

プラント動特性解析コードSuper-COPDの開発(Ⅲ)

—入出力用インターフェースプログラム説明書—

区 分 変 更	
変更後資料番号	PNC TN9460 96-005
決裁年月日	平成10年 3月 26日

1996年5月

動力炉・核燃料開発事業団
大洗工学センター

この資料は、動燃事業団社内における検討を目的とする社内資料です。ついては複製、転載、引用等を行わないよう、また第三者への開示又は内容漏洩がないよう管理して下さい。また今回の開示目的以外のことには使用しないよう注意して下さい。

本資料についての問合せは下記に願います。

〒311-13 茨城県東茨城郡大洗町成田町4002

動力炉・核燃料開発事業団

大洗工学センター

システム開発推進部・技術管理室

プラント動特性解析コードSuper-COPDの開発（Ⅲ） —入出力用インターフェースプログラム説明書—

岩崎 隆* 大滝 明**

要 旨

Super-COPDによるプラント動特性解析の効率化を図るため、入出力データ変換処理プログラムおよびポストデータ処理プログラムを開発した。前者はSuper-COPD入力フォーマットに合わせて実測データを解析の境界条件となるように変換するプリプロセッサであり、実測データの一次遅れ補正機能と単位および入出力形式の変換機能を有するプログラムである。一方、後者はテキスト形式の実測データをSuper-COPDの出力形式と同一フォーマットに変換するとともに、実測データ及びSuper-COPDによる計算結果を、汎用の図表処理ソフトから利用可能とするための入力形式に変換して出力するポスト処理機能を有する。

以上の入出力変換プログラムの開発／整備により、Super-COPDによるプラント動特性解析結果や実測データを統計処理及び図表処理することが容易に実行可能となった。

* 原子力システム株式会社

** 基盤技術開発部 熱流体技術開発室

Development of Plant Dynamics Analysis Code: Super-COPD (III)
- Input and output Interface Program Manual -

Takashi Iwasaki*
Akira Ohtaki**

[Abstract]

In order to facilitate plant dynamic analysis by Super-COPD, the pre- and post-processor programs of the code were developed. The pre-processor program converts the measured data into the input data as boundary conditions of Super-COPD. The post-processor program converts the measured data into the output format of Super-COPD. This data conversion enables direct comparisons between measured and calculated data through the plotting utility programs.

The pre- and post-processor programs developed in this report are very useful to set the input data and the post process data in short time.

* Nuclear Energy System Inc.

** Thermal Hydraulic Research Section, Advanced Technology Division, O-arai Engineering Center

プラント動特性解析コードSuper-COPDの開発(Ⅲ)
 -入出力用インターフェースプログラム使用説明書-

目次

	Page
1. はじめに	1
2. データ処理の概要	2
3. 境界条件入力処理プログラム	3
3.1 境界条件入力処理プログラムの機能	3
3.2 境界条件入力処理ツール	3
3.3 境界条件入力処理方法	3
3.4 境界条件入力処理結果	4
4. ポストデータ処理プログラム	5
4.1 ポストデータ処理プログラムの機能	5
4.2 ポストデータ処理ツール	5
4.3 ポストデータ処理方法	5
4.4 ポストデータ処理結果	6
5. EDITコードの機能追加	7
5.1 作図データ作成機能	7
5.2 作図データ作成ツール	7
5.3 操作方法	7
5.4 作図データ作成機能を用いた出力	7
6. おわりに	9
7. 参考文献	10
Appendix-A1 「試験データ処理プログラムリスト」	25
Appendix-A2 「試験データ処理プログラム実行コマンドプロシジャリスト」	35
Appendix-B1 「ポストデータ処理プログラムリスト」	39
Appendix-B2 「ポストデータ処理プログラム実行コマンドプロシジャリスト」	47
Appendix-C 「ポストデータ処理プログラムリスト」	51

図表リスト

	Page
表リスト	
表3.3-1 MIDASによる抽出データリスト	11
表3.3-2 編集済プラント運転データファイルの配列	12
表3.3-3 境界条件入力処理プログラムの制御用入力データ	13
表3.3-4 境界条件入力処理プログラムの制御用入力データの例	15
表3.4-1 境界条件入力処理結果	16
表4.3-1 ポストデータ処理プログラムの制御用入力データ	18
表5.3-1 試験解析評価用データ処理プログラムの制御用入力データ ..	20
表5.3-2 試験解析評価用データ処理プログラムの実行例	22
図リスト	
図2.1 データ処理フロー	23
図5.4-1 試験解析評価用データ処理結果の利活用例	24

1.はじめに

モジュール型汎用プラント動特性解析コードSuper-COPDは「もんじゅ」設計解析用コードCOPDをベースとして、熱流動計算モデルの汎用化を図るとともに、各種解析モデルをそれぞれ単独で実行可能なモジュール構造とし、各モジュール間のインターフェースを入力データで指定することにより任意のプラント体系の動特性解析を可能にしている。また、各モジュール毎に境界条件が設定できるため、解析対象を機器あるいは設備に限定した部分的な特性評価も可能である。

本報の境界条件入力処理プログラムとポストデータ処理プログラムは、「もんじゅ」のプラント運転データによるコード検証^{1),2)}及び各種特性試験の解析評価を効率的に行うことを目的として作成したプリポスト処理プログラムである。境界条件入力処理プログラムは、MIDAS³⁾によって収録されたプラント運転データをSuper-COPDの入力形式に変換するもので、境界条件作成用プリプロセッサ機能を有する。ポストデータ処理プログラムは、実測データの配列をSuper-COPDの出力ファイルと同じ記録形式に変換し、実測データと計算データとをSuper-COPDの出力処理コードEDITで統一的に取り扱うための変換ツールである。また、Super-COPDの出力処理コードEDITについては、コードのテキスト出力をパーソナル・コンピュータの汎用表計算及び作図ソフトウェアに読み込めるように機能拡張した。

本報の第2章ではデータ処理の概要を、第3章では境界条件入力処理プログラムの機能と利用方法、第4章ではポストデータ処理プログラムの機能と利用方法、第5章ではEDITコードの新たな機能の使用方法について説明している。なお、Super-COPDの機能および利用方法についてはコード使用説明書⁴⁾に詳しく記載されているので参照されたい。

2. データ処理の概要

MIDASによって収録された「もんじゅ」のプラント運転データおよびSuper-COPDの計算結果の処理フローを図2.1に示す。図中の編集済プラント運転データは、KaleidaGraphやMicrosoftExcel等の汎用ソフトを用いてデータ項目の間引きや数値の表示形式を変換したもので、これらの編集処理はパーソナル・コンピュータで行う。また、編集済プラント運転データは、情報センターがサポートしているFTPコマンドでパーソナル・コンピュータのハードディスクから大型計算機の直接アクセス記憶装置（DASD）にファイル転送を行うことによって、境界条件入力処理プログラムあるいはポストデータ処理プログラムの処理対象に用いられる。

大型計算機によるデータ処理は、境界条件入力処理とポストデータ編集処理とに大別できる。EDITコードはSuper-COPDの計算結果をプロット出力あるいはテキスト出力するためのポストプロセッサであるが、Super-COPDの出力形式と同一フォーマットに編集されたデータについても出力処理を行うことができる。また、EDITコードによってDASD上に作られたテキストファイルは、FTP機能でパーソナル・コンピュータのハードディスクにファイル転送することにより、KaleidaGraph等の汎用ソフトで統計処理や作図出力に使用できる。

3. 境界条件入力処理プログラム

3.1 境界条件入力処理プログラムの機能

あらかじめテキスト形式に編集したプラント運転データに対して、実測データの一次遅れ補正、Super-COPDコードの入力データ形式に合わせた単位および配列の変換、入力変数名の設定などを行う。

3.2 境界条件入力処理ツール

- | | |
|------------------|--------------------------|
| (1) データ入力処理プログラム | CONVERT.FORT (INTERM2S) |
| (2) ロードモジュール | CONVERT.LOAD (INTERM2S) |
| (3) 実行用コマンドプロシジャ | CONVERT.CLIST (EXINTM2S) |

3.3 境界条件入力処理方法

ここでは、MIDASから抽出したプラント運転データを参考例として、実測データをSuper-COPDの境界条件に変換するための一連の処理方法について説明する。

(1) 前処理

MIDASから抽出したプラント運転データの配列を表3.3-1に示す。前処理では、このデータテーブルを境界条件入力処理プログラムの入力形式にあらかじめ編集する。編集作業はパーソナル・コンピュータの表計算ソフト (KaleidaGraph, EXCEL) を用いて、以下の手順で行う。また、編集後のデータ配列を表3.3-2に示す。

- ①SEQ-NO、YYYY、MM、DD、HH、FF、SSはそれぞれシーケンス番号、西暦、月、日、時、分、秒に相当する。これらの列データを削除する。
- ②実測データのサンプリング周期にあわせた新たな時刻データを最左列に作成する。
- ③第1行のタイトルを削除する。
- ④データの形式を10桁の浮動小数点形式に変更する。
- ⑤第1行にデータ項目数（最大10）を追加する。
- ⑥任意の区切り記号のテキスト形式で保存する。

(2) FTPによる大型計算機へのファイル転送

①FTPを実行する前に、あらかじめTSSコマンド「PFD3.2」で大型計算機のDASDに新規データセットを割り当てる。

この時、新規データセットのファイル形式は以下の通り設定する。

レコード形式：VB

レコード長：前処理したテキスト形式ファイルの最大レコード長+4

ブロック長：レコード長*ブロック数+4

②前処理したファイルをTelnetフォルダに移動する。

③前処理したファイルをFTPコマンドでパーソナル・コンピュータから大型計算機へ転送する。

(3) 制御用入力データの作成

制御用入力データは境界条件入力処理プログラムに処理条件を与えるデータセットであり、大型計算機のDASDに順編成データセットあるいは区分型データセットとして作成する。また、制御用入力データの内容を表3.3-3(1/2)と表3.3-3(2/2)に示す。表3.3-3(1/2)は実測データを格納するプール変数情報であり、入力の様式および内容はSuper-COPDの入力情報と同じである。表3.3-3(2/2)は、FTPで転送したデータ列から境界条件に採用する実測データの選択情報、一次遅れ補正用時定数および体積流量を質量流量に換算するための温度等を与える入力変数である。制御用入力データの一例を表3.3-4に示す。

(4) 境界条件入力処理プログラムの実行

境界条件入力処理プログラムを実行するためのコマンドプロシジャをAppendix-A2に示す。プログラム上の論理機番と各種入出力ファイルの対応は以下の通りである。

機番5：制御用入力データ（入力）

機番6：Super-COPDの入力形式の境界条件データ（出力）

機番8：大型計算機に転送された前処理済みファイル（入力）

3.4 境界条件入力処理結果

境界条件入力処理プログラムの出力例を表3.4-1に示す。プログラム実行後の実測データは、境界条件を格納するプール変数毎にプロセス値と時間データが一對となったデータ配列に変換されている。変換後の時間データは0秒が起点となる。

なお、Super-COPDでは、境界条件を格納する変数一つ当たりの最大データ個数はデフォルトで30に設定されており、デフォルト数を超える場合にはあらかじめソースプログラム上の配列数を増やす必要がある。

4. ポストデータ処理プログラム

4.1 ポストデータ処理プログラムの機能

評価用データ処理プログラムは、編集済みプラント運転データの配列を Super-COPDの計算結果出力用コードEDITの入力様式に変換する。

4.2 ポストデータ処理ツール

- | | |
|-------------------|--------------------------|
| (1) 評価用データ処理プログラム | CONVERT.FORT (INTERM2E) |
| (2) ロードモジュール | CONVERT.LOAD (INTERM2E) |
| (3) 実行用コマンドプロシジャ | CONVERT.CLIST (EXINTM2E) |

4.3 ポストデータ処理方法

(1) 前処理

前記3.3の(1)項と同じ。

(2) FTPによる大型計算機へのファイル転送

前記3.3の(2)項と同じ。

(3) 制御用入力データ

制御用入力データは評価用データ処理プログラムに処理条件を与えるデータセットであり、大型計算機のDASDに順編成データセットあるいは区分型データセットとして作成する。また、制御用入力データの内容を表4.3-1(1/2)および表4.3-1(2/2)に示す。表4.3-1(1/2)は実測データを格納するプール変数情報であり、入力の様式および内容はSuper-COPDの入力情報と同じである。表4.3-1(2/2)は、FTPで転送したデータ列からポスト処理用の実測データを指定するための各種変数である。

(4) ポストデータ処理プログラムの実行

ポストデータ処理プログラムを実行するためのコマンドプロシジャをAppendix-B2に示す。プログラム上の論理機番と各種入出力ファイルの対応は以下の通りである。

- | | | |
|----|---|-------------------------------------------------|
| 機番 | 5 | : 制御用入力データ (入力) |
| 機番 | 6 | : 処理情報出力ファイル (出力) |
| 機番 | 8 | : 大型計算機に転送された編集済みプラント運転データ (入力) |
| 機番 | 9 | : Super-COPDコードの計算結果と同じ出力形式のプロット情報
ファイル (出力) |

機番 10 : Super-COPDコードの計算結果と同じ出力形式のプロットデータ
ファイル (出力)

4.4 ポストデータ処理結果

ポストデータ処理プログラムの実行によって、Super-COPDの計算結果と同じ出力形式のデータセットが論理機番9と論理機番10に作成される。各データセットの記録内容は、Super-COPDの計算結果と同様にEDITコードを用いてプロット形式あるいはテキスト形式で出力できる。また、Super-COPDの計算結果とのオーバープロットが可能であるため、実測値と計算値の比較評価を容易に行うことができる。

5. EDITコードの機能追加

5.1 作図データ作成機能

Super-COPDの計算結果およびポストデータ処理プログラムの出力データセットを基に、パーソナル・コンピュータの図表計算ソフト (KaleidaGraph, EXCEL) でアクセス可能な記録形式の新規データセットを作成する。この機能は、EDITコードのテキスト出力機能を拡張したものである。

5.2 作図データ作成ツール

- | | |
|------------------|------------------------|
| (1) データ変換処理プログラム | SPEDIT.FORT |
| (2) ロードモジュール | CONVERT.LOAD (EDSUMY) |
| (3) 実行用コマンドプロシジャ | CONVERT.CLIST (EDSUMY) |

5.3 操作方法

(1) 制御用入力データ

制御用入力データの内容を表5.3-1 (1/2)と表5.3-1 (2/2)に示す。各入力変数の様式と意味はEDITコードによるデジタル値出力処理の入力データと一致させている。

