

JAERI-M

6391

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・デバッキング

1976年2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.

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・デバッキング

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1月13日受理)

グラフィック・サブルーチン・ライブラリPGSLIBは、グラフィック・プログラミング用の道具として各種の対話型グラフィック・システムの開発に利用されている。このPGSLIBはプロッタ・プログラムをグラフィック・ディスプレイによってCRT面上でシミュレーション可能であり、グラフィック・ディスプレイは高速度で図形をCRT面上に表示できる。このPGSLIBとグラフィック・ディスプレイを用いることによって、プロッタ・プログラムのデバッキングはオンラインで短時間に実現できる。

JAERI - M 6391

On-line Debugging of the Plotter Program by a
Graphic Display

Yasuhiro NAKAMURA and Yoshio ONUMA

Computing Center, Tokai, JAERI

(Received January 13, 1976)

The graphic subroutine library PGSLIB is a graphic programming tool for the development of interactive graphic systems. It makes possible the simulation of a plotter program on the CRT by a graphic display, which displays graphic figures on the CRT at high speed. With the PGSLIB and graphic display, the plotter program can be debugged on-line in a short time.

目 次 な し

1 はじめに

グラフィック・プロッタ⁽¹⁾ (以下、簡単にプロッタ)は電子計算機の図形出力装置として広く利用されている。このプロッタを使用するためには、利用者はプロッタ・プログラム⁽²⁾を書き、デバッキングを行って、プログラムを完成させなければならない。

ところが、プロッタ・プログラムのデバッキングには何日も時間がかかるのが普通である。プロッタ・プログラムではプリンタによる印刷出力の他にプロッタによる図形出力があるが、この図形出力は印刷出力に比べて非常に多くの時間がかかり、しかもデバッキングには欠かせない。

このデバッキングに時間がかかる問題は、図形入出力装置としてのグラフィック・ディスプレイ^{(3), (4)}と、筆者らが先に開発したグラフィック・プログラミング用ソフトウェアPGSLIB (Plotter-compatible Graphic Subroutine Library)を用いて解決することができる。

実際、グラフィック・ディスプレイは図形を高速度でCRT (Cathode Ray Tube) 面上に表示でき、また、PGSLIBはプロッタ・プログラムをグラフィック・ディスプレイでシミュレーションさせることができる。したがって、利用者はプロッタで描かれるのと同じ図形をすぐCRT面上で知見でき、それゆえ、プロッタ・プログラムのデバッキングがオンラインで短時間に実現できる。

このようなグラフィック・ディスプレイ⁽⁵⁾を用いたオンライン・デバッキングについては、PGSLIB/BASICのマニュアルから自然と理解できるようになっている。しかし、そこではグラフィック・プログラミングが主たるテーマとなっているため、プロッタ・プログラムのデバッキングのみを行いたい利用者には少し詳し過ぎるきらいがあり、プロッタの利用者からはもっと簡単なマニュアルがほしいと切望されていた。

そこで、以下ではまず、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・デバッキングについてその概要を述べ、つぎに、PGSLIBによるプロッタ・プログラムのグラフィック・ディスプレイでのシミュレーションの方法、PGSLIBとグラフィック・ディスプレイの使用上の注意、およびデバッキング例などについて述べる。

2 オンライン・デバッキング

グラフィック・ディスプレイのプログラミング用ソフトウェアとして開発されたPGSLIBは、多くの対話型グラフィック・システム⁽⁶⁾の開発に利用されている。このPGSLIBはプロッタ・プログラムをグラフィック・ディスプレイによってCRT面上でシミュレーションできる特徴があり、一方グラフィック・ディスプレイはプロッタとは比較にならない程の高速度で図形をCRT面上に表示できる。

そこで、PGSLIBによってプロッタ・プログラムをグラフィック・ディスプレイで実行すれば、直ちにプロッタに出力される場合と同じ図形がCRT面上に表示される。ここで、利用

者はCRT面に表示された図形を直接目で確かめることによって、その図形が正しく描かれているか否かがわかり、したがって、またプロッタ・プログラムのエラーも見つけることができる。もし、プログラムのエラーが見つかったら、プログラムを修正し、コンパイル段階から再実行すればよい。

(7)
FACOM 230-75によるグラフィック・プログラムの実行では、デバッキング・モードと呼ばれる実行モードがある。このモードを指定すると、もしプログラムにエラーがある場合にはそのエラー・メッセージがCRT面上に表示される。さらに、そのエラーを無視してプログラムを先に進行させるか、あるいはプログラムを途中で強制終了させるかの判断と指示を利用者が行うことができる。

このようなオンライン・デバッキングの過程でCRT面に表示された図形はいつでも必要に応じてハードウェアのハード・コピー装置で記録しておくことができる。また、もっと鮮明なハード・コピーが必要であれば、CRT面の図形をPGSLIBを用いたプログラム・コントロー(5)ルによってプロッタ上へ出力することもできる。いずれの方法にしる、CRT面に表示された図形のハード・コピーがとれていれば、計算機による実行終了後でもハード・コピーとプリンタによる出力結果を基にしてプログラムのデバッキングを机上で有効に行うことができる。

このグラフィック・ディスプレイを用いたオンライン・デバッキングでは、グラフィック・ディスプレイにおける図形表示の高速性から、プロッタ・プログラムのデバッキングに要する時間は、プロッタのみによる場合に比べ大巾に短縮される。したがって、PGSLIBとグラフィック・ディスプレイを用いることによって、プロッタ・プログラムのデバッキングを短時間に集中して効率よく行うことができる。

3 PGSLIBによるシミュレーション

PGSLIBの命令は、PGS (Plotter-compatible Graphic Subroutine Package) 簡易GSP (Simplified Graphic Subroutine Package), および汎用GSP (General Graphic Subroutine Package)の3つのパッケージに大別される。(5)

PGSはプロッタ・プログラミングと同じ方法で図形表示のプログラミングを可能にし、簡易GSPはライトペンとキーボードを用いた対話処理のプログラミングを可能にする。また、汎用GSPは各種のグラフィック応用命令からなり、おもにPGSと簡易GSPによって書かれている。

これらの命令の中で、PGSの命令の大部分と汎用GSPの命令の約半分が、プロッタ・プログラムのグラフィック・ディスプレイによるシミュレーションを可能にしている。そしてPGSの命令の大部分はプロッタの基本ルーチンのシミュレーションに関係し、汎用GSPの命令の約半分はプロッタの応用ルーチンのシミュレーションに関係している。

