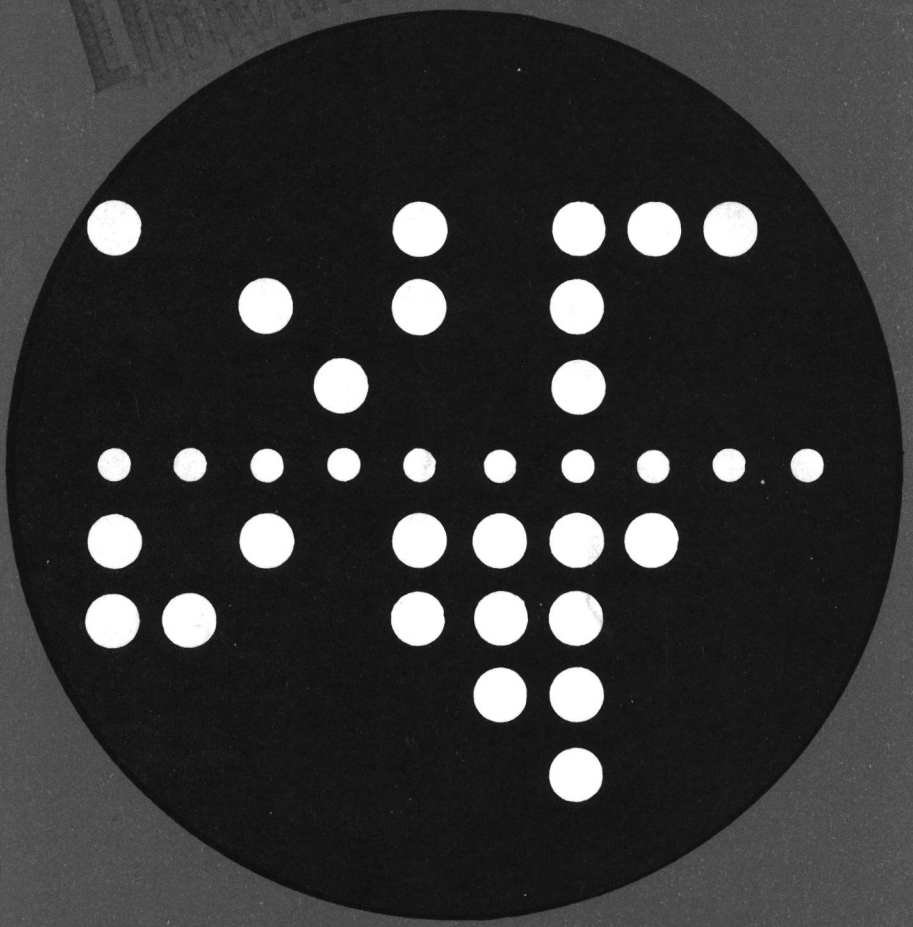
COMPUTING CENTRE NEWSLETTER

GRAPHIT



Commission of the European Communities
JOINT RESEARCH CENTRE
Ispra Establishment

SPECIAL ISSUE

GRAPHIT:

A DEVICE INDEPENDENT GRAPHIC OUTPUT SYSTEM

Herman I. de Wolde, Aurelio A. Pollicini

December 1978

SUMMARY

The GRAPHIT system has been designed to make different plotting devices accessible for computer programs with graphic output for one particular device.

The present version has been written for a large collection of scientific and technical programs which are using the subroutine library for a Calcomp 1136 penplotter. Through GRAPHIT, these programs have now also access to a GOULD 5200 electrostatic plotter and a BENSON 1102 penplotter.

The connection to a Tektronix 4015 will be developed later. Any other device may be linked to the system by a simple FORTRAN program, for which the flowscheme is given in this report.

INTRODUCTION

The basic idea of GRAPHIT is to store the calls from the main program to graphic subroutines in an Intermediate File and to interprete these calls afterwards when a particular plotting device has been chosen. To achieve this, the actual plotting routines are replaced by dummy routines which take care of the link to the Intermediate File.

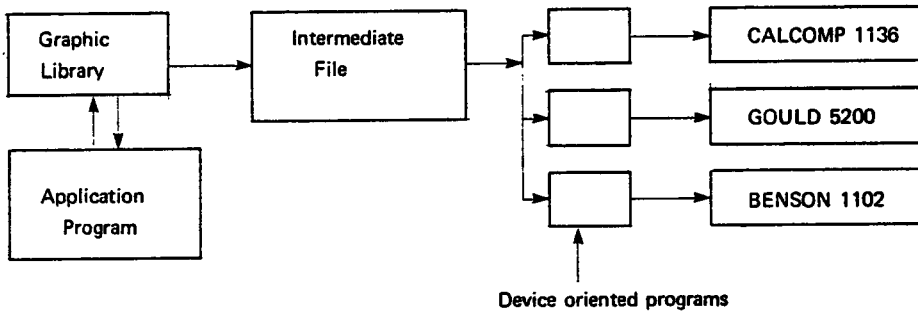


Fig.1

The Intermediate File has a compressed format. The new plotting routines in the graphic library pass the calls from the main program to a routine called BUFFER, which compresses the data and, when the buffer is complete, writes the record to the Intermediate File.

Reversely, the routine READB decompresses the data and informs the device oriented programs on the tasks which have to be performed.

Some devices, like the CALCOMP 1136, use hardware characters. For the time being, this has been solved by transfering the hardware character callings to the software character generator.

However the definition of the Intermediate File, as will be explained hereafter, offers enough room to treat each character request as a separate call.

The whole design of GRAPHIT is based on flexibility and trasparency, which means that not always the most economic solution has been chosen.

STRUCTURE OF THE INTERMEDIATE FILE

The basic elements of the Intermediate File are expressed in bytes. The shortest element is one byte. Other functions need up to nine bytes. The general configuration is as follows:

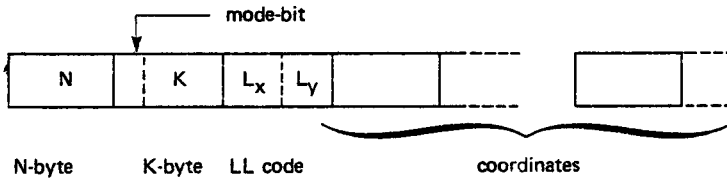


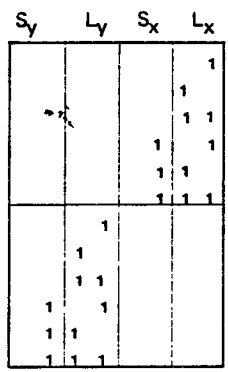
Fig.2

The N-byte contains a fixed number which indicates the particular meaning of the function and defines as such also the number of associated bytes. There are thus 255 different functions possible of which, for the here presented configuration, only 5 are used.

* N=1 The function defines a penmovement from the present position to a position as specified by the x,y coordinates.
The first bit of the K-byte is called the "mode-bit".
If the mode bit is zero, the coordinates are specified incremental.
If the mode bit is one, the coordinates are given as absolute values in relation to the lower-left angle of the drawing surface.
The value of K, without the mode-bit, defines the pen mode:

K=2 pen down (writing)
K=3 pen up (no writing)
K=0 no modification in pen mode

The LL code in the third byte defines the number of bytes for respectively the x coordinates and y coordinates. If the coordinates are defined as incremental (mode bit in K byte is zero), the LL code contains also the sign according to the following table



1	1 byte	x	pos
2	2 bytes	x	pos
3	3 bytes	x	pos
5	1 byte	x	neg
6	2 bytes	x	neg
7	3 bytes	x	neg
16	1 byte	y	pos
32	2 bytes	y	pos
48	3 bytes	y	pos
80	1 byte	y	neg
96	2 bytes	y	neg
112	3 bytes	y	neg

Fig.3

The following bytes contains either the absolute x and y coordinates or the incremental distances.

The coordinates are specified as integer numbers with an accuracy of 0.001 cm. To save space and maintain the accuracy, one in ten values is given as absolute value and the following nine are specified incremental. However this ratio may be changed easily; see the description of the routine BUFFER.

