EUR 4499 e

COMMISSION OF THE EUROPEAN COMMUNITIES

THE DESIGNING OF SPECIAL PURPOSE SLIDERULES AND THE RELATED CODES "ACCESS" AND "COOLER"

by

I. DE WOLDE

1970

Joint Nuclear Research Center
Ispra Establishment - Italy
Scientific Data Processing Center (CETIS)

LEGAL NOTICE

This report is on sale at the addresses listed on cover page 4

at the price of FF 7.75 FB 70.— DM 5.10 Lit. 870.— Fl. 5.10

When ordering, please quote the EUR number and the title, which are indicated on the cover of each report.

Printed by L. Vanmelle - Ghent
Luxembourg, September 1970

EUR 4499 e

THE DESIGNING OF SPECIAL PURPOSE SLIDERULES AND THE RELATED CODES "ACCESS" AND "COOLER",
by I. DE WOLDE

Commission of the European Communities
Joint Nuclear Research Center — Ispra Establishment (Italy)
Scientific Data Processing Center (CETIS)
Luxembourg, September 1970 — 52 Pages — 5 Figures — FB 70,—

One may distinguish a class of technical and scientific computation problems for which a problem-oriented sliderule might be very useful.

Several sliderules have already been developed by Cetis-Euratom, together with some tools to facilitate the designing.

This report describes the obtained experiences together with two computer programmes i.c. "ACCESS" and "COOLER".

"ACCESS" converts the numerical descriptions of sliderules into the actual drawings. "COOLER" is a programme which designs and draws circular sliderules for the conversion of leak-rates but by programming other relations it may be used for other problems also.

EUR 4499 e

THE DESIGNING OF SPECIAL PURPOSE SLIDERULES AND THE RELATED CODES "ACCESS" AND "COOLER",
by I. DE WOLDE

Commission of the European Communities
Joint Nuclear Research Center — Ispra Establishment (Italy)
Scientific Data Processing Center (CETIS)
Luxembourg, September 1970 — 52 Pages — 5 Figures — FB 70,—

One may distinguish a class of technical and scientific computation problems for which a problem-oriented sliderule might be very useful.

Several sliderules have already been developed by Cetis-Euratom, together with some tools to facilitate the designing.

This report describes the obtained experiences together with two computer programmes i.c. "ACCESS" and "COOLER".

"ACCESS" converts the numerical descriptions of sliderules into the actual drawings. "COOLER" is a programme which designs and draws circular sliderules for the conversion of leak-rates but by programming other relations it may be used for other problems also.

EUR 4499 e

COMMISSION OF THE EUROPEAN COMMUNITIES

THE DESIGNING OF SPECIAL PURPOSE SLIDERULES AND THE RELATED CODES "ACCESS" AND "COOLER"

by

I. DE WOLDE

1970

Joint Nuclear Research Center
Ispra Establishment - Italy
Scientific Data Processing Center (CETIS)

ABSTRACT

One may distinguish a class of technical and scientific computation problems for which a problem-oriented sliderule might be very useful.
Several sliderules have already been developed by Cetis-Euratom, together with some tools to facilitate the designing.
This report describes the obtained experiences together with two computer programmes i.c. " ACCESS " and " COOLER ".
" ACCESS " converts the numerical descriptions of sliderules into the actual drawings. " COOLER " is a programme which designs and draws circular sliderules for the conversion of leak-rates but by programming other relations it may be used for other problems also.

KEYWORDS

COMPUTERS
TOOLS
DESIGN
PROGRAMMING
LEAKS
FORTRAN

CONTENT:

INTRODUCTION *)

It has been experienced that there exists a class of technical and scientific computation problems which might be solved conveniently by a specially designed sliderule.

This class is characterized by:

1. not too many input parameters,
2. few output data,
3. rather complicated calculation process, specially including algebraic or numerical functions,
4. frequent executions,
5. instantaneous results requested.

Thus characterized problems may be handled by special computer produced tables, by nomograms or by special purpose sliderules. For several problems the latter solution has been applied by CETIS-EURATOM. Also some tools have been developed to facilitate the designing of special purpose sliderules. This report describes these aids and the obtained experiences.

Two computer programmes are mentioned:
The code "ACCESS" converts numerical descriptions of sliderules into the actual drawings with the aid of a CALCOMP-plotter. This programme can be applied as an independent programme or as a part of larger code.

The code "COOLER" designs and draws circular sliderules for leak-rate conversion calculations. "COOLER" might be used more generally by removing the two subroutines where the actual relations are calculated and substituting them by other FORTRAN programmed expressions. The problem of the leak-rate conversions has been described somewhat extensively as an illustration for the entire report.

No distinction has been made as far as concerns the basic shape

*) Manuscript received on 21 April 1970

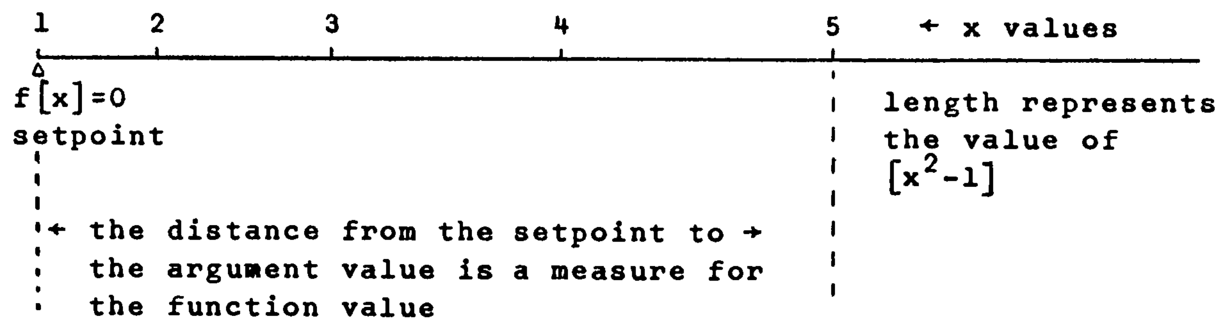
of the sliderules, circular or rectangular. The most important difference is that a circular sliderule can easily be provided with more moving parts.

THE SLIDERULE

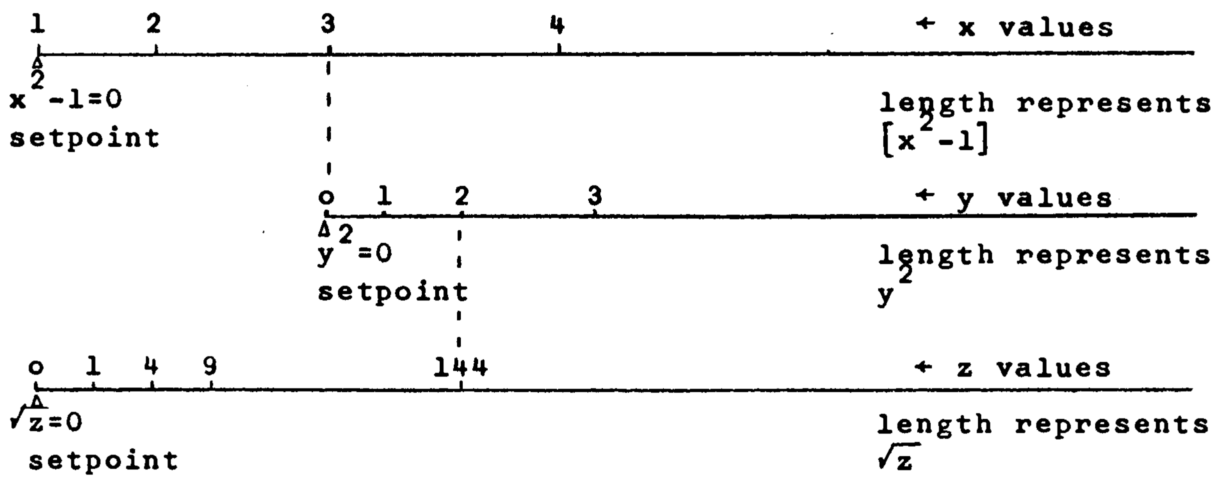
A sliderule may be considered as a primitive analog machine. The two reversible basic principles are:

1. function value representation for a given argument only for strict monotonic functions,
2. addition of function values.

The first principle is illustrated by a scale for the function $f[x] = x^2-1$ with $x \geq 1$ to have a monotonic branch of the function



The addition of two function values is performed physically as is illustrated for the relation $\sqrt{z} = x^2-1 + y^2$:



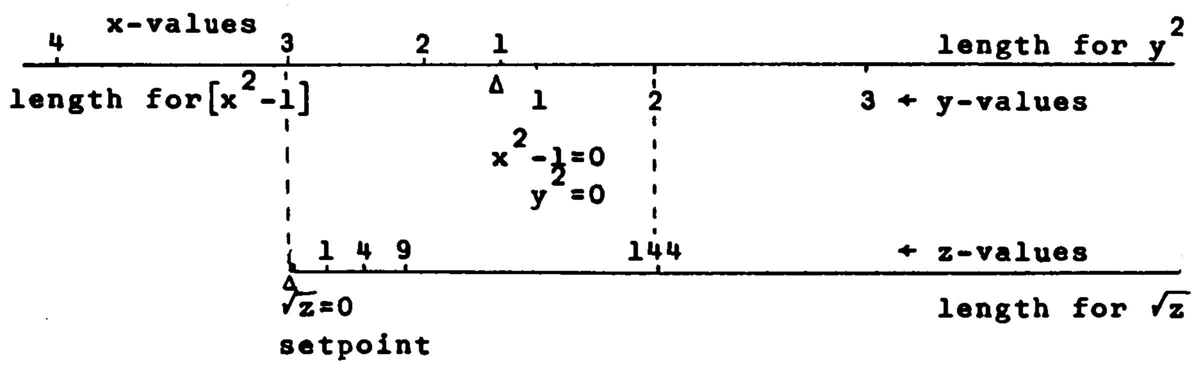
If now the latter example is executed as a traditional sliderule, the x-scale and the z-scale will be engraved on the body of the rule and the y-scale on the central slide (cs.). Furthermore, a windowslide (ws.) is necessary.

The elaborations for the calculation of z at given x and y will be:

1. put the ws. on the x-value,
2. move the cs. until the setpoint is under the marker,
3. move the ws. to the y-value,
4. read the z-value.

A certain simplification of the tool is possible by combinations of scales which leads ultimately to sliderule applications for more complex functions.

In the given example, the positive direction of each scale points to the right. However, the procedure will be simplified if the positive direction of the x-scale points to the left as is shown in the next example:



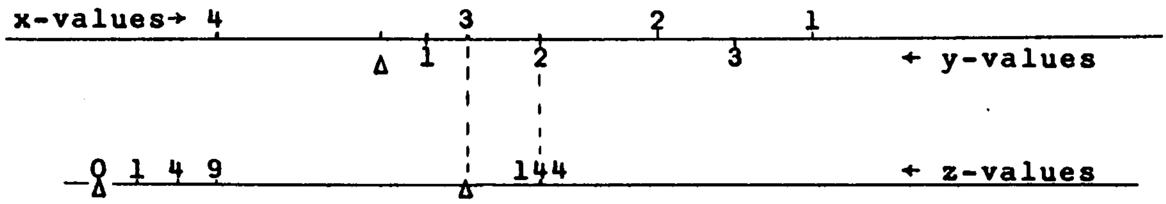
The necessary elaborations to calculate z from $\sqrt{z}=x^2-1+y^2$ at given x and y become:

1. move the cs. until the x-value coincides with the setpoint of the z-scale,
2. move the ws. over the y-value,
3. read the z-value.

