

JAERI - M
90-146

HISTORIANを用いた大型計算コードの変換・整備

1990年9月

松本 潔・熊倉 利昌*

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

JAERI-M 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。
入手の問合せは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて、お申しこしください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 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。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division,
Department of Technical Information, Japan Atomic Energy Research Institute, Tokaimura,
Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1990

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

HISTORIANを用いた大型計算コードの変換・整備

日本原子力研究所東海研究所原子炉安全工学部

松本 潔・熊倉 利昌*

(1990年8月3日受理)

一般に、外部の機関で開発されたコンピュータコードを導入する場合、開発に用いられたコンピュータAと、導入・利用するコンピュータBとが異なるときには、A版からB版への変換が必要である。この変換に当たっては、従来の方法では、A版の内容を完全にB版に書き換えてしまうことが多かった。この場合、1) プログラムの内容がオリジナルのA版と完全に一致することはないので、書き換えられた部分についてオリジナルのステートメントが失われてしまった、2) これを解決するためにA版の書き換えたステートメントをコメント行として残しておくと、B版のソースプログラムが煩雑で読みにくいものになってしまった、3) 開発元から改良、変更、訂正等の連絡があったとき、また、こちらで改良等を行おうとしたとき、自分で書いたものではなく、かつ変換によって煩雑化してしまったプログラムの中から該当位置を探し書き換えることは、多大な時間と労力を必要とした、といった問題が生じていた。

これを解決するための1つの方法として、汎用ソースプログラム履歴管理ソフトウェアパッケージHISTORIANを用いた変換・整備方法を紹介する。この方法によって変換された計算コードは、改良、訂正、実験解析や感度解析のためのプログラムの変更等に対して、大変便利なものとなる。

本報告書は、HISTORIANを用いた変換・整備方法について解説し、実例としてPWR炉心溶融進展過程解析コードMELPROG-PWR/MOD1/V3のCRAYバージョンから原研FACOMバージョンへの変換・整備について報告するものである。本報告書は、今後大型のコードを開発あるいは導入するといった作業に従事している方々に有用なものとなろう。

Computer Code Conversion Using HISTORIAN

Kiyoshi MATSUMOTO and Toshimasa KUMAKURA*

Department of Reactor Safety Research
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received August 3, 1990)

When a computer program written for a computer A is converted for a computer B, in general, the A version source program is rewritten for B version. However, in this way of program conversion, the following inconvenient problems arise.

- 1) The original statements to be rewritten for B version are lost.
- 2) If the original statements of the A version rewritten for B version would remain as comment lines, the B version source program becomes quite large.
- 3) When update directives of the program are mailed from the organization which developed the program or when some modifications are needed for the program, it is difficult to point out the part to be updated or modified in the B version source program.

To solve these problems, the conversion method using the general-purpose software management aid system, HISTORIAN, has been introduced. This conversion method makes a large computer code a easy-to-use program for us to update, modify or improve after the conversion.

This report describes the planning and procedures of the conversion method and the MELPROG-PWR/MOD1 code conversion from the CRAY version to the JAERI FACOM version as an example. This report

* Nuclear Application and System Analysis Co. Ltd.

would provide useful information for those who develop or introduce large programs.

Keywords: HISTORIAN, Conversion, CRAY, FACOM, MELPROG

目 次

1. はじめに	1
2. HISTORIANの概要	2
3. HISTORIANを用いた変換の手順	3
4. HISTORIANを用いたMELPROG-PWR/MOD1コードの変換	4
4.1 入手したMELPROGコードの調査	4
4.2 原研大型コンピュータへの変換	5
4.2.1 変換方針	5
4.2.2 変換手順	6
4.2.3 変換作業項目	7
4.2.4 変換作業結果	9
4.3 テスト計算	11
4.4 変換版MELPLOGの使用方法	12
4.4.1 HISTORIAN使用方法	12
4.4.2 MELPROG使用方法	12
4.5 今後のバージョン・アップについて	13
4.6 問題点	14
5. おわりに	15
謝 詞	15
参考文献	16
付録 A MELPROG変換用ツールのFORTRANプログラム	55
付録 B MELPROG変換ツール用コマンドプロシジャー	64
付録 C MELPROG計算結果の図形処理機能の整備	68

Contents

1. Introduction	1
2. Overview of HISTORIAN	2
3. Program Conversion Methodology Using HISTORIAN	3
4. MELPROG-PWR/MOD1 Conversion Using HISTORIAN	4
4.1 Investigation of MOLPROG Source Program	4
4.2 Conversion to FACOM Computer in JAERI	5
4.2.1 Conversion Planning	5
4.2.2 Conversion Procedures	6
4.2.3 Tasks in Conversion procedures	7
4.2.4 Converted Results	9
4.3 Test Calculation by Converted MELPROG	11
4.4 How to Use Converted MELPLOG	12
4.4.1 How to Use HISTORIAN	12
4.4.2 How to Use MELPROG	12
4.5 Version-up of Converted MELPROG in Future	13
4.6 Some Problems in Conversion Experience	14
5. Concluding Remarks	15
Acknowledgment	15
References	16
Appendix A FORTRAN Source Lists of Tools for MELPROG Conversion	55
Appendix B Command Procedure Lists for MELPROG Conversion	64
Appendix C Implementation of Data Processing Capability for Graphics to MELPROG	68

図 表 一 覧

- 表 4. 1 提供されていない外部関数
 表 4. 2 新・旧シンボル名一覧
 表 4. 3 MELPLOG DECK 名と副プログラム名一覧
 表 4. 4 作成した副プログラム名一覧
 表 A. 1 変換作業用プログラムツール メンバ名一覧
 表 B. 1 変換作業用コマンドプロジェクト名一覧
 表 C. 1 FLUIDS 部の図形処理用データ
 表 C. 2 CORE 部の図形処理用データ
 表 C. 3 STRUCTURE 部の図形処理用データ
 表 C. 4 DEBRIS 部の図形処理用データ
- 図 2. 1 HISTORIAN によるソースプログラムの修正・管理の概要
 図 3. 1 変換・整備作業手順とソースプログラムの移り変わり
 図 4. 1 変換作業によるソースプログラムの移り変わり
 図 4. 2 副プログラム RMPROG の変換ソースプログラム
 図 4. 3 HISTORIAN ディレクティブの一部
 図 4. 4 COMPSOC バッチ用 JCL の例
 図 4. 5 変換・整備用ソースからのコンパイル、リンク JCL
 図 4. 6 変換・整備用ソースからの再リンク JCL
 図 4. 7 新バージョンソースからのコンパイル、リンク JCL
 図 4. 8 新バージョンソースからの再リンク JCL
 図 4. 9 入力データのエコーバック
 図 4. 10 図形処理情報のプリント出力制御 CPU タイム制御情報
 図 4. 11 図形処理のサイズ情報（各変数の次元の大きさを示す）
 図 4. 12 SCM, LCM 転送副プログラムの呼び出し回数と転送語数
 図 4. 13 第1回メジャーエディットの一部
 図 4. 14 最終メジャーエディットの一部
 図 4. 15 HISTORIAN を用いたジョブの流れ
 図 4. 16 HISTORIAN NEWLIB のクリエーションラン JCL
 図 4. 17 HISTORIAN アップデイトラン JCL
 図 4. 18 MELPLOG NEW ラン JCL
 図 4. 19 MELPLOG リスタートラン JCL
 図 4. 20 新バージョンのディレクティブが変換版と競合する場合
 図 4. 21 新バージョンのディレクティブが変換版と競合したときの修正例
 図 4. 22 新バージョン作成のため HISTORIAN アップデイト方法
 図 C. 1 SPL データ出力のための MELPLOG の図形データ処理フロー

- 図 C. 2 インクルード メンバ \$SPLCM
- 図 C. 3 SPL-PLOT による図形出力 JCL
- 図 C. 4 SPL-PLOT による図形出力の置換用メインプログラム
- 図 C. 5 図形処理出力の例 1 (燃料棒表面温度)
- 図 C. 6 図形処理出力の例 2 (冷却材流速)
- 図 C. 7 図形処理出力の例 3 (水素発生率)
- 図 C. 8 図形処理出力の例 4 (水素発生量の総和)

1. はじめに

一般に、外部の機関で開発されたコンピュータコードを導入する場合、開発に用いられたコンピュータAと、導入・利用するコンピュータBとが異なるときには、A版からB版への変換が必要である。この変換に当たっては、従来の方法では、A版の内容を完全にB版に書き換えてしまうことが多かった。この場合、以下のような問題が生じていた。

- 1) プログラムの内容がオリジナルのA版と完全に一致することはない。したがって、書き換えられた部分についてオリジナルのステートメントが失われてしまった。
- 2) 1)を解決するために、A版の書き換えたステートメントをコメント行として残しておくと、B版のソースプログラムが煩雑で読みにくいものになってしまった。
- 3) 開発元から改良、変更、訂正等の連絡があったとき、また、こちらで改良等を行おうとしたとき、自分で書いたものではなく、かつ煩雑になってしまったプログラムの中から該当位置を探し書き換えには、多大な時間と労力を必要とした。

これらを解決するための1つの方法として、汎用ソースプログラム履歴管理ソフトウェアパッケージ HISTORIAN¹⁾を用いる方法を紹介する。この方法によると、ソースプログラムに対して、変換に必要な変更等の命令を与えることによって新たなソースプログラムを作成することができる。さらに、変換後のコードの利用に当たって、特にソースプログラムに手を加える必要のある場合、すなわち、

- 1) プログラムの改良・訂正、
- 2) プログラムのバージョンアップ、
- 3) プログラムのある機能の感度解析に必要な変更、
- 4) 実験解析を行うために必要なプログラムの変更、

等々の場合も、書き換える必要のあるステートメントのみを外から与えることができ、作業時間の節約、プログラム管理の容易さからも大変便利である。

本報告書は、以上述べたHISTORIANを用いたプログラムの変換方法について報告するものである。第2章でHISTORIANの概要を紹介し、第3章では本変換方法について解説し、第4章では実際に行ったMELPROG-PWR/MOD1/V3の変換・整備を実例として詳細に報告する。

本報告書は、大型のコードを開発あるいは導入するといった作業に従事している方々に有益な参考になるものと考える。

2. HISTORIANの概要

HISTORIAN は、ソースプログラムや文字データ等の保管、修正、修正履歴の管理を行うことを目的として OPCODE 社（米国テキサス州）で開発されたソフトウェアパッケージであり、日本原子力研究所（以下「原研」という。）原子炉安全工学部が導入し、原研計算センターの大型コンピュータ FACOM M780 上で利用できるように変換・整備されている。

HISTORIAN の詳細については、参考文献（1）を参照していただくこととし、ここでは概要を紹介する。

図 2.1 に HISTORIAN によるソースプログラムの修正・管理の概要を示す。修正・管理の対象となるソースプログラムは、HISTORIAN の CREATION RUN によって「NEWLIB」と呼ばれる「ライブラリ(LIBRARY)」ファイルに再編成される。この NEWLIB は、修正等が行われる UPDATE RUN では OLDDLIB となり、OLDDLIB に対する修正等の指示データファイル「アップデート・ディレクティブ(Update Directive)」と共に HISTORIAN によって処理され、新たなソースプログラムが作成される。

HISTORIAN は、修正等が施される以前のデータも全てライブラリ内に保存、管理することができるため、利用者は必要に応じて任意の時点に戻ってライブラリを復元することができる。また、HISTORIAN は異なるコンピュータ間におけるライブラリの互換性を保証しているので、修正等の履歴を保存したまま異なったコンピュータ上での引き続いた管理を可能にしている。

ライブラリは、利用者によって定義された「デック(DECK)」と呼ばれるモジュールによって構成され、各々のデックは「テキストレコード(TEXT RECORD)」と呼ばれる論理レコードから成り立つ。これらは、かつて行われていたカードによる管理システムにおけるカードデック及び個々のカードにそれぞれ対応している。テキストレコードは、圧縮したカードイメージのデータに識別と履歴管理の目的で HISTORIAN が付け加えた制御文字を含んでいる。利用者がライブラリに対して行う全ての操作は、デックまたはテキストレコードを指定して実施される。

HISTORIAN はまた、「コモンデック(COMMON DECK or COMDECK)」と呼ばれるモジュールもライブラリの構成要素として持つことができる。コモンデックは、デックと同様の構造を持つが、デックとはその利用法において異なる。すなわち、いくつものデックで共通に使用されているようなテキストレコードを一度だけコモンデックとして定義することにより、他のデック及びコモンデック内にそれらを複写することができる。プログラムが複数のルーチンで同じ変数や配列等を含んでいる場合、これらの変数や配列等の修正には非常に手数を要し誤りも犯し易いが、コモンデックを用いてこれらの共通部分を一ヶ所で定義することにより手数及び誤りを減らすことができる。

データの修正等種々の HISTORIAN に対する指令は、「ディレクティブ(DIRECTIVE)」と呼ばれる HISTORIAN パッケージが提供する命令群を用いて行われる。HISTORIAN は、このディレクティブもライブラリに登録する。ディレクティブの登録は、いくつかにまとめられて「コレクションセット(CORRECTION SET)」単位に行われる。コレクションセットの定義は、利用者が自由に行うことができる。また、HISTORIAN は、コレクションセットの除

去・追加を利用者に自由に許可しているため、プログラムの開発段階で行われるバージョンの管理等に便利である。コレクションセットは、その名前と処理を行った時刻（順序）によって HISTORIAN により管理される。

3. HISTORIAN を用いた変換の手順

図 3.1 に変換の手順を示す。図中の丸で囲まれた数字は作業の順序を示しており、その時の変換対象ソースプログラムの移り変わりも同時に図に示してある。以下に図中の順序の数字に対応した変換手順を解説する。

- ① オリジナルソースプログラムから HISTORIAN の Creation Run により NEWLIB を作成する。
 - ② 1 DECK/メンバー の P0 (分割型順編成ファイル) のソースプログラムを作成し、さらに COMDECK をインクルードデータセットとして作成する。
 - ③ TSS 端末を利用して手作業にてプログラムの変換を実施する。
 - ④ コンパイル、リンク後サンプルデータによりテストランを実施する。このときサンプル出力と比較し、変換の正否をチェックする。
 - ⑤ ④により正しく変換が行われたことが確認できたら、その変換されたソースプログラムから HISTORIAN のアップデートのためディレクトイブを作る。
 - ⑥ ⑤で作成されたディレクトイブを使用して HISTORIAN の Update Run により OLDDLIB から COMPILE ファイルを作成する。
 - ⑦ コンパイル、リンク後サンプルデータによりテストランを実施する。
 - ⑧ ④で得られた結果と ⑦ の結果を比較して満足できる結果が得られたなら正しいディレクトイブが得られたことになり作業を終了する。
- この結果、変換されたプログラムとしては図中の a), b), c) の資源以外は不要となる。

以上が HISTORIAN を用いたプログラムの変換のあらましであるが、おそらくこれだけの解説では具体的なイメージが涌いてこないと思われる。次章以降に実際の変換例を紹介するので、そちらを参照していただいた方が分かりやすいであろう。

去・追加を利用者に自由に許可しているため、プログラムの開発段階で行われるバージョンの管理等に便利である。コレクションセットは、その名前と処理を行った時刻（順序）によって HISTORIAN により管理される。

3. HISTORIAN を用いた変換の手順

図 3.1 に変換の手順を示す。図中の丸で囲まれた数字は作業の順序を示しており、その時の変換対象ソースプログラムの移り変わりも同時に図に示してある。以下に図中の順序の数字に対応した変換手順を解説する。

- ① オリジナルソースプログラムから HISTORIAN の Creation Run により NEWLIB を作成する。
 - ② 1 DECK/メンバー の P0 (分割型順編成ファイル) のソースプログラムを作成し、さらに COMDECK をインクルードデータセットとして作成する。
 - ③ TSS 端末を利用して手作業にてプログラムの変換を実施する。
 - ④ コンパイル、リンク後サンプルデータによりテストランを実施する。このときサンプル出力と比較し、変換の正否をチェックする。
 - ⑤ ④により正しく変換が行われたことが確認できたら、その変換されたソースプログラムから HISTORIAN のアップデートのためディレクトイブを作る。
 - ⑥ ⑤で作成されたディレクトイブを使用して HISTORIAN の Update Run により OLDDLIB から COMPILE ファイルを作成する。
 - ⑦ コンパイル、リンク後サンプルデータによりテストランを実施する。
 - ⑧ ④で得られた結果と ⑦ の結果を比較して満足できる結果が得られたなら正しいディレクトイブが得られたことになり作業を終了する。
- この結果、変換されたプログラムとしては図中の a), b), c) の資源以外は不要となる。

以上が HISTORIAN を用いたプログラムの変換のあらましであるが、おそらくこれだけの解説では具体的なイメージが涌いてこないと思われる。次章以降に実際の変換例を紹介するので、そちらを参照していただいた方が分かりやすいであろう。

4. HISTORIANを用いたMELPROG-PWR/MOD1 コードの変換

MELPROG-PWR/MOD1²⁾は、PWR の炉心損傷事故時における炉心損傷溶融過程を詳細に解析するためのコードであり、米国サンディア国立研究所(SNL)でCRAY コンピュータ用に開発された。日本原子力研究所（原研）では、当該コードを米国原子力規制委員会(U. S. NRC)を通じて入手し、原研の大型コンピュータ FACOM M-780 に変換・整備し、併せて図形出力機能を追加することとした。本コードは、MELPROG-PWR/MOD1/V3 という開発途中のバージョンであり、今後頻繁に開発元から改良・改訂の情報が送られてくることが予想される。したがって FACOM M-780への変換に当たっては、これに迅速な対応をとることができると対策をとっておく必要がある。そこで、本変換・整備では、HISTORIAN を用いる方法を採用した。

以下に HISTORIAN を用いた MELPROG コードの CRAY 版から原研 FACOM 版への変換・整備の実際について作業の順を追って詳細に解説する。なお、解説上の都合から第3章までに記述した内容が重複して現れることがあることをお断りしておく。

4. 1 入手した MELPROG コードの調査

MELPROG-PWR/MOD1 は、PWR の炉心損傷事故時における炉心溶融進展過程を解析するために、米国サンディア国立研究所で開発された CRAY 計算機用の計算コードである。

入手した MELPROG-PWR/MOD1 は CRAY 計算機用の計算コードであり、開発途中の MELPROG-PWR/MOD1/V3 バージョンとして入手した。入手した資料は次の通りである。

- ① MELPROG-PWR/MOD1/V3 ソースプログラム (34,906 ライン)
- ② サンプル入力データ 1 (定常計算用) (792 ライン)
- ③ サンプル入力データ 2 (過渡計算、リスタート用) (277 ライン)
- ④ メカニックス入力データ
- ⑤ サンプル計算用ソースプログラム アップデート・カード
- ⑥ サンプル出力 (リスト)
- ⑦ インプットマニュアル及び導入コメント

ソースプログラムは HISTORIAN と同様の機能を持った CRAY 計算機用のソフトウェア・パッケージ UPDATE によりコーディングされている。ソースプログラムの構成は、

- (1) COMDECK COMDECK 数 34, そのライン数 379
- (2) DECK DECK 数 164, そのライン数 34,527

であり、総ライン数は 34,906 である。ここで、DECK とはサブルーチン等の一般的なプログラムのモジュールであり、COMDECK とはコモンデックと呼ばれる HISTORIAN(UPDATE)特有のモジュールである。すなわち、プログラム内各所で共通に使われるテキストレコードを

一度だけ COMDECK として定義しておけば、他の DECK および COMDECK 内で呼び出すことにより利用できるというものである。この他に、今回ソースプログラムとして提供はされていないが必要な外部関数として表 4.1 に示すものがある、という情報が得られた。特にバンドマトリクス解法ルーチン (BGLSDC, BGLSSL) は開発元に特有のもので、副プログラム DSOL の IMETH=3 とするとそれらは利用されず SLATECライブラリの他の副プログラムが代用できるという情報が提供された。

付録 A に示したプログラムを利用した調査および STREAM77(FACOM 提供のプログラム変換・解析ソフトウェアツール)³⁾を用いたプログラム解析により以下のことがわかった。

- ① 単精度演算である。
- ② 未定義関数が、提供された情報の他に EOF, ISELOND, SAMPTRM, DSLVIC および VIKI があり、表 4.1 に示した未定義外部関数のうち AND および SPTSL が未定義ではなかった。
- ③ 6 文字を超えるシンボル名（変数／配列名、副プログラム名、コモンブロック名）が多数存在する。
- ④ コモンブロック名と副プログラム名の一一致がある。
- ⑤ UPDATE 機能を利用したソースプログラムの選択がある。

さらに、計算結果を図形処理するための出力が、原研が所有する図形処理ツールに都合のよいものではないので、これに変わる図形処理用出力を整備することとした。ただし、これに関しては、変換作業とは異なるので、解説を付録 C に掲載することとした。

4. 2 原研大型コンピュータへの変換

4. 2. 1 変換方針

4. 1 の調査結果をもとに文法上の相違による変換・整備を行い、かつ精度上の理由から FACOM 版では倍精度化を実施する。これまで CDCコンピュータ あるいは CRAY コンピュータ用計算コードの FACOM コンピュータへの変換にはほとんど倍精度化を実施しており、変換作業においてはこの倍精度化が作業のほとんどを占めている。又、計算コードの変換・整備作業では、その変換を実際のプログラム上に施しており、その欠点としてはソースプログラムが非常に見にくいものになってしまふことがあげられる。次に UPDATE で書かれたプログラムの場合、開発元から提供される新バージョンはアップデート・ディレクトイブ（修正指示カード）のみであり、変換・整備後のソースプログラムのバージョンアップに際しては、一般にそのディレクトイブとソースプログラムを比較して手作業でバージョンアップを行っている。これでは新バージョンの作成に迅速に対処できない。もう一つの欠点は、各研究者が自分用の改造を施したいとき、いくつものソースプログラムが存在することになり、又バージョンアップ毎にその改造を新しいプログラムに加えなければならない。このことを解決するため本変換・整備では以下の方法を採用する。

- ① FACOM の FORTRAN-77 のコンパイラがもつ自動精度拡張機能を利用して倍精度化を実施し、変換・整備ラインの減少をはかる。
- ② HISTORIAN を利用した変換・整備を実施する。

この方法の採用には 2 つの問題点がある。1 つは自動変換ツール (STREAM 77 等) が利用できず、ほとんどが手作業となるため変換作業量が多くなる。変換ツールが HISTORIAN のアップデート・ディレクティブを作成してくれれば問題はないが現在のところまだそのようなツールがない。もう 1 つの問題点はデバッグが容易にかつ迅速に出来ないことである。前者については自動精度拡張機能を利用するためその変換ラインが軽減でき、後者については下に述べる変換手順により問題を解決する。

4. 2. 2 変換手順

変換手順を計画するに当たり、前項 4.2.1 で述べたオリジナル・ソースプログラムの変換を指示する HISTORIAN 用のアップデート・ディレクティブを如何に作成するかという問題を解決しておかなければ手順は決まらない。これには、アップデート・ディレクティブを直接書き下すという方法が最も単純ではあるが、書き下す内容が分脈性のあるものではなく、見落とし、変数名の写し間違い等の可能性が非常に高く、労を多くして益の少ない方法である。そこで、図 4.1 に示すように、先ずオリジナルのソースプログラム上の書換えを必要とする箇所にその変換をほどこし、これを入力として予め作成しておいたツールを用いて HISTORIAN 用のアップデート・ディレクティブを得る、という方法の方を考えた。この方法によればより分かりやすく間違いが少なくて済むであろう。HISTORIAN 用のアップデート・ディレクティブを作成するプログラムツールは、付録 A ⑤ DIRECTIVE である。図 4.1 には 3 つの状態のソース・プログラムの例をあげている。第一はオリジナルのソースプログラム、第二は書換えデバッグ中のソースプログラム、そして最後は変換バージョン作成のための HISTORIAN のアップデート・ディレクティブを各々示している。

*CONVSTR 及び *CONVEND は変換ツール DIRECTIVE のための命令であり、それぞれ「変換開始」、「変換終了」を意味する。この 2 つの命令にはさまれた部分が変換のために書き換えられた部分であり、変換ツール DIRECTIVE を用いてこのソースプログラムを処理すると、アップデート・ディレクティブが作られる。ここで、

*I SUBA.10

は、「ライン SUBA.10 の次に "CALL MTRACE('SUBA ')" を挿入する」、また、

*D SUBA.11.12

は、「SUBA.11 から SUBA.12 を削除し、以下のステートメントと置き換える」というのみである。この例では、デバッグのための副プログラム実行フラグを設定する MTRACE 副プログラムの呼び出しと、7 文字シンボル名の変更にともなう変換を示している。

以上述べた HISTORIAN 用アップデート・ディレクティブ作成法を前提に、図 3.1 に示した順序の数字に対応した変換手順を解説する。

①オリジナルソースプログラムから HISTORIAN の Creation Run により NEWLIB を作成する。ただし、COMDECK の引用に当たっては HISTORIAN の *CALL ディレクティブによらず FORTRAN77 の INCLUDE ディレクティブを利用する。このため NEWLIB を作成するとき前処理として *CALL を *INCLUDE で行い、また COMDECK 部分を除いたもので行う。

②STREAM77 の PENGINE を利用して 1 DECK/メンバー の P0 (分割型順編成ファイル) のソースプログラムを作成し、さらに COMDECK をインクルードデータセットとして作成す

- る。同時に CENGINE を用いてプログラミングの解析情報を得る。
- ③ TSS 端末を利用して手作業にてプログラムの変換・整備を実施する。
- ④ コンパイル、リンク後サンプルデータによりテストランを実施する。このときサンプル出力と比較し、変換の正否をチェックする。
- ⑤ ④により正しく変換が行われたことが確認できたら、そのソースプログラムを入力として、作成したツール（付録 A の ⑤ DIRECTIVE）を用いて HISTORIAN 用アップデイト・ディレクティブを作る。
- ⑥ ⑤で作成されたディレクティブを使用して HISTORIAN の Update Run により OLDLIB から COMPILE ファイルを作成する。
- ⑦ コンパイル、リンク後サンプルデータによりテストランを実施する。
- ⑧ ④で得られた結果と ⑦ の結果を比較して満足できる結果が得られたなら正しいディレクティブが得られたことになり作業を終了する。
- この結果、変換されたプログラムとしては図中の a, b, c の資源以外は不要となる。

4. 2. 3 変換作業項目

以上の手順を実行するために必要な作業項目を以下に示す。なお、以下の番号は上記の手順番号と対応するものではない。

- 1) COMDECK を FORTRAN の Include に変更する。
- 2) 1 DECK 毎に 1 メンバーの P0 ファイルを作成する。
- 3) プログラミング解析情報を得る。
- 4) コンパイルエラー情報を得る。
- 5) 倍精度化を実施する。
- 6) 6 文字を超えるシンボル名を 6 文字化する。
- 7) COMMON 変数／配列への初期値設定を BLOCK DATA 副プログラムへ移動を行う。
- 8) 未定義副プログラムの作成及び FACOM コンピュータで模擬する。
- 9) 変換・整備用デバッグのための整備を行う。
- 10) その他文法上のエラー発生とならない計算機上の違いによる変更をおこなう。
- 11) オリジナルのプログラミングエラーの修正を行う。
- 12) 変換・整備作業に必要なソフトウェアツールを開発する。

各項目の実施方法を以下に記す。

- 1) STREAM77 の PENGINE を利用する。
- 2) 同上
- 3) STREAM77 の CENGINE を利用して、副プログラム参照関係、木構造、未定義副プログラム、7 文字シンボル名等の情報を得る。
- 4) FORTRAN77 コンパイラーによりコンパイル情報を得る。
- 5) FORTRAN77 のコンバイラーオプションの自動精度拡張機能を利用する。その方法はパラメータで AUTODBL (DBLPAD) を指定することでなされる。この自動精度拡張機能は次の方法で行われる。⁴⁾

AUTODBL オプション : 自動精度拡張機能を動作させる場合には、AUTODBL オプションを指定しなければならない。オプションの記述は次のようにする。

AUTODBL(value)

