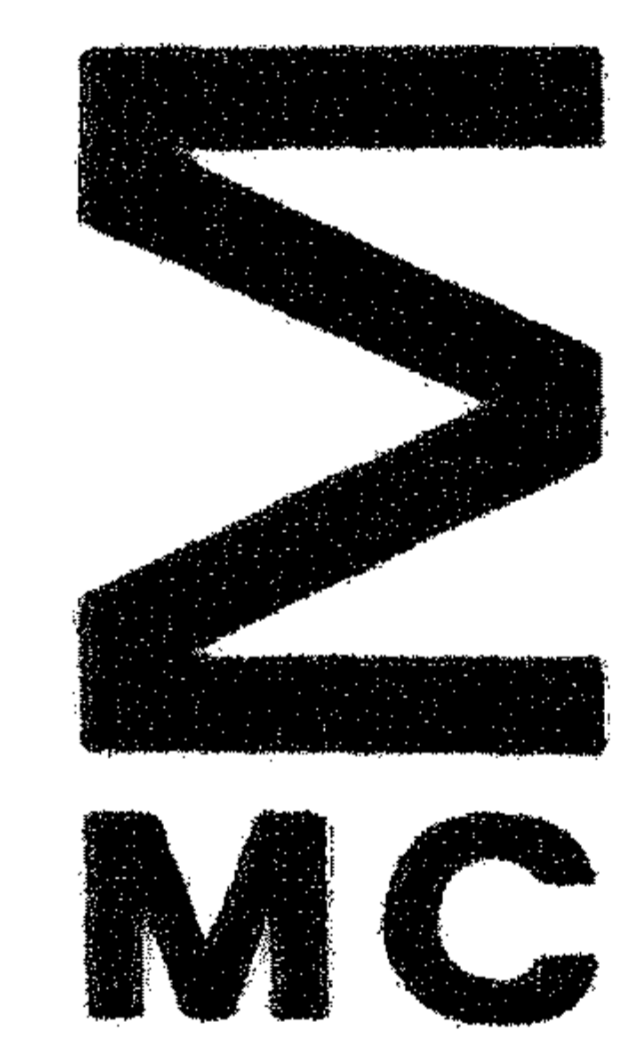


**stichting
mathematisch
centrum**



LR 0.1

OKTOBER 1971

PLOTPROCEDURES

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

framaxval

auteur: A.C. IJsselstein (MC)

Onderwerp: het voorbereiden van een tekenruimte op de plotter, i.e. het
definieren van een kader en het tekenen van een assenkruis;

Gebruiksaanwijzing:

declaratie:

```

procedure framaxval (length, xl, xu, ox, dx, nx, mx, px,
                      height, yl, yu, oy, dy, ny, my, py);
value length, xl, xu, ox, dx, nx, mx, px,
        height, yl, yu, oy, dy, ny, my, py;
real length, xl, xu, ox, dx, nx, mx, px,
        height, yl, yu, oy, dy, ny, my, py;
<procedure body>;

```

parameters:

length, height

:<expression>;

lengte en hoogte in cm. van de tekenruimte, tevens de lengte van
de assen in de x-richting resp. y-richting;
maximumwaarde voor height is 27,25;

xl, yl :<expression>;

ondergrens (in dits) van de x-waarden resp. y-waarden;

xu, yu :<expression>;

bovengrens (in dits) van de x-waarden resp. y-waarden;

ox, oy :<expression>;

coördinaten (in dits) van het snijpunt van de assen in de x-richting
resp. y-richting;

de as in de x-richting, hierna kortweg x-as genoemd, loopt van
(xl,oy) naar (xu,oy) en heeft een lengte van length cm.;

deze as wordt niet getekend als oy < yl of oy > yu (de waarden
van dx, nx, mx en px zijn dan irrelevant);

de as in de y-richting, hierna kortweg y-as genoemd, loopt van (ox,y1) naar (ox,yn) en heeft een lengte van height cm.; deze as wordt niet getekend als $ox < xl$ of $ox > xu$ (de waarden van dy, ny, my en py zijn dan irrelevant);

dx, dy :<expression>;
 onderlinge afstand in dits van de aan te brengen verdelingsstrepen op de x-as resp. y-as; het aanbrengen van deze verdelingsstrepen wordt onderdrukt als $abs(dx) > xu-xl$ resp. $abs(dy) > yu-y1$ (de waarden van nx, mx, px resp. ny, my, py zijn dan irrelevant); als dx resp. dy positief is worden bij deze verdelingsstrepen de bijbehorende waarden ingetekend overeenkomstig de waarden van nx, mx resp. ny, my;
 deze waarde-aanduidingen blijven achterwege (en de waarden van nx, mx resp. ny, my zijn dan irrelevant) als dx en/of dy negatief is;

nx, ny :<expression>;
 aantal cijfers voor de decimale punt in de waarde-aanduidingen bij x-as resp. y-as;

mx, my :<expression>;
 aantal cijfers achter die decimale punt;

px, py :<expression>;
 onderlinge afstand in dits van de aan te brengen onderverdelingsstrepen op de x-as resp. y-as;
 het aanbrengen van deze strepen wordt onderdrukt indien $px \geq dx$ resp. $py \geq dy$;
 deze onderverdelingsstrepen worden niet voorzien van waarde-aanduidingen;

Gebruikte procedures: fixplot, plotframe, move, paper-feed, shape, plot, coord, (LR1.4.6);

