

金 沢 大 学 計 算 機 セ ン タ ー

## 利 用 の 手 引 き

計 算 機 に グ ラ フ を 書 か せ て み よ う

GRACE IVを用いたプログラミングから検索、出力までの実際

執 筆 者

工 学 部

助 手

中 野 康 英

## はじめに

私達が計算機から、あるいは実験装置から得ている数値情報の中には、数値のまま見るよりはグラフ化した方が理解が容易な情報が少なからずある。ところが、人間がプロットする能力には、さほどの進歩はないにもかかわらず、計算機からの数値情報は、計算機的能力向上に伴ってどんどん大量化してきている。そこでプロットもついでに計算機にやらせたいという要望が生じてくる。

この手引書は、初めて計算機にグラフを書かせてみようとしている人およびその初心者に対して、グラフ化のプログラミングから出力までの実際をできるだけ実例にそってやさしく解説しようとするものである。グラフ化のプログラミングに際しては、主として“GRACE IV”を用いる。“GRACE IV”とは、図形が簡単に描けるようにそれに有用なサブルーチンをまとめて登録したものである。この手引書では、GRACE IVの機能のうちグラフ化に際してよく使われそうな機能を、できるだけ多く実例の中で紹介しようと思う。

したがって、どうしても広く浅くという傾向をまぬがれ得ない(※1)。その欠点を克服するのは、各ユーザの「見やすいグラフを作りたい」という熱意であろう。各ユーザは、この手引書の範囲にとどまらずさらに工夫、学習されることを希望する。参考文献を巻末に掲載する。

---

※1 この手引書でのGRACE IV サブルーチンの紹介およびそれらの機能説明は、よく使われると思われるもののみを選択した。ちなみにGRACE IV サブルーチンの数は、ここで紹介されたものの約4倍ある。

# 第1章 グラフ作画の前に

## 1. 1 問題の設定

この手引書では、次に示す問題のグラフ化を実例として解説を進める。

【問題】 図1に示すような長さ $L$ の棒の両端AとBを $0^{\circ}\text{C}$ に保ち棒の中心点Cを $T_0$  [ $^{\circ}\text{C}$ ] ( $T_0 > 0$ ) に加熱すると、棒の温度 $T$ は点Cを境にして対称な直線状の分布を持つ。いま、ある瞬間に点Cにおける加熱を中止すると、棒の温度は下がってゆく。

この時の温度分布の時間変化を計算する。但し、棒の側面からの熱放散は無視する。

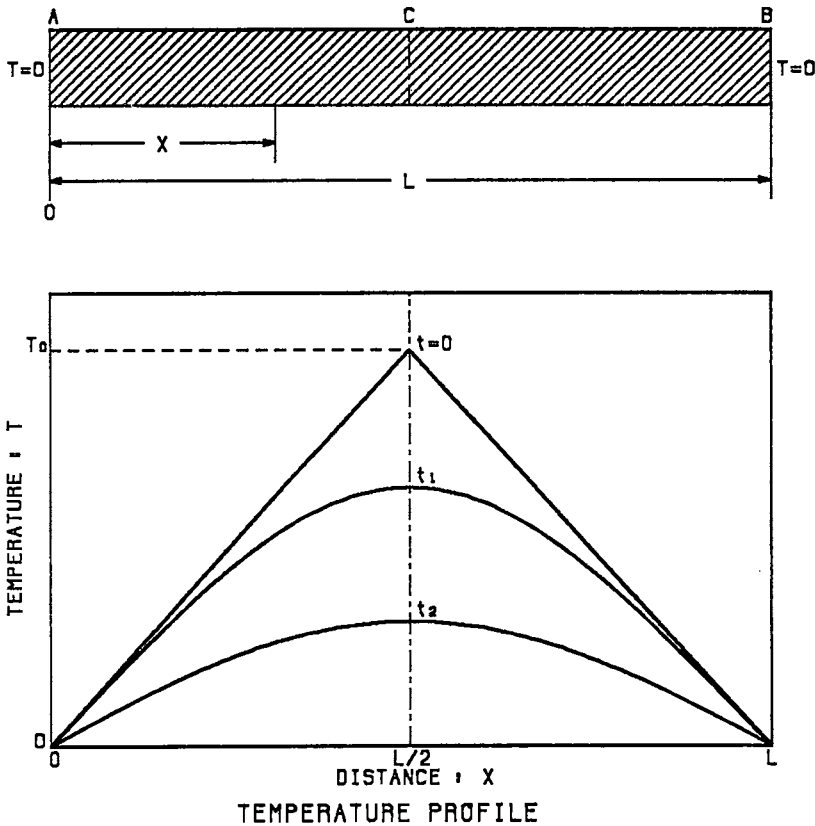


図1 問題のモデル

一般には、まず問題を解くことから始めなければならないが、ここではそれが目的ではないので解法は省略して次に解だけを示す。このような式が苦手な方は完全に読み飛ばしてもよい。

【解】 ある時間  $t$  [min] 後の、ある距離  $X$  [cm] の点の温度  $T$  [°C] は、次のように表わされる。

$$T = \frac{8T_0}{\pi^2} \sum_{n=1}^{\infty} \left( \sin \frac{1}{2} n\pi \right) \left( \sin \frac{n\pi}{L} X \right) \left[ \exp \left( -\frac{n^2 \pi^2}{L^2} k t \right) \right] \frac{1}{n^2} \quad (\text{eq. 1})$$

ここで  $k$  は棒の温度伝導率 [cm<sup>2</sup>/min] である。

## 1. 2 目標の設定

ここでは、上の解に基づいて幾つかの時刻での棒全体の温度分布を表わすグラフ (図10)、および温度分布と温度勾配を同時に表わすグラフ (図19) を作る。さらに、実験による測定値が得られたとして上のグラフ上に重ねてプロットする (図15)。

## 1. 3 用語の確認

### (1) プログラムについて

☆プログラムは全て DENNETSU.FORT77 というデータセットに適当なメンバ名をつけて登録するのでプログラムの区別はメンバ名だけで呼ぶ。また、例えばメンバ名 ABC を "memABC" のように略記する。同様にサブルーチン ABC を "subABC" と略記する。

☆ステートメントを呼ぶために行番号を使い、それを "L. 番号" と略記する。

☆説明の便利のために、行の左端に "井番号" をつけることがある。

☆図中でアンダーラインを施した部分は、端末から入力すべき部分である。

### (2) 図について

☆1枚の紙に描かれたグラフ (複数の場合もある) 全体を "絵" と呼ぶことにする。NLP の1ページに書かれたもの全体である。

☆1対の軸を持つ、いわゆるグラフを "グラフ" と呼ぶ。

☆ "絵" の中の適当な一部分を "ページ" と呼ぶ。"ページ" というのは、通常はここでいう "絵" を示す言葉として用いられるが、ここではサブルーチン名との関係で紛らわしい使い方になった。"ページ" は、プログラムの中で指定され、通常、軸の部分だけ、プロット部分だけ、コメント部分だけ等を意味することになる。

☆1枚の絵は、2種類の座標系を持つ。1つは、絵の左下隅を原点としたcm単位の座標系で、これをユーザ座標と呼び、ここでは変数の後にUをつけて、例えば(XU, YU)のように表わすことにする。もう1つは、ユーザが指定したX-Y軸によって構成される座標系で、これをグラフ座標と呼び、変数の後にGをつけて、例えば(XG, YG)のように表わすことにする。

### (3) 説明について

☆図形を画かせるプログラムの実例を説明している節のタイトルは枠で囲んで示す。

☆脚注は各章毎の通し番号とする。

## 1. 4 解の計算

解の計算値は、図2-1に示すmemNUMERICとmemSUBCALNMを結合して実行すれば得られる(※1)。解の計算部分は、グラフ化のプログラムでもそのまま使用できるように、別のメンバとした。

計算に当って、 $L=100\text{cm}$ 、 $T_0=70^\circ\text{C}$ 、 $k=12\text{cm}^2/\text{min}$ (鉄の値)を用いた。また、現象は中心点Cに関して対称だから、AからCまでだけを5等分し、 $X=10, 20, 30, 40, 50\text{cm}$ の各点について計算した。時間は、 $t=1, 5, 10, 20, 40, 80, 240\text{min}$ 後の各時刻について計算した。実行の様子、データ、結果を合わせて図2-2に示す。

### 【図2の説明】

memNUMERICは、殆ど入出力文だけからなっている。AL, T0, AKは、それぞれeq.1の $L, T_0, k$ に対応する。配列TIMに計算すべき時間を与える。NTIMEは、その数である。EPSは、eq.1の級数の加算を適当な所で打ち切るための指標である。L.600~1000は、計算条件の表示と計算結果の表の準備であり、図2-2の#3の部分を表示する。subCALNUMは、ある時間TIM(J)における棒の各点の温度の計算を行っている。このサブルーチンをDOループの中に置くことにより、次々に幾つかの時間の各点の温度を計算している。また、同じループの中で計算結果の出力#4(図2-2の#4の部分)を行っている。

---

※1 このプログラムは(問題、解法も含めて)、「化学工学協会編、化学工学プログラミング演習、培風館」に掲載された例題を基礎としている。

※2 Fコマンドについては文献5の第1章を参照せよ。

memSUBCALNM は、subCALNUM だけからなるプログラムである。ここではL.180までで、eq.1 に基づいてXの各点の温度を計算し、L.190~210 で温度を対称に折り返している。

これらの2つのメンバを結合して実行させるには、図2-2の#1のように入力すれば良い(※2)。ひき続いて#2のようにデータを入力すれば、#3、#4のような結果と、若干のメッセージが表示される。

```
READY
LIST DENNETSU.FORT77(NUMERIC)
AB9999.DENNETSU.FORT77(NUMERIC)
00100 COMMON/CNM/ SUM(0:10)
00200 DIMENSION TIM(10)
00300 READ(5,*) AL,TO,AK
00400 READ(5,*) NTIME,(TIM(J),J=1,NTIME)
00500 READ(5,*) EPS
00600 PRINT '(1H1,///,3(10X,A3,F5.0,A10,/) ,/,
00700 + 3X,A4,28X,A23,/,3X,A5,5(6X,A2,13,A3),/)',
00800 + 'L =',AL,' CM',TO='TO',C='K =',AK,' CM**2/MIN',
00900 + 'TIME',TEMPERATURE PROFILE (C)',
01000 + '(MIN)',('X=',1,' CM',1=10,50,10)
01100 DO 10 J=1,NTIME
01200 CALL CALNUM(TIM(J),AL,TO,AK,EPS)
01300 PRINT '(2X,F5.0,5F14.3)',TIM(J),(SUM(I),I=1,5)
01400 10 CONTINUE
01500 STOP
01600 END
END OF DATA
READY
LIST DENNETSU.FORT77(SUBCALNM)
AB9999.DENNETSU.FORT77(SUBCALNM)
00010 SUBROUTINE CALNUM(TIME,AL,TO,AK,EPS)
00020 COMMON/CNM/ SUM(0:10)
00030 PA1=3.14159
00040 P1=8.*TO/PA1/PA1
00050 P2=0.5*PA1
00060 P3=PA1/AL
00070 P4=P3*P3*AK
00080 DO 10 I=1,5
00090 X=I*10.
00100 AN=1.
00110 SUM(I)=0.
00120 20 EXN=P1*EXP(-AN*AN*P4*TIME)/AN/AN
00130 IF(EXN.LT.EPS) GO TO 10
00140 FN=SIN(P2*AN)*SIN(AN*P3*X)*EXN
00150 SUM(I)=SUM(I)+FN
00160 AN=AN+1.
00170 GO TO 20
00180 10 CONTINUE
00190 SUM(0)=0.
00200 DO 30 I=6,10
00210 30 SUM(I)=SUM(10-I)
00220 RETURN
00230 END
END OF DATA
```

図2-1 memNUMERIC とmemSUBCALNM および、その実行の様子(その1)

```

READY
#1 F GO DENNETSU EL('NUMERIC,SUBCALNM')
  FORT7CLG STARTED TIME = 14:02:52
  FORTRAN 77 COMPILER ENTERD
  END OF COMPILATION (OPTIMIZATION COMPILER)
  00300 ?
#2 100 70 12
  00400 ?
#2 7 1 5 10 20 40 80 240
  00500 ?
#2 0.01
#3
#3
#3      L = 100.      CM
#3      T0= 70.      C
#3      K = 12. CM**2/MIN
#3
#3      TIME          TEMPERATURE PROFILE (C)
#3      (MIN)        X= 10 CM      X= 20 CM      X= 30 CM      X= 40 CM      X= 50 CM
#3
#4      1.           14.004       28.005       42.000       55.889       64.518
#4      5.           13.997       27.975       41.585       52.997       57.757
#4      10.          13.936       27.561       39.988       49.212       52.691
#4      20.          13.231       25.606       35.991       43.021       45.521
#4      40.          10.846       20.682       28.555       33.653       35.419
#4      80.          6.798        12.931       17.798       20.922       21.999
#4      240.         1.022        1.944        2.675        3.145        3.307
END OF GO,SEVERITY CODE=00
MAX CODE=0 ENDED TIME=14:04:13
READY

```

図2-2 memNUMERIC とmemSUBCALNM および、その実行の様子 (その2)

## 第2章 グラフ出力までの手順

### 2. 1 基礎的プログラム

#### (1) プログラムの構成

図2のmemNUMERICで得られた計算結果をグラフ化する基礎的プログラムmemPLBASICを図3に示す。

実行は、memSUBCALNMと結合して行う。以後、特に断らない限り、実行はmemSUBCALNMと結合して行うものとする。

2. 2の方法で実行し、2. 3, 2. 4の方法で出力して得られたグラフを図4に示す。温度の高いものから、それぞれ時間  $t=1, 5, 10, 20, 40, 80, 240$ min 後に対応している。このような " $t=20$ min" 等のコメントを各曲線につけたい。その方法は後述する (ref. 3. 1)。

memPLBASICの行番号は、図2のmemNUMERICと対応している。したがって、L.10, 1050 等が新しく追加された行であり、L.200, 300 等の行は削除されたことがわかる。また、特に重要と思われる追加行には左端に " $\yen$ " マークをつけた。

#### 【memPLBASIC (図3) の説明】

☆L.1000までは型宣言文および入出力文である。

△memNUMERICとは異なり、入力データは全て PARAMETER 文または DATA 文で与えた (※1) (L.10, 410)。

△L.20, 220, 420, 430 は、グラフ化のためのデータである。詳細は後に説明する。

△L.600~1000 は、数値出力およびその準備である。数値出力が不要なら、この部分は省略する。

☆L.1010~1050 は、グラフを描くための準備である。

△L.1010, 1020 のCONDEV (文献1)、GRINIT (文献1) は、グラフを描こうとする場合、まず無条件にこの順序で呼ぶ。あらゆる GRACE IV サブルーチンの出現以前になければならない。

CONDEVの引数は例のように '6202' とすること。

---

※1 一般にこうする必要があるわけではない。



```

READY
LIST DENNETSU.FORT77(PLBASIC)
AB9999.DENNETSU.FORT77(PLBASIC)
00010    PARAMETER (AL=100., TO=70., AK=12., NTIME=7)
00020    PARAMETER (XL=AL/10.+1.)
00100    COMMON/CNM/ SUM(0:10)
00210    DIMENSION TIM(NTIME),XDT(3),IXF(3),YDT(2),IYF(3),IXF2(2),IYF2(2)
£ 00220    CHARACTER ENDCMT*20/'TEMPERATURE PROFILE'/
00410    DATA TIM/1.,5.,10.,20.,40.,80.,240./, EPS/1.E-2/
00420    DATA XDT/XL,0.,10./,IXF/-2,2,2/,YDT/0.,70./,IYF/2,2,2/,
00430    + IXF2/-2,2/,IYF2/11,2/
00600    WRITE(6,'(1H1,///,3(10X,A3,F5.0,A10,/),/,
00700    + 3X,A4,28X,A23,/,3X,A5,5(6X,A2,13,A3),/)' )
00800    + 'L =',AL,' CM','T0=',TO,' C','K =',AK,' CM**2/MIN',
00900    + 'TIME',ENDCMT//'(C)',
01000    + '(MIN)',('X=',1,' CM',1=10,50,10)
¥ £ 01010    CALL CONDEV('6202')
¥ £ 01020    CALL GRINIT
¥ £ 01030    CALL PLOTS(0.,0.,16)
¥ 01040    CALL GRXAXS(4.,3.,19.,XDT,IXF,-2,'DISTANCE (CM)',13)
¥ 01050    CALL GRYAXS(4.,3.,15.,YDT,IYF,2,ENDCMT(1:12)//'(C)',15)
01100    DO 10 J=1,NTIME
01200    CALL CALNUM(TIM(J),AL,TO,AK,EPS)
01300    WRITE(6,'(2X,F5.0,5F14.3)') TIM(J),(SUM(1),I=1,5)
¥ 01310    CALL GRPGN1(XDT,SUM,IXF2,IYF2,0)
01400    10 CONTINUE
¥ £ 01410    CALL PLOTE(ENDCMT)
01500    STOP
01600    END

```

END OF DATA  
READY

```

#1 F CL DENNETSU EL('PLBASIC,SUBCALNM') PLOT(G)
FORT7CLG STARTED TIME = 14:31:12
#2 ENTER LOGICAL UNIT(FOR PLOT) NO. AND MEMBER NAME
#3 16 N01
FORTRAN 77 COMPILER ENTERD
END OF COMPILATION (OPTIMIZATION COMPILER)
LINKAGE START ***LIB('APPF.LINKLIB')***
**MEMBER NAME** TEMPMN WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
#4
#4
#4          L = 100.          CM
          ..... (数值出力中略) .....
#4 240.          1.022          1.944          2.675          3.145          3.307
#5 MAX CODE=1 ENDED TIME=14:33:08
READY

```

図3 memPLBASIC とその実行の様子

△L.1030のPLOTS (文献2) は、1枚の絵を描く最初に呼ぶ。このPLOTSからL.1410のPLETEまでが、1枚の絵に対応する。その後、またPLOTSを呼べば、改ページして次の絵を描き始める。PLOTSの引数は、例のままにするのが無難である。

△L.1040, 1050のGRXAXS, GRYAXSは、それぞれX軸、Y軸およびその目盛、標題等の作画である。詳細は後に示す。

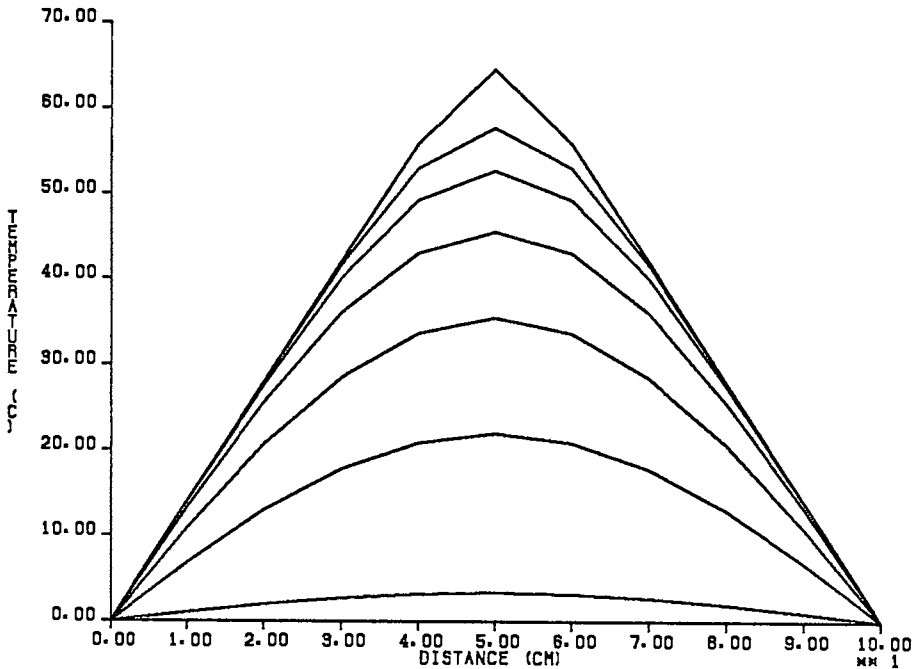


図4 memPLBASIC により作画された絵 (縮尺1/2)

☆L.1100~1300 およびL.1400は、memNUMERICのままである。各時間の温度分布を計算し、数値出力している。数値出力が不要ならば、L.1300は省略する。

☆L.1310のGRPGN1が得られた計算結果をプロットしているところで、与えられたデータを折れ線グラフにする機能を持つ。詳細は後述する。

☆L.1410のPLETE (文献2) は、1枚の絵の最後に必ず呼ばなければならない。引数ENDCMTは、20文字の文字列である(※2)。この文字列は、後に検索の際に参照されるのみで大きな意味はない。

☆結局、グラフを書かせるプログラムには、“ $\$$ ”をつけた部分が、この順序で必ず実行されなければならない(※3)。

## (2) 主要サブルーチンの詳細

サブルーチンの説明は、以下全て、次の順序で行う。

### サブルーチン名 (文献番号)

- 1 機能
- 2 呼出し形式
- 3 引数の説明
- 4 具体例との対応
- 5 補足

### (ア) GRXAXS (文献1)

- 1 X軸およびその目盛、目盛値、標題の作画を行う。また、絵が複数のX軸を持つ場合には、特定のX軸の指定のためにも用いられる。
- 2 CALL GRXAXS (XU, YU, AL, DATA, IFORM, K, ILAB, NCHAR)
- 3 ☆XU, YU……X軸の始点 (原点ではない) のX, Y ユーザ座標値。単位はcm (※4)。  
目盛や標題のスペースを考慮して、YUは1.5以上にするのが良い。  
☆AL……X軸の長さ。単位はcm。但し、1枚の絵のサイズは25cm × 25cm 以下に

---

※2 直接引用符で括って文字列を書いても良いが、その場合はぴったり20文字にしないと、余白に勝手な文字列が入り込む。例のように、20文字の CHARACTER 宣言 (L.220) をすれば余白には正しく空白が入る。

