JAERI-M
6623

過渡状態熱水力挙動解析コードRELAP-4の整備
（その1 FACOM230/75システムへの変換）

1976年7月

鴻坂厚夫・熊倉利昌*

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

JAERI-M 6623

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)

日本原子力研究所東海研究所
鴻坂厚夫・熊倉利昌*

(1976年6月21日受理)

NEA-Computer Program Libraryを通じて，1974年末に公開されたRELAP-4 コードを原機種IBM 360よりFACOM 230/75システム用に変換整備した。特に，コードの入力部分であるfree formatインプットルーティンは，いくつかの安全解析コードにも使用されているもので，一般性と計算機間の互換性があるようにFORTRAN化した。又，変換完了後，MKS 単位系といくつかの実験相関式を組み込んだヴァーションも作成した。

* 日本ソフトウェア開発株式会社

Implimentation of the Transient Thermal-Hydraulic Analysis Code RELAP-4
part 1
Conversion to the FACOM 230/75 Computer System

Atsuo KOHSAKA and Toshimasa KUMAKURA*
Division of Reactor Safety Evaluation, Tokai, JAERI

(Received June 21, 1976)

The transient thermal-hydraulic analysis code RELAP-4 which was distributed through NEA-Computer Program Library at the end of 1974 has been converted from IBM 360 to the FACOM 230/75 computer system. A version of the code has also been prepared which incorporates the MKS system of units and several additional correlations.

* Nippon Software Kaihatsu Ltd.

目　　次

1. 序　　論

RELAP−4は，アメリカにおいてRELAP−3を基礎にそれを大巾にレベルアップして作られた冷却材喪失事故解析コードで,BWR及びPWRのブローダウン計算や炉心詳細計算などができる。このコードは，1974年末NEA−CPL を通じて我国に公開されたが，公開当時，原研においても同種のコードを開発中であり，reference code として，重要な位置を占めるものである。更に，このコードはUSNRCの規制用コードパッケージWREM の中で，中心的なコードであり，その特質や欠点を調査する必要性からもFACOM への変換が急がれた。

NEA−CPLより入手したRELAP−4プログラム・パッケージは，

(1) RELAP−4本体

(2) NRTS(National Reactor Testing Station)− Enviromental Subroutine Library

　i) free format インプットルーティン

　ii) テープI/O ルーティン

(3) バイナリー形式の蒸気表

(4) プロッタープログラムPLOTR−4

(5) サンプルインプット

から成っている。RELAP−4 本体はカードにして23,000数からなる巨大なプログラムであるが，すべてFORTRANで書かれており，FACOMへの形式的な変換は，概して，要易であった。プログラム中の実変数はすべて倍精度になっており，オーバァーレイ構造で183K語のメモリーを必要とする。一方，NRTSサブルーティン・ライブラリーの多くはAssembler で書かれており，原機種IBM 360 への依存性が強い。これらのシステムルーティンの変換整備には，FORTRAN ルーティンの変更とAssembler ルーティンのFORTRAN 化，及びビット演算によるマスキングやシフト演算のシミュレーションが必要であったが，それぞれのサブルーティンについて，例題を作って確認しながら段階的に作業を進めた。Restart やプロッター機能に関しては，特殊な機能を持つテープアクセスルーティンが使用されていたが，FACOM への変換では，FORTRANでシミュレーション可能な機能のみに留めた。更に，蒸気表については，後で述べるように,IBM のバイナリー形式からFACOMのバイナリー形式に変換を行った。最後に，付随の簡単なサンプルインプット(RELAP−4マニュアルに掲載されているケースで，6ノード／9ジャンクションの問題)で計算を行ない，IBMで行なった計算と一致する結果を得た。

以上でコードの変換は終了したが，ひき続き，MKS単位系といくつかの実験相関式をオプショナルに組み込んだバァーションも作成した。このバァーション作成に必要なプログラムの変更は，すでに傍島氏(安全工学第1研究室)により，RELAP−4のCDCバァーションを用いて確認されていたものである。

RELAP−4コードに採用されている物理的モデルの説明やインプットの記述は文献1.に，又，新しく追加された実験相関式の内容については文献3にゆずり，本報告書では，上記システムルーティンの機能と変換に際しての工夫の説明，restart やプロッターのためのデータテープの構造，及び，MKS単位系と相関式を追加したバァーションでの新しいインプットの説明を行なうことにする。

2. 汎用 free format インプットルーティン

2.1. 特長と変換上の要点

RELAP−4で用いられている free format インプットルーティンは，NRTSでEnviromental Subroutine Library と呼ばれるシステムライブラリーの一部分であって，他では，たとえば，CONTEMPT−LTやUSNRC の規制用コードパッケージであるWREM の主要部分にも使用されている。従って，これをFACOMに変換しておく事は，今後導入コードを整備する上で極めて好都合であるばかりでなく，新たに大規模な計算を行なうプログラムを開発する際にも応用できる。

このシステムルーティンの特長として，次の2点が挙げられる。

(1) ある論理的にまとまったデータ群に対し，データ番号（RELAP−4のマニュアルではカード番号と呼ばれ，以下これを用いる）を付して入力データの処理を行なう。この方式により，

i) データ群の入力の順序は自由である。

ii) 1ケース内でのデータ群の置き換えが容易であり，更に，複数個のケースについては，必要なデータ群のみをカード番号をキーとして置き換えるだけでよい。

などの利点がある。

(2) 個々のカード番号，及びそれぞれの番号に従う個々のデータについて，プログラムが要求する番号やデータの個数と型（整数型，実数型，文字型など）と比較チェックし，更に不足カード番号や過剰カード番号も検出する。このような両面からのインプットのチェックが，1ケース分のデータ全体について行なわれる事は，RELAP−4が主として対象とするような大規模な計算に際しては，有効である。

RELAP−4の入力部分を構成するルーティンは，オリジナルバァーションでは，

INP, INP2, INP8, LINK, MODER, DCVIC, INPPCK, INPUPK

であり，LINK, DCVIC, INPPCK, INPUPKはAssemblerで書かれている。それぞれのルーティンの機能については後述するがFACOMへ変換整備に当って，

i) オリジナル機能をできるだけ損なわないようにする。

ii) Assembler ルーティンをFORTRAN化する事により，計算機間の互換性と将来の修正・追加変更を容易にする。

の2点を目標とした。このためにまず，FORTRAN ルーティンはできるだけそのままにし，次にAssemblerルーティンの機能をFORTRANでプログラムした。この際，キャラクター，及びビットをハンドリングするためにマスキングやシフト演算を基本機能として必要としたが，それらをFORTRANでシミュレーションするために，新たに，サブルーティンMCONST, IAPICK, IAPACK, ALPACK を作成した。この中MCONSTは，上記の演算に必要な8進数をまとめて定義したもので，これを変更するだけでインプットルーティンを他の計算機でも使用できるように工夫した。このようにして作成されたFACOM 用汎用インプットルーティンのソースリストは，付録1に示してある。

2.2 入力データ処理の手順

入力データ処理は，

(1) 1ケース分のデータをプログラム本体に用意した大きいwork area（倍精度）へ，一括して読み込む。

(2) プログラム中の別のarea（倍精度）へ，カード番号をキーとしてwork areaからデータを拾い出し転送する。

(3) このareaからプログラムの実際の変数へデータを代入する。

の3段階に分けて行なわれる。このような，やや複雑な手順を用いる理由は，RELAP－4プログラムに現われる変数の型が整数（単精度）と倍精度の実数の二種類であるので，1つのカード番号に対応する一連のデータと一括して倍精度として扱い，その後で変数の型に合わせてふりわける必要があるためである。以下，その手順の詳細を述べる。

(1) **第1段階** プログラム本体の中にwork areaとして大きいarray（data arrayと呼ぶ）を倍精度で用意しておき，1ケース分のデータを次の手順でarrayにストアする。

i）一枚のカード上にformat freeで書かれたデータを互いに識別し，それぞれのデータの型（整数型，実数型，文字型）を決める。その型に応じて値を倍精度のバイナリー型式で表現する。この時，カード上の最初のデータは，もし，そのカードが継続記号で始まっていなければ，カード番号と見なされ，以下がそれに従うデータと見なされる。そして，データにエラーがあれば，そのカード番号に対してエラーフラグが立てられる。

ii）上述のバイナリーで表現された値と型に関する情報を，arrayの先頭の1倍精度語を空けて，前方からストアしてゆく。この時，型に関する情報は，それぞれ2ビットで表現され，それらが倍精度語の中に圧縮された後，データ値の後方に続けられる（第1図参照）。2ビット表現は，次のような分類になっている。

					ビット表現
データ	数値データ	$=0$		---	00
		$\neq 0$	整数	---	01
		$\neq 0$	実数	---	10
	文字データ			---	11

