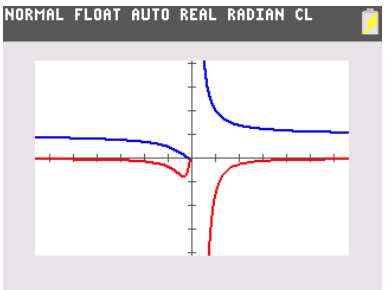
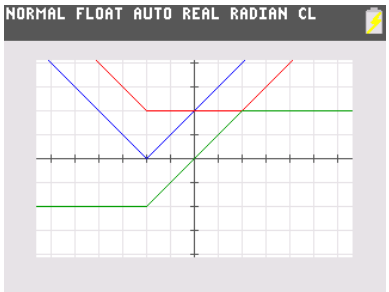
Cahiers T³ Europe
Vlaanderen nr. 39

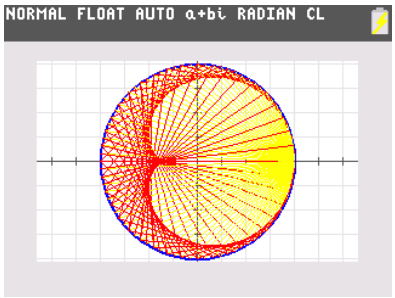
Kennismaking met de TI-84 Plus Color

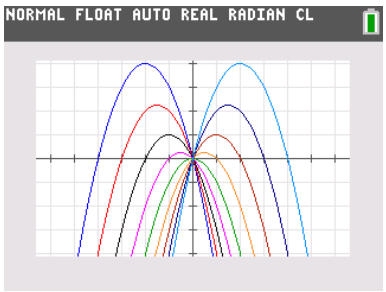
Wiskunde voor de 3de graad ASO

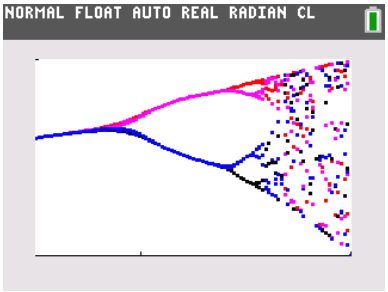
Didier Deses











KATHOLIEKE UNIVERSITEIT LEUVEN

Kennismaking met de **TI-84 Plus Color**

Dr Didier Deses[1]

[1]Leerkracht wiskunde KA Koekelberg, medewerker aan het departement wiskunde van de VUB, stuurgroep T^3-Vlaanderen

Voorwoord

We bekijken de nieuwigheden van de **TI-84 Plus Color** . Naast de enkele veranderingen in de menu's en opties wordt vooral aandacht besteed aan het gebruik van het nieuwe kleurenscherm. Enkele klassieke oefeningen worden beschouwd, alsook een voorbeeld uit de optica. Hiermee maken we enkele zeer eenvoudige programma's. Deze zullen voluit het nieuwe kleurenscherm gebruiken. We leggen een link naar de kunstwereld, hetgeen tenslotte leidt naar een knutseloefening voor leerlingen.

Inhoudsopgave

1 Inleiding **3**

2 Grafieken met kleur **9**
2.1 Analyse 9
2.2 Meetkunde 23
2.3 Goniometrie 25
2.4 Statistiek 26

3 Programmeren **33**
3.1 Raaklijnen 33
3.2 Families van krommen 33
3.3 Matrices en emoticons 36

4 Uitgebreidere programeerprojecten **39**
4.1 Kaustiek van een parabool 39
4.2 De cardioïde: over wiskunde, optica, kunst en timmeren 42
4.2.1 Kaustieken in een kopje 42
4.2.2 String art 46
4.2.3 Een bijzondere familie cirkels 48
4.3 Oppervlakte tussen twee krommen 49
4.4 Model van Verhulst en chaos 52
4.5 Kaarten tellen 56
4.5.1 Blackjack of 21 56
4.5.2 Hi-Lo count 56
4.5.3 Blackjack op de **TI-84 Plus Color** 58

5 Appendix: Eventjes leren programmeren **61**

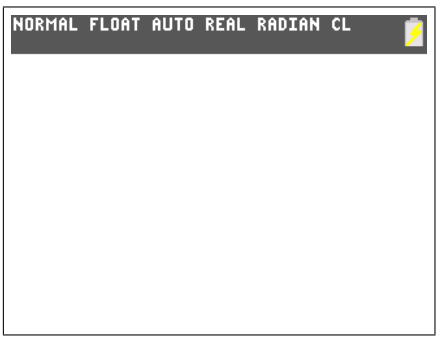
Hoofdstuk 1

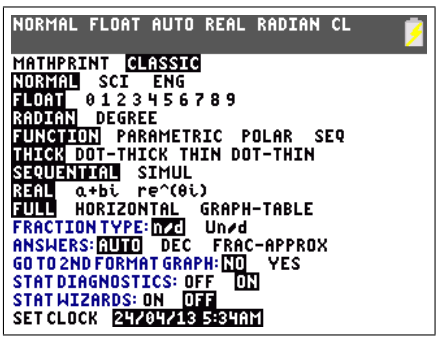
Inleiding

Deze tekst gaat over de mogelijkheden van de nieuwe **TI-84 Plus Color** . Deze nieuwe versie van de **TI-84 Plus** verschilt bijna niet in gebruik met het oudere zwart-wit toestel. De toetsen, functies en menu's zijn hetzelfde gebleven. Voor wie de **TI-84 Plus** gewoon is, zal de overstap vlot gaan.

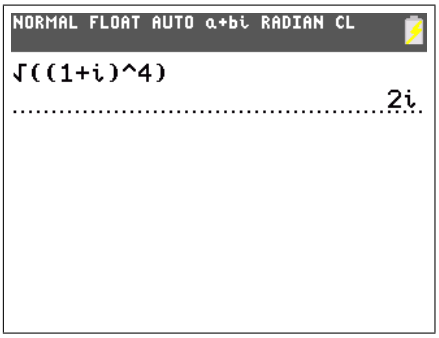
Het verschil tussen beide toestellen zit in het nieuwe scherm. De **TI-84 Plus Color** heeft een kleurenscherm dat een hogere resolutie heeft (320×240 pixels). Daarnaast zijn de losse baterijen vervangen door een accu, die men via een usb-lader kan opladen.

