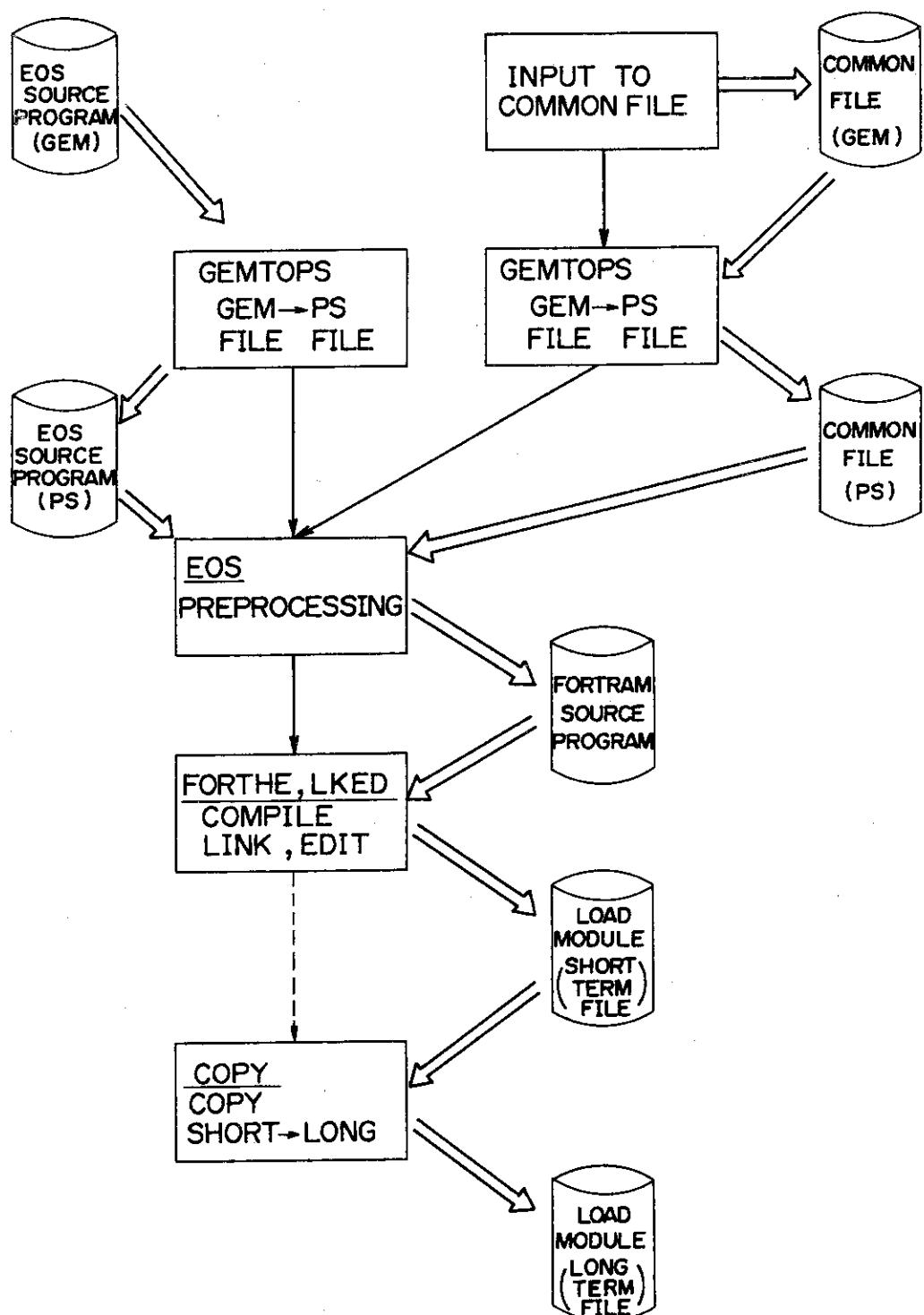


Fig. 3-11 Procedure to generate a load module of the ERATO code.

FACOM-OSIV/F4-GEM--VC2LCSE-DATE 80.07.23--TIME 14.43.57 --LIB=J9079.JCL.CNTL-----

NO.8	MODULE NAME PCL LEVEL	BLOCKS 6 41	DATE 80.04.16	TIME 14.22.07	7-R-----8
------	--------------------------	-------------------	------------------	------------------	-----------

```

//JCLG-JOB                                     00000100
// EXEC JCLG                                    00000200
//SYSIN DD DATA,DLH='+++'                      00000300
// JUSER 57309079,YU.TANAKA,0341.120,ERATO      00000400
--T.3-W.1-P.0-C.1-I.2                         00000500
  OOPTP MSGCLASS=I                           00000600
// EXEC GEMTOPSM,DSN='J3221.ERTOF',MOD=A&CMXXEL,SYSOUT=I 00000700
//PSLIB DD DSN=&TEMP1,UNIT=WK10,SPACE=(TRK,(30,10)),DISP=(NEW,PASS) 00000800
// EXEC GEMTOPS,DSN='J3221.ERTIFS',SYSOUT=I 00000900
//PSLIB DD DSN=&TEMP2,UNIT=WK10,SPACE=(TRK,(30,10)),DISP=(NEW,PASS) 00001000
// EXEC LMGD,LM='J3227.EGSLM',SYSOUT=I 00001100
//FT09F001 DD DSN=&TEMP1,UNIT=WK10,DISP=(OLD,PASS) 00001200
//          DD DSN=&TEMP2,UNIT=WK10,DISP=(OLD,PASS) 00001300
// EXPAND TPDISK,DDN=FT10F001,DSN=TFILE,RSIZE=80,BSIZE=3200 00001400
//FT11F001 DD DUMMY                            00001500
//SYSIN DD *                                     00001600
  &OPTP FORT=1 3END                          00001700
  &IODVC SIN=9 3END                          00001800
/*
// EXEC FORTHE,A=NOSOURCE,SYSOUT=I            00002000
// EXPAND TPDISK,DDN=SYSIN,DSN=TFILE,RSIZE=80,BSIZE=3200,DISP=OLD 00002100
// EXEC LKEDCT,SYSOUT=I,A='MAP'               00002200
//LINK.SYSLIB DD DSN=SYS1.FORTLIB,DISP=SHR   00002300
//          DD DSN=SYS9.COM.LOAD,DISP=SHR    00002400
//          DD DSN=J3227.OLYMPUSL.LOAD,DISP=SHR 00002500
//          DD DSN=J3231.GTPARM.LOAD,DISP=SHR  00002600
//LINK.SYSLMD DD DSN='J3231.E1D50FSH.LGAD,UNIT=TDS, 00002610
//          DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(10,10,10)) 00002620
+++
//                                     00002700
//                                     00002800

```

Fig.3-12 JCL program for the procedure in Fig.3-11.

こゝで，“XXX. LOAD”は長期保存ファイル上のロード・モジュール名，“△△△. LOAD”は短期保存ファイル上のロード・モジュール名である。

ところで、既に述べたように、一連のERATO計算では、計算メッシュ数の異なる多数のロード・モジュールを使って計算を進めるので、混乱を避ける為、ロード・モジュールの命名法については十分注意する必要がある。現在、利用している命名法は次のとおりである（第3-1表参照）。

E	1	D	5	0	F	S	5	
①	②	③	④	⑤				(3-12)

- ① ERATO 1 のロード・モジュールである事を示す。
- ② プログラムが倍精度である事を示す。
- ③ $N_\psi = N_\chi = 50$ が最大メッシュである事を示す。
- ④ F版である事を示す。
- ⑤ 平衡プログラムの出力が単精度で 512×512 メッシュである事を示す（ERATO 1 に
関係）。

DIARY関係のプログラムについても同様の手順でロード・モジュールの作成を行えばよい。

3.4 ディスクと磁気テープの初期化

DIARYと中間結果保存用磁気テープは、一連のERATO計算に先立って初期化せねばならない。

3.4.1 ディスクの初期化

まず最初にディスク上の領域確保が必要である。これをTSSで行うには次のようにすればよい。

```
ALLOC F (FT0 1F 001) DA (XXX·DATA) SPACE (12 1)
      T NEW UNIT (USER)
```

```
ALLOC F (F) DA (XXX0·DATA) SPACE (12 1)
      T NEW UNIT (USER)
```

ここで、XXX·DATAは、DIARY FILEの名前で、XXX0·DATAはこのファイルのバックアップ用ファイル名である。

このようにして確保されたファイルを初期化するにはINITIALプログラム（ソース：J3331.INITIAL.FORT...GEMファイル、ロード・モジュール：J3331.INITIAL.LOAD）を実行すればよい。即ち、

```
CALL 'J3331.INITIAL.LOAD'
```

領域確保と INITIAL プログラム実行を、同一のセッションで行わない場合は、XXX. DATA を定義名 FT01F001 に結合しなければならない。

データが既に蓄積されている DIARY FILE を対象に INITIAL プログラムを実行すると蓄積データを全て消去して初期化してしまうので注意しなければならない。

3.4.2 磁気テープの初期化

大量のデータを保存しておく必要上、磁気テープの記録密度は、6250 BPI を標準とする。従って、磁気テープの初期化には、この密度を指定せねばならない。

MTOUT プログラムで、データを磁気テープに書き出す時の JCL プログラムは第 3-13 図(a)に示すように磁気テープの DD 文において DISP = (**[MOD]**, KEEP) としてデータを順次蓄積するようになっている。従って新しい磁気テープに書き込みはじめる場合には、この JCL プログラムをそのまま使えばよいが、一度使用された磁気テープを再利用する際には、最初の一回だけは、DISP = (**[NEW]**, KEEP) と指定する必要がある。

3.5 ERATO 計算の実行

第 3-1 図で経路 $B_2, B_3 \rightarrow E_2'$ に沿ってメッシュ数 $N_\psi = N_\chi = 50$ までの計算を実行する為の JCL プログラムを第 3-13(a) 図に示す。この JCL プログラムは J3331.EJCL50.M.CNTL に格納されている。ジョブの実行に際しては、この JCL プログラムを使って端末から SUBMIT すればよい。

前半はストリーム・プロシジャー "ERATO" の定義部分であって実際の計算を行う場合ロード・モジュールをえない限り変更する必要はない。この部分で **[]** で囲んだ SPACE 指定は、行列 A, B, DL^T を格納するファイルの大きさを示すものでメッシュ数の最大値に応じて第 2-5, 2-12 表から求めて書き替えねばならない。

後半は、パラメータ指定を行って、ストリーム・プロシジャーを展開実行する部分である。ここで、データ・セット名 DSN0 は、平衡計算の結果の入っているファイル名、また、DSN1～DSN4 は ERATO 1～ERATO 4 のロード・モジュール・データ・セット名、DIARY は、DIARY FILE 名である。ロード・モジュール・データ・セット名および、DIARY FILE 名は標準値 (E1D50FS5, E2D50F, E3D50F, E4D50F および DIARYIN0) が与えられているが、これらはもちろん展開時に変更可能である。ネームリスト & NEWRUN は、ERATO の実行の為の入力データであって、その内容は、原 ERATO のマニュアル⁷⁾ に説明してある。通常の計算実行に必要なパラメータについてのみ第 3-3 表にまとめておく。

なお、第 3-12(a) 図に示した例では、ERATO の計算を 1 度だけ行うようになっているが、この数はシステム資源の許す範囲で任意である。

また、原研計算センターで巨大ジョブの実行等を行う必要上、ジョブ資源の指定文、JUSER 文をカード入力する場合には、J3331.EJCL50M.CNTL の主要部を JCL プログラム J3331.EJCL50L.CNTL を使って第 3-14 図の JCL プログラムをカードに穿孔してジョブの投入を行えばよい。J3331.EJCL50L.CNTL の内容は、J3331.EJCL50M.CNTL

```

//JCLG JOB          00000010
// EXEC JCLG        00000020
//SYSIN DD DATA,DLIM='+++' 00000030
// JUSER 54313331,SU.TOKUDA,0021.120,ESGUIDPM 00000040
  T.2 W.1 C.5 P.0 I.5 00000050
  OPTP MSGCLASS=I,MSGLEVEL=(1,1) 00000060
//***** 00000070
//* THIS JCL FOR MT OUTPUT * 00000080
//* ::::::::::: * 00000090
//*- DIARY FILE === J3331.DIARYING.DATA * 00000100
//* * 00000110
//***** 00000120
//ERATO PROC DSN0=,DSN1=E1050FS5,DSN2=E2D50F,DSN3=E3D50F,DSN4=E4D50F, 00000130
//  DIARY=DIARYG,S=# 00000140
//ERATO1 EXEC PGM=TEMPNAME,COND=(4,LT) 00000150
//STEPLIB  DD DSN=J3331.&DSN1..LOAD,DISP=SHR 00000160
//FTG4F001 DD DSN=&&FILE04,UNIT=WK10,DISP=(NEW,PASS), 00000170
//  SPACE=(TRK,(10,10)) 00000180
//FTG6F001 DD SYSOUT=I 00000190
//FTG8F001 DD DSN=&&FILE08,UNIT=WK10,DISP=(NEW,PASS), 00000200
//  SPACE=(TRK,(10,10)) 00000210
//FTG11F001 DD DSN=&DSN0..DATA,DISP=SHR 00000220
//FTG16F001 DD DSN=&&FILE16,UNIT=WK10,DISP=(NEW,PASS), 00000230
//  SPACE=(TRK,(10,10)) 00000240
//FTG17F001 DD DSN=&&FILE17,UNIT=WK10,DISP=(NEW,PASS), 00000250
//  SPACE=(TRK,(10,10)) 00000260
//SYSPRINT DD SYSOUT=I 00000270
//ERATO2 EXEC PGM=TEMPNAME,COND=(4,LT) 00000280
//STEPLIB  DD DSN=J3331.&DSN2..LOAD,DISP=SHR 00000290
//FTG6F001 DD SYSOUT=I 00000300
//FTG8F001 DD DSN=&&FILE08,DISP=(OLD,PASS) 00000310
//FTG17F001 DD DSN=&&FILE17,DISP=(OLD,PASS) 00000320
//SYSPRINT DD SYSOUT=I 00000330
//ERATO3 EXEC PGM=TEMPNAME,COND=(4,LT) 00000340
//STEPLIB  DD DSN=J3331.&DSN3..LOAD,DISP=SHR 00000350
//FTG4F001 DD DSN=&&FILE04,DISP=(OLD,PASS) 00000360
//FTG6F001 DD SYSOUT=I 00000370
//FTG7F001 DD DSN=&&FILE07,UNIT=WK10,DISP=(NEW,PASS), 00000380
//  SPACE=(CYL,(60,10),,CONTIG), 00000390
//  DCB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000400
//FTG8F001 DD DSN=&&FILE08,DISP=(OLD,PASS) 00000410
//FTG9F001 DD DSN=&&FILE09,DISP=(NEW,PASS),UNIT=WK10, 00000420
//  SPACE=(CYL,(60,10),,CONTIG), 00000430
//  DCB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000440
//FTG17F001 DD DSN=&&FILE17,DISP=(OLD,DELETE) 00000450
//SYSPRINT DD SYSOUT=I 00000460
//ERATO4 EXEC PGM=TEMPNAME,COND=(4,LT), 00000470
//  PARM='&DSNC&S&DSN1&S&DSN2&S&DSN3&S&DSN4' 00000480
//STEPLIB  DD DSN=J3331.&DSN4..LOAD,DISP=SHR 00000490
//FTG1F001 DD DSN=J3331.&DIARY..DATA,DISP=OLD 00000500
//FTG5F001 DD DSN=&&FILE05,UNIT=WK10,DISP=(NEW,PASS), 00000510
//  SPACE=(TRK,(10,10)) 00000520
//FTG6F001 DD SYSOUT=I 00000530
//FTG7F001 DD DSN=&&FILE07,DISP=(OLD,DELETE) 00000540
//FTG8F001 DD DSN=&&FILE08,DISP=(OLD,PASS) 00000550
//FTG9F001 DD DSN=&&FILE09,DISP=(OLD,DELETE) 00000560
//FTG10F001 DD DSN=&&FILE10,DISP=(NEW,PASS),UNIT=WK10, 00000570
//  SPACE=(TRK,(10,10)), 00000580
//  DCB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000590
//FT33F001 DD DSN=&&ERT33,DISP=(NEW,DELETE),UNIT=WK10, 00000600
//  SPACE=(CYL,(60),,CONTIG), 00000610
//  DCB=(RECFM=F,BLKSIZE=19063,DSORG=DA) 00000620

```

(続く)

Fig.3-13 JCL programs for the ERATO calculations.

