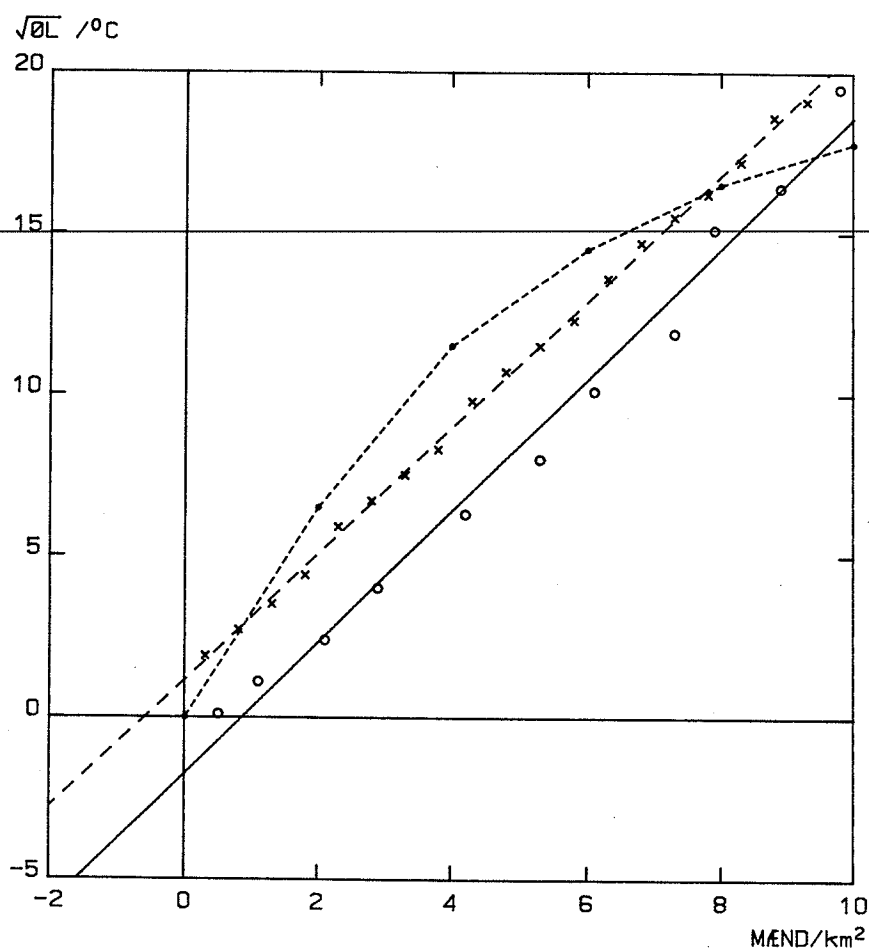




JOHS. F. MUNCH - PETERSEN

LINEÆR REGRESSION PÅ QL

II. PROGRAMUDSKRIFT

Den polytekniske Lærestalt, Danmarks tekniske Højskole
Lyngby 1986

INDLEDNING

Denne IFH-rapport nr. 181, "LINEÆR REGRESSION PÅ QL. II. Programudskrift " kan ikke læses som en selvstændig rapport. Den er dokumentation til "LINEÆR REGRESSION PÅ QL. I. Vejledning", IFH-rapport nr. 180, som forudsættes læst. Rapport 180 beskriver programmets hensigt og brug, og er derfor et nødvendigt referencebilag under læsningen af rapport 181.

Kendskab til QL's brugervejledning og til BASIC (SUPERBASIC) er nødvendige forudsætninger. IFH-rapporterne 177, 178, 179 henvises der til for så vidt angår en række grafiske fremgangsmåder.

Jeg har tilstræbt at gøre dokumentationen så fuldstændig som mulig, da jeg forudser, at mange brugere vil justere programmet til deres personlige behov.

Eksempler kunne være logaritmiske akser, skalatal med to decimaler, "fine" akseinddelinger med talværdier for hver femte / tiende streg og tegnefelt med linier tværs over feltet i st. for akseinddelingsmærker.

Programmet kan arbejde med to samtidige punktsæt med hvert sit symbol. Sættene kan slås sammen / ændres ved revision af symbolerne, reduceres ved fjernelse af punkter, udvides med nye punkter osv.

Derimod kan programmet ikke indlæse to af hinanden uafhængige filer til samtidig brug, da det vil kræve et omfattende kontrolsystem for numre, symboler osv. Jeg foreslår, at der fx laves et kort specialprogram til den slags opgaver. En sådan ny fil, korrekt opbygget, vil kunne anvendes af nærværende program, hvis den er lagret under "navn_data".

Herudover kan en række PROCedurer anvendes i andre sammenhænge.

12. december 1986.

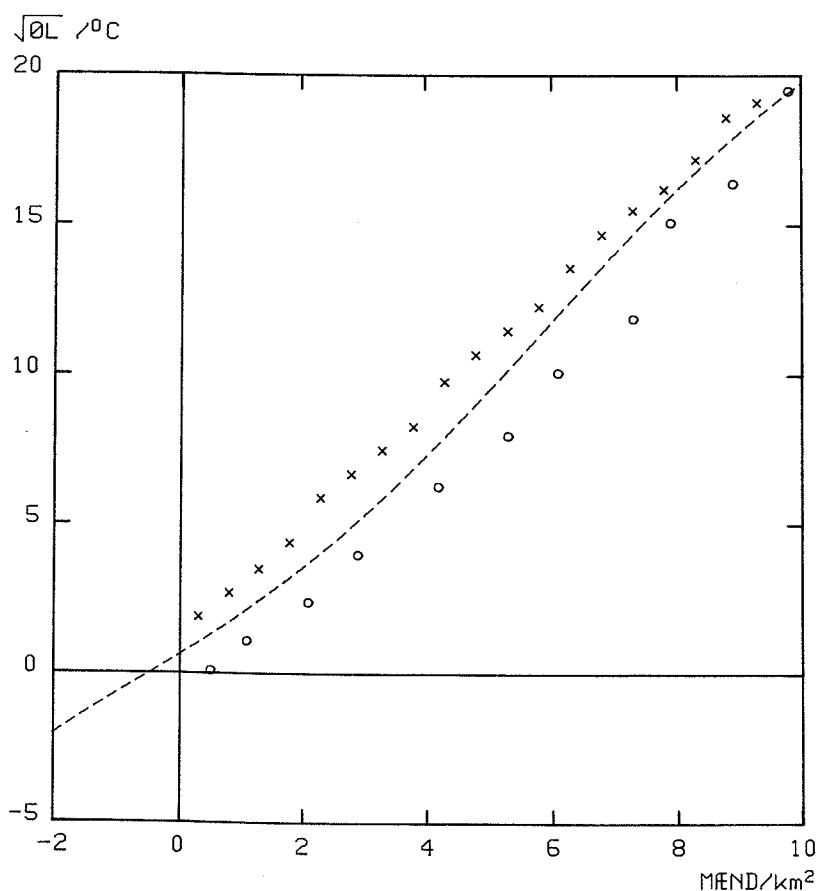
Johs. F. Munch-Petersen

IFH-rapporter om grafik og karakterer på QL, printer, plotter:

- Nr.177 RUMGITTER PÅ QL. En Introduktion til Grafik på QL.
- Nr.178 BRUGERDEFINEREDE KARAKTERER PÅ QL.
- Nr.179 DOWNLOAD - KARAKTERER PÅ QL / CPA80S - printeren.
- Nr.180 LINEÆR REGRESSION PÅ QL. I, Vejledning.
- Nr.181 LINEÆR REGRESSION PÅ QL. II, Programudskrift.

INDHOLDSFORTEGNELSE

INDLEDNING.	3
INDHOLDSFORTEGNELSE.	4
PROGRAM "dth_regression" (SERVOGOR-plotter).	5
FLOWDIAGRAMMER.	6
PROCEDURE-OVERSIGT.	21
BOOT-PROGRAM "fejlboot".	25
BOOT2-PROGRAM "fejlboot2".	27
PAUSEPROGRAM "regresskærm".	29
HOVEDPROGRAM "dth_regression".	31
PROGRAM "regression" (ROLAND DXY-plotter).	93
OVERSIGTER OVER VARIABLER, VARIABELLISTE.	105
TEORI.	129



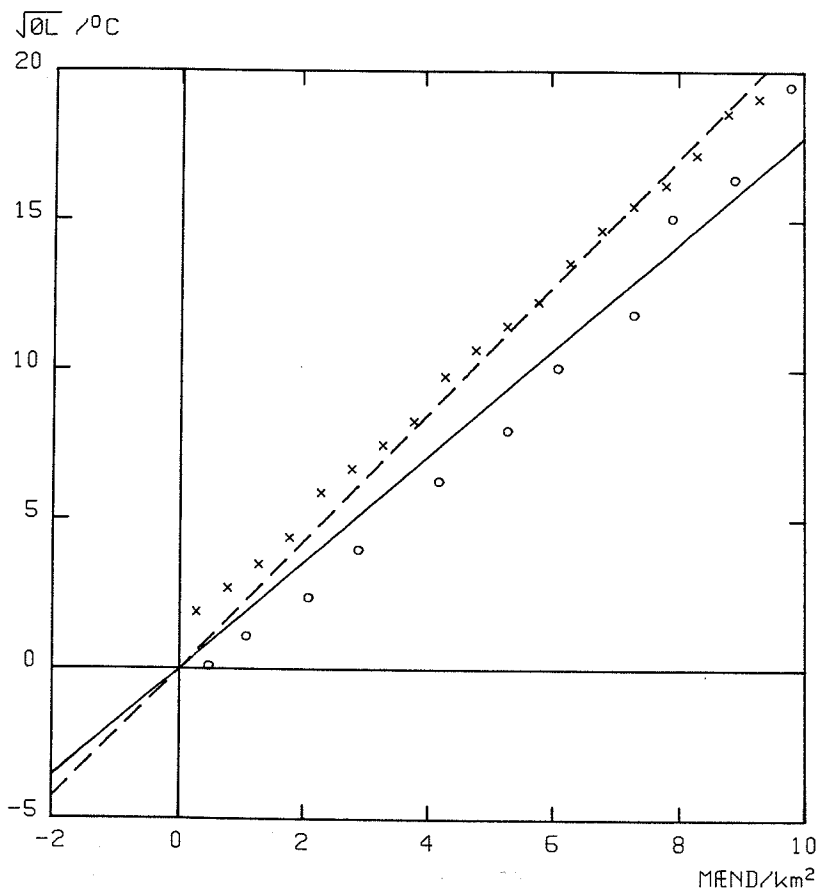
ØLFORBRUG PR.
FAMILIEOVERHOVED
I TYNDTBEFOLKEDE OMRÅDER

DEN GENNEMSITLIGE
FAMILIESTØRRELSE FOR
OMRÅDERNE: 6.3 PERSONER

KURVE INDLAGT:

$$y(x) = 1.8 \cdot (x + 0.5) + \sin(-x/2)$$

PROGRAM "DTH_REGRESSION"
SERVOGOR 281 PLOTTER



ØLFORBRUG PR.
FAMILIEOVERHOVED
I TYNDBEFOLKEDE OMRÅDER

DEN GENNEMSNTLIGE
FAMILIESTØRRELSE FOR
OMRÅDERNE: 6.3 PERSONER

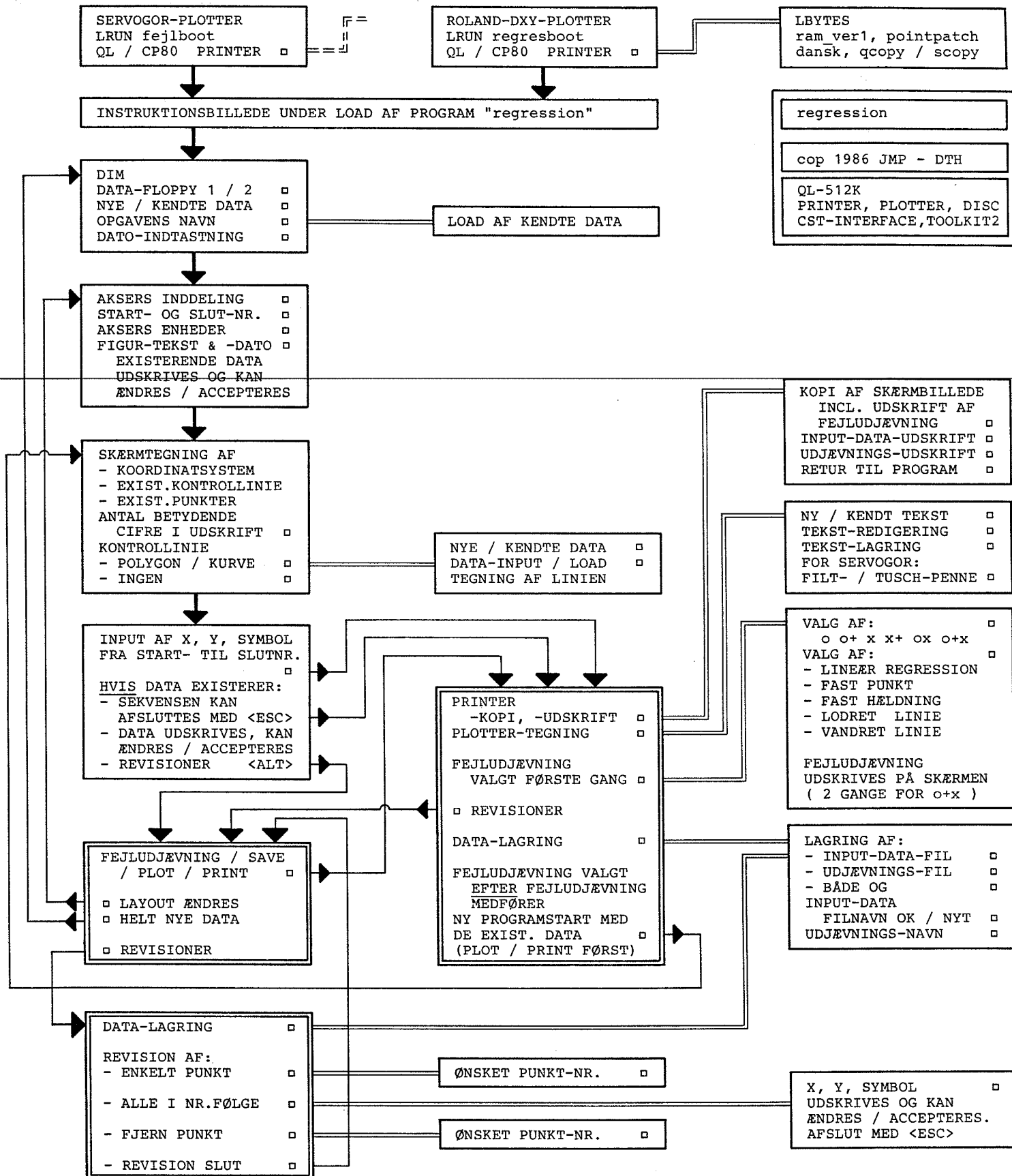
LINIER GENNEM PUNKT 0,0

PUNKTERNE o :
GRØNLAND 1985
LINIENS HÆLDNING 1.78
KORRELATIONSKOEFFICIENT 0.98

PUNKTERNE x :
SAHARA 1977
LINIENS HÆLDNING 2.14
KORRELATIONSKOEFFICIENT 0.99

o ————
x - - - -

FLOWDIAGRAM

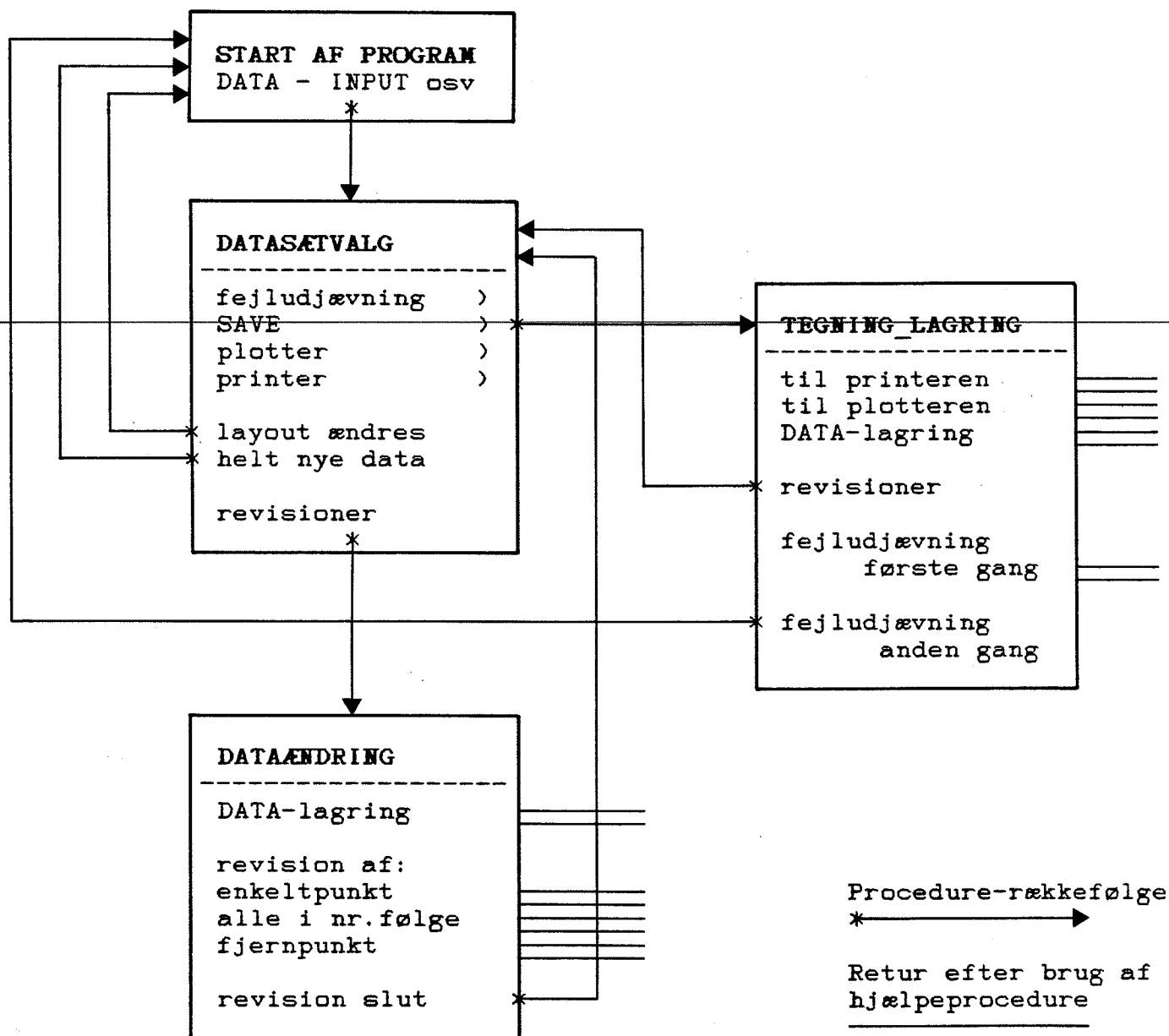


→ PROCEDURE-RÆKKEFØLGE
 == RETUR EFTER BRUG AF PROCEDURE

PROGRAMMETS TRE HOVED-MENUER:

DATASÆTVALG, TEGNING_LAGRING, DATAÆNDRING.

Forenklet oversigt over FLOWDIAGRAMMET.



DETALJEREDE OVERSIGTER OVER DE FORSKELLIGE STADIER I PROGRAMMET

med linienumre og bl.a variabelen alt\$'s indflydelse på flow vises på de følgende sider:

START-PROCEDURER I

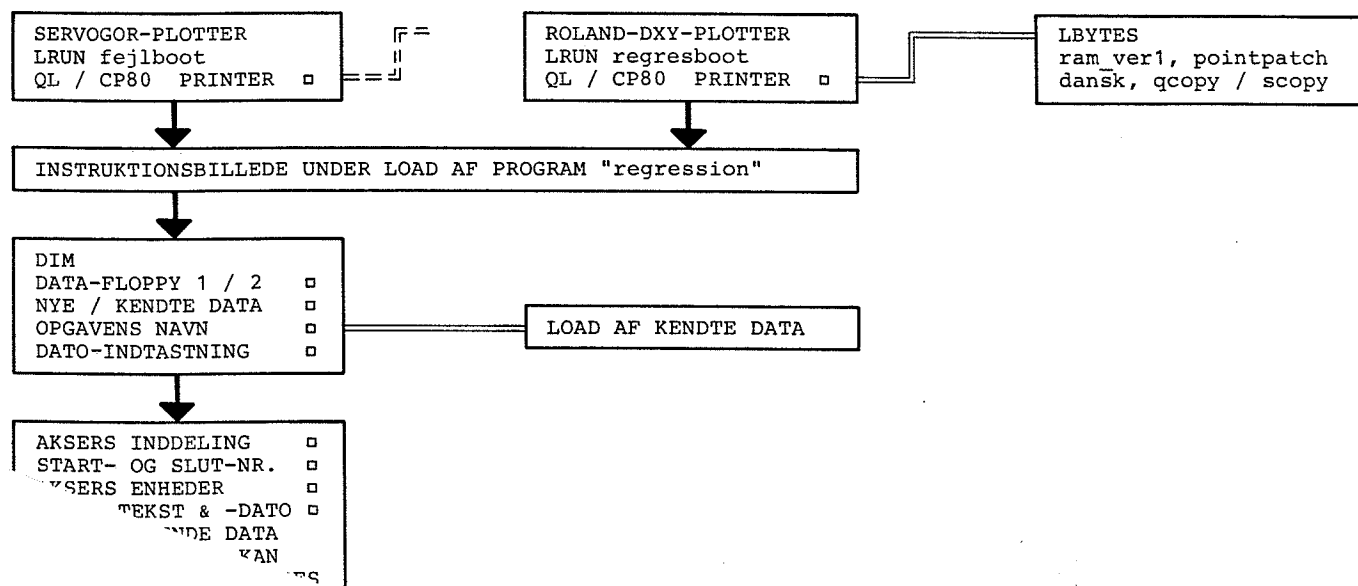
FEJLUDJÆVNINGS-PROCEDURER

STARTPROCEDURER II

PLOTTER-PROCEDURER

TEGNE-LAGRE-PROCEDURER

ÆNDRINGS-PROCEDURER



Efter boot og skærbilledet

- datoen sættes
- programmets skærm-layout etableres
- eventuelt LOADEs en eksisterende DATA-fil
- DATA vises og udprintes om ønsket
- Der fortsættes til programmets centrale del, "REPeat hovedloop", hvorfra de overordnede valg træffes. Strengvariablen alt\$ styrer forløbet, indledningsvis med værdien "i" for nye, "l" for kendte DATA.

```

fejlboot /
regresboot

```

```

i RAM2 :
load$= "ny"
pr$
dansk0
dansk1

```



SKÆRMBILLEDE "regres_scr" under LOAD af programmet



```

(dth)regression
linie 100...490

```

```

load$= "gl", se
linie 200...240

```

```

load$= "ny"
DIM etc

```

450 *

450 *

460 *

```

Kendte DATA 470 *

```

```

load$= "gl"

```

```

alt$= "i" / "l"

```



```

REPEAT
hovedloop    linie 600

Se START-PROCEDURER II

```

START-PROCEDURER I.

START med boot, TIL hovedloop, linie 600.

Maskinkoder: qltegn0, qltegn1, ram_ver1,
scopy_bin / qcopy_bin,
nytpatch, samt
cp80græsk_bin (CPA80S).
Plotter : SERVOGOR ved "fejlboot".
ROLAND DXY ved "regresboot".

dobb_vindue 30300

tidspunkt 16000

FLP 1/2
Nye / kendte DATA

16220,16410,16420 *

16690 *

SDATE 16820

starttid 16830

filservice 17000

måned_til_streng
17500

udjævn_vindue 30000

loadepoces 18200

18260 GET DATA

18380 *

xykanal4 18500

Evt. valg 18660

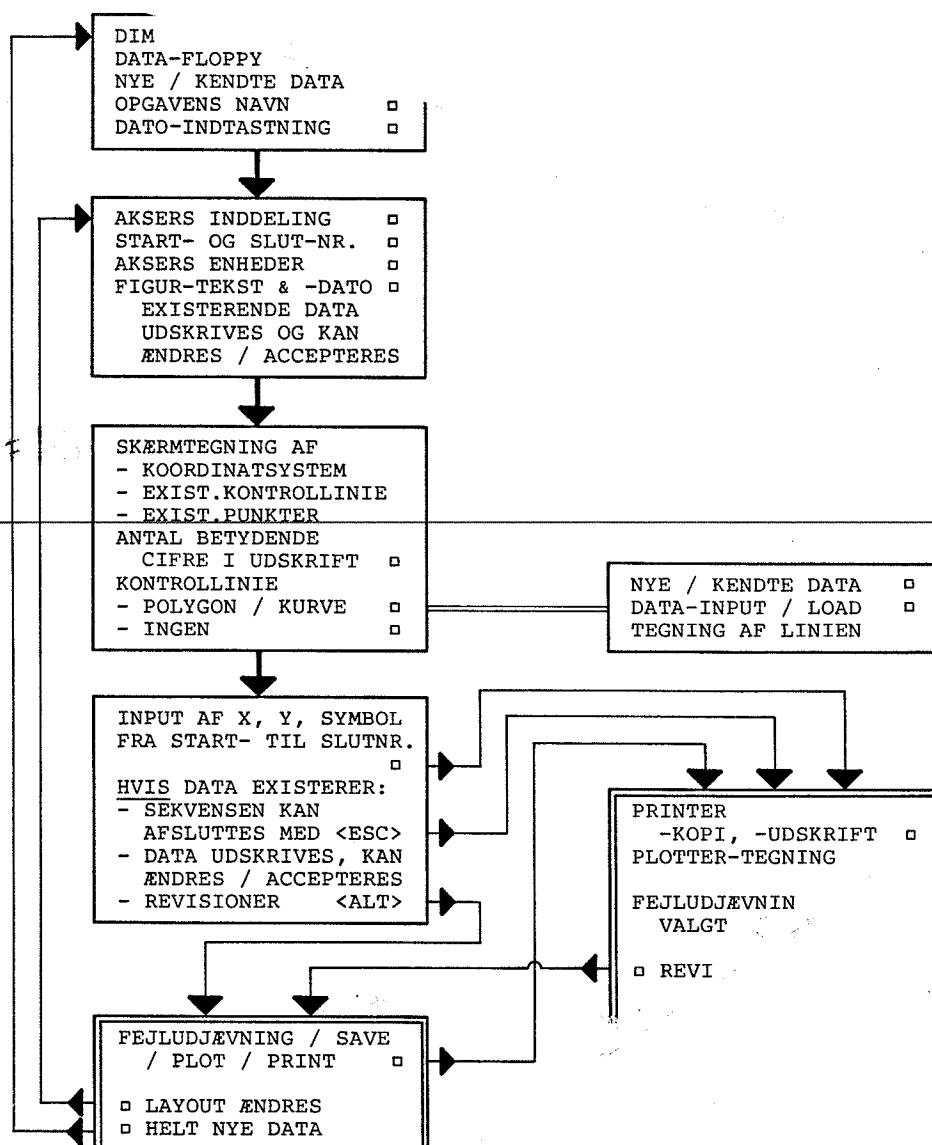
inputudskrift 14500

14560,14570

printerstreng 15000

ciffertal, talput, svarput VISES IKKE.

start/slut * envejsflow
tovejsflow

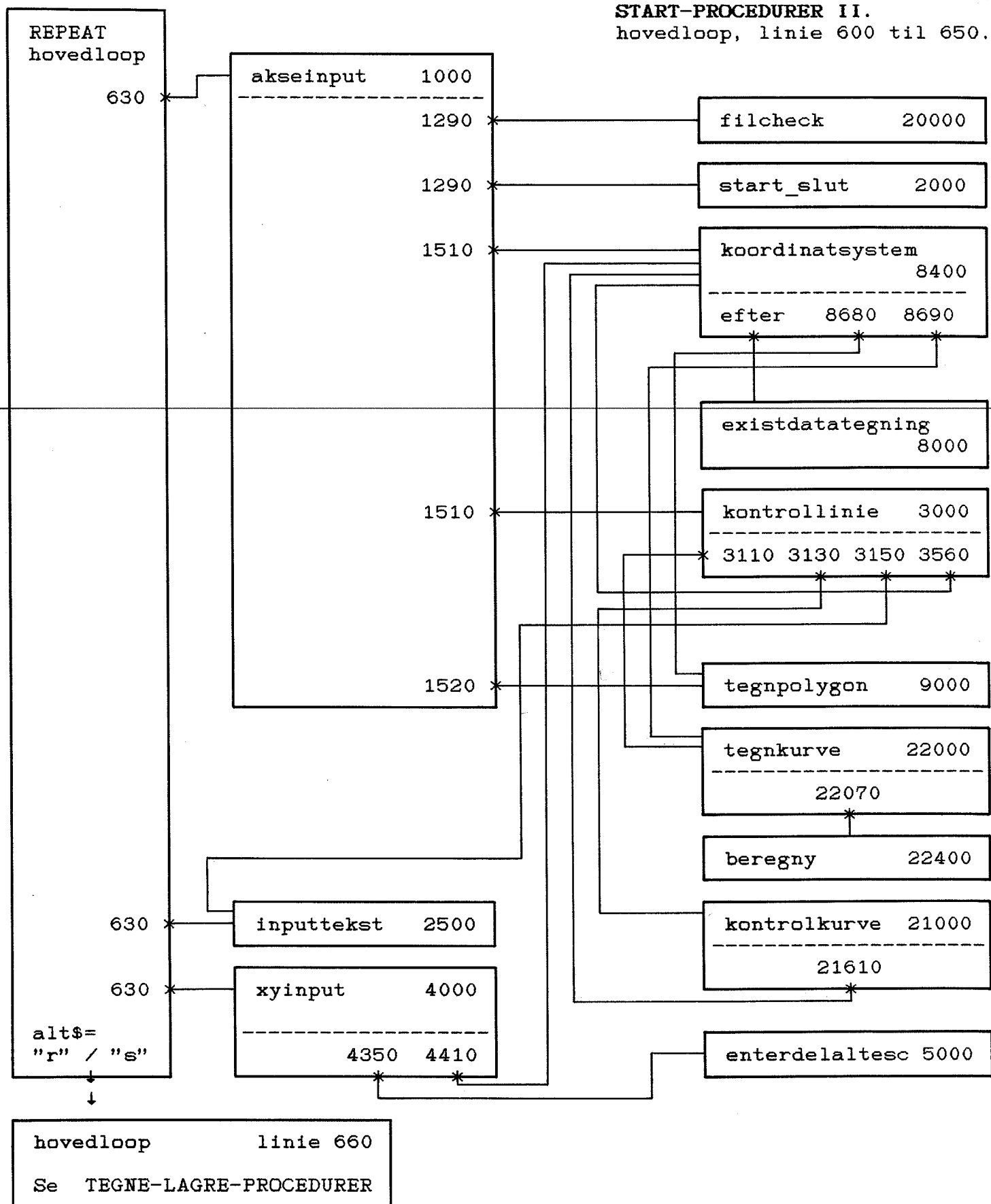


Med værdierne alt\$ = "i" eller "l":

- INPUT af akser, inddelinger, figurnavn og -dato
- Start- og slutnummer, herunder kontrol hvis kendte DATA findes. Endvidere INPUT af x, y, symbol, hvis et kendt sæt udvides.
- Koordinatsystemet tegnes, kendte DATA udtegnes.
- Kontrollinie skal afvises / vælges som ny eller kendt.
- INPUT af x, y, symbol for en eventuelt valgt ny kontrolpolygon.
- INPUT af DATA for en eventuelt valgt ny kontrolkurve.
- INPUT af x, y, symbol for punkter. Kendte DATA kan accepteres, helt eller delvist, revideres mv.
- Sekvensen forlades med "s" (normalt) eller "r" = revisioner.

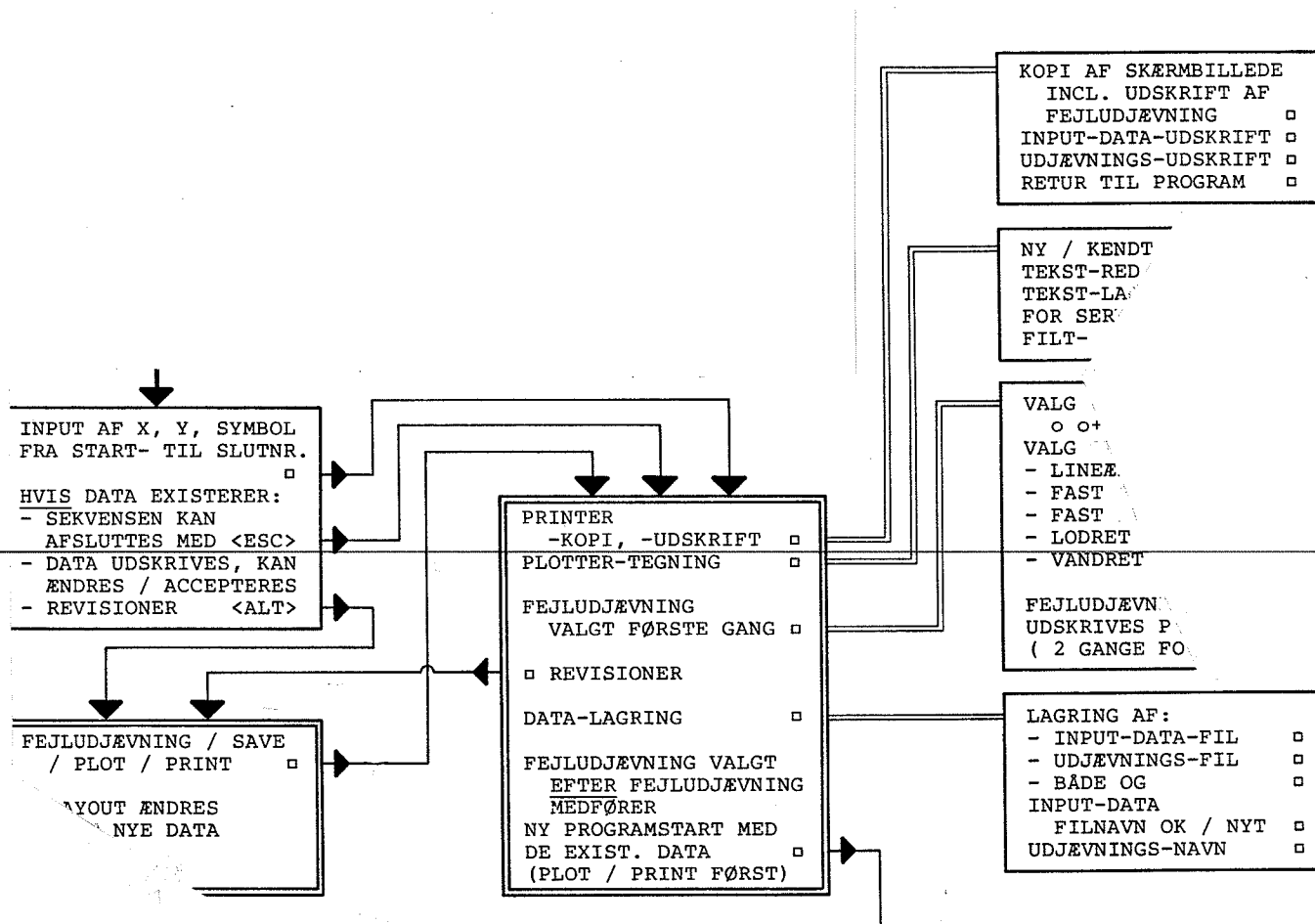
START-PROCEDURER II.

hovedloop, linie 600 til 650.



ciffertal, talput, svarput VISES IKKE.

start/slut *-----> envejsflow
tovejsflow



Efter START-PROCEDURERNE :

alt\$ = "r", linie 760 > >>> ÆNDRINGS-PROCEDURER.

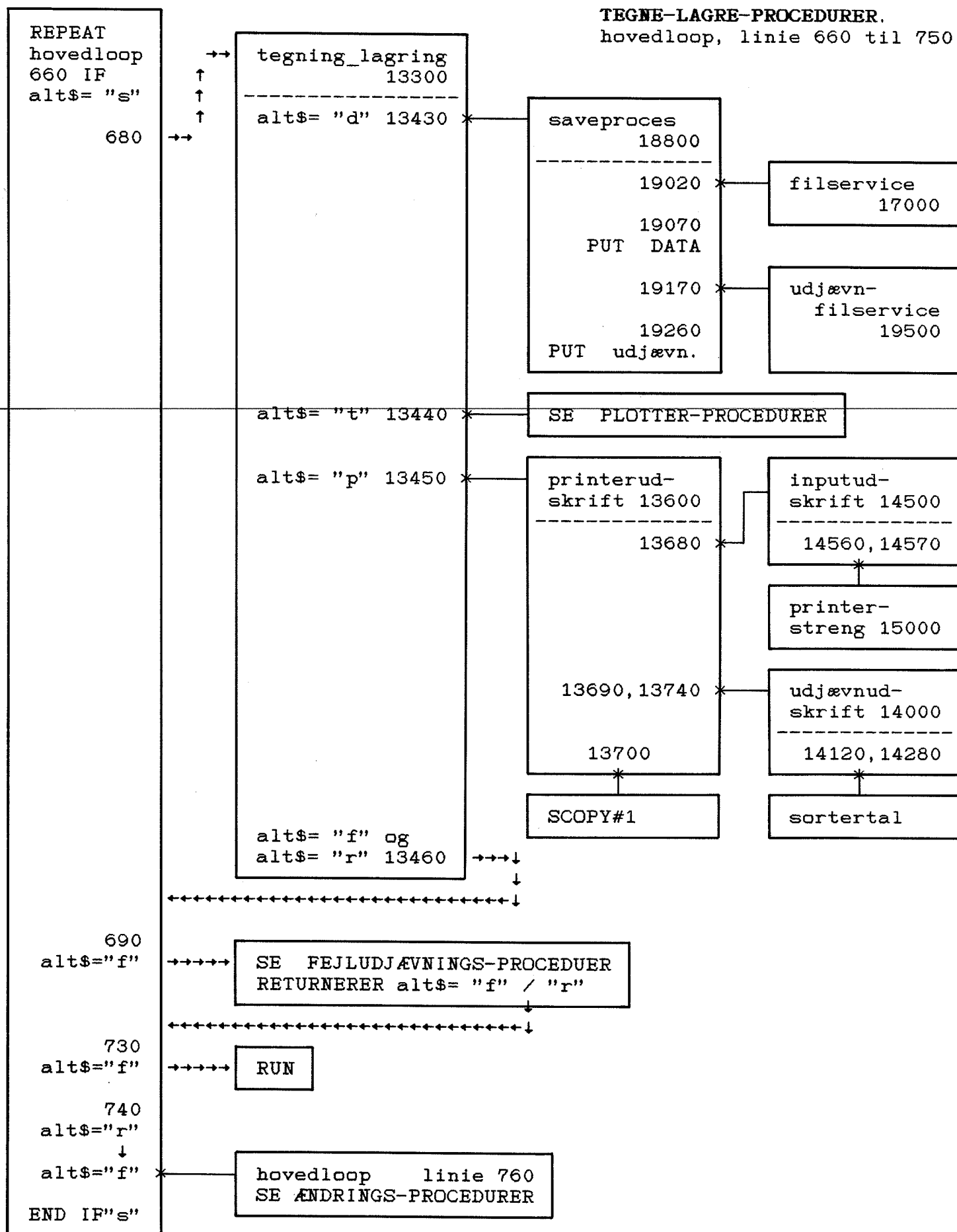
HER VISES KUN SITUATIONEN

alt\$ = "s", der fører til "tegning_lagring" med valgene:

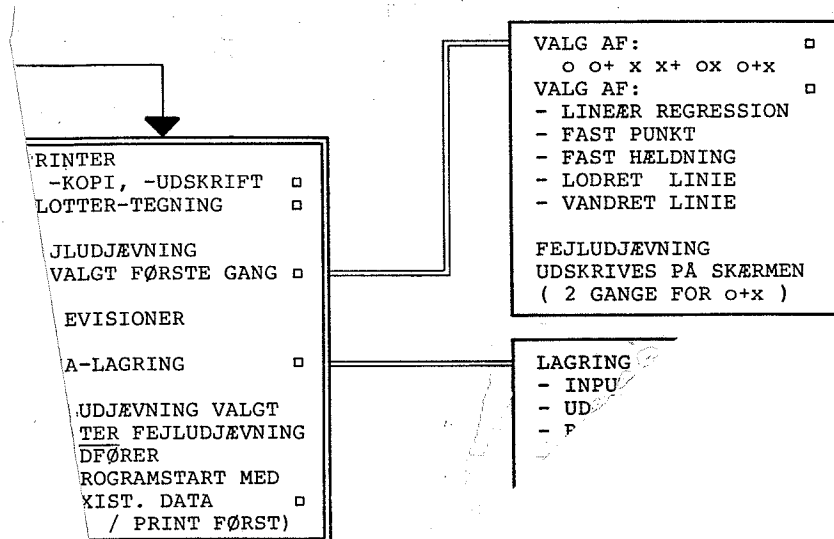
- SAVE, lagring af INPUT-DATA og / eller udjævnings-DATA, udskrift af eksist. filnavne, kontrol af valgt filnavn etc
- PLOTTER-PROCEDURER
- printerudskrift af INPUT-DATA og / eller udjævnings-DATA.
- fejludjævning, retur til hovedloop med alt\$ = 'f', linie 690 >>>> FEJLUDJÆVNINGS-PROCEDURER. Derefter gentages tegning_lagring med de samme valg. Af disse fører retur til program altid til samme resultat, MEN FORNYET VALG AF fejludjævning, "f", giver nu RUN, i hovedloop's linie 730, se FLOWDIAGRAM pag. 6.
- retur til program, til hovedloop med alt\$ = "r", i linie 740 bliver alt\$ = "f", linie 770 >>> ÆNDRINGS-PROCEDURER.

TEGNE-LAGRE-PROCEDURER.

hovedloop, linie 660 til 750.



+++++ envejsflow
start/slut * tovejsflow
ciffertal, talput, svarput VISES IKKE.



INGEN YDERLIGERE KOMMENTARER.

FEJLUDJÆVNINGS-PROCEDURER.

