

JAERI-Data/Code

97-055



原子カコードのVPP500におけるベクトル化、
並列化及び移植(移植編)
—平成8年度作業報告書—

1998年1月

根本俊行*・川崎信夫*・田辺豪信*・渡辺秀雄*・川井 渉*
鈴木信太郎*・原田裕夫・庄司 誠・久米悦雄・藤井 実

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

本レポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合わせは、日本原子力研究所研究情報部研究情報課（〒319-1195 茨城県那珂郡東海村）あて、お申し越してください。なお、このほかに財団法人原子力弘済会資料センター（〒319-1195 茨城県那珂郡東海村日本原子力研究所内）で複写による実費領布をおこなっております。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195, Japan.

© Japan Atomic Energy Research Institute, 1998

編集兼発行 日本原子力研究所
印刷 (株)原子力資料サービス

原子力コードのVPP500におけるベクトル化，並列化及び移植（移植編）

—平成8年度作業報告書—

日本原子力研究所計算科学技術推進センター

根本 俊行*・川崎 信夫*・田辺 豪信*

渡辺 秀雄*・川井 渉*・鈴木信太郎*

原田 裕夫・庄司 誠・久米 悦雄

藤井 実

(1997年12月2日受理)

本報告書は、平成8年度に計算科学技術推進センター情報システム管理課で行った原子力コードのVPP500における高速化及び移植作業のうち、移植作業部分について記述したものである。

原子力コードのVPP500における高速化及び移植作業は、平成8年度に11件行われた。これらの作業内容は、同種の作業を行うユーザに有益な情報を提供することを意図して、「並列化編」、「ベクトル化編」、「移植編」の3冊にまとめられている。

本報告書の「移植編」では、軽水炉安全解析コードRELAP5/MOD3.2及びRELAP5/MOD3.2.1.2、原子核データ処理システムNJOY、2次元多群ディスクリート・オーディネーツ輸送コードTWOTRAN-IIに対して行った移植作業と汎用図形処理解析システムIPLOTに対して行ったUNIX OSへの移行調査作業について記述されている。別冊の「並列化編」では、2次元相対論的電磁粒子コードEM2D、円筒座標系直接数値シミュレーションコードCYLDNS、ダイヤモンド型結晶を扱う分子動力学コードDGRに対して行ったベクトル並列化作業について記述されている。また、別冊の「ベクトル化編」では、中性子・光子輸送計算コードDORT-TORT、気体流動解析コードFLOWGR、相対論的ボルツマン・ウェーリング・ウーレンバック法によるシミュレーションコードRBUUに対して行ったベクトル化作業について記述されている。

日本原子力研究所（東海駐在）：〒319-1195 茨城県那珂郡東海村白方白根2-4

* 外来研究員：富士通株式会社

* 富士通株式会社

Vectorization, Parallelization and Porting of Nuclear Codes
on the VPP500 System (Porting)
— Progress Report Fiscal 1996 —

Toshiyuki NEMOTO*, Nobuo KAWASAKI*, Hidenobu TANABE*,
Hideo WATANABE*, Wataru KAWAI*, Shintaro SUZUKI*,
Hiroo HARADA, Makoto SHOJI, Etsuo KUME
and Minoru FUJII

Center for Promotion of Computational Science and Engineering
(Tokai Site)

Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received December 2, 1997)

Several computer codes in the nuclear field have been vectorized, parallelized and transported on the FUJITSU VPP500 system at Center for Promotion of Computational Science and Engineering in Japan Atomic Energy Research Institute. These results are reported in 3 parts, i.e., the vectorization part, the parallelization part and the porting part. In this report, we describe the porting.

In this porting part, the porting of reactor safety analysis code RELAP5/MOD3.2 and RELAP5/MOD3.2.1.2, nuclear data processing system NJOY and 2-D multigroup discrete ordinate transport code TWOTRAN-II are described. And also, a survey for the porting of command-driven interactive data analysis plotting program IPLOT are described. In the parallelization part, the parallelization of 2-Dimensional relativistic electromagnetic particle code EM2D, Cylindrical Direct Numerical Simulation code CYLDNS and molecular dynamics code for simulating radiation damages in diamond crystals DGR are described. And then, in the vectorization part, the vectorization of two and three dimensional discrete ordinates simulation code DORT-TORT, gas dynamics analysis code FLOWGR and relativistic Boltzmann-Uehling-Uhlenbeck simulation code RBUU are described.

* On leave from FUJITSU, Ltd

* FUJITSU, Ltd

Keywords: RELAP5/MOD3.2, RELAP5/MOD3.2.1.2, NJOY, TWOTRAN-II , IPLOT,
Porting, VPP500, Nuclear Codes

目 次

1. はじめに	1
2. RELAP5/MOD3.2 コードのインストール	2
2.1 コード概要	2
2.2 変換前作業	2
2.3 変換作業 -1	4
2.4 変換作業 -2	5
2.5 M780 版と VPP500 版との違い	9
2.6 スチームテーブルの作成	11
2.7 実行ロードモジュールの作成	11
2.8 実行方法について	13
2.9 RELAP5/MOD3.2 コードの出力結果の検証	15
2.10 まとめ	15
3. RELAP5/MOD3.2.1.2 コードのインストール	45
3.1 コード概要	45
3.2 変換方針	45
3.3 変換のための準備	46
3.4 変換作業 -1	46
3.5 変換作業 -2	47
3.6 M780 版と VPP500 版との違い	48
3.7 実行ロードモジュールの作成	49
3.8 実行方法について	50
3.9 RELAP5/MOD3.2.1.2 コードの出力結果の検証	50
3.10 まとめ	50
4. NJOY94 のインストール及び倍精度化	61
4.1 NJOY94.25 の生成	61
4.2 VPP へのインストール	61
4.3 VPP 上での倍精度化	62
4.4 まとめ	70
5. TWOTRAN-II コードのインストール	82
5.1 M780 上での動作確認	82
5.2 VPP500 への移植	83
5.3 VPP500 上での検証	84
5.4 まとめ	87
6. IPLOT の移植調査	102
6.1 はじめに	102

6.2	IPLOT について	102
6.3	移植における問題点	103
6.4	解決策の提案	104
6.5	まとめ	106
7.	おわりに	112
	謝 辞	112
	参考文献	113
	付録 A. RELAP5/MOD3.2 のツリー構図	115
	付録 B. RELAP5/MOD3.2.1.2 のツリー構図	131
	付録 C. NJOY94.25 の修正内容	147

Contents

1. Introduction	1
2. Installation of RELAP5/MOD3.2	2
2.1 Overview of RELAP5/MOD3.2	2
2.2 Pre-installation	2
2.3 Porting-1	4
2.4 Porting-2	5
2.5 The Difference between M780 Version and VPP500 Version	9
2.6 Making of Steam-tables	11
2.7 Making of Load-modules	11
2.8 Method of Execution	13
2.9 Evaluation of Calculated Results	15
2.10 Summary	15
3. Installation of RELAP5/MOD3.2.1.2	45
3.1 Overview of RELAP5/MOD3.2.1.2	45
3.2 Plan of Installation	45
3.3 Arrangements of Installation	46
3.4 Porting-1	46
3.5 Porting-2	47
3.6 The Difference between M780 Version and VPP500 Version	48
3.7 Making of Load-modules	49
3.8 Method of Execution	50
3.9 Evaluation of Calculated Results	50
3.10 Summary	50
4. Installation of NJOY94 and Its Improvement to Double Precision Version ..	61
4.1 Installation in SUN Work Station	61
4.2 Installation in VPP500	61
4.3 Improvement to Double Precision Version	62
4.4 Summary	70
5. Installation of TWOTRAN-II	82
5.1 Implementation of TWOTRAN-II on M780	82
5.2 Porting of TWOTRAN-II to VPP500	83
5.3 Verification of Computation Results on VPP500	84
5.4 Summary	87
6. A Survey for Porting of IPLOT	102
6.1 Introduction	102

6.2 Overview of IPLOT	102
6.3 Problems in Porting of IPLOT	103
6.4 Proposition of Solution	104
6.5 Summary	106
7. Concluding Remarks	112
Acknowledgements	112
References	113
Appendix A. The Tree Structure of RELAP5/MOD3.2	115
Appendix B. The Tree Structure of RELAP5/MOD3.2.1.2	131
Appendix C. Change Contents of NJOY94.25	147

1. はじめに

計算科学技術推進センター情報システム管理課では、原子力コードをセンターが保有する各種のスーパーコンピュータ上に整備し、それぞれのスーパーコンピュータに最適な高速化を施す作業を実施している。この作業は、コンピュータ資源の効率的利用を図るとともにユーザの計算待ち時間の短縮を通じてユーザの仕事の効率化を図っている。

原子力コードの富士通(株)製分散メモリ型ベクトル並列計算機 VPP500/42(以下 VPP500)における高速化及び移植作業は、平成8年度に11件行なわれた。これらの作業内容は、同種の作業を行うユーザに有益な情報を提供することを意図して、「並列化編」、「ベクトル化編」、「移植編」の3冊にまとめられている。

本報告書の「移植編」では、軽水炉安全解析コード RELAP5/MOD3.2 及び RELAP5/MOD3.2.1.2、原子核データ処理システム NJOY、2次元多群ディスクリット・オーディネーツ輸送コード TWOTRAN-II についての VPP500 への移植作業及び汎用図形処理解析システム IPLOT の UNIX OS への移行調査作業について記述されている。

別冊の「並列化編」では、2次元相対論的電磁粒子コード EM2D、円筒座標系直接数値シミュレーションコード CYLDNS、ダイヤモンド型結晶を扱う分子動力学コード DGR に対して行ったベクトル並列化作業について記述されている。

また、別冊の「ベクトル化編」では、中性子・光子輸送計算コード DORT-TORT、気体流動解析コード FLOWGR、相対論的ボルツマン・ウェーリング・ウーレンベック法によるシミュレーションコード RBUU に対して行ったベクトル化作業について記述されている。

第2章「RELAP5/MOD3.2 コードのインストール」、第3章「RELAP5/MOD3.2.1.2 コードのインストール」では、RELAP5 コードに対して行った VPP500 向けの移植作業について述べる。尚、本コードについては、富士通(株)製汎用大型計算機 M780/20(以下 M780)向けにも移植作業を行った。こちらの作業についても記述している。

第4章「NJOY94 のインストール及び倍精度化」、第5章「TWOTRAN-II コードのインストール」では、それぞれのコードに対して行った VPP500 向けの移植作業について述べる。

第6章「IPLOT の移行調査」では、一部の処理に M780 計算機用のアセンブラ言語が用いられているコードを UNIX OS 向けに移植する際に、C 言語を用いて該当処理を実現する方法について述べる。

尚、本報告書の第2章及び第3章の作業は根本が、第4章及び第5章の作業は川崎が、第6章を田辺がそれぞれ担当した。

2. RELAP5/MOD3.2 コードのインストール

軽水炉安全解析コード RELAP5/MOD3.2 [1] の M780, および VPP500 へのインストールを行った。このコードは、アイダホ国立研究所 (INEL) が開発しており、不定期に配布される。主に解析機能の追加、補正等がある場合に配布されるが、'MOD3.2' は配布時のバージョンを表している。原研では過去に RELAP5 コードとして、RELAP5/MOD2, RELAP5/MOD3 等が導入されている。RELAP5/MOD3.2 コードは、オリジナルのコードから使用する計算機に適合するコーディング部分を抜き出して使用する。しかし、M780, および VPP500 に使用されている FORTRAN77EX, および VPP FORTRAN77EX/VP コンパイラに適合するものではなく、これらに一番近い IBM 用コーディングを抽出して変換した。ただし、これは、単に抽出するだけで IBM 計算機で実行できる仕様になっておらず、大部分が CRAY 計算機のコーディング仕様である。この抽出したコードに、同時に配布される変換ツールで IBM 用に変換することで、IBM 版の RELAP5/MOD3.2 コードができあがる。ここでは、変換の手順、およびその内容について説明する。

2.1 コード概要

加圧水型原子炉 (PWR) の事故や異常過渡変化を解析するための熱水力最適評価コードであり、PWR の炉心、蒸気発生器、加圧器などの主要機器をモデル化するためのツールや安全保護系をモデル化するための制御系モデルを備えている。これらを用いて、PWR の総合実験解析、個別効果実験解析、および商用炉の解析を行う [1]。

2.2 変換前作業

RELAP5/MOD3.2 コードの変換作業を行うには、主に M780 上で行うことにする。これは、次の理由からなる。

- 1) ユーザは VPP500, および M780 上で RELAP5/MOD3.2 コードを実行する。
- 2) 過去に配布された RELAP5 コードの変換に使用されてきた変換ツールが M780 上にある。
- 3) 一般的な FORTRAN コードの解析ツールは M780 の方が充実している。

2.2.1 フロッピーからのロード

RELAP5/MOD3.2 コードは 6 枚の MS-DOS 形式の 3.5 インチフロッピーに、ソース、サンプル入力データ、サンプル出力データ等を格納して配布された。このうち、1 枚に RELAP5 ソース、サンプル問題等が格納され、他の 5 枚のフロッピーには IBM, Sun, DEC(RISC), DEC(ALPHA), HP の各計算機によるサンプル問題の出力結果が格納されていた。また、

ソース、出力結果等は ZIP 形式で圧縮されていた。ソースが格納されていたフロッピー、および IBM 計算機のサンプル問題出力結果が格納されていたフロッピーの内容を Table 2.1, および 2.2 に示す。

ファイル抽出方法は、パソコン上でフロッピーよりファイルをダウンロードし、これらを VPP500 のフロントエンドプロセッサである GSP にファイル転送した。GSP 上に ZIP の圧縮・解凍ツールを用意して、そこで ZIP 形式ファイルの解凍を行った。そして、ここでできたファイルをすべて M780 に転送した。

2.2.2 大文字化

RELAP5/MOD3.2 コードは元々小文字によってコーディングされている。しかし、変換ツール、および解析ツールが小文字対応になっていないため大文字化した。また、入力データも大文字化した。

2.2.3 ソース抽出

ここでは、RELAP5/MOD3.2 コードは RELAP5 本体部、および Environmental Library 部より構成される。これらより使用する計算機に適応するコーディング部分、および RELAP5/MOD3.2 が持つ各種解析機能を抽出する作業について述べる。INEL から送付されたオリジナルのソースには *COMDEC, *DECK, および '*IF DEF, ~' によって定義される各種パラメータが挿入されている。*COMDEC で宣言されているものはインクルードファイル、*DECK で宣言されているものはメイン、および副プログラムのそれぞれ 1 メンバとして抽出され、'*IF DEF, ~' は '~' にパラメータが指定してあるが、抽出ツール 'SELECTX'[2] の実行時に指定されたパラメータと合致するものが抽出される（これ以下、このパラメータを選択パラメータと呼ぶ）。このイメージを Fig. 2.1 に示す。

(1) RELAP5 本体部

IBM 版のサンプル問題出力結果には、選択パラメータの一覧が出力されており、これを参考に M780, および VPP500 用の選択パラメータを決定した。これらを Table 2.3 に示す。

これらの選択パラメータを指定した抽出ツール 'SELECTX' の実行 JCL を Fig. 2.2 に示す。オリジナルソースにはインクルードファイルも含まれている。'SELECTX' はオリジナルソースから、指定した選択パラメータに応じた抽出作業を行うが、抽出したソース、インクルードファイルをそれぞれ PO ファイルに出力する機能を持っている。

(2) Environmental Library 部

Environmental Library ソースは主に、RELAP5/MOD3.2 コード本来の解析の一部を行うものではなく、ファイル、データ等のチェック、移動等のファイル操作関係を行うライブラリと RELAP5/MOD3.2 コードの入力データのひとつであるスチームテーブルを作成するプログラム、およびこのプログラムの入力データ、その他ツール等が含まれている。Environmental Library ソースの抽出に使用した選択パラメータを Table 2.4 に、'SELECTX' の実行 JCL を Fig. 2.3 に示す。

2.2.4 Environmental Library ソースの分離-1

抽出ツール 'SELECTX' を実行した Environmental Library ソースから、スチームテーブル入力データ、および FORTRAN 記述でないものを分離する。残ったファイルは FORTRAN プログラムから構成されるが、このファイルを次章で述べるツールによる倍精度化、および IMPLICIT 文の挿入を行ってから、RELAP5/MOD3.2 コードの計算で使用される Environmental Library 部とスチームテーブル作成ルーチン・ツール等を分離する。これは、入力データは倍精度化する必要がないこと、また倍精度化の改良をしてしまうと、書式が変わってしまうための理由による。

FORTRAN 記述でないものは主に C 言語によるツール等であるが、これは M780, VPP500 での実行には必要ないものなので、インストールの対象外にした。

2.3 変換作業-1

ここでは、ツールを使用して変換作業の一部を行なう。主な作業内容は倍精度化である。また、この作業終了後、スチームテーブル作成等のためソースの一部を分離する。

2.3.1 実数の倍精度化、および実数配列と共有する整数配列の次元上げ

'SELECTX' にてオリジナルソースより抽出したソースは、一部を除いてほぼ CRAY 仕様のコーディングになっている。この抽出済ソースに対し、実数配列と共有する整数配列の次元上げを行うことにより、IBM 版のソースができることになっている。実数配列と共有する整数配列を次元上げる理由は、CRAY は実数型、および整数型の変数はともに単精度が 64 ビットであるのに対し、IBM、富士通の FORTRAN77EX、および VPP FORTRAN77EX/VP は単精度が 32 ビットであるからである。実数の精度を同じにするためには、実数を倍精度化 (64 ビット) しなくてはならない。RELAP5/MOD3.2 コードでは EQUIVALENCE 文や副プログラムの引数の引き渡しで、多くの実数と整数の共有がなされている。実数を倍精度化すると整数配列にずれが生じる。そこで、整数配列の (先頭に) 次元を一つ加え、大きさを 2 で宣言すると、実数配列と同じ大きさになる (Fig. 2.4 参照)。実際に使用する整数の値は、添字が 2 にエリアに格納する。

この作業を行うツールとして、コードと一緒に配布された 'CNV32.F'[2] がある。このツールは以前の RELAP5 コードの配布時に添付されてきたものと同じものである。このコードを PO ファイルの全メンバに対して行う改良版 'CONV32'[2] があり、これを使用して変換することにした。このツールは整数配列の次元上げを対象とする配列名を入力することで、ここに指定された配列は強制的に次元上げがされる。入力データはひとつ前の RELAP5 コードのバージョンである RELAP5/MOD3 コードの変換時に用いたデータに、今回調査を行って次元上げをすべき整数配列名を加えた。この入力データを RELAP5 本体部については Fig. 2.5, Environmental Library については Fig. 2.6 に示す。なお、Environmental Library は全ルーチンで共通の次元上げ対象の整数配列が無い (ローカルのものも存在する)、ダミーの入力データである。

このツールを実行した場合の変更の様子を Fig. 2.7, ツールを実行するコマンドプロシジャを Fig. 2.8 に示す.

2.3.2 IMPLICIT文の挿入

オリジナルより抽出したIBM版に対し、倍精度・整数次元上げツール‘CONV32’を実行しても完全な倍精度版にはならない。これは、一部の副プログラムには‘IMPLICIT REAL*8 (A-H,O-Z)’が存在しないからである。本来ならIBM版をオリジナルより抽出すれば、各副プログラムにはIMPLICIT文が存在するはずであるが、作成者側のミスにより、IMPLICIT文が挿入されてなかったものと思われる。このため、‘CONV32’を実行した後も、単精度の変数・配列が残る場合がある。IMPLICIT文の無い副プログラムには、富士通(株)製プログラム変換支援パッケージ‘STREAM77’[3]を使用して、IMPLICIT文を挿入した。‘STREAM77’は、すでにIMPLICIT文が存在する副プログラムには挿入しない機能を持っているので、機械的にこのツールを実行するだけでよい。挿入例を Fig. 2.9 に、‘STREAM77’の実行JCLを Fig. 2.10 に示す。

2.3.3 Environmental Library ソースの分離-2

ツールによる変換後、Environmental Library ソースより、RELAP5/MOD3.2 コードを実行する時に使用されるライブラリとスチームテーブル作成ルーチン部・ツール等を分割する。分割を行うための指標は、副プログラムの中にPROGRAM文があるか無いかである。PROGRAM文があるものはスチームテーブル作成ルーチンやツール等のメインルーチンであり、無いものはEnvironmental Library である。

スチームテーブル作成メインルーチンを調査したところ、ASTEMというメンバのみがスチームテーブル作成ルーチンに必要で、ほかのEnvironmental Library はスチームテーブル作成ルーチン、およびRELAP5本体の両方で使用される。なお、ASTEMはいくつかの副プログラムとブロックデータから成る。

よって、Environmental Library ソースより、PROGRAM文の含んでいるスチームテーブル作成プログラム部・ツール等、およびASTEMを分離した。

今回のRELAP5/MOD3.2 コードはスチームテーブル入力データを14種類使用する。このデータを作成するスチームテーブル作成プログラムも14種類あった。これらのプログラム名、およびデータ名を Table 2.5 に示す。

2.4 変換作業-2

2.2および2.3でソースの抽出、およびツールによる変換を行った。しかし、ツールでは対応できない変換やオリジナルコードに含まれているエラーは、手作業によって変換・修正を行った。

2.4.1 不要なインクルードファイル呼び出しの削除

RELAP5/MOD3.2 コードは RELAP5 本体部, Environmental Library 部の各副プログラムにおいて, 多数のインクルードファイルの呼び出しがあるが, RELAP5 本体部においては, 選択する機能によっては, または使用する計算機によっては不要なものがある. また, 旧バージョンで使用していたインクルードファイルの修正忘れ等もあった. さらにインクルードファイル自体が無いために, コンパイルエラーとなってしまいうものもあった. これらは, 調査したところ, (インクルードファイルが無くても) 実行には影響がないことが判明した. ここで述べた不要なインクルードファイルの呼び出し部分は, 削除(コメント化)した. 副プログラム毎の削除したインクルードファイル呼び出しを Table 2.6 に示す.

2.4.2 インクルードファイル名の変更

RELAP5 本体部と Environmental Library 部のインクルードファイルで同名のものがある. M780 上ではロードモジュール作成時に RELAP5 本体部と Environmental Library 部は別々のステップでコンパイルするので問題ないが, VPP500 上でのロードモジュール作成では, FORTRAN ソースのコンパイルは(使いやすさを重視して) 1ステップで行う仕様にしたため, 同じ名前ではいけない. また M780 上でも混乱を防ぐために Environmental Library 部のインクルードファイル名を変更した. 変更したものは2つであり, これらを Table 2.7 に示す.

2.4.3 不要な副プログラムの削除

この時点でコンパイルを行い, コンパイルエラーが発生する副プログラムで, どこからも呼び出しが無いものを削除した. これらは主に選択した機能外のもの, 計算機に依存するもの, 図形処理用のもの等である(本来ならば, コード作成者側が 'IF DEF' 文により, 選択されないようにすべきものが大部分である). 削除した副プログラムを Table 2.8 に示す.

2.4.4 副プログラム名の長さの変更

副プログラム名に7文字のものがあつた. FORTRAN77EX も VPP FORTRAN77EX/VPP もこの長さは許容範囲であるので, 本来は修正する必要はない. しかし, これまでの RELAP5 コードの経験より, 原研外の計算機に移植することがあり, そのためには, より汎用的なコーディングをしておいたほうが良いこと, M780 上に存在する FORTRAN 静的解析ツールには, FORTRAN77EX の前の旧コンパイラ FORTRAN77 仕様(プログラム名は6文字まで)のものが多くことより, 副プログラム名の長さを6文字にした. 変更した副プログラム名を Table 2.9 に示す.

2.4.5 次元上げによる引数のマッチング

2.3で述べたように, 倍精度化ツール 'CONV32' によって, 実数配列・変数と共有する整数の次元上げを行った. ここで, 第1添字が2の部分に整数の値が入ることは述べた. よってプロ

グラム中で整数として使用する箇所は、第1添字が2の部分を使用する。ところで、サブルーチンの呼び出しの際、実引数にこの整数配列を(添字付きで)指定し、受け側で実数配列、または整数の変数で受け取る場合がある。この場合、受け側が実数配列の場合は、整数配列の第1添字には‘1’を指定し、受け側が整数の変数の場合には、第1添字に‘2’を指定する必要がある。‘CONV32’は、受け側の配列、または変数までを見極める機能は無い。RELAP5/MOD3.2コードの場合、実引数に配列を指定している場合は、受け側である仮引数も配列である場合が多いので、‘CONV32’は、実引数に次元上げの対象となった整数配列がある場合には、第1添字を‘1’にする仕様になっている。そのため、受け側の仮引数が整数の変数の場合には、第1添字を‘2’に修正する処置をした。修正箇所の例を Fig. 2.11 に示す。

2.4.6 非互換項目の対処

RELAP5/MOD3.2コードのオリジナルはCRAY仕様のコーディングがなされている。それが、IBM版として抽出した中にも含まれている箇所がいくつかあった。これを、M780、およびVPP500上で実行できるように、これらの仕様に変更した。

(1) 動的領域割当ての回避

FORTRAN77EX、およびVPP FORTRAN77EX/VPでは、動的に領域の確保はできない。RELAP5/MOD3.2コードの開発したCRAY計算機は、これが可能である。今回の交換でIBM版として抽出した箇所には、RELAP5本体部で動的領域確保の箇所があった。このコーディング例を Fig. 2.12 に示す。これによると、

- ローカル配列の宣言の大きさにCOMMON変数(NUとNP)を使用している
- POINTER文とLOC関数(引数に指定された変数・配列の絶対アドレスを返す組込み関数)によってローカル配列と仮引数で宣言された配列TABLESが共有されるようになっている。例えば、TABLES(1)のアドレスがIPT1に入る。IPT1はポインタ変数と呼ばれる。POINTER宣言文によって、IPT1のアドレスにUTBLの先頭アドレスが一致させられている
- 同様に仮引数Sに対して、ローカル配列IS(整数型)、およびLS(論理型)と共有している

ということを行っている。このイメージ図を Fig. 2.13 に示す。

対処として、S、IS、LSの場合は、呼び出し側の実引数にSに相当する配列を追加し、受け側の仮引数にIS、LSを追加する。これは、通常の整合配列となり、IS、LSは何ら変更なく使用できる。TABLES、UTBL、PTBL等の場合は、これらUTBL等のローカル配列を文関数として宣言し、実際に使用される添字から(TABLESの)インデックスの計算をして、TABLESの値を返すようにした。変更を加えた例としてサブルーチンSTPUPUを Fig. 2.14 に示す。この変更を行った副プログラムを Table 2.10 に、呼び出し側の実引数に変更を加えたものを Table 2.11 に示す。ともに、‘C32’で始まるコメント行が変更を加える前のもの、その次の行にあるものが変更を加えたものである。また Fig. 2.14 の変更を行ったものは、Table 2.10 に示した副プログラムすべてである。

(2) 値域の変更

実数の表せる範囲(値域)はVPP500の採用しているIEEE形式、M780が採用しているM形式、CRAYが採用しているCRAYワードとですべて異なる(Fig. 2.15 参照)。一番範囲がせまいのはM形式である。そこで、プログラム中で最小値を設定する場所では、M形式で表せる最小値に近い値に設定することにした。修正した副プログラムはサブルーチン STATEP(RELAP5 本体部)である。オリジナルの STATEP を Fig. 2.16, 修正した STATEP を Fig. 2.17 に示す。

(3) 行番号なしの DO ループの修正

サブルーチン BPPRAM(RELAP5 本体部)では行番号なしの DO ループが2箇所あった。行番号なしの DO ループとは次の形の DO ループである。

```
DO I = 1, N
.
.
.
END DO
```

これは、(VPP FORTRAN77EX/VP の文法書には記述されていないが)VPP500 上では正常にコンパイルすることができるが、M780 上ではコンパイルエラーとなる。そのため、行番号付きの DO ループに修正を行った。修正した箇所を Fig. 2.18 に示す。

(4) サービスルーチンの修正

CRAY 用のサービスルーチンを使用しているところがあり、これを M780, および VPP500 で用意されているサービスルーチンに置き換えた。

サブルーチン FABEND(Environmental Library 部)は、プログラムの実行中、何らかのエラーが発生した場合に、呼び出される。このルーチンでは、エラートレースを出力するサービスルーチン ABORT を呼び出しているが、これをサービスルーチン ERRTRA の呼び出しに変更した。この箇所を Fig. 2.19 に示す。

サブルーチン INP(Environmental Library 部)では、日付、時刻を得るサービスルーチン ED ATE, および ECLOCK を呼び出している。これを、旧バージョンの変換時に作成したサブルーチン GTINF に置き換えた。GTINF はサービスルーチン DATE, および CLOCK より成り立っている。修正箇所を Fig. 2.20, サブルーチン GTINF を Fig. 2.21 に示す。

(5) 文字型変数の長さの修正

RELAP5/MOD3.2 コードでは、入力データ(書式付き)を読み込む場合、一行ずつ文字型の変数で読み、その後文字、数値を分離していく。この最初の読み込みに使われる文字型変数が 96 バイトで宣言されている。RELAP5/MOD3.2 コードの入力データは、80 バイト分あれば十分であり、また計算機のファイルの構成の上でも便利である。修正したのはサブルーチン INP である。修正した箇所を Fig. 2.22 に示す。

2.4.7 エラー修正

RELAP5/MOD3.2 コードに限らず、これまで配布されてきた RELAP5 コードにおいて、オリジナルソースにはいくつかのエラーが含まれていた。ここでは、その修正について述べる。

(1) リスタート

RELAP5/MOD3.2 コードは、解析問題の規模によって CPU 時間が極めて長時間に至る場合があるため、リスタート機能を備えている。これは、最初のジョブで COMMON に含んでいる配列等の値をバイナリ形式でファイルに出力し、次のジョブでこのファイルを読み込んでリスタート実行するものである。サブルーチン RSTREC(RELAP5 本体部) ではリスタートファイルの書き出し、サブルーチン RRESTF(RELAP5 本体部) ではリスタートファイルの読み込みをしている。これらのオリジナルの書き出し、読み込み部分を Fig. 2.23, および Fig. 2.24 に示す。

例えば Fig. 2.23 の書き出しの場合、配列 HA の 1+IXB-1 ~ LX+IXB-1 の範囲を書き出している。しかし、調査の結果、これでは必要量の半分しか書き出していないことが判明した。プログラムでは、HA は配列 FA と EQUIVALENCE 文によって領域共有がなされている。HA は 4 バイト整数型で宣言され、FA は 8 バイト実数型で宣言されている。旧バージョンでは FA が直接 WRITE 文で指定されており、そこで書き出す範囲は今回の範囲と同じであった。そこで、WRITE 文での HA の使用をやめ、FA を使用することにした。読み込み部分も同様の処置をした。RSTREC, および RRESTF の修正箇所をそれぞれ Fig. 2.25, および Fig. 2.26 に示す。

(2) コーディングのずれ

ファンクション CHECK において、余分な記号(ピリオド)とコーディングのずれがあった。これを正しく修正した。これを Fig. 2.27 に示す。

2.4.8 装置参照番号の変更

いくつかの入出力のための装置参照番号の変更を行った。これは、M780, および VPP500 の標準入出力に合わせることで、これまでの RELAP5 コードの使用番号と同一にすること等の理由のためである。この変更について Table 2.12 に示す。

2.5 M780 版と VPP500 版との違い

M780 版と VPP500 版とのソースでは、計算機の仕様が異なるため、一部の副プログラムで異なる箇所がある。ここでは、仕様の違いによるコーディングの変更点を述べる。

2.5.1 COMMON の順序付けのための変更

RELAP5/MOD3.2 コードは COMMON ブロックをメモリ上で順序付けて配置する必要がある。それは、複数の COMMON を、あたかもひとつの COMMON のように使用している箇所があるためである。このため Table 2.13 に示すような順序で COMMON ブロックを配置する。

M780ではリンク時に、順序付けするCOMMONを指定することができる(次章参照)。そのため、プログラム中で特に意識する必要はない。

VPP500はリンク時にCOMMONの順序付き配置を制御することはできない。このため、プログラム中で制御を行う。VPP500は、COMMONブロックに含まれる配列、または変数の使用順によって、メモリ上にこのCOMMONブロックが割り当てられる性質を持つ。また、ブロックデータがプログラム中にある場合は、ブロックデータの定義順にCOMMONブロックが割り当てられる。

RELAP5/MOD3.2コードでは、ブロックデータBLKDTA(RELAP5本体部)を含んでいる。このため、このままVPP500で実行すると、BLKDTA内の配列・変数の定義順にCOMMONブロックが割り当てられる。そこで、新たにブロックデータAAAAAAを作成し、Table 2.13に示したCOMMONブロックに含まれる配列・変数を初期化した。これらのCOMMONブロックの中で、BLKDTAで初期化されているものは、ブロックデータAAAAAAに移動した。ブロックデータで初期化されていないものは、ダミーで配列・変数に'0'を定義した。オリジナルのブロックデータBLKDTAをFig. 2.28、新規に作成したブロックデータAAAAAAをFig. 2.29、変更を加えたブロックデータBLKDTAをFig. 2.30に示す。

2.5.2 アセンブラとC

RELAP5/MOD3.2コードでは、アドレスの取得とCPUの残り時間の取得にM780ではアセンブラを使用する。VPP500ではCPUの残り時間の取得機能がないため、アドレスの取得のみにC言語を使用する。M780で使用するアドレスを取得するアセンブラルーチンLOCBとCPUの残り時間を取得するアセンブラルーチンRCPUをFig. 2.31に示す。またLOCBの呼び出しの一例であるファンクションLOCF(RELAP5本体部)をFig. 2.32、RCPUの呼び出すREMTIM(Environmental Library部)の一部をFig. 2.33に示す。

VPP500版のアドレス取得CルーチンlocbcをFig. 2.34に示す。また、locbcの呼び出しの一例であるサブルーチンLOCFをFig. 2.35に示す。

2.5.3 スチームテーブル読み込み

これまでのRELAP5コードにおいて、オリジナルでは、OPEN文によってスチームテーブルファイルと論理機番の連結を行っていた。これまで原研にインストールしたバージョンでは、OPEN文をコメント化し、JCLにて連結を行っていた。これは、OSがMSPであったこと、スチームテーブルが1種類しかなかったことからである。しかし、RELAP5/MOD3.2コードからは、スチームテーブルが14種類になり、読み込むスチームテーブルは入力データによって内部で自動的に決定され、OPEN文のFILE指定子に自動的にこの名前が入る仕様になっている。このため、OPEN文による連結を採用し、スチームテーブルファイルはM780、VPP500ともにID:J0001のセッションに置くことにし、プログラム中から直接ここに読みに行くようにした。OPEN文でスチームテーブルファイルを指定する場合、原研の仕様として、MSPの場合ファイル名はIDを含むフルネームを指定する必要がある。また、VPP500の場合、カレントディレクトリ上にスチームテーブルファイルがあれば、これらのファイル名のみ指定すればよい

が、ID:J0001に置いておくので、絶対パスを用いて指定する。M780版のOPEN文とVPP500のOPEN文(ともにサブルーチンICMPN1(RELAP5本体部))をそれぞれFig. 2.36, Fig. 2.37に示す。

2.6 スチームテーブルの作成

スチームテーブルを作成するために、スチームテーブル作成プログラムを実行する。スチームテーブルは前述した通り、14種類あり、それぞれについての作成プログラムを実行すればよい。

2.6.1 M780 上での作成

M780上でスチームテーブルを作成するには、コンパイル時にそれぞれのスチームテーブル作成メインルーチン、および2.3で述べたASTEMをコンパイルし、リンク時にEnvironmental Libraryを結合し、実行時にそれぞれの入力データを読み込めばよい。14種類のスチームテーブルはすべてこの方法で作成した。これらのうちの一つであるスチームテーブルTPFBLOODの作成JCLをFig. 2.38に示す。

2.6.2 VPP500 上での作成

VPP500上でスチームテーブルを作成する場合も、MSP上での作成とほぼ同様であるが、Environmental Libraryをライブラリとしてリンク時に連結せず、スチームテーブル作成ルーチンから呼び出す副プログラムのみEnvironmental Libraryより抽出し、メインルーチン、およびASTEMといっしょにコンパイルして実行ロードモジュールを作成した。Environmental Libraryより抽出したものはサブルーチンECLOCK, EDATE, FABENDの3つである。スチームテーブルTPFBLOODの作成実行シェルスクリプトをFig. 2.39に示す。

2.7 実行ロードモジュールの作成

ここではM780, およびVPP500でのロードモジュールを作成するためのコンパイル手順・リンク等について述べる。

2.7.1 M780 上でのロードモジュール作成

M780上でロードモジュールを作成する場合、コードがAE対応(拡張メモリ使用)していないため、NOAEモードでロードモジュールを作成する。また前述したが、RELAP5/MOD3.2コードの特性としてCOMMONブロックの順序付けを行う必要がある。MSPではこの作業をリンク時に行う。

(1) リンケージ入力データの作成

RELAP5/MOD3.2 コードでの順序付けする COMMON ブロックは 2.5.1 で示した。MSP 上ではこれをリンク時に入力データとして、COMMON の順番を指定すると、この順番でロードモジュールができる。入力データでは ORDER 文によって COMMON ブロックを指定する。この入力データを Fig. 2.40 に示す。なお、ORDER 文の一番最初にしている 'NEMOTO' はダミーサブルーチンである。このルーチンは SUBROUTINE 文、RETURN 文、END 文だけからなるものである。これを指定した理由はリンケージ制御の仕様で、一番最初には副プログラムを指定することになっているからである。

(2) コンパイル・リンク

RELAP5/MOD3.2 のコンパイルステップは 3 段階ある。第 1 ステップがアセンブラ、第 2 ステップは RELAP5 本体部、第 3 ステップは Environmental Library である。第 2、第 3 ステップのコンパイルオプションには、'NOAE' を指定する。これは、拡張メモリを使用しないコードであるためである。ロードモジュールを作成する JCL を Fig. 2.41 に示す。

2.7.2 VPP500 上でのロードモジュールモジュール作成

VPP500 上でロードモジュールを作成する場合、M780 の手法とは異なる。それは

- M780 のような複数段階のコンパイルは行わない
- COMMON の順序付けは、ブロックデータ AAAAAA で行う
- アセンブラではなく C 言語のソースを使用する

である。COMMON の順序付けと C 言語については 4 章で述べた。複数段階コンパイルを行わないのは、使いやすさを重視したためである。

UNIX 計算機において、通常ロードモジュールを作成する場合は Makefile を使用する。しかし、RELAP5/MOD3.2 コードでは、副プログラムの数が多いため Makefile の大きさの制限を越えてしまう。また、TSS でのコンパイル・リンクは、リンク時にメモリが不足することより実行できない。そのため、バッチ方式によって、すべての副プログラムをコンパイルし、ロードモジュールを作成するシェルスクリプト cmp.sh、およびコンパイルは行わず、すべてのオブジェクトモジュールをリンクしてロードモジュールを作成するだけのシェルスクリプト cmp2.sh を作成した。これらを Fig. 2.42, 2.43 に示す。cmp2.sh はいくつかの副プログラムを修正した場合、コンパイルのみを TSS で行い、その後のリンクを行う場合のために作成した。シェルスクリプト内には、qsub 時に必要なパラメータ (実行クラス等) を指定する仕様にした。パラメータは '#0\$' 以降に指定する。なお、RELAP5/MOD3.2 コード内の作業配列の大きさによっては通常のコンパイルクラス 'gspc' ではメモリが不足する場合がある。その時には、許可クラスである 'gsps' で実行する。

2.7.3 作業領域について

RELAP5/MOD3.2 コードは解析対象によっては、作業領域として使用される配列を大きくしてロードモジュールを作成する必要がある。作業領域の大きさの指定は M780 版が 2 つのインクルードファイル、VPP500 版がこれらにブロックデータを加えた 3 箇所で行なっている。これらを Table 2.14 に示す。作業領域の変更場所はすべてパラメータ文で、変数 LFSIZ を値を変えればよい。変更する場所を Fig. 2.44 ～ 2.46 に示す。

2.8 実行方法について

ここでは M780, および VPP500 上での実行方法について述べる。

2.8.1 M780 上での実行

M780 上で実行する JCL を Fig. 2.47 に示す。ここで注意するのは、これまでの旧 RELAP5 コードの実行 JCL では、スチームテーブルファイルと装置指定子を連結していたが、前述通り、OPEN 文による連結になったので、JCL 内ではスチームテーブルの指定は無い。実際に読み込むスチームテーブルはセッション ID: J0001 に登録したものを読む。よって、原研外で実行するときには注意が必要である。

2.8.2 VPP500 上での実行

VPP500 上で RELAP5/MOD3.2 コードを実行する場合、最初のスタート実行、およびリスタート実行は通常 VPP500 で行う。しかし、ROPS[4] により図形出力をする場合は VPP500 ではできない。これは ROPS が図形ライブラリとして DISSPLA[5] を使用しているが、VPP500 上にはこれが無いためである。ROPS は RELAP5 コードの実行でできたリスタートファイルを入力としている。そのため、ROPS 使用時は、リスタートファイルを M780 に転送する必要がある。

(1) スタート・リスタートジョブ実行

スタートジョブ、およびリスタートジョブの実行を行うシェルスクリプト go.sh を Fig. 2.48 に示す。このシェルも M780 の実行 JCL と同様にスチームテーブルの連結は、プログラム内の OPEN 文によって行っているため、スチームテーブルの宣言をしていない。スチームテーブルは /dg04/ufs01/j0001 配下に置いてある。

(2) ROPS のためのリスタートファイル転送

VPP500 上に図形ライブラリ DISSPLA が無いために ROPS が使えないことは述べた。そのため、リスタートファイルを M780 へ転送する。ただしリスタートファイルはバイナリファイルなので、数値表現方式の異なる M780 へは、そのままでは転送できない。そのため、次の手順を踏む。

- 1) リスタートファイルに対し、IEEE形式からM形式に変換を行うコンバートプログラムを実行する。
- 2) 1) でできたM形式のリスタートファイルをM780よりJCLにてファイル転送ツールGETDAT (もしくはGETDAT2) を実行することで、VPP500より、M780へ転送する。
- 3) 転送されたリスタートファイルを使用してROPSを実行する。

1) で使用するコンバートプログラムは、リスタートファイルを読み込んで正しい書式で出力する。出力する際はVPP500の入出力データ変換機能を用いて書き出す。これは、RELAP5/MOD3.2コードでリスタートファイルの書き出しは文字、整数、実数をすべて倍精度実数型の配列によって書き出している。入出力データ変換機能はバイナリファイルの入出力でIEEE形式 → M形式、またはM形式 → IEEE形式に変換するものである。入出力するデータの型にあった変数・配列を用いないと、この入出力データ変換機能は正常に動作しない。そのため、コンバートプログラムを使用する。コンバートプログラムをFig. 2.49に示す。またコンバートプログラムの実行シェルスクリプトをFig. 2.50に示す。ここで論理機番1番はRELAP5の実行でできたリスタートファイルを、論理機番60番ではコンバートプログラムによってできるリスタートファイルを指定する。また、入出力データ変換機能を動作させるため、実行オプションで'-W1,-C60,-G60'を指定している。なお、コンバートプログラムは、RELAP5の実行でできたリスタートファイルに含まれるすべての文字、整数、実数を出力するものではない。これは、一部のデータでは、文字、整数、実数がランダムに現れるところがあるからで、これは正しい書式で書き出すことはできない。しかし、この部分のデータは、ROPSでは使用しないので問題はない。よって、コンバート済のリスタートファイルをリスタートジョブの入力データとして使用することはできない。

2) では、VPP500でできたコンバート済リスタートファイルをM780へ転送する。このファイルは書式無しバイナリファイルであるため、UNIX上とMSP上のファイルのデータ構造が異なるので、単純に通常のFTPによって転送することはできない。そのため、M-VPP連携機能を使用してファイル転送を行う。転送を行うため、いくつかのファイルを前もって準備する。M780側はJCLとJCLから呼び出すカタログドプロシジャ、GSP側は転送コマンドを登録する。これらのファイルをTable 2.15に示す。

ここでJCL、カタログドプロシジャ、コマンドにおいてGETDATとGETDAT2と2種類あるがこれは、機能が若干異なる。GETDATは、転送するコンバート済リスタートファイルがホームディレクトリ上にあるものに限られる。GETDAT2は、絶対パスを指定することでホームディレクトリ以外の場所になっても転送可能な機能を持つ。

JCLのGETDAT、およびGETDAT2をFig. 2.51、2.52に示す。これらの共通するパラメータは、FNAMEにはコンバート済リスタートファイル、DSにはM780側のデータセットを指定する。GETDAT2のPASSにはコンバート済リスタートファイルが存在するディレクトリまでの絶対パスを指定する。指定条件としては、絶対パスはすべて小文字、コンバート済リスタートファイルは大文字のファイル名でなければならない。なお、実行クラスは「Y」クラ

スのみである。JCLのGETDAT, およびGETDAT2からはカタログドプロシジャのGETDAT, GETDAT2を呼び出す。

カタログドプロシジャのGETDAT, GETDAT2をFig. 2.53, 2.54に示す。これらは、VPP500側のコマンドgetdat, もしくはgetdat2の起動を行う。また、コンバート済リスタートファイルの名前をVPP500側へのパラメータとして渡す。これらのカタログドプロシジャは各自セッションのPROCLIB.CNTLに複写して使用する。

VPP500コマンドのgetdat, getdat2をFig. 2.55, 2.56に示す。ここで、VPP500上のコンバート済リスタートファイルをM780側に複写(転送)する命令を定義する。これらのコマンドは各自の\$HOME/binに登録して使用する。

JCLのGETDAT, またはGETDAT2を実行することで、ファイル転送が行われる。転送後のリスタートファイルは通常のROPSの使い方で利用できる。

2.9 RELAP5/MOD3.2 コードの出力結果の検証

M780上, およびVPP500上で今回変換したコードが正常に動作するか確認するために、9種類のサンプル問題を実行した。またその結果をコードと共に送付されてきたサンプル出力結果と比較を行ってみた。比較には、計算最終ステップにおける計算体系の全質量(total mass)を使って行なった。この結果をTable 2.16に示す。なお、サンプル問題のedrstはリスタート機能のチェック用、同じくedstripはストリップファイル(リスタートファイルの編集ファイル)作成用で、本来の解析計算とは関係ないため、この表から除いた。ただし、両者とも正常に動作している。計算結果は計算機の違いと思われる誤差を含むが、ほぼ一致し、妥当な結果を得ることができた。よって、コードは正しく変換できたと思われる。

2.10 まとめ

RELAP5シリーズは不定期に原研に配布される安全性評価コードである。配布の度にバージョンアップ・機能追加等があり、巨大化してきている。そのため、変換作業にも労力があるが、これまでの変換技術の蓄積があり、コードの大きさに係わらず、比較的短期間で変換作業が終了できるようになっている。しかし、今回は機能が大きく追加されたため、多少戸惑ったところがあった。ただ、次回のバージョンアップ・機能追加が大幅にあるとは考えにくいので、次回の変換作業は短期間で済むと思われる。

最後に付録Aに、RELAP5/MOD3.2コードのツリー構図を示した。参考にされたい。

Table 2.3 Chosen parameters for M780 and VPP500 (RELAP5)

パラメータ	内 容
BLKDTA	ブロックデータの採用
BORONT	解析機能のひとつ
CROSSFL	解析機能のひとつ
DOPTION	解析機能のひとつ
FOURBYTE	INTEGER が 4 バイトのもの
SPHACCM	解析機能のひとつ
GESEP	解析機能のひとつ
IBM	IBM 計算機用
LEVEL	解析機能のひとつ
NEWWTRP	解析機能のひとつ
IN32	32ビット計算機用
TFRONT	解析機能のひとつ

Table 2.4 Chosen parameters for M780 and VPP500 (Environmental Library)

パラメータ	内 容
BLKDTA	ブロックデータの採用
IBM	IBM 計算機用
IN32	32ビット計算機用
FOURBYTE	INTEGER が 4 バイトのもの

Table 2.1 List of distributed files

フロッピー	ファイル名	内 容
NO. 1	relap.zip readme.dis pkunzip.exe	RELAP5ソース等 ソースの展開方法 ZIP 解凍ロードモジュール
NO. 2	ibm.zip pkunzip.exe readme	IBM 計算機によるサンブル問題出力結果 ZIP 解凍ロードモジュール サンブル問題出力結果の解凍方法

Table 2.2 Contents of ZIP-files

ZIP ファイル	ファイル名	内 容
relap.zip	cnv32.f dinstlis dutilty envr1.s goodies indecfs selaps select.f uspltt.f	IBM 用変換ツール ソース、入力データ抽出シェルスクリプト ソースのロードモジュール作成シェルスクリプト Environmental library ソース ソース、入力データ抽出シェルスクリプト サンブル問題(10 ケース) RELAP5本体ソース ソース抽出ツール ソース抽出
ibm.zip	ans79.p dinstlis.log edhtrk.p edhtrkd.p edhtrkn.p edr-st.p edstrip.st marpzd4.p pump2.p typpwr.p	サンブル問題 ANS79 出力結果 IBM 計算機上でのツール、コンパイル等のlog サンブル問題 EDHTRK 出力結果 サンブル問題 EDHTRKD 出力結果 サンブル問題 EDHTRKN 出力結果 サンブル問題 EDRST 出力結果 サンブル問題 EDSTRIP 出力結果 サンブル問題 MARPZD4 出力結果 サンブル問題 PUMP2 出力結果 サンブル問題 TYPPWR 出力結果

Table 2.5 Steam-table making programs and input data

No.	プログラム	入力データ
1	STGH20	TPFH20
2	STG020	TPFD20
3	STGH2	TPFH2
4	STGLI	TPFLI
5	STGK	TPFK
6	STGHE	TPFHE
7	STGN2	TPFN2
8	STGNA	TPFNA
9	STGNAK	TPFNAK
10	STGLIPB	TPFLIPB
11	STGNH3	TPFNH3
12	STGH20N	TPFH20N
13	STGGLYC	TPFGLYC
14	STGBLOOD	TPFBLOOD

Table 2.6 Include-files which deleted INCLUDE statements (1/2)

副プログラムの名	呼び出しを廃止したインクルードファイル
BLKOTA	FPLIM , FPINSC, FPARM1, FPARM2, FPINSH, FPCTRL, FPMDL
DYSTEP	CPMDAT, SCDDAT, COUPL , SCDOOT, SLUMPV, NDXARA, TBLSP , PVMVR5
FWRAG	SCDDAT, DEBCOM
GNINIT	SCDDAT, CPMDAT, CDFINC, CDFINV, CMATP, CMPTIM, CORA , COUPL CPRDAT, CPROPS, DBACCT, DEBCOM, EFFPRP, EHTO , EHTCO , FARAYS FECOM , FGINVC, FGRCOM, FPLIM , FPCOEF, FPCTRL, FPKHT FPINSH, FPMAS , FPNL , FPNONV, GRSCGR, GRSPARG, GRSPRG HARDPN, HFUSON, INTCOM, IOCOM, IPARM , IPARMM, MADATC, MATDAT MISCON, NBTIM , NDXARA, NHTARA, NRCOM , OXCOM1, OXCOM2, OXCOM3 PARM , PARMM, PLNDAT, PRDAT , PTSCOM, RADATA, RGACCT, SCDCOM SCDOOT, SLBCOM, SLUMPV, SOLCOM, TBLSP , THPLOT, TRNOT1, TRNOT2 UOXCUM, UPHDPN, SCOPW, CBOILX, CR2 , CVALS , HYDRPP, NOVAL PROPTS, SPEED , STMPRP, TTPSC , ZA , OLDCOM
HTADV	FPFAST
HTITDP	FPFAST
IHTCMP	FPFAST, SCDDAT
INPUTD	SCDDAT
IRKIN	SCDDAT, NDXARA, SCOPW, CPMDAT, COUPL
ISNGJ	CMP3DT
MAJOUT	SCDDAT, CPMDAT, COUPL , FPFAST, SLUMPV
MOVER	SCDDAT
RELAP5	DCON , FPLIM , FPCOEF, FPCTRL, FPINSC, FPINSH, FPNL FVOL , MATSRS, SCDDAT, SCDDATC, SCOPW, CPMDAT, IOCOM , SOLCOM SCDCOM, INTCOM, NHTARA, FGRCOM, SCDOOT, DBACCT, RGACCT, EFFPRP TRNOT1, TRNOT2, CMATP , NDXARA, EHTO , MISCON, TBLSP , FGINVC MATDAT, SLBCOM, GRSCGR, GRSPARG, GRSPRG, CDFINC, RADATA, CPRDAT PRDAT , MADATC, EHTCO , HFUSON, FARAYS, NBTIM , CMPTIM, THPLOT PTSCOM, CDFINV, PLNDAT, OXCOM1, OXCOM2, OXCOM3, DEBCOM, FPNONV FPMAS , SLUMPV, HARDPN, FECOM , NRCOM , UOXCUM, COUPL , IPARM IPARMC, PARM , PARMC , FPFAST, PVMVR5, VEL
RNEWP	CPMDAT, COUPL , SCDDAT
RNONCN	SCDDAT
R3DCMP	CMP3DT

