

J A E R I - M

7 1 0 0

2 次元データの標準的グラフ化  
副プログラム：S T D P L

1 9 7 7 年 5 月

藤 村 統 一 郎

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

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問合せは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

2 次元データの標準的グラフ化  
副プログラム：STDPL

日本原子力研究所東海研究所原子炉工学部

藤 村 統 一 郎

( 1977年4月30日受理 )

グラフを描くためのサブルーチン、STDPL が開発された。データは 2 つの成分を持つものとし、洗練されたパラメーターを用いて x - y 平面上に描かれる。基本プロッター・サブルーチンを組み合わせることにより、機能的なサブルーチン集合体、F ーシリーズも併せて開発され、STDPL で使われている。

一般に、実験や計算で得られた結果を視覚化することは、その傾向を把握するうえで重要であるが、基本サブルーチンのみでこれをプロットすることは厄介であり、かつエラーを伴い易い。STDPL は可能な限り入力パラメーターのエラーを検出し、これを修正してプロットを続けるよう設計されている。

いくつかの典型的な例題と、これらのサブルーチンの FORTRAN リストも与えられる。

JAERI-M 7100

Subprogram STDPL for Standard  
Graph Drawing of Two-Dimensional  
Data

Toichiro FUJIMURA

Division of Reactor Engineering, Tokai, JAERI

(Received April 30, 1977)

The subroutine STDPL has been developed for drawing the graphs of data. The data should have two components, and the graphs are drawn on x-y plane using refined parameters. A functional subroutine package, F-Series, was also developed by combining the basic plotter subroutines, which is employed in STDPL.

Visualization of the data is important in grasping trend of the results obtained by experiment or calculation. However, plotting them by use of the basic subroutines alone is laborious and tends to give errors. The STDPL is designed to detect input-parameter errors and correct them as much as possible to continue the plotting.

Some typical examples and FORTRAN list of the subroutines are also given.

## 目 次

1. はじめに .....	1
2. F-シリーズの機能 .....	3
3. STDPL の機能 .....	8
3.1 標準仕様プロット .....	8
3.2 特殊仕様プロット .....	9
4. 例題 .....	15
5. おわりに .....	24
謝 辞 .....	24
参考文献 .....	25
付録 A F-シリーズのFORTRAN リスト .....	26
付録 B STDPL のFORTRAN リスト .....	33
付録 C 各サブルーチンの必要語数 .....	48
付録 D FFIT, MULTI, NOTE で使えるシンボル・マーク一覧表 .....	49

## Contents

1. Introduction .....	1
2. Description of F-Series .....	3
3. Description of STDPL .....	8
3.1 Standard Plotting .....	8
3.2 Special Plotting .....	9
4. Some Examples .....	15
5. Concluding Remarks .....	24
Acknowledgments .....	24
References .....	25
Appendix A FORTRAN Lists of F-Series .....	26
Appendix B FORTRAN Lists of STDPL .....	33
Appendix C Computer Core Memory Required for Each Subroutine ..	48
Appendix D Table of Available Characters for FFIT, MULTI and NOTE .....	49

## 1. はじめに

我々が通常手にする数量的なデータは、空間と空間あるいは時間と空間等2つの成分をもっている場合が多い。これらはそれぞれ、ある共通の物理的な単位で表わされた数の列になっていて、それを眺めながらデータの傾向を把握したり、またもう少し分り易くするために、これらを一つ一つ手でプロットするのは大変である。

しかし、すぐさま計算機を使って自動的にグラフ化を試みてもそれなりに困難さはある。まず、グラフィック・プロッター<sup>\*</sup>のソフトウェアを理解しなければならず、計算機を使ううえでのターン・アラウンドの問題もある。また、手でプロットするときは、どんな量をどの位置にプロットしたら良いかというスケーリングの問題や、グラフの目盛や標題をどのように書くかという問題は直感的に解決できるが、これをプログラムで行うとすれば相当厄介であろう。

約10年前、原研にも CALCOMP 社製のグラフィック・プロッター<sup>1)</sup>が導入され、あちこちでこれが用いられるようになったが、線を引くとか、文字を書くとかの基本的なサブルーチンしか用意されておらず<sup>1)</sup>、初めて使う者は誤りを犯し易くなかな目的のグラフを描くことができなかった。その後、ソフトウェアもより機能的なサブルーチンを揃えるなどして整備され、使い易くなつたが<sup>2) 3)</sup>、一層総合的なプログラムが要望されるようになった。

GPLOT<sup>4)</sup>は一般的でかつ汎用性をもった最初のプロット用副プログラム、GPLOT 1<sup>4)</sup>に基いており、データもフリー・フォーマットを用いてカードで与えるほか、棒グラフ表示ができる等の特色をもっている。これらの改良版である GPLOTZ<sup>5)</sup>は1次元配列に格納されているデータの任意の区間を取り出してプロットしたり、一定の範囲からはみ出す場合に特別な処置を施したりするようになっている。また、1次元配列のため、データの数が不揃いな何ケースものデータを一度に扱う場合、2次元配列を用いるのに比べて記憶容量に無駄がないという長所をもつている。

一方、できるだけプロッター独特の制約を離れ、多少のエラーも克服する、簡単に使えるプログラムが強く望まれ、標記のプログラム、FーシリーズおよびSTDPLを開発するに至った。

これらのプログラムは「FORTRANは使い慣れているが、プロッターは初めて」という者でも容易に使えるよう配慮されているが、より要求に合ったグラフを描くためにもプロッター・マニュアル<sup>1) 2)</sup>を一読されることを勧めたい。

プロッターを使うときは、通常次の事項\*\*に留意する必要がある。

- ① プロッター使用の申込（手続きとジョブ制御カードの組み方）
- ② プロッター用紙（大きさや進み方）
- ③ プロッター用テープの制御（書始めと書終り）
- ④ 座標の移動（原点の位置）

\* 以下、グラフィック・プロッターをプロッターと呼ぶなど略語をしばしば用いる。

\*\* 項目①については所内資料（計算センター：Computer 情報 No.27(1976)を、項目②ないし⑤については文献 1), 2) を参照されたい。

⑤ データのスケーリング（データを表わす数と座標平面の点との対応）

STDPLは、データを与えたとき、標準的な仕様と、ユーザーが種々のパラメーターを規定する特殊な仕様との二通りのプロットが可能なマルチ・エントリーのサブルーチンであり、後者の使い方で相当のレイ・アウトを考慮したプロットが可能である一方、前者を用いると上記留意事項の②ないし⑤を全く考えずに済む\*。

また、F-シリーズは独立したサブルーチンの集合体であり、おのおのが、データの存在する範囲を調べたり、スケーリングを行ったり、あるいはグラフの目盛を書く等のまとまった機能を有している。これらはSTDPLにおいて随所で使われているが、ユーザーが呼び出すこともできる。

第2章と第3章では、JSSL（原研・科学用サブルーチン・ライブラリー）<sup>6)</sup>の登録申込書の書式を模擬してこれらの使い方の詳細を述べるが、一方しか使わないときは他方を読みとばしても構わない。第4章ではユーザーのプログラムのディバッグの仕方を含めた例題が与えられるが、これらのサブルーチンがプロッターだけでなく、グラフィック・ディスプレイ<sup>7)</sup>やCOM(Computer Output Microfilm)<sup>8)</sup>においても効果的に働くことが理解されよう。

なお、これらはJSSL<sup>6)</sup>に登録されており、ソース・カードなしに使用できるが、プログラムの詳細やその改編については、巻末のFORTRANリスト（付録A～C）を参照されたい。

---

\* このような使い方で充分なときは直接、第3章の初めと3.1節を参照されたい。

## 2. F-シリーズの機能

F-シリーズは基本サブルーチンを使い易くまとめた機能的なサブルーチンの集合体であり、各々は独立していてユーザーが単独に呼び出すこともできる。以下、それらの機能および使用方法について説明する。

データの存在する範囲を求める

CALL FRANGE ( MC , MCMAX , NP , AR , ARMIN , ARMAX )

データを入れた配列において

$$AR(i, j), j = 1 \sim NP(i), i = 1 \sim MC \quad (1)$$

の範囲で、最大値と最小値を検する (Fig. 1 参照)。

MC —— 配列 AR の中で、実際に存在する、データのケースの数。整数型の入力\*。  $MC \geq 1$ 。  
MCMAX —— 配列 AR の大きさを示す第 1 の寸法。整数型の入力。  $MCMAX \geq MC$ 。 例えばユーザーのプログラムで  $AR(20, 50)$  という配列宣言になっているとき、20を入れる。

NP —— 各ケースのデータの数。例えば第  $i$  ケースのデータの数が 5 のとき、 $NP(i) = 5$  とする。整数型配列  $NP(NDN)$  で、入力。  $NDN \geq \max_{1 \leq i \leq MC} NP(i)$ 。

AR —— データを入れる。実数型配列  $AR(MCMAX, NDA)$  で、入力。  $NDA \geq \max_{1 \leq i \leq MC} NP(i)$ 。

ARMIN — (1)式の範囲での AR の最小値。実数型の出力。

ARMAX — (1)式の範囲での AR の最大値。実数型の出力。

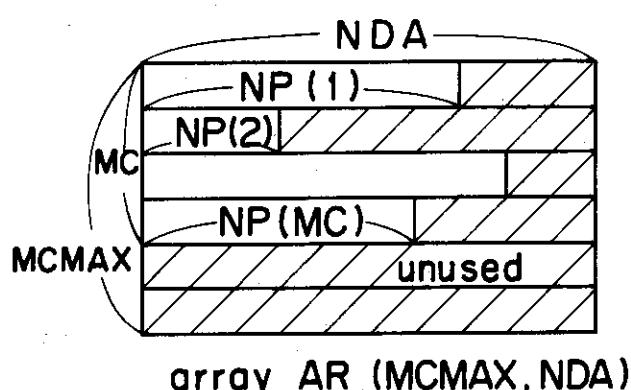


Fig. 1 Data arrangement in array AR

\* 該当する引数が入力用のパラメーターであることを単に「入力」と記す。出力についても同様で、特に入出力のときは予じめ入れた値が変わっているので注意を要する。

データのスケーリング(変換)を行う

CALL FTRANS (M, MDIM, N, Z, X1, X2, Y1, Y2, LL)

2次元配列Zに入れられているデータ

$z(i, j), j = 1 \sim n_i, i = 1 \sim m$

を、 $x_1$ が $y_1$ に、 $x_2$ が $y_2$ になるよう線型または常用対数変換を行う(Fig. 1 および 2 を参照)。

M —— データのケースの数 m。整数型の入力。 $M \geq 1$ 。

MDIM —— 配列Zの大きさを示す第1の寸法。整数型の入力。 $MDIM \geq M$ 。

N —— 各ケースのデータの数  $n_i$ を入れる。整数型の入力。 $N(i) \geq 1, i = 1 \sim M$ 。

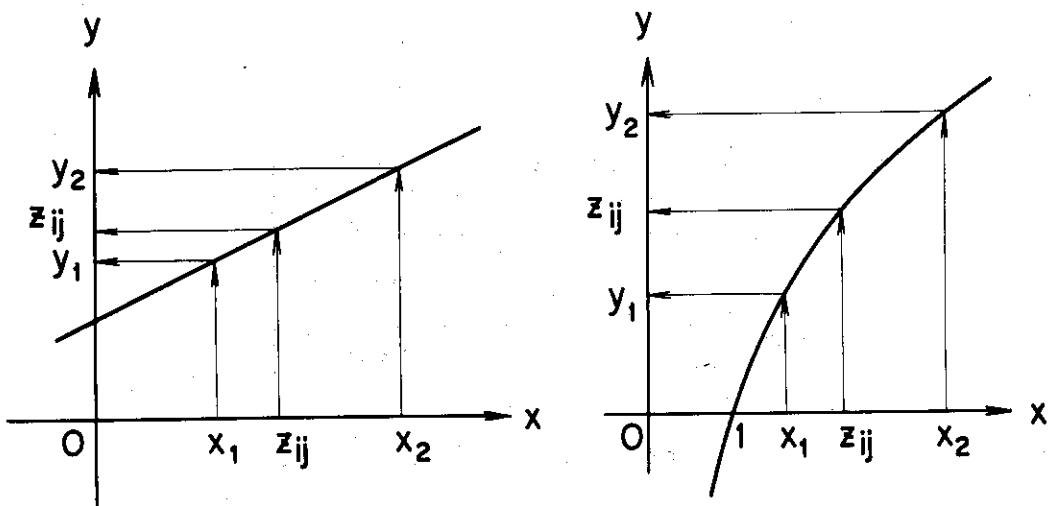
Z —— データ  $z_{ij}$ を入れると、変換された値が出力される。実数型配列Z(MDIM, NZ)で、入出力。 $NZ \geq \max_{1 \leq i \leq M} N(i)$ 。対数変換のときは  $Z(i, j) > 0$  であること。

X1, X2, Y1, Y2 —— 変換の基準となる数。X1がY1に、X2がY2に変換される。実数型の入力。 $X1 < X2, Y1 < Y2$ 。対数変換のときは  $X1 > 0$  とすること。

LL —— 変換の型を指定する。1のとき線型、2のとき対数変換である。整数型の入力。

[注] 制限外の指定等に対し、警告のプリントをして次の処理を行う。

- ①  $X1 > X2$  または  $Y1 > Y2$  のとき、それぞれ値を入れかえる。
- ②  $X1 = X2$  または  $Y1 = Y2$  のとき、 $Z(i, j) \equiv Y1$  と変換する。
- ③  $LL \neq 1, 2$  のとき、 $LL = 1$  とする。
- ④  $LL = 2, Z(i, j) \leq 0$  のとき、 $LL = 1$  とする。



(a) Linear transformation

(b) Logarithmic transformation

Fig. 2 Transformation of data

データを内挿する  
CALL FFIT ( X, Y, NP, JALT, LS, IERP )

内挿ルーチンSMOOT<sup>3)</sup>をオプションで用い、データ点間を折線または曲線でプロットする  
(Fig. 3 参照)。

X — 変換済みのデータの x 座標。実数型配列 X (NDX) で、入力。  $NDX \geq NP$ 。

Y — 変換済みのデータの y 座標。実数型配列 Y (NDY) で、入力。  $NDY \geq NP$ 。

NP — データの数。整数型の入力。  $NP \geq 1$ 。

JALT — シンボルや曲線の描き方を指定する<sup>1)</sup>。

①  $JALT > 0$  のとき、シンボルは ( $JALT - 1$ ) 個おきに描き、それらを結ぶ曲線も描く。

②  $JALT = 0$  のとき、曲線のみ描く。

③  $JALT < 0$  のとき、シンボルは ( $|JALT| - 1$ ) 個おきに描き、曲線は描かない。

整数型の入力。

LS — シンボル・マークの識別番号 (付録 D または文献 1) 参照)。整数型の入力。

$0 \leq LS \leq 127$ 。但し  $LS \leq 15$  が望ましい。

IERP — 折線か曲線かを指定する。  $NP \geq 3$ ,  $JALT \geq 0$  のとき、2 を指定すると SMOOTH を使って曲線による内挿をする。他の場合は折線プロットである。整数型の入力。  $1 \leq IERP \leq 2$ 。