※3 文献7では、OPNDEV, BGNDEV, RSTDEV, CLSDEVをこの順に実行させることになっているが、“Ⓜ”をつけた部分は上記のサブルーチンのうちのプロット使用の際に必要な部分だけを抜き出したものである。従って、既に上記4サブルーチンを使用しているユーザは、BGNDEV, RSTDEVを省略し、OPNDEVの代わりにCONDEV, GRINIT, PLOTSを、CLSDEVの代わりにPLOTEを使用すれば良い。但し、その場合BGNDEVはBGNDEV (0., 25., 0., 25., 25., 25.) としたことになる。  
また、RSTDEVはRSTDEV (-1., 1.) としたことになる。  
これらの数値を変更したい場合は、PLOTSの引数を変えなければならない。その方法は、ここでは割愛しユーザの独習にまかせる (ref. 文献2)。

※4 厳密には正しくない。それはinch単位からの換算のためである。以下、単位をcmとしている場合は同様である。正確には“1cm”=254/240cmである。

するのが無難である(※5)。目盛値のスペースを考慮して(XU + AL)は、24以下にするのが無難である。

☆DATA, IFORM……対で、目盛および軸の種類に関する情報を与える。

△ DATA ……配列である。目盛に関する情報を与える。データ型はIFORMで与えられる。情報の与え方には、主として次の(a) (b) 2通りがある。

(a) 数値増分形式 (IFORM(1)=-2の場合)

1つは、予め目盛が決められる場合で、目盛の初期値(始点の目盛値)と増分値および目盛の数を表1のように与えれば良い。

表1 配列DATAの与え方(数値増分形式)

D A T A	内 容
(1)	目盛の数
(2)	目盛の初期値
(3)	目盛の増分値

したがって、DATAの配列の大きさは3である。また、軸の終点の目盛値は $DATA(2) + DATA(3) \times \{DATA(1) - 1\}$

となる。このような情報の与え方を数値増分形式と呼ぶ。

なお、数値増分形式で対数軸に関する目盛を決める場合、目盛の増分値には意味が無くDATA(3)は無視される。

(b) 配列形式 (IFORM(1) ≥ 2の場合)

もう1つの方法は、予め目盛が決められない場合などに用いられるもので、プロットすべき全データを、そのままDATAとして与えれば良い。DATAの配列の大きさは、データ数となる。このような情報の与え方を配列形式と呼ぶ。

配列形式で情報を与えると、計算機が適当に目盛を決めてくれる。実際には、データのうちの最大値と最小値のみから目盛を決めているようなので、この2つさえ与えれば良いようである。したがって、数値増分形式で与える場合も最大値、最小値のみを与える配列形式で与えた方が便利なこともある。

---

※5 図形をNLPへ出力する際にXYパラメータで“D=L”とすればX軸側は34cmまで画くことができる。

△IFORM……整数配列であり、大きさは3である。上記の配列 DATA に関する情報と軸の種類を与える。それぞれの与え方は、表2の通りである（主なものだけ）。

表2 整数配列 I F O R Mの与え方

I F O R M	意 味	区 別	値
(1)	DATAの形式	配列形式 数値増分形式	DATAのデータ数 - 2
(2)	DATAのデータ型	整数値 実数値	1 2
(3)	X軸の種類	整数軸 実数軸 対数軸	1 2 3

☆K……目盛、目盛値、標題の作画に関する指定である。意味は、表3の通りである。

表3 軸の始点から終点を見た時のKの与え方

軸の右側に目盛、目盛値、標題を作画する	- 2
軸の左側に目盛、目盛値、標題を作画する	2
軸の右側に目盛、標題を作画する	- 1
軸の左側に目盛、標題を作画する	1
何も作画しない	0

K=0 の場合、何も作画しないがグラフ座標の設定は行なう。したがって、複数の座標軸から特定の1本を指定する場合に用いられる。（ref. 3. 5）。

☆ I L A B , N C H A R……対で標題を与える。

△ I L A B ……標題の文字列である。

△ N C H A R……標題の文字数である。

- 4 L. 1040で使用されている。X軸は、ユーザ座標系の(4, 3) cmを始点とし、19cmの長さを持つ。

目盛は、数値増分形式で与えてあり(L. 420)、初期値は0で10単位で、ALまで目盛られている(L. 20で目盛数XLがAL/10.+1. になっていることに注意)。なお、図4でX軸の右端に“\*\*1”と書かれているのは、“×10の1乗”という意味である。

標題は、“DISTANCE L (CM)”の13文字である。

目盛、目盛値、標題が軸の右側（下側）に作画（ $K=-2$ ）されている。

- 5 ここで述べた3種類の軸の他に、項目軸、年月軸、極座標軸、年期軸、多角形軸が作画できる（ref. 文献1）。

(イ) GRYAXS（文献1）

- 1 GRXAXSと同様であり、Y軸に関するものである。

したがって、説明はY軸だけに特有な若干の補足にとどめる。

- 3 ☆XUは、X軸の場合と同様の理由で3.5以上が良い。

☆(YU+AL)は、同様に24.5以下が良い。

- 4 L.1050で使用されている。Y軸は、ユーザ座標系の(4,3)cmを始点とし15cmの長さを持つ。当然のこととして、X軸の始点とY軸の始点は一致する必要はない。

目盛は、配列形式で最大値、最小値のみを与えた(L.420)。目盛の間隔は、計算機が適当に決めたものである。

標題は、ENDCMT文字列(L.220)の部分列と'(C)'をつなぎ合わせたものである。ENDCMT(1:12)と'(C)'の間の//は、文字列と文字列をつなぎ合わせる符号(連結演算子)である。

目盛、目盛値、標題は軸の左側に作画( $K=2$ )されている。

ここで標題の文字が縦書きに作画されていることに注意されたい。そのため“( )”は、その態をなしていない。Y軸の標題文字を横書きで縦方向に作画する方法は後述する(ref. 3. 2)。

(ウ) GRPGN1（文献1）

- 1 折れ線グラフを作画する。

- 2 CALL GRPGN1(XDATAG, YDATAG, IXFORM, IYFORM, ISYMBL)

このサブルーチンを呼び出す前にGRXAXSおよびGRYAXSが呼び出されていなければならない。

- 3 ☆ 

XDATAG, IXFORM
YDATAG, IYFORM

 …それぞれ対でプロットすべき X, Yデータを与える。

両者の関係は、GRXAXSのDATAとIFORMの関係と同様である。

△XDATAG, YDATAG……配列である。プロット点のX, Yグラフ座標値を順序正しく与える。計算機は一切並べ換えをしない。与え方は、数値増分形式、配列形式など。

XDATAGとYDATAGで与え方の形式が違っていても、順序さえ対応していれば良い。

与えられた点は、順序通り直線で結ばれる。

△IXFORM, IYFORM……整数配列であり大きさは2である。大きさが異なる以外は GRXAXS のIFORMと同じである。(IFORM(3)に相当するものは必要がない。)  
 ☆ISYMBL……各プロット点に作画したいシンボルを指定する。  
 指定は、表4のように行う。

表4 ISYMBLの与え方

ISYMBLの値	シンボル
0	シンボルの作画はしない
1	△
2	×
3	+
4	□
5	◇
'文字'	文字

但し、1～5で指定されたシンボルはプロットすべき点を中心にして作画されるが、文字は、その点のすぐ右上に作画されるので注意を要する。

- 4 プロット点のX座標は数値増分形式で、Y座標は配列形式で与えてある(L.420,430)。  
 また、計算点のプロットは必要としないので、ISYMBLは0としてある。  
 もう少し滑らかな曲線にするには、プロット点即ち計算点をふやせば良い。この計算点のまま滑らかな曲線を引く方法は後述する(ref. 3. 3)。
- 5 シンボルの作画サイズの標準値は0.2cmである。作画サイズを変更するには、sub GRSSIZ (文献1)を用いる。

## 2. 2 実行および図形検索 (グラフ用プログラムからグラフィックデータセットまで)

さて、プログラムの編集が終わったら実行に移る。実行・出力の方法には、図5に示すように多くの方法がある。

通常はTSSでは、F PLOT(F) (経路&g)から、PSPNLP (経路&i)の方法、またバッチでは、FB PLOT(16) (経路&k)の方法が便利であろう。

ここでは、図形が最初から思うように書けない場合も考えて、図形検索 (英数字出力の際のSORPに当る) を経てNLPへ出力する方法を中心にして述べる。

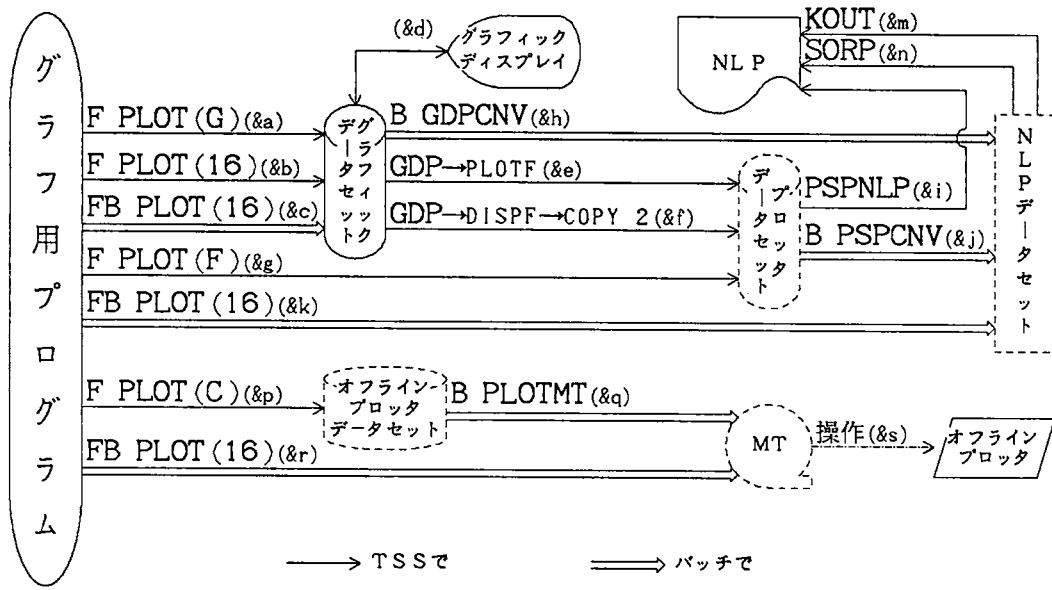


図5 グラフ用プログラムの実行・出力方法

★図5の注

- &a ref. 2. 2 (1) (ア) および文献5の第1章
- &b ref. 3. 7 および文献5の第1章
- &c ref. 2. 2 (1) (イ) および文献6の第4章
- &d ref. 第4章, 2. 2 (2), 文献3, および文献5の第3章
- &e ref. 2. 3 (1), 文献3, および文献5の第3章
- &f ref. 4. 3, 2. 3 (1), 文献3, および文献5の第3章
- &g ref. 2. 3 (2) (ア), 文献5の第1章
- &h ref. 2. 3 (1) および文献5の第2章
- &i ref. 2. 3 (1)
- &j ref. 2. 3 (1) および文献5の第2章
- &k ref. 2. 3 (2) (イ), 文献6の第4章
- &m ref. 2. 4 および文献17の第6章
- &n ref. 2. 4 および文献9の第4章
- &p ref. 2. 5 および文献5の第1章
- &q ref. 2. 5, 文献5の第2章 および文献8の1
- &r ref. 2. 5, 文献6の第4章 および文献8の1
- &s ref. 文献8の1



なお、図形検索は、グラフィックディスプレイ上でしか行えないので、実行経路は、必ずグラフィック・データセットを経由するものでなければならない。ここでは、そのうち F PLOT (G) (経路&a) または FB PLOT (16) (経路&c) から B GDPCNV (経路&h) の方法について述べる。

図形検索を必要としない場合、この節は読み飛ばして直接 2.3 の (2) へ進んでも構わない。

## (1) 実行

### (ア) TSS で (F PLOT (G))

一般に、TSS でグラフ用プログラム AB.FORT77 (CD) を実行し、グラフィックデータセットを作るには次のようにする (図5の&a) (※6)。

```
F CL AB(CD) PLOT(G) [IN (EF.DATA(GH))] [OUT(IJ)]
```

[ ] 内は省略可能である。

グラフを作画させるには、大きな容量を必要とするので OP パラメータとして CLG (簡略形 CL) を指定した方が無難である。

後半部 IN 以降は、実行に際して EF.DATA (GH) からデータを読み込み、数値出力を IJ.OUTLIST に書き込ませる場合である。実行の様子を図3の後半部に示す。

### 【図3の後半部の説明】

ここでは2つのメンバを結合して実行しなければならないので、図3の#1のようにする。すぐに#2のようなメッセージが表示されるので、“16”と“適当なメンバ名”を入力する(#3) (※7)。このメンバ名は PLOTS と PLOTE に挟まれた1枚の絵に与えられる名前である (※8)。いくつかのメッセージが表示され、しばらくたってから、数値出力がある場合は、それを出力し (#4)、実行が終了すると#5を表示し、READYを表示して止まる。

---

※6 F コマンドについては、文献5の第1章を参照せよ。

※7 既存のメンバ名を指定してはいけない。

※8 メンバ名は、1つしか指定できない。したがって、1つのプログラムの実行で複数枚の絵を描こうとする場合には、別の手法を用いなければならない。その手法については後述する (ref. 3. 7)。

(イ) バッチで (FB PLOT (16))

一般に、バッチで (ア) と同じことをするには、TSS 端末より、次のように打ち込む (図5の &c) (※9)。

```
FB CL AB(CD) PLOT(16) [IN(EF(GH))] [R(1024K)]
```

PLOT(16) が、グラフ用プログラムを実行するためのパラメータである。また、グラフを描くには、大きな容量を必要とするので R(1024K) とした方が安全である (※10)。

実際の実行の手順を図6に示す。

```
READY
#1 FB CL DENNETSU.FORT77(PLBASIC)/DENNETSU.FORT77(SUBCALNM) PLOT(16) R(1024K)
#2 ENTER OUTPUT DEVICE (NLP | GDP | MMB | OFF) : GDP
#3 ENTER DATASETS NAME(MEMBER NAME) : ONLINE.OUTLIST(FIRST)
#4 JOB AB9999S(JOB02979) SUBMITTED
  ENQC00* AB9999S  ACCEPTED CLASS=C,IN/JOB=AB9999          CN(02)
  READY
#5 17.25.23 JOB 2979 JEM1651 AB9999S ENDED AT N1 CN(02)
  READY
```

図6 バッチでグラフィック・データセットを作る手順

【図6の説明】

ここでは2つのメンバを結合して実行するので#1のようにする。通常はデータセット名は1つだけで良い。

#1を入力すると、#2の前半の部分を表示してカーソルは:の後ろで止まる。どこにグラフ用のデータセットを作るか聞いているので、この場合はGDPと入力する。次に#3で、そのデータセットにつける名前を聞いてくる。GDPの場合のデータセット名は、ONLINE.OUTLISTとすると、後で便利である (ref. 2. 3)。メンバ名は、区別し易いように適当に指定する。そうすると計算機は適当なJCL文を作って、それをサブミットし (#4)、実行が終了すると#5のようなメッセージを表示して知らせてくれる (※11)。

実行が終了すれば、SORPで出力検索をすることになるが、ここにはグラフィックデータセット

※9 FB コマンドについては、文献6の第4章を参照せよ。

※10 また同様の理由でTSSで実行する場合には、LOGON時にサイズ指定SIZE(2048)をした方が安全である。但し図2程度のグラフならばリージョン指定は必要ない。

は現われない(※12)。したがって、実行が正常に終了しているかどうかだけを見て、正常なら出力は全部消してしまってもかまわない。グラフィック・データセットの検索は、次に示す別の方法で行う。

## (2) 図形検索(図5の&d)

図形検索は、通常のディスプレイ装置では行なえない。グラフィック・ディスプレイ装置(F9434 F9431, TEK4114)(※13)を用いて行う。

ここでは、F9430シリーズを用いる場合を中心に説明する。

図形検索に用いるGDPモード(※14)は、絵の一部分を取り出したり、縮小・拡大したり、またそれらを重ね合わせたりという機能など、他にもいくつかの有用な機能を持つが、それらの機能については、後述(第4章)することにして、ここでは図形が正しく書かれているかどうか、またそれをNLPへ出力するかどうかの判断だけを行うことにする。

図形検索の手順を図7に示す。

- 
- ※11 但し、このメッセージはユーザから計算機に何かを送信した時などに表示されるもので、ただじっと待っていても表示してくれない。ただ、[ENTER]キーを押すだけでも良い。
  - ※12 #2のところではNLPとすれば、SORP上には出力クラスKで現われる。但し、覗いても訳のわからない符号が並んでいるだけである。
  - ※13 センター特殊端末室にある。F9431, F9434、画面の大きさとハードコピー装置が異なる以外は、ここでの機能としては同じ。但し、F9431のキーボードは若干見にくい。TEK4114(操作法は文献13を参照せよ)は、F9430シリーズに比較して解像度が高い。通常の図形検索は、ハードコピー料金を考慮してF9431で行うのが良いであろう。
  - ※14 GDPについては、文献5の第3章および文献3を参照せよ。

```

READY
#1 GDP
#2 MODE IN=FT01F001,OUT=FT04F001,OUTDEV=TEK
FACOM OSIV/F4 GDP V02L20 DATE 82.05.26 TIME 14.50.49
PAGE 1
MODE IN=FT01F001,OUT=FT04F001,OUTDEV=TEK
#3 PRINTF +
PRINTF +

GRAPHIC FILE LISTER LISTFM
DATE 82.05.26 PAGE 1
MEMBERNAME DEVTYPE SIZE MAKE-DATE COMMENT
ACCUM
1 DECO1 PSP 12288 82.05.26 TEMP.PROF.DECO
AB9999 934 601 23 130 0
2 FIRST PSP 4608 82.05.26 TEMPERATURE PROFILE
AB9999S 292 157 23 89 0
3 NO1 PSP 4608 82.05.26 TEMPERATURE PROFILE
AB9999 292 157 23 89 0
#4 DISPF FIRST
DISPF FIRST
図形表示
#5 END

```

図7 図形検索の手順 (F9430 シリーズの画面例)

【図7の説明】

＃1、＃2は、そのまま入力する。F9430シリーズの場合もOUTDEV=TEKと入力する。しばらく入力禁止の状態となるので注意を要する。入力禁止の状態ではカーソルは入力した行の先頭にとどまったままである。このカーソルが次の行へ移動したら入力可能となる。

なお、＃2は、先頭カラムから1カラム以上さがって入力しなければならない。以下、＃3、＃4、＃5も同様であり、この規則は＃1でGDPを入力してから、このモードが終るまで（即ち、次にREADYが表示されるまで）適用される（※15）。＃2を入力後、若干のメッセージが表示され、カーソルが次の行で止まるので、＃3を入力する。

＃3を入力すると、ユーザが、自分のグラフィック・データセット上に現在保管しているメンバの名前等が表示される。図7の例の場合、DECO1とFIRSTとNO1の3つのメンバが存在していることを示している。（メンバ名はABC順に表示される）メンバ名のあとに、いくつかの情報が表示され、その行の最後（※16）には、COMMENTが表示される。これが、前述のPLOTE (ref.

※15 但し、DISPFモード（会話モード）は除く。

2. 1 (1)) で与えたコメントである。

したがって、いくつもの同じようなプログラムを実行させる場合は、このコメントまたはメンバ名で区別できるようにしなければならない。また、このコメントが正しく表示されていれば、そのプログラムは少なくともPLOT Eまでは実行されたことを示している。このコメントの後に、さらに情報の表示が続くが、それは改行されて次の行に表示される(※16)。表示の意味は割愛する。

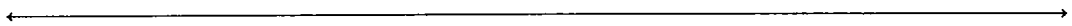
図形検索すべきメンバの存在が確認されたら、#4を入力する。#4の行が再表示された後、画面は、一旦全部消去され図形の表示が始まる(TEK4114では左下部の入出力域も再表示される。従って図形と重なることがある)。プログラムが途中までしか実行されていない場合でも、そこまでは表示してくれる。これがNLPへの出力ともしっかり異なる点である。

但し、F9430シリーズのグラフィック・ディスプレイは解像度が悪いので、細部までは明瞭には表示されない(※17)。高解像度が要求される場合には、TEK4114を用いると良い。

図形を表示し終わったらカーソルは左上で止まる(TEK4114では、左下である)。ここで、いくつかの会話モードのコマンドを入力することができるが、それについては後述する(ref. 第4章)。表示結果を検討した後にENDと入力し、DISPFモード(会話モード)を終らせる。図形表示は消え、カーソルはまた左上に現われる(TEK4114では、左下である)。ここで#5を入力し、GDPモードを終らせる。グラフが正しく描かれていれば、NLPへ出力すれば良い。その方法を次節に示す。

但し、グラフィック・ディスプレイ上にはあっても、25cm × 25cmの寸法を超えた部分はNLPには出力されない(※5)。寸法内に納まっているかどうかは、DISPFモード(会話モード)でスケールを表示させて調べれば良い(ref. 4. 4)。

なお、表示画面のハードコピーを取りたい場合は、[SHIFT] キーを押しながら[PRINT] キーを押す(TEK4114では右上隅の[HARD-COPY] キー、またはハードコピー装置の[ COPY ] キーを押す。前法では、図形と左下の入出力部分4行が、後法では図形のみがコピー出力される)。



※16 F9430シリーズの画面に表示しうる1行の文字数は80文字であるにもかかわらず、表示すべきデータセットの1行の文字数は、それを超えている。そのため1行分の表示をするのに画面上では2行が使用される。また、TEK4114では、表示行が左下部分に4行分しかないので、その以上の行数はスクロール・アップしてしまう。但し、右下のローラーによって簡単にスクロール・ダウンまたはアップすることができる。

※17 グラフィック・データセットには正しく書かれているので、これをNLPへ出力すれば細部も明瞭となる。

画面がそのまま、すぐそばのグラフィック・プリンターに出力される。使い終わったら使用枚数を、近くのオフライン使用申込み書に書き入れる。F9434およびTEK4114のハードコピー使用料金は、F9431に比べて割高である。

## 2. 3 NLP 出力準備 (NLP データセットへ)

(1) 図形検索を経由した場合 (グラフィック・データセットから NLP データセットへ)

○グラフィック・データセットの全メンバ (ここではDECO1, NO1, FIRST) を NLP に出力するには

B GDPCNV

とする (図5の&h) (※18)。すぐにグラフ出力の際のXYパラメータ (※19) を聞いて来るので (図8の#2)、変更がある場合は適当に修正し (図8の#3) (※20)、変更がなければYESと打ち込む (図8の#4)。図8の#9のようなメッセージが出力されて、ジョブは自動的にサブミットされる。

