JULIA-FRACTALEN OP TI-92

Regis Ockerman Sint-Jozef-Klein-Seminarie Sint-Niklaas T^3 - Vlaanderen

Inleiding

In een artikel in jaargang 1993 heeft collega J. De Moor geschreven hoe men, als toepassing van de theorie der complexe getallen, kan komen tot het tekenen van Julia-fractalen.
Er was ook een programma bij ontwikkeld in Turbopascal.
Ik heb deze zaken, tot mijn grote tevredenheid, een aantal jaren gebruikt om het 5° jaar af te sluiten.
Dan hadden we de complexe getallen gezien en vrijwilligers hadden op zaterdagvoormiddag een reeks van 8 lessen Turbopascal gehad.
Vorig schooljaar wou ik voor mijn allerlaatste les in een 5° jaar Julia-fractalen laten zien op TI 92.
Ik ben er ook in geslaagd!!.
Gezien de mogelijkheid om op de TI-92 rechtstreeks met complexe getallen te rekenen was het programma veel korter. De uitvoersnelheid was natuurlijk veel lager.

I. Theorethische Achtergrond

Deze is volledig terug te vinden in het vermelde artikel. Ik zal me beperken tot de essentie.

1. Baan van een punt

In het " vlak van Gauss" wordt een complex getal voorgesteld als een punt. Beschouwen we nu de transformatie: $f : C \to C; z \to z^2$ dan kunnen we aan ieder origineel een beeld toekennen. We kunnen de transformatie ook een aantal maal na elkaar uitvoeren. We zouden dan het volgende kunnen krijgen

$1 + i \to (1 + i)^2 = 2i \to -4 \to 16 \to \ldots$

Als we al deze punten verbinden in het complexe vlak spreekt men over de baan van een punt.

2. Gebieden van het complexe vlak

Voor de transformatie $f : C \to C; z \to z^2$ speelt de kromme met vergelijking $|z| = 1$ een belangrijke rol. Dit is namelijk namelijk de eenheidscirkel in het complexe vlak. Hij wordt gevormd door de punten waarvan de modulus 1 is.
We kunnen nu drie gevallen onderscheiden : $(z = r(\cos \alpha + i \sin \alpha))$.

-) Voor punten binnen de cirkel is $r < 1$ en naderen de opeenvolgende beeldpunten naar 0.
   Deze punten vormen het **binnengebied .**
-) Voor punten buiten de cirkel is $r > 1$ en zullen de opeenvolgende beeldpunten zich steeds verder verwijderen van het punt 0 of divergeren. Deze punten vormen het **buitengebied.**
-) Voor de punten op de cirkel is $r = 1$. De beeldpunten blijven dus altijd op de cirkel liggen .
   Dit is de **randkromme.**

3. Dekpunten (of fixpunten)

Een dekpunt van een transformatie is een punt waarvoor geldt $f(z) = z$. Passen we dit toe voor $f(z) = z^2$, dan krijgen we $z^2 = z$ of de vergelijking $z^2 - z = 0$, met als oplossingen $z = 0$ en $z = 1$.
Bekijken we die twee dekpunten, dan zien we dat er grondige verschillen zijn:

-) Nemen we punten in de omgeving van het punt $z = 0$, dan weten we dat de beelden uiteindelijk terug in 0 terecht komen. Dit punt noemen we het **stabiel dekpunt.**
-) Nemen we punten in de omgeving van het punt $z = 1$,dan kan er van alles gebeuren:
   -) Punten in de omgeving, maar binnen de cirkel, gaan uiteindelijk naar 0.
   -) Punten in de omgeving, maar buiten de cirkel, gaan divergeren.
   -) Punten op de cirkel, hebben beeldpunten die altijd op de cirkel blijven.

Dit punt noemen we het **instabiel dekpunt.**

4. Bepalen van de randkromme

Om punten van de randkromme te bepalen, gaan we generaties terug de inverse beeelden zoeken van het instabiel dekpunt.

In de eerste ronde moeten we dus oplossen $f(z) = z^2 = 1$. Dit levert $z = \pm 1$.

Voor de 2° generatie moeten we oplossen $z^2 = -1$. Dit levert de oplossingen $z = \pm i$.

Voor de 3° generatie moeten we oplossen $z^2 = \pm i$. Dit betekent dus ook dat we de 2°machtwortels moeten bepalen uit $\pm i$. Dit doen we best onder goniometrische vorm.