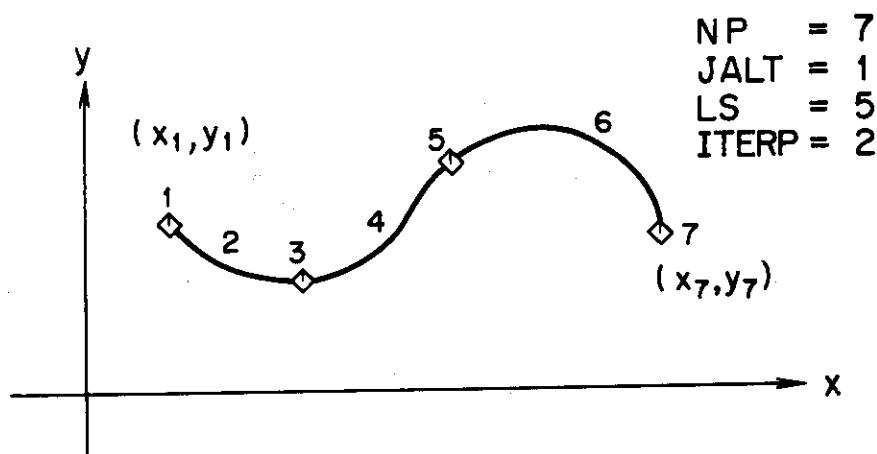


Fig. 3 Example of interpolation by FFIT

プロット用テープの制御と座標の移動を行う  
CALL FCONT ( XG, YG, MCN )

XG, YG — 新しい原点の座標 (mm)。実数型の入力。

MCN —— 制御用パラメーター。原点移動のみを行うとき 0, プロッターを使い始めるとともに原点移動を行うとき 1, 原点移動の後、プロッターを使い終るとき 2 とする。整数型の入力。

目盛りを描く

CALL FDIAL ( XS , YS , THETA , DIST , NL , HEIGHT )

等間隔に配置される、同じ長さの線分を描く (Fig. 4 参照)。

XS , YS — 描き初めの線分の始点の座標 (mm)。実数型の入力。

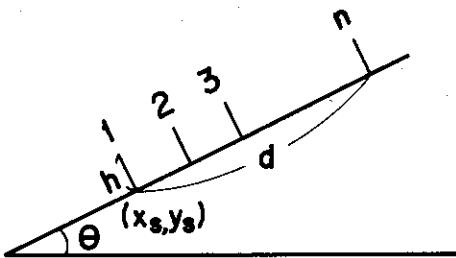
THETA — 線分の始点が進む方向が、x 軸となす角  $\theta$  (度)。実数型の入力。DIST — 初めの線分と終りの線分との間隔  $d$  (mm)。実数型の入力。DIST > 0.。NL — 線分の数  $n$ 。整数型の入力。NL  $\geq 1$ 。HEIGHT — 各線分の長さ  $h$  (mm)。実数型の入力。HEIGHT  $\geq 0$ .。

Fig. 4 Marks drawn by FDIAL

線分を描く

CALL FSEGM ( XS , YS , XE , YE )

2 点の座標を与えることにより、線分を描く (Fig. 5 参照)。

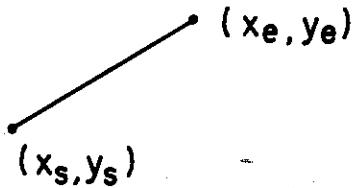
XS , YS , XE , YE — それぞれ、始点および終点の、x および y 座標である (mm)。  
実数型の入力。

Fig. 5 Segment drawn by FSEGM

等差的に変わる数の列を書く

CALL FSEQU ( XS , YS , THETA , DIST , FPNS , FPNE , NT , HEIGHT ,  
NN , ISTYL )

数の列を、上下あるいは左右から見ても揃っているように描く (Fig. 6 参照)。

XS, YS — 最初に書く数の位置 (mm)。実数型の入力。

THETA — 数の配置の進む方向が、x 軸となす角  $\theta$  (度)。実数型の入力。

DIST — 最初と最後の数の隔り  $\ell$  (mm)。実数型の入力。DIST  $\geq 0$ 。

FPNS, FPNE — 最初および最後の数。実数型の入力。

NT — 書く数の全数  $n$ 。整数型の入力。NT  $\geq 1$ 。

HEIGHT — 書かれる数の大きさ  $h$  (mm)。実数型の入力。HEIGHT  $> 0$ 。

NN — 小数点以下の桁数を指定する<sup>1)</sup>。-1のときは整数で、0のときは整数に小数点を付けた形で書かれる。整数型の入力。 $-1 \leq NN \leq 11$ 。

ISTYLY — 書き方を指定する。1のときはFig. 6(a), 2のときはFig. 6(b)の形式で書く。整数型の入力。

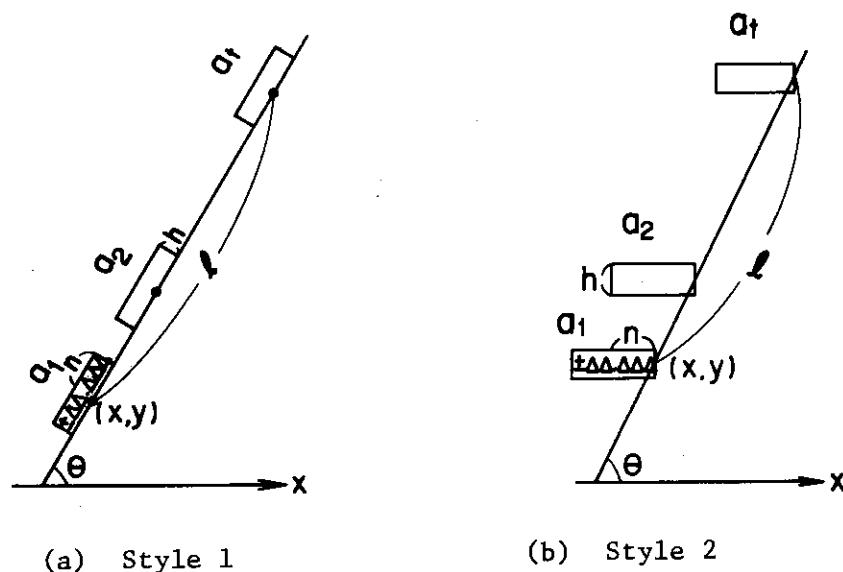


Fig. 6 Number sequence drawn by FSEQU

### 3 STDPL の機能

各ケースのデータの数  $n_i$  が不揃いの、  $m$  ケースのデータ  $P_{ij}$  が 2 つの成分  $(x_{ij}, y_{ij})$  をもち、 おのとの成分は 2 つの 2 次元配列に格納されているとする。<sup>\*</sup> STDPL はかかるデータを、

- ① 決められたパラメーター値を用いる標準的な仕様
- ② パラメーター値をユーザーが指定する特殊な仕様

の何れかにより、 同一  $x-y$  座標平面上に  $m$  本の線としてプロットするマルチ・エントリーのサブルーチンである。データが 1 ケースしかなく、 1 次元配列に格納されているときも勿論可能である。また、 このサブルーチンは可能な限り入力パラメーターのエラーを検出し、 かつ有意な範囲でそれを修復していく独特の機能を有しているが、 複雑さを避けるためここでは説明を省略する（これらは付録 B の DATA 文で記述し尽されている）。以下二通りの使い方について説明する。

#### 3.1 標準仕様プロット

STDPL を標準仕様で用いるとき、 1 つのジョブの中でプロッター関係のルーチンはこのメイン・エントリーを唯一度しか使えない。これにより、 グラフは  $200 \text{ mm} \times 150 \text{ mm}$  の大きさで 1 枚だけ描かれる。メイン・エントリーの使用法は次の通りである。

```
CALL STDPL (MC, MCMAX, NP, XP, YP, LLX, LLY, IDSS)
```

MC —— データのケースの数  $m$ 。整数型の入力。  $MC \geq 1$ 。

MCMAX —— 配列 XP, YP の大きさを決める第 1 の寸法。整数型の入力。  $MCMAX \geq MC$ 。

NP —— 各ケースに属するデータの数。  $NP(i) = n_i$  とする。整数型配列 NP (NDN) で入力。

$NP(i) \geq 1, NDN \geq \max_{1 \leq i \leq MC} NP(i)$ 。

XP —— データの  $x$  成分。  $XP(i, j) = x_{ij}$  とする。実数型配列 XP (MC MAX, NDX) で、 入出力。  $NDX \geq \max_{1 \leq i \leq MC} NP(i)$ 。但し、  $x$  軸を対数にとるとときは、  $XP(i, j) > 0$ 。

であること。

YP —— データの  $y$  成分。  $YP(i, j) = y_{ij}$  とする。実数型配列 YP (MC MAX, NDY) で、 入出力。  $NDY \geq \max_{1 \leq i \leq MC} NP(i)$ 。但し、  $y$  軸を対数にとるとときは、  $YP(i, j) > 0$ 。

であること。

LLX ——  $x$  軸のとり方。 1 のとき線型、 2 のとき対数にとる。 整数型の入力。

LLY ——  $y$  軸のとり方。 LLX と同様に指定する。 整数型の入力。

---

\*  $i$  はケースについての添数で  $1 \leq i \leq m$ ,  $j$  は各ケースの中でのデータの添数で  $1 \leq j \leq n_i$  である。

IDSS — 標準使用であることを示すため 1 とする。 整数型の入力。

### 3.2 特殊仕様プロット

STDPL を特殊仕様で用いると、次のような利点がある。

- ① 1 枚のグラフを完成させるに必要なパラメーターの値を指定できる。
- ② 1 つのジョブの中で、データを与え直し、かつ座標を移動することにより、何枚ものグラフが描ける。
- ③ このサブルーチンで 1 枚のグラフを描いている途中で、他のプロッター・ルーチンを呼び出して、このグラフに必要な事項を書き足すことができる。
- ④ このサブルーチンの前後に、他のプロッター・ルーチンを接続できる。

これらは、以下に記す、STDPL の各エントリーの特殊仕様による使用法の説明、および第 4 章の例題を通して明らかにされよう。

まず、STDPL の各エントリーが 1 つのジョブの中でどのような順序で呼ばれているかを概観してみよう。

Fig. 7 の流れ図において、2 重枠のエントリーは必ず呼なければならず、他は呼ばなくてよい。そのときは該当する部分を標準仕様とほぼ同様にプロットする。空白の枠は、その場所で他のプローティング関係のサブルーチンを何回呼んでもよいことを意味するが、この場合、原点の移動を中心とした、位置関係に充分注意しなければならない。また、矢印で示される途中からの帰り道はその回路を何度も通っても良いことを示す。STDPL から CLOSE までは、1 枚のグラフを完成させるので、以下これを 1 サイクルと呼ぶことにする。各エントリーの使用法は次の通りである。

データを与え、特殊仕様であることを示す

CALL STDPL (MC, MCMAX, NP, XP, YP, LLX, LLY, IDSS)

MC から YP までの指定の仕方は 3.1 節の標準仕様のときと同じである。

LLX — x 軸に関する指定。1, 2 のとき、標準仕様と同じく x 軸をそれぞれ、線型、対数にとる。-1 のとき、x 軸を線型にとり、かつ  $x = 0$  の所に区切り線を入れる。-2 のときは x 軸を対数にとり、10 のべき乗の所に区切り線を入れる。(Fig. 8 参照\*)。従って、「LLX=1, 2 のとき」および「LLX=-1, -2 のとき」で、これらの区切り線がグラフの枠内に現れないとき」はともに区切り線は書かれない。整数型の入力。

LLY — y 軸に関する指定。LLX と同様に指定する。整数型の入力。

IDSS — 特殊仕様であることを示すと同時にプロッター用紙の種類を指定する。2 のとき、特殊仕様でふつうのプロッター用紙を用いる。3 のときは、特殊仕様で大きいプロッター用紙\*\* (y 方向が約 90 cm) を用いる。整数型の入力。

\* この図は区切り線の説明のためのものであり、実際のプロットとは多少異なる。

\*\* この種の用紙は現在、原研で使われていない。

データのおよぶ範囲を知る

CALL RANGE ( XPMIN, XPMAX, YPMIN, YPMAX )

XPMIN — データの x 成分を入れた配列 XP の最小値  $\min_{1 \leq i \leq MC, 1 \leq j \leq NP(i)}$  XP(i,j)。  
実数型の出力。

XPMAX — XPMIN と同様な、配列 XP の最大値。実数型の出力。

YPMIN — XPMIN と同様な、配列 YP の最小値。実数型の出力。

YPMAX — XPMAX と同様な、配列 YP の最大値。実数型の出力。

このエントリーが呼ばれないときに次の DIAL が呼ばれると、XPMIN, XPMAX, YPMIN, YPMAX は DIAL の実引数 XLW, XUP, YLW, YUP のチェックに使われる。次の DIAL も呼ばれないときは、それら 4 つの実引数の代わりとなる。

グラフの大きさ等を指定する

CALL DIAL ( WIDTH, HEIGHT, XLW, XUP, YLW, YUP, MOPEN )

WIDTH — グラフ枠の横 (x) 方向の幅 (mm) (Fig. 9 参照)。実数型の入力。  
WIDTH  $\leq 1000.$ 。

HEIGHT — グラフ枠の縦 (y) 方向の幅 (mm)。実数型の入力。HEIGHT  $\leq 200.$   
(大きい用紙のときは、HEIGHT  $\leq 750.$ .)

