JAERI-M
8166

冷却材喪失事故解析コードRELAP4／MOD5及びMOD6のFACOM230／75システムへの変換整備

1979年3月

鴻坂　厚夫・石谷　隆広*・熊倉　利昌*・奈良岡賢逸*

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

冷却材喪失事故解析コード RELAP4／MOD5 及び MOD6 の FACOM230／75システムへの変換整備

日本原子力研究所東海研究所安全解析部
鴻坂 厚夫・石谷 隆広*・熊倉 利昌*・奈良岡賢逸*

（1979年2月13日受理）

冷却材喪失事故解析コードとして広く用いられている RELAP－4 コードは，原型版であるMOD2 にはじまり， MOD3，MOD5，MOD6 へと発展してきた。物理モデルの改良と詳細化が進み，取り扱える現象の範囲と程度が広がってきている。それに伴い，プログラムの規模も巨大化し，計算機の記憶容量の制限内に収めるために多くの工夫がなされており，そのことはコードのレベルアップとともに増加している。本報告では，主としてMOD5 に組込まれた Dynamic Storage Allocation 機能やMOD6 で登場した PRELOAD プリプロセッサーの概要と，FACOM への変換整備について記述し，テストランの結果についても触れる。

*）外来研究員：富士通

JAERI-M 8166

Implementation of the Thermal-Hydraulic Transient Analysis Code RELAP4/MOD5 and MOD6 on the FACOM 230/75 Computer System

Atsuo KOHSAKA, Takahiro ISHIGAI*,
Toshimasa KUMAKURA* and Ken-itsu NARAOKA*

Division of Reactor Safety Evaluation
Tokai Research Establishment, JAERI

(Received February 13, 1979)

Development efforts have continued on the extensively used LOCA analysis code RELAP-4, as seen in its history; that is, from the prototype version MOD2 to the latest one MOD6 which is capable of one-through calculations from blowdown to reflood phase of PWR-LOCA. Many improvements and refinements of the models have enlarged the scopes and extents of phenomena to treat. Correspondingly the size of program has increased version to version, and special programing techniques have continuously been introduced to manage the program within limited capacity of core memory. For example, the Dyanmic Storage Allocation of MOD5 and the PRELOAD Preprocessor newly incorporated in MOD6 are those designed for the CDC computer with relatively small core size. Described are these programing techniques in detail and experiences on implementation of the codes on FACOM 230/75, together with some results of confirmatory calculations.

Keywords: RELAP-4 Code, LOCA Analysis, Code Implementation, Dynamic Storage Allocation, PRELOAD Preprocessor

---

* Visiting Engineer, FUJITSU

目　　次

Contents

1. 序

冷却材喪失事故解析コードとして広く用いられているRELAP－4コードは，１９７４年にはじめてわが国に導入されたRELAP－4／MOD2（BE計算原研版）にはじまり，MOD3（WREM，EM計算），MOD5（BE計算改良），そして，MOD6（PWRブローダウン再冠水一貫計算）へと発展してきた。物理モデルの改良と詳細化が進み，取り扱える現象の範囲と程度が大きく拡大し，それに伴いプログラムの規模は巨大化の一途をたどっている。一方，プログラミングの観点からみても，機能が追加拡大されてきている。すなわち，MOD2に組み込まれたFree Format Inputルーチン，MOD5から付け加ったDynamic Allocation機能，そしてMOD6になって登場したPRELOAD（プリプロセッサー），といった具合である。このような機能の追加は，コードが書かれた計算機の変遷と密接な関係があるが，コード変換作業の困難さは増す一方である。したがって，RELAP－4シリーズの変換整備については，コードのバージョンアップの各段階での機能を十分理解し，それぞれのオリジナル機能と形態を極力損わないように模擬して整備し，以後のコードの整備に供するようにしておくことが肝要である。開発に用いられた計算機種は，MOD2とMOD3がIBM 360，MOD5とMOD6がCDC 7600であり，Dynamic Storage AllocationやPRELOAD機能は，いずれも，CDC 7600が持つコアサイズの制限内に巨大なコードを収めるべく工案されたものである。特に，MOD6のDynamic Storage Allocationでは，ボリュームやジャンクションに関する変数も作業領域上に割り付けられるようになっており，その構造もMOD5とは異った新しいものである。

MOD5とMOD6における物理モデルの詳細とモデルの相異及びインプットデータの説明などについては，それぞれのコードマニュアル[(1),(2)]に譲り，本報告書ではもっぱらコードの変換整備上の問題点や各種機能の詳細について記述する。（但し，Free Format Inputルーチンについては，文献(3)で記述してあるので本報告書では触れない。）なお，サンプルデータによるテストランの結果についても簡単に触れる。

2. RELAP－4／MOD5の変換整備

2.1 変換概要

Original material は，１９７７年4月にNEA Data Bank（前Computer Program Library）より入手したもので，主として次のものから成っている。

RELAP－4本体とプロッタープログラム
INEL Environmental Subroutine Library
Free Format Input Routines
Dynamic Storage Allocation Routines
OVERLAY カード
サンプルインプット（８ケース）

RELAP－4本体は，FORTRAN カード約４５０００枚からなる巨大なプログラムである。MOD5になって本体がこのように大きくなったのは，新しい計算モデルやオプションの追加の他に，INEL（Idaho Nuclear Engineering Laboratory）が，新たにＩＢＭバーションとＣＤＣバーションを混在させたファイルを提供し，各ユーザは必要なもの以外をコメントカードにして使用するという配布体制をとったことによる。この配布体制は現在も続いている。

本体は巨大であるがプログラムそのものは標準FORTRAN で書かれており，全体として特筆すべき変換上の問題点はなかった。プロッタープログラムは，ＭＯＤ３(WREM)のそれと機能的に同一のものであるが，プロットできる物理量の種類が増えている。変換では特に問題を生じなかった。

2.1.1 Free Format Inputルーチン

ＭＯＤ２を整備した際に作成したルーチンが，そのまま使用可能であったが，新たにINP９を作成する必要があった。これは，既にサブルーチンLINK などを用いて読み込まれた処理済みのデータを消去し，作業領域を開放するために用いられる。Environmental Subroutine Libraryに含まれていた同ルーチンのＩＢＭバーションを，FACOM用に修正した。

2.2 Dynamic Storage Allocation 機能

2.2.1 概　要

新たにＭＯＤ５に広範に組み込まれた機能である。元来この機能は，コアが比較的小さいＣＤＣにおいて，LCM(Large Core Memory)やディスクなどの副記憶装置を，実行時にコアのように使用するべく用意されたものである。ＩＮＥＬでは，これとＩＢＭ用にも変換し，ＭＯＤ５のＩＢＭバーションに用いている。約１０個のルーチンからなり，ＦＴＢパッケージ

と呼ばれる。基本ルーチンは，主として次のものからなっている。

INITAL	DSCRIB	LCONTG	LOCF
GETCOR	IDFIND	DELETE	
RESERV	LOCATE	SHIFT	

RELAP−4／MOD5では，実行開始時に主記憶（Fast Core）上に大きな作業領域を確保し，その上に必要に応じて種々のマトリックスやベクトルを割り付けたり消去したりして，コアを有効利用できるようになっている。以下に，その機能の概要を示す。

可変コアサイズ機能による作業領域の確保

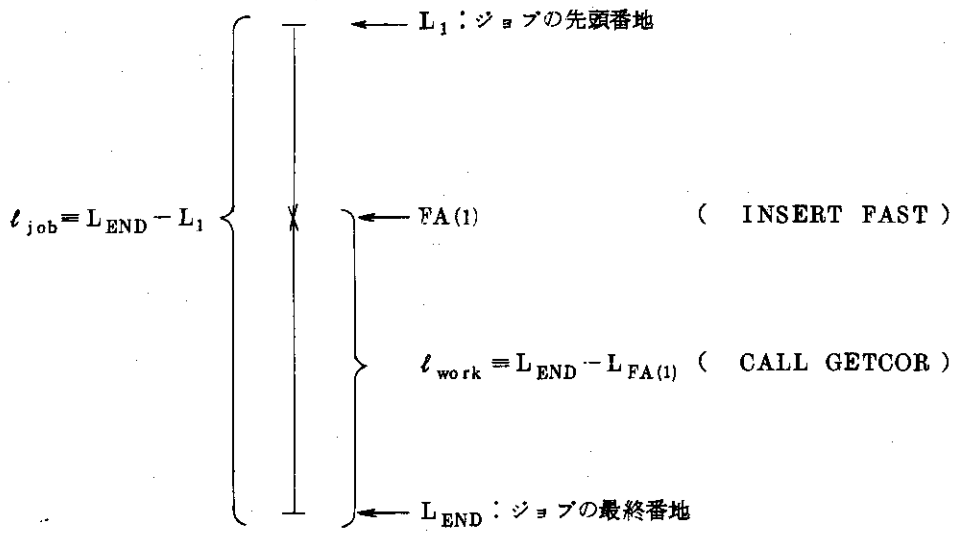
ジョブ制御カードで要求した，ジョブを実行するのに必要なコアの大きさを，実行開始時に認識し，そのうち，プログラム中に指定されたポイントから，ジョブの領域の最終番地までを作業領域として確保する。IBMでは，

COMMON／FAST／FA(1)

なるステートメントにより，FA(1)をマークポイントとして使用し，そのプログラム上の場所をLinkage Editorのコントロールステートメント

INSERT FAST

によって指定する。すなわち，FA(1)からジョブの最終番地までが作業領域となるわけであり，サブルーチンGETCORを呼ぶことにより，その長さを知ることができるようになっている。以上のことを2.1図に示す。



2.2.2 “ファイル”と基本ルーチン

(1) 作業領域でのファイルの割り付けと消去

実行時に，必要となった各種データについての領域を割り付けることができるが，通常この

領域は1次元のアレイとして扱われ，ファイルと呼ばれる。割り付けられたファイルの名前，長さ，作業領域上の位置などの情報は，通常作業領域の最後尾に特別に定義されるFile Descriptionファイルに書き込まれる。これは，２００個の倍精度語からなるファイルで，1つのデータファイルに対して4個の倍精度語が使用される。ファイル名としては，

±n.D0　（例えば，1.D0，2.D0，−3.D0　など）

なる倍精度実数値そのものを用いるものと約束されている。File Descriptionファイルのフォーマットの一般形及びファイルを割り付けたり消去したりした時の具体的な内容を付録1に示す。

(2) 基本ルーチン（FTBパッケージ）

FTBパッケージを構成する基本ルーチンの，それぞれのオリジナル機能の概略を述べる。

(i) INITAL

GETCORを呼ぶことにより，ポイントFA(1)からジョブの最終番地までを作業領域として定義し，その長さ（図2.1ではℓ_{work}）などの情報からFile Descriptionファイルを初期設定する。（付録1.2／step 1）

(ii) RESERV

実行中に必要となったデータ領域を，ID=±n.D0という名前のファイルとして作業領域上に割り付ける。このため，DSCRIB，IDFIND，LOCATE が呼ばれる。（付録1.2／step 2）

(iii) DSCRIB

ファイル名，ファイルの長さなどの情報をFile Descriptionファイルに書き込み，ファイルを定義する。この時，そのファイルが具体的に作業領域上のどこに位置するかは未定である。（付録1.2／step 2.1）

(iv) IDFIND

File Descriptionファイルの中から，指定したファイル名IDを見つけ出し，そのIDが書かれている番地を返す。（付録1.2／step 2.2）

(v) LOCATE

作業領域中の空き領域上に，ファイルの割り付けを行い，その開始位置の情報を返す。（付録1.2／step 2.3）

(vi) LCONTG

作業領域に断片的に存在する空き領域の中，最長の連続領域の開始位置の情報を返す。

(vii) DELETE

不必要になったファイルをFile Descriptionファイルから消去する。この時，対応する作業領域上のスペースは開放され，空き領域となる。（付録1.2／step 4）

(viii) SHIFT（MOD 6ではFTBSFTに名前が変えられている）

割り付けたファイルに実際にデータを書き込んだ場合，予定したより少ない領域で済むのが通例である。この時，そのファイルを実際の長さに短縮し，かつ，作業領域上の空き領域の前方，または，後方に可能な限りシフトする。したがって，そのファイルの先頭番地に変化が無ければ，ファイルの短縮のみを行うことになる。

(IX) LOCF

実行中に，FORTRAN変数の番地を知るためのルーチン（IBMではバイト単位で数えられるがFACOMではワード単位で数えられる。）

2.2.3 オーバーレイ構造と“ファイル”の割り付け

オーバーレイ構造は，Dynamic Storage Allocation機能と密接に関係している。すなわち，図2.2に示す如く，5つのマークポイントFA(1)，DEFINV，DEFTRV，DEFL1V，DEFL2Vが，

```
COMMON/FAST/FA(1)
    〃    /DEFINC/DEFINV
    〃    /DEFTRC/DEFTRV   } FORTRANステートメント
    〃    /DEFL1C/DEFL1V
    〃    /DEFL2C/DEFL2V
```

および，

```
INSERT FAST
    〃    DEFINC
    〃    DEFTRC   } Linkage Editorのコントロールステートメント
    〃    DEFL1C
    〃    DEFL2C
```

により定義される。それらを基準にして，FA(1)からL_{END}に至る作業領域を，不可侵領域（プログラム領域）と実作業領域とに各実行段階毎に（オーバーレイの主たるツリー毎に）区別し，それぞれをファイルとして定義する。そして実作業領域が，さらにいくつかのデータファイルとして分割されていく仕組みになっている。

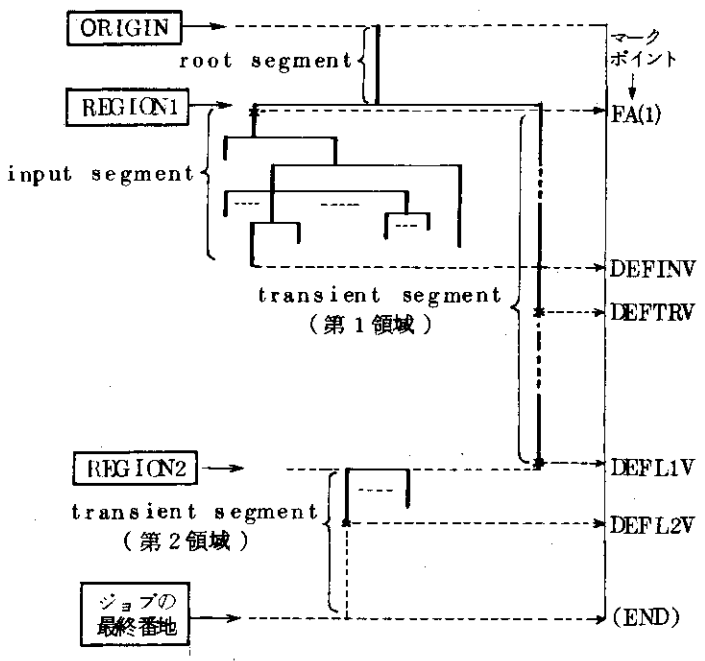


図2.2 オーバレイ構造とマークポイント（MOD 5 IBMバージョン）

MOD5では，固定した内容のファイルとしては，蒸気表ファイル，アイスコンデンサーデータ・ファイル，フィルデータ・ファイルが考えられている。ファイルの割り付けにのみ着目した時の，各種データ・ファイルを定義するRELAP4本体のサブルーチンINPUT，INVOL，INFILL，INICEのフローチャートを付録2に示す。

次に，データファイルが具体的にどのように定義されるかを図2.3で示す。ただし，ファイルの定義は，ジョブに必要なコアサイズの指定の仕方及び実行段階によって異なるので，図2.3では，プログラム領域の最終番地（およそDEFL2Vの番地）がジョブの最終番地（L_{END}）に近い場合で（但し，２００語は必要），インプット処理段階を例にとった。

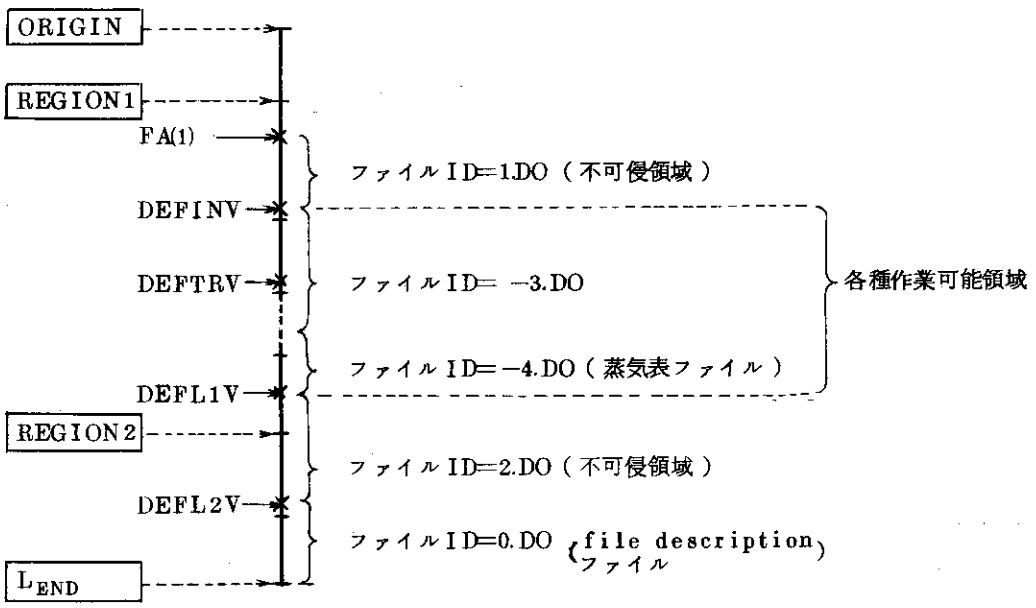


図2.3　ファイル割り付け例

上図において，L_{END}〜DEFL2Vを十分に大きくとった場合には，各種作業可能領域はID＝2.D0なるファイルによって2分される形となり，蒸気ファイルはL_{END}の方に定義される。

2.2.4　FACOMにおけるシミュレーション

MOD5では，Dynamic Storage Allocation 機能が，プログラム全体にわたって広範に組み込まれていること，また，MOD6，MOD7へと発展していく際にこの種の機能は増すことはあっても減ることはないであろうとの判断から，ほぼオリジナル機能と形態が保たれる形でシミュレーションを行った。

(1)　FTBパッケージ

FTBパッケージの基本ルーチンは，GETCORを除いてFORTRANで書かれており，詳しい調査の結果極く僅かな変更のみで使用可能であった。GETCORは，元来，ジョブ制御カードに指定したコアサイズで決まるジョブの最終番地と，

COMMON／FAST／FA(1)（FORTRAN ステートメント）
INSERT FAST　（Linkage Editor コントロールステートメント）

で決められるFA(1)の番地を，実行開始時に認識し，作業領域を規定するのに用いられ，アッ

センブラー言語で書かれている。FACOMでは，後述の如く，作業領域として固定した長さのものをCOMMONステートメントで確保する方法でシミュレーションした。したがってGET-COR としては，長さを単に定数値として定義する簡単なFORTRAN プログラムを作成した。付録3に，FACOM用FTB基本ルーチンのソースリストを示す。

(2) 可変コアサイズ機能

FACOMでは，EXECカードで指定した大きさの作業領域を実行時に確保する方法がないわけではない。しかし，そのためのサブルーチンをFASPで書く必要がある。さらに，このサブルーチンに必要なハードウェア上の機能が，FORTRAN-H の標準モードで作成した実行形式プログラムでは既に他の目的に利用されていることなどの理由から，この機能をはずし，固定した長さのCOMMON領域を作業領域として定義することとした。すなわち，

```
  COMMON/FAST/FA(1),DEFINV,DEFTRV,DEFL1V,
*                   DEFL2V,FAWORK(16900),FAEND
```

なるCOMMONステートメントにより，作業領域FAWORKを定義した。したがって，問題により，より多くのコアを必要とする時は，FAWORKのディメンジョンを変更する必要がある。

(3) オーバーレイ構造上のマークポイント

上記の方式を採用したため，オーバーレイ構造上のマークポイントのうち，DEFINV，DEFTRV，DEFL1V，DEFL2Vは無意味となったわけであるが，MOD5のプログラム論理を変えないために定義しておく必要がある。

3. RELAP/MOD 6.の変換整備

3.1 変換概要

Original material は1978年7月NEAデータバンクより入手したものであるが，新たにPRELOAD が付け加わった。

(1) RELAP-4 本体，Plotter progrom

(2) INEL Environmental Subroutine library

i) Free Format Input routines

ii) Dynamic Storage Allocation routines

(3) PRELOAD program & input

(4) Sample problem (8cases)

RELAP4本体はFORTRAN カードで約74,000枚からなりMOD5に比べて一層巨大なプログラムとなっている。INEL Environmental Subroutine LibraryはRELAP4/MOD5 と同一のルーティンであった（但し，SHIFTと言うサブルーティンのみがFTBSFTという名前に変更されていた）。又，Plotter program についても同一であったので特に新たに整備する必要がなかった。MOD6で新たに追加された機能としてプリプロセッサー

センブラー言語で書かれている。FACOMでは，後述の如く，作業領域として固定した長さのものをCOMMONステートメントで確保する方法でシミュレーションした。したがってGET-COR としては，長さを単に定数値として定義する簡単なFORTRAN プログラムを作成した。付録3に，FACOM用FTB基本ルーチンのソースリストを示す。

(2) 可変コアサイズ機能

FACOMでは，EXECカードで指定した大きさの作業領域を実行時に確保する方法がないわけではない。しかし，そのためのサブルーチンをFASPで書く必要がある。さらに，このサブルーチンに必要なハードウェア上の機能が，FORTRAN-H の標準モードで作成した実行形式プログラムでは既に他の目的に利用されていることなどの理由から，この機能をはずし，固定した長さのCOMMON領域を作業領域として定義することとした。すなわち，

```
  COMMON/FAST/FA(1),DEFINV,DEFTRV,DEFL1V,
*                   DEFL2V,FAWORK(16900),FAEND
```

なるCOMMONステートメントにより，作業領域FAWORKを定義した。したがって，問題により，より多くのコアを必要とする時は，FAWORKのディメンジョンを変更する必要がある。

(3) オーバーレイ構造上のマークポイント

上記の方式を採用したため，オーバーレイ構造上のマークポイントのうち，DEFINV，DEFTRV，DEFL1V，DEFL2Vは無意味となったわけであるが，MOD5のプログラム論理を変えないために定義しておく必要がある。

3. RELAP/MOD 6.の変換整備

3.1 変換概要

Original material は1978年7月NEAデータバンクより入手したものであるが，新たにPRELOAD が付け加わった。

(1) RELAP-4 本体，Plotter progrom

(2) INEL Environmental Subroutine library

i) Free Format Input routines

ii) Dynamic Storage Allocation routines

(3) PRELOAD program & input

(4) Sample problem (8cases)

RELAP4本体はFORTRAN カードで約74,000枚からなりMOD5に比べて一層巨大なプログラムとなっている。INEL Environmental Subroutine LibraryはRELAP4/MOD5 と同一のルーティンであった（但し，SHIFTと言うサブルーティンのみがFTBSFTという名前に変更されていた）。又，Plotter program についても同一であったので特に新たに整備する必要がなかった。MOD6で新たに追加された機能としてプリプロセッサー

PRELOADがある。これは64kwというCDC7600のコアサイズの制限の中で巨大なプログラムを納めるべく，計算に用いられないプログラムオプションに応じた一切の不要ルーティンを，リンケージの段階で取り除くために開発されたものである。この機能により最適化されたリンケージ制御カードが作成され，最小のコアサイズで実行可能となる。我々が入手したバージョンはIBM用であるので，結果としてIBM用リンケージ制御カードを作り出す。したがって我々はこの作成されたIBMリンケージ制御カードからFACOM230−75用を作り出すルーティンを作成した。このことにより今後，予想されるRELAP4の新しいバージョンを整備する際にはこのルーティンを用いれば良い。又，このルーティンを単独で利用することによりIBM版リンケージ制御カードをFACOM版に変換することもできる。PRELOADによりFACOM230−75版では280kwのコアサイズを必要とするものが，サンプルケース8ケース共210kw内外につまり256kw内に納った。変換に特に問題がなかった。

3.2 Dynamic Storage Allocation 機能

3.2.1 概　要

MOD6においてもMOD5と同様にDynamic Storage Allocation機能，すなわち，FTBパッケージが用いられている。

FTBにより制御される“ファイル”は，サブルーチン間で受け渡すかどうかにより，2種類に分けられる。1つは，“一時ファイル”であり，1つのサブルーチン内で発生，使用そして消去され，他のサブルーチンに渡すことのないものである。他の1つは，“登録ファイル”であり，他のサブルーチンでも必要なデータが入っているので，FTBとは別にファイル情報がCOMMON/CONTRL/に登録され，(注)複数のサブルーチンで利用されるものである。

MOD6において，この“登録ファイル”が増加している。例えば，MOD5では固定されたCOMMONにあったボリューム・データとジャンクション・データ，そして新しく加えられたリフラッド計算用のムービング・メッシュ，ヒート・スラブ・データ等のファイルが，加わっている。“登録ファイル”の一覧を表3.1に示す。

注．“登録ファイル”は，コントロール番号と呼ばれる各々に特定な番号を持ち，この番号をインデックスとして，以下のCOMMONにファイル情報を格納する。

COMMON/CONTRL/FILID(30), INDEX(30), FILSIZ(30)

ここで，FILID, INDEXおよびFILSIZは，それぞれ，ファイルID番号，作業領域FAでの先頭インデックスおよびファイルの長さを格納する。

例えば，コントロール番号iのファイルは，ID番号がFILID(i)で，FA上の

FA(INDEX(i))〜FA(INDEX(i)+FILSIZ(i)−1)

に対応している。

表3.1 “登録ファイル”

コントロール番号	ファイルの内容	MOD 5	MOD 6
1	INPUT DATA(INPUT PHASE)AND SCRATCH(TRAN PHASE)	○	○
2	ICE CONDENSER ARRAY	○	○
3	FILL DATA	○	○
4	WATER PROPERTY DATA	○	○
5	FLOW SMOOTHING STORAGE	○	○
6	HEAT EXCHANGER ARRAY	○	○
7	FILE 2 READ BUFFER	×	○
8	DATA DEFINING IMPLICIT EQUATIONS	○	○
9	MOVING MESH HEAT SLABS	×	○
10	VOLUME DATA	×	○
11	JUNCTION DATA	×	○
12	DELTW ARRAY FOR NIFTE	×	○
13	AREA TO HOLD INFORMATION FOR ADVANCEMENT RETRY	×	○

○：有，×：無

ＦＴＢパッケージは，基本的にはＭＯＤ５のものと同一であり，サブルーチンＳＨＩＦＴが，FTBSFTに改名された点のみが，ＭＯＤ５のものと異なる。このため，ＭＯＤ６の変換整備にあたって，ＦＴＢパッケージを変換することなく，ＭＯＤ５変換時に整備したものを一部修正して使用することができた。

ＭＯＤ６においては，ＦＴＢの上位ルーチンとしてDSSHFTが，追加されている。このDSSHFTは，“登録ファイル”全体を，作業領域内の前方または後方へ空領域をつめて移動し，作業領域を整理するものである。

オーバレイ構造は，ＭＯＤ５が，リージョン指定とマーク・ポイントによりプログラム領域を２分割していたのと異なり，ＭＯＤ６では，単一領域にまとめられている。

ボリューム・データとジャンクション・データは，ＭＯＤ６で“登録ファイル”にされたために，ＭＯＤ５とは構造が大幅に異なる。すなわち，ＭＯＤ５では，それぞれの変数に対して固定長の配列が割り当てられていたが，ＭＯＤ６では，各ボリューム（またはジャンクション）毎にデータがまとめられており，領域の使用に無駄のないようになっている。

3.2.2 オーバレイ構造と“ファイル”の割付け

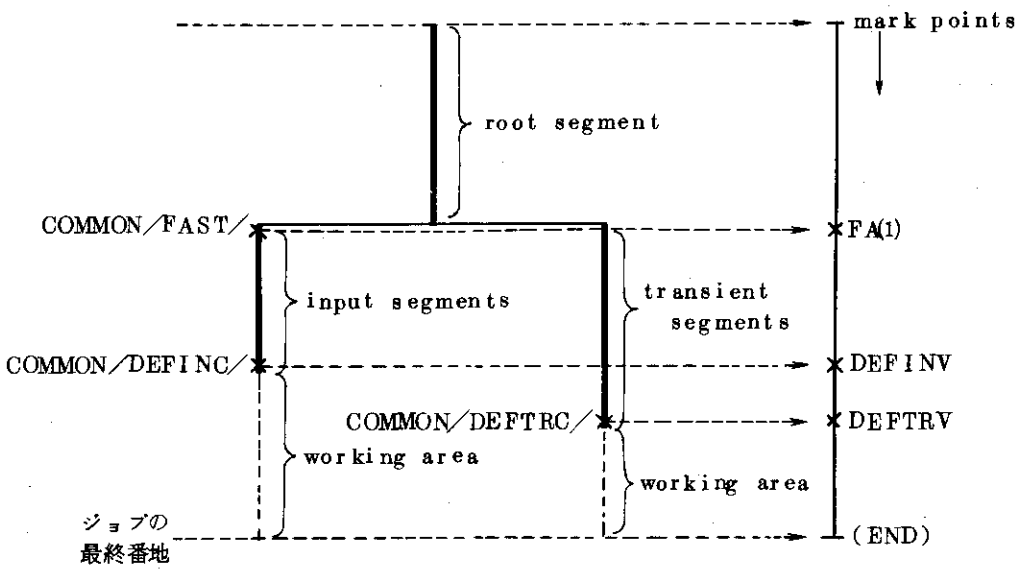
ＭＯＤ６のオーバレイ構造では，ＭＯＤ５と異なり，プログラム領域が１つにまとめられている。このため，Dynamic Storage Allocation 用のマーク・ポイントは，２つのみとなった。この様子を図3.1ａに示す。

Dynamic Storage Allocation 用の作業領域に，ＭＯＤ５と同様にＣＯＭＭＯＮ／FAST／ 内に固定長でとり，マーク・ポイントを適当に与えてルート・セグメント上に模擬した。この様子を図3.1ｂに示す。

この際，ＭＯＤ５では長さが１６,９００倍精度語であった作業領域を，２２,０００倍精度語に拡大した。最大のサンプル・インプット（ＰＷＲデータ）を計算するために，この長さが必要であった。

マーク・ポイントを利用した“ファイル”の割付け方法は，ＭＯＤ５と同一なのでここでは

a. (IBM version)



b. (FACOM version)

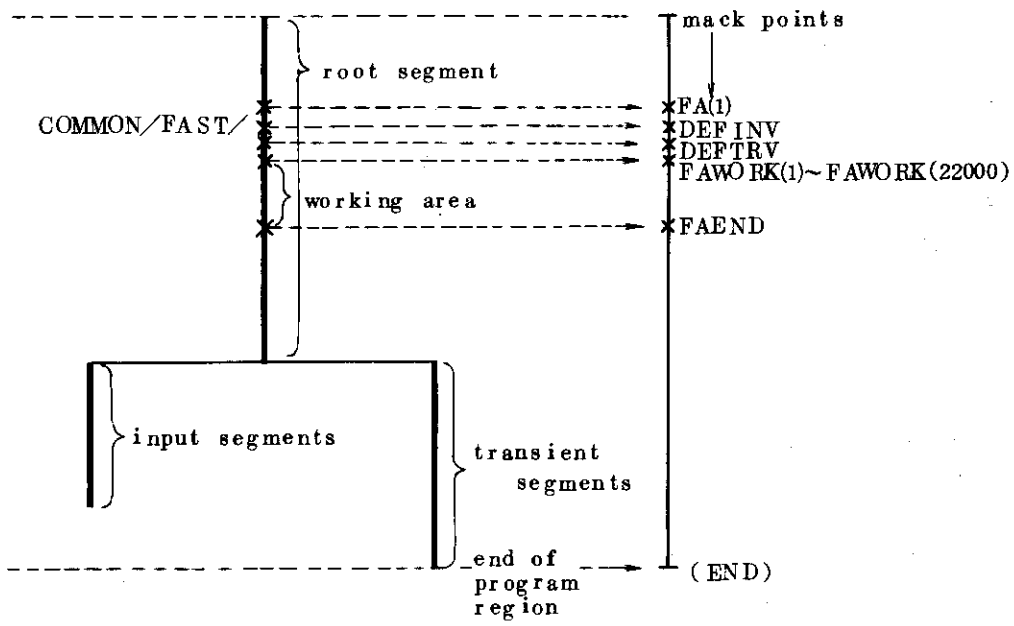


図 3.1　オーバーレイ構造とDynamic Storage Allocation用マーク・ポイント

省略する。

新たに“登録ファイル”となったジャンクション・データ，ボリューム・データ，およびムービング・メッシュ用の“ファイル”生成に注目したフローを付録3に示す。また，サンプル・インプットのリフラッド計算において，インプット・データの処理が終了し，計算に移る時点での“登録ファイル”の情況を図3.2に示す。

