

SHARE PROGRAM LIBRARY AGENCY

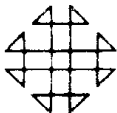


PROGRAM NUMBER

086003

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257



CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM
(for IBM S/360, 1130 and 1800)

SHARE Program Library Agency
Triangle Universities Computation Center
P. O. Box 12076
Research Triangle Park, N. C. 27709

This form should be completed and submitted with the program package to PID at the address shown above. Standards and instructions for submitting programs are in your *User Group Reference Manual* or the *Contributed Program Submittal Standards Manual* available from PID.

- ① Program Order Number (to be filled in by PID) 360D-08.6.003
- ② System Type (machine) S / 360
- ③ Search Key / F O R T R A N / / S U B R O U T I N E /
/ F O R / P L O T T I N G / O N / / P R I N
T E R /
- ④ Name of Author (if different than submitter's) LAWRENCE ISRAEL
- ⑤ Submitter's Name (direct technical inquiries to) LAWRENCE ISRAEL
- ⑥ Submitter's Address
Technical Assistance
Currently Not Available
- ⑦ Title of Program PLOT: A SUBROUTINE FOR
PLOTTING ON A PRINTER
- ⑧ Submitter's User Group Affiliation Code and Installation Code S A P C
- ⑨ Submitter's Own Program Identification and Suffix (optional)
- ⑩ Primary Subject Code 08.6
- ⑪ Secondary Subject Codes
- ⑫ Operating or Monitor System Required N O N E
- ⑬ New or Revision Code (if revision, show prior Program Order Number in item 1) N
- ⑭ Year Completed 67
- ⑮ Date of Submittal 1.0.0.1.6.7
- ⑯ Documentation (number of original pages submitted) 17
- ⑰ Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide

- Purpose
- Programming Language used
- Version and modification level or release number of IBM Programming System used, or program order number for non-IBM authored program used
- Field of application
- Type of routine (main program, subroutine, etc.)
- Specific description of machine requirements
- Engineering Changes (EC) level of equipment (if pertinent)

ABSTRACT

THE SUBROUTINE PLOT IS USED FOR PLOTTING ON A PRINTER. IT WILL PRINT ONE TO NINE SETS OF DEPENDENT VARIABLES AGAINST AN INDEPENDENT VARIABLE, AND/OR A CURVE OF CALCULATED VALUES.

THE PROGRAM IS WRITTEN IN BASIC FORTRAN IV FOR THE S/360.

DISCLAIMER

Triang's University Communication Center (TUCC) serves solely as the filing location for submitted programs and does not warrant, endorse, or assume any liability for the accuracy, reliability, or performance of the contributed programs.

(Please attach additional pages if necessary) Total pages attached _____

Permission to Publish

"I hereby give anyone permission to reprint, reproduce, and distribute this program to anyone else."

- (18) Signature of Submitter and Date *[Signature]* 10/1/67
- (19) Signature of Installation Addressee *[Signature]* 10/1/67

T4SF

PLOT: A SUBROUTINE FOR
PLOTTING ON A PRINTER

TABLE OF CONTENTS

Deck Key	4
Purpose	5
Method	5
Language	6
Sample Program	6

Direct Inquiries To:
Lawrence Israel
Air Products & Chemicals, Inc.
P. O. Box 538
Allentown, Pennsylvania 18104

DECK KEY

Deck #1 Source Deck; sequence 0000 through 2070 in cc 77-80; PLOT in cc 73-76; 208 cards.

Deck #2 Sample Problem; sequence 0010 through 0120 in cc 77-80; SAMP in cc 73-76; 12 cards.

PURPOSE: This subroutine will plot, on a printer, from one to nine curves from a table of one to nine dependent variables and an independent variable, and/or plot a calculated curve when given the name of a subroutine to calculate Y from X.

This program was inspired by the IBM Scientific Subroutine Package PLOT routine.

METHOD: The subroutine is called as follows:

CALL PLOT (N0,A,N,M,IFUNC,FUNC,XMAX,XMIN,YMAX,YMIN)

where N0 is the chart number.