Appropriate location of the scales within the limited size of a sliderule may be obtained by introducing a place correction term c. The example relation may be written as:

$$\sqrt{z} + c = [x^2 - 1 + c] + y^2$$

For c = -5 the sketch of the sliderule becomes:



The given example shows a very important by-product of the slide-rule construction:

> The represented expression can be solved in any direction that is for a n-parameter relation, each of the variables can be calculated if the other n-1 parameters are given.

Not every relation can be represented by a sliderule.
The most simple class of elaborative expressions is of the type:

$$f_1[x_1] + f_2[x_2] + \ldots. = 0$$

The addition may easily be performed by moving line-intervals.

Sometimes a transformation of a more complex function may yield the previous form:

$$f_1[x_1] \cdot f_2[x_2] = f_3[x_3]$$

By taking the logarithms of the functions in this case, the standard form is obtained

A more complicated system is required by:

$$f_1[x_1] \cdot f_2[x_2] + f_3[x_3] = f_4[x_4]$$

As an intermediate result has to be calculated:

$$y = f_1[x_1] \cdot f_2[x_2]$$

or transformed by taking the logarithms:

$$\log y = \log f_1[x_1] + \log f_2[x_2]$$

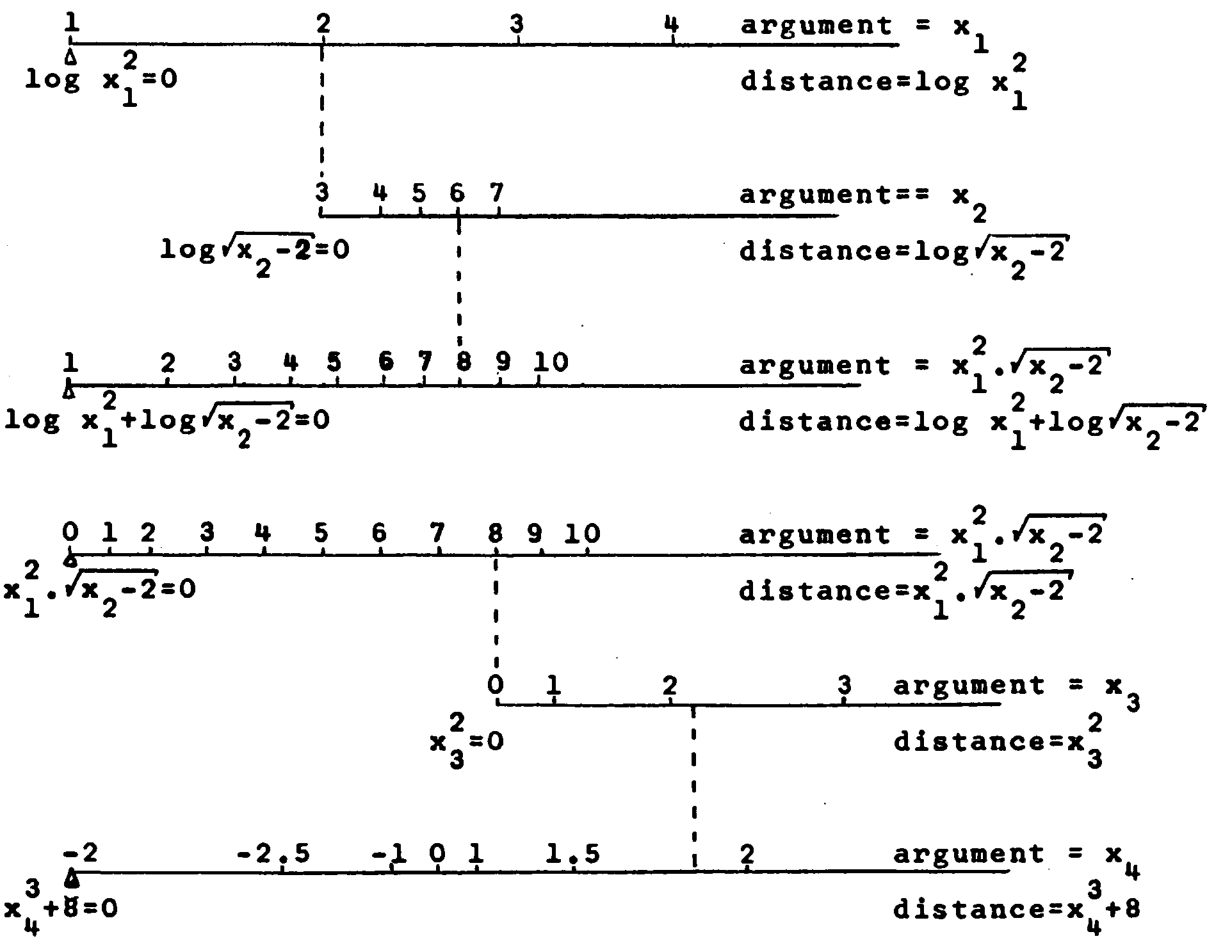
but the calculated result $f_1[x_1] \cdot f_2[x_2]$ appears now as an argument value and not as an interval-length as is required for the addition with $f_3[x_3]$.

The solution for this type of expressions will be given by the next example.

Consider the function:

$$x_1^2 \cdot \sqrt{x_2 - 2} + x_3^2 = x_4^3 + 8$$

A scale diagram for the values $x_1=2$, $x_2=6$, $x_3=$ s= 2.24 shows:

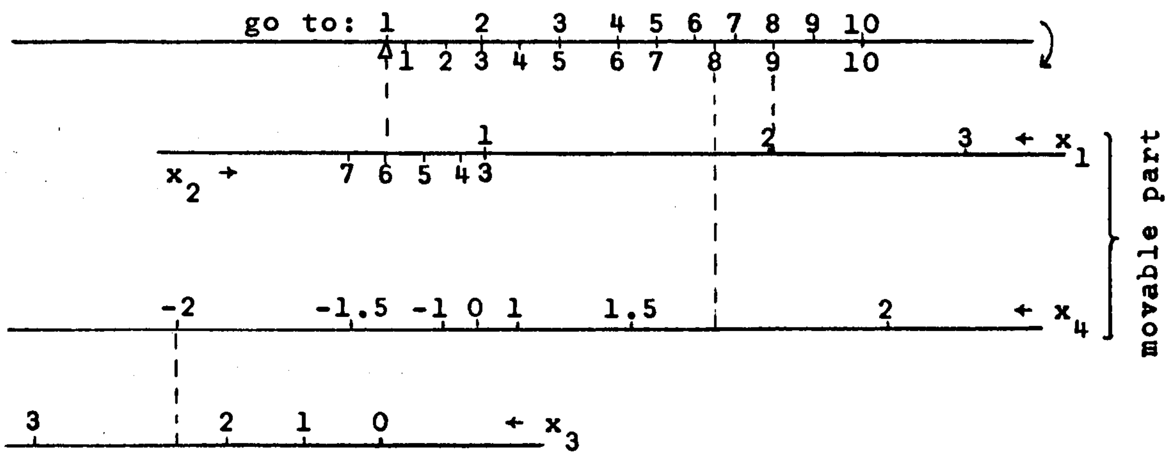


If the 2nd and the 5th scale of this diagram were engraved on the movable part of an actual sliderule, the elaborations for the calculation of the x_4-value would be:

1. put the windowslide on the x_1-value,
2. adjust the centralslide,
3. move the ws. to the x_2-value,
4. read the $x_1^2 . \sqrt{x_2 - 2}$-value on the 3rd scale,
5. move the ws. to the $x_1^2 . \sqrt{x_2 - 2}$-value on the 4th scale,
6. adjust the cs,
7. move the ws. to the x_3-value,
8. read the x_4-value.

However, the design of the sliderule and the procedure of the calculation can be simplified by:

1) choosing opposite positive directions for the scales 1 and 2 on the cs. of the sliderule,
2) combining the scales 3 and 4 to a "go to scale',
3) choosing opposite positive directions for the "go to scale" and the x_3-scale.



Now the procedure of the calculation has been reduced to:

1. move the cs. until the x_2-value coincides with the setpoint of the "go to scale",

2. move the ws. on the x_1-value and read value on upper part of the "go to scale",
3. move the ws. over this value on lower part of this scale,
4. move the cs. until the setpoint of the x_4-scale coincides with the given x_3-value,
5. read the x_4-value under the marker of the ws.

A MATHEMATICAL DEVICE

As has been shown by the previous examples, a function of the type:

$$y = \sum_{i=1}^{I} \prod_{j=1}^{J} f_{ji}[x_j]$$

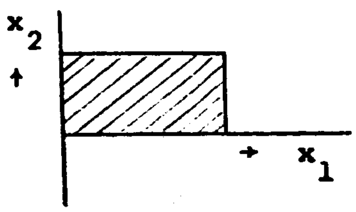
is accessible for sliderule application.

Now arises the problem of converting a function of the type:

$$y = f\ [x_1, x_2, \ldots, x_n]$$

into the standard form.

If normal algebraic operations do not yield results, a mathematical device for this conversion , as developed by Mrs.C.TAMAGNINI (see ref.),might help. The applied approximation has shown to be useful in many cases, although it is not a general solution.
The method will be explained for a three-parameter function, $y = f[x_1, x_2]$. The extension to more parameters is easy.
Consider the function $y = f[x_1, x_2]$ in a defined range:



The approximating function must be of the type:

$$y^* = \sum_{i=1}^{I} f_{1i}[x_1] \cdot f_{2i}[x_2]$$

to make it accessible for sliderule application.

Choose a point $[x_{10},x_{20}]$ in the definition field of y and consider the two curves $f_{11}[x_1]$ and $f_{21}[x_2]$ obtained by the intersection of the y-surface with the two planes parallel to the planes $x_1=0$ respectively $x_2=0$ and passing through the point $[x_{10},x_{20}]$.

As a first approximation one may try:

$$y_1 = \frac{f_{11}[x_1].f_{21}[x_2]}{f[x_{10},x_{20}]} = c_1.f_{11}[x_1].f_{21}[x_2]$$

in which $f[x_{10},x_{20}] = \frac{1}{c_1}$ can be calculated numerically.

The choice of the point $[x_{10},x_{20}]$ may be performed by covering the definition field of y by a lattice and successive tryings of the lattice points for the best approximation.

Sequentially, the surface

$$y - y_1 = f[x_1,x_2] - c_1 f_{11}[x_1].f_{21}[x_2]$$

can be treated in the same way etc., until

$$z = y - c_1 f_{11}[x_1].f_{21}[x_2] - c_2 f_{12}[x_1].f_{22}[x_2] - c_3 f_{13}[x_1].f_{23}[x_2] - \dots$$

is small enough within the field of definition.

Then:

$$y^* = \sum_{i=1}^{I} c_i.f_{1i}[x_1].f_{2i}[x_2]$$

is the requested approximation.

A MECHANICAL DEVICE

A handy tool in the laboratory is the sliderule with exchangeable scales. The annexed design has proofed to be very useful. The actual scales can be calculated and drawn on graph-paper. The drawing is then adjusted on the body of the rule and the transparent covers are fastened. The protruding part of the drawing can be removed by a razor blade.

