

JAERI-M
6711

簡易GSPによるグラフィック・
プログラミング

1976年9月

中村 康弘・小沼 吉男

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.

簡易GSPによるグラフィック・プログラミング

日本原子力研究所東海研究所計算センター

中村 康弘 ・ 小沼 吉男

(1976年8月19日受理)

簡易GSPはFACOM GSP を単純化することによって開発された、対話型グラフィック処理のための一つのプログラミング用ソフトウェアである。これは18個のグラフィック処理命令からなり、これらの命令は図形表示と対話処理のための2つの命令群に大別される。本報告では、これらの命令の機能と使い方について詳しく述べた。

Graphic Programming by the Simplified GSP

Yasuhiro NAKAMURA and Yoshio ONUMA

Computing Center, Tokai, JAERI

(Received August 19, 1976)

The simplified GSP is a programming software for interactive computer graphics, developed by simplifying the FACOM GSP. It has 18 graphic instructions; these are divided into two instruction groups for displaying graphic figures and processing man-computer interaction. Functions and usage of the instructions are described in detail.

目 次

| | |
|-----------------------------|----|
| 1. はじめに | 1 |
| 2. 簡易GSPの概要 | 2 |
| 2.1 簡易GSPの命令 | 2 |
| 2.1.1 GDS定義命令 | 2 |
| 2.1.2 ディスプレイ語作成命令 | 2 |
| 2.1.3 ディスプレイ制御命令 | 2 |
| 2.1.4 割込み処理命令 | 2 |
| 2.1.5 トラッキング処理命令 | 2 |
| 2.1.6 補助入出力処理命令 | 2 |
| 2.2 GDS, 図形単位, 座標系と文字 | 5 |
| 2.2.1 GDS | 5 |
| 2.2.2 図形単位 | 5 |
| 2.2.3 座 標 系 | 5 |
| 2.2.4 文字の表示 | 5 |
| 3. 簡易GSPの機能 | 7 |
| 3.1 GDS定義命令 | 7 |
| 3.1.1 GDASET命令 | 7 |
| 3.2 ディスプレイ語作成命令 | 8 |
| 3.2.1 GASSDS命令 | 8 |
| 3.2.2 GENAME命令 | 8 |
| 3.2.3 GWRITE命令 | 9 |
| 3.3 ディスプレイ制御命令 | 9 |
| 3.3.1 GDSPLY命令 | 10 |
| 3.3.2 GEMOVE命令 | 10 |
| 3.3.3 GALTER命令 | 11 |
| 3.3.4 GERASE命令 | 12 |
| 3.4 割込み処理命令 | 12 |
| 3.4.1 GAMASK命令 | 13 |
| 3.4.2 GETIDF命令 | 14 |
| 3.4.3 GLIST命令 | 15 |
| 3.4.4 GJUMP命令 | 15 |
| 3.5 トラッキング処理命令 | 16 |
| 3.5.1 GBGTRK命令 | 16 |
| 3.5.2 GRDTRK命令 | 17 |
| 3.5.3 GENTRK命令 | 17 |

| | | |
|-------|-----------------|----|
| 3.6 | 補助入出力処理命令 | 17 |
| 3.6.1 | GREADD 命令 | 18 |
| 3.6.2 | GRDASW 命令 | 19 |
| 3.6.3 | GSLAMP 命令 | 19 |
| 4. | 使用例 | 21 |
| 4.1 | 例題 | 21 |
| 4.2 | グラフィック・プログラム | 21 |
| 5. | 使用上の注意 | 27 |
| 5.1 | グラフィック・プログラムの実行 | 27 |
| 5.2 | エラー・メッセージ | 27 |
| 6. | おわりに | 30 |
| | 参考文献 | 30 |

1. は じ め に

FACOM230-75のグラフィック・ディスプレイ⁽¹⁾ (FACOM6233A) に対するプログラミングは、通常FACOM GSP⁽²⁾ (Graphic Subroutine Package), または PGSLIB/BASIC^{(3), (4)} (Plotter-compatible Graphic Subroutine Library)を用いて行うことができる。

FACOM GSP は、42個の基本命令からなり、広範囲なグラフィック処理に対して適用可能であるが、プログラミングが細かく、複雑になりがちである。また、一般にGSPは、異なる電子計算機間、または異なるグラフィック・ディスプレイ間で、プログラムの互換性に乏しい面がある。

一方、PGSLIB/BASICの場合は、プロッタ・プログラム、またはCOM(Computer Output Microfilm System) プログラムを容易に図形表示用のプログラムとして利用でき、かつ、これにライトペンや文字キーボードなどを用いる対話処理用のプログラムを追加するだけで、対話型のグラフィック・プログラムに発展させることができる。しかし、FACOM GSPに比べると、かなり命令が制限されているので、書きにくい面がある。

そこで、PGSLIB/BASICの基本部分であるPGS⁽³⁾の座標系と長さの単位に合わせてFACOM GSPを単純化し、さらに基本命令とそのパラメータについてグラフ処理を中心とした適用分野から整理したのが簡易GSP (Simplified GSP)である。簡易GSPはFACOM GSPとPGSとの間に位置し、PGSに比べて、より広範囲な命令をもっている。また、簡易GSPの一部の命令(対話処理に関するもの)は、PGSLIB/BASICに含まれている。したがって、PGSLIB/BASICの命令に加えて簡易GSPの命令を用いることによって、プロッタ・プログラム、またはCOMプログラムを基に、より本格的な対話型グラフィック・プログラムの作成ができる。

以下、まずこの簡易GSPの概要について述べ、つぎに簡易GSPに属する個々の命令について、その機能、呼出し形式、引数の意味、注意事項、および参考事項などについて詳しく述べる。さらに、簡易GSPを使用した具体的なプログラム例を掲げ、それについて説明する。

2. 簡易GSPの概要

2.1 簡易GSPの命令

簡易GSPの命令は、Table 2.1とTable 2.2に示すようにGDS定義命令、ディスプレイ語作成命令、ディスプレイ制御命令、割込み処理命令、トラッキング処理命令、および補助入出力処理命令の6つに大別される。

2.1.1 GDS定義命令

これは、ディスプレイ語(グラフィック・ディスプレイに対する命令)の格納される領域としてのGDS(Graphic Data Set)の定義とその初期化を行う命令である。

なお、グラフィック・ディスプレイのオープンとクローズはPGSの命令によって行われる。

2.1.2 ディスプレイ語作成命令

これは、使用するGDSを指定する命令、図形単位に名前を付ける命令、および文字列の表示に関するディスプレイ語列(ディスプレイ語の並び)を作成する命令からなる。

なお、ビームのセット、点の表示、およびベクトルの表示に関するディスプレイ語列の作成はPGSの命令によって行われる。

2.1.3 ディスプレイ制御命令

これは、GDS内に作成されたディスプレイ語列をグラフィック・ディスプレイのバッファ・メモリへ転送してCRT面に図形を表示する命令、CRT面に表示された図形単位を任意の位置に平行移動する命令、表示に関する特性を変更する命令、およびCRT面の図形を消去する命令からなる。

2.1.4 割込み処理命令

これは、ファンクション・キー、またはトラッキングによって起る割込みを受付けるか否かの指定を行う命令、割込みの識別記号を読取る命令、割込みに伴ってプログラムの流れを制御する命令、および割込みの識別記号をリストで指定する命令からなる。ここで、最初のもの以外はPGSLIB/BASICにも含まれている。

2.1.5 トラッキング処理命令

これは、トラッキング・シンボルを発生させる命令、トラッキング・シンボルの位置座標を読取る命令、およびトラッキング・シンボルを消去する命令からなる。これらはすべてPGSLIB/BASICにも含まれている。

2.1.6 補助入出力処理命令

これは、文字キーボードから各種のデータを入力する命令、ASW(Alteration Switch)の状態を読取る命令、およびファンクション・キーのランプの点灯、または消滅を行う命令からなる。ここで、最初のものはPGSLIB/BASICにも含まれている。