Table 2.9 Modification of subroutine name

旧サブルーチン名	新サブルーチン名
STPU2PU	STPU2U

Table 2.10 Modification of dummy arguments for subroutines

副プログラム	修正内容
STPU2U	SUBROUTINE STPU2U (TABLES,S,SKP,MLU,IS,LS) SUBROUTINE STPU2U (TABLES,S,SKP,MLU,IS,LS)
STPUTP	SUBROUTINE STPUTP (TABLES,S,SKP,MLU) SUBROUTINE STPUTP (TABLES,S,SKP,MLU,IS,LS)
STPU0P	SUBROUTINE STPU0P (TABLES,S,SKP,MLU) SUBROUTINE STPU0P (TABLES,S,SKP,MLU,IS,LS)
STPU00	SUBROUTINE STPU00 (TABLES,S,SKP,MLU,ITYPE) SUBROUTINE STPU00 (TABLES,S,SKP,MLU,ITYPE,IS,LS)
STPU2P	SUBROUTINE STPU2P (TABLES,S,SKP,MLU) SUBROUTINE STPU2P (TABLES,S,SKP,MLU,IS,LS)
STPU2T	SUBROUTINE STPU2T (TABLES,S,SKP,MLU) SUBROUTINE STPU2T (TABLES,S,SKP,MLU,IS,LS)
STPU2U	SUBROUTINE STPU2PU (TABLES,S,SKP,MLU) SUBROUTINE STPU2U (TABLES,S,SKP,MLU,IS,LS)
SVPUPU	SUBROUTINE SVPUPU (TABLES,S,SKP,MLU,LVTR) SUBROUTINE SVPUPU (TABLES,S,SKP,MLU,LVTR,IS,LS)
SVPU2P	SUBROUTINE SVPU2P (TABLES,S,SKP,MLU,LVTR) SUBROUTINE SVPU2P (TABLES,S,SKP,MLU,LVTR,IS,LS)

Table 2.6 Include-files which deleted INCLUDE statements (2/2)

副プログラム名	呼び出しを廃止したインクルードファイル
SCNREQ	SCDDAT, SCDDCOM, CPMDAT, NDARA, FPFAS, HARDPN, INTCOM, MISCON PLNDAT, SCDDAT, TBLSP, THPLOT, IPARM, COUPL, SLUMPV
STATEP	SCDDAT
TRAN	SCDDAT, PYWR5, MACHAF
TRNSET	SCDDAT, CPMDAT, CMP3DT, FPFAS, COUPL, IPARM
TSETSL	CMP3DT
VIMPLT	CMP3DT

Table 2.7 Modification of include-file name

旧インクルードファイル名	新インクルードファイル名
FTBCOM	FTBCOM2
STCOM	STCOM3

Table 2.8 Deleted subprograms and include-files

RELAP5本体部	ABER, ABSORP, ACTOKP, AKZIRC, ALOXD, AMUX, ANGLE, AREA, ATOMFR, ATRMLC, ATRMLT, ATROUT, ATRSSO, ATRUOD, AUMESH, BACADV, CALTAV, CHEBYTR, CNTRLC, CRUCBR, DIRECT4, DIRECT8, DIR16, DIR2FULL, DIR2X2, DIR4FULL, GASCATC, GASCATH, GASONS, GEOMETRY, HELPT, IRSPVMC, MAJCOU, NONNEMC, NONNEMH, NONNETC, NONNETH, NONNEC, NONNEH, NONPLM, NONPLMH, NONTWOC, NONTWOH, NORHESP, OUTINTR, OUTPOINT, OUTPUTSS, OUTPUTTR, PCONTRL, PJUNDAT, PSCTRCH, PVMFXREC, PVOLDAT, RELPOWER, RICNSL, RRLINO, RR5PVMC, RSTEP, RSWTCH, RTEMP, RTPROP, SCDPRI, SCDP5H, SNAPIT, STPU2PU, TCOUPL, TMLTAX, TRIDIAO, TRUPT
同インクルード	PLTCON, PLTGET, PLTLCM, PLTZDS
Environment Library 部	ABORT, CDCVC, CMPRES, CMPRES1, DATEHP, ETIME, FTIME, IBMASH, ISHFT, LOC, READNONB, SELECTO, TIMEHP, UPTOSCS
同インクルード	STCOMC

Table 2.11 Modification of actual arguments for CALL statements (3/4)

STACC	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
STATEP	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1, TT(IX), TT(IX))
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1, TT(IX), TT(IX))
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1, TT(IX), TT(IX))
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1, TT(IX), TT(IX))
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1, TT(IX), TT(IX))
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1, TT(IX), TT(IX))
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TT(IX), 36, 1, TT(IX), TT(IX))
	C32	CALL STPUPU (FA(NDXSTD), TSAT(IX), 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), TSAT(IX), 36, 1, TSAT(IX), TSAT(IX))
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, 2, PROP, PROP)
	C32	CALL SVRUPU (FA(NDXSTD), TT(IXVF), SCSKP, NH4, IHL04A(1, IXVFF))
	C32	CALL SVRUPU (FA(NDXSTD), TT(IXVF), SCSKP, NH4, IHL04A(1, IXVFF), TT(IXVF), TT(IXVF))
	C32	CALL SVRUPU (FA(NDXSTD), TSAT(IXVF), SCSKP, NH2, IHL02A(1, IXVFF))
	C32	CALL SVRUPU (FA(NDXSTD), TSAT(IXVF), SCSKP, NH2, IHL02A(1, IXVFF), TSAT(IXVF), TSAT(IXVF))
	C32	CALL SVRUPU (FA(NDXSTD), TT(IXVF), SCSKP, NH5, IHL04A(1, IXVFF))
	C32	CALL SVRUPU (FA(NDXSTD), TT(IXVF), SCSKP, NH5, IHL04A(1, IXVFF), TT(IXVF), TT(IXVF))
	C32	CALL SVRUPU (FA(NDXSTD), TT(IXVF), SCSKP, NH5, IHL04A(1, IXVFF))
	C32	CALL SVRUPU (FA(NDXSTD), TT(IXVF), SCSKP, NH5, IHL04A(1, IXVFF), TT(IXVF), TT(IXVF))
TSATE	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1, PROP, PROP)

Table 2.11 Modification of actual arguments for CALL statements (4/4)

TSATE	C32	CALL STPUTP (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUTP (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1, S, S)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPU2T (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2P (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPU2P (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1, S, S)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1, S, S)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1, S, S)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1)
	C32	CALL STPU2P (FA(NDXSTD), S, 36, 1, S, S)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)
	C32	CALL STPU00 (FA(NDXSTD), PROP, 36, 1, 2)
	C32	CALL STPU00 (FA(NDXSTD), PROP, 36, 1, 2, PROP, PROP)
VEXPLT	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1)
	C32	CALL STPUPU (FA(NDXSTD), PROP, 36, 1, PROP, PROP)

Table 2.15 Files for restart-file transfer

M780側	JCL	GETDAT	表出	GETDAT2
VPP側	コマンド	getdat	表出	getdat2

Table 2.16 Executed results of sample problems

入力データ	TOTAL MASS 値				
	VPP500での結果		M780での結果		サンプリング結果
	値	単位	値	単位	
ANS79	1.5554	KG	1.5554	KG	1.5554
EDHTRK	0.79801	KG	0.79801	KG	0.78044
EDHTRKD	0.71228	KG	0.71228	KG	0.72154
EDHTRKN	0.78821	KG	0.78821	KG	0.77266
MARPZ04	85449.	KG	85449.	KG	85417.
PUMP2	75.468	LB	75.468	LB	75.468
TYPPWR	1.00392e+05	LB	1.00391e+05	LB	1.00380e+05

Table 2.12 Modification of unit reference number

変数名	用途	初値	変更後
INPUT	入力データ読み込み	1 1	5
OUTPUT	計算結果出力	1 2	6
RSIPLT	リストアウトファイル	1 3	2 3
STRIPF	ストリップアップファイル	1 4	2 4
PLOTFL	プロット用ワークファイル	1 5	2 5
STH2XT	スチームテーブル読み込み	1 6	2 6
EOIN	ワークファイル	5 1	5 1
COUPLF	COUPLEファイル	1 9	1 9
INPOUT	入力データエコーバック	2 0	2 0
TTY	TTY 出力	6	6 6

Table 2.13 Order of arrangement of COMMON Blocks

順序	1	2	3	4	5	6	7
COMMON名	FAST	GENRL	COMCTL	STATEC	VIRTUL	RMADAC	FTBET

Table 2.14 Declarations of work area in RELAP5/MOD3.2

変数名	インクルード	副プログラム
M780版	FAST (RELAP5本体部) FTBCOM2 (Environment Library)	なし
VPP 版	FAST, FTBCOM2	AAAAA

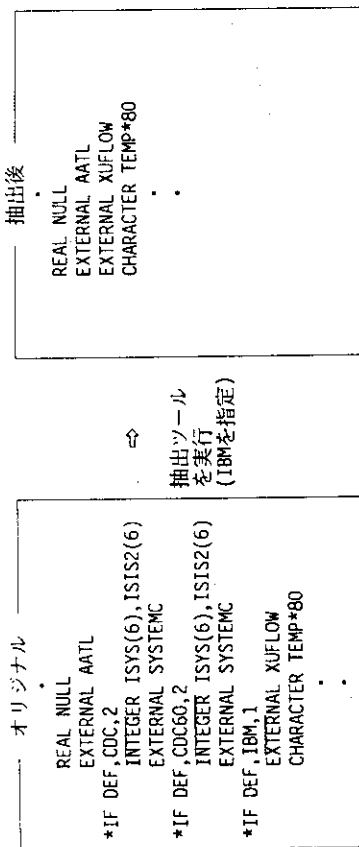


Fig. 2.1 Example of selecting program-statement by SELECTX

```

TWC(3 2 4 0 3) SRP
/**
/** ***** RELAP5/MOD2/C36 SELECT-X PROGRAM *****
/** *****
/** *****
/** *****
/** *****
EXEC PGM=SELECTX, PARM='ELM(*),MSGLEVEL(9)'
//STEPLIB DD DSN=J0001.R5TOOL.LOAD, DISP=SHR
//INSOC DD DSN=J0001.R5M32.RELAP5, DISP=SHR, LABEL=(,,IN)
//OUTSOC DD DSN=J9127.00.RELAP5.SELECT, DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=80, BLKSIZE=22000, RECFM=FB, DSORG=PO),
// SPACE=(TRK,(10,10,30),RLSE), UNIT=TSSWK
//OUTINC DD DSN=J9127.00.ENVR.L.INCLUDE,
// DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=80, BLKSIZE=22000, RECFM=FB, DSORG=PO),
// SPACE=(TRK,(3,3,10),RLSE), UNIT=TSSWK
//SYSPRINT DD SYSOUT=*, DCB=(RECFM=FBA, LRECL=137, BLKSIZE=6850)
//SYSIN DD *
*DEFINE IBM, IN32, FOURBYT, BLKOTA
$DEFINE IBM, IN32, FOURBYT, BLKOTA
/*
//
    
```

Fig. 2.3 JCL for SELECTX (Environmental Library)

```

TWC(3 4 3 0 5) SRP
/**
/** ***** RELAP5/MOD3.2 SELECT-X PROGRAM *****
/** *****
/** *****
/** *****
EXEC PGM=SELECTX, PARM='ELM(*),MSGLEVEL(9)'
//STEPLIB DD DSN=J0001.R5TOOL.LOAD, DISP=SHR
//INSOC DD DSN=J0001.R5M32.RELAP5, DISP=SHR, LABEL=(,,IN)
//OUTSOC DD DSN=J9127.00.RELAP5.SELECT, DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=80, BLKSIZE=32000, RECFM=FB, DSORG=PO),
// SPACE=(TRK,(100,10,60),RLSE), UNIT=TSSWK
//OUTINC DD DSN=J9127.00.RELAP5.INCLUDE,
// DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=80, BLKSIZE=32000, RECFM=FB, DSORG=PO),
// SPACE=(TRK,(10,10,60),RLSE), UNIT=TSSWK
//SYSPRINT DD SYSOUT=*, DCB=(RECFM=FBA, LRECL=137, BLKSIZE=6850)
//SYSIN DD *
*DEFINE BLKOTA, CROSSFL, FOURBYT, SPHACCM, GESEP, IBM
*DEFINE LEVEL, NEWTRP, BORONT, DOPTION, IN32, TFRONT
$DEFINE BLKOTA, CROSSFL, FOURBYT, SPHACCM, GESEP, IBM
$DEFINE LEVEL, NEWTRP, BORONT, DOPTION, IN32, TFRONT
/*
//
    
```

Fig. 2.2 JCL for SELECTX (RELAP5)

整数配列IAと共有する実数配列Aを
倍精度化する場合

REAL*4 A(3)
INTEGER IA(3)
EQUIVALENCE (A(1), IA(1))

REAL A(3)
A(1) A(2) A(3)

INTEGER IA(3)
IA(1) IA(2) IA(3)

↓ GRAY 同様の精度を確保するために実数を倍精度化する。

REAL*8 A(3)
A(1) A(2) A(3)

INTEGER IA(3)
IA(1) IA(2) IA(3)

↓ 実数を倍精度化したために、共有する整数配列とズレが生じる。

REAL*8 A(3)
A(1) A(2) A(3)

INTEGER IA(2,3)
IA(1,1) IA(2,1) IA(1,2) IA(2,2) IA(1,3) IA(2,3)

整数配列の次元を上げることで、倍精度実数配列とのズレを無くす。整数を定義・参照する場合は、第1添字が2の部分を使用する(太線の部分)。

REAL*8 A(3)
INTEGER IA(2,3)
EQUIVALENCE (A(1), IA(1,1))

← 倍精度実数配列Aにした時の正しい整数配列の宣言

Fig. 2.4 Concept of making a double precision

RKONX	- ADDED FOR RELAP5/M003.2
RKONNY	- ADDED FOR RELAP5/M003.2
RKONNZ	- ADDED FOR RELAP5/M003.2
RKONHR	- ADDED FOR RELAP5/M003.2
RKONG	- ADDED FOR RELAP5/M003.2
RKONGT	- ADDED FOR RELAP5/M003.2
RKONGM	- ADDED FOR RELAP5/M003.2
RKONVR	- ADDED FOR RELAP5/M003.2
RKONC	- ADDED FOR RELAP5/M003.2
RKORIF	- ADDED FOR RELAP5/M003.2
RKONZF	- ADDED FOR RELAP5/M003.2
RKONCF	- ADDED FOR RELAP5/M003.2
RKOSYM	- ADDED FOR RELAP5/M003.2
RKOTHO	- ADDED FOR RELAP5/M003.2
RKOTPS	- ADDED FOR RELAP5/M003.2
RKOTZU	- ADDED FOR RELAP5/M003.2
RKOTZD	- ADDED FOR RELAP5/M003.2
RKONNM	- ADDED FOR RELAP5/M003.2
RKOCBY	- ADDED FOR RELAP5/M003.2
RKOKIT	- ADDED FOR RELAP5/M003.2
RKOTOT	- ADDED FOR RELAP5/M003.2
RKONXY	- ADDED FOR RELAP5/M003.2
RKONND	- ADDED FOR RELAP5/M003.2
RKAZF	- ADDED FOR RELAP5/M003.2
RKZFI	- ADDED FOR RELAP5/M003.2
RKZFG	- ADDED FOR RELAP5/M003.2
RKCFI	- ADDED FOR RELAP5/M003.2
RKCFG	- ADDED FOR RELAP5/M003.2
RKZID	- ADDED FOR RELAP5/M003.2
RKCFV	- ADDED FOR RELAP5/M003.2
RKONV	- ADDED FOR RELAP5/M003.2
RKOFBS	- ADDED FOR RELAP5/M003.2
RKOSN	- ADDED FOR RELAP5/M003.2
RKOCID	- ADDED FOR RELAP5/M003.2
RKORID	- ADDED FOR RELAP5/M003.2
RKORCL	- ADDED FOR RELAP5/M003.2
RKOMCR	- ADDED FOR RELAP5/M003.2
RKORF	- ADDED FOR RELAP5/M003.2
ACCTRP	- ADDED FOR RELAP5/M003.2
ANGLTP	- ADDED FOR RELAP5/M003.2
AXSPEC	- ADDED FOR RELAP5/M003.2
CLSTRP	- ADDED FOR RELAP5/M003.2
CP111	- ADDED FOR RELAP5/M003.2
CP1EN	- ADDED FOR RELAP5/M003.2
CP1NLM	- ADDED FOR RELAP5/M003.2
CP1OPT	- ADDED FOR RELAP5/M003.2
CP1TNO	- ADDED FOR RELAP5/M003.2
CP1TRP	- ADDED FOR RELAP5/M003.2
CP1TYP	- ADDED FOR RELAP5/M003.2
CP1VNO	- ADDED FOR RELAP5/M003.2
CW1DCT	- ADDED FOR RELAP5/M003.2
CW1GEN	- ADDED FOR RELAP5/M003.2
CW1INT	- ADDED FOR RELAP5/M003.2
CW1LEN	- ADDED FOR RELAP5/M003.2
CW1NDP	- ADDED FOR RELAP5/M003.2
CW1NPA	- ADDED FOR RELAP5/M003.2

Fig. 2.5 Input data for CONV32 (RELAP5) (1/8)

INTRNO	- ADDED FOR RELAP5/MOD3.2
INVCNT	
INVFND	
INVHOS	
INVJUN	
INVQFS	
INVYNO	
INVVNX	
INXRFL	
IP	
IPMTBL	
IPWVNM	
IPWVPC	
IPWVTL	
IPWVTR	
IPR	
IPUZDI	
IPUCTR	
IPUHMI	
IPUMTK	
IPURVI	
IPUSPI	
IPUTDI	
IPUTMI	
IPUTRP	
IREG	
IRFLX	
IRHDF	
IRN	
IRNR	
ISEPST	
ISPTR	
ITEROF	
IVERT	
IWEOF	
IVRN	
IXIP	
IXIPR	
IXIPX	
IXIRN	
IXIRNR	
IXIRNX	
IXIVRN	
IXSOPR	
IXW	
IXCOFX	
IXCOPF	
JA	
JB	
JC	
JCEX	
JCNX1	
JCNX2	
JCNX3	
JCNXD	
JCNXS	
JHLD1	

Fig. 2.5 Input data for CONV32 (RELAP5) (4/8)

JHLD2	- ADDED FOR RELAP5/MOD3.2
JHLD3	- ADDED FOR RELAP5/MOD3.2
JHLD4	- ADDED FOR RELAP5/MOD3.2
JHLD5	- ADDED FOR RELAP5/MOD3.2
JHLD6	- ADDED FOR RELAP5/MOD3.2
JHLD7	- ADDED FOR RELAP5/MOD3.2
JLR	
JRH	
JUNFTL	
JUNNO	
LHTRFL	
LIA	
LIC	
LICH	
LIJ	
LJN	
LTV	
LTVN	
LIVNN	
LLVECT	
LNONCN	
LNONWF	
LPACKR	
LSUCES	
LVABRP	
LVACCH	
LVHZEL	
LVJTMX	
LVQUSR	
LVNGFR	
LVPRZ	
LVPTR	
LVPLMP	
LVRVOL	
LVSEPR	
LVTURB	
LVTVOL	
LWVALV	
LWJFR	
LV30	
MOBVOL	
MIETAB	
MICODE	
MIPCK	
MTBL	
MTBLN	
MTBNUM	
MTBPTR	
NCHPS	
NCTBLE	
NCTBLT	
NCTBTN	
NCTBTX	
NCTDPV	
NCTPC	
NCTTRP	
NCTTRX	

Fig. 2.5 Input data for CONV32 (RELAP5) (5/8)

NCVTBL	
NDBVOL	
NEMPTR	- ADDED FOR RELAP5/MOD3.2
NFEEDS	
NGTBLS	
NHTSTR	
NINDX	- ADDED FOR RELAP5/MOD3.2
NIX	
NJC	
NJCN	
NJCO	
NJUNS	
NLOOPS	
NMAPP	- ADDED FOR RELAP5/MOD3.2
NMIET	
NMTBLS	
NMZHT	
NNZ	- ADDED FOR RELAP5/MOD3.2
NNZ2	- ADDED FOR RELAP5/MOD3.2
NPUMPS	
NRELHT	
NRH	
NRHSPF	
NRSET	
KSEP	
NSTCTL	- ADDED FOR RELAP5/MOD3.2
NTABL	
NTDPVS	
NTRNV1	
NTRNV2	
NTRPC1	
NTRPC2	
NTRPFF	
NTRPHL	
NTRPNO	
NTRPNV	
NTRPOF	
NTRPOP	
NTRPS1	
NTRPS2	
NTRTR1	
NTRTR2	
NUSVAR	
NUSYS	
NVC	
NVCN	
NVCO	
NVOLS	
NVR	
NVRP	- ADDED FOR RELAP5/MOD3.2
OPNTRP	- ADDED FOR RELAP5/MOD3.2
OPTONS	- ADDED FOR RELAP5/MOD3.2
PRZJNK	- ADDED FOR RELAP5/MOD3.2
PRZLN	- ADDED FOR RELAP5/MOD3.2
PRZMOS	- ADDED FOR RELAP5/MOD3.2
PRZVAX	- ADDED FOR RELAP5/MOD3.2
QSATER	- ADDED FOR RELAP5/MOD3.2

Fig. 2.5 Input data for CONV32 (RELAP5) (6/8)

QTER	
REFSET	- ADDED FOR RELAP5/MOD3.2
RKDERI	
RKDNPT	
RKDOPI	
RKOPPT	
RKFDDC	
RKHTNO	
RKNDEN	
RKNSC	
RKNSCR	
RKNSFB	
RKNUM	
RKNUMD	
RKNWFB	
RKOFFA	
RKOFFD	
RKOWN	- ADDED FOR RELAP5/MOD3.2
RKONSR	- ADDED FOR RELAP5/MOD3.2
RKONTH	- ADDED FOR RELAP5/MOD3.2
RKONZ	- ADDED FOR RELAP5/MOD3.2
RKOPT	
RKSPTR	
RKVOLN	- ADDED FOR RELAP5/MOD3.2
RRIBLN	- ADDED FOR RELAP5/MOD3.2
RRTBLS	- ADDED FOR RELAP5/MOD3.2
SETNO	
SFLAG	- ADDED FOR RELAP5/MOD3.2
SMGRH	
SMRRHO	
SMXDRH	
SMXRHO	
SRGJUN	
SRGVOL	- ADDED FOR RELAP5/MOD3.2
STCCF1	- ADDED FOR RELAP5/MOD3.2
STCCF2	- ADDED FOR RELAP5/MOD3.2
STHZXV	
STJCK1	
STJCK2	
STJPK1	- ADDED FOR RELAP5/MOD3.2
STJPK2	- ADDED FOR RELAP5/MOD3.2
STLTE1	
STLTE2	
STRAP1	- ADDED FOR RELAP5/MOD3.2
STRAP2	- ADDED FOR RELAP5/MOD3.2
STRCL1	
STRCL2	
STRDC1	
STRDC2	
STROP1	- ADDED FOR RELAP5/MOD3.2
STROP2	- ADDED FOR RELAP5/MOD3.2
STREX1	
STREX2	
STRGEO	-CHANGED TO STRGEO FROM STRGEOH
STRPE1	
STRPE2	
STRTE1	

Fig. 2.5 Input data for CONV32 (RELAP5) (7/8)


```

PROC 0 IN('J9127.@.RELAP5.SELECT') OUT('J9127.KKK.DATA')
CONTROL LIST MSG
FREEALL
DEL (WWW W.DATA '&OUT')
ALLOC DA(WWW) NEW T CAT SP(1 1) F(W) UNIT(TSSWK) REU
OPENFILE W OUTPUT
SET W = &OUT
WRITE W
PUTFILE W
CLOSEFILE W
LIB 'J9909.SEAK.LOAD'
ALLOC DA(W.DATA) NEW T CAT SP(1 1) UNIT(TSSWK)
ALLOC DA(W.DATA) F(SYSPRINT) SHR REU
DI '&IN'
ATTR ABC LRECL(80) BLKSIZE(22000) RECFM(F B) DSORG(PO)
ALLOC DA('&OUT') NEW T CAT SP(50 50) DIR(50) RELEASE +
US(ABC) UNIT(TSSWK)
FREE F(W)
ALLOC F(FT02F001) DA(WWW) REU
ALLOC F(FT07F001) DUMMY
ALLOC F(FT08F001) DA('J0001.R5M32.MLIST5M') SHR
ALLOC F(FT16F001) DUMMY
ALLOC F(FT17F001) DUMMY
FORT177 R5M32.CONV32.FORT177 GO SSW(D)
EXIT
    
```

Fig. 2.8 Command procedure of CONV32

```

SUBROUTINE ABC
REAL A(10),B(10)
REAL*8 A(10),B(10)
INTEGER IA(10),IB(10)
INTEGER IA(2,10),IB(2,10)
EQUIVALENCE(A(1),IA(1)),(B(1),IB(1))
EQUIVALENCE(A(1),IA(1,1)),(B(1),IB(1,1))
    
```

↓ STREAM77 の実行

```

SUBROUTINE ABC
IMPLICIT REAL*8 (A-H,O-Z)
REAL A(10),B(10)
REAL*8 A(10),B(10)
INTEGER IA(10),IB(10)
INTEGER IA(2,10),IB(2,10)
EQUIVALENCE(A(1),IA(1)),(B(1),IB(1))
EQUIVALENCE(A(1),IA(1,1)),(B(1),IB(1,1))
    
```

Fig. 2.9 Example of insertion of IMPLICIT statement by STREAM77

```

REAL*8 A(12),B(10)
INTEGER IA(2,12), IB(2,10)
EQUIVALENCE (A(1), IA(1,1)), (B(1), IB(1,1))

CALL SUB( IA(1,2), IB(1,5) )
RETURN
END
SUBROUTINE SUB( A, K )
REAL*8 A(10)
REAL*8 C(10)
C(4) = A(6)
N = K + 100
RETURN
END
    
```

← IA, IB は倍精度化された実数配列と共有されているので次元上げられている。
 ← IA(2,2) はサブルーチンSUB で配列の先頭アドレスとして渡す。IB(2,5) は整数の変数として渡す。
 ← A は倍精度実数配列で、K は変数として受け取る。
 ↓
 IAとA にズレが生じる。

```

REAL*8 A(10),B(10)
INTEGER IA(2,10), IB(2,10)
EQUIVALENCE (A(1), IA(1,1)), (B(1), IB(1,1))

CALL SUB( IA(1,2), IB(2,5) )
RETURN
END
SUBROUTINE SUB(A,K)
REAL*8 A(10)
REAL*8 C(10)
C(4) = A(6)
N = K + 100
RETURN
END
    
```

← 整数として受け取る場合は、変更は第1添字を'2'にする。配列として渡す場合は、第1添字変更不要。

Fig. 2.11 Coordination of actual arguments (integer array) and dummy arguments (real array)

```

TWC(6 4 8 0 9) SRP
//*****
//**          STREAM77 ( C-ENGINE )
//*****
//CENG EXEC CENG, REGION, STRM77=
// A= READ, ERASE, NOT, NOTAB', INC='NOINCLUDE'
//FT05F001 DD *
NOSUB,
SUP(0101,0102,0200),
SUP(0300,0400,0500,0600,0700,0800,0900),
SUP(1000,1100,1200,1300,1400,1500,1600)
/*
//FT06F001 DD SYSOUT=*
//FT12F001 DD DSN=J9127.R5M32.RELAP5.NEW.FORT77, DISP=(NEW,CATLG),
// DCB=(LRECL=80, BLKSIZE=22000, RECFM=FB, DSORG=PO),
// SPACE=(TRK,(200,100,100),RLSE), UNIT=TSSWK
//FT14F001 DD DSN=J9127.R5M32.RELAP5.NEW.INCLUDE, DISP=(NEW,CATLG),
// DCB=(LRECL=80, BLKSIZE=22000, RECFM=FB, DSORG=PO),
// SPACE=(TRK,(50,20,50),RLSE), UNIT=TSSWK
//SYSIN DD DSN=J9127.@CC.RELAP5.SELECT,
// DISP=SHR, LABEL=(,,IN)
//SYSINC DD DSN=J9127.@CC.RELAP5.INCLUDE,
// DISP=SHR, LABEL=(,,IN)
//SYSPRINT DD SYSOUT=*
//
    
```

Fig. 2.10 JCL for STREAM77

```

SUBROUTINE STPUPU(TABLES,S,SKP,NLU)
*INCLUDE STCOM2
★COMMON /STCOM/ ,NU,NP, , ← インクルードSTCOM2の内容
1 ,IT2P, IT3P, IT4P
REAL*8 TABLES(*),S(SKP,* )
REAL*8 UTBL(*),PTBL(*),TPHTBL(6,NSP),SPTBL(8,NU,NP)
POINTER(IPT1,UTBL),(IPT2,PTBL),(IPT3,TPHTBL),(IPT4,SPTBL)
INTEGER IS(2,SKP,* )
LOGICAL LS(2,SKP,* )
POINTER(ISP1,IS),(ISP2,LS)

IPT1 = LOC(TABLES(1))
IPT2 = LOC(TABLES(IT2P))
IPT3 = LOC(TABLES(IT3P))
IPT4 = LOC(TABLES(IT4P))
ISP1 = LOC(S(1,1))
ISP2 = ISP1
    
```

Fig. 2.12 Example of dynamic memory allocation program

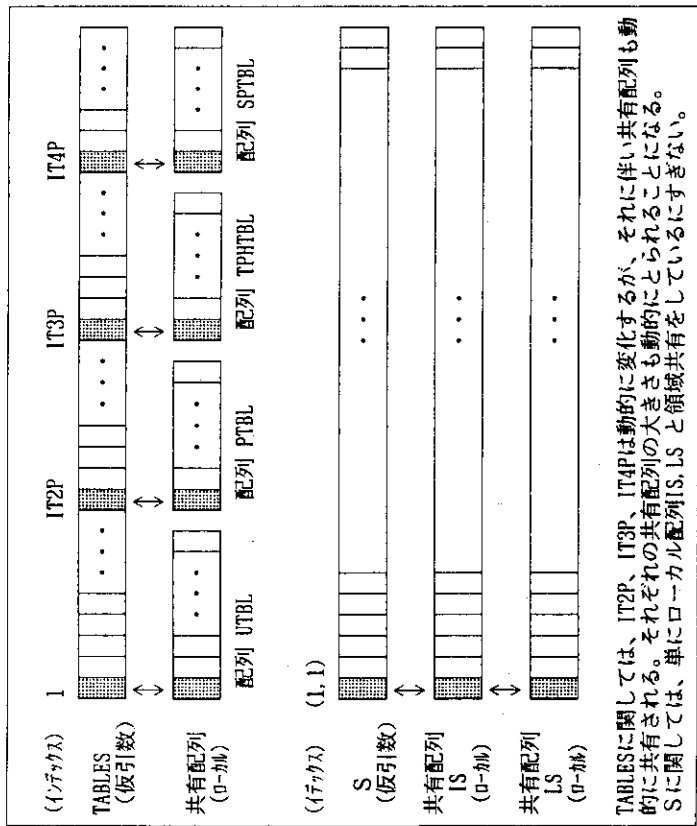


Fig. 2.13 Concept of dynamic memory allocation (Subroutine STPUPU)

```

C32 SUBROUTINE STPUPU(TABLES,S,SKP,NLU)
SUBROUTINE STPUPU(TABLES,S,SKP,NLU,IS,LS)
IMPLICIT REAL*8 (A-H,O-Z)
★COMMON /STCOM/ ,NU,NP, , ← インクルードSTCOM2の内容
1 ,IT2P, IT3P, IT4P
REAL*8 TABLES(*),S(SKP,* )
REAL*8 UTBL(*),PTBL(*),TPHTBL(6,NSP),SPTBL(8,NU,NP)
POINTER(IPT1,UTBL),(IPT2,PTBL),(IPT3,TPHTBL),(IPT4,SPTBL)
INTEGER IS(2,SKP,* )
LOGICAL LS(2,SKP,* )
POINTER(ISP1,IS),(ISP2,LS)

文脈数化 ⇒
UTBL(LX) = TABLES(LX)
PTBL(LX) = TABLES(LX+IT2P-1)
TPHTBL(LX,LY) = TABLES((LY-1)*6 + LX + IT3P - 1)
SPTBL(LX,LY,LZ) = TABLES(8*NU*(LZ-1) + (LY-1)*8 + LX + IT4P-1)

C32 IPT1 = LOC(TABLES(1))
C32 IPT2 = LOC(TABLES(IT2P))
C32 IPT3 = LOC(TABLES(IT3P))
C32 IPT4 = LOC(TABLES(IT4P))
C32 ISP1 = LOC(S(1,1))
C32 ISP2 = ISP1

END
SUBROUTINE ISTATE
C32 CALL STPUPU(FA(NDXSTD),PROP,36,1)
CALL STPUPU(FA(NDXSTD),PROP,36,1,PROP,PROP) ← 表行数の追加
END
    
```

Fig. 2.14 Modification of subroutine STPUPU

CRAWWORD形式の単精度実数 (64ビット)	
指数部	仮数部
符号ビット 15 ビット	48 ビット
絶対値の範囲: 約 10^{-2466} ~ 10^{2466}	
有効桁数: 約 14	
M形式の倍精度実数 (64ビット)	
指数部	仮数部
符号ビット 7 ビット	56 ビット
絶対値の範囲: 約 5.4^{-79} ~ 7.2^{75}	
有効桁数: 約 16	
IEEE形式の倍精度実数 (64ビット)	
指数部	仮数部
符号ビット 11 ビット	52 ビット
絶対値の範囲: 約 2.2^{-308} ~ 1.8^{308}	
有効桁数: 約 16	

Fig. 2.15 Floating point format of Cray, M, and IEEE

```

SUBROUTINE STATEP
  GCONDN = GX1/MAX(1.0E-200,GX2)
  IF (ABS(GROET).LT.1.0E-200) GROET = 1.0E-200
  END
  
```

Fig. 2.16 Original subroutine STATEP

```

SUBROUTINE STATEP
  GCONDN = GX1/MAX(1.0E-200,GX2)
  GCONDN = GX1/MAX(1.0D-75,GX2)
  C32 IF (ABS(GROET).LT.1.0E-200) GROET = 1.0E-200
  IF (ABS(GROET).LT.1.0E-75) GROET = 1.0E-75
  END
  
```

Fig. 2.17 Modification of subroutine STATEP

```

SUBROUTINE BPPARM (IBLKRC,INDEXC,INDEXI,INDNZ,IPOINT,
& IPOINI,LOCBGN,LREORD,LRINV,LRINVI,LSTART,MAPA,MULJUN,
& NDX11,NEWIPR,NEWIRN,NUMBRD)
  NBLKP = NBLK + 1
  DO I = 1,NBLK+1
    IBLKRC(NBLK+1) = LSTART(I)
    IBLKRC(2,NBLK+1) = LSTART(2,I)
  END DO
  RETURN
END
C32
  
```



```

SUBROUTINE BPPARM (IBLKRC,INDEXC,INDEXI,INDNZ,IPOINT,
& IPOINI,LOCBGN,LREORD,LRINV,LRINVI,LSTART,MAPA,MULJUN,
& NDX11,NEWIPR,NEWIRN,NUMBRD)
  NBLKP = NBLK + 1
  DO I = 1,NBLK+1
    DO 1000 I = 1,NBLK+1
      IBLKRC(NBLK+1) = LSTART(I)
      IBLKRC(2,NBLK+1) = LSTART(2,I)
    END DO
    CONTINUE
  END DO
  RETURN
END
1000
C32
C32
C32
  
```

Fig. 2.18 Modification of subroutine BPPARM

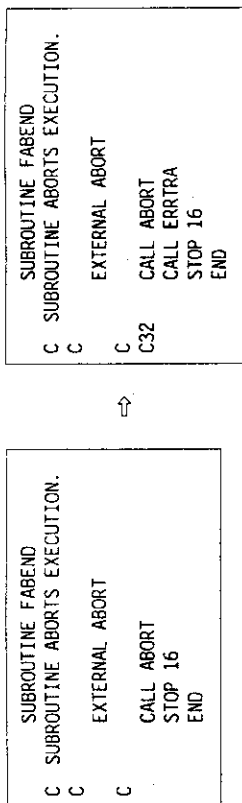


Fig. 2.19 Modification of subroutine FABEND

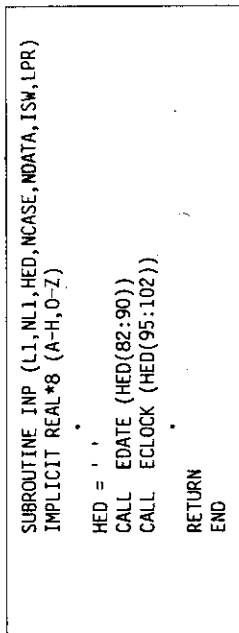


Fig. 2.20 Modification of subroutine INP

```

C ----- THIS SUBROUTINE WAS GENERATED FOR JAERI CONVERSION -----
SUBROUTINE GTINF (CONT,TYPE,OUTC,OUTI)
IMPLICIT INTEGER (A-Z)
C
C << INPUT >>
CONT : REQUIRED.
      'TIME' : 'DATE', 'CPU'
TYPE  : CHARACTER (C) OR INTEGER (I) TYPE.
C
C << OUTPUT >>
OUTC  : OUTPUT (CHARACTER)
      'TIME' : HH:MM:SS
      'DATE' : YY/MM/DD
OUTI  : OUTPUT (INTEGER)
      'TIME' : HHMMSS
      'DATE' : YYMMDD
      'CPU'  : SSSSSS (SEC)
C
CHARACTER TYPE*1,OUTC*(*),CONT*(*)
INTEGER OUTI
C
C << LOCAL VARIABLES >>
C
REAL*8 RTIME
CHARACTER CDATE*8,CTIME*8
C
LCONT = LEN(CONT)
LOUTC = LEN(OUTC)
IF (CONT(1:LCONT) .EQ. 'TIME' )
THEN
CALL TIME (ITIME)
ITIME = ITIME/1000
HH = ITIME/3600
ITIME = ITIME-3600*HH
MM = ITIME/ 60
SS = ITIME- 60*MM
IF (TYPE .EQ. 'C' )
THEN
WRITE (CTIME,'(I2.2,'',I2.2,'',I2.2,'',I2.2)') HH,MM,SS
OUTC(1:LOUTC) = CTIME
ELSE
OUTI = HH*10000 + MM*100 + SS
ENDIF
ELSE
IF (CONT(1:LCONT) .EQ. 'DATE' )
THEN
CALL DATE (CDATE)
IF (TYPE .EQ. 'C' )
THEN
OUTC(1:LOUTC) = CDATE
ELSE
READ (CDATE,'(I2,I1,I2,I1,I2)') YY,MM,DD
OUTI = YY*10000 + MM*100 + DD
ENDIF
ELSE
IF (CONT(1:LCONT) .EQ. 'CPU' )
THEN
    
```

Fig. 2.21 Subroutine GTINF (1/2)

```

C25  CALL CLOCK ( RTIME,2,2 )
      OUTI = RTIME/1000 000.0 + 0.5
      OUTJ = RTIME/1000 000.000+ 0.500
      ENDIF
C    RETURN
      END
    
```

Fig. 2.21 Subroutine GTINF (2/2)

```

SUBROUTINE INP (L1,NL1,HED,NCASE,NDATA,ISW,LPR)
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER BCD*96,ERB*80,DLRS*8
IKP = 0
READ (INPUT,1100,END=902) BCD
FORMAT (A)
IF (NCH.EQ.0) WRITE (INPUT,1001) NCASE
IF (NCH.EQ.0) WRITE (INPUT,1001) NCASE
FORMAT (' LISTING OF INPUT DATA FOR CASE',I3,' ')
END
    
```

↓

```

SUBROUTINE INP (L1,NL1,HED,NCASE,NDATA,ISW,LPR)
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER BCD*96,ERB*80,DLRS*8
IKP = 0
READ (INPUT,1100,END=902) BCD
FORMAT (A)
IF (NCH.EQ.0) WRITE (INPUT,1001) NCASE
IF (NCH.EQ.0) WRITE (INPUT,1001) NCASE
FORMAT (' LISTING OF INPUT DATA FOR CASE',I3,' ')
END
    
```

Fig. 2.22 Modification of subroutine INP

```

SUBROUTINE RSTREC
IMPLICIT REAL*8 (A-H,O-Z)
INTEGER*4 HA(2*LFSIZ)
EQUIVALENCE (HA(1),FA(1))
IXB = COMDAT(I) - IAS
WRITE (RSTPLT) COMDNLN(I),IWRD4
WRITE (RSTPLT) (HA(IN+IXB-1),IN=1,COMDNLN(I))
RETURN
END
    
```

Fig. 2.23 Original subroutine RSTREC

```

SUBROUTINE RRESTF (NOGOF)
IMPLICIT REAL*8 (A-H,O-Z)
*INCLUDE FAST
INTEGER*4 HA(2*LFSIZ)
EQUIVALENCE (HA(1),FA(1))
READ (RSTPLT,END=74,ERR=62) LX,IWRD
READ (RSTPLT,END=74,ERR=62) (HA(IN+IXB-1),IN=1,LX)
IF (RDEBUT) WRITE (OUTPUT,3003) ICCS(I),LX
RETURN
END
    
```

Fig. 2.24 Original subroutine RRESTF

```

SUBROUTINE RSTREC
IMPLICIT REAL*8 (A-H,O-Z)
INTEGER*4 HA(2*LFSIZ)
EQUIVALENCE (HA(1),FA(1))
IXB = COMDAT(I) - IAS
WRITE (RSTPLT) COMDNLN(I),IWRD4
WRITE (RSTPLT) (HA(IN+IXB-1),IN=1,COMDNLN(I))
WRITE (RSTPLT) (FA(IN+IXB-1),IN=1,COMDNLN(I))
C32
RETURN
END
    
```

Fig. 2.25 Modification of subroutine RSTREC

```

SUBROUTINE RRESTF (NOGOF)
  IMPLICIT REAL*8 ( A-H,O-Z )
  *INCLUDE FAST
  INTEGER*4 HA(2*LFSIZ)
  EQUIVALENCE (HA(1),FA(1))

  READ (RSTPLT,END=74,ERR=62) LX,IMRD
  READ (RSTPLT,END=74,ERR=62) (HA(IN+IXB-1),IN=1,LX)
  READ (RSTPLT,END=74,ERR=62) (FA(IN+IXB-1),IN=1,LX)
  IF (ROFEBUG) WRITE (OUTPUT,3003) ICGS(1),LX

  RETURN
END
    
```

Fig. 2.26 Modification of subroutine RRESTF

```

FUNCTION CHECK(KU,KR,T)
  40 CONTINUE
  .IF (IFLAG.EQ. 1) THEN
    WRITE(*,800) CHECK, ' ',ROUT(IROUT)
  ELSEIF (IFLAG.EQ. 2) THEN
    WRITE(*,800) CHECK, 'P',ROUT(IROUT)
  ELSEIF (IFLAG.EQ. 3) THEN
    WRITE(*,800) CHECK, 'D',ROUT(IROUT)
  ENDIF
  800 FORMAT(' ',G12.4,' IS OUT OF RANGE FOR ',A,' IN SUB.-',A6)
  GO TO 25
END
    
```

Fig. 2.27 Modification of function CHECK

```

BLOCK DATA BLKDTA
  IMPLICIT REAL*8 ( A-H,O-Z )
  C32CLUE EPLIM
  *INCLUDE COMCTL
  *INCLUDE CONTRL
  *INCLUDE FAST
  C32CLUE FFINSC
  C32CLUE FPARM1
  C32CLUE FPARM2
  C32CLUE FFINSH
  C32CLUE FPCTRL
  C32CLUE FPMOL
  *INCLUDE FTBCOM
  *INCLUDE MAXMEM
  *INCLUDE MAXMEM
  *INCLUDE MAXMEM
  *INCLUDE NPACOM
  *INCLUDE RCOMP
  *INCLUDE RKNMAB
  *INCLUDE RMADAC
  *INCLUDE TSCTLC
  *INCLUDE UFILEF
  *INCLUDE UFILES
  C
  C DATA FOR COMCTL COMMON.
  DATA COMDAT(1)/0/
  C DATA FOR CONTRL COMMON.
  DATA FAIL/.FALSE./, IROUTE/-1/, NCASE/-1/
  C DATA FOR FPCTRL COMMON.
  C DATA FOR FPMOL COMMON.
  C DATA FOR FTBCOM COMMON.
  C32 DATA FIRST/0.0/
  DATA FIRST/0.000/
  C DATA FOR MAXMEM COMMON.
  DATA MAXSCH/0/
  C DATA FOR MAXMEM, MAXFCV COMMONS.
  C32 DATA WMOLES/18.015340,20.03100,2.01590,6.939,39.10,4.0026,
  DATAMOLEES/18.015340,20.03100,2.01590,6.93900,39.1000,4.002600,
  C32 & 28.0134,22.9898,33.84,173.2,17.03,18.01534,92.095,19.0/
  & 28.013400,22.989800,33.8400,173.200,17.0300,18.0153400,92.09500,19
  #.000/
  DATA TPFNAM/TPFH20',TPFD20',TPFH2',TPFLI',TPFK',TPFHE',
  & TPFN2',TPFNA',TPFNAK',TPFLIPB',TPFNH3',TPFH2ON',TPFGLYC',
  & TPFBL000'/
  DATA TPFNCL /MXNFLD* ' ', TPFNIN /MXNFLD* ' ' /
  DATA FSYMBL/'H2O','D2O','H2','LI','K','HE','N2','NA','NAK',
  & 'LIPB','NH3','H2ON','GLYCEROL','BLOOD'/
  C DATA FOR NPACOM COMMON.
  DATA NPANSG/
  * MESSAGE NO DATA ENTERED WITH VALUE COMMAND, REENTER DATA.
  * MESSAGE NO USER VARIABLES DEFINED IN RELAPS INPUT, THUS NO USER
  * INPUT ALLOWED.
  * MESSAGE USER INPUT NOT PROPERLY PAIRED, REENTER DATA.
  * MESSAGE VARIABLE INPUT DOES NOT MATCH USER VARIABLES DEFINED IN
  * RELAPS INPUT.
  C DATA FOR RCOMP COMMON.
  *INCLUDE COMPAF
    
```

Fig. 2.28 Block data BLKDTA for W780 (1/3)

```

C32 *1700.00,1.72400,1850.00,1.63900,2000.00,1.56800,2150.00,1.50700,
*2300.1,457,2450.1,415,2600.1,382,3100.1,323,
*2300.00,1.45700,2450.00,1.41500,2600.00,1.38200,3100.00,1.32300,
C32 *3600.1,333,4100.1,406,4600.1,538,5100.1,730/
*3600.00,1.33300,4100.00,1.40600,4600.00,1.53800,5100.00,1.73000/
C32 DATA M4C/32.34.45,122.38.35,212.40.95,392.43.55,
DATAM4C/32.00,34.4500,122.00,38.3500,212.00,40.9500,392.00,43.5500
#
C32 *752.46.8,2012.51.35,2732.52.65,3092.56.55,
*752.00,46.800,2012.00,51.3500,2732.00,52.6500,3092.00,56.5500,
C32 *3452.63.05,3812.72.8,4352.89.7,4532.94.25,
*3452.00,63.0500,3812.00,72.800,4352.00,89.700,4532.00,94.2500,
C32 *4712.98.15,4892.100.1,5144.101.4,8000.101.4/
*4712.00,98.1500,4892.00,100.100,5144.00,101.400,8000.00,101.400/
DATA M5A/0.1,0.1,0.22,0.10/
C32 DATA M5B/392.6,936,752.8,092,1112,9,827,1472.10,983,
DATAM5B/392.00,6.93600,752.00,8.09200,1112.00,9.82700,1472.00,10.9
#8300,
C32 *1832.12.717,2192.14.451,2552.17.341,2912.20.809,
*1832.00,12.71700,2192.00,14.45100,2552.00,17.34100,2912.00,20.8090
#0,
C32 *3272.25.433,3632.31.792,3992.39.306/
*3272.00,25.43300,3632.00,31.79200,3992.00,39.30600/
C32 DATA M5C/0,28.392,1480.34.476,1675.85.176,1787.5.34.47,
DATAM5C/0.00,28.39200,1480.00,34.47600,1675.00,85.17600,1787.500,3
#4.4700,
C32 *3500.34.476/
*3500.00,34.47600/
C DATA FOR TSCTLC COMMON.
C32 DATA ERRH/8.0E-3/,ERRLO/8.0E-4/
DATA ERRH/8.00-3/,ERRLO/8.00-4/
C DATA FOR UTILEF AND UTFILES COMMONS.
C32 DATA INPUT/11/,OUTPUT/12/,RSTPLT/13/,STRIP/14/,PLOTFL/15/,
* STH2XT/16/,EOIN/51/,COUPL/19/,INPOUT/20/
DATA INPUT/05/,OUTPUT/06/,RSTPLT/23/,STRIP/24/,PLOTFL/25/,
* STH2XT/26/,EOIN/51/,COUPL/19/,INPOUT/20/
DATA JBINFO/17/
C32 DATA TTY/6/
DATA TTY/66/
DATA FILSCH/'FTBI','INDTA','OUTDTA','PLOTFL','RSTPLT','STRIP',
&'JBINFO','NULL','COFFILE','COUPL','','O','DUMPFILL','DUMPFIL2' /
END
    
```

Fig. 2.28 Block data BLKDATA for WT80 (3/3)

```

C DATA FOR RKINATB COMMON.
C32 DATA RKNITBL /1.0E0,1.020,2.0E0,1.020,4.0E0,1.021,
DATA RKNITBL /1.000,1.02000,2.000,1.02000,4.000,1.02100,
C32 *6.0E0,1.022,8.0E1,1.022,1.0E2,1.023,
*6.000,1.02200,8.001,1.02200,1.002,1.02300,
C32 *1.5E2,1.024,2.0E2,1.025,4.0E2,1.028,6.0E2,1.030,
*1.502,1.02400,2.002,1.02500,4.002,1.02800,6.002,1.03000,
C32 *8.0E2,1.032,1.0E3,1.033,
*8.002,1.03200,1.003,1.03300,
C32 *1.5E3,1.037,2.0E3,1.039,4.0E3,1.048,6.0E3,1.054,
*1.503,1.03700,2.003,1.03900,4.003,1.04800,6.003,1.05400,
C32 *8.0E3,1.060,1.0E4,1.064,
*8.003,1.06000,1.004,1.06400,
C32 *1.5E4,1.074,2.0E4,1.081,4.0E4,1.098,6.0E4,1.111,
*1.504,1.07400,2.004,1.08100,4.004,1.09800,6.004,1.11100,
C32 *8.0E4,1.119,1.0E5,1.124,
*8.004,1.11900,1.005,1.12400,
C32 *1.5E5,1.130,2.0E5,1.131,4.0E5,1.126,6.0E5,1.124,
*1.505,1.13000,2.005,1.13100,4.005,1.12600,6.005,1.12400,
C32 *8.0E5,1.123,1.0E6,1.124,
*8.005,1.12300,1.006,1.12400,
C32 *1.5E6,1.125,2.0E6,1.127,4.0E6,1.134,6.0E6,1.146,
*1.506,1.12500,2.006,1.12700,4.006,1.13400,6.006,1.14600,
C32 *8.0E6,1.162,1.0E7,1.181,
*8.006,1.16200,1.007,1.18100,
C32 *1.5E7,1.233,2.0E7,1.284,4.0E7,1.444,6.0E7,1.535,
*1.507,1.23300,2.007,1.28400,4.007,1.44400,6.007,1.53500,
C32 *8.0E7,1.586,1.0E8,1.598,
*8.007,1.58600,1.008,1.59800,
C32 *1.5E8,1.498,2.0E8,1.343,4.0E8,1.065,6.0E8,1.021,
*1.508,1.49800,2.008,1.34300,4.008,1.06500,6.008,1.02100,
C32 *8.0E8,1.012,1.0E9,1.007/
*8.008,1.01200,1.009,1.00700/
C DATA FOR RMADAC COMMON.
C32 DATA M1A/0.1,0.1,0.1,0.1,0.1/
DATA M1B/0.41562/,MIC/0.000075/
C32 DATA M1B/0.4156200/,MIC/0.00007500/
C32 DATA M2A/0.1,0.1,0.1,0.1,0.1/
DATA M2B/46.05/,M2C/3.8775E6/
DATA M2B/46.0500/,M2C/3.877506/
C32 DATA M3A/0.1,0.1,0.4,0.18/
DATA M3B/32.7,5,1700.14.506/
C32 DATA M3B/32.00,7.500,1700.00,14.50600/
C32 DATA M3C/200.57.114,300.59.118,400.61.122,500.63.126,
DATAM3C/200.00,57.11400,300.00,59.11800,400.00,61.12200,500.00,63.
#12600,
C32 *600.64.629,700.66.13,800.67.134,1000.69.138,
*600.00,64.62900,700.00,66.1300,800.00,67.13400,1000.00,69.13800,
C32 *2000.80.16/
*2000.00,80.1600/
DATA M4A/0.1,0.1,0.40,0.32/
C32 DATA M4B/500.3.341,650.2.671,800.2.677,950.2.439,
DATAM4B/500.00,3.34100,650.00,2.67100,800.00,2.67700,950.00,2.4390
#0,
C32 *1100.2.242,1250.2.078,1400.1.940,1500.1.823,
*1100.00,2.24200,1250.00,2.07800,1400.00,1.94000,1500.00,1.82300,
C32 *1700.1.724,1850.1.639,2000.1.568,2150.1.507,
    
```