* N=3 Defines the end of the Intermediate File. This function does not contain other arguments.

* N=4 Closes one buffer record. This function does not contain other arguments.

* N=6 Other pen colour is wanted. The K byte contains the number of the newpen.

* N=7 Initiates a new picture segment. The following K byte contains the number of the segment, K< 256. This feature allows for partial graphic output. One may instruct the device oriented programs to draw only specific pictures from the Intermediate File.

Any other functions can be added with other positive values of N, with the limitation that a complete set of values is of the format:

N,X,Y,K

in which

N and K are of integer*4 type and
X and Y are of real*4 type

Depending on the significance of the N value, some of the other arguments may be omitted.

Additional options, specified by N, need modifications of the routines BUFFER and READB.

The compressing of the data as explained before, is performed by the routine BUFFER.
Presently the Intermediate File is composed of records of 800 bytes. Each record is ended with one byte of value 4. The Intermediate File recognizes only one coordinate system; the (0,0) point is always the lower-left point of the drawing surface and cannot be replaced.
The units of lenghts are centimeters.
The maximum field for a single distance is 3 bytes, which corresponds to a maximum value of 16777 cm, that limits the size of the drawings in both directions.

SUBROUTINE BUFFER

The subroutine BUFFER takes care of the transfers from the calling routines to the Intermediate File and compresses the data according to the previous description.
The present version gives error messages and halts the executions of the job for:

```
N<0
N=2
N=5
N>7
```

Consequently, any of these values may be used for future options. The calling sequence to the routine is:

```
CALL BUFFER (N,X,Y,K)
```

in which some of the arguments may be without significance, depending on the value of N.

The present version is based on records of 800 bytes. The length of the records may be changed by modifying the statements:

```
DIMENSION NBUFF(200),LB(800)
DATA LBUFF,LLB/200,792/
```

in which:

```
LLB=4*LBUFF-8
```

An equivalent correction has to be applied in the routine READB, which reads the Intermediate File. Presently, BUFFER writes to the Intermediate File the coordinates one time in ten as absolute values and nine times in ten, the incremental values.
This feature has been included to save space and maintain the accuracy.
However, other proportions may be defined by modifying the statement:

```
DATA    COUNT,NCOUNT/10,10/
```

For example, if one defines these values both equal to one, the Intermediate File contains only absolute coordinates.
The routine READB does not require any related modifications.

SUBROUTINE READB

The subroutine READB reads the Intermediate File and supplies the calling program each time, with a series of arguments in the form:

N, X, Y, K

in which not all the arguments are always present. However, if the coordinates X and Y are specified, these are always given as absolute values; the incremental values are converted into distances in relation to the lower-left point of the drawing surface.
The routine keeps track of the last pen position. The coordinates of this point are always represented by XOLD and YOLD.

If during the creation of the Intermediate File a program or system failure occurs, resulting in an abnormal closing of the file, READB will try to correct this by inserting a normal exit. This feature allows for the recuperation of partial graphic output.

SUBROUTINE PICTNO

A new routine called PICTNO is available to accomodate the users of a permanent Intermediate File who want to apply segmentation of the drawings.
One may, for example, produce a large number of drawings on the quick Gould plotter, then select a few elements on the accurate Calcomp and Benson plotters.
The calling statement to this routine is:

```
CALL PICTNO(K)
```

in which K is a consecutive element number.
PICTNO is called before the related segment.
The programmer has to take care that each of the pictures has sufficient drawing space available when previous pictures are omitted.
The easiest way is to organize the drawings in a series of rectangles and to place the pen at the lower left corner of each following segment before calling PICTNO(K).

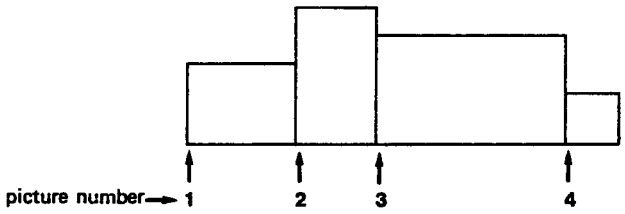


Fig.4

A more sophisticated routine will be developed later.

DEVICE ORIENTED PROGRAMS

Each plotting device needs a separate main program with the following functions:

- Read special input depending on the type of device. For example: scale factors, line weights, erase option
- Read requested segment numbers
- Read the Intermediate File through the routine READB
- Analyse each task element and decide if it is relevant for the particular device
- Apply multiplication factors
- Skip not requested elements
- Close the graphic output file
- Submit messages

All the different main programs have the same structure as is illustrated in the following flowscheme.

Any additional option, specified by a positive value of N other than 1, 3, 4, 6 or 7 may be inserted at the indicated place, without disturbing the general flow of the main program. However care must be taken that in the case of a real pendisplacement of a deleted segment, the values XDEL and YDEL have to be corrected.

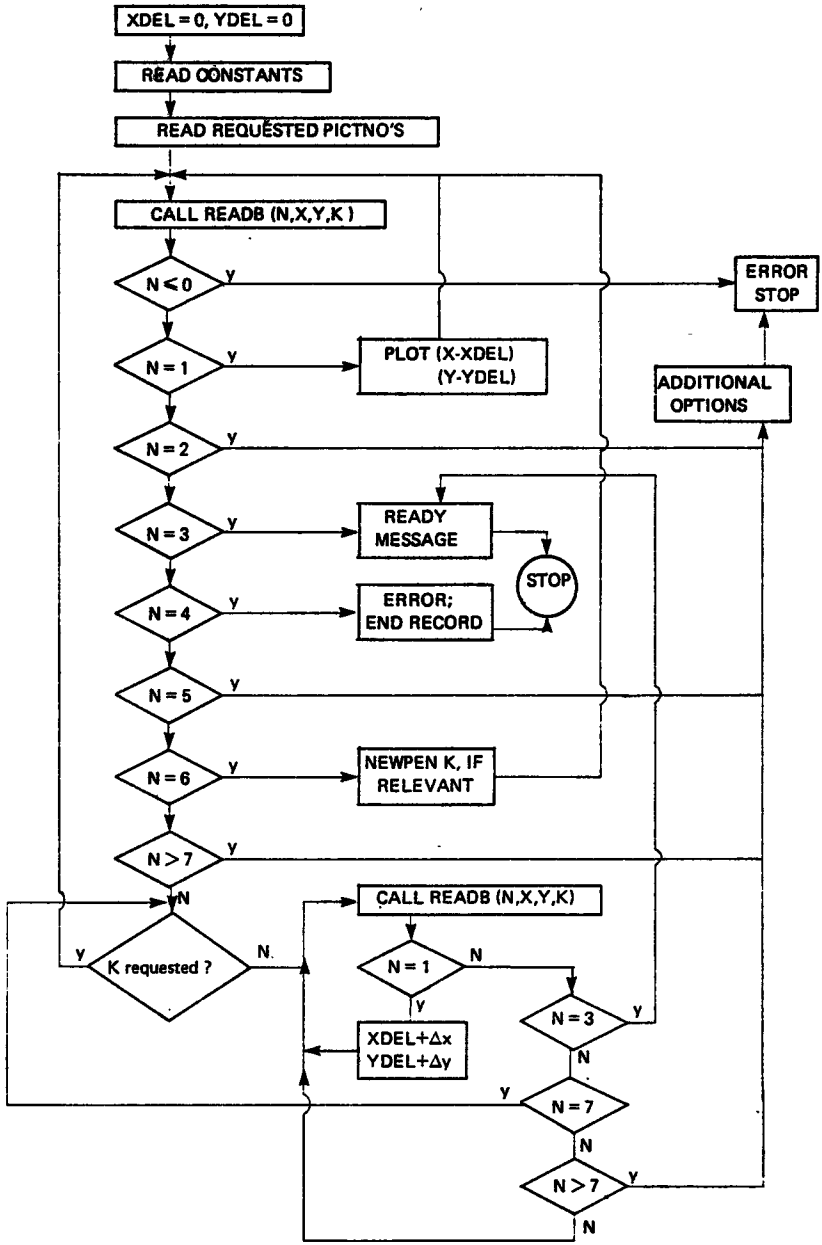

XDEL = 0, YDEL = 0
READ CONSTANTS
READ REQUESTED PICTNO'S
CALL READB (N,X,Y,K)
N ⩽ 0
y
ERROR STOP
N = 1
y
PLOT (X-XDEL) (Y-YDEL)
ADDITIONAL OPTIONS
N = 2
y
N = 3
y
READY MESSAGE
STOP
N = 4
y
ERROR; END RECORD
N = 5
y
N = 6
y
NEWPEN K, IF RELEVANT
N > 7
y
N
CALL READB (N,X,Y,K)
y
K requested ?
N.
N = 1
N
y
XDEL+Δx YDEL+Δy
N = 3
y
N
y
N = 7
N
N > 7
y
N

Fig.5

THE APPLICATIONS OF GRAPHIT

This chapter describes shortly the applications of GRAPHIT, together with the necessary control cards.
The here specified configuration is only valid for the computer installation of the JRC, Ispra Establishment.
Modification and streamlining of the procedures might be necessary in the future.