○また、グラフィック・データセットの一部のメンバだけを NLP に出力するには、GDP モードを終らせる前に

└PLOT┘メンバ名, OUT=FT02F001

としてグラフィック・データセットの必要なメンバだけをプロッタ・データセット (GDPOUT.OUTLIST となる) に書き換えてから (図5の経路&e)、GDP モードを終わらせ

PSPNLP GDPOUT.OUTLIST

または

B PSPCNV D(GDPOUT) [R(1024K)]

とする (図5の経路&i, &k (※21))。前法の場合、SORP や出力要求なしに直接 NLP へ出力される。但し、その際のXYパラメータは標準のものに固定されている。後法の場合は、B GDPCNVと同様にXYパラメータを変更することができる。

○グラフィック・データセットの一部のメンバだけを NLP に出力するもう一つの方法は、DISPF

---

※18 詳細は文献5の2. 3を参照せよ。

※19 詳細は文献14の2. 1を参照せよ。

※20 #2の行へカーソルを移動して直接修正した方が便利である。



### 【図8の説明】

まず、#1のように打ち込む。

すぐにXYパラメータを聞いて来るが(#2)、これに対する応答は上述した。XYパラメータの変更がなくなってYESと打ち込むと(#4)、サブミットするべきJOBストリームがEDITモードになるので(#5)、“PLOT F L+”の“+”の代わりにNLPに出力したいメンバ名を入力する(#6) ※24)。NLP出力したいメンバが複数個ある場合は#7のように必要な個数だけL.120を複写して修正すれば良い ※25)。

それをサブミットし(#8)、最後にEDITモードを終らせる(#10)。

(2) 図形検索を経由しない場合(グラフ用プログラムの実行でNLPデータセットへ)

(ア) TSSで(F PLOT(F)→PSPNLP)

一般に、TSSでグラフ用プログラムAB.FORT77(CD)を実行し、プロッタデータセットに書き込むには、次のようにする(図5の経路&g)

```
F CL AB(CD) PLOT(F) [IN(EF.DATA(GH))] [OUT(IJ)]
```

キーオペランドPLOTの値をFに変更すること以外は、2.2(1)(ア)で示した、経路&aと同じである。

すぐに図3の#2のようなメッセージが表示されるので、“16”と“適当なデータセット名(ユーザ指定名のみ)” ※26)を入力する(図3の#3)。これは、プロッタデータセットの名前であり、その名前は“適当なデータセット名”。OUTLISTとなる。あとは図3と同様である。

プロッタデータセットをNLP出力する方法およびNLPデータセットに変換する方法は2.3

---

※23 GDPOUT, OUTLISTは順データセットであり、複数の図形がある場合は順番に入っているだけで、メンバ名はついていない。

※24 グラフィック・ディスプレイではFSモードは使えないので図8のような面倒な方法をとった。この部分を通常のディスプレイで行う場合は、もちろんFSモードで修正すれば良い。このようなFSモードの使えないEDITモードのサブコマンドについては、文献15または文献16を参照せよ。

※25 L.129まで挿入できる。

※26 PLOT(G)の場合は、“適当なメンバ名”であったが、プロッタデータセットは順データセットなので、PLOT(F)では“データセット名”となる。但し、ユーザが特別に意識して区別する必要はない。



(1) で既に述べた (図5の経路&i と&j)。但し、プロットデータセット名 GDPOUT.OUTLIST の代わりに、上で指定した“適当なデータセット”、OUTLIST とする。

(1) バッチで (FB PLOT (16)) (図5の経路&k)

コマンドは、2. 2 (1) (1) と全く同じである (図6の#1)。同様に図6の#2の前半のメッセージが表示されるので、今度は、NLP と答える。そうすると、図6とは異なり、データセット名は聞いてこず、図8の#2と同趣旨のメッセージが表示されるので、応答してやれば図6の#4のようなメッセージが表示され、ジョブがサブミットされる。

## 2. 4 NLP 出力要求 (NLP データセットを NLP 出力へ)

B GDPCNV や FB PLOT (16) 等の NLP データセット上に図形を作るジョブが終了したら、出力検索 SORP (※27) を行う。

SORP コマンド入力後、メニュー画面が現われたら

DISP

と入力して、クラスKの出力があるかどうかを確認する。このクラスKで出力されているのが図形である。但し、FB PLOT (16) で実行した場合には、図形に関する情報もクラスKとなっている。

K クラス出力が無い場合は、絵が出力されなかったのであり、X クラス出力の最初に戻って理由を検討することとなる。

既に述べたように、K クラス出力があっても、図形に関する出力検索は SORP では行なえない。覗いてみても、訳のわからない符号が並んでいるだけである。しかし、K クラス出力の1つ前のデータセット (FB では2つのK クラス出力のうち前の方) には、図形に関する情報が書かれているので参考にできる。詳細な説明は割愛するが

\*\*\*\*\* GDFILE INFORMATION LIST \*\*\*\*\*

の次の行の TOTAL NUMBER (※28) が1枚の絵を書くのに要した記憶容量を表わしていると考えて良く、この数値が予期した値より著しく小さければ (経験により予期できるようになる)、絵の一部しか作画されていないと考えられる。逆に絵が複雑すぎて割り当てられた記憶容量をオーバーしてしまった場合には、その旨を示すメッセージが表示される (※29)。

---

※27 詳細は文献9の第4章を参照せよ。

※28 TOTAL NUMBER は通常の画面の右外側にはずれているので、見にくい。そのかわりに、通常の画面に現われているNUMBERを見ても通常の用は足りる。

また、複数枚の絵を出力した場合は、同様な情報が複数組繰り返されているはずである。

この図形に関する情報が予期に反していた場合も、Xクラス出力の最初に戻って理由を検討することとなる。

さて、成功にしろ失敗にしろ作画した絵をNLPに出力するには、絵の書かれている出力をデータセット番号(※30)で指定して、例えば

```
KEEP  5
```

と入力し、必要な出力だけを確保して

```
END  RD
```

でSORPを終らせ、トークンカードリーダーで出力要求をすればよい。こうして得られたのが図4である。クラスK以外の出力も同時にKEEPした場合は、NLP出力はクラス別に2つに別れて出て来るので注意を要する。

また、SORPを必要としない場合には、KOUTコマンド(※31)を利用して、次のように入力する。

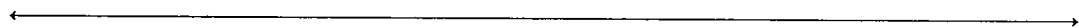
```
KOUT  ジョブ識別子
```

ここでジョブ識別子とは、サブミットジョブにつけられた名前(課題番号の最後に英字1文字を付加したもの)の最後の1文字のことであり、通常はSである。

上のコマンドを入力すると、トークンカードリーダーからの出力要求なしで、出力クラスKのものだけがNLPへ出力される(図5の終路&m)。

## 2. 5 オフラインプロッタ出力

NLP出力は非常にきれいなものだが、オフラインプロッタは、更にきれいに図形を描いてくれる。また、インクペン(例えばロットリングなど)で書いたり、カラーで書いたりすることもできる。さらに大寸法の図形も描くことができる。84.1×59.4cmまで。ただし、手数と所要時間は極端に増加する。また、線の太さを変えるには、subCONLIN(ref. 3. 1)を使うよりも、subNEWPEN(※32)を使う方が有効である。このサブルーチンを用いれば、3本(※33)



※29 図8の#3の入力のときに、記憶容量を若干増加させるための指定がある。詳細は、文献14の2. 1を参照せよ。

※30 DISPと入力した後の画面でDSNOで示されている番号。

※31 文献17の6. 3を参照せよ。

※32 文献2参照。

のインクペンを使うことができる。カラーについても同じで、このサブルーチンを用いれば3色（※33）のカラーを使うことができる。

以下、簡単にオフラインプロッタへの出力方法について述べる。

グラフ用プログラムの実行は、TSS では

```
F CL AB(CD) PLOT(C) [IN(EF.DATA(GH))] [OUT(IJ)]
```

と入力する（図5の&p）。AB(CD), EF.DATA(GH), IJ等は第2章2節の説明と同様の意味である。引き続いて同節(1)(ア)と同様に"16"と"適当なメンバ名"を入力すればオフラインプロッタデータセットが作られる。

次にMT（磁気テープ）を用意して（※34）

```
B PLOTMT D(メンバ名(※35))
```

と入力して、オフラインプロッタデータセットをMT上に移す（図5の&q）（ref. 文献5, 8）。このMTを用いてオフラインプロッタに作画させる（図5の&s）（ref. 文献8）。

また、バッチで実行するには、まずMTを用意して（※34）、第2章2節(1)(イ)と同様に

```
FB CL AB(CD) PLOT(16) L(@APPC.LINKLIB) [IN(EF(GH))]  
R(1024K)
```

と入力する（図5の&r）。Lパラメータについては文献5を参照せよ。この場合はメンバ名等は聞いてこない。

あとの操作はTSSの場合と同様である。

---

※33 若干の操作の工夫をすれば、3本あるいは3色以上のペンを使うこともできる。

※34 個人のものがないユーザは、センタに用意されているものを一時的に使うことができる。

※35 オフラインプロッタデータセットを作った時に"16"といっしょに与えた"適当なメンバ名"である。

## 第3章 グラフの工夫 (GRACE IV 諸機能の実例集)

### 3. 1 グラフに枠、リファレンス線を書き、文字を入れる

プログラム (memPLDECORA) を図9に、出力結果を図10に示す。

```
1 0 ~ 2 1 0      m e m P L B A S I C ( 図 3 ) と 同 じ
000220          CHARACTER ENDCMT*20/'TEMP.PROF.DECO',//TCHA*3
4 1 0 ~ 1 0 3 0  m e m P L B A S I C と 同 じ
¥ 001031        CALL CONLIN(1,0,0.)
1 0 4 0 ~ 1 3 1 0 m e m P L B A S I C と 同 じ
¥ 001320        CALL CBCRE(TIM(J),-1,TCHA,3,1)
001330          CENTX=0.5*AL
¥ 001340        CALL GRTRCU(CENTX,SUM(5),XUS,YUS)
¥ 001350        CALL CBTXTC(XUS,YUS,'TIME='//TCHA//'MIN',11)
001400          10 CONTINUE
¥ 001401        CALL GRFRAM
¥ 001402        CALL CONLIN(0,0,0.)
¥ 001403        CALL GRGRID(2)
¥ 001404        CALL CONLIN(0,3,1.)
¥ 001405        CALL GRREF(1,CENTX)
¥ 001406        CALL CONCHA(0.42)
¥ 001407        CALL CBTXTC(9.,1.,'TEMPERATURE PROFILE',19)
1 4 1 0 ~ 1 6 0 0 m e m P L B A S I C と 同 じ
```

図9 memPLDECORA

プログラムは、図3のmemPLBASICを基礎としており、それとの変更部分だけを抜粋して書いた。やっとグラフらしくなったので、この後第2節で若干の修正をしてmemPLDECORBとし、以後は原則としてこのプログラムmemPLDECORBを基に、必要な部分のみを変更することとする。なお、数値出力部分は、プログラムの単純化のため全て削除した。

#### (1) 枠を書く

GRFRAM (文献1) (L.1401) の機能である。X軸、Y軸を決めた後、適当な所で呼べば良い。

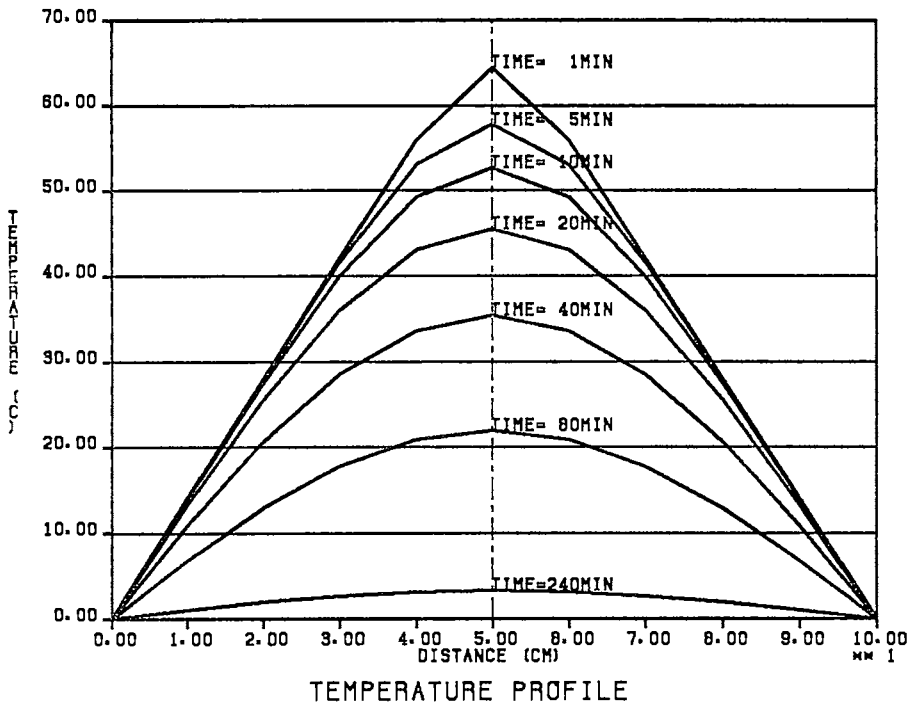


図10 memPLDECORAにより作画された絵(縮尺1/2)

(2) リファレンス線を引く

GRGRID (L.1403) またはGRREF (L.1405) の機能である。

GRGRID は、全ての目盛からリファレンス線を引く(図10のY軸)、GRREFは、指定された値のところだけにリファレンス線を引く(図10のX軸)。

また、リファレンス線と温度分布曲線の太さを変えたいので、線の太さと種類を変える subCONLIN を使った(L.1031,1402,1404)。

(3) 適当な位置に文字を入れる

文字列を作画する機能を持つのは、CBTXTC (L.1350, 1407) またはGTXTCである。

但し、どちらも作画開始点の座標値は、ユーザ座標系で与えなければならない。そのため、グラフの中の適当な位置に文字を入れるには、グラフ座標値をユーザ座標値に変換しなければならない。その機能を持つのが、GRTRGU (L.1340) である。この逆の変換は、GRTRUGで行う。

さらに、変数TIM(J)に記憶されている数値を文字に変換しなければならない。この機能を持つのが、CBCRE(L.1320)またはCBCIEである。前者が実数を、後者が整数を文字に変換する。

また、作画文字の大きさを変えたい場合もある。そのような時に用いるのが、CONCHA(L.1406)である。

#### (4) 主要サブルーチンの詳細

##### (ア) GRGRID (文献1)

- 1 指定された軸の全ての目盛からリファレンス線を引く。
- 2 CALL GRGRID(K)  
このサブルーチン呼び出す前に、GRXAXSおよびGRYAXSが呼び出されていなければならない。
- 3 ☆K……どの軸からリファレンス線を引くか指定する。Kの値の意味は次の通り  
1 : X軸    2 : Y軸    3 : X軸およびY軸
- 4 L.1403で使用されている。K=2であり、Y軸の全ての目盛からリファレンス線が引かれている。

##### (イ) GRREF (文献1)

- 1 X軸またはY軸から指定された座標値にリファレンス線を引く。  
1回の呼出しで、リファレンス線が1本だけ引かれる。
- 2 CALL GRREF(K,VALG)  
このサブルーチン呼び出す前に、GRXAXSおよびGRYAXSが呼び出されていなければならない。
- 3 ☆K……GRGRIDの引数Kと意味は同じ。但し、K=3の値はとりえない。  
☆VALG……リファレンス線を引くグラフ座標値である。整数軸ならば整数値を指定する。
- 4 L.1405で使用されている。K=1, VALG=0.5 \* ALだからX軸の midpoint からリファレンス線が引かれている。

##### (ウ) CONLIN (文献1)

- 1 これ以後の作画に使う線の太さと線種を指定する。但し、文字やシンボルの線は変化しない。  
初めてこのサブルーチン呼び出す前は、標準値(実細線)が使われる。
- 2 CALL CONLIN(IWIDTH,IMODE,PITCH)
- 3 ☆IWIDTH…細線か太線かの指定。

0 : 細線    1 : 太線

☆IMODE…線種の指定。

0 : 実線    1 : 破線    2 : 点線    3 : 1点鎖線

破線の1線分の長さは1/2ピッチ、点線は1/5ピッチ

☆PITCH…1ピッチの長さの指定（単位はcm）。実線の場合は無意味。

- 4 L.1031で、実太線を指定し、軸、折れ線グラフ、枠までを書いた。
- L.1402で、実細線を指定し、Y軸のリファレンス線を引いた。
- L.1404で、ピッチ1cmの1点鎖線、細線を指定し、X軸のリファレンス線を引いた。

(エ) CBTXTC (文献1)

- 1 指定されたユーザ座標位置から文字列を横書きに作画する。
- 2 CALL CBTXTC (XU, YU, CHAR, NCHAR)
- 3 ☆XU, YU……文字列の作画開始点のX, Y ユーザ座標値である。文字列は、この点が左下端になるように作画される。  
☆CHAR, NCHAR……作画すべき文字列とその文字数である。GRXAXS (ref. 2. 1 (2) (ア)) のILAB とNCHARの関係と同じである。
- 4 L.1350では、それぞれの温度分布曲線の中心点に時間を書いている。その際、作画開始点のX, Y ユーザ座標値は、直前のGRTRGU (L.1340) によって求めている。また、時間TIM (J) の値は、L.1320のCBCREで文字に変換し、文字変数 TCHA (L.220 で型宣言) に記憶させている。その TCHA を間に挟んで、'TIME=' と 'MIN ' を連結して、合計11文字 (空白も含む) を作画している。  
L.1407では、(9, 1) cm の点から標題を書かせている。なお、直前のCONCHA (L.1406) で文字を大きくしている。

(オ) GTXTTC (文献1)

- 1 指定されたユーザ座標値から、文字列を横書きまたは縦書きに作画する。
- 2 CALL GTXTTC (XU, YU, CHAR, NCHAR, IANGLE, MCHAR)
- 3 ☆IANGLE, MCHAR以外はCBTXTCと同じ。  
☆IANGLE…横書きか縦書きかの指定。  
1 : 横書き    2 : 縦書き  
☆MCHAR…出力用である。整定数を指定する。作画した文字数が出力される。文字がカタカナでない場合は、NCHARと同じ値となる。

- 4 このプログラムでは使用していないが、L.1407のCBTXTCの代りに用いるとすれば、次のようになる。

```
CALL GTXTC(9.,1.,'TEMPERATURE PROFILE',19,1,MCHAR)
```

(カ) GRTRGU (文献1)

- 1 グラフ座標系での座標値をユーザ座標系での座標値に変換する。
- 2 CALL GRTRGU(XG,YG,XU,YU)  
このサブルーチンを呼び出す前に、GRXAXSおよびGRYAXSが呼び出されていなければならない。
- 3 ☆XG,YG……変換すべきX,Yグラフ座標値である。但し、軸の始点の座標を指定してはならない。  
☆XU,YU……出力用である。ユーザ座標値に変換されたX,Y値の出力である。
- 4 L.1340で、温度分布曲線のピークのX,Yグラフ座標値、CENTX,SUM(5)をユーザ座標値に変換して、それぞれXUS,YUSに記憶させている。そしてL.1350でこのユーザ座標値XUS,YUSを用いて文字列の作画開始点を指定している。

(キ) GRTRUG (文献1)

- 1 ユーザ座標値からグラフ座標値に変換する。  
GRTRGUと同様なので、詳細は割愛する。

(ク) CBCRE (文献1)

- 1 実数型データを文字型データに変換する。
- 2 CALL CBCRE(VARI,NDEC,CHAR,NCHAR,MODE)
- 3 ☆VARI……文字変換すべき実数データである。  
☆CHAR……出力用文字型変数である。変換された文字が出力される。  
☆NDEC,NCHAR,MODE……これらの組で文字の格納方法を指定する。  
△NDEC……変換された文字の小数点以下の桁数の指定。指定は、表5のように行う。  
△NCHAR……変換された文字を記憶する領域の文字数である。  
△MODE……変換された文字を記憶する時に、NCHARの領域に右から(下位から)詰めるか、左から(上位から)詰めるかを指定する。但し、左から詰める場合、先頭領域は符号のために占有され、正の場合は空白となる。  
正：右詰め、下位から      負、0：左詰め、上位から



表5 NDECの与え方

NDEC	変換された文字の構成
正	整数部+小数点+小数点以下 (NDEC) 桁
0	整数部+小数点まで
-1	整数部のみ
-2以下	整数部を下位から ( NDEC -1) 桁切り取る

☆これらの引数の関係の例を表6に示す。

表6 subCBCREの引数の関係

VARI	NCHAR	NDEC	MODE	CHAR
12.34	8	2	0	┌12.34┐
			1	┐┐┐12.34
		0	0	┌12.┐┐┐┐
			1	┐┐┐┐┐12.
	4	2	0	┌12.
			1	2.34
		-1	0	┌12┐
			1	┐┐12
		-2	0	┌1┐┐
			1	┐┐┐1

- 4 L.1320で実変数TIM(J)の整数部だけ (NDEC=-1) を文字変数 TCHA (L.220で文字型宣言) に3文字分領域を確保して、右詰め (MODE=1) にして記憶している。この場合、左詰めにすると、TIM(7)=240の時に、TCHA='┌24' となってしまふ。  
この TCHA を、L.1350のCBTXTCでグラフ上に書き込んでいる。
- 5 数値データの文字変換には、別法がある。ここの例の場合には次のようにすれば良い

(※1)。

```
WRITE(TCHA, '(A5,I3,A3)') 'TIME=',INT(TIM(J)+0.1),  
'MIN'
```

または、

```
WRITE(TCHA, ('TIME=',I3,'MIN')) INT(TIM(J)+0.1)
```

この場合、引用符の中の引用符は2つの連続した引用符(' ')となる(※2)。

なお、それに伴って、L.220のTCHAの大きさ、L.1350の文字列部も変更しなければならない。

(ケ) CBCIE (文献1)

- 1 整数型データを文字型に変換する。
- 2 CALL CBCIE(IVARI,CHAR,NCHAR,MODE)
- 3 CBCREと、ほぼ同じなので割愛する。
- 4 このプログラムでは用いていないが、L.1320のCBCREの代りに用いるとすれば、次のようになる(※1)。

```
ITIM=INT(TIM(J)+0.1)
```

```
CALL CBCIE(ITIM,TCHA,3,1)
```