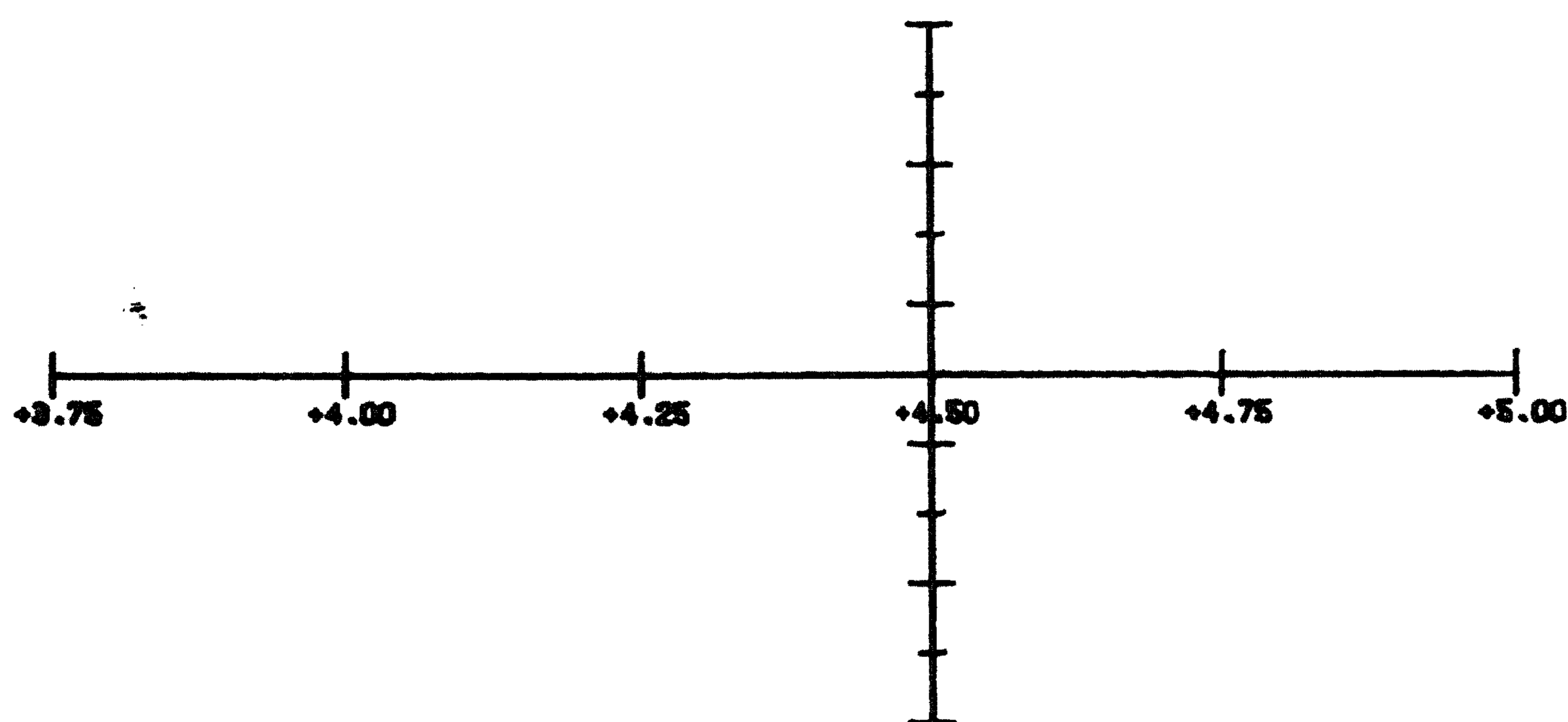
Opmerkingen:

- 1e. bij een aanroep begint de procedure met het aanbrengen van 15cm. paper-feed (vanaf de momentane penpositie) alvorens over te gaan tot het bepalen van het nieuwe kader;

- 2e. teneinde zeker te stellen dat de verdelingsstrepen met bijbehorende waarde-aanduidingen niet buiten het kader komen te vallen alsook de gebruiker ruimte te geven om nader commentaar bij zijn tekening te plotten zal framaxval de met length en height bepaalde tekenruimte met een marge vergroten:
links van x_1 met 6cm. (i.e. $(x_u - x_1) * 6 / \text{length}$ dits),
rechts van x_u met 2cm. (i.e. $(x_u - x_1) * 2 / \text{length}$ dits),
onder y_1 en boven y_u met $(27.94 - \text{height}) / 2$ cm. (i.e. $(y_u - y_1) * (27.94 - \text{height}) / (\text{height} * 2)$ dits);
- 3e. indien de verdeelstrepen zo dicht op elkaar liggen dat de waarde-aanduidingen elkaar overlappen of raken dan wordt het intekenen van die waarden onderdrukt;
- 4e. na afloop van een aanroep van framaxval is de pen van de plotter omhoog;

Voorbeeld van een aanroep:

framaxval (10,3.75,5.00,4.5,+0.25,1,2,3,5,0.0,5.0,2.5,-1,9,7,.5);
levert:



Methoden en prestaties:

Een as wordt getekend m.b.v. de procedure plot. Via plot worden ook de verdelingsstrepen en de onderverdelingsstrepen getekend. Beide soorten markerstrepen staan loodrecht op de as met aan weerszijden van de as een lengte van 1.5 resp. 0.8mm.

De verdelingsstrepen op de x-as zijn gesitueerd in de punten $(x_1+kx*/dx/,oy)$ met $kx = 0,1,2,\dots,[(xu-x_1)/dx]$, de verdelingsstrepen op de y-as zijn geplaatst in $(ox,y_1+ky*/dy/)$ met $ky = 0,1,2,\dots,[(yu-y_1)/dy]$.

De onderverdelingsstrepen op de x-as worden geplaatst in de punten $(x+hx*px,oy)$ met $hx = 1,2,\dots$ en $hx * px < dx$ voor elke $x < xu$ die een abscis is van een verdeelstreep op die x-as.