ここで、value は、以下の 8 種のパラメタのいずれか一つである。

- NONE
- abcde
- DBLPAD
- DBLPAD4
- DBL
- DBL4
- DBK8

パラメタの記述に誤りがあった場合は、NONE とみなされる。

DBLPAD サブパラメタ

実数型、倍精度実数型、複素数型及び倍精度複素数型の変数、定数、配列及び関数を精度拡張の対象にする。また、これらのものと記憶領域を共有するデータや共通ブロックに属するデータ及び引数として使用されているデータで、論理型、整数型、4 倍精度実数型及び4 倍精度複素数型のものを記憶領域合わせの対象にするものである。(abcdeサブパラメタで 33336 を意味する)

精度拡張（プロモート）： 精度拡張は、実数型、倍精度実数型、複素数型及び倍精度複素数型の定数、変数、配列又は関数に対して行うことができる。

精度拡張が指定された場合、データの型は、次のように変換される。

実数型	→倍精度実数型
倍精度実数型	→4 倍精度実数型
複素数型	→倍精度複素数型
倍精度複素数型	→4 倍精度複素数型

6) STREAM77 の CENGINE により 6 文字を超えるシンボル名の一覧を得る。6 文字を超えるシンボル名は後から母音を取り除く方法でシンボル名を決定し、そのシンボル名がプログラム内でユニークなものであるか否かを STREAM 77 を利用してチェックする。

7) FORTRAN77 によるコンパイルの結果エラーの発生した DATA 文より、移動すべきデータ

を検出する。

- 8) STREAM77 で得られた未定義ルーチンとソースプログラムからその機能を把握し、処理上不可欠なものは作成又は他から引用し、不必要的ものはダミー化する。FACOM 機上で模擬するものは本体にもその模擬のための変更をほどこす。
- 9) プログラムの動的動作を認識するためのデバッグ副プログラムとそれを呼び出す文を各副プログラムの先頭に追加する。
- 10) EQUIVALENCE とされた整数型変数／配列と実数型変数／配列に対して、次のような引用があるものは各々対応した変換を行う。例えば、A と IA が EQUIVALENCE とされているとき、

$$\begin{array}{ccc} \text{IA}(n)=\text{IA}(m) & \xrightarrow{\quad} & A(n)=A(m) \\ \text{IA}(n)=0 & \xrightarrow{\quad} & A(n)=0.0 \end{array}$$

と変更する。又、実引数が定数で与えられ、仮引数でその値に変更を加えるときは実引数の定数を変数にする。例えば、

```
CALL A(A1,A2,1,A4)
↓
KDUM=1
CALL A(A1,A2,KUDM,A4)
```

とする。ただし、正確に模擬するためには KDUM を DATA 文で与えなければならない。

- 11) これは、オリジナルバージョンで使用されていたときには発見されなかったエラーが、変換作業中に発見され修正されるというもので、変換作業ではよくあることである。この種のエラー修正は、オリジナルバージョンにも直ちに反映されるべきである。
- 12) これは、ツールとしてのプログラムを作成しこれを用いて行った方が効率の良い作業に関してプログラムツールを予め作っておこうというものである。

4. 2. 4 変換作業結果

(1) 変換作業中のソースプログラム例

図 4.2 に副プログラム RMPPRG の変換ソースプログラム例をあげ、又、図中にその変換理由をあげる。

(2) 作成した HISTORIAN のディレクティブ

図 4.3 に付録 A ⑤のツール DIRECTIV により作成した変換用ディレクティブの一部をあげる。

(3) 6 文字を超えるシンボル名と変換後の新シンボル名

表 4.2 に 6 文字を超えるオリジナルのシンボル名と変換後のシンボル名一覧を、変数／配列名（共通（コモン）ブロック内、ローカル）、副プログラム名および共通ブロックについて示す。なおこの情報は、MELPROG のメインプログラム MDRIVER 上にある。

(4) DECK 名と副プログラム名の一覧

表 4.3 に MELPROG の DECK 名とその DECK 内の副プログラム名および副入口名の一覧を示す。なおこの情報は MELPROG のメインプログラム MDRIVER 上にある。

(5) 作成した未定義副プログラム他

表 4.4 に本作業で引用もしくは作成した副プログラムの一覧を示す。表中のDECK 名の (U) および (G) は各々未定義副プログラム、変換・整備の必要上作成したプログラムを表している。未定義副プログラムのうちマトリクス解法ルーチンのほとんどは、PWR 熱水力挙動最適予測解析コード TRAC から引用したものであり、最初の未定義副プログラム名より多く存在しているが、これは 2 次、3 次引用で必要となった副プログラムが存在したためである。副プログラム SGTSL は TRAC にも存在せず、IBMコンピュータ用の LINPACK ライブラリから引用したものである。(G)で示される副プログラムのうち、ECHO は入力データのエコーバック、MABEND はダミールーチンが呼び出されたときなどの異常処理、MTRACE は副プログラム動的実行のトレース、BLKDA および BLKDB は共通ブロック変数／配列への初期値設定、SPLXXX は計算結果の図形処理のためのデータ処理を行うものである。なお SPLXXX ルーチンに関しては付録 C で詳しく述べる。

(6) 変換・整備後のソース・プログラムのライン数

変換・整備後のソース・プログラムのライン数は、以下の通りとなった。

①オリジナル・ソースプログラムのライン数	
・ COMDECK ライン数	379
・ ソースプログラムライン数	34527
②変換・整備作業中のライン数	
・ Include ライン数	519
・ ソースプログラムライン数	39441
③作成したライブラリのライン数	
・ ソースプログラムライン数	2327
④変換・整備用ディレクトイブ	
・ COMDECK用 ライン数	64
・ ソースプログラム用 ライン数	4329
⑤HISTORIAN を利用してのソースプログラムのライン数	37343

(7) 作成したソフトウェアツールとその使い方

付録Bにソフトウェアツールを利用するときのコマンド・プロシージャを載せた。なお、主なソフトウェアツールの FORTRAN リストは、付録Aに載せてある。各々の利用方法は各付録を参照されたい。以下に簡単な説明を記す。

- ①CHGSYM : ディレクティブの 6 文字を超えたシンボル名の変更
- ②COMPSOC : ソースプログラムリストのコンパクト出力
- ③DIRECTIV : 変換・整備ソースプログラムからディレクティブを作成
- ④UPDCHG : COMDECK, DECK のCARDID再番号付
- ⑤UPDLIST : ディレクティブのディレクテブラインのみのリスト出力

以上はすべて端末上で処理するものであるが、COMPSOC だけは、この他に図4.4 に示す JCL (ジョブ制御文) によりバッチ処理をする。

(8) MELPROG のコンパイル、リンク用 JCL

図 4.5 から図 4.8 に MELPROG のコンパイル、リンク用バッチ JCL を示す。

4. 3 テスト計算

MELPROG の変換・整備が正しく行われたことを確認するため、MELPROG-PWR/MOD1/V3 を入手したときに提供されたサンプルデータについてテスト計算を実施した。このサンプルデータは 2 ケースあり、その内の 1 つは、PWR 原子炉実体系の定常計算用データである。もう 1 つは、この定常計算の結果からリスタートする過度計算用データである。テスト計算の結果、定常計算ではサンプル出力に良く一致した結果が得られた。リスタート計算では、燃料棒の温度の計算結果についてはサンプル出力とほとんど一致した結果が得られ、他の物理量の計算結果についても、サンプル出力と差が出た場合でも出力桁 6 術の内、上 4 術以上は一致した。このサンプルデータでは温度にほとんど変化がなく、過渡解析として比較すべきデータがなかった。そこで、温度が上昇するような過渡計算となる実験のデータを利用して、計算結果にどのような影響を与えるか、SNL の CRAY で計算された結果と比較するテスト計算を実施した。本計算ではタイムステップ数が CRAY の結果と一致し、計算結果の諸変数についても良く一致した結果が得られた。図4.9 は入力データのエコーバック、図 4.10 は図形処理情報のプリント出力制御情報と計算時間制御情報、図 4.11 は図形処理のサイズ情報、図 4.12 はデータ転送用副プログラム SCM, LCM の呼び出し回数と転送語数の出力情報であるが、これらは変換・整備作業で追加した機能であり、MELPROG を利用する上で非常に重要なものであるため、変換・整備終了後もチェック用にプリントするようにしている。上記テスト計算の出力の中から実験解析計算について、図 4.13 に第 1 回目のメジャーエディットの一部を、図 4.14 に最終回のメジャーエディットの一部を示す。

4. 4 変換版 MELPROG の使用方法

4. 4. 1 HISTORIAN 使用方法

MELPROG の変換・整備バージョンを得るには HISTORIAN を使用しなければならない。必要な資源は MELPROG オリジナル版のソースプログラム（実際には COMDECK 部を除き、かつ共通部分の展開には *CALL でなく *INCLUDE に加工したもの）と、それから作成される HISTORIAN の NEWLIB と、変換・整備用のアップデート・ディレクティブである。

図 4.15 に、これらジョブのフロー図を示す。また、図 4.16 に HISTORIAN の NEWLIB 作成の JCL をコンパクト出力とプリンタ出力の 2 つの例について示す。さらに、図 4.17 には HISTORIAN を使用した変換・整備バージョンあるいは、その変換・整備バージョンと新バージョンの同時変更（開発元からの新バージョンあるいは利用者毎の改良バージョンも含む）のアップデイトランの JCL を同様にコンパクト出力とプリンタ出力の 2 つの例について示す。アップデイト・ディレクティブは HISTORIAN のコマンド *READ 1 により FT-01F001 の DD 文で指定されるが、いくつものアップデイトが必要なときはその DD 文を連結して指定すれば良い。

4. 4. 2 MELPROG 使用方法

MELPROG の使用方法は提供された入力マニュアル及び導入コメントにあるため、ここでは変換・整備により追加・変更されたファイルとデータを主体に、その使用方法について述べる。必要な DD 文についてのコメントは図 4.18 と図 4.19 の JCL 例の中にコメントとして記述してある。図 4.18 には MELPROG の NEW ラン（初期実行）の例として、サンプル入力データの定常計算データと実験解析用データの 2 つの計算の JCL 例をあげ、図 4.19 にはリストアートランの JCL 例をあげた。

(1) 追加した DD 文とそのデータの指定方法

以下に追加した DD 文とそのデータの指定方法について述べる。この中には図形処理に関する説明もされているが、その詳細については付録 C を参照されたい。

① FT99F001

当該ジョブの使用可能 CPU 時間と SPL 図形情報の出力制御データを指定する。

(a) 第 1 枚目のカード

当該ジョブの Resource 指定の CPU 時間指定 T.n の n を与える（フリー-フォーマット）と n の値により計算可能な CPU 時間が設定される。

(b) 第 2 枚目のカード

SPL 図形情報の出力制御データであり、2 つのデータが必要（フリー-フォーマット）である。

(i) 第 1 データ

3 ビットを使用して 3 つの情報出力を制御している。各ビットが "1" なら

出力要請、"0"なら出力抑止を意味し、この3ビットのビット状態により10進数に変更してデータを指定する。例えば1ビット目と3ビット目を"1"とするなら $1+4=5$ を指定する。以下に各ビットと対応する情報について記述する。

- ・第1ビット目 SPL ラベル情報の出力 (1)
- ・第2ビット目 図形用データ出力物理時刻情報の出力 (2)
- ・第3ビット目 SPL 変数内アドレスとプログラム変数の対応 (4)
関係情報の出力 (データ構造が複雑なため)

(ii) 第2データ

これは変換・整備中のデバッグ情報の出力指定であるため、一般には0を指定のこと。

②FT06F001

FORTRAN のエラー処理メッセージや変換・整備の情報出力のためのDD文である。変換・整備の出力には、出力制御される SPL 情報の他に、入力データのエコーバック、1タイムステップあたりの CPU 時間や残り CPU 時間のチェック時間、図形処理データの大きさのサマリー情報、エラーメッセージ、および LCM, SCM 間模擬による転送語数情報等がある。

③FT21F001

図形処理のための SPL-PLOT, ROPS 用バイナリデータファイルである。入力データでグラフ出力オプションにより出力される。

(2) 主な DD 文の説明

以下に主な DD 文を説明する。

- | | | |
|-------------|-------------------------|------|
| ①FT95F001 : | 入力データファイル (5番にかわるもの) | (入力) |
| ②FT96F001 : | 6番にかわるプリント出力ファイル | (出力) |
| ③FT93F001 : | リストアート入力ダンプファイル | (入力) |
| ④FT98F001 : | リストアート出力ダンプファイル | (出力) |
| ⑤FT97T001 : | Mechanical Property データ | (入力) |

4, 5 今後のバージョン・アップについて

MELPROG の変換・整備バージョン、即ち FACOM バージョンは、第3章に記載したように、HISTORIAN によって作成される。変換のためには、オリジナル・ソースプログラムの他に FACOM バージョンを作成するためディレクティブが必要であり、これらを入力として HISTORIAN によってオリジナルのラインの削除及び新しい CARD ID でのラインの追加が行われる。

したがって、今後開発元から新バージョンのディレクティブが提供され、それを FACOM バージョンに反映させようとするとき、そのディレクティブに修正を加える必要が生じる場合がある。それは、図 4.20 に示すような変換・整備箇所と新バージョンの修正箇所が

競合するとき発生する。この場合、新バージョンのディレクティブの CARD ID を、新しく作成されるソースプログラムの新 CARD ID に変更しなければならない。例を図 4.21 に示す。この例では、変換版ソースプログラムが EXAM.8 の代わりに TEST.2 となっているため、提供された新バージョン・ディレクティブの *D EXAM.8,9 を *D TEST.2,EXAM.9 と変更している。また、修正ラインに変換が必要なときは、直接そのラインを変換し、ライン数の増減が生じないように配慮しなければならない。これは、さらに次の新バージョンのディレクティブの CARD ID との不一致を防止するために必要なことである。

新バージョンの変換版 (FACOM 版) を作成するには、図 4.22 に示すように変換・整備ディレクティブの後にファイル連結により、順次新バージョンのディレクティブ用ファイルを定義すれば良い。ただしその都度 HISTORIAN の Creation Run により NEWLIB を順次作成した場合はこの限りでない。

4. 6 問題点

以下に本変換・整備において筆者らが感じた問題点とその対処法等について記す。

(1) FORTRAN コンパイラの違いによる問題

これは CRAY FORTRAN コンパイラ CFT と FACOM FORTRAN コンパイラ FORTRAN77 の相違によってプログラム動作に違いが生じるという問題である。副プログラムの実引数が定数で、かつ呼び出される副プログラム上でその定数に値を設定しても、その引数毎にその値のための領域がとられ、他に何も影響を起こさないのが CFT コンパイラである。ここでは、引数に限らず定数は全てその値が等しいか否かにかかわらず別々の領域がとられると想像されるが、その定数領域が変更されるのは実引数に現れた場合のみである。これに対して、FACOM の FORTRAN77 コンパイラは、等しい定数にはプログラム内唯一の領域のみが確保されて様々な用途に利用されるため、絶対に上述のような仮引数の変更を行ってはならない。このような場合には、新たな変数を与え、初期値にその定数を設定しなければならない。MELPROG で具体的にこの様なコーディングがされていたものは内挿ルーチンである。副プログラム LINITNX (LINNNTX) であった。

(2) 図形処理で実行中にその大きさが増加する場合の問題

図形処理で SPL-PLOT もしくは ROPS を利用するに当たっては、実行に先だって SPL 変数の大きさが判明しており、かつ不变でなければならない。しかし CORE モジュールの部分では、再メッシュ化によりその大きさが増加する。これに対処するため、あらかじめ初期のメッシュサイズより 10だけ大きい値とした。

(3) HISTRIAN を利用した変換・整備上の問題

HISTRIAN を利用した変換・整備は、筆者らにとって初めての試みであった。この

ため、共通部分の展開に関しては作業の効率上 HISTORIANソースプログラムについても *CALL ディレクティブを利用せず FORTRAN *INCLUDEを利用した。作業終了後に判明したことであるが、HISTORIAN で *CALL ディレクティブを利用するべきであり、作業の迅速さに何ら影響しないことがわかった。

5. おわりに

大型コンピュータコードの変換後の利用の便を考え、汎用履歴管理ソフトウェアパッケージ HISTORIAN を用いた変換方法を採用した。そして、この方法によって、

- ①オリジナル版のソースプログラム
- ②変換に必要な変更部分のみのソースプログラム
- ③変換版のソースプログラム

が作成された。これによって、変換後のコードの利用に当たって、特にソースプログラムに手を加える必要のある場合、すなわち、

- 1) プログラムの改良・訂正、
- 2) プログラムのバージョンアップ、
- 3) プログラムのある機能の感度解析に必要な変更、
- 4) 実験解析を行うために必要なプログラムの変更、

等々の場合も、書き換える必要のあるステートメントのみを外から与えることができ、作業時間の節約、プログラム管理の容易さからも大変便利である。

今回の変換・整備の結果、HISTORIAN を用いた変換方法は、特に異機種のコンピュータ用に開発されたプログラムを導入する場合、非常に有効な方法であることが分かった。本報告書が、今後 MELPROG を利用する読者のみならず、大型のコードの開発あるいは導入・整備といった作業に携わる方々のお役に立てれば幸いである。

謝辞

本変換・整備に当たり、HISTORIAN および TRAC コードの副プログラムを快く提供下さった原子炉工学部伝熱流動研究室（当時、原子炉安全工学部安全工学第2研究室）の諸氏に感謝の意を表します。

ため、共通部分の展開に関しては作業の効率上 HISTORIAN ソースプログラムについても *CALL ディレクティブを利用せず FORTRAN *INCLUDE を利用した。作業終了後に判明したことであるが、HISTORIAN で *CALL ディレクティブを利用すべきであり、作業の迅速さに何ら影響しないことがわかった。

5. おわりに

大型コンピュータコードの変換後の利用の便を考え、汎用履歴管理ソフトウェアパッケージ HISTORIAN を用いた変換方法を採用した。そして、この方法によって、

- ①オリジナル版のソースプログラム
- ②変換に必要な変更部分のみのソースプログラム
- ③変換版のソースプログラム

が作成された。これによって、変換後のコードの利用に当たって、特にソースプログラムに手を加える必要のある場合、すなわち、

- 1) プログラムの改良・訂正、
- 2) プログラムのバージョンアップ、
- 3) プログラムのある機能の感度解析に必要な変更、
- 4) 実験解析を行うために必要なプログラムの変更、

等々の場合も、書き換える必要のあるステートメントのみを外から与えることができ、作業時間の節約、プログラム管理の容易さからも大変便利である。

今回の変換・整備の結果、HISTORIAN を用いた変換方法は、特に異機種のコンピュータ用に開発されたプログラムを導入する場合、非常に有効な方法であることが分かった。本報告書が、今後 MELPROG を利用する読者のみならず、大型のコードの開発あるいは導入・整備といった作業に携わる方々のお役に立てれば幸いである。

謝辞

本変換・整備に当たり、HISTORIAN および TRAC コードの副プログラムを快く提供下さった原子炉工学部伝熱流動研究室（当時、原子炉安全工学部安全工学第 2 研究室）の諸氏に感謝の意を表します。

ため、共通部分の展開に関しては作業の効率上 HISTORIANソースプログラムについても *CALL ディレクトイブを利用せず FORTRAN *INCLUDEを利用した。作業終了後に判明したことであるが、HISTORIANで *CALL ディレクトイブを利用すべきであり、作業の迅速さに何ら影響しないことがわかった。

5. おわりに

大型コンピュータコードの変換後の利用の便を考え、汎用履歴管理ソフトウェアパッケージ HISTORIAN を用いた変換方法を採用した。そして、この方法によって、

- ①オリジナル版のソースプログラム
- ②変換に必要な変更部分のみのソースプログラム
- ③変換版のソースプログラム

が作成された。これによって、変換後のコードの利用に当たって、特にソースプログラムに手を加える必要のある場合、すなわち、

- 1) プログラムの改良・訂正、
- 2) プログラムのバージョンアップ、
- 3) プログラムのある機能の感度解析に必要な変更、
- 4) 実験解析を行うために必要なプログラムの変更、

等々の場合も、書き換える必要のあるステートメントのみを外から与えることができ、作業時間の節約、プログラム管理の容易さからも大変便利である。

今回の変換・整備の結果、HISTORIAN を用いた変換方法は、特に異機種のコンピュータ用に開発されたプログラムを導入する場合、非常に有効な方法であることが分かった。本報告書が、今後 MELPROG を利用する読者のみならず、大型のコードの開発あるいは導入・整備といった作業に携わる方々のお役に立てれば幸いである。

謝辞

本変換・整備に当たり、HISTORIAN および TRAC コードの副プログラムを快く提供下さった原子炉工学部伝熱流動研究室（当時、原子炉安全工学部安全工学第2研究室）の諸氏に感謝の意を表します。

参考文献

- (1) 株式会社エリック "HISTORIAN 使用者マニュアル"
- (2) S. S. Dodanjh et. al, "MELPROG-PWR/MOD1 : A Two-Dimensional, Mechanistic Code for Analysis of Reactor Core Melt Progression and Vessel Attack Under Severe Accident Conditions," NUREG/CR-5193 SAND88-1824 (1989)
- (3) 富士通株式会社, "FACOM OS IV/F4 MSP STREAM77 説明書 (FORTRAN プログラム変換支援パッケージ/CRAY・CDC FORTRAN 用)"
- (4) 富士通株式会社, "FACOM OS IV/F4 MSP FORTRAN77 使用手引書"

表4.1 提供されていない外部関数

FUNCTION	PURPOSE
GETJTL	Used to determine if job time limit exceeded (CTSS specific - may be deleted)
SYSCALL	Used to determine if job time limit exceeded (CTSS specific - may be deleted)
SECOND	Get number of CPU seconds from start of job
MOVLEV	Moves data from one array to another
AND (N,1)	Boolean bit-by-bit multiplier. May replace with IAND (N,1) or MOD (N,2)
SGEFA	SLATEC routine
SGESL	SLATEC routine
SCEEV	SLATEC routine
SPTSL	SLATEC routine
BGLSDC	Banded Matrix Decomposition Routine
BGLSSL	Banded Matrix Back Substitution Routine
SAMPON	Used to monitor CPU usage (CTSS specific - may be deleted)
SAMPLE	Used to monitor CPU usage (CTSS specific - may be deleted)
ERF	Error Function Routine

表4.2 新・旧シンボル名一覧 (1/2)

MELPROG/MOD1/000 (05/31/87)			
CONVERSION COMMENTS.			
1. SYMBOL NAME(S).			
1.1 VARIABLE/ARRAY NAME(S).			
(SEQ)	OLD NAME	=	NEW NAME COMMON/LOCAL
(1)	ALFSTRC	=	ALFSTR
(2)	AMOLTEN	=	AMOLT N
(3)	CORTOTF	=	CORT TF
(4)	DCORIUM	=	DCORI M
(5)	DELTATI	=	DELT TI
(6)	DELZRO2	=	DELZR 2
(7)	DENSAWN	=	D NSAWN
(8)	DIFCOEF	=	DIFCO F
(9)	DMPINTM	=	DMP NTM
(10)	DRHODTM	=	DRH DTM
(11)	DTSLMIN	=	DTSLM N
(12)	FACTR10	=	FACT 10
(13)	FACTR11	=	FACT 11
(14)	FACTR12	=	FACT 12
(15)	FACTR13	=	FACT 13
(16)	FARIGHT	=	F RIGHT
(17)	FCTR10P	=	F TR10P
(18)	FCTR11P	=	F TR11P
(19)	FCTR12P	=	F TR12P
(20)	FCTR13P	=	F TR13P
(21)	FDRIGHT	=	FDR GHT
(22)	GASCOEF	=	GASCO F
(23)	H2TOTAL	=	H2TOT L
(24)	IBPCONA	=	IBPC NA
(25)	IBPCON1	=	IBPC N1
(26)	IBPCON2	=	IBPC N2
(27)	IBPCIN3	=	IBPC N3
(28)	ICCELLBOT	=	IC LLB T
(29)	IFIRST1	=	IF RST1
(30)	IGAMZRO	=	IG MZRO
(31)	IIFAILM	=	IIFAI LM
(32)	IRESTRT	=	IREST T
(33)	ISTPCPT	=	ISTPCP
(34)	ITERMAX	=	ITERM X
(35)	ITERMIN	=	ITERM N
(36)	KSERIES	=	KSERI S
(37)	LAINTAB	=	LAINT B
(38)	LALRHBR	=	L LRHBR
(39)	LALRHBRZ	=	L LRHBZ
(40)	LARIGHT	=	LAR GHT
(41)	LASTFPM	=	L STFPM
(42)	LDACELL	=	LDAC LL
(43)	LDARING	=	LDAR NG
(44)	LDELTAZ	=	LDELT Z
(45)	LDHSLDP	=	LDHSLD
(46)	LDHSVDP	=	LDHSV D
(47)	LDMDCOR	=	LDMDC R
(48)	LDRHOGR	=	LDRH CR
(49)	LDRHODP	=	LDRH DP
(50)	LDRHODT	=	LDRH DT

表 4.2 新・旧シンボル名一覧 (2/2)

C	(51)	LFAROUT	=	LFARO T (C)
C	(52)	LFHDOUT	=	LFHDO T (C)
C	(53)	LGAMHYD	=	LG MHYD (C)
C	(54)	LGAMROD	=	LGAMR D (C)
C	(55)	LHEIGHT	=	LHE GHT (C)
C	(56)	LPRNTAB	=	LPRNT B (C)
C	(57)	LPROTAB	=	LPROT B (C)
C	(58)	LRADIUS	=	LRADI S (C)
C	(59)	LRE1TVM	=	LRE TVM (C)
C	(60)	LRESIDU	=	LRESID (C)
C	(61)	LRHOZRO	=	LRH ZRO (C)
C	(62)	LSCELLM	=	LSC LLM
C	(63)	LSCELLV	=	LSC LLV
C	(64)	LTINTTAB	=	LTINT B (C)
C	(65)	LVINTAB	=	LVINT B (C)
C	(66)	NRBANDS	=	NRB NDS
C	(67)	PSYSTEM	=	PSYST M
C	(68)	QZRFORM	=	QZRF RM
C	(69)	SEDINTM	=	SED NTM (C)
C	(70)	SMOOTH1	=	SMO TH1 (C)
C	(71)	SMOOTH2	=	SMO TH2
C	(72)	SPECRAD	=	SPECR D
C	(73)	TIMEPLT	=	TIM PLT (C)
C	(74)	TINTPLT	=	T NTPLT (C)
C	(75)	TOTMASS	=	TOTM SS
C	(76)	TSEDTM	=	TSED TM (C)
C	(77)	UPCOREF	=	UPCOR F (C)
C	(78)	UPCWER	=	UPCWER R (C)
C	(79)	VFRIGHT	=	VFR GHT
C	(80)	VIKGRAFA	=	V KGRFA
C	(81)	VIKGRAFE	=	V KGRFE
C	(82)	VIKGRAFS	=	V KGRFS
C	1.2 PROGRAM	NAME(S).		←副プログラム名
C	(SEQ)	OLD NAME	=	NEW NAME
C	(1)	BALANCE	=	BALANC
C	(2)	CNTNUTY	=	CNTN TY
C	(3)	COMBINE	=	COMBIN
C	(4)	DCORE3C	=	DCOR 3C
C	(5)	FINTERP	=	FINT RP
C	(6)	ISECOND	=	ISEC ND
C	(7)	LININTX	=	LIN NTX
C	(8)	MCALL2D	=	MC LL2D
C	(9)	MDRIVER	=	MDRIV R
C	(10)	MECHPRP	=	M CHPRP
C	(11)	MOCQEF1	=	MOCO F1
C	(12)	PCALL2D	=	PC LL2D
C	(13)	SAMPTRM	=	SAMPTR
C	(14)	SETPROP	=	SETPR P
C	(15)	TRNSPRT	=	TRNSPR
C	(16)	UO2PROP	=	UO2PR P
C	1.3 COMMON BLOCK	NAME(S).		←共用ブロック名
C	(SEQ)	OLD NAME	=	NEW NAME
C	(1)	CHENCK1	=	CH NCK1
C	(2)	CHENCK2	=	CH NCK2
C	(3)	CONTAIN	=	CONTA N
C	(4)	IMBEDD	=	IMBEDC
C	(5)	MELHOOK	=	MELHO K

副プログラム名と
一致
(SAME SUB PROGRAM)

表4.3 MELPROG DECK名と副プログラム名一覧 (1/4)

