Calculator Users' Club

Introductory Package

HEWLETT hp PACKARD

INTRODUCTION

This introductory program package is an extract from the Calculator Users' Club (CUC) program library — which now has more than 1000 programs.

The CUC was originally formed for 9820 calculator users, to give them the opportunity to exchange programs they had written. With the introduction of the 9821 calculator and more recently the 9825 calculator, the CUC has expanded to include them.

Obviously, because the 9825 is a newcomer to the CUC, there are presently fewer programs for it. However, the "9820 family" of calculators use a similar programming language and with minor modifications the programs written for the 9820 and 9821 can be adapted to the 9825. In fact, the extra programming power of HPL will often allow you to further optimise the programs.

MODEL 9825A

TABLE OF CONTENTS

hp HEWLETT · PACKARD
hp HEWLETT · PACKARD
hp HEWLETT · PACKARD
hp HEWLETT · PACKARD

MODEL 9825A

THREE DIMENSIONAL HIDDEN LINE PLOT

This program is a modification of the CUC program log no. 0020 with the title: "Plot of Z = *f* (X, Y)".

This program differs from the above mentioned in the following respects:

1 The sine and cosine of tilt and rotation angle are only calculated once, thereby increasing the speed considerable.

2 The variables X and Y are used to calculate *f* (X, Y) and the result is stored in Z, making the programming of the function somewhat easier.

3 The program uses 202 r-Variables to store the minimum and maximum values of the plotted Y, and is therefore able to skip the plotting of "hidden lines" (see examples).

Plot equations used:

$$X' = (X - \tfrac{1}{2}\,(Xmax + Xmin))\,/\,(Xmax - Xmin)$$

$$Y' = (Y - \tfrac{1}{2}\,(Ymax + ymin))\,/\,(Ymax - Ymin)$$

$$Z' = (Z - \tfrac{1}{2}\,(Zmax + Zmin))\,/\,(Zmax - Zmin)$$

$$X'' = X' \cos(rot) - Y' \sin(rot)$$

$$Y'' = Z' \cos(tilt) - (X' \sin(rot) + Y' \cos(rot)) \sin(tilt)$$

$$Xplot = .7\,(scale)\,X''$$

$$Yplot = (scale)\,Y''$$

Scale command: SCL – .5, .5, – .5, .5

The subroutine to calculate Z can be found under lable "1".

HARDWARE REQUIREMENTS: 9825A (standard 6,844 bytes)

MEMORY: 1508 bytes

ROMS: 98212A Gen I/O-9862 Plotter Rom or 98214A Gen I/O-Ext I/O-9862 Plotter Rom.

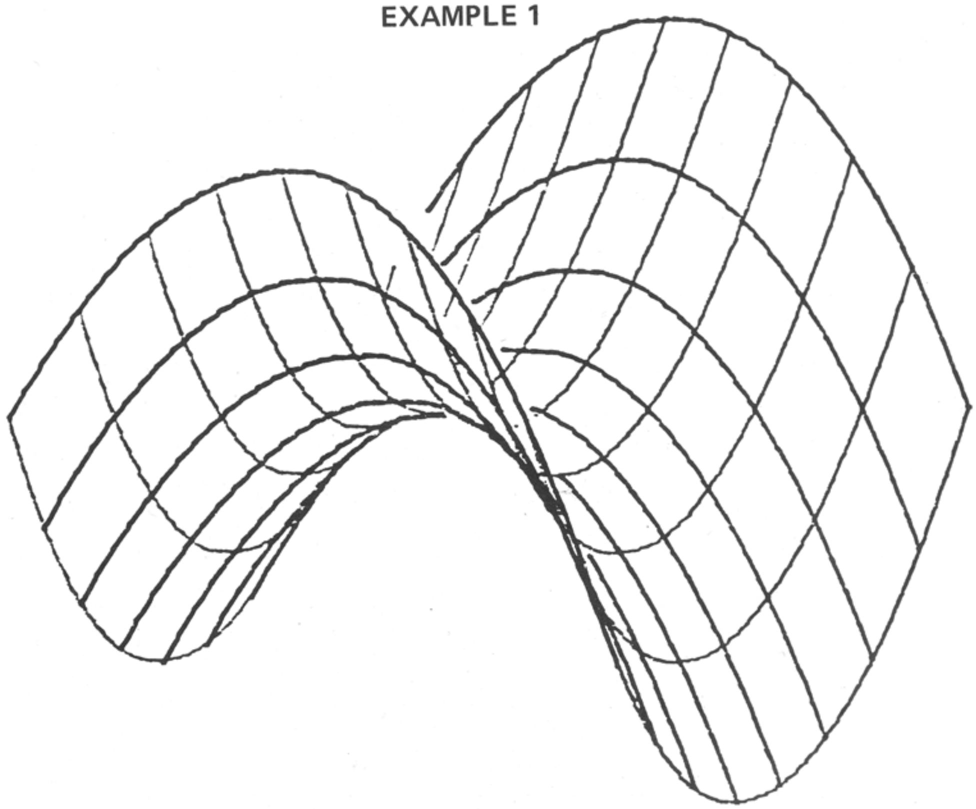
PERIPHERALS: 9862A Plotter.

STEP	DISPLAY	INSTRUCTIONS
1		ldf N, Load the file on which the program is stored.
2		fetch line 46, EXECUTE
		This brings the first line of the subroutine "1" in the display, to calculate f (X, Y) = Z
		Enter the correct subroutine to calculate Z from X and Y. Variable C may be used as
		parameter or temporary store. The last line of the subroutine should be terminated
		with : ret (return) STORE
3		Run
4	X MAX =	Enter X Max, CONTINUE
5	X MIN =	Enter X Min, CONTINUE
6	Y MAX =	Enter Y Max, CONTINUE
7	Y MIN =	Enter Y Min, CONTINUE
8	Z MAX =	Enter Z Max, CONTINUE
9	Z MIN	Enter Z Min, CONTINUE
10	TILT ANGLE =	Enter Tilt Angle, CONTINUE
11	ROTATION ANGLE =	Enter Rotation Angle, CONTINUE
12	X LINES PLOTTED	Enter X lines plotted, CONTINUE
13	Y LINES PLOTTED	Enter Y lines plotted, CONTINUE
		The number given under X lines plotted and Y lines plotted actually stands for the
		number of X and Y increments. The number of lines plotted will be one more than
		specified.
		In order to achieve a high degree of resolution, it is advisable to specify at least 25
		Y-lines. With a high number of Y lines the plotting of X lines may be omitted. (See note)
14	PLOTS/X LINE:	Enter Plots X/line, CONTINUE
15	PLOTS/Y LINE:	Enter Plots Y/line, CONTINUE
		The number given under PLOTS/LINE should not be larger than 100.
16	SCALE:	For scale specify anything between 0 and 1, CONTINUE
17		The Function is now plotted. Y-lines first.
	Note:	If only Y-lines are required; sfg EXECUTE may be pressed while the calculator is
		plotting. The program will then stop after the Y-lines are completed. X-lines can be
		given a different color by changing pens at this point. The program will continue
		X-lines after CONTINUE
		Is sgf, EXECUTE is not pressed while the calculator is plotting, then the stop after
		plotting of the Y-lines will be omitted and the program will produce the specified
		number of X-lines.

```
X MAX=
2.000
X MIN=
-2.000
Y MAX
2.000
Y MIN=
-2.000
Z MAX=
4.000
Z MIN=
-4.000
TILT ANGLE=
18.000
ROTATION ANGLE
33.000
X LINES PLOTTED
9.000
Y LINES PLOTTED
9.000
PLOTS/XLINE
39.000
PLOTS/YLINE
39.000
SCALE
.500
```

EXAMPLE 1

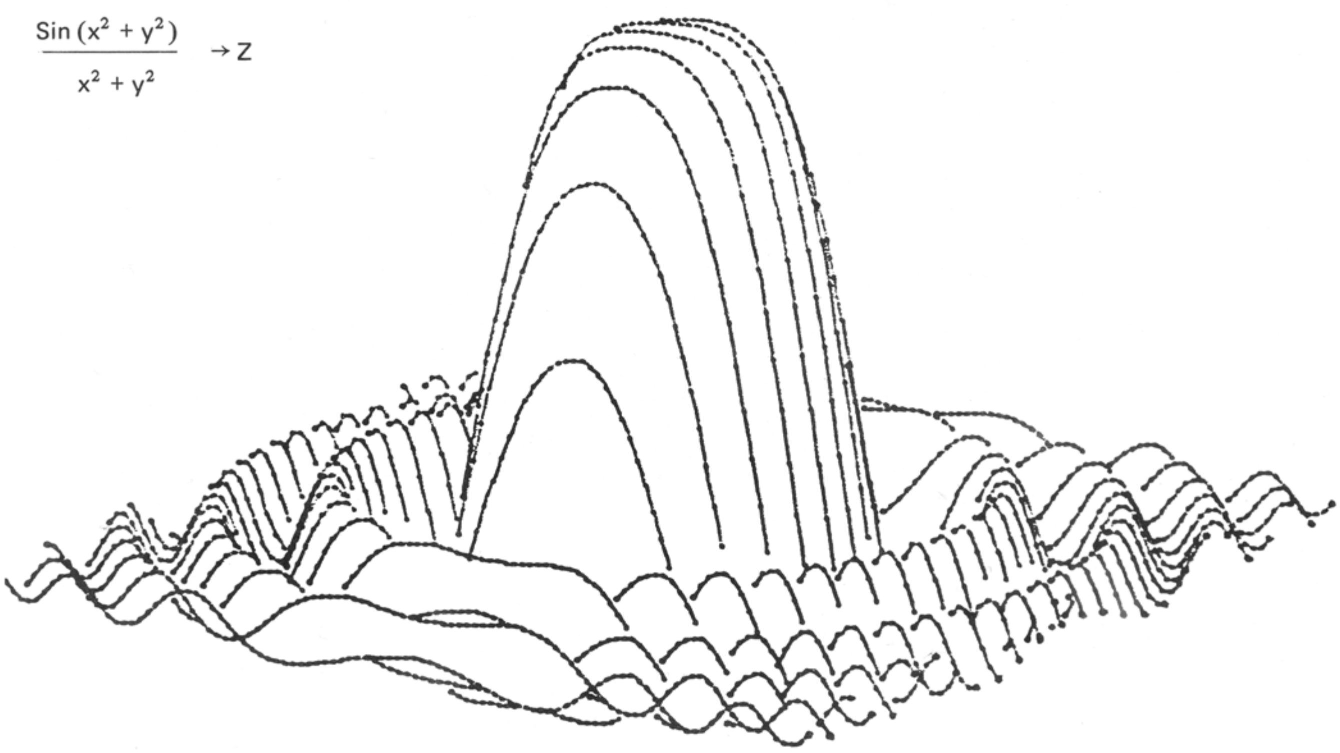
YY−XX → Z



```
X MAX=
30.000
X MIN=
-30.000
Y MAX
30.000
Y MIN=
-30.000
Z MAX=
1.000
Z MIN=
-1.000
TILT ANGLE=
15.000
ROTATION ANGLE
33.000
X LINES PLOTTED
25.000
Y LINES PLOTTED
25.000
PLOTS/XLINE
100.000
PLOTS/YLINE
100.000
SCALE
.8
```

EXAMPLE 2

$$\frac{\sin(x^2 + y^2)}{x^2 + y^2} \rightarrow Z$$



PROGRAM LISTING

```
0: scl -.5,.5,-.5,.5;enp "X MAX=",r10
1: enp "X MIN=",r9
2: enp "Y MAX",r12
3: enp "Y MIN=",r11
4: enp "Z MAX=",r14
5: enp "Z MIN=",r13
6: enp "TILT ANGLE=",r0;cos(r0)→r0;sin(acs(r0))→r1
7: enp "ROTATION ANGLE",r2;cos(r2)→r2;sin(acs(r2))→r3
8: enp "X LINES PLOTTED",r4,"Y LINES PLOTTED",r5,"PLOTS/XLINE",r6
9: enp "PLOTS/YLINE",r7,"SCALE ",r8;spc 3;r8→Z
10: 223→A
11: -1→rint(A);jmp (A-1→A)=121
12: 1→rA;jmp (A-1→A)=19
13: (r10-r9)/r6→r15;(r12-r11)/r5→r16;r9→X;r12→Y;20.5→A
14: gsb "1"
15: gsb "2"
16: X+r15→X;if X<=r10;gto -2
17: A-int(A)+r16r3/r15r2+20→A;20→B
18: rint(A)→rB;r(int(A)+101)→r(int(B)+101)
19: A+1→A;B+1→B;if 122<=A;A-1→A
20: if B#r6+21;gto -2
21: A-int(A)+20→A;pen
22: r9→X;Y-r16→Y;if r11<=Y;gto -8
23: 223→A;if flg0;cfg 0;stp
24: -1→rint(A);jmp (A-1→A)=121
25: 1→rint(A);jmp (A-1→A)=19
26: (r10-r9)/r4→r15;(r12-r11)/r7→r16;r10→X;r11→Y;20.5→A
27: gsb "1"
28: gsb "2"
29: Y+r16→Y;if Y<=r12;gto -2
30: A-int(A)+r15r2/r16r3+20→A;20→B
31: rint(A)→rint(B);r(int(A)+101)→r(int(101)+B)
32: A+1→A;B+1→B;if 122<=A;A-1→A
33: if B#r6+21;gto -2
34: A-int(A)+20→A;pen
35: r11→Y;X-r15→X;if r9<=X;gto -8
36: ltr -.4,-.45,431;stp
37: "2":(X-(r10+r9)/2)/(r10-r9)→r19
38: (Y-(r11+r12)/2)/(r12-r11)→r18
39: (Z-(r14+r13)/2)/(r14-r13)→Z
40: .7r8(r19r2-r18r3)→r17
41: r8(Zr0-(r19r3+r18r2)r1)→r18
42: if rint(A)>rint(18);r18→rint(A);sfg 1
43: if r(int(A)+101)<=r18;r18→r(int(A)+101);sfg 1
44: if flg1;cfg 1;plt r17,r18;A+1→A;ret
45: pen;A+1→A;ret
46: "1":XX+YY→C;if C=0;1→Z;ret
47: 180sin(C)/πC→Z;ret
48: end
*2499
```

MEMORY ALLOCATION

VARIABLES							
A	R-Counter	r122					
B	R-Counter	•					
C	Temporary	•					
X	X Value	•					
Y	Y Value	•					
Z	Computed Z	•					
		•					
r0	cos tilt	•					
r1	sin tilt	•					
r2	cos rot.	•					
r3	sin rot.	•					
r4	nr. X lines	•					
r5	nr. Y lines	r222					
r6	plots/Xline	r223	= – 1				
r7	plots/Yline	r224	and up:				
r8	scale		not used				
r9	Xmin						
r10	Xmax						
r11	Ymin						
r12	Ymax						
r13	Zmin						
r14	Zmax						
r15	Delta X						
r16	Delta Y						
r17	X Plot						
r18	Y Plot						
r19	Temporary						
r20							
•							
•							
•							
•							
•	minimum values					FLAGS	
•						0	
•						1	
•						2	
•						3	
•						4	
•						5	
•						6	
•						7	
r121	= 1					8	
						9	
						10	
						11	
						12	
						13	
						14	
						15	

MODEL 9825A

NUMERICAL INVERSION OF LAPLACE TRANSFORM

This program computes the numerical inversion of Laplace transform. For known F(p) it finds f(t) $= L^{-1}$ F(p) at points $t_j = t_{min} + j \triangle t$, for j=0, 1, 2, ... and $t_j \leqslant t_{max}$ for all j. A series expansion of n terms is used, $n \leqslant 20$.

The recommended value of n is 10.

HARDWARE REQUIREMENTS: 9825A (standard 6,844 bytes)

MEMORY: 562 bytes.

ROMS:

PERIPHERALS:

STEP	DISPLAY	INSTRUCTIONS
1		idf N (load the file on which the program is stored.
2		fxd N EXECUTE or flt N EXECUTE as desired.
3		fetch 16, EXECUTE and key in and store F(p) in r21, taking P from C.
		F(p) → r21; gto 13 STORE end STORE.
4		RUN
5	N =	Input N, CONTINUE
6	Tmin	" Tmin, CONTINUE
7	Tmax	" Tmax, CONTINUE
8	DT	" △t, CONTINUE
		Note: In step 3 variables r53 to r172 are free to store F(p).

EXAMPLE

```
N=
         10.000000
T=        2.000000
          8.000000
          0.500000


T
F(T)

          2.000000
          0.398940

          2.500000
          0.356826

          3.000000
          0.325733

          3.500000
          0.301574

          4.000000
          0.282095

          4.500000
          0.265961

          5.000000
          0.252316

          5.500000
          0.240570

          6.000000
          0.230330

          6.500000
          0.221294

          7.000000
          0.213244

          7.500000
          0.206013

          8.000000
          0.199471
```

PROGRAM LISTING

```
0: prt "N=";ent "N=",A;prt A;1→B→r0
1: Br(B-1)→rB;jmp (B+1→B)>A
2: .5A→C;2/r(C-1)→rB;A+C→X
3: B+1→B;B-A→Y;Y↑C*r(2Y)/r(C-Y)rYr(Y-1)→rB;jmp X<=B
4: (-1)↑(C+1)→r52
5: B+1→B;0→rB;B-X→Y→Z;int(.5(Y+1))→C;if 2Y>A;.5A→Z
6: rB+r(A+C)/r(Y-C)/r(2C-Y)→rB;jmp (C+1→C)>Z
7: rBr52→rB;-r52→r52;jmp -3(2A+.5A>B)+1
8: 1.5A→C;1→B
9: r(C+B)→rB;jmp (B+1→B)>A
10: ent "T MIN",X,"T MAX",Y,"DT",Z;prt "T=",X,Y,Z;spc 2;prt "T","F(T)";spc
11: 1→B;0→r0
12: Bln(2)/X→C;jmp 4
13: rBr21+r0→r0;jmp 1-2((B+1→B)<=A)
14: r0ln(2)/X→r0;prt X,r0;spc ;jmp 1-4((X+Z→X)<=Y)
15: stp
16: 1/ΓC→r21;gto 13
17: end
*2607
```

15 REGRESSION CURVES

The program makes least squares regression to 14 biparametric and one triparametric curves, prints their parameters and plots the curves. The curves are:

$y = a + bx$	$y = a + b/x^2$	$y = ax^b$
$y = 1/(a + bx)$	$y = 1/(a + b/x^2$	$y = a\,e^{bx}$
$y = a + bx^2$	$y = a + b\sqrt{x}$	$y = a\,e^{bx^2}$
$y = a + b/x$	$y = a + b/\sqrt{x}$	$y = a + b \ln x$
$y = 1/(a + b/x)$	$y = 1/(a + b/\sqrt{x})$	$y = a + bx + cx^2$

Input medium is the Calculator keyboard.