PGSLIB/BASICのマニュアルでは、これらの命令についてグラフィック・プログラミングという観点から説明されているが、ここではプロッタ・プログラムがどのようにしてグラフィック・ディスプレイによってシミュレーションされるかという観点から述べることにする。

3.1 PGSによるシミュレーション

3.1.1 PGS 命令

プロッタ・プログラムは通常Table 3.1に示すような基本ルーチンを用いて書かれている。この基本ルーチンの機能は同じくTable 3.1に示されているPGS 命令によりグラフィック・ディスプレイでシミュレーションされる。

Table 3.1 基本ルーチンとPGS 命令

基本ルーチン	機 能	PGS 命令	接 続 命 令
PLOTS	プロッタをオープンする	GPLOTS	PLOTS
PLOT	直線を描く	GPLOT	PLOT
SYMBOL	文字またはプロットング・ポイント用記号を描く	GSYMBL	SYMBOL
NUMBER	数字を描く	GNUMBR	NUMBER
SCALE	データをスケールリングする	GSCALE	SCALE
AXIS	座標軸を描く	GAXIS	AXIS
LINE	データのグラフを描く	GLINE	LINE
FACTOR	図形の拡大・縮小に関する尺度因子を与える	GFACTR	FACTOR
WHERE	ペンの位置および尺度因子を得る	GWHERE	WHERE
NEWPEN	ペンの色を指定する	GNEWPN	NEWPEN

注： GNEWPNは何も実行しない)

これらの基本ルーチンとPGS 命令の間では、それぞれ対応するものの呼出し名は異っている。しかしながら、対応するものの引数は同じであり、さらに後述されるPGSの接続命令と拡張命令によって、プロッタ・プログラムは、2, 3のステートメントが追加されるだけでグラフィック・プログラムとして実行され、シミュレーションされる。

このようにプロッタ・プログラムはPGSによってグラフィック・ディスプレイでシミュレーションされるが、プロッタとグラフィック・ディスプレイとのちがいから、それぞれにおける座標系の取り方、点、直線、文字の描き方はつぎのようになる。

(1) 座 標 系

プロッタではFig. 3.1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を描くことができる。しかし、描かれた図形を見やすくするために、通常は1つの図形をプリンタ1ページ(縦27.9cm×横38.1cm)程度の大きさに区切って描くことが多い。座標の原点は最初にオペレータによってプロッタ・シート上で左下端にセットされるが、それ以後はプロッタの基本ルーチンの使用により利用者の責任で原点の変更ができる。また、プロッタ・プログラムでは座標の値や長さはmm単位で表わされる。

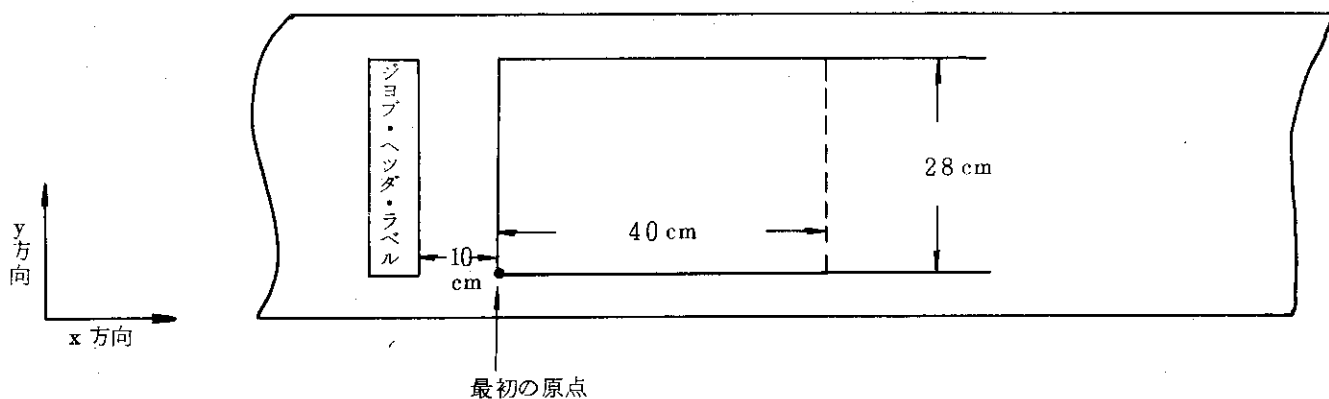


Fig. 3.1 プロッタの座標系

これに対して、グラフィック・ディスプレイではFig. 3.2 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のように縦 30cm、横 40cm の矩形が CRT 面内に含まれる。したがって、プロッタでプリンタ 1 ページ程度の大きさに描かれた図形は、そのままの大きさに CRT 面上に表示できる。PGS により原点はプロッタと同じく最初に自動的に CRT 面の左下端にセットされるが、それ以後はプロッタでの原点変更に伴ない、原点変更が行われる。また、PGS は座標の値や長さをプロッタの基本ルーチンと同じく mm 単位で受けつけることができる。

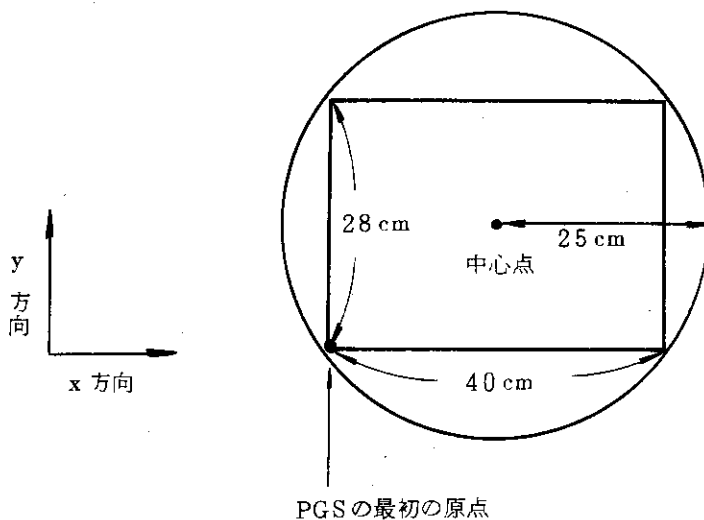


Fig. 3.2 グラフィック・ディスプレイの座標系 (PGSLIB の場合)

