

THIS program produces a graph of a function (expressible in the form $y=f(x)$ using standard BASIC notation) for a range of x values. It is self-scaling once it has been given the range of x , finding the lowest and highest y values. It also draws the lines $x=0$, and $y=0$, when they fall within the x or y values.

METHOD

On the UK101, reasonable resolution can be obtained using the graphics characters. The screen is 46 by 16 chars, but the graphics include 8 horizontal bars.

These bars can increase the resolution to 46 by 128 allowing a reasonable curve to be drawn.

The first problem encountered was how to change the function of x every time the program is run. The simplest answer is this:

```
100 PRINT " Please type : "
110 PRINT " 5000 Y=f(X) "
120 PRINT " GOTO 1234 "
130 STOP
1234 program
```

This is not ideal. The best answer is to make the program alter itself. Microsoft BASIC is memory efficient for the reason that the commands are abbreviated by the use of tokens.

What does a Basic line look like in memory? Consider this line:

10 Y=X

If the relevant section of memory is examined, the line is stored as follows:

14 3 10 0 89 171 88 0

The "14 3" in the first and second byte means the next BASIC line is stored at memory location $14 + 3 \star 256$ (=782 decimal). The "10 0" in the next two bytes indicates that this is BASIC line number $10 + 0 \star 256$ (=10 decimal). 89 is the ASCII code for Y, and 88 for X. So somehow 171 means "=", and 0 means the end of the line.

So far:

```
14 3 10 0 89 171 88 0
 782 line 10 Y = X END of line
```

A full list of tokens is given in Table 2 only those underlined are useful for the function of x .

TABLE 2: Tokens. Those underlined are used

128	END	151	PRINT	174	<u>INT</u>
129	FOR	152	CONT	175	<u>ABS</u>
130	NEXT	153	LIST	176	<u>USR</u>
131	DATA	154	CLEAR	177	<u>FRE</u>
132	INPUT	155	NEW	178	<u>POS</u>
133	DIM	156	TAB(179	<u>SQR</u>
134	READ	157	TO	180	<u>RND</u>
135	LET	158	FN	181	<u>LOG</u>
136	GOTO	159	SPC(182	<u>EXP</u>
137	RUN	160	THEN	183	<u>COS</u>
138	IF	161	NOT	184	<u>SIN</u>
139	RESTORE	162	STEP	185	<u>TAN</u>
140	GOSUB	163	<u>+</u>	186	<u>ATN</u>
141	RETURN	164	<u>-</u>	187	<u>PEEK</u>
142	REM	165	<u>*</u>	188	<u>LEN</u>
143	STOP	166	<u>/</u>	189	<u>STR\$</u>
144	ON	167	<u>↑</u>	190	<u>VAL</u>
145	NULL	168	<u>AND</u>	191	<u>ASC</u>
146	WAIT	169	<u>OR</u>	192	<u>CHR\$</u>
147	LOAD	170	<u>=</u>	193	<u>LEFT\$</u>
148	SAVE	171	<u>></u>	194	<u>RIGHT\$</u>
149	DEF	172	<u><</u>	195	<u>MID\$</u>
150	POKE	173	<u>SGN</u>	197 to 211	BASIC error codes

Thus if we input the function of x , we can find the suitable line in memory and poke into it the function. The line which we look for is line 5000:

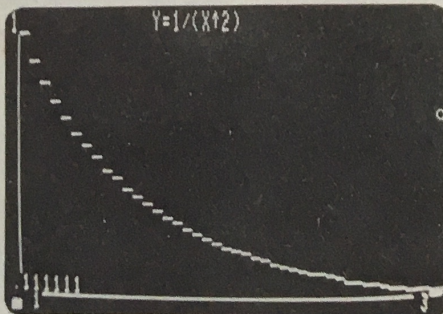
5000 # # # # # # # # # # etc.