なお、文字キーボードによるコメント入力やCRT面の図形のハード・コピー出力(COM、またはプロッタ)は、PGSの命令によって行われる。

Table 2.1 簡易GSPの命令I (図形表示関係)

| 分類 | 命 令 | 機 能 |
|-----------|--|---|
| GDS定義命令 | (1) GDASET | GDSを定義し、かつその領域の初期化を行う。 |
| ディスプレイ命令語 | (1) GASSDS (2) GENAME (3) GWRITE | 使用するGDSを指定し、かつその領域の初期化を行う。 この命令以後に作成される図形単位に名前を付ける。 整数型、実数型、または文字型のデータを指定されたFORMAT文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GDSに格納する。 |
| ディスプレイ命令 | (1) GDSPLY (2) GEMOVE (3) GALTER (4) GERASE | 指定されたGDS内に作成されたディスプレイ語列をディスプレイ・バッファへ転送し、CRT面に図形を表示する。 CRT面に表示されている図形単位を指定された位置に平行移動する。 CRT面に表示されている図形単位の表示特性を指定されたものに変更する。 CRT面の全図形、または指定された図形単位を消去する。 |

Table 2.2 簡易GSPの命令Ⅱ（対話処理関係）

| 分類 | 命 令 | 機 能 |
|--------------------------------------|------------|--|
| 割 込 み 処 理 命 令 | (1) GAMASK | ファンクション・キー（1～32），およびトラッキング（34，35）による割込みの中でどれを受入れ，どれを受入れないかを指定する。 |
| | (2) GETIDF | 割込み処理の待行列に格納された識別記号を発生した順に取出す。 |
| | (3) GLIST | 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。 |
| | (4) GJUMP | 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た文番号に分岐する。 |
| ト ラ ッ キ ン グ 命 令 | (1) GBGTRK | 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) GRDTRK | 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讀取る。 |
| | (3) GENTRK | CRT面上からトラッキング・シンボルを消去する。 |
| 補 処 理 入 出 力 命 令 | (1) GREADD | 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入力された文字列を指定されたFORMAT文に従って，整数型，実数型，または文字型のデータに変換し，かつ指定された変数に格納する。 |
| | (2) GRDASW | ディスプレイ・コンソール上のASWの状態を讀込む。 |
| | (3) GSLAMP | ファンクション・キー（1～32）のランプの点灯，消滅を行う。 |

2.2 GDS, 図形単位, 座標系と文字

第3章での簡易GSPの命令に関する説明に入る前に, 簡易GSPによるグラフィック・プログラミングの際の基本的な事項である, GDS, 図形単位, 座標系, および文字の表示について述べる。

2.2.1 GDS

すでに2.1で述べたように, GDSはGDS定義命令によって定義され, かつその領域が初期化される。ディスプレイ語作成命令によって作成されたディスプレイ語列はGDSに格納されてゆく。ディスプレイ制御命令によって, ディスプレイ語列がGDSからグラフィック・ディスプレイのバッファ・メモリへ転送されるとCRT面に図形が表示される。このディスプレイ語列のバッファ・メモリへの転送はGDS単位で行われる。

定義されたGDSは, ディスプレイ語作成命令によってGDSが指定されるたびに, その領域が初期化(消去)される。しかし, GDS内のディスプレイ語列に対応する図形が, ディスプレイ制御命令によってCRT面から消去されても, GDSの内容は保存される。したがって, ディスプレイ語列を再び作ることなく, 消去された図形の再表示がGDSによって可能である。

GDSは利用者のプログラム内に幾つでも作成できる。もし, 1つのGDSで十分な場合は, GDSの定義や指定, およびGDSの転送に関する機能はPGSの命令で間に合う。この場合, GDSは内部的に作成され, 利用者による定義や指定は表面上行われない。

このGDSの概念は, FACOM GSPのGDOA (Graphic Data Output Area), またはIBM2250 GSP⁽⁵⁾のGDSに類似したものである。

2.2.2 図形単位

CRT面に表示される図形のある一まとまりは, 図形単位, またはエレメント(Element)と呼ぶ。これは1つのGDS内に複数個作成可能である。図形単位にはディスプレイ語作成命令によって名前を付けることができる。名前の付けられた図形単位は, ライトペンによって認識可能である。すなわち, CRT面に表示された図形単位がライトペンによってピックされると割込みが起り, その図形単位の名前が割込みの識別記号として割込み処理命令によって読取られる。図形単位の名前は(name 1, name 2)で与えられる。ここでname 1, name 2は整数型の定数, または変数である(内容は文字データでもよい)。

2.2.3 座標系

点, ベクトル, または文字の合成によって図形を描くときの基本となる座標系は, つぎのFig.2.1に示すようにPGSの場合と全く同じである。最初の原点はCRT面の左下端にあり, 図形表示の範囲はx軸方向40cm, y軸方向28cmの矩形で囲まれる部分が適当である。座標の指定は, PGSの場合と同じく実数データで, かつ絶対座標方式に限られる。座標や長さの単位はmmである。

2.2.4 文字の表示

文字キーボードから入力したり, CRT面に表示できる文字は英数字と特殊文字が可能である。これらをCRT面に表示するとき, 文字の大きさはつぎのTable 2.3に示すようにVSM, SML, LAG, VLGの4種類が可能である。プログラミングの際, 文字の大きさの指定は, Table 2.3に示すような記号による方法と, 文字の高さをmm単位で与える方法の2通りがあ

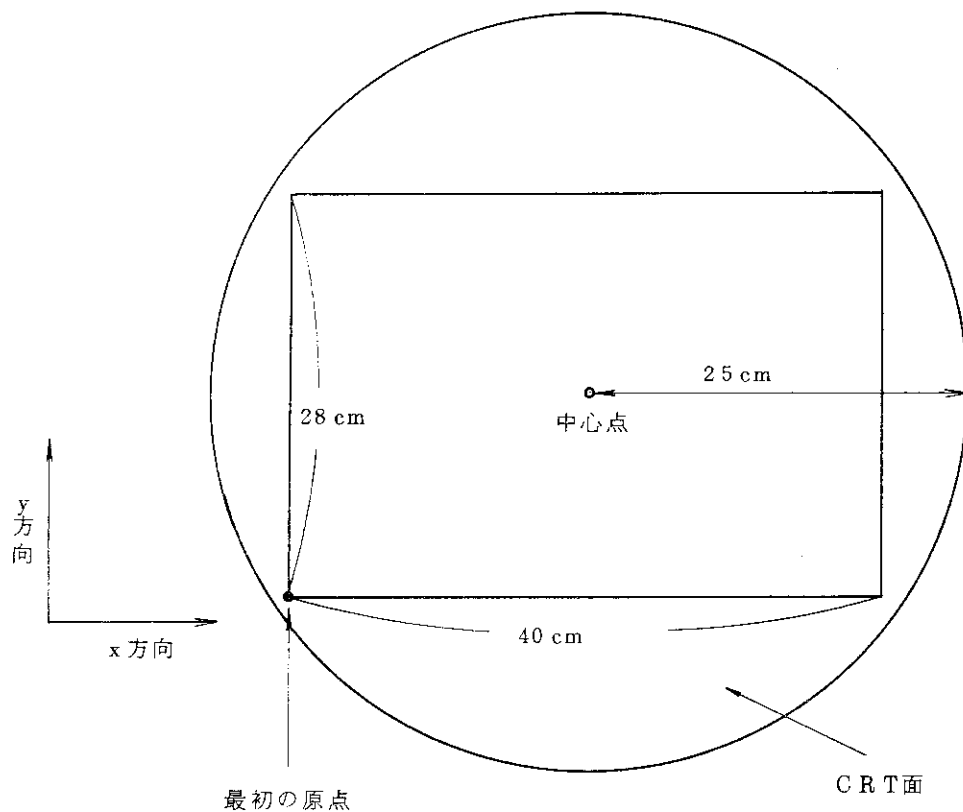


Fig. 2.1 グラフィック・ディスプレイの座標系
(簡易GSP)