(2) 点と直線

プロッタではペンによってプロッタ・シート上に点や直線が描かれ、それらの合成によって意味のある図形が描かれる。これに対してグラフィック・ディスプレイでは、プロッタ

のペンの動きがビームによってCRT面上でシミュレーションされる。すなわち、ビームによって点や直線が表示され、その結果プロッタで描かれるのと同じ図形が表示される。なお、プロッタではどんな複雑な図形でも描くことができるが、後述されるように、グラフィック・ディスプレイではCRT面に表示できる図形の量には限界がある。

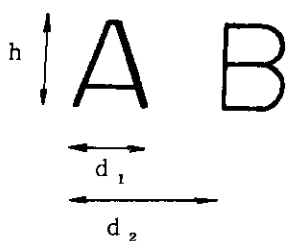
(3) 文字

プロッタでは任意の大きさと任意の角度からなる文字を描くことができる。これに対してグラフィック・ディスプレイではTable 3.2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VSM, SML, LAG, VLGの4種類の大きさの文字でしかも角度0でしか描くことができない。

Table 3.2 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)



そこで、PGSでは、プロッタでmm単位で指定された文字は、指定された文字の高さ h によって、つぎのように上記4種類の文字のいずれかにセットされる。

$0 \text{ mm} \leq h < 4.2 \text{ mm}$ 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.

なお、後述されるプロッタ・タイプ・シンボル命令によって、グラフィック・ディスプレイでもプロッタと同じような任意の大きさと任意の角度からなる文字をソフトウェアで作成し表示することができる。しかし、この場合、メモリと時間の点でハードウェア・シンボルに劣っている。

3.1.2 接続命令

すでに3.1.1で述べたように、プロッタの基本ルーチンとグラフィック・ディスプレイのPGS命令の間では、それぞれ対応する呼出し名が異なっている。そこで、つぎの3.1.3で述べる拡張命令を呼出す2, 3のステートメントの追加を除いては、プロッタ・プログラムが書き換えなしでそのままグラフィック・ディスプレイで実行できるようにするために、基本ルーチンと同じ呼出し名をもつ接続命令がある。この接続命令は対応するPGS命令を自動的に呼出すだけの働きをする。たとえば、接続命令PLOTはPGS命令のGPLOTを呼出す。

3.1.3 拡張命令

プロッタでは、基本ルーチンによって生成されたプロッタに対する命令の集合としてのプロッタ・コマンドは、コア上のバッファに記憶され、バッファが一杯になると自動的に磁気テープなどに出力される。

これに対してグラフィック・ディスプレイでは、PGS 命令によって生成されたグラフィック・ディスプレイに対する命令の集合としてのディスプレイ語列は、一度コア上のディスプレイ語列出力領域GDOA (Graphic Data Output Area) に記憶される。これ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、初めて図形が CRT 面上に表示される。

表示された図形が正しく描かれているか否かを確認するためには、ある程度の時間、グラフィック・プログラムを待ち状態にさせておく必要がある。図形が表示され、プログラムが待ち状態になったとき、利用者は文字キーボードを用いて、日付け、使用者名、タイトル、説明文などのコメントを図形につけることができる。

図形の確認やコメント入力が終わる、つぎに新しい図形を表示するためには、現在表示されている図形を消去しなければならない。これはグラフィック・ディスプレイの画面が直径 50cm の円形に限られているためである。このような場合、プロッタではペンを X 軸方向にある長さ移動するだけで新しい図形を描くことができる。

プロッタではペンで描かれたものが後で記録としてそのまま使用できる。ところがグラフィック・ディスプレイでは、CRT 面に表示されたものは実行が終ると消去されてしまうので別の形で記録しておかなければならない。そのためにはハードウェアのハード・コピー装置が手軽に使用できる。また、鮮明なハード・コピーが必要ならば、プロッタへの出力も可能である。

このような、図形の表示、図形に対するコメント入力、図形の消去、図形のプロッタへのハード・コピー出力、およびグラフィック・プログラムの待ち状態化に関する、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は、GPLOT命令の拡張の形で実現されている。⁽⁵⁾

さらに、コメント入力やプロッタへのハード・コピー出力に関する操作は、PGS ボタンと呼ばれるライトペン・コマンドによって制御される。この PGS ボタンは、グラフィック・プログラムが実行に入るとすぐ Fig. 3.3 のように表示され、実行中はほとんど表示されている。



図 3.3 PGS ボタンとメッセージ

Fig. 3.3 PGS ボタンとメッセージ

Fig. 3.3で

KEY, END, VLG, LAG, SML, VSM, HARDCOPY, NOCOPY

の各文字列がPGS ボタンであるが、特に、

KEY, END, VLG, LAG, SML, VSM

の各ボタンはコメント入力に関するコメント入力用ボタンであり、

HARDCOPY, NOCOPY

の各ボタンは、プロッタへのハード・コピー出力に関するハード・コピー用ボタンである。

以下に、GPLOTの拡張命令について述べるが、利用者は直接GPLOT、または接続命令PLOTを呼出すことによって拡張命令を実行できる。

(1) 表示命令

```
CALL GPLOT (0.0, 0.0, 777)
```

この命令によって、それまでPGS 命令によって生成されGDOAに記憶されていたディスプレイ語列がグラフィック・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。ここでプログラムは待ち状態に入り、利用者は表示された図形の確認や文字キーボードによるコメント入力が可能となる。コメント入力はKEY ボタンをライトペンで指すことによって始めることができる。コメント入力がすべて完了したら、END ボタンをライトペンで指すとプログラムは待ち状態を脱し、GDOA を消去して、次へ進む。もし、コメント入力を行わないときは初めから END ボタンを指せばよい。^{(6), (7)}
なお、コメント入力に関する詳しい操作法はグラフィック・ディスプレイの操作マニュアルを参照下さい。

(2) ハード・コピー命令

```
CALL GPLOT (0.0, 0.0, 888)
```

この命令によって、プログラムはまず待ち状態に入り、利用者はCRT 面の図形をプロッタへハード・コピー出力できる。実際、HARDCOPY ボタンをライトペンで指すたびにCRT 面上の図形はPGS ボタンを除いてプロッタ上に出力される。ハード・コピーがすべて完了したら、NOCOPY ボタンをライトペンで指すとプログラムは待ち状態を脱し、次へ進む。もし、ハード・コピーを行わないときは初めからNOCOPY ボタンを指せばよい。
なお、ハードウェアのハード・コピー装置を利用していつでもプログラムに関係なくCRT 面に表示された図形のハード・コピーをとることができる。

