PLOTTERRUTINER FÖR PDP-15

JAN-OLOF LINDELL

RAPPORT RE-101, oktober 1971

P L O T T E R R U T I N E R F Ö R P D P - 1 5

Examensarbete utfört vid
Instutionen för Reglerteknik
vid
Lunds Tekniska Högskola
av
Jan-Olof Lindell

Handledare: Johan Wieslander

Oktober 1971

SUMMARY

The division of Automatic Control of the Technical University of Lund owns a PDP-15 computer.
A xy-plotter, in the shape of a storage oscilloscope made by Tektronix, is connected to the computer.
Below follows a description of the structure and function of a number of PLOTTING ROUTINES for the xy-plotter. The plotting routines are written in Assembler language (MACRO-15) and are primarily intended for use with programs written in Fortran language.

SAMMANFATTNING

Instutionen för Reglerteknik i Lund har en PDP-15 dator.
Till denna är ansluten en xy-skrivare i form av ett minnesoscilloskop av Tektronix tillverkning.
Nedan följer en beskrivning av uppbyggnaden och funktionen av ett antal PLOTTERRUTINER till denna skrivare. Plotterrutinerna är skrivna i Assemblerspråk (MACRO-15), och främst avsedda att användas i program skrivna i Fortran.

INNEHÅLLSFÖRTECKNING

INLEDNING

En xy-skrivare till en dator kan liknas vid en penna som kan flyttas till en given koordinat på ett xy-plan, med pennan antingen nedsänkt i skrivläge eller upplyft i icke-skrivläge. Vid förflyttning av pennan från en koordinat till en annan är förflyttningshastigheten lika stor i y-led som i x-led. Detta innebär att pennan endast kan rita räta linjer i axelriktningarna samt i riktningarna med 45, 135, 225 och 315 graders riktningsvinkel. Vill man rita räta linjer i andra riktningar eller rita böjda kurvor, måste man approximera dessa med räta linje-element i ovan nämnda möjliga riktningar, vilket kan ge snygga linjer om bara steglängden på xy-skrivaren är tillräckligt liten.

För att xy-skrivaren skall vara användbar fordras alltså att man har tillgång till ett antal program som för varje steg bestämmer i vilken av ovan nämnda riktningar pennan skall flyttas för att man skall få en så god approximation som möjligt till en önskad figur. Föreliggande examensarbete har bestått i att skriva ett antal sådana program till en xy-skrivare i form av ett minnesoscilloskop, som är ansluten till en PDP-15 dator på Institutionen för Reglerteknik vid Lunds Tekniska Högskola.

"Pennan" motsvaras vid den här omtalade skrivaren, alltså av en elektronstråle och "pennan uppe" (icke skrivläge) motsvaras av att elektronstrålen är släckt, och "pennan nere" (skrivläge) av att elektronstrålen är tänd. På grund av det enklare uttryckssättet har jag dock även i fortsättningen talat om "pennan" och "pennan uppe" eller "pennan nere", och man bör då hålla i minnet det rätta förhållandet.

Programmen är främst avsedda att användas som subrutiner i fortranprogram. De är skrivna i PDP-15 Assembler (kallat MACRO-15). Viss hjälp, främst för att ta reda på vilka egenskaper nämnda subrutiner bör ha, har jag fått från ett liknande arbete utfört i Japan för en xy-skrivare till en PDP-7, nämligen DECUS no 7-45, februari 1968.

Ett programbibliotek som detta är uppbyggt kring uppgiften att av en mängd mätpunkter rita en kurva.

För att börja med själva kurvritningen så löses denna av programmet LINE.

Oftast behövs det före uppritningen av en kurva en skalning av mätvärdena för att dessa skall få plats på tillgängligt utrymme på oscilloskopskärmen. Programmet SCALE sköter om detta.

Naturligtvis vill man också ha möjlighet att rita koordinataxlar. För den sakens skull finns ett speciellt program som heter AXIS.

På axlarna vill man ha siffror och i övrigt vill man också ha möjlighet att skriva text. Det är alltså nödvändigt att ha ett program som ritar upp vilket som helst av de tecken som finns på en skrivmaskin. Detta gör programmet SYMBOL.

I programbiblioteket ingår också ett program, NUMBER, som skriver ut ett avrundat decimalvärde av ett tal, som i fortranprogrammet representeras i flytande räkning.

I alla ovannämnda program är man beroende av att kunna dra en linje från en godtycklig koordinat till en annan godtycklig koordinat d.v.s. att dra en rät linje i en godtycklig ritning. Denna uppgift ombesörjes av programmet PLOTTA som alltså är det grundprogram som de övriga bygger på. Nedan följer en närmare beskrivning av vart och ett av programmen.

PLOTTA

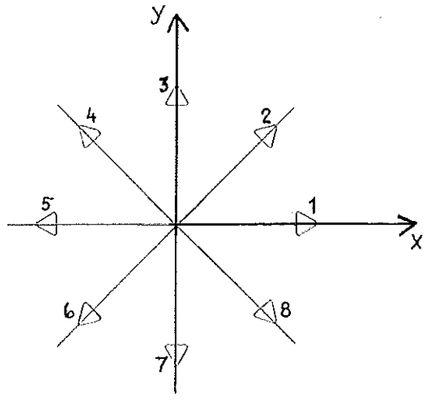
PLOTTA anropas från fortran med följande sats:

CALL PLOTTA (IX,IY,IPEN)

Vid detta anrop ritar pennan en approximation av en rät linje från nuvarande koordinat (d.v.s. den koordinat pennan stannade på vid ett tidigare anrop av PLOTTA) till koordinaten IX/IY med pennan i det tillstånd (uppe eller nere) som anges av IPEN. IPEN användes också för att ange om nytt koordinatsystem, med origo i IX och IY, skall väljas innan nästa anrop av PLOTTA.

IX och IY måste vara heltal och ha sådana värden, att punkten IX/IY verkligen hamnar på oscilloskopskärmen. Möjliga värden på IX och IY bestämmes alltså av var man valt sitt origo tidigare. Skärmen har en storlek av ungefär 1000 steg i x-led och 800 steg i y-led.

Fig. 1



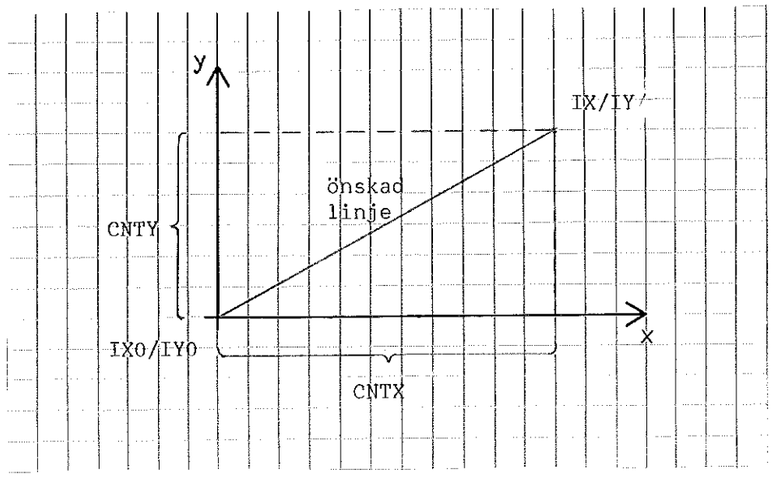
I fig. 1 är de 8 riktningar utritade i vilka xy-skrivaren kan rita räta linjer utan approximation. Då en linje skall ritas i annan riktning, approximeras den med linje-element från de 2 "möjliga riktningar", som ligger på ömse sidor om den önskade riktningen, d.v.s. linjer med riktnings-

vinkel 0-45^{o} bygges upp av linje-element i riktning 1 och 2
45-90^{o} - " - 2 och 3
90-135^{o} - " - 3 och 4

osv.

För att bestämma den önskade linjens riktning bildas uttrycken CNTX = IX - IXO och CNTY = IY - IYO, där IXO och IYO är lägeskoordinaterna i det för tillfället rådande koordinatsystemet, för pennans utgångsposition. (CNTX och CNTY anger också antalet steg vi ska gå i x- resp. y-led och användes därför också som räknare.)

Fig. 2



Tecknen på CNTX och CNTY anger i vilken kvadrant linjens riktning ligger. Man gör därefter arrangemang för att pennan skall röra sig endast i denna kvadrant. Arrangemanget består i att man tilldelar de variabler, som anger ökningen i x- resp. y-led för varje steg, värdet -1 eller 1. Dessa variabler kallas i programmet DELX och DELY och får för de olika kvadranterna följande värden:

Fig. 3

	1:a kvadr.	2:a kvadr.	3:e kvadr.	4:e kvadr.
DELX	1	-1	-1	1
DELY	1	1	-1	-1

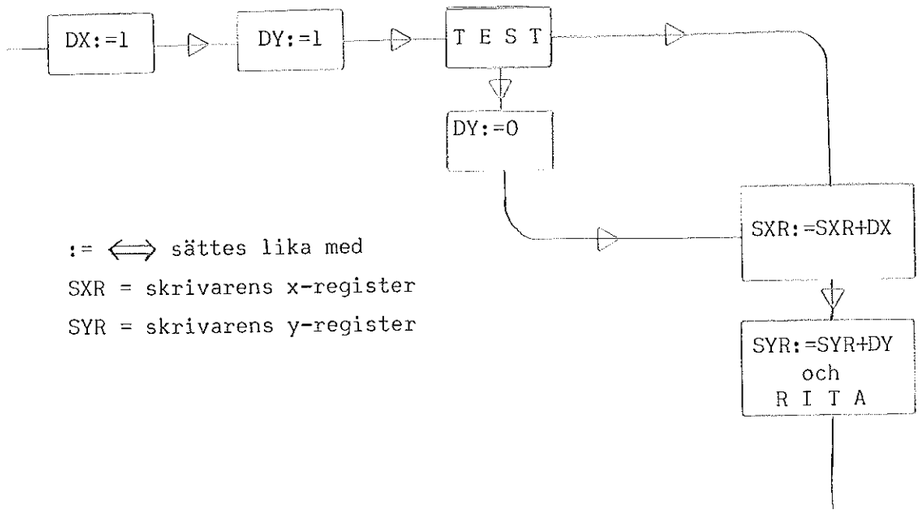
Härefter göres CNTX och CNTY positiva, och vi kan i fortsättningen göra alla kommande steg som om den önskade linjen låg i första kvadranten, I fortsättningen antar vi därför att den önskade linjen ligger i första kvadranten.

Men, man behöver bestämma riktningen på 45^{o} när, och detta görs enkelt genom att jämföra storleken av CNTX och CNTY, som ju nu båda är positiva (se fig. 2). Om CNTX > CNTY är riktningsvinkeln $0\text{-}45^{o}$, om CNTY > CNTX är riktningsvinkeln $45\text{-}90^{o}$.

Fig. 4 visar den loop som styr förflyttningen av pennan i varje steg.

Fig. 4

Med antagandet att den önskade linjen ligger i första kvadranten blir enligt fig. 3 DELX=1 och DELY=1 och vi får:



:= ⟺ sättes lika med
SXR = skrivarens x-register
SYR = skrivarens y-register

Från testet finns två vägar:

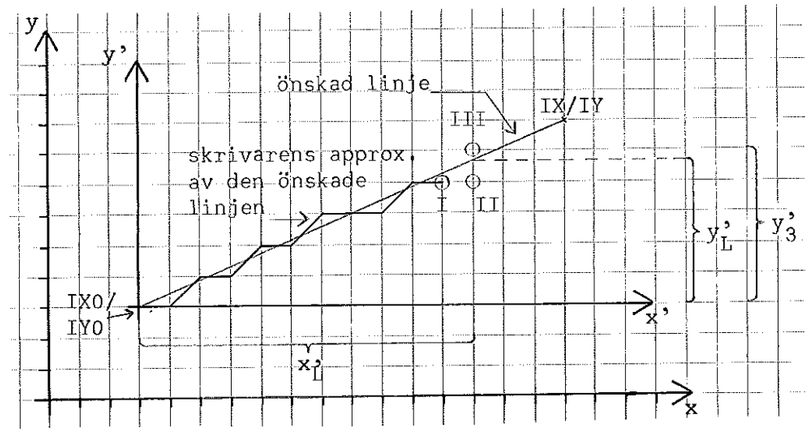
1) Skrivarens x-register ökas med ett och y-registret likaledes med ett. Vid kommandot RITA förflyttar sig pennan ett steg i 45^{o}-vinkel p.g.a. att skrivarens förflyttningshastighet är lika i x- och y-led.