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、

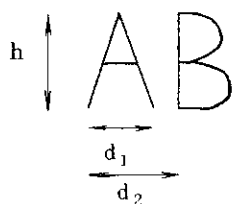
- 0mm ≤ h < 4.2mm ならば VSM,
- 4.2mm ≤ h < 6.5mm ならば SML,
- 6.5mm ≤ h < 9.3mm ならば LAG,
- 9.3mm ≤ h ならば VLG

が適用される。

Table 2.3 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 (largh) | 4.9 | 6.8 | 6.5 |
| VLG (very large) | 7.0 | 9.8 | 9.3 |

(単位はmm)



3. 簡易GSPの機能

3.1 GDS定義命令

グラフィック・ディスプレイのオープン命令の呼出しが行われた後では、グラフィック・プログラミングは作成されたディスプレイ語列が格納されるGDSの定義から始まる。このGDSの定義はGDASET命令によって行われる。

なお、グラフィック・ディスプレイのオープンとクローズは、それぞれPGSのGPLOTS (またはPLOTS) 命令とGPLOT (またはPLOT) 命令によって行いことができる。

3.1.1 GDASET命令 (Define a Graphic Data Set)

(1) 機能

GDSを定義し、かつその領域の初期化を行う。

(2) 呼出し形式

```
CALL GDASET ( gds , n )
```

(3) 引数の説明

gds : ディスプレイ語列が格納される領域としてのGDSを実数型の配列名で指定する。

n : 領域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

(i) GDSの定義を行う前に、グラフィック・ディスプレイのオープンがPGSのGPLOTS命令によって行われねばならない。

(ii) gdsの大きさとしてのnの値は、つぎの式で計算される値よりも大きくなければならない。

$$13 \text{ 語} + (N+M) \times 4 \text{ 語} + [(5 \times N + 3 \times M + D) \times \frac{4}{9}] \text{ 語}$$

ここで、Nは名前付き図形単位の数、Mは名前無し図形単位の数、およびDはディスプレイ語作成命令によって作られるディスプレイ語数(オーダー数)の総和である(4語でディスプレイ語は9オーダー作られる)。ディスプレイ語作成命令によって作られるディスプレイ語数は、点、またはベクトルの場合はその長さに応じて1~3オーダーであり、文字の場合は1文字当たり1オーダーである。

(iii) gdsはDIMENSION文によって配列として前もって宣言されていなければならない(その大きさはn以上)。

(5) 参考事項

(i) GDSが1つですむ場合は、GDSを作る必要はなく、PGSの命令GPLOTSと拡張命令GPLOTによってGDSは内部的に作成され、処理される。

3.2 ディスプレイ語作成命令

点, ベクトル, または文字の合成による図形を CRT 面に描くためには, それらをディスプレイ語列として GDS 内に作成しなければならない。定義された GDS のどれを使用するかの指定は GASSDS 命令によって, また, 図形単位に対する名前付けは GENAME 命令によって行われる。ビームのセット, および点とベクトルの表示に関するディスプレイ語列は PGS の GPLOT 命令によって作成されるが, 文字の表示に関するディスプレイ語列は GWRITE 命令によって作成される。

なお, 図形表示に関する PGS の命令や汎用 GSP⁽³⁾ の命令を用いて, 図形表示のためのディスプレイ語列を作成し, それを GDS に格納することができる。

3.2.1 GASSDS 命令 (Assign the Graphic Data Set)

(1) 機能

使用する GDS を指定し, かつその領域の初期化を行う。

(2) 呼出し形式

```
CALL GASSDS ( gds )
```

(3) 引数の説明

gds : 使用する GDS を実数型の配列名で指定する。

(4) 注意事項

- (i) 指定される gds は, GDASET 命令によって前もって定義されていなければならない。
- (ii) この命令は同じ gds で何回でも呼出すことができる。しかし, 呼出しのたびに gds は初期化されるので, gds 内のディスプレイ語列は後述される GDSPLY 命令によって, 前もってバッファ・メモリへ転送しておく必要がある。

(5) 参考事項

- (i) この命令以後に作成されるディスプレイ語列は, この命令で指定された gds に格納される。

3.2.2 GENAME 命令 (Specify an Element Name)

(1) 機能

この命令以後に作成される図形単位に名前を付ける。

(2) 呼出し形式

```
CALL GENAME ( name 1, name 2 )
```

(3) 引数の説明

name 1, name 2 : 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

- (i) ライトペンでピックされる図形単位には, この命令によって名前を付けなければならない。

(ii) (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。

(5) 参考事項

(i) name 1 = name 2 = 0 のときは, この命令以後に作成される図形単位は名前無しである。

(ii) この命令が呼出されないときは, 図形単位は名前無しで作られる。

3.2.3 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 FORMAT 文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 GDS に格納する。

(2) 呼出し形式

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

(3) 引数の説明

- xcoor, ycoor : 表示される文字列の出発点 (左下隅) の座標を実数型の定数, または変数で指定する。
- height : 文字列の高さを実数型の定数, または変数で指定する。
- data : 表示するデータを定数, または変数 (配列名も可能) で指定する。
- angle : 文字列の x 座標軸となす角度を定数, または変数で指定する。
- fmtno : データを表示するときの FORMAT 文をその文番号で指定する。

(4) 注意事項

(i) 座標 (xcoor, ycoor) は, Fig. 2.1 の座標系の下での値が mm 単位で指定されなければならない。

(ii) 文字列の高さ height は mm 単位か, または VSM, SML, LAG, VLG のいずれか 1 つで指定されなければならない。

(iii) 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。

(iv) FORMAT 文の文番号は %n (FORTRAN-C/D), または &n (FORTRAN-H) で表わされなければならない。

(v) 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) 参考事項

(i) この命令は, PGS の GSYMBL (または SYMBOL) 命令 (文字列の表示) と GNUMBR (または NUMBER) 命令 (数字の表示) の拡張となっている。

3.3 ディスプレイ制御命令

ディスプレイ語作成命令によって GDS 内に作成されたディスプレイ語列は, GDSPLY 命

令によってグラフィック・ディスプレイのバッファ・メモリへ転送される。ディスプレイ語列のバッファ・メモリへの転送によってのみCRT面への図形表示は行われる。つぎに、CRT面に表示された図形単位は、GEMOVE命令によって任意の位置に平行移動される。また、図形単位についてその表示の特性、すなわち、輝度、および線種などは、GALTER命令によって変更される。さらに、CRT面の図形の消去はGERASE命令によって行われる。

3.3.1 GDSPLY命令 (Display the Elements)

(1) 機能

指定されたGDS内に作成されたディスプレイ語列をディスプレイ・バッファへ転送し、CRT面に図形を表示する。

(2) 呼出し形式

```
CALL GDSPLY(gds)
```

(3) 引数の説明

gds : ディスプレイ・バッファへ転送するGDSを実数型の配列名で指定する。

(4) 注意事項

(i) この命令の引数として与えられたgdsは、GDASET命令によって定義され、GASSDS命令によって指定されたものでなければならない。

(ii) 与えられたgdsは、その領域が一杯になる前にこの命令によりディスプレイ・バッファへ転送されなければならない。

3.3.2 GEMOVE命令 (Move the Element)

(1) 機能

CRT面に表示されている図形単位を指定された位置に平行移動する。

(2) 呼出し形式

```
CALL GEMOVE(name1, name2, xcoord, ycoord)
```

(3) 引数の説明

name1, name2 : CRT面に表示されている図形単位の名前を整数型の定数、または変数で指定する。

xcoord, ycoord : 移動後の図形単位の始点の座標を実数型の定数、または変数で指定する。

(4) 注意事項

(i) name1 ≠ 0 でなければならない。

(ii) 座標 (xcoord, ycoord) は、Fig.2.1の座標系の下での値がmm単位で指定されなければならない。

3.3.3 GALTER 命令 (Alter the Set Control Register)

(1) 機能

CRT面に表示されている図形単位の表示特性を指定されたものに変更する。

(2) 呼出し形式

CALL GALTER (name1, name2, condit)