THE PROGRAMME "ACCESS"

"ACCESS" (for Automatic Compiling of Circular Sliderule Scales), converts numerical descriptions of sliderule scales into actual drawings with the aid of a CALCOMP-plotter. "ACCESS" may be used as an independent programme or it can be incorporated in another programme which provides the numerical descriptions as has been done in the case of the programme "COOLER" which is described also.

Programme "ACCESS" may produce drawings in the frame of a rectangular coordinates system with $0 \leq x$ and $0 \leq y \leq 70$ cm, combining the components:

1. dotted arcs of any length,
2. full-line arcs of any length,
3. straight lines given in polar coordinates with regard to a centre point in cartesian coordinates.

The straight lines (division marks), may be given with an absolute length or in ratio to the radius, thus enabling an automatic scale enlargement of the components.
The programme contains also an option for surpressing a division mark if the distance between two division marks becomes smaller than a specified value.
A complete description of the options will be given in the input list.

"ACCESS" INPUT DESCRIPTION

The actual input for the "ACCESS" programme consists of a collection of fixed point-, (I-format) and floating point-numbers (E-format). A fixed point number is written without a decimal point,

utmost right in its field.
Floating point data are written with a decimal point and are eventually supplied with a fixed point exponent of 10.
The position of such a number in its field is irrelevant, only the exponent must be placed at the utmost right.

The next list gives a detailed description of the input. The symbols refer to the input sheet.

N = number of drawings to be performed;
format: I6

For each drawing the next specifications have to be repeated:

card 1 format: 2I3, I6, E12.4, 4(2I1, E10.4)

I_o = 0 the angle and size of each division mark are given in one card,

I_o = 1 all the angles are specified first, followed by a set of cards with the lengths of the division marks in the same sequence,

I_1 = 0 the angles are given in radians,

I_1 = 1 the angles are given in degrees, minutes and seconds,

I_2 = 0 the lengths of the division marks are given in cm (size(I)),

I_2 = 1 the lengths of the division marks have to be calculated from given ratios in relation to the radius R,

ARCMIN the minimum arc length in cm between two division marks: in the case that two division marks come closer than arcmin, the smallest is surpressed. If they are equal in size no action will be taken,

J, K, C(I) four additional arcs or circles may be specified on the centre coordinates x_o, y_o, but with different radius,

J = 0: an arc between the smallest and the largest angle of the division marks' specification will be drawn with a radius of C(I) cm,

J = 1: a full circle around x_o, y_o will be drawn with a radius of C(I) cm,

K = 0: continuous arc,

K = 1: dotted arc.

More contrasting arcs may be obtained by repeating the definitions, eventually with a somewhat smaller radius.

card 2 format: 4E6.2

R radius in cm,

S (only if $I_2=1$).
The length of a divisionmark at radius 10 cm and size(I)=1. The actual lengths of the division marks are calculated: $S_{ACT}(I) = 0.1 \times R \times S \times SIZE(I)$.
This factor is used for automatic scale enlargement.

X_o, Y_o Coordinates of the centre point. The drawings are produced in a frame of rectangular coordinates with $0 \leqslant X$ and $0 \leqslant Y \leqslant 70$ cm.
Thus $R_{MAX} \leqslant X_o$ and $R_{MAX} \leqslant Y_o \leqslant 70 - R_{MAX}$, in which R_{MAX} is the radius of the largest circle.

The angles and the lengths of the division marks may be specified in several ways depending on the indicators I_o, I_1 and I_2, as already been described. The symbols on the input sheet have the following meaning:

DEG(I), MIN(I), SEC(I) (only if $I_1=1$).
The angles of the division marks with the X-axis, counterclockwise, in degrees, minutes and seconds,

RAD(I) (only if $I_1=0$).
The angles of the division marks with the X-axis, counterclockwise, in radians

SIZE(I) if $I_2=0$:

the lengths of the division marks in cm,

if I_2=1:

the length ratios; the actual lengths will be calculated: $S_{ACT}(I) = 0.1 \times R \times S \times SIZE(I)$ cm.

If SIZE(I) has a positive value, the division mark is pointing outwards from the outer point of the radius R, otherwise the division marks point to the centre.

* the last card of the angles specification must have a "*" in the first column.

The number of scales[N] to be drawn in one run is not limited.

PROBLEM ACCESS | DATE | PAGE | OF

	I_0 I_1	
1		N
2		I_0 I_1 I_2 ARCMIN ∂K C_1 ∂K C_2 ∂K C_3 ∂K C_4
3		R S X_0 Y_0
4		
5	0 1	DEG(1) MIN(1) SEC(1) SIZE(1)
6	0 1	DEG(2) MIN(2) ----------
7	0 1	----------------------
8	0 1	* DEG(L) MIN(L) SEC(L) SIZE(L)
9		
10	0 0	RAD(1) SIZE(1)
11	0 0	RAD(2) ----
12	0 0	------------------
13	0 0	* RAD(L) SIZE(L)
14		
15	1 1	DEG(1) MIN(1) SEC(1)
16	1 1	DEG(2) ----------
17	1 1	----------------
18	1 1	* DEG(L) MIN(L) SEC(L)
19	1 1	SIZE(1) SIZE(2) ---------------------------------- SIZE(L)
20		
21	0 1	RAD(1)
22	0 1	RAD(2)
23	0 1	------------
24	0 1	* RAD(L)
25	0 1	SIZE(1) SIZE(2) ---------------------------------- SIZE(L)

N TIMES

PROBLEM ACCESS INPUT EXAMPLE | DATE | PAGE 1 OF

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
     3
  0  0     0  0.01      10  5.0     10  5.5     10 0.2      11 4.0
  5.0   0.0   10.0  10.0
      0.0          0.4
      0.17609      0.2
      0.30103      0.3
      0.39794      0.2
      0.47712      0.3
      0.54407      0.2
      0.60206      0.3
      0.65321      0.2
      0.69897      0.4
      0.77815      0.3
      0.84510      0.3
      0.90309      0.3
      0.95425      0.3
      1.0          0.4
      1.17609      0.2
      1.30103      0.3
      1.39794      0.2
      1.47712      0.3
      1.54407      0.2
      1.60206      0.3
      1.65321      0.2
      1.69897      0.4
```

PROBLEM ACCESS INPUT EXAMPLE | DATE | PAGE 2 OF

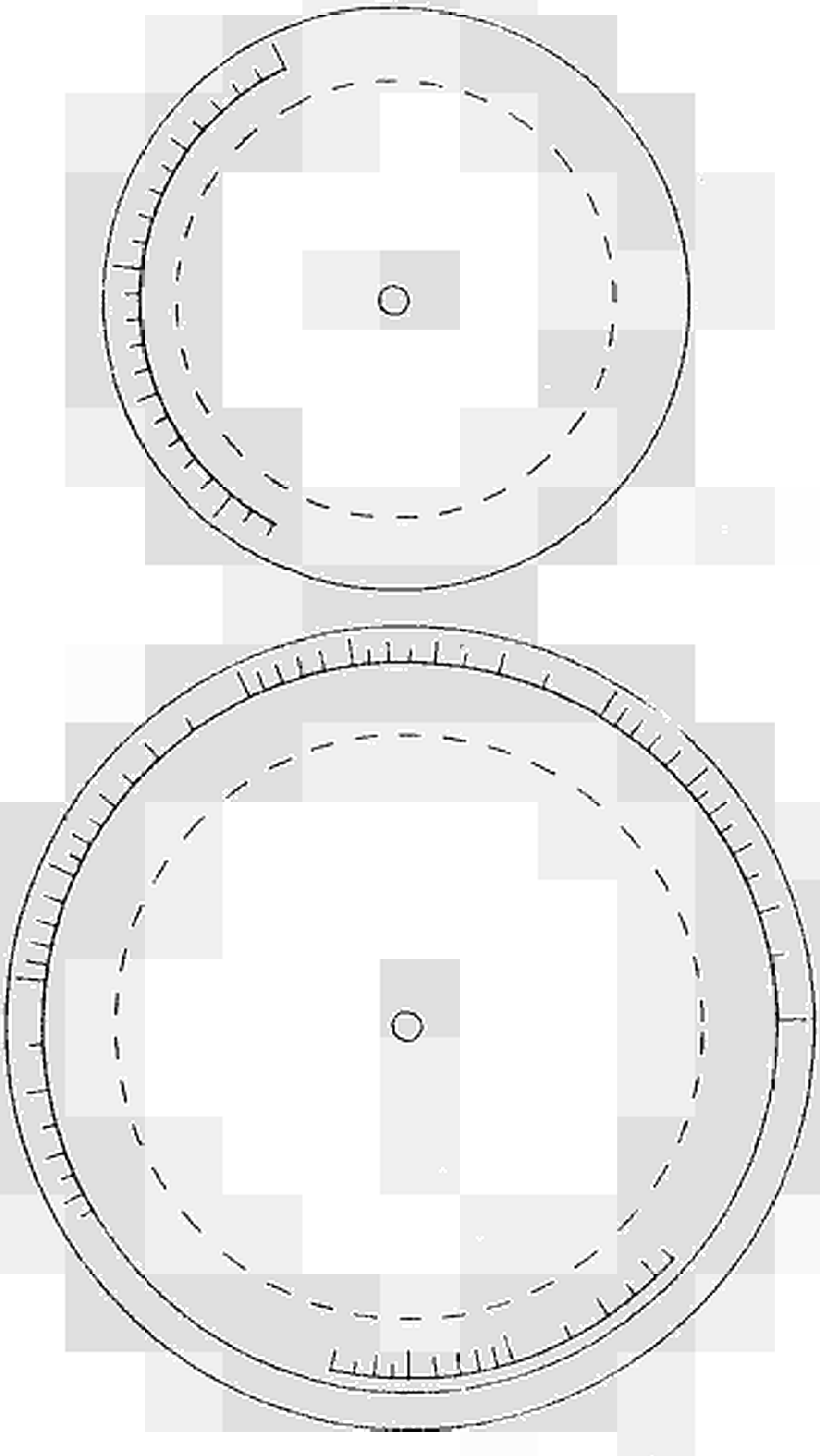
	1.77815	0.3
	1.84510	0.3
	1.90309	0.3
	1.95425	0.3
	2.0	0.4
	2.17609	0.2
	2.30103	0.3
	2.39794	0.2
	2.47712	0.3
	2.54407	0.2
	2.60206	0.3
	2.65321	0.2
	2.69897	0.4
	2.77815	0.3
	2.84510	0.3
	2.90309	0.3
	2.95425	0.3
	3.0	0.4
	3.17609	0.2
	3.30103	0.3
	3.39794	0.2
	3.47712	0.3
	3.54407	0.2
	3.60206	0.3
*	3.65321	0.2

PROBLEM ACCESS INPUT EXAMPLE | DATE | PAGE 3 OF

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
  0  0     0  0.01      10   4.0    10 0.2      00  3.5     11  3.0
  3.5   0.0   10.0  20.0
      2.0          0.4
      2.1          0.2
      2.2          0.2
      2.3          0.2
      2.4          0.2
      2.5          0.3
      2.6          0.2
      2.7          0.2
      2.8          0.2
      2.9          0.2
      3.0          0.4
      3.1          0.2
      3.2          0.2
      3.3          0.2
      3.4          0.2
      3.5          0.3
      3.6          0.2
      3.7          0.2
      3.8          0.2
      3.9          0.2
      4.0          0.4
      4.1          0.2
*     4.2          0.2
```

PROBLEM ACCESS INPUT EXAMPLE | DATE | PAGE 4 OF

```
  0  0      0 0.01      00   4.8
  4.8   0.0   10.0  10.0
      4.47712     -0.30
      4.54407     -0.20
      4.60206     -0.30
      4.65321     -0.20
      4.69897     -0.40
      4.77815     -0.30
      4.84510     -0.30
      4.90309     -0.30
      4.95425     -0.30
      5.0         -0.40
      5.17609     -0.20
      5.30103     -0.30
      5.39794     -0.20
      5.47712     -0.30
*     5.54407     -0.20
```



THE FORTRAN LISTING OF "ACCESS"

The next pages give the FORTRAN listing of "ACCESS". The programme has been written in FORTRAN IV for the IBM 360/65 with a CALCOMP-plotter. The subroutines which are called for and not presented in the next listing, are special CALCOMP-routines which are published elsewhere (see ref.).