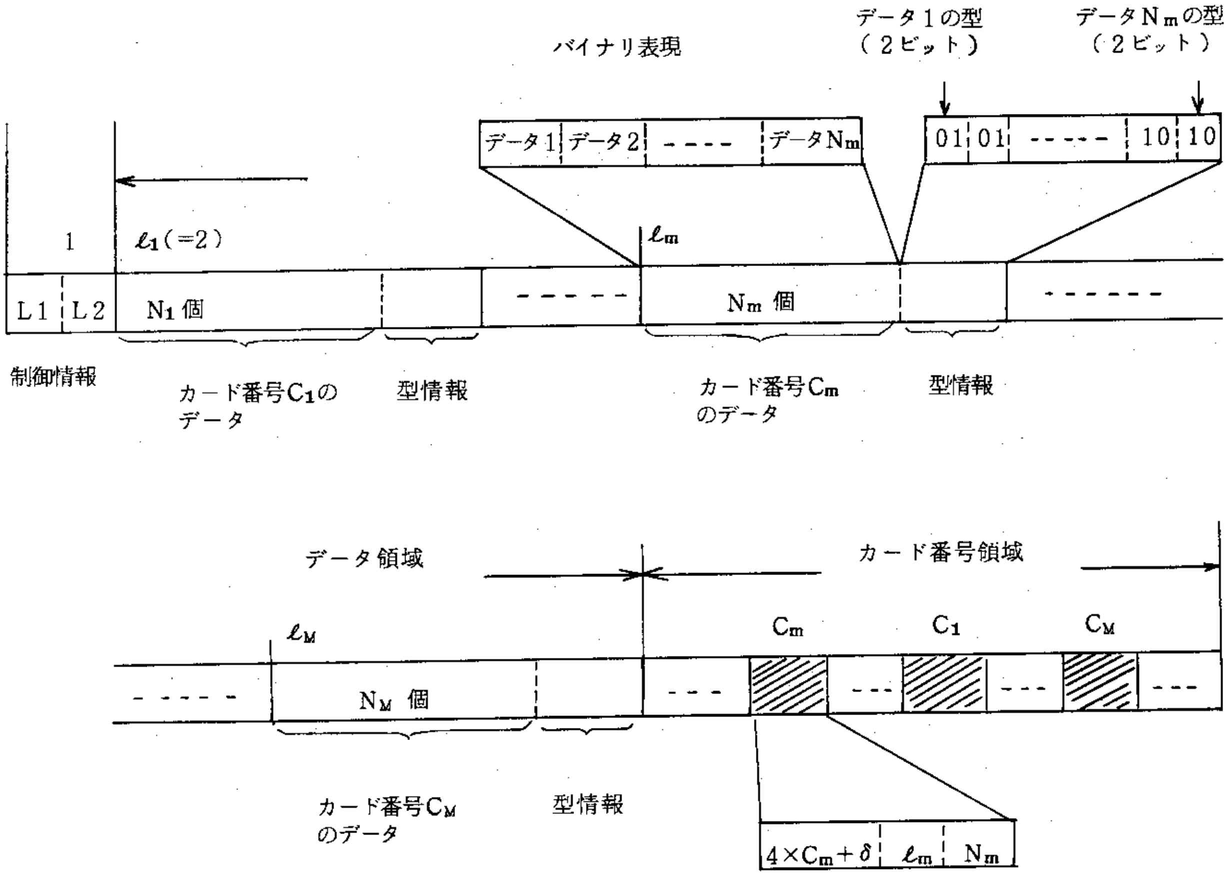
iii）カード番号とそれに対応するデータの個数，及びストアした一連のデータのarrayにおける開始位置の情報をarrayの後方からストアしてゆく。この時，カード番号とエラー情報は合成された単精度整数で表わされ，データの個数とarrayでの開始位値はそれぞれ$\frac{1}{2}$語の整数として表現され，全体として1倍精度語に納められる。前述の合成整数は，カード番号を4倍し（2ビットシフトし），それにエラーフラグを付け加えたものである。

iv）データカード毎に上述の操作をくり返すが，同じカード番号が現われたら，後から入力されたカードの情報がarrayの中で置き換わる。

1ケース分のデータの入力が終ったら，arrayの中の値などが入れられている前方のデータ領域とカード番号などが入っている後方のカード番号領域の間にできる空白領域をとり除き，arrayの先頭に空けておいた1倍精度語の前半にデータ領域の長さを，後半にカード番号領域の長さを

制御情報としてストアする。更に，カード番号領域を番号の大きさでソートする。（最小番号が最後尾にくる）

カード番号C_1，…，C_m，…，C_Mからなるデータセットが，この順序で入力された場合のdata array の構造を第1図に示す。



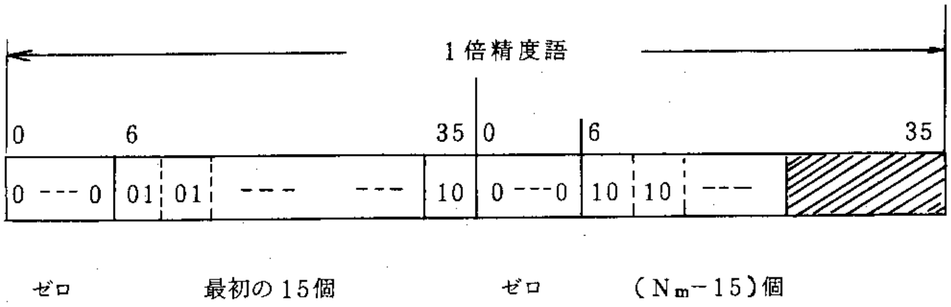
第1図　data array の構造

第1図で，

L1，L2	制御情報
ℓ_1，ℓ_m，ℓ_M	倍精度で数えたarray上の位置
N_1，N_m，N_M	データの個数
δ	エラーフラグ $\begin{cases} 0 & \text{エラーなし} \\ 1 & \text{エラーあり} \end{cases}$

$C_m > C_1 > C_M$

型情報は，1倍精度語には，前半分に15個（30ビット），後半分に15個（30ビット），合計30個入れられるようになっている。その時の様子をカード番号C_mについて，$15 < N_m < 30$ の場合を例にとって拡大して書くと，第2図のようになる。



第 2 図

前半分と後半分の先頭の使用しない部分にはゼロが入れられる。原機種IBMでは，それらは2ビットづつであるが，FACOMでは1語長がIBMより長いのでゼロの部分が増えている。

一般にN_m 個の型情報をストアするのに必要な語数は，ガウスの記号を用いれば$\left[\frac{N_m+29}{30}\right]$であり，arrayの先頭には制御情報として，

$$L1 = \sum_{m=1}^{M} \left(N_m + \left[\frac{N_m+29}{30}\right] \right)$$

$$L2 = M$$

がストアされている。

(2) **第2段階** カード番号をキーとしてdata array からプログラム中の別のareaへのデータの転送は，第3図に示すようにサブルーティンINP2をCALLする事によって行なう。この時サブルーティンには，1回の転送についてカード番号，入力されるべきデータの最小個数と最大個数，データが持つべき型などの指定をした整数array (format array と呼ぶ)と，data array からデータが一時的に転送されるべきarea (buffer arrayと呼ぶ)とを与えねばならない。このformat arrayのある要素は，サブルーティンからの情報を受ける場所としても使用される。

format array をL(1)，…，L(n)とした時，各要素の意味を略述する。

L(1)　最初のカード番号

L(2)　最終のカード番号。カード番号がL(1)とL(2)の間にあるデータの読み込みを指定し，もし，L(2)= 0 ならば，L(1)の番号を持つデータの読み込みだけを意味する。

L(3)　データの最小個数

L(4)　データの最大個数

L(5)　buffer array へデータをストアする時の間隔(通常は0である)

L(6)　buffer array へデータを転送する時の開始位置

L(7)，L(8)，……

　　データが持つべき型(整数型，実数型，文字型)

このformat arrayの指定にもとづいて，data arrayからbuffer arrayへのデータの転送は，以下の手順で行なわれる。

i）指定されたカード番号をdata arrayのカード番号領域で探し，存在すれば対応するデータがストアされている場所（開始位置）と個数（それぞれ$\frac{1}{2}$語に収められている）の情報をとり出す。*

ii）データの開始位置と個数から，カード入力時に作った型情報がストアされている場所が解かり，そこに2ビット表現でパックされている情報をとり出す。

iii）このデータの型とformat arrayのL(7)，L(8)，… で指定した型とを比較チェックする。一致すれば，所定のbuffer arrayにデータの転送を行なう。

iv）以上の手続きをL(1)，L(2)で指定した範囲内のカード番号について繰返す。

(3) **第3段階**　buffer arrayに転送されてきたデータをプログラムの実際の変数にその型に応じて代入する。この時，実数型の変換（倍精度）には単にarrayの要素が代入されるが，整数型の変数（単精度）には，arrayの要素の後半分が代入される。すべてのデータの代入が終了すれば，未処理のデータがあるかどうかをチェックする。

第1,2,3段階に分けて行なわれるインプットデータの処理手順とサブルーティンの関係を第3図に示す。

サブルーティンINP，INP 2，INP 8のCALLの仕方については，文献2に詳しく述べられている。又，インプットデータの書き方については，文献1,2で説明がなされているので本報告書では特にとりあげない。インプットの書き方に関してFACOMへの変換の際に以下の制限と追加を行なった。

1）有効数字が約8ケタ，即ち，単精度整数の範囲以上の仮数部を持つ実数型データを書く事はできない。

2）同じデータの繰返し入力，すなわちxというデータとn回反復して入力したい場合，

n(x)