A is the location of an array, consisting of M sets of N numbers.

The first set are the values of the independent variable, X.

The second and succeeding sets are the values of the dependent variables, the Y's.

N is the number of values of each of the variables. If N = 0, this means no tabulated points are to be plotted.

M is the number of variables, dependent and independent; maximum value is 10.

IFUNC is zero or negative if no calculated curve wanted, positive if one is wanted.

FUNC is the name of a subroutine of the form SUBROUTINE FUNC (X,Y) which returns a value of Y when given a value of X. The program calling PLOT must have an EXTERNAL FUNC statement.

XMAX, XMIN are given to indicate the extreme values of X for the plot.

If XMAX = XMIN, plot calculates the extreme values of the graph from the tabulated points.

YMAX, YMIN are similar to XMAX and XMIN, but for Y.

The program prints the chart number. It then calculates the extremes of X and Y, unless these are specified. A rough scale is calculated, assuming 50 divisions vertically and 100 horizontally. A subroutine, SCAL, then reduces the rough scale to 1, 2, 2.5, or 5 times 10 to an integral power. The end points are adjusted to be multiples of the scale.

If a calculated curve is specified, an array, CALC, is filled with values of Y for each value of X.

YPR, the print position, is set to the maximum value of Y, and all independent variable sets, including calculated values, are searched for values of Y within one half scale division of YPR. If any are found, an integer (1-9) is printed on that line at the proper X value. The value of the integer is that of the set of dependent variables whose Y value was matched. An asterisk is printed for a calculated value. The search is done first on calculated value, then on ascending sets of Y. Only the last match found, if any, prints.

The print position of YPR is decremented by the scale, and the process repeated until the minimum value of Y is reached.

LANGUAGE: Subroutine PLOT and its associated subroutine SCAL are written in S/360 Basic Fortran IV. Alphanumeric characters (integers and asterisks) are derived from fixed point calculations, so extreme care should be used when converting the program to other machines.

SAMPLE PROGRAM: The sample program was run with BPS Fortran. The sample main program generates a few values of $Y = \sin X$, and gives as the calculation subroutine a program which returns $Y = \sin X$ when given X .

BPS FORTRAN IVD COMPILER VERSION 3 LEVEL 0 DEC 1966

```

/JOB      MULTIPLE.GO
/FTC      NAME=TEST,NODECK,LIST,MAP,SIZE=64K

```

```

BEGIN COMPILATION

```

```

  DIMENSION P(10)

```

```

  EXTERNAL SINUS

```

```

  DO 1 I=1,5

```

```

    P(I) = I - .001

```

```

    1 P(I+5) = SIN (P(I))

```

```

    CALL PLOT (1,P,5,2,1,SINUS,4.999,.001,.999,-.999)

```

```

    CALL EXIT

```

```

  END

```

```

SAMP0010
SAMP0020
SAMP0030
SAMP0040
SAMP0050
SAMP0060
SAMP0070
SAMP0080

```

STORAGE MAP VARIABLES (TAG C = COMMON, E = EQUIVALENCE)

NAME	TAG	REL ADR	NAME	TAG	REL ADR	NAME	TAG	REL ADR
P		000090	I		000088			

EXTERNAL REFERENCES

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
SIN	000080	PLOT	000080	EXIT	000084

CONSTANTS

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
CCCC001 414FFBF7	000094 0000E4	CCCC0005 41FFBF76	0000DB 0000EB	CCCC0002	0000DC

IMPLIED EXTERNAL REFERENCES

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
TACOM	CC0124				

STATEMENT NUMBER	REL ADR	STATEMENT NUMBER	REL ADR	STATEMENT NUMBER	REL ADR
00001	0001B2	SIZE OF COMMON	00009	PROGRAM	004B8

```

END OF COMPILATION TEST
COMPILED TIME WAS 0011.55 SECONDS

```

BEGIN COMPILATION

S.0001 SUBROUTINE SINUS (X,Y)
 S.0002 Y=SIN(X)
 S.0003 RETURN
 S.0004 END

SAMP0090
 SAMP0100
 SAMP0110
 SAMP0120

STORAGE MAP VARIABLES (TAG C = COMMON, E = EQUIVALENCE)