DYNAMIC STORAGE INFORMATION

START OF DYNAMIC POOL=324514(OCT)

CONTRL.NO.	FILE-ID	INDEX(DEC)	SIZE(DEC)
1	-8.	102	840
3	5.	2840	15
4	-3.	3199	7628
8	6.	2855	24
9	7.	942	1898
10	3.	3033	166
11	4.	2879	154

CORE REQUIRED BY DYNAMIC STORAGE=000000051712 (OCTAL)WORDS

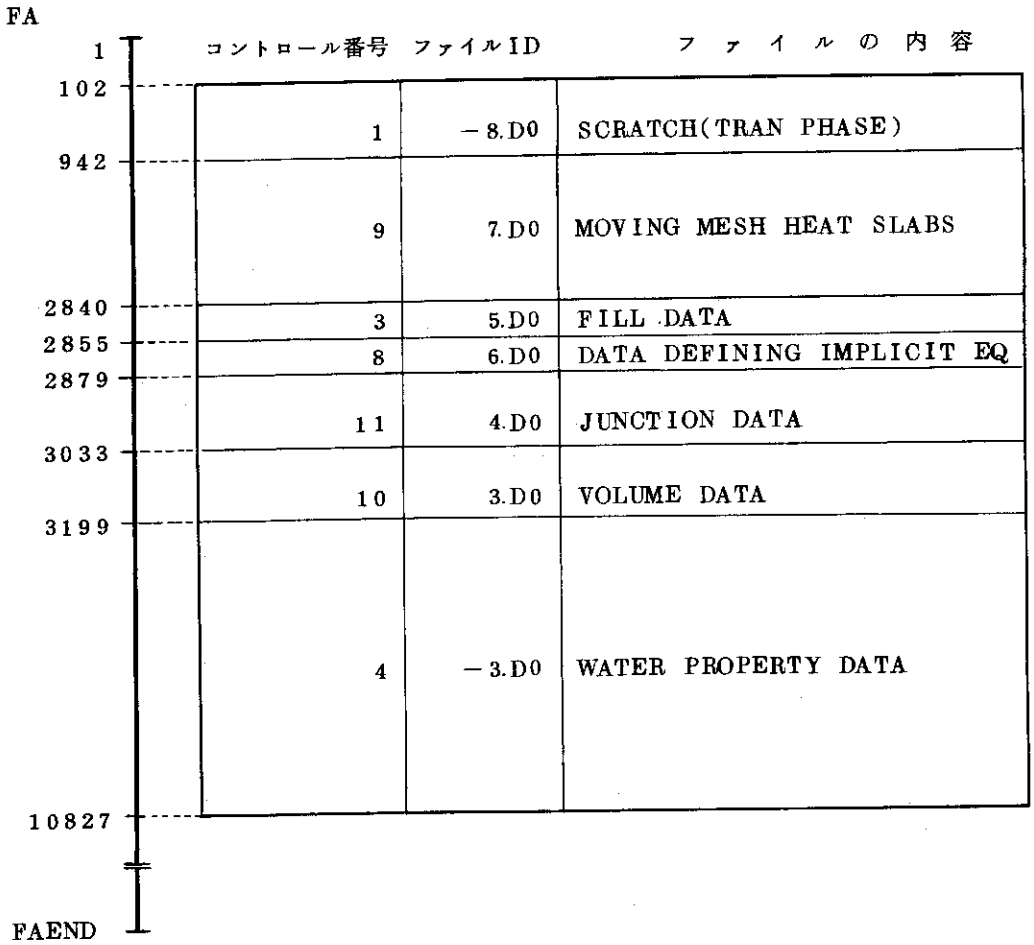


図3.2 インプット処理終了時点での“登録ファイル”の情況例
（サンプル・インプット，リフラッド問題）

3.2.3 MOD6における作業領域の使用方法

MOD5とMOD6でボリュームとジャンクションのデータ領域の利用の仕方が根本的に異る。MOD5ではボリュウム・データとジャンクションデータは固定した長さのCOMMON領域にとられていたが，MOD6では，“登録ファイル”として作業領域内に動的に割り付けられ，それにより領域の節約が計られている。さらに，データのまとめ方も大きく異っている。すなわち，MOD5では，データは変数毎の固定した長さの連続領域として取られていたが，MOD6では，以下に述べるようなEQUIVALENCE手法により各種データをまとめて（ボリュウム毎，ジャンクション毎に各種物理量をまとめる）連続領域に取りその長さは可変である。

ボリュウムとジャンクションデータはそれぞれINDEX(10)とINDEX(11)を通してFA上に割り付けられるが各種物理量がボリュウム毎，或いはジャンクション毎に一まとめになっている。

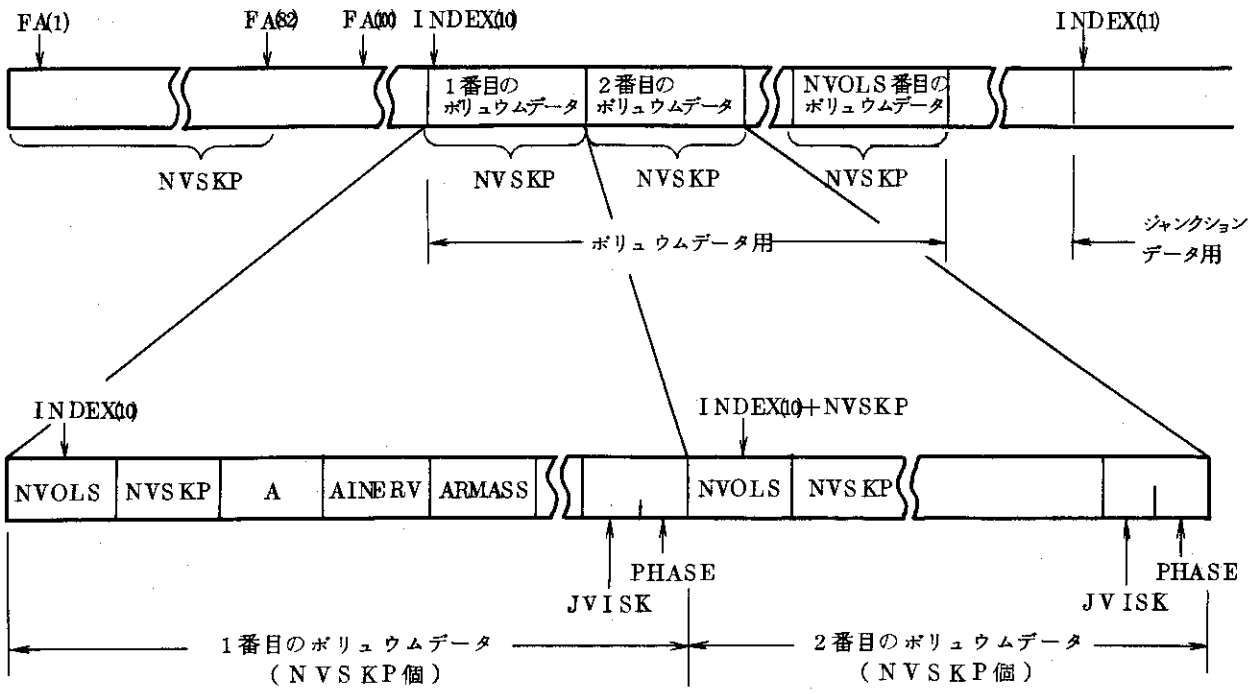


図3.3 FA上のボリュウム・データ領域

図3.3において先頭の領域FA(1)～FA(100)はEQUIVALENCE文を用いて次のように対応づけるために特別に定義された領域である。

```
COMMON/FAST/FA(100)
      :
INTEGER IZ(2,1)
EQUIVALENCE (FA(1),IZ(1,1))
EQUIVALENCE (NVOLS(1,1),IZ(1,1))
EQUIVALENCE (NVSKP(1,1),IZ(1,2))
EQUIVALENCE (A(1),FA(3))
```

```
EQUIVALENCE (AINERV(1),FA(4))
EQUIVALENCE (ARMASS(1),FA(5))
      :              :
      :              :
EQUIVALENCE (JTPMW(1),IZ(2,81))
EQUIVALENCE (JVISK(1,1),IZ(1,82))
EQUIVALENCE (PHASE(1,1),IZ(2,82))
      :              :
```

この対応づけはボリュウムデータについてのものであり，NVOLSおよびNVSKPはそれぞれボリュウムの数とボリュウムのデータ数（倍精度語数）である。具体的なデータの参照は次のように行われる。

(1) 物理量ARMASSをボリュウム毎に参照する場合，

1番目のボリュウムは　ARMASS(INDEX(10),

2番目のボリュウムは　ARMASS(INDEX(10)+NVSKP),

3番目のボリュウムは　ARMASS(INDEX(10)+NVSKP*2),

:　　:

n番目のボリュウムは　ARMASS(INDEX(10)+NVSKP*(n−1)),

:　　:

NVOLS番目のボリュウムは　ARMASS(INDEX(10)+NVSKP*(NVOLS−1)),

(2) n番目のボリュウムの各種物理量を参照する場合，

ここで，I=INDEX(10)+NVSKP*(n−1) とすると，

A(I),AINERV(I),ARMASS(I),……PHASE(1,I)

となる。

ジャンクションデータについても，同様の対応づけが，同じFA(1)〜FA(100) を通して行われる。

4. PRELOADプリプロセッサー

4.1 機能の概要

INELでは，PELAP4/MOD6の開発機種であるCDC7600のコアサイズが64kwという制限があるため，RELAP4のような巨大なプログラムをいかに小さなコアサイズで実行するかに苦慮している。そして，その一つの方法としてINELでこのPRELOADプリプロセッサーを開発した。これはPELAP4/MOD6の計算に用いられないプログラムオプションを把握し，それらに対応した不要ルーチンを実行形式プログラムから取り除くためのもので

```
EQUIVALENCE (AINERV(1),FA(4))
EQUIVALENCE (ARMASS(1),FA(5))
      :              :
      :              :
EQUIVALENCE (JTPMW(1),IZ(2,81))
EQUIVALENCE (JVISK(1,1),IZ(1,82))
EQUIVALENCE (PHASE(1,1),IZ(2,82))
      :              :
```

この対応づけはボリュウムデータについてのものであり，NVOLSおよびNVSKPはそれぞれボリュウムの数とボリュウムのデータ数（倍精度語数）である。具体的なデータの参照は次のように行われる。

(1) 物理量ARMASSをボリュウム毎に参照する場合，

1番目のボリュウムは　ARMASS(INDEX(10),

2番目のボリュウムは　ARMASS(INDEX(10)+NVSKP),

3番目のボリュウムは　ARMASS(INDEX(10)+NVSKP*2),

:　　:

n番目のボリュウムは　ARMASS(INDEX(10)+NVSKP*(n−1)),

:　　:

NVOLS番目のボリュウムは　ARMASS(INDEX(10)+NVSKP*(NVOLS−1)),

(2) n番目のボリュウムの各種物理量を参照する場合，

ここで，I=INDEX(10)+NVSKP*(n−1) とすると，

A(I),AINERV(I),ARMASS(I),……PHASE(1,I)

となる。

ジャンクションデータについても，同様の対応づけが，同じFA(1)〜FA(100) を通して行われる。

4. PRELOADプリプロセッサー

4.1 機能の概要

INELでは，PELAP4/MOD6の開発機種であるCDC7600のコアサイズが64kwという制限があるため，RELAP4のような巨大なプログラムをいかに小さなコアサイズで実行するかに苦慮している。そして，その一つの方法としてINELでこのPRELOADプリプロセッサーを開発した。これはPELAP4/MOD6の計算に用いられないプログラムオプションを把握し，それらに対応した不要ルーチンを実行形式プログラムから取り除くためのもので

ある。このため，RELAP4のインプットデータを解析して用いられないプログラムオプションを選出し，それらの不要ルーチンをリンケージの際取り除く形のオーバーレイ制御カードを

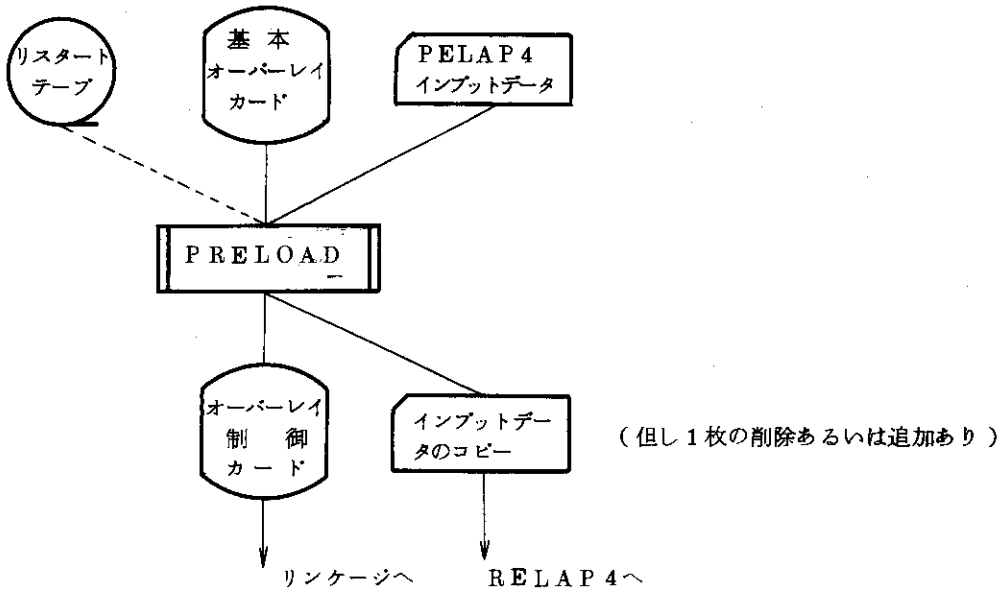


図4.1　PRELOADの処理フロー

作成する。上図はPRELOADの処理フローを示している。PRELOADへの入力データは主として2種類あり，1つは，基本オーバーレイカードであり，もう1つは，RELAP4のインプットデータである。前者はすべてのプログラムオプションに対して用意されたリンケージ制御カードであり，必要なカードのみがリンケージ処理プログラムに渡される。なお，このリンケージの対象から除かれるルーチンの総数は119個である。

4.2　処理の概要

(1)　RELAP4/MOD6のオーバーレイ構造

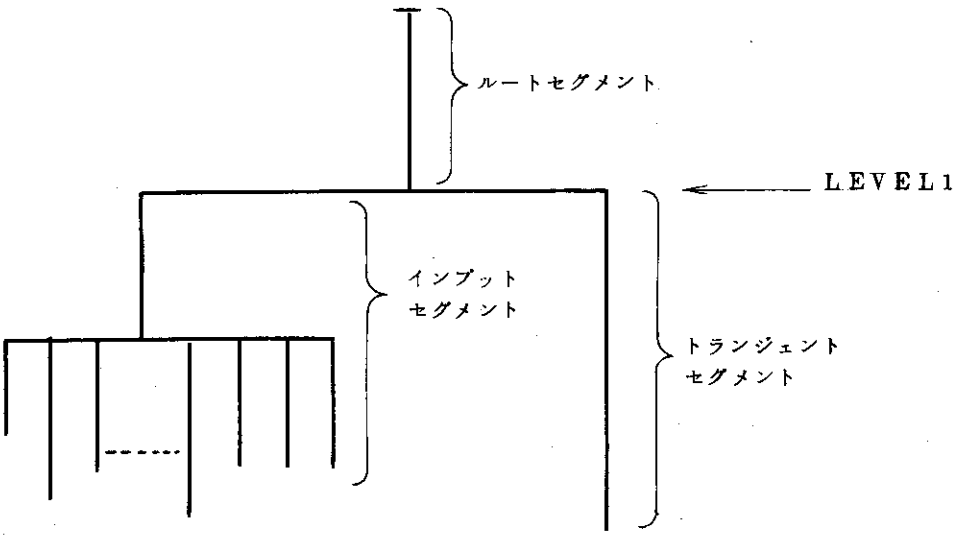


図4.2　RELAP4/MOD6のオーバーレイ構造の概略

PRELOADにより上図のルートセグメントとトランジェントセグメントの大きさが縮少される。

(2) 基本オーバーレイカードの構成

PRELOADは次に示すような基本オーバーレイカードを操作し，リンケージ処理プログラムのためのオーバーレイ制御カードを作成する。

```
REPLACE  A
REPLACE  B
REPLACE  C
REPLACE  E
REPLACE  F
REPLACE  G
    :
REPLACE  X
REPLACE  Y
REPLACE  Z
    :
    :
OVERLAY  LEVEL1
    :
    :
    :
OVERLAY  LEVEL1
INSERT   a1,a2,a3,……an
INSERT   b1,b2,b3,……
    :
INSERT   A,B,C
INSERT   E,F,G
    :
ENTRY    MAIN
```

トランジェントセグメントから除くためのコントロールカード（トランジェントセグメントにこの逆のINSERT文が対応している。）

トランジェントセグメントから除くためのコントロールカード（対応するINSERT文がない）

トランジェントセグメントへ組み込むためのコントロールカード

図 4.3　基本オーバーレイカードの例

(3) 処理手順

1) 基本オーバーレイカードとフラグの関係

PRELOADは54種のプログラムフラグを持ち，RELAP4/MOD6のインプットデータからRELAP4/MOD6のプログラムオプションの選択を認識し，それらとフラグを対応させている。これらの対応関係は表4－1に示す。PRELOADはプログラムオプションが選択された場合に，それに対応するフラグを“ON”（.TRUE.）にし，逆の場合“OFF”（.FALSE.）にする。つまり，フラグが“ON”の場合にそのフラグに対応するルーチンは必要となる。PRELOADは，フラグが“ON”の場合に基本オーバーレイカードから

そのフラグに対応する“REPLACE”カードを削除し，“INSERT” カードを採用する。又，それが“OFF”の場合は“REPLACE”カードを採用し，“INSERT” カードを削除する。このようにして，PRELOADは最適なオーバーレイ構造を完成させている。

これらフラグは複数のルーチンと対応しているので表4.2にフラグとルーチンの関係を示す。以下に基本オーバーレイカードがフラグの状態によりどのように完成するかを例に上げる。

（例）

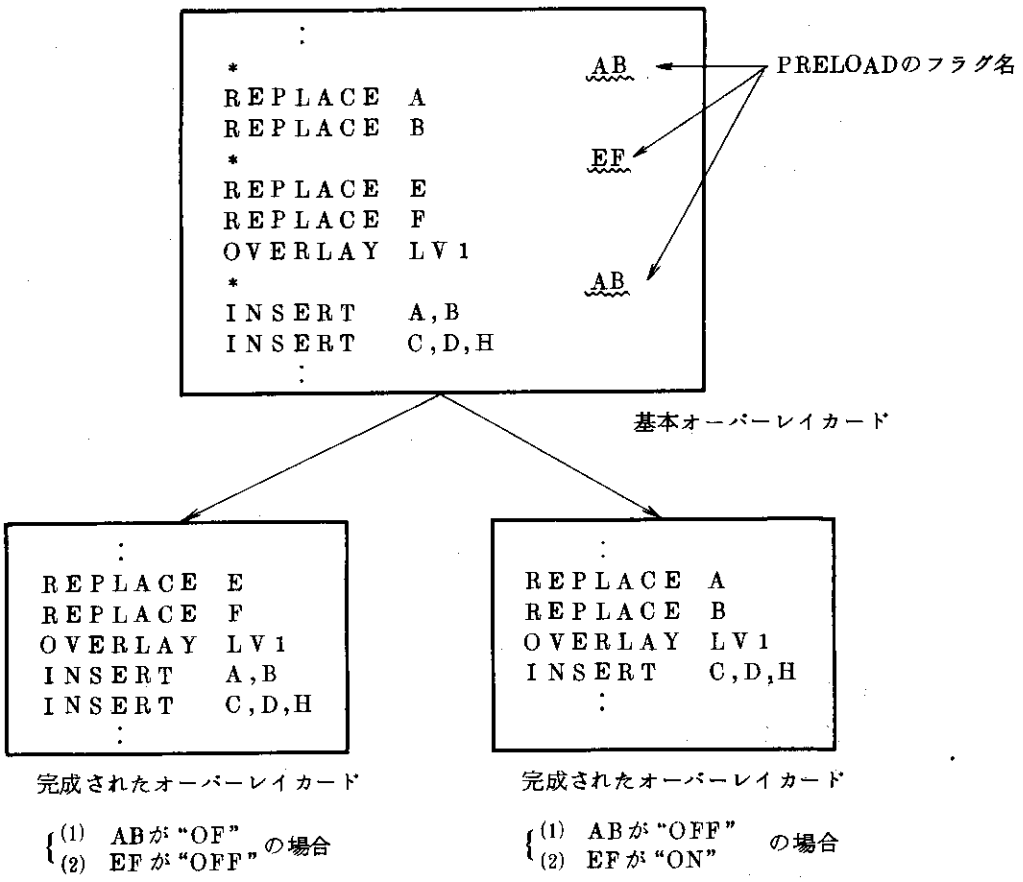


図 4.4　処理手順の例

表4.1　RELAP4/MOD6のプログラムオプションとPRELOADのフラグの関係

カード番号	シンボル名	“ON”条件	フラグ名
010001	IUNITS	=2，4	SI
	NTDV	>0	TDV
	NPMPC	>0	NPMP
	NCKV	>0	NCKV
	NLK	>0	NLK
	NFLL	>0	FILL
	NSLB	>0	NSLB
	NCOR	>0	NCOR
	NHTX	>0	HTXQ
	ISPROG	=1	EM，IEMEC
	〃	=2	FLOOD
	〃	=3	CONT

カード番号	シンボル名	"ON"条件	フラグ名
010003	IEMPS	=1	EM
	〃	≧10	FRAPBE
	IEMEC	=1	IEMEC
030004	XMLCFA	<999999.0	SMOTH
05XXXY	XREAD	≠0	TDV
060003	JCNDWN	≧1 and ≦NJUN	DWNFLD
060004	IX	=1,2	PLNENT
	〃	=2,3	DWNPEN
060006	POLY	≠0.0	SWESP
0600X2	WLCI	>0.0	WALLIS
08000X	――――	カードあり	FLOSMO
08XXXY	JCHOKE	=3,6	FLORSR,STAG
	〃	=5	}FLORSR
	ICHOKE	≦10	
	MVMIX	=1,2	MIXFLO
	IHQCOR	≠0	ENTR
	IFLOOD	=1	FALBAK
	IADJUN	>0	HORSLP
082000	ISTAGP	=1	STAG
11XXX0	8 th word	≠0	PWRMIX
13XXYY	ISATFL	=1,−1	BWRMIX
	〃	=2	PWRMIX
140000	NODEL	=−1	PWRRET
150000	――――	カードなし	NSLB,HTS2
	NSUR	=−1	HTRC,QDOT
	〃	=1	HTR1
	〃	=2	HTR2
	〃	=3	HTR3
	〃	=4	HTR4
15XXX4	ISHD	=1	HNCV
205000	NXP	≠0	SPRAY
400011	――――	カードあり	REFLD
400016	――――	カードあり	REFLD
400017	ISHVOL(2)	>0	SUPHT
400018	ISUPEX	=0	VAPEXP
	〃	=1	VAPOR1
フラグ同志により決定されるフラグ(関係式の値がフラグにセットされる)	.NOT.SI		EM
	EM.OR.FRAPBE		FRAP
	.NOT.FRAPBE		TKANDC
	HTS2.OR.STAG		STH2O5
	FLOOD.OR.REFLD		FLODRE
	TDV.OR.PWRRET.OR.FLOSMO		TAPE2
	HTS1.OR.HTS2.OR.HTS3.OR.HTS4.OR.REFLD		HTMOD6

表4.2 フラグとルーティンの関係

	フラグ名	ルーティン名
1	BWRMIX	BWRECC, BWRMIX
2	CONT	STAIR, ICETRA
3	DWNFLD	DWNFLD
4	DWNPEN	DWNPEN
5	EM	REFRAP, DEFORM, CELMOD, CTHSTR, FTHEXP, GTHCON
6	ENGL	EDIT
7	ENTR	ENQADD, ENTRAN
8	FALBAK	FALBAK
9	FILL	FILL
10	FLODRE	FLDEJ, FLDEBW, FLDES, FLDNIF, FLDBAL
11	FLOOD	FLDHTC, FLDRAF, FLDHTX, FLDPRW
12	FLOCAR	FLOCAR
13	FLORSR	FLORSR
14	FLOSMO	SMOOTH
15	FRAP	PRESWL, SWELL, MH2EM, MWROUT, MWRIN, GVISCO EPLAS, PLNT
16	FRAPBE	RFCELM, RFCLMT, RFCPIR, RFCTHS, RFDFRM, RFEMSF, RFEMFC, RFEMSZ, RFFELM, RFFPIR, RFFRAP, RFFTHC, RFFTXP, RFGAPH, RFGTHC, RFMACB, RFPRSI, RFTKAN
17	HTMOD6	TWQW, TWFIND, HSUAB, REYFAC, SUPFAC
18	HNCV	HNCV
19	HTRC	HTRC
20	HTS1	HTS1
21	HTS2	HTS2
22	HTS3	HTS3
23	HTS4	HTS4
24	HTXQ	HTXQ
25	IEMEC	ECCADJ
26	MIXFLO	MIXFLO
27	MMESH	INITM, COMBL, COMBM, COMBS, DROPM, GETM, MOVES, SHIFTM, TERMM
28	NCOR	RKEN, RNDO, CCC, SCRM
29	NSLB	COND, CSBP, SENSE, REAC, TKANDC, SENG, TAVE, PCHF
30	PLNENT	PLNENT
31	PWRMIX	PWRMIX, PWRTAU
32	NPMP	PUMP
33	QDOT	QDOT

	フラグ名	ルーティン名
34	REFLD	GRIF, STWAL, TRAWET, VENTR, XENTR, DEENTR VGENT
35	SMOTH	SMOTH
36	STAG	STAGP
37	SPRAY	SPRAY, MQFRON, QRAD, SOLV, ZQUEN
38	SUPHT	SUPHT
39	SWSEP	BWRS
40	TAPE2	RCDN, TRDAT
41	TDV	――――
42	TKANDC	TKANDC
43	SI	SICNVT, EDITSI
44	VAPEXP	VAPEXP
45	VAPOR1	VAPOR1
46	WALLIS	WALLIS
47	HORSLP	――――
48	LEVCAL	――――
49	NLK	――――
50	PWRRET	――――
51	STH2O5	――――
52	WILSON	――――
53	GLOCAL	――――
54	NCKV	――――

（注） 対応ルーティン名のないフラグは現在使用されていない。
（基本オーバーレイカードセットにそのフラグ名がないものである）

2) RELAP4/MOD6のインプットデータのコピー

一般にPRELOADはRELAP4/MOD6のデータを読み込んでそのままコピーするが，削除・創成されるカードが各々一枚ある。削除されるカードのカード番号は“010005”で，これはCDC計算機のみ有効なデータである。これはCDCオーバーレイのオプティマイズを決定するもので，IBM版では意味を持たない。PRELOADはインプットデータの処理後このカードを削除する。なお，このカードを直接RELAP4本体に入力すればエラーとなる。次に，創成されるカードはカード番号“010004”であり，これはRELAP4/MOD6インプットデータの説明にはない。このデータもやはりPRELOADのための情報である。リスタートジョブの場合，そのインプットデータだけでは採用されたプログラムオプションをPRELOADが認識できない。そこで，新たに“010004”カードを設け，オリジナルジョブの実行時に決められたプログラムオプションに対するフラグの情報を書き込むようにしている。即ち，オリジナルジョブの実行の際にPRELOADは図4.5に示すように“ON”と“OFF”のフラグに対してそれぞれ1または0を対応させ，その情報を2語の中にパックして格納する。そしてRELAP4本体が，そのデータを読み込んでリスタートテープに情報として書き出す。

リスタードジョブの場合は，PRELOADがリスタートテープからこの情報を読んでフラグを決定している。このことからリスタートジョブはオリジナルジョブと同一の実行形式プログラム（EB）で実行が可能である。

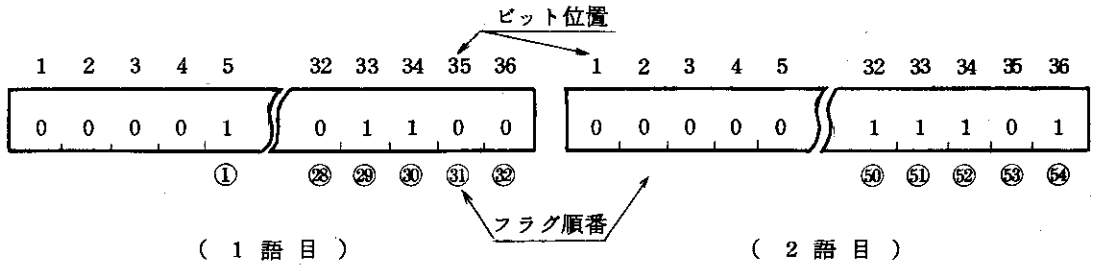


図4.5　フラグとビットの関係

4.3　FACOMへの変換

PRELOADのFACOMへの変換には2通りの方法があった。1つは，基本オーバーレイカードをFACOM用に変換すること（FACOM用リンケージ制御カードにする）と，もう1つは，PRELOADにより完成されたIBM用リンケージ制御カードをFACOM用に変更することである。今回我々は後者を選んだがそれは，今後予想されるRELAP4の新しいバージョンを整備する際にはこのルーティンをPRELOADに追加することで良い。又，このルーティンを単独で利用することによりIBM版リンケージ制御カードをFACOM版に変換すること等である。（但し，この方法でも基本オーバーレイカードを多少修正する必要がある。すなわち，labeled COMMON名はコメント化，サブルーティンの主入口名がFACOM版で異る場合の修正等が必要である。）

PRELOADを使用してのリンケージではソースプログラムを修正せず不要ルーティンの削除を行うため，未定義ルーティンエラーが生ずる。これは利用の面から不便さがある。したがって，これら未定義ルーティンは，STOP，END のダミールーティンとして作成し，ライブラリーのように自動コールの対象とし不要ルーティンのかわりにリンケージされるようにした。

このPRELOAD機能を利用する場合，毎回リンケージの必要があるがPRELOADとリンケージに要する時間は約1分位である。又，この機能を利用する場合繁雑なJCLを必要とするので利用ジョブマクロを作成した。RELAP4/MOD6のジョブフローと私用ジョブマクロの使用方法は，“5. MOD5/MOD6 の使い方”で記す。提供されたサンプルデータ8ケースについてPRELOADを使用した場合のコアサイズを表4.3に示す。なお，表中のコアサイズにはDynamic Storage Allocation機能が使用する作業領域（FAWORK）43kw（22000倍精度語）を含んでいる。

表4.3　標準オーバーレイとPRELOADによるオーバーレイのコアサイズ

ケース番号	サンプルデータセット名	標準オーバーレイコアサイズ(kw)	PRELOADによるコアサイズ(kw)
1	6 volume	280	215
2	Restart	280	215
3	Re−restart	280	215
4	Re−edit	280	185
5	Typical PWR	280	216
6	Flood	280	210
7	BWR 100% break	280	219
8	Reflood	280	222

5. MOD5／MOD6の使い方

5.1 MOD5の使い方

(1) 実行に必要となるコア・サイズと入出力機番

(i) コア・サイズ：260k語，C.4で実行可能。

（FAWORK(16900) で，実行時のパラメータとしてSIZE=9を指定した時）

(ii) 入出力機番

F02：境界条件入力ファイル

F03：リスタート・ファイル（入力）

F04：リスタート・ファイル（出力）

READ：インプット・データ・ファイル

PRINT：ライン・プリンター出力ファイル

F15：蒸気表

F16：作業用ファイル

(2) 使用JCL例

(i) オリジナル・ジョブ

```
¥HRUN PLP4M5,J0285.RLP4M5, SIZE=9
      EBのPROGNAME   EBファイル名
¥TAPE F04,J○○○○.XXXXXX,NEW,△△△△△△
          リスタート出力ファイル名   MTのボリュウム通番
¥DiSKTO READ,J0285.RLP4M5DA(CASE1)
             インプット・データ・ファイル名 エレメント名
```

表4.3　標準オーバーレイとPRELOADによるオーバーレイのコアサイズ

ケース番号	サンプルデータセット名	標準オーバーレイコアサイズ(kw)	PRELOADによるコアサイズ(kw)
1	6 volume	280	215
2	Restart	280	215
3	Re−restart	280	215
4	Re−edit	280	185
5	Typical PWR	280	216
6	Flood	280	210
7	BWR 100% break	280	219
8	Reflood	280	222