START OG SLUT i hovedloop, linie 700.

```

regrestype          10000
-----
pkt$: symboler, der er
      defineret:   o x ox

                        10190
valg$: fejludjævning af:
      o / x /   ox under et
      o+x      hver for sig
      o+       o, men x vises
      x+       x, men o vises
rg$: symboler, der lige
      nu udjævnes: o x ox
k=  0 for valg$ = o o+ ox
k= 20 for valg$ = x x+
k=  0, 20 for valg$ = o+x

                        10240
rgnr(100) >> "sortertal"
      k= 0 / 50 for o / x
rgtal, rgoantal, rgxantal
      til antal symboler.
udj$(40,24)      skærmtekst
fjl(40)          SAVE / PLOT

                        10270
IF valg$= "o+x": rg$= "o"

                        10390
hæld$ "r" "l" "v" "h" "p"

                        10510

```

IF valg\$= "o+x":

linie 10270 : rg\$= "o"

 tegning_lagring,
 linie 13320 : RETURN

 hovedloop,
 linie 700 : rg\$= "x"
 fejludjævning
 tegning_lagring (normal)

```

fejludjævning      11000
-----
                        11150,11170 *
11500,11510,12010,12020
Linier, afbildet
næsten lodret / vandret

                        11870,12050,12110 *
12200

```

```

udtegning          13300
-----
                        13010 *
                        710  *

```

sortertal 19600

hæld\$ INSTR"lv": fejludjævning

akseparametre 12500

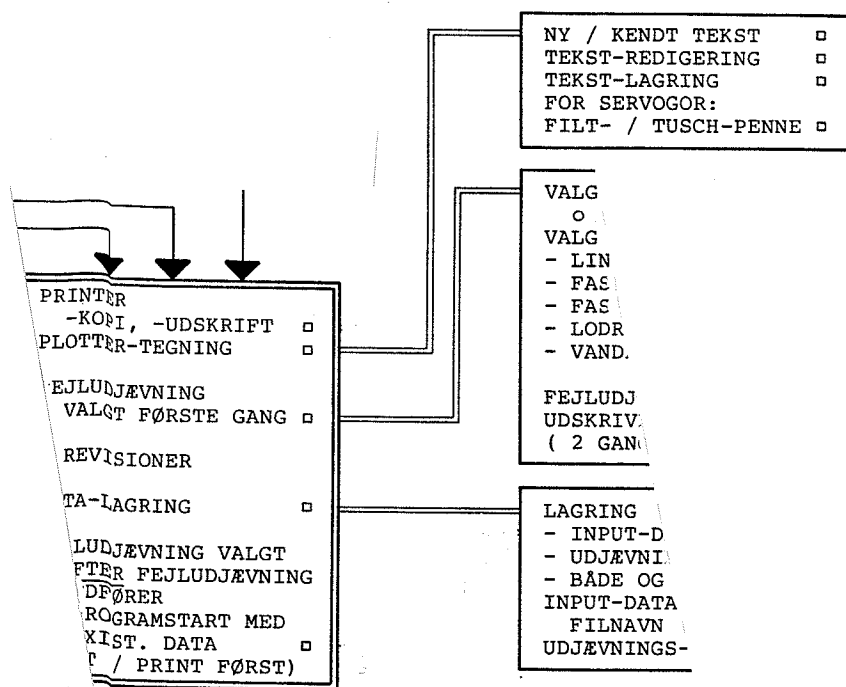
 koordinatsystem 8400
 existdatategning 8000
 Se START-PROCEDURER II

 tegning_lagring 13300
 Se TEGNE-LAGRE-PROCEDURER

ciffertal, talput, svarput VISES IKKE.

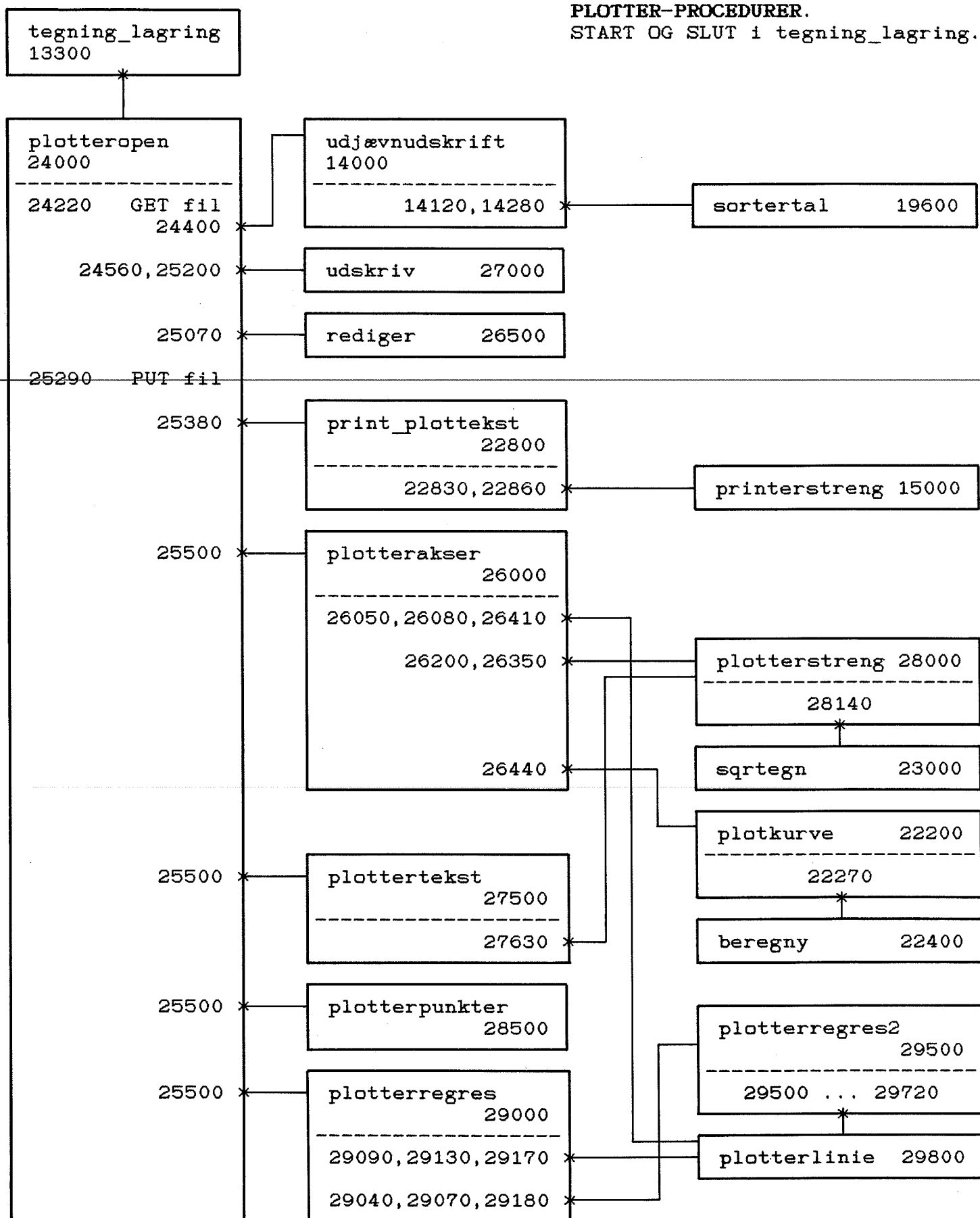
start/slut

 ++++++ envejsflow
 *----- tovejsflow



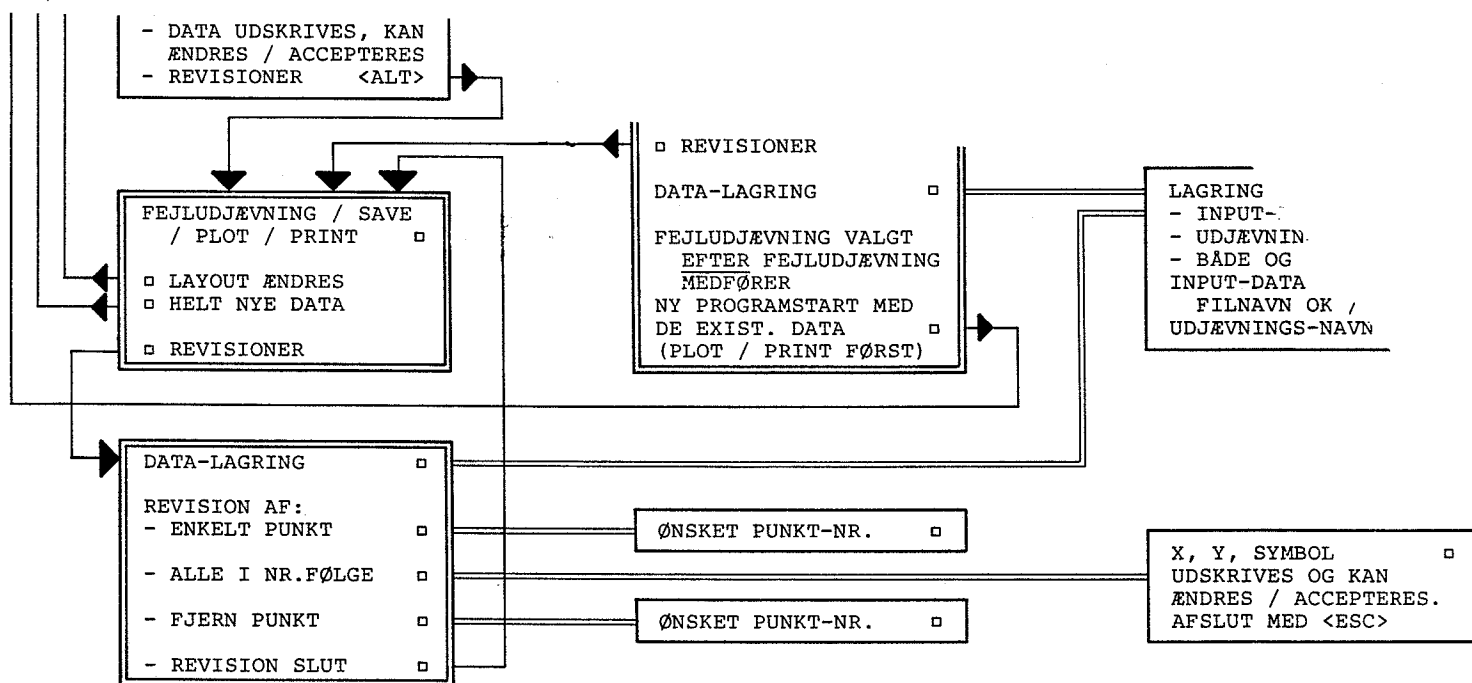
INGEN YDERLIGERE KOMMENTARER.

PLOTTER-PROCEDURER.
 START OG SLUT i tegning_lagring.



ciffertal, talput, svarput VISES IKKE.

start/slut *-----> envejsflow
 tovejsflow



Bemærk, at alt\$ = "a" og "e" stort set er identiske.

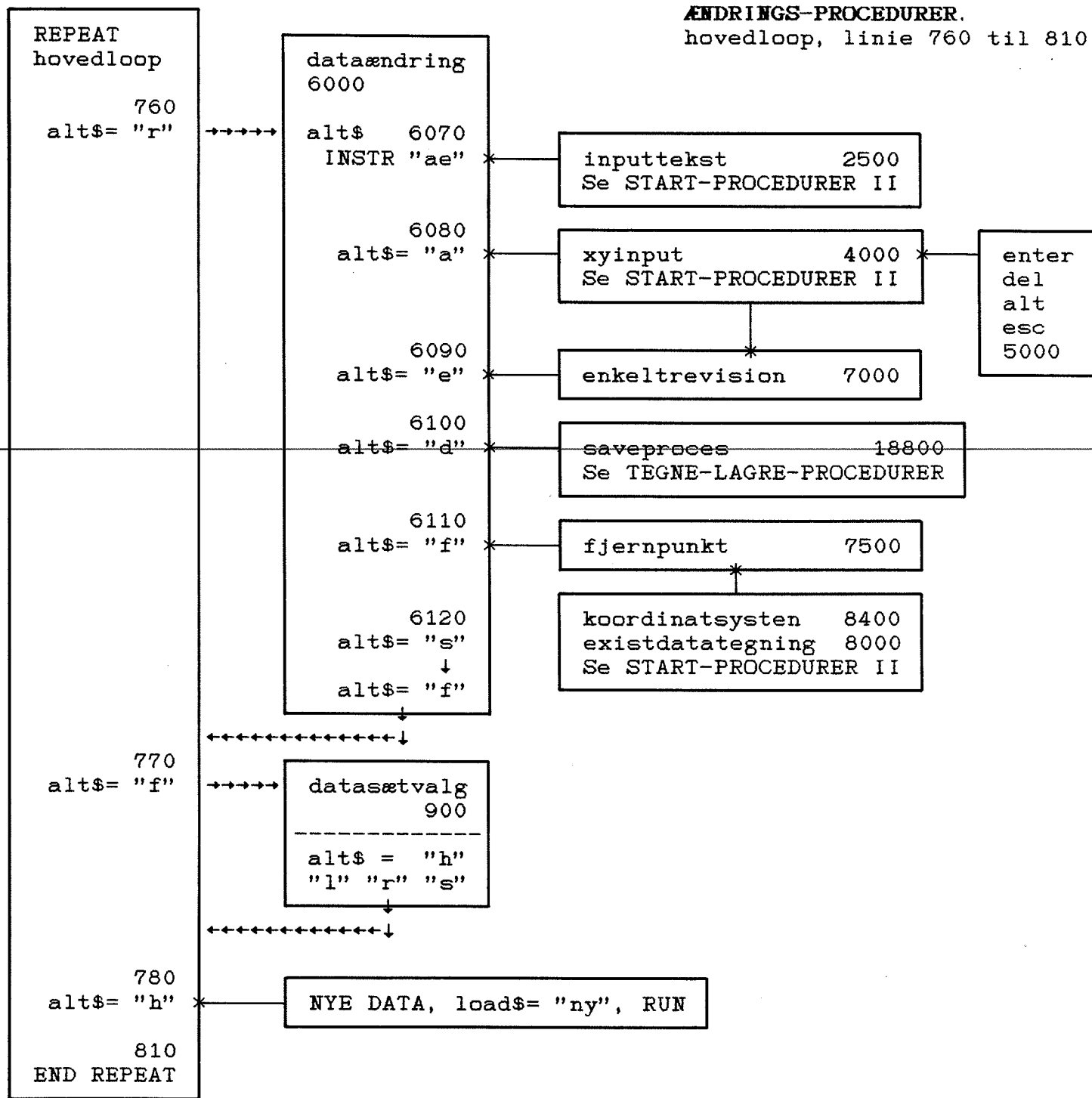
alt\$ = "a" giver alle numre fra start til slut,
med <ESC> som en mulighed undervejs.

alt\$ = "e" fungerer på samme måde, med start i det opgivne nummer.

Se dog kommentarerne til linie 2500 ff. <ALT> er ikke mulig, hvis
alt\$ = "e" - naturligvis.

ÆNDRINGS-PROCEDURER.

hovedloop, linie 760 til 810.



cifertal, svarput, talput VISES IKKE.

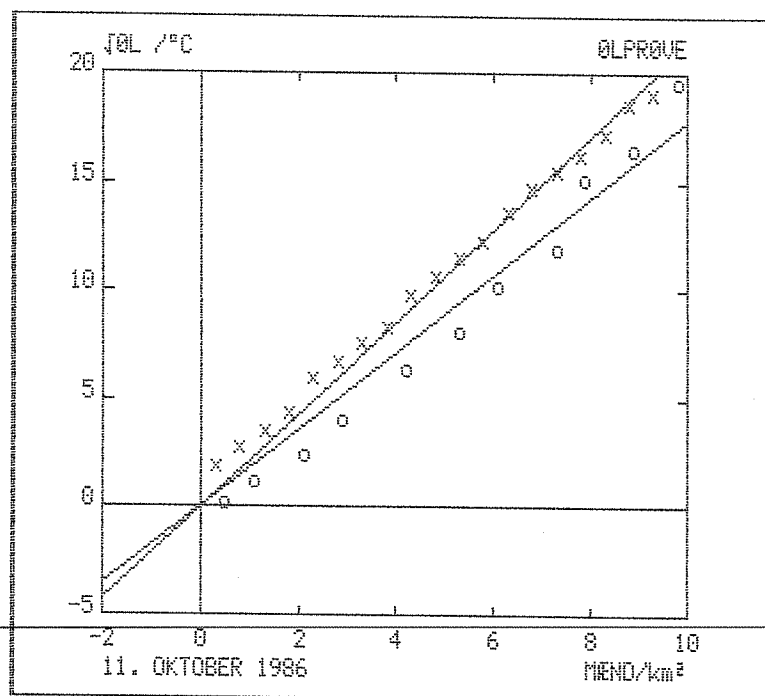
start/slut

+++++

envejsflow

*-----

tovejsflow



RET LINIE GENNEM PUNKT X,Y =
0.000 , 0.000

FILENS NAVN : ØLPRØVE
EDB-DATO : 15.DEC.1986: 10.5
FIGURNAVN : ØLPRØVE
FIGUR-DATO : 11. OKTOBER 1986

PROCEDURE-OVERSIGHT

OVERSIGT OVER PROCedurer mv EFTER linienummer:

```

100 REMark          PROGRAM dth_regression  (SERVOGOR 281)
600 REPEAT          hovedloop
900 DEFINE PROCEDURE datasætvalg
1000 DEFINE PROCEDURE akseinput
2000 DEFINE PROCEDURE start_slut
2500 DEFINE PROCEDURE inputtekst
3000 DEFINE PROCEDURE kontrollinie
4000 DEFINE PROCEDURE xyinput
5000 DEFINE PROCEDURE enterdelaltesc
6000 DEFINE PROCEDURE dataændring
7000 DEFINE PROCEDURE enkeltrevision
7500 DEFINE PROCEDURE fjernpunkt
8000 DEFINE PROCEDURE existdatategning
8400 DEFINE PROCEDURE koordinatsystem
9000 DEFINE PROCEDURE tegnpolygon
10000 DEFINE PROCEDURE regrestype
11000 DEFINE PROCEDURE fejludjævning
12500 DEFINE PROCEDURE akseparametre (xparam, yparam)
13000 DEFINE PROCEDURE udtegning
13300 DEFINE PROCEDURE tegning_lagring
13600 DEFINE PROCEDURE printerudskrift
14000 DEFINE PROCEDURE udjævnudskrift
14500 DEFINE PROCEDURE inputudskrift
15000 DEFINE PROCEDURE printerstreng (strg$)
15200 DEFINE PROCEDURE ciffertal (tal, cifre)
15400 DEFINE PROCEDURE talput (kanal, linie, tab, min, max, ent, slut)
15800 DEFINE PROCEDURE svarput (kanal, linn, tab, strg$, kontrast, papir, blæk)
16000 DEFINE PROCEDURE tidspunkt
17000 DEFINE PROCEDURE filservice (ch)
17500 DEFINE PROCEDURE måned_til_streng
18000 DEFINE PROCEDURE sluttid
18200 DEFINE PROCEDURE loadeproces
18500 DEFINE PROCEDURE xykanal4
18800 DEFINE PROCEDURE saveproces
19500 DEFINE PROCEDURE udjævnfilservice
19600 DEFINE PROCEDURE sortertal (kanal)
20000 DEFINE PROCEDURE filcheck (ordre)
21000 DEFINE PROCEDURE kontrolkurve
22000 DEFINE PROCEDURE tegnkurve
22200 DEFINE PROCEDURE plotkurve
22400 DEFINE PROCEDURE beregny
22800 DEFINE PROCEDURE print_plottekst
23000 DEFINE PROCEDURE sqrtegn
24000 DEFINE PROCEDURE plotteropen
26000 DEFINE PROCEDURE plotterakser
26500 DEFINE PROCEDURE rediger (lin)
27000 DEFINE PROCEDURE udskriv (lin)
27500 DEFINE PROCEDURE plottertekst
28000 DEFINE PROCEDURE plotterstreng (su$, csz, xsu, ysu)
28500 DEFINE PROCEDURE plotterpunkter
29000 DEFINE PROCEDURE plotterregres
29500 DEFINE PROCEDURE plotterregres2
29800 DEFINE PROCEDURE plotterlinie (x1, y1, x2, y2)
30000 DEFINE PROCEDURE udjævn_vindue
30200 DEFINE PROCEDURE redaktion
30300 DEFINE PROCEDURE dobb_vindue
30400 DEFINE PROCEDURE c80dir (pdir$)
30500 DEFINE PROCEDURE c80stat (pstat$)
30600 DEFINE PROCEDURE c80list (linie1, linie2)
30700 DEFINE PROCEDURE c80program

```

ALFABETISK OVERSIGT OVER PROCedurer mv :

```

100 REMark          PROGRAM dth_regression  (SERVOGOR 281)
1000 DEFine PROCedure akseinput
12500 DEFine PROCedure akseparametre (xparam,yparam)
22400 DEFine PROCedure beregny
15200 DEFine PROCedure ciffertal (tal,cifre)
30400 DEFine PROCedure c80dir (pdir$)
30600 DEFine PROCedure c80list (linie1,linie2)
30700 DEFine PROCedure c80program
30500 DEFine PROCedure c80stat (pstat$)
  900 DEFine PROCedure datasætvalg
  6000 DEFine PROCedure dataændring
30300 DEFine PROCedure dobb_vindue
  7000 DEFine PROCedure enkeltrevision
  5000 DEFine PROCedure enterdelaltesc
  8000 DEFine PROCedure existdatategning
11000 DEFine PROCedure fejludjævning
20000 DEFine PROCedure filcheck (ordre)
17000 DEFine PROCedure filservice (ch)
  7500 DEFine PROCedure fjernpunkt
  600 REPEAT        hovedloop
  2500 DEFine PROCedure inputtekst
14500 DEFine PROCedure inputudskrift
21000 DEFine PROCedure kontrolkurve
  3000 DEFine PROCedure kontrollinie
  8400 DEFine PROCedure koordinatsystem
18200 DEFine PROCedure loadeproces
17500 DEFine PROCedure måned_til_streng
22200 DEFine PROCedure plotkurve
24000 DEFine PROCedure plotteropen
26000 DEFine PROCedure plotterakser
29800 DEFine PROCedure plotterlinie (x1,y1,x2,y2)
28500 DEFine PROCedure plotterpunkter
29000 DEFine PROCedure plotterregres
29500 DEFine PROCedure plotterregres2
28000 DEFine PROCedure plotterstreng (su$,csz,xsu,ysu)
27500 DEFine PROCedure plottertekst
15000 DEFine PROCedure printerstreng (strg$)
13600 DEFine PROCedure printerudskrift
22800 DEFine PROCedure print_plottekst
30200 DEFine PROCedure redaktion
26500 DEFine PROCedure rediger (lin)
10000 DEFine PROCedure regrestype
18800 DEFine PROCedure saveproces
18000 DEFine PROCedure sluttid
19600 DEFine PROCedure sortertal (kanal)
23000 DEFine PROCedure sqrtegn
  2000 DEFine PROCedure start_slut
15800 DEFine PROCedure svarput (kanal,linn,tab,strg$,kontrast,papir,blæk)
15400 DEFine PROCedure talput (kanal,linie,tab,min,max,ent,slut)
22000 DEFine PROCedure tegnkurve
13300 DEFine PROCedure tegning_lagring
  9000 DEFine PROCedure tegnpolygon
16000 DEFine PROCedure tidspunkt
19500 DEFine PROCedure udjævnfilservice
30000 DEFine PROCedure udjævn_vindue
14000 DEFine PROCedure udjævnudskrift
27000 DEFine PROCedure udskriv (lin)
13000 DEFine PROCedure udtegning
  4000 DEFine PROCedure xyinput
18500 DEFine PROCedure xykanal4

```


BOOT-PROGRAM "FEJLBOOT"
BOOT2-PROGRAM "FEJLBOOT2"

```

100 REMark *** fejlboot til dth_regression ***
110 flp$= 'FLP1_': DATA_USE flp1_: DEST_USE flp1_: PROG_USE flp1_
120 PRINT#2; '\ ' TAST PRINTERTYPE: '\ ' QL-PRINTER q'
130 PRINT#2; ' CPA80-PRINTER c'
140 REPEAT p
150 INK#2;5: OVER#2;-1: AT#2;3,22: PRINT#2; '?': pr$= INKEY$(12)
160 IF LEN(pr$)=1 AND pr$ INSTR 'qc': IF INKEY$(20) <> pr$: EXIT p
170 END REPEAT p: OVER#2;0: INK#2;2: AT#2;3,22: PRINT#2; pr$
180 PRINT#2; '\ ' PLOTTERTYPE: SERVOGOR 281': TK2_EXT
190 dansk0= RESPR (5224): LBYTES qltegn0, dansk0
200 dansk1= dansk0 + 876: LBYTES qltegn1, dansk1
210 FOR n=0 TO 2: CHAR_USE#n; dansk0, dansk1
220 PRINT#2; '\ ' qltegn LOADED 0 0 0 0 0 0 0 0
230 STRIP 7: INK 2: AT 1,1: PRINT 'FOR INPUT I PROGRAM regression GÆLDER: '
240 AT 3, 1: PRINT 'TAL OG STRENGE INDTASTES VED CURSOR. '
250 AT 11,1: PRINT 'INPUT I FORM AF EEN KARAKTER / EET TAL '
260 PAPER 2: INK 7: AT 5,2: PRINT 'DE KAN REVIDERES FØR < ENTER >.'
270 PRINT ' I EN DEL TILFÆLDE VIL EXISTERENDE DATA'
280 PRINT ' BLIVE UDSKREVET. DISSE KAN ACCEPTERES'
290 PRINT ' VED TASTNING AF < ENTER > ALENE.'
300 PRINT ' REVISIONER MÅ INDTASTES FULDT UD.'
310 PRINT '\ '\ ' fx SVAR PÅ ET SPØRGSMÅL / KOMMANDO'
320 PRINT ' SKAL INDTASTES UDEN BRUG AF ENTER.'
330 PRINT '\ '\ ' OM MULIGT CHECKES INPUT FOR FEJL.'
340 IF pr$=='c': KOPI= dansk0 + 1464: LBYTES scopy_bin, KOPI: CALL KOPI
350 IF pr$=='q': KOPI= dansk0 + 1464: LBYTES qcopy_bin, KOPI: CALL KOPI
360 IF pr$=='c': PRINT#2; ' scopy_bin LOADED': BAUD 4800: TRA 1
370 IF pr$=='q': PRINT#2; ' qcopy_bin LOADED'
380 ramf= dansk0 + 2900: LBYTES ram_Ver1, ramf: CALL ramf
390 PRINT#2; ' RAM_ver1 LOADED'
400 prik= dansk0 + 4680: LBYTES nytpatch, prik: CALL prik
410 PRINT#2; ' pointpatch LOADED'
420 STRIP#2;2: INK#2;7: INK 2: FORMAT ram2_2: REMark USYNLIG FORMAT
430 OPEN_NEW#5; ram2_: PUT#5; 'ny', pr$, dansk0, dansk1: CLOSE#5
440 AT#2;17,2: PRINT#2; ' TAST < ENTER > FOR AT FORTSÆTTE ': INK 5: INK#2;5
450 n= 0: REPEAT loop
460 OVER-1: OVER#2;-1: AT 13,38: PRINT '?': AT#2;17,8: PRINT#2; '< ENTER >'
470 OVER 0: n= n + 1: BLOCK 7,8,227,51,2 + (n MOD 2) * 5
480 AT#2;17,36: b$= INKEY$(12): IF b$<>'': EXIT loop
490 END REPEAT loop: OVER 0: INK 7: OVER#2;0: INK#2;2: PAPER#2;7
500 IF pr$=='c' THEN
510 CLS#2: PRINT#2; '\ ' ER PRINTEREN TÆNDT J / N': INK#2;5: OVER#2;-1
520 REPEAT 12
530 AT#2;1,30: PRINT#2; '?': b$=INKEY$(12): IF b$<>' ' AND b$=='j': EXIT 12
540 END REPEAT 12: OVER#2;0: INK#2;2: AT#2;1,30: PRINT#2; b$
550 EW FLP1_cp80græsk_bin
560 AT#2;5,2: PRINT#2; 'QL-DATAMATEN & CPA80-PRINTEREN'
570 PRINT#2; ' HAR NU KARAKTERSÆTTET "qltegn"'
580 INK#2;0: PRINT#2; '\ ' SLUK ALDRIG FOR PRINTEREN !': INK#2;2
590 PRINT#2; '\ ' SKER DET ALLIGEVEL VED EN FEJLTAGELSE'
600 PRINT#2; ' KAN "qltegn" GEN-LOADES MED <ALT> + q'
610 PRINT#2; '\ ' BOGSTAVER PÅ SKÆRM OG PRINTER ER ENS,'
620 PRINT#2; ' MEN PLOTTEREN ACCEPTERER KUN 0 0 0 0 0 0 0 0
630 AT#2;17,2: PRINT#2; ' TAST < ENTER > FOR AT FORTSÆTTE ': INK#2;5
640 REPEAT 13
650 OVER#2;-1: AT#2;17,8: PRINT#2; '< ENTER >'
660 b$= INKEY$(12): IF b$<>'': EXIT 13
670 END REPEAT 13: OVER#2;0: INK#2;2: INK 7
680 END IF
690 LRUN fejlboot2

```

```

100 LBYTES regres_scr,131072: MRUN dth_regression
110 REMark DETTE ER fejlboot2

```

Programmet "fejlboot" starter hovedprogrammet "dth_regression".

RENAME til "boot" vil medføre automatisk indlæsning ved opstart, men er upraktisk, hvis man fra samme diskette vil have mulighed for brug af begge de to plottere.

Det udfører en række indledende manøvrer:

- FLP1 gøres til DEFAULT-diskette (linie 110).
- Printertypen fastlægges, pr\$ (linie 150).
- TOOLKIT2 er forudsat etableret permanent (linie 180).
TK2_EXT sikrer, at TOOLKIT2's kommandoer "overlejrer" eventuelle identiske navne som eksisterer i INTERFACE's EXTRA. Samtlige nye kommandoer kan udskrives med "EXTRAS".
- "qltegn" indlæses (875 + 587 bytes til skærmtegn, linie 190 til 210).
- "scopy_bin" / "qcopy_bin" indlæses. Skærmkopi til CPA80S- hhv QL-printer (linie 340 f).
- "ram Ver1" indlæses. RAM kan nu FORMATTERES og bruges som man en FLP (linie 380).
- "nytpatch" indlæses. Den danske QL-version tegner nu korrekt, een pixel som svar på POINT (linie 400).
- FORMAT RAM2_2 (linie 420). Heri lagres variablen "ny", dvs load\$ samt pr\$ og adresserne til "qltegn", som alle skal overføres til hovedprogrammet, se dette linie 110.
- information om programmets angivelse af de to slags INPUT, der demonstreres ved blinkende spørgsmålstegn / rektangel (linie 230 ff og 450 ff.
- for CPA80S-printeren DOWNLOADES "græsk karaktersæt" med tegn, programmet bruger, se senere, fx linie 15020 i hovedprogram. ("EW FLP1_cp80græsk_bin" i linie 550).
- sluttelig "LRUN fejlboot2". Program og variabler forsvinder, bortset fra de, der er lagret i RAM.

Når linie 220 og 620 har de korrekte specialtegn, skønt der er brugt QL-printer til programlistningen, skyldes det, at jeg for overskuelighedens skyld har indklæbet disse.

Programmet "fejlboot2" fungerer som et "pauseprogram".

Mens hovedprogrammet indlæses, 6 minutter, vises et uddrag af de væsentligste muligheder i programmet. Skærm billedet er et lille program "regresskærm", se de følgende sider, som her indlæses i maskinkodet form, så det opbygges hurtigt.

Samtidig har jeg ved kommandoen "MRUN dth_regression" opnået, at hovedprogrammet MERGES ind og derfor ikke sletter skærm billedet. Se i IFH-rapport 177 hvorledes en serie skærm billeder momentant kan afløse hinanden i instruktionssekvensen.

PAUSEPROGRAM "REGRESSKÆRM"

INPUT ER :

JM-P 1986

1. KOORDINATER X,Y TIL PUNKTER MED FORTLØBENDE NUMRE 00 ... 100.
2. EN KONTROL-POLYGON (MAX. 8 PUNKTER) / KONTROL-KURVE (3-LEDDET FORMEL)
3. KOORDINATSYSTEMETS ENHEDER OG (LINEÆRE) AKSE-INDELINGER.

PUNKTERNE 00 ... 100 KAN INDIVIDUELT MARKERES MED o ELLER x .
 DATA-SÆTTET KAN INDTASTES, INDLÆSES, UDVIDES, REVIDERES, FEJLUDJÆVNES ETC.
 KOORDINATER ER STRENG-VARIABLER ! INDTAST EVT. DECIMALER KONSEKVENT.

OUTPUT ER:

FEJL-UDJÆVNING FOR x / FOR o / FOR x OG o UNDER ET / TO LINIER.
 GENTAGNE UDTEGNINGER AF KOMBINATIONER AF x OG o.

FEJL-UDJÆVNING BETYDER EN RET LINIE, MINDSTE KVADRATERS METODE, SOM :

- A. LINEÆR REGRESSION.
- B. LODRET / VANDRET LINIE.
- C. RET LINIE GENNEM GIVET PUNKT.
- D. RET LINIE MED GIVEN HÆLNING.

UDTEGNING PÅ PRINTER / PLOTTER.

UDSKRIFT / LAGRING AF DATASÆT OG FEJL-UDJÆVNING PÅ PRINTER / FLOPPY.

PROGRAM "regresskærm"

OK for SBYTES.

For SCOPY anbefales REMark ved DELETE og SBYTES,
samt fjernelse af to gange STRIP mv.

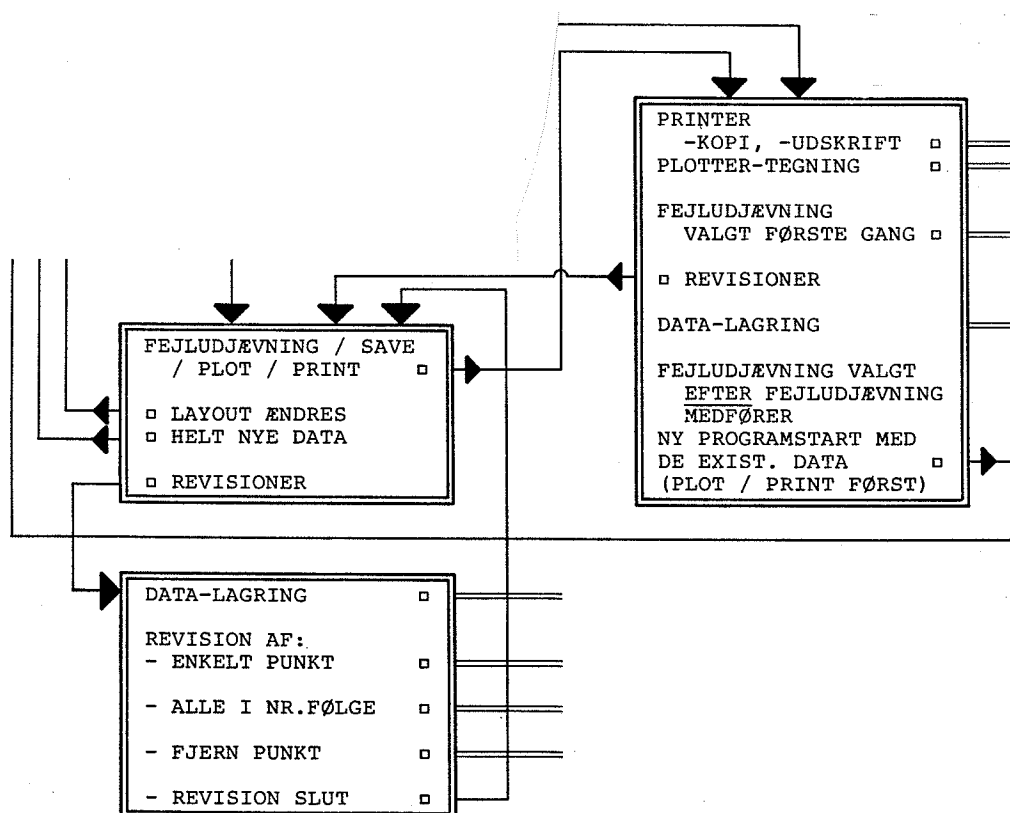
```

100 SCALE 100,0,0:MODE 4
110 WINDOW 512,256,0,0: PAPER 2:CLS
120 WINDOW 504,252,4,2: PAPER 2: INK 7: BORDER 1,7: CLS: BORDER 5
130 WINDOW#2; 436,202,10,10: PAPER#2;2: INK#2;7
140 WINDOW#0; 436, 42,10,202:PAPER#0;2: INK#0;7
150 k=1: AT#k;0,69: PRINT#k;'@ JM-P 1986'
160 STRIP#k;7: INK#k;2: CURSOR#k;0,100,18,3: PRINT#k;' INPUT ER : ': PAPER#k;
2: INK#k;7: PRINT#k
170 AT#k;2,3: PRINT#k;'1. KOORDINATER X,Y TIL PUNKTER MED FORTLØBENDE NUMRE
00 ... 100.'
180 PRINT#k;' 2. EN KONTROL-POLYGON ( MAX. 8 PUNKTER ) / KONTROL-KURVE ( 3
-LEDDET FORMEL )'
190 PRINT#k;' 3. KOORDINATSYSTEMETS ENHEDER OG ( LINEÆRE ) AKSE-INDDDELINGE
R.'
200 PRINT#k;''
210 PRINT#k;' PUNKTERNE 00 ... 100 KAN INDIVIDUELT MARKERES MED ELLER
,
220 STRIP#k;0: INK#k;7: AT#k;6,53: PRINT#k;' o ': AT#k;6,63: PRINT#k;' x ': P
APER#k;2: INK#k;7
230 PRINT#k;' DATA-SÆTTET KAN INDFASTES, INDLÆSES, UDVIDES, REVIDERES, FEJL
UDJÆVNES ETC.'
240 PRINT#k;' KOORDINATER ER STRENG-VARIABLER ! INDFAST EVT. DECIMALER KONS
EKVENT.'
250 BORDER 0: BLOCK 500,1,2,102,7
260 BORDER 0: BLOCK 504,8,0,103,2
270 BORDER 0: BLOCK 500,1,2,111,7: BORDER 5
280 STRIP#k;7: INK#k;2: CURSOR#k; 0,100,18,113: PRINT#k;' OUTPUT ER: ': PAPER
#k;2: INK#k;7: PRINT#k
290 AT#k;13,3: PRINT#k;'FEJL-UDJÆVNING FOR x / FOR o / FOR x OG o UNDER
ET / TO LINIER.'
300 PRINT#k;' GENTAGNE UDTEGNINGER AF KOMBINATIONER AF x OG o.'
310 PRINT#k;' FEJL-UDJÆVNING BETYDER EN RET LINIE, MINDSTE KVADRATERS METOD
E, SOM :
320 PRINT#k;' A. LINEÆR REGRESSION.'
330 PRINT#k;' B. LODRET / VANDRET LINIE.'
340 PRINT#k;' C. RET LINIE GENNEM GIVET PUNKT.'
350 PRINT#k;' D. RET LINIE MED GIVEN HÆLDNING.'
360 PRINT#k;''
370 PRINT#k;' UDTEGNING PÅ PRINTER / PLOTTER.'
380 PRINT#k;' UDSKRIFT / LAGRING AF DATASÆT OG FEJL-UDJÆVNING PÅ PRINTER
/ FLOPPY.'
390 DELETE regres_scr
400 SBYTES regres_scr,131072,32768
410 WINDOW 512,256,0,0:BORDER 0:RECOL 0,0,0,0,0,0,7,7
420 SCOPY: PAUSE 200

```

linie 400 131072 er skærbilledadressen.
 32768 er længden.

HOVEDPROGRAM "REGRESSION"



```

100 REMark ***** PROGRAM dth_regression ***** (SERVOGOR 281) *****
110 OPEN#7;ram2_: GET#7; load$, pr$, dansk0,dansk1: CLOSE#7
120 trat= 1: trap= 0: baudtal=9600: IF pr$== 'c': trap= 1: baudtal=4800
130 OVER#0;0: BORDER#0;0: INK#0;7: ktr$='i': alt$='f'
140 BAUD baudtal: TRA trap: ALTKEY 'q','ew cp80græsk_bin',''
190 :
200 IF load$= 'gl' THEN
210 CSIZE#4;0,0: udjævn_vindue: koordinatsystem: existdatategning
220 CLS#4 : kontrollinie : CLS#4 : IF ktr$=='p': tegnpolygon
230 OPEN#3,ser1: BPUT#3;27,108,10,27,81,77,27,67,72,27,78,10: CLOSE#3
240 END IF
300 IF load$= 'ny' THEN
310 n= 0: DIM p$(110,1): DIM g$(6,7): FOR n=0 TO 6: g$(n)='0'
320 DIM x$(110,7): DIM y$(110,7): FOR n= 0 TO 110: x$(n)='0': y$(n)='0'
330 DIM rgrnr(100): DIM udj$(40,24): DIM fjl(40)
340 alt$='i': dato$='': dat$='': filnavn$='': fil$='': fldat$=' '
350 hæld$='': ktr$='i': ox$= 'o': pdir$=' '
360 pkt$= 'ox': pstat$='': q$='': rg$='': sl$='': st$='': strg$=' '
370 tal$='': valg$='': xe$='': ye$='': z$=' '
380 blæk= 0: ch= 0: cifre= 0: ent= 0: kanal= 0
390 kontrast= 0: ktrtal= 0: linie= 0: linie1= 0: linie2= 0: linn= 0
400 max= 0: min= 0: nygl= 0: n= 0: ørdre= 0: papir= 0: pt= 0
410 p= 0: rgoantal= 0: rgtal= 0: rgxantal= 0: slutnr= 0: startnr= 0
420 starttid= 0: tab= 0: tal= 0: ttal= 0: xakse= 0: xantal=0
430 xfast= 0: xhæld= 0: xnedad= 0: xopad= 0: xparam= 0: xs= 0: yakse= 0
440 yantal= 0: yparam= 0: yfast= 0: yhjøre= 0: yvenst= 0: yskala= 0
450 dobb_vindue : MODE 4 : OVER 0: tidspunkt
460 udjævn_vindue: FOR n= 0,1,4,8: CLS#n
470 IF nygl= 2 : loadeproces: alt$= 'l'
480 load$='gl': OPEN#7;ram2_: PUT#7 \ 0,'gl',pr$,dansk0,dansk1: CLOSE#7
490 END IF

600 REPEAT hovedloop
610 IF alt$ INSTR 'il' THEN
620 FOR n= 0,1,4,8: CLS#n
630 ktr$= 'i' : akseinput: inputtekst: pt= startnr - 1: xyinput
640 nygl= 2 : FOR n=0,4,8: PAPER#n;2: INK#n;7: CLS#n
650 END IF : REMark EXIT MED alt$= 'r' 's'
660 IF alt$== 's' THEN
670 DIM rgrnr(100): DIM udj$(40,24): DIM fjl(40)
680 tegning_lagring: REMark EXIT MED alt$= 'f' 'r'
690 IF alt$== 'f': regrestype : tegning_lagring
700 IF valg$== 'o+x': rg$= 'x': fejludjævning: tegning_lagring
710 REMark EXIT MED alt$= 'f' 'r'
720 hæld$= '': rg$= '': valg$= ' '
730 IF alt$== 'f': RUN
740 IF alt$== 'r': alt$= 'f' : REMark EXIT MED alt$= 'f'
750 END IF
760 IF alt$== 'r': datasændring: REMark EXIT MED alt$= 'f'
770 IF alt$== 'f': datasætvalg: REMark EXIT MED alt$= 'r' 's' 'l' 'h'
780 IF alt$== 'h' THEN
790 OPEN#7; ram2_: PUT#7 \ 0,'ny',pr$,dansk0,dansk1: CLOSE#7: RUN
800 END IF
810 END REPEAT hovedloop

900 DEFINE PROCEDURE datasætvalg
910 CLS#0: PRINT#0;' - FEJLUDJÆVNING / SAVE'\ ' / PLOT / PRINT f'
920 PRINT#0;' - LAYOUT ÆNDRES l' \ ' - REVISIONER r'
930 PRINT#0;' - HELT NYE DATA h': svarput 0,4,24,'fsplrh',0,2,7
940 alt$= q$: IF alt$ INSTR 'fesp': alt$= 's'
950 END DEFINE datasætvalg
990 :

```

linie 100 ff Som beskrevet i rapporterne 177 .. 180 overføres variabler fra boot via RAM. Der etableres mulighed for ny download i CPA80S.

linie 200 ff Ved STOP og ny RUN af et program sikres, at variabler bevares, men (nyt) layout mv skal genindtastes. Se også linie 730.

linie 300 ff Her nulstilles - initialiseres - en række

GLOBALE VARIABLER:

p\$(110) Symboler for punkter, "o" "x" "o"
 g\$(6,7) Værdier for akseinddelinger, se "akseinput", linie 1000.
 x\$(110,7), y\$(110,7) x,y for punkter, lagret som strenge.

Jeg henviser til OVERSIGT OVER VARIABLER, VARIABELLISTE, pag. 105 ff, hvor der også angives LOCAL variabler etc, etc.

