Het tekenen van een cirkel

H.A. Lauwerier
M.I., U.v.A., Amsterdam

Samenvatting

De meetkunde is bezig aan een come-back. Niet alleen door de ruimtemeetkunde in wiskunde-B, maar zeker ook door de mogelijkheden die de computer kan bieden. Dat daarbij de ene cirkel de andere niet is blijkt uit dit artikel. De auteur ziet goede mogelijkheden om computerexperimentjes en theorie met elkaar af te wisselen.

Vroeger tekenden we een cirkel m.b.v. een passer, al of niet voorzien van hulpstukken. Leerlingen konden met goed gereedschap èn de nodige technische vaardigheid, fraaie uit cirkels opgebouwde tekeningen maken, maar dat had meer met het vak tekenen dan met wiskunde te maken.

Een echte wiskundige behoefde het niet zo nauw te nemen. Voor hem was de cirkel een abstracte figuur, gevormd door een verzameling punten, een oneindig dunne lijn waarvan elk punt evenver lag van een gegeven punt, een middelpunt.
Wat zich op papier of schoolbord bevond was maar bijzaak, een geheugensteuntje. Met dit soort verhalen wist menig wiskundeleraar zijn gebrek aan technische tekenvaardigheid te verhullen. Gelukkig viel er ook weinig te tekenen omdat meetkunde als zelfstandig vak de facto vele jaren afgeschaft was.
Maar de tijden veranderen. Meetkunde is terug van weggeweest. We luisteren niet meer zo vaak naar de radio maar kijken des te meer naar de televisie. De computers dringen steeds meer naar voren. Muziek wordt voortaan gedigitaliseerd en via compact disc en computer tot klinken gebracht. Op dezelfde wijze worden televisiebeelden verwerkt en soms op een surrealistische wijze, die doet denken aan computer art, getransformeerd. Voor het wiskunde-onderwijs en voor toepassers van wiskunde is nu de microcomputer het universele tekeninstrument, waarmee meetkundige constructies in gedigitaliseerde vorm zichtbaar gemaakt kunnen worden.
In beginsel kan een computer niet veel meer doen dan een recht lijntje trekken tussen twee in Cartesische coördinaten gegeven punten. Dat is vooral goed te zien wanneer de computer gebruikt wordt door een plotter aan te sturen. De dikte van de getekende lijn is daarbij in de regel iets als 0.3 mm. Wat we op het beeldscherm te zien krijgen is een lichtlijn opgebouwd uit zg. pixels (pictural elements). Het beeldscherm is opgebouwd uit een fijnmazig rooster van beeldpunten, b.v. 400 horizontaal bij 850 verticaal, waarbij elk beeldpunt als een klein lampje aan of uit kan staan. Bevat het beeldscherm veel beeldpunten, hetgeen uitgedrukt wordt met de term 'een groot oplossend vermogen', dan ziet een cirkel er op het beeldscherm, althans op enige afstand, heel acceptabel uit.

Als echte wiskundigen trekken we ons van de technische onvolkomenheden maar niet te veel aan. We veronderstellen slechts dat onze computer in staat is een recht lijntje te tekenen tussen twee punten met opgegeven coördinaten (x_1, s_1) en (x_2, s_2). In een vorig artikel [1] heb ik laten zien dat we aan de gebruikelijke Basic-opdrachten slechts een drietal grafische opdrachten behoeven toe te voegen om (bijna) alles te kunnen tekenen. (In ons geval een H.P. 86.)

De eerste opdracht is niet zo essentieel maar dient slechts het gemak.

scale a, b, c, d

wil zeggen dat op het beeldscherm een dusdanig coördinatenstelsel is aangebracht dat linker- en rechterrand corresponderen met $x = a$, $x = b$ en dat onder- en bovenrand gegeven zijn door $y = c$, $y = d$.

De tweede opdracht:

move p, q

wil zeggen dat de elektronische tekenpen of de plotterpen in de positie (p, q) gebracht wordt.

De derde opdracht:

draw r, s

tekent dan en recht lijntje naar het punt (r, s).