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
              C-----PROGRAM ACCESS BY HERMAN I. DE WOLDE-----MAY 1969----------------
              C
              C       PROGRAM ACCESS DRAWS CIRCULAR SLIDE RULE SCALES
              C       THE DIVISION MARKS MAY BE GIVEN IN RADIANS OR DEGREES
              C       THE LENGTH OF EACH MARK IS SPECIFIED IN CM
              C       AT A STANDARD RADIUS OF 10 CM
              C       OR IN ABSOLUTE LENGTH UNITS ACCORDING TO IND(2)
              C
ISN 0002              DIMENSION IND(10),C(4),RAD(1000),DIVL(1000),MC(4),KC(4),X(1000,2),
                     1Y(1000,2)
ISN 0003              DIMENSION XX(2),YY(2)
ISN 0004              DATA STAR/'*    '/
ISN 0005              PI=3.141593
              C
              C-----READ INPUT DATA---------------------------------------------------
              C
ISN 0006              CALL FININ(0.0,1.0)
ISN 0007              READ (5,102) N
ISN 0008          102 FORMAT (2I6,F12.4,4(2I1,F10.4))
ISN 0009              IN=0
ISN 0010          104 IN=IN+1
ISN 0011              IF (IN.LE.N) GO TO 110
ISN 0013          105 CALL FINTRA
ISN 0014              WRITE (6,106) N
ISN 0015          106 FORMAT (1H1/I6,' SLIDERULE SCALES HAVE BEEN DRAWN')
ISN 0016              GO TO 200
ISN 0017          110 READ (5,111) IA,IB,IC,ID,ARCMIN,(MC(I),KC(I),C(I),I=1,4)
ISN 0018          111 FORMAT (4I3,F12.4,4(2I1,F10.4))
ISN 0019              WRITE (6,112)
ISN 0020          112 FORMAT (1H1/' ***** INPUT DATA *****'//)
ISN 0021              WRITE (6,114) IA,IB,ID,ARCMIN,(MC(I),KC(I),C(I),I=1,4)
ISN 0022          114 FORMAT (1H ,2I3,I6,E12.4,4(2I1,E10.3))
ISN 0023              IND(1)=IB
ISN 0024              IND(2)=ID
ISN 0025              INDO=IA
ISN 0026              READ (5,116) R,S,XC,YC
ISN 0027              WRITE (6,115)R,S,XC,YC
ISN 0028          115 FORMAT (1H ,F6.2,F6.3,2F6.2)
ISN 0029          116 FORMAT (4F6.2)
ISN 0030              IF (IND(1).LE.0) GO TO 130
ISN 0032              I=0
ISN 0033          118 I=I+1
ISN 0034              READ (5,122) ALF,RA,RB,RC,DIVL(I)
ISN 0035              WRITE (6,123) ALF,RA,RB,RC,DIVL(I)
ISN 0036          123 FORMAT (1H ,A1,F5.0,2F6.0,F6.2)
ISN 0037          122 FORMAT (A1,F5.2,3F6.2)
ISN 0038              RA=RA+RB/60.+RC/3600.
```

```
ISN 0039            RAD(I)=(RA/180.)*PI
ISN 0040            IF (ALF.EQ.STAR) GO TO 140
ISN 0042            GO TO 118
              C
ISN 0043        130 I=0
ISN 0044        132 I=I+1
ISN 0045            READ (5,134) ALF,RAD(I),DIVL(I)
ISN 0046            WRITE (6,133) ALF,RAD(I),DIVL(I)
ISN 0047        133 FORMAT (1H ,A1,F17.5,F6.2)
ISN 0048        134 FORMAT (A1,F17.5,F6.2)
ISN 0049            IF (ALF.NE.STAR) GO TO 132
              C
              C-----ORDER THE ANGLES------------------------------------------
              C
ISN 0051        140 NA=I
ISN 0052            IF (INDO.EQ.0) GO TO 143
ISN 0054            READ (5,136) (DIVL(I),I=1,NA)
ISN 0055            WRITE (6,135) (DIVL(I),I=1,NA)
ISN 0056        135 FORMAT (1H ,12F6.2)
ISN 0057        136 FORMAT (12F6.2)
ISN 0058            WRITE (6,137)
ISN 0059        137 FORMAT (1H1/5X,'DEGREES',5X,'MINUTES',5X,'SECONDS',6X,'LENGTH'/)
ISN 0060            DO 141 I=1,NA
ISN 0061            IF ((10*(I/10)-I).EQ.0) WRITE (6,138)
ISN 0063        138 FORMAT (/)
ISN 0064            RA=180.*RAD(I)/PI
ISN 0065            RAA=AINT(RA)
ISN 0066            RB=60.*(RA-RAA)
ISN 0067            RBB=AINT(RB)
ISN 0068            RCC=60.*(RB-RBB)
ISN 0069            WRITE (6,139) RAA,RBB,RCC,DIVL(I)
ISN 0070        139 FORMAT (3F12.0,F12.2)
ISN 0071        141 CONTINUE
ISN 0072        143 CONTINUE
ISN 0073            I=1
ISN 0074        142 I=I+1
ISN 0075            IF (I.GT.NA) GO TO 146
ISN 0077            IF (RAD(I)-RAD(I-1))144,142,142
ISN 0078        144 IA=I
ISN 0079        145 AA=RAD(IA)
ISN 0080            AB=DIVL(IA)
ISN 0081            RAD(IA)=RAD(IA-1)
ISN 0082            RAD(IA-1)=AA
ISN 0083            DIVL(IA)=DIVL(IA-1)
ISN 0084            DIVL(IA-1)=AB
ISN 0085            IA=IA-1
ISN 0086            IF (IA.EQ.1) GO TO 142
ISN 0088            IF (RAD(IA)-RAD(IA-1))145,142,142
```

```
ISN 0089          146 CONTINUE
                C
                C-----LENGTHS OF DIVISION MARKS-----------------------------------
                C
ISN 0090              IF (IND(2).LE.0) GO TO 152
ISN 0092              DO 150 I=1,NA
ISN 0093          150 DIVL(I)=0.1*R*S*DIVL(I)
ISN 0094          152 CONTINUE
                C
                C-----DRAW THE REQUESTED ARCS-------------------------------------------
                C
ISN 0095              CALL FINIM(XC,YC)
ISN 0096              AA=RAD(1)
ISN 0097              AB=RAD(NA)
ISN 0098              PSIA=AA
ISN 0099              PSIB=AB
ISN 0100              K=1
ISN 0101              CALL ARC(R,AA,AB,K)
                C
ISN 0102              DO 160 I=1,4
ISN 0103              IF (C(I).LT.0.01) GO TO 160
ISN 0105              K=KC(I)+1
ISN 0106              AA=PSIA
ISN 0107              AB=PSIB
ISN 0108              IF (MC(I).NE.0)AB=PSIA+2.*PI
ISN 0110              CALL ARC(C(I),AA,AB,K)
ISN 0111          160 CONTINUE
                C
                C-----TEST AND DRAW THE DIVISION MARKS---------------------------------
                C
ISN 0112              DO 164 I=1,NA
ISN 0113              X(I,1)=R*COS(RAD(I))
ISN 0114              X(I,2)=(R+DIVL(I))*COS(RAD(I))
ISN 0115              Y(I,1)=R*SIN(RAD(I))
ISN 0116              Y(I,2)=(R+DIVL(I))*SIN(RAD(I))
ISN 0117          164 CONTINUE
                C
ISN 0118              DO 170 I=2,NA
ISN 0119              IF (DIVL(I).EQ.DIVL(I-1)) GO TO 170
ISN 0121              IF ((DIVL(I)*DIVL(I-1)).LT.0.0) GO TO 170
ISN 0123              AA=(X(I,1)-X(I-1,1))**2+(Y(I,1)-Y(I-1,1))**2
ISN 0124              IF (AA.GE.(ARCMIN**2)) GO TO 170
ISN 0126              IF (ABS(DIVL(I)).GT.ABS(DIVL(I-1))) GO TO 168
ISN 0128              X(I,2)=X(I,1)
ISN 0129              Y(I,2)=Y(I,1)
ISN 0130              GO TO 170
ISN 0131          168 X(I-1,2)=X(I-1,1)
ISN 0132              Y(I-1,2)=Y(I-1,1)
```

```
ISN 0133         170 CONTINUE
              C
ISN 0134             DO 180 I=1,NA
ISN 0135             XX(1)=X(I,1)
ISN 0136             XX(2)=X(I,2)
ISN 0137             YY(1)=Y(I,1)
ISN 0138             YY(2)=Y(I,2)
ISN 0139             CALL LINE (XX,YY,2,1,1)
ISN 0140         180 CONTINUE
ISN 0141             CALL FINIM (-XC,-YC)
ISN 0142             GO TO 104
ISN 0143         200 CONTINUE
ISN 0144             STOP
ISN 0145             END
```