The program prints, resp. plots input data, calculates the coefficients a,b, resp. c and plots a curve chosen. The curve is chosen by its number (see the table below).

If an illegal operation was made or overflow appeared while commulating the coefficients (e.g. division by zero, square root of a negative number, etc.), there may be an error in some results or they may have no sense. In such a case the word OVERFLOW is printed and it is necessary to decide which of the regression curves, if any, have sense.

If the determinant of the system of equations (coefficients a, b, c are its solution) is zero, the system has no solution and a message DET = 0 is printed.

The program calculates R^2, squared coefficient of correlation.

Curve No.	**equation**	**interval x**	**interval y**
1	$y = a + bx$	$(-\infty + \infty)$	$(-\infty + \infty)$
2	$y = 1/(a + bx)$	$(-\infty + \infty)$	
3	$y = a + bx^2$	$(-\infty + \infty)$	$(-\infty + \infty)$
4	$y = a + b/x$	$\neq 0$	$(-\infty + \infty)$
5	$y = 1/(a + b/x)$	$\neq 0$	$\neq 0$
6	$y = a + b/x^2$	$\neq 0$	$(-\infty + \infty)$
7	$y = 1/(a + b/x^2)$	$\neq 0$	$\neq 0$
8	$y = a + b\sqrt{x}$	$\geqslant 0$	$(-\infty + \infty)$
9	$y = a + b/\sqrt{x}$	$\geqslant 0$	$(-\infty + \infty)$
10	$y = 1/(a + b/\sqrt{x})$	> 0	$\neq 0$
11	$y = ax^b$	> 0	> 0
12	$y = a\,e^{bx}$	$(-\infty + \infty)$	> 0
13	$y = a\,e^{bx^2}$	$(-\infty + \infty)$	> 0
14	$y = a + b \ln x$	> 0	$(-\infty + \infty)$
15	$y = a + bx + cx^2$	$(-\infty + \infty)$	$(-\infty + \infty)$

HARDWARE REQUIREMENTS: 9825A (standard 6,844 bytes)

MEMORY: 3480 bytes

ROMS: 98212A Gen I/O-9862 Rom (optional) or 98214A Gen I/O-Ext. I/O-9862 Plotter Rom (Optional)

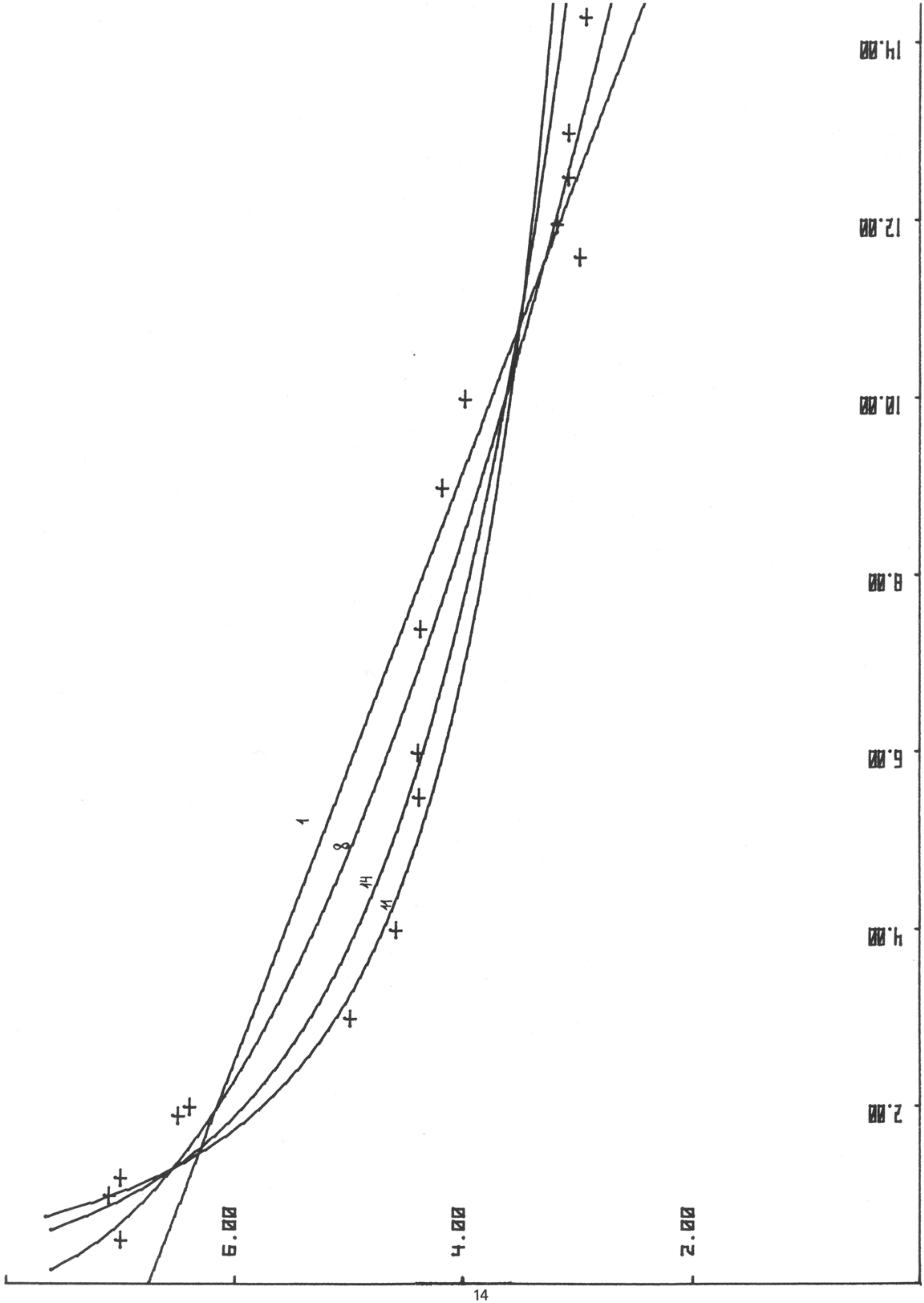
PERIPHERALS: 9862A Plotter (Optional)

STEP	DISPLAY	INSTRUCTIONS
1		ldf N; Load the file on which the program is stored.
2		fxd n or flt n (optional) EXECUTE
3		RUN
4	GRAPH	CONTINUE for Plot.
		0,CONTINUE if Plot is not desired.
5	X MIN	X Min, CONTINUE
6	X MAX	X Max, CONTINUE
7	Y MIN	Y Min, CONTINUE
8	Y MAX	Y Max, CONTINUE
9	FXD/FLT	f CONTINUE
		CONTINUE: Program sets f = 2
		Note: f specifies format of graph axes subscripts: $f > 0$ fxd f
		$f \leq 0$ flt (–f)
10	X	Enter X, CONTINUE
		CONTINUE if all data were entered.
11	Y	Enter Y, CONTINUE
12	CURVE NO.	Number of curve, CONTINUE (see table)
13	AXES?	CONTINUE for Axes Plot.
		0,CONTINUE if Axes are not desired.
14	X TIC	Distance of Tic marks at X–Axes, CONTINUE
15	Y TIC	Distance of Tic marks at Y–Axes, CONTINUE

EXAMPLE

```
  X
  Y
                 0.500
                 7.000

                 1.000
                 7.100

                 1.200
                 7.000

                 1.900
                 6.500

                 2.000
                 6.400

                 3.000
                 5.000

                 4.000
                 4.600

                 5.500
                 4.400

                 6.000
                 4.410

                 7.400
                 4.390

                 9.000
                 4.200

                10.000
                 4.000

                11.600
                 3.000

                11.970
                 3.200

                12.500
                 3.100

                13.000
                 3.100

                14.300
                 2.950
```

```
============

CURVE NO.       1
Y=A+B*X


A
 6.744102717e 00
B
-2.985962061e-01
 R SQUARE
 8.911127758e-01


============

CURVE NO.       8
Y=A+B*√X


A
 8.162760420e 00
B
-1.432041693e 00
 R SQUARE
 9.441761628e-01


============

CURVE NO.      11
Y=A*X↑B


A
 7.038346217e 00
B
-2.918398603e-01
 R SQUARE
 9.037221275e-01


============

CURVE NO.      14
Y=A+B*ln(X)


A
 6.887760682e 00
B
-1.413628289e 00
 R SQUARE
 9.346465586e-01
```


14.00
12.00
10.00
8.00
6.00
4.00
2.00
6.00
4.00
2.00
1
8
14
11

PROGRAM LISTING

```
0: "0":sfg 14
1: cfg 13;ent "GRAPH?",X;prt "X","Y";spc ;if flg13;gto "D"
2: "G":cfg 13;ent X
3: if flg13;Z→r0;gto "B";if flg15;prt "","OVERFLOW";spc 2
4: ent Y
5: prt X,Y;spc ;Z+1→Z;r9+X→r9;r10+XX→r10;r11+(XXX→C)→r11
6: r12+(CX→C)→r12;r13+1/X→r13;r14+1/XX→r14;r15+1/C→r15
7: r16+(rX→A)→r16;R17+1/A→r17;r18+Y→r18;r19+YY→r19
8: r20+1/Y→r20;r21+1/YY→r21;r22+(XY→C)→r22;r23+1/C→r23
9: r24+(CX→C)→r24;r25+1/C→r25;r26+(X/Y→C)→r26;r27+1/C→r27
10: r28+Y/XX→r28;r29+AY→r29;r30+1/AY→r30;r31+Y/A→r31
11: r32+(ln(X)→A)→r32;r33+AA→r33;r34+AY→r34;r35+(ln(Y)→B)→r35
12: r36+BB→r36;r37+BX→r37;r38+BXX→r38;r39+AB→r39
13: gto "G";if flg1;ltr X-.0064(r2-r1),Y-.0064(r4-r3),221;lbl "+"
14: "D":sfg 1;ent "X MIN ",r1,"X MAX",r2,"Y MIN",r3,"Y MAX",r4
15: 2→r40;ent "FXD/FLT",r40;scl r1,r2,r3,r4;gto "G"
16: "B":cfg 13;cfg 15;ent "CURVE NO.",C;if flg13;gto "0"
17: jmp C
18: r9→A;r10→B;r18→X;r19→Y;r22→r7;gto "A"
19: r9→A;r10→B;r20→X;r21→Y;r26→r7;gto "A"
20: r10→A;r12→B;r18→X;r19→Y;r24→r7;gto "A"
21: r13→A;r14→B;r18→X;r19→Y;r27→r7;gto "A"
22: r13→A;r14→B;r20→X;r21→Y;r23→r7;gto "A"
23: r14→A;r15→B;r18→X;r19→Y;r28→r7;gto "A"
24: r14→A;r15→B;r20→X;r21→Y;r25→r7;gto "A"
25: r16→A;r9→B;r18→X;r19→Y;r29→r7;gto "A"
26: r17→A;r13→B;r18→X;r19→Y;r31→r7;gto "A"
27: r17→A;r13→B;r20→X;r21→Y;r30→r7;gto "A"
28: r32→A;r33→B;r35→X;r36→Y;r39→r7;gto "A"
29: r9→A;r10→B;r35→X;r36→Y;r37→r7;gto "A"
30: r10→A;r12→B;r35→X;r36→Y;r38→r7;gto "A"
31: r32→A;r33→B;r18→X;r19→Y;r34→r7;gto "A"
32: r0(r10r12-r11r11)+2r9r10r11-r10r10r10→r8
33: if (r8-r9r9r12→r8)=0;sfg 15;gto "C"
34: r18(r10r12-r11r11)+r22(r10r11-r9r12)→B
35: (B+r24(r9r11-r10r10))/r8→B;r0(r12r22-r11r24)→A
36: (A-r9r12r18+r10(r9r24+r11r18-r10r22))/r8→A
37: r0(r10r24-r11r22)+r9(r10r22+r11r18-r9r24)→r7
38: (r7-r10r10r18)/r8→r7;A(r0r22-r9r18)→r8
39: (r8+r7(r0r24-r10r18))/(r0r19-r18r18)→r8;gto "C"
40: "A":(BX-Ar7)/(Br0-AA→r8)→B;if flg15;gto "C"
41: (r0r7-AX)/r8→A;if (C=11)+(C=12)+(C=13);exp(B)→B
42: AAr8/(Yr0-XX)→r8
43: "C":spc 2;fxd 0;prt "=============","","CURVE NO.",C;gsb "S"
44: if flg15;prt "DET = 0";gto "B"
45: spc 2;flt 9;prt "A",B,"B",A;if C=15;prt "C",r7
46: prt " R SQUARE",r8;if 1-flg1;gto "B"
47: (r2-(r1→X))/100→Z
48: "F":gsb "T"
49: plt X,Y;jmp -1+2((X+Z→X)>r2)
50: pen;cfg 13;ent "AXES?",X;if 1-flg13;gto "B"
51: ent "X TIC",r5,"Y TIC",r6;r1→X;if (r1<=0)(0<=r2);0→X
52: r3→Y;if (r3<=0)(0<=r4);0→Y
53: axe X,Y,r5,r6;X→A;Y+(r4-r3)/80→B;fxd abs(r40);jmp 1+(r40>0)
54: flt -r39
55: if A>r1;A-r5→A;jmp 0
```

```
56: if (A+r5→A)>r2;jmp 2
57: ltr A,B,212;lbl A;jmp -1
58: X+(r2-r1)/80→A;Y→B
59: if B>r3;B-r6→B;jmp 0
60: if r4<=(B+r6→B);gto "B"
61: ltr A,B,211;lbl B;jmp -1
62: "S":jmp C
63: prt "Y=A+B*X";ret
64: prt "1/Y=A+B*X";ret
65: prt "Y=A+B*X↑2";ret
66: prt "Y=A+B/X";ret
67: prt "1/Y=A+B/X";ret
68: prt "Y=A+B/X↑2";ret
69: prt "1/Y=A+B/X↑2";ret
70: prt "Y=A+B*√X";ret
71: prt "Y=A+B/√X";ret
72: prt "1/Y=A+B/√X";ret
73: prt "Y=A*X↑B";ret
74: prt "Y=A*exp(B*X)";ret
75: prt "Y=A*exp(B*X↑2)";ret
76: prt "Y=A+B*ln(X)";ret
77: prt "Y=A+B*X+C*X↑2";ret
78: "T":jmp C
79: B+AX→Y;ret
80: 1/(B+AX)→Y;ret
81: B+AXX→Y;ret
82: B+A/X→Y;ret
83: 1/(B+A/X)→Y;ret
84: B+A/XX→Y;ret
85: 1/(B+A/XX)→Y;ret
86: B+A√X→Y;ret
87: B+A/√X→Y;ret
88: 1/(B+A/√X)→Y;ret
89: BX↑A→Y;ret
90: Bexp(AX)→Y;ret
91: Bexp(AXX)→Y;ret
92: B+Aln(X)→Y;ret
93: B+AX+XXr7→Y;ret
94: end
*31573
```

HISTOGRAM

The program generates, prints, and plots histogram for a series of data which are entered via either the calculator's keyboard, the 9863A Paper Tape Reader or from the data cartridge.

The program outputs for each cell of the histogram:

- no. of cell
- absolute frequency in cell
- relative frequency in cell

In addition, the following characteristics are calculated: X_{min}, X_{max}, range, mean, variance, skewness, kurtosis, number of data. It is possible to plot a normal curve.

Reading of the data from the Paper Tape or entering from the keyboard is finished and the output of the results is started through flg 13. Thus, the character causing the sfg 13-instruction into the calculator must be located at the end of the paper-tape.

On the cartridge, the data must be stored on files of equal size (max. 540 bytes (8 bytes is 1 data item)).

Either the last number must be the last number of the last file or the last number must be followed by 9.10^{99}. (e.g. 9e99)

HARDWARE REQUIREMENTS: 9825A (standard 6.844 bytes)

MEMORY: 2162 bytes

ROMS: 98212A 9862 Plotter-General I/O or 98214A 9862 Plotter-General I/O-Extended I/O.

PERIPHERALS: 9862A Plotter
9863A Paper Tape Reader (optional)

STEP	DISPLAY	INSTRUCTIONS
1		ldf N. Load the file on which the program is stored.
2		RUN
3	PRINT DATA?	CONTINUE for input data-print
		0, CONTINUE if no data-print required.
4	INPUT	enter input code, CONTINUE. Input code is printed:
		1 Keyboard
		2 Paper Tape
		3 Cartridge
5	FIRST FILE	Number of the first file of cartridge on which the data is stored, CONTINUE.
6	LAST FILE	Number of the last file of cartridge on which the data is stored, CONTINUE.
7	FILE SIZE	Enter file size in bytes, CONTINUE
8	FXD/FLT (FIG)	f, CONTINUE. (f determines the format of description of x-axis resp. $f < 0, f \geqslant 0$,
		fxd f, flt (f)
9	FXD/FLT (PRINT)	f', CONTINUE. (f'determines the format of the results printed. $f' < 0, f' \geqslant 0$, fxd f',
		flt (f')
10	CELL WIDTH	Enter cell-width, CONTINUE
11	X0	Enter lower limit of the first cell, CONTINUE
12	MAX. NO. CELLS	Enter number of cells, CONTINUE
		CONTINUE: then 100 cells are supposed.
13	X	Enter data, CONTINUE
		CONTINUE, if all data is entered.
14	NORMAL CURVE?	CONTINUE, for normal curve overlay.
		0, CONTINUE if normal curve is not desired.
		Note: i-th cell of the histogram is an interval.
		$< x_0 + (i-1)\,h, x_0 + ih)$
		where h is the cell width.