XLW — グラフで扱う x の下限。実数型の入力。XLW  $\leq$  XPMIN。

XUP — XLW と同様な x の上限。実数型の入力。XUP  $\geq$  XPMAX。

YLW — XLW と同様な y の下限。実数型の入力。YLW  $\leq$  YPMIN。

YUP — XUP と同様な y の上限。実数型の入力。YUP  $\geq$  YPMAX。

MOPEN — プロッターの使い始めの指示。このジョブの中で、プロッター関係のルーチンとしては、このサイクルが最初であるとき 1, その他のとき 0 とする。整数型の入力。

このエントリーを呼ばないときは標準値、WIDTH = 200., HEIGHT = 150.,  
XLW = XPMIN, XUP = XPMAX, YLW = YPMIN, YUP = YPMAX, MOPEN = 1 が入  
れられる。

グラフの標題を書く

CALL TITLE ( LGT, NG, LXT, NX, LYT, NY )

LGT — グラフ全体の標題。文字型の入力。

NG — LGT で与える文字の数。整数型の入力。約  $(7 \times \text{WIDTH} + 656) / 48$  文字まで書かれる。

LXT — x 軸に書く標題。文字型の入力。

NX — LXT で与える文字の数。整数型の入力。約  $(7 \times \text{WIDTH} + 855) / 45$  文字ま

で書かれる。

LYT — y 軸に書く標題。 文字型の入力。

NY — LYT で与える文字の数。 整数型の入力。 約  $(14 \times \text{HEIGHT} + 345) / 90$  文字まで書かれる。

このエントリーを呼ばないときは、 標題として何も書かない

各ケースのデータをプロットする  
CALL MULTI (JALT, LS, IPEN)

JALT — 各ケースのデータ点を折線で結んでいくとき、 データの位置を明らかにするシンボル・マークを何個おきにプロットするかを示す<sup>1)</sup>。 i をケースの添数とするとき、

① JALT(i) > 0 とする、 シンボル・マークは JALT(i)-1 個おきに描き、 データ点の間は実線で結ぶ。

② JALT(i) = 0 とすると、 シンボル・マークは何も描かず、 データ点の間は実線で結ぶ。

③ JALT(i) < 0 とすると、 シンボル・マークは JALT(i) -1 個おきに描き、 データ点の間は結ばない。

整数型配列 JALT (NDJ) で、 入力。  $NDJ \geq MC$ 。

LS — 各ケースに使う、 シンボル・マークの識別番号（付録D参照）。 例えば第 i ケースのために三角のシンボル・マーク△を用いるとき、 LS(i) = 2 とする。 整数型配列 LS (NDL) で、 入力。  $NDL \geq MC$  で、  $0 \leq LS(i) \leq 127$ 。 但し、  $LS(i) \leq 15$  が望ましい。

IPEN — 各ケースをプロットするペンの種類<sup>2)</sup>\*を指定する。 整数型配列 IPEN (NDI) で、 入力。  $NDI \geq MC$ ,  $1 \leq IPEN(i) \leq 3$ 。

標準仕様のとき、  $JALT(i) = (NP(i) - 1 - MOD(NP(i) - 1, 20)) / 20$  ( $1 \leq NP(i) \leq 20$  ならば 1,  $21 \leq NP(i) \leq 40$  ならば 2, ……というふうになっている),  $LS(i) = i$ ,  $IPEN(i) = 1$  である。

グラフに注釈や表を入れる  
CALL NOTE (XTA, YTA, HTA, LTA, STA, NTA, MTA)

このエントリーの機能はプロッター基本サブルーチン<sup>1)</sup> の SYMBOL や NUMBER と同じであるが、 現在描こうとしているグラフに関し、 その許された範囲 (Fig. 9 参照) からはみ出さないかどうかのチェックを行っている。 分り易くするため、 分けて説明する。

① シンボル・マークを描くとき

XTA — マークの中央の位置の x 座標 (mm)。 実数型の入力。

YTA — 同様な y 座標 (mm)。 実数型の入力。

HTA — マークの大きさ (mm)。 実数型の入力。  $HTA \geq 0$ 。

\* 現在はインクの色により分けられている。 ふつう 1 頭であるが、 2(赤)や 3(青)を使うと重なったとき識別し易い。

LTA ——マークの識別番号（付録D参照）。 整数型の入力。  $0 \leq LTA \leq 127$ 。  
 $LTA \leq 15$  が望ましい。

STA —— 0. とする。 実数型の入力。

NTA ——マークの中央までペンを上げて持って来るとき, -1, ペンを下げたままで持つて来るとき-2とする。 整数型の入力。

MTA シンボル・マークを描くことを示すため1とする。 整数型の入力。

② 文字または数字を書くとき

XTA — 文字または数字の左下の位置の x 座標 (mm)。 実数型の入力。

YTA — 同様な y 座標 (mm)。 実数型の入力。

HTA — 文字または数字の大きさ (mm)。  $HTA \geq 0.$ 。 実数型の入力。

LTA — 文字または数字。 文字のとき文字型, 数字のとき実数型で入力。

STA — 文字または数字を書く方向が x 軸となす角度 (度)。 実数型の入力。

NTA — 文字の数または数字の小数点以下の桁数。 整数型の入力。 文字のとき  $NTA \geq 1$ 。  
 数字のとき  $-1 \leq NTA \leq 11$ 。 但し、整数として書くとき -1, 整数に小数点を付けた形で書くとき 0 とする。

MTA 文字を書くとき 2, 数字を書くとき 3 とする。 整数型の入力。

このエントリーを呼ばないときは、注釈として何も書かない。但し、標準仕様では各ケースの番号とそのケースに使われるシンボル・マークの対応表をグラフの枠の外に書く (Fig. 14 参照)。

プロッターの使い終りの制御をする CALL CLOSE (MCLOSE)
---

MCLOSE—このジョブの中で、プロッター関係のルーチンとしては、このサイクルが最後であるとき 1, その他のとき 0 とする。 整数型の入力。

特殊仕様では、最初に原点を少し移動してグラフを描き、更に原点を移動して 1 つのサイクルを終る (Fig. 9 参照)。従って、データを与え直しながら何枚ものグラフを描くとき、プラッタ用紙は x 方向に (40. + WIDTH + 105.) mm ずつ進むことになる。

なお、STDPL は意味のない付属サブルーチン、STAY を、作図量が一定の量に達する度に呼んでいる。これは、例えばグラフをディスプレイ上に表示するときなどに役立つ (第 4 章例 2 参照)。

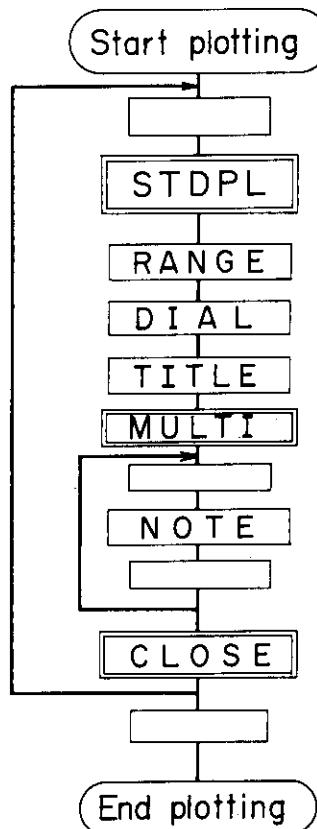


Fig. 7 Order of calling the entries of STDPL

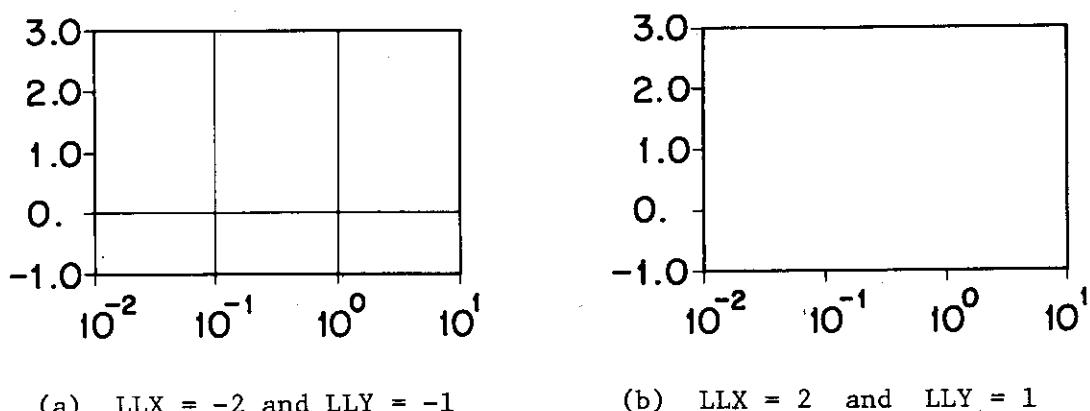


Fig. 8 Frame of a graph with or without section lines for a semi-log plot

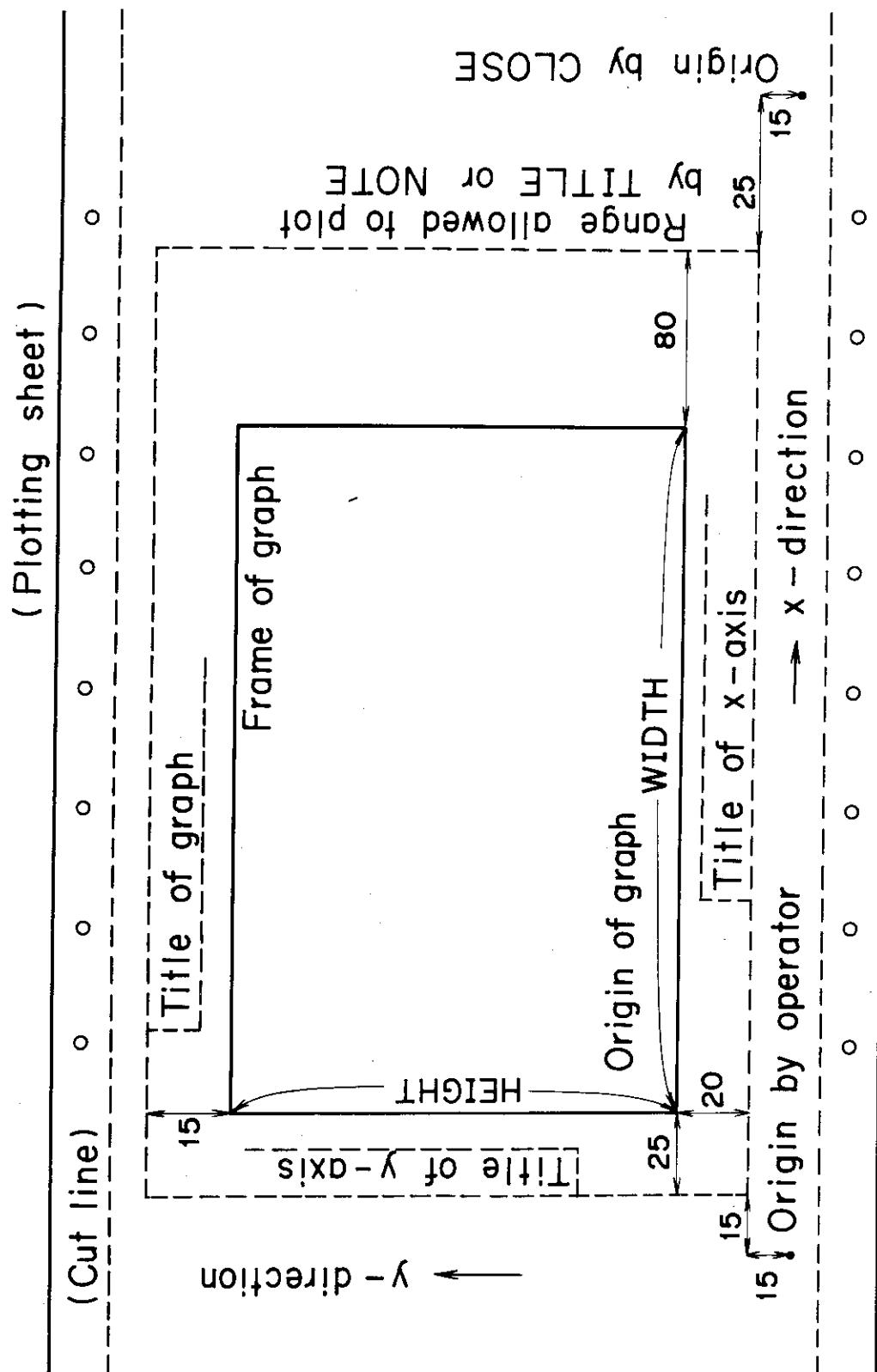


Fig. 9 Specifications of a graph in units of mm

## 4 例 题

ユーザーがF-シリーズおよびSTDPLを容易に理解し、使用し得るために、プログラムのディバッグを含めた典型的な例を掲げよう。

例1： STDPLを特殊仕様で用いる例を考え、ユーザーがそのプログラムのコーディングを一応終えたとしよう（Fig. 10参照）。これは故意にエラーを起した例であるが、このプログラムが正しいかどうかをライン・プリンターのみでチェックするため特殊なFD文、

```
¥FD F 77, DUMMY, DEVD = DA
```

が使われている。この結果がFig. 11であり、重度のエラーの指摘および軽度のエラーの修復の仕方が示される。このチェックを繰り返すことにより、重度のエラーが無くなれば何らかのグラフが描かれるが、初期の目的に合ったグラフを手にするには軽度のエラーも無くなるまで修正することが不可欠である。

例2： これは空間メッシュ数をx軸にとり、原子炉の中性子実効増倍率をy軸にとって、差分法と有限要素法の計算結果を比較しようとしたものである（Fig. 12参照）。

このプログラムは既に例1のようなプリンターによるディバッグは終えたとし、プロッターでプロットする前に、更にディスプレイにより確かめようとしたものである。<sup>7)</sup> この例もSTDPLの特殊仕様による場合であるが、1つのサイクルの前および中でディスプレイの（即ち、プロッター関係の）ルーチン<sup>7)</sup>が多数呼ばれている。TRACE文、

```
TRACE (1) *
```

によりディバッグするためFORTRAN-Dで行っているが、例1とともにFORTRAN-Hによるディバッグも可能である。

例3： これはFig. 13および14に示されるように、ごくありふれた例として試作したものである。Fig. 13ではそれを

- ① F-シリーズ
- ② STDPLの特殊仕様
- ③ 基本ルーチン

の3通りの方法により同一にプロットするプログラムが与えられる。但し、Fig. 13(a)はプログラム全体であり、Fig. 13(b)と13(c)ではその変更部分のみが示される。これらはそれぞれの仮引数の使い方を明確にするとともに、幾分無理はあるがFORTRAN文の数により使い易さを定める目安を与えよう。更にSTDPLの標準仕様の場合には、標題が書けないとか、グラフの目盛がきりの悪い値になる等の恨はあるが

```
CALL STDPL (3, 4, NP, XP, YP, 1, 1, 1)
```

の1行で済み、自由度の多さと使い易さの調節の難しさが伺われよう。

例4： 最後に、多少複雑でかつ一般的なSTDPLの特殊仕様によるグラフを示そう（Fig. 15および16参照）。

これは、円筒型をした原子炉の中心軸からの距離に対する中性子核反応率分布を半対数座標で表したものである。中性子は元来5つに組分けされ、5ケースのデータであったものを、見易く

するため、2枚のグラフに分けて描いたものである。このプログラム・リスト (Fig. 15) に示される2つのサイクルの継ぎ方が、グラフを多く描く場合の参考になろう。

なお、例3のFig. 13(a)がプロッター用のジョブ制御カードの組み方を示すのに対し、このFig. 15はCOM<sup>8)</sup>のそれを示している。\*

---

\* プロッターによるグラフは不鮮明なので、Fig. 14はCOMによるものを用いた。従って、Fig. 16とともにグラフの寸法が変っている。

#NO 2309.

