

Villy Lymann

~~Rud. Wulffsgade 24 - 2. tv.~~

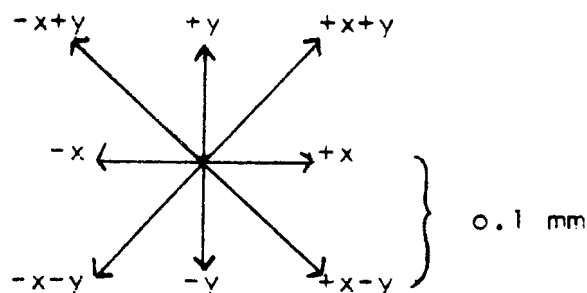
~~DK-8000 Århus C~~

PLOTTERSYSTEMET VED AARHUS UNIVERSITETS REGNECENTER

Generelt om plotteren

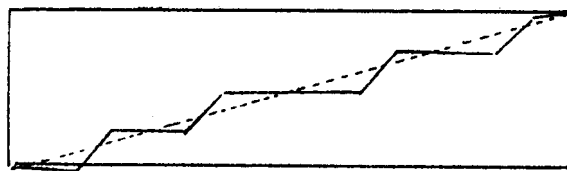
Regnecentret raader over en Calcomp kurvetegner, model 563. Denne er tilsluttet, saa den kan køres direkte af GIER eller styres off-line af et magnetbaand via vor converter.

Enhver tegning paa plotteren bestaar af smaa rette linjestykker af størrelsesorden 0.1 mm. Plotteren kan i en enkelt operation bevæge pennen i forhold til papiret i enten x-aksens eller y-aksens retning eller i begge retninger saaledes:



x-retningen kontrolleres af en stepmotor, der bevæger pennen vandret hen over papiret; y-retningen kontrolleres af en anden stepmotor, der fører papiret frem under pennen.

Da plotteren foruden disse retlignede step kun kan udføre de to operationer PEN OP og PEN NED, maa GIER i almindelighed i stedet for at tegne rette linier lave en zigzag tilnærmelse som f.eks.



Normalt vil man dog ikke kunne se forskellen - dels paa grund af de smaa step og dels paa grund af pennens tykkelse.

Plotterens maksimale hastighed er 300 step pr. sekund, papirets bredde ca. 75 cm. Normalt køres med kuglepenn, men vore operatører kan montere tuschpenne til skønskrift.

Plotteroversætteren

For at lette brugen af plotteren i algol 4-programmer er der lavet en speciel oversætter med navnet pl, dvs. den skal i HJÆLP 3 kaldes ved f.eks.

```
algol,pl<
l,algol,pl,n,l<
```

eller lignende. Oversætteren beslaglægger ekstra ca. 120 celler i ferritlageret og 441 celler i bufferen. Den indeholder foruden de nedenfor beskrevne standardvariable og -procedurer en sekvens til at foretage den tidligere beskrevne zigzag-approximation og eventuelt levere output til en baandstation.

Standardvariable

Integer xxx,yyy;

Disse indeholder til enhver tid pennens position i forhold til origo, maalt i hele step. Origo er pennens position ved start af programmet, men kan ændres enten ved kald af plotaxes eller ved, at man ændrer værdien af xxx eller yyy. Vil man saaledes udraabe pennens øjeblikkelige position til origo gøres dette ved sætningen xxx:=yyy:=0;

Boolean magnetbaand;

Er denne sand, vil alle plotterprocedurer levere output til en baandstation. Den initieres ved sætningen

```
magnetbaand:=ka on;
```

der udføres i det øjeblik, man kalder run.