(コ) CONCHA (文献1)

- 1 これ以後に作画する文字のサイズを指定する。初めてこのサブルーチン  
を呼び出す前は、標準サイズ0.28cmが用いられる。
- 2 CALL CONCHA(SIZE)
- 3 ☆ SIZE ……文字のサイズ(単位はcm)である。0.07cmの整数倍を指定するのが良  
いようである。
- 4 L.1406で、文字のサイズをそれまでの標準サイズ0.28cmから0.42cmに変更した。  
したがって、次のCBTXTC(L.1407)で書かれた標題の文字は大きくなっている。

---

※1 実数TIM(J)は、整数化される時には、小数点以下が切り捨てられるので、0.1を加えてから整数化する。

※2 (")ではない。

### 3. 2 文字に小文字も使い、Y軸の標題を横書きにする

プログラム (memPLSYMBOL) を図11に、出力結果を図12に示す。

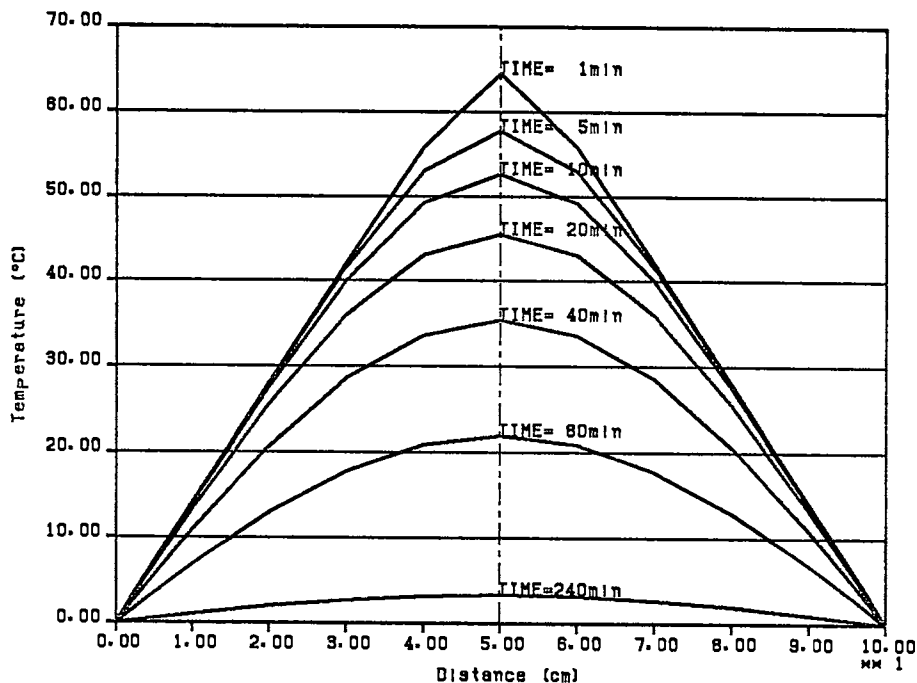
```
1 0 ~ 2 1 0 memPLDECORA (図9) と同じ
00211 DIMENSION LXCMT(13),LYCMT(16),LMIN(3)
00220 CHARACTER ENDCNT*20/'TEMP.PROF.SYMBOL'/,TCHA*3
4 1 0 ~ 3 3 0 memPLDECORA と同じ
00440 DATA LXCMT/68,150,177,178,135,165,137,146,64,77,137,164,93/
00450 DATA LYCMT/99,146,164,167,146,169,135,178,179,169,146,64,77,255,
00460 + 67,93/
00470 DATA LMIN/164,150,165/
1 0 1 0 ~ 1 0 3 1 memPLDECORA と同じ
01040 CALL GRXAXS(4.,3.,19.,XDT,IXF,-2,'DISTANCE (CM)',0)
01050 CALL GRYAXS(4.,3.,15.,YDT,IYF,2,'TEMPERATURE (C)',0)
¥ 01060 CALL SYMBOL(12.,1.6,0.28,68,0.,0)
01070 DO 20 L=2,13
¥ 01080 20 CALL SYMBOL(9999.,9999.,0.28,LXCMT(L),0.,0)
¥ 01090 CALL SYMBOL(1.7,8.,0.28,99,90.,0)
01091 DO 30 L=2,16
¥ 01092 IF(LYCMT(L).EQ.167) CALL SYMBOL(1.8,9999.,0.35,LYCMT(L),90.,0)
¥ 01093 IF(LYCMT(L).NE.167) CALL SYMBOL(1.7,9999.,0.28,LYCMT(L),90.,0)
01094 30 CONTINUE
1 1 0 0 ~ 1 3 4 0 memPLDECORA と同じ
01350 CALL CBTXTC(XUS,YUS,'TIME='//TCHA,8)
01360 DO 40 L=1,3
¥ 01370 40 CALL SYMBOL(9999.,9999.,0.28,LMIN(L),0.,0)
1 4 0 0 ~ 1 6 0 0 memPLDECORA と同じ
```

図11 memPLSYMBOL

プログラムは図9のmemPLDECORAを基礎としており、それとの変更 部分だけを抜粋して書いた。小文字を使うことも、Y軸の標題を横書きで縦方向に作画することも、subSYMBOL の機能である。以下に、subSYMBOL について詳述する。

#### (ア) SYMBOL (文献2)

- 1 指定されたユーザ座標位置から、指定された角度、大きさを文字列を横書きに作画する。作画できる文字には、大文字の英数字のほか、英小文字、いくつかのギリシャ文字、特殊記号やカタカナなどが含まれる。但し、キーボード上にない文字・記号(例えば英小文字)は1文字ずつしか作画されない。
- 2 CALL SYMBOL (XU, YU, HEIGHT, TEXT, ANGLE, NCHAR)
- 3 ☆XU, YU……作画開始点のX, Y ユーザ座標値である。但し、XU, YU ≥ 9999. と指定



TEMPERATURE PROFILE

図12 memPLSYMBOL により作画された絵 (縮尺1/2)

すると、1つ前に呼ばれたSYMBOLまたはCBTXTC, GTXTC等で作画された文字列に後続して作画される。また、XU, YUのどちらかのみ、この指定をすると、ユーザ座標値の指定された側はその値に従い、9999.以上を指定した側は、前の文字列に後続した位置になる。

☆HEIGHT……このサブルーチンで 作画する文字のサイズである。CONCHAのSIZEと同じ意味である。CONCHAのサイズ指定とは、それぞれ全く独立に機能する。

☆TEXT, NCHAR……対で、作画する文字またはその整数コードを与える。

△ TEXT ……作画すべき文字列 (キーボード上の文字・記号) であるか、または作画すべき1文字または1記号の整数コード (※3) である。

△NCHAR…… TEXT に文字列が記憶されている場合は、その文字数である。TEXT に、

※3 文献2のP.10, 11 を参照せよ。

整数コードが記憶されている場合は、NCHAR=0とする。

☆ANGLE……文字列または文字が横方向（X軸方向）と、なす角度である（反時計廻り）。

- 4 X, Y軸の標題と時間の単位（min）を小文字とした。したがって、GRXAXS（L.1040）、GRYAXS（L.1050）の引数NCHARを0とし、これらのサブルーチンでは標題を作画しない。L.1060～1080で、X軸の標題を書いている。L.1060のsubSYMBOLで、まず書き出しの位置を決め（※4）、あとは、XU, YUにそれぞれ9999.を入れて後続して作画させている（L.1070, 1080）。作画する小文字の整数コードは、あらかじめ配列LXCMTに、DATA文で与えている（L.440）。なお、空白も1つの整数コードとして入力されている。また、L.1060は、次のようにしても良い。

```
CALL SYMBOL(12., 1.6, 0.28, 'D', 0., 1)
```

Y軸の標題も、X軸の標題と同様に書かせているが（L.1090～1094）、ANGLEは90.となっている。また、標題の文字中には“p”が含まれているため、この文字だけ作画位置をずらすと同時に、少しサイズを大きくしてやらなければならない（※5）。そのためにL.1092が用意されている。なお、整数コード167は、小文字のpにあたる。ここでは、1mm下げて、5/4倍の大きさにしているが、まだぴったりとはしていない。しかし、この程度が限度のようである。

L.1360、1370では、時間の単位“min”を書いている。この場合も、データは予め与えてある（L.470）。単位は、直前に書いた数値（L.1350）に続けて書けば良いので、ここでは、XU, YUともに9999.が与えてある。なお、subSYMBOLを用いれば、グラフ全体の標題を書かせている2行（L.1406, 1407）は、次の1行で置き換えることができる。

```
CALL SYMBOL(9., 1, 0.42, 'TEMPERATURE PROFILE', 0., 19)
```

また、Y軸の標題を大文字のまま単に横並びにするだけならば、次のようにすれば良い。

```
CALL SYMBOL(2., 8.5, 0.28, 'TEMPERATURE (C)', 90., 15)
```

memPLDECORAに、上の1行だけを取り入れたものを改めてmemPLDECORBにSAVEし、以後は、このPLDECORBを基礎として必要な部分のみの変更を行うこととする。以後の対比が容易なように、ここでPLDECORBの行番号を100番おきにつけ直して図13に再掲する。



※4 整数コード68は、大文字のDである。

※5 特別の指定をしないと、小文字の“p”は“m”や“e”と同じ位置から同じサイズで書いてしまう。

```

00100    PARAMETER (AL=100., TO=70., AK=12., NTIME=7)
00200    PARAMETER (XL=AL/10.+1.)
00300    COMMON/CNM/ SUM(0:10)
00400    DIMENSION TIM(NTIME),XDT(3),IXF(3),YDT(2),IYF(3),IXF2(2),IYF2(2)
00500    CHARACTER ENDGMT*20/'TEMP.PROF.DECO'/,TCHA*3
00600    DATA TIM/1.,.5.,10.,20.,40.,80.,240./, EPS/1.E-2/
00700    DATA XDT/XL,0.,10./,IXF/-2,2,2/, IYF/2,2,2/,IXF2/-2,2/,IYF2/11,2/,
00800    +    YDT/0.,70./
00900    CALL CONDEV('6202')
01000    CALL GRINIT
01100    CALL PLOTS(0.,0.,16)
01200    CALL CONLIN(1,0,0.)
01300    CALL GRXAXS(4.,3.,19.,XDT,1,XF,-2,'DISTANCE (CM)',13)
01400    CALL GRYAXS(4.,3.,15.,YDT,IYF,2,'TEMPERATURE (C)',0)
01500    CALL SYMBOL(2.,8.5,0.28,' TEMPERATURE (C)',90.,15)
01600    DO 10 J=1,NTIME
01700    CALL CALNUM(TIM(J),AL,TO,AK,EPS)
01800    CALL GRPGNI(XDT,SUM,IXF2,IYF2,0)
01900    CALL CBCRE(TIM(J),-1,TCHA,3,1)
02000    CENTX=0.5*AL
02100    CALL GRTRGU(CENTX,SUM(5),XUS,YUS)
02200    CALL CBXTTC(XUS,YUS,'TIME='//TCHA//'MIN',11)
02300    10 CONTINUE
02400    CALL GRFRAM
02500    CALL CONLIN(0,0,0.)
02600    CALL GRGRID(2)
02700    CALL CONLIN(0,3,1.)
02800    CALL GRREF(1,CENTX)
02900    CALL CONCHA(0.42)
03000    CALL CBXTTC(9.,1.,'TEMPERATURE PROFILE',19)
03100    CALL PLOTE(ENDGMT)
03200    STOP
03300    END

```

図13 memPLDECORB

3. 3 測定値も同時にプロットし、それらの点を近似曲線で結ぶ。

計算点も滑らかな曲線で結ぶ

プログラム (memPLOBSERV) を図14に、出力結果を図15に示す。  
このプログラムは、図13のmemPLDECORBを基礎としており、それとの変更部分だけを抜粋して書いた。なお、DATAとして測定値(4つの時間での各点の測定温度(TEMPOB))がふえ、DATA文が多くなったので、全てのDATA文は、サブプログラムBLOCK DATAにまとめた。また、それに伴いPARAMETER文で与えていたデータも全てDATA文で与えることとした。

```

0210    COMMON/BLDT/ TIM (4) ,XDT (3) ,IXF (3) ,YDT (2) ,IYF (3) ,IXF2 (2) ,IYF2 (2) ,
0220    +          TEMPOB (4,0:10) ,XDTARR (0:10) ,IXYF (2) ,AL ,TO ,AK ,EPS
0230    COMMON/CHA/ ENDCMT
0300    COMMON/CNM/ SUM (0:10)
0410    DIMENSION XFIT (101) ,YFIT (101) ,A (11) ,B (11) ,C (11) ,D (11) ,
0420    +          YDFOB (0:10) ,CAP (9) ,VW (150)
0500    CHARACTER ENDCMT*20 ,TCHA*3
9000~1500 memPLDECORB (図13) と同じ
1510    ISYM=1
1600    DO 10 J=1 ,4
1700    CALL CALNUM (TIM (J) ,AL ,TO ,AK ,EPS)
¥ 1710    CALL G1FIT1 (XDTARR ,SUM ,11 ,101 ,XFIT ,YFIT ,A ,B ,C ,D)
1800    CALL GRPGNI (XFIT ,YFIT ,IXYF ,IXYF ,0)

1900~2200 memPLDECORBと同じ
2210    DO 30 I=0 ,10
2220    30 YDFOB (I) =TEMPOB (J ,I)
¥ 2230    CALL GRPNT1 (XDT ,YDFOB ,IXF2 ,IYF2 ,ISYM)
2240    ISYM=ISYM+1
2250    CALL CONLIN (1 ,1 ,0.3)
¥ 2260    CALL GRAPP1 (XDTARR ,YDFOB ,IYF2 ,IYF2 ,4 ,101 ,CAP ,VW)
2270    CALL CONLIN (1 ,0 ,0.)

2300~3300 memPLDECORBと同じ
3310 C*****
3320    BLOCK DATA
3330    COMMON/BLDT/ TIM (4) ,XDT (3) ,IXF (3) ,YDT (2) ,IYF (3) ,IXF2 (2) ,IYF2 (2) ,
3340    +          TEMPOB (4,0:10) ,XDTARR (0:10) ,IXYF (2) ,AL ,TO ,AK ,EPS
3350    COMMON/CHA/ ENDCMT
3360    CHARACTER ENDCMT*20/'TEMP.PROF.OBSERV'/
3370    DATA AL ,TO ,AK ,EPS /100. , 70. , 12. , 1.E-2/
3380    DATA TIM/5. ,20. ,40. ,80./
3390    DATA XDT/11. ,0. ,10. / ,IXF/-2,2,2/ ,IYF/2,2,2/ ,IXF2/-2,2/ ,IYF2/11,2/ ,
3400    +    YDT/0. ,70. / ,IXYF/101,2/
3410    DATA (TEMPOB (1,1) ,I=0,10)
3420    + /2.4 , 13.9 , 27.0 , 39.0 , 51.3 , 52.6 , 48.0 , 43.4 , 25.5 , 13.4 , 1.7/
3430    DATA (TEMPOB (2,1) ,I=0,10)
3440    + /2.9 , 15.7 , 27.2 , 33.4 , 41.2 , 43.0 , 39.4 , 36.8 , 23.3 , 15.0 , 0.9/
3450    DATA (TEMPOB (3,1) ,I=0,10)
3460    + /2.8 , 9.6 , 21.8 , 29.8 , 31.2 , 34.2 , 31.4 , 28.3 , 21.6 , 8.9 , 3.2/
3470    DATA (TEMPOB (4,1) ,I=0,10)
3480    + /3.6 , 9.4 , 13.0 , 17.8 , 19.1 , 20.6 , 21.9 , 17.6 , 11.3 , 5.2 , 1.5/
3490    DATA XDTARR/0. ,10. ,20. ,30. ,40. ,50. ,60. ,70. ,80. ,90. ,100./
3500    END

```

図14 memPLOBSERV

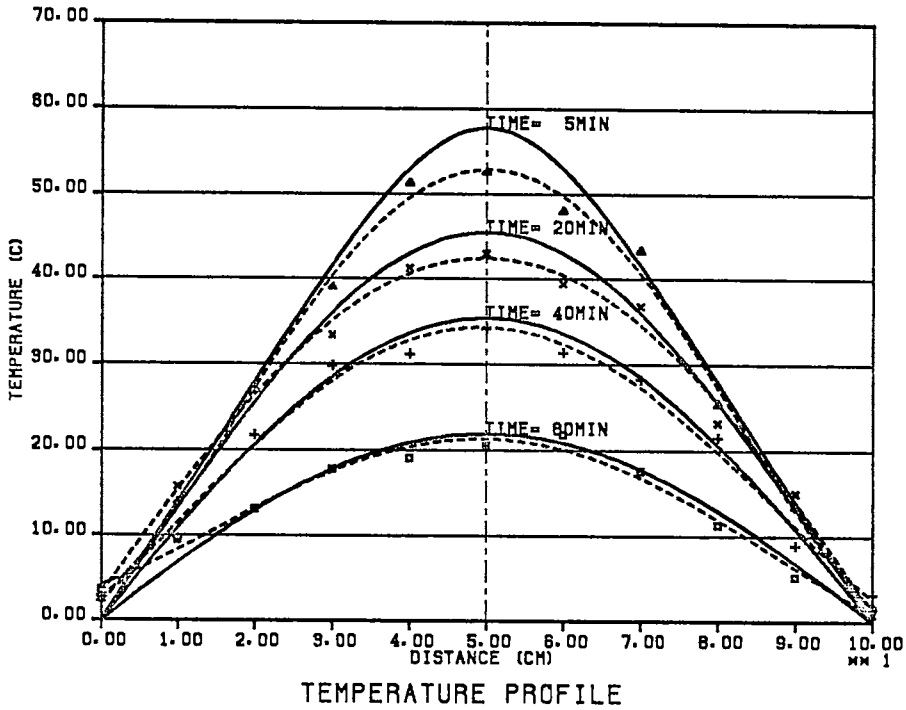


図15 memPLOBSERV により作画された絵 (縮尺1/2)

(1) 測定値のプロット

GRPNT1 (L.2230) の機能である。点グラフを作る機能を持っている。プロットの数だけX, Yグラフ座標値を与え、作画シンボルを指定すれば良い。

(2) 近似曲線を描く

GRAPP1 (L.2260) の機能である。与えられたデータ点 (グラフ座標系) に対して指定された次数の近似曲線を描く。

(3) 計算点を滑らかな曲線で結ぶ。

GIFIT1 (L.1710) とGRPGN1 (L.1800) またはGLINE2 (※6) を組み合わせればできる (※7)。GIFIT1は、与えられたデータ点 (グラフ座標値でもユーザ座標値でも良



い) をスプライン補間する機能を持つ。但し、計算を行うだけでプロットはしない。そこで、出力された計算値をGRPGN1 (グラフ座標値) または、GLINE2 (※6) (ユーザ座標値) を用いて、直線でつなぎ合わせる。その際、補間の計算点を細かくとれば滑らかな曲線となる。

#### (4) 主要サブルーチンの詳細

##### (ア) GRPNT1 (文献1)

- 1 X, Y グラフ座標値によって与えられたデータ点の全てに指定されたシンボルを作画する。点グラフとなる。
- 2 CALL GRPNT1 (XDATAG, YDATAG, IXFORM, IYFORM, ISYMBL)  
このサブルーチンを呼び出す前に、GRXAXS および GRYAXS が呼び出されていないと  
ならない。
- 3 引数は、全て GRPGN1 (ref. 2. 1 (ウ)) と同じ。
- 4 L. 2230 で、温度測定値をプロットしている。XDATAG には、X 軸を書く時に使用した  
数値増分形式の XDT をそのまま使用した。

YDATAG は、測定値を入れるところであるが、測定値は 2 次元配列 (TEMPOB) に記憶されているため、1 次元配列 YDTOB に変換して (L. 2210, 2220) 使用している。

シンボルの指定は、測定時間によって異なるように、ISYM で与えている (L. 1510, 2230, 2240)。

このサブルーチンは、データ点をプロットするだけであり、図 15 の破線は、別のサブルーチン (L. 2260 の GRAPP1) で描いている。

なお、測定値 TEMPOB は、BLOCK DATA 中の DATA 文で与えた。(L. 3410 ~ 3480)

##### (イ) GRAPP1 (文献1)

- 1 X, Y グラフ座標値によって与えられたデータ点に対して指定された次数の近似曲線を計算し、指定された数の点列に直して、折れ線グラフを作る。同時に、近似曲線の各係数の値も出力する。但し、入力データ点は作画されない。

なお、近似曲線の計算においては、X を独立変数、Y を従属変数とみなし、最小二乗法に

---

※6 GLINE1 でも良い。

※7 あるいは、GIFIT2 (円弧補間) と GCIRC3 を組み合わせても良い。

よって、次の形の式の係数Cを決定する。

$$Y=C_0+C_1X+C_2X^2+\dots+C_mX^m$$

ここでmは指定された次数である。

2 CALL GRAPP1 (XDATAG, YDATAG, IXYFRM, IXYFRM, MDIMEN, NPOINT, COEFF, VW)

このサブルーチンを呼び出す前に、GRXAXSおよびGRYAXSが呼び出されていないなければならない。

3 ☆XDATAG, YDATAG……近似曲線を作るべき点の、X, Yグラフ座標値である。実数配列である。この場合は必ず実数でなければならず、また、必ず配列形式で与えなければならない。

☆IXYFRM……既に何回も出てきたIXFORM, IYFORMと同じである。但し、この場合、XDATAG, YDATAGは、ともに配列形式で実数値しか取り得ないのでIXFORMとIYFORMは等しくなる。それをIXYFRMと書いた。整数配列IXYFRMの与え方を表7に示す。

表7 整数配列 I X Y F R M の与え方

I X Y F R M	値
(1)	XDATAGまたはYDATAGのデータ数
(2)	2 (実数値)