(a) J3331.EJCL50M.CNTL for the path ($B_2, B_3 \rightarrow E'_2$) in Fig.3-1.

(続き)

```

//MTOUT EXEC PGM=TEMPNAME,COND=(4,LT)          C0000630
//STEPLIB DD DSN=J3331.MTOUTD70.LOAD,DISP=SHR   C0000640
//FT50F001 DD DSN=&&FILE50,DISP=(OLD,DELETE)    C0000650
//FT04F001 DD DSN=&&FILE04,DISP=(OLD,DELETE)    C0000660
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,DELETE)    C0000670
//FT10F001 DD DSN=&&FILE10,DISP=(OLD,DELETE)    C0000680
//FT16F001 DD DSN=&&FILE16,DISP=(OLD,DELETE)    C0000690
//SYSPRINT DD SYSOUT=I                         C0000700
//FT06F001 DD SYSOUT=I                         C0000710
//FT60F001 DD DSN=J2636.ERTDUMP0,VOL=SER=011096,UNIT=MT62, C0000720
//          LABEL=(1,SL),DISP=(MDC,KEEP),        C0000730
//          DCB=(DEN=4,DSORG=PS,RECFM=V8S,BLKSIZE=19000,LRECL=18996) C0000740
//COPY   EXEC PGM=JSDGGENER,COND=(4095,LT)       C0000750
//STEPLIB DD DSN=SYS1.LINKLIB,DISP=SHR         C0000760
//SYSUT1 DD DSN=J3331.&DIARY..DATA,DISP=OLD    C0000770
//SYSUT2 DD DSN=J3331.&DIARY.O.DATA,DISP=SHR,UNIT=(,,DEFER) C0000780
//SYSIN  DD GUMMY                           C0000790
//SYSPRINT DD SYSOUT=I                       C0000800
// PEND
//*
//*-----
//STEP1 EXEC ERATD,DSNC="J2870.IN005T40"      C0000840
//ERATD1.FT05F001 DD *
&NEWRUN NPSI=14,NCHI=14,WNTORE=-3.0,AL0=-0.8, C0000850
      REXT=0.0,NDIVSI=0,SCALE=1.0,NITMAX=10 &END C0000860
//*-----                                     C0000870
//                                         C0000880
//                                         C0000890

```

```

//JCLG JOB                                     00000010
// EXEC JCLG                                     00000020
//SYSIN DD DATA,DLM='++'                         00000030
// JUSER 54313331,SH.TOKUDA,CO21.120,ESGUIDPM   00000040
T.2 W.2 C.5 P.0 I.5                           00000050
OPTP MSGCLASS=1,MSGLEVEL=(1,1)                 00000060
//***** THIS JCL FOR WHDRL EXECUTION          00000070
//*      :::::                                *
//*      DIARY FILE === J3331.DIARY.G.DATA     00000080
//*      *                                *
//*****                                         00000090
//*****                                         00000100
//*****                                         00000110
//*****                                         00000120
//ERATO PROC DSN0=,DSN1=E1D50FS5,DSN2=E2D50F,DSN3=E3D50F,DSN4=E4D50F, 00000130
//          DSN5=E5D70F,DIARY=DIARYG,S=*          00000140
//ERATO1 EXEC PGM=TEMPNAME,COND=(4,LT)          00000150
//STEPLIB DD DSN=J3331.&DSN1..LOAD,DISP=SHR    00000160
//FTC4F001 DD LSN=&&FILE04,UNIT=WK10,DISP=(NEW,PASS), 00000170
//          SPACE=(TRK,(10,10))                  00000180
//FTC6F001 DD SYSOUT=I                         00000190
//FTC8F001 DD DSN=&&FILE06,UNIT=WK10,DISP=(NEW,PASS), 00000200
//          SPACE=(TRK,(10,10))                  00000210
//FT11F001 DD DSN=&&DSN0..DATA,DISP=SHR        00000220
//FT16F001 DD DSN=&&FILE16,UNIT=WK10,DISP=(NEW,PASS), 00000230
//          SPACE=(TRK,(10,10))                  00000240
//FT17F001 DD DSN=&&FILE17,UNIT=WK10,DISP=(NEW,PASS), 00000250
//          SPACE=(TRK,(10,10))                  00000260
//SYSPRINT DD SYSOUT=I                         00000270
//ERATO2 EXEC PGM=TEMPNAME,COND=(4,LT)          00000280
//STEPLIB DD DSN=J3331.&DSN2..LOAD,DISP=SHR    00000290
//FT06F001 DU SYSOUT=I                         00000300
//FT06F001 DD DSN=&&FILE06,DISP=(OLD,PASS)      00000310
//FT17F001 DU DSN=&&FILE17,DISP=(OLD,PASS)      00000320
//SYSPRINT DD SYSOUT=I                         00000320
//ERA103 EXEC PGM=TEMPNAME,COND=(4,LT)          00000340
//STEPLIB DD DSN=J3331.&DSN3..LOAD,DISP=SHR    00000350
//FT04F001 DD DSN=&&FILE04,DISP=(OLD,PASS)      00000360
//FT06F001 DU SYSOUT=I                         00000370
//FT07F001 DU DSN=&&FILE07,UNIT=WK10,DISP=(NEW,PASS), 00000380
//          SPACE=(CYL,(60,10),,CONTIG),           00000390
//          ECB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000400
//FT08F001 DU DSN=&&FILE08,DISP=(OLD,PASS)      00000410
//FT09F001 DU DSN=&&FILE09,DISP=(NEW,PASS),UNIT=WK10, 00000420
//          SPACE=(CYL,(60,10),,CONTIG),           00000430
//          ECB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000440
//FT17F001 DU DSN=&&FILE17,DISP=(OLD,DELETE)    00000450
//SYSPRINT DD SYSOUT=I                         00000460
//ERATO4 EXEC PGM=TEMPNAME,COND=(4,LT),          00000470
//          PARM='&DSN0&S&DSN1&S&DSN2&S&DSN3&S&DSN4' 00000480
//STEPLIB DU DSN=J3331.&DSN4..LOAD,DISP=SHR    00000490
//FT01F001 DD DSN=J3331.&DIARY..DATA,DISP=OLD   00000500
//FT50F001 DD DSN=&&FILE50,UNIT=WK10,DISP=(NEW,PASS), 00000510
//          SPACE=(TRK,(10,10))                  00000520
//FT06F001 DD SYSOUT=I                         00000530
//FT07F001 DU DSN=&&FILE07,DISP=(OLD,DELETE)    00000540
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,PASS)      00000550
//FT09F001 DU DSN=&&FILE09,DISP=(OLD,DELETE)    00000560
//FT10F001 DD DSN=&&FILE10,DISP=(NEW,PASS),UNIT=WK10, 00000570
//          SPACE=(TRK,(10,10)),                 00000580
//          ECB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000590
//FT33F001 DD DSN=&&ERT33,DISP=(NEW,DELETE),UNIT=WK10, 00000600
//          SPACE=(CYL,(60),,CONTIG),             00000610
//          ECB=(RECFM=F,BLKSIZE=1906c,DSORG=DA)   00000620

```

続<

(b) J3331.EJCL50W.CNTL for the path $(B_2, B_3 \rightarrow E'_4)$ in Fig.3-1.

(続き)

```

//ERATO5 EXEC PGM=TEMPNAME,COND=(4,LT)          00000630
//STEPLIB  DD DSN=J3331.&DSN5..LOAD,DISP=SHR   CCCCC640
//FT04F001 DD DSN=&&FILE04,DISP=(OLD,DELETE)  00000650
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,DELETE)  00000660
//FT10F001 DD DSN=&&FILE10,DISP=(OLD,DELETE)  00000670
//FT16F001 DD DSN=&&FILE16,DISP=(OLD,DELETE)  00000680
//FT22F001 DD DSN=&&FILE22,DISP=(NEW,PASS),UNIT=WK10,  00000690
//           SPACE=(TRK,(10,10))                  00C00700
//SYSPRINT DD SYSOUT=I                         00000710
//FT06F001 DD SYSOUT=I                         00000720
//ERATO6 EXEC PGM=ARGUS,COND=(4,LT)            00000730
//STEPLIB  DD DSN=J3331.E6.LOAD,DISP=SHR   00000740
//FT50F001 DD DSN=&&FILE50,DISP=(OLD,DELETE)  00000750
//FT22F001 DD DSN=&&FILE22,DISP=(OLD,DELETE)  00000760
//FT96F001 DD DSN=&&FILE96,DISP=(NEW,DELETE),UNIT=WK10,  00000770
//           SPACE=(TRK,(10,10))                  00000780
//FT97F001 DD DSN=&&FILE97,DISP=(NEW,DELETE),UNIT=WK10,  00000790
//           SPACE=(TRK,(10,10))                  00000800
//FT98F001 DD DSN=&&FILE98,DISP=(NEW,DELETE),UNIT=WK10,  00000810
//           SPACE=(TRK,(10,10))                  00000820
//FT99F001 DD DSN=&&FILE99,DISP=(NEW,DELETE),UNIT=WK10,  00000830
//           SPACE=(TRK,(10,10))                  00000840
//FT106F001 DD SYSOUT=I                         00000850
//FT78F001 DD SYSOUT=U,                         00000860
//           DCB=(BLKSIZE=160,LRECL=160,BUFL=160,BUFNO=1) 00000870
//COPY    EXEC PGM=JS0GGENER,COND=(4C5,LT)        00000880
//STEPLIB  DD DSN=SYS1LINKLIB,DISP=SHR       00000890
//SYSUT1  DD DSN=J3331.&DIARY..DATA,DISP=OLD  00000900
//SYSUT2  DD DSN=J3331.&DIARY.O.DATA,DISP=SHR,UNIT=(,,DEFER) 00000910
//SYSIN   DD DUMMY                           00000920
//SYSPRINT DD SYSOUT=I                         00000930
// PEND
//*
//-----*
//STEP1 EXEC ERATO,DSN0='J2870.IN005T40'      00000960
//ERATO1.FT05F001 DD *
&NEWRUN NPSI=14,NCHI=14,WNTURE=-3.0,ALO=-0.06,  00000980
      REXT=0.0,NDIVSI=0,SCALE=1.0,NITMAX=10 &END  00001000
//-----*
//                                         00001010
//                                         00001020

```

```

//JCLG JOB                                     00000010
// EXEC JCLS                                    00000020
//SYSIN DD DATA,DLM='++'                         00000030
// JUSER 54313331,SR.TOKUDA,0021.120,ESGUIDPM   00000040
// T.2 W.2 C.5 P.0 I.5                           00000050
//OPTP MSGCLASS=I,MSGLEVEL=(1,1)                 00000060
//***** THIS JCL FOR GUESS VALUE ESTIMATION. *
//*:                                         * 00000070
//*: DIARY FILE === J3331.DIARYG.DATA          *
//*:                                         * 00000080
//*:                                         * 00000090
//*:                                         * 00000100
//*:                                         * 00000110
//*****                                         00000120
//ERATO PROC DSN0=,DSN1=E1D50F$5,DSN2=E2D50F,DSN3=E3D50F,DSN4=E4D50F, 00000130
//          DIARY=DIARYG,S=#
//RUN1 EXEC PGM=TEMPNAME,COND=(4095,LT)          00000140
//STEPLIB DD DSN=J3331.&DSN1..LOAD,DISP=SHR      00000150
//FT04F001 DD DSN=&&FILE04,UNIT=WK10,DISP=(NEW,PASS), 00000160
//          SPACE=(TRK,(10,10))                   00000170
//FT06F001 DD SYSOUT=I                           00000180
//FT08F001 DD DSN=&&FILE03,UNIT=WK10,DISP=(NEW,PASS), 00000190
//          SPACE=(TRK,(10,10))                   00000200
//FT11F001 DD DSN=&DSNG..DATA,DISP=SHR          00000210
//FT16F001 DD DSN=&&FILE16,UNIT=WK10,DISP=(NEW,DELETE), 00000220
//          SPACE=(TRK,(10,10))                   00000230
//FT17F001 DD DSN=&&FILE17,UNIT=WK10,DISP=(NEW,PASS), 00000240
//          SPACE=(TRK,(10,10))                   00000250
//SYSPRINT DD SYSOUT=I                           00000260
//RUN2 EXEC PGM=TEMPNAME,COND=(4095,LT)          00000270
//STEPLIB DD DSN=J3331.&DSN2..LOAD,DISP=SHR      00000280
//FT06F001 DD SYSOUT=I                           00000290
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,PASS)       00000300
//FT17F001 DD DSN=&&FILE17,DISP=(OLD,PASS)       00000310
//SYSPRINT DD SYSOUT=I                           00000320
//RUN3 EXEC PGM=TEMPNAME,COND=(4095,LT)          00000330
//STEPLIB DD DSN=J3331.&DSN3..LOAD,DISP=SHR      00000340
//FT04F001 DD DSN=&&FILE04,DISP=(OLD,DELETE)     00000350
//FT06F001 DD SYSOUT=I                           00000360
//FT07F001 DD DSN=&&FILE07,UNIT=WK10,DISP=(NEW,PASS), 00000370
//          SPACE=(CYL,(60,10),,CONTIG),           00000380
//          DCB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000390
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,PASS)       00000400
//FT09F001 DD DSN=&&FILE09,DISP=(NEW,PASS),UNIT=WK10, 00000410
//          SPACE=(CYL,(60,10),,CONTIG),           00000420
//          DCB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000430
//FT17F001 DD DSN=&&FILE17,DISP=(OLD,DELETE)     00000440
//SYSPRINT DD SYSOUT=I                           00000450
//RUN EXEC PGM=TEMPNAME,COND=(4095,LT),          00000460
//          PARM='&DSN0&S&DSN1&S&DSN2&S&DSN3&S&DSN4'
//STEPLIB DD DSN=J3331.&DSN4..LOAD,DISP=SHR      00000470
//FTC1F001 DD DSN=J3331.&DIARY..DATA,DISP=OLD    00000480
//FT50F001 DD DSN=&&FILE50,UNIT=WK10,DISP=(NEW,DELETE), 00000490
//          SPACE=(TRK,(10,10))                   00000500
//FT06F001 DD SYSOUT=I                           00000510
//FT07F001 DD DSN=&&FILE07,DISP=(OLD,DELETE)     00000520
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,DELETE)     00000530
//FT09F001 DD DSN=&&FILE09,DISP=(OLD,DELETE)     00000540
//FT10F001 DD DSN=&&FILE10,DISP=(NEW,DELETE),UNIT=WK10, 00000550
//          SPACE=(TRK,(10,10)),                  00000560
//          DCB=(RECFM=VBS,BLKSIZE=19000,LRECL=18996) 00000570
//FT33F001 DD DSN=&&ERT33,DISP=(NEW,DELETE),UNIT=WK10, 00000580
//          SPACE=(CYL,(60),,CONTIG),             00000590
//          DCB=(RECFM=F,BLKSIZE=19068,DSORG=DA)    00000600
//                                          00000610
//                                          00000620

```

続く

(c) J3331.EJCL50G.CNTL for the path ($B_2, B_3 \rightarrow E'_1$) in Fig.3-1.