P.O/PCH 0  
 W.O/PAGE 40  
 T.O/TIME 10S  
 /  
 PLT/  
 /SSL

```
*GJOB 1142309,FUJIMURA,T+446.01
*DFOPT C=NOOPT
C ERRONEOUS EXAMPLE
TRACE (1) *
DIMENSION XP(3,6),YP(3,6),NP(3)
DIMENSION JALT(3),LS(3),IPEN(3)
CALL STDPL (-1,-3,NP,XP,YP,-6,-4,-8)
CALL DIAL (-100.,250.,1,0E60,-1,0E60,1,0E-77,-1,0E-77,-2)
CALL RANGE (X1,X2,Y1,Y2)
CALL TITLE (1H ,-1,TITL ,-4 ,8H12345678+0)
CALL MULTI (JALT,-130,4)
CALL NOTE (-100.,800.,-10.,-130.,-60.,-3,-1)
CALL CLOSE (-1)
STOP
END
*DLDIEDRUN GRFD=ON,PLTLIB=CALL
*FD F77,DUMMY,DEVD=DA
*JEND
```

Fig. 10 Program list for Example 1

```
ERROR IN STDPL, MC L.E, 0
ERROR IN STDPL, MCMAX L.T, MC
WARNING IN STDPL, IDSS N.E, 1 TO 3, IDSS=1
WARNING IN STDPL, LLX N.E, 1 OR 2 IN IDSS=1, OR ABS(LLX) N.E, 1 OR 2 IN IDSS=2 OR 3, LLX=1
WARNING IN STDPL, LLY N.E, 1 OR 2 IN IDSS=1, OR ABS(LLY) N.E, 1 OR 2 IN IDSS=2 OR 3, LLY=1
ERROR IN STDPL, PREVIOUS ERROR
WARNING IN STDPL, WIDTH L.E, 0, OR G.T, 1000., WIDTH=200.
WARNING IN STDPL, MOPEN N.E, 0 OR 1, MOPEN=1
ERROR IN STDPL, PREVIOUS ERROR
ERROR IN STDPL, WRONG ORDER CALLING
ERROR IN STDPL, PREVIOUS ERROR
WARNING IN STDPL, NG L.E, 0 OMIT PLOT THIS PART
WARNING IN STDPL, NX L.E, 0 OMIT PLOT THIS PART
WARNING IN STDPL, NY L.E, 0 OMIT PLOT THIS PART
ERROR IN STDPL, PREVIOUS ERROR
ERROR IN STDPL, PREVIOUS ERROR
WARNING IN STDPL, HTA L.T, 0., HTA=ABS(HTA)
WARNING IN STDPL, MTA N.E, 1 TO 3
OMIT PLOT THIS PART
> ERROR IN STDPL, PREVIOUS ERROR
WARNING IN STDPL, MCLOSE N.E, 0 OR 1, MCLOSE=1
```

Fig. 11 Printer output of Example 1

```

*NO 2309.
    / C,2/CORE 145
    / T,0/TIME 105
    / W,0/PAGE 40
    / P,0/PCH 0
    / V 2
    / GDP/ 2
    /SSL

*GJOB 1142309, FUJIMURA, T,446,01
*DFOBT B=MAP,C=NQOPT
C DEBUG BY GRAPHIC DISPLAY
DIMENSION XP(3,6),YP(3,6),NP(3)
DIMENSION JALT(3),LS(3),IPEN(3)
COMMON /XDOA1/ XDOA(3072)
TRACE (1) *
READ (5,12) MC,MAX,MC,1DSS
READ (5,12) NP
READ (5,12) JALT
READ (5,12) LS
READ (5,12) IPEN
DO 17 I=1,MC
NP1=NP(I)
READ (5,14) (XP(I,J),J=1,NP1)
17 CONTINUE
DO 19 I=1,MC
NP1=NP(I)
READ (5,22) (YP(I,J),J=1,NP1)
19 CONTINUE
CALL PLOTS (XDOA,3072)
CALL STDPL (MC,MC,MAX,NP,XP,YP,1, 1,1DSS)
CALL DIAL (75.,85.,0.,14.,0.965,1.09,0.)
CALL TITLE (1H ,1,21HN (N=NUMBER OF CELLS),21,10HE1GENVALUE,10)
CALL MULTI (JALT,LS,IPEN)
CALL PLOT (0.,0.,444)
CALL NOTE (35.,46.5,2.,0.0,-1,1)
CALL NOTE (43.,45.,3.,9HTNOTRAN ,0.,9,2)
CALL NOTE (35.,36.5,2.,1.0,-1,1)
CALL NOTE (43.,35.,3.,9HFEM (P=1),0.,9,2)
CALL NOTE (35.,26.5,2.,2.0,-1,1)
CALL NOTE (43.,25.,3.,9HFEM (P=2),0.,9,2)
CALL PLOT (0.,0.,777)
CALL PLOT (0.,0.,888)
CALL PLOT (0.,0.,666)
CALL CLOSE (1)
12 FORMAT (12I6)
14 FORMAT (12F6.0)
22 FORMAT (6E12.5)
STOP
END
SUBROUTINE STAY
CALL PLOT (0.,0.,777)
RETURN
END
SUBROUTINE GSYMBL (X,Y,H,BCD,THETA,N)
CALL SYMB4 (X,Y,H,BCD,THETA,N)
RETURN
END
*DLIEDRUN GRFD=ON,PGSLIB=CALL,GDBG=ON,GSMP=ON
*DATA
 3   3   2
 6   5   3
 1   1   1
 0   1   2
 1   2   3
 2   4   5   7   10  14
 2   4   5   7   10
 2   3   4
 0,9681  1,0745  1,0824  1,0859  1,0877  1,0884
 1,0080  1,0738  1,0522  1,0844  1,0866
 1,0777  1,0863  1,0868
*JEND

```

Fig. 12 Program list for Example 2

```

*NO      2309*
/
C.2/CORE 128
*P0/PAGE  40
P.0/PCH   0
/          2
PLT/
/SSL

#GJOB 1142309,FUJIMURA,T,446-01
*HFORT
C      EXAMPLE TO COMPARE THE STATEMENTS OF FORTRAN
COMMON
  DIMENSION XP(4,50),YP(4,50)
  DIMENSION NP(4),JALT(4),LS(4),IPEN(4)
  DIMENSION UND(2),UPP(2),WIDTH(2)
  READ (1)(NP(),I=1,3)
  READ (1)(JALT(),I=1,3)
  READ (1)(LS(),I=1,3)
  READ (1)(IPEN(),I=1,3)
  DO 31 I=1,3
  NP1=NP(I)
  READ (1) (XP(I,J),J=1,NP1)
  READ (1) (YP(I,J),J=1,NP1)
  31 CONTINUE
COMMON
CHANGE
C      PLOT BY F(UNCTIONAL) SERIES
  DIMENSION X(50),Y(50)
  CALL FCNT (40.,35.,1)
    CALL FRANGE (3.,4.,NP,XP,UND(1),UPP(1))
    CALL FRANGE (3.,4.,NP,YP,UND(2),UPP(2))
  DO 41 K=1,2
  UND(K)=INT(UND(K))-1
  UPP(K)=INT(UPP(K))+1
  41 WIDTH(K)=10.*(UPP(K)-UND(K))
*** BE CAREFUL OF TOO WIDE PLOT
  CALL FSEGM (0., 10., WIDTH(1),0.)
  CALL FSEGM (WIDTH(1),0.,WIDTH(1),WIDTH(2))
  CALL FSEGM (WIDTH(1),+WIDTH(2),0.,WIDTH(2))
  CALL FSEGM (0.,+WIDTH(2),0.,0.)
  DO 51 K=1,2
  NL = WIDTH(K)/10. +1
  DIST = 10.* (NL -1)
  IF (K,EQ,1) CALL FDIAL (0.,-2., 0.,DIST,NL,2.)
  IF (K,EQ,2) CALL FDIAL (0., 0.,90.,DIST,NL,2.)
  NL=WIDTH(K)/20. +1
  DIST=20.(NL-1)
  FNL =UND(K)+DIST*(UPP(K)-UND(K))/WIDTH(K)
*** BE CAREFUL OF TOO MANY DIGITS
  IF (K,EQ,1) CALL FSEGU (1.5,-5., 0.,DIST,UND(K),FNL,NL,2.,2,1)
  IF (K,EQ,2) CALL FSEGU (-3.,1.5,90.,DIST,UND(K),FNL,NL,2.,2,1)
  51 CONTINUE
  CALL FTRANS (3.,4.,NP,XP,UND(1),UPP(1)+0.,WIDTH(1)+1)
  CALL FTRANS (3.,4.,NP,YP,UND(2),UPP(2)+0.,WIDTH(2)+1)
  DO 71 I=1,3
  NP1=NP(I)
  DO 65 J=1,NP1
    X(J)=XP(I,J)
    65 Y(J)=YP(I,J)
    CALL NEWPEN (IPEN())
    CALL FFIT (A,Y,NP(I),JALT(),LS(),1)
  71 CONTINUE
  CX1*WIDTH(1)+10.
  DO 81 I=1,3
  CY1*I+.4*FLOAT(I)-1
  CALL PLOT (CX1,CY1,3)
  CALL SYMBOL (CX1+9.,CY1,2.,LS(),0.,-2)
  CALL SYMBOL (CX1+15.,CY1-1.,2.,4MCASE,0.,4)
  CALL NUMBER (CX1+25.,CY1-1.,2.,FLOAT(I),0.,-1)
  81 CONTINUE
  CALL FCNT (WIDTH(1)+105.,-35.,2)
CHANGE
  STOP
  END
*HIEDRUN GRFD=ON,PLTLIB=CALL
*PLOT
*DISKTO FO1,J2309.EX1
*JEND

```

(a) by F-Series

Fig. 13 Program list for Example 3

```

CHANGE
C      PLOT BY STDPL (SPECIAL)
CALL STDPL (3*4,NP,XP,YP,1,1,2)
CALL RANGE (UND(1),UPP(1),UND(2),UPP(2))
DO 41 K=1,2
UND(K)=INT(UND(K))-1
UPP(K)=INT(UPP(K))+1
+1 WIDTH(K)=10.*(UPP(K)-UND(K))
CALL JAL (WIDTH(1),WIDTH(2),UND(1),UPP(1),UND(2),UPP(2),1)
CALL MULTI (JALT,LS,IPEN)
CX1=WIDTH(1)*10.
DO 81 I=1,3
CY1=11.+4.0*FLOAT(I-1)
CALL PLOT (CX1,CY1,3)
CALL NOTE (CX1+9.,CY1+2.,LS(1),0.,-2,1)
CALL NOTE (CX1+15.,CY1+1.,2.,4HCASE,0.,4+2)
CALL NOTE (CX1+25.,CY1-1.,2.,FLOAT(1),0.,-1,3)
81 CONTINUE
CALL CLOSE (1).
CHANGE

```

## (b) by special plotting of STDPL

```

CHANGE
C      PLOT BY BASIC ROUTINE
DIMENSION X(52),Y(52)
CALL PLOTS (BUFF+1024)
CALL PLOT (40.,35.,-3)
UND(1)=XP(1,1)
UND(2)=YP(1,1)
UPP(1)=XP(1,1)
UPP(2)=YP(1,1)
DO 35 I=1,3
NPI=NP(1)
DO 35 J=1,NPI
IF (XP(I,J).LT.,UND(1)) UND(1)=XP(I,J)
IF (YP(I,J).LT.,UND(2)) UND(2)=YP(I,J)
IF (XP(I,J).GT.,UPP(1)) UPP(1)=XP(I,J)
IF (YP(I,J).GT.,UPP(2)) UPP(2)=YP(I,J)
35 CONTINUE
DO 41 K=1,2
UND(K)=INT(UND(K))-1
UPP(K)=INT(UPP(K))+1
41 WIDTH(K)=10.*(UPP(K)-UND(K))
C** BE CAREFUL OF TOO WIDE PLOT
DO 43 K=1,2
43 UPP(K)=(UPP(K)-UND(K))/WIDTH(K)
CALL AXIS (0.,0.,1H,-1,WIDTH(1),0.,UND(1),UPP(1),10.)
CALL AXIS (0.,0.,1H,+1,WIDTH(2),90.,UND(2),UPP(2),10.)
CALL PLOT (WIDTH(1),0.,3)
CALL PLOT (WIDTH(1),WIDTH(2),2)
CALL PLOT (0.,WIDTH(2),2)
DO 71 I=1,3
NPI=NP(1)
DO 45 J=1,NPI
X(J)=XP(I,J)
45 Y(J)=YP(I,J)
X(NPI+1)=UND(1)
X(NPI+2)=UPP(1)
Y(NPI+1)=UND(2)
Y(NPI+2)=UPP(2)
CALL NWFPFN (IPEN(1))
CALL LINE (X,Y,NPI+1,JALT(1),LS(1))
71 CONTINUE
CX1=WIDTH(1)*10.
DO 81 I=1,3
CY1=11.+4.0*FLOAT(I-1)
CALL PLOT (CX1,CY1,3)
CALL SYMBOL (CX1+9.,CY1+2.,LS(1),0.,-2)
CALL SYMBOL (CX1+15.,CY1-1.,2.,4HCASE,0.,4)
CALL NUMBFR (CX1+25.,CY1-1.,2.,FLOAT(1),0.,-1)
81 CONTINUE
CALL PLOT (WIDTH(1)+105.,-35.,-3)
CALL PLOT (0.,0.,999)
CHANGE

