

FM-7用プロッタールーチンの作製

秦野 和郎・葉 建勳・竹松 英夫

Implementation of a Plotter Routine for FM-7

Kazuo HATANO, Jian xun YE and Hideo TAKEMATSU

A plotter routine for personal computer FM-7 is developed and implemented. It drives MILOT Jounior (XY plotter) and CRT display. The program is compatible to that of main frame computer. Small scale graphic programs used on main from computer can run on personal computer.

1. まえがき

計算結果を「図」の形で出力すると直観的な資料がコンパクトな形で得られ、従って結果を整理しやすくなる事が多い。この目的に従来からXYプロッターとか蓄積管表示装置などが使われている。

大型計算機の利用者の多くはFORTRANを使ってプログラムを作製する。従って大抵の計算センターにはXYプロッターや蓄積管表示装置に出力するための標準的なサブルーチンライブラリが用意されている。これは通称カルコン仕様のプロッタールーチンと呼ばれており計算機によって若干の相違はあるとしてもこのプロッタールーチンを使ってプログラムを書いておけばどの計算機にもそのままか或いはわずかの変更でXYプロッター、蓄積管表示装置などに作図することができる。

最近パーソナルコンピュータが急速に普及しハード、ソフトともに充実してきている。安価な8ビットパソコンにすらCP/M-80なるモニターが提供されその制御のもとでFORTRANコンパイラが動く。その性質上、主記憶が少なく計算速度は遅いが応用によっては従来大型計算機を使っていた計算をパソコンで行うことができるようになってきている。

ほとんどのパソコンではCRTディスプレイはキャラクターディスプレイとしての機能だけでなくグラフィックディスプレイとしての機能をも併せ持っている。しかしその機能はメーカー提供のBASICインタープリタを使ってプログラムを作製するときのみ便利に使うことができる。FORTRANプログラムでその機能を活用するためには特別のプログラムを作らなければならない。同じことはXYプロッターについても言える。高度の機能を持ったXYプロッターを現在安価に入手できるが

それをFORTRANプログラムで活用するためには適当なプログラムを作製しなければならない。

大型計算機で使われているプログラムをパソコンで動かすために、又その逆にパソコンで大型計算機用のプログラムを準備するためにパソコン用のプロッタールーチンを作製した。その際、手近かに使えるという利点を生かすためにいくつかの工夫を試みた。本稿ではその概要を述べる。

2. 使用する装置とその機能

本章ではここで使用したパソコンシステムの概要を述べる(図2.1)。

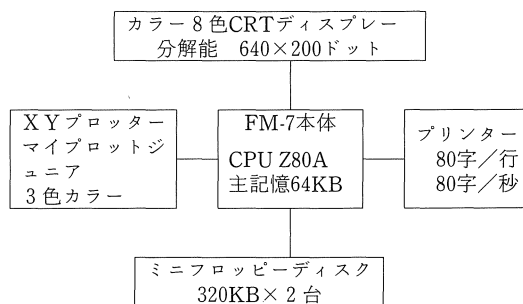


図2.1 システムの概略図

2.1 FM-7 (Fujitsu Micro 7)

このパソコンのCPUはMBL68B09であるが、Z80カードを装着することによってCP/M-80を使うことができる。CPTディスプレイへの表示のためにDisplay Sub Systemを持っている。これにより640×200ドットの分解能を持つグラフィック表示が可能である。Display Sub Systemは独立した専用のCPU(サブCPU)と64

KBのメモリを有している。そのメモリは主として8色のカラー表示のためのVideo RAMとして使われている。主CPUとサブCPUとは128バイトの共有RAM領域を介してデータのやりとりをする。

主CPUのアドレス空間内にVideo RAMを有するパソコンではVideo RAMへの読み書きが容易であるが、FM-7のように主CPUのアドレス空間とは全く別にVideo RAMを有するパソコンではユーザープログラムからVideo RAMへのアクセスは容易でない。この難点を解決するためにFM-7では基本入出力システム、FBIOSの中にDisplay Sub Systemと通信する機能を持たせている。このFBIOSの使用法は利用者に公開されているのみでなくCP/M-80のBDOS callで使用することができる。具体的にはRCB (Request Control Block) なる8バイトの領域を確保してその先頭アドレスをDEレジスタに乗せ、Cレジスタに255をロードして5番地をcallすればよい。

CP/M-80のもとでFORTRAN-80なるFORTRANコンパイラが動く。これに関連するソフトウェアにMACRO-80なるマクロアセンブラ、LINK-80なるリンケージローダ、LIB-80なるライブラリマネージャがあり便利に使うことができる。特にこれらの処理系はコマンド体系がよくできており非常に使い易い。FORTRAN-80はANSI-FORTRAN66という一時代前の規格をもとにして作製されており幾分古めかしい感じがするが実用上はこれで十分である。

2.2 XYプロッタ(グラフィック社製マイプロットジュニア, DA6100)

このXYプロッタは高度のインテリジェンスを持っているので比較的容易に使うことができる。豊富な機能を持っているが、ここではその一部しか使わない。

有効作図範囲は24.5cm×18cm, 最大作図速度は20cm/sec, ステップサイズは0.1mmで3種類のペンを使用することができる。

パソコン本体とはセントロニクスインターフェースを介して接続する。プリンタもこのインターフェースを介して接続されるのでプリンタとXYプロッタの両方を同時に使用することはできない。