と書くことが出来る。唯，xが文字型データの場合は，8文字以内でなければならない。

---

* この作業の終了時に，カード番号とエラー情報を合成した整数の符号ビットを0から1にしておき，後で，プログラムが読み込まなかったデータの検出に供する。

- RELAP－4サブルーティン INPUT
  - INP（第1段階）……data array を構築する。
    - DCVIC　format free で書かれたBCDデータを識別し，型を分類し，データ値をバイナリで表現する。
    - INPPCK　データの型を2ビットで表現し，ワード内へパックする。
    - IAPACK　カード番号に対応した一連のデータの data array に於ける開始位置と個数をそれぞれ$\frac{1}{2}$ワードにパックする。
    - IAPICK　上記$\frac{1}{2}$ワードにパックした情報をとり出す。
- RELAP－4インプット読み込みルーティン
  - INP 2（第2段階）………　format array の指定に従い，カード番号に対応するデータを data array から buffer array へ転送する。
    - LINK　format array で指定されたカード番号を探し，対応するデータの data array に於ける開始位置と個数を調べる。
      - IAPICK　（上述）
    - MODER　カード入力時に調べたデータの型と format array で指定した型とを比較する。
      - INPUPK　ワード内に2ビット表現で圧縮されている型情報をとり出す。
  - （第3段階）………………　データをプログラムの変数に，その型に合わせて代入する。
- INP 8　過剰カード番号を検出する。

第3図　入力処理手順とインプットルーティンの関係

3. Restart 及びプロッター機能

3.1. テープアクセスルーティン

事故解析の計算では，一般に現象を非常に細かいタイムステップで追跡してゆくため，計算時間が膨大なものになり，結果として得られるデータの数は莫大である。従って，restart 機能やプロッタの機能が重要なものになる。RELAP－4 にはこれらの機能があるが，オリジナル・バァージョンでは,テープに対するデータの書き込みと読み出しは，Assembler で書かれたNRTS のシステムルーティン，

BUFIN，BUFOUT，BUFDLY，BUFWFF，BUFSKP

を用いて行なわれる。それらの一般形は，

BUFXXX(LU，LOC，N1，IST，N2)

であり，パラメータの意味は，

LU ：ロジカルユニットの番号

LOC：データ転送の対象となるディメンジョンを持つ変数

N1 ：転送されるべきデータの個数

IST：ユニットLUに対する最新のアクセス命令について，その処理状態を示すパラメータ

N2 ：ユニットLUに対するアクセス命令の実行が終了した時，実際に転送されたデータの数

ルーティン毎に，オリジナル機能の概略を述べる。

1) BUFIN

ユニットLUからN1 個のデータを変数LOC に遂次続み込むためのアクセス命令を発生させる。プログラムの実行は先に進むが，このタスクの終了状態は，IST によって示される。

IST ＝ 1 ： アクセス完了

2 ： アクセス中エラー発生

3 ： end of file

4 ： LUがロジカル番号 5, 6, 7.のいずれかである。

2) BUFOUT

LOCからユニットLUへN1個のデータを転送するためのアクセス命令を発生させる。プログラムの実行は先に進むが，このタスクの終了状態は，ISTによって示され，その意味はBUFINの場合と同じである。

3) BUFDLY

ユニットLU に対する最新のアクセス命令の実行が完了するまで，プログラムの実行を止める。

4) BUFWEF

ユニットLUにend of file マークを書く。

5) BUFSKP

このルーティンの場合，LOCはレコード数，N1はファイルの個数を意味し，それらが持つ符号に応じて，ユニットLU 上のファイルの前進後退，更に，ファイル上でレコードの前進後退を行なう。

これらテープI/O ルーティンのすべての機能がRELAP－４で使用されているわけではなく，又，機能の一部には，FORTRAN でシミュレーションする事が困難なものもある。そこで，FACOMへの変換では，BUFDLYの機能を落とし（即ち，ノーオペレーションとし），BUFSKPはrewind機能のみとし，BUFIN，BUFOUT，BUFWEF の主たる機能をFORTRAN化した。

3.2　RELAP－4データテープ

RELAP－４では，restart の為の情報とプロッターの為の情報を１本のテープに編集するようになっている。このテープはRELAP－４データテープと呼ばれ，それを用いての restart やプロッターテープ作成には，テープの構造に応じた特別な読み方が必要である。データテープの標準的な構造を図で示す。

NREST=1 ↓

HEADER LABEL	TITLE	パラメータ	LB 1	プロット 情報	-----	LB 1	プロット 情報	LB 2	restart 情報

NREST=2 ↓

LB 1	プロット 情報	----	LB 1	プロット 情報	LB 2	restart 情報		----		LB 3	LB3	EOF

HEADER LABEL ： 文字型データで次の内容をもつ。

"RELAP4 DATA TAPE. TITLE FOLLOWS.␣␣␣␣
RELAP4/002 04/16/74␣ ←date →"（16×4バイト）

TITLE ：カードよりインプットするタイトルの前部 （文字型／18×4バイト）

パラメータ：次の17個の problem dimension パラメータ（整数）

LDMP, NEDI, NTC, NTRP, NVOL, NBUB, NTDV, NJUN, NPMPC, NCKV, NLK, NFLL, NSLB, NGOM, NMAT, NCOR, NHTX,

LB1 ： プロット情報識別ラベルで次の内容をもつ。

"PLOT", "{TITLE}", NVOL, NJUN

（文字型／4バイト，文字型／18×4バイト，整数×２）

LB2 ： restart 情報制御データで次の内容をもつ。

NREST, NPLOT, "{TITLE}"

（整数×２，文字型／24×4バイト）

ここで，

NREST ： restart レコードの番号

NPLOT ： プロット情報の番号

LB3 ： TRAILER LABELで次の内容をもつ。

"END OF RELAP4 DATA. EOF FOLLOWS."

（文字型／8×4バイト）

3.3 プロッタープログラム PLOTR－4

すでに述べたように，RELAP－4は，restartやプロッターの為の情報を1本のテープに出力する。1つのプロットレコードに納められている情報は，種々の物理量について，ある時間メッシュ点での計算結果を一括したものであり，それが適当なメッシュ間隔で出力される。更に，図で示したように，所々restartレコードが挿入される。PLOTR－4は，このような構造をもつデータテープから，プロットレコードを選び，そのレコードの中からプロットしたい物理量の値をとり出し，別にディスク上にとった領域へ書き込んでゆく。この操作は，いくつかの物理量について同時に行なわれる。プログラムは，こうして編集し直したデータを最終的に1本のプロット用テープに書き出す。

オリジナルバァーションはマイクロフィルムにも使用できるが，FACOM への変換では，この機能はダミーとした。PLOTR－4は，インプットの簡略化のために，種々のパラメータに標準値が内蔵されていること，横軸（時間軸）指定カードを種々の物理量について共用できること，自動スケーリングを行なうこと，などの工夫がなされている。1回のPLOTR－4による作図は，最大20 個の物理量までに制限されている。

4　蒸気表の変換

RELAP－4は，冷却材喪失事故の解析計算に必要な水，蒸気の物性値をバイナリー形式で書かれた表として読むようになっており，オリジナル・パッケージには，IBM 360のバイナリー形式で作られた蒸気表が付加されている。この蒸気表はFACOMではアクセスできないため，RELAP－4の蒸気表読み込みルーティンSTH2OIを用いた簡単なプログラムを作成し，IBM 360を使用してバイナリーデータをBCDデータに変換しテープに納めた。データはすべてSIユニット（K，J，m，kg）で，テープはカード形式に編集した。

蒸気表の内容を以下に述べる。

独立変数は温度と圧力でそれぞれのメッシュ点の数は，

温度　　47点（臨界温度以下40点）

圧力　　29点（臨界圧力以下27点）

である。納められている物性量は，

v　　比体積　（m^3／kg）

u　　比内部エネルギー（J／kg）

$\beta = \frac{1}{v}\left(\frac{\partial v}{\partial T}\right)_P$　　熱膨張係数　（K^{-1}）

$\kappa = \frac{1}{v}\left(\frac{\partial v}{\partial P}\right)_T$　　等温圧縮率　（m^3／J）

$C_p = \left(\frac{\partial h}{\partial T}\right)_P$　　等圧比熱　（J／kg K）

3.3 プロッタープログラム PLOTR-4

すでに述べたように，RELAP-4は，restartやプロッターの為の情報を1本のテープに出力する。1つのプロットレコードに納められている情報は，種々の物理量について，ある時間メッシュ点での計算結果を一括したものであり，それが適当なメッシュ間隔で出力される。更に，図で示したように，所々restartレコードが挿入される。PLOTR-4は，このような構造をもつデータテープから，プロットレコードを選び，そのレコードの中からプロットしたい物理量の値をとり出し，別にディスク上にとった領域へ書き込んでゆく。この操作は，いくつかの物理量について同時に行なわれる。プログラムは，こうして編集し直したデータを最終的に1本のプロット用テープに書き出す。

オリジナルバァーションはマイクロフィルムにも使用できるが，FACOM への変換では，この機能はダミーとした。PLOTR-4は，インプットの簡略化のために，種々のパラメータに標準値が内蔵されていること，横軸（時間軸）指定カードを種々の物理量について共用できること，自動スケーリングを行なうこと，などの工夫がなされている。1回のPLOTR-4による作図は，最大20個の物理量までに制限されている。

4 蒸気表の変換

RELAP-4は，冷却材喪失事故の解析計算に必要な水，蒸気の物性値をバイナリー形式で書かれた表として読むようになっており，オリジナル・パッケージには，IBM 360のバイナリー形式で作られた蒸気表が付加されている。この蒸気表はFACOMではアクセスできないため，RELAP-4の蒸気表読み込みルーティンSTH2OIを用いた簡単なプログラムを作成し，IBM 360を使用してバイナリーデータをBCDデータに変換しテープに納めた。データはすべてSIユニット（K，J，m，kg）で，テープはカード形式に編集した。