EXAMPLE:

There are these numbers on the punched tape:

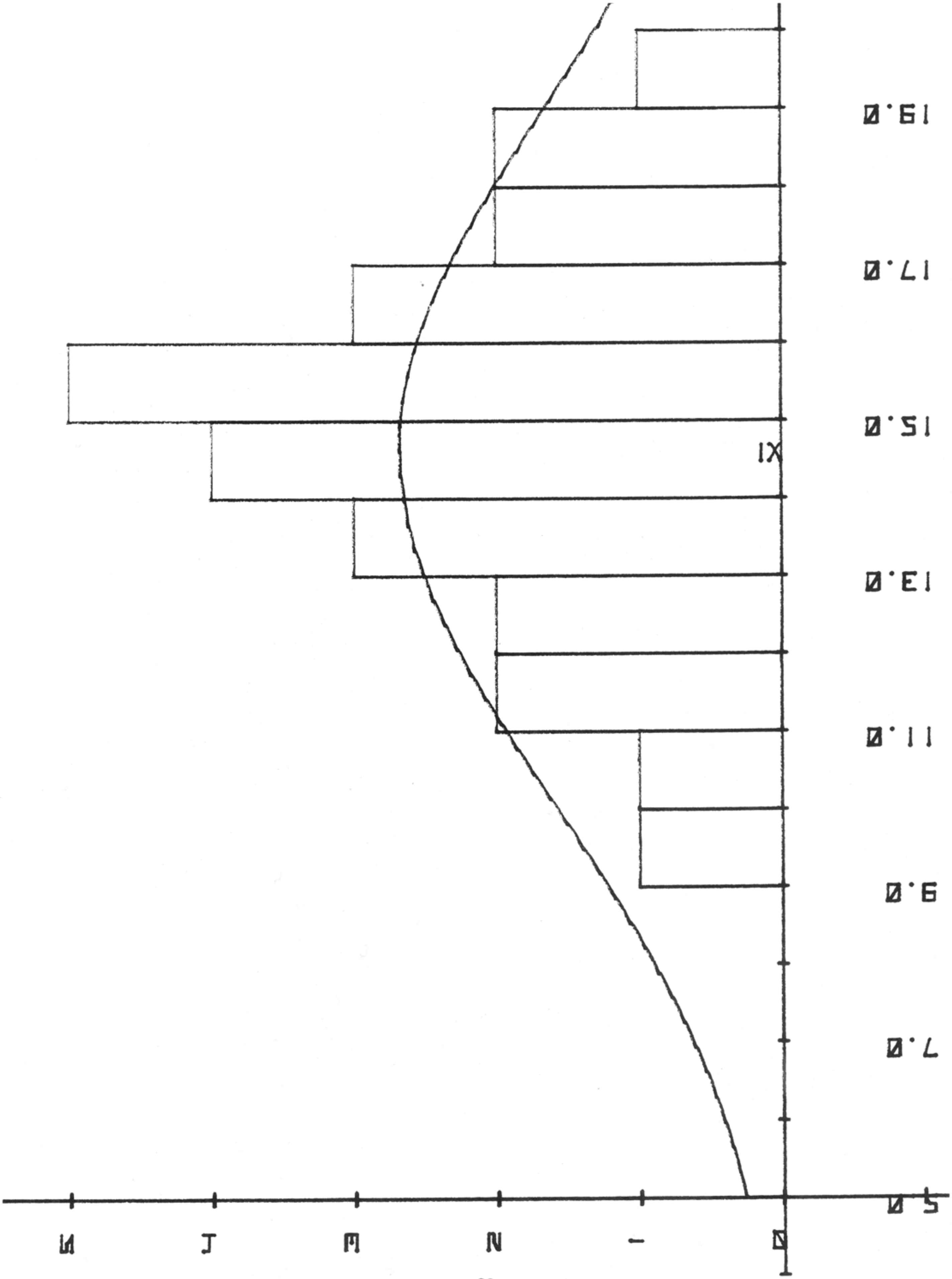
9.1 11.5 15.6 15 12.2 14.1 15.3 17.9 10.5 26.3 14.2 15.6
16.1 18.4 21.6 11.2 13.1 15.1 14.7 16.8 13.6 3.8 12 13
19.3 18.5 14.2 16.8 17.7 4.4 ,

The results are:

```
INPUT CODE:

KEYBOARD        1
TAPE READER     2
CARTRIDGE       3

CELL WIDTH
         1.00000
X0       5.00000

         9.10000
        11.50000
        15.60000
        15.00000
        12.20000
        14.10000
        15.30000
        17.90000
        10.50000
        26.30000
        14.20000
        15.60000
        16.10000
        18.40000
        21.60000
        11.20000
        13.10000
        15.10000
        14.70000
        16.80000
        13.60000
         3.80000
        12.00000
        13.00000
        19.30000
        18.50000
        14.20000
        16.80000
        17.70000
         4.40000
```

```
CELL NO.
NO.OBS.IN CELL
REL.FREQUENCY
         1.00000
         0.00000
         0.00000

         2.00000
         0.00000
         0.00000

         3.00000
         0.00000
         0.00000

         4.00000
         0.00000
         0.00000

         5.00000
         1.00000
         0.03333

         6.00000
         1.00000
         0.03333

         7.00000
         2.00000
         0.06667

         8.00000
         2.00000
         0.06667

         9.00000
         3.00000
         0.10000

        10.00000
         4.00000
         0.13333

        11.00000
         5.00000
         0.16667
```

```
        12.00000
         3.00000
         0.10000

        13.00000
         2.00000
         0.06667

        14.00000
         2.00000
         0.06667

        15.00000
         1.00000
         0.03333

        16.00000
         0.00000
         0.00000

X MIN
X MAX
RANGE    3.80000
        26.30000
        22.50000

NO.LESS THAN X0
         2.00000
NO.TOO LARGE
         2.00000

MEAN    14.59667
VARIANCE
        20.02671
SKEWNESS
        -0.14331
KURTOSIS
         4.01062
N       30.00000
```


19.0
17.0
15.0
X̄
13.0
11.0
9.0
7.0
5.0
0
1
2
3
4
5

PROGRAM LISTING

```
0: cfg 1;cfg 13;-(1e99→r3)→r4;fxd 0
1: ent "PRINT DATA?",X;if flg13;sfg 1
2: prt "INPUT CODE:","","KEYBOARD        1","TAPE READER    2"
3: prt "CARTRIDGE       3";spc 8;ent "INPUT",r2;jmp 1+2(r2#3)
4: ent "FIRST FILE NO.",r7;r7-1→r7;ent "LAST FILE NO.",r8,"FILE SIZE",r9
5: r9/8→r9;prt "FROM FILE NO.",r7+1,"TO FILE NO.",r8,"FILE SIZE",r9
6: spc 2;ent "FXD/FLT(FIG)",r0,"FXD/FLT(PRINT)",r1
7: ent "CELL WIDTH",A,"X0",B;100→Y;fxd abs(r1);if 0>r1;flt -r1
8: prt "CELL WIDTH",A,"X0",B;spc 2;ent "MAX.NO.CELLS",Y;cfg 13
9: "A":jmp 8+(4r2-11)r2
10: ent X;jmp 2
11: red 7,X
12: if flg13;gto "B"
13: "C":if flg1;prt X
14: C+1→C;if B>X;r5+1→r5;gto +3
15: if B+AY<=X;r6+1→r6;gto +2
16: rint((X-B)/A+14)+1→rint((X-B)/A+14)
17: if X>r4;X→r4
18: if r3>X;X→r3
19: r10+X→r10;r11+XX→r11;r12+XXX→r12;r13+XXXX→r13;gto "A"
20: "D":if Z#0;gto +3
21: if (r7+1→r7)>r8;rew;gto "B"
22: ldf r7,r114
23: rint((Z+1→Z)+113)→X;if X=9e99;rew;gto "B"
24: gto "C";if Z=r9;0→Z
25: "B":if (int((r4-B)/A+1)→Z)<=Y;Z→Y
26: spc 4;prt "CELL NO.","NO.OBS.IN CELL","REL.FREQUENCY";spc ;1→Z
27: prt Z,rint(Z+13),rint(Z+13)/C,"";jmp (Z+1→Z)>Y
28: spc 3;0→r7
29: if (Z-1→Z)>0;gto +0;if rint(Z+13)>r7;rint(Z+13)→r7
30: scl B-A,B+A(Y+1),-.15(r7+3),1.1r7;1→r8
31: if r7>20;5→r8;if r7>80;10→r8;if r7>200;50→r8
32: if r7>800;100→r8;if r7>2e3;500→r8
33: axe B,0,A,r8;prt "X MIN","X MAX","RANGE",r3,r4,r4-r3;spc
34: if r5>0;prt "NO.LESS THAN X0",r5
35: if r6>0;prt "NO.TOO LARGE",r6
36: spc 2;1→Z;B→X
37: plt X,0;plt X,r(Z+13);plt X+A→X,r(Z+13);plt X,0;jmp (Z+1→Z)>Y
38: 0→Z;1→r9;if Y>10;2→r9;if Y>25;5→r9;if Y>50;10→r9
39: fxd abs(r0);if 0>r0;flt -r0
40: ltr B+AZ,-.15(r7+3),222;lbl B+AZ;jmp (Z+r9→Z)>Y
41: 0→Z;fxd 0
42: ltr B-A,Z,221;lbl Z;jmp (Z+r8→Z)>r7
43: fxd abs(r1);if 0>r1;flt -r1
44: prt "MEAN",r10/C→r5,"VARIANCE",(r11-Cr5r5)/(C-1)→r6,"SKEWNESS"
45: prt (r12-3r5r11+2Cr5r5r5)/Cr6√r6,"KURTOSIS"
46: prt (r13-4r5r12+6r5r5r11-3Cr5r5r5r5)/Cr6r6,"N",C;spc 8
47: ltr r5-.0064A(Y+2),.001r7;lbl "X"
48: ltr r5-.0064A(Y+2),.015(1.25r7+.5);lbl "-"
49: ent "NORMAL CURVE?",Z;if 1-flg13
50: AY/50→r2;B→X;AC/√(2πr6)→Z
51: plt X,Zexp(-(X-r5)(X-r5)/2r6);jmp (X+r2→X)>B+AY
52: pen;end
*2667
```

MODEL 9825A

DESIGN OF SMALL TRANSFORMERS WITH POWER RATINGS FROM 2,5 TO 220 VA

This program is for the design of small transformers with a maximum of three secondary windings. It is arranged in such a way that no special knowledge of the physical principles of transformer operation is required when using it.

After entering the desired current and voltage values along with the flux density and the iron quality, two to four standard core sizes are printed out for consideration.

The selected transformer is optimised and the key values and winding data are printed.

Optimisation is on the basis of minimum power losses and maximum use of winding space.

The program contains the following standard core sizes:

E142, M42, EI46, EI54, M55, EI60, EI66, M65, EI78, M74, EI84A, M85A, EI84B, M85B, M102A, M102B.

The internal winding table covers all standard (laquered) wire sizes from 0.07 to 2.12 mm. diameter.

Notes:

The voltages are entered in volts, currents in amps. When choosing the maximum flux density it should be kept in mind that even good material reaches the verge of saturation at around 15000 Gauss. No limit is incorporated in the program. Guideline: 10000 to 14000 Gauss.

The loss figure V_{10} is a specific value for the core material used. Guideline: 1.7 to 3.0 Watts/kg. For a power frequency of 60Hz the figure 50 in line 59 is to be changed to 60.

HARDWARE REQUIREMENTS: 9825A (standard 6.844 bytes)

MEMORY: 1st program: 3334 bytes. 2nd. program 2814 bytes. 3rd. program 1718 bytes.

ROMS:

PERIPHERALS:

STEP	DISPLAY	INSTRUCTIONS
		Load Fileno. On which the 1st. program is stored.
		RUN
1	U PRIM.	Enter primary voltage CONTINUE
2	U SEC. 1	Enter 1st. secondary voltage CONTINUE
3	I SEC. 1 A	Enter 1st. secondary current (Ampère) CONTINUE
4	U SEC. 2	When not available, press 0, Continue.
		The program then goes to step 8.
5	I SEC. 2 A	
6	U SEC. 3	analogous to 4
7	I SEC 3 A	
8	B. MAX. GAUSS	Enter maximum induction. CONTINUE
		Approximate value: 11000–13500 Gauss.
9	V 10 W/kg.	Enter loss factor of core lamination. CONTINUE
		Approximate value: 1.3 – 3.0 W/kg.
10		On the printer 2 – 4 possible standard core sizes are given with a identification figure
		and the power efficiency.
11	Which solution?	Enter selected identification figure. CONTINUE
12	Next part of program.	The next program is automatically loaded into the calculator.
13		The calculator dimensions the transformer.
14	Next part of program.	The last program is automatically loaded into the calculator and the calculator prints
		out the transformer-data.

EXAMPLE

```
POSSIBLE
   SOLUTIONS


(12) M 85A

PSEC (VA)  65.00
PSEC/P MAX(%)
          109.33


(13) EI 84B

PSEC (VA)  65.00
PSEC/P MAX(%)
          103.65


(14)M 85B

PSEC (VA)  65.00
PSEC/P MAX(%)
           75.00


(15)M 102A

PSEC (VA)  65.00
PSEC/P MAX(%)
           61.90
```

```
TRANSFORMER
(13) EI 84B

PSEC (VA)  65.00
PSEC/P MAX(%)
          103.65
B MAX(GAUSS)
        10000.00
V  10(W/KG) 1.50


SPACE FACTOR
OF WINDING (%)
              96

LOSSES  (W)
P CU PRIM.(W)
          52.000
P CU SEC1(W)
           1.158
P CU SEC2(W)
           0.589
P CU SEC3(W)
           0.279
P CU TOTAL 3.942
P FE (W)   2.100
P V TOTAL ------
---        6.042

EFFICIENCY(%) 91
TEMP.RISE (C) 23

RESISTANCE
R PRIM (OHM)
           15.24
R SEC1(OHM) 0.51
R SEC2(OHM) 0.59
R SEC3(OHM) 1.11

CURRENT DENSITY
MAX (A/MM)   3.6
PRIM(A/MM)   2.2
SEC1(A/MM)   2.7
SEC2(A/MM)   2.5
SEC3(A/MM)   3.2

OPEN CIRCUIT
VOLTAGE(V)
SEC1(VOLT) 26.51
SEC2(VOLT) 21.18
SEC3(VOLT  16.09
```

```
  WINDING DATA
----------------
1. PRIMARY
V PRIM.(V)
          220.00
I PRIM.(MA)
         387.504
WIND.PRIM.   843
WIRE PRIM.(MM)
           0.450

2. SECONDARY 1
V SEC.1(V) 25.00
I SEC1 (MA)
        1500.000
WIND.SEC1    102
WIRE SEC1(MM)
           0.850

3. SECONDARY 2
V SEC 2(V) 20.00
I SEC2(MA)
         1000.00
WIND.SEC2     81
WIRE SEC2(MM)
           0.710

4. SECONDARY 3
V SEC3 (V) 15.00
I SEC3(MA)
          500.00
WIND.SEC3     62
WIRE SEC3(MM)
           0.450
```

1st. part of program.

```
0: gto +3
1: dsp "NEXT PART OF PROGRAM"
2: ldf F,3,3
3: ent "U PRIM",r1,"U SEC1",r2,"I SEC1(A)",r3,"U SEC2",r4;if r4=0;gto +4
4: ent "I SEC2(A)",r5,"U SEC3",r6
5: if r6=0;gto +2
6: ent "I SEC3(A)",r7
7: ent "B MAX(GAUSS)",r8,"V 10(W/KG)",r0;cfg 5;60→r9
8: r2r3+r4r5+r6r7→r10
9: prt "POSSIBLE","   SOLUTIONS";0→A;spc 2;cfg 1
10: if flg5=1;3→A
11: 1.4→r11→r12;.12→r13;1.89→r14;.54→r15;7.93→r16;17.8→r17
12: gto "C1"
13: if flg5=1;6→A
14: 1.5→r11;1.2→r12;.13→r13;2.6→r14;.7→r15;8.3→r16;14→r17
15: gto "C1"
16: if flg5=1;9→A
17: 1.6→r11→r12;.2→r13;2.2→r14;.64→r15;9→r16;14→r17
18: gto "C1"
19: if flg5=1;12→A
20: 1.8→r11→r12;.24→r13;2.45→r14;.72→r15;10.3→r16;11→r17
21: gto "C1"
22: if flg5=1;15→A
23: 2→r11;1.7→r12;.3→r13;3.35→r14;.85→r15;10.95→r16;8.29→r17
24: gto "C1"
25: if flg5=1;18→A
26: 2→r11→r12;.34→r13;2.7→r14;.81→r15;11.4→r16;8.23→r17
27: gto "C1"
28: if flg5=1;21→A
29: 2.2→r11→r12;.45→r13;3→r14;.91→r15;12.52→r16;7.2→r17
30: gto "C1"
31: if flg5=1;24→A
32: 2.7→r11;2→r12;.6→r13;3.9→r14;1.1→r15;13.6→r16;5.6→r17
33: gto "C1"
34: if flg5=1;27→A
35: 2.6→r11→r12;.74→r13;3.5→r14;1.11→r15;14.7→r16;5→r17
36: gto "C1"
37: if flg5=1;30→A
38: 3.2→r11;2.3→r12;.9→r13;4.5→r14;1.2→r15;15.66→r16;4.3→r17
39: gto "C1"
40: if flg5=1;33→A
41: 2.8→r11→r12;.92→r13;3.8→r14;1.21→r15;15.9→r16;4.3→r17
42: gto "C1"
43: if flg5=1;36→A
44: 3.2→r11;2.9→r12;1.28→r13;4.9→r14;1.13→r15;16.7→r16;3.6→r17
45: gto "C1"
46: if flg5=1;39→A
47: 4.2→r11;2.8→r12;1.4→r13;3.8→r14;1.21→r15;16.65→r16;3.8→r17
48: gto "C1"
49: if flg5=1;42→A
50: 4.5→r11;2.9→r12;1.88→r13;4.9→r14;1.13→r15;19.3→r16;3.2→r17
51: gto "C1"
52: if flg5=1;45→A
53: 3.5→r11;3.4→r12;1.98→r13;6.1→r14;1.35→r15;19.4→r16;2.6→r17
54: gto "C1"
55: if flg5=1;48→A
```

```
56: 5.2→r11;3.4→r12;2.95→r13;6.1→r14;1.35→r15;22.8→r16;2.3→r17
57: gto "C1"
58: "C1":r15+r15↑2/3→Z;.61√(r9/Z)→r18
59: 4.44*1e-8*r8r11r12*50→r19
60: 2r18r16*1.23*.0175/r19→r20
61: r19r18r15r14*.4*.4(1-r20/100)100→r21
62: if flg5=0;A+3→A
63: if A>48;sfg 8
64: if r21>1.96r10;sfg 8;if A>3;cfg 8
65: if flg8=1;prt "NO SOLUTION"
66: if flg8=1;prt "THIS PROGRAM";spc 8;cfg 8;gto 0
67: if flg5=0;if r21<=.7r10;jmp -57+A
68: if flg5=0;if r21>1.95r10;gto +22
69: if flg5=1;spc 2;prt "TRANSFORMER"
70: if A=3;prt "(1) EI 42"
71: if A=6;prt "(2) M 42"
72: if A=9;prt "(3) EI 48"
73: if A=12;prt "(4)  EI 54"
74: if A=15;prt "(5)  M55"
75: if A=18;prt "(6)  EI 60"
76: if A=21;prt "(7)  EI 66"
77: if A=24;prt "(8)  M65"
78: if A=27;prt "(9) EI 78"
79: if A=30;prt "(10) M 74"
80: if A=33;prt "(11) EI 84A"
81: if A=36;prt "(12) M 85A"
82: if A=39;prt "(13) EI 84B"
83: if A=42;prt "(14)M 85B"
84: if A=45;prt "(15)M 102A"
85: if A=48;prt "(16)M 102B"
86: spc 1;fxd 2;100r10/r21→Z;prt "PSEC (VA)",r10,"PSEC/P MAX(%)",Z
87: if flg5=1;prt "B MAX(GAUSS)",r8,"V  10(W/KG)",r0;gto 107
88: if A=48;gto +2
89: if flg5=0;spc 2;jmp -79+A
90: if flg5=0;spc 6;ent "WHICH SOLUTION",Z;sfg 5
91: if Z=1;gto 10
92: if Z=2;gto 13
93: if Z=3;gto 16
94: if Z=4;gto 19
95: if Z=5;gto 22
96: if Z=6;gto 25
97: if Z=7;gto 28
98: if Z=8;gto 31
99: if Z=9;gto 34
100: if Z=10;gto 37
101: if Z=11;gto 40
102: if Z=12;gto 43
103: if Z=13;gto 46
104: if Z=14;gto 49
105: if Z=15;gto 52
106: if Z=16;gto 55
107: 15→F;gto 1
*22315
```