2. DECK NAME(S) AND SUB PROGRAM NAME(S)		
(SEQ)	DECK NAME	SUB PROGRAM NAME(S) (ENTRY)
(1)	AVG	AVG
(2)	AXIAL	AXIAL
(3)	BAL2D	BALANCE
(4)	BCSS	BCSSET
(5)	BEDBC	BEDBC
(6)	BEDSUM	BEDSUM
(7)	BOLT	BOLT
(8)	CANDLE	CANDLE
(9)	CDIA2D	CDIAM
(10)	CNGLOD	CNGLOD
(11)	CNTNUTY	CNTNUTY
(12)	COLU	COLUMN
(13)	COMBINE	COMBINE
(14)	CONFUN	CONFUN
(15)	-----	CONGAS
(16)	-----	CONUO2
(17)	CORE	CORE
(18)	COREBC	COREBC
(19)	COREHT	COREHT
(20)	COREIN	COREIN
(21)	CORPNT	CORPNT
(22)	CORPOW	CORPOW
(23)	CORPTT	CORPTT
(24)	CORSET	CORSET
(25)	CREUTK	CREUTK
(26)	CRGEOM	CRGEOM
(27)	CRINIT	CRINIT
(28)	CRMECH	CRMECH
(29)	CRMESH	CRMESH
(30)	CRPROP	CRPROP
(31)	CRZROX	CRZROX
(32)	CYLN	CYLIN
(33)	DBCALC	DBCALC
(34)	DBFORM	DBFORM
(35)	DBINP	DBINP
(36)	DBINTR	DBINTR
(37)	DBPNTT	DBPNTT
(38)	DBPROP	DBPROP
(39)	DBSRC	DBSRC
(40)	DEBID	DEBID
(41)	DEBPNT	DEBPNT
(42)	DEBRIS	DEBRIS
(43)	DOUBLE	DOUBLE
(44)	DSOL	DSOL
(45)	----	DSOLF
(46)	----	DSOLB
(47)	DYNAM	DYNAM
(48)	EGGCRT	EGGCRT
(49)	ELLIPSE	ELLIPS
(50)	ERRHT	ERRHT

表4.3 MELPROG DECK名と副プログラム名一覧 (2/4)

C	(51)	EXPX2D	EXPEX
C	(52)	FACE	FACE
C	(53)	FALFA	FALFA
C	(54)	FINDIC	FINDIC
C	(55)	FINT	FINTERP
C	(56)	FKPM	FKPM
C	(57)	FLDS2D	FLUIDS
C	(58)	FLDYN	FLDYN
C	(59)	FLDY2D	FLDYN
C	(60)	FLIN2D	FLDIN
C	(61)	FRIC3	FRIC3
C	(62)	FUNRD	FUNRD
C	(63)	FWALL	FWALL
C	(64)	F4PROP	F4PROP
C	(65)	GAUSS	GAUSS
C	(66)	GPLATE	GPLATE
C	(67)	GRF	GRF
C	(68)	---	BFALOC
C	(69)	---	BFOUT
C	(70)	---	BFCLOS
C	(71)	HCPH2	HCPH2
C	(72)	HCPST	HCPST
C	(73)	HTC	HTC
C	(74)	HTCOR	HTCOP
C	(75)	-----	HTCORM
C	(76)	-----	CHEN
C	(77)	-----	CHF
C	(78)	-----	CHF1
C	(79)	-----	TMSFBM
C	(80)	-----	DFHT
C	(81)	-----	HVFILM
C	(82)	HTDBRS	HTDBRS
C	(83)	-----	DRYOUT
C	(84)	HTMELT	HTMELT
C	(85)	H2DF2D	H2DIFU
C	(86)	IFOP	IFOPEN
C	(87)	IMBEDD	IMBEDD
C	(88)	IMPX2D	IMPEX
C	(89)	JAC21H	JAC21H
C	(90)	JOIN	JOINT
C	(91)	J1H2D	JAC1H
C	(92)	J112D	JAC11
C	(93)	J212D	JAC21
C	(94)	J312D	JAC31
C	(95)	J41H2D	JAC41H
C	(96)	J412D	JAC41
C	(97)	KNBS2D	KNOBS
C	(98)	LCMRW	RWLCM
C	(99)	LININT	LININTX
C	(100)	LININT2	LININT
C	(101)	LINKS	LINK
C	(102)	LMP	LMP
C	(103)	---	COPY
C	(104)	LOAD	LOADZ
C	(105)	LPACK	LPACK
C	(106)	MATPRP	MATPRP
C	(107)	MCALL2D	MCALL2D
C	(108)	MCF12D	MOCDEF1
C	(109)	MCHP	MECHPRP
C	(100)	MDRIVER	MDRIVER
C	(111)	MECH	MECH
C	(112)	MECHIN	MECHIN
C	(113)	MERGE	MERGE
C	(114)	MEST2D	MESTAB
C	(115)	MOCF2D	MOCDEF
C	(116)	MOPD2D	MOPRED
C	(117)	MOST2D	MOSTAB
C	(118)	MOVC	MOVC
C	(119)	MOVEIT	MOVIEIT
C	(120)	MPRGIN	MPRGIN

表 4.3 MELPROG DECK名と副プログラム名一覧 (3/4)

C	(121)	MPROG	RMPROG	(MPROG , DCORE3C)
C	(122)	MSTRCT	MSTRCT	
C	(123)	MTPT2D	MATPRT	
C	(124)	NEWSUP	NEWSUP	
C	(125)	NUM	NUM	
C	(126)	OUTCOR	OUTCOR	
C	(127)	OUTPUT	OUTPUT	
C	(128)	PCALL2D	PCALL2D	
C	(129)	PERF	PERF	
C	(130)	PHASE	PHASE	
C	(131)	PIT2D	PIT	
C	(132)	PLOT2D	PLOT	
C	(133)	PNTMP2	PNTMP	
C	(134)	POINT2D	POINT	
C	(135)	POOL	POOL	
C	(136)	POWER	POWER	
C	(137)	PRFS	PERFS	
C	(138)	PRNT2D	FLPRNT	
C	(139)	RDIN	RADIN	
C	(140)	RDPNTT	RDPNTT	
C	(141)	RDPT	RADPNT	
C	(142)	READ	READ	
C	(143)	REMESH	REMESH	
C	(144)	RESTRT	RESTRT	
C	(145)	RMIM	RADMIM	
C	(146)	SEARCH	SEARCH	
C	(147)	SHIFT	SHIFT	
C	(148)	SMOOZE	SMOOZE	
C	(149)	SMOVE	SMOVEZ	
C	(150)	SORMAT	SORMAT	
C	(151)	SOUT	OUTSTR	
C	(152)	STBC2D	SETBC	
C	(153)	STGM	STGEOM	
C	(154)	STHT	STRCHT	
C	(155)	STIN	STRCIN	
C	(156)	STIN2	STINIT	
C	(157)	STPR	STRPRP	
C	(158)	STPR2D	SETPROP	
C	(159)	STPT	STRCPT	
C	(160)	STPW	STPOWR	
C	(161)	STRC	STRUCT	
C	(162)	STRESS	STRESS	
C	(163)	STRPTT	STRPTT	
C	(164)	STRSET	STRSET	
C	(165)	STSE	STSET	
C	(166)	THERMO	THERMO	
C	(167)	-----	CPVV	
C	(168)	-----	HLS	
C	(169)	-----	HVS	
C	(170)	-----	THCL	
C	(171)	-----	ERROR	
C	(172)	-----	CPLL	
C	(173)	-----	THCV	
C	(174)	-----	SIGMA	
C	(175)	-----	VISCL	
C	(176)	-----	VISCV	
C	(177)	-----	POLY	
C	(178)	-----	FPROP	
C	(179)	-----	CPVV1	
C	(180)	-----	RHOLIQ	
C	(181)	-----	MZIRC	
C	(182)	TOVIKI	TOVIKI	
C	(183)	TRANS	TRANZ	
C	(184)	TRIS	TRYSLV	
C	(185)	TRNSPRT	TRNSPRT	
C	(186)	TSTP2D	TSTEP	
C	(187)	UO2P	UO2PROP	
C	(188)	---	CP	
C	(189)	---	FTHCON	
C	(190)	---	SPHT	

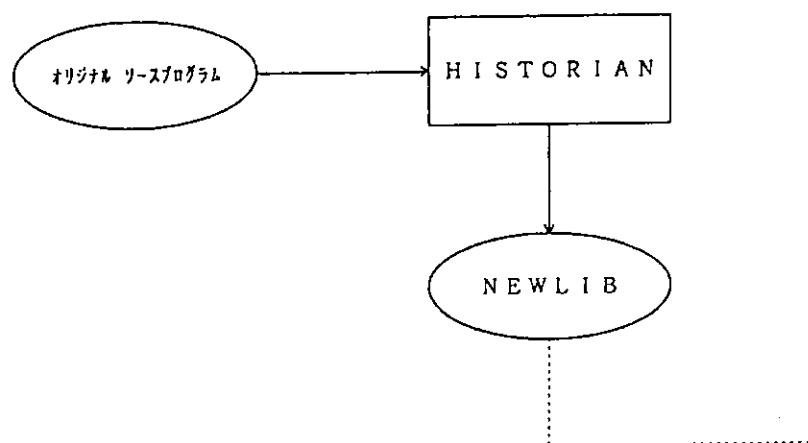
表 4.3 MELPROG DECK名と副プログラム名一覧 (4/4)

C	(191)	----	RHO
C	(192)	----	FLOW
C	(193)	UPFAIL	UPFAIL
C	(194)	VFAC	VIEWFC
C	(195)	VSGESL	VSGESL
C	(196)	WFRIC1	WFRIC1
C	(197)	ZEROD	ZEROD
C	(198)	ZRHT2D	ZRHEAT
C	(199)	ZROXID	ZROXID

表 4.4 作成した副プログラム名一覧

3. ADDED LIBRARY NAME(S). (UNSATISFIED ENTRY NAME(S) AND GENARATED FOR CONVERSION.)			SUB PROGRAM NAME(S) (ENTRY)
SEQ	DECK NAME		
(1)	BALAN	(U)	BALAN
(2)	BALBAK	(U)	BALBAK
(3)	DUMMYRTN	(U)	SAMPLE
(4)	-----	(U)	SAMPON
(5)	-----	(U)	SYCALL
(6)	-----	(U)	SAMPTR
(7)	-----	(U)	BGLSDC
(8)	-----	(U)	BGLSSL
(9)	ECHO	(G)	ECHO
(10)	GETJCL	(S)	GETJCL
		U	
(11)	HQR	(U)	HQR
(12)	HQR2	(U)	HQR2
(13)	ISECOND	(U)	ISEC ND
(14)	MABEND	(G)	MABEND
(15)	MOVLEV	(U)	MOVLEV (MOVLEN)
(16)	MTRACE	(G)	MTRACE (MTRON , MTROUT)
(17)	ORTHES	(U)	ORTHES
(18)	ORTTRAN	(U)	ORTTRAN
(19)	R1MACH	(U)	R1MACH
(20)	SCOPY	(U)	SCOPY
(21)	SCOPYM	(U)	SCOPYM
(22)	SECOND	(U)	SECOND
(23)	SGEDI	(U)	SGEDI
(24)	SGEEV	(U)	SGEEV
(25)	SGEFA	(U)	SGEFA
(26)	SGESL	(U)	SGESL
(27)	SGTSL	(U)	SGTSL
(28)	MDRIVER	(G)	BLKDA
(29)	-----	(G)	BLKDB
(30)	-----	(G)	SPLINI (SPLLPT,SPLDPT,SPLERR)
(31)	-----	(G)	SPLFLD
(32)	-----	(G)	SPLCOR
(33)	-----	(G)	SPLSTR
(34)	-----	(G)	SPLDBL
(35)	-----	(G)	SPLDBD
(36)	-----	(G)	SPLDBS
(37)	-----	(G)	SPLDBP
(38)	-----	(G)	SPLCLR

CREATION RUN



UPDATE RUN

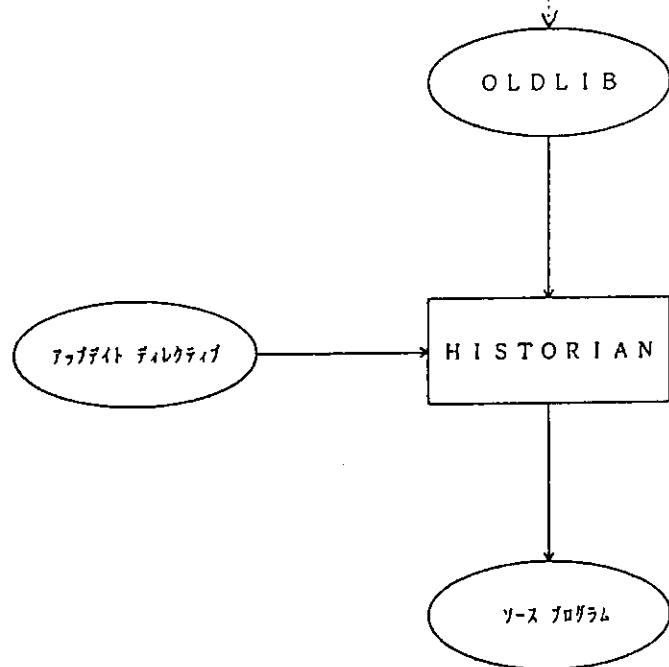


図 2.1 HISTORIANによるソースプログラムの修正・管理の概要

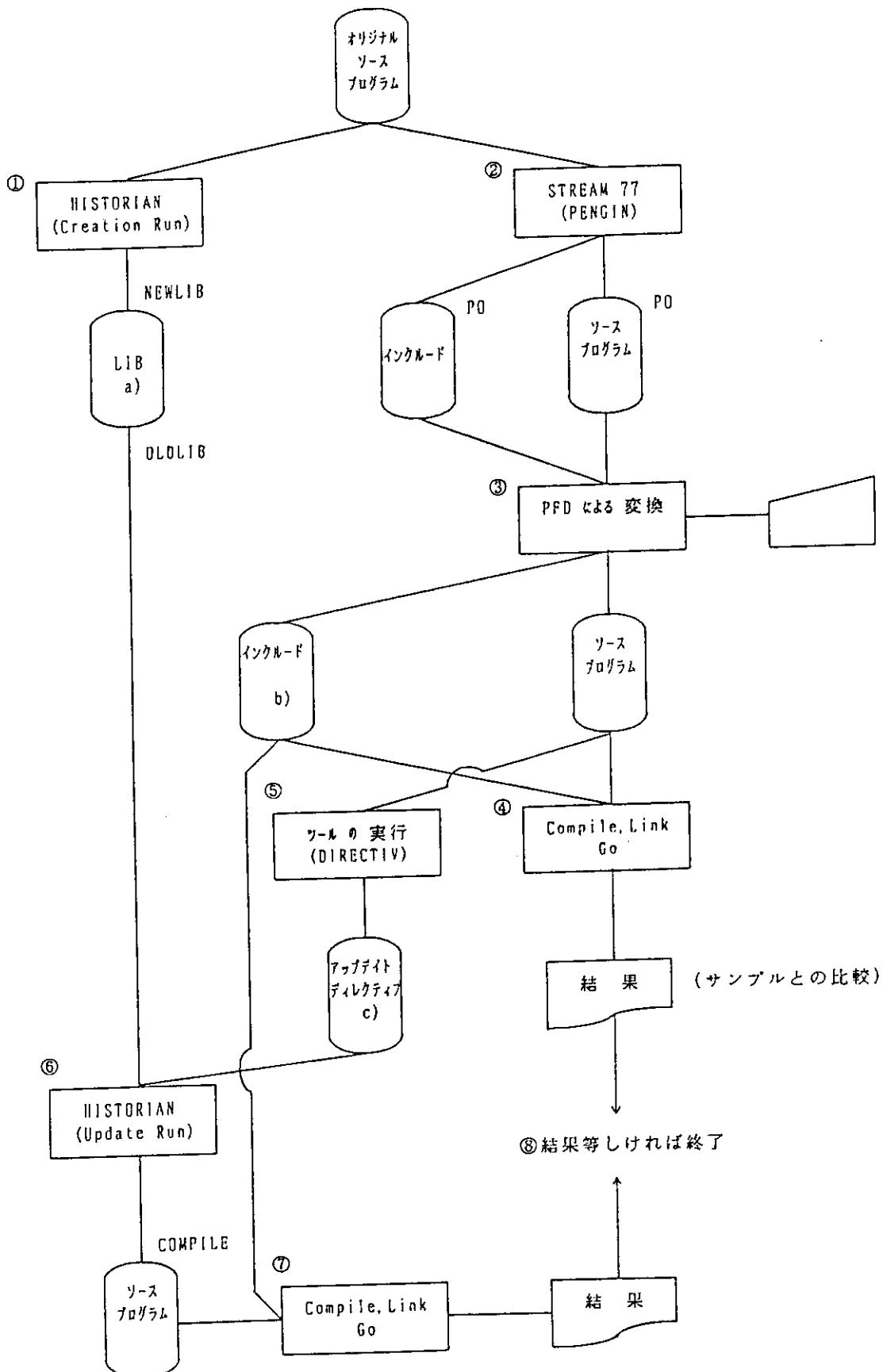


図 3.1 變換・整備作業手順とソースプログラムの移り変わり

オリジナルソースプログラム

C		S U B A 10
	A B C D E F G = M A X (1, 0, A)	S U B A 11
	A B C D E F G = B * A B C D E F G	S U B A 12
C		S U B A 13



変換作業中デバッグソースプログラム

C		S U B A 10
* CONVSTR * I SUBA,10		
CALL MTRACE ('S U B A')		
* CONVEND		
CDEL A B C D E F G = M A X (1, 0, A)		S U B A 11
CDEL A B C D E F G = B * A B C D E F G		S U B A 12
* CONVSTR * D SUBA,11,12		
A B C D F G = M A X (1, 0, A)		
A B C D F G = B * A B C D F G		
* CONVEND		
C		S U B A 13



HISTORIAN アップデート・ディレクティブ

* I SUBA,10		
CALL MTRACE ('S U B A')		
* D SUBA,11,12		
A B C D F G = M A X (1, 0, A)		
A B C D F G = B * A B C D F G		

図 4.1 變換作業によるソースプログラムの移り変わり

```

*OK MPROG
  SUBROUTINE RMPROG
*INCLUDE MELHOOK ← *CALL MELHOOK & Include 3行
C
*INCLUDE CORPRM
C
*INCLUDE CORCOM
C
*INCLUDE SPACE
*INCLUDE PCOM
*INCLUDE MS
*INCLUDE PARAM
*INCLUDE FLDPRM
*INCLUDE IO
*INCLUDE PNTRAD
*INCLUDE PTST
*INCLUDE POINTD
*INCLUDE CONPNT
*INCLUDE CONTAIN
COMMON/GRFLNK/IGRFHD
  INTEGER DMPFLG
  REAL LTDUMP
  INTEGER ZBETA(6)
  DATA ILEFTO /0/
C  DEFINE BASIC PROPERTIES
C
C  SET FLAG TO ALWAYS GET HEADER ON GRAPH OUTPUT FILE
C
CDEL DATA IGRFHDL/1/ ← コモン変数への初期値
*CONVSTR *D MPROG.29 ← デルクテイフ作成の指示行(開始)
*INCLUDE *SPLCM
C
C  TITIME.....USED TIME FOR THREE TIME STEPS CALCULATION.
C  TSTIME.....FIRST TIME STEP CALCULATION BEGIN TIME.
C  TETIME.....FIRST TIME STEP CALCULATION END TIME.
C  IFIRST....SET FLAG FOR TITIME.
C  LBLPUT....FLAG FOR SPL PLOT TO PUT LABEL.
C
C  REAL TITIME,TSTIME,ETIME
C  DATA IFIRST / 0 /
C  DATA LBLPUT / 0 /
C  CALL MTRACE ('RMPROG ') ← 副プログラムトレース用の変更
*CONVEND ← デルクテイフ作成の指示行(終了)
C
C LENGTH OF THE SCM DATA BLOCK
C
C  LENSCM = 60000
C
C SET I/O DEVICES
C
C  NTTY=91
C  NOUT=96
C  NIN=95
C  NDUMP=92
C  NGUESS=93
C  NMCH=97
C  NSAVE=98
C  NCOUT=54
C  NPLT=82
C
C SET INTERNAL IO UNITS
C
C  IOFLDS=1
C  IOPINS=2
C  IOSTRC=3
C  IORAD=4
C  IODBED=5
C  ISTEP=0
C  ZBETA(3)=0
C  CALL GETJTL(TIMEI)
*CONVSTR *I MPROG.57 ← HISTORIANとデルクテイフをあらかじめ
C  ECHO BACK FOR INPUT CARD DATA. MPROG.57a に付随する注釈の追加をあらかじめ
C  CALL ECHO ( 5 ,NIN,NOUT) } 付与する
C  CALL ECHO (NIN,NIN,NOUT)
C  CALL ECHO (NIN,NIN, 6 )
*CONVEND } エラーのエコーポイント

  MPRO 1
  MPRO 2
  MPRO 3
  MPRO 4
  MPRO 5
  MPRO 6
  MPRO 7
  MPRO 8
  MPRO 9
  MPRO 10
  MPRO 11
  MPRO 12
  MPRO 13
  MPRO 14
  MPRO 15
  MPRO 16
  MPRO 17
  MPRO 18
  MPRO 19
  MPRO 20
  MPRO 21
  MPRO 22
  MPRO 23
  MPRO 24
  MPRO 25
  MPRO 26
  MPRO 27
  MPRO 28
  MPRO 29
  MPRO 30
  MPRO 31
  MPRO 32
  MPRO 33
  MPRO 34
  MPRO 35
  MPRO 36
  MPRO 37
  MPRO 38
  MPRO 39
  MPRO 40
  MPRO 41
  MPRO 42
  MPRO 43
  MPRO 44
  MPRO 45
  MPRO 46
  MPRO 47
  MPRO 48
  MPRO 49
  MPRO 50
  MPRO 51
  MPRO 52
  MPRO 53
  MPRO 54
  MPRO 55
  MPRO 56
  MPRO 57

```

図 4.2 副プログラム RMPROG の変換ソースプログラム (1/4)

```

C          MPRO  58
C READ AND INITIALIZE ALL INPUT DATA      MPRO  59
C          MPRO  60
C          CALL MPRGIN      MPRO  61
C          MPRO  62
C DUMP OFF THE INITIAL CONDITIONS      MPRO  63
C          MPRO  64
C          IDUMMY=INEW      MPRO  65
C          INEW=IOLD      MPRO  66
C          CALL GRF      MPRO  67
C          INEW=IDUMMY      MPRO  68
C          CALL RESTRT(1)      MPRO  69
C          MPRO  70
C          IMSEC = 4      MPRO  71
C          CALL SAMPLE('TFILE',IMSEC)      MPRO  72
C          CALL SAMPON      MPRO  73
C          MPRO  74
C          RETURN      MPRO  75
C          MPRO  76
C BEGIN TIME SOLUTION      MPRO  77
C          MPRO  78
C          ENTRY MPROG      MPRO  79
*CONVSTR *I MPROG.79
    CALL MTRACE ('MPROG' )
    IF (IFIRST.EQ.0) CALL SECOND (TSTIME ) } 1 time step (T)
    I*DEBF = 0 } a CPU time を計算する
} タイムステップの変更
*CONVEND
C          IPRNT=9000000      MPRO  80
C          MPRO  81
C          CHECK IF TIME TO STOP      MPRO  82
C          MPRO  83
C          MPRO  84
C          IF(MLINK.NE.1) THEN      MPRO  85
C          IF(TIME+DTN.GE.ENDTIM) THEN      MPRO  86
C          DTN=ENDTIM-TIME      MPRO  87
C          IPRNT=ISTEP+1      MPRO  88
C          ENDIF      MPRO  89
C          ENDIF      MPRO  90
C TAKE A FLUIDS STEP      MPRO  91
C          CALL FLDYN      MPRO  92
C          MPRO  93
C BRANCH TO FUEL ROD MODULE      MPRO  94
C          MPRO  95
C          CDEL ENTRY DCORE3C      MPRO  96
*CONVSTR *D MPROG.97
    ENTRY DCOR 3C } ハストニアデータベースを読み込み、MPROG.97
} を削除し、以下のランクを追加することを示す
} ワクタシンボル名の
} メタシンボル名化
*CONVEND
C          IF(MODLSW.EQ.1) CALL CORE      MPRO  98
C          MPRO  99
C PERFORM THE STRUCTURES HEAT TRANSFER      MPRO 100
C          MPRO 101
C          CALL STRUCT      MPRO 102
C          MPRO 103

```

図4.2に手書きの説明が記載されています。主な注釈は以下の通りです。

- 手書き記入: 1 time step (T) a CPU time を計算する タイムステップの変更
- 手書き記入: ハストニアデータベースを読み込み、MPROG.97 を削除し、以下のランクを追加することを示す ワクタシンボル名のメタシンボル名化
- 手書き記入: ワクタシンボル名のメタシンボル名化

図4.2 副プログラム RMPROGの変換ソースプログラム (2/4)

```

C MPRO 104
C EXECUTE DEBRIS MODULE MPRO 105
C MPRO 106
C CDEL IF(LDBRIS.NE.1) CALL DEBRIS MPRO 107
*CONVSTR *D MPROG.107
C?? IF(LDBRIS.NE.1) CALL DEBRIS
IF(LDBRIS.NE.0) CALL DEBRIS
} オリジナルのプログラム
} ドアと窓があるための修正
*CONVENT
C MPRO 108
C PERFORM RADIATION HEAT TRANSFER CALCULATION MPRO 109
C MPRO 110
C IF(LRAD.EQ.1) CALL RADMIM MPRO 111
C MPRO 112
CDEL IF(MLINK.EQ.0) TSEDITM=(IFIX(TIME/SEDINTM+1.0E-4) + 1.0)*SEDINTM MPRO 113
*CONVSTR *D MPROG.113
IF(MLINK.EQ.0) TSED TM=(IFIX(TIME/SED NTM+1.0E-4) + 1.0)*SED NTM
} フラッシュディスクの書き換え
*CONVENT
C DUMP OFF THE SOLUTION MPRO 114
C MPRO 115
C MPRO 116
*CONVSTR *I MPROG.116
IF ( TIME .GE. TGRAFM )
THEN
IF ( IYDEBF .EQ. 0 )
THEN
IF ( IYDEBL .EQ. 0 ) CALL SPLDBL
CALL SPLDBD
CALL SPLDBP
ENDIF
IYDEBF = 0
IF ( LBLPUT .EQ. 0 )
THEN
LBLPUT = 1
CALL SPLLPT ( DUM,DUM )
ENDIF
CALL SPLDPT ( DUM,TIME )
ENDIF
*CONVENT
IF(TIME.LE.(TGRAFM-1.E-8))GO TO 10 MPRO 117
CALL GRF MPRO 118
IF(MLINK.EQ.0) TGRAFM=(IFIX(TIME/GFINTM+1.0E-4) + 1.0)*GFINTM MPRO 119
10 CONTINUE MPRO 120
MPRO 121
C PRODUCE MIMDMP DUMP AT SAME TIME AS TRCDMP MPRO 122
IF(MLINK.EQ.0.AND.TIME.GE.ENDTIM) TDUMPM = TIME MPRO 123
IF(TIME.LE.(TDUMPM-1.0E-8))GO TO 20 MPRO 124
CALL RESTRT(1) MPRO 125
CALL RESTRT(1)
CDEL IF(MLINK.EQ.0) TDUMPM=(IFIX(TIME/DMPINTM+1.0E-4) + 1.0)*DMPINTM MPRO 126
*CONVSTR *D MPROG.126
IF(MLINK.EQ.0) TDUMPM=(IFIX(TIME/DMP NTM+1.0E-4) + 1.0)*DMP NTM
} 副利点をカットする
} 過剰なロケン
*CONVENT
TIME1=60.0*( IFIX(TIME/60.0 +1.0E-4) ) MPRO 127
CDEL IF(MLINK.EQ.0 .AND. MOD(DMPINTM,60.0).EQ.0.0) TDUMPM=TIME1+DMPINTMM MPRO 128
*CONVSTR *D MPROG.128
IF(MLINK.EQ.0 .AND. MOD(DMP NTM,60.0).EQ.0.0) TDUMPM=TIME1+DMP NTM
*CONVENT
20 CONTINUE MPRO 129