(続き)

```
//COPY EXEC PGM=JSOGENER,COND=(4095,LT)          00000630
//STEPCLIB DD DSN=SYS1LINKLIB,DISP=SHR           00000640
//SYSUT1 DD DSN=J3331.&DIARY..DATA,DISP=OLD        000C0650
//SYSUT2 DD DSN=J3331.&DIARY..0..DATA,DISP=SHR,UNIT=(,,DEFER) 00000660
//SYSIN DD DUMMY                                00000670
//SYSPRINT DD SYSOUT=I                          00000680
// PEND
///*
//*-----*
//STEP1 EXEC ERATO,DSNC='J2870.INC05T40'      00000710
//RUN1.FT05F001 DD *
&NEWRUN NPSI=12,NCHI=12,WNTCRE=-3.0,AL0=-0.06,    00000720
REXT=0.0,NDIVSI=0,SCALE=1.0,NITMAX=10 &END       00000730
//*-----*
++                                         00000740
//                                         00000750
//                                         00000760
//                                         00000770
```

Table 3-3 NAMELIST Input Parameters for ERATO Run

Quantity	Type	Default Value	Meaning
AL0	R	-0.15	Eigenvalue guess ω_0^2
REXT	R	0.0	Defines conducting wall
WNTORE	R	-2.0	Toroidal wave number, n
NCHI	I	15	Number of poloidal intervals, N_χ
NPSI	I	14	Number of radial intervals, N_ψ
NITMAX	I	10	Maximum number of iteration
NFIG	I	0	Plots flow pattern if ≠ 0
IVAL	I	0 (2)	Prints diagnostic data if = 2

から、最初と最後の部分を取り除いたものである。

定常的な計算実行の為には、上記 JCL プログラム、EJCL 50M (M ← MT job), EJCL 50L (L ← Large job) を使えばよいが、ERATO 6を含めた全プログラムを起動する際には、J 3331. EJCL 50 W. CNTL (第 3 - 13(b) 図, W ← Whole program job) を、また、磁気テープに中間データを保存しない時には、J 3331. EJCL 50 G. CNTL (第 3 - 13(c) 図, G ← Guess value estimation job) を使えばよい。実際の計算実行にあたっては、下に示すように原 JCL プログラムを J 3331 のファイルからコピーして使用するのが適当である。

```
ALLOC F(F 1) DA(EJCL 50□.CNTL) SP(2 2)
                                T CA NEW
COPY 'J 3331. EJCL 50□. CNTL' EJCL 50□. CNTL
```

ここで、□ の部分には、G, L, M, Wが入る。

3.6 DIARY FILE の参照

DIARY FILE (DIARYG) に蓄積された計算結果を参照する為に、3種類のプログラム、PRINT0, PRINT1, ERTG が用意されている (第 3 - 1 表)。

3.6.1 PRINT0およびPRINT1

PRINT0およびPRINT1は、計算結果の主要部と表の形で出力するプログラムである。

DIARY FILE (J 3331.DIARYG.DATA) の内容を PRINT0 プログラムで出力するには、下に示すように行えよ。

```

//JCLG JOB
// EXEC JCLG
//SYSIN DD DATA,DLM='++'
// JUSER .....
T._W._C.5I.__
//*
// DD DSN=J3331.EJCL50L.CNTL,DISP=SHR

```

Fig.3-14 JCL program for a large-scale ERATO calculation.

Table 3-4 Parameters in the NAMELIST (&PRI) for line printer output of the DIARY FILE

Parameter	Meaning
NO	Data with serial numbers which are larger than or equal to NO are printed out.
WN	Toroidal mode number (n)
REXT	Position of a conducting wall (b/a)
DSN	'Name of equilibrium data file (# name of load module)'

(1) TSS での実行

PROF PRE (J 3331)

EX 'J 3331. TSS. CLIST (PRINT 0)' DA (DIARYG)

(2) リモート・バッチ・ジョブとしての実行

GEM 3 'J 3331. JCL . CNTL' ; U PRINT 0 ; SUB ; EOM ; END

PRINT 1 プログラムを実行する場合には、PRINT 0 とある所を PRINT 1 に変更すればよい。

なお、PRINT 0 および PRINT 1 の実行に必要な入力データは第 3 - 4 表に示すとおりでネームリスト形式 (&PRI) で読み込まれる。リモート・バッチ・ジョブの場合は、J 3331. JCL . CNTL の内容のうち、&PRI の部分を修正してからジョブの投入を行わねばならない。PRINT 0 の実行例を第 3 - 15 図に、また、PRINT 1 の実行例を第 3 - 16 図に示す。

3.6.2 CHANGE プログラム

CHANGE プログラムのロード・モジュールは、J 3331.CHANGE . LOAD に格納されており、これを使って "Check flag" を書き替えるには、以下のような手順に従えばよい。

(1) DIARY FILE の名前を変える。

例えば、DIARY (旧) → DIARY 1 (新)。

即ち、RENAME DIARY . DATA DIARY1 . DATA

(2) Check flag 修正後のデータを入れる為に、DIARY FILE を確保 (旧名ファイル) してファイル定義名を FT 02 F 001 とする。

即ち、

```
ALLOC F (FT 02 F 001) DA (DIARY . DATA)
      SPACE (12 1) T NEW UNIT (USER)
```

(3) 修正前のDIARY FILE (この場合、DIARY1) のファイル定義名を FT 01 F 001 とする。

即ち、

ALLOC F (FT 01 F 001) DA (DIARY1 . DATA) SHR

(4) CHANGE プログラムを実行する。

即ち、

CALL 'J 3331. CHANGE . LOAD'

(5) 必要な修正を行う。

即ち、計算機からの問い合わせ(?)に対して、下に示すように (下線部分) 入力する事によって修正が行われる。

INPUT "SERIAL NUMBER"

? 2

①

NO NPSI NCHI NORMALIZATION } ②

2 14 14 - 0.18717 }

0 ... NOUPDATE 1 ... UPDATE 10 ... END OF UPDATE ③

Nu	Np3	Nch3	MULTILAYER	MAXLAYER	N11	NEG	Gauss	REAL	N	NchE	NchL	ENUL
1	45	25	-0.626200U-02	-0.626200U-02	5	4	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
2	30	40	-0.246200U-03	-0.246200U-03	7	5	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002120
3	45	25	-0.639600U-02	-0.639600U-02	0	3	-0.600000U-02	0.0	-3.0	1	0	J2870..IN002110
4	45	25	-0.639600U-02	-0.639600U-02	10	2	-0.600000U-02	0.0	-3.0	1	0	J2870..IN002110
5	20	30	-0.589960U-03	-0.589960U-03	5	6	-0.600000U-03	0.0	-2.0	1	0	J2870..IN002110
6	45	25	-0.709350U-01	-0.709350U-01	16	1	-0.500000U-01	0.0	-3.0	1	0	J2870..IN002110
7	70	25	-0.707200U-01	-0.707200U-01	3	0	-0.710000U-01	0.0	-3.0	1	0	J2870..IN002110
8	31	34	-0.456490U-03	-0.456490U-03	13	0	-0.600000U-01	0.0	-2.0	1	0	J2870..IN002110
9	31	34	-0.105940U-02	-0.105940U-02	11	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
10	31	31	-0.169210U-02	-0.169210U-02	11	1	-0.130000U-02	0.0	-2.0	1	0	J2870..IN002110
11	31	41	-0.164310U-02	-0.164310U-02	11	1	-0.140000U-02	0.0	-2.0	1	0	J2870..IN002110
12	31	31	-0.338800U-01	-0.338800U-01	11	0	-0.700000U-01	0.0	-3.0	1	0	J2870..IN002110
13	33	33	-0.257430U-01	-0.257430U-01	6	0	-0.300000U-01	0.0	-3.0	1	0	J2870..IN002110
14	33	33	-0.140670U-02	-0.140670U-02	5	0	-0.140000U-02	0.0	-2.0	1	0	J2870..IN002110
15	33	33	-0.140660U-02	-0.140660U-02	3	1	-0.140000U-02	0.0	-2.0	1	0	J2870..IN002110
16	37	57	-0.1462660U-02	-0.1462660U-02	6	0	-0.146000U-02	0.0	-2.0	1	0	J2870..IN002110
17	35	35	-0.197740U-03	-0.197740U-03	6	0	-0.277000U-01	0.0	-3.0	1	0	J2870..IN002110
18	30	30	-0.117430U-02	-0.117430U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
19	32	32	-0.145770U-02	-0.145770U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
20	34	34	-0.1192520U-02	-0.1192520U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
21	30	36	-0.140650U-02	-0.140650U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
22	30	40	-0.1059260U-02	-0.1059260U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
23	30	40	-0.117430U-02	-0.117430U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
24	36	36	-0.308440U-01	-0.308440U-01	6	0	-0.200000U-01	0.0	-3.0	1	0	J2870..IN002110
25	32	32	-0.294260U-01	-0.294260U-01	11	0	-0.200000U-01	0.0	-3.0	1	0	J2870..IN002110
26	34	36	-0.140650U-02	-0.131320U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002110
27	29	29	-0.1059260U-02	-0.1059260U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002120
28	35	35	-0.140650U-02	-0.140650U-02	31	0	-0.600000U-02	0.0	-2.0	1	0	J2870..IN002120
29	30	30	-0.308440U-01	-0.308440U-01	7	0	-0.200000U-01	0.0	-3.0	1	0	J2870..IN002120
30	32	32	-0.294260U-01	-0.294260U-01	7	0	-0.200000U-01	0.0	-3.0	1	0	J2870..IN002120
31	31	31	-0.5626280U-01	-0.5626280U-01	6	0	-0.630000U-01	0.0	-3.0	1	0	J2870..IN002120
32	34	34	-0.2226670U-01	-0.2226670U-01	16	0	-0.300000U-01	0.0	-3.0	1	0	J2870..IN002110
33	29	29	-0.1059260U-02	-0.1059260U-02	4	1	-0.100000U-01	0.0	-3.0	1	0	J2870..IN002120
34	37	37	-0.495150U-01	-0.495150U-01	4	0	-0.500000U-01	0.0	-3.0	1	0	J2870..IN002120
35	37	37	-0.425070U-01	-0.425070U-01	5	0	-0.450000U-01	0.0	-3.0	1	0	J2870..IN002120
36	35	35	-0.125560U-01	-0.125560U-01	5	0	-0.122000U-00	0.0	-3.0	1	0	J2870..IN002120
37	35	35	-0.373750U-01	-0.373750U-01	4	1	-0.572000U-01	0.0	-3.0	1	0	J2870..IN002120
38	40	40	-0.571050U-01	-0.571050U-01	6	0	-0.400000U-01	0.0	-3.0	1	0	J2870..IN002120
39	37	37	-0.125560U-01	-0.125560U-01	10	1	-0.500000U-01	0.0	-3.0	1	0	J2870..IN002120
40	35	35	-0.571190U-01	-0.571190U-01	6	1	-0.450000U-01	0.0	-3.0	1	0	J2870..IN002120
41	33	33	-0.371170U-01	-0.371170U-01	6	0	-0.400000U-01	0.0	-3.0	1	0	J2870..IN002120
42	35	35	-0.125560U-01	-0.125560U-01	3	0	-0.620000U-01	0.0	-3.0	1	0	J2870..IN002120
43	37	37	-0.571190U-01	-0.571190U-01	4	1	-0.500000U-01	0.0	-3.0	1	0	J2870..IN002120
44	37	37	-0.213160U-01	-0.213160U-01	4	1	-0.500000U-01	0.0	-3.0	1	0	J2870..IN002120
45	35	35	-0.453360U-01	-0.453360U-01	4	1	-0.400000U-01	0.0	-3.0	1	0	J2870..IN002120
46	35	35	-0.125560U-01	-0.125560U-01	4	1	-0.400000U-01	0.0	-3.0	1	0	J2870..IN002120

Fig. 3-15 Example 4) of output lists by the PRINTO program.

