資　料

地調月報 Bull. Geol. Surv. Japan, vol. 30, (8), p. 469-477, 1979　　681.322.06

コンピュータによる図形表示（II）

——鳥瞰図——

中塚　正*

Computer Graphics (III)

——Bird's-eye View——

Tadashi NAKATSUKA

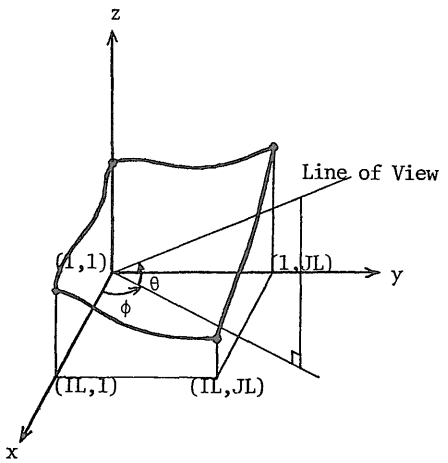
1. はじめに

本稿においては，地質調査所コンピュータ TOSBAC-3400/51のカーブプロッタ（LCM 3401 D）用に筆者が開発した鳥瞰図作成サブプログラムについて，その原理と使用法を解説する．3次元的な図形を平面上に表現する手法の一つとして，数量的表現に適したコンターマップ（等値線図）を描くソフトウェアについては別稿ですでに述べた．鳥瞰図は数量的表現にはあまり適さないが，直視的であるので定性的な形状の把握に適しており，コンターマップとは対照的な得失をもつ．これら両手法を目的に応じて使い分けることにより，各種データの迅速かつ正確な解析・解釈に役立つならば，筆者の望むところである．

2. 概　要

サブルーチン "PVIEW" は，高さ分布として与えられた格子点データによって表現される曲面の鳥瞰図を描くサブプログラムである．描く図の大きさはカーブプロッタの制約から 30 cm×24cm 以下でなければならず，図を描く位置はサブルーチンが自動的に決定するが，格子の大きさ・高さ方向のスケールおよび視線方向は引数によって指定できる．また，格子点データは2次元の整合配列として引数で指定する．なお，格子内ではデータを線型に補間し，隣り合う格子点データ間を線分によって表現する．

鳥瞰図は基本的には，表現すべき曲面と等間隔な平行平面群との交線を視線方向に垂直な平面に投影したものであり，通常互いに直交する2組の平面群との交線を投影する．コンターマップは1組の水平な平面群との交線



第1図

を水平面（視線方向が鉛直）に投影したものに等しく，広義には一種の鳥瞰図とも見られるが，鳥瞰図は一般的には斜め方向の視線をもつ．本サブルーチンでは2組の平面群として，格子点データの格子を形成する直線群に1対1に対応した鉛直な平面群を用いており，その間隔はデータ点間隔と同一である．また，プログラム処理の都合上，視線方向の伏角・偏角（第1図の θ・ϕ）を0-90°に規制している．

3. 原　理

本サブルーチンサブプログラムのソースリストを第2図に，処理のフローチャートを第3図に示す．鳥瞰図を描くに際して独特の問題は，視線方向に関して手前にある部分によって奥の部分が隠される場合の処理である．描くべき曲面を透明なものとして奥にある部分も同様に描く方法もあるが，その場合，曲面が複雑な形態をして