```

図 4.2 副プログラム RMPROG の変換ソースプログラム (3/4)

```

C
CDEL CALL SYSCALL(4L1023,ZBETA)
CDEL ILEFT=ZBETA(3)
CDEL IDELT=ILEFT0-ILEFT
CDEL ILEFT0=ILEFT
CDEL IF(ILEFT.GT.45000000) GO TO 50
CDEL IF(ILEFT.GE.2*IDELT) GO TO 50
CDEL CALL SYSCALL(4L1029,ZBETA)
CDEL IF(ZBETA(5).EQ.0) GO TO 50
*CONVSTR *D MPROG.131,138
  IF ( IFIRST .EQ. 0 )
    .
    . THEN
      IFIRST = 1
      CALL SECOND ( TETIME )
      TETIME = (ETIME - TSTIME)*3.0
      IF ( TETIME .LT. 1.0E-2) TETIME = 0.1E0
      WRITE (6,6010) TIMEI,TETIME,TSTIME,ETIME
6010  FORMAT(/
      ' ===== MELPROG TIME DATA ====='
      ' (1) AVAILABLE CPU TIME.          (SEC) : ',F20.5/
      ' (2) CHECKED LEFT CPU TIME.     (SEC) : ',F20.5/
      ' (3) FIRST TIME STEP BEGIN TIME. (SEC) : ',F20.5/
      ' (4) FIRST TIME STEP END TIME.   (SEC) : ',F20.5/
      ' (2) >= ( (4)-(3) ) * 3'//)
    .
    ENDIF
C   CALL SYSCALL(4H1023,ZBETA)
C   ILEFT=ZBETA(3)
C   IDELT=ILEFT0-ILEFT
C   ILEFT0=ILEFT
C   IF(ILEFT.GT.45000000) GO TO 50
C   IF(ILEFT.GE.2*IDELT) GO TO 50
C   CALL SYSCALL(4H1029,ZBETA)
C   IF(ZBETA(5).EQ.0) GO TO 50
*CONVENT
  CALL SECOND(TIMEU)
  TIMEI=TIMEI-TIMEU
  CDEL IF(TIMEL.GT.300) GO TO 50
*CONVSTR *D MPROG.141
  IF(TIMEL.GT.TETIME) GO TO 50
*CONVENT
  ENDIM=TIME+0.9*DTN
  50 CONTINUE
C
C
C SWAP SOME STORAGE
C
  IOLD=INEW
  RETURN
END

```

MPRO 130
MPRO 131
MPRO 132
MPRO 133
MPRO 134
MPRO 135
MPRO 136
MPRO 137
MPRO 138

HISTORIANのデータをあらかじめ
MPROG.131から138を削除し以下の
のラインを追加する

} FALOMバージョンの利用
} 制限を考慮して整備
} (CPU time 計算値と
} time step が同じCPU時間)

図 4.2 副プログラム RMPROG の変換ソースプログラム (4/4)

```

*IDENT J8706SVO ← 変換をあらわす IDE を示す。
*I AVG.3
  CALL MTRACE ( 'AVG' )
*I AXIAL.11
  CALL MTRACE ( 'AXIAL' )
*D BAL2D.2
  SUBROUTINE BALANC (MF, MF1, MZ, MR)
*I BAL2D.10
  CALL MTRACE ( 'BALANCE' )
*I BCSS.20.
  CALL MTRACE ( 'BCSSET' )
*I BEDBC.9
  CALL MTRACE ( 'BEDBC' )
*I BEDSUM.12
  CALL MTRACE ( 'BEDSUM' )
*D CANDLE.40
  DATA FRIC/5.E-3/,IGAIN/5/,ITMAX/20/
*I CANDLE.48
  CALL MTRACE ( 'CANDLE' )
*D CANDLE.90
  IC LLB T=ICELLM
*D CANDLE.94
  ICOPPM(1)=LCOPPM+5*NCELL+IC LLB T
*D CANDLE.104,107
  ICOPPM(1)=LCOPPM+4*NCELL+IC LLB T
  ICOPPM(2)=LCOPPM+2*NCELL+IC LLB T
  ICOPPM(3)=LCOPPM+2*NCELL+IC LLB T
  ICOPPM(4)=LCOPPM+3*NCELL+IC LLB T
*I CDIA2D.5
  CALL MTRACE ( 'CDIAM' )
*I CNGLOD.10
  CALL MTRACE ( 'CNGLOD' )
*D CNTNUTY.2
  SUBROUTINE CNTN TY
*I CNTNUTY.7
  CALL MTRACE ( 'CNTNUTY' )
  :
  :

```

図4.3 HISTORIANディレクティブの一部

```

//JCLG JOB
//JCLG EXEC JCLG
//SYSIN DD DATA,DLM='++'
// JUSER 1234XXXX,AA.8BBBBBBB,1234.56
T.O W.4 I.O C.O OPN
OPTP NOTIFY=JXXXX,PASSWORD=?????????
/*JOBPARM K=0
//**COMP EXEC PGM=JRQCPRT,PARM='TYPE2'    PRINTER IMAGE
//COMPML PROC MELOUT=,SYSOT2='*'
//COMP EXEC PGM=JRQCPRT,PARM='TYPE3'    SOURCE IMAGE
//UTYIN   DD DSN=&MELOUT,DISP=SHR
//UTYNLP  DD SYSOUT=&SYSOT2
//UTYLIST DD SYSOUT=&SYSOT2
//      PEND
//      EXEC COMPML,
//      MELOUT=J3076.aaF2.DATA
//      EXEC COMPML,
//      MELOUT=J3076.aaF3.DATA
//*
//      EXEC COMPML,
//      MELOUT=J3076.aaLP1.DATA
//*
//      MELOUT=J3076.aaLP2.DATA
//*
//      MELOUT=J3076.aaLP3.DATA
//*
//      MELOUT=J3076.aaLP4.DATA
+++
//
```

} COMDECK, DECK の -E₂
} リスト用ファイル
} ハードリスト用ファイル

図4.4 COMPSOCバッチ用 JCL の例

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.BBBBBBBB,1234.56          00040000
   T.3  I.4 W.2 C.2 OPN                         00050000
   OPTP NOTIFY=JXXXX,PASSWORD=????????        00060000
//FORT1 EXEC FORT77,SO=J3076.TK.MELPROG,      00070000
// A='ELM(*),AUTODBL(DBLPAD),OPT(3),FLAG(E),NONUM'
//SYSPRINT DD DUMMY                           00080000
//SYSINC DD DSN=J3076.TK.MELPROG.INCLUDE,DISP=SHR 00090000
///*    LINKAGE                                00100000
//LKED EXEC LKED,LM=J3076.TK.MELPROG,A='LET,MAP', 00110000
// PRVLIB='J3076.TK.MELPLIB',                  00120000
//   GGS='J3080.*$PLEDIT'                     00130000
//SYSLMOD DD DSN=J3076.TK.MELPROG.LOAD,DISP=SHR,UNIT= 00140000
//SYSLIN  DD DSN=J3076.TK.MELPUPD.OBJ,DISP=SHR < サンプルコードを変更
//          DD DSN=&OBJ,DISP=(OLD,DELETE)          サブルーチンのアドレス
//          DD DSN=J3076.&OBJ,DISP=SHR           00150000
//SYSIN  DD *                                 00160000
  ENTRY  MDRIVR                            00170000
  NAME   MELPROG(R)                         00180000
/*
++
*/

```

図 4.5 変換・整備用ソースからのコンパイル、リンク JCL

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.BBBBBBBB,1234.56          00040000
   T.1  I.3 W.0 C.2 OPN                         00050000
   OPTP NOTIFY=JXXXX,PASSWORD=????????        00060000
//FORT1 EXEC FORT77,SO=J3076.TK.MELPROG,      00070000
// A='ELM(AAAAAAA),AUTODBL(DBLPAD),OPT(3),FLAG(E),NONUM'
//SYSPRINT DD DUMMY                           00080000
//SYSINC DD DSN=J3076.TK.MELPROG.INCLUDE,DISP=SHR 00090000
//LKED EXEC LKED,LM=J3076.TK.MELPROG,A='LET,MAP', 00100000
// PRVLIB='J3076.TK.MELPLIB',                  00110000
//   GGS='J3080.*$PLEDIT'                     00120000
//SYSLIN  DD DSN=J3076.TK.MELPUPD.OBJ,DISP=SHR 00130000
//          DD DSN=&OBJ,DISP=(OLD,DELETE)          00140000
//OLDLM  DD DSN=J3076.TK.MELPROG.LOAD,DISP=SHR 00150000
//SYSIN  DD *
  INCLUDE OLDLM(MELPROG)
  ENTRY  MDRIVR                            00160000
  NAME   MELPROG(R)                         00170000
/*
++
*/

```

図 4.6 変換・整備用ソースからの再リンク JCL

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.BBBBBBBB,1234.56          00040000
  T.3 I.4 W.2 C.2 OPN                         00050000
    OPTP NOTIFY=JXXXXX,PASSWORD=??
//FORT1 EXEC FORT77,SD=J3076.TK.@@COMPILE,      HISTORIANに記録された
// A='AUTODBL(DSLPAD),OPT(3),FLAG(E),NONUM'   COMPILE ソースプログラム
//SYSPRINT DD DUMMY                           00070000
//SYSINC DD DSN=J3076.TK.MELPNEW.INCLUDE,DISP=SHR ← 新バージョンの追加
//SYSINC DD DSN=J3076.TK.MELPNEW.INCLUDE,DISP=SHR ← インクルード・ディレクト
//*   LINKAGE                                     00100000
//LKED  EXEC LKED,LM=J3076.TK.MELPROG,A='LET,MAP',
// PRVLIB='J3076.TK.MELPLIB',                   00110000
//   GGS='J3080.%SPLEDIT'                      00120000
//SYSMOD DD DSN=J3076.TK.MELPROG.LOAD,DISP=SHR,UNIT=
//SYSIN  DD DSN=&&08J,DISP=(OLD,DELETE)           00150000
//SYSIN  DD DSN=J3076.02.OBJ,DISP=SHR           00160000
//SYSIN  DD *
  ENTRY  MDRIVR                               00170000
  NAME   MELPNEW(R)                           00180000
/*
*/
//*
//+
//
```

図 4.7 新バージョンソースからのコンパイル、リンク JCL

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.BBBBBBBB,1234.56          00040000
  T.1 I.3 W.0 C.2 OPN                         00050000
    OPTP NOTIFY=JXXXXX,PASSWORD=??
//FORT1 EXEC FORT77,SD=J3076.TK.@@PO,
// A='ELM(AAAAAAA),AUTODBL(DSLPAD),OPT(3),FLAG(E),NONUM'
//SYSPRINT DD DUMMY                           00070000
//SYSINC DD DSN=J3076.TK.MELPNEW.INCLUDE,DISP=SHR
//LKED  EXEC LKED,LM=J3076.TK.MELPROG,A='LET,MAP',
// PRVLIB='J3076.TK.MELPLIB',                   00080000
//   GGS='J3080.%SPLEDIT'                      00090000
//OLDLM  DD DSN=J3076.TK.MELPROG.LOAD,DISP=SHR
//SYSIN  DD *
  INCLUDE OLDLM(MELPNEW)
  ENTRY  MDRIVR                               00100000
  NAME   MELPNEW(R)                           00110000
/*
*/
//*
//+
//
```

図 4.8 新バージョンソースからの再リンク JCL

PAGE
1

SEQ. *...*...1.....*...2.....*...3.....*...4.....*...5.....*...6.....*...7.....*...8

```

1 MELPROG ANALYSIS RUN01 FOR JAERI
2
3 MELPROG MOD 1 (2D FLUIDS + CORE)
4
5 MODIFIED INLET VELOCITIES 2/87
6 INCREASED FUEL INITIAL TEMPERATURE 2/87
7
8
9 MELPROG INPUT DATA IN FREE FORM
10
11 BEGIN GENERAL INPUT SECTION
12
13
14 RESTART FILE (1=YES,0=NO)
15 0
16
17 DUMPING TIME INTERVAL
18 100.0
19 GRAPHICS INTERVAL
20 5.0
21 6.0
22 EDIT INTERVAL
23 100.0
24 SHORT EDIT INTERVAL
25 10.
26 MAIN PRINT FLAG
27 500
28 PRINT FLAGS (5)
29 1 2 1 1 1
30
31 NUMBER OF VESSEL MESH CELLS
32 7
33 NUMBER OF RADIAL RINGS
34 1
35 FLUIDS CELL NUMBER FOR CORE BOTTOM CELL
36 1
37 FLUIDS CELL NUMBER FOR CORE TOP CELL
38 7
39 BOTTOM CELL NBOTH
40 1
41 RADIAL EXTENT OF CORE
42 1
43 RADIATION SWITCH (0/1=OFF/ON)
44 1
45 RADIATION COEFFICIENT
46 0.15
47 START TIME IN SEC
48 1500.
49 PROBLEM END TIME
50 4000.

```

SEQ. PAGE
1

SEQ. *...*...1.....*...2.....*...3.....*...4.....*...5.....*...6.....*...7.....*...8

図4.9 入力データの構造

```

===== SPL PLOT PRINT FLAGS =====
(1) INPUT FLAG = 0
(2) SPL LABELS (I$OPT1) = 0
(3) SPL DUMP TIMES (I$OPT2) = 0
(4) SPL LOCATIONS (I$OPT4) = 0

===== MELPROG TIME DATA =====
(1) AVAILABLE CPU TIME. (SEC) : 1200.00000 ← 計算可能CPU
(2) CHECKED LEFT CPU TIME. (SEC) : 0.68720 ← 計算済みCPU
(3) FIRST TIME STEP BEGIN TIME. (SEC) : 0.33037 ← エラーハンドル
(4) FIRST TIME STEP END TIME. (SEC) : 0.39277
(2) >= ((4)-(3))*3+0.5 (SEC)

```

図 4.10 図形処理情報のプリント出力制御情報とCPUタイム制御情報

```

=====
= S P L P L O T I N F O R M A T I O N S =
=====

( 1) NO. OF WORD(S) = 3984 (13) NZMAX = 30
( 2) NO. OF VARB(S) = 124 (14) NBED = 1
( 3) MF = 4 (15) N$AXRG = 7
( 4) MZ = 9 (16) N$DTMX = 22
( 5) MR = 3 (17) N$DRMX = 20
( 6) NO. OF FIELD = 4 (18) N$DMMX = 18
( 7) NZP1 = 8 (19) I$LOC2 = 154
( 8) NRP1 = 2 (20) I$LOC5 = 14
( 9) KMAX = 7 (21) I$LOC7 = 140
(10) NRING = 1 (22) I$LOC8 = 126
(11) MXCSTR = 25 (23) I$LOS1 = 70
(12) NZZ = 7 (24) I$LOS2 = 287
(25) I$LOS3 = 7

```

図 4.11 図形処理のサイズ情報（各変数の次元の大きさを示す）

```

=====
= M O V L E V I N F O R M A T I O N S .
=
= NO. OF 8 BYTES WORDS IS 1062075343.
=
= NO. OF CALLS IS 344309.
=

```

図 4.12 SCM, LCM転送副プログラムの呼び出し回数と転送語数

PAGE
3

HEPROG ANALYSIS RUN01 FOR JAERI

HEPROG-PWR/MOD1 MAJOR EDIT

TIME = 1600.1672 SEC, STEP NUMBER 264, DELTA TIME = 0.40000+00 SEC

-36-

FIELD 1 SOLUTION IN RING 1 AT TIME 1600.17

AXIAL NODE	TEMPERATURE K	VOLUME FRACTION	AXIAL VELOCITY M/S	RADIAL VELOCITY M/S	ALPHA*RHO KG/M**3	ALPHA*RHO*U J/M**3	DENSITY KG/M**3	PRESSURE PA	PH2/PTOT
1	530.653	0.9999274	3.08423D-02	0.0	0.3608	9.72788D+05	8.733514	2.00005D+06	0.0
2	568.125	0.9999274	3.38348D-02	0.0	0.3290	9.11316D+05	7.965251	2.00004D+06	0.0
3	599.660	0.9999274	3.62809D-02	0.0	0.3072	8.69217D+05	7.439271	2.00003D+06	0.0
4	613.374	0.9999274	3.73651D-02	0.0	0.2988	8.52955D+05	7.236165	2.00003D+06	0.0
5	621.352	0.9999274	3.80241D-02	0.0	0.2941	8.43979D+05	7.123981	2.00002D+06	0.0
6	616.288	0.9999274	3.77212D-02	0.0	0.2971	8.49626D+05	7.194814	2.00001D+06	0.0
7	581.671	0.9999274	3.24267D-02	0.0	0.3191	8.92297D+05	7.728004	2.00000D+06	0.0

図 4.13 第1回メジャーワーティックトの一部(1/4)

TOTAL HYDROGEN PRODUCED BY CORE = 0.0 KG
 TOTAL MASS TRANSFERRED FROM FUEL RODS = 0.0 KG
 TOTAL MASS TRANSFERRED FROM CONTROL RODS = 0.0 KG

***** CORE MODULE OUTPUT AT 1600.17 S *****

TEMPERATURE DISTRIBUTION FOR FUEL ROD NUMBER 1 IN RING 1

CELL	558.8	558.8	558.4	557.9	557.5	557.1	556.9	555.3	555.2	555.2	555.2
2	610.7	610.7	610.3	609.9	609.5	609.1	608.9	605.6	605.5	605.4	605.4
3	629.0	629.0	628.7	628.4	628.1	627.8	627.6	625.2	625.2	625.1	625.1
4	636.2	636.2	635.9	635.7	635.4	635.2	635.0	633.0	633.0	632.9	632.9
5	640.8	640.8	640.6	640.3	640.1	639.9	639.8	637.9	637.9	637.9	637.8
6	636.2	636.2	636.0	635.7	635.5	635.2	635.1	633.2	633.1	633.1	633.1
7	583.5	583.5	583.4	583.3	583.2	583.1	583.0	582.7	582.7	582.7	582.7

DISTRIBUTION OF HYDROGEN PRODUCED DURING THIS TIME STEP (G)

AXIAL NODE	1	RADIAL RING
1	0.0	
2	0.0	
3	0.0	
4	0.0	
5	0.0	
6	0.0	
7	0.0	

DETAILED CORE MODULE OUTPUT FOR RING 1

DATA FOR FUEL ROD NUMBER 1 THERE ARE 9.00 OF THESE IN THIS RING
 THE POWER FACTOR IS 1.00000+00 THE INTERNAL PRESSURE IS 0.0 PA

DATA FOR AXIAL REGION 1 OF FUEL ROD NUMBER 1													
FLUIDS CELL	ZBOT (M)	RADIAL REGION	REGION TYPE	MATERIAL TYPE	NODES	GAP	RINNER (MM)	ROUTER (MM)	TINNER (K)	TOUTER (K)	MASS (G)	WEIGHT % DISTRIBUTION IN LIQUID STEEL U OR AIC ZR 0	
1 1	0.0 0.0	1 2	SOLID SOLID	ZRO2 ZRY4	5 3	YES NO	0.0 4.205	4.135 4.815	558.8 555.3	556.9 555.2	29.416 11.415	0.0 0.0	
DATA FOR AXIAL REGION 2 OF FUEL ROD NUMBER 1										MASS WEIGHT % DISTRIBUTION IN LIQUID STEEL U OR AIC ZR 0			
2 3 4 5 6 7	0.102 0.203 0.305 0.406 0.508 0.610	1 2 1 2 1 2	SOLID SOLID SOLID SOLID SOLID SOLID	ZRY4 ZRY4 ZRY4 ZRY4 ZRY4 ZRY4	5 5 5 5 5 5	YES YES YES YES YES YES	0.0 4.205 4.205 4.205 4.205 4.205	4.135 4.135 4.135 4.135 4.135 4.135	608.9 605.6 605.6 605.6 605.6 605.6	53.732 11.415 11.415 11.415 11.415 11.415	0.0 0.0 0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0 0.0 0.0	
DATA FOR AXIAL REGION 3 OF FUEL ROD NUMBER 1										MASS WEIGHT % DISTRIBUTION IN LIQUID STEEL U OR AIC ZR 0			
7	0.610	1 2	SOLID SOLID	ZRO2 ZRY4	5 3	YES NO	0.0 4.205	4.135 4.815	583.5 582.7	583.1 582.7	29.416 11.415	0.0 0.0	0.0 0.0

図 4.13 第1回メジヤ - エディットの一部(3/4)

STRUCTURE OUTPUT SUMMARY

NODE	RING	IDCEL	MCHNO0	NAME	IFAIL	BNOOP	LIFE	FRAO	LOAD	STRESS	Avg TEMP	TSURF1	TSURF2
1	1	1	0	OUTER WALL	0	1	0.0	0.0	0.0	0.0	4.9040+02	5.0930+02	4.9000+02
2	1	2	0	OUTER WALL	0	1	0.0	0.0	0.0	0.0	4.9060+02	5.2620+02	4.9000+02
3	1	3	0	OUTER WALL	0	1	0.0	0.0	0.0	0.0	5.0120+02	5.7610+02	5.0000+02
4	1	4	0	OUTER WALL	0	1	0.0	0.0	0.0	0.0	5.2100+02	5.8940+02	5.2000+02
5	1	5	0	OUTER WALL	0	1	0.0	0.0	0.0	0.0	5.4090+02	5.9950+02	5.4000+02
6	1	6	0	OUTER WALL	0	1	0.0	0.0	0.0	0.0	5.0130+02	5.8620+02	5.0000+02
7	1	7	0	OUTER WALL	0	1	0.0	0.0	0.0	0.0	4.9040+02	5.5500+02	4.9000+02

DEBRIS FLAGS

NODE	RING
1	0.
2	0.
3	0.
4	0.
5	0.
6	0.
7	0.

FLUID RADIATION HEAT SOURCE, W/M**3

CELL	NODE	RING	FIELD 1	FIELD 2	FIELD 3
1	1	1	-1.363150+02	0.0	0.0
2	2	1	3.788420+02	0.0	0.0
3	3	1	2.452340+02	0.0	0.0
4	4	1	1.582460+02	0.0	0.0
5	5	1	8.489690+01	0.0	0.0
6	6	1	-3.527410+01	0.0	0.0
7	7	1	-7.178290+02	0.0	0.0

CELL NODE RING RADIATION HEAT FLUXES TO STRUCTURES, W/M**2

1	1	1	PINS	-5.574670+02	WALL	7.948640+02	0.0
2	2	1	PINS	-1.007610+03	WALL	1.807060+03	0.0
3	3	1	PINS	-7.502110+02	WALL	1.326290+03	0.0
4	4	1	PINS	-6.862710+02	WALL	1.247840+03	0.0
5	5	1	PINS	-6.331760+02	WALL	1.150480+03	0.0
6	6	1	PINS	-7.436130+02	WALL	1.351790+03	0.0
7	7	1	PINS	-4.292730+02	WALL	5.699250+02	0.0

図 4.13 第1回メジャーエディットの一部 (4/4)

FIELD 1 SOLUTION IN RING 1 AT TIME 4000.00						
AXIAL NODE	TEMPERATURE K	VOLUME FRACTION	RADIAL VELOCITY M/S	ALPHA=RHO KG/M**3	ALPHA=RHO.U J/M**3	DENSITY KG/M**3
1	723.016	0.9998882	-0.08730-05	0.0	0.1361	4.148770-05
2	1320.084	0.9999136	-6.29730-05	0.0	0.1110	4.461000-05
3	1552.322	0.9999245	-1.171690-04	0.0	0.1080	4.736070-05
4	1642.205	0.9999256	-1.721270-04	0.0	0.1037	4.691740-05
5	1649.776	0.9999262	-5.667910-04	0.0	0.1040	4.720060-05
6	1552.181	0.9999265	-4.997880-04	0.0	0.0420	2.394770-05
7	1239.857	0.9999268	-5.085270-04	0.0	0.0173	2.101970-05

HEAT TRANSFER COEFFICIENT (W/M**2-K) IN RING 1 AT TIME 4000.00						
NODE	HTC 1-2 V	HTC 1-2 L	HTC 1-3	HTC 1-L	HTC 2-3	HTC 2-4
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0

HEAT TRANSFER COEFFICIENT (W/M**2-K) IN RING 1 AT TIME 4000.00						
NODE	HTC 1-8	HTC 1-9	HTC 2-8	HTC 2-9	HTC 3-8	HTC 3-9
1	6.55780+01	5.408800+01	0.0	0.0	0.0	0.0
2	5.73750+01	4.905530+01	0.0	0.0	0.0	0.0
3	4.89220+01	4.274770+01	0.0	0.0	0.0	0.0
4	4.655020+01	4.788970+01	0.0	0.0	0.0	0.0
5	4.689220+01	3.892280+01	0.0	0.0	0.0	0.0
6	2.908910+01	2.614610+01	0.0	0.0	0.0	0.0
7	1.811100+01	1.572550+01	0.0	0.0	0.0	0.0

HEAT TRANSFER (W/M**3) IN RING 1 AT TIME 4000.00						
NODE	Q 1<->1 0	Q 2<->1 0	Q 1<->3 0	Q 1<->4 0	Q 2<->3 0	Q 2<->4 0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0

図 4.14 最終メッシュ - エディットの一部 (1/4)

HEAT TRANSFER (W/M**3) IN RING 1 AT TIME 4000.00							
NODE	Q 1<--8	Q 2<--8	Q 3<--8	Q 4<--8	Q 5<--8	Q 6<--8	Q 7<--8
1	-5.11752D+01	0.0	0.0	0.0	0.0	0.0	0.0
2	-1.48750D+02	0.0	0.0	0.0	0.0	0.0	0.0
3	-1.99891D+02	0.0	0.0	0.0	0.0	0.0	0.0
4	-1.94361D+02	0.0	0.0	0.0	0.0	0.0	0.0
5	-1.02008D+03	0.0	0.0	0.0	0.0	0.0	0.0
6	1.73461D+02	0.0	0.0	0.0	0.0	0.0	0.0
7	-1.62190D+02	0.0	0.0	0.0	0.0	0.0	0.0

INTERFACIAL AREAS (1/M) IN RING 1 AT TIME 4000.00							
NODE	AREA 1-2	AREA 1-3	AREA 1-4	AREA 2-3	AREA 2-4	AREA 3-4	
1	0.0	9.15594D-06	3.00000D-04	4.02228D-10	1.31793D-08	0.0	
2	0.0	1.59595D-06	3.00000D-04	4.59665D-11	8.64057D-09	0.0	
3	0.0	1.20000D-06	3.00000D-04	3.02175D-11	7.55438D-09	0.0	
4	0.0	9.56660D-05	3.00000D-04	2.373345D-09	7.44294D-09	0.0	
5	0.0	8.97954D-06	3.00000D-04	2.21014D-10	7.38392D-09	0.0	
6	0.0	6.00000D-04	3.00000D-04	1.47054D-08	7.35271D-09	0.0	
7	0.0	6.00000D-04	3.00000D-04	1.466449D-08	7.32244D-09	0.0	

GEOMETRY INFORMATION IN RING 1 AT TIME 4000.00							
NODE	STRUCTURE Z	FLOW AREA R	FLOW AREA M**2	VOLUME M**3	DELTA Z M	DELTA R M	
	VOL FRAC			M**3			
1	0.97723	7.19054D-04	0.0	2.54165D-03	1.01600D-01	8.92352D-02	
2	0.96528	9.31054D-04	0.0	2.54165D-03	1.01600D-01	8.92352D-02	
3	0.96029	1.00093D-03	0.0	2.54165D-03	1.01600D-01	8.92352D-02	
4	0.95969	1.01240D-03	0.0	2.54165D-03	1.01600D-01	8.92352D-02	
5	0.95937	1.01859D-03	0.0	2.54165D-03	1.01600D-01	8.92352D-02	
6	0.95920	1.02285D-03	0.0	2.54165D-03	1.01600D-01	8.92352D-02	
7	0.95903	1.11964D-03	0.0	2.54165D-03	1.01600D-01	8.92352D-02	

図 4.14 最終メッシュ一エティットの一部 (2/4)

***** CORE MODULE OUTPUT AT 4000.00 S *****

TOTAL HYDROGEN PRODUCED BY CORE = 0.010 KG
 TOTAL MASS TRANSFERRED FROM FUEL RODS = 0.0 KG
 TOTAL MASS TRANSFERRED FROM CONTROL RODS = 0.0 KG

