

会話型図形処理システムGRAPHMANの使用について

石氷, 結花
九州大学大型計算機センター研究開発部

武富, 敬
九州大学大型計算機センター研究開発部

平野, 広幸
九州大学大型計算機センター業務係

池田, 悟
九州大学大型計算機センター業務係

他

<https://doi.org/10.15017/1468114>

出版情報：九州大学大型計算機センター広報. 18 (2), pp.80-118, 1985-03-15. 九州大学大型計算機センター
バージョン：
権利関係：



会話型図形処理システム GRAPHMAN の使用について

石氷 結花* , 武富 敬* , 平野 広幸** , 池田 悟** , 菅崎 直弘** , 景川 耕宇*

目 次

1. はじめに	80
2. GRAPHMANの概要	81
3. 使用方法	87
3. 1 TSSによる使用	87
3. 2 バッチによる使用	89
4. PDB (Picture Data Base) とピクチャ	90
4. 1 PDBの作成と使用	90
4. 2 ピクチャの作成	92
4. 3 ピクチャの表示	96
4. 4 ピクチャの編集	97
5. マクロデータセット	101
6. グラフィック画面のNLP, XYプロッタへの出力	103

1. はじめに

センターの計算機利用の中で、図形処理も大きな比重を占めている。特に科学技術分野では、計算結果の数値データをグラフや図形の形に整理して提示することが多い。しかし、現状ではこれらには、PSP, HCBS, PLOT10などの基本図形処理ライブラリを用いて作図プログラムを作成するという厄介な作業がつきまとう。また、作図用の数値データも数や量が増えるとその管理も煩雑になりがちである。も

* 九州大学大型計算機センター 研究開発部

** 同 業務掛

っと手軽で、作図データのデータバンク機能を備えた、マンマシンインターフェイスのよい図形処理システムはないものか。

先に公開した会話型図形処理システムGRAPHMAN (GRAPHic MAN-machine system for scientific calculation results) [1, 2] は、これらの困難を軽減し、図形処理の効率向上をはかるのに有効な1つのシステムであると思われる。本稿では、このGRAPHMANの特徴、機能、使用方法の概要について述べる。

2. GRAPHMANの概要

GRAPHMANは、科学技術計算分野での数値データに対して、会話的に図形の作成・表示・編集を行うことのできる会話型図形処理システムである。***

はじめに、GRAPHMANの特徴を挙げる。GRAPHMANで使用するデータセットや用語については図2.1のシステム構成図を参照されたい。

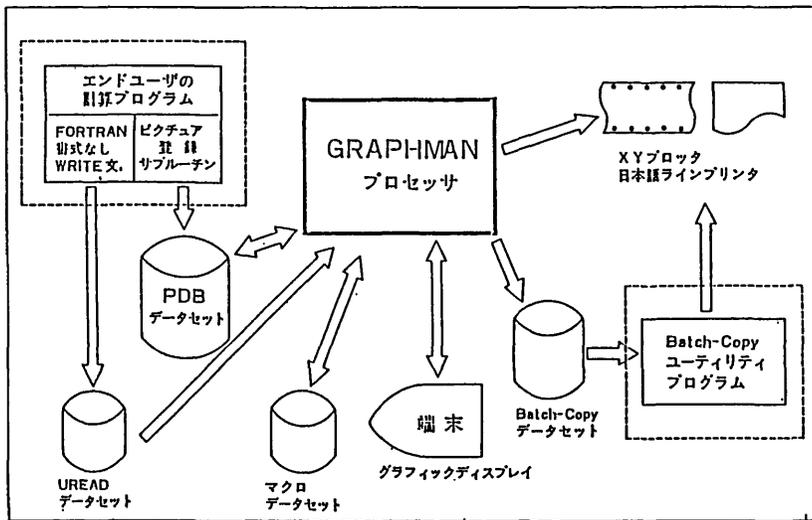


図 2.1 システム構成図

(1) 作図データのデータバンク機能がある。

GRAPHMANでの作図データの管理はPDB (Picture Data Base) データセットで行われる。これは、GRAPHMANの図形処理の元になるデータベースであり、利用者の計算結果データがGRAPHMANで取り扱われるためには、このPDBに登録されて'ピクチャ'

*** センターでは、2階端末室にある F9432A/T4105 の各グラフィック端末で使用できる。

と呼ばれる図形単位を形成しなければならない。PDBとピクチャの関係は図2. 2のようになる。

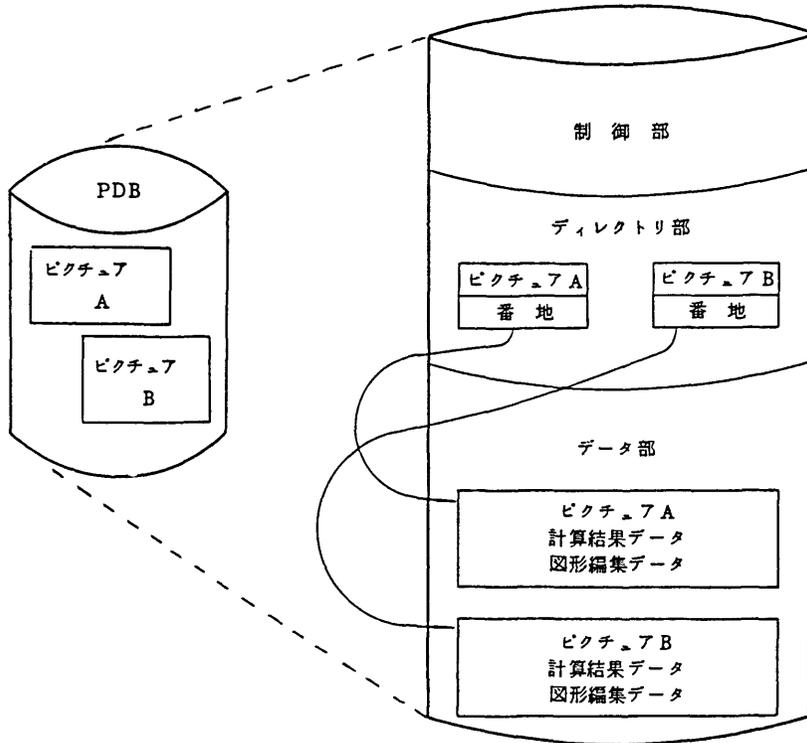
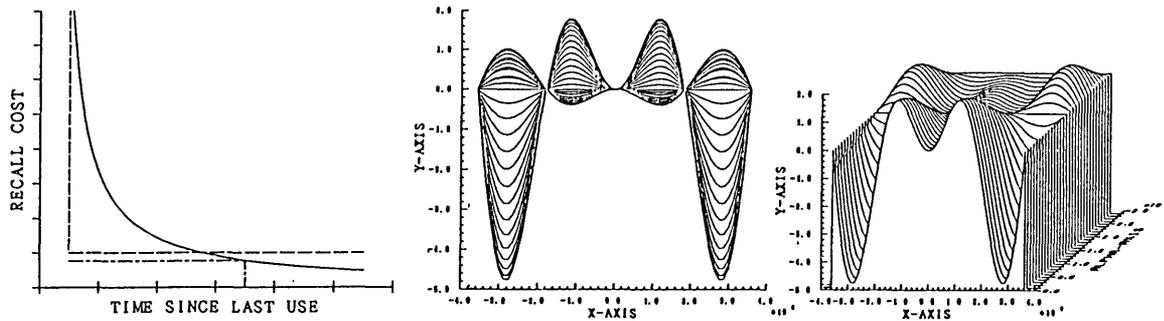


図2.2 PDBの論理構造

PDBはこのように区分データセットに似た独自の構造を持ちピクチャを管理する。1つ1つのピクチャはそれぞれ、利用者の入力した数値データと、それをもとに自動的に生成された作画のためのパラメタの集まりである図形編集データから成っている。現在利用者がPSPやHCBSのサブルーチンを使用して図形を描くときには、軸の作画や線分1つ1つに対してプログラムする必要があり、多くの時間と労力が必要とされる。GRAPHMANでは、これらの図形編集データ（線分の種類、センタシンボルの種類、スケールの単位、隠線処理の有無、軸の最大値・最小値など）に生成時に標準値が与えられているので、特にプログラムの必要はない。

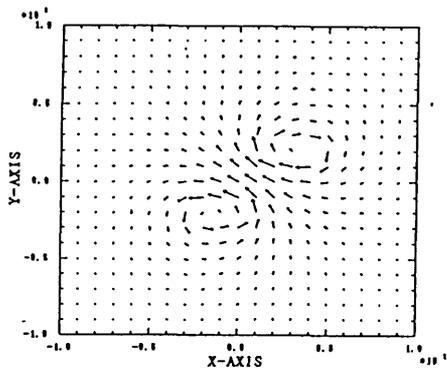
(2) 8つの図形パターンが作図プログラムなしで作図でき、図形の表示が容易に行える。

計算結果データは、一度GRAPHMANのPDBに登録してピクチャを作成しておくといくらでもすぐに取り出して表示することができる。また、図形全体と成分（軸、線分、コメントなど）とに分けて表示したり、図形の拡大・縮小も行うことができる。以下にGRAPHMANで対応できる8つの図形パターンと実際の実出力例を掲げる。