(2) EDITコードの実行

実行方法は従来のEDITコードの運用方法と同様である。詳細についてはコードシステム解説書[4]を参照されたい。また、作図データ処理プログラムの実行例を表5.3-2に示す。表の左から第1列は時間情報、2列目以降はプロセス量であり、1行目はコメント行となっている。ファイル中では、12桁の実数型データがCSV (Comma Separated Values)形式で記録されている。

(3) FTPによるファイル転送

前記(2)項の処理結果を、FTPコマンドで大型計算機からパーソナル・コンピュータのハードディスクにファイル転送する。

5.4 作図データ作成機能を用いた出力例

パーソナル・コンピュータに転送したデータファイルは、汎用の図表計算ソフト (KaleidaGraph, EXCEL等) により作図処理や統計処理に行えるため、試験データと解析結果との比較評価が効率的に実施できる。一例として、実測値と計算値の偏差

の時間変化をグラフ化し、複数ループの同一プロセス量を比較した結果を図5.4-1に示す。

6. おわりに

実測データによるコード検証あるいは各種特性試験に対する解析作業の効率化を図るため、Super-COPDの入出力に係わるインターフェースプログラムを開発・整備した。

(1)境界条件入力処理プログラムの作成

「もんじゅ」のプラント運転データをSuper-COPDの境界条件として使用するために、実測データの一次遅れ補正機能、単位および入力形式の変換機能をもつプリプロセッサを作成した。この境界条件入力処理プログラムは、予めKaleidaGraphやMicrosoftExcel等の汎用ソフトでテキスト形式に変換した入力ファイル进行处理するため、データ収録システムが異なる場合でも対応できる。

(2)ポストデータ処理プログラムの作成

テキスト形式のプラント運転データを入力して、Super-COPDの出力形式と同一フォーマットの新規データファイルを作成する。これによって作成されたデータファイルは、Super-COPDの計算結果出力用コード"EDIT"を用いて時系列プロットあるいはテキスト形式で出力できるため、計算値と実測値の比較評価が容易になった。

(3)EDITコードの機能拡張

前項で述べたEDITコードのテキスト出力を、ローカルエリアネットワークのFTP機能により大型計算機からパーソナル・コンピュータにファイル転送した後、KaleidaGraphやMicrosoftExcel等の汎用ソフトでアクセスできるようEDITコードの機能を拡張した。パーソナル・コンピュータ上の汎用ソフトでは、解析結果あるいは実測データについて統計処理や各種様式の図表出力が可能であるため、コードの検証解析や試験解析を様々な形式で表示できるようになり、解析結果の妥当性を短時間で効率的に評価することが可能になった。

7. 参考文献

- [1] 大滝 明、仲井 悟、「モジュール型プラント動特性解析コードSuper-COPDの開発 —もんじゅ総合機能試験データによる検証—」,1995年2月, PNCZN9410 95-060

- [2] 大滝 明、「モジュール型プラント動特性解析コードSuper-COPDの開発 (Ⅱ) —もんじゅ試運転データによる自然循環解析機能の検証—」,1995年12月, PNC ZN9410 96-054

- [3] 村中、「マイダス (MIDAS: 「もんじゅ」プラントデータ収録システム) ” Macグラフ作図” 利用手順」,平成5年7月, SS-93-005
高速増殖原型炉もんじゅ建設所 技術開発部テクニカルメモ

- [4] 仲井 悟、岩崎 隆、他 「モジュール型プラント動特性解析コード Super-COPD —コードシステム解説書—」,1989年2月, PNC N9520 89-001

表3.3-1 MIDASによる抽出データリスト

SEQ-NO	YYYY	MM	DD	HH	FF	SS	A039	A051	A075	A078	A319
14401	1995	10	5	21	0	0	-3.5315	1627.1	3.475	2.8797	-162.19
14411	1995	10	5	21	0	10	-1.6096	1627.4	3.2375	2.8859	-159.14
14421	1995	10	5	21	0	20	-4.1409	1628.6	4.125	2.825	-154.57
14431	1995	10	5	21	0	30	-0.76587	1624.1	4.225	2.825	-161.72
14441	1995	10	5	21	0	40	1.1089	1629.3	4.4	2.8359	-162.54
14451	1995	10	5	21	0	50	-3.5315	1624.3	3.7875	2.9047	-166.88
14461	1995	10	5	21	1	0	1.0154	1629.7	2.925	2.875	-157.97
14471	1995	10	5	21	1	10	-1.3752	1625.3	3.275	2.8422	-157.15
14481	1995	10	5	21	1	20	-3.1565	1621.2	3.5625	2.8234	-162.77
14491	1995	10	5	21	1	30	-1.1409	1628.4	4.1375	2.8609	-163.36
14501	1995	10	5	21	1	40	-1.563	1634.7	3.8	2.8469	-156.8
14511	1995	10	5	21	1	50	-3.2034	1624.7	3.1875	2.8109	-165.7
14521	1995	10	5	21	2	0	-4.9377	1628.1	3.3625	2.8375	-159.14
14531	1995	10	5	21	2	10	-1.8909	1617.3	3.4125	2.8437	-160.78
14541	1995	10	5	21	2	20	1.0154	1619.8	3.7875	2.9172	-156.45
14551	1995	10	5	21	2	30	-5.219	1612.6	2.975	2.8172	-165.12
14561	1995	10	5	21	2	40	1.2029	1623.4	3.6	2.8688	-161.49
14571	1995	10	5	21	2	50	-3.3909	1631	3.7625	2.8734	-160.55
14581	1995	10	5	21	3	0	-6.1565	1619.2	4.0125	2.8437	-163.59
14591	1995	10	5	21	3	10	-2.4536	1627.2	4.225	2.8609	-164.06
14601	1995	10	5	21	3	20	-3.9534	1630	4.0625	2.7641	-160.55
14611	1995	10	5	21	3	30	-0.43774	1627.6	3.3875	2.8672	-151.64
14621	1995	10	5	21	3	40	-1.0002	1626.6	2.9625	2.8375	-155.74
14631	1995	10	5	21	3	50	-4.5627	1622.8	2.5375	2.8484	-166.05
14641	1995	10	5	21	4	0	0.12476	1624.8	4.325	2.85	-162.19
14651	1995	10	5	21	4	10	-4.8909	1627.8	3.025	2.8437	-151.05
14661	1995	10	5	21	4	20	-3.344	1624.1	3.9625	2.8516	-157.85
14671	1995	10	5	21	4	30	-4.0002	1629.1	4.0875	2.8281	-162.19
14681	1995	10	5	21	4	40	-4.5159	1619	3.0125	2.8312	-158.91
14691	1995	10	5	21	4	50	-3.5786	1628.8	3.5375	2.8531	-159.49
14701	1995	10	5	21	5	0	1.5779	1631.4	3.075	2.8953	-158.79
14711	1995	10	5	21	5	10	-4.7034	1617.8	4.1625	2.8141	-161.25
14721	1995	10	5	21	5	20	-3.2974	1633.1	3.7625	2.8563	-160.2
14731	1995	10	5	21	5	30	-1.1877	1632.4	3.4375	2.8391	-169.57

表3.3-2 編集済プラント運転データファイルの配列

8,

0.000E+00, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 4.082E+03, 4.311E+03, 3.246E+02
 1.000E+01, 3.251E+02, 3.257E+02, 3.235E+02, 3.249E+02, 4.109E+03, 4.335E+03, 3.246E+02
 2.000E+01, 3.251E+02, 3.256E+02, 3.236E+02, 3.249E+02, 4.127E+03, 4.311E+03, 3.246E+02
 3.000E+01, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 4.047E+03, 4.261E+03, 3.246E+02
 4.000E+01, 3.251E+02, 3.256E+02, 3.236E+02, 3.249E+02, 4.041E+03, 4.355E+03, 3.246E+02
 5.000E+01, 3.251E+02, 3.257E+02, 3.236E+02, 3.249E+02, 3.998E+03, 4.347E+03, 3.246E+02
 6.000E+01, 3.251E+02, 3.257E+02, 3.235E+02, 3.248E+02, 4.061E+03, 4.304E+03, 3.246E+02
 7.000E+01, 3.251E+02, 3.257E+02, 3.236E+02, 3.249E+02, 3.941E+03, 4.355E+03, 3.246E+02
 8.000E+01, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.970E+03, 4.332E+03, 3.246E+02
 9.000E+01, 3.251E+02, 3.257E+02, 3.235E+02, 3.249E+02, 3.710E+03, 4.365E+03, 3.246E+02
 1.000E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.904E+03, 4.309E+03, 3.246E+02
 1.100E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.248E+02, 3.866E+03, 4.380E+03, 3.246E+02
 1.200E+02, 3.251E+02, 3.256E+02, 3.236E+02, 3.248E+02, 3.774E+03, 4.347E+03, 3.246E+02
 1.300E+02, 3.251E+02, 3.256E+02, 3.236E+02, 3.250E+02, 3.822E+03, 4.348E+03, 3.246E+02
 1.400E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.248E+02, 3.625E+03, 4.331E+03, 3.246E+02
 1.500E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.248E+02, 3.765E+03, 4.330E+03, 3.246E+02
 1.600E+02, 3.251E+02, 3.256E+02, 3.236E+02, 3.249E+02, 3.694E+03, 4.311E+03, 3.246E+02
 1.700E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.616E+03, 4.315E+03, 3.246E+02
 1.800E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.541E+03, 4.357E+03, 3.246E+02
 1.900E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.250E+02, 3.678E+03, 4.286E+03, 3.246E+02
 2.000E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.249E+02, 3.668E+03, 4.321E+03, 3.246E+02
 2.100E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.250E+02, 3.642E+03, 4.370E+03, 3.246E+02
 2.200E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.679E+03, 4.289E+03, 3.246E+02
 2.300E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.250E+02, 3.540E+03, 4.332E+03, 3.246E+02
 2.400E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.591E+03, 4.325E+03, 3.246E+02
 2.500E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.646E+03, 4.324E+03, 3.246E+02
 2.600E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.250E+02, 3.567E+03, 4.324E+03, 3.246E+02
 2.700E+02, 3.251E+02, 3.256E+02, 3.236E+02, 3.250E+02, 3.586E+03, 4.343E+03, 3.246E+02
 2.800E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.249E+02, 3.630E+03, 4.347E+03, 3.246E+02
 2.900E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.547E+03, 4.317E+03, 3.246E+02
 3.000E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.249E+02, 3.634E+03, 4.374E+03, 3.246E+02
 3.100E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.249E+02, 3.493E+03, 4.350E+03, 3.246E+02
 3.200E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.249E+02, 3.528E+03, 4.347E+03, 3.246E+02
 3.300E+02, 3.250E+02, 3.257E+02, 3.235E+02, 3.249E+02, 3.532E+03, 4.378E+03, 3.246E+02
 3.400E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.619E+03, 4.350E+03, 3.246E+02
 3.500E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.248E+02, 3.587E+03, 4.337E+03, 3.246E+02
 3.600E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.250E+02, 3.510E+03, 4.337E+03, 3.246E+02
 3.700E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.250E+02, 3.517E+03, 4.363E+03, 3.246E+02
 3.800E+02, 3.251E+02, 3.257E+02, 3.236E+02, 3.250E+02, 3.519E+03, 4.350E+03, 3.246E+02
 3.900E+02, 3.251E+02, 3.257E+02, 3.235E+02, 3.250E+02, 3.542E+03, 4.355E+03, 3.246E+02
 4.000E+02, 3.251E+02, 3.256E+02, 3.235E+02, 3.249E+02, 3.459E+03, 4.365E+03, 3.246E+02

表3.3-3 境界条件入力処理プログラムの制御用入力データ(1/2)

変数名	意味	単位	備考
NTEMP	フィッティングするプール変数TEMPの数 (MAX. 10)	—	
NTMP(i)	i番目にフィッティングするプール変数TEMPの番号	—	
NFLWN	フィッティングするプール変数FLWNの数 (MAX. 10)	—	
NFNNET(i)	i番目にフィッティングするプール変数FLWNの流路網番号	—	
NFNPAS(i)	i番目にフィッティングするプール変数FLWNの流路番号	—	
NPRSN	フィッティングするプール変数PRSNの数 (MAX. 10)	—	
NPNNET(i)	i番目にフィッティングするプール変数PRSNの流路網番号	—	
NPNNOD(i)	i番目にフィッティングするプール変数PRSNの圧力ノード番号	—	
NPMPN	フィッティングするプール変数PMPNの数 (MAX. 10)	—	
NHNPMP(i)	i番目にフィッティングするプール変数PMPNのポンプ番号	—	
NHNNUM(i)	i番目にフィッティングするプール変数PMPNのプロセス量番号	—	
NVLVN	フィッティングするプール変数VLVNの数 (MAX. 10)	—	
NVNVLV(i)	i番目にフィッティングするプール変数VLVNのバルブ番号	—	
NVNNUM(i)	i番目にフィッティングするプール変数VLVNのプロセス量番号	—	
	i=1~10: データ数		

表3.3-3 境界条件入力処理プログラムの制御用入力データ(2/2)

変数名	意味	単位	備考
IBTEMP(i)	i番目にフィッティングするプール変数TEMPに対応する実測データのブロック番号	—	
INTEMP(i)	i番目にフィッティングするプール変数TEMPに対応する実測データのブロック内の位置	—	
TMTEMP(i)	i番目にフィッティングするプール変数TEMPに対応する実測データの時定数	sec	
IBFLWN(i)	i番目にフィッティングするプール変数FLWNに対応する実測データのブロック番号	—	
INFLWN(i)	i番目にフィッティングするプール変数FLWNに対応する実測データのブロック内の位置	—	
FLTEMP(i)	i番目にフィッティングするプール変数FLWNを質量流量に変換するときに使用する密度算出用の温度	℃	0.0の時は変換しない
IBPRSN(i)	i番目にフィッティングするプール変数PRSNに対応する実測データのブロック番号	—	
INPRSN(i)	i番目にフィッティングするプール変数PRSNに対応する実測データのブロック内の位置	—	
IBPMPN(i)	i番目にフィッティングするプール変数PMPNに対応する実測データのブロック番号	—	
INPMPN(i)	i番目にフィッティングするプール変数PMPNに対応する実測データのブロック内の位置	—	
IBVLVN(i)	i番目にフィッティングするプール変数VLVNに対応する実測データのブロック番号	—	
INVLVN(i)	i番目にフィッティングするプール変数VLVNに対応する実測データのブロック内の位置	—	
TIMES	フィッティング開始する時間	sec	
	i=1~10:データ数		

表3.3-4 境界条件入力処理プログラム制御用入力データの例

```
&POOLI
  NTEMP = 3,
  NTMP (1) = 130, 230, 330,
  IBTEMP (1) = 1, 1, 1,
  INTEMP (1) = 2, 3, 4,
  TMTEMP (1) = 16.0, 16.0, 16.0,
  TIMES      = 0.0,
&END
```