Følgende væsentlige variabler skal dog kort omtales her:

load\$ "ny" ved programstart / nyt datasæt. Giver DIM, nulstillede / definerede variabler.
 "gl" sikrer, at dette ikke sker ved en ny RUN.
 nygl 1 = ny fil oprettes / 2 = fil eksisterer.
 pkt\$ "o" "x" "ox" dvs de aktuelt benyttede symboler.
 ox\$ sikrer, at alle punkter, indtastet i nummerfølge, får samme symbol, indtil der bevidst skiftes.
 alt\$ styrer valg af PROCedurer, se variabellisten.
 For fejludjævning se bl.a linie 2500 ff.
 valg\$ styrer, hvilke punkter, der skal udtegnas / indgå i en fejludjævning, se linie 10030 ff.
 rg\$ styrer, hvilke punkter der analyseres, se linie 10030 ff.
 rgtal antal punkter i fejludjævningen, se linie 11060 ff.
 rgoantal antal punkter med symbol o, se linie 10260.
 rgxantal antal punkter med symbol x, se linie 10260.
 hæld\$ angiver regrestypen, se linie 10360 ff.

udj\$(40,24) k= 0 / 20. Udjævnings-udskrift på skærm, se linie 11700.
 fjl(40) k= 0 / 20. Udjævnings-DATA til SAVE og plotter, se linie 10260, 19270.
 rgrnr(100) k= 0 / 50. Numre til PROCedure "sortertal", linie 10250.

linie 600 "hovedloop", se PROCEDURE-OVERSIGT: START-PROCEDURER II, TEGNE-LAGRE-PROCEDURER, ÆNDRINGS-PROCEDURER.

DATA-LAGRING	d
TEGNING PÅ PLOTTER	t
PRINTER-KOPI,-DATA	p
FEJLUDJÆVNING	f
REVISIONER	r ?

SKIRMBILLEDE: KOPI	s
INPUTDATA-UDSKRIFT	i
UDJÆVNING-UDSKRIFT	u
RETUR TIL PROGRAM	r ?

DATA-LAGRING	d
REVISION - SLUT	s
- ENKELTE PUNKTER	e
- ALLE I NR. FØLGE	a
FJERN PUNKT	f ?

linie 900 ff Se PROCEDURE-OVERSIGT: ÆNDRINGS-PROCEDURER.

- FEJLUDJÆVNING / SAVE	
/ PLOT / PRINT	f
- LAYOUT ÆNDRER	l
- REVISIONER	r
- HELT NYE DATA	h ?

```

1000 DEFINE PROCEDURE akseinput
1010 LOCAL ny$, figurloop, loop2, loop3, stsl, k, glst$, glsl$
1020 REPEAT figurloop
1030   IF nygl= 2 THEN
1040     AT 3,1: PRINT'TIDLIGERE VÆRDIER ( ACCEPTERES MED < ENTER > ) ='
1050     FOR n= 1 TO 6: AT 4 + n, 48: PRINT g$(n)
1060   END IF
1070   PRINT#4; \ ' X- OG Y-AKSERNE: ' \ \ ' Højest 5-7 inddelinger'
1080   PRINT#4; \ ' X   minimum = ' \ ' X   maximum = ' \ ' Inddeling pr.'
1090   PRINT#4; \ ' Y   minimum = ' \ ' Y   maximum = ' \ ' Inddeling pr.'
1100   REPEAT loop2
1110     REPEAT loop3
1120       FOR n= 1 TO 6
1130         talput 4,4+n,16,-32000,32000,1,0: IF ttal >=-32000: g$(n)=tal$
1140         AT#4;4 + n,16: PRINT#4; g$(n)
1150       END FOR n: PRINT#4; \ ' KORREKT j / n'
1160       svarput 4,12,17,'jn',0,2,7: IF q$== 'j': EXIT loop3
1170       FOR n= 1 TO 6: AT#4;4 + n,16: PRINT#4; FILL$( ' ',6)
1180     END REPEAT loop3
1190     k= 0: IF g$(2)-g$(1) >= .1 AND g$(5)-g$(4) >= .1: k= 1
1200     IF k= 1 AND g$(3) > 0 AND g$(6) > 0 THEN
1210       xantal= (g$(2)-g$(1)) /g$(3): yantal= (g$(5)-g$(4)) /g$(6)
1220       IF xantal==INT(xantal) AND yantal==INT(yantal): EXIT loop2
1230     END IF : FOR n= 1 TO 8: AT#4;4 + n,16: PRINT#4; FILL$( ' ',6)
1240   END REPEAT loop2

1260   AT#4;11,0: CLS#4,2: AT#4;13,1: PRINT#4;'FORTLØBENDE PUNKTNUMRE:'
1270   PRINT#4;' 0 - 100, DER UDSKRIVES' \ ' 00,01,...10,11....99,100'
1280   REPEAT stsl
1290     ordre= 1: filcheck ordre: start_slut: ordre= 2: filcheck ordre
1300     IF NOT ordre: EXIT stsl
1310   END REPEAT stsl
1320   OVER 0: AT 17,48: PRINT'   ': AT 18,48: PRINT'   ': OVER 1
1330   AT#4;11,0: CLS#4,2: AT#4;12,1: PRINT#4;'X-AKSENS ENHED:'
1340   PRINT#4; \ ' Y-AKSENS ENHED:': AT 13,52 - LEN(xe$): PRINT xe$
1350   PRINT#0; \ ' EKSPONENTER FAS VED: ' \ ' 0 = CTRL + SHIFT + 0'
1360   PRINT#0; \ ' 2 = CTRL + SHIFT + 2' \ ' 3 = CTRL + SHIFT + 3'
1370   PRINT#0; \ ' 4 = CTRL + SHIFT + 4'
1380   AT#4;13,1: INPUT#4; ny$: IF LEN(ny$) AND ny$ <> ' ': xe$= ny$
1390   AT#4;13,1: PRINT#4; xe$: AT 15,52 - LEN(ye$): PRINT ye$
1400   AT#4;15,1: INPUT#4; ny$: IF LEN(ny$) AND ny$ <> ' ': ye$= ny$
1410   AT#4;15,1: PRINT#4; ye$: AT 17,52 - LEN(fil$): PRINT fil$
1420   AT#4;16,1: INPUT#4; 'FIGURTEKST:' \ ' '; ny$
1430   IF LEN(ny$) AND ny$ <> ' ': fil$= ny$
1440   AT#4;17,1: PRINT#4; fil$: AT 19,52 - LEN(dat$): PRINT dat$
1450   AT#4;18,1: INPUT#4; 'Figur DATO:' \ ' '; ny$
1460   IF LEN(ny$) AND ny$ <> ' ': dat$= ny$
1470   AT#4;19,1: PRINT#4; dat$

1490   yskala= g$(5) - g$(4): xs= (g$(2) - g$(1)) / (g$(5) - g$(4))
1500   SCALE 1.25 * yskala, g$(1) /xs -.15 * yskala, g$(4) -.15 * yskala
1510   koordinatsystem: existdatategning: kontrollinie: CLS#4
1520   IF ktr$== 'p': tegnpolygon
1530   CLS#0: AT#0; 1,1: PRINT#0;'KORREKT FIGUR j / n'
1540   svarput 0,2,20,'jn',0,2,7: IF q$== 'j': EXIT figurloop
1550   IF q$== 'n': ktr$= 'i': FOR n= 0,1,4,8: CLS#n
1560 END REPEAT figurloop
1570 END DEFINE akseinput

```

- linie 1000 ff PROCEDURE "akseinput"
Figurstørrelse, akseinddelinger, punktnumre,
figurtekster, enheder mv fastlægges.
- linie 1100 ff Evt. eksisterende (nygl= 2) værdier af akseinddelinger udskrives i 1050. Kan revideres / accepteres. PROCEDURE "talput" (15400) har næstsidste parameter, ent= 1, dvs. at indtastning af < ENTER > i stedet for et tal bevarer den eksisterende værdi af g\$(n).
- linie 1160 Et eksempel på "fortrydelsesspørgsmål" :
PROCEDURE "svarput" (15800) bruges til at give en chance for ny indtastning.
- linie 1190 ff Kontrol af, at inddelingerne har rimelige størrelser og "går op" i minimums- og maksimumsværdierne. Sære påfund (?) kan ikke fanges, fx inddelinger pr. 1/7 mellem heltal.
- linie 1290 PROCEDURE "filcheck" (20000) sikrer, at brugeren er klar over - ved eksisterende datasæt - at programmet stryger punkter uden for de senest indtastede start- og slutnumre. "Fortrydelsesret" er indbygget. PROCEDURE "start_slut", se linie 2000.
- linie 1350 ff Programmets specialtegn for eksponenter, kvadratrods og grad C udskrives på begge printere og plottere. Andre tegn og symboler kan indlægges, se PROCEDUREN "sqrtegn" (23000). Når de er vist listet på en QL-printer, er det "efterkopiering" fra min side. På en CPA80S-printer findes tegnene, se linie 15020 !
- linie 1380 m.fl. IF LEN(ny\$) AND ny\$ <> ' ' sikrer, at det tastede er alvorligt ment. Stavefejl kan ikke fanges.
- linie 1490 f Tegningen og det kvadratiske felt skaleres, med plads til tekster, enheder og ledsage-kommentarer tilhøjre. Jeg prøvede mig lidt frem - der kunne være regnet.
- linie 1510 PROCEDUREN "kontrollinie" (3000) udtegner ikke selv en polygon, hvis en sådan vælges. Vælges der derimod en kurve, vil PROCEDUREN "kontrolkurve" (21000) bruge PROCEDUREN "tegnkurve" (22000). Derfor linie 1520. Se "existdatategning", "koordinatsystem", "tegnpolygon" (8000,8400,9000), "kontrolkurve", "tegnkurve" (21000, 22000).
- linie 1530 ff "Fortrydelsesmulighed". Kurve-definitionen er som omtalt ovenfor anbragt i PROCEDUREN "kontrolkurve" med egen fortrydelsesret - kurver er svære at definere.

```

2000 DEFine PROCedure start_slut
2010   AT#4;17,1: PRINT#4; 'ANGIV START-NR.='
2020   AT#4;18,8: PRINT#4; 'SLUT-NR.='
2030   IF nygl= 2: AT 17,48: PRINT st$: AT 18,48: PRINT sl$
2040   talput 4,17,20,0,100,1,0: IF ttal >= 0: startnr= ttal
2050   st$= FILL$('0',startnr < 10) & startnr: AT#4;17,20: PRINT#4; st$
2060   talput 4,18,20,0,100,1,0: IF ttal >= 0: slutnr = ttal
2070   sl$= FILL$('0',slutnr < 10) & slutnr : AT#4;18,20: PRINT#4; sl$
2080   PAUSE 50
2090 END DEFine start_slut

2500 DEFine PROCedure inputtekst
2510   PAPER#4;2: INK#4;7: CLS#4: FOR n= 105,133: BLOCK#4; 138,3,6,n,7
2520   AT#4; 1,1: PRINT#4;'INDTAST KOORDINATER TIL' \ ' PUNKT NR.'
2530   AT#4; 4,1: PRINT#4;'EKSISTERENDE VÆRDIER AF' \ ' X,Y ='
2540   AT#4; 8,1: PRINT#4;' NYE VÆRDIER AF X,Y= '
2550   AT#4;11,1: PRINT#4;'PUNKTET TEGNES MED'
2560   IF alt$ INSTR'aeil': AT#4;12,1: PRINT#4;'SKIFT x / o MED'
2570   AT#4;14,1: PRINT#4;'PUNKT:      OK'
2580   AT#4;15,1: PRINT#4;'PUNKT:      FEJL'
2590   IF alt$ INSTR 'l' : AT#4;16,1: PRINT#4;'DATASÆT ÆNDRES'
2600   IF alt$ INSTR'aelw': AT#4;17,1: PRINT#4;'INDTASTN. SLUT'

2620   CLS#0: PAPER#0;7: INK#0;2
2630   IF alt$ <> 'i' THEN
2640     CURSOR#0;0,100, 7, 7: PRINT#0;' FORLAD SEKVENSEN MED: '
2650     CURSOR#0;0,100, 7,17: PRINT#0;'      EFTER X,Y INPUT '
2660     CURSOR#0;0,100, 7,27: PRINT#0;' eller '
2670     CURSOR#0;0,100, 7,37: PRINT#0;'      I ST.F. X-VÆRDI '
2680     PAPER#0;2: INK#0;7: CURSOR#0;0,100,13,37: PRINT#0;'"esc"'
2690     CURSOR#0;0,100,13,17: PRINT#0;'<ESC>'
2700     FOR n= 7,47: BLOCK#0; 138,1,7,n,7
2710   END IF : PAPER#0;2: INK#0;7

2730   AT#8;2,0: PRINT#8;'<ENTER>' \ '< DEL >'
2740   IF alt$ INSTR'aeil': AT#8;0,0: PRINT#8;'<SHIFT>'
2750   IF alt$ INSTR 'l' : AT#8;4,0: PRINT#8;'< ALT >'
2760   IF alt$ INSTR'aelw': AT#8;5,0: PRINT#8;'< ESC >'
2770 END DEFine inputtekst

2800 REMark alt$ = i: NYT DATASÆT, xyinput UDEN <ESC> <ALT> (340)
2810 REMark alt$ = l: EKSISTERENDE DATA, <ESC> <ALT> OK (470)
2820 REMark alt$ = a OG e: ÆNDRINGER, NY <ALT> IKKE MULIG (6080/6090)
2830 REMark alt$ = w (kontrollinie): <SHIFT> OG <ALT> IKKE MULIG (3150)

```

linie 2000 ff PROCEDURE "start_slut", smlg. linie 1290.
 linie 2500 ff PROCEDURE "inputtekst".

Denne procedure - og de tilknyttede - benyttes til at indtaste punkters x, y, symbol, herunder også til at spare tid ved brug af evt. eksisterende DATA, som kan accepteres ved brug af < ENTER >, eller nyindtastes / revideres.

Der er derfor behov for, at brugeren klart kan se de aktuelle muligheder. En ny fil skal fx "fyldes op" med DATA, førend indtast-sekvensen forlades, mens en eksisterende fil måske skal accepteres i sin helhed. Sekvensen kan derfor ikke forlades før alle DATA fra start til slut er indtastet for en ny fil, mens det for en eksisterende fil skal være muligt at forlade sekvensen med < ESC > = resten accepteres, eller med < ALT > = revisioner ønskes.

Se især alt\$ og dennes betydning, linie 2730 .. 2830:
 Skal især ses i sammenhæng med PROCEDURENE "xyinput" (4000, bemærk 4350) og "enterdelaltesc" (5000, bemærk 5030 .. 5060, 5400, 5430).
 Se endvidere "enkeltrevision" (7000, bemærk 7150).
 Smlgn. den analoge, kortere PROCEDURE "kontrollinie" (3000, bemærk 3360, 3470).

Der udskrives evt. eksisterende DATA: x, y, symbol for punktnumre fra start til slut udskrives, ændres / accepteres et for et.

Herefter skal det samlede sæt DATA for punktet enten accepteres (< ENTER >), eller forkastes (< DEL >), for symbolet dog < SHIFT >. For eksisterende DATA er der endvidere muligheden < ESC > = revision afsluttet og < ALT > = revision forfra.

Dette vises ved, at de aktuelle muligheder nu blinker (der benyttes OVER-1, se bl.a IFH-rapport 177):

nyt INPUT	eksist. DATA / nyt layout / revision af alle punkter	revision af enkeltpunkt	kontrol- linie
SHIFT	SHIFT	SHIFT	---
ENTER	ENTER	ENTER	ENTER
DEL	DEL	DEL	DEL
---	ALT	---	---
---	ESC	ESC	ESC
alt\$= "i"	"l" "a"	"e"	"w"
340,	470,610,940,2560 .. 2630,	2560,2760,	2600,
610,	2740 .. 2760,4180 ..4220,	6070,6090,	3010,
2560,	4350,5050,5060,5180,5200,	7070,7080,	3150,
2740	5290,5310,5420,5430,6070,	7160,20300	3560
	6080,20330		

```

3000 DEFine PROCedure kontrollinie
3010 LOCAL a$, b$, gem$, koordinat, xyslut, t: gem$= alt$: t= 0
3020 PAPER#4;2: INK#4;7: AT#4;11,0: CLS#4,2: CLS#0
3030 AT#4;13,1: PRINT#4;'ANTAL BETYDENDE CIFRE'\ ' I ANALYSEUDSKRIFT:'
3040 talput 4,14,21,1,6,0,0: cifre = ttal
3050 AT#4;16,1: PRINT#4;'KONTROL-LINIE ?' \ ' - POLYGON,MAX.8 PKT. p'
3060 PRINT#4;' - KURVE, y = f(x) k' \ ' - INGEN LINIE i'
3070 CLS#0: svarput 0,0,22,'ikp',0,2,7: ktr$= q$: IF ktr$=='i': RETURN
3080 IF nygl= 2 AND p$(102) <> '' THEN
3090 PRINT#0; \ ' NY KONTROLLINIE n' \ ' EKSIST. DATA-SÆT e'
3100 svarput 0,3,22,'en',0,2,7: IF q$== 'e': ktr$= 'p'
3110 IF q$== 'e' AND p$(104)='k': ktr$= 'k': tegnkurve
3120 END IF : IF q$== 'e': CLS#0: RETURN
3130 IF ktr$=='k': ktrtal=5: kontrolkurve: IF ktr$=='k': CLS#0: RETURN

3150 CLS#0: alt$= 'w': inputtekst: pt= 100
3160 REPEAT koordinat
3170 PAPER#4;2: INK#4;7: pt= pt+1: IF pt=109: pt=108: EXIT koordinat
3180 a$= FILL$(' ',23): FOR n= 6,9: AT#4;n,1: PRINT#4; a$
3190 STRIP#4;7: INK#4;2: AT#4;2,11: PRINT#4;' '; pt ; ' '
3200 AT#4;9,1:PRINT#4;' ': AT#4;6,1:PRINT#4;! x$(pt);', ' ;y$(pt);' '
3210 IF p$(pt)= ' ': p$(pt)= 'o'
3220 STRIP#4;0: INK#4;7: AT#4;11,21: PRINT#4; ! p$(pt);' '
3230 STRIP#4;7: INK#4;2: AT#4;8,1: PRINT#4;' NY VÆRDI AF X = '
3240 talput 4,9,2,g$(1),g$(2),1,1
3250 IF ttal<g$(1) AND (x$(pt)-g$(1))*(g$(2)-x$(pt))>=0: tal$=x$(pt)
3260 IF ttal>g$(2) AND pt > 101: p$(pt)=' ': pt= pt-1: EXIT koordinat
3270 x$(pt)= tal$: AT#4; 9,2: PRINT#4; x$(pt);', '
3280 AT#4; 8,1: PRINT#4;' OG NY VÆRDI AF Y = '
3290 talput 4,9,4 + LEN(x$(pt)),g$(4),g$(5),1,0
3300 IF ttal<g$(4) AND (y$(pt)-g$(4))*(g$(5)-y$(pt))>=0: tal$=y$(pt)
3310 y$(pt)= tal$: AT#4; 9,4 + LEN(x$(pt)): PRINT#4; y$(pt);' '
3320 OVER - 1: INK 4: CURSOR x$(pt) / xs, y$(pt), -3, -3: PRINT 'o'

3340 PAPER#4;2: OVER#8; -1: INK#8;5
3350 REPEAT xyslut
3360 AT#8;2,0: PRINT#8; '<ENTER>' \ '< DEL >' \ \ '< ESC >'
3370 b$= INKEY$(15): IF CODE(b$) THEN
3380 IF CODE(b$)= 27 AND pt > 101: EXIT koordinat: REMark ESC
3390 IF CODE(b$)= 10: OVER 1: INK 2 : REMark ENTER
3400 CURSOR x$(pt) / xs, y$(pt), -3, -3: PRINT 'o'
3410 IF CODE(b$)= 194: pt= pt - 1: t= t + 1: REMark DEL
3420 EXIT xyslut
3430 END IF
3440 END REPEAT xyslut
3450 END REPEAT koordinat
3460 FOR n= 4,8: OVER#n;0: PAPER#n;2: INK#n;7
3470 AT#8;2,0: PRINT#8; '<ENTER>' \ '< DEL >' \ \ '< ESC >'
3480 OVER 1: INK 2: CURSOR x$(pt) / xs, y$(pt), -3, -3: PRINT 'o'
3490 IF pt < 108 AND p$(pt+1)='o' THEN
3500 CLS#0: PRINT#0;' LINIE FOR 101 - '; pt ; \ ' EKSIST.DATA OMFATTER'
3510 PRINT#0; ' FLERE PUNKTER ! ' \ ' SLET RESTEN s'
3520 PRINT#0; ' NY INDTAENING n': svarput 0,4,20,'sn',3,2,7
3530 IF q$== 's': FOR p= pt + 1 TO 108: p$(p)= ' ': x$(p)= 0: y$(p)= 0
3540 IF q$== 'n': GO TO 3150
3550 END IF : FOR n= 4,8: OVER#n;0: PAPER#n;2: INK#n;7
3560 ktrtal= pt-100: alt$=gem$: IF t: koordinatsystem: existdatategning
3570 END DEFine kontrollinie

```

- linie 3000 PROCEDURE "kontrollinie".
- linie 3010 gem\$ til "oplagring" af alt\$. alt\$ = 'w', linie 3150 er nødvendig: Se "inputtekst" (2500), oversigt over alt\$.
- linie 3030 antal betydende cifre. PROCEDURE "ciffertal" har cifre som parameter. Sikrer, at fx 12.1, .1234567 og 1234567 udskrives på en af brugeren valgt, rimelig måde, fx med 4 betydende cifre som 12.10, 0.1235 og 1235000.
- linie 3090 Hvis DATA eksisterer, kan brugeren vælge mellem en ny / en eksisterende "kontrol"-linie til illustration af fx kendte, tidligere forsøgsresultater eller til udtegnings af en muligvis bedre tilnærmelse end en ret linie, som vist i IFH-rapport 180. Vælges "ingen" linie vil en evt eksisterende linie ikke blive tegnet. Vælges en eksist. linie, tegnes polygon / kurve i overensstemmelse med de kendte DATA, uanset om brugeren sagde det modsatte. Vælges "ny" linie som polygon, fortsætter PROCEDUREN på samme måde som i "xyinput" (4000) og "enterdelaltesc" (5000). Polygonen udtegnes fx i linie 220, 1520, 8680. Vælges "ny" linie som kurve, bruges "kontrolkurve" og "tegnkurve" (21000,22000), som definerer og udtegner.
- linie 3490 ff Brugeren gøres opmærksom på, at det indlæste, eksist. datasæt omfatter flere punkter, der nu slettes. Det er, naturligvis, kun under den aktuelle udtegnings af disse DATA slettes, ikke på filen på disketten. Brugeren kan under en evt. senere DATA-lagring vælge at erstatte den gamle fil med den nye eller at lagre under et nyt navn.
- linie 3560 Smlgn. 3010 og 3410: Hvis der er foretaget revisioner, bør skærmtegningen rentegnes (t > 0).

Bemærk generelt,

at en polygon har 2 ... 8 punkter med x- og y-værdier, som andre punkter, dog med numrene 101 .. 100 + ktrtal, hvor ktrtal er antallet af punkter, max. 8, og at disse punkter har deres eget symbol, et lille kvadrat / o på hhv skærm / plotter.

at en kurve benytter numrene 101 .. 105 til sine DATA, se linie 21000 ff.

```

4000 DEFine PROCedure xyinput
4010 LOCAl pox(3), a$, pscr$, koordinat
4020 REPeat koordinat
4030 PAPER#4;2: INK#4;7: OVER -1: DIM pox(3)
4040 a$= FILL$(' ',23): FOR n= 6,9: AT#4; n,1: PRINT#4; a$
4050 :
4060 pt= pt + 1: IF pt= slutnr + 1: alt$= 's': EXIT koordinat
4070 pscr$= pt: IF pt < 10: pscr$= '0' & pscr$
4080 STRIP#4;7:INK#4;2: AT#4;2,11:PRINT#4;! pscr$ !!
4090 AT#4;9,1: PRINT#4;' ': AT#4;6, 1:PRINT#4;! x$(pt);', ';y$(pt);' '
4100 IF LEN(p$(pt)) AND p$(pt) INSTR 'ox' THEN
4110 OVER -1: INK 5: CURSOR x$(pt) / xs, y$(pt), -3, -6: PRINT p$(pt)
4120 pox(1)= x$(pt): pox(2)= y$(pt): pox(3)= CODE(p$(pt))
4130 END IF
4140 IF p$(pt)= ' ': p$(pt)= ox$
4150 STRIP#4;0: INK#4;7: AT#4;11,21: PRINT#4;! p$(pt);' '

4170 STRIP#4;7: INK#4;2: AT#4;8,1: PRINT#4;' NY VÆRDI AF X = '
4180 talput 4,9,2,g$(1),g$(2),1,(alt$ INSTR 'ael' > 0)
4190 IF ttal<g$(1) AND (x$(pt)-g$(1))*(g$(2)-x$(pt))>=0:tal$=x$(pt)
4200 IF ttal > g$(2) AND alt$ INSTR 'ael' THEN
4210 OVER 1: INK 7: CURSOR x$(pt) / xs, y$(pt),-3,-6: PRINT p$(pt)
4220 pt= pt-1: alt$= 's': EXIT koordinat: REMark esc
4230 END IF
4240 x$(pt)= tal$: AT#4; 9,2: PRINT#4; x$(pt);', '
4250 AT#4; 8,1: PRINT#4;' OG NY VÆRDI AF Y = '
4260 talput 4,9,4 + LEN(x$(pt)),g$(4),g$(5),1,0
4270 IF ttal<g$(4) AND (y$(pt)-g$(4))*(g$(5)-y$(pt))>=0: tal$=y$(pt)
4280 y$(pt)= tal$: AT#4; 9,4 + LEN(x$(pt)): PRINT#4; y$(pt);' '
4290 :
4300 IF pox(3) THEN
4310 INK 2: CURSOR pox(1) / xs, pox(2), -3, -6: PRINT CHR$(pox(3))
4320 END IF
4330 INK 4: CURSOR x$(pt) / xs, y$(pt), -3, -6: PRINT p$(pt)
4340 :
4350 enterdelaltesc: IF alt$== 'r' OR alt$== 's': EXIT koordinat
4360 END REPeat koordinat
4370 pkt$= p$(startnr)
4380 FOR p= st$ + 1 TO sl$: IF pkt$ <> p$(p): pkt$='ox': EXIT p
4400 CLS#0: PRINT#0;\ ' RENTEGNING J / N': svarput 0,1,20,'jn',0,2,7
4410 IF q$== 'j': koordinatsystem: existdatategning
4420 END DEFine xyinput

4500 REMark existdatategning: p$(pt) OVER 1, INK 7 >>> hvid
4510 REMark xyinput, før talput (4110) OVER -1, INK 5 >>> rød
4520 REMark gl, efter talput (4300) OVER -1, INK 2 >>> sort
4530 REMark ny do do (4320) OVER -1, INK 4 >>> grøn
4540 REMark SHIFT udskift (5110) OVER -1, INK 4 >>> sort
4550 REMark SHIFT nyt tegn (5160) OVER -1, INK 4 >>> grøn
4560 REMark ALT revision (5190) OVER 1, INK 7 >>> hvid
4570 REMark ENTER / ESC (5260/5300) OVER 1, INK 7 >>> hvid
4580 REMark DEL (5340) OVER -1, INK 2 >>> hvid
4590 REMark SOM STRAKS ANDRES (4110) OVER -1, INK 2 >>> rød

```

- linie 4000 PROCEDURE "xyinput"
- linie 4010 pox(3) bruges til x, y, CODE(SYMBOL) under indtastning, således at et udtegnet punkt fra et eksisterende DATA-sæt (4110) kan fjernes (4300) og erstattes med det nye, der måske har fået nye DATA (4320).
- SE OVERSIGTEN OVER PUNKT-UDTEGNING, linie 4500 .. 4590.
Bemærk brugen af OVER -1.
- a\$ er en simpel slette-streng (4040,4090,4180,4290), som fjerner evt. tidligere udskrift i REPEAT koordinat.
- pscr\$ sørger for, at numre bliver tocifrede, fx 03.
- linie 4110 CURSOR x / xs, y, -3, -6 placerer symboler korrekt, jfr. IFH-rapport 177. Skala, xs, se linie 1490 f.
- linie 4140 Reminiscenser fra tidligere (til kommende ?) udgaver, hvor der kan forekomme fejl-definitioner.
- linie 4180 og 4200 ff Se "talput" (15400). Næstsidste parameter er 1, dvs at < ENTER > vil acceptere eksisterende værdier, herunder 0 hvis der ikke er indlæst en eksisterende fil. Den sidste parameter er 1, hvis der er indlæst eksisterende DATA, alt\$ INSTR "ael", således at < ESC > er tilladt. Her er det muligheden for at indtaste "esc" i st. for et tal, der sættes i kraft, smlg. kommentarerne til linie 2500.
- linie 4350 "enterdelaltesc", se linie 5000 og linie 2500 ff.
- linie 4380 Efter en evt. symbol-revision må "pkt\$" beregnes påny.
- linie 4400 Brugeren kan ønske rentegning, hvis revision har skadet en akse under flytningerne etc.

Bemærk linie 4500 til 4590 om udtegnings af punktet.

Brugeren vil opfatte farverne - efter nogen tid måske ? Eksisterende punkter udtegnes hvide. Det aktuelle INPUT-punkt vil vise sig rødt og skifte farve, evt. også position, grønt efter INPUT. Det bliver hvidt efter accept.

```

5000 DEFine PROCedure enterdelaltesc
5010  LOCal b$, xyslut
5020  OVER#8; -1: INK#8;5
5030  REPeat xyslut
5040  AT#8;0,0: PRINT#8; '<SHIFT>' \ \ '<ENTER>' \ '< DEL >'
5050  IF alt$== 'l': AT#8;4,0: PRINT#8; '< ALT >'
5060  IF alt$ <>'i': AT#8;5,0: PRINT#8; '< ESC >'

5080  b$= INKEY$(5)
5090  IF KEYROW(7) THEN
5100    IF KEYROW(7)= 1 THEN
5110      OVER -1: INK 4: CURSOR x$(pt) / xs, y$(pt),-3,-6: PRINT p$(pt)
5120      z$= p$(pt): STRIP#4;0: INK#4;7: REMark SHIFT
5130      IF z$= 'o': ox$= 'x': p$(pt)= 'x'
5140      IF z$= 'x': ox$= 'o': p$(pt)= 'o'
5150      AT#4;11,21: PRINT#4; ! ox$ !!: PAPER#4;2: INK#4;7
5160      OVER -1: INK 4: CURSOR x$(pt) / xs, y$(pt),-3,-6: PRINT p$(pt)
5170    END IF
5180    IF KEYROW(7)= 4 AND alt$= 'l' THEN
5190      OVER 1: INK 7: CURSOR x$(pt) / xs, y$(pt),-3,-6: PRINT p$(pt)
5200      alt$= 'r': EXIT xyslut: REMark ALT
5210    END IF
5220  END IF

5240  IF LEN(b$) THEN
5250    IF CODE(b$)= 10 THEN
5260      OVER 1: INK 7: CURSOR x$(pt) / xs, y$(pt),-3,-6: PRINT p$(pt)
5270      EXIT xyslut: REMark ENTER
5280    END IF
5290    IF CODE(b$)= 27 AND alt$ <>'i' THEN
5300      OVER 1: INK 7: CURSOR x$(pt) / xs, y$(pt),-3,-6: PRINT p$(pt)
5310      alt$= 's': EXIT xyslut: REMark ESC
5320    END IF
5330    IF CODE(b$)= 194 THEN
5340      OVER -1: INK 2: CURSOR x$(pt) / xs, y$(pt),-3,-6: PRINT p$(pt)
5350      pt= pt - 1: EXIT xyslut: REMark DEL
5360    END IF
5370  END IF
5380  END REPeat xyslut

5400  OVER#8;0: PAPER#8;2: INK#8;7
5410  AT#8;0,0: PRINT#8; '<SHIFT>' \ \ '<ENTER>' \ '< DEL >'
5420  IF alt$== 'l': AT#8;4,0: PRINT#8; '< ALT >'
5430  IF alt$ <>'i': AT#8;5,0: PRINT#8; '< ESC >'
5440  END DEFine enterdelaltesc

```

linie 5000 PROCEDURE "enterdelaltesc"
 Kommentarerne til linie 2500 ff gentages neden for på denne side.

linie 5080 INKEY\$(5) giver en passende forsinkelse til en rimelig blink-frekvens for < ENTER > < DEL > osv.

linie 5090 Aflæser tastaturet, jfr. brugervejledningen.

linie 5100 KEYROW(7) = 1 er < SHIFT >.

linie 5130 f Symbol-ombytning.

linie 5180 KEYROW(7) = 4 er < ALT >, tilladt for alt\$= "1".

linie 5240 ff CODE(b\$) = 10, 27, 194 er hhv < ENTER >, < ESC > og < DEL >.

nyt INPUT	eksist. DATA / nyt layout / revision af alle punkter	revision af enkeltpunkt	kontrol- linie
SHIFT	SHIFT	SHIFT	---
ENTER	ENTER	ENTER	ENTER
DEL	DEL	DEL	DEL
---	ALT	---	---
---	ESC	ESC	ESC
alt\$= "i"	"1" "a"	"e"	"w"
340,	470,610,940,2560 .. 2630,	2560,2760,6070,	2600,
610,	2740 .. 2760,4180 ..4220,	6090,7070,7080,	3010,
2560,	4350,5050,5060,5180,5200,	7160,20300	3150,
2740	5290,5310,5420,5430,6070,		3560
	6080,20330		

linie 5400 Påny OVER 0 i kanal 8. Specialkommandoer i en PROCEDURE bør altid ophæves igen (5020/5400).

```

6000 DEFINE PROCEDURE dataændring
6010 LOCAL ændrloop
6020 REPEAT ændrloop
6030   CLS#0: PRINT#0;' DATA-LAGRING          d'
6040   PRINT#0;' REVISION          - SLUT  s' \ ' - ENKELTE PUNKTER  e'
6050   PRINT#0;' - ALLE I NR. FØLGE  a' \ ' FJERN PUNKT          f'
6060   svarput 0,4,24,'dseaf',0,2,7: alt$= q$
6070   IF alt$ INSTR 'ae': CLS#0: inputtekst
6080   IF alt$== 'a': pt= startnr - 1: xyinput
6090   IF alt$== 'e': enkeltrevison: CLS#4
6100   IF alt$== 'd': saveproces
6110   IF alt$== 'f': fjernpunkt: CLS#4
6120   IF alt$== 's': alt$= 'f': CLS#0: EXIT ændrloop
6130 END REPEAT ændrloop
6140 pkt$= p$(startnr)
6150 FOR p= st$ + 1 TO sl$: IF pkt$ <> p$(p): pkt$='ox': EXIT p
6160 END DEFINE dataændring

7000 DEFINE PROCEDURE enkeltrevison
7010 LOCAL nrloop: CLS#0: PAPER#0;7: INK#0;2
7020 CURSOR#0;0,100,7, 7: PRINT#0;'INDTAST ØNSKET PKT.NR. '
7030 CURSOR#0;0,100,7,17: PRINT#0;'          PLUS          '
7040 CURSOR#0;0,100,7,27: PRINT#0;'AFSLUT DOG MED          '
7050 CURSOR#0;0,100,7,37: PRINT#0;'ELLER PKT.NR.=          '
7060 PAPER#0;2: INK#0;7: CURSOR#0;0,100,103,37: PRINT#0;' esc '
7070 CURSOR#0;0,100,97,17: PRINT#0;'<ENTER>'
7080 CURSOR#0;0,100,97,27: PRINT#0;'< ESC >'
7090 BLOCK#0; 138,1,7,47,7
7100 AT#4;19,1: PRINT#4;'ØNSKET PKT.NR.'
7110 REPEAT nrloop
7120   AT#4;19,21: PRINT#4;'          '
7130   talput 4,19,21,startnr, slutnr,1,1
7140   IF ttal > slutnr: EXIT nrloop
7150   pt= ttal - 1: xyinput
7160   IF alt$== 's': EXIT nrloop
7170 END REPEAT nrloop
7180 END DEFINE enkeltrevison

7500 DEFINE PROCEDURE fjernpunkt
7510 LOCAL fjnr: CLS#0
7520 PRINT#0;' DATASÆTTET TRUNKERES,' \ ' SÅ DET HAR FORTLØBENDE'
7530 PRINT#0;' NUMRE FRA STARTNR.'; startnr;'.'
7540 PRINT#0;' ANGIV NR.PÅ DET PUNKT,' \ ' SOM SKAL FJERNES:'
7550 talput 0,4,19,startnr,slutnr,0,0: fjnr= ttal: OVER 1: INK 0
7560 CURSOR x$(fjnr) / xs, y$(fjnr), -3, -6: PRINT p$(fjnr): INK 7
7570 IF fjnr < slutnr THEN
7580   FOR n= fjnr TO sl$-1: p$(n)=p$(n+1): x$(n)=x$(n+1): y$(n)=y$(n+1)
7590 END IF
7600 p$(slutnr)= '': x$(slutnr)= 0: y$(slutnr)= 0: slutnr= slutnr-1
7610 sl$= FILL$('0',slutnr < 10) & slutnr
7630 CLS#0: PRINT#0;\ ' RENTEGNING J / N': svarput 0,1,20,'jn',0,2,7
7640 IF q$== 'j': koordinatsystem: existdatategning
7650 END DEFINE fjernpunkt

```

```

linie 6000      PROCedure "dataændring".

linie 6070 ff   alt$ = "a" eller "e", revision af alle i nummerfølge /
                et enkelt punkt ( og evt. de følgende ) benytter samme
                INPUT-princip som for indtastning af nye DATA, nemlig
                PROCedurerne "inputtekst" og "xyinput" (2500 og 4000),
                hvor "enkeltrevision", se nedenfor, fastsætter en bru-
                gervalgt værdi af første nummer ( pt= ttal - 1, 7140 )

linie 6100      "saveproces", linie 18800 ff.

linie 6110      "fjernpunkt", se nedenfor.

linie 6120      alt$= "s" bliver til "f", se ÆNDRINGS-PROCEDURER.
                Se hovedloop hvor linierne 760 >>> 770, 940 >>> 660.

linie 6140      Smlgn. kommentarer til linie 4380, da der kan være ud-
                ført revisioner i denne PROCEDURE.
                Der fortsættes til "datasætvalg" (900).

linie 7000      PROCedure "enkeltrevision" er kommenteret ovenfor, se
                ÆNDRINGS-PROCEDURER. Bemærk den "følgende" linie 6140.


linie 7500      PROCedure "fjernpunkt", se ÆNDRINGS-PROCEDURER.

linie 7520      DATA-sættet trunkeres, så der stadig er et fortløbende
                sæt numre fra start-nummer til slut-nummer.
                Det er klogt umiddelbart herefter, i "datasætvalg", at
                bede om printerudskrift af INPUT-DATA, linie 900 ff.
                Bemærk linie 6140, som passerer umiddelbart efter, at
                revisionen er afsluttet.

```

```

8000 DEFine PROCedure existdatategning
8010 IF slutnr= startnr: RETURN
8020 OVER 1: INK 7
8030 FOR p= st$ TO sl$
8040 IF p$(p) INSTR pkt$: CURSOR x$(p)/xs, y$(p), -3, -6: PRINT p$(p)
8050 END FOR p
8060 END DEFine existdatategning

8400 DEFine PROCedure koordinatsystem
8410 LOCAL xkoor, ykoor
8420 OVER 1: INK 7: BORDER 1,2: PAPER 0: CLS
8430 LINE g$(1)/xs, g$(4)/1 TO g$(1)/xs, g$(5)/1 TO g$(2)/xs, g$(5)/1
8440 LINE TO g$(2)/xs, g$(4)/1 TO g$(1)/xs, g$(4)/1
8450 IF g$(1) * g$(2) < 0: LINE 0, g$(4)/1 TO 0, g$(5)/1
8460 IF g$(4) * g$(5) < 0: LINE g$(1)/xs, 0 TO g$(2)/xs, 0
8470 REMark LINE men ikke CURSOR vil ikke æde en streng, derfor g$/1
8480 FOR n= 0 TO yantal
8490 ykoor= g$(4) + g$(6) * n: tal$= ykoor
8500 IF ABS(ykoor) < 1: ciffertal ykoor, 1
8510 IF ABS(g$(6)) < 1: IF ykoor= ykoor DIV 1: tal$= tal$ & '.0'
8520 POINT g$(1) / xs, ykoor: LINE_R TO yskala / 50, 0
8530 POINT g$(2) / xs, ykoor: LINE_R TO -yskala / 50, 0
8540 CURSOR g$(1) / xs, ykoor, -4 -6 * LEN(tal$), -8: PRINT tal$
8550 END FOR n
8560 FOR n= 0 TO xantal
8570 xkoor= g$(1) + g$(3) * n: tal$= xkoor
8580 IF ABS(xkoor) < 1: ciffertal xkoor, 1
8590 IF ABS(g$(3)) < 1: IF xkoor= xkoor DIV 1: tal$= tal$ & '.0'
8600 POINT xkoor / xs, g$(4)/1: LINE_R TO 0, yskala / 50
8610 POINT xkoor / xs, g$(5)/1: LINE_R TO 0, -yskala / 50
8620 CURSOR xkoor / xs, g$(4), -3 * LEN(tal$), 4: PRINT tal$
8630 END FOR n
8640 CURSOR g$(1) / xs, g$(5), 0, -15: PRINT yes$
8650 CURSOR g$(1) / xs, g$(4), 0, 15: INK 4: PRINT dat$
8660 CURSOR g$(2) / xs, g$(4), -6*LEN(xe$), 15: INK 7: PRINT xe$
8670 CURSOR g$(2)/xs, g$(5), -6*LEN(fil$), -15: INK 4: PRINT fil$: INK 7
8680 IF ktr$== 'p': tegnpolygon
8690 IF ktr$== 'k': tegnkurve
8700 END DEFine koordinatsystem

9000 DEFine PROCedure tegnpolygon
9010 LOCAL xx, yy, dx, dy, antal%: INK 2
9020 FOR n= 101 TO 100 + ktrtal
9030 CURSOR x$(n) / xs, y$(n), -3, -3: PRINT 'o'
9040 IF n= 101: NEXT n
9050 xx= (x$(n) - x$(n-1)) ^2 / (xs^2): yy= (y$(n) - y$(n-1)) ^2
9060 antal%= 50 * SQRT (xx + yy) / yskala
9070 dx= (x$(n)-x$(n-1)) / antal%: dy= (y$(n)-y$(n-1)) / antal%
9080 FOR p= 1 TO antal% - 1
9090 POINT (x$(n-1) + dx * p) / xs, y$(n-1) + dy * p
9100 END FOR p
9110 END FOR n: INK 7
9120 END DEFine tegnpolygon

```

linie 8000 PROCEDURE "existdatategning".

linie 8010 f tegne\$ er fjernet i programmets nyere udgaver.
 Hensigten var at sørge for, at der kun udtegnedes det symbol, som der var bedt om, på en udjævningstegning. Da PROCEDUREN benyttes på mange måder, ville det blive nødvendigt med mange omstillinger.
 I stedet omstilles pkt\$ midlertidigt, i linie 13010 ff, til skærmudtegning. Til plotterudtegning bruges formuleringen i linie 28530 til samme formål.

