

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-437

PLOTルーチン形式による図形出力装置
の使用について

石塚只夫・磯部俊夫

1981年6月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

PLOTルーチン形式による図形出力装置 の使用について*

石塚 只夫** 磯部 俊夫**

1. まえがき

電子計算機の計算結果をグラフ等図形の形で出力することは、一般に広く行なわれている。航空宇宙技術研究所（以後航技研と略記）でもデータロン205時代から、電子計算機システムにXYプロッタが導入され、使用されていた。現在航技研計算センターでは、2台の大型XYプロッタが稼働している。中央機FACOM230-75には、オンラインでグラフィック・ディスプレイ（以後グラフィック・ディスプレイと略記）が接続されている。またデータ処理用のミニコンピュータNOVAには、ストレージ型CRTグラフィック・ディスプレイ・ターミナル（以後NOVAグラフィック・ディスプレイと略記）が接続されている。中央機とデータ処理システムは互にチャンネルで結ばれ、オンラインでデータ交換ができる。したがって現在計算センターで使用可能な図形出力装置はXYプロッタ、グラフィック・ディスプレイおよびNOVAグラフィック・ディスプレイの3種類である。これらは、それぞれのハードウェアの構造が異なるのはもちろん、提供されているソフトウェアも異なる。たとえば、XYプロッタを使って図形を描かせるためのプログラムは、そのままではグラフィック・ディスプレイの上に図形を描きだすことはできない。グラフィック・ディスプレイ用ソフトウェアを使って、プログラム・コンバージョンを行う必要がある。同一計算センター内にある図形出力装置を使用するのに、それぞれのソフトウェアでプログラムを書かなければならないというのは大変不便である。共通のソフトウェアで使えれば、利用者にとって好都合である。

各種の図形出力装置を一つのソフトウェアで使いこなそうという試みが、いくつか行なわれてい

る。^{1)~3)} この場合、使用者が既に使用している図形出力装置のソフトウェアで書かれたプログラムを、翻訳プログラムにより他の新しい図形出力装置のソフトウェアに置き替えるやり方と、図形出力用の標準言語を新しく設計し、各図形出力装置とのインターフェースを標準言語の中でのとりやり方がある。最初の方法では、使用者は現在使っている図形出力装置の言語でプログラムを書けば、そのまま他の図形出力装置に出力することができる。また新しい装置のソフトウェアの勉強をしなくてもよいという利点がある。一方、従来の装置にない機能が新しい装置に付加されても、それらが充分生かせなく、実行時の能率が悪くなる欠点をもつ。新しい標準図形出力用言語を作る方法では、各図形装置の機能を生すことができ、効率も上げることができる。しかし、プログラム開発に多くの時間と人手を要する。使用者にも新しい言語に慣れてもらう必要がある。

航技研では計算機使用者の多くがXYプロッタを使ってデータを図形化している。XYプロッタを使ったプログラムの蓄積も多い。そこでわれわれはXYプロッタ用言語を用い他のグラフィック装置を使用する方法で図形出力ルーチンの統一を計ることにした。

われわれはグラフィック・ディスプレイを従来のXYプロッタ用ソフトウェアで動かすプログラムを開発した。このプログラムの中には、XYプロッタのためのプログラムそのまま、ラインプリンタ上に図形を出力するプログラムも含めた。ラインプリンタ上に文字を印字して図形を描かせる方法は、特別な機器を必要とせず、簡易に図形が得られるという利点があるため広く使われている。もっともラインプリンタ上に高々百数十行の文字列を並べ図を描かせるのであるから、粗い図形しか得られないが、使い方によっては役立つであろう。またNOVAグラ

** 昭和56年5月7日受付

** 計算センター

フィックでは、中央機でXYプロッタ用のプログラムを走らせ、発生した図形データをオンラインでNOVAに転送し、NOVAグラフィックへ図形を出力させる実験を行なった。

最後に、線画出力のために必要な機能について述べ、装置のモデル化を行ない、一般化したインターフェース・ルーチンを提案する。

2. 各グラフィック機器の基本ソフトウェア

ここでは、XYプロッタ、グラフィック・ディスプレイおよびNOVAグラフィックの基本ソフトウェアの概要を説明し、XYプロッタのソフトウェアでグラフィック・ディスプレイを使用する方法を述べる。

2.1 XYプロッタ

航技研計算センターでは、XYプロッタ用ソフトウェアとして、カルコンプ社で開発されたカルコンプ・プロットング・システム⁴⁾を使用している。これはサブルーチン形式のプログラム・パッケージで、ベーシック・ソフトウェア、ファンクショナル・ソフトウェアから構成されている。ベーシック・ソフトウェアは、XYプロッタを制御するための基本ソフトウェアを中心に英数字、グラフ等XYプロッタを使用するために最低限必要と考えられるサブルーチンを集めたものである。ファンクショナル・ソフトウェアは、円、楕円、格子、矢印等の種々の作図をパラメータを与えるのみで容易に作図するためのサブルーチンを集めたものである。XYプロッタ使用者は、これらのサブルーチンを使い作図プログラムを作る。このカルコンプ・プロットング・システムにおいて、最も基本的なサブルーチンは、“PLOTルーチン”である。

PLOTルーチンの機能は、プロッタのペンの現在位置から指定された位置まで、ペンをダウンあるいはアップして、ペンを移動することである。ペンをダウンで移動すると、紙の上に線が描かれる。すなわち現在位置(X_0, Y_0)から指定された位置(X, Y)へ直線を引くことが、このPLOTルーチンの目的である。XYプロッタとプロットング・

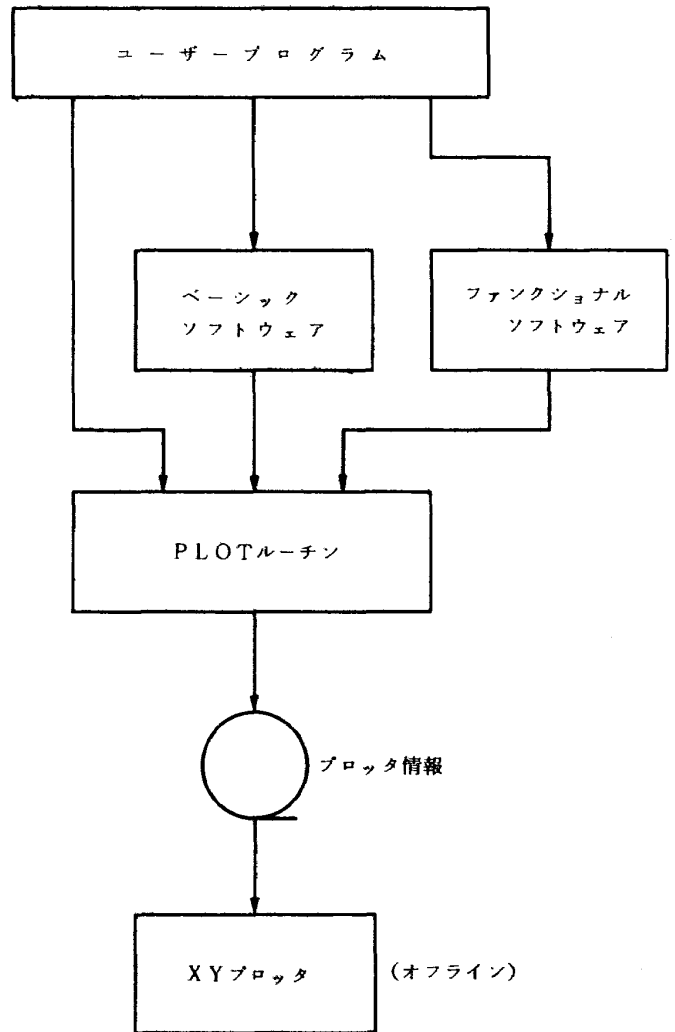


図1 プロットング・システムの構造

システムのインターフェースを受けもっていると考えてもよい。ベーシック・ソフトウェア、ファンクショナル・ソフトウェアによる作図も、最終的にはすべてこのPLOTルーチンをCALLし、作図される。図1にプロットング・ソフトウェアの構造を示す。

2.2 グラフィック・ディスプレイ

FACOM230-75にオンラインで接続されているグラフィック・ディスプレイ装置(F6234A)は、ディスプレイ装置使用者がオンラインでユーザ・プログラムとコミュニケーションを行なう機能をもっている。このグラフィック・ディスプレイ装置をユーザ・プログラムから容易にアクセスするための基本ソフトウェアとして、富士通GRASP⁵⁾(GRAphics Service Program)がある。これもサブルーチ

ンの集まりで構成されている。GRASPは、オンライン処理および複数台のディスプレイ装置を制御できるように設計されている。したがってXYプロッタのように、バッチ処理のみで使用され、出力先が1台のプロッタに限定されているシステムのような単純明快なシステム構成になっていない。ディスプレイに描こうとするイメージを、イメージデータという概念でとらえ、ディスプレイ上に図形を表示するのは、イメージデータをディスプレイ装置に出力するということになる。描こうとする図形を計算機内のデータに変換するイメージデータの生成とイメージの表示とが明確に分けられている。GRASPには、イメージ表示のほかに、文字キーボード、ファンクション・キーボードからのデータ入力用サブルーチン群、ユーザとのコミュニケーションのためのアテンション・サブルーチン群が用意されている。さて、イメージの表示がどのように行なわれているか見てみよう。ここで使用されているディスプレイはストレージ型のもので、画面の左下を原点(0,0)

とし、右上が(1023,1023)となるようにアドレス付けされている。イメージ生成サブルーチンには

- (1) 画面上の任意の位置にポインタを設定する。
- (2) なん本かの連続した線分を生成する。
- (3) なん点かの点を生成する。
- (4) 文字を生成する。

の4種類がある。これらのサブルーチンによりイメージデータを生成した後、イメージ制御サブルーチンEXECにより、イメージデータがディスプレイに表示される。イメージ制御サブルーチンの中には、画面上のイメージ全部を消去するためのルーチンも用意されている。(GRASPには、プロッタ・システムにおけるファンクショナル・サブルーチンに相当するルーチンは含まれていない。)

以上がGRASPによるイメージ発生概要である。図2はGRASPの基本構造を示したものである。

2.3 NOVAグラフィック

このグラフィック・ディスプレイは、データ処理

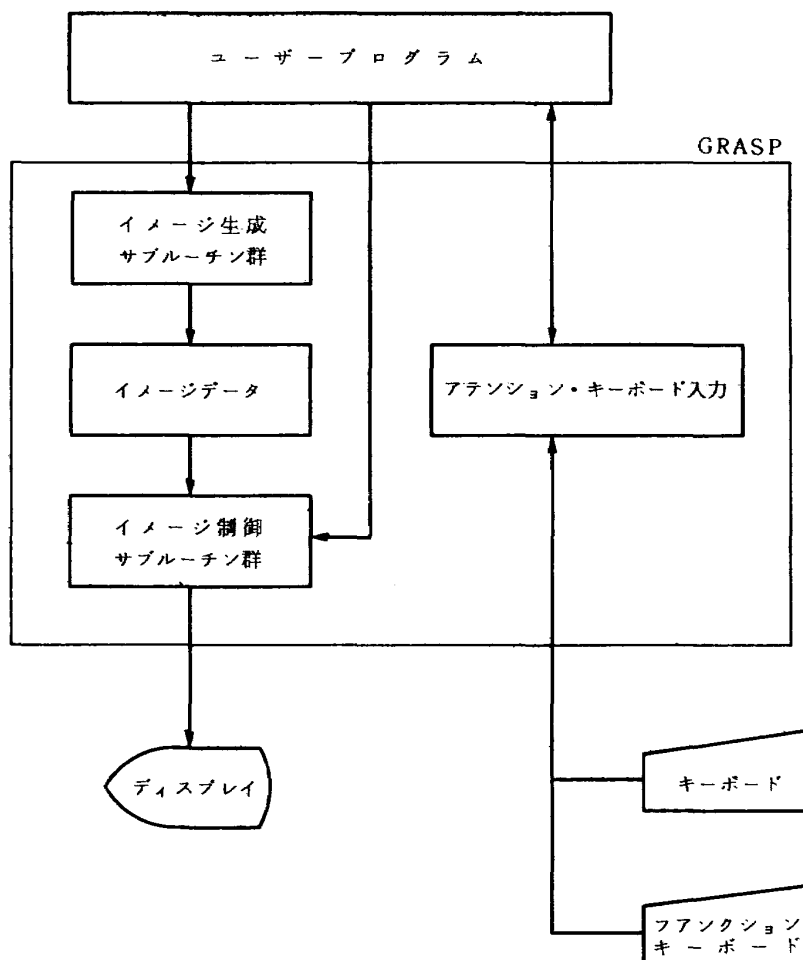


図2 GRASPの基本構造

設備のミニコンピュータNOVA(TACC-840M)に接続されているグラフィック・ディスプレイ・ターミナルTR-7985で、ストレージ型のディスプレイある。ソフトウェアは、タケダ理研(株)から提供されているグラフィック・ディスプレイ・ライブラリ⁶⁾を使用している。これもFORTRANサブルーチンの集まりである。このライブラリは、ミニコンピュータのソフトウェアでもあり、規模は前の2つのシステムほど大きなものではない。画面は左下が原点(0,0)であり、右上は(1023,780)^{注)}のアドレスが付けてある。図形を描かせるための基本サブルーチンはVCTRGルーチンで、これはXYプロッタにおけるPLOTルーチンと全く同じ機能を持ち、さらに指定した座標へ点を表示する機能を兼ね備えている。VCTRGルーチンで描く直線は、モード指

定ルーチンにより、実線、点線、一点鎖線等5種類の選択ができる。またプロットング・システムにおけるファンクショナル・ルーチンに相当するいくつかのサブルーチンと画面消去のサブルーチンが用意されている。