The present plotting devices are:

- Calcomp 1136 3 pens
  axial penmovement about 9 cm/sec
  increment size 0.05 mm
  plot size in y-direction 83 cm

- Benson 1102 3 pens
  axial penmovement 5 cm/sec
  increment size 0.05 mm
  plot size in y-direction 32 cm

- Gould 5200 electrostatic plotter
  200 points/inch, resolution 0.13 mm
  linear paper speed 4 cm/sec
  plot size in y-direction 28 cm

From these specifications it is clear that the Gould plotter is most fit for test runs: The device is very quick but the accuracy is not very high. The turnaround time is favoured by a cutting device which makes the drawings immediately available.
The Benson plotter may be used for accurate drawings of limited size as the plotting area is only 32 cm high. This size is sufficient for a large part of the production and it has the advantage that the user needs not to bother about economic paper use in the y-direction.
For the time being, all these plotters are used off-line.

The next diagram gives the relations between the hardware and software components. The letters point to the related paragraphs.

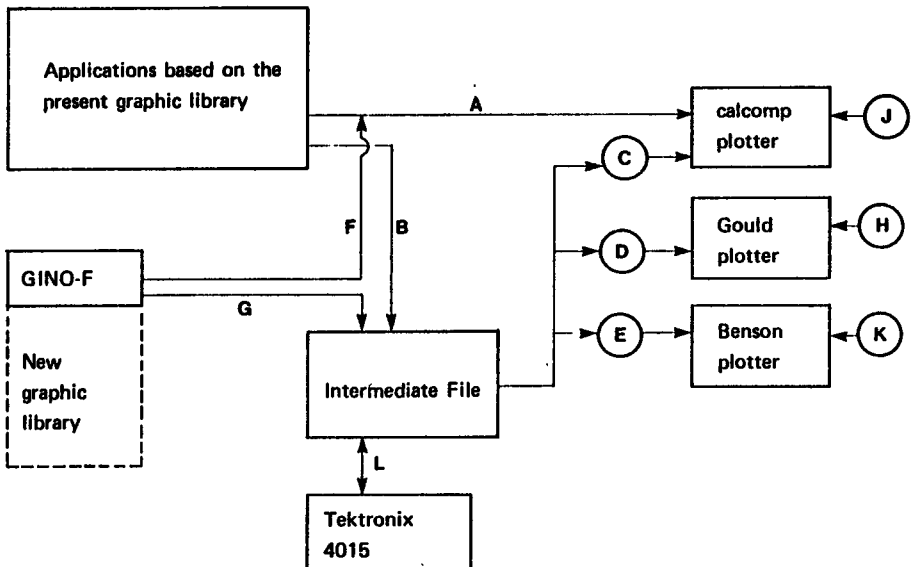


Fig.6 GRAPHIT SYSTEM

A. Existing applications directly to Calcomp

The Calcomp plotter may be used as normal, without the services of the intermediate file. The system library SYS1.EURLIB contains the calcomp routines as are described in the manual which may be obtained from the Installation Notes by the job:

```
//..... job .....
$     LINES 003
//    EXEC LIHNO,MEMB=PCAL
```

The output file is created on private tape as always. The DD statement for this file is:

```
//GO.FT16F001 DD UNIT=TP9,LABEL=(,,,OUT),DISP=(NEW,PASS),
//    VOLUME=(PRIVATE,SER=EUxxxx),DSNAME=PICT,
//    DCB=(RECFM=VS,BLKSIZE=488,LRECL=484,DEN=2)
```

in which EUxxxx is the tapenumber.

B. Existing applications to Intermediate File

The graphic output of programs which are using the standard Calcomp calling modes may be stored in an intermediate file. This file has to be created by the user and may be temporary or permanent and may be stored on disk or tape. If the file is temporary normally on scratch disk, the preparation of the graphic data, for a specific device, has to be executed in a second jobstep (see par. C). The next example gives the deck composition for the creation of a temporary intermediate file on scratch disk:

```
//    EXEC FTGCLG,PRM=GRAPH,VLB=COPICA,ULB=DISK
//CMP.SYSIN DD *
        fortran deck
//GO.FT16F001 DD DSN=&INTERF,SPACE=(CYL,(2,2)),
//    DISP=(NEW,PASS),UNIT=SYSDA,
//    DCB=(RECFM=VS,LRECL=800,BLKSIZE=804)
//GO.SYSIN DD *
        input data
/*
```

The called procedure may also be FTL or FTLG. Programs which exist as load modules have to be re-linkedited. The plot command routines PLOT, FINTRA, FACTOR, FINIM, WHERE, NEWPEN and SYMBL4 are replaced by pseudo routines with the same names out of the library SYS1.LIBGRAPH. These pseudo routines take care that the plot commands are stored, through the routine BUFFER, in the Intermediate File. Normally this job step has to be followed by a second step which creates the devices oriented tape file. (See par. C).

The following example gives the DD statement for a permanent file on a private tape, with density 3:

```
//GO.FT16F001 DD DSN=dsname,DISP=(NEW,PASS),UNIT=TP9,
//    LABEL=(1,SL),VOL=(PRIVATE,SER=EUyyyy),
//   DCB=(RECFM=VS,LRECL=800,BLKSIZE=804)
```

Problems concerning the scales of the drawings when programming for the Calcomp plotter and testing an other device are solved on the level of the device oriented programs (see D and E). When a system or program failure occurs during the creation of the Intermediate File, the system closes the file with an EOF and the existing part may still be plotted.

C. Intermediate File to Calcomp plotter

If the intermediate file is temporary and will be processed in the same job as it was created, the control cards are:

```
//STEP2 EXEC CALCOMP,TAPE='EUxxxx'
//CLP.SYSIN DD *
        picturenumbers which are read as (12I6)
/*
```

in which EUxxxx is the output tape identification.

The input data give the number of the pictures which are to be processed. These numbers need not to be oriented and may contain blank fields. If you want to draw all the images or when the picturenumbers are not defined, it is sufficient to insert a blank card here.
If the Intermediate File has been stored on tape, the card configuration becomes:

```
//     EXEC CALCOMP,TAPE='EUxxxx'
//CLP.FT15F001 DD DSN=dsname,DISP=(OLD,PASS),
//    UNIT=TP9,LABEL=(1,SL,,IN),VOL=(PRIVATE,SER=EUyyyy)
//CLP.SYSIN DD *
        picturenumbers
/*
```

D. Intermediate File to Gould plotter

If the Intermediate File has been defined as temporary and will be processed in the same job as it was created, the control cards become:

```
//STEP2    EXEC GOULD,TAPE='EUxxxx'
//GLD.SYSIN DD *
FACTOR,IFORM,NCOPY,INT,MAX (F6.2,4I6)
INP(i)                     (12I6)
/*
```

in which

FACTOR is the scale factor. If not specified the program takes a default value of 0.32. This is the ratio between the paper height of the Gould plotter and the Calcomp plotter.

IFORM is the background pattern number
= 0 blank background
= 1 squared background
=-1 black background with erased lines

NCOPY is the number of copies wanted

INT is the line intensity
=-2 triple
=-1 double
= 0 normal
= 1 half
= 2 one-third

MAX is the maximum length of the plot in x direction

INP(i) are the desired drawing segments. If omitted all segments will be processed.