2nd. part of program.

```
0: dsp " ";1.2r10/r1→r22;0→A→B→C→X→Y→r60;if r10<=15;1.1r22→r22
1: .07→r70;.08→r71;.09→r72;.1→r73;.112→r74;.125→r75;.14→r76
2: .16→r77;.18→r78;.2→r79;.224→r80;.25→r81;.28→r82;.315→r83
3: .355→r84;.4→r85;.45→r86;.5→r87;.56→r88;.63→r89;.71→r90
4: .75→r91;.8→r92;.85→r93;.9→r94;.95→r95;1→r96;1.06→r97
5: 1.12→r98;1.18→r99;1.25→r100;1.32→r101;1.4→r102;1.5→r103
6: 1.6→r104;1.7→r105;1.8→r106;1.9→r107;2→r108;2.12→r109
7: 1.13√(r22/r18)→Z;B+1→B
8: if r(69+B)<=Z;gto -1
9: if r(69+B)>Z;r(69+B)→r24
10: "C2":r1/r19→r25;22r25r16/(r24↑2*10↑5)→r26
11: if flg1=1;gto "C8";cfg 1
12: 1.13√(r3/r18)→Z;C+1→C
13: if r(69+C)<=Z;gto -1
14: if r(69+C)>Z;r(69+C)→r27
15: "C3":r2/r19→Z;22Zr16/(r27↑2*10↑5)→r28
16: r22r26r2/r1+r3r28→Z;r2+Z→r29
17: r29/r19→r30;22r30r16/(r27↑2*10↑5)→r31
18: if flg2=1;gto "C8";cfg 2
19: if r4=0;gto +8
20: 1.13√(r5/r18)→Z;X+1→X
21: if r(69+X)<=Z;gto -1
22: if r(69+X)>Z;r(69+X)→r32
23: "C4":r4/r19→Z;22Zr16/(r32↑2*10↑5)→r33
24: r22r26r4/r1+r5r33→Z;r4+Z→r34
25: r34/r19→r35;22r35r16/(r32↑2*10↑5)→r36
26: if flg3=1;gto "C8";cfg 3
27: if r6=0;gto +8
28: 1.13√(r7/r18)→Z;Y+1→Y
29: if r(69+Y)<=Z;gto -1
30: if r(69+Y)>Z;r(69+Y)→r37
31: "C5":r6/r19→Z;22Z16/(r37↑2*10↑5)→r38
32: r22r26r6/r1+r7r38→Z;r6+Z→r39
33: r39/r19→r40;22r40r16/(r37↑2*10↑5)→r41;cfg 4
34: "C8":r1/r19→r42;(.107r24)↑2*r42/r14→r43
35: (.107*r27)↑2*r30/r14→r44
36: (.107r32)↑2*r35/r14→r45
37: (.107r37)↑2*r40/r14→r46
38: .13+1.15(r43+r44+r45+r46)→r47
39: 1.277r22/r24↑2→r48;1.277r3/r27↑2→r49
40: if r4=0;gto +2
41: 1.277r5/r32↑2→r50
42: if r6=0;gto +2
43: 1.277r7/r37↑2→r51
44: r60+1→r60
45: if r47>.97r15;gto "C6"
46: if r47>.92r15;gto "C7"
47: if r60>50;gto "C7"
48: if r48>r49;sfg 1;r48→Z
49: if r49>r48;sfg 2;r49→Z;cfg 1
50: if Z<=r50;sfg 3;r50→Z;cfg 2
51: if Z<=r51;sfg 4;cfg 3
52: if flg1=1;B+1→B;r(69+B)→r24;gto "C2"
53: if flg2=1;C+1→C;r(69+C)→r27;gto "C3"
54: if flg3=1;X+1→X;r(69+X)→r32;gto "C4"
55: if flg4=1;Y+1→Y;cfg 4;r(69+Y)→r37;gto "C5"
```

```
56: "C6":cfg 1;cfg 2;cfg 3;cfg 4
57: if r(69+B)<=.07;gto +2
58: if r48<=r49;sfg 1;r48→Z
59: if r(69+C)<=.07;gto +3
60: if r49<=r48;sfg 2;r49→Z;cfg 1
61: if r4=0;gto +3
62: if r(69+X)<=.07;gto +3
63: if Z>50;sfg 3;r50→Z;cfg 2
64: if r6=0;gto +3
65: if r(69+Y)<=.07;gto +2
66: if Z>r51;sfg 4;cfg 3
67: if flg1=1;B-1→B;r(69+B)→r24;gto "C2"
68: if flg2=1;C-1→C;r(69+C)→r27;gto "C3"
69: if flg3=1;X-1→X;r(69+X)→r32;gto "C4"
70: if flg4=1;Y-1→Y;cfg 4;r(69+Y)→r37;gto "C5"
71: "C7":cfg 1;cfg 2;cfg 3;cfg 4;fxd 3
72: 16→F;gto 1
73: end
*29618
```

3rd. part of program.

```
0: dsp " ";spc 2;100r47/r15→Z;prt "SPACE FACTOR";fxd 0
1: prt "OF WINDING (%)",Z;spc 1;fxd 3
2: prt "LOSSES  (W)";r26r22↑2→r52;prt "P CU PRIM.(W)",52
3: r31r3↑2→r53;prt "P CU SEC1(W)",r53
4: if r4=0;gto +4
5: r36r5↑2→r54;prt "P CU SEC2(W)",r54
6: if r6=0;gto +2
7: r41r7↑2→r55;prt "P CU SEC3(W)",r55
8: r52+r53+r54+r55→r56;prt "P CU TOTAL",r56
9: (r8/10000)↑(2.2r8/10000)r0r13→Z
10: prt "P FE (W)",Z
11: Z+r56→r56;prt "P V TOTAL ---------",r56;spc 1;fxd 0
12: 100r10/(r10+r56)→Z;prt "EFFICIENCY(%)",Z
13: r17r56→Z;prt "TEMP.RISE (C)",Z;fxd 2
14: spc 1;prt "RESISTANCE","R PRIM (OHM)",r26
15: prt "R SEC1(OHM)",r31
16: if r4=0;gto +2
17: prt "R SEC2(OHM)",r36
18: if r6=0;gto +2
19: prt "R SEC3(OHM)",r41
20: fxd 1;spc 1;prt "CURRENT DENSITY","MAX (A/MM)",r18
21: prt "PRIM(A/MM)",r48,"SEC1(A/MM)",r49
22: if r4=0;gto +2
23: prt "SEC2(A/MM)",r50
24: if r6=0;gto +2
25: prt "SEC3(A/MM)",r51
26: fxd 2;spc 1;prt "OPEN CIRCUIT    VOLTAGE(V)","SEC1(VOLT)",r29
27: if r4=0;gto +2
28: prt "SEC2(VOLT)",r34
29: if r6=0;gto +2
30: prt "SEC3(VOLT",r39
31: spc 2;prt "  WINDING DATA","----------------"
32: prt "1. PRIMARY"
33: prt "V PRIM.(V)",r1;1000r22(r56+r10)/r10→r22;fxd 3
34: prt "I PRIM.(MA)",r22;fxd 0
35: prt "WIND.PRIM.",r42;fxd 3
36: prt "WIRE PRIM.(MM)",r24;fxd 2
37: spc 1;prt "2. SECONDARY 1"
38: prt "V SEC.1(V)",r2;1000r3→r3;fxd 3
39: prt "I SEC1 (MA)",r3;fxd 0
40: prt "WIND.SEC1",r30;fxd 3
41: prt "WIRE SEC1(MM)",r27
42: if r4=0;gto +4
43: spc 1;prt "3. SECONDARY 2";fxd 2
44: prt "V SEC 2(V)",r4;1000r5→r5;prt "I SEC2(MA)",r5;fxd 0
45: prt "WIND.SEC2",r35;fxd 3;prt "WIRE SEC2(MM)",r32
46: if r6=0;gto +4
47: spc 1;prt "4. SECONDARY 3";fxd 2
48: prt "V SEC3 (V)",r6;1000r7→r7;prt "I SEC3(MA)",r7;fxd 0
49: prt "WIND.SEC3",r40;fxd 3;prt "WIRE SEC3(MM)",r37
50: spc 8
51: end
*23529
```

MEMORY ALLOCATION

VARIABLES							
A	Transformer address			r43	layer heigth Prim.		
B	Primary Optimisation			r44	layer heigth Sec. 1		
C	Sec. 1 "			r45	layer heigth Sec. 2		
X	Sec. 2 "			r46	layer heigth Sec. 3		
Y	Sec. 3 "			r47	layer heigth total		
Z	Intermediate value			r48	Current density Prim.		
				r49	Current density Sec. 1		
r1	V Primary			r50	Current density Sec. 2		
r2	V Sec. 1			r51	Current density Sec. 3		
r3	I Sec. 1			r52	Copper loss Prim.		
r4	V Sec. 2			r53	Copper loss. Sec. 1.		
r5	I Sec. 2			r54	Copper loss. Sec. 2		
r6	V. Sec. 3			r55	Copper loss Sec. 3		
r7	I. Sec. 3			r56	Copper loss total		
r8	B. MAX.						
r9	max. wire temp.			r70	Wire Table		
r10	P Sec.						
r11	lamination thickness						
r12	lamination width						
r13	dead weitht			r109			
r14	layer width.						
r15	layer height						
r16	turn length						
r17	heating factor						
r18	max. current density						
r19	spec. turn number						
r20	intermediate value						
r21	transferable power						
r22	I Prim.						
r23	—						
r24	Wire Prim						
r25	Turnno. Prim						
r26	Resist. Prim					FLAGS	
r27	Wire Sec. 1					0	
r28	Intermediate value					1	Primary Optimisation
r29	Uo Sec. 1					2	Sec. 1 Optimisation
r30	Turnno. Sec. 1					3	Sec. 2 Optimisation
r31	Resist. Sec. 1					4	Sec. 3 Optimisation
r32	Wire Sec. 1					5	—
r33	Intermediate value					6	—
r34	Uo Sec. 2					7	—
r35	Turnno. Sec. 2					8	No solution
r36	Resist. Sec. 2					9	—
r37	Wire Sec. 2					10	—
r38	Intermediate Value					11	—
r39	Uo Sec. 3					12	—
r40	Turn no. Sec. 3					13	—
r41	Resist. Sec. 3					14	—
r42	Turnno. Prim.					15	—

PIPE NETWORK ANALYSIS HARDY – CROSS METHOD

This program balances flows in a pipe network using the Hardy-Cross method of successive approximations. Head losses are computed using the Hazen-Williams formula. The maximum size network which can be analyzed is 20 loops and 50 pipes.

The inputs required are:

- Pipe and loop indentification numbers
- Pipe diameter (inches)
- Pipe length (feet)
- C-factor for Hazen-Williams formula
- Estimated flow in million galons per 24 hours (MGD)
- Convergence factor

The output is:

- Flow in pipe (MGD)
- Head loss in pipe (feet and pounds per square inch)
- Velocity in pipe (feet per second)

C Factors may be as high as 140 for new pipe and as low as 40 or 50 for old, badly tuberculated or rough pipe. The user is referred to standard hydraulics texts and may read: "Hydraulic Tables" by G. S. Williams and A. Hazen, 3rd Edition Revised, Jon Wiley & Sons, Inc. New York, 1933.

HARDWARE REQUIREMENTS: 9825A (standard 6.844 bytes)

MEMORY: 1412 bytes.

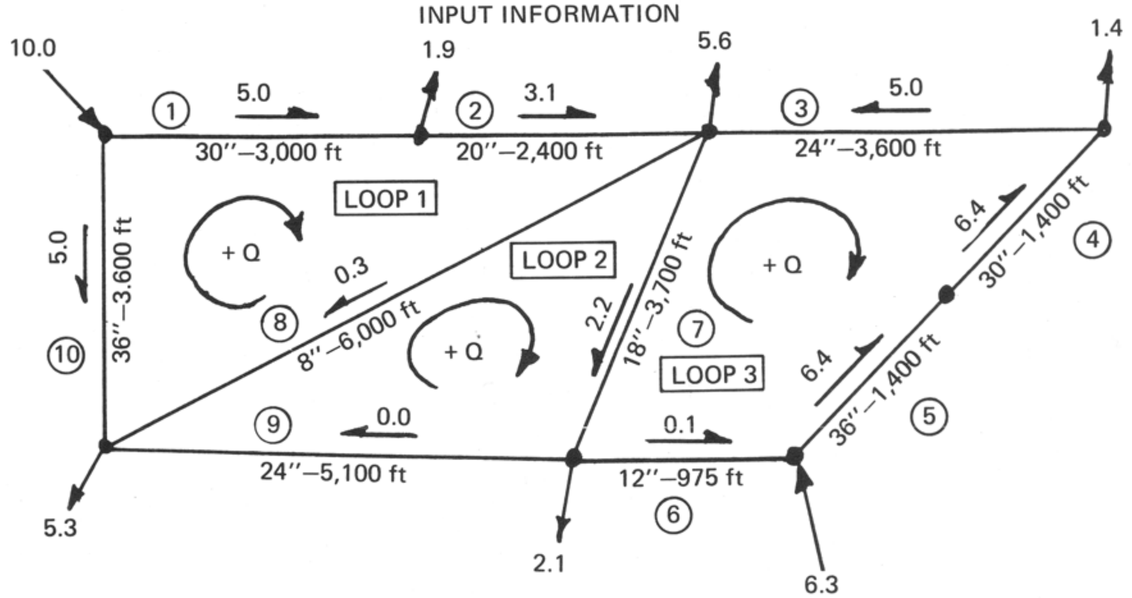
ROMS:

PERIPHERALS:

STEP	DISPLAY	INSTRUCTIONS
1		ldf N (load the file on which the program is stored)
		RUN
2	LINES?	Enter the total number of pipe lines in the problem (50 Max) CONTINUE
3	LOOPS?	Enter the total number of loops in the problem (20 Max) CONTINUE
4	CONVER. FACTOR	Enter Convergence Factor. (The smaller the number, the more accurate the solution)
		0.1 will usually give sufficient accuracy. CONTINUE
5	LINE NO.	Enter the first line number. (The maximum number of lines is 50) CONTINUE
6	PRI. LOOP	Enter the primary loop number (the loop from which the algebraic sign of the flow,
		+ or – , will be assigned to this pipe). CONTINUE
7	SEC. LOOP	Enter the number of the other loop in which this pipe appears, if any. If none, enter 0.
		CONTINUE
8	DIAM.	Enter the diameter of this pipe in inches. CONTINUE
9	LENGTH	Enter the length of this pipe in feet. CONTINUE
10	C–FACTOR	Enter the C Factor for this pipe (Hazen-Williams) CONTINUE
11	Q	Enter the estimated flow in this pipe in MGD. The entry must be consistent with sign
		convention adopted in Step 6. Use either clockwise or counter-clockwise flow around
		each loop as positive; however, this must be consistent throughout each problem.
		CONTINUE
12	LINE NO.	Repeat Steps 5–11 until the data for each line is entered. CONTINUE
13	1, 2, 3	After the data for the final pipe is entered, the calculator will begin the iterative pro-
		cedure and display the number of the iteration as it is completed. When the conver-
		gence factor is satisfied, the calculator will print the output, then return to Step 2 to
		begin a new problem.