2.4 PLOTルーチンへの統一

2.1から2.3までに述べた各図形出力装置が現在計算センタで使用されている。ソフトウェアの図形生成ルーチンをまとめると、表1のようになる。この表から分かるように、各ルーチンのCALL形式は異なるが、各ルーチンが実際に行なっているのは現在の位置から与えられた位置まで直線を引くということである。XYプロッタ以外のディスプレイでは、さらに与えられた位置に点を表示する機能も持っている。ここで各ソフトウェアで (X_p, Y_p) に一点表示、および (X_0, Y_0) から (X, Y) への直線を引く命令シーケンスを作ると、表2のようになる。

注) 現在TR-7985の他に画面の大きさが(4096,3120)の4014ディスプレイも、使用可能。

表1 各図形出力装置の図形成生ルーチン

図形出力装置	ルーチン名	機能
XYプロッタ (プロットング・システム)	PLOT(X, Y, I)	I=2 ペンをダウンして(X, Y)まで移動する (現在の位置から(X, Y)まで線を描く)
		I=3 ペンをアップして(X, Y)まで移動する
グラフィック・ディスプレイ (GRASP)	MVPOS(X, Y)	ポインタを(X, Y)へ移す
	PLINE(X, Y)	現ポインタから(X, Y)まで線を描く
	PPNT(X, Y)	(X, Y)に点を表示する
NOVAグラフィック	VCTRG(X, Y, I)	I=0 (X, Y)に点を表示する I=1 ポインタを(X, Y)に移動する I=2 現ポインタから(X, Y)まで線を描く

(ここで示した各ルーチンの引数は、機能説明に必要なものだけとした。実際のGRASP, NOVAの各ルーチンではさらにいくつかの引数が必要である。)

表2 各図形出力装置での点・直線表示の命令シーケンス

表示	XYプロッタ	GRASP	NOVA
点 (X_p, Y_p)	CALL PLOT($X_p, Y_p, 3$) CALL PLOT($X_p, Y_p, 2$)	CALL PPNT(X_p, Y_p)	CALL VCTRG($X_p, Y_p, 0$)
線 $(X_0, Y_0), (X, Y)$	CALL PLOT($X_0, Y_0, 3$) CALL PLOT($X, Y, 2$)	CALL MVPOS(X_0, Y_0) CALL PLINE(X, Y)	CALL VCTRG($X_0, Y_0, 1$) CALL VCTRG(X, Y, 2)

XYプロッタでは、点を表示するという概念がなく、あえて点を表示しようとする長くない線を描くということになる。このXYプロッタ流の考え方をグラフィック・ディスプレイにも適応するとXYプロッタでの

CALL PLOT(X, Y, I)は
GRASPでは

I = 2 のとき CALL PLINE(X, Y)

I = 3 のとき CALL MVPOS(X, Y)

NOVAグラフィックでは

I = 2 のとき CALL VCTRG(X, Y, 2)

I = 3 のとき CALL VCTRG(X, Y, 1)

と等価である。

プロットング・システムで書かれたプログラムは、最終的にPLOTルーチンをCALLして作図する。そこでCALL PLOT(X, Y, I)のステートメントを上記のように書き替えば、プロットング・システムで書いたプログラムで、それぞれのディスプレイ装置に図形が描けることになる。プログラミング上は、ディスプレイ用のPLOTルーチン形式のルーチンを作ればよい。

グラフィック・ディスプレイ用PLOTルーチンは、次のようになる。

```
SUBROUTINE PLOT(X, Y, I)
```

```
IF(I. EQ. 3) GO TO 10
```

```
CALL PLINE(X, Y)
```

```
GO TO 20
```

```
10 CALL MVPOS(X, Y)
```

```
20 CALL EXEC
```

```
RETURN
```

```
END
```

NOVAグラフィック用PLOTルーチンでは

```
SUBROUTINE PLOT(X, Y, I)
```

```
IA = I
```

```
IF(IA. EQ. 3) IA = 1
```

```
CALL VCTRG(X, Y, IA)
```

```
RETURN
```

```
END
```

これらのPLOTルーチンをプロットング・システムのPLOTルーチンと置き替えば、XYプロッタ用のプログラムが直ちにそれぞれのディスプ

レイにも使用できる。実際には、XYプロッタ用のPLOTルーチンは表1で示した機能のほかに、いくつかの機能をもっている。また画面の大きさも各々異なるため座標変換等の処理が必要になり、上記のプログラムより複雑なものになる。

3. プログラムとその使用法

ここでは、グラフィック・ディスプレイおよびラインプリンタ上へプロッター形式のプログラムで図形を出力するプログラムの概要とプログラムの使用法について述べる。また、NOVAと中央機FACOM 230-75との交信により、図形データを中央機で発生させ、NOVAディスプレイに出力した応用例を紹介する。なおサブルーチン・プログラムの使用法の説明で、パラメータはその最初の文字がI~Nで始まるものは整数形、それ以外のは単精度実数型である。またパラメータの説明の中で、入力が入力パラメータ、出力は出力パラメータであることを示す。

3.1 グラフィック・ディスプレイのためのプログラムGOPEN

GOPENは、XYプロッタ用のプログラムに何らの変更をも加えずグラフィック・ディスプレイを使える様にしたプログラムである。このプログラムは次のような設計方針で作られた。

- (1) プロットング・システムのベーシックおよびファンクショナル・ルーチンのすべてが使用できる。
- (2) 既存のXYプロッタ用プログラムからも簡単にグラフィック・ディスプレイに出力することができる。
- (3) グラフィック・ディスプレイの取扱い、操作が容易である。
- (4) グラフィック・ディスプレイに表示した図形を、対話形式の操作でXYプロッタへ出力できる。

これらの機能を使い使用者は、すみやかに計算結果を図示でき、またXYプロッタ用プログラムのディバックを効率的に行なうことができる。

本プログラムGOPENは、XYプロッタ・ルーチンとグラフィック・ルーチンの中間に位置している。

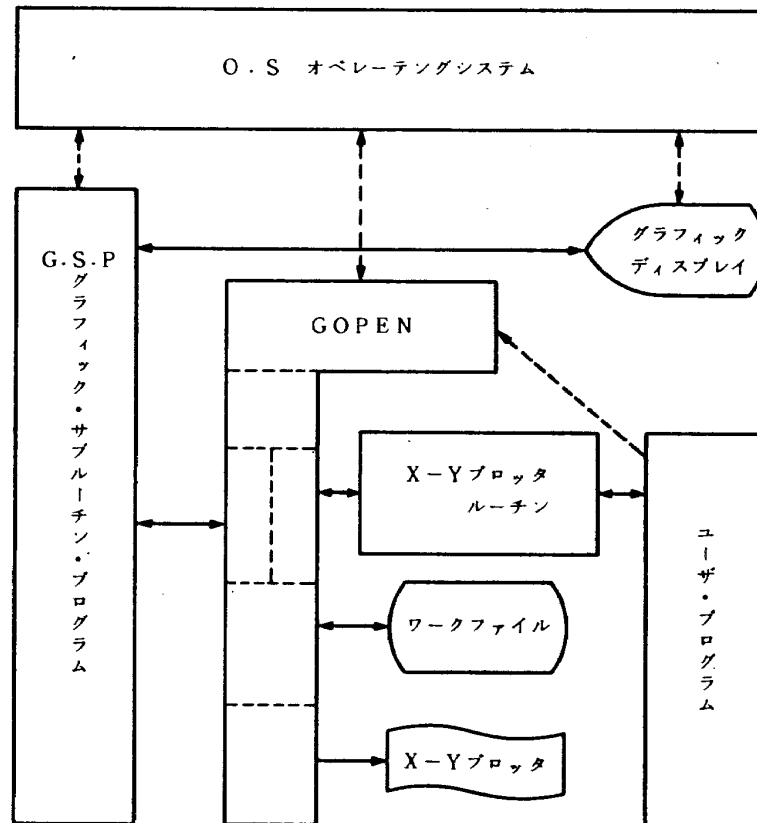


図3 システム概念図

主たる機能は、XYプロッタ・ルーチンで発生した図形データ、パラメータをグラフィック・ルーチンに変換するプログラム・インターフェースの働きをすることである。このGOPENにより、使用者はグラフィック・ルーチンを意識することなく、XYプロッタの知識だけでグラフィック・ディスプレイを使用できる。またグラフィック・ディスプレイの操作盤上のキーボード、スイッチを使い、オンラインでユーザプログラムと対話ができる。

図3にシステムの概念図を示す。

3.1.1 プログラム概要

GOPENは、XYプロッタ・ルーチンの知識とグラフィック・ディスプレイの2種類のキーボード・スイッチ操作のみで、容易にグラフィック・ディスプレイを使うことができるプログラムである。GOPENに入力されるPLOT用コマンドは、ワークファイルに蓄積されると同時に、グラフィック・ディスプレイのためのコマンドに変換される。また使用者から入力されるキーボード・パラメータの解釈を行ない、表示画面の消去、原点の設定、図形データのXYプロッタへの出力の処理を行なう。これらの

処理を行なうため、GOPENは(1)初期設定部、(2)プロッタ・コマンド蓄積部、(3)プロッタコマンド解釈部、(4)グラフィック実行部、(5)XYプロッタ出力部、(6)クローズ処理部の6つの部分から構成されている。各部分を説明する。

(1) 初期設定部

ここでは次の機能をもつ

- a) グラフィックの使用宣言
- b) 作業用領域の定義
- c) グラフィックのプログラム・パラメータの定義
- d) GOPENのパラメータの定義
- e) 使用者指定サイズによる座標変換係数の決定

(2) プロッタ・コマンド蓄積部

プロッタ・ルーチンからのデータ、コマンドを使用者の指定した作業用ボリューム・ワークファイルに格納する。ここに格納されたデータは、使用者がXYプロッタへ出力指定を出したときに使用される。

(3) プロッタ・コマンド解釈部

プロッタ・ルーチンから入力される次のコマンドの解釈、変換処理を行なう。

- a) ペンのアップ, ダウン
- b) ペンの移動
- c) 原点移動
- d) 図形データをグラフィック・ディスプレイ用データへ変換
- e) 終了コマンドの検出
- (4) グラフィック実行部
- (3)で変換されたデータの表示と使用者とのキーボ

- ードによる通信を行ない, 次のような処理を行なう。
- a) グラフィック・ディスプレイへの表示
- b) 表示データの区切を使用者に通告
- c) 使用者からの指令の実行
 - ① 表示画面の消去
 - ② XYプロッタへの出力
- (5) XYプロッタ出力部
- 使用者からのXYプロッタ出力指令によるXYプロ

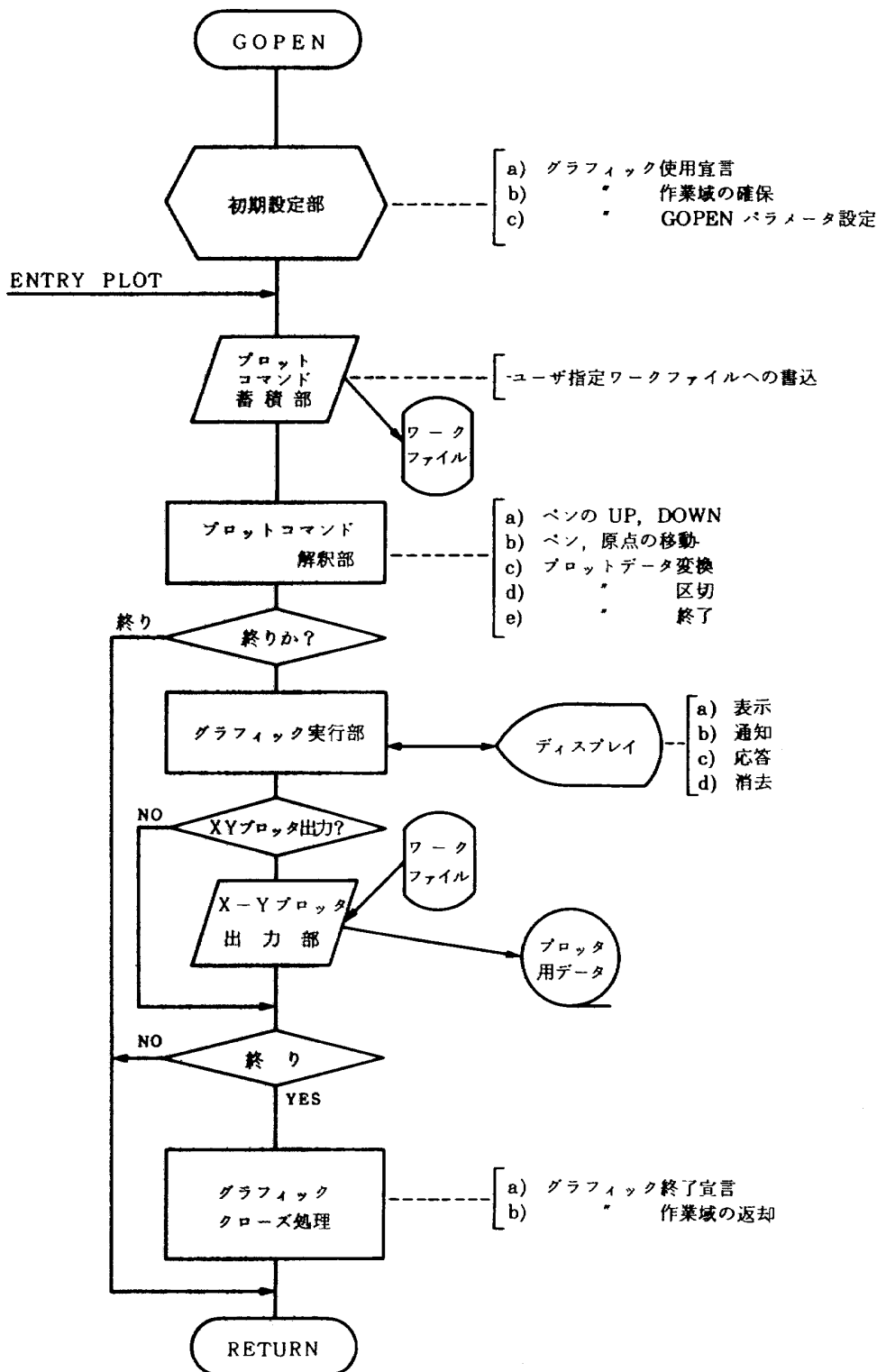


図4 GOPENのフロー概要

ロッタへの出力と作業用ファイルの初期化を行なう。

(6) グラフィック・クローズ処理

- a) グラフィックの使用終了宣言
- b) グラフィック, GOPENの作業域の返却

GOPENのフローの概要を図4に示す。

3.1.2 サブルーチンの説明

使用者はサブルーチンCALLの形式でGOPENにENTRYする。ENTRY名はGOPENとPLOTである。

(1) サブルーチン GOPEN

目的：グラフィック・ディスプレイ使用のための初期設定を行なう。プログラムの始めに必ず呼び出さねばならない。これによりPLOTコマンドはGOPENプログラムと結合される。