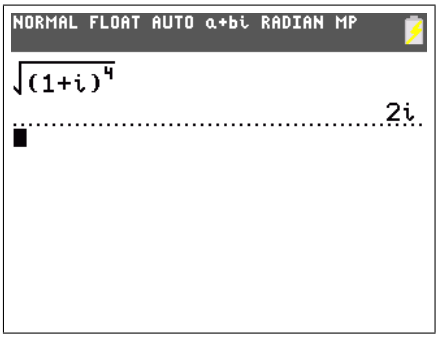
Aan de software is er bijna niets veranderd. Enkele verbeteringen zijn wel noemenswaardig. Wanneer men de **TI-84 Plus Color** aanzet, merkt men bovenaan een statusbalk. Hierin kan men onmiddellijk zien wat de staat van de batterij is en of het toestel in graden of radialen rekent. Ook andere opties die men via `mode` kan veranderen, worden hier weergegeven.



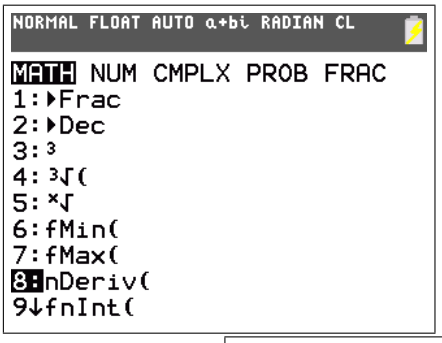


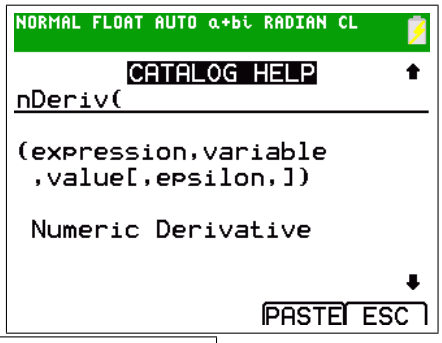
Nieuw voor de **TI-84 Plus Color** is optie `[mathprint]` onder `mode`. Dit geeft de stijl aan waarin formules worden ingegeven. In de `[classic]`-mode zal het toestel alle formules op één lijn zetten, in `[mathprint]` worden de formules getoond zoals ze in de wiskunde worden geschreven.

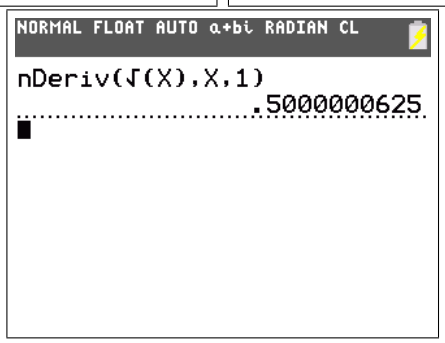




Een andere vernieuwing is de ingebouwde helpfunctie. Als je in een menu een commando vindt, kan je altijd [+] gebruiken om een korte omschrijving te krijgen. We geven het voorbeeld van [math][nDeriv] dat de afgeleide van een functie in een punt numeriek benadert.