EXAMPLE

INPUT INFORMATION



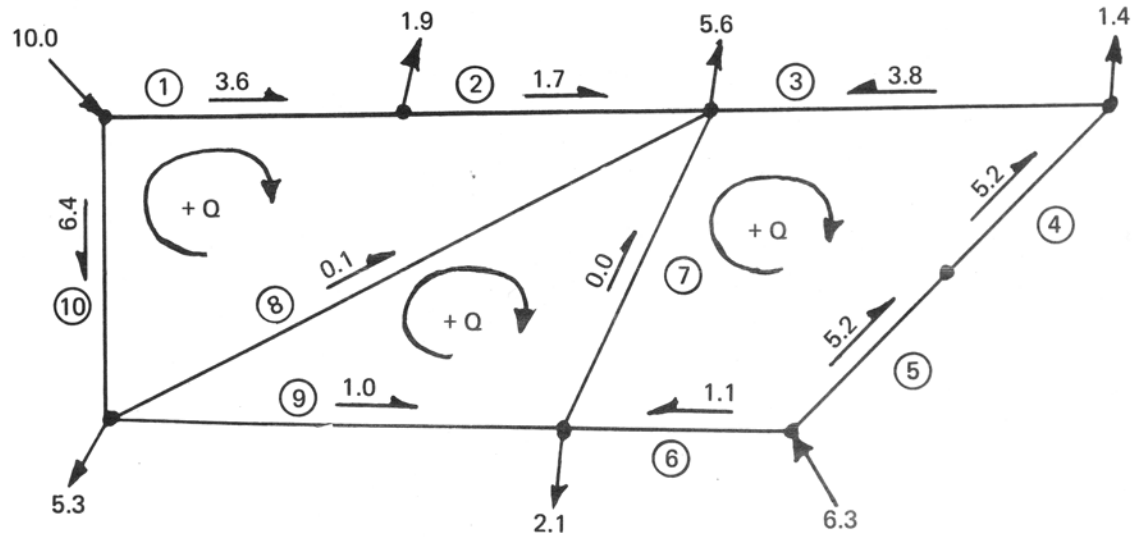
Pipes No. 6 & 9: C = 100
Other Pipes: C = 130
Outflows from network must equal inflows.
Flows in pipes are assumed; outflows at each pipe junction must equal inflows.

```
LINES        10.00                 3                        7
LOOPS         3.00                 3                        3
CONVER.FACTOR                      0                        2
              0.10                24                       18
                                3600                     3700
                                 130                      130
                              -5.000                   -2.200
LINE NO.
PRI LOOP
SEC LOOP                           4                        8
DIAM.IN.                           3                        1
LENGTH                             0                        2
C-FACTOR                          30                        8
FLOW(MGD)                       1400                     6000
                                 130                      130
                              -6.400                    0.300
                 1
                 1
                 0                 5                        9
                30                 3                        2
              3000                 0                        0
               130                36                       24
             5.000              1400                     5100
                                 130                      100
                              -6.400                    0.000
                 2
                 1
                 0                 6                       10
                20                 3                        1
              2400                 0                        0
               130                12                       36
             3.100               975                     3600
                                 100                      130
                              -0.100                   -5.000
```

The Example Solution is shown on the following page. *********

EXAMPLE SOLUTION

(flows rounded to nearest 0.1 MGD)



OUTPUT:

INTERATIONS 27

PIPE NO.	FLOW	HEAD LOSS(FT)	HEAD LOSS(PSI)	VEL FT/SEC
1	3.606	0.48	0.2	1.136
2	1.706	0.69	0.3	1.209
3	-3.788	-1.88	-0.8	-1.865
4	-5.188	-0.44	-0.2	-1.635
5	-5.188	-0.18	-0.1	-1.135
6	1.112	2.50	1.1	2.191
7	0.030	0.00	0.0	0.027
8	-0.076	-0.47	-0.2	-0.337
9	-1.018	-0.38	-0.2	-0.501
10	-6.394	-0.69	-0.3	-1.399

PROGRAM LISTING

```
0: cfg ;csv;fxd 2;50→X
1: ent "LINES?",A,"LOOPS",B,"CONV.FAC.",C
2: prt "LINES",A,"LOOPS",B,"CONVER.FACTOR",C;spc 3
3: prt "LINE NO.","PRI LOOP","SEC LOOP","DIAM.IN.","LENGTH"
4: prt "C-FACTOR","FLOW(MGD)";spc 2
5: fxd 0;X+1→X;if .X-50>A;gto 11
6: ent "LINE NO.",Y;prt Y;Y1e4→rX;ent "PRI. LOOP",Y;prt Y;Y1e2+rX→rX
7: ent "SEC.LOOP",Y;prt Y;Y+rX→rX;ent "DIAM.",Y;prt Y;Y.001+rX→rX
8: ent "LENGTH",Z;prt Z;ent "C-FACTOR",r0;prt r0
9: fxd 3;ent "Q",r(X-50);prt r(X-50)
10: Z(1917216.897/(r0↑(1/.54)*Y↑(2.63*(1/.54))))→r(X+161);spc 2;gto 5
11: 0→r0
12: 50→X
13: X+1→X;if X>A+50;gto 18
14: (r(X+161)abs(r(X-50))↑.852→r211)r(X-50)→r(X+50)
15: int(rX)→Y;Y-int(Y1e-4)1e4→Y;Y-(int(Y/100)→Z)100→Y
16: if Y#0;r(150+Y)+r211→r(150+Y);r(170+Y)-r(X+50)→r(170+Y)
17: r(150+Z)+r211→r(150+Z);r(170+Z)+r(X+50)→r(170+Z);gto 13
18: r0+1→r0;dsp r0;dsp ;dsp ;dsp ;170→Y
19: Y+1→Y;if Y-170>B;gto 31
20: if abs(rY)<=C;jmp -1
21: 170→Y
22: Y+1→Y;if Y-170>B;gto 24
23: .54rY/r(Y-20)→r(Y+20);jmp -1
24: 50→X
25: X+1→X;if X>A+50;gto 28
26: int(rX)→Y;Y-int(Y1e-4)1e4→Y;Y-(int(Y/100)→Z)100→Y
27: r(X-50)-r(190+Z)+(Y#0)r(190+Y)→r(X-50);gto -2
28: 150→X
29: X+1→X;if X>170;gto 12
30: 0→rX→r(X+20);jmp -1
31: prt "*********";spc ;prt "  OUTPUT:";spc 2;prt "INTERATIONS",r0;spc 2
32: prt "PIPE NO.","FLOW","HEAD LOSS(FT)","HEAD LOSS(PSI)","VEL FT/SEC"
33: spc 2;0→X
34: fxd 0;X+1→X;if X>A;prt "----------------";spc 9;gto 0
35: prt int(r(X+50)/1e4);fxd 3;prt rX;fxd 2;prt r(X+100)
36: fxd 1;prt r(X+100)/2.308
37: fxd 3;prt rX*1.547/(((r(X+50)-int(r(X+50)))1000/24)↑2*π)
38: spc ;prt "    ----";spc ;jmp -4
39: end
*18956
```

MEMORY ALLOCATION

VARIABLES							
A	Number of lines (pipes)			r171			
B	Number of loops			•	Total of		
C	Convergence Fac.			•	(Head loss/Q)		
X	Multi Purpose			•	for each loop		
y	Multi. Purpose			•			
z	Multi Purpose			r190			
r0	Multi Purpose			r191	Correction		
r1	Flow In			•	Factors for		
•	Pipes			•	each loop		
•	Pipe number			•			
•	same as			r210			
•	r Number						
r50				r211	$K (Q)^{1.85} = X Q$		
r51	Pipe Data			r212			
•	(Stored as one			•	K-factor of		
•	Number)			•	each pipe		
•	Pipe no. Pri. loop	diameter		r262			
•	0 1 1 2 11.						
•	secondary						
•	loop no.						
r100							
r101							
•	HL = Head						
•	Loss in each pipe						
•							
•							
r150							
r151							
•	Total Head					FLAGS	
•	loss for each					0	
•	loop					1	
•						2	
r170						3	
						4	
						5	
						6	
						7	
						8	
						9	
						10	
						11	
						12	
						13	
						14	
						15	

HORIZONTALLY MOVABLE PLANE FRAME

This program calculates the joint-moments in a horizontally movable plane frame by method of Hardy Cross.

The loads can be inputed (optional) in following forms:

- uniformly distrubuted load
- loads distributed in symmetric trapez. or triangular form
- direct input of fixed-end moments
- horizontal point-loads

The capacity is limited by : $6 (s + 1) (f + 1) + s + 2f \leqslant 162$

there is s . . . number of spans
f . . . number of floors

The moments of inertia are constant through every bar, but variable through the framework.

Input data are:

- number of spans
- number of floors
- admitted summa-error for one joint.
- spans (++)
- column heights (++)
- moments of inertia of beams (++) ; constant or variable
- optional: hinge or fixed end at the bottom — for each column; (also possible elastic bedded, by input of "I/H" for fictive underground column)
- moments of inertia of columns (++) ; constant or variable
- vertical loads
- horizontal (point) loads

(++) Input from left to right and from bottom to top. Printed, the data for each floor will be separated by "SPACE", (At vertical loads by "====")

The results are joint moments in form:

- number of joint (see fig. 1)
- moments (see fig. 2)

All the moments in the results and in the data (if direct inputed) are positive in a clockwise direction.

If the results will be summed up (flg 5 = 1), the horizontal point loads are positive from left to right.

Option of results:

- the joint-moments will be given for each load separate—; the number of loads is not limited.
- the joint-moments will be gradually summed up; the number of vertical loads is not limited, but the horizontal load is limited by one (in each floor). The horizontal load must be inputed as the last.

Input of "next load" —— see instructions.

In case of very strong columns and elastic beams, the convergence of results can be slow especially for horizontal loads. The admitted summa-error can be changed by pressing:

STOP; n → C; CONTINUE

Fig. 1 — Example of numbering for a "3 x 3" — frame.

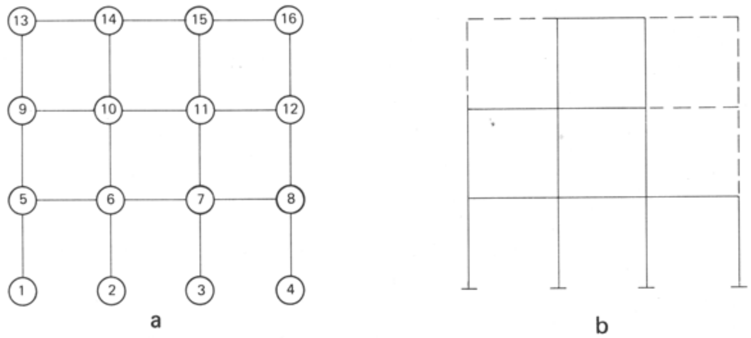
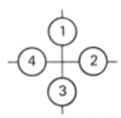


Fig. 2 — Succession of results in a joint.



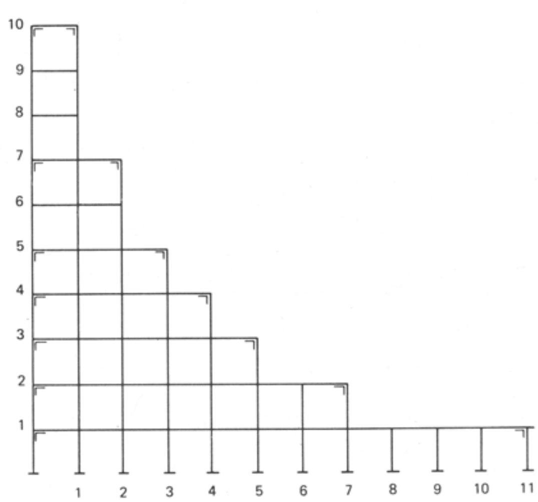
For elastic bedded column at the bottom, it's necessary to calculate the stiffness (I/H) of the column an·then the corresponding stiffness of the fictive underground column. For example:

Bedding 50% . . . . multiplicator 1,0
" 60% . . . . multiplicator 1,5 etc.

This fictive stiffness will be inputed in step 10 (see users instructions) instead of for hinge foreseen input of zero.

For a frame in form Fig. 1b, 0 (zero) must be inputed for the moments of inertia of all missing bars.

Fig. 3 — Scheme of capacity.



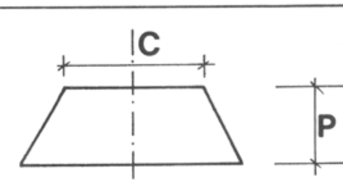
HARDWARE REQUIREMENTS: 9825A (standard 6.844 bytes)

MEMORY: 2546 bytes.

ROMS:

PERIPHERALS:

STEP	DISPLAY	INSTRUCTIONS
1		ldf N. Load the file on which the program is stored.
2		fxd N, EXECUTE (optional as desired)
3	flg.	Optional sfg N; sfg N; CONTINUE
		flg 1 = 1 mom. of inertia of beams = const.
		2 = 1 mom. of inertia of columns = const.
		3 = 1 direct input of fixed-end moments
		5 = 1 results will be summed up.
		7 = 1 the frame will be calculated as not movable.
		If no flags desired, only press CONTINUE
4	SPANS No.	n Number of spans CONTINUE
5	FLOORS No.	n Number of floors CONTINUE
6	TOL.	n Admitted summa-error for one joint (abs.) CONTINUE
		Tolerance is an approximate value. Because of timesaving the transference and equa-
		lisation of moments, are executed in one tour. So after last equalisation will remain the
		"back" and "down" sended transferences. Theoretically possible is a double tolerance,
		practically frequent about 1.5 x. Its recommended to take a half of the desired tol.
		value. Calculating the horizontal loads for summa of all column-moments of one
		floor 0.5 x tolerance will be considered for one column. This is: 1 xtol. for 2 columns,
		1.5 x tol. for 3 columns, 2xtol. for 4 columns and s.o.
7	L	n Spans (widths) for each span from left to right CONTINUE
8	H	n Column heights for each floor from bottom to top CONTINUE
9	IB	n Moments of inertia of beams from left to right and from bottom to top. (If flg 1 = 1
		– one inputs) CONTINUE
10	HI?	Optional for each column (at the bottom only): CONTINUE –– fixed end.
		0, CONTINUE –– hinged. Elastic bedded –– see program description.
11	IC	n Moments of inertia of columns from left to right and from bottom to top. (If flg
		2 = 1 – one input). CONTINUE
12	P UN	If uniformaly distr. load: n (load for length unit) CONTINUE (succession like step 9)
		If Trapez. distr. load: CONTINUE (appears step 13)
13	TR. C	n CONTINUE (see fig.)
14	P	n Load for Length unit
		(see fig.) CONTINUE

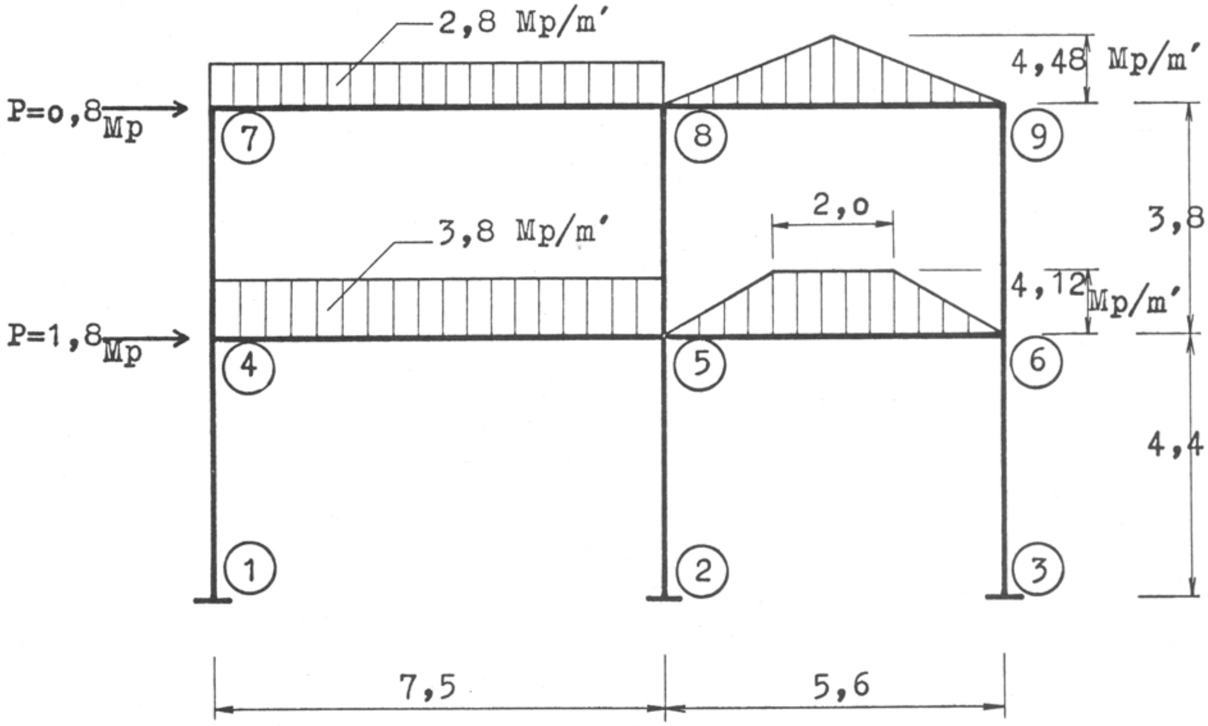


STEP	DISPLAY	INSTRUCTIONS
		Note: Steps 13 and 14 only appear if load in step 12 not inputed
15	ML	If flg 3 = 1 steps 12, 13 and 14 will stay out. If flg 3 = 0 steps 15 and 16 will stay out.
		n Fixed-end moment at left end of beam (positiv in Colckwise dir.) CONTINUE
16	MR	n Fixed-end moment at right end of beam (positiv in Clockwise dir.) CONTINUE
		Succession like step 9!
17	NEXT LO.	If next load vertical: cont 27 EXECUTE
		If next load horizontal: cont 59 EXECUTE
		(For vert. load program goes to step 12, for horizontal load to step 18).
18	P →	n Horizontal point loads – from bottom to top (positiv from left to right) CONTINUE
		The results are joint-moments. Printed:
		– number of joint (see fig. 1 – progr. descr.
		– moments (see fig. 2 – progr. descr.
		– only at the bottom: if hinge , a note "HINGE" will be printed
		– if calculated as no movable (flg 7 = 1) at the top of results a note
		"NO MOV. !" will be printed.
	Notes:	– to step 12: If no vertical loads press: Stop, cont 59 EXECUTE (program goes to
		step 18).
		– to flg 5 = 0. The results will be given for each load separage. Number of vertical
		and horizontal loads is not limited.
		– to flg 5 = 1 The results will be successively summed up. The number of vertical
		loads is not limited but the horizontal load is limited by one and must be inputed as
		last.
		– to flg 7 = 1 The frame will be calculated as no movable for unlimited number of
		vertical loads. If then a horizontal load inputed, flg 7 will be automatically cleared
		and the frame calculated for horizontal joint loads.
		– To the results for the bottom joints: There appears as 3rd. number a moment in
		the fictive underground column. These columns are foreseen to make possible hinges
		and elastic bedding at the bottom.

EXAMPLE 1

Guldan, Die Cross-Methode (Springer 1955)

1 a Horizontally movable plane frame (Sides 102 – 107)



Note: Because of example the form of load was changed for the beams 5-6 and 8-9.

All flags = 0.

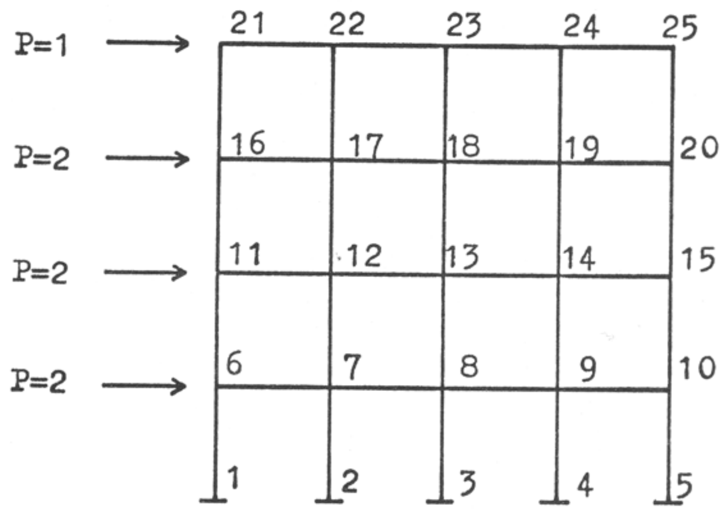
1b The same data, but joint-moments for vertical and horizontal load summed up (flg 5 = 1)

1c The same data. The frame calculated as no movable. (flg 7 = 1) – See Guldan Die Cr. Meth. Side 207.