Fig. 2.28 Block data BLKDATA for WT80 (2/3)

```

BLOCK DATA AAAAAA
IMPLICIT REAL*8 ( A-H,O-Z )
FAST,GENRL,COMCTL,CONTRL,STATEC,VIRTUL,RMADAC,FTBFET
DATA FOR FAST COMMON
T.NEMOTO
PARAMETER (LFSIZ=1000000)
COMMON /FAST/ FA(LFSIZ)
DATA FA(1)/0.000/
REAL*8 FA
CVPP DATA FOR GENRL COMMON
COMMON /GENRL/ CTITLE,PTITLE
CHARACTER CTITLE*108,PTITLE*64
DATA CTITLE/' '/

C DATA FOR COMCTL COMMON.
INTEGER NCOMS,NFILES
PARAMETER (NCOMS=104, NFILES=50)
COMMON /COMCTL/ COMDAT(NCOMS),COMDLN(NCOMS),FILID(NFILES),
& FILSIZ(NFILES),FILNDX(O:NFILES),FILFLG(NCOMS+1),SAFE1
REAL FILID,SAFE1
REAL*8 FILID,SAFE1
INTEGER COMDAT,COMDLN,FILSIZ,FILNDX,FILFLG
DATA COMDAT(1)/0/
CVPP DATA FOR CONTRL COMMON
T.NEMOTO
COMMON /CONTRL/ DTHY,DTHI,DTN,DT,TIMEHY,TIMEHT,ERRMAX,TMASS,EMASS,
& TMASSO,EMASSO,CPUREM,STDIRN,GRAVNC,TESTDA,SAFE2
& AFLAG,PRINT,SUCCESS,DONE,NCOUNT,NSTSP,NREPET,HELP,NARY,IRROUTE,
& IROUTE,INCASE,FAIL,UNITI,CHNGNO(90),IEXTRA,PAGENO,IEXTR2,
& IHLP(2),
& SAFE2
REAL*8 DTHY,DTHI,DTN,DT,TIMEHY,TIMEHT,ERRMAX,TMASS,EMASSO,EMASS,
& EMASSO,CPUREM,STDIRN,GRAVNC,TESTDA,SAFE2
& IROUTE,PRINT,SUCCESS,DONE,NCOUNT,NSTSP,NREPET,HELP,NARY,IRROUTE,
& IEXTRA,PAGENO,IEXTR2,CPUREI(2,5),IHP(2,5),IEXTRA,PAGENO,IEXTR2,
& IHP(2),
& IROUTE,INCASE,FAIL,UNITI,CHNGNO(90),IEXTRA,PAGENO,IEXTR2,
& IHLP(2),
& SAFE2
CVPP DATA FOR STATEC COMMON
T.NEMOTO
COMMON /STATEC/ ADVN(5),WHOLEA(5),RAX(5),DCVAX(5),CVAOX(5),
& UAOX(5),TREF(5),VISAO(5),THCA(5),THCB(5),NCTYPE(5),QN(5),
& DCONST(5),TAO,NONCN,NONWF
& NONDUM,SAFE4
REAL*8 ADVN,WHOLEA,RAX,DCVAX,CVAOX,UAOX,TREF,VISAO,THCA,THCB,
& NCTYPE,QN,DCONST,TAO,SAFE4
INTEGER NONCN,NONWF,NONHE,NONHY,NONNI,NONKR,NONXE,NONAIR,NONAR,
& NONDUM
DATA ADVN(1)/0.000/
DATA FOR VIRTUL COMMON
T.NEMOTO
COMMON /VIRTUL/ ZZZ(LFSIZ)
REAL*8 ZZZ
DATA ZZZ(1)/0.000/
CVPP DATA FOR RMADAC COMMON
MIA,MIB,MAC,M5A,M5B,M5C
& M5A,M5B,MAC,M5A,M5B,M5C
REAL*8 MIB(1),MIC(1),M2B(1),M3B(4),M3C(18)
REAL*8 M4B(40),M4C(32),M5B(22),M5C(10)
INTEGER MIA(2,4),M2A(2,4),M3A(2,4),M4A(2,4),M5A(2,4)
DATA MIA/O,1,0,1,0,1,0,1/
DATA MIB/O,4156200/,M1C/O,00007500/

```

Fig. 2.29 Block data AAAAAA (1/2)

```

DATA M2A/O,1,0,1,0,1,0,1/
DATA M2B/46.0500/,M2C/3.877506/
DATA M3A/O,1,0,1,0,4,0,18/
DATA M3B/32.00,7.500,1700.00,14.50600/
DATA M3C/200.00,57.11400,300.00,59.11800,400.00,61.12200,500.00,63.
#12600,
*600.00,64.62900,700.00,66.13000,800.00,67.13400,1000.00,69.13800,
* 2000.00,80.1600/
DATA M4A/O,1,0,1,0,40,0,32/
DATA M4B/500.00,3.34100,650.00,2.67100,800.00,2.67700,950.00,2.4390
#0,
*1100.00,2.24200,1250.00,2.07800,1400.00,1.94000,1500.00,1.82300,
*1700.00,1.72400,1850.00,1.63900,2000.00,1.56800,2150.00,1.50700,
*2300.00,1.45700,2450.00,1.41500,2600.00,1.38200,3100.00,1.32300,
*3600.00,1.33300,4100.00,1.40600,4600.00,1.53800,5100.00,1.73000/
DATA M4C/32.00,34.4500,122.00,38.3500,212.00,40.9500,392.00,43.5500
#1,
* 752.00,46.800,2012.00,51.3500,2732.00,52.6500,3092.00,56.5500,
* 3452.00,63.0500,3812.00,72.800,4352.00,89.700,4532.00,94.2500,
*4712.00,98.1500,4892.00,100.100,5144.00,101.400,8000.00,101.400/
DATA M5A/O,1,0,1,0,22,0,10/
DATA M5B/392.00,6.93600,752.00,8.09200,1112.00,9.82700,1472.00,10.9
#8300,
*1832.00,12.71700,2192.00,14.45100,2552.00,17.34100,2912.00,20.8090
#0,
* 3272.00,25.43300,3632.00,31.79200,3992.00,39.30600/
DATA M5C/O,28.39200,1480.00,34.47600,1675.00,85.17600,1787.500,3
#4.4700,
* 3500.00,34.47600/
CVPP DATA FOR FTBFET COMMON
T.NEMOTO
COMMON /FTBFET/ FTBFV
REAL*8 FTBFV
DATA FTBFV/O.000/
END

```

Fig. 2.29 Block data AAAAAA (2/2)

```

BLOCK DATA BLKDTA
IMPLICIT REAL*8 ( A-H,O-Z )
C32CLUE FPLIM
*INCLUDE COMCTL
*INCLUDE CONTRL
*INCLUDE FAST
C32CLUE FPINSC
C32CLUE EPARM1
C32CLUE EPARM2
C32CLUE FPINSH
C32CLUE FPCIRL
C32CLUE FPMDBL
*INCLUDE FTBCOM
*INCLUDE MAXMEM
*INCLUDE MAXFCD
*INCLUDE NPACOM
*INCLUDE RCOMP
*INCLUDE RKNATB
*INCLUDE RMADAC
*INCLUDE TSCITL
*INCLUDE UFILES
*INCLUDE UFILES
CVPP COMCTL, CONTRL, RMADAC COMMON BLOCK DEFINES moved
CVPP to ORDERING BLOCK DATA "AAAAAA". T.NEMOTO
C
C DATA FOR COMCTL COMMON.
CVPP DATA COMDAT(1)/0/
C DATA FOR CONTRL COMMON.
CVPP DATA FAIL/.FALSE./, IROUTE/-/, NCASE/-/
C DATA FOR FPCIRL COMMON.
C DATA FOR FPMDBL COMMON.
C DATA FOR FTBCOM COMMON.
C32 DATA FIRST/0.0/
DATA FIRST/0.000/
DATA MAXSCM/0/
C DATA FOR MAXMEM COMMON.
C DATA FOR MAXFBV, MAXFCV COMMONS.
C32 DATA WHOLES/18.0153400,20.031,2.0159,6.939,39.10,4.0026,
DATA WHOLES/18.0153400,20.03100,2.015900,6.93900,39.1000,4.002600,
C32 & 28.0134,22.9898,33.84,173.2,17.03,18.01534,92.095,19.0/
&28.013400,22.989800,33.8400,173.200,17.0300,18.0153400,92.09500,19
#.000/
DATA TPFNAM/'TPFH20', 'TPFD20', 'TPFH2', 'TPELI', 'TPFK', 'TPFHE',
& 'TPFN2', 'TPFNA', 'TPFNAK', 'TPFLIPB', 'TPFNH3', 'TPFH2ON', 'TPFGLYC',
& 'TPFBL000' /
DATA TPFNCL /MXNFELD* ' /, TPFNIN /MXNFELD* ' /
DATA FSYMBL /'H2O', 'D2O', 'H2', 'LI', 'K', 'HE', 'N2', 'NA', 'NAK',
& 'LIPB', 'NH3', 'H2ON', 'GLYCEROL', 'BLOOD' /
C DATA FOR NPACOM COMMON.
DATA NPAMSB/
* MESSAGE NO DATA ENTERED WITH VALUE COMMAND, REENTER DATA.
* MESSAGE NO USER VARIABLES DEFINED IN RELAP5 INPUT, THIS NO USER
* INPUT ALLOWED.
* MESSAGE USER INPUT NOT PROPERLY PAIRED, REENTER DATA.
* MESSAGE VARIABLE INPUT DOES NOT MATCH USER VARIABLES DEFINED IN
* RELAP5 INPUT.
C DATA FOR RCOMP COMMON.

```

Fig. 2.30 Block data BLKDTA for VPP500 (1/3)

```

*INCLUDE CMPALF
C DATA FOR RKNATB COMMON.
DATA RKNATB /1.0E0,1.020,2.0E0,1.020,4.0E0,1.020,4.0E0,1.021,
DATA RKNATB /1.000,1.02000,2.000,1.02000,4.000,1.02100,
C32 & 6.0E0,1.022,8.0E1,1.022,1.0E2,1.023,
* 6.000,1.02200,8.001,1.02200,1.002,1.02300,
C32 & 1.5E2,1.024,2.0E2,1.025,4.0E2,1.028,6.0E2,1.030,
* 1.502,1.02400,2.002,1.02500,4.002,1.02800,6.002,1.03000,
C32 & 8.0E2,1.032,1.0E3,1.033,
* 8.002,1.03200,1.003,1.03300,
C32 & 1.5E3,1.037,2.0E3,1.039,4.0E3,1.048,6.0E3,1.054,
* 1.503,1.03700,2.003,1.03900,4.003,1.04800,6.003,1.05400,
C32 & 8.0E3,1.060,1.0E4,1.064,
* 8.003,1.06000,1.004,1.06400,
C32 & 1.5E4,1.074,2.0E4,1.081,4.0E4,1.098,6.0E4,1.111,
* 1.504,1.07400,2.004,1.08100,4.004,1.09800,6.004,1.11100,
C32 & 8.0E4,1.119,1.0E5,1.124,
* 8.004,1.11900,1.005,1.12400,
C32 & 1.5E5,1.130,2.0E5,1.131,4.0E5,1.126,6.0E5,1.124,
* 1.505,1.13000,2.005,1.13100,4.005,1.12600,6.005,1.12400,
C32 & 8.0E5,1.123,1.0E6,1.124,
* 8.005,1.12300,1.006,1.12400,
C32 & 1.5E6,1.125,2.0E6,1.127,4.0E6,1.134,6.0E6,1.146,
* 1.506,1.12500,2.006,1.12700,4.006,1.13400,6.006,1.14600,
C32 & 8.0E6,1.162,1.0E7,1.181,
* 8.006,1.16200,1.007,1.18100,
C32 & 1.5E7,1.233,2.0E7,1.284,4.0E7,1.444,6.0E7,1.535,
* 1.507,1.23300,2.007,1.28400,4.007,1.44400,6.007,1.53500,
C32 & 8.0E7,1.586,1.0E8,1.598,
* 8.007,1.58600,1.008,1.59800,
C32 & 1.5E8,1.498,2.0E8,1.343,4.0E8,1.065,6.0E8,1.021,
* 1.508,1.49800,2.008,1.34300,4.008,1.06500,6.008,1.02100,
C32 & 8.0E8,1.012,1.0E9,1.007/
* 8.008,1.01200,1.009,1.00700/
C DATA FOR RMADAC COMMON.
CVPP DATA MIA/O,1,0,1,0,1,0,1/
C32 DATA MIB/O.4156200/, MIC/O.0000750/
CVPP DATA M2A/O,1,0,1,0,1,0,1/
C32 DATA M2B/46.057, M2C/3.8775E6/
CVPP DATA M3A/O,1,0,1,0,4,0,18/
C32 DATA M3B/32,7.5,1700,14.506/
CVPP DATA M3C/200,57.114,300,59.118,400,61.122,500,63.126,
C32 DATA M3C/200,57.114,300,59.118,400,61.122,500,63.126,
CVPP DATAM3C/200,57.11400,300,59.11800,400,61.12200,500,63.12600,
C32 & 600,64.629,700,66.13,800,67.134,1000,69.138,
CVPP *600,64.62900,700,66.1300,800,67.13400,1000,69.13800,
C32 * 2000,80.16/
CVPP * 2000,80.16000/
C32 DATA M4A/O,1,0,1,0,40,0,32/
CVPP DATA M4B/500,3.341,650,2.671,800,2.677,950,2.439,
C32 DATA M4B/500,3.341,650,2.671,800,2.677,950,2.439
CVPP DATAM4B/500,00,3.34100,650,00,2.67100,800,00,2.67700,950,00,2.43900
CVPP #0.
C32 & 1100,2.242,1250,2.078,1400,1.940,1500,1.823,
CVPP *1100,00,2.24200,1250,00,2.07800,1400,00,1.94000,1500,00,1.82300,

```

Fig. 2.30 Block data BLKDTA for VPP500 (2/3)

```

C32 * 1700.,1.724,1850.,1.639,2000.,1.568,2150.,1.507,
CVPP *1700.00,1.72400,1850.00,1.63900,2000.00,1.56800,2150.00,1.50700,
C32 * 2300.,1.457,2450.,1.415,2600.,1.382,3100.,1.323,
CVPP *2300.00,1.45700,2450.00,1.41500,2600.00,1.38200,3100.00,1.32300,
C32 * 3600.,1.333,4100.,1.405,4600.,1.538,5100.,1.730/
CVPP *3600.00,1.33300,4100.00,1.40500,4600.00,1.53800,5100.00,1.73000/
C32 DATA MAC/32.,34.45,122.,38.35,212.,40.95,392.,43.55,
CVPP DATAMAC/32.00,34.4500,122.00,38.3500,212.00,40.9500,392.00,43.5500
CVPP #
C32 * 752.,46.8,2012.,51.35,2732.,52.65,3092.,56.55,
CVPP * 752.00,46.800,2012.00,51.3500,2732.00,52.6500,3092.00,56.5500,
C32 * 3452.,63.05,3812.,72.8,4352.,89.7,4532.,94.25,
CVPP * 3452.00,63.0500,3812.00,72.800,4352.00,89.700,4532.00,94.2500,
C32 * 4712.,98.15,4892.,100.1,5144.,101.4,8000.,101.4/
CVPP *4712.00,98.1500,4892.00,100.100,5144.00,101.400,8000.00,101.400/
CVPP DATA M5A/0,1,0,1,0,22,0,10/
C32 DATA M5B/392.,6.936,752.,8.092,1112.,9.827,1472.,10.983,
CVPP DATAM5B/392.00,6.93600,752.00,8.09200,1112.00,9.82700,1472.00,10.98300,
CVPP #B300,
C32 * 1832.,12.717,2192.,14.451,2552.,17.341,2912.,20.809,
CVPP *1832.00,12.71700,2192.00,14.45100,2552.00,17.34100,2912.00,20.80900
CVPP #0,
C32 * 3272.,25.433,3632.,31.792,3992.,39.306/
CVPP * 3272.00,25.43300,3632.00,31.79200,3992.00,39.30600/
C32 DATA MSG/0.,28.392,1480.,34.476,1675.,85.176,1787.5,34.47,
CVPP DATAMSG/0.00,28.39200,1480.00,34.47600,1675.00,85.17600,1787.500,34.4700,
CVPP #4.4700,
C32 * 3500.,34.476/
CVPP * 3500.00,34.47600/
C DATA FOR TSCTLC COMMON.
C32 DATA ERRH/8.0E-3/,ERRLO/8.0E-4/
DATA ERRH/8.00E-3/,ERRLO/8.00E-4/
C DATA FOR UFILE AND UFILES COMMONS.
C32 DATA INPUT/11/,OUTPUT/12/,RSTPLT/13/,STRIPF/14/,PLOTFL/15/,
DATA INPUT/16/,E0IN/51/,COUPL/19/,INPOUT/20/
C32 * STHXT/16/,E0IN/51/,COUPL/19/,STRIPF/24/,PLOTFL/25/,
DATA INPUT/05/,OUTPUT/06/,RSTPLT/23/,STRIPF/24/,PLOTFL/25/,
* STHXT/26/,E0IN/51/,COUPL/19/,INPOUT/20/
DATA JBINFO/17/
C32 DATA TTY/6/
DATA TTY/66/
DATA FLSCH/ FTB1', 'INDTA', 'GOUTDA', 'PLOTFL', 'RSTPLT', 'STRIPF',
& 'JBINFO', 'NULL', 'COFFILE', 'COUPL', ' ', '0', 'DUMPFILL', 'DUMPFIL2' /
END
    
```

Fig. 2.30 Block data BLKDTA for VPP500 (3/3)

```

*** RCPU ***
THIS FUNCTION RETURNS THE REMAINING CPU TIME
FROM CURRENT JOB STEP TIME
X = RCPU ( < XE, XL > )
WHERE RCPU : REMAINING TIME ( SEC ) REAL*8
XE : ELAPSED TIME ( SEC ) REAL*8 ( OPTION )
XL : LIMITED TIME ( SEC ) REAL*8 ( OPTION )
** CAUTION ** IF RESULT IS NEGATIVE, ERROR IS DETECTED.
CSECT 12(15)
B X'7'
DC CL7*RCPU
DC 14,12,12(13)
LR RCPU,12
USING RCPU,12
ST 13,SAVE+4
LA 13,SAVE
* * *
GET ELAPSED TIME
L 4,X'214'
LM 2,3,160(4)
SRDL 2,12
STM 2,3,TIME
OI TIME,X4E
LD 2,TIME
LDR 2,2
BNP ERROR
DO 2,CORV01
LTR 1,1
BZ E01
L 5,0(1)
EQU 2,0(5)
*
E01
* * *
GET LIMITED TIME
LM 2,3,224(4)
SRDL 2,12
STM 2,3,TIME
OI TIME,X4E
LD 0,TIME
LDR 0,0
BNP ERROR
DO 0,CORV01
LTR 1,1
BZ LO1
L 5,4(1)
STD 0,0(5)
EQU *
LO1
*
    
```

Fig. 2.31 Assembly routines RCPU and LOCB (1/2)

```

* * REMAINING = LIMITED - ELAPSED
* *
* * SOR 0,2
* *
* * EQU *
* * L 13,SAVE+4
* * LM 0,12,20(13)
* * XR 15,15
* * MVI 12(13),X'FF'
* * BR 14
* *
* * EQU *
* * LD 0,MEGA
* * B RETURN
* *
* * DS 18F
* * DC D'-1.0'
* * DC D'1000000.0'
* * DS D
* * EQU X'4E'
* * END
* *
* * *** LOCB ***
* *
* * THIS FUNCTION RETURNS THE LOCATION OF VARIABLE.
* *
* * I = LOCB ( < X > )
* *
* * WHERE LOCB : LOCATION (BY BYTE)
* * X : CARIABLE
* *
* *
* * CSECT
* * L 0,0(1)
* * SLL 0,1
* * SRL 0,1
* * SRL 0,4
* * < SRL 0,1 > *** BY BYTE
* * < SRL 0,4 > *** BY 8 BYTE WORD
* * BR 14
* * END

```

Fig. 2.31 Assembly routines RCPU and LOCB (2/2)

```

C32 FUNCTION LOCF(ARG)
C32 FUNCTION LOCF(/X/)
C32 I = LOCF(ARG)
C32 LOCF = ISHFT(I,-3)
C32 IMPLICIT REAL*8 (A-H,O-Z)
C32 COMMON / FAST / A(1)
C32 DIMENSION X(1)
C32 LOGICAL FIRST
C32 DATA FIRST / .TRUE. /
C32 IF ( FIRST )
C32 THEN
C32 FIRST = .FALSE.
C32 IPOINT = LOCB(A)/8 - 2
C32 ENDIF
C32 LOCF = LOCB(X)/8.- IPOINT
C32 RETURN
C32 END

```

Fig. 2.32 Function LOCF for M780

```

C32 FUNCTION REMTIM(TIMREM)
C32 IMPLICIT REAL*8(A-H,O-Z)
C32 REAL REMTIM,TIMREM
C32 REAL*8 REMTIM,TIMREM
C32 REMTIM = 1.0E6
C32 REMTIM = 1.006
C32 REMTIM = RCPU ( )
C32 RETURN
C32 END

```

Fig. 2.33 Function REMTIM for M780


```
void locbc_(int x, int *y)
{
  x = 0x7fffffff;
  *y = x;
}
```

Fig. 2.34 C routine locbc

```
C32 FUNCTION LOCF(ARG)
FUNCTION LOCF(X)
C32 I = LOC(ARG)
C32 LOCF = ISHFT(I,-3)
CVPP COMMON / FAST / A(1)
*INCLUDE FAST
DIMENSION X(1)
LOGICAL FIRST
DATA FIRST / .TRUE. /
IF ( FIRST )
  THEN
    FIRST = .FALSE.
    IPOINT = LOCB(A)/8 - 2
    CALL LOCBC(FA,I1111)
    IPOINT = I1111/8 - 2
  ENDIF
  LOCF = LOCB(X)/8 - IPOINT
  CALL LOCBC(X,I1111)
  LOCF = I1111/8 - 2
RETURN
END
```

Fig. 2.35 Function LOCF for VPP500

```
SUBROUTINE ICOMPNI
  IMPLICIT REAL*8 ( A-H,O-Z )
  IF (LENFLN.LT.0) LENLFN = 40
  OPEN (UNIT=STH2XT,FILE=TPFNAM(N)(1:LENLFN),STATUS='OLD',
&
  & STATUS='OLD',
&
  & FORM='UNFORMATTED',IOSTAT=K)
  IF (K.EQ.0) THEN
    IF ( IAND(IHLPPR(2),ISHFT(1,22)).NE.0 )
      WRITE (OUTPUT,1015) FSYMBL(N),TPFNAM(N)(1:LENLFN)
    WRITE (TTY,1015) FSYMBL(N),TPFNAM(N)(1:LENLFN)
  1015 FORMAT (' THERMODYNAMIC PROPERTIES FILE FOR ',A,
&
& ' OBTAINED FROM LFN ',A,',')
  RETURN
END
```

Fig. 2.36 Subroutine ICOMPNI for M780

```
SUBROUTINE ICOMPNI
  IMPLICIT REAL*8 ( A-H,O-Z )
  IF (LENLFN.LT.0) LENLFN = 40
  OPEN (UNIT=STH2XT,FILE=TPFNAM(N)(1:LENLFN),STATUS='OLD',
CVPP OPEN (UNIT=STH2XT,FILE='J0001.R5M32.'//TPFNAM(N)(1:LENLFN),
&
& FILE=' /gg04/uf501/j0001/RELAP5M32ST/'//TPFNAM(N)(1:LENLFN),
&
& STATUS='OLD',
&
& FORM='UNFORMATTED',IOSTAT=K)
  IF (K.EQ.0) THEN
    IF ( IAND(IHLPPR(2),ISHFT(1,22)).NE.0 )
      WRITE (OUTPUT,1015) FSYMBL(N),TPFNAM(N)(1:LENLFN)
    WRITE (TTY,1015) FSYMBL(N),TPFNAM(N)(1:LENLFN)
  1015 FORMAT (' THERMODYNAMIC PROPERTIES FILE FOR ',A,
&
& ' OBTAINED FROM LFN ',A,',')
  RETURN
END
```

Fig. 2.37 Subroutine ICOMPNI for VPP500

```

*
* RELAP5/MOD3.2
*
* APR. 15 1996
*
*=====
*
* ORDER NEMOTO,FAST,GENRL,COMCTL,CONTRL,STATEC
* ORDER STATEC,VIRTUL,RMADAC,FTBFET
* ENTRY RELAP5
* NAME RELAP5
    
```

Fig. 2.40 linkage control data

```

TWC(6 4 8 2 5) SRP
//ASM EXEC ASM,OBJJS='300,100',SO='J0001.LT.ENVR19S',Q='.ASM'
//SYSRINT DD DUMMY
//FORTEX EXEC FORTEX,SO='J0001.R5M32.RELAP5',RGN=8M,ERGN=28M,
// A='NOAE,ELM(*)',SSN(D)',DISP=MOD
//SYSRINT DD DUMMY
//SYSINC DD DSN=J0001.R5M32.RELAP5,INCLUDE,DISP=SHR
//FORTEX EXEC FORTEX,SO='J0001.R5M32.ENVR1',RGN=8M,ERGN=28M,
// A='NOAE,ELM(*)',SSN(D)',DISP=MOD
//SYSRINT DD DUMMY
//SYSINC DD DSN=J0001.R5M32.ENVR1,INCLUDE,DISP=SHR
//LINK EXEC LKEDEXT,A='MAP',LN='J9127.R5M32EX2',UNIT=TSSWK
//SYSIN DD DSN=J0001.R5M32.LINK.PO.DATA(LINKNOAE),DISP=SHR
//
    
```

Fig. 2.41 JCL for loadmodule making

```

#!/bin/csh -f
#$-eo
#$-q gspc
#$-lt 1:00:00
#$-C RELAP5 MOD32
cd $QSUB_WORKDIR
fcpx -c -0 ../src/loc.c >& cmp.log
frtpx -c -W,-sc -Oe,-S -I../src/inc ../src/*.f >>& cmp.log
frtpx *.o >>& cmp.log
    
```

Fig. 2.42 Shell-script for loadmodule making

```

TWC(5 4 8 1 5) SRP
//FORTEX EXEC FORTEX,SO='J9127.R5M32.ENVR1.OTHER',RGN=8M,ERGN=18M,
// A='NOAE,ELM(STGBLOOD,ASTEM),SSN(D),FLAG(E)',OBJJS='100,100'
//SYSINC DD DSN=J9127.R5M32.ENVR1.INCLUDE,DISP=SHR
//LINK EXEC LKEDEX,COND=(16,LE),PRVLIB='J9127.R5M32.ENVR1',A='MAP'
//SYSIN DD *
ENTRY STGBLOO
/*
//GO EXEC GOEX,COND=(16,LE)
//FT06F001 DD DSN=J9127.R5M32.TPFBL000.OUTLIST,DISP=(NEW,CATLG),
// DCB=(LRECL=137,BLKSIZE=19043,RECFM=FBA,DSORG=PS),
// SPACE=(TRK,(10,10),RLSE),UNIT=TSSWK
//SYSIN DD DSN=J9127.R5M32.ENVR1.OTHER.DATA(STGBL001),
// DISP=SHR,LABEL=(,,,IN)
//FT01F001 DD DSN=J9127.R5M32.TPFBL000,DISP=(NEW,CATLG),
// SPACE=(TRK,(20,20),RLSE),UNIT=TSSWK
//FT02F001 DD DSN=J9127.FT02C.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(20,20),RLSE),UNIT=TSSWK
//FT07F001 DD DSN=J9127.R5M32.FT07.OUTLIST,DISP=(NEW,CATLG),
// DCB=(LRECL=137,BLKSIZE=19043,RECFM=FBA,DSORG=PS),
// SPACE=(TRK,(10,10),RLSE),UNIT=TSSWK
//
//
    
```

Fig. 2.38 JCL for Steam-table making (TPFBLOOD)

```

#!/bin/csh -f
#$-eo
#$-q vpps
#$-lt 0:30:00
#$-C RELAP5STEAM
cd $QSUB_WORKDIR
#-----relap5-----
setenv fu01 ../table/TPFBLOOD
setenv fu02 ../out/FT02.DATA
setenv fu07 ../out/FT07.DATA
timeX -hn a.out < ../input/STGBL001 >& ../out/TPFBLOOD.outlist
    
```

Fig. 2.39 Shell-script for Steam-table making (TPFBLOOD)

```
#!/bin/csh -f
#$$-eo
#$$-q gspc
#$$-lt 1:00:00
#$$-C RELAP5 MOD32
cd $SUB_WORKDIR
ftpx -o ../r5m32vpps *.o >&& cmp.log
```

Fig. 2.43 Shell-script for loadmodule making (linkage only)

```
C
INTEGER LFSIZ
CVC PARAMETER (LFSIZ=2200000)
PARAMETER (LFSIZ=280000)
COMMON /FAST/ FA(LFSIZ)
REAL FA
REAL*8 FA
C32 INTEGER IA(LFSIZ)
C32 INTEGER IA(2,LFSIZ)
C32 EQUIVALENCE (FA(1),IA(1))
EQUIVALENCE (FA(1),IA(1,1))
```

Fig. 2.44 Include-file FAST

```
C
C FTB COMMON DESCRIPTION
INTEGER LFSIZ
CVC PARAMETER (LFSIZ=500000)
PARAMETER (LFSIZ=280000)
COMMON /FAST/ A(LFSIZ)
REAL*8 A,B(LFSIZ)
INTEGER IA(2,LFSIZ)
EQUIVALENCE (A(1),IA(1,1),B(1))
COMMON /FTB/ FIRST,SIZE,MAXZ,MINZ,NOLINK,NOFILS,LINK1,LASDES,
* NEXDES,NDSK2,RECLM,IRECLT,HILO,DLT,DLY,DPN,SZZ
INTEGER FIRST,SIZE(7),MAXZ(7),MINZ(7),NOLINK,NOFILS,LINK1,LASDES,
* NEXDES,NDSK2,RECLM,IRECLT(5),SZZ(8)
LOGICAL HILO,DLT,DLY,DPN(4)
SAVE /FAST, /FTB/
```

Fig. 2.45 Include-file FTBCOM2

```
BLOCK DATA AAAAAA
IMPLICIT REAL*8 ( A-H,O-Z )
FAST,GENRL,COMCTL,CONTRL,STATEC,VIRTUL,RMADAC,FTBFET
DATA FOR FAST COMMON T.NEMOTO
PARAMETER (LFSIZ=280000)
COMMON /FAST/ FA(LFSIZ)
DATA FA(1)/0.000/
REAL*8 FA
DATA FOR GENRL COMMON T.NEMOTO
COMMON /GENRL/ CTITLE,PTITLE
CHARACTER CTITLE*108,PTITLE*64
DATA CTITLE/'.'/
END
```

Fig. 2.46 Block data AAAAAA

```
TWC(6 4 8 2 5) SRP
// EXEC LMGOEX,LM='J0001.R5M32EX',PNM=RELAP5,A=NOERRPT
//FT06F001 DD DSN=J9127.TYPPWR1.TEST.OUTPUT,DISP=(NEW,CATLG),
// DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043,DSORG=PS),
// SPACE=(TRK,(20,20),RLSE),UNIT=TSSWK
//SYSIN DD DSN=J0001.R5M32.PO.DATA(TYPPWR),DISP=SHR,LABEL=(,.,.IN)
//FT17F001 DD DSN=J9127.TYPPWR1.FT17.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(100,50),RLSE),UNIT=TSSWK
//FT19F001 DD DSN=J9127.TYPPWR1.FT19.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(100,50),RLSE),UNIT=TSSWK
//FT20F001 DD DSN=J9127.TYPPWR1.FT20.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(100,50),RLSE),UNIT=TSSWK
//FT23F001 DD DSN=J9127.TYPPWR1.RST.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(100,50),RLSE),UNIT=TSSWK
//FT24F001 DD DSN=J9127.TYPPWR1.F24.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(100,50),RLSE),UNIT=TSSWK
//FT25F001 DD DSN=J9127.TYPPWR1.F25.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(100,50),RLSE),UNIT=TSSWK
//FT51F001 DD DSN=J9127.TYPPWR1.FT41.DATA,DISP=(NEW,CATLG),
// SPACE=(TRK,(100,50),RLSE),UNIT=TSSWK
//FT66F001 DD SYSOUT=*
//
```

Fig. 2.47 JCL for Execution

```

#bin/csh -f
#$-eo
#q vpps
#-t 0:30:00
#-C RELAP5 MOD32
cd $SUB_WORKDIR
#-----relap5-----
setenv fu20 ../out/typwr.enco.data
setenv fu23 ../out/typwr.restart.data
setenv fu66 ../out/typwr.f66.data
timeX -hN j0001/relap5mod32/1m/r5m32vpps < ../input/TYPWR && *
../out/typwr.out
    
```

Fig. 2.48 Shell-script for execution

```

*voc1 total, scalar
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER(NN=30001)
DIMENSION D(NN), ID(2,NN), IW(2)
REAL*4 SD(2*NN)
CHARACTER*8 CD(NN)
EQUIVALENCE(D(1),ID(1,1),CD(1),SD(1)),(W,IW(1))
WRITE(6,*) ' ***** RESTART FILE CONVERT *****'
CPP
IEEE -> M ERROR processing (exponent overflow, underflow)
CALL ERRSET (142,256, -1,1,1,142)
CALL ERRSET (144,256, -1,1,1,144)
CALL ERRSET (146,256, -1,1,1,146)
CALL ERRSET (292,256, -1,1,1,292)
icnt = 0
100 READ(1,END=999) IW(1),IW(2)
WRITE(60) IW(1),IW(2)
icnt = icnt + 1
ND1 = 3
ND2 = 3
IF(IW(1).GT.NN) THEN
    IW(1) = NN
END IF
READ(1,END=999) (D(I),I=1,IW(1))
IF('RELAP5/M'.EQ.CD(1)) THEN
    IF(IW(1).LT.3) ND1 = IW(1)
    WRITE(60) (CD(I),I=1,ND1)
ELSE IF('PLOTINF'.EQ.CD(1)) THEN
    WRITE(60) CD(1),((ID(I,J),I=1,2),J=2,3)
ELSE IF('PLOTALF'.EQ.CD(1)) THEN
    WRITE(60) (CD(I),I=1,IW(1))
ELSE IF('PLOTINM'.EQ.CD(1)) THEN
    WRITE(60) CD(1),((ID(1,I),ID(2,I),I=2,IW(1))
ELSE IF('RESTART'.EQ.CD(1)) THEN
    WRITE(60) CD(1),((ID(1,J),I=1,2),J=2,3)
ELSE IF('PLOTREC'.EQ.CD(1)) THEN
    WRITE(60) CD(1),((SD(I),I=3,IW(1))*2)
ELSE
    WRITE(60) ((ID(I,J),I=1,2),J=1,IW(1))
    IF(IW(1).LT.3) ND2 = IW(1)
    WRITE(60) (D(I),I=1,ND2)
END IF
GO TO 100
999 write(6,*) ' ***** convert finish !! ***** '
write(6,*) ' RECORDS = ', ICNT
STOP
END
    
```

Fig. 2.49 Convert program

```
#!/bin/csh
#$-eo
#$-q vpps
#$-lt 0:30:00
#$-C RELAP5_M0D32
cd $QSUB_WORKDIR
setenv fu01 ../out/typpwr.restart.data
setenv fu00 ../ROPS.TYPPWR.RESTART.DATA
timeX -H j0001/relap5mod32/1m/r5conv -W1, -C60, -G60 && convert.out
```

Fig. 2.50 Shell-script for execution of conversion

```
TWC(5 4 8 1 9) SRP CLASS(Y)
//JOBPROC DD DSN=J9127.PROCLIB.CNTL,DISP=SHR
//GET EXEC GETDAT,TYPE=FBIN,SPEC=(1000,200)',
// DS=J9127.ROPS.TYPPWR.RST.DATA',
// FNAME=ROPS.TYPPWR.RESTART.DATA',
//
```

Fig. 2.51 JCL for GETDAT

```
TWC(5 4 8 1 5) SRP CLASS(Y)
//JOBPROC DD DSN=J9127.PROCLIB.CNTL,DISP=SHR
//GET EXEC GETDAT,TYPE=FBIN,SPEC=(1000,100)',
// DS=J9127.TYP.DATA',
// PASS=/DG03/JFS09/J9127/TEST/BINA',
// FNAME='T.DAT',
//
```

Fig. 2.52 JCL for GETDAT2

```
//GETDAT PROC SYSOUT=**,DS=,FNAME=,TYPE='FBIN',SPEC=(50,50)',
//, UNIT=TSSWK
//*-----*
//GETDAT EXEC PGM=GETDAT,PARM=#&FNAME'
//FT99F001 DD DSN=8DS,DISP=(NEW,CATLG),
// SPACE=(TRK,&SPEC,RLSE),
// DCB=(RECFM=VBS,DSORG=PS),
// DATATYPE=&TYPE,UNIT=&UNIT
//FT06F001 DD SYSOUT=*
```

Fig. 2.53 Cataloged procedure for GETDAT

```
//GETDAT PROC SYSOUT=**,DS=,FNAME=,TYPE='FBIN',SPEC=(50,50)',
//, PASS=,UNIT=TSSWK
//*-----*
//GETDAT EXEC PGM=GETDAT2,PARM=#&PASS &FNAME'
//FT99F001 DD DSN=8DS,DISP=(NEW,CATLG),
// SPACE=(TRK,&SPEC,RLSE),
// DCB=(RECFM=VBS,DSORG=PS),
// DATATYPE=&TYPE,UNIT=&UNIT
//FT06F001 DD SYSOUT=*
```

Fig. 2.54 Cataloged procedure for GETDAT2

```
#!/bin/csh -f
# command getdat
umask 077
#echo cp $1 $fu99 | atoa
#cp $HOME/$1 $fu99 | & atoa
echo cp $1 $fu99
cp $1 $fu99
```

Fig. 2.55 'getdat' command

```
#!/bin/csh -f
# command getdat2
umask 077
echo cp `echo $1 | atoa` /$2 $fu99
cp `echo $1 | atoa` /$2 $fu99
```

Fig. 2.56 'getdat2' command

3. RELAP5/MOD3.2.1.2 コードのインストール

軽水炉安全解析コード RELAP5/MOD3.2.1.2 [1] の M780, および VPP500 へのインストールを行った。このコードは、アイダホ国立研究所 (INEL) が開発しており、不定期に配布される。主に解析機能の追加、補正等がある場合に配布されるが、'MOD3.2.1.2' は配布時のバージョンを表している。原研では過去に RELAP5 コードとして、RELAP5/MOD2, RELAP5/MOD3, RELAP5/MOD3.2 等が導入されている。RELAP5/MOD3.2.1.2 コードは、前回配布された RELAP5/MOD3.2 コードの時から、数カ月で配布された。RELAP5/MOD3.2 コードは、その前に配布されたバージョン (RELAP5/MOD3) とコードが大きく変わっていた。しかし、今回は (前回と配布された) 間隔が短いため、コードの大きな変更はなかった。このために、2章で述べた RELAP5/MOD3.2 コードの変換方法とは異なる手法をとった。ここでは、この変換手順、およびこの内容について説明する。

3.1 コード概要

2章で述べた RELAP5/MOD3.2 コード [1] と同様である。ただし、RELAP5/MOD3.2 コードの解析機能補正、およびコードに含んでいたエラーの修正を行っていると思われる。

3.2 変換方針

今回配布された RELAP5/MOD3.2.1.2 コードは、これまでと同様、オリジナルのコードから使用する計算機に適合するコーディング部分を抜き出して使用する仕様になっている。M780, VPP500 の仕様に一番近い IBM 用コーディングを抽出し、変換をすることになるが、単に抽出するだけでは IBM 計算機で実行できる仕様になっていない。これは、大部分が CRAY 計算機用のコーディング仕様であり、この抽出したコードに対し、同時に配布される変換ツールで IBM 用に変換することで、IBM 版の RELAP5/MOD3.2.1.2 コードができるためである。RELAP5/MOD3.2 コードの変換は、抽出した IBM 版のオリジナルコードからツール、および手作業によって行ったが、コードが大きく、(ツールで対応できない) 手作業で変換すべき箇所が多かった。そこで今回の変換では、RELAP5/MOD3.2.1.2 コードが RELAP5/MOD3.2 コードと大きな変更がない点に注目し、次の手法をとった。この手法は、一部手作業が必要であるが、大部分の作業がツールによって行なわれ、システム化されている。このシステムの流れ図を Fig. 3.1 に示す。また下の手順で [] 内に Fig. 3.1 の流れ図で行なわれるステップを示す [2]。

- 1) RELAP5/MOD3.2, および RELAP5/MOD3.2.1.2 のオリジナルファイルより IBM 版を抽出し [SELECTX], これらと比較して異なる箇所だけを (ステートメント単位で) 取り出す [FORTCOMP]. この取り出したものを以後アップデートカードと呼ぶ。
- 2) 1) で取り出したアップデートカードに対して M780, および VPP500 のコンパイラの仕様に適応するように変換ツールを実行する [CONV32]. 場合によっては、このアップデートカードに手作業による変換を加える。

- 3) 2) で作成された変換済のアップデートカードを、RELAP5/MOD3.2 の変換済コードに組み込み [SEQNUM, UPDATE], さらに変換ツールではカバーできない変換対象部分を手作業によって変換する [補正]. これで RELAP5/MOD3.2.1.2 の変換済コードが完成する.

この手法を採用することは、RELAP5/MOD3.2 コードと RELAP5/MOD3.2.1.2 コード間の異なる箇所のみ変換することになるので、大幅な作業の削減になる。

3.3 変換のための準備

RELAP5/MOD3.2.1.2 コードは 6 枚の MS-DOS 形式の 3.5 インチフロッピーに、ソース、サンプル入力データ、サンプル出力データ等を格納して配布された。このうち、1 枚に RELAP5 ソース、サンプル問題等が格納され、他の 5 枚のフロッピーには DEC(ALPHA), IBM, Sun, HP の各計算機によるサンプル問題の出力結果と、変更のあった副プログラムが格納されていた。また、ソース、出力結果等は ZIP 形式で圧縮されていた。ソースが格納されていたフロッピー、および IBM 計算機のサンプル問題出力結果が格納されていたフロッピーの内容を Table 3.1, および 3.2 に示す。

ファイル抽出方法は、2 章で述べた RELAP5/MOD3.2 コード同様に、パソコン上でフロッピーよりファイルをダウンロードし、これらを VPP500 のフロントエンドプロセッサ GSP にファイル転送して ZIP 形式の解凍を行い、さらにこれらファイルを M780 に転送した。また、M780 に転送されたファイルはすべて大文字化した。これは、変換ツールが小文字対応になっていないためである。

3.4 変換作業 - 1

ここでは、ツールによってシステム化された変換について述べる。このシステムの流れは Fig. 3.1 に示した。この流れにそって作業を進める。

3.4.1 抽出作業

Fig. 3.1 の 'SELECTX'[2] に相当する部分である。RELAP5/MOD3.2.1.2 コードを構成する RELAP5 本体部、および Environmental Library 部より使用する計算機に適應するコーディング部分、および RELAP5/MOD3.2.1.2 が持つ各種解析機能を抽出する作業を行った。これは、「2.2.3 ソース抽出」で述べた作業と同じものである。選択パラメータも RELAP5/MOD3.2 コードのものと同じである。例として、RELAP5 本体部の抽出する JCL を Fig. 3.2 に示す。

次に述べる旧版との比較作業のために、RELAP5/MOD3.2 コードに対しても、この抽出作業を行う。

3.4.2 比較作業

Fig. 3.1 の 'FORTCOMP'[2] に相当する部分である。ここでは、抽出作業でオリジナルより取り出した RELAP5/MOD3.2, および RELAP5/MOD3.2.1.2 の IBM 版コードを、ツール

‘FORTCOMP’によって比較して、異なる箇所(アップデートカード)を取り出す。比較は同名の副プログラムごとに行い、異なる箇所はシーケンス番号とともに出力される。アップデートカードの例を Fig. 3.3 に示す。また、比較作業を行う JCL の例を Fig. 3.4 に示す。

3.4.3 変換作業

Fig. 3.1 の ‘CONV32’[2] に相当する部分である。RELAP5/MOD3.2 コードの変換で使ったツール ‘CONV32’ と同じものを使用する。比較作業でできたアップデートカードに対して、この変換を行う。‘CONV32’ は、次元上げをすべき整数配列名を入力する必要がある。RELAP5/MOD3.2 コードの変換時に使用した入力データに、今回調査を行って次元上げすべき整数配列名を加えた。この入力データを RELAP5 本体部について Fig. 3.5 に示す。Environmental Library は RELAP5/MOD3.2 コード同様に全ルーチンで共通の次元上げ対象の整数配列が無いため、ダミーの入力データを用いた。変換を施したアップデートカードの例を Fig. 3.6 に示す。また、このツールを実行するコマンドプロシジャを Fig. 3.7 に示す。

3.4.4 アップデート作業

Fig. 3.1 の ‘SEQNUM’[2] と ‘UPDATE’[2] の2つのステップで行う作業で、変換作業でできた変換済アップデートカードを使って、(変換済)RELAP5/MOD3.2 コードをアップデート(挿入・削除)する。初めに、ツール ‘SEQNUM’ によって、RELAP5/MOD3.2 の IBM 版ソースに対応するシーケンス番号を RELAP5/MOD3.2 の変換済コードにつける。その後ツール ‘UPDATE’ が、アップデートカードを使って、シーケンス番号付き RELAP5/MOD3.2 コードにアップデートを行う。このイメージを Fig. 3.8 に示す。この作業により、(まだ完全ではないが) RELAP5/MOD3.2.1.2 コードが作成される。ツール ‘SEQNUM’ によるシーケンス番号を付ける JCL を Fig. 3.9、ツール ‘UPDATE’ によってアップデートを行う JCL を Fig. 3.10 に示す。

3.5 変換作業-2

3.4で作成された RELAP5/MOD3.2.1.2 コードは、まだ完全なものではない。これは、ツールでは対応できない変換やオリジナルコードに含まれるエラーが存在するためである。

3.5.1 不要な副プログラムの分離・削除

RELAP5/MOD3.2 コードと同様、コードの実行に不必要な副プログラムを分離・削除した。RELAP5 本体部に関しては、どこからも呼び出しが無い副プログラムを削除した。分離・削除した副プログラムを Table 3.3 に示す。

Environmental Library に関しては、RELAP5/MOD3.2 コード同様、FORTRAN 記述でないもの、スチームテーブルの入力データ、および作成ルーチンである。

3.5.2 IMPLICIT 文の挿入

今回採用した変換方法は、RELAP5/MOD3.2 コードと RELAP5/MOD3.2.1.2 コードを比較して異なる部分のみを変換し、RELAP5/MOD3.2 コードに組み込むことである。これら

2つのコードは、大きな違いがないために、RELAP5/MOD3.2.1.2 コードの大部分は、RELAP5/MOD3.2 コードのものである。このため、前回の交換で挿入した IMPLICIT 文は、そのまま存在する可能性が高い。しかし、副プログラムが大幅に変わっていたり、今回追加された副プログラムに、IMPLICIT 文が無い可能性もあるため、前回同様 STREAM77[3] を使用して IMPLICIT 文が存在しない副プログラムに IMPLICIT 文を挿入することにした (STREAM77 の起動方法は 2 章で述べた)。

3.5.3 整数配列の次元上げ

EQUIVALENCE 文等により実数型配列と共有する整数型配列は、倍精度化に伴い次元上げをする必要がある。整数型配列の次元上げは 2.3.1 で述べたツール 'CONV32'[2] で行った。しかし、'CONV32' の仕様として、すべての副プログラムで共通に次元上げする整数型配列は入力データで指定するが、副プログラム内だけの対象となる整数型配列は、この副プログラム内で整数型配列名を宣言することになっている。この宣言文は '*IN32 (整数型配列名)' という形式で、オリジナルコードに挿入されている。しかし、今回の交換では、アップデートカードに対して 'CONV32' を実行しているため、このような宣言文は含まれていない。このため、手作業によって次元上げを行った。次元上げを行った副プログラム名、および整数型配列を Table 3.4 に示す。

3.5.4 行番号なし DO ループの修正

行番号のない DO ループがサブルーチン CHKLEV に 2 か所、サブルーチン LEVEL に 9 か所、TRNSET に 2 か所存在した。また、LEVEL には行番号なしの DO ~ WHILE が 1 か所あった。これらは、VPP500 では実行できるが、M780 では実行できないので行番号付きの形に変えた。修正例として CHKLEV のひとつを Fig. 3.11 に示す。

3.5.5 エラー修正

サブルーチン GCSUB において (タイプミスと思われる) コーディングのずれがあった。これを正しく修正した。これを Fig. 3.12 に示す。

3.6 M780 版と VPP500 版との違い

M780 版と VPP500 版とのソースでは、計算機の仕様が異なるため、一部の副プログラムで異なる箇所がある。ここでは、仕様の違いによるコーディングの変更点を述べる。

3.6.1 COMMON の順序付け

RELAP5/MOD3.2.1.2 コードは、これまでに配布された RELAP5 コードと同様、メモリ上で COMMON の順序付けが必要である (M780, および VPP500 での COMMON の順序付けの方法は 2 章で述べた)。

RELAP5/MOD3.2 コードと RELAP5/MOD3.2.1.2 コードの順序付けの対象となる COMMON は、まったく同じである。このため、M780 においては、RELAP5/MOD3.2 コードで作成し

たCOMMONの順序を指定するリンケージ制御文を、ロードモジュールの作成時に使用することが可能である。

VPP500においては、ブロックデータにおいてCOMMONの順序付けを行うが、これもRELAP5/MOD3.2コードと同じもの(ブロックデータAAAAAA)が使用できる。

3.6.2 アセンブラとC

RELAP5/MOD3.2.1.2コードのM780版は、RELAP5/MOD3.2コードと同様にアドレスの取得とCPUの残り時間の取得にアセンブラを使用する。これらのアセンブラを呼び出すルーチンの仕様は、RELAP5/MOD3.2コードと変わっていないため、同じアセンブラを使用する。

同様にVPP500版も前回と同じCルーチンを使用する(ただし、アドレス取得のみ)。

3.6.3 スチームテーブルについて

今回のスチームテーブルの作成ルーチン、および入力データはRELAP5/MOD3.2コードのものと同様であった。このため、RELAP5/MOD3.2.1.2コードの実行には、M780、VPP500共に前回作成したスチームテーブルを使用する。また、スチームテーブル読出ルーチンも前回のものと同じであるため、M780版、VPP500版のそれぞれを使用した。