Hvis man lader plotteroutput gaa til en baandstation og slaar plotteren fra paa krydsfeltet, vil Gier kunne plotte 18cm/sec i stedet for 3cm/sek. Desuden faar man muligheden for flere tegninger, hvis f.eks. pennen løber tør.

boolean station number;

Denne indeholder i de første 10 bits nummeret paa den baandstation, hvortil eventuelt output leveres, og i de næste 10 bits det antal filomarks, der skal overspringes inden skrivning startes. Inden skrivning startes vil maskinen spole baandet tilbage og sikre sig,

at det ikke er et HJÆLP 3-baand eller et SAMBA-baand; I saa fald kommer udskriften ny tape paa skrivemaskinen, og man maa enten montere et andet baand eller, hvis man er sikker paa at man maa skrive paa baandet, taste et c. Ved start udfører oversætteren sætningen:

```
station number:=10 1 10 0;
```

dvs. den vil skrive paa station 1 uden overspringelse af nogen file.

```
real
---- scalex,scaley;
```

Disse er faste skalafaktorer og indeholder længden i centimeter af en enhed i henholdsvis x-aksens og y-aksens retning. Ved start udfører plotteroversætteren sætningen

```
scalex:=scaley:=1.0;
```

svarende til, at en enhed i plotterprocedurerne giver en centimeter paa papiret. Vær forsigtig, hvis pennen ikke staar i (0,0), naar scalex eller scaley ændres.

Standardprocedurer

```
procedure plotline(x0,y0,x1,y1);
value x0,y0,x1,y1; real x0,y0,x1,y1;
```

Denne tegner den (næsten) rette linie fra (x0,y0) til (x1,y1). Hvis første og andet punkt er ens, vil pennen blive flyttet til dette punkt, men ikke blive sænket. Dette kan f.eks. benyttes, hvis man ønsker, at tegningen skal starte 10 centimeter fra papirets venstre kant, saaledes:

```
plotline(-76,0,-76,0); xxx:=0; plotline(10,0,10,0); xxx:=0;
```

```
procedure plotgraph(x0,x1,x,y,dx);
value x0,x1; real x0,x1,x,y,dx;
```

Tegner funktionen $y=y(x)$ i intervallet $[x_0,x_1]$ ved at interpolere lineært mellem funktionsværdierne i x_0,x_0+dx,x_0+2dx ----

Funktionen $y=\sin(1/x)$ kan saaledes tegnes ud i intervallet $[0.01,10]$ ved sætningen

```
plotgraph(0,10,x,sin(1/x),1);
```

eller

```
plotgraph(10,0.01,x,sin(1/x), if x<1 then 0.01 else 0.5);
```

der tegner den anden vej og benytter større nøjagtighed nær nul eller

```
for i:=1 step 1 until 1000 do
  y[i]:=sin( 10/i ); scalex:=0.01;
  plotgraph(1,1000,x,y[x],1.0);
```

Bemærk, at vi ikke kan benytte en integer som dummy variabel, og at steplængden ogsaa skal være af type real.

```
procedure plotcurve(x,y,t,t1,t2,n);
```

```
value t1,t2,n; real x,y,t,t1,t2; integer n;
```

Tegner kurven $x=x(t), y=y(t)$ for t løbende i intervallet $[t1,t2]$ ved at dele dette interval op i n delintervaller og interpolere lineært mellem de fremkomne $n+1$ punkter. En regulær 17kant med centrum i $(8,8)$ og radius 15 centimeter kan laves saaledes:

```
plotcurve(8+15*cos(t),8+15*sin(t),t,0,2*pi,17);
```

```
procedure plotcond(x,y,t,e1,e2,b1,b2);
```

```
real x,y,t,e1,e2; boolean b1,b2;
```

Tegner samme kurve som `plotcurve`, men administrationen er mere generel. $e1$ er startværdien for t ; $e2$ er et udtryk af formen $t+dt$, hvor det opfattes som en skridtlængde, der igen kan afhænge af t . Proceduren vil kun tegne den del af kurven, for hvilken $b2$ er sand, og den hopper ud i det øjeblik $b1$ bliver falsk. Administrationen kan udtrykkes i sætningen:

```
for t:=e1,e2 while b1 do
```

```
  if b2 then TEGN KURVEN;
```