呼び出し形式

CALL GOPEN(XL, YL, IWORK, IDV)

パラメータの説明

XL } : 入力, ディスプレイに表示したい図形の
YL } : 入力, 最も大きいX方向, Y方向の長さを指定。

(YLは航技研XYプロッタではNARRで最大30 cm, WIDEで90.0 cmである)

IWORK: 入力, ワークファイルの磁気ディスクあるいは磁気テープの論理機番(1~90)

(レコードサイズ50バイト)

IDV: 入力, XYプロッタ出力機番, 99とする。

(2) サブルーチン PLOT

目的：XYプロッタ用ルーチンPLOTと同機能をもつ。また表示画面に対する処理手続きの依頼待合せとGOPENプログラムの終了手続きを行う。

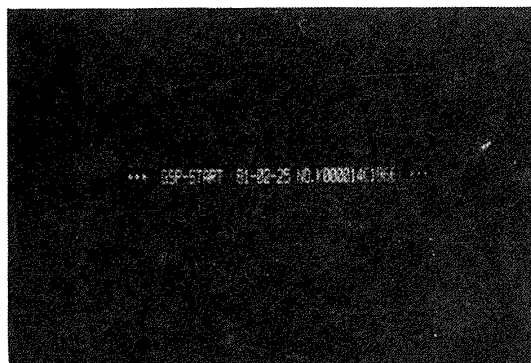
呼び出し形式

CALL PLOT(XL, YL, IPEN)

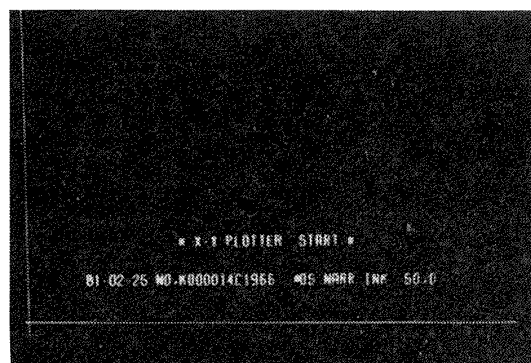
パラメータの説明

XL } : 入力, 現在のポインタの位置から画面上
YL } : 入力, の(XL, YL)まで線を引く。

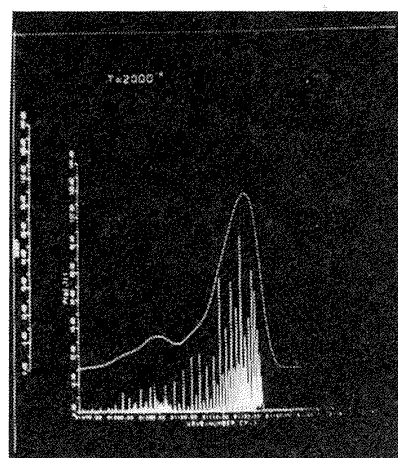
IPEN: 入力, IPEN = -3 ~ 3 のときXYプロッタ用ルーチンPLOTのIPENパラメータと同じ。ただし, 原点移動の



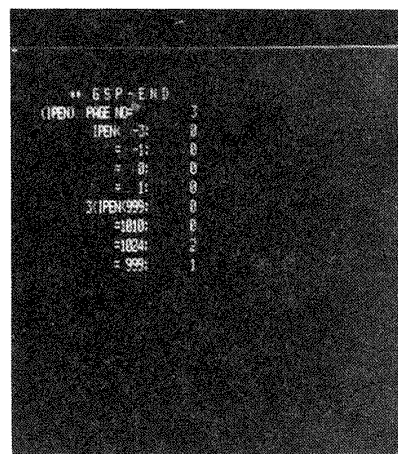
(1)



(2)



(3)



(4)

図5 GOPENによるディスプレイへの表示

-3, -2 のときは, 処理依頼待合せ (表示画面に対する消去と原点の移動を問合せる) をする。

IPEN = 1000, 一枚の作図が終了したことを計算機に知らせる。XYプロッタに出力するかどうかの処理依頼待合せとなる。

IPEN = 999, GOPEN, PLOT
ルーチンの終了。IPEN = 1000 の機能も有する。

(注意) 待合せ事象に入るとグラフィック・ディスプレイのベルが鳴る。使用者はオーバーレイスイッチあるいはファンクション・キーを使いプログラムに指令を与える。IPEN = -3, -2, 999, 1000 のとき待合せ事象となる。

-2, -3 のとき; 原点移動が行なわれるので, 表

示画面を見て, 画面の消去か, そのまま続行かの指定を行なう。

1000 のとき; ディスプレイ上に表示されている図形をXYプロッタに出力するかどうかを指定する。

999 のとき; プログラムが終了したことを使用者に知らせ, 1000 と同じ処理をする。

3.1.3 使用法と使用例

通常のXYプロッタ用の作図プログラムの前にサブルーチンGOPENをCALLすればグラフィック・ディスプレイと結合される。描く図がディスプレイ上に全部表示できる (ディスプレイ上に一度に表示できる図を1ページ分の図と呼ぶことにする) とときは, GOPENをCALLするだけでよい。数ページ分の図形を描くときはページの切目に

CALL PLOT(X, Y, 1000)

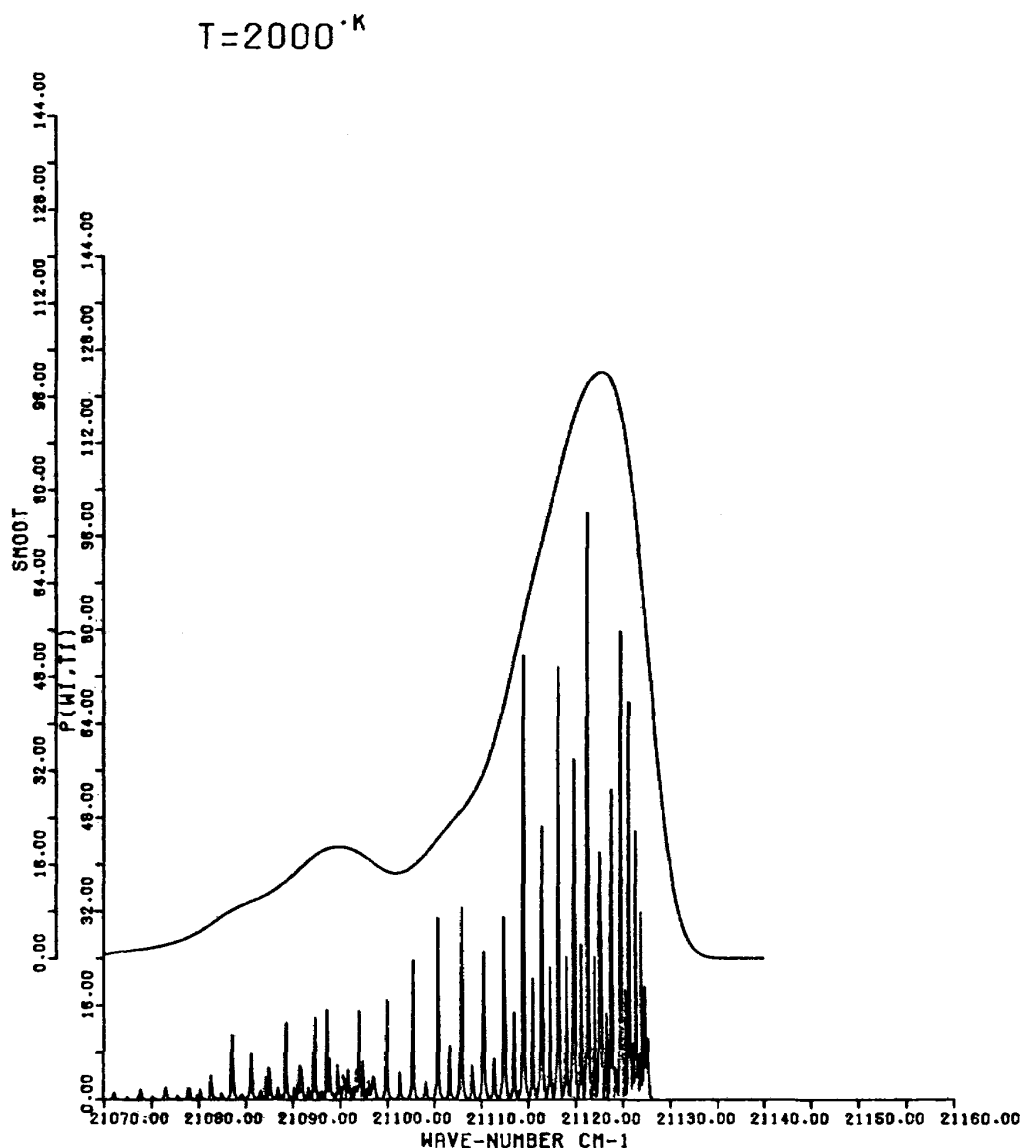


図6 図5-3をXYプロッタに出力したもの

を入れておく。このときの X, Y は意味をもたない。プログラムがランし、GOPEN を実行すると、ディスプレイ上に図 5-(1) が表示され、GOPEN が実行していることを使用者に知らせる。このときブザーも鳴り、計算機が待合せ事象になる。ファンクションキー“1”を入力すると実行を開始する。ここで“2”を入力すると、プログラムは終了してしまう。プログラムが PLOTS を実行すると図 5-(2) が表示され、プロッタ・ルーチンの実行開始を知らせる。PLOT コマンドで原点移動が発せられると待合せ事象となり、使用者の指示を受ける。PLOT のコマンドはオンラインでディスプレイ上に図形を描いて行く。このようにして 1 ページ分の図形が描き終ると、待合せ事象となり、今ディスプレイ上に描いた図形を XY プロッタへ出力するかどうかの指令を待つ。図 5-(3) はディスプレイに表示された 1 ページ分の図形である。また図 6 はこの図形を XY プロッタに出力して得たものである。プログラムが終了すると今迄実行したプログラムの情報を図 5-(4) の形で表示して終る。

図 7 はこの操作手順を示したものである。

具体的な制御カード、プログラムの例は付録 1 に示してある。

3.2 ラインプリンタへの出力用プログラム CPlot

これは 2 次元配列を図形出力の画面とみなし、配列内に線図を描き、この 2 次元配列のイメージをラインプリンタへ出力するためのプログラムである。図 8 のような大きさ (NX, NY) の 2 次元配列名 IPC 内に直線を引く。線の通過している IPC の要素を 1, 通過していない所を 0 とする。この配列 IPC のイメージをラインプリンタに印字する。

プログラムは、2 次元配列 IPC に PLOT ルーチン形式で線分データを格納するルーチン CPlot と、CPlot でデータを書き込まれた IPC をラインプリンタ上へ印字するルーチン LPSCRN からなる。

ルーチン CPlot の機能は、与えられた線分を図 8 のように整数 2 次元アドレスの列に分解することである。この機能は、線図作図処理において最も基本的な処理である。例えば、PLOT ルーチンは、

XY プロッタのための整数 2 次元アドレスを発生させるルーチンである。整数 2 次元アドレスから次の整数 2 次元アドレスへの移動単位をステップ・サイズと呼ぶ。ステップ・サイズの最も簡単なものは、上下左右のいずれかの方向 (4 方向) へ 1 単位移動するものである。8 方向のステップ・サイズは 4 方向のものにさらに 45° の移動を許したものである。図 9 に 8 方向のステップ・サイズを示す。16 方向、