3.7 実行ロードモジュールの作成

ここではM780、およびVPP500でのロードモジュールを作成するためのコンパイル手順・リンク等について述べる。

3.7.1 M780上でのロードモジュール作成

M780上でのロードモジュールの作成方法は、RELAP5/MOD3.2コードと(リンケージ入力データを含めて)すべて同じである。ロードモジュール作成JCLをFig. 3.13に示す。

3.7.2 VPP500上でのロードモジュール作成

VPP500上でのロードモジュールの作成方法も、RELAP5/MOD3.2コードとすべて同じである。すべての副プログラムをコンパイルし、ロードモジュールを作成するシェルスクリプトcmp.sh、およびコンパイルを行わずすべてのオブジェクトモジュールをリンクしてロードモジュールを作成するシェルスクリプトcmp2.shをFig. 3.14, Fig. 3.15に示す。

3.7.3 作業領域について

RELAP5/MOD3.2.1.2コードは解析対象によっては、作業領域として使用される配列を大きくしてロードモジュールを作成する必要がある。作業領域の指定はRELAP5/MOD3.2コードのものと同じであるので、2章を参考にされたい。

3.8 実行方法について

実行方法については、M780版、VPP500版ともにRELAP5/MOD3.2コードとすべて同じである。M780での実行JCLをFig. 3.16、VPP500でのスタート・リスタート用実行シェルスクリプトFig. 3.17に示す。

また2章で述べたように、VPP500では図形出力(ROPS[4])のために、リスタートファイルをコンバートプログラムによってM形式に変換しM780へファイル転送を行う必要がある。リスタートファイルの構造はRELAP5/MOD3.2コードのものと同じであるので、コンバートプログラムの使用方法も同じである。

3.9 REALP5/MOD3.2.1.2 コードの出力結果の検証

M780上、およびVPP500上で今回変換したコードが正常に動作するか確認するために、9種類のサンプル問題を実行した。またその結果をコードと共に送付されてきたサンプル出力結果と比較を行ってみた。比較には、計算最終ステップにおける計算体系の全質量(total mass)を使って行なった。この結果をTable 3.5に示す。なお、サンプル問題のedrstはリスタート機能のチェック用、同じくedstripはストリップファイル(リスタートファイルの編集ファイル)作成用で、本来の解析計算とは関係ないため、この表から除いた。ただし、両者とも正常に動作している。計算結果は一致し、妥当な結果を得ることができた。よって、コードは正しく変換できたと思われる。

3.10 まとめ

前回配付されたRELAP5/MOD3.2コードは全ての副プログラムについて、一から変換を行ったために、作業量、および工数を多く必要とした。今回のRELAP5/MOD3.2.1.2コードは、RELAP5/MOD3.2コードと大きな変更がなかったために、今回の(異なる箇所のみ変換する)変換方法を採用した。このために、大幅に作業量、工数を少なくすることができた。

RELAP5シリーズのように、バージョンアップごとに配付され変換を行う必要があるものは、その時のコードの状態によって、変換方法を使い分けることが大事であると思われる。

最後に付録Bに、RELAP5/MOD3.2.1.2コードのツリー構図を示した。参考にされたい。

Table 3.1 List of distributed files

フロッピー	ファイル名	内 容
N.O. 1	r1p3212.zip guides.asc pkunzip.exe	RELAPSソース等 ソースの展開方法 ZIP 解凍ロードモジュール
N.O. 2	ibmrisc.zip pkunzip.exe	IBM 計算機によるサンプル問題出力結果 ZIP 解凍ロードモジュール

Table 3.2 Contents of ZIP-files

ZIP ファイル	ファイル名	内 容
r1p3212.zip	CWN32.F DINSTLS DUTILTY ENVR.S GOODIES INDEXES SELAP.S SELECT.F USPLIT.F	IBM 用変換ソース ソース、入力データ抽出シエルスクリプト ソースのロードモジュール作成シエルスクリプト Environmental Library ソース ソース、入力データ抽出シエルスクリプト ソース、入力データ抽出シエルスクリプト RELAPS本体ソース ソース抽出ツール ソース抽出ツール
ibmrisc.zip	ANS79.P DINSTLS.LOG EDHTRK.P EDHTRKO.P EDHTRKN.P EDRST.P EDSTRIP.P EDSTRIP.ST MARPDZ04.P PUMP2.P TYPPWR.P	サンプル問題 ANS79 出力結果 IBM 計算機上でのツール、コンパイル等のlog サンプル問題 EDHTRK 出力結果 サンプル問題 EDHTRKO出力結果 サンプル問題 EDHTRKN出力結果 サンプル問題 EDRST 出力結果 サンプル問題 EDSTRIP出力結果 サンプル問題 EDSTRIP strip file 出力結果 サンプル問題 MARPDZ04出力結果 サンプル問題 PUMP2 出力結果 サンプル問題 TYPPWR 出力結果

Table 3.3 Separated or deleted subprograms

RELAPS本体部	ABER, ABSORP, ACTOKP, AKZIRC, ALDXD, AMUX, ANGLE, AREA, ATOMER, ATRMLC, ATRMLT, ATROUT, ATRSSO, ATRUDD, AJMESH, BACROV, DEBINT, MAJCOU, MCHCK, NONTMOH, PCONTRL, POUNDAT, PYOLDAT, SNAPIT
Environmental Library	スチームデータベース・ルーチン・入力データ FORTRAN 記述でないもの

Table 3.4 Modification of integer arrays

副プログラム名	次元を上げた整数型配列
FWRAG	BADFW
GEDRY	IHL, IH2
IEDIT	IPX
ILEVEL	VOLNO
LEVEL	TESTT
PHANTV	NWAPP
RBRNCH	INIT
RCHNG	CHNGIN
RMPLJ	INIT
RSNGJ	INIT
RTURB	INIT, LFLAG
RVALVE	INIT
STAPEP	LFLAG

Table 3.5 Executed results of sample problems

入力データ	TOTAL MASS 値			
	VPP500での結果	M780での結果	サンプル結果	サンプル結果
ANS79	1.5554	1.5554	KG	1.5554
EDHTRK	0.78044	0.78044	KG	0.78044
EDHTRKO	0.72154	0.72154	KG	0.72154
EDHTRKN	0.77266	0.77266	KG	0.77266
MARPZD4	85417.	85417.	KG	85417.
PUMP2	75.468	75.468	LB	75.468
TYPPWR	1.00380e+05	1.00380e+05	LB	1.00380e+05

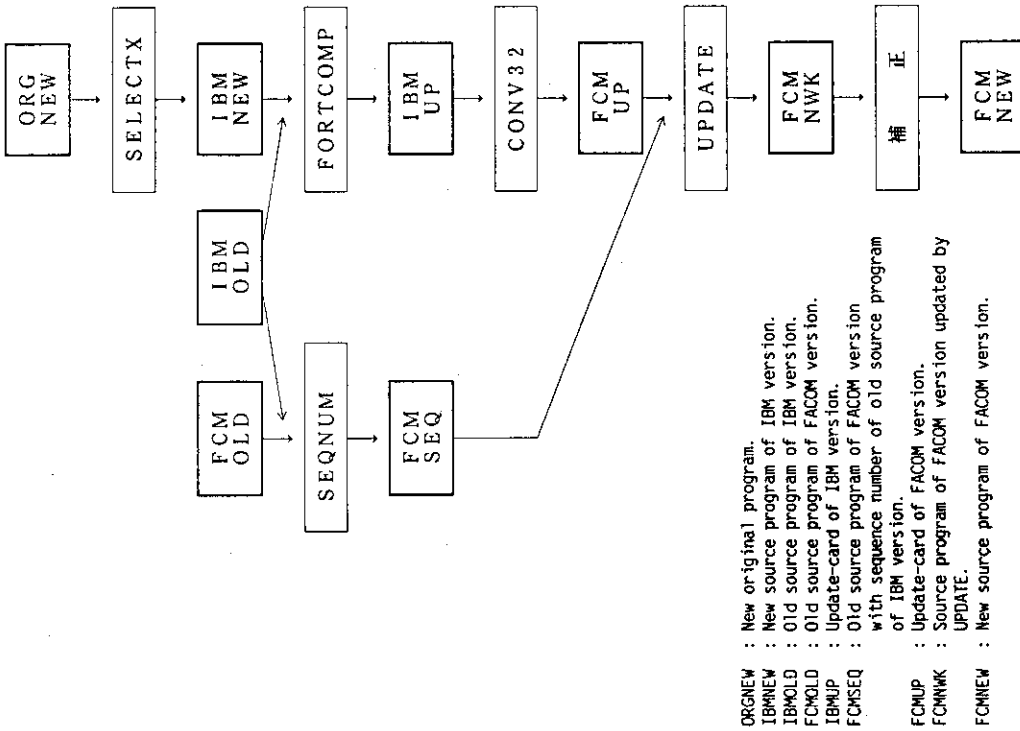


Fig. 3.1 General flow chart of conversion of RELAP5 by using tools

```

TWC(3 4 3 0 5) SRP
/*
/****** RELAP5/MOD3.2 SELECT-X PROGRAM *****
/******
/*
// EXEC PGM=SELECTX,PARM='ELM(*),MSGLVL(9)'
//STEPLIB DD DSN=J0001.R5TOOL.LOAD,DISP=SHR
//INSDC DD DSN=J9127.00.R5M3212.SELECT.S,DISP=SHR,LABEL=(...,IN)
//OUTSOC DD DSN=J9127.00.R5M3212.SELECT,DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=80,BLKSIZE=32000,RECFM=F8,DSORG=PO),
// SPACE=(TRK,(100,50,60),RLSE),UNIT=TSSWK
//OUTINC DD DSN=J9127.00.R5M3212.SELECT.INCLUDE,
// DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=80,BLKSIZE=32000,RECFM=F8,DSORG=PO),
// SPACE=(TRK,(10,10,60),RLSE),UNIT=TSSWK
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=137,BLKSIZE=6850)
//SYSIN DD *
*DEFINE BLKDTA,CROSSFL,FOURBYT,SPHACCM,GESEP,IBM
$DEFINE LEVEL,NEWTRP,BORONT,DOPTION,IN32,TFRONT
$DEFINE BLKDTA,CROSSFL,FOURBYT,SPHACCM,GESEP,IBM
$DEFINE LEVEL,NEWTRP,BORONT,DOPTION,IN32,TFRONT
/*
//
  
```

Fig. 3.2 JCL for SELECTX (RELAP5)

```

*DELETE 111 - 112
*INSERT 112
PS = (PMRHD(1)/PMRFL(1))*(ETA*TT*HD + (1.0-ETA)*HT*SLOPE)
  
```

Fig. 3.3 Example of Update-card (subroutine PUMP2)

```

TWC(3 4 8 0 9) SRP
/*
/****** RELAP5/MOD2/C36 FORTRAN COMPARE PROGRAM *****
/******
/*
// EXEC PGM=FORTRCOMP
//COMPARE DD DSN=J0001.R5TOOL.LOAD,DISP=SHR
//STEPLIB DD DSN=J9127.00.R5M32.SELECT,
//OLDSOC DD DISP=SHR,LABEL=(...,IN)
//NEWSOC DD DSN=J9127.00.R5M3212.SELECT,DISP=SHR,LABEL=(...,IN)
//UPDATCD DD DSN=J9127.00.UPDATCD.DATA,DISP=(NEW,CATLG),
// UNIT=TSSWK,SPACE=(TRK,(100,10,50))
//SYSPRINT DD DSN=J9127.00.R5M32.FORTRCOMP,DISP=(NEW,CATLG),
// DCB=(LRECL=137,RECFM=FBA,DSORG=PS,BLKSIZE=27400),
// SPACE=(TRK,(50,50),RLSE),UNIT=TSSWK
//
//
  
```

Fig. 3.4 Example of JCL for FORTRCOMP

ACTRP	
ANGLTP	
AXSPEC	- ADDED FOR RELAP5/MOD3.2
CLSTRP	- ADDED FOR RELAP5/MOD3.2
CMPIII	
CMPLN	
CMNUM	
CMPOPT	
CMPTNO	
CMTRP	
CMTTY	
CMVNO	
CNVDCI	
CNVGEN	
CNVINT	
CNVLEN	
CNVNCP	
CNVNPA	
CNVNUM	
CNVOPT	
CNVPCX	
CNVPMH	
CNVSCP	
CNVSCT	
CNVTRP	
CNVTTY	
COMPID	
CONIID	
CPUREI	- ADDED FOR RELAP5/MOD3.2
CRPNUM	- ADDED FOR RELAP5/MOD3.2
CRVNUM	- ADDED FOR RELAP5/MOD3.2
CURCLM	
CURCHI	
CURCAJ	
CURCRS	
CURCTL	
CURVSP	
CVRTNO	
FLOWAP	
FLOWP3	
FRACTV	
FPHSMT	
FPHSNO	
FPHSP	
FPHSPR	
FPHSPT	
FPHSVN	
FPINT	
FPIPTR	
FPIVB	
FPIVE	
FPLVB	
FPLVE	
FPNBLN	
FPNHS	
FPNSP	
FPNSYS	

Fig. 3.5 Input data for CONV32 (RELAP5) (1/8)

FPSCHI	
FPSCHR2	
FPSCH3	
FPSCHA	
FPSNO	
FPSYSP	
FPYOLN	
FPHSKP	
GENDR	
GENOPS	
GENTRP	
GTBPTR	
GTINFO	
GLEN	
GNUM	
GTRP	
GTTY	
HTBNT	
HTBNTR	
HTBNTS	
HTBYC	
HTBYO	
HTCMP	
HTFLAG	
HTGAP	
HTGMR	
HTGOM	
HTISCR	
HTMAXL	
HTNMPT	
HTNUSR	
HTOPT	
HTRFLG	
HTSRT	
HTSTNO	
HTSTYP	
HIXFT	
HTXIT	
IA	
IALPOF	
IALPOF	
IAREOF	
IDBVOL	
IDIN	
IGLREL	
IHH1	
IHH2	
IHH3	
IHLD1	
IHLD10	
IHLD2	
IHLD2A	
IHLD3	
IHLD4	
IHLD4A	
IHLD5	
IHLD6	

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED (RACHTC)

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

Fig. 3.5 Input data for CONV32 (RELAP5) (2/8)

IHLDTA	- ADDED FOR RELAP5/MOD3.2
IHLDB	- ADDED FOR RELAP5/MOD3.2
IHLD9	- ADDED FOR RELAP5/MOD3.2
IHTPTR	
IJ1	
IJ1X	
IJ1W	
IJ2	
IJ2X	
IJ2W	
IJFLG	
IMAP	
IMW	- ADDED FOR RELAP5/MOD3.2
INDZHS	
INDZHT	
INHCNT	- ADDED FOR RELAP5/MOD3.2
INHINO	- ADDED FOR RELAP5/MOD3.2
INSRFT	
INTRNI	
INTRNO	
INVCNT	
INVCNT	
INWFND	
INWHOS	
INWHTE	
INWJUN	
INVOFS	
INVVNO	
INVVNX	
INXRFL	
IP	
IPMTBL	
IPWVNH	
IPWVPC	
IPWVTL	
IPWVTR	
IPWVTR	
IPR	
IPUZDI	
IPUCTR	
IPURMI	
IPURTK	
IPURVI	
IPUSPI	
IPUTDI	
IPUTMI	
IPUTRP	
IREG	
IRKFLX	
IRKFLX	(RADHTC)
IRKOFF	
IRKOFF	(RADHTC)
IRN	
IRNR	
ISEPST	
ISPTR	- ADDED FOR RELAP5/MOD3.2
ITEMOF	
IVERT	- ADDED FOR RELAP5/MOD3.2

Fig. 3.5 Input data for CONV32 (RELAP5) (3/8)

IVEWOF	
IVRN	- ADDED FOR RELAP5/MOD3.2
IXCOFP	- ADDED FOR RELAP5/MOD3.2
IXCOFX	- ADDED FOR RELAP5/MOD3.2
IXIP	- ADDED FOR RELAP5/MOD3.2
IXIPR	- ADDED FOR RELAP5/MOD3.2
IXIPX	- ADDED FOR RELAP5/MOD3.2
IXIRN	- ADDED FOR RELAP5/MOD3.2
IXIRNR	- ADDED FOR RELAP5/MOD3.2
IXIRNX	- ADDED FOR RELAP5/MOD3.2
IXIRNX	- ADDED FOR RELAP5/MOD3.2
IXSOPR	- ADDED FOR RELAP5/MOD3.2
IXW	- ADDED FOR RELAP5/MOD3.2
JA	
JB	
JC	
JCEX	- ADDED FOR RELAP5/MOD3.2.1.2
JCNFND	
JCNX1	
JCNX2	
JCNX3	
JCNXD	
JCNXS	
JHLD1	- ADDED FOR RELAP5/MOD3.2
JHLD2	- ADDED FOR RELAP5/MOD3.2
JHLD3	- ADDED FOR RELAP5/MOD3.2
JHLD4	- ADDED FOR RELAP5/MOD3.2
JHLD5	- ADDED FOR RELAP5/MOD3.2
JHLD6	- ADDED FOR RELAP5/MOD3.2
JHLD7	- ADDED FOR RELAP5/MOD3.2
JJET	
JR	
JRH	
JUNFTL	
JUNNO	
JUNNO	- ADDED FOR RELAP5/MOD3.2.1.2
LFLAG2	
LHTRFL	
LIA	
LIC	
LICN	
LIJ	
LJN	
LIV	
LIVN	
LIVN	
LLVECT	
LNONCN	
LNONCN	
LNONMF	
LPACKR	
LSACES	
LV30	
LVABRP	
LVACCM	
LVAJUN	- ADDED FOR RELAP5/MOD3.2.1.2
LVAVOL	- ADDED FOR RELAP5/MOD3.2.1.2
LVHZFL	
LVJTHX	

Fig. 3.5 Input data for CONV32 (RELAP5) (4/8)

LVJUSR
LVNDFR
LVPRZ
LVPTR
LVPUMP
LVRVOL
LVSEPR
LVURB
LVVOL
LVVALV
LVWIFR
MICODE
MIETAB
MIPCK
MTBL
MTBLEN
MTBNUM
MTBTR
MOPPS
NCTBLE
NCTBLT
NCTBTM
NCTBTX
NCTDPV
NCTPC
NCTTRP
NCTTRX
NCVTBL
NDBVOL
NEMPTR
NEEDS
NGTBLS
NHTSTR
NINDEX
NIX
NJC
NJCN
NJCO
NJUNS
NLOOPS
NMAPP
MILET
MTBLS
MIZHT
MIZ
MIZ2
NPUMPS
NRELHT
NRH
NRHSPK
NRSET
NRSET
NSEP
NSTCTL
NTABL

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2.1.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

(RADHTC)

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

Fig. 3.5 Input data for CONV32 (RELAP5) (5/8)

NTRPVS
NTRW1
NTRW2
NTRPC1
NTRPC2
NTRPFF
NTRPML
NTRPMO
NTRPMV
NTRPOF
NTRPOP
NTRPS1
NTRPS2
NTRIR1
NTRIR2
NUSVAR
NUSYS
NVALHI
NVALHX
NVC
NVCH
NVCO
NVOLS
NVR
NVRP
OPNTRP
OPTONS
PRZJMX
PRZLLN
PRZMDS
PRZVNA
QSAITER
QTER
RRESET
RKDEMI
RKDNPT
RKDOP1
RKDPPT
RKFOED
RKHINO
RKINDN
RKNSC
RKNSCR
RKNSFB
RKNUM
RKNUMD
RKRVFB
RKQAZF
RKOCBY
RKOCFG
RKOCFI
RKOCID
RKOFBS
RKOFBV
RKOFFA
RKOFFD
RKOTHO

- ADDED FOR RELAP5/MOD3.2.1.1.2

- ADDED FOR RELAP5/MOD3.2.1.1.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

(RADHTC)

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

- ADDED FOR RELAP5/MOD3.2

Fig. 3.5 Input data for CONV32 (RELAP5) (6/8)


```

RK010T - ADDED FOR RELAP5/M003.2
RK01PS - ADDED FOR RELAP5/M003.2
RK01ZD - ADDED FOR RELAP5/M003.2
RK01ZU - ADDED FOR RELAP5/M003.2
RK0KIT - ADDED FOR RELAP5/M003.2
RK0MCR - ADDED FOR RELAP5/M003.2
RK0MND - ADDED FOR RELAP5/M003.2
RK0NCF - ADDED FOR RELAP5/M003.2
RK0NGT - ADDED FOR RELAP5/M003.2
RK0NHR - ADDED FOR RELAP5/M003.2
RK0NNG - ADDED FOR RELAP5/M003.2
RK0NNN - ADDED FOR RELAP5/M003.2
RK0NNH - ADDED FOR RELAP5/M003.2
RK0NNX - ADDED FOR RELAP5/M003.2
RK0NNY - ADDED FOR RELAP5/M003.2
RK0NNZ - ADDED FOR RELAP5/M003.2
RK0NR - ADDED FOR RELAP5/M003.2
RK0NRH - ADDED FOR RELAP5/M003.2
RK0NTH - ADDED FOR RELAP5/M003.2
RK0NVR - ADDED FOR RELAP5/M003.2
RK0NXY - ADDED FOR RELAP5/M003.2
RK0NZ - ADDED FOR RELAP5/M003.2
RK0NZF - ADDED FOR RELAP5/M003.2
RK0OPT - ADDED FOR RELAP5/M003.2
RK0RCL - ADDED FOR RELAP5/M003.2
RK0RF - ADDED FOR RELAP5/M003.2
RK0RID - ADDED FOR RELAP5/M003.2
RK0RIF - ADDED FOR RELAP5/M003.2
RK0SN - ADDED FOR RELAP5/M003.2
RK0SYH - ADDED FOR RELAP5/M003.2
RK0SYN - ADDED FOR RELAP5/M003.2
RK0ZFG - ADDED FOR RELAP5/M003.2
RK0ZFI - ADDED FOR RELAP5/M003.2
RK0ZID - ADDED FOR RELAP5/M003.2
RK0ZTR - ADDED FOR RELAP5/M003.2
RKVOLN - ADDED FOR RELAP5/M003.2
RTBLN - ADDED FOR RELAP5/M003.2
RTBLS - ADDED FOR RELAP5/M003.2
SETNO - ADDED (RADHTC )
SELAG - ADDED FOR RELAP5/M003.2
SHDRH - ADDED FOR RELAP5/M003.2
SHDRHO - ADDED FOR RELAP5/M003.2
SHDRH - ADDED FOR RELAP5/M003.2
SHGRHO - ADDED FOR RELAP5/M003.2
SRGJUN - ADDED FOR RELAP5/M003.2
SRGVOL - ADDED FOR RELAP5/M003.2
STC0F1 - ADDED FOR RELAP5/M003.2
STC0F2 - ADDED FOR RELAP5/M003.2
STH2AV - ADDED FOR RELAP5/M003.2
STJCK1 - ADDED FOR RELAP5/M003.2
STJCK2 - ADDED FOR RELAP5/M003.2
STJPK1 - ADDED FOR RELAP5/M003.2
STJPK2 - ADDED FOR RELAP5/M003.2
STLTE1

```

Fig. 3.5 Input data for CONV32 (RELAP5) (7/8)

```

STLTE2 - ADDED FOR RELAP5/M003.2
STRAP1 - ADDED FOR RELAP5/M003.2
STRAP2 - ADDED FOR RELAP5/M003.2
STRCL1 - ADDED FOR RELAP5/M003.2
STRCL2 - ADDED FOR RELAP5/M003.2
STROCI - ADDED FOR RELAP5/M003.2
STROCC - ADDED FOR RELAP5/M003.2
STROP1 - ADDED FOR RELAP5/M003.2
STROP2 - ADDED FOR RELAP5/M003.2
STREX1 - ADDED FOR RELAP5/M003.2
STREX2 - ADDED FOR RELAP5/M003.2
STRGEO - CHANGED TO STRGEO FROM STRGEO
STRPE1 - ADDED FOR RELAP5/M003.2
STRPE2 - ADDED FOR RELAP5/M003.2
STRTE1 - ADDED FOR RELAP5/M003.2
STRTE2 - ADDED FOR RELAP5/M003.2
STRXL1 - ADDED FOR RELAP5/M003.2
STRXL2 - ADDED FOR RELAP5/M003.2
STSATP - ADDED FOR RELAP5/M003.2
STSCL1 - ADDED FOR RELAP5/M003.2
STSCL2 - ADDED FOR RELAP5/M003.2
STSJPT - ADDED FOR RELAP5/M003.2
STRREQ - ADDED FOR RELAP5/M003.2
STWPK1 - ADDED FOR RELAP5/M003.2
STWPK2 - ADDED FOR RELAP5/M003.2
SYSMAT - ADDED FOR RELAP5/M003.2
SYSOPT - ADDED FOR RELAP5/M003.2
SYSVOL - ADDED FOR RELAP5/M003.2
TBLNUM - ADDED FOR RELAP5/M003.2
TSPCTR - ADDED FOR RELAP5/M003.2
TSPFAC - ADDED FOR RELAP5/M003.2
TSPPMJ - ADDED FOR RELAP5/M003.2
TSPPRS - ADDED FOR RELAP5/M003.2
TTER - ADDED FOR RELAP5/M003.2.1.2
TURCTR - ADDED FOR RELAP5/M003.2.1.2
TURSCH - ADDED FOR RELAP5/M003.2.1.2
TURUPJ - ADDED FOR RELAP5/M003.2.1.2
VCMFND - ADDED FOR RELAP5/M003.2.1.2
VCTRL - ADDED FOR RELAP5/M003.2.1.2
VCTRLD - ADDED FOR RELAP5/M003.2.1.2
VCTRLN - ADDED FOR RELAP5/M003.2.1.2
VCTRLS - ADDED FOR RELAP5/M003.2.1.2
VCTRLX - ADDED FOR RELAP5/M003.2.1.2
VLPMDX - ADDED FOR RELAP5/M003.2.1.2
VLVNH - ADDED FOR RELAP5/M003.2.1.2
VLVTRP - ADDED FOR RELAP5/M003.2.1.2
VOLMAT - ADDED FOR RELAP5/M003.2.1.2
VOLNO - ADDED FOR RELAP5/M003.2.1.2

```

Fig. 3.5 Input data for CONV32 (RELAP5) (8/8)

```
*DELETE 111 - 112
*INSERT 112
CSA PS = (PMPRHD(I)/PMPRFL(I))*(ETA*TT*HD + (1.0-ETA)*HT*SLOPE)
PS = (PMPRHD(I)/PMPRFL(I))*(ETA*TT*HD + (1.000-ETA)*HT*SLOPE)
```

Fig. 3.6 Example of converted Update-Card (subroutine PUMP2)

```
PROC 0 IN('J9127.@@UPDTCO.DATA') OUT('J9127.LLL.DATA')
CONTROL LIST MSG
FREEALL
DEL (WWW.W.DATA '&OUT')
ALLOC DA(WWW) NEW T CAT SP(1 1) F(W) UNIT(TSSWK) REU
OPENFILE W OUTPUT
SET W = &OUT
WRITE W
PUTFILE W
CLOSEFILE W
LIB 'J9909.SEAK.LOAD'
ALLOC DA(W.DATA) NEW T CAT SP(1 1) UNIT(TSSWK)
ALLOC DA(W.DATA) F(SYSPRINT) SHR REU
DI '&IN'
ATTR ABC LRECL(80) BLKSIZE(22000) RECFM(F B) DSOrg(P0)
ALLOC DA('&OUT') NEW T CAT SP(50 50) DIR(50) RELEASE +
US(ABC) UNIT(TSSWK)
FREE F(W)
ALLOC F(FT02F001) DA(WWW) REU
ALLOC F(FT07F001) DUMMY
ALLOC F(FT08F001) DA('J9127.R5M3212.MLIST5W') SHR
ALLOC F(FT66F001) DUMMY
ALLOC F(FT77F001) DUMMY
FORT77 R5M32.CONV32.FORT77 ELM(*) GO SSN(D)
EXIT
```

Fig. 3.7 Command procedure for CONV32

変換済旧版(RELAP5/MOD3.2) コード

```
SUBROUTINE PUMP2 (W,H,T,PS,I,IR,IM)
IMPLICIT REAL*8 (A-H,O-Z)
C
C INTERPOLATE PUMP HOMOLOGOUS CURVES.
.
H = PMPRHD(I)*AA*HD
T = PMPRTK(I)*AA*TD
C32 IF (HT .EQ. 0.0 .AND. W .EQ. 0.0) RETURN
IF (HT .EQ. 0.000 .AND. W .EQ. 0.000) RETURN
C CALCULATION OF HEAD VERSUS FLOW SLOPE.
PS = PMPRHD(I) / PMPRFL(I) *
C32 * (ETA * TT * HD + (1.0 - ETA) * HT * SLOPE)
* (ETA * TT * HD + (1.000- ETA) * HT * SLOPE)
300 RETURN
END
```

旧IBM版(RELAP5/MOD3.2) コード

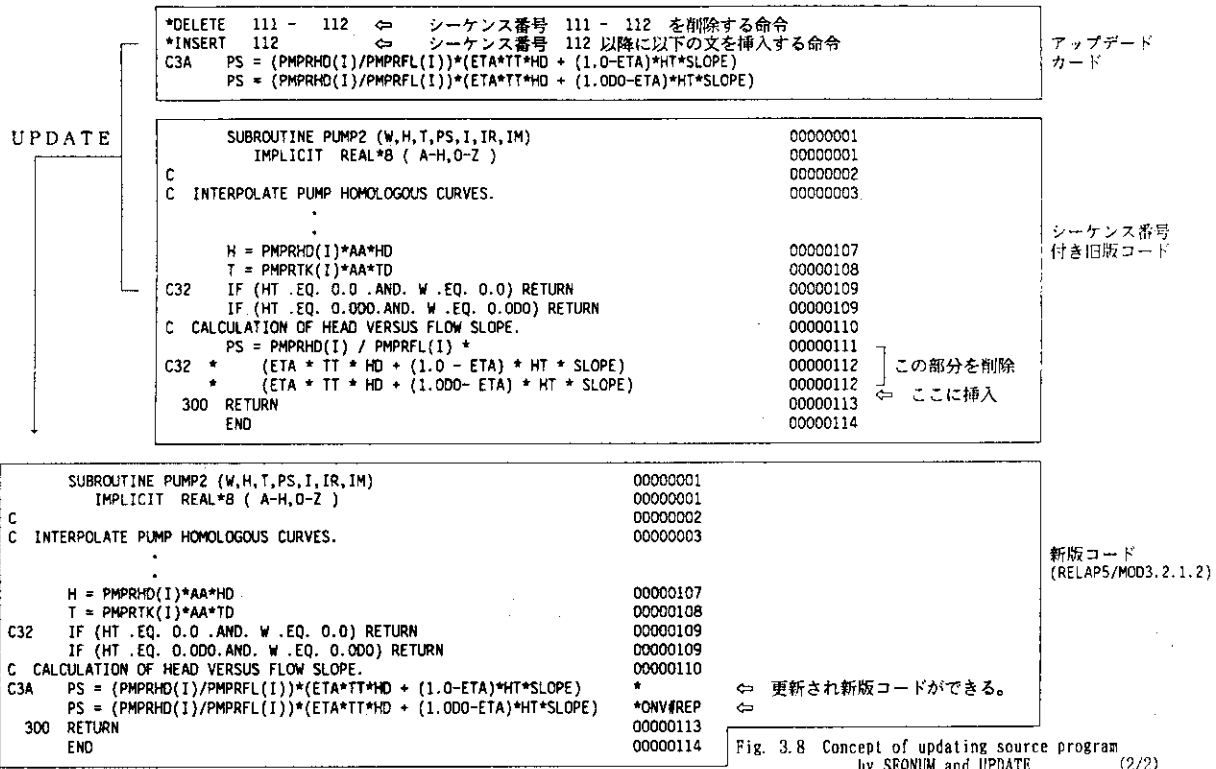
```
SUBROUTINE PUMP2 (W,H,T,PS,I,IR,IM)
C
C INTERPOLATE PUMP HOMOLOGOUS CURVES.
.
H = PMPRHD(I)*AA*HD
T = PMPRTK(I)*AA*TD
IF (HT .EQ. 0.0 .AND. W .EQ. 0.0) RETURN
C CALCULATION OF HEAD VERSUS FLOW SLOPE.
PS = PMPRHD(I) / PMPRFL(I) *
* (ETA * TT * HD + (1.0 - ETA) * HT * SLOPE)
300 RETURN
END
```

SEQNUM

SUBROUTINE PUMP2 (W,H,T,PS,I,IR,IM)	00000001
IMPLICIT REAL*8 (A-H,O-Z)	00000001
C	00000002
C INTERPOLATE PUMP HOMOLOGOUS CURVES.	00000003
.	
H = PMPRHD(I)*AA*HD	00000107
T = PMPRTK(I)*AA*TD	00000108
C32 IF (HT .EQ. 0.0 .AND. W .EQ. 0.0) RETURN	00000109
IF (HT .EQ. 0.000 .AND. W .EQ. 0.000) RETURN	00000109
C CALCULATION OF HEAD VERSUS FLOW SLOPE.	00000110
PS = PMPRHD(I) / PMPRFL(I) *	00000111
C32 * (ETA * TT * HD + (1.0 - ETA) * HT * SLOPE)	00000112
* (ETA * TT * HD + (1.000- ETA) * HT * SLOPE)	00000112
300 RETURN	00000113
END	00000114

シーケンス番号付き
旧版コードができる

Fig. 3.8 Concept of updating source program by SEQNUM and UPDATE (1/2)



```

TWC(2 4 3 0 5) SRP
/** T.2 W.1 I.5 C.2 SRP
/**
/**
/** ***** RELAP5/MOD2/C36 SEQNUMBER PROGRAM *****
/**
/** ***** EXEC PGM=SEQNUM *****
/** DO DSN=J0001.R5T00L.LOAD,DISP=SHR
/**CHKSOC DO DSN=J9127.R5M32.SELAP.S,DISP=SHR,LABEL=(,IN)
/**INSOC DO DSN=J0001.R5M32.RELAP5.FORT77,DISP=SHR,LABEL=(,IN)
/**OUTSOC DO DSN=J9127.R5M32N.RELAP5.FORT77,DISP=(NEW,CATLG),
/** DCB=(LRECL=80,BLKSIZE=22000,RECFM=FB,DSORG=PO),
/** SPACE=(TRK,(200,20,60),RLSE),UNIT=TSSWK
/**SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=137,BLKSIZE=6850)
/**
    
```

Fig. 3.9 JCL for SEQNUM

```

TWC(5 4 4 0 5) SRP
/**
/** ***** RELAP5/MOD2/C36 UPDATE PROGRAM *****
/**
/** ***** EXEC PGM=UPDATE *****
/**STEPLIB DD DSN=J0001.R5T00L.LOAD,DISP=SHR
/**UPDATE DD DSN=J9127.LLL.DATA,DISP=SHR,LABEL=(,IN)
/**OLDSOC DD DSN=J9127.R5M32N.RELAP5.FORT77,DISP=SHR,LABEL=(,IN)
/**NEWSOC DD DSN=J9127.R5M32N.RELAP5.FORT77,DISP=(NEW,CATLG),
/** SPACE=(TRK,(200,20,60),RLSE),UNIT=TSSWK
/**SYSPRINT DD DSN=J9127.UPDATE.OUTPUT,DISP=(NEW,CATLG),
/** DCB=(LRECL=137,BLKSIZE=15344,RECFM=FBA,DSORG=PS),
/** SPACE=(TRK,(10,10),RLSE),UNIT=TSSWK
/**
    
```

Fig. 3.10 JCL for UPDATE

```

SUBROUTINE CHKLEV
  . . .
  ISSYS = FILNDX(30)
  DO M = 1, NLOOPS(ISSYS)
    IXVFF = VCTRLS(LIV(ISSYS))
    IXJFF = VCTRLS(2,LIV(2,ISSYS))
    IXJFF = JCNXS(LIJ(ISSYS))
    IXJFF = JCNXS(2,LIJ(2,ISSYS))
    CALL LEVEL(1)
    ISSYS = ISSYS + LPSKP
  ENDDO
  IF ( IAND(IMOCTL(1), ISHFT(1,1)).NE.0 .AND. SUCCES.EQ.0 ) THEN
    RETURN
  END
  . . .

```

↓

```

SUBROUTINE CHKLEV
  . . .
  ISSYS = FILNDX(30)
  DO M = 1, NLOOPS(ISSYS)
    DO 8000 M = 1, NLOOPS(2,ISSYS)
      IXVFF = VCTRLS(LIV(ISSYS))
      IXJFF = VCTRLS(2,LIV(2,ISSYS))
      IXJFF = JCNXS(LIJ(ISSYS))
      IXJFF = JCNXS(2,LIJ(2,ISSYS))
      CALL LEVEL(1)
      ISSYS = ISSYS + LPSKP
    ENDDO
    8000 CONTINUE
  IF ( IAND(IMOCTL(1), ISHFT(1,1)).NE.0 .AND. SUCCES.EQ.0 ) THEN
    RETURN
  END
  . . .

```

Fig. 3.11 Modification of DO-loop (subroutine CHKLEV)

```

SUBROUTINE GCSUB(GC,DGCDP,HFACT,ERROR)
  . . .
  DGCDP2 = 3.0 * DGCDP / GC
  DGCDP2 = 3.000* DGCDP / GC
  DGCDP1 = MAX(0.25,MIN(1.5,(ONE + 0.5*DGCDP2*DELPL1)))*DGCDP
  C
  DELP = (GT - GC) / (DGCDP1 - DGTDP)
  C
  C SET ITERATION LIMITS.
  C
  C DPMIN = ABS(CONV*GT/DGTDP)
  C
  . . .
  RETURN
  END

```

↓

```

SUBROUTINE GCSUB(GC,DGCDP,HFACT,ERROR)
  . . .
  DGCDP2 = 3.0 * DGCDP / GC
  DGCDP2 = 3.000* DGCDP / GC
  C3AGCDP1 = MAX(0.25,MIN(1.5,(ONE + 0.5*DGCDP2*DELPL1)))*DGCDP
  DGCDP1 =
  1 MAX(0.2500,MIN(1.500,(ONE + 0.500*DGCDP2*DELPL1)))*DGCDP
  C
  DELP = (GT - GC) / (DGCDP1 - DGTDP)
  C
  C SET ITERATION LIMITS.
  C
  C3APMIN = ABS(CONV*GT/DGTDP)
  DPMIN = ABS(CONV*GT/DGTDP)
  C
  . . .
  RETURN
  END

```

Fig. 3.12 Correction of coding mistake

```

TWC(6 4 8 3 5) SRP
// EXEC LMGOEX, LM='J9127.R5M3212B', PMW=RELAP5, A=NDERRPRT
//FT06F001 DD DSN=J9127.EDHTRK3.TEST. OUTLIST, DISP=(NEW,CATLG),
// DCB=(RECFM=FBA, LRECL=137, BLKSIZE=19043, DSORG=PS),
// SPACE=(TRK, (20, 20), RLSE), UNIT=TSSWK
//SYSIN DD DSN=J9127.R5M3212.PO.DATA(EDHTRK).DISP=SHR, LABEL=(, , IN)
//FT17F001 DD DSN=J9127.EDHTRK3.FT17.DATA, DISP=(NEW,CATLG),
// SPACE=(TRK, (100, 50), RLSE), UNIT=TSSWK
//FT19F001 DD DSN=J9127.EDHTRK3.FT19.DATA, DISP=(NEW,CATLG),
// SPACE=(TRK, (100, 50), RLSE), UNIT=TSSWK
//FT20F001 DD DSN=J9127.EDHTRK3.FT20.DATA, DISP=(NEW,CATLG),
// SPACE=(TRK, (100, 50), RLSE), UNIT=TSSWK
//FT23F001 DD DSN=J9127.EDHTRK3.RST.DATA, DISP=(NEW,CATLG),
// SPACE=(TRK, (100, 50), RLSE), UNIT=TSSWK
//FT24F001 DD DSN=J9127.EDHTRK3.F24.DATA, DISP=(NEW,CATLG),
// SPACE=(TRK, (100, 50), RLSE), UNIT=TSSWK
//FT25F001 DD DSN=J9127.EDHTRK3.F25.DATA, DISP=(NEW,CATLG),
// SPACE=(TRK, (100, 50), RLSE), UNIT=TSSWK
//FT51F001 DD DSN=J9127.EDHTRK3.FT41.DATA, DISP=(NEW,CATLG),
// SPACE=(TRK, (100, 50), RLSE), UNIT=TSSWK
//FT66F001 DD SYSOUT=*
//
    
```

Fig. 3.16 JCL for execution (M780)

```

#!/bin/csh -f
#$-eo
#$-q vpps
#$-lt 0:30:00
#$-C RELAP5_M003212
cd $QSUB_WORKDIR
#-----relap5-----
setenv fu23 ../out/typplr.restart.data
setenv fu66 ../out/typplr.f66.data
timev -Hn r5m3212vpps < ../input/TYPPLR && ../out/typplr.outlist
    
```

Fig. 3.17 Shell-script for execution (VPP500)

```

TWC(6 4 8 2 7) SRP
//ASMP EXEC ASM, OBJS='300,100', SO='J0001.R5M3212.RELAP5', Q=''.ASM'
//SYSPRINT DD DUMMY
//FORTEK EXEC FORTEK, SO='J0001.R5M3212.RELAP5', RGN=8M, ERGN=28M,
// A='NOAE,ELM(*)', SSM(O), DISP=MOD
//SYSPRINT DD DSN=J9127.R5M3212.OUTLIST, DISP=(NEW,CATLG),
// DCB=(LRECL=137, BLKSIZE=19043, RECFM=FBA, DSORG=PS),
// SPACE=(TRK, (20, 20), RLSE), UNIT=TSSWK
//SYSINC DD DSN=J0001.R5M3212.RELAP5, INCLUDE, DISP=SHR
//FORTEK EXEC FORTEK, SO='J0001.R5M3212.ENVR', RGN=8M, ERGN=28M,
// A='NOAE,ELM(*)', SSM(O), DISP=MOD
//SYSPRINT DD DSN=J9127.R5M3212.OUTLIST, DISP=MOD
//SYSINC DD DSN=J0001.R5M3212.ENVR, INCLUDE, DISP=SHR
//LINK EXEC LKEXCT, A='MAP', LM='J9127.R5M3212B', UNIT=TSSWK
//SYSIN DD DSN=J0001.R5M3212.LINK.PO.DATA(LINKNOAE), DISP=SHR
//
    
```

Fig. 3.13 JCL for loadmodule making (M780)

```

#!/bin/csh -f
#$-eo
#$-q gspc
#$-lt 1:00:00
#$-C RELAP5_M003212
cd $QSUB_WORKDIR
fcpx -c -O ../src/loc.c >& cmp.log
frtpx -c -W, -sc -Oe, -S -I../src/inc ../src/*.f >>& cmp.log
frtpx -o r5m3212vpps *.o >>& cmp.log
    
```

Fig. 3.14 Shell-script for loadmodule making (VPP500)

```

#!/bin/csh -f
#$-eo
#$-q gspc
#$-lt 1:00:00
#$-C RELAP5_M003212
cd $QSUB_WORKDIR
frtpx -o r5m3212vpps *.o >>& cmp.log
    
```

Fig. 3.15 Shell-script for loadmodule making (Linkage only, VPP500)

4. NJOY94 のインストール及び倍精度化

原子核データ処理システム NJOY[6] の VPP500 へのインストール及び倍精度化の作業を行った。NJOY は種々のコンピュータシステムに対応したコード・パッケージであり、本作業ではまずこのソース・コードを SUN ワークステーション上に生成した。そしてこれを VPP500 に転送し、インストールした。次いで VPP500 上で倍精度化を実施した。また、ここで対象にした NJOY のバージョンは NJOY94.25 である。以下に、これらの作業内容について報告する。

4.1 NJOY94.25 の生成

NJOY94.25 は、基本的には NJOY94.0 のソースプログラム・ファイル(src)にバージョン 25 までの修正パッチ (マシン共通, マシン依存) を充てる形で生成される。こうしたバージョンの更新は、これらと一緒に配付されたバージョン・コントロール・プログラム(upd)により実現される。SUN ワークステーション上での NJOY94.25 生成イメージを Fig. 4.1 に示す。また、本生成処理で得たモジュールの一覧を Table 4.1 に示す。

生成処理の妥当性を確認するため、配布ファイルのテストケース 1~4(入力: in1~in4)を SUN 上で実施した。各ケースのランニング結果と配布された結果(out1~out4)を diff コマンドにより比較したところ、出力行数、構成は一致したが計算値に若干の差異を生じていた。これは、SUN 上のランニングではいずれもアンダーフローが発生しており、計算環境の違いによるものと考え生成処理は妥当と判断した。

4.2 VPP500 へのインストール

「4.1 NJOY94.25 の生成」で生成した 23 モジュール (詳細は Table 4.1 参照) のソースプログラム・ファイル(x...x.f)を VPP500 に転送し、まず単精度レベルのインストールを行った。ここでは、SUN - VPP500 のマシン差に起因する不具合を修正することが狙いである。コンパイル、リンク、及びテストケース 1 の実行を通して、検出できた不具合の原因は以下の 4 種類であった。

- (1) 動作マシン名 ('sun' → 'vpp')
- (2) 日付・時間に関するルーチン (etime → clock, idate → date, fdate → time)
- (3) 8 進表示 (o 形編集) (変換処理の組込み)
- (4) common ブロックサイズの食い違い (ダミー領域の挿入)

インストールの妥当性を確認するため、「4.1 NJOY94.25 の生成」の場合と同様に、配布されたテストケース 1~4 を実行し、各ケースのランニング結果と配布された結果(out1~out4)を diff コマンドにより比較した。SUN の場合と同様、出力行数、構成は一致したが、計算値に若干の差異を生じており、テストケース 3, 4 ではアンダーフローが生じていた。cpu, コンパイラ等の計算環境の違いによるものと考えられ、不具合が生じているとは言い難い。コンパイル、リンク、コンパイル・リンク、及びテストケース実行のシェルスクリプトを Fig. 4.2~4.5 に示す。

4.3 VPP500 上での倍精度化

NJOY94.25 には一部で `real*8` で倍精度データを扱っているルーチンもあるが、全体としては倍精度化されていない。また、共通ブロックの利用に関して、同じ領域に対して異なる型で共用する使い方を多数利用している。このため倍精度化にあたっては、精度拡張機能だけでなく領域を共有するデータ間で記憶域を合わせる機能をも兼ね備えたコンパイラ・オプション (`-Ap`) を選択した。その他、最適化、ベクトル化のオプションは、計算結果への影響をできるだけ抑えるため、`-Ob,S -Wv,-sc` のオプションを選択した。コンパイル、リンク、及びコンパイル・リンクのシェルスクリプトを Fig. 4.6 ~ 4.8 に示す。

4.3.1 作業内容

作業は配布されたテストケース 1 ~ 8 を実行して行くなかで、倍精度化に伴う不具合を検出し修正するという方法で進めた。以下に検出した不具合とその修正について述べる。

NJOY で用いられているデータの型は、文字型、論理型を除くと、4 バイトの整数型、4 バイトの実数型 (以下、実数型 1 と呼ぶ)、及び `real*8` で定義された 8 バイトの実数型 (以下、実数型 2 と呼ぶ) の 3 種類である。単精度から倍精度への変更に伴い大きく変わるのは、実数型 1 の記憶域の大きさである (4 バイトから 8 バイトに拡張される)。また、`-Ap` の記憶域合わせの機能により以下の整数型データに対して、8 バイトの領域が確保されることも大きく変わる場所である。但し、整数型データとしての記憶域は前半 4 バイトであり、後半 4 バイトはダミーとなる。

- (1) 実数型 1、実数型 2 のデータと `equivalence` により記憶域を共有する整数型データ
- (2) 共通ブロックに属する整数型データ
- (3) 引数として使用されている整数型データ

これらの影響は、各データを単独で扱う (参照、更新、代入、演算、等) 場合には、コンパイラで吸収できる。しかし他の変数・配列との関係で、データ長と記憶域の大きさの対応関係 (等しい、倍、等) を陽に意識して処理ロジックを組み立てているところでは、コンパイラで吸収することはできない。ここに不具合が発生する。つまり、以下のようなデータ長と記憶域の大きさの対応関係がずれる所に倍精度、及び記憶域合わせに伴う不具合が発生する。この対応関係を Fig. 4.9 に示す。

- (1) 整数型のデータ長 - 対応する領域の大きさ関係のずれる所
- (2) 整数型 - 実数型 1 のデータ長関係のずれる所
- (3) 実数型 1 - 実数型 2 のデータ長関係のずれる所

4.3.1.1 整数型のデータ長 - 対応する領域の大きさ関係のずれに伴う不具合

この不具合は整数型データに対する書式なし入力 - 出力において検出された。この入力 - 出力は同一の整数型データに対するものであったが、入力並びの長さと出力並びの長さが異なってし

まい、この不具合が発生した。

テストケース1のreconrモジュールにおいて、tape21ファイルのレコード群(mat:1306, mf:1, mt:451)の先頭レコードにポジショニングするためレコードの検索を行うfindfルーチンをコールしたところ、以下のエラーメッセージを出力して異常終了した。ここでtape21ファイルは、直前のmoderモジュールにおいて書式なしwriteで作成されたバイナリ・ファイルである。

```
***error in findf*** mat1306 mf 1 mt451 not on tape 21
```

異常終了の原因は、書式なしread/writeの仕様が同じ整数型データであっても、記憶域合わせされたものに対するものとそうでないローカルなものに対するものとで異なるところにある。記憶域合わせされた整数型データに対しては後半のダミーの4バイトを合わせて8バイトとしてread/writeの対象にしており、ローカルな整数型データに対しては記憶域の4バイトだけを対象にしている。実際、tape21ファイルのmat, mf, mtに対する入出力では、書込み時は記憶域合わせされた共通ブロック内の整数型変数を利用し、読み込み時はfindfルーチン内で定義されたローカルな整数型変数を利用している。つまり、前者ではダミーデータを含む8バイトで書き出し、後者ではこれを4バイト分ずつ読み出す結果となってしまった。したがって、(mat, mf, mt)のデータはtape21ファイルを中継することにより、書込み前と読み込み後で対応関係にずれが生じることになり、対応レコードを検出できなかった。この対応関係をFig. 4.10に示す。

この種の不具合に対処するため全モジュールの書式なしread文、write文の洗い出しを行い、入力並び、及び出力並びの中に出てくる整数型データを調査した。結果として、共通ブロックに属するものが圧倒的に多かったので、共通ブロックに属さずサブルーチンの引数に使われていないローカルな整数型データに対し以下の対処を行った。

- (1) 実数型1のデータとの記憶域合わせ(例えば、equivalence(rvpp2,nw2))
- (2) 記憶域合わせをした整数型配列への置き換え

4.3.1.2 整数型-実数型1のデータ長関係のずれに伴う不具合

(1) 引数の型の相違に伴うアドレス境界指定例外の発生

テストケース3のviewrモジュールにおいてtape36ファイルの第3レコードを読み込む際、infreeサブルーチンの中でアドレス境界指定例外(jwe0019i/SIGBUS/BUS _ ADRLN)が発生した。この異常終了の原因は、サブルーチン側の仮引数を実数型配列であるのに対し、call元の実引数は整数型配列であったがこの整数型配列が8バイト境界に調整されなかったことにある。記憶域合わせの機能では、8バイト境界の調整まではされない。つまり、実数型、及び整数型データのアドレス境界が、単精度の場合共に4バイトであったものが倍精度化に伴いそれぞれ8バイト、4バイトと異なってしまったが、実引数側の整数型配列は8バイト境界に調整されなかったためアドレス境界指定例外が発生した。

この種の不具合に対処するため、M780上のツールであるSTREAM77[3]により引数チェックを行い、実引数と仮引数の型がそれぞれ整数型、実数型となっている箇所の洗い出しを行った。こうした箇所の中で8バイト境界が保証できないところでは、equivalence文により実引数と実数型変数との領域共有を行い8バイト境界化を図った。また、更にSTREAM77の出力結果をチェックし引数の型が一致しない箇所を検討したが、特に問題となる箇所は検出できなかった。

(2) 共通ブロック内データの対応ずれ

テストケース7のacerモジュールにおいて、aceprtルーチンが出力するインフォメーションブロックの各項目の並びが以下のように一つずつずれて、そして間に0が挿入されたような形で出力された。

a, b, c, . . . → a, 0, b, 0, c, 0 . . .