Hoe tekenen we nu een cirkel wanneer we alleen rechte lijntjes kunnen tekenen. Het antwoord is natuurlijk dat we de cirkel vervangen denken door een regelmatige veelhoek met een groot aantal, b.v. 400 zijden en dat we die veelhoek presenteren als imitatie van de cirkel. Gebruiken we een plotter op A_4-schaal dan is het resultaat aldus volkomen bevredigend.
Is de cirkel in Cartesische coördinaten bepaald door, zo eenvoudig mogelijk, de vergelijking:

$$x^2 + y^2 = 1$$

dan kunnen we de hoekpunten van een ingeschreven regelmatige n-hoek vastleggen door:

$$x = \cos(kh), \qquad y = \sin(kh),$$

waarbij $h = 2\pi/n$ en $k = 0(1)n$.

Het volgende programma zou dus een cirkel moeten geven:

```
10  scale −1, 1, −1, 1
20  h = pi/200
30  move 1, 0
40  for k = 1  to  400
50  x = cos(k * h)  : y = sin(k * h)
60  draw x, y
70  next k
80  end
```

Wie het program op deze wijze, uiteraard met een paar kleine aanpassingen, uitvoert ziet misschien op het beeldscherm geen cirkel verschijnen maar een ellips. De reden is dat dan op het beeldscherm voor de horizontale en voor de verticale afmetingen een verschillende schaling is gebruikt. Is het beeldscherm vierkant dan is er niets aan de hand, maar is het beeldscherm b.v. een liggende rechthoek met een verhouding 4:3, dan kan opdracht 10 vervangen worden door:

scale −4/3, 4/3, −1, 1

Een andere mogelijkheid is dat we de x-schaal verkorten in de opdrachten 30 en 50:

```
move 3/4, 0
x = (3/4) * cos (k * h) enz.
```

De computer rekent wel snel en de meeste tijd gaat meestal zitten in de uitvoering van de grafische opdrachten. We hebben dus niet zo'n behoefte het bovenstaande programma te verbeteren en te bekorten. Maar eenvoudiger kan het wel. Daartoe gebruiken we de transformatie-formules.

$$(1) \qquad T \left\{ \begin{array}{l} x' = x\cos\alpha - y\sin\alpha \\ y' = x\sin\alpha + y\cos\alpha \end{array} \right.$$

welke de rotatie van een punt P om de oorsprong O over de hoek α in positieve draaizin beschrijven. Die formules zijn van essentieel belang voor de computermeetkunde en dienen tot de kern van het moderne meetkunde-onderwijs te behoren.
Een eenvoudige afleiding ervan geven we in de appendix.

In het computerprogramma stellen we ons tot taak een regelmatige veelhoek $P_0, P_1, P_2 \ldots$ te tekenen als surrogaat voor de cirkel. Elk punt P_k kan uit het voorafgaande punt P_{k-1} verkregen worden door een kleine rotatie, d.w.z. T waarbij α een kleine hoek is als $\pi/100$.
Het programma kan er dan als volgt uitzien:

```
 10  gclear
 20  scale −1, 1, −1, 1
 30  a = cos (pi/100)  :  b = sin (pi/100)
 40  x = 1  :  y = 0
 50  move 1, 0
 60  for k = 1  to  200
 70  z = x
 80  x = a * x − b * y
 90  y = b * z + a * y
100  draw x, y
110  next k
120  end
```

Het programma spreekt voor zich. De transformatie T is in de regels 80 en 90 als:

$$(2) \qquad T \left\{ \begin{array}{l} x' = ax - by \\ y' = bx + ay \end{array} \right.$$

Omdat bij de berekening van y' gebruik wordt gemaakt van de vorige waarde van x, inmiddels vervangen door x', moet die oude waarde van x als z in regel 70 even bewaard worden. In regel 30 zijn de waarden van a en b als:

$$a = \cos\alpha, \quad b = \sin\alpha$$

van te voren berekend.
Wanneer we hetzelfde programma uitvoeren met een grotere hoek α kunnen we fraaie regelmatige figuren krijgen. Nemen we b.v.:

$$\alpha = 2\pi p/q,$$

waarbij p en q onderling ondeelbare natuurlijke getallen zijn, dan krijgen we in het algemeen een diagonalenpatroon van een regelmatige veelhoek met q hoekpunten, een sterveelhoek. Aldus in fig. 1 het resultaat van het programma met $p = 11$ en $q = 32$. (Zie volgende pagina).