NO	NPSI	NCAL	NORMALIZATION	RAYLEIGH	NIT	MAX NEG	GUESS	REXI	N	NCHEK	SI	NCOMP	TIME	DATE		
1	25	25	-0.626010-02	-0.622610-02	5	6	1	-0.600000-02	0.0	-2.0	1	3952	14:49: 6	80-02-25		
	DSN = J287C.IN002110#E1D40551#E3D40#E3D40#E4D40															
2	20	20	-0.572090-03	-0.169260-03	7	6	0	-0.600000-02	0.0	-2.0	1	0	3194	15:53:40	80-02-25	
	DSN = J287C.IN002110#E1D40551#E3D40#E3D40#E4D40															
3	25	25	-0.639610-02	-0.639610-02	6	6	3	-0.600000-02	0.0	-3.0	1	0	3952	16:37:28	80-02-25	
	DSN = J287C.IN002110#E1D40551#E3D40#E3D40#E4D40															
4	25	25	-0.639920-02	-0.639920-02	10	9	2	-0.100000-01	0.0	-3.0	1	0	2883	3952	17: 0:45	80-02-25
	DSN = J287C.IN002110#E1D40551#E3D40#E4D40															
5	30	30	-0.589540-03	-0.589540-03	5	6	6	-0.600000-03	0.0	-2.0	1	0	5642	5642	17:17:51	80-02-25
	DSN = J287C.IN002110#E1D40551#E2D40#E3D40#E4D40															
6	25	25	-0.702320-01	-0.706580-01	10	9	1	-0.500000-01	0.0	-3.0	1	0	1395	3952	17:30:46	80-02-25
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
7	25	25	-0.707570-01	-0.707570-01	3	9	0	-0.710000-01	0.0	-3.0	1	0	3952	3952	19: 5:52	80-02-25
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
8	31	31	-0.285490-03	-0.156280-03	13	12	0	-0.700000-01	0.0	-2.0	1	0	5507	6016	19:28: 4	80-02-25
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
9	31	31	-0.105950-02	-0.128170-02	11	10	0	-0.600000-02	0.0	-2.0	1	0	2006	6016	20:37:19	80-02-25
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
10	31	31	-0.109510-02	-0.110610-02	11	10	1	-0.130000-02	0.6	-2.0	1	0	2349	6016	21: 0:28	80-02-25
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
11	31	31	-0.164310-02	-0.114330-02	11	10	1	-0.140000-02	0.0	-2.0	1	0	5638	6016	21:17:46	80-02-25
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
12	31	31	-0.33e630-01	-0.383870-01	11	10	0	-0.700000-01	0.0	-3.0	1	0	5066	6016	9:30:44	80-02-26
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
13	33	33	-0.257430-01	-0.257430-01	6	10	0	-0.300000-01	0.0	-3.0	1	0	6800	6800	12:14:13	80-02-26
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
14	33	33	-0.140670-02	-0.140670-02	8	10	0	-0.164000-02	0.0	-2.0	1	0	6800	6800	12:30: 7	80-02-26
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
15	35	35	-0.146460-02	-0.140460-02	3	10	1	-0.140000-02	0.0	-2.0	1	0	7632	7632	13:23:30	80-02-26
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
16	37	37	-0.126560-02	-0.126580-02	6	5	0	-0.140000-02	0.0	-2.0	1	0	7249	8512	15:57: 2	80-02-26
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
17	35	35	-0.197740-01	-0.197720-01	6	5	0	-0.257000-01	0.0	-3.0	1	0	6809	7632	16:18:50	80-02-26
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
18	30	30	-0.179130-02	-0.179520-02	31	30	0	-0.600000-02	0.0	-2.0	1	0	5267	5642	2:43:26	80-02-27
	DSN = J287C.1N002110#E1D40551#E2D40#E3D40#E4D40															
19	32	32	-0.145570-02	-0.145540-02	31	30	0	-0.603000-02	0.0	-2.0	1	0	5961	6402	3:17: 3	80-02-27

Fig. 3-16 Example 4) of output lists by the PRINTL program.

```

? 1                                ④
? 99                               ⑤
NO NPSI NCHI NORMALIZATION ..... } ②
3   14   14   - 0.1786 ..... } ②
0 ... NOUPDATE 1 ... UPDATE 10 ... END OF UPDATE ..... } ③
? 0                                ⑥
NO NPSI NCHI NORMALIZATION ..... } ②
4   14   14   - 0.1876 ..... } ②
0 ... NOUPDATE 1 ... UPDATE 10 ... END OF UPDATE ..... } ③
? 10                               ⑦
END OF UPDATE

```

ここで、各行は次のような意味をもつ。

- ① 修正をする最初のデータの“SERIAL NUMBER”的入力
- ② 修正の対象となる固有値情報の出力。
- ③ 修正を行うか否かについての問い合わせに対する答え方の指示。
- ④ 修正する事を示す。
- ⑤ NCHEKの修正値の入力。
- ⑥ 修正しない事を示す。
- ⑦ 修正作業の終了を示す。

3.6.3 ERTG プログラム

ERTGは、図形入出力サブシステム、ARGUS¹¹⁾のもとで動くプログラムで、現在、標準的に用意されているのは、下に示すようなテクトロニクス端末用のTSS マクロ命令（コマンド・プロシージャ）である。

即ち、

```

EX 'J 3331. TSS. CLIST (ARGUS)' 'LOAD (ERTG)
      F 01 (DIARYG)'

```

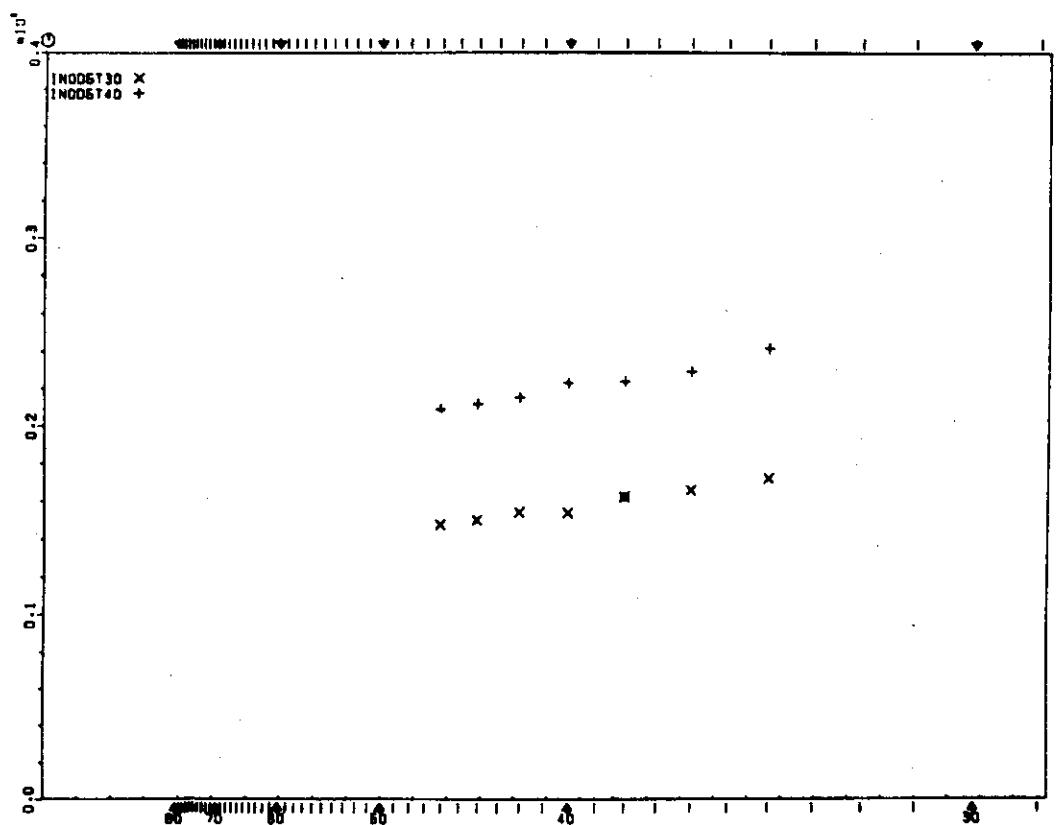
ERTGでは、希望するデータのみを出力する為にデータを選別する為のネームリスト入力 & COMがある。入力パラメータとそのデフォルト値は第3-5表に示すとおりである。この表にある2つ以上のパラメータを指定して図を描く場合には、全ての条件を満すデータのみを出力する。

ERTGの機能は、既に述べたように、不安定性の成長率の二乗（固有値）を $N\psi^{-2}$ の関数として図示（収束曲線の図示）するプログラムであるが、一本の収束曲線上に属する点の中には異なる平衡データ・ファイル、異なるロード・モジュール（ERATO 1 ロード・モジュール）を使って得られたものが存在する。これらの点のうち希望の条件を満すデータ点のみを選択的に表示することも、第3-5表の入力パラメータの指定によって可能である。

第3-17図に示したのは、INTOR型プラズマの不安定解析⁴⁾を行った際にERTGを使って得た収束曲線の例である。

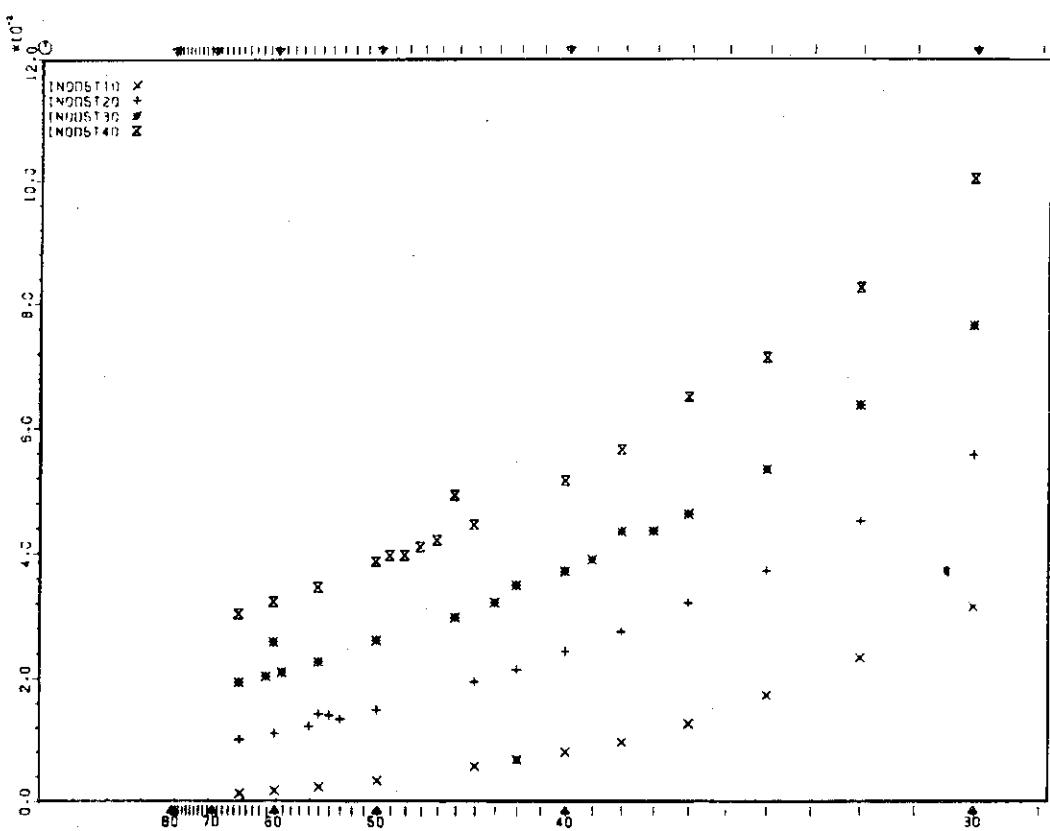
Table 3-5 Parameters in the NAMELIST (&COM)
for graphic output of the DIARY FILE

Parameter	Meaning [default value]
NO	Data with serial number which are larger than or equal to NO are plotted [1]
WN	Toroidal mode number (n)
REXT	Position of a conducting wall (b/a)
DSN	'Name of equilibrium data file (# name of load module)'
YMAX	Maximum value of the ordinate[1.0]
YMIN	Minimum value of the ordinate[0.0]
XMAX	Maximum value of the abscissa (Mesh number)
IS	Scale option for the ordinate IS = 1 Automatic scaling IS = 4 Manual scaling



(a) &COM? WN=-1.0, YMAX=0.4, &END (n=1 kink mode).

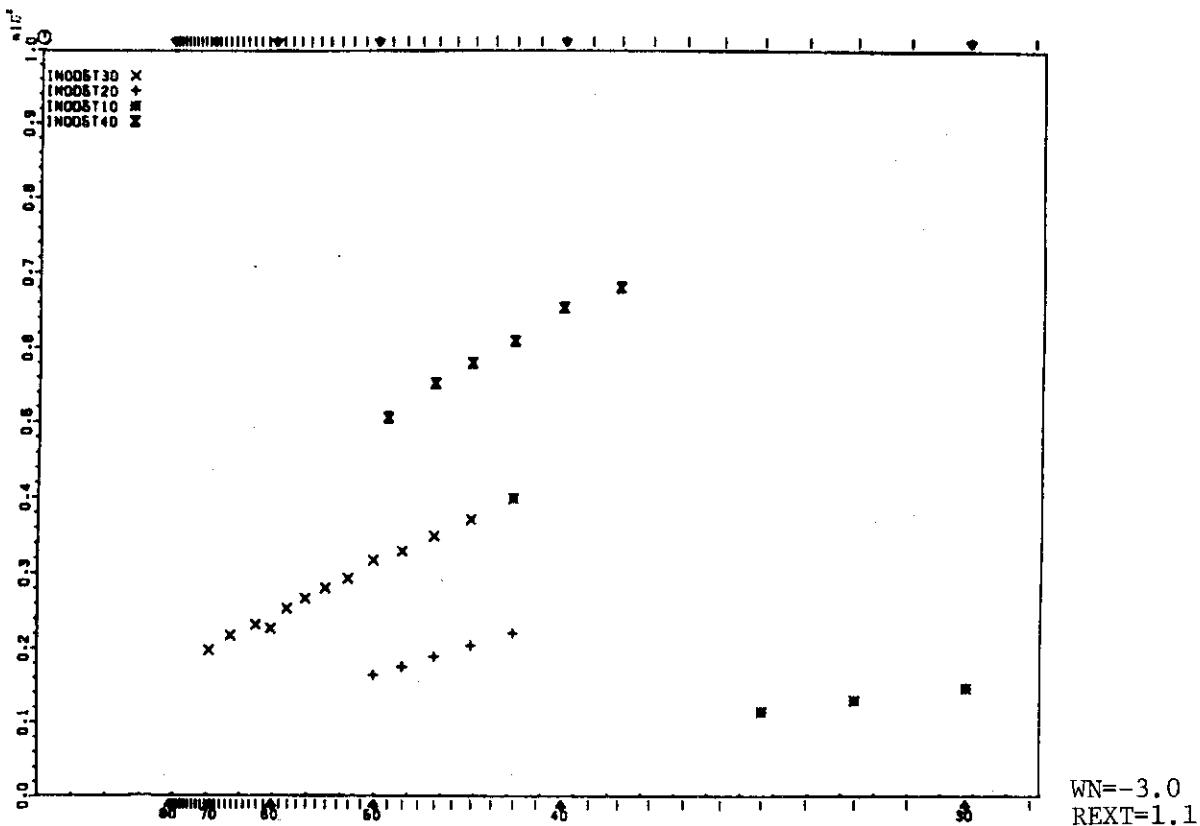
WN=-1.0
REXT=1.4
YMAX=0.4



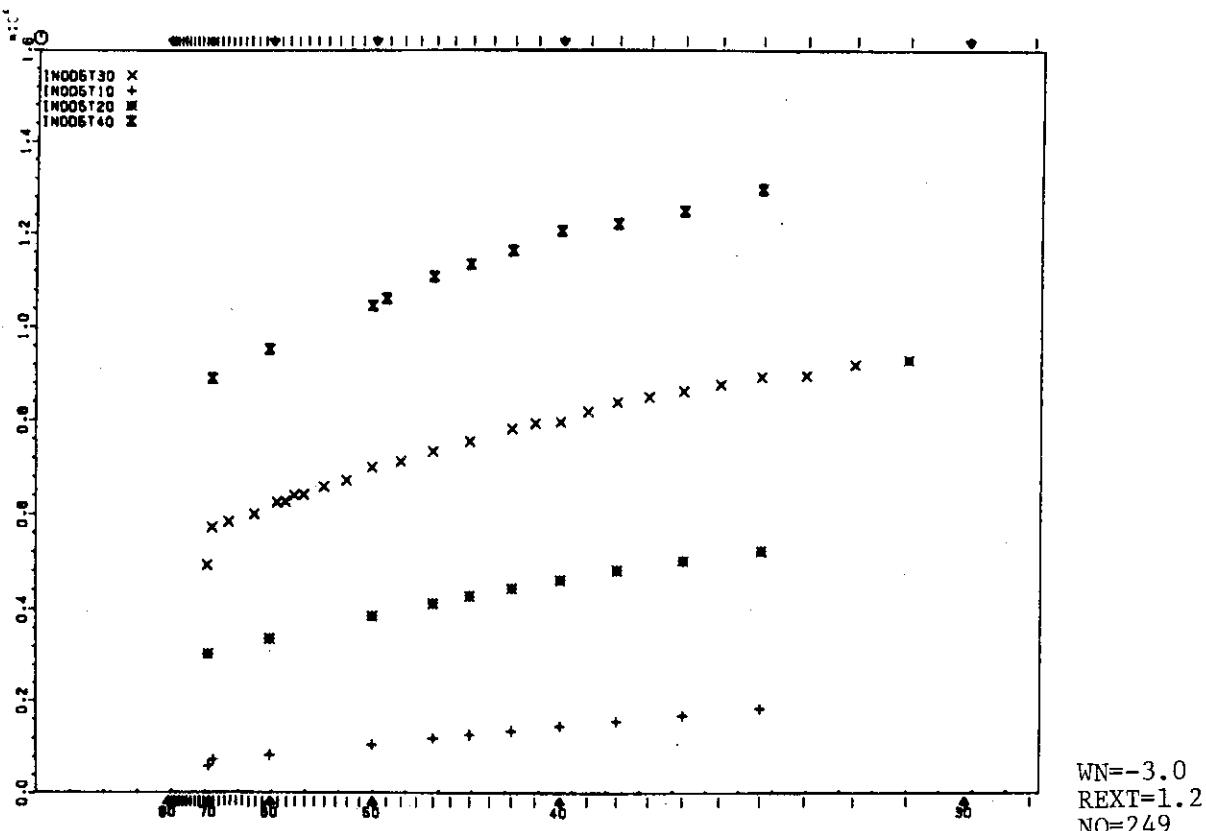
(b) &COM? WN=-3.0, REXT=0.0, &END (n=3 internal mode).