☆MDIMEN……求める近似多項式の次数の上限を指定する。当然  $1 \leq MDIMEN \leq$  データ数-1 である。

☆NPOINT……求めた近似多項式に基いて近似曲線を作画する時の点列数である。

☆COEFF……出力用である。実数配列である。求められた近似多項式の係数で、

$C_0, C_1, C_2, \dots, C_m$  の順に記憶される。したがって、配列の大きさは ( MDIMEN+1 ) 以上である。

☆VW……作業域である。実数配列である。大きさは、データ数×13である。

4 L.2230でプロットした測定点を入力データ点として、L.2260で近似曲線を作画している。その際、直前のL.2250で線種を太破線に変更し、近似曲線を作画し終わったら、また前の線種太実線に戻している (L.2270)。

また、GRAPP1 (L.2260) のXDATAGは、配列形式で与えなければならないので、これまでXDATAGとして用いてきた数値増分形式のXDTは使用できない。そこで、新しく配列形式のデータXDTARRを導入した。XDTARRは、BLOCK DATAのDATA文

で与えている。(L.3490)

IXYFRMとしては、GRPNT1 (L.2230) で用いた IYF2 をそのまま用いた。

ここでは、近似曲線の次数は4次とした。また、滑らかな曲線とするため、作画点列を101点とした。なお、近似多項式の係数は、CAPに記憶されるが(配列宣言はL.420)、このプログラムでは、その出力結果は使用していない。次の第4節でこれを書き出してみる。

なお、VWはL.420で配列宣言している。

(ウ) GIFIT1 (文献1)

- 1 与えられたデータ点列(グラフ座標系でもユーザ座標系でも良い)の全てを通る滑らかな曲線を、スプライン補間法で求め、その曲線を指定された数の補間点列として出力する。但し、独立変数 $X$ に対して従属変数 $Y$ が1価関数となるような点列でなければならない。また、このサブルーチンは計算結果を出力するだけで作画はしない。
- 2 CALL GIFIT1(XDATA, YDATA, NDATA, NPOINT, XDOUT, YDOUT, A, B, C, D)
- 3 ☆XDATA, YDATA……実数配列である。補間する入力点列の $X$ ,  $Y$ 座標値である。必ず配列形式で与える。点列は、 $X$ 座標値の小さい順に並んでいる必要はない(※8)。  
XDATAには等しいものがあるてはならない。  
☆NDATA……入力点列の数である。  
☆NPOINT……出力点列の数である。したがって、(NPOINT-NDATA)の数の点が補間される。補間点数の割り付けは、入力点列の間隔には無関係に均等に割り付けられ、余りは最初の間隔から順に割り付けられる。  
☆XDOUT, YDOUT……出力用である。実数配列である。補間後の点列の $X$ ,  $Y$ 座標値が出力される。この中には、入力点列も含まれている。配列の大きさは、NPOINT以上である。  
☆A, B, C, D……作業領域である。実数配列である。大きさは、NDATA以上である。
- 4 L.1710でスプライン補間を行っている。入力データは、直前のL.1700で計算した $X$ の各点の温度である。XDATAは、配列形式でなければならないので、GRAPP1の場合と同様に、XDARRを使用している。  
出力点を101個とし、各計算点間を10等分して、それぞれ9点の補間をしている。

---

※8 この場合、警告のメッセージが表示されるが、計算機が勝手に並べ換えてくれる。

出力点列は、XFIT, YFIT に出力させている (配列宣言はL.410)。  
出力させた点列データ XFIT, YFIT を、直後のL.1800のGRP GN1で折れ線グラフにしている。

#### (エ) GLINE2 (文献1)

- 1 X, Y ユーザ座標値で与えられたデータ点列を折れ線でつなぎ合わせる。

作画線種が実線でない場合、データ点位置とは無関係に、指定されたピッチを優先して、連続的に折れ線を作成する。したがって、データ点上が空白になることもありうる。これを避けるには、GLINE1 を用いれば良い。

GLINE1 は、各2点を単位として作画する。つまり、一方の点を必ず指定線種の始点として作画を始める。また、他方の点が空白になるようならピッチを無視して、最後の線分をその点まで延長する。したがって、データ点上には、必ず折れた実線が書かれる。

これはまた、データ点列間隔が指定線種のピッチより短い場合には、必ずGLINE2 を使わなければならないことを意味している。

- 2 CALL GLINE2 (XDATAU, YDATAU, NDATA, SINF, ISINF)

- 3 ☆XDATAU, YDATAU……実数配列である。データ点列のX, Y ユーザ座標値である。  
必ず配列形式で与える。

☆NDATA……データ点列の数である。

☆SINF……通常は、ユーザがこの変数に値をセットする必要はない。そのまま用いる。

☆ISINF……通常は0とする。

- 4 このプログラムでは使用していないが、次の第4節のmemPLOUTGRPの中で、GLINE1の方がグラフ座標系外に破線を引くのに用いられている。

### 3. 4 グラフの外側にマークを作画し、線を引き、文字を書く

プログラム (memPLOUTGRP) を図16に、出力結果を図17に示す。

プログラムは、図14のmemPLOBSERVを基礎としており、それとの変更部分だけを抜粋して書いた。グラフの上にかかれた表は、近似曲線の各係数である。プログラムの詳細な説明は割愛するので、各自学習されたい。新しく使ったサブルーチンは、GMARK1 (L.2265, 2266) と GLINE1 (L.2269)、GPOLY1 (L.2420) の3つである。これらは全てユーザ座標系で使われる。

GMARK1 (文献1) は、シンボルの作画機能をもつ。ここでは、上の表の MARK 欄のシンボルを作画している。GLINE1 (文献1) は、データ点列を折れ線で結ぶ機能を持つ。ここでは、同じ

```

 2 1 0 ~ 2 2 0   m e m P L O B S E R V ( 図 1 4 ) と 同 じ
0221   +           , XLINE(2), YPS, ISYM
0230   COMMON/CHA/ ENDCMT
0300   COMMON/CNM/ SUM(0:10)
0410   DIMENSION XFIT(101), YFIT(101), A(11), B(11), C(11), D(11),
0420   +           YDFOB(0:10), CAP(9), VW(150), YLINE(2)
0500   CHARACTER ENDCMT*20, TCHA*3, ONELIN*75
 9 0 0 ~ 1 5 0 0   m e m P L O B S E R V と 同 じ
1520   CALL CBTXTC(4., 23.,
1530   + 'T = C(0) + C(1)*X + C(2)*(X**2) + C(3)*(X**3) + C(4)*(X**4)', 59)
1540   WRITE(ONELIN, '(' TIME', 3X, ' MARK', 5(8X, ' C(', 11, ') '))')
1550   +           (1, 1=0, 4)
1560   CALL CBTXTC(2., YPS, ONELIN, 72)
1570   YPS=YPS-0.5
1 6 0 0 ~ 2 2 6 0   m e m P L O B S E R V と 同 じ
¥ 2261   WRITE(ONELIN, '(12, ' MIN', 10X, 5E12.3)')
¥ 2262   +           INT(TIM(J)+0.1), (CAP(1), 1=1, 5)
¥ 2263   CALL CBTXTC(2., YPS, ONELIN, 75)
2264   YPSM=YPS+0.14
¥ 2265   CALL GMARK1(1, 4.5, YPSM, ISYM, 0.2)
¥ 2266   CALL GMARK1(1, 5.5, YPSM, ISYM, 0.2)
2267   YLINE(1)=YPSM
2268   YLINE(2)=YPSM
¥ 2269   CALL GLINE1(XLINE, YLINE, 2)
2270   CALL CONLIN(1, 0, 0.)
2280   ISYM=ISYM+1
2290   YPS=YPS-0.5
2 3 0 0 ~ 2 4 0 0   m e m P L O B S E R V と 同 じ
2410   YPS=YPS+0.3
¥ 2420   CALL GPOLY1(1.8, YPS, 21.3, 2.7, 0.)
2 5 0 0 ~ 3 3 5 0   m e m P L O B S E R V と 同 じ
3341   +           , XLINE(2), YPS, ISYM
3350   COMMON/CHA/ ENDCMT
3360   CHARACTER ENDCMT*20/'TEMP.PROF.OUTGRP'/
3 3 7 0 ~ 3 4 9 0   m e m P L O B S E R V
3491   DATA XLINE/4., 6./, YPS/22./, ISYM/1/
3500   END

```

図 1 6 memPLOUTGRP

く MARK 欄のシンボルを通る直線を引いている。

GPOLY1 (文献 1) は、長方形の作画機能を持つ (※ 9)。ここでは、上の表の枠を書いている。

また、データの書き出しは WRITE 文で、一旦、文字変数 ONELIN (L. 500 で型宣言) に書き (L. 2261, 2262)、それを CBTXTC で絵の上書き出す (L. 2263) という方法をとっている。なお、WRITE 文の FORMAT は、直接引用符で括弧で与えているので、その内側で使う引用符は全て二つの連続した引用符 (``) としている。

$$T = C(0) + C(1) \times X + C(2) \times (X \times X) + C(3) \times (X \times X \times X) + C(4) \times (X \times X \times X \times X)$$

TIME	MARK	C(0)	C(1)	C(2)	C(3)	C(4)
5MIN	▲	0.267E+01	0.697E+00	0.450E-01	-0.103E-02	0.513E-05
20MIN	◆	0.312E+01	0.124E+01	0.120E-02	-0.271E-03	0.133E-05
40MIN	+	0.221E+01	0.880E+00	0.832E-02	-0.352E-03	0.181E-05
80MIN	□	0.414E+01	0.329E+00	0.115E-01	-0.295E-03	0.144E-05

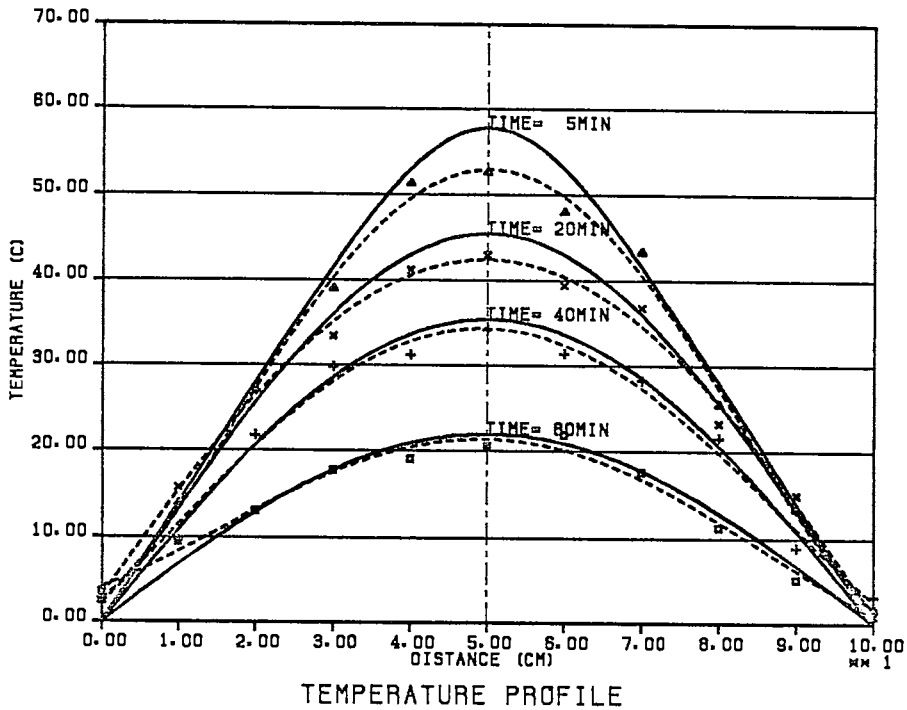


図17 memPLOTGRP により作画された絵 (縮尺1/2)

### 3. 5 複数の座標軸を持つグラフを書く 矢印を書く

実際のグラフでは、1つのグラフに圧力と温度を同時にプロットしたり、X軸のある部分だけを拡大して2重にプロットしたりすることがよくある。このように、複数の座標軸を持つグラフの作画例を次に示す。

ここでは、棒の温度勾配、即ち  $\partial T / \partial X$  も同時にプロットすることにした。即ち、Y軸が2つあ

※9 GRACE IV は、この他にユーザ座標系では円や円弧やまた種々の多角形を作画する機能を持っている。GRTRGU と併用すれば、適当なグラフ座標位置に、それらを作画することができる。

ることになる。

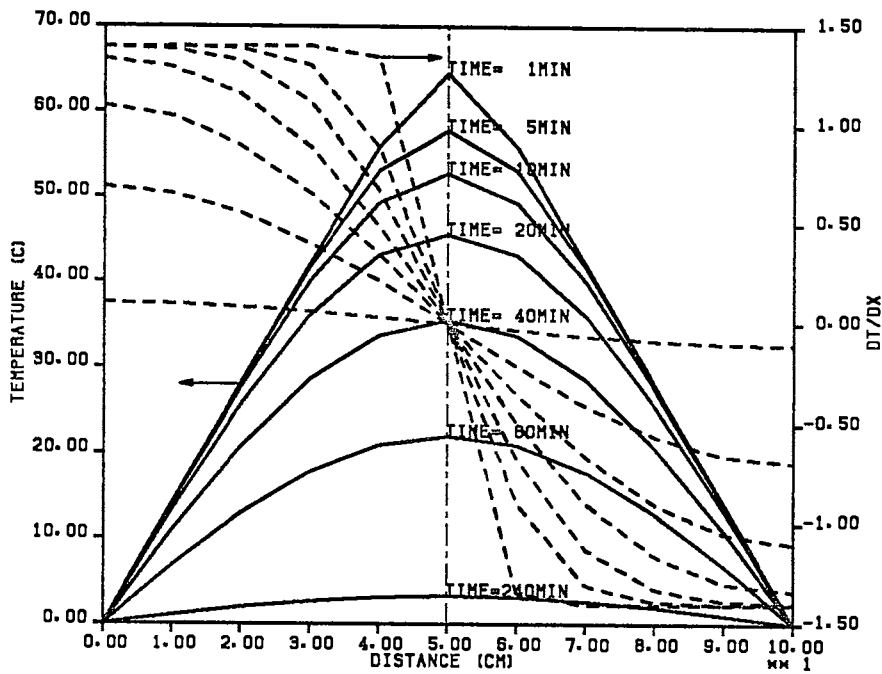
そのプログラム (memPL2YAXES) を図18に、出力結果を図19に示す。

```
1 0 0 ~ 2 0 0 memPLDECORB (図13) と同じ
0300 COMMON/CNM2/ SUM(0:10),SUM2(0:10)
0400 DIMENSION TIM(NTIME),XDT(3),IXF(3),YDT(2),IYF(3),IXF2(2),IYF2(2),
0410 + IYF3(3),SUM21(0:10)
0500 CHARACTER ENDGMT*20/'TEMP.PROF.2YAXES'/,TCHA*3
6 0 0 ~ 7 0 0 memPLDECORB と同じ
0800 + YDT/0.,70./,IYF3/11,2,2/
9 0 0 ~ 1 5 0 0 memPLDECORB と同じ
1510 IREP=0
1600 DO 10 J=1,NTIME
1700 CALL CLDTEX(TIM(J),AL,T0,AK,EPS)
¥ 1710 CALL GRYAXS(4.,3.,15.,YDT,IYF,0.,'.',0)
1 8 0 0 ~ 2 2 0 0 memPLDECORB と同じ
2210 IF(IREP.EQ.0) THEN
2220 X2=0.2*AL
2230 CALL GRTRGU(X2,SUM(2),XUS,YUS)
2240 DO 11 I=0,10
2250 11 SUM21(I)=SUM2(I)
2260 CALL GRYAXS(21.,3.,15.,SUM21,IYF3,-2,'DT/DX',0)
2270 CALL SYMBOL(23.1,10.,0.28,'DT/DX',90.,5)
2280 CALL CONLIN(0,0,0.)
* 2281 CALL ARROW(XUS,YUS,180.,1.5,0.3)
2282 X4=0.4*AL
2283 CALL GRTRGU(X4,SUM2(4),XUS,YUS)
* 2284 CALL ARROW(XUS,YUS,0.,1.5,0.3)
2285 IREP=1
2286 ELSE
¥ 2287 CALL GRYAXS(21.,3.,15.,SUM21,IYF3,0,'DT/DX',0)
2288 END IF
2289 CALL CONLIN(1,1,0.5)
2290 CALL GRPGN1(XDT,SUM2,IXF2,IYF2,0)
2291 CALL CONLIN(1,0,0.5)
2 3 0 0 ~ 3 3 0 0 memPLDECORB と同じ
但し、2 5 0 0、2 6 0 0 は削除する
```

図18 memPL2YAXES

プログラムは、図13のmemPLDECORBを基礎としており、それとの変更部分だけを抜粋して書いた。subCALNUMの代りに、 $\partial T / \partial X$ も同時に計算するsubCLDTEXを新たに作り、それだけからなるmemSUBDTEXを作った。実行は、そのSUBDTEXと結合して行う。なお、温度勾配は、温度と同じ点で計算され、配列SUM2に記憶させて戻される。memSUBDTEXは、memSUBCALNMと、ほぼ同じなので詳細は割愛する。

複数の座標軸を持つ場合には、プロットに矢印をつけることが多いので、ここで矢印も書くことにした。



TEMPERATURE PROFILE

図19 memPL2YAXES により作画された絵 (縮尺1/2)

なお、Y軸を複数にしたことに伴い、Y軸のリファレンス線は削除した (memPLDECORBの L.2500, 2600)。

(1) 複数の座標軸を持つグラフを書く

このためには、新しいサブルーチンは必要としない。ただ、GRYAXSの使い方の工夫をするだけである。即ち、L.1800で温度をプロットする前に、L.1710でGRYAXSを呼び、温度用のY軸を決めている。しかし、ここではGRYAXSの引数Kは0となっており、何ら作画はしていない。Y軸を設定しているのみである。なお、このY軸の作画は既にL.1400で行っている。このY軸設定は、次にGRAXSを呼び出すまで有効である。次にL.2290で温度勾配をプロットするが、その前には、温度勾配用のY軸を設定しなければならない。それをやっているのが、L.2210~2288である (但し、L.2220, 2280~2284は別)。ここでは、予め目盛が決められない場合を想定し、配列形式で目盛に関するデータを与えている。したがって、温度用Y軸のように予めY軸を



作画しておくことはできない。そこで、最初に温度勾配用Y軸を呼んだ時だけ、目盛、目盛値、標題を作画し (L.2260, 2270)、2回目からはY軸の設定だけで何も作画しないようにしている (L.2287)。なお、L.2240, 2250 は、温度勾配の最大、最小値が存在する最初のプロットデータを以後のY軸設定用に別に確保するためのものである。

## (2) 矢印を書く

subARROW の機能である。ユーザ座標系で用いられる。以下、詳細を説明する。

### (ア) ARROW (文献4)

- 1 ユーザ座標系で指定した点を、尾または矢先として指定した長さ角度で矢印を書く。矢の部分の大きさも指定する。
- 2 CALL ARROW (XU, YU, ANGLE, AXLNG, ARLNG)
- 3 ☆XU, YU……矢印の矢先または尾のX, Yユーザ座標値である。AXLNGが正の場合は尾となり、負の場合は矢先となる。  
☆ANGLE……矢印が横方向となす角度を反時計廻りで指定する (単位は度)。  
0 ~ 360 までの値をとりうる。  
☆AXLNG……矢印の軸の部分の長さを指定する (単位は cm)。正ならば矢印は (XU, YU) が尾となり、負ならば (XU, YU) が矢先となる。  
☆ARLNG……矢印の矢の部分の長さを指定する (単位は cm)。
- 4 L.2281 で、温度分布曲線に矢印をつけている。ユーザ座標値は、温度分布用Y軸が設定されていた時に、グラフ座標値から変換した値 (L.2230) を用いている。矢印の向きは左向きなので、ANGLE=180. とし、指定座標位置が尾なので、AXLNGは正とした。AXLNGを種々に変えたい場合には、ARLNGがAXLNGに比例して変化するようにしておくが良い。  
同様に、L.2284 で温度勾配曲線から矢印を引いている。向きが右向きのため、ANGLE=0 とした以外は、同様の操作をしている。

## 3. 6 1枚の絵に複数のグラフを書く 絵の各部分にページ指定をする

プログラム (memPL4GRAPH) を図20に、出力結果を図21に示す。  
プログラムは、図13のmemPLDECORBを基礎としており、それとの変更部分だけを抜粋して書いた。このプログラムでは、PLDECORB で書いた7つの時間のうち4つの時間の温度分布を別々

```

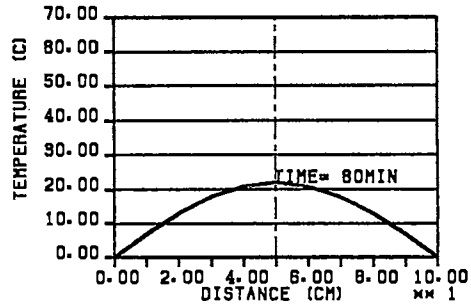
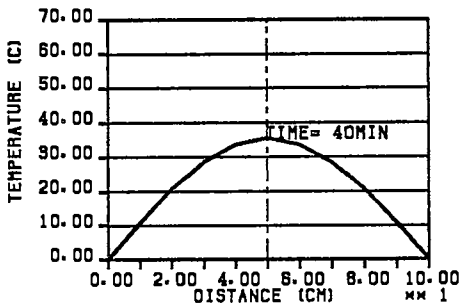
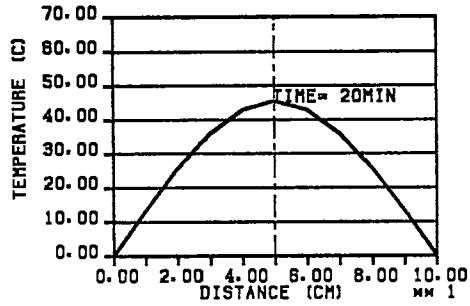
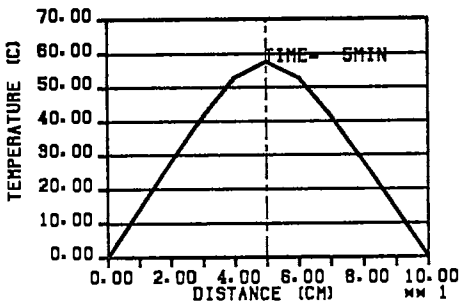
0010      CHARACTER ENDCMT*20/'TEMP.PROF.4GRAPH'/
   20 ~ 40      memPLDECORB (図13) のL. 900~1100と同じ
0050      NPAGE=30
¥ 0060      CALL PLGRPH( 5., 4.,15.,NPAGE)
¥ 0070      CALL PLGRPH(20.,16.,15.,NPAGE)
¥ 0080      CALL PLGRPH(40., 4., 4.,NPAGE)
¥ 0090      CALL PLGRPH(80.,16., 4.,NPAGE)
¥ 0091      CALL PAGE(NPAGE, 'TITLE')
¥ 0092      CALL PLOT(0.,0.,-3)
   93 ~ 97      memPLDECORBのL. 2900~3300と同じ
0098 C *****
0099      SUBROUTINE PLGRPH(TM,XS,YS,NPAGE)
1000 ~ 400      memPLDECORBと同じ
0500      CHARACTER TCHA*3,PGCMT*8
0600      DATA EPS/1.E-2/
   700 ~ 800      memPLDECORBと同じ
¥ 1110      WRITE(PGCMT, '(''XYAXES'',12.2)') INT(TM+0.1)
¥ 1120      CALL PAGE(NPAGE,PGCMT)
¥ 1130      CALL PLOT(0.,0.,-3)
   1140      NPAGE=NPAGE+10
   1200      CALL CONLIN(1,0,0.)
¥ 1300      CALL GRXAXS(XS,YS,8.,XDT,IXF,-2,'DISTANCE (CM)',13)
¥ 1400      CALL GRYAXS(XS,YS,6.,YDT,IYF,2,'TEMPERATURE (C)',0)
   1410      XSM=XS-2.2
   1420      YSM=YS+1.5
   1500      CALL SYMBOL(XSM,YSM,0.28,'TEMPERATURE (C)',90.,15)
1510 ~ 1530      memPLDECORBのL. 2500~2700と同じ
   1540      CENTX=0.5*AL
   1550      CALL GRREF(1,CENTX)
   1560      CALL CONLIN(1,0,0.)
   1570      CALL GRFRAM
   1700      CALL CALNUM(TM,AL,TO,AK,EPS)
¥ 1710      WRITE(PGCMT, '(''TEMPPL'',12.2)') INT(TM+0.1)
¥ 1720      CALL PAGE(NPAGE,PGCMT)
¥ 1730      CALL PLOT(0.,0.,-3)
   1740      NPAGE=NPAGE+10
1800 ~ 2200      memPLDECORBと同じ、但し2000は1540へ移動
   3210      RETURN
   3300      END

