TM-70-2011-2

TECHNICAL MEMORANDUM

AUPLOT II - A SYSTEM OF DATA HANDLING AND PLOT INSTRUCTION SUBROUTINES FOR COMPUTER GRAPHICS

Bellcomm

BELLCOMM, INC.

955 L'ENFANT PLAZA NORTH, S.W., WASHINGTON, D.C. 20024

COVER SHEET FOR TECHNICAL MEMORANDUM

TITLE- AUPLOT II - A System of Data Handling and Plot Instruction Subroutines for Computer Graphics

TM-70-2011-2

DATE- November 20, 1970

FILING CASE NO(S)- 103-9

AUTHOR(S)- R. F. Jessup

FILING SUBJECT(S) (ASSIGNED BY AUTHOR(S))- Computer Graphics

ABSTRACT

AUPLOT II is a set of Fortran callable subroutines designed to facilitate computer plotting in scientific and simulation applications. It is an implementation of logical and step-by-step data collection and plotting procedures which are largely independent of specific computer and plotting devices. AUPLOT II may be used in either interactive or batch environments. It features deferred plotting from a data and instruction queue which is temporarily stored on a secondary file. Use of the secondary file minimizes space requirements for computer graphics in the AUPLOT user's program.

BELLCOMM. INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

SUBJECT: AUPLOT II, A System of Data Handling and Plot Instruction Subroutines for Computer Graphics
Case 103-9

DATE: November 20, 1970

FROM: R. F. Jessup

TM-70-2011-2

TECHNICAL MEMORANDUM

INTRODUCTION

The generation of a graph is a two phase process. One first obtains data, then maps it into an image space. AUPLOT, a set of Fortran callable subroutines for computer preparation of graphs, is designed in accordance with these two phases. It is intended to facilitate the use of computer driven plotters in scientific and simulation programming.

Phase 1 of AUPLOT consists of subroutine calls to be compiled into a user's program. These calls are summarized in Table 1. They provide for systematic collection of data, as it is generated, from up to 256 program variables, and they also provide a logical and step-by-step plotting instruction set. The instruction set provides independent subroutine calls for the separate aspects of a plot, that is, the data to be plotted and such items as title, scales, background grid, and legend. If desired, only the data need be specified for a plot; the program will make appropriate selections, through default options, for any items omitted from the calling program.

Phase 1 normally transmits the collected data to a secondary storage file. The plotting instruction calls are keyed and also placed on the file so that actual plotting is deferred. This organizational feature of AUPLOT relieves the system user of the need for estimating and providing space required for his graphical data; it also relieves him of the task of allowing for the space required by plotting subroutines.

Phase 2 of AUPLOT arranges the collected data into appropriate plotting order and, through a number of built-in subroutines keyed to the Phase 1 instructions, maps the data into 3-D or conventional X-Y graphs via the on-line printer or teleprinter (80-column printer or teletypewriter) or an off-line plotting unit. For off-line plotting, Phase 2 in the current version of AUPLOT drives the SC4020 microfilm recorder through its extensive software library.

Figure 1 pictorially represents the overall relationship between the Phase 1 and Phase 2 routines of AUPLOT, the secondary storage file, and the hardware devices. In normal operation the secondary storage file IOPLT contains a sequence of data tables, plot requests, and legend information recorded in accordance with Phase 1 subroutine calls within a user's program. Upon user's program exit from the computer, a separate execution of AUPLOT Phase 2 processes the Fortran file IOPLT into on-line printer graphs and the file PLOT containing instruction codes to drive the SC4020.

AUPLOT is available in a time sharing mode. This mode features bypassing the secondary file by linking of Phase 1 subroutine calls directly to their Phase 2 counterparts. The linking is achieved automatically by selection of a special AUPLOT subroutine at user program load time. In this mode AUPLOT permits interactive user program operation.

AUPLOT contains built-in provisions for interfacing additional plot routines, and can therefore be used with other plotting devices. It consists of 25 Fortran and two machine language subroutines and currently runs on the UNIVAC 1108 time-sharing computer (References 2 and 3). AUPLOT as presently implemented is an extension of an earlier version described in References 4 and 5.

AUPLOT DATA COLLECTION AND STORAGE

Five subroutines are provided in AUPLOT Phase 1 for the collection and storage of up to 256 user program variables (or more upon system recompilation). For identification purposes, both within the AUPLOT system and in the preparation of plotting instructions, each variable is assigned an identifier. Each identifier (i in the following CALL statements) takes the form of a non-zero integer or character. When used in character form, the identifier is placed along appropriate data axes of the plotted results.

For data collection there are five subroutines:

```
CALL COLECT (i, a)

CALL COLIST (i, a, n)

CALL COLIMP (i, a)

CALL PLTOUT

CALL CULOUT (i)
```

One data word per subroutine call may be set aside by use of the subroutine COLECT. Entire arrays $\{a_j, j = 1, 2, \ldots, n\}$ may be set aside by use of COLIST. Both subroutines build an internal table of data identifiers and indices, and they initiate data movement from the user's program area to AUPLOT storage. Table building and data movement may be performed separately under control of the subroutines COLIMP and PLTOUT, respectively. Choice of collection technique is open to the AUPLOT user. The internal data arrangement for plotting is identical among all three techniques.

In each collection subroutine the data represented by the argument a takes the form of a real variable or a constant. In a COLECT statement a may be a constant, a variable, a subscripted variable, or any other one word Fortran expression. In a COLIST statement the argument a should be a one dimensional array or equivalent. In a COLIMP statement a should be a variable or a subscripted variable with a constant subscript.

The subroutine CULOUT performs the inverse of data collection. CULOUT is used to release data after plotting has been accomplished. Data identifiers associated with such released data are then available for reuse in the program. If i is not zero in the subroutine CALL, CULOUT will release the identifier i; if i is zero, all identifiers are released.

There are eight subroutines for the assignment of additional data attributes. These attributes, which include lower and upper plotting limits, background density and logarithmic scale, are generally substituted for default attributes whenever the identified data is processed by any of the plotting routines built into AUPLOT. The subroutines include:

```
CALL PLTICS (i, Tl, Tu)

CALL PLTICH (i)

CALL PLTICA (i, j)

CALL PLWIND (i, w)

CALL PLWINH (i)

CALL PLWINA (i, j)

CALL PLTDEN (i, d)

CALL PLTLOG (i, n)
```

Lower and upper plotting bounds are normally taken as the data minimum and maximum values, respectively. The limits may be assigned explicitly by the PLTIC subroutines. PLTICS specifies the usc of Tl and Tu, PLTICH specifies that the current limits be held from plot to plot, even if additional collecting takes place, and PLTICA specifies that the limits associated with j be assigned to i. The limits may be assigned implicitly as a window w or a variation about the data midpoint by the PLWIN subroutines. The window w is given directly, held, or assigned from j by PLWIND, PLTICH, or PLWINA, respectively. The special limits attributes may be revoked by a call to PLTICS with Tl = Tu = 0.

The assignment of the grid density is made through the subroutine PLTDEN. Normally the grid lines are drawn at about 10 lines per inch for a standard value of d = 10; thus, a call to PLTDEN with d = 50 to 100 gives a corresponding spacing of approximately 1/2 to 1 inch. A call with d = 0 causes the grid to be omitted. A logarithmic scale is indicated by the PLTLOG subroutine with n = 1. Data to be logarithmically scaled should be positive with positive exponent range up to 10^{10}. A call to PLTLOG with n = 0 resets the attribute to linear scaling.

PHASE 1 PLOTTING INSTRUCTION SET

The AUPLOT system contains four built-in plotting subroutines for the on-line printer, four for the time-sharing teleprinter, and four for the off-line SC4020 microfilm recorder. For the on-line printer and the time-sharing teleprinter, one to four dependent variables can be plotted in rectangular form against one independent variable. The subroutines are designed for a 51 row dependent variable range and a 121 or 61 column independent variable range for the printer or teleprinter, respectively. Calling sequences follow for the printer, with each argument a data identifier from a Phase 1 data collection subroutine. For the teleprinter read TTX, TT2X, etc.

CALL PTX (i_t, i_a)

CALL PT2X (i_t, i_a, i_b)

CALL PT3X (i_t, i_a, i_b, i_c)

CALL PT4X (i_t, i_a, i_b, i_c, i_d)

The above subroutines plot the dependent variables i_a, i_b, and i_c, etc., as functions of the independent variable i_t. An appropriate numerical scale will be placed in the lower margin for i_t and in the left margin for i_a. No scale will appear for i_b, i_c, and i_d. They may be allowed to range free over the 51 rows, or they may be scaled to the plotting limits of i_a via calls to PLTICA.

For the SC4020, three rectangular plot routines are provided, along with one 3-dimensional perspective routine:

```
CALL QTX     (i_t, i_a)
CALL QTSXY   (i_t, i_a, i_b)
CALL QTSXYZ  (i_t, i_a, i_b, i_d)
CALL Q3DPER  (i_t, i_a, i_b)
```

QTX refers to a rectangular graph on microfilm with horizontal axis i_t and vertical axis i_a. The axes will be marked with the characters represented by i_t and i_a, respectively, and an appropriate grid pattern and scales will be automatically selected. A QTSXY graph consists of a rectangular graph of i_t versus i_a on the lower half of the output page and of i_t versus i_b on the upper half. It may be thought of as two QTX graphs, each reduced in height and fitted onto a single page. A QTSXYZ graph consists of three QTX type graphs on the same page. The plot i_t versus i_a appears on the lower third, i_t versus i_b on the middle third, and i_t versus i_c on the upper third.

A perspective projection of 3-dimensional data into a 2-dimensional plot is obtained by the Q3DPER subroutine. The data to be plotted may represent a curve or a surface in 3-D space. Q3DPER will draw a curve if i_a, i_b, and i_c identify equal amounts of data. On the other hand, if the number of i_c entries is a multiple of the number of i_a and i_b entries, then a surface will be drawn. Data collection for a surface is illustrated by the following Fortran statements.

```
                  CALL COLIST ('A', A, M)
                  CALL COLIST ('B', B, N)
                  DO 10 I = 1, M
                  DO 10 J = 1, N
             10        CALL COLECT ('C', F(A(I), B(J)))
```

Figures 3 through 10 give representative facsimiles of the plots produced by the built-in subroutines.

The AUPLOT system contains several additional plotting instruction subroutines. These subroutines assist the user by providing for control of the plot legend and built-in plotting options. A title appears across the top of each graph as a result of the statement:

CALL PLTITL (t)

where t is a string of up to 48 characters of the form nH... or equivalent. A title so requested continues in use on subsequent graphs until another call to PLTITL is made.

Normally the printer and teleprinter graphs give a sequence of points marked with an asterisk, while the microfilm graphs show curves formed by connecting unmarked points with short line segments. Both can be switched to marking points with an arbitrary character i by use of the statement:

CALL PLCHAR (i)

where i is of the form 1Hi or equivalent. The effect of PLCHAR can be revoked by a call with i = 0.