If the Intermediate File has been stored on tape, the card configuration becomes:

```
//      EXEC GOULD,TAKE='EUxxxx'
//GLD.FT15F001 DD DSN=dsname,DISP=(OLD,PASS),UNIT=TP9,
                LABEL=(1,SL,,IN),VOL=(PRIVATE,SER=EUyyyy)
//GLD.SYSIN DD *
          data
/*
```

in which

EUyyyy is the identification of your Intermediate File tape
EUxxxx is the identification of the output file tape.

E. Intermediate File to Benson plotter

The temporary Intermediate File may be processed in the same job as it was created, with a tape file output for the Benson plotter, by the following jobstep:

```
//STEP2   EXEC BENSON,TAPE='EUxxxx'
//BNS.SYSIN DD *
FACTOR          (F6.2)
INP(i)          (12I6)
/*
```

in which

FACTOR is the scale factor, if omitted the program multiplies all the input values with 0.37 being the ratio between the Benson plotter y-size and the Calcomp plotter y-size

INP(i) are the desired picture seqments. If omitted the program processes the complete intermediate file.

If the Intermediate File is a permanentone stored on tape, the job becomes:

```
//      EXEC BENSON,TAPE='EUxxxx'
//BNS.FT15F001 DD DSN=dsname,DISP=(OLD,PASS),UNIT=TP9,
//             LABEL=(1,SL,,IN),VOL=PRIVATE,SER=EUyyyy)
//BNS.SYSIN DD *

         data
/*
```

F. GINO-F to Calcomp Plotter

The subroutines of the GINO-F package are available for direct drawing on the Calcomp plotter by the next job:

```
//   EXEC FTGCLG,PRN=GINOF,VLB=COPICA,ULB=DISK
//CMP.SYSIN DD *
         fortran deck
/*
//GO.FT16F001 DD UNIT=TP9,LABEL=(,,,OUT),DISP=(NEW,PASS),
//   VOLUME=(PRIVATE,SER=EUxxxx),DSNAME=PICT,
//   DCB=(RECFM=VS,BLKSIZE=488,LRECL=484,DEN=2)
//GO.SYSIN DD *
       input data
/*
```

G. GINO-F to Intermediate File

The link of the GINO-F package to the Intermediate File is available but it will be explained in a following report on the Extended Graphic Library.

J. Calcomp Subroutine Library

The Standard Calcomp subroutine library remains available as it has been described in paragraph A.

H. Gould Subroutine Library

The use of the Gould Subroutine Library will be limited to its function as a back-end to the Intermediate File. Only in special cases a user may request for a direct access to this library.
The special functions of the Gould plotter, for example the surface grey levels, will be embedded in the general graphics library.

K. Benson Subroutine Library

Also this library is generally not accessible for the users; the use of the Benson plotter is only possible through the Intermediate File.

L. Tektronix and the Intermediate File

It is planned to apply the Tektronix display for the scannig of the Intermediate File.
However this link is not yet available.

SURVEY OF LIBRARIES

The GRAPHIT system makes use of the following libraries:

SYS1.EURLIB which contains all the present Calcomp routines
SYS1.LIBGRAPH which contains:
BUFFER(*), PLOT(*) (with entries FACTOR, FINIM, WHERE), FINTRA(*), NEWPEN(*), PICTNO(*), SYMBL4(*)
SYS1.LIBPROGM which contains:
READB(*), CALCO(*), BENSON(*), GOULD(*)
SYS1.LIBBENS which contains the BENSON library
SYS1.LIBGOULD which contains the GOULD library