(3) 消去命令

```
CALL GPLOT (0.0, 0.0, 666)
```

この命令によって、それまでCRT 面に表示されていた図形はすべて消去され、引き続いてプログラムが実行に入ったときと同じ画面が再表示される (Fig. 3.3 参照)。このとき、座標の原点は最初の位置 (CRT 面左下端) にリセットされ、GDOA も消去される。この命令ではプログラムが待ち状態に入ることはない。

3.2 汎用GSPによるシミュレーション

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・スムーズイング命令

の7つに大別される。この内、(4), (5), (7)はグラフィック・ディスプレイのみに対して有効な命令であるが、(1), (2), (3), (6)はプロッタの応用ルーチンに対応するグラフィック応用命令である。このグラフィック応用命令はその呼出し名や呼出し形式が対応するプロッタの応用ルーチンと同じであり、プロッタと同じ機能をグラフィック・ディスプレイでシミュレーションできる。そこで、ここでは(1), (2), (3), (6)のグラフィック応用命令についてのみ簡単に述べる。これらの詳細についてはプロッタ関係のマニュアルを参照下さい。^{(8), (9), (10)}

3.2.1 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り、プロッタにおけるファンクショナル・ルーチン群の機能をグラフィック・ディスプレイでシミュレーションできる。

3.2.2 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り、プロッタにおけるTHREEDルーチン群⁽⁹⁾の機能をグラフィック・ディスプレイでシミュレーションできる。

3.2.3 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り、プロッタにおけるGPLOT1ルーチン⁽¹⁰⁾の機能をグラフィック・ディスプレイでシミュレーションできる。ただし、グラフィック・ディスプレイによるこのグラフ表示は、PGS ボタンによって制御され進行するようになっている。⁽⁵⁾

3.2.4 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、その呼出し形式はGSYMBL命令と同じである。SYMB4命令は英数字や特殊文字をソフトウェアで作成し表示するが、この命令はプロッタのSYMBOLルーチンの機能をグラフィック・ディスプレイでシミュレーションできる。しかし、GDOAやバッファ・メモリを沢山必要とし、表示時間がGSYMBL命令よりもかかる欠点がある。

4 PGSLIB とグラフィック・ディスプレイの使用について

前章までに述べてきたことから、プロッタ・プログラムはPGSLIBによってグラフィック・ディスプレイでシミュレーションされ、それゆえ、プロッタ・プログラムのオンライン・デ

バッキングが可能となる。ここでは、これを実際に行う場合に注意すべき事項について述べる。すなわち、プログラミング上注意すべき事項、グラフィック用ジョブ制御文の作り方、グラフィック・ディスプレイで実行するときの注意すべき事項、およびグラフィック・プログラムが実行されるときによく起きるエラー・メッセージについて述べる。

4.1 プログラミング上の注意

4.1.1 図形の大きさ

プロッタ・シートの大きさとCRTの大きさの相違から、1つの図形はプリンタ1ページ程度の大きさに分割して表示すべきである。また必要があれば、尺度因子に関する命令(GFACTRまたはFACTOR)を用いて図形がCRT面に適切に入るよう拡大・縮小を行うとよい。

4.1.2 図形表示量

(1) GDOA

PGSLIBではGDOAの大きさは一応1024語に決っているので、この限界を越えぬ前に表示命令を呼出す必要がある。表示命令を呼出すたびにプログラムが待ち状態に入るのがわずらわしい場合は

```
CALL GPLOT (0.0, 0.0, 444)
```

によって図形表示を行うとよい。この命令では図形表示後GDOAの消去は行われるが、プログラムは待ち状態には入らず、すぐつぎへ進む(したがってこの命令ではコメント入力を行えない)。図形表示量が多い場合は、この命令をところどころで呼出すようにしておき、少し時間をかけて図形の確認が必要なところでは

```
CALL GPLOT (0.0, 0.0, 777)
```

を用いるとよい。

(2) バッファ・メモリ

現在設置されているグラフィック・ディスプレイのバッファ・メモリの大きさは8kW(1W=16bit)である。このバッファ・メモリの大きさを越えてCRT面に図形を表示することはできない。また、限界を越えなくともそれに近くなると画面がちらついて見にくくなる場合がある。このときは1画面を分割して2画面以上にしなければならない。

4.1.3 文字の大きさと角度

すでに3.1.1の(3)で述べたように、プロッタとグラフィック・ディスプレイとでは文字の描かれ方が異なっている。このことから2, 3注意すべき事項がある。

- (1) 文字のパターンや文字と文字との間隔の取り方がちがっていることから、たとえば、プロッタでは直線による図形と文字列とが重ならないようになっていても、それがグラフィック・ディスプレイで表示されたときは重なる場合がある。
- (2) Y座標軸のメモリの描かれ方はプロッタとグラフィック・ディスプレイとでは異なる。それはグラフィック・ディスプレイでは文字が角度0でのみ表示されるからである。
- (3) 上記の(1), (2)のようなことを特にさけたい場合は、GSYMBL命令をつぎのような簡単

なサブルーチンで置き換えるとよい。このとき、GDOAの量が前よりも多くなるので、表示命令を途中で多目に呼出す必要がある。

Table 4.1 SYMB4を用いたGSYMBLサブルーチン

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.12.19 PAGE 2

* SOURCE STATEMENT *

1	SUBROUTINE GSYMBL (X,Y,H,BCD,THETA,N)
2	CALL SYMB4 (X,Y,H,BCD,THETA,N)
3	RETURN
4	END

4.1.4 プログラムの大きさ

現在グラフィック・プログラムで使用できるコア容量とCPU使用時間は、最大でそれぞれ256K語、30分である。

4.2 グラフィック用ジョブ制御文

プロッタ・プログラムをグラフィック・ディスプレイでシミュレーションし、デバッキングする場合に使用されるジョブ制御文の代表的な構成例を示し、その意味を簡単に説明する。

FACOM 230-75ではグラフィック・プログラムは、プロッタ・プログラムの場合と同じようにFORTRAN-C/DおよびFORTRAN-Hの両方で実行できる。ここでは、まずFORTRAN-C/Dの場合の例をFig. 4.1に示す。

