

JAERI-M

6023

PGSLIB / BASIC

グラフィック・プログラミングのための  
基本ソフトウェア

1975年3月

中村 康弘・小沼 吉男

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、複製などのお問い合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしてください。

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.

PGSLIB/BASIC

グラフィック・プログラミングのための基本ソフトウェア

日本原子力研究所東海研究所原子炉工学部

中村康弘・小沼吉男

(1975年2月5日受理)

グラフィック・サブルーチン・ライブラリPGSLIBがFACOM230-60電子計算機とFACOM6233Aグラフィック・ディスプレイに対して開発され、現在までに原研における約30個のグラフィック・プログラムによって利用されている。PGSLIBはグラフィック・ディスプレイを用いた対話型グラフィック処理のプログラミングに便利な道具であり、PGS、簡易GSPおよび汎用GSPの三つのパッケージに大別される。基本的にはFACOM GSPで書かれた約150個のグラフィック・サブルーチン命令からなる。PGSはCALCOMPプロッタ・プログラミングと同じ方法でCRT面への図形表示のプログラミングを可能とし、またCRT面に表示された図形のハード・コピーをプロッタへ出力できる。簡易GSPはライトペンやキーボードによる人間と計算機との対話処理のプログラミングを可能とする。汎用GSPはグラフ表示やライトペン・スミーズイングなどの各種グラフィック機能を有し、おもにPGSと簡易GSPを用いて書かれている。PGS、簡易GSPおよび汎用GSPの各パッケージの中で、グラフィック・プログラミングの上で基本的でかつよく使用されるサブルーチン命令の集まりはPGSLIB/BASICと呼ばれる。本報告ではこのPGSLIB/BASICの命令について、その機能、呼出し形式、使用上の注意およびプログラム例が詳細に記述されている。

PGSLIB/BASIC : A Basic Software for Graphic Programming

Yasuhiro NAKAMURA and Yoshio ONUMA

Division of Reactor Engineering, Tokai, JAERI

( Received February 8, 1975 )

The graphic subroutine library, PGSLIB, has been developed for FACOM 230-60 computer with FACOM 6233A graphic display and utilized in some 30 graphic programs of the JAERI. The library is a convenient programming tool for interactive computer graphics using graphic display. It has about 150 graphic subroutine instructions, written in FACOM GSP and classified into three packages, i.e., PGS, simplified GSP and general GSP. PGS is used to write a program for displaying graphic figures on the CRT screen in the same way as the CALCOMP plotter programming and to produce hardcopy images of the CRT screen on the plotter. Simplified GSP is used to write a program for processing man-machine interaction using lightpen and keyboards. General GSP, written mainly in PGS and simplified GSP, has various graphic functions such as graphing, lightpen-smoothing, etc. The subset of subroutine instructions included in PGSLIB is called PGSLIB/BASIC. It consists of the basic and frequently used instructions for graphic programming in PGS, simplified GSP and general GSP. In this report, the functions, calling sequences, usage notes, and programming examples of the PGSLIB/BASIC instructions are described in detail.

## 目 次

1	はじめに	1
2	PGSLIB/BASICの概要	2
2.1	PGS, 簡易GSPと汎用GSP	2
2.1.1	PGS	2
2.1.2	簡易GSP	2
2.1.3	汎用GSP	2
2.2	PGSLIB/BASIC	2
2.3	プロッタとグラフィック・ディスプレイ	2
2.3.1	装置	3
2.3.2	プログラム	3
2.3.3	座標系	3
2.3.4	文字	4
3	PGSによる図形表示のプログラミング	10
3.1	PGSの概要	10
3.1.1	PGS命令	10
3.1.2	接続命令	10
3.1.3	拡張命令	10
3.2	PGSの機能	11
3.2.1	GPLOTS命令	11
3.2.2	GPLOT命令	13
3.2.3	GSYMBL命令	14
3.2.4	GNUMBR命令	15
3.2.5	GSCALE命令	16
3.2.6	GAXIS命令	16
3.2.7	GLINE命令	17
3.2.8	GFACTR命令	18
3.2.9	GWHERE命令	19
3.2.10	拡張命令	19
4	簡易GSPによる対話処理のプログラミング	22
4.1	簡易GSPの概要	22
4.1.1	割込処理命令	22
4.1.2	トラッキング処理命令	22
4.1.3	データ入出力処理命令	22
4.2	簡易GSPの機能	23
4.2.1	GETIDF命令	23
4.2.2	GLIST命令	24

4.2.3	GJUMP命令	24
4.2.4	GBGTRK命令	24
4.2.5	GRDTRK命令	25
4.2.6	GENTRK命令	25
4.2.7	GREADD命令	26
4.2.8	GWRITE命令	26
5	汎用GSPによるグラフィック・プログラミング	28
5.1	汎用GSPの概要	28
5.1.1	二次元図形表示命令	28
5.1.2	三次元図形表示命令	28
5.1.3	汎用グラフ表示命令	28
5.1.4	テキスト処理命令	28
5.1.5	対話型計算処理命令	28
5.1.6	プロッタ・タイプ・シンボル命令	29
5.1.7	ライトペン・スムービング命令	29
5.2	汎用GSPの機能	29
5.2.1	GPLOT1命令	29
5.2.2	GSMOOT命令	31
6	グラフィック・プログラミングの例	33
6.1	例題	33
6.2	各プログラムの説明	33
6.2.1	プロッタ・プログラム	33
6.2.2	グラフィック・プログラム(その一)	34
6.2.3	グラフィック・プログラム(その二)	34
6.3	各プログラムによる図形出力	35
7	おわりに	40
	参考文献	40

## 1 はじめに

図形入出力装置としてのグラフィック・ディスプレイ<sup>(1),(2)</sup>は、電子計算機からの多量の情報をグラフなどによってCRT (Cathode Ray Tube) 面に高速度で図形表示できるばかりでなく、またライトペン、ファンクション・キーボードおよび文字キーボードなどの使用によって計算機へ情報を直接入力することができる。このためグラフィック・ディスプレイは人間が計算機と対話をしながら、多量の情報を処理し、判断を下してゆく場合に便利な道具であり、近年、計算機による情報処理の多くの分野で利用されつつある。

特に原子力計算の分野においては、対話形式<sup>(3)</sup>による、ガンマ線スペクトルの解析、核断面積データの処理<sup>(4)</sup>、実験データのカーブ・フィッティングおよび原子炉の地震シミュレーション解析<sup>(5),(6)</sup>などのために以前からグラフィック・ディスプレイが利用され、それが非常に有効であることが実証されている。<sup>(7)</sup>

グラフィック・ディスプレイはこのように優れた機能をもったオンライン図形処理装置であるが、そのプログラミングはプロッタの場合に比べて数倍面倒と言はざるを得ない。ちなみに、プログラミングに必要な命令の数をプロッタとグラフィック・ディスプレイで比較してみると、プロッタの場合はCALCOMPの基本サブルーチン・パッケージ<sup>(8)</sup>で9個の命令であるが、グラフィック・ディスプレイの場合はGSP (Graphic Subroutine Package)<sup>(9)</sup>で約40~50個の命令となる。<sup>(2),(10)</sup>

そこで筆者らは、実質的にはプロッタの場合の約2倍程度の命令数でグラフィック・プログラミングを可能とするグラフィック・サブルーチン・ライブラリPGSLIB<sup>(11),(12)</sup> (Plotter-compatible Graphic Subroutine Library)をFACOM 230-60計算機とFACOM6233<sup>(1)</sup> Aグラフィック・ディスプレイに対して開発した。PGSLIBはプロッタ・プログラムのグラフィック・プログラムへの発展を容易に実現できるので、現在までに原子力計算の分野における約30個のグラフィック・プログラムによって利用されてきた。また、PGSLIBは約150個のグラフィック・サブルーチン命令で構成されるが、その内グラフィック・プログラミングの上で基本的かつよく使用される命令の集まりはPGSLIB/BASICと呼ばれる。PGSLIB/BASIC以外の大半の命令は、選択的に使用されるものであり、おもにPGSLIB/BASICの命令を用いて書かれている。

以下、このPGSLIB/BASICについて、まずその概要を述べ、つぎにそれによるグラフィック・プログラミングの方法と具体的なプログラム例を詳しく述べる。

## 2 PGSLIB/BASICの概要

### 2.1 PGS, 簡易GSPと汎用GSP

PGSLIBは機能的にはPGS<sup>(11)</sup>(Plotter-compatible Graphic Subroutine Package), 簡易GSP<sup>(12)</sup>(Simplified GSP), および汎用GSP<sup>(13)</sup>(General GSP)と呼ばれる三つのグラフィック・サブルーチン・パッケージに分けられる。

#### 2.1.1 PGS

このパッケージはCALCOMPプロッタ・プログラミングと同じ方法でCRT面への図形表示のプログラミングを可能とする。またCRT面上の図形を消去したり, CRT面上の図形のハード・コピーをプロッタ上に出力できる。さらに, CRT面上の図形に対してライトペンと文字キーボードの使用による, 日付け, 使用者名, タイトル, 説明文などのコメント入力が行なえる。

#### 2.1.2 簡易GSP

このパッケージはPGSに合わせてFACOM GSP<sup>(2)</sup>を単純化したものである。このため図形の表示や消去の処理の他に, ライトペン, ファンクション・キーボードおよび文字キーボードなどの使用による人間と計算機との対話処理が簡単にプログラミングできる。

#### 2.1.3 汎用GSP

このパッケージは, グラフ表示<sup>(13)</sup>, ライトペン・スムージング<sup>(14)</sup>, 二次元図形表示<sup>(15)</sup>, 三次元図形表示<sup>(16)</sup>, テキスト処理<sup>(17)</sup>, および対話型計算処理<sup>(18)</sup>などに関する, おもにPGSと簡易GSPの命令によって書かれた, 各種のグラフィック処理命令からなる。したがってこのパッケージの使用によって, ユーザはPGSや簡易GSPだけによるよりもっと簡単にグラフィック・プログラミングができる。

## 2.2 PGSLIB/BASIC

上記のPGS, 簡易GSPおよび汎用GSPの各パッケージの中でグラフィック・プログラミングの面からみて基本的かつよく使用される命令の集まりがPGSLIB/BASICである。図1に簡単に示すように, PGSでは図形の表示や消去, ハード・コピー出力およびコメント入力の機能に関する命令を中心にその約9割の命令が, また簡易GSPではライトペン, ファンクション・キーボードおよび文字キーボードなどの処理に関する命令を中心にその約4割の命令が, さらに汎用GSPではグラフ表示とライトペン・スムージングに関する命令のみが, それぞれPGSLIB/BASICに属する。表1~表3はPGSLIB/BASICに属する命令をパッケージごとに示した一覧表である。

なお, PGS, 簡易GSPおよび汎用GSPの命令でPGSLIB/BASICに属さない多くの命令については別途JAERI-Mレポート<sup>(16,17,18)</sup>で述べる。

### 2.3 プロッタとグラフィック・ディスプレイ

PGSLIB/BASIC(またはPGSLIB)ではPGSによって, 元々プロッタ用に作成された図形出力プログラムはグラフィック・ディスプレイ用の図形表示プログラムとして利用で



きる。さらに簡易GSPによって、ライトペン、ファンクション・キーボードおよび文字キーボードなどによる対話処理のプログラムを作り、これを先の図形表示プログラムに追加することにより、本格的な対話型のグラフィック・プログラムにすることができる。このようにPGSLIB/BASICはプロッタ・プログラムのグラフィック・プログラムへの発展を容易に実現できる。

そこで、ここではPGSLIB/BASICによってプロッタ・プログラムをグラフィック・プログラムとして利用することを前提として、プロッタとグラフィック・ディスプレイを対比させ、それらの類似点や相違点に注意しながらそれぞれの特徴について述べる。なお、ここでプロッタとグラフィック・ディスプレイについては、それぞれ現在日本原子力研究所計算センター<sup>(19)</sup>に設置されているCALCOMP900とFACOM6233Aを比較の対象とする。

### 2.3.1 装置

プロッタは出力のみのオフライン図形処理装置である(図2)。これに対してグラフィック・ディスプレイは入出力とも可能なオンライン図形処理装置である(図3)。

プロッタではプロッタ・シート上をペンが動くことによって図形が描かれる。一方グラフィック・ディスプレイではCRT面上をビームが動くことによって図形が描かれ、またライトペン、ファンクション・キーボード、文字キーボードおよびASW(Alteration Switch)などによって計算機へ情報が入力される。さらにCRT面に表示された図形のハード・コピーを得るために、プログラム・コントロールによるプロッタ・ハード・コピーとハードウェアによるハード・コピー装置の両方が利用できる。

### 2.3.2 プログラム

プロッタではその基本サブルーチン・パッケージを用いてプロッタ用のプログラムが作られる。プロッタ・プログラムが計算機によって実行されると、図形情報はプロッタ・コマンドに変換され、そしてコマンドはコア上のバッファに蓄積される。バッファが一杯になるとコマンドは自動的に磁気テープに出力される。このテープをプロッタにかけると図形が描かれる(図2)。

一方グラフィック・ディスプレイではグラフィック・プログラミング用の基本ソフトウェアとしてのPGSLIB/BASICを用いてグラフィック・ディスプレイ用のプログラムが作られる。グラフィック・プログラムが計算機で実行されると、図形情報はグラフィック・ディスプレイの命令語の並びとしてのディスプレイ語列に変換され、そしてディスプレイ語列はコア上の出力領域GDOA(Graphic Data Output Area)に蓄積される。GDOAの内容は表示命令によってグラフィック・ディスプレイのバッファ・メモリに転送され、CRT面上に図形が表示される。ここでライトペン、ファンクション・キーボードおよび文字キーボードなどが使用されると計算機に対して割込が起り、その結果グラフィック・プログラムはそれらによって与えられた入力情報を受取ることができる(図3)。