De onderverdelingsstrepen op de y-as komen in de punten $(ox,y+hy*py)$ met $hy = 1,2,\dots$ en $hy * py < dy$ voor elke $y < yu$ die een ordinaat is van een verdeelstreep op die as.

De waarde-aanduidingen bij de verdeelstrepen worden ingetekend m.b.v. fixplot op een afstand van 0.5mm. van de bijbehorende verdeelstreep. De waarde-aanduidingen bij de x-as staan onder die x-as als $yu-oy \geq oy-y_1$ en anders erboven, die bij de y-as staan links van die y-as als $xu-ox \geq ox-x_1$ en anders rechts ervan.

De cijfertekens staan loodrecht op de x-richting met een hoogte van $h = 1.4\text{mm}$. en een breedte $b = h*4/7$. Wil men grotere cijfers, bijvoorbeeld 2.1mm. hoog, dan is dat te bereiken door in de procedure de statement $\langle h:= 14;\rangle$ in regel 13 te vervangen door $\langle h:= 21;\rangle$.

De breedte van de waarde-aanduiding is $(nx+mx+jx) * b$ mm. met $jx = 2$ als $mx = 0$ en $jx = 3$ als $mx > 0$ resp. $(ny+my+jy) * b$ mm. met $jy = 2$ als $my = 0$ en $jy = 3$ als $my > 0$.

Algol tekst:

Zie volgende twee pagina's.

```

procedure framaxval(length,xl,xu,ox,dx,nx,mx,px,height,yl,yu,oy,dy,ny,my,py);
value length,xl,xu,ox,dx,nx,mx,px,height,yl,yu,oy,dy,ny,my,py;
real length,xl,xu,ox,dx,nx,mx,px,height,yl,yu,oy,dy,ny,my,py;
begin real rx,sx,ry,sy,h,scx,scy,b,k,td,a,tp,s,db,d,pb,t,p;
      boolean nov,nod,nop;
      procedure fixplot0(n,m,z); value n,m,z; integer n,m; real z;
      begin if z=0 then fixplot(n,m,0) else fixplot(n,m,z) end fixplot0;

      plotframe(0,0,1,1,500,2794); move(16); paperfeed(1000);
      length:= length*100; rx:= xu-xl; sx:= rx*200/length;
      height:= height*100; ry:= yu-yl; sy:= ry*(2794-height)/(height*2);
      plotframe( xl-sx*3, yl-sy, xu+sx, yu+sy, length+800, 2794 );
      h:= 14; shape(0,h,0); scx:= plot(0,0,11); scy:= plot(0,0,12); b:= scx*h*4/7; h:= scy*h;

horiz: if oy<yl ∨ yu<oy then goto vertic; plot(xu,oy,2); plot(xl,oy,1);
      nov:= dx<0; dx:= abs(dx); nod:= dx>rx; if nod then goto vertic;
      k:= if mx=0 then 2 else 3; nop:= px>dx; if dx<(nx+mx+k)*b then nov:= true; td:= scy*15;
      a:= if oy*2 < (yu+yl)*(1+1/8) then oy-(td+scy*5+h) else oy+td+scy*5;
      td:= sign(oy-a)*td; tp:= td*8/15; s:= entier((nx+mx+k)/2)*b; db:= xu+dx/4;
      plot(xl,oy+td,2); plot(xl,oy-td,1);
      if ¬ nov then begin coord(xl-s,a,true); fixplot0(nx,mx,xl) end;
      for d:= xl+dx step dx until db do
      begin if nop then goto vd; pb:= d-px/2; t:= +1;
            for p:= d-dx+px step px until pb do
            begin t:= -t; plot(p,oy+t*tp,2); plot(p,oy-t*tp,1) end;
            vd: plot(d,oy+td,2); plot(d,oy-td,1);
            if nov then begin coord(d-s,a,true); fixplot0(nx,mx,d) end
            end;

vertic: if ox<xl ∨ xu<ox then goto ready; plot(ox,yu,2); plot(ox,yl,1);
      nov:= dy<0; dy:= abs(dy); nod:= dy>ry; if nod then goto ready;
      k:= if my=0 then 2 else 3; nop:= py>dy; if dy<h*1.5 then nov:= true; td:= scx*15;
      a:= if ox*2 < (xu+xl)*(1+1/8) then ox-(td+scx*5+(ny+my+k)*b) else ox+td+scx*5;
      td:= sign(ox-a)*td; tp:= td*8/15; s:= h/2; db:= yu+dy/4;
      plot(ox+td,yl,2); plot(ox-td,yl,1);
      if nov then begin coord(a,yl-s,true); fixplot0(ny,my,yl) end;
      for d:= yl+dy step dy until db do
      begin if nop then goto dv; pb:= d-py/2; t:= +1;

```

```
for p:= d-dy+py step py until pb do
begin t:= -t; plot(ox+t*tp,p,2); plot(ox-t*tp,p,1) end;
dv: plot(ox+td,d,2); plot(ox-td,d,1);
if mov then begin coord(a,d-s,true); fixplot0(ny,my,d) end
end;
ready: move(16)
end framaxval;
```

LR 0.1.1

OKT. 1971

1-6

drawcurvesection

auteur: A.C. IJsselstein (MC)

Onderwerp: plotten van krommen $f(x,y) = 0$;