Bemærk, at hæld\$ og rg\$ og valg\$ alle sættes lig "" i programmets start, linie 350 ff. De antager værdier i PROCEDUREN "regrestype" (10030, 10190 ff), der kaldes fra hovedloop, linie 690 f. De "nulstilles" påny efter fejludjævningen, i linie 720.

linie 8430 LINE x,y TO x,y TO x,y
 linie 8440 LINE TO x,y TO x,y

linie 8450 Tegnefeltets begrænsningslinier har korte streger for de valgte akseinddelinger. Kun hvis x = 0 eller y = 0 optræder inden for tegnefeltet, trækkes der streger i selve feltet.

linie 8470 CURSOR x\$, y\$ er OK.
 Men LINE x\$ / 1, y\$ / 1 er, af uransagelige årsager, nødvendig !

linie 8520 POINT for at styre LINE_R, relative koordinater for akseinddelingsmærker.

linie 8540 CURSOR for de tilhørende tal, der udskrives pænt med "cifertal" (15200), suppleret med 8500, 8510.
 "Små" inddelinger bliver fx 0.6 0.8 1.0 etc.

linie 9000 PROCEDURE "tegnpolygon".
 Disseker selv den simple geometri, der bare giver en langsom udtegning på grund af regnearbejdet. Jeg har ved forsøg fundet, at der opnås en rimeligt ensartet afstand mellem prikkerne i linien.
 xs og yskala: Se linie 1490 ff. Tænkepause.

```

10000 DEFine PROCedure regrestype
10010 LOCAl typevalg, vlg, r, s: r= 0: s= 50
10020 REPeat typevalg
10030   IF LEN(pkt$) < 2: rg$= pkt$: valg$= pkt$: EXIT typevalg

10050   FOR n= 0,4: PAPER#n;2: INK#n;7: CLS#n
10060   STRIP#4;7: INK#4;2: AT#4;1,1: PRINT#4;' SKÆRMBILLEDE MED '
10070   PAPER#4;2: INK#4;7: BLOCK#4;96,1,12,19,2: BLOCK#4;108,1,6,20,7
10080   PRINT#4;\ ' LINIE FOR FEJLUDJÆVNING' \ ' AF o   KUN o VISES'
10090   PRINT#4; TO 7;'x VISES OGSÅ' \
10100   PRINT#4;\ ' LINIE FOR FEJLUDJÆVNING' \ ' AF x   KUN x VISES'
10110   PRINT#4; TO 7;'o VISES OGSÅ' \
10120   PRINT#4;\ ' EEN LINIE FOR UDJÆVNING' \ ' AF o + x UNDER EET' \
10130   PRINT#4;\ ' LINIERNE FOR o OG FOR x' \ '   PÅ SAMME SKÆRM' \
10140   AT#4;17,1: PRINT#4;' INDFAST VALG: ': STRIP#4;7: INK#4;2
10150   AT#4;4,20: PRINT#4;'   o ': AT#4; 5,20: PRINT#4;' o+ '
10160   AT#4;8,20: PRINT#4;'   x ': AT#4; 9,20: PRINT#4;' x+ '
10170   AT#4;12,20: PRINT#4;' ox ': AT#4;15,19: PRINT#4;' o+x '
10180   REPeat vlg
10190   AT#4;17,19: PRINT#4;'           ': AT#4;17,20: INPUT#4; valg$
10200   IF LEN(pkt$)= 1 OR valg$== 'ox' THEN
10210     IF valg$ INSTR pkt$: rg$= valg$: EXIT vlg: ELSE NEXT vlg
10220   END IF
10230   IF LEN(valg$) AND (valg$ INSTR 'x+' OR valg$ INSTR 'o+x') THEN
10240     FOR n= st$ TO sl$
10250       IF p$(n)=='o': r=r+1: rgrnr(r)= n: ELSE : s=s+1: rgrnr(s)= n
10260     END FOR n : rgoantal= r: rgxantal= s - 50
10270     IF valg$(1) <> '+': rg$= valg$(1): EXIT vlg
10280   END IF
10290   END REPeat vlg: PAPER#4;2: INK#4;7

10310   PAPER#0;7: INK#0;2: PRINT#0;\ ' KORREKT VALG : ';
10320   PRINT#0; valg$; '           ': svarput 0,1,21,'jn',3,2,7
10330   IF q$== 'j': EXIT typevalg
10340   END REPeat typevalg

10360   PRINT#0; \ ' REGRESSION,LINÆR r' \ ' LODRET LINIE      1'
10370   PRINT#0; ' VANDRET LINIE      v' \ ' HÆLDNING FASTLAGT h'
10380   PRINT#0; ' PUNKT FORUDBESTEMT p'
10390   svarput 0,4,22,'rlvhp',0,2,7: hæld$= q$
10400   IF hæld$== 'p' THEN
10410     CLS#0: AT#0;1,1: PRINT#0;'PUNKTETS KOORDINATER' \ ' X, Y ='
10420     talput 0,2,8,-32000,32000,0,0: xfast= ttal
10430     cifertal xfast, cifre: AT#0;2,8: PRINT#0; tal$ ;','
10440     talput 0,2,10 + LEN( tal$ ),-32000,32000,0,0: yfast= ttal
10450   END IF
10460   IF hæld$== 'h' THEN
10470     CLS#0: AT#0;1,1: PRINT#0;'HÆLDNINGSKOEFFICIENT' \ ' Y / X ='
10480     talput 0,2,9,-32000,32000,0,0: xhæld= ttal
10490   END IF

10510   fejludjævning
10520   END DEFine regrestype

```

- linie 10000 PROCEDURE "regrestype"
Her fastlægger brugeren de for fejludjævningen krævede
variabler hæld\$, rg\$, valg\$ m.fl.
- Bemærk, at hæld\$ og rg\$ og valg\$ alle sættes lig "" i
programmets start, linie 350 ff. De antager værdier i
PROCEDUREN "regrestype" (10030, 10190 ff), der kaldes
fra hovedloop, linie 690 f. De "nulstilles" påny efter
fejludjævningen, i linie 720.
- For valg\$= "o+x" :
Se bl.a. linierne 700 og 10270 samt pag. 15.
- linie 10200 Enten pkt\$= "o" / "x", dvs kun eet symbol for punkter,
eller valg\$= "ox" , dvs udjævning for begge "slags"
punkter under et.
- linie 10210 valg\$ INSTR pkt\$ kan kun opfyldes for:
valg\$= "o" og pkt\$= "o"
valg\$= "x" og pkt\$= "x"
valg\$= "ox" og pkt\$= "ox", alt andet afvises.
- linie 10230 valg\$ er ikke en "nulstreng".
pkt\$ må indeholde to karakterer, jfr. linie 10200.
valg\$ må være "x" "x+" "o" "o+" "o+x" "x+" "+".
- linie 10270 valg\$= "+" og "+x" afvises.
rg\$ fastlægges.
Bemærk at for "o+x" startes der med rg\$= "o".
- linie 10240 ff rgoantal og rgxantal optælles.
- linie 10250 rgrnr(100) er nulstillet i linie 670.
Er symbolet "o", bliver rgrnr(01..49) = punktnummeret.
Er symbolet "x", bliver rgrnr(51..99) = punktnummeret.
Se linie 19780 i PROCEDUREN "sortertal".
- linie 10260 rgoantal = antal punkter med symbol "o".
rgrnr(01....rgoantal) indeholder punktnumrene.
rgxantal = antal punkter med symbol "x".
rgrnr(51...50 + rgxantal) indeholder punktnumrene.
- linie 10310 ff Fortryder du ?
- linie 10360 Valg af fejludjævnings-metode.
- linie 10400 Fast punkts koordinater.
- linie 10460 Hældningskoefficient.
- linie 10510 PROCEDURE "fejludjævning" (11000).

```

11000 DEFine PROCedure fejludjævning
11010   LOCal k, sumx, sumy, sakx, saky, sapxy, sakxf, sakyf, sapxyf
11020   LOCal gn timer, gny, spredx, spredy, korrel

11040   korrel=0
11050   sumx=0: sumy=0: sakx=0: saky=0: sapxy=0: sakxf=0: sakyf=0
11060   sapxyf=0: rgtal= sl$ - st$ + 1: k= 0: IF rg$=='x': k= 20
11070   IF pkt$== 'ox' AND rg$== 'o': rgtal= rgoantal
11080   IF pkt$== 'ox' AND rg$== 'x': rgtal= rgxantal

11100   FOR n= 0,4: PAPER#n;2: INK#n;7: CLS#n
11110   udj$(1+k)= 'SERIEN ' & st$ & '-' & sl$ & ' HAR ' & rgtal
11120   udj$(2+k)= 'PUNKTER MED SYMBOL ' & rg$
11130   FOR n= 1 TO 2: AT#4; n, 1: PRINT#4; udj$(n+k)
11140   IF rgtal < sl$ - st$ + 1 THEN
11150     PRINT#4; \ ' NUMRENE ER: ': sortertal 4: udj$(20+k)= 'ja'
11160     INPUT#0; \ ' NUMMER-UDSKRIFT ? ': svarput 0,1,20,'jn',0,2,7
11170     IF q$== 'j': sortertal 3: BPUT#3;12: CLOSE#3
11180   END IF

11200   AT#4;3,0: CLS#4,2: IF hæld$ INSTR 'ph': udj$(4+k)= 'RET LINIE'
11210   IF hæld$== 'r': udj$(4+k)= 'LINEÆR REGRESSION.'
11220   IF hæld$== 'l': udj$(4+k)= 'LODRET LINIE'
11230   IF hæld$== 'v': udj$(4+k)= 'VANDRET LINIE'
11240   IF hæld$== 'h': udj$(5+k)= 'MED GIVEN HÆLDNING.'
11250   IF hæld$ INSTR'vlp': udj$(5+k)= 'GENNEM PUNKT X,Y ='

11270   FOR n= st$ TO sl$
11280     IF p$(n) INSTR rg$: sumx= sumx + x$(n): sumy= sumy + y$(n)
11290   END FOR n
11300   gn timer= sumx /rgtal: gny= sumy /rgtal: fjl(1+k)= gn timer: fjl(2+k)= gny
11310   IF hæld$ INSTR'rh': udj$(7+k)= 'TYNGDEPUNKT X,Y ='
11320   IF hæld$== 'v': cifertal 0, cifre: udj$(6+k)= tal$ & ' , '
11330   IF hæld$ INSTR'rh': cifertal gn timer,cifre: udj$(8+k)= tal$ & ' , '
11340   IF hæld$== 'l': cifertal gn timer, cifre: udj$(6+k)= tal$ & ' , '
11350   IF hæld$== 'l': cifertal 0, cifre: udj$(6+k)= udj$(6+k) & tal$
11360   IF hæld$ INSTR'rhpv' THEN
11370     cifertal gny, cifre: n = 0: IF hæld$== 'v': n= -2
11380     udj$(8+k+n)= udj$(8+k+n) & tal$
11390   END IF

11410   FOR n= st$ TO sl$
11420     IF p$(n) INSTR rg$ THEN
11430       sakx= sakx + (x$(n) -gn timer)^2: saky= saky + (y$(n) -gny)^2
11440       sapxy= sapxy + (x$(n) -gn timer) * (y$(n) -gny)
11450     END IF
11460   END FOR n: spredx= SQRt(sakx/(rgtal-1)): fjl(3+k)= spredx
11470     spredy= SQRt(saky/(rgtal-1)): fjl(4+k)= spredy

11490   IF hæld$ INSTR 'rh' THEN
11500     IF ABS(sapxy) * xs < sakx / 50: hæld$= 'v'
11510     IF ABS(sapxy) * xs > sakx * 50: hæld$= 'l'
11520     IF hæld$ INSTR 'vl': fejludjævning: udj$(7+k)='': RETURN
11530   END IF

11550   REMark UDSKRIFT FORTSÆTTER PÅ NÆSTE SIDE

```

linie 11000 PROCEDURE fejludjævning.

linie 11040 f LOCALE variabler nulstilles.

linie 11060 Det samlede antal punkter er rgtal.
k sættes til 0 eller 20.

Nedenfor gentages et uddrag af kommentarerne til linie 300 ff,
suppleret med et oversigtskema:

valg\$	styrer, hvilke punkter, der skal udtegnes / indgå i en fejludjævning, se nedenfor og linie 10030 ff..
rg\$	styrer, hvilke punkter, der fejludjævnes, se nedenfor og linie 10030 ff.
rgtal	antal punkter i fejludjævningen, se linie 11060 ff.
rgoantal	antal punkter med symbol 0, se linie 10260.
rgxantal	antal punkter med symbol x, se linie 10260.
hæld\$	angiver regrestypen, se nedenfor og linie 10360 ff.
udj\$(40,24)	k= 0 / 20. Udjævnings-udskrift på skærm, se linie 11700.
fjl(40)	k= 0 / 20. Udjævnings-DATA til SAVE og plotter, se linie 10260, 19270.
rgrnr(100)	k= 0 / 50. Numre til PROCEDURE "sortertal", linie 10250.

pkt\$	valg\$	rg\$	fejludjævning, k	sortertal, k
"o"	"o"	"o"	rgtal 0	0
"x"	"x"	"x"	rgtal 20	50
"ox"	"o"	"o"	rgoantal 0	0
"ox"	"x"	"x"	rgxantal 20	50
"ox"	"ox"	"ox"	rgtal 0	0
"ox"	"o+"	"o"	rgoantal 0	0
"ox"	"x+"	"x"	rgxantal 20	50
"ox"	"o+x"	"o" og "x"	rgoantal & rgxantal 0 & 20	0 & 50

SE OGSÅ AFSNITTET OM "TEORI", pag. 129.

Bemærk linie 11490 ff og 12010 ff: Linier, der på skærmen udtegnes med hældninger, der - på gr. af det begrænsede antal pixler - ser ud som om de er næsten lodrette / vandrette, henvises til analyse som lodrette / vandrette linier. Et nyt layout med en finere inddeling af x- eller y-aksen kan evt. vise, om der reelt eksisterer en hældning.

Der er ikke indført beregning af spredningen på liniens hældning, uanset at dette, for alm. lineær regression, er simpelt. Kritiske brugere kan udvide programmet med de mere detaljerede ønsker, de måtte have.

```

11700  REMark fejludjævning FORTSAT FRA FORRIGE SIDE

11720  IF hæld$== 'l' THEN
11730      ciffertal spredx, cifre-1+(cifre=1)
11740      udj$(8+k)= 'MIDDELFEJL X= ' & tal$
11750  END IF

11770  IF hæld$== 'v' THEN
11780      ciffertal spredy, cifre-1+(cifre=1)
11790      udj$(8+k)= 'MIDDELFEJL Y= ' & tal$
11800  END IF

11820  FOR n= 4 TO 8: AT#4; n, 1: PRINT#4; udj$(n+k)
11830  IF hæld$ INSTR'lv': udtegnning: RETURN

11850  IF hæld$== 'r' THEN
11860      xhæld= sapxy / sakx: ciffertal xhæld, cifre: fjl(5+k)= xhæld
11870      udj$(10+k)= 'HÆLDNING = ' & tal$ : akseparametre gn, gny
11880      korrel= sapxy^2 / sakx / saky
11890  END IF

11910  IF hæld$== 'p' THEN
11920      ciffertal xfast, cifre: udj$(6+k)= tal$ & ' , '
11930      ciffertal yfast, cifre: udj$(6+k)= udj$(6+k) & tal$
11940      AT#4;6,1: PRINT#4; udj$(6+k): fjl(8+k)= xfast: fjl(9+k)= yfast
11950      FOR n= st$ TO sl$
11960          IF p$(n) INSTR rg$ THEN
11970              sakxf= sakxf +(x$(n)-xfast)^2: sakyf= sakyf +(y$(n)-yfast)^2
11980              sapxyf= sapxyf + (x$(n)-xfast) * (y$(n)-yfast)
11990          END IF
12000      END FOR n
12010      IF ABS(sapxyf) * xs < sakxf / 50: hæld$= 'v'
12020      IF ABS(sapxyf) * xs > sakxf * 50: hæld$= 'l'
12030      IF hæld$ INSTR 'vl': fejludjævning: RETURN
12040      xhæld= sapxyf / sakxf: ciffertal xhæld, cifre: fjl(5+k)= xhæld
12050      udj$(10+k)= 'HÆLDNING= ' & tal$: akseparametre xfast,yfast
12060      korrel=( saky - sakyf + sapxyf * xhæld) / saky
12070  END IF

12090  IF hæld$== 'h' THEN
12100      ciffertal xhæld, cifre: fjl(5+k)= xhæld
12110      udj$(10+k)= 'HÆLDNING= ' & tal$: akseparametre gn, gny
12120      korrel= ( 2 * sapxy * xhæld - sakx * xhæld^2 ) / saky
12130  END IF

12150  IF korrel > 0: korrel= SQRT( korrel ) : ELSE korrel= 0
12160  ciffertal korrel, 2 + ( korrel > .98 ): fjl(7+k)= korrel
12170  udj$(18+k)= 'KORRELATIONSKOEFFICIENT' : udj$(19+k)= '=' & tal$

12190  FOR n= 10 TO 19: AT#4; n, 1: PRINT#4; udj$(n+k)
12200  udtegnning
12210  END DEFine fejludjævning

```

OVERSIGT OVER ujd\$(40,24)

linie 0 ... 19 for "o"			linie 20 ... 39 for "x".		
hæld\$=	"r"	"h"	"p"	"l"	"v"
0	-	-	-	-	-
1	 SERIEN st\$ - sl\$ HAR rgtal				
2	 PUNKTER MED SYMBOL o/x				
3	-	-	-	-	-
4	LINEÆR REGRESS.	RET LINIE	RET LINIE	LODRET LINIE	VANDRET LINIE
5	-	M.GIVEN HÆLDN.		GENNEM PUNKT X,Y=	
6	-	-	x, y fast	gnx, 0	0, gny
7	 TYNGDEPKT. X,Y =			-	-
8	 gnx, gny			MDLFJL= spredx	MDLFJL= spredy
9	-	-	-	-	-
10	 HÆLDNING =			-	-
	 xhæld				
11	-	-	-	-	-
12	 SKÆR. M. Y-AKSEN			-	-
13	 i 0, yakse			-	-
14	-	-	-	-	-
15	 SKÆR. M. X-AKSEN			-	-
16	 i xakse, 0			-	-
17	-	-	-	-	-
18	 KORREL. KOEFF.			-	-
19	 = korrel			-	-

```

12500 DEFine PROCedure akseparametre ( xparam, yparam )
12510   udj$(12+k)= 'SKÆRING MED Y-AKSEN I'
12520   yakse= yparam - xparam * xhæld
12530   ciffertal 0      , cifre: udj$(13+k)= tal$ & ' , '
12540   ciffertal yakse, cifre: udj$(13+k)= udj$(13+k) & tal$
12550   udj$(15+k)= 'SKÆRING MED X-AKSEN I'
12560   xakse= xparam - yparam / xhæld
12570   ciffertal xakse, cifre: udj$(16+k)= tal$ & ' , '
12580   ciffertal 0      , cifre: udj$(16+k)= udj$(16+k) & tal$
12590   fjl(14+k)= xakse: fjl(15+k)= yakse
12600   yvenst= yparam +(g$(1) - xparam) * xhæld: fjl(10+k)= yvenst
12610   yhjre= yparam +(g$(2) - xparam) * xhæld: fjl(11+k)= yhjre
12620   xnedad= xparam +(g$(4) - yparam) / xhæld: fjl(12+k)= xnedad
12630   xopad = xparam +(g$(5) - yparam) / xhæld: fjl(13+k)= xopad
12640 END DEFine akseparametre

13000 DEFine PROCedure udtegning
13010 IF pkt$== 'ox' AND LEN(valg$)= 1 THEN
13020   pkt$= rg$: BORDER 1,2: CLS: koordinatsystem: existdatategning
13030   pkt$= 'ox': REMark NY TEGNING TIL UDJÆVN.   pkt$ STILLES TILBAGE
13040 END IF
13050 INK 7: IF rg$== 'x': INK 4

13070 IF hæld$=='l': LINE gn timer / xs, g$(4)/1 TO gn timer / xs, g$(5)/1: INK 7: RETURN
13080 IF hæld$=='v': LINE g$(1)/xs , gny TO g$(2)/xs , gny: INK 7: RETURN

13100 IF (yvenst-g$(4)) * (g$(5)-yvenst) > 0 THEN
13110   IF (yhøjre-g$(4)) * (g$(5)-yhøjre) > 0 THEN
13120     LINE g$(1) / xs, yvenst TO g$(2) / xs, yhjre: INK 7: RETURN
13130   END IF
13140   IF xhæld > 0: LINE g$(1) / xs, yvenst TO xopad / xs, g$(5)/1
13150   IF xhæld < 0: LINE g$(1) / xs, yvenst TO xnedad / xs, g$(4)/1
13160   INK 7: RETURN
13170 END IF

13190 IF (yhøjre-g$(4)) * (g$(5)-yhøjre) > 0 THEN
13200   IF xhæld > 0: LINE xnedad / xs, g$(4)/1 TO g$(2) / xs, yhjre
13210   IF xhæld < 0: LINE xopad / xs, g$(5)/1 TO g$(2) / xs, yhjre
13220   INK 7: RETURN
13230 END IF

13250 LINE xnedad / xs, g$(4)/1 TO xopad / xs, g$(5)/1: INK 7
13260 END DEFine udtegning

```

linie 12500 PROCedure "akseparametre".
 og 13000 Simple geometriske beregninger, hvori der ud fra
 gnx og gny beregnes:
 yvenst, yhjre, xnedad, xopad hhv xakse, yakse,
 der er koordinater til udjævningsliniens skæring med
 figurens begrænsningslinier hhv x- og y-aksen.

linie 13010 Se kommentarer til linie 8010 ff.

linie 13100 Seks mulige linieforløb. Hvid / grøn for "o" / "x".

```

13300 DEFine PROCedure tegning_lagring
13310 LOCAl udskriftloop
13320 IF valgs$== 'o+x' AND rg$== 'o' THEN
13330   CLS#0: PRINT#0;\ ' UDJÆVNING FOR PUNKT  o'
13340   PRINT#0;' TAST < ENTER > FOR   x' \ ' ALT KAN LAGRES, TEGNES'
13350   PRINT#0;' OG UDSKRIVES SENERE': AT#0;4,22: INPUT#0; q$: RETurn
13360 END IF
13370 REPeat udskriftloop
13380 BLOCK#4; 154,2,0,200,7
13390 CLS#0: PRINT#0;' DATA-LAGRING           d'\ ' TEGNING PA PLOTTER  t'
13400 PRINT#0;' PRINTER-KOPI,-DATA  p' \ ' FEJLUDJÆVNING           f'
13410 PRINT#0;' REVISIONER           r'
13420 svarput 0,4,24,'dtpfr',0,2,7: alt$= q$
13430 IF alt$=='d': saveproces: IF hæld$ <> '': fejludjævning
13440 IF alt$=='t': plotteropen: IF hæld$ <> '': fejludjævning
13450 IF alt$=='p': printerudskrift: REMark SE DENNE
13460 IF alt$ INSTR 'fr': EXIT udskriftloop
13470 END REPeat udskriftloop: REMark RETUR TIL LINIE 600 ff
13480 END DEFine tegning_lagring

```

```

13600 DEFine PROCedure printerudskrift
13610 CLS#0: PRINT#0;\ ' PRINTER TÆNDT J / N': svarput 0,1,22,'j',0,2,7
13620 REPeat printerloop
13630 CLS#0: PRINT#0
13640 PRINT#0;' SKÆRMBILLEDE: KOPI  s' \ ' INPUTDATA-UDSKRIFT  i'
13650 PRINT#0;' UDJÆVNING-UDSKRIFT  u' \ ' RETUR TIL PROGRAM    r'
13660 svarput 0,4,23,'siur',0,2,7
13670 IF q$=='r': EXIT printerloop
13680 IF q$=='i': inputudskrift
13690 IF q$=='u' AND hæld$ <> '': udjævnudskrift
13700 IF q$=='s' THEN
13710   OPEN#3,ser1: BAUD baudtal: TRA trat: BPUT#3;27,108,10: CLOSE#3
13720   SCOPY#1: TRA trap
13730   IF hæld$ <> ' ' THEN
13740     udjævnudskrift: OPEN#3,ser1: FOR n= 1 TO 47: BPUT#3;10
13750     BPUT#3;27,108,10,27,81,77,27,67,72,27,78,10: CLOSE#3
13760     ELSE : OPEN#3,ser1: BPUT#3;12: CLOSE#3
13770   END IF
13780 END IF
13790 END REPeat printerloop
13800 END DEFine printerudskrift

```

linie 13300 PROCedure "tegning_lagring".

linie 13320 ff Udskrift for "o+x" sker samlet, til sidst.

linie 13600 PROCedure "printerudskrift":

linie 13610 Gavnlig, da en slukket printer standser programmet.

linie 13680 f "inputudskrift" (14500). "udjævnudskrift" (14000).

linie 13720 "SCOPY" dækker scopy_bin og qcopy_bin: TRA 1.

linie 13730 ff "udjævnudskrift", hvis udjævning er foretaget.
47 x BPUT#3;10 (= LF, LINEFEED) passer netop.

```

14000 DEFine PROCedure udjævnudskrift
14010 LOCal k, lp: IF valg$== 'o+x' : rg$= 'x': rgtal= rgxantal
14020 k= 0: IF rg$== 'x': k= 20
14040 OPEN#3,ser1: BAUD baudtal: TRA trap
14050 BPUT#3,27,108,10,27,81,77,27,67,72,27,78,10
14060 IF hæld$ INSTR'lv': PRINT#3;udj$(4+k)&' '&udj$(5+k)&' '&udj$(6+k)
14070 IF hæld$ == 'r' : PRINT#3; udj$(4+k)
14080 IF hæld$ INSTR'hp': PRINT#3; udj$(4+k) &' '& udj$(5+k)
14090 IF hæld$ == 'p' : PRINT#3; udj$(6+k)
14100 IF hæld$ == 'h' : PRINT#3; udj$(10+k)
14110 CLOSE#3
14120 sortertal 3

14140 REPEAT lp
14150 IF hæld$ INSTR 'lv': PRINT#3; udj$(8+k)
14170 IF hæld$ INSTR 'rhp' THEN
14180 PRINT#3; udj$(7+k) &' ' & udj$(8+k)
14190 PRINT#3; \ ; udj$(10+k)
14200 PRINT#3; \ ; udj$(12+k) &' ' & udj$(13+k)
14210 PRINT#3; \ ; udj$(15+k) &' ' & udj$(16+k)
14220 PRINT#3; \ ; udj$(18+k) &' ' & udj$(19+k)
14230 END IF
14250 IF LEN(valg$) < 3 OR rg$== 'o': EXIT lp
14260 :
14270 rg$= 'o': k= 0: rgtal= rgoantal
14280 PRINT#3: sortertal 30
14290 END REPEAT lp
14300 BPUT#3;12: CLOSE#3
14310 END DEFine udjævnudskrift

14500 DEFine PROCedure inputudskrift
14510 LOCal n, n$, ktrp: ktrp= 186 - 49 * (pr$=='c')
14520 OPEN#3,ser1: BAUD baudtal: TRA trap
14530 BPUT#3,27,108,10,27,81,77,27,67,72,27,78,10
14540 PRINT#3;'FILENS NAVN : ' ; filnavn$ ; \ 'EDB-DATO : ' ; dato$
14550 PRINT#3;'FIGURNAVN : ' ; fil$ ; \ 'FIGUR-DATO : ' ; dat$
14560 PRINT#3; \ \ ; TO (14-LEN(xe$));: printerstreng xe$;
14570 PRINT#3; TO (26-LEN(ye$));: printerstreng ye$
14580 PRINT#3; \ \ 'NR. X Y SYMBOL' \ \
14590 FOR n= startnr TO slutnr
14600 n$= n: IF n < 10: n$= '0' & n$
14610 PRINT#3; ' ' ; n$ ; TO (14 - LEN(x$(n))); x$(n);
14620 PRINT#3; TO (26 - LEN(y$(n))); y$(n); TO 35; p$(n)
14630 IF NOT (n - startnr + 1) MOD 10: PRINT#3
14640 IF NOT (n - startnr -39) MOD 40 AND (slutnr + ktrtal) > n THEN
14650 PRINT#3; \ 'FORTSÆTTES ...' : BPUT#3,12: PRINT#3; \ ... FORTSAT'
14660 PRINT#3; \ 'NR. X Y SYMBOL' \
14670 END IF
14680 END FOR n: PRINT#3
14690 IF ktrtal= 0 OR ktr$== 'i': BPUT#3;12: CLOSE#3: RETURN
14700 FOR n= 101 TO 100 + ktrtal
14710 PRINT#3; n;TO (14-LEN(x$(n))); x$(n);TO (26-LEN(y$(n))); y$(n);
14720 IF ktr$=='p': PRINT#3;TO 35;CHR$(ktrp): ELSE PRINT#3;TO 35;p$(n)
14730 END FOR n: BPUT#3,12: CLOSE#3
14740 END DEFine inputudskrift

```

linie 14000 PROCEDURE "udjævnudskrift".
 linie 14010 For valg\$ = "o+x": Først "x", så "o" (14250 ff).
 linie 14060 ff Udskrifter. udj\$(n), se kommentar til linie 11700 ff.
 linie 14120 Se "sortertal", linie 19600.
 og 14280 Parameter 3 giver udskrift i kanal 3, incl. filnavn
 mv, mens parameter 30 kun giver udskrift i kanal 3
 uden (gentagelse af) filnavn mv.

linie 14500 PROCEDURE "inputudskrift".
 linie 14510 ktrp = 186 for QL-printer, 137 for CPA80S-printer.
 Bruges i linie 14720 til udskrift af polygon-symbol.

linie 14560 PROCEDURE "printerstreng" (15000).
 Efter udsendelsen af IFH-rapport 180 er der i linie
 15130 tilføjet "PRINT#3", således at begge printere
 skriver de to akseenheder på hver sin linie, så der
 er plads til længere betegnelser.

linie 14640 Sideskift, ny overskrift for mere end 40 punkter.

linie 14690 ktrtal OG ktr\$, da der kan være indlæst DATA for
 en kontrollinie, selv om der aktuelt er sagt "i",
 dvs at der ikke skal tegnes en kontrollinie.
 Uaktuelle kontrollinie-DATA skal ikke udskrives.

```

15000 DEFine PROCedure printerstreng (strg$)
15010 LOCAL b$, codep, p
15020 IF pr$== 'c': PRINT#3; strg$: RETURN
15030 FOR p= 1 TO LEN(strg$)
15040   codep= CODE(strg$(p)): b$= strg$(p)
15050   IF codep= 137: b$= 'O': BPUT#3; 27, 83, 0
15060   IF codep= 131: b$= '3': BPUT#3; 27, 83, 0
15070   IF codep= 160: b$= '2': BPUT#3; 27, 83, 0
15080   IF codep= 182 THEN
15090     BPUT#3;27,76,13,0,8,4,2,255,128,128,128,128,128,128,128,128
15100     BPUT#3; 27, 84: IF p= LEN(strg$): RETURN : ELSE NEXT p
15110   END IF
15120   PRINT#3; b$ ;: BPUT#3; 27, 84
15130 END FOR p: PRINT#3
15140 END DEFine printerstreng

```

```

15200 DEFine PROCedure ciffertal (tal, cifre)
15210 LOCAL t1, fortegn, logt, expo, læn, dec: t1= tal
15220 IF t1= 0: tal$= FDEC$( 0,cifre + (cifre > 1),cifre - 1): RETURN
15230 fortegn= 1: IF t1 < 0: fortegn= -1
15240 t1 = ABS(t1)
15250 IF t1 <= 1E-5: tal$= FDEC$(0,cifre+(cifre > 1),cifre-1): RETURN
15260 logt= INT( LOG10(t1) ): expo= cifre - logt - 1
15270 t1 = INT ( t1 * 10^expo + .51 )
15280 IF expo < 1 THEN
15290   tal$= FDEC$( t1, cifre, 0): tal$= tal$ & FILL$('0', -expo)
15300 ELSE
15310   t1= t1 * 10^-expo
15320   IF logt < 1: læn= expo + 2: dec= expo
15330   IF logt > 0: læn= cifre + 1: dec= expo
15340   tal$= FDEC$( t1, læn, dec)
15350 END IF
15360 IF fortegn < 0: tal$= '-' & tal$
15370 END DEFine ciffertal

```

GENERELT ANVENDELIGE PROCEDURER, linie 15000..19900.

linie 15000 PROCEDURE "printerstreng".

linie 15020 Da CPA80S-printeren har tegnene DOWNLOADED fra boot, linie 550, eller via < ALT > + q, kan det følgende overspringes. QL-printeren har imidlertid ikke denne DOWNLOAD-facilitet.

linie 15050 f Smlgn. linie 1350 ff.
 < CTRL > + < SHIFT > 0 / 3 / 2 / v har ASCII= codep.
 codep 137, 131, 160 giver små, hævede karakterer, da de medfører BPUT#3; 27, 83, 0.
 codep 182 er < CTRL > + < SHIFT > v = kvadratrods.
 Her indføres en ad hoc brugerdefineret karakter i QL-printeren med 13 koder for pixelkolonner, se dels brugsanvisningen, dels IFH-rapport 178 og 179.

linie 15100 BPUT#3; 27, 84 giver normal skrift.

linie 15120 Udskrift af karakter, normal skrift.

linie 15200 PROCEDURE "cifertal".

linie 15220 Hvis tal= 0, kan FEDC\$ umiddelbart bruges som vist, idet fx cifre = 2 giver 0.0 (3 pladser, 1 decimal)
 og cifre = 1 giver 0 (1 plads, 0 decimaler).

linie 15250 Meget små tal. Prøv !

linie 15280 f Heltal. cifre - (logt + 1) < 0.

linie 15300 ff Decimaltal.

```

15400 DEFine PROCedure talput ( kanal,linie,tab,min,max,ent,slut )
15410 LOCAl tastind, kontrol, k, p, lenn, n, m, tal
15420 REPeat tastind
15430 AT# kanal; linie, tab: INPUT#kanal; tal$
15440 REMark GØR ttal og tal$ GLOBALE ! talput FRIGIVER DEM BEGGE.
15450 IF slut AND tal$== 'esc': ttal= max + 1: EXIT tastind
15460 REMark IF tal$ == 'i': instruktion
15470 k= 0: p= 0: lenn= LEN (tal$)
15480 REPeat kontrol
15490 IF lenn= 0 THEN
15500 IF ent AND KEYROW(1)=1: ttal= min -1: EXIT tastind
15510 EXIT kontrol
15520 END IF
15530 IF tal$(1)= '-' AND lenn > 1: k= 1
15540 FOR m= 1 + k TO lenn
15550 IF tal$(m) INSTR '0123456789': k= k+1
15560 IF tal$(m)= '.' AND (lenn-1)*(1-p) > 0: p= 1 : k= k + 1
15570 END FOR m
15580 IF k < lenn OR p > 1: EXIT kontrol: ELSE tal= tal$
15590 REMark minus og punktum alene er ikke et tal. Max. 1 punktum
15600 SElect ON tal= min TO max: ttal= tal: EXIT tastind
15610 EXIT kontrol
15620 END REPeat kontrol
15630 AT#kanal; linie, tab: PRINT#kanal; FILL$(' ',lenn)
15640 END REPeat tastind
15650 END DEFine talput

15800 DEFine PROCedure svarput(kanal,linn,tab,strg$,kontrast,papir,blæk)
15810 LOCAl k,v,loop: k= kanal
15820 REMark kontrast: 0=INTET 1=RESULTAT 2='?' 3='?' & RESULTAT
15830 INK#k;4: IF ( papir + blæk= 6 ) OR ( papir + blæk = 7 ): INK#k;7
15840 IF kontrast > 1: PAPER#k; blæk: AT#k; linn, tab: PRINT#k; ' '
15850 REPeat loop
15860 OVER#k;-1:AT#k;linn,tab:PRINT#k;'?': OVER#k;0: q$= INKEY$(#k,12)
15870 IF LEN(q$) AND q$ INSTR strg$ THEN
15880 PAPER#k; papir: INK#k; blæk
15890 IF kontrast= 1 OR kontrast = 3: PAPER#k; blæk: INK#k; papir
15900 AT#k; linn,tab: PRINT#k; q$: EXIT loop
15910 END IF
15920 END REPeat loop
15930 PAPER#k;papir:INK#k;blæk: REPeat vt: IF INKEY$(#k,20)<>q$:EXIT vt
15940 END DEFine svarput

```

PROCedurerne "talput" og "svarput" har to formål:

- at opnå kontrol med brugerens INPUT, der accepteres eller afvises med en ny indtastningsmulighed, på samme sted på skærmen. "talput" til tal, "svarput" til korte strenge. Se også "tput" i IFH-rapport 177, tillægget.
- at spare indtastningsbesvær: talput (kanal,linie,tab,....) er fx meget nemmere end: AT#n; linie,tab : INPUT#n; tal\$, suppleret med en række IF'er til kontrol af min, max, < ENTER >, < ESC >.

Bemærk, at "svarput"s kontrast i en række tilfælde kan benyttes til permanent at ændre PAPER og INK i et vindue, se linie 15930, men så skal selve indtast-situationens farver overvejes nøje !

linie 15400 PROCEDURE "talput".
Der er tre specialtilfælde, hvoraf de to første kun accepteres, hvis parameter slut hhv ent er > 0:

linie 15450 "esc" accepteres, med ttal = max + 1

linie 15490 < ENTER > accepteres, med ttal = min - 1.

linie 15460 Denne mulighed er udnyttet i denne programversion.

Som eksempel kan henvises til linie 4180 ff, hvor programmet afgør, hvad en for stor / lille talværdi skal bruges til.

linie 4180 ff ligger i en indtastløkke !

Tilsvarende medfører < ENTER >, at INPUT ignoreres, dvs at variablen beholder sin gamle værdi (evt. 0)

linie 15470 k = antal talkarakterer i INPUT-strengen.
p = antal punktummer.

linie 15530 Et minustegn accepteres som et tal, hvis det er den første karakter i en fler-karakter-streng.

linie 15560 Eet punktum accepteres i en fler-karakter-streng.

linie 15630 Om igen, fejl, det gamle fjernes.

linie 15800 PROCEDURE "svarput"
"svarput"s kanal, linn, tab, strg\$ svarer til
"talput"s kanal, linie, tab, min, max.
Her skal kontrolleres, at strengen har det rette indhold og ikke er en nulstreng (' ' INSTR 'xyz' er ok).
"svarput" er ikke foolproof for flerkarakter-streng, smlgn. linie 21340, der heller ikke er sikker, omend acceptabel, da brugeren næppe ved ubevidste fejltast kan opnå den forkerte funktion.

linie 15830 f styrer farve / kontrast på det blinkende ? og på det indtastede resultat. Se REMark i linie 15820.

EKSEMPEL PAPER 2 : INK 7. papir = 2. blæk = 7. papir + blæk = 9.

kontrast	0	1	2	3
15830	PAPER 2 INK 4	PAPER 2 INK 4	PAPER 2 INK 4	PAPER 2 INK 4
15840	PAPER 2 INK 4	PAPER 2 INK 4	PAPER 7 INK 4	PAPER 7 INK 4
felt	rødt	rødt	hvidt	hvidt
15860	 OVER - 1			
?	hvidt	hvidt	rødt	rødt
15880	PAPER 2 INK 7	PAPER 2 INK 7	PAPER 2 INK 7	PAPER 2 INK 7
15890	PAPER 2 INK 7	PAPER 7 INK 2	PAPER 2 INK 7	PAPER 7 INK 2
felt	rødt	hvidt	rødt	hvidt
streng	hvid	rød	hvid	rød
kontrast	0 / 0	0 / ja	ja / 0	ja / ja

15930 DET OPRINDELIGE : PAPER 2 : INK 7

```

16000 DEFine PROCedure tidspunkt
16010   LOCal fildata$, fl, fundet, tids$, maned, dtal$, tim$
16020   LOCal spmloop1, spmloop2, datoloop
16030   REPeat spmloop1
16040     CSIZE 2,0: CSIZE#2; 2,0: CLS 1: FOR n= 9,30: BLOCK 240,1,12,n,7
16050     STRIP 7: INK 2: AT 1, 1: PRINT 'SKAL FLP      ELLER      '
16060             AT 2, 1: PRINT 'BENYTTES TIL DATA      '
16070     FOR n= 132,240: BLOCK 2,10, n,10,2
16080     PAPER 2: INK 7: AT 1,10: PRINT 1: AT 1,19: PRINT 2
16090     BLOCK 2,22,10,9,7: BLOCK 14,10,228,20,2
16100     svarput 1,2,19,'12',3,7,2: BLOCK 2,10,240,20,2
16110     fldat$= 'flp' & q$ & ' '
16120     IF fldat$ == DATAD$: fl= 0: ELSE : fl= q$: DATA_USE fldat$
16130     PAPER 2: INK 7: CLS: AT 1,1: PRINT'DATA-FLOPPY:' & fldat$
16140     AT 4,1: PRINT 'FLOPPY KLAR J / N'
16150     svarput 1,4,20,'j',0,2,7: AT 4,20: PRINT ' '
16160     IF FOP_DIR(#3,fldat$)= 0 : EXIT spmloop1
16170     IF FOP_DIR(#3,fldat$):AT 6,1: PRINT fldat$(1 TO 4)&' MANGLER.';
16180   END REPeat spmloop1
16190   IF fl: DATA_USE flp1_: IF fl= 1: DATA_USE flp2_
16200   CLOSE#3: FOR n= 4,6: AT n,1: PRINT FILL$ (' ',21)

16220   filservice 2
16230   REPeat spmloop2
16240     AT 1,0: CLS#1,2: AT 9,1: PRINT'TAST n ELLER k : '
16250     AT 3,1: PRINT'SKAL DER OPRETTEES EN'\ ' helt NY TEGNING'
16260     AT 6,1: PRINT'-ELLER TEGNES VIDERE'\ ' med KENDTE DATA'
16270     STRIP 7: INK 2: AT 4,6: PRINT 'NY': AT 7,6: PRINT 'KENDTE'
16280     AT 4,18: PRINT' n ': AT 7,18: PRINT' k ': AT 9,18: PRINT ' '
16290     svarput 1,9,19,'nk',1,2,7: nygl= 1: IF q$== 'k': nygl= 2
16300     :
16310     PAPER 2: INK 7: CLS: AT 1,1: PRINT 'OPGAVENS NAVN ER'
16320     AT 2,1: INPUT filnavn$: fildata$= fldat$ & filnavn$ &'_data'
16330     fundet= 0: IF FOP_IN(#3,fildata$)= 0: fundet= 1: CLOSE#3
16340     IF nygl + fundet== 1: tids$= FILL$(' ',17):CLS#2: EXIT spmloop2
16350     IF nygl + fundet== 3 THEN
16360       OPEN_IN#6; fildata$: GET#6; dato$: CLOSE#6
16370       AT 16,1: PRINT 'KENDTE dato$ ='\ ' '; dato$
16380       tids$= FILL$(' ',17): CLS#2: EXIT spmloop2
16390     END IF
16400     AT 2,0: CLS#1,1: AT 20,1: PRINT 'DATA-FLOPPY: ' & fldat$
16410     IF nygl= 2: AT 20,1: PRINT 'NAVNET FINDES IKKE.': filservice 2
16420     IF nygl= 1: AT 20,1: PRINT'NAVNET ER BRUGT FØR.': filservice 2
16430   END REPeat spmloop2
16440   IF nygl=2: STRIP 7: INK 2: AT 20,1: PRINT 'NAVNET FUNDET.      '

16460   PAPER 2: INK 7: CLS#2
16470   AT 5,1: PRINT'DEN DATO DER NU SKAL' \ ' INDSTAVES BRUGES VED'
16480   AT 7,1: PRINT'LAGRING AF DATA.' \ \ ' TIL FIGURUDTEGNINGEN'
16490   AT 10,1: PRINT'KAN DER ANGIVES EN' \ ' ANDEN DATO,fx DATOEN'
16500   AT 12,1: PRINT'FOR EN FORSØGSSERIE.'

16520   REMark UDSKRIFT FORTSÆTTER PÅ NÆSTE SIDE

```

linie 16000 PROCedure "tidspunkt".

linie 16040 En række kommandoer for at få spørgsmål og svar til at stå i kontrast, CSIZE 2,0, se også IFH-rapport 177.

linie 16110 DATA-floppyen er nu fldat\$. Program-floppyen er stadig det, der blev angivet i boot, linie 110, DATA_USE etc, (DEFAULT), i reglen FLP1.

linie 16120 Hvis DEFAULT (program) og fldat\$ (DATA-filer) er den samme floppy, er fl = 0. Ellers er fl = 1, og der omstilles midlertidigt til DATA-floppyen som DEFAULT.