5. MOD5／MOD6の使い方

5.1 MOD5の使い方

(1) 実行に必要となるコア・サイズと入出力機番

(i) コア・サイズ：260k語，C.4で実行可能。

(FAWORK(16900) で，実行時のパラメータとしてSIZE=9を指定した時)

(ii) 入出力機番

F02：境界条件入力ファイル

F03：リスタート・ファイル(入力)

F04：リスタート・ファイル(出力)

READ：インプット・データ・ファイル

PRINT：ライン・プリンター出力ファイル

F15：蒸気表

F16：作業用ファイル

(2) 使用JCL例

(i) オリジナル・ジョブ

```
¥HRUN PLP4M5,J0285.RLP4M5, SIZE=9
      EBのPROGNAME   EBファイル名
¥TAPE F04,J○○○○.XXXXXX,NEW,△△△△△△
          リスタート出力ファイル名   MTのボリュウム通番
¥DiSKTO READ,J0285.RLP4M5DA(CASE1)
             インプット・データ・ファイル名 エレメント名
```

¥DISKP1 F15,J1598.RLP4ST
蒸気表ファイル

¥DISK F16

(ii) リスタート・ジョブ

¥HRUN RLP4M5,J0285.RLP4M5,SIZE=9

¥TAPE F03,J○○○○.XXXXXX,OLD,△△△△△△
リスタート入力ファイル名

¥TAPE F04,J○○○○.□□□□□□,NEW,△△△△△△

¥DISKTO READ,J0285.RLP4M5DA(CASE2)

¥DISKP1 F15,J1598.RLP4ST

¥DISK F16

5.2 MOD6の使い方

5.2.1 標準オーバーレイによる実行

これはPRELOAD機能を使用せずに実行する場合で，コアサイズは280 kw である。（実行時にSIZE=5を指定，又，FAWORKは22000の場合である。）

JCLはMOD5と同様であるがFBファイル名とプログラム名（PROGNAME）が異る。

¥HRUN RLP4M6,J0285.RLP4M6

¥TAPE F04,J○○○○.XXXX,NEW,△△△△△△
ファイル名　ボリュウム通番

¥DISKP1 F15,J1598.RLP4ST

¥DISK F16

¥DATA

5.2.2 PRELOADによる実行

RELAP4/MOD6はPRELOAD機能を使用して実行する場合，JCLが繁雑になるので私用ジョブマクロを作成した。以下にRELAP4/MOD6のジョブフロー，私用ジョブマクロの説明と例を上げる。その私用ジョブマクロを付録5に示す。

(1) RELAP4/MOD6のジョブフロー

RELAP4/MOD6のジョブフローを図5.1に示す。

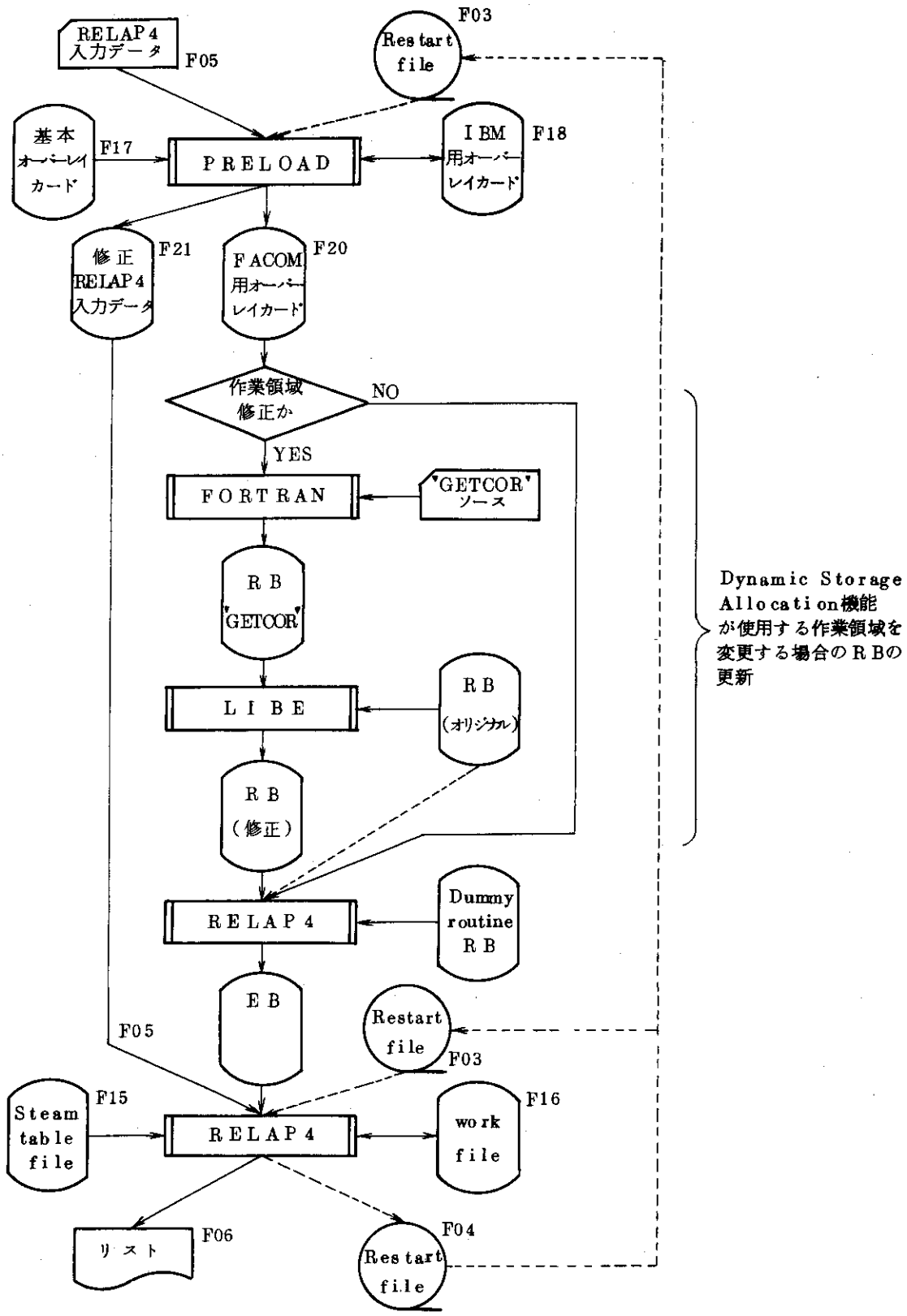


図 5.1　RELAP4/MOD6のジョブフローチャート

(2) 私用ジョブマクロ名とそのジョブステップ名

前図に対応して私用ジョブマクロを作成した。以下に，ジョブステップとそれに対応するマクロ名を示す。

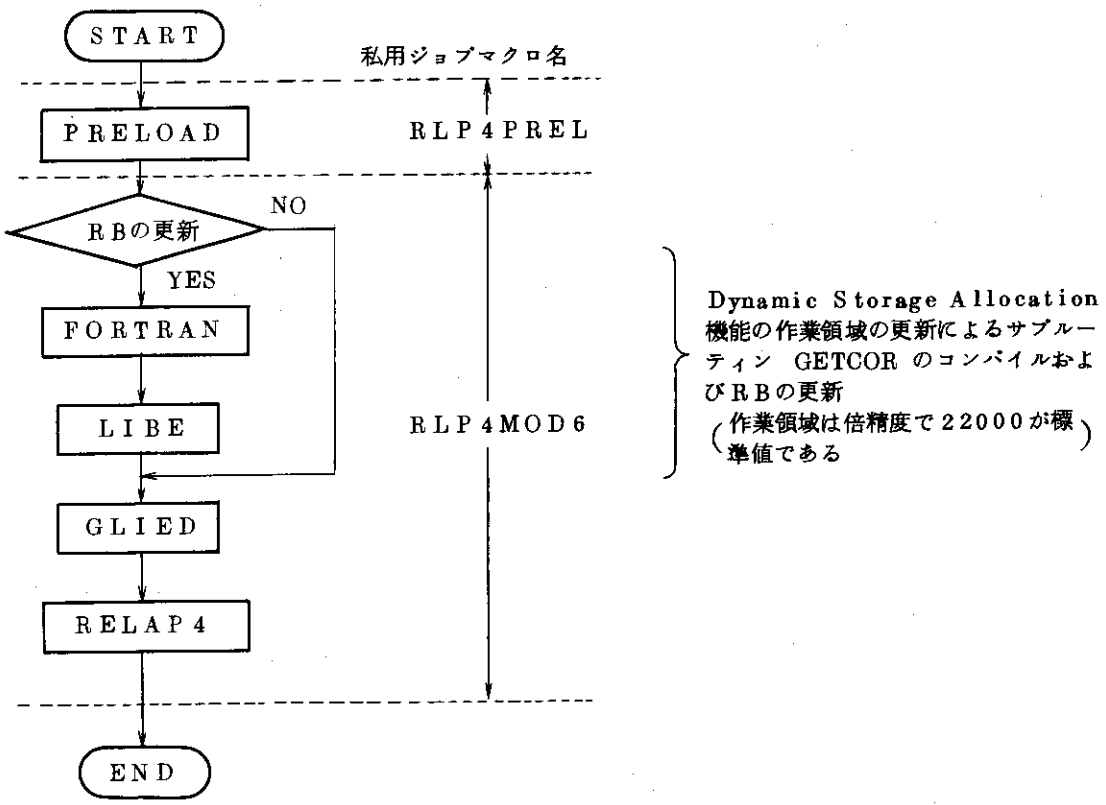


図 5.2 私用ジョブマクロのジョブステップフロー

ここで，この私用ジョブマクロは2つに分割してあるがPRELOADで入力となるRELAP4の入力データがSYSIN(カード直接入力)型式で与えられる場合も考慮したためで常に対で使用しなければならない。又，RELAP4の実行までに要する時間は約1分である。

5.2.3 私用ジョブマクロの使い方

以下にa．パラメーターとその意味，b．組み込みジョブステップ，c．組み込み使用I/Oを示す。

(1) RLP4PREL

a.

パラメーター	意　　味	標　準　値
NAME	ジョブステップ名	1
SYSOUT	SYSOUTクラス名	A
SIZE	I／O 作業領域	5
E	実行パラメーター	――

b. PRELOAD

c. F17, F18, F20, F21, F06(PRINT)

(2) RLP4MOD6

a.

パラメーター	意　　味	標　準　値
NAME	ジョブステップ名	1
SYSOUT	SYSOUTクラス名	A
FAWORK	RELAP4作業領域FAWORKの大きさ	ON
LKP	GLIED実行パラメーター	———
MAP	GLIED MAPパラメーター	MAP
RNP	RELAP4実行パラメーター	———
SIZE	I／O作業領域	5

b. FORTRAN　サブルーティンGETCORのコンパイル
LIBE　サブルーティンGETCORのRBの更新
} パラメーターFAWORKが"ON" 以外実行

GLIED

RELAP4

c. F05(READ), F06(PRINT), F15, F16

(3) 私用ジョブマクロの使用例

以下に私用ジョブマクロを使ってのRELAP4／MOD6の例を上げる。

(i) 一般のオリジナルジョブ

```
¥RLP4PREL
¥DATA
¥RLP4MOD6
¥TAPE F04,～          ←― 必要に応じて
¥JEND
```

(ii) 一般のリスタートジョブ

```
¥RLP4PREL
¥DISKTO READ,～
¥TAPE F03,～
¥RLP4MOD6
¥TAPE F03,～
¥TAPE F04,～
¥JEND
```

同一ファイル

(iii) 作業領域(FAWORK)の変更

作業領域(FAWORK)が標準値22000では不足(又は過剰)している場合にその値を修正したい時以下のように指定すれば良い。

```
¥RLP4PREL
```

```
¥DATA
¥RLP4MOD6 FAWORK=○○○○○
¥TAPE F03,～                    ←―― 必要に応じて
¥JEND
```

(V) 特殊な使用例

RELAP4/MOD6使用者の個人的な新しいルーティンの組み込みやRELAP4/MOD6のサブルーティンの更新をしたい場合のジョブについて例を上げる。

a. ソースプログラムのよりコンパイル・ラン

```
¥RLP4PREL
¥DISKTO READ,～
¥HFORT SFNAME=～
¥RLP4MOD6 FAWORK=22000   ←―― (注)
¥TAPE F03,～
¥JEND
```

b. RBよりのリンケージ・ラン

```
¥LIBE
EDIT DDNEW0,DDOLD0
FIN
¥DISKTO DDOLD0,～
¥LIBEDISK DDNEW0,FORT.RELBIN
¥RLP4PREL
¥DISKTO READ,～
¥TAPE F03,～
¥RLP4MOD6 FAWORK=22000   ←―― (注)
¥TAPE F03,～
¥TAPE F04,～
¥JEND
```

(注) 新しいRBの置換のため，FAWORKの大きさを変更する必要がない場合でも指定する必要がある。

5.3 プロッターの使い方

プロッタープログラムPLOTR4はMOD5,MOD6 で同一である。今回，PLOTR4についても私用ジョブマクロを作成してあり，以下に使用方法等を述べる。なお，マクロ名は"RLP4PLOT"であり，マクロリストを付録5に示す。

(1) 私用ジョブマクロの使い方

a. パラメーター

パラメーター	意　　味	標　準　値
NAME	ジョブステップ名	1
SYSOUT	SYSOUTクラス名	A
SIZE	I／O作業領域	40
RNP	実行パラメーター	———
COM	COM，PLTの選択（EB）	ON

（注）　パラメーター“COM”はONの場合はCOMでNO，PLTなど，つまりON以外はPLTとなる。

b. 組み込みジョブステップ

PLOTR4

c. 組み込みI／O

F06(PRINT),F11,F12,～, F30

ここでF11～F30はRCDSIZE=18,BLKSIZE=3114 である。

(2) 私用ジョブマクロの使用例

(i) COMの場合

使用コアサイズは“SIZE=40” の場合95kwである。

```
¥RLP4PLOT
¥DATA
¥TAPE F01,～
¥TAPE F02,～        ←―― 必要に応じて
¥GCOM35
¥JEND
```

(ii) PLTの場合

使用コアサイズは“SIZE=40” の場合93kwである。

```
¥RLP4PLOT   COM=PLT
¥DISKTO READ,～
¥TAPE F01,～
¥PLOT
¥JEND
```

6. テストラン

MOD5とMOD6の整備確認テストは，ともに，開発機関より提供されたサンプルインプットを用いて行った。

(1) MOD5

入手したMOD5はIBMバージョンであり，これにはサンプルアウトプットが付いていな

パラメーター	意　　味	標　準　値
NAME	ジョブステップ名	1
SYSOUT	SYSOUTクラス名	A
SIZE	I／O作業領域	40
RNP	実行パラメーター	――
COM	COM，PLTの選択（EB）	ON

（注）　パラメーター“COM”はONの場合はCOMでNO，PLTなど，つまりON以外はPLTとなる。

b.　組み込みジョブステップ

PLOTR4

c.　組み込みI／O

F06(PRINT),F11,F12,～, F30

ここでF11～F30はRCDSIZE=18,BLKSIZE=3114　である。

(2)　私用ジョブマクロの使用例

(i)　COMの場合

使用コアサイズは“SIZE=40”　の場合95kwである。

```
¥RLP4PLOT
¥DATA
¥TAPE F01,～
¥TAPE F02,～        ←―― 必要に応じて
¥GCOM35
¥JEND
```

(ii)　PLTの場合

使用コアサイズは“SIZE=40”　の場合93kwである。

```
¥RLP4PLOT   COM=PLT
¥DISKTO READ,～
¥TAPE F01,～
¥PLOT
¥JEND
```

6.　テストラン

MOD5とMOD6の整備確認テストは，ともに，開発機関より提供されたサンプルインプットを用いて行った。

(1)　MOD5

入手したMOD5はIBMバージョンであり，これにはサンプルアウトプットが付いていな

かったので，あらかじめ同種の計算機で計算を行い，その結果とＦＡＣＯＭでの計算結果と比較した。提供されたサンプルケースは８ケースあり，典型的なＰＷＲの大破断やＢＷＲ大破断問題が含まれている。両者の間には有意な差は認められなかった。（後に，ＭＯＤ５の検討のために設けられたワーキンググループにおいて，ＭＯＤ３で問題とされた自動タイムステップコントロールオプションを用いた際の解の不安定性についても検討が行われ，ＭＯＤ５ではモデルの改良により著しく安全性が増しているとの報告がなされている。）

(2) ＭＯＤ６

ＭＯＤ５の改良版であり，特に，ＰＷＲ再冠水過程解析に力点をおいたモデルの改良（特に，熱伝達や臨界熱流束の相関式と計算論理の大巾な改訂）がなされている。入手したバージョンはＩＢＭバージョンであるが，開発機種ＣＤＣ７６００のサンプルアウトプットも入手したので，それとの比較を行うことができた。サンプルケースは８ケースあり，ＭＯＤ５のものと同等のものが大部分で，新たにFLECHT再冠水実験の解析問題が加えられている。両者の間には有意な差は全く認められなかった。FLECHT解析問題のノード区分と計算結果を図 6.1～図 6.9 で示す。

(3) ＭＯＤ５とＭＯＤ６の比較

ＭＯＤ６がＭＯＤ５の改良版であることから，同じ問題について結果がどのように異るかについても若干の検討を行った。用いたサンプルデータは，ケース＃１（６ボリュウム問題）とケース＃５（ＰＷＲ大破断問題）である。

特に，以下に述べるような熱伝達相関式オプションの違いによる影響に着目して，比較を行った。すなわち，RELAP4／MOD6ではカード１５００００でNSUR＝－１とすると，RELAP4／MOD5 のHTC計算の論理及び相関式を使って計算する。他方，NSUR＝２を指定すると，ＭＯＤ６のブローダウンＨＴＣ相関式を使用して計算する。ブローダウンＨＴＣの相関式でＭＯＤ５とＭＯＤ６で共通しているのは，subcooled liquid forced convectionのDittus－Boelter相関式と低流量・高ボイド率時のfree convection＋radiation（モード７）の２つのみである。また限界熱流束（CHF）の相関式は全く異ったものになっている。ＭＯＤ５ではＣＨＦを圧力範囲に応じてＢ＆Ｗ－２，Barnett，及び，Modified Barnettの相関式から選択する。ＭＯＤ６では流量及びサブクール度に応じて，Tong，Hsu and Becker，Modified Zuber の相関式の中から選択する。

計算結果を比較すると，燃料棒温度に有意な差が認められる。これは，以下に述べるように，主として熱伝達相関式と計算論理の違いによると思われる。

(i) ６ボリューム問題

図 6.１０はノード区分，図 6.１１～図 6.１３はＭＯＤ５とＭＯＤ６のNSUR＝２の計算結果を比較したものである。０～ 0.２秒の間の定常状態では，燃料棒表面温度はＭＯＤ５（ＭＯＤ６でNSUR＝－１と同等）の場合は６６３．１１４ °Fで，MOD6（NSUR＝２）の場合は６８１．０２９ °Fである。この違いは，未飽和核沸騰の相関式がThom（ＮＳＵＲ＝－１）からModifield Chen（NSUR＝２）に変わり，そのＨＴＣが小さくなるので燃料棒温度が高く計算されたことによる。ＣＨＦはこの問題の場合にはNSUR＝２の方が大きく計算されるので，約１秒まで核沸騰を持続した後，短時間の高流量遷移沸騰（Modi-

fied Tong-Young)を経て，高流量膜沸騰（Condie-Bengston)のモードをとる。NSUR=－1のときには，破断後すぐに遷移沸騰に転ずるために燃料棒表面温度が上昇する。

(ii) Typical PWR 問題

図6.14にはノーディングを示したものであり，図6.15～図6.17は，MOD6（NSUR=－1）とMOD6（NSUR=2)とを比較したものである。図6.17には，さらにMOD5の結果も合わせて示してある。図6.17のSR28は，ホットアッセンブリ軸方向中心の被覆管表面温度である。DNBはNSUR=2の方が早く起り，5～6秒の第1PCTはNSUR=2の方が高い。炉心流量（図6.16，WV43)における差異は6～8秒における逆流のふるまいが目立つ程度である。SR28が，MOD5の計算とMOD6（NSUR=－1）の計算とで大きく異っている（カーブ1とカーブ3）ことの理由は目下調査中であるが，この問題では破断流についてStagnation property オプションが採用されており，それに関連した部分に相異があるためと思われる。何故なら，このオプションを用いないで同じ比較をしたところ，差はほとんど認められなかったからである。NSUR=2のとき，3.5秒～6秒の間は，ホットアッセンブリ中心部では蒸気への熱伝達が行われているが，相関式としては高流量膜沸騰の相関式によってHTCの計算が行われている。