(*) The routines marked with an asterisk are listed in the appendix.

```
$     TIME 002
$     LINES 002
$     CLASS 2
//STEPA        EXEC FTGC
//CMP.SYSIN    DD *
      SUBROUTINE READB(N,X,Y,K)
C
C     VERSION WITH COMPRESSED STORAGE - JAN.1978 BY HERMAN I.DE WOLDE
C     THE CONSTRUCTION OF THE BUFFER IS EXPLAINED
C     IN THE COMMENTS OF ROUTINE 'BUFFER'.
C
      DIMENSION NBUFF(200),LB(800),KB(4)
      LOGICAL*1 LB,KB
      EQUIVALENCE (NBUFF(1),LB(1)),(KB(1),KBFULL)
      DATA LBUFF,LLB,IBUF/200,792,0/
      DATA XOLD,YOLD/0.0,0.0/
C
C
      IF(IBUF.GT.0) GO TO 100
      READ (15,END=250) (NBUFF(I),I=1,200)
      IBUF=1
C
  100 KBFULL=0
      KB(4)=LB(IBUF)
      IBUF=IBUF+1
      N=KBFULL
      IF(N.GT.0) GO TO 125
C
C-----ERROR EXIT----------------------------------------------------------
C
  110 WRITE (6,120) N
  120 FORMAT (1H1/' ERROR AT READING BUFFER N= ',I10/)
      WRITE(6,122)
  122 FORMAT(//' HEXADECIMAL CONTENTS OF ERRONEOUS RECORD'//)
      WRITE(6,123) (NBUFF(I),I=1,200)
  123 FORMAT(20(10X,10(Z8,' ')))
      STOP
C
C
C-----N=1 CALCULATE ABSOLUTE COORDINATES X,Y-------------------------
C
  125 IF(N.GT.1) GO TO 220
      KBFULL=0
      KB(4)=LB(IBUF)
      IBUF=IBUF+1
      K=KBFULL
      ICO=0
      IF(K.LT.128) GO TO 126
      ICO=1
      K=K-128
  126 CONTINUE
      KBFULL=0
      KB(4)=LB(IBUF)
      IBUF=IBUF+1
      LLCODE=KBFULL
      IY=1
```

```
      KY=0
      IF(LLCODE.LT.80) GO TO 130
      IY=(-1)
      LLCODE=LLCODE-64
  130 IF(LLCODE.LT.32) GO TO 140
      KY=2
      LLCODE=LLCODE-32
  140 IF(LLCODE.LT.16) GO TO 150
      KY=KY+1
      LLCODE=LLCODE-16
  150 IX=1
      IF(LLCODE.LT.4) GO TO 160
      IX=(-1)
      LLCODE=LLCODE-4
  160 KX=LLCODE
C
C-----SECURITY CHECK-------------------------------------------------
C
      IF(KX.LT.1.AND.KX.GT.3) GO TO 170
      IF(KY.GT.0.AND.KY.LT.4) GO TO 190
  170 WRITE (6,180) KX,KY
  180 FORMAT (1H1/' ERROR IN READB KX=',I5,' KY=',I5/)
      STOP
C
  190 KBFULL=0
      DO 200 I=1,KX
      IA=4+I-KX
      KB(IA)=LB(IBUF)
  200 IBUF=IBUF+1
      IF(ICO.EQ.0) GO TO 202
      X=0.001*FLOAT(KBFULL*IX)
      GO TO 204
  202 CONTINUE
      X=0.001*FLOAT(KBFULL*IX)+XOLD
  204 CONTINUE
      XOLD=X
      KBFULL=0
      DO 210 I=1,KY
      IA=4+I-KY
      KB(IA)=LB(IBUF)
  210 IBUF=IBUF+1
      IF(ICO.EQ.0) GO TO 212
      Y=0.001*FLOAT(KBFULL*IY)
      GO TO 214
  212 CONTINUE
      Y=0.001*FLOAT(KBFULL*IY)+YOLD
  214 CONTINUE
      YOLD=Y
      RETURN
C
C-----N=3 CLOSES FILE------------------------------------------------
C
  220 IF(N.EQ.2) GO TO 110
      IF(N.EQ.3) RETURN
C
C-----N=4 READ NEW BUFFER--------------------------------------------
```

```
C
      IF(N.GT.4) GO TO 230
      READ (15,END=250) (NBUFF(I),I=1,200)
      IBUF=1
      GO TO 100
C
C-----N=6 NEW PEN-----------------------------------------------------------
C
  230 IF(N.EQ.5) GO TO 110
      IF(N.GT.6) GO TO 240
      KBFULL=0
      KB(4)=LB(IBUF)
      IBUF=IBUF+1
      K=KBFULL
      RETURN
C
C-----N=7 PICTURE NUMBER----------------------------------------------------
C
  240 IF(N.GT.7) GO TO 110
      KBFULL=0
      KB(4)=LB(IBUF)
      IBUF=IBUF+1
      K=KBFULL
      RETURN
C
C-----IN CASE OF EOF, READB SETS N=3 AND RETURNS----------------------------
C
  250 WRITE(6,260)
  260 FORMAT (1H1/' ERROR CONDITION EOF ON INTERMEDIATE FILE'/
     1' THE SYSTEM GRAPHIT TRIES TO SAVE THE CURRENT OUTPUT'/
     2' PLEASE CHECK THE OCCURENCE OF FINAL MESSAGE'/
     3' REGULAR EXIT.....')
      N=3
      RETURN
      END
/*
//STEPB         EXEC FTL,NC=NCAL
//LKED.SYSLMOD DD DSN=SYS1.LIBPROGM,UNIT=DISK,VOL=SER=COPICA,
//              DISP=(OLD,KEEP)
//LKED.SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)
//              DD *
  NAME READB(R)
/*
$     TIME 002
$     LINES 003
$     CLASS 2
//              EXEC FTGC
//CMP.SYSIN     DD *
C-----PROGRAM CALCOM BY HERMAN I. DE WOLDE-----MAY 1977----------------
C
C     CALCOM READS THE INTERMEDIATE GRAPHIC FILE
C     AND TRANSLATES THE ARGUMENTS INTO ACTUAL SUBROUTINE CALLS
C
C     THE PRESENT VERSION RECOGNIZES THE FOLLOWING SITUATIONS
C
C     N=1 CALL PLOT(X,Y,K)
```

```
C      N=2 ERROR CONDITION
C      N=3 CALL FINTRA
C      N=4 CLOSES ONE BUFFER RECORD
C      N=5 ERROR
C      N=6 CALL NEWPEN(K)
C      N=7 CALL PICTNO(K)
C      N=GT.7 ERROR
C
       DIMENSION NPIC(100),INP(12)
C      XDEL AND YDEL ARE THE SUM OF ALL DELETED PICTURES
       DATA XDEL,YDEL/0.0,0.0/
       NREC=0
C
C-----READ THE PICTURE NUMBERS, IF ANY.--------------------------------
C
       NAP=0
    40 READ (5,50,END=70) (INP(I),I=1,12)
    50 FORMAT (12I6)
       DO 65 I=1,12
       IF(INP(I).LE.0) GO TO 65
       IF(NAP+1).LE.100) GO TO 60
       WRITE (6,55)
    55 FORMAT (1H1/' REQUESTED NUMBER OF PICTURES.GT.100'/
      1' ONLY FIRST HUNDRED WILL BE DONE.'//)
       GO TO 70
    60 NAP=NAP+1
       NPIC(NAP)=INP(I)
    65 CONTINUE
       GO TO 40
    70 CONTINUE
       IF(NAP.LE.0) GO TO 90
       WRITE(6,80) (NPIC(I),I=1,NAP)
    80 FORMAT (1H1/' LIST OF REQUESTED PICTURES'/(20I6)/)
    90 CONTINUE
C
C-----READ BUFFER------------------------------------------------------
C
   100 CALL READB(N,X,Y,K)
       IF(N.GT.0) GO TO 120
   105 WRITE (6,110) N,NREC
   110 FORMAT (1H1/' ERROR MAIN CALCOM N=',I10/' NUMBER OF CALLS NREC=',I
      110/)
       STOP
C
   120 IF(N.GT.1) GO TO 130
       CALL PLOT(X-XDEL,Y-YDEL,K)
       XPIC=X
       YPIC=Y
       NREC=NREC+1
       GO TO 100
C
   130 IF(N.EQ.2) GO TO 105
   140 IF(N.GT.3) GO TO 150
       NREC=NREC+1
       CALL FINIM(0.0,0.0)
       CALL FINTRA
```

```
      GO TO 220
C
  150 IF(N.EQ.4) GO TO 105
      IF(N.EQ.5) GO TO 105
C
  160 IF(N.GT.6) GO TO 170
      NREC=NREC+1
      CALL NEWPEN(K)
      GO TO 100
C
C
C-----K IS PICTURE NUMBER-----------------------------------------------------
C
  170 IF(N.GT.7) GO TO 105
      XPICA=XPIC
      YPICA=YPIC
      XPICB=XPIC
      YPICB=YPIC