* 物理探査部

```
      SUBROUTINE PVIEW(SA, DA, FS, DM, IL, JL, F)
      DIMENSION L(1502), F(IL,JL)
      IF(SA.LE.0..OR.SA.GE.90.) GO TO 5
      IF(DA.LE.0..OR.DA.GE.90.) GO TO 5
      IF(FS.EQ.0.) GO TO 5
      IF(DM.LE.0.) GO TO 5
      IF(IL.LE.1.OR.JL.LE.1) GO TO 5
      PAI = 3.14159265 / 180.
      CSA = COS(SA * PAI) * 5. * DM
      SSA = SIN(SA * PAI) * 5. * DM
      CDA = COS(DA * PAI)
      SDA = SIN(DA * PAI)
      CI = CSA * SDA
      CJ = SSA * SDA
      CF = CDA * 5. / FS
      FI = FLOAT(IL-1)
      FJ = FLOAT(JL-1)
      XO = 752. + (FI*SSA - FJ*CSA) / 2.
      YO = FI*CI + FJ*CJ + 50.
      PX = FI*SSA + FJ*CSA
      IF(PX.GT.1500..OR.YO.GT.1200.) GO TO 6
      CALL MOVE(9)
      CALL MOVE(12)
      MX = -50
      MY = 1
      DO 1 I=1,1502
      L(I) = 0
    1 CONTINUE
      KKK = 1
      J = 1
      I = IL + 1
   10 K = 1
      KK = 1
      I = I - 1
      GO TO 30
   11 GO TO 40
   12 LM = 0
      J = J + 1
      GO TO 31
   13 IY = IFIX(PYO)
      IF(IY.GE.MY) GO TO 14
      CALL MOVE(10)
      GO TO 44
   14 CALL MOVE(9)
      IF(I.EQ.1) GO TO 2
      GO TO 70
   15 K = 2
      KK = 2
      I = I - 1
      GO TO 30
   16 GO TO 40
   17 LM = IFIX(PYO)
      J = J - 1
      GO TO 31
   18 CALL MOVE(9)
      IF(I.EQ.1) GO TO 2
      GO TO 70
    2 DO 3 I=1,1502
      L(I) = 0
    3 CONTINUE
```

第 2 図(1)　サブルーチン “PVIEW” のソースリスト

```
   KKK = 2
   I = 1
   J = JL + 1
20 K = 3
   KK = 2
   J = J - 1
   GO TO 30
21 GO TO 40
22 LM = 0
   I = I + 1
   GO TO 32
23 IY = IFIX(PYO)
   IF(IY.GE.MY) GO TO 24
   CALL MOVE(10)
   GO TO 44
24 CALL MOVE(9)
   IF(J.EQ.1) GO TO 4
   GO TO 74
25 K = 4
   KK = 1
   J = J - 1
   GO TO 30
26 GO TO 40
27 LM = IFIX(PYO)
   I = I - 1
   GO TO 32
28 CALL MOVE(9)
   IF(J.EQ.1) GO TO 4
   GO TO 74
30 FI = FLOAT(I-1)
   FJ = FLOAT(J-1)
   PXI = XO - FI*SSA
   PXJ = XO + FJ*CSA
   PYI = YO - FI*CI
   PYJ = YO - FJ*CJ
   PX = PXI + PXJ - XO
   PYO = PYI + PYJ - YO
   PY = PYO + F(I,J)*CF
   IX = IFIX(PX + 0.5)
   GO TO (11,16,21,26,81), K
31 PPX = PX
   PPY = PY
   FJ = FLOAT(J-1)
   PX = PXI + FJ*CSA
   PYO = PYI - FJ*CJ
   PY = PYO + F(I,J)*CF
   DX = PX - PPX
   DY = PY - PPY
   PXY = PX*PPY - PY*PPX
   GO TO (51,54,85), KK
32 PPX = PX
   PPY = PY
   FI = FLOAT(I-1)
   PX = PXJ - FI*SSA
   PYO = PYJ - FI*CI
   PY = PYO + F(I,J)*CF
   DX = PX - PPX
   DY = PY - PPY
   PXY = PX*PPY - PY*PPX
   GO TO (51,54,85), KK
```