De aanleiding tot het schrijven van dit stukje was een gesprek met een computerfanaat die me vertelde voor het tekenen van een cirkel de transformatie (2) te gebruiken met de waarden $a=1$ en $b=0.01$ (of kleiner). Met de kennis die we inmiddels verworven hebben weten we dat die methode niet deugt of althans onnauwkeurig is omdat $a^2+b^2>1$ is.

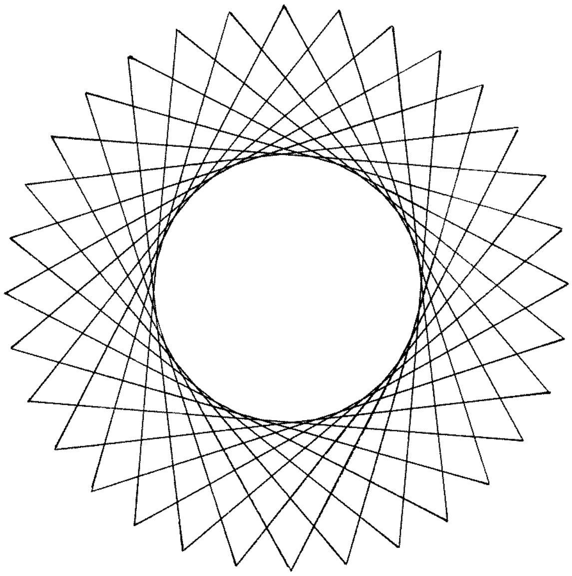

fig. 1

```
10 REM ***STERVEELHOEK***
20 RAD
30 GCLEAR
40 SCALE -1,1,-1,1
50 DISP "KIES P,Q"
60 INPUT P,Q
70 A=COS (2*P*PI /Q) @
   B=SIN (2*P*PI /Q)
80 X=1 @ Y=0
90 MOVE 1,0
100 FOR K=0 TO Q
110 Z=X
120 X=A*X-B*Y
130 Y=B*Z+A*Y
140 DRAW X,Y
150 NEXT K
160 END
```

Programma fig. 1

Laten we de transformatie (2) nog eens toepassen op het punt $P(x,y)$ met als resultaat $P'(x',y')$. Een kleine berekening geeft:

$$OP'^2 = x'^2+y'^2 = (ax-by)^2+(bx+ay)^2 = (a^2+b^2)(x^2+y^2)$$

zodat:

$$(3) \qquad OP' = OP\sqrt{a^2+b^2}.$$

Wanneer dus $\sqrt{a^2+b^2}>1$ beweegt de puntenrij $P_0, P_1, P_2\ldots$ over een spiraal. Met een klein beetje analyse kunnen we een schatting van de fout maken bij het tekenen van een cirkel wanneer $a=1$ en b erg klein is. Voor elke iteratiestap is T, bijna, een rotatie over de hoek α bepaald door $b=\sin\alpha$. Omdat α erg klein is geldt bij benadering $\sin\alpha\approx\alpha$ zodat $\alpha=b$. Voor een complete cirkel hebben $2\pi/\alpha$, d.w.z. $2\pi/b$, iteratiestappen nodig.
Bij elke stap is volgens (3) de straal OP met een factor $\sqrt{1+b^2}$ toegenomen. Totaal dus:

$$(1+b^2)^{2\pi/b} \approx 1+b\pi.$$

Dat betekent dat we voor $b=0.01$ ongeveer 628 stappen nodig zouden hebben en dat we theoretisch toch een fout van drie procent (0.01π) moeten verwachten. Op het beeldscherm van een computer zal dat niet zo opvallen, maar wanneer de tekening m.b.v. een plotter op tekenpapier wordt uitgevoerd zien we duidelijk dat we in werkelijkheid een spiraal hebben getekend (zie fig. 2).