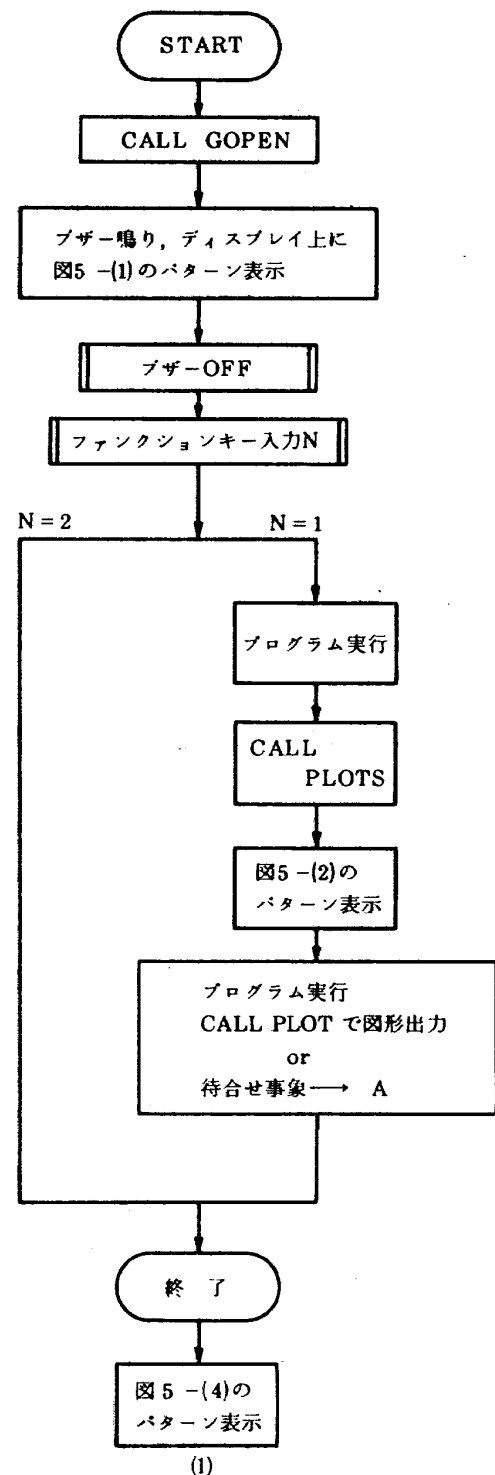


図 7 操作手順 [] は使用者入力を示す

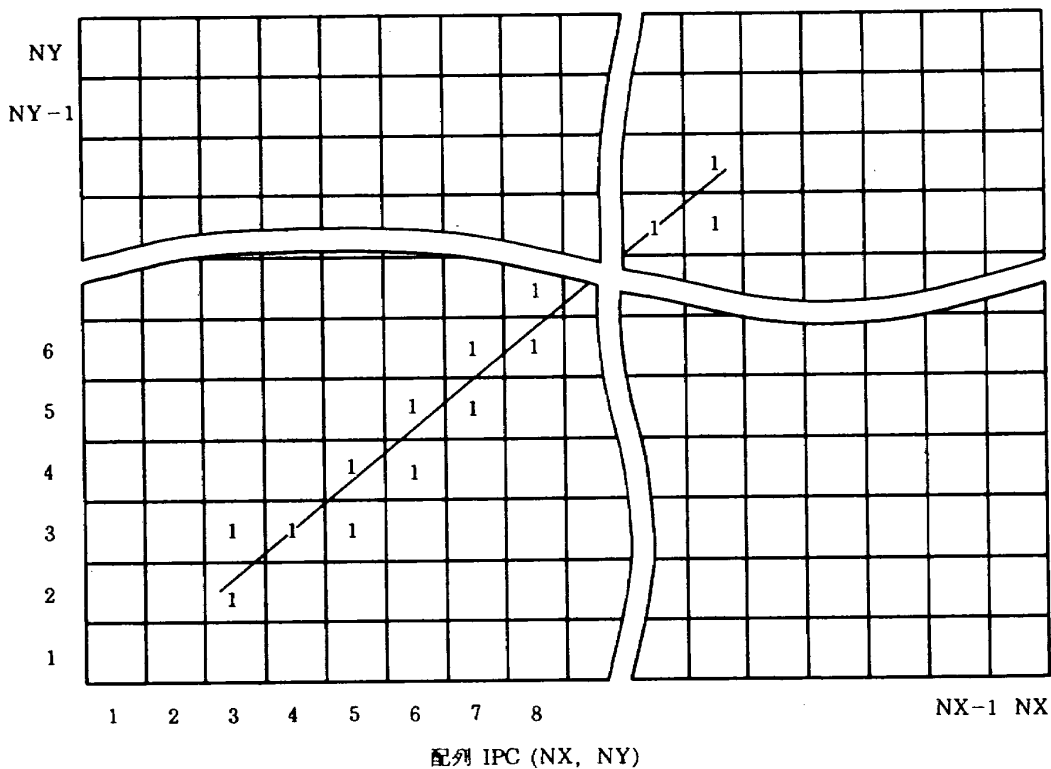
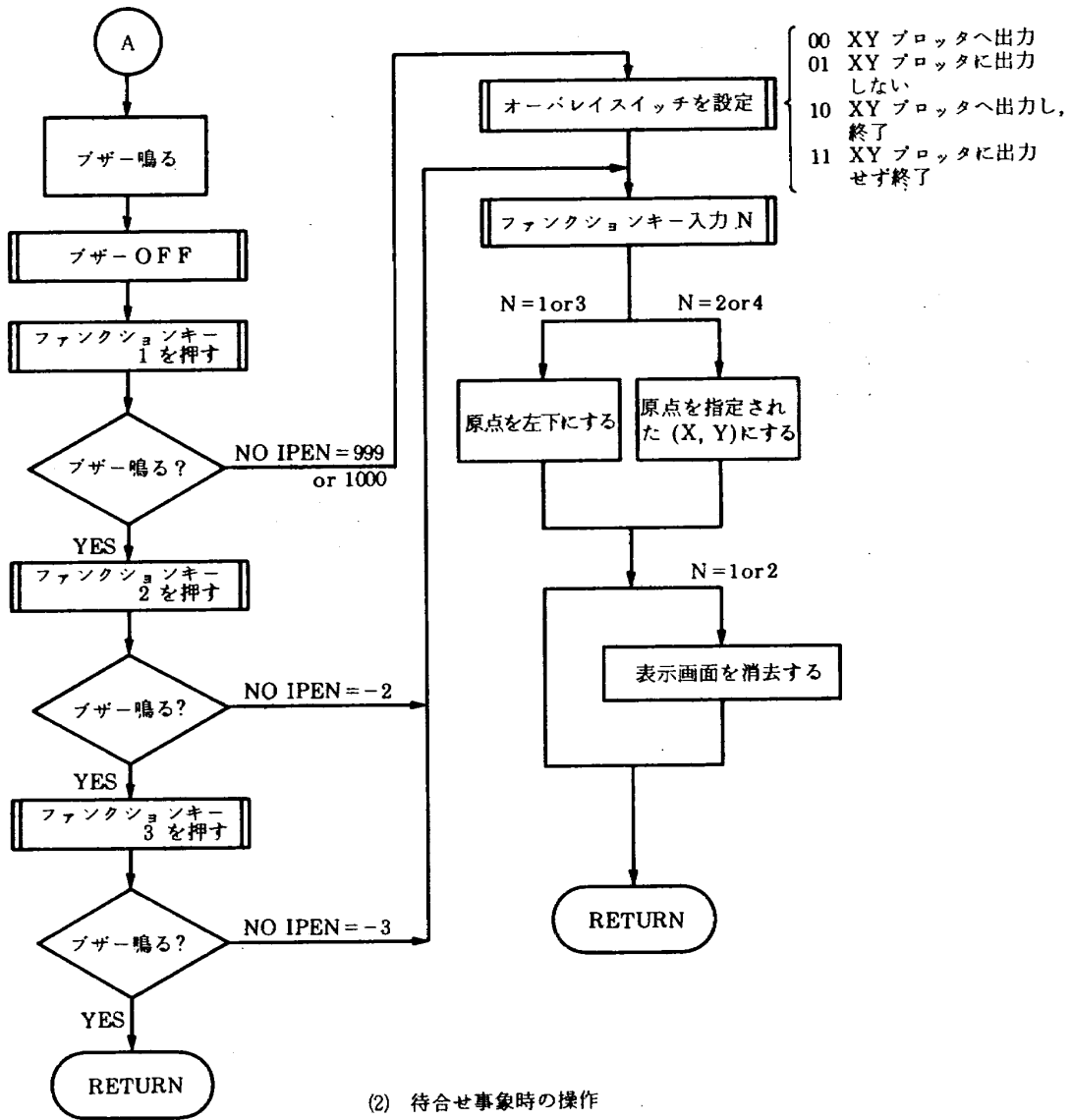


図8 2次元配列に線を引く、線の通る要素を1とする

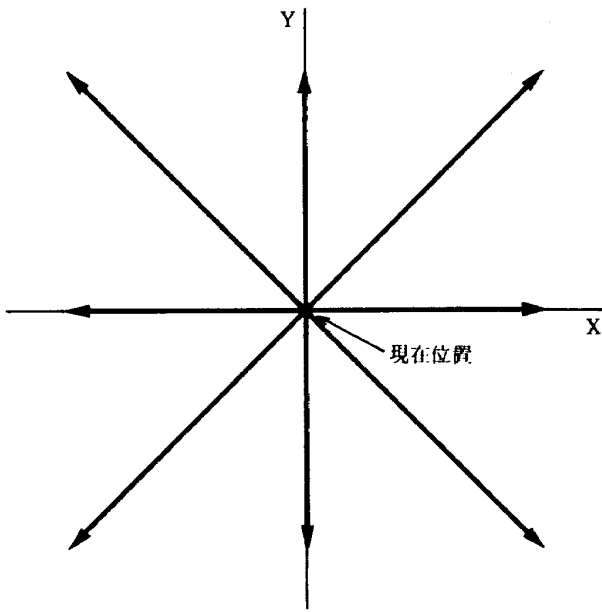


図 9 8方向のステップサイズ

24 方向のステップ・サイズのもの考えられている。CPLOTでは、4方向、8方向の両方のステップ・サイズをもつ。8方向の場合の処理フローを図10に示す。ここでは、1ステップでの線分のx方向、y方向の増加分 Δx 、 Δy を最初に求めておき、順次 Δx 、 Δy を加算し線分をたどる。このようにして求めた値を整数化し整数2次元アドレスを決定する。ここで $|\Delta x|$ 、 $|\Delta y| \leq 1.0$ でかつ $|\Delta x|$ 、 $|\Delta y|$ のいずれか一方は1.0になるように Δx 、 Δy を決める。

3.2.1 サブルーチンの説明

(1) サブルーチン CPLOTS

目的：CPLOTの初期値を設定する。

呼び出し形式

CALL CPLOTS(IPC, NX, NY, X0, Y0, WX, WY, MD)

パラメータの説明

IPC：出力、線がかかれる2次元配列名
全要素に0がセットされる。

$\left. \begin{matrix} NX \\ NY \end{matrix} \right\}$ ：入力、IPCの大きさ、IPC(NX, NY)

$\left. \begin{matrix} X0 \\ Y0 \end{matrix} \right\}$ ：入力、IPC上に描こうとする図形の左下の座標値、 $(X0, Y0)$ の点がIPC上ではIPC(1, 1)になる。

WX：入力、図形のX方向の長さ、 $(X0 + WX,$

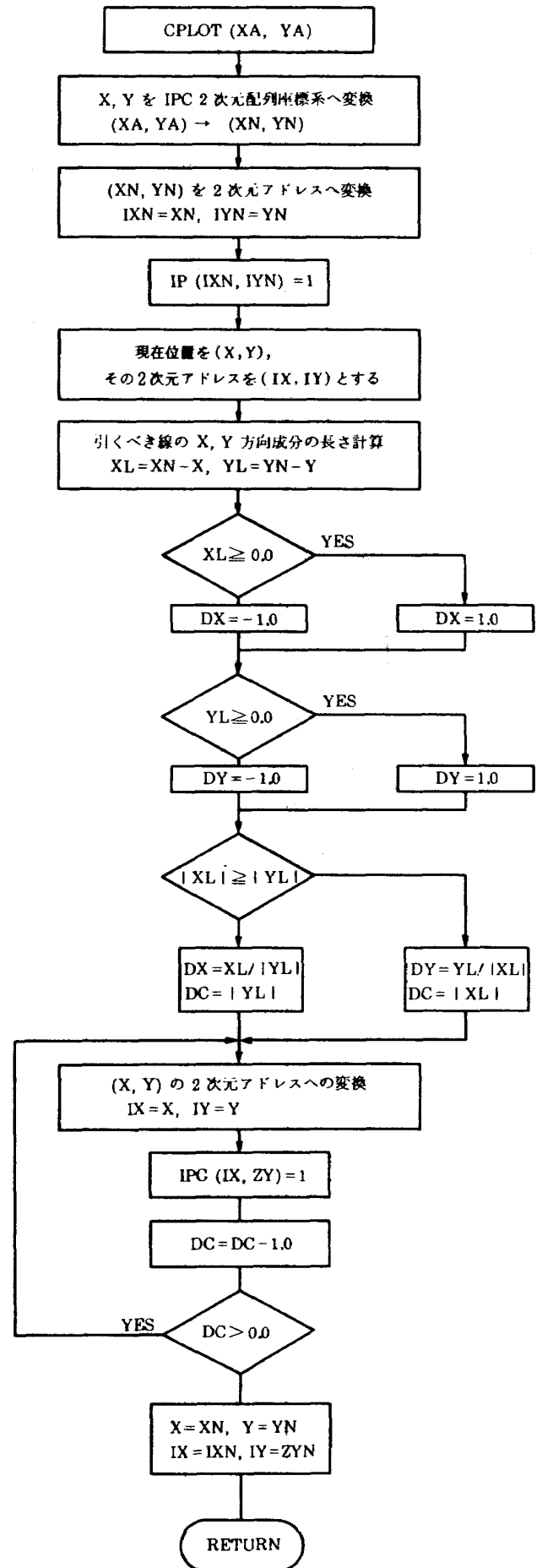


図 10 CPLOT のフローチャート (8方向)

Y0)がIPC(NX,1)になる。

WY : 入力, 図形のY方向の長さ, (X0, Y0 + WY)がIPC(1, NY)になる。

MD : 入力, ステップ・サイズの種類指定

MD=0 8方向

MD=1 4方向

(2) サブルーチン CPLOT

目的: 整数型2次元配列上にXYプロッタ用ルーチンPLOTと同様な線を引く。線の通る配列要素には1がセットされる。

呼び出し形式

CALL CPLOT(X, Y, IW)

パラメータの説明

X } : 入力, 現在の点から(X, Y)まで線を引く。
Y }

IW : 入力, |IW|=2 線を引く
|IW|≠2 線を引かず(X, Y)へ移動

IW<0 (X, Y)へ移動後原点を(X, Y)へ移す。

注意: このルーチンを使用する前に必ずCPLOTSをCALLしておくこと。

(3) サブルーチン CPLOTB

目的: CPLOTSルーチンで指定した画面領域の2次元配列をビット単位で利用する。したがって, このルーチンをCALLすることによりメモリが36倍(FACOM230-75は1語36ビット)有効に使用できる。これはメモリ内に線画の画面を生成する場合, 便利に使用できる。

このルーチンとCPLOTおよびプロットング・システムを使用すれば, たとえば1024×1024の画素をもつ線画を29(1024/36)×1024のメモリに容易に作ることができる。

呼び出し形式

CALL CPLOTB(NXB)

パラメータの説明

NXB: 入力, メモリ画面のX方向の画素数,
0 < NXB ≤ 36 * NX

NXB=0のとき語処理(1画素1語)

注意: このルーチンはCPLOTをCALLする前にCALLしておくこと。

(4) サブルーチン LPSCRN

目的: 整数型2次元配列ICP(NX, NY)の内容をラインプリンタ上にNXカラム, NY行に印字する。ICPの要素の内容が0のときはブランク, 0でないときは指定した一文字が印字される。またICPの値が1から15までのときは, 濃淡で印字することもできる。