  180 IF (NAP.EQ.0) GO TO 100
  190 DO 200 I=1,NAP
      IF(K.EQ.NPIC(I)) GO TO 217
  200 CONTINUE
  210 CALL READB(N,X,Y,K)
      IF(N.NE.1) GO TO 215
      XPICB=X
      YPICB=Y
  215 CONTINUE
      IF(N.EQ.3) GO TO 140
      IF(N-7) 210,190,105
C
  217 XDEL=XDEL+XPICB-XPICA
      YDEL=YDEL+YPICB-YPICA
      GO TO 100
C
  220 WRITE (6,230) NREC
  230 FORMAT (1H1/' REGULAR EXIT PROGRAM CALCOM'/
     1' NUMBER OF CALLS =',I10/)
      STOP
      END
/*
//             EXEC FTL,PRM=PROGM,VLB=COPICA,ULB=DISK
//LKED.SYSLMOD DD DSN=SYS1.LIBPROGM,UNIT=DISK,VOL=SER=COPICA,
//             DISP=(OLD,KEEP)
//LKED.SYSLIN  DD DSN=&LOADSET,DISP=(OLD,DELETE)
//             DD *
  NAME CALCOM(R)
/*
//STEPA  EXEC AHC
//CMP.SYSIN    DD *
INDICA   CSECT
         USING *,11
         STM   14,12,12(13)
         LR    11,15
         LR    4,13
         LA    13,SAVET
         ST    4,4(13)
```

```
          ST    13,8(4)
          L     12,0(1)
          EXTRACT ATIOT,'S',FIELDS=(TIOT)
          L     2,ATIOT
          MVC   JBNM(L'JBNM),0(2)
          LM    15,1,PARMS
          SVC   255
          B     *+L'*(15)
          B     *+L'*+4
          B     HASP
          MVI   PRNM,C' '
          MVC   PRNM+1(L'PRNM-1),PRNM
HASP      L     15,AGIO
          CNOP  0,4
          BAL   1,CALLO
          DC    A(JHEURE)
          DC    A(JDATE)
CALLO     BALR  14,15
          MVC   0(3,12),PRNM+1
          MVI   3(12),X'40'
          MVC   4(46,12),PRNM+4
          L     13,4(13)
          LM    14,12,12(13)
          MVI   12(13),X'FF'
          BR    14
SAVET     DS    18F
PRNM      DS    CL20
          DC    C' '
JBNM      DS    CL8
          DC    C' '
JDATE     DS    0C
JAN       DS    CL2
          DC    C'/'
JMO       DS    CL2
          DC    C'/'
JJO       DS    CL2
          DC    C' '
JHEURE    DS    0C
JHE       DS    CL2
          DC    C':'
JMI       DS    CL2
          DC    C':'
JSE       DS    CL2
ATIOT     DC    A(0)
AGIO      DC    V(GIORNO)
PARMS     DC    F'8'
          DC    0F'0',C'AROS'
          DC    A(PRNM)
          END
/*
//STEPB  EXEC PGM=IEWL,PARM='LIST,MAP,NCAL'
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD UNIT=SYSSQ,SPACE=(CYL,(1,1))
//SYSLMOD  DD UNIT=DISK,VOL=SER=COPICA,DSN=SYSI.LIBPROGM(INDICA),
//            DISP=(OLD,KEEP)
//SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)
```

```
$     TIME 003
$     LINES 004
$     CLASS 2
//              EXEC LIBRAP,A='SYSU.LIBMF2',E='COPICB',K='DISK'
//INS.SYSIN     DD DATA
-OPT INDEX(DEWOLDE)
-ADD GRAPHIT,LIST
-DESC GRAPHIT VERSION JUNE 1978
-PGMR DEWOLDE
$     TIME 002
$     LINES 002
$     CLASS 2
//STEPS         EXEC FTGC
//CMP.SYSIN     DD *
      SUBROUTINE PLOT(X,Y,I)
C
C     MOVE PEN TO COORDINATES X,Y WITH RESPECT TO CURRENT AXIS SYSTEM.
C     I=2 PEN IS DOWN, I=3 PEN IS UP.
C     TRANSLATE X,Y INTO ABSOLUTE COORDINATES XX AND YY.
C     XCUR AND YCUR IS ZERO POINT OF CURRENT AXIS SYSTEM
C     IN RELATION TO ABSOLUTE AXIS SYSTEM.
C
      DATA XTOT,YTOT,XCUR,YCUR,FAC/0.0,0.0,0.0,0.0,1.0/
C
  100 XTOT=X
      YTOT=Y
      XX=X*FAC+XCUR
      YY=Y*FAC+YCUR
      N=1
      CALL BUFFER(N,XX,YY,I)
      RETURN
C
      ENTRY FACTOR(F)
      FAC=F
      RETURN
C
C     FINIM SETS NEW CURRENT AXIS SYSTEM
C     AND MOVES PEN TO NEW ZERO POINT
      ENTRY FINIM(X,Y)
      XX=X*FAC+XCUR
      YY=Y*FAC+YCUR
      N=1
      I=3
      CALL BUFFER(N,XX,YY,I)
      XCUR=XCUR+X*FAC
      YCUR=YCUR+Y*FAC
      XTOT=0.0
      YTOT=0.0
      RETURN
C
C     WHERE GIVES THE PEN POSITION IN RELATION TO THE CURRENT AXIS
      ENTRY WHERE(X,Y,F)
      X=XTOT
      Y=YTOT
      F=FAC
      RETURN
```

```
      END
/*
//STEP6         EXEC FTL,NC=NCAL
//LKED.SYSLMOD DD DSN=SYS1.LIBGRAPH,UNIT=DISK,VOL=SER=COPICA,
//              DISP=(OLD,KEEP)
//LKED.SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)
//              DD *
  ALIAS FACTOR
  ALIAS FINIM
  ALIAS WHERE
  NAME PLOT(R)
/*
//STEP9         EXEC FTGC
//CMP.SYSIN     DD *
      SUBROUTINE FINTRA
      N=3
      CALL BUFFER(N,XX,YY,I)
      END FILE 16
      RETURN
      END
/*
//STEP10        EXEC FTL,NC=NCAL
//LKED.SYSLMOD DD DSN=SYS1.LIBGRAPH,UNIT=DISK,VOL=SER=COPICA,
//              DISP=(OLD,KEEP)
//LKED.SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)
//              DD *
  NAME FINTRA(R)
/*
//STEP13        EXEC FTGC
//CMP.SYSIN     DD *
      SUBROUTINE NEWPEN(IA)
      N=6
      CALL BUFFER(N,XX,YY,IA)
      RETURN
      END
/*
//STEP14        EXEC FTL,NC=NCAL
//LKED.SYSLMOD DD DSN=SYS1.LIBGRAPH,UNIT=DISK,VOL=SER=COPICA,
//              DISP=(OLD,KEEP)
//LKED.SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)
//              DD *
  NAME NEWPEN(R)
/*
$     TIME 002
$     LINES 002
$     CLASS 2
//STEP15        EXEC FTGC
//CMP.SYSIN     DD *
      SUBROUTINE PICTNO(IA)
      N=7
      CALL BUFFER(N,XX,YY,IA)
      RETURN
      END
/*
//STEP16        EXEC FTL,NC=NCAL
//LKED.SYSLMOD DD DSN=SYS1.LIBGRAPH,UNIT=DISK,VOL=SER=COPICA,
```

```
//             DISP=(OLD,KEEP)
//LKED.SYSLIN  DD DSN=&LOADSET,DISP=(OLD,DELETE)
//             DD *
  NAME FICTNO(R)
/*
//STEP11       EXEC FTGC
//CMP.SYSIN    DD *
      SUBROUTINE SYMBL4(X,Y,SIZE,THETA,BCD,NA)
      DIMENSION BCD(1)
      LOGICAL*1 BCD
      BETA=0.0
      CALL SYMBOL(X,Y,SIZE,THETA,BCD,NA,BETA)
      RETURN
      END
/*
//STEP12       EXEC FTL,NC=NCAL
//LKED.SYSLMOD DD DSN=SYS1.LIBGRAPH,UNIT=DISK,VOL=SER=COPICA,
//             DISP=(OLD,KEEP)
//LKED.SYSLIN  DD DSN=&LOADSET,DISP=(OLD,DELETE)
//             DD *
  NAME SYMBL4(R)
/*
$     TIME 002
$     LINES 002
$     CLASS 2
//STEPC        EXEC FTGC
//CMP.SYSIN    DD *
//             EXEC FTGC
//CMP.SYSIN    DD *
      SUBROUTINE BUFFER(N,X,Y,K)
C
C     VERSION WITH COMPRESSED STORAGE - JAN.1970 BY HERMAN LOE VOLDE
C
C     N=1 MOVE PEN TO COORDINATES X,Y, BOTH IN CM
C                              K=2 PEN DOWN
C                              K=3 PEN UP
C     N=2 NOT USED, ERROR CONDITION