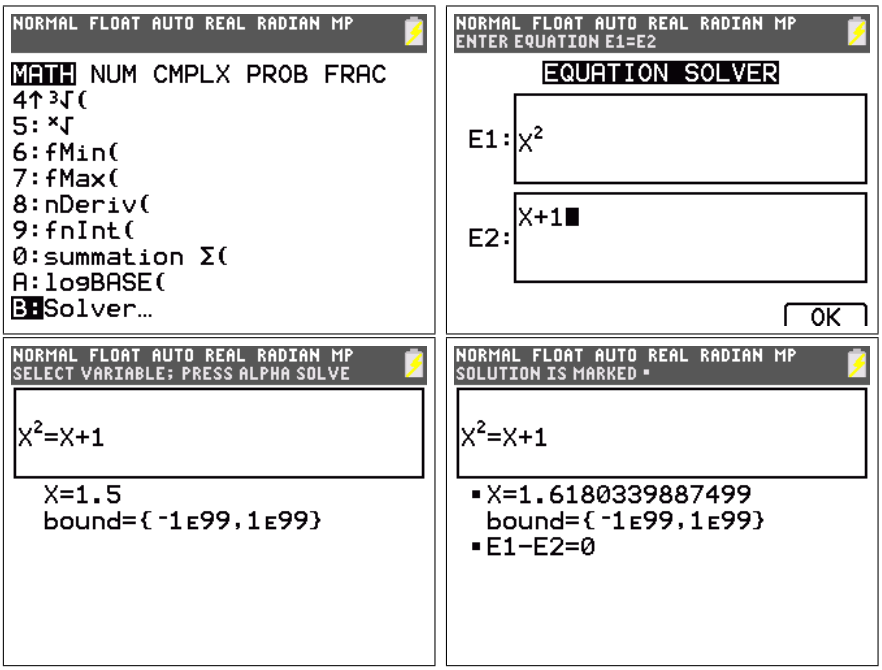




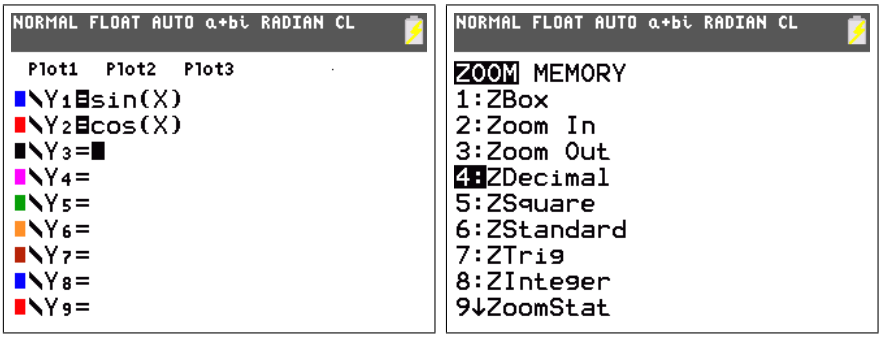


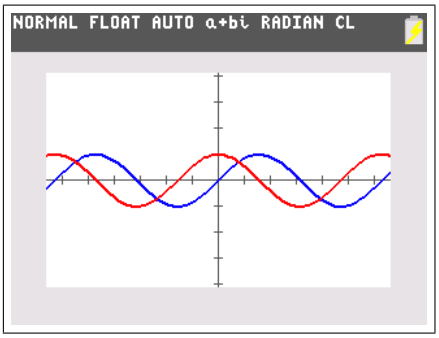
Tenslotte is ook de `Solver` lichtjes aangepast in het nieuwe systeem. Via [math][Solver...] geraak je erin. Nu kan je het linkerlid (E_1) en het rechterlid (E_2) van een vergelijking ingeven. Nadat je [OK] hebt gekozen verschijnt de vergelijking $E_1 = E_2$. Geef onderaan voor X een benadering van de oplossing (de variabele `bound` geeft aan tussen welke twee grenzen een oplossing moet gezocht worden). Ga nu op de regel `X=...` staan en druk dan [alpha][SOLVE]. Je aanvankelijke benadering verdwijnt en wordt vervangen door de berekende (benaderde) oplossing. In onderstaand voorbeeld zoeken we *de gulden snede* als x-waarde waar de parabool $y = x^2$ en de rechte

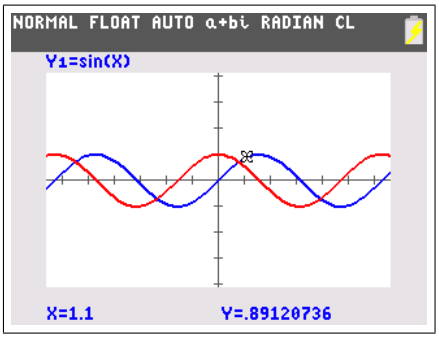
$y = x + 1$ elkaar snijden. We geven een eerste benadering (1.5) en laten de `Solver` een betere benadering vinden.



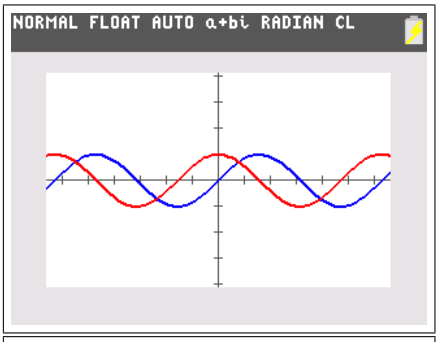
Het zal niet verbazen dat de grootste nieuwigheden van de **TI-84 Plus Color** te vinden zijn in de grafische eigenschappen van het toestel. Indien we gewoonweg de grafiek van een functie maken, zien we dat opeenvolgende grafieken in verschillende kleuren komen te staan. Wanneer er informatie op het scherm bijkomt (bijvoorbeeld bij het gebruik van `trace`), komt dit niet op het grafisch venster meer, maar eronder.

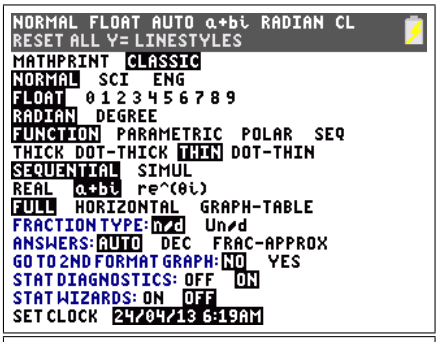


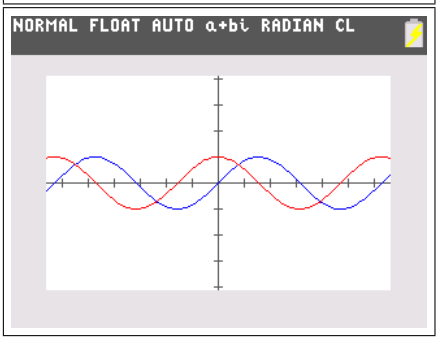


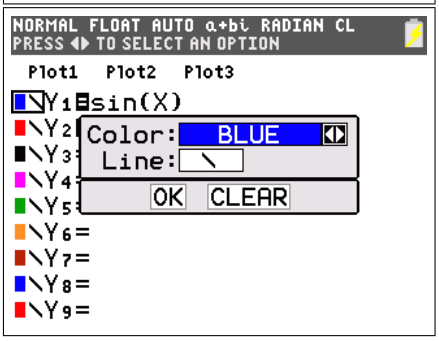


Standaard worden de grafieken met een vette lijn getekend. Door [mode] te gebruiken kan men dit veranderen. De instellingen [thick] en [thin] zorgen voor een dikke of dunne lijn, met [dot-thick] en [dot-thin] worden de punten van de grafiek niet met elkaar verbonden door een lijn. Je kan deze opties ook grafiek per grafiek wijzigen door vooraan op de lijn $y_1 =$ te gaan staan.