DETTE ER NØDVENDIGT, DA TO AF DTH'S FEM INTERFACES HAR EN INDBYGGET FUNKTION, DER FÅR QL TIL AT BRUGE DEFAULT FLOPPYEN, HVIS DEN ANGIVNE FLOPPY IKKE ER ISAT ! KONTROLLEN I linie 16160 FUNGERER DA IKKE, OG SAVE KAN BLIVE UDFØRT PÅ DEN GALE FLOPPY, HVIS DEN RIGTIGE MANGLER.

linie 16160 Kontrol af om DATA-floppyen er på plads.

linie 16220 PROCeduren "filservice" (17000) udskriver alle filer med tilføjelsen _data til et navn, dvs alle DATA-filer i dette program, SAVED tidligere.

linie 16290 nygl = 2 har betydning, fx i linie 470 for LOAD af den kendte fil.

linie 16310 ff Her kontrolleres om det opgivne navn
 - for nye filer: er brugt før >> fejl
 - for kendte filer: ikke eksisterer >> fejl
 Er der fejl, gives ny indtastmulighed og filservice.

Bemærk, at de nye interfaces skelner mellem STORE og små bogstaver, og at navne IKKE må ende på Æ Ø Å.

linie 16460 Dato skal indtastes, så filer datostemples ved SAVE. De fleste filer bliver revideret - og lagret på flere forskellige disketter.

```

16600 REMark tidspunkt FORTSAT FRA FORRIGE SIDE
16610 REPEAT datoloop
16620   REMark tids$= 02.MAJ 1988: 14.1 / 12.MARTS 87: 21.0
16630   AT#2;1,1: PRINT#2;'INDTAST DATO : '
16640   AT#2;3,1: PRINT#2;'ARSTAL 86..99 '
16650   AT#2;6,1: PRINT#2;'MANED 1..12 '
16660   AT#2;9,1: PRINT#2;'DATO 1..31 '
16670   AT#2;12,1: PRINT#2;'TIME 0.0..23.9 '
16680   talput 2,4,11,86,99,0,0: tids$(10 TO 12)= INT(ttal)&'::'
16690   talput 2,7,11, 1,12,0,0: maned= INT(ttal): maned_til_streng
16700   talput 2,10,11,1,31,0,0: dtal$= INT(ttal)
16710   IF ttal < 10 THEN dtal$='0'& INT(ttal)
16720   tids$(1 TO 3)= dtal$&'.'
16730   talput 2,13,11,0,23.9,0,0: tim$= FILL$(' ',4)
16740   tim$(1 TO 3)= FILL$('0',<ttal < 10>) & INT(ttal) & '.'
16750   tim$(4)= 10 * ttal MOD 10
16760   tids$(14 TO 17)= tim$(1 TO 4)
16770   AT#2;16,1: PRINT#2;'NYE dato$ =' \ ' ';tids$
16780   AT#2;18,1: PRINT#2;'KORREKT J / N '
16790   svarput 2,18,17,'jn',0,2,7: IF q$== 'j': EXIT datoloop
16800   IF q$== 'n': CLS#2
16810 END REPEAT datoloop
16820 SDATE 1900+tids$(10 TO 11),maned,dtal$,tim$(1 TO 2),tim$(4)*6,0
16830 dato$= tids$: starttid= DATE: FOR n= 0 TO 2: CSIZE#n; 0,0
16840 END DEFINE tidspunkt

17000 DEFINE PROCEDURE filservice ( ch )
17010   PAPER#ch;2: INK#ch;7: CSIZE#ch; 0,0: CLS#ch: BORDER#ch; 9-ch*1.5
17020   PRINT#ch; fldat$ ; ' INDEHOLDER:'
17030   IF ch= 4: PRINT#4; \ 'TAST < F5 >,' \ 'HVIS VINDUET FYLDES.'
17040   IF ch= 2: PRINT#2; \ 'TAST < F5 >,' HVIS VINDUET FYLDES.'
17050   WSTAT#ch; fldat$ & ' _data'
17060   INPUT#ch; \ 'RETUR TIL PROGRAM: '; FILL$(' ',ch); ' < ENTER > '; z$
17070   IF ch= 2: BORDER#2;1,0: CSIZE#2; 2,0
17080   IF ch= 4: BORDER#4;0 : CLS#4
17090 END DEFINE filservice

17500 DEFINE PROCEDURE maned_til_streng
17510   SELECT ON maned
17520     =1: tids$(4 TO 9)= 'JAN.19'
17530     =2: tids$(4 TO 9)= 'FEB.19'
17540     =3: tids$(4 TO 9)= 'MARTS '
17550     =4: tids$(4 TO 9)= 'APRIL '
17560     =5: tids$(4 TO 9)= 'MAJ 19'
17570     =6: tids$(4 TO 9)= 'JUNI '
17580     =7: tids$(4 TO 9)= 'JULI '
17590     =8: tids$(4 TO 9)= 'AUG.19'
17600     =9: tids$(4 TO 9)= 'SEPT. '
17610     =10:tids$(4 TO 9)= 'OKT.19'
17620     =11:tids$(4 TO 9)= 'NOV.19'
17630     =12:tids$(4 TO 9)= 'DEC.19'
17640   END SELECT : REMark GL.DAGS DATO. TIMER MED 1 DECIMAL
17650 END DEFINE maned_til_streng

```

linie 16680 ff INPUT kontrolleres, dog er der ikke gjort forsøg på
at stoppe 31. februar og lign.

linie 16690 Se PROCeduren "måned_til_streng" nedenfor.
Dansk notation med forkortelser af måned / årstal,
så tid\$ / dato\$ altid har samme længde.

linie 16730 timer med een decimal (6 minutter) er måske lidt
søgt, men der er ikke plads i layout til to.

linie 16780 Fortrydelsesmulighed.

linie 16820 SDATE. Filer datostemples fra nu af korrekt.

linie 16830 starttid af hensyn til senere lagringer af filer.

linie 17000 PROCedure "filservice" udnytter WSTAT.

linie 17500 PROCedure "måned_til_streng".
Format, se linie 16620.

```

18000 DEFine PROCedure sluttid
18010   LOCal timetal, times$
18020   timetal= dato$(14 TO 17) + INT((DATE-starttid)/360 +.5)/10
18030   times$= timetal: IF timetal < 10: times$='0'&times$
18040   IF LEN(times$)= 2: times$= times$&'.0'
18050   dato$= dato$(1 TO 12) & ' ' & times$
18060 END DEFine sluttid

18200 DEFine PROCedure loadepoces
18210   LOCal load$, gldato$, n, p
18220   CLS#0: CLS#4: load$= fldat$ & filnavn$ & '_data'
18230   PRINT#0; ' EXISTERENDE DATA FOR'
18240   PRINT#0; ' '; filnavn$ \ ' LOADes FRA ' & fldat$
18250   OPEN_IN#6; load$
18260   GET#6; gldato$, startnr, slutnr, dat$, fil$, xe$, ye$, ktrtal
18270   FOR n= 1 TO 6: GET#6; g$(n)
18280   FOR n= startnr TO slutnr : GET#6; x$(n), y$(n), p$(n)
18290   FOR n= 101 TO 100 + ktrtal: GET#6; x$(n), y$(n), p$(n)
18300   CLOSE#6: PRINT#0; ' '; filnavn$ \ ' _data LOAded': PAUSE 50

18320   xantal= (g$(2)-g$(1)) / g$(3): yantal= (g$(5)-g$(4)) / g$(6)
18330   yskala= g$(5) - g$(4): xs= (g$(2) - g$(1)) / (g$(5) - g$(4))
18340   SCALE 1.25 * yskala, g$(1) /xs -.15 *yskala, g$(4) -.15 *yskala
18350   st$= FILL$('0',startnr < 10) & startnr
18360   sl$= FILL$('0',slutnr < 10) & slutnr
18370   IF ktrtal: ktr$= 'k': IF p$(102)= 'a': ktr$= 'p'
18380   pkt$= p$(startnr) : xykanal4
18390   FOR p= st$ + 1 TO sl$: IF pkt$ <> p$(p): pkt$='ox': EXIT p
18400 END DEFine loadepoces

18500 DEFine PROCedure xykanal4
18510   LOCal n, n$, p
18520   PAPER#4;2: INK#4;7: CLS#4: BORDER#4;1: p= 0
18530   PRINT#4; \ ' NR. X Y $'
18540   UNDER#4;1: PRINT#4; FILL$(' ',25): UNDER#4;0
18550   IF ktrtal= 0: FOR n= startnr TO slutnr
18560   IF ktrtal: FOR n= startnr TO slutnr, 101 TO 100 + ktrtal
18570   p= p + 1: n$= n: IF n < 10: n$= '0' & n$
18580   IF NOT p MOD 10: UNDER#4;1
18590   PRINT#4; FILL$(' ',4-LEN(n$)); n$;
18600   PRINT#4; FILL$(' ',8-LEN(x$(n))); x$(n);
18610   PRINT#4; FILL$(' ',8-LEN(y$(n))); y$(n);
18620   PRINT#4; FILL$(' ',3); p$(n); ' ': UNDER#4; 0
18630   IF NOT p MOD 10:CLS#0: INPUT#0; \ ' FORTSÆT MED <ENTER> '; z$
18640   END FOR n
18650   BORDER#4;0: CLS#0: PRINT#0; \ ' PRINTERUDSKRIFT J / N'
18660   svarput 0,2,21,'jn',0,2,7: IF q$== 'j': inputudskrift
18670 END DEFine xykanal4

```

linie 18000 PROCEDURE "sluttid" bruges i "saveproces" (18800).

linie 18200 PROCEDURE "loadepoces".

linie 18260 GET (PUT) er meget lettere end INPUT (PRINT).
Alle DATA-typer kan blandes, heltal, decimaltal,
streng. De skal naturligvis have en bestemt rækkefølge ved GET og PUT. Ønsker man at bruge disse kommandoer til enkelte DATA ved hjælp af pointer, bør de tre typer ikke blandes !

linie 18320 ff De nødvendige DATA hentes.
Programmets afledte variabler beregnes.
Der er pt ingen indlæsning af udjævningsfiler !

linie 18380 DATA udskrives, se PROCEDUREN "xykanal4" nedenfor.

linie 18500 PROCEDURE "xykanal4".
Udskrift af nummer, x, y, symbol, 10 numre ad gangen.
Incl. en eventuel kontrol-polygon, -kurve (18560).

linie 18650 Jeg anbefaler generelt at få denne printerudskrift nu.

```

18800 DEFine PROCedure saveproces
18810  LOCal save$, save2$, z$, fl1, fl2, k, fl3

18830  CLS#0: PRINT#0; ' ER ' & fldat$(1 TO 4) & ' KLAR ? J / N'
18840  REPeat fl1
18850    svarput 0,1,22,'j',0,2,7: IF FOP_DIR(#3,fldat$)=0: EXIT fl1
18860  END REPeat fl1: CLOSE#3

18880  CLS#0: PRINT#0;\ ' LAGRING AF'\ ' - INPUTDATA      i'
18890  PRINT#0;' - FEJLUDJÆVNING f'\ ' - BADE OG          b'
18900  svarput 0,4,22,'ifb',0,2,7: IF hæld$= '' AND q$== 'f': RETURN

18920  sluttid
18930  IF q$ INSTR 'ib' THEN
18940    REPeat fl2
18950      k= 0: CLS#0: PRINT#0;\ ' FILNAVNEN ER NU' \ ' '; filnavn$
18960      PRINT#0;' TAST ok ELLER NYT NAVN': INPUT#0;' '; z$
18970      IF LEN(z$) < 1: NEXT fl2
18980      IF z$=='ok' OR z$=='j': z$= filnavn$: k= 1
18990      filnavn$= z$: save$= fldat$ & filnavn$ & '_data'
19000      IF FOP_IN(#3,save$) OR k= 1: EXIT fl2
19010      CLS#0: PRINT#0;' FILNAVNEN EKSISTERER' \ ' BEKRÆFT IGEN MED ok'
19020      PRINT#0;' OG DEN GL.FIL SLETTER': filservice 4
19030    END REPeat fl2: CLOSE#3

19050  CLS#0: PRINT#0;' EXISTERENDE DATA FOR'
19060  PRINT#0;' '; filnavn$ \ ' LAGRES PÅ ' & fldat$: DELETE save$
19070  OPEN_NEW#6; save$
19080    PUT#6; dato$, startnr, slutnr, dat$, fil$, xe$, ye$, ktrtal
19090    FOR n= 1 TO 6: PUT#6; g$(n)
19100    FOR n= startnr TO slutnr : PUT#6; x$(n), y$(n), p$(n)
19110    FOR n= 101 TO 100 + ktrtal: PUT#6; x$(n), y$(n), p$(n)
19120  CLOSE#6: CLS#0: PRINT#0; \ ' '; filnavn$ \ ' '_data LAGRET'
19130  PRINT#0;' dato$ ( slut ) = ' \ ' '; dato$: PAUSE 100
19140  END IF

19160  IF q$ INSTR 'bf' AND hæld$ <> '' THEN
19170    udjævnfilservice
19180    REPeat fl3
19190      CLS#0: PRINT#0; ' DATASÆTTET KAN HAVE'
19200      PRINT#0;' FLERE SÆT UDJÆVNINGER.' \ ' ANGIV SUPPL. TEKST TIL'
19210      PRINT#0;' '; filnavn$ & '_udj_'; INPUT#0; z$
19220      save2$= fldat$ & filnavn$ & '_udj_' & z$
19230      IF FOP_IN(#3,save2$): EXIT fl3
19240      PRINT#0;' FILNAVNEN EKSISTERER': udjævnfilservice
19250    END REPeat fl3
19260    OPEN_NEW#6; save2$
19270    FOR n= 1 TO 40: PUT#6; fjl(n)
19280    PUT#6; hæld$, valg$, pkt$, rg$, rgtal, rgoantal, rgxantal
19290  CLOSE#6: PRINT#0;' '; filnavn$ & '_udj_' & z$; ' LAGRET': PAUSE 100
19300  END IF
19310 END DEFine saveproces

```

linie 18800 PROCEDURE "saveproces".

linie 18920 Tidsjusteret dato, PROCEDURE "sluttid" (18000).

linie 18950 k= 0: DATA-filnavn angives som en mulighed.

linie 18960 Brugervalg af filnavn: ok / j eller nyt navn.

linie 18970 < ENTER > accepteres ikke her.

linie 18980 DATA-filnavn accepteret. z = 1: z\$ = DATA-filnavn.

linie 18990 DATA-filnavn= z\$, dvs det gl. navn / et nyt navn.

linie 19000 Hvis DATA-filnavn er nyt, (FOP_IN), eller har fået ok, (k = 1), er alt i orden.

linie 19010 k = 0 og det nye navn eksisterer. Nyt ok kræves.

linie 19240 Udjævningsfiler kan ikke bruge eksisterende navne.

linie 19270 fj1(40,1), (k= 0: 1..15, k= 20: 21..35) indeholder:

hæld\$ =		"r"	"h"	"p"	"l"	"v"
1	gnx	+	+	+	+	+
2	gny	+	+	+	+	+
3	spredx	+	+	+	+	+
4	spredy	+	+	+	+	+
5	xhæld	+	+	+		
6						
7	korrel	+	+	+		
8	xfast			+		
9	yfast			+		
10	yvenst	+	+	+		
11	yhøjre	+	+	+		
12	xnedad	+	+	+		
13	xopad	+	+	+		
14	xakse	+	+	+		
15	yakse	+	+	+		

```

19500 DEFine PROCedure udjævnfilservice
19510  PAPER#4;2: INK#4;7: CLS#4: BORDER#4;3
19520  PRINT#4; fldat$ ; ' HAR FØLGENDE' \ 'UDJÆVNING-FILER:'
19530  PRINT#4; \ 'TAST < F5 >,' \ 'HVIS VINDUET FYLDES.'
19540  WSTAT#4; fldat$ & filnavn$ & '_udj_'
19550  PAUSE 50: CLS#0: INPUT#0; \ ' FØRTSÆT PROGRAM: '\ < ENTER > '; z$
19560  BORDER#4;0
19570  END DEFine udjævnfilservice

19600 DEFine PROCedure sortertal ( kanal )
19610  LOCAL loop1, loop2, p, t, k, slut
19620  IF kanal= 3 THEN
19630    OPEN#3,ser1: BAUD baudtal: TRA trap
19640    BPUT#3,27,108,10,27,81,77,27,67,72,27,78,10: PRINT#3; \
19650    PRINT#3;'FILENS NAVN : '; filnavn$ ; \ ;'EDB-DATO : '; datos$
19660    PRINT#3;'FIGURNAVN : '; fil$ ; \ 'FIGUR-DATO : '; dat$
19670  END IF

19690  IF kanal MOD 3 = 0 THEN
19700    kanal= 3: PRINT#3; \ 'SERIEN '; st$ ; '-' ; sl$ ; ' HAR ';
19710    PRINT#3; rgtal; ' PUNKTER MED SYMBOLET '; rg$ ; '.'
19720    IF rgtal== slutnr - startnr + 1: PRINT#3: RETURN
19730    PRINT#3; 'NUMRENE ER: ';
19740  END IF

19760  p= 0: t= 0 : k= 0: rgtal= rgoantal: slut= rgtal
19770  IF rg$== 'x': k= 50: rgtal= rgxantal: slut= rgtal + 50
19780  REPEAT loop1
19790    k= k + 1: p= k: z$= FILL$('0',rgrnr(p) < 10) & rgrnr(p)
19800    PRINT#kanal; FILL$(' ',kanal=4); z$;
19810    REPEAT loop2
19820      IF rgrnr(k+1) <> rgrnr(k) + 1: EXIT loop2
19830      k= k + 1: IF k= slut: EXIT loop2
19840    END REPEAT loop2
19850    z$= FILL$('0',rgrnr(k) < 10) & rgrnr(k)
19860    IF k > p : PRINT#kanal; '-'; z$ ;: t= t + 1
19870    IF k < slut : PRINT#kanal; ', ' ;: t= t + 1
19880    IF t MOD 3 = 0 AND kanal = 4: PRINT#4
19890    IF k = slut : EXIT loop1
19900  END REPEAT loop1
19910  IF kanal= 3: PRINT#3; \ \
19920  REMark sortertal 3 SKRIVER IKKE "BPUT#3,12: CLOSE#3" ETC
19930  END DEFine sortertal

```

linie 19500 PROCedure "udjævnfilservice" bruger WSTAT.

linie 19600 PROCedure "sortertal"

linie 19620 På printeren udskrives disse oplysninger. Kanal 4, på skærmen, får dem direkte fra programmet, smlgn. linie 11110 ff.

linie 19690 Som 19620, dog udskrives de fælles oplysninger ikke to gange, kun punktoptællinger, for valg\$= 'o+x', se linie 14120 og 14280.

linie 19790 I hovedløkken, loop1, k tælles op, og p sættes lig k, som udgangspunkt for optælling af en eventuel "serie" af fortløbende numre.

linie 19800 På skærmen (kanal 4) sættes en tom plads foran z\$.

linie 19810 f loop2
 til optælling af "serier" af fortløbende numre.
 k tælles op, men p er konstant.
 loop2 forlades, når rgrnr(k) ikke indeholder fortløbende numre i sit register over punktnumre med det aktuelle symbol.

linie 19860 f Der sættes "-" mellem fortløbende numre og "," efter "serier" og enkeltnumre.
 Eksempel: 12-15,21-23,26,29,31-36

 t bruges til passende linieskift i den smalle kanal 4.

```

20000 DEFine PROCedure filcheck ( ordre )
20010 LOCAL skr$, k, u: k= 0
20020 IF nygl = 1: ordre= 0: RETURN
20030 IF ordre= 1: glst$= st$: glsl$= sl$: RETURN

20050 IF st$ > glst$ THEN
20060   skr$= FILL$('0',startnr < 11) & (startnr - 1)
20070   CLS#0: PRINT#0;' NR. ' ; glst$ ; ' TIL ' ; skr$
20080   PRINT#0;' VIL BLIVE SLETTET.': k= 1
20090 END IF
20100 IF glsl$ > sl$ THEN
20110   skr$= FILL$('0',slutnr < 9) & (slutnr + 1): IF k= 0: CLS#0
20120   AT#0; 2 * k, 1: PRINT#0;'NR. ' ; skr$ ; ' TIL ' ; glsl$
20130   PRINT#0;' VIL BLIVE SLETTET.': k= k + 2
20140 END IF
20150 IF st$ < glst$ OR glsl$ < sl$: GO TO 20270: REMark ARRAY UDVIDES
20170 IF k= 0: ordre= 0: RETURN : REMark INGEN ÆNDRING I start / slutnr

20190 AT#0; 4,1: PRINT#0;'OK ??? J / N': svarput 0,4,16,'jn',0,2,7
20200 IF q$== 'j' THEN
20210   IF k=1 OR k=3: FOR n= glst$ TO st$-1: x$(n)=0:y$(n)=0:p$(n)=''
20220   IF k=2 OR k=3: FOR n= sl$ TO glsl$-1: x$(n)=0:y$(n)=0:p$(n)=''
20230   ordre= 0: CLS#0: inputudskrift: CLS#0: RETURN
20240 END IF : REMark BRUGER FORTRYDER: linie 20250
20250 st$= glst$: sl$= glsl$: startnr= st$: slutnr= sl$: RETURN

20270 k= 5: CLS#0: PRINT#0; \ ' DE NYE PUNKTER ' \ ' INDTASTES NU'
20280 alt$= 'e': koordinatsystem: existdatategning: inputtekst
20290 IF st$ < glst$ THEN
20300 FOR u= st$ TO glst$-1: pt= u -1: IF p$(u)= '': alt$= 'e': xyinput
20310 END IF
20320 IF glsl$ < sl$ THEN
20330 FOR u= glsl$+1 TO sl$: pt= u -1: IF p$(u)= '': alt$= 'e': xyinput
20340 END IF
20350 alt$= 'l': ordre= 0: CLS#0: inputudskrift: CLS#0
20360 PAPER#4;2: INK#4;7: CLS#4
20370 END DEFine filcheck

21000 DEFine PROCedure kontrolkurve
21010 LOCAL ft$, fk, lp1, n, k2, pn
21020 FOR n=101 TO 108: x$(n)= '0': y$(n)= '0': p$(n)= ''
21030 OVER 0: PAPER 0: INK 7: BORDER 1,2: CLS: CLS#0: CLS#4
21040 AT 1,45: PRINT 'TAST': PRINT \   konstant1 * ( xf ^ konstant2 )'
21050 PRINT '   konstant1 * ( konstant2 ^ xf )'
21060 PRINT '   konstant1 * SIN( xf * konstant2 )'
21070 PRINT '   konstant1 * TAN( xf * konstant2 )'
21080 PRINT '   konstant1 * LOG( xf * konstant2 )'
21090 PRINT '   konstant': BLOCK 354,3,0,102,2: BLOCK 354,3,0,143,2
21100 STRIP 7:INK 2:AT 1,3:PRINT' y= f(x) OPBYGGES AF 3 LED AF FORMEN: '
21110 AT 3,45: PRINT ' pot ': AT 4,45: PRINT ' sin '
21120 AT 5,45: PRINT ' cos ': AT 6,45: PRINT ' tan '
21130 AT 7,45: PRINT ' log ': AT 8,45: PRINT ' kon '
21140 AT 12,3: PRINT ' xf = korrektion + fortegn * x '
21150 FOR n= 16 TO 20 STEP 2
21160   AT n,3: PRINT' LED1 ': AT n, 10: PRINT' + '
21170   AT n, 25 + 5 * (n > 14): PRINT' LED3 '
21180 END FOR n
21190 AT 16,42: PRINT ' TYPE   1 ': AT 18,42: PRINT ' * = 2   / = 3 '
21200 REMark

```

UDSKRIFT FORTSÆTTER PÅ NÆSTE SIDE

linie 20000 PROCEDURE "filcheck", her kun brugt i linie 1290.

linie 20020 f Nye filer uden interesse.
I første omgang skal de eksisterende start- og slutnumre bare gemmes til senere brug.

linie 20050 f Nu er (måske) nye start- og slutnumre indtastet.

linie 20060 f Er det nye startnummer højere end det eksisterende ?

linie 20110 f Er det nye slutnummer lavere end det eksisterende ?

linie 20170 $k = 0$. Der var ingen problemer.

linie 20190 $k > 0$. Må det "overskydende" slettes ?

linie 20200 f Der slettes og returneres med $k = 0$ til linie 1290.

linie 20240 Brugeren fortrød. Gamle start- og slutnumre bruges påny. Retur til linie 1290 med $k = 3$.

linie 21000 PROCEDURE "kontrolkurve".

KURVENS LIGNING DEFINERES VED :

3 LED, der hver kan have en af flg. 6 former :

```

pot      konstant1 * ( xf ^ konstant2 )
xpo      konstant1 * ( konstant2 ^ xf )
sin      konstant1 * SIN( xf * konstant2 )
tan      konstant1 * TAN( xf * konstant2 )
log      konstant1 * LOG10( xf * konstant2 )
kon      KONSTANT

```

Der er 5 TYPER forbindelser mellem de tre led :

```

1      LED1  +  LED2  +  LED3
2      LED1  +  ( LED2  *  LED3 )
3      LED1  +  ( LED2  /  LED3 )
4      ( LED1  +  LED2 ) *  LED3
5      ( LED1  +  LED2 ) /  LED3

```

Endvidere er $xf = \text{korrektion} + (\text{fortegn} * x)$

$y = f(x)$ tegnes fra $x = \text{start}$ til slut ,
idet kurven ikke tegnes i det omfang den måtte
ligge uden for det definerede udtegningsfelt.

```

21300 REMark                                kontrolkurve FORTSAT FRA FORRIGE SIDE
21310 AT 20,42: PRINT ' * = 4      / = 5 '
21320 AT 16,21:PRINT' + ':AT 18,24:PRINT' * / ': AT 20,24:PRINT' * / '
21330 AT 20, 2:PRINT' ( ' :AT 20,19:PRINT' )' : AT 16,14:PRINT' LED2 '
21340 AT 18,17:PRINT' LED2 ':AT 20,9: PRINT' ' : AT 20,13:PRINT' LED2 '
21350 PAPER 0: INK 7: PAPER#4;2: INK#4;7
21360 AT 22,4: PRINT 'TYPE NR.': talput 1,22,13,1,5,0,0: p$(105)= tal$
21370 FOR n= 1 TO 3
21380 REPEAT lp1
21390 AT#4; n*3 -2, 1: INPUT#4; 'LED' & n & ' '; ft$
21400 IF LEN(ft$)=3 AND ft$ INSTR 'potxpologsintankon': EXIT lp1
21410 END REPEAT lp1
21420 fk= CODE(ft$(1)) + 32 * (CODE(ft$(1)) < 95): p$(n+100)= CHR$(fk)
21430 IF ft$='kon': AT#4; n*3 -1, 3: PRINT#4; 'KONSTANT='
21440 IF ft$<>'kon': AT#4; n*3 -1, 3: PRINT#4; 'konstant1='
21450 talput 4, n*3 -1, 15, -32000, 32000, 0, 0: x$(n+100)= tal$
21460 IF ft$<>'kon' THEN
21470 AT#4; n*3, 3: PRINT#4; 'konstant2='
21480 talput 4, n*3, 15, -32000, 32000, 0, 0: y$(n+100)= tal$
21490 END IF
21500 END FOR n
21510 IF p$(105) INSTR '35' AND x$(103)='0' AND p$(103)='k': x$(103)='1'

21530 AT#4;12,1: PRINT#4;'korrektion=' : talput 4,12,15,-32000,32000,0,0
21540 AT#4;13,1: PRINT#4;'fortegn +,-': svarput 4,13,17, '+-', 0, 2, 7
21550 y$(104)= tal$: y$(105)= '1': IF q$= '-': y$(105)= '-1'
21560 AT#4;15,1: PRINT#4; 'x start=' \ ' x slut ='
21570 FOR n=1 TO 2: talput 4,14+n,11,-32000,32000,0,0: x$(n+103)= tal$

21590 k2= 0: pn= 0: p$(104)= 'k'
21600 FOR n=101 TO 103: k2=k2+ABS(y$(n)): IF y$(n)=1 AND p$(n)='p': pn=1
21610 IF k2= 0 OR ( pn= 1 AND k2= 1) THEN
21620 CLS#0: PRINT#0; \ ' BRUG "POLYGON" \ ' TIL RETTE LINIER.'
21630 PRINT#0; ' VENT 3 SEKUNDER.': PAUSE 150: ktrtal= 0: ktr$= 'p'
21640 FOR n= 101 TO 105: x$(n)= '0': y$(n)= '0': p$(n)= ''
21650 END IF

21670 koordinatsystem: existdatategning
21680 CLS#0: PRINT#0;\ ' ER KURVEN OK J / N': svarput 0,2,19,'jn',0,2,7
21690 IF q$= 'n': kontrolkurve
21700 END DEFINE kontrolkurve

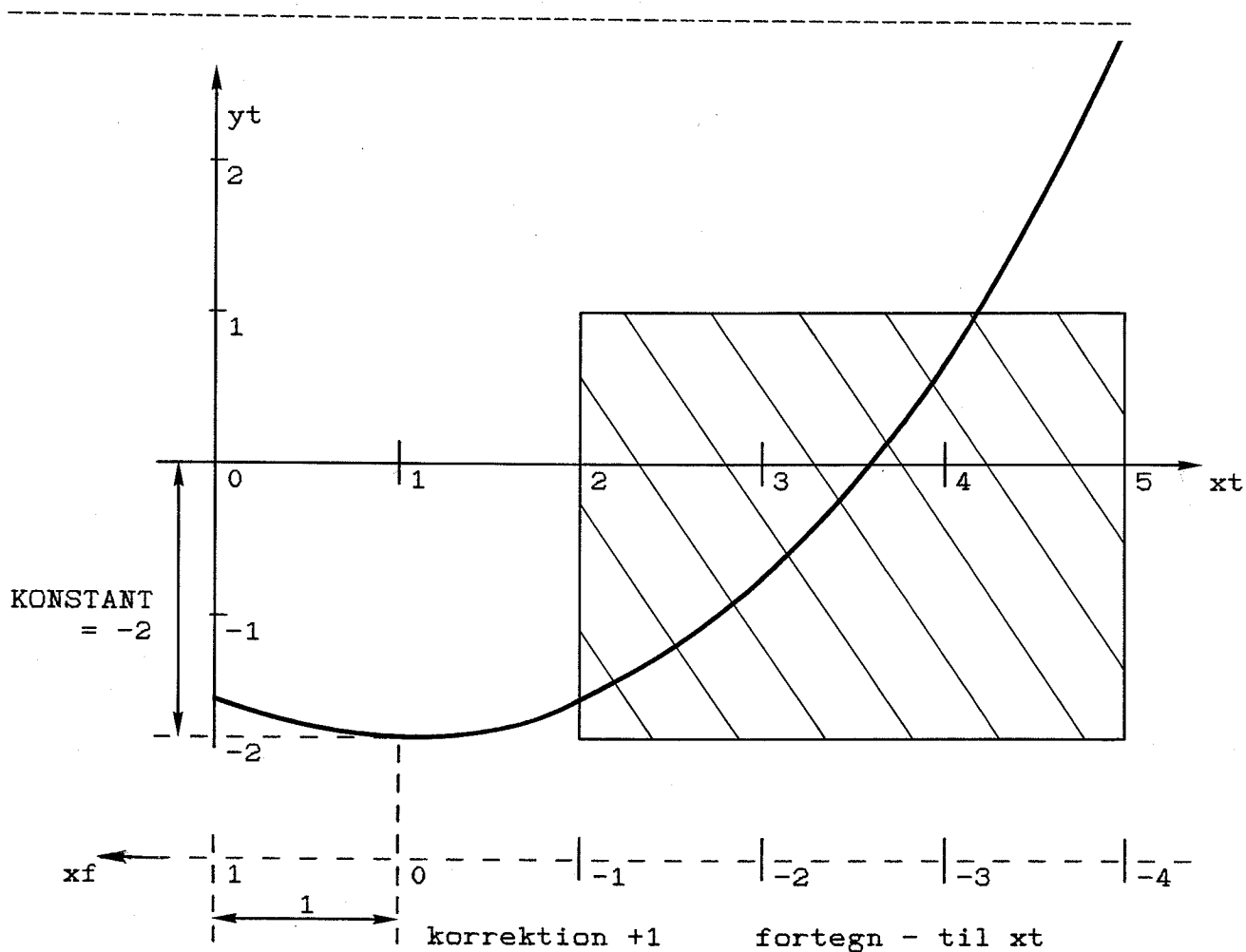
22000 DEFINE PROCEDURE tegnkurve
22010 LOCAL type, trin, retn, xf, xt, yt, gly, tl
22020 INK 2: type= p$(105): yt= g$(5): tl= 0
22030 trin= (g$(2) - g$(1)) / 137.5: retn= y$(105) * (x$(105)-x$(104))
22040 FOR xt= x$(104) TO x$(105) STEP trin
22050 IF xt < g$(1): IF retn > 0: NEXT xt: ELSE EXIT xt
22060 IF xt > g$(2): IF retn > 0: EXIT xt: ELSE NEXT xt
22070 xf= y$(104) + y$(105) * xt: gly= yt: beregny
22080 IF ABS(gly - yt) > 1E8 : EXIT xt: REMark ?? ***
22090 tl= tl + 1 + ABS(gly-yt)/trin/2: IF tl < 4: NEXT xt
22100 tl= 0: IF (yt-g$(4)) * (g$(5)-yt) >= 0 : POINT xt / xs, yt
22110 END FOR xt: INK 7
22120 END DEFINE tegnkurve

```

linie 21300 ff Kurvens variabler lagres i det samme område af punktarray'erne som en polygon:

ktrtal = 5	n	x\$(n)	y\$(n)	p\$(n)	BEMÆRKNING
ktr\$ = "k"	101	konst1	konst2	ft\$	LED1
	102	konst1	konst2	ft\$	LED2
	103	konst1	konst2	ft\$	LED3
	104	start	korrr	"k"	polygon har
	105	slut	fortegn	type	symbol

linie 21510 Brugerfejl: Type 3 / 5 og LED3 = 0: LED 3 >>> 1
 linie 21590 ff Her kontrolleres, om der er brugt den langsomme kurveprocedure til at definere en ret linie.



DATA					
101	0.3	2	pot		$y_t = 0.3 * (1 - x_t)^2 - 2$
102	2	'	kon		xmin = 2
103	0	'	kon		xmax = 5
104	0	1	k		ymin = -2
105	5	-1	1		ymax = 1

linie 22000 ff PROCedure "tegnkurve".

Kurven tegnes som en prikket linie med prikafstand bestemt i linie 22090. linie 22080 er nok gavnlige.

```

22200 DEFine PROCedure plotkurve
22210 LOCal type, trin, yt, xf, xt, gly: yt= g$(5)
22220 type= p$(105): trin= (g$(2) - g$(1)) / 55
22230 PRINT#5; 'W270 2270 430 2430': PRINT#5; 'F'; pen+1; 'L2 5'
22240 FOR xt= x$(104) TO x$(105) STEP trin
22250   xf= y$(104) + y$(105) * xt
22260   gly= yt: beregny: IF ABS(gly - yt) > 1E8: EXIT xt: REMark ?? **
22270   IF xt = x$(104) THEN
22280     PRINT#5; 'H'; INT(xt/skx-xnul); '/'; INT(yt/sky-ynul); 'K'
22290   ELSE PRINT#5; 'I'; INT(xt/skx-xnul); '/'; INT(yt/sky-ynul); 'K'
22300   END IF
22310 END FOR xt: PRINT#5; 'H; L0; W0 3380 0 2800'
22320 END DEFine plotkurve

22400 DEFine PROCedure beregny
22410 LOCal y(3), fk: DIM y(3)
22420 FOR n= 1 TO 3
22430   fk= CODE(p$(n+100))
22440   SElect ON fk
22450     = 107: y(n)= x$(n+100)
22460     = 108: y(n)= x$(n+100) * LOG10 ( xf * y$(n+100))
22470     = 112: y(n)= x$(n+100) * ( xf ^ y$(n+100) )
22480     = 120: y(n)= x$(n+100) * ( y$(n+100) ^ xf )
22490     = 115: y(n)= x$(n+100) * SIN ( xf * y$(n+100))
22500     = 116: y(n)= x$(n+100) * TAN ( xf * y$(n+100))
22510   END SElect
22520 END FOR n
22530 SElect ON type
22540   = 1: yt= y(1) + y(2) + y(3)
22550   = 2: yt= y(1) + y(2) * y(3)
22560   = 3: yt= y(1) + y(2) / y(3)
22570   = 4: yt= ( y(1) + y(2) ) * y(3)
22580   = 5: yt= ( y(1) + y(2) ) / y(3)
22590 END SElect
22600 END DEFine beregny

22800 DEFine PROCedure print_plottekst
22810   IF plot$(lin)(25)= ' ': PRINT#3
22820   IF plot$(lin)(25)= '$' THEN
22830     BPUT#3; 27, 80: q$=plot$(lin)(1 TO 18): printerstreng q$
22840   END IF
22850   IF plot$(lin)(25)= '#' THEN
22860     BPUT#3; 27, 77: q$=plot$(lin)(1 TO 24): printerstreng q$
22870   END IF
22880 END DEFine print_plottekst
22990 :
23000 DEFine PROCedure sqrtegn
23010 LOCal size, a, y, q, n, l
23030 TRA 0: size= INT( 8.3*csz ): a= 56+size: y= 80+size: q= 72+size
23040 FOR n= p + 2 TO LEN(su$)
23050   IF su$(n)= ' ': l= n - p - 2: EXIT n: ELSE l= LEN(su$) - p - 1
23060 END FOR n
23070 PRINT#5; 'H';:PRINT#5; 'T';:BPUT#5; a, 13:PRINT#5; 'I';:PRINT#5; 'T';
23090 BPUT#5; y, y, a, a, a, a, a, a, q, q: FOR n= 1 TO l: BPUT#5; q, q, q, q
23100 TRA 1: PRINT#5; CHR$(13);
23110 END DEFine sqrtegn

```

- linie 22200 PROCEDURE "plotkurve" (SERVOGOR 881)
- linie 22210 ff En række indledende manøvrer:
 yt = g\$(5) af hensyn til linie 22270, første gang.
 trin til kurvebergeningen, stipling L2 5.
 Vinduesdefinition 200 x 200 mm. 27,47 som 0-punkt.
 Pennummer, smlgn. linie 25430 ff.
- linie 22260 PROCEDURE "beregny" nedenfor.
 Endvidere en forsikring mod, at y går mod uendelig.
- linie 22280 f "H" er hævet, "I" er sænket pen. Uanset, at kurven egentlig startes uden for vinduet, finder plotteren ud af det. Fjernes 22280, startes kurven fra sidste position forud: afslutningen på y-enheds udskrift.
- linie 22310 Maksimalt vindue, fuld streg linie.
- linie 22400 PROCEDURE "beregny".
 Lige ud ad landevejen.
- linie 22800 PROCEDURE "print_plottekst"
- linie 22810 ff plot\$(lin)(25) kan være " " "#" "\$",
 for hhv en tom, en almindelig, en stor-skrift linie.
 BPUT#3; 27, 77 hhv 80 giver elite / pica skrift, der nogenlunde modsvarer printerens skriftstørrelser.
- linie 23000 PROCEDURE "sqrtegn".
- linie 23020 TRA 0 er nødvendig af hensyn til visse af plotterens kommandokarakterer.
 "size" er et sært udtryk, der imidlertid rammer plet for begrænsede variationer i skriftstørrelsen, som har en ulineær sammenhæng mellem normale og selvdefinerede tegn. Se IFH-rapport 180. DXY-plotteren er langt lettere at styre i denne henseende.
- linie 23040 Kvadratrodstegnets vandrette streg tegnes henover de følgende tegn, til der optræder et mellemrum / til strengens slutning.

```

24000 DEFine PROCedure plotteropen
24010 LOCal plot$(19,25),q$,navn$,gnlin$,lin,gllin,nylin,linloop
24020 LOCal lm,lp,pen,xnul,ynul,skx,sky,csz,su$,xsu,ysu
24030 LOCal save3$,ft(12),fl,z$,slcf: DIM ft(9): z$=' ': fl= 0
24040 lin= 0: skx= (g$(2) - g$(1)) / 2000: sky= (g$(5) - g$(4)) / 2000
24050 xnul= g$(1) / skx - 270: ynul= g$(4) / sky - 430

24070 BORDER#0;2,7:CLS#0: FOR n=0 TO 19:plot$(n)(1 TO 25)=FILL$( ' ',25)
24080 save3$= fldat$ & filnavn$ & '_plottekst': z= 0: CLS#4

24100 IF DATAD$ <> fldat$: fl= fldat$(4): DATA_USE fldat$
24110 FOR n=1 TO 9: IF FOP_IN(#3,save3$ & n)=0: ft(n)= n: CLOSE#3
24120 FOR n=1 TO 9: IF ft(n): z$= z$ & ft(n)
24130 IF LEN(z$) : BORDER#4;3: WSTAT#4: save3$: BORDER#4;0
24140 IF fl : DATA_USE flp1_: IF fl= 1: DATA_USE flp2_
24150 REPEAT lp
24160 PRINT#0; ' VÆLG PLOTTERTEKST: '\ NY TEKST : TAST 0'
24170 INK#0;4: PRINT#0; ' EXISTERENDE FIL, '\ SE OVENFOR: ';
24180 INK#0;7: PRINT#0; 'slutciffer'\ ANDEN FIL: TAST 10'
24190 talput 0,4,22,0,10,0,0: slcf= ttal: IF ttal= 0: EXIT lp
24200 IF tal$ INSTR z$ THEN
24210 OPEN_IN#6: save3$ & tal$
24220 FOR lin= 0 TO 19: GET#6; q$: plot$(lin)(1 TO 25)= q$
24230 CLOSE#6: EXIT lp
24240 END IF
24250 CLS#0: PRINT#0; '\ FILNAVN: '\ FLP' & FILL$( ' ',20)\
24260 svarput 0,2,4,'12',0,2,7: z= FOP_DIR(#3,'flp' & q$ & '_')
24270 IF z= 0: AT#0;2,5: PRINT#0; ' _': INPUT#0: navn$
24280 IF z= 0 AND FOP_IN(#3,'flp' & q$ & '_' & navn$)= 0 THEN
24290 z$= 'a': OPEN_IN#6; 'flp' & q$ & '_' & navn$
24300 FOR lin= 0 TO 19: GET#6; q$: plot$(lin)(1 TO 25)= q$
24310 CLOSE#3: CLOSE#6: EXIT lp
24320 END IF
24330 END REPEAT lp

24350 PAPER#4;2: INK#4;7: CLS#4: PRINT#4; '\ I DETTE FELT KAN TEKST'
24360 PRINT#4; ' TIL PLOTTER REDIGERES.': INK#4;4
24370 PRINT#4; '\ PÅ FIGUREN SKRIVES '\ KUN AKSEENHEDERNE.'
24380 IF hæld$ <> ' ' THEN
24390 PRINT#4; '\ FEJLUDJÆVNINGEN '\ SKRIVES IKKE.'
24400 PRINT#4; '\ PRINTERUDSKRIFT AF '\ UDJÆVNING: J / N'
24410 svarput 4,11,22,'jn',0,2,4: IF q$= 'j': udjævnukskrift
24420 END IF : INK#4;7
24430 AT#4;5,0: CLS#4,2: AT#4;7,2: PRINT#4; 'SKRIFTHØJDE, mm 4 6'
24440 PRINT#4; ' SKIFT MED : # $ '\ TEGN / LINIE : 24 18'
24450 PRINT#4; ' MAX. ANTAL : 17 3 '\ LINIER : 0 15'
24460 PRINT#4; '\ <CTRL> + <SHIFT> '\ + 2 >> 2. POTENS.'
24470 PRINT#4; ' + 3 >> 3. POTENS.' '\ + 0 >> GRADTEGN.'
24480 PRINT#4; ' + V >> KVADRATROD.': INK#4;4
24490 PRINT#4; ' * = PLOTTER-SYMBOL '\ FOR KONTROL-PUNKT.'
24500 INK#4;0: FOR n= 0 TO 19: AT#4; n,0: PRINT#4; n
24510 AT#4; 0,0: PRINT#4; '0..456789012345678901234': INK#4;7

24530 CLS#0: PRINT#0; '\ VÆLG LINIENUMMER '\ TAST 0 - 19'
24540 talput 0,2,20,0,19,0,0: lin= ttal
24550 CLS#4: FOR n=0 TO 19: AT#4; n, 10-(LEN(n) > 9) : PRINT#4; n;
24560 IF LEN(z$): FOR n=0 TO 19: IF plot$(n)(25) INSTR '#$': udskriv n
24570 REMark UDSKRIFT FORTSÆTTER PÅ NÆSTE SIDE