呼び出し形式

CALL LPSCRN(ICP, NX, NY, IBC)

パラメータの説明

ICP: 入力, 印字する2次元配列名

NX } : 入力, ICPの大きさ, ICP(NX, NY)
NY }

IBC: 入力, 印字文字を指定

印字文字をブランクにするとIPCを濃淡パターンとみなし, 1のときをブランクとして15まで順次に重ね印字で濃淡パターンを作ってゆく。このときIPCの値が1以下のときは1, 15以上のときは15とする。

注意: NXが128を越えたときは128カラムまでを印字する。

3.2.2 使用例

CPLOTを使用し, ラインプリンタ上にプロットング・システムにより作図する。図11はプロットング・システムのSYMBOLルーチンにより英数字を書かせたものである。図11-(1)は8方向のステップ・サイズで, 図11-(2)は4方向で描いたものである。図12は簡単なグラフをLINE, AXISルーチンを使って描いたものである。(図11, 図12のプログラム・リストは付録2に示してある。)

3.3 NOVAグラフィック・ディスプレイ

データ処理設備のNOVAのグラフィック・ディスプレイでは2.4で示したNOVAグラフィック・用PLOTルーチンを作れば, PLOT形式で作図することができる。NOVAのFORTRANは中央機FACOM230-75のFORTRANと完全な互換性がなく, 中央機でのプログラムをNOVAで使用することはほとんどない。したがってNOVAグラフィック

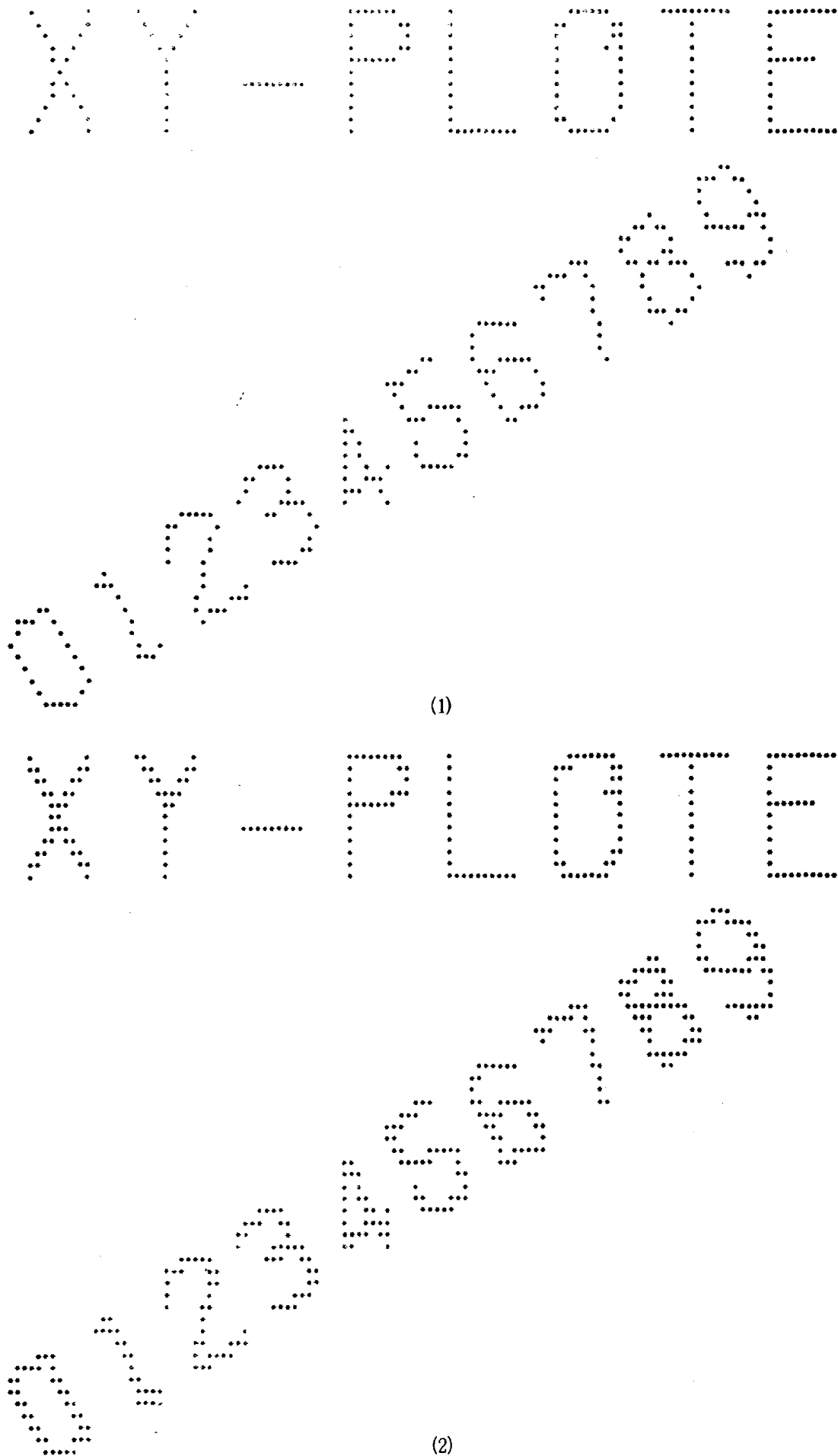


図 11 SYMBOL ルーチンを使い英数字を書く, (1)はステップサイズ 8 方向, (2)は 4 方向

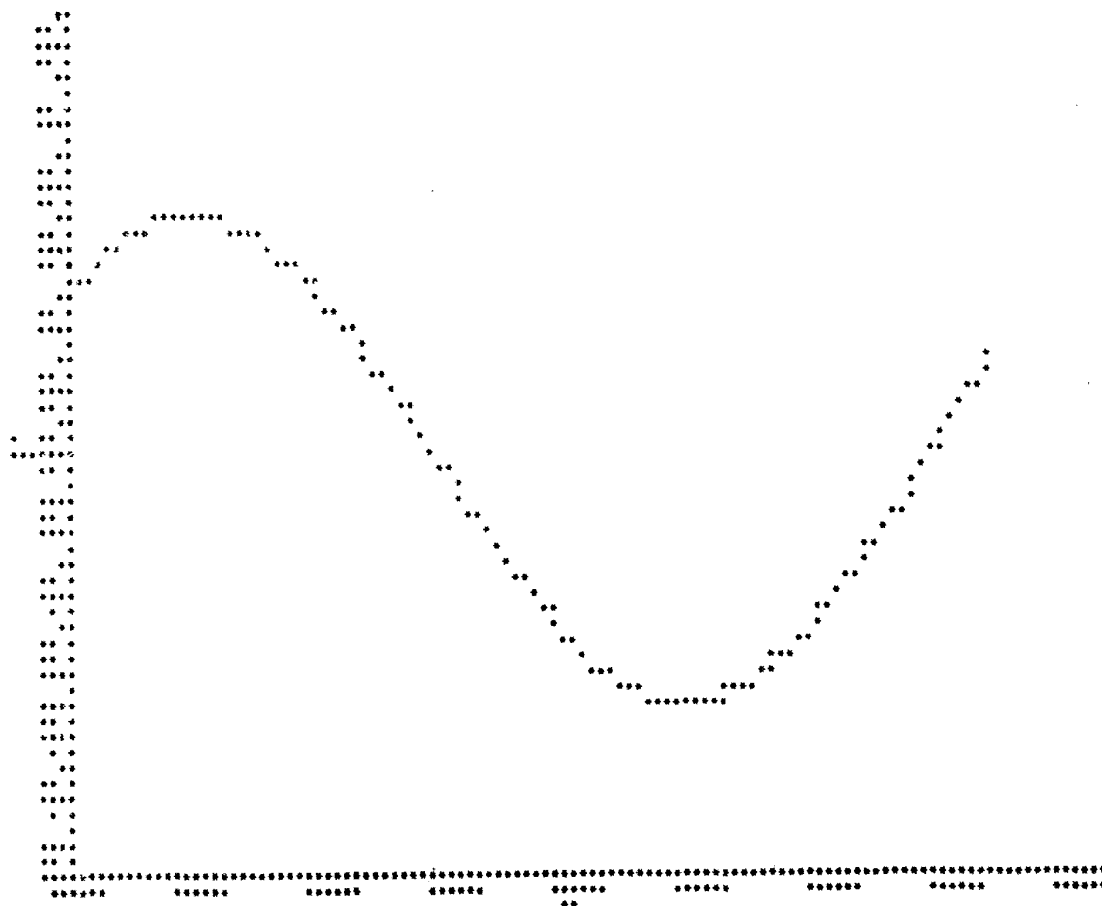


図 12 LINE, AXIS ルーチンを使いグラフを描かせる。

ック・ディスプレイをプロットング・システムと同じような仕様にする必要はそれほどない。

ここでは、図形データを中央機のプロットング・システムを使い発生させ、この図形データをNOVAグラフィック・ディスプレイへ出力する方法について述べる。中央機とNOVAは、チャンネル結合でむすばれている。NOVAのファイルと中央機のファイルのデータは、データ転送プログラムCCA⁷⁾を介し、お互に交換できる。中央機で発生した図形データをいったん、中央機のディスクに格納し、このデータをCCAによりNOVAのファイルに転送する。NOVAではこの転送されたデータをグラフィックへ出力する。このとき中央機で作る図形データは、次のような2通りが考えられる。

1) ダミーのPLOTルーチンを作り、PLOTデータ(X, Y, I)をトラップし、Xを0~1023の整数IX, Yを0~779の整数IYに変換したデータ(IX, IY, I)のデータ列を作る。

2) ルーチンCPLOTを用い、図形データを1024×780ビット、24960語^{注)}のメモリ内に描く。この24,960語をNOVAに転送する。

2)の方法では、図形の複雑さに無関係に約23K語のデータを転送する必要がある。単純な図形の場合は、1)の方が有利である。

中央機で図形データを発生させ、NOVAグラフィック・ディスプレイ上に図形を表示させた応用例として、コンピュータアニメーションを製作した例を紹介する。これは小惑星ミッション⁸⁾をアニメーション化したもので、小惑星の運動を示し、地球から発射された宇宙船がいつ、どのような小惑星にどのようにして接近してゆくかをシミュレーションする。小惑星の位置、宇宙船の位置を中央機で計算し、プロットング・システムで図形データを発生、データファイルに格納する。このデータをCCAで

(注) NOVA(1語16ビット)との関係上1語32ビットで計算)

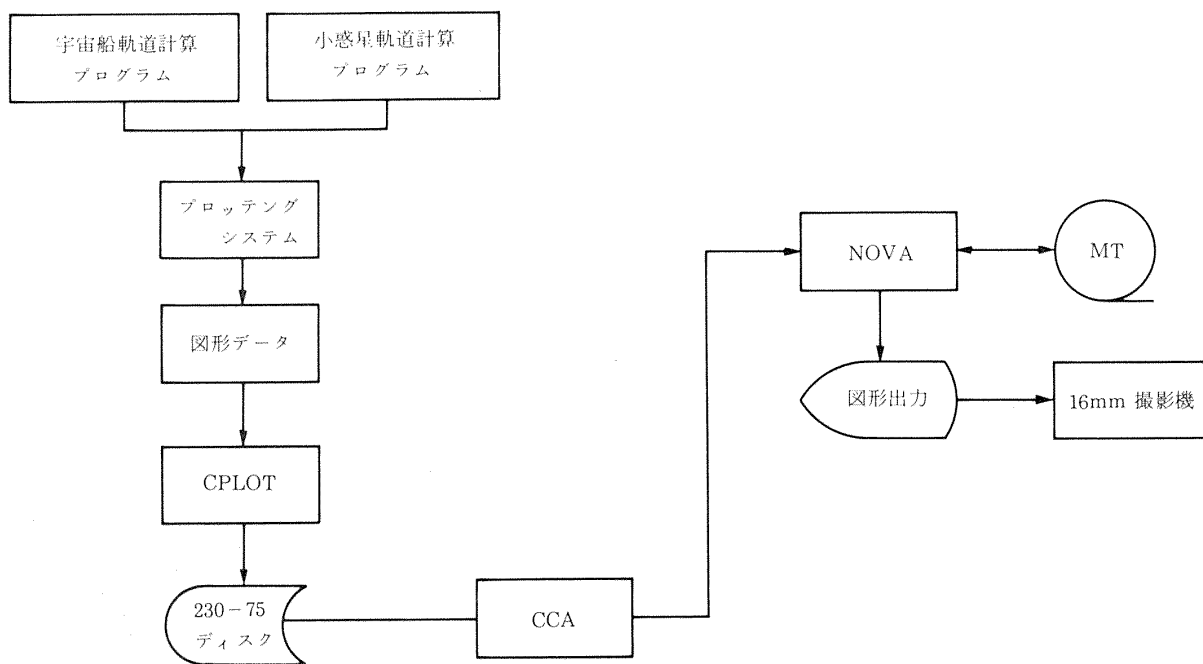
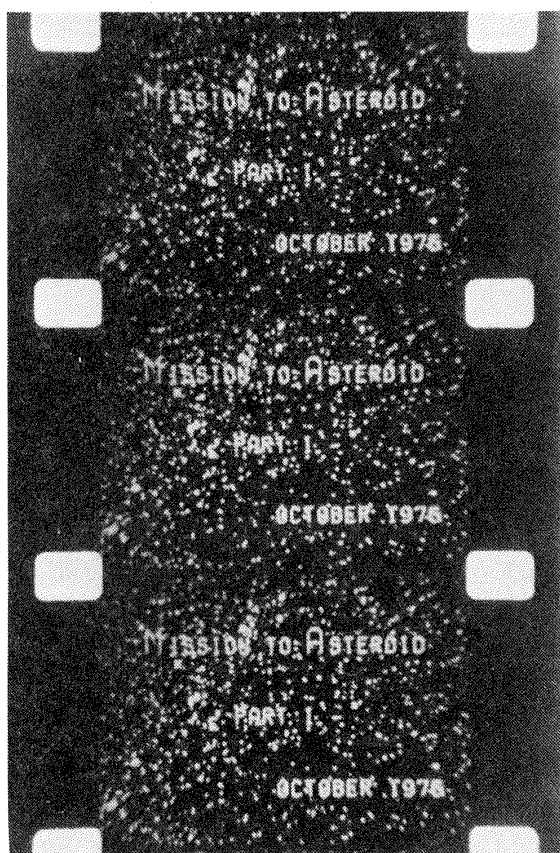
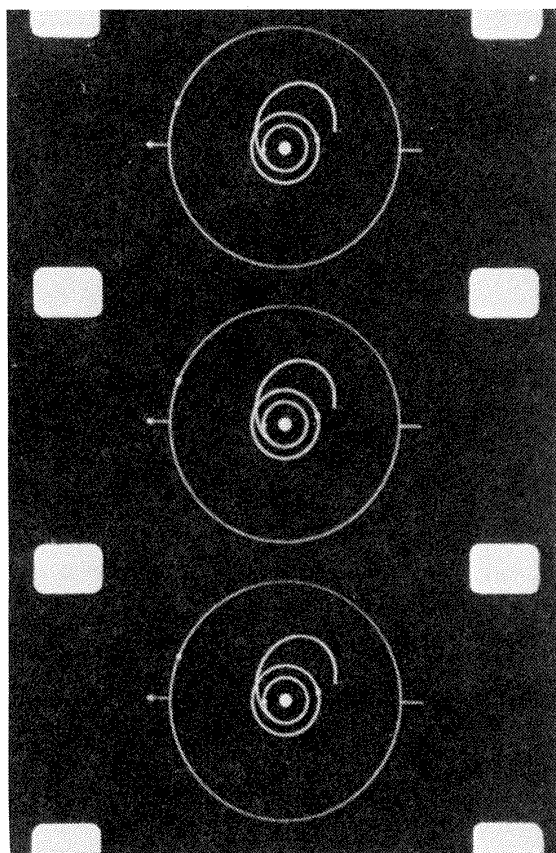