Gebruiksaanwijzing:

declaratie:

```
procedure drawcurvesection (f, xlb, ylb, xub, yub, x, y, dir,  
                             size, tol, mark, del);  
value xlb, ylb, xub, yub, dir, tol, mark;  
integer mark, del;  
real xlb, ylb, xub, yub, dir, size, tol;  
real procedure f;  
<procedure body>;
```

parameters:

f :<procedure identifier>;

de identifier van de door de gebruiker te geven functieprocedure $f(x,y)$;

xlb, ylb, xub, yub

:<expression>;

definieren de rechthoek $W = (xlb \leq x \leq xub) * (ylb \leq y \leq yub)$ waarin het te tekenen deel van de kromme $f(x,y) = 0$ zich bevindt;

dit deel van de kromme dient zowel te beginnen als te eindigen op de rand van deze werkkruimte W en dient verder binnen W continu te zijn zonder dubbelpunten of keerpunten te bevatten;

x, y :<variable>;

coördinaten van de punten van de kromme $f(x,y) = 0$;

ingang: de coördinaten van het beginpunt;

uitgang: de coördinaten van het eindpunt;

dir :<expression>;

geschatte waarde in graden van de hoek - gemeten in tegenwijzer-richting - tussen de positieve x-as en de vector van beginpunt

naar volgende steunpunt;

size :<expression>;

een maat voor de dichtheid van de te berekenen steunpunten, welke gebruikt worden voor de interpolatie in de procedure plotcurve; elk drietal opvolgende steunpunten ligt binnen een vierkantje met zijde size;

tol :<expression>;

een maat voor de nauwkeurigheid waarmee de steunpunten bepaald worden; tol wordt gebruikt als relatieve tolerantie in de nul-puntsprocedure zeroin;

mark :<expression>;

markeert in de tekening (d.m.v. symbolen met een diameter van ca 2mm.) de gebruikte steunpunten volgens onderstaande tabel; bovendien wordt de kromme als streepjeslijn getekend als mark < 0;

abs(mark) = 0 of 1	geen	abs(mark) = 6	Y
2	+	7	X
3	x	8	x
4	*	9	□
5	*	10	◇

del :<expression>;

als del = 0 wordt uitsluitend de kromme getekend, waarbij de steunpunten volgens de waarde van mark gemarkeerd worden;

als del = 1 worden bovendien via de regeldrukker per berekend steunpunt (x,y) de waarden van x,y en f(x,y) afgedrukt, gevolgd door de opmerking "offside" indien (x,y) buiten de werkruimte W ligt;

als del = 2 worden tenslotte nog de relevante delen van de zoekvierkantjes afgebeeld in de tekening en in coördinaten weergegeven via de regeldrukker;

Gebruikte procedures: zeroin (mca 2310, [1], sectie 231);

deplotprocedures plotcurve en plot (LR1.4.6);

de outputprocedures voor de regeldrukker prsym, space, nlcr, printtext en flot (LR1.4.3);

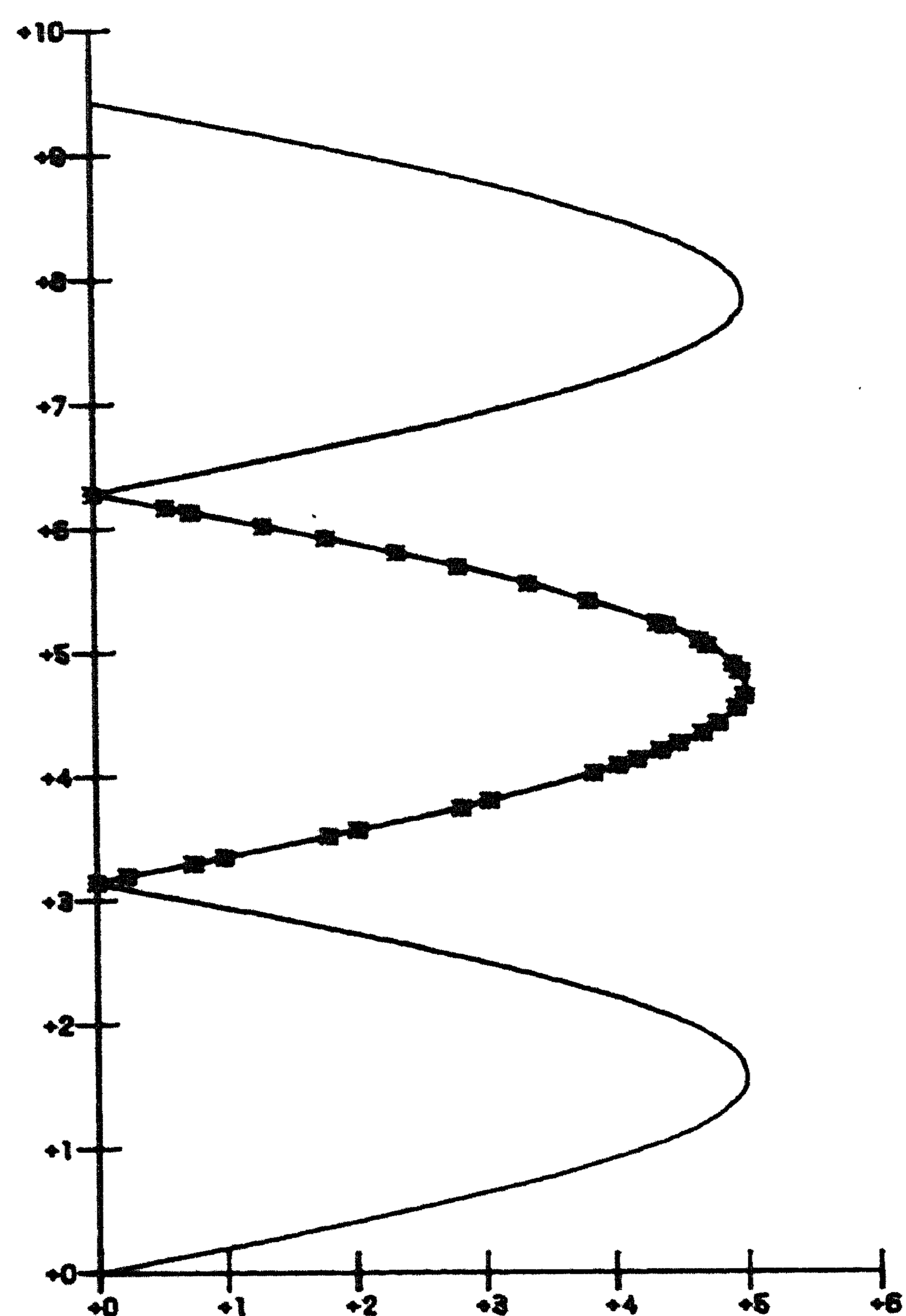
Opmerkingen:

- 1e. alle parameters hebben betrekking op de data space;
- 2e. zoals elke plotopdracht (zie LR1.4.6, inleiding) moet een aanroep van de procedure drawcurvesection voorafgegaan zijn door een definitie van het kader;
hiervoor staan ter beschikking framaxval (LR0.1) en plotframe (LR1.4.6-1.2);
- 3e. bij beëindiging is de pen van de plotter omhoog boven het punt met abscis xmax van het kader en met de ordinaat van het eindpunt van de kromme;

Voorbeeld van een aanroep:

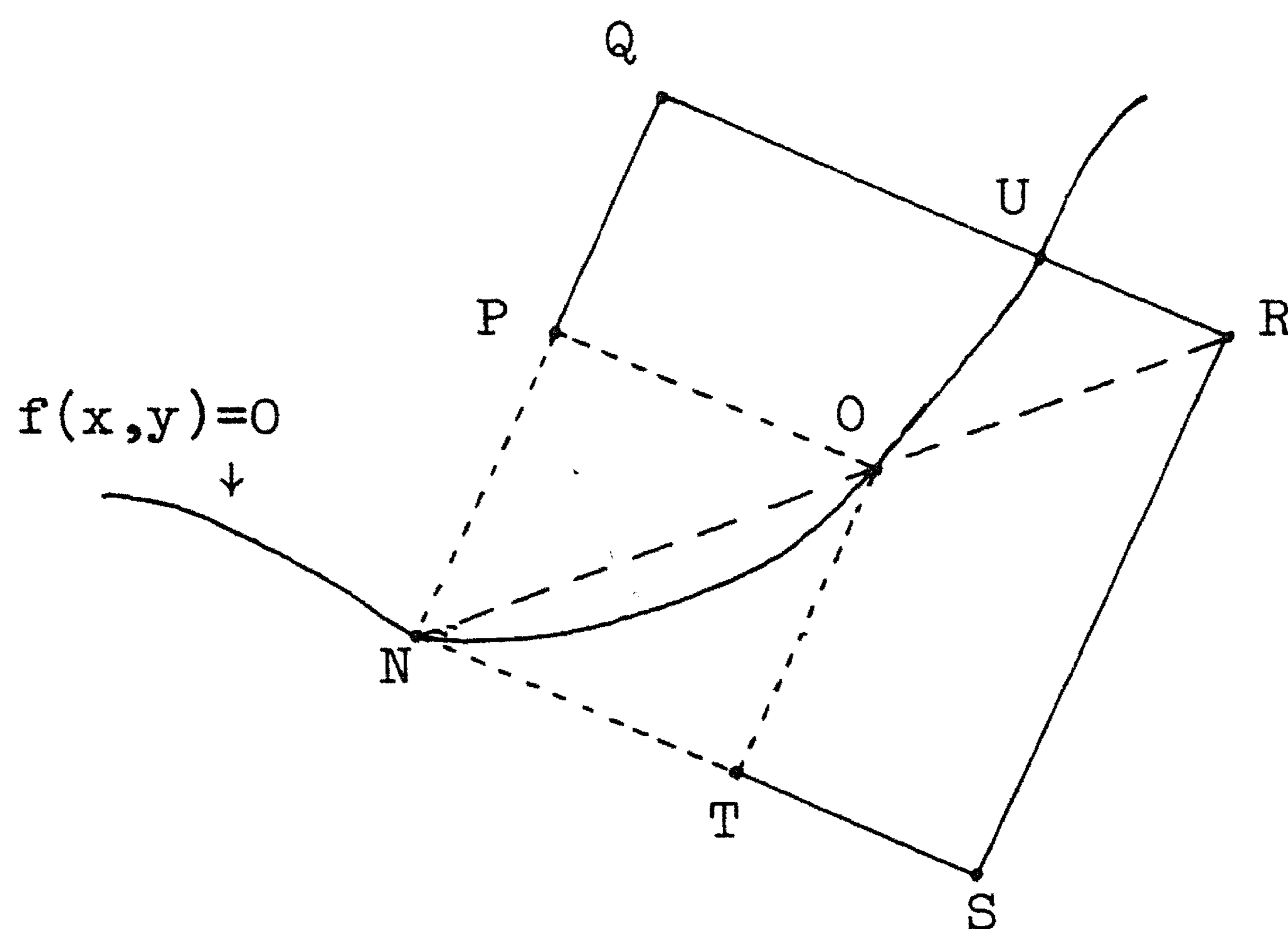
```
framaxval(6,0,6,0,1,1,0,2,10,0,10,0,1,2,0,2);
begin  real procedure g(x,y); value x,y; real x,y; g:= x-abs(sin(y))×5;
      real procedure a(x); value x; real x; a:= if x > 4 then .25 else .75;
      pi:= 3.141592653590;
sect1: x:= 0; y:= 0;
      drawcurvesection(g,0,0,5.5,pi,x,y,20,0.4,108,0,0);
sect2: x:= 0; y:= pi × 2;
      drawcurvesection(g,0,pi,5.5,pi×2,x,y,340,a(x),108,5,0);
sect3: x:= 0; y:= pi × 3;
      drawcurvesection(g,0,pi×2,5.5,pi×3,x,y,340,a(x),108,0,0);
end;
```

levert: $x = +.1009094947109_{10}^{-9}$ en $y = +.6283185307169_{10}^{+1}$ en de tekening welke op de volgende bladzijde staat afgebeeld.