WN=-3.0
REXT=0.0

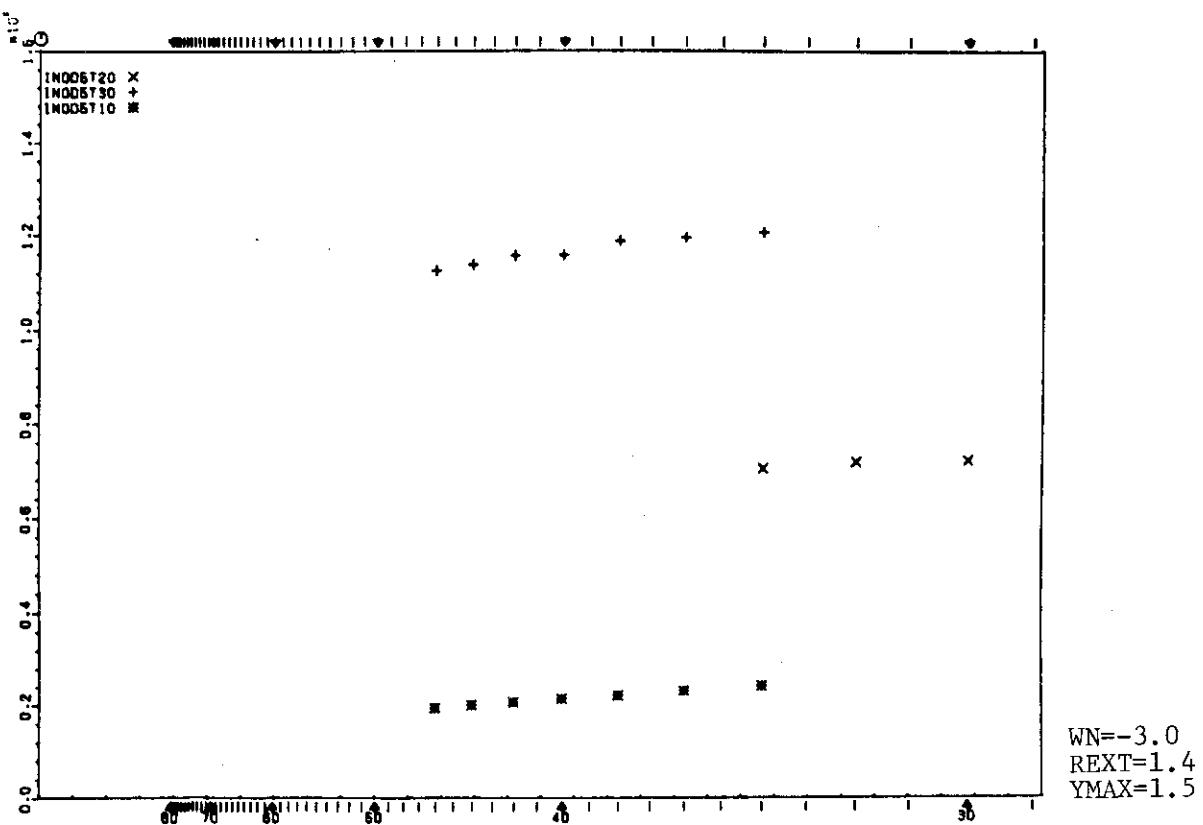
Fig. 3-17 Examples⁴⁾ of output graphs by the ERTG program.



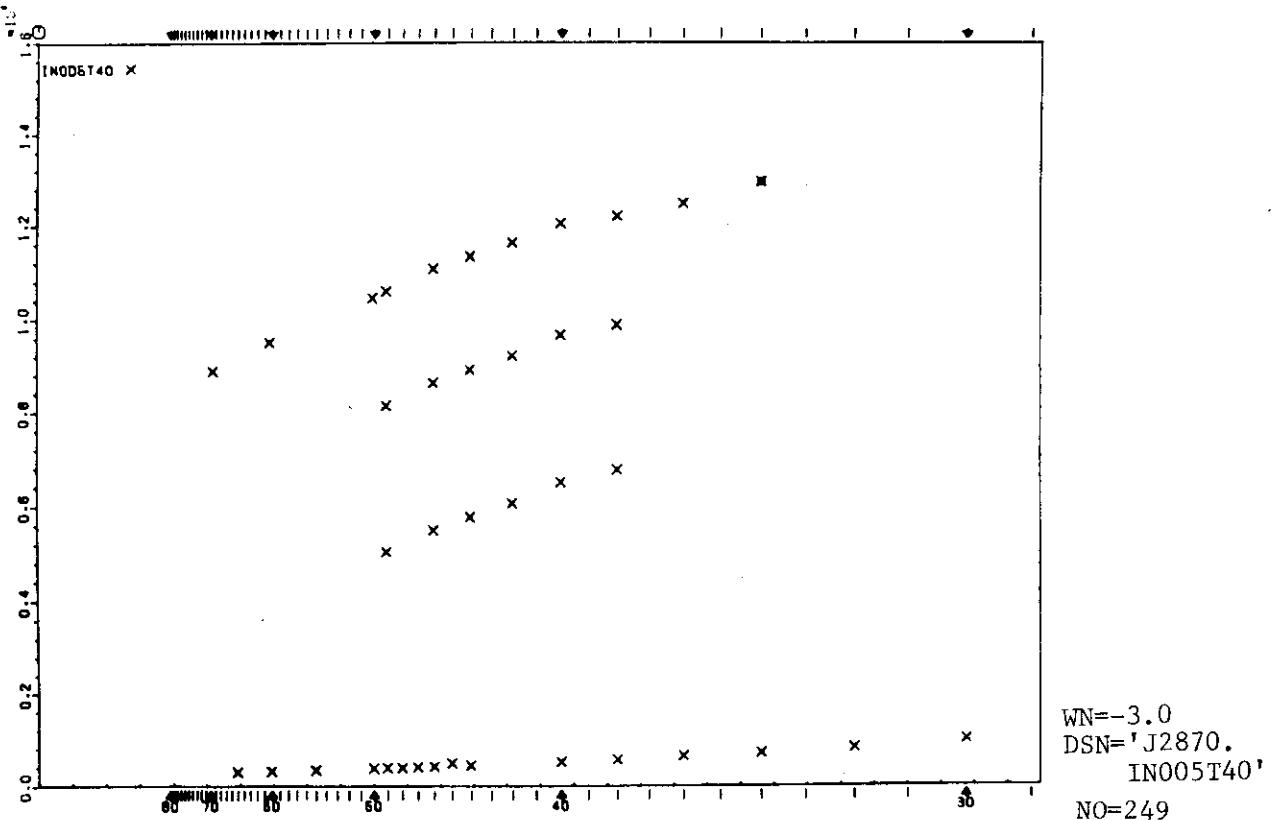
(c) &COM? WN=-3.0, REXT=1.1, &END (n=3 kink mode).



(d) &COM? WN=-3.0, REXT=1.2, NO=249, &END (n=3 kink mode).



(e) &COM? WN=-3.0, REXT=1.4, YMAX=1.5, &END (n=3 kink mode).



(f) &COM? WN=-3.0, DSN='J2870.IN005T40', NO=249, &END (n=3 kink mode).

3.7 磁気テープに保管されている情報の利用

ERATO-J(F版)では、平衡データから得られたメトリック量、固有ベクトル、局所エネルギー分布等の大量の情報を磁気テープに書き出し、必要に応じていつでも再現できるようにプログラム、JCLプログラムが整備されている。この為に用意されているプログラムのロード・モジュールに関する制限事項はメッシュ数の最大値が70までという事である。これ以上のメッシュ数を必要とする場合を扱うにはEOSプリプロセッサーを利用して新規にロード・モジュールを作成しなければならない。

関連するプログラムは、J 3331. EMTDUMP 0. FORT、およびJ 3331. MTIN. FORTで、ソース・プログラム・リストは、第3-18図および第3-9図に示してある。これらに対応するロード・モジュールは、J 3331. EMTDUMP 0. LOAD およびJ 3331. MTIN70D. LOADである。

JCLプログラムは、3種類用意されていて、次のような使い方をする(第3-19図)。

(a) J 3331. EMTDUMP 0. CNTL

磁気テープに格納されている情報を確認する為のもので、実行にあたって入力データは必要ない。出力されるデータは、磁気テープに含まれている主要な情報である。JCLプログラムは第3-20(a)図に示す。このプログラムの出力結果は、MTから出力されたものであるという点を除けばPRINT 0の出力結果(第3-15図参照)と同一である。

(b) J 3331. EMTDUMP 1, CNTL

磁気テープに含まれている情報をERATO 5で編集してライン・プリンター上に出力するもので入力データは“Serial number”である。JCLプログラムは第3-20(b)図に示す。INTORプラズマの解析過程で得た出力例を第3-21図に示す。

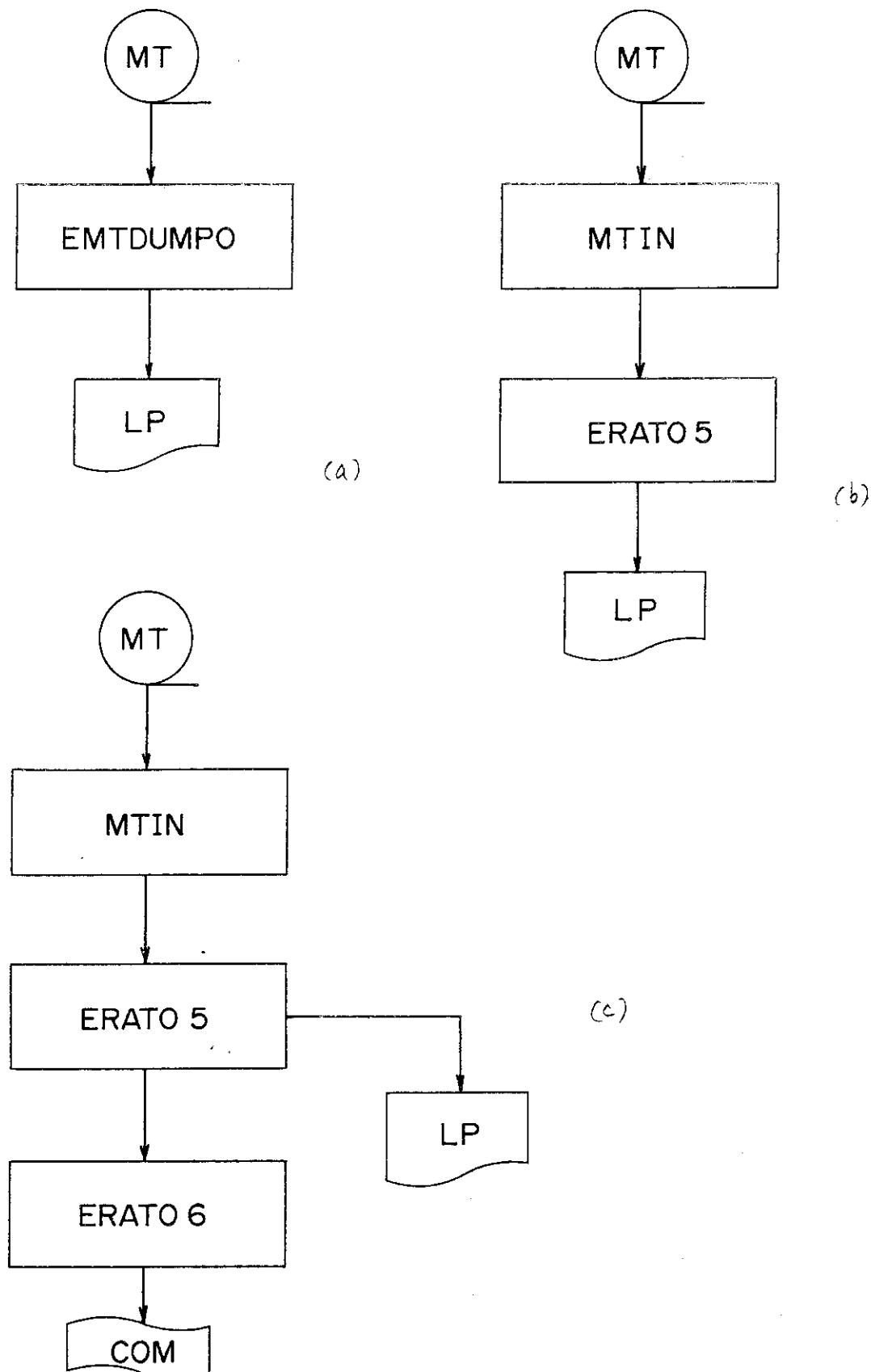
(c) J 3331. EMTDUMP 2, CNTL

磁気テープに含まれている情報をERATO 5で編集してライン・プリンター上に出力すると同時に、ERATO 6で图形化してCOMフィルムに出力する。入力データは(b)と同じである。JCLプログラムは、第3-20(c)図に示すとおりである。このJCLプログラムによるライン・プリンター出力は、EMTDUMP 1の場合と全く同じものが得られる。第3-21図に示した計算例についてのCOM出力を第3-22図に示す。

FACOM OSIV/F4 GEM VOL2LOGE DATE 80.07.22 TIME 11.34.57 LIB=J9079.ERTMTINO.FORT

NO.1	MODULE NAME	MAIN	BLOCKS	3	7-R-----6	-----8
LEVEL	16		DATE	80.04.28	TIME	15.19.19
<hr/>						
*DECK FMAIN						
IMPLICIT REAL*8(A-H,D-Z)						
*CALL COMBL5						
*CALL COMEQU						
*CALL NEQRUN						
LOGICAL NLREG						
INTEGER IDATE(2)						
LOGICAL*1 DSN160)						
REAL*8 AL(3)						
C-----						
NDES = 16						
NSAVE = 8						
WRITE(6,600)						
C-----						
EIGEN VALUE INFORMATION						
C-----						
50 CONTINUE						
READ(6U,END=410)						
* NU,NCHK,IITIME,IDATE,DSN,NPSI,NCHI,*NTURE,REXT,AL,NEG,NCONV,						
* NCOMP,NIT,NITMAX,NUIVSI						
C-----						
NAMELIST (NSAVE=0)						
C-----						
READ (6U ,NCHRUN)						
C-----						
WRITE INFORMATION						
C-----						
WRITE(6,610) NU,NPSI,NCHI,AL(2),AL(3),NIT,NEG,AL(1),						
* REXT,*NTURE,NCHK,NUIVSI,(DSN(JU),JU=1,14)						
C-----						
SAIP DATA						
C-----						
READ(60)						
DO 320 JPSI=1,NPSI						
READ(60)						
320 CONTINUE						
DO 330 JPSI=1,NPSI+1						
READ(60)						
READ(60)						
READ(60)						
READ(60)						
330 CONTINUE						
GO TO 50						
C-----						
STOP						
C-----						
410 CONTINUE						
600 FORMAT(1H1,'NU',1X'NPSI',1X'NCHI',2X'NORMALIZATION',4X,						
*RAYLEIGH',4X'NIT',1X'NEG',2X'GUESS',6X'REXT',6X'N',2X'NCHK',						
* 1X'NUIVSI',4X'EQU'L')						
610 FORMAT(1H ,13,2X2(1Z,2X),2(1X,E13.5),2(2X,1Z),1X,E13.5,2X						
* ,F7.3,2X,F5.1,2X,1Z,2X,1Z,2X,14A1)						
C-----						
STOP						
END						

Fig. 3-18 Source program list of EMTDUMPO (EOS source).