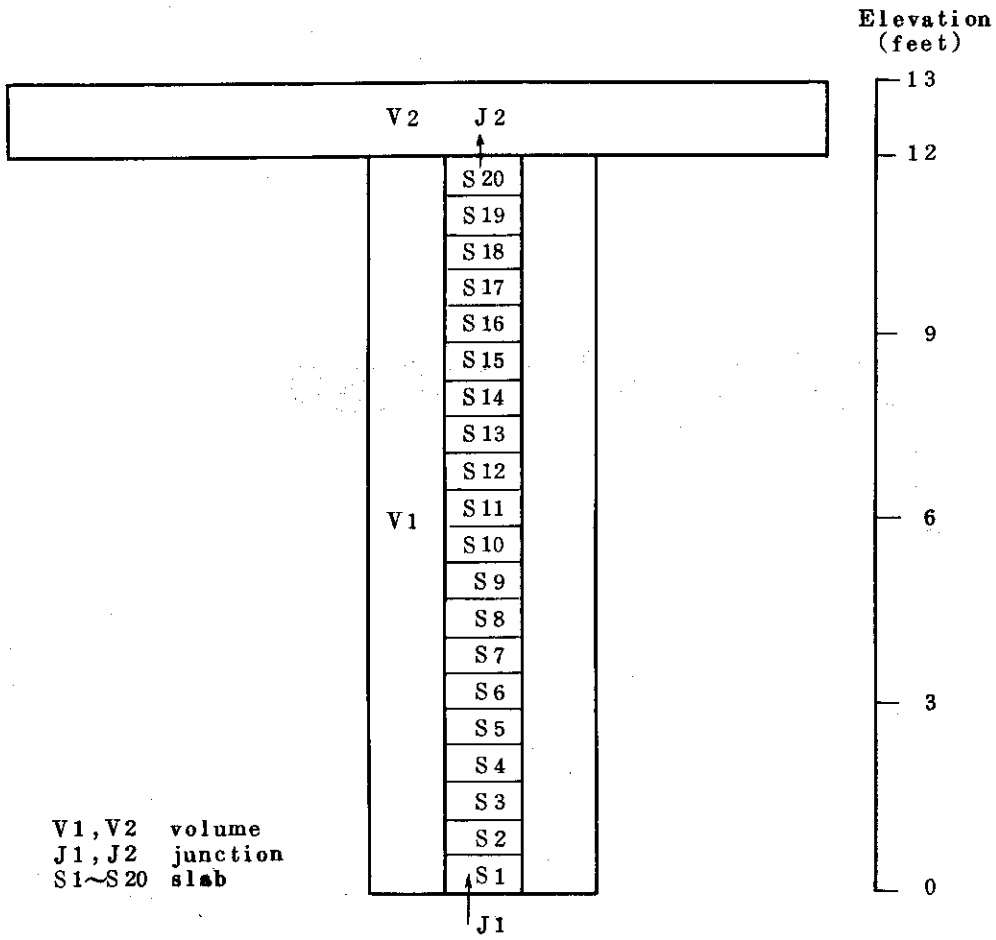


図6.1 FLECHT問題ノード区分

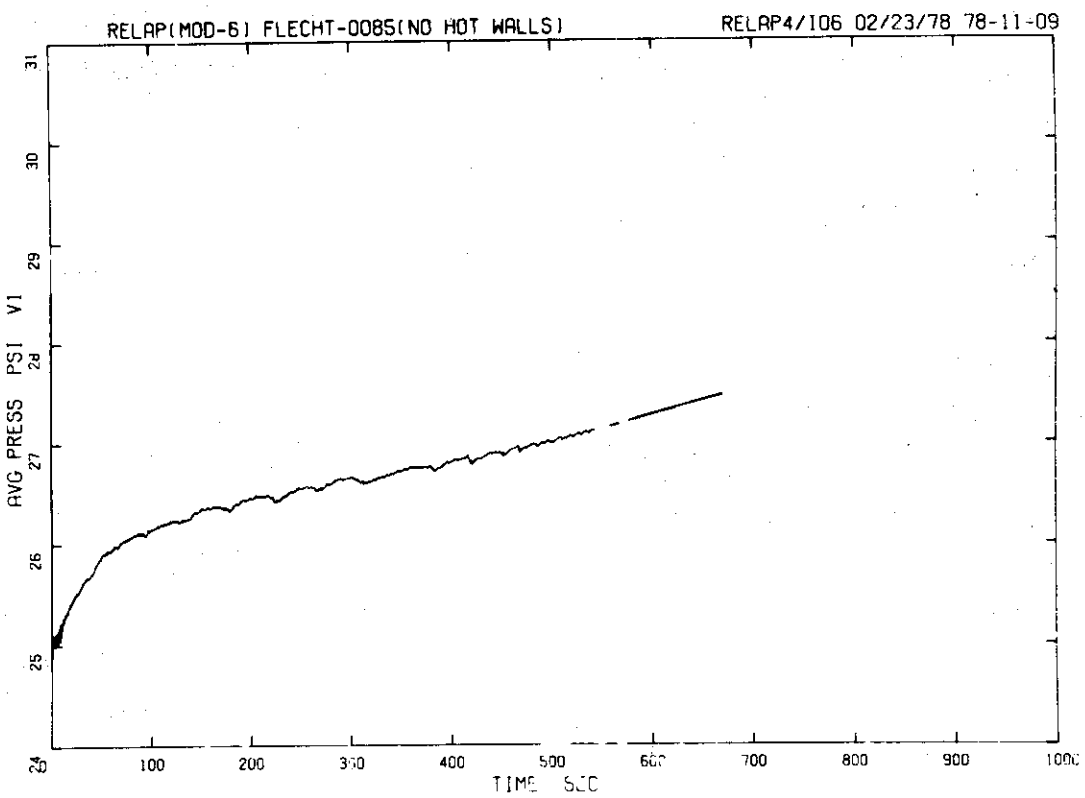


図 6.2　炉心部圧力

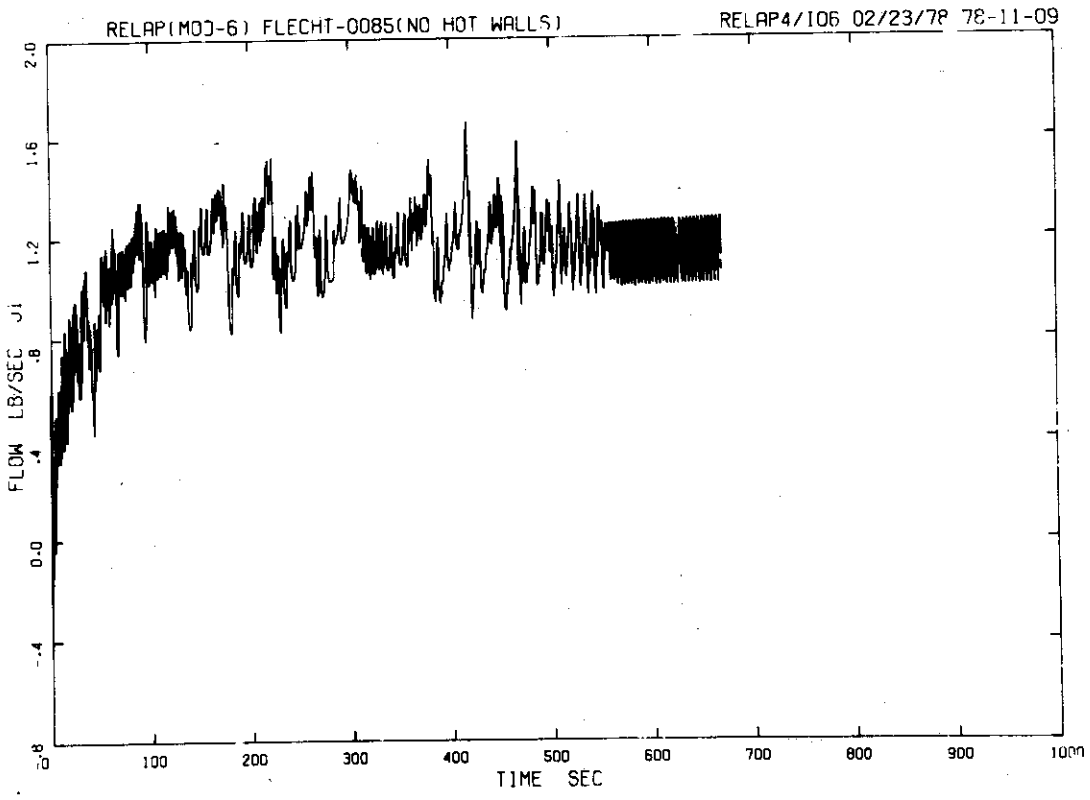


図 6.3　炉心入口流量

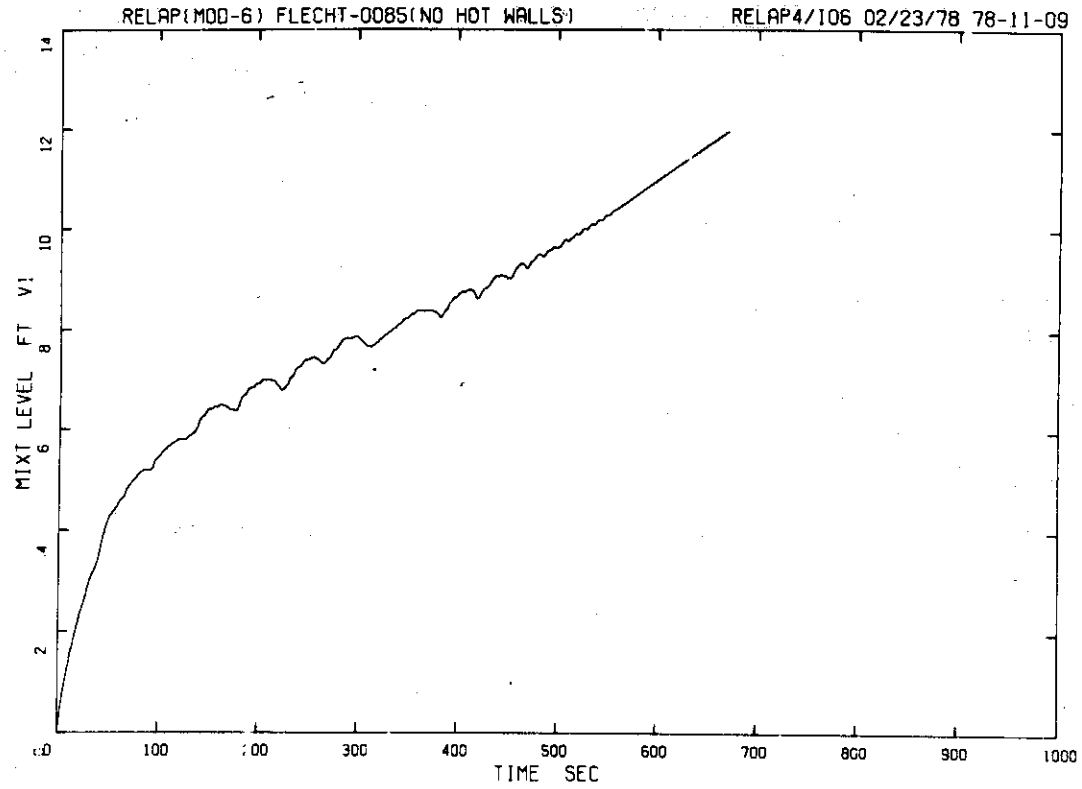


図 6.4　炉 心 水 位

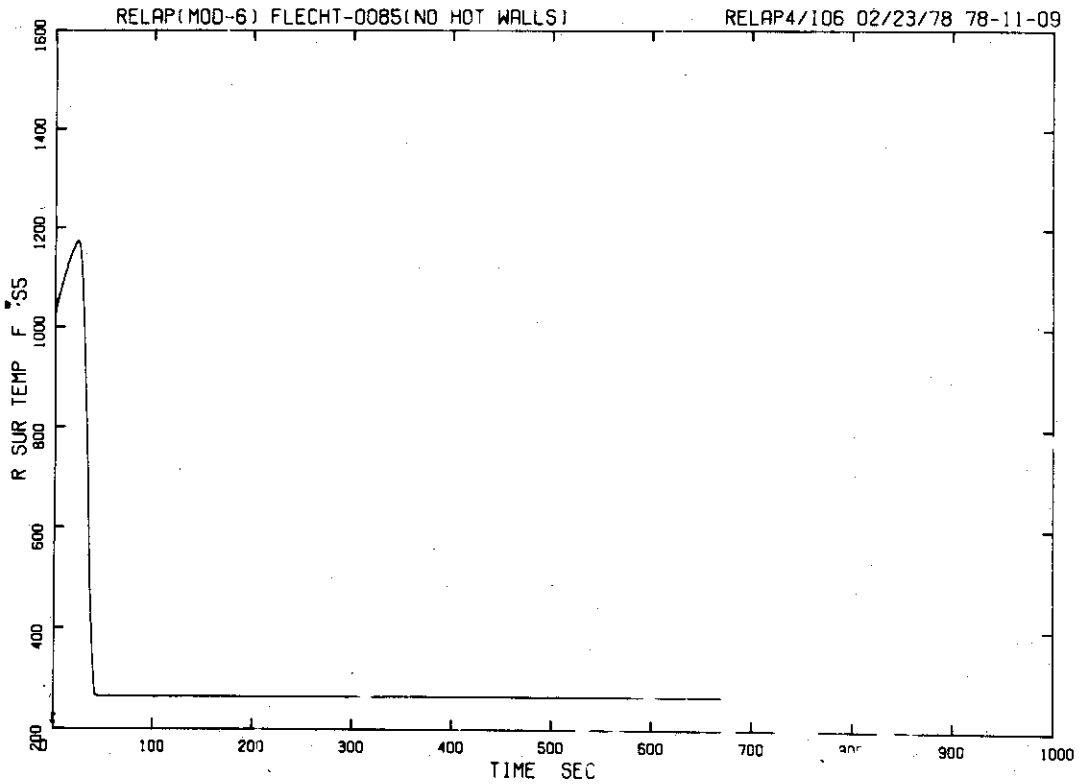


図 6.5　ヒートスラブ 5（炉心入口より 3 ft ）表面温度

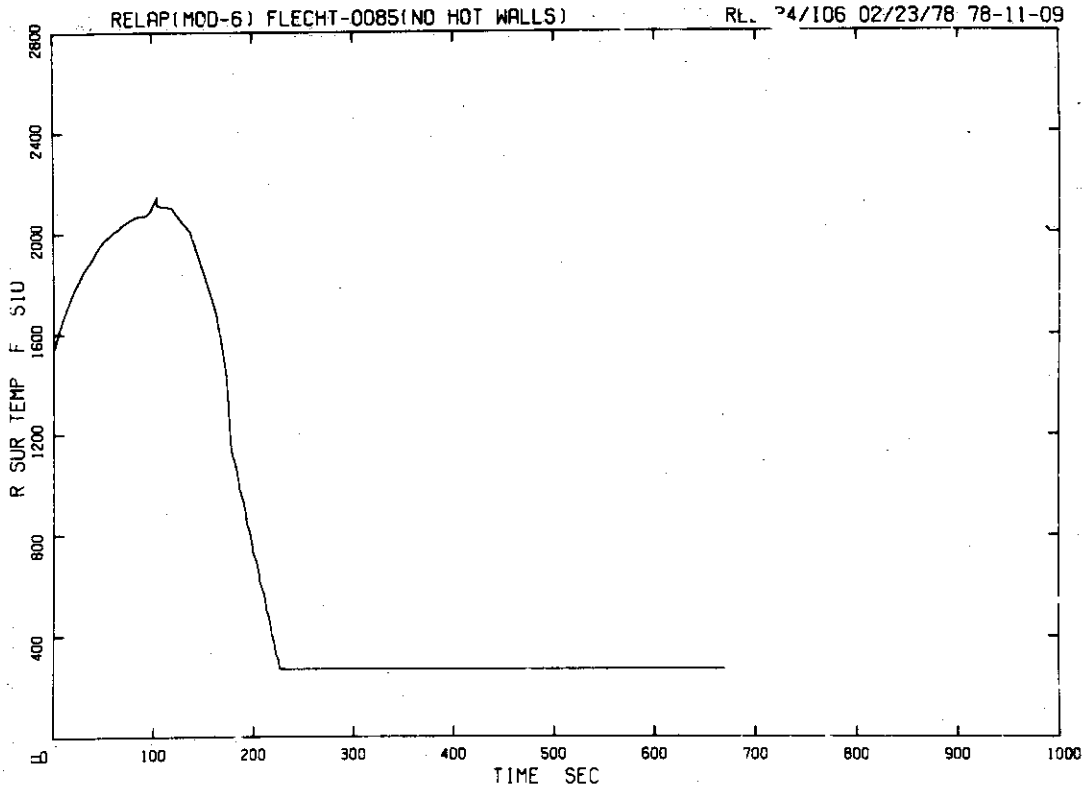


図 6.6　　ヒートスラブ 10（炉心入口より 6 ft）表面温度

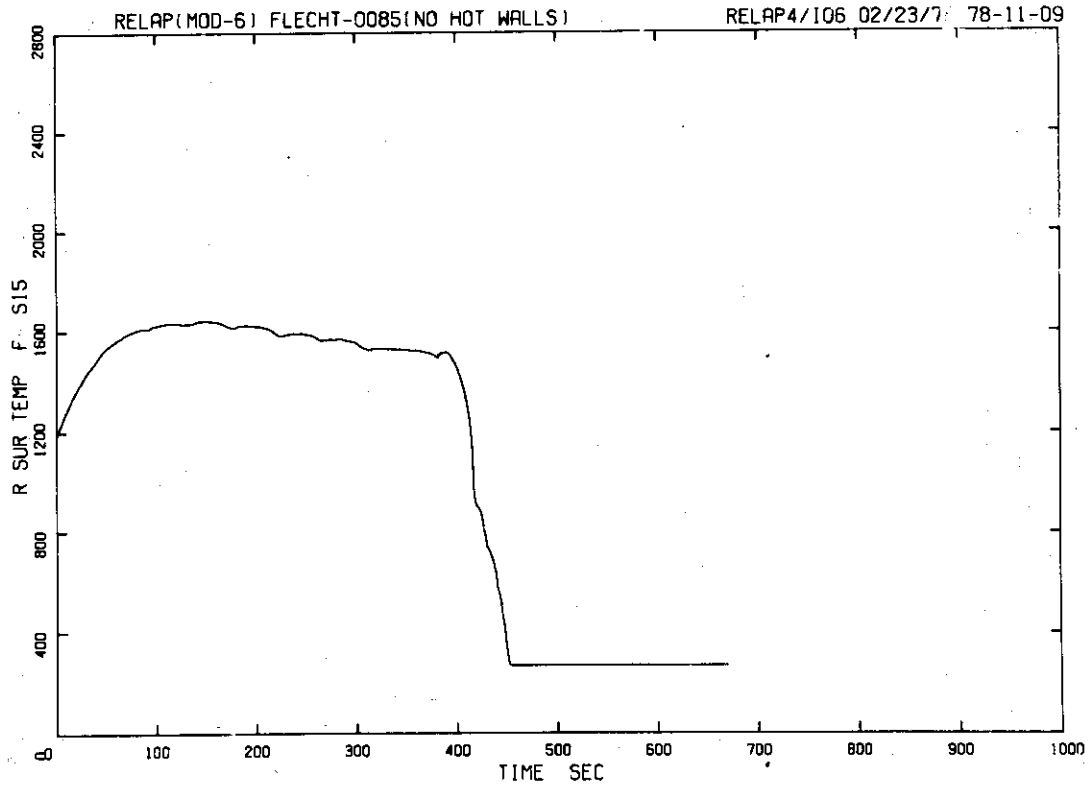


図 6.7　　ヒートスラブ 15（炉心入口より 9 ft）表面温度

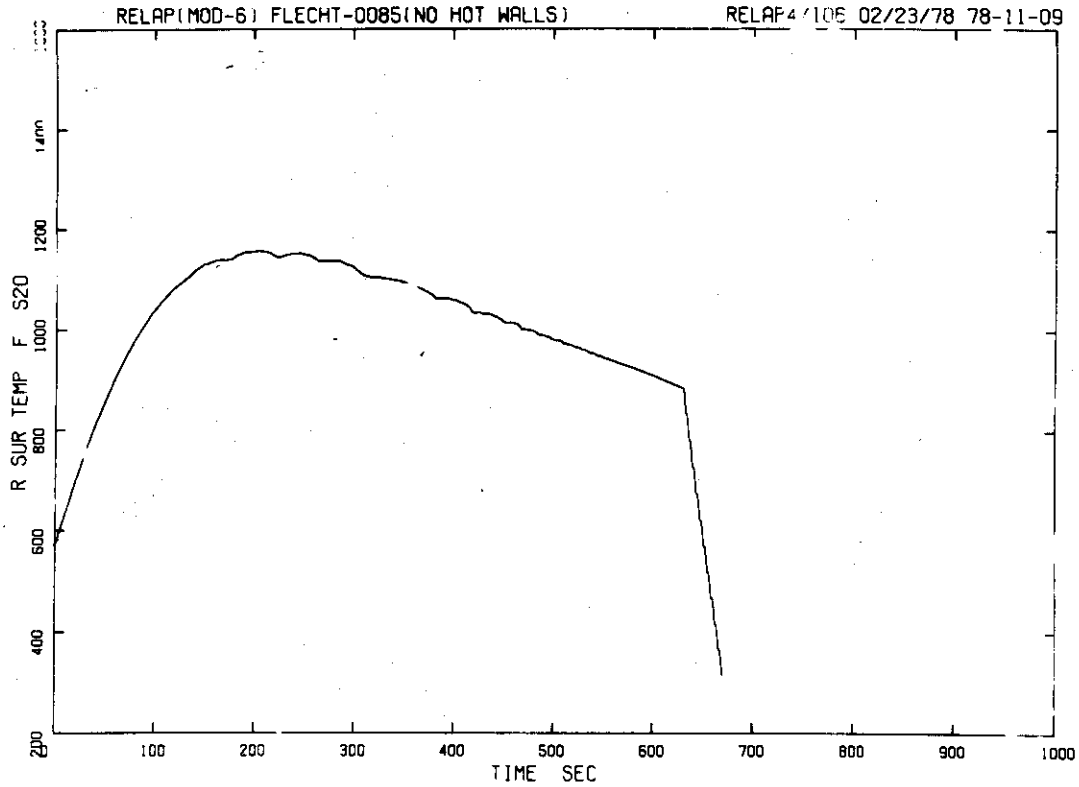


図 6.8　　ヒートスラブ 20（炉心入口より 12 ft）表面温度

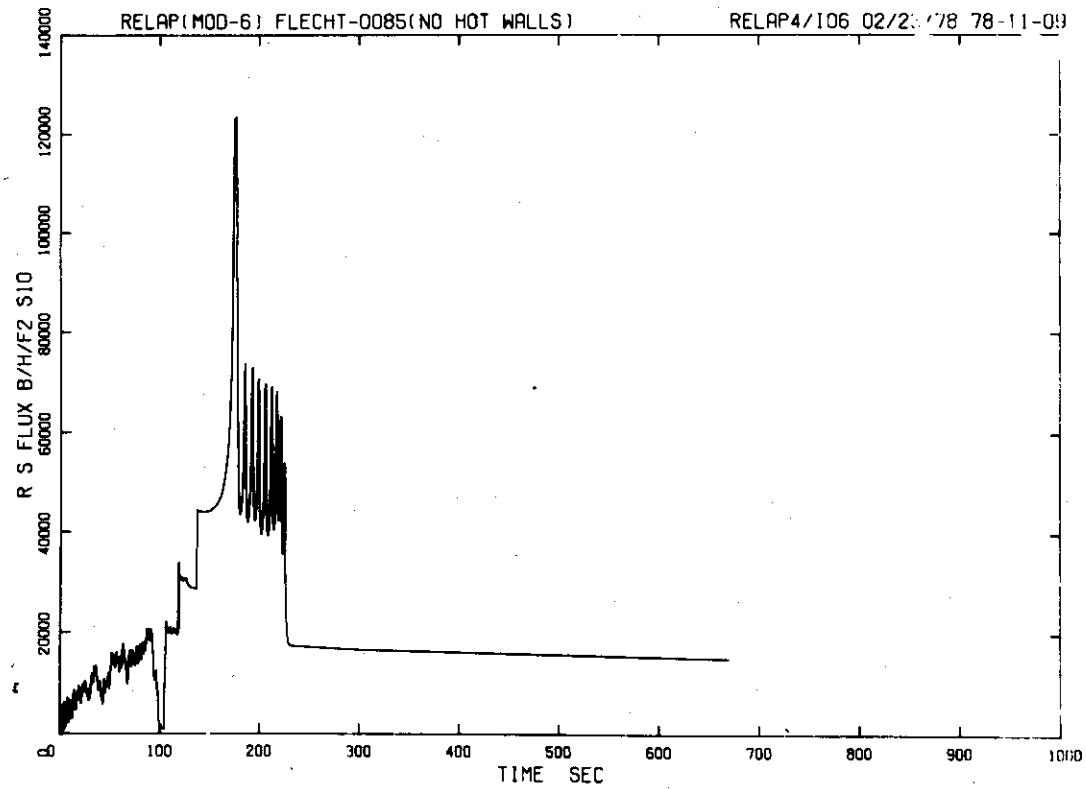
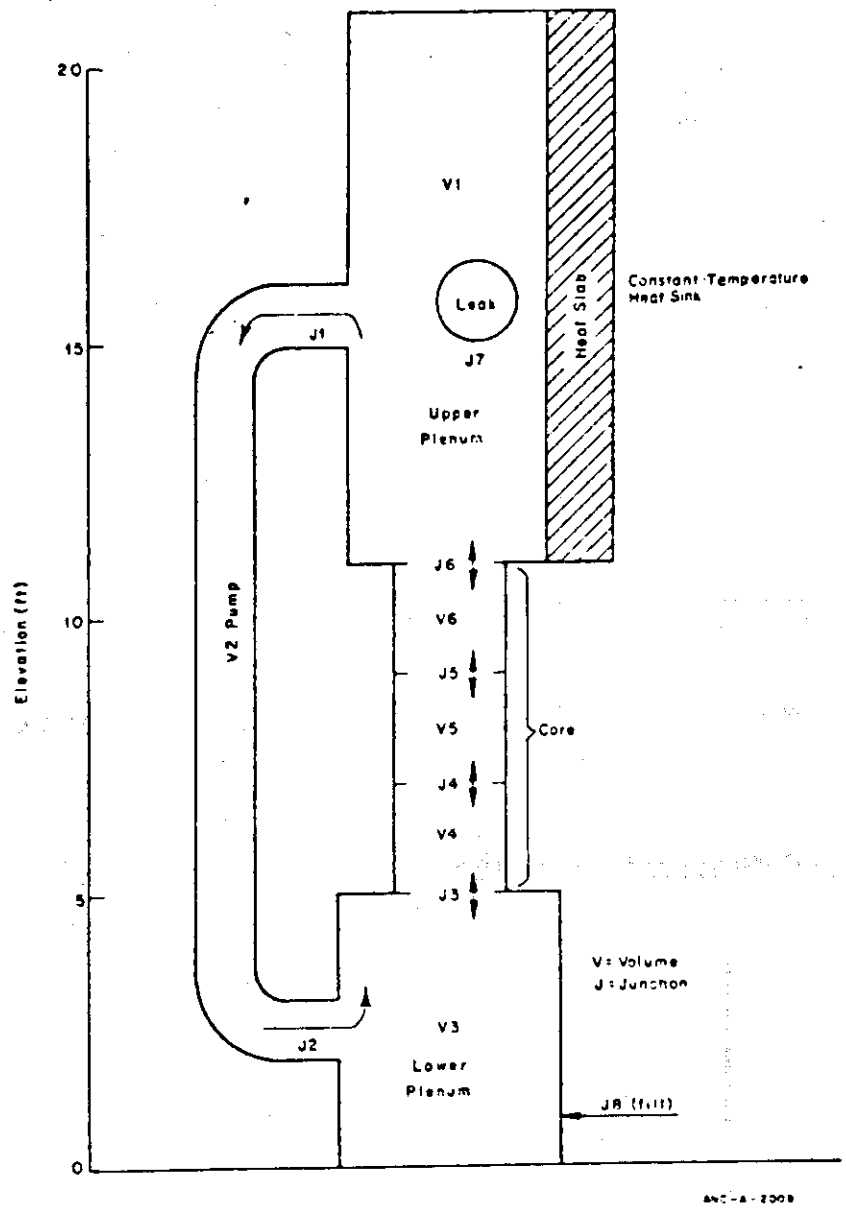


図 6.9　　ヒートスラブ 10（燃料棒中間点）熱流束



Schematic of RELAP4 six-volume sample problem.

図6.10　6ボリューム問題ノード区分

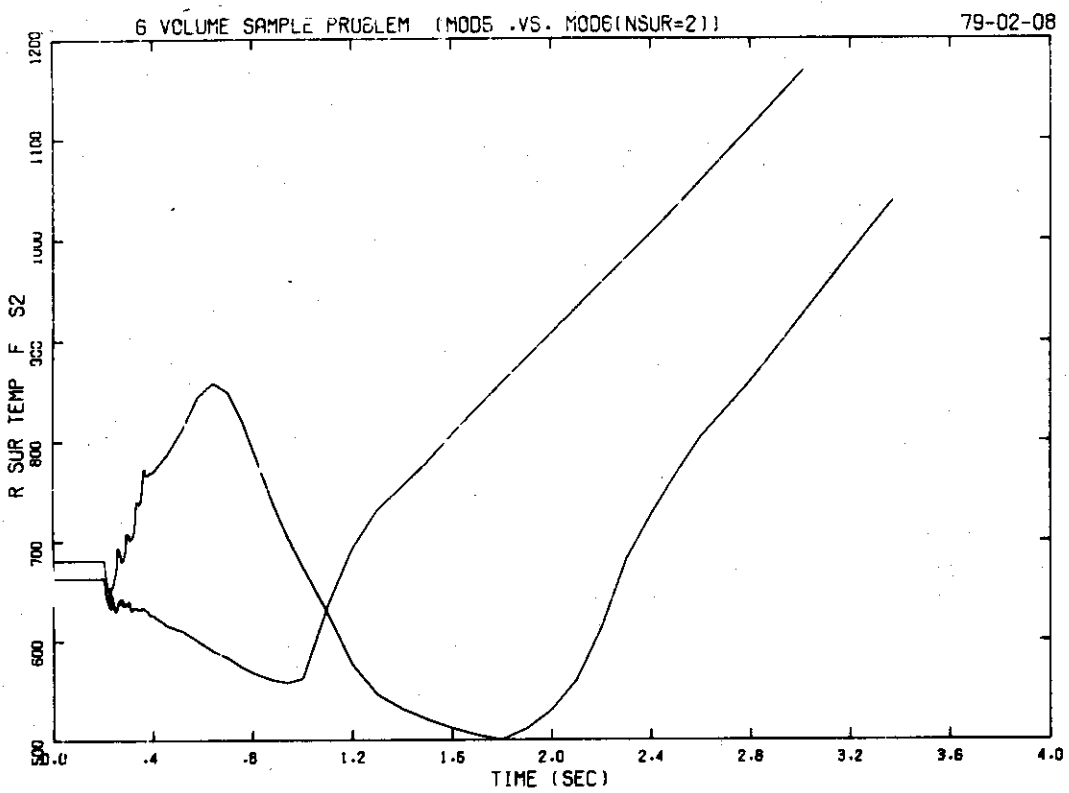


図 6.11　燃料棒表面温度

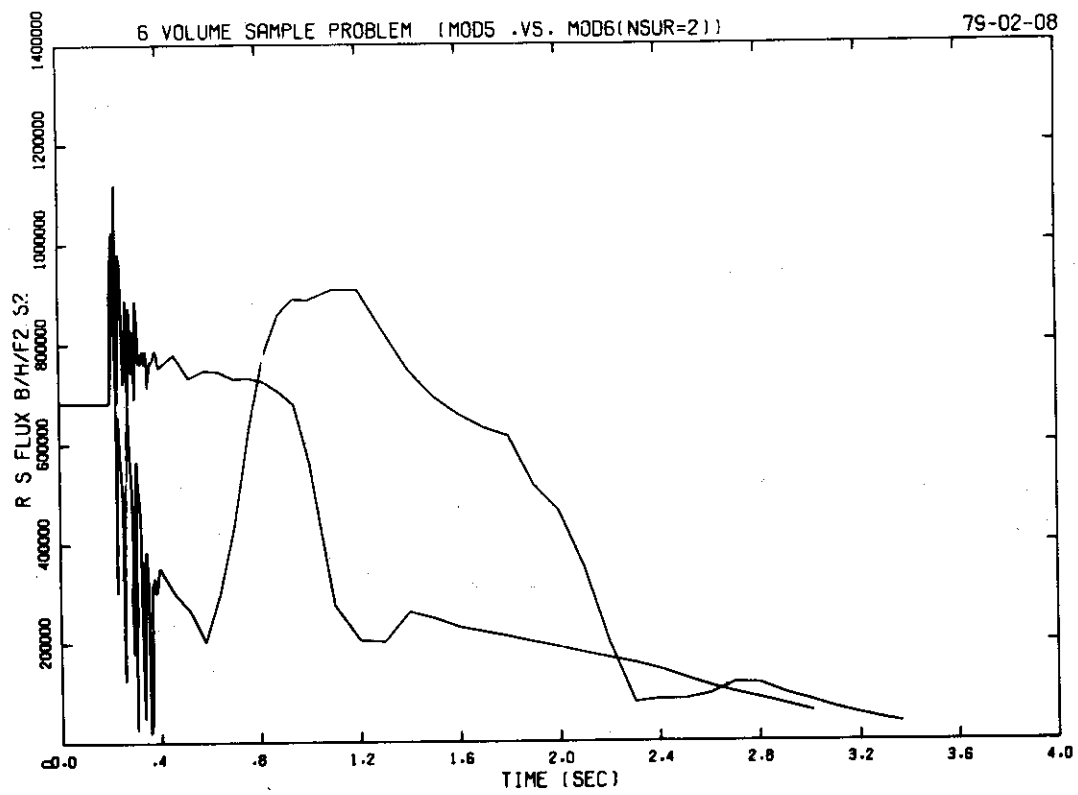


図 6.12　燃料棒表面熱流束

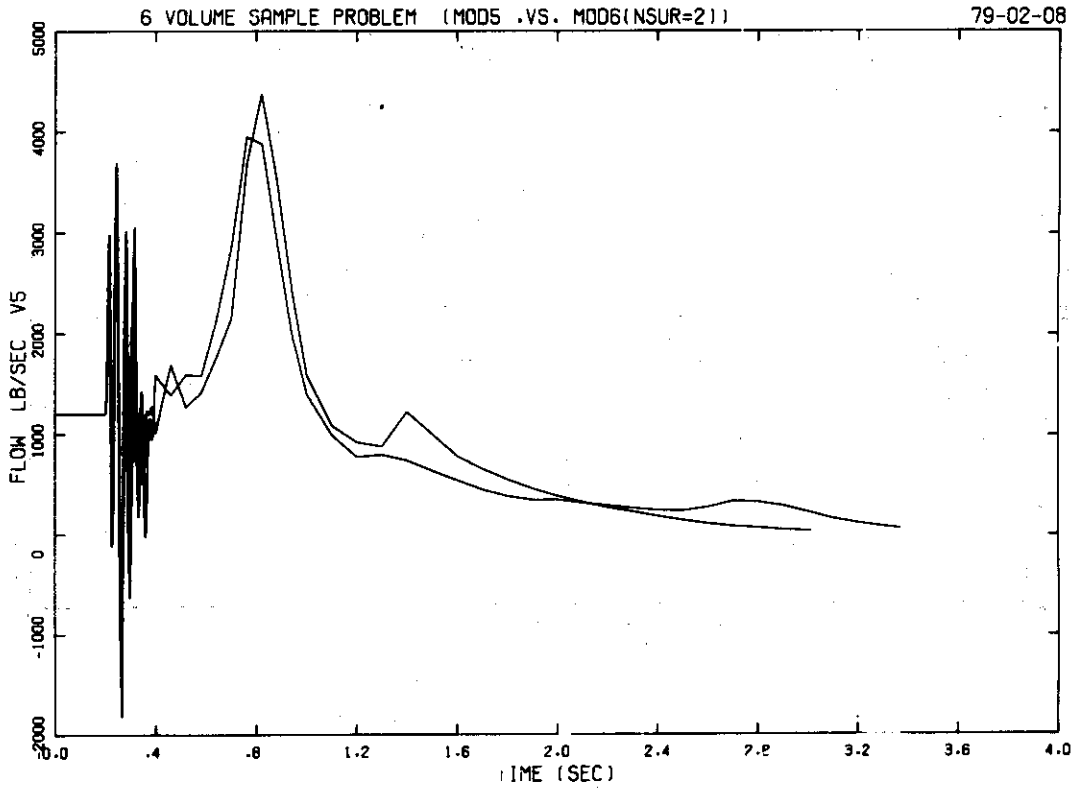

6 VOLUME SAMPLE PROBLEM (MOD5 .VS. MOD6(NSUR=2))
79-02-08
FLOW LB/SEC V5
5000
4000
3000
2000
1000
0
-1000
-2000
0.0
.4
.8
1.2
1.6
2.0
2.4
2.8
3.2
3.6
4.0
TIME (SEC)

図 6.13　炉心平均流量

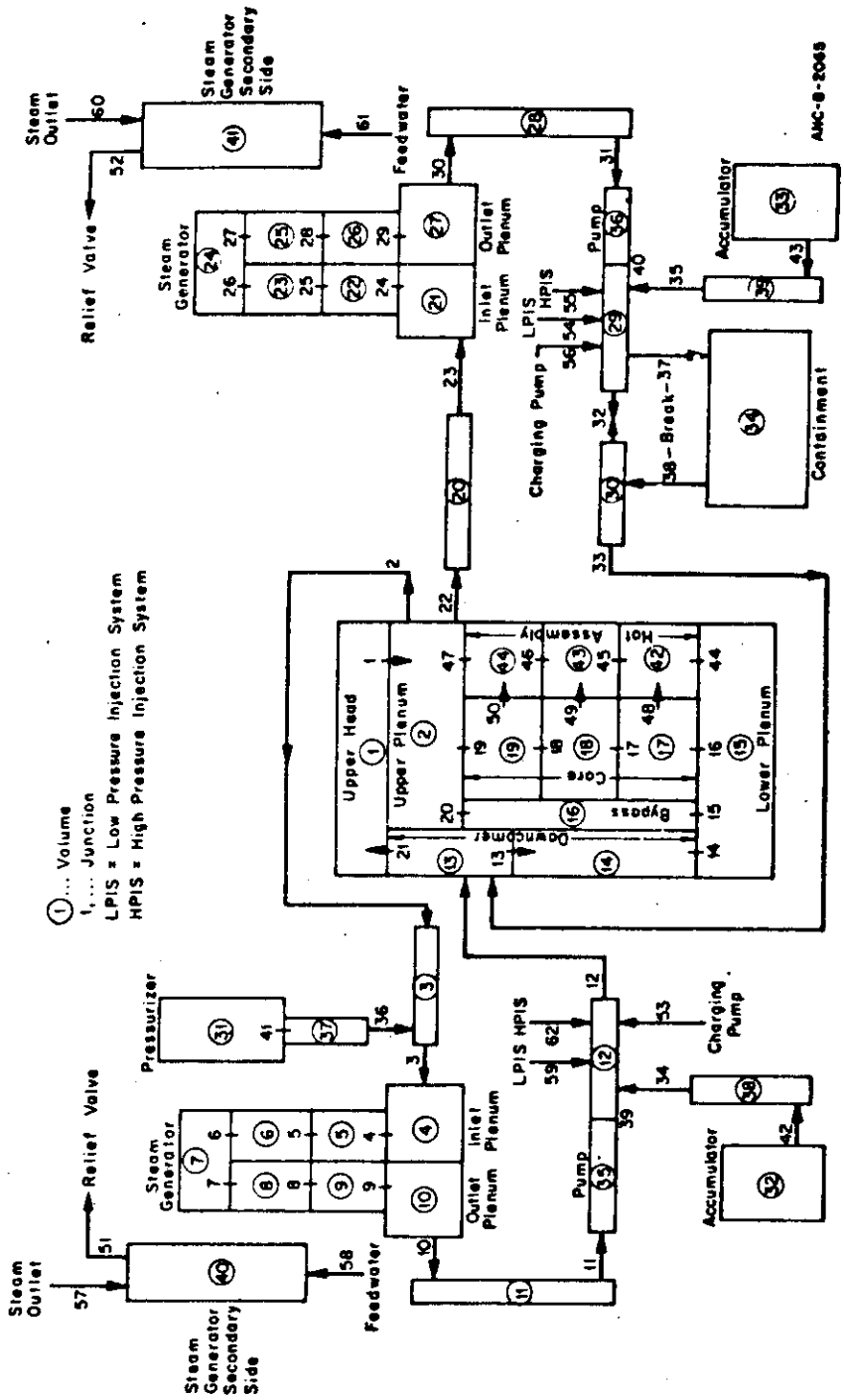


図6.14 PWR Flood System Model for RELAP4-EM Base Case Calculation

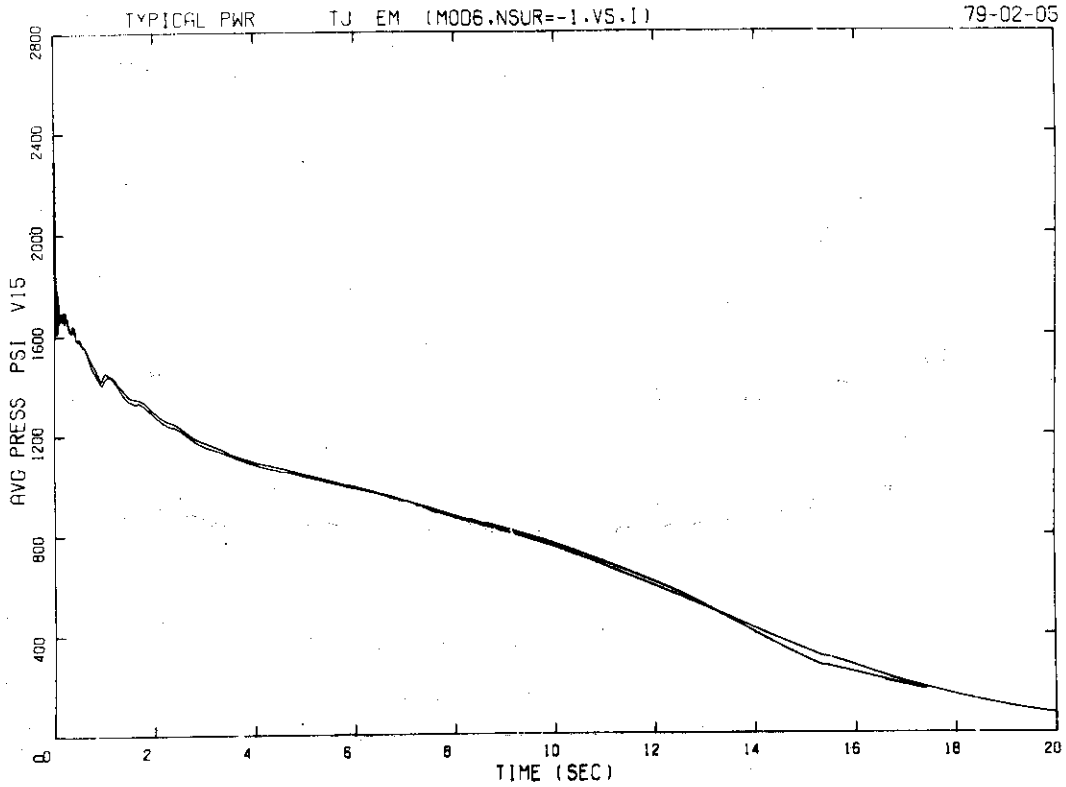


図6.15 下部プレナム圧力

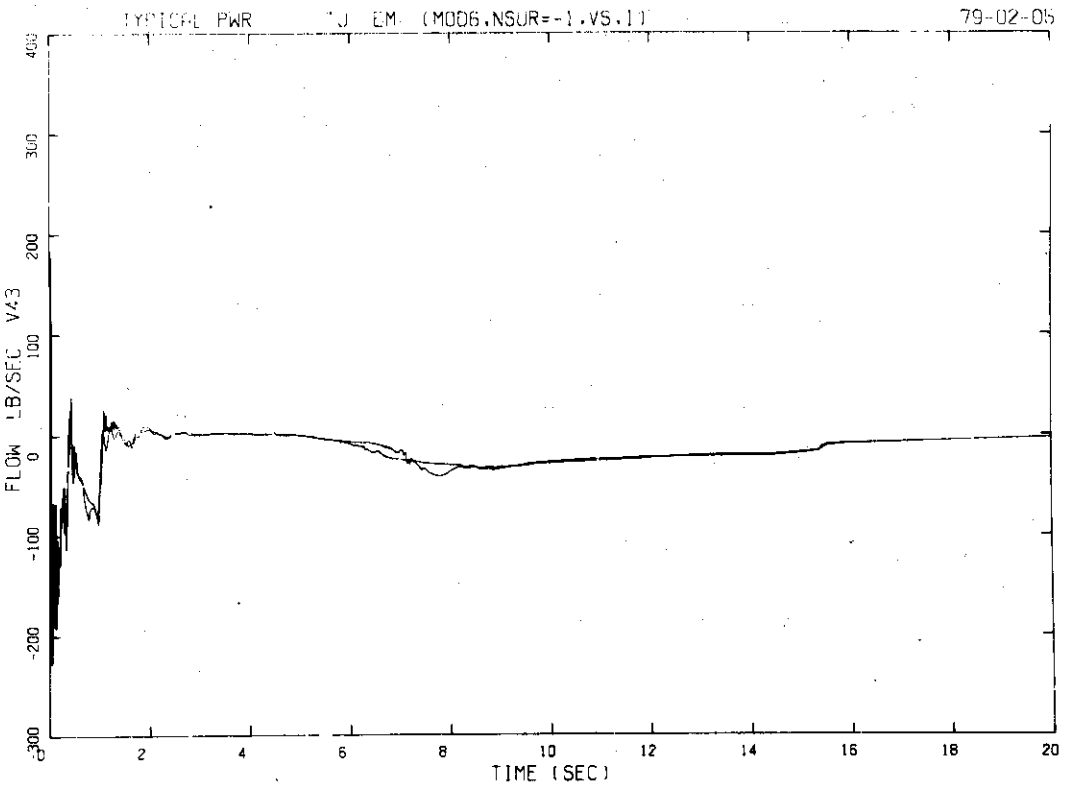


図6.16 炉心流量

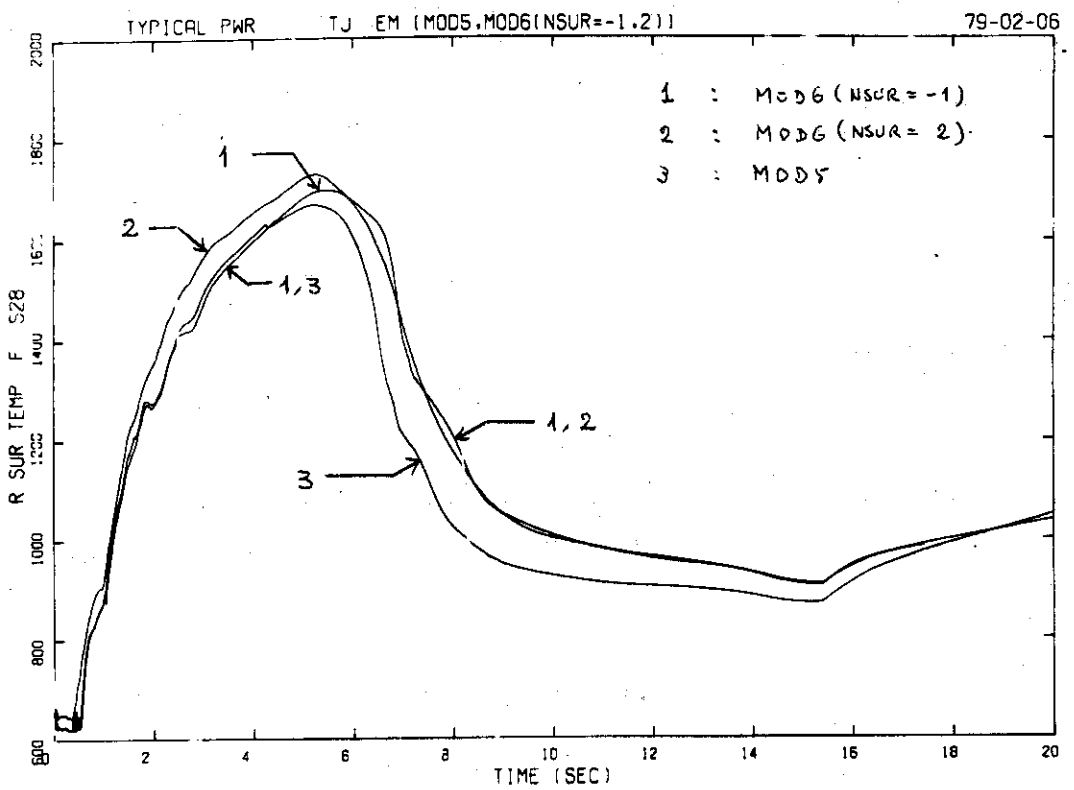


図 6. 17　ホットアッセンブリ軸方向中間燃料棒表面温度 .