2) Skrivarens x-register ökas med ett och dess y-register förblir oförändrat. Vid RITA flyttas pennan ett steg i x-riktningen.

Man inser genast att man med dessa två möjligheter får en linje vars riktningsvinkel är mellan 0 och 45 grader.
Om man finner att CNTY är större än CNTX vid jämförelsen dem emellan ovan, nollställes DX istället för DY efter TEST i fig. 4, så att man får de möjliga stegen i 45^{o} - riktningen och y-led istället för 45^{o}-riktningen och x-led. Man får då en linje med riktningsvinkel mellan 45^{o} och 90^{o}.

Det som nu återstår är att <u>fördela</u> stegen i de två riktningarna, så att linjen får den önskade riktningen. Denna fördelning sker vid TEST i fig. 4 och vi behöver alltså en passande algoritm för detta test.

Fig. 5



Om pennan befinner sig i punkt I i fig. 5 kan den i nästa steg förflyttas till antingen punkt II eller punkt III. Det gäller alltså att avgöra om II eller III ligger närmast den önskade linjen.
Avståndet mellan II och III är ett steg och vi kan således testa om
$y'_3 - y'_L < 0{,}5$ i vilket fall man går till III eller om
$y'_3 - y'_L \geq 0{,}5$ varvid man går till II.
Ur fig. 2 fås att riktningskoeff. C för linjen = $C = \frac{CNTY}{CNTX}$
Linjens ekv. blir då: $y'_L = C \cdot x'_L$ och för varje steg vi tar i x-led beräknas y'_L genom denna ekvation.

CNTX ger ett mått på hur långt linjen sträcker sig (se sid. 3) och detta utnyttjas för att få en räknare som talar om då vi nått slutpunkten. Räknaren består av en term CNTX' = CNTX - x_L' som minskas med ett för varje steg vi tar i x-led. Om vi dessutom tar ett steg i y-led ökas en annan term, CNTY' = y_3' - CNTY, med ett. Följaktligen finns varken x_L' eller y_3' tillgängliga i programmet utan endast CNTX' och CNTY', varför testet får följande utseende:

$$C \cdot CNTX' + CNTY' - 0{,}5 \begin{cases} \geq 0 \text{ , gå till II} \\ \text{eller} \\ < 0 \text{ , gå till III} \end{cases}$$

$$\Longleftrightarrow$$

$$\frac{CNTY}{CNTX}\,(CNTX - x_L') + y_3' - CNTY - 0{,}5 \Longleftrightarrow$$

$$\Longleftrightarrow$$

$$CNTY - C \cdot x_L' + y_3' - CNTY - 0{,}5$$

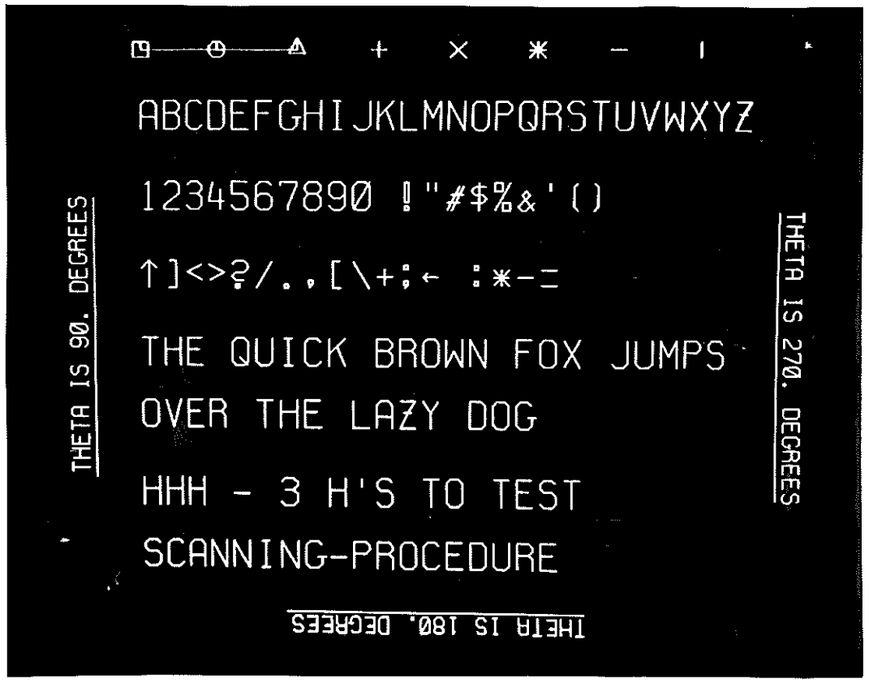
$$\Longleftrightarrow -y_L' + y_3' - 0{,}5$$

Räknaren CNTX' talar om då vi nått slutpunktens x-koordinat, IX.
Genom att vi för varje steg,genom testet ovan,har hållit oss så nära den önskade linjen som möjligt, borde vi därmed också ha uppnått slutpunktens y-koordinat, IY. Detta blir dock icke alltid fallet, beroende på att linjens uträknade riktningskoeff., C, inte är en exakt storhet utan erhållen genom divisionen CNTY/CNTX.
Man tar inte hänsyn till den rest man får vid divisionen och C är alltså alltid något för liten. Då C multipliceras med x_L' förstoras felet och y_L' (= $C \cdot x_L'$) kan därmed bli så mycket mindre (mer än 0,5 steg) att vi hamnar i en slutpunkt som är ett y-steg mindre än den vi önskade. För att råda bot på detta måste man, då CNTX' har angett att man är klar, testa om man skall gå ytterligare ett steg i y-led för att nå fram till IY.
Som vi såg ovan ändrades CNTY' med ett för varje steg vi tog i y-led, och CNTY' kan därför användas som räknare i y-led.
Då vi har uppnått den riktiga slutpunkten IX/IY, sättes IXO och IYO (se sid. 3) lika med IX resp. IY eller lika med noll beroende på om IPEN var positiv eller negativ. Därefter är subrutinen klar för ett nytt anrop.

SYMBOL

SYMBOL är en subrutin som ger möjlighet att på skärmen rita de 71 bokstäver, siffror och andra tecken som finns på bild I.

Bild I



(Bild I är ett resultat av testprogrammet SYMLIB, se bil. 7)

De översta tecknen på bild I är tecken som användes i subrutinen LINE för att särskilt utmärka vissa mätpunkter. För att få utskrivet ett sådant tecken krävs följande anrop i fortran:

CALL SYMBOL (IX,IY,HEIGHT,THETA,IBCD,K,MARK)

Härvid ritas ett av MARK specificerat tecken med sin mittpunkt i koordinaten IX,IY och storleken, uttryckt i steg på skärmen (ett steg ung. 0,2 mm), bestämd av HEIGHT.
K anger om linje skall dras från pennans föregående position till IX,IY (som tecken 2 och 3 på bild I) eller enbart tecknet skall ritas ut.
THETA och IBCD användes ej.

För att få utskrivet en följd av de övriga tecknen i bild I måste i en formatsats i fortranprogrammet specificeras vad som skall skrivas. I formatsatsen användes därvid H-format.
For att programmet SYMBOL skall kunna leta rätt på formatsatsen måste en variabel anta värdet av formatsatsens absoluta adress. Detta kan i fortran ordnas med en ASSIGN-sats och ett anrop på SYMBOL enligt denna form kommer då att få utseendet:

```
      ASSIGN 15 TO IFMT
            :
            :
      CALL SYMBOL (IX,IY,HEIGHT,THETA,IFMT,n,MARK)
            :
            :
            :
15 FORMAT (nH ABCDEF ......)
```

ASSIGN-satsen medför i exemplet ovan att IFMT antar värdet av den absoluta adressen till den sats som i fortranprogrammet har satsnumret 15.

Vid utskriften är IX,IY koordinaten för nedre vänstra hörnet, betraktat i skrivriktningen, av det första tecknet.
HEIGHT anger som tidigare storleken av tecknet uttryckt i antal steg på oscilloskopskärmen.
THETA anger skrivriktningens vinkel i förhållande till horisontalriktningen.
I bild I visas resultatet av att THETA är 90, 180 och 270 grader.
I bild II visas resultatet när THETA är 0, 30, 60 .....330 grader.

Bild II

Bild II är ett resultat av test programmet JOL,se bil. 8.

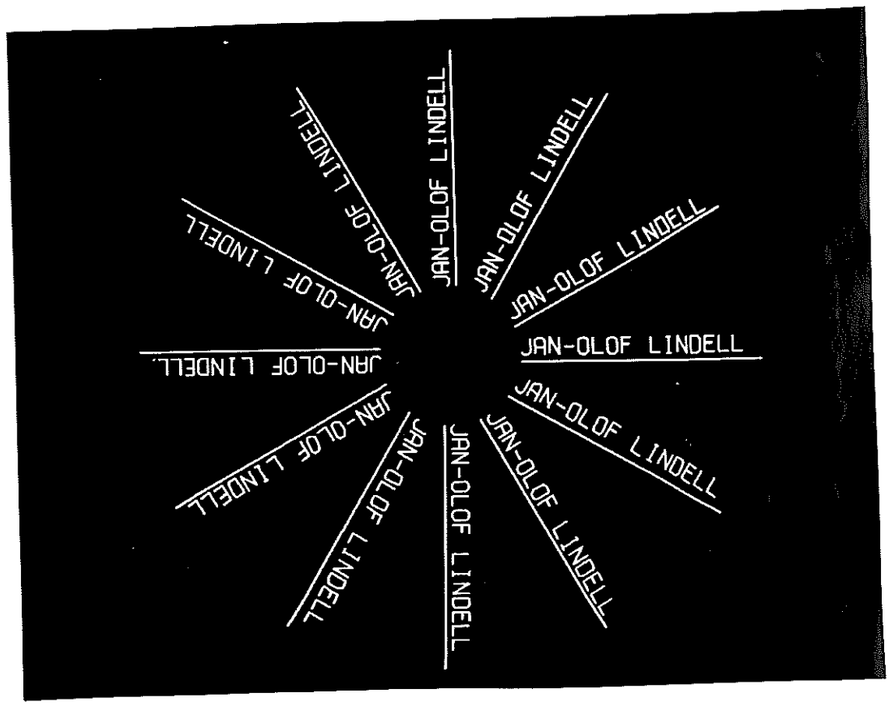


Fig. 6

APPENDIX A

CHARACTER SET

Printing Character	7-bit ASCII	6-bit Trimmed ASCII	Printing Character	7-bit ASCII	6-bit Trimmed ASCII
@	100	00	(Space)	040	40
A	101	01	!	041	41
B	102	02	"	042	42
C	103	03	#	043	43
D	104	04	$	044	44
E	105	05	%	045	45
F	106	06	&	046	46
G	107	07	'	047	47
H	110	10	(	050	50
I	111	11	)	051	51
J	112	12	*	052	52
K	113	13	+	053	53
L	114	14	,	054	54
M	115	15	-	055	55
N	116	16	.	056	56
O	117	17	/	057	57
P	120	20	0	060	60
Q	121	21	1	061	61
R	122	22	2	062	62
S	123	23	3	063	63
T	124	24	4	064	64
U	125	25	5	065	65
V	126	26	6	066	66
W	127	27	7	067	67
X	130	30	8	070	70
Y	131	31	9	071	71
Z	132	32	:*	072	72
[*	133	33	;	073	73
\	134	34	<	074	74
]*	135	35	=	075	75
↑*	136	36	>	076	76
←*	137	37	?	077	77
Null	000				
Horizontal Tab	011				
Line Feed	012				
Vertical Tab	013				
Form Feed	014				
Carriage Return	015				
Rubout	177				

Notes:

(1) All other characters are illegal to MACRO-9 and are flagged and ignored.
(2) * = Illegal as source, except in a comment or text.