```

図20 memPL4GRAPH

の4つのグラフに書いて、1枚の絵に納めた。1つずつのグラフを書くプログラムは、ほぼ同じなので、それをサブルーチンとし、PLGRAPHと名づけた。

また、この絵を図形検索する場合のことを考えて、それぞれのグラフを別々に図形検索できるように、各グラフにページ指定をすることにした。これは、図形検索を行うグラフィック・ディスプレイ上では、作画速度が遅いので、検索をできるだけ必要な部分のみに留めるためである。また、このページ指定を利用して不必要な部分を削除したり、適当な部分どうしを組み合わせたりすることもでき



### TEMPERATURE PROFILE

図21 memPL4GRAPH により作画された絵 (縮尺1/2)

る。これらの操作の詳細は第4章で説明する。

#### (1) 1枚の絵に複数のグラフを書く

1つのグラフを書くたびに、X、Y軸を決め直せば良い。それ以外は、1つのグラフを書くのと同じである。ここでは、subPLGRPH内のL.1300、1400でX、Y軸を決めており、その始点を適当に変えるために始点のユーザ座標値を引数でもらっている(L.99)。また、X、Y軸の長さは、それぞれ短くしてある。

L.60～90で、このサブルーチンを4回呼び4つのグラフを1枚の絵の中に納めている。その際、時間と始点のユーザ座標値とページ番号を引数で指定している。

## (2) 絵の各部分にページ番号を指定する

ページ番号指定は、各グラフ毎に限る必要はなく、1枚の絵の適当な部分に対して行うことができる。1枚の絵の部分というのは、連続して作画される1区切りのことであり、作画順序の飛び飛びの部分に、同一のページ番号を指定することはできない。また、1つのページをさらに適当な部分に分けて、フィグ番号(※10)を指定することができるが、ここでは詳細は割愛する。

このようなページ番号やフィグ番号の指定は、後に図形検索をする場合、あるいはグラフィック・ディスプレイ上で図形の合成や削除を行う際の便利のためである。

ページ番号の指定は、subPAGEとsubPLOTを組み合わせると行うことができる。また、フィグ番号の指定は、subPLOTで行うことができる。

### (ア) PAGE (文献2)

#### PLOT (※11) (文献2)

- 1 この2つのサブルーチンは、この順に連続して使われて1枚の絵の適当な部分にページ番号をつける。なお、PLOTSから最初のPAGE番号指定までの作画部分には、無条件に0番のページ番号が与えられる。
- 2 CALL PAGE(NPAGE,PGCMT)  
CALL PLOT(0.,0.,-3) (※12)
- 3 ☆NPAGE……1枚の絵の適当な部分に与えるページ番号を指定する。必ずしも1から順に指定する必要はない。但し、番号は4096を超えてはならない。  
☆PGCMT……5文字以上8文字以内の文字列である。このページに与えられるコメントであり、GDPモードでページ情報を出力させた時にページ番号とともに出力される。

---

※10 文献3および文献5では、グラフ番号という言葉を使っているが、ここでは紛わしさを避けるためにフィグ番号と呼ぶことにした。

※11 subPLOTは、他にも幾つかの機能を持つが、ここでの説明はページ指定に関する機能だけとし、他の機能については割愛した。

※12 subPLOTの1, 2番目の引数は、ユーザ座標原点の移動を行うためのものであり、この例のように同じX, Y座標を持ったグラフを座標原点をずらして複数個書くには、この引数を有効に利用する方が便利である。但し、グラフィック・ディスプレイ上に絵を表示させる場合には、若干の面倒が伴う。

☆ subPLOT の引数は、例のままにするのが無難である。

- 4 ここでは、1つのグラフをさらに軸やリファレンス線の部分と、温度分布のプロットなどの部分に分けてページ指定をしている。そのため、1つのグラフ内の作画順序を変えた。前者の部分には、L.1120, 1130 でページ番号を指定しており、後者の部分には、L.1720, 1730 でページ番号を指定している。ページ番号は、30から10ずつふやしていき、引数でやりとりしている。

PGCMTにはL.1110, 1710 のWRITE文で軸部かプロット部かを区別するとともに時間も書き込んで、ページの区別が明瞭になるように工夫している。

次の4章で、このプログラムを利用して図形検索および各部分の削除、合成等の操作を行う。

### 3. 7 1 回の実行で複数枚の絵を書く

図21の4つのグラフを、それぞれにタイトルをつけて4枚の絵として出力する。

プログラム (memPL4PICTR) を図22に示す。出力結果は図21の4つのグラフのそれぞれが、ひとつずつの絵になっただけなので割愛する。プログラムは図13のmemPLDECORBを基礎としており、それとの変更部分だけを抜粋して書いた。ここでも図20のmemPL4GRAPHと同様に、subPLGRPHを作った。ページ指定もmemPL4GRAPHと同様である。

ひとつのプログラムの実行で複数枚の絵を書くには、それぞれの絵の作画部分を、subPLOTS (L.1100) と subPLOTE (L.1860) で挟んでやれば良いのだが、それを GDP で覗くためには、それぞれの絵にメンバ名をつけておかなければならない (※13)。

それには subPLOTS を用いる。

subPLOTS は、これまで引数を3個として用いてきたが、このサブルーチンは4個目の引数を持つこともできる。その4個目の引数が、このPLOTSと次のPLOTEで挟まれた部分で作画する絵に与えられるメンバ名となる。このメンバ名は5文字以上8文字以下でなければならない。

同じメンバ名を重複して指定することはできないので、ここでは " MMBR " という文字変数を定義し (L.500)、時間によって異なるメンバ名を指定している (L.1010)。なお、L.1010

---

※13 直接プロットデータセットを作る場合 (図5の&g) または直接NLPへ出力する場合 (図5の&k) は、それぞれの絵が順番に順データセットに書き足されるだけである。従って、メンバ名を指定してはならない。即ちsubPLOTSの最後の引数 (ここでは " MMBR ") を削除すればよい。それに伴って、L.1010およびL.500の " MMBR \*8, " も削除される。

```

0020      CALL CONDEV('6202')
0030      CALL GRINIT
0060      CALL PLGRPH( 5.)
0070      CALL PLGRPH(20.)
0080      CALL PLGRPH(40.)
0090      CALL PLGRPH(80.)
0096      STOP
0097      END
0098      C *****
0099      SUBROUTINE PLGRPH(TM)
          1 0 0 ~ 4 0 0  memPLDECORB (図13)と同じ
0500      CHARACTER MMBR*8,ENDCMT*20
          6 0 0 ~ 8 0 0  memPLDECORBと同じ
¥ 1010      WRITE(MMBR,('TIME',I2.2)) INT(TM+0.1)
¥ 1100      CALL PLOTS(0.,0.,16,MMBR)
1120      CALL PAGE(30,'XYAXES')
1130      CALL PLOT(0.,0.,-3)
1150      CALL CONCHA(0.28)
1 2 0 0 ~ 1 5 0 0  memPLDECORBと同じ
1510      CENTX=0.5*AL
1 5 2 0 ~ 1 5 5 0  memPLDECORBのL. 2 5 0 0 ~ 2 8 0 0 と同じ
1560      CALL CONLIN(1,0,0.)
1570      CALL GRFRAM
1700      CALL CALNUM(TM,AL,TO,AK,EPS)
1720      CALL PAGE(40,'TEMPPL')
1730      CALL PLOT(0.,0.,-3)
1800      CALL GRPGN1(XDT,SUM,IXF2,IYF2,0)
1810      CALL PAGE(50,'TITLE')
1820      CALL PLOT(0.,0.,-3)
1 8 3 0 ~ 1 8 4 0  memPLDECORBのL. 2 9 0 0 ~ 3 0 0 0 と同じ
¥ 1850      WRITE(ENDCMT,('TEMP.PROF.TIME=',I2.2,'MIN')) INT(TM+0.1)
¥ 1860      CALL PLOTE(ENDCMT)
1870      RETURN
1880      END

```

図22 memPL4PICTR

で用いられている " I2.2 " は、整数を2カラムとって (I2)、桁数を必ず2桁にして (.2) 出力することを指示している (※14)。

- 
- ※14 TM=5.0 のときは、MMBR='TIME05' となり  
 TM=10.0 のときは、MMBR='TIME10' となり  
 TM=240.0 のときは、MMBR='TIME \*\*' となる。

※15 データセット名は、ONLINE.OUTLIST となる。

結局、このプログラムmemPL4PICTRを実行すると、時間5, 20, 40, 80分の温度分布曲線が、それぞれTIME05, TIME20, TIME40, TIME80というメンバ名のグラフィックデータセット（※15）にできることになる。

このプログラムを繰り返して実行すると、それぞれの絵に同一のメンバ名を指定してしまうことになるので、その場合はメンバ名を変更するか、当該メンバを消去しておかなければならない。

そのような面倒を避けるには、それぞれの絵につけるメンバ名を、いちいち入力してやればよい。それにはL.1010を次の2行に置き換え

```
1010 PRINT '( ' INPUT MEMBER NAME FOR THE PICTURE '
      + // ' ' AT TIME= ' ',F3.0) ',TM
1020 READ(5,*) MMBR
```

それぞれの絵に対してメンバ名を入力してやれば良い。

なお、このように subPLOTS が4つの引数を持つプログラムの実行は、次のようにする。

TSS では、Fコマンドを用い PLOT パラメータに論理機番即ち " 16 " を指定する（図5の &b）。

```
F CL DENNETSU EL( ' PL4PICTR,SUBCALNM ' ) PLOT(16)
```

この場合、メンバ名の入力要求はして来ない。

また、バッチでは、FBコマンドを用い図6と同様に

```
FB CL DENNETSU(PL4PICTR)/DENNETSU(SUBCALNM) PLOT(16)
```

と入力し（図5の&c）、#2で " MMB " と入力する。そうすると、#3ではデータセット名だけの入力要求があり、メンバ名の入力要求はない。そこで " ONLINE.OUTLIST " と入力すれば良い。後は図6と同様である。

## 第4章 グラフィック・ディスプレイ上での操作 (GDPの諸活用法) (※1)

### 4. 1 GDP を始める前に

既に2. 2 (2) で述べたようにセンターにはグラフィックディスプレイが3台ある。ここでは、F9431またはF9434を使う場合を中心に説明する。

まず、READYモードでGDPと入力する。この状態をGDPモードと呼ぶことにする。GDPモードで入力できる制御文には、次のものがある。これらの制御文の詳細については、文献3または文献5の第3章を参照せよ。

MODE, PRINTF, DISPF, PLOT, END

但し、これらの制御文を入力する時は、先頭に1個以上の空白を入れなければならない。これらのうち制御文DISPFを入力すると、さらに会話コマンドを入力することができる。この状態を、DISPFモードと呼ぶことにする。このDISPFモードでは、次の会話コマンドを入力することができる。

COPY, DELETE, KEEP, LIST, END, FREE, MOVE, NEXT, SCALE

\_\_部は、それぞれの簡略形である。DISPFモードでは、カーソルは画面左上 (TEK4114では左下) にあり、そこが会話コマンドの入力位置である。但し、これらの会話コマンドの先頭には空白はいらない。

ここでは、図20のプログラムPL4GRAPHで作画した図21を例にして、この絵の必要なページだけを図形検索する方法、および会話コマンドを用いて、時間20と40の曲線を1つのグラフにまとめて出力する方法を紹介する。

### 4. 2 絵の一部分のみを図形検索し、その絵全体を出力する

例えば図21の右上の部分 (TIME=20MIN) の温度分布曲線のみを図形検索するには図23のようにする。

---

※1 GDPは通常の端末でも動作するが、図形を描かせるためのDISPFコマンドはグラフィックディスプレイ (2. 2 (2) 参照) でしか動作しない。



```

READY
#1 GDP
#2 MODE IN=FT01F001,OUT=FT04F001,OUTDEV=TEK
FACOM OSIV/F4 GDP V02L20 DATE 82.06.03 TIME 13.49.49
      PAGE 1
MODE IN=FT01F001,OUT=FT04F001,OUTDEV=TEK
#3 PRINTF +,FORM=1
PRINTF +,FORM=1
#4          GRAPHIC FILE LISTER LISTFD
DATE 82.06.03          PAGE 1
#4          MEMBER-NAME = DECO1          MAKE-DATE = 82.06.03          TEMP.PROF.DECO
#4          ( 1) LEVEL1          ID = 1
#4          ( 1. 1) LEVEL2          ID = 0          SIZE = 13
056
#4          GRAPHIC FILE LISTER LISTFD
DATE 82.06.03          PAGE 1
#4          MEMBER-NAME = SYMBL1          MAKE-DATE = 82.06.03          TEMP.PROF.SYM
0L
#4          ( 1) LEVEL1          ID = 1
#4          ( 1. 1) LEVEL2          ID = 0          SIZE = 13
824
#5          GRAPHIC FILE LISTER LISTFD
DATE 82.06.03          PAGE 1
#5          MEMBER-NAME = TEST1          MAKE-DATE = 82.06.03          TEMP.PROF.4GR
PH
#5          ( 1) LEVEL1          ID = 1
#5          ( 1. 1) LEVEL2          ID = 0          SIZE =
768
#5          ( 1. 2) LEVEL2          ID = 30          SIZE = 4
608          XYAXES05
#5          ( 1. 3) LEVEL2          ID = 40          SIZE = 1
536          TEMPPL05
#5          ( 1. 4) LEVEL2          ID = 50          SIZE = 4
608          XYAXES20
#5          ( 1. 5) LEVEL2          ID = 60          SIZE = 1
536          TEMPPL20
#5          ( 1. 6) LEVEL2          ID = 70          SIZE = 4
608          XYAXES40
#5          ( 1. 7) LEVEL2          ID = 80          SIZE = 1
536          TEMPPL40
#5          ( 1. 8) LEVEL2          ID = 90          SIZE = 4
608          XYAXES80
#5          ( 1. 9) LEVEL2          ID = 100          SIZE = 1
536          TEMPPL80
#5          ( 1. 10) LEVEL2          ID = 110          SIZE =
768          TITLE
#6 DISPF TEST1,PAGE=60
DISPF TEST1,PAGE=60

      図形表示 (DISPFモード)
#7 END

```

図23 ページ指定を持つ絵の検索手順

### 【図23の説明】

＃2までは前述のまま（ref. 図7）。このまま入力する。

今度は、ページ情報も知りたいので＃3と入力する。この場合メンバ名を指定していないので、ユーザがグラフィック・データセット上に保有している全てのメンバについて出力されるが（＃4、＃5）、メンバ名が予めわかっている（例えばTEST1とすると）、次のようにすればよい。

```
└PRINTF└TEST1,FORM=1
```

この場合は、＃5だけが表示される。表示は、ディスプレイのカラム数が不足しているため1行分が改行されて2行となって表示されるので、注意を要する。第2行目に、メンバ名とPLOTに書いたコメント等が表示される。1行おいて、ページ情報の表示が始まる。第3項目に“ID=”で表示されているのがページ番号である。1項目おいて（改行されて）subPAGEで指定したコメントが表示されている。あとで参照する便利のために、この画面はハードコピーをとっておくと良い（TEK4114では左下の英数字表示行が4行しかないが、右横のローラーで簡単にスクロールできるので、ハードコピーをとる必要はない）。[SHIFT]キーを押しながら[PRINT]キーを押すと、そばのグラフィックプリンターにコピーされる。コピー中はカーソルが消え、どんな入力も受け付けない。コピーが終了すると、またカーソルが現われる。

ここでは、TIME=20MINの温度分布曲線のみを表示させたいので、ページ60を表示させればよい。

```
└DISPF└TEST1,PAGE=60
```

と入力する（＃6）（※2）。画面は一旦、全て消去され目的の曲線のみと、この図形のメモが表示される（図24）。メモは、画面右上から表示され始めるが改行されて左上へ続く（TEK4114では左下に改行しないで1行で表示される）。メモの内容は、先頭から識別番号（01）、メンバ名（TEST1）、ページ番号（000060）、フィグ番号（××××××）である。ページ番号は途中で改行されてしまう。また、フィグ番号は指定していないので“×”が入っている。作画が終了すると、カーソルはアンダーラインの形となって左上に現われる（※3）（TEK4114では広いバンドのまま左下のメモの最後尾に戻る）。

この曲線だけで絵全体の判断がつけば、ここで図形検索をやめれば良い。更に別のページ（例えばページ50）を表示させるには、DISPFモードのまま

```
└NEXT TEST1,50
```

---

※2 ページ指定を省略すると、全ページ即ち1つの絵全体が表示される。

※3 READYモード、GDPモードでは、カーソルは広いバンドである。

とすれば良い（TEK4114では、前の図形のメモの直後からそのままタイプする）。但し、この場合は表示されていた図形（ページ60）は消去され、新しいページ（ページ50）のみが表示される。複数のページを重ねて表示する方法は、次の第3節で説明する。