```
LEVEL 15 ( 1 JAN 68)                    OS/360  FORTRAN H                         DATE  70.023/15.57.22
          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002                  SUBROUTINE ARC (R,PSIA,PSIB,K)
               C
               C          SUBROUTINE ARC DRAWS AN ARC COUNTER CLOCKWISE
               C          FROM 'PSIA' TO 'PSIB', BOTH BEING ANGLES WITH THE X-AXIS
               C          R=RADIUS IN CM
               C          K=1,FULL LINE
               C          K=2,DASHED LINE
               C
ISN 0003                  DIMENSION X(1000),Y(1000)
ISN 0004                  PI=3.141593
ISN 0005                  DO 99 I=1,1000
ISN 0006               99 Y(I)=R
ISN 0007                  IF (PSIA.EQ.PSIB)PSIB=PSIA+2.*PI
ISN 0009                  DEL=1./(R*20.)
ISN 0010                  IA=1
ISN 0011                  X(IA)=PSIA
ISN 0012              100 IA=IA+1
ISN 0013                  X(IA)=X(IA-1)+DEL
ISN 0014                  IF (IA.EQ.1000) GO TO 110
ISN 0016                  IF (X(IA).LE.PSIB) GO TO 100
ISN 0018                  GO TO (104,106),K
ISN 0019              104 CALL LINEPO(X,Y,IA,1,1)
ISN 0020                  GO TO 108
ISN 0021              106 CALL DASHPO(X,Y,IA,1,1)
ISN 0022              108 CONTINUE
ISN 0023                  GO TO 120
ISN 0024              110 GO TO (112,114),K
ISN 0025              112 IK=1000
ISN 0026                  CALL LINEPO (X,Y,IK,1,1)
ISN 0027              113 IA=1
ISN 0028                  X(IA)=X(1000)
ISN 0029                  GO TO 100
ISN 0030              114 IK=1000
ISN 0031                  CALL DASHPO (X,Y,IK,1,1)
ISN 0032                  GO TO 113
ISN 0033              120 CONTINUE
ISN 0034                  RETURN
ISN 0035                  END
```

COMPUTER DESIGNED SLIDERULES

In this chapter a complete case of a special purpose sliderule is described. It concerns a problem for which a general solution could not be found. So a computer programme "COOLER" has been developed which designs and draws the abacus scales for each demanded case. The previously described programme "ACCESS" has been incorporated into "COOLER", so no extensive data transfers were required.

The actual relations are calculated in the subroutines "QLIQ" and "QGAS". By programming other subroutines, one might apply the programme "COOLER" also for other sliderules.

The applied CALCOMP-subroutines are published elsewhere.

The leaktightness of technical installations and components is often tested under circumstances which differ completely from the intended operation condition. The differences may concern temperature, pressure and filling medium. In the Euratom report, EUR 2982.e, one has derived the relations between the leak rate Q, the diameter D, and the length of a capillary L, for gases and for liquids, in the range $10^{-2} < D < 10^{2}$, especially concerning sealings.
It has been assumed that leaks of sealings and joints always occur through a number of small capillaries. The assumption of a mean diameter for all capillaries, where the leak flow-rate is caused by capillaries of various diameter, has been justified in the same report.

The flow rate of gases, in and around the transient range of pure molecular flow and viscous flow is given by:

$$Q_g = 10^{-4} \times \frac{D^3}{L}\left[0.093 \times \frac{D}{n_g} \times [p_1^2 - p_2^2] + 2.88 \times \sqrt{\frac{T}{M}} \times [p_1 - p_2]\right] \quad \text{clusec} \qquad (1)$$

in which: Q_g = gas leak rate in centilusec

(A flow rate of 1 lusec causes a pressure increase of 10^{-3} mm Hg in a vacuum of 1 litre in 1 sec.

1 lusec = 1.32×10^{-3} atm.cc/sec.

D = the diameter of the capillary in μ

L = the length of the capillary in cm

n_g = the dynamic viscosity of the gas in centipoise

T = the temperature in ^oK

M = the molecular weight of the gas

p_1 = the fill pressure in atm.

p_2 = the exit pressure in atm.

The basic formula for a laminar liquid flow is:

$$Q = 0.882 \times 10^{-6} \times \frac{D^4}{L} \times \frac{\rho_\ell}{\eta_\ell} \times [p_1 - p_2] \quad \text{mg/hour} \qquad (2)$$

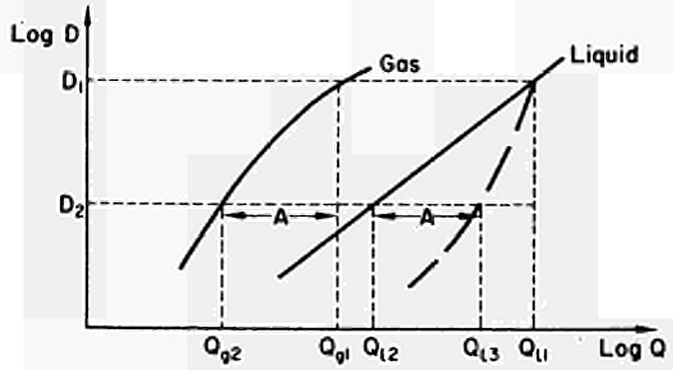
in which: Q = leak rate in mg/hour

ρ_ℓ = the viscosity in centipoise

η_ℓ = the specific gravity of the liquid

In some cases a liquid leak flow rate is influenced by two phenomena, i.e. surface tension effect and evaporation of the liquid during leaking. Then, the basic formula should be corrected according to the formulae given in the above mentioned report.

The conversion of, for example, a gas leak to a liquid leak at different pressure and temperature, may be performed by a graphical presentation of the relations (1) and (2) for a standard capillary of 1 cm length.



Assume a gas leak of Q_{g1} clusec has been estimated. This leak might have been caused by one capillary of diameter D_1 and of 1 cm length, (see the above figure). Such a capillary would cause a liquid leak of $Q_{\ell 1}$ mg/hour. However, if there is not only one single capillary, but more than one with, for example, an average diameter D_2, the liquid leak rate will be different. The number of capillaries with an arbitrary average diameter, D_2, causing a total gas leak Q_{g1}, can be calculated:

$$n = \frac{Q_{g1}}{Q_{g2}} \qquad (3)$$

in which Q_{g2} is the gas leak of one capillary.
The equivalent liquid leak rate will now be:

$$Q_{\ell 3} = \frac{Q_{g1}}{Q_{g2}} \times Q_{\ell 2} \qquad (4)$$

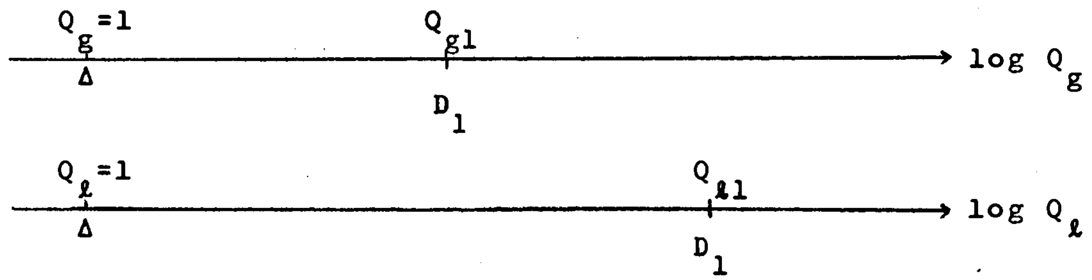
in which $Q_{\ell 2}$ is the liquid leak rate for one capillary of diameter D_2. In logarithmic notation:

$$\log Q_{\ell 3} = \log Q_{\ell 2} + \left[\log Q_{g1} - \log Q_{g2}\right] \qquad (5)$$

The second term in expression (5) is equal to the distance A in the above figure. By shifting this distance, as done in the figure, one point ($Q_{\ell 3}$, D_2) of the liquid leak curve has been found. More points may be constructed by choosing other D_i values.

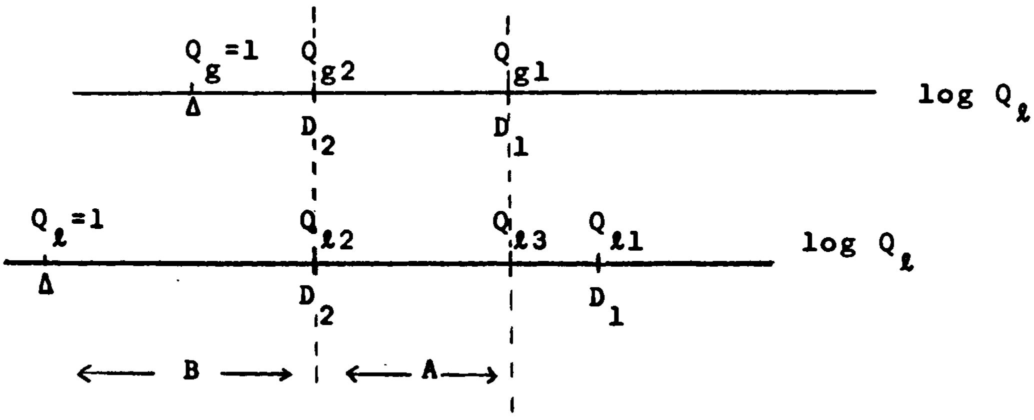
The final liquid leak curve gives an impression of the prospective leak range. A more accurate information is obtained if the gas test is repeated with another medium or at another temperature c.q. pressure. The intersection of the two leak curves gives the liquid leak and at the same time the average diameter of the capillaries.

For multiple conversion calculations, it is easier to design a slide rule.
The principles of this system will be shown for a conversion calculation as sketched in the aforegoing figure.



The length of the interval ($Q_g = 1$, Q_{g1}) represents the function's value log Q_{g1}. For $Q_g = 1$ the function value log Q_g is zero. The calculation according to expression (5) can thus be performed by moving line intervals. The marks on both sides of each scale denounce respectively the values of Q and D, belonging to the value of log Q, which is in turn expressed in mm.

Assume a gas leak of Q_{g1} clusec has been estimated. The appropriate diameter of the capillary, D_1, can be read directly. The same D_1 on the liquid scale gives immediately $Q_{\ell 1}$. Next a D_2 is chosen on the gas scale. The scales are positioned so that the D_2's on both scales coincide:



If in this position the window slide is put over Q_{g1}, Q_3 may be read immediately because:

$$B = \log Q_{\ell 2}$$
$$A = \log Q_{g1} - \log Q_{g2} \qquad (6)$$

The summation of A and B is equivalent to the expression (5);the combined manipulations to calculate a Q_3 from a given D_2 are thus:

1. put the windowslide on the D_1-value (gas scale),
2. move the central slide until D_2-value (liquid scale) is under the marker,
3. move the ws. over the D_1-value,
4. read the $Q_{\ell 3}$-value under the marker.

It is clear that the actual zero points of the functions, log Q, are not used.
The actual sliderule is designed circular.

The disadvantage of the sliderule system in this case is that the for each medium, temperature and pressure a separate scale must be designed, as it is not possible to write expression (1) in additive form, adapted for sliderule summation.
It would be possible to design a sliderule with general scales for liquids, expressing ρ,η and p_1-p_2 out of expression (2), com-

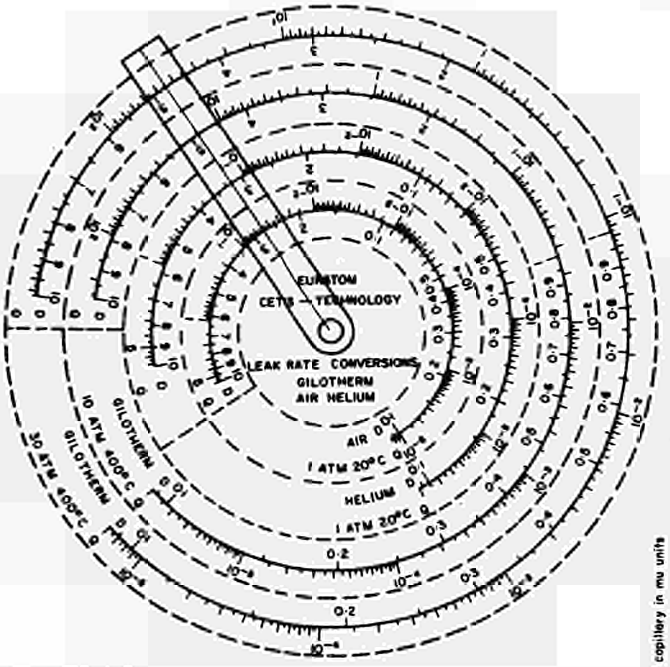
bined with specific scales for gaseous media at desired temperatures and pressures. However, this type of sliderule would be of a more complicated structure, so only the type with specific scales for gases and specific scales for liquids has been developed until now.

A computer programme has been written which calculates the relations (1) and (2) for given pressures, temperatures, viscosities and specific gravities.

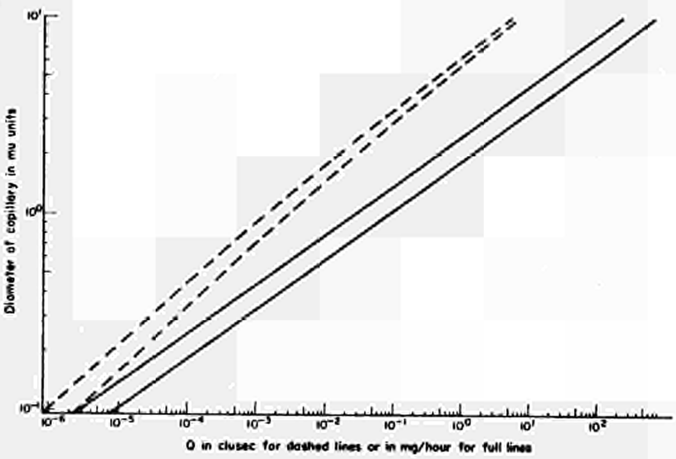
This programme, named "COOLER" from Conversion Of Leak Rates, draws the curves with log versus log scales and designs also a circular abacus for quick and multiple use. A simple example with only four scales is given for illustration.
However, any combination of media may be treated in the same way and the number of scales is not limited except for considerations of clearness and handiness.

Also two or three scales on the same circumference and more than two sliderule plates can be designed in one run.
The output of the program "COOLER" consists of:

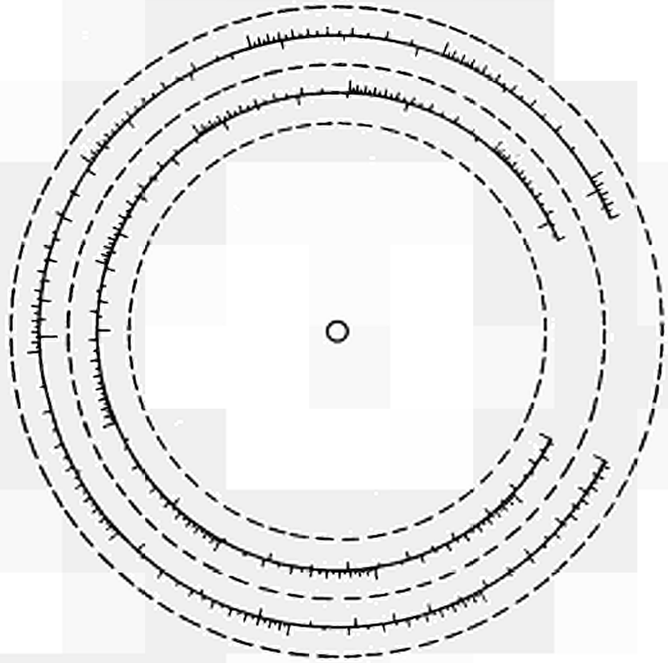
1. a graph which may be used for the conversion of leak rates as is shown in the figure on page 28,
2. a drawing of the scales for an abacus, without the D, respectively Q-values,
3. for each scale of the abacus, a table of D, respectively Q-values for the principal scale division marks.



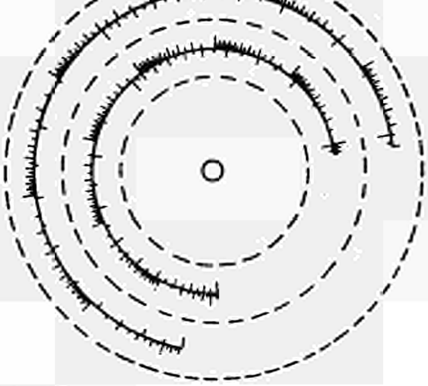
The slide rule



Computor Output 1



Computor Output 2



Computor Output 2

"COOLER" INPUT DESCRIPTION

The symbols refer to the input sheet where also a numerical example is given. The input example originates a simple slide-rule as is given by the illustrations.

NSCAL = number of specified scales

For each scale the programme requires the data:

DMIN	= minimum diameter of the capillaries,
DMAX	= maximum diameter of the capillaries,
ETHA	= viscosity of the medium,
P1	= pressure at the entrance of the capillaries (atm.),
P2	= pressure at the exit of the capillaries (atm.),
T	= temperature (^{o}K) only for gases,
EM	= molecular weight only for gases,
CORDA	= accumulating x coordinate,
CORDB	= accumulating y coordinate,
R	= radius of the scale,
PSI	= initial angle with the x-axis in radians,
SCALE	= the length of one decade in cm, (the first scale as specified in the example input requires 2 cm between $Q=10^{-6}$ and $Q=10^{-5}$),
RHO	= specific gravity only for liquids,
RCIR, RCIRA	= full dotted circles around the latest specified x-y coordinates,
TITLE	= any alphamerical description of the scale which appears in the printed output to identify the scales.

PROBLEM COOLER | DATE | PAGE | OF