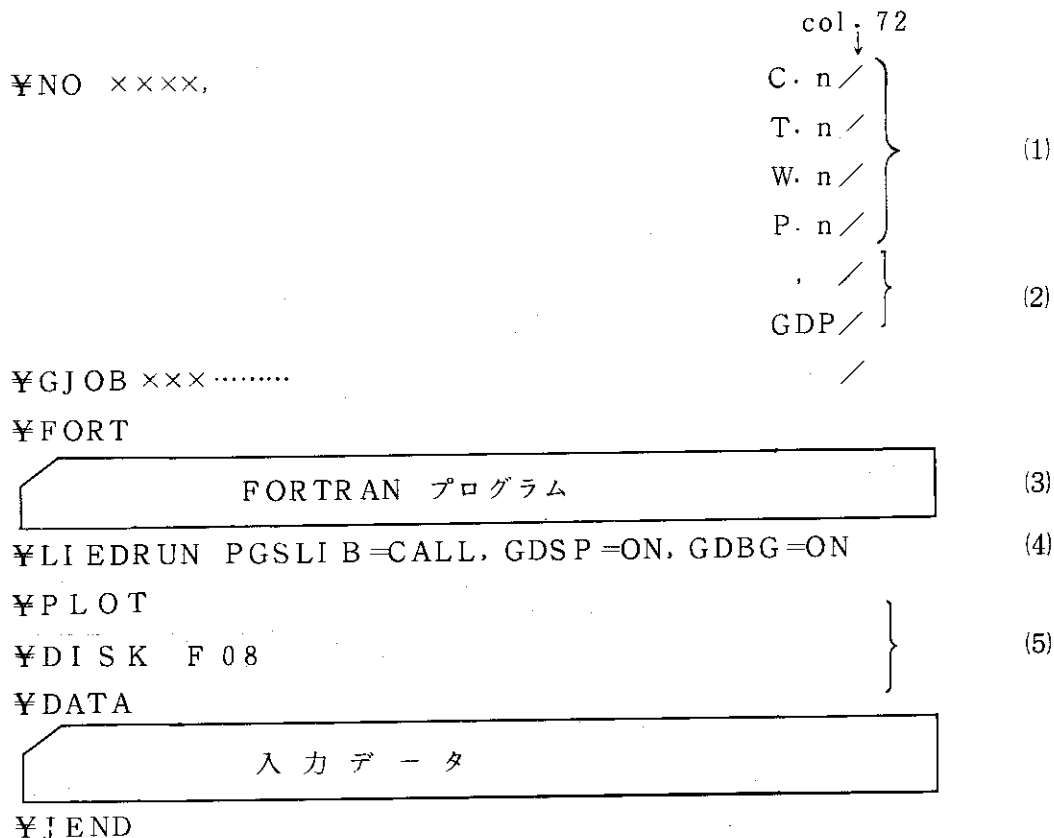


Fig. 4.1 グラフィック用ジョブ制御文

Fig. 4.1 で、(1)は一般ジョブと同じように使用するシステム資源の範囲を示す NO 文の追加情報 1 であり、グラフィック・ジョブでは主記憶量，CPU 時間，ライン・プリンタ出力量，カード出力量はそれぞれ，256K 語，30 分，240 ページ，0 枚以下でなければならない。

(2)は NO 文の追加情報 2 で，グラフィック・ジョブ・クラスの指定を意味する。このジョブ・クラスでは，通常 1 ジョブだけは他の一般ジョブ・クラスのジョブに比べ優先的な扱いを受ける。

(3)は，プロッタ・プログラムに拡張命令呼出しに関する 2，3 のステートメントが追加され，グラフィック・プログラムとして実行される FORTRAN プログラムである。

(4)は，グラフィック・プログラムの結合編集および実行に関するジョブ制御文である。PGSLIB=CALL は，結合編集時にグラフィック・サブルーチン・ライブラリとしての PGSLIB を他のライブラリに優先して自動コールすることを意味する。GDSP=ON は実行時にグラフィック・ディスプレイを使用することを意味し，GDBG=ON はグラフィック・プログラムをデバッキング・モードで実行することを意味する。デバッキング・モードではプログラムのエラー・メッセージは CRT 面に表示され，かつプリンタにも出力される。GDBG=ON を省略するとエラー・メッセージは全然出力されない。

(5)は，実行段階で CRT 面に表示された図形をプロッタにハード・コピーするときに指定するジョブ制御文である。プロッタ・ハード・コピーを全然行わないときは，このジョブ制御文はそう入してはならない。

Fig. 4.1 で上で説明されなかったその他のジョブ制御文などについては，一般ジョブの場合と同じである。

ジョブ制御文 ¥FORT と ¥LIEDRUN については，それぞれ ¥DFORT，¥DLIEDRUN で置き換えてもよい。

つぎに，FORTRAN-H でグラフィック・プログラムを実行する場合は，Fig. 4.1 で

```
¥FORT
```

の代わりに

```
¥HFORT MODE=CDEX
```

また，

```
¥LIEDRUN PGSLIB=CALL, GDSP=ON, GDBG=ON
```

の代わりに

```
¥HLIEDRUN GLIED=LIED, PGSLIB=CALL,  
GDSP=ON, GDBG=ON
```

を用いればよい。

4.3 グラフィック・ディスプレイの使用

4.3.1 セミ・オープン室

グラフィック・ディスプレイは現在セミ・オープン室に設置され，通常は FACOM 230-75 の A システムに接続されている。このグラフィック・ディスプレイを使用するグラフィック・ジョブに関する操作はすべてセミ・オープン室で行うことができる。

グラフィック・ディスプレイを使用する場合は、まずセミ・オープン室隣の廊下にある黒板の使用時間帯に使用者の名前を使用日の前日までに書き込んでおかなければならない。グラフィック・ディスプレイの使用が完了したら、備付のグラフィック・ディスプレイ使用記録簿に使用者名、使用時間帯などを記録する。

4.3.2 グラフィック・ジョブの実行

グラフィック・ディスプレイの使用は、セミ・オープン室のカード・リーダーにジョブ・デックを投入することから始まる。投入されたグラフィック・ジョブのFACOM 230-75 による処理状況は、セミ・オープン室のコンソール・ディスプレイに表示される。

グラフィック・プログラムの翻訳、結合編集が終り、実行に入ると、PGSLIB によってコンソール・ディスプレイ上に2度続けて

```
*** GRAPHIC START ***
```

のメッセージが表示される。このメッセージが表示されたら利用者はグラフィック・ディスプレイの前にゆき、以後はプログラムの仕様にしたがってグラフィック・ディスプレイを操作すればよい。

グラフィック・プログラムが実行に入ったとき、グラフィック・ディスプレイのCRT 面にはすぐにPGS ボタンとウインクするメッセージが表示される (Fig. 3.3参照)。PGS ボタンについてはプログラムが実行中ほとんど表示されているが、メッセージは利用者の図形が表示される直前にCRT 面から消去される。