NAME	TAG	REL ADR	NAME	TAG	REL ADR	NAME	TAG	REL ADR
------	-----	---------	------	-----	---------	------	-----	---------

X		000000	Y		000004			
---	--	--------	---	--	--------	--	--	--

EXTERNAL REFERENCES

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
------	---------	------	---------	------	---------

SIN	000008				
-----	--------	--	--	--	--

CONSTANTS

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
------	---------	------	---------	------	---------

IMPLIED EXTERNAL REFERENCES

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
------	---------	------	---------	------	---------

IFCOM	00000C				
-------	--------	--	--	--	--

STATEMENT NUMBER REL ADR STATEMENT NUMBER REL ADR STATEMENT NUMBER REL ADR

END OF COMPILATION SINUS
 SIZE OF COMMON 00000 PROGRAM 00274
 COMPILE TIME WAS 0009.03 SECONDS

BEGIN COMPILATION
S.00001

```

SUBROUTINE PLOT (NC,A,N,M,IFUNC,FUNC,XLAX,XLIN,YLAX,YLIN)
PURPOSE - TO PLOT A GRAPH WITH ONE INDEPENDENT VARIABLE AND UP
TO 9 DEPENDENT VARIABLES, WITH THE ADDITIONAL ABILITY TO PLOT A
CALCULATED CURVE. THE INDEPENDENT VARIABLE IS PLOTTED ON A
HORIZONTAL AXIS, THE DEPENDENT ONES ON A VERTICAL AXIS. WIDTH
IS 100 PRINT POSITIONS, HEIGHT IS 50. EVERY POINT OF EACH
DEPENDENT VARIABLE IS INDICATED BY A NUMBER (1-9), WHILE THE
CALCULATED POINTS ARE DENOTED BY ASTERISKS.
PARAMETER USAGE.
  NO  A FIXED POINT NUMBER, UP TO 3 DIGITS, PRINTED AS THE
      CHAPT NUMBER
  A   A VECTOR WHOSE FIRST N POSITIONS CONTAIN THE INDEPENDENT
      VARIABLE, AND WHOSE NEXT M SETS OF N POSITIONS CONTAIN
      THE DEPENDENT VARIABLES
  N   NUMBER OF OBSERVATIONS
  M   NUMBER OF VARIABLES (INDEPENDENT & DEPENDENT)
  IFUNC GREATER THAN ZERO IF A CALCULATED CURVE IS TO BE
      PRINTED
  FUNC SUBROUTINE TO GENERATE CALCULATED CURVE. IF ONE WANTED,
      ELSE IS A DUMMY. PROGRAM CALLING PLOT MUST HAVE AN
      EXTERNAL FUNC. SUBROUTINE CALLED BY CALL FUNC (X,Y),
      WHERE X IS GIVEN TO SUBROUTINE AND Y RETURNED.
  XLAX,XLIN,YLAX,YLIN MAXIMUM AND MINIMUM VALUES OF THE
      INDEPENDENT AND DEPENDENT VARIABLES TO BE USED IN THE
      PLOT IF XLAX = XLIN, THE PROGRAM CALCULATES ITS OWN
      MAXIMUM AND MINIMUM FOR THE INDEPENDENT VARIABLE.
      SIMILARLY FOR YLAX = YLIN
REQUIRED SUBROUTINES: FUNC (IF USED), AND SCAL.
DIMENSION IOUT(1),XPR(1),A(1),CALC(12)
CALC IS 1 LARGER THAN NEEDED IN CASE XMIN + 100*XSCAL
SHOULD BE LARGER THAN XMAX. THIS PREVENTS SLOPES OVER INTO
NEXT LOCATION.      LOCK AROUND CARD =950 TO SEE WHAT I MEAN.
CALC IS WHERE CALCULATED FUNCTION GOES.
EQUIVALENCE (IOUT(1),XPR(1))
1 FORMAT(1H,40X,7H CHART ,I3)
118 FORMAT (1H ,12X,2H- ,101A1,*,*)
2 FORMAT (1H ,E12.3,*,*,101A1,*,*,E12.3)
7 FORMAT (1H ,14X,10(10*.....),1H+)
8 FORMAT(1HC,9X,11E10.3)
PRINT CHART NO.
WRITE (3,1) NO
TCOUNT = 4
IF NO EXTREMES OF X GIVEN, FIND THEM
IF (XLAX - XLIN) 20,13,20
10 XMIN = A(1)
XMAX = XMIN
DO 15 J = 1,N
IF (A(J) - XMIN) 11,12,12
12 IF (A(J) - XMAX) 15,15,14
11 XMIN = A(J)
GO TO 15
14 XMAX = A(J)
15 CONTINUE
GO TO 202