インフォメーションブロックの出力並びには整数型配列(nxs(i), i=3,6), (jxs(i), i=1,22)の各要素が含まれており、この配列はequivalence文により共通ブロックnxst, jxstの各項目と領域共有が図られている。このequivalence文による対応関係が、倍精度化、及び記憶域合わせによりずれてしまったところにこの障害の原因がある。共通ブロックnxst, jxstの各項目はその記憶域サイズが共に8バイトになっているが、それに対応させる配列nxs(i), jxs(i)はaceprtルーチン内で定義されたローカルな整数型配列であるため、各記憶域サイズが4バイトである。従って、対応のずれが生じる結果となった。この対応関係をFig. 4.11に示す。こうした対応のずれを調整するため、配列nxs(i), jxs(i)の各要素の記憶域サイズを8バイトに拡張した。拡張は、例えば以下のように、equivalence文により実数型配列と領域共有化することにより実現した。

```
dimension      vnxs(16),vjxs(32)
equivalence    (vnxs(1),nxs(1)),(vjxs(1),jxs(1))
```

同じような対応ずれが他にないか調査するため、equivalence文の洗い出しを行い同様の対処を行った。結果として、acerモジュールには他に5箇所あった。

4.3.1.3 実数型1 - 実数型2のデータ長関係のずれに伴う不具合

(1) コンテナ配列内の領域共有関係の不具合

テストケース2のccccrモジュールにおいて、以下のメッセージを出力して異常終了した。異常終了箇所はreservサブルーチンであり、コンテナ配列a(i)の中にデータセットscrtのデータ域を確保する処理であった。

```
***error in reserv****storage exceeded.
      need ***** more words for id scrt.
```

コンテナ配列は入出力バッファ等のため、必要に応じて必要なサイズのデータ域を確保・解放して利用する作業域である。この配列内では、次のデータセット域を管理するデータとして、カレントデータ域ポインタ、及びそのデータ域サイズを持っている。この管理データが破壊されていたため、新たなデータ域を確保しようとした時、上記メッセージを出力して領域オーバで異常終了した。コンテナ配列の構成を Fig. 4.12 に示す。

このデータ破壊の原因は、コンテナ配列(実数型配列)内に8バイトデータを格納する際連続する2つの配列要素を割り当てているが、倍精度化に伴いこの対応関係がずれたことによる。実際 $real*8$ の配列 ha を equivalence 文によりコンテナ配列 a に対応付けて、この ha に8バイトのデータを代入しているが、 ha の格納位置 $l1h$ と a の格納位置 $(l1, l1+1)$ との対応は、(4.1)式により行っている。

$$l1h = (l1 - 1)/2 + 1 \quad (l1: \text{奇数}) \quad (4.1)$$

従って、 $a(l1)$ に8バイトデータを格納すべきところ、倍精度化に伴い $a((l1-1)/2+1)$ に格納してしまい、この位置のデータを破壊してしまった。概念的な関係を Fig. 4.13 に示す。このような対応ずれに対処するため、本来格納すべき位置に8バイトデータが格納できるように(4.2)式の処置を行った。

$$l1h = (l1 - 1)/2 + 1 \quad \Rightarrow \quad l1h = l1 \quad (4.2)$$

また、同じような不具合を洗い出すため $real*8$ を含む全サブルーチンを抽出し、調査・修正を行った。

(2) サブルーチンの入力バッファ域における領域共有関係の不具合

(1)と同じテストの中で、ccccr モジュールの出力結果に非キャラクタデータが出力されていた。本来、input ファイルからキャラクタデータ(第2, 3, 5, 7レコード)を読み込んでそのままの形で output ファイルに書き出すのであるが、非キャラクタデータを出力していた。障害の原因は、infree サブルーチンでキャラクタデータを読み込む場合、まず入力バッファの配列 $z(i)$ に読み込み、次いでこの配列と領域共有する $real*8$ の配列 $hz(i)$ により8バイトデータとして扱っているが、倍精度化に伴いこの対応関係がずれたことによる。Fig. 4.14 にこの対応関係を示す。

このような対応ずれに対処するため、Fig. 4.14 の $z(1)$ と $z(2)$ に対応するデータを chz にまとめるサブルーチン $vpchz(chz, z)$ を作成した。ここで、 chz は $real*8$ の変数、 z は $z(1)$, $z(2)$ から成る実数型配列である。

また同じような不具合を洗い出すため、(1)の場合と同様、 $real*8$ を含む全サブルーチンを抽出し、調査・修正を行った。

4.3.2 動作確認

4.3.1 の修正後、再度配布されたテストケース 1～8 を実施し、内容確認を行った。確認は、配布された出力結果ファイルと本テストで得た出力結果ファイルとを比較・検討する方法で行った。基本的には diff コマンドによる比較を行い、図等のポストスクリプトファイルはプリンタ出力して比較を行った。配布された出力結果ファイルは、VPP500 とは異なる異機種で計算機で作成されたものであり、処理ロジックが同じであっても、VPP500 のものに一致するとは限らない。また、配布データについては計算経過の検証ができないため、VPP500 との差異調査ができない。このため比較・検討に際しては、単精度における出力結果ファイル、及び SUN 上での出力結果ファイルも参考にした。つまり、単精度版、SUN 版のコードは倍精度版のオリジナルであるため、倍精度版の処理結果の妥当性については、単精度版、SUN 版のものとの近似性を考慮して判断した。テストケース 1～8 における入出力ファイルの対応関係を Table 4.2 に示す。またテストケース 1～8 の実行シェルスクリプトを Fig. 4.15～4.22 に示す。

4.3.2.1 テストケース 1

< out1 の比較 >

diff コマンドにより比較したところ異なるところは以下の 3 点であった。

(1) reconr モジュールで出力された以下のメッセージ中のカウンタ値

points added by linearization = 127 (配布データ)

points added by linearization = 128 (テスト結果)

(2) broadr モジュールで出力された以下のメッセージ中のカウンタ値

points in = 902 points out = 824 (配布データ)

points in = 901 points out = 824 (テスト結果)

(3) 計算数値の異なる箇所は 5 行 (16 項目)。大半は最終桁の相違

(1), (2) の結果については、VPP500 上の単精度版の結果も SUN 版の結果も倍精度版のテスト結果に一致していた。(1), (2) のような NJOY のカウント値は、計算過程の中である条件を満足するものの個数を示している。例えば、(2) の points in 値は、条件式 `abs(eg).ge.er` において偽となる組 (eg, er) の個数である。本テストの reconr モジュールにおいて、この値が倍精度版と単精度版の場合で 1 異なる結果を得ていた。カウント値の差に影響した eg, er の値は以下の通りであり、このデータにより倍精度版の方が 1 大きくなった。

倍精度: `eg=9.999999999999998e-06 er=1.0e-05`

単精度: `eg=9.99999975e-06 er=9.99999975e-06`

計算精度が異なることにより、こうしたカウント値は異なった値を示すことになる。従って、単精度版、SUN 版の値との近似性を考慮して、(1), (2) の差異は計算環境によるものと判断した。(3) については最終桁の相違であり同様に計算環境によるものと判断した。

< pend1 の比較 >

diff コマンドにより比較したところ、数値的には概ね一致していたが、出力構成の面で異なる

箇所が3箇所あった。倍精度のテスト結果を単精度のものと比較したところ構成面では一行のズレもなく一致した。従って、out1の場合と同様、構成面の差異は計算環境により生じたものと判断した。

4.3.2.2 テストケース2

< out2 の比較 >

diff コマンドにより比較したところ、出力行数、構成は一致した。また、数値的な差異のほとんどは最終桁が1異なるものであったが、reconr, broadr モジュールで出力されたカウンタ値にテストケース1の場合と同様、差異が見られた。この点について単精度版、SUN版の結果を参照したが、四者共それぞれ少しずつ異なった値を示しており、計算環境の差によるものと判断した。

4.3.2.3 テストケース3

< out3 の比較 >

diff コマンドにより比較したところ、計算数値の異なる箇所は5行(5項目)のみであり、その差は最終桁が1異なるのみであった。計算環境の差によるものと判断した。

< plot3 の比較 >

plot3 ファイルはポストスクリプトファイルであるため、配布データ、倍精度のテスト結果共にプリンタに出力して目視確認を行ったところ、出力図は一致した。

4.3.2.4 テストケース4

< out4 の比較 >

本テスト実行中に無効フローティングオペレーション例外(jwe0292i-e)が発生していたが、diff コマンドによる結果比較では25行の差異しか認められず、最終桁が1異なるだけであった。無効フローティングオペレーション例外は、groupr モジュールのconver サブルーチンにおける以下の文で発生していた。

```
izan = nint(c11)                                up11.120
(ここで、c11 = 2.22097e+16 )
```

このc11の値は、入力ファイルt511のmat:1395, mf:8, mt:457の時の値をtape21を経由してa(1)に読み込んだものであり、データ作成時の誤りである。倍精度版に誤りがあるとは言えない。

4.3.2.5 テストケース5

< out5 の比較 >

本テストはmoder-moder-error-covr-viewr モジュールの5ステップから構成されているが、viewr モジュールの出力部分の行数が配布データのものと比較して異なるため、moder ~ covr モジュールからの出力部分とviewr モジュールからの出力部分とに分けて比較した。

(1) moder ~ covr モジュールからの出力部分

diff コマンドによる結果比較では、23,586 行のうち 50 行に差異がみられ、その差は最終桁が 1 異なるだけであった。計算環境の差によるものと判断した。

(2) viewr モジュールからの出力部分

この部分にはプロットの情報が出力されている。その個数・行数が以下のように異なっており、20 個分のプロット情報が倍精度側に追加されていた。

```
配布データ      : 1,855 個 / 39,486 行
倍精度のテスト結果: 1,875 個 / 39,866 行
```

単精度版、SUN 版の結果はそれぞれ 1,856 個、1,861 個と四者共異なる値を示しており、また以下の plot5 の比較でも判るように、目視レベルでは見分けられないほど小さな差異でもあるため、計算環境の差が現れているものと判断できる。

< plot5 の比較 >

plot5 ファイルはポストスクリプトファイルであるため、配布データ、倍精度のテスト結果共にプリンタに出力して目視確認を行ったところ、出力図は一致した。

4.3.2.6 テストケース 6

< plot6 の比較 >

plot6 ファイルはポストスクリプトファイルであるため、配布データ、倍精度のテスト結果共にプリンタに出力して目視確認を行ったところ、出力図は一致した。out6 のデータは配布されていないため、参考に単精度版の結果と比較した。diff コマンドによる比較では、時刻・時間を除いて一致した。

4.3.2.7 テストケース 7

< out7 の比較 >

テストケース 4 の場合と同様、無効フローティングオペレーション例外 (jwe0292i-e) が発生していた。原因はテストケース 4 と全く同じ (データ、場所、ロジック) であった。また、アンダフローが 7 件発生していた。このような条件のもと、diff コマンドによる比較を行った。異なるところは以下の 2 点であった。

(1) reconr モジュールで出力された以下のメッセージ中のカウンタ値

```
points removed by backthinning = 118 (配布データ)
points removed by backthinning = 119 (テスト結果)
```

(2) acer モジュールで出力された計算数値

(1) の結果については、単精度版の結果も、SUN 版の結果も倍精度のテスト結果に一致しており、計算環境によるものと思われる。(2) については約 1 万行に差異が見られるが、大多数は

最終桁の差である。また、参考として、単精度版、SUN版の結果とも比較したが、いずれも数値的には倍精度の結果に近かった。計算環境の差と思われる。

アンダフローは broadr モジュールの hunky, funky の 2 サブルーチン、及び acer モジュールの ptleg サブルーチンで発生した。発生メカニズムは 2 つある。1 つは指数演算 $\exp(-a*a + \dots)$ の値を使った演算の中で発生している。 $a = 26.4$ で 10^{-308} より小さくなる。もう 1 つは、演算誤差が関係する。つまり、以下の演算で値が 0 とならず、 $6.17 \dots \times 10^{-16}$ となり、この値を 21 乗 (do ループで 20 回) することにより 10^{-308} より小さい値となる。倍精度版に誤りがあるとは言えない。

$$a(iamu+20) = -1.0 + ((-0.02) + \dots + (-0.02))$$

50 個

< ace7 の比較 >

出力行数、構成は一致したが、計算数値の差異が約 8 千行に見られた。しかし大多数は最終桁の差であり、計算環境の差によるものと判断した。

4.3.2.8 テストケース 8

< out8 の比較 >

本テストは moder-reconr-broadr-heatr-groupr-acer モジュールの 6 ステップから構成されているが、acer モジュールの出力部分の行数が配布データと倍精度の結果とで大きく異なるため、moder ~ groupr モジュールからの出力部分と acer モジュールからの出力部分とに分けて比較した。

(1) moder ~ groupr モジュールからの出力部分

diff コマンドによる結果比較では 303 行に差異がみられた。broadr モジュールで出力されたカウンタ値と計算数値の行に分けられる。カウンタ値のほうは単精度版、SUN 版の値と比較したところ、単精度版、SUN 版の値は一致していたが、倍精度版、配布データのものとは異なっていた。例えば、倍精度版、配布データ、単精度版の順に 4236, 4237, 4235 であった。計算環境の差が現われているものと判断した。計算数値のほうは大半が最終桁の差異であり、計算環境の差が現われているものと判断した。

(2) acer モジュールからの出力部分

配布データと倍精度版の結果との大きな差は、acer モジュールの aceprt サブルーチンから出力されたインフォメーションブロックの値が異なること、及び倍精度版の reaction 情報に damage の欄が追加されていることである。

インフォメーションブロックには、zaid, awr, len2, nes, ... 等の 33 項目が含まれており、うち 22 項目に差がみられた。単精度版と SUN 版の値は一致していたが、倍精度版、配布データの値とは異なっていた。三者の値を比較すると、単精度の値は配布データよりは倍精度の値に近い値を示している。例えば、倍精度版、配布データ、単精度版の順に 87938, 83703,

87931であり、計算環境の差が現われているものと判断した。一方、damageの欄には、対応するエネルギーテーブルが展開されており、この分が配布データと倍精度の結果データとの行数の差になっている。またdamageの欄は単精度版、SUN版の結果にも出力されており、数値的には概ね一致していた。従って、計算環境の差が現われているものと考えられる。

以上、テストケース1～8においては、倍精度版、単精度版、SUN版の結果、及び配布データの四者の間に種々の違いを生じていた。それらの差異要因は、計算環境、精度の違いによるものであり、倍精度化に伴う変更に起因するものは見い出せなかった。

4.4 まとめ

本作業では、倍精度化をコンパイラオプション-Ap (データの精度拡張及び記憶域合わせ)により対応した。コンパイラ機能により、演算、代入等の操作は保証されるため、対象となるデータを対応する変数あるいは配列要素の間で正しく受け渡すことができれば、この計算システムの処理ロジックは保証される。倍精度コンパイルによって、この受け渡し関係が崩れるとしたら以下の場所に限られる。

- (1)read/write インタフェース
- (2)common 文による領域共有インタフェース
- (3)equivalence 文による領域共有インタフェース
- (4)サブルーチンによる引数の受け渡しインタフェース

read/write インタフェースにおける問題点は、整数型のデータに関して、記憶域合わせされたデータとそうでないデータに対する扱いが異なるところにある。これは記憶域合わせの機能により生じてくるものであり実数型の場合には起こらない。書式なしread/writeインタフェースの扱いに限られてくる。従って4.3.1.1の対応により解決が図られている。common文による領域共有インタフェースにおいては、逆にこの記憶域合わせの機能により各データと共有領域との対応関係が保たれる。equivalence文による領域共有インタフェースの問題点は、記憶域合わせされた整数型データ、及び実数型データの記憶域が4バイトから8バイトに拡張されたことにより、領域の対応ずれが生じたところにある。NJOYで使われているequivalence文の中で問題になるところは、以下の二点であり、4.3.1.2(2)、4.3.1.3(1)、及び4.3.1.3(2)でこの問題をカバーしている。

- (1)common ブロック内の整数型、及び実数型データの並びを記憶域合わせされていない整数型配列で領域共有化している箇所
- (2)実数型配列 a と real*8 の配列 h との間で以下の関係により領域共有化している箇所 $[b(i)] = [a(2i-1) : a(2i)] \quad (i = 1, 2, \dots)$

サブルーチンによる引数の受け渡しインタフェースにおいては、記憶域合わせの機能により型が異なる場合でも引数間のデータの対応関係が保たれるが、4.3.1.2(1)のようなアドレス境界の問題が残る。NJOYにおいて引数間でアドレス境界の問題が起きる可能性があるのは、実引数：整数型、仮引数：実数型の場合だけであり、4.3.1.2(1)でカバーしている。

以上から、倍精度化に伴う修正は網羅できたと考えているが、全体を通して確認テストの不足を感じている。配布されたテストケース1～8はサンプルケースであり、カバーする範囲に限りがある。この不足を少しでも補うため、ユーザに実際のデータによる追試験をして頂いた。この試験の中で入力データの不具合を検出したが、幸い、回避する修正プログラムをユーザに提示して頂いた。これを倍精度用に変更しソース修正を行った。

本作業における修正内容を付録 C に示す。

最後に、精度に関してであるが、アンダフローの発生状況により倍精度化の効果が推測できる。精度向上に伴いアンダフローの発生回数が少なくなっており、テストケース4, 7にその現象が顕著に現れている。テストケース1～8のアンダフローの発生状況を Table 4.3 に示す。

Table 4.1 Modules of NJOY94

No.	モジュール名	生成ファイル名	No.	モジュール名	生成ファイル名
1	acer	acer.f	13	moder	moder.f
2	broadr	broadr.f	14	njoy	njoy.o
3	ccccr	ccccr.f	15	plotr	plotr.f
4	covr	covr.f	16	powr	powr.o
5	dtfr	dtfr.f	17	purf	purf.o
6	errorr	errorr.f	18	reconr	reconr.f
7	gantr	gantr.f	19	resxsr	resxsr.f
8	groupr	groupr.f	20	thermr	thermr.f
9	heartr	heartr.f	21	unresr	unresr.f
10	leapr	leapr.f	22	viewr	viewr.f
11	matxsr	matxsr.f	23	winsr	winsr.f
12	mixr	mixr.f	24	xnjoy	xnjoy

(備考) (1) x...x.f : FORTRAN ソースプログラム ファイル
 (2) x...x.o : オブジェクトプログラム ファイル
 (3) xnjoy : NJOYのロードモジュール ファイル

※配布ファイル ... 1994年10月現在
 (7/7/70) ... (7/7/70) ... (7/7/70)
 src ... 91.00 ... (7/7/70) ...
 upd.o ... 91.25 ... (7/7/70) ...
 upd.o ... SUN ... (7/7/70) ...
 makefile ... SUN ... (7/7/70) ...
 install ... SUN ... (7/7/70) ...
 install ... SUN ... (7/7/70) ...

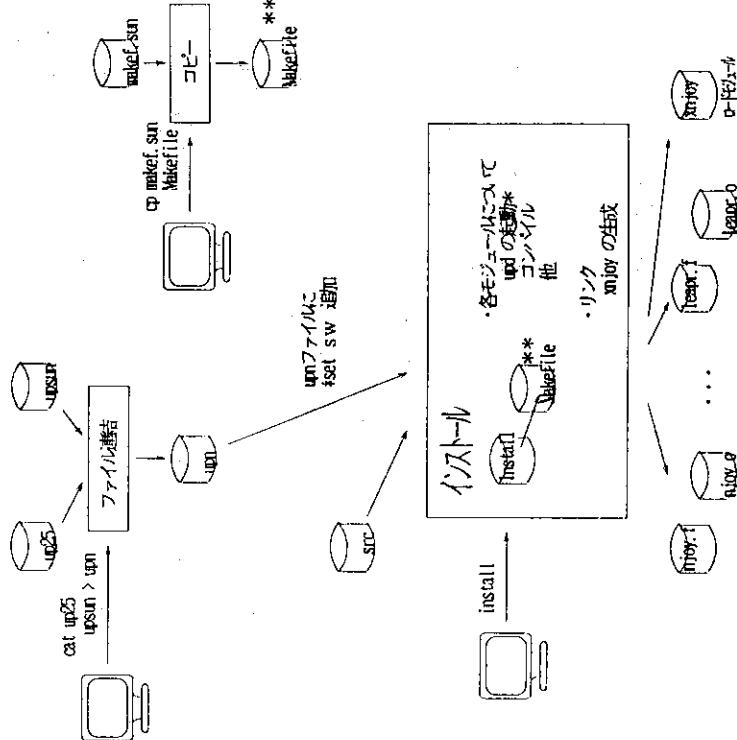


Fig. 4.1 Outline of NJOY94.25 program generation

Table 4.2 Input files for execution of test cases

実行 ケース	実行 ファイル名	入力ファイル								比較対象の 出力結果ファイル
		t511	t322	t404	gam23	gam27	emi61			
1	din1	○								out1, pend1
2	din2			○						out2
3	din3				○	○				out3, plot3
4	din4	○								out4
5	din5	○								out5, plot5
6	din6	○								plot6
7	din7	○								out7, acer7
8	din8							○		out8

Table 4.3 Number of under-flow occurrences

ケース	倍精度 (VPP)		単精度 (VPP)		参考: 単精度 (S/10)
	0	7	0	0	
ケース1	0	0	0	0	7/700-発生
ケース2	0	0	0	1	7/700-発生
ケース3	0	0	8	2	7/700-発生
ケース4	0	0	0	0	0
ケース5	0	0	0	0	0
ケース6	0	0	9	3	7/700-発生
ケース7	0	7	0	0	7/700-発生
ケース8	0	0	0	0	7/700-発生

```
#!/bin/csh -f
#$-q gspc
#$-eo
#$-C CNJOY
cd $QSUB_WORKDIR
frtpr -c -Ob -S -Z comp.NJOY -Wv -sc -Sw *.f
```

Fig. 4.2 Shell script for compilation (single precision)

```
#!/bin/csh -f
#$-q gspc
#$-eo
#$-C LNJOY
cd $QSUB_WORKDIR
frtpr -o xnjoy *.o && link.NJOY
```

Fig. 4.3 Shell script for linkage (single precision)

```
#!/bin/csh -f
#$-q gspc
#$-eo
#$-C CLNJOY
cd $QSUB_WORKDIR
frtpr -o xnjoy -Ob -Z comp.XNJOY -Wv -sc -Sw *.f
```

Fig. 4.4 Shell script for compilation and linkage (single precision)

```
#!/bin/csh -f
#$-eo
#$-q vppm
#$-C ENJOY
cd $QSUB_WORKDIR

echo 'getting endf tape 511'
cp ../1511 tape20
echo 'getting thermal tape 322'
cp ../1322 tape26
echo 'running njoy'
cat>input <<EOF
0
5
#modert
20 -21
#recoart
-21 -22
#endf tape for c-nat from endf/b tape 511*/
1306 3/
.005 0. 7/
#6-c-nat from tape 511*/
#processed by the njoy nuclear data processing system/
#see original endf/b-v tape for details of evaluation/
0/
#broadrt
-22 -23
1306 1/
.005/
300.
0/
.
.
.
#modert
-23 25
#stopt
EOF
../xnjoy -Wl -u <input
echo 'saving output files'
cp output outl
cp tape25 pendl
```

qsub /シケル-ス
0 /ケル-スケル

テストケース
本体

Fig. 4.5 Shell script for execution of test cases

	整数型 integer		実数型 1 real	実数型 2 real*8
	ローカル*1 記号**2	ローカル*1 記号**2		
単精度	4 (4)	4 (4)	4 (4)	8 (8)
倍精度	4 (4)	4 (8**3)	8 (8)	8 (8)

上段：データ長
下段：記憶域
サイズ
(単位：バイト)

- ① 整数型のデータ長に対応する領域の大きさ関係のずれる所
- ② 整数型1-実数型1のデータ長関係のずれる所
- ③ 実数型1-実数型2のデータ長関係のずれる所

*1: 該当ルーチン内で定義されている整数型データ
*2: 上記条件により記憶域合わせされた整数型データ
*3: 記憶域合わせされた整数型データに対応する領域の大きさ

Fig. 4.9 Relation between data length and memory area size

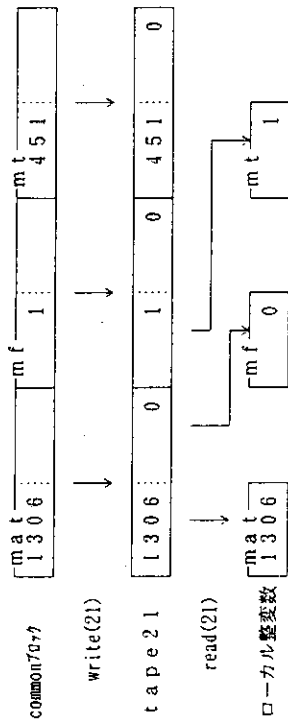


Fig. 4.10 Unformatted read/write interface for integer type data

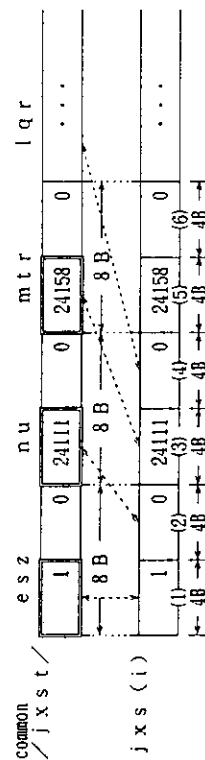


Fig. 4.11 Difference of memory area size of integer type data

```
#!/bin/csh -f
##$-q $spc
##$-eo
##$-C DCLNJOY
cd $QSUB_WORKDIR
frtpr -c -Ap -Ob, -S -Z comp. DNJOY -Wv, -sc -Sw *.f
```

Fig. 4.6 Shell script for compilation (double precision)

```
#!/bin/csh -f
##$-q $spc
##$-eo
##$-C DCLNJOY
cd $QSUB_WORKDIR
frtpr -o xdnjoy *.o & link DNJOY
```

Fig. 4.7 Shell script for linkage (double precision)

```
#!/bin/csh -f
##$-q $spc
##$-eo
##$-C DCLNJOY
cd $QSUB_WORKDIR
frtpr -o xdnjoy -Ap -Ob, -S -Z comp. DNJOY -Wv, -sc -Sw *.f
```

Fig. 4.8 Shell script for compilation and linkage (double precision)

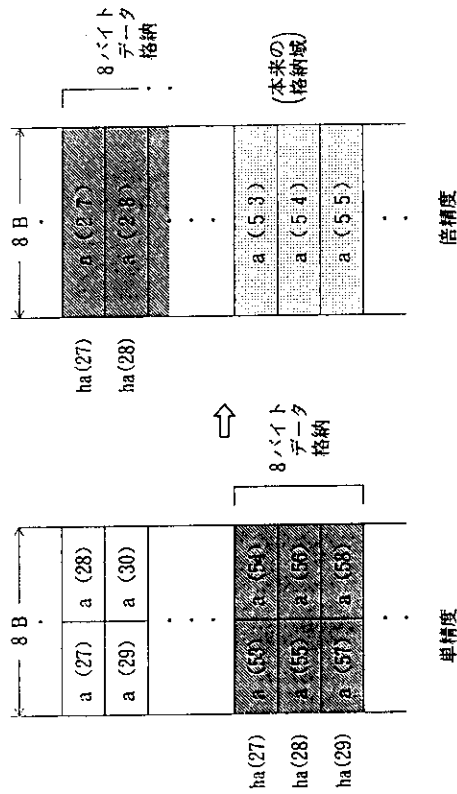


Fig. 4.13 Relation between real type array 'a' and real*8 type array 'ha'

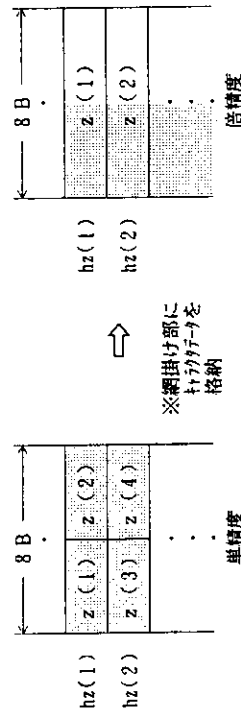


Fig. 4.14 Relation between real type array 'z' and real*8 type array 'hz'

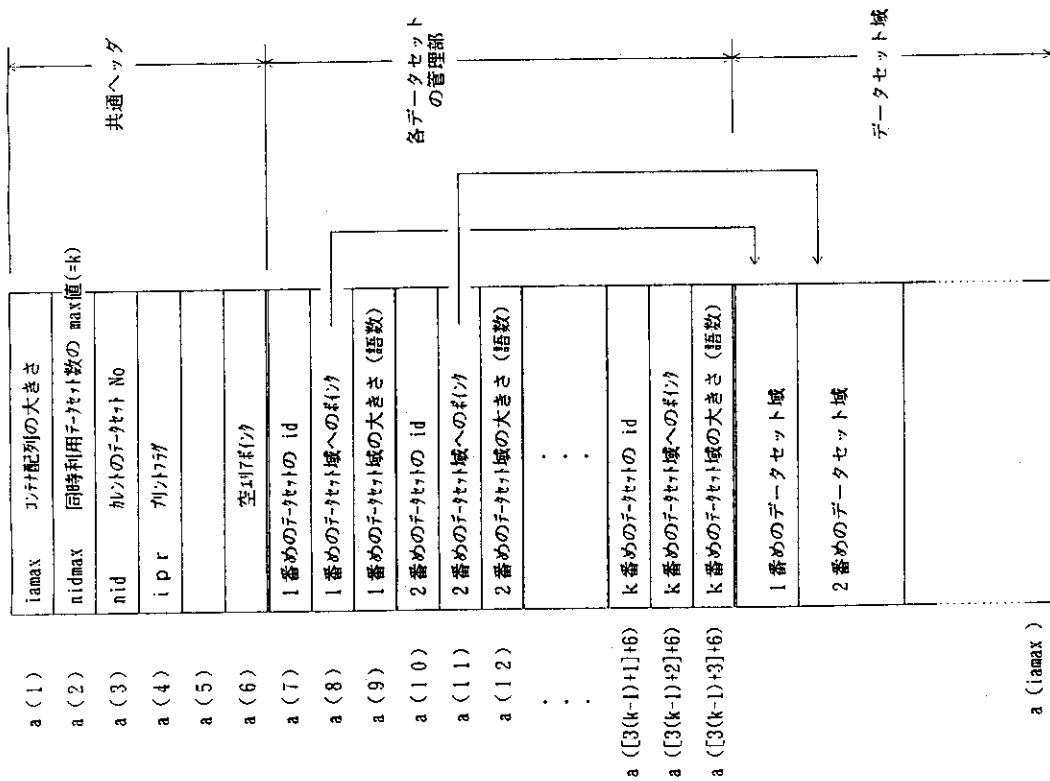


Fig. 4.12 Structure of container array 'a'

```

3 1 #total*/
3 2 #elastic*/
3 4 #inelastic*/
3 51 #discrete inelastic*/
3 -68 #continued*/
3 91 #continuum inelastic*/
3 102 #n.g*/
3 103 #(n.p)*/
3 104 #(n.d)*/
3 107 #(n.a)*/
3 203 #total h production*/
3 204 #total h2 production*/
3 207 #total he4 production*/
3 221 #free thermal scattering*/
3 229 #graphite inelastic thermal scattering*/
3 230 #graphite elastic thermal scattering*/
3 251 #muon*/
3 252 #xi*/
3 253 #gamma*/
3 301 #total heat production*/
3 444 #total damage energy production*/
6 2 #elastic*/
6 51 #discrete inelastic*/
6 -68 #continued*/
6 91 #continuum inelastic*/
6 221 #free thermal scattering*/
6 229 #graphite inelastic thermal scattering*/
6 230 #graphite elastic thermal scattering*/
17 51 #inelastic gamma production*/
16 102 #capture gamma production*/
0/
0/
#moder*
-23 25
#stop*
EOF
../xdnjoy -Wl. -u <input
echo 'saving output files'
cp output dou1
cp tape25 pend1

```

Fig. 4.15 Shell script for execution of test case 1 (2/2)

```

#!/bin/csh -f
#$-eo
#0$-q vppm
#0$-C EXDNJOY
cd $SUB_WORKDIR
echo 'getting endf tape 511'
cp ../1511 tape20
echo 'getting thermal tape 322'
cp ../1322 tape26
echo 'running njoy'
cat>input <<EOF
5
#moder*
20 -21
#reconr*
-21 -22
#pendf tape for c-nat from endf/b tape 511*/
1306 3/
.005 0. 7/
#6-c-nat from tape 511*/
#processed by the njoy nuclear data processing system*/
#see original endf/b-v tape for details of evaluation*/
0/
#broadr*
-22 -23
1306 1/
.005/
300.
0/
#heatr*
-21 -23 -22/
1306 1/
444
#thermr*
26 -22 -24
1065 1306 8 1 1 0 1 221 0
300
.05 1.2
#thermr*
26 -24 -23
1065 1306 8 1 4 1 1 229 0
300.
.05 1.2
#groupr*
-21 23 0 -24
1306 3 3 3 1 1 1
#carbon in graphite*/
300
L.e10

```

Fig. 4.15 Shell script for execution of test case 1 (1/2)

```

3 259 #1/vt/
6 2 #elastic#/
6 16 #n2n#/
6 17 #n,3n#/
6 18 #fission#/
6 51 #discrete inelastic#/
6 -59 #continued#/
6 91 #continuum inelastic#/
0/
3 1 #total#/
3 2 #elastic#/
3 18 #fission#/
3 102 #capture#/
6 2 #elastic#/
0/
3 1 #total#/
3 2 #elastic#/
3 18 #fission#/
3 102 #capture#/
6 2 #elastic#/
0/
0/
#cccr#
-25 21 22 0
1 1 #t1ant njoy#/
#cccr tests for njoy87#/
50 0 1 4 1
#pu238# #pu238# #endfb4# # 1050 # 1050 10.89
1 0 50 -1
0 2 3821e02 3.3003e-11 1.7461e-12 0. 1.e10 0.0
3 6
300 900 2100
1.e5 1.e4 1000. 100. 10. 1
#stop#
EOF
../xdnjoy -Wl, -u <input
echo saving output files'
cp output dout2
    
```

Fig. 4.16 Shell script for execution of test case 2 (2/2)

```

#!/bin/csh -f
#&S-eo
#&S-q yppm
#&S-C ENJOY
cd $QSUB_WORKDIR
echo getting endf tape 404'
cp ../t404 tape20
echo running njoy'
cat>input <<EOF
0
4
#moder#
20 -21
#reconr#
-21 -22
#pendf tape for pu-238 from endf/b-iv tape 404#/
1050 3/
.005 0. 7/
#94-pu-238 from endf/b tape t404#/
#processed by the njoy nuclear data processing system#/
#see original endf/b-iv tape for details of evaluation#/
0/
#broadr#
-22 -23
1050 3 0 1 0 /
.005/
300. 900. 2100.
0/
#unrest#
-21 -23 -22
1050 3 7 1
300 900 2100
1.e10 1.e5 1.e4 1000. 100. 10. 1
0/
#group#
-21 -22 0 -25
1050 5 0 4 3 3 7 1
#94-pu-238#/
300. 900. 2100.
1.e10 1.e5 1.e4 1000. 100. 10. 1
.1 0.025 0.8208e06 1.4e06
3 1 #total#/
3 2 #elastic#/
3 16 #n2n#/
3 17 #n3n#/
3 18 #fission#/
3 102 #capture#/
3 251 #ubar#
3 252 #xi#/
3 253 #gamma#/
    
```

Fig. 4.16 Shell script for execution of test case 2 (1/2)

```

l
#h# 1 1
#u# 92 92
view
#36 37/
#stop#
EOF
../xdnjoy -W1.-u <input
echo 'saving output and plot files'
cp output dout3
cp tape37 plot3
    
```

Fig. 4.17 Shell script for execution of test case 3 (2/2)

```

#!/bin/csh -f
#0$-eo
#0$-q vppm
#0$-C EXDNJOY
cd $OSUB_WORKDIR
echo 'getting photoatomic tape gam23'
cp ../gam23 tape30
echo 'getting photoatomic tape gam27'
cp ../gam27 tape32
echo 'running njoy'
cat>input <<EOF
0
4
#reconr#
30 31
#pndf tape for photon interaction cross sections from dlcTe*/
1 1 0
.001/
#1-hydrogen*/
92 1 0
.001/
#92-uranium*/
0/
#gaminr#
32 31 0 33
1 3 3 4 1
#12 group photon interaction library*/
-1 0/
-1 0/
0/
#dtlfr#
33 34 31 36
1 1 0
5 12 4 5 16 1 0
#pheat#
1 621 1
0/
#h# 1 1 0./
#u# 92 1 0./
0/
#matlxt#
0 33 35/
1 #12lant njoy*/
1 1 1 2
#12-group photon interaction library*/
#s#
12
#gscat#
1
    
```

Fig. 4.17 Shell script for execution of test case 3 (1/2)

```

#!/bin/csh -f
#0$-eo
#0$-q vppm
#0$-C EXDNJOY
cd $QSUB_WORKDIR
echo 'getting endf tape 511'
cp ../t511 tape20
echo 'running njoy...'
cat>input <<EOF
0
5
#moder*
20 -21
#reconr*
-21 -22
#u-235 10% pendf for errorr test problem from t511*#
1395 0 0
-10/
0/
#errorr*
-21 -22 0 23 0/
1395 19 1 1
3 0 0
0 33
1
1. e0 1. e3
#group*
-21 -22 0 24
1395 3 0 3 0 1 1 1
#u-235 multigroup nubar calculation*#
0.
1. e10
3 452 #total nubar*#
0/
0/
#errorr*
-21 0 24 25 23/
1395 1 1 1
0 31
7
1. e0 1. e1 1. e2 1. e3 1. e4 1. e5 1. e6 1. e7
#stop
EOF
.../xdnjoy -Wl. -u <input
echo 'saving output'
cp output dout4
    
```

Fig. 4.18 Shell script for execution of test case 4

```

#!/bin/csh -f
#0$-eo
#0$-q vppm
#0$-C EXDNJOY
cd $QSUB_WORKDIR
echo 'getting endf tape 511'
cp ../t511 tape30
echo 'running njoy...'
cat>input <<EOF
0
5
moder
30 -31
moder
-31 -32
errorr
-31 -32 0 -33/
1306 19 1/
2 0 0
0 33
1
1e-5 2e7/
covr
-33 0 34/
/
/
1306/
viewr
34 35/
stop
EOF
.../xdnjoy -Wl. -u <input
echo 'saving output and plot'
cp output dout5
cp tape35 plot5
    
```

Fig. 4.19 Shell script for execution of test case 5


```

#!/bin/csh -f
##$-eo
##$-o vppm
##$-C EXDNJOY
cd $QSUB_WORKDIR
echo 'getting endf tape 511'
cp ../t511 tape30
echo 'running njoy'
cat>input <<EOF
0
5
plotr
31/
/
1/
<<endf/b-v carbont/
<<total <c>ross <s>section*/
4/
1e3 2e7/
.
.5 10/
/
5 30 1306 3 1/
/
1/
<<endf/b-v carbont/
*(n,ja) with fake data/
1 0 2 1 1.3e7 .32/
/
/
/
/
5 30 1306 3 107/
/
<<endf/b-v mat1306*/
2/
0/
-1 0/
<<s>mith & <s>mith 1914*/
0/
1.1e7 .08 .05 .05/
1.2e7 .10 .05 .05/
1.3e7 .09 .04 .04/
1.4e7 .08 .03 .03/
-1/
3/
0/
-1 2/
<<b>lack & <b>lue 2008*/
0/
1.15e7 .07 .02 0 .2e6 0./
1.25e7 .11 .02 0 .2e6 0./
1.35e7 .08 .015 0 .2e6 0./
1.45e7 .075 .01 0 .2e6 0./

```

Fig. 4.20 Shell script for execution of test case 6 (1/2)

```

-1/
1/
<<endf/b-v carbont/
<<elastic <mat4>*/
-1 2/
/
/
/
/
/
5 30 1306 4 2/
/
1/
<<endf/b-v j>i-6*/
*(n,2h)ja >neutron distribution*/
-1 2/
/
/
0 12e6 2e6/
/
/
5 30 1303 5 24/
/
1/
<<endf/b-v j>i-6*/
*(n,2h)ja >neutron spectra vs <E>*/
4 0 2 2/
10. 2.e7/
/
1e-11 1e-6/
<<c>ross <s>section (barns/e<v>)**/
5 30 1303 5 24 0. 12/
/
1e10 <m>e<vt/
1e3 2e-11 1e2/
2/
/
5 30 1303 5 24 0. 16/
/
1e14 <m>e<vt/
1e4 2e-10 2e3/
3/
/
5 30 1303 5 24 0. 20/
/
1e20 <m>e<vt/
1e5 2e-9 4e4/
99/
viewr
31 32/
stop
EOF
../xdnjoy -Wl. -u <input
echo 'saving plot file'
cp tape32 plot6

```

Fig. 4.20 Shell script for execution of test case 6 (2/2)

```
#!/bin/csh -f
##$-eo
##$-q vpps
##$-C EXDNJOY
cd $SUB WORKDIR
echo getting endf tape 511
cp ../L511 tape20
echo running njoy...
cat>input <<EOF
0
5
$moder*
20 -21
$reconr*
-21 -22
$pendf tape for u-235 from endf/b-v tape 511* /
.005 /
$92-u-235 from endf/b-v tape 511* /
$processed by the njoy nuclear data processing system* /
$see original endf/b-v tape for details of evaluation* /
0 /
$broadr*
-22 -23
1395 1 0 1 0 /
.005 /
300.
0 /
$heatr*
-21 -23 -24/
1395 /
$group*
-21 -24 0 -25
1395 3 2 9 0 1 1 /
$u-235 from tape 511* /
300.
1.0e10
16 /
0 /
0 /
$acer*
-21 -24 -25 26 27 /
1 /
$enjoy test problem 7* /
1395 300. /
0.05 /
/
$stop*
EOF
../xdnjoy -Wl -u <input
echo saving output and ace files
cp output dout1
cp tape26 ace7
```

Fig. 4.21 Shell script for execution of test case 7

```
#!/bin/csh -f
##$-eo
##$-q vpps
##$-C EXDNJOY
cd $SUB WORKDIR
echo getting endf tape for ni-61
cp ../enif1 tape20
cat>input <<EOF
0
6
$moder
20 -21
$reconr
-21 -22
$pendf tape for endf/b-vi.1 28-ni-61a* /
2834 1 0 /
.01 0. 7 /
$28-ni-61a from endf/b-vi.1 1124 (hetrick.fu:orn1)* /
0 /
$broadr
-22 -23
2834 1 /
.01 /
300 /
0 /
$heatr
-21 -23 -24/
2834 6 0 1 0 2 /
302 303 304 402 443 444
$group
-21 -24 0 -22
2834 3 3 9 4 1 1 1
$ni61a endf/b-vi.1 30x12* /
300
1e10
3 /
3 251 $mubar* /
3 252 $xi* /
3 253 $gamma* /
3 259 $1/v* /
6 /
16 /
0 /
0 /
$acer
-21 -24 0 25 26
1 1 1 /
$28-ni-61a from endf-vi.1* /
2834 300. /
.01 /
/
$stop
EOF
echo "running njoy"
../xdnjoy -Wl -u <input
cp output dout8
```

Fig. 4.22 Shell script for execution of test case 8

5. TWOTRAN-II コードのインストール

2次元多群ディスクリット・オーディネーツ輸送コード TWOTRAN-II[7] の VPP500 へのインストールを行った。TWOTRAN-II コードにはベクトル化及び機能向上された版があるが、ここでは、M780 において現在利用しているオリジナル版を対象にしている。このオリジナル版では、FACOM OS IV/F4 MSP FORTRAN77(V10) のコンパイラ・システムを利用していた。本作業では、VPP500 における作業の前に、M780 において OS IV/F4 MSP FORTRAN77 EX(V12) のコンパイラ・システムを利用した動作確認を行った。これは、FORTRAN77 EX が VPP500 のコンパイラ UXP/M VPP FORTRAN77 EX/VP に親和性があるためであり、まず M780 上でコンパイラ仕様の吸収を図った。次いで、VPP500 上に移送し、インストール作業を行った。以下に、これらの作業内容について報告する。

5.1 M780 上での動作確認

FORTRAN77 EX によるコンパイルでは、FORTRAN77(V10) の場合にできるだけ合わせるようにコンパイラ・オプションの選択を行った。大きな違いは、機能体系が異なっている最適化のオプションである。FORTRAN77(V10) によるコンパイルでは OPTIMIZE(3) が指定されていたので、これと同等である OPT(E) を選択した。

FORTRAN77 EX によるロード・モジュールの実行では、以下のエラー・メッセージを5個出力し、エラー発生回数が打切り回数(10回)に達した旨のメッセージ(JWE0903I-U)を出力して異常終了した。

```
JWE0013I-E A FLOATING DIVISION EXCEPTION WAS DETECTED. PSW= ...
ERROR OCCURS AT OUTER SSN 00000156 ...
```

異常終了の原因は、サブルーチン OUTER の中の DO 390 において、以下の A(LVEL+G-1) による除算の式が不変式の先行評価の対象になっており、この時 A(LVEL+G-1) の値が 0.0 であったためである。不変式の先行評価を抑止するため、@OPTIONS XOPT(NOPREEX) 文をサブルーチン OUTER の先頭に挿入した。

```
IF (IEVT.EQ.2) TA=TA+EV/A(LVEL+G-1)
```

異常動作は、上記のもの以外には認められなかった。出力リストの比較においても、FORTRAN77(V10) の場合と FORTRAN77 EX の場合とでは出力方法の違いを除き、数値的に一致した。TWOTRAN-II のメンバー一覧を Table 5.1 に示す。FORTRAN77(V10) によるコンパイル・リンク、及び実行の JCL を Fig. 5.1 に、FORTRAN77 EX によるものを Fig. 5.2 に示す。また、動作確認に使用したテストデータを Fig. 5.3 に示す。

5.2 VPP500 への移植

本作業では、FORTRAN77 EX により動作確認したソース・コードを対象にした。これに対応するファイルをVPP500に移送し、コンパイル・リンク、実行、及び動作確認を行った。VPP500におけるTWOTRAN-IIのソース・ファイル一覧をM780上のもものと対比してTable 5.1に示す。コンパイラ・オプションは、FORTRAN77 EXのものを継承した。従って、最適化レベルは-0eを選択した。但し、VPP500のコンパイラはデフォルトが-0eであるため、パラメータの指定は省略した。実行にあたっては、TWOTRAN-IIで使用している入出力ファイルと装置参照番号XXとの対応を、環境変数fuXXを用いてシェル・スクリプトの中で以下のように定義した。但し、5番と6番は別扱いとした。装置参照番号5番は、標準入力であるためシェル・スクリプト内実行コマンドの直接入力とした。6番はTWOTRAN-IIからの出力リストであり、標準出力としておいた。コンパイル・リンク、実行のシェル・スクリプトをFig. 5.4, Fig. 5.5に示す。

```

setenv fu03 NAFLUX
setenv fu08 NDUMP1
setenv fu09 NDUMP2
setenv fu17 NEDIT
setenv fu18 NEXTRA
setenv fu30 ITFLUX
setenv fu31 IAFLUX
setenv fu32 ISNCON
setenv fu33 IFIXSR
setenv fu34 ISOTXS

```

動作確認には前節のテスト・データ (Fig. 5.3) を使用し、M780のFORTRAN77(V10)側の出力リストとVPP500の出力リストとを比較することにより行ったが、NEUTRON BALANCE他の値で顕著な差を検出した。VPP500側の結果には最適化レベル-0eの影響で、実行時に副作用が生じている可能性があったため、M780との比較の前に、最適化レベルを最低の-0bにして実行した結果との比較を行った。同じVPP500上での実行結果であるが、-0eを選択した場合と-0bを選択した場合とで異なる結果(大多数は後2~3桁の違い)を出力しており、最適化の副作用が生じていることがわかった。

副作用によって生じた計算誤差を解消するため、演算の評価方法の変更、及び不変式の先行評価に関する最適化のメッセージを出力するよう、コンパイラ・オプションに-Eepを指定した。これにより、副作用が生じる可能性のある最適化(演算評価方法の変更、不変式の先行評価)が行われた箇所の洗い出しを行った。対象となる最適化メッセージは、jwd8220i-i(副作用が生じる可能性のある最適化実施)、jwd8201i-i(不変式のループ外への移動)、jwd8206i-i(ループ内の不変項による除算をループ外でその逆数の乗算に変更)、及びjwd8208i-i(総和演算の計算方法を変更)の4個であり、以下の18ルーチンで出力されていた。

Main	csmesh	csprep	editor	fiscal
genflo	ifinxs	in	inital	initq
inner	inpt11	inpt15	out	outer
rebal	setbc	tests		

このうち計算誤差に関係するルーチンは以下の5ルーチンであった。対応するDOループのDO文の直前に、式の評価順序を変更する最適化を抑制する最適化制御行'VOCL LOOP,NOEVAL'、及び不変式の先行評価の最適化を抑制する最適化制御行'VOCL LOOP,NOPREEX'の挿入を行った。挿入した最適化制御行、及び対応DOループの一覧をFig. 5.6に示す。

fiscal	inital	outer	rebal	tests
--------	--------	-------	-------	-------

以上の対応により、最適化レベル -0e を選択した時の出力結果と -0b のもののが一致した。M780の出力結果とは、EFFECTIVE ABSORPTION, NEUTRON BALANCE, 及び IN SCATTERの値に顕著な差が見られたが、他は概ね頭から4~5桁の範囲で一致していた。後者については、計算環境の差が現れていることが判る。

5.3 VPP500 上での検証

VPP500に移送したTWOTRAN-IIの動作の妥当性を確認するため、更に出カリストの比較・差異原因の調査を行った。テスト・データとしては、前節で使用したテスト・データ(Fig. 5.3)(以降、テスト・データ1と呼ぶ)に加えて、新たにテスト・データ2を使用した。テスト・データ2をFig. 5.7に示す。以降、前者のテストをテスト・ケース1、後者のテストをテスト・ケース2と呼ぶ。確認は以下の出カリストをM780のFORTRAN77(V10)による結果とVPP500の結果とを比較することにより行った。

(1)TWOTRAN-IIの出カリスト

(2)TWOTRAN-IIが出力するバイナリ・ファイルITFLUXを入力として

スメア・プログラム(TWSMR)が出力する出カリスト

本テストの動作環境をFig. 5.8に示す。また、TWSMRのコンパイル・リンク、及び実行のシェル・スクリプトをFig. 5.9、及びFig. 5.10に示す。テスト・ケース1、及びテスト・ケース2におけるTWSMRの入カデータをFig. 5.11、及びFig. 5.12に示す。

5.3.1 テスト・ケース1

5.3.1.1 TWOTRAN-IIの出カリスト

前節にあるように、EFFECTIVE ABSORPTION, NEUTRON BALANCE, 及び IN SCATTERの値に顕著な差が見られた。それぞれの値をTable 5.2に示す。但し、NEUTRON BALANCEの値につ

いては、17組のうち1組だけ示してある。

(1)EFFECTIVE ABSORPTION 値について

問題の値は、サブルーチン CSPREP において、IHT=4, G=2, M=3の時のAAJ(3)として算出されている。具体的には、以下の代入文により算出されている。

$$AAJ(3)=C(4,2,3)-C(5,2,3)-C(6,3,3)-C(7,4,3)$$

計算時の各値、及びその計算値であるAAJ(3)の値をTable 5.3に示す。Table 5.3から判るように、C(4,2,3), C(5,2,3), C(6,3,3), C(7,4,3)の値はM780, VPP500いずれのものも近い値を示しているが、これらの計算値であるAAJ(3)の値については、その開きが大きくなっている。M780, VPP500の計算環境の差に伴う計算誤差であることが判る。

(2)NEUTRON BALANCE 値について

問題の値は、サブルーチン GSUMS の以下の代入文においてBAL(IGP)として算出されている。

$$BAL(IGP)=1.-(NL(IGP)+ABG(IGP)+SOUT(IGP))/$$

$$+(FG(IGP)+QG(IGP)+SIN(IGP))$$

計算時の各値、及びその計算値であるBAL(IGP)の値をTable 5.4に示す。Table 5.4から判るように、(1)と同様、計算式中の各要素の値はM780, VPP500いずれのものも近い値を示しているが、これらの計算値であるAAJ(3)の値については、その開きが大きくなっている。M780, VPP500の計算環境の差に伴う計算誤差であることが判る。

(3)IN SCATTER 値について

問題の値は、サブルーチン OUTER の以下の代入文においてSIN(G)として算出されている。

$$SIN(G)=TS-FG(G)-QG(G)$$

計算時の各値及びその計算値であるSIN(G)の値をTable 5.5に示す。SIN(G)の値の開きは、M780, VPP500の計算環境の差に伴う計算誤差によるものであるということがTable 5.5から判る。これは(1), (2)の場合と同様である。