グラフィック・ジョブの実行が終ると、計算結果などはセミ・オープン室に設置されている2台のプリンタのどちらかに出力される。

4.3.3 ハード・コピー

グラフィック・ディスプレイに表示された図形はハード・コピー装置またはプロッタでハード・コピー出力できる。ハード・コピー装置の場合はいつでもPRINT キーを押すことによってハード・コピーできる。プロッタに出力する場合は、ASW (Alteration Switch) 8個を全部オフにセットしておいてプログラムを実行させ、プログラムで

```
CALL GPLOT (0.0, 0.0, 888)
```

に到達したところでHARDCOPY ボタンをライトペンで指せばよい。ハード・コピーが終ったらNOCOPY ボタンをライトペンで指すとプログラムは次へ進むことができる。

プロッタ・ハード・コピーの制御はファンクション・キーの8番, 16番によっても行うことができる。すなわち、ファンクション・キーの8番がHARDCOPY ボタンに対応し、ファンクション・キーの16番がNOCOPY ボタンに対応する。ただし、HARDCOPY ボタンによる場合はPGS ボタン以外の図形のみがハード・コピーされるのに対して、ファンクション・キーの8番による場合はPGS ボタンも含めてすべての図形がハード・コピーされる。

このようなハード・コピーの際、PGSLIB は図形に関するプロッタ・コマンドを磁気ディスクに出力するだけである。したがって、ハード・コピーに要する時間はわずかである。磁気ディスクからプロッタ用の磁気テープへの書き込みは後でシステム出力制御プログラムによって行われる。このプロッタ用の磁気テープはオペレータによってCALCOMP 900/937/1136 プロッタでプロットされ、利用者に配布される。

4.3.4 ロール・イン／ロール・アウト

FACOM 230-75でのジョブ処理状況が混雑しているときで、グラフィック・プログラムが待ち状態になっているときは、プログラムがロール・アウトされる状態がおきる場合がある。このときは、CRT面の左下に下記のメッセージが表示される。

```
GSP NO JOB WA ROLL OUT TYU
```

しかし、利用者がライトペンや文字キーボードを使い始めるとプログラムはすぐロール・インされるような仕組みになっている。すなわち、利用者が考えているときはロール・アウトされがちであるが、使い始めるとすぐロール・インされる。したがって、主記憶の有効利用が可能となっている。

4.3.5 エラー・メッセージ

グラフィック・ジョブの実行中に起きるグラフィック・ジョブ特有のエラー・メッセージについてはつぎの4.4で述べる。それ以外のエラー・メッセージについては一般ジョブと変わらないので特に説明しない。

4.4 エラー・メッセージ

翻訳、結合編集の段階ではグラフィック・ジョブ特有のエラー・メッセージは出力されない。しかし、実行段階に入ると、一般ジョブでは出力されないようなグラフィック・ジョブ特有のエラー・メッセージが出力される場合がある。

グラフィック・プログラムをデバッキング・モードで実行すると、プログラムに関するエラー・メッセージはCRT面の左上につぎのように表示される。

```
GS 0010 ee GSP サブルーチン名 KILL CONT
```

ここで、eeはエラー・コードであり、KILLはライトペンで指すと、プログラムが終制終了となる。エラーを無視してプログラムを先に進めたい場合は、CONTをライトペンで指せばよい。CRT面に表示されたエラー・メッセージはプリンタにも出力される。したがって、実行終了後でもエラーの原因を探ることができる。エラーの程度が軽いものについてはCRT面には表示されず、プリンタのみに出力されるものもある。

デバッキング・モードを指定しなかったときは、エラーが起きてもエラー・メッセージは出力されず、プログラムはエラーを無視して先へ進む。

以下に、プロッタ・プログラムをグラフィック・ディスプレイでシミュレーションするとき起りがちないくつかのエラーについて簡単に説明する。

(1) エラー・コード25 : バッファ・オーバ・フロー

表示命令が実行されたとき、ディスプレイ・バッファがオーバした意味である。これをさけるためには、1画面に表示される図形を分割して、2画面以上に表示すべきである。

(2) エラー・コード26 : スクリーン・リミット・オーバ

コメント入力の実行中に起きるが、利用者は無視することができる。プリンタに出力されるエラー・メッセージが困る場合は、デバッキング・モードを指定しないで実行すればよい。

(3) エラー・コード41 : GDOAオーバ・フロー

ディスプレイ語列の出力領域としてのGDOAがパンクした意味である。これをさけるためには、表示命令を少し多目に実行すればよい。

5 デバッキング例

Table 5.1はプロッタ・プログラムのリストである。Table 5.2はこのプログラムをFACOM 230-75のFORTRAN-C/Dで実行したときのジョブ制御文のリストである。このプログラムによってプロッタには5個のグラフ図形が出力されるが、Fig. 5.1は1番目に出力されたグラフ図形である。

このプロッタ・プログラムをグラフィック・ディスプレイでシミュレーションし、デバッキングするために、Table 5.1のプログラムに拡張命令呼出しのステートメントを追加したのがTable 5.3のプログラムである。このグラフィック・プログラムをFACOM 230-75のFORTRAN-C/Dで実行したときのジョブ制御文のリストがTable 5.4である。Fig. 5.2は1番目にCRT面に表示されたグラフ図形をプロッタにハード・コピーしたものである。このプロッタ・ハード・コピーはCRT面に表示された図形の $1/2$ の大きさで出力される。

6 おわりに

以上、述べてきたように、プロッタ・プログラムのデバッキングは、グラフィック・ディスプレイとPGSLIBを用いることによって短時間に集中して効率良く行うことができる。もし、プロッタ・プログラムが数値計算に多くの時間がかかる場合は、数値計算の部分と図形出力の部分に分けてデバッキングを行うとよい。逆に、数値計算の時間が少ないものについてはデバッキングのみならず本番もグラフィック・ディスプレイで処理できる。その場合、グラフィック・ディスプレイに表示された図形の記録には、ハードウェアのハード・コピー装置またはプロッタが利用できる。

このPGSLIBを用いる方法では、プロッタ・プログラムの中に表示命令を呼出すステートメントを追加するようになっている。このようなステートメントの追加は、PGSLIBを単にプロッタ・プログラムのデバッキング用に限定して作成すれば、必要ではなくなる。すなわち、GDOAが一杯になったら自動的にディスプレイ語列をディスプレイ・バッファへ転送するようになればよい。しかし、現在のPGSLIBは対話型グラフィック・プログラミング用の道具として作られているため、図形の表示に利用者が介入できるようになっている。