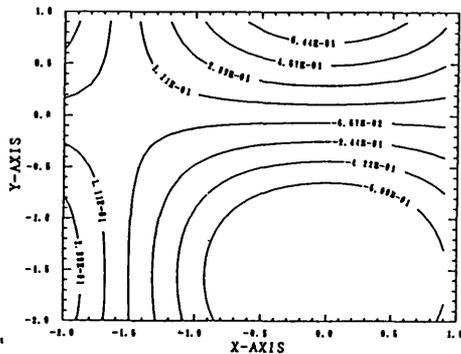


① 1次元図の出力例

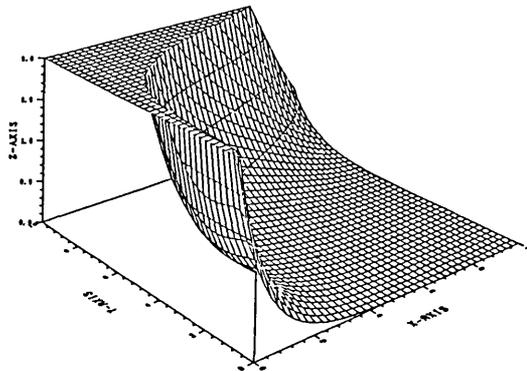
② 多曲線の3次元図の出力例



③ 2次元流れ図の出力例



④ 等高線図の出力例



⑤ 鳥瞰図の出力例

図 2.3 GRAPHMANにおける図形パターンの出力例

- ① 1次元図 -----> 図2. 3の例①
- ② 多曲線の3次元図(複数曲線の斜投図) -----> 図2. 3の例②
- ③ 2次元メッシュ図(2次元FEMデータの表示)
- ④ 2次元流れ図(各点に流れの方向・大きさを表示) -----> 図2. 3の例③
- ⑤ 等高線図($z = f(x, y) = \text{const.}$ の追跡図) -----> 図2. 3の例④
- ⑥ 鳥瞰図($z = f(x, y)$ の投視図) -----> 図2. 3の例⑤
- ⑦ 3次元メッシュ図(3次元FEMデータの表示)
- ⑧ ダミー図(任意の図形の表示)

(3) 図形のデバッグが容易に行える.

本来作図用のさまざまな編集データは計算結果に大きく依存しており、たった1つのデータの変更が何ヶ所もの作図プログラムの修正を招くことも多い。GRAPHMANでは前述のようにピクチャの持つ図形編集データはパラメタの集まりであるので、編集が容易に行える。編集のためには、図形操作EDITコマンドを用いる。編集対象としている図形のデータパターンにより、何種類かの図形操作EDITコマンドを使い分ける。こうして、パラメタの値を変更してはグラフィック端末で表示させながらデバッグを容易にしかも迅速に進めることができる。

(4) 数値データとのインターフェイスが簡単である.

計算結果データをPDBに登録してピクチャを作成するためには、利用者の計算プログラムの中でピクチャ登録サブルーチンを直接呼び出して登録する方法と、計算プログラムから一度UREADデータセットと呼ぶデータセットにデータをはき出しておいてGRAPHMANのUREADユーティリティで中に取り込む方法とがある。また、GRAPHMANは演算・制御コマンドを持つので簡単な計算はもとより、GRAPHMANの中でデータの作成や加工も行える。

(5) コメントが付けられる.

作図された図形に図2. 4に示す文字種、線種、センタシンボルなどを用いてコメントを付加できる。これらのコメントも図形操作EDITコマンドで容易に編集することができる。

(6) バッチでも動作可能である.

バッチ下でも動作が可能なので、大きなリージョンや長い処理時間を必要とするTSS向きでない処理はこちらで行うとよい。

Example of Comments

1. CHARACTER

```
(1) 1234567890
(2) ABCDEFGHIJKLMNOPQRSTUVWXYZ
(3) abcdefghijklmnopqrstuvwxyz
(4) αβθδεφγηι κλμνοπ ρστυφωξζ
(5) ΑΒΘΔΕΦΓΗΙ ΚΛΜΝΟΠ ΡΣΤΤΨΩΞΖ
```

2. CENTER SYMBOL

```
(1) ○ (5) ▲ (9) ◇
(2) △ (6) □ (10) ◆
(3) □ (7) + (11) .
(4) ⊙ (8) ×
```

3. KIND OF LINE

```
(1) —— (11) ——
(2) ---- (12) ----
(3) - - - - (13) - - - -
(4) ——— (14) ———
(5) - - - - (15) - - - -
```

4. ARROWS

5. BOX

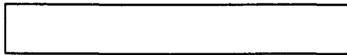


図 2.4 コメントに使用できる文字種，線種，センタシンボルなど

(7) マクロが使用できる。

GRAPHMANではマクロデータセットを持つことができるので、定型的な処理に備えてコマンドをマクロ化して使用することができる。特に、GRAPHMAN起動中に使用したコマンドのロギングをマクロとして編集することもできるのは有用である。

(8) NLP, XYプロッタに図形を出力することができる。

満足のいく図形が得られたら、BCOPYユーティリティ機能でNLPやXYプロッタに出力することができる。

次に各論にはいる前に、GRAPHMANの使用に必要な予備知識にふれておく。GRAPHMANでは、次の3つの座標系を用いることを知っておく必要がある。

- ・GD (Graphic Device) 座標系
- ・FRAME座標系
- ・AXIS座標系

GD座標系は、ピクチャとは無関係な絶対空間で、ディスプレイ装置に依存しない絶対座標系を定める。すなわち装置によらずに縦と横を100×100の無次元単位の領域として定義するものと考えてよい。GRAPHMANを起動した当初は、グラフィックディスプレイの全面がGD座標系として定義されているが、DEVPARAコマンドで任意に作図領域を変更できる。このDEVPARAコマンドはバッチ処理で用いると、NLP上の作図領域をA4、B5サイズなどに自由に変更できて利用価値が高い。ピクチャを作図領域(GD空間)の中のどの位置に、どの位の大きさで描くかを指定するには、FRAME座標系を設定する。FRAME座標系は、それぞれのピクチャごとにGD座標系の単位でORIGIN(原点)とFRAME(枠の大きさ)を指定すれば決定でき、やはり100×100の無次元単位の領域を定義する。ORIGINとFRAMEの指定は、図形操作コマンドのうちピクチャごとの座標系の情報を編集するEDITコマンドで行う。さらに、実際のデータ値に即した単位で示されるものがAXIS座標系であり、3者の関係は図2.5の様になる。この図の場合、ディスプレイ全体をGD空間とすると、このFRAME空間は、EDITコマンドのORIGINサブコマンドとFRAMEサブコマンドで次のように指定したことになる。(> はGRAPHMANのコマンドモードのプロンプティングであり、>> はサブコマンドモードのプロンプティングである。)

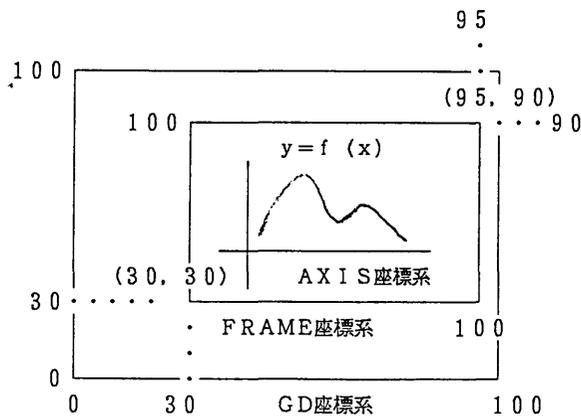


図2.5 GRAPHMANの3つの座標系の関係

```
>EDIT
>>ORIGIN X(30) Y(30) -----> 原点をGD座標系の値で設定する
>>FRAME X(95) Y(90) -----> 枠の大きさをGD座標系の値で設定する
>>
```

次に、GRAPHMANで会話的に処理を進めて行くうえで利用価値が高いグラフィックカーソルについて述べる。現在センターでは富士通の F9432A 3台とテクトロニクス社の T4105 2台でグラ

グラフィックカーソルを使用できる。グラフィックカーソルを用いると前述の3つの座標系を意識せずに、直接図形やタイトル、矢印などを描く位置を決めることができる。コマンドあるいはサブコマンドのオペランドに*を持つものは、こうしてグラフィックカーソルを使用して座標値の入力を行うことができる。以下にグラフィックカーソルの使い方の例をそれぞれの座標系の単位での位置を知るためのPICKINGコマンドを用いて示す。

>PICKING * X Y GD -----> ある点の座標をグラフィックカーソルで与えて
そのGD座標系での値を変数X, Yに入れる

グラフィックカーソルが現われるので、目的の点へ動かしてやり、F9432Aであればカーソルの先のボタンを押し、T4105であれば任意のキーを押し。

>PRINT X Y -----> 座標値(X,Y)を表示する
X = 75. 2137
Y = 26. 5083
>

3. 使用方法

3.1 TSSによる使用

GRAPHMANは、コマンドGRAPHMANで起動され、ENDGで終了する。GRAPHMANの中では2つのモードがあり、コマンドモードでは > , サブコマンドモードでは >> のプロンプティングがあらわれる。GRAPHMANの中から先頭に % を付けることでTSSコマンドを呼び出すことができる。

コマンド	オペランド
GRAPHMAN	[NS (80 変数領域の大きさ)]

1) オペランドの説明

NS (80 |変数領域の大きさ) : 変数領域 (Named Storage) の大きさを指定する。
省略値は80K。

2) 使用例

- ① 新規にPDBをつくり、データを入力してピクチャを登録する。

READY

F9430 F9432A GRAPH ---> F9432Aで使用する場合はこのコマンドが前もって必要
READY

GRAPHMAN

--- FOLLOWING DEVICES ARE SUPPORTED NOW. ---

- 1 NON GRAPHIC DISPLAY (CHARACTER DISPLAY)
- 2 F9431, F9432, F9433, F9434
- 3 T4006
- 4 T4010, T4012, T4013
- 5 T4014, T4015, T4016
- 6 NLP (F6715D, ETC)
- 7 XY-PLOTTER (F6201D, F6202B)
- 8 F9430 & SCOPY (F9436PL2)
- 9 NLP (A4 SIZE)

PLEASE SELECT DEVICE NUMBER -----> 2 --> 装置番号の選択
PLEASE WAIT A MINUTE. INITIALIZATION IN PROCEED
ING NOW

>CREATE PDB(GRAPHMAN.DATA) REC(5 5) --> PDBの創設(4.1参照)

>X=:0 1 2 3 4 5 6 7 8 9 -----> データの直接入力(4.2参照)

...X(10) IS ALLOCATED.