SC4020 plots are normally composed as frames, one frame of photosensitive paper or film per plot instruction. In the AUPLOT built-in routines the frame is advanced before the plot is drawn. To inhibit the frame advance one may use the statement

CALL PLTSIM (i)

with i = 2. To advance the frame for the next plot instruction and inhibit thereafter, a call to PLTSIM with i = 1 is required. To restore to normal advancing for each plot a call should be made with i = 0. The subroutine PLTSIM provides for the construction of multiple curve plots on the SC4020. Its use is applicable to each of the built-in microfilm plot routines.

Multiple Curves may be identified or tagged as they are drawn by use of the

CALL PLTAGS (i)

statement where i represents a string of six characters. The string i will be placed near the first and last points of each rectangular plot to follow. Its effect may be revoked by a call with i = 0. PLTAGS is not applicable to 3-D plots.

The perspective projection technique employed by Q3DPER is illustrated in Figure 2.* No background grid is provided. Instead, the three reference axes are drawn with data identifiers attached. The axes origin is normally placed at the data minimum value, i.e., min(X, Y, Z). Any other origin may be specified by use of the

CALL Q3DORG (o, i)

statement where o is a point in (X, Y, Z) space; i = 1 indicates that reference axes are to be drawn through the origin, and i = -1 indicates that the axes are to be omitted. The effect of Q3DORG may be revoked by a call with i set to 0.

The observer of an Q3DPER graph is normally positioned at 100 times the data maximum value. Repositioning may be specified by the

CALL Q3DEYE (e, i)

statement where e is a point in (X, Y, Z) space; i = 1 indicates that the Q3DPER graph is to be drawn as seen from the observer's position e, while i = 0 indicates a reversion to normal e selection.

Sometimes the 3-D effect of a curve on surface can be enhanced by shading, i.e., dropping perpendicular lines from the curve to a reference plane perpendicular to one of the data axes. To achieve this effect one may use the

Q3DSHA (q, i)

*This technique is based on a vector algebraic projection calculation from Reference 6. Additional aids to understanding and perhaps extending the 3-D feature may be found in References 7 through 11.

statement where q positions the plane along the ith axis, i = 1, 2, 3. Figure 10 illustrates an Q3DPER graph with shading by Q3DSHA. A call with i set to 0 revokes this effect.

Once all plot requests are inserted, this fact is indicated through a call to the subroutine PLTEND. No arguments are required.

AUPLOT debugging aids include a number of diagnostic printouts, which are explained in Table 5. Additional printout is available via the subroutine PLTDMP. PLTDMP, which is called without arguments, tabulates both Phase 1 and Phase 2 storage areas as they appear at the point of call. Table 4 is intended as an aid to interpreting PLTDMP printouts. This table defines each AUPLOT symbol and shows those subroutines in which the symbol is used or modified.

ADDITIONAL PHASE 1 SUBROUTINES

Some additional subroutines have been provided in Phase 1 for linking the AUPLOT user to the SC4020 frame advance and printing subroutines. The first provides for a frame advance without plotting and is indicated by the statement

```
CALL AUFRAM (3)
```

Printed information may be placed at selected points on the plotting surface through a call to either AUPRIN or AURITE. AUPRIN is an interface with the SC4020 library routine APRNTV which uses the charatron tube to print "extruded" characters of fixed size. The margin information in Q3DPER graphs is printed by APRNTV. The calling sequence for AUPRIN is given by

```
CALL AUPRIN (kx, ky, mx, my, m, c)
```

where (kx, ky) is the starting point, center of first character, in the printed message. The pair (kx, ky) may be given as integers, $0 < kx, ky < 1024$, in reference to SC4020 raster units or as floating point in reference to units of the data being plotted. Use of AUPRIN with kx and ky in data units should follow calls for QTX or similar graphs.

The pair (mx, my) represents X and Y increments between successive characters. Raster units are required. A message to be printed in the horizontal positive X direction may have (mx, my) = (8, 0). For the vertical direction one may use (0, 12).

The integer m gives the number of characters to be printed, while c gives the Hollerith array containing the characters. AURITE is an interface with the SC4020 library routine RITE2V, which uses the vector feature of the SC4020 to draw characters of various sizes. It is called by the statement

CALL AURITE (kx, ky, r, i, m, c)

where r is a raster position at which a carriage return is to be simulated, $0 < r < 1024$. On such a return, printing resumes on a new line under the first character. The argument i indicates the direction of printing successive characters; 0 indicates top to bottom, 90 left to right, 180 bottom to top, and 270 right to left (upside down).

The other arguments are as given in AUPRIN. The SC4020 library provides for changing the size of RITE2V characters through the library routine CHSIZV. A link to CHSIZV is provided by the

CALL AURSIZ (C_x, C_y, f_x, f_y, k_t)

statement where C_x and C_y are width and height keys, respectively. Normally each is assigned a value of 3 for characters of the size given by the PLTITL statement. Other width and height keys may be selected from the ranges 1 through 15 and 1 through 9, respectively. AURSIZ also links the AUPLOT user to the SC4020 library routine RITSTV. This routine selects an alphabet or table of SC4020 characters and defines the field within which the characters are drawn. Field dimensions (C_x and C_y for width and height, respectively) are given in SC4020 raster units. The integer k_t, $1 \leq k_t \leq 3$, is used as an index for selecting one of the SC4020 tables TABL1V, TABL2V, and TABL3V.

AUPLOT INTERNAL DATA COLLECTION AND STORAGE PROCEDURES

The AUPLOT data collection process consists of several steps taken to move data from the user's program area to a set of storage blocks for efficient linking to the built-in plotting subroutines. Each step centers about one or more of the following tables: IPLMAP, PLTDAT, ITAPE, IDLINK, and IDATA.

The purpose of the table IPLMAP, or plot map, is to provide overall control of data movement within the AUPLOT system. It has the following format for each of k = 1, 2,..., up to 256 four word entries:

$IPLMAP_{1,k}$ Array size from COLIST. 0 if the data identifier was taken from a COLECT or COLIMP call.

$IPLMAP_{2,k}$ Data identifier.

$IPLMAP_{3,k}$ Internal index for data from COLIST or COLIMP. 0 if data is from COLECT.

$IPLMAP_{4,k}$ Collection code identifying data source: COLECT, COLIST, or COLIMP.

Construction of the plot map takes place under control of the subroutines COLECT, COLIST, COLIMP, and CULOUT. CULOUT sets one or all entries to 0, while the other subroutines supply the information indicated.

The purpose of the plot map internal index is to permit movement of data for non-zero plot map entries from scattered program variables into a subscripted buffer, or table ITAPE, for efficient forwarding to Phase 2. Normally ITAPE is written as a data record on the secondary file, and has the following format.

$ITAPE_1$ Size of current plot map plus 4

$ITAPE_2$ 2

$ITAPE_{3,4}$ 0

$ITAPE_{4+k}$ Current data value corresponding to $k^{\underline{th}}$ plot map entry. Whenever the subroutine COLIST is used, it may be necessary to indicate that a current data value has already been supplied for this entry. A special flag of 1 is used to indicate this.

ITAPE is forwarded to Phase 2 by the subroutine PLTOUT. Movement of data into ITAPE is in accordance with collection subroutine usage. Whenever PLTOUT is called, either directly by the AUPLOT user, as in conjunction with COLIMP, or from within various Phase 1 subroutines, the current value of each program variable identified by COLIMP is moved into ITAPE. PLTOUT also moves successive values of data identified by COLIST into ITAPE and thence to Phase 2, until the specified number of elements have been collected. The subroutine COLECT, on the other hand, moves the data of its calling sequence into an intermediate buffer PLTDAT. The subroutine PLTOUT then loads COLECT data into ITAPE from this buffer. PLTDAT has the following format, where k = 1, 2, ..., up to 256 as in the plot map:

$PLTDAT_k$ Current data value corresponding to each k^th^ plot map entry established by COLECT. A flag 1 appears if the current value has already been forwarded to Phase 2.

Both of the tables IPLMAP and ITAPE are normally placed on the intermediate file IOPLT. They are keyed in the same manner as Phase 1 instruction records for appropriate processing by Phase 2.

In order to achieve fast plotting speeds, it is necessary that the data from the user's program be arranged into separate blocks of core storage. Furthermore, in order that the user have flexibility in requesting plots, it is necessary that the plotting variables be co-resident in storage. To achieve these objectives, Phase 2 reads the data associated with each plot map identifier into sub-blocks of the IDATA table.

Initial layout of the IDATA table is in accordance with the current number of variables as given by the plot map. Two variables are each assigned one half of the space in IDATA. Three variables are each assigned one third, and so forth. Blocks released as a result of user program CULOUT requests are made available for reassignment, either to new variables or possibly to existing block extensions.

Assignment of blocks in the IDATA table is under control of the Phase 2 subroutine APPLOT and is reflected in the table IDLINK. IDLINK contains one 5 word entry for each non-zero data identifier in the plot map. It has the following format.

$IDLINK_{1,k}$ Starting subscript of $k^{\underline{th}}$ block assignment in IDATA.

$IDLINK_{2,k}$ Input data count, including attributes.

$IDLINK_{3,k}$ Block size.

$IDLINK_{4,k}$ Block overflow counter.

$IDLINK_{5,k}$ Input data count at next block overflow.

Block overflow as indicated by $IDLINK_{4,k}$ is a result of high data collections. A diagnostic printout is issued to alert the system user. Space is then made available for continuing storage by truncating the overflowing block. Every other value of the stored data is released. Thereafter, for n = 1, 2, 3, ..., overflows, respectively, each 2^n value of the overflowing variable is accepted from the secondary file or directly from the user's program. As a result of this scheme the AUPLOT system user is relieved of data counting chores, and is assured of getting at least good estimates of all requested plots.

OPERATING SYSTEM INSTRUCTIONS

There are three modes of use of the AUPLOT system: time-sharing, deferred plotting via subroutine call, and deferred plotting via separate Phase 2 execute. Mode selection is made by control card instructions to the computer executive system. The following cards give typical instructions for the UNIVAC 1108 EXEC VIII.

(a) Time-sharing with AUPLOT file J8 for up to 8 plotting variables.

```
@     RUN

@     ASG,TM     PLOTFILE,T,PLOT      .OFF-LINE PLOTTER FILE

@     FOR,IS     A,A

... user's program ...

@     MAP,IS     B,B

      IN         A

      IN         PLTNOW/AUPLOT

      LIB        AUPLOT*J8

@     XQT        B

... user's input ...

@     FIN
```

(b) Deferred plotting via user's program call to the Phase 2 control subroutine APPLOT. This mode may be used to "stack" a sequence of plot runs. It may require additional MAP instructions to establish program overlays.

```
@     RUN

@     ASG,TM     PLOT FILE,T,PLOT

@     ASG,T      31,F4                         SECONDARY FILE IOPLT

@     FOR,SI     A,A

... user's program ...

      CALL APPLOT

      . . .

@     MAP,SI     B,B

      IN         A

      IN         PLATER/AUPLOT

      LIB        AUPLOT*J64

@     XQT        B

... user's input ...

@     FIN
```

(c) Deferred plotting via separate execution of the Phase 2 absolute PHASE2. This is the normal mode of AUPLOT operation and is illustrated with the large capacity file AUPLOT*J256 for up to 256 plotting variables.

```
@     RUN

@     ASG,T     31,F4

@     FOR,SI    A,A

... user's program ...

@     MAP,SI    B,B

      IN        A
```

```
        IN       PLATER/AUPLOT
        LIB      AUPLOT*J256
    @   XQT      B
    ... user's input ...
    @   ASG,TM   PLOTFILE,T,PLOT
    @   XQT      AUPLOT*J256.PHASE2
    @   FIN
```

SUMMARY

AUPLOT is a set of computer programs designed to facilitate the use of plotting devices for displaying data from scientific applications and simulation programs. In its plot later mode AUPLOT overcomes core storage constraints and minimizes input and output swapping of large program overlays. AUPLOT achieves its objectives through a data handling feature and system of plotting instructions designed to interface with various plotting device library routines. Scales, titles, background grid, and legend information are specified through simple subroutine calls or as determined by built-in default options. Types of graphs provided by built-in routines include up to 4 rectangular x-y curves on the on-line printer or time-shared teleprinter, and rectangular x-y, split screen x-y, and 3-D perspective curves, all with or without overlays, on the SC4020 microfilm recorder (Reference 1).