C     N=3 CLOSES FILE
C     N=4 CLOSES ONE BUFFER RECORD
C     N=5 NOT USED, ERROR CONDITION
C     N=6 NEW PEN
C     N=7 K IS PICTURE NUMBER
C     N.GT.7 NOT USED, ERROR CONDITION
C
C     ONE CALL (N,X,Y,K) IS COMPRESSED AS FOLLOWS
C     FIRST BYTE CONTAINS N
C     SECOND BYTE CONTAINS K, IF ANY
C     NEXT HALF BYTE CONTAINS NUMBER OF BYTES OF DELTA X, IF ANY
C     NEXT HALF BYTE CONTAINS NUMBER OF BYTES OF DELTA Y, IF ANY
C     NEXT BYTE(S) CONTAIN DELTA X, FIXED, IF ANY
C     NEXT BYTE(S) CONTAIN DELTA Y, FIXED, IF ANY
C     ALL VALUES POSITIVE
C
      DIMENSION NBUFF(200),LB(800),KB(4)
      LOGICAL*1 LB,KB
      EQUIVALENCE (NBUFF(1),LB(1)),(KB(1),KBFULL)
```

```
      DATA LBUFF,LLB,IBUF/200,792,1/
      DATA XOLD,YOLD/0.0,0.0/
      DATA ICOUNT,NCOUNT/10,10/
C
C     TO INCREASE BUFFER SIZE CHANGE DIMENSIONS OF BUFF AND LB
C     AND CONSTANT LBUFF AND LLB=4*LBUFF-8
C     SAME MODIFICATION IN SUBROUTINE 'READB'
C
      IF(N.GT.0) GO TO 120
  100 WRITE (6,110) N
  110 FORMAT (1H1/' ERROR AT LOADING BUFFER N= ',I10/)
      WRITE(6,500) (NBUFF(I),I=1,200)
  500 FORMAT(5X,10Z8/)
      STOP
C
C     LOAD N IN ONE BYTE
C
  120 KBFULL=N
      LB(IBUF)=KB(4)
      IBUF=IBUF+1
      IF(N.GT.1) GO TO 160
C
C-----N=1 MOVE PEN TO ABSOLUTE COORDINATES X,Y-------------------------
C     PRESENT PEN POSITION IS XOLD,YOLD
C     CALCULATE XDELTA AND YDELTA
C     PEN ADJUSTMENT IS K
C
      KBFULL=K
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LT.NCOUNT) GO TO 125
      ICOUNT=0
      KBFULL=KBFULL+128
  125 CONTINUE
      LB(IBUF)=KB(4)
      IBUF=IBUF+1
      IF(ICOUNT.GT.0) GO TO 126
      IX=IFIX(1000.0*(X+0.0005))
      IY=IFIX(1000.0*(Y+0.0005))
      GO TO 127
  126 CONTINUE
      IX=IFIX(1000.0*(X-XOLD+0.0005))
      IY=IFIX(1000.0*(Y-YOLD+0.0005))
  127 CONTINUE
      XOLD=X
      YOLD=Y
      IIX=IABS(IX)
      KX=1
      IF(IIX.LT.256) GO TO 130
      KX=2
      IF(IIX.LT.65536) GO TO 130
      KX=3
  130 IIY=IABS(IY)
      KY=1
      IF(IIY.LT.256) GO TO 135
      KY=2
      IF(IIY.LT.65536) GO TO 135
```

```
      KY=3
C
C-----COMPOSE THE LLCODE AND LOAD INTO BUFFER-------------------------
C
  135 LLCODE=KX
      IF(IX.LT.0) LLCODE=LLCODE+4
      LLCODE=LLCODE+KY*16
      IF(IY.LT.0) LLCODE=LLCODE+64
      KBFULL=LLCODE
      LB(IBUF)=KB(4)
      IBUF=IBUF+1
      KBFULL=IIX
      DO 140 I=1,KX
      IA=4+I-KX
      LB(IBUF)=KB(IA)
  140 IBUF=IBUF+1
      KBFULL=IIY
      DO 150 I=1,KY
      IA=4+I-KY
      LB(IBUF)=KB(IA)
  150 IBUF=IBUF+1
      GO TO 180
C
C-----N=3 END OF FILE-------------------------------------------------
C
  160 IF(N.EQ.2) GO TO 100
      IF(N.EQ.3) GO TO 190
      IF(N.EQ.4) GO TO 100
      IF(N.EQ.5) GO TO 100
C
C-----N=6 NEW PEN-----------------------------------------------------
C
      IF(N.GT.6) GO TO 170
      KBFULL=K
      LB(IBUF)=KB(4)
      IBUF=IBUF+1
      GO TO 180
C
C-----N=7 K IS PICTURE NUMBER-----------------------------------------
C
  170 IF(N.GT.7) GO TO 100
      KBFULL=K
      LB(IBUF)=KB(4)
      IBUF=IBUF+1
C
C-----TEST BUFFER-----------------------------------------------------
C
  180 CONTINUE
      IF(IBUF.LT.LLB) RETURN
  190 KBFULL=4
      LB(IBUF)=KB(4)
      WRITE (16) (NBUFF(I),I=1,200)
      IBUF=1
      RETURN
      END
/*
```

```
//STEPD         EXEC FTL,NC=NCAL
//LKED.SYSLMOD DD DSN=SYS1.LIBGRAPH,UNIT=DISK,VOL=SER=COPICA,
//              DISP=(OLD,KEEP)
//LKED.SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)
//              DD *
  NAME BUFFER(R)
/*
```

```
//STEP2        EXEC FTGC
//CMP.SYSIN    DD *
C-----PROGRAM GOULD BY HERMAN I. DE WOLDE-----JAN. 1978----------------
C                                                    SEP. 1978---REVISED
C
C     GOULD READS THE INTERMEDIATE GRAPHIC FILE
C     AND TRANSLATES THE ARGUMENTS INTO ACTUAL SUBROUTINE CALLS
C
C     THE PRESENT VERSION RECOGNIZES THE FOLLOWING SITUATIONS
C
C     N=1 CALL PLOT
C     N=2 ERROR CONDITION
C     N=3 END OF FILE  CALL PLOT (0.0,0.0,999)
C     N=4 CLOSES ONE BUFFER RECORD  RESOLVED BY SUBROUTINE READB
C     N=5 NOT USED  ERROR CONDITION
C     N=6 NEWPEN IS IGNORED BY GOULD
C     N=7 START OF A NEW PICTURE
C     N.GT.7 ERROR
C
      DIMENSION NPIC(100),INP(12)
      DIMENSION AREA(13)
C     XDEL AND YDEL ARE THE SUM OF ALL DELETED PICTURES
      DATA XDEL,YDEL/0.0,0.0/
      DATA SHIFT/5.0/
C     SHIFT IS THE DISPLACEMENT OF THE ORIGIN TO MAKE ROOM FOR
C        JOB IDENTIFICATION
      XDEL=XDEL-SHIFT
      XOFF=0.0
      YOFF=0.0
      XLAST=0.0
      FACTOR=0.32
C-----THE NEXT STATEMENT IS ONLY FOR METRIC INPUT COMBINED WITH
C     THE GOULD LIBRARY IN INCHES
      FACTOR=0.125984
      XMAX=-100.0
      YMAX=-80.0
      IFORM=0
      NCOPY=1
      NREC=0
      NAP=0
      READ (5,30,END=70) FACTOR,IFORM,NCOPY,LINE,MAX
   30 FORMAT (F6.2,4I6)
      IF(MAX.NE.0) XMAX=-FLOAT(MAX)
C-----THE NEXT STATEMENT IS ONLY FOR METRIC INPUT COMBINED WITH
C     THE GOULD LIBRARY IN INCHES
      XMAX=XMAX/2.54
C-----THE NEXT STATEMENT IS ONLY FOR METRIC INPUT COMBINED WITH
C     THE GOULD LIBRARY IN INCHES
      FACTOR=FACTOR*0.3937
      IF (NCOPY.EQ.0) NCOPY=1
C-----THE NEXT STATEMENT IS ONLY FOR METRIC INPUT COMBINED WITH
C     THE GOULD LIBRARY IN INCHES
      IF (FACTOR.EQ.0.0) FACTOR=0.125984
      IF (FACTOR.EQ.0.0) FACTOR=0.32
C
C-----READ THE PICTURE NUMBERS, IF ANY.-------------------------------
```

```
C
   40 READ (5,50,END=70) (INP(I),I=1,12)
   50 FORMAT (12I6)
      DO 65 I=1,12
      IF(INP(I).LE.0) GO TO 65
      IF(NAP+1).LE.100) GO TO 60
      WRITE (6,55)
   55 FORMAT (1H1/' REQUESTED NUMBER OF PICTURES.GT.100'/
     1' ONLY FIRST HUNDRED WILL BE DONE.'//)
      GO TO 70
   60 NAP=NAP+1
      NPIC(NAP)=INP(I)
   65 CONTINUE
      GO TO 40
   70 CONTINUE
      IF(NAP.LE.0) GO TO 90
      WRITE(6,80) (NPIC(I),I=1,NAP)
   80 FORMAT (1H1/' LIST OF REQUESTED PICTURES'/(20I6)/)
   90 CONTINUE
C
C-----PLOT INITIALIZATION AND JOB IDENTIFICATION----------------------