パソコンからXYプロッタへの出力はプリンタへの出力と同じ形式で行えばよい。すなわち、プロッタコマンドをアスキーコードで1つつつ送出すればよい。1つのコマンドの最後には(CR/LF) (復帰改行コード)をつける。従ってXYプロッタ用のプロッタルーチンは全部FORTRANで書くことができる。たとえば点(1, 2)から点(4, 3)へ線を引くには2つのコマンド

M100, 200 (CR/LF)

D400, 300 (CR/LF)

をXYプロッタへ送出すればよい。但しここで点(α, β)は左下隅を原点としその点からx方向へ α cm, y方向へ β cm移動した点とする。この動作をFORTRAN80で実現するには

IX=100

IY=200

WRITE (2, 10) IX, IY

10 FORMAT (1H , 1HM, I4, 1H, , I4)

IX=400

IY=300

WRITE (2, 20) IX, IY

20 FORMAT (1H , 1HD, I4, 1H, , I4)

とすればよい

以下にここで使用したプロッタコマンドを列挙する。

(i) MOVE コマンド

M \bar{x} , \bar{y} (CR/LF)

\bar{x} , \bar{y} は0.1mmを単位とする原点からのそれぞれx方向, y方向への距離である。共に4桁以内の整数で $0 \leq \bar{x} \leq 2450$, $0 \leq \bar{y} \leq 1800$ でなければならない。

ペンを上げて(\bar{x} , \bar{y})へ移動する。

(ii) DRAW コマンド

D \bar{x} , \bar{y} (CR/LF)

\bar{x} , \bar{y} はMOVEコマンドと同じである。ペンを下げて現在位置から(\bar{x} , \bar{y})へ線を引く。

(iii) PRINT コマンド

P $\bar{c}_1\bar{c}_2\cdots\bar{c}_n$ (CR/LF)

$\bar{c}_1\bar{c}_2\cdots\bar{c}_n$ で指定されるn個の文字をプロットする。 \bar{c}_i はアスキー文字である。

(iv) MARK コマンド

N \bar{n} (CR/LF)

\bar{n} で指定された特殊文字(記号)をプロットする。 \bar{n} は記号の番号である。

(v) ALPHA SCALE コマンド

S \bar{n} (CR/LF)

プロットされる文字, マークの大きさを指定する。このコマンドを送出した後, PRINTコマンドでプロットされる文字の大きさは, $0.7 \times (\bar{n} + 1)$ mm となる。又, MARKコマンドでプロットされるマークの大きさは $0.4 \times (\bar{n} + 1)$ mm となる。

(vi) ALPHA ROTATE コマンド

Q \bar{n} (CR/LF)

\bar{n} は0, 1, 2, 3のいずれかである。PRINTコマンドでプロットされる文字の回転角度を指定する。文字の回転角度が $90^\circ \times \bar{n}$ になる。

(vii) AXIS コマンド

$X\bar{p}, \bar{q}, \bar{r}$ (CR/LF)

座標軸を描く。 $\bar{p}=0$ でY軸, 1でX軸が描かれる。 \bar{q} で指定された長さ(0.1mmを単位とする)毎に目盛線を入れて \bar{r} で指定された回数だけ軸を延長する。軸の全長は $\bar{q} \times \bar{r} \times 0.1\text{mm}$ となる。

(viii) HOME コマンド

H (CR/LF)

プロッタを初期化する。

(ix) SPEED コマンド

$T\bar{n}$ (CR/LF)

ペンダウン時のペンの速度を指定する。 \bar{n} は0~10の整数である。ペンの速度が $\bar{n} \times 20/10\text{cm/sec}$ になる。使用する用紙の紙質に応じて \bar{n} を変える。

(x) NEW PEN コマンド

$J\bar{n}$ (CR/LF)

\bar{n} は0, 1, 2, 3のいずれかである。 $\bar{n}=0$ とすると現在持っているペンをペンストックへ収納する。1~3のときは \bar{n} 番目のペンを選択する。

マイプロットジュニアにはこの他にも円弧を描いたり、片カナをプロットしたり破線、鎖線などをプロットする機能があるが、ここではそれらを使わない。

ここで作製したXYプロッタールーチンでは上に列挙したコマンドのみを使用する。

3. 作製したプロッタールーチン

前章で述べたような環境では、いくつかの使用形態を持つプロッタールーチンを作製することができる。

図3.1に示すようにいくつかのプログラムを作製した。

(□が一つのプログラムになっている)

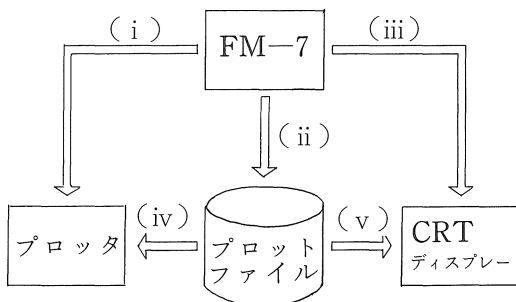


図3.1 データの流れ (プロッタールーチンのモード)

(i) FM-7で計算した結果を直ちにプロッタに出力するモード。

(ii) 計算結果を一旦フロッピーディスクにファイルとして出力し計算終了後(iv)又は(v)のルーチンを使ってXYプロッタ又はCRTディスプレイに出力するモード。フ

ロッピーディスクへの出力はXYプロッタコマンドの形とする。

(iii) 計算結果を直ちにCRTディスプレイに出力するモード。

(iv) フロッピーディスクに出力されたXYプロッターコマンドを読み出し、XYプロッタに出力するルーチン。実際にはこのルーチンを作製する必要はない。ファイルからXYプロッタへの単純な転送であるからCP/M-80の標準的なコマンドPIP(Peripheral Interchange Program)を使用することができる。(ii)で出力されたファイルのファイル名をABC.OUTとすると