AUPLOT was initially implemented as an output feature of a trajectory simulation program developed in collaboration with D. H. Novak (Reference 13). Suggestions of numerous users at Bellcomm have helped guide its further development for general applications. All these suggestions are kindly appreciated.

R. F. Jessup

2011-RFJ-vh R. F. Jessup

Attachments

REFERENCES

[1] Programmers Reference Manual, SC4020 Microfilm Recorder, Stromberg-Carlson Document 9500056, October, 1964.

[2] Programmers Reference Manual, Univac 1108 Fortran V, Sperry-Rand Document UP-4060, 1966.

[3] Programmers Reference Manual, Univac 1108 EXEC VIII, Sperry-Rand Document UP-4144, 1968.

[4] Jessup, R. F., "AUPLOT Graphical Data Processing," Bellcomm, Inc., Memorandum for File B69-10117, October 27, 1968.

[5] Jessup, R. F., "AUPLOT System Description," Bellcomm, Inc., Memorandum for File B69-10116, October 21, 1968.

[6] Caldwell, S. F., "Automated Three Dimensional Perspective Plotting," Bellcomm, Inc., Memorandum for File B69-09044, September 18, 1969.

[7] Lathrop, K. D., PLOT3D, UAIDE Computer Program Abstract, October 1, 1963.

[8] Baldwin, H. L., "MPLOT, Orbit Displays Using the SC-4020," General Dynamics Report, Number GD/A 63-0459, July, 1963.

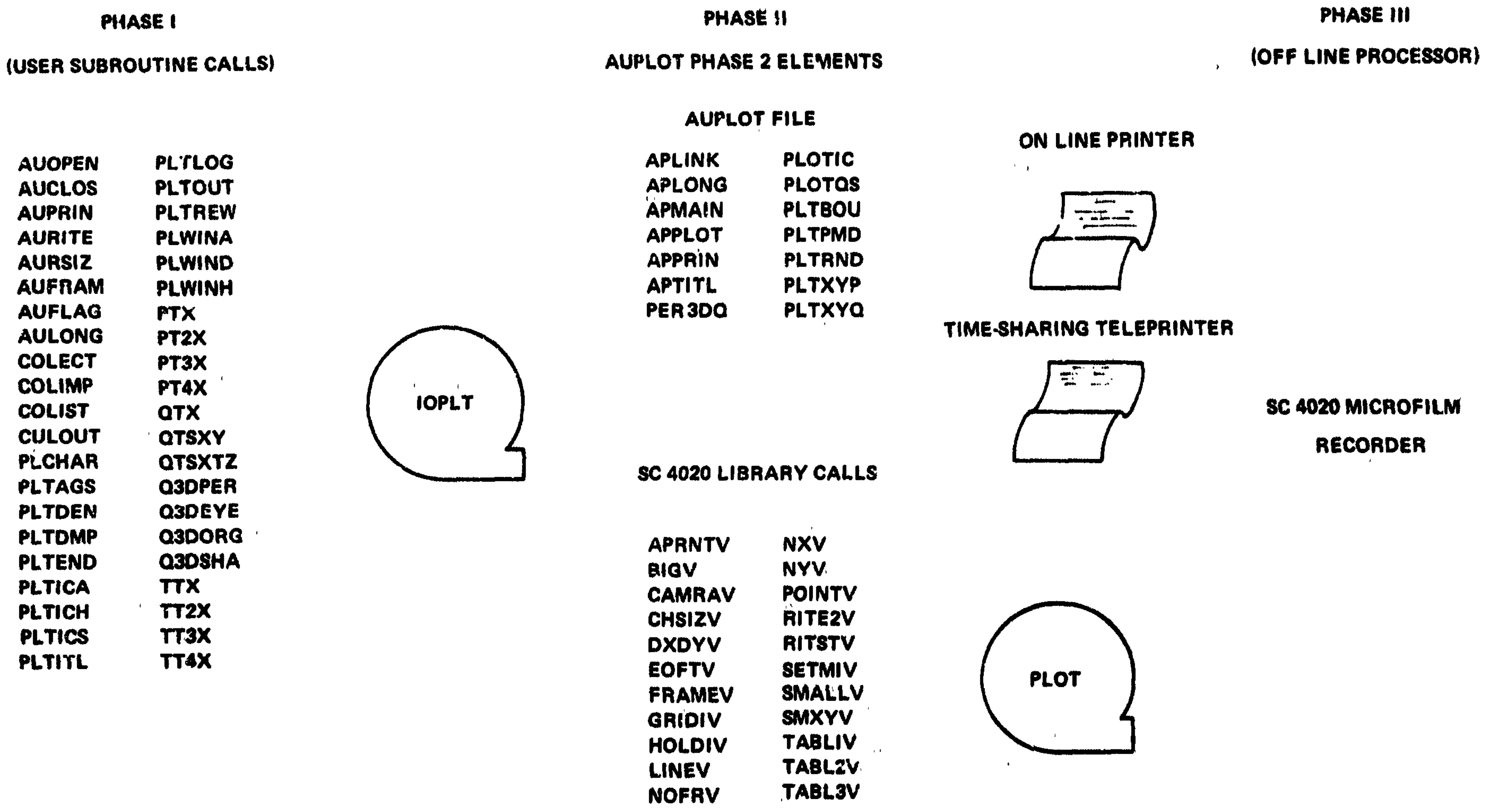
[9] Kubert, B. et al, "The Perspective Representation of Functions of Two Variables," JACM, Vol. 15, No. 2, April, 1968.

[10] McCormick, C. W., editor, "The NASTRAN Users Manual, NASA SP-222, October, 1969.

[11] Lerman, H. N., "A Planar Solution to 3-Dimensional Plotting," Software Age, July, 1970.

[12] Jessup, R. F., "A Note on Rounding the Endpoints of an Interval," Bell Telephone Laboratories, Memorandum for File, July 3, 1965.

[13] Novak, D. H., "Virtual Mass Technique for Computing N-Body Solutions," Bellcomm, Inc., Technical Memorandum in preparation.



PHASE I

(USER SUBROUTINE CALLS)

AUOPEN
AUCLOS
AUPRIN
AURITE
AURSIZ
AUFRAM
AUFLAG
AULONG
COLECT
COLIMP
COLIST
CULOUT
PLCHAR
PLTAGS
PLTDEN
PLTDMP
PLTEND
PLTICA
PLTICH
PLTICS
PLTITL

PLTLOG
PLTOUT
PLTREW
PLWINA
PLWIND
PLWINH
PTX
PT2X
PT3X
PT4X
QTX
QTSXY
QTSXTZ
Q3DPER
Q3DEYE
Q3DORG
Q3DSHA
TTX
TT2X
TT3X
TT4X

IOPLT

PHASE II

AUPLOT PHASE 2 ELEMENTS

AUPLOT FILE

APLINK
APLONG
APMAIN
APPLOT
APPRIN
APTITL
PER3DQ

PLOTIC
PLOTQS
PLTBOU
PLTPMD
PLTRND
PLTXYP
PLTXYQ

SC 4020 LIBRARY CALLS

APRNTV
BIGV
CAMRAV
CHSIZV
DXDYV
EOFTV
FRAMEV
GRIDIV
HOLDIV
LINEV
NOFRV

NXV
NYV
POINTV
RITE2V
RITSTV
SETMIV
SMALLV
SMXYV
TABL1V
TABL2V
TABL3V

ON LINE PRINTER

TIME-SHARING TELEPRINTER

PLOT

PHASE III

(OFF LINE PROCESSOR)