Bild II visar också tydligt, att vid små bokstäver och vid vissa vinklar uppstår det avrundningsfel vid beräkningen av de vinklade koordinaterna. Dessa fel gör sig märkbara genom att bokstäverna förlorar sin rätta form. Detta blir ju dock mindre märkbart ju större bokstäver man använder, eftersom det största fel man gör sig skyldig till vid varje avrundning är mindre än 1/2 steg.
n anger antal tecken i formatsatsen, som skall skrivas ut.
MARK har här ingen betydelse.

För att åskådliggöra SYMBOL:s funktionssätt är det lättast att arbeta med ett specifikt exemple. Vi antar att vi vill skriva "SYMBOL" på skärmen. Formatsatsen skall då ha följande utseende: FORMAT (6HSYMBOL).
Då denna sats kompileras bildas först en hoppsats, eftersom en formatsats ej skall utföras. Värdet på IFMT är alltså adressen till denna hoppsats. Först i nästa ord, d.v.s. i ordet med adressen IFMT+1 ligger den första delen av innehållet i formatsatsen lagrat. I en formatsats användes en representation som kallas ASCII-kod och som använder 7 bitar för att representera ett tecken. Tecknen ligger lagrade så att 5 tecken lagras i 2 18-bits ord. I fig. 6 finns ASCII-koden för de tecken som används i SYMBOL. Med ledning av fig. 6 får vi följande lagring av innehållet i den formatsats som vi använder som exempel:

Fig. 7

1:a ordet

	(			6			H		Tecken	
	0	5	0	0	6	6	1	1	ASCII	Oktalt
	0	1 0 1	0 0 0	0	1 1 0	1 1 0	1	0 0 1		Binärt

2:a ordet

		S			Y			
	0	1	2	3	1	3	1	
	0 0 0	1	0 1 0	0 1 1	1	0 1 1	0 0 1	0

3:e ordet

	M			B			O	
	1	1	5	1	0	2	1	1
	1	0 0 1	1 0 1	1	0 0 0	0 1 0	1	0 0 1

4:e ordet

		L			)			
	7	1	1	4	0	5	1	
	1 1 1	1	0 0 1	1 0 0	0	1 0 1	0 0 1	0

Orden ovan hämtas ett åt gången från sin lagringsplats och varje tecken maskas ut och omvandlas omedelbart till den 6-bits ASCII-kod som finnes i fig. 6. Detta går utmärkt eftersom 6 bitar räcker till för att representera de tecken vi använder i programmet SYMBOL.
De tecken som föregår "SYMBOL" ovan skall inte skrivas ut på skärmen. Man letar alltså alltid först efter det första H:et som finns i formatsatsen. Alla tecken som kommer efter detta skall skrivas på skärmen.

I programmet finnes en tabell (PLTBL) som för vart och ett av de möjliga tecknen innehåller en beskrivning på hur tecknet skall ritas.
Vår uppgift är nu att förflytta oss till den punkt i tabellen där beskrivningen av det första tecknet, i vårt exempel "S", börjar. För den skull finns i programmet ytterligare en tabell (ADTBL), som talar om var i PLTBL beskrivningen av vårt "S" börjar.
För att komma rätt i ADTBL användes tecknets 6-bits ASCII-kod. För "S" är denna 23 (okt). I den oktala adressen ADTBL + 23 hittar vi alltså en hänvisning till var i PLTBL beskrivningen av "S" börjar.

I ADTBL + 23 finner vi: 500044
44 innebär att beskrivningen av "S" börjar i adressen PLTBL+ 44 (okt).
50 i början av ordet är ett mått på antalet koordinater som behövs för att beskriva bokstaven "S". Antalet koordinater (d.v.s. oktala siffror) är här 100-50 (okt) d.v.s. 24 st. (dec).

Om vi letar upp adressen PLTBL + 44 hittar vi följande: 463717
I beskrivningen har man utgått ifrån att HEIGHT = 7 steg. Med detta antagande skall gälla att varje tecken skall rymmas i en rektangel av storleken 4 steg x 7 steg. Siffrorna 463717 beskriver hur man skall flytta pennan inom denna rektangel för att erhålla det önskade tecknet, i detta fallet "S". Siffrorna är omväxlande x-koordinaten och y-koordinaten, så att ett ord bygges upp enligt följande mönster: $X_0Y_0X_1Y_1X_2Y_2$. Till X_0/Y_0 går man med pennan uppe. Därefter fortsätter man till de övriga koordinaterna med pennan nere. 463717 ger då följande resultat (fig. 8a):

Fig. 8a

X_0=4 Y_0=6
X_1=3 Y_1=7
X_2=1 Y_2=7

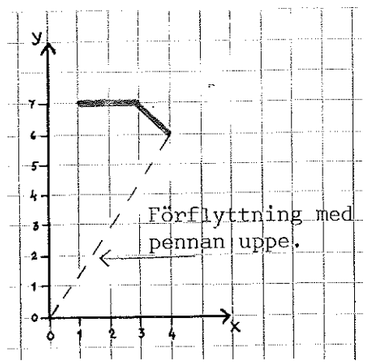
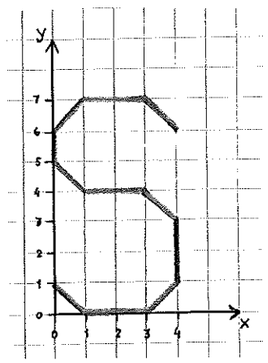


Fig. 8b



Därefter krävs ett nytt ord för att fortsätta beskrivningen. De följande orden är:

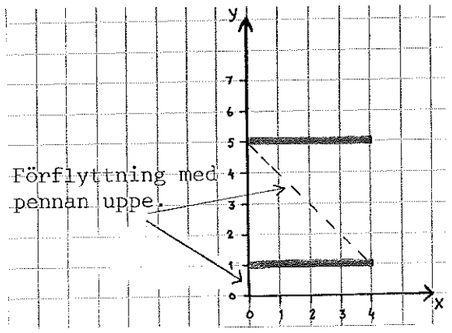
```
PLTBL + 45 : 060514
      + 46 : 344341        (se fig. 8b)
      + 47 : 301001
```

Beskrivningen fortsätter i detta exempel tills 24 (dec) oktala siffror har avverkats.

I vårt exempel "SYMBOL" behöver man aldrig lyfta pennan då man ritar en bokstav. Det finns dock tecken i programmet, som kräver en förflyttning av pennan i upplyft läge. Ett exempel är likhetstecken, " = ". För att beteckna att pennan skall lyftas för ett koordinatpar, användes beteckningen 77. Beskrivningen av likhetstecknet har följaktligen utseendet: 0 1 4 1 7 7 0 5 4 5 0 0

$X_0Y_0X_1Y_1X_2Y_2$ $X_3Y_3X_4Y_4X_5Y_5$

Fig. 9



Varje koordinat, som plockas fram i beskrivningen kan, innan den ritas ut på skärmen, omvandlas med hänsyn till värdena på argumenten HEIGHT och THETA. Då koordinaterna är slutgiltigt bestämda, användes programmet PLOTTA för själva ritandet.
Härefter återstår bara att flytta pennan till utgångspunkten för nästa bokstav. Denna skall rymmas i en rektangel, som, om vi fortfarande antar att HEIGHT = 7, skall befinna sig på 6 stegs avstånd (i skrivriktningen) från "S" - rektangeln.

Utskriften fortsätter tills, i vårt exempel, 6 st. bokstäver har ritats. Det är argumentet n, som håller räkning på antalet tecken, som skall skrivas, så även om formatsatsen innehåller fler tecken, så skrives endast n st. ut.

De första tecknen i bild I, de som användes av programmet LINE, har en speciell plats längst bak i ADTBL och den exakta adressen i ADTBL styres av värdet på argumentet MARK. En annan skillnad därvid är, att pennan efter ritandet av ett sådant tecken, stannar i mittpunkten på det ritade tecknet, för att möjliggöra att man nästa gång pennan flyttas skall kunna få en linje dragen därifrån, som fallet är med de 2 första tecknen i bild I.

NUMBER

En variabel, vars värde representeras i fortran med flytande räkning kan med programmet NUMBER skrivas ut på skärmen, korrekt avrundat, i sitt decimala värde. I Reglerteknikks programbibliotek har detta program senare fått vika för ett program PLTNUM, skrivet av Johan Wieslander, som skriver ut värdet av variabeln i E- eller F-format (för definition av E- och F-format se en fortranmanual) istället. Detta senare program användes av programmet AXIS för att skriva ut värden på koordinataxlarna. Därvid användes E-formatet, och den subrutin som skriver i detta format kallas OFNEF. Här beskrives dock endast programmet NUMBER.

NUMBER har följande argument:

NUMBER (IX,IY,IHGT,THETA,FPN,NODD)

NUMBER använder sig vid ritandet av siffrorna av programmet SYMBOL och de fyra första argumenten, IX,IY,IHGT,THETA är desamma som återfinnes först i SYMBOL. FPN är den variabel vars värde skall skrivas på skärmen och NODD anger hur många decimaler som skall medtagas vid utskriften.

Som nämndes ovan använder NUMBER programmet SYMBOL för utskriften av siffrorna. Men, i SYMBOL specificeras det som skall skrivas i en formatsats.
För att kunna använda SYMBOL måste man alltså skaffa fram 7-bits ASCII-koderna för de i talet ingående siffrorna och lagra dessa med 5 siffror i 2 ord, som förhållandet är i en formatsats.

Talet som skall skrivas på skärmen är lagrat med flytande räkning. För att få fram de decimala siffrorna multipliceras eller divideras talet med 10 tills heltalssiffran ligger mellan 0 och 10. En räknare håller reda på hur många gånger man har dividerat resp. multiplicerat med 10 för att talets ursprungliga värde ej skall gå förlorat. Därefter användes ett systemprogram, .AX (el. FIX) som omvandlar heltalsdelen av ett tal i flytande räkning till fix räkning. Resultatet blir alltså ett tal mellan 0 och 10 som utgör det ursprungliga talets första signifikativa siffra.

Denna siffra subtraheras därefter bort, återstoden multipliceras med 10, programmet FIX användes och vi har fått fram nästa siffra.

Om det ursprungliga talet var större än 1 håller den förut omtalade räknaren reda på hur många heltalssiffror det finns. Då dessa har plockats fram skall det sättas en decimalpunkt och därefter fortsätter framplockningen av siffror så långt som argumentet NODD anger.

Om det ursprungliga talet var mindre än 1 måste heltalssiffran vara en 0. Därefter skall sättas en decimalpunkt och sedan användes räknaren för att hålla reda på hur många nollor som skall föregå den första signifikativa siffran. Även i detta fall reglerar argumentet NODD hur många siffror som skall plockas fram.

När siffrorna på ovan beskrivet sätt har plockats fram skall de omvandlas till ASCII-form och därefter i tur och ordning lagras i en formatsats. Som framgår i beskrivningen av SYMBOL måste emellertid först i formatsatsen ligga ett H.
Självklart måste man också ange om talet är positivt eller negativt. Efter H:et kommer följaktligen ett tecken. Därefter lagras heltalssiffrorna, en decimalpunkt och decimalsiffrorna.
Eftersom 5 tecken skall lagras i 2 ord enligt den mönster som beskrives i fig. 7 i SYMBOL, krävs ett antal räknare för att hålla reda på i vilka positioner de olika siffrorna skall lagras.

Ett tal i flytande räkning kan anges med en noggrannhet av högst 6 decimala siffror. NUMBER tillser att aldrig mer än 6 signifikativa siffror skrives ut. Skulle fler siffror önskas fylles automatiskt på med nollor. NUMBER har också en gräns för heltalsdelen för talet. Överstiger heltalsdelen 8 siffror skrives 8 siffror ut följt av tre asterisker för att markera att talet var för stort.

Avrundningen i NUMBER sker så att talet 5 adderas till den siffra i talet, som står omedelbart efter den sista signifikativa siffran som skall skrivas ut. Som exempel visas nedan i tabell hur NUMBER behandlar olika tal.

Fig. 10

Tal (dec.) värde	Värde på argumentet NODD	Utskrift med NUMBER	Kommentar
123,456	2	+123.46	Avrundning efter sista siffran angiven av NODD.
12345,6	3	+12345.600	Avrundning efter 6 sign. siffror.
$4{,}56789 \cdot 10^{10}$	1	+45678900***	Heltalsdelen för stor.
-456,78	0	-457.	Inga decimaler.
-456,78	-1	-457	Ingen decimalpunkt.

