
プログラムのページ (担当 鈴木 誠 道)

**75-05 ラインプリンタによるカーブ
プロッタのシミュレータ**
大野 義 夫*
1. ま え が き

著者の環境ではカーブプロッタの利用が不便であるので、デバッグ用に、プロッタへの出力と同じ図形をプリンタに出力するためのパッケージを作成した。使用した計算機は UNIVAC 1106 である。

プログラムのリンク編集時に、このパッケージをリンクすると図形出力がプリンタに出され、本来のプロッタ用パッケージをリンクするとプロッタ制御用の磁気テープが作られる。

2. 利用方法

このパッケージには次のサブルーチン副プログラムとその補助ルーチンが含まれている。座標系の単位は1インチであるが、初期値設定副プログラムを変更すれば1センチを単位とすることもできる。

PLOTS (ib, isize, iu) 装置番号 iu の磁気テープファイルをオープンする。出力バッファには大きさ isize 語の配列 ib を使用する。

PLOT (x, y, ip) ペンを現在位置から点 (x, y) まで動かす。このとき、 $|ip|=2$ なら線を描きながら、 $|ip|=3$ なら線を描かず動かし、 $ip < 0$ のときは、ペン移動後の位置を新しい座標原点とする。ip=999 のときは、磁気テープファイルをクローズする。

FACTOR(s) 尺度因子を s とする。

WHERE (x, y, s) 現在のペンの座標、尺度因子を x, y, s に与える。

NEWPEN (ipen) ペン ipen を選択する。

この他のプロッタ用ルーチンでも、間接的にこれらのルーチンを呼ぶものはそのまま利用できる。

3. アルゴリズム

プロッタの用紙 (約 90 cm × 32 m) を x 方向に 100 等分し、それぞれの短冊を 1 レコードに対応させたラ

ンダムアクセスファイルに図形を記憶させる。用紙幅に余裕をもたせて、約 140 cm × 32 cm の短冊を 1 レコード (330 × 8 = 2,640 語) とし、図形の描かれた短冊に対応するレコードのみを生成する。

プリンタの活字の大きさに合わせて、1つの短冊を y 方向 330 等分、x 方向 128 等分し、第 i 行第 j 列の小区画をレコードの第 i 行第 (j/18+1) 語中の 2 ビットに対応させる。その区画に図形が描かれていないとき、その 2 ビットを 00 とし、第 ip 番のペン (1 ≤ ip ≤ 3) によって図形が描かれているとき値 ip を与える。

各レコードの内容は、PLOT(0., 0., 999) が呼ばれたときに、プリンタ用紙の 5 ページずつに分割され、プロッタ用紙の限界を示す枠をつけて印刷される。

4. プログラム

このパッケージのルーチンを Fig. 1 (次 1029 頁から 1030 頁参照) に示す。これらのルーチンは 1 語 36 ビットの機械向きに書かれている。

DEFINE FILE iu (maxr, isize, 1 HU, iv) 装置番号 iu のランダムアクセスファイルを定義する。レコード数は最大 maxr 個で、レコードの大きさは isize 語とする。入出力は書式なしで行う。

WRITE (iu*ir) iarea 配列 iarea の内容をレコード ir としてファイル iu に書きこむ。

READ (iu*ir) iarea ファイル iu のレコード ir を配列 iarea に読みこむ。

CALL BETA プリンタ用紙の上下に余白をとらず、“ベタ打ち”を行う。

CALL NBETA “ベタ打ち”をやめる。

AND (i, j) 変数 i, j のビットごとの論理積を与える関数。

FLD (i, j, k) 変数 k の第 i ビットから j ビットのビットパターンを値として与える関数 (i は 0 を始点として数える)。

なお、紙面の制約のため、Fig. 1 では引数のエラーチェックを行う部分をすべて削除してある。また、線分を描くアルゴリズムは文献 1) を利用している。

5. 例

このパッケージを利用して Fig. 2 (p. 1030 参照)