```

PLOT0000
 PLOT0010
 PLOT0020
 PLOT0030
 PLOT0040
 PLOT0050
 PLOT0060
 PLOT0070
 PLOT0080
 PLOT0090
 PLOT0100
 PLOT0110
 PLOT0120
 PLOT0130
 PLOT0140
 PLOT0150
 PLOT0160
 PLOT0170
 PLOT0180
 PLOT0190
 PLOT0200
 PLOT0210
 PLOT0220
 PLOT0230
 PLOT0240
 PLOT0250
 PLOT0260
 PLOT0270
 PLOT0280
 PLOT0290
 PLOT0300
 PLOT0310
 PLOT0320
 PLOT0330
 PLOT0340
 PLOT0350
 PLOT0360
 PLOT0370
 PLOT0380
 PLOT0390
 PLOT0400
 PLOT0410
 PLOT0420
 PLOT0430
 PLOT0440
 PLOT0450
 PLOT0460
 PLOT0470
 PLOT0480
 PLOT0490
 PLOT0500
 PLOT0510
 PLOT0520
 PLOT0530

```

S.0022      20 XMAX=XLAX
S.0023      XMIN=XLIN
C
S.0024      C CALCULATE RAW SCALE SIZE
          202 XSCAL=(XMAX-XMIN)/100.
C
S.0025      C ROUTINE TO CALCULATE EXACT SCALE SIZE AND END POINTS
          CALL SCAL (XSCAL,XMAX,XMIN)
C
S.0026      C IF NO EXTREMES OF Y GIVEN, FIND THEM
          IF (YLAX - YLIN) 110,112,113
          112 L = N + 1
          YMIN = A(L)
          YMAX=YMIN
          LL = MN
          DO 40 J = 1,LL
          26 IF(A(J)-YMIN) 28,26,26
          28 YMIN=A(J)
          30 YMAX=A(J)
          40 CONTINUE
          110 YMAX=YLAX
          YMIN=YLIN
C
S.0041      C GET SCALE SIZE AND END POINTS
          201 YSCAL = (YMAX - YMIN) / 50.
          CALL SCAL (YSCAL,YMAX,YMIN)
          PRINT TOP SCALE
          XPR(1) = XMIN
          DO 200 JP=1,10
          XPR(JP+1) = XPR(JP) + XSCAL * 10.
          C TO MAKE SURE THAT ZERO REALLY PRINTS AS ZERO, NOT A SMALL NUMBER
          C CAUSED BY ROUNDING ERRORS.
          240 XPR(JP+1) = 0.
          200 CONTINUE
          WRITE (3,8) (XPR(JP),JP=1,11)
          WRITE (3,7)
          C IF CALCULATED CURVE WANTED GET VALUES BETWEEN XMIN AND XMAX
          210 IF (MFUNC) 210,210,211
          211 F = XMIN
          JP = 1
          212 CALL FUNC(F,CALC(JP))
          213 IF (F - XMAX) 212,210,210
          212 F = F + XSCAL
          JP = JP + 1
          GO TO 212
          210 CONTINUE
C
S.0043      C START POINT AT MAXIMUM Y
          YDP = YMAX
          CLEAR PRINT LTAG
          230 DO 55 JP = 1,101
          55 FOUT(JP) = 0
          C IF CALCULATED CURVE WANTED SET UP PRINTS
          215 IF (MFUNC) 214,214,215
          215 F = XMIN
          C SCAN ALL VALUES OF Y FOR X BETWEEN FOUT AND XMAX