表3.4-1 境界条件入力処理結果(1/2)

```

&NAMAA2
  NTEMP = 3,
  NTMP ( 1)= 130,      ITMAX ( 1) = 38,
  XTEMP (1, 1) =
    0.00, 10.00, 20.00, 30.00, 40.00, 50.00, 60.00,
    70.00, 80.00, 90.00, 100.00, 110.00, 120.00, 130.00,
    140.00, 150.00, 160.00, 170.00, 180.00, 190.00, 200.00,
    210.00, 220.00, 230.00, 240.00, 250.00, 260.00, 270.00,
    280.00, 290.00, 300.00, 310.00, 320.00, 330.00, 340.00,
    350.00, 360.00, 370.00,
  YTEMP (1, 1) =
    325.10, 325.10, 325.10, 325.10, 325.10, 325.10, 325.10,
    325.10, 325.10, 325.10, 325.10, 325.10, 325.10, 325.10,
    325.10, 325.10, 325.10, 325.10, 325.10, 325.10, 325.10,
    325.10, 325.10, 325.09, 325.07, 325.05, 325.08, 325.11,
    325.10, 325.11, 325.09,
  TGAIN ( 1) = 1.0,      ITST ( 1) = 2,
  NTMP ( 2)= 230,      ITMAX ( 2) = 38,
  XTEMP (1, 2) =
    0.00, 10.00, 20.00, 30.00, 40.00, 50.00, 60.00,
    70.00, 80.00, 90.00, 100.00, 110.00, 120.00, 130.00,
    140.00, 150.00, 160.00, 170.00, 180.00, 190.00, 200.00,
    210.00, 220.00, 230.00, 240.00, 250.00, 260.00, 270.00,
    280.00, 290.00, 300.00, 310.00, 320.00, 330.00, 340.00,
    350.00, 360.00, 370.00,
  YTEMP (1, 2) =
    325.60, 325.58, 325.58, 325.64, 325.68, 325.71, 325.66,
    325.65, 325.64, 325.62, 325.58, 325.61, 325.59, 325.65,
    325.64, 325.63, 325.58, 325.66, 325.67, 325.72, 325.65,
    325.67, 325.62, 325.64, 325.56, 325.63, 325.60, 325.69,
    325.63, 325.72, 325.68, 325.72, 325.64, 325.63, 325.57,
    325.63, 325.61, 325.71,
  TGAIN ( 2) = 1.0,      ITST ( 2) = 2,
  NTMP ( 3)= 330,      ITMAX ( 3) = 38,
  XTEMP (1, 3) =
    0.00, 10.00, 20.00, 30.00, 40.00, 50.00, 60.00,
    70.00, 80.00, 90.00, 100.00, 110.00, 120.00, 130.00,
    140.00, 150.00, 160.00, 170.00, 180.00, 190.00, 200.00,
    210.00, 220.00, 230.00, 240.00, 250.00, 260.00, 270.00,
    280.00, 290.00, 300.00, 310.00, 320.00, 330.00, 340.00,
    350.00, 360.00, 370.00,
  YTEMP (1, 3) =
    323.50, 323.55, 323.55, 323.58, 323.58, 323.54, 323.55,
    323.51, 323.50, 323.50, 323.54, 323.59, 323.57, 323.51,
    323.53, 323.55, 323.51, 323.49, 323.50, 323.49, 323.50,
  
```

表3.4-1 境界条件入力処理結果(2/2)

```
323.49, 323.50, 323.49, 323.51, 323.53, 323.56, 323.51,  
323.50, 323.50, 323.50, 323.50, 323.50, 323.50, 323.50,  
323.51, 323.53, 323.55,  
    TGAIN( 3) = 1.0,    ITST( 3) = 2,  
&END
```

表4.3-1 ポストデータ処理プログラムの制御用入力データ(1/2)

変数名	意味	単位	備考
NOTEMP	出力するプール変数TEMPの項数 (max. 10)	—	
NPTEMP(i)	i項目に出力するプール変数TEMPの数	—	
NQTEMP(j,i)	i項目に出力するプール変数TEMPの番号	—	
NOFLWN	出力するプール変数FLWNの項数 (max. 10)	—	
NPFLWN(i)	i項目に出力するプール変数FLWNの数	—	
NQFLWN(j,i)	i項目に出力するプール変数FLWNの流路網番号	—	
NRFLWN(j,i)	i項目に出力するプール変数FLWNの流路番号	—	
NOPRSN	出力するプール変数PRSNの項数 (max. 10)	—	
NPPRSN(i)	i項目に出力するプール変数PRSNの数	—	
NQPRSN(j,i)	i項目に出力するプール変数PRSNの流路網番号	—	
NRPRSN(j,i)	i項目に出力するプール変数PRSNの圧力ノード番号	—	
NOPMPN	出力するプール変数PMPNの項数 (max. 10)	—	
NPPMPN(i)	i項目に出力するプール変数PMPNの数	—	
NQPMPN(j,i)	i項目に出力するプール変数PMPNのポンプ番号	—	
NRPMPN	i項目に出力するプール変数PMPNのプロセス量番号	—	
NOVLVN	出力するプール変数VLVNの項数 (max. 10)	—	
NPVLVN(i)	i項目に出力するプール変数VLVNの数	—	
NQVLVN(j,i)	i項目に出力するプール変数VLVNのバルブ番号	—	
NRVLVN(j,i)	i項目に出力するプール変数PRSNのプロセス量番号	—	

表4.3-1 ポストデータ処理プログラムの制御用入力データ(2/2)

変数名	意味	単位	備考
IBTEMP(i)	i番目にフィッティングするプール変数TEMPに対応する実測データのブロック番号	—	
INTEMP(i)	i番目にフィッティングするプール変数TEMPに対応する実測データのブロック内の位置	—	
IBFLWN(i)	i番目にフィッティングするプール変数FLWNに対応する実測データのブロック番号	—	
INFLWN(i)	i番目にフィッティングするプール変数FLWNに対応する実測データのブロック内の位置	—	
IBPRSN(i)	i番目にフィッティングするプール変数PRSNに対応する実測データのブロック番号	—	
INPRSN(i)	i番目にフィッティングするプール変数PRSNに対応する実測データのブロック内の位置	—	
IBPMPN(i)	i番目にフィッティングするプール変数PMPNに対応する実測データのブロック番号	—	
INPMPN(i)	i番目にフィッティングするプール変数PMPNに対応する実測データのブロック内の位置	—	
IBVLVN(i)	i番目にフィッティングするプール変数VLVNに対応する実測データのブロック番号	—	
INVLVN(i)	i番目にフィッティングするプール変数VLVNに対応する実測データのブロック内の位置	—	
	i=1~10:データ数		

表5.3-1 試験解析評価用データ処理プログラム制御用入力データ(1/2)

カードNo.	FORMAT	変数名	意味
1	314	ISELCT	選択機能オプション =1; 時系列バイナリー・データ出力処理 =2; 時系列FORMATデータ出力処理 =3; 提出用テープ作成処理 =4; プロット図作成処理 =5; オーバープロット図作成処理 =6; デジタル値出力処理
		MACHIN	使用マシン識別オプション =1; CRAY1 =2; IBM =3; FACOM(プロット単位系cm)
		NUMREC	バイナリ・ブロックデータの1ブロック当たりのアウトプットデータ数(COPDにより決まる)
2	8A4	TITLE1	タイトル(32文字)
3	8A4	TITLE2	タイトル(32文字)
4	20A4	CMENT1	コメント(80文字)
5	20A4	CMENT2	コメント(80文字)

表5.3-1 試験解析評価用データ処理プログラム制御用入力データ(2/2)

カードNo.	FORMAT	変数名	意味
6	314	IUNIT1	時系列データ又は、COPD でユニット10より出力されたデータの読み込みを行うユニット番号
		IUNIT2	COPD でユニット9より出力されたデータの読み込みを行うユニット番号
		INPUTD	入力データ識別オプション =1; 時系列バイナリ・データ (EDIT出力) =2; 時系列フォーマット・データ (EDIT出力) =3; バイナリ・ブロック・データ (COPD出力)
7	214	IOUT	出力データ数 (MAX=10)
		JOUT	出力データ・タイプ・オプション =1; Eタイプで出力する =2; Fタイプで出力する
8	314,2X, 3A4	LOOPX	出力するデータのプール変数の種類 1:TEMP 2:FLWN 3:PRSN 4:PMPN 5:VLVN
		INDXX1	出力するデータのプール変数番号添え字 例 TEMP (J) J → INDXX1 FLWN (I, J) I → INDXX2; 1次元配列の時は不要
		INDXX2	
		MEMO	ヘッダタイトル (12 文字)

備考 No.8のカードをIOUT枚用意する。
No7、No.8のカードを繰り返しセット可能。

表5.3-2 試験解析評価用データ処理プログラムの実行例

TIME, SH	, SH (D/C)	, EV	, EV (D/C)	, POFC	, PUMP	, OF. T	
0.00000E+00,	0.00000E+00,	-0.38208E-01,	0.44220E-01,	-0.43336E-01,	-0.11102E-04,	0.45026E-01,	0.12980E+01,
0.51000E+00,	-0.67077E-03,	-0.38682E-01,	0.45141E-01,	-0.43055E-01,	0.75031E-04,	0.45035E-01,	0.12980E+01,
0.10100E+01,	-0.23207E-02,	-0.40162E-01,	0.47078E-01,	-0.42175E-01,	0.13664E-03,	0.45032E-01,	0.12979E+01,
0.15100E+01,	-0.47479E-02,	-0.42733E-01,	0.49834E-01,	-0.40425E-01,	0.23829E-04,	0.44975E-01,	0.12979E+01,
0.20100E+01,	-0.78310E-02,	-0.46249E-01,	0.53181E-01,	-0.37532E-01,	-0.44374E-04,	0.44830E-01,	0.12979E+01,
0.25100E+01,	-0.11454E-01,	-0.50382E-01,	0.56862E-01,	-0.33357E-01,	-0.16191E-03,	0.44584E-01,	0.12978E+01,
0.30100E+01,	-0.15478E-01,	-0.54735E-01,	0.60624E-01,	-0.27937E-01,	-0.28676E-03,	0.44227E-01,	0.12978E+01,
0.35100E+01,	-0.19729E-01,	-0.58935E-01,	0.64241E-01,	-0.21484E-01,	-0.32613E-03,	0.43738E-01,	0.12977E+01,
0.40100E+01,	-0.23999E-01,	-0.62684E-01,	0.67528E-01,	-0.14352E-01,	-0.24745E-03,	0.43114E-01,	0.12977E+01,
0.45100E+01,	-0.28065E-01,	-0.65769E-01,	0.70344E-01,	-0.69779E-02,	-0.39292E-04,	0.42356E-01,	0.12977E+01,
0.50100E+01,	-0.31715E-01,	-0.68052E-01,	0.72589E-01,	0.17656E-03,	0.13120E-03,	0.41470E-01,	0.12976E+01,
0.55100E+01,	-0.34771E-01,	-0.69449E-01,	0.74203E-01,	0.66863E-02,	0.27297E-03,	0.40461E-01,	0.12976E+01,
0.60100E+01,	-0.37103E-01,	-0.69927E-01,	0.75162E-01,	0.12205E-01,	0.43897E-03,	0.39335E-01,	0.12976E+01,
0.65100E+01,	-0.38638E-01,	-0.69486E-01,	0.75473E-01,	0.16492E-01,	0.63034E-03,	0.38101E-01,	0.12975E+01,
0.70100E+01,	-0.39342E-01,	-0.68192E-01,	0.75163E-01,	0.19427E-01,	0.84092E-03,	0.36766E-01,	0.12975E+01,
0.75100E+01,	-0.39208E-01,	-0.66166E-01,	0.74276E-01,	0.21006E-01,	0.10608E-02,	0.35340E-01,	0.12975E+01,
0.80100E+01,	-0.38261E-01,	-0.63549E-01,	0.72857E-01,	0.21325E-01,	0.12774E-02,	0.33832E-01,	0.12974E+01,
0.85100E+01,	-0.36557E-01,	-0.60456E-01,	0.70947E-01,	0.20561E-01,	0.14763E-02,	0.32254E-01,	0.12974E+01,
0.90100E+01,	-0.34173E-01,	-0.56961E-01,	0.68572E-01,	0.18948E-01,	0.16419E-02,	0.30613E-01,	0.12974E+01,
0.95100E+01,	-0.31204E-01,	-0.53096E-01,	0.65754E-01,	0.16739E-01,	0.17589E-02,	0.28919E-01,	0.12973E+01,
0.10000E+02,	-0.27889E-01,	-0.49049E-01,	0.62646E-01,	0.14274E-01,	0.18134E-02,	0.27251E-01,	0.12973E+01,