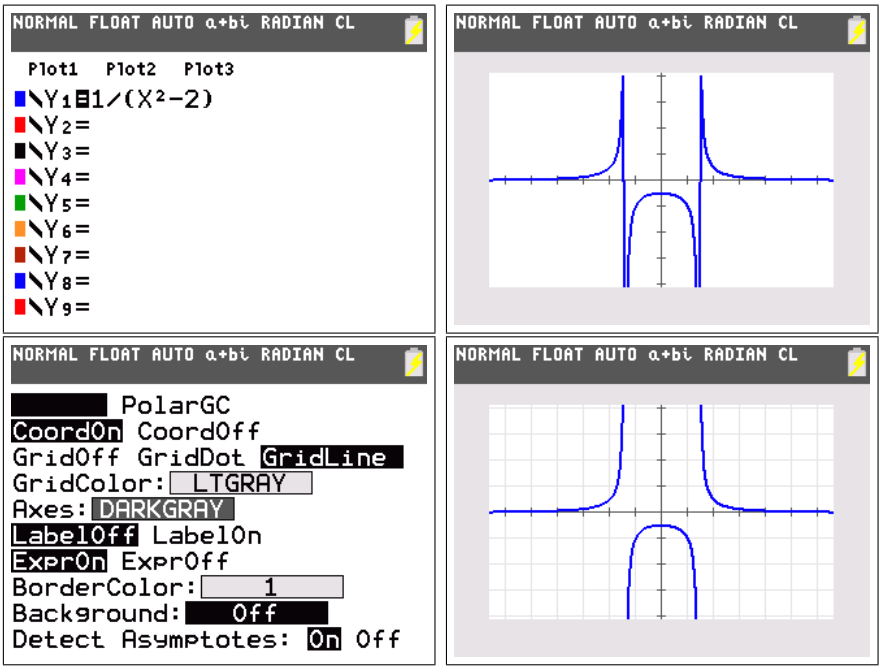








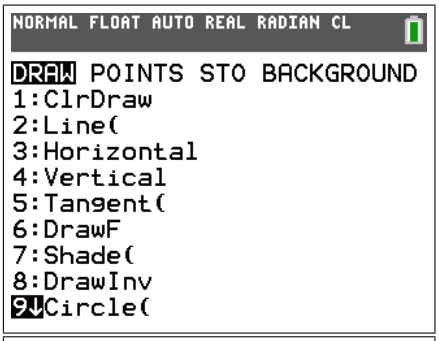
Verdere opties i.v.m. grafieken vind je onder [2nd][format]. Je kan hier kiezen om een raster te maken en de kleur aan te passen. Verder is de optie [Detect Asymptotes] nieuw, je kan hiermee zorgen dat de **TI-84 Plus Color** nagaat of er verticale asymptoten zijn alvorens twee punten te verbinden.

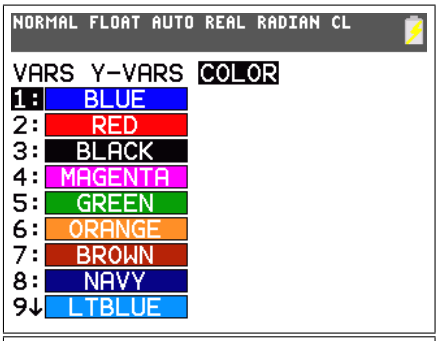


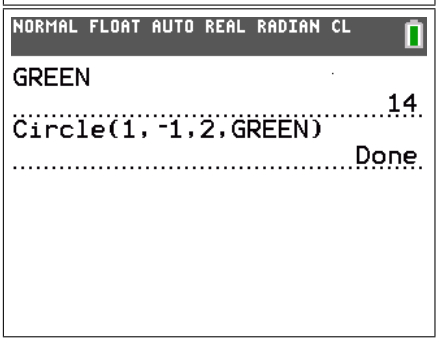
Intern worden alle kleuren van de rekenmachine voorgesteld door een specifiek getal. Voor sommige commando's wordt dit getal gebruikt om de kleur te bepalen. Onderstaande tabel geeft de kleurcodes.

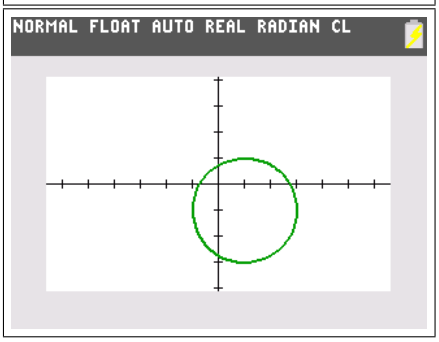
kleur	code
blauw	10
rood	11
zwart	12
paars	13
groen	14
oranje	15
bruin	16
donker blauw	17
licht blauw	18
geel	19
wit	20
licht grijs	21
medium grijs	22
grijs	23
donker grijs	24

Het is ook mogelijk om de kleuren te halen uit `vars` `[color]`.


NORMAL FLOAT AUTO REAL RADIAN CL
DRAW POINTS STO BACKGROUND
1:ClrDraw
2:Line(
3:Horizontal
4:Vertical
5:Tangent(
6:DrawF
7:Shade(
8:DrawInv
9↓Circle(


NORMAL FLOAT AUTO REAL RADIAN CL
VARS Y-VARS COLOR
1: BLUE
2: RED
3: BLACK
4: MAGENTA
5: GREEN
6: ORANGE
7: BROWN
8: NAVY
9↓ LTBLUE


NORMAL FLOAT AUTO REAL RADIAN CL
GREEN
14
Circle(1,-1,2,GREEN)
Done


NORMAL FLOAT AUTO REAL RADIAN CL

Hoofdstuk 2

Grafieken met kleur

2.1 Analyse

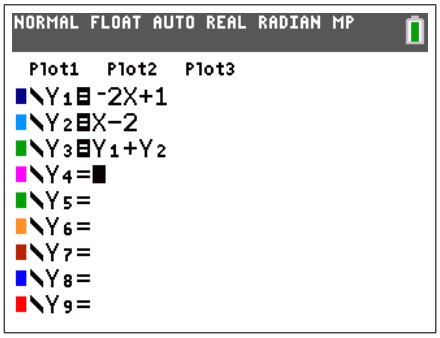
Opdracht 1. Beschouw twee rechten $y_1 = a_1x + b_1$ en $y_2 = a_2x + b_2$. Bestudeer de som $y = y_1 + y_2$.

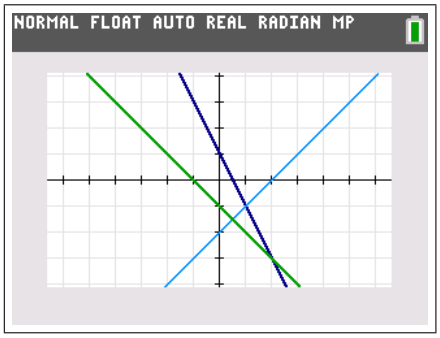
Oplossing. We kunnen de som schrijven als

$$y = y_1 + y_2 = (a_1 + a_2)x + (b_1 + b_2)$$

Volgende conclusies kunnen hieruit getrokken worden. Leerlingen uit de 2de graad kunnen hiervoor gemakkelijk de bewijzen uitschrijven.