謝　辞

MOD6の整備に関する経験のまとめにおいて，計算センターの方々の御協力を得た。また，テストランの結果の検討では，安全性コード開発室の方々に御協力頂いた。特に，MOD5とMOD6の比較検討では，原子炉データ解析室田辺文也氏に作業の一部を分担して頂いた。
合わせてここに謝意を表する。

参考文献

(1) "RELAP4/MOD5-A Computer Program for Transeient Thermal-Hydraulic Analysis of Nuclear Reactor and Related Systems"
ANCR-NUREG-1335, 1976.

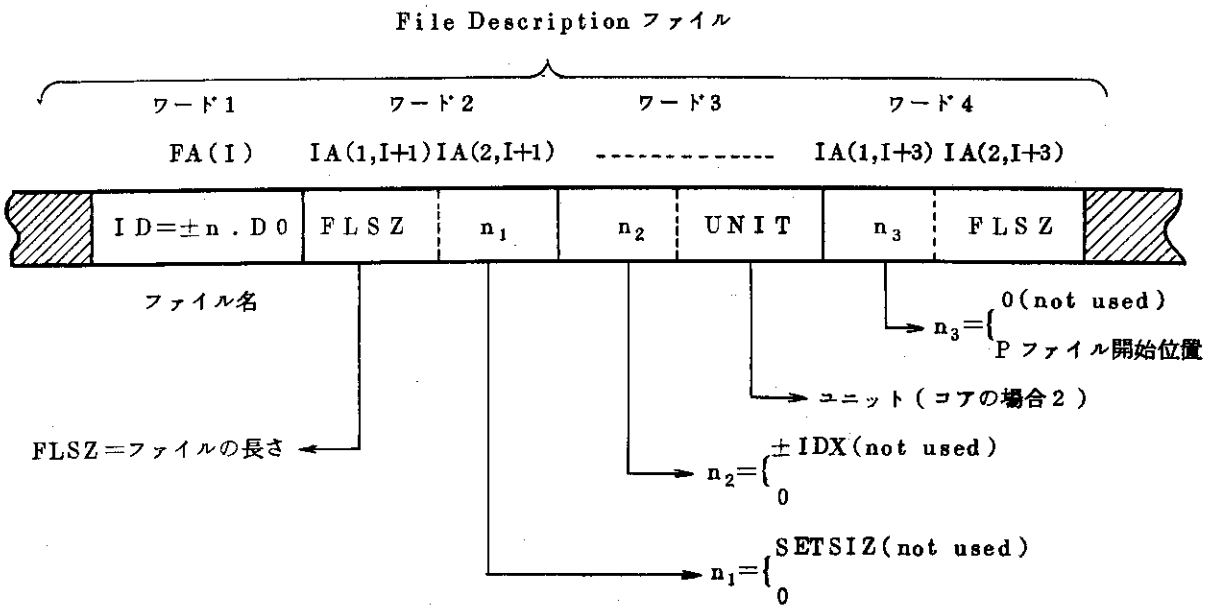
(2) "RELAP4/MOD6-A Computer Program for Transient Thermal-Hydraulic Analysis of Nuclear Reactor and Related Systems User's Manual"
EG&G, CDAP-TR-003, 1978.

(3) 過渡状態熱水力挙動解析コードRELAP-4の整備
（その1　FACOM 230/75システムへの変換）
JAERI-M 6623

付録1　File Descriptionフォーマットおよび，初期設定と割り付け／消去

1.1　File Descriptionフォーマット

作業領域上へのデータファイルの割り付けは，以下に示す4倍精度語で構成されるFile Descriptionを介して行われる。（それら自身1つの特別なファイルを構成する）



1.2　初期設定と割り付け／消去

File Descriptionファイルの初期設定と具体的なファイルの割り付け／消去の様子を，簡単なプログラムフローと対応づけて示す。

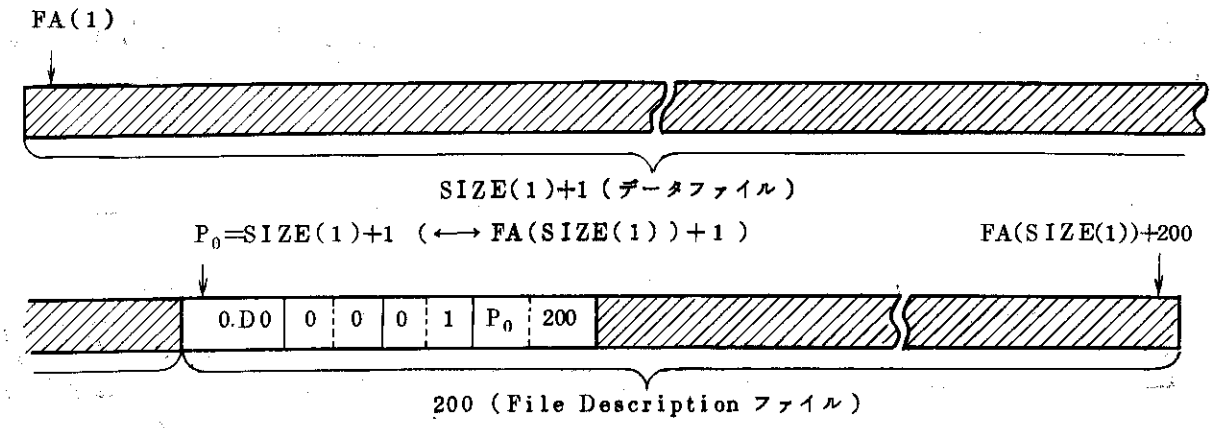
(1)　プログラムフローの例

```
step 1   CALL INITAL(2)
step 2   RESERV(ID1,FS1,2,J)
         ------------------------------------
         (step 2.1) CALL DSCRIB(ID1,FS1,1,U)
         (step 2.2) CALL IDFIND(ID1,I1)
                    IA(2,I1+1)=0
         (step 2.3) CALL LOCATE(U,FS1,I2)
                    IA(1,I1+3)=I2
                    IA(2,I1+3)=FS1
         ------------------------------------
step 3   CALL RESERV(ID2,FS2,2,J)
         ---------
step 4   CALL DELETE(IDm)
         ---------
```

$$\left(\begin{array}{ll} ID_1, ID_2, \cdots\cdots, ID_m & : \text{ファイル名} \\ FS_1, FS_2 & : \text{ファイルサイズ} \\ U & : \text{ユニット} \end{array}\right)$$

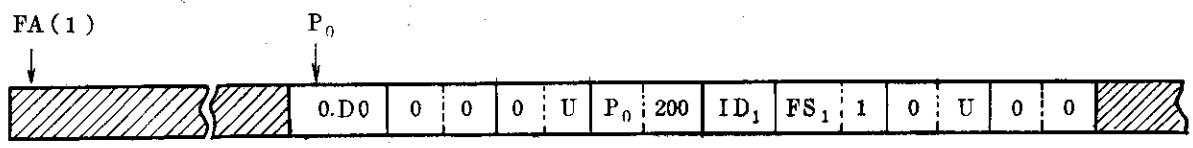
(2) 初期設定と割り付け／消去

Step 1 CALL INITAL(2)



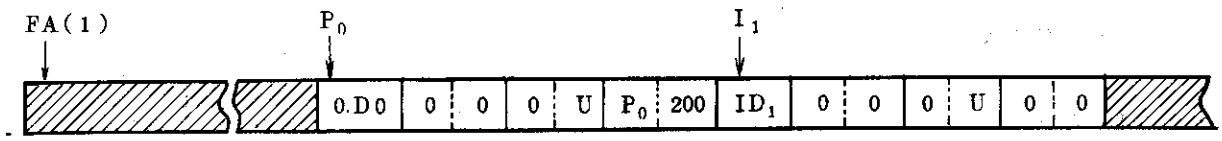
Step 2 CALL RESERV(ID_1, FS_1, 2, J)

(step 2.1) CALL DSCRIB(ID_1, FS_1, 1, U)



(step 2.2) CALL IDFIND(ID_1, I_1)

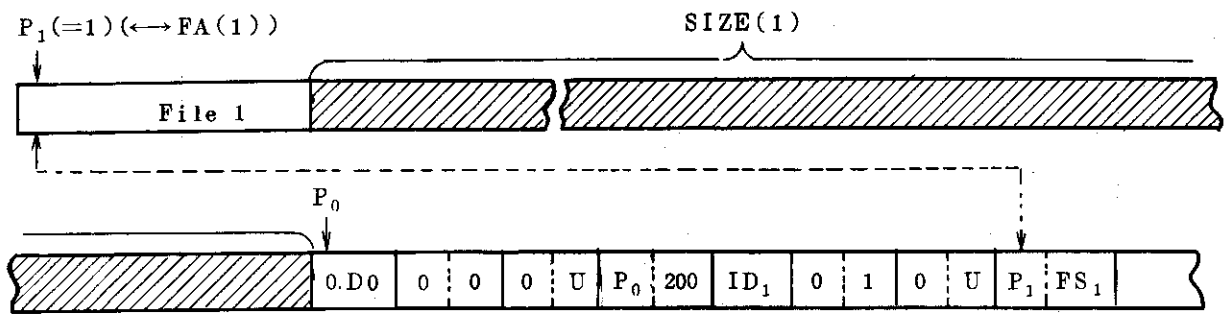
IA(2, I1+1)=0



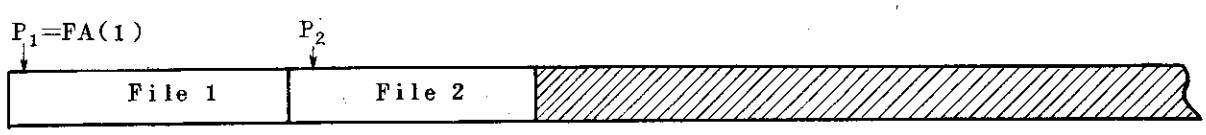
(step 2.3) CALL LOCATE(U, FS_1, I2)

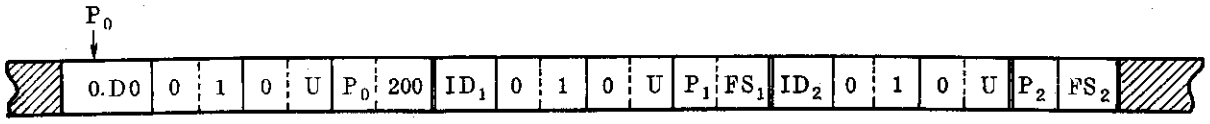
IA(1, I1+3)=I2

IA(2, I1+3)=FS_1



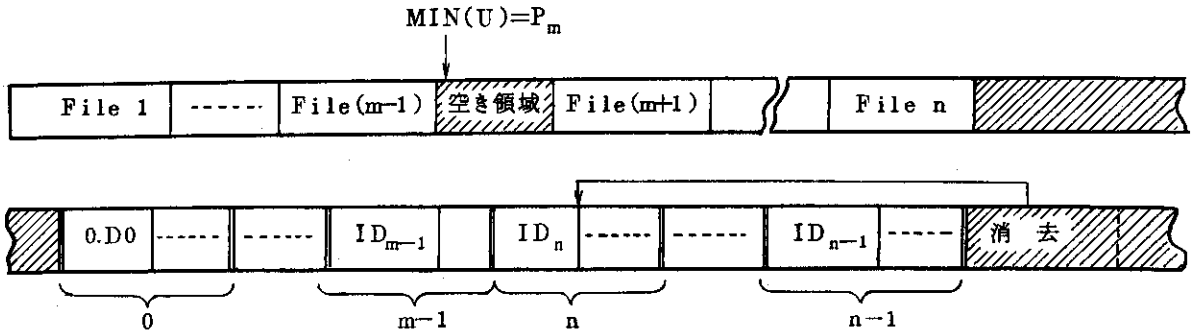
Step 3. CALL RESERV(ID_2, FS_2, 2, J)





Step 4. CALL DELETE(ID_m)

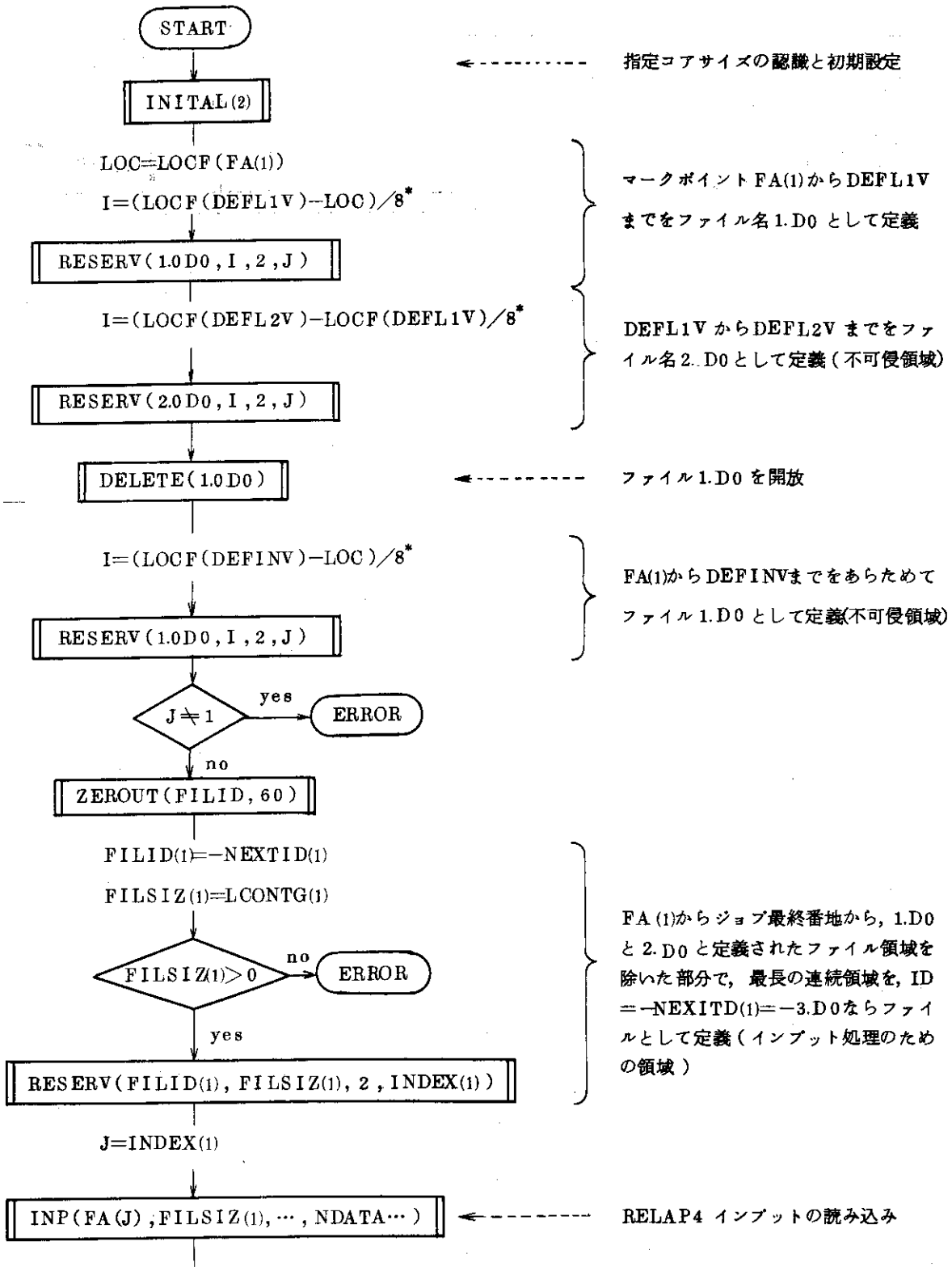
（ファイル ID_1 , ……, ID_{m-1} , ID_m , ……, ID_n から ID_m の消去）

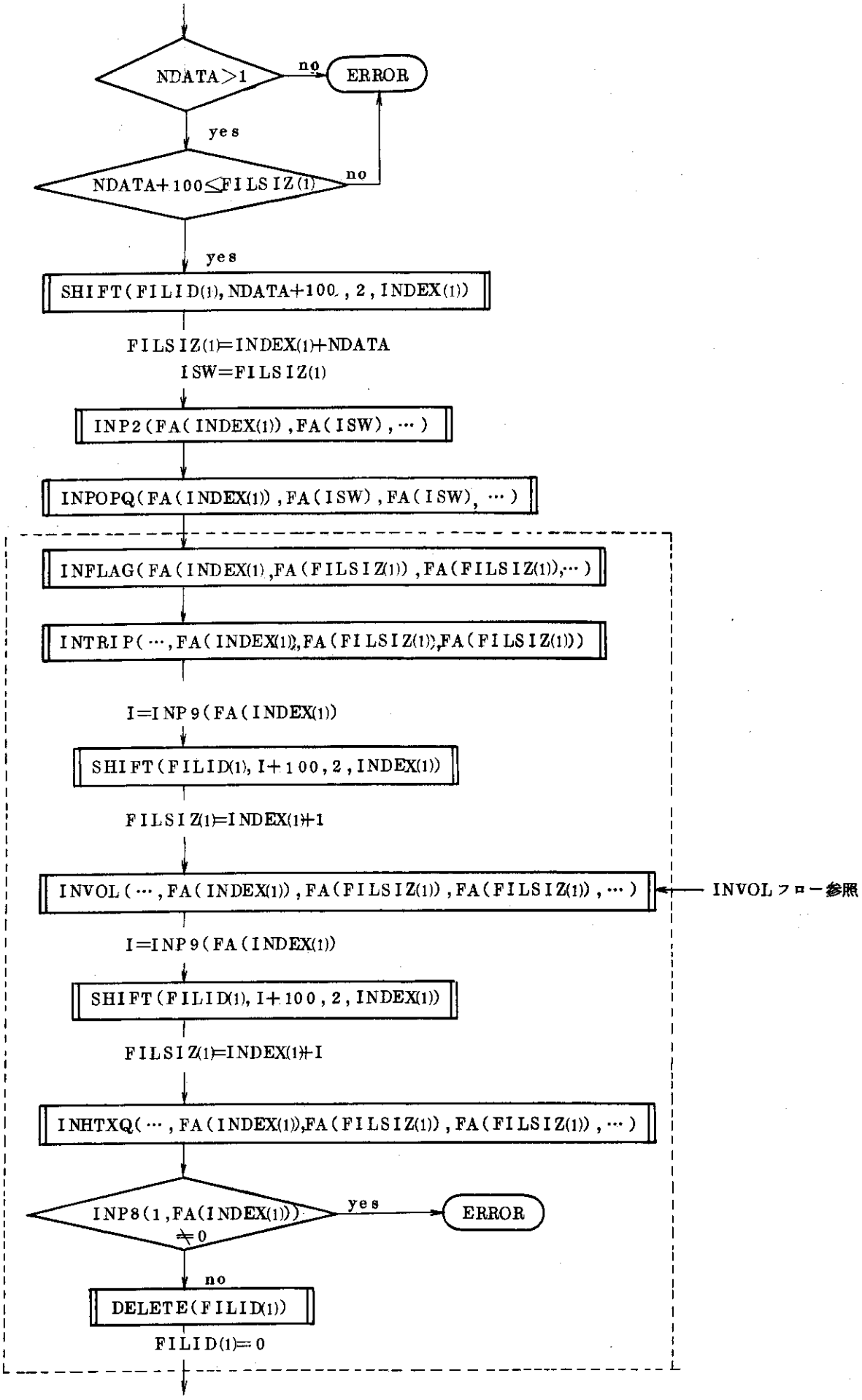


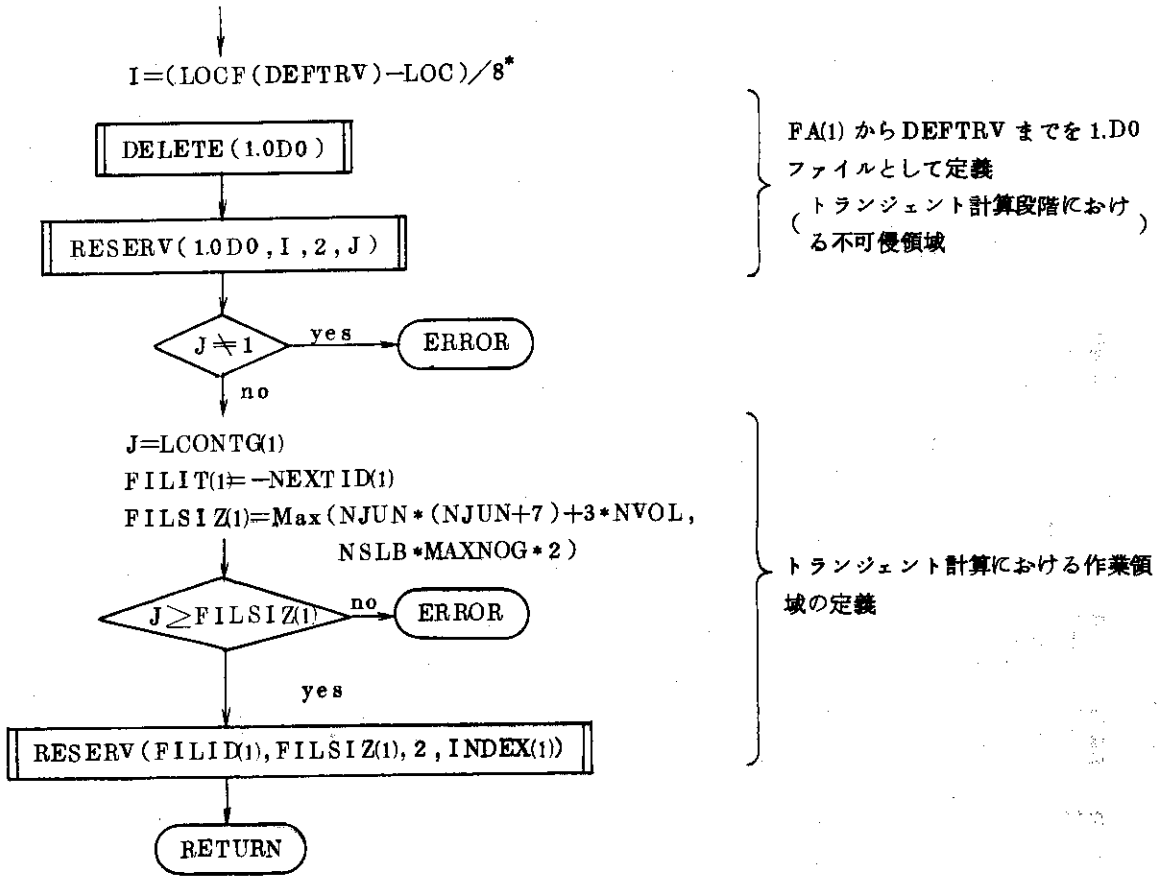
付録2. データファイルの割り付けフロー（MOD5）

RELAP4本体のサブルーチンINPUT，INVOL，INICE，INFILL におけるファイルの割り付け／消去のフローを示す。尚，フロー中，＊が印されている8は，IBM（8バイト）の場合であり，FACOMでは2となる。

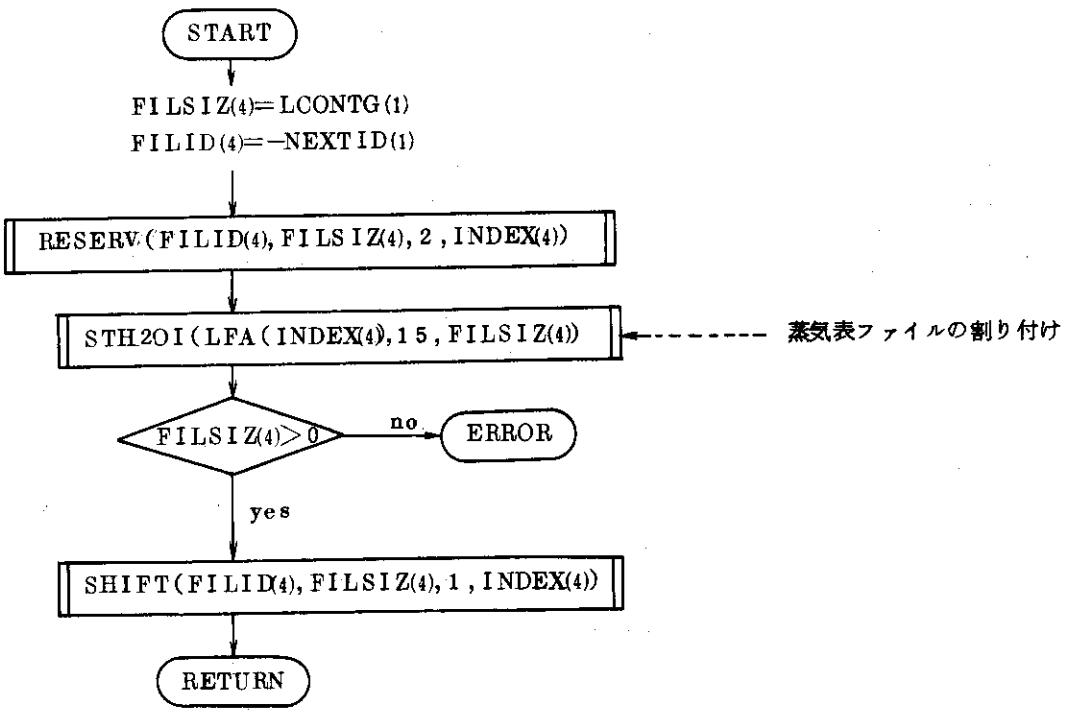
(1) INPUT



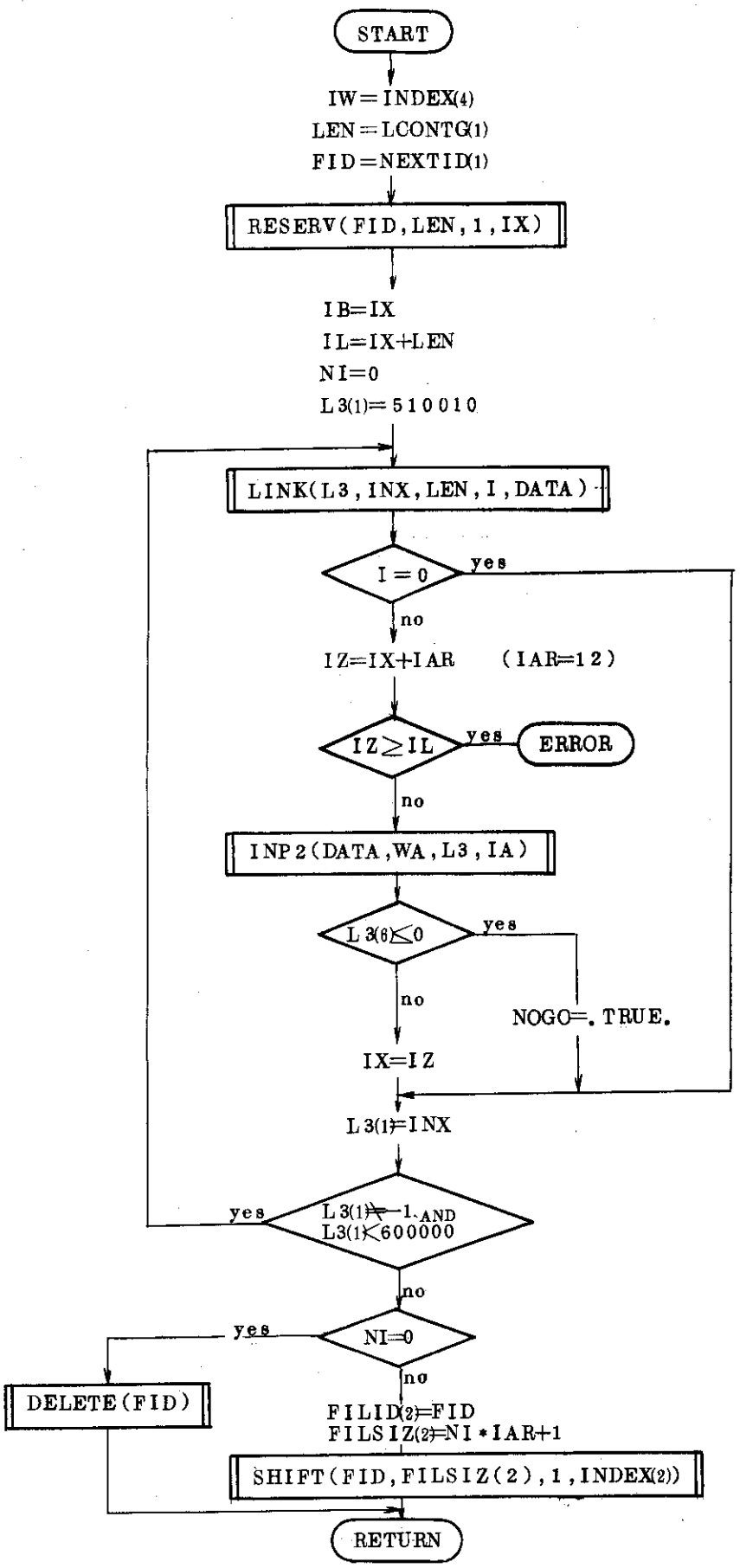

NDATA>1
no
ERROR
yes
NDATA+100≦FILSIZ(1)
no
yes
SHIFT(FILID(1), NDATA+100, 2, INDEX(1))
FILSIZ(1)=INDEX(1)+NDATA
ISW=FILSIZ(1)
INP2(FA(INDEX(1)), FA(ISW), …)
INPOPQ(FA(INDEX(1)), FA(ISW), FA(ISW), …)
INFLAG(FA(INDEX(1)), FA(FILSIZ(1)), FA(FILSIZ(1)), …)
INTRIP(…, FA(INDEX(1)), FA(FILSIZ(1)), FA(FILSIZ(1)))
I=INP9(FA(INDEX(1)))
SHIFT(FILID(1), I+100, 2, INDEX(1))
FILSIZ(1)=INDEX(1)+1
INVOL(…, FA(INDEX(1)), FA(FILSIZ(1)), FA(FILSIZ(1)), …)
INVOLフロー参照
I=INP9(FA(INDEX(1)))
SHIFT(FILID(1), I+100, 2, INDEX(1))
FILSIZ(1)=INDEX(1)+I
INHTXQ(…, FA(INDEX(1)), FA(FILSIZ(1)), FA(FILSIZ(1)), …)
INP8(1, FA(INDEX(1)))
≠0
yes
ERROR
no
DELETE(FILID(1))
FILID(1)=0