(2)

```
40 IF(IX-MX) 41,43,42
41 CALL MOVE(3)
   MX = MX - 1
   GO TO 40
42 CALL MOVE(4)
   MX = MX + 1
   GO TO 40
43 GO TO (12,17,22,27,82), K
44 IF(IY-MY) 45,47,46
45 CALL MOVE(1)
   MY = MY - 1
   GO TO 44
46 CALL MOVE(2)
   MY = MY + 1
   GO TO 44
47 GO TO (14,24,83,87), KKK
50 LM = MIN0(LP,L(MX))
51 FX = FLOAT(MX) + 0.5
   IF(FX.LT.PX) GO TO 57
   IF(K.EQ.4) GO TO 52
   IF(J.EQ.JL) GO TO 67
   J = J + 1
   GO TO 31
52 IF(I.EQ.1) GO TO 67
   I = I - 1
   GO TO 32
53 LM = MIN0(LP,L(MX-1))
54 FX = FLOAT(MX) - 0.5
   IF(FX.GT.PX) GO TO 56
   IF(K.EQ.3) GO TO 55
   IF(J.EQ.1) GO TO 67
   J = J - 1
   GO TO 31
55 IF(I.EQ.IL) GO TO 67
   I = I + 1
   GO TO 32
56 MX = MX - 1
57 FY = (FX*DY + PXY) / DX
   IY = IFIX(FY)
60 IF(IY-MY) 61,63,62
61 IF(MY.EQ.LM) CALL MOVE(9)
   CALL MOVE(1)
   MY = MY - 1
   GO TO 60
62 IF(MY.EQ.LM) CALL MOVE(10)
   CALL MOVE(2)
   MY = MY + 1
   GO TO 60
63 LP = L(MX)
   IF(MY.LT.LP) GO TO 64
   L(MX) = MY
   CALL MOVE(10)
   GO TO 65
64 CALL MOVE(9)
65 IF(KK.EQ.2) GO TO 66
   CALL MOVE(4)
   MX = MX + 1
   GO TO 50
66 CALL MOVE(3)
   GO TO 53
```