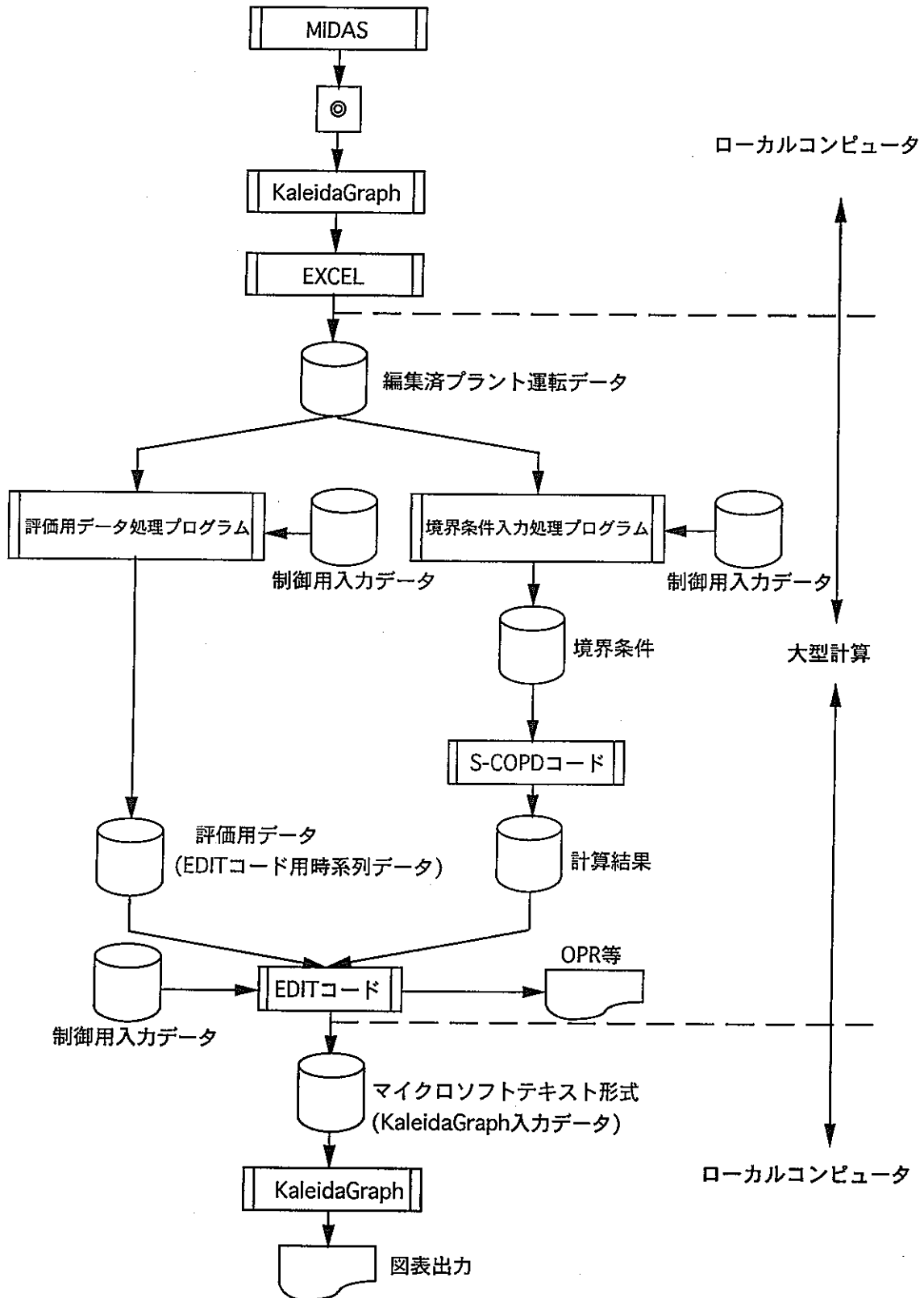


図2.1 データ処理フロー

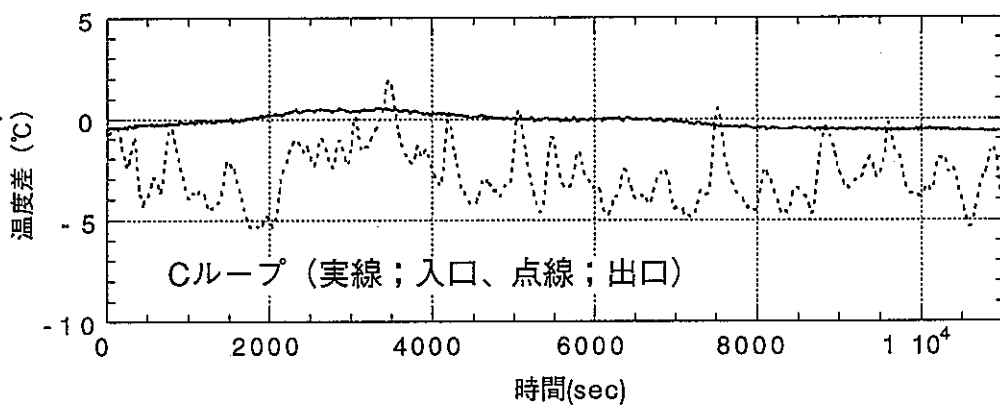
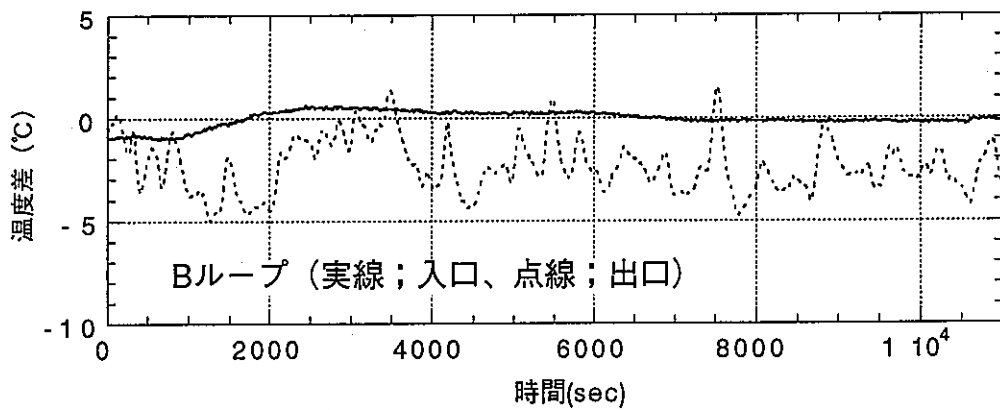
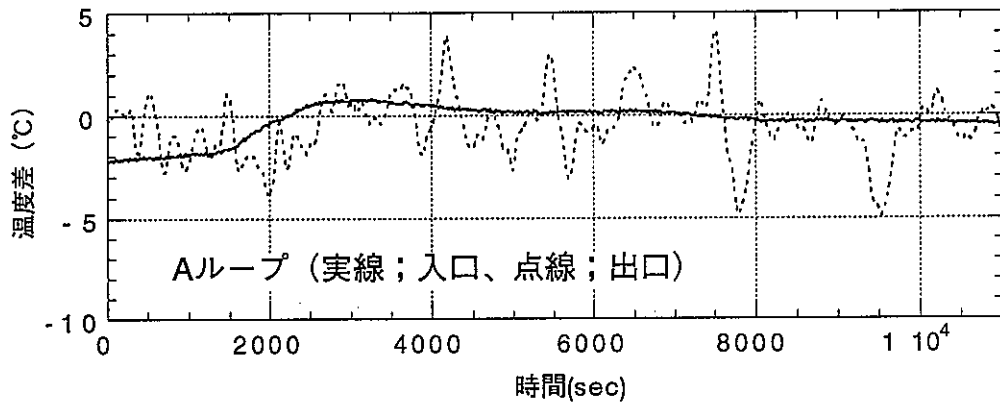


図5.4-1 試験解析評価用データ処理結果の利活用例

Appendix-A 1

「試験データ変換処理プログラムリスト」

```

C
C MAKE SUPER COPD DATA
C MONJYU ==> SUPER COPD
C
PARAMETER (NMAX=25)
CHARACTER LINE (NMAX)*9, COMMA (NMAX)*1
DIMENSION VAL (2000, NMAX)
DIMENSION TTIME (2000),
* NTMP (10) , YTEMP (2000, 10),
* NFNNET (10), NFNPAS (10), YFLWN (2000, 10),
* NPNNET ( 1), NPNNOD ( 1), YPRSN (2000, 1),
* NHNPMP ( 1), NHNNUM ( 1), YPMPN (2000, 1),
* NVNVLV ( 1), NVNNUM ( 1), YVLVN (2000, 1),
* IBTEMP (10), INTEMP (10), TMTEMP (10),
* IBFLWN (10), INFLWN (10), FLTEMP (10),
* IBPRSN ( 1), INPRSN ( 1),
* IBPMPN ( 1), INPMPN ( 1),
* IBVLVN ( 1), INVLVN ( 1)
C
NAMELIST/POOLI/ IBTEMP, INTEMP, TMTEMP,
* IBFLWN, INFLWN, FLTEMP,
* IBPRSN, INPRSN,
* IBPMPN, INPMPN,
* IBVLVN, INVLVN,
* NTEMP, NTMP,
* NFLWN, NFNNET, NFNPAS,
* NPRSN, NPNNET, NPNNOD,
* NPMPN, NHNPMP, NHNNUM,
* NVLVN, NVNVLV, NVNNUM,
* TIMES
C
C
DATA IBTEMP, INTEMP/10*0, 10*0/
DATA IBFLWN, INFLWN/10*0, 10*0/, FLTEMP/10*0.0/
DATA IBPRSN, INPRSN/ 0, 0/
DATA IBPMPN; INPMPN/ 0, 0/
DATA IBVLVN, INVLVN/ 0, 0/
DATA NTMP /10*0/
DATA NFNNET, NFNPAS/10*0, 10*0/
DATA NPNNET, NPNNOD/ 0, 0/
DATA NHNPMP, NHNNUM/ 0, 0/
DATA NVNVLV, NVNNUM/ 0, 0/
DATA TTIME/2000*0.0/
DATA YTEMP/20000*0.0/
DATA YFLWN/20000*0.0/
DATA YPRSN/ 2000*0.0/
DATA YPMPN/ 2000*0.0/
DATA YVLVN/ 2000*0.0/
C
FUNCR (XOLD, XNEW, YOLD, TCON, GAIN, DT)
* = ( (TCON + 0.5*DT)*XNEW - (TCON - 0.5*DT)*XOLD

```



```

*          - 0.5*GAIN*DT*YOLD ) / (0.5*GAIN*DT )
  FUNCRR (XOLD, YNEW, YOLD, TCON, GAIN, DT)
1      = ((TCON+0.5*DT)*YNEW - (TCON-0.5*DT)*YOLD) / (0.5*GAIN*DT) - XOLD
C
C      CALL ERRSET (132, 255, -1, 1, 1, 0)
C
      REWIND5
      READ (5, POOLI)
C      WRITE (6, POOLI)
C
C      READ (8)
C ADD START
      READ (8, *)N
C      N = 0
C      DO 1112 I=1, 20
C      READ (8, 1111, ERR=1113) (LINE (J), COMMA (J), J=1, I)
C      WRITE (6, 1111) (LINE (J), COMMA (J), J=1, I)
C      N = N + 1
C      IF (COMMA (I).NE. ', ' ) GO TO 1113
C      REWIND 8
C1112 CONTINUE
C1111 FORMAT (20 (A9, A1))
C1113 CONTINUE
C ADD END
      JTOTAL = 0
      NDUMMY = 0
1      READ (8, *, END=2) TIME
      IF (TIME. LT. TIMES) THEN
          NDUMMY = NDUMMY + 1
          GO TO 1
      END IF
      JTOTAL = JTOTAL + 1
      IF (JTOTAL. EQ. 1) TIMES = TIME
      TTIME (JTOTAL) = TIME - TIMES
      IF (JTOTAL. GE. 2000) GO TO 2
C      WRITE (6, *) JTOTAL, TTIME (JTOTAL)
      GO TO 1
2      CONTINUE
C
      DO 10 I1=1, NTEMP
      REWIND 8
C      DO 11 I5=1, IBTEMP (I1) + (IBTEMP (I1) - 1) * JTOTAL
C 11 READ (8, *) DUMMY
      DO 11 I5=1, NDUMMY+1
11 READ (8, *) DUMMY
      DO 12 I3=1, JTOTAL
      READ (8, *, END=2) (VAL (I3, I4), I4=1, N)
      YTEMP (I3, I1) = VAL (I3, INTEMP (I1))
C      WRITE (6, *) I3, I1, YTEMP (I3, I1)
12 CONTINUE
      YTEMP (1, I1) = VAL (1, INTEMP (I1))

```

```

DO 10 I3=2, JTOTAL
XOLD = VAL (I3-1, INTEMP (I1))
XNEW = VAL (I3 , INTEMP (I1))
YOLD = YTEMP (I3-1, I1)
DT = TTIME (I3) - TTIME (I3-1)
TCON = TMTEMP (I1)
YTEMP (I3, I1) = FUNCR (XOLD, XNEW, YOLD, TCON, 1. 0, DT)
C WRITE (6, *) I3, DT, YTEMP (I3, I1)
10 CONTINUE
DO 13 I1=1, NTEMP
DO 13 NN=1, 3
DO 13 I4=2, JTOTAL
13 YTEMP (I4, I1)
* = 0.5 * (0.5 * (YTEMP (I4-1, I1) + YTEMP (I4+1, I1)) + YTEMP (I4, I1))
C
DO 20 I1=1, NFLWN
REWIND 8
C DO 21 I5=1, IBFLWN (I1) + (IBFLWN (I1) - 1) * JTOTAL
C 21 READ (8, *) DUMMY
DO 21 I5=1, NDUMMY+1
21 READ (8, *) DUMMY
TX = FLTEMP (I1)
CALL SODIUM (2, TX, RO)
DO 22 I3=1, JTOTAL
READ (8, *, END=2) (VAL (I3, I4), I4=1, N)
IF (TX. GT. 0. 0) THEN
YFLWN (I3, I1) = VAL (I3, INFLWN (I1)) * RO / 3600. 0
C WRITE (6, *) I3, TX, RO, VAL (I3, INFLWN (I1)), YFLWN (I3, I1)
ELSE
YFLWN (I3, I1) = VAL (I3, INFLWN (I1))
END IF
22 CONTINUE
C DO 20 I3=2, JTOTAL
C XOLD = VAL (I3-1, INFLWN (I1))
C XNEW = VAL (I3 , INFLWN (I1))
C YOLD = YFLWN (I3-1, I1)
C DT = TTIME (I3) - TTIME (I3-1)
C TCON = TMFLWN (I1)
C YFLWN (I3, I1) = FUNCRR (XOLD, XNEW, YOLD, TCON, 1. 0, DT)
C WRITE (6, *) I3, DT, YFLWN (I3, I1)
20 CONTINUE
C DO 23 I1=1, NFLWN
C DO 23 NN=1, 3
C DO 13 I4=2, JTOTAL
C 23 YFLWN (I4, I1)
C * = 0.5 * (0.5 * (YTEMP (I4-1, I1) + YTEMP (I4+1, I1)) + YTEMP (I4, I1))
C WRITE (6, *) YFLWN (I3, I1)
C
DO 30 I1=1, NPRSN
REWIND 8
C DO 31 I5=1, IBPRSN (I1) + (IBPRSN (I1) - 1) * JTOTAL