I=(LOCF(DEFTRV)−LOC)/8*
DELETE(1.0D0)
RESERV(1.0D0,I,2,J)
J≠1
yes
ERROR
no
J=LCONTG(1)
FILIT(1)=−NEXTID(1)
FILSIZ(1)=Max(NJUN*(NJUN+7)+3*NVOL,
NSLB*MAXNOG*2)
J≥FILSIZ(1)
no
ERROR
yes
RESERV(FILID(1),FILSIZ(1),2,INDEX(1))
RETURN
FA(1) からDEFTRV までを 1.D0
ファイルとして定義
(トランジェント計算段階における不可侵領域)
トランジェント計算における作業領域の定義

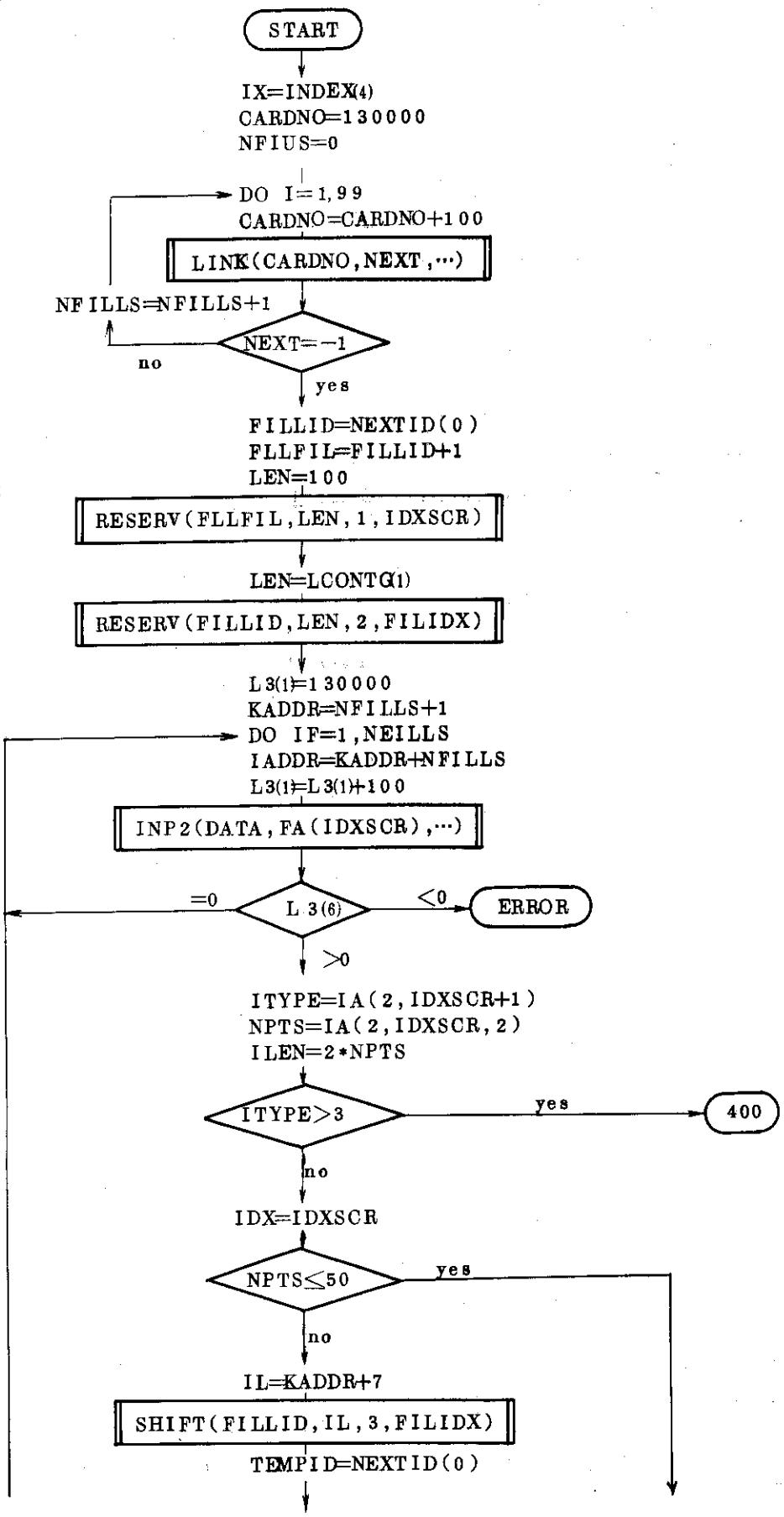
(2) INVOL

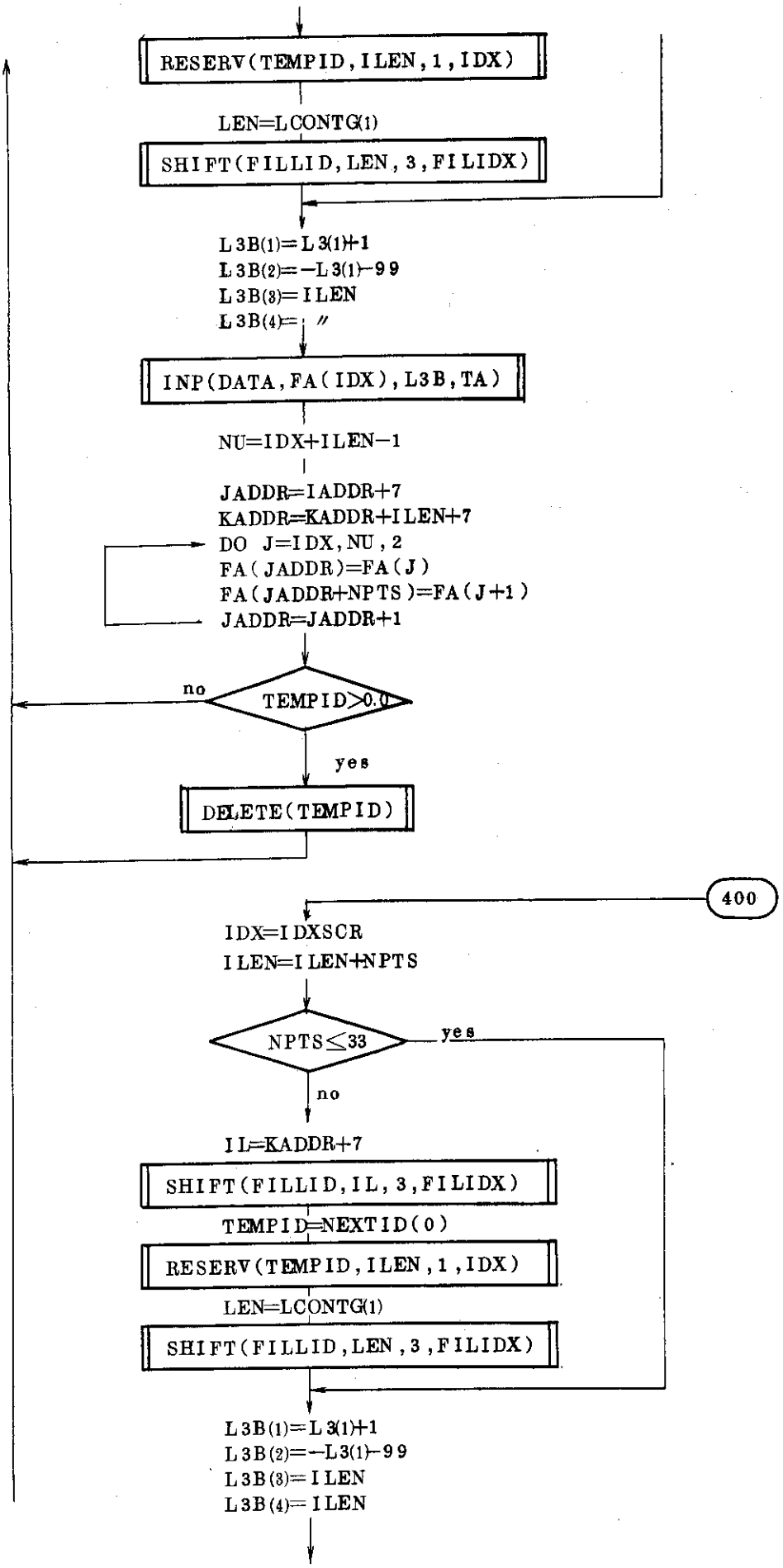

START
FILSIZ(4)=LCONTG(1)
FILID(4)=−NEXTID(1)
RESERV(FILID(4),FILSIZ(4),2,INDEX(4))
STH2OI(LFA(INDEX(4)),15,FILSIZ(4))
蒸気表ファイルの割り付け
FILSIZ(4)>0
no
ERROR
yes
SHIFT(FILID(4),FILSIZ(4),1,INDEX(4))
RETURN

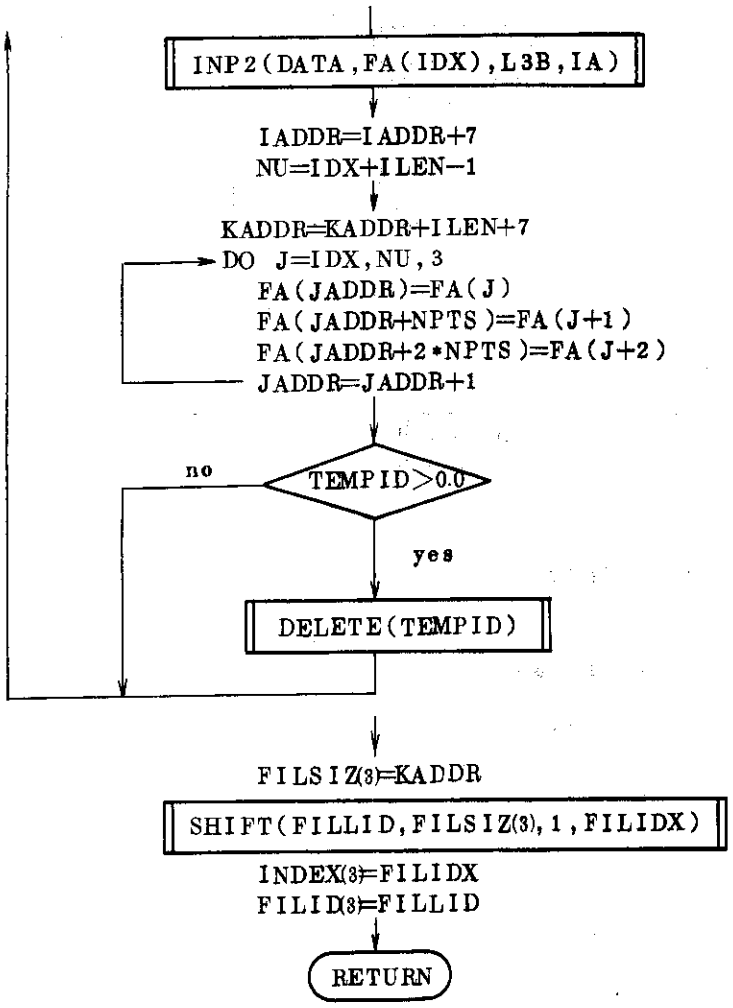
(3) INICE


START
IW=INDEX(4)
LEN=LCONTG(1)
FID=NEXTID(1)
RESERV(FID,LEN,1,IX)
IB=IX
IL=IX+LEN
NI=0
L3(1)=510010
LINK(L3,INX,LEN,I,DATA)
I=0
yes
no
IZ=IX+IAR (IAR=12)
IZ≥IL
yes
ERROR
no
INP2(DATA,WA,L3,IA)
L3(6)≤0
yes
no
NOGO=.TRUE.
IX=IZ
L3(1)=INX
yes
L3(1)≤-1 AND L3(1)<600000
no
yes
NI=0
no
DELETE(FID)
FILID(2)=FID
FILSIZ(2)=NI*IAR+1
SHIFT(FID,FILSIZ(2),1,INDEX(2))
RETURN

(4) INFILL

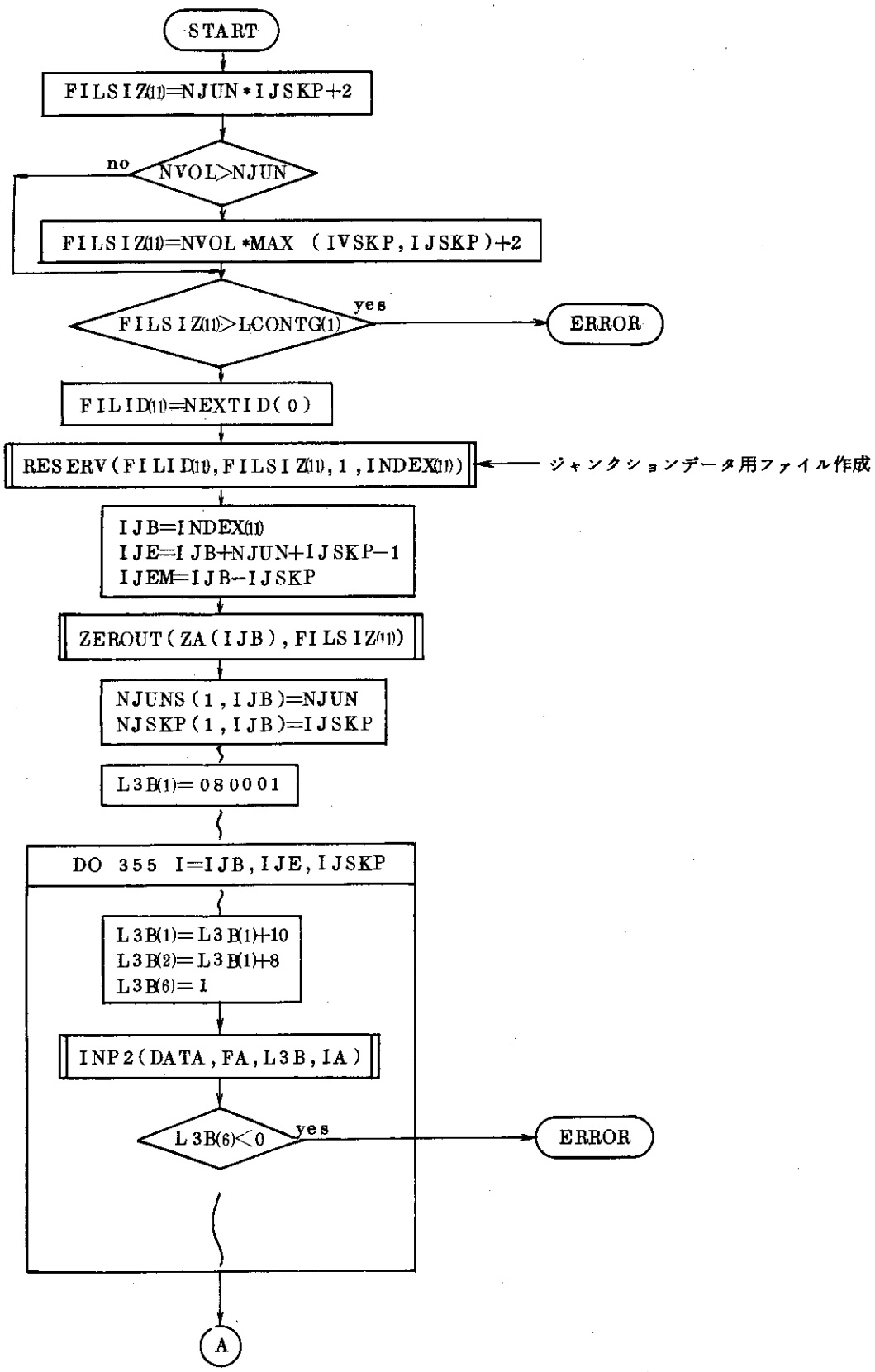

START
IX=INDEX(4)
CARDNO=130000
NFIUS=0
DO I=1,99
CARDNO=CARDNO+100
LINK(CARDNO,NEXT,…)
NFILLS=NFILLS+1
NEXT=−1
no
yes
FILLID=NEXTID(0)
FLLFIL=FILLID+1
LEN=100
RESERV(FLLFIL,LEN,1,IDXSCR)
LEN=LCONTG(1)
RESERV(FILLID,LEN,2,FILIDX)
L3(1)=130000
KADDR=NFILLS+1
DO IF=1,NEILLS
IADDR=KADDR+NFILLS
L3(1)=L3(1)+100
INP2(DATA,FA(IDXSCR),…)
=0
L 3(6)
<0
ERROR
>0
ITYPE=IA(2,IDXSCR+1)
NPTS=IA(2,IDXSCR,2)
ILEN=2*NPTS
ITYPE>3
yes
400
no
IDX=IDXSCR
NPTS≤50
yes
no
IL=KADDR+7
SHIFT(FILLID,IL,3,FILIDX)
TEMPID=NEXTID(0)

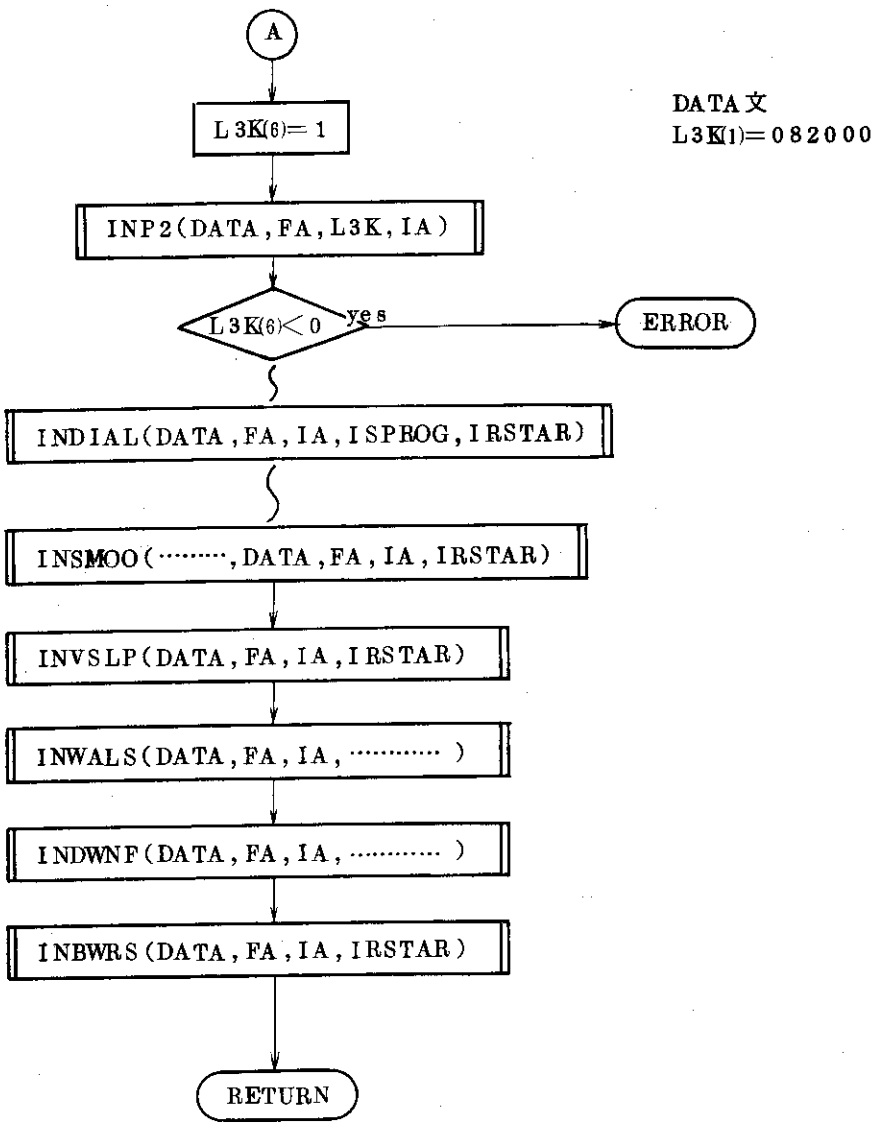

RESERV(TEMPID,ILEN,1,IDX)
LEN=LCONTG(1)
SHIFT(FILLID,LEN,3,FILIDX)
L3B(1)=L3(1)+1
L3B(2)=-L3(1)-99
L3B(3)=ILEN
L3B(4)=〃
INP(DATA,FA(IDX),L3B,TA)
NU=IDX+ILEN-1
JADDR=IADDR+7
KADDR=KADDR+ILEN+7
DO J=IDX,NU,2
FA(JADDR)=FA(J)
FA(JADDR+NPTS)=FA(J+1)
JADDR=JADDR+1
TEMPID>0.0
no
yes
DELETE(TEMPID)
400
IDX=IDXSCR
ILEN=ILEN+NPTS
NPTS≤33
yes
no
IL=KADDR+7
SHIFT(FILLID,IL,3,FILIDX)
TEMPID=NEXTID(0)
RESERV(TEMPID,ILEN,1,IDX)
LEN=LCONTG(1)
SHIFT(FILLID,LEN,3,FILIDX)
L3B(1)=L3(1)+1
L3B(2)=-L3(1)-99
L3B(3)=ILEN
L3B(4)=ILEN


INP2(DATA,FA(IDX),L3B,IA)
IADDR=IADDR+7
NU=IDX+ILEN-1
KADDR=KADDR+ILEN+7
DO J=IDX,NU,3
FA(JADDR)=FA(J)
FA(JADDR+NPTS)=FA(J+1)
FA(JADDR+2*NPTS)=FA(J+2)
JADDR=JADDR+1
no
TEMPID>0.0
yes
DELETE(TEMPID)
FILSIZ(3)=KADDR
SHIFT(FILLID,FILSIZ(3),1,FILIDX)
INDEX(3)=FILIDX
FILID(3)=FILLID
RETURN

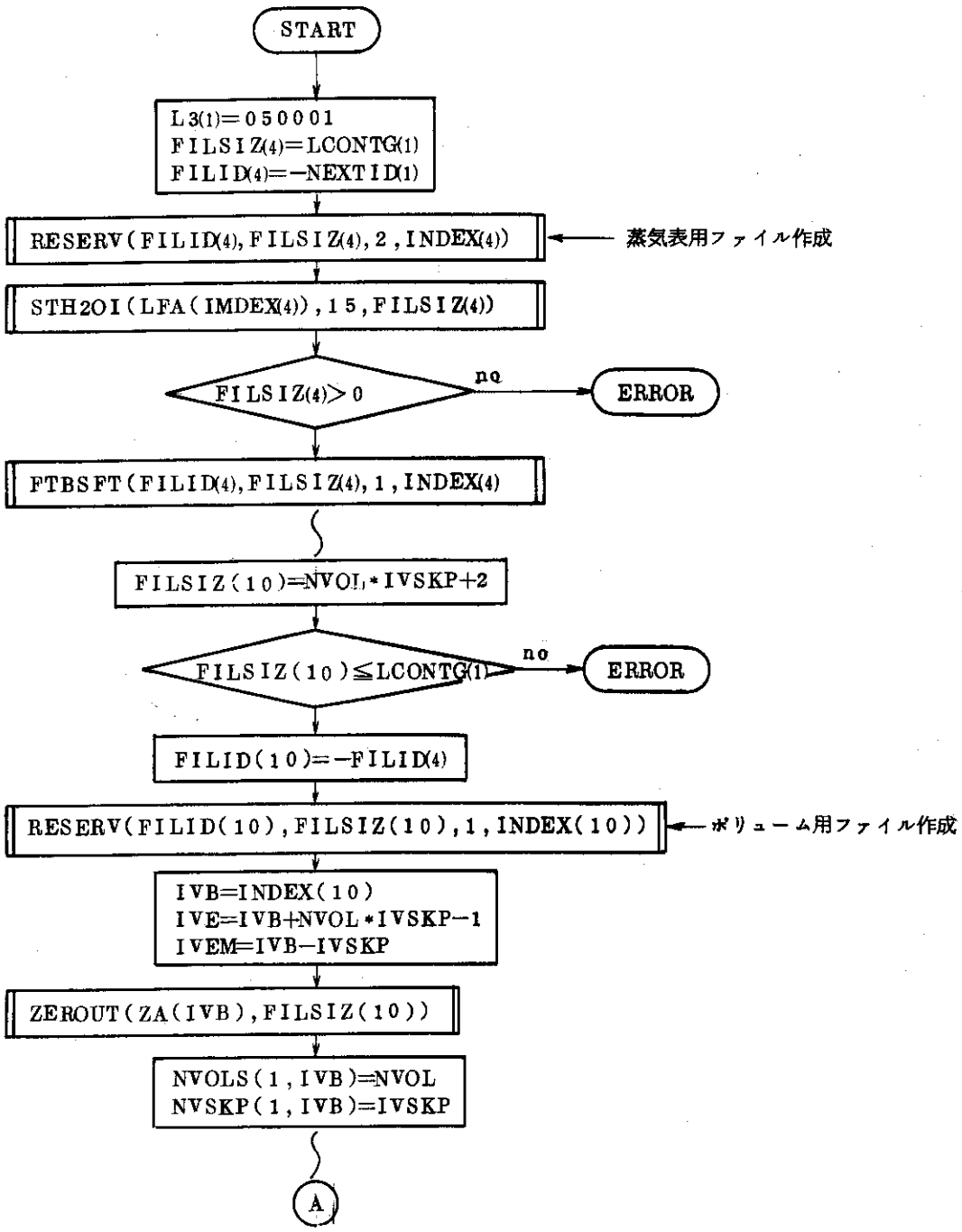
付録 3. データ・ファイルの割付けフロー（MOD 6）

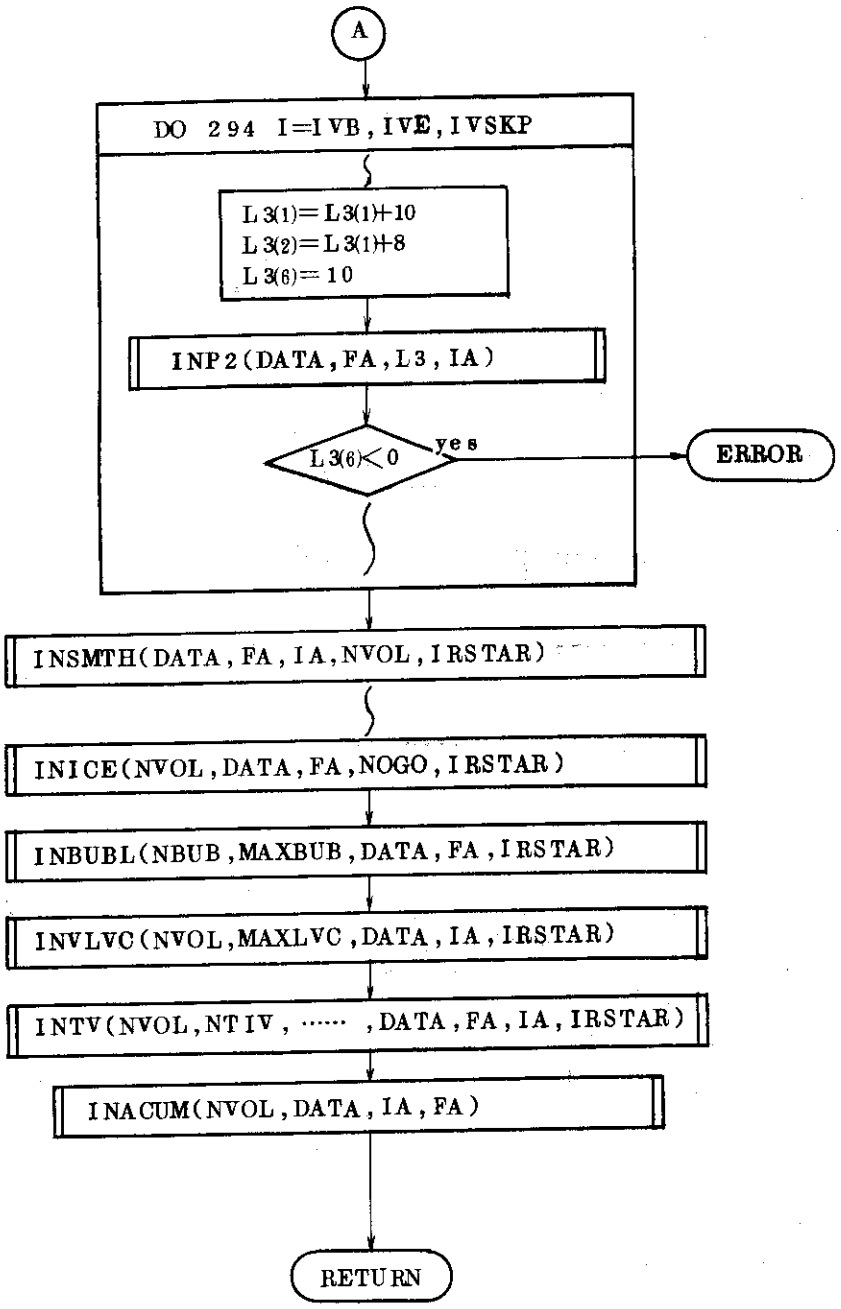
1. INJUN1



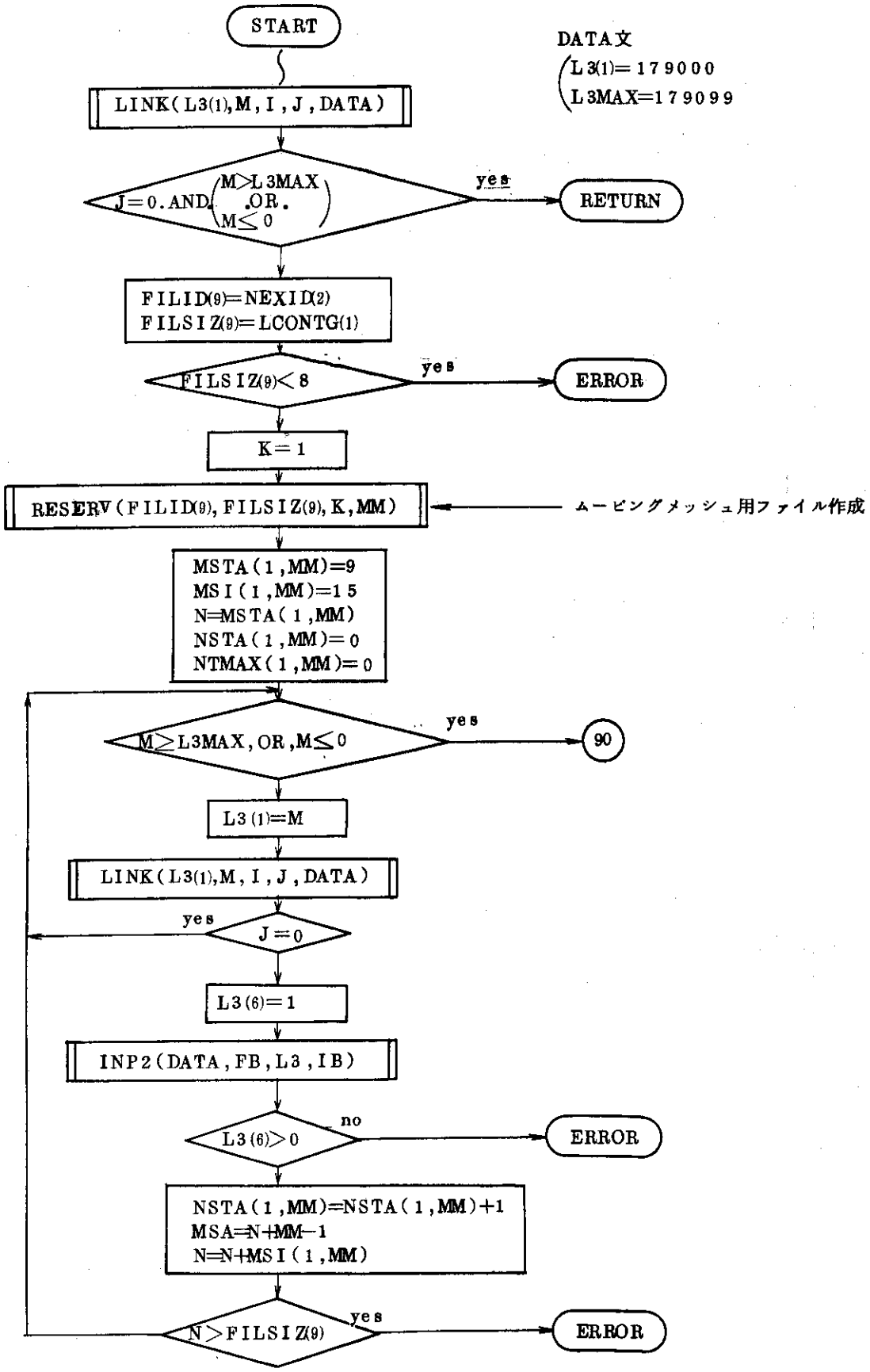

A
L3K(6)= 1
INP2(DATA,FA,L3K,IA)
L3K(6)< 0
yes
ERROR
INDIAL(DATA,FA,IA,ISPROG,IRSTAR)
INSMOO(·····,DATA,FA,IA,IRSTAR)
INVSLP(DATA,FA,IA,IRSTAR)
INWALS(DATA,FA,IA,·····)
INDWNF(DATA,FA,IA,·····)
INBWRS(DATA,FA,IA,IRSTAR)
RETURN
DATA文
L3K(1)=082000