Bar	b/h (cm)	I (dm4)
4–5, 5–6	25/70	71.5
7–8, 8–9	25/65	57.2
1–4	30/50	31.3
2–5	40/40	21.3
3–6, 4–7, 5–8	30/40	16.0
6–9	30/30	6.8

EXAMPLE 2

Takabeya, Mehrstöckige Rahmen (Ernst 1967) – Part C, Moment Tables I.125 and II.102



All spans: L = 1
" floors: H = 1

All beams: I = 1
" columns: I = 1

For verical loads: Fixed-end moments in all beams:

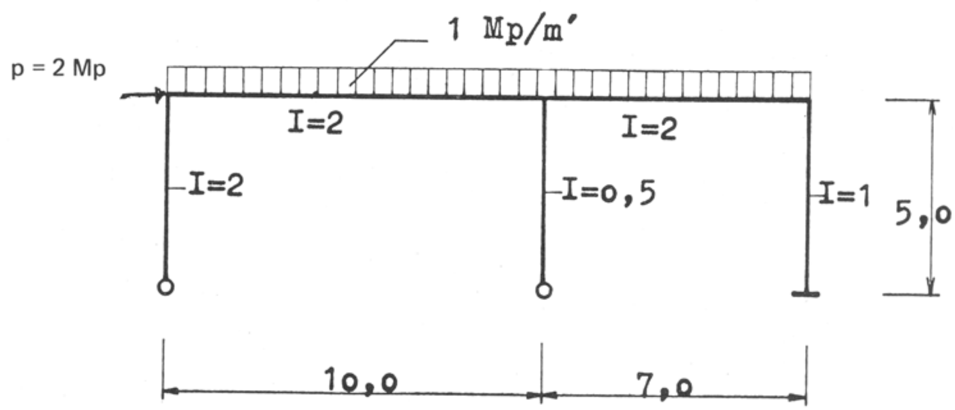
ML = 1.00
MR = – 1.00

(Calculated with: flg 1 = 1; flg 2 = 1; flg 3 = 1)

EXAMPLE 3

(With hinges)

Calculated with: flg 1 = 1



EXAMPLE 1a

```
FRAME

L
                7.50
                5.60
H
                4.40
                3.80

I BEAM
               71.50
               71.50

               57.20
               57.20

I COL.
               31.30
               21.30
               16.00

               16.00
               16.00
                6.80

VERT.LOAD

UN.             3.80
TR.C=           2.00
                4.12
====
UN.             2.80
TR.C=           0.00
                4.48
====
JOINT MOM.
                1.00

               -1.87
                0.00
                1.87
                0.00

                2.00

                1.66
                0.00
               -1.66
                0.00

                3.00

                1.30
                0.00
               -1.30
                0.00

                4.00

               -4.78
               10.03
               -5.24
                0.00
```

```
                5.00

                3.35
               14.16
                2.30
              -19.81

                6.00

                1.61
                0.00
                1.84
               -3.45

                7.00

                0.00
                5.24
               -5.24
                0.00

                8.00

                0.00
               12.17
                3.36
              -15.53

                9.00

                0.00
                0.00
                1.71
               -1.71
HOR.LOAD
                1.80
                0.80
JOINT MOM.
                1.00

                2.61
                0.00
               -2.61
                0.00

                2.00

                2.11
                0.00
               -2.11
                0.00

                3.00

                1.51
                0.00
               -1.51
                0.00
```

```
                4.00

                0.20
               -2.12
                1.92
                0.00

                5.00

                0.82
               -1.34
                1.97
               -1.46

                6.00

                0.29
                0.00
                1.33
               -1.61

                7.00

                0.00
               -0.51
                0.51
                0.00

                8.00

                0.00
               -0.43
                0.87
               -0.44

                9.00

                0.00
                0.00
                0.36
               -0.36
```

EXAMPLE 1b

```
VERT.LOAD

UN.            3.80
TR.C=          2.00
               4.12
====
UN.            2.80
TR.C=          0.00
               4.48
====
JOINT MOM.
               1.00

              -1.87
               0.00
               1.87
               0.00


               2.00

               1.66
               0.00
              -1.66
               0.00


               3.00

               1.30
               0.00
              -1.30
               0.00


               4.00

              -4.78
              10.03
              -5.24
               0.00


               5.00

               3.35
              14.16
               2.30
             -19.81


               6.00

               1.61
               0.00
               1.84
              -3.45
```

```
               7.00

               0.00
               5.25
              -5.24
               0.00


               8.00

               0.00
              12.18
               3.36
             -15.53


               9.00

               0.00
               0.00
               1.71
              -1.71

HOR.LOAD
               1.80
               0.80
JOINT MOM.
               1.00

               0.74
               0.00
              -0.73
               0.00


               2.00

               3.77
               0.00
              -3.77
               0.00


               3.00

               2.81
               0.00
              -2.81
               0.00


               4.00

              -4.58
               7.92
              -3.33
               0.00
```

```
               5.00

               4.17
              12.82
               4.28
             -21.27


               6.00

               1.89
               0.00
               3.17
              -5.06


               7.00

               0.00
               4.74
              -4.74
               0.00


               8.00

               0.00
              11.75
               4.23
             -15.97


               9.00

               0.00
               0.00
               2.07
              -2.07
```

EXAMPLE 1C

```
VERT.LOAD

UN.          3.80
TR.C=        2.00
             4.12
====
UN.          2.80
TR.C=        0.00
             4.48
====
NO MOV.!!
JOINT MOM.
             1.00

            -2.91
             0.00
             2.91
             0.00


             2.00

             0.76
             0.00
            -0.76
             0.00


             3.00

             0.65
             0.00
            -0.65
             0.00
```

```
  4.00

 -5.69
 11.51
 -5.82
  0.00


  5.00

  1.99
 15.20
  1.53
-18.71


  6.00

  1.04
  0.00
  1.30
 -2.34


  7.00

  0.00
  6.20
 -6.20
  0.00


  8.00

  0.00
 12.81
  1.98
-14.79


  9.00

  0.00
  0.00
  1.12
 -1.12
```

EXAMPLE 2

```
L
                1.000
                1.000
                1.000
                1.000
H
                1.000
                1.000
                1.000
                1.000

I BEAM
CON             1.000
I COL.
CON             1.000
VERT.LOAD

ML
MR
                1.000
               -1.000

                1.000
               -1.000

                1.000
               -1.000

                1.000
               -1.000

====
                1.000
               -1.000

                1.000
               -1.000

                1.000
               -1.000

                1.000
               -1.000

====
                1.000
               -1.000

                1.000
               -1.000

                1.000
               -1.000

                1.000
               -1.000
```

EXAMPLE 2 (continue)

```
----
          1.000
         -1.000

          1.000
         -1.000

          1.000
         -1.000

          1.000
         -1.000

====
JOINT MOM.
          1.000

         -0.148
          0.000
          0.148
          0.000


          2.000

          0.017
          0.000
         -0.017
          0.000


          3.000

          0.000
          0.000
         -0.000
          0.000


          4.000

         -0.017
          0.000
          0.017
          0.000


          5.000

          0.148
          0.000
         -0.148
          0.000


          6.000

         -0.423
          0.720
         -0.297
          0.000
```

```
          7.000

          0.047
          1.034
          0.034
         -1.115


          8.000

          0.000
          0.983
          0.000
         -0.983


          9.000

         -0.047
          1.115
         -0.034
         -1.034


         10.000

          0.423
          0.000
          0.297
         -0.720


         11.000

         -0.360
          0.761
         -0.401
          0.000


         12.000

          0.033
          1.025
          0.042
         -1.101


         13.000

         -0.000
          0.987
         -0.000
         -0.987


         14.000

         -0.033
          1.101
         -0.042
         -1.025
```

```
         15.000

          0.360
          0.000
          0.401
         -0.761


         16.000

         -0.449
          0.791
         -0.342
          0.000


         17.000

          0.052
          1.014
          0.027
         -1.094


         18.000

          0.000
          0.993
          0.000
         -0.993


         19.000

         -0.052
          1.094
         -0.027
         -1.014


         20.000

          0.449
          0.000
          0.342
         -0.791


         21.000

          0.000
          0.573
         -0.573
          0.000


         22.000

          0.000
          1.075
          0.082
         -1.157
```

EXAMPLE 2 (continue)

```
          23.000

           0.000
           0.962
          -0.000
          -0.962


          24.000

           0.000
           1.157
          -0.082
          -1.075


          25.000

           0.000
           0.000
           0.573
          -0.573
HOR.LOAD
           2.000
           2.000
           2.000
           1.000
JOINT MOM.
           1.000

           0.757
           0.000
          -0.757
           0.000


           2.000

           0.862
           0.000
          -0.862
           0.000


           3.000

           0.848
           0.000
          -0.848
           0.000


           4.000

           0.862
           0.000
          -0.862
           0.000
```

```
           5.000

           0.757
           0.000
          -0.757
           0.000


           6.000

           0.314
          -0.778
           0.464
           0.000


           7.000

           0.585
          -0.583
           0.672
          -0.673


           8.000

           0.550
          -0.597
           0.644
          -0.597


           9.000

           0.585
          -0.673
           0.672
          -0.583


          10.000

           0.314
           0.000
           0.464
          -0.778


          11.000

           0.168
          -0.566
           0.398
           0.000


          12.000

           0.324
          -0.448
           0.627
          -0.504
```

```
          13.000

           0.308
          -0.454
           0.601
          -0.454


          14.000

           0.324
          -0.504
           0.627
          -0.448


          15.000

           0.168
           0.000
           0.398
          -0.566


          16.000

           0.025
          -0.294
           0.269
           0.000


          17.000

           0.102
          -0.234
           0.394
          -0.262


          18.000

           0.093
          -0.237
           0.381
          -0.237


          19.000

           0.102
          -0.262
           0.394
          -0.234


          20.000

           0.025
           0.000
           0.269
          -0.294
```

EXAMPLE 2 (continue)

```
        21.000

         0.000
        -0.097
         0.097
         0.000

        22.000

         0.000
        -0.072
         0.155
        -0.083

        23.000

         0.000
        -0.074
         0.148
        -0.074

        24.000

         0.000
        -0.083
         0.155
        -0.072

        25.000

         0.000
         0.000
         0.097
        -0.097
```

EXAMPLE 3

```
FRAME

L
            10.00
             7.00
H
             5.00

I BEAM
CON          2.00
I COL.
             2.00
             0.50
             1.00

VERT.LOAD

UN.          1.00
UN.          1.00
====
```

```
JOINT MOM.
              1.00

HINGE
              0.00
              0.00
              0.00
              0.00

              2.00

HINGE
              0.00
              0.00
              0.00
              0.00

              3.00

              1.80
              0.00
             -1.80
              0.00

              4.00

              0.00
              4.98
             -4.98
              0.00

              5.00

              0.00
              7.31
              1.16
             -8.48

              6.00

              0.00
              0.00
              2.01
             -2.01

HOR.LOAD
              2.00
JOINT MOM.
              1.00

HINGE
              0.00
              0.00
              0.00
              0.00
```

```
              2.00

HINGE
              0.00
              0.00
              0.00
              0.00

              3.00

              3.91
              0.00
             -3.91
              0.00

              4.00

              0.00
             -1.86
              1.86
              0.00

              5.00

              0.00
             -0.88
              1.44
             -0.56

              6.00

              0.00
              0.00
              2.80
             -2.80
```

PROGRAM LISTING

```
0: csv;prt "FRAME";dsp "FLG";stp ;spc 1
1: ent "SPANS NR",A,"F1OORS NR.",B
2: 6(A+1)(B+1)→r3
3: ent "TOL",C
4: prt "L";r3+5→Y→r1
5: ent "L",rY;prt ·rY
6: if Y-r1<=A-2;Y+1→Y;jmp -1
7: prt "H"
8: ent "H",r(Y+1);prt r(Y+1)
9: if Y-r1<=A+2(B-1)-3;Y+2→Y;jmp -1
10: spc 1;6A+16→Y;r3+5→X→r1
11: prt "I BEAM"
12: if flg1;ent "IB",r0;prt "CON",r0
13: if flg1=0;ent "IB",r0;prt r0;if r0=0;1e-30→r0
14: r0/rX→rY
15: if X=r1+A-1;r1-1→X;Y+6→Y;if flg1=0;spc
16: X+1→X
17: if Y<=r3-8;Y+6→Y;jmp -4
18: 9→Y;X+A→X→r1
19: cfg 13;ent "HI?",rY;if flg13;1e30→rY
20: Y+6→Y;if Y<=6A+9;jmp -1
21: prt "I COL."
22: if flg2;ent "IC",r0;prt "CON",r0
23: if flg2=0;ent "IC",r0;prt r0
24: r0/rX→rY
25: if Y=6(.5(X-r1)+1)(A+1)+6A+9;X+2→X;if flg2=0;spc 1
26: if Y<=r3-3;Y+6→Y;jmp -3
27: 6A+16→Y;r3+5→X→r1;prt "VERT.LOAD";spc 1
28: if flg3;prt "ML","MR";jmp 8
29: cfg 13
30: if flg3;jmp 6
31: ent "P UN",r0;if flg13=0;prt "UN.",r0;rX→r2;jmp 2
32: ent "TR.C=",r2,"P",r0;prt "TR.C=",r2,r0
33: .5(rX-r2)→r2
34: r0(rX↑3-r2↑2(2rX-r2))/12/rX→r0;-r0→r2
35: jmp 2
36: ent "ML",r0,"MR",r2;prt r0,r2;spc 1
37: r(Y-2)+r0→r(Y-2);r(Y+3)+r2→r(Y+3);sfg 6
38: if X=r1+A-1;r1-1→X;Y+6→Y;prt "===="
39: X+1→X
40: if Y<=r3-8;Y+6→Y;cfg 13;jmp -11
41: 6→Y;0→r2
42: r(Y-1)+rY+r(Y+1)+r(Y+2)→r0
43: if abs(r0)>abs(r2);r0→r2
44: r(Y-2)+r(Y+3)+r(Y+4)→X
45: if Y<=6(A+1)B;X+r(Y+6A+9)→X
46: if Y>6(A+1)B;jmp 3
47: r0r(Y+6A+9)/X→r1
48: r(Y-1)-r1→r(Y-1);r(Y+6A+6)-.5r1→r(Y+6A+6)
49: r0r(Y+3)/X→r1
50: rY-r1→rY;if Y>6A+9;r(Y-6A-7)-.5r1→r(Y-6A-7)
51: r0r(Y-2)/X→r1
52: r(Y+1)-r1→r(Y+1);r(Y-4)-.5r1→r(Y-4);if Y=r3;jmp 4
53: r0r(Y+4)/X→r1
54: r(Y+2)-r1→r(Y+2);r(Y+7)-.5r1→r(Y+7)
55: Y+6→Y;jmp -13
```

```
56: if abs(r2)>C;gto 41
57: if flg7;prt "NO MOV.!!";gto 80
58: if flg4;jmp 9
59: r3+A+6→X→r1
60: if flg6=0;prt "HOR.LOAD";cfg 5;cfg 7;jmp 2
61: 0→rX;if flg6;jmp 2
62: ent "P→",rX;prt rX
63: if X-r1<=2(B-2);X+2→X;jmp -2
64: if X-r1#0;rX+r(X-2)→r(X-2)
65: rXr(X-1)→rX
66: if X-r1>0;X-2→X;jmp -2
67: 1→X;0→r2
68: 6X(A+1)+9→Y;0→r0→r1
69: r0+rY→r0
70: if Y<=6A+6X(A+1)+3;Y+6→Y;jmp -1
71: r1+r(Y-3)+r(Y-6A-10)→r1
72: if Y>6X(A+1)+9;Y-6→Y;jmp -1
73: r(r3+A+2X+4)-r1→r1
74: if abs(r1)>r2;abs(r1)→r2
75: r(Y-3)+.5r1rY/r0→r(Y-3)
76: r(Y-6A-10)+.5r1rY/r0→r(Y-6A-10)
77: if Y<=6A+6X(A+1)+3;Y+6→Y;jmp -2
78: if X<=B-1;X+1→X;jmp -10
79: sfg 4
80: cfg 6
81: if abs(r2)>.5(A+1)C;gto 41
82: prt "JOINT MOM.";5→Y
83: prt (Y+7)/6-1;spc 1
84: if Y<=6A+5;if r(Y+4)=0;prt "HINGE"
85: prt rY,r(Y+3),r(Y+1),r(Y+2)
86: if flg5;jmp 2
87: 0→rY→r(Y+3)→r(Y+1)→r(Y+2)
88: spc 2
89: if Y<=r3-5;Y+6→Y;jmp -6
90: cfg 4;dsp "NEXT LO.";stp
91: end
*8549
```

GEARING SPECIFICATIONS

This program calculates specifications for spiral-toothed and straight-toothed gears with any profile-offset. The formula are taken from DIN 3960 (August 1960)

Formula abbreviations:

Program	**DIN 3960**	**Dimension**	
T (1)	t_1	(–)	Teethnumber of the pinion
T (2)	t_2	(–)	Teethnumber of the gear
M (N)	m_n	(mm)	Pitch
Alpha (0)	α_o	(degrees)	Pressure angle
X (1)	x_1	(–)	Profile-offset of the pinion
X (2)	x_2	(–)	Profile-offset of the gear.
Beta	β_o	(degrees)	Bevel angle
H (KW)/M (N)	h_{kw}/m_n	(–)	Heigth of the cutting-tool related to the pitch. (DIN 3972)
M (S)	m_s	(mm)	Real module
A	a	(mm)	Axle base
D (O)	d_o	(mm)	Pitch diameter.
D (K)	d_k	(mm)	Outside diameter
H	h	(mm)	Tooth-heigth
T'	t'	(–)	Teethnumber for testing the base tangent length.
h	h	(mm)	Base tangent length
W (REQ.)	W_{req}	(mm)	The required wheel-width for testing the base tangent length.
			(The measurement-device needs 5mm. support at the tooth-profile)

HARDWARE REQUIREMENTS: 9825A (standard 6.844 bytes)

MEMORY: 1228 bytes.