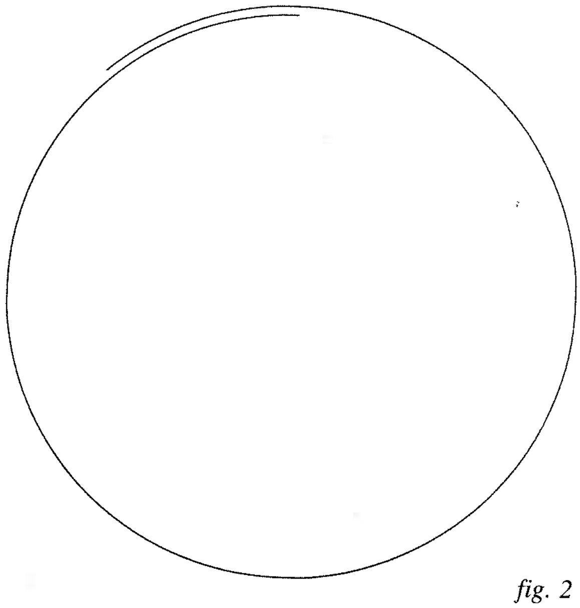

fig. 2

```
10 GCLEAR
20 SCALE -1.2,1.2,-1.2,1.2
30 A=1 @ B=.01
40 X=1 @ Y=0
50 MOVE X,Y
60 FOR K=0 TO 700
70 DRAW X,Y
80 Z=X
90 X=A*X-B*Y
100 Y=B*Z+A*Y
110 NEXT K
120 END
```

Programma fig. 2

De conclusie is dat het heel verstandig is in het onderwijs computerexperimenten te laten afwisselen met kleine stukjes theorie. Men kan b.v. beginnen met een experimenteel gebruik van de transformatie T in de vorm (2). Goniometrie is daarbij voorlopig niet nodig. Het eerste experiment (2) is de baan te bepalen van een ander beginpunt P_0, d.w.z. de puntenrij $P_0, P_1, P_2, \ldots$ waarbij elk volgend punt uit het vorige wordt verkregen door de transformatie T toe te passen.

We gebruiken daartoe het volgende programma, een variant van het vorige programma:

```
10   gclear
20   scale −1, 1, −1, 1
30   disp "kies a, b"
40   input a, b
50   x = 0.5  :  y = 0
60   for k = 1 to 1000
70   move x, y  :  draw x, y
80   z = x
90   x = a * x − b * y
100  y = b * z + a * y
110  next k
120  end
```

In deze versie kunnen de coëfficiënten a, b op regel 40 via het toetsenbord worden ingelezen. De repeterende cyclus 60–110 bevat in regel 70 de opdracht om het k-de baanpunt op het scherm af te beelden. Het beginpunt ($\frac{1}{2}$, 0) op regel 50 kan natuurlijk vervangen worden door elk ander beginpunt. Kiezen we om te beginnen a = 0.98 en b = 0.25 dan krijgen we een fraaie spiraal waarvan de beeldpunten zich van de oorsprong verwijderen. Hadden we a = 0.98 en b = 0.15 gekozen dan vinden we een naar binnen draaiende spiraal (zie fig. 3).