LINE

Programmet LINE kan sägas vara en praktisk tillämpning av PLOTTA. LINE plockar ur en x-vektor och en y-vektor ut ett antal x/y-koordinater och drar räta linjer mellan dem och/eller märker ut vissa av dem med speciella markeringar. De speciella markeringarna är de som omtalas i SYMBOL och som återfinnes överst på bild I, sid. 7.

LINE har följande argument:

LINE (IX(*),IY(*),HEIGHT,N,K,J,MARK)

IX(*) resp. IY(*) är det första elementet ur x- resp. y-vektorn, som skall plottas. Elementen i vektorerna måste vara heltal och hålla sig inom de begränsningar som betingas av PLOTTA.
HEIGHT är storleken på de eventuella markeringar som skall utmärka vissa punkter.
N är antalet punkter som skall plottas.
K är intervallet mellan de element i vektorerna som skall användas. Vart K:e element plottas.
Argumentet J bestämmer om linje skall dras mellan punkterna eller ej, samt hur ofta punkter skall speciellt markeras med ett av MARK specificerat tecken.
Om J = M drages en linje mellan varje punkt och vid var M:e punkt sättes en speciell markering av en typ som anges av argumentet MARK.
Om J = O drages en linje mellan punkterna, men inga speciella markeringar sättes ut.
Om J = -M drages ingen linje mellan punkterna, men vid var M:e punkt sättes en speciell markering ut.

Funktionen av LINE är enkel. Den består i huvudsak av ett antal räknare, som håller reda på de olika egenskaper, som ovan beskrivits. Själva grovgörat sköts därefter genom anrop av subrutinerna PLOTTA och SYMBOL.

Bilderna III, IV, V och VI illustrerar vad som händer då ett par vektorer, IX och IY, skall plottas och argumenten HEIGHT,K,J och MARK varieras. Bilderna är ett resultat av testprogrammet LISS (se bil. 9). Programmet ritar en Lissajou-figur. Origo har lagts mitt på skärmen och vektorerna IX och IY innehåller vardera 361 element med följande utseende:

$$IX(A) = \frac{1000}{3} \cdot \sin (A-1)$$

$$A = 1,2,3.....361$$

$$IY(A) = \frac{1000}{3} \cdot \sin 2(A-1)$$

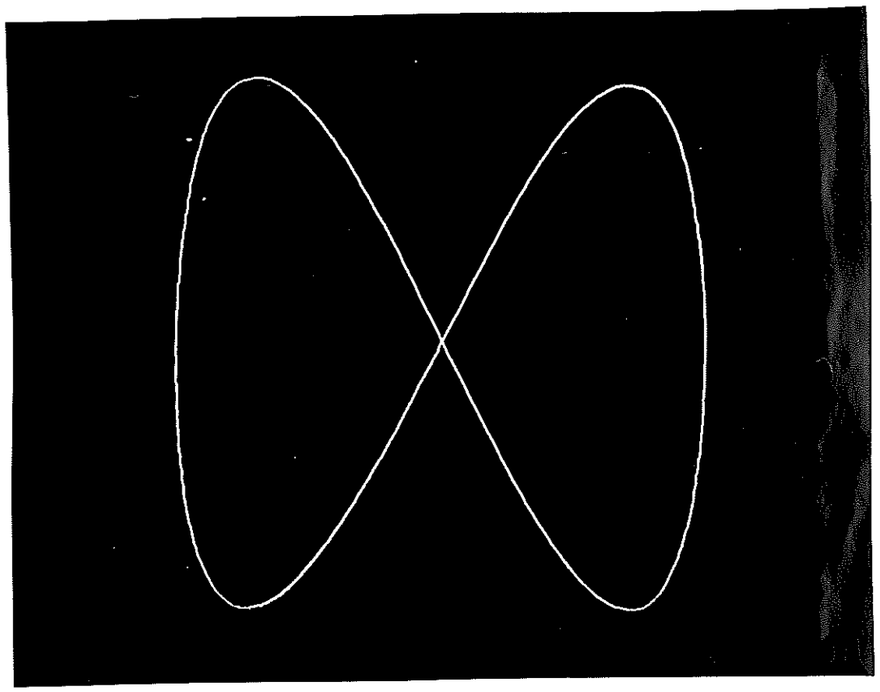


BILD III

HEIGHT=(21) ingen betydelse

N=361
K=1
J=0
MARK=(2) ingen betydelse

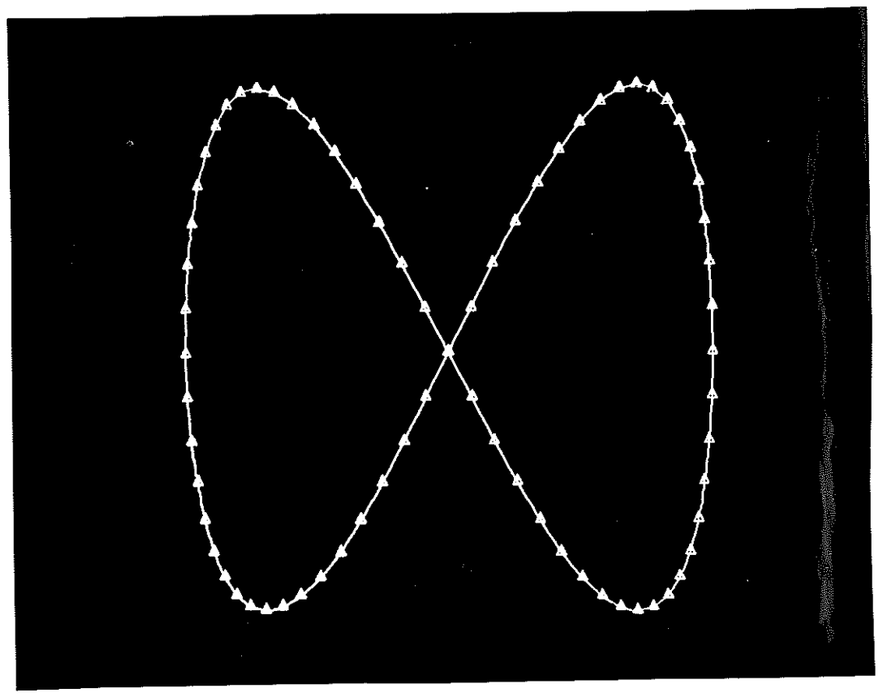


BILD IV

HEIGHT=21
N=361
K=1
J=5
MARK=2

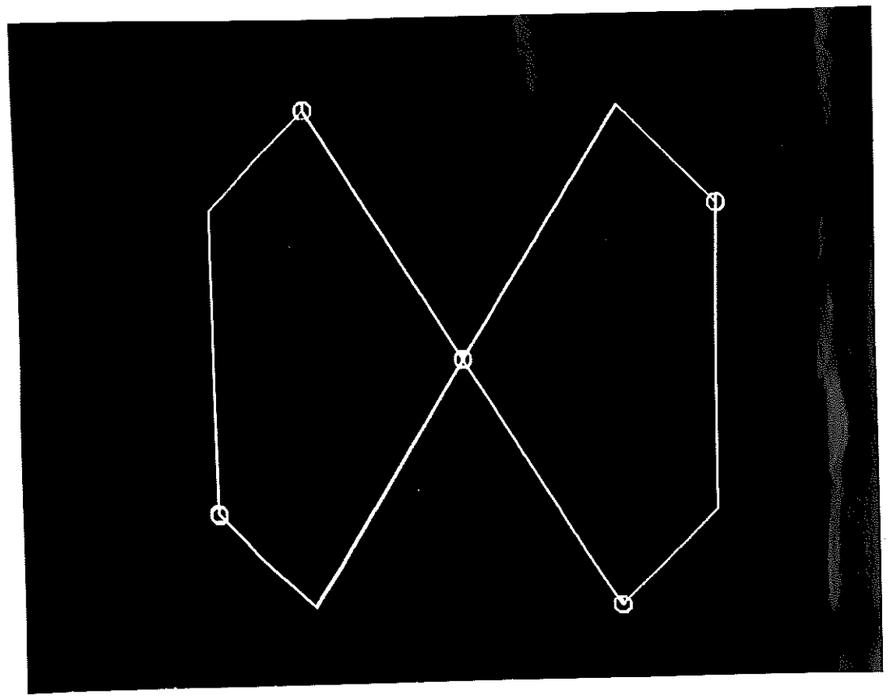


BILD V

HEIGHT=35
N=11
K=36
J=2
MARK=1

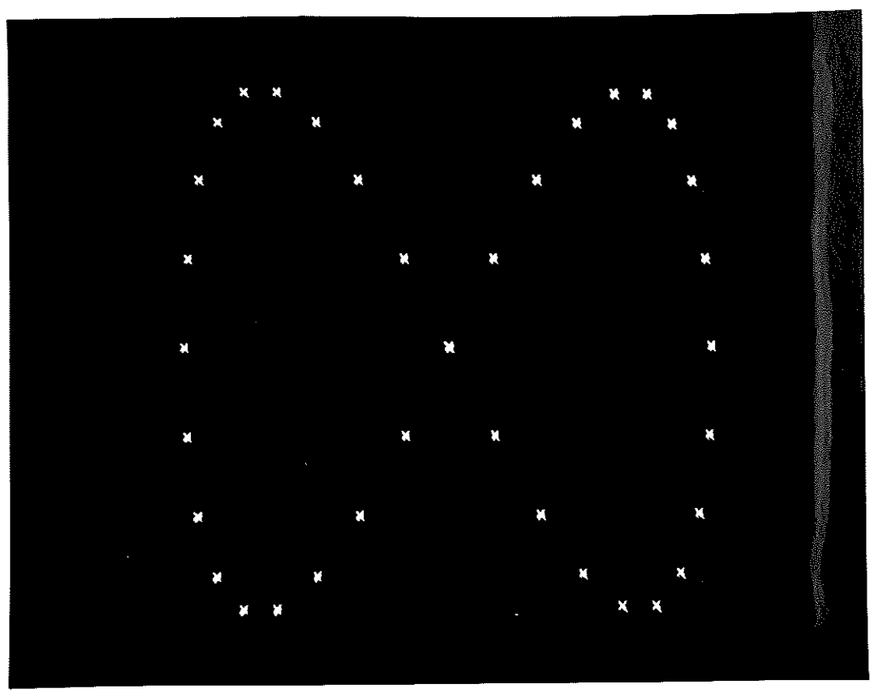


BILD VI

HEIGHT=14
N=361
K=1
J=-10
MARK=4

eller

HEIGHT=14
N=37
K=10
J=-1
MARK=4

eller

HEIGHT=14
N=73
K=5
J=-2
MARK=4

eller

HEIGHT=14
N=181
K=2
J=-5
MARK=4

SCALE

Som framgick av beskrivningen av LINE måste de vektorer som skall plottas med subrutinen LINE innehålla element, som är heltal mellan 0 och ung. 1000. Ibland kanske man inte vill använda hela skärmen för att rita en kurva och då begränsas elementens utseende ytterligare. Om värdena inte ligger inom dessa begränsningar eller inte är heltal måste man skala om dem så att de passar.

SCALE är ett program, som, med kännedom om max-och minvärdena av en vektors element, skalar dessa så att de passar inom givna begränsningar och därefter omvandlar dem till heltal. För att få fram max-och minvärdena av en vektor kan man använda en subrutin, MINMAX, som också ingår i Reglertekniks program-bibliotek och är författad av Johan Wieslander.

SCALE använder följande argument:

SCALE (U(*),IU(*),NPOI,W,UMIN,UMAX,SMIN,DS)

Av dessa argument är IU(*), SMIN och DS namn på de variabler som SCALE använder för att lagra resultatet av skalningen i.

U(*) är det första elementet i den vektor som skall skalas.
IU(*) är motsvarande element i den färdigskalade och heltalsomvandlade vektorn.
NPOI är antalet värden som skall skalas.
W är tillgängligt utrymme på skärmen, uttryckt i centimeter.
UMIN är det minsta värdet som finns i vektorn U och UMAX är det största.
DS är namnet på den variabel där SCALE lagrar inverterade värdet av den skal-faktor, som användes vid skalningen av U. DS har dimensionen värde/cm och användes när man skall sätta ut siffervärden på den koordinataxel, som vanligen åtföljer en ritad kurva. DS är alltså ett av de argument som AXIS behöver för att kunna rita en riktig koordinataxel.