SC 4020 MICROFILM RECORDER

FIGURE 1 - GRAPH PREPARATION VIA AUPLOT

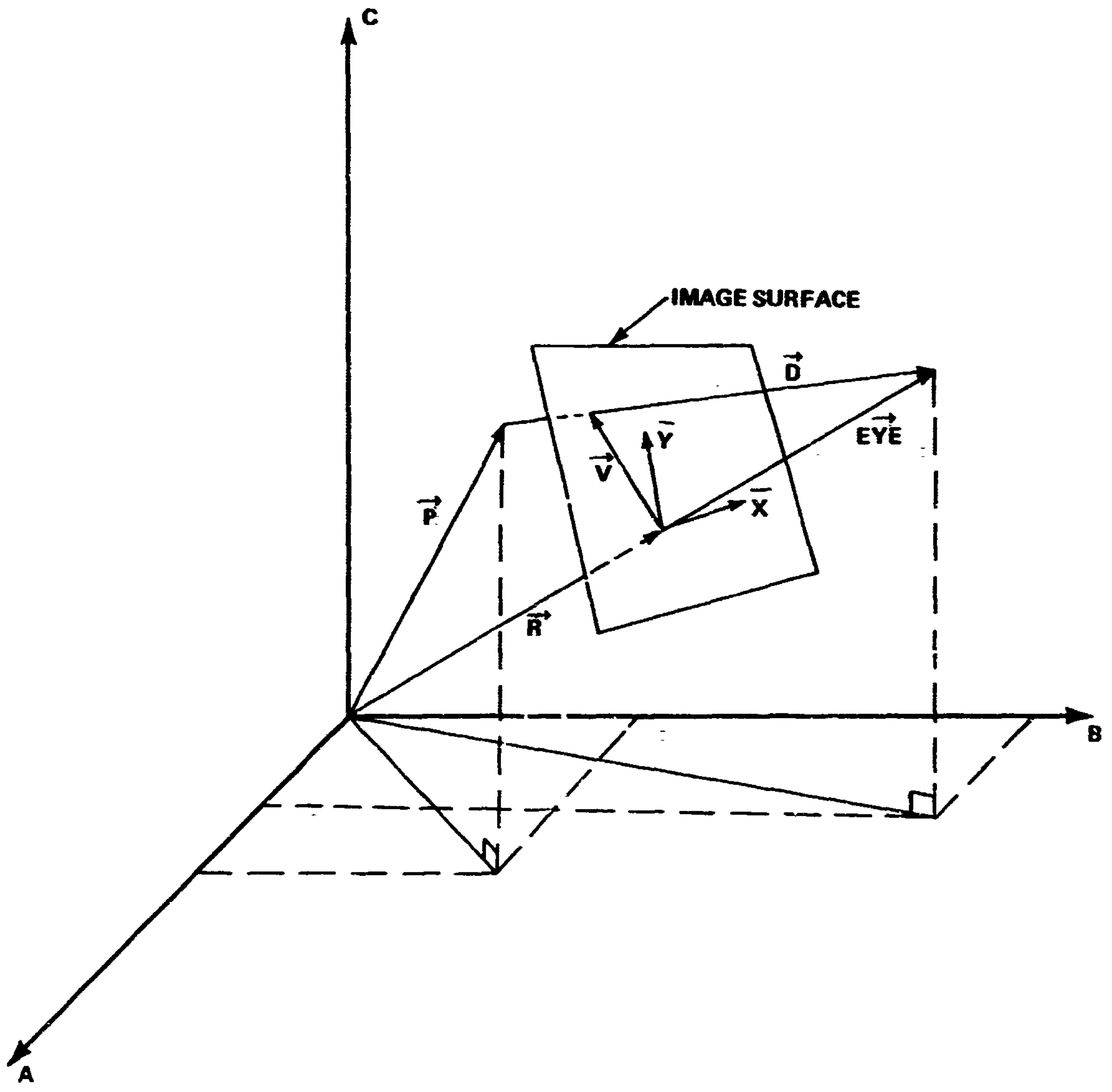


FIGURE 2 - PROJECTION GEOMETRY EMPLOYED IN THE Q3DPER PLOTTING ROUTINE

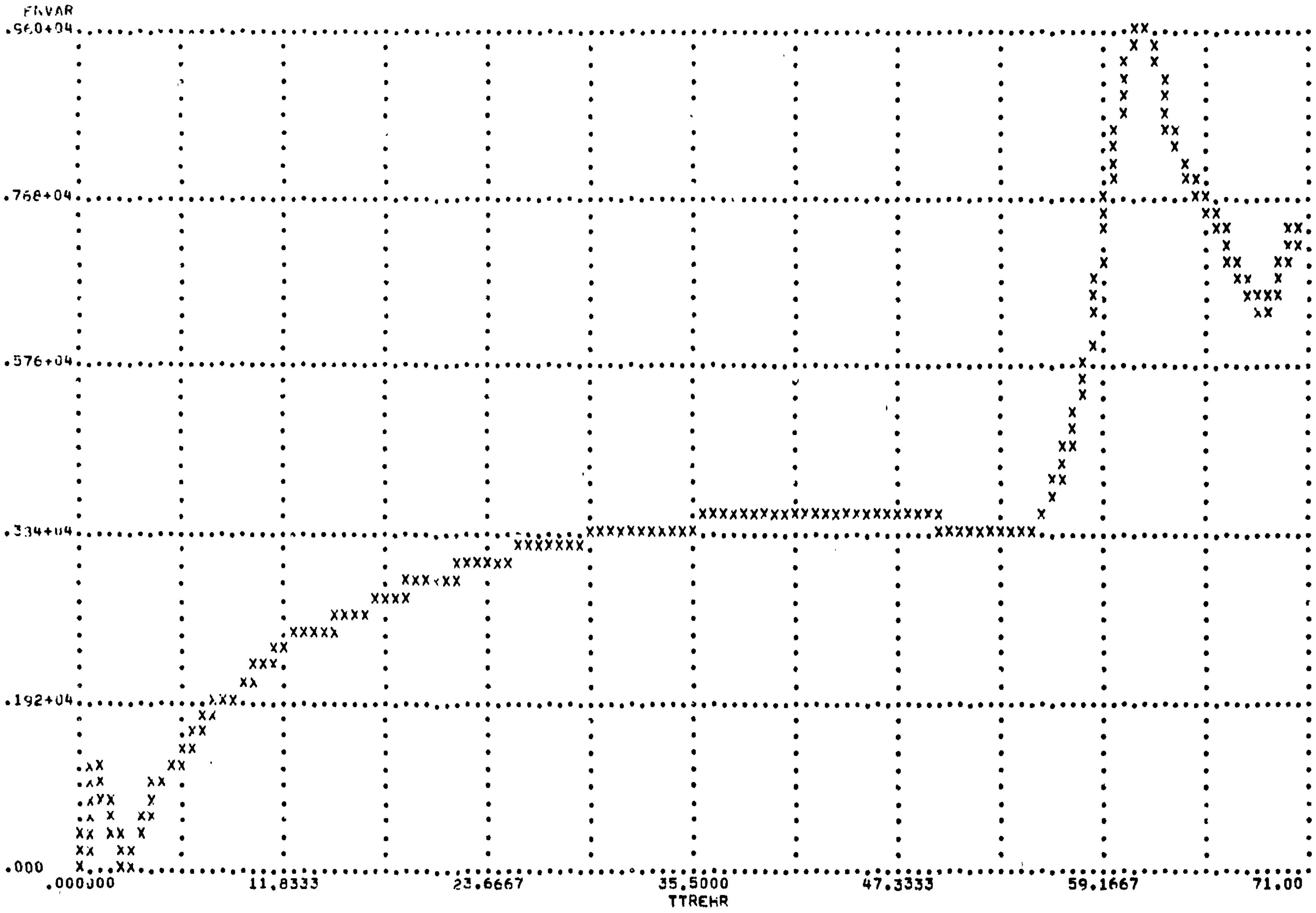


FIGURE 3 - A PTX EXAMPLE

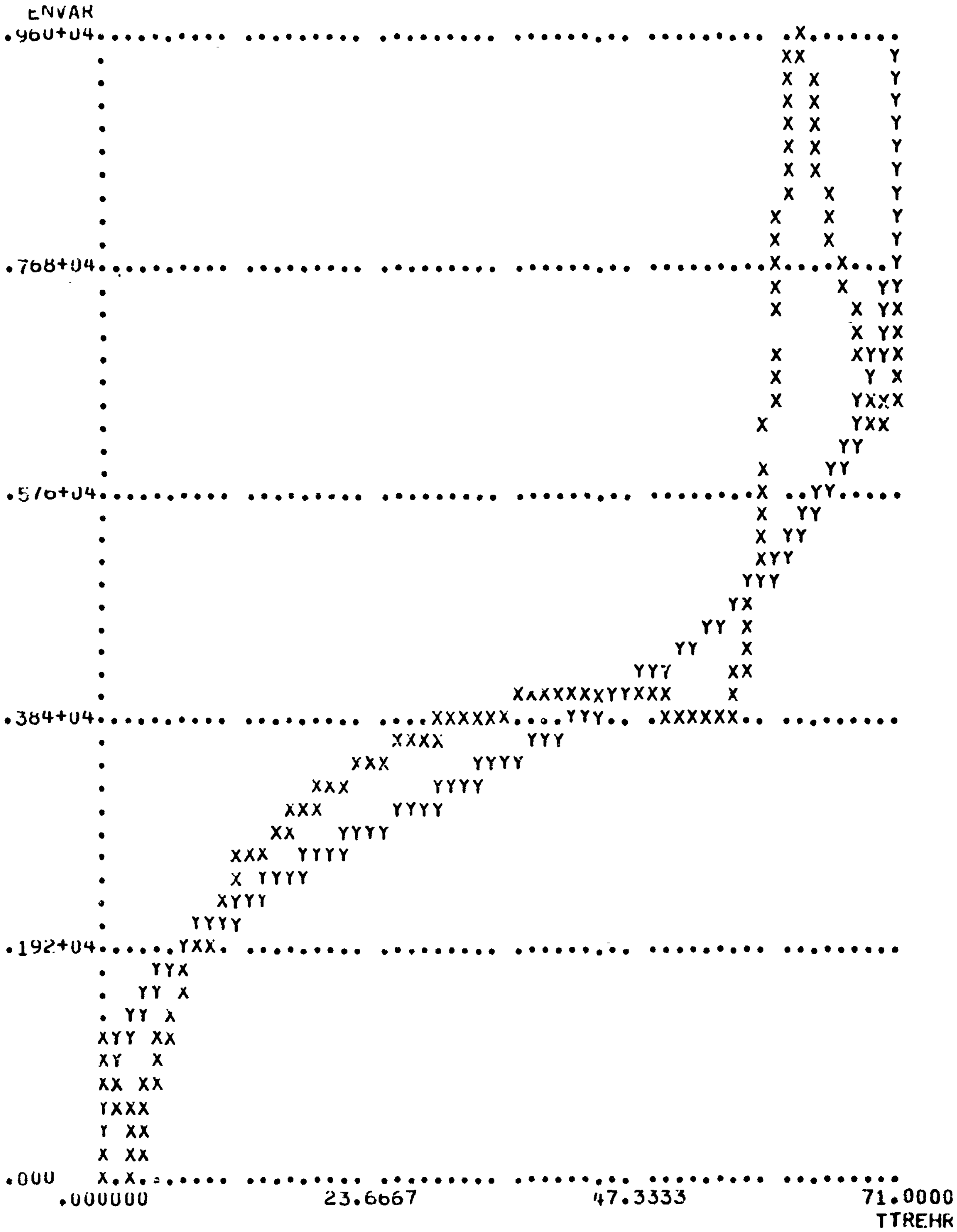


FIGURE 4 - A TT2X EXAMPLE

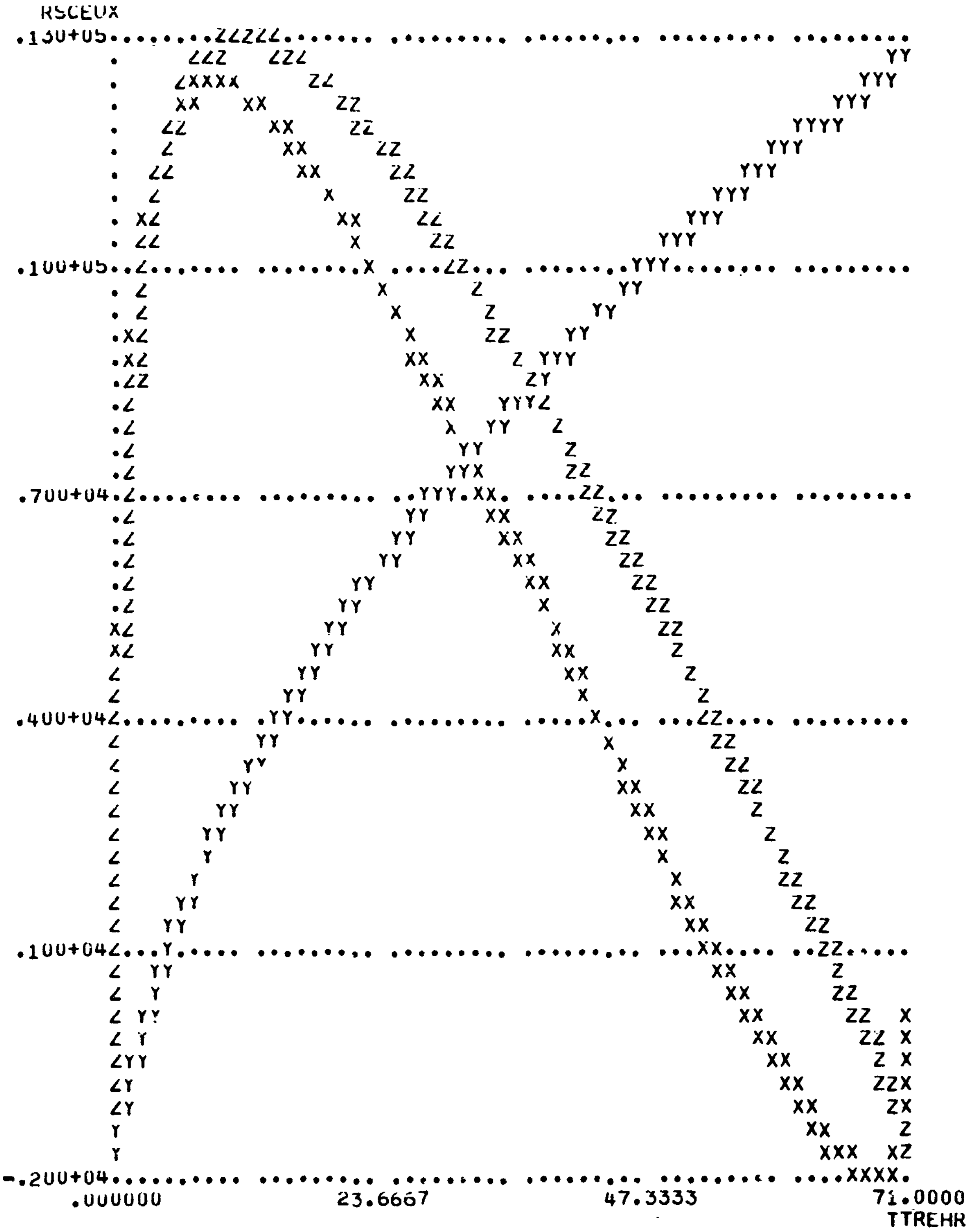


FIGURE 5 - A TT3X EXAMPLE. THIS EXAMPLE IS ALSO GIVEN WITH AN ADDITIONAL CURVE AS A PT4X GRAPH IN FIGURE 6

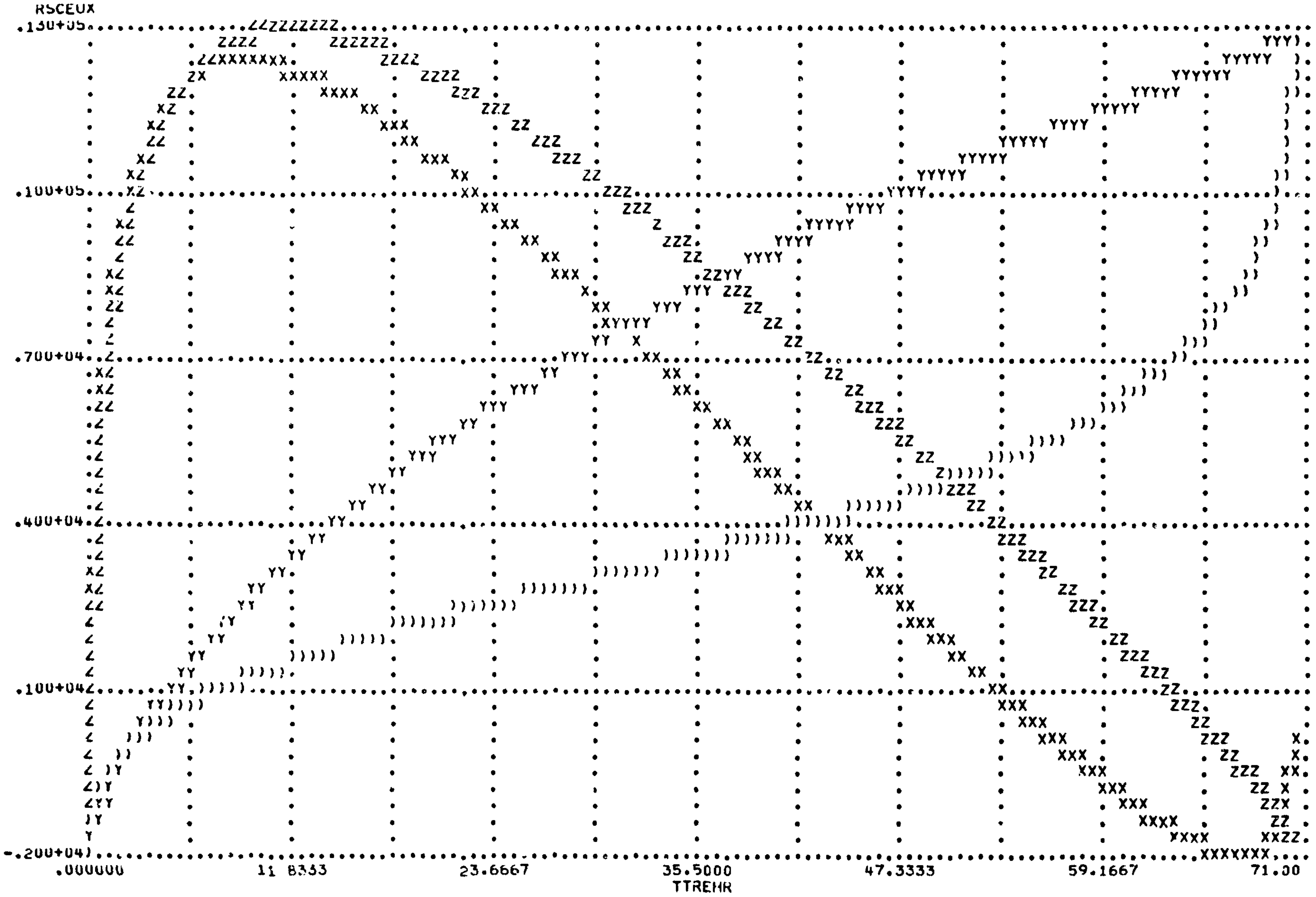


FIGURE 6 · A PT4X EXAMPLE

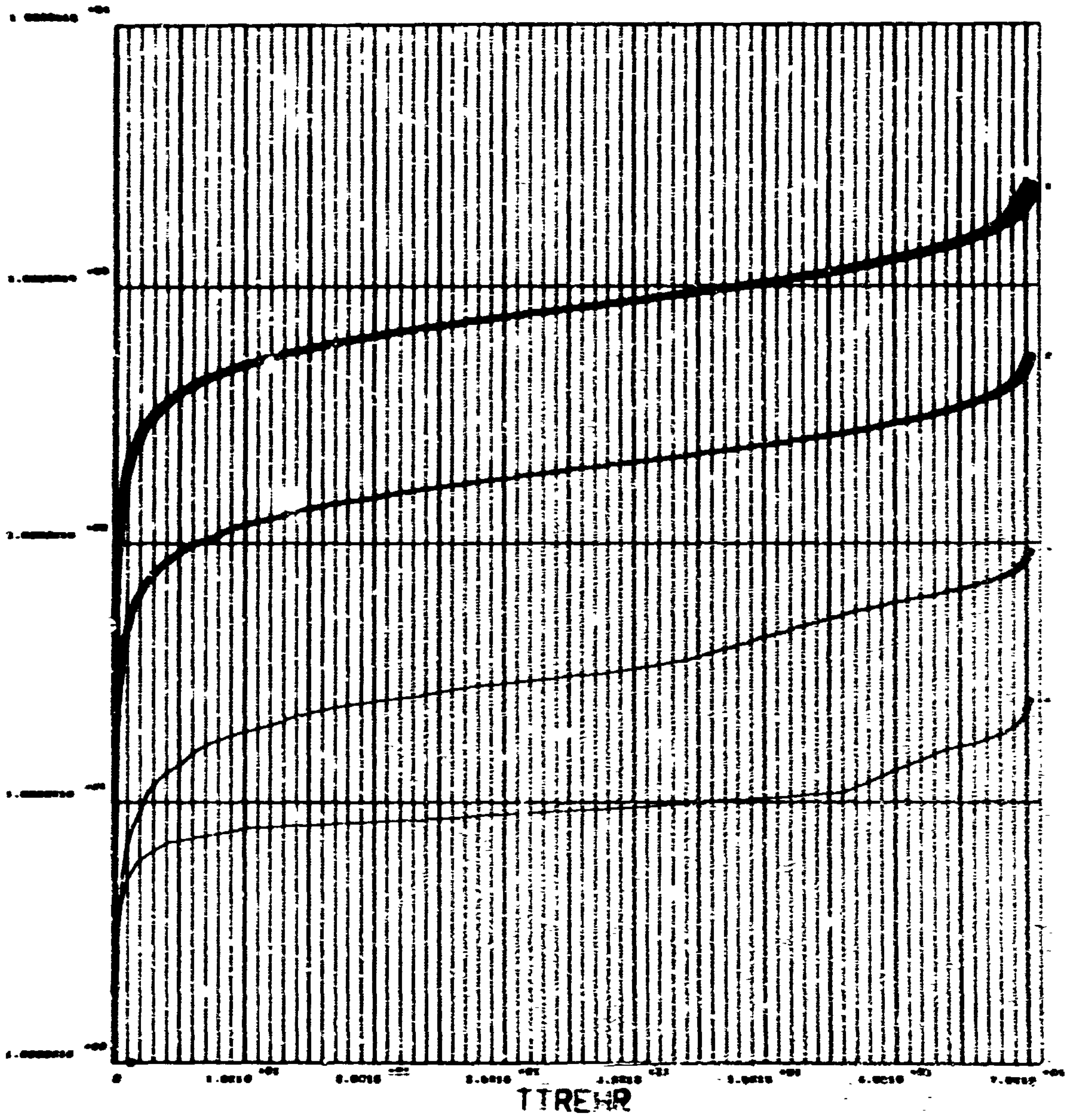


FIGURE 7 - A QTX EXAMPLE: 4 SUPERIMPOSED CURVES. CONSTRUCTED BY CALLS TO PLTSIM, PLTLOG, PLTAGS, AND QTX