* 慶応義塾大学情報科学研究所

```

SUBROUTINE PLOTS (BUF, BUFSZ, LUNIT)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
PREVX = -1
PREVY = -1
PREVC = -1
CURR = 2
RECORD(1) = 2
DO 30 I = 2, 100
20 RECORD(I) = 0
RCOUNT = 1
DO 30 J = 1, 330
DO 30 K = 1, 8
30 AREA(I,J) = 0
PLOTSW = 0
CURX = 0.0
CURY = 0.0
SCAL = 1.0
COLOR = 1
XOFF = 0.0
YOFF = 2.0
UNIT = LUNIT
DEFINE FILE UNIT(130,2640,1HU,1AV)
RETURN
END

```

```

SUBROUTINE PLOT (XX,YY,IPEN)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
REAL QCURX,QCURY,XX,YY
IF (IPEN .NE. 999) GO TO 10
WRITE (UNIT,CURR) AREA
CALL DISPLY
RETURN
10 IPEN = IABS(IPEN)
IF (IPEN .EQ. 2) GO TO 40
CURX = XOFF + XX*CONV*SCAL
CURY = YOFF + YY*CONV*SCAL
IF (IPEN .EQ. 3) RETURN
50 XOFF = CURX
YOFF = CURY
30 RETURN
40 QCURX = CURX
QCURY = CURY
CURX = XOFF + XX*CONV*SCAL
CURY = YOFF + YY*CONV*SCAL
CALL LINEG2 (INT(10.0*QCURX+0.5)+LOWX+120,
* LOWY-INT(6.0*QCURY+0.5),
* INT(10.0*QCURX+0.5)+LOWX+120,
* LOWY-INT(6.0*QCURY+0.5) )
IF (IPEN - 2) 50, 30, 50
END

```

```

SUBROUTINE FACTOR (S)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF,S
SCAL = S
RETURN
END

```

```

SUBROUTINE NEWPEN (CC)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
COLOR = CC
RETURN
END

```

```

SUBROUTINE WHERE (CORX,CORY,SCALF)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
REAL CORX,CORY,SCALF
CORX = (CURX - XOFF) / (CONV * SCAL)
CORY = (CURY - YOFF) / (CONV * SCAL)
SCALF = SCAL
RETURN
END

```

```

SUBROUTINE LINEG2 (IXA,IYA,IXB,IYB)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
DIMENSION M1(8),M2(8),XSTEP(8),YSTEP(8)
DATA M1/1,3,1,7,5,3,5,7/M2/2,2,8,8,4,4,6,6/
DATA XSTEP/1,1,0,-1,-1,-1,0,1/

```

```

DATA YSTEP/0,1,1,1,0,-1,-1,-1/
IXXA = IXA
IYYA = IYA
DELTA = IXB - IXA
DELTA = IYB - IYA
IF(DELTA .EQ. 0 .AND. DELTA .EQ. 0) RETURN
DIFF = IABS(DELTA) - IABS(DELTA)
OCT = 1
IF (DELTA .LT. 0) OCT = OCT + 4
IF (DELTA .LT. 0) OCT = OCT + 2
IF (DIFF .LT. 0) OCT = OCT + 1
IF (OCT .EQ. OCT/2*2) GO TO 10
DELTA = IABS(DELTA)
DELTAB = IABS(DELTA)
GO TO 20
10 DELTAA = IABS(DELTA)
DELTAB = IABS(DELTA)
20 BETA = 2 * DELTAB
ALPHA = 2 * (DELTAA - DELTAB)
NABLA = BETA - DELTAA
CALL SPOT (IXXA,IYYA,COLOR)
MOVE1 = M1(OCT)
MOVE2 = M2(OCT)
30 IF (NABLA .LT. 0) GO TO 40
IXXA = IXXA + XSTEP(MOVE2)
IYYA = IYYA + YSTEP(MOVE2)
NABLA = NABLA - ALPHA
GO TO 50
40 IXXA = IXXA + XSTEP(MOVE1)
IYYA = IYYA + YSTEP(MOVE1)
NABLA = NABLA + BETA
50 CALL SPOT (IXXA,IYYA,COLOR)
DELTAA = DELTAA - 1
IF (DELTAA .NE. 0) GO TO 30
RETURN
END