図 13 アニメーション製作ブロック図



(1)



(2)

図 14 中央機で図形データを発生させ、NOVA グラフィックに表示させて製作したアニメーション・フィルムの一部

NOVA磁気テープへ転送する。転送された磁気テープから一面分のデータを読み出し、グラフィック上に図形を表示する。表示された図形は1コマ1コマ16mm撮影機で撮影し、アニメーションを作る。図13にアニメーション製作のブロック図を示す。図14はこのようにして撮影したアニメーション・フィルムの一部である。図14-(1)は上記2)の方法で、図14-(2)は1)の方法で図形データを作り、NOVAに転送したものである。なおアニメーション製作にNOVAグラフィック・ディスプレイを使用したのは、

- 1) 1コマずつのコマ取りのためグラフィック・ディスプレイの使用時間が延べ十数時間に及び、
 - 2) NOVAグラフィック・ディスプレイのほうが中央機のグラフィック・ディスプレイに比べ、図形が安定し鮮明である、
- という理由による。

4. 線画とその出力

XYプロッタは線画を描くための装置である。前章までで述べたように、XYプロッタに限らずグラフィック・ディスプレイ等の装置にも、XYプロッタ用のプロットング・システムで線画を出力することができることを明らかにした。ここでは線画はコンピュータ内でどのように表現されるかを考え、線画を図形出力装置に出力するためにはどのような機能を備える必要があるかを検討する。

4.1 デジタル化された線画

線画を次のように定義する。

有限長の長さをもついく本かの曲線といくつかの点から構成される図形。

ある曲線 l は媒介変数 t を使って

$$\begin{cases} x = \varphi(t) \\ y = \phi(t) \end{cases} \quad \alpha \leq t \leq \beta$$

と表現できる。

ここで閉区間 $\alpha \leq t \leq \beta$ から、 α, β を含む n 個の点をサンプリングする。

$$\alpha = t_1 < t_2 < \dots < t_n = \beta$$

このサンプリングされた t によって決まる点列 $(\varphi(t_1), \phi(t_1)), (\varphi(t_2), \phi(t_2)), \dots, (\varphi(t_n), \phi(t_n))$

$\phi(t_n)$ を順次直線でむすんでできる折線をデジタル化された曲線と呼び、 L_n と書く。 L_n の集合 $\{L_n\}$ と点の集合 $\{P\}$ で構成される図形をデジタル化された線画と呼ぶ。

4.2 線画図形出力装置のモデル化と線画の出力

デジタル化された線画図形を出力するための装置に最少限必要な機能は、次の2点である。

- (1) 画面上に $i \times j$ の格子点で代表される2次元番地が付けられている。
- (2) ある任意の格子 (X_a, Y_a) から任意の格子 (X_b, Y_b) へ近似的な直線を引くことが可能である。

(2)の機能はハードウェアでも、ソフトウェアでも実現できる。3.2で述べたCPLOTルーチンは、この機能をソフトウェアで行なった一例である。

さて、このモデル化された線画図形出力装置にデジタル化された曲線 L_n を描かせる。 L_n は n 個の点列からできているとする。この手順は、

- (1) $k=1$ とする。
- (2) k 番目、 $k+1$ 番目の点 $P_k(X_k, Y_k), P_{k+1}(X_{k+1}, Y_{k+1})$ をとりだし、これらの点と画面上の格子点との対応付けを行なう。ここで求まる格子点を P'_k, P'_{k+1} とする。
- (3) P'_k と P'_{k+1} 間を直線でむすぶ。
- (4) $k=k+1$
- (5) k が n より小さければ(2)へ、 $k=n$ なら終了である。

点 P を表示するには、 P に対応する画面上の格子点へ点を打てばよい。このようにして、デジタル化された線画を図形出力装置に出力することができる。出力される図形の精度は画面の格子 $i \times j$ の大きさにより決定される。

4.3 出力装置へのインターフェイス・ルーチン

上記線画出力の手順(2)で $k > 1$ の場合、点 P_k と格子点との対応付けは、 $k=k-1$ ですで行なわれているので、その時の結果を利用すれば、あらたに対応付けをする必要がない。この考え方で、プロットング・システムの“PLOTルーチン”が作られている。このようにすると $k=1$ の場合と、 $k > 1$ の場合に分けて処理する必要が生ずる。 $k=1$ の

処理を PLOT ルーチンではペン・アップとし、 $k > 1$ のときはペン・ダウンとし、一つの PLOT ルーチンで処理を行なっている。しかし、このためペンをアップするか、ダウンするかの指定をするパラメータが必要となる。またプログラマーは、ペンの位置が現在どこにあるかを常に意識している必要がある。^{注1)}

さて、線画の構造は、4.1 で述べたように曲線と点より構成されている。したがって1本の曲線およびいく点かの点の単位で出力する方がより能率的である。1本の曲線単位で出力すれば、プログラマーは $k=1$ の場合とか、 $k > 1$ の場合とかを意識する必要がなくなる。

ここで曲線と点を描くルーチン^{注2)}を加えて、次のような基本的インターフェイス・ルーチンを考える。

(1) PLOT ルーチン

機能：現在のプロットング・システムの PLOT ルーチンと同じ。

(2) PLOTL ルーチン

機能：デジタル化された曲線を出力する。

(3) PLOTP ルーチン

機能：点の集合を出力する。

(2), (3) のルーチンは(1)のルーチンを使って簡単に作ることができ、また(1)のルーチンは(2)のルーチンで書き直すこともできる。

この3つを最も基本なルーチンとし、これらを使って線図作図ソフトウェアを構築する。各図形出力装置は、この3つの基本ルーチンとインターフェイスを取る。プログラマーは基本ルーチンのみを考慮し、プログラムを作成すればよい。このようにしてできたプログラムは、ハードウェアにインデpendなものとなり、プログラムのメンテナンス、蓄積に多大な効果をもたらすであろう。

5. むすび

現在計算センタで使用されている XY プロッタ以外の線画図形出力装置にも、XY プロッタ用ソフト

ウェアのプロットング・システムをそのままで使用するにはどのようにしたらよいか、を考察した。その結果、各装置とプロットング・システムのインターフェイスとして PLOT 形式のインターフェイス・プログラムを用意すればよいということが分った。そして、グラフィック・ディスプレイおよびラインプリンタ用の PLOT 形式のインターフェイス・プログラムを製作し、その有効性を確かめた。グラフィック・ディスプレイでは、CRT 上に表示された図形を XY プロッタへ出力する機能をも備えている。この機能を使えば、XY プロッタ・ジョブでは、ジョブ終了後、XY プロッタへの図形データを、XY プロッタへ出力する前に、CRT 上でモニターを行ない、不要なものはキャンセルし、本当に有効な図形だけを XY プロッタへ出力するという計算機運用システムを作ることができる。

また線画および線画図形出力装置の一般化したモデル化を試みた。それにもとづいて、ハードウェアに依存しない一般的な線画作図ソフトウェアの構築法も考察してみた。今後、この方法で航技研計算センタ独自のハードウェアに取らわれない線画作図ソフトウェアを作り上げて行きたいと考えている。

文 献

- 1) 大島, 佐藤, 島田; プロットデータの標準化に関する一試案, 昭和 54 年度情報処理学会第 20 回全国大会講演論文集 pp. 567 ~ 568
- 2) 木村, 笹島; XY プロッタ用言語 (PLOTTRAN) とその処理系, 昭和 45 年度情報処理学会第 11 回全国大会講演論文集 pp. 251 ~ 252
- 3) 水島, 坂田; プロッタ用言語の試作, 昭和 45 年度情報処理学会第 11 回全国大会講演論文集 pp. 257 ~ 258
- 4) 吉沢ビジネス・マシンズ; ソフトウェア・マニュアル CALCOMP PLOTTER プログラミング, 昭和 48 年 4 月
- 5) 富士通 (株); FACOM GRASP 文法書, 昭和 50 年 4 月

注1) プロットング・システムではペンの現在位置を求めるためのルーチン WHERE が用意されている。

注2) GRASP は一部この考え方により、システムが作られている。

- 6) タケダ理研(株) ; Software Manual 4014グラフィック・ディスプレイ・ターミナル・ライブラリ解説書
- 7) 航技研計算センター ; CCA利用手引書, 昭和51年4月
- 8) 松島, 磯部 ; 小惑星へのミッションの検討, 第20回宇宙科学技術連合講演会講演集, p28-31 1976

付録1 GOPENの使い方

GOPENを使いグラフィック・ディスプレイ上にXYプロッタ用プログラムを表示するときの、制御カード配列をリスト1に示す。現在GOPENは、リロケータブルな形式のプログラム・ライブラリーとして、ファイル名“K000014.RBSL”に格納されている。したがって、GOPENを使用するとき、このファイル名を指定して、実行可能なプログラムを編集する必要がある。このファイルにはXYプロッタへ出力するルーチンも含まれているので、計算センターのプログラム・ライブラリーの中に入っている本来のPLOTルーチンはロードされない。この操作は、リスト1の%LIEDで総て行なわれる。また%XYはXYプロッタを、%GDはグラフィック・ディスプレイを使用するための制御文である。GOPENでは論理番号01のワークファイルを使用する。このファイルを確保するための制御文が%DISKWKである。

```

%NO      K000014.ABCD
%KJOB    CCLS
%FORTRAN

      .
      CALL GOPEN (30.0,28.0,1.99)
      CALL PLOTS (IP,JP,KP,50.0)
      .
      CALL SYMBOL( . . . . )
      .
      CALL PLOT  (0.0,0.0,1000)
      .
      CALL PLOT  (0.0,0.0,999)
      STOP
      END
%LIED    SL=ON,LF=K000014.RBSL,LV=USER01
%RUN

%XY
%GD
%DISKWK  F01,WF=WORK,RCDSIZE=50
%JEND

```

リスト1 GOPENのプログラム構成

付録2 CPLOTの使い方

CPLOTは“PLOT”形式でラインプリンタ上に図形を描くルーチンである。ユーザプログラムの中で出されている図形を描くコマンドは

```
CALL PLOT
```

である。したがってこれを

```
CALL CPLOT
```

に変更する必要がある。これにはダミーのPLOTルーチンを作り、この中でCPLOTをCALLすればよい。ダミーのPLOTルーチンをリスト2に示す。

リスト3は図11を作図するプログラムである。

ステップサイズ8方向のCPLOTルーチンのリストをリスト4に示す。

リスト5は図12を作図するプログラムである。このプログラムではCPLOTにリスト4を使用した。

図12をXYプロッタに出力してみる。リスト5の12と18番目のステートメントをそれぞれ

```
CALL PLOTS
```

```
CALL PLOT(X, Y, 999)
```

に替え、ダミーのPLOTルーチンを取除けばよい。

付図1はXYプロッタに出力されたグラフである。

このときのプログラムをリスト6に示す。

```

1      DIMENSION IP(120,60)
2      NX=120
3      NY=60
4      DO 600 L=1,2
5      IND=L-1
6      WRITE(6,501)
7      501 FORMAT(1H1)
8      CALL CPLOTS(IP,NX,NY,0.0,0.0,24.0,18.0,IND)
9      CALL SYMBOL(1.0,15.0,3.0,10HXY-PLOTTER ,0.0,10)
10     CALL SYMBOL(1.5,0.0,2.5,10H0123456789,30.0,10)
11     CALL LPSCRN(IP,NX,NY,1H*)
12     600 CONTINUE
13     STOP
14     END

```

リスト3 図11作図プログラム

```

1      SUBROUTINE CPLOT(XA,YA,IW)
2      DIMENSION IPC(NX,NY)
3      DATA @/1.0/
4      XNOW=(XA-X0)*DLX+1.5
5      YNOW=(YA-Y0)*DLY+1.5
6      IXNOW=XNOW
7      IYNOW=YNOW
8      IF(IW.NE.2) GO TO 9
9      IF(IXNOW.LT.1) GO TO 8