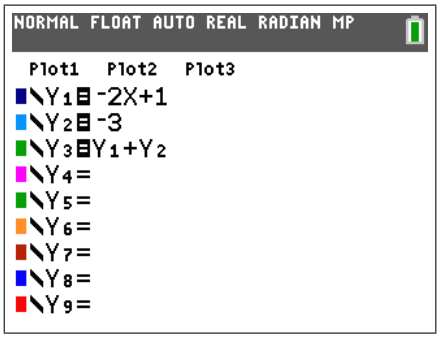
(i) De som van twee rechten zal opnieuw een rechte zijn, we spreken van de *somrechte*.

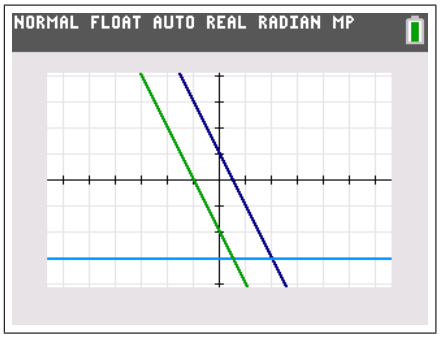




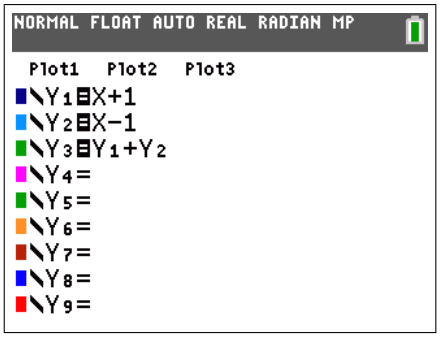
Merk op dat we hier de twee gegeven rechten in blauwtinten hebben gezet en dat de somrechte zelf in het groen komt.

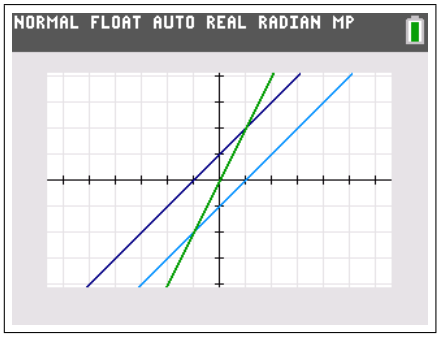
(ii) De somrechte van een rechte en een horizontale rechte, is een rechte evenwijdig met de eerste rechte.



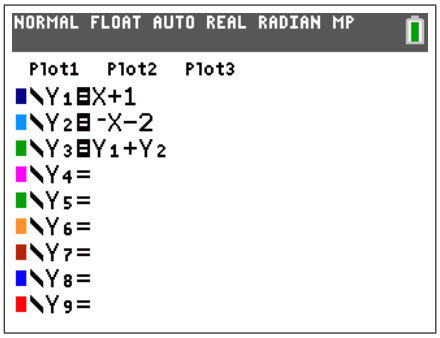


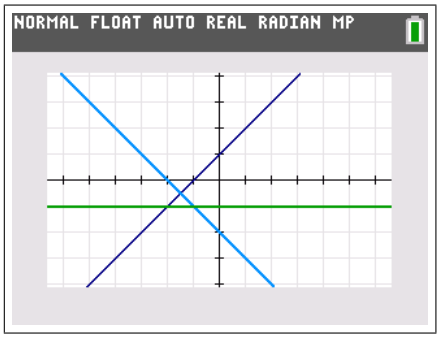
(iii) De somrechte van twee stijgende rechten is stijgend. De somrechte van twee dalende rechten is dalend.





(iv) Als twee rechten tegengestelde hellingshoeken hebben, dan is hun somrechte horizontaal.





Opdracht 2. Beschouw twee rechten $y_1 = a_1x + b_1$ en $y_2 = a_2x + b_2$. Bestudeer het product $y = y_1y_2$.

Oplossing. We kunnen het product schrijven als

$$y = y_1y_2 = a_1a_2x^2 + (a_1b_2 + a_2b_1)x + b_1b_2$$

Een andere schrijfwijze voor de rechten zal echter handiger zijn.

$$y_1 = a_1 x + b_1 = a_1(x - x_1)$$

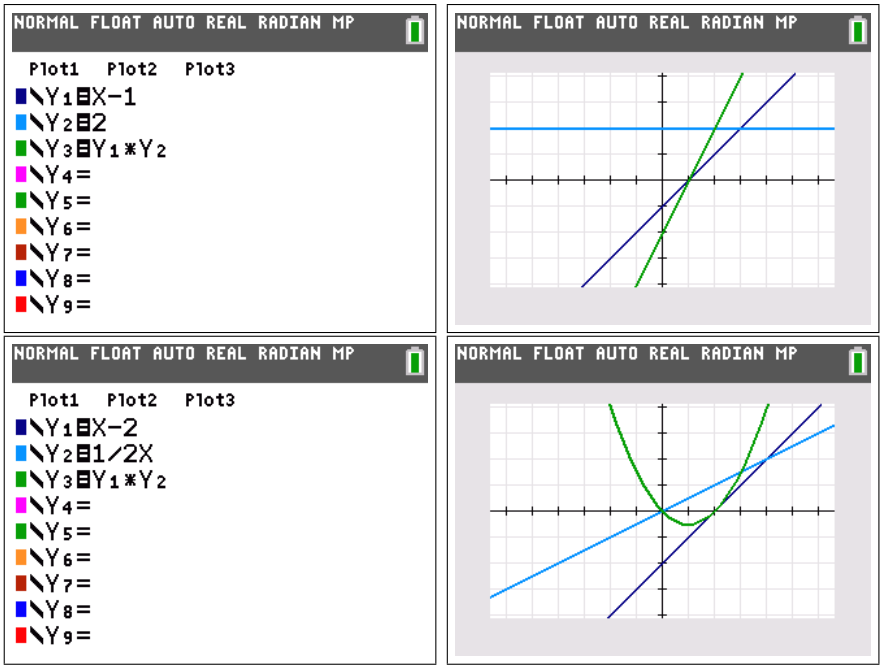
$$y_2 = a_2 x + b_2 = a_2(x - x_2)$$

Hierin zijn $x_1 = -\frac{b_1}{a_1}$ en $x_2 = -\frac{b_2}{a_2}$ de nulwaarden van de rechten. Het product is dan (als a_1 en a_2 niet nul zijn)