The # symbols are looked for in memory. When they are found the function $y=f(x)$ is poked in (the function *must* be in standard BASIC notation). The most convenient way to end a line is to make it a multi-line statement, so the colon and REM are both POKED in at the end of the line. We now have the function of our graph in the form:

5000 $y=f(x)$: REM # # # # # etc.
5010 RETURN

Values of the range of x are then inputted, and the highest and lowest values of y are found for scaling purposes. The graph can then be plotted using the graphic characters.

SOFTWARE IDEAL FOR MATHEMATICS DEMONSTRATION.

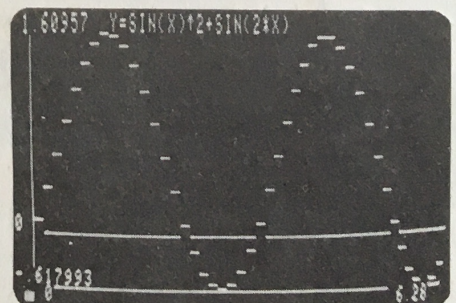


Left: Graph of $y = 1/x^2$. Note that the function must be keyed into the computer as if a BASIC statement in itself.

Right: A more complex, cyclic function. The Graph Plotter is self-scaling.

Some suggested graphs

- 1) $Y=ABS(X)$ for $x=-1$ to 1
- 2) $Y=-1 \star ABS(X)$ for $X=-1$ to 1
- 3) $Y=SIN(X) \uparrow 2 + SIN(2 \star X)$ for $X=0$ to $2 \star Pi$
- 4) $Y=1 \div (X \uparrow 2)$ for $x = 1$ to 3



OK
LIST

```

10 REM *****
20 REM **** GRAPH PLOT ****
30 REM **** ===== ****
40 REM ****
50 REM **** by T.Walsh ****
60 REM **** for a 8K ****
70 REM **** Compukit UK101****
80 REM *****
100 FORA=1TO16:PRINT:NEXT
110 PRINT" GRAPH PLOT"
120 PRINT" ====="
130 PRINT" by T.Walsh"
140 PRINT:PRINT
150 INPUT" Type graph in form Y=f(X)";A$
160 IFLen(A$)>37THEN150
170 FORA=2800TO3100
180 IFPEEK(A)=35ANDPEEK(A+1)=35THEN210
190 NEXT
200 PRINT" Program error":STOP
210 Q=LEN(A$):LP=A
220 A$=RIGHT$(A$,Q-2)
230 W=0
240 POKELP,89:POKELP+1,171
250 FORA=1TOLEN(A$)
260 W=W+1
270 D$=MID$(A$,A,1)
280 IFASC(D$)>47ANDASC(D$)<58THEN500
290 IFD$="X"ORD$="("THEN500
300 IFD$=")"THEN500
310 IFD$=" "ORD$="."THEN500
320 IFD$="+ "THENZ=163:GOTO510
330 IFD$="- "THENZ=164:GOTO510
340 IFD$="*"THENZ=165:GOTO510
350 IFD$="/"THENZ=166:GOTO510
360 IFD$="↑"THENZ=167:GOTO510
370 D$=MID$(A$,A,3)
380 IFD$="SGN"THENZ=173:GOTO520
390 IFD$="INT"THENZ=174:GOTO520
400 IFD$="ABS"THENZ=175:GOTO520
410 IFD$="SQP"THENZ=179:GOTO520
420 IFD$="RND"THENZ=180:GOTO520
430 IFD$="LOG"THENZ=181:GOTO520
440 IFD$="EXP"THENZ=182:GOTO520
450 IFD$="COS"THENZ=183:GOTO520
460 IFD$="SIN"THENZ=184:GOTO520
470 IFD$="TAN"THENZ=185:GOTO520
480 IFD$="ATN"THENZ=186:GOTO520
490 PRINT"Error in function":GOTO880
500 POKELP+W+1,ASC(D$):GOTO530
510 POKELP+W+1,Z:GOTO530
520 POKELP+W+1,Z:A=A+2
530 NEXT:POKELP+W+2,58
540 POKELP+W+3,142
550 F$="Y="+A$
560 PRINT:PRINT" Enter the range of x (low";
570 INPUT" then high)";LO,HI:PRINT:PRINT
580 IFLO>HITHEN560
590 X=LO:GOSUB5000
600 YH=Y:YL=Y
610 FORX=LOTOHISTEP(HI-LO)/46
620 GOSUB5000
630 IFY>YHTHENYH=Y
640 IFY<YLTHENYL=Y
650 NEXT
660 FORA=1TO16:PRINT:NEXT
670 FORA=1TO15
680 POKF53261+64*(A-1),143
690 NEXT
700 FORA=54222TO54285
710 POKEA,135:NEXT
720 S=(HI-LO)/46
730 IFHI>OANDLO<OTHEN2000
740 IFYH>OANDYL<OTHEN3000
750 FORB=1TO46:X=(B-1)*S+LO:GOSUB5000
760 D=((Y-YL)/(YH-YL))*14+1
770 X=B:Y=D+1:GOSUB1000:NEXT
780 A$=F$
790 FORA=1TOLEN(A$)
800 POKF53279+A-LEN(A$)/2,ASC(MID$(A$,A,1))
810 NEXT
820 A=54221:A$=STR$(LO):GOSUB6000
830 A$=STR$(YH):A=53259:GOSUB6000
840 A$=STR$(YL):A=54155:GOSUB6000