```

リスト4-1

付録3 デバッグ用PLOTルーチン

プロットング・システムを使って作ったプログラムのデバッグ用ルーチンのリストをリスト7に示す。

このルーチンは、プロットング・システムの

```
PLOTS
```

```
PLOT
```

```
SYMBOL
```

```
AXIS
```

```
FACTOR
```

```
WHERE
```

```
NEWPEN
```

ルーチンがCALLされると、そのルーチンの引数の値をラインプリンタ上に印字するものである。

リスト6のプログラムに、このデバッグ用ルーチンを付けて実行すると付図2のようなプロット情報が得られる。

```

1      SUBROUTINE PLOT(X,Y,IP)
2      CALL CPLOT(X,Y,IP)
3      RETURN
4      END

```

リスト2 PLOT(ダミー)

```

10      IF(IXNOW.GT.NX) GO TO 8
11      IF(IYNOW.LT.1) GO TO 8
12      IF(IYNOW.GT.NY) GO TO 8
13      IPC(IXNOW,IYNOW)=1
14      8 IF(IXA.NE.IXNOW) GO TO 10
15      IF(IYA.EQ.IYNOW) GO TO 9
16      10 DX=XNOW-X
17      DY=YNOW-Y
18      IF(DX.GE.0.0) GO TO 11
19      ADX=-DX
20      DX1=-0
21      GO TO 12
22      11 ADX=DX
23      DX1=0
24      12 IF(DY.GE.0.0) GO TO 13
25      ADY=-DY
26      DY1=-0
27      GO TO 14
28      13 ADY=DY
29      DY1=0
30      14 IF(ADX.GE.ADY) GO TO 15
31      DX=ADX/ADY
32      IF(DX1.LT.0.0) DX=-DX
33      DC=ADY
34      DY=DY1
35      GO TO 23
36      15 DY=ADY/ADX
37      IF(DY1.LT.0.0) DY=-DY
38      DC=ADX
39      DX=DX1
40      23 IX=X
41      IY=Y
42      IF(IX.LT.1) GO TO 24
43      IF(IX.GT.NX) GO TO 24
44      IF(IY.LT.1) GO TO 24
45      IF(IY.GT.NY) GO TO 24
46      IPC(IX,IY)=1
47      24 DC=DC-0
48      IF(DC.LE.0.0) GO TO 9
49      X=X+DX
50      Y=Y+DY
51      GO TO 23
52      9 X=XNOW
53      Y=YNOW
54      IXA=IXNOW
55      IYA=IYNOW
56      IF(IW.GT.0) GO TO 123
57      X0=X
58      Y0=Y
59      GO TO 123
60      ENTRY      CPLOTS(IPC,NX ,NY ,X0H,Y0H,DLXH,DLYH)
61      XU=X0H
62      YU=Y0H
63      DLX=FLOAT(NX-1)/DLXH
64      DLY=FLOAT(NY-1)/DLYH
65      X=1.5
66      Y=1.5
67      IXA=1
68      IYA=1
69      DO 610 I=1,NX
70      DO 611 J=1,NY
71      IPC(I,J)=0
72      611 CONTINUE
73      610 CONTINUE
74      123 RETURN
75      END

```

```

1      DIMENSION XD(62),YD(62)
2      DIMENSION IP(120,60)
3      NX=120
4      NY=60
5      DEL=0.1
6      X=-0.1
7      DO 600 I=1,60
8      X=X+DEL
9      XD(I)=X
10     YD(I)=SIN(X)+COS(X)
11     600 CONTINUE
12     CALL CPLOTS(IP,NX,NY,-1.5,-1.5,18.,27.)
13     CALL SCALE (XD,16.0,60,1)
14     CALL SCALE(YD,25.0,60,1)
15     CALL AXIS(0.,0.,1HX,-1,16.,0.,XD(61),XD(62))
16     CALL AXIS(0,0,0.,1HY,1,25.,90.,YD(61),YD(62))
17     CALL LINE (XD,YD,60,1,0,0)
18     CALL LPSCRN(IP,NX,NY,1H*)
19     STOP
20     END

```

リスト 5

```

1      DIMENSION XD(62),YD(62)
2      DIMENSION IP(120,60)
3      NX=120
4      NY=60
5      DEL=0.1
6      X=-0.1
7      DO 600 I=1,60
8      X=X+DEL
9      XD(I)=X
10     YD(I)=SIN(X)+COS(X)
11     600 CONTINUE
12     C  CALL CPLOTS(IP,NX,NY,-1.5,-1.5,18.,27.)
13     CALL PLOTS(1,0,-1,100,0)
14     CALL SCALE (XD,16.0,60,1)
15     CALL SCALE(YD,25.0,60,1)
16     CALL AXIS(0.,0.,1HX,-1,16.,0.,XD(61),XD(62))
17     CALL AXIS(0,0,0.,1HY,1,25.,90.,YD(61),YD(62))
18     CALL LINE (XD,YD,60,1,0,0)
19     C  CALL LPSCRN(IP,NX,NY,1H*)
20     CALL PLOT(0,0,0,0,999)
21     STOP
22     END

```

リスト 6

```

1      C  SUBROUTINE PLOT(X,Y,I)
2      SUBROUTINE PLOT(X,Y,I,PLOT)
3      DATA IB,FACT,XO,YO/0,1,0,0,0,0,0/
4      DIMENSION XD(7),YD(7),ID(7),IBCD(2)
5      IND=1
6      YO=Y*FACT
7      XO=X*FACT
8      IB=IB+1
9      XD(IB)=XO
10     YD(IB)=YO
11     ID(IB)=I,PLOT
12     IF(I,PLOT,E0,999) GO TO 10
13     IF(IB,E0,7) GO TO 10
14     GO TO 123
15     ENTRY PLOTS(I,J,K,F)

```

リスト 7-1 ディバック用 PLOT ルーチン

```

15      WRITE(6,500) I,J,K,F
16      500 FORMAT(1H ,5HPLOTS,5X,3I5,5X,F10.2)
17      GO TO 123
18      ENTRY SYMBOL (X,Y,H,IBCD,R,N)
19      IND=2
20      IF(IB,NE,0) GO TO 10
21      2 WRITE(6,501) X,Y,H,IBCD,R,N
22      501 FORMAT(1H ,6HSYMBOL,5X,I2,3F10.2,2X,2A4,5X,F10.2,I5)
23      GO TO 123
24      ENTRY AXIS(X,Y,IBCD,N,F1,F2,F3,F4)
25      IND=3
26      IF(IB,NE,0) GO TO 10
27      3 WRITE(6,502) X,Y,IBCD,N,F1,F2,F3,F4
28      502 FORMAT(1H ,4HAXIS,5X,2F10.2,2X,2A4,5X,I3,6F10.2)
29      GO TO 123
30      ENTRY FACTOR(FCTR)
31      IND=4
32      FACT=FCTR
33      IF(IB,NE,0) GO TO 10
34      4 WRITE(6,504) FACT
35      504 FORMAT(1H ,6HFACTOR,5X,F10.2)
36      GO TO 123
37      ENTRY WHERE(RX,RY,RFCTR)
38      IND=5
39      IF(IB,NE,0) GO TO 10
40      5 RX=XO
41      RY=YO
42      RFCTR=FACT
43      WRITE(6,505) XO,YO,FACT
44      505 FORMAT(1H ,5HWHERE,3F15.2)
45      GO TO 123
46      ENTRY NEWPEN(INP)
47      IND=6
48      IF(IB,NE,0) GO TO 10
49      6 WRITE(6,506) INP
50      506 FORMAT(1H ,6HNEWPEN,5X,I1)
51      GO TO 123
52      10 WRITE(6,503) ((XD(I),YD(I),ID(I)),I=1,IB)
53      503 FORMAT(1H ,4HPLOT,7(1H ,1H(, F5,1,1H,,F5,1,1H,,I3,1H))
54      IB=0
55      GO TO (123,2,3,4,5,6),IND
56      123 RETURN
57      END

```

リスト 7-2

```

1      SUBROUTINE GROPEN
2      COMMON/ COMCRT/ ICTRL(800),IGDC(200),IGDSA(30)
3      + ,IGRSPN,NULL,ICRT,IGDSN
4      DATA NULL/-6/
5      DIMENSION IATDN(2),IDATA(1)
6      DIMENSION IGDSAI(30),IDCR(1)
7      DATA IOPNS/0/
8      DIMENSION IBUF(1),BUF(1)
9      IF(IOPNS,NE,0) GO TO 123
10     IOPNS=1
11     CALL INGSP(IGRSPN,NULL,ICRT,800)
12     CALL INDEV(IGRSPN,1,ICRT,1024,128,1,IGDC,200)
13     CALL INGDS(ICRT,IGDSN,IGDSA)
14     CALL SGDSL(IGDSN,50,0,0,0,950,0,900,0)
15     CALL CRATL(ICRT,IATN)
16     CALL ENATN(IATN,1,-16,32,36)

```

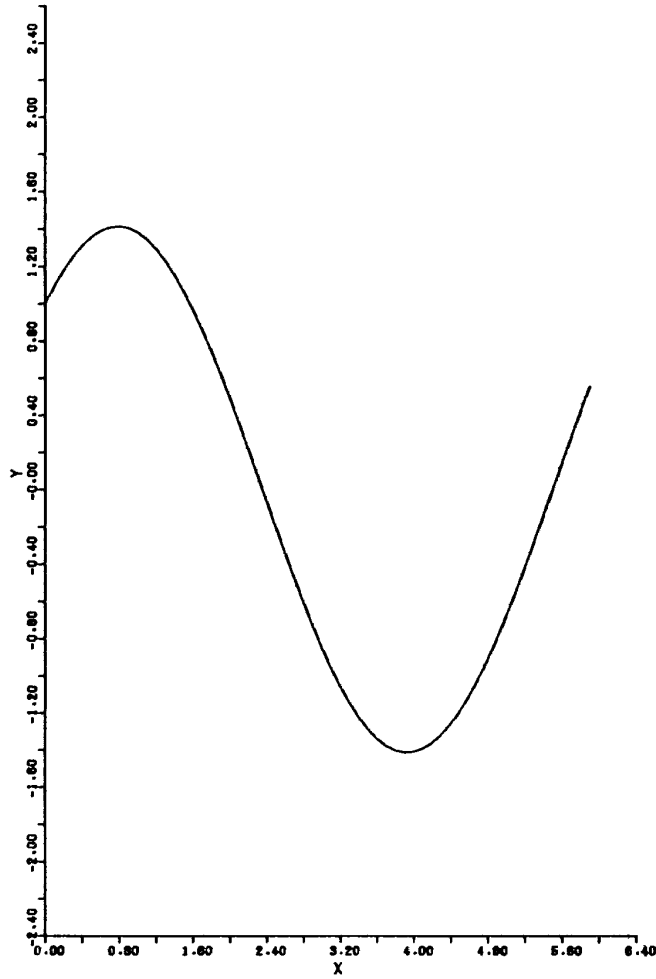
リスト 8-1 GROPEN


```

16      CALL INGDS(ICRT,IGDSN,IGDSA)
17      CALL SDATM(IGDSN,3)
18      CALL SCHAM(IGDSN,4)
19      IXIP=50
20      IYIP=970
21      ISP=0
22      GO TO 123
23      ENTRY GOPENN(NO)
24      NULL=NO
25      GO TO 123
26      ENTRY GERASE
27      IF(ISP.EQ.0) GO TO 9
28      CALL PLINE(IGDSN,BUFX,BUFY)
29      CALL EXEC(IGDSN)
30      ISP=0
31      9 CALL ERASE(ICRT)
32      GO TO 123
33      ENTRY GALRM
34      CALL SALRM(ICRT)
35      GO TO 123
36      ENTRY GCLOSE
37      CALL TMGDS(IGDSN)
38      IOPNS=0
39      CALL TMDEV(ICRT)
40      CALL TMGSP(IGRSPN)
41      GO TO 123
42      ENTRY GINPD(IDATA)
43      IND=1
44      GO TO 10
45      ENTRY GINPC(IDATA)
46      IND=2
47      GO TO 10
48      ENTRY GINPP(IDATA)
49      IND=3
50      GO TO 10
51      ENTRY GINPF(IFD)
52      IND=4
53      10 CONTINUE
54      CALL RQATN(IATN,IATD,2,IATDN,1,-16,32,36)
55      GO TO (11,12,13,14),IND
56      11 CONTINUE
57      IF(IATD.GT.16) GO TO 15
58      IDATA(1)=0
59      IDATA(2)=16*IATDN(1)+IATD
60      GO TO 123
61      15 IF(IATD.NE.32) GO TO 16
62      K=3
63      IDATA(1)=1
64      GO TO 17
65      16 CONTINUE
66      IDATA(1)=2
67      IDATA(2)=IATDN(1)
68      IDATA(3)=IATDN(2)
69      GO TO 123
70      12 CONTINUE
71      IF(IATD.NE.32) GO TO 10
72      K=2
73      17 CONTINUE
74      CALL GREAD(ICRT,IDATA(K),84,11)
75      IF(I1.GT.0) I1=0
76      IDATA(K-1)=84+I1
77      GO TO 123
78      13 CONTINUE
79      IF(IATD.NE.36) GO TO 10
80      IDATA(1)=IATDN(1)
81      IDATA(2)=IATDN(2)
82      GO TO 123
83      14 CONTINUE
84      IFD=16*IATDN(1)+IATD
85      GO TO 123