### 2.3.3 座標系

プロッタでは図4に示すように、プロッタ・シート上で縦は約28cm、横は一応無制限の範囲で図形が描かれる。しかし、プロッタ用のプログラムをグラフィック・ディスプレイに利用する場合は、後述されるように横方向は40cmの範囲内に図形が描かれるのが望ましい。座標

の原点は最初プロッタ・シート上左下端にセットされるが、ユーザの責任で以後自由に原点の変更ができる。座標の値はmm単位で表わされる。

一方グラフィック・ディスプレイでは図5に示すように、CRT面は直径50cmの円形であり、ピタゴラスの定理によって明らかなように縦30cm、横40cmの矩形がCRT面に含まれる。そこでプロッタとの関連で縦28cm、横40cmの範囲内に図形が描かれるのが望ましい。原点はプロッタと同じく最初CRT面の左下端にセットされるが、ユーザの責任で以後自由に原点を変更できる。座標の値もプロッタと同じくmm単位で表わされる。

#### 2.3.4 文字

プロッタでは任意の大きさと任意の角度からなる文字が描かれる。

一方グラフィック・ディスプレイでは表4に示すように、ハードウェアのシンボル・ジェネレータによって生成される4種類の大きさの文字しか描かれず、しかも角度は0度のみである。文字の大きさはVSM, SML, LAG, VLGの指定、またはmm単位での大きさの指定が可能である。mm単位で指定された場合はその大きさによって上記4種類のいずれかにセットされる。すなわち指定された文字の高さhによって

$0\text{mm} \leq h < 4.2\text{mm}$ ならば VSM,

$4.2\text{mm} \leq h < 6.5\text{mm}$ ならば SML,

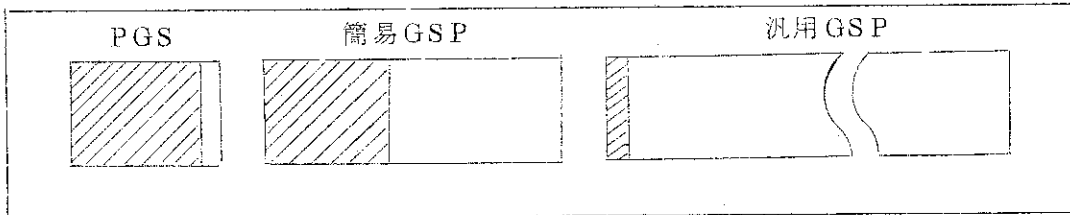
$6.5\text{mm} \leq h < 9.3\text{mm}$ ならば LAG,

$9.3\text{mm} \leq h$  ならば VLG

が使用される。

なお、グラフィック・ディスプレイでプロッタのような任意の大きさと任意の角度からなる文字を描く場合は文字をソフトウェアで生成しなければならない。

PGSLIB




 は PGSLIB/BASIC に属する命令を示す

図1 PGS, 簡易GSP, 汎用GSP, PGSLIB/BASIC と PGSLIB の相互関係

表1 PGSLIB/BASIC に属する PGS の命令

命 令	機 能
GPLOTS GPLOT GSYMBL GNUMBR GSCALE GAXIS GLINE GFACTR GWHERE	グラフィック・ディスプレイをオープンする 直線を描く。またはグラフィック・ディスプレイをクローズする 文字またはプロットイング・ポイント用記号を描く。 数字を描く データをスケールリングする 座標軸を描く データのグラフを描く 図形の拡大・縮小に関する尺度因子を与える ビームの位置および尺度因子を得る。
拡張命令 (GPLOT)	図形の名前付け, 図形の表示や消去, およびコメント入力やハード・コピー出力を行なう
接続命令	上記の各 PGS 命令に対応して PLOTS, PLOT, SYMBOL, NUMBER, SCALE, AXIS, LINE, FACTOR, WHERE の命令があり, こ れらの命令はそれぞれ対応する PGS 命令を呼出す働きを する

表2 PGSLIB/BASICに属する簡易GSPの命令

命 令	機 能
GETIDF	ライトペン, ファンクション・キーボードおよびトラッキング・シンボルからの割込に関する識別記号を発生した順に取り出す
GLIST	割込に関する識別記号のリストをつくる
GJUMP	割込を識別し, GLISTで指定された識別記号に対応する文番号へプログラムを分岐する
GBGTRK	CRT面上にトラッキング・シンボルを表示する
GRDTRK	トラッキング・シンボルの座標を取り出す
GENTRK	CRT面からトラッキング・シンボルを消去する。
GREADD	文字キーボードからデータをFORMAT付きで入力する
GWRITE	データをFORMAT付きでCRT面へ表示する

表3 PGSLIB/BASICに属する汎用GSPの命令

命 令	機 能
GPLOT1	与えられたデータを自動スケールし, 線型または対数の座標軸と共にグラフを描く。またグラフの重ね合せやそれらの区別をつける
GSMOOT	与えられたデータをライトペンによってスムーzingする

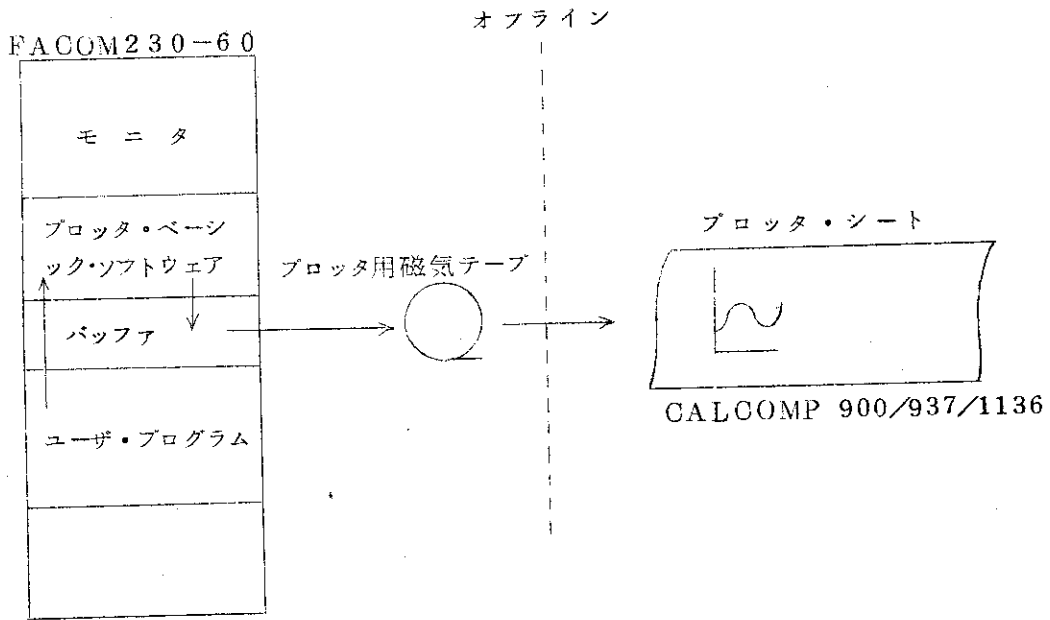


図2 プロッタ

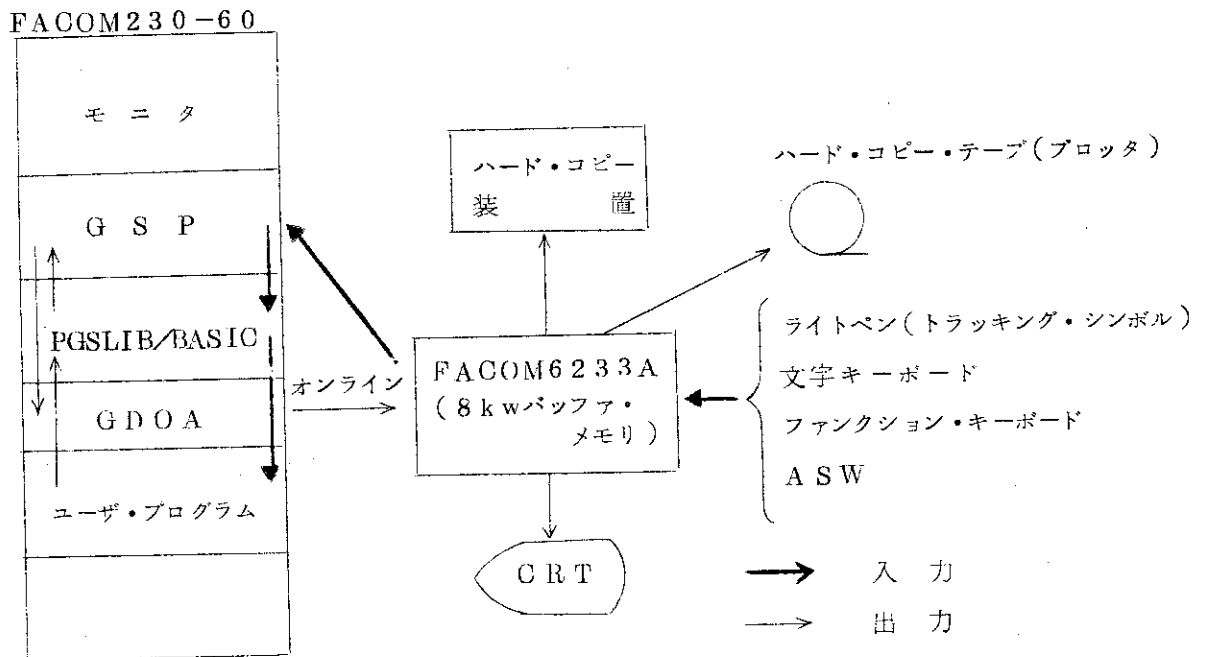


図3 グラフィック・ディスプレイ

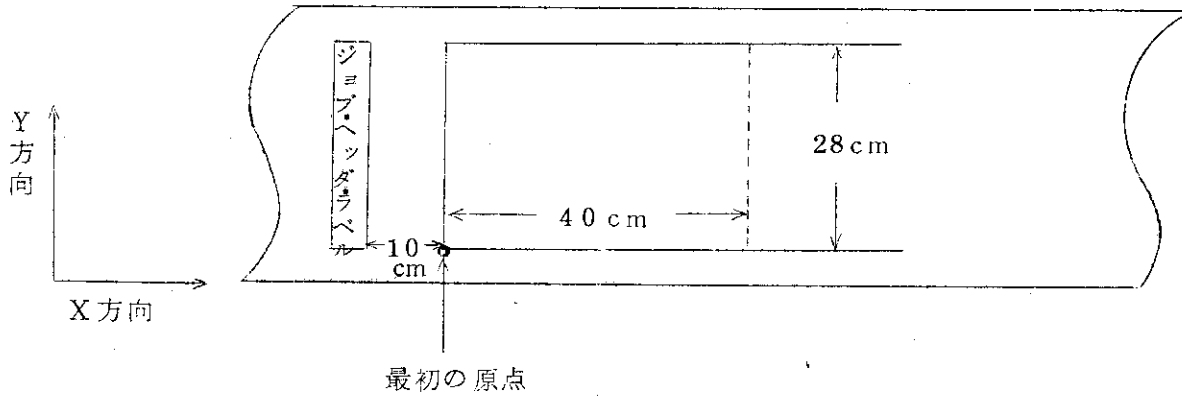


図4 プロッタの座標系

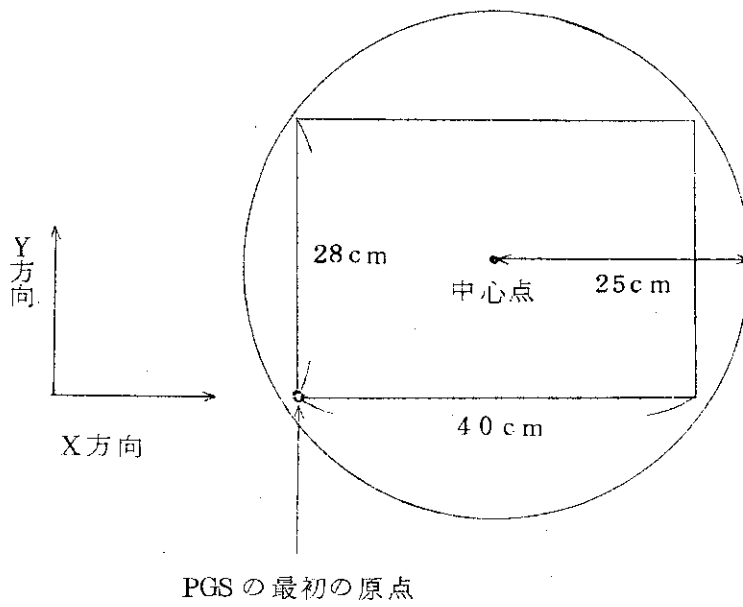
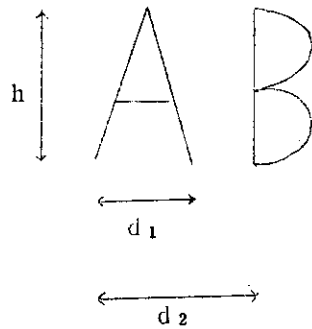


図5 グラフィック・ディスプレイの座標系

表4 文字の大きさ

大きさの種類	$d_1$	$d_2$	$h$
VSM(very small)	2.1	2.9	2.3
SML(small)	3.1	4.4	4.2
LAG(large)	4.9	6.8	6.5
VLG(very large)	7.0	9.8	9.3