B>A: PIP _LST:=ABC.OUT

によりXYプロッタへ出力できる。

(v) フロッピーディスクへ出力されたXYプロッターコマンドを解読してCRTディスプレイに出力するルーチン。

通常は(i)のモードを使うのがよい。しかし長時間の計算後にXYプロッタへ出力するときは(i)のモードでは不都合が生ずる。

XYプロッタを長時間待機させておくとペン先が乾いてしまう。従ってXYプロッタを使うときは連続的に動作させるようにするのが望ましい。そのために(ii)と(iv)のモードを使うと具合がよい。すなわち結果をまづフロッピーディスクに出力して計算が終了した後、XYプロッターに出力する(ほとんどの計算センターではこの形態でのみXYプロッターを使用できる)。計算実行中はXYプロッタの電源を入れておく必要はない。当然ペンにはキャップをかぶせて乾燥を防ぐことができる。

ファイルからXYプロッタに出力する前にまづ、CRTディスプレイに出力し結果が妥当なものであるかどうかを手軽に確認することができる。すなわち(v)のルーチンによりファイルからXYプロッタコマンドを読んでCRTディスプレイに表示する。このプログラムはコンプリートプログラムの形になっている。

プログラムのデバックの段階では(iii)のモードが有用である。この段階ではハードコピーを必要としない事が多いのでCRTディスプレイへの表示だけで十分である。

(i), (ii), (iii)は利用者プログラムから呼ぶことができるサブルーチンパッケージの形になっている。(i), (ii), (iii)のいずれを使うときも利用者プログラムは全く同じでよい。

(i), (ii)は同一のプログラムで(i)か(ii)の選択は後出のPLOTSの引数によってきめる。

(iii)はサブルーチン名が(i), (ii)と同じになっている。(i), (ii)か(iii)の選択はリンク時に指定する。

次に図3.1の(i), (ii), (iii)に含まれるサブルーチンの仕様

を簡単に述べる。

まづこれらのサブルーチン名と引数を列挙する。

```
PLOTS (ND, NFILE, NU)
PLOT (X, Y, IP)
SYMBOL (X, Y, HEIGHT, IASC, THETA,
N)
NUMBER(X, Y, HEIGHT, FN, THETA, N)
SCALE (DATA, SIZE, N, K)
AXIS(X, Y, LABEL, N, SIZE, THETA, DMIN,
DDEL)
LINE (XDAT, YDAT, N, K, J, L)
LINMRK (HMRK)
AXSPRM(AXSTP, HEG1, HEG2, AIN1, AIN2)
FACTOR (FAC)
WHERE (X, Y, FAC)
NEWPEN (IP)
SPEED (ISDEG)
HDCOPY (IBUF, ICTB, ICTG)
```

引数はI~Nで始まる変数は2バイト型整数、それ以外の文字で始まる変数は4バイト実数である。但し、IASCはN>0のときは文字列、NFILE、LABELは文字列である。

以上14種のサブルーチンは使用するモードにより用法に若干の相違がある。

PLOTS (ND, NFILE, NU)

は図形出力の初期設定を行う。これと対になるのがPLOT (X, Y, 999)である。PLOTS (ND, NFILE, NU)が呼ばれてPLOT (X, Y, 999)が呼ばれるまでが1枚の図になる。従って図1枚について1回づつこれらのサブルーチンを呼ばなければならない。

モード(i)のときにはPLOTSの引数NUに2を与える。ND, NFILEには何を書いてもよい(無視される)。モード(ii)のときはPLOTSの引数はすべて無視される。PLOTSが呼ばれると画面を消去する。(i), (ii)いずれのモードでもPLOT (X, Y, 999)を呼ぶと計算を一時中断しRETURNキーを押すと再開する。モード(i)のときはこの間に紙を取り換える。

NUに6以上の整数を与えてPLOTSを呼ぶとモード(iii)になる。このモードではファイルを開く。どのフロッピーディスクにファイルを作るかをNDで指示する。ND=1ならドライブA, ND=2ならドライブBにファイルが作られる。作るべきファイルのファイル名をNFILEに11文字で与える。たとえばB:ABC.XYPを出力ファイルとするなら

CALL PLOTS (2, 'ABC□□□□□XYP', 8)
と呼ぶ。サブルーチンPLOTSの内部では、

```
CALL OPEN (8, 'ABC□□□□□XYP', 2)
```

を呼んでB:ABC.XYPを開く。

以降呼ばれるプロッタールーチンにより生成されるXYプロッターコマンドはファイルB:ABC.XYPに出力されていく。最後に

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

を呼ぶとサブルーチンPLOTの内部で

```
ENDFILE □ 8
```

を実行しファイルを閉じる。

1つの計算で複数個の図を書くのが普通であるがこのようなときはたとえば次のようにするとよい。

```
IMPLICIT INTEGER * 2 (I-N)
```

```
DIMENSION NFILE (6)
```

```
ND= 2
```

```
NU= 8
```

```
MAI=500
```

```
.....
```

```
ENCODE (NFILE, 10) MAI
```

```
10 FORMAT (3HABC, I3, 2X, 3HXYP)
```

```
CALL PLOTS (ND, NFILE, NU)
```

```
{(1枚目の図)}
```

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

```
MAI=MAI+1
```

```
ENCODE (NFILE, 10) MAI
```

```
CALL PLOTS (ND, NFILE, NU)
```

```
{(2枚目の図)}
```