>Y=:4 6 2 12 11 6 9 15 10 11

...Y(10) IS ALLOCATED.

>NEWPIC EXAM 'LINE GRAPH'-----> 新しいピクチュアEXAMの登録開始

>>ONEDIM 10 X Y -----> ここでは配列X, Yに値をもつ1本の曲線を登録

>>PICEND -----> ピクチュアデータ入力終わり

>>END S -----> ピクチュア登録終了

>ENDG -----> GRAPHMAN終了

READY

② すでに作成している PDB のピクチャを使って作図する。

```

略
>
>PDB GRAPHMAN.DATA -----> 使用するPDBの指定
>USE EXAM -----> 使用するピクチャの指定
>DISPLAY -----> 図形の作図
>ERASE -----> 画面の消去
>ENDG
READY
    
```

3.2 バッチによる使用

プロシジャ名	記号パラメタ
GRAPHMAN	[, NS = { <u>80</u> 変数領域の大きさ }] [, SYSOUT = { K S <u>O</u> U Q }]

1) パラメタの説明

NS (80 | 変数領域の大きさ) : 変数領域 (Named Storage) の大きさを指定する。

省略値は 80K.

SYSOUT = { K | S | O | U | Q } : 結果の出力先を指定する。省略値は O。Q は XY プロッタ出力。

2) 関連する DD 名

なし

3) 使用例

① 新規に PDB をつくり、データを入力してピクチャを登録する。

```

// EXEC GRAPHMAN
//SYSIN DD *
    
```

6 -----> はじめに必ず出力装置を選択する。NLP (=6) か XY プロッタ (=7) のみ

```

CREATE PDB('F9999.GRAPHMAN.DATA') REC(5 5)
X=:0 1 2 3 4 5 6 7 8 9
Y=:4 6 2 12 11 6 9 15 10 11
NEWPIC EXAM 'LINE GRAPH'
ONEDIM 10 X Y
PICEND
END S
ENDG
/*
//

```

- ② すでに作成しているPDBのピクチャを使って作図する。

```

// EXEC GRAPHMAN
//SYSIN DD *
6
PDB 'F9999.GRAPHMAN.DATA'
USE EXAM
DISPLAY
ENDG
/*
//

```

4. PDB(Picture Data Base)とピクチャ

4.1 PDBの作成と使用

GRAPHMANで取り扱うためには、すべての図形はPDB (Picture Data Base) と呼ばれるデータセットに登録されている必要がある。PDBはCREATEコマンドで作成する。

コマンド	オペランド
CREATE	PDBデータセット名 REC (初期レコード数 [増分レコード数])

・オペランドの説明

PDBデータセット名：新規に作成するPDBデータセット名を指定する。省略不可。

REC (初期レコード数 [増分レコード数])：PDBのレコード数を指定する。1レコードは
6144バイト。初期レコード数は省略不可。

CREATEコマンド使用時には、特にデータセットの割り当ては必要なく、上記のようにPDBデータセット名とレコード数を与えればよい。また、PDBは順データセット形式でなければならないので、PDBデータセット名として PDB. DATA (XXX) などのメンバ名を持つ形式で入力するとエラーとなる。すでに作成済みであれば次回からはPDBコマンドで使用を宣言する。

コマンド	オペランド
PDB	{PDBデータセット名 *}

・オペランドの説明

PDBデータセット名|*：使用するPDBデータセット名を指定する。*を指定すると、当GRAPHMANセッション中だけで有効なWORK-PDBが作成されGRAPHMAN終了時に消滅する。

PDBは何個でも持てるので、GRAPHMANの中で複数個のPDBを切り換えて使用する時にもPDBコマンドを投入する。その他PDBを操作するためのコマンドを表4.1に示す。これらのコマンドによって使用状況を知ったり、複数のPDBの間でピクチャのやりとりなども行うことができる。

表 4.1 その他の PDB 操作コマンド

コ マ ン ド 名	機 能
SPACE	PDBの使用状況情報の出力
LISTC	ピクチャのディレクトリ情報の出力
CONDENSE	PDBの圧縮
PDBCOPY	PDBのコピー
DELPIC	ピクチャの削除
RENAME	ピクチャ名の変更
TRANSFER	ピクチャのコピー

4.2 ピクチャの作成

利用者の計算結果データはPDBに登録される際に、図形編集データが付加されてピクチャと呼ばれる図形単位となる。GRAPHMANでは図形の表示・消去・編集などは、すべてこのピクチャを対象とする。ピクチャの登録には、大きく分けて2つの方法がある。

①GRAPHMANの変数領域 (Named Storage) 上にデータを持ってきた後、NEW PICコマンドでPDBに新ピクチャとして登録する方法。

②ピクチャ登録サブルーチンを使用して、計算プログラムの中で一気にピクチャ登録をする方法。まず①の方法について述べる。GRAPHMANの変数領域 (Named Storage) 上にデータを持ってくるにはまた2つの方法がある。

(1) GRAPHMANの演算・制御コマンドを用いて確保する方法

GRAPHMANには、算術代入や繰り返し・条件分岐などが可能な演算・制御コマンドが用意されている。また、20種類の基本関数とPIとRADの2つの自動設定システム定数が使用できるので、これらによりGRAPHMANの中でじかに計算結果データそのものを作成できる。ただし、GRAPHMANでは整数と単精度の実数しか取り扱えないうえに変数の型宣言コマンドは用意されていないので、変数命名の際には次のことに留意されたい。

- ・ I, J, K, L, M, Nで始まる変数・・・整数
- ・ その他の英文字で始まる変数・・・実数

表 4. 2 に演算・制御コマンドの一覧を示す。

表 4.2 演算・制御コマンド

コマンド名	機能
ARRAY	配列変数を確保する
FREE	変数領域を解放する
CLEAR	すべての変数領域を解放する
PRINT	変数の値を出力する
WHILE~WEND	繰り返しの制御を行う
IF~ELSE~ENDIF	条件分岐の制御を行う
DO~DOEND	繰り返しの制御を行う
CANCEL	制御コマンドの入れ子を途中で解除する
算術代入コマンド	算術式の値を変数に代入する
配列代入コマンド	配列の各要素に値を代入する

(2) UREADユーティリティを使用する方法

この方法は、あらかじめFORTRANの書式なしWRITE文を使ってデータをデータセットにはき出しておいて、そのデータセットからGRAPHMANのUREADコマンドを用いてGRAPHMANの変数領域上に読みこむ方法である。使用例を以下に示す。

・結果をUREAD.DATAに出力するFortranプログラムの例

```

DIMENSION X(61) Y(61) T(30)
RAD=3.14159/180.0
OPEN(UNIT=1,FILE='F9999.UREAD.DATA',STATUS='OLD')
DO 1 I=1,61
  X(I)=-360+(I-1)*12
1 CONTINUE
DO 3 J=1,30
  T(J)=J*12
  DO 2 K=1,61
    Y(K)=X(K)*SIN(X(K))*RAD)/T(J)*SIN(T(J)*RAD)
2 CONTINUE
WRITE(1) (X(I),I=1,61),(Y(K),K=1,61)
3 CONTINUE
CLOSE(1)
STOP
END

```

・このプログラムを実行した後、GRAPHMANの中で次のように行くとピクチャ登録が完了す

る。

```

>UOPEN UREAD.DATA -----> データセットのオープン
>NEWPIC P1 'TEST' -----> ピクチャP1の登録開始
>>DO I=1,30 -----> 30本の曲線を登録するためのループの設定
>>FREE X Y -----> 使用する変数領域の解放
>>ARRAY X(61) Y(61) -----> 変数領域の配列の確保
>>UREAD (X(J),J=1,61),(Y(K),K=1,61)
                -----> GRAPHMAN変数領域上へのデータの読み込み
>>ONEDIM 61 X Y -----> 配列X, Yに値をもつ1本の曲線を登録
>>DOEND
>>PICEND -----> ピクチャデータ入力終了
>>END S -----> ピクチャ登録終了
    
```

これら(1)か(2)のいずれかの方法でGRAPHMANの変数領域にデータが作成できたらNEWPICコマンドでPDBにピクチャを登録する。概要でも述べたとおり、このNEWPICコマンド入力後のサブコマンドモードで表4.3のいずれかのデータパターンを選択する。こうすると、変数領域上の計算結果データから作図のための種々のパラメタの集まりである図形編集データが作成され、両者でピクチャとなってPDBに格納される。なお、データパターン登録のための8種のサブコマンド(表4.3)は、目的とする図形編集データ作成のために、オペランドとなるデータ配列に対してそれぞれ細かい条件を定めているので、マニュアル[2]を参照されたい。以下にNEWPICコマンドの入力形式を示す。

コマンド	オペランド
NEWPIC	ピクチャ名 ['文字列']

・オペランドの説明

ピクチャ名：ピクチャ名を指定する。ピクチャ名は、英数字で始まる8文字以内の文字列と括弧で囲んだ整数(1~99999)から構成される。(例：PIC(100))
この整数部分は、添字がわりに繰り返し制御コマンドと共に使用して、複数の図