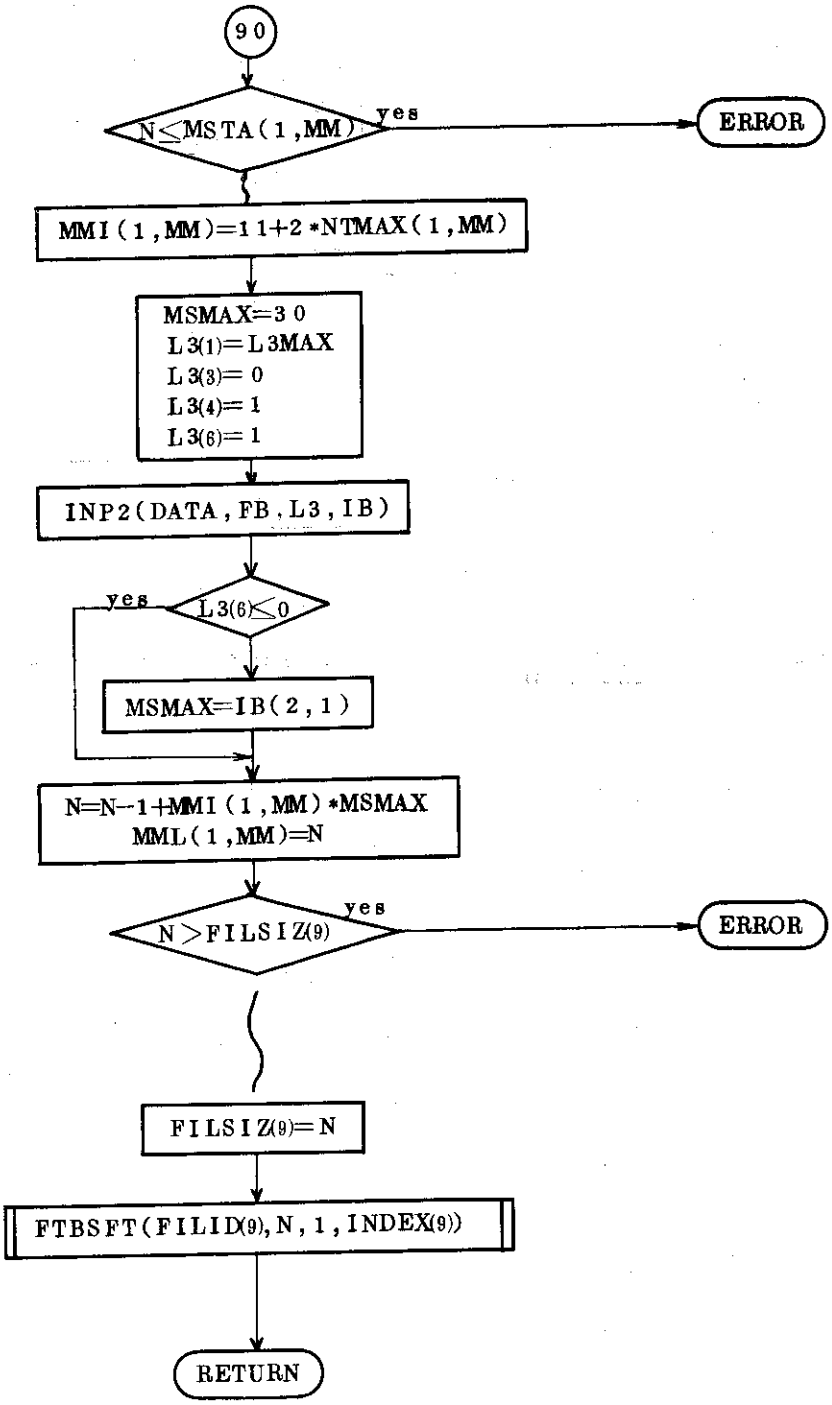
2. INVOL


START
L3(1)=050001
FILSIZ(4)=LCONTG(1)
FILID(4)=−NEXTID(1)
RESERV(FILID(4),FILSIZ(4),2,INDEX(4))
蒸気表用ファイル作成
STH2OI(LFA(IMDEX(4)),15,FILSIZ(4))
FILSIZ(4)>0
no
ERROR
FTBSFT(FILID(4),FILSIZ(4),1,INDEX(4))
FILSIZ(10)=NVOL*IVSKP+2
FILSIZ(10)≦LCONTG(1)
no
ERROR
FILID(10)=−FILID(4)
RESERV(FILID(10),FILSIZ(10),1,INDEX(10))
ボリューム用ファイル作成
IVB=INDEX(10)
IVE=IVB+NVOL*IVSKP−1
IVEM=IVB−IVSKP
ZEROUT(ZA(IVB),FILSIZ(10))
NVOLS(1,IVB)=NVOL
NVSKP(1,IVB)=IVSKP
A


A
DO 294 I=IVB, IVE, IVSKP
L3(1)=L3(1)+10
L3(2)=L3(1)+8
L3(6)=10
INP2(DATA, FA, L3, IA)
L3(6)<0
yes
ERROR
INSMTH(DATA, FA, IA, NVOL, IRSTAR)
INICE(NVOL, DATA, FA, NOGO, IRSTAR)
INBUBL(NBUB, MAXBUB, DATA, FA, IRSTAR)
INVLVC(NVOL, MAXLVC, DATA, IA, IRSTAR)
INTV(NVOL, NTIV, ····· , DATA, FA, IA, IRSTAR)
INACUM(NVOL, DATA, IA, FA)
RETURN

3. INMOVE


START
DATA文
L3(1)=179000
L3MAX=179099
LINK(L3(1),M,I,J,DATA)
J=0.AND.(M>L3MAX .OR. M≤0)
yes
RETURN
FILID(9)=NEXID(2)
FILSIZ(9)=LCONTG(1)
FILSIZ(9)<8
yes
ERROR
K=1
RESERV(FILID(9),FILSIZ(9),K,MM)
ムービングメッシュ用ファイル作成
MSTA(1,MM)=9
MSI(1,MM)=15
N=MSTA(1,MM)
NSTA(1,MM)=0
NTMAX(1,MM)=0
M≥L3MAX,OR,M≤0
yes
90
L3(1)=M
LINK(L3(1),M,I,J,DATA)
yes
J=0
L3(6)=1
INP2(DATA,FB,L3,IB)
L3(6)>0
no
ERROR
NSTA(1,MM)=NSTA(1,MM)+1
MSA=N+MM-1
N=N+MSI(1,MM)
N>FILSIZ(9)
yes
ERROR


90
N≤MSTA(1,MM)
yes
ERROR
MMI(1,MM)=11+2*NTMAX(1,MM)
MSMAX=30
L3(1)=L3MAX
L3(3)=0
L3(4)=1
L3(6)=1
INP2(DATA,FB,L3,IB)
yes
L3(6)≤0
MSMAX=IB(2,1)
N=N-1+MMI(1,MM)*MSMAX
MML(1,MM)=N
yes
N>FILSIZ(9)
ERROR
FILSIZ(9)=N
FTBSFT(FILID(9),N,1,INDEX(9))
RETURN

付録4　FTB基本ルーチンソートリスト

```
      BLOCK DATA                                                        FTB 0010
      COMMON /FAST/ A(1)                                                FTB 0020
      REAL*8 A                                                          FTB 0030
      INTEGER IA(2,1)                                                   FTB 0040
      EQUIVALENCE (A(1),IA(1,1))                                        FTB 0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       FTB 0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,        FTB 0070
     * SZZ                                                               FTB 0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   FTB 0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),             FTB 0090
     * SZZ(8)                                                            FTB 0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     FTB 0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      DATA FIRST/'****'/,SIZE/7*0/,NDSK2/7/                             FTB 0120
      END                                                               FTB 0130
      SUBROUTINE DELETE (ID)                                            DLET0010
      COMMON /FAST/ A(1)                                                DLET0020
      REAL*8 A                                                          DLET0030
      INTEGER IA(2,1)                                                   DLET0040
      EQUIVALENCE (A(1),IA(1,1))                                        DLET0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       DLET0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,        DLET0070
     * SZZ                                                               DLET0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   DLET0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),             DLET0090
     * SZZ(8)                                                            DLET0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     DLET0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      REAL*8 ID                                                         DLET0120
      INTEGER UNIT                                                      DLET0130
      IF (ID .EQ. 0.D0) CALL ERROR (19)                                 DLET0150
      CALL IDFIND (ID,I1)                                               DLET0160
      IF (I1 .EQ. 0) CALL ERROR (11)                                    DLET0170
      IF (IA(1,I1+2) .NE. 0) CALL ERROR (9)                             DLET0180
      UNIT = IA(2,I1+2)                                                 DLET0190
      LOCSIZ = IA(2,I1+3)                                               DLET0200
      NOFILS = NOFILS - 1                                               DLET0210
      IF (NOFILS) 100,110,129                                           DLET0220
  100 IF (IA(1,LASDES+3) .GE. MIN(1)) GO TO 101                         DLET0230
      MIN(1) = IA(1,LASDES+3)                                           DLET0240
      GO TO 102                                                         DLET0250
  101 IF (IA(1,LASDES+3) .GE. MAX(1)) MAX(1) = IA(1,LASDES+3) +         DLET0260
     * IA(2,LASDES+3)                                                   DLET0270
  102 NEXDES = LASDES - 4                                               DLET0280
      LASDES = NEXDES - 4                                               DLET0290
      NOFILS = 48                                                       DLET0300
      NOLINK = NOLINK - 1                                               DLET0310
      SIZE(1) = SIZE(1) + 200                                           DLET0320
      DLT = .TRUE.                                                      DLET0330
      GO TO 130                                                         DLET0340
  110 NEXDES = LASDES                                                   DLET0350
      LASDES = IA(1,NEXDES+197) + 4                                     DLET0360
      GO TO 130                                                         DLET0370
  129 NEXDES = LASDES                                                   DLET0380
      LASDES = NEXDES - 4                                               DLET0390
  130 IF (IA(1,I1+3) .EQ. 0) GO TO 133                                  DLET0400
      IF (IA(1,I1+3) .GE. MIN(UNIT)) GO TO 131                          DLET0410
      MIN(UNIT) = IA(1,I1+3)                                            DLET0420
      GO TO 132                                                         DLET0430
  131 IF (IA(1,I1+3) .GE. MAX(UNIT)) MAX(UNIT) = IA(1,I1+3) + LOCSIZ    DLET0440
  132 SIZE(UNIT) = SIZE(UNIT) + LOCSIZ                                  DLET0450
      IF (UNIT.EQ.1) DLT = .TRUE.                                       DLET0460
  133 IF (I1 .EQ. NEXDES) RETURN                                        DLET0470
      A(I1) = A(NEXDES)                                                 DLET0480
      A(I1+1) = A(NEXDES+1)                                             DLET0490
      A(I1+2) = A(NEXDES+2)                                             DLET0500
      A(I1+3) = A(NEXDES+3)                                             DLET0510
      RETURN                                                            DLET0530
      END                                                               DLET0540
      SUBROUTINE DMPLST                                                 DMPL0010
      COMMON /FAST/ A(1)                                                DMPL0020
      REAL*8 A                                                          DMPL0030
      INTEGER IA(2,1)                                                   DMPL0040
      EQUIVALENCE (A(1),IA(1,1))                                        DMPL0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       DMPL0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,        DMPL0070
     * SZZ                                                               DMPL0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   DMPL0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),             DMPL0090
     * SZZ(8)                                                            DMPL0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     DMPL0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      INTEGER FTB/'FTB'/                                                DMPL0120
      IF (FIRST .NE. FTB) CALL FABEND                                   DMPL0130
      WRITE (6,5) SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,NEXDES        DMPL0140
    5 FORMAT ('0SIZE ='7I10/'  MAX ='7I10/'  MIN ='7I10/'0NOLINK ='     DMPL0150
     * I5,5X'NOFILS ='I5,5X'LINK1 ='I6,5X'LASDES ='I6,5X'NEXDES ='I6)   DMPL0160
      N1 = LINK1                                                        DMPL0170
      DO 1 I = 1,NOLINK                                                 DMPL0180
      N2 = N1 + 199                                                     DMPL0190
      IF (I.NE.NOLINK) GO TO 3                                          DMPL0200
      IF (NOFILS.EQ.0) RETURN                                           DMPL0210
      N2 = N1 + 4*NOFILS - 1                                            DMPL0220
    3 WRITE (6,2) I,(A(J),IA(1,J+1),IA(2,J+1),IA(1,J+2),IA(2,J+2),      DMPL0230
     * IA(1,J+3),IA(2,J+3),J=N1,N2,4)                                   DMPL0240
    2 FORMAT ('0'55X'LINK'I3/2(13X'ID  SETSIZ  NOSETS  DOPEN UNIT   DLODMPL0250
     *C     LOCSIZ')/(F15.0,I7,I8,I7,I5,I8,I9,F16.0,I7,I8,I7,I5,I8,I9)) DMPL0260
    1 N1 = IA(1,N2)                                                     DMPL0270
      RETURN                                                            DMPL0290
      END                                                               DMPL0300
```

```
      SUBROUTINE DSCRIB (ID,SETSIZ,NOSETS,UNIT)                         DSCR0010
      COMMON /FAST/ A(1)                                                DSCR0020
      REAL*8 A                                                          DSCR0030
      INTEGER IA(2,1)                                                   DSCR0040
      EQUIVALENCE (A(1),IA(1,1))                                        DSCR0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       DSCR0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       DSCR0070
     * SZZ                                                              DSCR0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   DSCR0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            DSCR0090
     * SZZ(8)                                                           DSCR0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     DSCR0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      REAL*8 ID                                                         DSCR0120
      INTEGER SETSIZ,NOSETS,UNIT                                        DSCR0130
      IF (ID .EQ. 0.D0) CALL ERROR (19)                                 DSCR0190
      IF (SETSIZ .LE. 0) CALL ERROR (38)                                DSCR0200
      IF (NOSETS .LE. 0) CALL ERROR (31)                                DSCR0210
      IF (UNIT.LE.0 .OR. UNIT.GT.NDSK2) CALL ERROR (42)                 DSCR0220
      IF (UNIT.GT.2 .AND. SETSIZ.GT.RECLEN(UNIT-2)) CALL ERROR (37)     DSCR0230
      CALL IDFIND (ID,I1)                                               DSCR0240
      IF (I1 .GT. 0) CALL ERROR (14)                                    DSCR0250
      LASDES = NEXDES                                                   DSCR0260
      NOFILS = NOFILS + 1                                               DSCR0270
      A(NEXDES) = ID                                                    DSCR0280
      IA(1,NEXDES+1) = SETSIZ                                           DSCR0290
      IA(2,NEXDES+1) = NOSETS                                           DSCR0300
      IA(1,NEXDES+2) = 0                                                DSCR0310
      IA(2,NEXDES+2) = UNIT                                             DSCR0320
      IA(1,NEXDES+3) = 0                                                DSCR0330
      IA(2,NEXDES+3) = 0                                                DSCR0340
      NEXDES = NEXDES + 4                                               DSCR0350
      IF (NOFILS .LT. 49) RETURN                                        DSCR0360
      IF (200 .GT. SIZE(1)) CALL ERROR (34)                             DSCR0370
      CALL SHFTLK                                                       DSCR0380
      CALL LOCATE (1,200,I1)                                            DSCR0390
      A(NEXDES) = 0.D0                                                  DSCR0400
      IA(2,NEXDES+1) = 0                                                DSCR0410
      IA(1,NEXDES+2) = 0                                                DSCR0420
      IA(2,NEXDES+2) = 1                                                DSCR0430
      IA(1,NEXDES+3) = I1                                               DSCR0440
      IA(2,NEXDES+3) = 200                                              DSCR0450
      NOLINK = NOLINK + 1                                               DSCR0460
      NOFILS = 0                                                        DSCR0470
      LASDES = NEXDES                                                   DSCR0480
      NEXDES = I1                                                       DSCR0490
      IA(1,NEXDES+197) = LASDES-4                                       DSCR0500
      RETURN                                                            DSCR0520
      END                                                               DSCR0530
      SUBROUTINE ERROR (ERR)                                            ERR 0010
      COMMON /FAST/ A(1)                                                ERR 0020
      REAL*8 A                                                          ERR 0030
      INTEGER IA(2,1)                                                   ERR 0040
      EQUIVALENCE (A(1),IA(1,1))                                        ERR 0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       ERR 0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       ERR 0070
     * SZZ                                                              ERR 0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   ERR 0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            ERR 0090
     * SZZ(8)                                                           ERR 0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     ERR 0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      INTEGER ERR                                                       ERR 0120
      WRITE (6,2) ERR                                                   ERR 0170
    2 FORMAT ('0'31X'******** ERROR NUMBER'I3,' FROM FTB PACKAGE.')     ERR 0180
      CALL DMPLST                                                       ERR 0200
      CALL FABEND                                                       ERR 0210
      RETURN                                                            ERR 0220
      END                                                               ERR 0230
      SUBROUTINE FTBSFT(ID,FLSZ,LOHI,INDX)                              SHFT0010
      COMMON /FAST/ A(1)                                                SHFT0020
      REAL*8 A                                                          SHFT0030
      INTEGER IA(2,1)                                                   SHFT0040
      EQUIVALENCE (A(1),IA(1,1))                                        SHFT0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       SHFT0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       SHFT0070
     * SZZ                                                              SHFT0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   SHFT0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            SHFT0090
     * SZZ(8)                                                           SHFT0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     SHFT0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      REAL*8 ID                                                         SHFT0120
      INTEGER FLSZ                                                      SHFT0130
      LOGICAL HILOS                                                     SHFT0140
      IF (ID .EQ. 0.D0) CALL ERROR (19)                                 SHFT0180
      IF (FLSZ .LE. 0) CALL ERROR (16)                                  SHFT0190
      IF (LOHI.LE.0 .OR. LOHI.GT.3) CALL ERROR (22)                     SHFT0200
      CALL SHFTLK                                                       SHFT0205
      CALL IDFIND (ID,I1)                                               SHFT0210
      IF (I1 .EQ. 0) CALL ERROR (11)                                    SHFT0220
      IF (IA(2,I1+1).NE.0) CALL ERROR (43)                              SHFT0230
      MMIN = MIN(1)                                                     SHFT0250
      IF (LOHI .EQ. 3) GO TO 150                                        SHFT0260
    1 IA(2,I1+2) = 0                                                    SHFT0270
      HILOS = HILO                                                      SHFT0280
      IF (HILOS .AND. LOHI.EQ.2) HILO = .FALSE.                         SHFT0290
      IF (.NOT.HILOS .AND. LOHI.GE.2) HILO = .TRUE.                     SHFT0300
      MSIZE = SIZE(1)                                                   SHFT0310
      IF (IA(1,I1+3).LT.MIN(1)) MIN(1) = IA(1,I1+3)                     SHFT0320
```

```
      IF (IA(1,I1+3).GE.MAX(1)) MAX(1) = IA(1,I1+3) + IA(2,I1+3)        SHFT0330
      CALL LOCATE (1,FLSZ,I2)                                           SHFT0340
      HILO = HILOS                                                      SHFT0350
      SIZE(1) = MSIZE                                                   SHFT0360
      IF (I2 .EQ. IA(1,I1+3)) GO TO 140                                 SHFT0370
      IF (LOHI .EQ. 3) CALL ERROR (48)                                  SHFT0380
      L = IA(2,I1+3)                                                    SHFT0390
      IF (FLSZ .LT.L) L = FLSZ                                          SHFT0400
      J = IA(1,I1+3)                                                    SHFT0410
      J1 = J + L - 1                                                    SHFT0420
      IF (J .LT. I2) GO TO 110                                          SHFT0430
      M = I2                                                            SHFT0440
      DO 100 I = J,J1                                                   SHFT0450
      A(M) = A(I)                                                       SHFT0460
  100 M = M + 1                                                         SHFT0470
      GO TO 130                                                         SHFT0480
  110 M = I2 + L - 1                                                    SHFT0490
      N = J1                                                            SHFT0500
      DO 120 I = J,J1                                                   SHFT0510
      A(M) = A(N)                                                       SHFT0520
      M = M - 1                                                         SHFT0530
  120 N = N - 1                                                         SHFT0540
  130 IA(1,I1+3) = I2                                                   SHFT0550
  140 SIZE(1) = SIZE(1)-FLSZ+IA(2,I1+3)                                 SHFT0560
      IA(2,I1+3) = FLSZ                                                 SHFT0570
      IA(1,I1+1) = FLSZ                                                 SHFT0580
      IA(2,I1+2) = 1                                                    SHFT0590
      IF (LOHI .EQ. 3) MIN(1) = MMIN                                    SHFT0600
      INDX = IA(1,I1+3)                                                 SHFT0610
      DLT = .TRUE.                                                      SHFT0620
      RETURN                                                            SHFT0640
  150 IF (FLSZ .GT. IA(2,I1+3)) GO TO 160                               SHFT0650
      IF (MAX(1) .LE. IA(1,I1+3)) MAX(1) = IA(1,I1+3) + IA(2,I1+3)      SHFT0660
      IF (MIN(1).GE.IA(1,I1+3)) MMIN = IA(1,I1+3) + FLSZ                SHFT0665
      GO TO 140                                                         SHFT0670
  160 MIN(1) = IA(1,I1+3)                                               SHFT0680
      IF (MIN(1) .GE. MAX(1)) CALL ERROR (48)                           SHFT0690
      GO TO 1                                                           SHFT0700
      END                                                               SHFT0710
C     COMMON /COVRA1/ CVRA1(1)                                          CDC 0020
C     RETURN                                                            CDC 0030
C     END                                                               CDC 0040
C     SUBROUTINE SOVRA2                                                 SOV20010
C     COMMON /FAST/ FA(1)                                               SOV20020
C     RETURN                                                            SOV20030
C     END                                                               SOV20040
C     SUBROUTINE DEFIN                                                  DEFI0010
C     COMMON /DEFINC/ DEFINV                                            DEFI0020
C     RETURN                                                            DEFI0030
C     END                                                               DEFI0040
C     SUBROUTINE DEFTR                                                  DEFT0010
C     COMMON /DEFTRC / DEFTRV(33800)                                    IBM 0015
C     COMMON /DEFTRC / DEFTRV(  750)                                    CDC 0020
C     RETURN                                                            DEFT0030
C     END                                                               DEFT0040
C     SUBROUTINE DEFL1                                                  DEF10010
C     COMMON /DEFL1C/ DEFL1V                                            DEF10020
C     RETURN                                                            DEF10030
C     END                                                               DEF10040
C     SUBROUTINE DEFL2                                                  DEF20010
C     COMMON /DEFL2C/ DEFL2V(200)                                       DEF20020
C     COMMON /DEFLLC/ DEFLLV                                            CDC 0025
C     LEVEL 2, DEFLLV                                                   CDC 0026
C     RETURN                                                            DEF20030
C     END                                                               DEF20040
      SUBROUTINE GETCOR                                                 GTCB0010
      IMPLICIT REAL*8 (A-H,O-Z)                                         GTCB0020
      COMMON /FAST/ FA(100),DEFINV,DEFTRV,FAWORK(22000)                 GTBC0030
     * ,FAEND                                                           GTCB0035
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       GTCB0040
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       GTCB0050
     * SZZ                                                              GTCB0060
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   GTCB0070
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            GTCB0080
     * SZZ(8)                                                           GTCB0090
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     GTCB0100
      LOGICAL HILO,DLY,DLT,DPN(4)                                       GTCB0110
C                                                                       GTCB0120
      LOC = LOCF(FA(1))                                                 GTCB0130
      SZZ(1) = (LOCF(FAEND) - LOC)/2 + 1                                GTCB0140
      DO 10 I = 2,7                                                     GTCB0150
      SZZ(I) = 0                                                        GTCB0160
   10 CONTINUE                                                          GTCB0170
      SZZ(8) = 1                                                        GTCB0180
C                                                                       GTCB0190
      DO 20 I = 1,5                                                     GTCB0200
      IRECLN(I) = 0                                                     GTCB0210
   20 CONTINUE                                                          GTCB0220
      WRITE (6,600) SZZ(1), SZZ(2), SZZ(8)                              GTCB0230
  600 FORMAT (1H0,'FAST CORE SIZE = ',I6,5X,'BULK CORE SIZE = ',I6,5X,  GTCB0240
     * 'BULK INDEX = ',I6)                                              GTCB0250
      RETURN                                                            GTCB0260
C                                                                       GTCB0270
      ENTRY IFUNIT (IUNIT)                                              GTCB0280
      RETURN                                                            GTCB0290
      END                                                               GTCB0300
      SUBROUTINE IDFIND (/ID/,/I1/)                                     IDF20010
      COMMON /FAST/ A(1)                                                IDF20020
      REAL*8 A                                                          IDF20030
      INTEGER IA(2,1)                                                   IDF20040
```

```
      EQUIVALENCE (A(1),IA(1,1))                                        IDF20050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       IDF20060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       IDF20070
     * SZZ                                                              IDF20075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   IDF20080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            IDF20090
     * SZZ(8)                                                           IDF20095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     IDF20110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JARI0111
      REAL*8 ID                                                         IDF20120
      INTEGER FTB/'FTB'/                                                IDF20125
      IF (FIRST .NE. FTB) CALL FABEND                                   IDF20126
      J = NOLINK                                                        IDF20130
      I = LASDES                                                        IDF20140
      K = NOFILS                                                        IDF20150
      IF (K .EQ. 0) GO TO 100                                           IDF20160
   20 IF (ID .EQ. A(I)) GO TO 80                                        IDF20170
      K = K - 1                                                         IDF20180
      IF (K .EQ. 0) GO TO 60                                            IDF20190
      I = I - 4                                                         IDF20200
      GO TO 20                                                          IDF20210
   60 J = J - 1                                                         IDF20220
      IF (J .EQ. 0) GO TO 65                                            IDF20230
      K = 49                                                            IDF20240
      I = IA(1,I+197)                                                   IDF20250
      GO TO 20                                                          IDF20260
   65 I1 = 0                                                            IDF20270
      RETURN                                                            IDF20280
   80 I1 = I                                                            IDF20290
      RETURN                                                            IDF20300
  100 I = NEXDES                                                        IDF20310
      GO TO 60                                                          IDF20320
      END                                                               IDF20330
      SUBROUTINE INITAL (LOWHI)                                         INTL0010
      COMMON /FAST/ A(1)                                                INTL0020
      REAL*8 A                                                          INTL0030
      INTEGER IA(2,1)                                                   INTL0040
      EQUIVALENCE (A(1),IA(1,1))                                        INTL0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       INTL0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       INTL0070
     * SZZ                                                              INTL0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   INTL0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            INTL0090
     * SZZ(8)                                                           INTL0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     INTL0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JARI0111
      INTEGER FTB/'FTB'/                                                INTL0180
      IF (FIRST.NE.FTB) CALL GETCOR                                     INTL0200
      FIRST = FTB                                                       INTL0440
      NOLINK = 1                                                        INTL0450
      NOFILS = 1                                                        INTL0460
      SIZE(1) = SZZ(1) - 200                                            INTL0490
      IF (SIZE(1).GT.0) GO TO 30                                        INTL0500
      WRITE (6,1001)                                                    INTL0501
 1001 FORMAT ('0******** NOT SUFFICIENT FAST CORE SPACE FOR FTB ROUTINEINTL0502
     *S')                                                               INTL0503
   32 CALL FABEND                                                       INTL0504
   31 WRITE (6,1002)                                                    INTL0505
 1002 FORMAT ('0******** ARGUMENT ERROR TO FTB INITILIZATION')          INTL0506
      GO TO 32                                                          INTL0507
   30 IF (LOWHI.LT.1 .OR. LOWHI.GT.2) GO TO 31                          INTL0510
      IF (LOWHI.EQ.1) GO TO 13                                          INTL0520
      HILO = .FALSE.                                                    INTL0530
      MAX(1) = SIZE(1) + 1                                              INTL0540
      MIN(1) = 1                                                        INTL0550
      LINK1 = MAX(1)                                                    INTL0560
      GO TO 14                                                          INTL0570
   13 HILO = .TRUE.                                                     INTL0580
      MIN(1) = 201                                                      INTL0590
      MAX(1) = SZZ(1) + 1                                               INTL0600
      LINK1 = 1                                                         INTL0610
   14 A(LINK1) = 0.                                                     INTL0620
      A(LINK1+1) = 0.                                                   INTL0630
      IA(1,LINK1+2) = 0                                                 INTL0640
      IA(2,LINK1+2) = 1                                                 INTL0660
      IA(1,LINK1+3) = LINK1                                             INTL0670
      IA(2,LINK1+3) = 200                                               INTL0680
      NEXDES = LINK1+4                                                  INTL0700
      LASDES = LINK1                                                    INTL0710
      SIZE(2) = SZZ(2)                                                  INTL0740
      MIN(2) = SZZ(8)                                                   INTL0745
      MAX(2) = MIN(2) + SIZE(2)                                         INTL0750
      DO 15 I = 1,4                                                     INTL0790
   15 DPN(I) = .FALSE.                                                  INTL0800
      DO 23 I = 3,7                                                     INTL0810
      MIN(I) = 1                                                        INTL0812
      SIZE(I) = SZZ(I)                                                  INTL0813
      MAX(I) = SIZE(I) + 1                                              INTL0820
      RECLEN(I-2) = IRECLN(I-2) - 2                                     INTL0830
      IRECL4(I-2) = IRECLN(I-2) + 3                                     INTL0840
   23 IRECLT(I-2) = IRECL4(I-2) + IRECLN(I-2)                           INTL0850
      DLY = .FALSE.                                                     INTL0860
      DLT = .FALSE.                                                     INTL0870
      RETURN                                                            INTL0886
      END                                                               INTL0890
      FUNCTION ISFDES (ID)                                              ISFD0010
      COMMON /FAST/ A(1)                                                ISFD0020
      REAL*8 A                                                          ISFD0030
      INTEGER IA(2,1)                                                   ISFD0040
      EQUIVALENCE (A(1),IA(1,1))                                        ISFD0050
```

```
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       ISFD0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       ISFD0070
     * SZZ                                                              ISFD0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   ISFD0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            ISFD0090
     * SZZ(8)                                                           ISFD0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     ISFD0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      REAL*8 ID                                                         ISFD0120
      IF (ID .EQ. 0) CALL ERROR (19)                                    ISFD0140
      CALL IDFIND (ID,I1)                                               ISFD0150
      ISFDES = 0                                                        ISFD0155
      IF (I1 .NE. 0) ISFDES = 1                                         ISFD0160
      RETURN                                                            ISFD0210
      END                                                               ISFD0220
      FUNCTION LCONTG (UNIT)                                            LCON0010
      COMMON /FAST/ A(1)                                                LCON0020
      REAL*8 A                                                          LCON0030
      INTEGER IA(2,1)                                                   LCON0040
      EQUIVALENCE (A(1),IA(1,1))                                        LCON0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       LCON0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       LCON0070
     * SZZ                                                              LCON0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   LCON0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            LCON0090
     * SZZ(8)                                                           LCON0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     LCON0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      INTEGER UNIT                                                      LCON0130
      LOGICAL HILOS                                                     LCON0135
      IF (UNIT.LE.0 .OR. UNIT.GT.7) CALL ERROR (42)                     LCON0160
      LCONTG = 0                                                        LCON0165
      IF (SIZE(UNIT) .EQ. 0) RETURN                                     LCON0170
      CALL SHFTLK                                                       LCON0175
      HILOS = HILO                                                      LCON0176
      HILO = .FALSE.                                                    LCON0177
      CALL LOCATE (UNIT,0,I2)                                           LCON0180
      HILO = HILOS                                                      LCON0185
      IF (UNIT .NE. 1) GO TO 100                                        LCON0190
      IF (NOFILS .LT. 48) GO TO 110                                     LCON0200
      I2 = I2 - 200                                                     LCON0210
      IF (I2 .GE. 0) GO TO 110                                          LCON0220
      RETURN                                                            LCON0230
  100 IF (UNIT .GT. 2) I2 = I2*RECLEN(UNIT-2)                           LCON0240
  110 LCONTG = I2                                                       LCON0250
      RETURN                                                            LCON0270
      END                                                               LCON0300
      SUBROUTINE LOCATE (UNIT,SIZ,IS)                                   LCAT0010
      COMMON /FAST/ A(1)                                                LCAT0020
      REAL*8 A                                                          LCAT0030
      INTEGER IA(2,1)                                                   LCAT0040
      EQUIVALENCE (A(1),IA(1,1))                                        LCAT0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       LCAT0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,       LCAT0070
     * SZZ                                                              LCAT0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   LCAT0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            LCAT0090
     * SZZ(8)                                                           LCAT0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     LCAT0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      INTEGER UNIT,SIZ,IS                                               LCAT0120
      LOGICAL SW,SET                                                    LCAT0130
      SW = .TRUE.                                                       LCAT0140
      IS = SIZ                                                          LCAT0150
      IF (IS .NE. 0) GO TO 10                                           LCAT0160
      IF (SIZE(UNIT) .EQ. 0) RETURN                                     LCAT0170
      KSTR = MIN(UNIT)                                                  LCAT0180
      KEND = MAX(UNIT)                                                  LCAT0190
      MNS = KEND - KSTR                                                 LCAT0200
      IS = 1                                                            LCAT0210
      SW = .FALSE.                                                      LCAT0220
      SET = .TRUE.                                                      LCAT0225
   10 IF (HILO) GO TO 11                                                LCAT0230
      IEND = MAX(UNIT)                                                  LCAT0240
      ISTR = IEND - IS                                                  LCAT0250
      GO TO  12                                                         LCAT0260
   11 ISTR = MIN(UNIT)                                                  LCAT0270
      IEND = ISTR + IS                                                  LCAT0280
   12 J = NOLINK                                                        LCAT0290
      I = LASDES                                                        LCAT0300
      K = NOFILS                                                        LCAT0310
      IF (K .EQ. 0) GO TO 100                                           LCAT0320
   20 IF (UNIT .NE. IA(2,I+2)) GO TO 21                                 LCAT0330
      JSTR = IA(1,I+3)                                                  LCAT0340
      JEND = JSTR + IA(2,I+3)                                           LCAT0350
   22 IF (JSTR.LT.ISTR) GO TO 23                                        LCAT0360
      IF (JSTR.LT.IEND) GO TO 150                                       LCAT0370
   23 IF (JEND.LE.ISTR) GO TO 40                                        LCAT0380
      IF (JEND.LE.IEND) GO TO 150                                       LCAT0390
      GO TO 40                                                          LCAT0400
   21 IF (UNIT.NE.1 .OR. IA(2,I+2).EQ.2) GO TO 41                       LCAT0410
      JSTR = IABS(IA(1,I+2))                                            LCAT0420
      IF (JSTR .EQ. 0) GO TO 41                                         LCAT0430
      JEND = JSTR + IA(2,JSTR+1)                                        LCAT0440
      GO TO 22                                                          LCAT0450
   40 IF (SW) GO TO 41                                                  LCAT0460
      NSTR = IEND - JEND                                                LCAT0480
      IF (NSTR.LE.0) GO TO 41                                           LCAT0510
      IF (NSTR.GE.MNS) GO TO 41                                         LCAT0520
      MNS = NSTR                                                        LCAT0530
```

```
      KSTR = JSTR                                                       LCAT0540
      KEND = JEND                                                       LCAT0550
      SET = .FALSE.                                                     LCAT0555
   41 K = K - 1                                                         LCAT0560
      IF (K .EQ. 0) GO TO 60                                            LCAT0570
      I = I - 4                                                         LCAT0580
      GO TO 20                                                          LCAT0590
   60 J = J - 1                                                         LCAT0600
      IF (J .EQ. 0) GO TO 65                                            LCAT0610
      K = 50                                                            LCAT0620
      I = IA(1,I+197) + 4                                               LCAT0630
      GO TO 20                                                          LCAT0640
   65 IF (SW) GO TO 25                                                  LCAT0650
      IS = MNS                                                          LCAT0660
      IF (SET) RETURN                                                   LCAT0670
      JSTR = KSTR                                                       LCAT0680
      JEND = KEND                                                       LCAT0690
  154 MNS = JSTR - MIN(UNIT)                                            LCAT0695
      SET = .TRUE.                                                      LCAT0696
      GO TO 157                                                         LCAT0700
   25 SIZE(UNIT) = SIZE(UNIT) - IS                                      LCAT0710
      IS = ISTR                                                         LCAT0720
      IF (IS .NE. MIN(UNIT)) GO TO 26                                   LCAT0730
      MIN(UNIT) = IEND                                                  LCAT0740
      GO TO 27                                                          LCAT0750
   26 IF (IEND .EQ. MAX(UNIT)) MAX(UNIT) = IS                           LCAT0760
   27 IF (.NOT.DLY) RETURN                                              LCAT0770
      DLY = .FALSE.                                                     LCAT0780
      DO 28 I = 1,5                                                     LCAT0790
   28 CALL IFUNIT(I)                                                    LCAT0800
      RETURN                                                            LCAT0810
  100 I = NEXDES                                                        LCAT0820
      GO TO 60                                                          LCAT0830
  150 IF (.NOT.SW) GO TO 154                                            LCAT0835
      IF (JSTR.EQ.MIN(UNIT)) MIN(UNIT) = JEND                           LCAT0840
      IF (JEND.EQ.MAX(UNIT)) MAX(UNIT) = JSTR                           LCAT0850
      IF (HILO) GO TO 151                                               LCAT0860
  157 IEND = JSTR                                                       LCAT0870
      ISTR = IEND - IS                                                  LCAT0880
      IF (ISTR .GE. MIN(UNIT)) GO TO 12                                 LCAT0890
      GO TO 152                                                         LCAT0900
  151 ISTR = JEND                                                       LCAT0910
      IEND = ISTR + IS                                                  LCAT0920
      IF (IEND .LE. MAX(UNIT)) GO TO 12                                 LCAT0930
  152 IF (SW) CALL ERROR (47)                                           LCAT0940
      RETURN                                                            LCAT0960
      END                                                               LCAT0970
      REAL FUNCTION NEXTID*8 (/DUMMY/)                                  NXTD0010
      COMMON /FAST/ A(1)                                                NXTD0020
      REAL*8 A                                                          NXTD0030
      INTEGER IA(2,1)                                                   NXTD0040
      EQUIVALENCE (A(1),IA(1,1))                                        NXTD0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       NXTD0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,        NXTD0070
     * SZZ                                                               NXTD0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   NXTD0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),             NXTD0090
     * SZZ(8)                                                            NXTD0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     NXTD0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JARI0111
      INTEGER FTB/'FTB'/                                                NXTD0115
      REAL*8 M,MN                                                       NXTD0120
      IF (FIRST.EQ.FTB) GO TO 10                                        NXTD0140
      WRITE (6,11)                                                      NXTD0141
   11 FORMAT ('0******* INITAL NOT FIRST CALL TO FTB ROUTINES')         NXTD0142
      CALL FABEND                                                       NXTD0143
   10 J = NOLINK                                                        NXTD0150
      I = LASDES                                                        NXTD0160
      K = NOFILS                                                        NXTD0170
      M = 0                                                             NXTD0180
      IF (K .EQ. 0) GO TO 100                                           NXTD0190
   20 MN = DABS (A(I))                                                  NXTD0200
      IF (MN .GT. M) M = MN                                             NXTD0210
      K = K - 1                                                         NXTD0220
      IF (K .EQ. 0) GO TO 60                                            NXTD0230
      I = I - 4                                                         NXTD0240
      GO TO 20                                                          NXTD0250
   60 J = J - 1                                                         NXTD0260
      IF (J .EQ. 0) GO TO 65                                            NXTD0270
      K = 49                                                            NXTD0280
      I = IA(1,I+197)                                                   NXTD0290
      GO TO 20                                                          NXTD0300
   65 NEXTID = M + 1.D0                                                 NXTD0310
      RETURN                                                            NXTD0320
  100 I = NEXDES                                                        NXTD0330
      GO TO 60                                                          NXTD0340
      END                                                               NXTD0350
      SUBROUTINE RESERV (ID,FILSIZ,LOHI,I2)                             RSRV0010
      COMMON /FAST/ A(1)                                                RSRV0020
      REAL*8 A                                                          RSRV0030
      INTEGER IA(2,1)                                                   RSRV0040
      EQUIVALENCE (A(1),IA(1,1))                                        RSRV0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       RSRV0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,        RSRV0070
     * SZZ                                                               RSRV0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   RSRV0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),             RSRV0090
     * SZZ(8)                                                            RSRV0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     RSRV0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JARI0111
```

```
      REAL*8 ID                                                         RSRV0120
      INTEGER FILSIZ,LOHI                                               RSRV0130
      LOGICAL HILOS                                                     RSRV0140
      IF (FILSIZ .GT. SIZE(1) .OR. FILSIZ .LE. 0) CALL ERROR (16)       RSRV0170
      IF (LOHI.LE.0 .OR. LOHI.GT.2) CALL ERROR (22)                     RSRV0180
      CALL SHFTLK                                                       RSRV0190
      CALL DSCRIB (ID,FILSIZ,1,1)                                       RSRV0200
      CALL IDFIND (ID,I1)                                               RSRV0210
      IA (2,I1+1) = 0                                                   RSRV0220
      HILOS = HILO                                                      RSRV0230
      IF (LOHI .EQ. 2) HILO = .NOT.HILO                                 RSRV0240
      CALL LOCATE (1,FILSIZ,I2)                                         RSRV0250
      HILO = HILOS                                                      RSRV0260
      IA(1,I1+3) = I2                                                   RSRV0270
      IA(2,I1+3) = FILSIZ                                               RSRV0280
      RETURN                                                            RSRV0310
      END                                                               RSRV0320
      SUBROUTINE SHFTLK                                                 SHFK0010
      COMMON /FAST/ A(1)                                                SHFK0020
      REAL*8 A                                                          SHFK0030
      INTEGER IA(2,1)                                                   SHFK0040
      EQUIVALENCE (A(1),IA(1,1))                                        SHFK0050
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,       SHFK0060
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,        SHFK0070
     * SZZ                                                              SHFK0075
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,   SHFK0080
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),            SHFK0090
     * SZZ(8)                                                           SHFK0095
C     LOGICAL HILO,DLY,DLT,DPN*1(4)                                     SHFK0110
      LOGICAL HILO,DLY,DLT,DPN(4)                                       JAR10111
      INTEGER JT(2)                                                     SHFK0120
      IF (.NOT.DLT) RETURN                                              SHFK0130
      DLT = .FALSE.                                                     SHFK0140
      IF (NOLINK .EQ. 1) RETURN                                         SHFK0150
      JT(1) = 0                                                         SHFK0160
      MMAX = MAX(1)                                                     SHFK0170
      MMIN = MIN(1)                                                     SHFK0180
      MLINK = 1                                                         SHFK0190
      J = LINK1                                                         SHFK0200
  100 JT(2) = JT(1)                                                     SHFK0210
      L = J + 196                                                       SHFK0220
      J = IA(1,L+3)                                                     SHFK0230
      IF (HILO .AND. J.LT.MIN(1)  .OR.  .NOT.HILO .AND. J.GE.MAX(1))    SHFK0240
     * GO TO 190                                                        SHFK0250
      IA(2,L+2) = 0                                                     SHFK0260
      IF (HILO) GO TO 105                                               SHFK0270
      MIN(1) = J                                                        SHFK0280
      GO TO 110                                                         SHFK0290
  105 MAX(1) = J + 200                                                  SHFK0300
  110 MSIZE = SIZE(1)                                                   SHFK0310
      CALL LOCATE (1,200,I2)                                            SHFK0320
      SIZE(1) = MSIZE                                                   SHFK0330
      IA(2,L+2) = 1                                                     SHFK0340
      JT(1) = J - I2                                                    SHFK0350
      IF (JT(1) .EQ. 0) GO TO 150                                       SHFK0360
      IA(1,L+3) = I2                                                    SHFK0370
      J1 = J                                                            SHFK0380
      J2 = I2                                                           SHFK0390
      IF (HILO) GO TO 130                                               SHFK0400
      DO 120 I = 1,200                                                  SHFK0410
      J1 = J1 - 1                                                       SHFK0420
      J2 = J2 - 1                                                       SHFK0430
  120 A(J2+200) = A(J1+200)                                             SHFK0440
      GO TO 150                                                         SHFK0450
  130 DO 140 I = 1,200                                                  SHFK0460
      A(J2) = A(J1)                                                     SHFK0470
      J2 = J2 + 1                                                       SHFK0480
  140 J1 = J1 + 1                                                       SHFK0490
  150 MLINK = MLINK + 1                                                 SHFK0500
      IF (MLINK .EQ. NOLINK) GO TO 200                                  SHFK0510
      IF (JT(1).EQ.0) GO TO 100                                         SHFK0520
      K = IA(1,I2+199) + 196                                            SHFK0530
      IA(1,K+1) = I2 + 192                                              SHFK0540
      J = I2                                                            SHFK0550
      GO TO 100                                                         SHFK0560
  200 IF (MMAX.GT.MAX(1)) MAX(1) = MMAX                                 SHFK0570
      IF (MMIN.LT.MIN(1)) MIN(1) = MMIN                                 SHFK0580
      IF (NOFILS .EQ. 0) GO TO 210                                      SHFK0590
      JT(2) = JT(1)                                                     SHFK0600
  210 LASDES = LASDES - JT(2)                                           SHFK0610
      NEXDES = NEXDES - JT(1)                                           SHFK0620
      RETURN                                                            SHFK0630
      END                                                               SHFK0640
```