図形検索を終らせるには、DISPFモードで

END

と入力してDISPFモードを終らせカーソルがバンドに戻ってから、更に

␣END

と入力して（#7）GDPモードを終らせる。

この絵全体をNLPに出力するには、2. 3で述べた方法を用いれば良い。但し、COPY ␣2を用いる場合は注意を要する（ref. 4. 3）。

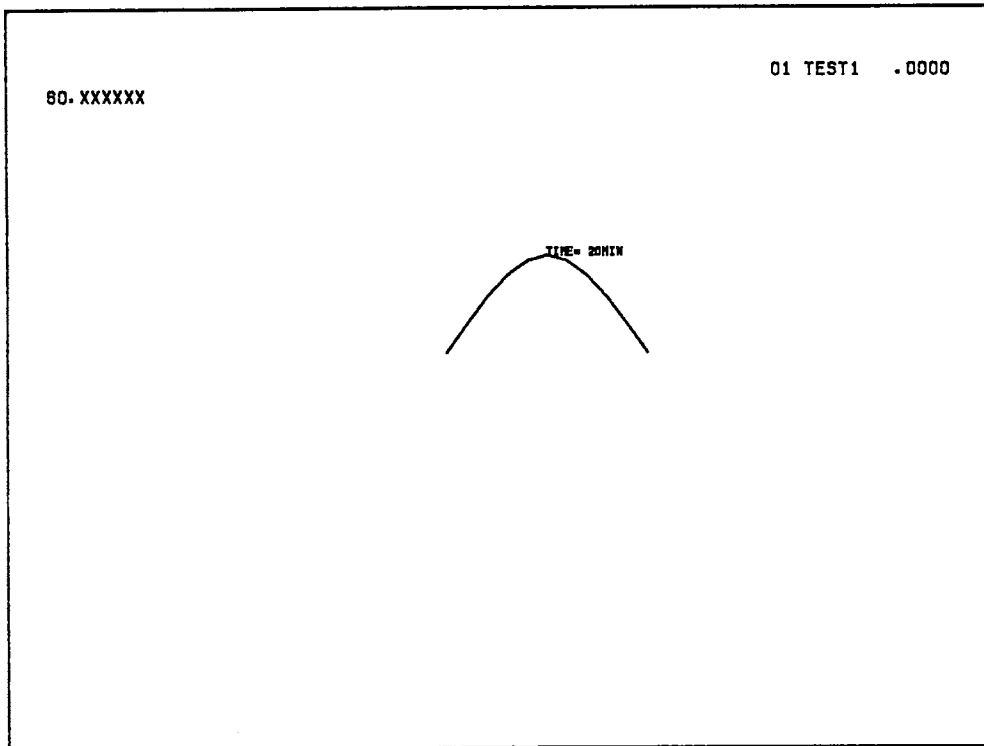


図2 4 ページ60を表示したところ

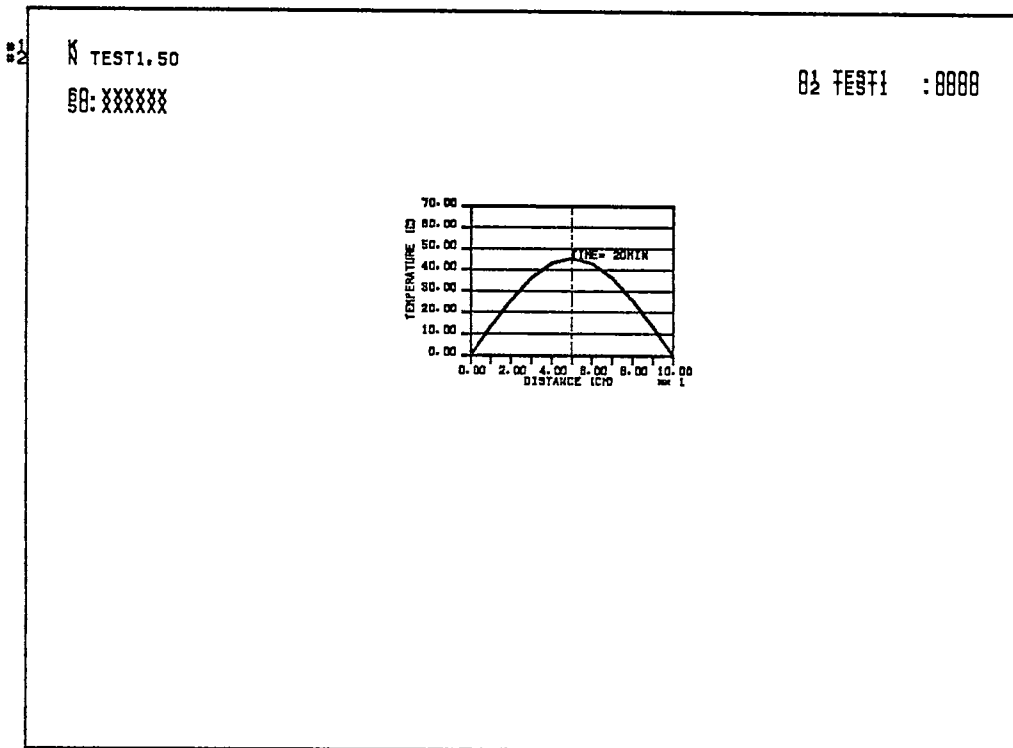


図25 ページ60と50を同時に表示したところ

#### 4.3 絵の一部分のみを取り出して出力させる

最初のページを表示させるまでは、第2節と同じ。

ここで、この上に更にページ50を書き足したい時はDISPFモードで、まず

KEEP

と入力する(図25の#1)。この会話コマンドによりページ60は、画面上に保持される。このようなページを保持モードにあるという。次に

NEXT TEST1,50

とする(図25の#2)。そうするとページ60の上にページ50が書き足される。この状態の表示を図25に示す。

このページ50のように、まだ保持されていないページ、および保持モードから解放されたページを、動作モードにあるという。

ここで、更に

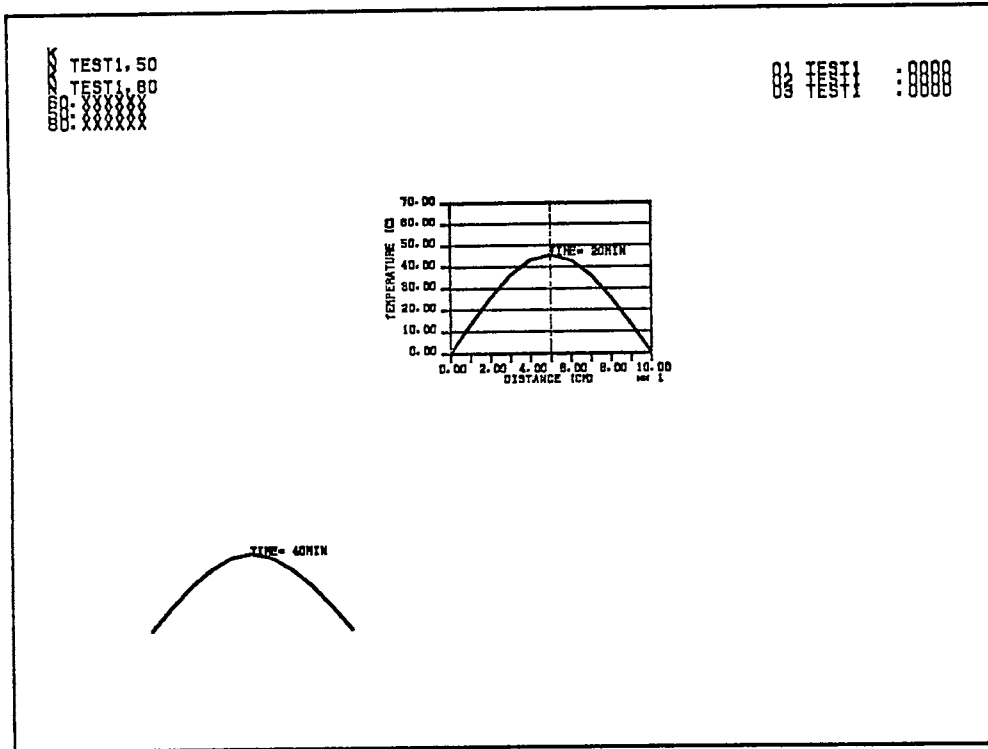


図26 ページ60, 50, 80を同時に表示したところ

### NEXT TEST1, 80

とするとページ50は保持されていないので、ページ50は消去され、ページ60と80だけが表示される(※4)。ページ50も必要な場合には、NEXTを入力する前にKEEPを入力し保持モードにしておかなければならない。なお、保持できる図形の数、最大9個までである。従って、動作モードの図形と合わせて、最大10個の図形を画面上に表示することができる。

また、表示されている図形の一部を消すには(※4)

### DELETE 識別番号

と入力すれば良い。識別番号とは、画面右上の表示の左端に示される"01"のような番号である。

※4 GDPモードでは、表示されている図形のうち他の部分を表示させたままで一部のみを消すという動作はできない。一部を消すということは、一部を除いた図形を最初から表示し直すということである。

入力には、“0”を略して“1”だけで良い。識別番号を省略すると、動作モードにある図形以外の全ての図形が消去される。

こうして、適当なものだけを画面に表示させる。この表示されている部分のみをNLPに出力するには、ここで

#### COPY 2

と入力(※5)(図5の&f)した後にDISPFモード、GDPモードを終らせ  
READYモードで

PSPNLP GDPOUT.OUTLIST

と入力すれば良い(図5の&i)。

なお、COPY 2は、現在画面に表示されている図形(保持モードも動作モードも含む)のみを、プロット用データセット(GDPOUT.OUTLIST)に書き換える機能を持っている。

また、PSPNLPは、プロット用データセットをNLPに出力するコマンドである(※6)。

#### 4. 4 表示図形の移動、拡大、縮小、回転等を行い、適当な位置で適当な大きさにした後出力させる

これらの事は、全てDISPFモードの会話コマンドを用いて行う。

ここでは図21の2つの時間(20と40MIN)のグラフを1つのグラフ(図32)にまとめ、このグラフだけの絵として出力してみる。

まず、ページ60を表示させ保持する。

次に、ページ50を表示させ(図25)、これも保持する。

更に、ページ80を表示させる(図26)。

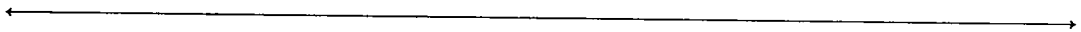
そしてこれを、ページ50,60の図形と重ね合わせる(図28)。それには

MOVE  └X方向移動距離,Y方向移動距離

とすればよい。移動距離の単位はcmである。なお、MOVEコマンドで移動するのは、動作モードにある図形だけである。移動距離がわからない場合は

SCALE

と入力し(図27の#1)(※7)、cm単位のスケールを表示させて調べる(図27)。スケール



※5 COPY 2の“2”は、いつも“2”である。

※6 2.3を参照せよ。

また、B PSPCNV D(GDPOUT) R(1024K)としてもよい。

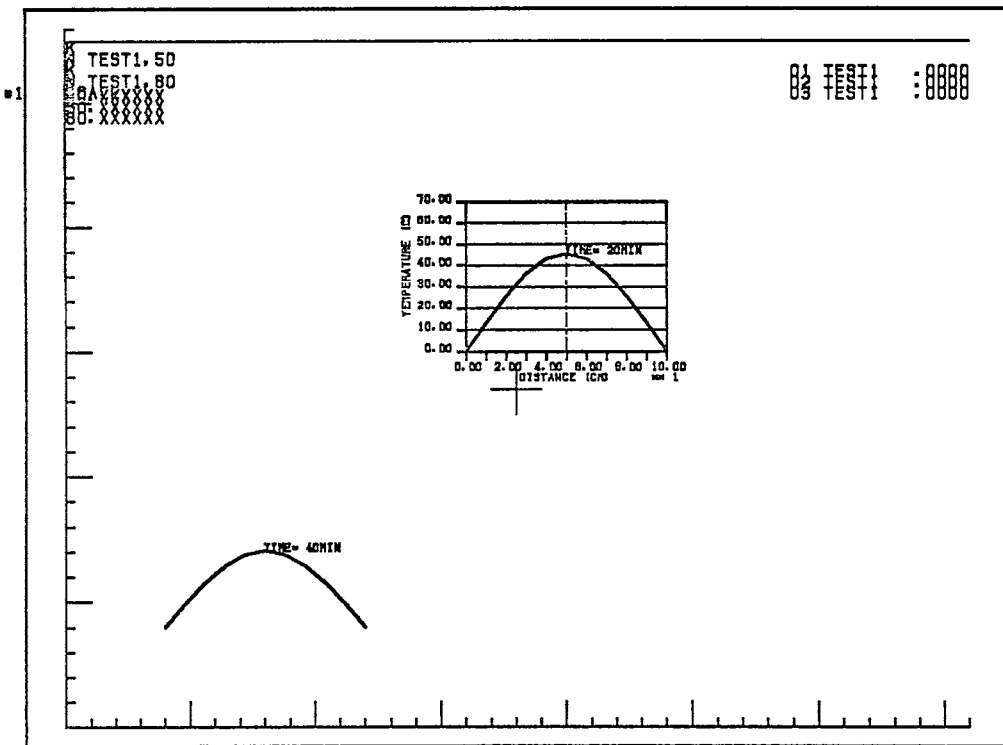


図27 スケールを表示させたところ

を消すには、もう1度SCALEと入力すれば良い。なお、スケールはNLPには出力されない。

各移動距離がわかったので

`MOVE 12,11`

と入力する。画面は、一旦、全て消去され保持モードであるページ60、50は、前と同じ位置に表示されるが、動作モードであるページ80の図形は移動して、ページ50、60の図形と重なる

(図28)。必要ならば、更にいくつかの図形を表示し移動させる。こうして適当な図形ができれば、それをNLPへ出力させれば良いが、このままでは図の位置が右上へ偏っているし、(NLPへは、表示画面のうち左下から25×25cmの範囲の図形だけが出力される)、サイズも小さい。そこで、この図形を2倍の大きさにし、中心付近へ移動させる。図形を拡大、縮小するのもMOVEコマンドの機能である。即ち次のようにする。

※7 右上から改行されたメモに重なって見にくくなる。

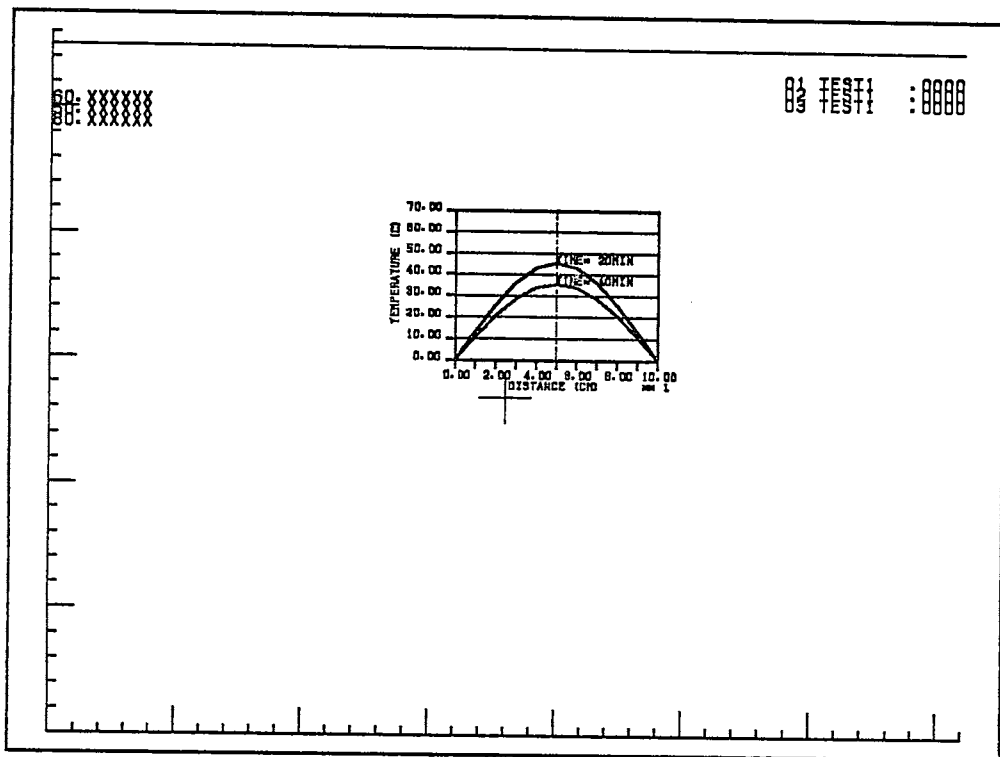


図28 ページ80 を移動させて(M 12,11)  
ページ60, 50 に重ね合わせたところ

MOVE  $\square$ X方向移動距離, Y方向移動距離, 全体の倍率

または

MOVE  $\square$ X方向移動距離, Y方向移動距離, (X倍率, Y倍率, 文字倍率)

MOVE コマンドは動作モードにある図形だけを対象とするので、ここではまず動作モードにあるページ80を拡大、移動する。

MOVE -13, -11, 2

とすると、ページ80は2倍に拡大され(7,8)が左下端になるように移動する(図29)。

その際の計算には、次の注意を要する。

- ・図形の表示位置は、いつも最初の表示位置から計算される。
- ・同一の図形について複数の MOVE コマンドを用いた場合は、移動距離は加算される。

このページ80の場合を曲線の左下端について考えていくと、最初の表示位置は(4,4)である(ref. 図27)。倍率を2倍にしたので、表示位置は(8,8)となる。移動距離は、前に、



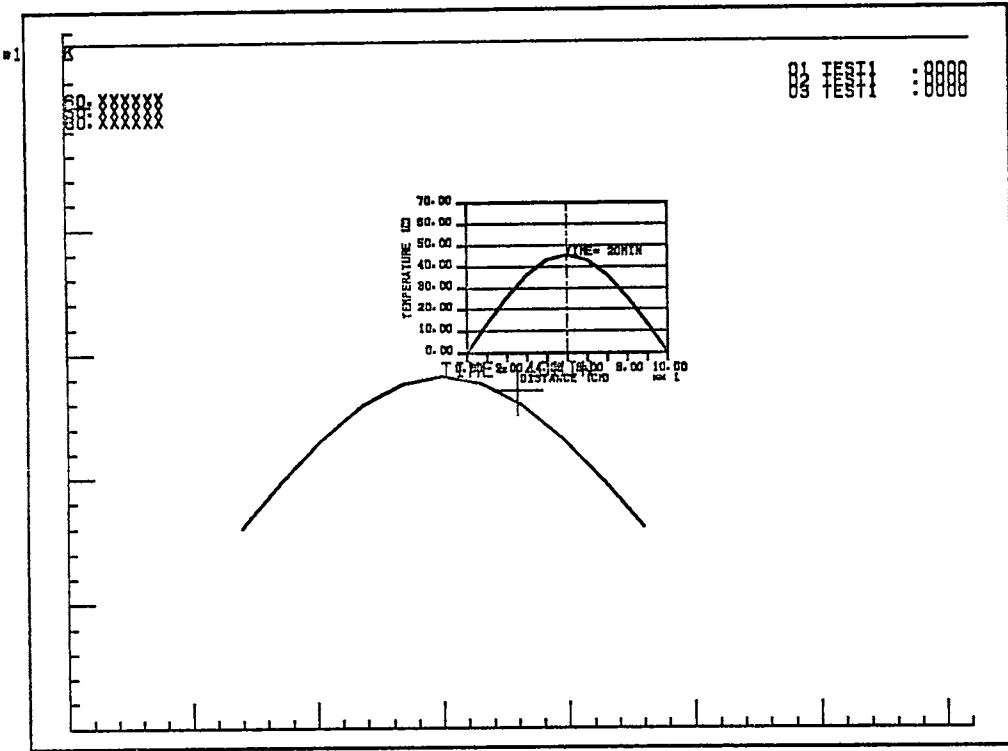


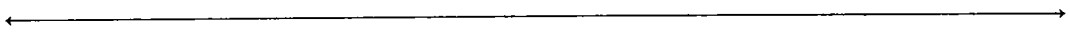
図29 ページ80 を拡大・移動させたところ(M -13,-11,2)  
 ページ60, 50 は、そのまま

X方向=12、Y方向=11を指定しているなので、ここでの指定と加算すると、X方向=-1、Y方向=0となる。従って、最終表示位置は(7,8)となる(※8)。

この拡大、移動したページ80を保持して(図29の#1)、次にページ50を拡大移動する(※9)。それには、まずページ50を動作モードにしなければならない。それには、次のようにする。

FREE 2

FREE の後の数字は、識別番号である。FREE コマンドを入力すると、画面は一旦全て消去され、前と全く同じ図形が表示されるが、指定された図形が動作モードとなっている。ページ50を拡大移



※8 このような計算が面倒ならば、NEXT TEST1,80 と入力して、もう一度、新たにページ80を表示させ、MOVE -1,0,2とすれば良い。

※9 ページ50 が先でも60 が先でも良い。

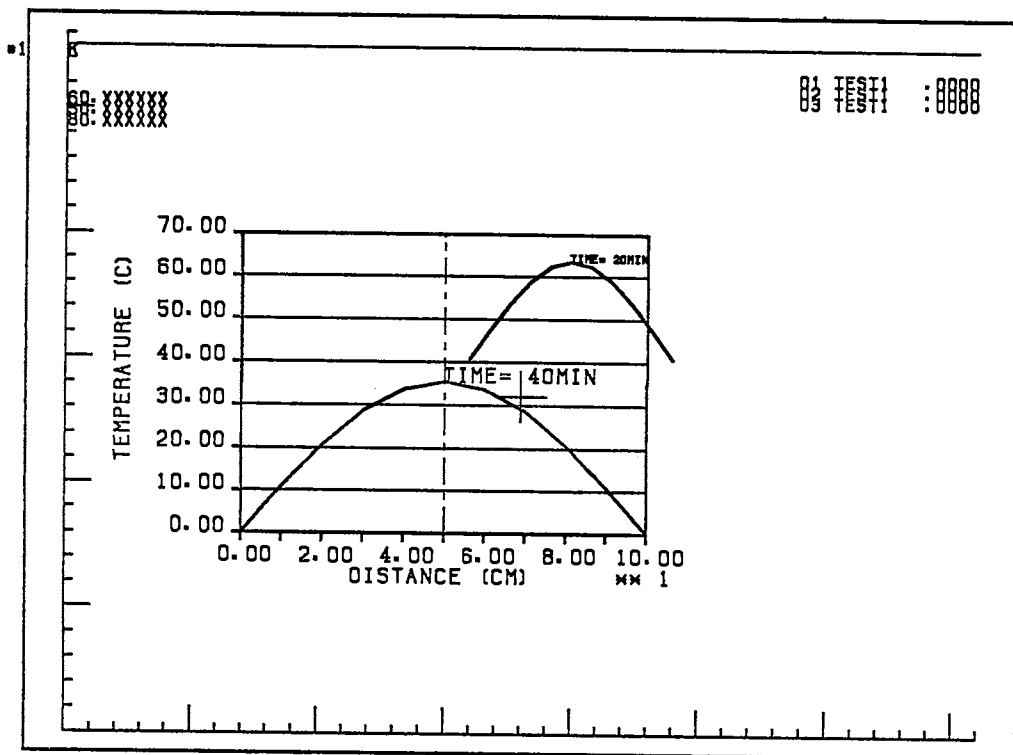


図30 ページ50も拡大・移動して(F 2), (M -25, -22, 2)  
ページ80と重ね合わせたところ

動して、ページ80に重ねる際の移動距離 $\Delta X$ 、 $\Delta Y$ は、次のように計算する。

$$(7, 8) = 2(16, 15) + (\Delta X, \Delta Y)$$

a      b      c

$$\therefore (\Delta X, \Delta Y) = (-25, -22)$$

なお、aは移動後の位置、bは倍率、cは現在の表示位置である。

したがって、この場合の MOVE コマンドは、次のようになる。

MOVE -25, -22, 2

これでページ50は拡大移動して、ページ80に重なる(図30)。ページ50も保持し(図30の#1)、同様にページ60もページ80、50に重なる。今度は、文字だけは拡大しないことにしよう。次のようにする。

FREE 1

MOVE -25,-22,(2,2,1)

これで目的にした表示画面は、ほぼできたが、さらにタイトルを書かせてみよう。まず、ページ60も保持し(図31の#1)、次にタイトルのページ110を表示させる(図31)。

NEXT TEST1,110 (図31の#2)

文字のサイズを2.5倍にし、作画開始点を(9,1)から(2,3.5)に移動させる。

MOVE -7,2.5,(1,1,2.5)

倍率は、全体倍率で指定するよりも、X、Y倍率を、それぞれ1にして文字倍率だけを2.5とした方が、移動距離の計算が簡単である。これで目的の図形はできた(図32)。

これをNLPへ出力させれば良いから(※10)、第3節で述べたように、COPY 2と入力し(図32の#1)、DISPFモード(図32の#2)、GDPモードを終らせ、READYモードで、PSPNLP GDPOUT.OUTLISTと入力すれば良い。ただし、前述したようにプロッタ用データセットGDPOUT.OUTLISTは、順データセットなので、このデータセットに入っている全ての図形が出力される。従って、一旦NLPへ出力したら、GDPOUT.OUTLISTは消去して