5.3.1.2 TWSMR の出力リスト

最終桁が異なる程度のものが散見されたが顕著な差異は見当たらなかった。テスト・ケース1では、VPP500における不具合は検出できなかった。

5.3.2 テスト・ケース2

本テスト・ケースでは、最初の起動で、以下のメッセージを出力して異常終了した。

***ERROR** EXTENDED CORE STORAGE EXCEEDED

これは TWOTRAN-II で使用されるコンテナ配列 A の大きさが不足したためであった。起動時、配列 A の大きさは 70,000 であったが、必要な作業域の大きさは SMALL CORE が 32,025, LARGE CORE が 81,386, 計 113,411 であった。拡張領域不足で異常終了した。この対応として、メイン・ルーチンとサブルーチン INPT11 との 2 ルーチンに拡張のための修正を加えた。拡張の例を Fig.5.13 に示す。

上記の領域不足の他には異常終了せず、テストを継続した。テスト結果は以下に示すとおりである。

5.3.2.1 TWOTRAN-II の出力リスト

テスト・ケース 1 の場合と同様に、以下の値で顕著な差が見られた。

```
EFFECTIVE ABSORPTION
NEUTRON BALANCE
IN SCATTER
INNER ITERATIONS
```

INNER ITERATIONS の値の差異を Table 5.6 に示す。問題の値は、サブルーチン INNER において IITNO として算出されている。この処理部分を Fig. 5.14 に示す。Fig. 5.14 においては、IITL=29, EPSI=1.00E-04 であり、IITNO が 30, または ERR が 1.00E-04 より小さくなるまで 100 以降のループが実行される。ループ回数が INNER ITERATIONS 値 (IITNO) である。また ERR は以下の計算値 ($I=2 \sim IT, J=2 \sim JT$) の最大値である。

$$ABS(1.0 - (FLUXA(I-1, J-1) + FLUXA(I-1, J) + FLUXA(I, J-1) + FLUXA(I, J)) / T)$$

Table 5.6 において、INNER ITERATIONS の値が M780, VPP500 で異なる箇所の ERR 値及びこれを計算する各値の例を Table 5.7 ~ 9 に示す。

Table 5.7, 5.9 の場合、各 FLUXA 及び T の値は、頭 2 ~ 4 桁の範囲で一致している。FLUXA, T 自体も計算値であることを考えれば、計算環境の差に伴う計算誤差と思われる。これらの値をもとに ERR が計算されており、その差が INNER ITERATIONS 値に現れている。

Table 5.8 の場合は各 FLUXA 及び T の値が異なっている。これらは ERR が最大値となる時の値であり、計算誤差を考えると、M780 と VPP500 とで同じ組み合わせの値が選択されるとは限らない。従って、この時の全ての FLUXA(I, J) の値どうしを比較した。Fig. 5.15 に M780, VPP500 における FLUXA(I, J) の値の抜粋を示す。全データを比較したところ、頭 3 ~ 4 桁の範囲で一致している。T の値も FLUXA の計算値であることを考えれば、計算環境の差に伴う計算誤差が ERR の計算値の差異に現れ、INNER ITERATIONS の値に影響していることが判る。

5.3.2.2 TWOSMR の出力リスト

最終桁が異なる程度のものが散見されたが顕著な差異は見当たらなかった。テスト・ケース 1 と同様、テスト・ケース 2 においても VPP500 における不具合は検出できなかった。

5.4 まとめ

TWOTRAN-IIの利用形態の一つとして、TWOTRAN-IIを起動し、その出力ファイルであるITFLUXを入力としてTWSMRを起動する形態がある。従って、TWOTRAN-IIをVPP500にインストールする場合、その成否はTWSMRの出力データの妥当性に大きく依存する。幸い、この出力データには大きな差異が認められず、本作業を終えることができた。但し、確認テストとしては、2ケースしか行うことができず充分とは言えない状況である。今後、ユーザに使用していただき、品質向上を図って行きたい。

Table 5.1 List of source program files

M780における メンバー一覧		VPPにおける ソース・ファイル一覧	
CLEAR	CSMESH	clear.f	csmesh.f
CSPREP	DATE1	csprep.f	datel.f
DUMPER	DUMPRD	dumper.f	dumprd.f
ECHECK	ECRD	echeck.f	ecrd.f
ECWR	EDCALL	ecwr.f	edcall.f
EDITOR	EDMAP	editor.f	edmap.f
ERROR	FIDO	error.f	fido.f
FINAL	FISCAL	final.f	fiscal.f
FIXUP	GENFLO	fixup.f	genflo.f
GRID21	GRID22	grid21.f	grid22.f
GRID23	GRIND2	grid23.f	grind2.f
GSUMS	IFINQF	gsums.f	ifinqf.f
IFINSN	IFINXS	ifinsn.f	ifinxs.f
IFOUT	IFRITE	ifout.f	ifrite.f
IN	INITAL	in.f	inital.f
INITQ	INNER	initq.f	inner.f
INPT11	INPT12	inpt11.f	inpt12.f
INPT13	INPT14	inpt13.f	inpt14.f
INPT15	INPUT1	inpt15.f	input1.f
LOAD	MAIN	load.f	Main.f
MAPPER	MONITR	mapper.f	monitr.f
MPLY	NEWPAR	mply.f	newpar.f
OUT	OUTER	out.f	outer.f
OUTPT3	OUTT31	outpt3.f	outt31.f
OUTT32	PCMBAL	outt32.f	pcmbal.f
PNGEN	READQF	pngen.f	readqf.f
REBAL	REED	rebal.f	reed.f
RITE	SAVEAF	rite.f	saveaf.f
SECOND	SETBC	second.f	setbc.f
SNCON	STORAF	sncon.f	storaf.f
TESTS	WRITE	tests.f	write.f
	以上 62メンバ		以上 62ファイル

Table 5.2 Values of EFFECTIVE ABSORPTION, NEUTRON BALANCE and IN SCATTER

	EFFECTIVE ABSORPTION	NEUTRON BALANCE	IN SCATTER
M780	0.75623E-06	9.03606415E-05	1.0132790E-06
VPP	0.81398e-06	3.69548798e-06	1.1920929e-07

Table 5.3 Values of AAJ(3) and its elements

	入力データ	M780内格納データ	VPP内格納データ
C(4, 2, 3)	0.37485	0.37484998	0.374850005
C(5, 2, 3)	0.36830	0.36830002	0.368299991
C(6, 3, 3)	0.65492E-03	6.5491982E-03	6.54920004e-03
C(7, 4, 3)	0.0	0.00000000E+00	0.00000000e+00
AAJ(3)	—	0.75623E-06	0.81398e-06
手計算値	0.8E-06	0.7618E-06	0.81306E-06

Table 5.4 Values of BAL(IGP) and its elements

	M780	VPP
NL(IGP)	0.369148374	0.369128734
ABG(IGP)	0.629906952	0.629921556
SOUT(IGP)	2.09323025	2.09345150
FG(IGP)	0.999999762	1.00000000
QG(IGP)	0.0	0.00000000
SIN(IGP)	2.09256554	2.09251308
BAL(IGP)	9.03606415E-05	3.69548798e-06

Table 5.5 Values of SIN(G) and its elements

	M780	VPP
TS	1.00000095	1.00000012
FG(G)	0.999999940	1.00000000
QG(G)	0.0	0.00000000E+00
SIN(G)	1.01327896E-06	1.19209290e-07

Table 5.6 Difference of values of INNER ITERATIONS

OUTER ITERATIONS (OITNO)	GROUP NO. (G)	INNER ITERATIONS(IITNO)	
		M780	VPP
1	2	28	29
5	6	30	29
6	6	30	21
7	6	30	19
8	6	16	17
9	6	19	15
10	6	27	16
11	5	8	9
12	6	24	7
13	6	7	5

Table 5.7 Values of ERR and its elements(OITNO=1, G=2, IITNO=28)

	M 7 8 0	V P P
ERR	9.82284546E-05	1.00016594e-04
FLUXA(I-1, J-1)	8.97915289E-03	8.95133521e-03
FLUXA(I-1, J)	5.01132384E-03	5.00416430e-03
FLUXA(I, J-1)	8.99003446E-03	8.96465965e-03
FLUXA(I, J)	5.01738489E-03	5.00529399e-03
T	2.79951431E-02	2.79226620e-02

Table 5.8 Values of ERR and its elements(OITNO=7, G=6, IITNO=19)

	M 7 8 0	V P P
ERR	3.13758850E-04	9.14335251e-05
FLUXA(I-1, J-1)	2.78179968E-09	5.24935047e-08
FLUXA(I-1, J)	2.87512902E-10	6.49311232e-07
FLUXA(I, J-1)	7.94529997E-10	4.56294558e-09
FLUXA(I, J)	1.41956380E-09	5.84554716e-09
T	5.28174482E-09	7.12148164e-07

Table 5.9 Values of ERR and its elements(OITNO=11, G=5, IITNO=8)

	M 7 8 0	V P P
ERR	9.91821289E-05	1.02043152e-04
FLUXA(I-1, J-1)	4.14973474E-05	4.14906244e-05
FLUXA(I-1, J)	7.22014112E-04	7.21901422e-04
FLUXA(I, J-1)	8.60663931E-06	8.60529872e-06
FLUXA(I, J)	1.84975710E-04	1.84947683e-04
T	9.56997974E-04	9.56847449e-04

2	**PVITR	SHUT	DOWN	MARGIN	TEMP=NORMAL	RZ	.96	2.2	NEW	VERSION	TEST
**VSOP	-----	EXTM(X76)	OVERALL	SURVEY							
00001000	0	0	6	4	2	3	1	0	0	0	1
00002002	3	0	3	0	4	5	12	0	0	0	30
00003000	0	1200	2	0	0122131	0	7	3	1	0	00005000
00004000	0.0	0.1	0.0	0.01		0.5					00006004
00005000	0.00050	1.0	1.0	0.0							00007000
00006000	8	253	4	43							00008000
00007000	4	4	4	43							00009000
00008000	VSOP COMPOSITION NO. 19										00009106
00009000	0.15777E+00	0.37606E-04	0.52566E-04	0.15777E+00	0.14879E+00	0.0					00025400
00025400	0.0	0.0	0.0	0.0	0.0	0.0					00025500
00025500	0.24414E+00	0.49149E-03	0.18060E-03	0.24414E+00	0.23942E+00	0.89435E-02	0.0025700				00025600
00025600	0.0	0.0	0.0	0.0	0.0	0.0					00025800
00025800	0.24784E+00	0.25065E-02	0.58570E-03	0.24784E+00	0.23222E+00	0.42275E-02	0.0025900				00026000
00026000	0.0	0.0	0.0	0.0	0.0	0.0					00026200
00026200	0.24814E+00	0.24981E-02	0.41977E-02	0.24814E+00	0.24564E+00	0.13117E-01	0.00026100				00026400
00026400	0.0	0.0	0.0	0.0	0.0	0.0					00026600
00026600	VSOP COMPOSITION NO. 73										00026800
00026800	0.29233E+00	0.47761E-02	0.0	0.29233E+00	0.27174E+00	0.0					00027000
00027000	0.0	0.0	0.0	0.0	0.0	0.0					00027200
00027200	0.60395E+00	0.16654E+00	0.0	0.60395E+00	0.42954E+00	0.15813E-01	0.00074300				00027400
00027400	0.0	0.0	0.0	0.0	0.0	0.0					00027600
00027600	0.19367E+01	0.14956E+01	0.0	0.19367E+01	0.41664E+00	0.78695E-02	0.00074500				00027800
00027800	0.0	0.0	0.0	0.0	0.0	0.0					00028000
00028000	0.11002E+02	0.10571E+02	0.0	0.11002E+02	0.43127E+00	0.24452E-01	0.00074700				00028200
00028200	0.0	0.0	0.0	0.0	0.0	0.0					00028400
00028400	VSOP COMPOSITION NO. 75										00028600
00028600	0.24335E+00	0.83655E-05	0.0	0.24335E+00	0.22948E+00	0.0					00028800
00028800	0.0	0.0	0.0	0.0	0.0	0.0					00029000
00029000	0.37485E+00	0.0	0.0	0.37485E+00	0.36830E+00	0.13863E-01	0.00076100				00029200
00029200	0.0	0.0	0.0	0.0	0.0	0.0					00029400
00029400	0.37804E+00	0.16938E-04	0.0	0.37804E+00	0.35771E+00	0.65492E-02	0.00076300				00029600
00029600	0.0	0.0	0.0	0.0	0.0	0.0					00029800
00029800	0.38027E+00	0.13562E-03	0.0	0.38027E+00	0.38013E+00	0.26317E-01	0.00076500				00029900
00029900	0.0	0.0	0.0	0.0	0.0	0.0					00030003
00030003	0.0	123.0	220.0	3							00030104
00030104	0.0	100.0	520.0	3							00030200
00030200	1.4	1	1	1	2	39					00030300
00030300	1.4	1	1	1	2	39					00030404
00030404	1.4	1	1	1	2	33					
	1.0	1	3	0	3						
	1.4	0.0	3								
	1.4	2		51	2	103					
	1.3	43									
	0.0	33.0	63.00	93.00	123.0	130.0					
	136.0	220.0	3								
	0.0	100.0	520.0	620.0	3						

Fig. 5.3 Test data

```

TWC(6 4 8 3 4) SRP
//STEP1 EXEC XFOR77.S0=J9133.TWOTRAN.ORG
//A=ELM(*).AE.RGN=8192k.OPT=3
//STEP2 EXEC XLKED.A=MAP
//STEP3 EXEC GO
//*****
//FT05F001 DD DSN=J9133.TWOTRAN2.NEW.CNTL(INPUT).DISP=SHR.LABEL=C...IN
//FT05F001 DD DSN=J9133.TWOTRAN2G.DATA(TWMICHE).DISP=SHR.LABEL=C...IN
//FT06F001 DD DSN=J9133.TWOTRAN.RO.UTLIST.DISP=(NEW.CATLG)
//DCB=(RECFM=FBA.LRECL=137.BLKSIZE=19043)
//SPACE=(TRK,(40,10).RLSE).UNIT=TSSWK
//FT08F001 DD DSN=J9133.TWMIC.FT08.DATA.DISP=(NEW.CATLG.CATLG)
//UNIT=TSSWK
//SPACE=(TRK,(50,50).RLSE).DCB=(RECFM=VBS.LRECL=800.BLKSIZE=8004)
//FT09F001 DD DSN=J9133.TWMIC.FT09.DATA.DISP=(NEW.CATLG.CATLG)
//UNIT=TSSWK
//SPACE=(TRK,(50,50).RLSE).DCB=(RECFM=VBS.LRECL=800.BLKSIZE=8004)
//FT30F001 DD DSN=J9133.TWMIC.FT30.DATA.DISP=(NEW.CATLG.CATLG)
//UNIT=TSSWK
//SPACE=(TRK,(50,50).RLSE).DCB=(RECFM=VBS.LRECL=800.BLKSIZE=8004)
//EXPAND DISK.DD=FT30F001.SPC=50.50
//EXPAND DISK.DD=FT17F001.SPC=50.50
//EXPAND DISK.DD=FT18F001
//EXPAND DISK.DD=FT40F001
//EXPAND DISK.DD=FT31F001.SPC=100.50
//EXPAND DISK.DD=FT32F001
//EXPAND DISK.DD=FT33F001
//EXPAND DISK.DD=FT34F001
//EXPAND DISK.DD=FT30F001
//*
    
```

Fig. 5.1 JCL for compilation, linkage and execution (FORTRAN77)

```

TWC(6 4 8 3 4) SRP
//STEP1 EXEC XFOR77.S0=J9133.TWOTRAN.NK
//A=ELM(*).AE.B=NOMAP.RGN=8192k.OPT=E
//STEP2 EXEC LKED.A=MAP
//STEP3 EXEC GOEX
//*****
//FT05F001 DD DSN=J9133.TWOTRAN2.NEW.CNTL(INPUT).DISP=SHR.LABEL=C...IN
//FT05F001 DD DSN=J9133.TWOTRAN2G.DATA(TWMICHE).DISP=SHR.LABEL=C...IN
//FT06F001 DD DSN=J9133.TWOTRAN.RO.UTLIST.DISP=(NEW.CATLG)
//DCB=(RECFM=FBA.LRECL=137.BLKSIZE=19043)
//SPACE=(TRK,(40,10).RLSE).UNIT=TSSWK
//FT08F001 DD DSN=J9133.TWMIC.FT08EX.DATA.DISP=(NEW.CATLG.CATLG)
//UNIT=TSSWK
//SPACE=(TRK,(50,50).RLSE).DCB=(RECFM=VBS.LRECL=800.BLKSIZE=8004)
//FT09F001 DD DSN=J9133.TWMIC.FT09EX.DATA.DISP=(NEW.CATLG.CATLG)
//UNIT=TSSWK
//SPACE=(TRK,(50,50).RLSE).DCB=(RECFM=VBS.LRECL=800.BLKSIZE=8004)
//FT30F001 DD DSN=J9133.TWMIC.FT30EX.DATA.DISP=(NEW.CATLG.CATLG)
//UNIT=TSSWK
//SPACE=(TRK,(50,50).RLSE).DCB=(RECFM=VBS.LRECL=800.BLKSIZE=8004)
//EXPAND DISK.DD=FT30F001.SPC=50.50
//EXPAND DISK.DD=FT17F001.SPC=50.50
//EXPAND DISK.DD=FT18F001
//EXPAND DISK.DD=FT40F001
//EXPAND DISK.DD=FT31F001.SPC=100.50
//EXPAND DISK.DD=FT32F001
//EXPAND DISK.DD=FT33F001
//EXPAND DISK.DD=FT34F001
//EXPAND DISK.DD=FT30F001
//*
    
```

Fig. 5.2 JCL for compilation, linkage and execution (FORTRAN77 EX)

```

#!/bin/csh -f
#0$-q $SPC
#0$-eo
#0$-C CLTRA2
cd $QSUB__WORKDIR
frtpr -o xtra2 -Z comp.XTRA2 -Wv -sc -Eep -Si t.f
    
```

Fig. 5.4 Shell script for compilation and linkage

```

#!/bin/csh -f
#0$-eo
#0$-q vpps
#0$-C ETRA2
cd $QSUB__WORKDIR
echo running TWOTRAN2/vpp
setenv fu03 NAFLUX
setenv fu06 output
setenv fu08 NDUMPI
setenv fu09 NDUMPI2
setenv fu17 NEDIT
setenv fu18 NEXTRA
setenv fu30 ITFLUX
setenv fu31 IAFLUX
setenv fu32 ISNCON
setenv fu33 IFIXSR
setenv fu34 ISOTXS
../xtra2 -#i -u < input
echo 'TWOTRAN2/vpp finished'
    
```

Fig. 5.5 Shell script for execution

```

*** SUBROUTINE FISCAL ***
C*vppt B insert 'fvocl loop.noeval.nopreex' for vpp
*VOCL LOOP.NOEVAL.NOPREEX
C*vppt E
DO 130 J=1, JT
DO 130 I=1, IT
IB=IDX(I)
JA=IDYA(J)
TA=FISSA(I, J)*A5(I)*YH(J)
FT(IB, JA)=FT(IB, JA)+TA*CHIA(IGP)
130 T=T+TA
vpp.0010
vpp.0020
vpp.0030
00650000
00660002
00670002
00680002
00690002
00700002
00760002

*** SUBROUTINE INITIAL ***
C*vppt B insert 'fvocl loop.noeval' for vpp
*VOCL LOOP.NOEVAL
C*vppt E
DO 100 M=1, MM
WNU(M)=COSMU(M)*WGT(M)
WETA(M)=COSETA(M)*WGT(M)
SUMMU=SUMMU+WNU(M)
100 SUMETA=SUMETA+WETA(M)
vpp.0010
vpp.0020
vpp.0030
00000660
00000870
00000680
00000690
00000700

C*vppt B insert 'fvocl loop.noeval' for vpp
*VOCL LOOP.NOEVAL
C*vppt E
DO 460 G=1, IGM
T=CHI(G)
IF (IPVT.EQ.1) T=T/PV
CHIA(G)=T
CHI(G)=T
460 CHIA(IGP)=CHIA(IGP)+CHIA(G)
vpp.0040
vpp.0050
vpp.0060
00002300
00002310
00002320
00002330
00002340
00002350

*** SUBROUTINE OUTER ***
C*vppt B insert 'fvocl loop.noeval' for vpp
*VOCL LOOP.NOEVAL
C*vppt E
DO 290 J=1, JT
K=IDYA(J)
DO 290 M=1, MM
290 QQ(CH, K)=QQ(CH, K)+WNU(M)*YH(J)*(QRI(I, M)+QR2(I, M))*A4(ITP)
vpp.0010
vpp.0020
vpp.0030
00001890
00001900
00001910
00001920

C*vppt B insert 'fvocl loop.noeval' for vpp
*VOCL LOOP.NOEVAL
C*vppt E
DO 310 I=1, IT
K=IDX(I)
DO 310 M=1, MM
310 QQ(K, I)=QQ(K, I)+WETA(M)*(QB1(I, M)+QB2(I, M))*A3(I)
vpp.0040
vpp.0050
vpp.0060
00001940
00001950
00001960
00001970
    
```

Fig. 5.6 VOCL statements (1/2)

```

*** SUBROUTINE OUTER *** (続き)
C*vppt B insert ' *vocl loop, noeval' for vpp
*VOCL LOOP, NOEVAL
C*vppt E
  DO 330 I=1, IT
    K=JDX(I)
    DO 330 M=1, NM
      330 QQ(K, JM)=QQ(K, JM)+WETA(M)*(QT1(I, M)+QT2(I, M))+A3(I)

C*vppt B insert ' *vocl loop, noeval' for vpp
*VOCL LOOP, NOEVAL
C*vppt E
  DO 350 J=1, JM
    DO 350 I=1, IM
      350 TS=TS+QQ(I, J)

*** SUBROUTINE REBAL ***
C*vppt B insert ' *vocl loop, noeval' for vpp
*VOCL LOOP, NOEVAL
C*vppt E
  DO 230 J=1, JM
    T=T+PR(IP, J)-FL(IP, J)-FR(I, J)+FL(I, J)
    DO 230 I=1, IM
      230 T=T+AB(I, J)

*** SUBROUTINE TESTS ***
C*vppt B insert ' *vocl loop, noeval' for vpp
*VOCL LOOP, NOEVAL
C*vppt E
  DO 150 J=1, JM
    DO 150 I=1, IM
      150 T=T+P(I, J)+T(I, J)
  
```

Fig. 5.6 VOCL statements (2/2)

```

4
VHTR-S62P2 < CB-EFF, MACRO CAL, HOMO-FUEL(WITH CR-CLAD) > 180SYM,
66, SA, PI E=6, OW/O TEMP= 300/ 300K BP-2, OX12DX2 4-COLM
FUEL: SB=31, 8P(33/31P), ABS=105/75MM PI=36, 1CM X-Y MODEL WHITE B. C.
CASE-BE (CR=IN, BP=IN) TAPB=J3873, $TWRS$E1, E2E 42*70 MESH 87. 05.
0 1 4 6 5 6 1 0 0 0 1
16 0 16 0 4 7 12 0 0 0 0 0 30
0 0 1190 1 0 0 0 120001 0 42 70 1
0.0
0.0001 1.0 290.00 9 26 4 43
0.0001 17 4 26 4 43
3
CR-INNER-HE (PO)
4. 4159E-02 2. 4066E-04 0. 0000E+00 5. 7596E-02 0. 0000E+00 0. 0000E+0000000300
5. 5982E-02 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000000500
1. 7988E-01 7. 7635E-04 0. 0000E+00 1. 8485E-01 0. 0000E+00 0. 0000E+0000000500
1. 8296E-01 1. 3621E-03 0. 0000E+00 0. 0000E+00 0. 0000E+0000000600
1. 9767E-01 1. 8643E-03 0. 0000E+00 0. 0000E+00 0. 0000E+0000000700
1. 9710E-01 1. 1108E-03 6. 3705E-07 0. 0000E+00 0. 0000E+0000000700
1. 9834E-01 8. 0460E-03 0. 0000E+00 2. 0064E-01 8. 1335E-17 9. 2441E-0800000900
1. 9251E-01 9. 4311E-04 0. 0000E+00 0. 0000E+00 0. 0000E+0000001100
2. 0962E-01 1. 9318E-02 0. 0000E+00 2. 1192E-01 0. 0000E+00 1. 9100E-0500001100
1. 9253E-01 8. 1500E-05 0. 0000E+00 0. 0000E+00 0. 0000E+0000001300
2. 3232E-01 4. 2000E-02 0. 0000E+00 2. 3461E-01 0. 0000E+00 0. 0000E+0000001300
1. 9260E-01 6. 5701E-05 2. 8727E-08 0. 0000E+00 0. 0000E+0000001400
CR-INNER-HE (PT)
0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000001600
1. 1953E-02 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000001600
0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000001800
3. 1051E-03 -2. 1470E-04 0. 0000E+00 0. 0000E+00 0. 0000E+0000001900
0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000002000
2. 7038E-03 -3. 4612E-04 0. 0000E+00 0. 0000E+00 -6. 7165E-17 4. 4800E-0800002200
0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000002300
2. 3190E-03 -3. 1205E-04 0. 0000E+00 0. 0000E+00 0. 0000E+0000002400
0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+00 6. 1836E-0700002400
2. 3071E-03 -1. 4965E-05 0. 0000E+00 0. 0000E+00 0. 0000E+0000002500
0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000002600
2. 2972E-03 -7. 9568E-06 -1. 6154E-08 0. 0000E+00 0. 0000E+0000002700
CR(PO)
2. 1001E-01 3. 6126E-03 0. 0000E+00 2. 6958E-01 0. 0000E+00 0. 0000E+0000003000
2. 4823E-01 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000004000
4. 3392E-01 4. 0887E-02 0. 0000E+00 4. 6040E-01 0. 0000E+00 0. 0000E+0000005000
4. 0697E-01 1. 7736E-02 0. 0000E+00 0. 0000E+00 0. 0000E+0000006000
1. 1217E+00 7. 1451E-01 0. 0000E+00 1. 1442E+00 0. 0000E+00 0. 0000E+0000007000
4. 2004E-01 1. 2548E-02 0. 0000E+00 3. 9150E+00 1. 0675E-14 1. 1336E-0400009000
3. 8783E+00 3. 4658E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000010000
4. 1860E-01 9. 6838E-03 0. 0000E+00 8. 7673E+00 0. 0000E+00 4. 5959E-0300001100
8. 7404E+00 8. 3262E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000012000
4. 2576E-01 3. 0566E-02 0. 0000E+00 1. 8556E+01 0. 0000E+00 0. 0000E+0000001300
1. 8539E+01 1. 8107E+01 0. 0000E+00 0. 0000E+00 0. 0000E+0000014000
4. 4476E-01 1. 5256E-02 4. 6297E-06 0. 0000E+00 0. 0000E+00 0. 0000E+0000001600
0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000001800
3. 2336E-02 0. 0000E+00 0. 0000E+00 0. 0000E+00 0. 0000E+0000001700
  
```

Fig. 5.7 Test data (1/7)

Table with multiple columns of numerical data, including test data points and labels like FUJII (P0), CR-OUTER-HE (P0), and BLOCK-HE (P0). Each entry typically consists of a label, a value, and a range.

Fig. 5.7 Test data (3/7)

Fig. 5.7 Test data (2/7)

11.0075	11.8031	12.3166	12.5776	13.1031	13.3043	00054100
14.0681	14.6031	15.0	15.7938	18.05	20.3063	00054200
22.5	24.8188	27.0	29.3313	31.5	33.8438	00054300
36.10	3					00054400
0.0	2.6053	5.2106	7.8159	10.4212	13.0264	00054500
15.6317	18.2370	20.8423	23.4476	24.0	25.6347	00054600
26.1697	28.3335	27.1347	27.6602	27.9212	28.4347	00054700
29.2303	29.5057	30.1847	30.8847	31.5847	32.2637	00054800
32.5391	33.3347	33.8482	34.1092	34.6347	34.8359	00054900
35.5997	36.1347	37.0	38.0	39.0	40.0	00055000
41.0	41.8347	42.3697	43.1335	43.3347	43.8602	00055100
44.1212	44.6347	45.4303	45.7057	46.3847	47.0847	00055200
47.7847	48.4637	48.7391	49.5347	50.0482	50.3092	00055300
50.8347	51.0359	51.7997	52.3347	53.0	54.7112	00055400
57.3165	59.9218	62.5270	65.1324	67.7377	70.3430	00055500
72.9482	75.5535	78.1588	80.7641	83.3694	87.3694	00055600

Fig. 5.7 Test data (7/7)

00048500	-31 1	-11 6	-31 4	-51 6	-71 2	-71 2	00048500
-500048600	-31 2	-11 4	-31 8	-99	-71 8	-71 8	-500048600
00048700	-31 1	-11 5	-31 8	-51 4	-71 8	-71 8	00048700
-500048800	-31 2	-11 3	-312	-99	-71 8	-71 8	-500048800
00048900	-31 1	-11 4	-3112	-51 6	-71 8	-71 8	00048900
-500049000	-31 1	-11 3	-3114	-51 4	-71 8	-71 8	-500049000
00049100	-31 1	-11 3	-3114	-99	-71 8	-71 8	00049100
-500049200	-31 1	-11 3	-3114	-51 4	-71 8	-71 8	-500049200
00049300	-31 1	-11 3	-3114	-99	-71 8	-71 8	00049300
-500049400	-31 1	-11 3	-3114	-51 4	-71 2	-71 2	-500049400
00049500	-31 1	-11 3	-3114	-99	-71 8	-71 8	00049500
-500049600	-31 1	-11 3	-3114	-51 4	-71 8	-71 8	-500049600
00049700	-31 1	-11 3	-3114	-99	-71 8	-71 8	00049700
-500049800	-31 1	-11 3	-3114	-51 4	-71 8	-71 8	-500049800
00049900	-31 1	-11 3	-3114	-99	-71 8	-71 8	00049900
-500050000	-31 1	-11 3	-3114	-51 4	-71 2	-71 2	-500050000
00050100	-31 1	-11 3	-3114	-99	-71 8	-71 8	00050100
-500050200	-31 1	-11 3	-3114	-51 4	-71 8	-71 8	-500050200
00050300	-31 1	-11 4	-3112	-99	-71 8	-71 8	00050300
-500050400	-31 2	-11 3	-3112	-51 5	-71 8	-71 8	-500050400
00050500	-31 1	-11 3	-3112	-99	-71 8	-71 8	00050500
-500050600	-31 2	-11 3	-3112	-51 4	-71 3	-71 3	-500050600
00050700	-31 1	-11 5	-31 8	-99	-71 8	-71 8	00050700
-500050800	-31 1	-11 4	-31 8	-51 6	-71 1	-71 1	-500050800
00050900	-31 2	-11 4	-31 8	-99	-71 8	-71 8	00050900
-500051000	-31 1	-11 6	-31 4	-51 4	-71 3	-71 3	-500051000
00051100	-71 8	-51 4	-31 4	-99	-71 8	-71 8	00051100
-500051200	-71 8	-51 4	-31 1	-5116	-71 2	-71 2	-500051200
00051300	-71 8	-51 4	-31 1	-99	-71 8	-71 8	00051300
-900051400	-71 8	-51 4	-31 4	-5110	-71 5	-71 5	-900051400
00051500	-71 8	-51 7	-31 3	-51 6	-71 3	-71 3	00051500
-900051600	-99	-91 7	-71 2	-51 8	-71 8	-71 8	-900051600
00051700	-99	-91 7	-99	-91 7	-99	-99	00051700
00051900	-99	-91 7	-99	-91 7	-99	-99	00051900
00052000	-99	-91 7	-99	-91 7	-99	-99	00052000
00052200	-99	-91 7	-99	-91 7	-99	-99	00052200
00052300	-99	-91 7	-99	-91 7	-99	-99	00052300
00052400	-99	-91 7	-99	-91 7	-99	-99	00052400
00052500	-99	-91 7	-99	-91 7	-99	-99	00052500
00052600	-99	-91 7	-99	-91 7	-99	-99	00052600
00052700	-99	-91 7	-99	-91 7	-99	-99	00052700
00052800	-99	-91 7	-99	-91 7	-99	-99	00052800
00052900	-99	-91 7	-99	-91 7	-99	-99	00052900
00053000	-99	-91 7	-99	-91 7	-99	-99	00053000
00053100	-99	-91 7	-99	-91 7	-99	-99	00053100
00053200	-99	-91 7	-99	-91 7	-99	-99	00053200
00053300	-99	-91 7	-99	-91 7	-99	-99	00053300
00053400	-99	-91 7	-99	-91 7	-99	-99	00053400
00053500	-99	-91 7	-99	-91 7	-99	-99	00053500
00053600	-99	-91 7	-99	-91 7	-99	-99	00053600
00053700	-99	-91 7	-99	-91 7	-99	-99	00053700
00053800	-99	-91 7	-99	-91 7	-99	-99	00053800
00053900	-99	-91 7	-99	-91 7	-99	-99	00053900
00054000	-99	-91 7	-99	-91 7	-99	-99	00054000
0.967231	0.7	1.3790	1.6544	2.2563	2.45	2.45	0.967231
0.0327691	3.2245	3.75	3.9512	4.1031	4.6781	4.6781	0.0327691
4.0.0	5.4019	5.6031	6.1286	6.3896	6.9031	6.9031	4.0.0
0.0	7.6987	7.9631	8.6531	9.3531	10.0531	10.0531	0.0
1.6							1.6
1.0							1.0
3							3
13							13
13							13
170							170

Fig. 5.7 Test data (6/7)

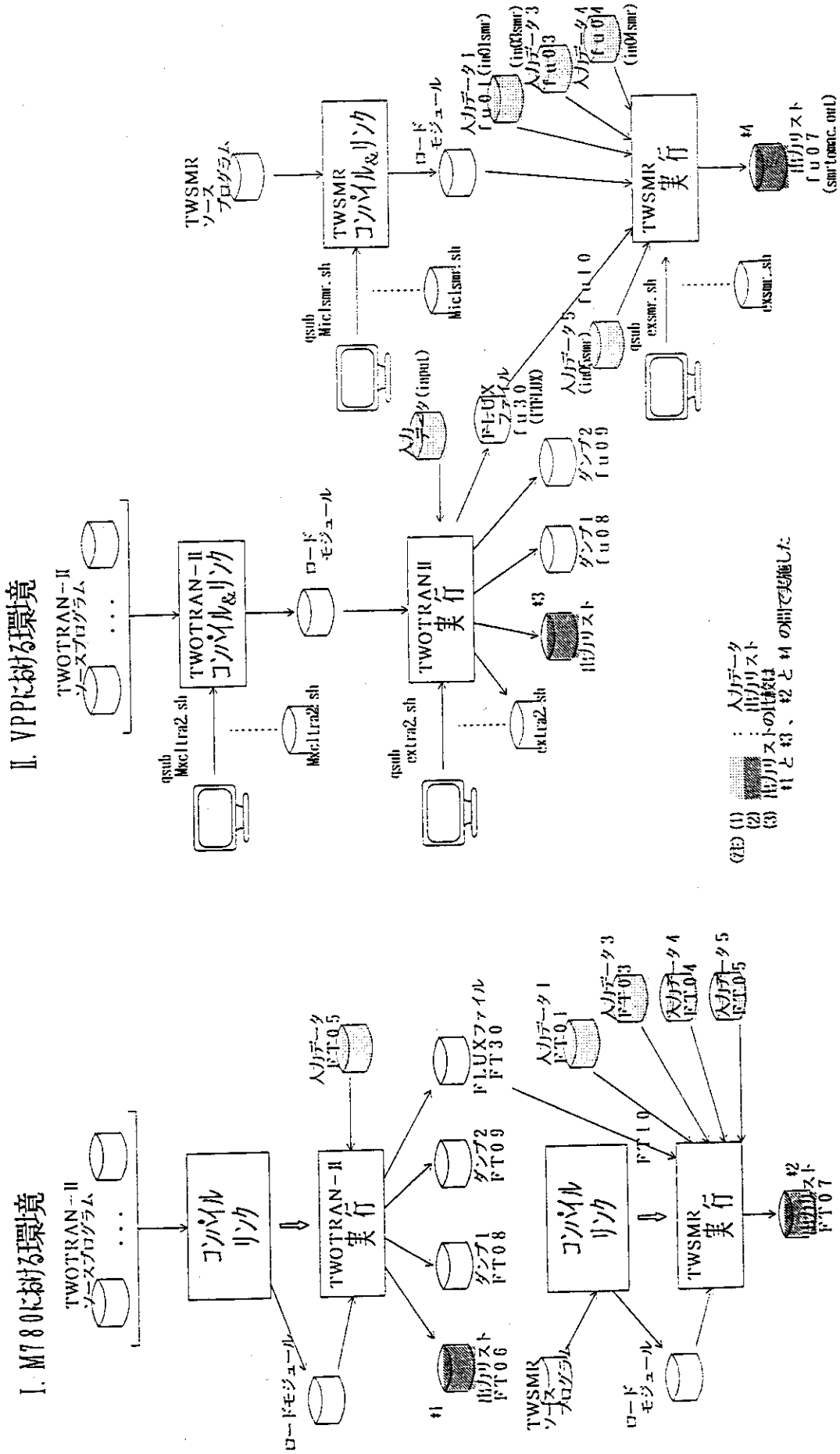


Fig. 5.8 Execution environment on M780 and VPP500

```
#!/bin/csh -f
#0$-q gspc
#0$-eo
#0$-C DCNJOY
cd $QSUB_WORKDIR
fritpx -o xtwsmr -Ob, -S -Z COMP.smr -Wv, -sc -Sw *.f
```

Fig. 5.9 Shell script for compilation and linkage (TWSMR)

```
#!/bin/csh -f
#0$-eo
#0$-q vpps
#0$-C ETWSMR
cd $QSUB_WORKDIR
echo running TWSMR/vpp *
setenv fu01 in01smr
setenv fu03 in03smr
setenv fu04 in04smr
setenv fu07 smr1omac.out
setenv fu10 ITFLUX
../xtwsmr -Wl, -u < in05smr
echo TWSMR/vpp finished
```

Fig. 5.10 Shell script for execution (TWSMR)

in01smr														
3	4	5	12	VSOP COMPOSITION NO. 75										
0.24335E+00	0.83655E-05	0.0	0.0	0.24335E+00	0.22948E+00	0.0	0.0	0.37485E+00	0.36830E+00	0.13863E-01	0.00000113	0.00000115		
0.0	0.0	0.0	0.0	0.37485E+00	0.0	0.0	0.0	0.37804E+00	0.35771E+00	0.65492E-02	0.00000117	0.00000118		
0.0	0.0	0.0	0.0	0.37804E+00	0.16938E-04	0.0	0.0	0.38027E+00	0.38013E+00	0.20317E-01	0.00000119	0.00000120		
0.0	0.0	0.0	0.0	0.38027E+00	0.13562E-03	0.0	0.0	0.0	0.0	0.0	0.00000121	0.00000122		
VSOP COMPOSITION NO. 19														
0.15777E+00	0.37606E-04	0.52566E-04	0.15777E+00	0.15777E+00	0.14879E+00	0.0	0.0	0.0	0.0	0.0	0.00000123	0.00000125		
0.0	0.0	0.0	0.0	0.24414E+00	0.49149E-03	0.18060E-03	0.24414E+00	0.23942E+00	0.89435E-02	0.00000126	0.00000127	0.00000128		
0.0	0.0	0.0	0.0	0.24784E+00	0.25065E-02	0.58570E-03	0.24784E+00	0.23222E+00	0.42275E-02	0.00000129	0.00000130	0.00000131		
0.0	0.0	0.0	0.0	0.24814E+00	0.24981E-02	0.41977E-02	0.24814E+00	0.24564E+00	0.13117E-01	0.00000132	0.00000133	0.00000135		
VSOP COMPOSITION NO. 73														
0.29233E+00	0.47761E-02	0.0	0.0	0.29233E+00	0.27117E+00	0.0	0.0	0.0	0.0	0.0	0.00000136	0.00000137		
0.0	0.0	0.0	0.0	0.60395E+00	0.16654E+00	0.0	0.0	0.60395E+00	0.42954E+00	0.15813E-01	0.00000138	0.00000139		
0.0	0.0	0.0	0.0	0.19367E+01	0.14956E+01	0.0	0.0	0.19367E+01	0.41664E+00	0.78695E-02	0.00000140	0.00000141		
0.0	0.0	0.0	0.0	0.11002E+02	0.10571E+02	0.0	0.0	0.11002E+02	0.43127E+00	0.24452E-01	0.00000142	0.00000143		
in03smr														
RZ	7	3												
2	83.0	2	30.0	2	30.0	2	30.0	2	30.0	5	7.0	10	6.0	
10	84.0	4	100.0	4	420.0	4	100.0							
in04smr														
1	33	1	12											
0	0	0	0									0.00000161		
in05smr														
4	5	12												
0	0	0										0.00000106		

Fig. 5.11 Input data for test case 1 (TWSMR)

0.0	2.6053	5.2106	7.8159	10.4212	13.0264	00003000
15.6317	18.2370	20.8423	23.4476	24.0	25.6347	00003100
26.1697	26.9335	27.1347	27.6602	27.9212	28.4347	00003200
29.2303	29.5057	30.1847	30.8847	31.5847	32.2637	00003300
32.5391	33.3347	33.8482	34.1092	34.6347	34.8359	00003400
35.5997	36.1347	37.0	38.0	39.0	40.0	00003500
41.0	41.8347	42.3697	43.1335	43.3347	43.8602	00003600
44.1212	44.6347	45.4303	45.7057	46.3847	47.0847	00003700
47.7847	48.4637	48.7391	49.5347	50.0482	50.3082	00003800
50.8347	51.0359	51.7997	52.3347	53.0	54.7112	00003900
57.3165	59.9218	62.5270	65.1324	67.7377	70.3430	00004000
72.9482	75.5535	78.1588	80.7641	83.3694		00004100
6	1 42	1 70				00004200
0	0 0	0 0				00004300
5	0 0	0 0				00004400
1	4 9	9				00004500
1	17	10 12				00004600
1	25	13 17				00004700
1	33	18 22				00004800
1	34	23 57				00004900
1	33	58 59				00005000
1	25	60 60				00005100
1	17	61 61				00005200
1	4	62 62				00005300
0	0 0	0 0				00005400
0	0 0	0 0				00005500
4	1 5	11 11				00005510
1	7	12 12				00005520
1	11	13 15				00005530
1	12	16 17				00005540
1	14	18 25				00005550
1	12	26 27				00005560
1	11	28 30				00005570
1	7	31 31				00005580
1	5	32 32				00005590
0	0 0	0 0				00005591
0	0 0	0 0				00005592
3	18	25 37 37				00005600
16	27	38 38				00005700
12	30	39 41				00005800
11	31	42 43				00005900
9	32	44 51				00006000
11	31	52 53				00006100
12	30	54 56				00006200
16	27	57 57				00006300
18	25	58 58				00006400
0	0 0	0 0				00006500
0	0 0	0 0				00006600
2	19	24 38 38				00006700
17	26	39 39				00006800
14	29	40 42				00006900
12	30	43 44				00007000
11	31	45 50				00007100
12	30	51 52				00007200
14	29	53 55				00007300
17	26	56 56				00007400
19	24	57 57				00007500
0	0 0	0 0				00007600
0	0 0	0 0				00007700

Fig. 5.12 Input data for test case 2 (TWSMR) (4/4)

1	20	23	41	41	00007800
18	25	42	43	00007900	
16	27	44	45	00008000	
15	28	46	49	00008100	
16	27	50	51	00008200	
18	25	52	53	00008300	
20	23	54	54	00008400	
0	0	0	0	00008500	
0	0	0	0	00008600	
in04smr					
1	4	9	9		
1	17	10	12		
1	25	13	17		
1	33	18	22		
1	34	23	57		
1	33	58	59		
1	25	60	60		
0	0	0	0		
in05smr					
4	7	12			00000106

Fig. 5.12 Input data for test case 2 (TWSMR) (3/4)

```

*** PROGRAM MAIN ***
C*pp# B extension of size of 'container array A' 70000 to 120000
COMMON IA ( 250 ), A (120000 )
C.NK COMMON IA ( 250 ), A (70000 )
C*pp# E

C*pp# B extension of size of 'container array A' 70000 to 120000
MAXLEN =120000
C.NK MAXLEN =70000
C*pp# E

*** SUBROUTINE INPTIL ***

C*pp# B expansion A block to 120000
C.450 FORMAT (/RHOSTORAGE,4X,8HREQUIRED,8H ALLOWED/11H SMALL CORE,2(2X,00006190
C.116)/11H LARGE CORE,2(2X,16))
450 FORMAT (/RHOSTORAGE,4X,8HREQUIRED,1X,8H ALLOWED/11H SMALL CORE,
1 2(2X,17)/11H LARGE CORE,2(2X,17))
C*pp# E
  
```

Fig. 5.13 Example of work area extension for container array A

```

100 CONTINUE
...
IITNO=IITNO+1
ERR=0.0
DO 390 J=2,JT
DO 390 I=2,IT
380 ERR=AMAX1(ERR,ABS((FLUXA(I-1,J-1)+FLUXA(I-1,J)+FLUXA(I,J-1)+FLUXA(I,J))/7))
390 CONTINUE
IF (IITNO.GT.IITL) GO TO 440
IF (ERR.GT.EPS1) GO TO 100
  
```

Fig. 5.14 Calculation part of IITNO in subroutine INNER

```

*** M78.0におけるFLUXA(I,J)値 ***
0.160834473E-02 0.161247305E-02 0.161937950E-02 0.162796398E-02 0.163869862E-02 0.165073457E-02 0.165049918E-02
0.162241782E-02 0.166338476E-02 0.168314530E-02 0.167354496E-02 0.168614541E-02 0.167730064E-02
0.167719426E-02 0.168290338E-02 0.169562025E-02 0.169562025E-02 0.170269888E-02 0.170561089E-02
0.171536999E-02 0.172588444E-02 0.173198324E-02 0.173198324E-02 0.175393348E-02
0.176221575E-02 0.176856015E-02 0.176990009E-02 0.176990009E-02 0.178486140E-02
0.17687562E-02 0.178374909E-02 0.17856015E-02 0.17856015E-02 0.178393348E-02
0.176360097E-02 0.176166324E-02 0.176166324E-02 0.174046424E-02
0.176365212E-02 0.174737895E-02 0.174737895E-02 0.174046424E-02
0.171046983E-02 0.164126314E-02 0.153864210E-02 0.153864210E-02
0.140331895E-02 0.124184659E-02 0.104928471E-02 0.104928471E-02
0.830563428E-03 0.590559864E-03 0.311031094E-03 0.311031094E-03
0.300829349E-02 0.301527046E-02 0.302186259E-02 0.302186259E-02
0.303425788E-02 0.304606666E-02 0.306653526E-02 0.306653526E-02
0.306715630E-02 0.307887932E-02 0.308889827E-02 0.308889827E-02
0.309206293E-02 0.310307811E-02 0.311434339E-02 0.311434339E-02
0.311944215E-02 0.312829227E-02 0.31466945E-02 0.31466945E-02
0.315708503E-02 0.317710149E-02 0.321966037E-02 0.321966037E-02
0.324966677E-02 0.326203904E-02 0.327933230E-02 0.327933230E-02
0.329670494E-02 0.330962664E-02 0.331759861E-02 0.331759861E-02
0.332819301E-02 0.333693064E-02 0.334270881E-02 0.334270881E-02
0.334521478E-02 0.334641268E-02 0.334510836E-02 0.334510836E-02
0.333915859E-02 0.333287101E-02 0.331675145E-02 0.331675145E-02
0.325406133E-02 0.312870299E-02 0.294914359E-02 0.294914359E-02
0.269868232E-02 0.238668011E-02 0.201675063E-03 0.201675063E-03
0.156931707E-02 0.139634828E-02 0.586216647E-03 0.586216647E-03
0.415680092E-02 0.415686868E-02 0.416311994E-02 0.416311994E-02
0.416477304E-02 0.416546942E-02 0.416673720E-02 0.416673720E-02
  
```

```

0.175624143E-02 0.157073955E-02 0.133822777E-02 0.133822777E-02
0.106762140E-02 0.764137658E-03 0.401653233E-03 0.401653233E-03
0.138681859E-02 0.139036763E-02 0.139461120E-02 0.139461120E-02
0.139952102E-02 0.140423862E-02 0.140927994E-02 0.140927994E-02
0.141474383E-02 0.142129324E-02 0.142533332E-02 0.142533332E-02
0.142561470E-02 0.143595012E-02 0.143753190E-02 0.143753190E-02
0.144035486E-02 0.144318026E-02 0.144956994E-02 0.144956994E-02
0.145351957E-02 0.146588846E-02 0.148027653E-02 0.148027653E-02
0.150241167E-02 0.150884804E-02 0.151384785E-02 0.151384785E-02
0.152291649E-02 0.153129699E-02 0.153874268E-02 0.153874268E-02
0.154259126E-02 0.154511782E-02 0.154827267E-02 0.154827267E-02
0.155089911E-02 0.155416992E-02 0.154684111E-02 0.154684111E-02
0.155294465E-02 0.155146210E-02 0.154684111E-02 0.154684111E-02
0.152133836E-02 0.148977056E-02 0.139270304E-02 0.139270304E-02
0.128133327E-02 0.113943405E-02 0.966519304E-03 0.966519304E-03
0.770179191E-03 0.550153956E-03 0.288116047E-03 0.288116047E-03
0.744315711E-03 0.745384492E-03 0.748003135E-03 0.748003135E-03
0.753189030E-03 0.758777314E-03 0.763386532E-03 0.763386532E-03
0.768763304E-03 0.770819082E-03 0.773587709E-03 0.773587709E-03
0.776026398E-03 0.779292895E-03 0.783672269E-03 0.783672269E-03
0.787476136E-03 0.787144554E-03 0.791327679E-03 0.791327679E-03
0.798139547E-03 0.798553551E-03 0.802066934E-03 0.802066934E-03
0.805951247E-03 0.809041962E-03 0.815952489E-03 0.815952489E-03
0.817859487E-03 0.824575581E-03 0.826637856E-03 0.826637856E-03
0.821827096E-03 0.824281015E-03 0.823400624E-03 0.823400624E-03
0.822589906E-03 0.820193219E-03 0.819980458E-03 0.819980458E-03
0.818578934E-03 0.815971600E-03 0.812718855E-03 0.812718855E-03
0.802050245E-03 0.773215608E-03 0.728206924E-03 0.728206924E-03
0.667268954E-03 0.593182747E-03 0.50388877E-03 0.50388877E-03
0.399749872E-03 0.285047339E-03 0.150520325E-03 0.150520325E-03
  
```

Fig. 5.15 Example of comparison of FLUXA(I,J) (IITNO=7, G=6, IITNO=19)

6. IPLOT の移植調査

6.1 はじめに

IPLOT は、伝熱流動研究室で開発した、汎用図形処理解析プログラムである [8]。本プログラムの主な特徴は、インタラクティブに図形の描画を行うことが可能であることと、データ解析用の各種サブルーチンが実装されていること、更に、ユーザ自身で作成したサブルーチンを対話的に呼び出して使用することが可能となっていることである。

現在、情報システム管理課に設置されている M780 上で利用されているが、昨今の計算機環境の変化に伴い、UNIX OS 環境への移植が必要となった。そこで、M780 から UNIX OS 環境へのプログラムの移植にあたり、必要となる変換項目について調査を行った。なお、対象とする UNIX OS は、SunOS 4.1.3 とする。

以下、その報告である。

6.2 IPLOT について

ここでは、IPLOT について簡単に紹介する。IPLOT は、二相流動試験データ処理解析を目的として、伝熱流動研究室で開発した、汎用図形処理解析プログラムである [8]。本プログラムは、対話的に処理を進めていくことで、図形の描画を行うことが可能となっており、そのため、様々な内部コマンドが用意されている。また、データ解析を行うためのサブルーチン群が整備されており、内部コマンドを介して、これらのサブルーチンを任意にコールすることができる。さらに、必要に応じて、ユーザが独自のサブルーチンを作成し、これを利用することも可能となっている。

IPLOT の起動手順は次のようになっている。

1. SYSPRINT を端末に割り当てる。
2. 図形の出力機番を割り当てる。
3. ライブラリを組み込む。
4. ロードモジュールの実行。

なお、ここで組み込むライブラリは、前述したデータ解析用のサブルーチン群を意味している。また、ロードモジュールは、図形の出力先に応じて、図形端末用、NLP 出力用、テクトロニクス端末用等に別れているため、出力先に応じたロードモジュールを使用する。これらロードモジュールの違いは、作成時にリンクされるライブラリが異なっているだけである。

これら一連の手順は、コマンドプロシジャを使用して行われる。IPLOT を実行する際のコマンドプロシジャを Fig. 6.1 に示す。

IPLOT の操作は、対話的に行われる。図形を出力するまでの基本的な手順は、次の通りである。

1. データが格納してあるファイルを設定。
2. データの読み込み。

3. データの作図.