Methoden en prestaties:

De kromme wordt door drawcurvesection getekend via de procedure plotcurve. De aan plotcurve mee te geven steunpunten worden door drawcurvesection zelf gegenereerd aan de hand van de waarde van de parameter size. Uitgaande van de laatste twee gebruikte steunpunten wordt een volgend steunpunt bepaald m.b.v. een zoekvierkantje met zijde size, waarvan een hoekpunt samenvalt met het voorlaatste steunpunt N en waarvan de diagonaal in N door het laatst gebruikte steunpunt O gaat. Op de omtrek van dit zoekvierkantje NQRS wordt - met uitzondering van de stukken NP en NT tussen het hoekpunt N en de projecties P en T van O op de zijden door N - een snijpunt U met de kromme gezocht m.b.v. de nulpuntprocedure zeroin. In nevenstaande tekening is dus PQRST het relevante deel van de omtrek waarop U gezocht wordt.

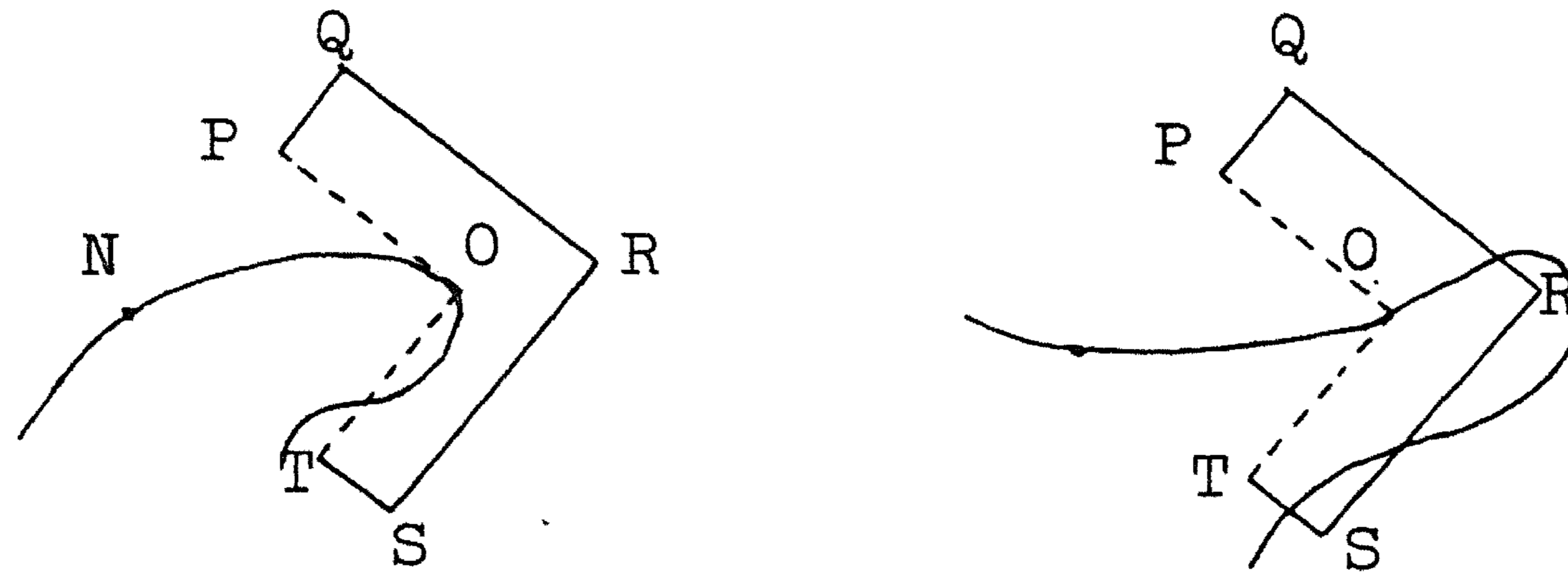


Indien het gevonden punt U buiten de werkruimte $W = (x_{lb} < x < x_{ub}) * (y_{lb} < y < y_{ub})$ ligt wordt een ander steunpunt gezocht middels een zoekrechthoekje binnen W waarvan een zijde samenvalt of evenwijdig is met de overschreden zijde van W. Ligt het punt U niet buiten W dan wordt U aan plotcurve meegegeven. De tekening wordt beëindigd als U - rekening houdend met de gebruikte tol - op de rand van W ligt. Het hierboven beschreven proces van genereren wordt, na de doorschuivingen $N \rightarrow O$ en $O \rightarrow U$, herhaald als U binnen W ligt.

Bij het starten van dit proces is nog geen tweede punt van de kromme bekend. Dit wordt door drawcurvesection ondervangen door de diagonaal te bepalen uit de parameter dir, een fictief tweede punt te nemen (n.l. dat punt op de diagonaal dat op een afstand $size/\sqrt{2}$ van het beginpunt ligt) en een zoekvierkantje te hanteren met zijdelengte $size/\sqrt{2}$. Indien het dan gevonden tweede steunpunt van de kromme op de rand van W ligt dan eindigt drawcurvesection met de foutmelding "second point on border, drawcurvesection fails".