```

```

          PLOT0540
          PLOT0550
          PLOT0560
          PLOT0570
          PLOT0580
          PLOT0590
          PLOT0600
          PLOT0610
          PLOT0620
          PLOT0630
          PLOT0640
          PLOT0650
          PLOT0660
          PLOT0670
          PLOT0680
          PLOT0690
          PLOT0700
          PLOT0710
          PLOT0720
          PLOT0730
          PLOT0740
          PLOT0750
          PLOT0760
          PLOT0770
          PLOT0780
          PLOT0790
          PLOT0800
          PLOT0810
          PLOT0820
          PLOT0830
          PLOT0840
          PLOT0850
          PLOT0860
          PLOT0870
          PLOT0880
          PLOT0890
          PLOT0900
          PLOT0910
          PLOT0920
          PLOT0930
          PLOT0940
          PLOT0950
          PLOT0960
          PLOT0970
          PLOT0980
          PLOT0990
          PLOT1000
          PLOT1010
          PLOT1020
          PLOT1030
          PLOT1040
          PLOT1050
          PLOT1060
          PLOT1070
          PLOT1080

```

```

S.0065      JP = 1
C          IS POINT WITHIN HALF A SCALE OF PRINT POSITION
S.0066      220 IF (ABS(YPR-CALC(JP)) - .5 * YSCAL) 216,217,218
C          IF EXACTLY BETWEEN PRINT POSITIONS ONLY PRINT IT ONCE
S.0067      217 IF (YPR - CALC(JP)) 218,216,216
C          RELIEVE IT OR NOT THIS IS AN ASTERISK (NUMBER TOO LARGE TO WRITE
C          AS ONE NUMBER)
S.0068      216 IOUT(JP) = 92*16777216
S.0069      218 IF (F - XMAX) 219,214,214
S.0070      219 JP = JP + 1
S.0071      F = F + XSCAL
S.0072      GO TO 220
C          RUN DOWN EACH SET OF DEPENDENT VARIABLES
C          IF NO POINTS WANTED
S.0073      214 IF (N) 70,70,300
S.0074      300 DO 221 J = 2,M
S.0075      DO 222 L = 1,N
C          CALCULATE SUBSCRIPT FOR A
S.0076      LL = (J - 1) * N + L
C          IS IT WITHIN HALF A SCALE OF PRINT POSITION
S.0077      IF (ABS(YPR - A(LL)) - .5 * YSCAL) 223,224,225
C          IF EXACTLY HALFWAY BETWEEN, PRINT ONLY ONCE
S.0078      224 IF (YPR - A(LL)) 225,223,223
C          FIND HORIZONTAL POSITION
S.0079      223 JP = (A(LL) - XMIN) / XSCAL + 1.5
C          IF OFF GRAPH, FORGET IT
S.0080      IF (JP - 1) 225,226,226
S.0081      226 IF (JP - 101) 227,227,225
C          THIS GIVES 1,2,3 ETC. FOR J=2,3,4 ETC
S.0082      227 IOUT(JP) = (J - 17) * 16777216
S.0083      225 CONTINUE
S.0084      222 CONTINUE
S.0085      221 CONTINUE
S.0086      70 ICOUNT = ICOUNT + 1
C          PRINT VALUE ON VERTICAL AXIS EVERY FIVE POSITIONS
S.0087      IF (ICOUNT - 5) 120,119,120
S.0088      120 WRITE (3,116) (IOUT(JP),JP=1,101)
S.0089      GO TO 20
C          MAKE ZERO PRINT AS ZERO, NOT SMALL NUMBER
S.0090      119 IF (ABS(YPR) - .5 * YSCAL) 232,232,233
S.0091      232 F = 0.
S.0092      233 F = YPR
S.0093      234 WRITE (3,2) F, (IOUT(JP),JP=1,101),F
S.0094      ICOUNT = 0
S.0095      IF REACHED YMIN, STOP
S.0096      40 IF (YPR - YMIN) 46,46,45
C          ELSE DECREMENT Y
S.0097      45 YPR = YPR - YSCAL
S.0098      GO TO 232
S.0099      46 WRITE (3,7)
C          PRINT BOTTOM SCALE
S.0100      XPP(1) = XMIN
S.0101      DO 50 JP = 1,10
S.0102      XPP(JP+1) = XPP(JP) + XSCAL * 10.