(a) Flow chart for the JCL program EMTDUMPO.

(b) Flow chart for the JCL program EMTDUMP1.

(c) Flow chart for the JCL program EMTDUMP2.

```

//JCLG JOB                                     00000010
// EXEC JCLG                                     00000020
//SYSIN DD DATA,DLM='++'                         00000030
// JUSER 54313331,SU.TOKUDA,0021.120,ESGUIDPM   00000040
T.O W.1 C.5 P.0 I.5                           00000050
OPTP MSGCLASS=I MSGLEVEL=(1,1)                 00000060
//***** THIS JCL FOR MT DUMPO                 * 00000061
//*****                                         * 00000062
//*****                                         * 00000063
//MTIN EXEC PGM=TEMPNAME,COND=(4,LT)           00000070
//STEPLIB DD DSN=J3331.EMTDUMPO.LOAD,DISP=SHR 00000080
//FT60F001 DD DSN=J2636.ERTDUMPO,VOL=SER=011096,UNIT=MT62, 00000090
//          LABEL=(1,SL),DISP=OLD               00000100
//FT50F001 DD DSN=&&FILE50,DISP=(NEW,PASS),UNIT=WK10, 00000110
//          SPACE=(TRK,(10,10))                  00000120
//FT04F001 DD DSN=&&FILE04,DISP=(NEW,PASS),UNIT=WK10, 00000130
//          SPACE=(TRK,(10,10))                  00000140
//FT08F001 DD DSN=&&FILE08,DISP=(NEW,PASS),UNIT=WK10, 00000150
//          SPACE=(TRK,(10,10))                  00000160
//FT10F001 DD DSN=&&FILE10,DISP=(NEW,PASS),UNIT=WK10, 00000170
//          SPACE=(TRK,(10,10))                  00000180
//FT16F001 DD DSN=&&FILE16,DISP=(NEW,PASS),UNIT=WK10, 00000190
//          SPACE=(TRK,(10,10))                  00000200
//SYSPRINT DD SYSOUT=I                          00000210
//FT06F001 DD SYSOUT=I                         00000220
//                                         *        00000230

```

Fig. 3-20 (a) JCL program EMTDUMPO.

```

//JCLG JOB                                     00000010
// EXEC JCLG                                     00000020
//SYSIN DD DATA,DLM='++'                         00000030
// JUSER 54313331,SU.TOKUDA,0021.120,ESGUIDPM   00000040
T.O W.1 C.5 P.0 I.5                           00000050
OPTP MSGCLASS=I MSGLEVEL=(1,1)                 00000060
//***** THIS JCL FOR MT DUMPI                 * 00000061
//*****                                         * 00000062
//*****                                         * 00000063
//MTIN EXEC PGM=TEMPNAME,COND=(4,LT)           00000070
//STEPLIB DD DSN=J3331.MTIND70.LOAD,DISP=SHR 00000080
//FT60F001 DD DSN=J2636.ERTDUMPO,VOL=SER=011096,UNIT=MT62, 00000090
//          LABEL=(1,SL),DISP=OLD               00000100
//FT50F001 DD DSN=&&FILE50,DISP=(NEW,PASS),UNIT=WK10, 00000110
//          SPACE=(TRK,(10,10))                  00000120
//FT04F001 DD DSN=&&FILE04,DISP=(NEW,PASS),UNIT=WK10, 00000130
//          SPACE=(TRK,(10,10))                  00000140
//FT08F001 DD DSN=&&FILE08,DISP=(NEW,PASS),UNIT=WK10, 00000150
//          SPACE=(TRK,(10,10))                  00000160
//FT10F001 DD DSN=&&FILE10,DISP=(NEW,PASS),UNIT=WK10, 00000170
//          SPACE=(TRK,(10,10))                  00000180
//FT16F001 DD DSN=&&FILE16,DISP=(NEW,PASS),UNIT=WK10, 00000190
//          SPACE=(TRK,(10,10))                  00000200
//SYSPRINT DD SYSOUT=I                          00000210
//FT06F001 DD SYSOUT=I                         00000220
//FT05F001 DD *                                00000230
5
//ERAT05 EXEC PGM=TEMPNAME,COND=(4,LT)         00000250
//STEPLIB DD DSN=J3331.E5D70F.LOAD,DISP=SHR 00000260
//FT04F001 DD DSN=&&FILE04,DISP=(OLD,DELETE) 00000270
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,DELETE) 00000280
//FT10F001 DD DSN=&&FILE10,DISP=(OLD,DELETE) 00000290
//FT16F001 DD DSN=&&FILE16,DISP=(OLD,DELETE) 00000300
//FT22F001 DD DSN=&&FILE22,DISP=(NEW,PASS),UNIT=WK10, 00000310
//          SPACE=(TRK,(10,10))                  00000320
//SYSPRINT DD SYSOUT=I                          00000330
//FT06F001 DD SYSOUT=I                         00000340
//                                         *        00000350

```

(b) JCL program EMTDUMPI.

```

//JCLG JOB                                     00000010
// EXEC JCLG                                     00000020
//SYSIN DD DATA,DLM='+++'                      00000030
// JUSER 54313331,SU.TOKUDA,0021.120,ESGUIDPM   00000040
//T.3 W.1 C.5 P.0 I.5                           00000050
OPTP MSGCLASS=I MSGLEVEL=(1,1)                 00000060
//***** THIS JCL FOR MT DUMP2 *               00000061
//*****                                         00000062
//*****                                         00000063
//MTIN  EXEC PGM=TEMPNAME,COND=(4,LT)          00000070
//STEPLIB  DD DSN=J3331.MTIND70.LOAD,DISP=SHR  00000080
//FT60F001 DD DSN=J2636.ERTDUMPO,VOL=SER=011096,UNIT=MT62, 00000090
//           LABEL=(1,SL),DISP=OLD                00000100
//FT50F001 DD DSN=&&FILE50,DISP=(NEW,PASS),UNIT=WK10, 00000110
//           SPACE=(TRK,(10,10))                  00000120
//FT04F001 DD DSN=&&FILE04,DISP=(NEW,PASS),UNIT=WK10, 00000130
//           SPACE=(TRK,(10,10))                  00000140
//FT08F001 DD DSN=&&FILE08,DISP=(NEW,PASS),UNIT=WK10, 00000150
//           SPACE=(TRK,(10,10))                  00000160
//FT10F001 DD DSN=&&FILE10,DISP=(NEW,PASS),UNIT=WK10, 00000170
//           SPACE=(TRK,(10,10))                  00000180
//FT16F001 DD DSN=&&FILE16,DISP=(NEW,PASS),UNIT=WK10, 00000190
//           SPACE=(TRK,(10,10))                  00000200
//SYSPRINT DD SYSOUT=I                         00000210
//FT06F001 DD SYSOUT=I                         00000220
//FT05F001 DD *                                00000230
5
//ERATO5 EXEC PGM=TEMPNAME,COND=(4,LT)          00000250
//STEPLIB  DD DSN=J3331.E5D70F.LOAD,DISP=SHR  00000260
//FT04F001 DD DSN=&&FILE04,DISP=(OLD,DELETE)    00000270
//FT08F001 DD DSN=&&FILE08,DISP=(OLD,DELETE)    00000280
//FT10F001 DD DSN=&&FILE10,DISP=(OLD,DELETE)    00000290
//FT16F001 DD DSN=&&FILE16,DISP=(OLD,DELETE)    00000300
//FT22F001 DD DSN=&&FILE22,DISP=(NEW,PASS),UNIT=WK10, 00000310
//           SPACE=(TRK,(10,10))                  00000320
//SYSPRINT DD SYSOUT=I                         00000330
//FT06F001 DD SYSOUT=I                         00000340
//ERATO6 EXEC PGM=ARGUS,COND=(4,LT)             00000350
//STEPLIB  DD DSN=J3331.E6.LOAD,DISP=SHR       00000360
//FT50F001 DD DSN=&&FILE50,DISP=(OLD,DELETE)    00000370
//FT22F001 DD DSN=&&FILE22,DISP=(OLD,DELETE)    00000380
//FT96F001 DD DSN=&&FILE96,DISP=(NEW,DELETE),UNIT=WK10, 00000390
//           SPACE=(TRK,(10,10))                  00000400
//FT97F001 DD DSN=&&FILE97,DISP=(NEW,DELETE),UNIT=WK10, 00000410
//           SPACE=(TRK,(10,10))                  00000420
//FT98F001 DD DSN=&&FILE98,DISP=(NEW,DELETE),UNIT=WK10, 00000430
//           SPACE=(TRK,(10,10))                  00000440
//FT99F001 DD DSN=&&FILE99,DISP=(NEW,DELETE),UNIT=WK10, 00000450
//           SPACE=(TRK,(10,10))                  00000460
//FT06F001 DD SYSOUT=I                         00000470
//FT78F001 DD SYSOUT=U,                         00000480
//           DCB=(BLKSIZE=160,LRECL=160,BUFL=160,BUFNO=1) 00000490
//                                         00000500

```

(c) JCL program EMTDUMP2.

***** FOURIER ANALYSIS *****

I. CS, C : KSI-PSI, KSI-CHI, KSI-PAI

1	-0.0147	1.00001	:	3.0400-03	3.3424E-03
2	-0.0441	1.0009	:	3.5680-03	4.3290-03
3	-0.0735	1.0026	:	4.4230-03	5.4260-03
4	-0.1029	1.0051	:	5.4430-03	6.5850-03
5	-0.1324	1.0085	:	6.6370-03	8.0580-03

Fig. 3-21 Example 4) of output lists by EMTDUMP1 and EMTDUMP2.

6	0.1612	1.0129	: c.32240-03	1.05330-04	-6.36950-03
7	0.1912	1.6163	: 5.2775-03	1.2165-04	-1.6375-02
8	0.2216	1.0249	: 1.2265-02	1.2655-02	-1.1375-02
9	0.2500	1.0326	: 1.3110-02	1.7470-02	-1.3100-02
10	0.2794	1.0422	: 1.3655-02	1.7470-02	-1.3100-02
11	0.3018	1.0529	: 1.3670-02	2.1490-04	-1.7370-04
12	0.3312	1.0650	: 1.9135-02	2.5050-02	-1.8050-02
13	0.3616	1.0787	: 2.1350-02	2.7523-04	-2.0800-04
14	0.3911	1.0841	: 2.2770-02	3.0060-04	-2.2570-04
15	0.4205	1.1113	: 2.3565-02	3.2200-04	-2.4770-04
16	0.4509	1.1305	: 2.4220-02	3.3835-02	-2.6610-02
17	0.4852	1.1521	: 2.5710-02	3.9150-02	-2.9150-02
18	0.5147	1.1762	: 3.3345-02	4.3615-02	-3.0707-02
19	0.5441	1.2021	: 3.5140-02	4.7245-02	-3.5370-02
20	0.5735	1.2333	: 3.8105-02	5.1555-02	-3.8560-02
21	0.6039	1.2672	: 4.1700-02	5.6020-02	-4.0910-02
22	0.6324	1.3055	: 4.1800-02	5.6300-02	-4.0570-02
23	0.6618	1.3487	: 4.1950-02	5.6410-02	-3.8350-02
24	0.6912	1.3753	: 5.3575-02	5.1985-02	-5.2110-02
25	0.7206	1.4539	: 5.2565-02	5.2026-02	-5.4700-02
26	0.7500	1.5184	: 5.4825-02	5.6150-02	-5.3010-02
27	0.7794	1.5932	: 5.1170-02	5.8390-02	-5.8660-02
28	0.8088	1.6809	: 5.9110-02	4.7000-02	-4.8010-02
29	0.8382	1.7843	: 6.6350-02	3.7170-02	-2.2490-02
30	0.8676	1.8066	: 6.1360-02	4.0550-02	-4.4370-02
31	0.8971	2.0601	: 6.1580-02	6.5910-03	-2.3450-03
32	0.9265	2.2465	: 6.1010-02	7.3760-03	-3.1100-03
33	0.9559	2.4686	: 5.9450-02	1.5090-02	-1.5090-02
34	0.9853	2.8038	: 5.6150-02	7.2190-02	-2.5560-02