TEMPERATURE DISTRIBUTION FOR FUEL ROD NUMBER 1 IN RING 1

CELL	737.8	737.7	737.5	737.4	737.3	737.2	737.1	737.0	737.1
1	1333.0	1333.0	1332.4	1331.8	1331.2	1330.6	1330.3	1330.3	1330.1
2	1329.9	1598.3	1598.3	1597.1	1595.9	1594.7	1593.8	1593.4	1560.4
3	1699.4	1699.4	1697.9	1696.4	1694.9	1694.9	1693.8	1693.4	1649.4
4	1708.6	1708.6	1707.0	1705.5	1704.0	1702.7	1702.7	1702.2	1655.7
5	1607.9	1607.9	1606.5	1605.1	1603.8	1602.5	1601.9	1560.0	1559.9
6	1259.2	1259.2	1258.4	1257.5	1256.6	1255.8	1255.3	1251.6	1251.6
7	1259.2	1259.2	1258.4	1257.5	1256.6	1255.8	1255.3	1251.6	1251.4

DISTRIBUTION OF HYDROGEN PRODUCED DURING THIS TIME STEP (G)

AXIAL NODE	RADIAL RING 1
1	0.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	0.00000

図 4.14 最終メジャーエネルギー分布図(3/4)

DETAILED CORE MODULE OUTPUT FOR RING 1

DATA FOR FUEL ROD NUMBER 1 THERE ARE 9.00 OF THESE IN THIS RING
 THE POWER FACTOR IS 1.00000+00 THE INTERNAL PRESSURE IS 2.000020+06 PA

DATA FOR AXIAL REGION 1 OF FUEL ROD NUMBER 1											
FLUIDS CELL	ZBOT (m)	RADIAL REGION	REGION TYPE	MATERIAL TYPE	NODES	GAP (mm)	RINNER (mm)	ROUTER (mm)	TINNER (K)	TOUTER (K)	MASS (g)
1	0.0	1	SOLID	ZRO2	5	NO	0.0	4.135	737.8	737.2	29.416
1	0.0	2	GAP LIQUID	U-ZR0-S	1	NO	4.135	4.205	737.2	0.651	84.63
1	0.0	3	SOLID	ZRY4	3	NO	4.205	4.815	737.2	11.415	0.0
1	0.0	4	NEW SOLID	U-ZR0-S	1	NO	4.815	6.292	737.2	34.030	0.0
DATA FOR AXIAL REGION 2 OF FUEL ROD NUMBER 1											
FLUIDS CELL	ZBOT (m)	RADIAL REGION	REGION TYPE	MATERIAL TYPE	NODES	GAP (mm)	RINNER (mm)	ROUTER (mm)	TINNER (K)	TOUTER (K)	MASS (g)
2	0.102	1	SOLID	U02	5	NO	0.0	4.135	1333.0	1330.3	53.732
2	0.102	2	GAP LIQUID	U-ZR0-S	1	NO	4.135	4.205	1330.3	1.318	12.69
2	0.102	3	SOLID	ZRY4	3	NO	4.205	4.741	1330.3	9.940	0.0
2	0.102	4	NEW SOLID	ZRO2	1	NO	4.741	4.852	1330.2	1330.0	1.993
2	0.102	5	NEW SOLID	U-ZR0-S	1	NO	4.852	5.386	1330.0	11.343	0.0
DATA FOR AXIAL REGION 3 OF FUEL ROD NUMBER 1											
FLUIDS CELL	ZBOT (m)	RADIAL REGION	REGION TYPE	MATERIAL TYPE	NODES	GAP (mm)	RINNER (mm)	ROUTER (mm)	TINNER (K)	TOUTER (K)	MASS (g)
3	0.203	1	SOLID	U02	5	YES	0.0	3.944	1598.3	1593.4	48.880
3	0.203	2	NEW SOLID	ZRO2	1	NO	4.536	4.949	1560.4	1559.7	6.869
3	0.203	3	LIQUID	U-ZR0-S	1	NO	4.949	4.959	1559.7	0.212	20.78
4	0.305	1	SOLID	U02	5	YES	0.0	3.949	1699.4	1693.4	49.003
4	0.305	2	NEW SOLID	ZRO2	1	NO	4.628	4.906	1649.4	1640.7	4.419
DATA FOR AXIAL REGION 4 OF FUEL ROD NUMBER 1											
FLUIDS CELL	ZBOT (m)	RADIAL REGION	REGION TYPE	MATERIAL TYPE	NODES	GAP (mm)	RINNER (mm)	ROUTER (mm)	TINNER (K)	TOUTER (K)	MASS (g)
5	0.406	1	SOLID	U02	5	YES	0.0	4.009	1708.4	1702.2	50.511
5	0.406	2	NEW SOLID	ZRO2	1	NO	4.689	4.876	1655.7	1655.2	2.050
6	0.508	1	SOLID	U02	5	YES	0.0	4.082	1607.9	1601.9	52.364
6	0.508	2	NEW SOLID	ZRO2	1	NO	4.722	4.861	1560.0	1559.7	2.472

図 4.14 最終メジャーエディツトの一部(4/4)

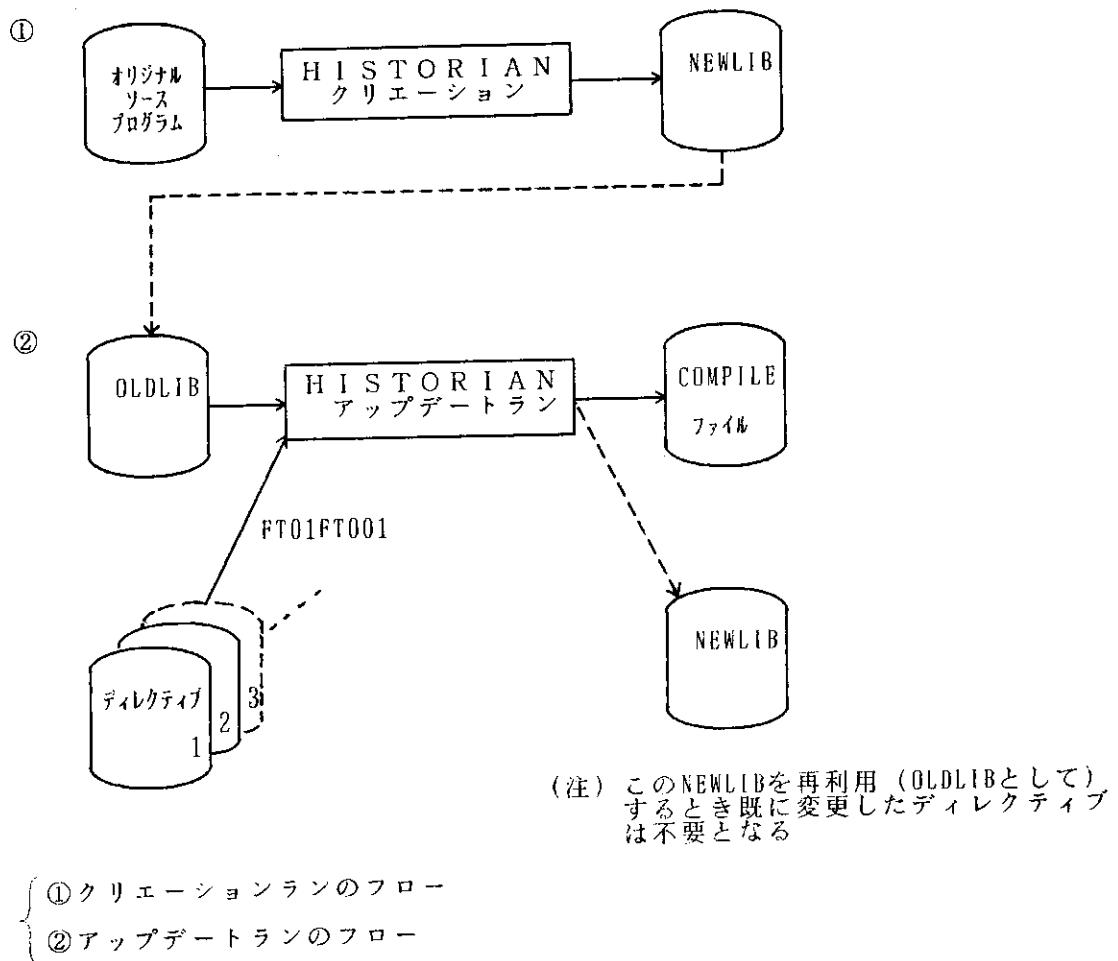


図 4.15 HISTORIANを用いたジョブの流れ

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.BBBBBBBB,1234.56          00040000
   T.1 C.0 W.0 I.1 OPN                         00050000
   OPTP NOTIFY=JXXXXX,PASSWORD=?????????        00060000
/*
//HISTOR EXEC PGM=HISTOR                     00070000
//STEPLIB DD DISP=SHR,DSN=J2286.HISTOR.LOAD    00080000
//FT05F001 DD *                                00090000
HISTORIAN(N,T,C=0)                           00100000
*LI 108                                         00110000
*READ 1                                         00120000
*/
//*   FT01F001 : INPUT DECK (*READ 1)           00130000
//*   FT06F001 : OUTPUT                          00140000
//*   FT21F001 : OLDLIB             (DIRECT)      00150000
//*   FT22F001 : NEWLIB             (DIRECT)      00160000
//*   FT23F001 : COMPILE            (SEQUENTIAL)  00170000
//*   FT25F001 : SCRATCH            (DIRECT)      00180000
//*   FT26F001 : SOURCE              (SEQUENTIAL)  00190000
//*   FT27F001 : SCRATCH            (DIRECT)      00200000
//*   FT28F001 : COMPATIBLE OLDLIB (SEQUENTIAL)  00210000
//*   FT29F001 : COMPATIBLE NEWLIB (SEQUENTIAL)  00220000
//*   FT28F001 : COMPATIBLE OLDLIB (SEQUENTIAL)  00230000
//*   FT29F001 : COMPATIBLE NEWLIB (SEQUENTIAL)  00240000
//FT01FO01 DD DSN=J3076.TK.MELPROG.FORT77,LABEL=(,,,IN),DISP=SHR 00250000
//FT06F001 DD DSN=&&COMPACT,                   00260000
//  UNIT=WK10,SPACE=(TRK,(50,50)),DISP=(NEW,PASS),       00270000
//  DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)           00280000
//FT22F001 DD DISP=(NEW,CATLG,DELETE),DSN=J3076.TK.MELPROG.LIB, 00290000
//  SPACE=(4080,(2000,500),RLSE),UNIT=D0950,          00300000
//  DCB=(RECFM=F,BLKSIZE=4080)                        00310000
//FT23F001 DD DUMMY                            00320000
//FT26F001 DD DUMMY                            00330000
//FT25F001 DD DISP=(,DELETE),UNIT=TSSWK,        00340000
//  SPACE=(4080,(50,10),RLSE)                      00350000
//FT27F001 DD DISP=(,DELETE),UNIT=TSSWK,SPACE=(4080,(50,10),RLSE) 00360000
//COMPACT EXEC PGM=JRQCPRT,PARM='TYPE2,CC=YES'     00370000
//UTYIN  DD DSN=&&COMPACT,DISP=(OLD,DELETE)      00380000
//UTYNLP DD SYSOUT=*                           00390000
//UTYLIST DD SYSOUT=*                           00400000
++                                           00410000
//                                           00420000

```

図 4.16 HISTORIAN NEWLIBのクリエーションラン JCL (1/2)
(コンパクト出力用)

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                               00020000
//SYSIN DD DATA,DLM='++'                         00030000
// JUSER 1234XXXX,AA.BBBBBBBB,1234.56           00040000
   T.1 C.0 W.0 I.1 OPN                          00050000
   OPTP NOTIFY=JXXXXX,PASSWORD=?????????        00060000
/*
//HISTOR EXEC PGM=HISTOR                      00070000
//STEPLIB DD DISP=SHR,DSN=J2286.HISTOR.LOAD    00080000
//FT05F001 DD *                                00090000
HISTORIAN(N,T,C=0)                           00100000
*LI 59                                         00110000
*READ 1                                         00120000
/*
//** HISTORIAN(N,T,C=0)                         00130000
//**      N: LIBRARY CREATION MODE               00140000
//**      T: USE 72 COLUMNS                        00150000
//**      D: USE 80 COLUMNS                        00160000
//**      L: USE 90 COLUMNS                        00170000
//**      C: OUTPUT NO COMPILE FILE              00180000
//**      FT01F001 : INPUT DECK (*READ 1)         00190000
//**      FT06F001 : OUTPUT                         00200000
//**      FT21F001 : OLDDLIB                      (DIRECT) 00210000
//**      FT22F001 : NEWLIB                       (DIRECT) 00220000
//**      FT23F001 : COMPILER                      (SEQUENTIAL) 00230000
//**      FT25F001 : SCRATCH                      (DIRECT) 00240000
//**      FT26F001 : SOURCE                        (SEQUENTIAL) 00250000
//**      FT27F001 : SCRATCH                      (DIRECT) 00260000
//**      FT28F001 : COMPATIBLE OLDDLIB          (SEQUENTIAL) 00270000
//**      FT29F001 : COMPATIBLE NEWLIB           (SEQUENTIAL) 00280000
//**      FT01F001 DD DSN=J3076.TK.MELPROG.FORT77,LABEL=(,,,IN),DISP=SHR 00290000
//**      FT06F001 DD SYSOUT=*
//**      FT22F001 DD DISP=(NEW,CATLG,DELETE),DSN=J3076.TK.MELPROG.LIB, 00300000
//**                  SPACE=(4080,(2000,500),RLSE),UNIT=D0950,          00310000
//**                  DCB=(RECFM=F,BLKSIZE=4080)                   00320000
//**      FT23F001 DD DUMMY                         00330000
//**      FT26F001 DD DUMMY                         00340000
//**      FT25F001 DD DISP=(,DELETE),UNIT=TSSWK,       00350000
//**                  SPACE=(4080,(50,10),RLSE)            00360000
//**      FT27F001 DD DISP=(,DELETE),UNIT=TSSWK,SPACE=(4080,(50,10),RLSE) 00370000
//**                                         00380000
//**                                         00390000
//**                                         00400000
//**                                         00410000
//**                                         00420000
*/

```

図 4.16 HISTORIAN NEWLIBのクリエーションラン JCL (2/2)
(プリンター出力用)

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.BBCCCCCC,1234.56          00040000
// T.2 C.0 W.4 I.4 OPN                         00050000
//OPTP NOTIFY=JXXXXX,PASSWORD=?????????        00060000
//***** EXECUTION HISTORIAN                  00070000
//HISTOR EXEC PGM=HISTOR                     00080000
//STEPLIB  DD DISP=SHR,DSN=J2286.HISTOR.LOAD   00090000
//FT05F001 DD *                                00100000
HISTORIAN(P,F,2,3)                            00110000
*LI 108                                         00120000
*READ 1                                         00130000
/*
//*  HISTORIAN(P,F,1,2,3)                       00140000
//*    P: UPDATE MODE                          00150000
//*    F: FULL MODE (FULL DECK(S) OUTPUT TO COMPILE FILE) 00160000
//*    1: PRINT ECHO BACK FOR DIRECTIVE       00170000
//*    2: PRINT *CALL, *IF, *ELSE, AND *ENDIF LINE(S) 00180000
//*    3: PRINT DELETE AND INSET LINE(S)      00190000
//*FT01F001 : INPUT DECK (*READ 1)            00200000
//*FT06F001 : OUTPUT                           00210000
//*FT21F001 : OLDDLIB             (DIRECT)     00220000
//*FT22F001 : NEWLIB              (DIRECT)     00230000
//*FT23F001 : COMPILE               (SEQUENTIAL) 00240000
//*FT25F001 : SCRATCH              (DIRECT)     00250000
//*FT26F001 : SOURCE                (SEQUENTIAL) 00260000
//*FT27F001 : SCRATCH              (DIRECT)     00270000
//*FT28F001 : COMPATIBLE OLDDLIB (SEQUENTIAL) 00280000
//*FT29F001 : COMPATIBLE NEWLIB (SEQUENTIAL) 00290000
//*FT01F001 DD DSN=J3076.TK.MELPROG.DIRECTIV,LABEL=(,,IN),DISP=SHR 00300000
//          DD DSN=J3076.TK.MELPNEW.DIRECTIV,LABEL=(,,IN),DISP=SHR 00310000
//*FT06F001 DD DISP=(NEW,CATLG,DELETE),DSN=J3076.TK.@@HIST.LIST, 00320000
//          DD DISP=(NEW,PASS),DSN=&&COMPACT, 00330000
//          UNIT=TSSWK,SPACE=(TRK,(50,50),RLSE), 00340000
//          DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043) 00350000
//FT21F001 DD DISP=SHR,DSN=J3076.TK.MELPROG.LIB 00360000
//FT22F001 DD DISP=(NEW,DELETE,DELETE),DSN=&&NEWLIB, 00370000
//          SPACE=(4080,(2000,500),RLSE),UNIT=TSSWK, 00380000
//          DCB=(RECFM=F,BLKSIZE=4080) 00390000
//FT23F001 DD SPACE=(TRK,(50,50),RLSE),UNIT=TSSWK, 00400000
//          DCB=(LRECL=80,BLKSIZE=6400,RECFM=FB), 00410000
//          DISP=(NEW,PASS,DELETE),DSN=&&COMP 00420000
//*DISP=(NEW,CATLG,DELETE),DSN=J3076.TK.@@COMPIL.FORT77 00430000
//*FT23F001 DD DISP=SHR,DSN=J3076.TK.MELPNWPS.FORT77 00440000
//FT26F001 DD DUMMY                           00450000
//FT25F001 DD DISP=(,DELETE),UNIT=TSSWK, 00460000
//          SPACE=(4080,(50,10),RLSE) 00470000
//FT27F001 DD DISP=(,DELETE),UNIT=TSSWK,SPACE=(4080,(50,10),RLSE) 00480000
//***** PRINT OUT HISTORIAN OUTPUT           00490000
//COMPACT EXEC PGM=JRQCPRT,PARM='TYPE2,CC=YES' 00500000
//*UTYIN  DD DSN=J3076.TK.@@HIST.LIST,DISP=(OLD,DELETE) 00510000
//UTYIN  DD DSN=&&COMPACT,DISP=(OLD,DELETE) 00520000
//UTYNLP DD SYSOUT=*                         00530000
//UTYLIST DD SYSOUT=*                         00540000
//***** FORTRAN AND LINKAGE                 00550000
//*FORT77 EXEC FORT77,SO='J3076.TK.@@COMPIL', 00560000
//*          A='AUTODBL(DBLPAD),ALC,OBJ,NOSOURCE,INCLUDE,FLAG(I)' 00570000
//*LKED   EXEC LKED,B=MAP                      00580000
//+
//+
//*

```

図 4.17 HISTORIAN アップデイトラン JCL (1/2)
(コンパクト出力用)

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                     00030000
// JUSER 1234XXXX,AA.BBBBBBBB,1234.56        00040000
   T.2 C.0 W.4 I.2 OPN                      00050000
   OOPTP NOTIFY=JXXXXX,PASSWORD=?????????    00060000
/*
//HISTOR EXEC PGM=HISTOR                   00070000
//STEPLIB DD DISP=SHR,DSN=J2286.HISTOR.LOAD 00080000
//FT05F001 DD *                            00090000
HISTORIAN(P,F,2,3)                         00100000
*LI 59                                      00110000
*READ 1                                     00120000
/*
//** HISTORIAN(P,F,1,2,3)                    00130000
//**   R: UPDATE MODE                      00140000
//**   F: FULL MODE (FULL DECK(S) OUTPUT TO COMPILE FILE) 00150000
//**   1: PRINT ECHO BACK FOR DIRECTIVE      00160000
//**   2: PRINT *CALL, *IF, *ELSE, AND *ENDIF LINE(S) 00170000
//**   3: PRINT DELETE AND INSET LINE(S)     00180000
//** FT01F001 : INPUT DECK (*READ 1)        00190000
//** FT06F001 : OUTPUT                      00200000
//**   FT21F001 : OLDLIB          (DIRECT)    00210000
//**   FT22F001 : NEWLIB          (DIRECT)    00220000
//**   FT23F001 : COMPILER         (SEQUENTIAL) 00230000
//**   FT25F001 : SCRATCH         (DIRECT)    00240000
//**   FT26F001 : SOURCE          (SEQUENTIAL) 00250000
//**   FT27F001 : SCRATCH         (DIRECT)    00260000
//**   FT28F001 : COMPATIBLE OLDLIB (SEQUENTIAL) 00270000
//**   FT29F001 : COMPATIBLE NEWLIB (SEQUENTIAL) 00280000
//FT01F001 DD DSN=J3076.TK.MELPROG.DIRECTIV,LABEL=(,,,IN),DISP=SHR 00290000
//          DD DSN=J3076.TK.MELPNEW.DIRECTIV,LABEL=(,,,IN),DISP=SHR 00300000
//FT06F001 DD SYSPUT=*                      00310000
//FT21F001 DD DISP=SHR,DSN=J3076.TK.MELPROG.LIB 00320000
//FT22F001 DD DISP=(NEW,DELETE,DELETE),DSN=&NEWLIB, 00330000
//          SPACE=(4080,(2000,500),RLSE),UNIT=TSSWK, 00340000
//          DCB=(RECFM=F,BLKSIZE=4080)           00350000
//FT23F001 DD SPACE=(TRK,(50,50),RLSE),UNIT=TSSWK, 00360000
//          DCB=(LRECL=80,BLKSIZE=6400,RECFM=FB), 00370000
//          DISP=(NEW,PASS,DELETE),DSN=&COMPIL 00380000
//          DISP=(NEW,CATLG,DELETE),DSN=J3076.TK.&COMPIL.FORT77 00390000
//FT26F001 DD DUMMY                          00400000
//FT25F001 DD DISP=(,DELETE),UNIT=TSSWK, 00410000
//          SPACE=(4080,(50,10),RLSE)           00420000
//FT27F001 DD DISP=(,DELETE),UNIT=TSSWK,SPACE=(4080,(50,10),RLSE) 00430000
*/
//+
//+

```

図 4.17 HISTORIAN アップデイトラン JCL (2/2)
(プリンター出力用)

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.88888888,1234.56          00040000
   T.4 W.2 I.0 C.4 OPN                         00050000
   OOPTP NOTIFY=JXXXXX,PASSWORD=?????????      00060000
00070000
//MELPROG EXEC PGM=MELPROG                  00080000
//STEPLIB DD DISP=SHR,DSN=J3076.TK.MELPROG.LOAD 00090000
//FT95F001 DD DISP=SHR, (MELIN ) INPUT DATA FILE 00100000
// LABEL=(,,,IN),                                00110000
// DSN=J3076.TK.MELPROG.DATA(MELINSS)          00120000
//FT97F001 DD DISP=SHR, (MELMCH) MECHANICAL PROPERTY INPUT FILE 00130000
// LABEL=(,,,IN),                                00140000
// DSN=J3076.TK.MELPROG.DATA(MELMECH)          00150000
//FT99F001 DD *           TIME DATA ABOVE T.N    00160000
   4           * TIME T.N                         00170000
   2 , 0     * SPLPACK LABEL INFORMATION FLAG AND DEBUG FLAG. 00180000
/*
///* OPTION 1 BIT : PRINT SPL LABELS      ( SET 1 ) 00190000
///* OPTION 2 BIT : PRINT SPL DUMP TIMES ( SET 2 ) 00200000
///* OPTION 3 BIT : PRINT SPL LOCATIONS ( SET 4 ) 00210000
///* DCB=(LRECL=144,BLKSIZE=8640,RECFM=FBA)        00220000
//DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)          00230000
//DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)          00240000
//DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)          00250000
//DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)          00260000
//FT71F001 DD SYSOUT=*, (TTY ) OUTPUT TO TERMINAL 00270000
//DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)          00280000
//FT91F001 DD SYSOUT=*, (TTY ) OUTPUT TO TERMINAL 00290000
//DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)          00300000
//FT21F001 DD DUMMY      ( ) GRAPHIC OUTPUT FILE FOR SPL PLOT 00310000
//FT92F001 DD DUMMY      (MELGRF) GRAPHIC OUTPUT FILE 00320000
//FT93F001 DD DUMMY      (MELRST) RESTART INPUT FILE 00330000
//*FT96F001 DD DUMMY      ( ) MELPROG OUTPUT FILE (LINE PRINTER) 00340000
//FT96F001 DD SYSOUT=*, (MELOUT) MELPROG OUTPUT FILE (LINE PRINTER) 00350000
//DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)          00360000
//*FT98F001 DD DUMMY      (MELDMP) RESTART OUTPUT FILE 00370000
//FT98F001 DD DISP=SHR, (MELDMP) RESTART OUTPUT FILE 00380000
//DSN=J3076.TK.MELPRST.DATA                      00390000
//*FT98F001 DD UNIT=00950, (MELDMP) RESTART OUTPUT FILE 00400000
//*DSN=J3076.TK.MELPRST.DATA,DISP=(NEW,CATLG,DELETE), 00410000
//*SPACE=(TRK,(20,20),RLSE)                       00420000
//FT82F001 DD DUMMY      ( ) PLOT    OUTPUT FILE 00430000
//FT81F001 DD DUMMY      ( ) GRAPH   OUTPUT FILE 00440000
//FT54F001 DD DUMMY      ( )          OUTPUT FILE 00450000
//+
//+

```

図 4.18 MELPROG NEW ラン JCL (1/2)

(サンプル入力データ用：定常計算)

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                     00030000
// JUSER 1234XXXX,AA.38898888,1234.56       00040000
   T.7 W.4 I.3 C.4 OPN                      00050000
   OOPT NOTIFY=JXXXXX,PASSWORD=?????????    00060000
//*   T.8 W.5 I.3 C.4 OPN NGT                 00070000
//MELPROG EXEC PGM=MELPNEW                  00080000
//STEPLIB DD DISP=SHR,DSN=J3076.TK.MELPROG.LOAD 00090000
//FT95F001 DD DISP=SHR, (MELIN ) INPUT DATA FILE 00100000
// LABEL=(,,,IN),
//  DSN=J3076.TK.MELPROG.DATA(DF301)          00110000
//FT97F001 DD DISP=SHR, (MELMCH) MECHANICAL PROPERTY INPUT FILE 00120000
// LABEL=(,,,IN),
//  DSN=J3076.TK.MELPROG.DATA(MELMECH)        00130000
//FT99F001 DD *                                TIME DATA ABAVE T.N 00140000
   7           * TIME T.N                      00150000
   0 , 0      * SPLPACK LABEL INFORMATION FLAG AND DEBUG FLAG. 00160000
/*
//*2 , 0      * SPLPACK LABEL INFORMATION FLAG AND DEBUG FLAG. 00170000
//* OPTION 1 BIT : PRINT SPL LABELS ( SET 1 ) 00180000
//* OPTION 2 BIT : PRINT SPL DUMP TIMES ( SET 2 ) 00190000
//* OPTION 3 BIT : PRINT SPL LOCATIONS ( SET 4 ) 00200000
//FT06F001 DD SYSOUT=*, (          ) FORTRAN TRACE BACK AND ERROR 00210000
//  DCB=(LRECL=144,BLKSIZE=8640,RECFM=FBA)      00220000
//FT07F001 DD SYSOUT=*, (          ) ERROR OUTPUT MESSAGE 00230000
//  DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)      00240000
//FT71F001 DD SYSOUT=*, (TTY ) OUTPUT TO TERMINAL 00250000
//  DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)      00260000
//FT91F001 DD SYSOUT=*, (TTY ) OUTPUT TO TERMINAL 00270000
//  DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)      00280000
//FT21F001 DD DISP=SHR, (          ) GRAPHIC OUTPUT FILE FOR SPL PLOT 00290000
//  DSN=J3076.TK.MELPNWSP.DATA                00300000
//FT92F001 DD DUMMY   (MELGRF) GRAPHIC OUTPUT FILE 00310000
//FT93F001 DD DUMMY   (MELRST) RESTART INPUT FILE 00320000
//*FT96F001 DD DUMMY                           00330000
//FT96F001 DD SYSOUT=*, (MELOUT) MELPROG OUTPUT FILE (LINE PRINTER) 00340000
//  DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)      00350000
//*FT98F001 DD DUMMY   (MELDMP) RESTART OUTPUT FILE 00360000
//FT98F001 DD DISP=SHR, (MELDMP) RESTART OUTPUT FILE 00370000
//  DSN=J3076.TK.MELPNEWR.DATA                00380000
//*FT98F001 DD UNIT=D0950, (MELDMP) RESTART OUTPUT FILE 00390000
//*.  DSN=J3076.TK.MELPNEWR.DATA,DISP=(NEW,CATLG,DELETE), 00400000
//* SPACE=(TRK,(20,20),RLSE)                   00410000
//FT82F001 DD DUMMY   (          ) PLOT     OUTPUT FILE 00420000
//FT81F001 DD DUMMY   (          ) GRAPH    OUTPUT FILE 00430000
//FT54F001 DD DUMMY   (          )             OUTPUT FILE 00440000
++
//*

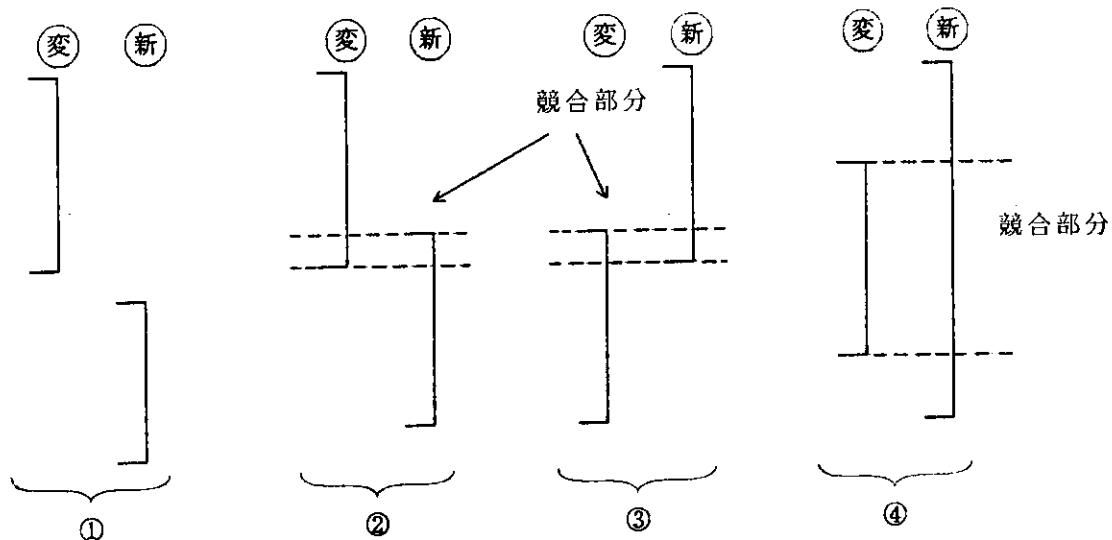
```