Vi kan tegne den del af den ovenfor beskrevne 17-kant, der ligger i første kvadrant, ved sætningen

```

plotcond(8+15*cos(t),8+15*sin(t),t,
0.0,t+0.1,t<=2*pi,cos(t)>=8/15^sin(t)>=8/15);
procedure plotaxes(x min,x max,y min,y max,boo);
value x min,x max,y min,y max;
integer x min,x max,y min,y max; boolean boo;

```

Tegner et retvinklet koordinatsystem med origo i pennens øjeblikkelige position, x-akse fra x min til x max og y-akse fra y min til y max. Hvis boo er sand, vil plotaxes desuden sætte tal paa aksene. Længden af en enhed er stadig styret af scalex og scaley. Hvis man f.eks. kun vil have tal paa x-aksen, kan det gøres saaledes:

```

plotaxes (-10,10,-5,5,yyy=0);
real procedure plotchar(t,x,y,h);
value t,x,y,h; integer t; real x,y,h;

```

Svarer til writechar. Tegnet med talværdi t tegnes, saa det har nederste venstre hjørne i punktet (x,y). Uppercasetegn faas ved at addere 128 til tegnets talværdi eller ved explicit at kalde plotchar med t=60. h er højden i centimeter af et stort bogstav uanset værdien af scalex og scaley. Værdien af plotchar er x-koordinaten til det næste tegn paa papiret. Et program, der skriver en strimmel ud indtil første CR kan se saaledes ud:

```

x:=0; y:=1;
for t:=1:n while t#64 do x:=plotchar(t,x,y,0.3);
procedure plottext(s,x,y,h,d);
value h,d; real x,y,h,d; string s;

```

Svarer til writetext. s er en textstring, der vil blive tegnet ud (med plotchar) med første karakter i (x,y). h og d er henholdsvis bogstavhøjde og linieafstand i centimeter, uafhængigt af scalex og scaley. Efter et kald af plottext vil (x,y) være koordinaterne til næste karakter paa papiret; deraf følger, at der paa kaldsstedet skal staa simple variable paa x's og y's pladser. Eksempel:

```
x:=5; y:=-3;
plottext(<Jeg græmmes>,x,y,0.3,1.0);
```

```
real procedure plotInteger(t,x,y,h);
```

```
value t,h; integer t; real x,y,h;
```

Enesteforskæl fra plotchar er, at selve talværdien af t tegnes ud. Tallet vil fylde saa mange anslag, som der er cifre i t plus eventuelt et fortegn.

```
integer procedure define symbol(b1,b2);
```

```
value b1,b2; boolean b1,b2;
```

Indsætter et nyt tegn i alfabetet. Et tegn beskrives saaledes: Vi nummerer punkterne (0,0),(0,1).....,(0.5),(1,0).....,(4,5) med tallene fra 0 til 29. Et tegn kan da bestaa af op til 14 linestykker mellem disse 30 punkter. Numrene paa de indgaende punkter pakkes i b1 og b2 med 5 bits pr. nummer. Desuden skal de to første tal angive, hvor mange punkter vi skal igennem, og hvor vi skal starte.

Sætningen:

```
lille e:=define symbol(5 11 5 25 5 18 5 12 5 7 5 8
                    5 15 5 21 5 26 5 19 5 13 5 8);
```

vil bevirke, at beskrivelsen af et e indsættes paa en ledig plads i tabellen over tegn, og i lille e indsættes en pointer. Herefter vil sætningen

```
plotchar(lille e, .....
```

tegne et e. Efter at have udført sætningen

```
KV:=define symbol(5 5 5 0 5 1 5 7 5 6 5 0, false);
```

kan vi ved at variere bogstavhøjden tegne et kvadrat af vilkaarlig størrelse ved hjælp af plotchar. Vil man løfte pennen undervejs, indskyder man det fiktive punkt 31. Et lighedstegn hedder saaledes (5 5 5 8 5 20 5 31 5 21 5 9, false).

procedure drejplot;

Hvert kald af drejplot vil rotere koordinatsystemet 90 grader i positiv retning omkring pennens øjeblikkelige position.

procedure slutplot;

Retter koordinatsystemet op, hvis drejplot har været anvendt. Hvis der har været plotteroutput til en baandstation flyttes pennen helt ud til venstre side, og der sættes et filemark paa baandet. Da operatøren selv kan beslutte at plote via tape - og normalt vil gøre det skal alle plotterprogrammer slutte med at kalde slutplot.

procedure plotstop;

Vil ved off-line plotting medføre, at converteren standser; nyttigt, hvis man vil tegne i flere farver.