```


BPS FORTRAN IVD COMPILER VERSION 3 LEVEL 0 DEC 1966

STATEMENT NUMBER	REL ADR	STATEMENT NUMBER	REL ADR	STATEMENT NUMBER	REL ADR	STATEMENT NUMBER	REL ADR
00014	000712	00015	00072A	00020	000744	00202	000754
00112	000780	00026	0007DE	00028	000802	00030	000820
00040	000838	00110	000852	00201	000862	00240	0008C0
00200	0008D0	00211	000944	00213	000954	00212	000984
00210	0009A2	00230	0009AA	00055	0009B2	00215	0009E2
00220	0009F2	00217	000A1A	00216	000A30	00218	000A44
00219	000A52	00214	000A70	00300	000A7C	00224	000AD0
00223	000AEE	00226	000B38	00227	000B46	00225	000B5E
00222	000B5E	00221	000B72	00070	000B86	00120	000BA0
00119	000B66	00232	000BFC	00233	000C0A	00234	000C12
00080	000C6C	00345	000C7A	00086	000C8C	00231	000CE0
00090	000CFC						

SIZE OF COMMON 00000 PROGRAM 03498

 END OF COMPILATION PLOT
 COMPILATION TIME WAS 0033.04 SECONDS

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
------	---------	------	---------	------	---------

ALOG10	0000A4				
--------	--------	--	--	--	--

CONSTANTS

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
------	---------	------	---------	------	---------

CCCCCCCC1	0000FC	41AC0C0C	0000F0	411AC000	0000F4
-----------	--------	----------	--------	----------	--------

IMPLIED EXTERNAL REFERENCES

NAME	REL ADR	NAME	REL ADR	NAME	REL ADR
------	---------	------	---------	------	---------

TECOM	000104	FRXPT	000120		
-------	--------	-------	--------	--	--

STATEMENT NUMBER	REL ADR	STATEMENT NUMBER	REL ADR	STATEMENT NUMBER	REL ADR
------------------	---------	------------------	---------	------------------	---------

00001	000172	00002	000186	00020	000104
00003	0001E4	00005	000208	00007	000228
00004	000254	00012	000280	00010	000242
00013	0002FC	00015	000322	00014	000234

SIZE OF COMMON 00000 PROGRAM 0003C

END OF COMPILATION
SCAL
COMPIATION TIME WAS 0014.06 SECONDS

BPS FORTRAN IVD COMPILER VERSION 3 LEVEL 0 DEC 1966

/DATA

BEGIN LOADING

STORAGE MAP
 IBCOM AT LOCATION 0C11C0
 *TEST AT LOCATION 0C278C
 TEST AT LOCATION 0C278E
 *SINUS AT LOCATION 0C2998
 SINUS AT LOCATION 0C29AC
 *PLOT AT LOCATION 0C2AB0
 PLOT AT LOCATION 0C2AB8
 *SCAL AT LOCATION 0C386C
 SCAL AT LOCATION 0C386E
 *FRXPI AT LOCATION 0C3BDC
 FRXPI AT LOCATION 0C3BDE
 *EXIT AT LOCATION 0C3C6C
 EXIT AT LOCATION 0C3C6E
 *ALOG AT LOCATION 0C3C7E
 ALOG AT LOCATION 0C3C80
 *ALOG10 AT LOCATION 0C3C8C
 ALOG10 AT LOCATION 0C3C8E
 *COS AT LOCATION 0C3D8C
 COS AT LOCATION 0C3D8E
 *SIN AT LOCATION 0C3D9C
 SIN AT LOCATION 0C3D9E

CHART 1