```

linie 24000 PROCedure "plotteropen".

linie 24040 f Der vælges et tegnefelt på 200 x 200 mm, med nulpunkt
i 27,43. (Mål angives i 1/10 mm). Smlgn. vinduet i
linie 26040 f.

linie 24100 ff Se forklaringen om DEFAULT til linie 16120.

linie 24120 og 24200. Eksisterende plottetekst-filers slutcifre sættes
i strengen z\$ og bruges som kontrol i linie 24200.

linie 24190 og 24230. REPEAT lp forlades såvel for ny fil (0) som
for eksisterende fil (slutciffer).
Kun slcf = 10 er tilbage: INPUT i linie 24250.

linie 24290 z\$ sættes lig "a" af hensyn til linie 24560.

linie 24550 f Linienumre og en eventuelt ønsket fil udskrives.

```

25000 REMark                plotteropen FORTSAT FRA FORRIGE SIDE
25010 REPEAT linloop
25020   CLS#0: BORDER#0;5: PRINT#0; \ ' FLYT CURSOR: ' ;: STRIP#0;7: INK#0;2
25030   AT#0; 1,14: PRINT#0; ' ← ': AT#0; 1,18: PRINT#0; ' → '
25040   PAPER#0;2: INK#0; 7: PRINT#0; ' RET VED AT SKRIVE'
25050   PRINT#0; ' OVENI DET GAMLE.' \ ' AFSLUT MED < ESC >.: ' : BORDER#0;2,7

25070   rediger lin: gllin= lin
25080   CLS#0: PRINT#0; '   NYT LINIENR.:   0 - 19'
25090   PRINT#0; '   FÆRDIG       : TAST 20': INK#0;4
25100   PRINT#0; '   ANDRING AF LINIENR, FX' \ ' LINIE XX >> YY :': INK#0;7
25110   PRINT#0; '   TAST   XX.YY': talput 0,4,18,0,20,0,0: lin= ttal
25120   IF lin= 20 THEN
25130     CLS#0: PRINT#0; \ '   OK ?   J / N': svarput 0,1,16,'jn',0,2,7
25140     IF q$== 'j': BORDER#0;0: EXIT linloop: ELSE lin= gllin
25150   END IF
25160   IF (lin-INT(lin)) THEN
25170     gnlin$= tal$: gllin= INT(lin): nylin= gnlin$( LEN(gllin)+2 TO)
25180     plot$(nylin)(1 TO 25)= plot$(gllin)(1 TO 25)
25190     plot$(gllin)(1 TO 25)= FILL$( ' ',25)
25200     udskriv gllin: udskriv nylin: lin= nylin
25210   END IF
25220 END REPEAT linloop
25230 CLS#0:   PRINT#0; ' TEKST-LAGRING UNDER'
25240 INK#0;4: PRINT#0; ' filnavn_plottekst* ' \ ' HVOR * ER ET NUMMER.'
25250 INK#0;7: PRINT#0; ' TAST nr ( ELLER 0 )' \ ' (EN EXIST.FIL SLETTER)'
25260 talput 0,3,22,0,9,0,0
25270 IF ttal THEN
25280   DELETE save3$ & tal$: OPEN_NEW#6; save3$ & tal$
25290   FOR lin= 0 TO 19: q$= plot$(lin)(1 TO 25): PUT#6; q$
25300   CLOSE#6: slcf= ttal
25310 END IF
25320 CLS#0: PRINT#0; \ \ '   TEKST-UDSKRIFT PÅ' \ \ '   PRINTER   J / N'
25330 svarput 0,3,18,'jn',0,2,7
25340 IF q$== 'j' THEN
25350   CLS#0: PRINT#0; \ '   PRINTER TÆNDT J / N': svarput 0,1,22,'j',0,2,7
25360   OPEN#3,ser1: BAUD baudtal: TRA trap: BPUT#3;27,108,10
25370   PRINT#3; save3$ & slcf: PRINT#3; \ \
25380   FOR lin= 0 TO 19: print_plottekst
25390   BPUT#3;27,80: BPUT#3;12: CLOSE#3
25400 END IF : INK#0;4: CLS#0

25420 PRINT#0; \ '   SLUK PLOTTEREN.'
25430 PRINT#0; \ '   FILTPENNE       f' \ '   TUSCHPENNE       t'
25440 pen= 1: svarput 0,3,20,'ft',0,2,4: CLS#0: IF q$== 'f': pen= 7
25450 PRINT#0; '   ANBRING PAPIR A3.' \ '   TÆND PLOTTEREN.' \ '   PEN ' ;
25460 PRINT#0; pen ; '   TIL RAMMEN.' \ '   PEN ' ; pen+1 ; '   TIL RESTEN.'
25470 PRINT#0; '   KLAR ?   J / N': svarput 0,4,20,'j',0,2,4

25490 OPEN#5,ser2: BAUD 1200: TRA 1: PRINT#5; CHR$(1); 'P'; CHR$(35); 4
25500 plotterakser: plottertekst: plotterpunkter: plotterregres

25520 PRINT#5; 'FOHO/OK': BORDER#0;0: CLS#0;0
25530 PRINT#0; \ '   ER PENNENE FJERNET': svarput 0,1,21,'j',0,2,4
25540 PRINT#5; CHR$(3): CLOSE#5: TRA trap: BAUD baudtal: INK#0;7
25550 END DEFINE plotteropen

```

linie 25000 "plotteropen" fortsat

linie 25070 PROCEDURE "rediger lin)", linie 26500. "lin" er det først valgte linienummer (24540).
"gllin" benyttes i linie 25130, hvor brugeren skal konfirmere, at han er færdig med at redigere linier. Svarer han "n", fortsættes i "REPEAT linloop" til linie 25070.

linie 25160 ff Her flyttes en tekst fra linie x til linie y:
I linie 25110 er der svaret "x.y".
gllin\$ = "x.y". gllin = x. nylin = y.
plot\$(nylin) får gllin's tekst.
plot\$(gllin) fyldes med ' '.
nylin og gllin udskrives, lin sættes lig nylin.
PROCEDURE "udskriv (lin)" i linie 27000.

linie 25230 ff Den færdige tekst kan eventuelt lagres. Ofte skal den bruges ved nye analyser af det igangværende DATA-sæt, eller revideres lidt til brug ved et nyt sæt DATA.

linie 25320 ff Teksten kan udskrives med PROCEDURE "print_plottekst" i linie 22800.

linie 25420 ff Start af plotter. Den skal slukkes, så den initialiseres og det bliver muligt at lægge papir i (der på SERVOGOR-plotteren fastholdes elektrostatisk). Denne plotter har - i modsætning til ROLAND DXY - 4 holdere, der kan holde tuschpenne "fugtige". Pennene efterlader farve. Holderne er således uegnede for farvepenne, især filt-penne.
På SERVOGOR bruges derfor penne nr. 1 og 2 til tusch og nr. 7 og 8 til filtpenne ("pen" hhv "pen + 1").

linie 25490 TRA skal være 1, af hensyn til Æ Ø Å. "CHR\$(1)" osv er PLOTTER OPEN og dansk tegnsæt.

linie 25500 henviser til PROCEDURER i linie 26000, 27500, 28500 og 29000. Se PLOTTER-PROCEDURER.

linie 25520 f betyder pen retur, hævet, til 0,0 (absolut x,y). Herfter skal pennene fjernes, så de ikke tørrer ud, efterfulgt af PLOTTER OFF og retablering af BAUD og TRA.

BEMÆRK, AT ET STOP MED < CTRL > + < SPACE > I EN PLOTTERPROCEDURE MASKE NOK KAN GÅ, HVIS DER GENSTARTES MED "RUN", MEN ABSOLUT IKKE VED BRUG AF "SMARTE" GENVEJE TIL PRINTEREN !! SE linie 120, 140.

```

26000 DEFine PROCedure plotterakser
26010  Local x1, xk, yk
26020  PRINT#5;'F'; pen ; 'H270/430K': REMark GROV PEN
26030  PRINT#5;'W270 2270 430 2430' : FOR n= 1 TO 2: PRINT#5;'V'
26040  IF g$(1) * g$(2) < 0 THEN
26050      plotterlinie 0,g$(4),0,g$(5): plotterlinie 0,g$(5),0,g$(4)
26060  END IF
26070  IF g$(4) * g$(5) < 0 THEN
26080      plotterlinie g$(1),0,g$(2),0: plotterlinie g$(2),0,g$(1),0
26090  END IF
26100  PRINT#5;'W0 3380 0 2800': PRINT#5; 'H270/430K'
26110  PRINT#5;'X2000 ' ; INT( 2000 / xantal ) ; ' 40 0'
26120  PRINT#5;'Y2000 ' ; INT( 2000 / yantal ) ; '-40 0'
26130  PRINT#5;'X2000 ' ; INT(-2000 / xantal ) ; '-40 0'
26140  PRINT#5;'Y2000 ' ; INT(-2000 / yantal ) ; ' 40 0'

26160  REMark PROGRAMMET ER MERGED IND OVENI DXY-PROGRAMMET

26180  PRINT#5;'F'; pen+1 : REMark FIN PEN, 4 mm SKRIFT
26190  x1= INT( g$(2) / skx -xnul +6 + 20 * LEN(g$(2)) - 40 * LEN(xe$) )
26200  plotterstreng xe$, 1.111111, x1, 260: PRINT#5;'Z40 0 40'
26210  FOR n= xantal TO 0 STEP -1
26220      xk= g$(1)+g$(3)*n: tal$= xk: IF ABS(xk) < 1 : ciffertal xk, 1
26230          IF ABS(g$(3)) < 1: IF xk= xk DIV 1: tal$= tal$ & '.0'
26240      x1= INT( xk / skx - xnul + 6 - 20 * LEN(tal$) )
26250      PRINT#5; 'H' ; x1; '/ 350 K'
26260      PRINT#5; 'B' ; tal$ ; CHR$(13)
26270  END FOR n
26280  FOR n= 0 TO yantal
26290      yk= g$(4)+g$(6)*n: tal$= yk: IF ABS(yk) < 1 : ciffertal yk, 1
26300          IF ABS(g$(6)) < 1: IF yk= yk DIV 1: tal$= tal$ & '.0'
26310      x1= INT( 240 - LEN(tal$) * 40 + .5 )
26320      PRINT#5; 'H' ; x1 ; '/' ; INT( yk / sky -ynul ) ; ' K'
26330      PRINT#5; 'B' ; tal$ ; CHR$(13)
26340  END FOR n
26350  plotterstreng ye$,1.111111,240 -LEN(tal$)*40, g$(5)/sky -ynul +90
26360  IF ktr$== 'p' THEN
26370      FOR p= 101 TO 100 + ktrtal
26380          x1= INT( x$(p) / skx - xnul + 7.5 )
26390          PRINT#5; 'H' ; x1 ; '/' ; INT( y$(p) / sky - ynul + .5 ) ; ' K'
26400          PRINT#5; 'L0': PRINT#5; 'IO 7': PRINT#5; 'L2 5'
26410          IF p < 100 + ktrtal: plotterlinie x$(p),y$(p),x$(p+1),y$(p+1)
26420      END FOR p: PRINT#5;'L0'
26430  END IF
26440  IF ktr$== 'k': plotkurve
26450  END DEFine plotterakser

```

- linie 26000 PROCEDURE "plotterakser".
- linie 26020 ff Kommandokarakter er bl.a:
 H = hævet I = sænket pen
 J = relative K = absolutte koordinater
 ROLAND DXY-plotteren benytter tilsvarende kommandoer, men de forbliver i kraft.
 SERVOGOR kræver - i hvert fald - karakteren J eller K efter hvert sæt koordinater, som en del af plottekomdoen.
 Koordinater skilles med / (På DXY med komma).
- F1 = pen nr.1 (1...8). På DXY har jeg renummeret med små etiketter til nr. 0...7, svarende til QL'ens farveskala, men internt bruger DXY også nr. 1...8.
- linie 26030 Wxmin xmax ymin ymax (een streng) definerer vinduet som udtegnes med V (to gange, hvis pennen er tør)
- linie 26050 PROCEDURE "plotterlinie (x1,y1,x2,y2)", linie 29800, er en stor hjælp.
- linie 26100 Maksimalt tegnefelt. Gå med hævet pen til 270/430 i absolutte koordinater (= tegnefeltets nedre, venstre hjørne.
- linie 26110 f Akseinddelingsmærker langs x- og y-akserne, indad, se brugsanvisningen for fortegn mv.
- linie 26160 gør kun opmærksom på, at de to programmer er ens, når bortses fra plotter-procedurerne, der blev udarbejdet separat og MERGED ind i hovedprogrammet.
 OPRINDELIG BRUGTE DE TO PROGRAMMER EKSAKT DE SAMME LINIENUMRE, MEN DER ER SKET SMA JUSTERINGER, DER IDAG GØR EN MERGE MULIG, HVIS DER CHECKES. DLINE HELLERE FØRST, HVIS PROCEDURER / PROGRAMMER SKAL JUSTERES.
- linie 26180 ff Nu bruges PROCEDUREN "plotterstreng" m. (4 parametre) linie 28000, til at plotte de fire enheder / navne i hjørnerne af tegnefeltet, korrigeret med LEN til at "flugte" med akseinddelingernes tal.
 Endvidere udskrives de to aksers akseinddelingstal, med speciel hensyntagen til ordentlig udskrift af små tal (linie 26220 f og PROCEDURE "cifertal", linie 15200).
- linie 26200 Z40 0 40 = skriftstørrelse 4 x 4 mm, hældning 0.
- linie 26360 ff Her udplottes en eventuel kontrolpolygon.
 L0 = fuld linie. L2 5 = en af de mulige stiplinger.
 I0 7 = cirkel med radius 0.7 mm.
- linie 26440 PROCEDURE "plotkurve", linie 22200.

```

26500 DEFine PROCedure rediger ( lin )
26510  LOCAL a$, b$, tab, codeb, skrivloop, w: tab= 0
26520  a$= plot$(lin)(1 TO 24) & FILL$(' ',11)
26530  IF plot$(lin)(25)= '$' THEN
26540      CSIZE#4;1,0: INK#4;4: lm= 18
26550  ELSE : CSIZE#4;0,0: INK#4;7: lm= 24: plot$(lin)(25)= '#'
26560  END IF
26570  AT#4; lin, tab: CURSEN#4
26580  REPEAT skrivloop
26590      AT#4; lin, tab: b$= INKEY$(#4,-1): codeb= CODE(b$)
26600      SElect ON codeb
26610      = 27: EXIT skrivloop
26620      = 37 TO 127, 130,134,138,162,166,170,175,184,186
26630      REMark INCL.Æ,Ø,Å,"qltegn"s tegn for expo og grad, EXCL.33-35
26640      a$(tab+1)= b$: AT#4; lin, tab: PRINT#4; a$(tab+1);
26650      tab= tab + (tab < lm-1)
26660      = 32
26670      FOR n= 34 TO tab + 1 STEP -1: a$(n+1)= a$(n)
26680      a$(tab+1)= ' '
26690      AT#4; lin,tab: FOR n= tab + 1 TO lm: PRINT#4; a$(n);
26700      tab= tab + (tab < lm-1)
26710      = 35,36
26720      IF codeb= 35: plot$(lin)(25)= '#': CSIZE#4;0,0: INK#4;7: lm=24
26730      IF codeb= 36 THEN
26740          w= 0: FOR n= 19 TO 24: IF a$(n) <> ' ': w= 1
26750          IF w THEN
26760              AT#4; lin,0: PRINT#4; ' STRENGEN ER FOR LANG ! ' ;
26770              PAUSE 50: plot$(lin)(25)= '#': CSIZE#4;0,0: INK#4;7: lm=24
26780              ELSE
26790                  plot$(lin)(25)= '$': CSIZE#4;1,0: INK#4;4: lm=18
26800              END IF
26810          END IF
26820          AT#4;lin,0: PRINT#4; a$(1 TO lm);
26830          = 194, 202
26840          w = tab + (codeb= 202): IF w = 0: NEXT skrivloop
26850          FOR n= w TO 34: a$(n)= a$(n+1)
26860          a$(35)= ' ': AT#4; lin, w -1: FOR n= w TO lm: PRINT#4; a$(n);
26870          IF codeb= 194: tab= tab - (tab > 0)
26880          = 192: tab= tab - (tab > 0)
26890          = 200: tab= tab + (tab < lm-1)
26900          END SElect
26910      END REPEAT skrivloop
26920      CURDIS#4: plot$(lin)(1 TO 24)= a$(1 TO 24): CSIZE#4;0,0: INK#4;7
26930  END DEFine rediger

27000 DEFine PROCedure udskriv (lin)
27010  IF plot$(lin)(25)= '#': CSIZE#4;0,0: INK#4;7: lm= 24
27020  IF plot$(lin)(25)= '$': CSIZE#4;1,0: INK#4;4: lm= 18
27030  AT#4;lin,0: PRINT#4; plot$(lin)(1 TO lm);
27040  END DEFine udskriv

```

- linie 26500 PROCEDURE "rediger (lin)".
 En forenklet udgave af et tekstbehandlingsanlæg, som benytter "overskriv"-metoden.
 Hver linie behandles for sig.
- a\$ er en streng med 35 tomme pladser (' ').
 Heraf vises kun de første 24, og det er kun disse, der senere indgår i den endelige plot\$(lin).
- Det er således muligt, under redigeringen, at visse dele af linien skubbes ud til højre, når der indskydes et ord midt på linien. Det overskydende er ikke synligt, men heller ikke tabt, og vil dukke op igen, hvis den synlige del af linien afkortes.
- linie 26520 a\$ fyldes op med de første 24 karakterer fra plot\$, dvs enten ' ' eller en eksisterende tekst.
- linie 26530 f Eksisterer linien, er plot\$(lin)(25) = '\$' / '#', og farve, CSIZE samt karakterantal / linie (lm = 18 / 24) fastlægges. Hvor intet er fastlagt, bruges '#'.
 26570 og 26910. Se TOOLKIT om CURSEN / CURDIS. Stoppes midt i tekstbehandlingen, er man ikke fortabt: <CTRL> + "c" retablerer den normale CURSOR.
- linie 26600 ff codeb =
 27 er ESC, liniens redigering afsluttet.
 32 er SPACE, mellemrum (' ').
 35, 36 er '#' hhv '\$'.
 for 36 må 'lm' kontrolleres, evt. omstilles til 35.
 192, 200 er venstre / højre pilen.
 194, 202 er <CTRL> + venstre / pilen.
 linie 26620 er de tilladte tegn, se REMark 26630.
- tab er 0....23, n er 1...24 i plot\$(lin)(n).
 Der er lidt bogholderi i disse tal, og det hele bør ved lejlighed forbedres, så de "flyvende" udskrivinger, når fx SPACE benyttes, undgås. Det forstås bedst ved et forsøg, hvad jeg mener.
- linie 26910 Linien er færdig. plot\$(lin)(1 to 24) er klar.
- linie 27000 PROCEDURE "udskriv (lin)".
 Hvis den aktuelle linie er tom, skrives ingenting, linie 27030.

```

27500 DEFine PROCedure plottertekst
27510  LOCAL ned, venst, glcs, st, sl, p
27520  PRINT#5; 'Z36 0 36': REMark 3.6 / 4.8 mm SKRIFT (csz)
27530  ned= g$(5)/sky -ynul +50: venst = g$(2)/skx -xnul +170: glcs= 1
27540  FOR lin= 0 TO 19
27550    st= 0: sl= 0: csz= 1: IF plot$(lin)(25)= '$': csz= 4/3
27560    FOR p= 1 TO 24: IF plot$(lin)(p) <> ' ': st= p: EXIT p
27570    IF st= 0: ned= ned - 82: IF lin= 19: RETURN: ELSE NEXT lin
27580    FOR p= 24 TO 1 STEP -1: IF plot$(lin)(p) <> ' ': sl= p: EXIT p
27590    ned= ned - 108 + 27 * (glcs = 1) * ( csz = 1)
27600    xsu= venst + (st - 1) * 36 * csz
27610    REMark csz=1 ELLER 4/3: 2.8 mm EXTRA FØR / EFTER 4.8 mm SKRIFT
27620    REMark plotterstreng BRUGER TO PEN-BREDDER (NR. 1 OG 2)
27630    plotterstreng plot$(lin)(st TO sl), csz, xsu, ned: glcs= csz
27640  END FOR lin: PRINT#5; 'F'; pen+1: PRINT#5; 'Z36 0 36'
27650  END DEFine plottertekst

```

```

28000 DEFine PROCedure plotterstreng ( su$, csz, xsu, ysu )
28010  LOCAL b$, p, op, stp, c%, codes: stp= pen + 1 - (csz >1.2)
28020  FOR p= 0 TO LEN(su$) - 1
28030    codes= CODE(su$(p+1))
28040    SELECT ON codes
28050      = 42: b$= 'o': op= 12 * csz: c%= 18 * csz
28060      = 137: b$= 'O': op= 18 * csz: c%= 27 * csz
28070      = 131: b$= '3': op= 18 * csz: c%= 27 * csz
28080      = 160: b$= '2': op= 18 * csz: c%= 27 * csz
28090      = 182: op= 0: c%= csz
28100      = REMAINDER: b$= su$(p+1): op= 0: c%= 36 * csz
28110    END SELECT
28120    PRINT#5; 'F'; stp; 'Z'; c%; '0'; c%;: c%= 36 * csz
28130    PRINT#5; 'H'; INT(xsu + p * c% +.5); '/'; INT(ysu + op + .5); 'K'
28140    IF codes= 182: sqrtegn: IF p= LEN(su$) - 1: RETURN: ELSE NEXT p
28150    PRINT#5; 'B'; b$; CHR$(13)
28160  END FOR p
28170  END DEFine plotterstreng

```

```

28500 DEFine PROCedure plotterpunkter
28510  PRINT#5; 'F'; pen+1: PRINT#5; 'Z30 0 30': REMark 3 mm SKRIFT
28520  FOR p= startnr TO slutnr
28530    IF hald$ <> '' AND LEN(valg$)= 1 AND p$(p) <> rg$: NEXT p
28540    x1= INT( x$(p) / skx - xnul + 12 * (p$(p)= 'o') + .5 )
28550    PRINT#5; 'H'; x1; '/'; INT( y$(p) / sky - ynul ); 'K'
28560    IF p$(p)= 'o': PRINT#5; 'IO 12';: ELSE PRINT#5; 'M2'
28570  END FOR p
28580  IF valg$= 'o+x' THEN
28590    PRINT#5; 'H'; INT( g$(2)/skx -xnul +180 ); '/'; 445; 'K'
28600    PRINT#5; 'M2'
28610    PRINT#5; 'H'; INT( g$(2)/skx -xnul +192 ); '/'; 505; 'K'
28620    PRINT#5; 'IO 12'
28630    IF ktr$ INSTR 'pk' THEN
28640      PRINT#5; 'H'; INT( g$(2)/skx -xnul +187 ); '/'; 565; 'K'
28650      PRINT#5; 'IO 7'
28660    END IF
28670  END IF
28680  END DEFine plotterpunkter

```

linie 27500 PROCEDURE "plottertekst".

linie 27550 Styling af skriftstørrelse.

linie 27560 Find første karakter i linien.

linie 27570 st = 0 : linien er tom, videre.

linie 27580 Find sidste karakter.

linie 27630 PROCEDURE "plotterstreng" (4 parametre), nedenfor.

linie 27640 Uanset hvad "plotterstreng" brugte i sidste linie, stilles pen og skriftstørrelse til det normale.

linie 28000 PROCEDURE "plotterstreng" med 4 parametre.
su\$ er teksten, csz er skriftstørrelsen.
xsu og ysu er koordinaterne (absolutte).

linie 28040 ff codes bruges til at skalere og / eller flytte visse karakterer opad.
42 er '*', der dels er plottersymbol for en eventuel kontrolpolygons knæpunkter, dels fungerer som multiplikationstegn.
137, 131, 160 er <CTRL> + '0' / '3' / '2', dvs
gradtegn / tredje / anden potens.
182 er <CTRL> + 'v', dvs kvadratrodstegnet.

linie 28120 ff Pen og skriftstørrelse bestemmes.
Hævet pen til korrekt position.
Hvis codes = 182: PROCEDURE "sqrtegn", linie 23000.
Ellers skrives den normale karakter.
Dette foregår som sagt enklere på en DXY-plotter.

linie 28500 PROCEDURE "plotterpunkter".

linie 28530 Hvis der er foretaget regressionsanalyse og der kun er krævet et ud af to symboler udtegnet ...

linie 28540 Udgangspunktet for x = en cirkel med radius 1.2 mm (IO 12), er ikke det samme som for x = symbol M2.

linie 28580 Når der er to regressionslinier, skal der udtegnes symboler forneden i tekstfeltet, med de tilhørende linieprøver, se linie 29060.

```

29000 DEFine PROCedure plotterregres
29010   LOCal x1, x2, y1
29020   IF hæld$= '': RETURN
29030   PRINT#5; 'F'; pen: PRINT#5; 'Z40 0 40': k= 0: IF rg$== 'x': k= 20
29040   IF LEN(valg$) < 3: FOR n= 1 TO 2: plotterregres2

29060   IF valg$== 'o+x' THEN
29070     rg$= 'o': k= 0: FOR n=1 TO 2: plotterregres2
29080     x1= g$(2) + 250 * skx: x2= x1 + 350 * skx
29090     y1= (510 + ynul) * sky: plotterlinie x1, y1, x2, y1

29110     IF ktr$ INSTR 'kp' THEN
29120       PRINT#5; 'L2 5'
29130       y1= (570 + ynul) * sky: plotterlinie x1, y1, x2, y1
29140     END IF

29160     rg$= 'x': k= 20: PRINT#5; 'L2 10'
29170     y1= (450 + ynul) * sky: plotterlinie x1, y1, x2, y1
29180     plotterregres2 : PRINT#5; 'L0': REMark RETUR TIL FULD LINIE
29190   END IF : REMark x PUNKTERET. FULD LINIE TEGNES 2 x ( 29040,70 )
29200 END DEFine plotterregres

29500 DEFine PROCedure plotterregres2
29510   REMark fjl (1)=gnx (2)=gny (10)=yvenst (11)=yhøjre (12)=xnedad
29520   REMark      (13)=xopad  ( + k= 0 for o / 20 for x )

29540   IF hæld$== 'l': plotterlinie fjl(1+k), g$(4), fjl(1+k), g$(5): RETURN
29550   IF hæld$== 'v': plotterlinie g$(1), fjl(2+k), g$(2), fjl(2+k): RETURN

29570   IF (fjl(10+k)-g$(4)) * (g$(5)-fjl(10+k)) > 0 THEN
29580     IF (fjl(11+k)-g$(4)) * (g$(5)-fjl(11+k)) > 0 THEN
29590       plotterlinie g$(1), fjl(10+k), g$(2), fjl(11+k): RETURN
29600     END IF
29610     IF fjl(5+k) > 0: plotterlinie g$(1), fjl(10+k), fjl(13+k), g$(5)
29620     IF fjl(5+k) < 0: plotterlinie g$(1), fjl(10+k), fjl(12+k), g$(4)
29630     RETURN
29640   END IF

29660   IF (fjl(11+k)-g$(4)) * (g$(5)-fjl(11+k)) > 0 THEN
29670     IF fjl(5+k) > 0: plotterlinie fjl(12+k), g$(4), g$(2), fjl(11+k)
29680     IF fjl(5+k) < 0: plotterlinie fjl(13+k), g$(5), g$(2), fjl(11+k)
29690     RETURN
29700   END IF

29720   plotterlinie fjl(12+k), g$(4), fjl(13+k), g$(5)/1
29730 END DEFine plotterregres2

29800 DEFine PROCedure plotterlinie ( x1, y1, x2, y2 )
29810   LOCal q1, q2, w1, w2
29820   q1 = INT(x1 / skx - xnul + .5): q2 = INT(x2 / skx - xnul + .5)
29830   w1 = INT(y1 / sky - ynul + .5): w2 = INT(y2 / sky - ynul + .5)
29840   PRINT#5; 'H' ; q1 ; '/' ; w1 ; 'KI' ; q2 ; '/' ; w2 ; 'K'
29850 END DEFine plotterlinie

```

linie 29000 PROCEDURE "plotterregres".

linie 29020 Nødvendig sikring mod fejltastninger.

linie 29030 Pen, skriftstørrelse og k til fjl(40) fastlægges.

linie 29040 PROCEDURE "plotterregres2", nedenfor, tegner linien.
Hvis der kun er een regressionslinie, bruges fuld
linie, som tegnes to gange, så der ikke er "huller".

linie 29060 ff Hvis der er to linier, tegnes o-linien to gange med
med fuld linie, x-linien med stiplet linie (L2 5).
Herudover laves linieprøver ud for symbolerne, under
tekstfeltet, o i linie 29080 f, x i linie 29170, og
en eventuel kontrollinie i linie 29110 ff.

linie 29500 PROCEDURE "plotterregres2".
Smlgn. PROCEDUREN "udtegning", linie 13000.

linie 29800 PROCEDURE "plotterlinie" (to koordinatsæt).
Den gør det nemmere at definere en linie på "normal"
måde.

```

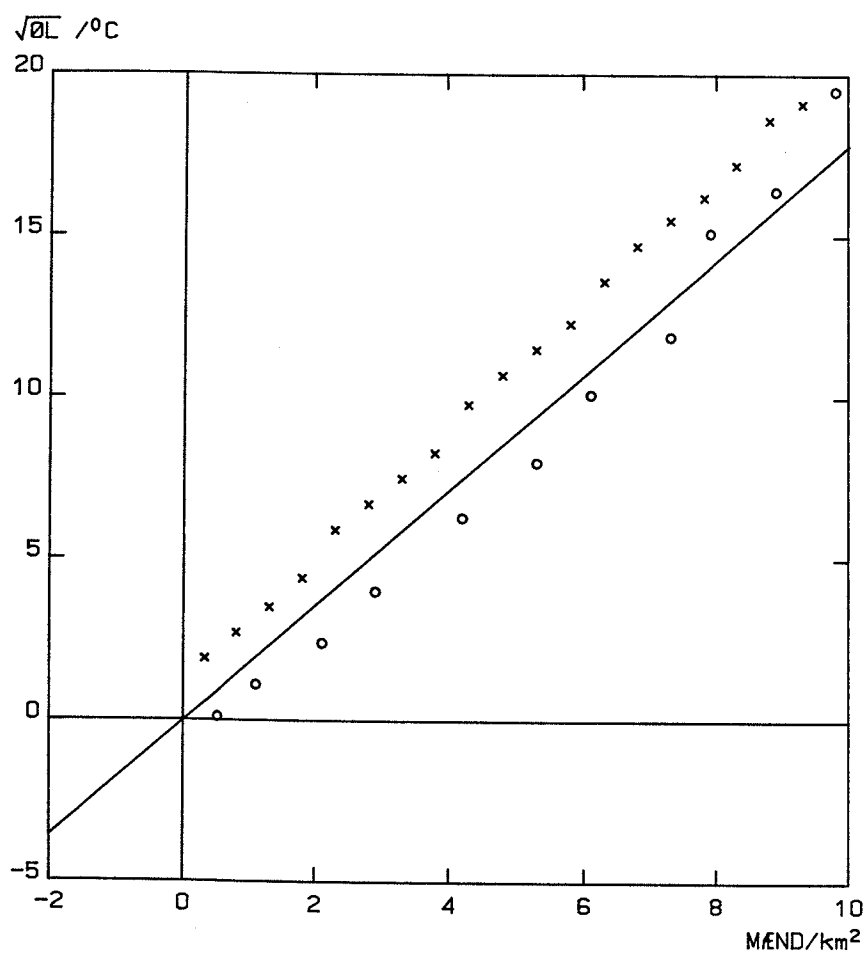
30000 DEFine PROCedure udjævn_vindue
30010 OPEN#4; con_: OPEN#8; con_: FOR n=0,1,2,4,8: CHAR_USE#n; dansk0, dansk1
30020 WINDOW#0; 512,256, 0, 0: PAPER#0; 0: CLS#0
30030 WINDOW#0; 154, 54, 358, 202: PAPER#0; 2: INK#0; 7: BORDER#0; 0
30040 WINDOW#1; 358, 256, 0, 0: PAPER#1; 0: INK#1; 7: BORDER#1; 1, 2
30050 WINDOW#2; 436, 202, 0, 0: PAPER#2; 7: INK#2; 0: BORDER#2; 0
30060 WINDOW#4; 154, 202, 358, 0: PAPER#4; 2: INK#4; 7: BORDER#4; 0
30070 WINDOW#8; 42, 60, 460, 120: PAPER#8; 2: INK#8; 7: BORDER#8; 0
30080 END DEFine udjævn_vindue
30200 DEFine PROCedure redaktion
30210 WINDOW#0; 512, 256, 0, 0: PAPER#0; 0: CLS#0
30220 WINDOW#0; 436, 54, 0, 202: PAPER#0; 0: INK#0; 4: BORDER#0; 0
30230 WINDOW#1; 512, 202, 0, 0: PAPER#1; 2: INK#1; 7: BORDER#1; 0
30240 WINDOW#2; 436, 202, 0, 0: PAPER#2; 7: INK#2; 0: BORDER#2; 0
30250 CLS#0: CLS#2: FOR n=0 TO 2: CHAR_USE#n; dansk0, dansk1: CSIZE#n; 0, 0
30260 END DEFine redaktion
30300 DEFine PROCedure dobb_vindue
30310 OPEN#4; con_: FOR n= 0,1,2,4: CHAR_USE#n; dansk0, dansk1
30320 WINDOW#0; 512, 256, 0, 0: PAPER#0; 0: CLS#0
30330 WINDOW#0; 436, 33, 0, 223: PAPER#0; 2: INK#0; 7: BORDER#0; 1, 0
30340 WINDOW#1; 280, 223, 0, 0: PAPER#1; 2: INK#1; 7: BORDER#1; 1, 0
30350 WINDOW#2; 232, 223, 280, 0: PAPER#2; 2: INK#2; 7: BORDER#2; 1, 0
30360 WINDOW#4; 76, 33, 436, 223: PAPER#4; 2: INK#4; 7: BORDER#4; 1, 0
30370 END DEFine dobb_vindue
30400 DEFine PROCedure c80dir (pdir$)
30410 CLS#0: PRINT#0; \ ' PRINTER TÆNDT J / N': svarput 0,1,22, 'j', 0,2,7
30420 OPEN#3, ser1: BAUD baudtal: TRA trap: BPUT#3; 27,108,5: PRINT#3; \ \
30430 IF LEN(pdir$)= 4: pdir$= pdir$ & ' _'
30440 DIR#3; pdir$: BPUT#3; 12: CLOSE#3
30450 END DEFine c80dir
30500 DEFine PROCedure c80stat (pstat$)
30510 CLS#0: PRINT#0; \ ' PRINTER TÆNDT J / N': svarput 0,1,22, 'j', 0,2,7
30520 OPEN#3, ser1: BAUD baudtal: TRA trap: BPUT#3; 27,108,5: PRINT#3; \ \
30530 IF LEN(pstat$)= 4: pstat$= pstat$ & ' _'
30540 STAT#3; pstat$: PRINT#3; \ : WSTAT#3; pstat$: BPUT#3; 12: CLOSE#3
30550 END DEFine c80stat
30600 DEFine PROCedure c80list (linie1, linie2)
30610 IF linie1 < 0: linie1= - linie1: GO TO 30630
30620 CLS#0: PRINT#0; \ ' PRINTER TÆNDT J / N': svarput 0,1,22, 'j', 0,2,7
30630 OPEN#3, ser1: BAUD baudtal: TRA trap
30640 BPUT#3; 27,108,5,27,81,77,27,67,72,27,78,10
30650 LIST#3, linie1 TO linie2: REMark xx,yy eller xx TO yy
30660 BPUT#3; 12: CLOSE#3
30670 END DEFine c80list
30700 DEFine PROCedure c80program
30710 LOCAL begynd, slut
30720 CLS#0: PRINT#0; \ ' PRINTER TÆNDT J / N': svarput 0,1,22, 'j', 0,2,7
30730 begynd= 1: RESTORE 30780: REMark TRA og BAUD i c80list
30740 FOR n= 1 TO 35
30750 READ slut: IF slut= 0: EXIT n
30760 c80list - begynd TO slut: begynd= slut
30770 END FOR n: OPEN#3, ser1: PRINT#0; \ ' SLUT': BPUT#3; 12: CLOSE#3
30780 DATA 990,1990,2990,3990,4990,5990,7990,9990,10990,11690,12490
30790 DATA 13290,13990,14990,15390,15990,16590,17990,18790,19490
30800 DATA 19990,21290,22190,23990,24990,25990,26490,27490,28990
30810 DATA 29990,30990,31990, 0
30820 END DEFine c80program

```

linie 30000 ff ANVENDTE VINDUER OG HJÆLPEPROCEDURER BESKRIVES IKKE.
SE QL's BRUGSANVISNING OG IFH-rapport nr. 177.

linie 31000 ff Gengives kun for de første par linier.
STOP i linie 31000
GO TO xxx: REMark procedurenavn i de følgende linier
sørger for at en procedure-linieoversigt opdateres
løbende under et programs opbygning.

31000 STOP: REMark HOLD STYR PÅ PROCEDURE - NUMRE
31010 GO TO 100: REMark nyt program, DIM, variabler.
31020 GO TO 600: REMark hovedloop
31030 GO TO 900: REMark datasætvalg
31040 GO TO 1000: REMark akseinput
31050 GO TO 2000: REMark start_slut



ØLFORBRUG PR.
FAMILIEOVERHOVED
I TYNDBEFOLKEDE OMRÅDER

DEN GENNEMSNITLIGE
FAMILIESTØRRELSE FOR
OMRÅDERNE: 6.3 PERSONER

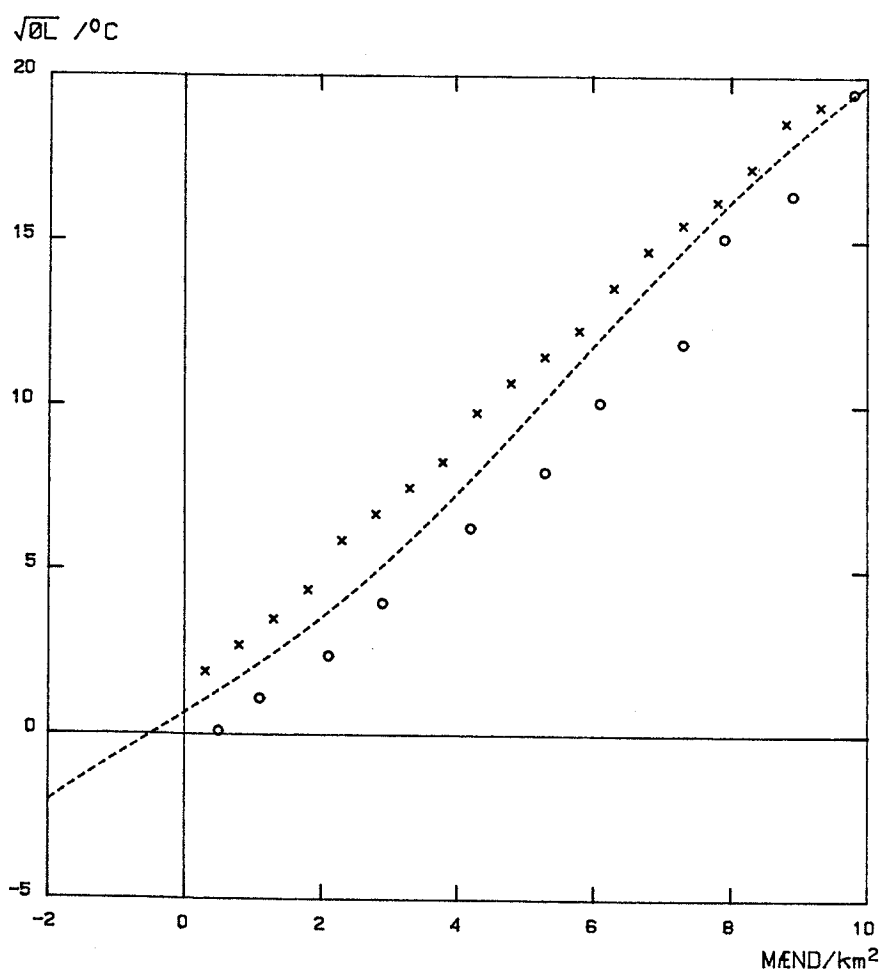
LINIE GENNEM PUNKT 0,0
FOR GRUPPE o :
HÆLDNING 1.78

GRUPPE x ER VIST,
UDEN AT INDGÅ I ANALYSEN

KORRELATIONSKOEFFICIENT
0.98

o GRØNLAND 1985
x SAHARA 1977

PROGRAM "REGRESSION"
 ROLAND DXY 880 PLOTTER



ØLFORBRUG PR.
 FAMILIEOVERHOVED
 I TYNDBEFOLKEDE OMRÅDER

DEN GENNEMSITLIGE
 FAMILIESTØRRELSE FOR
 OMRÅDERNE: 6.3 PERSONER

KURVE INDLAGT:

$$y(x) = 1.8 \cdot (x + 0.5) + \sin(-x/2)$$

○ GRØNLAND 1985
 × SAHARA 1977

```

100 REMark *** regresboot til regression ***
110 flp$= 'FLP1_': DATA_USE flp1_: DEST_USE flp1_: PROG_USE flp1_
120 PRINT#2; '\' TAST PRINTERTYPE: ' \ ' QL-PRINTER      q'
130 PRINT#2; '   CPA80-PRINTER      c'
140 REPEAT p
150   INK#2;5: OVER#2;-1: AT#2;3,22: PRINT#2; '?': pr$= INKEY$(12)
160   IF LEN(pr$)=1 AND pr$ INSTR 'qc': IF INKEY$(20) <> pr$: EXIT p
170 END REPEAT p: OVER#2;0: INK#2;2: AT#2;3,22: PRINT#2; pr$
180 PRINT#2; '\' PLOTTERTYPE: ROLAND DXY 880': TK2_EXT
190 dansk0= RESPR (5224): LBYTES qltegn0, dansk0
200 dansk1= dansk0 + 876: LBYTES qltegn1, dansk1
210 FOR n=0 TO 2: CHAR_USE#n; dansk0, dansk1
220 PRINT#2; '\' qltegn      LOADED  0 0 0 0 2 3 4'
230 STRIP 7: INK 2: AT 1,1: PRINT'FOR INPUT I PROGRAM regression GÆLDER: '
240 AT 3, 1: PRINT'TAL OG STRENGE INDTASTES VED CURSOR. '
250 AT 11,1: PRINT'INPUT I FORM AF EEN KARAKTER / EET TAL '
260 PAPER 2: INK 7: AT 5,2: PRINT 'DE KAN REVIDERES FØR < ENTER >.'
270 PRINT '   I EN DEL TILFÆLDE VIL EXISTERENDE DATA'
280 PRINT '   BLIVE UDSKREVET. DISSE KAN ACCEPTERES'
290 PRINT '   VED TASTNING AF < ENTER > ALENE.'
300 PRINT '   REVISIONER MÅ INDTASTES FULDT UD.'
310 PRINT'\'\ '   fx SVAR PÅ ET SPØRGSMÅL / KOMMANDO'
320 PRINT '   SKAL INDTASTES UDEN BRUG AF ENTER.'
330 PRINT'\'\ '   OM MULIGT CHECKES INPUT FOR FEJL.'
340 IF pr$=='c': KOPI= dansk0 + 1464: LBYTES scopy_bin, KOPI: CALL KOPI
350 IF pr$=='q': KOPI= dansk0 + 1464: LBYTES qcopy_bin, KOPI: CALL KOPI
360 IF pr$=='c': PRINT#2; '   scopy_bin LOADED': BAUD 4800: TRA 1
370 IF pr$=='q': PRINT#2; '   qcopy_bin LOADED'
380   ramf= dansk0 + 2900: LBYTES ram_Ver1, ramf: CALL ramf
390 PRINT#2; '   RAM_ver1 LOADED'
400   prik= dansk0 + 4680: LBYTES nyptpatch, prik: CALL prik
410 PRINT#2; '   pointpatch LOADED'
420 STRIP#2;2: INK#2;7: INK 2: FORMAT ram2_2: REMark USYNLIG FORMAT
430 OPEN_NEW#5; ram2_: PUT#5; 'ny', pr$, dansk0, dansk1: CLOSE#5
440 AT#2;17,2: PRINT#2; ' TAST < ENTER > FOR AT FORTSÆTTE ': INK 5: INK#2;5
450 n= 0: REPEAT loop
460   OVER-1: OVER#2;-1: AT 13,38: PRINT'?': AT#2;17,8: PRINT#2; '< ENTER >'
470   OVER 0: n= n + 1: BLOCK 7,8,227,51,2 + (n MOD 2) * 5
480   AT#2;17,36: b$= INKEY$(12): IF b$<>'': EXIT loop
490 END REPEAT loop: OVER 0: INK 7: OVER#2;0: INK#2;2: PAPER#2;7
500 IF pr$=='c' THEN
510   CLS#2: PRINT#2; '\'   ER PRINTEREN TÆNDT J / N': INK#2;5: OVER#2;-1
520   REPEAT 12
530     AT#2;1,30: PRINT#2; '?': b$=INKEY$(12): IF b$<>' ' AND b$=='j': EXIT 12
540   END REPEAT 12: OVER#2;0: INK#2;2: AT#2;1,30: PRINT#2; b$
550   EW FLP1_cp80græsk_bin
560   AT#2;5,2: PRINT#2; 'QL-DATAMATEN & CPA80-PRINTEREN'
570   PRINT#2; '   HAR NU KARAKTERSÆTTET "qltegn"'
580   INK#2;0: PRINT#2; '\'   SLUK ALDRIG FOR PRINTEREN !': INK#2;2
590   PRINT#2; '\'   SKER DET ALLIGEVEL VED EN FEJLTAGELSE'
600   PRINT#2; '   KAN "qltegn" GEN-LOADES MED <ALT> + q'
610   PRINT#2; '\'   BOGSTAVER PÅ SKÆRM OG PRINTER ER ENS,'
620   PRINT#2; '   MEN PLOTTEREN ACCEPTERER KUN 0 2 3 4'
630   AT#2;17,2: PRINT#2; ' TAST < ENTER > FOR AT FORTSÆTTE ': INK#2;5
640   REPEAT 13
650     OVER#2;-1: AT#2;17,8: PRINT#2; '< ENTER >'
660     b$= INKEY$(12): IF b$<>'': EXIT 13
670   END REPEAT 13: OVER#2;0: INK#2;2: INK 7
680 END IF
690 LRUN regresboot2

```

```
100 LBYTES regres_scr,131072: MRUN regression
110 REMark DETTE ER regresboot2
```

linie 100 boot'en til "regression" er "regresboot".
Tast "LRUN flp1_regresboot".