蒸気表の内容を以下に述べる。

独立変数は温度と圧力でそれぞれのメッシュ点の数は，

温度　　47点（臨界温度以下40点）

圧力　　29点（臨界圧力以下27点）

である。納められている物性量は，

v　　比体積　（m^3／kg）

u　　比内部エネルギー（J／kg）

$\beta = \frac{1}{v}\left(\frac{\partial v}{\partial T}\right)_P$　　熱膨張係数　（K^{-1}）

$\kappa = \frac{1}{v}\left(\frac{\partial v}{\partial P}\right)_T$　　等温圧縮率　（m^3／J）

$C_p = \left(\frac{\partial h}{\partial T}\right)_P$　　等圧比熱　（J／kg K）

である。表は，飽和（飽和水，飽和蒸気）に関するものと，単相（未飽和水，過熱蒸気）に関するものに分けられ，それぞれについて，

1）飽和に関しては，

温度の関数としての表

圧力の関数としての表

2）単相に関しては，

温度と圧力の2変数の関数としての表

が与えられている。

5. MKS単位系オプションと実験相関式オプションの追加

安全工学第1研究室では，RELAP－4のCDC バァーションに，MKS単位系オプションと実験相関式オプションを追加した。本コードのFACOM への変換完了後，これらの追加部分を組入れたバァーションを作成した。

5.1. MKS 単位系オプションの追加

RELAP－4で採用されている単位系は，入力，及び出力がFLBS 単位系，内部での計算がSI単位系（k，J，m，kg）である。このうち，入力，及び出力部分にMKS 単位系をオプショナルに追加した。但し，補助記憶装置（テープ，ディスク）への入出力時の単位系は元のままである。これに伴なうインプットの追加の変更を以下に述べる。

1）カード番号010005　MKS 単位系オプションカードでもし与えられなければ，入出力ともFLBSとなる。

ワード1　INUNIT　入力の単位系

0：FLBS，1：MKS

ワード2　MOUNIT　出力の単位系

0：FLBS，1：MKS

2）カード番号010001の4番目のワードを,テープエディットジョブの場合の出力単位系オプションとして用いた。

ワード4　MOUNIT　出力の単位系

0：FLBS，1：MKS

3）プロッタープログラムPLOTR－4の単位系オプションとして，タイトルカードの後に，新たに次のカードを付け加えた。

$$\text{カラム}1 = \begin{cases} 0 : \text{FLBS} \\ 1 : \text{MKS} \end{cases}$$

である。表は，飽和(飽和水，飽和蒸気)に関するものと，単相(未飽和水，過熱蒸気)に関するものに分けられ，それぞれについて，

1) 飽和に関しては，

温度の関数としての表

圧力の関数としての表

2) 単相に関しては，

温度と圧力の2変数の関数としての表

が与えられている。

5. MKS単位系オプションと実験相関式オプションの追加

安全工学第1研究室では，RELAP-4のCDC バァーションに，MKS単位系オプションと実験相関式オプションを追加した。本コードのFACOM への変換完了後，これらの追加部分を組入れたバァーションを作成した。

5.1. MKS単位系オプションの追加

RELAP-4で採用されている単位系は，入力，及び出力がFLBS 単位系，内部での計算がSI単位系(k, J, m, kg)である。このうち，入力，及び出力部分にMKS 単位系をオプショナルに追加した。但し，補助記憶装置(テープ，ディスク)への入出力時の単位系は元のままである。これに伴なうインプットの追加の変更を以下に述べる。

1) カード番号010005　MKS 単位系オプションカードでもし与えられなければ，入出力ともFLBSとなる。

ワード1	INUNIT	入力の単位系
		0：FLBS，1：MKS
ワード2	MOUNIT	出力の単位系
		0：FLBS，1：MKS

2) カード番号010001の4番目のワードを，テープエディットジョブの場合の出力単位系オプションとして用いた。

ワード4	MOUNIT	出力の単位系
		0：FLBS，1：MKS

3) プロッタープログラムPLOTR-4の単位系オプションとして，タイトルカードの後に，新たに次のカードを付け加えた。

$$カラム1 = \begin{cases} 0：FLBS \\ 1：MKS \end{cases}$$

5.2 実験相関式オプションの追加

追加された相関式は，以下のものである。

1) 気液分離モデルに関するWilsonの相関式

オリジナルバァーションでは，過途状態の計算を通じていくつかの一定の気泡上昇速度V_Bしか与える事ができないので，新たにWilsonのV_B相関式が使えるようにした。これに伴ない，カード番号06×××1に次のようなインプを追加した。

ワード1	ALPH	- 気泡分布勾配
ワード2	VBUB	＞0：気泡上昇速度（一定）
		＜0：Wilson相関式
ワード3	JJVOL	Wilson相関式を用いて計算されたV_Bの値を出力したいボリュームの番号

2) 放出係数C_Dに関する相関式

オリジナルバァーションでは，放出係数C_Dとしては，ジャンクション毎に定数を入力し，過渡状態計算を通じて，それらの値が用いられる。そこで，新たに安全工学第1研究室において種々の実験結果に基づいて開発された，クォリティ依存性をとり入れたC_Dの相関式（文献3）を，オプショナルに組み込んだ。それに伴ない，次のインプットを追加した。

カード番号　08×××Yのワード1～18の後，

ワード19	ICD	＝0：クォリティ依存性を考慮しない。
		＝1：クォリティ依存性を考慮する。

謝　　辞

MKS単位系の導入と相関式の追加作業には，安全コード開発室・鈴木重直氏の協力を得たので，ここに謝意を表する。

参　考　文　献

1) K. V. Moore, W. H. Rettig ; RELAP－4 A Computer Program for Transient Thermal － Hydraulic Analysis, ANCR－1127.

2) NRTS Enviromental Subrontine Mannal (a part of CONTEMPT－LT write-up (EW 340))

3) 島宗 他，JAERI－M 6318.