(単位はmm)



### 3 PGSによる図形表示のプログラミング

#### 3.1 PGSの概要

グラフィック・ディスプレイのプログラムはCRT面への図形表示部分、~~と~~ライトペン、ファンクション・キーボードおよび文字キーボードなどによる対話処理部分に分けられる。図形表示部分と対話処理部分は両方共GSPによってプログラミングできるが、それを実際に行なうことは相当に面倒である。

これに対してPGSLIB/BASICによる方法では、すでに2.3で述べたように、図形表示部分はPGSによって、また対話処理部分は簡易GSPによって、それぞれGSPによるよりも簡単にプログラミングできる。そこで本章ではPGSによる図形表示のプログラミングについて述べ、次の章で簡易GSPによる対話処理のプログラミングについて述べる。

さて、PGSLIB/BASICに属するPGSの命令はすでに表1に示したが、それらの命令は

##### (1) PGS 命令

GPLOTS, GPLOT, GSYMBL, GNUMBR, GSCALE, GAXIS, GLINE, GFACTR, GWHERE

##### (2) 接続命令

PLOTS, PLOT, SYMBOL, NUMBER, SCALE, AXIS, LINE, FACTOR, WHERE

##### (3) 拡張命令

GPLOT(xcoor, ycoor, ibeam)で、ibeam=333, 444, 666, 777, 888のときの三つに大別される。

#### 3.1.1 PGS 命令

これはプロッタの基本サブルーチン

PLOTS, PLOT, SYMBOL, NUMBER, SCALE, AXIS, LINE, FACTOR, WHEREと同じ機能をグラフィック・ディスプレイで実現する。したがって、つぎに述べる接続命令と一緒に、プロッタ用の図形出力プログラムをグラフィック・ディスプレイ用の図形表示プログラムとして実行可能とする。

#### 3.1.2 接続命令

これはプロッタの基本サブルーチンと同じ名前をもち、たとえばPLOTS命令はGPLOTS命令のみを呼び出すように、対応するPGS命令のみを呼出す働きをする。接続命令はこれだけの機能しかないので、これ以上述べることはない。

#### 3.1.3 拡張命令

これは、プロッタにはないグラフィック・ディスプレイ特有の機能を実現させるもので、GPLOT(xcoor, ycoor, ibeam)のibeamの値によって区別されるつぎのような機能がある。

ibeam = 333 . . . GDOAの初期化(消去), またはCRT面に表示される図形単位に対する名前付け  
 = 444 . . . GDOAのディスプレイ・バッファへの転送による図形の表示, およびGDOAの初期化



- = 6 6 6 . . . . CRT面に表示されている全図形または図形単位の消去
- = 7 7 7 . . . . GDOAのディスプレイ・バッファへの転送による図形の表示、  
文字キーボードからのコメント入力、およびGDOAの初期化
- = 8 8 8 . . . . CRT面に表示されている図形のプロッタへのハード・コピー  
出力

ここで、図形単位とはCRT面に表示される図形のある一まとまりを言う。図形単位に名前がつけられると（図形単位、図形単位の名前を、別にそれぞれエレメント、エレメント名とも呼ぶ）、その図形がCRT面に表示されたときライトペンによって認識可能となり、図形単位での処理が可能となる。特に、ある図形単位についてそれがライトペンによって認識されたとき、ある定まった働きをするものをライトペン・ボタン（または簡単にライト・ボタン）と呼ぶ。

つぎに、拡張命令による文字キーボードからのコメント入力やプロッタへのハード・コピー出力は、それぞれコメント入力用、ハード・コピー用のライトペン・ボタンを使用して行なわれる。コメント入力用ボタンは

KEY, END, VLG, LAG, SML, VSM

の6個のボタンからなり、ハード・コピー用ボタンは

HARDCOPY, NOCOPY

の2個のボタンからなる。これらのボタンを総称してPGSボタンと呼ぶ。PGSボタンはグラフィック・ディスプレイのオープン直後に図6に示すようにCRT面上に表示される。図6でCRT面中央のメッセージは特にPGSメッセージと呼ばれ、ユーザの図形が表示されるまでウイंकしながら表示し続ける。PGSボタンはユーザのプログラムが実行されている間中表示されている。PGSボタンの内ハード・コピー用ボタンと同じ機能がファンクション・キーの8番、16番によっても行なわれる。すなわち、8番キーがHARDCOPYボタンに対応し、16番キーがNOCOPYボタンに対応する。PGSボタンの操作は図7のフローチャートのように行なわれるが、さらに詳細についてはグラフィック・ディスプレイの操作マニュアル<sup>(8), (20)</sup>を参照下さい。

### 3.2 PGSの機能

#### 3.2.1 GPLOTS命令 (Open Graphic Display)

##### (1) 機能

グラフィック・ディスプレイをオープンし、使用可能状態にする。

##### (2) 呼出し形式

```
CALL GPLOTS (gdoa, nwords)
```

##### (3) 引数の説明

gdoa : ディスプレイ語列の出力領域の配列を実数型の配列名で指定する。

nwords : 配列gdoaの大きさを整数型の定数または変数で指定する。

##### (4) 注意事項

- 1) この命令はPGS命令の中で最初に呼出されねばならない。
- 2) この命令の他にDIMENSION文によって配列gdoaの宣言をしなければならぬ。

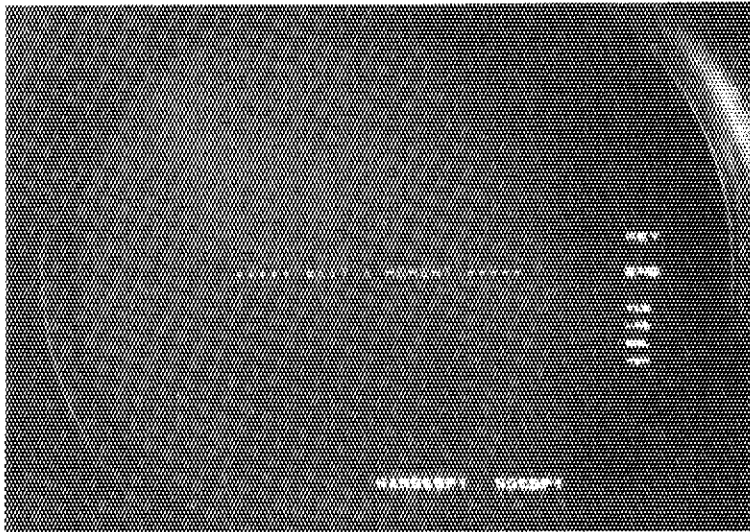


図6 PGS ボタンとメッセージ

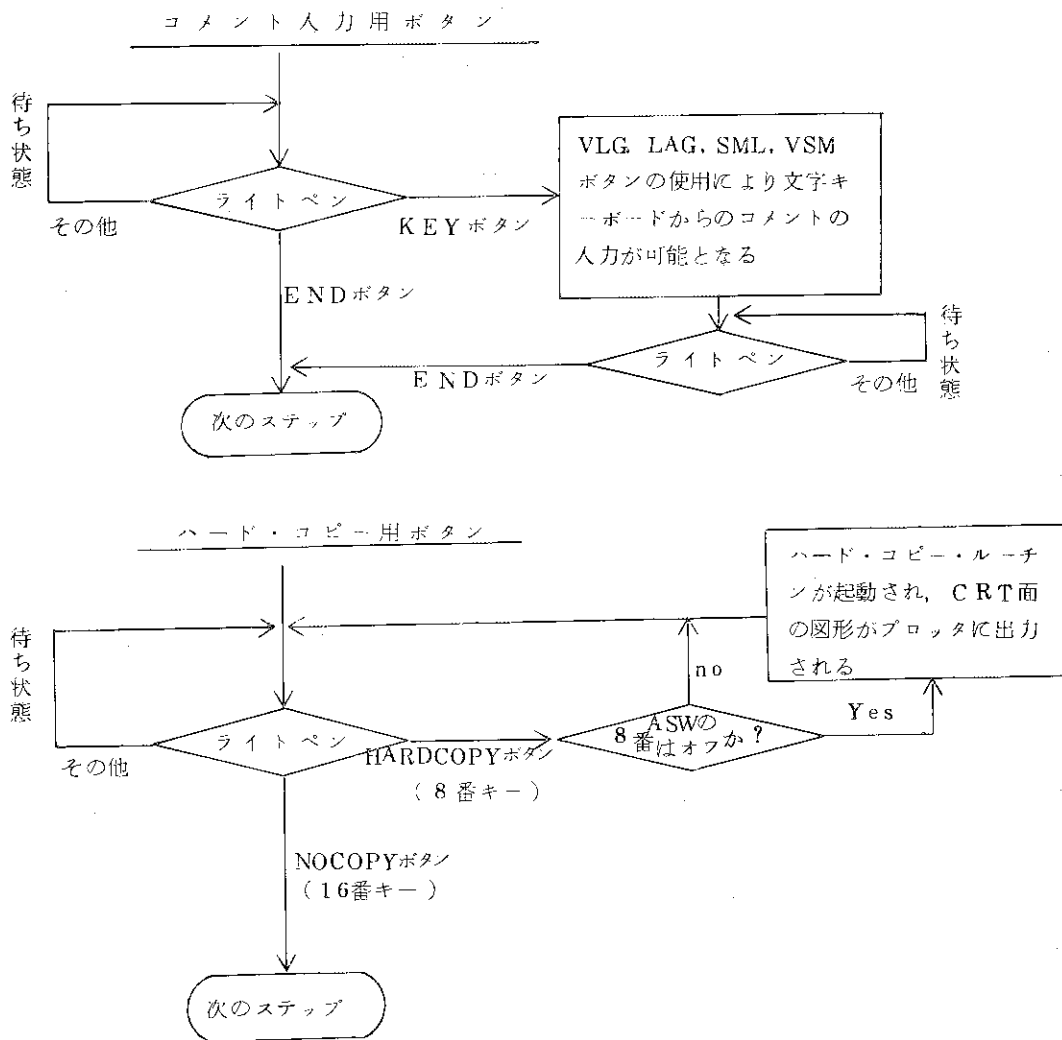


図7 PGS ボタンの操作

3) nwords は 1 0 2 4 以下でなければならない。

(5) 参考事項

1) ユーザはつぎの事項を確認することで、この命令によってグラフィック・ディスプレイがオープンされたことを知ることができる。

(a) 計算機のコンソール・タイプライタ上に

\*\*\* GRAPHIC START \*\*\*

のメッセージが出力される。

(b) CRT面にPGSメッセージとPGSボタンが表示される。

(c) ファンクション・キー8番, 16番にランプがつく。

2) この命令によってグラフィック・コントロール・テーブルGCT(1000語)が生成される。GCTにはライトペンによって認識可能な図形単位名が自動的に格納される。GCTはGSPによって管理されるので、通常ユーザはタッチする必要がない。

3) この命令によってビームと座標の原点はCRT面の左下端にセットされる。すなわち、CRT面上の中心点からX軸方向-200.0mm, Y軸方向-140.0mm離れた点が最初の原点(0.0, 0.0)となる。

4) グラフィック・ディスプレイ装置のASWの8番がオフのときは、この命令によってプロッタ・ハード・コピーのためにプロッタ・テープのオープンも行なわれる。

### 3.2.2 GPLOT命令 (Plot Vector and Close Graphic Display)

(1) 機能

CRT面上でのビームによるベクトル表示やビームの位置決めのためのディスプレイ語列を作成し、これをGDOAに格納する。またはグラフィック・ディスプレイをクローズし、使用不能状態にする。

(2) 呼出し形式

```
CALL GPLOT(xcoor, ycoor, ibeam)
```

(3) 引数の説明

xcoor, ycoor : ビーム移動の終了点の座標を実数型の定数または変数で指定する。

ibeam : ビームの状態について次のいずれか一つを整数型の定数または変数で指定する。

ibeam=2 現在のビーム位置から座標(xcoor, ycoor)の位置までのベクトルを表示する(ビーム・オン)。

ibeam=3 現在のビーム位置から座標(xcoor, ycoor)の位置にビームを移動する(ビーム・オフ)。

ibeam=-2 ibeam=2と同じ機能を実行後、ビームの終了点を新しい原点とする。

ibeam=-3 ibeam=3と同じ機能を実行後、ビームの終了点を新しい原点とする。

ibeam=9.99 グラフィック・ディスプレイをクローズし、使用不能状

態にする。

(4) 注意事項

1) 座標(xcoor, ycoor)の値はmm単位で表わされなければならない。

(5) 参考事項

1) ibeam=999のときは、CRT面上の全図形が消去され、またファンクション・キーの8番、16番のランプも消える。さらにASWの8番がオフならばプロッタ・テープもクローズされる。

2) ibeam=±2, ±3, 999以外の機能は、拡張命令のところで後述される。

### 3.2.3 GSYMBL命令(Plot Symbols)

(1) 機能

文字列(英数字, 特殊記号の一文字またはそれらの並び), またはプロットング・ポイント用記号(Centered Symbol)を表示するためのディスプレイ語列を作成し, これをGDOAに格納する。

(2) 呼出し形式

(a) CALL GSYMBL(xcoor, ycoor, height, string, angle, nchars)

(b) CALL GSYMBL(xcoor, ycoor, height, icode, angle, nbeam)

(3) 引数の説明

(a) 文字列(英数字, 特殊記号)の場合

xcoor, ycoor : 表示する文字列の出発点(左下隅)の座標  
 height : 文字列の高さを実数型の定数または変数で指定する。  
 string : 文字列を文字定数または変数(配列も可)で指定する。  
 angle : 文字列のX座標軸となす角度を実数型の定数または変数で指定する。  
 nchars : 文字列を構成する文字の個数を整数型の定数または変数で指定する。