Voor een goed functioneren van dit proces is het noodzakelijk dat er steeds een eenduidig bepaald snijpunt U van een traject PQRST met de kromme $f(x,y) = 0$ bestaat. De gebruiker dient dus de parameter size (locaal) zeker zo klein te kiezen dat situaties als hieronder afgebeeld zich niet voordoen. Indien er geen snijpunt van een traject PQRST met de kromme bepaald kan

worden dan stopt drawcurvesection het hele proces en vermeldt via de regel-
drukker "drawcurvesection fails".



Verder zij nog opgemerkt dat een dubbelpunt of een keerpunt in een kromme geen probleem vormt indien er werkruimten W gekozen worden zó dat zo'n punt slechts op de randen van die ruimten voorkomt. Dit opsplitsen van de kromme $f(x,y) = 0$ in secties m.b.v. werkruimten kan ook worden toegepast om variatie aan te brengen in de parameters size, tol of mark.

Literatuur:

- [1] T.J. Dekker and W. Hoffmann, ALGOL60 procedures in numerical algebra, part 2, 2nd edition (Amsterdam, Mathematical Centre Tracts 23, 1971).

Algol tekst:

zie volgende vier pagina's.

```

procedure drawcurvesection(f,xlb,ylb,xub,yub,x,y,dir,size,tol,mark,del);
value xlb,ylb,xub,yub,dir,tol,mark; integer mark,del; real xlb,ylb,xub,yub,x,y,dir,size,tol;
real procedure f;
begin integer sfp,sfq,sfr,sfs,sft,j,num,rev,i,dirx,diry;
real dns,dnq,s1,co,dnr,a1,a2,aa,p1,p2,fp,b1,b2,bb,t1,t2,ft,c1,c2,cc,q1,q2,fq,r1,r2,fr,
s1,s2,fs,u,eps,tol1,tol2,tol3,tol4,x1,y1,siz,x2,y2,x3,y3,s,bound,x1f,y1f,x2f,y2f,
dx,dy,difx,dify,z;
boolean bviax,cviax,found,movx,movy,xb,yb,tcx,tcy;

boolean procedure zx(stx,sty,finx,tg); value stx,sty,finx,tg; real stx,sty,finx,tg;
begin x:=stx; y:=sty; zx:=zeroin(x,finx,f(x,y+(x-stx)*tg),tol*abs(x)+eps);
y:=sty+(x-stx)*tg
end zx;
boolean procedure zy(stx,sty,finy,tg); value stx,sty,finy,tg; real stx,sty,finy,tg;
begin y:=sty; x:=stx; zy:=zeroin(y,finy,f(x+(y-sty)*tg,y),tol*abs(y)+eps);
x:=stx+(y-sty)*tg
end zy;

boolean procedure pqrst(n1,n2,o1,o2,dns,dnq);
value n1,n2,o1,o2,dns,dnq; real n1,n2,o1,o2,dns,dnq;
begin dnr:=sqrt(dns*dns+dnq*dnq); a1:=o1-n1; a2:=o2-n2; aa:=sqrt(a1*a1+a2*a2);
if dns=dnq ^ aa>dnr*0.9 then begin dnr:=aa*1.12; dns:=dnq:=dnr/sqrt(2) end;
a1:=a1/aa; a2:=a2/aa; s1:=dns/dnr; co:=dnq/dnr;
b1:=+a1 * co - a2 * s1; b2:=+a1 * s1 + a2 * co; bviax:=abs(b1) > abs(b2);
c1:=+a1 * s1 + a2 * co; c2:=-a1 * co + a2 * s1; cviax:=abs(c1) > abs(c2);

p1:=n1 + aa * co * b1; p2:=n2 + aa * co * b2; fp:=f(p1,p2); sfp:=sign(fp);
q1:=n1 + dnq * b1; q2:=n2 + dnq * b2; fq:=f(q1,q2); sfq:=sign(fq);
r1:=n1 + dnr * a1; r2:=n2 + dnr * a2; fr:=f(r1,r2); sfr:=sign(fr);
s1:=n1 + dns * c1; s2:=n2 + dns * c2; fs:=f(s1,s2); sfs:=sign(fs);
t1:=n1 + aa * s1 * c1; t2:=n2 + aa * s1 * c2; ft:=f(t1,t2); sft:=sign(ft);

if del=2 then
begin plot(p1,p2,2); plot(q1,q2,1); plot(r1,r2,1); plot(s1,s2,1); plot(t1,t2,1);
plot(n1,n2,2); nlcr; for j:=25,26,27,28,29 do begin space(23); prsym(j) end;
nlcr; space(10); printtext({x:});
for u:=p1,q1,r1,s1,t1 do begin space(3); flot(13,3,u) end;
nlcr; space(10); printtext({y:});