グラフィック・ディスプレイでデバッキングを行うために、図形の表示や消去などに関する2, 3のステートメントが追加されたプロッタ・プログラムはそのままプロッタで処理可能である。それは、原研のプロッタ基本ルーチンではそのような命令をNOP (NO OPERATION)として処理するように改良されているからである。ただし、ジョブ制御文については全く同じではできないので変更の必要がある。

Table 5.1 プロッタ・プログラムのリスト

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.12.16 PAGE 1

* SOURCE STATEMENT *

```

1 DIMENSION TV(5),XX(52),YY(52)
2 DIMENSION BUFF(1024)
3 C
3 CALL PLOTS(BUFF(1),1024)
4 CALL FACTOR(2.0)
5 C
5 DLTX=0.01
6 TV(1)=20.0
7 TV(2)=25.0
8 TV(3)=25.4
9 TV(4)=10.0
10 TV(5)=12.0
11 DO 250 I=1,5
12 DLTX=DLTX*2.0
13 XX(1)=DLTX
14 DO 240 J=1,50
15 YY(J)=.005*XX(I,J)**(I)+.04*XX(J)**(I-1)+.3*XX(J)**(I-2)
16 240 XX(J+1)=XX(J)+DLTX
17 C
17 CALL SYMBOL(10.,90.,3.0,17,LINE PLOT EXAMPLE,0.0,17)
18 CALL SYMBOL(20.,80.,3.0,3HNO,0.0,3)
19 CALL NUMBER(40.,80.,3.0,FLOAT(I),0.0,-1)
20 CALL SCALE(XX,65.,50,1,TV(I))
21 CALL SCALE(YY,100.,50,1,TV(I))
22 CALL AXIS(0.0,0.0,0.8HVERTICAL,8,100.,90.0,YY(51),YY(52),TV(I))
23 CALL AXIS(0.0,0.0,0.6HHORIZ,6,65.,0.0,XX(51),XX(52),TV(I))
24 CALL LINE(XX,YY,50,1,1,1)
25 CALL PLOT(150,0.0,0,-3)
26 250 CONTINUE
27 CALL PLOT(0.0,0.0,999)
28 C
28 STOP
29 END
    
```

Table 5.2 プロッタ・プログラムに対するジョブ制御文のリスト

FACOM 230-60/75 MONITOR6/7 SYSTEM=V04/L15 BATCH=V04/L01 JOB CONTROL LANGUAGE LIST 75.12.16 PAGE 1

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

```

1 *NO 1803, /
P.1/PCH 0
2 VM *GJOB 0411803,NAKAMURA,Y,341.02 /GRAPHICS
3 VM *FORT
4 VM *LIEDRUN
5 VM *PLOT
6 *JEND
    
```