Op deze wijze krijgen we steeds meer punten van de randkromme.
In dit geval kenden we de oplossing reeds, maar deze werkwijze zal ons straks nog van pas komen.

5. De theorie van JULIA

Passen we in het complexe vlak de transformatie toe $f(z) = z^2 + c$, met c een complexe constante, dan krijgen we een situatie die vergelijkbaar is met de voorgaande. We krijgen een **binnengebied**, een **buitengebied** en een **randkromme**, die de scheiding vormt tussen de twee gebieden.
Deze randkromme noemen we ook **fractaal.**
Stilzwijgend wordt verondersteld dat het binnengebied gesloten is. Dit is maar zó voor bepaalde waarden van c. (Zie het artikel van Koen Stulens).

6. Bepalen van een fractaal

We passen passen dit concreet toe voor $f(z) = z^2 - 3/4$.
We bepalen eerst de dekpunten. Daarvoor moeten we oplossen
$z = z^2 - 3/4$ of $4z^2 - 4z - 3 = 0$.
De oplossingen zijn $z = -1/2$ en $z = 3/2$. Als we deze situatie vergelijken met de voorgaande, hebben we alle reden om aan te nemen dat 3/2 het instabiel dekpunt is.
Om de inverse beelden te bepalen van de eerste generatie moeten we oplossen $z^2 - 3/4 = 3/2$.
Dit geeft $z = \pm 3/2$.
Om de 2° generatie te zoeken moeten we oplossen : $z^2 - 3/4 = -3/2$ of $z^2 = -3/4$. Dit levert

$$z = \pm i\sqrt{3}/2$$

We hebben reeds een paar punten, die we kunnen uitzetten in het vlak.
Verdere berekeningen laten we beter doen door een programma.

II. Opbouw van een Programma

1. Bepalen van de dekpunten

Deze moeten voldoen aan: $z^2 + c = z$ of aan $z^2 - z + c = 0$.
We kunnen hiervoor een functie maken, met naam fixpt, als volgt .
We toetsen in: APPS, 7, 3, →, 2, ↓, ↓, fixpt, Enter, Enter en komen zo in het editorscherm.
We krijgen te zien :

```
: fixpt()
: Func
:
: EndFunc
```

We editeren de titel : fixpt(cj) op de eerste lijn .
Onder de lijn Func typen we : csolve($z^2 - z + cj = 0$,z) en ↓
We keren terug naar het Home scherm. (APPS, 1)
We kunnen nu de functie aanroepen door de naam en de parameter.
Zo geeft fixpt(-3/4) ons als resultaat -1 en 3/2 . Voor fixpt(i) krijgen we iets met dubbele wortelvormen.
We kunnen dit vermijden door de groene toets ◊ samen met Enter in te drukken, of door in te geven fixpt(i x 1.0). Dan wordt benaderend gerekend.
We gebruiken voor de parameter de notatie cj, omdat het gevaarlijk is te werken met éénletter-variabelen.

2. Wat is het instabiel dekpunt ?

We vermoeden dat 3/2 het instabiel dekpunt is. Dit is makkelijk te controleren.
We nemen een punt in de omgeving Vb 1.51.
We typen nu het volgende in : 1.51 Enter.
Vervolgens $z^2 - 3/4 \mid z = ans(1)$.
Als we nu enkele malen Enteren krijgen we de opeenvolgende iteraties.

Er verschijnt : 1.5301 1.59121 1.78194 2.4253 5.13207 25.5881 ....
wat er op wijst dat de punten divergeren.

3. Programma

Voor het opstellen van een programma gaan we er van uit dat het instabiel dekpunt bepaald is.
Als we naar het voorbeeld kijken, zien we dat we de originelen willen bepalen van het dekpunt d, dus moeten we oplossen $z^2 + c = d$.
We moeten dus de vierkantswortels bepalen uit d - c. Dit geeft een resultaat, waarmee we de beeldpunten afdrukken.
Voor de 2° generatie inverse beelden hebben we de d vervangen door het eerste resultaat en zo gaat het verder. We krijgen dus schematisch:

```
Start
Zet de iteratieteller op nul
Herhaal zolang de bovengrens van de teller niet bereikt is
        bereken vkw(d - c) → resultaat (= complex getal)
        gebruik het complex getal om punt op het scherm te zetten
        zet ook het symmetrisch punt uit ( 2 tegengestelde 2° machtswortels)
        vervang d door resultaat of - d door resultaat
        verhoog de teller
Stop
```