表 4.3 NEWPIC コマンドのサブコマンド一覧

サブコマンド名	機能
ONEDIM	データパターンONEDIMの計算結果データを登録する
MESH2D	データパターンMESH2Dの計算結果データを登録する
FLOW2N	データパターンFLOW2Nの計算結果データを登録する
FLOW2R	データパターンFLOW2Rの計算結果データを登録する
CONT2N	データパターンCONT2Nの計算結果データを登録する
CONT2C	データパターンCONT2Cの計算結果データを登録する
CONT2R	データパターンCONT2Rの計算結果データを登録する
MESH3D	データパターンMESH3Dの計算結果データを登録する
BEGIN	今までに登録されたデータを取り消し、初期状態に戻す
PICEND	データの登録が終了したことを宣言する
END	NEWPICサブコマンドモードを終了する

形に同じ処理をしたい時などに有効である。

'文字列' : ピクチャに対するコメントを72字以内の文字列で与えることができる。作成し

ておくと、LISTD、LISTCコマンドで参照する時にわかりやすい。

次に②について述べる。この方法では、GRAPHMANコマンドと同じ機能を持つサブルーチン（ライブラリ名'APP1.PICLIB'のピクチャ登録サブルーチン、付録参照）をFortranの計算プログラムで直接呼び出して一気にPDBにピクチャを登録する。ピクチャ登録サブルーチンを使用したプログラムの例を以下に掲げる。このプログラムは、前述のUREADユーティリティのプログラム例と同じ計算を行っているものなので、NEWPICコマンドによる登録方法と比較するとパラメタの渡し方などがほとんど同じことがわかる。

```

DIMENSION X(61) Y(61) T(30)
RAD=3.14159/180.0
CALL OPNPDB('F9999.GRAPH.DATA&',1)
CALL NEWPIC('P1',1,'TEST&')
DO 1 I=1,61
  X(I)=-360+(I-1)*12
1 CONTINUE
DO 3 J=1,30
  T(J)=J*12
  DO 2 K=1,61
    Y(K)=X(K)*SIN(X(K))*RAD/T(J)*SIN(T(J))*RAD)
2 CONTINUE
CALL ONEDIM(61,X,Y,'&')
3 CONTINUE
CALL PICEND
CALL CLSPDB
STOP
END

```

上記のプログラムを実行するときは次のようにライブラリを指定する。

```
RUN LIB('APP1.PICLIB')
```

4.3 ピクチャの表示

PDBへのピクチャの登録が終了したら表示させてみて、必要であれば編集を行う。そのためにはまず、USEコマンドで処理したい1つのピクチャを宣言する。

コマンド	オペランド
USE	ピクチャ名

・オペランドの説明

ピクチャ名：表示や編集の対象となるピクチャ名を指定する。

USEコマンドによる使用宣言は、再びUSEコマンドでピクチャを切り換えるか、PDBを切り換えるか、NEWPIC/OLDPICコマンドでピクチャの登録なり編集を行うまで有効である。図形を表示するためには主にDISPLAYコマンドを使用し、図形を消去するためには主にERASEコマンドを用いる。表示に関しては、図形を拡大・縮小して表示するZOOMINGコマンドや、サブコマンドモードで1つのコメントや1つの曲線ごとに表示するGOサブコマンドなどがある。特にDISPLAYコマンドでは、図形の枠、軸、コメント、折線データ、ピクチャ本体のいずれかだけ表示したりしなかったりやオペラン

ドの選択で行えるので、一部分だけ変更してまず確認した後で全体を表示させるということができる。

コマンド	オペランド
DISPLAY	[{ <u>BIRD</u> <u>NOBIRD</u> }] [{ <u>FRAME</u> <u>NOFRAME</u> }] [{ <u>AXIS</u> <u>NOAXIS</u> }] [{ <u>LINE</u> <u>NOLINE</u> }] [{ <u>COMMENT</u> <u>NOCOMMENT</u> }] [{ <u>PIC</u> <u>NOPIC</u> }]

・オペランドの説明

{BIRD | NOBIRD} : 3次元表示を行う時はBIRDと指定する。省略時は3次元表示しない。

{FRAME | NOFRAME} : 枠を作図しない時はNOFRAMEと指定する。省略時は枠を作図する。

{AXIS | NOAXIS} : 軸を作図しない時はNOAXISと指定する。省略時は軸を作図する。

{LINE | NOLINE} : 折線データを作図しない時はNOLINEと指定する。省略時は折線データを作図する。

{COMMENT | NOCOMMENT} : コメントを作図しない時はNOCOMMENTと指定する。省略時はコメントを作図する。

{PIC | NOPIC} : ピクチャ本体を作図しない時はNOPICと指定する。省略時はピクチャ本体を作図する。

4.4 ピクチャの編集

PDBにピクチャとして登録されるときに作図用パラメタの集まりである図形編集データが標準値を持って生成されることはすでに述べた。この図形編集データは表4.4に示すように8種類の情報に分けることができる。ピクチャを編集するということは、これらの図形編集データを対応する図形操作EDITコマンドで修正することを言う。ただし、すべてのピクチャが8種全部の作図情報を持っているのではない。図形編集データは、データパターンに依存しているので、それぞれのピクチャの属するデータ

表 4.4 図形編集データと図形操作 EDIT コマンド

図形編集データ	説 明	図形操作 EDIT コマンド
COORDSYS	FRAME座標系, AXIS座標系を設定するためのパラメタ	EDIT
COMMENT	コメントの大きさ, 位置, 文字列などコメントを作画するためのパラメタ	EDIT COMMENT
LINE	任意の折線データの座標値, 線種類などのパラメタ	EDIT LINE
CURVE	1次元図における曲線に関するパラメタ	EDIT ONEDIM
MESH	2次元メッシュ図を作画するためのパラメタ	EDIT MESH2D
VECTOR	2次元流れ図を作画するためのパラメタ	EDIT FLOW2D
CONTOUR	等高線図を作画するためのパラメタ	EDIT CONTOUR
PERSPECT	透視図(多曲線の3次元図, 鳥瞰図, 3次元メッシュ構成図)を作画するためのパラメタ	EDIT PERSPECT

パターンごとに必要とされる情報しか持たない。そこで、ピクチャの編集のためには、表 4. 5を参照して、

- ①自分の編集しようとしている図形はどのデータパターンであるか
- ②そのデータパターンはどのような図形編集データを持つか
- ③その図形編集データを変更するにはどの図形操作 EDIT コマンドを使用するのか

を判断することから始める。それぞれの図形操作 EDIT コマンドはサブコマンドモードを持ち、各パラメタの値の変更はサブコマンドで行う。すべての図形操作 EDIT コマンドには LIST サブコマンドが用意されており、これを使用すると図 4. 1のように編集可能なパラメタの一覧とそれらが現在持つ値(登録直後であれば、自動的に生成された標準値)が、使用するサブコマンドと対で実際の入力形式そのままに出力されるので、参考にするとよい。各パラメタの意味や値の単位などについては、実際の使用時にマニュアル [2] を参照されたい。サブコマンドを使用してパラメタの値を変更したら作図させてみる。付録のコマンド一覧にあるように、DISPLAY コマンドあるいは ERASE コマンドなどは、サブコマンドモードでも使用できるので、編集を終了しなくても即時に表示させて確認できる。満足のいくものが得られたら、END サブコマンドで編集を終了する際に SAVE オペランドを指定して、PDB 内の図形編集データを更新し、納得がいかなければ、SAVE 指定をせずに終了して更新をやめることができる。図 4. 2と図 4. 3に、1次元図(データパターン ONEDIM)と鳥瞰図(データパターン CONT2N)について、ピクチャの編集例を変更履歴とともに示す。

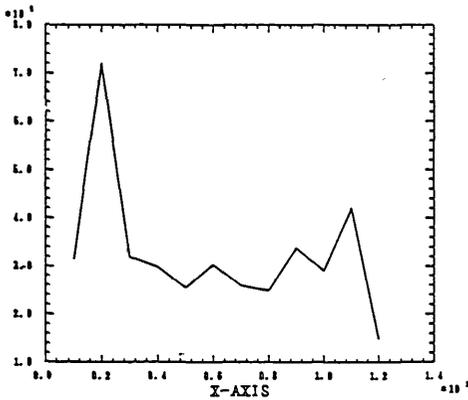
表 4.5 データパターンと図形編集データと図形操作 EDIT コマンド

データパターン	図形編集データ	図形操作 EDIT コマンド
ONEDIM	COORDSYS COMMENT LINE CURVE PERSPECT	EDIT EDIT COMMENT EDIT LINE EDIT ONEDIM EDIT PERSPECT
MESH2D	COORDSYS COMMENT LINE MESH	EDIT EDIT COMMENT EDIT LINE EDIT MESH2D
FLOW2N	COORDSYS COMMENT LINE VECTOR	EDIT EDIT COMMENT EDIT LINE EDIT FLOW2D
FLOW2R	COORDSYS COMMENT LINE VECTOR	EDIT EDIT COMMENT EDIT LINE EDIT FLOW2D
CONT2N	COORDSYS COMMENT LINE CONTOUR PERSPECT	EDIT EDIT COMMENT EDIT LINE EDIT CONTOUR EDIT PERSPECT
CONT2C	COORDSYS COMMENT LINE CONTOUR PERSPECT	EDIT EDIT COMMENT EDIT LINE EDIT CONTOUR EDIT PERSPECT
CONT2R	COORDSYS COMMENT LINE CONTOUR PERSPECT	EDIT EDIT COMMENT EDIT LINE EDIT CONTOUR EDIT PERSPECT
MESH3D	COORDSYS COMMENT LINE PERSPECT	EDIT EDIT COMMENT EDIT LINE EDIT PERSPECT
DUMMY	COORDSYS COMMENT LINE	EDIT EDIT COMMENT EDIT LINE