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

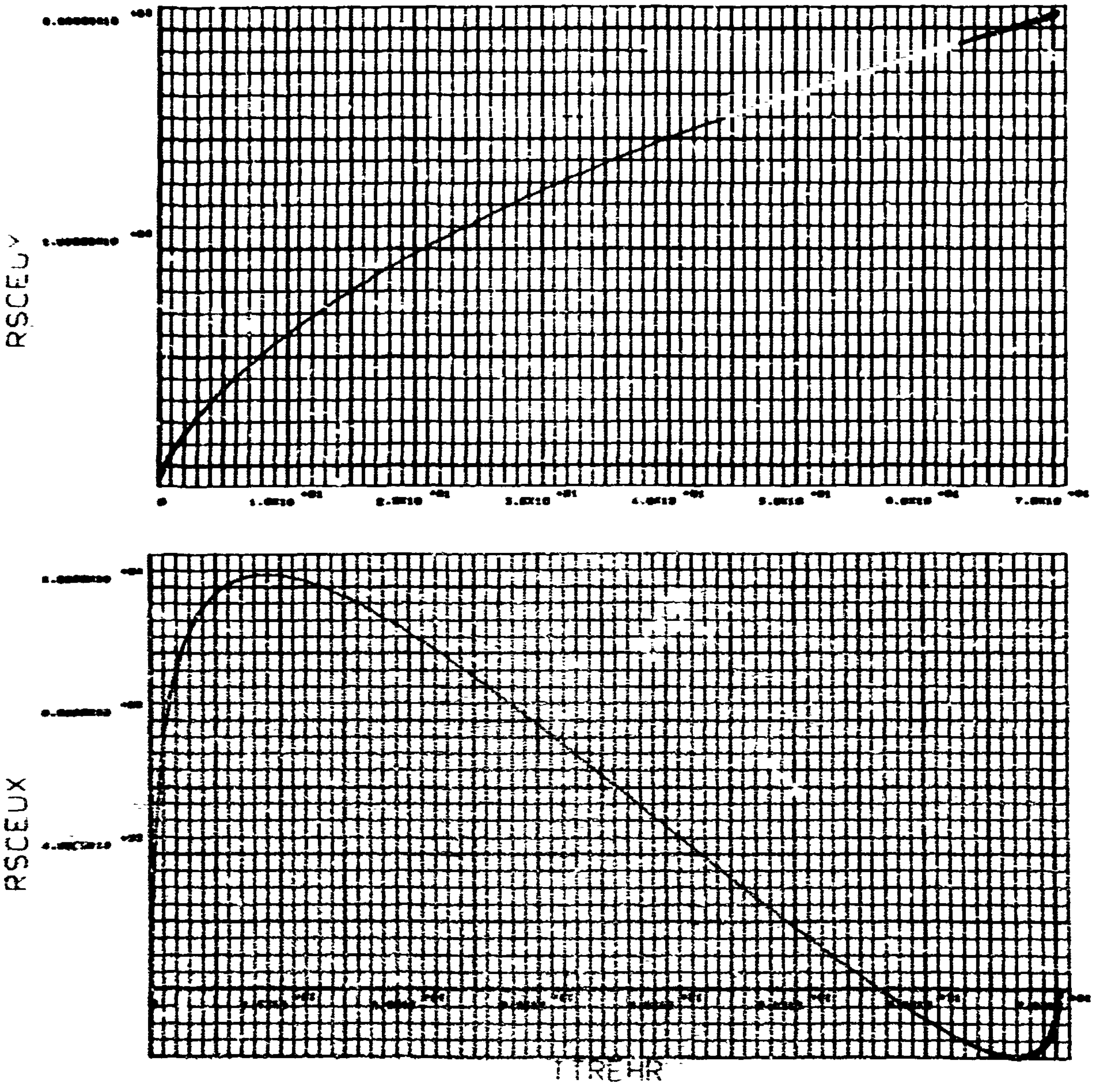


FIGURE 8 - A QTSXY EXAMPLE

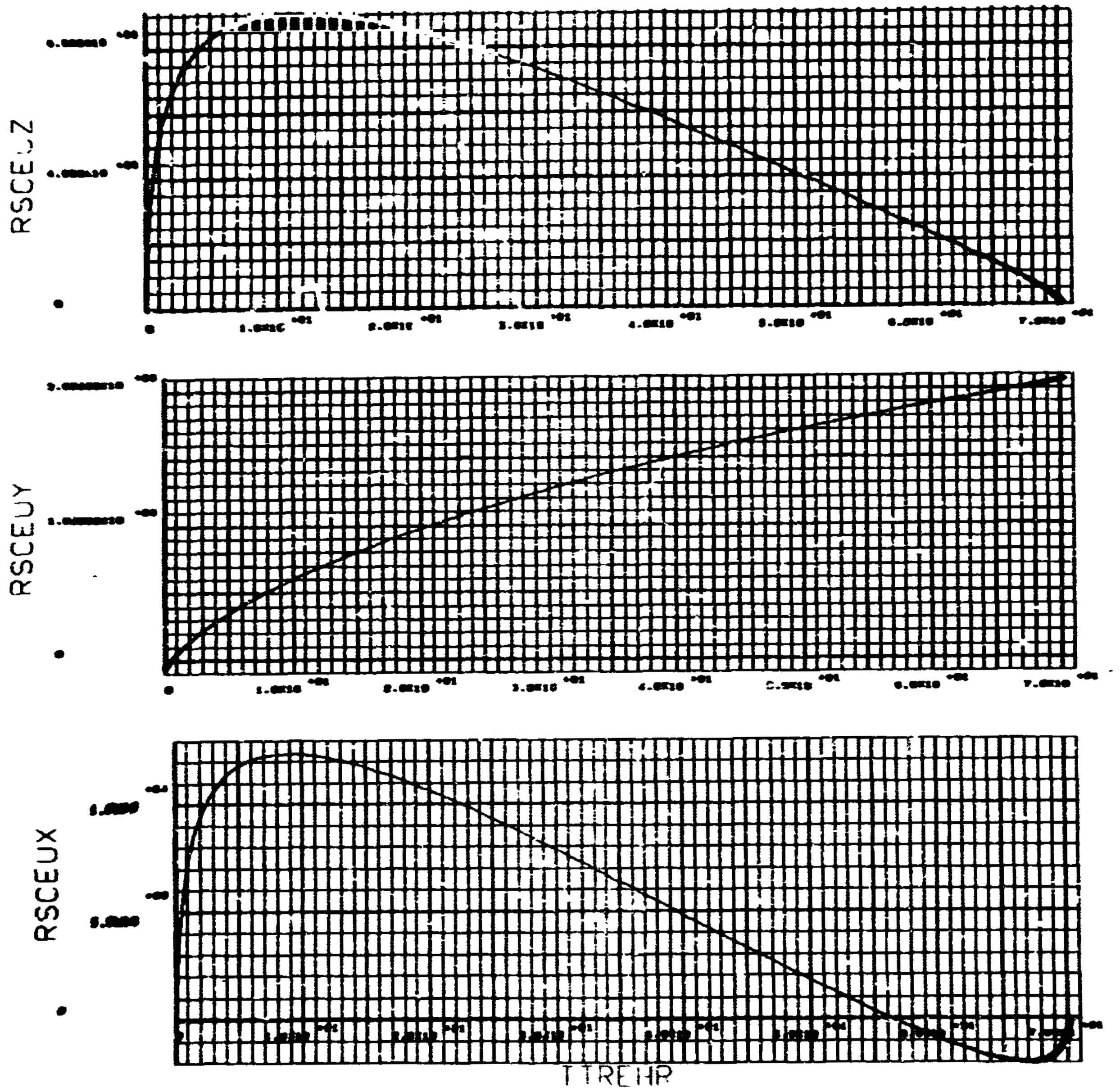


FIGURE 9 - A QTSXYZ EXAMPLE

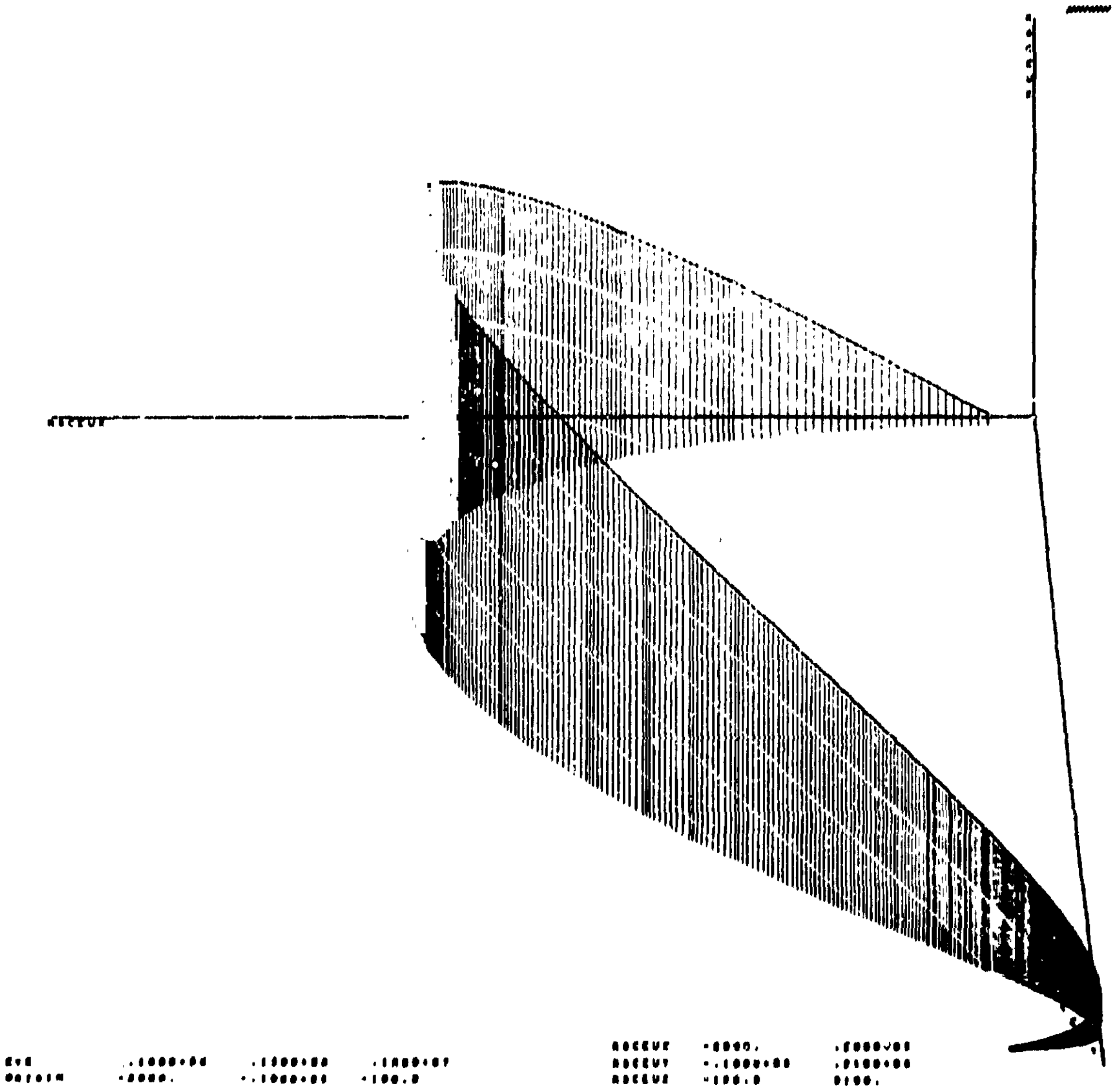


FIGURE 10 - A Q3DPER EXAMPLE. CONSTRUCTED BY CALLS TO Q3DEYE, Q3DORG, Q3DSHA, AND Q3DPER

TABLE 1

User Program Calls to Subroutines in AUPLOT

COLECT (i, a)	Reserve storage as necessary for the identifier i. Insert the current value of a.
COLIST (i, a, m)	Reserve storage as necessary for the identifier i. Insert the current array values a_1, a_2, ..., a_m.
COLIMP (i, a)	Reserve storage for the identifier i.
PLTOUT	Move to storage the current value of all variables whose space was reserved by COLIMP.
CULOUT (i)	Release storage reserved for the identifier i. If i = 0, release all reserved storage.
PLTICS (i, T_l, T_u)	For the data identifier i, set lower and upper plotting limits T_l and T_u, respectively.
PLWIND (i, w)	For the identifier i, set as plotting limits an interval of length w about the data midpoint.
PLTICH (i)	Hold current data bounds of i as plotting limits.
PLWINH (i)	Hold distance between current data bounds of i as plotting window.
PLTICA (i, j)	Assign to i the plotting limits of j.
PLWINA (i, j)	Assign to i the plotting window of j.
PLTLOG (i, m)	Assign to i the logarithmic plotting attribute m, m = 1. If m = 0, the attribute is reset to linear.