ROMS:

PERIPHERALS:

STEP	DISPLAY	INSTRUCTIONS
1		ldf N. Load the file on which the program is stored.
2		fxd N, EXECUTE or flt N, EXECUTE (optional)
3		RUN
4	T (1)	t_1, enter teethnumber of the pinion CONTINUE
	T (2)	t_2, enter teethnumber of the gear CONTINUE
	M (N)	m_n, enter pitch CONTINUE
	ALPHA (O)	α_0, enter pressure angle CONTINUE
8		print-out from input of step 4.
9	X (1)	x1, enter profile-offset of the pinion CONTINUE
	X (2)	x2, enter profile-offset of the gear CONTINUE
	BETA (O)	β o, enter bevel angle CONTINUE
	H (KW)/M (N)	h_{kw}/m_n, enter height of the cutting-tool related to the pitch CONTINUE
10		print-out from input of step 9
11		print-out from m_s, a and for gear 1 and 2 the values d_o, d_k, h, t', L, W_{req}.
12		jump to step 4.

EXAMPLES

```
GEARING
SPECIFICATIONS
ACC.TO DIN3960
----------------
T(1)=      58.000
T(2)=      20.000
M(N)=       8.000
ALPHA(0)=  20.000
X(1)=       0.000
X(2)=       0.400
BETA(0)=   11.000
H(KW)/M(N)=
            1.250
----------------
M(S)=       8.150
A=        320.934
----------------

   GEAR 1

D(0)=     472.685
D(K)=     488.472
H=         17.894
T'=         7.000
L=        160.363
W(REQ.)=   33.672
----------------

    GEAR 2

D(0)=     162.995
D(K)=     185.183
H=         17.894
T'=         3.000
L=         63.594
W(REQ.)=   16.322
----------------
```

```
T(1)=      43.000
T(2)=      29.000
M(N)=       5.500
ALPHA(0)=  20.000
X(1)=       0.000
X(2)=       0.127
BETA(0)=    0.000
H(KW)/M(N)=
            1.250
----------------
M(S)=       5.500
A=        198.690
----------------

   GEAR 1

D(0)=     236.500
D(K)=     247.482
H=         12.366
T'=         5.000
L=         76.378
W(REQ.)=    5.000
----------------

    GEAR 2

D(0)=     159.500
D(K)=     171.879
H=         12.366
T'=         4.000
L=         59.540
W(REQ.)=    5.000
----------------
```

PROGRAM LISTING

```
0: spc ;prt "GEARING","SPECIFICATIONS","ACC.TO DIN3960";gsb 39
1: csv;cfg ;fxd 3
2: ent "T(1)=",r2,"T(2)=",r3,"M(N)=",r4,"ALPHA(0)=",r5
3: prt "T(1)=",r2,"T(2)=",r3,"M(N)=",r4,"ALPHA(0)=",r5
4: ent "X(1)=",r6,"X(2)=",r7,"BETA(0)=",r8,"H(KW)/M(N)=",r12
5: prt "X(1)=",r6,"X(2)=",r7,"BETA(0)=",r8,"H(KW)/M(N)=",r12
6: 2(r6+r7)tan(r5)/(r2+r3)+(tan(r5)/cos(r8)→r9)→Z
7: Z-π(atn(r9)→r9)/180→Z
8: gsb 21
9: Y→C
10: A+4→B
11: if C=0;gto 22
12: B→X
13: gsb 21
14: if 0<=(Y→r0)C;r0→C;B→A;gto -4
15: B→r1
16: (A+r1)/2→X
17: gsb 21
18: if abs(Y)<=1e-10;gto 22
19: if 0>CY;X→r1;gto -3
20: X→A;Y→C;gto -4
21: tan(X)-πX/180-Z→Y;ret
22: gsb 39
23: prt "M(S)=",r4/cos(r8)→r10
24: prt "A=",r10(r2+r3)cos(r9)/2cos(X)→r11
25: gsb 39
26: spc ;prt "    GEAR 1";spc
27: prt "D(0)=",r2r10→A
28: prt "D(K)=",2(r11+r4-r7r4)-r3r10→C
29: prt "H=",(C-A-2r6r4+2r12r4)/2
30: atn(tan(r8)cos(r9))→X
31: tan(acs(r2cos(r9)/(r2+2r6cos(r8))))/cos(X)↑2-2r6tan(r5)/r2→Z
32: prt "T'=",int(r2(Z-(tan(r9)-πr9/180→Y))/π+1)→Z
33: prt "L=",r4cos(r5)((Z-.5)π+Yr2)+2r6r4sin(r5)→Z
34: prt "W(REQ.)=",Zsin(X)+5cos(X)
35: gsb 39
36: if flg1;spc 8;gto 1
37: sfg 1;r2→Z;r3→r2;Z→r3;r6→Z;r7→r6;Z→r7
38: spc ;prt "     GEAR 2";spc ;gto 27
39: prt "----------------";ret
40: end
*711
```

MEMORY ALLOCATION

VARIABLES	
A	a, d_{o1}, d_{o2}
B	b
C	f (a), d_{k1}, d_{k2}
X	α sb, $\alpha\,\beta$g
Y	div., ev α_{so}
Z	div.,
r0	f (b)
r1	b′
r2	t1, t2
r3	t2, t1
r4	m_n
r5	α o
r6	x_1, x_2
r7	x_2′ x_1
r8	βo
r9	tgαso,αso
r10	m_s
r11	a
r12	h_{kw}/m_n

FLAGS	
0	used
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

MODEL 9825A

SKEW RAY TRACE THROUGH A CENTERED SPHERICAL SURFACE

Introduction

The usual simple ray trace programmes are normally limited to dealing with meridional rays only.

The following programme, based on ref 1 and an algol procedure, developed by the author for use in a spot diagram programme, can accept any direction of the incident ray and any point of intersection with a spherical surface, either reflective or refractive.

The meridonal ray trace then becomes merely a special case of the skew ray trace.

Co-ordinate System

The right handed coordinate system used, is shown in Fig, 1. The system is centered at the pole of the surface being considered as shown.

The symbols are:—

x_{in}, y_{in}, z_{in}	Coordinates of the intersection of the ray with the preceeding surface, referred to the pole of that surface.
$l_{in}, m_{in}, n_{in},$	Directional cosines of the ray, emerging from the preceeding surface.
N	Refractive index of the medium preceeding the surface being considered.
N′	Refractive index of the medium following the surface being considered.
t	Distance, from the preceeding surface pole, of the "present" surface.
c	Curvature, i.e. the reciprocal of the radius. of the surface being considered.
x_{out} y_{out} z_{out}	Coordinates of the point of intersection of the ray with the surface being considered.
l_{out} m_{out} n_{out}	Directional Cosines of the ray emerging from the surface being considered.

Sign convention

The positive directions of x,y and z are as shown in Fig. 1.

The signs of the indices can be positive or negative according to the following rules:

1. If the ray has undergone refraction at the surface, then the sign of N' is the same as the sign of N.
2. If the ray has been reflected and is being traced through an "unfolded" optical system, then the sign of N' must be opposite to the sign of n. Furthermore, $|N| = |N'|$. (see example).
3. The sign for curvature is positive if the centre of curvature of the surface in question lies to the right of the pole.

Programme Operation

The calculation is entered with the values of x_{in}, y_{in}, z_{in}, l_{in}, m_{in} and n_{in} referred to either a plane surface of zero refracting power placed in a convenient point (usually the entrance pupil) or the first surface. The programme will then ask for the values of c (zero for plane surfaces), t, N and N'.

On completing the computation, the values of x_{out}, y_{out}, z_{out}, l_{out}, m_{out} and n_{out} will be printed; the display will show "TRANSFER?" at this point. If it is desired to proceed to the next surface, press RUN PROGRAM and the values of x_{out} etc. will be transferred to x_{in} etc. New values of t, N and N' will have to be entered.

If it is desired to terminate the computation enter any number on TRANSFER?

Example

A meridional ray was traced through the edge of a Mangin mirror to determine its point of intersection with the plane of the best focus (see Fig. 2a).

The preliminary drawing of the system, as prepared for computation, is shown in Fig. 2b. According to this drawing, these are four surfaces to be traversed including the plane at the end.

The input ray has been chosen to lie in the yz plane, to be parallel to the y axis (1 = 0, m = 1, n = 0) and to be 3.5 units distant form it.

The various numbers in Fig. 2b (except the effective index values) are calculated by the repeated use of the programme, once for each surface, giving as the final desired result, the offset of the ray intercept at the target plane (0.000659).

The actual print out is included.

References

1 Handbook of Military Infra Red Technology
W.L. Wolfe 1965
Office of Naval Research
Department of the Navy.

HARDWARE REQUIREMENTS: 9825A (standard 6.844 bytes)

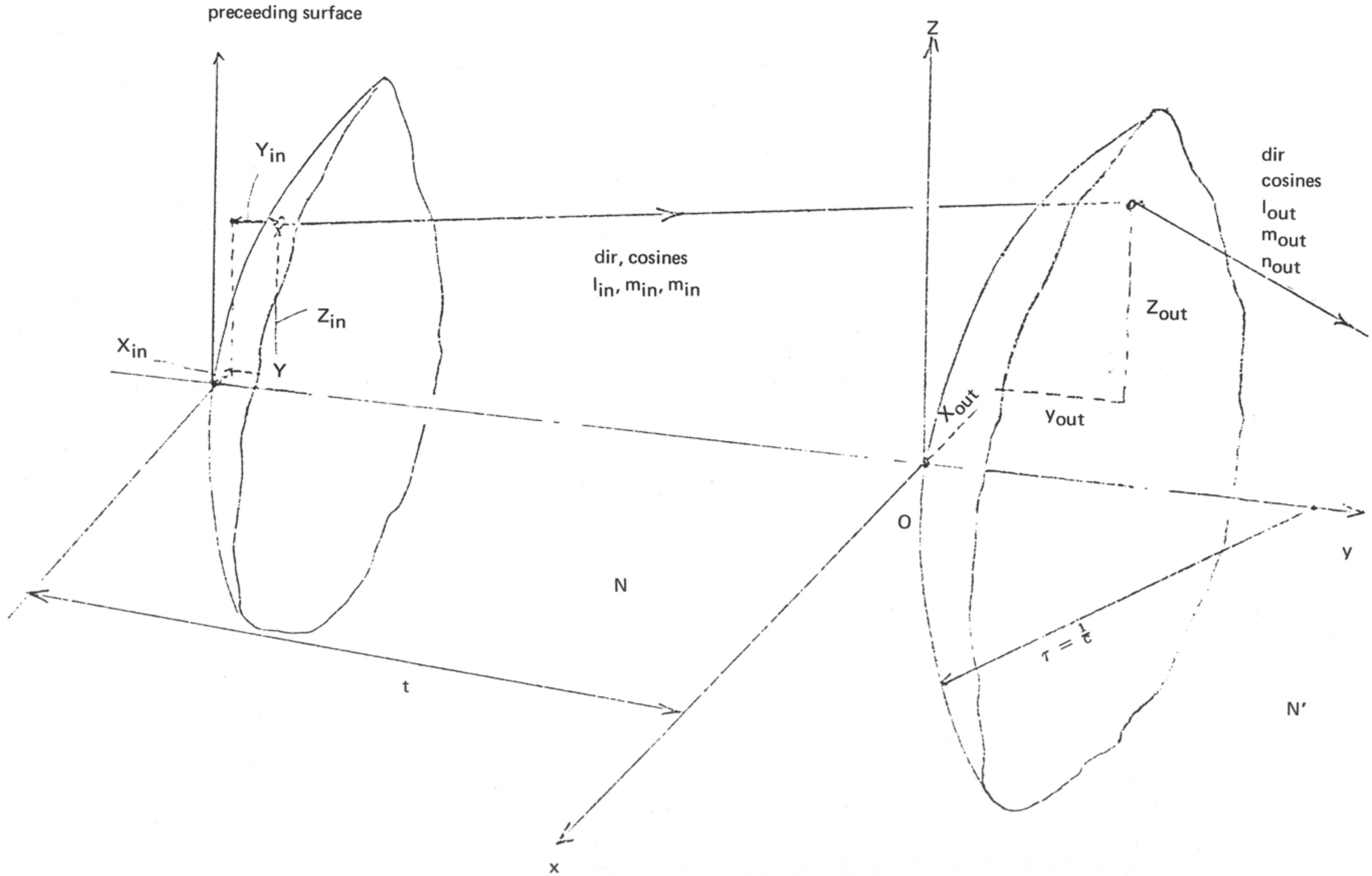
MEMORY: 1100 bytes.

ROMS:

PERIPHERALS:

STEP	DISPLAY	INSTRUCTIONS
1		ldf. Load the file on which the program is stored.
2		RUN
3	X IN	enter X, CONTINUE
4	Y IN	enter Y, CONTINUE
5	Z IN	enter Z, CONTINUE
6	L IN	enter L, CONTINUE
7	M IN	enter M, CONTINUE
8	N IN	enter N, CONTINUE
9	CURVATURE	enter the value of surface curvature, CONTINUE
10	DST. TO PREC. SURF.	enter the distance (always positiv), CONTINUE
11	PREC.	enter refractive index preceding surface, CONTINUE
12	FOLL. INDEX	enter refractive index preceding following surface CONTINUE
13		computation
14	TRANSFER	If this was the last surface, enter any number and CONTINUE, to terminate.
		Otherwise just CONTINUE to transfer the values of X, Y, Z, L, M, N just calculated
		to the input registers and return control to step 9.

FIG 1



SKEW RAY GEOMETRY

MANGIN MIRROR

Fig 2a

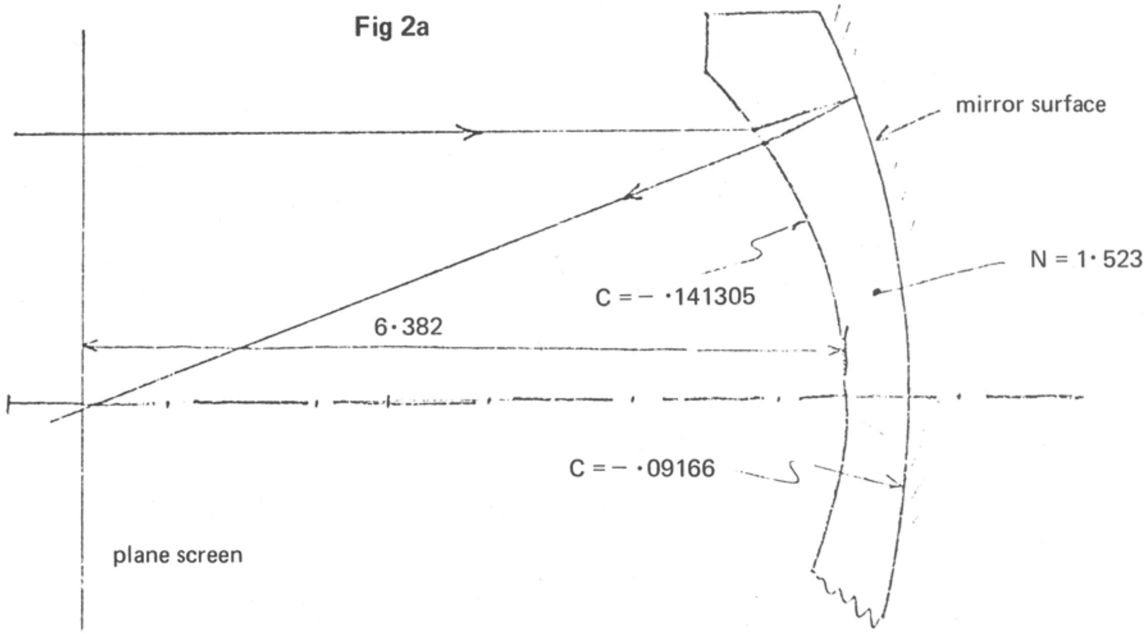
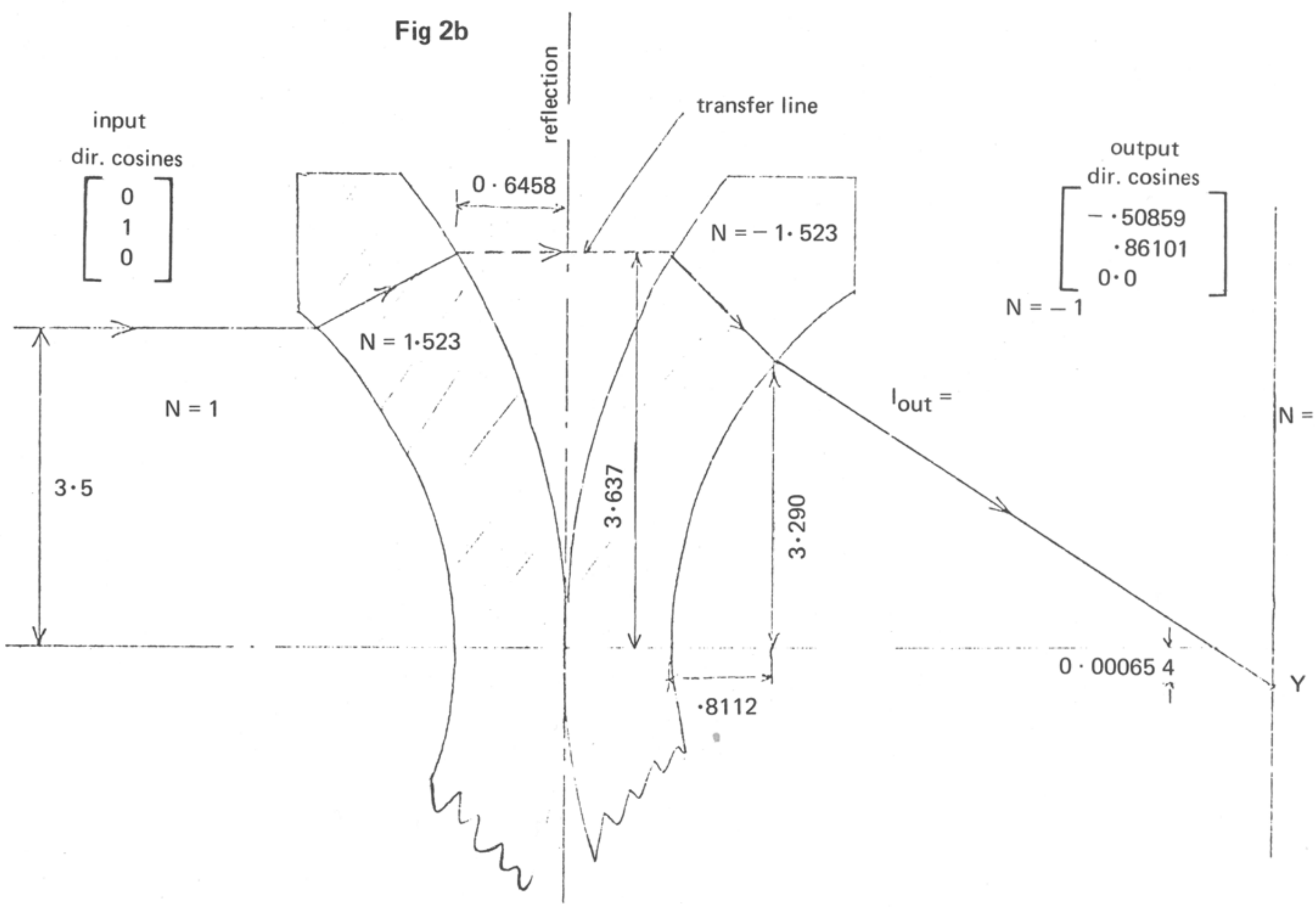