Även i SMIN lagras ett värde, som beräknas för att användas i AXIS. Det är ett värde, som är ett heltal • DS och sådant att SMIN < UMIN. Heltalet väljes så, att SMIN ligger så nära UMIN som möjligt. Det användes som lägsta siffervärde att skrivas på koordinataxeln.

Huvuduppgiften för subrutinen SCALE är att få fram en lämplig skalfaktor till den vektor, som skall skalas. Hur skall då en skalfaktor vara beskaffad för att vara lämplig?

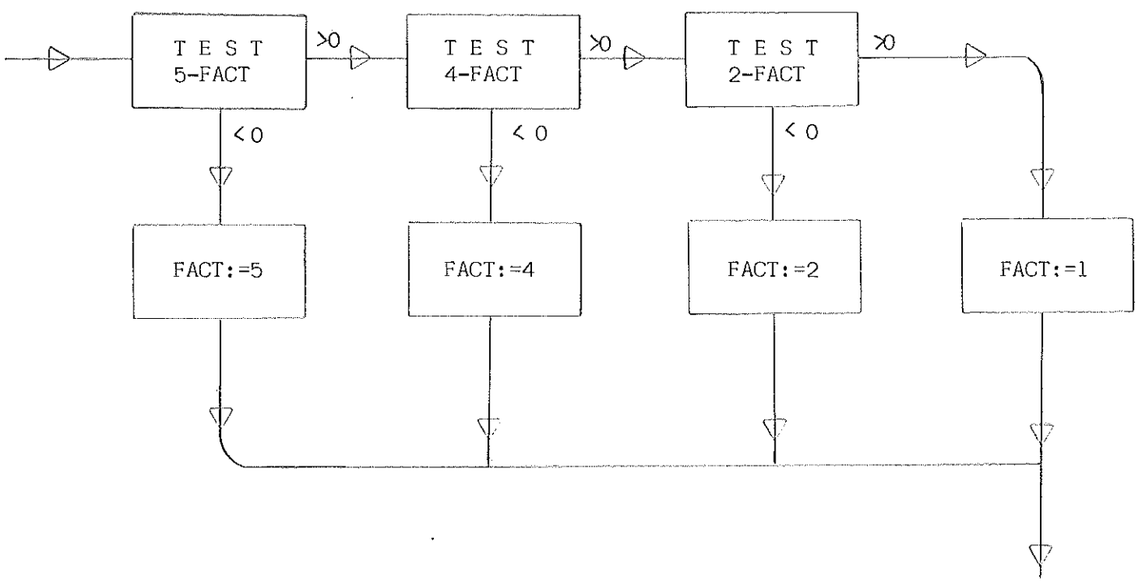
Först kan konstateras att skalfaktorn bör vara så stor som möjligt, för att man optimalt skall utnyttja det utrymme som står till förfogande för uppritningen. Största teoretiska skalfaktorn får man som resultat, om man dividerar tillgängligt utrymme, W, med det värdesintervall, UMAX-UMIN, som skall plottas. Denna skalfaktor är dock olämplig av den anledningen att DS får ett otympligt värde, som försvårar omvandlingen av uppmätta längder på skärmen till vektorn U:s ursprungsvärden.

Inom rittekniken har man standardiserat vissa faktorer för dylika omvandlingar. Dessa standardiserade faktorer är 1, 2, (2,5) och 5. (2,5 är inte internationellt standardiserad.)
De skalfaktorer som motsvarar dessa omvandlingsfaktorer är 1,5, (4) och 2.
Naturligtvis duger alla skalfaktorer som erhålls ur de ovanstående genom en multiplikation med ett godtyckligt antal tiopotenser.
Som skalfaktor bör man således välja den av ovanstående faktorer som ligger närmast under den teoretiskt störst beräknade skalfaktorn.

SCALE börjar med att beräkna uttrycket FACT = $\frac{W}{UMAX-UMIN}$

FACT omvandlas därefter till ett tal mellan 0 och 10 medan en räknare håller reda på den ursprungliga storleken, d.v.s. antalet tiopotenser.
FACT jämföres därefter med de standardiserade värdena på skalfaktorer enligt nedanstående figur.

Fig. 11



Efter att FACT sålunda fått ett standardiserat värde, multipliceras FACT med det antal tiopotenser som FACT ursprungligen hade och som ovan nämnda räknare håller reda på.

DS beräknas genom sambandet DS = $\frac{1}{FACT}$ och lagras.

UMIN divideras med DS, resultatet avrundas nedåt till heltal och multipliceras därefter åter med DS. Vi erhåller på så sätt ett värde som är mindre än UMIN och lika med ett heltal ·DS. Detta värde lagras i SMIN.
Härefter återstår bara omvandlingen av elementen i vektorn U.
Varje element, U(n), minskas med UMIN. Resultatet multipliceras med skalfaktorn FACT, multipliceras med en faktor, SPC, som anger antalet steg på skärmen per cm, omvandlas till heltal och lagras som elementet IU(n) i vektorn IU.

AXIS

Till varje kurva som ritas på ett xy-plan bör det finnas två koordinataxlar. Dessa bör ge upplysningar om

1) vad kurvan beskriver, d.v.s. vilken variabel som representeras i x- resp. y-riktningen,
2) vilken sort dessa variabler beskrives i samt
3) vilken skala som använts vid uppritningen.

AXIS är ett program som ritar en rät linje, en axel, i godtycklig riktning. På den räta linjen sättes på varannan centimeter ett litet skalstreck. Vid vartannat skalstreck utsättes ett siffervärde (i E-format). Vid slutet av axeln skrives, om så önskas, en valfri text som måste specificeras i en formatsats. Slutet på texten ligger i jämnhöjd med slutet på axeln. Texten skrives på motsatt sida om axeln i förhållande till skalstreck och siffror.

AXIS argument är:

AXIS (IX,IY,SL,IFMT,N,THETA,SMIN,DS)

IX och IY är koordinaterna för början av axeln.
SL:s absolut-värde anger axelns längd i cm. Om SL > 0 sättes skalstreck och talvärden på högra sidan av axeln, som vanligt är vid en horisontell axel. Den eventuella texten skrives på vänster sida om axeln. Om SL < 0 bytes placeringarna, så att skalstreck och talvärden sättes på vänster sida om axeln, som brukligt är för vertikala axlar, och texten på höger sida.
IFMT och N är argument, som användes då AXIS anropar SYMBOL för att skriva ut text på axeln. Betydelsen av dessa argument framgår i beskrivningen av SYMBOL. Om N=0 skrives ingen text ut.
THETA anger axelns lutningsvinkel i grader i förhållande till en horisontell axel.
SMIN är det siffervärde som skall skrivas vid det första skalstrecket på axeln.
DS anger värde/cm längs axeln och varje siffervärde, som skrives ut, ökas med 4 * DS från föregående tal. Avståndet mellan varje siffervärde på axeln är nämligen 4 cm.
SMIN och DS kan beräknas av SCALE.

För att räkna ut axelns ändpunkter måste först en del andra beräkningar göras.
Axelns längd måste omvandlas från cm till antal steg, LENG = SL · SPC.
SPC = steps per centimeter - för denna skrivare = 50.
Dessutom måste man ha tillgång till SINTH = sin (THETA) och COSTH = cos (THETA).
Eftersom multiplikation med sin (THETA) och cos (THETA) användes många gånger inom AXIS finnes två speciella små subrutiner i AXIS, MULSIN och MULCOS, som har hand om detta.
Axelns ändpunkter, X2/Y2, beräknas nu genom sambanden

$X2 = IX + LENG \cdot COSTH$

$Y2 = IY + LENG \cdot SINTH$

och axeln ritas ut genom ett anrop av PLOTTA.

Koordinaterna för skalstreckens start- och ändpunkter beräknas på liknande sätt.
Startpunkten, XSC/YSC, beräknas genom sambanden

$XSC = IX + NSM \cdot COSTH$

$YSC = IY + NSM \cdot SINTH$

där NSM är avståndet i steg från axelns startpunkt till startpunkten för skalstrecket. För varje skalstreck som ritas ökas NSM med 100 (motsvarande 2 cm).
Skalstrecken skall vara 4 steg höga och ändpunkternas koordinater fås därför genom sambanden

$XM = XSC \pm 4 \cdot SINTH$

$YM = YSC \mp 4 \cdot COSTH$

där tecknen framför $4 \cdot SINTH$ resp. $4 \cdot COSTH$ avgöres av, om skalstrecket skall ligga till höger eller till vänster om axeln. Detta bestämmes i sin tur av tecknet på argumentet SL.
Skalstrecket ritas genom anrop av PLOTTA.

Då ett skalstreck har ritats testas en räknare för att avgöra om ett tal skall skrivas ut.
Det första siffervärde, som skall skrivas och alltså skrivas vid det första skalstrecket, är det som anges av argumentet SMIN. Siffror skall skrivas vid vartannat skalstreck, d.v.s. vid var 4:e cm. Sambandet

$SCNUM = SMIN + (n-1) \cdot 4 \cdot DS$

kan således användas för att bestämma värdet på det tal som skall skrivas ut.
Talet skall ha ett mellanrum till skalstrecket på 4 steg och, eftersom siffrorna är 14 steg höga, hamnar nedre vänstra hörnet av första siffran 14 + 4 = 18 steg under skalstreckets ändpunkt eller 4 steg över skalstrecket, beroende på vilken sida om axeln skalstreck och siffror skall ligga.

Denna koordinat justeras med avseende på THETA på samma sätt som då skalstreckets ändpunkt beräknades. Siffervärdet skrives sedan ut i E-format genom anrop av subrutinen OFNEF (en reviderad version av NUMBER, se beskrivning av NUMBER). Det sista skalstrecket på axeln ritas alltid utan siffervärde.

Texten skall skrivas med en höjd av 14 steg. Varje tecken har då (se SYMBOL) en bredd av 6/7 • 14 = 12 steg och textens längd, FL, beräknas som
FL = N •12 steg.
Genom att räkna baklänges från axelns ändpunkt och med hänsyn till att texten skall skrivas längs axeln och på 4 stegs avstånd från densamma, kan man räkna ut textens startkoordinater med hänsyn tagen till THETA, på samma sätt som då siffervärdet skulle skrivas ut. Enda skillnaden är att texten skrives på motsatt sida om axeln i förhållande till siffrorna.
Nu har vi fått fram alla de argument som SYMBOL använder för att skriva text och texten kan skrivas ut.

— — — — — — — — — — — — — —

För att demonstrera användningen av de olika programmen har jag skrivit ett demonstrationsprogram i fortran, DEMEX (se bil. 10), som anropar samtliga ovan beskrivna subrutiner.

Bild VII illustrerar användningen av SYMBOL (och indirekt PLOTTA).
I bild VIII användes även AXIS (och indirekt SCALE och OFNEF).
Bild IX slutligen, illustrerar även LINE. Denna sista bild är ganska representativ för det sätt på vilket subrutinerna vanligen användes.