```

## (c) by basic plotter subroutines

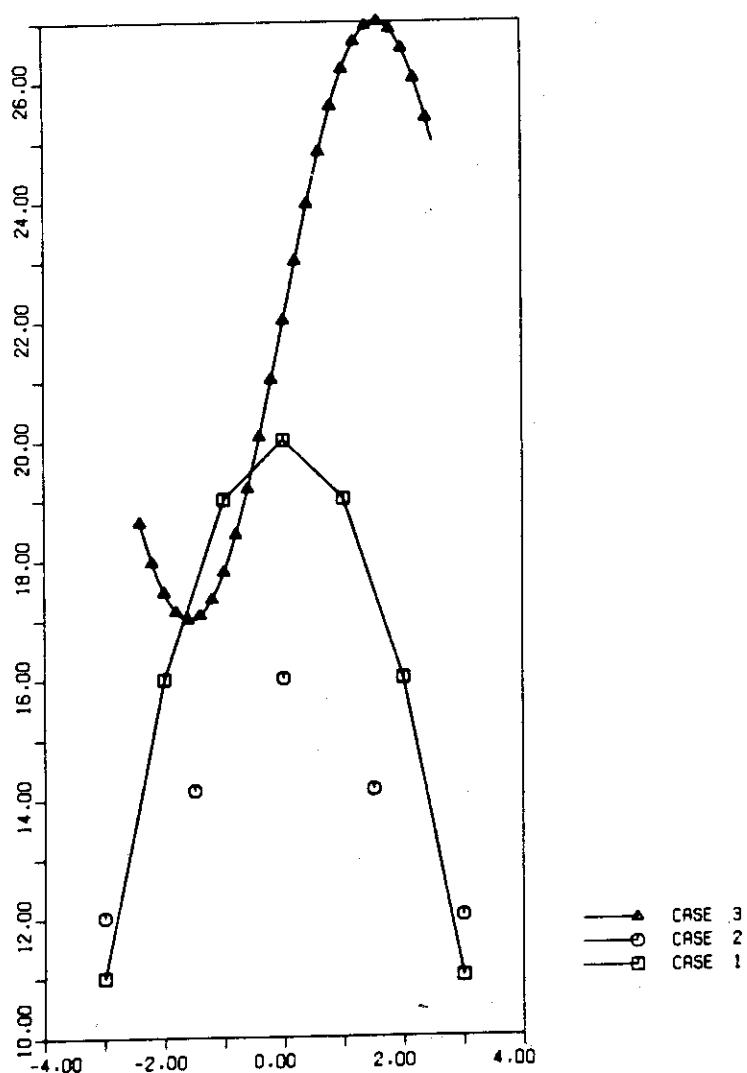


Fig. 14 Graph of Example 3 shown in Fig. 13

#NO 2309.

```

    /
C,2/CORE 128
W,0/PAGE 40
P,0/PCH 0
    /
    2
C35/
    /SSL

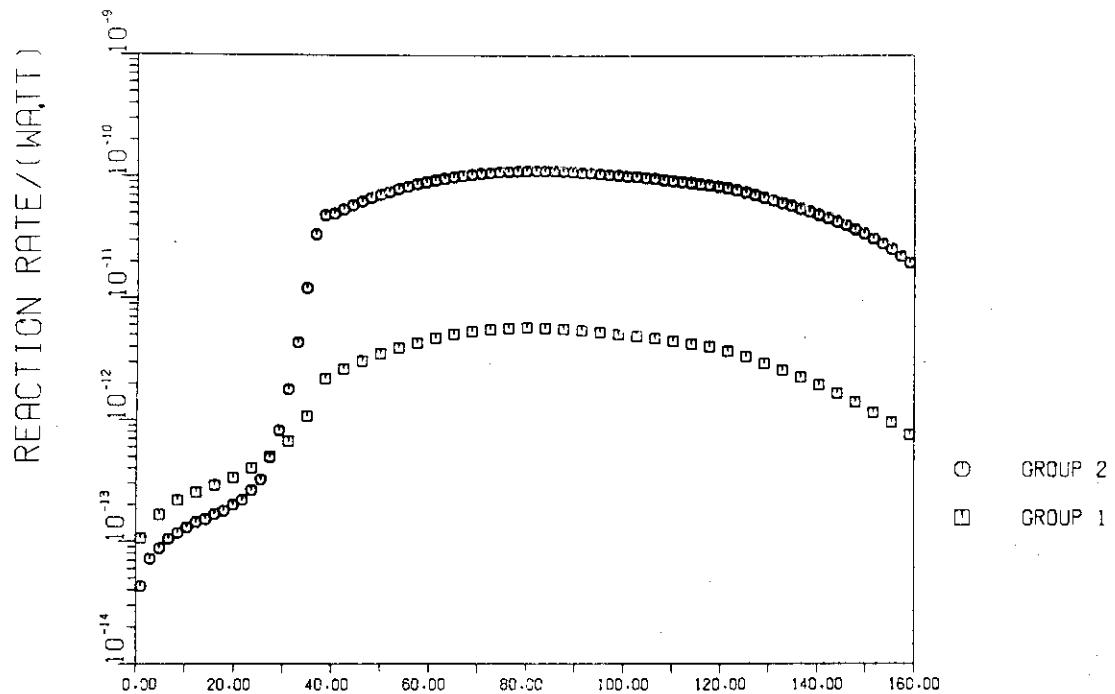
```

```

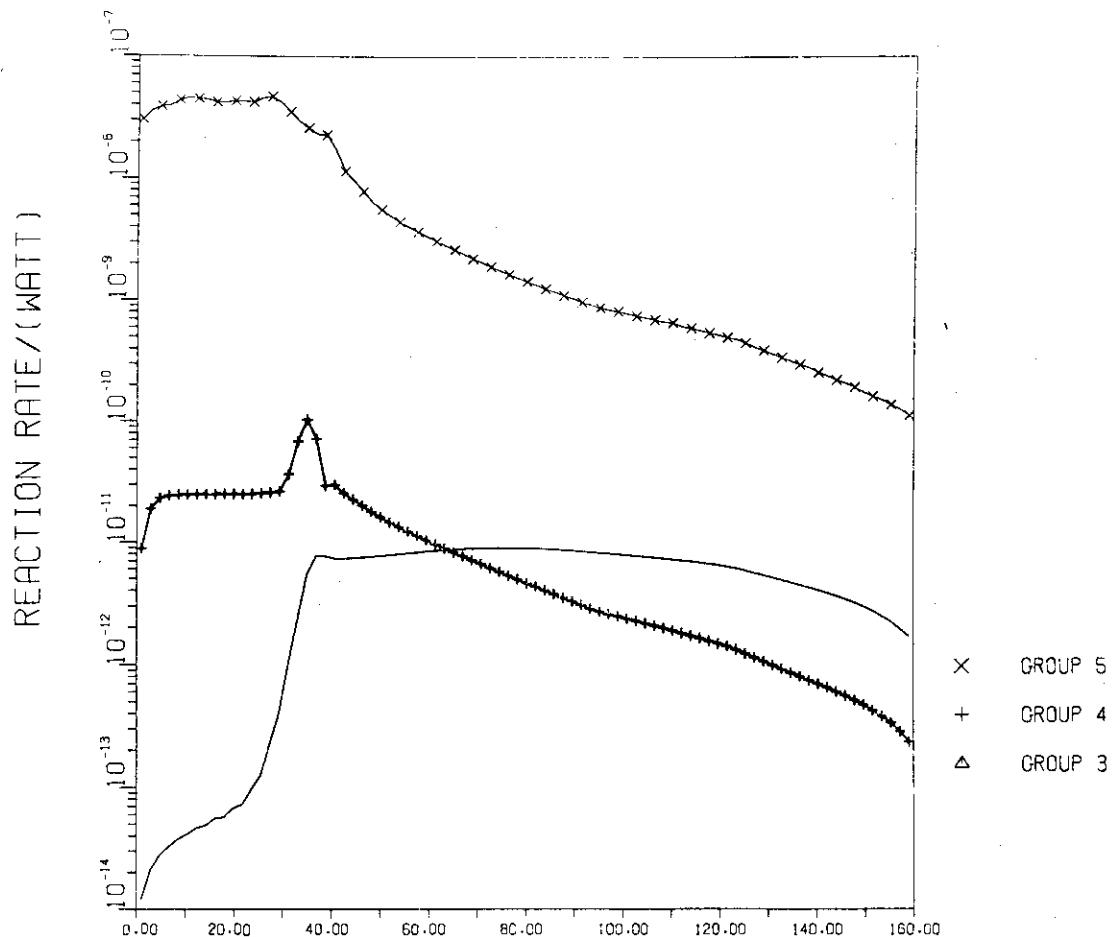
*GJOB 1142309,FUJIMURA,T,446,01
*HFORT
C EXAMPLE OF REACTION RATE
C PLOT BY STDPL (SPECIAL)
DIMENSION XP(5,100),YP(5,100)
DIMENSION NP(5),JALT(5),LS(5),IPEN(5)
JJ=-3
LL=-1
CALL PLOTS (BUFFER,1024)
DO 81 M=1,2
MC =M+1
DO 51 I=1,MC
NP(I)=85
DO 31 J=1,85
31 XP(I,J)= (160./85.)*(J-0.5)
READ (1) (YP(I,J),J=1,85)
JJ=JJ+1
JALT(I)=JJ
LL=LL+1
LS(I)=LL
IPEN(I)=1
51 CONTINUE
CALL STDPL (MC ,5,NP,XP,YP,1,2,2)
CALL RANGE (XPMIN,XPMAX,YPMIN,YPMAX)
CALL MAEX (YPMIN,YP1,IY1)
CALL MAEX (YPMAX,YP2,IY2)
IY2=IY2+1
CALL DIAL (160.,(IY2-IY1)*25.,0.,160.,10,**IY1,10,**IY2,0)
CALL TITLE (1H ,1,13HDISTANCE (CM),13,20HREACTION RATE/(WATT),20)
CALL MULTI (JALT,LS,IPEN)
DO 61 I=1,MC
CALL NOTE (160.,+10.,+20.,+10.,*1 ,3.,LS(I),0.,-1,1)
CALL NOTE (160.,+20.,+18.,5+10.,*1,3.,7H GROUP ,0.,7,2)
CALL NOTE (160.,+38.,+18.,5+10.,*1,3.,FLOAT(LS(I)+1),0.,-1,3)
61 CONTINUE
CALL CLOSE (0)
CALL PLOT (0.,0.,666)
81 CONTINUE
CALL PLOT (0.,0.,999)
66 FORMAT (2X,1P10E12,4)
STOP
END
*HLIEDRUN GRFD=ON,COMLIB=CALL
*GCOM35
*DISKTO F01,J2309,FLUX
*JEND

```

Fig. 15 Program list for Example 4



(a) First cycle



(b) Second cycle

Fig. 16 Graph of Example 4 by special plotting of STDPL  
(see Fig. 15)

## 5 おわりに

STDPL の開発に先駆け、多重構造のデータをグラフ化するプログラムの作成を手懸けたのは数年前であったが、パラメーターの設定の仕方等多くのユーザーの意見をいただいた。これらを参考にして開発した F-シリーズや STDPL が最適であるかどうかの議論は難かしく、この報告書の中でもなされていないが、開発の途中の度重なる変遷を経て特に過不足のないものになっていると確信している。

また、プロッターのエラーはふつうの FORTRAN のそれとは異質のものであり、有意な範囲にあれば多少指定したパラメーター値を修正しても、ひとまず目的に近いグラフを得ることが重要となる。エラー処理もこの観点から、可能な限り検出し、かつ初期の目的が損なわれないよう修正されるが、これも大きな特色の一つといえよう。

一旦グラフが作成されると、目盛や注釈の書き方などに不満が残るのが常である。しかし、パラメーターを増やすにも使い易さの面で限度があるので、満足のいく特殊なプロットをしたい向きは、F-シリーズや基本サブルーチンを使うことを勧めたい。著者も、中性子断面積データを BNL 第 2 版と同形にプロットしたことがあるが<sup>9)</sup>、かかる特殊なプロットは STDPL の特殊仕様のオプションをもってしても不可能である。

ユーザーから GOLOT 1,<sup>4)</sup> GPLOTZ<sup>5)</sup> と STDPL の相違についてよく尋ねられる。双方とも内挿の機能<sup>10)</sup>はないが、\* 一口で言えば、前者はデータの数が極端に多いときに適し、かつ表示方法も豊かであるのに対し、後者は簡単にデータの傾向をみたいというときに適するものと思う。

2 次元データのプロットに関連して、3 次元データの x-y 平面上でのグラフ化が想起されるが、それらについては等高線図<sup>11)</sup> や透視図<sup>12)</sup> の作成を行うプログラムも整っている。また、プロッター以外のディスプレイや COM の使用についても触れたが、詳しいことは関連する文献<sup>7)8)</sup> を参照されたい。

## 謝　　辞

この報告書を書くにあたり、原子炉数値解析研究室の筒井恒夫氏には、例題の設定や作成について少ながらぬ力添えをいただいた。中村康弘氏をはじめとする計算センター、システム・プログラム係の方々には、プロッターやディスプレイなどのソフトウェアについて色々と御教示をいただいた。また、核設計研究室の長谷川明氏には、GPLOTZ 等について伺うことが多かった。ここに謹んで謝意を表します。

\* 2 次元の図形的な意味での内挿は FFIT で行える。

## 5 おわりに

STDPL の開発に先駆け、多重構造のデータをグラフ化するプログラムの作成を手懸けたのは数年前であったが、パラメーターの設定の仕方等多くのユーザーの意見をいただいた。これらを参考にして開発した F-シリーズや STDPL が最適であるかどうかの議論は難かしく、この報告書の中でもなされていないが、開発の途中の度重なる変遷を経て特に過不足のないものになっていると確信している。

また、プロッターのエラーはふつうの FORTRAN のそれとは異質のものであり、有意な範囲にあれば多少指定したパラメーター値を修正しても、ひとまず目的に近いグラフを得ることが重要となる。エラー処理もこの観点から、可能な限り検出し、かつ初期の目的が損なわれないよう修正されるが、これも大きな特色の一つといえよう。

一旦グラフが作成されると、目盛や注釈の書き方などに不満が残るのが常である。しかし、パラメーターを増やすにも使い易さの面で限度があるので、満足のいく特殊なプロットをしたい向きは、F-シリーズや基本サブルーチンを使うことを勧めたい。著者も、中性子断面積データを BNL 第 2 版と同形にプロットしたことがあるが<sup>9)</sup>、かかる特殊なプロットは STDPL の特殊仕様のオプションをもってしても不可能である。

ユーザーから GOLOT 1,<sup>4)</sup> GPLOTZ<sup>5)</sup> と STDPL の相違についてよく尋ねられる。双方とも内挿の機能<sup>10)</sup>はないが、\*一口で言えば、前者はデータの数が極端に多いときに適し、かつ表示方法も豊かであるのに対し、後者は簡単にデータの傾向をみたいというときに適するものと思う。

2 次元データのプロットに関連して、3 次元データの x-y 平面上でのグラフ化が想起されるが、それらについては等高線図<sup>11)</sup> や透視図<sup>12)</sup> の作成を行うプログラムも整っている。また、プロッター以外のディスプレイや COM の使用についても触れたが、詳しいことは関連する文献<sup>7)8)</sup>を参照されたい。

## 謝 辞

この報告書を書くにあたり、原子炉数値解析研究室の筒井恒夫氏には、例題の設定や作成について少ながらぬ力添えをいただいた。中村康弘氏をはじめとする計算センター、システム・プログラム係の方々には、プロッターやディスプレイなどのソフトウェアについて色々と御教示をいただいた。また、核設計研究室の長谷川明氏には、GPLOTZ 等について伺うことが多かった。ここに謹んで謝意を表します。