We maken nu het programma jti(), als volgt : APPS, 7, 3, ↓, ↓, jti, Enter, Enter. Dit programma krijgt 3 parameters mee: de Juliaconstante →cj, het fixpunt → dj en de bovengrens van de iteratieteller → gj
We editeren de eerste lijn en gaan verder:

```
jti(cj,dj,gj)
Prgm
        PlotsOff
        FnOff
        ClrDraw
        0 → tj
        While tj < gj
                √(dj - cj) → dj
                int(real(dj)*55)  → xj
                int(imag(dj)*35) → yj
                PxlOn  - yj + 50, xj + 120
                PxlOn  yj + 50, - xj + 120
                        If rand(2) = 1 Then
                                - dj → dj
                        EndIf
                tj + 1 → tj
        EndWhile
EndPrgm
```

Voor we het programma laten lopen moeten we nog een paar zaken instellen.
We kiezen MODE en onder Graph de optie FUNCTION. We kiezen bij Complex format de optie RECTANGULAR. We bevestigen door twee maal te Enteren. Onder APPS, 2 kunnen we in de functie-editor kijken of er niets aangevinkt staat, ook geen statistische plots. Een goede controle is APPS, 4; waarbij we wel een grafisch scherm mogen zien, maar geen grafieken. We kunnen nu ook best de assen afzetten. Daarvoor F1, 9, Grid OFF, Axes OFFen twee maal Enteren.
De drie instructies: PlotsOff, FnOff, ClrDraw moeten er voor zorgen dat alles goed loopt, zodat er niet eerst iets anders getekend wordt voor de fractaal.
Zou er toch iets mis gaan, kan ook eens kijken of er niets aanstaat bij poolvorm, parametervorm, 3D.
De TI-92 werkt met 240 x 100 pixels. De getallen 120 en 50 geven dus het midden van het scherm.
De getallen 55 en 35 zorgen ervoor dat het scherm voldoende gevuld is, zonder dat men er buiten zit.
Deze zijn min of meer experimenteel bepaald en kunnen eventueel nog geoptimaliseerd worden.
Rand(2) kan als resultaat 1 of 2 geven.
Men roept het programma aan, in het Home-scherm, met de naam en de drie parameters.
VB jti(-0.75, 1.5, 1000). Het programma kan stoppen door te drukken op ON en dan ESC.

III. Gebruiksvriendelijk Programma

1.Totaalprogramma voor het tekenen van de fractaal

```
julia()
Prgm
© REGIS OCKERMAN
ClrHome
        Dialog
                Title "J U L I A  F R A C T A L E N "
                Request "Geef complexe  c ",cj
                Text  "Kies dan het fixpunt"
        EndDlog
        expr(cj)→cj
                0.5*(1+√(1-4cj))→temp1
                0.5*(1-√(1-4cj))→temp2
                PopUp  {string(temp1),string(temp2)},nj
        If nj=1 Then
                temp1→dj
                        Else
                temp2→dj
        EndIf

PlotsOff
FnOff
ClrDraw
        0→tj
        While tj < 2500
                √(dj-cj)→dj
                int(real(dj)*55)  → xj
                int(imag(dj)*35) → yj
                PxlOn  - yj + 50, xj + 120
                PxlOn  yj + 50, - xj + 120
                        If rand(2) = 1 Then
                                - dj → dj
                        EndIf
                tj + 1 → tj
        EndWhile
EndPrgm
```

Het programma wordt aangeroepen met julia().
We moeten geen parameter meegeven, maar die wordt opgevraagd in een dialogbox.
In het editorscherm typen we: f3, 1, →, 5, Enter. Dit geeft de box.
Dan geven we de titel in, vragen de cj op met request, en geven nog een tekst.
De opgevraagde cj moet omgezet worden tot een getal.
Vervolgens berekenen we de 2 fixpunten, niet met solve, maar met de gewone formule.
We stoppen beide oplossingen in tijdelijke variabelen.
De keuze gebeurt in een PopUp-menu, waarbij via een lichtbalk uiteindelijk één van de oplossingen doorgegeven wordt aan de variabele dj.
De code, die hiervoor moet geschreven worden, zie je in het programma.
Het © symbool halen we uit f2, 9
De lijn ClrHome zorgt er voor dat dialogbox en keuzemenu op een "proper" scherm komen
Wij voeren standaard 2500 iteraties uit, tenzij het programma onderbroken wordt.
De rest lijkt op het voorgaande programma.