```

```

SUBROUTINE SPOT (X,Y,C)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(130),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
IF (X .NE. PREVX .OR. Y .NE. PREVY) GO TO 10
IF (C .NE. PREVY) GO TO 10
RETURN
10 R = X / 128 + 1
IF (R .EQ. CURR) GO TO 20
WRITE (UNIT,CURR) AREA
DO 30 I = 1, RCOUNT
IF (RECORD(I) .EQ. R) GO TO 40
30 CONTINUE
RCOUNT = RCOUNT + 1
RECORD(RCOUNT) = R
DO 50 I = 1, 330
DO 50 J = 1, 8
50 AREA(I,J) = 0
60 CURR = R

```

```

20 CALL SPOT1 (MOD(X,128),Y,C)
PREVX = X
PREVY = Y
PREVC = C
RETURN
40 READ (UNIT,R) AREA
GO TO 60
END

```

```

SUBROUTINE SPOT1 (X,Y,C)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(130),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
DIMENSION CNST(16)
DATA CNST/
* 0d37777777777777,0d07777777777777,0d37777777777777,
* 0d37777777777777,0d37777777777777,0d37777777777777,
* 0d37777777777777,0d37777777777777,0d37777777777777,
* 0d37777777777777,0d37777777777777,0d37777777777777,
* 0d36377777777777,0d31777777777777,0d07777777777777/
ROW = Y + 1
COL = X / 16 + 1
POS = 15 - MOD(X,16)
AREA(ROW,COL) =
* AND(AREA(ROW,COL),CNST(POS+1)) + C*4*POS
PLOTSW = 1
RETURN
END

```

```

SUBROUTINE DISPLY
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(330,0),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREYV,PREVC,CURR,
* UNIT,RCOUNT, PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
COMMON /E/ CHAR(128),R
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
LOGICAL PRNTSW
DIMENSION DATA(4)
DATA DATA/1H',1H*',1H0,1H"/
IF (PLOTSW .EQ. 0) GO TO 999
IF (RCOUNT .EQ. 1) GO TO 10
RR = RCOUNT - 1
DO 30 I = 1, RR
DO 30 J = 1, RR
IF (RECORD(I) .LE. RECORD(J+1)) GO TO 30

```

```

WORK = RECORD(I)
RECORD(I) = RECORD(J+1)
RECORD(J+1) = WORK
30 CONTINUE
10 CALL BETA
DO 500 RSEQ = 1, RCOUNT
R = RECORD(RSEQ)
READ (UNIT'R') AREA
ROW = 0
DO 400 PAGE = 1,5
PRNTSW = .FALSE.
DO 400 LINE = 1,66
ROW = ROW + 1
DO 50 COL = 1,8
IF (AREA(ROW,COL) .NE. 0) GO TO 60
50 CONTINUE
IF (.NOT. PRNTSW) GO TO 400
CALL FRAME (ROW)
IF (LINE .NE. 2) GO TO 360
WRITE (6,70) R,CHAR
70 FORMAT (2H C,12,128A1)
GO TO 400
360 IF (LINE .NE. 66) GO TO 90
WRITE (6,100) (CHAR(I),I=2,127)
100 FORMAT (1H ,3X,1HV,126A1,1HV)
GO TO 400
90 WRITE (6,110) CHAR
110 FORMAT (1H ,3X,128A1)
GO TO 400
60 IF (PRNTSW) GO TO 80
PRNTSW = .TRUE.
IF (LINE .EQ. 1) GO TO 80
CALL FRAME ((PAGE-1)*66+1)
WRITE (6,120) PAGE,(CHAR(I),I=2,127)
120 FORMAT (2H R,12,1H",126A1,1H")

IF (LINE .EQ. 2) GO TO 80
CALL FRAME ((PAGE-1)*66+2)
WRITE (6,70) R,CHAR
IF (LINE .EQ. 3) GO TO 80
NLINE = LINE - 1
DO 130 I = 3,NLINE
CALL FRAME ((PAGE-1)*66+1)
WRITE (6,110) CHAR
130 CONTINUE
80 CALL FRAME (ROW)
DO 180 K = 1,8
DATA = AREA(ROW,K)
CPNTR = (K-1) * 16
DO 180 KK = 1,16
CCOLOR = FLD(2*KK+2,DATA) + 1
CSAVE = CDATA(CCOLOR)
IF (CSAVE .EQ. CDATA(I)) GO TO 180
CHAR(CPNTR+KK) = CSAVE
180 CONTINUE
IF (LINE .NE. 1) GO TO 140
WRITE (6,120) PAGE,(CHAR(I),I=2,127)
GO TO 400
140 IF (LINE .NE. 2) GO TO 150
WRITE (6,70) R,CHAR
GO TO 400
150 IF (LINE .NE. 66) GO TO 160
WRITE (6,100) (CHAR(I),I=2,127)
GO TO 400
160 WRITE (6,110) CHAR
400 CONTINUE
500 CONTINUE
999 CALL NBETA
RETURN
END

SUBROUTINE FRAME (ARG)
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(300,8),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREVY,PREVC,CURR,
* UNIT,RCOUNT,PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
COMMON /E/ CHAR(128),R
DATA BLANK,MINUS,PLUS,COLON/1H-,1H.,1H+,1H:/
DO 190 K = 1,128
190 CHAR(K) = BLANK
IF (R .EQ. RLOWX) GO TO 200
IF (R .EQ. RUPX) GO TO 210
IF (R .LT. RLOWX .OR. R .GT. RUPX) RETURN
IF (ARG .NE. UPY .AND. ARG .NE. LOWY) RETURN
DO 220 X = 1,128
220 CHAR(K) = MINUS
RETURN
200 IF (ARG.LT.UPY .OR. ARG.GT.LOWY) RETURN
IF (ARG.EQ.UPY .OR. ARG.EQ.LOWY) GO TO 230
CHAR(LOWX) = COLON
RETURN
230 DO 240 K = LOWX,128
240 CHAR(K) = MINUS
CHAR(LOWX) = PLUS
RETURN
210 IF (ARG.LT.UPY .OR. ARG.GT.LOWY) RETURN
IF (ARG.EQ.UPY .OR. ARG.EQ.LOWY) GO TO 250
CHAR(UPX) = COLON
RETURN
DO 260 K = 1,UPX
260 CHAR(K) = MINUS
CHAR(UPX) = PLUS
RETURN
END