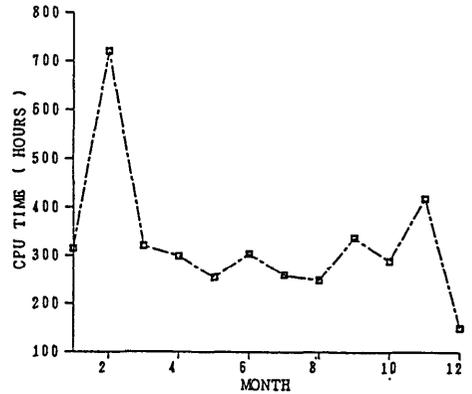
```

>
>EDIT -----> FRAME, AXIS座標系に関する編集の開始
>>LIST -----> サブコマンドと編集可能なパラメタの一覧を表示する
ORI X (45. 0000) Y (10. 0000)
FRA XW (45. 0000) YW (45. 0000) D (1) R (0) OP (1001)
X Y (0. 0) MIN (2. 70000) MAX (13. 0000) SC (LINE) +
  SP (1) MX (0) SD (3) S (0. 0) F (F) D (1) UP (0) +
  INT (0) ST (4. 00000) T (' TIME SINCE LAST USE' )
Y X (0. 0) MIN (0. 895238) MAX (2. 50000) SC (LINE) +
  SP (1) MY (0) SD (4) S (0. 0) F (F) D (1) RI (0) +
  INT (0) ST (4. 00000) T (' RECALL COST' )
Z MIN (0. 0) MAX (2. 00000) SC (LINE) S (2. 00000) +
  F (G) D (1) INT (0) ST (3. 00000) T (' ALPHA-AXIS' )
>>
  
```

図 4.1 LISTサブコマンドの使用例



あるマクロを実行して得られた1次元図



編集後に得られた1次元図

↓

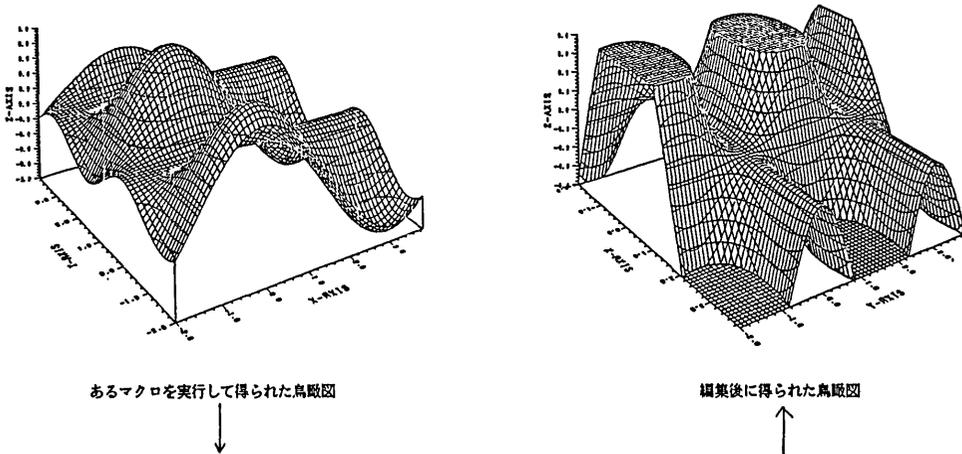
以下のコマンド、サブコマンドで編集する。

```

>ERASE -----> 図形の消去
>EDIT -----> ピクチャごとの座標系の情報を編集する
>>X SD(4) SP(0) F(I) UP(0) S(3) T('MONTH')
  *X軸に関して次の情報を変更する
  目盛りの描き方、自動スケールによる余白の有無、目盛り数値の表示単位、上側の目盛りの
  有無、目盛り数値表示の大きさ、X軸タイトル
>>Y SD(4) SP(0) F(I) RI(0) S(3) T('CPU TIME( HOURS )') +
>>MIN(100) MAX(800)
  *Y軸に関して次の情報を変更する
  目盛りの描き方、自動スケールによる余白の有無、目盛り数値の表示単位、右側の目盛りの
  有無、目盛り数値表示の大きさ、Y軸タイトル、最大値、最小値
>>FRAME OP(1001)
  *FRAME空間の枠作図に関して次の情報を変更する
  上と右の枠を描かない指定 (X, YサブコマンドのUP(0), RI(0)と同時に指定しない
  と効果はない)
>>END S -----> 図形編集データを置き換える
>EDIT ONEDIM -----> データパターンONEDIMの情報を編集する
>>LIST -----> 現在持っている曲線作図に関する値の表示
CRV 1 LK(1) C(7) SPL(0) SYM(0) INT(0) +
  S(2. 00000) AX(L1) T(' ')
>>CRV 1 LK(14) SYM(6)
  *線種を変更し、センタシンボルを描く
>>DISPLAY -----> 図形の表示
>>END S -----> 図形編集データを置き換える
  
```

↑

図 4.2 1次元図(データパターン ONEDIM)のピクチャ編集例



以下のコマンド/サブコマンドで編集する。

```

>ERASE -----> 図形の消去
>EDIT -----> ピクチャごとの座標系の情報を編集する
>>Z MIN(-0.4) MAX(0.4)
    *Z軸に関して、最大値、最小値を変更する
>>END S -----> 図形編集データを置き換える
>EDIT PERSPECT -----> 3次元表示に関する情報を編集する
>>LIST -----> 現在の値の表示
VIE A (30. 0000) B (0. 0) G (-40. 0000) +
    L (400. 000) D (350. 000)
OBJ XW (200. 000) YW (200. 000) ZW (150. 000) R (0)
INT MX (0) MY (0)
HID 1
CON 0
>>VIE A(-60)
    *視点の位置を変更する
>>END S -----> 図形編集データを置き換える
>DISPLAY -----> 図形の表示
    
```

図 4.3 鳥瞰図 (データパターン CONT2N) のピクチャ編集例

5. マクロデータセット

定型的な処理に備えて、GRAPHMANではコマンドをマクロ化して使用することができる。マクロデータセットはレコード長80バイトの区分データセットで、マクロはそのメンバとして、TSSのEDITコマンドか、GRAPHMANのEDIT MACROコマンドで作成・編集する。GRAPHMANのマクロではパラメタが使用できるので、配列の大きさや曲線の数などを実行時にそのつど変えて何通りにも処理することができ、利用価値が高い。マクロを作成するに当たっては、次のことに留意する必要がある。

①マクロは先頭行にMACRO文、最終行にMEND文を置く。

②MACRO文には、マクロ名と仮パラメタの並びを置く。この仮パラメタは英字で始まる8文字以内の英数字で、最大20個まで指定できる。

③仮パラメタはマクロ内 (MACRO文以外) では先頭に&をつけて表わす。これが、マクロ実行時に実パラメタに置き換わる。

④GRAPHMAN制御コマンドのDO、IF、WHILEなどの入れ子はマクロの中で完結していなければならない。

以上の制約を守って、マクロを作成し実行してみる。

```
>%ATTR #GG RECFM(F B) LRECL(80) BLK(3120) DS(PO)
>%ALLOC DA(GMACRO.DATA) SP(30 10) DI(10) NE CA T -
    USING(#GG)
```

----> 新規の場合、マクロデータセットはあらかじめ確保しておく

>DMACRO GMACRO.DATA -----> マクロデータセットの宣言

>EDIT MACRO M1 NEW -----> メンバM1を新規に作成する

0010	MACRO M1 N	---->マクロは必ずMACRO文から始まる
0020	PDB PDB. DATA	
0030	ARRAY X (&N) Y (&N) Z (&N &N)	
0040	DO I=1, &N	
0050	X (I) = -2 + (I - 1) / 10	
0060	Y (I) = X (I)	
0070	DOEND	
0080	DO .J=1, &N	
0090	CY=COS (Y (J))	
0100	DO K=1, &N	
0110	Z (K, J) =SIN (X (K)) *CY	
0120	DOEND	
0130	DOEND	
0140	MEND	---->マクロは必ずMEND文で終わる

>>END S -----> マクロのEDIT終了

>#M1 41 -----> マクロ実行時には、'#マクロ名 パラメタ'の形式で入力する

さらに、GRAPHMANでは起動後に使用したGRAPHMANコマンドをロギングしているので、このコマンド履歴をマクロとして編集することができる。その例を以下に示す。この場合、マクロデータセットは既存で使用宣言したものとする。

>

>EDIT MACRO M2 SYS ---> オペランドSYSはロギングしているコマンド履歴を指定
 することを意味する。これをメンバM2としてマクロ化する

>>LIST -----> 内容の表示

```

0010
.
.
.
.
0090 EDIT MACRO M2 SYS -----> 最後に入力したコマンド
    
```

先立って投入した
GRAPHMAN
コマンド

このコマンド列を前記の規則に従って編集すれば、マクロとして使用することができる。

6. グラフィック画面のNLP, XYプロッタへの出力

GRAPHMANではBCOPYユーティリティという画面のハードコピー機能があり、グラフィック端末に表示した図形をNLPやXYプロッタへ出力させることができる。そのためには、あらかじめ図形の出力先データセット（BCOPYデータセットという）を確保しておき、そこにいったんはき出しておいで、GRAPHMAN終了後にこのBCOPYデータセットからNLPやXYプロッタへGOUTコマンドで出力させるという方法をとる。まず、グラフィック端末に表示した図形をデータセットに出力するためには、次のような処理が必要である。