付　録

```
      SUBROUTINE MCONST(LIM,M1,M2,M3,M4,M5,M6,M7,M8)
      LIM= 17777777770
      M1=3777777777770
      M2=3777777777760
      M3=      3777770
      M4=0000000000020
      M5=3777777777740
      M6=0000000000010
      M7=4000000000000
      M8=0000000000030
      RETURN
      END
      SUBROUTINE INP (L1,NL1,HED,NCASE,NDATA,ISW)
      DOUBLE PRECISION L1(1),HED(12)
C     DOUBLE PRECISION BCD(10),BLANK/'  '/,A1,A2,ERB(10),DLRS/'********'
C    * PTRN/Z7FFFFFFFFFFFFFFF/,PTRN2/Z7FFFFFFFFFFFFFFE/
      DOUBLE PRECISION BLANK,A1,A2,DLRS,A3
      DIMENSION BCD(80),BCDL(80),ERB(80),LERB(80)
C     INTEGER IA1(2),IA2(2),MSK1/Z7FFFFFFF/,MSK2/Z7FFFFFFE/,
C    * LIM/Z1FFFFFFF/,IKP/0/
C     INTEGER*2 HA1(4),HA2(4)
      INTEGER IA1(2),IA2(2),HA1(4),HA2(4),IA3(2)
      LOGICAL NUSE,ND,NC,CERR,NX,EOF
C     LOGICAL*1 BCDL(80),BLNK,SLS/'/'/,EQS/'='/,PLS/'+'/,AST/'*'/,
C    * LDLRS,CMA/','/,LERB(80),PER/'.'/
C     EQUIVALENCE (A1,IA1(1),HA1(1)),(A2,IA2(1),IA21,HA2(1)),
      EQUIVALENCE (A1,IA1(1)),(A2,IA2(1),IA21),(ERB(1),LERB(1)),
     1 (BCD(1),BCDL(1)),(A3,IA3(1))
C    * (BLANK,BLNK),(DLRS,LDLRS),(ERB(1),LERB(1)),(BCD(1),BCDL(1))
      DATA BLANK,DLRS,SLS,EQS,PLS,AST,LDLRS,BLNK,CMA,PER
     1 /'        ','********','/','=','+','*','*',' ',',','.'/
      DATA IKP,HA1(1),HA1(2),HA2(1),HA2(2)/0,2*0,2*0/
      COMMON/MCNST/LIM,MSK1,MSK2,MSK3,MSK4,MSK5,MSK6,MSK7,MSK8
C
      CALL MCONST(LIM,MSK1,MSK2,MSK3,MSK4,MSK5,MSK6,MSK7,MSK8)
C
      DO 10 I = 1,11
   10 HED(I) = BLANK
      CALL DATE (HED(12))
C     CALL HEADER (-2)
      CALL HEADER (-2,IDUM1,IDUM2)
C     CALL HEADER (-3)
      CALL HEADER (-3,IDUM1,IDUM2)
      NCASE = NCASE + 1
      NUSE = .TRUE.
      EOF = .FALSE.
      IF (NDATA .GT. 0) NUSE = .FALSE.
      ISW = 0
      NT = NL1
      NB = 2
      IF (NUSE) GO TO 20
      A2 = L1(1)
      NT = NT - IA2(1)
      NB = NB + IA2(2)
      IF (NB .GE. NT) GO TO 901
      N1 = NL1
      N2 = IA2(1) + IA2(2) + 1
      DO 12 I = 1,IA21
      A1 = L1(N2)
      IA1(1) = IAND(IA1(1),MSK1)
      L1(N1) = A1
      N1 = N1 - 1
      N2 = N2 - 1
   12 CONTINUE
C
   20 NCN = 0
      NW = 0
      NX = .TRUE.
      NUSE = .TRUE.
      CERR = .FALSE.
      ND = .TRUE.
      N1 = NB
      N2 = (NT+NB)/2
      N3 = N2
   21 NC = .TRUE.
      READ (5,1100,END=902) BCD
 1100 FORMAT (80A1)
      IF (NCN .EQ. 0) WRITE (6,1001) NCASE
 1001 FORMAT ('1LISTING OF INPUT DATA FOR CASE'I3/' ')
      NCN = NCN + 1
      DO 22 I = 1,80
      IF (BCDL(I) .NE. BLNK) GO TO 23
   22 CONTINUE
      GO TO 30
   23 IF (BCDL(I) .NE. PLS) GO TO 24
      IF (ND) GO TO 903
      IKP = 1
      NC = .FALSE.
      GO TO 50
   24 IF (BCDL(I).EQ.AST .OR. BCDL(I).EQ.LDLRS) GO TO 30
      IF (BCDL(I) .NE. EQS) GO TO 25
      BCDL(I) = BLNK
C     DO 26 N6 = 1,10
C  26 HED(N6) = BCD(N6)
      CALL ALPACK(HED,BCD,80)
      BCDL(I) = EQS
      GO TO 30
   25 IF (BCDL(I) .EQ. SLS) GO TO 27
      IF (BCDL(I) .NE. PER) GO TO 28
  922 NCASE = -NCASE
   27 NUSE = .FALSE.
      GO TO 30
   28 NC = .FALSE.
      NX = .FALSE.
   30 IF (ND) GO TO 50
      IA1(1) = 0
```

```
      IF (CERR) IA1(1) = 1
      A2 = L1(N2)
      IF (IA2(2) .NE. 1) GO TO 904
      A2 = L1(NB)
      IF (IA2(2).LE.0 .OR. IA2(2).GT.LIM) GO TO 904
      N5 = 4*IA2(2)
      IA1(1) = IA1(1) + N5
  941 HA1(3) = NB
      NW = NW - 1
      HA1(4) = NW
      CALL IAPACK(HA1(3),HA1(4),IA1(2))
      L1(NT) = A1
      IF (NW .EQ. 0) GO TO 45
      N3 = N2
      DO 32 I = 1,NW
      A2 = L1(N3+1)
      IF (IA2(2).GE.0 .AND. IA2(2).LE.2) GO TO 35
      IA2(1) = 0
      IA2(2) = 3
      L1(N3+1) = A2
   35 L1(NB) = L1(NB+1)
      N3 = N3 + 1
      NB = NB + 1
   32 CONTINUE
      N3 = NW
   37 N4 = N3
      IF (N4.GT.30) N4 = 30
      CALL INPPCK (L1(NB),N4,L1(N2+1))
      NB = NB + 1
      N2 = N2 + 30
      N3 = N3 - 30
      IF (N3 .GT. 0) GO TO 37
   45 IF (N5 .EQ. 0) GO TO 49
      IF (NT .EQ. NL1) GO TO 46
      N4 = NT + 1
      DO 39 I = N4,NL1
      A1 = L1(I)
      CALL IAPICK(HA1(3),HA1(4),IA1(2))
      IF (IAND(IA1(1),MSK2) .EQ. N5) GO TO 38
   39 CONTINUE
   46 IF (NW .EQ. 0) GO TO 40
   49 NT = NT - 1
      GO TO 40
   38 WRITE (6,3001)
 3001 FORMAT (10X'CARD ABOVE IS REPLACEMENT CARD.')
      N6 = HA1(4) + (HA1(4)+29)/30
      N3 = NB - 1
      IF (HA1(4) .EQ. NW) GO TO 42
      L1(I) = L1(NT)
      NB = HA1(3)
      N5 = NB + N6
      IF (N5 .GT. N3) GO TO 47
      DO 41 N2 = N5,N3
      L1(NB) = L1(N2)
      NB = NB + 1
   41 CONTINUE
      DO 43 N2 = N4,NL1
      A2 = L1(N2)
      CALL IAPICK(HA2(3),HA2(4),IA2(2))
      IF (HA2(3) .LE. HA1(3)) GO TO 43
      HA2(3) = HA2(3) - N6
      CALL IAPACK(HA2(3),HA2(4),IA2(2))
      L1(N2) = A2
   43 CONTINUE
      IF (NW .NE. 0) GO TO 40
   47 L1(I) = L1(N4)
      NT = N4
      GO TO 40
   42 A2 = L1(NT)
      HA2(3) = HA1(3)
      CALL IAPACK(HA2(3),NW,IA2(2))
      L1(I) = A2
      NB = NB - N6
      N5 = HA1(3)
      DO 44 I = NB,N3
      L1(N5) = L1(I)
      N5 = N5 + 1
   44 CONTINUE
   40 CERR = .FALSE.
      ND = .TRUE.
      N1 = NB
      N2 = (NT+NB)/2
      N3 = N2
      NW = 0
   50 IF (EOF) GO TO 61
      WRITE (6,1101) NCN,BCD
 1101 FORMAT (I6,4X80A1)
      IF (NC) GO TO 60
      IF (IKP .EQ. 0) GO TO 51
      BCDL(IKP) = BLNK
      IKP = 0
   51 N4 = 1
      IF (N1+40 .GE. N2) GO TO 905
      CALL DCVIC (BCD,L1(N1),L1(N3),N5,N4)
      IF (ND) NCX = NCN
      IF (N5 .EQ. 0) GO TO 59
      ND = .FALSE.
C     DO 58 I = 1,N5
C     IF (L1(N1) .NE. PTRN) GO TO 58
C     L1(N1) = PTRN2
C     WRITE (6,2011) I,PTRN,PTRN2
C2011 FORMAT (' ********WORD'I3,' HAS UNALLOWED BIT PATTERN('Z16,'), HA
C    *S BEEN CHANGED TO'Z17)
C  58 N1 = N1 + 1
      N1=N1+N5
      N3 = N3 + N5
      NW = NW + N5
   59 IF (N4 .EQ. 0) GO TO 21
```

```
      ND = .FALSE.                                                      INP02030
      ISW = 2                                                           INP02040
      CERR = .TRUE.                                                     INP02050
      L1(N1) = DLRS                                                     INP02060
      IA1(2) = 3                                                        INP02070
      L1(N3) = A1                                                       INP02080
      N1 = N1 + 1                                                       INP02090
      N3 = N3 + 1                                                       INP02100
      NW = NW + 1                                                       INP02110
      DO 52 I = 1,80                                                    INP02120
   52 ERB(I) = BLNK                                                     INP02130
      LERB(N4) = LDLRS                                                  INP02140
      WRITE (6,1102) ERB,N4                                             INP02150
 1102 FORMAT (' '9X80A1,4X'* POINTS TO CARD ERROR AT COL.'I2)           INP02160
      DO 53 I = 1,80                                                    INP02170
      LERB(1) = BCDL(I)                                                 INP02180
      BCDL(I) = BLNK                                                    INP02190
      IF (I .LT. N4) GO TO 53                                           INP02200
      IF (LERB(1).EQ.BLNK .OR. LERB(1).EQ.CMA) GO TO 51                 INP02210
   53 CONTINUE                                                          INP02220
      GO TO 21                                                          INP02230
   60 IF (NUSE) GO TO 21                                                INP02240
   61 IA1(1) = NL1 - NT                                                 INP02250
      IA1(2) = NB - 2                                                   INP02260
      NDATA = IA1(1) + IA1(2) + 1                                       INP02270
      IF (NX) NDATA = -NDATA                                            INP02280
      L1(1) = A1                                                        INP02290
      NT = NT + 2                                                       INP02300
      DO 54 I = NT,NL1                                                  INP02310
C     IF (L1(I) .LE. L1(I-1)) GO TO 54                                  INP02320
      A2=L1(I)                                                          INP02330
      A3=L1(I-1)                                                        INP02340
      IF (IA2(1).LE.IA3(1)) GO TO 54                                    INP02350
      A1 = L1(I)                                                        INP02360
      N1 = I                                                            INP02370
   56 L1(N1) = L1(N1-1)                                                 INP02380
      N1 = N1 - 1                                                       INP02390
      IF (N1 .LT. NT) GO TO 55                                          INP02400
C     IF (A1 .GT. L1(N1-1)) GO TO 56                                    INP02410
      A2=L1(N1-1)                                                       INP02420
      IF (IA1(1).GT.IA2(1)) GO TO 56                                    INP02430
   55 L1(N1) = A1                                                       INP02440
   54 CONTINUE                                                          INP02450
      NT = NT - 1                                                       INP02460
      DO 70 I = NT,NL1                                                  INP02470
      L1(NB) = L1(I)                                                    INP02480
   70 NB = NB + 1                                                       INP02490
      CALL HEADER (2,96,HED)                                            INP02500
      WRITE (6,2000)                                                    INP02510
 2000 FORMAT ('1')                                                      INP02520
      RETURN                                                            INP02530
  901 WRITE (6,2001)                                                    INP02540
 2001 FORMAT (' ********INSUFFICIENT STORAGE ALLOCATION FOR PREVIOUS DAINP02550
     *TA, PROCESSING TERMINATED.')                                      INP02560
  911 ISW = 3                                                           INP02570
      RETURN                                                            INP02580