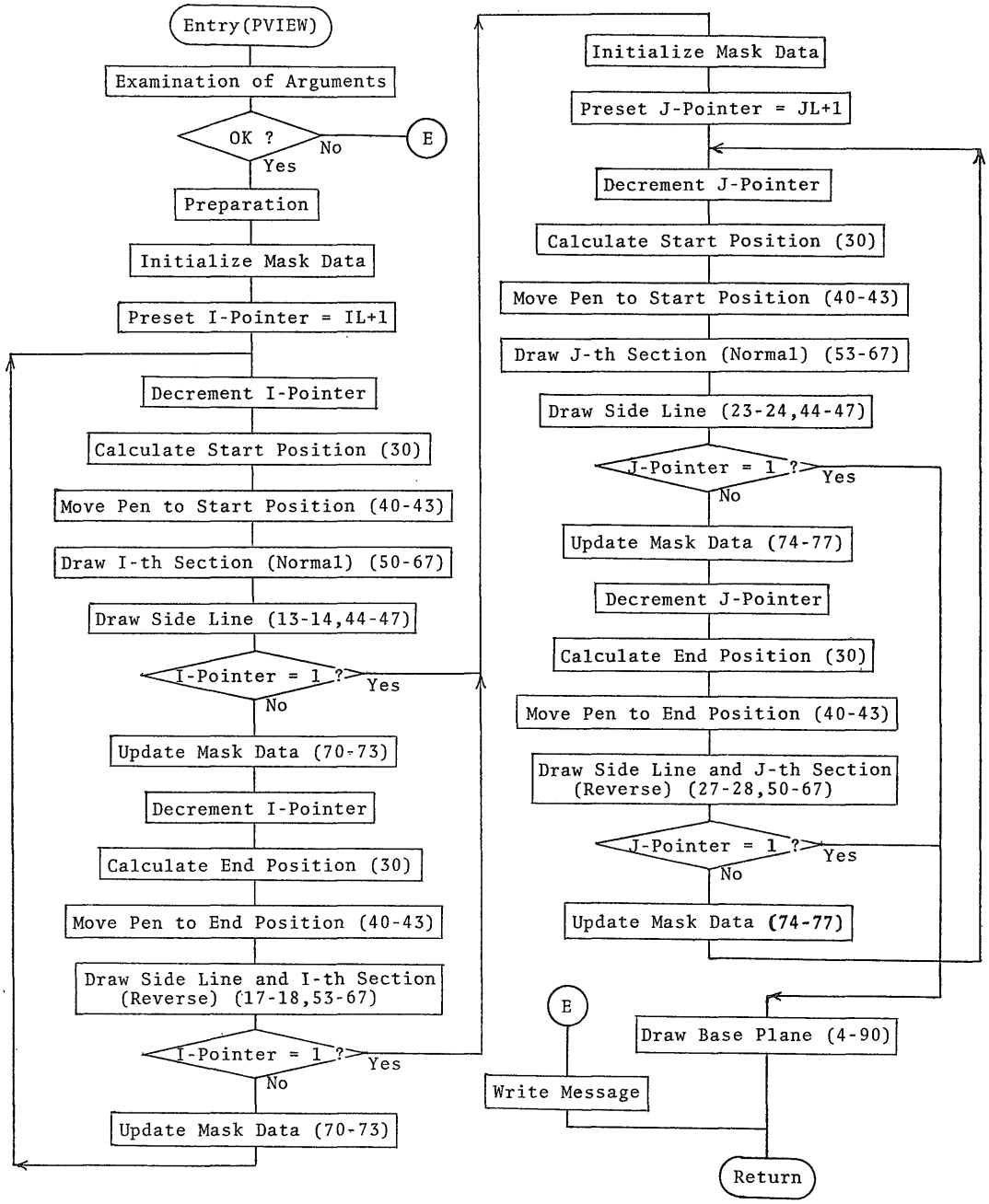
(3)

```
67 GO TO (13,18,23,28,90), K
70 M = 0
71 M = M + 1
   PXJ = PXI + SSA
   PYJ = PYI + CI
   FJ = FLOAT(M-1)
   PPX = PXI + FJ*CSA
   PPY = PYI - FJ*CJ + F(I,M)*CF
   PX = PXJ + FJ*CSA
   PY = PYJ - FJ*CJ + F(I-1,M)*CF
   DX = PX - PPX
   DY = PY - PPY
   PXY = PX*PPY - PY*PPX
   IX = IFIX(PPX+0.5)
72 FX = FLOAT(IX) + 0.5
   IF(FX.GT.PX) GO TO 73
   FY = (FX*DY + PXY) / DX
   IY = IFIX(FY)
   IF(IY.GT.L(IX)) L(IX) = IY
   IX = IX + 1
   GO TO 72
73 IF(M.LT.JL) GO TO 71
   GO TO 78
74 M = 0
75 M = M + 1
   PXI = PXJ - CSA
   PYI = PYJ + CJ
   FI = FLOAT(M-1)
   PPX = PXJ - FI*SSA
   PPY = PYJ - FI*CI + F(M,J)*CF
   PX = PXI - FI*SSA
   PY = PYI - FI*CI + F(M,J-1)*CF
   DX = PX - PPX
   DY = PY - PPY
   PXY = PX*PPY - PY*PPX
   IX = IFIX(PPX+0.5)
76 FX = FLOAT(IX) - 0.5
   IF(FX.LE.PX) GO TO 77
   FY = (FX*DY + PXY) / DX
   IY = IFIX(FY)
   IX = IX - 1
   IF(IY.GT.L(IX)) L(IX) = IY
   GO TO 76
77 IF(M.LT.IL) GO TO 75
78 GO TO (15,10,25,20,90), K
 4 K = 5
   KK = 3
   I = IL
80 KKK = 3
   GO TO 30
81 PY = PY0
   GO TO 40
82 IY = IFIX(PY)
   GO TO 44
83 CALL MOVE(10)
   KKK = 4
   IF(J.NE.1) GO TO 84
   J = JL
   GO TO 31
84 I = 1
```

(4)

```
   GO TO 32
85 PY = PY0
   DY = PY - PPY
   PXY = PX*PPY - PY*PPX
86 FX = FLOAT(MX) + 0.5
   IF(FX.GT.PX) GO TO 88
   FY = (FX*DY + PXY) / DX
   IY = IFIX(FY)
   GO TO 44
87 CALL MOVE(4)
   MX = MX + 1
   GO TO 86
88 CALL MOVE(9)
   IF(I.NE.1) GO TO 80
90 CALL MOVE(12)
   GO TO 9
 5 WRITE(6,7)
   GO TO 9
 6 WRITE(6,8)
   GO TO 9
 7 FORMAT('0*** ILLEGAL ARGUMENT ***')
 8 FORMAT('0*** TOO WIDE PICTURE ***')
 9 RETURN
   END
```