(b) プロットング・ポイント用記号(Centered Symbol)の場合

xcoor, ycoor : 表示するプロットング・ポイント用記号(ここでは以後簡単に記号とよぶ)の中心座標  
 height : 記号の高さ  
 icode : 記号のコード番号を整数型の定数または変数で指定する。  
 angle : 記号のX座標軸となす角度  
 nbeam : ビームの状態について次のいずれか一つを整数型の定数または変数で指定する。

nbeam=-1 ビーム・オフで表示する位置(xcoor, ycoor)へ行き, 記号を表示する。

nbeam=-2 ビーム・オンで表示する位置(xcoor, ycoor)へ行き, 記号を表示する。

(4) 注意事項

1) 文字列や記号の高さheightはmm単位で表わされなければならない。

2) 角度 angle は X 座標軸に対する左回りの度数を表わすが、現在は 0 度と 90 度の指定のみ有効である (記号の場合は 0 度のみ有効)。

3) コード番号 icode と記号の対応は次の通りである ( $0 \leq \text{icode} \leq 15$ )。

icode	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
記号	□	○	△	+	×	◇	↑	↘	Z	Y	✕	*	⊗		☆	—

図8 コード番号とプロット・ポイント用記号の対応

(5) 参考事項

- 1) 文字定数は nH か ▼ ▼ のいずれかで指定可能である。
- 2) 文字についてはハードウェア的に角度 0 のものしかないので、0 度と 90 度の場合はずきのように表示される。一文字のみの場合は 0 度、90 度共同じとなる。長さが 2 以上の文字列の場合、0 度では文字が X 座標軸に平行に横に並ぶのに対して、90 度では Y 座標軸に平行に並ぶ。
- 3) 後述される簡易 GSP の GWRITE 命令は GSYMBL 命令 (文字列の場合) よりも広い機能をもっている。

3.2.4 GNUMBR 命令 (Plot Number)

(1) 機能

実数 (浮動小数点) を 10 進数 (固定小数点) 形式の文字列で表示するためのディスプレイ語列を作成し、これを GDOA に格納する。

(2) 呼出し形式

```
CALL GNUMBR(xcoor, ycoor, height, fltpn, angle, ndecs)
```

(3) 引数の説明

- xcoor, ycoor : 表示される 10 進数文字列の出発点 (左下隅) の座標
- height : 10 進数文字列の高さ
- fltpn : 表示する実数を実数型の定数または変数で指定する。
- angle : 10 進数文字列の X 座標軸となす角度
- ndecs : 小数点以下に表示される桁数について次のいずれか一つを整数型の定数または変数で指定する。
  - ndecs > 0 整数部分と小数点と小数点以下 ndecs 桁が表示される。
  - ndecs = 0 整数部分と小数点のみ表示される。
  - ndecs = -1 整数部分のみ表示される。
  - ndecs < -1 整数部分から |ndecs| - 1 だけ頭が切られて表示される。

(4) 注意事項

- 1) 角度 angle は 0 度と 90 度のみ有効である。

(5) 参考事項

1) 角度 angle が 90 度の場合の文字表示は GSYMBL 命令の参考事項 2) で述べた方法で行なわれる。

2) 後述される簡易 GSP の GWRITE 命令は GNUMBR 命令よりも広い機能をもっている。

### 3.2.5 GSCALE 命令 (Scale Data)

#### (1) 機能

データのプロットに先立ち、データのグラフが CRT 面内におさまるようにスケーリングし、データの最小値と増分を求める。

#### (2) 呼出し形式

```
CALL GSCALE (data, width, ndata, incrm, div)
```

#### (3) 引数の説明

data : スケーリングするデータが格納されている配列を実数型の配列名で指定する。

width : データの展開巾の長さを実数型の定数または変数で指定する。

ndata : 配列 data に格納されているデータの個数を整数型の定数または変数で指定する。

incrm : 配列 data に格納されているデータを取り出す間隔を整数型の定数または変数で指定する。

incrm=1 1 連続的にデータを取り出す。

incrm=k k k-1 個おきにデータを取り出す。

div : つねに 10.0 を指定する。

#### (4) 注意事項

1) データの展開巾 width は mm 単位で表わされなければならない。

2) 配列 data の大きさはデータ数 ndata  $\times$  incrm より incrm + 1 多くなければならない。

3) 最小値と増分に関する計算結果は、それぞれ data (ndata  $\times$  incrm + 1), data (ndata  $\times$  incrm + incrm + 1) に格納される。

4) この命令が使用される時は、後述される GAXIS 命令と GLINE 命令に先んじて呼出されなければならない。

#### (5) 参考事項

1) 最小値と増分は後述される GAXIS 命令と GLINE 命令で利用することができる。

2) 通常は incrm=1 として使用される。

3) この命令を使用せず、最小値と増分を自分で計算し、それぞれ data (ndata  $\times$  incrm + 1), data (ndata  $\times$  incrm + incrm + 1) に格納してもよい。

### 3.2.6 GAXIS 命令 (Plot Axis)

#### (1) 機能

線型の座標軸を表示するためのディスプレイ語列を作成し、これを GDOA に格納する。

#### (2) 呼出し形式

CALL GAXIS(xcoor, ycoor, title, nchars, width, angle, start, delta, div)

## (3) 引数の説明

- xcoor, ycoor : 表示する座標軸の出発点の座標
- title : 座標軸のタイトルを文字定数または変数(配列も可)で指定する。
- nchars : タイトルを構成する文字の個数とタイトルの表示の仕方について整数型の定数または変数で指定する。  
           nchars > 0 座標軸に対して左回りの側にタイトルを表示する。  
           nchars < 0 座標軸に対して右回りの側にタイトルを表示する。
- width : 座標軸の長さ
- angle : 座標軸の傾きを示すX座標軸となす角度
- start : 座標軸の目盛の出発値(最小値)を実数型の定数または変数で指定する。
- delta : 座標軸の目盛について、目盛間の値の単位長さ当りの増分を実数型の定数または変数で指定する。
- div : つねに10.0を指定する。

## (4) 注意事項

- 1) 座標軸の傾きを示す角度 angle は0度(X座標軸)と90度(Y座標軸)のみ有効である。

## (5) 参考事項

- 1) 座標軸の目盛の出発値(最小値) start や増分 delta には、GSCALE 命令で算出された最小値や増分が利用できる(自分で計算したものでよい)。
- 2) 座標軸のタイトル title は、座標軸に平行でかつ座標軸の中央部分に表示される。

## 3.2.7 GLINE 命令 (Plot Line)

## (1) 機能

2つの配列に格納されているデータを順に結んでプロットするディスプレイ語列を作成し、これをGDOAに格納する。

## (2) 呼出し形式

CALL GLINE(xdata, ydata, ndata, incrmt, lintyp, icode)

## (3) 引数の説明

- xdata, ydata : X方向およびY方向のデータが格納されている配列を実数型の配列名で指定する。
- ndata : 配列 xdata と ydata に格納されているデータ (xdata(i), ydata(i)) の個数
- incrmt : 配列 xdata と ydata に格納されているデータ (xdata(i), ydata(i)) を取り出す間隔
- lintyp : データをプロットする方法について次のいずれか一つを整数型の定

数または変数で指定する。

$lityp=0$  データの各点をベクトルで結ぶが、プロット・ポイント用記号は表示しない。

$lityp=l$  データの各点をベクトルで結び、 $(l-1)$ 点おきに  $icode$  で示すプロット・ポイント用記号を表示する。

$lityp=-l$  データの  $(l-1)$ 点おきに  $icode$  で示すプロット・ポイント用記号を表示するが、データの各点はベクトルで結ばない。

$icode$  : 表示するプロット・ポイント用記号のコード番号

#### (4) 注意事項

- 1) この命令の呼出し前に、 $xdata$ の最小値と増分はそれぞれ  $xdata(ndata \times incrmt + 1)$ と  $xdata(ndata \times incrmt + incrmt + 1)$ に、 $ydata$ の最小値と増分はそれぞれ  $ydata(ndata \times incrmt + 1)$ と  $ydata(ndata \times incrmt + incrmt + 1)$ に格納されていなければならない。

#### (5) 参考事項

- 1) 通常は  $incrmt=1$ として使用される。
- 2) この命令でプロットされる点の座標  $(x, y)$ は次のように計算される。

$$x = (xdata(i) - xstart) / xdelta$$

$$y = (ydata(i) - ystart) / ydelta$$

ここで、 $xstart$ と  $xdelta$ 、 $ystart$ と  $ydelta$ は、それぞれ  $GSCALE$ 命令またはユーザによって与えられた  $xdata$ の最小値と増分、 $ydata$ の最小値と増分である。

### 3.2.8 GFACTR命令 (Set Scale Factor)

#### (1) 機能

図形の拡大・縮小に関する尺度因子を与える。

#### (2) 呼出し形式

```
CALL GFACTR(factor)
```

#### (3) 引数の説明

$factor$  : 尺度因子を実数型の定数または変数で指定する。

#### (4) 注意事項

- 1) 一度この命令を呼出すと、次の変更があるまで有効となる。
- 2) この命令を使用しない場合は  $factor=1.0$ と見なされる。

#### (5) 参考事項

- 1) この命令が呼出されてから次に再び呼出されるまでの間に表示される図形は、この命令が呼出されなかった場合の図形の  $factor$ 倍でCRT面上に表示される。このとき  $factor > 1.0$ ならば拡大を意味し、 $0 < factor < 1.0$ ならば縮小を意味する。また  $factor=1.0$ ならばもとの図形と同じ大きさである。



## 3.2.9 GWHERE 命令 (Where is the beam?)

## (1) 機能

ビームの現在位置の座標と尺度因子を得る。

## (2) 呼出し形式

```
CALL GWHERE(X, Y, FACTOR)
```

## (3) 引数の説明

X, Y : ビームの現在位置の座標を格納する変数を実数型で指定する。  
FACTOR : 現在の尺度因子を格納する変数を実数型で指定する。

## (4) 注意事項

1) X, Y, FACTORは定数でなく変数でなければならない。

## 3.2.10 拡張命令 (Control Graphic Display)

## (1) 機能

GDOAの初期化または図形単位の名前付け, 図形の表示とGDOAの初期化, 全図形または図形単位の消去, 図形の表示とコメント入力とGDOAの初期化, および図形のプロッタへのハード・コピー出力を行なう。

## (2) 呼出し形式

```
CALL GPLOT(xcoor, ycoor, ibeam)
```

## (3) 引数の説明

xcoor, ycoor : (0.0, 0.0)または図形単位名(name1, name2)を, 定数または変数で指定する(name1, name2は整数型とする)。

ibeam : 拡張命令の機能について次のいずれか一つを整数型の定数または変数で指定する。

ibeam=333 (xcoor, ycoor)=(0.0, 0.0)のときはディスプレイ語列の出力領域GDOAを初期化する(消去する)。また(xcoor, ycoor)=(name1, name2)のときはこの命令以後に生成される図形のディスプレイ語列に対して図形単位名(name1, name2)をつける。

ibeam=444 GDOAをディスプレイ・バッファへ転送し, CRT面に図形を表示する。転送後にGDOAを初期化する。

ibeam=666 (xcoor, ycoor)=(0.0, 0.0)のときはCRT面に表示されている全図形を消去し, つぎにPGSメッセージとPGSボタンを再表示し, さらにビームと座標の原点をCRT面の左下端にリセットする。また(xcoor, ycoor)=(name1, name2)のとき

は CRT 面に表示されている図形について (name 1, name 2) の名前で指定される図形単位のみを消去する。

ibeam=777 GDOA をディスプレイ・バッファへ転送し, CRT 面に図形を表示する。つぎにプログラムは待ち状態に入り, コメント入力用ボタンを受付ける。したがってここで文字キーボードからのコメント入力が可能となる。コメント入力が全部完了するとプログラムは待ち状態を脱し, つぎに GDOA を初期化する。

ibeam=888 プログラムは直ちに待ち状態に入り, ハード・コピー用ボタンを受付ける。したがってここで CRT 面上の図形をプロッタへハード・コピー出力できる。ハード・コピーが全部完了するとプログラムは待ち状態を脱し, つぎへ進む。

#### (4) 注意事項

- 1) (name 1, name 2) が図形単位名を表わすときは, name 1 ≠ 0 でなければならない。
- 2) ibeam=444, 777, 888 のときは, (xcoor, ycoor) はつねに (0.0, 0.0) と指定しなければならない。
- 3) プロッタへのハード・コピー出力のとき, ASW 8 番がオンであるとハード・コピー出力は行なわれぬ。したがってハード・コピーを行なうときは, まず GPLOTS 命令によるグラフィック・ディスプレイのオープン時に ASW の 8 番をオフとしてプロッタ・テープをオープンしておき, つぎにハード・コピー用ボタンをライトペンで指すときも ASW の 8 番がオフとなっていなければならない。
- 4) コメント入力は END ボタンによって, またハード・コピーは NOCOPY ボタンによって完了する。したがって END ボタンまたは NOCOPY ボタンが要求されているときにライトペンで指されないとプログラムはいつまでも待ち状態となるので注意しなければならない。