図 4.18 MELPROG NEW ラン JCL (2/2)
(実解析データ用)

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                               00020000
//SYSIN DD DATA,DLM='++'                         00030000
// JUSER 1234XXXX,AA.BBBBBBSS,1234.56           00040000
// T.6 W.4 I.1 C.4 OPN                          00050000
//OPTP NOTIFY=JXXXXX,PASSWORD=?????????          00060000
//MELPROG EXEC PGM=MELPNEW                      00070000
//STEPLIB DD DISP=SHR,DSN=J3076.TK.MELPROG.LOAD   00080000
//FT95FO01 DD DISP=SHR, (MELIN ) INPUT DATA FILE 00090000
// LABEL=(.,IN),                                00100000
// DSN=J3076.TK.MELPROG.DATA(DF310)             00110000
//FT97FO01 DD DISP=SHR, (MELMCH) MECHANICAL PROPERTY INPUT FILE 00120000
// LABEL=(.,IN),                                00130000
// DSN=J3076.TK.MELPROG.DATA(MELMECH)           00140000
//                                         TIME DATA ABAVE T.N 00150000
//FT99FO01 DD *                                 00160000
//   * TIME T.N                                 00170000
// 2 , 0    * SPLPACK LABEL INFORMATION FLAG AND DEBUG FLAG. 00180000
/*
//** OPTION 1 BIT : PRINT SPL LABELS      ( SET 1 ) 00190000
//** OPTION 2 BIT : PRINT SPL DUMP TIMES ( SET 2 ) 00200000
//** OPTION 3 BIT : PRINT SPL LOCATIONS ( SET 4 ) 00210000
//FT06FO01 DD SYSOUT=*, (          ) FORTRAN TRACE BACK AND ERROR 00220000
// DCB=(LRECL=144,BLKSIZE=8640,RECFM=FBA)        00230000
//FT07FO01 DD SYSOUT=*, (          ) ERROR OUTPUT MESSAGE 00240000
// DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)        00250000
//FT71FO01 DD SYSOUT=*, (TTY ) OUTPUT TO TERMINAL 00260000
// DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)        00270000
//FT91FO01 DD SYSOUT=*, (TTY ) OUTPUT TO TERMINAL 00280000
// DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)        00290000
//FT21FO01 DD DISP=SHR, (          ) GRAPHIC OUTPUT FILE FOR SPL PLOT 00300000
// DSN=J3076.TK.MELPSPL.DATA                   00310000
//FT92FO01 DD DUMMY, (MELGRF) GRAPHIC OUTPUT FILE 00320000
//FT93FO01 DD DISP=SHR, (MELRST) RESTART INPUT FILE 00330000
// DSN=J3076.TK.MELPNEWR.DATA                 00340000
//**FT96FO01 DD DUMMY                         00350000
//FT96FO01 DD SYSOUT=*, (MELOUT) MELPROG OUTPUT FILE (LINE PRINTER) 00360000
// DCB=(LRECL=133,BLKSIZE=7980,RECFM=FBA)        00370000
//FT98FO01 DD DUMMY, (MELDMP) RESTART OUTPUT FILE 00380000
//FT82FO01 DD DUMMY, (          ) PLOT OUTPUT FILE 00390000
//FT81FO01 DD DUMMY, (          ) GRAPH OUTPUT FILE 00400000
//FT54FO01 DD DUMMY, (          ) OUTPUT FILE    00410000
//                                         00420000
++                                     00430000
//
```

図 4.19 MELPROG リスタートラン JCL



(変) : 変換・整備により修正の必要な箇所
 (新) : 新バージョンにより修正の必要な箇所

①と④は修正不要
②と③は修正要

図 4.20 新バージョンのディレクティブが変換版と競合する場合

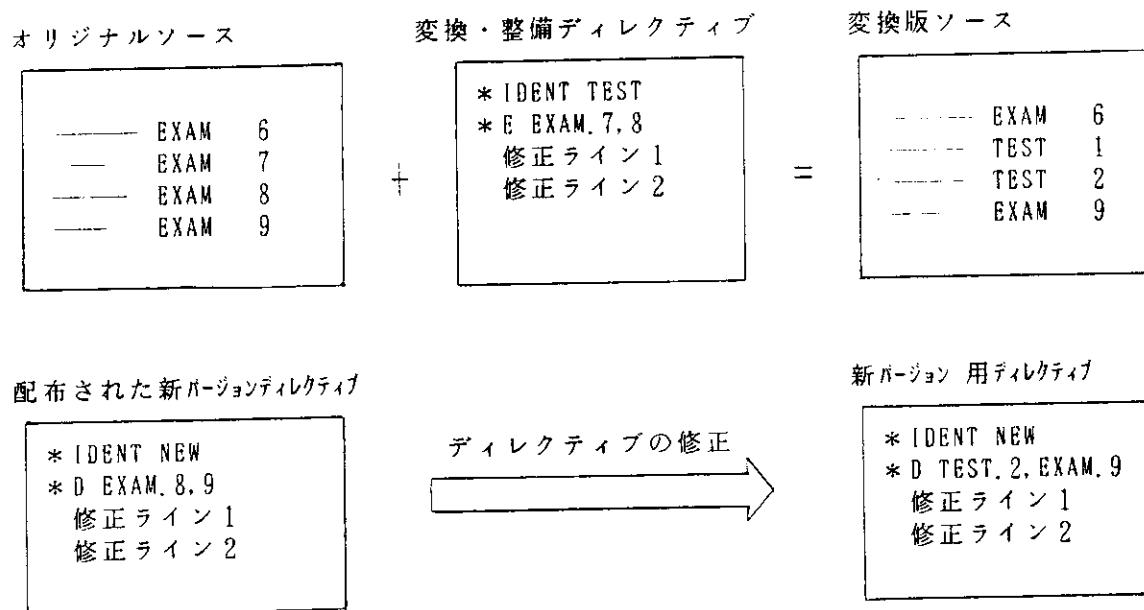


図 4.21 新バージョンのディレクティブが変換版と競合したときの修正例

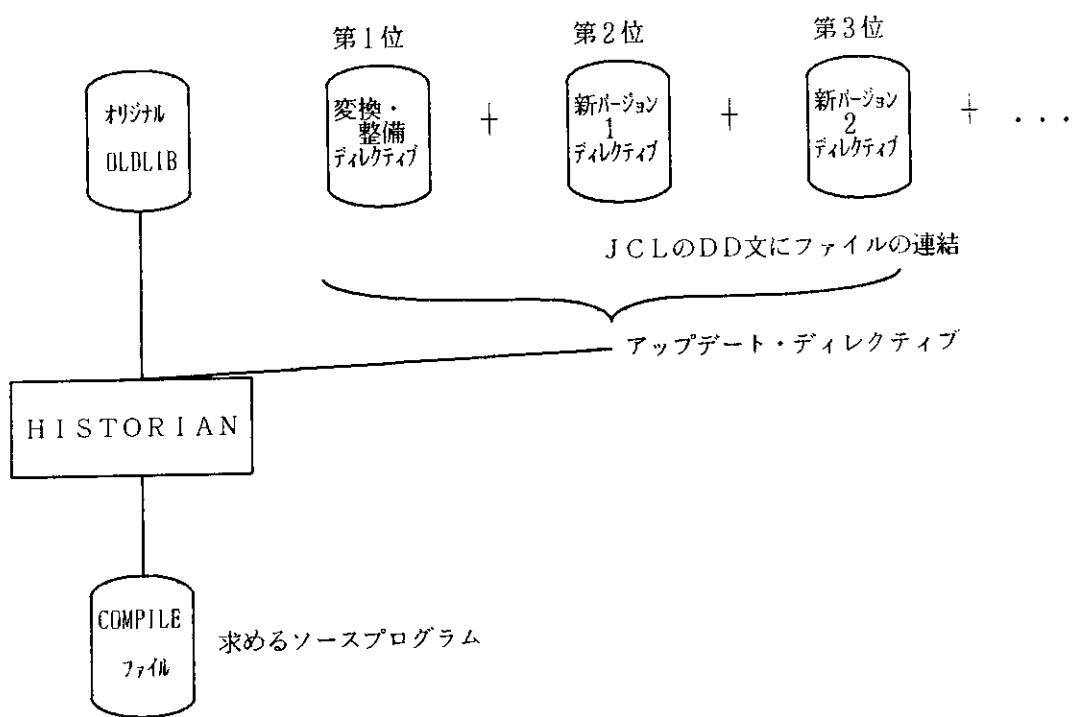


図 4.22 新バージョン作成のためHISTORIANアップディート方法

付録A MELPROG 変換用ツールの FORTRAN プログラム

A. 1 変換用ツールのメンバー名とその処理内容

表 A.1 に MELPROG 変換用プログラムツールのメンバー名とそのライン数を示す。また、以下に各メンバの解説を記す。

①CHGSYM

これは、新しく MELPROG のアップデート・ディレクティブを入手した際、現バージョンと同じ 6 文字を超えるシンボル名が出現した場合、そのシンボル名を変更するものであり、同時に *CALL 文を *INCLUDE 文に変更する。

②COMPDECK

これは、COMPSOC と同じ働きをするが、フルスペルのディレクティブを対象としたものである。CARD ID の再番号付けも行う。リストは省略する。

③COMPNE

これは、COMPSOC と同じ働きをするが、変換後のソースプログラムを対象とし、CARD ID の再番号付けは行わない。リストは省略する。

④COMPSOC

これは、MELPROG のオリジナル・ソースプログラムの CARD ID の再番号付けを行い、コンパクト出力するためのファイルを作成する。同時に COMDECK 及び、DECK 名の出力ページ一覧も出力する。

⑤DIRECTIV

これは、変換・整備ソース・プログラムから HISTORIAN用ディレクティブを作成するものである。

⑥UPDCHG

これは、オリジナルのソース・プログラムから CARD ID の再番号付を行うものである。CARD ID の最初の 4 衔は COMDECK 名あるいは DECK 名の最初の 4 文字を付け、次の 4 衔はカードの順番をつける。

⑦UPDLIST

これは、アップデート・ディレクティブラインの内ディレクティブのみを出力するものである。これにより変換・整備用ディレクティブとの整合性をチェックする。

A. 2 変換用ツールの FORTRAN プログラムリスト

(1) CHGSSYM

```

C          00010004
C  CHANGE EXCEED 6 CHARACTERS SYMBOL TO 6 CAHRACTERS SYMBOL. 00020004
C          00030004
C          IMPLICIT INTEGER (A-Z) 00040004
C          PARAMETER ( QNAMES = 200 ) 00050000
C          CHARACTER*9 OLDNAM(QNAMES),NEWNAM(QNAMES) 00060000
C          CHARACTER*9 OLDN      /NEWN 00070000
C          CHARACTER*80 CARD,CARDWK 00080005
C          CHARACTER*1 HEAD 00090001
C          DIMENSION NAMLEN(QNAMES) 00100000
C          DATA CARDWK / '*INCLUDE' / 00110005
C          NAMES = 0 00120000
C          DEBUG = 0 00130003
C          INP   = 1 00140000
C          OUT   = 2 00150000
C          NAMTBL = 3 00160000
C          CR    = 5 00170000
C          LP    = 6 00180000
C          INF   = 10 00190006
C 100 CONTINUE 00200000
C          READ (NAMTBL,1000,END=200) HEAD,OLDN,NEWN 00210001
C 1000 FORMAT (A1,20X,A8,9X,A8) 00220002
C          IF ( HEAD .EQ. '*' ) GO TO 100 00230001
C          NAMES = NAMES + 1 00240000
C          IF ( NAMES .GT. QNAMES ) 00250000
C          THEN 00260000
C          WRITE (LP,*)' ----- SYMBOL NAMES ARRAY IS OVER.' 00270006
C          STOP 16 00280000
C          ENDIF 00290000
C          OLDNAM(NAMES) = OLDN 00300000
C          NEWNAM(NAMES) = NEWN 00310000
C          NAMLEN(NAMES) = INDEX(OLDN(1:9),',') - 1 00320000
C          IF ( DEBUG .EQ. 1 ) WRITE (LP,6910) NAMES,OLDN,NEWN 00330006
C 6910 FORMAT (' SEQ,OLD NAME, NEW NAME = ',I5,2X,A,2X,A) 00340001
C          GO TO 100 00350000
C 200 CONTINUE 00360000
C          IF ( NAMES .EQ. 0 ) 00370000
C          THEN 00380000
C          WRITE (LP,*)' ----- NO. OLD SYMBOL(S) IS 0.' 00390000
C          STOP 16 00400000
C          ENDIF 00410000
C          00420000
C          INPL   = 0 00430000
C          CHANGE = 0 00440000
C          CHGLIN = 0 00450005
C 300 CONTINUE 00460000
C          READ (INP,'(A)',END=600) CARD 00470000
C          INPL = INPL + 1 00480000
C          IF ( CARD(1:1) .NE. '*' .AND. CARD(1:1) .NE. 'C' ) 00490005
C          THEN 00500000
C          EXIST = 0 00510005
C          DO 500 I=1,NAMES 00520005
C 400  CONTINUE 00530005
C          FOUND = INDEX(CARD(1:72),OLDNAM(I)(1:NAMLEN(I))) 00540000
C          IF ( FOUND .NE. 0 ) 00550000
C          THEN 00560000
C          CHANGE = CHANGE + 1 00570000
C          IF ( EXIST .EQ. 0 ) 00580005
C          THEN 00590005
C          WRITE (INF,6010) 'OLD',INPL,CARD 00600006
C          WRITE (OUT,'(*/*',',A)') CARD(1:76) 00610005
C          ENDIF 00620005
C          EXIST = EXIST + 1 00630005
C          ST = FOUND 00640000
C          EN = FOUND + NAMLEN(I) - 1 00650000
C          CARD(ST:EN) = NEWNAM(I)(1:NAMLEN(I)) 00660000
C          GO TO 400 00670005
C          ENDIF 00680000
C 500  CONTINUE 00690000
C          IF ( EXIST .NE. 0 ) 00700005
C          THEN 00710005
C          WRITE (INF,6010) 'NEW',EXIST,CARD 00720006
C          CHGLIN = CHGLIN + 1 00730005
C          ENDIF 00740005

```

```

      ELSE          00750005
      .IF ( CARD(1:6) .EQ. '*CALL' ) 00760005
      .  THEN        00770005
      .    WRITE (OUT,'(*/*/,A)') CARD(1:76) 00780006
      .    WRITE (INF,6010) 'OLD',INPL,CARD 00790006
      .    CARDWK(10:80) = CARD(7:80) 00800005
      .    CARD         = CARDWK 00810005
      .    CHGLIN       = CHGLIN + 1 00820005
      .    WRITE (INF,6010) 'NEW',INPL,CARD 00830006
      .  ELSE        00840005
      .  .IF ( CARD(1:4) .EQ. '*CA' ) 00850005
      .  .  THEN      00860005
      .  .    WRITE (OUT,'(*/*/,A)') CARD(1:76) 00870006
      .  .    WRITE (INF,6010) 'OLD',INPL,CARD 00880006
      .  .    CARDWK(10:80) = CARD(5:80) 00890005
      .  .    CARD         = CARDWK 00900005
      .  .    CHGLIN       = CHGLIN + 1 00910005
      .  .    WRITE (INF,6010) 'NEW',INPL,CARD 00920006
      .  .  ENDIF     00930000
      .  .  WRITE (OUT,'(A)') CARD 00940000
      .  .  GO TO 300 00950000
      . 600 CONTINUE 00960000
      .  .  WRITE (LP,6020) INPL,CHGLIN,CHANGE 00970006
      .  .  WRITE (INF,6020) INPL,CHGLIN,CHANGE 00980006
      .  .  STOP      00990000
      . 6010 FORMAT (1X,A,I5,2X,A) 01000005
      . 6020 FORMAT ('/NO. OF INPUT LINE(S) = ',I6/ 01010000
      .  .  'NO. OF CHANGED LINE(S) = ',I6/ 01020005
      .  .  'NO. OF CHANGED SYMBOL(S) = ',I6) 01030005
      .  END        01040005
                           01050000

```

④ COMPSOC

```

C   SOURCE COMPACT PRINT TOOL.  ( CODED BY 'UPD' CD AND DK ) 00010007
C   THIS SOURCE IS USED FOR ORIGINAL SOURCE. 00020007
C
C   IMPLICIT INTEGER ( A-Z ) 00030007
C   CHARACTER=72 IA 00040007
C   CHARACTER IB=4,IC=4, NAME*10, ID*4,CDDK*4 00050004
C   CHARACTER           SNAME*10, SID*4 00060003
C   CHARACTER ENDDEC*8 00070003
C   PARAMETER ( QMAXNM = 300 ) 00080000
C   CHARACTER*50 DAT(QMAXNM) 00090001
C   EQUIVALENCE ( IA(1:1) , IB (1:1) ) 00100002
C   EQUIVALENCE ( IA(5:5) , NAME(1:1) ) 00110002
C   DATA IC, ID / '*CD' , '*DK' / 00120000
C
C   DEBUG = 1 00130000
C   DEBUG = 0 00140000
C   INS = 0 00150001
C   PAGE = 0 00160001
C   NEWFIL = 175 00170000
C   LP = 10 00180000
C   INP = 1 00190004
C   INF = 2 00200001
C   INFP = 3 00210000
C   CSEQ = 0 00220002
C   DSEQ = 0 00230002
C
C   MAXLIN = 107 00240000
C   MAXLIN = 80 00250000
C   LINES = MAXLIN 00260000
C   ONESUB = 0 00270001
C   SSEQ = 0 00280000
C   SPAGE = 0 00290000
                           00300000
                           00310000

```

```

      WRITE (1NF,2000)                                     00320001
100 CONTINUE                                         00330000
      IF ( DEBUG .EQ. 1 )                               00340000
      . THEN                                           00350000
      . IF ( INS .GT. 2000 ) GO TO 900               00360000
      ENDIF                                           00370000
      READ (INP,'(A)',END=900) IA                   00380000
      INS = INS + 1                                 00390000
      LINES = LINES + 1                            00400000
      IF ( LINES .GT. MAXLIN )                      00410000
      . THEN                                           00420000
      . LINES = 1                                    00430000
      PAGE = PAGE + 1                           00440000
      IF ( MOD(PAGE,NEWFIL) .EQ. 1 )                00450001
      . THEN                                           00460001
      . REWIND LP                                00470001
      LP = LP + 1                                00480001
      WRITE(6,*), ' START FILE NO. = ',LP          00490006
      ENDIF                                           00500001
C      WRITE (LP,6010) PAGE                         00510000
      PPAGE = PAGE                                00520000
      ELSE                                            00530000
      PPAGE = 0                                    00540000
      ENDIF                                           00550000
      IF ( IB .EQ. IC )                           00560000
      . THEN                                           00570000
      CSEQ = CSEQ + 1                           00580000
      CDDK = NAME                                00590003
C      WRITE (1NF,2010) SCSEQ,IC,NAME,PAGE        00600001
      IF ( SSEQ .NE. 0 )                           00610001
      . WRITE (1NF,2010) SSEQ,SID,SNAME,SPAGE,ONESUB,INS-1 00620001
      SSEQ = CSEQ                                00630000
      SPAGE = PAGE                               00640000
      SNAME = NAME                               00650000
      SID = IC                                  00660000
      ONESUB = 0                                 00670000
      ELSE                                           00680000
      . IF ( IB .EQ. ID )                         00690000
      . THEN                                           00700000
      DSEQ = DSEQ + 1                           00710000
      CDDK = NAME                                00720003
C      WRITE (1NF,2010) DSEQ,ID,NAME,PAGE        00730001
      IF ( SSEQ .NE. 0 )                           00740001
      . WRITE (1NF,2010) SSEQ,SID,SNAME,SPAGE,ONESUB,INS-1 00750001
      SSEQ = DSEQ                                00760000
      SPAGE = PAGE                               00770000
      SNAME = NAME                               00780000
      SID = ID                                  00790000
      ONESUB = 0                                 00800000
      ENDIF                                           00810000
C      WRITE (LP,6020) INS,IA                   00820000
      ONESUB = ONESUB + 1                        00830003
      WRITE (LP,6020) IA,CDDK,ONESUB,PPAGE       00840003
      GO TO 100                                 00850000
900 CONTINUE                                         00860000
      WRITE (1NF,2010) SSEQ,SID,SNAME,SPAGE,ONESUB,INS 00870001
      WRITE (1NF,2020) INS,CSEQ,DSEQ              00880000
      REWIND LP                                00890001
      WRITE (6,9910) LP-10                         00900003
      REWIND INF                               00910001
      CALL SORT ( INF,DAT,QMAXNM,IA,IC,ENDDEC,NP ) 00920002
      WRITE (INFP,7010) IC                         00930002
      DO 1100 I=1,NP                           00940002
      WRITE (INFP,7020) I,DAT(I)                 00950002
1100 CONTINUE                                         00960002
      CALL SORT ( INF,DAT,QMAXNM,IA,ID,ENDDEC,NP ) 00970002
      WRITE (INFP,7010) ID                         00980002
      DO 1200 I=1,NP                           00990002
      WRITE (INFP,7020) I,DAT(I)                 01000002
1200 CONTINUE                                         01010002
      STOP                                           01020002
9910 FORMAT (' NO. OF FILE IS ',I2,'.')           01030003
C6010 FORMAT (' PAGE : ',I5.5)                  01040003
C6020 FORMAT (' ',I6,2X,A)                      01050000
6020 FORMAT (' ',A,A,I4,4X,I5.0)                01060003
7010 FORMAT (' ',A,' SEQ.          PAGE LINES TOTAL') 01070002
7020 FORMAT (' ',I4,A)                          01080002
2000 FORMAT (' ',I4,SEQ.          PAGE LINES TOTAL') 01090002
2010 FORMAT (' ',I5,3X,A,A,I7,I7,I7)            01100001
2020 FORMAT (
      // 'ENDDECK'                                /
      . ' NO. OF SOURCE LINES = ',I7/             01130001
      . ' NO. OF COMDECKS --- = ',I7/            01140000
      . ' NO. OF DECKS      = ',I7)              01150000
      01160000
END

```

```

C ..... SUBROUTINE SORT ( INF, DAT, QMAXNM, IA, IC, ENDDEC, NP )          01170002
C IMPLICIT INTEGER ( A-Z )                                              01180002
C CHARACTER DAT(1)=50,IA=50,IC=4,ENDDEC=8                                01190002
C REWIND INF                                                       01200002
C NP = 0                                                               01210002
100 CONTINUE
  READ ( INF, '(A)', END=120 )  IA
  IF ( IA(10:13) .EQ. IC )
    THEN
      NP = NP + 1
      IF ( NP .GT. QMAXNM )
        THEN
          WRITE ( 6,9910)
          STOP 16
        ENDIF
      DAT(NP) = IA
    ELSE
      IF ( IA(01:08) .EQ. ENDDEC ) GO TO 120
    ENDIF
    GO TO 100
120 CONTINUE
  REWIND INF
C----- SORT -----
  NPM1 = NP - 1
  IF (NPM1 .GT. 1)
    THEN
      ISP = 1
1100  CONTINUE
      CHANGE = 0
      ISPN = NPM1
      DO 1500 I=ISP,NPM1
        J = I + 1
        IF ( DAT(J)(14:23) .LT. DAT(I)(14:23) )
          THEN
            CHANGE I-TH & J-TH DATA. (J-TH DATA IS LESS THAN I-TH.)
            IF (CHANGE .EQ. 0) ISPN = I
            CHANGE = 1
            IA = DAT(J)
            DAT(J) = DAT(I)
            DAT(I) = IA
          ENDIF
        C
        1500  CONTINUE
        ISP = MAX0(1,ISPN-1)
        CHECK END OF SORTTING.
        C       CHANGE = 1.....CHANGED DATA.
        C       0.....NOT CHANGED ALL DATA.
        C       IF (CHANGE .EQ. 1) GO TO 1100
      ENDIF
      RETURN
9910 FORMAT (' ***** INSUFFICIENT SPACE *****')
END

```

(5) DIRECTIV

```

C GENERATE HISTRIAN DIRECTIVE FOR FACOM CONVERSION.          00010008
C IMPLICIT INTEGER (A-Z)                                         00020008
C CHARACTER*80 IA                                                 00030008
C CHARACTER*8 ICS,ICE,IB,DECKNM                                00040008
C CHARACTER*4 IC ,DKD,CDD                                       00050000
C EQUIVALENCE ( IA,IB,IC )                                     00060000
C CHARACTER*4 IDEL                                             00070003
C CHARACTER*16 IDENT(2)                                         00080003
C!!!!!! PLEASE CHANGE HISTRIAN UPDATE IDENT NAME.           00090000
C     IDENT NAME IS JYYMMIVN. (J YY MM A VN)
C             J : JAERI VERSION                               00100000
C             YY : YEAR                                    00110000
C             MM : MONTH                                   00120005
C             S : COMDECK(I)/SOURCE(S)                      00130007
C             VN : VERSION NO. (N)                         00140007
C (1) FOR COMDECK   JYYMMIVN                            00150007
C (2) FOR DECK     JYYMMSVN                           00160007
C     DATA IDENT(1) /'*IDENT -J8706IVO' /               00170007
C     DATA IDENT(2) /'*IDENT J8706SVO' /               00180007
C

```