Bortset fra linie 100 og linie 690 er denne boot
magen til den tidligere beskrevne "fejlboot" til
"dth_regression".

linie 100, "regresboot2" medfører det samme skærbillede som det,
der vises af "fejlboot2", under hovedprogrammets
indlæsning. Her er det "regression", der indlæses.

Hovedprogrammerne er også identiske, når bortses
fra plotter-procedurerne, som alle gengives, dvs
fra linie 22200 til 29850.

linie 100, "regression", næste side, er den eneste nye linie.

PROCedurerne "beregny" og "print_plottekst", linie
22400 til 22880 er identiske. Det samme gælder for
PROCedurerne "rediger (lin)" og "udskriv (lin)", i
linie 26500 til 26920.

linie 22200 ff Alle kommandostrengene indledes med "^".
Derved opnås, at der benyttes de avancerede RD-GL-
kommandoer i stedet for de simple DXY-kommandoer.
Plotteren er bevidst forud-indstillet på DXY, som
bruges af enklere programmer. Omstilles plotteren
til RD-GL, kan DXY ikke benyttes. Til gengæld vil
samtlige "^" kunne spares.
Der er komma mellem x og y i kommandostrengene, og
man behøver ikke at benytte de irriterende heltal-
udregninger som på SERVOGOR-plotteren.

linie 22230 Tegn det angivne INPUT WINDOW.

linie 22240 Stiplet linie, PEN UP, PLOT ABSOLUTE, SELECT PEN 1

linie 22280 PEN UP, PEN DOWN, FULD LINIE, HELE VINDUET, PEN UP

linie 23050 USER DEFINED CHARACTER, se brugsanvisning.
Den følger de almindelige karakterer i størrelse,
og regnes i længde som een karakter, uanset at den
er både højere og meget længere, med sin streg hen
over den følgende streng.

linie 25420 Nulstil penarmen manuelt er det eneste irriterende
ved denne billige udgave af ROLAND DXY, 880. Papir
fastholdes med magnet-tape, som er udmærket, dog
ikke, hvis det parkeres oven på en diskette.

linie 25460 **SE ENDELIG REMark OM PENNES NUMRE !**

```

100 REMark ***** PROGRAM regression ***** (ROLAND DXY 880) *****
110 OPEN#7; ram2_: GET#7; load$, pr$, dansk0, dansk1: CLOSE#7
120 trat= 1: trap= 0: baudtal=9600: IF pr$== 'c': trap= 1: baudtal=4800
130 OVER#0;0: BORDER#0;0: INK#0;7: ktr$='i': alt$= 'f'
140 BAUD baudtal: TRA trap: ALTKEY 'q', 'ew cp80græsk_bin', ''

200 IF load$= 'gl' THEN
210 CSIZE#4;0,0: udjævn_vindue: koordinatsystem: existdatategning
220 CLS#4 : kontrollinie : CLS#4 : IF ktr$== 'p': tegnpolygon
230 OPEN#3, ser1: BPUT#3;27,108,10,27,81,77,27,67,72,27,78,10: CLOSE#3
240 END IF
300 IF load$= 'ny' THEN
310 n= 0: DIM p$(110,1): DIM g$(6,7): FOR n=0 TO 6: g$(n)='0'
320 DIM x$(110,7): DIM y$(110,7): FOR n= 0 TO 110: x$(n)='0': y$(n)='0'
330 DIM rgrnr(100): DIM udj$(40,24): DIM fjl(40)
340 alt$='i': dato$='': dat$='': filnavn$='': fil$='': fldat$=' '
350 hæld$='': ktr$='i': ox$= 'o': pdir$=' '
360 pkt$= 'ox': pstat$='': q$='': rg$='': sl$='': st$='': strg$=' '
370 tal$='': valg$='': xe$='': ye$='': ze$=' '
380 blæk= 0: ch= 0: cifre= 0: ent= 0: kanal= 0
390 kontrast= 0: ktrtal= 0: linie= 0: linie1= 0: linie2= 0: linn= 0
400 max= 0: min= 0: nygl= 0: n= 0: ordre= 0: papir= 0: pt= 0
410 p= 0: rgoantal= 0: rgtal= 0: rgxantal= 0: slutnr= 0: startnr= 0
420 starttid= 0: tab= 0: tal= 0: ttal= 0: xakse= 0: xantal=0
430 xfast= 0: xhæld= 0: xnedad= 0: xopad= 0: xparam= 0: xs= 0: yakse= 0
440 yantal= 0: yparam= 0: yfast= 0: yhjøre= 0: yvenst= 0: yskala= 0
450 dobb_vindue : MODE 4 : OVER 0: tidspunkt
460 udjævn_vindue: FOR n= 0,1,4,8: CLS#n
470 IF nygl= 2 : loadeproses: alt$= 'l'
480 load$='gl': OPEN#7; ram2_: PUT#7 \ 0, 'gl', pr$, dansk0, dansk1: CLOSE#7
490 END IF

600 REPEAT hovedloop
610 IF alt$ INSTR 'il' THEN
620 FOR n= 0,1,4,8: CLS#n
630 ktr$= 'i' : akseinput: inputtekst: pt= startnr - 1: xyinput
640 nygl= 2 : FOR n=0,4,8: PAPER#n;2: INK#n;7: CLS#n
650 END IF : REMark EXIT MED alt$= 'r' 's'
660 IF alt$== 's' THEN
670 DIM rgrnr(100): DIM udj$(40,24): DIM fjl(40)
680 tegning_lagring: REMark EXIT MED alt$= 'f' 'r'
690 IF alt$== 'f': regrestype : tegning_lagring
700 IF valg$== 'o+x': rg$= 'x': fejludjævning: tegning_lagring
710 REMark EXIT MED alt$= 'f' 'r'
720 hæld$= '': rg$= '': valg$= ' '
730 IF alt$== 'f': RUN
740 IF alt$== 'r': alt$= 'f' : REMark EXIT MED alt$= 'f'
750 END IF
760 IF alt$== 'r': dataændring: REMark EXIT MED alt$= 'f'
770 IF alt$== 'f': datasætvalg: REMark EXIT MED alt$= 'r' 's' 'l' 'h'
780 IF alt$== 'h' THEN
790 OPEN#7; ram2_: PUT#7 \ 0, 'ny', pr$, dansk0, dansk1: CLOSE#7: RUN
800 END IF
810 END REPEAT hovedloop

900 DEFINE PROCEDURE datasætvalg
910 CLS#0: PRINT#0; ' - FEJLUDJÆVNING / SAVE' \ ' / PLOT / PRINT f'
920 PRINT#0; ' - LAYOUT ÆNDRES l' \ ' - REVISIONER r'
930 PRINT#0; ' - HELT NYE DATA h': svarput 0,4,24, 'fsplrh', 0,2,7
940 alt$= q$: IF alt$ INSTR 'fsp': alt$= 's'
950 END DEFINE datasætvalg

```

```

22200 DEFine PROCedure plotkurve
22210 LOCAL type, trin, yt, xp(2), yp(2), xf, xt, gly
22220 type= p$(105): trin= (g$(2) - g$(1)) / 55: yt= g$(5)
22230 PRINT#5; '^IW 300,300,2500,2500;'
22240 PRINT#5; '^LT 2, .7; ^PU; ^PA; ^SP1;'
22250 FOR xt= x$(104) TO x$(105) STEP trin
22260   xf= y$(104) + y$(105) * xt
22270   gly= yt: beregny: IF ABS(gly - yt) > 1E8: EXIT xt: REMark ?? **
22280   IF xt = x$(104): PRINT#5; '^PU'; xt/skx-xnul ;','; yt/sky-ynul
22290   IF xt > x$(104): PRINT#5; '^PD'; xt/skx-xnul ;','; yt/sky-ynul
22300 END FOR xt: PRINT#5; '^LT ; ^IW; ^PU;'
22310 END DEFine plotkurve

```

```

22400 DEFine PROCedure beregny
22410 LOCAL y(3), fk: DIM y(3)
22420 FOR n= 1 TO 3
22430   fk= CODE(p$(n+100))
22440   SElect ON fk
22450   = 107: y(n)= x$(n+100)
22460   = 108: y(n)= x$(n+100) * LOG10 ( xf * y$(n+100))
22470   = 112: y(n)= x$(n+100) * ( xf ^ y$(n+100) )
22480   = 120: y(n)= x$(n+100) * ( y$(n+100) ^ xf )
22490   = 115: y(n)= x$(n+100) * SIN ( xf * y$(n+100))
22500   = 116: y(n)= x$(n+100) * TAN ( xf * y$(n+100))
22510 END SElect
22520 END FOR n
22530 SElect ON type
22540 = 1: yt= y(1) + y(2) + y(3)
22550 = 2: yt= y(1) + y(2) * y(3)
22560 = 3: yt= y(1) + y(2) / y(3)
22570 = 4: yt= ( y(1) + y(2) ) * y(3)
22580 = 5: yt= ( y(1) + y(2) ) / y(3)
22590 END SElect
22600 END DEFine beregny

```

```

22800 DEFine PROCedure print_plottekst
22820 IF plot$(lin)(25)='$' THEN
22830   BPUT#3;27,80: q$=plot$(lin)(1 TO 18): printerstreng q$
22840 END IF
22850 IF plot$(lin)(25)='#' THEN
22860   BPUT#3;27,77: q$=plot$(lin)(1 TO 24): printerstreng q$
22870 END IF : IF plot$(lin)(25)=' ': PRINT#3
22880 END DEFine print_plottekst

```

```

23000 DEFine PROCedure sqrtegn
23010 LOCAL l
23020 FOR n= p + 2 TO LEN(su$)
23030   IF su$(n)=' ': l= n - p - 2: EXIT n: ELSE l= LEN(su$) - p - 1
23040 END FOR n
23050 PRINT#5; '^UC 0,3,99,2,-4,2,12,': 1 + 6 * l ;',0;'
23060 END DEFine sqrtegn

```

```

24000 DEFine PROCedure plotteropen
24010 LOCal plot$(19,25),q$,navn$,gnlin$,lin,gllin,nylin,linloop
24020 LOCal lm,lp,xnul,ynul,skx,sky,csz,su$,xsu,ysu
24030 LOCal save3$, ft(12),fl,z$,slcf: DIM ft(9): z$= '': fl= 0
24040 lin= 0: skx= (g$(2) - g$(1)) / 2200: sky= (g$(5) - g$(4)) / 2200
24050 xnul= g$(1) / skx - 300: ynul= g$(4) / sky - 300

24070 BORDER#0;2,7:CLS#0: FOR n=0 TO 19:plot$(n)<1 TO 25>=FILL$(' ',25)
24080 save3$= fldat$ & filnavn$ & '_plottekst': z= 0: CLS#4

24100 IF DATAD$ <> fldat$: fl= fldat$(4): DATA_USE fldat$
24110 FOR n=1 TO 9: IF FOP_IN(#3,save3$ & n)=0: ft(n)= n: CLOSE#3
24120 FOR n=1 TO 9: IF ft(n): z$= z$ & ft(n)
24130 IF LEN(z$) : BORDER#4;3: WSTAT#4; save3$: BORDER#4;0
24140 IF fl : DATA_USE flp1_: IF fl= 1: DATA_USE flp2_
24150 REPEAT lp
24160 PRINT#0;' VÆLG PLOTTERTEKST:'\ ' NY TEKST : TAST 0'
24170 INK#0;4: PRINT#0;' EXISTERENDE FIL,'\ ' SE OVENFOR: ';
24180 INK#0;7: PRINT#0;'slutciffr'\ ' ANDEN FIL: TAST 10'
24190 talput 0,4,22,0,10,0,0: slcf= ttal: IF ttal= 0: EXIT lp
24200 IF tal$ INSTR z$ THEN
24210 OPEN_IN#6; save3$ & tal$
24220 FOR lin= 0 TO 19: GET#6; q$: plot$(lin)<1 TO 25>= q$
24230 CLOSE#6: EXIT lp
24240 END IF
24250 CLS#0: PRINT#0; '\ ' FILNAVN:'\ ' FLP' & FILL$(' ',20)\ '
24260 svarput 0,2,4,'12',0,2,7: z= FOP_DIR(#3,'flp' & q$ & '_')
24270 IF z= 0: AT#0;2,5: PRINT#0; '_': INPUT#0; navn$
24280 IF z= 0 AND FOP_IN(#3,'flp' & q$ & '_' & navn$)= 0 THEN
24290 z$= 'a': OPEN_IN#6; 'flp' & q$ & '_' & navn$
24300 FOR lin= 0 TO 19: GET#6; q$: plot$(lin)<1 TO 25>= q$
24310 CLOSE#3: CLOSE#6: EXIT lp
24320 END IF
24330 END REPEAT lp

24350 PAPER#4;2: INK#4;7: CLS#4: PRINT#4;\ ' I DETTE FELT KAN TEKST'
24360 PRINT#4;' TIL PLOTTER REDIGERES.': INK#4;4
24370 PRINT#4;\ ' PÅ FIGUREN SKRIVES'\ ' KUN AKSEENHEDERNE.'
24380 IF hæld$ <> '' THEN
24390 PRINT#4;\ ' FEJLUDJÆVNINGEN'\ ' SKRIVES IKKE.'
24400 PRINT#4; '\ ' PRINTERUDSKRIFT AF' \ ' UDJÆVNING: J / N'
24410 svarput 4,11,22,'jn',0,2,4: IF q$== 'j': udjævnukskrift
24420 END IF : INK#4;7
24430 AT#4;5,0: CLS#4,2: AT#4;7,2: PRINT#4; 'SKRIFTHØJDE, mm 4 6'
24440 PRINT#4; ' SKIFT MED : # $'\ ' TEGN / LINIE : 24 18'
24450 PRINT#4; ' MAX. ANTAL : 17 3'\ ' LINIER : 0 15'
24460 PRINT#4; '\ ' <CTRL> + <SHIFT>' \ ' + 2 >> 2. POTENS.'
24470 PRINT#4; ' + 3 >> 3. POTENS.' \ ' + 0 >> GRADTEGN.'
24480 PRINT#4; ' + V >> KVADRATROD.': INK#4;4
24490 PRINT#4; ' * = PLOTTER-SYMBOL' \ ' FOR KONTROL-PUNKT.'
24500 INK#4;0: FOR n= 0 TO 19: AT#4; n,0: PRINT#4; n
24510 AT#4; 0,0: PRINT#4; '0..456789012345678901234': INK#4;7

24530 CLS#0: PRINT#0; '\ ' VÆLG LINIENUMMER' \ ' TAST 0 - 19'
24540 talput 0,2,20,0,19,0,0: lin= ttal
24550 CLS#4: FOR n=0 TO 19: AT#4; n, 10-<LEN(n) > 9): PRINT#4; n;
24560 IF LEN(z$): FOR n=0 TO 19: IF plot$(n)<25> INSTR '#$': udskriv n
24570 REMark UDSKRIFT FORTSÆTTER PÅ NÆSTE SIDE

```

```

25000 REMark                plotteropen FORTSAT FRA FORRIGE SIDE
25010 REPEAT linloop
25020   CLS#0:BORDER#0;5:PRINT#0; \ ' FLYT CURSOR: ';; STRIP#0;7: INK#0;2
25030   AT#0; 1,14: PRINT#0; ' + ': AT#0; 1,18: PRINT#0; ' → '
25040   PAPER#0;2: INK#0; 7: PRINT#0;' RET VED AT SKRIVE'
25050   PRINT#0;' OVENI DET GAMLE.' \ ' AFSLUT MED < ESC >.'; BORDER#0;2,7

25070   rediger lin: gllin= lin
25080   CLS#0: PRINT#0;'   NYT LINIENR.:   0 - 19'
25090   PRINT#0;'   FÆRDIG           : TAST 20': INK#0;4
25100   PRINT#0;'   ENDRING AF LINIENR, FX' \ ' LINIE XX >> YY :': INK#0;7
25110   PRINT#0;'   TAST   XX.YY': talput 0,4,18,0,20,0,0: lin= ttal
25120   IF lin= 20 THEN
25130     CLS#0: PRINT#0; \ '   OK ?   J / N': svarput 0,1,16,'jn',0;2,7
25140     IF q$== 'j': BORDER#0;0: EXIT linloop: ELSE lin= gllin
25150   END IF
25160   IF (lin-INT(lin)) THEN
25170     gnlin$= tal$: gllin= INT(lin): nylin= gnlin$( LEN(gllin)+2 TO)
25180     plot$(nylin)(1 TO 25)= plot$(gllin)(1 TO 25)
25190     plot$(gllin)(1 TO 25)= FILL$(' ',25)
25200     udskriv gllin: udskriv nylin: lin= nylin
25210   END IF
25220 END REPEAT linloop

25240 CLS#0:   PRINT#0;' TEKST-LAGRING UNDER'
25250 INK#0;4: PRINT#0;' filnavn_plottekst* \ ' HVOR * ER ET NUMMER.'
25260 INK#0;7: PRINT#0;' TAST nr < ELLER 0 > \ ' <EN EXIST.FIL SLETTES)'
25270 talput 0,3,22,0,9,0,0
25280 IF ttal THEN
25290   DELETE save3$ & tal$: OPEN_NEW#6; save3$ & tal$
25300   FOR lin= 0 TO 19: q$= plot$(lin)(1 TO 25): PUT#6; q$
25310   CLOSE#6: slcf= ttal
25320 END IF
25330 CLS#0: PRINT#0; \ '   TEKST-UDSKRIFT PA' \ '   PRINTER   J / N'
25340 svarput 0,3,18,'jn',0,2,7
25350 IF q$== 'j' THEN
25352   CLS#0:PRINT#0; \ '   PRINTER TÆNDT J / N': svarput 0,1,22,'j',0,2,7
25360   OPEN#3,ser1: BAUD baudtal: TRA trap: BPUT#3;27,108,10
25370   PRINT#3; save3$ & slcf: PRINT#3; \ '
25380   FOR lin= 0 TO 19: print_plottekst
25390   BPUT#3;27,80: BPUT#3;12: CLOSE#3
25400 END IF : INK#0;4: CLS#0

25420 PRINT#0; '   SLUK PLOTTEREN.' \ '   NULSTIL PEN-ARMEN.'
25430 PRINT#0; '   ANBRING PAPIR A3.' \ '   TÆND PLOTTEREN.'
25440 PRINT#0; '   KLAR ?   J / N': svarput 0,4,18,'j',0,2,4: CLS#0
25450 PRINT#0; \ '   PEN 1 TIL RAMMEN.' \ '   PEN 0 TIL RESTEN.'
25460 REMark PEN-NUMRE RENUMMEREREDE 0 ... 7
25470 PRINT#0; \ '   KLAR ?   J / N': svarput 0,4,18,'j',0,2,4

25490 OPEN#5,ser2: BAUD 1200: TRA 1: PRINT#5; '^IN; '
25500 plotterakser: plottertekst: plotterpunkter: plotterregres

25520 PRINT#5;' ^SP0;': BORDER#0;0: INK#0;7: CLS#0;0
25530 PRINT#0; \ '   ER PENNENE FJERNET' \ '           J / N'
25540 svarput 0,2,18,'j',0,2,7: CLOSE#5: TRA trap: BAUD baudtal
25550 END DEFINE plotteropen

```

```

26000 DEFine PROCedure plotterakser
26010 LOCAL xhn, xk, yk
26020 PRINT#5; '^SP2;': REMark GROV PEN TIL RAMMEN
26030 FOR n= 1 TO 2: PRINT#5; '^PU; ^PA 300,300;'; '^ER 2200,2200; '
26040 IF g$(1) * g$(2) < 0 THEN
26050   plotterlinie 0,g$(4),0,g$(5): plotterlinie 0,g$(5),0,g$(4)
26060 END IF
26070 IF g$(4) * g$(5) < 0 THEN
26080   plotterlinie g$(1),0,g$(2),0: plotterlinie g$(2),0,g$(1),0
26090 END IF : PRINT#5; '^TL 1,0; '
26100 PRINT#5; '^PU; ^PA ' ; g$(1)/skx -xnul ;','; g$(4)/sky -ynul ;'; '
26110 FOR i= 1 TO yantal - 1: PRINT#5; '^PR 0, ' ; g$(6)/sky ;'; '^YT; '
26120 PRINT#5; '^TL 0,1.4; '
26130 PRINT#5; '^PU; ^PA ' ; g$(1)/skx -xnul ;','; g$(5)/sky -ynul ;'; '
26140 FOR i= 1 TO xantal - 1: PRINT#5; '^PR ' ; g$(3)/skx ;',0 ; '^XT; '
26150 PRINT#5; '^TL 0,1;': xhn= g$(2) / skx - xnul
26160 PRINT#5; '^PU; ^PA ' ; xhn ;','; g$(5)/sky -ynul ;'; '
26170 FOR i= 1 TO yantal - 1: PRINT#5; '^PR 0, ' ; -g$(6)/sky ;'; '^YT; '
26180 PRINT#5; '^TL 1.4,0; '
26190 PRINT#5; '^PU; ^PA ' ; g$(1)/skx -xnul ;','; g$(4)/sky -ynul ;'; '
26200 FOR i= 1 TO xantal - 1: PRINT#5; '^PR ' ; g$(3)/skx ;',0 ; '^XT; '

26220 PRINT#5; '^SP1; ^SI 0.27,0.405;': REMark FIN PEN, 4 mm SKRIFT
26230 plotterstreng xe$,1, xhn +5 + 20.5*LEN(g$(2)) -40.5*LEN(xe$), 130
26240 FOR n= xantal TO 0 STEP -1
26250   xk= g$(1)+g$(3)*n: tal$= xk: IF ABS(xk) < 1 : ciffertal xk, 1
26260   IF ABS(g$(3)) < 1: IF xk= xk DIV 1: tal$= tal$ & '.0'
26270   PRINT#5; '^PU; ^PA ' ; xk/skx - xnul +5 - 20*LEN(tal$);',220; '
26280   PRINT#5; '^LB ' ; tal$ ; CHR$(3)
26290 END FOR n
26300 FOR n= 0 TO yantal
26310   yk= g$(4)+g$(6)*n: tal$= yk: IF ABS(yk) < 1 : ciffertal yk, 1
26320   IF ABS(g$(6)) < 1: IF yk= yk DIV 1: tal$= tal$ & '.0'
26330   PRINT#5; '^PU; ^PA ' ; 280-LEN(tal$)*40.5 ;','; yk/sky -ynul;'; '
26340   PRINT#5; '^LB ' ; tal$ ; CHR$(3)
26350 END FOR n
26360 plotterstreng ye$, 1, 280 - LEN(tal$) * 40.5, g$(5)/sky -ynul +90
26370 IF ktr$== 'p' THEN
26380   FOR p= 101 TO 100 + ktrtal
26390     PRINT#5; '^PU; ^PA ' ; x$(p)/skx -xnul;','; y$(p)/sky -ynul;'; '
26400     PRINT#5; '^LT; ^CI 6; ^LT 2,.7; '
26410     IF p < 100 + ktrtal: plotterlinie x$(p),y$(p),x$(p+1),y$(p+1)
26420   END FOR p: PRINT#5; '^LT; '
26430 END IF
26440 IF ktr$== 'k': plotkurve
26450 END DEFine plotterakser

```

linie 26030 ER = rektangel 220 x 220 mm, 0-punkt i 30,30 mm.

linie 26090 TL +/- 1, n = akseinddelings-mærker.

linie 26220 SI b,h = skriftstørrelse.

linie 26200 PR = PLOT RELATIVE.

linie 26280 LB streng = skriv tekst. LT = fuld / stiplet linie.

linie 26400 CI radius = cirkel.

```

26500 DEFine PROCedure rediger ( lin )
26510 LOCAL a$, b$, tab, codeb, skrivloop, w: tab= 0
26520 a$= plot$(lin)<1 TO 24> & FILL$(' ',11)
26530 IF plot$(lin)<25>= '$' THEN
26540     CSIZE#4;1,0: INK#4;4: lm= 18
26550 ELSE : CSIZE#4;0,0: INK#4;7: lm= 24: plot$(lin)<25>= '#'
26560 END IF
26570 AT#4; lin, tab: CURSEN#4
26580 REPEAT skrivloop
26590     AT#4; lin, tab: b$= INKEY$(<#4,-1>): codeb= CODE(b$)
26600     SELECT ON codeb
26610     = 27: EXIT skrivloop
26620     = 37 TO 127, 130,131,134,137,138,160,162,166,170,182
26630     REMark INCL.Æ,Ø,Å, grad, 2.,3.potens, kvadratrod, EXCL.32-36
26640     a$(tab+1)= b$: AT#4; lin, tab: PRINT#4; a$(tab+1);
26650     tab= tab + (tab < lm-1)
26660     = 32
26670     FOR n= 34 TO tab + 1 STEP -1: a$(n+1)= a$(n)
26680     a$(tab+1)= ' '
26690     AT#4; lin,tab: FOR n= tab + 1 TO lm: PRINT#4; a$(n);
26700     tab= tab + (tab < lm-1)
26710     = 35,36
26720     IF codeb= 35: plot$(lin)<25>='#': CSIZE#4;0,0: INK#4;7: lm=24
26730     IF codeb= 36 THEN
26740         w= 0: FOR n= 19 TO 24: IF a$(n) <> ' ': w= 1
26750         IF w THEN
26760             AT#4; lin,0: PRINT#4;' STRENGEN ER FOR LANG ! ';
26770             PAUSE 50: plot$(lin)<25>='#': CSIZE#4;0,0: INK#4;7: lm=24
26780             ELSE plot$(lin)<25>='$': CSIZE#4;1,0: INK#4;4: lm=18
26790             END IF
26800         END IF
26810         AT#4;lin,0: PRINT#4; a$(1 TO lm);
26820         = 194, 202
26830         w = tab + (codeb= 202): IF w = 0: NEXT skrivloop
26840         FOR n= w TO 34: a$(n)= a$(n+1)
26850         a$(35)=' ': AT#4; lin, w -1 : FOR n= w TO lm: PRINT#4; a$(n);
26860         IF codeb= 194: tab= tab - (tab > 0)
26870         = 192: tab= tab - (tab > 0)
26880         = 200: tab= tab + (tab < lm-1)
26890         END SELECT
26900     END REPEAT skrivloop
26910     CURDIS#4: plot$(lin)<1 TO 24>= a$(1 TO 24): CSIZE#4;0,0: INK#4;7
26920 END DEFine rediger

```

```

27000 DEFine PROCedure udskriv (lin)
27010 IF plot$(lin)<25>= '#': CSIZE#4;0,0: INK#4;7: lm= 24
27020 IF plot$(lin)<25>= '$': CSIZE#4;1,0: INK#4;4: lm= 18
27030 AT#4;lin,0: PRINT#4; plot$(lin)<1 TO lm>;
27040 END DEFine udskriv

```

```

27500 DEFine PROCedure plottertekst
27510  LOCAl ned, venst, glcs, st, sl, p
27520  ned= g$(5)/sky -ynul + 50: venst = g$(2)/skx -xnul + 170: glcs= 1
27530  FOR lin= 0 TO 19
27540    st= 0: sl= 0: csz= 1: IF plot$(lin)(25)= '$': csz= 4/3
27550    FOR p= 1 TO 24: IF plot$(lin)(p) <> ' ': st= p: EXIT p
27560    IF st= 0: ned= ned - 90: IF lin= 19: RETurn : ELSE : NEXT lin
27570    FOR p= 24 TO 1 STEP -1: IF plot$(lin)(p) <> ' ': sl= p: EXIT p
27580    ned= ned - 120 + 30 * (glcs = 1) * ( csz = 1)
27590    xsu= venst + (st - 1) * 40.5 * csz
27600    REMark csz= 1 ELLER 4/3. 3 mm EXTRA FØR OG EFTER 5.4 mm SKRIFT
27610    REMark plotterstreng BRUGER TO PEN-BREDDER (NR. 1 OG 8)

27630    plotterstreng plot$(lin)(st TO sl), csz, xsu, ned: glcs= csz
27640  END FOR lin: PRINT#5; '^SP 1; ^SI .27, .405 ;'
27650 END DEFine plottertekst

28000 DEFine PROCedure plotterstreng ( su$, csz, xsu, ysu )
28010  LOCAl b$, p, op, pen, c, codes: pen= 1 + (csz > 1.2)
28020  FOR p= 0 TO LEN(su$) - 1
28030    codes= CODE(su$(p+1))
28040    SELEct ON codes
28050      = 42: b$= 'o': op= 14 * csz: c= .5 * csz
28060      = 137: b$= 'O': op= 20 * csz: c= .75 * csz
28070      = 131: b$= '3': op= 20 * csz: c= .75 * csz
28080      = 160: b$= '2': op= 20 * csz: c= .75 * csz
28090      = 182: op= 0: c= csz
28100      = REMAINDER : b$= su$(p+1): op= 0: c= csz
28110    END SELEct
28120    PRINT#5; '^SP'; pen ; '^SI'; .27 * c ; ', ' ; .405 * c ; ', ' ;
28130    PRINT#5; '^PA; ^PU'; xsu + p * 40.5 * csz ; ', ' ; ysu + op ; ', ' ;
28140    IF codes= 182: sqrtegn: IF p= LEN(su$) - 1: RETurn : ELSE NEXT p
28150    PRINT#5; '^LB'; b$ ; CHR$(3) ;
28160  END FOR p
28170 END DEFine plotterstreng

28500 DEFine PROCedure plotterpunkter
28510  PRINT#5; '^SP 1; ^SI 0.18, 0.36;': REMark ?? mm SKRIFT
28520  FOR p= startnr TO slutnr
28530    IF hæld$ <> ' ' AND LEN(valg$)= 1 AND p$(p) <> rg$: NEXT p
28540    PRINT#5; '^PU; ^PA'; x$(p)/skx -xnul; ', ' ; y$(p)/sky -ynul; ', ' ;
28550    IF p$(p)= 'o': PRINT#5; '^CI 12; '
28560    IF p$(p)= 'x': PRINT#5; '^CS 15; ^LBE'; CHR$(3); '^CS; '
28570  END FOR p
28580  IF valg$== 'o+x' THEN
28590    PRINT#5; '^PU; ^PA'; g$(2)/skx -xnul +180 ; ', ' , ; 320
28600    PRINT#5; '^CS 15; ^LBE'; CHR$(3); '^CS; '
28610    PRINT#5; '^PU; ^PA'; g$(2)/skx -xnul +180 ; ', ' , ; 380
28620    PRINT#5; '^CI 12; '
28630    IF ktr$ INSTR 'pk' THEN
28640      PRINT#5; '^PU; ^PA'; g$(2)/skx -xnul +180 ; ', ' , ; 440
28650      PRINT#5; '^CI 6; '
28660    END IF
28670  END IF
28680 END DEFine plotterpunkter

linie 28600      CS 15 = karaktersæt nr. 15, dvs "mærker". LBE = x.

```

```

29000 DEFine PROCedure plotterregres
29010  LOCal x1, x2, y1
29020  IF hæld$= '': RETurn
29030  PRINT#5;'^SP 2; ^SI 0.27, 0.405;': k= 0: IF rg$== 'x': k= 80
29040  IF LEN(valg$) < 3: FOR n= 1 TO 2: plotterregres2

29060  IF valg$== 'o+x' THEN
29070    rg$= 'o': k= 0: FOR n=1 TO 2: plotterregres2
29080    x1= g$(2) + 250 * skx: x2= x1 + 350 * skx
29090    y1= (380 + ynul) * sky: plotterlinie x1, y1, x2, y1
29100    :
29110    IF ktr$ INSTR 'kp' THEN
29120      PRINT#5;'^LT -2,0.7;'
29130      y1= (440 + ynul) * sky: plotterlinie x1, y1, x2, y1
29140    END IF
29150    :
29160    rg$= 'x': k= 20: PRINT#5;'^LT -2,1.5;'
29170    y1= (320 + ynul) * sky: plotterlinie x1, y1, x2, y1
29180    plotterregres2 : PRINT#5;'^LT;'
29190  END IF : REMark o FULD, x PUNKTERET LINIE, RETUR TIL FULD LINIE
29200 END DEFine plotterregres

29500 DEFine PROCedure plotterregres2
29510  REMark fjl (1)=gnx (2)=gny (10)=yvenst (11)=yhøjre (12)=xnedad
29520  REMark      (13)=xopad ( + k= 0 for o / 20 for x )
29530  :
29540  IF hæld$== 'l': plotterlinie fjl(1+k),g$(4),fjl(1+k),g$(5):RETurn
29550  IF hæld$== 'v': plotterlinie g$(1),fjl(2+k),g$(2),fjl(2+k):RETurn

29570  IF (fjl(10+k)-g$(4)) * (g$(5)-fjl(10+k)) > 0 THEN
29580    IF (fjl(11+k)-g$(4)) * (g$(5)-fjl(11+k)) > 0 THEN
29590      plotterlinie g$(1), fjl(10+k), g$(2), fjl(11+k): RETurn
29600    END IF
29610    IF fjl(5+k) > 0: plotterlinie g$(1),fjl(10+k),fjl(13+k),g$(5)
29620    IF fjl(5+k) < 0: plotterlinie g$(1),fjl(10+k),fjl(12+k),g$(4)
29630    RETurn
29640  END IF

29660  IF (fjl(11+k)-g$(4)) * (g$(5)-fjl(11+k)) > 0 THEN
29670    IF fjl(5+k) > 0: plotterlinie fjl(12+k),g$(4),g$(2),fjl(11+k)
29680    IF fjl(5+k) < 0: plotterlinie fjl(13+k),g$(5),g$(2),fjl(11+k)
29690    RETurn
29700  END IF

29720  plotterlinie fjl(12+k), g$(4), fjl(13+k), g$(5)/1
29730 END DEFine plotterregres2

29800 DEFine PROCedure plotterlinie ( x1, y1, x2, y2 )
29810  LOCal q1, q2, w1, w2
29820  q1 = x1 / skx - xnul: q2 = x2 / skx - xnul
29830  w1 = y1 / sky - ynul: w2 = y2 / sky - ynul
29840  PRINT#5;'^PU; ^PA'; q1 ;','; w1 ; '^PD'; q2 ;','; w2 ;',';
29850 END DEFine plotterlinie

```


OVERSIGTER OVER VARIABLER
VARIABELLISTE

OVERSIGTER OVER VARIABLER

SAMTLIGE VARIABLER i alfabetisk order er listet på de følgende sider.

Der angives type for de enkelte variable:

Procedure, Integer, String, String array, Repeat variable, Parameter (til procedure x), Local. Globale variable og decimaltal får ikke en "etiket". Det benyttede program "XREF" udskriver herudover visse "ord", jeg ikke opfatter som variable, fx con_ og ram2_, der jo kun er en del af en vinduesdefinition / en formatteringsproces. Det er imidlertid praktisk også at være i stand til at finde disse steder i programmet.

For hver variabel er udskrevet samtlige linienumre, hvor den bruges. Jeg har i et vist omfang indrammet væsentlige linienumre, fx linier med definition, startværdi, DIM etc.

Alt er ikke helt konsekvent, specielt ikke for variable der optræder lokalt / globalt forskellige steder i programmet. Jeg har nogenlunde manuelt lappet på manglerne.

Programmet "XREF" er en af lektor E. Borchersen bearbejdet version af programmet fra QL-USER, juli 1986.

Vigtige styringsvariable mv er beskrevet / listet flg. steder:

linie 100	Globale variable, vigtige styringsvariable.
linie 2500	alt\$ i relation til PROCeduren "enterdelaltesc", der
linie 5000	styrer angivelse af valgmuligheder under INPUT af x,y.
linie 4500	Brugen af OVER 1 / -1 under udtegn af punkter.
linie 10000	pkt\$, valg\$, rg\$, rgtal, rgoantal, rgxantal, rgrnr(n), der styrer fejludjævningen.
linie 11000	pkt\$, valg\$, rg\$, k, udj\$(n), fjl(n), rgrnr(n), bruges under udskrifter / lagring af fejludjævningen.
linie 11700	udj\$(n), til udskrifter på skærm og printer af fejlud- jævningen.
linie 15830	Brugen af parametrene kontrast, papir, blæk, OVER - 1 i PROCeduren "svarput" med blinkende spørgsmålstegn.
linie 19270	Sammenhængen mellem hæld\$ og fjl(40,1)
linie 21000	PROCeduren "kontrolkurve" forklares lidt nærmere, med eksempel. ktrtal, ktr\$, x\$(n), y\$(n), p\$(n), xt, yt.

VARIABELLISTE

akseinput Procedure
630 1000 1570

akseparametre Procedure
11870 12050 12110 12500 12640

alt\$ String
130 340 470 610 660 690 730 740
740 760 770 780 940 940 940 2560
2590 2600 2630 2740 2750 2760 3010 3150
3560 4060 4180 4200 4220 4350 4350 5050
5060 5180 5200 5290 5310 5420 5430 6060
6070 6080 6090 6100 6110 6120 6120 7160
13420 13430 13440 13450 13460 20280 20300 20330
20350

antal% Integer LOCAL
9010 9060 9070 9070 9080

a LOCAL
23010 23030 23070 23090 23090 23090 23090 23090
23090 23090

a\$ String LOCAL
3010 3180 3180 4010 4040 4040 26510 26520
26640 26640 26670 26670 26680 26690 26740 26810
26840 26840 26850 26850 26910

baudtal
120 120 140 13710 14040 14520 19630 25360
25540 30420 30520 30630

begynd LOCAL
30710 30730 30760 30760

beregny Procedure
22070 22260 22400 22600

blæk Parameter, Procedure svarput
380 15800 15830 15830 15840 15880 15890 15930

b\$ String LOCAL
3010 3370 3370 3380 3390 3410 5010 5080
5240 5250 5290 5330 15010 15040 15050 15060
15070 15120 26510 26590 26590 26640 28010 28050
28060 28070 28080 28100 28150

c80dir Procedure
30400 30450

c80list Procedure
30600 30670 30760

c80program Procedure
 30700 30820

c80stat Procedure
 30500 30550

ch Parameter, Procedure filservice
 380 17000 17010 17010 17010 17010 17010 17010
 17020 17030 17040 17050 17060 17060 17070 17080

ciffertal Procedure
 8500 8580 10430 11320 11330 11340 11350 11370
 11730 11780 11860 11920 11930 12040 12100 12160
 12530 12540 12570 12580 15200 15370 26220 26290

cifre Parameter, Procedure ciffertal
 380 3040 10430 11320 11330 11340 11350 11370
 11730 11730 11780 11780 11860 11920 11930 12040
 12100 12530 12540 12570 12580 15200 15220 15220
 15220 15250 15250 15250 15260 15290 15330

codeb LOCAL
 26510 26590 26600 26720 26730 26830 26860

codep LOCAL
 15010 15040 15050 15060 15070 15080

codes LOCAL
 28010 28030 28040 28140

con_
 30010 30010 30310

csz LOCAL Parameter, Procedure plotterstreng
 23030 24020 27550 27550 27590 27600 27630 27630
 28000 28010 28050 28050 28060 28060 28070 28070
 28080 28080 28090 28100 28120

c% Integer LOCAL
 28010 28050 28060 28070 28080 28090 28100 28120
 28120 28120 28130

dansk0
 110 480 790 30010 30250 30310

dansk1
 110 480 790 30010 30250 30310

datasætvalg Procedure
 770 900 950

dataændring Procedure
 760 6000 6160

datoloop		LOCAL	Repeat variable				
16020	16610	16790	16810				
dato\$		String					
340	14540	16360	16370	16830	18020	18050	18050
19080	19130	19650					
dat\$		String					
340	1440	1440	1460	1470	8650	14550	18260
19080	19660						
dec		LOCAL					
15210	15320	15330	15340				
dobb_vindue		Procedure					
450	30300	30370					
dtal\$		String LOCAL					
16010	16700	16710	16720	16820			
dx		LOCAL					
9010	9070	9090					
dy		LOCAL					
9010	9070	9090					
enkeltrevision		Procedure					
6090	7000	7180					
enterdelatasc		Procedure					
4350	5000	5440					
ent		Parameter, Procedure talput					
380	15400	15500					
existdatategning		Procedure					
210	1510	3560	4410	7640	8000	8060	13020
20280	21670						
expo		LOCAL					
15210	15260	15270	15280	15290	15310	15320	15320
15330							
fejludjævning		Procedure					
700	10510	11000	11520	12030	12210	13430	13440
figurloop		LOCAL	Repeat variable				
1010	1020	1540	1560				
filcheck		Procedure					
1290	1290	20000	20370				
fildata\$		String LOCAL					
16010	16320	16330	16360				

filnavn\$		String					
<u>340</u>	14540	16320	16320	18220	18240	18300	18950
18980	18990	18990	19060	19120	19210	19220	19290
19540	19650	24080					
filservice		Procedure					
16220	16410	16420	<u>17000</u>	17090	19020		
fil\$		String					
<u>340</u>	1410	1410	1430	1440	8670	8670	14550
18260	19080	19660					
fjernpunkt		Procedure					
6110	<u>7500</u>	7650					
fj1		Array					
<u>330</u>	670	11300	11300	11460	11470	11860	11940
11940	12040	12100	12160	12590	12590	12600	12610
12620	12630	19270	29540	29540	29550	29550	29570
29570	29580	29580	29590	29590	29610	29610	29610
29620	29620	29620	29660	29660	29670	29670	29670
29680	29680	29680	29720	29720			
finr		LOCAL					
<u>7510</u>	7550	7560	7560	7560	7570	7580	
fk		LOCAL					
<u>21010</u>	21420	21420	22410	22430	22440		
f11		LOCAL					
<u>18810</u>	<u>18840</u>	18850	18860				
f12		LOCAL					
<u>18810</u>	<u>18940</u>	18970	19000	19030			
f13		LOCAL					
<u>18810</u>	<u>19180</u>	19230	19250				
fldat\$		String					
<u>340</u>	16110	16120	16120	16130	16160	16170	16170
16320	16400	17020	17050	18220	18240	18830	18850
18990	19060	19220	19520	19540	24080	24100	24100
24100							
f1p1_							
16190	24140						
f1p2_							
16190	24140						
f1		LOCAL					
<u>16010</u>	16120	16120	16190	16190	24030	24030	24100
24140	24140						
fortegn		LOCAL					
<u>15210</u>	15230	15230	15360				

ft	LOCAL						
<u>24030</u>	24030	24110	24120	24120			
ft\$	String LOCAL						
<u>21010</u>	21390	21400	21400	21420	21420	21430	21440
21460							
fundet	LOCAL						
<u>16010</u>	16330	16330	16340	16350			
gem\$	String LOCAL						
<u>3010</u>	3010	3560					
glcs	LOCAL						
<u>27510</u>	27530	27590	27630				
gldato\$	String LOCAL						
<u>18210</u>	18260						
gllin							
<u>24010</u>	25070	25140	25170	25170	25180	25190	25200
glsl\$	String LOCAL						
<u>1010</u>	20030	20100	20120	20150	20220	20250	20320
20330							
glst\$	String LOCAL						
<u>1010</u>	20030	20050	20070	20150	20210	20250	20290
20300							
gly	LOCAL						
<u>22010</u>	22070	22080	22090	22210	22260	22260	
gnlin\$	String						
<u>24010</u>	25170	25170					
gnx	LOCAL						
<u>11020</u>	11300	11300	11330	11340	11430	11440	11870
12110	13070	13070					
gny	LOCAL						
<u>11020</u>	11300	11300	11370	11430	11440	11870	12110
13080	13080						
g\$	String Array						
<u>310</u>	310	1050	1130	1140	1190	1190	1190
1190	1200	1200	1210	1210	1210	1210	1210
1210	1490	1490	1490	1490	1490	1490	1500
1500	3240	3240	3250	3250	3250	3260	3290
3290	3300	3300	3300	4180	4180	4190	4190
4190	4200	4260	4260	4270	4270	4270	8430
8430	8430	8430	8430	8430	8440	8440	8440
8440	8450	8450	8450	8450	8460	8460	8460
8460	8490	8490	8510	8520	8530	8540	8570

fortsættes ...