$$y = y_1 y_2 = a_1 a_2 (x - x_1)(x - x_2)$$

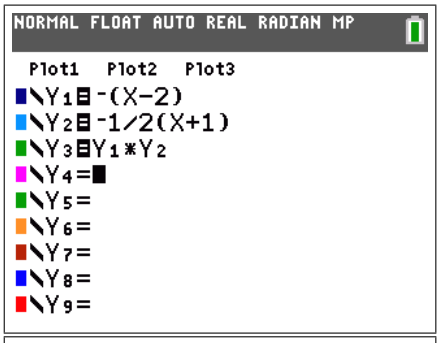
Volgende conclusies kunnen hieruit getrokken worden. Leerlingen uit de 2de graad kunnen hiervoor gemakkelijk de bewijzen uitschrijven.

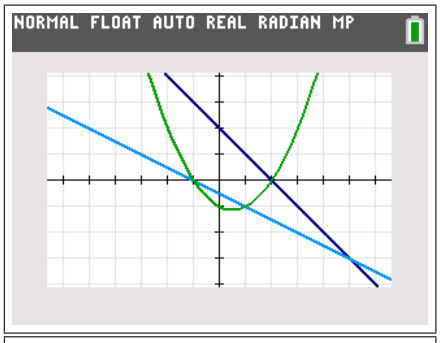
(i) Indien tenminste één van beide rechten horizontaal is, zal het product opnieuw een rechte zijn. In de andere gevallen zal het resultaat een parabool zijn, de *productparabool*.

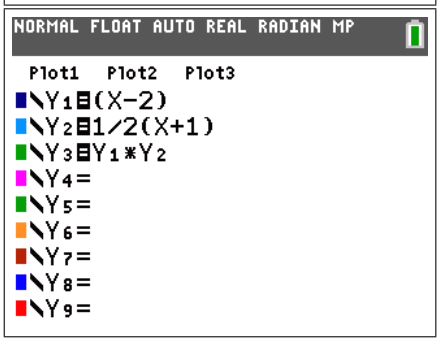


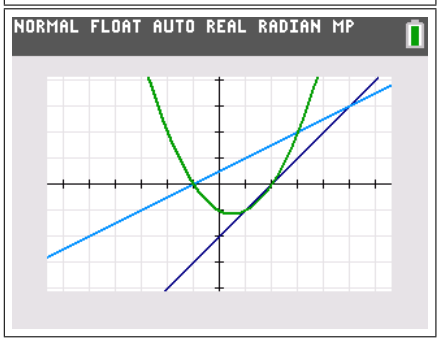
We beperken ons vanaf nu tot het geval van een productparabool.

(ii) Het product van twee dalende of twee stijgende rechten zal een dalparabool zijn.



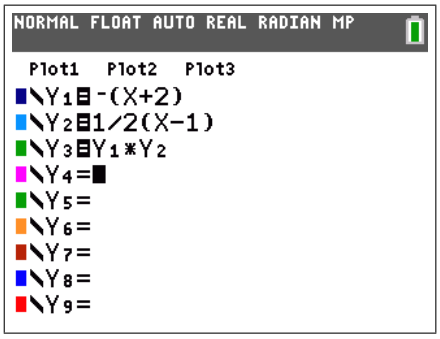


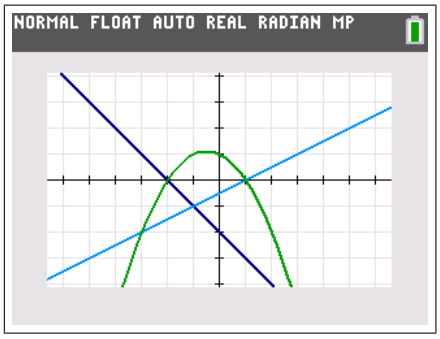




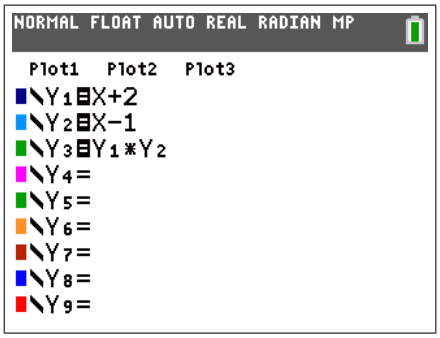
We zien aan dit voorbeeld dat, indien er een ontbinding in twee rechten bestaat van een parabool, dat deze zeker niet uniek is.

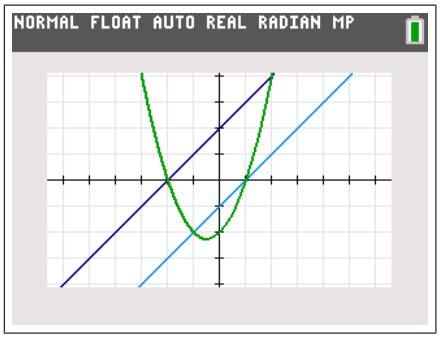
(iii) Het product van een stijgende en een dalende rechte zal een bergparabool zijn.





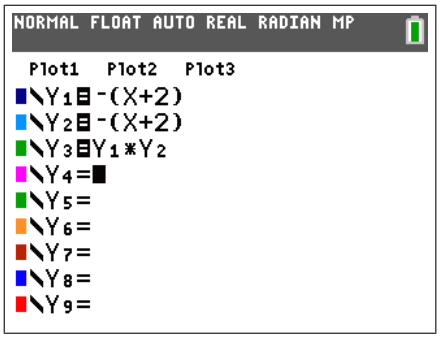
(iv) De nulwaarden van de rechten zijn tevens de nulwaarden van de productparabool.