  902 IF (NCN .NE. 0) GO TO 921                                         INP02590
      ISW = 1                                                           INP02600
      RETURN                                                            INP02610
  921 WRITE (6,2002)                                                    INP02620
 2002 FORMAT (' ********END OF FILE ENCOUNTERED BEFORE END(.) CARD.')   INP02630
      EOF = .TRUE.                                                      INP02640
      ISW = 2                                                           INP02650
      GO TO 922                                                         INP02660
  903 WRITE (6,2003)                                                    INP02670
 2003 FORMAT (' ********CONTINUATION CARD INDICATED, BUT NO PREVIOUS DAINP02680
     *TA CARD.  TREATED AS NEW DATA CARD.')                             INP02690
      ISW = 2                                                           INP02700
      GO TO 28                                                          INP02710
  904 WRITE (6,2004)                                                    INP02720
 2004 FORMAT (' ********UNRECOGNIZABLE CARD NUMBER')                    INP02730
      IA1(1) = 4*NCX + 2 + IA1(1)                                       INP02740
      N5 = 0                                                            INP02750
      NW = 1                                                            INP02760
      ISW = 2                                                           INP02770
      GO TO 941                                                         INP02780
  905 WRITE (6,2005)                                                    INP02790
 2005 FORMAT (' ********INSUFFICIENT STORAGE FOR DATA, PROCESSING TERMIINP02800
     *NATED.')                                                          INP02810
      GO TO 911                                                         INP02820
      END                                                               INP02830
C     SUBROUTINE INP2 (LOC1,LOC2,ICS)                                   IN200010
      SUBROUTINE INP2 (LOC1,LOC2,ICS,IDUM)                              IN200020
      DOUBLE PRECISION LOC1(1),LOC2(1)                                  IN200030
      INTEGER ICS(1)                                                    IN200040
      INTEGER FIRST,LAST,C,ADD                                          IN200050
     DIMENSION  IDUM(1)                                                 IN200060
C                                                                       IN200070
C    ROUTINE TO TRANSFER DATA FROM INPUT CARD BUFFER TO DESIGNATED      IN200080
C    STORAGE AREA IN CORE                                               IN200090
C                                                                       IN200100
C    SUBROUTINE ARGUMENTS                                               IN200110
C    LOC1(1)         STARTING LOCATION OF INPUT BUFFER -                IN200120
C    LOC2(1)         STARTING LOCATION OF MOVED INFORMATION             IN200130
C    ICS             CONTROL INFORMATION                                IN200140
C    ICS(1) =FIRST,CARD NUMBER TO PROCESS FIRST                         IN200150
C    ICS(2) =LAST ,FINAL CARD NUMBER TO PROCESS                         IN200160
C                    MAY BE 0,0 IF ONLY ONE CARD                        IN200170
C                    ) 0 - CARD NUMBERS SEQUENTIAL                      IN200180
C                    ) 0 - CARD NUMBERS INCREASING, NOT SEQUENTIAL      IN200190
C    ICS(3) =MIN     MINIMUM NUMBER OF DATA WORDS ON CARD               IN200200
C    ICS(4) =MAX     MAXIMUM NUMBER OF DATA WORDS ON CARD -             IN200210
C                    0 IF NO UPPER LIMIT                                IN200220
C    ICS(5) =NJ      DISPLACEMENT OF DATA - DATA STORED EVERY NJ+1 LOC  IN200230
C    ICS(6) =J       ABS(J)=STARTING ADDRESS OF DATA IN LOC2            IN200240
C                    MOVE/CHECK DATA - )0 => MOVE,  )OR=0 => CHECK      IN200250
C                    UPON EXIT, ICS(6) CONTAINS NUMBER OF ITEMS MOVED   IN200260
C                    IF J)0,  -(NUMBER OF ITEMS CHECKED) IF J)0         IN200270
```

```
C
      FIRST=ICS(1)
      LAST =ICS(2)
      MIN  =ICS(3)
      MAX  =ICS(4)
      NJ   =ICS(5)
      J    =ICS(6)
      NLAST=1
      NCASE=1
      NMOVE=1
      IF(LAST) 105,110,120
C     TO PROCESS ANY CARDS BETWEEN FIRST AND LAST  GO TO 120
  105 NCASE=2
      GO TO 120
C     ONE CARD ONLY
  110 LAST=FIRST
  120 C=FIRST
       ADD = 0
C     CHECK IF DATA TO BE MOVED - NO - SET NMOVE=2
      IF(J.LE.0) NMOVE=2
      NC=0
      MGO=1
C     CHECK IF MAX NUMBER OF DATA ITEMS SPECIFIED
C     IF NONE - SET MGO=2
       IF (MAX.EQ.0) MGO = 2
  130 C=C+ADD
C     GET CARD LOCATION
  135 CALL LINK(C,NEXT,LOC,NDATA, LOC1)
C     NDATA           NUMBER OF DATA FIELDS ON CARD
       IF (NCASE .NE. 1) GO TO 170
       IF (NDATA) 160,150,180
  150  IF (NEXT-IABS(LAST)) 155,165,255
  155  IF (NEXT.LT.0) GO TO 255
      GO TO 165
  160  WRITE (6,1000) C
 1000  FORMAT ('0******** ILLEGAL FORMAT ON CARD'I10)
      GO TO 265
  165  WRITE (6,1001) C
 1001  FORMAT ('0******** CARD'I10,' MISSING IN SEQUENCE')
      GO TO 265
C     CHECK DATA COUNT
  170 IF(NDATA ) 160,175,180
C     NO DATA - SET CARD NUMBER TO NEXT HIGHER CARD LOCATION
  175 C=NEXT
      NCARD=2
      IF(C) 255,185,185
  180 NCARD=1
  185  IF (C-IABS(LAST)) 195,190,255
  190 NLAST=2
C     THIS IS THE LAST CARD
  195  IF (NCARD .NE. 1) GO TO 135
       NOM = NC + NDATA
       IF (MGO .NE. 1) GO TO 210
C     TEST NUMBER OF DATA WORDS
       IF (NOM .LE. MAX) GO TO 210
C     TOO MANY WORDS
       WRITE (6,1002) FIRST,C
 1002  FORMAT ('0******** TOO MANY NUMBERS ON CARDS'I10,' THROUGH'I10)
      GO TO 265
C
C     CHECK MODE OF DATA
C
  210  CALL MODER (LOC1,ICS,LOC,NDATA, LTYPE,NC)
      IF(LTYPE.GT.10000) GO TO 225
      IF(LTYPE) 220,230,215
C     FLOATING POINT NUMBER FOUND FOR FIXED POINT
  215  WRITE (6,1003) LTYPE,C
 1003  FORMAT ('0******** WORD'I3,' ON CARD'I10,' SHOULD BE IN INTEGER F
     *ORMAT,')
      GO TO 265
C     FIXED POINT NUMBER FOUND FOR A FLOATING POINT
  220 LTYPE=-LTYPE
       WRITE (6,1004) LTYPE,C
 1004  FORMAT ('0******** WORD'I3,' ON CARD'I10,' SHOULD BE IN FLOATING
     *POINT FORMAT')
      GO TO 265
C     ALPHA-NUMERIC DATA EXPECTED - NOT FOUND
  225 LTYPE=LTYPE-10000
       WRITE (6,1005) LTYPE,C
 1005  FORMAT ('0******** WORD'I3,' ON CARD'I10,' SHOULD BE IN ALPHANUME
     *RIC FORMAT')
      GO TO 265
C     CHECK IF DATA TO BE MOVED TO NEW LOCATION
  230  IF (NMOVE .NE. 1) GO TO 245
C     MOVE TO NEW LOCATION
       L = LOC
       MAXNUM = L + NDATA - 1
      LOCB= J + (NJ+1)*NC
       DO 240 LOCA = L,MAXNUM
       LOC2(LOCB) = LOC1(LOCA)
      LOCB=LOCB + NJ+1
  240 CONTINUE
C     INCREMENT NC
  245 NC=NC+NDATA
C     CHECK FOR LAST CARD
       IF (NLAST .NE. 1) GO TO 255
C     ANOTHER CARD EXPECTED
       ADD = 1
      GO TO 130
C     LAST CARD
  255 IF(NC.GE.MIN) GO TO 270
       LOCA = IABS(LAST)
      IF(NC.GT.0) GO TO 260
       WRITE (6,1006) FIRST,LOCA
 1006  FORMAT ('0******** CARDS'I10,' THROUGH'I10,' MISSING')
      GO TO 265
  260  WRITE (6,1007) FIRST,LOCA
 1007  FORMAT ('0******** TOO FEW NUMBERS ON CARDS'I10,' THROUGH'I10)
```

```
  265  ICS(6) = -1                                                   IN201360
       GO TO 280                                                     IN201370
  270  IF (NMOVE.EQ.1) GO TO 275                                     IN201380
        ICS(6) = -NC                                                 IN201390
       GO TO 280                                                     IN201400
  275  ICS(6) = NC                                                   IN201410
  280  RETURN                                                        IN201420
       END                                                           IN201430
       FUNCTION INP8 (LIST,A)                                        IN800010
C       DOUBLE PRECISION A(1),AST/'*********'/,BL/' '/,AD,N          IN800020
C    INP8  ON RETURN IS THE NUMBER OF EXTRANEOUS CARDS FOUND.        IN800030
C  HERE EXTRANEOUS MEANS NO REFERENCE BY LINK (HENCE, ALSO INP2,4,5) IN800040
C       INTEGER SCR(2),MSK1/Z00000002/,MSK2/Z7FFFFFFC/               IN800050
        DOUBLE PRECISION A(1),AST,BL,AD,N                            IN800060
        INTEGER SCR(2)                                               IN800070
        DATA AST,BL/'*********',' '/                                 IN800080
        COMMON/MCNST/LIM,M1,M2,M3,MSK1,MSK2,M6,M7,M8                 IN800090
        EQUIVALENCE (AD,SCR(1))                                      IN800100
       ICNT=0                                                        IN800110
        AD = A(1)                                                    IN800120
       I1=SCR(2)+2                                                   IN800130
       I2=I1+SCR(1)-1                                                IN800140
       IF (I2 .LT. I1) GO TO 4                                       IN800150
       DO 1 I=I1,I2                                                  IN800160
        AD = A(I)                                                    IN800170
        IF (SCR(1) .LT. 0) GO TO 1                                   IN800180
        IF (ICNT.EQ.0 .AND. LIST.NE.0) WRITE (6,2)                   IN800190
    2 FORMAT(1H0,9H********* ,33HTHE FOLLOWING CARDS WERE NOT USED,//) IN800200
       ICNT=ICNT+1                                                   IN800210
       N=BL                                                          IN800220
        SCR(2) = IAND(SCR(1),MSK1)                                   IN800230
        IF (SCR(2) .NE. 0) N = AST                                   IN800240
        SCR(1) = IAND(SCR(1),MSK2)/4                                 IN800250
        IF (LIST.NE.0) WRITE (6,3) N,SCR(1),N                        IN800260
    3   FORMAT (' 'A8,I10,A8)                                        IN800270
    1 CONTINUE                                                       IN800280
    4   INP8 = ICNT                                                  IN800290
       RETURN                                                        IN800300