#### (5) 参考事項

- 1) ibeam=444 と ibeam=777 のちがいは, 777 では図形表示後コメント入力のためにプログラムが待ち状態となるのに対して, 444 では待ち状態に入らない。
- 2) 図形は名前がつけられるとライトペンによって認識可能となり, 図形単位での処理が可能となる。
- 3) ハード・コピー出力はファンクション・キーの 8 番, 16 番の制御によっても行なわれる。すなわち, 8 番が HARDCOPY ボタンに対応し, 16 番が NOCOPY ボタンに対応する。そして, ライトペン・ボタンによるときは PGS ボタンを除いた全図形のハード・コピーが出力されるのに対して, ファンクション・キーによるときは PGS ボタンも含めた全図形のハード・コピーが出力される。
- 4) ハード・コピー中, ライトペンによるときは PGS ボタンが, またファンクション・キ

一によるときは8番, 16番キーのランプが消去されているが, ハード・コピーが終ると PGS ボタンが表示されるか, または8番, 16番キーのランプがつく。

## 4 簡易GSPによる対話処理のプログラミング

### 4.1 簡易GSPの概要

PGSによってプロッタ用の図形出力プログラムはグラフィック・ディスプレイ用の図形表示プログラムとして実行可能となった。しかし、このグラフィック・プログラムだけではグラフィック・ディスプレイの機能を十分生かしたものとは言い難い。グラフィック・ディスプレイの真の特長は、CRT面に表示された図形を基にライトペン、ファンクション・キーボードおよび文字キーボードなどによる計算機と人間との対話処理を可能にするところにある。このような対話処理に対するプログラミングは、これから述べる簡易GSPによってGSPによるよりも簡単に行なうことができる。しかもPGSによる図形表示プログラムに簡易GSPによる対話処理のプログラムを追加する形で可能である。

さてPGSLIB/BASICに属する簡易GSPの命令はすでに表2で示したが、それらの命令は

#### (1) 割込処理命令

GETIDF, GLIST, GJUMP

#### (2) トラッキング処理命令

GBGTRK, GRDTRK, GENTRK

#### (3) データ入出力処理命令

GREADD, GWRITE

の三つに大別される。

#### 4.1.1 割込処理命令

これはライトペンやファンクション・キーボードなどによって計算機に指示を与える場合に使用される。CRT面に表示された図形(名前付)をライトペンで指すと、計算機に対して割込が起り、その図形単位名が割込の識別記号として割込処理待行列に格納される。格納された識別記号はGETIDF命令によって読み出すことができる。さらにGLIST命令とGJUMP命令の対使用によって、読み出した識別記号を予め指定した識別記号のリストとチェックし、一致したならばその識別記号に対応した文番号にプログラムを分岐するようにできる。

#### 4.1.2 トラッキング処理命令

これはCRT面上の位置座標をライトペンとトラッキング・シンボルによって計算機に入力する場合に使用される。トラッキング・シンボルは十字形をしており、GBGTRK命令によってCRT面上に表示される。表示されたトラッキング・シンボルはライトペン・スイッチがオンの間ライトペンの動きに追従する。トラッキング・シンボルが移動するたびに計算機に対して割込が起り、トラッキングシンボルの中心座標はGRDTRK命令で読むことができる。トラッキング・シンボルはGENTRK命令によってCRT面から消去される。

#### 4.1.3 データ入出力処理命令

これは整数型、実数型や文字型のデータを文字キーボードによって計算機に入力したり、逆に計算機のデータをCRT面上に表示する場合に使用される。データの入力にはGREADD命令

によって、またデータの出力表示はGWRITE命令によって行なうことができる。入力や出力のフォーマットはFORTRANのFORMAT文で与えられる。

#### 4.2 簡易GSPの機能

##### 4.2.1 GETIDF命令(Get Attention Identifier)

(1) 機能

割込処理待行列に格納された識別記号を発生した順に取り出す。

(2) 呼出し形式

```
CALL GETIDF (NAME1, NAME2)
```

(3) 引数の説明

NAME1, NAME2: 識別記号が格納される変数を整数型で指定する。

(4) 注意事項

- 1) NAME1, NAME2は定数でなく変数でなければならない。
- 2) この命令は割込が発生するまで待ち状態となるので注意しなければならない。
- 3) 取り出された識別記号は割込処理待行列から取り除かれる。

(5) 参考事項

1) この命令はつぎの表5に示すようなライトペン、ファンクション・キーボード、およびトラッキング・シンボルによって発生した割込に関する識別記号を取り出すことができる。

表5 割込と識別記号の種類

割込の種類	識別記号	
	1語	2語
ファンクション・キー番号1	0	1
ファンクション・キー番号2	0	2
}	{	}
ファンクション・キー番号32	0	32
ライトペンにトラッキング・シンボルが追随しているとき、 ライトペンのスイッチをオフにしたときに起る割込	0	34
ライトペンにトラッキング・シンボルが追随しているとき に起る割込(ライトペンのスイッチがオン)	0	35
図形単位名が(name1, name2)の図形をライトペン で指したときに起る割込	name1 (≠0)	name2

## 4.2.2 GLIST 命令 (Specify Name List)

## (1) 機能

つぎの GJUMP 命令で指定される文番号に対応する識別記号をリストで指定する。

## (2) 呼出し形式

```
CALL GLIST (name1-1, name1-2 [, name2-1, name2-2] .....
```

## (3) 引数の説明

name $i$ -1, name $i$ -2: GJUMP 命令の文番号に対応する識別記号を整数型の定数または変数で指定する。

## (4) 注意事項

- 1) この命令には必ず文番号をつけなければならない。
- 2) name $i$ -1, name $i$ -2 に添字が変数となる配列名は指定できない。

## (5) 参考事項

- 1) GLIST 命令は GJUMP 命令と対で使用されるが、この関係は FORTRAN の ASSIGN 文と割当て型 GO TO 文との関係に似ている。
- 2) この命令は FACOM GSP に属する GLIST 命令と同じである。

## 4.2.3 GJUMP 命令 (Jump to Statement Number)

## (1) 機能

この命令で引用した識別記号が GLIST 命令で指定された識別記号のどれかと一致すると、その識別記号に対応する、この命令で指定される文番号に分岐する。

## (2) 呼出し形式

```
CALL GJUMP (¥stno, id-1, id-2, ¥stnoi [, ¥stno2] .....
```

## (3) 引数の説明

stno : GJUMP 命令と対になる GLIST 命令の文番号  
 id-1, id-2 : GLIST 命令で指定された識別記号のどれと一致するかの判定をうける識別記号が格納されている変数を整数型で指定する。  
 stnoi : 分岐する文番号

## (4) 注意事項

- 1) id-1, id-2 は定数でなく変数でなければならない。
- 2) GLIST 命令で指定された識別記号が GJUMP 命令で指定した文番号 stnoi の数より多い場合は、はみ出た識別記号はすべて最後の文番号に対応する。

## (5) 参考事項

- 1) この命令は FACOM GSP に属する GLIST 命令と同じである。

## 4.2.4 GBGTRK 命令 (Begin Lightpen Tracking)

## (1) 機能

CRT 面上の指定された座標位置にトラッキング・シンボルを表示する。

## (2) 呼出し形式

```
CALL GBGTRK (xcoor, ycoor, distan)
```

## (3) 引数の説明

xcoor, ycoor : トラッキング・シンボルの表示位置の座標 (シンボルの中心座標)

distan : トラッキング・シンボルの移動間隔の長さを実数型の定数または変数で指定する。

## (4) 注意事項

1) 座標 (xcoor, ycoor) や長さ distan は mm 単位で表わされなければならない。

## (5) 参考事項

1) この命令の実行前にすでにトラッキング・シンボルが表示されていた場合は、この命令はその位置から座標 (xcoor, ycoor) の位置にトラッキング・シンボルが移動する働きとなる。

2) トラッキング・シンボルが表示されると以後このシンボルはライトペン・スイッチがオンの間ライトペンの動きに追随する。シンボルが移動するたびに識別記号 (0, 35) または (0, 34) の割込が発生し、識別記号は割込処理待行列に格納される ((0, 35) はライトペン・オンの間、(0, 34) はライトペン・オフのとき発生する)。

## 4.2.5 GRDTRK 命令 (Read the Current Location of the Tracking Symbol)

## (1) 機能

この命令が実行された時点でのトラッキング・シンボルの位置座標を取り出す。

## (2) 呼出し形式

```
CALL GRDTRK (XCOORD, YCOORD)
```

## (3) 引数の説明

XCOORD, YCOORD: トラッキング・シンボルの現在位置座標が格納される変数を実数型で指定する。

## (4) 注意事項

1) ライトペンの動きにトラッキング・シンボルが追随するたびに発生する割込を GETIDF 命令で知ってから GRDTRK 命令を呼出すこと。

2) XCOORD, YCOORD は定数でなく変数でなければならない。

## 4.2.6 GENTRK 命令 (End Lightpen Tracking)

## (1) 機能

CRT 面上からトラッキング・シンボルを消去する。

## (2) 呼出し形式

```
CALL GENTRK
```

## (3) 引数なし

## (4) 注意事項

1) この命令によって割込処理待行列にあるトラッキング・シンボルに関する識別記号はすべて取り除かれる。

## (5) 参考事項

1) この命令の実行前にすでにトラッキング・シンボルが消去されていた場合は、この命令は何もしない。

## 4.2.7 GREADD命令 (Read Data from Alphanumeric Keyboard)

## (1) 機能

CRT面上の指定された座標位置にカーソルを表示し、文字キーボードからキー・インされた文字列を指定されたフォーマットに従って整数型、実数型または文字型の定数に変換する。

## (2) 呼出し形式

```
CALL GREADD(xcoor, ycoor, height, DATA, angle, Yfmtno)
```

## (3) 引数の説明

xcoor, ycoor : カーソルの表示位置を示す座標  
 height : 表示される文字列の高さ  
 DATA : 変換されたデータが格納される変数または配列名を指定する。  
 angle : 文字列のX座標軸となす角度  
 fmtno : FORMAT文の文番号

## (4) 注意事項

- 1) 文字列の高さ height は  $\mu$  単位か、または VLG, LAG, SML, VSM のいずれか一つで指定が可能である。
- 2) DATA は定数でなく変数でなければならない。データの型は FORMAT 文に対応するように指定する。
- 3) 角度 angle は X 座標軸に対する左回りの度数を表わすが、現在は文字をハードウェア・シンボルにのみ限っているため 0 度のみ有効である。
- 4) FORMAT 文の中に許される欄記述子は F, E, D, I, A, H および X 型だけである。
- 5) この命令が実行されるとカーソルが指定された座標位置に表示され、以後 FROMAT 文に従ってデータの文字列が全部入力されるまでプログラムは待ち状態となるので注意しなければならない。
- 6) キー・インが完了すると (END キーが押されたとき), CRT 面に表示されていたデータの文字列は消去される。しかし入力された文字列はデータに変換され DATA に格納される。

## (5) 参考事項

- 1) FORMAT 文に従って入力中キー・インのミスをして END キーを押さない限り、この命令のレベルで ←, →, CANCEL キーを使って修正が可能である。
- 2) キー・インしたものをそのままの形でずっと表示させておく場合はつぎの GWRITE 命令が使用できる。

## 4.2.8 GWRITE命令 (Write Data on the CRT)

## (1) 機能

整数型、実数型または文字型のデータを指定されたフォーマットに従う文字列で表示するためのディスプレイ語列を作成し、これを GDOA に格納する。

## (2) 呼出し形式

```
CALL GWRITE(xcoor, ycoor, height, data, angle, Yfmtno)
```



## (3) 引数の説明

`xcoor, ycoor` : 表示される文字列の出発点(左下隅)の座標  
`height` : 文字列の高さ  
`data` : 表示するデータを定数または変数で指定する。  
`angle` : 文字列のX座標軸となす角度  
`fntno` : FORMAT文の文番号

## (4) 注意事項

- 1) 文字列の高さ `height` は  $\mu$ 単位か、または VLG, LAG, SML, VSMのいずれか一つで指定が可能である。
- 2) 角度 `angle` は X座標軸に対する左回りの度数を表わすが、現在は文字をハードウェア・シンボルにのみ限っているため 0度のみ有効である。
- 3) FORMAT文の中に許される欄記述子は F, E, D, I, A, H および X型だけである。
- 4) この命令はディスプレイ語列の作成を行なうものなので、この命令だけではデータの文字列は CRT 面上に表示されない。CRT 面上への表示は、PGS の拡張命令によって行なわなければならない。

## (5) 参考事項

- 1) この命令は、PGS の GSYMBL 命令(文字列)と GNUMBR 命令(数字の文字列)の拡張となっている。

## 5 汎用GSPによるグラフィック・プログラミング

### 5.1 汎用GSPの概要

前章までに述べてきたPGSと簡易GSPによって、グラフィック・プログラミングはGSPだけによるよりもっと簡単にできるようになった。しかしPGSと簡易GSPによる方法でもグラフィック・プログラミングはプロッタ・プログラミングに比べてまだ面倒である。そのため、PGSLIBにはおもにPGSと簡易GSPによって書かれた、より高レベルの汎用グラフィック処理命令があり、これらはまとめて汎用GSPと呼ばれる。この汎用GSPの使用によりユーザはPGSと簡易GSPによるプログラミング部分を少なくできる。

汎用GSPの命令は

- (1) 二次元図形表示命令
- (2) 三次元図形表示命令
- (3) 汎用グラフ表示命令
- (4) テキスト処理命令
- (5) 対話型計算処理命令
- (6) プロッタ・タイプ・シンボル命令
- (7) ライトペン・スムージング命令

の七つに大別される。

#### 5.1.1 二次元図形表示命令

これは二次元の基本的な図形や種々のグラフを表示するための命令からなる。各命令はプロッタ用のファンクショナル・サブルーチン群がPGSによってグラフィック化されたものであり、したがって機能的にプロッタ用のものとほとんど同じである。