Fig 2b



```
INPUT COORDS.=
        3.5000e 00
        0.0000e 00
        0.0000e 00

INPT.DIR.COSNS.=
        0.0000e 00
        1.0000e 00
        0.0000e 00

CURVATURE=
      -1.4131e-01
PREC.SURF.DIST.=
        0.0000e 00
PRE-INDEX=
        1.0000e 00
POST-INDEX=
        1.5230e 00

OUTPUT COORDS.=
        3.5000e 00
      -9.2613e-01
        0.0000e 00

OUTP.DIR.COSNS.=
        1.8554e-01
        9.8264e-01
        0.0000e 00


NEXT SURFACE


INPUT COORDS.=
        3.5000e 00
      -9.2613e-01
        0.0000e 00

INPT.DIR.COSNS.=
        1.8554e-01
        9.8264e-01
        0.0000e 00

CURVATURE=
      -9.4660e-02
PREC.SURF.DIST.=
        4.4610e-01
PRE-INDEX=
        1.5230e 00
POST-INDEX=
      -1.5230e 00

OUTPUT COORDS.=
        3.6371e 00
        6.4586e-01
        0.0000e 00

OUTP.DIR.COSNS.=
      -4.9371e-01
        8.6963e-01
        0.0000e 00
```

```
NEXT SURFACE


INPUT COORDS.=
        3.6371e 00
        6.4586e-01
        0.0000e 00

INPT.DIR.COSNS.=
      -4.9371e-01
        8.6963e-01
        0.0000e 00

CURVATURE=
        1.4131e-01
PREC.SURF.DIST.=
        4.4610e-01
PRE-INDEX=
      -1.5230e 00
POST-INDEX=
      -1.0000e 00

OUTPUT COORDS.=
        3.2900e 00
        8.1127e-01
        0.0000e 00

OUTP.DIR.COSNS.=
      -5.0858e-01
        8.6102e-01
        0.0000e 00


NEXT SURFACE


INPUT COORDS.=
        3.2900e 00
        8.1127e-01
        0.0000e 00

INPT.DIR.COSNS.=
      -5.0858e-01
        8.6102e-01
        0.0000e 00

CURVATURE=
        0.0000e 00
PREC.SURF.DIST.=
        6.3820e 00
PRE-INDEX=
      -1.0000e 00
POST-INDEX=
      -1.0000e 00

OUTPUT COORDS.=
      -4.9775e-04
        0.0000e 00
        0.0000e 00

OUTP.DIR.COSNS.=
      -5.0858e-01
        8.6102e-01
        0.0000e 00
```

PROGRAM LISTING

```
0: cfg 1;ent "X IN",r1,"Y IN",r2,"Z IN",r3
1: "B":prt "INPUT COORDS.=",r1,r2,r3;spc ;jmp 1+flg1
2: ent "L IN",r4,"M IN",r5,"N IN",r6
3: prt "INPT.DIR.COSNS.=",r4,r5,r6;spc
4: ent "CURVATURE",C,"DST.TO PRC.SURF.",r0,"PREC.INDEX",A,"FOLL.INDEX",B
5: prt "CURVATURE=",C,"PREC.SURF.DIST.=",r0,"PRE-INDEX=",A
6: prt "POST-INDEX=",B;spc
7: if C=0;(r0-r2)/r5→X;gto "C"
8: r1r4+r3r6+r5(r2-r0-1/C)→r7
9: r1r1+r2r2+r3r3-2r2(r0+1/C)+r0(r0+2/C)→r8
10: √(r7r7-r8)→r9
11: -r7+r9→X
12: if abs(1/C)>abs(Xr5+r2-r0);X→r17;gto +2
13: -r7-r9→X
14: "C":Xr4+r1→r11
15: Xr5+r2-r0→r12
16: Xr6+r3→r13
17: A/B→Y
18: r5-C(r11r4+r12r5+r13r6)→r7
19: √(1-YY(1-r7r7))→r8
20: r8-Yr7→r9
21: (Y>0)-(0>Y)→X
22: X(Yr4-Cr9r11)→r14
23: Yr5-r9(Cr12-1)→r15
24: X(Yr6-Cr9r13)→r16
25: Xr12→r12
26: prt "OUTPUT COORDS.="
27: prt r11,r12,r13;spc
28: prt "OUTP.DIR.COSNS.="
29: prt r14,r15,r16;spc 3
30: cfg 13;ent "TRANSFER?",X
31: if flg13=0;gto "A"
32: prt "NEXT SURFACE";spc 2;sfg 1
33: 1→X
34: r(10+X)→rX;X+1→X;jmp X>6
35: gto "B"
36: "A":spc 5
37: end
*9213
```

MEMORY ALLOCATION

VARIABLES							
A	PREC. INDEX						
B	FOLL. INDEX						
C	CURVATURE						
X	temp. storage						
Y	–"– –"–						
r1	X input						
r2	Y input						
r3	Z input						
r4	L input						
r5	M input						
r6	N input						
r7	temp. storage						
r8	" "						
r9	" "						
r10	available						
r11	X out						
r12	Y out						
r13	Z out						
r14	L out						
r15	M out						
r16	N out						
						FLAGS	
						0	
						1	
						2	
						3	
						4	
						5	
						6	
						7	
						8	
						9	
						10	
						11	
						12	
						13	
						14	
						15	

MODEL 9825A

LOAN AMORTISATION SCHEDULE

The required input for this program are the principle, the interest rate, payments/year and total number of payments. The output includes the payment number, interest at a period, interest to date, principle at a period, the principle to date and the principle balance.

A fully formatted output on the 9871A Printer is optional.

MEMORY: 3144 bytes

ROMS: 98212A Gen I/O-9862 Plotter or 98213A Gen I/O-Ext. I/O or 98214A Gen I/O-Ext.-I/O-9862 Plotter (optional)

PERIPHERALS: 9871A Printer (optional)

STEP	DISPLAY	INSTRUCTIONS
1		ldf N. Load the file on which the program is stored.
2		sfg 10, EXECUTE if 9871A Printer Output is not desired.
3	PRINCIPLE	RUN
3	PRINCIPLE	Enter Principle, CONTINUE
4	INT. RATE	Enter Interest Rate, CONTINUE
5	PAYMENTS/YEAR	Enter number of Payments per year. CONTINUE
		If only total number of payments to be made is known merely press CONTINUE.
6	MONTH 1st. PAYMENT	Enter month of the year in which first payment is done (e.g. 6 = June) CONTINUE
7	PRESENT YEAR	Enter Year CONTINUE
8	NO. OF PAYMENTS	Enter total number of Payments if no data entry was make at step 5. Then CONTINUE.
		Merely press CONTINUE if data entry at step 5.
9	PAYMENT	Enter Payment. CONTINUE
		Print-Out: Payment & No. of Periods
10	NEW PAYMENT?	If Yes: enter new Payment CONTINUE. Otherwise just CONTINUE.
11	CHANGE PAYMENT	If Yes: enter new payment CONTINUE. Otherwise just CONTINUE.
12		Print-out of Loan amortisation schedule.

EXAMPLE

LOAN
AMORTISATION
SCHEDULE

PRINCIPLE=
1000.00
INTEREST RATE =
4.00
PAYMENTS/YEAR=
12.00
PAYMENT= 80.00
NO.OF PAYMENTS=
12.79

THE INFORMATION
BELOW IS IN THE
FOLLOWING ORDER

PAYMENT NO.
INT.THIS PER.
INTEREST TO DATE
PRIN.THIS PERIOD
PRIN. TO DATE
PRIN.BALANCE

1.00
3.33
3.33
76.67
76.67
923.33

2.00
3.08
6.41
76.92
153.59
846.41

3.00
2.82
9.23
77.18
230.77
769.23

4.00
2.56
11.79
77.44
308.21
691.79

5.00
2.31
14.10
77.69
385.90
614.10

6.00
2.05
16.15
77.95
463.85
536.15

7.00
1.79
17.94
78.21
542.06
457.94

8.00
1.53
19.47
78.47
620.53
379.47

9.00
1.26
20.73
78.74
699.27
300.73

10.00
1.00
21.73
79.00
778.27
221.73

11.00
0.74
22.47
79.26
857.53
142.47

12.00
0.47
22.94
79.53
937.06
62.94

13.00
0.21
23.15
62.94
1000.00
0.00

LAST PAYMENT=
63.15

LOAN AMORTIZATION SCHEDULE

PRINCIPLE AMOUNT	1000.00
INTEREST RATE	4.00
PAYMENTS PER YEAR	12.00
PAYMENT AMOUNT	80.00
NO. OF PAYMENTS	12.79

PAYMENT NUMBER	PAYMENT DATE	INTEREST THIS PERIOD	INTEREST TO DATE	PRINCIPLE THIS PERIOD	PRINCIPLE TO DATE	PRINCIPLE BALANCE
1	JAN 1976	3.33	3.33	76.67	76.67	923.33
2	FEB 1976	3.08	6.41	76.92	153.59	846.41
3	MAR 1976	2.82	9.23	77.18	230.77	769.23
4	MAY 1976	2.56	11.79	77.44	308.21	691.79
5	MAY 1976	2.31	14.10	77.69	385.90	614.10
6	JUN 1976	2.05	16.15	77.95	463.85	536.15
7	JUL 1976	1.79	17.94	78.21	542.06	457.94
8	AUG 1976	1.53	19.47	78.47	620.53	379.47
9	SEP 1976	1.26	20.73	78.74	699.27	300.73
10	OCT 1976	1.00	21.73	79.00	778.27	221.73
11	NOV 1976	0.74	22.47	79.26	857.53	142.47
12	DEC 1976	0.47	22.94	79.53	937.06	62.94
13	JAN 0	0.21	23.15	62.94	1000.00	0.00

LAST PAYMENT 63.15

LOAN AMORTIZATION SCHEDULE

PRINCIPLE AMOUNT	1000.00
INTEREST RATE	4.00
PAYMENTS PER YEAR	12.00
PAYMENT AMOUNT	68.46
NO. OF PAYMENTS	15.00

PAYMENT NUMBER	PAYMENT DATE	INTEREST THIS PERIOD	INTEREST TO DATE	PRINCIPLE THIS PERIOD	PRINCIPLE TO DATE	PRINCIPLE BALANCE
1	JAN 1976	3.33	3.33	65.13	65.13	934.87
2	FEB 1976	3.12	6.45	65.34	130.47	869.53
3	MAR 1976	2.90	9.35	65.56	196.03	803.97
4	MAY 1976	2.68	12.03	65.78	261.81	738.19
5	MAY 1976	2.46	14.49	66.00	327.81	672.19
6	JUN 1976	2.24	16.73	66.22	394.03	605.97
7	JUL 1976	2.02	18.75	66.44	460.47	539.53
8	AUG 1976	1.80	20.55	66.66	527.13	472.87
9	SEP 1976	1.58	22.13	66.88	594.01	405.99
10	OCT 1976	1.35	23.48	67.11	661.12	338.88
11	NOV 1976	1.13	24.61	67.33	728.45	271.55
12	DEC 1976	0.91	25.52	67.55	796.00	204.00
13	JAN 0	0.68	26.20	67.78	863.78	136.22
14	FEB 0	0.45	26.65	68.01	931.79	68.21
15	MAR 0	0.23	26.88	68.21	1000.00	0.00

LAST PAYMENT 68.44

PROGRAM LISTING

```
0: 1→r0;fxd 2;ent "PRINCIPLE?",A
1: ent "INTEREST RATE?",B
2: 12→Z;ent "PAYMENTS/YEAR?",Z;B/100Z→B;12/Z→r23
3: 1→r20;1976→r21;if flg10;jmp 2
4: ent "MONTH-1st PAYMT",r20;ent "PRESENT YEAR",r21
5: cfg 13;-1→C;ent "NO.OF PAYMENTS?",C;if flg13=0;jmp 6
6: cfg 13;ent "PAYMENT?",X
7: prt "PAYMENT=",X
8: prt "NO.OF PERIODS=",ln(X/(X-AB))/ln(1+B)→C
9: ent "NEW PAYMENT?",X;if flg13=0;jmp -2
10: gto "H"
11: if C=0;ent "PAYMENT?",X;jmp 8
12: 1→X→Y
13: BY+Y→Y
14: if (X+1→X)<=C;jmp -1
15: int(ABY/(Y-1)*100+.5)/100→X
16: prt "PAYMENT=",X
17: ent "CHANGE PAYMENT?",X;if flg13=0;prt "NEW PAYMENT IS",X
18: ln(X/(X-AB))/ln(1+B)→C;jmp 4
19: cfg 13;ent "PRDS.THIS PAYMT?",rr0;if flg13;jmp 3
20: ent "NEXT PAYMENT?",r(r0+1);if (r0+2→r0)>3;r(r0-2)+r(r0-4)→r(r0-2)
21: jmp -2
22: "H":if C=0;r1→C
23: if flg10=0;gto "C"
24: spc 5;prt "        LOAN","  AMORTISATION","     SCHEDULE";spc 3
25: prt "PRINCIPLE=",A,"INTEREST RATE =",100BZ
26: prt "PAYMENTS/YEAR=",Z,"PAYMENT=",X,"NO.OF PAYMENTS=",C
27: spc 2;prt "THE INFORMATION","BELOW IS IN THE ","FOLLOWING ORDER"
28: spc 2;prt "PAYMENT NO.","INT.THIS PER.","INTEREST TO DATE"
29: prt "PRIN.THIS PERIOD","PRIN. TO DATE","PRIN.BALANCE"
30: prt "----------------";gto "D"
31: "C":fmt /,40x,"LOAN AMORTIZATION SCHEDULE";wrt 6
32: fmt 3/,20x,"PRINCIPLE AMOUNT",5x,f8.2;wrt 6,A
33: fmt 20x,"INTEREST RATE",8x,f8.2;wrt 6,100BZ
34: fmt 20x,"PAYMENTS PER YEAR",6x,f6.2,2/;wrt 6,Z
35: fmt 20x,"PAYMENT AMOUNT",7x,f8.2;wrt 6,X
36: fmt 20x,"NO. OF PAYMENTS",6x,f8.2,2/;wrt 6,C
37: fmt 5x,"PAYMENT",7x,"PAYMENT",7x,"INTEREST",7x,"INTEREST",z;wrt 6
38: fmt 6x,"PRINCIPLE",5x,"PRINCIPLE",5x,z;wrt 6
39: fmt "PRINCIPLE";wrt 6
40: fmt 5x,"NUMBER",9x,"DATE",8x,"THIS PERIOD",5x,"TO DATE",6x,z;wrt 6
41: fmt "THIS PERIOD",5x,"TO DATE",7x,"BALANCE";wrt 6
42: fmt 5x,"-------",7x,"-------",6x,"-----------",5x,"-------",5x,z;wrt 6
43: fmt "-----------",4x,"-----------",5x,"---------",/;wrt 6
44: "D":0→C→Y→Z;cfg 12;r0→r19
45: "B":if r0=1;gto "G"
46: if r(r19-r0+1)>Y;gto "G"
47: r(r19-r0+2)→X;r0-2→r0;if r0>1;r(r19-r0+1)-r(r19-r0-1)→r24;jmp 2
48: if r0=1;ln(X/(X-AB))/ln(1+B)→r24
49: if flg10;spc 1;prt "PAYMENT=",X,"NO"
50: fmt /,20x,"PAYMENT AMOUNT",7x,f8.2;wrt 6,X
51: fmt 20x,"NO.OF PAYMENTS",6x,f8.2,/;wrt 6,r24
52: "G":Y+1→Y;int(100AB+.5)/100→r22;Z+r22→Z;C+X-r22→C;A+r22-X→A
53: if flg10=0;gto "E"
54: spc 1;prt Y,r22,Z,X-r22,C,A;gto "F"
55: "E":fmt 1x,f8.0,9x,z;wrt 6,Y
```

```
56: if r20<=1;fmt "JAN",z;wrt 6;gto "A"
57: if r20<=2;fmt "FEB",z;wrt 6;gto "A"
58: if r20<=3;fmt "MAR",z;wrt 6;gto "A"
59: if r20<=3;fmt "MAR",z;wrt 6;gto "A"
60: if r20<=5;fmt "MAY",z;wrt 6,Z;gto "A"
61: if r20<=6;fmt "JUN",z;wrt 6;gto "A"
62: if r20<=7;fmt "JUL",z;wrt 6;gto "A"
63: if r20<=8;fmt "AUG",z;wrt 6;gto "A"
64: if r20<=9;fmt "SEP",z;wrt 6;gto "A"
65: if r20<=10;fmt "OCT",z;wrt 6;gto "A"
66: if r20<=11;fmt "NOV",z;wrt 6;gto "A"
67: if r20<=12;fmt "DEC",z;wrt 6;gto "A"
68: "A":fmt 1x,f4.0,z;wrt 6,r21
69: fmt 6x,f8.2,7x,f8.2,7x,z;wrt 6,r22,Z
70: if 0>X-r22;fmt z;wrt 6
71: fmt f8.2,6x,f8.2,7x,f8.2;wrt 6,X-r22,C,A
72: "F":if (r20+r23→r20)>12;r20-12→r20;r21+1→r21
73: if flg12;gto "I"
74: if A=0;gto "I"
75: if X>A;A+int(100AB+.5)/100→X;sfg 12
76: gto "B"
77: "I":if flg10=0;fmt /,20x,"LAST PAYMENT",9x,f8.2,9/;wrt 6,X;jmp 2
78: spc 1;prt "LAST PAYMENT=";prt X;spc 9
79: end
*19839
```

HEWLETT PACKARD

5951–0782

Printed in Federal Republic of Germany 3/77