```
 NSCAL
    DMIN        DMAX        ETHA         P1          P2          T
    EM          CØRDA       CØRDB        R           PSI         SCALE
    RHØ         RCIR        RCIRA      T   I   T   L   E







     4
    0.1        10.0        0.018       1.0         0.0       293.0
   29.0        20.0       10.0         3.0         0.0         2.0
    0.0         3.7         2.3    AIR 1ATM 20C
    0.1        10.0        0.019       1.0         0.0
    4.0         0.0         0.0        4.4         0.0
    0.0         5.1         0.0    HELIUM 1ATM 20C
    0.1        10.0         0.24      10.0         1.0         0.0
    0.0         0.0        14.0        5.8         0.0         3.8667
    0.8         5.1         6.5    OILØTHERM 10ATM400C
    0.1        10.0         0.24      30.0         1.0         0.0
    0.0         0.0         0.0        7.2         0.0         4.8
    0.8         7.9         0.0    OILØTHERM 30ATM400C
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
               C-----PROGRAM'COOLER' BY HERMAN I.DE WOLDE------------------------------
               C
               C      'CO'NVERSION  'O'F  'LE'AK  'R'ATES
               C
ISN 0002              DIMENSION QDQ(2,400),DQTAB(3,100)
ISN 0003              DIMENSION BCD(72),TIT(20,6),X(100),Y(100),XX(20,100),YY(20,100),NS
                     1CX(20)
ISN 0004              READ (5,98) NSCAL
ISN 0005           98 FORMAT (I6)
ISN 0006              DO 200 INSC=1,NSCAL
ISN 0007              READ (5,100) DMIN,DMAX,ETHA,P1,P2,T
ISN 0008          100 FORMAT (6E12.4)
ISN 0009              READ (5,100)EM,CORDA,CORDB,R,PSI,SCALE
ISN 0010              READ (5,103) RHO,RCIR,RCIRA,(TIT(INSC,I),I=1,6)
ISN 0011          103 FORMAT (3E12.4,6A4)
ISN 0012              CALL FINIM(CORDA,CORDB)
ISN 0013              IF(RCIR.LT.1.E-5) GO TO 101
ISN 0015              CALL CIRCLE(RCIR,2)
ISN 0016          101 IF(RCIRA.LT.1.E-5)GO TO 104
ISN 0018              CALL CIRCLE(RCIRA,2)
ISN 0019          104 CONTINUE
ISN 0020              DMIN=DMIN-0.1*DMIN
ISN 0021              IND=1
ISN 0022              IF(EM.GT.0.0) GO TO 102
ISN 0024              IND=2
ISN 0025          102 CONTINUE
ISN 0026              CALL CIRCLE(0.25,1)
ISN 0027              CALL DQVAL(DMIN,DMAX,ETHA,RHO,P1,P2,T,EM,QDQ,DQTAB,NPOIN,IND,NDQT)
               C
ISN 0028              WRITE (6,130) R,PSI
ISN 0029          130 FORMAT (/////' MAIN DIVISION MARKS OF THE SCALE R=',E12.5,'  PSI='
                     1,E12.5//)
ISN 0030              WRITE (6,131) (TIT(INSC,I),I=1,6)
ISN 0031          131 FORMAT (1H ,6A4/)
ISN 0032              DO 140 I=1,NDQT
ISN 0033              XX(INSC,I)=DQTAB(3,I)
ISN 0034          140 YY(INSC,I)=DQTAB(1,I)
ISN 0035              NSCX(INSC)=0-NDQT
ISN 0036              IF(IND.GT.1)NSCX(INSC)=NDQT
ISN 0038              NQDT=NPOIN-NDQT
ISN 0039              DO 135 I=1,NQDT
ISN 0040              IF(QDQ(2,I).LT.0.3) GO TO 135
ISN 0042              WRITE (6,132) QDQ(1,I)
ISN 0043          135 CONTINUE
ISN 0044          132 FORMAT ('  Q=',1PE12.4)
ISN 0045          136 FORMAT ('  D=',1PE12.4)
ISN 0046              DO 137 I=1,NDQT
ISN 0047              IF(DQTAB(2,I).GT.-0.3) GO TO 137
```

```
ISN 0049            WRITE (6,136) DQTAB(1,I)
ISN 0050        137 CONTINUE
          C
ISN 0051            NQDT=NPOIN-NDQT
ISN 0052            AMAX=QDQ(1,NQDT)
ISN 0053            IF(AMAX.GT.QDQ(1,NPOIN)) GO TO 106
ISN 0055            AMAX=QDQ(1,NPOIN)
ISN 0056        106 AMIN=QDQ(1,NQDT+1)
ISN 0057            IF(AMIN.LT.QDQ(1,1)) GO TO 108
ISN 0059            AMIN=QDQ(1,1)
ISN 0060        108 RANGE=ALOG10(AMAX)-ALOG10(AMIN)
ISN 0061            CALL CIRCS(R,PSI,SCALE,QDQ,RANGE,XF,YF,NPOIN)
ISN 0062        200 CONTINUE
          C
          C-----DRAW THE GRAPHS------------------------------------------------
          C
ISN 0063            CALL FINIM(30.,-20.)
ISN 0064            CALL GRLOG(XX,YY,NSCX,NSCAL)
ISN 0065            CALL FINIM(0.,0.)
ISN 0066            CALL FINTRA
ISN 0067            STOP
ISN 0068            END
```