```

```

    for u:= p2,q2,r2,s2,t2 do begin space(3); flot(13,3,u) end;
    n1cr; space(10); printtext(⟨f:⟩);
    for u:= fp,fq,fr,fs,ft do begin space(3); flot(13,3,u) end
end;

pqrst:=
if sfp ≠ sfq then (if bviax then zx(p1,p2,q1,b2/b1) else zy(p1,p2,q2,b1/b2)) else
if sfq ≠ sfr then (if cviax then zx(q1,q2,r1,c2/c1) else zy(q1,q2,r2,c1/c2)) else
if sfr ≠ sfs then (if bviax then zx(r1,r2,s1,b2/b1) else zy(r1,r2,s2,b1/b2)) else
if sfs ≠ sft then (if cviax then zx(s1,s2,t1,c2/c1) else zy(s1,s2,t2,c1/c2)) else
false
end pqrst;

integer procedure check; check:=
( if x-xlb>tol1 ∧ y-ylb>tol2 ∧ xub-x>tol3 ∧ yub-y>tol4 then +1 else
  if xlb-x>tol1 ∨ ylb-y>tol2 ∨ x-xub>tol3 ∨ y-yub>tol4 then -1 else 0 );

boolean procedure qrsok;
begin if r1<xlb ∨ r1>xub ∨ r2<ylb ∨ r2>yub then begin x3:= r1; y3:= r2 end else
  if q1<xlb ∨ q1>xub ∨ q2<ylb ∨ q2>yub then begin x3:= q1; y3:= q2 end else
  if s1<xlb ∨ s1>xub ∨ s2<ylb ∨ s2>yub then begin x3:= s1; y3:= s2 end else goto ok;
  qrsok:= false; goto fin;
ok: qrsok:= true;
fin:
end qrsok;

procedure pout; if del>0 then
begin n1cr; prsym(98); flot(13,3,x); prsym(87); flot(13,3,y); prsym(99);
  printtext(⟨ with f= ⟩); flot(13,3,f(x,y))
end pout;

start: eps:= 10-10; dir:= dir×3.14159265359/180;
  tol1:= tol×abs(xlb) + eps; tol2:= tol×abs(ylb) + eps;
  tol3:= tol×abs(xub) + eps; tol4:= tol×abs(yub) + eps;
firstp: x1:= x; y1:= y; num:= 1; if del>0 then n1cr; pout;
secondp: siz:= size/sqrt(2); x2:= cos(dir)×siz + x1; y2:= sin(dir)×siz + y1;
  rev:= -1; found:= pqrst(x1,y1,x2,y2,siz,siz);
  if found then begin i:= check; if i=1 then goto sp; if i=0 then go to border end;

```



```

goto if qrsok then out else prep;
sp: x2:= x; y2:= y; num:= 2; pout;

nextp: rev:= +1; found:= pqrst(x1,y1,x2,y2,size,size);
      if !found then goto (if qrsok then out else prep);
np: x3:= x; y3:= y; num:= num+1; pout; i:= check; if i=-1 then goto offside;
      if num=3 then begin plotcurve(0,mark,1); plotcurve(x1,y1,2); plotcurve(x2,y2,2) end;
      if i=0 then goto border;
inside: plotcurve(x3,y3,2); x1:= x2; y1:= y2; x2:= x3; y2:= y3; goto nextp;
border: if num = 1 then begin n1cr; printtext({second point on border,}); goto out end;
      plotcurve(x3,y3,2); goto finish;
offside: if del>0 then printtext({offside}); num:= num-1;

prep: s:= if num=1 then siz else size; xb:=yb:= false;
      movx:= abs(x2-x1)<2*(tol*abs(x1)+eps); movy:= abs(y2-y1)<2*(tol*abs(y1)+eps);
      dirx:= sign(x2-x1); tcx:= dirx=0; diry:= sign(y2-y1); tcy:= diry=0;
      if x1b-x3>tol1 then begin xb:= true; bound:= x1b; goto xbound end else
      if x3-xub>tol3 then begin xb:= true; bound:= xub; goto xbound end else
      if y1b-y3>tol2 then begin yb:= true; bound:= y1b; goto ybound end else
      if y3-yub>tol4 then begin yb:= true; bound:= yub; goto ybound end;

xbound: if tcx then dirx:= sign(bound-x1)*rev;
      if diry=0 then diry:= if yub-y1 < y1-y1b then +1 else -1;
      x1f:= if movx then x1*(1-dirx*2*tol) else x1; difx:= abs(x2-x1f);
      dx:= if dirx= sign(bound-x1f) then abs(bound-x1f) else s;
      y1f:= if movy then y1*(1-diry*2*tol) else y1; dify:= abs(y2-y1f);
      dy:= if dify<s*0.9 then s else dify*1.12; z:= y1f + diry * dy;
      dy:= if z<y1b then y1f-y1b else if z>yub then yub-y1f else dy;
      x2f:= if difx>dify then x2 else x1f + dirx * dify * dx/dy;
      y2f:= if difx>dify then y1f + diry * difx * dy/dx else y2; goto goon;

ybound: if tcy then diry:= sign(bound-y1)*rev;
      if dirx=0 then dirx:= if xub-x1 < x1-x1b then +1 else -1;
      y1f:= if movy then y1*(1-diry*2*tol) else y1; dify:= abs(y2-y1f);
      dy:= if diry= sign(bound-y1f) then abs(bound-y1f) else s;
      x1f:= if movx then x1*(1-dirx*2*tol) else x1; difx:= abs(x2-x1f);
      dx:= if difx<s*0.9 then s else difx*1.12; z:= x1f + dirx * dx;
      dx:= if z<x1b then x1f-x1b else if z>xub then xub-x1f else dx;

```

```

y2f:= if dify>difx then y2 else y1f + diry × difx × dy/dx;
x2f:= if dify>difx then x1f + dirx × dify × dx/dy else x2;

goon: if dirx×diry > 0 then begin dns:= dx; dnq:= dy end else begin dns:= dy; dnq:= dx end;
found:= pqrst(x1f,y1f,x2f,y2f,dns,dnq); if found then goto (if num=1 then sp else np);
if num=1^rev=-1 then rev:=+1 else if num>1^rev=+1 then rev:=-1 else goto out;
try: if xb^tcx then goto xbound; if yb^tcy then goto ybound;

out: x:= if num=1 then x1 else x2; y:= if num=1 then y1 else y2;
nlcr; printtext(⌘drawcurvesection fails⌘); nlcr;
finish: if del>0 then nlcr; if num>2 then plotcurve(x,y,3);
plot(plot(0,0,13),plot(0,y,6),-2);
end drawcurvesection;

```

LR 0.1.2

OKT. 1971

2-10