---

\* 2 次元の図形的な意味での内挿は FFT で行える。

## 参考文献

- 1) 計算センター：“Graphic Plotter マニュアル”，（所内資料）（1970）
- 2) 計算センタ： “GRAPHIC PLOTTER マニュアル (CALCOMP 900/937/1136) ”, (所内資料) (1972)
- 3) 吉沢ビジネス・マシンズ(株)：“CALCOMP プログラミング マニュアル-II”, (1969)
- 4) 長谷川 明：“汎用グラフ作成サブルーチン GPLOT 1, 自動グラフ作成コード GPLOTC の開発”, JAERI-memo 4255 (公開) (1970)
- 5) 長谷川 明：“多重データ比較用サブルーチン GPLOTZ の開発 (汎用グラフ作成サブルーチン GPLOT 1の改良及び断面積実験データ検索システム NESTORへの応用) ”, JAERI-M 5550 (1974)
- 6) 藤村統一郎, 西田雄彦, 浅井清編：“JSSL マニュアル”, JAERI-M 7102(1977発刊予定)
- 7) 中村康弘, 小沼吉男：“グラフィック・ディスプレイによるプロッタ・プログラムのオンライン・デバッグ”, JAERI-M 6391 (1976)
- 8) 計算センター：“グラフィック COM プログラミング・マニュアル”, (所内資料) (1976)
- 9) 藤村統一郎：“中性子断面積データのプロッティングコード XPA 1, XPSL, XPBF のマニュアル”, (所内資料) (1968)
- 10) 例えば, 伊勢武治, 藤村統一郎：“最近の内挿法のアルゴリズムと計算プログラム”情報処理 Vol. 17, No.5 (1976)
- 11) 吉沢ビジネス・マシンズ(株)：“ソフトウェア・マニュアル GPCP-I”, (1972); および藤村統一郎, 山崎和彦：“等高線を描くための原研仕様プログラム (JGPCP) ”, (所内資料) (1973)
- 12) 吉沢ビジネス・マシンズ(株)：“THREE・D-II”, (1974)

## Appendix A FORTRAN Lists of F-Series

```

SUBROUTINE FRANGE (MC,MCMAX,NP,AR,ARMIN,ARMAX)
DIMENSION NP(1),AR(MCMAX,1)
IF (MC.LT.1 .OR. MC.GT.MCMAX) GO TO 41
ARMIN=AR(1,1)
ARMAX=AR(1,1)
DO 33 I=1,MC
NP1=NP(I)
IF (NP1-1) 41,33,25
25 DO 31 J=1,NP1
IF (AR(I,J).LT.ARMIN) ARMIN=AR(I,J)
31 IF (AR(I,J).GT.ARMAX) ARMAX=AR(I,J)
33 CONTINUE
GO TO 45
41 WRITE (6,42)
42 FORMAT (4X,' ERROR IN FRANGE : MC.LT.1 OR MC.GT.MCMAX OR ',1
          ' NP(I).LT.1 ')
45 RETURN
END

```

```

SUBROUTINE FTRANS (M,MDIM,N,Z,X1,X2,Y1,Y2,LL)
C      M          NO. OF CASES
C      MDIM       MAX. NO. OF CASES
C      N          NP(I) IS THE NO. OF DATA IN I-TH CASE
C      Z          Z(I,J) IS THE DATA OF J-TH POINT IN I-TH CASE
C      X1,X2,Y1,Y2
C                  TRANSFORM (X1,X2) TO (Y1,Y2)
C      LL          1/2 LINEAR/LOG(10)
C
C      DIMENSION N(1),Z(MDIM,1)
C      DIMENSION MESE(5),MESW(5)
C      DATA MESE /20H ERROR IN FTRANS, /
C      DATA MESW /20H WARNING IN FTRANS, /
C---  ERROR
  82 FORMAT (10X,5A4,' M L.T. 1 ')
  84 FORMAT (10X,5A4,' MDIM L.T. M ')
  86 FORMAT (10X,5A4,' N(I) L.T. 1 ')
C---  WARNING
  92 FORMAT (10X,5A4,' X1 G.T. X2 OR Y1 G.T. Y2, ',
  1     ' X1=X2, X2=X1 OR Y1=Y2, Y2=Y1 ')
  94 FORMAT (10X,5A4,' X1=X2 OR Y1=Y2, Z=Y1 ')
  96 FORMAT (10X,5A4,' LL N.E. 1,2, LL=1 ')
  98 FORMAT (10X,5A4,' Z(I,J) L.E. 0, WHEN LL=2, LL=1 ')
C=====  ERROR CHECK
      IF (M.GE.1) GO TO 13
      WRITE (6,82) MESE
      RETURN
  13 IF (MDIM.GE.M) GO TO 15
      WRITE (6,84) MESE
      RETURN
  15 DO 21 I=1,M
      IF (N(I).GE.1) GO TO 21
      WRITE (6,86)
      RETURN
  21 CONTINUE
      X1F=X1
      X2F=X2
      IF (X1-X2) 27,25,23
  23 WRITE (6,92) MESW
      X1F=X2
      X2F=X1
      GO TO 27
  25 WRITE (6,94) MESW
  27 CONTINUE
      Y1F=Y1
      Y2F=Y2
      IF (Y1-Y2) 37,35,33
  33 WRITE (6,92) MESW
      Y1F=Y2
      Y2F=Y1
      GO TO 37
  35 WRITE (6,94) MESW
  37 CONTINUE
      LLF=LL
      IF (LL.EQ.1 .OR. LL.EQ.2) GO TO 45
      WRITE (6,96) MESW
      LLF=1
  45 IF (LLF.EQ.1) GO TO 51

```

```

CALL FRANGE (M,MUIM,N,Z,ZMIN,ZMAX)
IF (ZMIN.GT.0.) GO TO 51
WRITE (6,98) MESN
LLF=1
51 CONTINUE
C===== TRANSFORM
IF (X1F.NE.X2F .AND. Y1F.NE.Y2F) GO TO 61
DO 55 I=1,M
NI=N(I)
DO 55 J=1,NI
55 Z(I,J)=Y1F
GO TO 81
C
61 IF(LLF,GE,2) GO TO 75
C--- LIN.
A=(Y2F-Y1F)/(X2F-X1F)
B=-A*X1F+Y1F
GO TO 77
75 C=ALOG10(X1F)
A=(Y2F-Y1F)/( ALOG10(X2F)-C)
B=-A*C+Y1F
77 DO 79 I=1,M
NI =N (I)
DO 79 J=1,NI
IF(LLF,EQ,1) Z (I,J)=A*Z (I,J)+B
IF(LLF,EQ,2) Z (I,J)=A*ALOG10(Z (I,J))+B
79 CONTINUE
81 CONTINUE
RETURN
END

```

```

SUBROUTINE FFIT (X,Y,NP,JALT,LS,ITERP)
DIMENSION X(1),Y(1)
DIMENSION MES1(4),MES2(3),MES3(3),MES4(8)
DATA MES1 /16H ERROR IN FFIT, /
DATA MES2 /12H NP L,T, 1 /
DATA MES3 /12H LS L,T, 0 /
DATA MES4 /32H ITERP N,E, 1,2 WHEN NP G.E. 3 /
14 FORMAT (4X,32A4)
IF (NP.GE.1) GO TO 23
WRITE (6,14) MES1,MES2
RETURN
23 IF (LS.GE.0) GO TO 27
WRITE (6,14) MES1,MES3
RETURN
27 IF (ITERP.EQ.1 ,OR, ITERP.EQ.2) GO TO 29
IF (NP.LT.3) GO TO 29
WRITE (6,14) MES1,MES4
RETURN
29 IF (NP.GE.3 ,AND, JALT.GE.0 ,AND, [ITERP.EQ.2]) GO TO 51
C---- LINEAR
JABS=JABS(JALT)
I=0
IF (NP.LT.2) GO TO 41
37 I=I+1
IF (JABS.EQ.0) GO TO 39
IDI=(I-1)-((I-1)/JABS)*JABS
IF (IDI.EQ.0) CALL SYMBOL (X(IDI),Y(IDI),2.0,LS,0,,-1)
39 IF (JALT.LT.0) GO TO 41
IN=I+1
CALL PLOT (X(I),Y(I),3)
CALL PLOT (X(IN),Y(IN),2)
41 IF ((NP-1).GE.2) GO TO 37
C---- LAST PT.
IF (JABS.EQ.0) GO TO 43
ISD1=NP-1-((NP-1)/JABS)*JABS
IF (ISD1.NE.0) GO TO 43
CALL SYMBOL (X(NP),Y(NP),2.0,LS,0,,-1)
43 CONTINUE
GO TO 61
C---- SMOOTH
51 CONTINUE
CALL SMOOTH (X(1),Y(1),0)
DO 55 I=2,NP-1
55 CALL SMOOTH (X(I),Y(I),-2)
CALL SMOOTH (X(NP),Y(NP),-24)
IF (JALT.EQ.0) GO TO 61
C---- SYMBOL
DO 59 I=1,NP
ISB=I+(I-1)*JALT
IF (ISB.GT.NP) GO TO 61
CALL SYMBOL (X(ISB),Y(ISB),2.0,LS,0,,-1)
59 CONTINUE
61 CONTINUE
RETURN
END

```

```

SUBROUTINE FCNT (XG,YG,MCN)
DIMENSION MES1(4),MES2(4)
DATA MES1 /16H ERROR IN FCNT,/
DATA MES2 /16H MCN N.E. 0 - 2 /
IF (MCN.GE.0 ,OR, MCN.LE.2) GO TO 23
WRITE (6,14) MES1,MES2
14 FORMAT (4X,32A4)
RETURN
23 IF (MCN=1) 31+35+39
31 CALL PLOT (XG,YG,-3)
RETURN
35 CALL PLOTS (III,1024)
CALL PLOT (XG,YG,-3)
RETURN
39 CALL PLOT (XG,YG,-3)
CALL PLOT (0.,0.,999)
RETURN
END

```

```

SUBROUTINE FSEGM (XS,YS,XE,YE)
CALL PLOT (XS,YS,3)
CALL PLOT (XE,YE,2)
RETURN
END

```

```

SUBROUTINE FDIAL (XS,YS,THETA,DIST,NL,HEIGHT)
DIMENSION MES1(4),MES2(3),MES3(4)
DATA MES1 /16H ERROR IN FDIAL,/
DATA MES2 /12H NL L.T. 1 /
DATA MES3 /16H HEIGHT L.E. 0. /
14 FORMAT (4X,32A4)
IF (NL.GE.1) GO TO 23
WRITE (6,14) MES1,MES2
RETURN
23 IF (HEIGHT.GT.0.) GO TO 27
WRITE (6,14) MES1,MES2
RETURN
27 CONTINUE
THETR=THETA*3.1415926/160.
DL=0.
IF (NL.GT.1) DL=DIST/FLOAT(NL-1)
DO 45 I=1,NL
XII=XS+DL*COS(THETR)*FLOAT(I-1)
YII=YS+DL*SIN(THETR)*FLOAT(I-1)
XII=XII-HEIGHT*SIN(THETR)
YII=YII+HEIGHT*COS(THETR)
CALL FSEGM (XII,YII,XII,YII)
45 CONTINUE
RETURN
END

```

```

SUBROUTINE FSEQU (XS,YS,THETA,DIST,FPNS,FPNE,NT,HEIGHT,NN,ISTYL)
DIMENSION MES1(4),MES2(3)
DIMENSION MES3(4),MES4(4),MESS(4)
DATA MES1 /16H ERROR IN FSEQU,/
DATA MES2 /12H NT L.T. 1 /
DATA MES3 /16H HEIGHT L.E. 0. /
DATA MES4 /16H NN N.E. -1 - 11/
DATA MES5 /16H ISTYL N.E. 1,2 /
EPS=1.0E-12
IF (NT.GE.1) GO TO 21
WRITE (6,14) MES1,MES2
14 FORMAT (4X,32A4)
RETURN
21 CONTINUE
IF (HEIGHT.GT.0.) GO TO 25
WRITE (6,14) MES1,MES3
RETURN
25 CONTINUE
IF (NN.GE.-1 .AND. NN.LE.11) GO TO 27
WRITE (6,14) MES1,MES4
RETURN
27 CONTINUE
IF (ISTYL.EQ.1 .OR. ISTYL.EQ.2) GO TO 29
WRITE (6,14) MES1,MESS
RETURN
29 CONTINUE
THETR=THETA*3.1415926/180.
DL=0.
IF (NT.NE.1) DL=DIST/FLOAT(NT-1)
DF=0.
IF (NT.NE.1) DF=(FPNE-FPNS)/FLOAT(NT-1)
DO 35 I=1,NT
FF=FPNS + DF*FLOAT(I-1)
AFF=A3S(FF)*(1.0+EPS)
IF (FF) 31,35,39
31 CONTINUE
NN3=NN+3
NDIGIT=ALOG10(AFF)+NN3
AFF=-AFF
IF (NDIGIT.LT.NN3) NDIGIT=NN3
GO TO 41
35 CONTINUE
NDIGIT=NN+2
GO TO 41
39 CONTINUE
NN2=NN+2
NDIGIT=ALOG10(AFF)+NN2
IF (NDIGIT.LT.NN2) NDIGIT=NN2
41 CONTINUE
XI=XS+DL*COS(THETR)*FLOAT(I-1)
YI=YS+DL*SIN(THETR)*FLOAT(I-1)
DGL=HEIGHT*(3.0*FLOAT(NDIGIT)-2.0)/7.0
IF (ISTYL.NE.1) GO TO 43
XI=XI-0.5*DGL*COS(THETR)
YI=YI-0.5*DGL*SIN(THETR)
GO TO 45
43 XI=XI-DGL

```

YI=YI+J,5\*HEIGHT  
45 CONTINUE  
THETB=THETA  
IF (ISTYL,EQ,2) THETB=0,  
CALL NUMBER (XI,YI,HEIGHT,AFF,THETB,NN)  
55 CONTINUE  
RETURN  
END

## Appendix B FORTRAN Lists of STDPL

SUBROUTINE STDPL (MC,MCMAX,NP,XP,YP,LLX,LLY,IDS)

```

=====
C
CCCCCC  STANDARD PLOT -----
C          PLOT THE DATA COMPOSED OF MULTI-CASE SUCH AS EACH
C          DATA POINT HAS X,Y-COMPONENT
C          DEVELOPED BY T. FUJIMURA MAY 12, 1972
C          REVISED FEB. 2 1977
C
C          MC   --- NUMBER OF CASE (MC,GE,1)
C          MCMAX --- MAXIMUM OF MC TO BE EXPECTED.
C                      THIS IS THE FIRST ARRAY CONST. OF XP,YP
C          NP(NDN) -           NP(I)=NUMBER OF POINTS OF I-TH CASE
C                      IF MC=1, NP MAY HAVE NO DIM.
C                      (NP(I),GE,1)
C          XP(MCMAX,NDX),YP(MCMAX,NDY)
C          --- (XP(I,J),YP(I,J)) IS X,Y COMPONENT OF J-TH POINT OF
C          I-TH CASE.
C          ARRAY WHICH IS ASSIGNED TO XP MUST HAVE A
C          CONSTANT DIMENSION (MCMAX,NDX) WHERE
C          NDX G.E. MAX. OF NP(I),I=1,MC (ALSO TO NDY)
C          IF MC=1, THESE ARRAY MAY BE 1-DIM
C          LLX   --- SCALING AND SECTION LINE ON X-AXIS
C          1/2 LINEAR/LOG IN IDS=1
C          1/2/-1/-2 ABS MEANS LIN. OR LOG AND NEGATIVE MEANS
C          TO DRAW Y-AXIS OR DECADE LINES IN IDS=2 OR 3
C          LLY   --- SAME AS LLX
C          IDS   --- IDENTIFY STANDARD OR SPECIAL PLOT
C          1/2/3 STANDARD/SPECIAL WITH ORDINARY SHEET/SPECIAL
C          WITH LARGE SHEET
C
C          DIMENSION NP(1),XP(MCMAX,1),YP(MCMAX,1)
C          DIMENSION BUFFER(1)
C          DIMENSION MES 1( 3),MES 2( 4),MES 3( 7),MES 4(10)
C          DIMENSION MES 6( 7),MES 7( 7),MES 8( 8),MES 9( 7),MES10( 7)
C          DIMENSION MES11( 8)           ,MES13( 7),MES14( 3),MES15( 3)
C          DIMENSION MES16( 3),MES17(12),MES18( 8),MES19( 5),MES20( 4)
C          DIMENSION MES21( 5),MES22( 4)           ,MES25( 4)
C          DIMENSION MES29( 3),MES30( 7)           ,MES33( 5)
C          DIMENSION MES35(11)           ,MES36( 8)
C          DIMENSION MES41( 9),MES42( 9),MES43( 9)
C          DIMENSION MES51( 5),MES52( 6)
C          DIMENSION MES57( 6),MES58( 8),MES59( 7),MES60( 7)
C          DIMENSION MES65( 8)           ,MES66(11),MES67( 8),MES68(11)
C          DIMENSION MES82( 7),MES83( 6),MES84( 7),MES85( 5)
C          DIMENSION MES86( 8)           ,MES88( 4)
C
C          DIMENSION JALT(1),LS(1)
C          DIMENSION IPEN(1)
C          DIMENSION XX(4),YY(4),XR(4),YR(4)
C
C          EQUIVALENCE (LTE,FTA)
C
C          DATA MES 1/12H MC L.E. 0 /
C          DATA MES 2/16H NP(I) L.T. 1 /