#### 5.1.2 三次元図形表示命令

これは三次元配列として与えられた三次元図形に関するデータを透視図として表示するための命令からなる。各命令はプロッタ用のTHREEDサブルーチン群がPGSによってグラフィック化されたものであり、したがって機能的にプロッタ用のものとほとんど同じである。

#### 5.1.3 汎用グラフ表示命令

これは与えられたデータを自動スケールし、線形または対数の座標軸と共にグラフを表示するための命令からなる。特にGLOT1と呼ばれる命令は、プロッタ用のGPLOT<sup>(13)</sup>1がPGSによってグラフィック化されたものであり、グラフ表示がPGSボタンによって制御される点を除けば、プロッタ用のものとほとんど同じである。

#### 5.1.4 テキスト処理命令

これは文字の入出力処理に関してグラフィック・ディスプレイをキャラクタ・ディスプレイ的に処理しやすくするための命令からなる。したがって、多くの文字データに関する編集処理を対話形式で行なう場合のプログラミングに有効である。

#### 5.1.5 対話型計算処理命令

これはグラフ表示の動的な変更、データ・テーブルの表示、データ・テーブルでのデータの

追加・修正・削除の処理，ライトペン・ボタンがピックされたときのブランチ処理などを行なうための命令からなる。したがって，対話形式で数値計算やデータ処理を行なう場合のグラフィック・プログラミングに有効である。

#### 5.1.6 プロッタ・タイプ・シンボル命令

これはプロッタと同じように任意の大きさ，任意の角度でシンボルを表示するための命令である。この命令はSYMB4と呼ばれ，呼出し形式はGSYMBL命令と同じである。SYMB4命令は英数字や特殊文字などをソフトウェアで作成して表示する。

#### 5.1.7 ライトペン・スムージング命令

これは配列で与えられたデータをライトペンでスムージング（平滑化）するための命令で，GSMOOTと呼ばれる。

上に述べたように汎用GSPの命令は多岐にわたっており，命令の数も相当になるので，ここではつぎにPGSLIB/BASICに属するGPLOT1命令とGSMOOT命令についてのみ説明する。

## 5.2 汎用GSPの機能

### 5.2.1 GPLOT1命令 (Plot Data)

#### (1) 機能

自動スケーリング（リニア・リニア，リニア・ログ，ログ・リニア，ログ・ログ），同一スケールでの多重プロット，および多様なグラフ表示（直線プロット，点線プロット，シンボル・マーク・プロット）を行なう。

#### (2) 呼出し形式

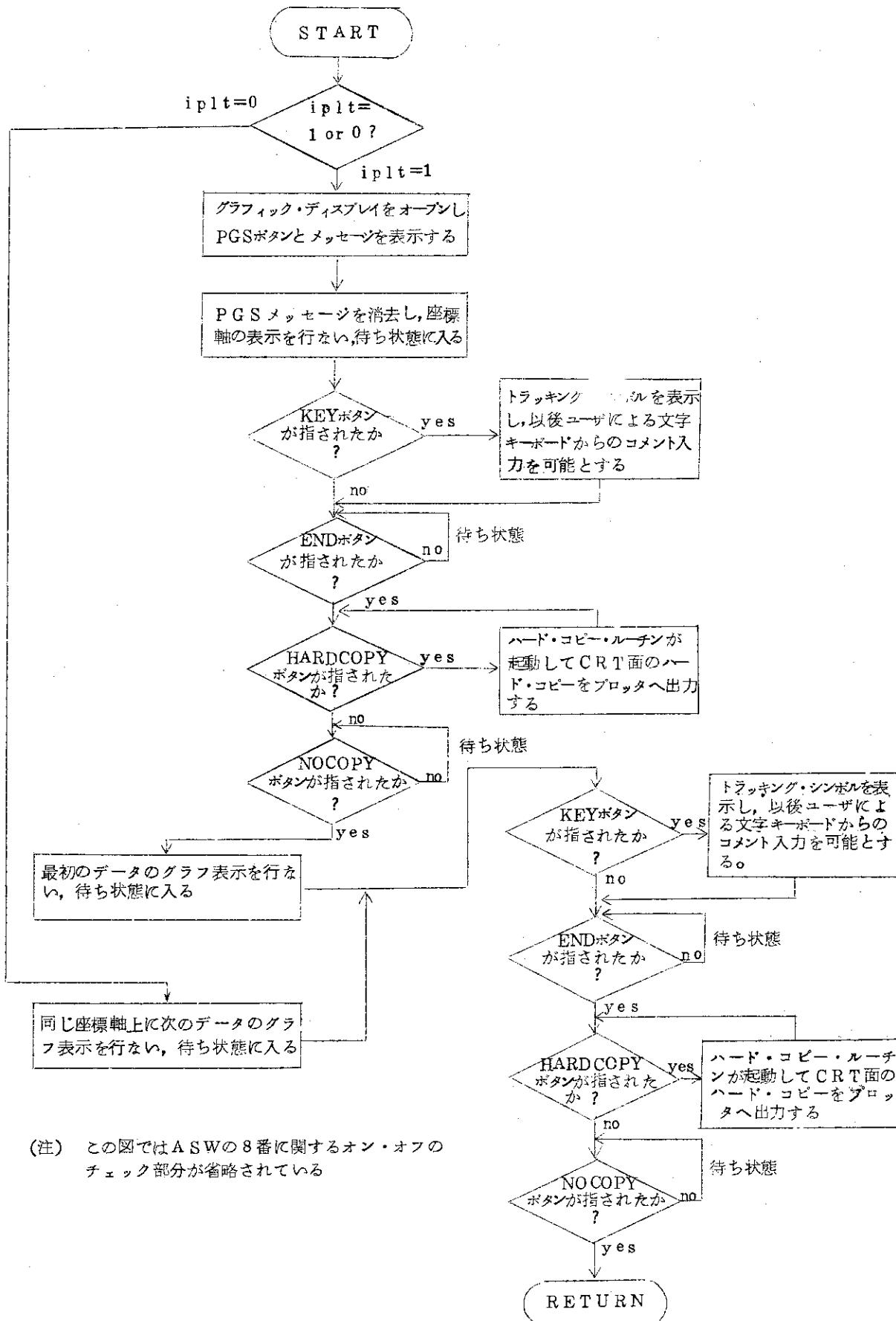
```
CALL GPLOT1 ( iplt, imax3, x, y, withx, withy, ip, np, ist, nlogx, nlogy,
              xwide, ywide, ixmin, iymin, ax1, ax2, ay1, ay2, mscale,
              ratiox, ratioy )
```

#### (3) 引数の説明

省略。詳細はプロッタ用GPLOT1に関するレポートであるJAERI-memo 4255<sup>(1)</sup>を参照のこと。

#### (4) 注意事項

- 1) この命令の中で最初にGPLOTS命令が呼ばれるので，ユーザ・プログラムでのGPLOTS命令の呼出しは不要である。
- 2) この命令が実行されるときグラフ表示の進行のためにPGSボタンの処理がユーザに要求される。ユーザは図9のフロー・チャートに従って処理すればよい。
- 3) その他使用上の注意についてはJAERI-memo 4255を参照のこと。



(注) この図ではASWの8番に関するオン・オフの  
チェック部分が省略されている

図9 Gplot1命令のフローチャート

## 5.2.2 GSMOOT 命令 (Smooth Data)

## (1) 機能

二つの配列に格納されているデータについて、ライトペンによるスムーザリングを可能にさせ、その結果を配列に格納する。

## (2) 呼出し形式

```
CALL GSMOOT(xdata, ydata, YSMOOT, ndata, ISMTH, nam1sc, nam2sc,
            nam1lb, nam2lb)
```

## (3) 引数の説明

- xdata, ydata: スムーザリングすべき x 方向および y 方向のデータが格納されている配列を、それぞれ実数型の配列名で指定する。
- YSMOOT : ライトペンによってスムーザリングされた結果が格納される配列を実数型の配列名で指定する。
- ndata : 配列 xdata または ydata に格納されているデータの個数を、整数型の定数または変数で指定する。
- ISMTH : スムーザリング・カーブが表示されているか否かの情報が格納される変数を整数型で指定する。  
 ISMTH=1 スムーザリング・カーブが表示されている。  
 ISMTH=0 スムーザリング・カーブが表示されていない。
- nam1sc, nam2sc : スムーザリング・カーブに付ける図形単位名を整数型の定数または変数で指定する。
- nam1lb, nam2lb : スムーザリング・コントロール用ライトペン・ボタンの図形単位名を整数型の定数または変数で指定する。

## (4) 注意事項

- この命令の呼出し前に、CRT 面上にスムーザリングすべきデータ (xdata(i), i=1, ndata) のグラフが表示されていなければならない。また、スムーザリング処理を制御するためのライトペン・ボタンがユーザの責任で表示されていなければならない。
- この命令が呼出される前にトラッキング・シンボルは CRT 面上に表示されていなければならない。
- 配列 xdata, ydata の大きさはそれぞれ ndata+2 以上でなければならない。さらに、xdata(ndata+1), xdata(ndata+2) にはそれぞれ xdata の最小値と増分、また ydata(ndata+1), ydata(ndata+2) にはそれぞれ ydata の最小値と増分が前もって格納されていなければならない。
- YSMOOT と ISMTH は定数でなく変数でなければならない。

## (5) 参考事項

- この命令の呼出しはスムーザリング・コントロール用のライトペン・ボタンがライトペンで指されたとき行なわれるようにするとよい。
- この命令が呼出されたら CRT 面上に表示されているトラッキング・シンボルをライトペンでひろい、スムーザリングの開始点を決めることができる。以後トラッキング・シンボルはライトペンの動きに追随すると共にスムーザリング・カーブが表示されてゆく。ラ

- イトペン・スイッチをオフにした時点でスムーディング処理は終わり、この命令が完了する。
- 3) この命令が呼出された時点で、もしスムーディング処理を直ちに止めたいときはスムーディング・コントロール用のライトペン・ボタンを指せばよい。
  - 4) 2) で表示されたスムーディング・カーブに満足できない場合は、くり返しこの命令が実行できるようにプログラミングするとよい。この命令が呼出されたときすでにスムーディング・カーブが表示されていればそれはすぐ消去される。
  - 5) この命令が呼出されたとき、ライトペン・スムーディングが始まる前に、配列 YSMOOT には配列 ydata がそのまま格納され、つぎにスムーディングされた部分のみ YSMOOT が変更される。
  - 6) YSMOOT (ndata+1) には ydata(ndata+1) が格納され、また YSMOOT (ndata+2) には ydata(ndata+2) が格納される。
  - 7) スムーディング・カーブに付けた図形単位名 (nam1sc, nam2sc) を利用して、この命令が実行された後でユーザの責任でスムーディング・カーブの処理が可能となる (たとえばスムーディング・カーブを表示させておくか、消去してしまうかなどの処理)。このとき ISMTH の値はユーザの責任で適切にセットされなければならない。

## 6 グラフィック・プログラミングの例

前章までに述べてきたPGSLIB/BASICの命令を用いてグラフィック・プログラムを作成し、実行した具体例をプロッタ・プログラムとの対比で示す。

## 6.1 例題

今A, B, Cを一枚の入力カードから与えられるパラメータとし,  $X(I)$  ( $I = 1, 2, \dots, 32$ )を

$$X(1) = -15.0, X(2) = -14.0, \dots, X(32) = 16.0$$

のように1.0きざみで変化する変数とする。このとき, 二次関数

$$Y(I) = A * (X(I) - B) ** 2 + C$$

を計算し, そのグラフを描くプログラムを考える。プログラムはA = 0が入力された時点で終了するものとする。

さて, 上記の問題に対して, つぎの三通りの方法でプログラムを作り実行させたものを, それぞれ表6, 表7, 表8に示す。

## (1) プロッタ・プログラム

一ページ, 一ケースとしてプロッタにグラフ出力する。ここで一ページはプロッタ用紙上で巾30cmとする。

## (2) グラフィック・プログラム(その一)

一画面, 一ケースとしてCRT面上にグラフ表示する。

## (3) グラフィック・プログラム(その二)

(2)と同じ方法でグラフ表示後, つぎのような三つの処理が選択して行なえるものとする。

- (a) ライトペンによる次のケースへの移行,
- (b) ライトペンによる再計算の実行とその結果のグラフ表示,
- (c) ライトペンと文字キーボードによるパラメータの変更。

ここで, (b), (c)については, グラフ表示の結果に満足できるまで繰返し操作可能とする。

## 6.2 各プログラムの説明

表6, 表7, 表8の各プログラムをISN(Internal Statement Number)に従って詳しく説明する。

## 6.2.1 プロッタ・プログラム

ISN

- 3           プロッタ・テープをオープンする。
- 4~6        Xの値の設定
- 7, 8       (Y20)入力カードからパラメータA, B, Cを読む。
- 9           Aが0ならばISN20(Y900)へ分岐する。
- 10, 11     (Y25) A, B, CおよびXの値からYの値を計算する。

- 12, 13 XおよびYの値に対するスケールリングを行なう。
- 14, 15 スケールリングの結果をもとにXおよびY軸を描く。
- 16 各X(I), Y(I)の値から座標値を求めてプロットイングし, 直線で結ぶ。
- 17 標題  $\nabla Y = A * (X - B) ** 2 + C \nabla$  を描く。
- 18, 19 X軸方向に30cm原点とペンを移動し; ISN7(¥20)へ分岐する。
- 20, 21 (¥900) プロッタ・テープをクローズしてプログラムを終了する。

6.2.2 グラフィック・プログラム(その一)