- ①出力先の宣言・・・DBCOPYコマンドで図形の出力先となる順データセット（BCOPYデータセット）を指定する。
- ②図形の出力・・・BCOPYコマンドを使用してグラフィック画面に描かれている図形ベクトルデータをDBCOPYコマンドで指定したデータセットに出力する。オペランドの指定で、図形の大きさを変えたり、一部をグラフィックカーソルで切り出したり、NLP上に横向きで出力したりできる。

なお、DBCOPYコマンドの投入時期はBCOPYコマンドで図形を出力する以前であれば図形が表示されているいないに関わらずいつでもよい。これらのコマンドの入力形式は以下のようである。

コマンド	オペランド
DBCOPY	データセット名

・オペランドの説明

データセット名：図形の出力先となる順データセットを指定する。すでに存在すればそのデータセット名を指定する。なければ当コマンドの投入以前に前もって確保しておく（後述の使用例参照）。

コマンド	オペランド
BCOPY	[{ * X1 (r 1) Y1 (r 2) X3 (r 3) Y3 (r 4) }] [XW (r)] [FRAME (i)]

・オペランドの説明

*：出力する領域をグラフィックカーソルで指定する。カーソルは2度現われるので、領域の対角の頂点を入力する。

X1 (r 1) Y1 (r 2) X3 (r 3) Y3 (r 4)：出力する領域をGD座標系の単位で指定する。省略時はGD空間全体が対象となる。

XW (r)：NLPに出力する際の大きさをmm単位で与える。省略すればGRAPHMANが適度にスケールした大きさとなる。また、XW(負の値)とするとNLP上で90度回転した形で出力される。

FRAME (i i)：NLPの作図領域全体(10の位)と図形を出力する領域(1の位)の双方の枠の作図に関する指定を1か0かで行う。省略時はFRAME(00)となり、どちらの枠も作図しない。

GRAPHMANの中でBCOPYコマンドは何回でも使用でき、1回ごとにNLP1枚の図形が得られる。ただし、BCOPYコマンドを実行しても図形データはDBCOPYで宣言したデータセットに出力されるだけであることに注意されたい。このデータセットから図形を出力するには、GRAPHMANセッションを

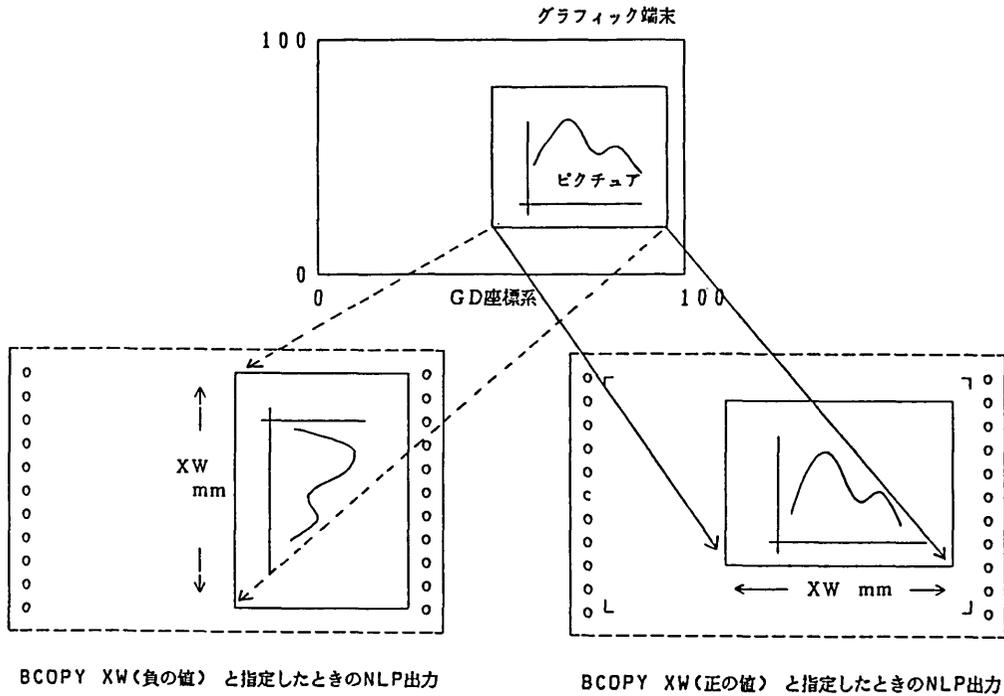


図 6.1 グラフィック画面とNLPへの出力の対応

終了した後に、GOUTコマンドを実行する。GOUTコマンドの入力形式と、それに先立つGRAPHMANコマンドも含めた使用例を以下に示す。

コマンド	オペランド
GOUT	データセット名 [SY ({ K S <u>O</u> U Q })]

1) オペランドの説明

データセット名：BCOPYコマンドで出力した図形データデータセットを指定する。省略不可。

SY ({ K | S | O | U | Q }) :出力クラスを指定する。省略値はSY (O) 。

2) 使用例

>

>%ATTR #GR RECFM(F) LRECL(1544) BLK(1544) DS(PS)

>%ALLOC DA(NLP.DATA) SP(30 10) NE CA T USING(#GR)

--> 新規に出力先データセットを作成するときは、あらかじめデータセットを確保しておく

>DBCOPY NLP.DATA -----> 出力先データセットの宣言

>DISPLAY -----> 図形のグラフィック画面への作図

>BCOPY * -----> BCOPYデータセットへの出力. この場合
出力領域をグラフィックカーソルで切り出す

>ENDG

READY

GOUT NLP.DATA SY(U) -----> 図形データをオープンNLPへ出力する

READY

3) 使用上の注意

このGOUTコマンドは、GRAPHMANの中からTSSコマンドを呼び出す形式の%を付けて%GOUTと入力して、GRAPHMANの中からも使用することができる。ただし、本来このユーティリティは、GRAPHMANの外で使用される仕様になっているので、%GOUTとして実行すると、BSAMエラーメッセージが出力されることがあるが、この時も図形は正常にNLPに出力されているので、無視して差し支えない。

参考文献

1. 九大大型計算機センターニュースNo. 305, 会話型図形処理システムGRAPHMANの使用について
- ✓ 2. 計算機マニュアル FACOM OSIV/F4 MSP GRAPHMAN説明書 (78AR-517 0-2), 富士通(株).

付録 1 GRAPHMAN コマンド一覧

[富士通マニュアル [2] より転載]

分類	コマンド名	機能概要	Type-1
演算・制御コマンド	ARRAY	配列変数を確保する。	○
	FREE	変数領域を解放する。	○
	CLEAR	すべての変数領域を解放する。	○
	PRINT	変数の値を出力する。	○
	算術代入コマンド	算術式の値を変数に代入する。	○
	配列代入コマンド	配列の各要素に値を代入する。	○
	WHILE~WEND	繰り返しの制御を行う。	○
	IF~ELSE~ENDIF	条件分岐の制御を行う。	○
	DO~DOEND	繰り返しの制御を行う。	○
	CANCEL	制御コマンドの入れ子を途中で解除する。	○
PDB操作コマンド	CREATE	PDBを創成する。	
	PDB	使用するPDBの宣言を行う。	
	DELPIC	PDBからピクチャを削除する。	
	CONDENSE	PDBを圧縮する。	
	SPACE	PDB内の領域の使用状況を出力する。	
	RENAME	ピクチャ名を変更する。	
	LISTC	PDBに登録されているピクチャ名の一覧を出力する。	
	TRANSFER	ピクチャを別のPDBへ複写する。	
PDBCOPY	あるPDBのピクチャをすべて別のPDBへ複写する。		
図形操作マ	USE	使用するピクチャを宣言する。	
	LISTD	ピクチャに関する詳細情報を出力する。	○
	DISPLAY	ピクチャを作画する。	○
	ERASE	表示されている画面全体を消去する。	○
	ERASEC	画面上のコマンド部分だけを消去する。	○
	AGAIN	ERASEされた図形をもう一度表示する。	○
	DBCOPY	Batch-Copyデータセットを宣言する。	○
D	BCOPY	現在画面に表示されている図形をBatch-Copyデータセットに出力する。	○

(注) Type-1の欄に○印のついているものはサブコマンドモードでも実行可能なコマンドである。

分類	コマンド名	機能概要	Type-1
図 形 操 作 コ マ ン ド	SCOPY	現在画面に表示されている図形のハードコピーをとる。	○
	COLOR	色の指定を行う。	○
	PICKING	画面上の任意の位置の座標値を得る。	○
	ZOOMING	画面上の一部を拡大・縮小する。	○
	NEWPIC [1] ONEDIM [2] MESH2D [3] FLOW2N [4] FLOW2R [5] CONT2N [6] CONT2C [7] CONT2R [8] MESH3D. [9] BEGIN [10] PICEND [11] END	新しいピクチャをPDBに登録する。 データパターンONEDIMの計算結果データを登録する。 データパターンMESH2Dの計算結果データを登録する。 データパターンFLOW2Nの計算結果データを登録する。 データパターンFLOW2Rの計算結果データを登録する。 データパターンCONT2Nの計算結果データを登録する。 データパターンCONT2Cの計算結果データを登録する。 データパターンCONT2Rの計算結果データを登録する。 データパターンMESH3Dの計算結果データを登録する。 いままでに登録されたデータを取消し、初期状態に戻す。 データの登録が終了したことを宣言する。 NEWPICのサブコマンドモードを終了させる。	
	OLDPIC [1] LIST [2] GET [3] END	PDBに格納されている計算結果データを変数領域にもってくる。 ピクチャに関する情報を出力する。またGETサブコマンドの使用方法に関する出力を行う。 PDBから計算結果データを指定した配列上にもってくる。 OLDPICのサブコマンドモードを終了させる。	
	EDIT [1] LIST [2] ORIGIN [3] FRAME [4] X [5] Y(又はL1) [6] L2 L3 R1 R2 R3 [7] Z [8] END	現在の図形編集データを変更する。 現在の図形編集データを出力する。 フレーム座標系の原点を設定する。 フレーム空間の大きさなどを設定する。 X軸に関するパラメタを設定する。 Y軸に関するパラメタを設定する。 Z軸に関するパラメタを設定する。 EDITのサブコマンドモードを終了させる。	
	EDIT COMMENT [1] LIST [2] INPUT [3] ~ [7] 行番号 [8] CHANGE [9] RENUM [10] DELETE [11] GO [12] END	コメントの編集を行う。 コメントに関する情報を出力する。 インプットモードに変更する。 コメントの登録、変更、削除を行う。 コメントに関するパラメタを変更する。 行番号を振り直す。 指定された行番号をもつコメントを削除する。 コメントを作画する。 EDIT COMMENTのサブコマンドモードを終了させる。	