```

DATA MES 3/28H IDSS N.E. 1 TO 3. IDSS=1 /  
 DATA MES 4/40H WIDTH L.E. 0. OR G.T. 1000.. WIDTH=200./  
 DATA MES 6/28H XLW G.T. XPMIN. XLW=XPMIN /  
 DATA MES 7/28H XUP L.T. XPMAX. XUP=XPMAX /  
 DATA MES 8/32H XLW=XUP. XPC(I,J) SCALED TO 0. /  
 DATA MES 9/28H YLW G.T. YPMIN. YLW=YPMIN /  
 DATA MES10/28H YUP L.T. YPMAX. YUP=YPMAX /  
 DATA MES11/32H YLW=YUP. YPC(I,J) SCALED TO 0. /  
 DATA MES13/28H MOPEN N.E. 0 OR 1. MOPEN=1 /  
 DATA MES14/12H NG L.E. 0 /  
 DATA MES15/12H NX L.E. 0 /  
 DATA MES16/12H NY L.E. 0 /  
 DATA MES17/48H LS(I) N.E. 0 TO 127. LS(I)=MOD(ABS(LS(I)),128) /  
 DATA MES18/32H MCLOSE N.E. 0 OR 1. MCLOSE=1 /  
 DATA MES19/20H WARNING IN STDPL. /  
 DATA MES20/16H ERROR IN STDPL./  
 DATA MES21/20H OMIT PLOT THIS PART/  
 DATA MES22/16H PREVIOUS ERROR /  
 DATA MES25/16H MCMAX L.T. MC /  
 DATA MES29/12H HTA EQ. 0. /  
 DATA MES30/28H HTA L.T. 0.. HTA=ABS(HTA) /  
 DATA MES33/20H PROTRUDED THE RANGE/  
 DATA MES35/44H HEIGHT L.E. 0. OR G.T. 200. IN IDSS=2 OR /  
 DATA MES36/32H G.T. 750. IN IDSS=3. HEIGHT=150./  
 DATA MES41/36H NG IS TOO LARGE. NG IS DECREASED /  
 DATA MES42/36H NX IS TOO LARGE. NX IS DECREASED /  
 DATA MES43/36H NY IS TOO LARGE. NY IS DECREASED /  
 DATA MES51/20H WRONG ORDER CALLING/  
 DATA MES52/24H DOUBLE TAPE OPERATION /  
 DATA MES57/24H IPEN(I) N.E. 1 TO 3. /  
 DATA MES58/32H IPEN(I)=MOD(ABS(IPEN(I)),3)+1 /  
 DATA MES59/28H LLX=2 IN XLW L.E. 0.. LLX=1/  
 DATA MES60/28H LLY=2 IN YLW L.E. 0.. LLY=1/  
 DATA MES65/32H LLX N.E. 1 OR 2 IN IDSS=1. OR /  
 DATA MES66/44H ABS(LLX) N.E. 1 OR 2 IN IDSS=2 OR 3. LLX=1 /  
 DATA MES67/32H LLY N.E. 1 OR 2 IN IDSS=1. OR /  
 DATA MES68/44H ABS(LLY) N.E. 1 OR 2 IN IDSS=2 OR 3. LLY=1 /  
 DATA MES82/28H LTA N.E. 0 TO 127 IN MTA=1./  
 DATA MES83/24H LTA=MOD(ABS(LTA),128) /  
 DATA MES84/28H NTA N.E. -1 OR -2 IN MTA=1 /  
 DATA MES85/20H NTA L.T. 1 IN MTA=2/  
 DATA MES86/32H NTA L.T. -1 OR G.T. 11 IN MTA=3/  
 DATA MES88/16H MTA N.E. 1 TO 3/

```

C
      DATA ICALL/0/,MTCONT/0/,IERR/0/
12 FORMAT (4X,32A4)
14 FORMAT (10X,' X-AXIS (' , ',F7.1,' , ',F7.1,' ), Y-AXIS (' , F7.1,
1      ' , ',F7.1,' ) ')
C---      PREV. ERROR CHECK
      IF (IERR,EQ,0) GO TO 213
      WRITE (6,12) MES20,MES22
213 CONTINUE
C---      CALL CHECK
      IF (MOD(ICALL,7),EQ,0) GO TO 5
      WRITE (6,12) MES20,MES51
      IERR=1
5 ICALL=1
  
```

```

C--- ARG. CHECK
IF (MC.GE.1) GO TO 23
WRITE (6,12) MES20,MES 1
IERR=1
23 CONTINUE
IF (MCMAX.GE.MC) GO TO 24
WRITE (6,12) MES20,MES25
IERR=1
24 CONTINUE
IDSSF=IDSS
IF (IDSS.GE,1 ,AND.IDSS,LE,3) GO TO 21
WRITE (6,12) MES19,MES3
IDSSF=1
21 CONTINUE
LLXF=LLX
IAZ1 =IABS(LLX)
IF (IAZ1 .LT.1 ,OR . IAZ1 .GT.2) GO TO 119
IF (IDSSF.GE.2) GO TO 121
IF (LLX.EQ.1 .OR. LLX.EQ.2) GO TO 121
119 WRITE (6,12) MES19,MES65,MES66
LLXF=1
121 CONTINUE
LLYF=LLY
IAY1=IABS(LLY)
IF (IAY1.LT.1 ,OR . IAY1.GT.2) GO TO 123
IF (IDSSF.GE.2) GO TO 129
IF (LLY.EQ.1 .OR. LLY.EQ.2) GO TO 129
123 WRITE (6,12) MES19,MES67,MES68
LLYF=1
129 CONTINUE
IF (IERR.GE.1) GO TO 37
DO 25 I=1,MC
NP1=NP(I)
IF (NP1.GE.1) GO TO 25
WRITE (6,12) MES20,MES2
IERR=1
25 CONTINUE
C--- MIN,MAX OF XP,YP
CALL FRANGE (MC,MCMAX,NP,XP,XPM1,XPMA)
CALL FRANGE (MC,MCMAX,NP,YP,YPM1,YPMA)
C--- CONSTANTS THROUGH STDPL
XT1=-25.
YT1=-20.
NSMB=128
C--- CONSTANTS FOR STANDARD PLOT
XWF=200.
XT2=XWF+80.
YHF=150.
YT2=YHF+15.
XLWF=XPM1
XUPF=XPMA
YLWF=YPM1
YUPF=YPMA
MOPF=1
ITIT=0
IF (IDSSF.EQ.1) GO TO 78
MCLSF=1

```

```

37 CONTINUE
    RETURN
C
C     ENTRY RANGE (XPMIN,XPMAX,YPMIN,YPMAX)
C=====
C     XPMIN,XPMAX
C     --- MIN. AND MAX. OF XP(I,J) WHERE I=1,MC J=1,NP(I)
C
C     ALSO TO YPMIN,YPMAX
C
C---    RREV. ERROR CHECK
    IF (IERR.LE.0) GO TO 38
    WRITE (6,12) MES20,MES22
38 CONTINUE
C---    CALL CHECK
    IF (ICALL.EQ.1) GO TO 131
    WRITE (6,12) MES20,MES51
    IERR=1
131 ICALL=2
    IF (IERR.GE.1) GO TO 39
    XPMIN=XPMI
    XPMAX=XPMA
    YPMIN=YPMI
    YPMAX=YPMA
39 RETURN
C
C     ENTRY DIAL (WIDTH,HEIGHT,XLW,XUP,YLW,YUP,      MOPEN)
C=====
C
C     WIDTH   --- LENGTH OF X (AZIMUTAL) DIRECTION (MM)
C     HEIGHT  --- LENGTH OF Y (VERTICAL) DIRECTION (MM)
C     XLW,XUP -- LOWER AND UPPER BOUND TO XP
C               IT MUST BE XLW.LE.XPMIN, XUP.GE.XPMAX
C               ALSO TO YLW,YUP
C     MOPEN   --- 1 -- OPEN PLOTTER TAPE
C               0 -- NOT OPEN
C---    PREV. ERROR / CALL CHECK
    IF (IERR.EQ.0) GO TO 231
    WRITE (6,12) MES20,MES22
231 IF (ICALL.GT.0 .AND. ICALL.LT.3) GO TO 141
    WRITE (6,12) MES20,MES51
    IERR=1
141 ICALL=3
C---    ARG. CHECK
    XWF=WIDTH
    IF (WIDTH.GT.0. .AND.WIDTH.LE.1000.) GO TO 41
    WRITE (6,12) MES19,MES4
    XWF=200.
41 CONTINUE
    XT2=XWF+80.
57 MOPF=MOPEN
    IF (MOPEN.EQ.0 .OR. MOPEN.EQ.1) GO TO 59
    WRITE (6,12) MES19,MES13
    MOPF=1
59 CONTINUE
    IF (IERR.GE.1) GO TO 385
    YHF=HEIGHT

```

```

IF (HEIGHT.LE.0. .OR. HEIGHT.GT.750.) GO TO 331
IF (IDSS.EQ.3 .OR. HEIGHT.LE.200.) GO TO 43
331 WRITE (6,12) MES19,MES35,MES36
YHF=150.
43 CONTINUE
YT2=YHF+15.
XLWF=XLW
IF (XLW.LE.XPMI ) GO TO 45
WRITE (6,12) MES19,MES6
XLWF=XPMI
45 XUPF=XUP
IF (XUP.GE.XPMA ) GO TO 47
WRITE (6,12) MES19,MEST
XUPF=XPMA
47 CONTINUE
49 YLWF=YLW
IF (YLW.LE.YPMI ) GO TO 51
WRITE (6,12) MES19,MES9
YLWF=YPMI
51 YUPF=YUP
IF (YUP.GE.YPMA ) GO TO 53
WRITE (6,12) MES19,MES10
YUPF=YPMA
53 CONTINUE
IF (LLXF.EQ.1 .OR. XLWF.GT.0.) GO TO 381
WRITE (6,12) MES19,MES59
LLXF=1
381 IF (LLYF.EQ.1 .OR. YLWF.GT.0.) GO TO 385
WRITE (6,12) MES19,MES60
LLYF=1
385 CONTINUE
RETURN

```

C

ENTRY TITLE (LGT,NG,LXT,NX,LYT,NY)

```

=====
C
C      LGT    --- TITLE OF GRAPH
C      NG     --- NUMBER OF CHAR. IN LGT
C      LXT    --- TITLE OF X AXIS
C      NX     --- NUMBER OF CHAR. IN LXT
C          ALSO TO LYT,NY
C---   PREV. ERROR / CALL CHECK
IF (IERR.EQ.0) GO TO 241
WRITE (6,12) MES20,MES22
241 IF (ICALL.GE.1 .AND. ICALL.LE.3) GO TO 145
WRITE (6,12) MES20,MES51
IERR=1
145 ICALL=4
IERTG=0
IERTX=0
IERTY=0
IF (NG.GE.1) GO TO 71
WRITE (6,12) MES19,MES14,MES21
IERTG=1
71 IF (NX.GE.1) GO TO 73
WRITE (6,12) MES19,MES15,MES21
IERTX=1

```

```

73 IF (NY.GE.1) GO TO 75
  WRITE (6,12) MES19,MES16,MES21
  IERTY=1
75 CONTINUE
  IF (IERR.GE.1) GO TO 76
C---  CONSTANTS FOR TITLE
  ITIT=1
  IF (IERTG.GE.1) GO TO 461
  DAT=8.0
  GRA=1.0/8.0
  HGT=7.0
  NGF=NG
  GTX=GRA*XWF
  GTY=YHF+DAT
  NG1=(7.0*(XT2-GTX)+2.0*HGT)/(6.0*HGT)
  IF (NG.LE.NG1) GO TO 461
  WRITE (6,12) MES19,MES41
  NGF=NG1
  IF (IERTX.GE.1 .AND. IERTY.GE.1) GO TO 76
  HAT=5.0
  RAT=1.0/3.0
  IF (IERTX.GE.1) GO TO 471
  NXF=NX
  XTX=RAT*XWF
  XTY=YT1
  NX1=(7.0*(XT2-XTX)+2.0*HAT)/(6.0*HAT)
  IF (NX.LE.NX1) GO TO 471
  WRITE (6,12) MES19,MES42
  NXF=NX1
  71 IF (IERTY.GE.1) GO TO 76
  NYF=NY
  YTX=XT1+HAT
  YTY=RAT*YHF
  NY1=(7.0*(YT2-YTY)+2.0*HAT)/(6.0*HAT)
  IF (NY.LE.NY1) GO TO 76
  WRITE (6,12) MES19,MES43
  NYF=NY1
76 CONTINUE
  RETURN

```

```

C   ENTRY MULTI (JALT,LS,IPEN)
C=====
C
C   JALT(NDJ)
C     --- JALT(I) IS A ALTERNATE NUMBER OF DATA POINT OF
C           I-TH CASE TO PLOT EACH SYMBOL (SEE PLOTTER MANUAL)
C           NDJ IS THE ARRAY CONST. OF JALT
C   LS(NDL) -- LS(I) IS A SYMBOL MARK NUMBER OF I-TH CASE
C           (WITH RESPECT SYMBOL, SEE PLOTTER MANUAL)
C           NDL IS THE ARRAY CONST. OF LS
C   IPEN(NDL) --- THE KIND OF PEN USED FOR I-TH CASE
C           NDL IS THE ARRAY CONST. OF IPEN
C---  PREV. ERROR /CALL CHECK
  IF (IERR.EQ.0) GO TO 251
  WRITE (6,12) MES20,MES22
251 IF (ICALL.GE.1 .AND. ICALL.LE.4) GO TO 171
  WRITE (6,12) MES20,MES51