```

```

850 A$=STR$(HI):A=54264-LEN(A$):GOSUB6000
860 POKF530,1:POKE57088,0
870 IFPEEK(57098)=254THEN860
880 POKF530,0:FORA=1TO39:POKELP+A-1,35:NEXT
890 RUN
1000 Z=54285:S1=INT((Y-INT(Y))*7+.5)
1010 POKF7-INT(Y)*64+X,128+S1:RETURN
2000 A=53262+15*64+(ABS(LO)*46/(HI-LO))
2010 POKEA,48
2020 FORB=OTO15
2030 POKEA-(B+1)*64,143
2040 NEXT:GOTO740
3000 A=54220
3010 A=A-INT(ABS(YL)*15/(YH-YL)+.5)*64
3020 POKEA,48
3030 A=A+2
3040 FORB=OTO45
3050 IFPEEK(A+B)=143THENPOKEA+B,208:GOTO3070
3060 POKEA+B,128
3070 NEXT:GOTO750
5000 #####
5010 RETURN
6000 FORB=1TOLEN(A$)
6010 POKEA+B,ASC(MID$(A$,B,1))
6020 NEXT:RETURN
OK

```

LINES

- 100 to 160 Inputs function
- 170 to 540 Finds line 5000 in memory and pokes the function of x in
- 550 to 650 Inputs the low and high values of x and from these calculates the maximum and minimum y values
- 570 to 850 Plots graph
- 860 to 890 Waits for any key to be pressed and then returns line 5000 to its original format
- 1000 to 1050 Plots the horizontal bar
- 2000 to 2040 Plots the line X=0 [if applicable]
- 3000 to 3070 Plots the line Y=0 [if applicable]
- 5000 to 5010 Function of graph in form Y=f(X)
- 6000 to 6010 Pokes the contents of A\$ into a part of the screen pointed to by the contents of A

IMPORTANT VARIABLES

- HI highest value of X
- LO lowest value of X
- LP position of line 5000 in memory
- YH highest value of Y
- YL lowest value of Y

NOTES

- 1) line 5000 should be retyped if the program is stopped halfway through
- 2) the function of X must be in standard Basic notation. Any error will mean that you will have to retype line 5000, and re-run the program.
- 3) all angles are in radians.
- 4) do not try to plot infinity [eg tan of a half of pi]