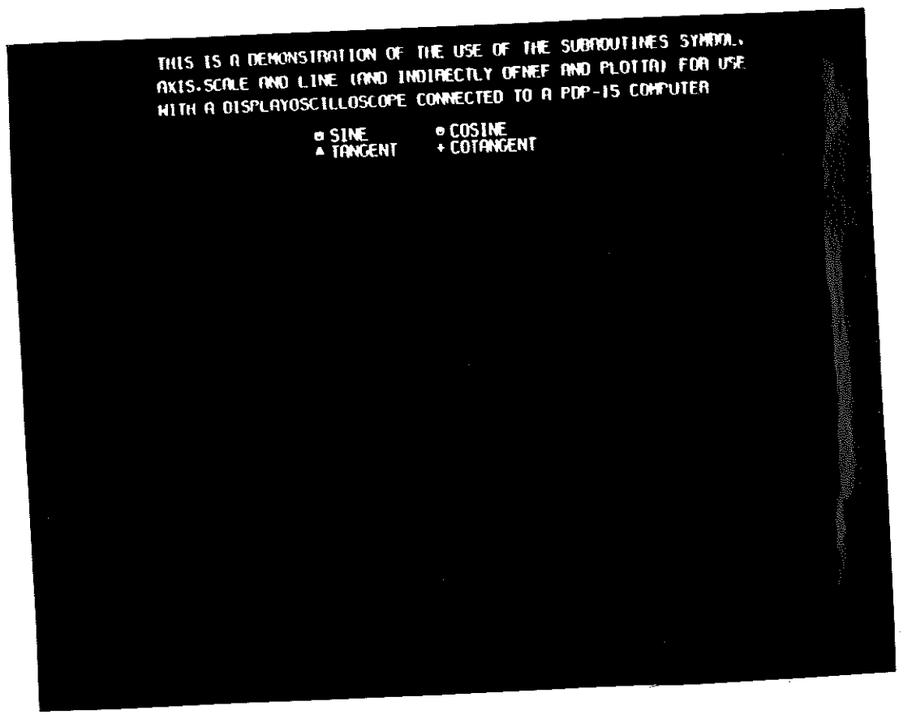


BILD VII

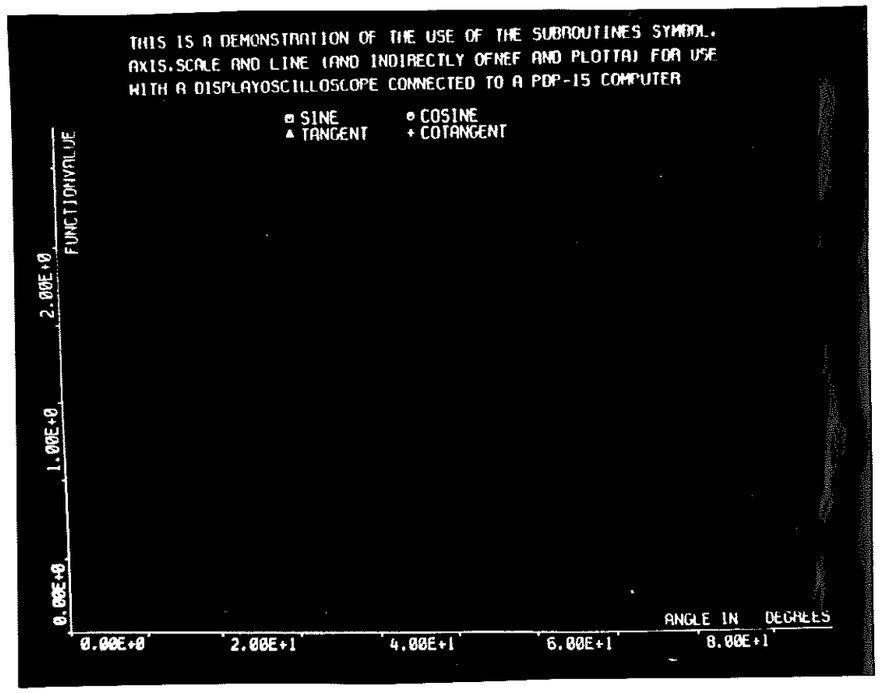


BILD VIII

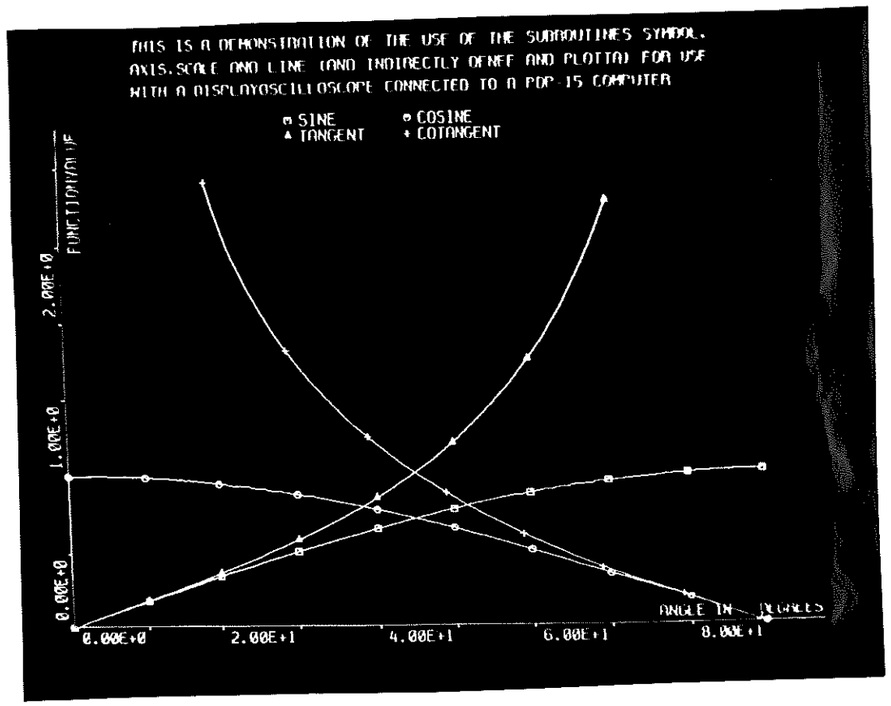


BILD IX

Slutligen bör det påpekas, att efter hand som arbetet med varje subrutin har blivit klart, har subrutinen införlivats med instutionens programbibliotek och börjat användas. Efter en tid har man kanske märkt, att vissa detaljer skulle bli mer ändamålsenliga för instutionen om de reviderades något, varför de i programbiblioteket ingående programmen inte helt överensstämmer med ovan beskrivna program. Detta innebär tyvärr också, att då ytterligare program har skrivits, dessa har måst baseras på de i programbiblioteket ingående reviderade programmen.

Ett exempel är PLOTTA (se bil. 1). I den ursprungliga versionen, som här beskrivits, kan IPEN bara anta värdena 1 och 2, medan man i t.ex. AXIS (se bil. 6) anropar PLOTTA med värdena 2 och 3 på IPEN. Ett annat exempel är LINE vars argument HEIGHT i programbiblioteket slopats och ersatts med en standardhöjd av 14 steg på alla markeringar.

I programmen, som beskrivits ovan, är det således icke helt säkert att de senare programmen kan anropa de tidigare så som beskrivits.