```
LEVEL 15 ( 1 JAN 68)                OS/360  FORTRAN H                    DATE  73.023/15.59.33

      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002                 SUBROUTINE CIRCS(R,PSI,SCALE,QDQ,RANGE,XF,YF,NPOIN)
              C          SUBROUTINE CIRC DRAWS A SEGMENT FOR A Q-INTERVAL GIVEN BY RANGE
              C          STARTING FROM THE COORDINATES(R,PSI)
              C          NPOIN SCALE DIVISIONS ARE DRAWN ACCORDING TO QDQ.
              C          'SCALE'  IS THE LENGTH OF ONE DECADE
              C
ISN 0003                 DIMENSION QDQ(2,400),X(1000),Y(1000)
ISN 0004                 DEL=1./(R*20.)
ISN 0005                 PI=3.14159
ISN 0006                 ANGM=PSI+(RANGE*SCALE)/R
ISN 0007                 X(1)=PSI
ISN 0008                 IA=1
ISN 0009             100 IA=IA+1
ISN 0010                 X(IA)=X(IA-1)+DEL
ISN 0011                 IF(X(IA).LT.ANGM) GO TO 100
ISN 0013                 SCA=SCALE/R
ISN 0014                 DO 102 I=1,IA
ISN 0015             102 Y(I)=R
ISN 0016                 CALL LINEPO(X,Y,IA,1,1)
ISN 0017                 CALL LINEPO(X,Y,IA,1,1)
ISN 0018                 CALL LINEPO(X,Y,IA,1,1)
ISN 0019                 XF=X(IA)
ISN 0020                 YF=Y(IA)
ISN 0021                 QMIN=QDQ(1,1)
ISN 0022                 AQMIN=ALOG10(QMIN)
ISN 0023                 DO 106 I=1,NPOIN
ISN 0024                 ORD=PSI+(ALOG10(QDQ(1,I))-AQMIN)*SCA
ISN 0025                 DI=QDQ(2,I)
ISN 0026                 CALL DIV(ORD,R,DI)
ISN 0027             106 CONTINUE
ISN 0028                 RETURN
ISN 0029                 END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002                   SUBROUTINE DQVAL(DMIN,DMAX,ETHA,RHO,P1,P2,T,EM,QDQ,DQTAB,NPOIN,IND
                          1,NDQT)
                 C         SUBROUTINE DQVAL CALCULATES THE VALUES OF Q WHERE THE Q DIVISION
                 C         MARKS AND THE D DIVISION MARKS MUST BE PLACED.
                 C
ISN 0003                   DIMENSION QDQ(2,400),DQTAB(3,100),QDTAB(2,300)
                 C
ISN 0004                   GO TO (100,102),IND
ISN 0005               100 CALL QGAS(DMIN,ETHA,P1,P2,T,EM,QMIN)
ISN 0006                   CALL QGAS(DMAX,ETHA,P1,P2,T,EM,QMAX)
ISN 0007                   GO TO 104
ISN 0008               102 CALL QLIQ(DMIN,ETHA,P1,P2,RHO,QMIN)
ISN 0009                   CALL QLIQ(DMAX,ETHA,P1,P2,RHO,QMAX)
ISN 0010               104 CALL DIVMRK(DMIN,DMAX,QMIN,QMAX,DQTAB,QDTAB,NDQT,NQDT)
                 C
ISN 0011                   DO 110 I=1,NQDT
ISN 0012                   QDQ(1,I)=QDTAB(1,I)
ISN 0013               110 QDQ(2,I)=QDTAB(2,I)
                 C
ISN 0014                   DO 118 I=1,NDQT
ISN 0015                   IA=I+NQDT
ISN 0016                   DD=DQTAB(1,I)
ISN 0017                   GO TO (112,114),IND
ISN 0018               112 CALL QGAS(DD,ETHA,P1,P2,T,EM,QA)
ISN 0019                   GO TO 116
ISN 0020               114 CALL QLIQ(DD,ETHA,P1,P2,RHO,QA)
ISN 0021               116 DQTAB(3,I)=QA
ISN 0022                   QDQ(1,IA)=QA
ISN 0023                   QDQ(2,IA)=DQTAB(2,I)
ISN 0024               118 CONTINUE
ISN 0025                   NPOIN=NQDT+NDQT
ISN 0026                   RETURN
ISN 0027                   END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002                 SUBROUTINE QLIQ(D,ETHA,P1,P2,RHO,QL)
              C          SUBROUTINE QLIQ CALCULATES THE LEAK RATE FOR A LIQUID
              C          THROUGH A STANDARD CAPILLARY OF UNIT LENGTH.
              C
ISN 0003                 QL=2.45*D**4*(P1-P2)*1.E-10/ETHA
ISN 0004                 QL=QL*RHO*3.6*1.E+6
ISN 0005                 RETURN
ISN 0006                 END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002              SUBROUTINE QGAS(D,ETHA,P1,P2,T,EM,QG)
              C       SUBROUTINE QGAS CALCULATES THE LEAK RATE FOR A GAS
              C       THROUGH A STANDARD CAPILLARY OF UNIT LENGTH.
              C
ISN 0003              ARGA=2.88*(P1-P2)*SQRT(T/EM)
ISN 0004              ARGB=0.093*D*(P1**2-P2**2)/ETHA
ISN 0005              QG=D**3*(ARGA+ARGB)*1.E-6
ISN 0006              QG=QG*100.
ISN 0007              RETURN
ISN 0008              END
```

```
       COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002            SUBROUTINE DIVMRK(DMIN,DMAX,QMIN,QMAX,DQTAB,QDTAB,NDQT,NQDT)
            C       SUBROUTINE DIVMRK CALCULATES THE NECESSARY DIVISION MARKS FOR THE
            C      1D/Q SCALES AND GIVES ALSO THE LENGTHS OF THE MARKS IN CM.
            C
ISN 0003            DIMENSION DQTAB(3,100),QDTAB(2,300),AA(18),EN(18)
ISN 0004            DATA AA/1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5,7.0,7.5,8.
                   10,8.5,9.0,9.5/
ISN 0005            DATA EN/0.4,0.1,0.2,0.1,0.2,0.1,0.2,0.1,0.3,0.1,0.2,0.1,0.2,0.1,0.
                   12,0.1,0.2,0.1/
            C
ISN 0006            NDQT=0
ISN 0007            NQDT=0
ISN 0008            FACT=1.E-20
            C
ISN 0009            DO 102 I=1,30
ISN 0010            DO 100 J=1,18
ISN 0011            ARGA=FACT*AA(J)
ISN 0012            ARGB=0.-EN(J)
ISN 0013            IF(ARGA.LT.DMIN) GO TO 98
ISN 0015            IF(ARGA.GT.DMAX) GO TO 98
ISN 0017            NDQT=NDQT+1
ISN 0018            DQTAB(1,NDQT)=ARGA
ISN 0019            DQTAB(2,NDQT)=ARGB
            C
ISN 0020         98 IF(ARGA.LT.QMIN) GO TO 100
ISN 0022            IF(ARGA.GT.QMAX) GO TO 100
ISN 0024            NQDT=NQDT+1
ISN 0025            QDTAB(1,NQDT)=ARGA
ISN 0026            QDTAB(2,NQDT)=EN(J)
ISN 0027        100 CONTINUE
            C
ISN 0028            FACT=FACT*10.
ISN 0029        102 CONTINUE
ISN 0030            RETURN
ISN 0031            END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002              SUBROUTINE DIV(PHI,R,EN)
                C     DIV DRAWS A LINE OF LENGTH 'EN' CM
                C     PERPENDICULAR ON A CIRCLE WITH RADIUS 'R'
                C     THIS LINE POINTS INWARDS IF 'EN' IS NEGATIVE
                C     'PHI' IS THE ANGLE OF THE RADIUS WITH THE X-AXIS.
                C
ISN 0003              DIMENSION X(2),Y(2)
ISN 0004              X(1)=R*COS(PHI)
ISN 0005              X(2)=(R+EN)*COS(PHI)
ISN 0006              Y(1)=R*SIN(PHI)
ISN 0007              Y(2)=(R+EN)*SIN(PHI)
ISN 0008              CALL LINE(X,Y,2,1,1)
ISN 0009              RETURN
ISN 0010              END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002                  SUBROUTINE CIRCLE(R,IK)
              C           SUBROUTINE CIRCLE DRAWS A CIRCLE WITH RADIUS R
              C           AROUND THE POINT (0,0)
              C           IK=1,CONTINOUS LINE
              C           IK=2,DASHED LINE
              C
ISN 0003                  DIMENSION X(1000),Y(1000)
ISN 0004                  PI=3.14159
ISN 0005                  DEL=1./(R*10.)
ISN 0006                  IA=1
ISN 0007                  X(IA)=0.
ISN 0008              100 IA=IA+1
ISN 0009                  X(IA)=X(IA-1)+DEL
ISN 0010                  IF(X(IA).LT.(2.*PI)) GO TO 100
ISN 0012                  DO 102 I=1,IA
ISN 0013              102 Y(I)=R
              C
ISN 0014                  GO TO (104,106),IK
ISN 0015              104 CALL LINEPO(X,Y,IA,1,1)
ISN 0016                  GO TO 108
ISN 0017              106 CALL DASHPO(X,Y,IA,1,1)
ISN 0018              108 CONTINUE
ISN 0019                  RETURN
ISN 0020                  END
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF
ISN 0002                SUBROUTINE GRLOG(XX,YY,NSCX,NSCAL)
                  C
                  C     GRLOG DRAWS 'NSCAL' CURVES ON LOG-LOG SCALE
                  C     CURVE I IS GIVEN BY IABS(NSCX(I)) POINTS
                  C     IF NSCX IS NEGATIVE A DOTTED CURVE WILL BE DRAWN.
                  C
                  C
ISN 0003                DIMENSION XX(20,100),YY(20,100),NSCX(20)
ISN 0004                DIMENSION AA(9),EN(9),V(100),W(100),BC(1),ALX(15),ALY(9)
ISN 0005                DATA AA/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0/
ISN 0006                DATA EN/0.3,0.1,0.1,0.1,0.2,0.1,0.1,0.1,0.1/
ISN 0007                DATA BC/'10'/
ISN 0008                DATA ALX/'Q IN CLUSEC FOR DASHED LINES OR IN MG/HOUR FOR FULL LINE
                       1S    '/
ISN 0009                DATA ALY/'DIAMETER OF CAPILLARY IN MU UNITS    '/
ISN 0010                YLENG=15.
ISN 0011                XLENG=23.
ISN 0012                XMIN=XX(1,1)
ISN 0013                XMAX=XX(1,1)
ISN 0014                YMIN=YY(1,1)
ISN 0015                YMAX=YY(1,1)
ISN 0016                DO 100 I=1,NSCAL
ISN 0017                N=IABS(NSCX(I))
ISN 0018                DO 100 J=1,N
ISN 0019                IF(XMIN.GT.XX(I,J))XMIN=XX(I,J)
ISN 0021                IF(XMAX.LT.XX(I,J))XMAX=XX(I,J)
ISN 0023                IF(YMIN.GT.YY(I,J))YMIN=YY(I,J)
ISN 0025                IF(YMAX.LT.YY(I,J))YMAX=YY(I,J)
ISN 0027            100 CONTINUE
ISN 0028                AX=XLENG/(ALOG10(XMAX)-ALOG10(XMIN))
ISN 0029                BX=0.-ALOG10(XMIN)
ISN 0030                AY=YLENG/(ALOG10(YMAX)-ALOG10(YMIN))
ISN 0031                BY=0.-ALOG10(YMIN)
                  C
                  C-----DRAW THE AXIS --------------------------------------------------
                  C
ISN 0032                V(1)=0.0
ISN 0033                W(1)=0.0
ISN 0034                V(2)=AX*(ALOG10(XMAX)+BX)
ISN 0035                W(2)=0.0
ISN 0036                CALL LINE(V,W,2,1,1)
ISN 0037                V(2)=0.0
ISN 0038                W(2)=AY*(ALOG10(YMAX)+BY)
ISN 0039                CALL LINE(V,W,2,1,1)
ISN 0040                FACT=1.E-20
ISN 0041                DO 104 I=1,30
```