Table 1 (Contd.)

PLTDEN (i, d)	Assign to i the density attribute d. For d = 10., 50., 100., the background grid will be spaced at approximately .1, .5, and 1.0 inches apart, respectively. For d = 0, no grid will be drawn.
PTX (i_t, i_x)	Plot t versus x on the online printer.
TTX (i_t, i_x)	Plot t versus x on the time-sharing teleprinter.
QTX (i_t, i_x)	Plot t versus x on the off-line plotter (SC4020).
PT2X (i_t, i_x, i_y)	Plot overlay of t versus x and t versus y on the online printer.
TT2X (i_t, i_x, i_y)	Plot overlay of t versus x and t versus y on the teleprinter.
QTSXY (i_t, i_x, i_y)	Plot t versus x on the lower half and t versus y on the upper half frame of the off-line plotter.
PT3X (i_t, i_x, i_y, i_z)	Plot overlay of t versus x, t versus y, and t versus z on the on-line printer.
TT3X (i_t, i_x, i_y, i_z)	Plot overlay of t versus x, t versus y, and t versus z on the teleprinter.
QTSXYZ (i_t, i_x, i_y, i_z)	Plot t versus x on the lower third, t versus y on the middle third, and t versus z on the upper third frame of the off-line plotter.
PT4X (i_t, i_a, i_b, i_c. i_d)	Plot overlay of t versus a, t versus b, t versus c, and t versus d on the on-line printer.
TT4X (i_t, i_a, i_b, i_c, i_d)	Plot overlay of t versus a, t versus b, t versus c, and t versus d on the teleprinter.

Table 1 (Contd.)

Q3DPER (i_x, i_y, i_z)	Plot 2-dimensional perspective projection of the 3-dimensional data {(x, y, z)} via the off-line plotter.
Q3DEYE (e, m)	For Q3DPER plots, set the observer's position at e = (e_x, e_y, e_z), m = 1. If m = 0, let plotting subroutine choose observer's position.
Q3DORG (o, m)	For Q3DPER plots, set the reference axes origin at o = (o_x, o_y, o_z), m = 1. If m = -1, omit the axes. If m = 0, let the plotting subroutine choose for the origin o = min (x, y, z).
Q3DSHA (p, m)	For Q3DPER plots, draw shading lines between the data representation and the plane x = p, y = p, or z = p for m = 1, 2, or 3, respectively. For m = 0 discontinue shading.
PLTITL (t)	Initiate graph titling with t of up to 48 characters.
PLTSIM (i)	For i = 2, inhibit off-line plotter frame advance incorporated as first step of each plot instruction. For i = 1, inhibit after next plot. The i = 1 option is best suited to constructing off-line plotter overlays. For i = 0, restore the normal frame advance in each plot instruction.
AUFLAG (i, f)	Place flag f in Phase 2 COMMON array, IEXTRA(i).
AUFRAM (3)	Advance one frame on the off-line plotter.
AUPRIN (k_x, k_y, m_x, m_y, m, c)	Using the off-line plotter character generator, place the string c of m characters on the current plot frame beginning at coordinates (k_x, k_y). For each character after the first, increment the coordinates by (m_x, m_y).

Table 1 (Contd.)

AUOPEN	Open the off-line plot file.
AUCLOS	Close the off-line plot file.
AURITE $(k_x, k_y, m_r, m_o, m, c)$	Using the off-line plotter vector capability, place the string c of m characters on the current plot frame beginning at coordinates (k_x, k_y). Simulate a typewriter carriage return after passing each m_r caster units. Orient the string according to the key m_o.
AURSIZ $(c_x, c_y, f_x, f_y, k_t)$	For the subroutine AURITE, select character size (c_x, c_y), character field size (f_x, f_y) and character table k_t, $1 \leq k_t \leq 3$.
PLTDMP	Print contents of AUPLOT storage areas.
PLTEND	Indicate to the Phase 2 programs that plotting is completed.
PLTEMP	Indicate to the Phase 2 programs that plotting is temporarily halted.
APPLOT	Perform the AUPLOT Phase 2 operation as a subroutine call from the user's program.

Table 2

Instruction Record Formats for Phase 1 Subroutines

PLMOUT	2	1					
PLTOUT	NPLMAP+4	2	0	0	$PLTDAT_1$	. . .	$PLTDAT_{NPLMAP}$
PLTEND	2	140					
PLTEMP	2	142					
PLTDMP	3	144	'PLTPMD'				
PLTLOG (i, n)	6	155	'PLOTIC'	0	i	n	
PLTDEN (i, d)	6	156	'PLOTIC'	0	i	d	
PLWIND (i, w)	6	160	'PLOTIC'	0	i	w	
PLWINA (i, j)	6	162	'PLOTIC'	0	i	j	
PLWINH (i)	5	164	'PLOTIC'	0	i		
PLTICS (i, t_l, t_u)	7	170	'PLOTIC'	0	i	t_l	t_u
PLTICA (i, j)	6	172	'PLOTIC'	0	i	j	
PLTICH (i)	5	174	'PLOTIC'	0	i		
AULONG (i)	5	190	'APLONG'	1	i		
PLTSIM (i)	5	190	'APLONG'	2	i		
PLCHAR (i)	5	190	'APLONG'	3	i		
PLTAGS (i)	5	190	'APLONG'	4	i		

Table 2 (continued)

AUFLAG (i, f)	6	190	'APLONG'	4+i	f					
Q3DEYE (e, n)	8	190	'APPRJ1'	0	e_1	e_2	e_3	n		
Q3DSHA (p, n)	6	190	'APPRJ2'	0	p	n				
Q3DORG (o, n)	8	190	'APPRJ3'	0	0_1	0_2	0_3	n		
AUOPEN	4	190	'APOPEN'	0						
AUCLOS	4	190	'APCLOS'	0						
PLTITL (t)	12	190	'APTITL'	0	t_1	t_2	. . .	t_8		
AUFRAM (i)	5	190	'APFRAM'	0	i					
AUPRIN (k_x, k_y, m_x, m_y, m, c)	10	190	'APPRIN'	0	k_x	k_y	m_x	m_y	m	c
AURITE (k_x,k_y,i_r,i_o, m, c)	10	190	'APRITE'	0	k_x	k_y	i_r	i_o	m	c
AURSIZ (c_x,c_y,f_x,f_y, k_t)	9	190	'APRSIZ'	0	c_x	c_y	f_x	f_y	k_t	
PTX (i_t, i_x)	6	200	'PLTXYP'	0	i_t	i_x				
TTX (i_t, i_x)	6	200	'PLTXYT'	0	i_t	i_x				

Table 2 (continued)

QTX (i_t, i_x)	6	240	'XYQ'	0	i_t	i_x			
PT2X (i_t, i_x, i_y)	7	300	'PLTXYP'	0	i_t	t_x	i_y		
TT2X (i_t, i_x, i_y)	7	300	'PLTXYT'	0	i_t	i_x	i_y		
QTSXY (i_t, i_x, i_y)	7	340	'TSXYQ'	0	i_t	i_x	i_y		
Q3DPER (i_a, i_b, i_c)	7	340	'APPROJ'	0	i_a	i_b	i_c		
PT3X (i_t, i_x, i_y, i_z)	8	400	'PLTXYP'	0	i_t	i_x	i_y	i_z	
TT3X (i_t,i_x,i_y,i_z)	8	400	'PLTXYT'	0	i_t	i_x	i_y	i_z	
QTSXYZ (i_t,i_x,i_y,i_z)	8	440	'TSXYZQ'	0	i_t	i_x	i_y	i_z	
PT4X (i_t,i_a.i_b,i_c,i_d)	9	500	'PLTXYP'	0	i_t	i_a	i_b	i_c	i_d
TT4X (i_t,i_a,i_b,i_c,i_d)	9	500	'PLTXYT'	0	i_t	i_a	i_b	i_c	i_d

Table 3

Definitions of AUPLOT COMMON Symbols

Symbol	Definition	Program Appearances
CAXIS	Orientation vector for determining "up" direction in perspective plots	PER3DQ
DATA	Phase 2 data storage table	APPLOT, COMMON PER3DQ PLOTIC, PLTBCU, PLTPMD, PLTXYP, PLTXYQ
EYE	Position of observer in perspective plots	PER3DQ, Q3DPER
ICHAR	Set of 3 characters for grid lines, background, and point plots, respectively, in printer plots	PLTXYP, COMMON, PLTPMD
1COMPO	Control flag for composing QTSXY and QTSXYZ plots	PLOTQS, PLTXYQ, COMMON, PLTPMD
IDATA	Integer equivalent of table DATA	APPLOT, PER3DQ, PLOTIC PLTXYP, PLTXYQ, COMMON
IDELTA	Table of possible block assignments in Table DATA	APPLOT, COMMON
IDLINK	Table of actual block assignments and data indices for table DATA	APPLOT, PER3DQ, PLOTIC PLTBOU, PLTPMD, COMMON
IEXTRA	Table of vacant storage locations for use in system extensions	AULONG, COMMON, APLONG PLTPMD
IOPLT	Fortran value of the intermediate file for Phase 1 to Phase 2 transition	APPLOT, AULONG, AUPINS, COMMON, PLATER, PLMOUT PLOTIC, PLTDMP, PLTPMD
IPCHAR	Data point plotting character	APLONG, AULONG, COMMON PER3DQ, PLTXYP, PLTXYQ

Table 3 (Continued)