ISN

- 3 グラフィック・ディスプレイ装置と, さらに必要ならば(プロッタへのハード・コピーをとる場合でASWの8番がオフならば), プロッタ・テープをオープンする。
- 4~17 プロッタ・プログラムと同じであるが, PGSLIB/BASICでは画面をつくらだすディスプレイ語列がコア・メモリ中のGDOAにつくられる。
- 18 GDOAにつくられたディスプレイ語列(画面)をディスプレイ装置のCRT面上に表示し, ライトペンによる文字キーボードからのコメント入力を可能とする。
- 19 CRT面上の画面をプロッタ・テープ上にハード・コピーしたい場合は, ライトペンでHARDCOPYボタンを指す。そうでなければNOCOPYボタンを指すことで次のステップに進む。
- 20, 21 CRT画面を消去してISN7(¥20)へ分岐する。
- 22, 23 (¥900) グラフィック・ディスプレイ装置を(必要ならプロッタ・テープも)クローズしてプログラムを終了する。

6.2.3 グラフィック・プログラム(その二)

ISN

- 1~19 グラフィック・プログラム(その一)と全く同じである。
- 20 これからつくられる画面に(▽EL▽, ▽RN▽)という図形単位名をつける。
- 21 ▽READ A NEXT CARD▽と表示する画面をつくる。したがってこの図形単位名は(▽EL▽, ▽RN▽)となる。
- 22~34 ISN20, 21のように順次つぎのような図形単位名付きの画面をつくる。

	図形単位名	画面の内容
22,23	(▽EL▽, ▽CA▽)	▽CALCULATE AGAIN▽
24,25	(▽EL▽, ▽CH▽)	▽CHANGE PARAM. FMT=F103▽
26~28 (¥50)	(▽EL▽, 1)	▽A= Aの値▽ (F103で描く)
29~31	(▽EL▽, 2)	▽B= Bの値▽ (F103で描く)
32~34	(▽EL▽, 3)	▽C= Cの値▽ (F103で描く)

- 35 ISN20~34でつくられた図形単位名付きの画面をCRT面上に表示する。
- 36~39 (¥60) 図形単位名がそれぞれ(▽EL▽, ▽RN▽), (▽EL▽, ▽CA▽), (▽EL▽, ▽CH▽)である三つの画面の内一つがライトペンで指されるまでプ



プログラムは待ち状態となる。ライトペンで指されると指された画面に対応して以後プログラムは文番号 ¥70, ¥80, ¥100 の内一つに分岐する。

- 40, 41 (¥70) 画面を消去して ISN7 (¥20) へ分岐する。したがってパラメータの再入力から実行することになる。
- 42, 43 (¥80) 画面を消去して ISN10 (¥25) へ分岐する。したがって Y の値の再計算から実行することになる。
- 44~47 (¥100) 図形単位名がそれぞれ ( $\nabla EL \nabla$ , 1), ( $\nabla EL \nabla$ , 2), ( $\nabla EL \nabla$ , 3) である三つの画面の内一つがライトペンで指されるまでプログラムは待ち状態となる。ライトペンで指されると指された画面に対応して以後プログラムは文番号 ¥110, ¥120, ¥130 の内一つに分岐する。
- 48~50 (¥110) パラメータ A を, FORMAT(F10.3) で文字キーボードから入力された値とおきかえる。そして ISN54 (¥150) へ分岐する。
- 51, 52 (¥120) パラメータ B を, FORMAT(F10.3) で文字キーボードから入力された値とおきかえる。そして ISN54 (¥150) へ分岐する。
- 53 (¥130) パラメータ C を, FORMAT(F10.3) で文字キーボードから入力された値とおきかえる。
- 54~56 (¥150) 図形単位名がそれぞれ ( $\nabla EL \nabla$ , 1), ( $\nabla EL \nabla$ , 2), ( $\nabla EL \nabla$ , 3) である三つの画面を消去して, ISN26 (¥50) へ分岐する。
- 57, 58 (¥900) グラフィック・ディスプレイ装置を(必要ならプロッタ・テープも)クローズしてプログラムを終了する。

### 6.3 各プログラムによる図形出力

図10はプロッタ・プログラムによりプロッタに出力された図形である。各に縮小して示してある。

図11はグラフィック・プログラム(その一)によりグラフィック・ディスプレイのCRT面上に表示された画面をプロッタにハード・コピー出力した図形である( ISN19 の実行時点でハード・コピーしたもの)。

図12はグラフィック・プログラム(その二)によりグラフィック・ディスプレイのCRT面上に表示された画面をディスプレイ装置の付属ハード・コピー装置により図形出力したものである( ISN 36 の実行時点でハード・コピーしたもの)。

```

1      ELEMENT TSTPLT
      C
      C * PLOTTER EXAMPLE *
      C
2      DIMENSION X(34), Y(34), BUFF(1024)
3      CALL PLOTS ( BUFF, 1024)
4      X(1)=-15.
5      DO 10 I=2,32
6      10 X(I)=X(I-1)+1.
7      20 READ (5,21) A, B, C
8      21 FORMAT (3F10,3)
9      IF ( A.EQ.0 ) GO TO 900
10     25 DO 30 I=1,32
11     30 Y(I)= A*(X(I)-B)**2+C
12     CALL SCALE ( X, 200., 32, 1, 10. )
13     CALL SCALE ( Y, 200., 32, 1, 10. )
14     CALL AXIS ( 0., 0., 'X - V A L U E', -13, 200., 0., X(33), X(34), 10. )
15     CALL AXIS ( 100., 0., 'Y - V A L U E', 13, 200., 90., Y(33), Y(34), 10. )
16     CALL LINE ( X, Y, 32, 1, 0, 0)
17     CALL SYMBOL( 60., 230., 8., 'Y=A*(X-B)**2+C', 0., 14 )
18     CALL PLOT ( 300., 0., -3 )
19     GO TO 20
20     900 CALL PLOT ( 0., 0., 999 )
21     STOP
22     END
    
```

表6 プロッタ・プログラム

```

1      ELEMENT TSTPGS
      C
      C * PGS EXAMPLE *
      C
2      DIMENSION X(34), Y(34), BUFF(1024)
3      CALL PLOTS ( BUFF, 1024)
4      X(1)=-15.
5      DO 10 I=2,32
6      10 X(I)=X(I-1)+1.
7      20 READ (5,21) A, B, C
8      21 FORMAT (3F10,3)
9      IF ( A.EQ.0 ) GO TO 900
10     25 DO 30 I=1,32
11     30 Y(I)= A*(X(I)-B)**2+C
12     CALL SCALE ( X, 200., 32, 1, 10. )
13     CALL SCALE ( Y, 200., 32, 1, 10. )
14     CALL AXIS ( 0., 0., 'X - V A L U E', -13, 200., 0., X(33), X(34), 10. )
15     CALL AXIS ( 100., 0., 'Y - V A L U E', 13, 200., 90., Y(33), Y(34), 10. )
16     CALL LINE ( X, Y, 32, 1, 0, 0)
17     CALL SYMBOL( 60., 230., 8., 'Y=A*(X-B)**2+C', 0., 14 )
18     C
19     CALL PLOT ( 300., 0., -3 )
20     CALL PLOT ( 0., 0., 777 )
21     CALL PLOT ( 0., 0., 888 )
22     CALL PLOT ( 0., 0., 666 )
23     GO TO 20
24     900 CALL PLOT ( 0., 0., 999 )
25     STOP
26     END
    
```

表7 グラフィック・プログラム(その二)

```

1      ELEMENT TSTGSP
      C
      C * SIMPLE GSP EXAMPLE *
      C
2      DIMENSION X(34), Y(34), BUFF(1024)
3      CALL PLOTS ('BUFF', 1024)
4      X(1)=-15.
5      DO 10 I=2,32
6      10 X(I)=X(I-1)+1.
7      20 READ (5,21) A, B, C
8      21 FORMAT (3F10.3)
9      IF (A.EQ.0) GO TO 900
10     25 DO 30 I=1,32
11     30 Y(I)= A*(X(I)-8)**2+C
12     CALL SCALE ( X, 200., 32., 1., 10. )
13     CALL SCALE ( Y, 200., 32., 1., 10. )
14     CALL AXIS ( 0., 0., 'X = V A L U E', -13, 200., 0., X(33), X(34), 10. )
15     CALL AXIS ( 100., 0., 'Y = V A L U E', 13, 200., 90., Y(33), Y(34), 10. )
16     CALL LINE ( X, Y, 32., 1., 0., 0 )
17     CALL SYMBOL ( 60., 230., B., 'Y=A*(X-B)**2+C', 0., 14 )
      C
18     CALL PLOT ( 0., 0., 777 )
19     CALL PLOT ( 0., 0., 888 )
20     CALL PLOT ('EL', 'RN', 333 )
21     CALL SYMBOL ( 220., 180., 6.5, 'READ A NEXT CARD', 0., 16 )
22     CALL PLOT ('EL', 'CA', 333 )
23     CALL SYMBOL ( 220., 165., 6.5, 'CALCULATE AGAIN', 0., 15 )
24     CALL PLOT ('EL', 'CH', 333 )
25     CALL SYMBOL ( 220., 150., 6.5, 'CHANGE PARAM. FMT=F10.3', 0., 23 )
26     50 CALL PLOT ('EL', 1, 333 )
27     CALL GWRITE ( 230., 135., 'LAG', A, 0., #51 )
28     51 FORMAT ('A=', F10.3)
29     CALL PLOT ('EL', 2, 333 )
30     CALL GWRITE ( 230., 120., 'LAG', B, 0., #52 )
31     52 FORMAT ('B=', F10.3)
32     CALL PLOT ('EL', 3, 333 )
33     CALL GWRITE ( 230., 105., 'LAG', C, 0., #53 )
34     53 FORMAT ('C=', F10.3)
35     CALL PLOT ( 0., 0., 444 )
      C
36     60 CALL GETIDF ( 11, 12 )
37     CALL GJUMP ( #61, 11, 12, #70, #80, #100 )
38     61 CALL GLIST ('EL', 'RN', 'EL', 'CA', 'EL', 'CH' )
39     GO TO 60
40     70 CALL PLOT ( 0., 0., 666 )
41     GO TO 20
42     80 CALL PLOT ( 0., 0., 666 )
43     GO TO 25
      C
44     100 CALL GETIDF ( 11, 12 )
45     CALL GJUMP ( #101, 11, 12, #110, #120, #130 )
46     101 CALL GLIST ('EL', 1, 'EL', 2, 'EL', 3)
47     GO TO 100
48     110 CALL GREADD ( 243., 128., 'LAG', A, 0., #111 )
49     GO TO 150
50     111 FORMAT (F10.3)
51     120 CALL GREADD ( 243., 113., 'LAG', B, 0., #111 )
52     GO TO 150
53     130 CALL GREADD ( 243., 98., 'LAG', C, 0., #111 )
54     150 DO 160 I=1,3
55     160 CALL PLOT ('EL', 1, 666 )
56     GO TO 50

```

```

57     900 CALL PLOT ( 0., 0., 999 )
58     STOP
59     END

```

表8 グラフィック・プログラム(その一)