4. IPLOT の終了.

ここで、データが格納されているファイルは、標準ファイルと呼ばれるもので、あらかじめ定められたフォーマットにしたがって、試験データが格納されている。

これらの操作は、IPLOT の内部コマンドを使用して行われる。また、必要であれば、データ解析用のサブルーチン呼び出しを利用して利用することも可能である。サブルーチンの呼び出しは、NUSER コマンドを使用して行われる。コマンド NUSER に続けて、使用するサブルーチン名と、パラメータを入力する。これにより、起動時に組み込んだライブラリから、指定したサブルーチンが呼び出される。

この他、起動から一連の作業手順をバッチジョブで実行することも可能である。この場合、実行する内部コマンドは SYSIN からの入力となる。Fig. 6.2 にバッチジョブとして実行する場合の JCL 例を示す。

6.3 移植における問題点

ここでは、IPLOT を SunOS 上に移植するにあたり、プログラムの修正が必要となる項目について述べる。調査を行ったソースファイル一覧を Table 6.1 に示す。

また、IPLOT の図形端末用ロードモジュールを作成する際の JCL 例を Fig. 6.3 に示す。ここで、SYSLIB で指定されている 'SYS9.CGSLGSP.LOAD' が描画した図形を図形端末に出力するためのライブラリである。描画した図形を NLP への出力用にロードモジュールを作り換える場合には、'SYS9.CGSLGSP.LOAD' の部分を 'SYS9.PNL.LOAD' に、テクトロニクス端末用にする場合は、'SYS9.PTS.LOAD' に、それぞれ変更すればよい。

IPLOT を SunOS 用に交換する際に問題となる点を述べると、まず第一に、アセンブラ言語を使用していることが挙げられる。また、Fig. 6.3 に示した JCL 例から判るように、ソースプログラム 'IPLOTV95.FOT77(ALLOC)' をコンパイルする際に、オプション LANGLVL(66) が指定されていることから、FORTRAN66 仕様の文法が用いられている可能性があること、更に、SYSLIB で指定されたライブラリ中のサブルーチンが使用されている場合、これと同等な機能を持つサブルーチンを用意する必要があることである。

以下、これらの問題点について順を追って説明する。

6.3.1 アセンブラ言語ルーチンの使用

IPLOT では、機能実現のためアセンブラ言語を使用している。SunOS 上への移植にあたっては、これらの部分の書き換えを行う必要がある。ソースファイル 'IPLOTV95.ASM' 中に含まれているサブルーチンのうち、実際に使用されているのは Table 6.2 に示した 2 つのルーチンである。

GDRTY ルーチンは、区分データセットのメンバ名を表示する機能を持っている。IPLOT では、ライブラリとして組み込まれた使用可能サブルーチン一覧の表示や、試験データの一覧表示等を行う場合に使用されている。

LINK ルーチンは、動的リンク機能を提供しているもので、IPLOTでは、データ解析サブルーチンを呼び出す際に使用されている。データ解析サブルーチンはライブラリファイルとして構築されており、IPLOTのロードモジュールとは独立して存在する。したがって、これらのサブルーチンを使用する場合には、動的に結合するための処理が必要となる。

上記2つのルーチンは、スーパーバイザコール(SVC)を利用して記述されている。このスーパーバイザコールは、OSのサービスファンクションを利用する際に用いられるものであるため、アセンブラ言語を用いて記述されている。したがって、これらのサブルーチンの持つ機能をSunOS上で実現する際には、新たにプログラムを作成する必要がある。

6.3.2 FORTRAN66 言語仕様の利用

Fig. 6.3に示したJCL例から、ソースファイル'IPL0TV95.F0T77'中にある、ALLOCルーチンをコンパイルする際に、オプションLANGVL(66)を指定していることから、FORTRAN66仕様の文法を用いている可能性があることが判る。そこで、ALLOCルーチンを調べてみたところ、内部ファイル入出力を行うために、ENCODE文、DECODE文を使用していることが判った。しかし、ALLOCルーチンは現在使用されていないことが、IPLOT使用説明書から判明したため、変換の必要はないと判断した。

6.3.3 M780固有ライブラリルーチンの使用

Fig. 6.3に示したJCL例より、SYSLIBで指定されたライブラリ内に含まれるサブルーチンを利用している場合、それらのサブルーチンがSunOS上に存在しないと、リンク時に未定義エラーとなってしまふ。そこで、試しにSunOS上でコンパイル・リンクを行ってみたところ、未定義となったサブルーチンが見つかった。Table. 6.3に、未定義となったサブルーチンの一覧を示す。これらは、SSL2, JSSL, IPFライブラリ中に含まれているもので、一般に、SunOS上には存在しない。したがって、SunOSへの移植を行うためには、これらの未定義となるサブルーチンを、どのように提供するか検討が必要となる。

また、この他にもカルコン仕様様の図形処理ライブラリ[9]を使用していることが挙げられるが、この件については後述する。

6.3.4 その他

前述したGDRTYルーチンでは、引数にMSP形式のデータセット名を渡している。しかし、MSPとSunOSでは、ファイルシステムが異なっているため、FORTRANソースプログラム中に記述されたデータセット名は、正しく解釈されない。また、同様の記述は、OPEN文の引数にも見られる。したがって、ソースプログラム中の該当部分をUNIXファイルシステムに合わせた形式で記述しなおす必要がある。

6.4 解決策の提案

前述したように、変換時に問題となるのは、アセンブラ言語で記述された2つのサブルーチンと、SunOS上に存在しないライブラリの利用についてである。ここでは、これらの問題点についての解決案を述べる。

6.4.1 アセンブラ言語ルーチンの書き換え

アセンブラ言語を用いて記述されたサブルーチン LINK の機能を実現するための方法として、共用ライブラリを作成し、実行時にこれ呼び出してサブルーチンのダイナミックリンクを行う方法がある。この機能を提供するためのインターフェイス・ルーチンを C 言語を用いて作成する。Fig. 6.4, 6.5, 6.6に簡単なプログラム例を示す。

Fig. 6.4は FORTRAN で記述されたサブルーチンである。これをコンパイルし、共用ライブラリを作成する。共用ライブラリを作成する際のコマンドを Fig. 6.7に示す。Fig. 6.5はメインルーチンである。ここで、サブルーチン SAMPLE の引数として指定した文字列が、共用ライブラリ中にあるサブルーチン名である。Fig. 6.6は、ダイナミックリンク処理を行うためのルーチンで、C 言語を用いて記述されている。なお、FORTRAN と C 言語との間で処理をやり取りするに際し、サブルーチンの命名規約や、引数の渡し方については SunOS の文法に従って記述した。

このサンプルプログラムを実行した結果を Fig. 6.8に示す。

次に、GDRTY ルーチンについてであるが、このサブルーチンの機能は、区分データセット内のメンバ名を表示するものであることは既に述べた。この区分データセットは、UNIX 環境ではディレクトリファイルとみなすことができるので、該当するディレクトリ内に存在するファイル名を取得するルーチンを、C 言語を用いて記述すればよい。Fig. 6.9にサンプルプログラムを示す。

しかし、IPLOT では、この GDRTY ルーチンを、使用可能なデータ解析サブルーチンの一覧表示にも使用している。M780 では、ライブラリファイルも区分データセットに構築され、各サブルーチンは、そのメンバファイルとなるため、特に区別する必要はなかった。しかし、前述したように、データ解析サブルーチンを共用ライブラリとしたことで、各サブルーチンの一覧を取得するためには、このライブラリからシンボル名を読み出す処理を行わなければならない。このためのサンプルプログラムを Fig. 6.10に示す。

このことから、GDRTY ルーチンは、異なる機能を持つ 2 つのサブルーチンに分割して作成する必要がある。

6.4.2 M780 固有ライブラリルーチンの利用

前述したように IPLOT では、M780 上に存在する複数のライブラリルーチンを使用している。これらのサブルーチンは、'SYS9.SSL2.LOAD', 'SYS9.JSSL.LOAD', 'SYS1.IPFLIB' 内にそれぞれ含まれており、一般に SunOS 上には存在しないものである。

従って、IPLOT を SunOS 上に移植する場合、これらのサブルーチンを新たに作成するか、または、他の方法を検討しなければならない。

移植に当たり、これらと同等の機能を持つサブルーチンを作成することも一つの方法ではあるが、もし、SunOS 上で利用できるものが存在するのであれば、これを使用するのが最も簡単な方法であると思われる。

科学計算用サブルーチンパッケージ SSL2 ライブラリについては、SunOS 用のものが市販されている。したがって、ライセンスを購入し、これを利用することが最も簡単である。また、JSSL ライブラリについては、やはり、科学計算用サブルーチン群であるが、これは、原研で開

発されたもので、市販されてはいない。しかし、原研所内であれば、許可を得てソースプログラムを入手することが可能であるので、これを利用すれば良い。

最後に、TSSCMD ルーチンであるが、これは、IPLOT 内から TSS コマンドを発行する際に利用されている。この機能を UNIX で実現しようとする場合には、サービス関数 `system` 等を使用して、プログラムを記述することができるが、昨今では、マルチウィンドウ端末の普及により、この機能を実現することに特に有効性を見出すことはできない。したがって、TSSCMD ルーチンの利用を廃止しても、問題は無いと考える。

6.4.3 図形処理ライブラリの使用について

IPLOT では、図形の最終的な出力先として、各種図形端末や NLP を指定することができる。しかし、これらの出力デバイスは M780 に接続されているもので、一般のワークステーションからアクセスすることはできない。

IPLOT を SunOS 上に移植する場合、図形の出力先としては、やはり X Window System を利用するのが一般的であると思われる。そこで、カルコンプ仕様の図形処理ライブラリで、出力デバイスに X Window System を使用できるものが存在するのであれば、それを利用するのが簡単である。

X Window System 版カルコンプライブラリは、いくつか市販されているものがあるので、それらを購入して利用するか、または、情報システム管理課で作成した、カルコンプ互換ライブラリ `piflib[10]` を利用することで、図形処理ライブラリに関する問題を解決することができる。

6.5 まとめ

M780 環境から SunOS 環境への移植を行うにあたっては、これまでに述べてきたように、OS に依存する処理の解決が問題となる。しかし、対話的に処理を行えるというプログラムの性格上、実行環境の違いによる単なる移植に留まらず、機能的な部分や使い勝手の部分に関して、まだまだ検討する余地が残っているものと考えられる。

したがって、移植作業を行うにあたっては、ユーザと密接にコミュニケーションを保ちつつ、修正の都度、随時確認を取りながら作業を進めていくことが有効であると考えられる。

このことから、作業期間が長期にわたるものと判断し、作業自体はユーザの方で行って頂くこととした。本報告書が変換の際の手助けとなれば幸いである。

Table 6.1 List of source files

データセット名	内容
IPL0TV95.ASM	アセンブラソース
IPL0TV95.DUMMY.FORT77	ダミールーチン
IPL0TV95.FORT77	コマンドソース他
IPL0TV95.INC	インクルードファイル
IPL0TV95.MAIN.FORT77	メインルーチン
IPL0TV95.SUB.FORT77	サブルーチンソース

Table 6.2 List of assembly routines

ルーチン名	機能
GDRTY	メンバ名の表示
LINK	プログラムの動的リンク

Table 6.3 Unresolved routines

ルーチン名	機能	ライブラリ名
LESQ1	最小2乗近似多項式	SSL2
SPLV	3次 spline 補間式による補間・微分	SSL2
INTRPL	Akima の方法による内挿補間	JSSL
TSSCMD	TSS コマンドの発行	IPF

```

PROC O L(O)
CONTROL NOL
WRITE *****
WRITE *
WRITE * IPLT.V9SPTS.LOAD *
WRITE *
WRITE *****
FREE F(FT06F001)
FREE F(FT08F001)
ATTR II LR(130)
ALLOC DA(*) US(II)
/*ALLOC DA(IPLTV95.FT06) F(FT06F001) US(II)
FREE ATTR(II)
ALLOC F(FT06F001) DA(*) REU
ALLOC F(FT08F001) DA(*) REU
IF &L.=&STR(O) THEN DO
LIB ('J2321.IPLTLM.LOAD')
WRITE ** 'J2321.IPLTLM.LOAD' WAS REFERRED **
END
ELSE DO
WRITE ** PRIVATE LIB '&L..LOAD' AND
WRITE 'J2321.IPLTLM.LOAD' WERE REFERRED **
LIB (&L..LOAD 'J2321.IPLTLM.LOAD')
WRITE ** IF OK, PLEASE PRESS ANY KEY **
WRITE ** IF NO, PLEASE PRESS "N" KEY **
READ KEY
WRITE &STR(&KEY)
IF &STR(&KEY) EQ 'N' THEN EXIT
END
.PTSIO
/*ALLOC F(FT85F001).DA(*) REU
/*ALLOC F(FT86F001).DA(*) REU
.CLDISP
CALL 'J2321.IPLTV95.PTS.LOAD(TEMPNAME)'
FREE F(FT08F001)
LIB
EXIT
    
```

Fig. 6.1 Command procedure for execution

```

T(02) W(02) C(04) I(02) GRP
/******
/* J2321.IPLTV95.CNTL(IPV95GD)
/******
//IPLT EXEC LMGO,LM='J2321.IPLTV95',Q='LOAD',PMM=NLP
//STEPLIB DD
// DD UNIT=D0930,DSN=J2321.IPLTLM.LOAD,DISP=SHR
// DD DSN=SYS1.IPLLIB,DISP=SHR.
//FT06F001 DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
//SYSIN DD *
( IPLT コマンド群)
/*
//NLP EXPAND GRNLP,SYSDOUT=N
//FT09F001 DD SYSOUT=*,DCB=(RECFM=FA,BLKSIZE=133)
//FT08F001 DD DUMMY
//CMDIN DD DUMMY
//CMDOUT DD SYSOUT=*
    
```

Fig. 6.2 JCL for execution

```

T(O4) I(O4) W(O1) C(O3)
/* =====
** J9136.IPLOTV95.CNTL(MKLCX)
** FOR MAKING IPLOT-LOAD-MODULE FOR CON
** =====
//FORT1 EXEC FORTEX,SO='J9136.IPLOTV95.MAIN',Q=' .FORT1',
// A='ELM(TSS,BLOCKD),NOS'
//SYSINC DD DSN=J9136.IPLOTV95.INC,DISP=SHR
//SYSPRINT DD DUMMY
//FORT2 EXEC FORTEX,SO='J9136.IPLOTV95',Q=' .FORT2',
// A='ELM(ALLOC),NOS,LANGVL(66)',DISP=MOD
//SYSINC DD DSN=J9136.IPLOTV95.INC,DISP=SHR
//SYSPRINT DD DUMMY
//FORT3 EXEC FORTEX,SO='J9136.IPLOTV95',Q=' .FORT3',
// A='ELM(*),NOS',DISP=MOD
//SYSINC DD DSN=J9136.IPLOTV95.INC,DISP=SHR
//SYSPRINT DD DUMMY
//FORT4 EXEC FORTEX,SO='J9136.IPLOTV95.SUB',Q=' .FORT4',
// A='ELM(*),NOS',DISP=MOD
//SYSINC DD DSN=J9136.IPLOTV95.INC,DISP=SHR
//SYSPRINT DD DUMMY
//FORT5 EXEC FORTEX,SO='J9136.IPLOTV95.DUMMY',Q=' .FORT5',
// A='ELM(RTLOT),NOS',DISP=MOD
//SYSINC DD DSN=J9136.IPLOTV95.INC,DISP=SHR
//SYSPRINT DD DUMMY
//ASM1 EXEC ASM,SO='J9136.IPLOTV95',Q=' .ASN',
// A='OBJ,BATCH,NOLIST',DISP=MOD
//SYSPRINT DD DUMMY
//MKLCX95 EXEC PGM=JAL,REGION=1024K,COND=(4,LT),
// PARM='MAP,LIST,LET'
//
// --- Continue ---
    
```

Fig. 6.3 JCI for compilation and linkage (1/2)

```

//SYSLIB DD DSN=SYS9.GGS.LOAD,DISP=SHR
// DD DSN=SYS9.CGSLGSP.LOAD,DISP=SHR
// DD DSN=SYS9.PNL.LOAD,DISP=SHR
// DD DSN=SYS2.IPFILIB,DISP=SHR
// DD DSN=SYS9.JSSL.LOAD,DISP=SHR
// DD DSN=SYS9.SSL2.LOAD,DISP=SHR
// DD DSN=SYS2.FORTLIB,DISP=SHR
//SYSPRINT DD SYSOUT=* ,DCB=(BLKSIZE=4840)
//SYSTEM DD SYSOUT=*
//SYSUT1 DD UNIT=VIO,SPACE=(TRK,(30,10))
//SYSLMOD DD DSN=J9136.IPLOTV95.CON.LOAD,UNIT=D0340,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(30,10,30),RLSE)
//SYSLIN DD DSN=OBJ,DISP=(OLD,DELETE)
// DD DDNAME=SYSIN
//SYSIN DD DUMMY
    
```

Fig. 6.3 The JCI for compilation and linkage (2/2)

```

*****
pai.f
*****
SUBROUTINE PAI( R )
R = ACOS( -1.0 )
RETURN
END
*****
double.f
*****
SUBROUTINE DOUBLE( R )
R = R*2
RETURN
END
    
```

Fig. 6.4 Sample program for dynamic linkage (subroutines)

```

*****
main.f
*****
CALL SAMPLE( 'pai', R )
WRITE(6,*) R = ', R
CALL SAMPLE( 'double', R )
WRITE(6,*) R = ', R
C
STOP
END
    
```

Fig. 6.5 Sample program for dynamic linkage (main routine)

```

*****
sample.c
*****
#include <stdio.h>
#include <errno.h>
#include <dlfcn.h>

extern int errno;

sample_( name, r, n )
float *r;
char *name;
int n;
{
    int flags;
    void (*value)() = NULL;
    void *handle;

    name[n] = NULL;
    if( NULL == ( handle = dlopen( "libsub.so", 1 )) ){
        fprintf( stderr, "Can not Load Shared Library\n" );
        exit(0);
    }

    printf( "%s\n", name );
    if( NULL == ( value = dlsym( handle, name )) ){
        fprintf( stderr, "Can not find symbols\n" );
        fprintf( stderr, "  errno = %d\n", errno );
        dlclose( handle );
        exit(0);
    }

    (*value)( r );
    dlclose( handle );
    return 0;
}
    
```

Fig. 6.6 Sample program for dynamic linkage (core)

```
% ld -o libsub.so -Bdynamic pai.o double.o
```

Fig. 6.7 Command of shared library making

```
R = 3.14159
R = 6.28319
```

Fig. 6.8 Results of sample program

```
#include <stdio.h>
#include <dirent.h>
main(argc,argv)
int argc;
char *argv[];
{
    DIR *dirp;
    struct dirent *d;

    if( NULL == (dirp = opendir(argv[1]))){
        fprintf(stderr, "%s No such file or Directory.\n", argv[1] );
        return( NULL );
    }

    while( ( d = readdir( dirp ) ) != NULL ){
        fprintf( stdout, "%s\n", d->d_name );
    }
    closedir( dirp );
}
```

Fig. 6.9 Sample program for files list making

```
#include <stdio.h>
#include <string.h>
#include <a.out.h>

main(argc,argv)
int argc;
char *argv[];
{
    FILE *fp;
    struct exec ex;
    struct nlist nl[50];
    char buf[128],*ip;
    int nsize,nitem=0,i=0;

    ip=buf;
    if( NULL == (fp=fopen(argv[1],"r"))){
        fprintf(stderr, " open NG\n " );
        exit(0);
    }
    fread(&ex,sizeof(ex),1,fp);
    fseek(fp,N_SYNOFF(ex),0);
    nitem=ex.a_syms/sizeof(struct nlist);
    fread(nl,sizeof(nl),1,fp);

    fseek(fp,N_STROFF(ex),0);
    fread(&nsize,sizeof(nsize),1,fp);

    fseek(fp,N_STROFF(ex),0);
    fread(buf,sizeof(char),nsize+4,fp);

    for(i=0;i<nitem;i++){
        if(nl[i].n_type==N_TEXT){
            fprintf(stdout,"%s\n",ip+nl[i].n_un.n_stxx);
        }
    }
}
```

Fig. 6.10 Sample program for subroutines list making

7. おわりに

計算科学技術推進センター情報システム管理課では、平成6年度末に導入したVPP500の利用を推進するために、ユーザからの依頼された原子力コードをVPP500に移植する作業を行ってきた。この作業は、米国でCRAYやIBM等の計算機用に開発された原子力コードを原研のVPP500で実行できるようにするもので、コンパイラ固有のサービスルーチンや指示行の変更、アーキテクチャの違いによる計算結果の精度チェック等が大きな仕事である。

一般にコードの移植作業は、比較的単純な作業と見られがちであるが、原研で扱われる大規模な原子力コードの移植においては、計算機システムについての知識のみならず、コード特有のクセまでも知っていないと簡単に移植できないこともある。

本報告書では、原子力コードをVPP500に移植する際に障害となった様々な問題点について、ひとつひとつ解決していった過程をできる限り詳細に記述している。本報告書が、今後大規模な原子力コードの移植に携わる人々に多少とも参考になれば幸いである。

謝 辞

本作業を行う上で、作業を依頼された原子炉安全工学部熱水力安全研究室の浅香英明氏(2章, 3章), 原子炉工学部高エネルギー中性子研究室の今野力氏(4章), 高温工学試験研究炉開発部HTTR技術開発室の野尻直喜氏(5章), 原子炉工学部伝熱流動研究室の井口正氏(6章)には、コード内容の把握について御協力頂きました。また、VPP500システムのハードウェア、ソフトウェアに関する相談については(株)富士通R&Dシステム部石附茂氏に、本作業を円滑に遂行するための各種事務処理については山田圭子氏に御協力を頂きました。ここにこれらの方々に感謝の意を表します。

最後に、本報告書を執筆する機会を与えて下さいました計算科学技術推進センター長浅井清氏、前次長秋元正幸氏(現原子炉安全工学部部長)、(株)富士通R&Dシステム部前部長橋本道夫氏及び現部長平沢健一氏に感謝致します。

7. おわりに

計算科学技術推進センター情報システム管理課では、平成6年度末に導入したVPP500の利用を推進するために、ユーザからの依頼された原子力コードをVPP500に移植する作業を行ってきている。この作業は、米国でCRAYやIBM等の計算機用に開発された原子力コードを原研のVPP500で実行できるようにするもので、コンパイラ固有のサービスルーチンや指示行の変更、アーキテクチャの違いによる計算結果の精度チェック等が大きな仕事である。

一般にコードの移植作業は、比較的単純な作業と見られがちであるが、原研で扱われる大規模な原子力コードの移植においては、計算機システムについての知識のみならず、コード特有のクセまでも知っていないと簡単に移植できないこともある。

本報告書では、原子力コードをVPP500に移植する際に障害となった様々な問題点について、ひとつひとつ解決していった過程をできる限り詳細に記述している。本報告書が、今後大規模な原子力コードの移植に携わる人々に多少とも参考になれば幸いである。

謝 辞

本作業を行う上で、作業を依頼された原子炉安全工学部熱水力安全研究室の浅香英明氏(2章, 3章), 原子炉工学部高エネルギー中性子研究室の今野力氏(4章), 高温工学試験研究炉開発部HTTR技術開発室の野尻直喜氏(5章), 原子炉工学部伝熱流動研究室の井口正氏(6章)には、コード内容の把握について御協力頂きました。また、VPP500システムのハードウェア、ソフトウェアに関する相談については(株)富士通R&Dシステム部石附茂氏に、本作業を円滑に遂行するための各種事務処理については山田圭子氏に御協力を頂きました。ここにこれらの方々に感謝の意を表します。

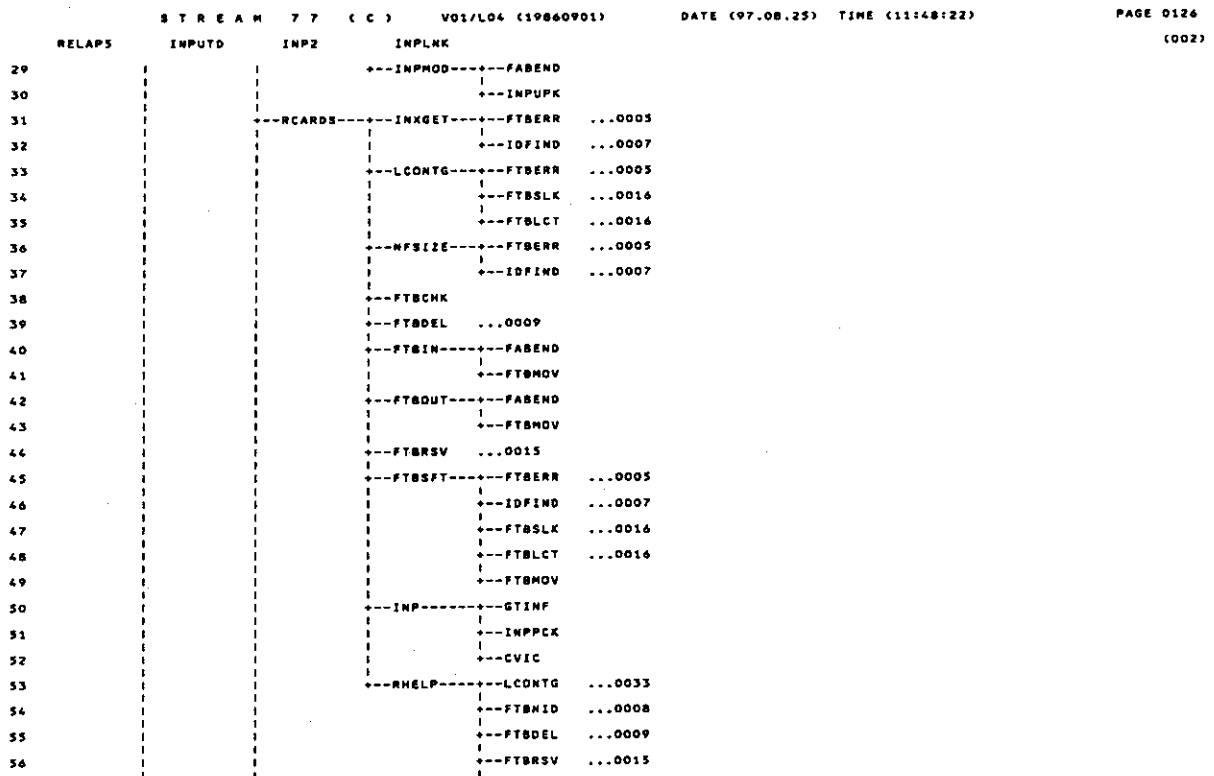
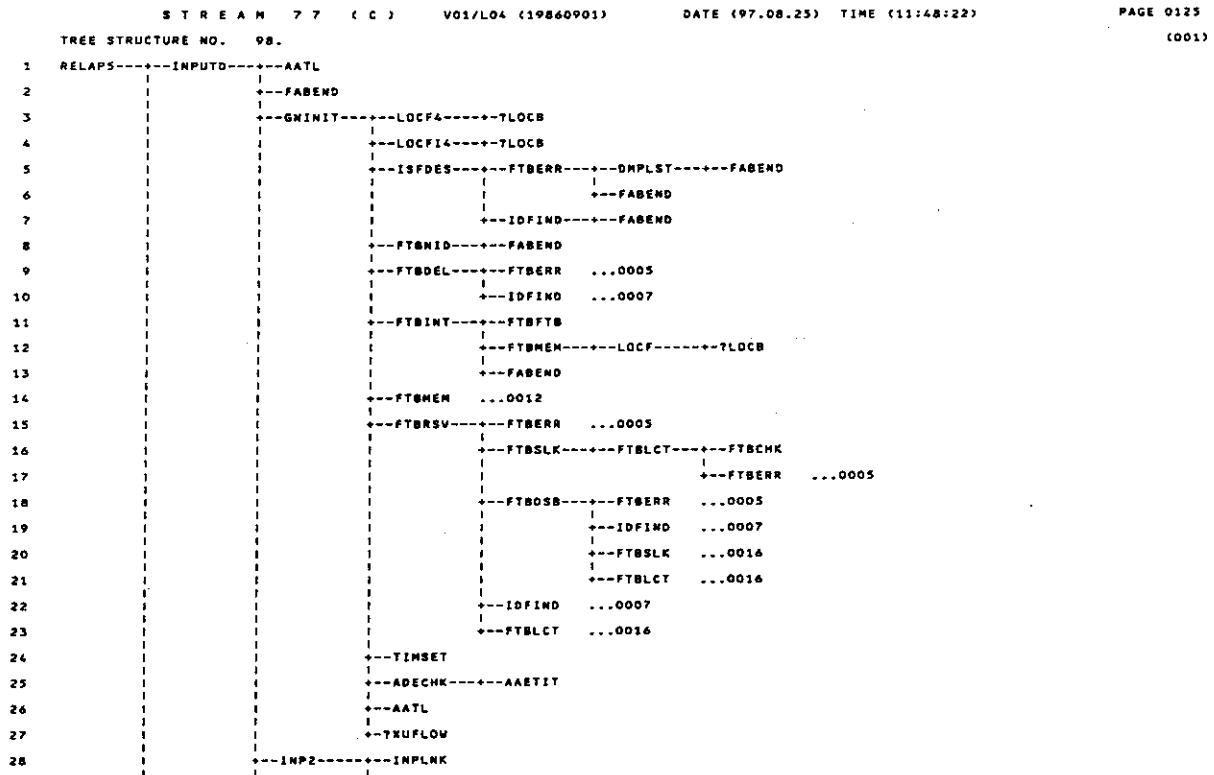
最後に、本報告書を執筆する機会を与えて下さいました計算科学技術推進センター長浅井清氏、前次長秋元正幸氏(現原子炉安全工学部部長)、(株)富士通R&Dシステム部前部長橋本道夫氏及び現部長平沢健一氏に感謝致します。

参考文献

- [1] V.H.Ransom.et al.; RELAP5/MOD2 Code Manual, Vol.1,Vol.2.BGG-SAAM-6337. INEL(Apr.1984)
- [2] 根本俊行, 石黒美佐子; 軽水炉安全解析コード RELAP5 の変換手順書 - IBM 版から FACOM 版へ -, 私信, 1990年3月
- [3] FACOM OS IV/F4 MSP STREAM77 説明書 富士通株式会社, 1992年5月
- [4] 安濃田良成, 細山田龍二, 島根由起夫, 浜田邦靖, 会話型汎用グラフィック・プロセッサ - ROPS 2.0 - ユーザーズ・マニュアル, JAERI-M 88-030, 1988年2月
- [5] DISSPLA POCKET GUIDE, CURRENT WITH VERSION 9.0, Integrated Software Systems Corporation, October 1981
- [6] R.E.MacFarlane,D.W.Muir, The NJOY Nuclear Data Processing System,Version 91,December 16,1994.
- [7] K.D.Lathrop and F.W.Brinkley, TWOTRAN-II:An Interfaced,Exportable Version of the TWOTRAN Code for Two-Dimensionnal Transport, Los Alamos National Laboratory,Report La-4848-MS,July 1973.
- [8] 汎用図形処理解析プログラム IPLOT 使用説明書, 1996年2月
- [9] CALCOMP ソフトウェアマニュアル グラフィック COM プログラミング - ベーシックソフトウェア -, 吉沢ビジネスマシズ (株),1972年6月
- [10] 情報システム管理課,piflib 利用手引書,1996年3月

付録 A RELAP5/MOD3.2 のツリー構図

次頁より、RELAP5/MOD3.2 コードのツリー構図を示す。



PAGE 0127
(003)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22)

RELAPS	INPUTD	RCARDS	RHELP	FTBRSV		
57				---FTBSFT	...0045	
58				---INP2	...0028	
59			---RDREDT			
60			---RNEWP	---INPB		
61			---RRSTD	---INP2	...0028	
62			---RRESTF	---LCCF4	...0003	
63				---LCNTGS	---FTBERR ...0005	
64					---IDFIND ...0007	
65				---LCONTG	...0033	
66				---LCONTG	...0033	
67				---FTBNID	...0008	
68				---AATL		
69				---FABEND		
70				---FMVLVR	---ISFDES ...0005	
71					---NFUNIT	---FTBERR ...0005
72					---IDFIND	...0007
73				---ISSFRG	---FTBERR ...0005	
74				---FTBSFT	...0045	
75				---FTBCHK		
76				---FTBDEL	...0009	
77				---FTBEXP	---FTBERR ...0005	
78					---FTBMEM	...0012
79					---FTBMOV	
80					---FTBSLK	...0016
81				---FTBOUT	...0042	
82				---FTBRDC	---FTBSLK ...0016	
83					---FTBERR	...0005
84					---FTBMOV	

PAGE 0128
(004)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22)

RELAPS	INPUTD	RNEWP	RRESTF	FTBRDC	FTBMOV
85					---FTBMEM ...0012
86				---FTBRSV	...0015
87				---FTBSFT	...0045
88				---INP2	...0028
89			---RCHNG	---FTBNID	...0008
90				---FTBDEL	...0009
91				---FTBRSV	...0015
92				---INP2	...0028
93			---INP2	...0028	
94			---RTSC	---INP10	---FABEND
95				---LCONTG	...0033
96				---FTBNID	...0008
97				---FTBDEL	...0009
98				---FTBRSV	...0015
99				---FTBSFT	...0045
100				---INP2	...0028
101			---RSSI	---ISFDES	...0005
102				---LCONTG	...0033
103				---FTBNID	...0008
104				---FTBDEL	...0009
105				---FTBRSV	...0015
106				---FTBSFT	...0045
107				---INP2	...0028
108				---INPLNK	
109			---RMIEDT	---INP10	...0094
110				---ISFDES	...0005
111				---LCONTG	...0033
112				---FTBNID	...0008

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0129

RELAPS	INPUTD	RNEUP	RMIEDT	FTBNID	
113				←--FMVLR	...0070
114				←--FTBDEL	...0009
115				←--FTBRV	...0015
116				←--FTBSFT	...0045
117				←--INP2	...0028
118				←--INPLNK	
119			←--RUSRVR	←--INP10	...0094
120				←--ISFDES	...0005
121				←--LCONTG	...0033
122				←--FTBNID	...0008
123				←--FMVLR	...0070
124				←--FTBDEL	...0009
125				←--FTBRV	...0015
126				←--FTBSFT	...0045
127				←--INP2	...0028
128				←--INPLNK	
129			←--RINTRV	←--LCONTG	...0033
130				←--FTBNID	...0008
131				←--FTBCHK	
132				←--FTBDEL	...0009
133				←--FTBIN	...0040
134				←--FTBMV	
135				←--FTBRV	...0015
136				←--FTBSFT	...0045
137				←--INP2	...0028
138				←--INPLNK	
139				←--SCHREG	←--LOCF ...0012
140				←--FABEND	

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0130

RELAPS	INPUTD	RNEUP	RINTRV	SCHREG	FABEND
141					←--RMBLNK
142			←--RTRIP	←--INP10	...0094
143				←--ISFDES	...0005
144				←--LCONTG	...0033
145				←--FTBNID	...0008
146				←--FMVLR	...0070
147				←--FTBCHK	
148				←--FTBDEL	...0009
149				←--FTBIN	...0040
150				←--FTBMV	
151				←--FTBRV	...0015
152				←--FTBSFT	...0045
153				←--INP2	...0028
154				←--INPLNK	
155			←--RNONCH	←--INP2	...0028
156			←--RNFLDS	←--INP10	...0094
157				←--LCONTG	...0033
158				←--FTBNID	...0008
159				←--INP2	...0028
160				←--INPLNK	
161				←--FTBDEL	...0009
162				←--FTBRV	...0015
163				←--FTBSFT	...0045
164				←--FMVLR	...0070
165			←--RCOMP	←--FTBNID	...0008
166				←--INP10	...0094
167				←--INP9	←--FABEND
168				←--ISFDES	...0005

PAGE 0131
(007)

RELAPS	INPUTD	RNEWP	RCOMP	ISFOES
169				LCNTG ...0033
170				LCNTGS ...0043
171				FABEND
172				FMVLR ...0070
173				FTBDEL ...0009
174				FTBRV ...0015
175				FTBSFT ...0045
176				INP2 ...0028
177				INPLK
178			R3DCMP	INP10 ...0094
179				FTBSFT ...0045
180			RACCUM	INP10 ...0094
181				LCNTGS ...0043
182				FTBMOV
183				FTBSFT ...0045
184				INP2 ...0028
185				INPLK
186				SETNDF FABEND
187			RBRNCH	INP10 ...0094
188				LCNTGS ...0043
189				FTBSFT ...0045
190				INP2 ...0028
191				FTBMOV
192				SETNDF ...0186
193				INPLK
194				FABEND
195			RCDELT	LCNTG ...0033
196				FTBMOV

PAGE 0132
(008)

RELAPS	INPUTD	RNEWP	RCOMP	RCDELT	FTBMOV
197					FTBSFT ...0045
198				RHTPLJ	INP10 ...0094
199					LCNTGS ...0043
200					FTBMOV
201					FTBSFT ...0045
202					INP2 ...0028
203					INPLK
204					SETNDF ...0186
205				RPIPE	INP10 ...0094
206					LCNTGS ...0043
207					FABEND
208					FTBMOV
209					FTBSFT ...0045
210					INP2 ...0028
211					INP5 FABEND
212					INP2 ...0028
213					INP6 FABEND
214					INPLK
215					SETNDF ...0186
216					INPLK
217				RPUMP	INP10 ...0094
218					LCNTGS ...0043
219					FTBMOV
220					FTBSFT ...0045
221					INP2 ...0028
222					RPNPDC INP2 ...0028
223					RPNPHD LCNTGS ...0043
224					FTBSFT ...0045

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:48:22) PAGE 0133
(009)

RELAPS	INPUT	RNEUP	RCMPN	RPUMP	RPMPD	FTBSFT
225						---INP2 ...0028
226					---RPNVJ---	---LCNTGS ...0063
227						---FABEND
228						---FTBMOV
229						---FTBSFT ...0045
230						---INP2 ...0028
231						---SETNDF ...0186
232					---SCHREQ	...0139
233				---RSNGJ---	---INP10	...0094
234					---LCNTGS	...0063
235					---FTBMOV	
236					---FTBSFT	...0045
237					---INP2	...0028
238					---SETNDF	...0186
239				---RSNGV---	---INP10	...0094
240					---LCNTGS	...0063
241					---FTBSFT	...0045
242					---FTBMOV	
243					---SETNDF	...0186
244					---INP2	...0028
245					---FABEND	
246				---RTMDJ---	---INP10	...0094
247					---LCNTGS	...0063
248					---FTBMOV	
249					---FTBSFT	...0045
250					---INP2	...0028
251					---SCHREQ	...0139
252					---SETNDF	...0186

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:48:22) PAGE 0134
(010)

RELAPS	INPUT	RNEUP	RCMPN	RTMDJ	SETNDF
253				---RTMDV---	---INP10 ...0094
254					---LCNTGS ...0063
255					---FABEND
256					---FTBMOV
257					---FTBSFT ...0045
258					---INP2 ...0028
259					---SCHREQ ...0139
260					---SETNDF ...0186
261				---RTURB---	---INP10 ...0094
262					---LCNTGS ...0063
263					---FTBSFT ...0045
264					---INP2 ...0028
265					---FTBMOV
266					---SETNDF ...0186
267					---INPLNK
268					---FABEND
269				---RVALVE---	---INP10 ...0094
270					---LCNTGS ...0063
271					---FABEND
272					---FTBMOV
273					---FTBSFT ...0045
274					---INP2 ...0028
275					---SETNDF ...0186
276			---RMTCHP---	---INP10	...0094
277				---INP9	...0167
278				---ISFDES	...0005
279				---LCNTG	...0033
280				---LCNTGS	...0063

PAGE 0135
(011)

S T R E A M 7 7 (C)				V01/L04 (19860901)		DATE (97.08.25) TIME (11:48:22)	
RELAPS	INPUTD	RNEWP	RHTCMP	LCNTGS			
281				---FTBNID	...	0008	
282				---FTBDEL	...	0009	
283				---FTBMOV			
284				---FTBRVS	...	0015	
285				---FTBSFT	...	0045	
286				---HT1INP	---	INP10	...0094
287					---	INP2	...0028
288					---	INP5	...0211
289					---	INPLNK	
290				---IMP2	...	0028	
291				---INP5	...	0211	
292				---INPLNK			
293			---RRADHT	---	LOCF	...	0012
294				---LOCF1	---	TLOC8	
295				---LCONTG	...	0033	
296				---FTBNID	...	0008	
297				---SIMUL			
298				---FTBMOV			
299				---FTBRVS	...	0015	
300				---IMP2	...	0028	
301				---INP5	...	0211	
302				---INPLNK			
303				---INPHDD	...	0029	
304				---FTBSFT	...	0045	
305				---FMVLR	...	0070	
306			---RMADAT	---	LOCF	...	0012
307				---IMP10	...	0094	
308				---IMP9	...	0167	

PAGE 0136
(012)

S T R E A M 7 7 (C)				V01/L04 (19860901)		DATE (97.08.25) TIME (11:48:22)	
RELAPS	INPUTD	RNEWP	RMADAT	INP9			
309				---LCONTG	...	0033	
310				---LCNTGS	...	0063	
311				---FTBNID	...	0008	
312				---FABEND			
313				---FMVLR	...	0070	
314				---FTBCHK			
315				---FTBDEL	...	0009	
316				---FTBIN	...	0040	
317				---FTBMOV			
318				---FTBRVS	...	0015	
319				---FTBSFT	...	0045	
320				---IMP2	...	0028	
321				---INPLNK			
322			---RRKIN	---	INP10	...	0094
323				---IMP9	...	0167	
324				---ISFDES	...	0005	
325				---LCNTGS	...	0063	
326				---LCONTG	...	0033	
327				---FTBNID	...	0008	
328				---INPLNK			
329				---IMP2	...	0028	
330				---FTBDEL	...	0009	
331				---FTBSFT	...	0045	
332				---FMVLR	...	0070	
333				---FTBRVS	...	0015	
334				---RRKINH			
335				---RRKINP	---	INP10	...0094
336					---	LCNTGS	...0063

S T R E A M 7 7 (C) VO1/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0137
(013)

RELAPS	INPUTD	RNEWP	RRKIN	RRKINP	LCNTGS
337					---FTBSFT ...0045
338					---INP2 ...0028
339					---INPLNK
340			---RGNTBL---	---INP10 ...0094	
341				---INP9 ...0167	
342				---LCONTG ...0033	
343				---LCNTGS ...0063	
344				---FTBNID ...0008	
345				---FABEND	
346				---FTBDEL ...0009	
347				---FTBMOV	
348				---FTBRVS ...0015	
349				---FTBSFT ...0045	
350				---FMVLWR ...0070	
351				---INP2 ...0028	
352				---INPLNK	
353			---RCNVR---	---INP10 ...0094	
354				---INP9 ...0167	
355				---ISFDES ...0005	
356				---LCONTG ...0033	
357				---FTBNID ...0008	
358				---FABEND	
359				---FMVLWR ...0070	
360				---FTBCHK	
361				---FTBDEL ...0009	
362				---FTBIN ...0040	
363				---FTBMOV	
364				---FTBRVS ...0015	

S T R E A M 7 7 (C) VO1/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0138
(014)

RELAPS	INPUTD	RNEWP	RCNVR	FTBRVS
365				---FTBSFT ...0045
366				---INP2 ...0028
367				---INPLNK
368			---FTBDEL ...0009	
369			---ITSTCK	
370			---ICMPS1---	---LCONTG ...0033
371				---FTBNID ...0008
372				---FABEND
373				---IPIPE
374				---ISNGJ-----FABEND
375				---ICNPF
376				---ISDCMP-----FABEND
377				---FTBDEL ...0009
378				---IMLP-----FTBNID ...0008
379				---ISFDES ...0005
380				---LCONTG ...0033
381				---FMVLWR ...0070
382				---FTBDEL ...0009
383				---FTBMOV
384				---FTBRVS ...0015
385				---FTBSFT ...0045
386				---FTBRVS ...0015
387				---STREAD
388				---FTBSFT ...0045
389			---ITRIP---	---FMVLWR ...0070
390				---ITRSCH-----LDCF ...0012
391				---SCNRER ...0139
392			---IGNTBL---	---ITRSCH ...0390

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0139
(015)

RELAPS	INPUTD	RNEWP	IGNTBL	ITRSCN		
393			---ICOMP	---FTBNID	...0008	
394				---ITRSCN	...0390	
395				---SCHREG	...0139	
396				---ISTATE	---STCSET	
397				---POLAT		
398				---STH2X1		
399				---STD2X1		
400				---STPU2T		
401				---STRTX	---STRSAT	
402					---STRX	---STRSAT
403				---STH2X3		
404				---STD2X3		
405				---STPUTP		
406				---STRTP	---STRSAT	
407				---STH2X6		
408				---STD2X6		
409				---STPUPU		
410				---STRPU1	---STRPU	---STRSAT
411				---STH2X2		
412				(STH2X1)		
413				---STD2X2		
414				(STD2X1)		
415				---STPU2P		
416				---STRPX	---STRSAT	
417					---STRX	...0402
418				---FABEND		
419				---STH2XF		
420				(STH2X6)		
				---STD2XF		
				(STD2X6)		
				---STRPU2	---STRPU	...0410
				---VISCOS		

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0140
(016)

RELAPS	INPUTD	RNEWP	ICOMP	ISTATE	VISCOS	
421					---THCOND	
422					---PSAYPD	
423					---PSTPD2	
424					---STPU00	
425					---STRSAT	
426					---SURFTM	
427				---IVELST	---FABEND	
428					---POLAT	
429				---FABEND		
430				---POLAT		
431				---IJPROP		
432				---INVJT	---ISFDES	...0005
433					---LCONTG	...0033
434					---FTBNID	...0008
435					---FTBDEL	...0009
436					---FTBRSV	...0015
437					---FTBSFY	...0045
438				---IELVTM		
439				---IVLVEL		
440				---FLOSTJ		
441				---FLOSTV		
442				---IPUMP	---PUMP2	---POLAT
443						---POLATR
444					---POLAT	
445				---IEDIT		
446				---FTBDEL	...0009	
447				---FTBRSV	...0015	
448				---FMVLR	...0070	

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0141 (017)

RELAP5	INPUTD	RNEWP	ICOMP	FNULWR
449			---INTCMP---	---LOCF ...0012
450				---ISFDES ...0005
451				---LCONTG ...0033
452				---FTBNID ...0008
453				---FTBDEL ...0009
454				---FTBMDV
455				---FTBRV ...0015
456				---HT1SST---POLAT
457				---FTBMDV
458				---MADATA---FABEND
459				---GAPCON---CELMDR
460				---CTHXPR
461				---CPLEXP---RUPLAS
462				---PLSTRN---POLAT
463				---GASTHC
464				---HTCOND---HTRC1---CONDN2---HTFILM
465				---NCWALL---NCPRDP (* 1)
466				---PSATPD
467				---DITTUS---PETUKV (* 2)
468				---SYCSET
469				---STH2X2
470				(---STH2X1)
471				---STD2X2
472				(---STD2X1)
473				---STPU2P
474				---STRPX ...0414
475				---FABEND
476				---CHFCAL---STH2X1
				---STD2X1
				---STPU2T

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0142 (018)

RELAP5	INPUTD	RNEWP	INTCMP	HT1SST	HTCOND	HTRC1	CHFCAL	STPU2T
477								---FABEND
478								---STH2X3
479								---STD2X3
480								---STPUTP
481								---VISCOS
482							---CHFKUT---STH2X2	
483							(---STH2X1)	
484							---STD2X2	
485							(---STD2X1)	
486							---STPU2P	
487							---STRPX ...0414	
488							---FABEND	
489							---SURFTN	
490							---THCOND	
491							---PREDNB---DITTUS ...0467	
492							---PSATPD	
493							---PSTPD2	
494							---STPU00	
495							---STRSAT	
496							---PREBUN---PSATPD	
497							---PSTPD2	
498							---STPU00	
499							---STRSAT	
500							---SUBOIL	
501							---PSTDNB---STH2X3	
502							---STD2X3	
503							---STPUTP	
504							---STRTP ...0406	
							---FABEND	
							---THCOND	

PAGE 0143
(019)

RELAPS	INPUTD	RNEWP	INTCMP	HTSST	HTCOND	HTRC1	PSTDNB	THCOND
505								---VISCOS
506								---DITTUS ...0447
507								---DITTUS ...0447
508								---CONDEN---DITTUS ...0447
509								---NONCND---PINTFC (3)
510								---SH2X0
511								---SD2X0
512								---STPUOP
513								---ZBRENT
514								*X-PINTFC ...0509
515						---POLAT		
516			---IRADHT---SIMUL					
517			---RADHT					
518			---IRFLHT---LCNTGS ...0063					
519			---LCONTG ...0033					
520			---FTBNID ...0008					
521			---FTBDEL ...0009					
522			---FTBMQV					
523			---FTBRV ...0015					
524			---FTBSFT ...0045					
525			---ITRSCN ...0390					
526			---INVHTS---ISFDES ...0005					
527			---LCONTG ...0033					
528			---FTBNID ...0008					
529			---FTBDEL ...0009					
530			---FTBRV ...0015					
531			---FTBSFT ...0045					
532			---IRKIN---LOCF ...0012					

PAGE 0144
(020)

RELAPS	INPUTD	RNEWP	IRKIN	LOCF
533			---POLAT	
534			---ICONVR---FABEND	
535			---ITRSCN ...0390	
536			---POLAT	
537			---SCHREQ ...0139	
538			---IMIEDT---LCONTG ...0033	
539			---FTBNID ...0008	
540			---FMVLWR ...0070	
541			---FTBRV ...0015	
542			---SCHREQ ...0139	
543			---ISSI---FMVLWR ...0070	
544			---IUSRVR---SCHREQ ...0139	
545			---WRPLID---ISFDES ...0005	
546			---LCONTG ...0033	
547			---FTBNID ...0008	
548			---DMPLST ...0005	
549			---FTBDEL ...0009	
550			---FTBRV ...0015	
551		---RSTRIP---INPS		
552			---ISFDES ...0005	
553			---LCONTG ...0033	
554			---FTBNID ...0008	
555			---SRESTF---AATL	
556			---INP2 ...0028	
557			---FTBRV ...0015	
558			---INPLNK	
559			---INP2 ...0028	
560			---FTBDEL ...0009	

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0145
(021)

RELAPS	INPUTD	RSTRIP	FTBDEL
561			←--FTBSFT ...0045
562		←--CMPCOM	←--LDCF4 ...0003
563			←--LDCF14 ...0004
564			←--INPB
565			←--LCONTG ...0033
566			←--FTBNID ...0008
567			←--FTBRVS ...0015
568			←--INP2 ...0028
569			←--FTBDEL ...0009
570			←--FTBSFT ...0045
571	←--FABEND		
572	←--TRNCTL	←--TRAN	←--DTSTEP
573			←--TIMER (TIMERSET)
574			←--RENTIM
575			←--TRCPU
576			←--CMPCOM
577			←--LDCF4 ...0003
578			←--MOVER
579			←--JPROP
580			←--HELPHD
581			←--HZFLD
582			←--HTHETA
583			←--STDSP
584			←--DRYER
585			←--GESEP
586			←--GESUB
587			←--STATE
588			←--HELPHD
			←--STACC
			←--STCSET
			←--STH2X6
			←--STD2X6
			←--STPUPU
			←--STRPU1 ...0410
			←--STH2X1
			←--STD2X1
			←--STPU2T

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:48:22) PAGE 0146
(022)