```

C          00230005
DATA ICS /*=CONVSTR*/
DATA ICE /*=CONVEND*/
DATA IDEL/*CDEL*/ /
DATA DKD /*=OK */ /
DATA CDD /*=CD */ /
C          00240005
C          00250000
C          00260000
C          00270003
C          00280003
C          00290000
C          00300007
LOGICAL SOCREQ
DATA SOCREQ /.FALSE./
INP    = 1
OUT   = 3
IF ( SOCREQ ) OUTS   = 9
LP     = 6
C          00310007
C          00320003
C          00330005
C          00340007
C          00350003
C          00360000
C          00370000
INREC = 0
INDEL = 0
OTREC = 0
OTDIR = 0
C          00380000
C          00390000
C          00400000
C          00410000
C          00420005
50 CONTINUE
WRITE (OUT,'(A)') IDENT(INP)
OTDIR = OTDIR + 1
100 CONTINUE
READ (INP,'(A)',END=1000) IA
INREC = INREC + 1
IF ( IC .EQ. DKD )
THEN
DECKNM= IA(05:12)
CALL LINES
WRITE (LP,6010) INREC,IA,DECKNM
OTREC = OTREC + 1
IF ( SOCREQ ) WRITE (OUTS,'(A)') IA
ELSE
.IF ( IC .EQ. CDD )
THEN
DECKNM= IA(05:12)
CALL LINES
WRITE (LP,6010) INREC,IA,DECKNM
OTREC = OTREC + 1
IF ( SOCREQ ) WRITE (OUTS,'(A)') IA
ELSE
.IF ( IC .EQ. IDEL )
THEN
INDEL = INDEL + 1
CALL LINES
WRITE (LP,6010) INDEL,IA,DECKNM
OTREC = OTREC + 1
IF ( SOCREQ ) WRITE (OUTS,'(A)') IA
ELSE
.IF ( IB .EQ. ICS )
THEN
OTDIR = OTDIR + 1
WRITE (OUT,'(A)') IA(10:72)
CALL LINES
WRITE (LP,6010) OTDIR,IA(10:72)//'
200 CONTINUE
READ (INP,'(A)',END=9000) IA
INREC = INREC + 1
IF ( IB .NE. ICE )
THEN
OTDIR = OTDIR + 1
WRITE (OUT,'(A)') IA
CALL LINES
WRITE (LP,6010) OTDIR,IA,DECKNM
GO TO 200
ENDIF
ELSE
OTREC = OTREC + 1
IF ( SOCREQ ) WRITE (OUTS,'(A)') IA
ENDIF
GO TO 100
C          00710003
C          00720000
C          00730000
C          00740001
C          00750000
C          00760003
C          00770003
C          00780000
C          00790000
C          00800000
C          00810000
C          00820000
C          00830001
C          00840000
C          00850003
C          00860003
C          00870000
C          00880000
C          00890001
C          00900001
C          00910007
C          00920000
C          00930000
C          00940000
C          00950000
1000 CONTINUE
REWIND INP
REWIND OUT
INP = INP + 1
OUT = OUT + 1
IF ( INP .LE. 2 )
THEN
INCINR = INREC
INCOTR = OTREC
INCOTD = OTDIR
INCDEL = INDEL
GO TO 50
ENDIF
C          00960005
C          00970005
C          00980005
C          00990005
C          01000005
C          01010005
C          01020005
C          01030005
C          01040005
C          01050005
C          01060005
C          01070005

```

```

SOCINR = INREC - INCINR          01080005
SOCOTR = OTREC - INCOTR          01090005
SOCOTD = OTDIR - INCOTD          01100005
SOCDEL = INDEL - INCDEL          01110005
WRITE (LP,6040)                  01120003
WRITE (LP,6020)  ' TOTAL  ',INREC ,OTREC ,OTDIR ,INDEL   01130005
WRITE (LP,6020)  ' INCLUDE  ',INCINR,INCOTR,INCOTD,INCDEL 01140005
WRITE (LP,6020)  ' SOURCE   ',SOCINR,SOCOTR,SOCOTD,SOCDEL 01150005
STOP                             01160000
9000 CONTINUE
WRITE (LP,6040)
WRITE (LP,6030)
STOP 16                           01170000
C
6010 FORMAT (1X,17,2X,A,'      ('',A,,')')
6020 FORMAT (///' ****=  

. ' =                                     ''/  

. ' =                                     ''/  

. ' =                                     ''/  

. ' = NO. OF INPUT SOURCE LINES = ',I7, ' ''/  

. ' = NO. OF OUTPUT SOURCE LINES = ',I7, ' ''/  

. ' = NO. OF OUTPUT DIRECTIVE = ',I7, ' ''/  

. ' = NO. OF DELETED LINES   = ',I7, ' ''/  

. ' =                                     ''/  

. ' =                                     ''/  

. ' =                                     ''/  

6030 FORMAT (//' ***** DATA ERROR, READ END OF FILE. *****')
6040 FORMAT (1H1)
END
SUBROUTINE LINES
IMPLICIT INTEGER (A-Z)
C
DATA LINMAX / 50 /
DATA PAGE / 0 /
DATA LINE / 0 /
C
LINE = LINE + 1
IF ( MOD(LINE,LINMAX) .EQ. 1 )
THEN
  IF ( PAGE .NE. 0 )
  THEN
    WRITE (6,6010) PAGE
  ENDIF
  PAGE = PAGE + 1
  WRITE (6,6020)
  WRITE (6,6010) PAGE
ENDIF
RETURN
6010 FORMAT (/' 1X,' SEQ. ....1....2....3....4',  

. ' ,....5....6....7....8',  

. ' PAGE ',13/)
6020 FORMAT (1H1)
END

```

(6) UPDCHG

```

C
C     OUTCD = OUTDK = 2 ..... COMDECK AND DECK ARE SAME FILE. 00010001
C     OUTCD = 2 , OUTDK = 3 ..... COMDECK AND DECK ARE EACH FILES. 00020001
C     IMPLICIT INTEGER ( A-Z ) 00030001
CHARACTER#72 IA 00040001
CHARACTER IB=4,IC=4, NAME=10, ID=4,CDDK=4 00050000
CHARACTER SID=4 00060000
EQUIVALENCE ( IA(1:1) , IB (1:1) ) 00070000
EQUIVALENCE ( IA(5:5) , NAME(1:1) ) 00080000
DATA IC, ID / 'CD' , 'DK' / 00090000
INS = 0 00100000
OUT = 0 00110000
LP = 6 00120000
INP = 1 00130000
OUTCD = 2 00140000
C     OUTDK = 3 00150000
OUTDK = 2 00160001
OUTCDL = 0 00170001
OUTDKL = 0 00180000
CSEQ = 0 00190000
00200000

```

```

DSEQ = 0          00210000
ONESUB = 0        00220000
100 CONTINUE      00230000
  READ (INP,'(A)',END=900) IA
  INS = INS + 1    00240000
  IF ( IB .EQ. IC )
    THEN
      CSEQ = CSEQ + 1  00250000
      CDDK = NAME      00260000
      OUT = OUTCD      00270000
      SID = IC         00280000
      ONESUB = 0        00290000
    ELSE
    .IF ( IB .EQ. ID )
    .THEN
      DSEQ = DSEQ + 1  00300000
      CDDK = NAME      00310000
      OUT = OUTDK      00320000
      SID = ID         00330000
      ONESUB = 0        00340000
    ENDIF
    ONESUB = ONESUB + 1  00350000
    IF ( SID .EQ. IC )
    .THEN
      OUTCDL = OUTCDL + 1  00360000
    ELSE
      OUTDKL = OUTDKL + 1  00370000
    ENDIF
    WRITE (OUT,6020) IA,CDDK,ONESUB
    GO TO 100
900 CONTINUE      00380000
  WRITE (LP,2020) INS,CSEQ,OUTCDL,DSEQ,OUTDKL
  STOP
6020 FORMAT (A,A,14)
2020 FORMAT (//,
  ' NO. OF SOURCE LINES   = ',I7/
  ' NO. OF COMDECKS      = ',I7/
  ' NO. OF COMDECK LINES = ',I7/
  ' NO. OF DECKS         = ',I7/
  ' NO. OF DECK LINES    = ',I7)
END

```

(7) UPDLIST

```

C      PRINT UPDATE(HISTRIAN) DIRECTIVE LIST WITHOUT SOURCE LINE. 00010002
C
C      IMPLICIT INTEGER (A-Z) 00020002
CHARACTER*80 CARD 00030002
C
INP = 1            00040002
OUT = 2            00050000
CR = 5             00060000
LP = 6             00070000
INPL = 0            00080000
IND = 0            00090000
300 CONTINUE      00100001
  READ (INP,'(A)',END=600) CARD
  INPL = INPL + 1    00110001
  IF ( CARD(1:1) .EQ. '*' )
    THEN
      IF (CARD(2:2) .EQ. 'I' .OR. CARD(2:2) .EQ. 'D' ) 00120000
      THEN
        IF ( CARD(2:8) .NE. 'INCLUDE' ) 00130000
        THEN
          IND = IND + 1 00140000
          WRITE (OUT,6010) IND,CARD
        ENDIF
      ENDIF
    ENDIF
  GO TO 300
600 CONTINUE      00150001
  WRITE (LP,6020) INPL,IND
  STOP
6010 FORMAT (1X,I5,2X,A) 00160000
6020 FORMAT (//,
  ' NO. OF INPUT LINE(S) = ',I6/ 00170001
  ' NO. OF DIRECTIVE     = ',I6) 00180001
END

```

表A.1 変換作業用プログラムツール・メンバ名一覧

DIRECTORY LIST OF J3076.TK.UTY.FORT77			
*****	*****	*****	*****
MEMBER NAME	PAGE NO.	NO. OF CARDS	*****
*****	*****	*****	*****
(NO.=001) CHGSYM	0001	105	
(NO.=002) COMPDECK	0002	171	
(NO.=003) COMPNEW	0006	168	
(NO.=004) COMPSOC	0009	169	
(NO.=005) DIRECTIV	0012	162	
(NO.=006) UPDCHG	0015	61	
(NO.=007) UPDLIST	0016	34	

		870 CARDS	

付録B MELPROG 変換ツール用コマンドプロシジャー

B. 1 コマンドプロシジャメンバ名とその処理

表 B.1 にコマンドプロシジャのメンバ名とそのライン数をあげる。また、以下に各コマンドプロシジャの解説を記す。

①CHGSYM

このコマンドプロシジャは新しく MELPLOGのアップデイト・ディレクティブを入手したとき 6 文字を超えるシンボル名でかつ、現バージョンで変更したシンボル名と等しいものを変更するプログラムを実行するものである。

②COMPSOC

MELPLOG ソースプログラムは UPDATE によりかかれており、そのリスト出力を行う COMPSOC プログラムを実行するためのコマンドプロシジャである。
(COMPNEW ; CMPDECK も同じく利用)

③DIRECTIV

変換・整備用ソースプログラムから HISTORIAN 用ディレクティブを作成するツール DIRECTIV を実行するためのコマンドプロシジャである。

④UPDCHG

オリジナルのソース・プログラムの73カラムから80カラムに CARD-ID を付加するツール UPDCHG を実行するためのコマンドプロシジャである。

⑤UPDLIST

アップデイト・ディレクティブからディレクティブラインを出力するツール UPDLIST を実行するコマンドプロシジャである。

以上コマンドプロシジャーの利用方法は、必要なソース・プログラムを UTY.FORT 77 のメンバーから PFD の 2 の EDIT でこのコマンドプロシジャーを実行して必要なファイルをアロケートのみ行い RUN コマンドにより実行する。

B. 2 コマンドプロシジャ・リスト

(1) CHGSYM

```

CHGSYM 00010001 PROC 1 MEM +
CHGSYM 00020001      UPD('J3076.TK.MELPNEY.DATA') +
CHGSYM 00030001      NAM('J3076.TK.MELPROG.DATA(CONVSYMB)')
CHGSYM 00040000 CONTROL LIST MSG
CHGSYM 00050003 FREE ATTR(SOC INP LP)
CHGSYM 00060001 ATTR SOC LRECL(80) BLKSIZE(24640) RECFM(F B) DSORG(PS)
CHGSYM 00070004 ATTR LP LRECL(137) BLKSIZE(23427) RECFM(F B) DSORG(PS)
CHGSYM 00080000 ATTR INP INPUT
CHGSYM 00090001 DEL 00.&MEM..DATA
CHGSYM 00100001 ALLOC DA(00.&MEM..DATA) NEW F(FT02F001) T SP(1 1) RELEASE +
                  USING(SOC) UNIT(TSSWK) REUSE
CHGSYM 00110001
CHGSYM 00120003 DEL 00.&MEM..LIST
CHGSYM 00130003 ALLOC DA(00.&MEM..LIST) NEW F(FT10F001) T SP(1 1) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
CHGSYM 00140005 ATTR DA('&UPD(&MEM)') SHR F(FT01F001) REUSE US(INP)
CHGSYM 00150002 ALLOC DA('&NAM') SHR F(FT03F001) REUSE US(INP)
CHGSYM 00160002
CHGSYM 00170003 FREE ATTR(SOC INP LP)

```

(2) COMPSOC

```

COMPSOC 00010004 PROC 0 SOC('J3076.00MELPPS.FORT77') +
COMPSOC 00020004      INC('J3076.00MELPPS.INCLUDE')
COMPSOC 00030000 CONTROL LIST MSG
COMPSOC 00040005 FREE ATTR(LP F01 INP)
COMPSOC 00050000 ATTR LP LRECL(137) BLKSIZE(23427) RECFM(F B) DSORG(PS)
COMPSOC 00060000 ATTR INP INPUT
COMPSOC 00070006 DEL 00.LP1.DATA 00.LP2.DATA 00.LP3.DATA 00.LP4.DATA 00.LP5.DATA
COMPSOC 00080006 DEL 00.F2.DATA
COMPSOC 00090006 DEL 00.F3.DATA
COMPSOC 00100006 ALLOC DA(00.LP1.DATA) NEW F(FT11F001) T SP(10 10) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
COMPSOC 00110000 ALLOC DA(00.LP2.DATA) NEW F(FT12F001) T SP(10 10) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
COMPSOC 00120006 ALLOC DA(00.LP3.DATA) NEW F(FT13F001) T SP(10 10) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
COMPSOC 00130001 ALLOC DA(00.LP4.DATA) NEW F(FT14F001) T SP(10 10) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
COMPSOC 00140006 ALLOC DA(00.LP5.DATA) NEW F(FT15F001) T SP(10 10) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
COMPSOC 00150001 ALLOC DA(00.F2.DATA) NEW F(FT02F001) T SP(2 2) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
COMPSOC 00160006 ALLOC DA(00.F3.DATA) NEW F(FT03F001) T SP(2 2) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
COMPSOC 00170001 ALLOC DA('&INC' '&SOC') F(FT01F001) SHR REUSE US(INP)
COMPSOC 00180006
COMPSOC 00190001
COMPSOC 00200006
COMPSOC 00210000
COMPSOC 00220006
COMPSOC 00230001
COMPSOC 00240004

```

(3) DIRECTIV

```

DIRECTIV 00010000 PROC 0 SOC('J3076.00MELPPS.FORT77') +
DIRECTIV 00020002      INC('J3076.00MELPPS.INCLUDE') +
DIRECTIV 00030002      OUTSOC('J3076.TK.MELPROG.DIRECTIV') +
DIRECTIV 00040002      OUTINC('J3076.TK.MELPINC.DIRECTIV')
DIRECTIV 00050000 CONTROL LIST MSG
DIRECTIV 00060002 FREE ATTR(LP SOC INP)
DIRECTIV 00070000 ATTR LP LRECL(137) BLKSIZE(23427) RECFM(F B) DSORG(PS)
DIRECTIV 00080002 ATTR SOC LRECL(80) BLKSIZE(24640) RECFM(F B) DSORG(PS)
DIRECTIV 00090000 ATTR INP INPUT
DIRECTIV 00100004 DEL 00.LP.DATA
DIRECTIV 00110004 DEL 00.F9.DATA
DIRECTIV 00120004 ALLOC DA(00.LP.DATA) NEW F(FT06F001) T SP(10 10) RELEASE +
                  USING(LP) UNIT(TSSWK) REUSE
DIRECTIV 00130000 ALLOC DA(00.F9.DATA) NEW F(FT09F001) T SP(100 50) RELEASE +
                  USING(SOC) UNIT(TSSWK) REUSE
DIRECTIV 00140004 ALLOC DA('&INC') SHR F(FT01F001) REUSE US(INP)
DIRECTIV 00150002 ALLOC DA('&SOC') SHR F(FT02F001) REUSE US(INP)
DIRECTIV 00160003 ALLOC DA('&OUTINC') SHR F(FT03F001) REUSE USING(SOC)
DIRECTIV 00180003 ALLOC DA('&OUTSOC') SHR F(FT04F001) REUSE USING(SOC)
DIRECTIV 00190003
DIRECTIV 00200002 FREE ATTR(LP SOC INP)

```

(4) UPDCHG

```

UPDCHG 00020005 PROC 0 SOC('J3076.MELSRC.FORT')
UPDCHG 00030000 CONTROL LIST MSG
UPDCHG 00040003 FREE ATTR(F02 INP)
UPDCHG 00050000 ATTR F02 LRECL(80) BLKSIZE(24640) RECFM(F B) DSORG(PS)
UPDCHG 00060000 ATTR INP INPUT
UPDCHG 00070006 DEL  @@.F2.INCLUDE @@.F3.FORT77
UPDCHG 00080006 ALLOC DA(@@.F2.INCLUDE) NEW F(FT02F001) T SP(30 30) RELEASE +
UPDCHG 00090002 USING(F02) UNIT(TSSWK) REUSE
UPDCHG 00100006 ALLOC DA(@@.F3.FORT77) NEW F(FT03F001) T SP(30 30) RELEASE +
UPDCHG 00110002 USING(F02) UNIT(TSSWK) REUSE
UPDCHG 00120000 ALLOC DA('&SOC') F(FT01F001) SHR REUSE US(INP)

```

(5) UPDLIST

```

UPDLIST 00010001 PROC 1 MEM ORG +
    UPD('J3076.TK.MELPNEW.DATA') +
    NEW('J3076.TK.MELPROG.DIRECTIV')
UPDLIST 00020001 CONTROL LIST MSG
UPDLIST 00030001 FREE ATTR(LP INP)
UPDLIST 00040001 ATTR LP LRECL(137) BLKSIZE(23427) RECFM(F B) DSORG(PS)
UPDLIST 00050001 ATTR INP INPUT
UPDLIST 00060005 DEL  @@.&MEM..LIST
UPDLIST 00070001 ALLOC DA(@@.&MEM..LIST) NEW F(FT02F001) T SP(1 1) RELEASE +
UPDLIST 00080001 USING(LP) UNIT(TSSWK) REUSE
UPDLIST 00090001 IF &ORG NE &STR(ORG) THEN +
    ALLOC DA('&NEW')      SHR F(FT01F001) REUSE US(INP)
UPDLIST 00100001 ELSE +
UPDLIST 00110003 ALLOC DA('&UPD(&MEM)') SHR F(FT01F001) REUSE US(INP)
UPDLIST 00120001 FREE ATTR(LP INP)

```

表B.1 変換作業用コマンドプロシジャー名一覧

***** MEMBER NAME *****	***** PAGE NO.. *****	***** NO. OF CARDS *****
(#NO 001) CHGSYM	0001	17
(#NO 002) COMPSOC	0001	24
(#NO 003) DIRECTIV	0002	20
(#NO 004) UPDCHG	0002	11
(#NO 005) UPDLIST	0002	15

		87 CARDS

付録 C MELPROG 計算結果の図形処理機能の整備

C. 1 図形処理機能の整備

計算結果の図形処理機能について、MELPROG 自身は開発元(SNL)の図形処理ツールの利用を前提に外部にファイルを出力しているが、原研では、原研が開発した図形処理用プログラム SPL-PLOT^(C-1) もしくは ROPS^(C-2)を利用して行うこととした。このためには、図形処理用出力データを、原研特有の図形処理用標準データ形式 (SPL 形式)⁽⁵⁾ としなければならない。図形処理用の物理量は時系列データでプリント出力しているものをすべて対象とする。プリント出力の物理量は、MELPROG では大きく 4 つの部分に分かれている。それは、FLUIDS 部、CORE 部、STRUCTURE 部、および DEBRIS 部である。それらの出力副プログラムはそれぞれ FLPRNT, OUTCOR, SOUT, OUTPUT である。

(1) SPL 形式用データ処理副プログラム名とその処理

以下に SPL 形式用にデータ処理を行うための副プログラムを記す。

- | | |
|-----------|--------------------------------|
| ① SPLINIT | : SPL 形式の初期処理 |
| SPLLPT | : SPL 形式の出力変数 (ラベル) 出力処理 (副入口) |
| SPLDPT | : SPL 形式のデータ出力処理 (") |
| SPLERR | : SPL 形式 処理のエラー処理 (") |
| ② SPLFLD | : FLUIDS 部のラベル設定とデータ設定と登録 |
| ③ SPLCOR | : CORE 部のラベル設定とデータ設定と登録 |
| ④ SPLSTR | : STRUCTURE 部のラベル設定とデータ設定と登録 |
| ⑤ SPLDBL | : DEBRIS 部のラベル設定 |
| ⑥ SPLDBD | : DEBRIS 部のダミーデータ設定 (未計算のため) |
| ⑦ SPLDBS | : DEBRIS 部の計算データ設定 |
| ⑧ SPLDBP | : DEBRIS 部のデータ部の SPL への登録 |
| ⑨ SPLCLR | : SPL 作業領域のゼロクリア |

(2) SPL-PLOT および ROPS のための図形データ設定方法

MELPROG では各部の計算とプリント出力は各部、別々に計算され、その都度出力している。SPL-PLOT, ROPS ではすべてのラベル情報を設定後、図形データを設定しなければならない。このため MELPROG では最初の図形データ出力時には各部ではレベル情報のみの設定を行い図形データは設定していない。2 回目の図形データの出力時に、はじめてデータ部を出力している。図 C.1 にその処理フローを示す。

(3) SPL のための共通ブロック (Include) の内容

図 C.2 にインクルードメンバ ¥SPLCM の内容を示す。

(4) SPL のための変数

表 C.1 から表 C.4 に SPL の変数、その次元および次元変数と、MELPROG プログラム上の変数の対応を示す。

C. 2 図形処理機能の使用法

図形処理は、SPL-PLOT もしくは ROPS を用いて行うため、その使用方法の詳細は参考文献 (C-1), (C-2) を参照されたい。ここでは SPL-PLOT の使用例として、図 C.3 にその JCL を示す。また、MELPROG のデータを利用して SPL-PLOT のロードモジュールを実行させるには領域が不足するため、参考文献(C-2)で指示されているように、図 C.4 に示す主プログラムにより作成されるロードモジュールを利用しなければならない。このロードモジュール作成には以下に示すコマンドを TSS 処理にて実施すれば良い。

（ロードモジュール再作成コマンド）

➡ : ENTER の意味