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

```
.....
```

これにより1枚目がファイルB:ABC500.XYPに、2枚目がファイルB:ABC501.XYPに出力される。

図形出力でもっとも頻繁に使うのが

```
PLOT (X, Y, IP)
```

である。ほとんどの場合PLOTSとPLOTの2つで十分である。

このサブルーチンを呼ぶとプロッタのペンが現在位置から点(X, Y)へ移動する。単位はcmである。IPによりペンの状態を指示する。

```
|IP|= 1 ならペンの状態は変わらない。
```

```
|IP|= 2 ならペンを下げて指定された点へ移動する。従って直線が引かれる。
```

```
|IP|= 3 ならペンを上げて指定された点へ移動する。
```

```
IP< 0 なら指定された点を新たな原点とする。PLOTSを呼んだ時点では左下隅が原点になっている。
```

```
IP = 0 ならなにもしない。
```

IP = 999 とすると1枚の図の終了処理を行う。
 どのような動作をするかは(i), (ii), (iii)の
 モードによって異なる(前出)。

次に

SYMBOL (X, Y, HEIGHT, IASC, THETA,
 N)
 は指定された点に(又は指定された点から)記号(又は
 文字列)を描く。

X, Y で点を指定し, HEIGHT で文字, 記号の高さ,
 THETA で文字の回転角を指定する(X軸正の方向から
 反時計まわりで単位は度)。但し, THETA は0.0, 90.0,
 180.0, 270.0のいずれかに限る。

角度の制限はマイプロットジュニア及び, FM-7の
 Display Sub System の制限に由来する。大型計算機で使
 われる大型のプロッタールーチンはその中に文字フォ
 ントを持っている。しかしパソコンではメモリの制約が大
 きいのでここではルーチン内に文字フォントは持たず,
 プロッターや Display Sub System の機能をそのまま使
 った。

N の値により動作が大きく異なる。

$N > 0$ のとき, 現在位置からペンを上げて点(X,
 Y)へ移動し IASC に与えられた N 個の文字を
 プロットする。但し X, Y が共に999.0ならペ
 ンの移動は行わず現在位置から文字をプロ
 ットする。

$N < 0$ のときには点(X, Y)を中心として記号を
 プロットする。記号の番号を IASC で指定す
 る。又 X, Y が共に999.0ならペンは移動しな
 い。

$N = -1$ のときペンを上げて移動する。

$N \leq -2$ のときペンを下げて移動する。

記号をプロットしながらグラフを書くときには
 $N = -2$ とするとよい。

NUMBER(X, Y, HEIGHT, FN, THETA, N)
 は数値をプロットする。HEIGHT, THETA はSYM-
 BOL の場合と同じである。

ペンを上げて点(X, Y)へ移動して FN で与えられる
 数字列をプロットする。但し X, Y が共に999.0のときは
 ペンは移動しない。

$0 \leq N \leq 9$ のとき FN の小数点以下 N 位までを
 プロットする。

$N = 0$ のとき整数部と小数点のみをプロットす
 る。

$N = -1$ のとき整数部のみをプロットする。

$-9 \leq N \leq -1$ のとき整数部の $10^{|N|}$ 位までをプロ
 ットする。

SCALE, AXIS, LINE, LINMRK, AXSPRM はこ
 れだけを一組にして使用する。グラフを描くのに用いる。
 LINE が主たる働きをする。これらのサブルーチンの機
 能は複雑であるからここでは説明を省略する。

FACTOR (FAC)

はプロット尺度を変える。現在の原点位置からの距離が
 FAC 倍される。

WHERE (X, Y, FAC)

はペンの現在位置, プロット尺度を与える。

NEWPEN (IP)

はペンの種類を変更する。IP によりペンの種類を指定す
 る。

SPEED (ISDEG)

はペンの速度を変える。マイプロットジュニアではペ
 ンの速度を10段階に変えることができる。ペンの速度が
 ISDEG \times 2 cm/sec になる。使用する紙の種類によ
 って変えるとよい。このサブルーチンはモード(iii)では無視さ
 れる。

HDCOPY (IBSF, ICTB, ICTG)

はモード(iii)でのみ意味がある。画面に表示された図をプ
 リンタへハードコピーする。XY プロッタに比較すると
 非常に雑な図になる。IBSF は作業領域で209バイト以上
 の長さでなければならない。ICTB, ICTG はそれぞれ
 128, 126とする。

さてモード(i), (ii)のサブルーチンは全部 FORTRAN
 80で書かれており, コンパイルして A:PLOT80.REL
 というファイルに入っている。モード(iii)のサブルーチ
 ンはほとんどが FORTRAN80で書かれているが10行程度
 は MACRO80で書いた。これらはコンパイルされアセン
 ブルされて LIB80により結合されて A:GRAF7.
 REL というファイルに入っている。モード(iv)のプログラ
 ムは FORTRAN80と MACRO80で書かれておりコンパ
 イル, アセンブル, リンクされて A:GRPL7.COM と
 いうファイルに入っている。

プロッタールーチン呼び出す利用者プログラムが
 B:SAMP.FOR というファイルに入っているとすると
 (i), (ii)のモードの操作手順は次のようになる。