RELAPS	TRNCTL	TRAN	DTSTEP	MOVER	STATE	STACC	STPU2T
589							←--STRTX ...0401
590							←--HELPHD
591						←--STATEP	←--STCSET
592							←--STH2X6
593							←--STD2X6
594							←--STPUPU
595							←--STRPU1 ...0410
596							←--STH2X1
597							←--STD2X1
598							←--STPU2T
599							←--STRTX ...0401
600							←--STH2X2
601							←--STH2X1
602							←--STD2X2
603							←--STD2X1
604							←--STPU2P
605							←--STRPX ...0414
606							←--VISCOS
607							←--THCONO
608							←--PSATPD
609							←--PSTPD2
610							←--STPUGO
611							←--STRSAT
612							←--SVPUPU
613							←--SVH2X2
614							←--SVPUP2
615							←--STVRPX
616							←--STRSAT
							←--STH2XF
							←--STH2X6
							←--STD2XF
							←--STD2X6
							←--STRPU2 ...0419

PAGE 0147
(023)

RELAPS	TRNCTL	TRAN	DTSTEP	MOVER	STATE	STATEP	STRPU2
617							←-SURFTN
618					←-VALVE	←-HELPHD	
619						←-POLAT	
620						←-FABEND	
621					←-VLVELA	←-HELPHD	
622				←-SSTCHK	←-CRAMER	←-DETMNT	
623					←-RMBLNK		
624				←-MIREC			
625				←-MAJDUT			
626				←-RSTREC	←-LOCF4	...0003	
627					←-ISFDES	...0005	
628					←-FTBCHK		
629					←-FTBIN	...0040	
630				←-PLTWRT	←-SQ02		
631			←-TRIP	←-FABEND			
632			←-YSTATE	←-POLAT			
633				←-STCSET			
634				←-STH2X1			
635				←-STD2X1			
636				←-STPU2T			
637				←-STRTX	...0401		
638				←-STH2X3			
639				←-STD2X3			
640				←-STPUTP			
641				←-STRTP	...0406		
642				←-STH2X6			
643				←-STD2X6			
644				←-STPUPU			

PAGE 0148
(024)

RELAPS	TRNCTL	TRAN	TSTATE	STRPU1	STRPU2		
645				←-STRPU1	...0410		
646				←-STH2X2			
647				←-STH2X1			
648				←-STD2X2			
649				←-STPU2P			
650				←-STRPX	...0414		
651				←-STH2XF			
652				←-STD2XF			
653				←-STRPU2	...0419		
654				←-PSATPD			
655				←-PSTPD2			
656				←-STPU00			
657				←-STRSAT			
658			←-HTADV	←-HT1TDP	←-FTBNOV		
659					←-HTCOND	...0464	
660					←-KLOSS		
661					←-MADATA	...0458	
662					←-POLAT		
663					←-QNR		
664				←-QFNOVE	←-FTBNOV		
665					←-HT2TDP	←-POLAT	
666						←-QNR	
667					←-MADATA2	←-GASTHC	
668						←-GAPCON	...0459
669						←-KLOSS	
670					←-FTBNOV		
671					←-HTCSOL		
672				←-HTRC2	←-POLAT		
					←-HTRC1	...0464	

PAGE 0149
(025)

RELAPS	TRNCTL	TRAN	HTADV	QFMOVE	HTRC2	HTRC1
673					---QFHTRC---	---CHFCAL ...0474
674						---HTRC1 ...0464
675					---QFSRCH	
676				---RADNT		
677			---HYDRD---	---VALVE ...0418		
678				---VOLVEL---	---HELPHD	
679				---LEVEL		
680				---TFRONT		
681				---PHANTV---	---HTHETA	
682					---NCFILM---	---NCPRDP ...0465
683						---PSATPD
684					---HTFILM	
685					---STCSET	
686					---ECCMXV---	---FIDIS
687						---HIFBUB
688						---HTHETA
689						---FIDISV
690					---FIDIS	
691					---FIDISV	
692					---HIFBUB	
693					---HELPHD	
694				---PHANTJ---	---HTHETA	
695					---STCSET	
696					---ECCMXJ---	---FIDIS2
697						---HTHETA
698					---FIDIS2	
699					---FIDISJ---	---BRVCEJ
700						---EPRIJ

PAGE 0150
(026)

RELAPS	TRNCTL	TRAN	HYDRO	PHANTJ	FIDISJ	EPRIJ
701						---ZPSLGJ
702						---GRDNRJ
703						---KATOKJ
704						---GRIFTJ
705						---HELPHD
706				---FWDRAG---	---HELPHD	
707					---VISCOS	
708				---HLOSS		
709				---VEXPLT---	---ACCUM---	---HELPHD
710					---HELPHD	
711					---PUMP---	---POLAT
712						---PUMP2 ...0442
713					---TURBST	
714					---STH2X6	
715					---STD2X6	
716					---STPUPU	
717					---STRPU1	...0410
718					---FABEND	
719				---JCHOKE---	---STCSET	
720					---STH2X0	
721					---STD2X0	
722					---STPUOP	
723					---STRSAT	
724					---STH2X1	
725					---STD2X1	
726					---STPU2T	
727					---STRTX	...0401
728					---STH2X2	
					(STH2X1)	

PAGE 0151
(027)

RELAPS	TRNCTL	TRAN	HYDRO	JCHOKE	STHX2
729					---STD2X2
730					(---STD2X1)
731					---STPU2P
732					---STRPX ...0414
733					---STH2X6
734					---STD2X6
735					---STPU2U
736					---STRPU1 ...0410
737					---HELPHD
738					---PSATP0
739					---PSTPD2
740					---STPU00
741					---STH2X3
742					---STD2X3
743					---STPUTP
744					---STRTP ...0406
745					---CCFL-----HELPHD
746					---JPROP ...0575
747					---VFINL-----HELPHD
748					---JPROP ...0575
749					---PACKER
750					---PRESEQ-----HELPHD
751					---SYSSOL-----PMINVD-----PMINV1
752					-----PMINV4
753					---PMINVR
754					---PMINVF-----PMINVD ...0730
755					---PMINVX
756					---HELPHD
					+X-PSET

PAGE 0152
(028)

RELAPS	TRNCTL	TRAN	HYDRO	VFINL	SYSSOL	PSET
757					---EQFINL-----HELPHD	
758					---MOVER ...0575	
759					---VIMPLT-----ACCUM ...0709	
760					---CCFL ...0744	
761					---HELPHD	
762					---JCHOKE ...0719	
763					---PUMP ...0711	
764					---TURBST	
765					---SYSSOL ...0750	
766					---JPROP ...0575	
767					---PIMPLT-----HELPHD	
768					---SIMPLY-----HELPHD	
769					---SYSSOL ...0750	
770					---BRNTRN	
771					---STATE ...0580	
772					---VLVELA ...0621	
773					---HTFINL	
774					---HELPHD	
775					---RKIN-----POLAT	
776					---CONVAR-----FABEND	
777					---POLAT	
778					---TRNFIN-----ISFDES ...0005	
779					---MFUNIT ...0071	
780					---FTBDEL ...0009	
781					---FTBEXP ...0077	
782					---FTBSFT ...0045	
783					---TRNSET-----LCONTG ...0033	
784					---FTBHD ...0008	

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:48:22) PAGE 0153
(029)

RELAPS	TRNCTL	TRNSET	FTBNID	
785			+--DMPLST	...0005
786			+--FTBCHK	
787			+--FTBDEL	...0009
788			+--FTBIN	...0040
789			+--FTBRV	...0015
790			+--FTBSFT	...0045
791			+--TCNVSL	
792			+--TSETSL	+--LCNTG ...0033
793				+--FTBNID ...0008
794				+--FABEND
795				+--FTBRV ...0015
796				+--FTBSFT ...0045
797				+--PMINVD ...0750
798				+--FTBDEL ...0009
799			+--FVLR	...0070
800			+--STGDU	
801			+--FILDM	
802			+--REEDIT	
803			+--STRIP	+--LCNTG ...0033
804				+--LCNTGS ...0063
805				+--FTBNID ...0008
806				+--FTBRV ...0015
807				+--FTBSFT ...0045
808			+--UNSQZ	

(CONT'D NO. 1)
 809 NCPROP---(STH2X2)
 810 | (STH2X1)
 +--THCOND
 |

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:48:22) PAGE 0154
(030)

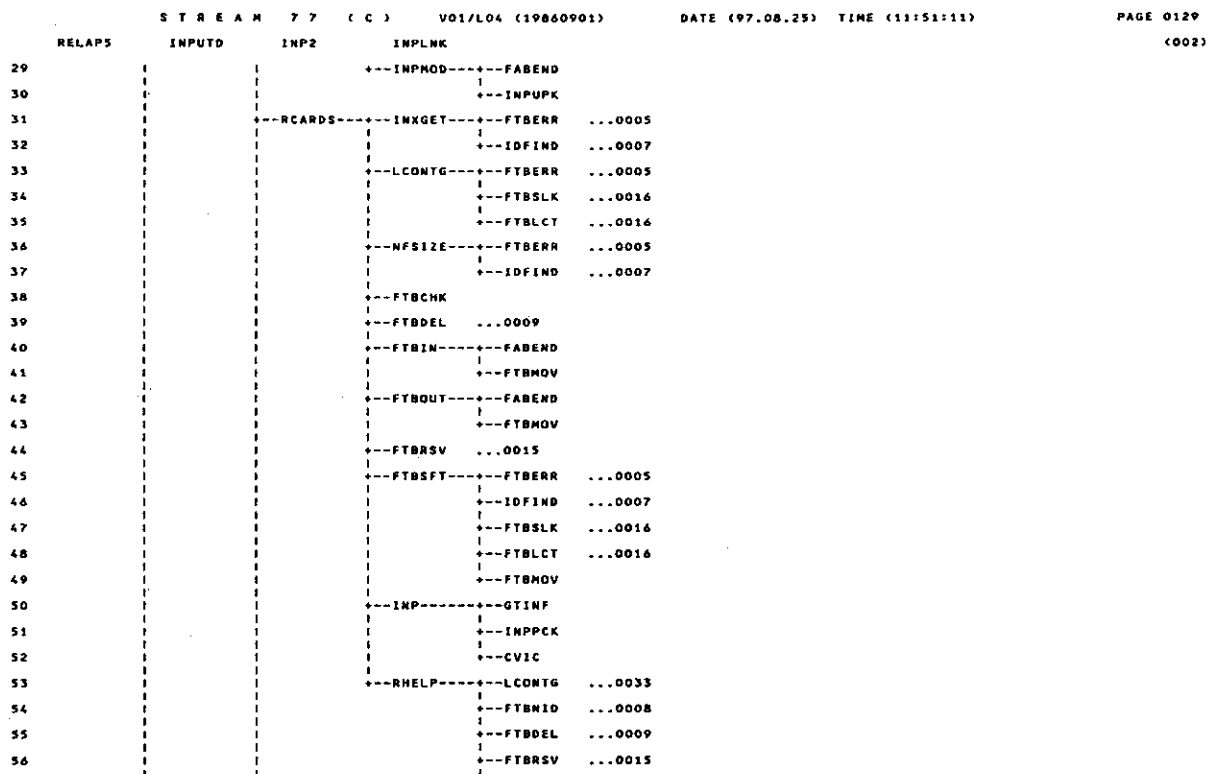
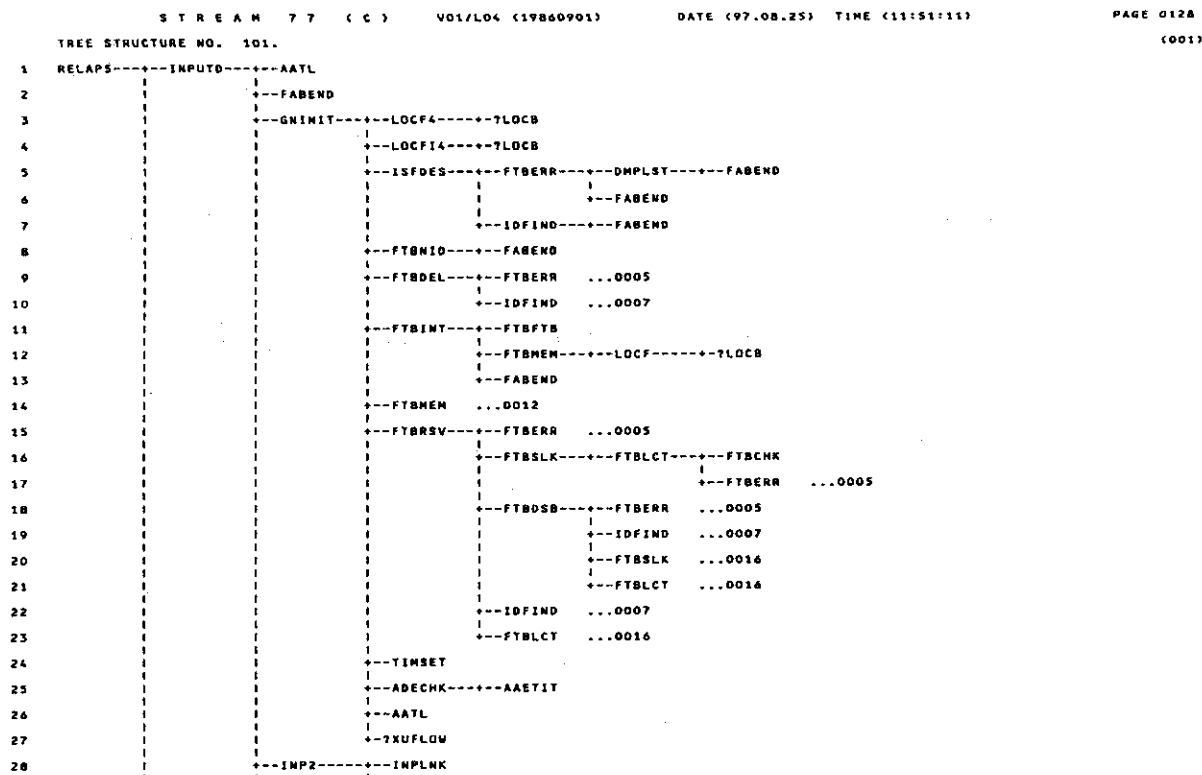
NCPROP	THCOND
811	+--VISCOS

(CONT'D NO. 2)
 812 PETUKV---VISCOS

(CONT'D NO. 3)
 813 PINTFC---(STH2X2)
 | (STH2X1)
 814 +--STD2X2)
 | (STD2X1)
 815 +--STPU2P)
 |
 816 +--THCOND)
 |
 817 +--VISCOS)

付録 B RELAP5/MOD3.2.1.2 のツリー構図

次頁より、RELAP5/MOD3.2.1.2 コードのツリー構図を示す。



PAGE 0130
(003)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RCARDS	RHELP	FTBRVS
57				---FTBSFT ...0045
58				---INP2 ...0028
59			---RDREDT	
60			---RNEVP	
61			---RRSTD	---INP2 ...0028
62			---RRESTF	---LDCF4 ...0003
63				---LCNTGS ---FTBERR ...0005
64				---IDFIND ...0007
65				---LCONTG ...0033
66				---FTBNID ...0008
67				---AATL
68				---FABEND
69				---FMVLWR ---ISFDES ...0005
70				---NFUNIY ---FTBERR ...0005
71				---IDFIND ...0007
72				---ISSFRG ---FTBERR ...0005
73				---FTBSFT ...0045
74				
75				---FTBCHK
76				---FTBDEL ...0009
77				---FTBEXP ---FTBERR ...0005
78				---FTBMEN ...0012
79				---FTBMOV
80				---FTBSLK ...0016
81				---FTBOUT ...0042
82				---FTBRDC ---FTBSLK ...0016
83				---FTBERR ...0005
84				---FTBMOV

PAGE 0131
(004)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEVP	RRESTF	FTBRDC	FTBMOV
85					---FTBMEN ...0012
86				---FTBRVS ...0015	
87				---FTBSFT ...0045	
88				---INP2 ...0028	
89			---RCHNG	---FTBNID ...0008	
90				---FTBDEL ...0009	
91				---FTBRVS ...0015	
92				---INP2 ...0028	
93			---INP2	...0028	
94			---RTSC	---INP10 ---FABEND	
95				---LCONTG ...0033	
96				---FTBNID ...0008	
97				---FTBDEL ...0009	
98				---FTBRVS ...0015	
99				---FTBSFT ...0045	
100				---INP2 ...0028	
101			---RSSI	---ISFDES ...0005	
102				---LCONTG ...0033	
103				---FTBNID ...0008	
104				---FTBDEL ...0009	
105				---FTBRVS ...0015	
106				---FTBSFT ...0045	
107				---INP2 ...0028	
108				---INPLNK	
109			---RMIEDT	---INP10 ...0094	
110				---ISFDES ...0005	
111				---LCONTG ...0033	
112				---FTBNID ...0008	

RELAPS	INPUTD	RNEWP	RMIEDT	FTBNID	DATE (97.08.25)	TIME (11:51:11)	PAGE 0132 (005)
113				---FMVLR ...0070			
114				---FTBDEL ...0009			
115				---FTBRV ...0015			
116				---FTBSFT ...0045			
117				---INP2 ...0028			
118				---INPLNK			
119			---RUSVR	---INP10 ...0094			
120				---ISFDES ...0005			
121				---LCONTG ...0033			
122				---FTBNID ...0008			
123				---FMVLR ...0070			
124				---FTBDEL ...0009			
125				---FTBRV ...0015			
126				---FTBSFT ...0045			
127				---INP2 ...0028			
128				---INPLNK			
129			---RINTRV	---LCONTG ...0033			
130				---FTBNID ...0008			
131				---FTCHK			
132				---FTBDEL ...0009			
133				---FTBIN ...0040			
134				---FTBNOV			
135				---FTBRV ...0015			
136				---FTBSFT ...0045			
137				---INP2 ...0028			
138				---INPLNK			
139				---SCHREQ---LDCF ...0012			
140				---FABEND			

RELAPS	INPUTD	RNEWP	RINTRV	SCNREQ	FABEND	DATE (97.08.25)	TIME (11:51:11)	PAGE 0133 (006)
141					---RMBLNK			
142			---RTRIP	---INP10 ...0094				
143				---ISFDES ...0005				
144				---LCONTG ...0033				
145				---FTBNID ...0008				
146				---FMVLR ...0070				
147				---FTCHK				
148				---FTBDEL ...0009				
149				---FTBIN ...0040				
150				---FTBNOV				
151				---FTBRV ...0015				
152				---FTBSFT ...0045				
153				---INP2 ...0028				
154				---INPLNK				
155			---RNOHCN	---INP2 ...0028				
156			---RMFLDS	---INP10 ...0094				
157				---LCONTG ...0033				
158				---FTBNID ...0008				
159				---INP2 ...0028				
160				---INPLNK				
161				---FTBDEL ...0009				
162				---FTBRV ...0015				
163				---FTBSFT ...0045				
164				---FMVLR ...0070				
165			---RCOMP	---FTBNID ...0008				
166				---INP10 ...0094				
167				---INP9---FABEND				
168				---ISFDES ...0005				

PAGE 0134
(007)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	RCOMPN	ISFDES
169				---LCNTG ...0033
170				---LCNTGS ...0063
171				---FABEND
172				---FMVLR ...0070
173				---FTBDEL ...0009
174				---FTBRV ...0015
175				---FTBSFT ...0045
176				---INP2 ...0028
177				---INPLNK
178			---R3DCMP	---INP10 ...0094
179				---FTBSFT ...0045
180			---RACCUM	---INP10 ...0094
181				---LCNTGS ...0063
182				---FTBMOV
183				---FTBSFT ...0045
184				---INP2 ...0028
185				---INPLNK
186				---SETNDF ---FABEND
187			---RBRNCH	---INP10 ...0094
188				---LCNTGS ...0063
189				---FTBSFT ...0045
190				---INP2 ...0028
191				---FTBMOV
192				---SETNDF ...0186
193				---INPLNK
194				---FABEND
195			---RCDELT	---LCNTG ...0033
196				---FTBMOV

PAGE 0135
(008)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	RCOMPN	RCDELT	FTBMOV
197					---FTBSFT ...0045
198				---RMTPLJ	---INP10 ...0094
199					---LCNTGS ...0063
200					---FTBMOV
201					---FTBSFT ...0045
202					---INP2 ...0028
203					---INPLNK
204					---SETNDF ...0186
205				---RPIPE	---INP10 ...0094
206					---LCNTGS ...0063
207					---FABEND
208					---FTBMOV
209					---FTBSFT ...0045
210					---INP2 ...0028
211					---INP5 ---FABEND
212					---INP2 ...0028
213					---INP6 ---FABEND
214					---INPLNK
215					---SETNDF ...0186
216					---INPLNK
217				---RPUMP	---INP10 ...0094
218					---LCNTGS ...0063
219					---FTBMOV
220					---FTBSFT ...0045
221					---INP2 ...0028
222					---RPNPDC ---INP2 ...0028
223					---RPNPMD ---LCNTGS ...0063
224					---FTBSFT ...0045

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:51:11) PAGE 0136
(009)

RELAPS	INPUTD	RNEWP	RCONPN	RPUMP	RPMND	FTBSFT
225						---INP2 ...0028
226					---RPMVJ---	---LCNTGS ...0063
227						---FABEND
228						---FTBNOV
229						---FTBSFT ...0045
230						---INP2 ...0028
231						---SETNDF ...0186
232					---SCHREQ	...0139
233				---RSGJ---	---INP10	...0094
234						---LCNTGS ...0063
235						---FTBNOV
236						---FTBSFT ...0045
237						---INP2 ...0028
238						---SETNDF ...0186
239				---RSGV---	---INP10	...0094
240						---LCNTGS ...0063
241						---FTBSFT ...0045
242						---FTBNOV
243						---SETNDF ...0186
244						---INP2 ...0028
245						---FABEND
246				---RTMDJ---	---INP10	...0094
247						---LCNTGS ...0063
248						---FTBNOV
249						---FTBSFT ...0045
250						---INP2 ...0028
251						---SCHREQ ...0139
252						---SETNDF ...0186

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:51:11) PAGE 0137
(010)

RELAPS	INPUTD	RNEWP	RCONPK	RTMDJ	SETNDF
253				---RTMDV---	---INP10 ...0094
254					---LCNTGS ...0063
255					---FABEND
256					---FTBNOV
257					---FTBSFT ...0045
258					---INP2 ...0028
259					---SCHREQ ...0139
260					---SETNDF ...0186
261				---RTURB---	---INP10 ...0094
262					---LCNTGS ...0063
263					---FTBSFT ...0045
264					---INP2 ...0028
265					---FTBNOV
266					---SETNDF ...0186
267					---INPLK
268					---FABEND
269				---RVALVE---	---INP10 ...0094
270					---LCNTGS ...0063
271					---FABEND
272					---FTBNOV
273					---FTBSFT ...0045
274					---INP2 ...0028
275					---SETNDF ...0186
276			---RHTCMP---	---INP10	...0094
277					---INP9 ...0167
278					---ISFDES ...0005
279					---LCNTG ...0033
280					---LCNTGS ...0063

PAGE 0138
(011)

S T R E A M 7 7 (C) VO1/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	RHTCMP	LCNTGS	
281				---FTBNID	...0008
282				---FTBDEL	...0009
283				---FTBMOV	
284				---FTBRVS	...0015
285				---FTBSFT	...0045
286				---HT1IMP	---INP10 ...0094
287				---INP2	...0028
288				---INP5	...0211
289				---INPLNK	
290				---INP2	...0028
291				---INP5	...0211
292				---INPLNK	
293			---RRADHT	---LOCF	...0012
294				---LOCF1	---7LOC8
295				---LCDNTG	...0033
296				---FTBNID	...0008
297				---SIMUL	
298				---FTBMOV	
299				---FTBRVS	...0015
300				---INP2	...0028
301				---INP5	...0211
302				---INPLNK	
303				---INPHOD	...0029
304				---FTBSFT	...0045
305				---FMVLWR	...0070
306			---RMADAT	---LOCF	...0012
307				---INP10	...0094
308				---INP9	...0167

PAGE 0139
(012)

S T R E A M 7 7 (C) VO1/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	RMADAT	INP9	
309				---LCDNTG	...0033
310				---LCNTGS	...0063
311				---FTBNID	...0008
312				---FABEND	
313				---FMVLWR	...0070
314				---FTBCHK	
315				---FTBDEL	...0009
316				---FTBIN	...0040
317				---FTBMOV	
318				---FTBRVS	...0015
319				---FTBSFT	...0045
320				---INP2	...0028
321				---INPLNK	
322			---RRKIN	---INP10	...0094
323				---INP9	...0167
324				---ISFDES	...0005
325				---LCNTGS	...0063
326				---LCDNTG	...0033
327				---FTBNID	...0008
328				---INPLNK	
329				---INP2	...0028
330				---FTBDEL	...0009
331				---FTBSFT	...0045
332				---FMVLWR	...0070
333				---FTBRVS	...0015
334				---RRKINH	
335				---RRKINP	---INP10 ...0094
336				---LCNTGS	...0063

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11) PAGE 0140
(013)

RELAPS	INPUTD	RNEWP	RRKIN	RRKINP	LCNTGS
337					---FTBSFT ...0045
338					---INP2 ...0028
339					---INPLNK
340			---RGNTBL---	---INP10	...0094
341				---INP9	...0167
342				---LCONTG	...0033
343				---LCNTGS	...0063
344				---FTBNID	...0008
345				---FABEND	
346				---FTBDEL	...0009
347				---FTBMOV	
348				---FTBR3V	...0015
349				---FTBSFT	...0045
350				---FMVLR	...0070
351				---INP2	...0028
352				---INPLNK	
353			---RCONVR---	---INP10	...0094
354				---INP9	...0167
355				---ISFDES	...0005
356				---LCONTG	...0033
357				---FTBNID	...0008
358				---FABEND	
359				---FMVLR	...0070
360				---FTCHK	
361				---FTBDEL	...0009
362				---FTBIN	...0040
363				---FTBMOV	
364				---FTBR3V	...0015

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11) PAGE 0141
(014)

RELAPS	INPUTD	RNEWP	RCONVR	FTBR3V
365				---FTBSFT ...0045
366				---INP2 ...0028
367				---INPLNK
368			---FTBDEL	...0009
369			---ITSTCK	
370			---ICMPN1---	---LCONTG ...0033
371				---FTBNID ...0008
372				---FABEND
373				---IPIPE
374				---ISNGJ-----FABEND
375				---ICMPF
376				---ISDCMP-----FABEND
377				---FTBDEL ...0009
378				---INLP-----FTBNID ...0008
379				---ISFDES ...0005
380				---LCONTG ...0033
381				---FMVLR ...0070
382				---FTBDEL ...0009
383				---FTBMOV
384				---FTBR3V ...0015
385				---FTBSFT ...0045
386				---FTBR3V ...0015
387				---STREAD
388				---FTBSFT ...0045
389			---ITRIP---	---FMVLR ...0070
390				---ITRSCN-----LDCF ...0012
391				---SCHREQ ...0139
392			---IGNTBL---	---ITRSCN ...0390

PAGE 0142
(015)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	IGNTBL	ITRSCN	
393			ICOMP	FTBNID	...0008
394				ITRSCN	...0390
395				SCNREG	...0139
396				ISTATE	STCSET
397				POLAT	
398				STH2X1	
399				STD2X1	
400				STRX	STRSAT
401				STRX	STRSAT
402				STH2X3	
403				STD2X3	
404				STRTP	STRSAT
405				STH2X6	
406				STD2X6	
407				STRPU1	STRPU STRSAT
408				STH2X2	(STH2X1)
409				STD2X2	(STD2X1)
410				STRPX	STRSAT
411				STRX	...0401
412				FABEND	
413				STH2XF	(STH2X6)
414				STD2XF	(STD2X6)
415				STRPU2	STRPU ...0407
416				VISCOS	
417				TNCOND	
418				PSATPD	
419				PSTPD2	
420				STRSAT	

PAGE 0143
(016)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	ICOMP	ISTATE	STRSAT
421				SURFTN	
422				IVELST	FABEND
423				POLAT	
424				FABEND	
425				POLAT	
426				FLOSTJ	
427				FLOSTV	
428				INVJT	ISFDES ...0005
429				LCONTG	...0033
430				FTBNID	...0008
431				FTBDEL	...0009
432				FTBRSV	...0015
433				FTBSFT	...0045
434				ILEVEL	IJPROP
435				IJPROP	
436				IELVTN	
437				IVLVEL	
438				INH2FLM	HTHETA
439				IPUMP	PUMP2 POLAT
440					POLATR
441					POLAT
442				IEDIT	
443				FTBDEL	...0009
444				FTBRSV	...0015
445				FMVLR	...0070
446			INTCMP	LOCF	...0012
447				ISFDES	...0005
448				LCONTG	...0033

RELAPS	INPUTD	RNEWP	IHTCMP	LCDNIG	DATE (97.08.25)	TIME (11:51:11)	PAGE 0144
449				FTBN1D ...0008			(017)
450				FTBDEL ...0009			
451				FTBNDV			
452				FTBRV ...0015			
453				HT1SST---POLAT			
454				FTBMOV			
455				HADATA---FABEND			
456				GAPCON---CELNDR			
457				---CYHNR			
458				---CPLEXP---RUPLAS			
459				---PLSTRM---POLAT			
460				---GASTHC			
461				HTCOND---HTRC1---CONDN2---HTFILM			
462				---NCWALL---NCPROP (* 1)			
463				---PSATPD			
464				---DITTUS---PETUKV (* 2)			
465				---STCSET			
466				---STH2X2			
467				(STH2X1)			
468				(STD2X2)			
469				(STD2X1)			
470				---STRPX ...0410			
471				---FABEND			
472				---CHFAL---HETHA			
473				---STH2X1			
474				---STD2X1			
475				---FABEND			
476				---STH2X3			
477				---STD2X3			
478				---VISCOS			

RELAPS	INPUTD	RNEWP	IHTCMP	HT1SST	HTCOND	HTRC1	CHFAL	VISCOS	PAGE 0145
477							CHFAL	VISCOS	(018)
478							---CHFAL	---STH2X2	
479								(STH2X1)	
480								(STD2X2)	
481								(STD2X1)	
482								---STRPX ...0410	
483								---FABEND	
484								---SURFTN	
485								---THCOND	
486								---PREDNB---DITTUS ...0464	
487								---PSATPD	
488								---PSTPD2	
489								---STRSAT	
490								---SUBOIL	
491								---PREBUN---PSATPD	
492								---PSTPD2	
493								---STRSAT	
494								---SUBOIL	
495								---PSTDNB---STH2X3	
496								---STD2X3	
497								---STATP ...0404	
498								---FABEND	
499								---THCOND	
500								---VISCOS	
501								---DITTUS ...0464	
502								---DITTUS ...0464	
503								---CONDEN---DITTUS ...0464	
504								---NONCND---PINTFC (* 3)	
								---STH2X1	
								---STD2X1	
								---ZBRENT	
								*X-PINTFC ...0500	

PAGE 0146

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	INTCMP	HT1SST	HTCOMD	HTRC1	CONDEN	NONCHD	PINTFC	(019)
505									+++POLAT	
506			---IRADHT---	---SIMUL						
507				---RADHT						
508			---IRFLHT---	---LCNTGS ...0063						
509				---LCONTG ...0033						
510				---FTBNID ...0008						
511				---FTBDEL ...0009						
512				---FTBMOV						
513				---FTBRVS ...0015						
514				---FTBSFT ...0045						
515				---ITRSCH ...0390						
516			---INVHTS---	---ISFDES ...0005						
517				---LCONTG ...0033						
518				---FTBNID ...0008						
519				---FTBDEL ...0009						
520				---FTBRVS ...0015						
521				---FTBSFT ...0045						
522			---IRK1N---	---LOCF ...0012						
523				---POLAT						
524			---ICONVR---	---FABEND						
525				---ITRSCH ...0390						
526				---POLAT						
527				---SCNREG ...0139						
528			---IMIEDT---	---LCONTG ...0033						
529				---FTBNID ...0008						
530				---FMVLWR ...0070						
531				---FTBRVS ...0015						
532				---SCNREG ...0139						

PAGE 0147

S T R E A M 7 7 (C) V01/L04 (19840901) DATE (97.08.25) TIME (11:51:11)

RELAPS	INPUTD	RNEWP	IMIEDT	SCNREG						(020)
533			---ISSI---	---FMVLWR ...0070						
534			---IUSRVR---	---SCNREG ...0139						
535			---WRPLID---	---ISFDES ...0005						
536				---LCONTG ...0033						
537				---FTBNID ...0008						
538				---DMPLST ...0005						
539				---FTBDEL ...0009						
540				---FTBRVS ...0015						
541			---RSTRIP---	---INPB						
542				---ISFDES ...0005						
543				---LCONTG ...0033						
544				---FTBNID ...0008						
545			---SRESTF---	---AATL						
546				---INP2 ...0028						
547				---FTBRVS ...0015						
548				---INPLNK						
549				---INP2 ...0028						
550				---FTBDEL ...0009						
551				---FTBSFT ...0045						
552			---CHPCOM---	---LOCF4 ...0003						
553				---LOCF14 ...0004						
554				---INPB						
555				---LCONTG ...0033						
556				---FTBNID ...0008						
557				---FTBRVS ...0015						
558				---INP2 ...0028						
559				---FTBDEL ...0009						
560				---FTBSFT ...0045						

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11) PAGE 0148
(021)

RELAPS	INPUTD	CMPCOM	FTBSFT
561	+	FABEND	
562	+	TRNCTL	+
563		+	TRAN
564			+
565			+
566			+
567			+
568			+
569			+
570			+
571			+
572			+
573			+
574			+
575			+
576			+
577			+
578			+
579			+
580			+
581			+
582			+
583			+
584			+
585			+
586			+
587			+
588			+

---TIMER (TIMSET)
 ---DMPCOM ---LOCF4 ...0003
 ---MOVER ---JPROP ---HELPHD
 ---HZFLOW ---HTHETA
 ---STDSP
 ---DRYER
 ---GESEP ---GESUB
 ---STATE ---HELPHD
 ---STACC ---STCSET
 ---STH2X6
 ---STD2X6
 ---STRPU1 ...0407
 ---STH2X1
 ---STD2X1
 ---STRTX ...0400
 ---HELPHD
 ---STATEP ---STCSET
 ---STH2X6
 ---STD2X6
 ---STRPU1 ...0407
 ---STPUPU
 ---STH2X1
 ---STD2X1
 ---STRTX ...0400
 ---STH2X2
 | (STH2X1)
 ---STD2X2
 | (STD2X1)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11) PAGE 0149
(022)

RELAPS	TRNCTL	TRAN	DTSTEP	MOVER	STATE	STATEP	STD2X2
589							+
590							+
591							+
592							+
593							+
594							+
595							+
596							+
597							+
598							+
599							+
600							+
601							+
602							+
603							+
604							+
605							+
606							+
607							+
608							+
609							+
610							+
611							+
612							+
613							+
614							+
615							+
616							+

---STRPX ...0410
 ---VISCDS
 ---THCOND
 ---PSATPD
 ---PSTPD2
 ---STRSAT
 ---SVN2X2
 ---STVRPX ---STRSAT
 ---STH2XF
 | (STH2X6)
 ---STD2XF
 | (STD2X6)
 ---STRPU2 ...0415
 ---SURFTN
 ---VALVE ---HELPHD
 ---POLAT
 ---FABEND
 ---VLVELA ---HELPHD
 ---SSTCHK ---CRAMER ---DETMNT
 ---RMBLNK
 ---MIREC
 ---MAJOUT
 ---RSTREC ---LOCF4 ...0003
 ---ISFDES ...0005
 ---FTBCHK
 ---FTBIN ...0040
 ---PLTVRT ---SQDZ
 ---COURN1
 ---COURNT
 ---CHKLEV ---LEVEL ---JPROP ...0565

PAGE 0150
(023)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	TRNCYL	TRAN	CHKLEV	LEVEL	JPROP
617					PHANTJ
618					HETHA
619					ECCXKJ
620					FIDIS2
621					FIDISJ
622					BRYCEJ
623					EPRIJ
624					ZFSLGJ
625					GDNWJ
626					KATOKJ
627					GRIFTJ
628					HELPHD
629					HELPHD
630					VLVELA ...0604
631				TRIP	FABEND
632				TSTATE	POLAT
633					STCSET
634					STH2X1
635					STD2X1
636					STRX ...0400
637					STH2X3
638					STD2X3
639					STRP ...0404
640					STH2X6
641					STD2X6
642					STRPU1 ...0407
643					STH2X2
644					STH2X1
					STD2X2
					STD2X1
					STRPX ...0410

PAGE 0151
(024)

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11)

RELAPS	TRNCYL	TRAN	TSTATE	STRPX
645				SYN2XF
646				STH2X6
647				STD2XF
648				STD2X6
649				STRPU2 ...0415
650				PSATPD
651				PSTPD2
652				STRSAT
653				HTADV
654				HT1TDP
655				FTBMOV
656				HTCOND ...0461
657				KLOSS
658				HADATA ...0455
659				POLAT
660				QMR
661				QFNOVE
662				FTBMOV
663				HT2TDP
664				POLAT
665				QMR
666				MATA2
667				GASTHC
668				GAPCDN ...0456
669				KLOSS
670				FTBMOV
671				HTCSOL
672				HTRC2
				POLAT
				HTRC1 ...0461
				QFHTRC
				CHFCAL ...0470
				HTRC1 ...0461
				QFSRCH
				RADHT
				HYDRO
				STCSET
				VALVE ...0601

S T R E A M 7 7 (C)				VO1/L04 (19860901)	DATE (97.08.25)	TIME (11:51:11)	PAGE 0152								
RELAPS	TRNCTL	TRAN	HYDRO	VALVE			(025)								
673				--VOLVEL	----	HELPHD									
674				--TFRONT											
675				--PHANTV	----	HTHETA									
676					----	ECMXY	----	HIFBUB							
677							----	HTHETA							
678							----	FIDISV							
679								FIDISV							
680								HIFBUB							
681								HTFILM							
682								HTLEV	----	XHOD					
683									----	HELM					
684									----	NCFILM	----	MCPROP	...	0462	
685										----	PSATPD				
686															
687															
688															
689															
690															
691															
692															
693															
694															
695															
696															
697															
698															
699															
700															

S T R E A M 7 7 (C)				VO1/L04 (19860901)	DATE (97.08.25)	TIME (11:51:11)	PAGE 0153
RELAPS	TRNCTL	TRAN	HYDRO	JCHOKE	STH2X2		(026)
701							
702							
703							
704							
705							
706							
707							
708							
709							
710							
711							
712							
713							
714							
715							
716							
717							
718							
719							
720							
721							
722							
723							
724							
725							
726							
727							
728							

PAGE 0154
(027)

S T R E A M 7 7 (C)				V01/L04 (19860901)		DATE (97.08.25) TIME (11:51:11)	
RELAPS	TRNCTL	TRAN	HYDRO	VFINL	SYSSDL	PMINVD	PMINV4
729						+-PMINVR	
730						+-PMINVF	---PMINV4 ...0727
731						+-PMINVX	
732						+-HELPHD	
733						+X-PSET	
734				---EQFINL	---HELPHD		
735				---HOVER	...0565		
736				---VINPLY	---ACCUM	...0691	
737					---CCFL	...0721	
738					---HELPHD		
739					---JCHOKE	...0699	
740					---PUMP	...0693	
741					---TURBST		
742					---SYSSDL	...0727	
743					---JPRDP	...0565	
744				---PIMPLY	---HELPHD		
745				---SIMPLY	---HELPHD		
746					---SYSSDL	...0727	
747				---BRMTRN			
748				---STATE	...0570		
749				---LEVEL	...0616		
750				---VLVELA	...0604		
751				---HYFINL			
752				---HELPHD			
753			---RKIN	---POLAT			
754			---CONVAR	---FABEND			
755				---POLAT			
756		---TRNFIN	---ISFDES	...0005			

PAGE 0155
(028)

S T R E A M 7 7 (C)				V01/L04 (19860901)		DATE (97.08.25) TIME (11:51:11)	
RELAPS	TRNCTL	TRNFIN	ISFDES				
757			---NFUNIT	...0071			
758			---FTBDEL	...0009			
759			---FTBEXP	...0077			
760			---FTBSFT	...0045			
761		---TRNSET	---LCONTG	...0033			
762			---FTBNID	...0008			
763			---DMPLST	...0005			
764			---FTBCHK				
765			---FTBDEL	...0009			
766			---FTBIN	...0040			
767			---FTBRVS	...0015			
768			---FTBSFT	...0045			
769			---TCNVSL				
770			---TSETSL	---LCONTG	...0033		
771				---FTBNID	...0008		
772				---FABEND			
773				---FTBRVS	...0015		
774				---FTBSFT	...0045		
775				---PMINVD	...0727		
776				---FTBDEL	...0009		
777			---FMVLWR	...0070			
778			---STGODU				
779			---FILDMP				
780	---REED1T						
781	---STRIP	---LCONTG	...0033				
782		---LCNTGS	...0063				
783		---FTBNID	...0008				
784		---FTBRVS	...0015				

S T R E A M 7 7 (C) V01/L04 (19860901) DATE (97.08.25) TIME (11:51:11) PAGE 0156
 (029)

RELAPS	STRIP	FTBSV
785		+---FTBSFT ...0045
786		+---UNSQ02
(CONT'D NO. 1)		
787	MCPROP----	+---STH2X2
		(STH2X1)
788		+---THCOND
789		+---VISCDS
(CONT'D NO. 2)		
790	PETUKV----	+---VISCDS
(CONT'D NO. 3)		
791	PINTFC----	+---STH2X2
		(STH2X1)
792		+---STD2X2
		(STD2X1)
793		+---THCOND
794		+---VISCDS

付録 C NJOY94.25 の修正内容

[修正モジュール]

修正モジュールとその修正理由の対応を以下に示す。(* : 修正有り)

	module name	n	1	2	4	5	6	9	10
1	acer			*			*		
2	broad					*			
3	ccccr				*				
4	covr			*					
5	dtfr				*				
6	errorr								
7	gaminr								
8	groupr							*	
9	heatr							*	
10	leapr								
11	matxsr			*	*				
12	mixr								
13	moder								
14	njoy				*	*			
15	plotr					*			
16	powr			*					
17	purr								*
18	reconr					*			
19	resxsr				*				
20	thermr								
21	unresr								
22	viewr		*			*			
23	wimsr			*					

[修正方法]

各修正箇所は、以下のようなコメントを付けて明示的に表している。

c*vpp*b ... n

修正記述

c*vpp*e

ここで、n は修正理由に対応しており、その対応は以下の通りである。

n	修正理由	本文中の対応
1	SUN-VPP の差に伴う修正	4.2(1) ~ (3)
2	書式なし read/write の修正	4.3.1.1
4	real*8 との領域共有に関する修正	4.3.1.3(1), (2)
5	引数の型の相違に伴うアドレス例外に関する修正	4.3.1.2(1)
6	common ブロック内のデータ対応ずれの修正	4.3.1.2(2)
9	入力データ不具合回避の修正	4.4
10	common ブロックのリンケージエラー対応	4.2(4)

[修正一覧]

次ページ以降に修正一覧を示す。


```

c
1 write(nps,'(a1,3i1.1,') show') char(92),
  vpp    ivp(1),ivp(2),ivp(3)
c*vpp*e vpp
2. UPDATE ... No.2 'read/write' without format for integer variable & array vpp
c
c
c
c
c *****
c subroutine gansum(matd,a) acsr.3524
c *****
c
c*vpp*b ... 2 vpp
  equivalence (rvp1,nw),(rvp2,nv2) vpp
c*vpp*e vpp
c
c *****
c subroutine acasix acsr.8329
c *****
c
c*vpp*b ... 2 vpp
  equivalence (rvp1,ivmte) vpp
  equivalence (rvp2,nea),(rvp3,nea),(rvp4,nea) vpp
c*vpp*e vpp
c
c*vpp*b ... 2 vpp
  write(nout) mte,mte acsr.8414
c      ivmte-vte vpp
c      write(nout) ivmte,ivmte vpp
c*vpp*e vpp
c
c *****
c subroutine throd(nin) acsr.8632
c *****
c
c*vpp*b ... 2 vpp
  equivalence (rvp1,nea),(rvp2,nea) acsr.8632
c*vpp*e vpp
c
c *****
c subroutine acefix(nin,itpse,nout,ndir,ntype,iprint,npilot) acsr.9741
c *****
c
c*vpp*b ... 2 vpp
  dimension ivpzo8(16),vpzo(16) vpp
  equivalence (vpzo(1),ivpzo8(1)) vpp
c*vpp*e vpp
c
c*vpp*b ... 2 vpp
  read (nin) hz,aw0,tz,hd,hko,hm,(izo(i),i=1,16), vpp
  read (nin) hz,aw0,tz,hd,hko,hm,(ivpzo8(i),awo(i),i=1,16), njoy.655
c

```

```

1 (nzs(1),i=1,16),(jxs(1),i=1,32) acsr.9784
do 142 i=1,16 vpp
  izo(1)-ivpzo8(1) vpp
142 continue vpp
c*vpp*e vpp
c *****
c subroutine corr(nin,mat,mt,mti,mti,a) covr.531
c *****
c
c*vpp*b ... 2 vpp
  equivalence (vpig,ig) vpp
c*vpp*e vpp
c
c *****
c subroutine covard(nin,mat,mt,mti,mti,a) covr.647
c *****
c
c*vpp*b ... 2 vpp
  equivalence (vpk1,kl),(vp1gp1,1gp1),(vpk,k) vpp
c*vpp*e vpp
c
c *****
c subroutine finds(mat,mf,mt,ntape) covr.1550
c *****
c
c*vpp*b ... 2 vpp
  equivalence (vpmath,math),(vpmfh,mfh),(vpmth,mth) vpp
c*vpp*e vpp
c
c *****
c subroutine preas(nscr,nout,xa,nr,nc,a,ia) covr.1615
c *****
c
c*vpp*b ... 2 vpp
  equivalence (vpig,ig) vpp
c*vpp*e vpp
c
c *****
c subroutine findg(mat,mf,ntape) matzar.1261
c *****
c
c*vpp*b ... 2 vpp
  equivalence (vpmath,math),(vpmfh,mfh),(vpmth,mth) vpp
c*vpp*e vpp
c
c *****
c subroutine sklprz(nunit,prt) njoy.655
c *****
c

```



```

c      1lh=(11-1)/mult+1
c      1lh=11
c*vppe
c      *****
c      subroutine dldata(nin)
c      *****
c*vppe
c      jj=(11-1)/mult+1ao
c      jj=1+(1ao-1)
c*vppe
c      *****
c      subroutine ruin
c      *****
c*vppe
c      do 132 i=1,nzw
c      do 132 lednam(i)=iz(1)
c      j=1
c      do 132 i=1,nedpos
c      call vpchz(lednam(1),iz(j))
c      j=j+2
c      132 continue
c*vppe
c      *****
c      subroutine matxr
c      *****
c*vppe
c      real*8 chz
c*vppe
c      nu=nwds-1
c      do 110 i=1,nu
c      110 iz(1)=iz(1+i)
c      huse(1)=hz(1)
c      huse(2)=hz(2)
c      call vpchz(chz,z(2))
c      huse(1)=chz
c      call vpchz(chz,z(4))
c      huse(2)=chz
c*vppe
c      *****
c      subroutine cmatrix
c      *****
c*vppe
c      iscrh=(iscrt-1)/mult+1
c      iscrh=iscrt
c*vppe
c      *****
c*vppe
c      iholl=(iholl-1)/mult
c      iholl=iholl-1
c*vppe
c      ihfild=(ifild-1)/mult
c      ihfild=ifild-1
c*vppe
c      imatc=(imatc-1)/mult+1
c      imatc=imatc
c*vppe
c      iivcon=(iivcon-1)/mult
c      iivcon=iivcon-1
c*vppe
c      imacon=(imacon-1)/mult+1
c      imacon=imacon
c*vppe
c      *****
c      subroutine ruimm
c      *****
c*vppe
c      real*8 chz
c*vppe
c*vppe
c      do 125 i=1,nholl
c      nuds=mult*lim
c      do 130 j=1,lim
c      do 130 j=1,nuds
c      hz(j)=hblank
c      130 continue
c      if (ntry.gt.0) write(ntty,
c      1 '(//, enter set identification.//)')
c      nuds=mult*lim
c      call infree(nin,z(1),nuds,ncv)
c      loc=(iholl-1)/mult+lim*(i-1)
c      do 135 j=1,lim
c      135 ha(j+loc)=hz(j)
c      call vpchz(chz,z(2*(j-1)+1))
c      ha(j+loc)=chz
c      135 continue
c      125 write(naysc,71) (ha(j+loc),j=1,lim)
c*vppe
c*vppe
c      do 136 i=1,ipart
c      hz(1)=hblank
c      136 continue
c      call infree(nin,z(1),ipart,ncv)

```

vpp
matxr.580
vpp
vpp

vpp
matxr.593
vpp
vpp

vpp
matxr.624
vpp
vpp

vpp
matxr.653
vpp
vpp

vpp
matxr.699
vpp
vpp

matxr.753

vpp
vpp
vpp

vpp
matxr.802
vpp
matxr.803
vpp
matxr.804
matxr.805
matxr.806
matxr.807
matxr.808
matxr.809
matxr.810
vpp
matxr.811
matxr.812
vpp
vpp
matxr.813
vpp
vpp
vpp
vpp
matxr.820

ccccr.2859
vpp
vpp

ccccr.2858

vpp
ccccr.3030
vpp
vpp

dtfr.496

vpp
dtfr.636
dtfr.637
vpp
vpp
vpp
vpp
vpp

matxr.3

vpp
vpp
vpp

matxr.468
matxr.469
matxr.470
matxr.471
matxr.472
vpp
vpp
vpp
vpp
matxr.514

vpp
matxr.553
vpp
vpp

c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine vlsr	viewr.3
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	equivalence (z(1),iholl(1))	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine set2d(iplot,x,y,dm,dyp,dym,n,ierrb)	viewr.676
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	if (igrd.eq.1) call grid2(1,1)	viewr.888
c	c	if (igrd.eq.1) call grid2(1,1,0)	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	6. UPDATE ... No.6 reference to common area by integer array	
c	c	*****	
c	c	subroutine acload(nin,nedis)	acer.4636
c	c	*****	
c	c	cvpppb ... 6	vpp
c	c	equivalence (nxs(1),len2),(jxs(1),esz)	acer.4660
c	c	dimension vnxs(16),vjxs(32)	vpp
c	c	equivalence (nxs(1),vnxs(1)),(jxs(1),vjxs(1))	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine aceprt	acer.6230
c	c	*****	
c	c	cvpppb ... 6	vpp
c	c	equivalence (nxs(1),len2),(jxs(1),esz)	acer.6253
c	c	dimension vnxs(16),vjxs(32)	vpp
c	c	equivalence (nxs(1),vnxs(1)),(jxs(1),vjxs(1))	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine acout(itype,nace,ndir)	acer.7310
c	c	*****	
c	c	cvpppb ... 6	vpp
c	c	equivalence (nxs(1),len2),(jxs(1),esz)	acer.7327
c	c	dimension vnxs(16),vjxs(32)	vpp
c	c	equivalence (nxs(1),vnxs(1)),(jxs(1),vjxs(1))	vpp
c	c	cvppp*	vpp
c	c	*****	

c	c	5. UPDATE ... No.5 difference between 'real argument' and 'dummy argument'	
c	c	*****	
c	c	subroutine thnb(j,lnw,bufn,e,s,nx)	broadr.1080
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	dimension bufn(1)	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine lnfree(nin,z,nza,ncw)	njoy.394
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	equivalence (vpchar,char)	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine contio(nin,nout,nscr,a,nb,nw)	njoy.765
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	equivalence (vpis1,is1),(vpis2,is2)	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine lineio(nin,nout,nscr)	njoy.1376
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	equivalence (vpis,ia)	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine plotr	plotr.3
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	equivalence (z(1),iholl(1))	vpp
c	c	cvppp*	vpp
c	c	*****	
c	c	subroutine reconr	reconr.3
c	c	*****	
c	c	cvpppb ... 5	vpp
c	c	dimension vplabel(17)	vpp
c	c	equivalence (vplabel(1),label(1))	vpp


```

c *****
c
c*vpp*b ... 10
c ** insert dummy area **
common /wtabl/ tr(41,27),ti(41,27),trs(41,27),tis(41,27),
1 tdummy(64,51),kidummy
c*vpp*e
c
c
c *****
c
c subroutine wwtab2
c *****
c
c*vpp*b ... 10
c ** insert dummy area **
common /wtabl/ tr(41,27),ti(41,27),trs(41,27),tis(41,27),
1 tdummy(64,51),kidummy
c*vpp*e
c

```

vpp
vpp
pur.1297
vpp
vpp

pur.1875

vpp
vpp
pur.1881
vpp
vpp