(3) 引数の説明

name1, name2 : CRT面に表示されている図形単位の名前を整数型の定数, または変数で指定する。

condit : 表示の特性をつぎのようなものから1つ選んで文字定数, または変数で指定する。

$$\left\{ \begin{array}{l} \text{BLK} \\ \text{DAK} \\ \text{BRT} \\ \text{VBR} \end{array} \right\}, \left\{ \begin{array}{l} \text{WNK} \\ \text{STD} \end{array} \right\}, \left\{ \begin{array}{l} \text{SLD} \\ \text{DOT} \\ \text{DSH} \end{array} \right\}, \left\{ \begin{array}{l} \text{ENA} \\ \text{DIS} \end{array} \right\} \text{ または } \left\{ \begin{array}{l} \text{VIS} \\ \text{INV} \end{array} \right\}$$

ここで, それぞれの文字定数の意味はつぎの通りである。

BLK : blank

DAK : dark

BRT : bright

VBR : very bright

図形単位の表示の輝度 (明るさの度合) を意味する。

WNK : winking image

STD : steady image

表示されている図形単位を明滅させるかしないかを意味する。

SLD : solid line

DOT : dotted line

DSH : dashed line

ベクトルを実数, 点線, または破線で表示するかを意味する。

ENA : enable lightpen detect

DIS : disable lightpen detect

表示されている図形単位をライトペンで検出できるようにするかしないかを意味する。

VIS : visible

INV : invisible

ディスプレイ・バッファに転送されたディスプレ

イ語列によって、図形単位を表示させるかしないかを意味する。

(4) 注意事項

(i) name 1 ≠ 0 でなければならない。

(5) 参考事項

(i) 指定しなかった特性については変更されない。

(ii) この命令が呼出されないときの図形単位の表示特性は、BRT, STD, SLD, ENA, VIS の状態にある。

3.3.4 GERASE 命令 (Erase the Buffer or the Element)

(1) 機能

CRT面の全図形、または指定された図形単位を消去する。

(2) 呼出し形式

```
CALL GERASE (type, name 1, name 2)
```

(3) 引数の説明

type : 消去の種類をつぎのものから1つ選んで文字定数、または変数で指定する。

$$\begin{array}{c} \nabla \{ \text{BUF} \} \nabla \\ \{ \text{ELM} \} \end{array}$$

ここで、それぞれの文字定数の意味はつぎの通りである。

BUF : ディスプレイ・バッファの初期化(消去)によって、CRT面の全図形を消去することを意味する。

ELM : CRT面に表示されている1つの図形単位を消去することを意味する。

name 1, name 2 : 消去する図形単位の名前を整数型の定数、または変数で指定する。

(4) 注意事項

(i) CRT面の全図形を消去するときは、name 1, name 2 はダミー・パラメータである。

(ii) (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。

(5) 参考事項

(i) この命令と同じ機能はPGSの拡張命令GPLOTによっても行うことができる。

3.4 割込み処理命令

今までに述べてきた命令によってCRT面に図形が表示されると、利用者はその図形を見ながら計算機に対していろいろな指示を与えることができる。これらの指示はライトペン、またはファンクション・キーボードによる計算機に対する割込みとして行われる。ライトペンの場合は、図形単位のピッキングとトラッキング・シンボルによるトラッキングの両方によって割込みが引き起される。

トラッキングとファンクション・キーボードによって引き起される割込みは、マスクをかけ

て受入れなくしたり、逆にマスクをはずして受入れ可能にすることができる。これらの指定は GAMASK 命令によって行われる（ライトペンによるピッキングについては、すでに 3.3 で述べた GALTER 命令によって、割込みの検出を可能にしたり、不可能にしたりできる）。つぎに、受入れが許可された割込みはその識別記号が割込み処理に関する待行列に格納される。格納された割込みは GETIDF 命令によって発生した順に読出される。さらに、GLIST 命令と GJUMP 命令の対使用によって、読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の流れを分岐するようにできる。なお、割込みの識別記号は、つぎの Table 3.1 に示すように図形単位名と同じ形式の 2 語で構成される。

Table 3.1 割込みの種類と識別記号

| 割 込 み 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・シンボルが追従しているとき、ライトペンのスイッチをオフにしたときに起る割込み（エンド・ポジション） | 0 | 34 |
| ライトペンにトラッキング・シンボルが追従しているとき、ライトペンのスイッチがオンの間に起る割込み（カレント・ポジション） | 0 | 35 |
| 名前が (name 1, name 2) の図形単位をライトペンでピックしたときに起る割込み | name 1 (≠ 0) | name 2 |

3.4.1 GAMASK 命令 (Define Attention Mask)

(1) 機 能

ファンクション・キー (1~32)、およびトラッキング (34, 35) による割込みの中でどれを受入れ、どれを受入れないかを指定する。

(2) 呼出し形式

```
CALL GAMASK (mask, iatten)
```

(3) 引数の説明

mask : マスクの状態をつぎのものから 1 つ選んで文字定数、または変数で指定する。

- {
 - ON
 - OF
 - MON
 - MOF
 - BON
 - BOF}

ここで、それぞれの文字定数の意味はつぎの通りである。

- ON : 引数 *iatten* で与えられる割込みに対してマスクをかける (割込みを受入れない)。割込みがファンクション・キーの場合は、*make* (キーを押したときに起る割込み), *break* (キーから離れたときに起る割込み) の区別をせず両方とも指定されたことを意味する。
- OF : 引数 *iatten* で与えられる割込みに対してマスクをはずす (割込みを受入れる)。割込みがファンクション・キーの場合は、*make*, *break* の区別をせず両方とも指定されたことを意味する。
- MON : ON と同じであるが、ファンクション・キーに対しては *make* のみを意味する。
- MOF : OF と同じであるが、ファンクション・キーに対しては *make* のみを意味する。
- BON : ON と同じであるが、ファンクション・キーに対しては *break* のみを意味する。
- BOF : OF と同じであるが、ファンクション・キーに対しては *break* のみを意味する。

iatten : つぎの Table 3.2 に示す割込みの種類 (図形単位のピッキングの場合を除く) を 1 つ選んで整数型の定数, または変数で指定する。

Table 3.2 引数 *iatten* の指定法

| 割込みの種類 | <i>iatten</i> |
|---------------------|---------------|
| ファンクション・キー番号 1 | 1 |
| ファンクション・キー番号 2 | 2 |
| } | } |
| ファンクション・キー番号 3 2 | 3 2 |
| トラッキング (エンド・ポジション) | 3 4 |
| トラッキング (カレント・ポジション) | 3 5 |

(4) 参考事項

(i) ファンクション・キー (1~32) とトラッキング (34, 35) の割込みは、GPLOTS 命令によって MOF の状態に初期設定されている。

3.4.2 GETIDF 命令 (Get Attention Identifier)

(1) 機能

割込み処理の待行列に格納された識別記号を発生した順に取出す。

(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

- (i) NAME1, NAME2は定数でなく、変数でなければならない。
- (ii) この命令は割込み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。

(5) 参考事項

- (i) 取出された識別記号は割込み処理の待行列から取除かれる。

3.4.3 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

- (i) この命令の頭には必ず文番号をつけなければならない。
- (ii) 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

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

3.4.4 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た文番号に分岐する。

(2) 呼出し形式

```
CALL GJUMP (stno, ID-1, ID-2, stno1 [, stno2]……)
```

(3) 引数の説明

stno : この命令と対になるGLIST命令の文番号を指定する。
 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受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 : 分岐する文番号を指定する。

(4) 注意事項

(i) ID-1, ID-2 は定数でなく, 変数でなければならない。

(ii) 文番号 stno, stnoi は, $\%n$ (FORTRAN-C/D), または $\&n$ (FORTRAN-H) で表わされなければならない。