分類	コマンド名	機能概要	Type-1
図 形 操 作 コ マ ン ド	EDIT LINE [1] LIST [2] INPUT [3] 行番号 [4] CHANGE [5] GET [6] RENUM [7] DELETE [8] GO [9] END	折線データを編集する。 折線データに関する情報を出力する。 インプットモードに変更する。 折線データの登録, 変更, 削除を行う。 折線データに関するパラメタを変更する。 折線データを指定した変数領域にもってくる。 行番号を振り直す。 指定された行番号をもつ折線データを削除する。 折線データを作成する。 EDIT LINEのサブコマンドモードを終了させる。	
	EDIT ONEDIM [1] LIST [2] CURVE [3] END	現在の図形編集データを変更する。 現在の図形編集データを出力する。 曲線に関するパラメタを変更する。 EDIT ONEDIMのサブコマンドモードを終了させる。	
	EDIT MESH2D [1] LIST [2] MESH [3] END	現在の図形編集データを変更する。 現在の図形編集データを出力する。 2次元メッシュ図のパラメタを変更する。 EDIT MESH2Dのサブコマンドモードを終了させる。	
	EDIT FLOW2D [1] LIST [2] VECTOR [3] END	現在の図形編集データを変更する。 現在の図形編集データを出力する。 表示するベクトルの種類, 大きさの指定を行う。 EDIT FLOW2Dのサブコマンドモードを終了させる。	
	EDIT CONTOUR [1] LIST [2] AUTOHVAL [3] HVAL [4] GET [5] RANDOM [6] CYLINDER [7] SPLINE [8] END	現在の図形編集データを変更する。 現在の図形編集データを出力する。 等高線に関するパラメタを変更する。 等高線に関するパラメタを変更する。 ([2] と [3] では等高値の指定の仕方が異なる。) 等高線に関するパラメタ値を変数領域にもってくる。 データパターンCONT2Rに関するパラメタを変更する。 データパターンCONT2Cに関するパラメタを変更する。 等高線を書くときスプライン補間するかしないかを指定する。 EDIT CONTOURのサブコマンドモードを終了させる。	
	EDIT PERSPECT [1] LIST [2] VIEWP [3] OBJECT [4] INTERVAL [5] HIDDEN [6] CONTOUR [7] ALPHA [8] GET [9] END	現在の図形編集データを変更する。 現在の図形編集データを出力する。 透視図の物体, 画面, 視点に関するパラメタを変更する。 物体の大きさに関するパラメタを変更する。 鳥瞰図の格子線に関するパラメタを変更する。 隠線処理に関するパラメタを変更する。 鳥瞰図を等高線でのみ表示するか否かを設定する。 多曲線の3次元図の α データを変更する。 多曲線の3次元図の α データを指定した変数領域にもってくる。 EDIT PERSPECTのサブコマンドモードを終了させる。	

分類	コマンド名	機能概要	Type-1
マ ク ロ 操 作 コ マ ン ド	DMACRO	マクロデータセットを宣言する。	○
	EDIT MACRO	マクロの編集を行う。	
	[1] LIST	マクロのテキストを出力する。	
	[2] INPUT	インプットモードに変更する。	
	[3] 行番号	テキストの登録, 変更, 削除を行う。	
	[4] CHANGE	テキスト内の文字列を変更する。	
	[5] RENUM	行番号を振り直す。	
	[6] DELETE	指定された行番号をもつテキストを削除する。	
	[7] MSAVE	編集中のマクロを指定したデータセットへ格納する。	
	[8] END	EDIT MACROのサブコマンドモードを終了させる。	
	#マクロ名	マクロを実行する。	○
ユ テ ィ リ テ ィ コ マ ン ド	UOPEN	UREADデータセットをオープンする。	○
	UCLOSE	UREADデータセットをクローズする。	○
	UREAD	データを読み込む。	○
	UBACK	UREADデータセットをバックスペースする。	○
	UREWIND	UREADデータセットをリワインドする。	○
	PAUSE	一時的に実行を停止する。	○
	WHAT	GRAPHMANシステムの状態を知る。	○
	SYSPARA	GRAPHMANシステムのパラメタを変更する。	
	DEVPARA	グラフィック装置に関するパラメタを変更する。	
	∅ TSS コマンド*	TSS コマンドを実行する。	○
	CALL	ユーザ作成プログラムを実行する。	○
	ENDG	GRAPHMANを終了する。	

付録 2

ピクチャ登録サブルーチン一覧

PDBCRE

1) 機能

PDBの創設を行う。

2) 形式

```
CALL PDBCRE (' PDBデータセット名&', nrec1, nrec2,
             ' ボリューム通し番号&', ' 装置指定名&', ' MSVグループ名&', logno)
```

' PDBデータセット名&'	創設するPDBの名前をユーザ識別子付きの完全名で指定する。末尾の&は省略不可。
nrec1, nrec2	PDBのレコード数をnrec1が初期値、nrec2が増分値として指定する。PDBは1レコードが6144バイト固定である。双方とも整数型(4バイト)。
' ボリューム通し番号&'	データセットを創成するボリュームを文字列として与える。省略する時には' &' のみを指定する。MSS使用時のみ必要。
' 装置指定名&'	データセットを創成する装置を文字列として与える。省略する時には' &' のみを指定する。MSS使用時のみ必要で、' MSS&' とする。
' MSVグループ名&'	本センターでは指定不可。' &' のみ指定する。
logno	PDBに割り当てる論理機番を指定する。

3) 注意事項

- ・ PDBの使用状況はSPACEコマンドで知ることができる。
- ・ PDBCREが呼ばれるとPDBのオープンも行われるので、次にPDBCREやOPNPDBが呼ばれるまでここで創設したPDBにピクチャが登録される。
- ・ 指定された名前のデータセットは既に存在していないこと
- ・ logno (論理機番) は計算プログラムの中で使用していない番号を指定する。

4) 使用例

データセット名' F9999.PDB.DATA' をMSSに新規に作成する。レコード数の初期値は30、増分値は10とし、機番としては99を使う。

```
CALL PDBCRE(' F9999.PDB.DATA&', 30, 10, ' MC0009&', ' MSS&', '&', 99)
```

OPNPDB

1) 機能

PDBのオープンを行う。ピクチャ登録の前に必ず呼び出すこと。ただし、PDBCREで創成したPDBをそのまま使用する時には必要なし

2) 形式

```
CALL OPNPDB (' PDBデータセット名&', logno)
```

' PDBデータセット名&'	使用するPDBの名前をユーザ識別子付きの完全名で指定する。末尾の&は省略不可。
----------------	---

logno PDBに割り当てる論理機番を指定する。

3) 注意事項

- ・指定するPDBは事前にPDBCREルーチンかCREATEコマンドで作成されていること。
- ・logno (論理機番) は計算プログラムの中で使用していない番号を指定する。

4) 使用例

既に作成済みのデータセット' F9999.PDB.DATA' をPDBとしてオープンする。機番としては99を使う。

```
CALL OPNPDB('F9999.PDB.DATA&'、99)
```

DELPIC

1) 機能

PDBからピクチャを削除する。

2) 形式

CALL DELPIC (pic, nopic)	
pic	削除するピクチャ名の文字列部分を長さ5～8の文字列として指定する
nopic	削除するピクチャ名の数字部分を指定する。1 ≤ nopic ≤ 99999。

3) 注意事項

- ・picは必要なだけ空白を加えて5文字以上にすること。
- ・NEWPICとPICENDの間で使用してはならない。

4) 使用例

ピクチャABC (1) を削除する。

```
CALL DELPIC('ABC '、1)
```

PDBCON

1) 機能

PDBの圧縮を行う。

2) 形式

CALL PDBCON ('バックアップデータセット名&')	
'バックアップデータセット名&'	PDBの圧縮を行う時のバックアップ用データセット名をユーザ識別子付きの完全名で指定する。末尾の&は省略不可。

3) 注意事項

- ・圧縮するPDBは既にオープンされていること。
- ・バックアップ用データセットは事前に存在していないこと。
- ・バックアップ用データセットは圧縮を始める前に確保され、圧縮が成功すると消去される。
- ・NEWPICとPICENDの間で使用してはならない。

4) 使用例

' F9999.PDB.DATA' の圧縮を行う。バックアップデータセットとしては' F9999.PDBC

OND. DATA' を用いる.

```
CALL PDBCON('F9999.PDBCOND.DATA&')
```

CLSPDB

1) 機能

PDBのクローズを行う.