2. Een andere functie voor de fixpunten

Een nadeel van het werken met Solve is de uitvoer. Lossen we vb. $x^2 - 1 = 0$ op, dan krijgen we : Solve ($x^2 - 1 = 0$,x) met als uitvoer $x = 1$ *or* $x = -1$.
Nu is dit vrij eenvoudig, maar als we dit benaderen voor $z^2 - z + i = 0$ dan krijgen we als uitvoer: z = 1.30024 - .62481. i or z = -.300243 + 0.624811. i. Deze uitdrukking is één geheel en het is vrij vervelend als we met een deel daarvan willen verder werken.
We kunnen dit wel als we de oplossingen in een lijst steken, want daar kunnen we een index opzetten. Daarvoor kunnen we de volgende instructie gebruiken:

exp ▷ list (solve ($x^2 - 1 = 0$,x),x) met als uitvoer {-1 1}
Typen we nu ans(1)[1] dan krijgen we uitsluitend -1.
Opgelet! Men moet de uitdrukking **exp ▷ list(** uit de catalog gaan halen.

We gaan nu het programma fixpt() aanpassen tot jfixpt()

```
: jfixpt(cj,rj)
: Func
: approx(exp ▷ list (csolve (z^2 - z + cj = 0, z), z))[rj]
: EndFunc
```

We kunnen nu in het home-scherm de functie aanroepen.
jfixpt(i,1) geeft als uitvoer -.300243 + .624811 .i en dit is makkelijk om mee verder te werken

Toemaatjes

-) Wil men voor een "open deur" alléén de San-Marco fractaal laten zien op een maximaal gevuld scherm, dan kunnen de getallen 55 en 35 vervangen worden door 75 en 50.
-) Wat op de TI 89
Dit toestel heeft dezelfde algebra als de 92+ , maar met een kleiner grafisch scherm van hoge kwaliteit. Er zijn 160 x 80 pixels.
Het is dus voldoende de programmalijnen aan te passen voor uitvoer naar het scherm. Dit wordt

```
int(real(dj)*30)  → xj
int(imag(dj)*20) → yj
PxlOn - yj + 40, xj + 80  en analoog
```

-) Men zou kunnen de gebruike variabelen local zetten, zodat ze niet kunnen doorgegeven worden aan andere programma's uit dezelfde folder. Men doet dit door achter de lijn Func of Prgm te zetten :
Local, gevolgd door de naam van de variabelen. VB Local cj, dj.
Dit durft wel eens problemen geven op een TI 92+, daarom is het hier niet gedaan.
Er is wel opzettelijk gekozen voor tweeletter-variabelen
-) Vergeet niet dat we hier met "source"- code zitten en niet met een *.exe file.
De code kan dus gewijzigd worden, ook door onkundigen of moedwilligen.
Het is nuttig de file te "locken" als volgt : 2nd VAR-LINK, ga met de lichtbalk naar de file, f1, 6.
Beter is nog het programma uit te printen of met de Graph Link een backup te maken op HD of floppy.
-) Men spreekt ook wel eens over fixpunt i.p.v. dekpunt; stabiel dekpunt wordt dan aantrekkend fixpunt en instabiel dekpunt wordt afstotend fixpunt.
-) In het artikel van Koen Stulens vinden we ook een aantal interessante c- waarden:
c = -1, c = 0.397 – 0.214.i , c = - 0.122 + 0.744.i . Verder c = 0,32 + 0.043.i, c = -1.25
Voor c = -2 krijgt men een lijnstuk.

Bibliografie

1. J. De Moor, Fractalen, Wiskunde en Onderwijs 19° Jaargang (1993) nr 74 pp. 161-171
2. K. Stulens, De Mandelbrot-verzameling, Wiskunde en Onderwijs 24° Jaargang (1998) nr 95 pp. 222-232.
3. H. Lauwerier, Fractals , Aramith Uitgevers Amsterdam 1987
4. Texas Instruments , Handleidingen voor TI - 92 en TI - 89