Alla ändringar som gjorts är dock av ringa omfattning och enkla att komma tillrätta med.

```
/       BILAGA 1
/
/
        .TITLE PLOTTA
/
/       AUTHOR J-O LINDELL OKT 1970
/
/
/       SUBROUTINE PLOTTA(IX,IY,IPEN)
/
/       PLOTS A STRAIGHT LINE TO A SPECIFIED POINT.
/       THIS IS THE BASIC PLOTTING ROUTINE.
/
/       IX   - X-COORDINATE OF THE ENDPOINT OF THE LINE.
/       IY   - Y-COORDINATE.
/       IPEN - SPECIFIES THE STATE OF THE PEN AND DEFINES
/              THE ORIGIN.
/              IABS(IPEN)=1: MOVE TO (IX,IY) WITH PEN DOWN.
/              IABS(IPEN)=2: MOVE TO (IX,IY) WITH PEN UP.
/              IPEN POSITIVE: THE ORIGIN IS NOT AFFECTED.
/              IPEN NEG.: THE ORIGIN IS MOVED TO THE LAST POINT.
/
/       - - - - - - - - - - - - - - - -
/
/       SUBROUTINES REQUIRED
/              NONE
/
        .GLOBL PLOTTA,.DA
SDFD=702601     /SKIP ON DEVICE FLAG
OMD=702621      /SET TO OUTPUT MODE
NOMD=702661     /SET TO NO OUTPUT MODE
LXD=702602      /LOAD X-REGISTER
LYD=702622      /LOAD Y-REGISTER, DISPLAY AND CLEAR FLAG
/               THE FLAG IS AUTOMATICALLY SET WHEN THE
/               PLOTTER IS READY TO RECEIVE A NEW PAIR OF
/               COORDINATES.
/
PLOTTA  0
        JMS* .DA
        JMP .+4
IXA     0
IYA     0
IPENA   0
        LAC* IXA
        DAC IXD
        LAC* IYA
        DAC IYD
        LAC* IPENA
        DAC IPEND
        SMA
        JMP PLOT1       /JUMP IF IPEN POSITIVE
        TCA
        DAC IPEN        /IPEN NOW POSITIVE
IPENN   LAC (DZM IXD    /IPEN WAS NEG. FROM THE BEGINNING
        SKP
PLOT1   LAC (SKP        /IPEN WAS POS. FROM THE BEGINNING
        DAC RESET
```

```
/       BILAGA 2
/
/
        .TITLE SYMBOL
/
/       AUTHOR J-O LINDELL JAN 1971
/       SUBROUTINE SYMBOL(IX,IY,IHGT,THETA,IFMT,N,MARK)
/
/       1) DRAWS A STRING OF CHARACTERS. THE CHARACTERS
/          ARE SPECIFIED IN HOLLERITH FORM IN A FORMAT
/          STATEMENT.
/
/       2) PLOTS A MARK. THE MARKS ARE USED TO MARK
/          SPECIAL DATA POINTS IN DIAGRAMS ETC.
/          A LINE MAY BE DRAWN FROM THE FORMER PEN
/          LOCATION TO IX/IY (2A) OR NOT (2B).
/
/       1),N>0:
/          IX    - THE X-COORDINATE OF THE LOWER LEFT
/                  CORNER OF THE FIRST CHARACTER TO BE
/                  PLOTTED.
/          IY    - THE Y-COORDINATE.
/          IHGT  - THE HEIGHT OF THE CHARACTERS MEASURED
/                  IN STEPS. PREFERRED VALUES ARE OF THE
/                  FORM IHGT=7*I, I=2,3,4....
/          THETA - THE ANGLE IN DEGREES BETWEEN THE
/                  WRITING DIRECTION OF THE TEXT AND
/                  THE HORIZONTAL AXIS.
/          IFMT  - THE ADDRESS OF THE FORMAT STATEMENT.
/                  IFMT MUST HAVE BEEN GIVEN A VALUE IN AN
/                  "ASSIGN SN TO IFMT"-STATEMENT WHERE
/                  SN IS THE FORMAT STATEMENT NUMBER.
/             N     - THE NUMBER OF CHARACTERS TO BE PLOTTED.
/             MARK  - NOT USED.
/
/       2A), N=0, A LINE IS DRAWN TO IX,IY.:
/          IX    - THE X-COORDINATE OF THE CENTER OF THE MARK.
/          IY    - THE Y-COORDINATE
/          IHGT  - THE HEIGHT OF THE MARK MEASURED IN STEPS.
/                  PREFERRED VALUES: SEE 1) ABOVE.
/          THETA - NOT USED.
/          IFMT  - NOT USED.
/          N     - MUST BE =0.
/          MARK  - SPECIFIES THE TYPE OF MARK TO BE PLOTTED.
/                  MARK LIST:
/                  0=SQUARE  1=OCTAGON  2=TRIANGLE
/                  3=PLUS  4=CROSS  5=ASTERISK
/                  6=HORIZONTAL BAR  7=VERTICAL BAR
/
/       2B), N<0, NO LINE IS DRAWN.:
/                  SAME AS 2A) BUT N<0.
/
/       - - - - - - - - - - - - - - - - - - - - - - - - -
/
/       SUBROUTINES REQUIRED:
/          PLOTTA
```

```
        DAC RBUFF□      /ROUNDBUFFER
        LLS 1
        AND (1
        TAD RBUFF
        JMP* MULSIN
MULCOS  0
        CLL
        LLS 3
        DAC .+3
        LAC COSTH
        MULS
        0
        DAC RBUFF
        LLS 1
        AND (1
        TAD RBUFF
        JMP* MULCOS
ADTBL   760222
        600130          /A
        500133          /B
        600137          /C
        620142          /D
        620145          /E
        640150          /F
        520152          /G
        640156          /H
        640160          /I
        660070          /J
        640072          /K
        720074          /L
        660075          /M
        700077          /N
        560101          /O
        620104          /P
        500107          /Q
        540113          /R
        500044          /S
        700050          /T
        640052          /U
        720054          /V
        660055          /W
        660057          /X
        660061          /Y
        600063          /Z
        700211          /Ä
        740217          /Ö
        700207          /Ü
        660175          /↑
        660220          /É
        760222          /SPACE
        520167          /₁₀
        660164          /"
        500122          /□
        520201          /Ä
        440223          /%
        600230          /&
        740166          /'
```

```
        700126          /(
        700120          /)
        520213          /*
        660205          /+
        640066          /,
        740117          /-
        660162          /.
        740043          //
        500037          /0
        660000          /1
        600002          /2
        460005          /3
        640012          /4
        560014          /5
        500017          /6
        640023          /7
        400025          /8
        500033          /9
        520233          /:
        500237          /;
        720173          /<
        660177          /=
        720174          />
        360243          /?
        600251          /SQUARE
        500254          /OCTAGON
        640260          /TRIANGLE
        620262          /PLUS
        620265          /CROSS
        460270          /ASTERISK
        700275          /HORIZ. BAR
        700277          /VERT. BAR
PLTBL   162720; 103000                          /1
        061737; 464401; 004000                  /2
        061737; 464534; 143443; 413010; 010000  /3
        170242; 323530                          /4
        470704; 344341; 301001                  /5
        463717; 060110; 304143; 341403          /6
        060747; 462120                          /7
        341405; 061737; 464534; 434130; 100103  /8-1
        143400                                  /8-2
        443313; 040617; 374641; 301001          /9
        010617; 374641; 301001; 770047          /0
        470000                                  //
        463717; 060514; 344341; 301001          /S
        074727; 200000                          /T
        070110; 304147                          /U
        072047                                  /V
        070024; 404700                          /W
        074077; 004700                          /X
        472407; 242000                          /Y
        074724; 143424; 004000                  /Z
        102122; 121121                          /,
        474130; 100100                          /J
        070003; 472540                          /K
        070040                                  /L
        000723; 474000                          /M
```

```
        000740; 470000                          /N
        463717; 060110; 304146;                 /O
        000737; 464534; 040000                  /P
        413010; 010617; 374641; 774022          /Q
        000737; 464534; 043443; 400000          /R
        034300                                  /-
        172621; 100000                          /)
        003624; 144434; 462202; 322210          /a
        372621; 300000                          /(
        000617; 374640; 430300                  /A
        000737; 464534; 043443; 413000          /B
        463717; 060110; 304100                  /C
        000737; 464130; 000000                  /D
        470704; 340400; 400000                  /E
        470704; 340400                          /F
        463717; 060110; 304143; 335300          /G
        070004; 444740                          /H
        371727; 201030                          /I
        101121; 201000                          /.
        171577; 373500                          /"
        272500                                  /'
        272212; 172777; 102021; 111000          /%
        460442                                  /<
        024406                                  />
        452705; 272000                          /↑
        014177; 054500                          /=
        461605; 143443; 320222; 202700          /Ä
        034323; 212500                          /+
        173730; 100000                          /]
        371710; 300000                          /[
        014523; 410523; 212523; 034300          /*
        074000                                  /ä
        430314; 031200                          /É
        000000                                  /SPACE
        004777; 070525; 270777; 222040; 422200  /%
        401425; 340110; 204200                  /&
        161525; 261677; 121121; 221200          /:
        161525; 261677; 102122; 121121          /;
        061737; 464534; 140312; 223277; 111020  /?-1
        211100                                  /?-2
        222404; 004044; 242200                  /SQUARE
        222414; 030110; 304143; 342422          /OCTAGON
        222501; 412522                          /TRIANGLE
        222420; 224202; 220000                  /PLUS
        220440; 220044; 220000                  /CROSS
        222420; 220242; 220440; 224400; 220000  /ASTERISK
        220242; 220000                          /HORIZ. BAR
        222420; 220000                          /VERT. BAR
RADC    324773
        216764          /FLOAT. FOR 0,01745329252
ROT     000020          /FLOAT. FOR 2↑15
        200000
THR     0               /THETA IN RADIANS (FLOAT.)
        0
        .END
```

```
/       BILAGA 3
/
/
        .TITLE NUMBER
/
/       AUTHOR, J-O LINDELL MARCH 1971
/
/
/       SUBROUTINE NUMBER(IX,IY,IHGT,THETA,FPN,NODD)
/
/       PLOTS ROUNDED DECIMAL VALUE OF FLOATING POINT NUMBER, FPN.
/
/       IX,IY,IHGT,THETA - SEE DEFINITION IN SUBROUTINE SYMBOL.
/       FPN  - FLOATING POINT NUMBER TO BE PLOTTED.
/       NODD - SPECIFIES THE NUMBER OF DIGITS TO BE PLOTTED
/              AFTER THE DECIMAL POINT
/              NODD=0:  INTEGER PART OF FPN AND DECIMAL
/                       POINT ARE PLOTTED.
/              NODD=-1: INTEGER PART OF FPN IS PLOTTED
/                       WITHOUT THE DECIMAL POINT.
/
/       - - - - - - - - - - - - - - - - - - - - - -
/
/       SUBROUTINES REQUIRED
/              SYMBOL
/              .PLT
/
        .GLOBL NUMBER,SYMBOL,.AG,.AX,.AH,.AW,.AM,.AI
        .GLOBL .AK,.AL,.DA,.BA
NUMBER  0
        JMS* .DA
        JMP .+7
NXOA    0
NYOA    0
NHGHT   0
NTHET   0
FPNA    0
NODDA   0
        LAW -6
        DAC FCNT□       /FORMATWORD-COUNT
        LAW -7
        DAC ACCCNT□     /ACCURACY-COUNT
        LAC (FORM
        IAC
        DAC FORMW
ZEROIN  DZM* FORMW      /ZEROING ALL FORMAT WORDS
        ISZ FORMW       /INCREMENT FORMW
        ISZ FCNT        /TEST FCNT
        JMP ZEROIN
        LAW -10
        DAC CNT8D□      /ONLY 8 INTEGER DIGITS PERMITTED
        LAC (FORM
        IAC
        DAC FORMW□      /ADDRESS TO FIRST WORD OF FORMAT
        LAC (1
        DAC CNT5C□
```

```
/       BILAGA 4
/
/
        .TITLE LINE
/
/       AUTHOR J-O LINDELL MARCH 1971
/
/       SUBROUTINE LINE(IX(*),IY(*),IHGT,N,K,J,MARK)
/
/       PLOTS DATA POINTS WITH OR WITOUT CONNECTING
/       LINE AND THE X-COORDINATES SPECIFIED BY THE
/       VECTOR IX AND THE Y-COORDINATES SPECIFIED BY
/       THE VECTOR IY.
/
/       IX(*)   - FIRST ELEMENT OF THE X-VECTOR
/                 TO BE PLOTTED
/       IY(*)   - FIRST ELEMENT OF THE Y-VECTOR
/       IHGT    - HEIGHT OF THE MARKS, IF ANY,
/                 MEASURED IN STEPS
/       N       - NUMBER OF POINTS TO BE PLOTTED
/       K       - EVERY K:TH ELEMENT OF IX AND IY ARE PLOTTED
/       J       - THE POINTS MAY BE CONNECTED BY A LINE
/                 AND THERE MAY BE A MARK AT SOME POINTS
/                 AS IS SPECIFIED BY J:
/                 J=0: LINEPLOT ONLY
/                 J=1: LINEPLOT AND A MARK AT EVERY I:TH POINT
/                 J=-1: NO LINEPLOT BUT A MARK AT EVERY I:TH POINT
/       MARK    - SPECIFIES THE TYPE OF MARK
/                 MARK LIST:
/                 0-SQUARE  1=OCTAGON  2=TRIANGLE
/                 3=PLUS  4=CROSS  5=ASTERISK
/                 6=HORIZ. BAR  7=VERTICAL BAR
/
/       - - - - - - - - - - - - - - - - - - - - -
/
/       SUBROUTINES REQUIRED
/                 SYMBOL
/                 PLOTTA
/
        .GLOBL LINE,.DA,SYMBOL,PLOTTA
LINE    0
        JMS* .DA
        JMP .+10
IXA     0
IYA     0
HGHT    0
NA      0
KA      0
JA      0
MARKA   0
        LAC* IXA
        DAC XC□
        LAC* IYA
        DAC YC□
        LAC* KA
        DAC K□
```

```
/       BILAGA 5
/
/
/
/
        .TITLE SCALE
/
/       AUTHOR J-O LINDELL MAY 1971
/
/
/       SUBROUTINE SCALE(U(*),IU(*),NPOI,W,UMIN,UMAX,SMIN,DS)
/
/       SCALES A VECTOR TO GIVEN SPECIFICATIONS FOR USE WITH
/       THE SUBROUTINE LINE. CALCULATES SMIN AND DS FOR
/       THE SUBROUTINE AXIS.
/
/       U(*)  - FIRST ELEMENT OF THE VECTOR TO BE SCALED
/       IU(*) - FIRST ELEMENT OF THE SCALED AND INTEGERED
/               VECTOR.
/       NPOI  - NUMBER OF VALUES TO BE SCALED.
/       W     - PERMITTED WIDTH IN CENTIMETERS ON THE SCREEN
/               FOR THE VECTOR.
/       UMIN  - MINIMUM VALUE OF THE VECTOR U.
/       UMAX  - MAXIMUM VALUE OF THE VECTOR U.
/               UMIN AND UMAX CAN BE CALCULATED BY
/               SUBROUTINE MINMAX (Q-28).
/       SMIN  - SEE DEFINITION IN SUBROUTINE AXIS.
/       DS    - SEE DEFINITION IN SUBROUTINE AXIS.
/
/       - - - - - - - - - - - - - - - - - - - - - - - - - - - -
/
/       SURROUTINES REQUIRED:
/               NONE
/
        .GLOBL .DA,.AA,.AB,.AG,.AH,.AI,.AJ,.AK,.AL,.AM
        .GLOBL .AN,.AX,AINT,SCALE
SCALE   0
        JMS* .DA
        JMP .+11
UA      0
IUA     0
NPOIA   0
WA      0
UMINA   0
UMAXA   0
SMINA   0
DSA     0
        DZM PTO
        DZM NPTO
        JMS* .AG        /LOAD
        .DSA UMAXA+400000
        JMS* .AJ        /SUBTRACT
        .DSA UMINA+400000
        JMS* .AN        /REVERSE DIVIDE
        .DSA WA+400000  /W/(UMAX-UMIN)
FPT     JMS* .AX        /FIND POWER OF TEN
```

```
/       BILAGA 6
/
/
/
/
        .TITLE AXIS
/
/       AUTHOR J-O LINDELL MAY 1971
/
/
/       SUBROUTINE AXIS(IX,IY,SL,IFMT,N,THETA,SMIN,DS)
/
/       DRAWS AN AXIS WITH MARKS,NUMBERS ( IN E-FORMAT,
/       ACCURACY 3 DEC. DIGITS) AND AN ARBITRARY TEXT.
/       THE DISTANCE BETWEEN THE MARKS IS 2 CENTIMETERS. A
/       NUMBER IS PLOTTED  AT EVERY 2:ND MARK. THE TEXT IS
/       PLOTTED WITH ITS END AT THE END OF THE AXIS.
/       FOR FURTHER INFORMATION OF THE POSITION OF MARKS,
/       NUMBERS AND TEXT: SEE ARGUMENT SL.
/
/       IX     - X-COORDINATE OF BEGINNING OF AXIS
/       IY     - Y-COORDINATE
/       SL     - VALUE OF SL IS THE LENGTH OF THE AXIS
/                IN CENTIMETERS.
/                IF SL>0 MARKS AND NUMBERS WILL BE PLOTTED ON
/                THE RIGHT SIDE OF THE AXIS AND THE TEXT ON
/                THE LEFT.
/                IF SL<0 THE POSITIONS ARE THE OPPOSITE.
/       IFMT   - ADRESS OF THE FORMAT STATEMENT OF THE TEXT.
/                IFMT MUST HAVE BEEN GIVEN A VALUE IN AN
/                "ASSIGN FN TO IFMT"-STATEMENT, WHERE FN
/                IS THE FORMAT STATEMENT NUMBER.
/       N      - NUMBER OF CHARACTERS IN THE TEXT.
/                IF N=0 NO TEXT WILL BE PLOTTED.
/       THETA  - THE ANGLE IN DEGREES BETWEEN THE AXIS TO BE
/                PLOTTED AND THE HORZONTAL AXIS.
/       SMIN   - A VALUE ROUNDED FROM UMIN AND < UMIN,
/                TO BE PLOTTED AT THE BEGINNING OF THE AXIS
/       DS     - VALUE PER CENTIMETER ALONG THE AXIS.
/
/       SMIN AND DS CAN BE CALCULATED BY THE  SUBROUTINE
/       SCALE ( 0-29).
/
/       - - - - - - - - - - -  - - - - - - - - - - - - - -
/
/       SUBROUTINES REQUIRED:
/                .PLT,SYMARK,PLTNUM
/
        .GLOBL OFNEF,PLOTTA,SYMBOL,.DA,.AG,.AH,.AI,.AJ
        .GLOBL .AK,.AL,.AW,.AX,SIN,COS,AXIS,.BA
AXIS    0
        JMS* .DA
        JMP .+11
IXA     0
IYA     0
SLA     0
```

```
        JMS* .AX
STTCA   TCA                 /STATEMENT TCA
        DAC HMMM□           /HOW MANY MARKS MORE
        JMS* PLOTTA
        JMP .+4
        .DSA IXA+400000
        .DSA IYA+400000
        .DSA (3             /PEN UP
        JMS* PLOTTA         /PLOT AXIS
        JMP .+4
        .DSA X2
        .DSA Y2
        .DSA (2             /PEN DOWN
        JMS* .AG
        .DSA SMINA+400000
        JMS* .AH
        .DSA SCNUM          /NEXT SCALE-NUMBER
        JMS* .AG
        .DSA HUND           /HUNDRED STEPS = 2 CM
        JMS* .AH
        .DSA NSM            /DISTANCE FROM ZERO TO NEXT SCALE-MARK
        LAC* IXA
        DAC XSC□            /X-COORD OF NEXT SCALE-MARK
        LAC* IYA
        DAC YSC□            /Y-COORD OF NEXT SCALE-MARK
OSMM    JMS* .AG            /ONE SCALE-MARK MORE
        .DSA FOUR           /MARK IS 4 STEPS HIGH
        JMS MULSIN
POM1    XX                  /PLUS-OR-MINUS, NOP OR TCA
        DAC S4□             /+ OR - 4*SIN(THETA)
        TAD XSC
        DAC XM□             /X-COORD OF END OF SCALE-MARK
        JMS* .AG
        .DSA FOUR
        JMS MULCOS
POM2    0                   /NOP OR TCA
        DAC C4□             /+ OR - 4*COS(THETA)
        TCA
        TAD YSC
        DAC YM□             /Y-COORD OF END OF SCALE-MARK
        JMS* PLOTTA
        JMP .+4
        .DSA XSC
        .DSA YSC
        .DSA (3             /PEN UP
        JMS* PLOTTA         /PLOT SCALEMARK
        JMP .+4
        .DSA XM
        .DSA YM
        .DSA (2             /PEN DOWN
        ISZ NCNT□           /NUMBER AT EVERY 2:ND SCALE-MARK
        JMP NONUM
        LAC DP              /DETERMINE POS. OF NUMBER
        SPA
        JMP .+5
        JMS* .AG
        .DSA ETEEN          /4 STEPS UNDER MARK. 4+HEIGHT=18
```