1. CS : RWVAL, RW1, RW2, RW3, RW4, RW5 : RVAL, RK1, RK2, RK3 : RMAX = 0.6450-05 : RMMAX = 1.3320-05

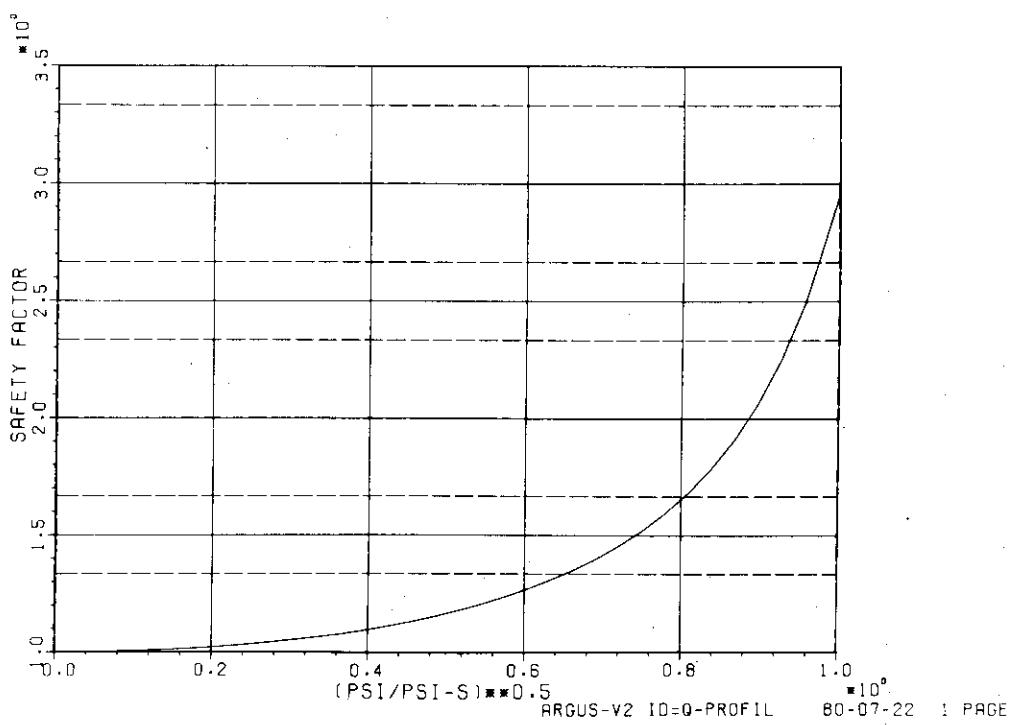
1	C.0147	-74.7660-04	-1.1500-04	-1.7950-05	-4.6930-04	-1.9340-07	1.2160-04	: 1.1850-05	4.2710-06	2.9510-06	4.6310-06
2	0.0441	-1.0550-03	-3.3500-04	-5.4200-05	-1.6320-03	-7.4160-07	3.6750-04	: 3.5640-05	1.3360-05	6.7450-05	1.5450-05
3	0.0735	-7.5190-03	-8.2280-04	-5.3600-05	-2.5100-03	-2.1460-06	6.6280-04	: 7.6410-05	2.3870-05	2.0160-05	3.1620-05
4	0.1124	-3.7040-03	-1.2120-03	-1.6540-04	-3.4870-04	-1.3190-03	1.44230-04	: 3.9330-05	3.9330-05	5.9560-05	5.9560-05
5	0.1324	-5.0450-03	-1.4500-03	-1.4500-03	-5.0910-03	-1.7130-05	1.5990-03	: 2.6460-04	7.7220-05	1.1100-04	1.1100-04
6	0.1618	-6.5110-03	-2.3100-03	-2.4180-04	-8.4280-03	-4.0140-05	3.5880-03	: 4.8610-04	1.3950-04	1.6520-04	2.0140-04
7	0.1912	-8.3120-03	-1.6340-03	-2.3910-04	-8.2010-02	-2.2010-03	8.5590-03	: 2.4810-04	2.6190-04	3.5000-04	3.5000-04
8	0.2266	-1.0320-02	-2.1900-03	-2.9170-04	-1.9780-02	-1.5110-04	1.4460-03	: 4.2530-04	5.7750-04	5.7750-04	5.7750-04
9	0.2500	-1.6760-04	-2.8500-03	-4.9590-04	-2.7710-02	-2.6670-04	1.4610-02	: 2.1120-02	6.9610-04	7.3350-04	9.0200-04
10	0.2794	-5.5710-04	-6.3020-04	-4.0050-04	-4.2110-04	-4.2110-04	3.4950-03	: 1.0860-03	1.0160-03	1.3330-03	1.3330-03
11	0.3088	-1.0150-02	-4.0220-03	-6.0300-04	-5.0250-02	-6.5950-04	4.4286-02	: 1.6170-03	1.8860-03	2.1060-03	2.1060-03
12	0.3382	-2.2370-02	-6.0500-03	-1.0240-03	-7.6470-02	-9.6610-04	6.2440-02	: 6.9860-03	2.3120-03	2.3120-03	2.5660-03
13	0.3676	-2.6100-02	-8.0520-03	-1.0050-03	-1.3990-02	-1.3990-03	8.3280-03	: 3.1870-03	2.1870-03	2.3770-03	2.3770-03
14	0.3971	-2.2270-02	-1.0960-02	-1.0330-02	-1.2610-02	-1.0030-03	1.1240-01	: 1.2130-02	4.2640-04	4.3220-03	4.3220-03
15	0.4265	-3.0990-05	-1.4920-02	-2.0250-03	-1.3830-01	-2.6530-03	1.4710-01	: 1.5560-02	5.5790-03	4.3740-03	5.6120-03
16	0.4559	-2.6160-02	-2.0200-02	-2.0880-03	-1.8110-01	-4.0900-03	1.8320-01	: 1.9870-02	7.2010-03	5.3880-03	7.2010-03
17	0.4853	-7.4140-02	-6.6380-02	-3.0390-03	-2.2750-01	-5.7140-03	2.3960-01	: 2.5450-02	9.2470-03	6.9800-03	9.5630-03
18	0.5147	-1.3220-02	-3.8840-02	-3.7560-03	-2.0700-01	-7.9220-03	3.0360-01	: 3.3070-02	1.1910-02	8.4130-02	1.2150-02
19	0.5441	-1.2500-02	-4.3160-02	-4.1980-03	-3.7260-01	-1.0860-02	3.0660-01	: 4.3310-02	1.5440-02	1.0820-02	1.0500-02
20	0.5735	-1.4800-02	-5.2960-02	-6.4230-03	-4.0350-01	-1.1440-02	4.9180-01	: 5.6480-02	2.0160-02	1.4010-02	2.2320-02
21	0.6039	-2.0540-02	-6.2650-02	-6.9150-03	-5.1360-01	-1.7340-02	6.2280-01	: 1.5560-02	2.6230-02	1.7420-02	2.7420-02
22	0.6324	-1.3650-02	-7.6330-02	-1.2240-02	-6.5100-01	-1.9490-02	7.7110-01	: 6.4340-02	3.3400-02	3.0270-02	3.0270-02
23	0.6618	-1.4180-02	-9.3510-02	-1.5700-02	-7.8030-01	-2.0540-02	9.2300-01	: 9.2450-02	4.0840-02	3.0050-02	3.0050-02
24	0.6912	-6.4950-02	-1.2400-01	-1.7640-02	-8.3390-01	-2.2580-02	1.0730-00	: 9.8780-02	2.0230-02	3.0110-02	3.0110-02
25	0.7206	-1.6210-01	-1.6440-01	-1.0900-01	-8.4030-01	-2.7550-02	1.2220+00	: 1.1240-01	5.5120-02	1.8860-02	3.0420-02
26	0.7500	3.0270-01	-2.0150-01	-7.6075-02	-8.8020-01	-3.2550-02	1.4400+00	: 1.3910-01	6.5740-02	2.0360-02	5.2980-02

27	6.7754	: 3.0390-01	-2.2730-01	-2.7500-02	-1.1030-00	-0.0210-02	1.7020+00	: 1.6480-01	6.1470-02	2.4090-02	5.9280-02
28	6.8085	: 2.8770-01	-2.7160-01	-2.4670-02	-1.3350-00	-3.4210-02	1.9630+00	: 1.6120-01	9.6700-02	2.0930-02	4.3560-02
29	0.8382	: 6.2760-01	-5.6190-01	-2.1090-02	-1.1550-00	-2.3660-02	2.2090+00	: 1.8420-01	1.0800-01	1.3870-02	6.2320-02
30	0.8676	: 6.9810-01	-5.1260-01	-2.7500-02	-1.3130-00	-3.8660-02	2.5000+00	: 1.3030-01	1.3030-01	1.4640-02	7.7890-02
31	0.8971	: 7.1480-01	-5.0150-01	-3.9230-02	-1.4530-00	-2.1410-02	2.7300+00	: 2.1560-01	1.4730-01	7.8030-03	6.0510-02
32	0.9265	: 8.1140-01	-5.6740-01	-4.1200-02	-1.4610-00	-2.5960-02	2.9040+00	: 2.8980-01	1.7660-01	6.8560-03	1.0630-01
33	0.9659	: 1.0000+00	-6.6790-01	-2.9550-02	-1.1870-00	-1.1280-02	2.1500+00	: 5.5530-01	2.1580-01	6.7670-03	3.3270-01
34	0.9853	: -7.1650-02	-8.8280-01	-2.5930-02	-1.5090-00	-3.9040-03	2.3500+00	: 1.0000+00	2.8670-01	1.1710-02	7.0160-01

J, CHI : XVAL, XMAX = 5.4660-02 : MK = 33 XKMAX = 5.006D-02

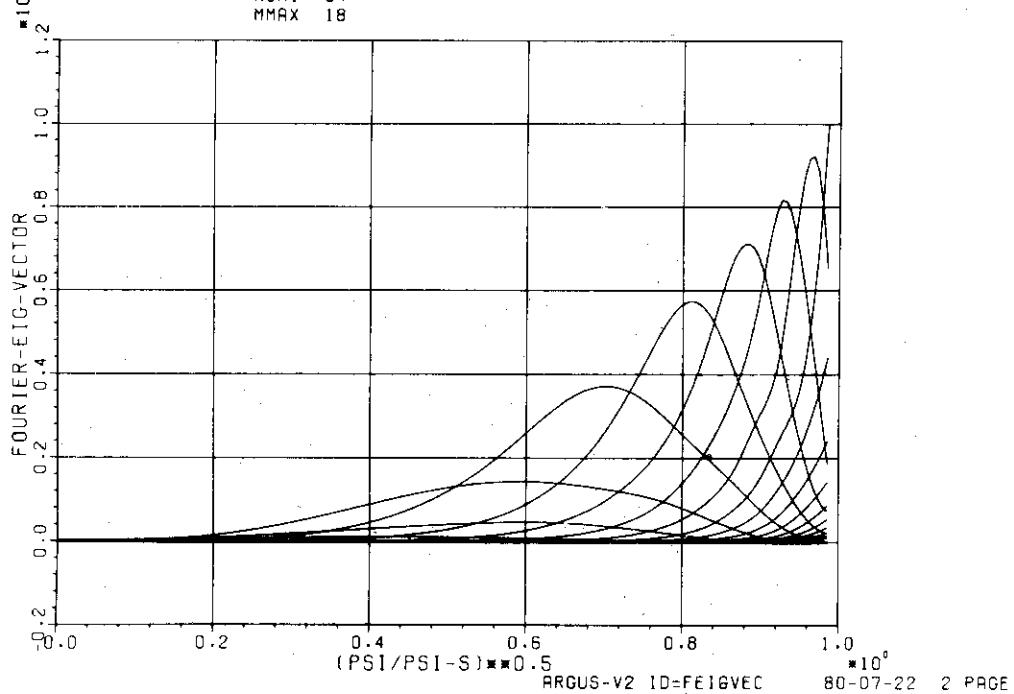
1	0.0	: 1.6000+00	1.0000+00	1.0000+00	1.0000+00	1.0000+00	1.0000+00	1.0000+00	1.0000+00	1.0000+00	1.0000+00
2	0.0203	: 6.0770-01	6.4420-01	6.4420-01	6.4420-01	6.4420-01	6.4420-01	6.4420-01	6.4420-01	6.4420-01	6.4420-01
3	0.0606	: -1.9710-01	-1.6490-02	-1.6490-02	-1.6490-02	-1.6490-02	-1.6490-02	-1.6490-02	-1.6490-02	-1.6490-02	-1.6490-02
4	0.0905	: -7.5100-01	-6.1670-01	-6.1670-01	-6.1670-01	-6.1670-01	-6.1670-01	-6.1670-01	-6.1670-01	-6.1670-01	-6.1670-01
5	0.1212	: -7.5970-01	-7.5970-01	-7.5970-01	-7.5970-01	-7.5970-01	-7.5970-01	-7.5970-01	-7.5970-01	-7.5970-01	-7.5970-01
6	0.1515	: -2.6560-01	-5.5250-01	-5.5250-01	-5.5250-01	-5.5250-01	-5.5250-01	-5.5250-01	-5.5250-01	-5.5250-01	-5.5250-01
7	0.1918	: 2.6810-01	-7.4060-02	-7.4060-02	-7.4060-02	-7.4060-02	-7.4060-02	-7.4060-02	-7.4060-02	-7.4060-02	-7.4060-02
8	0.2421	: 5.4440-01	3.3240-01	3.3240-01	3.3240-01	3.3240-01	3.3240-01	3.3240-01	3.3240-01	3.3240-01	3.3240-01
9	0.2624	: 4.5300-01	6.9980-01	6.9980-01	6.9980-01	6.9980-01	6.9980-01	6.9980-01	6.9980-01	6.9980-01	6.9980-01
10	0.2727	: 1.1710-01	4.0610-01	4.0610-01	4.0610-01	4.0610-01	4.0610-01	4.0610-01	4.0610-01	4.0610-01	4.0610-01
11	0.3030	: -2.2280-01	1.2220-01	1.2220-01	1.2220-01	1.2220-01	1.2220-01	1.2220-01	1.2220-01	1.2220-01	1.2220-01
12	0.3333	: -3.7210-01	-1.1140-01	-1.1140-01	-1.1140-01	-1.1140-01	-1.1140-01	-1.1140-01	-1.1140-01	-1.1140-01	-1.1140-01
13	0.3636	: -2.7770-01	-2.7770-01	-2.7770-01	-2.7770-01	-2.7770-01	-2.7770-01	-2.7770-01	-2.7770-01	-2.7770-01	-2.7770-01
14	0.3939	: -3.5560-02	-2.8170-01	-2.8170-01	-2.8170-01	-2.8170-01	-2.8170-01	-2.8170-01	-2.8170-01	-2.8170-01	-2.8170-01
15	0.4242	: 1.4880-01	-1.5970-01	-1.5970-01	-1.5970-01	-1.5970-01	-1.5970-01	-1.5970-01	-1.5970-01	-1.5970-01	-1.5970-01
16	0.4545	: 2.6410-01	1.2730-02	1.2730-02	1.2730-02	1.2730-02	1.2730-02	1.2730-02	1.2730-02	1.2730-02	1.2730-02
17	0.4848	: 1.7000-01	1.4500-01	1.4500-01	1.4500-01	1.4500-01	1.4500-01	1.4500-01	1.4500-01	1.4500-01	1.4500-01
18	0.5152	: -1.2370-02	1.9310-01	1.9310-01	1.9310-01	1.9310-01	1.9310-01	1.9310-01	1.9310-01	1.9310-01	1.9310-01
19	0.5455	: -1.6020-01	1.3940-01	1.3940-01	1.3940-01	1.3940-01	1.3940-01	1.3940-01	1.3940-01	1.3940-01	1.3940-01
20	0.5758	: -1.8860-01	2.7750-02	2.7750-02	2.7750-02	2.7750-02	2.7750-02	2.7750-02	2.7750-02	2.7750-02	2.7750-02
21	0.6061	: -9.4030-02	-7.9380-02	-7.9380-02	-7.9380-02	-7.9380-02	-7.9380-02	-7.9380-02	-7.9380-02	-7.9380-02	-7.9380-02
22	0.6364	: 3.9220-02	-1.3210-01	-1.3210-01	-1.3210-01	-1.3210-01	-1.3210-01	-1.3210-01	-1.3210-01	-1.3210-01	-1.3210-01
23	0.6667	: 1.6667	1.3160-01	-1.1120-01	-1.1120-01	-1.1120-01	-1.1120-01	-1.1120-01	-1.1120-01	-1.1120-01	-1.1120-01
24	0.6970	: 1.2760-01	-1.2760-01	-1.2760-01	-1.2760-01	-1.2760-01	-1.2760-01	-1.2760-01	-1.2760-01	-1.2760-01	-1.2760-01
25	0.7273	: 4.4340-02	4.4440-02	4.4440-02	4.4440-02	4.4440-02	4.4440-02	4.4440-02	4.4440-02	4.4440-02	4.4440-02
26	0.7576	: -5.4030-02	5.1000-02	5.1000-02	5.1000-02	5.1000-02	5.1000-02	5.1000-02	5.1000-02	5.1000-02	5.1000-02
27	0.7879	: -1.0140-01	8.2030-02	8.2030-02	8.2030-02	8.2030-02	8.2030-02	8.2030-02	8.2030-02	8.2030-02	8.2030-02
28	0.8162	: -7.4070-02	2.9390-02	2.9390-02	2.9390-02	2.9390-02	2.9390-02	2.9390-02	2.9390-02	2.9390-02	2.9390-02
29	0.8465	: -8.8250-04	-3.2260-02	-3.2260-02	-3.2260-02	-3.2260-02	-3.2260-02	-3.2260-02	-3.2260-02	-3.2260-02	-3.2260-02
30	0.8788	: 6.1430-02	-6.6500-02	-6.6500-02	-6.6500-02	-6.6500-02	-6.6500-02	-6.6500-02	-6.6500-02	-6.6500-02	-6.6500-02
31	0.9091	: 6.9970-02	-5.5930-02	-5.5930-02	-5.5930-02	-5.5930-02	-5.5930-02	-5.5930-02	-5.5930-02	-5.5930-02	-5.5930-02
32	0.9394	: 2.4310-02	-1.0990-02	-1.0990-02	-1.0990-02	-1.0990-02	-1.0990-02	-1.0990-02	-1.0990-02	-1.0990-02	-1.0990-02
33	0.9697	: -3.6400-02	3.7700-02	3.7700-02	3.7700-02	3.7700-02	3.7700-02	3.7700-02	3.7700-02	3.7700-02	3.7700-02
34	1.0000	: -6.3750-02	5.8380-02	5.8380-02	5.8380-02	5.8380-02	5.8380-02	5.8380-02	5.8380-02	5.8380-02	5.8380-02