$B > A : F80_ = SAMP$

$B > A : L80_SAMP, A : PLOT80/S, A : FOR-$
 $LIB/S, SAMP/N/E$

$B > SAMP$

又, (iii)のモードでの操作手順は次のようになる。

$B > A : F80_ = SAMP$

$B > A : L80_SAMP, A : GRAFM 7/S, A :$
 $FORLIB/S, SAMP/N/E$

$B > SAMP$

```

A:PIF CON:=B:RLTESIE1.FOR[NI]
1: C*
2: C***** PLTES1 ** PLTESIE1 *****
3: C*
4: C* PLOTTER TEST (1)
5: C*
6: C*****
7: PROGRAM PLTES1
8: IMPLICIT INTEGER*2 (I-N)
9: INTEGER*1 NFILE1
10: REAL *4 PI,STEP,XP,YF,FACT,SCALX,SCALY,SHIY
11: 1 ,XFL,YPL,XDAT,YDAT
12: DIMENSION NFILE1(11),XDAT(100),YDAT(100)
13: DATA PI / 3.14159 26535 /
14: MAI=101
15: ENCODE (NFILE1,1100) MAI
16: 1100 FORMAT (3HRL0,I3,2X,3HOUT)
17: CALL OPEN(5,'RLTESI1IDAT',2)
18: READ (5,1100) NUI,ND1,FACT
19: 1110 FORMAT(2I5,F10.5)
20: C --- SIN , COS ---
21: CALL PLOTS(ND1,NFILE1,NU1)
22: CALL FACTOR(FACT)
23: CALL PLOT(0.0,0.0,3)
24: CALL PLOT(18.0,0.0,2)
25: CALL PLOT(18.0,18.0,2)
26: CALL PLOT(0.0,18.0,2)
27: CALL PLOT(0.0,0.0,2)
28: CALL NEWPEN(2)
29: CALL SYMBOL(4.0,16.0,1.0,'SIN(X)',0.0,6)
30: CALL SYMBOL(11.0,2.0,1.0,'COS(X)',18.0,6)
31: CALL NEWPEN(3)
32: CALL SYMBOL(9.0,8.0,0.5,'SIN(X)',90.0,6)
33: CALL SYMBOL(15.0,17.0,0.5,'COS(X)',270.0,6)
34: CALL PLOT(2.0,2.0,-3)
35: NINT=100
36: STEP=2.0*PI/FLOAT(NINT)
37: SCALX=14.0/(2.0*PI)
38: SHIY =7.0
39: SCALY=7.0
40: NINT1=NINT+1
41: CALL NEWPEN(1)
42: CALL PLOT(0.0,0.0,3)
43: CALL PLOT(0.0,2.0*SCALY,2)
44: CALL PLOT(0.0,SCALY,3)
45: CALL PLOT(14.0,SCALY,2)
46: CALL PLOT(14.0,0.0,3)
47: CALL PLOT(14.0,2.0*SCALY,2)
48: DO 1120 I1=1,NINT1
49: XP=STEP*FLOAT(I1-1)
50: YF=SIN(XP)
51: XPL=XP*SCALX
52: YPL=YF*SCALY+SHIY
53: IPEN=2
54: IF (I1.EQ.1) IPEN=3
55: CALL PLOT(XPL,YPL,IPEN)
56: 1120 CONTINUE
57: DO 1130 I1=1,NINT1
58: XP=STEP*FLOAT(I1-1)
59: YF=COS(XP)
60: XPL=XP*SCALX
61: YPL=YF*SCALY+SHIY
62: IPEN=2
63: IF (I1.EQ.1) IPEN=3
64: CALL PLOT(XPL,YPL,IPEN)
65: 1130 CONTINUE
66: CALL PLOT(0.0,0.0,999)
67: C --- COS*SIN+COS ---
68: DO 1140 I1=1,NINT1
69: XP=STEP*FLOAT(I1-1)
70: YF=COS(XP)*SIN(XP)+COS(XP)
71: XDAT(I1)=XP
72: YDAT(I1)=YF
73: CONTINUE
74: 1140 MAI=MAI+1
75: ENCODE (NFILE1,1100) MAI
76: CALL PLOTS(ND1,NFILE1,NU1)
77: CALL PLOT(2.0,2.0,-3)
78: CALL SCALE(XDAT,20.0,NINT1,1)
79: CALL SCALE(YDAT,14.0,NINT1,1)
80: CALL NEWPEN(2)
81: CALL SYMBOL(0.5,0.5,0.8,'COS(X)*SIN(X)+COS(X)',
82: 1 ,0.0,20)
83: CALL NEWPEN(3)
84: CALL AXIS(0.0,0.0,'X-AXIS',6,16.0,0.0
85: 1 ,XDAT(NINT1+1),XDAT(NINT1+2))
86: CALL AXSPRM(1.0,0.21,0.28,0.6,1.1)
87: CALL AXIS(0.0,0.0,'Y-AXIS',-6,14.0,90.0
88: 1 ,YDAT(NINT1+1),YDAT(NINT1+2))
89: CALL NEWPEN(1)
90: CALL LINMRK(0.5)
91: CALL LINE(XDAT,YDAT,NINT1,1,10,0)
92: CALL AXSPRM(2.0,0.42,0.56,0.4,1.0)
93: CALL AXIS(0.0,14.0,'COS(X)*(1.0+SIN(X))',-19
94: 1 ,16.0,0.0,XDAT(NINT1+1),XDAT(NINT1+2))
95: CALL AXSPRM(2.0,0.42,0.56,0.6,1.2)
96: CALL AXIS(16.0,0.0,'Y',0,14.0,90.0
97: 1 ,YDAT(NINT1+1),YDAT(NINT1+2))
98: CALL PLOT(0.0,0.0,999)
99: STOP
100: END