OBS !

```
C       BILAGA 7
C
C
C
C
C
C       TITLE: SYMLIB (SYMBOL-LIBRARY)
5       CALL PLOTS(0,DUM,DUM)
        ASSIGN 10 TO IBC1
        ASSIGN 15 TO IBC2
        ASSIGN 20 TO IBC3
        ASSIGN 25 TO IBC4
        ASSIGN 26 TO IBC44
        ASSIGN 30 TO IBC5
        ASSIGN 31 TO IBC55
        ASSIGN 32 TO IBC6
        ASSIGN 33 TO IBC7
        ASSIGN 34 TO IBC8
10      FORMAT(26HABCDEFGHIJKLMNOPQRSTUVWXYZ)
15      FORMAT(20H1234567890 ₁₀"¤Ä%&'())
20      FORMAT(18H↑]<>?/.,{ö+;€Å:*-=)
25      FORMAT(25HTHE QUICK BROWN FOX JUMPS)
26      FORMAT(17HOVER THE LAZY DOG)
30      FORMAT(19HHHH - 3 H'S TO TEST)
31      FORMAT(18HSCANNING-PROCEDURE)
32      FORMAT(20HTHETA IS 90. DEGREES)
33      FORMAT(21HTHETA IS 180. DEGREES)
34      FORMAT(21HTHETA IS 270. DEGREES)
        DO 35 I=1,8
        J=I-1
35      CALL SYMBOL(20+100*I,750,35,0.,0,-(I-3)*(I-2),J)
        CALL SYMBOL(120,650,35,0.,IBC1,26,0)
        CALL SYMBOL(120,550,35,0.,IBC2,20,0)
        CALL SYMBOL(120,450,35,0.,IBC3,18,0)
        CALL SYMBOL(120,350,35,0.,IBC4,25,0)
        CALL SYMBOL(120,270,35,0.,IBC44,17,0)
        CALL SYMBOL(120,170,35,0.,IBC5,19,0)
        CALL SYMBOL(120,90,35,0.,IBC55,18,0)
        PAUSE 1
        CALL SYMBOL(50,210,21,90.,IBC6,20,0)
        CALL PLOTTA(60,204,2)
        CALL PLOTTA(60,564,1)
        CALL SYMBOL(685,25,21,180.,IBC7,21,0)
        CALL PLOTTA(685,31,2)
        CALL PLOTTA(306,31,1)
        CALL SYMBOL(940,550,21,270.,IBC8,21,0)
        CALL PLOTTA(930,550,2)
        CALL PLOTTA(930,180,1)
        PAUSE 2
        GO TO 5
        STOP
        END
```

```
C       BILAGA 8
C
C
C
C
C
C       TITLE: JOL
        CALL PLOTS(0,DUM,DUM)
        ASSIGN 10 TO IBCD
        DO 5 I=1,12
        R=I
        X=COS(30.*0.01745329252*R)
        Y=SIN(30.*0.01745329252*R)
        K=90.*X+10.*Y
        L=90.*Y-10.*X
        CALL PLOTTA(500+K,400+L,2)
        K=396.*X+10.*Y
        L=396.*Y-10.*X
        CALL PLOTTA(500+K,400+L,1)
5       CALL SYMBOL(500,400,21,30.*R,IBCD,22,0)
10      FORMAT(22H     JAN-OLOF LINDELL )
        STOP
        END
```

```
C       BILAGA 9
C
C
C
C
C
C       TITLE: LISS (LISSAJOU-FIGURE)
        DIMENSION J(361),K(361)
        DO 25 I=1,361
        R=I-1
        J(I)=1000./3.*SIN(R*0.0174533)+500.
25      K(I)=1000./3.*SIN(2.*R*0.0174533)+375.
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),21,361,1,0,2)
        PAUSE 1
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),21,361,1,5,2)
        PAUSE 2
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),35,11,36,2,1)
        PAUSE 3
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),35,361,1,-20,0)
        PAUSE 4
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),21,37,10,1,3)
        PAUSE 5
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),14,361,1,-10,4)
        PAUSE 6
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),21,361,1,10,5)
        PAUSE 7
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),21,361,1,10,6)
        PAUSE 10
        CALL PLOTS(0,0,0)
        CALL LINE(J(1),K(1),21,361,1,10,7)
        STOP
        END
```

```
C       BILAGA 10
C
C
C
C
C       PROGRAM DEMEX
C
C       AUTHOR J-O LINDELL JUNE 1971
C
C       DEMONSTRATION OF THE USE OF THE SUBROUTINES
C       PLOTTA,SYMBOL,MARK (INSIDE SYMBOL),NUMBER
C       (IN REVISED FORM CALLED OFNEF),LINE,SCALE AND AXIS.
C
        REAL MIN, MAX
        DIMENSION S(91), C(91), T(91), CT(91), ARGD(91)
        DIMENSION ISA(91), ISS(91), ISC(91), IST(71), ISCT(71)
1       CALL ERASE
        DO 10 I=1,91
        R=I-1
        ARGR=R*0.0174533
        S(I)=SIN(ARGR)
        C(I)=COS(ARGR)
        T(I)=S(I)/C(I)
        CT(I)=C(I)/S(I)
10      ARGD(I)=R
        ASSIGN 15 TO IFM15
        ASSIGN 16 TO IFM16
        ASSIGN 17 TO IFM17
        CALL SYMBOL(100,750,14,0.,IFM15,61)
        CALL SYMBOL(100,720,14,0.,IFM16,61)
        CALL SYMBOL(100,690,14,0.,IFM17,57)
        CALL MARK(300,657,1,0)
        CALL MARK(450,657,1,1)
        CALL MARK(300,637,1,2)
        CALL MARK(450,637,1,3)
        ASSIGN 20 TO IFM20
        ASSIGN 21 TO IFM21
        ASSIGN 22 TO IFM22
        ASSIGN 23 TO IFM23
        CALL SYMBOL(315,650,14,0.,IFM20,4)
        CALL SYMBOL(465,650,14,0.,IFM21,6)
        CALL SYMBOL(315,630,14,0.,IFM22,7)
        CALL SYMBOL(465,630,14,0.,IFM23,9)
C
C       THIS IS AN EXAMPLE OF THE USE OF THE SUBROUTINES
C       MARK AND SYMBOL.RESULT:SEE PICTURE 1
C
        PAUSE 1
        IC=1
        CALL MINMAX(S(1),91,MIN,MAX,IC)
        CALL MINMAX(C(1),91,MIN,MAX,IC)
        CALL MINMAX(T(1),71,MIN,MAX,IC)
        CALL MINMAX(CT(20),71,MIN,MAX,IC)
        CALL SCALE(S(1),ISS(1),91,12.2,MIN,MAX,AMINY,DS)
        CALL SCALE(C(1),ISC(1),91,12.2,MIN,MAX,AMINY,DS)
        CALL SCALE(T(1),IST(1),71,12.2,MIN,MAX,AMINY,DS)
```

```
      CALL SCALE(CT(20),ISCT(1),71,12.2,MIN,MAX,AMINY,DS)
      CALL SCALE(ARGD(1),ISA(1),91,18.,0.,90.,AMINX,DSX)
      ASSIGN 24 TO IFM24
      ASSIGN 25 TO IFM25
      CALL AXIS(0,0,19.5,IFM24,18,0.,AMINX,DSX)
      CALL AXIS(0,0,-13.,IFM25,13,90.,AMINY,DS)
C
C     THIS IS AN EXAMPLE OF THE USE OF THE SUBROUTINES
C     SCALE AND AXIS (MINMAX IS A SMALL SUBROUTINE
C     WHICH FINDS MINIMUM AND MAXIMUM OF A VECTOR
C     FOR THE SUBROUTINE SCALE.)RESULT:SEE PICTURE 2.
C
      PAUSE 2
      CALL LINE(ISA(1),ISS(1),91,1,10,0)
      CALL LINE(ISA(1),ISC(1),91,1,10,1)
      CALL LINE(ISA(1),IST(1),71,1,10,2)
      CALL LINE(ISA(20),ISCT(1),71,1,10,3)
C
C     THIS IS AN EXAMPLE OF THE USE OF THE SUBROUTINE
C     LINE.RESULT:SEE PICTURE 3.
C
15    FORMAT(61HTHIS IS A DEMONSTRATION OF THE USE
     5 OF THE SUBROUTINES SYMBOL,)
16    FORMAT(61HAXIS,SCALE AND LINE (AND
     5 INDIRECTLY OFNEF AND PLOTTA) FOR USE)
17    FORMAT(57HWITH A DISPLAYOSCILLOSCOPE
     5 CONNECTED TO A PDP-15 COMPUTER)
20    FORMAT(4HSINE)
21    FORMAT(6HCOSINE)
22    FORMAT(7HTANGENT)
23    FORMAT(9HCOTANGENT)
24    FORMAT(18HANGLE IN   DEGREES)
25    FORMAT(13HFUNCTIONVALUE)
      PAUSE 3
      GO TO 1
      STOP
      END
```