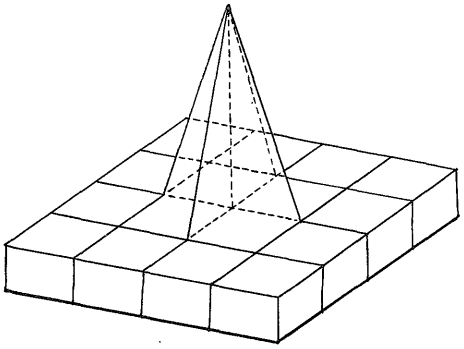
(5)



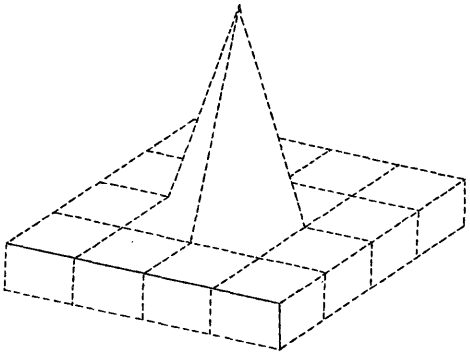
第３図　サブルーチン “PVIEW” のフローチャート

いるとでき上った図がこみ入りすぎて曲面の形状をうまく表現できないことがある．そこで本プログラムでは曲面を不透明とし，手前にある部分によって隠される奥の部分を描かないようにしている．すなわち，プログラム処理としては，線を描く順序を手前寄りからとし，順次マスクデータを更新しながらマスクされない部分の線を描いていく．この処理ではマスクデータをいかなる形で保存するかが問題となるが，ここでは次の方法を用いる．

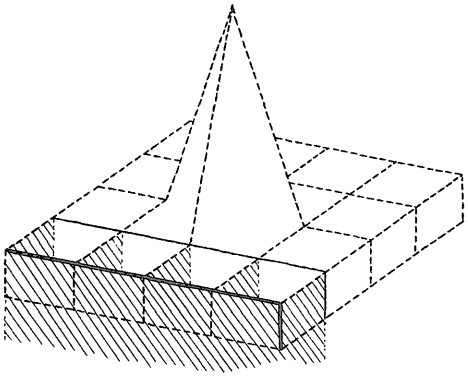
カーブプロッタはステップモータを使用したインクリ



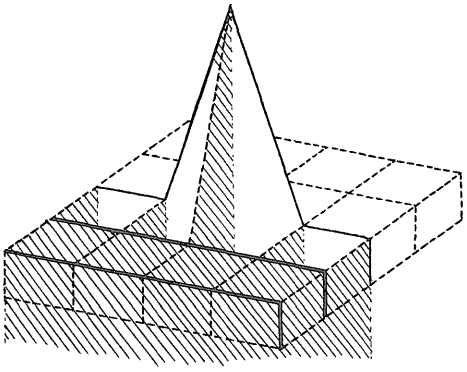
第 4 図



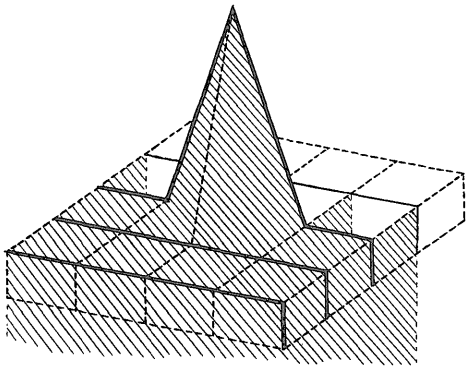
第 5 図(1)



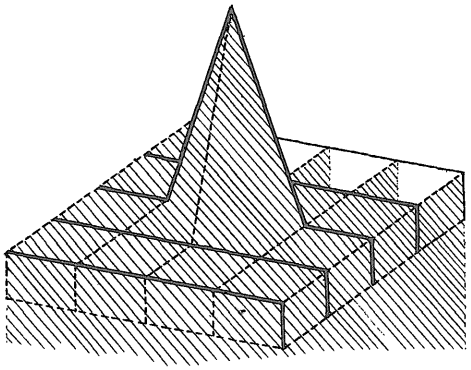
(2)