```

```

C 31 READ (8,*)DUMMY
    DO 31 I5=1,NDUMMY+1
31 READ (8,*)DUMMY
    DO 30 I3=1,JTOTAL
    READ (8,*,END=2) (VAL (I3, I4), I4=1, N)
    YPRSN (I3, I1) = VAL (I3, INPRSN (I1))
C    WRITE (6,*)YPRSN (I3, I1)
30 CONTINUE
C
    DO 40 I1=1,NPMPN
    REWIND 8
C    DO 41 I5=1, IBPMPN (I1)+(IBPMPN (I1) -1)*JTOTAL
C 41 READ (8,*)DUMMY
    DO 41 I5=1,NDUMMY+1
41 READ (8,*)DUMMY
    DO 40 I3=1,JTOTAL
    READ (8,*,END=2) (VAL (I3, I4), I4=1, N)
    YPMPN (I3, I1) = VAL (I3, INPMPN (I1))
C    WRITE (6,*)YPMPN (I3, I1)
40 CONTINUE
C
    DO 50 I1=1,NVLVN
    REWIND 8
C    DO 51 I5=1, IBVLVN (I1)+(IBVLVN (I1) -1)*JTOTAL
C 51 READ (8,*)DUMMY
    DO 51 I5=1,NDUMMY+1
51 READ (8,*)DUMMY
    DO 50 I3=1,JTOTAL
    READ (8,*,END=2) (VAL (I3, I4), I4=1, N)
    YVLVN (I3, I1) = VAL (I3, INVLVN (I1))
C    WRITE (6,*)YVLVN (I3, I1)
50 CONTINUE
C
C
C
C ----WRITE DATA ----
C
    JTOTAL = JTOTAL - 3
    WRITE (6,150)
C
C. TEMP
    IF (NTEMP.NE.0)THEN
    WRITE (6,110)NTEMP
    DO 115 I=1,NTEMP
    WRITE (6,111) I,NTMP (I), I, JTOTAL
    WRITE (6,112) I, (TTIME (J), J=1, JTOTAL)
    WRITE (6,113) I, (YTEMP (J, I), J=1, JTOTAL)
    WRITE (6,114) I, I
115 CONTINUE
    END IF
110 FORMAT (5X,'NTEMP = ',I3,',')

```

```

111 FORMAT (5X, 'NTMP (' , I2, ') = ' , I4, ', ', 5X, 'ITMAX (' , I2, ') = ' , I5, ', ')
112 FORMAT (5X, 'XTEMP (1, ' , I2, ') = ' , /7 (F8.2, ', ')
113 FORMAT (5X, 'YTEMP (1, ' , I2, ') = ' , /7 (F8.2, ', ')
114 FORMAT (5X, 'TGAIN (' , I2, ') = 1.0, ', 5X, 'ITST (' , I2, ') = 2, ')

```

C

C. FLWN

```

IF (NFLWN.NE.0) THEN
WRITE (6, 210)NFLWN
DO 215 I=1, NFLWN
WRITE (6, 211) I, NFPAS (I), I, NFNNET (I), I, JTOTAL
WRITE (6, 212) I, (TTIME (J), J=1, JTOTAL)
WRITE (6, 213) I, (YFLWN (J, I), J=1, JTOTAL)
WRITE (6, 214) I, I
215 CONTINUE
END IF
210 FORMAT (5X, 'NFLWN = ' , I3, ', ')
211 FORMAT (5X, 'NFPAS (' , I2, ') = ' , I4, ', ', 5X, 'NFNNET (' , I2, ') = ' , I5, ', '
*      , 5X, 'IFNMAX (' , I2, ') = ' , I5, ', ')
212 FORMAT (5X, 'XFLWN (1, ' , I2, ') = ' , /7 (F8.2, ', ')
213 FORMAT (5X, 'YFLWN (1, ' , I2, ') = ' , /7 (F8.2, ', ')
214 FORMAT (5X, 'FNGAIN (' , I2, ') = 1.0, ', 5X, 'IFNST (' , I2, ') = 2, ')

```

C

C. PRSN

```

IF (NPRSN.NE.0) THEN
WRITE (6, 310)NPRSN
DO 315 I=1, NPRSN
WRITE (6, 311) I, NPNNOD (I), I, NPNNET (I), I, JTOTAL
WRITE (6, 312) I, (TTIME (J), J=1, JTOTAL)
WRITE (6, 313) I, (YPRSN (J, I), J=1, JTOTAL)
WRITE (6, 314) I, I
315 CONTINUE
END IF
310 FORMAT (5X, 'NPRSN = ' , I3, ', ')
311 FORMAT (5X, 'NPNNOD (' , I2, ') = ' , I4, ', ', 5X, 'NPNNET (' , I2, ') = ' , I5, ', '
*      , 5X, 'IPNMAX (' , I2, ') = ' , I5, ', ')
312 FORMAT (5X, 'XPRSN (1, ' , I2, ') = ' , /7 (F8.2, ', ')
313 FORMAT (5X, 'YPRSN (1, ' , I2, ') = ' , /7 (F8.2, ', ')
314 FORMAT (5X, 'PNGAIN (' , I2, ') = 1.0, ', 5X, 'IPNST (' , I2, ') = 2, ')

```

C

C. PMPN

```

IF (NPMPN.NE.0) THEN
WRITE (6, 410)NPMPN
DO 415 I=1, NPMPN
WRITE (6, 411) I, NHNNUM (I), I, NHNPMP (I), I, JTOTAL
WRITE (6, 412) I, (TTIME (J), J=1, JTOTAL)
WRITE (6, 413) I, (YPMPN (J, I), J=1, JTOTAL)
WRITE (6, 414) I, I
415 CONTINUE
END IF
410 FORMAT (5X, 'NPMPN = ' , I3, ', ')
411 FORMAT (5X, 'NHNNUM (' , I2, ') = ' , I4, ', ', 5X, 'NHNPMP (' , I2, ') = ' , I5, ', '

```

```

*          , 5X, 'IHNMAX (' , I2, ' ) = ' , I5, ' , ' )
412 FORMAT (5X, 'XPMPN (1, ' , I2, ' ) = ' , /7 (F8.2, ' , ' ) )
413 FORMAT (5X, 'YPMPN (1, ' , I2, ' ) = ' , /7 (F8.2, ' , ' ) )
414 FORMAT (5X, 'HNGAIN (' , I2, ' ) = 1.0, ' , 5X, 'IHNST (' , I2, ' ) = 2, ' )
C
C. VLVN
      IF (NVLVN.NE.0) THEN
      WRITE (6, 510) NVLVN
      DO 515 I=1, NVLVN
          WRITE (6, 511) I, NVNNUM (I), I, NVNVLV (I), I, JTOTAL
          WRITE (6, 512) I, (TTIME (J), J=1, JTOTAL)
          WRITE (6, 513) I, (YVLVN (J, I), J=1, JTOTAL)
          WRITE (6, 514) I, I
515 CONTINUE
      END IF
510 FORMAT (5X, 'NVLVN = ' , I3, ' , ' )
511 FORMAT (5X, 'NVNNUM (' , I2, ' ) = ' , I4, ' , ' , 5X, 'NVNVLV (' , I2, ' ) = ' , I5, ' , '
*          , 5X, 'IVNMAX (' , I2, ' ) = ' , I5, ' , ' )
512 FORMAT (5X, 'XVLVN (1, ' , I2, ' ) = ' , /7 (F8.2, ' , ' ) )
513 FORMAT (5X, 'YVLVN (1, ' , I2, ' ) = ' , /7 (F8.2, ' , ' ) )
514 FORMAT (5X, 'VNGAIN (' , I2, ' ) = 1.0, ' , 5X, 'IVNST (' , I2, ' ) = 2, ' )
C
      WRITE (6, 300)
      STOP
100 FORMAT (20 (F9.3, 1X))
101 FORMAT (F9.3)
150 FORMAT (1X, '&NMAAA2')
300 FORMAT (1X, '&END')
      END
      SUBROUTINE SODIUM ( IJ, TX, X )
C
C      PURPOSE
C      SODIUM PROPERTIES FROM TEMPERATURE
C
C      USAGE
C      CALL SODIUM (IJ, TX, X)
C
C      DESCRIPTION PARAMETER AND I/O
C      IJ      - CLASSIFICATION ;I
C              = 0 TEMPERATURE ----> ENTALPH ( KCAL/KG )
C              = 1 TEMPERATURE ----> SPECIFIC HEAT ( KCAL/KG*C )
C              = 2 TEMPERATURE ----> SPECIFIC DENSITY ( KG/M**3 )
C              = 3 TEMPERATURE ----> PRANDTL NUMBER ( - )
C              = 4 TEMPERATURE ----> KINETIC VISCOSITY ( M**2/SEC )
C              = 5 TEMPERATURE ----> THERMAL CONDUCTIVITY ( KCAL/(M*SEC*C) )
C      TX      - TEMPERATURE (C) ;I
C      X      - OUTPUT          ;0
C
      IF ( IJ .EQ. 0 ) THEN
          X = 3.685E-08*TX**3-6.9343E-05*TX**2+0.343253*TX
      ELSE IF ( IJ .EQ. 1 ) THEN

```

```
      X = 0.34325 - 1.38686E-04*TX + 1.1055E-07*TX**2
ELSE IF ( IJ .EQ. 2 ) THEN
      X = 1000.0*(0.9500 - 2.2977E-04*TX - 1.4605E-08*TX**2
*      + 5.638E-12*TX**3)
ELSE IF ( IJ .EQ. 3 ) THEN
      X = 0.008901 - 1.167E-5*TX + 6.67E-09*TX**2
ELSE IF ( IJ .EQ. 4 ) THEN
      X = 6.27E-07 - 9.5E-10*TX + 5.56E-13*TX**2
ELSE IF ( IJ .EQ. 5 ) THEN
      X = 0.02388*(0.9292 - 5.8905E-04*TX + 1.1724E-07*TX**2)
END IF
RETURN
END
```

Appendix-A 2

「試験データ変換処理プログラム実行用コマンドプロシジャリスト」

```
PROC 0
/*
/*          CALCULATION
/*
FREE ALL
/*DEL @FT06
ATTR IN INPUT
/*ATTR F06 BL (800) LR (80) REC (F B) DS (PS)
ALLOC F (FT05F001) DA (CONVERT.DATA (M2STEST)) US (IN) SHR REU
/*ALLOC F (FT06F001) UNIT (DASD) DA (@FT06) SP (10, 5) T US (F06) -
/*NEW CAT REL REU
ALLOC F (FT06F001) DA (CONVERT.DATA (TEMP)) SHR
ALLOC F (FT08F001) DA (@TEMP.DATA) SHR US (IN) REU
CALL CONVERT.LOAD (INTERM2S)
WRITE CALCULATE COND CODE = &LASTCC
FREE ALL
END
```


Appendix-B 1

「ポストデータ処理プログラムリスト」

```

C
C   MAKE SUPER COPD DATA
C   MONJYU ==> EDIT COPD
C
PARAMETER (NMAX=25)
DIMENSION VAL(2000,NMAX)
DIMENSION NPTEMP(20),NQTEMP(10,20),
*       NPFLWN(10),NQFLWN(10,10),NRFLWN(10,10),
*       NPPRSN(5),NQPRSN(10,5),NRPRSN(10,5),
*       NPPMPN(5),NQPMPN(10,5),NRPMPN(10,5),
*       NPVLVN(5),NQVLVN(10,5),NRVLVN(10,5),
*       IBTEMP(10,20),INTEMP(10,20),
*       IBFLWN(10,10),INFLWN(10,10),
*       IBPRSN(10,5),INPRSN(10,5),
*       IBPMPN(10,5),INPMPN(10,5),
*       IBVLVN(10,5),INVLVN(10,5)
REAL*8   TTIME(2000),
*       TTEMP(2000,10,20),
*       TFLWN(2000,10,10),
*       TPRSN(1,1,1),
*       TPMPN(2000,1,1),
*       TVLVN(2000,1,1)
C
NAMELIST/POOLI/ IBTEMP,INTEMP,
*       IBFLWN,INFLWN,
*       IBPRSN,INPRSN,
*       IBPMPN,INPMPN,
*       IBVLVN,INVLVN,
*       TIMES
C
NAMELIST/POOLO/ NOTEMP,NPTEMP,NQTEMP,
*       NOFLWN,NPFLWN,NQFLWN,NRFLWN,
*       NOPRSN,NPPRSN,NQPRSN,NRPRSN,
*       NOPMPN,NPPMPN,NQPMPN,NRPMPN,
*       NOVLVN,NPVLVN,NQVLVN,NRVLVN
C
DATA   JBLOCK,JTOTAL,JREC1,NUMREC/0,0,0,25/
DATA   NOTEMP,NPTEMP,NQTEMP/0,20*0,200*0/
DATA   NOFLWN,NPFLWN,NQFLWN,NRFLWN/0,10*0,100*0,100*0/
DATA   NOPRSN,NPPRSN,NQPRSN,NRPRSN/0,5*0,50*0,50*0/
DATA   NOPMPN,NPPMPN,NQPMPN,NRPMPN/0,5*0,50*0,50*0/
DATA   NOVLVN,NPVLVN,NQVLVN,NRVLVN/0,5*0,50*0,50*0/
DATA   TTIME/2000*0.0/
DATA   TTEMP/400000*0.0/
DATA   TFLWN/200000*0.0/
DATA   TPRSN/0.0/
DATA   TPMPN/2000*0.0/
DATA   TVLVN/2000*0.0/
C
REWIND5
READ(5,POOLI)

```

```

        REWIND5
        READ (5, POOLO)
C
C   READ (8)
        READ (8, *)N
        JBLOCK = 0
        JTOTAL = 0
        NDUMMY = 0
1   READ (8, *, END=2) TIME
        IF (TIME.LT. TIMES) THEN
            NDUMMY = NDUMMY + 1
            GO TO 1
        END IF
        JTOTAL = JTOTAL + 1
        IF (JTOTAL.EQ.1) TIMES = TIME
        TTIME (JTOTAL) = TIME - TIMES
        WRITE (6, *) TTIME (JTOTAL)
        IF (MOD (JTOTAL, NUMREC).EQ.0) JBLOCK = JBLOCK + 1
        IF (JTOTAL.GE.2000) GO TO 2
        GO TO 1
2   CONTINUE
        IF (MOD (JTOTAL, NUMREC).NE.0) JBLOCK = JBLOCK + 1
C
        DO 10 I1=1, NOTEMP
        DO 10 I2=1, NPTEMP (I1)
        REWIND 8
C   DO 11 I5=1, IBTEMP (I2, I1) + (IBTEMP (I2, I1) - 1) * JTOTAL
C 11 READ (8, *) DUMMY
        DO 11 I5=1, NDUMMY+1
11  READ (8, *) DUMMY
        DO 10 I3=1, JTOTAL
        READ (8, *, END=2) (VAL (I3, I4), I4=1, N)
        TTEMP (I3, I2, I1) = VAL (I3, INTEMP (I2, I1))
        WRITE (6, *) TTIME (I3), TTEMP (I3, I2, I1)
10  CONTINUE
C
        DO 20 I1=1, NOFLWN
        DO 20 I2=1, NPFLWN (I1)
        REWIND 8
C   DO 21 I5=1, IBFLWN (I2, I1) + (IBFLWN (I2, I1) - 1) * JTOTAL
C 21 READ (8, *) DUMMY
        DO 21 I5=1, NDUMMY+1
21  READ (8, *) DUMMY
        DO 20 I3=1, JTOTAL
        READ (8, *, END=2) (VAL (I3, I4), I4=1, N)
        TFLWN (I3, I2, I1) = VAL (I3, INFLWN (I2, I1))
        WRITE (6, *) TTIME (I3), TFLWN (I3, I2, I1)
20  CONTINUE
C
        DO 30 I1=1, NOPRSN
        DO 30 I2=1, NPPRSN (I1)