       END                                                           IN800310
        SUBROUTINE DCVIC(BCD,BIN,CODE,NUM,NERR)                      DCV00010
C                                                                    DCV00020
        DOUBLE PRECISION BIN(1),CODE(1),BLANK,T1,T2                  DCV00030
        DIMENSION BCD(80),B(81),H(47),S(6),IT1(2),IT2(2)             DCV00040
        EQUIVALENCE (T1,IT1(1)),(T2,IT2(1))                          DCV00050
        DATA H/'0','1','2','3','4','5','6','7','8','9',              DCV00060
     1         '.','+','+','-','D','E','(',')','=',',','/',          DCV00070
     2         'A','B','C','F','G','H','I','J','K','L','M','N',      DCV00080
     3         'O','P','Q','R','S','T','U','V','W','X','Y','Z',' ','''/ DCV00090
        DATA BLANK,BLNK,DLR,AST,CMA/'        ',' ','$','*',','/      DCV00100
        LOGICAL LEXP,LPER,LZER                                       DCV00110
C                                                                    DCV00120
        DO 10 I=1,81                                                 DCV00130
   10   B(I)=BLNK                                                    DCV00140
        DO 11 I=1,80                                                 DCV00150
        IF (BCD(I).EQ.DLR .OR. BCD(I).EQ.AST) GO TO 12               DCV00160
   11   B(I)=BCD(I)                                                  DCV00170
C                                                                    DCV00180
   12   NW=0                                                         DCV00190
        NP=0                                                         DCV00200
C                                                                    DCV00210
   20   NP=NP+1                                                      DCV00220
        IF (NP.GT.81) GO TO 70                                       DCV00230
        C=B(NP)                                                      DCV00240
        IF (C.EQ.BLNK) GO TO 20                                      DCV00250
        IF (C.EQ.CMA)  GO TO 99                                      DCV00260
        NP=NP-1                                                      DCV00270
C                                                                    DCV00280
        NRPT=1                                                       DCV00290
   22   NTYP=0                                                       DCV00300
        LEXP=.FALSE.                                                 DCV00310
        LPER=.FALSE.                                                 DCV00320
        LZER=.TRUE.                                                  DCV00330
        INT=0                                                        DCV00340
        NSGN=1                                                       DCV00350
C                                                                    DCV00360
   30   NC=0                                                         DCV00370
   31   NP=NP+1                                                      DCV00380
        IF (NP.GT.81)  GO TO 99                                      DCV00390
        C=B(NP)                                                      DCV00400
        IF (NTYP.EQ.-2) GO TO  25                                    DCV00410
        IF (C.EQ.BLNK .OR. C.EQ.CMA) GO TO 50                        DCV00420
   25   DO  32  J=1,47                                               DCV00430
        IF (H(J).EQ.C) GO TO 33                                      DCV00440
   32   CONTINUE                                                     DCV00450
        GO TO 99                                                     DCV00460
C                                                                    DCV00470
   33   NG=NTYP+3                                                    DCV00480
        GO TO  (26,35,34,36),NG                                      DCV00490
   34   IF (J.LE.14) GO TO 37                                        DCV00500
        NTYP=-1                                                      DCV00510
        IF (J.NE.47)  GO TO  35                                      DCV00520
        NTYP=-2                                                      DCV00530
        GO TO 30                                                     DCV00540
   35   IF (J.EQ.18) GO TO 50                                        DCV00550
        GO TO 27                                                     DCV00560
   26   IF (J.EQ.47)   GO TO 29                                      DCV00570
   27   NC=NC+1                                                      DCV00580
        IF (NC.GT.8)   GO TO 28                                      DCV00590
        S(NC)=C                                                      DCV00600
        GO TO 31                                                     DCV00610
   28   NP=NP-1                                                      DCV00620
        GO TO 50                                                     DCV00630
   29   NTYP=-1                                                      DCV00640
        GO TO 50                                                     DCV00650
   37   NTYP=1                                                       DCV00660
C                                                                    DCV00670
   36   IF (J.GT.18) GO TO 99                                        DCV00680
        IF (J.GT.10) GO TO 38                                        DCV00690
```

```
      IF (J.EQ.1 .AND. LZER) GO TO 31                                   DCV00700
      INT=INT*10+(J-1)                                                  DCV00710
      LZER=.FALSE.                                                      DCV00720
      NC=NC+1                                                           DCV00730
      GO TO 31                                                          DCV00740
C                                                                       DCV00750
   38 J10=J-10                                                          DCV00760
      GO TO (39,40,40,40,41,41,43,50),J10                               DCV00770
   39 LPER=.TRUE.                                                       DCV00780
      LZER=.FALSE.                                                      DCV00790
      GO TO 30                                                          DCV00800
C                                                                       DCV00810
   40 IF (INT.EQ.0) GO TO 42                                            DCV00820
      IF (LEXP) GO TO 42                                                DCV00830
   41 LEXP=.TRUE.                                                       DCV00840
      NDG=NC                                                            DCV00850
      NSGN1=NSGN                                                        DCV00860
      INT1=INT                                                          DCV00870
      INT=0                                                             DCV00880
      NC=0                                                              DCV00890
      LZER=.TRUE.                                                       DCV00900
      NSGN=1                                                            DCV00910
   42 IF (J.EQ.14) NSGN=-1                                              DCV00920
      GO TO 31                                                          DCV00930
C                                                                       DCV00940
   43 IF (INT.GT.0) GO TO 44                                            DCV00950
      NP=NP-1                                                           DCV00960
      GO TO 99                                                          DCV00970
   44 NRPT=INT                                                          DCV00980
      GO TO 22                                                          DCV00990
C                                                                       DCV01000
   50 IF (NTYP) 51,99,52                                                DCV01010
   51 T1=BLANK                                                          DCV01020
      CALL ALPACK(T1,S,NC)                                              DCV01030
      IT2(2)=3                                                          DCV01040
      GO TO 60                                                          DCV01050
   52 IF (LEXP) GO TO 55                                                DCV01060
      IF (INT.NE.0) GO TO 53                                            DCV01070
      IT1(1)=0                                                          DCV01080
      IT1(2)=0                                                          DCV01090
      IT2(2)=0                                                          DCV01100
      GO TO 60                                                          DCV01110
   53 IF (LPER) GO TO 54                                                DCV01120
      IT1(1)=0                                                          DCV01130
      IT1(2)=NSGN*INT                                                   DCV01140
      IT2(2)=1                                                          DCV01150
      GO TO 60                                                          DCV01160
   54 NDG=NC                                                            DCV01170
      T1=DFLOAT(NSGN*INT)/10.D0**NDG                                    DCV01180
      GO TO 56                                                          DCV01190
   55 T1=DFLOAT(NSGN1*INT1)*10.D0**(NSGN*INT-NDG)                       DCV01200
   56 IT2(2)=2                                                          DCV01210
C                                                                       DCV01220
   60 IT2(1)=0                                                          DCV01230
      DO 61 I=1,NRPT                                                    DCV01240
      NW=NW+1                                                           DCV01250
      BIN(NW) =T1                                                       DCV01260
   61 CODE(NW)=T2                                                       DCV01270
      IF (NTYP.EQ.-2)  GO TO 30                                         DCV01280