```

図4.1

このようにXYプロッターに出すかCRTディスプレイに出すかはリンク時にどのライブラリを指定するかによりきめることができる。

モード(ii)により作製したプロッターファイルをCRTディスプレイに表示するには

A>GRPL 7

とタイプし、プロンプトに従ってファイル名をタイプすればよい。

以上のように、ここで作製したプロッタールーチンは非常に使用しやすいものであると思われる。

4. 使用例

図4.1に簡単なプログラム例を挙げる。このプログラムを動かすとモード(i)では図4.2、図4.3が、モード(ii)では

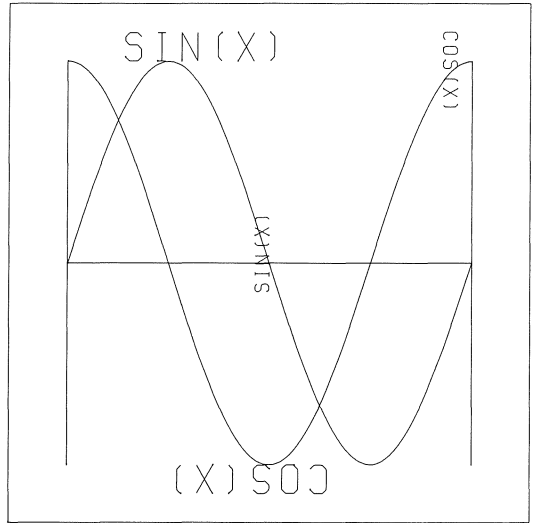


図4.2

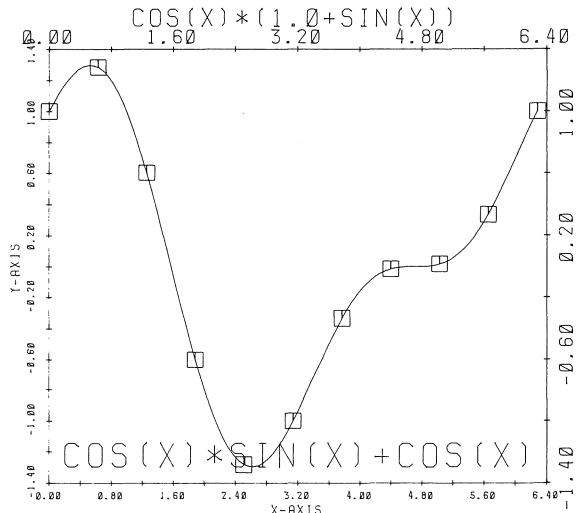


図4.3

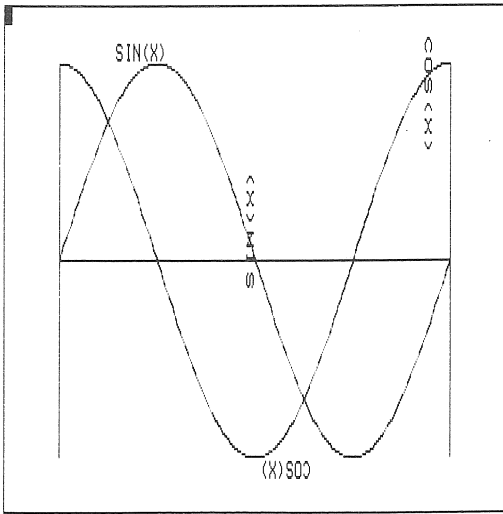


図4.4

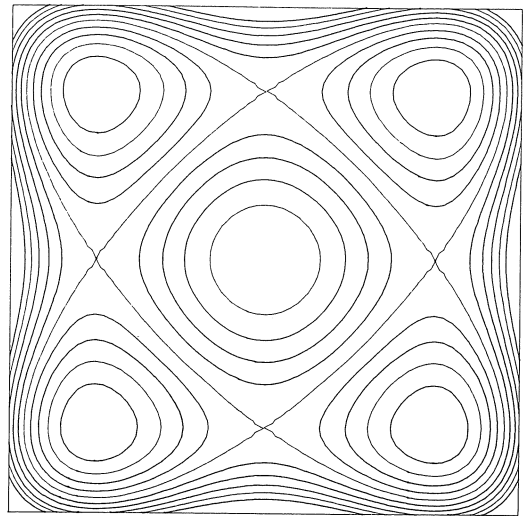


図5.1

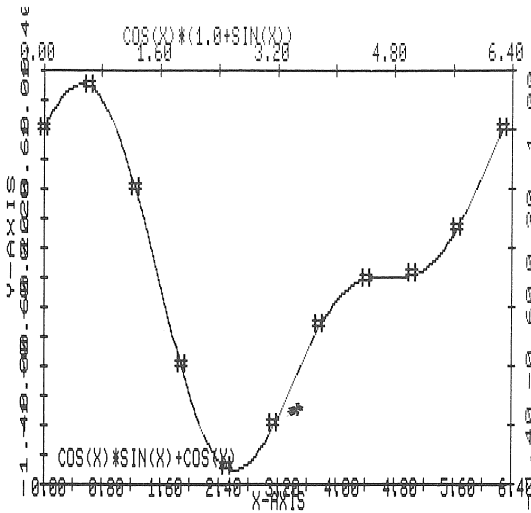


図4.5

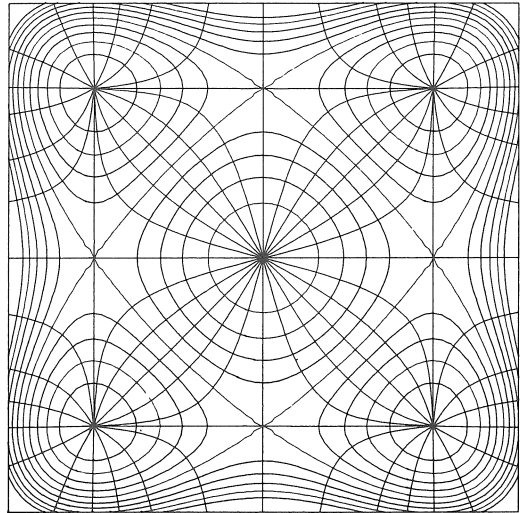


図5.2

図4.4, 図4.5が得られる。

図4.1で21行目から66行目までが図4.2を、76行目から98行目までが図4.3を描く。18行目で読むデータをここではNU 1に8, ND 1に2, FACTに1.0を与えた。