```

```

BLOCK DATA
IMPLICIT INTEGER(A-Z)
COMMON /D/ AREA(300,8),RECORD(100),LOWX,UPX,
* LOWY,UPY,RLOWX,RUPX,PREVX,PREVY,PREVC,CURR,
* UNIT,RCOUNT,PLOTSW,SCAL,CURX,CURY,COLOR,
* CONV,XOFF,YOFF
REAL CURX,CURY,SCAL,CONV,XOFF,YOFF
DATA LOWX,UPX,LOWY,UPY/10,117,200,84/
DATA RLOWX,RUPX,CONV/2,100,1.0/
END

```

Fig. 1

```

DIMENSION BUF(1000)
CALL PLOTS(BUF,1000,10)
CALL FACTOR(1./2.54)
CALL PLOT(1.,2.,-3)
CALL PLOT(5.,2.,2)
CALL PLOT(3.,6.,2)
CALL PLOT(0.,0.,2)
CALL SYMBOL(1.,7.,1.,6HTRIANGLE,0.,8)
CALL PLOT(0.,0.,999)
STOP
END

```

Fig. 2 プログラム例

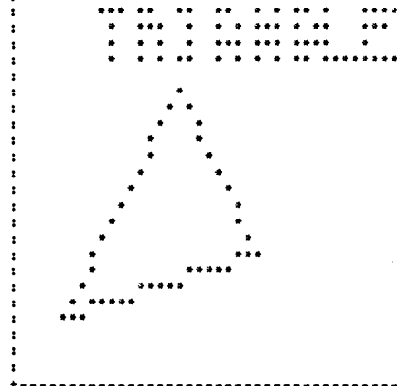


Fig. 3 プリンタ出力

TRIANGLE

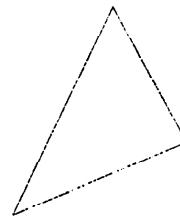


Fig. 4 プロッタ出力

のプログラムからプリンタ出力を行なった例(Fig. 3)と、プロッタ用パッケージによってプロッタに出力した例(Fig. 4)を示す。

参考文献

- 1) J.E. Bresenham: "Algorithm for computer control of a digital plotter", IBM Systems Journal, Vol. 4, No. 1 (1965).

(昭和50年5月27日受付)