$$Y = A * (X - B) ** 2 + C$$

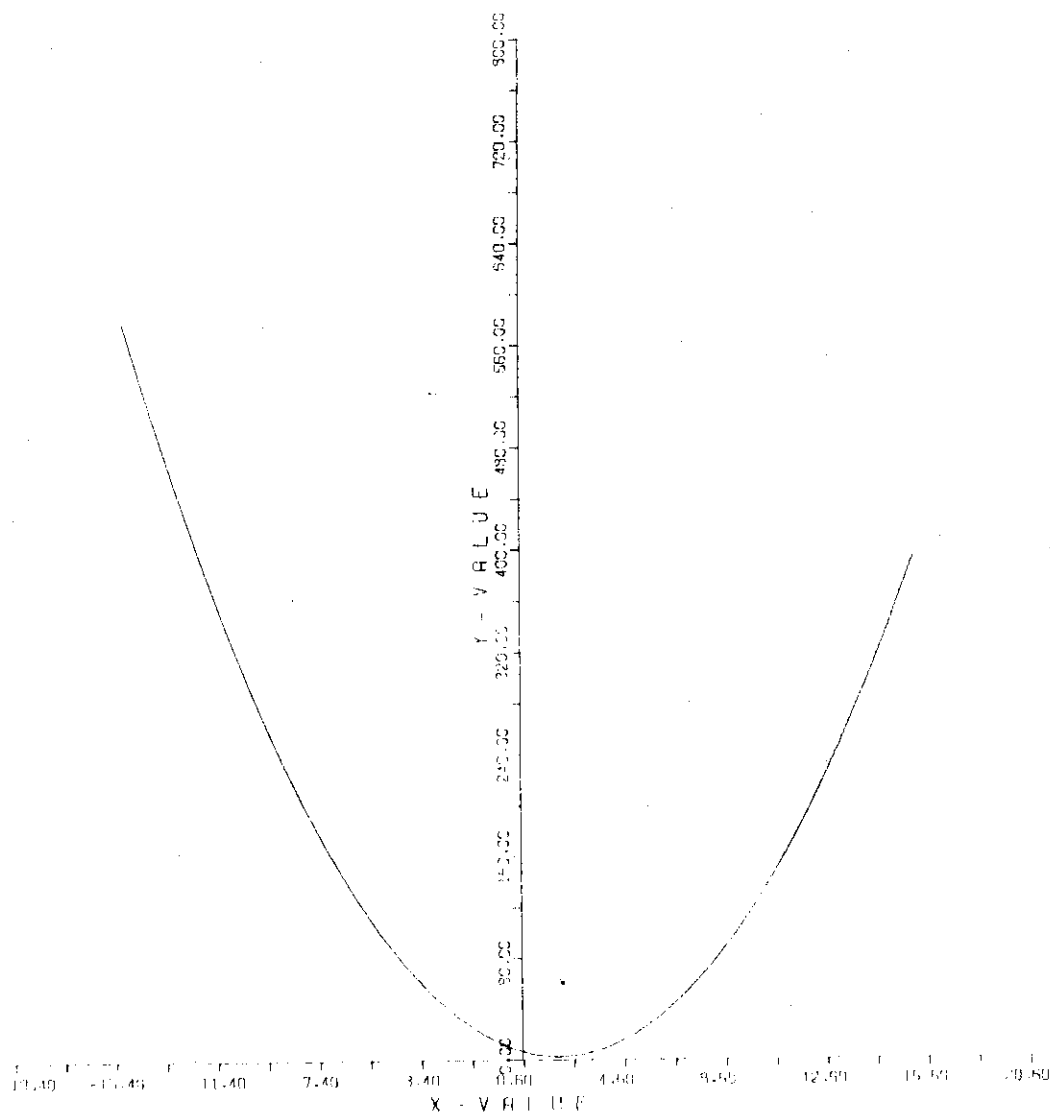


図 10 プロッタによる図形出力

$$Y=A*(X-B)**2+C$$

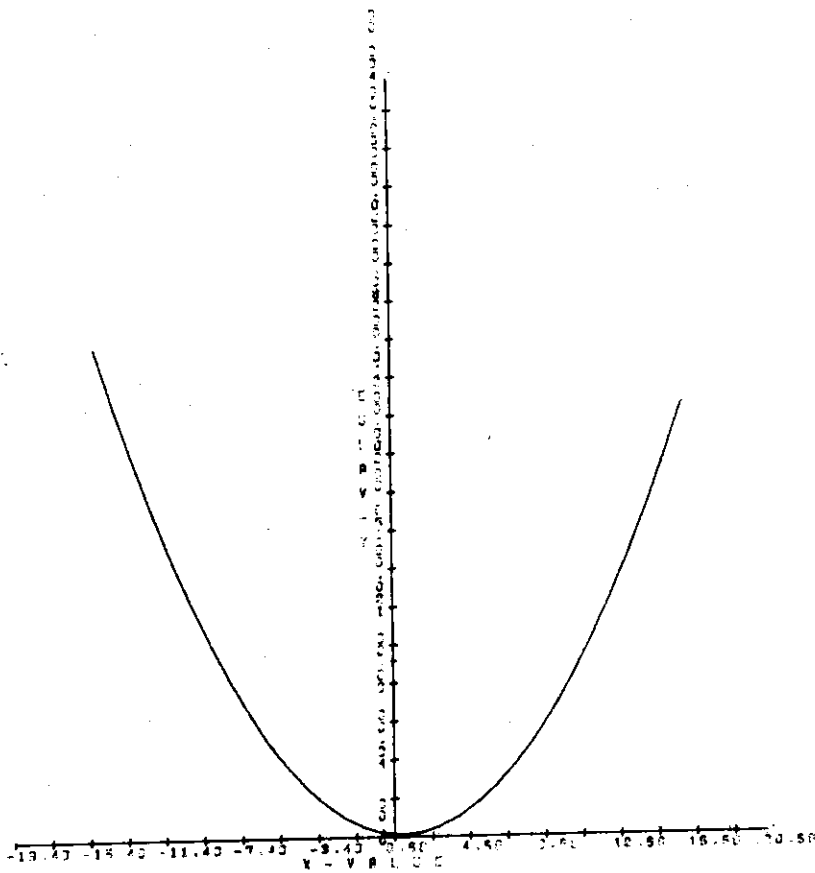
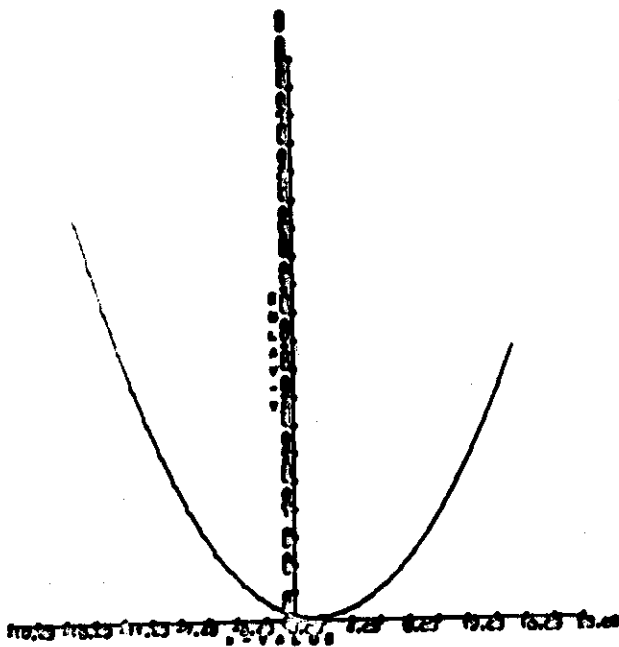


図11 プロッタ・ハード・コピーによる図形出力

$$Y=A*(X-B)**2+C$$



READ A NEXT CARD  
 CALCULATE AGAIN  
 CHANGE PARAM. FMT=F10.3  
 A= 2.000  
 B= 2.000  
 C= 2.000.

KEY  
 END  
 VLB  
 LAB  
 SML  
 VSM

HARDCOPY NOCOPY

図12 ハード・コピー装置による図形出力

## 7 お わ り に

PGSLIBは約150個のグラフィック処理命令からなるが、本レポートではその内グラフィック・プログラミングの上で基本的でかつよく使用される命令の集まりであるPGSLIB/BASICについてその使用法を詳しく述べてきた。

このPGSLIB/BASIC(またはPGSLIB)の使用により、ユーザはプロッタ・プログラムを基に、ライトペン、ファンクション・キーボードおよび文字キーボードなどを用いた対話型のグラフィック・プログラムを作ることができる。実際、現在までに開発されたグラフィック・プログラムの大半は、プロッタ・プログラムに対話処理のプログラムを追加する形で実現されている。このようにユーザはプロッタ・プログラミングの経験があれば、対話処理に関するプログラミングを覚えるだけで比較的簡単にグラフィック・プログラミングが可能となる。

PGSLIB/BASICの別な使い方として、グラフィック・ディスプレイによるプロッタ・プログラムのデバッグがある。グラフィック・ディスプレイの場合、図形は高速度でCRT面上に表示されるので、す早く計算結果を見たり、プログラムのデバッグができる。また、このとき必要に応じてCRT面上に表示された図形のハード・コピーをプロッタまたはハード・コピー装置で取ることができる。

PGSLIB/BASICは、プロッタとグラフィック・ディスプレイ(高級型)の間でのみプログラムの互換性が考慮されている。そこで現在、PGSLIB/BASICを拡張して、プロッタ、COM(Computer Output Microfilmer),およびグラフィック・ディスプレイ(簡易型と高級型の両方)の四つの装置間で上向きのプログラム互換性をもつグラフィック・サブルーチン・ライブラリGSLを設計・開発中である。

なお、PGSLIB/BASICによるグラフィック・ディスプレイの使い方不明な点は計算センター(中村, 小沼 - Tel 5369, 居室211)まで御相談下さい。

## 参 考 文 献

- (1) 富士通株: FACOMディスプレイ装置Ⅲ(2)グラフィック・ディスプレイ装置FACOM 6233A, 1972年9月
- (2) 富士通株: FACOM GSP文法編, 1973年4月
- (3) 小沼吉男, 笹本宣雄, 田坂完二, 中村康弘: グラフィック・ディスプレイを使った $\gamma$ 線スペクトルの平滑化, 昭和48年日本原子力学会年会(炉物理・炉工学)要旨集, 1973年3月
- (4) 中村康弘, 小沼吉男, 黒井英雄: ARCADIA-I: グラフィック・ディスプレイによる核断面積データのオンライン修正システム(内そうによる修正), JAERI-M 5658, 1974年4月
- (5) 中村康弘, 小沼吉男, 小林健介, 鈴木忠和: GORVFIT: 直交多項式を用いた最小二乗法による対話型カーブ・フィッティング・システム, JAERI-M. 5596, 1974年3月
- (6) 中村康弘, 鈴木忠和, 小林健介, 小沼吉男: GLP1: リニア・プログラミングによる対話

## 7 お わ り に

PGSLIBは約150個のグラフィック処理命令からなるが、本レポートではその内グラフィック・プログラミングの上で基本的でかつよく使用される命令の集まりであるPGSLIB/BASICについてその使用法を詳しく述べてきた。

このPGSLIB/BASIC(またはPGSLIB)の使用により、ユーザはプロッタ・プログラムを基に、ライトペン、ファンクション・キーボードおよび文字キーボードなどを用いた対話型のグラフィック・プログラムを作ることができる。実際、現在までに開発されたグラフィック・プログラムの大半は、プロッタ・プログラムに対話処理のプログラムを追加する形で実現されている。このようにユーザはプロッタ・プログラミングの経験があれば、対話処理に関するプログラミングを覚えるだけで比較的簡単にグラフィック・プログラミングが可能となる。

PGSLIB/BASICの別な使い方として、グラフィック・ディスプレイによるプロッタ・プログラムのデバッグがある。グラフィック・ディスプレイの場合、図形は高速度でCRT面上に表示されるので、す早く計算結果を見たり、プログラムのデバッグができる。また、このとき必要に応じてCRT面上に表示された図形のハード・コピーをプロッタまたはハード・コピー装置で取ることができる。

PGSLIB/BASICは、プロッタとグラフィック・ディスプレイ(高級型)の間でのみプログラムの互換性が考慮されている。そこで現在、PGSLIB/BASICを拡張して、プロッタ、COM(Computer Output Microfilmer),およびグラフィック・ディスプレイ(簡易型と高級型の両方)の四つの装置間で上向きのプログラム互換性をもつグラフィック・サブルーチン・ライブラリGSLを設計・開発中である。

なお、PGSLIB/BASICによるグラフィック・ディスプレイの使い方不明な点は計算センター(中村, 小沼 - Tel 5369, 居室211)まで御相談下さい。

## 参 考 文 献

- (1) 富士通㈱: FACOMディスプレイ装置Ⅲ(2)グラフィック・ディスプレイ装置FACOM 6233A, 1972年9月
- (2) 富士通㈱: FACOM GSP文法編, 1973年4月
- (3) 小沼吉男, 笹本宣雄, 田坂完二, 中村康弘: グラフィック・ディスプレイを使った $\gamma$ 線スペクトルの平滑化, 昭和48年日本原子力学会年会(炉物理・炉工学)要旨集, 1973年3月
- (4) 中村康弘, 小沼吉男, 黒井英雄: ARCADIA-I: グラフィック・ディスプレイによる核断面積データのオンライン修正システム(内そうによる修正), JAERI-M 5658, 1974年4月
- (5) 中村康弘, 小沼吉男, 小林健介, 鈴木忠和: GCRVFIT: 直交多項式を用いた最小二乗法による対話型カーブ・フィッティング・システム, JAERI-M. 5596, 1974年3月
- (6) 中村康弘, 鈴木忠和, 小林健介, 小沼吉男: GLP1: リニア・プログラミングによる対話

- 型カーブ・フィッティング・システム, JAERI-M 5597, 1974年3月
- (7) 幾島 毅, 小沼吉男, 中村康弘: GTOROTO-3, 4: グラフィック・ディスプレイによるブロック状炉心の地震シミュレーション・プログラム, JAERI-M 5981, 1975年1月
- (8) 中村康弘, 小沼吉男, 小林健介, 鈴木忠和: グラフィック・ディスプレイの使い方と各種グラフィック・プログラムの概要 (F230-35 ROS), JAERI-M 5659, 1974年4月
- (9) 吉沢ビジネス・マシズ機: CALCOMPプログラミング・マニュアル-I, 1969年11月
- (10) 日本IBM機: 2250映像表示装置/GSP, 1971年11月
- (11) 中村康弘, 小沼吉男, 小林健介, 鈴木忠和: プロッタ・プログラムをグラフィック・プログラムとして利用する方法, 情報処理学会第13回大会予稿集, 1972年12月
- (12) 中村康弘, 小沼吉男: 科学技術計算用グラフィック・サブルーチン・ライブラリPGSLIBについて, 情報処理学会第14回大会予稿集, 1973年12月
- (13) 長谷川 明: 汎用グラフ作成サブルーチンGPLOT1, 自動グラフ作成コードGPLOTCの開発, JAERI-memo 4255(公開), 1970年12月
- (14) 吉沢ビジネス・マシズ機: CALCOMPプログラミング・マニュアル-II, 1969年12月
- (15) CALCOMP, INC.: THREEED: A Perspective Drawing Software System  
July 1969
- (16) 小沼吉男, 中村康弘: テキスト・モード・グラフィック・サブルーチン・パッケージ,  
JAERI-M(to be published)
- (17) 小沼吉男, 中村康弘: 対話型計算処理用グラフィック・サブルーチン・パッケージ,  
JAERI-M(to be published)
- (18) 中村康弘, 小沼吉男: PGSLIB-A Plotter-Compatible Graphic Subroutine  
Library, JAERI-M(to be published)
- (19) 吉沢ビジネス・マシズ機: CALCOMP Graphic Output System 900/937/  
1136
- (20) 中村康弘, 小沼吉男: グラフィック・ディスプレイの使い方, JAERI-M(to be  
published)