5. 応用

パソコン上で動くプロッタールーチンができたので試みに名古屋大学大型計算機センターでライブラリープログラムとして使用されている等高線を書くプログラムと立体図を書くプログラムとをFM-7上で動かしてみた。プログラムにわずかの修正が必要であったが大型計算機で使われているプログラムがほとんどそのままの形でFM-7上で動く。

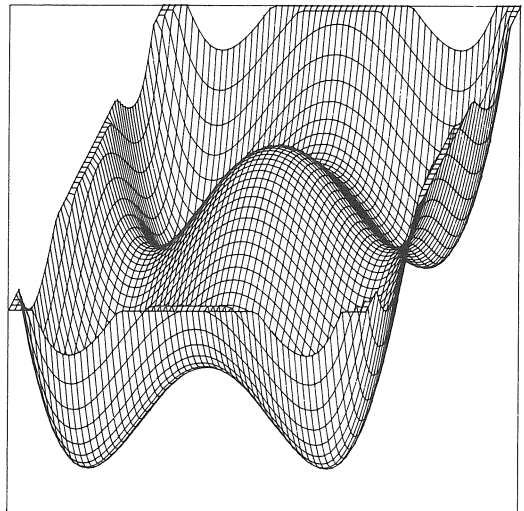


図5.3

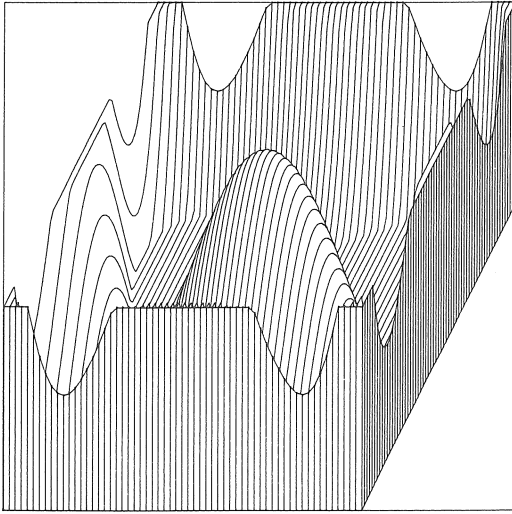


图5.4

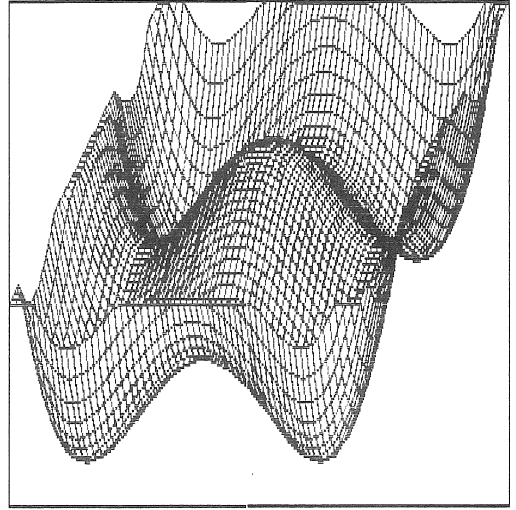


图5.7

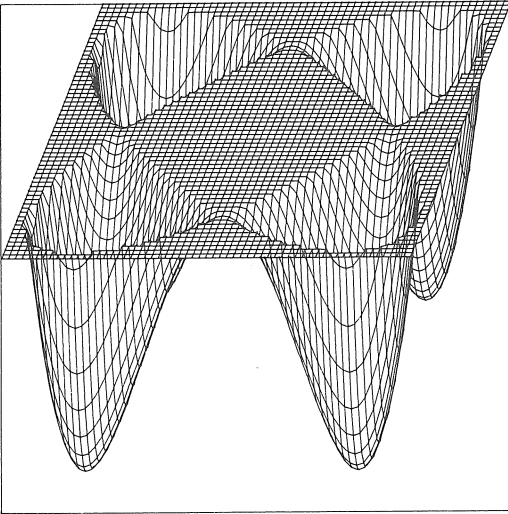


图5.5

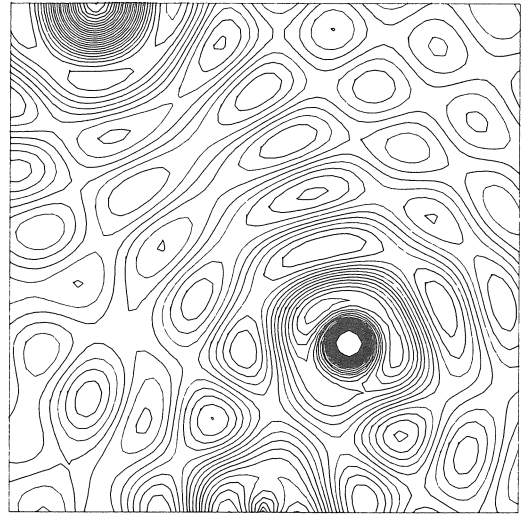


图5.8

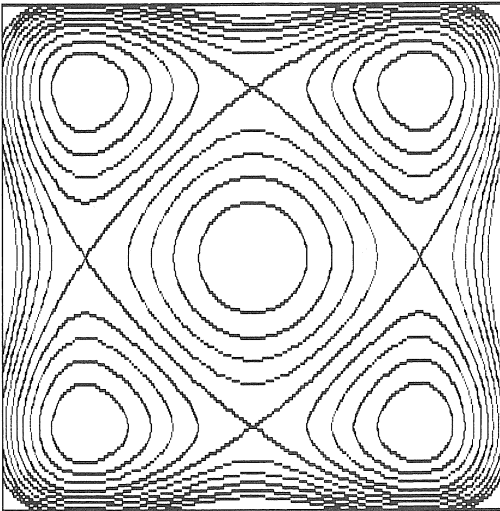


图5.6

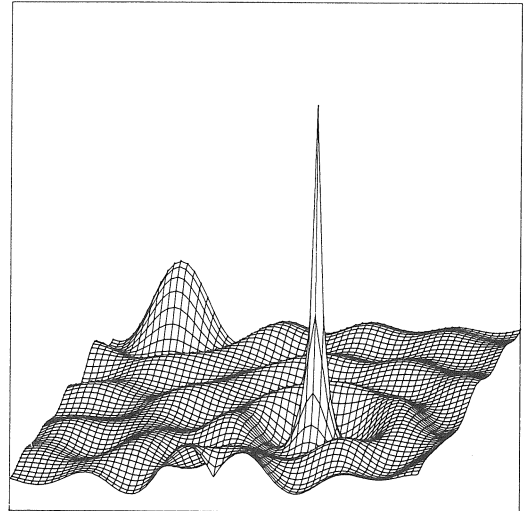


图5.9

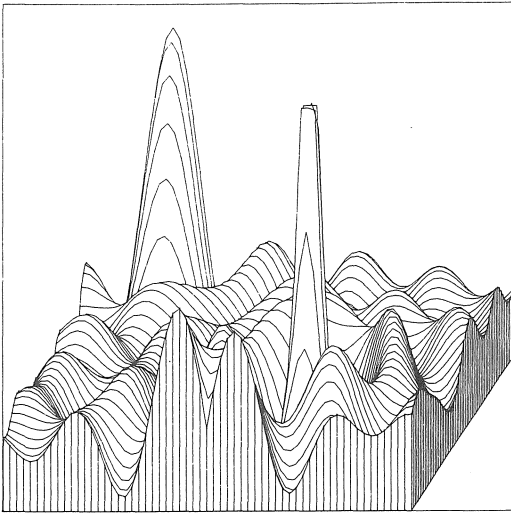


図5.10

図5.1～図5.5は

$$f(x, y) = (x^2 - 1)^2 + (y^2 - 1)^2 - 1$$

$$|x| \leq 1.5, |y| \leq 1.5$$

の等高線と立体図である。図5.2には更に等高線と直交する線が書き加えてある。(場が電界をあらわすならこれは電気力線をあらわす) 図5.6～図5.7はモード(iii)での図である。

図5.8～図5.10は

$$f(x, y) = \frac{\cos r_1}{r_1} + \frac{\sin r_2}{r_2}$$

$$+ \frac{\sin r_3 - r_3 \cos r_3}{r_3^2}$$