2) 形式

```
CALL CLSPDB
```

NEWPIC

1) 機能

ピクチャ登録の開始を宣言する.

2) 形式

```
CALL NEWPIC (pic, nopic, '文字列&')
```

pic	登録するピクチャ名の文字列部分を長さ5~8の文字列として指定する.
nopic	登録するピクチャ名の数字部分を指定する. $1 \leq \text{nopic} \leq 99999$.
'文字列&'	ピクチャに対するコメントを与える. 72文字以内で最後に必ず"&"を付加すること.

3) 注意事項

- ・picは必要だけ空白を加えて5文字以上にすること.
- ・指定したピクチャ名は既にPDBに存在しないこと.

4) 使用例

ピクチャABC (2) の登録を開始する.

```
CALL NEWPIC('ABC '、2、'&').
```

PICEND

1) 機能

ピクチャ登録の終了を宣言する.

2) 形式

```
CALL PICEND
```

3) 注意事項

- ・ピクチャの登録はNEWPICで始まりPICENDで終わること. この2つのルーチンの中で, データベースの登録サブルーチンを呼ぶ.

ONEDIM

1) 機能

データパターンONEDIM (1次元図および多曲線の3次元図に関する計算結果データ) を登録する.

2) 形式

```
CALL ONEDIM (n x, x, y, ' 文字列&')
```

n x	曲線を構成するデータ数. 整数型 (4バイト).
x (1~n x)	曲線のxデータ. 単精度の実数型1次元配列.
y (1~n y)	曲線のyデータ. 単精度の実数型1次元配列.
' 文字列&'	曲線識別のためのコメントを20文字以内で指定できる. ' &' は省略できない.

3) 使用例

```
DIMENSION X1(5),Y1(5),X2(6),Y2(6),X3(8),Y3(8),.....
.
.
CALL NEWPIC('ONEDIM',1,'&')
CALL ONEDIM(5,X1,Y1,'CURVE-1&')
CALL ONEDIM(6,X2,Y2,'CURVE-2&')
CALL ONEDIM(8,X3,Y3,'CURVE-3&')
CALL PICEND
```

MESH2D

1) 機能

データパターンMESH2D (2次元メッシュ図に関する計算結果データ) を登録する.

2) 形式

```
CALL MESH2D (n, x, y, m, n h)
```

n	ノード数. 整数型 (4バイト).
x	各ノードのx座標. 単精度の実数型1次元配列.
y	各ノードのy座標. 単精度の実数型1次元配列.
m	エレメント数. 整数型 (4バイト).
n h (1~4, 1~m)	ノード・エレメントテーブルデータ. 単精度の整数型2次元配列.

3) 注意事項

3角形エレメントの場合, 必ずn h (4, m) = 0としておく.

4) 使用例

```
DIMENSION X(100),Y(100),NH(4,300).....
.
.
CALL NEWPIC('MESH ',1,'EXAMPLE&')
```

```
CALL MESH2D(100,X,Y,300,NH)
CALL PICEND
```

FLOW2N

1) 機能

データパターン FLOW2N (2次元流れ図に関する計算結果データ) を登録する。

2) 形式

```
CALL FLOW2N (nx, ny, x, y, vx, vy, lmax)
```

nx	x 軸の格子線数. 整数型 (4バイト).
ny	y 軸の格子線数. 整数型 (4バイト).
x (1~nx)	x データの座標値. 単精度の実数型 1 次元配列.
y (1~ny)	y データの座標値. 単精度の実数型 1 次元配列.
vx (1~nx, 1~ny)	各格子点におけるベクトル v の x 成分. 単精度の実数型 2 次元配列.
vy (1~nx, 1~ny)	各格子点におけるベクトル v の y 成分. 単精度の実数型 2 次元配列.
lmax	2 次元配列の最初の次元の大きさを与える. 整数型 (4バイト).

3) 使用例

```
DIMENSION X(50),Y(50),VX(100,90),VY(100,90) . . .
.
.
CALL NEWPIC('FLOW ',1,'EXAMPLE&')
CALL FLOW2N(50,50,X,Y,VX,VY,100)
CALL PICEND
```

FLOW2R

1) 機能

データパターン FLOW2R (2次元流れ図に関する計算結果データ) を登録する。

2) 形式

```
CALL FLOW2R (n, x, y, vx, vy)
```

n	ランダムデータ数. 整数型 (4バイト).
x (1~n)	各ランダムデータ点の x 座標値. 単精度の実数型 1 次元配列.
y (1~n)	各ランダムデータ点の y 座標値. 単精度の実数型 1 次元配列.
vx (1~n)	各点におけるベクトル v の x 成分. 単精度の実数型 1 次元配列.
vy (1~n)	各点におけるベクトル v の y 成分. 単精度の実数型 1 次元配列.

3) 使用例

```
DIMENSION X(60),Y(60),VX(60),VY(60), . . .
.
.
CALL NEWPIC('FLOW2',1,'EXAMPLE&')
```

```
CALL FLOW2R(60,X,Y,VX,VY)
CALL PICEND
```

CONT2N

1) 機能

データパターンCONT2N (等高線図および鳥瞰図に関する計算結果データ) を登録する。

2) 形式

CALL CONT2N (nx, ny, x, y, z, lmax)	
nx	x軸の格子線数. 整数型 (4バイト).
ny	y軸の格子線数. 整数型 (4バイト).
x (1~nx)	格子点のx座標値. 単精度の実数型1次元配列.
y (1~ny)	格子点のy座標値. 単精度の実数型1次元配列.
z (1~nz)	格子点のz座標値. 単精度の実数型1次元配列.
lmax	2次元配列の最初の次元の大きさを与える. 整数型 (4バイト).

3) 使用例

```
DIMENSION X(50),Y(50),Z(100,90),...
.
.
CALL NEWPIC('CONT2N',1,'&')
CALL CONT2N(50,50,X,Y,Z,100)
CALL PICEND
```

CONT2C

1) 機能

データパターンCONT2C (等高線図および鳥瞰図に関する計算結果データ) を登録する。

2) 形式

CALL CONT2C (nr, nθ, r, θ, z, lmax)	
nr	r座標の分割数 (中心点も含めた同心円の数). 整数型 (4バイト).
nθ	θ座標の分割数. 整数型 (4バイト).
r (1~nr)	同心円のr座標値. 単精度の実数型1次元配列.
θ (1~nθ)	ある基準からなす角度 (ラジアン). 単精度の実数型1次元配列.
z (1~nr, 1~nθ)	各点における曲面 $z = f(r, \theta)$ の値. 単精度の実数型2次元配列.
lmax	2次元配列の最初の次元の大きさを与える. 整数型 (4バイト).

3) 使用例

```
DIMENSION R(50),THETA(50),Z(100,90),...
.
.
CALL NEWPIC('CONT2C',1,'&')
```

```
CALL CONT2C(50,50,R,THETA,Z,100)
CALL PICEND
```

CONT2R

1) 機能

データパターンCONT2R (等高線図および鳥瞰図に関する計算結果データ) を登録する。

2) 形式

```
CALL CONT2R (n, x, y, z)
```

または,

```
CALL CONT2R (-n, x, y, z, m, nh)
```

n	ランダムデータ数. 整数型 (4バイト). 負の符号で与えた時はm, nh も指定する.
x (1~n)	各点のx座標値. 単精度の実数型1次元配列.
y (1~n)	各点のy座標値. 単精度の実数型1次元配列.
z (1~n)	各点のz座標値. 単精度の実数型1次元配列.
m	エレメント数. 整数型 (4バイト).
nh (1~4, 1~m)	ノード・エレメントテーブルを与える. 整数型2次元配列.

3) 注意事項

- ・GRAPHMANはノード・エレメントテーブルが与えられていないと, ランダムデータx, yから自動的にテーブルを生成するので, この自動的に生成するテーブルでは不都合が生じる時に $n < 0$ としてノード・エレメントテーブルを与える.

4) 使用例

```
DIMENSION X(100),Y(100),Z(100),NH(4,200),...
```

·
·

```
CALL NEWPIC('CONT2R',1,'&')
```

```
CALL CONT2R(100,X,Y,Z)
```

または,

```
CALL CONT2R(-100,X,Y,Z,200,NH)
```

```
CALL PICEND
```

MESH3D

1) 機能

データパターンMESH3D (3次元メッシュ図に関する計算結果データ) を登録する。

2) 形式

```
CALL MESH3D (n, x, y, z, m, nh)
```

n	ノード数. 整数型 (4バイト).
---	-------------------

x (1~n)	各ノードのx座標値. 単精度の実数型1次元配列.
y (1~n)	各ノードのy座標値. 単精度の実数型1次元配列.
z (1~n)	各ノードのz座標値. 単精度の実数型1次元配列.
m	エレメント数. 整数型 (4バイト).
n h (1~8, 1~m)	ノード・エレメントテーブルを与える. 整数型2次元配列.

3) 使用例

```

DIMENSION X(150),Y(150),Z(150),NH(8,300),...
.
.
CALL NEWPIC('MESH3D',1,'&')
CALL MESH3D(150,X,Y,Z,300,NH)
CALL PICEND

```

◇ プログラムライブラリ開発課題の募集について

センターではプログラムライブラリ開発課題を随時募集しています。

開発計画をお持ちの方は「プログラムライブラリ開発課題申請書」に必要事項を記入し、センター図書室に提出して下さい。

なお、対象となる課題は広く各分野で使われる可能性のあるプログラム、またはある専門分野に限られるが、その分野では広く使われる可能性のあるプログラムで、新規開発および書換えを含みます。その他詳細については利用の手引ライブラリ編を参照し、不明な点はセンター図書室にお問合せ下さい。