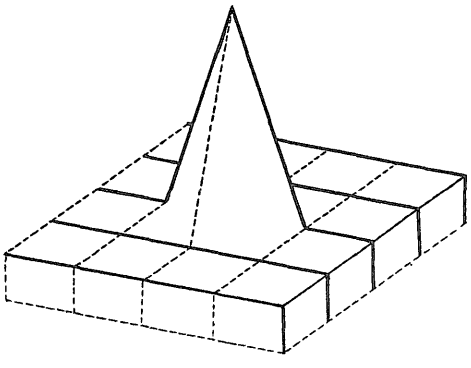
(3)



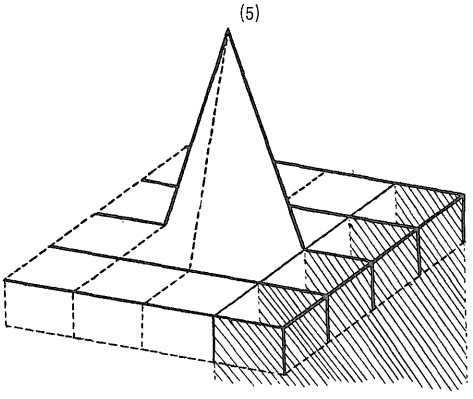
(4)



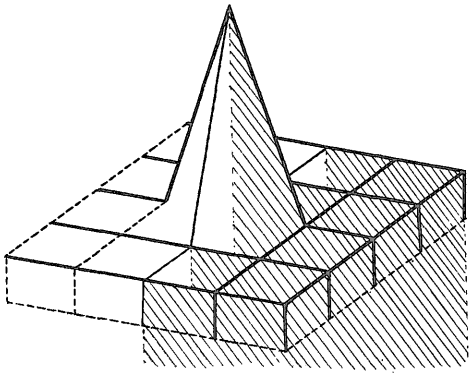
(5)



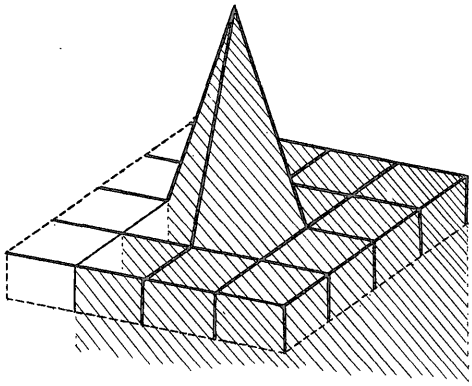
(6)



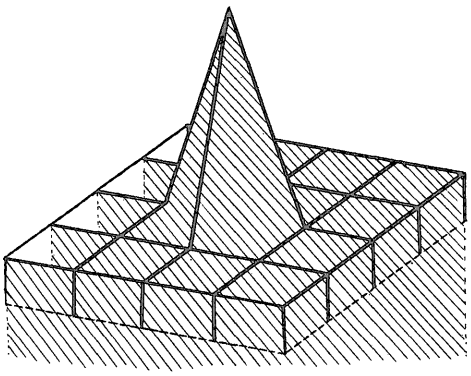
(7)



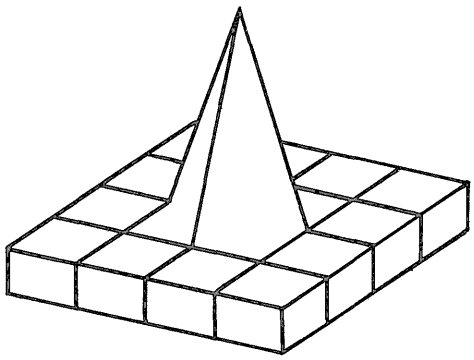
(8)



(9)



(10)



(11)