Symbol	Definition	Program Appearances
IPLCOL	Data collection code	COLECT, COMMON, PLTDMP
IPLDAT	Data collection buffer (integer form)	COMMON, PLTOUT
IPLMAP	Plot map	APPLOT, COLECT, COMMON PER3DQ, PLMOUT, PLOTIC, PLTDMP, PLTOUT, PLTPMD, PLTXYQ, PLTXYP
IPLONG	Control flag for composing long plots	APLONG, AULONG, COMMON, PLTPMD
IPLTAG	Six character tag for identifying successive curves of an overlay	APLONG, AULONG, COMMON, PLTPMD, PLTXYQ
ISUPIM	Control flag for composing overlays	APLONG, APPLOT, AULONG, COMMON, PER3DQ, PLTPMD, PLTXYQ, PLOTQS
IT	Table of block starting points in the table DATA for variables to be plotted	APPLOT, COMMON, PER3DQ PLTPMD, PLTXYP, PLTXYQ, PLOTQS
ITAPE	Storage table for intermediate file records (integer form)	APLONG, APPLOT, APPRIN APTITL, AULONG, AUPINP, AUPRIN, COMMON, PER3DQ PLATER, PLMOUT, PLOTIC, PLTDMP, PLTICS, PLTITL, PLTOUT, PLTXYP, Q3DPER, PLOTQS
ITITLE	Table of 48 characters to appear as a title for each built-in plot	APPLOT, APTITL, COMMON, PER3DQ, PLTITL, PLTPMD, PLTXYP, PLTXYQ,
I1	Table of plot map entries for variables to be plotted	APPLOT, APPRIN, COMMON, PLTPMD
JT	Table of overflow counters for variables to be plotted	APPLOT, COMMON, PER3DQ, PLTPMD
J1	Table of subscript step sizes for variables to be plotted	APPLOT, COMMON, PER3DQ, PLTPMD, PLTXYP, PLTXYQ, PLOTQS
LCAM99	Logical indication that the SC4020 PLOT file is open	APPLOT, APPRIN, COMMON, PLTPMD

Table 2 (Continued)

Symbol	Definition	Program Appearances
LIDOVR	Logical indication that the plot map has overflowed	COLECT, COMMON, PLMOUT, PLTPMD
LOGOUT	Logical indication that a subroutine walkback has been initiated	APPLOT, APPRIN, COMMON, PER3DQ, APMAIN, PLOTIC, PLTDMP, PLTNOW, PLTXYQ, PLOTQS
LPLMAP	Logical indication of a plot map change and that it should be forwarded to Phase 2 prior to the next collection buffer	COLECT, COMMON, PLMOUT PLTDMP, PLTOUT
LPLNOW	Logical indication that the APPLOT mode is time-sharing	APPLOT, COMMON, PLMOUT, PLTDMP, PLTNOW
MDATA	Maximum size of the table DATA	APPLOT, COMMON
MDELTA	Maximum size of the table IDELTA, the number of assignable blocks in DATA	APPLOT, COMMON
MDLINK	Maximum number of components of each entry in the table IDLINK	APPLOT, COMMON, PLTPMD
MPLARY	Maximum number of variables to be plotted in a single plot subroutine call	APPLOT, COMMON, PLTXYP
MPLMAP	Maximum number of data identifiers which may appear in the plot map	APPLOT, COLECT, COMMON, PLMOUT
MSPEC	Maximum number of data attributes allowed in each block of the table IDATA	APPLOT, COMMON, PER3DQ, PLTBOU, PLTPMD, PLTXYP, PLTXYQ
MTITLE	Maximum word count of the table ITITLE.	APPLOT, APTITL, COMMON, PLTDMP, PLTITL, PLTPMD

Table 3 (Continued)

Symbol	Definition	Program Appearances
NDELTA	Index to smallest block of the table IDELTA currently in use	APPLOT, COMMON, PLTPMD
NOVER	Overlay indicator for the SC4020 program GRIDIV	COMMON, PLTPMD, PLTXYQ PLOTQS
NPLARY	Number of variables being plotted by PLTXYP	APPLOT, COMMON, PLTXYP
NPLMAP	Number of data identifiers in current plot map	APPLOT, COLECT, COMMON, PLMOUT, PLOTIC, PLTDMP, PLMOUT, PLTPMD
NPTS	Number of points in current plot	APPLOT, COMMON, PLTPMD, PLTXYP, PLTXYQ, PLOTQS
NTAPE	Word count of intermediate file record	APPLOT, APPRIN, COMMON, PLATER, PLTDMP, PLTPMD
PLTDAT	Data collection buffer (real form)	COLECT, COMMON, PLTDMP, PLTOUT
TAPE	Intermediate file record (real form)	APPLOT, COMMON, PER3DQ, PLMOUT, PLOTIC, PLTDMP, PLTICS, PLTOUT, Q3DPER

Table 4

AUPLOT References to SC4020 Library Subroutines

Subroutine	Purpose	Program Appearances
APRNTV	Print with the SC4020 CHARACTRON feature	APPRIN, AUPRIN PER3DQ, PLTXYQ
BIGV	Set cathode ray picture area to big size	APLONG
CAMRAV	Select camera and initialize SC4020 output	APPRIN
CHSIZV	Set size of vector drawn characters	APPRIN, AUPRIN
DXDYV	Calculate GRIDIV parameters	PER3DQ, PLTXYQ
EOFTV	Close SC4020 output	APPRIN
GRIDIV	Determine data units to raster units scaling and draw background grid	PER3DQ, PLTXYQ
FRAMEV	Advance camera film	APPRIN, AUPRIN
HOLDIV	Hold plotting margins	APLONG
LINEV	Draw line segment between two points	PER3DQ, PLTXYQ
NOFRV	Give SC4020 frame count	APPRIN
NXV	Convert a number in X direction to raster unit	APPRIN, PER3DQ, PLTXYQ
NYV	Convert a number in Y direction to raster unit	APPRIN, PER3DQ, PLTXYQ

Table 4 (Contd.)

PØINTV	Plot data point	PLTXYQ
RITE2V	Write with vector drawn characters	APPRIN, AUPRIN PER3DQ, PLTXYQ
RITSTV	Select RITE2V character table	APPRIN, AUPRIN
SETMIV	Set plotting area margins	PER3DQ, PLTXYQ
SMALLV	Set camera field of view to small (standard) size	APLONG
SMXYV	Set logarithmic scaling indicators	PLTXYQ
TABLiV	RITE2V character tables, i = 1, 2, 3. TABLE1V is standard	APPRIN

AUPLOT Errors and Diagnostic Printouts

Table 5

COLECT *** IDENTIFIER OVERFLOW

This message indicates an overflow of the plot map data identifier list, which allows 64 entries in the standard system. A dump is initiated to show the plot map contents. User remedies include switching to a larger capacity AUPLOT file and use of the CULOUT feature to discard identifiers no longer needed.

APPLOT *** CALLS TO BUILT-IN PLOT SUBROUTINES

This message is initiated in Phase 2 in response to a Phase 1 PLTEMP or PLTEND instruction. It gives number of calls which have been made to the built-in subroutines.

APPLOT *** DATA OVERFLOW m i

This message indicates inability to assign Phase 2 storage to the data identifier i. User remedies include switching to a larger capacity AUPLOT file and use of the CULOUT feature to release storage for identifiers no longer needed.

APPLOT *** DATA TRUNCATION IN i

This message indicates overflow of the Phase 2 storage block assigned to i. Data truncation is initiated to enable continued processing and plotting. Data truncation results in plot preparation from each 2^n value of the original data associated with i, where n is the number of Phase 2 overflows in i. The user may request intermediate plots containing i, or he may use the PLTDMP feature to determine whether any ill effects are introduced.

s *** NO AUPLOT ELEMENT

A Phase 1 instruction subroutine has initiated a Phase 2 call to an undefined subroutine s. Check ITAPE_3 in each user coded instruction subroutine; make sure this name is edited into table ICALL of APLINK. Otherwise set ITAPE_3 to one of the unused ICALL entries, APPLT1, APPLT2, . . .,

Table 5 (Contd.)

APPLT6, PLOTU.

APPLOT *** NO COLLECTIONS FOR i

A plot request cannot be honored in Phase 2 due to inability to find the data identifier in the plot map. Check plotting instructions for misspelled data identifiers. Check also for APPLOT data overflow message.

APRITE *** RITE2V ERROR ON c

An off-scale condition or other RITE2V error has occurred in Phase 2 while printing the character string c. The subroutine AURITE may use floating point coordinates only after a plotting instruction call. Coordinates indicate the first character midpoint. Placement of the midpoint such that the character overflows the plot margin causes a RITE2V error.

APRSIZ *** TABLE INDEX OUT OF RANGE

This message is initiated whenever other than SC4020 vector character tables TABL1V, TABL2V, or TABL3V are selected. Indices 1, 2, or 3 are valid.

APCLOS *** SC4020 FRAME COUNT i k

This message is initiated by the Phase 1 instructions AUCLOS and PLTEND to show the SC4020 frame count as maintained by the SC4020 subroutine NOFRV.

APPROJ *** IMPOSSIBLE GRID

This message indicates an SC4020 subroutine GRID1V error return in the 3-dimensional plotting subroutine. A Phase 2 dump may be initiated to show data attributes. Check plotting limits for each 3-D variable.

PLOTIC *** ATTRIBUTE ASSIGNED FROM DATA NOT COLLECTED

The second argument of a PLTICA or a PLWINA instruction is missing from the plot map. Data attributes may be assigned only after collection of the data has been initiated.

Table 5 (Contd.)

PLOTIC *** ATTRIBUTE ASSIGNED TO DATA NOT COLLECTED

The first argument of one of the data attribute subroutines is missing from the plot map. Data attributes may be assigned only after collection of the data has been initiated.

PLTXYQ *** IMPOSSIBLE GRID

This message indicates an SC4020 subroutine GRID1V error return in the basic rectangular plotting subroutine. A Phase 2 dump may be initiated to show data attributes. Check plotting limits for variables being plotted.

Table 6

AUPLOT Program Storage Requirements

Element	1108 Storage Word Requirements					
	Code	Data	CPLT	CAUP	CVAR	Total
COMMON$_8$*			61	8357	1	
COMMON$_{64}$			421	25625	1	
COMMON$_{256}$			1549	36189	1	
AULONG	77	19	✓			
AUPINS	253	15	✓			
AUPRIN	137	30	✓			
COLECT	218	24	✓			
INDRSS	4				✓	
PLMOUT	69	16	✓			
PLTDMP	123	73	✓			
PLTICS	119	13	✓			
PLTITL	34	13	✓			
PLTOUT	90	16	✓		✓	
Q3DPER	156	24	✓			
PLATER	25	10	✓			
Totals**						
Phase 1_8	1305	253	'61		1	1620
Phase 1_{64}	1305	253	421		1	1980
Phase 1_{256}	1305	253	1549		1	3108

Table 6 (Contd.)

APMAIN	14	6	✓			
APPLOT	581	119	✓		✓	
APLINK	15	53				
APLONG	39	6	✓		✓	
APPRIN	213	35	✓		✓	
APTITL	43	14	✓		✓	
PER3DQ	945	148	✓		✓	
PLOTIC	208	44	✓		✓	
PLOTQS	184	11	✓		✓	
PLTBOU	66	14	✓		✓	
PLTPMD	142	178	✓		✓	
PLTRND	176	11				
PLTXYP	318	201	✓		✓	
PLTXYQ	484	51				
Totals**						
Phase 2_{8}	3428	891	61	8357		12737
Phase 2_{64}	3428	891	421	25625		30365
Phase 2_{256}	3428	891	1549	36189		42057

*The subscripts 8, 64, and 256 refer to AUPLOT files J8, J64, and J256, respectively.

**For separate execution of Phase 2. When Phase 2 is activated through user's program subroutine call or is merged with Phase 1 for interactive operation, the storage requirement becomes approximately Phase 1 and Phase 2 totals combined.