(5) 参考事項

(i) 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。

(ii) この命令は FACOM GSP に属する GJUMP 命令と同じである。

3.5 トラッキング処理命令

CRT 面上の位置座標はライトペンとトラッキング・シンボルによるトラッキング機能によって計算機に入力される。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の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とある間隔で計算機に割込みが入り, トラッキング・シンボルの中心座標は GRDTRK 命令で読取られる。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。

3.5.1 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) を実数型の定数, または変数で指定する。

distan : トラッキング・シンボルの割込みが入る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

(i) 座標 (xcoor, ycoor) は, Fig.2.1 の座標系の下での値が mm 単位で指定されなければならない。

(ii) 割込みの入る間隔の長さ distan は mm 単位で表わされなければならない。

(5) 参考事項

(i) 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。

(ii) 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のスイッチがオンの間ライトペンの動きに追従する。シンボルが移動するとき, 識別記号 (0, 35), ま

たは(0, 34)の割込みが発生し、識別記号は割込み処理の待行列に格納される。識別記号(0, 35)はライトペンのスイッチがオンの間発生し、(0, 34)はライトペンのスイッチがオンからオフになったときに発生する。

3.5.2 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読取る。

(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の座標が格納される変数を実数型で指定する。

(4) 注意事項

(i) ライトペンの動きにトラッキング・シンボルが追従するときに発生する割込みを GETIDF 命令で知ってから GRDTRK 命令を呼出すこと。

(ii) XCOORD, YCOORD は定数でなく、変数でなければならない。

(5) 参考事項

(i) XCOORD, YCOORD に格納される座標は、Fig. 2.1 の座標系の下での値が mm 単位で与えられる。

3.5.3 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) 参考事項

(i) この命令によって、割込み処理の待行列にある、トラッキング・シンボルに関する識別記号はすべて取除かれる。

(ii) 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。

3.6 補助入出力処理命令

文字キーボードによる各種データの入力には GREADD 命令によって行われる。また、ASW の状態の読込みは GRDA SW 命令によって行われる。さらに、ファンクション・キーのランプの点灯、または消滅は GSLAMP 命令によって行われる。

なお、PGSの拡張命令GPLOTによって、ライトペンと文字キーボードを用いた日付、利用者名、タイトル、説明文などのコメントの入力とそのCRT面への表示、およびCRT面に表示された図形のCOM、またはプロッタへのハード・コピー出力を行うことができる。

3.6.1 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入力された文字列を指定されたFORMAT文に従って、整数型、実数型、または文字型のデータに変換し、かつ指定された変数に格納する。

(2) 呼出し形式

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

(3) 引数の説明

- xcoor, ycoor : カーソルの表示位置を示す座標を実数型の定数、または変数で指定する。
- height : 表示される文字列の高さを実数型の定数、または変数で指定する。
- DATA : 変換されたデータが格納される変数(配列名も可能)を指定する。
- 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。
- fmtno : 文字列をデータに変換するときのFORMAT文をその文番号で指定する。

(4) 注意事項

- (i) 座標(xcoor, ycoor)は、Fig.2.1の座標系の下での値がmm単位で指定されなければならない。
- (ii) 文字列の高さheightはmm単位か、またはVSM, SML, LAG, VLGのいずれか1つで指定されなければならない。
- (iii) 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。
- (iv) FORMAT文の文番号はY_n(FORTRAN-C/D)、または&n(FORTRAN-H)で表わされなければならない。
- (v) 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。
- (vi) 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、以後はFORMAT文に従ってデータの文字列が全部入力されるまでこの命令は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

- (i) FORMAT文に従って文字列を入力中、入力ミスをしてENDキーを押さない限り、この命令のレベルで←, →, またはCANCELキーを使って文字列の修正が可能である。

(ii) 文字キーボードによる文字列の入力が完了すると (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FORMAT文に従ってデータに変換され, 変数DATAに格納される。

(iii) この命令の実行終了後も入力した文字列をそのままの形でずっと表示させておく場合は, GWRITE命令などを用いればよい。

3.6.2 GRDASW命令 (Read the Alteration Switch)

(1) 機能

ディスプレイ・コンソール上のASWの状態を読み込む。

(2) 呼出し形式

```
CALL GRDASW( IASW)
```

(3) 引数の説明

IASW : ASWの各スイッチのオン・オフ状態が格納される変数を整数型で指定する。

(4) 注意事項

(i) IASWは定数でなく, 変数でなければならない。

(5) 参考事項

(i) ASWの各スイッチのオン・オフ状態は, IASWの下8ビットにそれぞれ格納される。

3.6.3 GSLAMP命令 (Set a Function Key Lamp)

(1) 機能

ファンクション・キー (1~32) のランプの点灯, 消滅を行う。

(2) 呼出し形式

```
CALL GSLAMP( onoff, lamp)
```

(3) 引数の説明

onoff : ファンクション・キーのランプの状態をつぎのものから1つ選んで文字定数, または変数で指定する。

$$\nabla \left\{ \begin{array}{c} \text{ON} \\ \text{OF} \end{array} \right\} \nabla$$

ここで, それぞれの文字定数の意味はつぎの通りである。

ON : 引数 lamp で与えられるファンクション・キーのランプの点灯を意味する。

OF : 引数 lamp で与えられるファンクション・キーのランプの消滅を意味する。

lamp : 点灯, または消滅させるファンクション・キーのランプ番号 (キー番号と同じ) を整数型の定数, または変数で指定する。

(4) 注意事項

(i) ファンクション・キーの 8 番と 16 番には GPLOT 命令によってランプが点灯されている。これらのランプは消滅させてはならない。

4. 使 用 例

ここでは、プロッタ用に作成されたプログラムを基に、簡易GSPとPGSによって対話型グラフィック・プログラムを作成した具体例を示す。

4.1 例 題

Table 4.1 に示すプログラムは、二次関数

$$Y(I) = A * (X(I) - B) ** 2 + C \quad (I = 1, 2, \dots, 32)$$

を計算し、変数と関数値の組

$$(X(I), Y(I)) \quad (I = 1, 2, \dots, 32)$$

をプロッタにグラフ出力するものである。ここで、A, B, Cは、1枚の入力カードから与えられるパラメータであり、X(I) (I=1, 2, …, 32)は、

$$X(1) = -15.0, \quad X(2) = -14.0, \dots, X(31) = 15.0, \quad X(32) = 16.0$$

のように1.0きざみで-15.0から16.0まで変化する変数である。また、プログラムは異なるA, B, Cに対して何ケースも実行できるが、A=0が入力されると実行を終了する。

さて、このプログラムを基にして、つぎのような機能をもつ対話型グラフィック・プログラムを考える。

まず、1ケース、1画面としてCRT面に二次関数のグラフを表示する。つぎに、利用者はつぎのような5つの処理を適宜選択して行うことができる。

- (1) 次のケースへの移行
- (2) 再計算の実行とその結果のグラフ表示
- (3) 文字キーボードによるパラメータA, B, Cのオンライン変更
- (4) コメント入力とCRT面上の図形のハード・コピー出力(COM, またはプロッタ)
- (5) プログラムの実行の終了化

ここで、上記機能(1), (2), (3), (4), (5)の指定は、ライトペンのみならず、それぞれファンクション・キー1, 2, 3, 4, 5(ランプ付)によっても可能である。また、(2), (3), (4)については、利用者はグラフ表示の結果に満足できるまで繰り返し操作可能である。

4.2 グラフィック・プログラム

つぎのTable 4.2は、簡易GSPとPGSを用いて作成された、上記のような機能をもつグラフィック・プログラムである。また、Fig. 4.1は、このプログラムが実行されたときにCRT面に表示された図形のハード・コピーをCOMに出力し、プリントしたものである。