```

```
86     ENTRY GPTEXT(IDCR)
87     CALL PTEXT(IGDSN, IDCR(2), IDCR(1))
88     CALL STPOS(IGDSN, IXIP, IYIP)
89     CALL EXEC(IGDSN)
90     IXIP=50
91     IYIP=IYIP-30
92     IF(IYIP.LE.0) IYIP=970
93     GO TO 123
94     ENTRY GSPOS(IX0, IY0)
95     IXIP=IX0
96     IYIP=IY0
97     CALL ST POS(IGDSN, IXIP, IYIP)
98     GO TO 123
99     ENTRY GPLOT( IBUF, BUF, ICNO)
100    I=0
101    110 CONTINUE
102    I=I+1
103    IF(I.GT.1024) GO TO 78
104    IA=IBUF(I)
105    IC=IA/512
106    ICN=IA-IC*512
107    ICN=ICN-248
108    GO TO (71,72,73,74,75,76,77,88), ICN
109    72 CONTINUE
110    GO TO 79
111    74 CONTINUE
112    75 CONTINUE
113    GO TO 110
114    76 CONTINUE
115    I1=I+1
116    I=I+2
117    BUFX=BUF(I1)
118    BUFY=BUF(I)
119    CALL STPOS(IGDSN, BUFX, BUFY)
120    ISP=1
121    GO TO 110
122    77 CONTINUE
123    I1=I+1
124    CALL PLINE(IGDSN, BUF(I1), BUF(I+2), NULL, NULL, NULL, IC, 2, 2)
125    80 CONTINUE
126    CALL EXEC(IGDSN)
127    ISP=0
128    I=I+IC*2
129    GO TO 110
130    71 CONTINUE
131    73 CONTINUE
132    GO TO 79
133    88 I1=I+1
134    CALL PPNT (IGDSN, BUF(I1), BUF(I+2), NULL, NULL, NULL, IC, 2, 2)
135    GO TO 80
136    78 CONTINUE
137    ICN=0
138    79 CONTINUE
139    ICNO=ICN
140    GO TO 123
141    ENTRY GSDATL(X0, Y0, XL, YL, ISS)
142    IS=3
143    IF(ISS.EQ.1) IS=2
144    CALL SSCIS(IGDSN, IS)
145    CALL SDATL(IGDSN, X0, Y0, XL, YL)
146    GO TO 123
147    123 CONTINUE
148    RETURN
149    END
```



付図 1 図 11 の XY プロッタ出力

```

PLOTS      1      0      -1      100.00
AXIS       0.0      0.0      X      Y      -1      16.00      0.0      0.0      0.40
AXIS       0.0      0.0      Y      Y      1      25.00      90.00      -2.40      0.20
WHERE      0.0      0.0
PLOT ( 0.0, 17.0, 3) ( 0.3, 17.5, 2) ( 0.5, 17.9, 2) ( 0.8, 18.3, 2) ( 1.0, 18.6, 2) ( 1.3, 18.8, 2) ( 1.5, 18.9, 2)
PLOT ( 1.7, 19.0, 2) ( 2.0, 19.1, 2) ( 2.2, 19.0, 2) ( 2.5, 18.9, 2) ( 2.7, 18.7, 2) ( 3.0, 18.5, 2) ( 3.2, 18.2, 2)
PLOT ( 3.5, 17.8, 2) ( 3.7, 17.3, 2) ( 4.0, 16.9, 2) ( 4.2, 16.3, 2) ( 4.5, 15.7, 2) ( 4.7, 15.1, 2) ( 5.0, 14.5, 2)
PLOT ( 5.2, 13.8, 2) ( 5.5, 13.1, 2) ( 5.7, 12.4, 2) ( 6.0, 11.7, 2) ( 6.3, 11.0, 2) ( 6.5, 10.3, 2) ( 6.8, 9.6, 2)
PLOT ( 7.0, 9.0, 2) ( 7.3, 8.3, 2) ( 7.5, 7.8, 2) ( 7.8, 7.2, 2) ( 8.0, 6.7, 2) ( 8.3, 6.3, 2) ( 8.5, 5.9, 2)
PLOT ( 8.8, 5.6, 2) ( 9.0, 5.3, 2) ( 9.3, 5.1, 2) ( 9.5, 5.0, 2) ( 9.8, 4.9, 2) ( 10.0, 4.9, 2) ( 10.3, 5.0, 2)
PLOT ( 10.5, 5.2, 2) ( 10.8, 5.4, 2) ( 11.0, 5.7, 2) ( 11.3, 6.1, 2) ( 11.5, 6.5, 2) ( 11.8, 6.9, 2) ( 12.0, 7.3, 2)
PLOT ( 12.3, 8.0, 2) ( 12.5, 8.6, 2) ( 12.8, 9.3, 2) ( 13.0, 9.9, 2) ( 13.3, 10.6, 2) ( 13.5, 11.3, 2) ( 13.8, 12.0, 2)
PLOT ( 14.0, 12.7, 2) ( 14.3, 13.4, 2) ( 14.5, 14.1, 2) ( 14.8, 14.8, 2) ( 0.0, 0.0, 999) (

```

付図 2 ディバック用 PLOT ルーチンの出力

付録4 サブルーチンGROPENと PLOT(グラフィック用)

富士通から提供されている図形出力用サブルーチン・パッケージGRASPは、複数台のディスプレイを制御できるようにできている。そのためディスプレイ一台のみで使用している使用者には、GRASPはあまりにはん用すぎて、かえって使いにくくなっている。そこで必要な機能だけを取り出し、パラメータの数を最小限度少なくしたインターフェース・ルーチンを製作した。

このルーチンは、サブルーチン名GROPENで、12個のエントリー名をもつ。

(1) GROPEN

(1) GROPEN

目的；GRASPオープン・サブルーチン、最初にこのルーチンをCALLしておく。

呼び出し形式

CALL GROPEN

(2) GOPENN

目的；NULL定数を定める。このルーチンをCALLしなければNULLは-6になっている。

呼び出し形式

CALL GOPENN(N)

N；入力，NULL定数

(3) GERASE

目的；画面を消去する。

呼び出し形式

CALL GERASE

(4) GALRM

目的；ブザーを鳴らす。

呼び出し形式

CALL GALRM

ルーチン(5)~(8)はデータ入力に関するもので、これらのルーチンでは要求されているデータが入力されるまでプログラムは待たされる。

(5) GINPC

目的；キーボードから文字データを読み込む。

呼び出し形式

CALL GINPC(IDATA)

IDATA；出力，読み込んだ文字列がセットされる配列名，IDATA(1)には読み込んだ文字数がセットされる。最大84文字

(6) GINPP(IDATA)

目的；カーソルによる点の読み取り。

呼び出し形式

CALL GINPP(IDATA)

IDATA；出力，IDATA(1)にX，IDATA(2)にYがセットされる配列名。

$0 \leq X, Y \leq 1024$

(7) GINPE

目的；ファンクションキーからのデータを読み取る。

呼び出し形式

CALL GINPE(ID)

ID；出力，ファンクション番号(1~64)

(8) GINPD

目的；(5)~(7)のルーチンを一つのルーチンにまとめたもの。

呼び出し形式

CALL GINPD(IDATA)

IDATA；出力，データがセットされる配列名

IDATA(1)=0のときファンクションキー

1のとき文字

2のときカーソル

からの入力を示し，IDATA(2)以降にデータがセットされる。

(9) GSPOSI

目的；文字ポインタを与えられた点まで移動する。

呼び出し形式

CALL GSPOSI(IX,IY)

IX } 入力，移動点の座標
IY }

(10) GPTEXT

目的；文字列を表示する。

呼び出し形式

CALL GPTEXT(IDATA)

IDATA；入力，配列名，IDATA(1)に文字数，IDATA(2)以降に文字列をセットする。

(11) GSDATL

目的；バーチャル・ウィンドを定義する。

2：次へ続く

呼び出し形式

3：終了

CALL GSDATL(X0,Y0,XL,YL,ISS)

X0 } 入力, バーチャル・ウィンドの原点
Y0 }

XL } 入力, バーチャル・ウィンドの横と縦の
YL } 長さを指定

ISS ; 入力, ISS = 2 ならシザリングを行
なう。それ以外は行なわない。

(12) GPLOTF

目的；グラフィック用 PLOT ルーチンで作られ
るデータファイルの内容に従って, グラフ
ィック・ディスプレイに図形を出力する。

呼び出し形式

CALL GPLOTF(IBUF,BUF,ICNO)

IBUF,BUF ; 入力, 配列名, EQUIVA-
LENCE 文でむすばれた同一領域の配列,
大きさ 1024

ICNO ; 出力, 戻り状態を示す。

0 : 1024 を越えた

1 : 未定義

(13) GCLOSE

目的；GRASP クローズ・サブルーチン, 最後に
このルーチンを CALL する。

呼び出し形式

CALL GCLOSE

この GROPEN を使ってグラフィック・ディスプ
レイ用の " PLOT " ルーチンを作る。このルーチ
ンの中には, エントリー名が

PLOTS

FACTOR

WHERE

NEWPEN

のルーチンが入っており, 形式的にプロットング
・システムの " PLOT " ルーチンと同じものとな
っている。このルーチンを使うことにより, XY プ
ロット用のプログラムで, オンラインでグラフィッ
ク・ディスプレイに図形を出力できる。なおこのル
ーチンでは, 論理番号 50 の入出力装置をワークフ
ァイルとして使用している。

```

SUBROUTINE PLOT(XI,YI,IPENI)
DIMENSION IBUF(1024),BUF(1024)
DIMENSION ICXY(30)
COMMON ICTRL(800),IGDC(200),IGDSA(30)
+ ,IGRSPN,NULL,ICRT,IGDSN
EQUIVALENCE(DBUF(1),IBUF(1),BUF(1))
DATA IDWN,IUP,NEWPN,IPLOTS,IPLOTV,IFLCON,XORG,YORG,FACT,XO,YO,
C IFACT,IEC,IBI,IPD,IDC,XMAX,YMAX,XMIN,YMIN,IWC
C /255,254,253,252,251,250,2*0,0,1,0,2*0,0,5*0,2*-1,0E20,
C 2*1,0E20,0/
IPEN=IPENI
X=XI
Y=YI
IF(IFACT,NE,1) GO TO 10
X=X*FACT
Y=Y*FACT
10 CONTINUE
XO=X+XORG
YO=Y+YORG
IF(IPEN,EW,999) GO TO 900
IF(IPEN,GT,0) GO TO 20
XORG=XO
YORG=YO
IPEN=-IPEN
20 CONTINUE
IF(IPEN,EW,2) GO TO 30
IBI=IBI+1
IBUF(IBI)=IUP

```

```

50 CONTINUE
  IF(IPD,EQ,0) GO TO 60
  IBUF(IPD)=IBUF(IPD)+IDC*512
  IPD=0
  IDC=0
  GO TO 60
30 CONTINUE
  IF(IPD,NE,0) GO TO 31
  IBI=IBI+1
  IPD=IBI
  IBUF(IBI)=IDWN
31 CONTINUE
  IDC=IDC+1
60 CONTINUE
  IBI=IBI+1
  BUF(IBI)=XO
  IF(XO,LT,XMIN) XMIN=XO
  IF(XO,GT,XMAX) XMAX=XO
  IBI=IBI+1
  BUF(IBI)=YO
  IF(YO,LT,YMIN) YMIN=YO
  IF(YO,GT,YMAX) YMAX=YO
61 CONTINUE
  IF(IBI,LE,1000) GO TO 123
70 CONTINUE
  IBI=IBI+1
  IBUF(IBI)=IFLCON
  IBUF(IPD)=IBUF(IPD)+IDC*512
  IPD=0
  IDC=0
  IBI=0
  WRITE(50) IBUF
  NPLT=NPLT+1
  GO TO 123
  ENTRY PLOTS(I1,I2,I3,ALENG)
  GO TO 123
C
  ENTRY FACTOR(FACTI)
  X=FACTI-1,0
  IF(X,LT,0,0) X=-X
  IF(X,LT,1,0E-10) GO TO 910
  IFACT=1
  FACT=FACTI
  GO TO 123
910 IFACT=0
  FACT=1,0
  GO TO 123
C
  ENTRY WHERE(RX,RY,RECTR)
  RX=X
  RY=Y
  TECTR=FACT
  GO TO 123
900 CONTINUE
  IBI=IBI+1
  IBUF(IBI)=IPLTV
  IBI=0
  IF(IPD,EQ,0) GO TO 62
  IBUF(IPD)=IBUF(IPD)+IDC*512
  IPD=0
62 CONTINUE
  IDC=0
  WRITE(50) IBUF
  NPLT=NPLT+1
  CALL GOPEN
  CALL GERASE
  IBUF(1)=2U
  IBUF(2)=4HPUSH
  IBUF(3)=4H FUN

```

```

IBUF(4)=4HCTIO
IBUF(5)=4H N KE
IBUF(6)=4HY
CALL GSPOSJ(50,970)
CALL GPTEXT(IBUF)
CALL GINPF(IFD)
  REWIND 50
  ISS=0
  NPLTO=NPLT
80 CONTINUE
  XMA=XMAX
  YMA=YMAX
  XMI=XMIN
  YMI=YMIN
79 CONTINUE
  W=XMA -XMI
  H=YMA -YMI
  IF(W,GE,H) GO TO 1
  W=H
  1 CONTINUE
  XL=XMI+W
  YL=YMI+W
  CALL GSDATL(XMI,YMI,XL,YL,ISS)
  IF(NPLT,LE,0) GO TO 71
  NPLT=NPLT-1
  8 CONTINUE
  READ(50) IBUF
  CALL GPLOTG(IBUF,BUF,ICN)
  GO TO (71,8,71),ICN
71 CONTINUE
  NPLT=NPLTO
  REWIND 50
  CALL GINPF(IFD)
  IF(IFD,NE,1) GO TO 78
  SCL=0.001*W
  XO=XMI
  YO=YMI
  CALL GINPP(ICXY)
  XMI=ICXY(1)-50
  YMI=ICXY(2)
  XMI=SCL *XMI+XO
  YMI=SCL *YMI+YO
  CALL GINPP(ICXY)
  XMA=ICXY(1)-50
  YMA=ICXY(2)
  XMA=SCL *XMA+XO
  YMA=SCL *YMA +YO
  ISS=1
  CALL GERASE
  GO TO 79
78 CONTINUE
  CALL GERASE
  IF(IFD,NE,2) GO TO 80
  YMAX=-1.0E20
  YMIN=1.0E20
  XMAX=YMAX
  XMIN=YMIN
  GO TO 123
C
  ENTRY NEWPEN(NPEN)
123 CONTINUE
  RETURN
  END

```

航空宇宙技術研究所資料437号

昭和56年6月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町1880
電話武蔵野三鷹(0422)47-5911(大代表)〒182

印刷所 株式会社 三興印刷
東京都新宿区信濃町12 三河ビル