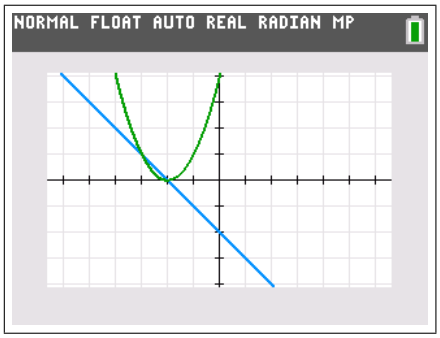




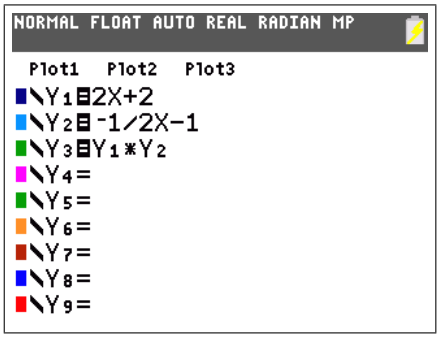
(v) Je kan bewijzen dat een parabool een productparabool is als en slechts als de discriminant $D \geq 0$ is.

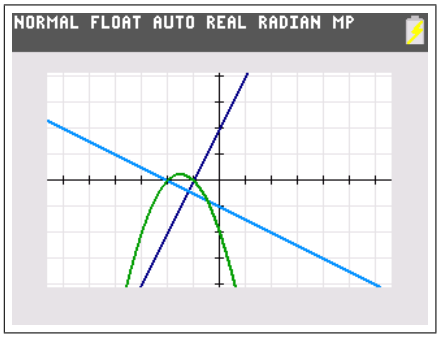
(vi) Een productparabool is afkomstig van twee dezelfde rechten, enkel en alleen als $D = 0$





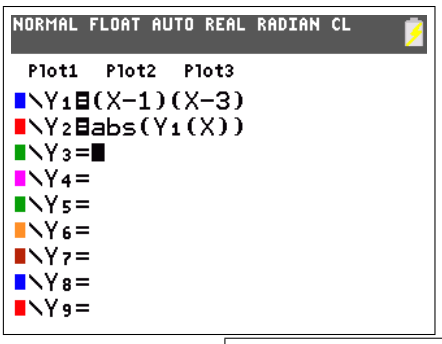
(vii) Twee rechten staan loodrecht op elkaar enkel en alleen indien de hoogstegraadscoëfficiënt van de productparabool gelijk is aan -1.

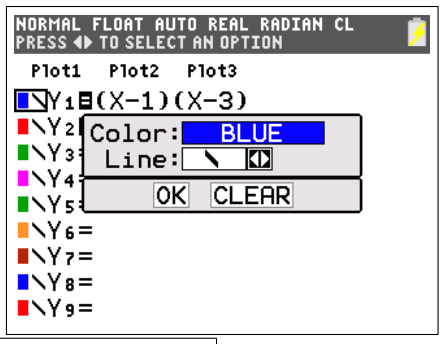


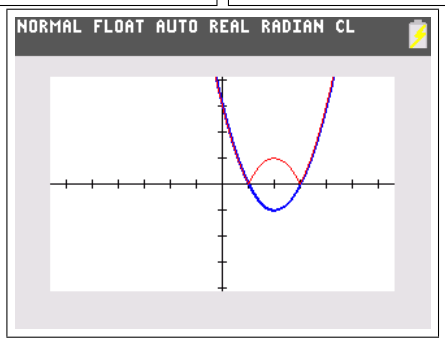


Opdracht 3. Ga na hoe $|.|$ de grafiek van een functie beïnvloedt. Gebruik als voorbeeld $f(x) = (x-1)(x-3)$ en beschouw de functies $|f(x)|$, $f(|x|)$ en $|f(|x|)|$.

Oplossing. Gebruik de **TI-84 Plus Color** om de grafiek van $f(x)$ in het vet te tekenen. De andere functie (bijvoorbeeld $|f(x)|$) komt in een andere kleur maar dunnetjes.

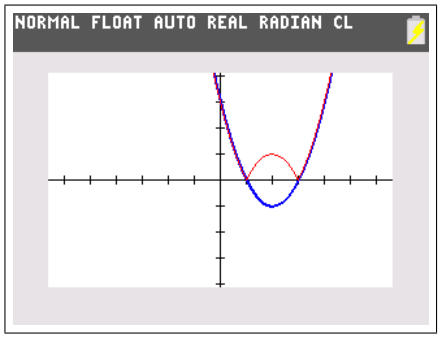




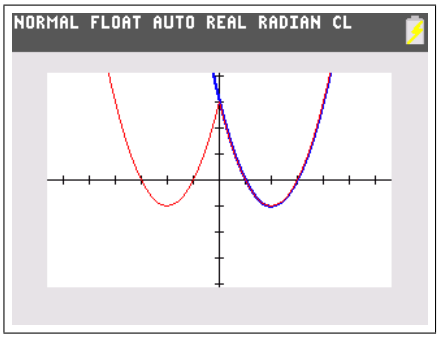


Je komt tot volgende conclusies:

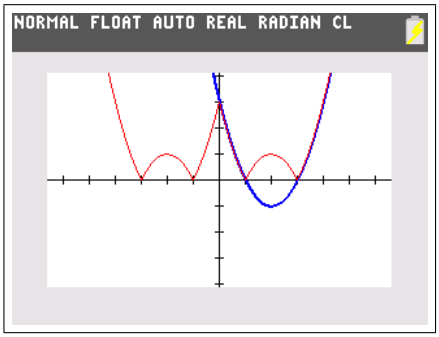
- De functie $|f(x)|$ is altijd positief. De grafiek van $|f(x)|$ is dus de grafiek van $f(x)$ waarbij alle delen onder de x-as gespiegeld worden t.o.v. deze as.