```

```

      REWIND 8
C     DO 31 I5=1, IBPRSN (I2, I1) + (IBPRSN (I2, I1) - 1) * JTOTAL
C 31  READ (8, *) DUMMY
      DO 31 I5=1, NDUMMY+1
      31  READ (8, *) DUMMY
      DO 30 I3=1, JTOTAL
      READ (8, *, END=2) (VAL (I3, I4), I4=1, N)
      TPRSN (I3, I2, I1) = VAL (I3, INPRSN (I2, I1))
C     WRITE (6, *) TPRSN (I3, I2, I1)
      30  CONTINUE
C
      DO 40 I1=1, NOPMPN
      DO 40 I2=1, NPPMPN (I1)
      REWIND 8
C     DO 41 I5=1, IBPMPN (I2, I1) + (IBPMPN (I2, I1) - 1) * JTOTAL
C 41  READ (8, *) DUMMY
      DO 41 I5=1, NDUMMY+1
      41  READ (8, *) DUMMY
      DO 40 I3=1, JTOTAL
      READ (8, *, END=2) (VAL (I3, I4), I4=1, N)
      TPMPN (I3, I2, I1) = VAL (I3, INPMPN (I2, I1))
C     WRITE (6, *) TPMPN (I3, I2, I1)
      40  CONTINUE
C
      DO 50 I1=1, NOVLVN
      DO 50 I2=1, NPVLVN (I1)
      REWIND 8
C     DO 51 I5=1, IBVLVN (I2, I1) + (IBVLVN (I2, I1) - 1) * JTOTAL
C 51  READ (8, *) DUMMY
      DO 51 I5=1, NDUMMY+1
      51  READ (8, *) DUMMY
      DO 50 I3=1, JTOTAL
      READ (8, *, END=2) (VAL (I3, I4), I4=1, N)
      TVLVN (I3, I2, I1) = VAL (I3, INVLVN (I2, I1))
C     WRITE (6, *) TVLVN (I3, I2, I1)
      50  CONTINUE
C
      IF (JREC1.EQ.0) THEN
      DO 120 I = 1, NOTEMP
      JREC1 = JREC1 + NPTEMP (I)
120   CONTINUE
      DO 130 I = 1, NOFLWN
      JREC1 = JREC1 + NPFLWN (I)
130   CONTINUE
      DO 140 I = 1, NOPRSN
      JREC1 = JREC1 + NPPRSN (I)
140   CONTINUE
      DO 150 I = 1, NOPMPN
      JREC1 = JREC1 + NPPMPN (I)
150   CONTINUE
      DO 160 I = 1, NOVLVN

```

```

          JREC1 = JREC1+NPVLVN(I)
160    CONTINUE
      ENDIF
C
C
C ----WRITE DATA FT09----
C
      WRITE(9) JBLOCK, JTOTAL, JREC1, NUMREC
      WRITE(9) NOTEMP, NPTEMP, NQTEMP
      WRITE(9) NOFLWN, NPFLWN, NQFLWN, NRFLWN
      WRITE(9) NOPRSN, NPPRSN, NQPRSN, NRPRSN
      WRITE(9) NOPMPN, NPPMPN, NQPMPN, NRPMPN
      WRITE(9) NOVLVN, NPVLVN, NQVLVN, NRVLVN
C
      WRITE(9) JBLOCK, JTOTAL, JREC1, NUMREC
C
C
C ----WRITE DATA FT10----
C
      DO 200 ICOUNT=1, JBLOCK
          WRITE(10) (TTIME(I), I=(ICOUNT-1)*NUMREC+1, ICOUNT*NUMREC)
          DO 210 I=1, NOTEMP
              DO 210 J=1, NPTEMP(I)
                  WRITE(10) (TEMP(K, J, I),
*                   K=(ICOUNT-1)*NUMREC+1, ICOUNT*NUMREC)
210    CONTINUE
          DO 220 I=1, NOFLWN
              DO 220 J=1, NPFLWN(I)
                  WRITE(10) (TFLWN(K, J, I),
*                   K=(ICOUNT-1)*NUMREC+1, ICOUNT*NUMREC)
220    CONTINUE
          DO 230 I=1, NOPRSN
              DO 230 J=1, NPPRSN(I)
                  WRITE(10) (TPRSN(K, J, I),
*                   K=(ICOUNT-1)*NUMREC+1, ICOUNT*NUMREC)
230    CONTINUE
          DO 240 I=1, NOPMPN
              DO 240 J=1, NPPMPN(I)
                  WRITE(10) (TPMPN(K, J, I),
*                   K=(ICOUNT-1)*NUMREC+1, ICOUNT*NUMREC)
240    CONTINUE
          DO 250 I=1, NOVLVN
              DO 250 J=1, NPVLVN(I)
                  WRITE(10) (TVLVN(K, J, I),
*                   K=(ICOUNT-1)*NUMREC+1, ICOUNT*NUMREC)
250    CONTINUE
200    CONTINUE
          WRITE(6, 110) JBLOCK, JTOTAL, JREC1, NUMREC
          STOP
100    FORMAT(20(F9.3, 1X))
110    FORMAT(1H , 80(' '))

```

```
*      ,/1H , ' OUTPUT DATA. .... JBLOCK=' , 15, ' '
*                                     , ' JTOTAL=' , 15, ' '
*                                     , ' JREC1=' , 15, ' '
*                                     , ' NUMBER=' , 15, ' '
*      ,/1H , 80('*'))
END
```

Appendix-B 2

「ポストデータ処理プログラム実行コマンドプロシジャリスト」

```
PROC 0
/*
/*          CALCULATION
/*
DEL @FT06
FREE ALL
ATTR IN INPUT
ATTR TI BL (6850) LR (137) REC (F B A) DS (PS)
ALLOC F (FT05F001) DA (SKSMNJ. DATA (M2E)) SHR REU
/*ALLOC F (FT06F001) UNIT (DASD) DA (@FT06) -
/*SP (50, 50) T US (TI) NEW CAT REL REU
ALLOC F (FT06F001) DUMMY
ALLOC F (FT08F001) DA (SSTXLS) SHR US (IN) REU
ALLOC F (FT09F001) UNIT (DASD) DA (@09. SSTXLSE) SP (10, 5) T -
NEW CAT REL REU
ALLOC F (FT10F001) UNIT (DASD) DA (@10. SSTXLSE) SP (10, 5) T -
NEW CAT REL REU
CALL SKSMNJ. LOAD (INTERM2E)
WRITE CALCULATE COND CODE = &LASTCC
FREE ALL
END
```


Appendix-C

「ポストデータ処理プログラムリスト」

SUBROUTINE EDITIT

C
C
C
C

FOR OUTPUT TITLE

COMMON /OPTONI/ ISELCT, INPUTD, MACHIN, JREC1
COMMON /TITLEI/ TITLE1(8), TITLE2(8), CMENT1(20), CMENT2(20)

C
C

DIMENSION CHARCA(10,2), CHARCC(10,2), CHARCD(10,2), CHARCE(10,2),
1 CHARCG(10,2), CHARCH(10,2), CHARCL(10,2), CHARCM(10,2),
2 CHARCN(10,2), CHARCO(10,2), CHARCP(10,2), CHARCR(10,2),
3 CHARCS(10,2), CHARCT(10,2), CHARCU(10,2), CHARCV(10,2),
4 CHARCY(10,2), CHARCB(10,2)

C

DATA CHARCA / ' A', ' AA', ' AAA', ' AAA ', ' AA ', ' AAAA', ' AAAA',
1 ' AA ', ' AA ', ' AA ', ' A ', ' AA ', ' AAA ', ' AAA',
2 ' AA', ' AAAA', ' AAAA', ' AA', ' AA', ' AA' /
DATA CHARCC / ' CCC', ' CCCC', ' CC ', ' CC ', ' CC ', ' CC ', ' CC ',
1 ' CC ', ' CCCC', ' CCC', ' CCC ', ' CCCC', ' CC',
2 ' ', ' ', ' ', ' ', ' CC', ' CCCC', ' CCC' /
DATA CHARCD / ' DDDD', ' DDDD', ' DD ', ' DD ', ' DD ', ' DD ', ' DD ',
1 ' DD ', ' DDDD', ' DDDD', ' DDD ', ' DDDD', ' DDD', ' DD',
2 ' DD', ' DD', ' DD', ' DDD', ' DDDD', ' DDD' /
DATA CHARCE / ' EEEE', ' EEEE', ' EE ', ' EE ', ' EEEE', ' EEEE', ' EE ',
1 ' EE ', ' EEEE', ' EEEE', ' EEEE', ' EEEE',
2 ' EEEE', ' EEEE', ' ', ' ', ' EEEE', ' EEEE' /
DATA CHARCG / ' GGG', ' GGGG', ' GG ', ' GG ', ' GG ', ' GG ', ' GG ',
1 ' GG ', ' GGGG', ' GGG', ' GGG ', ' GGGG', ' GG',
2 ' GGGG', ' GGGG', ' GG', ' GG', ' GGGG', ' GGG' /
DATA CHARCH / ' HH ', ' HH ', ' HH ', ' HH ', ' HHHH', ' HHHH', ' HH ',
1 ' HH ', ' HH ', ' HH ', ' HH ', ' HH', ' HH', ' HH', ' HH',
2 ' HHHH', ' HHHH', ' HH', ' HH', ' HH', ' HH' /
DATA CHARCL / ' LL ', ' LL ', ' LL ', ' LL ', ' LL ', ' LL ', ' LL ',
1 ' LL ', ' LLLL', ' LLLL',
2 ' ', ' ', ' ', ' ', ' LLLL', ' LLLL' /
DATA CHARCM / ' M ', ' MM ', ' MMM', ' MMMM', ' MM M', ' MM M', ' MM ',
1 ' MM ', ' MM ', ' MM ', ' M', ' MM', ' MMM', ' MMMM',
2 ' M MM', ' M MM', ' MM', ' MM', ' MM', ' MM' /
DATA CHARCN / ' NN ', ' NN ', ' NNN', ' NNNN', ' NNNN', ' NN N', ' NN ',
1 ' NN ', ' NN ', ' NN ', ' NN', ' NN', ' NN', ' NN',
2 ' N NN', ' NNNN', ' NNNN', ' NNN', ' NN', ' NN' /
DATA CHARCO / ' OOO', ' OOOO', ' OOO ', ' OO ', ' OO ', ' OO ', ' OO ',
1 ' OOO', ' OOOO', ' OOO', ' OOO ', ' OOOO', ' OOO', ' OO',
2 ' OO', ' OO', ' OO', ' OOO', ' OOOO', ' OOO' /
DATA CHARCP / ' PPPP', ' PPPP', ' PP ', ' PP ', ' PPPP', ' PPPP', ' PP ',
1 ' PP ', ' PP ', ' PP ', ' PPP', ' PPPP', ' PP', ' PP',
2 ' PPPP', ' PPP', ' ', ' ', ' ', ' ', ' ' /
DATA CHARCR / ' RRRR', ' RRRR', ' RR ', ' RR ', ' RRRR', ' RRRR', ' RR R',
1 ' RR ', ' RR ', ' RR ', ' RRR', ' RRRR', ' RR', ' RR',
2 ' RRRR', ' RRR', ' R ', ' RR ', ' RR ', ' RR' /

```

DATA CHARCS / 'SSS', 'SSSS', 'SS ', 'SS ', 'SSSS', 'SSS', ' ',
1          'SS ', 'SSSS', 'SSS', 'SSS ', 'SSSS', 'SS', ' ',
2          'SSS ', 'SSSS', 'SS', 'SS', 'SSSS', 'SSS' /
DATA CHARCT / 'TTT', 'TTTT', 'T', 'T', 'T', 'T', 'T',
1          'T', 'T', 'T', 'T', 'TTTT', 'TTTT', 'T', 'T',
2          'T', 'T', 'T', 'T', 'T', 'T', 'T' /
DATA CHARCU / 'UU', 'UU', 'UU', 'UU', 'UU', 'UU', 'UU',
1          'UU', 'UUU', 'UUU', 'UU', 'UU', 'UU', 'UU',
2          'UU', 'UU', 'UU', 'UU', 'UUU', 'UUU' /
DATA CHARCV / 'VV', 'VV', 'VV', 'VV', 'VV', 'VV', 'VV',
1          'VVV', 'VV', 'V', 'VV', 'VV', 'VV', 'VV',
2          'VV', 'VV', 'VVV', 'VVV', 'VV', 'V' /
DATA CHARCY / 'YY', 'YY', 'YY', 'YY', 'Y', 'Y', 'Y',
1          'Y', 'Y', 'Y', 'YY', 'YY', 'YY', 'YY',
2          'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y' /
DATA CHARCB /
1          ' ', ' ', ' ', ' ', ' ', ' ', ' ',
2          ' ', ' ', ' ', ' ', ' ', ' ', ' ' /

```

C
C

```

WRITE (6, 1000)
WRITE (6, 1100)
GO TO (1, 1, 3, 4, 5, 6), ISELCT

```

C DATA CHANGE

```

1 CONTINUE
DO 10 I=1, 10
WRITE (6, 1200) CHARCD (I, 1), CHARCD (I, 2), CHARCA (I, 1), CHARCA (I, 2),
1          CHARCT (I, 1), CHARCT (I, 2), CHARCA (I, 1), CHARCA (I, 2),
2          CHARCB (I, 1), CHARCB (I, 2), CHARCC (I, 1), CHARCC (I, 2),
2          CHARCH (I, 1), CHARCH (I, 2), CHARCA (I, 1), CHARCA (I, 2),
3          CHARCN (I, 1), CHARCN (I, 2), CHARCG (I, 1), CHARCG (I, 2),
4          CHARCE (I, 1), CHARCE (I, 2)

```

```

10 CONTINUE
GO TO 100

```

C TAPE OUT

```

3 CONTINUE
DO 30 I=1, 10
WRITE (6, 1300) CHARCT (I, 1), CHARCT (I, 2), CHARCA (I, 1), CHARCA (I, 2),
1          CHARCP (I, 1), CHARCP (I, 2), CHARCE (I, 1), CHARCE (I, 2),
2          CHARCB (I, 1), CHARCB (I, 2), CHARCO (I, 1), CHARCO (I, 2),
3          CHARCU (I, 1), CHARCU (I, 2), CHARCT (I, 1), CHARCT (I, 2)