Table 4.1 プロッタ・プログラムのリスト

FACOM 230-75 (M7) FORTRAN-D -750715- V01-L05 76.06.01 PAGE 1

* SOURCE STATEMENT *

```

C
C * PLOTTER EXAMPLE *
1  DIMENSION X(34), Y(34), BUFF(1024)
2  CALL PLOTS ( BUFF, 1024 )
3  X(1) = -15.
4  DO 10 I = 2, 32
5  X(I) = X(I-1) + 1.
6  20 READ (5, 21) A, B, C
7  21 FORMAT (3F10.3)
8  IF ( A * B * C ) GO TO 900
9  30 DO 30 I = 1, 32
10 Y(I) = A * X(I) ** 2 + C
11 CALL SCALE ( A, 200., 32, 1., 10. )
12 CALL SCALE ( Y, 200., 32, 1., 10. )
13 CALL AXIS ( 0., 0., 'X - V A L U E', -13.200., 0., X(33), X(34), 10. )
14 CALL AXIS ( 100., 0., 'Y - V A L U E', -13.200., 90., Y(33), Y(34), 10. )
15 CALL LINE ( X, Y, 32, 1., 0., 0 )
16 CALL SYMPO ( 60., 230., 8., 'Y=A*(X-R)**2+C', 0., 14 )
17 CALL PLOT ( 300., 0., -3 )
18 GO TO 20
19 900 CALL PLOT ( 0., 0., 999 )
20 STOP
21 ENH

```

Table 4.2 グラフィック・プログラムのリスト

FACOM 220-75 (M7) FORTRAN-D -760229- V06-LOT 76.07.16 PAGE 1

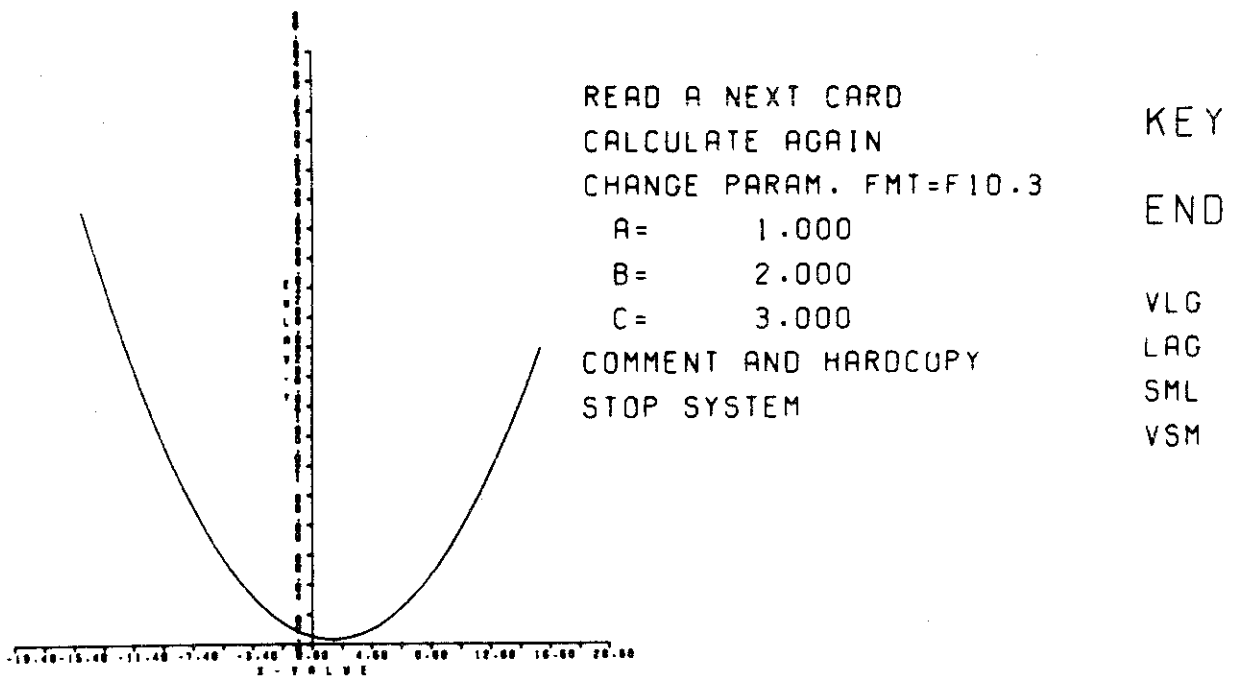
• SOURCE STATEMENT •

```

C
C   COMPUTER GRAPHICS PROGRAM
C   WRITTEN IN PGS AND SIMPLIFIED GSP
C   BY Y.NAKAMURA AND Y.ONUMA
C   COMPUTING CENTER, TOKAI, JAERI
C
1   DIMENSION X(34), Y(34), BUFF(1024)
2   DIMENSION GDS1(200)
3   DATA ISW/0/
4   CALL PLOTS ( BUFF, 1024)
5   X(1)=-15.
6   DO 10 I=2,32
7   X(I)=X(I-1)+1.
8   20 READ (5,21) A, B, C
9   21 FORMAT (3F10.3)
10  IF ( A.EQ.0 ) GO TO 900
11  25 DO 30 I=1,32
12  30 Y(I)= A*(X(I)-B)**2+C
13  CALL SCALF ( X, 200., 32, 1, 10. )
14  CALL SCALF ( Y, 200., 32, 1, 10. )
15  CALL AXIS ( 0., 0., X = V A L U E', -13, 200., 0., X(33), X(34), 10. )
16  CALL AXIS ( 100., 0., Y = V A L U E', 13, 200., 90., Y(33), Y(34), 10. )
17  CALL LINE ( X, Y, 32, 1, 0, 0)
18  CALL SYMBOL ( 60., 230., 8., 'Y=A*(X-B)**2+C', 0., 14 )
19  CALL PLOT ( 300., 0., -3 )
C
20  CALL PLOT ( 0., 0., 777 )
21  CALL PLOT ( 0., 0., 888 )
C
C   DISPLAY LIGHT BUTTONS AND LAMP FUNCTION KEYS(1-5)
C
22  IF (ISW.NE.0) GO TO 40
23  CALL GDASFT(GDS1,200)
24  CALL GASSDS(GDS1)
25  CALL SYMBOL ('EL', 'RN')
26  CALL SYMBOL ( 220., 180., 6.5, 'READ A NEXT CARD', 0., 16)
27  CALL SYMBOL ('EL', 'CA')
28  CALL SYMBOL ( 220., 165., 6.5, 'CALCULATE AGAIN', 0., 15)
29  CALL SYMBOL ('EL', 'CH')
30  CALL SYMBOL ( 220., 150., 6.5, 'CHANGE PARAM: FMT=F10.3', 0., 23)
31  CALL SYMBOL(220,0,90,0,6,5,'COMMENT AND HARDCOPY',0,0,20)
32  CALL SYMBOL('EL', 'SS')
33  CALL SYMBOL(220,0,75,0,6,5,'STOP SYSTEM',0,0,11)
C
34  DO 37 I=1,5
35  37 CALL GSLAMP('ON', I)
36  ISW=1
37  40 CALL GDSPLY(GDS1)
C
C   DISPLAY PARAMETERS
C
38  50 CALL PLOT(0,0,0,0,333)
39  CALL PLOT ( 'EL', 1, 333)
40  CALL GWRITE ( 230., 135., 'LAG', A, 0., #51 )
41  51 FORMAT ( 'A', F10.3)
42  CALL PLOT ( 'EL', 2, 333 )
43  CALL GWRITE ( 230., 120., 'LAG', B, 0., #52 )
44  52 FORMAT ( 'B', F10.3)
45  CALL PLOT ( 'EL', 3, 333 )
46  CALL GWRITE ( 230., 105., 'LAG', C, 0., #53 )
47  53 FORMAT ( 'C', F10.3)
48  CALL PLOT ( 0., 0., 444 )
C
C   SELECT LIGHT BUTTONS OR FUNCTION KEYS
C
49  60 CALL GETINF ( 11, 12 )
50  CALL GJUMP ( #61, 11, 12, #70, #80, #99, #90, #900,
51  1 #70, #80, #99, #90, #900)
52  GO TO 60
53  61 CALL GLIST ( 'EL', 'RN', 'EL', 'CA', 'EL', 'CH', 'EL', 'CO',
54  1 'EL', 'SS', 0, 1, 0, 2, 0, 3, 0, 4, 0, 5)
55  70 CALL PLOT ( 0., 0., 666 )
56  GO TO 20
57  80 CALL PLOT ( 0., 0., 666 )
58  GO TO 25
59  90 CALL GALTER('EL', 'CO', 'VBR')
60  CALL PLOT(0,0,0,0,777)
61  CALL PLOT(0,0,0,0,888)
62  CALL GALTER('EL', 'CO', 'BRT')
63  GO TO 60
C
C   CHANGE PARAMETERS
C
64  99 CALL GALTER('EL', 'CH', 'WNK')
65  100 CALL GETINF ( 11, 12 )
66  CALL GJUMP ( #101, 11, 12, #110, #120, #130)
67  101 CALL GLIST ('EL', 1, 'EL', 2, 'EL', 3)
68  GO TO 100
69  110 CALL GREADD ( 243., 128., 'LAG', A, 0., #111 )
70  GO TO 150
71  111 FORMAT (F10.3)
72  120 CALL GREADD ( 243., 113., 'LAG', B, 0., #111 )
73  GO TO 150
74  130 CALL GREADD ( 243., 98., 'LAG', C, 0., #111 )
75  150 CALL GALTER('EL', 'CH', 'STD')
76  DO 160 I=1,3
77  160 CALL PLOT ( 'EL', 1, 666 )
78  GO TO 50
79  900 CALL PLOT ( 0., 0., 999 )
80  STOP
81  END