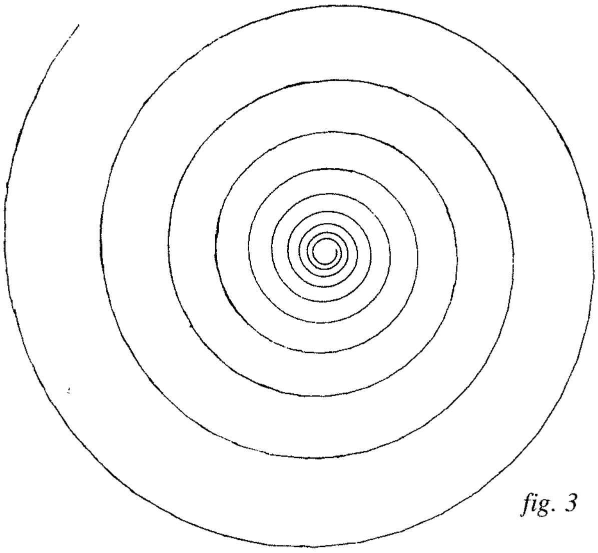

fig. 3

```
10 GCLEAR
20 SCALE -1,1,-1,1
30 A=.98 @ B=.15
40 X=1 @ Y=0
50 MOVE X,Y
60 FOR K=0 TO 400
70 DRAW X,Y
80 Z=X
90 X=A*X-B*Y
100 Y=B*Z+A*Y
110 NEXT K
120 END
```

Programma fig. 3

Het experiment kan nu vervolgd worden met een stukje theorie waarin duidelijk wordt gemaakt hoe de vorm van de spiraal door a en b bepaald is. We weten nu dat dit afhangt van $\sqrt{a^2 + b^2}$ en dat voor $a^2 + b^2 = 1$ alle punten op een cirkel liggen.

Het volgende experiment betreft een keuze van a en b waarvoor $a^2 + b^2 = 1$. Een voor de hand liggend voorbeeld is a = 0.8, b = 0.6. Als beginpunt kunnen we nu x = 1, y = 0 kiezen. Laten we de computer op een lage snelheid lopen zodanig dat we de puntenrij $P_0, P_1, P_2 \ldots$ goed kunnen volgen, dan constateren we dat na 166 stappen de cirkel gelijkmatig met punten gevuld is. De puntenrij is dan 17 maal rondgegaan. Men kan zich afvragen hoe dit verschijnsel, en soortgelijke verschijnselen, bij een andere keuze van a en b te verklaren of te voorspellen is. Dat lukt al met een minimum aan goniometrie. Aanvaarden we het door computerexperimenten ondersteunde feit dat T in de vorm (2) een rotatie beschrijft dan laten we (zie fig. 4) de transformatie werken op het punt P(1, 0).

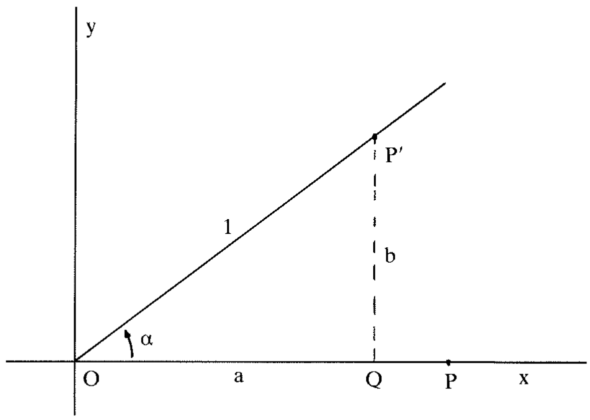

fig. 4

We lezen dan af dat de rotatiehoek bepaald is door:

$a = \cos \alpha$

of liever nog door:

$b = \sin \alpha$.

Voor b = 0.6 is dan α = 0.6435 of 36.87°. Voor elke 10 stappen gaat het beeldpunt iets meer dan de cirkel rond. Per stap is dit:

$$\frac{\alpha}{2\pi} = 0.102416..$$

gedeelte van de cirkelomtrek. Anderzijds is:

$$\frac{17}{166} = 0.102410..$$

een zeer goede benadering van het onmeetbare getal $\frac{\alpha}{2\pi}$. Van het een komt het ander, want nu doet zich de vraag voor hoe we van tevoren dergelijke goede benaderingen van onmeetbare getallen kunnen vinden. We brengen even in herinnering dat $\frac{22}{7}$ en $\frac{355}{113}$ heel goede benaderingen van π zijn. Het zou te ver voeren op de theoretische achtergrond in te gaan en daarom volstaan we met de mededeling dat dit op de theorie van de kettingbreuken berust, een vinding van Christiaan Huyghens die ze toepaste in de constructie van astronomische uurwerken waarbij tandwielen zo goed mogelijk een onmeetbare verhouding moesten overbrengen.