$$\left\{ \begin{array}{l} r_1 = \sqrt{(x-5)^2 + (y-5)^2} \\ r_2 = \sqrt{(x+15)^2 + (y+10)^2} \\ r_3 = \sqrt{(x-15)^2 + y^2} \end{array} \right.$$

$$|x| \leq 15, |y| \leq 15$$

の等高線と立体図である。これらの図を1枚描くのに必要な計算時間は30分位であった。

6. むすび

計算機はハードウェアだけでは何の役にも立たない。

豊富なソフトウェアとそれらを使いこなす技術が伴って始めて有用な道具になるのである。パソコンも小さいとは言え計算機であるから同じことが言える。ハードウェアについて言えば8セットパソコンですら15年前の中型機程度の能力は持っているように思われる。しかしそれが15年前の中型機と同じ程度に有用に使われているかどうかは疑がわしい。8ビットパソコンといっても基本ソフトウェアはCP/M-80, FORTRAN-80, COBOL-80等々、かなり豊富になってきている。これらは使い慣ればパソコンの有用性を大きく高めてくれるが十分に使われてはいないようである。価格が高いということが大きな理由かも知れないがむしろこれらのソフトウェアの持つ潜在的な能力が正しく認識されていないのではないかと思う。

本稿に例示した等高線、立体図は筆者の1名がFACOM230-60という大型機で12年前に書いたものである。これがそのまま安価な8ビットパソコンで処理できるようになったことに驚きを禁じ得ない。これだけの能力を持つパソコンを手近かに使えるのはありがたい事である。

8ビットパソコンでこれだけのことができるのであるから16ビットパソコンではもっと使い易く豊富な機能を持つプログラムを作れるであろう。ここで作製したプログラムを、16ビットパソコンMULTI16用書きかえるべく現在検討中である。

参考文献

- 1) グラフテック社：マイプロットジュニア取扱説明書，1983
- 2) 富士通：FM-7 ユーザーズマニュアルシステム仕様，1982
- 3) 富士通：FM-7 CP/M-80操作手引書，1982
- 4) 名古屋大学大型計算機センター：図形出力の手引，1977

(受理 昭和59年1月17日)