```

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



HARDCOPY NOCOPY

Fig. 4.1 グラフィック・プログラムによってCRT面に表示された図形のCOMへのハード・コピー

Table 4.2のプログラムについて各ステートメントの意味をその左側のISN (Internal Statement Number) の順に従って説明するとつぎのようになる。

- 1~2 : 変数Xと関数Y, およびグラフィック・データ・セットGDS 1 などに対する配列の宣言。
- 3 : 分岐用のスイッチに使用される変数ISWの初期設定。
- 4 : グラフィック・ディスプレイをオープンし, かつPGSの内部にグラフィック・データ・セットGDSを作成する。
- 5~7 : Xの値の設定。
- 8~9 : (¥20) 入力カードからパラメータA, B, Cを読み込む。
- 10 : Aが0ならばISN77 (¥900) へ分岐する。
- 11~12 : (¥25) A, B, C, およびXの値からYの値を計算する。
- 13~14 : X, およびYに対するスケーリングを行う。
- 15~16 : スケーリングの結果を基にX, およびY座標軸を描くためのディスプレイ語列を作成し, これをGDSに格納する。
- 17 : 各X(I), Y(I)の値からスケーリングに基づく座標値を求め, つぎに各座標点を直線で結ぶためのディスプレイ語列を作成し, これをGDSに格納する。
- 18 : 標題 $\nabla Y=A*(X-B)**2+C \nabla$ を描くためのディスプレイ語列を作成し, これをGDSに格納する。

- 19 : GDS内に作成されたディスプレイ語列をディスプレイ・バッファへ転送し、CRT面に二次関数のグラフを表示する。つぎに、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にする。
- 20 : CRT面上の図形のCOM, またはプロッタへのハード・コピー出力を可能にする。
- 21 : 最初の1回のみISN22~36の命令を実行する。2回目以降はISN37(¥40)へ分岐する。
- 22 : グラフィック・データ・セットGDS1を定義し、その領域を初期化する。
- 23 : GDS1を使用することを指定する。
- 24 : ISN25で作成される図形単位に名前(EL, RN)を付ける。
- 25 : 図形単位▽READ A NEXT CARD▽を描くためのディスプレイ語列を作成し、これをGDS1に格納する。
- 26~33 : ISN24, 25のように順次つぎのような名前付き図形単位のディスプレイ語列を作成し、これをGDS1に格納する。
- | | 名 前 | 図 形 単 位 |
|-------|----------|---------------------------|
| 26~27 | (EL, CA) | ▽CALCULATE AGAIN▽ |
| 28~29 | (EL, CH) | ▽CHANGE PARAM. FMT=F10.3▽ |
| 30~31 | (EL, CO) | ▽COMMENT AND HARDCOPY▽ |
| 32~33 | (EL, SS) | ▽STOP SYSTEM▽ |
- 34~36 : ファンクション・キー番号1, 2, 3, 4, 5のランプを点灯する。つぎにスイッチ用変数ISWに1をセットする。
- 37 : (¥40) ISN22~33でGDS1内に作成されたディスプレイ語列をディスプレイ・バッファへ転送し、CRT面に上記のような図形単位を表示する。
- 38 : (¥50) PGS用のGDSを初期化する。
- 39 : ISN40で作成される図形単位に名前(EL, 1)を付ける。
- 40~41 : 図形単位▽A=Aの値▽を描くためのディスプレイ語列を作成し、これをGDSに格納する。
- 42~47 : ISN39~41のように順次つぎのような名前付き図形単位のディスプレイ語列を作成し、これをGDSに格納する。
- | | 名 前 | 図形単位 |
|-------|---------|---------|
| 42~44 | (EL, 2) | ▽B=Bの値▽ |
| 45~47 | (EL, 3) | ▽C=Cの値▽ |
- 48 : ISN38~47でGDS内に作成されたディスプレイ語列をディスプレイ・バッファへ転送し、CRT面に上記のような図形単位を表示する。
- 49~52 : (¥60) 名前がそれぞれ(EL, RN), (EL, CA), (EL, CH), (EL, CO), (EL, SS)である5つの図形単位の内1つがライトペンでピックされるか、またはファンクション・キー番号1, 2, 3, 4, 5の内1つが押されるまでプログラムは待ち状態となる。ライトペン, またはファンクション

- ・キーで指定されると、指定された図形単位、またはファンクション・キーに対応して、以後プログラムは文番号¥70, ¥80, ¥99, ¥90, ¥900の内1つに分岐する。上記のもの以外が指定されると文番号¥60に分岐する。
- 53~54 : (¥70) CRT面上の図形を消去して ISN8 (¥20) へ分岐する。したがってパラメータの再入力によるつぎのケースを実行することになる。
- 55~56 : (¥80) CRT面上の図形を消去して ISN11 (¥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。
- 57~61 : (¥90) 名前が (EL, CO) である図形単位の輝度を VBR にセットする。つぎに、ライトペンによる文字キーボードからのコメント入力と CRT 面上の図形の COM, またはプロッタへのハード・コピー出力を可能にする。さらに、上記の図形単位の輝度を BRT にリセットし、ISN49 (¥60) へ分岐する。
- 62 : (¥99) 名前が (EL, CH) である図形単位を点滅させる。
- 63~66 : (¥100) 名前がそれぞれ (EL, 1), (EL, 2), (EL, 3) である3つの図形単位の内1つがライトペンでピックされるまでプログラムは待ち状態となる。ライトペンでピックされるとピックされた図形単位に対応して以後プログラムは文番号¥110, ¥120, ¥130の内1つに分岐する。
- 67~69 : (¥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 ISN73 (¥150) へ分岐する。
- 70~71 : (¥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 ISN73 (¥150) へ分岐する。
- 72 : (¥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。
- 73 : (¥150) 名前が (EL, CH) である図形単位の点滅を中止させる。
- 74~76 : 名前がそれぞれ (EL, 1), (EL, 2), (EL, 3) である3つの図形単位を消去して、ISN38 (¥50) へ分岐する。
- 77~79 : (¥900) グラフィック・ディスプレイをクローズして、プログラムの実行を終了する。