付録5　MOD6私用ジョブマクロリスト

```
*                                                                       PLT40010
*                 ***************************************               PLT40020
*                                                                       PLT40030
*                     RLP4PLOT (79.02.01)....SYSTEM A                   PLT40040
*                                                                       PLT40050
*                 ***************************************               PLT40060
*                                                                       PLT40070
 DEFINE   RLP4PLOT,MACU=J0265,SIZE=40,COM=ON,EN=PLOTR4M,               -PLT40080
                PNP=PLT4MPLT,PNC=PLT4MCOM,SYSOUT=A,NAME=1,RNP,P=,       PLT40090
*                                                                       PLT40100
*        MADE BY  A. KOHSAKA  AT FEB.  1, '79                           PLT40110
*                                                                       PLT40120
*        TEL=5767, ANZEN KAISEKI BU  ROOM # = 226                       PLT40130
*                                                                       PLT40140
*SET   CO=*COM                                                          PLT40150
*                                                                       PLT40160
*IF  CO=OFF                                                             PLT40170
*EXEC     *PNP,PARAM='SIZE=*SIZE,*RNP',NAME=PLT4*NAME,PRTY=(7,7)        PLT40180
*REND                                                                   PLT40190
*                                                                       PLT40200
*IF  CO=ON                                                              PLT40210
*EXEC     *PNC,PARAM='SIZE=*SIZE,*RNP',NAME=PLT4*NAME,PRTY=(7,7)        PLT40220
*REND                                                                   PLT40230
*                                                                       PLT40240
*FD       PROGLIB,FILE=(CATLG,*MACU*P*EN),DEVD=DA                       PLT40250
* FD      F,SUBTASK,FILE=(OLD,SYS1,EXELIB),VOL=(SPEC,DP0080),DEVD=DA,  PLT40260
*             UNIT=SYSTEM                                               PLT40270
*FD       F,SUBTASK,FILE=(OLD,SYS1,EXELIB),VOL=(SPEC,DP0060),DEVD=DA,  -PLT40280
             UNIT=SYSTEM                                               PLT40290
* FD      MESSAGE,FILE=(OLD,SYS1,FORTMSG),VOL=(SPEC,DP0080),DEVD=DA,   PLT40300
*             UNIT=SYSTEM                                               PLT40310
*FD       MESSAGE,FILE=(OLD,SYS1,FORTMSG),VOL=(SPEC,DP0060),DEVD=DA,   -PLT40320
             UNIT=SYSTEM                                               PLT40330
*FD       PRINT,FILE=(TEMP,PLT4PRT*NAME),UNIT=WKUNIT,VOL=WORK,DEVD=DA, -PLT40340
             SPACE=(TRK,10,10),SYSOUT=(*SYSOUT),DISP=DELETE            PLT40350
*FD       F11,FILE=(TEMP,F11),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40360
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40370
*FD       F12,FILE=(TEMP,F12),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40380
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40390
*FD       F13,FILE=(TEMP,F13),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40400
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40410
*FD       F14,FILE=(TEMP,F14),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40420
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40430
*FD       F15,FILE=(TEMP,F15),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40440
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40450
*FD       F16,FILE=(TEMP,F16),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40460
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40470
*FD       F17,FILE=(TEMP,F17),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40480
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40490
*FD       F18,FILE=(TEMP,F18),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40500
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40510
*FD       F19,FILE=(TEMP,F19),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40520
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40530
*FD       F20,FILE=(TEMP,F20),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40540
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40550
*FD       F21,FILE=(TEMP,F21),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40560
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40570
*FD       F22,FILE=(TEMP,F22),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40580
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40590
*FD       F23,FILE=(TEMP,F23),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40600
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40610
*FD       F24,FILE=(TEMP,F24),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40620
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40630
*FD       F25,FILE=(TEMP,F25),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40640
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40650
*FD       F26,FILE=(TEMP,F26),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40660
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40670
*FD       F27,FILE=(TEMP,F27),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40680
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40690
*FD       F28,FILE=(TEMP,F28),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40700
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40710
*FD       F29,FILE=(TEMP,F29),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40720
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40730
*FD       F30,FILE=(TEMP,F30),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,30,30),  -PLT40740
             DEVD=DA,DISP=DELETE,RCDSIZE=18,BLKSIZE=3114,TMOD=8        PLT40750
 DEND                                                                   PLT40760
*                                                                       PREL0010
*                 ***************************************               PREL0020
*                                                                       PREL0030
*                     RLP4PREL (79.02.01)....SYSTEM A                   PREL0040
*                                                                       PREL0050
*                 ***************************************               PREL0060
*                                                                       PREL0070
 DEFINE   RLP4PREL,MACU=J0265,SIZE=5,PN=PRELOAD,OVLD=PRELOADD,SYSOUT=A,-PREL0080
                OVLELM=OVERLAYF,NAME=1,E,P=,                            PREL0090
*                                                                       PREL0100
*        MADE BY  A. KOHSAKA  AT FEB.  1, '79                           PREL0110
*                                                                       PREL0120
*        TEL=5767, ANZEN KAISEKI BU  ROOM # = 226                       PREL0130
*                                                                       PREL0140
*EXEC     PRELOAD,PARAM='SIZE=*SIZE,*E',NAME=PREL*NAME,PRTY=(7,7)       PREL0150
*FD       PROGLIB,FILE=(CATLG,*MACU*P*PN),DEVD=DA                       PREL0160
* FD      F,SUBTASK,FILE=(OLD,SYS1,EXELIB),VOL=(SPEC,DP0080),DEVD=DA,  PREL0170
*             UNIT=SYSTEM                                               PREL0180
*FD       F,SUBTASK,FILE=(OLD,SYS1,EXELIB),VOL=(SPEC,DP0060),DEVD=DA,  -PREL0190
             UNIT=SYSTEM                                               PREL0200
* FD      MESSAGE,FILE=(OLD,SYS1,FORTMSG),VOL=(SPEC,DP0080),DEVD=DA,   PREL0210
*             UNIT=SYSTEM                                               PREL0220
*FD       MESSAGE,FILE=(OLD,SYS1,FORTMSG),VOL=(SPEC,DP0060),DEVD=DA,   -PREL0230
             UNIT=SYSTEM                                               PREL0240
```

```
*FD        PRINT,FILE=(TEMP,PRELFRT*NAME),UNIT=*KUNIT,VOL=*ORK,DEVD=DA,  -PREL0250
               SPACE=(TRK,10,10),SYSOUT=(*SYSOUT),DISP=DELETE            PREL0260
*FD        F17,FILE=(CATLG,*MACU*P*OVLD(*OVLELM)),DEVD=DA                PREL0270
*FD        F18,FILE=(TEMP,F18),VOL=*ORK,UNIT=*KUNIT,SPACE=(TRK,10,10),  -PREL0280
               DEVD=DA,DISP=DELETE,RCDSIZE=80,BLKSIZE=1200,TMOD=9        PREL0290
*FD        F20,FILE=(TEMP,OVLYIN),VOL=*ORK,UNIT=*KUNIT,DISP=PASS,TMOD=9,-PREL0300
               DEVD=DA,SPACE=(TRK,10,10),RCDSIZE=80,BLKSIZE=1200         PREL0310
*FD        F21,FILE=(TEMP,INPUT),VOL=*ORK,UNIT=*KUNIT,DISP=PASS,TMOD=9, -PREL0320
               DEVD=DA,SPACE=(TRK,20,20),RCDSIZE=80,BLKSIZE=1200         PREL0330
 DEND                                                                    PREL0340
*                                                                        RLP40010
*                 ************************************                   RLP40020
*                                                                        RLP40030
*                      RLP4MOD6 (79.02.01)....SYSTEM A                   RLP40040
*                                                                        RLP40050
*                 ************************************                   RLP40060
*                                                                        RLP40070
 DEFINE    RLP4MOD6,MACU=J0285,SIZE=5,RF=RLP4M6RB,PRVLIB=DUMMYRTN,      -RLP40080
               MAP=MAP,SYSOUT=A,NAME=1,LKP,RNP,FAWORK=ON,A=NOLIST,P=,    RLP40090
*                                                                        RLP40100
*         MADE BY  A. KOHSAKA  AT FEB.  1, '79                           RLP40110
*                                                                        RLP40120
*         TEL=5767, ANZEN KASEKI BU  ROOM # = 226                        RLP40130
*                                                                        RLP40140
*SET  CO=*FAWORK                                                         RLP40150
*IF   CO=OFF                                                             RLP40160
*                                                                        RLP40170
*       EXECUTION  FORTRAN                                               RLP40180
*                                                                        RLP40190
*EXEC      FORTRAN,PARAM='*A,BYNAME,CP,OPT2,AREA=64',NAME=FORT*NAME,    -RLP40200
               PRTY=(7,7)                                                RLP40210
*FD        PROGLIB,FILE=(CATLG,SYS1,FORT),DEVD=DA                        RLP40220
* FD       PROGLIB,FILE=(OLD,SYS1,FORT),UNIT=DR10,VOL=(SPEC,DR0010),     RLP40230
*               DEVD=DA                                                  RLP40240
*FD        SYSWK1,FILE=(TEMP,SWK1),UNIT=*KUNIT,VOL=*ORK,DEVD=DA,        -RLP40250
               DISP=DELETE,SPACE=(TRK,40,40)                             RLP40260
*FD        SYSPRT,FILE=(TEMP,FZ*NAME),UNIT=*KUNIT,VOL=*ORK,DEVD=DA,     -RLP40270
               SYSOUT=*SYSOUT,SPACE=(TRK,30,30),DISP=DELETE              RLP40280
*FD        RELBIN,FILE=(TEMP,FORT,RELBIN),UNIT=*KUNIT,VOL=*ORK,DEVD=DA, -RLP40290
               DISP=PASS,SPACE=(TRK,20,10(20))                           RLP40300
*FD        SOURCE,*                                                      RLP40310
      SUBROUTINE GETCOR                                                  RLP40320
      IMPLICIT REAL*8 (A-H,O-Z)                                          RLP40330
      COMMON /FAST/ FA(100),DEFINV,DEFTRV,FAWORK(*FAWORK)                RLP40340
     * ,FAEND                                                            RLP40350
      COMMON /FTB/ FIRST,SIZE,MAX,MIN,NOLINK,NOFILS,LINK1,LASDES,        RLP40360
     * NEXDES,NDSK2,RECLEN,IRECLT,IRECL4,IRECLN,HILO,DLY,DLT,DPN,        RLP40370
     * SZZ                                                               RLP40380
      INTEGER FIRST,SIZE(7),MAX(7),MIN(7),NOLINK,NOFILS,LINK1,LASDES,    RLP40390
     * NEXDES,NDSK2,RECLEN(5),IRECLT(5),IRECL4(5),IRECLN(5),             RLP40400
     * SZZ(8)                                                            RLP40410
      LOGICAL  HILO,DLY,DLT,DPN(4)                                       RLP40420
      LOC = LOCF(FA(1))                                                  RLP40430
      SZZ(1) = (LOCF(FAEND) - LOC)/2 + 1                                 RLP40440
      DO 10 I = 2,7                                                      RLP40450
      SZZ(I) = 0                                                         RLP40460
   10 CONTINUE                                                           RLP40470
      SZZ(8) = 1                                                         RLP40480
      DO 20 I = 1,5                                                      RLP40490
      IRECLN(I) = 0                                                      RLP40500
   20 CONTINUE                                                           RLP40510
      WRITE (6,600) SZZ(1), SZZ(2), SZZ(8)                               RLP40520
  600 FORMAT (1H0,'FAST CORE SIZE = ',I6,5X,'BULK CORE SIZE = ',I6,5X,   RLP40530
     * 'BULK INDEX = ',I6)                                               RLP40540
      RETURN                                                             RLP40550
      ENTRY  IFUNIT (IUINT)                                              RLP40560
      RETURN                                                             RLP40570
      END                                                                RLP40580
*                                                                        RLP40590
*       EXECUTION  LIBE                                                  RLP40600
*                                                                        RLP40610
*EXEC      LIBE,NAME=LIBEM*NAME,PRTY=(7,7)                               RLP40620
*FD        SYSPRT,FILE=(TEMP,LSPRT*NAME),UNIT=*KUNIT,VOL=*ORK,DEVD=DA,  -RLP40630
               SYSOUT=*SYSOUT,SPACE=(TRK,10,10),DISP=DELETE              RLP40640
*FD        SYSIN,*                                                       RLP40650
EDIT,R       DDNE=D,DDOLD0,DDOLD1                                        RLP40660
FIN                                                                      RLP40670
*FD        DDOLD0,FILE=(CATLG,*MACU*P*RF),DEVD=DA                        RLP40680
*FD        DDOLD1,FILE=(TEMP,FORT,RELBIN),DISP=DELETE,UNIT=*KUNIT,      -RLP40690
               VOL=*ORK,DEVD=DA                                          RLP40700
*FD        DDNEW0,FILE=(TEMP,RLP4RBTP),UNIT=*KUNIT,VOL=*ORK,DEVD=DA,    -RLP40710
               DISP=PASS,SPACE=(TRK,200,400(50))                         RLP40720
*REND                                                                    RLP40730
*                                                                        RLP40740
*       EXECUTION  GLIED                                                 RLP40750
*                                                                        RLP40760
*EXEC      GLIED,PARAM='*MAP,*A=30,*LKP',NAME=RLP4LK*NAME,PRTY=(7,7)     RLP40770
* FD       PROGLIB,FILE=(OLD,SYS1,LIED),VOL=(SPEC,DR0001),UNIT=DR01,    RLP40780
*               DEVD=DA                                                  RLP40790
*FD        PROGLIB,FILE=(CATLG,SYS1,LIED),DEVD=DA                        RLP40800
* FD       SYSLIB,FILE=(OLD,SYS1,FORTLIBH),VOL=(SPEC,DR0010),UNIT=DR10, RLP40810
*               DEVD=DA                                                  RLP40820
*FD        SYSLIB,FILE=(CATLG,SYS1,FORTLIBH),DEVD=DA                     RLP40830
*FD        PRVLIB,FILE=(CATLG,*MACU*P*PRVLIB),DEVD=DA                    RLP40840
*IF   CO=ON                                                              RLP40850
*FD        RELBIN,FILE=(CATLG,*MACU*P*RF),DEVD=DA                        RLP40860
*REND                                                                    RLP40870
*IF   CO=OFF                                                             RLP40880
*FD        RELBIN,FILE=(TEMP,RLP4RBTP),UNIT=*KUNIT,VOL=*ORK,DEVD=DA,    -RLP40890
               DISP=DELETE                                               RLP40900
```

```
*REND                                                                   RLP40910
*FD        EXELIB,FILE=(TEMP,RELAP4M6),VOL=WORK,UNIT=WKUNIT,DISP=PASS,  -RLP40920
               DEVD=DA,SPACE=(TRK,150,30)                               RLP40930
*FD        SYSPRT,FILE=(TEMP,GLDPRT*NAME),VOL=WORK,UNIT=WKUNIT,DEVD=DA, -RLP40940
               DISP=DELETE,SYSOUT=*SYSOUT,SPACE=(TRK,50,50)             RLP40950
*FD        DDOLDO,FILE=(TEMP,OVLYIN),VOL=WORK,UNIT=WKUNIT,DISP=DELETE,  -RLP40960
               DEVD=DA                                                  RLP40970
*FD        SYSIN,*                                                      RLP40980
NAME       RELAP4M6,ENTRY=ELM(FTMAIN),OVLY                              RLP40990
DOMAIN     HCM,RWX,OVLY                                                 RLP41000
INPUT      RELBIN                                                       RLP41010
CALL       PRVLIB                                                       RLP41020
SELECT,C   DDOLDO                                                       RLP41030
FIN                                                                     RLP41040
*                                                                       RLP41050
*        EXECUTION  RELAP4M6                                            RLP41060
*                                                                       RLP41070
*EXEC      RELAP4M6,PARAM='SIZE=*SIZE,*RNP',NAME=RLP4GO*NAME,PRTY=(7,7) RLP41080
*FD        PROGLIB,FILE=(TEMP,RELAP4M6),VOL=WORK,UNIT=WKUNIT,DEVD=DA,   -RLP41090
               DISP=DELETE                                              RLP41100
* FD       F,SUBTASK,FILE=(OLD,SYS1,EXELIB),VOL=(SPEC,DP0080),DEVD=DA,  RLP41110
*              UNIT=SYSTEM                                              RLP41120
*FD        F,SUBTASK,FILE=(OLD,SYS1,EXELIB),VOL=(SPEC,DP0060),DEVD=DA,  -RLP41130
               UNIT=SYSTEM                                              RLP41140
* FD       MESSAGE,FILE=(OLD,SYS1,FORTMSG),VOL=(SPEC,DP0080),DEVD=DA,   RLP41150
*              UNIT=SYSTEM                                              RLP41160
*FD        MESSAGE,FILE=(OLD,SYS1,FORTMSG),VOL=(SPEC,DP0060),DEVD=DA,   -RLP41170
               UNIT=SYSTEM                                              RLP41180
*FD        READ,FILE=(TEMP,INPUT),VOL=WORK,UNIT=WKUNIT,DISP=DELETE,     -RLP41190
               DEVD=DA                                                  RLP41200
*FD        PRINT,FILE=(TEMP,RLP4PRT*NAME),UNIT=WKUNIT,VOL=WORK,DEVD=DA, -RLP41210
               SPACE=(TRK,90,40),SYSOUT=*SYSOUT,DISP=DELETE             RLP41220
*FD        F15,FILE=(CATLG,J0000,J1598,RLP4ST)                          RLP41230
*FD        F16,FILE=(TEMP,F16),VOL=WORK,UNIT=WKUNIT,SPACE=(TRK,50,50),  -RLP41240
               DEVD=DA,DISP=DELETE                                      RLP41250
 DEND                                                                   RLP41260
```