```

READY
FORT77 ソースファイル名 OBJ(@@)➡
-- -
-- -
READY
ALLOC DA('J3080.¥PLTNLP4.LOAD') SHR F(OLDM) REUSE➡
LINK (@@ *) LO(TK.MELPROG) LET➡
-- -
-- -
INCLUDE OLDM(TEMPNAME)➡
ENTRY MAIN➡
NAME PLTNLP4(R)➡
➡
-- -
-- -
READY
FREE F(OLDM)➡

```

図 C.5 から 図 C.8 に追加した図形処理機能により得られたおもなプロット図を示す。

C. 3 図形処理対象物理量の使用作業領域の問題

今回の変換・整備では、図形処理用に出力する物理量（変数／配列）としては4つのモジュールでのプリント出力対象物理量全てを対象とした。このため1回の物理時刻の出力データ量が原子炉体系のデータの場合に非常に大きくなる。1回当たり0.5トラックから2トラックとなる場合がある。したがって、500点位のデータに対して250～1000トラックの容量が必要である。表 C.1 から表 C.4 の SPL 変数を吟味し、必要最小限におさめた方が良い。不要な変数が存在する場合には、変換・整備用のディレクティブを修正すれば良い。このとき、その修正方法としては不要なラインを取り除くだけでなく、コメントアウトとしてそのラインを保存した方がよい。このことによりそのディレクティブのライン数が変わらず、他のディレクティブに影響を与えることになる。変更箇所は各モジュールの各変数のラベル登録部（1回のみの実行部）と各変数のデータ登録部の2箇所である。ただし DEBRIS モジュールはダミーデータの登録部が存在するため、各々2箇所づつ存在する。

参考文献

- (C-1) 村松 健 他, "過渡現象の実験結果及び計算結果の編集・作図用標準プログラム
パッケージ SPLPACK-I の使用手引," JAERI-M 83-166 (1983)
- (C-2) 安濃田良成, 他 "会話型汎用グラフィック・プロセッサー ROPS 2.0 - ユーザーズ
・マニュアル," JAERI-M 88-030 (1988)

表C.1 FLUIDS部の図形処理用データ (1/2)

	SPL 変数	プログラム変数	次元			備 考
			1次元	2次元	3次元	
1	F-T	T	MF	MZ	MR	Temperature
2	ALPHA	ALPHA	MF	MZ	MR	Volume Fraction
3	VZ	VZ	MF	MZ	MR	Axial velocity
4	VR	VR	MF	MZ	MR	Radial velocity
5	ALRH	ALRH	MF1	MZ	MR	ALPHA*RHO
6	ALRHU	ALRHU	MF	MZ	MR	ALPHA*RHO*U
7	RH	RH	MF1	MZ	MR	Density
8	P	P	MZ	MR	—	Pressure
9	XPH2	XPH2	MZ	MR	—	PH2/PTOT
10	TSAT	TSAT	MZ	MR	—	Saturation Temp.
11	F-GAM21	F-GAM21	MZ	MR	—	GAMMA 2-1
12	DCORIM	DCORIM	MZ	MR	—	CORIUM diameter
13	GAM83	GAM83	MZ	MR	—	GAMMA 8-3
14	HTC12V		MZ	MR	—	HTRC 1-2 Vapor
15	HTC12L		MZ	MR	—	HTRC 1-2 Liquid
16	HTC13		MZ	MR	—	HTRC 1-3
17	HTC14		MZ	MR	—	HTRC 1-4
18	HTC23		MZ	MR	—	HTRC 2-3
19	HTC24		MZ	MR	—	HTRC 2-4
20	HTC34		MZ	MR	—	HTRC 3-4
21	F-HTC18		MZ	MR	—	HTRC 1-8
22	F-HTC19		MZ	MR	—	HTRC 1-9
23	HTC28		MZ	MR	—	HTRC 2-8
24	HTC29		MZ	MR	—	HTRC 2-9
25	HTC38		MZ	MR	—	HTRC 3-8
26	HTC39		MZ	MR	—	HTRC 3-9
27	HTC48		MZ	MR	—	HTRC 4-8
28	HTC49		MZ	MR	—	HTRC 4-9
29	Q1I		MZ	MR	—	Heat Transfer 1-I
30	Q2I		MZ	MR	—	Heat Transfer 2-I

表C.1 FLUIDS部の図形処理用データ (2/2)

	SPL 変数	プログラム変数	次 元			備 考
			1 次元	2 次元	3 次元	
31	F-Q13		MZ	MR	-	Heat Transfer 1-3
32	Q14		MZ	MR	-	Heat Transfer 1-4
33	Q23		MZ	MR	-	Heat Transfer 2-3
34	Q24		MZ	MR	-	Heat Transfer 2-4
35	Q34		MZ	MR	-	Heat Transfer 3-4
36	Q18		MZ	MR	-	Heat Transfer 1-8
37	Q28		MZ	MR	-	Heat Transfer 2-8
38	Q38		MZ	MR	-	Heat Transfer 3-8
39	Q48		MZ	MR	-	Heat Transfer 4-8
40	AREA12		MZ	MR	-	Interfacial area 1-2
41	F-AREA13		MZ	MR	-	Interfacial area 1-3
42	AREA14		MZ	MR	-	Interfacial area 1-4
43	AREA23		MZ	MR	-	Interfacial area 2-3
44	AREA24		MZ	MR	-	Interfacial area 2-4
45	F-AREA34		MZ	MR	-	Interfacial area 3-4
46	ALFSTR		MZ	MR	-	Structure Vol. fraction
47	FAZ		MZ	MR	-	Axial flow area
48	FAR		MZ	MR	-	Radial flow area
49	VOL		MZ	MR	-	Volume
50	DZ		MZ	MR	-	Delta Z
51	F-DR		MZ	MR	-	Delta R
52	CP		MF	MR	-	
53	K		MF	MR	-	
54	MU		MF	MR	-	
55	U		MF	MR	-	
56	MASS		NCOMP	NZZ	MR	Mass
57	RHO		NCOMP	NZZ	MR	Density
58	TOTMAS		NZZ	MR	-	Total mass
59	TOTRHO		NZZ	MR	-	Total density
60						

表C.2 CORE部の図形処理用出力データ

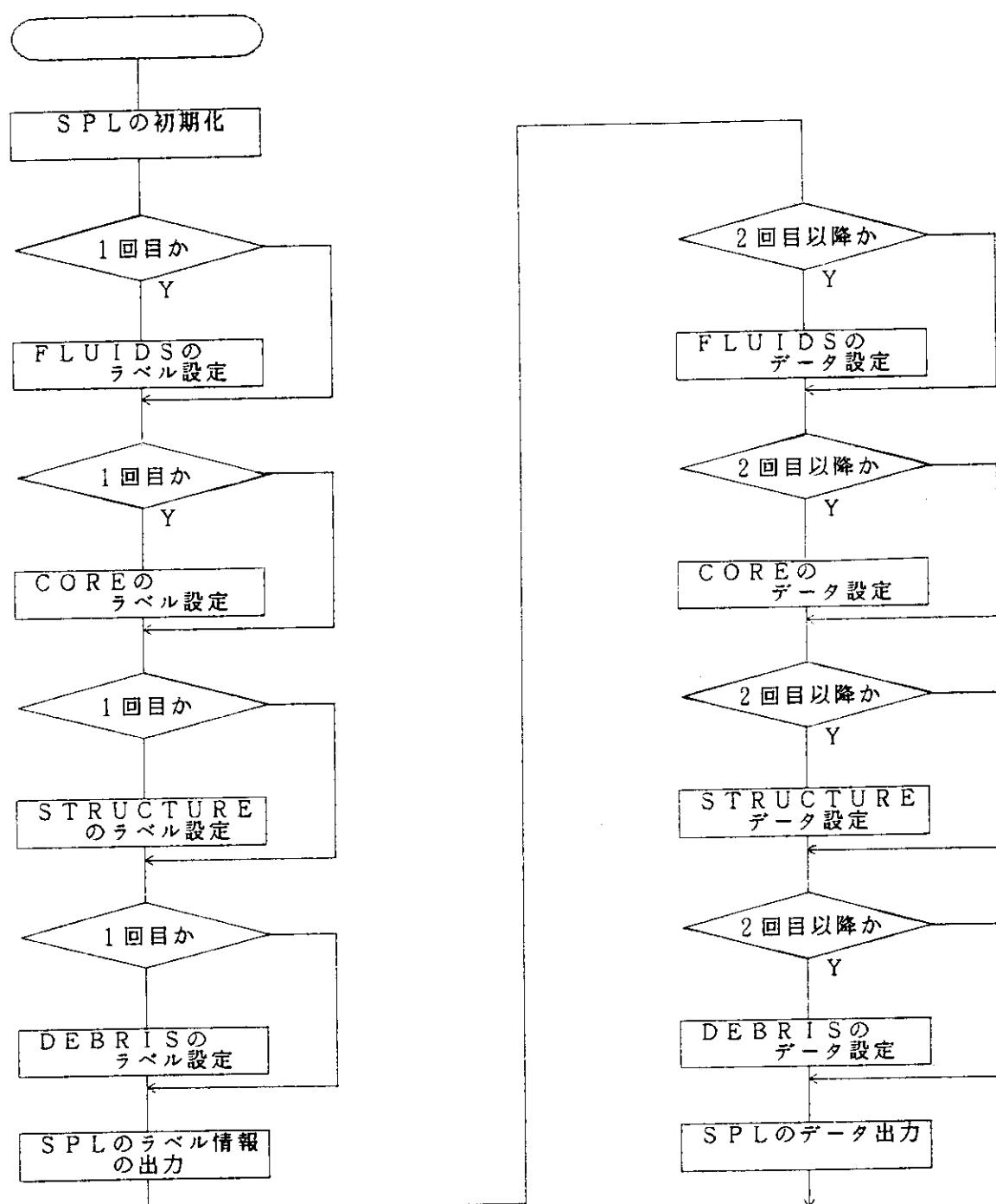
	SPL 変数	プログラム変数	次元			備考
			1次元	2次元	3次元	
1	C-H2CORE		-	-	-	Total hydrogen produced
2	TOTMSF		-	-	-	Total mass transfer(fuel)
3	TOTMSC		-	-	-	Tot. mass trans. (control rod)
4	TEMP	RNODE	ZNODE	NRING		Temperature
5	SUFT	ZNODE	NRING	-		Surface temperature
6	H2MASS	NRING	KMAX	-		Cell hydrogen produced
7	CNMREP	NRING	MXCSTR	-		
8	CPWFAC	NRING	MXCSTR	-		Power factor
9	TPRESS	NRING	MXCSTR	-		Internal Pressure
10	RINNER	RINNER	LOC5	-	-	Inner radius
11	C-ROUTER	ROUTER	LOC5	-	-	Outer radius
12	TINNER	TINNER	LOC5	-	-	Inner temperature
13	TOUTER	TOUTER	LOC5	-	-	Outer temperature
14	TOTMAS	TOTMAS	LOC5	-	-	Total mass
15	W01	W01	LOC5	-	-	Weight % U or AIC
16	W02	W02	LOC5	-	-	Weight % ZR
17	W03	W03	LOC5	-	-	Weight % O
18	W04	W04	LOC5	-	-	Weight % STEEL
19	RADI	RNODE	ZNODE	NRING		Radii of node
20	MASSES	RNODE	ZNODE	NRING		Masses of node

表C.3 STRUCTURE部の図形処理用出力データ

	SPL 変数	プログラム変数	次元			備考
			1次元	2次元	3次元	
1	S-T		LOS1	-	-	Temperature
2	LOADT		LOS2	-	-	Load temperature
3	XLIFE	XLIFE	LOS3	-	-	Life fraction
4	XLOAD	XLOAD	LOS3	-	-	X-Load temperature
5	STRESS	STRESS	LOS3	-	-	Stress
6	TEMP	TEMP	LOS3	-	-	Average temperature
7	TIN	TIN	LOS3	-	-	Inner temperature
8	TOUT	TOUT	LOS3	-	-	Outer temperature
9	DFLAG		KMAX	NRING	-	Debris flag

表C.4 DEBRIS部の図形処理用データ

	SPL 変数	アロケラム変数	次 元			備 考
			1 次元	2 次元	3 次元	
1	D-QTOT		NBED	—	—	Total power
2	ETOTAL		NBED	—	—	Total energy
3	TOTMSS		NBED	—	—	Total mass
4	AMOLTN		NBED	—	—	Molten mass
5	LIOMAS		NBED	—	—	UO2 mass
6	ZRMASS		NBED	—	—	ZR mass
7	ZUMASS		NBED	—	—	ZR-U mass
8	ZOMASS		NBED	—	—	ZRO2 mass
9	SSMASS		NBED	—	—	STEEL mass
10	CRMASS		NBED	—	—	CR mass
11	D-HEIG		NBED	—	—	BED height
12	POR		NBED	—	—	Average Porosity
13	TAVE		NBED	—	—	Average Temperature
14	TBOT		NBED	—	—	Bottom Temperature
15	TFLX		NBED	—	—	TFLUX
16	TTOP		NBED	—	—	Top Temperature
17	FLUX		NBED	—	—	Mass flux
18	DZ		NZMAX	NBED	—	Height
19	T		NZMAX	NBED	—	Temperature
20	EPS		NZMAX	NBED	—	EPS
21	D-VFU02		NZMAX	NBED	—	Fraction UO2
22			NZMAX	NBED	—	Fraction UO2
23			NZMAX	NBED	—	Fraction ZR
24			NZMAX	NBED	—	Fraction ZR
25			NZMAX	NBED	—	Fraction ZR-U
26			NZMAX	NBED	—	Fraction ZR-U
27			NZMAX	NBED	—	Fraction ZRO2
28			NZMAX	NBED	—	Fraction ZRO2
29			NZMAX	NBED	—	Fraction STEEL
30			NZMAX	NBED	—	Fraction STEEL
31			NZMAX	NBED	—	Fraction CR
32			NZMAX	NBED	—	Fraction CR
33			NZMAX	NBED	—	Fraction H2O
34			NZMAX	NBED	—	Area
35			NZMAX	NBED	—	Energy



(第1回目の図形データ出力時)

(第2回目以降の図形データ出力時)

各部処理前に各部の計算部分がある

図C.1 SPLデータ出力のためのMELPROGの図形データ処理フロー

```

*CONVSTR *AF          00010011
*CD *SPLCM           00020012
C   COMMON AREA FOR SPL-PLOT.          00030011
    INTEGER QSPLSZ,QSPLWK             00040001
    PARAMETER ( QSPLSZ = 50000 )       00050006
    PARAMETER ( QSPLWK = 10000 )       00060022
C
C   MEMORY = 206 + 80*NOVARB + MEMUSE 00070001
    COMMON / *WAREA / MEMORY,MEMAOJ,IA(QSPLSZ) 00080006
C
C   I*OPT ..... PRINT OPTION FOR SPL-INFORMATION(S). 00090001
C     = 0 ; NO PRINT ALL.            00100001
C     = 1 ; PRINT LABEL.          ( I*OPT1 = 1 ) 00110009
C     = 2 ; PRINT DUMP TIME.      ( I*OPT2 = 1 ) 00120009
C     = 4 ; PRINT LOCATION.      ( I*OPT4 = 1 ) 00130009
C
C   COMMON / *SPLWK / SPLWK (QSPLWK),LBLFLG,MEMUSE,NOVARB 00140009
    . ,I*DBG,I*OPT,I*OPT1,I*OPT2,I*OPT4 00150009
    . ,I*MF ,I*MZ ,I*MR ,I*IFL ,I*NZP1,I*NRPI 00160009
    . ,I*NF ,I*NCMP ,I*NZZ ,I*KMAX ,I*NRNG ,I*MXCS 00170010
    . ,I*NZMX ,I*NBED ,I*DEBL ,I*DEBF 00180021
    . ,I*DBSZ ,I*D1D(17),I*D2D(18) 00190021
    . ,N*AXRG,N*DTMX,N*DRMX,N*DMMX 00200021
    . ,I*LOC2,I*LOC5,I*LOC6,I*LOC7,I*LOC8 00210021
    . ,I*LOS1,I*LOS2,I*LOS3 00220021
    REAL SPLWK 00230021
C
C   SPLWK ! WORKING AREA FOR SAVE DATA. 00240021
C   LBLFLG ! PUT LABEL DATA FLAG. 00250021
C   MEMUSE ! USED MEMORY FOR SPLWK. 00260001
C   NOVARB ! NO. OF SPL VARIABLE(S). 00270001
C   I*OPT ! SPL PLOT INFORMATION PRINT FLAG(S) 00280019
C   I*OPT1 ! PRINT FLAG ( SPL LABEL INF.) 00290019
C   I*OPT2 ! PRINT FLAG ( SPL PUT TIME. ) 00300019
C   I*OPT4 ! PRINT FLAG ( SPL LOCATION(S) INF.) 00310019
C   I*MF ! MF 00320019
C   I*MZ ! MZ 00330019
C   I*MR ! MR 00340019
C   I*IFL ! IFL 00350019
C   I*NZP1 ! NZP1 00360019
C   I*NRPI ! NRP1 00370019
C   I*NF ! NF 00380019
C   I*NCMP ! NCMP 00390019
C   I*NZZ ! NZZ 00400019
C   I*KMAX ! KMAX 00410019
C   I*NRNG ! NRNG 00420019
C   I*MXCS ! MXCSTR 00430019
C   I*NZMX ! NZMAX 00440019
C   I*NBED ! NBED 00450019
C   I*DEBL ! LABEL PUT OR NOT FLAG FOR DEBRIS. 00460019
C   I*DEBF ! DUMMY DATA OR CALC. DATA SET FLAG. 00470019
C   I*DBSZ ! DEBRIS DATA AREA SIZE. 00480019
C   I*D1D ! DEBRIS ONE DIMENSIONAL AREA START LOCATION(S). 00490019
C   I*D2D ! DEBRIS TWO DIMENSIONAL AREA START LOCATION(S). 00500019
C   I*LOC2 ! LOCATION CORE 00510019
C   I*LOC5 ! LOCATION CORE 00520019
C   I*LOC6 ! LOCATION CORE 00530019
C   I*LOC7 ! LOCATION CORE 00540019
C   I*LOC8 ! LOCATION CORE 00550019
C   I*LOS1 ! LOCATION STRUCTURE 00560019
C   I*LOS2 ! LOCATION STRUCTURE 00570019
C   I*LOS3 ! LOCATION STRUCTURE 00580019
C
*CONVEND 00590019
          00600019
          00610019
          00620019
          00630019
          00640011

```

図 C.2 インクルードメンバ *SPLCM

```

//JCLG JOB                                     00010000
//JCLG EXEC JCLG                           00020000
//SYSIN DD DATA,DLM='++'                      00030000
// JUSER 1234XXXX,AA.88888888,1234.56        00040000
// T.3 W.O I.4 C.2 GRP OPN                  00050000
    OPTF NOTIFY=JXXXXX,PASSWORD=?????????      00060000
// *T.3 W.O I.5 C.1 OPN GRP NLP              00070000
// *SPLPLOT EXEC PGM=TEMPNAME                00080000
// *STEPLIB DD DSN=J3080.*PLTNLP4.LOAD,DISP=SHR 00090000
// SPLPLOT EXEC PGM=PLTNLP4                  00100000
// STEPLIB DD DSN=J3076.TK.MELPROG.LOAD,DISP=SHR 00110000
// FTO6F001 DD SYSOUT=*                      00120000
// FTO1F001 DD DISP=(NEW,DELETE),UNIT=WK10,SPACE=(TRK,(30,30)) 00130000
// *FT11F001 DD DSN=J3076.TK.MELPSPL.DATA,DISP=SHR       00140000
// FT11F001 DD DSN=J3076.TK.MELPNWSP.DATA,DISP=SHR       00150000
// EXPAND GRNLP                                00160000
// *FT05F001 DD DSN=J3076.TK.MELPROG.DATA(PLMLSS01),DISP=SHR 00170000
// FT05F001 DD DSN=J3076.TK.MELPROG.DATA(PLMLDF01),DISP=SHR 00180000
// ****                                         00190000
// * SPL-PLOT DATA DIRECT INPUT               *
// *                                         00200000
// *                                         00210000
// *                                         00220000
// *FT05F001 DD *                               00230000
// * 1MKSA     111111                          00240000
// *
// *
// *
// *10
// *
// *TCOND   1   1   1                         00250000
// *
// *
// *TCOND   1   2   1                         00260000
// * 1
// *TCOND   1   3   1                         00270000
// * 1
// *TCOND   1   4   1                         00280000
// * 1
// *TCOND   1   5   1                         00290000
// * 1
// *TCOND   1   6   1                         00300000
// * 1
// *TCOND   1   7   1                         00310000
// * 1
// *TCOND   1   8   1                         00320000
// * 1
// *TCOND   1   9   1                         00330000
// * 1
// *TCOND   1  10   1                         00340000
// * 1
// *
// *                                         00350000
// *                                         00360000
// *                                         00370000
// *                                         00380000
// *                                         00390000
// *                                         00400000
// *                                         00410000
// *                                         00420000
// *                                         00430000
// *                                         00440000
// *                                         00450000
// *                                         00460000
// *                                         00470000
// *                                         00480000
// *                                         00490000
// *                                         00500000
// *                                         00510000
// *
// *                                         00520000
// *                                         00530000
// *                                         00540000
// */

```

図C.3 SPL-PLOTによる図形出力JCL

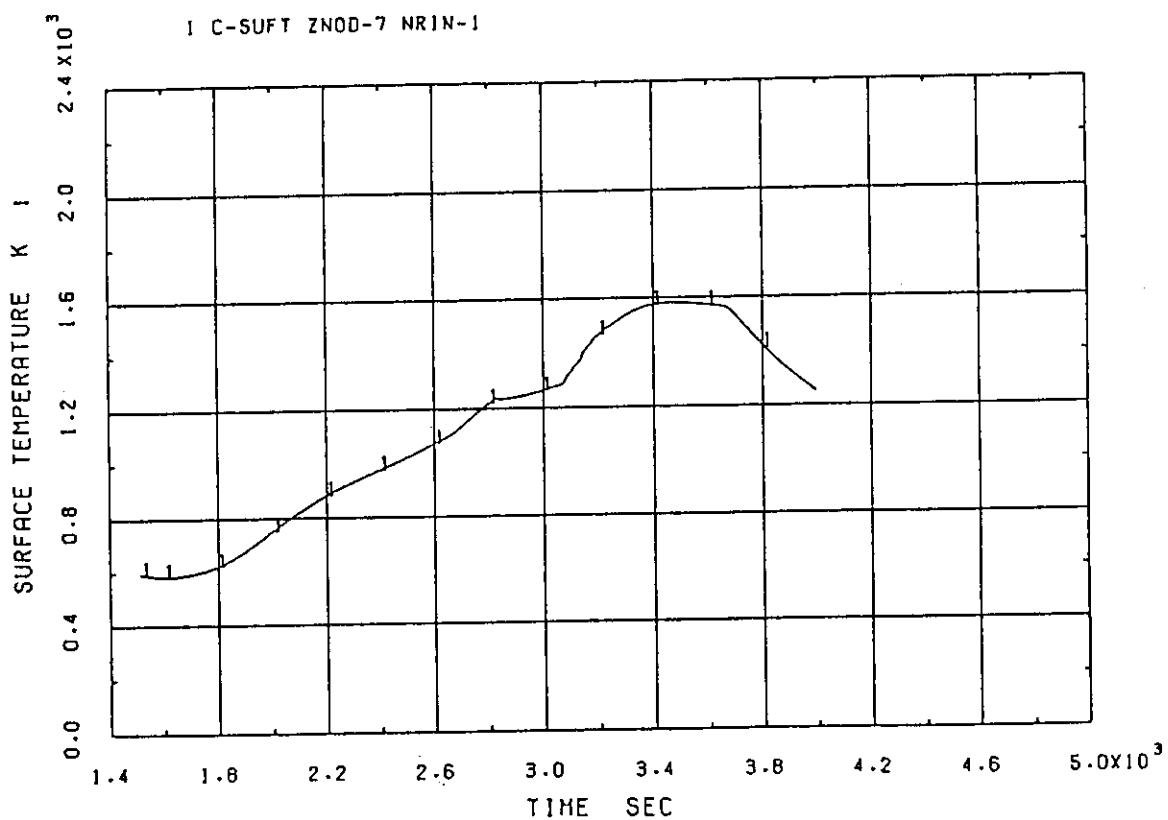
```

C
      IMPLICIT INTEGER ( A-Z )
      PARAMETER ( QIASZ = 70000 )
      PARAMETER ( QIBSZ = 50000 )
C
      COMMON / *WAREA / MEM1, IDM1, IA(QIASZ)
      COMMON / WAREA / MEM2, IDM2, IB(QIBSZ)
C
      MEM1 = QIASZ
      MEM2 = QIBSZ
      CALL *CHKON ('RDINPT   ')
      CALL *CHKON ('INITAL   ')
C
      CALL PLCNTL
C
      STOP
      END

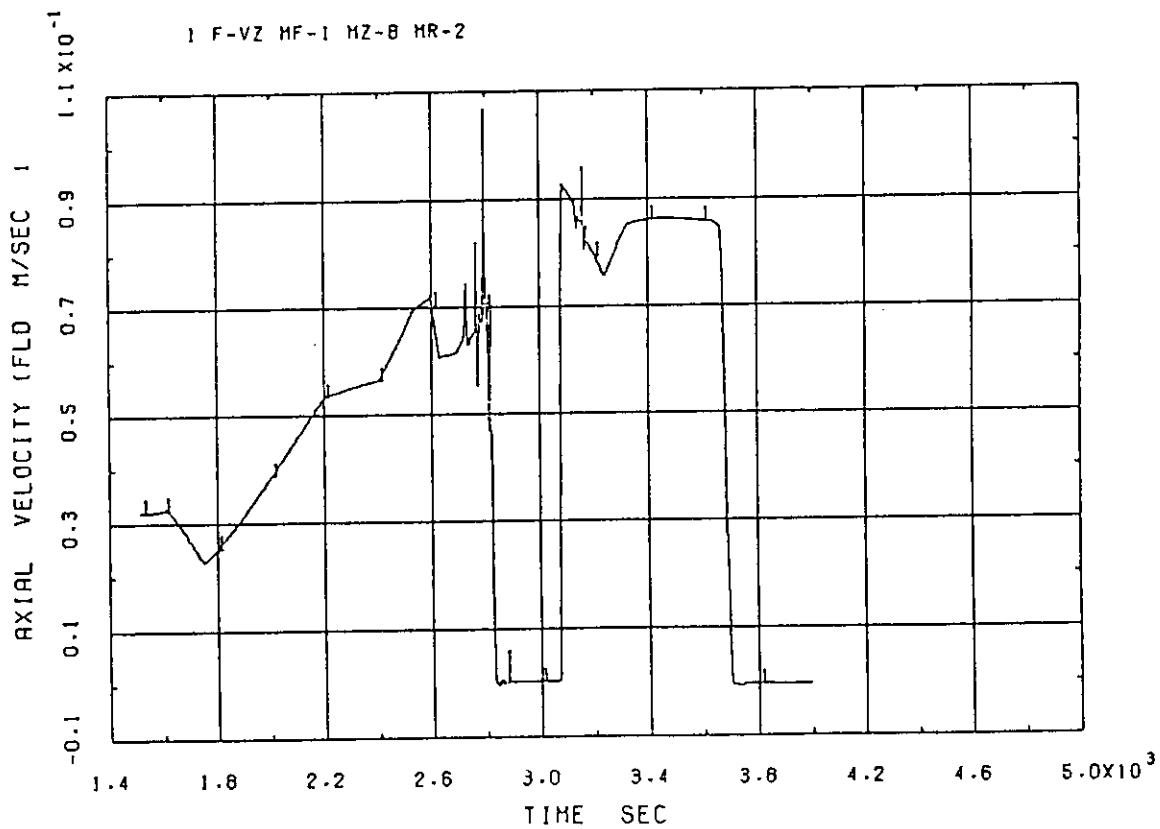
```

00010000
00020000
00030000
00040000
00050000
00060000
00070000
00080000
00090000
00100000
00110000
00120000
00130000
00140000
00150000
00160000
00170000

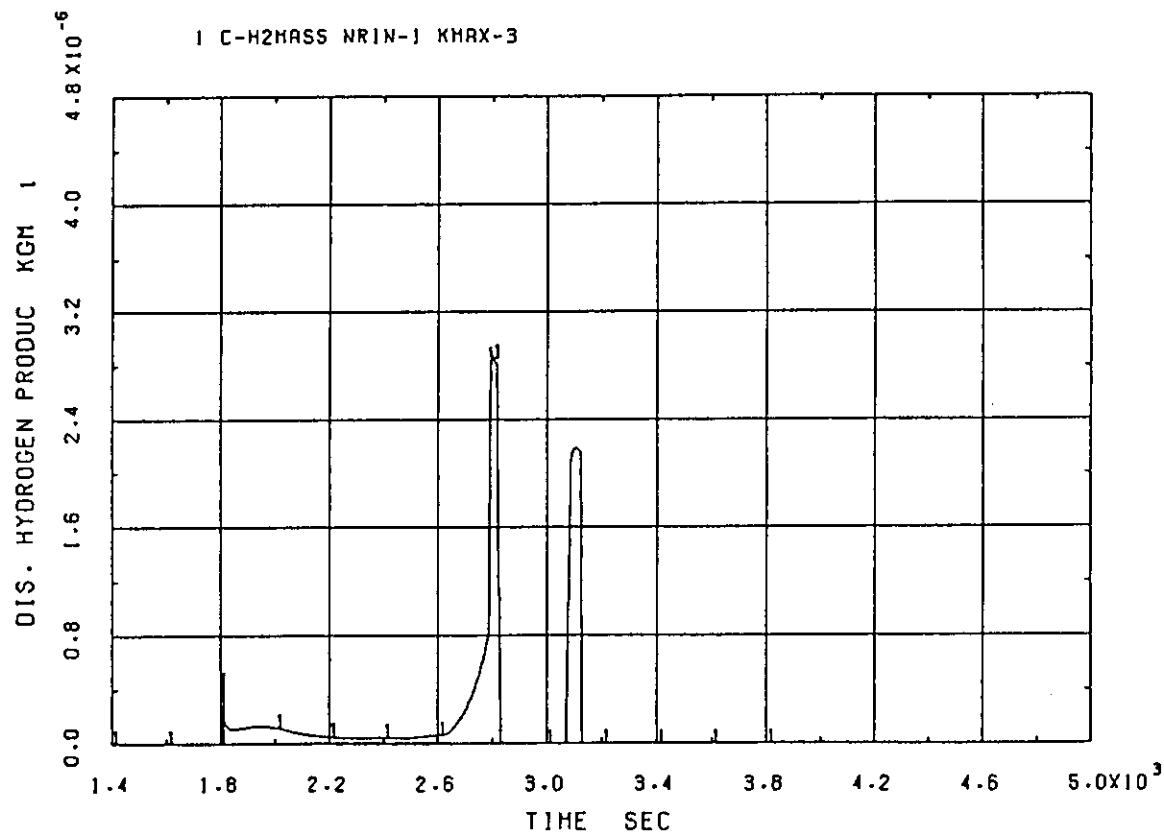
図C.4 SPL-PLOTによる図形出力の置換用メインプログラムリスト



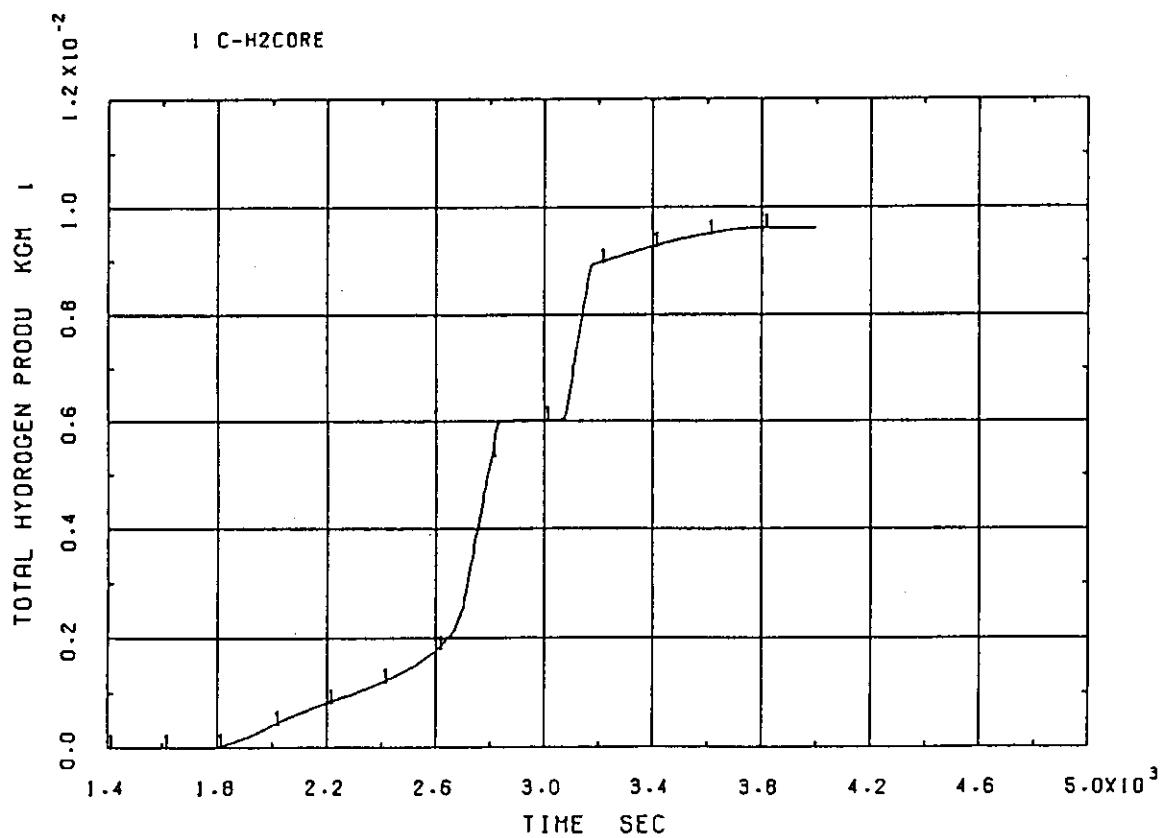
図C.5 図形処理出力の例1（燃料棒表面温度）



図C.6 図形処理出力の例2（冷却材流速）



図C.7 図形処理出力の例3（水素発生率）



図C.8 図形処理出力の例4（水素発生量の総和）