```
ISN 0042          DO 102 J=1,9
ISN 0043          ARGA=FACT*AA(J)
ISN 0044          ARGB=EN(J)
ISN 0045          IF(ARGA.LT.XMIN) GO TO 102
ISN 0047          IF(ARGA.GT.XMAX) GO TO 102
ISN 0049          V(1)=AX*(ALOG10(ARGA)+BX)
ISN 0050          W(1)=0.0
ISN 0051          V(2)=V(1)
ISN 0052          W(2)=0.0-ARGB
ISN 0053          CALL LINE(V,W,2,1,1)
ISN 0054          IF(ABS(ARGB).LT.0.25) GO TO 102
ISN 0056          ARGA=ARGA+0.0001*ARGA
ISN 0057          IF(ARGA.LT.1.0) ARGA=ARGA-0.0002*ARGA
ISN 0059          ARGA=ALOG10(ARGA)
ISN 0060          YT=W(2)-0.15
ISN 0061          XT=V(1)
ISN 0062          CALL NUMBER(XT,YT,0.15,0.0,ARGA,-1)
ISN 0063          XT=V(1)-0.25
ISN 0064          YT=YT-0.2
ISN 0065          CALL SYMBL4(XT,YT,0.2,0.0,BC,2)
ISN 0066      102 CONTINUE
ISN 0067          FACT=FACT*10.
ISN 0068      104 CONTINUE
ISN 0069          FACT=1.E-20
ISN 0070          DO 112 I=1,30
ISN 0071          DO 110 J=1,9
ISN 0072          ARGA=FACT*AA(J)
ISN 0073          ARGB=EN(J)
ISN 0074          IF(ARGA.LT.YMIN) GO TO 110
ISN 0076          IF(ARGA.GT.YMAX) GO TO 110
ISN 0078          V(1)=0.0
ISN 0079          W(1)=AY*(ALOG10(ARGA)+BY)
ISN 0080          V(2)=0.0-ARGB
ISN 0081          W(2)=W(1)
ISN 0082          CALL LINE(V,W,2,1,1)
ISN 0083          IF(ABS(ARGB).LT.0.25) GO TO 110
ISN 0085          ARGA=ARGA+0.0001*ARGA
ISN 0086          IF(ARGA.LT.1.0) ARGA=ARGA-0.0002*ARGA
ISN 0088          ARGA=ALOG10(ARGA)
ISN 0089          XT=V(2)-0.18
ISN 0090          YT=W(2)
ISN 0091          CALL NUMBER(XT,YT,0.15,0.0,ARGA,-1)
ISN 0092          XT=XT-0.25
ISN 0093          YT=YT-0.2
ISN 0094          CALL SYMBL4(XT,YT,0.2,0.0,BC,2)
ISN 0095      110 CONTINUE
ISN 0096          FACT=FACT*10.
ISN 0097      112 CONTINUE
```

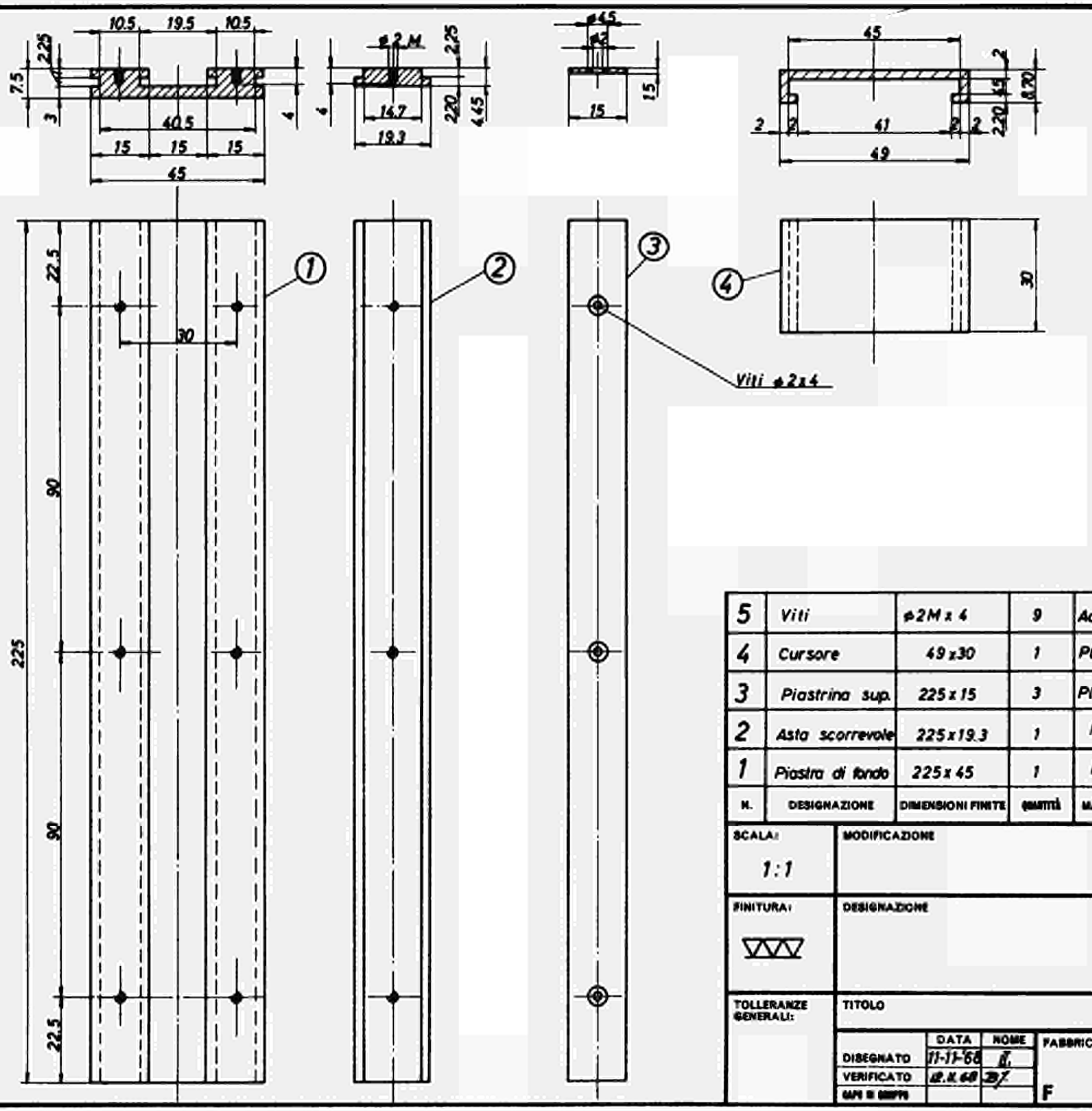
```
              C
              C-----DRAW THE ACTUAL CURVES -----------------------------
              C
ISN 0098            DO 126 IN=1,NSCAL
ISN 0099            INP=IABS(NSCX(IN))
ISN 0100            DO 120 I=1,INP
ISN 0101            V(I)=AX*(ALOG10(XX(IN,I))+BX)
ISN 0102        120 W(I)=AY*(ALOG10(YY(IN,I))+BY)
ISN 0103            IF(NSCX(IN).GT.0) GO TO 124
ISN 0105            CALL DASH(V,W,INP,1,1)
ISN 0106            GO TO 126
ISN 0107        124 CALL LINE(V,W,INP,1,1)
ISN 0108        126 CONTINUE
ISN 0109            YT=-1.
ISN 0110            XT=12.
ISN 0111            CALL SYMBL4(XT,YT,0.2,0.0,ALX,60)
ISN 0112            XT=-1.
ISN 0113            YT=AY*(ALOG10(YMAX)+BY)+0.5
ISN 0114            CALL SYMBL4(XT,YT,0.2,0.0,ALY,36)
ISN 0115            RETURN
ISN 0116            END
```

REFERENCES

P. MOINIL; J. PIRE: Programmation relative au CALCOMP.
Report EUR 2280.f (1965)

C. TAMAGNINI-MONGINI; G. GAGGERO: Design of an abacus for Gammagraphy Problems, (to be published)

H.I. DE WOLDE; J. AMESZ: Computer Designed Sliderules for the Conversion of Leak Flow Rates. Vacuum, 19,8 (1969)



N.	DESIGNAZIONE	DIMENSIONI FINITE	QUANTITÀ	MATERIALE
5	Viti	⌀2M x 4	9	Acc. inox.
4	Cursore	49 x 30	1	Plexiglas
3	Piastrina sup.	225 x 15	3	Plexiglas
2	Asta scorrevole	225 x 19.3	1	P.V.C
1	Piastra di fondo	225 x 45	1	P.V.C

SCALA: 1:1	MODIFICAZIONE
FINITURA: ▽▽▽	DESIGNAZIONE
TOLLERANZE GENERALI:	TITOLO

	DATA	NOME	FABBRICAZ. N.:
DISEGNATO	11-11-68	[illegible]	
VERIFICATO	12.11.68	[illegible]	
CAPO DI GRUPPO			F

NOTICE TO THE READER

All scientific and technical reports are announced, as and when they are issued, in the monthly periodical **"euro abstracts",** edited by the Centre for Information and Documentation (CID). For subscription (1 year : US$ 16.40, £ 6.17, BF 820) or free specimen copies please write to :

Handelsblatt GmbH
"euro abstracts"
Postfach 1102
D 4 Düsseldorf 1 (Deutschland)

or

Office de vente des publications officielles
des Communautés européennes
37, rue Glesener
Luxembourg

To disseminate knowledge is to disseminate prosperity — I mean general prosperity and not individual riches — and with prosperity disappears the greater part of the evil which is our heritage from darker times.

Alfred Nobel

SALES OFFICES

All reports published by the Commission of the European Communities are on sale at the offices listed below, at the prices given on the back of the front cover. When ordering, specify clearly the EUR number and the title of the report which are shown on the front cover.

SALES OFFICE FOR OFFICIAL PUBLICATIONS OF THE EUROPEAN COMMUNITIES

37, rue Glesener, Luxembourg (Compte chèque postal N° 191-90)

BELGIQUE — BELGIË
MONITEUR BELGE
rue de Louvain 40-42 - 1000 Bruxelles
BELGISCH STAATSBLAD
Leuvenseweg 40-42 - 1000 Brussel

DEUTSCHLAND
BUNDESANZEIGER
Postfach - 5000 Köln 1

FRANCE
SERVICE DE VENTE EN FRANCE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
26, rue Desaix - 75 Paris 15e

ITALIA
LIBRERIA DELLO STATO
Piazza G. Verdi, 10 - 00198 Roma

LUXEMBOURG
OFFICE DE VENTE DES PUBLICATIONS OFFICIELLES DES COMMUNAUTES EUROPEENNES
37, rue Glesener - Luxembourg

NEDERLAND
STAATSDRUKKERIJ
Christoffel Plantijnstraat - Den Haag

UNITED KINGDOM
H. M. STATIONERY OFFICE
P.O. Box 569 - London S.E.1

Commission of the European Communities
D.G. XIII - C.I.D.
29, rue Aldringer
Luxembourg

CDNA04499ENC