NO = 321

(a) Profile of safety factor versus $\sqrt{\psi}$.

NO = 321

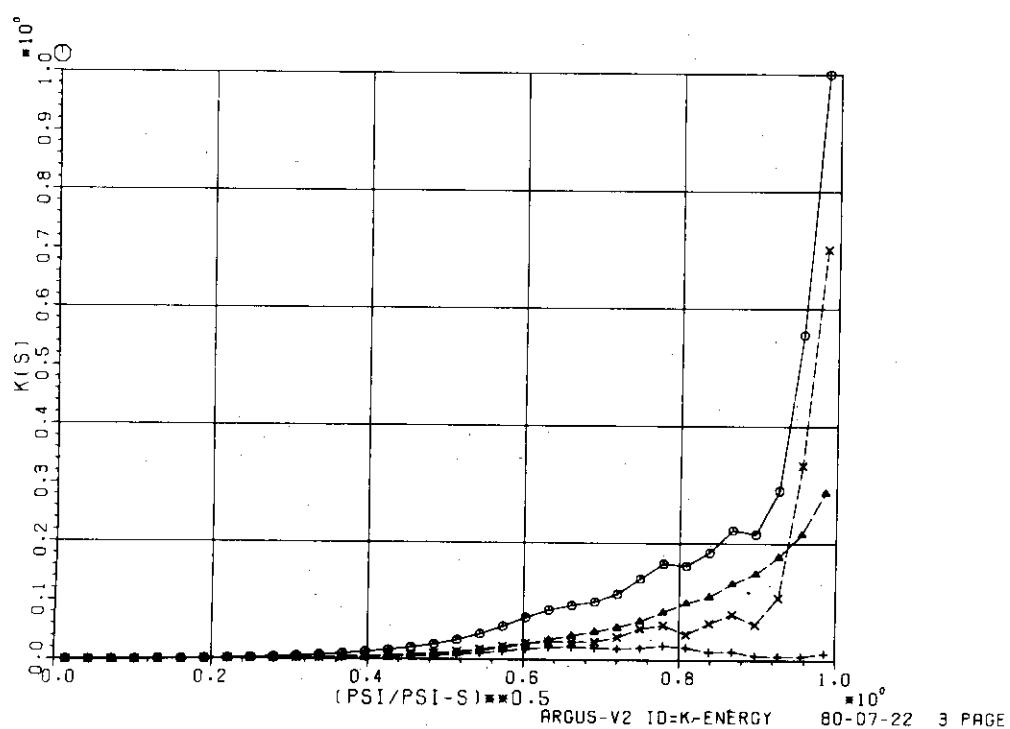
N	3
NPSI	34
NCHI	34
MMAX	18



(b) Mode structure of the displacement.

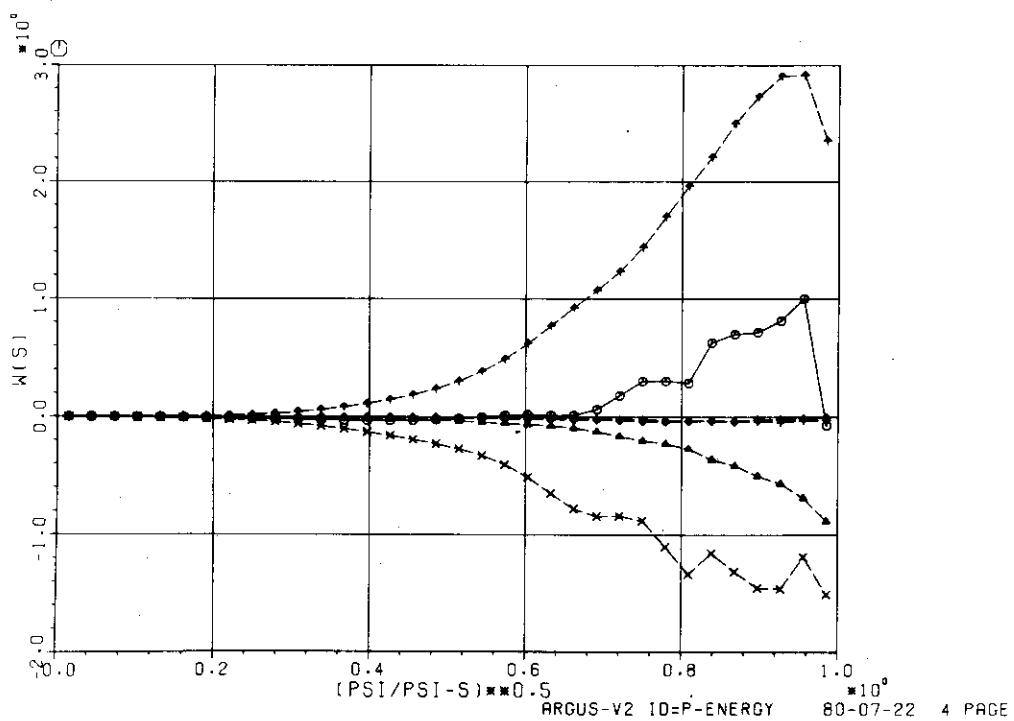
Fig.3-22 Example⁴⁾ of output graphs by EMTDUMP2.

NO = 321



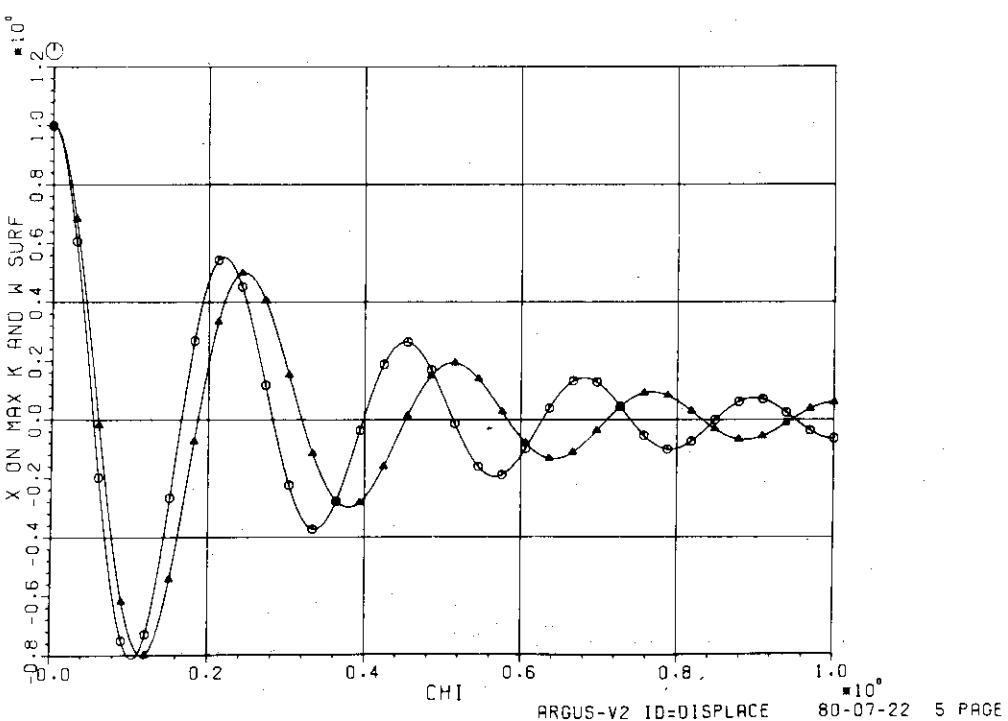
(c) Profile of the kinetic energy.

NO = 321



(d) Profile of the potential energy.

NO = 321



(e) Displacement along the poloidal direction.

4. ま と め

トーラス・プラズマの線形理想的 MHD 不安定性解析コード ERATO を使用する上で必要なデータ、計算の手順等をまとめると共に、著者らによって、現在までに ERATO に加えられた改良点等について記した。改良等が加えられたのは、主として出力データの処理に関する部分で、この改良の結果、ERATO の計算とそのデータの整理等が著しく容易になった。

ERATO コードは、トカマク・プラズマの MHD 不安定性の解析を行う上できわめて有用であるが、一連の計算を行って物理的に有意なデータを出す為には膨大な量の計算を必要とするのみならず、その個々の計算を実行するに際しても、場合によっては、非常に大きな計算機システム資源を要するので、現在の計算機事情のもとではその実行は必ずしも容易ではない。この意味からも、ERATO コードには、まだいくつかの改良、修正を要する部分が残っており、また、コードの拡張が望まれる部分もある。これらの点は、今後、種々のパラメータを変えて大きな計算を実行していく過程でその整備を続けていくことになる。特に、本報告書で対象とした改良か所は ERATO 5 に関する部分に限られているが、下に示すような ERATO 1, ERATO 2, ERATO 3, ERATO 4 に関する改良、拡張等の整備作業は、現在、検討中あるいは実行中である。

- * ERATO 1 : メトリック量の計算方式を整理・統一して、平衡コードの R-Z メッシュから安定性コードの $\psi-\chi$ メッシュへ諸量を変換する時に生ずる問題点を解消する。例えば、平衡コードとして $\psi-\chi$ メッシュを基本に考えた SELENE¹²⁾ の採用も考えられる。
- * ERATO 2 : 真空エネルギーの計算精度を高める。n = 0 変位に対する評価部分を完成し、水平方向の位置不安定性解析にも使えるようにする。ポロイダル・コイル・ケーシングのシェル効果を導入したコードを完成する。
- * ERATO 3 : ERATO 4 の性能を高める為に必要な変更を加える。行列の組み替えも考えられる¹³⁾。
- * ERATO 4 : 新アルゴリズムの採用で、高速化および主メモリ、補助メモリの節約等をはかる。

これらの本質的改良整備と平行して、ERATO ジョブの投入を容易にかつ誤りなく行えるようになる為のジョブ投入用プログラムの開発も進めている。これらの成果は、今後の報告書で順次発表される予定である。

謝 辞

MHD 安定性解析コード ERATO は、スイス・ローザンヌ工科大学の R. Gruber 氏らによって開発され、同氏らの御好意により原研に導入されたものである。同氏らには、ERATO 導入後も数々の助言や議論を通じて大変お世話になった。こゝにあらためて感謝の意を表したい。田中正俊核融合研究部次長および理論解析研究室の津田孝氏、伊藤公孝氏には、ERATO に関する研究全般にわたって有益な助言をいただくと共に、数々の討論にくわわっていただいた。小幡行雄核融合研究部長には、終始絶えざる御激励をいただいた。これらについて、こゝに深く感謝の意を表したい。原研計算センターの斎藤直之氏には、計算機システム使用の上の有益な助言と御配慮をいただいた。計算センターの皆様の御協力で大きなシステム資源を必要とする ERATO 計算が順調に行え本報告書をまとめ上げる事ができたことを深く感謝したい。また、田子精男富士通（株）第 2 原子力システム課長には、ERATO コード整備の過程での問題点解決に関して示された御理解と御助言について感謝したい。

References

- 1) Berger, D., Gruber, R., and Troyon, F., "A two dimensional MHD stability code with a finite hybrid expansion in both directions", Proceedings of the 2nd European Conference on Computational Physics, April 27-30, 1976 IPP 6/147 C3.
- 2) Tsunematsu, T., et al., "Beta limit of a large tokamak with a circular cross-section", submitted to Plasma Physics.
- 3) Takeda, T., et al., "Stability limits on β ", IAEA INTOR Workshop Report, Group 11, JAERI-M 8624 (January 1980) p.13.
- 4) Azumi, M., et al., "Evolution of stable high beta tokamak equilibria", 8th International Conference on Plasma Physics and Controlled Nuclear Fusion Research, Brussels 1980, IAEA-CN-38/K-1-1.
- 5) Tsunematsu, T., et al., "Convergence of solutions of the MHD stability code ERATO", to be published in Computer Physics Communications.
- 6) Tsunematsu, T., et al., "Stability analysis by ERATO code", JAERI-M 8616 (December 1979).
- 7) Gruber, R., et al., "ERATO stability code", to be published in Computer Physics Communications.
- 8) Gruber, R., "HYMNIA - band matrix package for solving eigenvalue problem", Computer Physics Communications 10 (1975) 30.
- 9) Kurita, G., et al., "Preprocessor EOS", in preparation.
- 10) Azumi, M., et al., "A fluid model numerical code system for tokamak fusion research", Fourth International Symposium on Computing Methods in Applied Sciences and Engineering, Versailles, December 1979.
- 11) Takeda, T., et al., "Graphic I/O subsystem ARGUS for fluid codes". in preparation.
- 12) Takeda, T. and Tsunematsu, T., "A numerical code SELENE to calculate axisymmetric toroidal MHD equilibria", JAERI-M 8042 (January 1979).
- 13) Scott, D.S., "implementing sparse matrix techniques in the ERATO code", ORNL/CSD/TM-117 (April 1980).