メンタル型であるので，描かれる図はすべて本質的に0.2 mm 単位に量子化されており，それ以下の分解能をもたない．このため描くべき線をプログラム内で0.2mm単位に量子化しても何ら不都合はない．従って，マスクデータも同様に量子化された位置に対して与えればよく，プログラム上では，各量子単位ごとにマスクされている部分の高さのデータを1次元配列（第2図中の配列 L）として記憶している．線を描くにあたっては，各量子ごとにマスクされているか否かの判定を行って，マスクされていない位置のみを実際に（ペンダウン状態で）描けばよい．

次に重要なことは，マスクデータをいかにして与えるかという問題である．1組の平面群との交線だけを投影する場合は比較的容易で前に描いた線の高さを各量子単位ごとに記憶しておくだけでよいが，2組の平面群との交線を投影する場合には描きつつある交線が他の組の平面群との交線によって隠されるべき場合があり，こうした場合にも正しい図が描かれるような処理が必要である．このための手法は次の如くである．

いまデータが $m\times n$ の要素をもつ2次元配列 F によって与えられているとし，$F(i,1)$, $F(i,2)$, …, $F(i,n)$ の各データ点を結ぶ線をIセクション i, $F(1,j)$, $F(2,j)$, …, $F(m,j)$ の各データ点を結ぶ線をJセクション j と呼び，$F(i,j)$ と $F(i,j+1)$ を結ぶ線をIレッグ ij, $F(i,j)$ と $F(i+1,j)$ を結ぶ線をJレッグ ij と呼ぶものとする．このとき，Iセクション i を描くときのマスクデータとして，Iセクション m，Iセクション $(m-1)$, …, Iセクション $(i+1)$ およびJレッグ $i1$，Jレッグ $i2$, …, Jレッグ in の各量子単位ごとの最高値を与え，Jセクション j を描くときには，Jセクション n，Jセクション $(n-1)$, …, Jセクション $(j+1)$ およびIレッグ $1j$，Iレッグ $2j$, …, Iレッグ mj の各量子単位ごとの最高値をマスクデータとする．このようなマスクデータの与え方を単純なモデル（第4図）を例として図示すると第5図のようになる．第5図で，太線はすでに描かれた線，細線の実線は今描こうとしている線，破線はこれ以後に描かれるべき線を示しており，斜線でハッチされている部分がその時点でマスクデータによりマスクされている

範囲である．なお，処理は(1)，(2)，…，(10)，(11)の順にすすむ．

なお，カーブプロッタのペンの1ステップ動作としては Up・Down のほかに8方位への移動が可能であるが，プログラムの簡略化のため，本プログラムでは4方位への移動のみを使用している．また，カーブプロッタの動作は非常に低速であるので，本プログラムでは作図速度を高めるため，線を描く方向を交互に反転し，ペンのむだな動きを防止している．

4. 使 用 法

サブルーチン "PVIEW" は，FORTRAN 言語の7個の引数をもった CALL 文

CALL PVIEW (*SA, DA, FS, DM, IL, JL, F*)

によって引用することができる．ここに，各引数の意味は次の通りである．

SA: 視線方向の偏角(ϕ)を度単位の値として与える．0以上90以下でなければならない．

DA: 視線方向の伏角(θ)を度単位の値として与える．0以上90以下でなければならない．

FS: 高さ分布データに対するスケールファクター(f)を与える．すなわち，配列 F に与えられたデータ値の f なる大きさを 1 mm の高さに対応させる．但し，図上では視線方向の伏角の影響により $\cos\theta$ 倍に描かれる．