Voor de kenners:

$$\frac{\alpha}{2\pi} = (0, 9, 1, 3, 4, 5. \ldots)$$

De rij benaderende breuken is:

$$\frac{1}{9}, \frac{1}{10}, \frac{4}{39}, \frac{17}{166}, \frac{89}{869}, \ldots$$

Appendix

De afleiding van de transformatieformules (1) berust op de overweging dat de som van de projecties van de samenstellende delen van een gebroken lijn $A_1A_2 \ldots A_m$ op een willekeurige lijn l gelijk is aan de projectie van A_1A_m, het lijnstuk dat de uiteinden verbindt. In fig. 5 laten we het zien voor een gebroken lijn ABC.

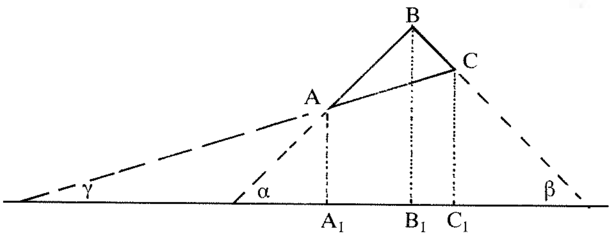

fig. 5

$$A_1B_1 + B_1C_1 = A_1C_1$$
$$AB\cos\alpha + BC\cos\beta = AC\cos\gamma$$

De projectie van een enkel lijnstuk AB op de lijn l wordt daarbij verkort met de cosinus van de projectiehoek: $A_1B_1 = AB\cos\alpha$.

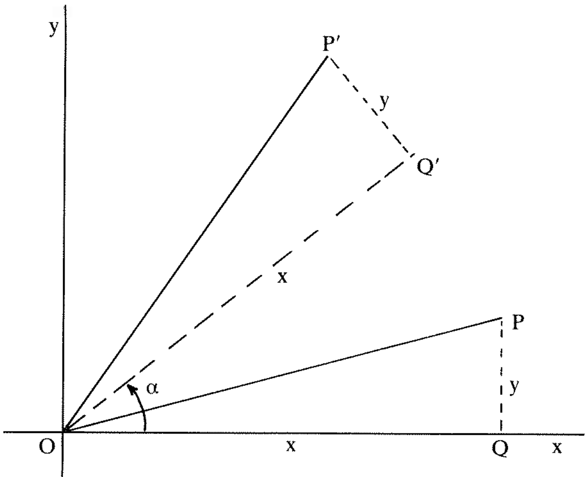

fig. 6

De afleiding van (1) kunnen we aflezen uit fig. 6. Is Q de projectie van O(x, y) op de X-as zodat OQ = y, dan denken we Δ OQP over de hoek α gedraaid om O als draaipunt. De nieuwe stand is Δ OQ′P′. De af te leiden coördinaten van P′ zijn x′ en y′.
Volgens het bovenstaande principe geldt:

$$\text{proj. } OQ' + \text{proj. } Q'P' = \text{proj. } OP'$$

Projecteren op de X-as geeft:

$$x\cos\alpha + y\cos(\alpha + 90°) = x'$$

ofwel:

$$x' = x\cos\alpha - y\sin\alpha$$

Projecteren op de Y-as geeft:

$$x\cos(90° - x) + y\cos\alpha = y'$$

dus:

$$y' = x\sin\alpha + y\cos\alpha.$$

```
10 REM ***SPIRAAL***
20 GCLEAR
30 SCALE -1,1,-1,1
40 DISP "KIES A,B"
50 INPUT A,B
60 X=1 @ Y=0
70 FOR K=0 TO 100
80 MOVE X,Y @ DRAW X,Y
90 Z=X
100 X=A*X-B*Y
110 Y=B*Z+A*Y
120 NEXT K
130 END
```

Programma vormt spiraal van punten wanneer $a^2 + b^2 < 1$