5. 使用上の注意

5.1 グラフィック・プログラムの実行

今までに述べてきた簡易GSPの命令によって、本格的な対話型グラフィック・プログラムの作成が可能となった。作成したグラフィック・プログラムをFACOM230-75によって実行する場合は、JAERI-M 6619⁽⁴⁾を参照するとよい。そこでは、PGSLIB/BASICによるグラフィック・プログラムの作成の他に、グラフィック用ジョブ制御文の作成から、グラフィック・ジョブの実行までの具体的な方法について述べられている。

つぎのTable 5.1は、Table 4.2のプログラムをFORTRAN-C/Dで実行したときのジョブ制御文のリストである。

5.2 エラー・メッセージ

グラフィック・プログラムの実行時にCRT面やプリンタに出力されるエラー・メッセージについては、JAERI-M 6619にPGSLIB/BASICの場合に限って記述されている。

そこで、ここではつぎのTable 5.2に簡易GSPの命令が実行されるときに出力されるエラー・メッセージのエラー・コード一覧表⁽⁶⁾を示す。

Table 5.1 グラフィック用ジョブ制御文のリスト

```

FACOM 230-60/75  MONITOR6/7  SYSTEM=V04/L17  BATCH=V04/L01  JOP CONTROL LANGUAGE LIST 76.07.16  PAGE 1
.....1.....2.....3.....4.....5.....6.....7.....8
1  *NO  1803*
                                     /
                                     #.O/PAGE 40
                                     P.O/PCH  0
                                     T.O/TIME 10S
                                     *
                                     GDP/
                                     C35/

2 VM *GJOB  1411803*NAKAMURA*Y*341.02
                                     /GRAPHICS
3 VM *DFORT
4 VM *DLIEDRJM GRFO=ON*PGSLIB=CALL*GDSP=ON*GDBG=ON
5 VM *GCOMB5
6 VM *DISK F08
7  *JEND
    
```

Table 5.2 エラー・コード一覧表

| エラー・コード | エラーの内容 | 命令 |
|---------|--|---|
| 2 | 未定義のパラメータが指定された。 | GDASET, GASSDS, GENAME, GPLOT (拡張命令も含む), GWRITE, GALTER, GERASE, GAMASK, GREADD, GRDASW, GSLAMP |
| 5 | リストと FORMAT文の内容が一致しない。 | GWRITE, GREADD |
| 6 | 許されない FORMAT文が指定された。 | GWRITE, GREADD |
| 11 | 指定された図形単位名について, 第1語目が0である。 | GENAME, 拡張命令 GPLOT |
| 12 | 指定された割込みに誤りがある。 | GAMASK |
| 13 | 指定されたファンクション・キーのランプ番号に誤りがある。 | GSLAMP |
| 15 | GLISTのパラメータがない。 | GJUMP |
| 23 | グラフィック・ディスプレイがオープンされていない。 | GDSPLY, GEMOVE, GALTER, GERASE, GAMASK, GETIDF, GBGTRK, GRDTRK, GENTRK, GREADD, GRDASW, GSLAMP, 拡張命令 GPLOT |
| 24 | グラフィック・ディスプレイ装置にないオプションが指定された。 | GDSPLY, GAMASK, GREADD, GRDASW, GSLAMP, 拡張命令 GPLOT |
| 25 | ディスプレイ・バッファがオーバ・フローした。 | GDSPLY, 拡張命令 GPLOT |
| 26 | トラッキング・シンボル, またはカーソルが CRT面の外に出た。 | GBGTRK, GREADD |
| 27 | 文字キーボードから入力された文字が80個を越えた。 | GREADD, 拡張命令 GPLOT |
| 29 | 許されない文字が指定された。 (INVALID CHARACTER) | GWRITE, GREADD, 拡張命令 GPLOT |
| 31 | 未定義のGDSが指定された。 | GASSDS, GDSPLY |
| 36 | 指定されたGDSに図形単位がない。 | GDSPLY |
| 41 | GDSなどのテーブルがオーバ・フローした。 | GPLOTS, GPLOT (拡張命令も含む), GDASET, GASSDS, GENAME, GWRITE, GDSPLY |
| 43 | 指定された名前の図形単位がない。 | GEMOVE, GALTER, GERASE, 拡張命令 GPLOT |
| 48 | リストの識別記号の指定が正しくない。 | GJUMP |
| 51 | GASSDSが呼ばれていない。 | GPLOT, GWRITE |
| 61 | 指定された座標値の絶対値が非常に大きいため, 命令実行に伴う座標計算中に演算オーバ・フローが起きた。 | GPLOT, GWRITE, GREADD |
| 63 | GSPモニタ, またはGSP命令自体のシステム・エラー | |

6. お わ り に

簡易GSPは、PGSに合わせてFACOM GSP を単純化することによって開発されたグラフィック・プログラミング用ソフトウェアである。

PGSはプロッタ・プログラム(またはCOMプログラム)をそのまま図形表示用のグラフィック・プログラムとして利用可能にするが、本格的な対話型グラフィック・プログラムの作成には不十分である。

一方、FACOM GSP は広範囲な対話型グラフィック処理のプログラミングに対して適用可能であるが、必要とする命令数が多くプログラミングが複雑になりがちである。

これらに対して、簡易GSPはプロッタ・プログラムを利用した図形表示プログラムに、本格的な対話型グラフィック処理のプログラムの追加を容易に可能にさせる。

かくして、簡易GSPによって、PGSやFACOM GSP のそれぞれの欠点を補い、長所を生かす形でグラフィック・プログラムの作成を行うことができる。

簡易GSPは、広く対話型グラフィック処理のプログラミングに対して適用可能であるが、特に、グラフ図形を中心とした対話型グラフィック処理のプログラミングに有効である。

参 考 文 献

- (1) 富士通(株) : FACOMディスプレイ装置Ⅱ(2)グラフィック・ディスプレイ装置
FACOM6233A, 1972
- (2) 富士通(株) : FACOM GSP 文法編, 1973
- (3) 中村康弘, 小沼吉男 : 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, JAERI-M 6023, 1975
- (4) 中村康弘, 小沼吉男 : グラフィック・ディスプレイのプログラミング, JAERI-M 6619, 1976
- (5) 日本IBM(株) : 2250映像表示装置/GSP, 1971
- (6) 富士通(株) : FACOM230 M-VI/VII GSP602/GSP603 使用手引書, 1974

6. お わ り に

簡易GSPは、PGSに合わせてFACOM GSP を単純化することによって開発されたグラフィック・プログラミング用ソフトウェアである。

PGSはプロッタ・プログラム(またはCOMプログラム)をそのまま図形表示用のグラフィック・プログラムとして利用可能にするが、本格的な対話型グラフィック・プログラムの作成には不十分である。

一方、FACOM GSP は広範囲な対話型グラフィック処理のプログラミングに対して適用可能であるが、必要とする命令数が多くプログラミングが複雑になりがちである。

これらに対して、簡易GSPはプロッタ・プログラムを利用した図形表示プログラムに、本格的な対話型グラフィック処理のプログラムの追加を容易に可能にさせる。

かくして、簡易GSPによって、PGSやFACOM GSP のそれぞれの欠点を補い、長所を生かす形でグラフィック・プログラムの作成を行うことができる。

簡易GSPは、広く対話型グラフィック処理のプログラミングに対して適用可能であるが、特に、グラフ図形を中心とした対話型グラフィック処理のプログラミングに有効である。

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
FACOM6233A, 1972
- (2) 富士通(株) : FACOM GSP 文法編, 1973
- (3) 中村康弘, 小沼吉男 : 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, JAERI-M 6023, 1975
- (4) 中村康弘, 小沼吉男 : グラフィック・ディスプレイのプログラミング, JAERI-M 6619, 1976
- (5) 日本IBM(株) : 2250映像表示装置/GSP, 1971
- (6) 富士通(株) : FACOM230 M-VI/VII GSP602/GSP603 使用手引書, 1974