```

```

30 CONTINUE
GO TO 100

```

C DATA PLOT

```

4 CONTINUE
DO 40 I=1, 10
WRITE (6, 1400) CHARCD (I, 1), CHARCD (I, 2), CHARCA (I, 1), CHARCA (I, 2),
1          CHARCT (I, 1), CHARCT (I, 2), CHARCA (I, 1), CHARCA (I, 2),
2          CHARCB (I, 1), CHARCB (I, 2), CHARCP (I, 1), CHARCP (I, 2),
3          CHARCL (I, 1), CHARCL (I, 2), CHARCO (I, 1), CHARCO (I, 2),
4          CHARCT (I, 1), CHARCT (I, 2)

```

```

40 CONTINUE
   GO TO 100
C OVER PLOT
   5 CONTINUE
   DO 50 I=1,10
   WRITE (6,1400) CHARCO (I, 1), CHARCO (I, 2), CHARCV (I, 1), CHARCV (I, 2),
1       CHARCE (I, 1), CHARCE (I, 2), CHARCR (I, 1), CHARCR (I, 2),
2       CHARCB (I, 1), CHARCB (I, 2), CHARCP (I, 1), CHARCP (I, 2),
3       CHARCL (I, 1), CHARCL (I, 2), CHARCO (I, 1), CHARCO (I, 2),
4       CHARCT (I, 1), CHARCT (I, 2)
50 CONTINUE
   GO TO 100
C SUMMARY
   6 CONTINUE
   DO 60 I=1,10
   WRITE (6,1500) CHARCS (I, 1), CHARCS (I, 2), CHARCU (I, 1), CHARCU (I, 2),
1       CHARCM (I, 1), CHARCM (I, 2), CHARCM (I, 1), CHARCM (I, 2),
2       CHARCA (I, 1), CHARCA (I, 2), CHARCR (I, 1), CHARCR (I, 2),
3       CHARCY (I, 1), CHARCY (I, 2)
60 CONTINUE
100 CONTINUE
   WRITE (6,1600)
   IF ( MACHIN.NE.1 ) GO TO 70
   WRITE (6,1700) (TITLE1 (I), I=1, 4 ), (TITLE2 (L), L=1, 4 )
   GO TO 200
70 CONTINUE
   WRITE (6,1800) (TITLE1 (I), I=1, 8 ), (TITLE2 (L), L=1, 8 )
200 CONTINUE
   WRITE (6,2000) (CMEN1 (I), I=1, 20), (CMEN2 (L), L=1, 20)
   IF ( ISELCT.EQ.5 ) WRITE (6,2100)
C
   RETURN
C
C
1000 FORMAT (1H1/
1 1H ,10X,27('E'), 3X,24('D'), 6X,5('I'), 3X,26('T')/
2 1H ,10X,27('E'), 3X,25('D'), 5X,5('I'), 3X,26('T')/
3 1H ,10X,27('E'), 3X,26('D'), 4X,5('I'), 3X,26('T')/
4 1H ,10X, 4('E'),26X, 4('D'),18X,5('D'), 3X, 5('I'),14X,4('T')/
5 1H ,10X, 4('E'),26X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
6 1H ,10X, 4('E'),26X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
7 1H ,10X, 4('E'),26X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
8 1H ,10X,27('E'), 3X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
9 1H ,10X,27('E'), 3X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
A 1H ,10X,27('E'), 3X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T'))
1100 FORMAT (
1 1H ,10X, 4('E'),26X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
2 1H ,10X, 4('E'),26X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
3 1H ,10X, 4('E'),26X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
4 1H ,10X, 4('E'),26X, 4('D'),19X,4('D'), 3X, 5('I'),14X,4('T')/
5 1H ,10X, 4('E'),26X, 4('D'),18X,5('D'), 3X, 5('I'),14X,4('T')/

```

```

6 1H , 10X, 27 ('E'), 3X, 26 ('D'), 4X, 5 ('I'), 14X, 4 ('T')/
7 1H , 10X, 27 ('E'), 3X, 25 ('D'), 5X, 5 ('I'), 14X, 4 ('T')/
8 1H , 10X, 27 ('E'), 3X, 24 ('D'), 6X, 5 ('I'), 14X, 4 ('T')/
9 1H0/1H0)
1200 FORMAT (1H , 10X, 11 (2A4, 2X))
1300 FORMAT (1H , 30X, 8 (2A4, 2X))
1400 FORMAT (1H , 20X, 9 (2A4, 2X))
1500 FORMAT (1H , 30X, 7 (2A4, 2X))
1600 FORMAT (1H0/1H0 )
1700 FORMAT (1H , 14X, ' +', 90 (' -'), ' +'/1H , 14X, ' I', 90X, ' I'/1H , 14X,
1 ' I', 5X, 4A8, 53X, ' I'/1H , 14X, ' I', 5X, 4A8, 53X, ' I')
1800 FORMAT (1H , 14X, ' +', 90 (' -'), ' +'/1H , 14X, ' I', 90X, ' I'/1H , 14X,
1 ' I', 5X, 8A4, 53X, ' I'/1H , 14X, ' I', 5X, 8A4, 53X, ' I')
2000 FORMAT (1H , 14X, ' I', 90X, ' I'/1H , 14X, ' I', 5X, 20A4, 5X, ' I'/1H , 14X,
1 ' I', 5X, 20A4, 5X, ' I'/1H , 14X, ' I', 90X, ' I'/1H , 14X,
2 ' +', 90 (' -'), ' +')
2100 FORMAT (1H1)
C
END
SUBROUTINE INDEX ( JPOOL, JDIMX, JDIMY, INDX )
C
C CALCULATE INDX NO. FROM JPOOL, JDIMX, JDIMY
C
C INPUT
C JPOOL ----- NUMBER OF POOL VARIABLE
C JDIMX----- X DIMENSION NO. OF POOL VARIABLE
C JDIMY----- Y DIMENSION NO. OF POOL VARIABLE
C OUTPUT
C INDX ----- INDEX NUMBER
C
C
C COMMON/ P / NOTEMP, NPTEMP (20), NQTEMP (10, 20)
* , NOFLWN, NPFLWN (10), NQFLWN (10, 10), NRFLWN (10, 10)
* , NOPRSN, NPPRSN (5) , NQPRSN (10, 5) , NRPRSN (10, 5)
* , NOPMPN, NPPMPN (5) , NQPMPN (10, 5) , NRPMPN (10, 5)
* , NOVLVN, NPVLVN (5) , NQVLVN (10, 5) , NRVLVN (10, 5)
C
C
C
IF ( JDIMX .EQ. -1) GO TO 800
IF ( JPOOL .LE. 0 ) GO TO 900
C
C
C
C ---POOL VARIABLE TEMP (JPOOL=1) ---
INDX = 0
IF ( JPOOL .GT. 1 ) GO TO 2
C
DO 1 I = 1, NOTEMP
DO 1 J = 1, NPTEMP (I)

```

```

      INDX = INDX + 1
      IF( JDIMX .EQ. NQTEMP(J,I) ) RETURN
1 CONTINUE
      GO TO 900
C
C
C ---POOL VALIABLE FLWN (JPOOL=2) ---
      2 CONTINUE
      DO 19 I = 1,NOTEMP
19 INDX = INDX + NPTEMP(I)
      IF( JPOOL .GT. 2 ) GO TO 3
C
      DO 20 I = 1,NOFLWN
      DO 20 J = 1,NPFLWN(I)
      INDX = INDX + 1
      IF( JDIMX .NE. NQFLWN(J,I) ) GO TO 20
      IF( JDIMY .EQ. NRFLWN(J,I) ) RETURN
20 CONTINUE
      GO TO 900
C
C
C ---POOL VALIABLE PRSN (JPOOL=3)
      3 CONTINUE
      DO 29 I = 1,NOFLWN
29 INDX = INDX + NPFLWN(I)
      IF( JPOOL .GT. 3 ) GO TO 4
C
      DO 30 I = 1,NOPRSN
      DO 30 J = 1,NPPRSN(I)
      INDX = INDX + 1
      IF( JDIMX .NE. NQPRSN(J,I) ) GO TO 30
      IF( JDIMY .EQ. NRPRSN(J,I) ) RETURN
30 CONTINUE
      GO TO 900
C
C
C ---POOL VALIABLE PMPN (JPOOL = 4)
      4 CONTINUE
      DO 39 I = 1,NOPRSN
39 INDX = INDX + NPPRSN(I)
      IF( JPOOL .GT. 4 ) GO TO 5
C
      DO 40 I = 1,NOPMPN
      DO 40 J = 1,NPPMPN(I)
      INDX = INDX + 1
      IF( JDIMX .NE. NQPMPN(J,I) ) GO TO 40
      IF( JDIMY .EQ. NRPMPN(J,I) ) RETURN
40 CONTINUE
      GO TO 900
C
C

```

```

C ---POOL VALIABLE VLVN (JPOOL = 5)
  5 CONTINUE
    DO 49 I = 1, NOPMPN
49  INDX = INDX + NPPMPN(I)
    IF( JPOOL .GT.5 ) GO TO 900
C
    DO 50 I = 1, NOVLVN
    DO 50 J = 1, NPVLVN(I)
      INDX = INDX + 1
      IF( JDIMX .NE. NQVLVN(J, I) ) GO TO 50
      IF( JDIMY .EQ. NRVLVN(J, I) ) RETURN
50  CONTINUE
    GO TO 900
C
C
C ---DATA INPUT FROM IUNIT2 AND TIME DATA INPUT FROM IUNIT1--
800 CONTINUE
    INDX = -1
    RETURN
C
C
C ---CAN'T DESIDE INDX ---
900 CONTINUE
    WRITE(6, 910) JPOOL, JDIMX, JDIMY
    INDX = -999
    RETURN
C
C
910 FORMAT(1H0, 10X,
1   '***INFORMATION OF NEXT DATA IS NOT CORRECT***' /
2   19X, 'LOOP =', I2, 5X, 'INDEX =', 2I4//)
C
    END
    SUBROUTINE INPUTS
C
    COMMON/ P      /  NOTEMP, NPTEMP (20), NQTEMP (10, 20)
*                   , NOFLWN, NPFLWN (10), NQFLWN (10, 10), NRFLWN (10, 10)
*                   , NOPRSN, NPPRSN (5) , NQPRS (10, 5) , NRPRS (10, 5)
*                   , NOPMPN, NPPMPN (5) , NQPMPN (10, 5) , NRPMPN (10, 5)
*                   , NOVLVN, NPVLVN (5) , NQVLVN (10, 5) , NRVLVN (10, 5)
    COMMON /NUMBEI/ JTOTAL, JBLOCK
    COMMON /UNITI / IUNIT1, IUNIT2, MINSEC
    COMMON /OPTONI/ ISELCT, INPUTD, MACHIN, JREC1
C
C
C --INPUT FROM IUNIT2--
C
    REWIND IUNIT2
    READ( IUNIT2 ) JBLOCK, JTOTAL, JREC1
C

```

```

      READ ( IUNIT2 ) NOTEMP, NPTEMP, NQTEMP
      READ ( IUNIT2 ) NOFLWN, NPFLWN, NQFLWN, NRFLWN
      READ ( IUNIT2 ) NOPRSN, NPPRSN, NQPRSN, NRPRSN
      READ ( IUNIT2 ) NOPMPN, NPPMPN, NQPMPN, NRPMPN
      READ ( IUNIT2 ) NOVLVN, NPVLVN, NQVLVN, NRVLVN
C
C
      IF ( ISELCT.NE.5 ) WRITE (6, 110)
      WRITE (6, 120) IUNIT2
      WRITE (6, 130) JTOTAL, JBLOCK, JREC1
C
      WRITE (6, 140) NOTEMP
      DO 240 I=1, NOTEMP
240 WRITE (6, 145) I, NPTEMP (I), I, (NQTEMP (J, I), J=1, NPTEMP (I))
C
      WRITE (6, 150) NOFLWN
      DO 250 I=1, NOFLWN
      WRITE (6, 155) I, NPFLWN (I), I, (NQFLWN (J, I), J=1, NPFLWN (I))
250 WRITE (6, 156) I, (NRFLWN (J, I), J=1, NPFLWN (I))
C
      WRITE (6, 160) NOPRSN
      DO 260 I=1, NOPRSN
      WRITE (6, 165) I, NPPRSN (I), I, (NQPRSN (J, I), J=1, NPPRSN (I))
260 WRITE (6, 166) I, (NRPRSN (J, I), J=1, NPPRSN (I))
C
      WRITE (6, 170) NOPMPN
      DO 270 I=1, NOPMPN
      WRITE (6, 175) I, NPPMPN (I), I, (NQMPN (J, I), J=1, NPPMPN (I))
270 WRITE (6, 176) I, (NRPMPN (J, I), J=1, NPPMPN (I))
C
      WRITE (6, 180) NOVLVN
      DO 280 I=1, NOVLVN
      WRITE (6, 185) I, NPVLVN (I), I, (NQVLVN (J, I), J=1, NPVLVN (I))
280 WRITE (6, 186) I, (NRVLVN (J, I), J=1, NPVLVN (I))
C
100 FORMAT (3I6)
110 FORMAT (1H1)
120 FORMAT (1H0, '***** INFORMATION OF INPUT DATA IN UNIT ', I2, ' *****',
1 ' SUB. INPUT1 *****')
130 FORMAT (1H ,10X, 'JTOTAL = ', I3, 5X, 'JBLOCK = ', I3, 5X, 'JREC1 = ', I3)
C
140 FORMAT (1H0, 5X, '**DIMENSION NO. OF POOL VALIABLE (TEMP)**' /
1 1H ,8X, 'NOTEMP= ', I5 )
145 FORMAT (1H ,10X, 'NPTEMP (', I3, ') = ', I5, 5X, 'NQTEMP (1, ', I3, ') = ', I5)
C
150 FORMAT (1H0, 5X, '**DIMENSION NO. OF POOL VALIABLE (FLWN)**' /
1 1H ,8X, 'NOFLWN= ', I5 )
155 FORMAT (1H ,10X, 'NPFLWN (', I3, ') = ', I5, 5X, 'NQFLWN (1, ', I3, ') = ', I5)
156 FORMAT (1H ,
33X, 'NRFLWN (1, ', I3, ') = ', I5)
C
160 FORMAT (1H0, 5X, '**DIMENSION NO. OF POOL VALIABLE (PRSN)**' /