... fortsat

8570	8590	8600	8610	8620	8640	8640	8650
8650	8660	8660	8670	8670	12600	12610	12620
12630	13070	13070	13080	13080	13100	13100	13110
13110	13120	13120	13140	13140	13150	13150	13190
13190	13200	13200	13210	13210	13250	13250	18270
18320	18320	18320	18320	18320	18320	18330	18330
18330	18330	18330	18330	18340	18340	19090	22020
22030	22030	22050	22060	22100	22100	22210	22220
22220	24040	24040	24040	24040	24050	24050	26040
26040	26050	26050	26050	26050	26070	26070	26080
26080	26080	26080	26190	26190	26220	26220	26230
26290	26290	26300	26350	27530	27530	28590	28610
28640	29080	29540	29540	29550	29550	29570	29570
29580	29580	29590	29590	29610	29610	29620	29620
29660	29660	29670	29670	29680	29680	29720	29720

hovedloop Repeat variable
 600 810

hæld#	String						
350	720	10390	10400	10460	11200	11210	11220
11230	11240	11250	11310	11320	11330	11340	11350
11360	11370	11490	11500	11510	11520	11720	11770
11830	11850	11910	12010	12020	12030	12090	13070
13080	13430	13440	13690	13730	14060	14070	14080
14090	14100	14150	14170	18900	19160	19280	24380
28530	29020	29540	29550				

inputtekst Procedure
 630 2500 2770 3150 6070 20280

inputudskrift Procedure
 13680 14500 14740 18660 20230 20350

k2 LOCAL
 21010 21590 21600 21600 21610 21610

kanal	Parameter, Procedure	svarput, sortertal, talput		
380	15400	15430	15430	15630
19600	19620	19690	19700	19800
19880	19910			19800
				19860
				15800
				15810
				19870

kontrast
 390 15800 15840 15890 15890

kontrolkurve Procedure
 3130 21000 21690 21700

kontrollinie Procedure
 220 1510 3000 3570

kontrol	LOCAL	Repeat variable		
15410	15480	15510	15580	15610
				15620

koordinatsystem			Procedure				
210	1510	3560	4410	7640	8400	8700	13020
20280	21670						

koordinat		LOCAL	Repeat variable				
3010	3160	3170	3260	3380	3450	4010	4020
4060	4220	4350	4360				

korrel		LOCAL					
11020	11040	11880	12060	12120	12150	12150	12150
12150	12160	12160	12160				

ktrp		LOCAL	
14510	14510	14720	

ktrtal							
390	3130	3560	9020	14640	14690	14700	18260
18290	18370	18550	18560	18560	19080	19110	21640
26370	26410						

ktr\$		String					
130	220	350	630	1520	1550	3070	3070
3100	3110	3130	3130	8680	8690	14690	14720
18370	18370	21640	26360	26440	26630	29110	

k		LOCAL					
1010	1190	1190	1200	11010	11060	11060	11110
11120	11130	11150	11200	11210	11220	11230	11240
11250	11300	11300	11310	11320	11330	11340	11350
11350	11380	11380	11460	11470	11520	11740	11790
11820	11860	11870	11920	11930	11930	11940	11940
11940	12040	12050	12100	12110	12160	12170	12170
12190	12510	12530	12540	12540	12550	12570	12580
12580	12590	12590	12600	12610	12620	12630	14010
14020	14020	14060	14060	14060	14070	14080	14080
14090	14100	14150	14180	14180	14190	14200	14200
14210	14210	14220	14220	14270	15410	15470	15530
15540	15550	15550	15560	15560	15580	15810	15810
15830	15830	15840	15840	15840	15860	15860	15860
15860	15860	15880	15880	15890	15890	15900	15900
15930	15930	15930	18810	18950	18980	19000	19610
19760	19770	19790	19790	19790	19820	19820	19830
19830	19830	19850	19850	19860	19870	19890	20010
20010	20080	20110	20120	20130	20130	20170	20210
20210	20220	20220	20270	29030	29030	29070	29160
29540	29540	29550	29550	29570	29570	29580	29580
29590	29590	29610	29610	29610	29620	29620	29620
29660	29660	29670	29670	29670	29680	29680	29680
29720	29720						

lenn		LOCAL					
15410	15470	15490	15530	15540	15560	15580	15630

liniel		Parameter, Procedure c80list				
390	30600	30610	30610	30610	30650	

linie2 Parameter, Procedure c80list
 390 30600 30650

linie Parameter, Procedure talput
 390 15400 15430 15630

linloop Repeat variable
 24010 25010 25140 25220

linn Parameter, Procedure svarput
 390 15800 15840 15860 15900

lin	Parameter, Procedure udskriv, rediger					Local	
22820	22830	22850	22860	22870	24010	24040	24220
24220	24300	24300	24540	25070	25070	25110	25120
25140	25160	25160	25170	25200	25290	25290	25380
26500	26520	26530	26550	26570	26590	26640	26690
26720	26760	26770	26780	26810	26850	26910	27000
27010	27020	27030	27030	27540	27550	27560	27570
27570	27580	27630	27640				

lm	LOCAL						
24020	26540	26550	26650	26690	26700	26720	26770
26780	26810	26850	26880	27010	27020	27030	

loadepoces Procedure
 470 18200 18400

loades\$ String LOCAL
 18210 18220 18250

load\$ String
 110 200 300 480

logt LOCAL
 15210 15260 15260 15320 15330

loop1 LOCAL Repeat variable
 19610 19780 19890 19900

loop2 LOCAL Repeat variable
 1010 1100 1220 1240 19610 19810 19820 19830
 19840

loop3 LOCAL Repeat variable
 1010 1110 1160 1180

loop LOCAL Repeat variable
 15810 15850 15900 15920

lp1 LOCAL Repeat variable
 21010 21380 21400 21410

lp LOCAL Repeat variable
 14010 14140 14250 14290 24020 24150 24190 24230
 24310 24330

læn	LOCAL			
15210	15320	15330	15340	

l	LOCAL			
23010	23050	23050	23090	

max	Parameter, Procedure talput			
400	15400	15450	15600	

min	Parameter, Procedure talput			
400	15400	15500	15600	

måned_til_streng	Procedure			
16690	17500	17650		

måned	LOCAL			
16010	16690	16820	17510	

m	LOCAL				
15410	15540	15550	15560	15570	

navn\$	String	Local		
24010	24270	24280	24290	

ned	LOCAL					
27510	27530	27570	27570	27590	27590	27630

nrloop	LOCAL	Repeat variable		
7010	7110	7140	7160	7170

nygl							
400	470	640	1030	2030	3080	16290	16290
16340	16350	16410	16420	16440	20020		

nylin	Local			
24010	25170	25180	25200	25200

ny\$	String	LOCAL					
1010	1380	1380	1380	1380	1400	1400	1400
1400	1420	1430	1430	1430	1450	1460	1460
1460							

n	Local						
310	310	310	320	320	320	400	460
460	620	620	640	640	640	640	1050
1050	1050	1120	1130	1130	1140	1140	1150
1170	1170	1230	1230	1550	1550	2510	2510
2700	2700	3180	3180	3460	3460	3460	3460
3550	3550	3550	3550	4040	4040	7580	7580
7580	7580	7580	7580	7580	8480	8490	8550
8560	8570	8630	9020	9030	9030	9040	9040
9050	9050	9050	9050	9070	9070	9070	9070
9090	9090	9110	10050	10050	10050	10050	10240
10250	10250	10250	10260	11100	11100	11100	11100

fortsættes ...

... fortsat

11130	11130	11130	11270	11280	11280	11280	11290
11370	11370	11380	11380	11410	11420	11430	11430
11440	11440	11460	11820	11820	11820	11950	11960
11970	11970	11980	11980	12000	12190	12190	12190
13740	14510	14590	14600	14600	14610	14610	14620
14620	14620	14630	14640	14640	14680	14700	14710
14710	14710	14710	14710	14720	14730	15410	16040
16040	16070	16070	16200	16200	16830	16830	18210
18270	18270	18280	18280	18280	18280	18290	18290
18290	18290	18510	18550	18560	18570	18570	18600
18600	18610	18610	18620	18640	19090	19090	19100
19100	19100	19100	19110	19110	19110	19110	19270
19270	20210	20210	20210	20210	20220	20220	20220
20220	21010	21020	21020	21020	21020	21150	21160
21160	21170	21170	21180	21370	21390	21390	21420
21430	21440	21450	21450	21470	21480	21480	21500
21570	21570	21570	21600	21600	21600	21600	21640
21640	21640	21640	22420	22430	22450	22450	22460
22460	22460	22470	22470	22470	22480	22480	22480
22490	22490	22490	22500	22500	22500	22520	23010
23040	23050	23050	23050	23060	23090	24070	24070
24110	24110	24110	24110	24120	24120	24120	24500
24500	24500	24550	24550	24550	24550	24560	24560
24560	26030	26210	26220	26270	26280	26290	26340
26670	26670	26670	26690	26690	26740	26740	26840
26840	26840	26850	26850	29040	29070	30010	30010
30250	30250	30250	30310	30310	30740	30750	30770

n\$ String LOCAL

14510	14600	14600	14600	14610	18510	18570	18570
18570	18590	18590					

op LOCAL

28010	28050	28060	28070	28080	28090	28100	28130
-------	-------	-------	-------	-------	-------	-------	-------

ordre Parameter, Procedure filcheck

400	1290	1290	1290	1290	1300	20000	20020
20030	20170	20230	20350				

ox\$ String

350	4140	5130	5140	5150
-----	------	------	------	------

papier Parameter, Procedure svarput

400	15800	15830	15830	15880	15890	15930
-----	-------	-------	-------	-------	-------	-------

pdir\$ String Parameter, Procedure c80dir

350	30400	30430	30430	30430	30440
-----	-------	-------	-------	-------	-------

pen LOCAL

22230	24020	25440	25440	25460	25460	26020	26180
27640	28010	28510	29030				

pkt\$	String						
360	4370	4380	4380	6140	6150	6150	8040
10030	10030	10030	10200	10210	11070	11080	13010
13020	13030	18380	18390	18390	19280		

plotkurve	Procedure					
22200	22320	26440				

plotterakser	Procedure					
25500	26000	26450				

plotterlinie	Procedure					
26050	26050	26080	26080	26410	29090	29130
29540	29550	29590	29610	29620	29670	29170
29800	29850					29720

plotteropen	Procedure					
13440	24000	25550				

plotterpunkter	Procedure					
25500	28500	28680				

plotterregres2	Procedure					
29040	29070	29180	29500	29730		

plotterregres	Procedure					
25500	29000	29200				

plotterstreng	Procedure					
26200	26350	27630	28000	28170		

plottertekst	Procedure					
25500	27500	27650				

plot\$	String	LOCAL				
22820	22830	22850	22860	22870	24010	24070
24300	24560	25180	25180	25190	25290	26520
26550	26720	26770	26780	26910	27010	27020
27550	27560	27580	27630			27030

pn	LOCAL					
21010	21590	21600	21610			

pox	LOCAL					
4010	4030	4120	4120	4120	4300	4310
4310						

printerloop	Repeat variable					
13620	13670	13790				

printerstreng	Procedure					
14560	14570	15000	15140	22830	22860	

printerudskrift	Procedure					
13450	13600	13800				

print_plottekst	Procedure					
22800	22880	25380				

pr#	String				
<u>110</u>	120	480	790	14510	15020

pscr#	String			
<u>4010</u>	4070	4070	4070	4080

pstat#	String	Parameter,	Procedure	c80dir
<u>360</u>	<u>30500</u>	30530	30530	30530 30540 30540

pt							
<u>400</u>	630	3150	3170	3170	3170	3170	3190
3200	3200	3210	3210	3220	3250	3250	3250
3260	3260	3260	3260	3270	3270	3290	3300
3300	3300	3310	3310	3310	3320	3320	3380
3400	3400	3410	3410	3480	3480	3490	3490
3500	3530	3560	4060	4060	4060	4070	4070
4090	4090	4100	4100	4110	4110	4110	4120
4120	4120	4140	4140	4150	4190	4190	4190
4210	4210	4210	4220	4220	4220	4240	4240
4260	4270	4270	4270	4280	4280	4280	4330
4330	4330	5110	5110	5110	5120	5130	5140
5160	5160	5160	5190	5190	5190	5260	5260
5260	5300	5300	5300	5340	5340	5340	5350
5350	6080	7150	20300	20330			

p	LOCAL						
<u>410</u>	3530	3530	3530	3530	4380	4380	4380
6150	6150	6150	8030	8040	8040	8040	8040
8050	9080	9090	9090	9100	<u>15010</u>	15030	15040
15040	15100	15100	15130	<u>15410</u>	15470	15560	15560
15580	<u>18210</u>	18390	18390	18390	<u>18510</u>	18520	18570
18570	18580	18630	<u>19610</u>	19760	19790	19790	19790
19860	23040	23050	23050	26370	26380	26390	26410
26410	26410	26410	26410	26420	<u>27510</u>	27560	27560
27560	27560	27580	27580	27580	27580	<u>28010</u>	28020
28030	28100	28130	28140	28140	28160	28520	28530
28530	28540	28540	28550	28560	28570		

p#	String	Array					
<u>310</u>	3080	3110	3210	3210	3220	3260	3490
3530	4100	4100	4110	4120	4140	4140	4150
4210	4220	4330	4370	4380	5110	5120	5130
5140	5160	5190	5260	5300	5340	6140	6150
7560	7580	7580	7600	8040	8040	10250	11280
11420	11960	14620	14720	18280	18290	18370	18380
18390	18620	19100	19110	20210	20220	20300	20330
21020	21360	21420	21510	21510	21590	21600	21640
22020	22220	22430	28530	28540	28560		

q1	LOCAL	
<u>29810</u>	29820	29840

q2	LOCAL	
<u>29810</u>	29820	29840

q	LOCAL						
<u>23010</u>	23030	23090	23090	23090	23090	23090	23090

q#	String	Local					
<u>360</u>	940	1160	1540	1550	3070	3100	3110
3120	3530	3540	4410	6060	7640	10330	10390
11170	13350	13420	13670	13680	13690	13700	15860
15870	15870	15900	15930	16110	16120	16290	16790
16800	18660	18900	18930	19160	20200	21550	21690
22830	22830	22860	22860	<u>24010</u>	24220	24220	24260
24280	24290	24300	24300	24410	25140	25290	25290
25340	25440						

ram2_							
110	480	790					

redaktion	Procedure						
<u>30200</u>	30260						

rediger	Procedure						
25070	<u>26500</u>	26920					

regrestype	Procedure						
690	<u>10000</u>	10520					

retn	LOCAL						
<u>22010</u>	22030	22050	22060				

rgoantal							
<u>410</u>	<u>10260</u>	11070	14270	19280	19760		

rgnr	Array						
<u>330</u>	670	<u>10250</u>	10250	19790	19790	19820	19820
19850	19850						

rgtal							
<u>410</u>	<u>11060</u>	<u>11070</u>	<u>11080</u>	11110	11140	11300	11300
11460	11470	14010	14270	19280	19710	19720	19760
19760	19770	19770					

rgxantal							
<u>410</u>	<u>10260</u>	11080	14010	19280	19770		

rg#	String						
<u>360</u>	700	720	10030	10210	10270	11060	11070
11080	11120	11280	11420	11960	13020	13050	13320
14010	14020	14250	14270	19280	19710	19770	28530
29030	29070	29160					

r	LOCAL						
<u>10010</u>	10010	10250	10250	10250	10260		

sakxf	LOCAL						
<u>11010</u>	11050	11970	11970	12010	12020	12040	12060

sakx	LOCAL						
11010	11050	11430	11430	11460	11500	11510	11860
11880	12120						
sakyf	LOCAL						
11010	11050	11970	11970	12060			
saky	LOCAL						
11010	11050	11430	11430	11470	11880	12060	12060
12120							
sapxyf	LOCAL						
11010	11060	11980	11980	12010	12020	12040	12060
sapxy	LOCAL						
11010	11050	11440	11440	11500	11510	11860	11880
12120							
save2\$	String LOCAL						
18810	19220	19230	19260				
save3\$	String LOCAL						
24030	24080	24110	24130	24210	25280	25280	25370
saveproces	Procedure						
6100	13430	18800	19310				
save\$	String LOCAL						
18810	18990	19000	19060	19070			
ser1							
230	13710	13740	13760	14040	14520	19630	25360
30420	30520	30630	30770				
ser2							
25490							
size	LOCAL						
23010	23030	23030	23030	23030			
skrivloop	LOCAL						
26510	26580	26610	26830	26900			
skr\$	String LOCAL						
20010	20060	20070	20110	20120			
skx	LOCAL						
22280	22290	24020	24040	24050	26190	26240	26380
27530	28540	28590	28610	28640	29080	29080	29820
29820							

sky	LOCAL						
22280	22290	24020	24040	24050	26320	26350	26390
27530	28550	29090	29130	29170	29830	29830	

slcf			
24030	24190	25300	25370

slutnr							
410	2060	2070	2070	4060	7130	7140	7550
7570	7600	7600	7600	7600	7600	7610	7610
8010	14590	14640	18260	18280	18360	18360	18550
18560	19080	19100	19720	20110	20110	20250	28520

sluttid		Procedure
18000	18060	18920

slut	LOCAL		Parameter, Procedure	talput		
15400	15450	19610	19760	19770	19830	19870
30710	30750	30750	30760	30760		19890

sl	LOCAL		
27510	27550	27580	27630

sl\$	String						
360	2030	2070	2070	4380	6150	7580	7610
8030	10240	11060	11110	11140	11270	11410	11950
18360	18390	19700	20030	20100	20150	20220	20250
20250	20320	20330					

sortertal		Procedure		
11150	11170	14120	14280	19600
				19930

spmloop1	LOCAL	Repeat variable
16020	16030	16160
		16180

spmloop2	LOCAL	Repeat variable
16020	16230	16340
		16380
		16430

spredx	LOCAL		
11020	11460	11460	11730

spredy	LOCAL		
11020	11470	11470	11780

sqrtegn		Procedure
23000	23110	28140

startnr							
410	630	2040	2050	2050	4370	6080	6140
7130	7530	7550	8010	14590	14630	14640	18260
18280	18350	18350	18380	18550	18560	19080	19100
19720	20060	20060	20250	28520			

starttid		
420	16830	18020

start_slut		Procedure					
1290	2000	2090					
stp		LOCAL					
28010	28010	28120					
strg\$		String		Parameter, Procedure		printerstreng,	svarput
360	15000	15020	15030	15040	15040	15100	15800
15870							
stsl		LOCAL		Repeat variable			
1010	1280	1300	1310				
st		LOCAL					
27510	27550	27560	27570	27600	27630		
st\$		String					
360	2030	2050	2050	4380	6150	8030	10240
11060	11110	11140	11270	11410	11950	18350	18390
19700	20030	20050	20150	20210	20250	20250	20290
20300							
sumx		LOCAL					
11010	11050	11280	11280	11300			
sumy		LOCAL					
11010	11050	11280	11280	11300			
su\$		String		LOCAL		Parameter, Procedure	
23040	23050	23050	24020	28000	28020	28030	28100
28140							
svarput		Procedure					
930	1160	1540	3070	3100	3520	4400	6060
7630	10320	10390	11160	13420	13610	13660	15800
15940	16100	16150	16290	16790	18660	18850	18900
20190	21540	21680	24260	24410	25130	25330	25350
25440	25470	25530	30410	30510	30620	30720	
s		LOCAL					
10010	10010	10250	10250	10250	10260		
tab		Parameter, Procedure			svarput,	talput	
420	15400	15430	15630	15800	15840	15860	15900
26510	26510	26570	26590	26640	26640	26640	26650
26650	26650	26670	26680	26690	26690	26700	26700
26700	26830	26860	26860	26860	26870	26870	26870
26880	26880	26880					
talput		Procedure					
1130	2040	2060	3040	3240	3290	4180	4260
7130	7550	10420	10440	10480	15400	15650	16680
16690	16700	16730	21360	21450	21480	21530	21570
24190	24540	25110	25260				
tal		LOCAL					
420	15200	15210	15410	15580	15600	15600	

tal\$	String						
<u>370</u>	1130	3250	3270	3300	3310	4190	4240
4270	4280	8490	8510	8510	8540	8540	8570
8590	8590	8620	8620	10430	10440	11320	11330
11340	11350	11380	11740	11790	11870	11920	11930
12050	12110	12170	12530	12540	12570	12580	15220
15250	15290	15290	15290	15340	15360	15360	15430
15450	15470	15530	15550	15560	15580	21360	21450
21480	21550	21570	24200	24210	25170	25280	25280
26220	26230	26230	26240	26260	26290	26300	26300
26310	26330	26350					

tastind		LOCAL	Repeat variable			
<u>15410</u>	<u>15420</u>	15450	15500	15600	15640	

tegning_lagring			Procedure	
680	690	700	<u>13300</u>	13480

tegnkurve		Procedure	
3110	8490	<u>22000</u>	22120

tegnpolygon		Procedure	
220	1520	8680	<u>9000</u> 9120

tidspunkt		Procedure
450	<u>16000</u>	16840

tids\$	String	LOCAL					
<u>16010</u>	16340	16380	16680	16720	16760	16770	16820
16830	17520	17530	17540	17550	17560	17570	17580
17590	17600	17610	17620	17630			

timetal		LOCAL	
<u>18010</u>	18020	18030	18030

time\$	String	LOCAL				
<u>18010</u>	18030	18030	18030	18040	18040	18050

tim\$	String	LOCAL			
<u>16010</u>	16730	16740	16750	16760	16820

t1	LOCAL					
<u>15210</u>	15210	15220	15230	15240	15240	15250
15270	15270	15290	15310	15310	15340	<u>22010</u>
22090	22090	22090	22100			22020

trap						
<u>120</u>	120	140	13720	14040	14520	19630
25540	30420	30520	30630			25360

trat	
<u>120</u>	13710

trin	LOCAL						
22010	22030	22040	22090	22210	22220	22240	
ttal							
420	1130	2040	2040	2060	2060	3040	3250
3260	3300	4190	4200	4270	7140	7150	7550
10420	10440	10480	15450	15500	15600	16680	16690
16700	16710	16710	16740	16740	16750	24190	24190
24540	25110	25270	25300				
typevalg	LOCAL	Repeat variable					
10010	10020	10030	10330	10340			
type	LOCAL						
22010	22020	22210	22220	22530			
t	LOCAL						
3010	3010	3410	3410	3560	19610	19760	19860
19860	19870	19870	19880				
udjævnfilservice	Procedure						
19170	19240	19500	19570				
udjævnudskrift	Procedure						
13690	13740	14000	14310	24410			
udjævn_vindue	Procedure						
210	460	30000	30080				
udj\$	String Array						
330	670	11110	11120	11130	11150	11200	11210
11220	11230	11240	11250	11310	11320	11330	11340
11350	11350	11380	11380	11520	11740	11790	11820
11870	11920	11930	11930	11940	12050	12110	12170
12170	12190	12510	12530	12540	12540	12550	12570
12580	12580	14060	14060	14060	14070	14080	14080
14090	14100	14150	14180	14180	14190	14200	14200
14210	14210	14220	14220				
udskriftloop	LOCAL	Repeat variable					
13310	13370	13460	13470				
udskriv	Procedure						
24560	25200	25200	27000	27040			
udtegning	Procedure						
11830	12200	13000	13260				
u	LOCAL						
20010	20300	20300	20300	20330	20330	20330	
valg\$	String						
370	700	720	10030	10190	10200	10210	10210
10230	10230	10230	10270	10270	10320	13010	13320
14010	14250	19280	28530	28580	29040	29060	

venst	LOCAL						
27510	27530	27600					
vlq	LOCAL	Repeat variable					
10010	10180	10210	10210	10270	10290		
vt	LOCAL						
15810	15930	15930					
w1	LOCAL						
29810	29830	29840					
w2	LOCAL						
29810	29830	29840					
w	LOCAL						
26510	26740	26740	26750	26830	26830	26840	26850
26850							
x1	LOCAL	Parameter, Procedure plotterlinie					
26010	26190	26200	26240	26250	26310	26320	26380
26390	28540	28550	29010	29080	29080	29090	29130
29170	29800	29820					
x2	LOCAL	Parameter, Procedure plotterlinie					
29010	29080	29090	29130	29170	29800	29820	
xakse							
420	12560	12570	12590				
xantal							
420	1210	1220	1220	8560	18320	26110	26130
26210							
xe#	String						
370	1340	1340	1380	1390	8660	8660	14560
14560	18260	19080	26190	26200			
xfast							
430	10420	10430	11920	11940	11970	11980	12050
xf	LOCAL						
22010	22070	22210	22250	22460	22470	22480	22490
22500							
xhæld							
430	10480	11860	11860	11860	12040	12040	12040
12060	12060	12100	12100	12120	12120	12520	12560
12600	12610	12620	12630	13140	13150	13200	13210
xkoor	LOCAL						
8410	8570	8570	8580	8580	8590	8590	8600
8610	8620						

xk	LOCAL						
26010	26220	26220	26220	26220	26230	26230	26240
xnedad							
430	12620	12620	13150	13200	13250		
xnul	LOCAL						
22280	22290	24020	24050	26190	26240	26380	27530
28540	28590	28610	28640	29620	29820		
xopad							
430	12630	12630	13140	13210	13250		
xparam		Parameter, Procedure akseparamatre					
430	12500	12520	12560	12600	12610	12620	12630
xsu	LOCAL						
24020	27600	27630	28000	28130			
xs							
430	1490	1500	3320	3400	3480	4110	4210
4310	4330	5110	5160	5190	5260	5300	5340
7560	8040	8430	8430	8430	8440	8440	8460
8460	8520	8530	8540	8600	8610	8620	8640
8650	8660	8670	9030	9050	9090	11500	11510
12010	12020	13070	13070	13080	13080	13120	13120
13140	13140	13150	13150	13200	13200	13210	13210
13250	13250	18330	18340	22100			
xt	LOCAL						
22010	22040	22050	22050	22050	22060	22060	22060
22070	22080	22090	22100	22110	22210	22240	22250
22260	22270	22280	22290	22310			
xx	LOCAL						
9010	9050	9060					
xyinput		Procedure					
630	4000	4420	6080	7150	20300	20330	
xykanal4		Procedure					
18380	18500	18670					
xyslut	LOCAL	Repeat variable					
3010	3350	3420	3440	5010	5030	5200	5270
5310	5350	5380					
x\$	String Array						
320	320	3200	3250	3250	3250	3270	3270
3290	3310	3320	3400	3480	3530	4090	4110
4120	4190	4190	4190	4210	4240	4240	4260
4280	4330	5110	5160	5190	5260	5300	5340
7560	7580	7580	7600	8040	9030	9050	9050
9070	9070	9090	11280	11430	11440	11970	11980
14610	14610	14710	14710	18280	18290	18600	18600

fortsættes ...

... fortsat

19100	19110	20210	20220	21020	21450	21510	21510
21570	21640	22030	22030	22040	22040	22240	22240
22270	22450	22460	22470	22480	22490	22500	26380
26410	26410	28540					

y1	LOCAL	Parameter, Procedure plotterlinie					
29010	29090	29090	29090	29130	29130	29130	29170
29170	29170	29800	29830				

y2	Parameter, Procedure plotterlinie						
29800	29830						

yakse							
430	12520	12540	12590				

yantal							
440	1210	1220	1220	8480	18320	26120	26140
26280							

yes	String						
370	1390	1390	1400	1410	8640	14570	14570
18260	19080	26350					

yfast							
440	10440	11930	11940	11970	11980	12050	

yhøjre							
440	12610	12610	13110	13110	13120	13190	13190
13200	13210						

ykoor	LOCAL						
8410	8490	8490	8500	8500	8510	8510	8520
8530	8540						

yk	LOCAL						
26010	26290	26290	26290	26290	26300	26300	26320

ynul	LOCAL						
22280	22290	24020	24050	26320	26350	26390	27530
28550	29090	29130	29170	29830	29830		

yparam	Parameter, Procedure akseparametre						
440	12500	12520	12560	12600	12610	12620	12630

yskala							
440	1490	1500	1500	1500	8520	8530	8600
8610	9060	18330	18340	18340	18340		

ysu	LOCAL	Parameter, Procedure plotterstreng				
24020	28000	28130				

yt	LOCAL						
22010	22020	22070	22080	22090	22100	22100	22100
22210	22210	22260	22260	22280	22290	22540	22550
22560	22570	22580					

yvenst							
440	12600	12600	13100	13100	13120	13140	13150

yy	LOCAL	
9010	9050	9060

y	LOCAL						
22410	22410	22450	22460	22470	22480	22490	22500
22540	22540	22540	22550	22550	22550	22560	22560
22560	22570	22570	22570	22580	22580	22580	23010
23030	23090	23090					

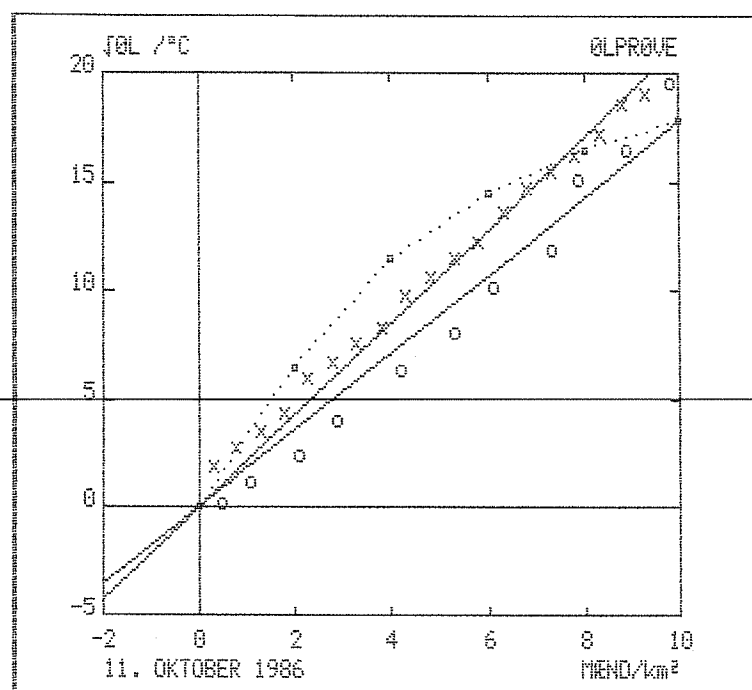
y\$	String Array						
320	320	3200	3300	3300	3300	3310	3310
3320	3400	3480	3530	4090	4110	4120	4210
4270	4270	4270	4280	4280	4330	5110	5160
5190	5260	5300	5340	7560	7580	7580	7600
8040	9030	9050	9050	9070	9070	9090	11280
11430	11440	11970	11980	14620	14620	14710	14710
18280	18290	18610	18610	19100	19110	20210	20220
21020	21480	21550	21550	21550	21600	21600	21640
22030	22070	22070	22250	22250	22460	22470	22480
22490	22500	26390	26410	26410	28550		

z			
24080	24260	24270	24280

z\$	String	LOCAL					
370	5120	5130	5140	17060	18630	18810	18960
18970	18980	18980	18980	18990	19210	19220	19290
19550	19790	19800	19850	19860	24030	24030	24120
24120	24130	24200	24290	24560			

endloop	LOCAL	Repeat variable
6010	6020	6120
		6130

TEORI



RET LINIE GENNEM PUNKT X,Y =
0.000 , 0.000

SERIEN 10-39 HAR 19 PUNKTER MED SYMBOLET x.
NUMRENE ER: 20-38

TYNGDEPUNKT X,Y = 4.800 , 10.53

HÆLDNING= 2.135

SKÆRING MED Y-AKSEN I 0.000 , 0.000

SKÆRING MED X-AKSEN I 0.000 , 0.000

KORRELATIONSKOEFFICIENT = 0.994

SERIEN 10-39 HAR 11 PUNKTER MED SYMBOLET o.
NUMRENE ER: 10-19, 39

TYNGDEPUNKT X,Y = 5.100 , 8.627

HÆLDNING= 1.782

SKÆRING MED Y-AKSEN I 0.000 , 0.000

SKÆRING MED X-AKSEN I 0.000 , 0.000

KORRELATIONSKOEFFICIENT = 0.982

TEORI.

Der er i programmet kun benyttet kendte, statistiske metoder, hvis baggrund er beskrevet i grundfagslæreboøger.

Deres værdi kan diskuteres - i forhold til andre metoder - men de er almindeligt brugt til analyse af mulige sammenhænge i et sæt forsøgsresultater.

Anvendt med fornuft, uden urimelige DATA for regressioner med linie med given hældning / gennem givet punkt, kan de bruges.

Jeg takker lektorerne Knut Conradsen og Poul Thyregod, IMSOR, DTH, for vejledning.

For at lette programmeringen har jeg ændret en række kendte betegnelser en smule.

I programmet benyttes til regression følgende variabler:

- antal : antal punkter ($n = 1$ TO antal).
- $x\$(n)$ og $y\$(n)$: koordinater til punkterne.
- sumx og sumy : summen af hhv x- og y-koordinaterne.
- gnx og gny : gennemsnitsværdier af koordinaterne,
 $gnx = \text{sumx} / n$ hhv $gny = \text{sumy} / n$
- sakx : Summen af Afvigelseernes Kvadrater for x,
 $sakx = \sum (x\$(n) - gnx)^2$
- saky : $saky = \sum (y\$(n) - gny)^2$
- sapxy : Summen af Afvigelseernes Produkt for x og y,
 $sapxy = \sum (x\$(n) - gnx) * (y\$(n) - gny)$.
- sakxf, sakyf, sapxyf
 svarer til sakx, saky, sapxy, idet der dog her benyttes afvigelserne i forhold til det faste punkts koordinater i st.f. gnx, gny.
- xfast og yfast : koordinater til et forudbestemt fast punkt, hvorigennem regressionslinien skal gå.
- xhæld : regressionsliniens hældning y / x .
 xhæld kan være forudbestemt, regression med fast hældning.
- korrel : korrelationskoefficienten, hvilket dog kun er helt korrekt for normal, lineær regression. IMSOR har "forklaringsgrad" som generelt ord, idet korrel^2 (pag. 55, linie 2, udtrykker, hvor stor en del af af y's totale variation, der beskrives af linien. Det, der ikke forklares, er SAK_{res} .

Herudover benyttes følgende variabler, der dog ikke indgår i de nedenfor udledte formler:

`r`, `s`, `rgtal`, `rgnr(100)`, `rgxantal`, `rgyantal`, `pk#`, `rg#`, `valg#`:
Styrer, hvilke og hvor mange punkter af det samlede punktsæt, der indgår i den aktuelle regression.

`udj#(40,24)` og `fjl40`:
Arrays for tekster til udskrivning,
hhv for lagring af udjævningsresultaterne.

`hæld#` : angiver den ønskede regressionstype:

"r" : almindelig, lineær Regression.
"p" : forudbestemt (fast) Punkt.
"h" : forudbestemt (fast) Hældning.
"l" : Lodret linie (gennem `gnx`, 0).
"v" : Vandret linie (gennem 0, `gny`).

Hvis linien,
med de valgte inddelinger af akserne,
på skærmen får hældning > 50 eller < 0.2 ,
skifter programmet automatisk over til
at analysere lodret hhv vandret linie.

Vælg evt. en ny, finere inddeling, så
skærmens begrænsede opløsningsevne kan
illustrere en muligt reelt eksisterende
hældning, der ikke er lodret / vandret.

`xs` : skala, som funktion af de valgte akse-
inddelinger.

`spredx`, `spredy` : den sædvanlige formel for spredningen,
 $\sqrt{(\text{sakx} / (\text{antal} - 1))}$, ...

`xakse`, `yakse`, `xopad`, `xnedad`, `yvenst`, `yhøjre`:
koordinater til regressionsliniens skæring
med akser og figurens begrænsningslinier.

`xparam`, `yparam` : indgår i beregningen af skæringspunkterne.

PROCedurer : `regrestype`, linie 10000.
fejludjævning, linie 11000.
akseparametre (`xparam`, `yparam`), linie 12500.
udtegning, linie 13300.

Det generelle grundlag.

$(\text{korrel})^2 = (\text{SAK}_{\text{TOT}} - \text{SAK}_{\text{RES}}) / \text{SAK}_{\text{TOT}}$, hvor

$\text{SAK}_{\text{TOT}} = \sum (y\$(n) - gny)^2 = \text{saky}$

$\text{SAK}_{\text{RES}} = \sum [y\$(n) - \text{kurve}(n)]^2$, der formuleres som
 $= \sum [y\$(n) - A - B \times (x\$(n) - gnx)]^2$,
 begge FOR $n = 1$ TO antal.

idet der, efter mindste kvadraters metode, søges den bedst mulige, rette linie, defineret som:

$y = A + B \times (x\$(n) - gnx)$.

Lineær regression, almindelig / forudbestemt hældning.

Der søges minimum af

$T = \sum [y\$(n) - A - B \times (x\$(n) - gnx)]^2$
 FOR $n = 1$ TO antal

Heraf fås

$\delta T / \delta A = 2 \times \sum [y\$(n) - A - B \times (x\$(n) - gnx)] \times (-1)$

$\sum y\$(n) - A \times \text{antal} - B \times \text{antal} \times gnx + B \times \text{antal} \times gnx = 0$

$A = \sum y\$(n) / \text{antal} = gny$

dvs RET LINIE GENNEM gnx , gny (uanset om B er forudbestemt)

$\delta T / \delta B = 2 \times \sum [y\$(n) - gny - B \times (x\$(n) - gnx)] \times (-1) \times (x\$(n) - gnx)$

$\sum (y\$(n) - gny) \times (x\$(n) - gnx) - B \times \sum (x\$(n) - gnx)^2 = 0$

$B = \text{sapxy} / \text{sakx} = \text{xhæld}$ for alm. lineær regression.

En linie med forudbestemt hældning, $xhæld$, har ikke
 $\delta T / \delta B = 0$:

$B = \text{xhæld}$ for forudbestemt hældning.

$\text{SAK}_{\text{RES}} = \sum [(y\$(n) - gny) - B \times (x\$(n) - gnx)]^2$

$= \text{saky} + B^2 \times \text{sakx} - 2 \times B \times \text{sapxy}$

$\text{korrel}^2 = (\text{saky} - \text{SAK}_{\text{RES}}) / \text{saky}$

Heraf fås da:

For forudbestemt hældning = $x_{hæld}$:

$$korrel^2 = (2 \times s_{apxy} \times x_{hæld} - s_{akx} \times x_{hæld}^2) / s_{aky}$$

Indsættes $x_{hæld} = s_{apxy} / s_{akx}$, fås

for (normal) lineær regression:

$$korrel^2 = (s_{apxy})^2 / s_{akx} / s_{aky}$$

Linie gennem forudbestemt punkt, x_{fast} , y_{fast}

$$y = y_{fast} + B \times (x_{\#(n)} - x_{fast})$$

$$T = \sum [(y_{\#(n)} - y_{fast}) - B \times (x_{\#(n)} - x_{fast})]^2$$

$$\delta T / \delta B = 2 \times \sum [(y_{\#(n)} - y_{fast}) - B \times (x_{\#(n)} - x_{fast})] \times (-1) \times (x_{\#(n)} - x_{fast})$$

$$\sum (y_{\#(n)} - y_{fast}) \times (x_{\#(n)} - x_{fast}) - B \times \sum (x_{\#(n)} - x_{fast})^2 = 0$$

$$B = s_{apxyf} / s_{akxf} = x_{hæld}$$

$$s_{aky} = SAK_{res}$$

$$= s_{aky} - \sum [(y_{\#(n)} - y_{fast}) - B \times (x_{\#(n)} - x_{fast})]^2$$

$$= s_{aky} - s_{akyf} - B^2 \times s_{akxf} + 2 \times B \times s_{apxyf}$$

$$= s_{aky} - s_{akyf} + s_{apxyf} \times x_{hæld}$$

$$korrel^2$$

$$= (s_{aky} - s_{akyf} + s_{apxyf} \times x_{hæld}) / s_{aky}$$

Udskrift af korrelationskoefficienten

$korrel^2$ er, for almindelig lineær regression, et tal mellem 0 og 1, men det kan blive negativt, ved urimelig brug af fast punkt / fast hældning, fx ved valg af negativ hældning for et punktsæt, der har en tydeligt positiv regressionslinie.

Fast punkt og fast hældning må kun vælges som en "korrektion" af praktiske grunde til det almindelige regressions-resultat.

$korrel$ udskrives overalt som et positivt tal eller 0. Et tal mindre end 0.6 betyder normalt at en anden tilnærmelse end en ret linie vil være bedre, hvis der altså er en korrelation.