      GO TO 20                                                          DCV01290
C                                                                       DCV01300
   70 NUM=NW                                                            DCV01310
      NERR=0                                                            DCV01320
      RETURN                                                            DCV01330
C                                                                       DCV01340
   99 IF (NERR.NE.0) GO TO 100                                          DCV01350
      WRITE(6,1000) NP                                                  DCV01360
 1000 FORMAT(' ******** ERROR AT COLUMN',I3)                            DCV01370
  100 NERR=NP                                                           DCV01380
      NUM=NW                                                            DCV01390
      RETURN                                                            DCV01400
      END                                                               DCV01410
      SUBROUTINE LINK(C,NEXT,LOC,NDATA,LOC1)                            LNK00010
      COMMON/MCNST/LIM,MSK1,MSK2,MSK3,MSK4,MSK5,MSK6,MSK7,MSK8          LNK00020
      INTEGER C                                                         LNK00030
      DOUBLE PRECISION LOC1(1),A                                        LNK00040
      DIMENSION IA(2)                                                   LNK00050
      EQUIVALENCE (A,IA(1))                                             LNK00060
C                                                                       LNK00070
      A=LOC1(1)                                                         LNK00080
      NT=1+IA(2)+IA(1)                                                  LNK00090
      NX=IA(1)                                                          LNK00100
      ISW=1                                                             LNK00110
      NEXT=-1                                                           LNK00120
      DO  1  I=1,NX                                                     LNK00130
      A=LOC1(NT)                                                        LNK00140
      IA(1)=IAND(IA(1),MSK1)                                            LNK00150
      IC=IA(1)/4                                                        LNK00160
      IF (IC.EQ.C)  GO TO  2                                            LNK00170
      IF (IC.LT.C)  GO TO  6                                            LNK00180
      GO TO  (5,6) , ISW                                                LNK00190
    5 NEXT=IC                                                           LNK00200
      ISW=2                                                             LNK00210
    6 NT=NT-1                                                           LNK00220
    1 CONTINUE                                                          LNK00230
      NDATA=0                                                           LNK00240
      GO TO 3                                                           LNK00250
    2 CALL IAPICK(NB,NW,IA(2))                                          LNK00260
      LOC=NB                                                            LNK00270
      NDATA=NW                                                          LNK00280
      IF (IAND(IA(1),MSK6).EQ.1) NDATA=-1                               LNK00290
      IA(1)=IOR(IA(1),MSK7)                                             LNK00300
      LOC1(NT)=A                                                        LNK00310
      IF (I.EQ.NX) GO TO 3                                              LNK00320
      A=LOC1(NT-1)                                                      LNK00330
      IC=IA(1)/4                                                        LNK00340
      IF (IC.LE.C) GO TO 3                                              LNK00350
      NEXT=IC                                                           LNK00360
```

```
    3 RETURN
      END
      SUBROUTINE MODER (LOC1,LOC3,J,M,N,NC)
      DOUBLE PRECISION LOC1(1)
C     INTEGER MODES/30/,LOC3(1),MUD(30)
      DIMENSION LOC3(1),MUD(30)
      DATA MODES/30/
C     MODE IS A SUBROUTINE THAT CHECKS THE PROBABLE MODE OF M NUMBERS AT
C     LOC1(J) AND ON BY USING THE FORMAT ARRAY AT LOC3(7) AND ON.  THE
C     NUMBER OF ITEMS ALREADY PROCESSED (NC) DETERMINES THE STARTING
C     POINT IN LOC3(7).  ON EXIT N=0 IF THE MODE IS IN ALL PROBABILITY
C     CORRECT, N=+ THE NUMBER OF THE ITEM IN ERROR IF IT SHOULD BE FIXED
C     AND IS NOT, N=- THE NUMBER OF THE ITEM IN ERROR IF IT SHOULD BE
C     FLOATING AND IS NOT.
C N .GT. 10000 MEANS ITEM N-10000 SHOULD BE ALPHA-NUMERIC AND IS NOT
      IF (J.LE.0 .OR. M.LE.0 .OR. NC.LT.0) CALL FABEND
      IM=1
      ILM=J+M
      NFUN = 1
      NUM=NC-1
      MA=7
  130 IF (IABS(LOC3(MA)) .GE. 2) GO TO 110
C     FOLL SECTION NOT IN A REPEAT
      IF (NUM .LT. 0) GO TO 121
      NUM = NUM - 1
      MA=MA+1
      GO TO 130
C     KEEP GOING THROUGH UNTIL PAST NO. ALREADY PROCESSED
  121 MD=MA
      NGO=1
      GO TO 120
C     TEST FOR TYPE OF REPEAT
  110 IF (LOC3(MA) .LT. 0) GO TO 235
C     A REPEAT THAT IS RESET FOR EACH CARD
      NGO = 2
  145 MD=MA
      MB=LOC3(MD)
  155 MD=MD+1
      IF (MB .NE. 0) GO TO 120
      GO TO 145
C     DO NOT RESET REPEAT FOR EACH CARD
  235 NGO=3
  135 MD=MA
      MB=-LOC3(MD)
  165 MD=MD+1
      IF (MB .EQ. 0) GO TO 135
      IF (NUM .LE. 0) GO TO 120
      MB = MB - 1
      NUM=NUM-1
      GO TO 165
  120 IF (MOD(IM,MODES).NE.1) GO TO 801
      CALL INPUPK (LOC1(ILM),MODES,MUD)
      IM=1
      ILM=ILM+1
  801 IF (MUD(IM).EQ.0 .AND. LOC3(MD).NE.(-1)) GO TO 440
      NBACK = MUD(IM) - 1
      IF (NBACK .EQ. 2) NBACK=-1
      IF (LOC3(MD) .EQ. NBACK) GO TO 440
C DESIRED MODE DOES NOT MATCH INPUT CONVERSION MODE
      IF (LOC3(MD)) 410,420,430
C SHOULD BE HOLLERITH
  410 N=10000+NFUN
      GO TO 115
C SHOULD BE INTEGER
  420 N=NFUN
      GO TO 115
C SHOULD BE REAL
  430 N=-NFUN
      GO TO 115
  440 IM = IM + 1
      IF (NFUN .GE. M) GO TO 311
      NFUN = NFUN + 1
      IF (NGO .NE. 1) GO TO 320
      MA = MA + 1
      GO TO 130
  320 MB = MB - 1
      IF (NGO .EQ. 2) GO TO 155
      GO TO 165
  311 N=0
  115 RETURN
      END
      SUBROUTINE INPPCK(L1,NW,L2)
      DOUBLE PRECISION L1,L2(1),A,B
      DIMENSION IA1(2),IA2(2)
      EQUIVALENCE (A,IA1(1)),(B,IA2(1))
C
      IF (NW.LE.0 .OR. NW.GT.30) GO TO 3
      IA1(1)=0
      IA1(2)=0
      N=0
      K=(NW+14)/15
      DO 2 J=1,K
      IS=0
      M=15
      IF (J.EQ.K) M=NW-15*(K-1)
      DO 1 I=1,M
      N=N+1
      B=L2(N)
      IS=IS+IA2(2)*2**(2*(15-I))
    1 CONTINUE
      IA1(J)=IS
    2 CONTINUE
      L1=A
      RETURN
    3 WRITE(6,10)
   10 FORMAT(' ********WRONG VALUE FOR THE NO. OF CODE DATA PER WORD')
      RETURN
      END
```

```
      SUBROUTINE INPUPK(L1,IDUM,M)                                      UPK00010
      COMMON/MCNST/LIM,M1,M2,M3,M4,M5,M6,M7,M8                          UPK00020
      DIMENSION M(30),IA(2)                                             UPK00030
      DOUBLE PRECISION L1,A                                             UPK00040
      EQUIVALENCE (A,IA(1))                                             UPK00050
      A=L1                                                              UPK00060
      N=0                                                               UPK00070
      DO 1 J=1,2                                                        UPK00080
      MSK=M8*2**(2*15)                                                  UPK00090
      DO 1 I=1,15                                                       UPK00100
      MSK=MSK/2**2                                                      UPK00110
      N=N+1                                                             UPK00120
    1 M(N)=IAND(IA(J),MSK)/2**(2*(15-I))                                UPK00130
      RETURN                                                            UPK00140
      END                                                               UPK00150
      SUBROUTINE ALPACK(H,B,N)                                          ALP00010
      DOUBLE PRECISION H(1)                                             ALP00020
      DIMENSION B(1)                                                    ALP00030
      ENCODE(N,1,H) (B(I),I=1,N)                                        ALP00040
    1 FORMAT(80A1)                                                      ALP00050
      RETURN                                                            ALP00060
      END                                                               ALP00070
      SUBROUTINE IAPACK(I1,I2,K)                                        IAP00010
      COMMON/MCNST/LIM,M1,M2,M3,M4,M5,M6,M7,M8                          IAP00020
      K=I1*2**18+I2                                                     IAP00030
      RETURN                                                            IAP00040
      ENTRY      IAPICK(I1,I2,K)                                        IAP00050
      I1=IAND(K,M3*2**18)/2**18                                         IAP00060
      I2=IAND(K,M3)                                                     IAP00070
      RETURN                                                            IAP00080
      END                                                               IAP00090
      INTEGER FUNCTION IAND(I,MSK)                                      AND00010
      INTEGER AND                                                       AND00020
      EXTERNAL AND                                                      AND00030
      IAND=AND(I,MSK)                                                   AND00040
      RETURN                                                            AND00050
      END                                                               AND00060
      INTEGER FUNCTION IOR(I,MSK)                                       IOR00010
      INTEGER OR                                                        IOR00020
      EXTERNAL OR                                                       IOR00030
      IOR=OR(I,MSK)                                                     IOR00040
      RETURN                                                            IOR00050
      END                                                               IOR00060
```