Andre procedurer

Foruden de beskrevne procedurer, der alle er bygget ind i plotteroversætter, findes paa disken yderligere to procedurer, der kan kopieres ind i et program ved sætningen

copy plot<

procedure plotfunct(f,x,y,g,h,s,t,m,n,d);

value m,n,d; integer m,n,s,t; real f,x,y,g,h,d;

Denne procedure tegner de konturer, der svarer til, at funktionen $f=f(x,y)$ antager værdierne $0, \pm d, \pm 2d \dots$ x og y , der angiver koordinaterne paa papiret, beregnes ud fra henholdsvis g og h , der igen er funktioner af s og t . Idet s og t gennemløber værdierne fra nul til henholdsvis m og n faar vi saaledes planen indelt i et (ikke nødvendigvis rektangulært) net, indenfor hvis masker plotfunct vil anvende lineær interpolation. Niveaukurver for funktionen $x^2-(y-2)^2$ for x i intervallet $[-5,5]$ og y i intervallet $[0,5]$ faas ved sætningen

plotfunct($x^2-(y-2)^2$,x,y,-5+0.5*x,0.5*t,s,t,21,10,0.5);

procedure plotdim(f,x,y,a,R,u,tværs,n);

value R,a,u,tværs,n; integer n;

real f,x,y,a,R,u; boolean tværs;

Tegner en perspektivisk tegning af funktionen $f(x,y)$ for x og y liggende i intervallet $[-a,a]$. R er afstanden fra øjet til forkanten af dette kvadrat, og u er vinklen mellem denne afstand og planen. Hvis tværs er falsk, tegnes en lodrette snit i funktionen parallelt med x -aksen; er tværs sand, indtegnes også snit vinkelret på x -aksen; Skalafaktorerne i planen justeres, så at kvadratets forkant fylder $2a$ enheder på papiret.

TEGNINGERNE PAA NÆSTE SIDE ER OUTPUT FRA

FØLGENDE PROGRAM:

```
a,pl<
begin integer i,p,q;
  real x,y,t,r,fl;
  copy plot<
  real procedure polær;
  begin
    r:=sqrt(x↑2+y↑2);
    fl:=arctan(y/(x+12));
    polær:=8.0*cos(r)↑2*abs(sin(2*fl+r))/(1.5+r↑(1.3));
  end;
  plotline(-78,0,-78,0); xxx:=0;
  plotline(11,0,11,0);
  plotaxes(-5,5,-5,5,yyy>0);
  x:=-5.0; y:=6.5;
  plottext(†<Demonstration af plotterprocedurer†,x,y,0.3,1);
  x:=-5.0; y:=6.3;
  for i:=1 step 1 until 34 do x:=plotchar(32,x,y,0.3);
  plotgraph(0.0,5.0,x,5.0*exp(-x/2)*sin(2*x),0.1);
  plotcond(-2.5+2.5*cos(t),4.0*sin(t),t,0.0,t+0.04,t≤6.3,
    entier(t*10)mod2=0);
  plotline(0,-10,0,-10); yyy:=0;
  plotfunct(sin(x+y)/(1+x),x,y,p/10.0,q/10.0,p,q,50,40,0.1);
  plotline(0,-9,0,-9); yyy:=0;
  plotdim(polær,x,y,6.0,30.0,0.5,false,150);
  slutplot;
end
t<
```

Demonstration of plotter procedure