DM: 配列 F によってデータを与える格子点の格子間隔(d)を mm 単位で与える．但し，図上での表現は視線方向の伏角・偏角により異なる．

IL, JL: 2次元整合配列 F の整合寸法．

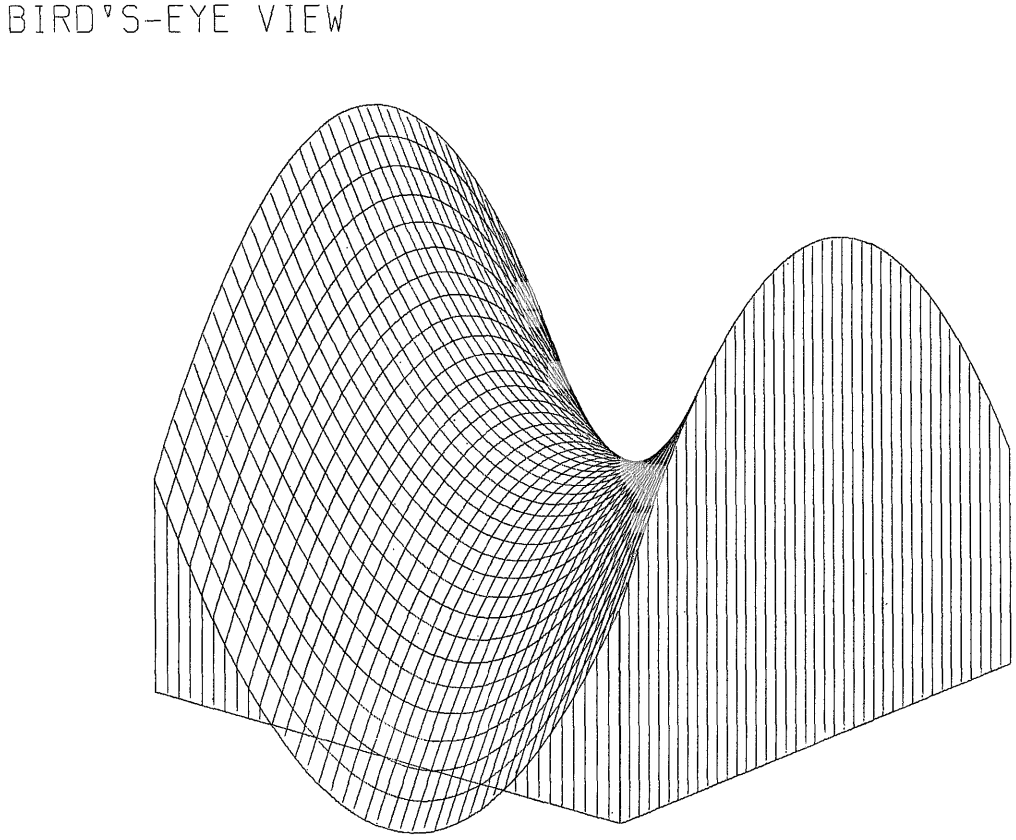
F：格子点での高さ分布データを与える2次元整合配列の配列名．

本サブルーチンは，現在，コンパイルされたオブジェクトモジュールの形で磁気ディスク上に永久ファイルとして登録されているので，数枚のコントロールカード(第6図の＊印部分およびカーブプロッタ使用のための FD コントロールカード) を付加することにより使用できる．

なお，参考までに本サブルーチンの使用例（プログラムおよび出力結果）を第6図および第7図に示す．この使用例において，TOSBAC-3400/51 の CPU 使用時間は約 110 秒，カーブプロッタへの出力に要する時間は約10分であった．

```
¥JOB            ANO=00006204
¥EXC    FTC
      DIMENSION A(41,41), HS(2)
      DATA HS/15HBIRD'S-EYE VIEW/
      DO 1 I=1,41
      II = (I-21) * (I-21)
      DO 1 J=1,41
      JJ = (J-21) * (J-21)
      A(I,J) = FLOAT(JJ-II+300)
    1 CONTINUE
      CALL KSPOOL
      CALL CPLOPN
      CALL PVIEW(40.,20.,5.,4.,41,41,A)
      CALL PORGN(10.,10.)
      CALL CHAR(HS,15,0.,0.,7.,0.,220.,1)
      CALL PENRST
      CALL CPLCLS
      STOP
      END
¥EXC    LED     LE
¥FD     B       DEV=SO,REC=36,BLK=120,NOB1,OLD,FIN=BIRD-EYE *
      *INCLUDEB
¥EXC    *       GO
¥FD     CPL     DEV=TO,BLK=120,REC=120,U,NOB1,NRWD
¥EOJ
```

第6図　鳥瞰図作図プログラム例



第7図　鳥瞰図作図出力例
（第6図プログラム例による出力）