```

```

IERR=1
171 ICALL=5
    IF (IERR.GE.1) GO TO 127
C---   ARG. CHECK
    DO 77 I=1,MC
        IF (LS(I).GE.0 .AND. LS(I).LT.NSMB) GO TO 77
        WRITE (6,12) MES19,MES17
77 CONTINUE
    DO 511 I=1,MC
        IPI=IPEN(I)
        IF (IPI.GE.1 .AND. IPI.LE.3) GO TO 511
        WRITE (6,12) MES19,MES57
511 CONTINUE
78 CONTINUE
C---   SCALE
    IF (XLWF.LT.XUPF) GO TO 515
    WRITE (6,12) MES19,MES 8
    DO 513 I=1,MC
        NPI=NPC(I)
        DO 513 J=1,NPI
513 XP(I,J)=0.
        GO TO 525
515 CALL FTRANS (MC,MCMAX,NP,XP,XLWF,XUPF,0.,XWF,LLXF)
525 IF (YLWF.LT.YUPF) GO TO 529
        WRITE (6,12) MES19,MES11
        DO 527 I=1,MC
            NPI=NPC(I)
            DO 527 J=1,NPI
527 YP(I,J)=0.
        GO TO 541
529 CALL FTRANS (MC,MCMAX,NP,YP,YLWF,YUPF,0.,YHF,LLYF)
541 CONTINUE
C---   MT OPEN
    IF (MTCNT.EQ.0) GO TO 153
    IF (MTCNT.EQ.1 .AND. MOPF.EQ.0) GO TO 531
    WRITE (6,12) MES20,MES52
    IERR=1
    GO TO 127
153 CONTINUE
    IF (MOPF.NE.0) CALL FCNT (0.,0.,1)
    MTCNT=1
531 CONTINUE
    CALL NEWPEN (1)
    CALL FCNT (40.,35.,0)
C---   DIAL
    IF (IABS(LLXF).NE.1) GO TO 555
    DX1=(XUPF-XLWF)/XWF
    CALL AXIS (0.,0.,1H ,-1,XWF,0.,XLWF,DX1 ,10.0)
    IF (LLXF.GT.0 .OR. XLWF.GE.0. .OR. XUPF.LE.0.) GO TO 563
    XZ0=-XLWF/DX1
    CALL FSEGM (XZ0,0., XZ0,YHF)
    GO TO 563
555 CALL LGAXS(0.,0.,1H ,-1,XWF,0.,XLWF,ALOG10(XUPF/XLWF)/XWF)
    IF (LLXF.GT.0) GO TO 563
    CALL MAEX (XLWF,AM1,IE1)
    CALL MAEX (XUPF,AM2,IE2)
    IF (IE1.GE.IE2) GO TO 563

```

```

DO 557 I=IE1+1,IE2
TX1=10**I
CALL FTRANS (1,1,1,TX1,XLWF,XUPF,0.,XWF,2)
557 CALL FSEGM (TX1,0.,TX1,YHF)
563 CONTINUE
IF (IABS(LLYF).NE.1) GO TO 565
DY1=(YUPF-YLWF)/YHF
CALL AXIS
1 (0.,0.,1H ,+1,YHF,90.,YLWF,DY1,10,0)
IF (LLYF.GT.0 .OR. YLWF.GE.0. .OR. YUPF.LE.0.) GO TO 573
YZ0=-YLWF/DY1
CALL FSEGM (0.,YZ0,XWF,YZ0)
GO TO 573
565 CALL LGAXS (0.,0.,1H ,1,YHF,90.,YLWF,ALOG10(YUPF/YLWF)/YHF)
IF (LLYF.GT.0) GO TO 573
CALL MAEX (YLWF,AM1,IE1)
CALL MAEX (YUPF,AM2,IE2)
IF (IE1.GE.IE2) GO TO 573
DO 567 I=IE1+1,IE2
TY1=10**I
CALL FTRANS (1,1,1,TY1,YLWF,YUPF,0.,YHF,2)
567 CALL FSEGM (0.,TY1,XWF,TY1)
573 CONTINUE
CALL FSEGM (0.,YHF,XWF,YHF)
CALL FSEGM (XWF,YHF,XWF,0.)
CALL STAY
83 IF (IDSSF.EQ.1 .OR. ITIT.LE.0) GO TO 85
C--- TITLE
IF (IERTG.EQ.0)
1CALL SYMBOL (GTX,GTY,HGT,LGT,0.,NGF)
IF (IERTX.EQ.0)
1CALL SYMBOL (XTX,XTY,HAT,LXT,0.,NXF)
IF (IERTY.EQ.0)
1CALL SYMBOL (YTX,YTY,HAT,LYT,90.,NYF)
85 CONTINUE
C--- MULTI LINES
DO 587 I=1,MC
C--- SET TO EACH I
NPI=NP(I)
IF (IDSSF.GE.2) GO TO 561
JALTF= (NPI-(MOD(NPI-1,20)+1))/20 +1
LSF=MOD(I,NSMB)
GO TO 86
561 CONTINUE
JALTF=JALT(I)
IALS=IABS(LS(I))
LSF=MOD(IALS,NSMB)
IPENF=MOD(IABS(IPEN(I)),3)+1
CALL NEWPEN (IPENF)
86 CONTINUE
JABS=IABS(JALTF)
J=0
IF (NPI.EQ.1) GO TO 91
C--- ROOP
87 J=J+1
XJ= XP(I,J)
YJ= YP(I,J)

```

```

IF (JABS.EQ.0) GO TO 89
IDJ=(J-1)-((J-1)/JABS)*JABS
IF (IDJ.EQ.0) CALL SYMBOL (XJ,YJ,2.0,LSF,0.,-1)
89 IF (JALTF.LT.0) GO TO 91
JN=J+1
XJN= XP(I,JN)
YJN= YP(I,JN)
CALL PLOT (XJ,YJ,3)
CALL PLOT (XJN,YJN,2)
91 IF ((NPI-J).GE.2) GO TO 87
C--- LAST PT.
IF (JABS.EQ.0) GO TO 93
ISDJ=NPI-1-((NPI-1)/JABS)*JABS
IF (ISDJ.NE.0) GO TO 93
XN= XP(I,NPI)
YN= YP(I,NPI)
CALL SYMBOL (XN,YN,2.0,LSF,0.,-1)
CALL STAY
93 CONTINUE
CALL STAY
587 CONTINUE
CALL NEWPEN (1)
IF (IDSSF.GE.2) GO TO 127
C--- MARK TABLE
LMF=YT2/4.0 +2.0
IF (MC.LE.LMF) LMF=MC
DO 125 L=1,LMF
CX1=X#F+10.
CY1=10.0+ 4.0*FLOAT(L-1) +1.0
CALL PLOT (CX1,CY1,3)
CALL SYMBOL (CX1+9.,CY1,2.,L,0.,-2)
CALL SYMBOL
1      (CX1+15.,CY1-1.,2.,4HCASE,0.,4)
CALL NUMBER (CX1+25.,CY1-1.,2.,FLOAT(L),0.,-1)
125 CONTINUE
CALL STAY
GO TO 161
127 CONTINUE
RETURN
ENTRY NOTE (XTA,YTA,HTA,LTA,STA,NTA,MTA)
=====
C      XTA --- X CO-ORDINATE OF LEFT-BOTTOM POINT OF THE NOTE
C      YTA --- Y CO-ORDINATE OF LEFT-BOTTOM POINT OF THE NOTE
C      HTA --- HEIGHT OF THE NOTE
C      LTA --- SYMBOL NO., CHARACTERS OR REAL NUMBER
C      STA --- ANGLE TO X-AXIS
C      NTA --- SYMBOL PLOT STYLE OR NUMBER OF CHARACTERS OR DIGITS
C      MTA --- 1/2/3 PLOT SYMBOL/CHARACTERS/NUMBER
C
C--- PREV. ERROR / CALL CHECK
IF (IERR.EQ.0) GO TO 611
WRITE (6,12) MES20,MES22
611 IF (ICALL.GE.5 .AND. ICALL.LE.6) GO TO 621
WRITE (6,12) MES20,MESS1
IERR=1
621 ICALL=6
C--- ARG. CHECK

```

```

IERRTA=0
HTAF=ABS(HTA)
IF (HTA.GT.0.) GO TO 625
IF (HTA.LT.0.) GO TO 623
WRITE (6,12) MES19,MES29
IERRTA=1
GO TO 625
623 WRITE (6,12) MES19,MES30
625 CONTINUE
IF (MTA.GE.1 .AND. MTA.LE.3) GO TO 627
WRITE (6,12) MES19,MES88
IERRTA=1
GO TO 159
627 IF (MTA-2) 631,633,635
631 IF (NTA.EQ.-1 .OR. NTA.EQ.-2) GO TO 133
WRITE (6,12) MES19,MES84
IERRTA=1
GO TO 159
633 IF (NTA.GE.1) GO TO 133
WRITE (6,12) MES19,MES85
IERRTA=1
GO TO 159
635 IF (NTA.GE.-1 .AND. NTA.LE.11) GO TO 133
WRITE (6,12) MES19,MES86
IERRTA=1
GO TO 159
133 CONTINUE
IF (IERR.GE.1) GO TO 159
IF (MTA.NE.1) GO TO 137
LTAF=LTA
IF (LTA.GE.0 .AND. LTA.LT.NSMB) GO TO 137
WRITE (6,12) MES19,MES82,MES83
LTAF=MOD(IABS(LTA ),NSMB)
137 CONTINUE
IF (IERRTA.GE.1) GO TO 159
C--- RANGE TO BE PLOT BY 'TABLE'
IF (MTA.NE.1) GO TO 641
C--- MARK
XX(1)=XTA-0.5*HTAF
XX(3)=XX(1)
XX(2)=XTA+0.5*HTAF
XX(4)=XX(2)
YY(1)=YTA-0.5*HTAF
YY(2)=YY(1)
YY(3)=YTA+0.5*HTAF
YY(4)=YY(3)
GO TO 151
641 NTAF=NTA
IF (MTA.EQ.2) GO TO 645
C--- NUMBER
LTE=LTA
CALL MAEX (FTA,FT1,IF1)
IF (IF1.LT.0) IF1=0
IF (FTA.LT.0) IF1=IF1+1
NTAF=NTAF+IF1+2
C--- HOLLERITH
645 CONTINUE

```

```

XX(1)=XTA
XX(3)=XTA
XX(2)=XTA+HTAF*(6./7.)*NTAF-HTAF*(2./7.)
XX(4)=XX(2)
YY(1)=YTA
YY(2)=YY(1)
YY(3)=YTA+HTAF
YY(4)=YY(3)
151 CONTINUE
C---      ROTATION
DO 155 KR=1,4
RAD=STA*3.1415926/180,
XR(KR)=XX(KR)*COS(RAD)-YY(KR)*SIN(RAD)
YR(KR)=XX(KR)*SIN(RAD)+YY(KR)*COS(RAD)
IF (XR(KR),LT.XT1 ,OR, XR(KR),GT.XT2) IERRTA=1
IF (YR(KR),LT.YT1 ,OR, YR(KR),GT.YT2) IERRTA=1
155 CONTINUE
IF (IERRTA,EQ.0) GO TO 157
WRITE (6,12) MES19,MES33
WRITE (6,14) XT1,XT2,YT1,YT2
IERRTA=1
159 CONTINUE
IF (IERRTA,GE.1) WRITE (6,12) MES21
IF (CIERR.GE.1 .OR, IERRTA,GE.1) GO TO 695
157 CONTINUE
IF (MTA,EQ.1) CALL SYMBOL (XTA,YTA,HTAF,LTA,STA,NTA )
IF (MTA,EQ.2) CALL SYMBOL (XTA,YTA,HTAF,LTA ,STA,NTA )
IF (MTA,EQ.3) CALL NUMBER (XTA,YTA,HTAF,FTA ,STA,NTA )
695 RETURN
ENTRY CLOSE (MCLOSE)
=====
C      MCLOSE --- 1 -- CLOSE PLOTTER TAPE AFTER CALLING THIS ENTRY
C      0 -- NOT CLOSE
C---      PREV. ERROR / CALL CHECK
IF (CIERR,EQ.0) GO TO 713
WRITE (6,12) MES20,MES22
713 IF (ICALL.GE.5 .AND. ICALL.LE.6) GO TO 715
WRITE (6,12) MES20,MES51
IERR=1
715 ICALL=7
C---      ARG. CHECK
MCLSF=MCLOSE
IF (MCLOSE,EQ.0 .OR.MCLOSE,EQ.1) GO TO 79
WRITE (6,12) MES19,MES18
MCLSF=1
79 CONTINUE
IF (IERR,GE,1) GO TO 99
161 CONTINUE
CALL FCONT (XWF+105.,-35.,0)
C---      SHIFT
95 CONTINUE
CALL NEWPEN (1)
IF (MCLSF,NE.1) GO TO 99
CALL FCONT (0.,0.,2)
MTCONT=2
99 RETURN
END

```

SUBROUTINE STAY  
RETURN  
END

## Appendix C Computer Core Memory Required for Each Subroutine

Subroutine	Core memory (word)
FRANGE	244
FTRANS	534
FFIT	442
FCONT	162
FDIAL	186
FSEGM	74
FSEQU	412
STDPL	3580
STAY	26

Appendix D Table of Available Characters in FFIT, MULTI and NOTE<sup>2)</sup>

0	□	16		32		48	○	64	◎	80	P	96	}	112	Σ
1	○	17		33	△	49	1	65	A	81	Q	97	{	113	□
2	△	18	^	34	▽	50	2	66	B	82	R	98	μ	114	≤
3	+	19	≡	35	#	51	3	67	C	83	S	99	π	115	≥
4	×	20	→	36	\$	52	4	68	D	84	T	100	∅	116	△
5	◊	21		37	%	53	5	69	E	85	U	101	○	117	[
6	†	22	≠	38	&	54	6	70	F	86	V	102	↓	118	]
7	×	23	±	39	¶	55	7	71	G	87	W	103	χ	119	＼
8	≡	24		40	(	56	8	72	H	88	X	104	₪	120	↑
9	Y	25		41	)	57	9	73	I	89	Y	105	λ	121	↶
10	‡	26		42	*	58	□	74	J	90	Z	106	∞	122	†
11	*	27	∫	43	+	59	□	75	K	91	[	107	δ	123	‡
12	⊗	28	∫	44	⁹	60	〈	76	L	92	＼	108	€	124	←
13	—	29	√	45	—	61	—	77	M	93	】	109	η	125	X
14	★	30	~	46	□	62	〉	78	N	94	^	110		126	↑
15	—	31	≈	47	/	63	?	79	O	95	—	111		127	↓