C
      CALL PLOTS(XMAX,YMAX,IFORM,NCOPY)
      CALL LINEWT(LINE)
      IF (IFORM.EQ.(-1)) CALL ERASE(1)
      CALL INDICA(AREA)
      CALL SYMBOL(0.0,10.0,0.2,AREA,270.0,50)
C
C
C-----READ BUFFER-----------------------------------------------------
C
  100 CALL READB(N,X,Y,K)
      IF(N.GT.0) GO TO 120
  105 WRITE (6,110) N,NREC
  110 FORMAT (1H1/' ERROR MAIN GOULD  N=',I10/' NUMBER OF CALLS NREC=',I
     110/)
      STOP
C
  120 IF(N.GT.1) GO TO 130
      XA=(X-XDEL)*FACTOR
      YA=(Y-YDEL)*FACTOR
      IF(XA.GT.XLAST) XLAST=XA
      XPIC=X
      YPIC=Y
      CALL PLOT(XA,YA,K)
      NREC=NREC+1
      GO TO 100
C
  130 IF(N.EQ.2) GO TO 105
  140 IF(N.GT.3) GO TO 150
      NREC=NREC+1
      CALL PLOT(XLAST+10.0,0.0,3)
      CALL PLOT(0.0,0.0,999)
      GO TO 220
C
  150 IF(N.EQ.4) GO TO 105
```

```
      IF(N.EQ.5) GO TO 105
C
  160 IF(N.GT.6) GO TO 170
      NREC=NREC+1
      GO TO 100
C
  170 IF(N.GT.7) GO TO 105
      XPICA=XPIC
      YPICA=YPIC
      XPICB=XPIC
      YPICB=YPIC
C
C-----K IS PICTURE NUMBER-----------------------------------------------
C
  180 IF (NAP.EQ.0) GO TO 100
  190 DO 200 I=1,NAP
      IF(K.EQ.NPIC(I)) GO TO 217
  200 CONTINUE
  210 CALL READB(N,X,Y,K)
      IF(N.NE.1) GO TO 215
      XPICB=X
      YPICB=Y
  215 CONTINUE
      IF(N.EQ.3) GO TO 140
      IF(N-7) 210,190,105
C
  217 XDEL=XDEL+XPICB-XPICA
      YDEL=YDEL+YPICB-YPICA
      GO TO 100
C
  220 WRITE (6,230) NREC
  230 FORMAT (1H1/' REGULAR EXIT PROGRAM GOULD'/
     1' NUMBER OF CALLS =',I10/)
C     ONLY FOR VERSION IN INCHES
C     ONLY FOR VERSION IN INCHES
      FACTOR=FACTOR*2.54
      WRITE (6,232) FACTOR
  232 FORMAT (' SCALING FACTOR IS ',F6.2/)
      STOP
      END
/*
//STEP2A        EXEC FTL,PRN=GOULD,VLB=COPICA,ULB=DISK
//LKED.SYSLMOD DD DSN=SYS1.LIBPROGM,UNIT=DISK,VOL=SER=COPICA,
//              DISP=(OLD,KEEP)
//LKED.SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)
//              DD *
  INCLUDE SYSLMOD(READB,INDICA)
  ENTRY MAIN
  NAME GOULD(R)
/*
//STEP2         EXEC FTGC
//CMP.SYSIN     DD *
C-----PROGRAM BENSON BY HERMAN I. DE WOLDE-----NOV. 1977---------------
C                                                    JUN. 1978---REVISED
C
C     BENSON READS THE INTERMEDIATE GRAPHIC FILE
```

```
C     AND TRANSLATES THE ARGUMENTS INTO ACTUAL SUBROUTINE CALLS
C
C     THE PRESENT VERSION RECOGNIZES THE FOLLOWING SITUATIONS
C
C     N=1 CALL TRAA(XA,YA,KA)
C     N=2 ERROR CONDITION
C     N=3 CALL PNUMA(0.0,0.0,NB,0.0,0.0)
C     N=4 CLOSES ONE BUFFER RECORD
C     N=5 ERROR
C     N=6 CALL PLUMA (K-1)
C     N=7 SKIP TO PICTURE K
C     N=GT.7 ERROR
C
      DIMENSION NPIC(100),INP(12)
      DIMENSION IBUF(275)
      DIMENSION AREA(13)
C     XDEL AND YDEL ARE THE SUM OF THE DELETED PICTURES
      DATA XDEL,YDEL/0.0,0.0/
      XMM=0.0
      LBUF=275
      ND=16
      NBLOC=1
      NREC=0
      NAP=0
      FACTOR=0.37
      CALL IBENA(IBUF,LBUF,ND)
C
C-----PLOT INITIALIZATION : 1) REACH THE ORIGIN; 2) IDENTIFY THE JOB.
C
      NB=1
      CALL PNUMA(5.0,-30.0,NB,0.0,0.0)
      CALL INDICA(AREA)
      XA=-5.0
      YA=18.
      CALL PCARA (XA,YA,0,AREA,50,0.3,0.5,0.0,-1.0)
C
      READ (5,30,END=70) FACTOR
   30 FORMAT (12F6.2)
      IF (FACTOR.EQ.0.0) FACTOR=0.37
C
C-----READ THE PICTURE NUMBERS. IF ANY.--------------------------------
C
   40 READ (5,50,END=70) (INP(I),I=1,12)
   50 FORMAT (12I6)
      DO 65 I=1,12
      IF(INP(I).LE.0) GO TO 65
      IF((NAP+1).LE.100) GO TO 60
      WRITE (6,55)
   55 FORMAT (1H1/' REQUESTED NUMBER OF PICTURES.GT.100'/
     1' ONLY FIRST HUNDRED WILL BE DONE.'//)
      GO TO 70
   60 NAP=NAP+1
      NPIC(NAP)=INP(I)
   65 CONTINUE
      GO TO 40
   70 CONTINUE
```

```
C
      IF (NAP.GT.0) GO TO 76
      WRITE(6,74)
   74 FORMAT(1H1/' ALL PICTURES WILL BE PROCESSED'/)
      GO TO 80
   76 WRITE(6,78) (NPIC(I),I=1,NAP)
   78 FORMAT(' LIST OF REQUESTED PICTURES'/(2016)/)
   80 CONTINUE
C
C-----READ BUFFER-----------------------------------------------------------
C
  100 CALL READB(N,X,Y,K)
      IF(N.GT.0) GO TO 120
  105 WRITE (6,110) N,NREC
  110 FORMAT (1H1/' ERROR MAIN BENSON N=',I10/' NUMBER OF CALLS NREC=',I
     110/)
      STOP
C
  120 IF(N.GT.1) GO TO 130
      XA=(X-XDEL)*FACTOR
      YA=(Y-YDEL)*FACTOR
      XPIC=X
      YPIC=Y
      KA=0
      IF(K.LT.3) KA=1
      IF(XA.GT.XMM) XMM=XA
      CALL TRAA(XA,YA,KA)
      NREC=NREC+1
      GO TO 100
C
  130 IF(N.EQ.2) GO TO 105
  140 IF(N.GT.3) GO TO 150
      NREC=NREC+1
      NB=999
      CALL PNUMA((XMM+5.0),0.0,NB,0.0,0.0)
      GO TO 220
C
  150 IF(N.EQ.4) GO TO 105
      IF(N.EQ.5) GO TO 105
C
  160 IF(N.GT.6) GO TO 170
      NREC=NREC+1
      KA=K-1
      CALL PLUMA(KA)
      GO TO 100
C
  170 IF(N.GT.7) GO TO 105
      XPICA=XPIC
      YPICA=YPIC
      XPICB=XPIC
      YPICB=YPIC
C
C-----K IS PICTURE NUMBER---------------------------------------------------
C
  180 IF (NAP.EQ.0) GO TO 100
  190 DO 200 I=1,NAP
```

```
      IF(K.EQ.NPIC(I)) GO TO 217
  200 CONTINUE
  210 CALL READB(N,X,Y,K)
C
      IF(N.NE.1) GO TO 215
      XPICB=X
      YPICB=Y
  215 CONTINUE
      IF(N.EQ.3) GO TO 140
      IF(N-7) 210,190,105
C
  217 XDEL=XDEL+XPICB-XPICA
      YDEL=YDEL+YPICB-YPICA
      GO TO 100
C
  220 WRITE (6,230) NREC
  230 FORMAT (1H1/' REGULAR EXIT PROGRAM BENSON'/
     1' NUMBER OF CALLS =',I10/)
      WRITE (6,232) FACTOR
  232 FORMAT (' SCALING FACTOR IS ',F6.2/)
      STOP
      END
/*
//STEP2A        EXEC FTL,PRN=BENS,VLB=COPICB,ULB=DISK
//LKED.SYSLMOD DD DSN=SYS1.LIBPROGM,UNIT=DISK,VOL=SER=COPICA,
//              DISP=(OLD,KEEP)
//LKED.SYSLIN  DD DSN=&LOADSET,DISP=(OLD,DELETE)
//              DD *
  INCLUDE SYSLMOD(READB,INDICA)
  ENTRY MAIN
  NAME BENSON(R)
/*
//              EXEC COMPRESS,DSN='SYS1.LIBPROGM',VOL=COPICA,UNI=DISK
//          EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1     DD UNIT=DISK,VOL=SER=COPICA,DISP=OLD
//SYSIN     DD *
   LISTPDS DSNAME=SYS1.LIBPROGM,VOL=3330-1=COPICA
/*
```