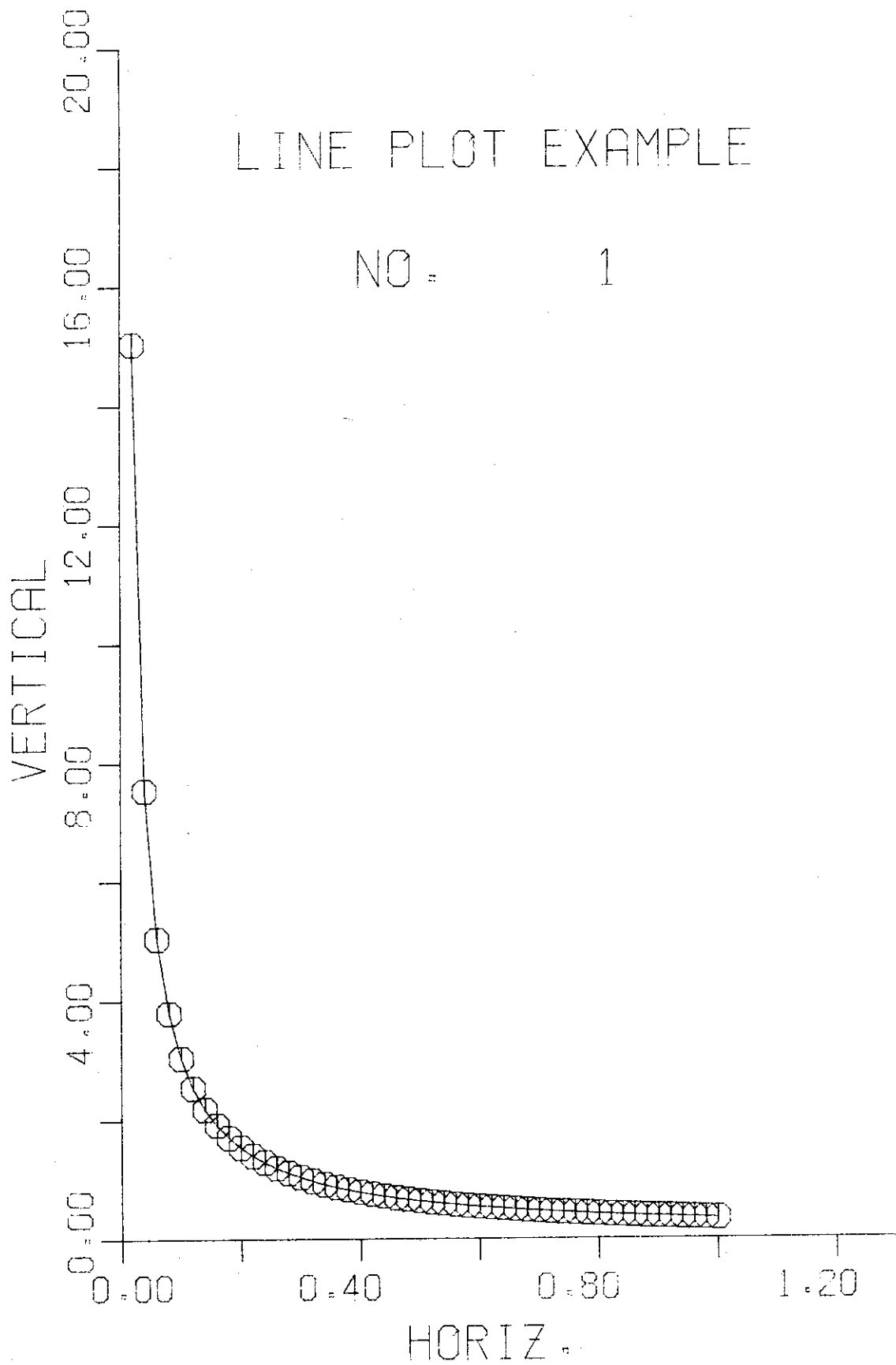


Fig. 51 プロッタ・プログラムによりプロッタに出力されたグラフ

Table 5.3 グラフィック・プログラムのリスト

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.12.18 PAGE 1

* SOURCE STATEMENT *

```

1      DIMENSION TV(5),XX(52),YY(52)
2      DIMENSION BUFF(1024)
3      C
4      CALL PLOTS(BUFF(1),1024)
5      CALL FACTOR(2,0)
6      C
7      DLTX=0.01
8      TV(1)=20.0
9      TV(2)=25.0
10     TV(3)=25.4
11     TV(4)=10.0
12     TV(5)=12.0
13     DO 250 I=1,5
14     DLTX=DLTX*2.0
15     XX(1)=DLTX
16     DO 240 J=1,50
17     YY(J)=.005*XX(J)**(I)+.04*XX(J)**(I-1)+.3*XX(J)**(I-2)
18     240 XX(J+1)=XX(J)+DLTX
19     C
20     CALL SYMBOL(10.,90.,.3,0.17)LINE PLOT EXAMPLE(0.0,17)
21     CALL SYMBOL(20.,80.,.3,0.3)HNO(0.0,3)
22     CALL NUMBER(40.,80.,.3,0.FLOAT(I),0.0,-1)
23     CALL SCALE(XX,65.,50,1,TV(I))
24     CALL SCALE(YY,100.,50,1,TV(I))
25     CALL AXIS(0.0,0.0,0.8)VERTICAL(8,100.,90.0,YY(51),YY(52),TV(I))
26     CALL AXIS(0.0,0.0,0.6)HORIZONTAL(6,65.,0.0,XX(51),XX(52),TV(I))
27     CALL LINE(XX,YY,50,1,1,1)
28     CALL PLOT(150.0,0.0,-3)
29     C
30     CALL PLOT(0.0,0.0,777)
31     CALL PLOT(0.0,0.0,888)
32     CALL PLOT(0.0,0.0,666)
33     C
34     250 CONTINUE
35     CALL PLOT(0.0,0.0,999)
36     C
37     STOP
38     END
    
```

Table 5.4 グラフィック・プログラムに対するジョブ制御文のリスト

FACOM 230-60/75 MONITOR6/7 SYSTEM=V04/L15 BATCH=V04/L01 JOB CONTROL LANGUAGE LIST 75.12.18
1.....2.....3.....4.....5.....6.....7.....8 PAGE 1

```

1      *NO 1803.
2      VM #GJOB      0411803,NAKAMURA.Y,341.02
3      VM #FORT
4      VM #LIEDRUN  PGSLIB=CALL.GDSP=ON.GDBG=ON
5      VM #PLOT
6      VM #DISK F08
7      *JEND
    
```

/

T.2/TIME 1M

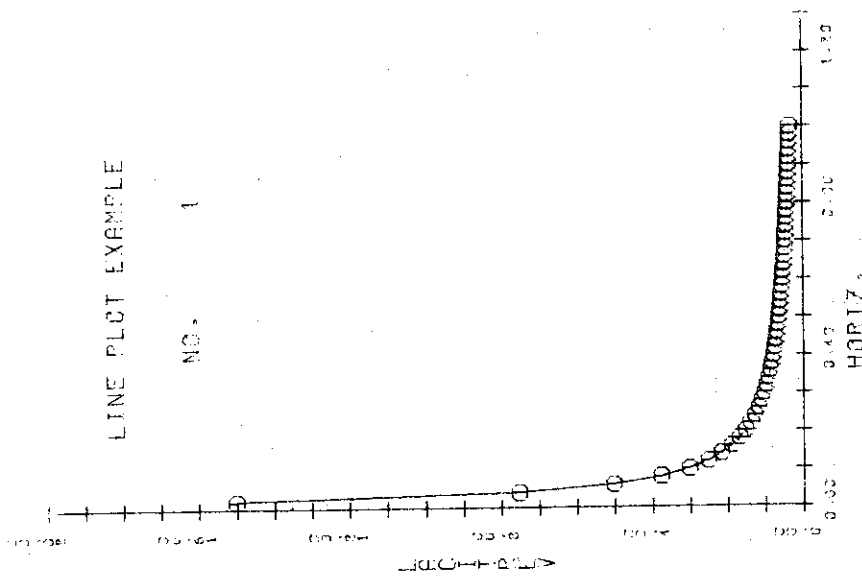
P.1/PCH 0

//

GDP/

/GRAPHICS.

KEY
END
VLG
LAG
SML
VSM



HARDCOPY NOCOPY

Fig. 52 グラフィック・プログラムによりCRT面に表示されたグラフの
プロッタ・ヘッド・コピー

昨年12月末に導入されたグラフィックCOM (Computer Output Microfilm System) の場合も、そのプログラムがプロッタ・プログラムとほぼ互換性をもつように現在COM用の基本ソフトウェアを開発している。したがって、COM用プログラムのデバックングにも上で述べた方法が利用できるであろう。

また、プロッタ・プログラムやCOM用プログラムのデバックングが簡易型グラフィック・ディスプレイとしてのソニー・テクトロニクス4014を用いてもできるように、その基本ソフトウェアPTS (Plotter-compatible Tektronix Graphic Subroutine Package) の開発を進めている。ソニー・テクトロニクス4014はTSS 端末として接続されているので、PTS が完成すれば遠方からTSS モードでデバックングできることになる。

本レポートでは、プロッタ・プログラムのオンライン・デバックングにしぼって述べてきたが、このプログラムは簡易GSP を用いることによってライトペンや文字キーボードを用いた本格的な対話型グラフィック・プログラムに拡張することもできる。

参 考 文 献

- (1) 吉沢ビジネス・マシンズ(株) : 超高速プロッタ 900・937・1136 システム概説マニュアル
- (2) 吉沢ビジネス・マシンズ(株) : CALCOMP プログラミング・マニュアル-I, 1969
- (3) 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
- (4) 富士通(株) : FACOM GSP 文法編, 1973
- (5) 中村康弘, 小沼吉男 : PGS LIB/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
- (6) 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, JAERI-M 5659, 1974
- (7) 中村康弘, 小沼吉男 : FACOM 230-75 グラフィック・ディスプレイの使い方,
JAERI-M (to be published)
- (8) 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
- (9) CALCOMP, INC. : THREE D : A Perspective Drawing Software System,
1969
- (10) 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GPLOT C
の開発, JAERI-memo 4255 (公開), 1970