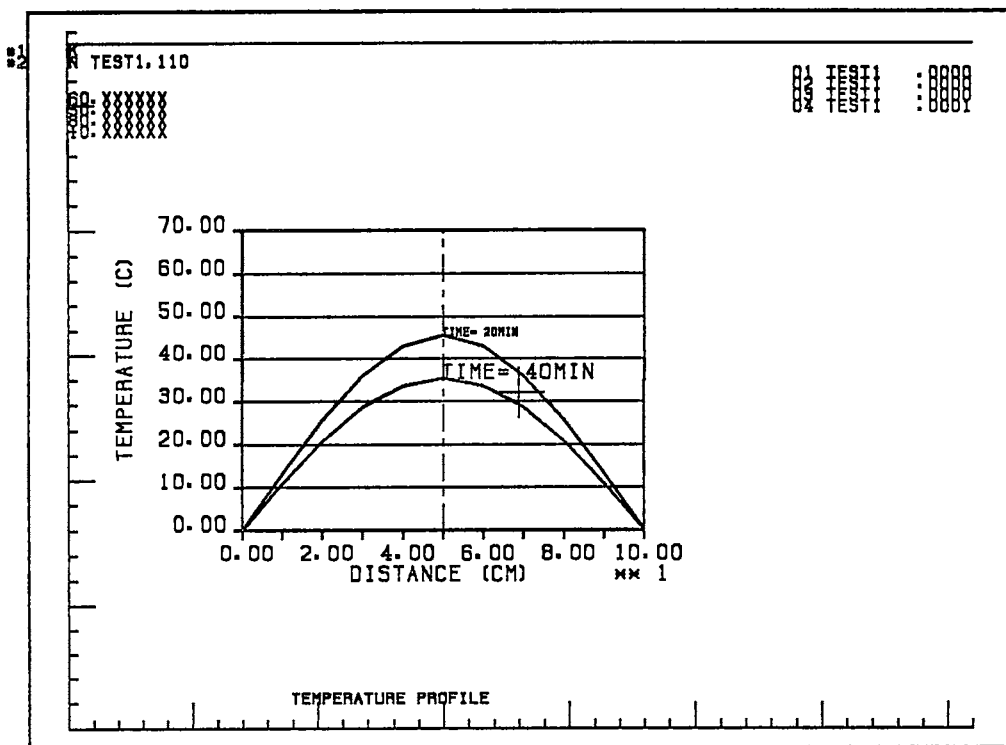


図31 ページ60も拡大・移動して(F 1)、(M -25,-22,(2,2,1))ページ80, 50と重ね合わせタイトルを表示したところ

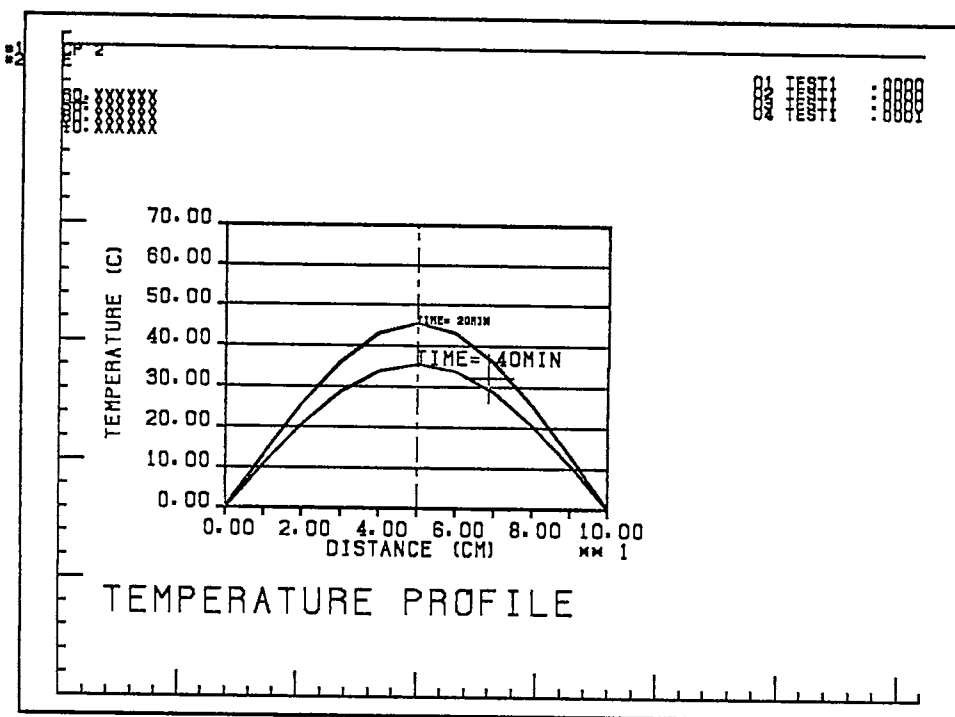


図3.2 タイトルを拡大・移動して(M -7,2.5, (1,1,2.5))できた最終図

おいた方が良いでしょう。

なお、表示図形を回転させることもできるが、ここでは詳細は割愛し簡単な説明にとどめる。回転の中心は表示画面上(18,13.5)の点でありスケールを表示させると“+”印で示される。

この機能を持つのは MOVE コマンドであり前述の機能と合わせて MOVE コマンドの一般形は次のようになる。

MOVE [X方向移動距離, Y方向移動距離, {全体の倍率 | (X倍率, Y倍率, 文字倍率)}, 回転角度

回転角度は反時計廻りを正とし、単位は度である。移動と回転を同時に指定した場合には移動が先に計算されてから回転が行なわれる。

従って、例えば図2.4で

※10 スケールおよび右上、左上の会話コマンドやメモはNLPには出力されない。

M\_2, -1.5, 1, 90

とすれば、ページ 60 は曲線の左端がちょうど回転中心の (18, 13.5) に移動し、そこで左端を中心に 90 度回転することになる。

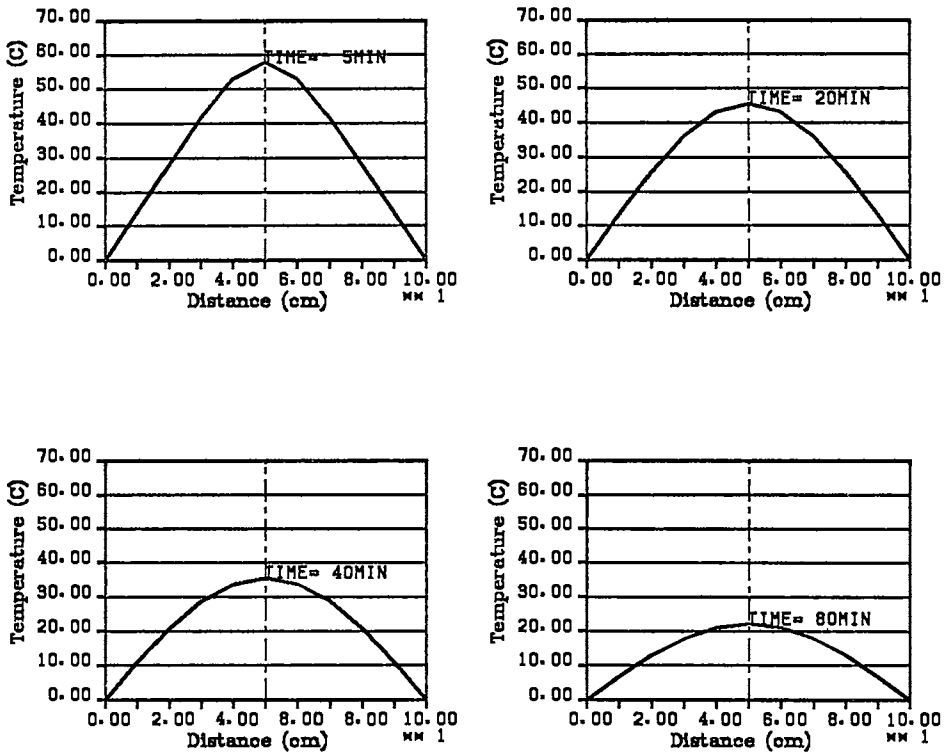


図 3 3 SAS 文字と組み合わせた絵

#### 4. 5 タイトルと X、Y 軸の標題の文字に SAS を用いる

まず、図 2 0 の memPL4GRAPH を若干修正し、X、Y 軸の標題とタイトルを作画しないようにする。また、SAS を用いて X 軸の標題、Y 軸の標題、タイトルを作画するプログラムを、それぞれ別々に作る (※ 1 1)。

これらを第 4 節で述べた方法で合成すれば良い。

まず、X、Y 軸の標題とタイトルのない図 2 1 に相当する図形を表示させ保持する。

次に、SAS を用いた X 軸の標題文字を表示し、適当なサイズで適当な位置に移動し保持する。また、新たに X 軸の標題文字を表示し、同様に別の位置に移動し保持する。このような操作を繰り返して、X 軸の標題文字が、全てできたら、Y 軸の標題文字、タイトルについても同様の操作を行う。このようにして得られたのが、図 3 3 である (小文字の " p " が正常であることに注意)。

---

※ 1 1 文献 1 0、1 1 を参照せよ。SAS 文字だけを書かせるには、PROC GSLIDE が便利である。

## 第5章 計算と作画を別々のプログラムで実行する (長い計算時間または倍精度を要する場合)

この手引書では、簡単な問題を取り扱っているので計算時間は比較的短いですが、実際にはもっと長い計算時間を必要とする場合の方が多いであろう。従って、1つのプログラムで計算も作画も行うと、TSS では実行できなかったり、また、バッチにしても1クラス上を指定しなければならないようなことが起こってくる。このような場合、むしろ計算と作画を別々のプログラムで実行させた方が能率的である。

また、倍精度計算を行う場合には " GRACE IV " の諸サブルーチンの引数は全て単精度型であるので、いちいち型指定をしなければならない。このような面倒を避けるには、計算は倍精度で行ない作画すべき結果は、単精度で出力しておく。その結果を読み取って作画は別のプログラムで行なえばよい。

### 5. 1 計算結果を全て同じデータセットに書き出す方法

計算用プログラム (memNUMERIC2) と、その実行の手順を図34に示す。このプログラムは、図2のmemNUMERICとほぼ同じである。

[図34の説明]

井1のL.700が計算結果を指定したデータセットに書き出している所である。WRITE文のファイル指定(※1)が"60"になっていることに注意されたい。この番号は5,6,7,16を除く1から99までの整数ならなんでもよい。また、FORMATが"12F15.5"となっており、通常のラインプリンターへの出力の際の制限カラム数を越えていることに注目されたい。但しTSS画面上では255カラムまでしか覗くことができないので、255以下にしたほうが便利である。

なお、このデータセットを覗くには、このプログラムの実行が終ってから書き出したデータセットをEDITしてやれば良い。この場合は

```
E NUMERIC.OUTLIST NON
```

※1 WRITE文で指定しているのは、正確には論理機番であるが、論理機番を決めればファイルは1つに決まってしまうので(FT 論理機番F001)、ここでは煩雑さを避けるために論理機番とファイルを同義語として扱う。

```

L DENNETSU.FORT77(NUMERIC2)
  AB9999.DENNETSU.FORT77(NUMERIC2)
  00100      PARAMETER (AL=100., T0=70., AK=12., NTIME=7)
  00200      COMMON/CNM/ SUM(0:10)
  00300      DIMENSION TIM(NTIME)
  00400      DATA TIM/1.,5.,10.,20.,40.,80.,240./, EPS/1.E-2/
  00500      DO 10 J=1,NTIME
  00600      CALL CALNUM(TIM(J),AL,T0,AK,EPS)
#1 00700      WRITE (60,'(12F15.5)') TIM(J),(SUM(I),I=0,10)
  00800      10 CONTINUE
  00900      STOP
  01000      END
      END OF DATA
      READY
#2 F CL DENNETSU EL('NUMERIC2,SUBCALNM') F(60)
  FORT7CLG STARTED TIME = 19:20:27
#3 LOGICAL UNIT NO. 60 : ENTER DATA SET(PS) NAME
#4 NUMERIC.OUTLIST
#5 LOGICAL UNIT NO. 60 : ENTER UNFORM OR FORM
#6 FORM
  FORTRAN 77 COMPILER ENTERD
  END OF COMPILATION
  LINKAGE START ***LIB( 'APPF.LINKLIB')***
  **MEMBER NAME** TEMPNN WAS NOT FOUND BUT HAS BEEN ADDED TO LIBRARY.
  MAX CODE=12 ENDED TIME=19:21:34
  READY

```

図34 memNUMERIC2 とその実行手順

と入力し、FSモードにして〔PF11〕、〔PF10〕のキーを操作してやれば良い。

なお、その際に適当にデータを変更したり、削除、追加したりすることもできる（※2）。

このプログラムの実行に際しては、WRITE文で指定したファイルに適当なデータセットを割り当てなければならない。それには、TSSでは#2のように入力する。Fパラメータの“60”がWRITE文で指定したファイル番号である。すぐにファイル60に割り当てる順データセット名を聞いてくる（#3）ので適当な名前のデータセットを割り当てる（#4）。属性は“DATA”または“OUTLIST”とする。OUTLISTとすれば、このデータセットは翌日の朝には勝手に消去される。DATAとすれば、勝手には消去されない。

※2 FコマンドのOUT パラメータで指定したデータセットの場合は、そのデータセットの内容を変更することはできない。



そのデータセット名が新しいものならば、そこに書き出される結果がFORMATつきか、そうでないかを聞いて来る（#5）ので、返答をしてやる（#6）。この場合は、FORMATつきなので（L.700） “ FORM ” と入力する。

既存のデータセット名を指定した場合は、#5の入力要求はなく、既存のデータセットの上に（古いものを消して）書き出す。

あとは、通常の実行の場合と同じである。

なお、バッチで実行するには#2の代わりに、TSS 端末から

```
FB CL DENNETSU (NUMERIC2)/DENNETSU (SUBCALNM) FILE (60)
```

と入力する。TSS の場合と同様な入力要求が来るので、同様に返答してやれば良い。

この結果を作画用プログラムに読ませて作画させれば良い。

作画用プログラムを memPONLY と名づける。このプログラムは、次の3行を除いては図13の memPLDECORB と全く同じである。変更点は、

L.100の “ TO=70. , AK=12. , ” が不必要のため削除されること。

L.300の COMMON 文が不必要になるかわりに、SUM(0:10) の配列宣言が必要になること。

L.1700の CALL CALNUM の代わりに

```
READ (50, '(12F15.5)') TIM(J), (SUM(I), I=0, 10)
```

が入ることである。

この READ 文が、計算用プログラムで書き出した結果を読み込む所である。READ 文の指定ファイル（※1） “ 50 ” も WRITE 文の場合と同様に適当なものを選べば良い。FORMAT は、計算用プログラムの WRITE 文のものと同一でなければならないのは当然である。

このプログラムの実行は図34と同様であり、TSS では

```
F CL DENNETSU (PONLY) PLOT (G) F (50)
```

と入力する。ファイル50に割り当てるデータセットを聞いて来るので、計算用プログラムの実行で結果を書き出したデータセット名（ここでは、NUMERIC.OUTLIST）を入力する。今度は、既存のデータセットなので、FORM か UNIFORM かは聞いて来ない。あとは、通常のグラフ用プログラムの実行と同様である。（ref. 2. 2 (1) (ア)）

バッチによる実行も同様なので割愛する。

## 5. 2 計算結果を区別して別々のデータセットに書き出す方法

今度は、計算結果を時間別に異なるデータセットに書き出し、そのうちの適当な時間の結果だけを作画する方法を示す。

計算用プログラム（memNUMERIC3）を図35に示す。プログラムは、図34の

```

00100      PARAMETER (AL=100., TO=70., AK=12., NTIME=7)
00200      COMMON/CNM/ SUM(0:10)
00300      DIMENSION TIM(NTIME)
00400      DATA TIM/1.,5.,10.,20.,40.,80.,240./, EPS/1.E-2/
00410      NFILE=60
00500      DO 10 J=1,NTIME
00600      CALL CALNUM(TIM(J),AL,TO,AK,EPS)
00700      WRITE (NFILE,'(12F15.5)') TIM(J),(SUM(I),I=0,10)
00710      NFILE=NFILE+1
00800      10 CONTINUE
00900      STOP
01000      END

```

図35 memNUMERIC3

memNUMERIC2 とほぼ同じである。memNUMERIC2 と異なる点は、結果を書き出すファイル番号を時間によって、いちいち変えている点だけである。

実行の方法もほぼ同じであるが、使用するファイルの全てに、いちいち異なるデータセットを割り当ててやらなければならない。まず

```

F CL DENNETSU EL ('NUMERIC3,SUBCALNM')
F (60/61/62/63/64/65/66)

```

と入力する。あとは図34と同様でありファイル60から順番に66まで割り当てるデータセットを聞いてくるので、それぞれに返答してやれば良い。

作画用プログラムは割愛する。特定の時間の温度分布曲線を1本だけプロットするのなら、memPONLY を若干変更して用いても良いし (L.1600 と、L.2300 を削除すれば良い)、また、図10と同様な絵を書くのなら、memPONLY の READ 文 (L.1700) のファイル番号を、memNUMERIC3 で用いたと同様の方法で次々に変え、それぞれのファイルにそれに対応したデータセットを割り当てれば良い。

この手引書で紹介した作画用サブルーチン一覧

サブルーチン名	機能の概要	説明ページ	文献
ARROW	矢印の作画	129	4
CBCIE	整数型データを文字に変換	114	1
CBCRE	実数型データを文字に変換	112	1
CBTXTC	文字列の作画	111	1
CONCHA	作画文字のサイズ指定	114	1
CONDEV	省略		1
CONLIN	作画に用いる線の線種, 太さ等の指定	110	1
GCIRC3	端点と半径から円弧を作画	121	1
GIFIT1	2次元点列のスプライン補間	123	1
GIFIT2	2次元点列の円弧補間	121	1
GLINE1	点列を直線でつなぎ合わせる (ピッチ優先)	124	1
GLINE2	点列を直線でつなぎ合わせる (点列優先)	124	1
GMARK1	シンボルの作画	124	1
GPOLY1	長方形の作画	125	1
GRAPP1	近似曲線の作画	121	1
GRFRAM	外枠の作画	108	1
GRGRID	リファレンス線の作画 (軸単位)	110	1
GRINIT	省略	88	1
GRPGN1	折れ線グラフの作画	94	1
GRPNT1	点グラフの作画	121	1
GRREF	リファレンス線の作画 (線単位)	110	1
GRSSIZ	作画シンボルのサイズ指定	95	1
GRTRGU	グラフ座標値からユーザ座標値へ変換	112	1
GRTRUG	ユーザ座標値からグラフ座標値へ変換	112	1
GRXAXS	X軸の作画	91	1
GRYAXS	Y軸の作画	94	1
GTXTC	文字列の作画	111	1
NEWPEN	オフラインプロットのペン指定	106	2
PAGE	PLOTと対で絵の1部にページ指定をする	132	2
PLOT	PAGEと対で絵の1部にページ指定をする	132	2
PLOTE	1枚の絵の作画の最後に置く	90	2
PLOTS	1枚の絵を作画する最初に置く	90	2
	1枚の絵のグラフィックデータセットにメンバ名を与える	133	
SYMBOL	文字列の作画 (小文字, ギリシャ文字等を含む)	115	2

参考文献

- 1 FACOM OSIV GRACE IV 解説書 (図形編) グラフィックアプリケーションパッケージ、富士通
- 2 FACOM OSIV PSP 文法書、富士通
- 3 FACOM OSIV GDP 使用手引書 (図形処理ユティリティプログラム)、富士通
- 4 FACOM OSIV/F4 PSL 解説書、富士通
- 5 車古正樹著、利用の手引き TSS 実習書 (全般用) TSS ユーザのための便利なコマンド (その1)、金沢大学計算機センター
- 6 車古正樹著、利用の手引き TSS 実習書 (全般用) TSS ユーザのための便利なコマンド (その2)、金沢大学計算機センター
- 7 車古正樹著、利用の手引き 図形出力利用法 (主としてGRACEについて)、金沢大学計算機センター (NLP 出力不能)
- 8 宮島昌克著、利用の手引き OFF-LINE XYプロッター デジタイザー編、金沢大学計算機センター
- 9 関崎正夫著、利用の手引き、金沢大学計算機センター
- 10 SAS/GRAPH USER'S GUIDE
- 11 車古正樹著、利用の手引き SAS (統計分析・図形処理) 使用方法、金沢大学計算機センター
- 12 山崎光悦著、利用の手引き グラフィック・ディスプレイによる会話型図形処理入門—GSP 使用方法を中心に—、金沢大学計算機センター
- 13 中野康英・西沢辰男・榎谷浩・松本樹典著、利用の手引き 高速再表示型グラフィック・ディスプレイによる会話型図形処理—IGL 紹介—、金沢大学計算機センター
- 14 車古正樹・沼田道代著、利用の手引き NLP 使用手引き、金沢大学計算機センター
- 15 関崎正夫著、利用の手引き TSS 入門—フルスクリーン機能のない端末について—、金沢大学計算機センター
- 16 FACOM OSIV/F4 MSP TSSコマンド文法書、富士通
- 17 車古正樹著、利用の手引き 日本語文章処理システム入門 —日本語エディタと入出力方法—、金沢大学計算機センター

## おわりに

この手引書では、主に GRACE IV を用いて計算機に図形を書かせる手順と、GDP コマンドを用いてグラフィックディスプレイ上に図形を表示させる方法について述べた。

計算機に図形を書かせるには、この他に、現在“利用の手引き”の発行されているものだけでも、GSP、SAS の2種があり、更に増加することが予想される。

各ユーザは1つの方法だけにとらわれず、それぞれの方法の特徴を良くつかんで手法を選択されることを望む。ちなみに GSP には、GDP の会話コマンドとは比較にならない会話機能がある。また、SAS は GRACE IV よりもはるかに簡単なプログラムで標準的なグラフを作ってくれるし、等高線や3次元の俯瞰図も書いてくれる。

また、GSP の手引書（参考文献12）でも扱っているように GSP と GRACE IV を組み合わせることは有効であるし、この手引書で扱ったように SAS と GRACE IV を組み合わせることもできる。このような組み合わせの形態も、各ユーザが各手法の特徴を良くつかんで工夫されることを望む。

なお、この手引書で述べたもう1つの内容である GDP コマンドによるグラフィック・ディスプレイ上での図形表示およびその操作は、SAS など他の手法を用いたものでもグラフィック・データセット（当センタで言う ONLINE.OUTLIST）上に書かれたものに対してならば、そのまま適用できる。