```



```

      1 1H , 8X, 'NOPRSN= ', I5 )
165 FORMAT (1H , 10X, 'NPPRSN (', I3, ') = ', I5, 5X, 'NQPRS (1, ', I3, ') = ', I5)
166 FORMAT (1H , 33X, 'NRPRS (1, ', I3, ') = ', I5)
C
170 FORMAT (1H0, 5X, '**DIMENSION NO. OF POOL VARIABLE (PMPN)**' /
      1 1H , 8X, 'NOPMPN= ', I5 )
175 FORMAT (1H , 10X, 'NPPMPN (', I3, ') = ', I5, 5X, 'NQPMPN (1, ', I3, ') = ', I5)
176 FORMAT (1H , 33X, 'NRPMPN (1, ', I3, ') = ', I5)
C
180 FORMAT (1H0, 5X, '**DIMENSION NO. OF POOL VARIABLE (VLVN)**' /
      1 1H , 8X, 'NOVLVN= ', I5 )
185 FORMAT (1H , 10X, 'NPVLVN (', I3, ') = ', I5, 5X, 'NQVLVN (1, ', I3, ') = ', I5)
186 FORMAT (1H , 33X, 'NRVLVN (1, ', I3, ') = ', I5)
      RETURN
C
C
1000 FORMAT ( 2I4)
      END
      SUBROUTINE INPUTO
C
      COMMON /NUMREC/ NUMREC
      COMMON /UNITI / IUNIT1, IUNIT2, MINSEC
      COMMON /OPTONI/ ISELCT, INPUTD, MACHIN, JREC1
      COMMON /TITLEI/ TITLE1 (8), TITLE2 (8), CMENT1 (20), CMENT2 (20)
C
C
      READ (5, 1000, END=999) ISELCT, MACHIN, NUMREC
      IF ( MACHIN.NE.1 ) THEN
          READ (5, 1300) (TITLE1 (I), I=1, 8)
          READ (5, 1300) (TITLE2 (I), I=1, 8)
      ELSE
          READ (5, 1200) (TITLE1 (I), I=1, 4)
          READ (5, 1200) (TITLE2 (I), I=1, 4)
      END IF
      READ (5, 1100) (CMENT1 (I), I=1, 20)
      READ (5, 1100) (CMENT2 (I), I=1, 20)
      RETURN
C
999 CONTINUE
      WRITE (6, 1400)
      STOP
C
C
1000 FORMAT (3I4)
1100 FORMAT (20A4)
1200 FORMAT ( 4A8)
1300 FORMAT ( 8A4)
1400 FORMAT (1H1/1H0, 50X, 28 ('*')/1H , 50X, '*', 26X, '*' /1H , 50X, '*', 5X,
      1 'EDIT1 CODE ENDED', 5X, '*' /1H , 50X, '*', 26X, '*' /1H , 50X,
      2 28 ('*') )
      END

```

```

      SUBROUTINE INPUT1 ( WORK, JPOOL, JDIMX, JDIMY, INDX )
C
C INPUT OF COPD-CODE'S BLOCKED DATA ( FT10 )
C
C INPUT
C   JPOOL ----- NUMBER OF POOL VARIABLE
C   JDIMX----- X DIMENSION NO. OF POOL VARIABLE
C   JDIMY----- Y DIMENSION NO. OF POOL VARIABLE
C OUTPUT
C   WORK ----- INPUT DATA ARRAY
C   INDX ----- INDEX NUMBER
C
C
C   COMMON /NUMBEI/ JTOTAL, JBLOCK
C   COMMON /NUMREC/ NUMREC
C   COMMON /OPTONI/ ISELCT, INPUTD, MACHIN, JREC1
C   COMMON /UNITI / IUNIT1, IUNIT2, MINSEC
C
C   DIMENSION WORK(1)
C   DIMENSION WORKD(2000)
C
C   DOUBLEPRECISION WORKD
C
C
C
C   NB      = 1
C   ISTAT   = 1
C   IEND    = NUMREC
C   REWIND IUNIT1
C
C   --INDX--
C   CALL INDEX(JPOOL, JDIMX, JDIMY, INDX)
C   IF( INDX .EQ. -999) RETURN
C
C   INDX2 = JREC1 * INDX
C 1000 CONTINUE
C   IF( INDX ) 1100, 1400, 1200
C
C
C   --INPUT FROM IUNIT2
C 1100 CONTINUE
C   CALL INPUTS
C   INDX = 0
C   INDX2 = JREC1 * INDX
C
C
C   GO TO 1000
C
C
C 1200 CONTINUE
C   DO 1300 I=1, INDX
C 1300 READ(IUNIT1)

```

```

C
1400 CONTINUE
      READ(IUNIT1) (WORKD(I), I=ISTAT, IEND)
C
      DO 141 I=ISTAT, IEND
141  WORK(I) = WORKD(I)
C
      IF( INDX.NE.JREC1 ) THEN
        DO 1600 I=1, INDX2
1600  READ(IUNIT1)
        END IF
        IF( NB.EQ.JBLOCK ) GO TO 1900
        NB = NB + 1
        ISTAT = IEND + 1
        IEND = IEND + NUMREC
        IF( NB.EQ.JBLOCK ) IEND = JTOTAL
        GO TO 1000
1900 CONTINUE
        IF( MINSEC.EQ.0 ) GO TO 1920
        IF( INDX.NE.0 ) GO TO 1920
        DO 1910 L=1, JTOTAL
1910  WORK(L) = WORK(L) / 60.0
1920 CONTINUE
        RETURN
C
C
C
      END
      SUBROUTINE INPUT2(WORK, JPOOL, JDIMX, JDIMY, INDX )
C
C FOR INPUT OF ZIKEIRESTU DATA
C
C WORK ----- INPUT DATA ARRAY
C JJ ----- LOOP NUMBER
C INDX ----- INDEX NUMBER
C
C
      COMMON /NUMBEI/ JTOTAL, JBLOCK
      COMMON /OPTONI/ ISELCT, INPUTD, MACHIN, JREC1
      COMMON /UNITI / IUNIT1, IUNIT2, MINSEC
C
C
      DIMENSION WORK(1), TITLE(8), CMENT(20)
C
C
      REWIND IUNIT1
      CALL INDEX(JPOOL, JDIMX, JDIMY, INDX)
      IF( INDX.EQ. -999) RETURN
      IF( INDX.GT.0 ) GO TO 100
      IF( INPUTD.EQ.1 ) THEN
        READ(IUNIT1) (TITLE(I), I=1, 8 )

```

```

        READ (IUNIT1)      (CMENT (I), I=1, 20)
        READ (IUNIT1)      JTOTAL
        READ (IUNIT1)      (WORK (I), I=1, JTOTAL)
ELSE IF ( INPUTD. EQ. 2 ) THEN
        READ (IUNIT1, 1000) (TITLE (I), I=1, 8 )
        READ (IUNIT1, 1100) (CMENT (I), I=1, 20)
        READ (IUNIT1, 1200) JTOTAL
        READ (IUNIT1, 1300) (WORK (I), I=1, JTOTAL)
END IF
        WRITE (6, 1500) IUNIT1
        WRITE (6, 1600) (TITLE (I), I=1, 8 )
        WRITE (6, 1700) (CMENT (I), I=1, 20)
        WRITE (6, 1800) JTOTAL
        IF ( MINSEC. EQ. 0 ) GO TO 200
        DO 210 I=1, JTOTAL
210 WORK (I) = WORK (I) / 60.0
200 CONTINUE
        GO TO 900
C
100 CONTINUE
        IF ( INPUTD. EQ. 1 ) THEN
                READ (IUNIT1)      (TITLE (I), I=1, 8 )
                READ (IUNIT1)      (CMENT (I), I=1, 20)
                READ (IUNIT1)      JTOTAL
                DO 40 II=1, INDX
40 READ (IUNIT1)      (WORK (I), I=1, JTOTAL)
                READ (IUNIT1)      (WORK (I), I=1, JTOTAL)
        ELSE
                READ (IUNIT1, 1000) (TITLE (I), I=1, 8 )
                READ (IUNIT1, 1100) (CMENT (I), I=1, 20)
                READ (IUNIT1, 1200) JTOTAL
                DO 41 II=1, INDX
41 READ (IUNIT1, 1300) (WORK (I), I=1, JTOTAL)
                READ (IUNIT1, 1300) (WORK (I), I=1, JTOTAL)
        END IF
C
900 CONTINUE
        RETURN
C
1000 FORMAT ( 8A4)
1100 FORMAT (20A4)
1200 FORMAT (I5)
1300 FORMAT (9E14.7)
1500 FORMAT (1H0, '***** INFORMATION OF INPUT DATA IN UNIT ', I2, ' *****',
1          ' SUB. INPUT2 *****')
1600 FORMAT (1H ,10X, 'TITLE = ', 8A4)
1700 FORMAT (1H ,10X, 'CMENT = ', 20A4)
1800 FORMAT (1H ,10X, 'JTOTAL = ', I3)
1900 FORMAT (1H1)
        END
CEDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1

```

```

CEDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1
CEDIT1EDIT1
CEDIT1EDIT1          E D I T I          C O D E          EDIT1EDIT1
CEDIT1EDIT1          M A I N  P R O G R A M          EDIT1EDIT1
CEDIT1EDIT1          1984.7          EDIT1EDIT1
CEDIT1EDIT1          EDIT1EDIT1
CEDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1
CEDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1EDIT1
C
C
      COMMON /OPTONI/ ISELCT, INPUTD, MACHIN, JREC1
C
C
      20 CONTINUE
      CALL INPUTO
      CALL EDITIT
      GO TO (1, 1, 3, 4, 4, 6), ISELCT
C ZIKEIRETSU DATA OUTPUT
      1 CALL DATOUT
      GO TO 20
C TAPE OF ZIKEIRETSU DATA
      3 CALL TAPOUT
      GO TO 10
C DATA PLOT & DATA OVER PLOT
      4 CALL PLOTO
      GO TO 10
C DATA SUMMARY OUTPUT
      6 CALL SUMMRY
      10 CONTINUE
      WRITE (6,100)
      100 FORMAT (1H1/1H0, 50X, 28('*')/1H , 50X, '*', 26X, '*' /1H , 50X, '*', 5X,
      1      'EDIT1 CODE ENDED', 5X, '*' /1H , 50X, '*', 26X, '*' /1H , 50X,
      2      28('*') )
      STOP
      END
      SUBROUTINE SUMMRY
C
C FOR DATA PRINT OUT
C
      COMMON /DATAI / WORKX (2000), WORKY1 (2000), WORKY2 (2000),
      1      WORKY3 (2000), WORKY4 (2000), WORKY5 (2000),
      2      WORKY6 (2000), WORKY7 (2000), WORKY8 (2000),
      3      WORKY9 (2000), WORKY10 (2000)
      COMMON /OPTONI/ ISELCT, INPUTD, MACHIN, JREC1
      COMMON /PLTIN2/ NOTE (6), NTITLX (6), NTITLY (6), IPEN (3)
      COMMON /SUMY / IOUT, JOUT, MEMO (30)
      COMMON /UNITI / IUNIT1, IUNIT2, MINSEC
C
C
      READ (5, 1000) IUNIT1, IUNIT2, INPUTD
      IF ( INPUTD.EQ.3 ) CALL INPUT1 (WORKX, 0, -1, -1, INDX )

```

```

        IF ( INPUTD.NE.3 ) CALL INPUT2(WORKX,0,-1,-1,INDX )
140 CONTINUE
    READ (5,1100,END=999)          IOUT, JOUT
    DO 100 I=1, IOUT
        L2      = I * 3
        L1      = L2 - 2
        READ (5,1200) LOOPX, INDXX1, INDXX2, (MEMO(J), J=L1, L2)
        GO TO (1,2,3,4,5,6,7,8,9,10), I
    1 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY1, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY1, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    2 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY2, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY2, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    3 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY3, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY3, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    4 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY4, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY4, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    5 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY5, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY5, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    6 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY6, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY6, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    7 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY7, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY7, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    8 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY8, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY8, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
    9 IF (INPUTD.EQ.3) CALL INPUT1 (WORKY9, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORKY9, LOOPX, INDXX1, INDXX2, INDX)
      GO TO 100
   10 IF (INPUTD.EQ.3) CALL INPUT1 (WORK10, LOOPX, INDXX1, INDXX2, INDX)
      IF (INPUTD.NE.3) CALL INPUT2 (WORK10, LOOPX, INDXX1, INDXX2, INDX)
100 CONTINUE
    CALL SUMOUT
    GO TO 140
999 CONTINUE
    RETURN

C
1000 FORMAT (3I4)
1100 FORMAT (2I4)
1200 FORMAT (3I4, 2X, 3A4)
    END
    SUBROUTINE SUMOUT

C
C FOR SUMMARY OUTPUT
C

```

```

C
COMMON /DATA1 / WORKX(2000), DATAY(2000,10)
COMMON /NUMBEI/ JTOTAL, JBLOCK
COMMON /SUMY / IOUT, JOUT, MEMO(30)

C
C
IOUT1 = IOUT * 3
IPAGE = JTOTAL / 50
JMAX = IPAGE * 50
IF( JMAX.LT. JTOTAL ) IPAGE = IPAGE + 1
NPAGE = 1
ISTART = 1
IEND = 50
WRITE(7,1610) (MEMO(I), I=1, IOUT1)
100 IF( NPAGE.GT. IPAGE) GO TO 999
IF( IOUT.GT.0 ) WRITE(6,1200) NPAGE, IPAGE
WRITE(6,1600) (MEMO(I), I=1, IOUT1)
IF( IEND.GT. JTOTAL ) IEND = JTOTAL
IF( IEND.EQ. ISTART ) GO TO 999
DO 200 I=ISTART, IEND
IF(JOUT.EQ. 1)WRITE(6,2100) WORKX(I), (DATAY(I, J), J=1, IOUT)
IF(JOUT.EQ. 2)WRITE(6,2200) WORKX(I), (DATAY(I, J), J=1, IOUT)
IF(JOUT.EQ. 1)WRITE(7,2110) WORKX(I), (DATAY(I, J), J=1, IOUT)
IF(JOUT.EQ. 2)WRITE(7,2210) WORKX(I), (DATAY(I, J), J=1, IOUT)
200 CONTINUE
ISTART = ISTART + 50
IEND = IEND + 50
NPAGE = NPAGE + 1
GO TO 100
999 CONTINUE
RETURN

C
C
1200 FORMAT (1H1, 51X, 'PAGE ', I2, ' / ', I2)
1600 FORMAT (1H , 6X, 'TIME', 10(3A4))
1610 FORMAT (8X, 'TIME', ', ', 10(3A4, ', '))
2100 FORMAT (1H , F10. 3, 10E12. 3)
2200 FORMAT (1H , F10. 3, 10F12. 3)
2110 FORMAT (11 (E12. 5, ', '))
2210 FORMAT (11 (F12. 5, ', '))

C
END

```