- De functie $f(|x|)$ is een even functie en dus symmetrisch t.o.v. de y-as. De grafiek van $f(|x|)$ is dus de grafiek van $f(x)$ voor $x \geq 0$, gespiegeld t.o.v. de y-as.



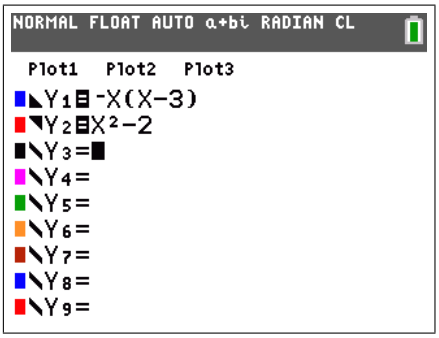
- De functie $|f(|x|)|$ is een combinatie van de vorige twee punten. De grafiek is dus de grafiek van $f(x)$ voor $x \geq 0$, gespiegeld t.o.v. de y-as, waarna alle delen onder de x-as nog eens gespiegeld worden t.o.v. deze as.

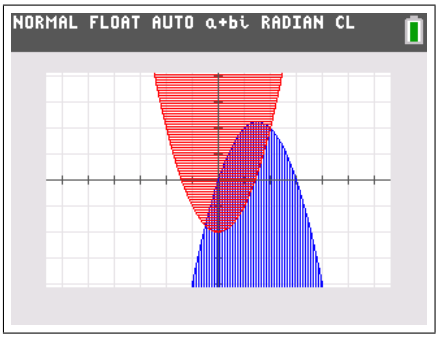


Opdracht 4. Los het volgend stelsel ongelijkheden in het vlak grafisch op.

$$\begin{cases} y + x(x-3) < 0 \\ y > x^2 - 2 \end{cases}$$

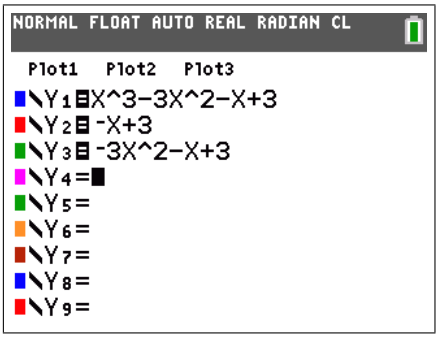
Oplossing. Je kan op de **TI-84 Plus Color** zeer eenvoudig de grafiek maken van ongelijkheden met twee onbekenden. Daarvoor herleid je de ongelijkheid naar de vorm $y > f(x)$ (of $y \leq f(x)$, ...). Geef nu de functie $f(x)$ in via [y=] en gebruik de pijltjestoetsen om het symbool voor de vergelijking te veranderen in de gewenste arcering (boven arceren = groter dan, onder arceren = kleiner dan). Je kan dit meerdere keren doen om de grafiek van een stelsel ongelijkheden met twee onbekenden grafisch op te lossen. Bijvoorbeeld:

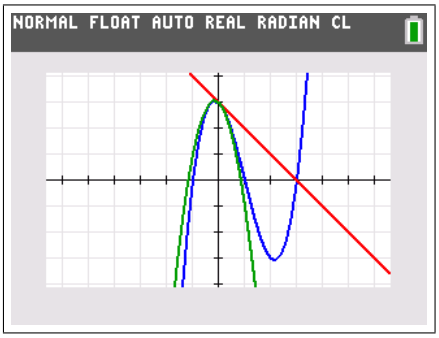




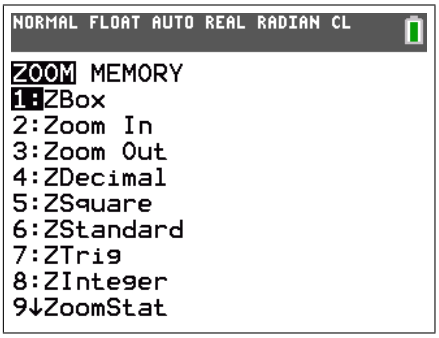
Opdracht 5. Gebruik de **TI-84 Plus Color** om te achterhalen wat het verband is tussen de grafiek van een veeltermfunctie en de grafiek van diezelfde functie waarbij het voorschrift slechts tot een bepaalde graad beperkt wordt. Wat is het verband tussen de grafiek $y = x^3 - 3x^2 - x + 3$ en de rechte $y = -x + 3$ en de parabool $y = -3x^2 - x + 3$.

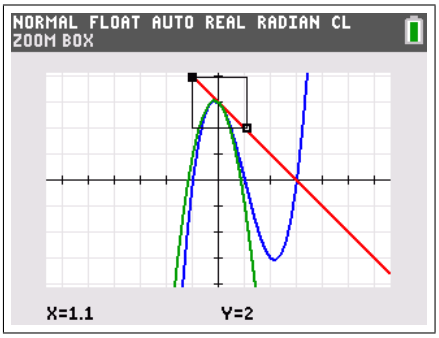
Oplossing. We maken de grafieken met de **TI-84 Plus Color** .

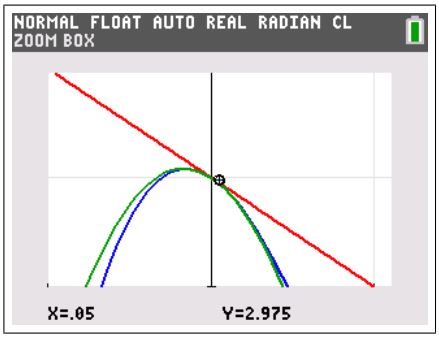




We kunnen nu zoom [ZBox] gebruiken om de grafiek te vergroten.



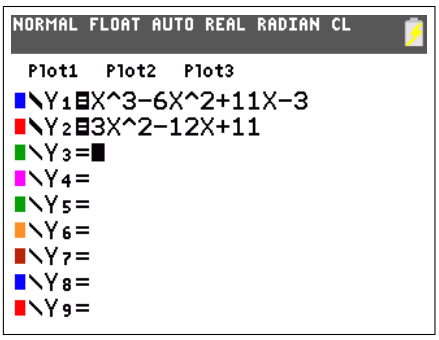


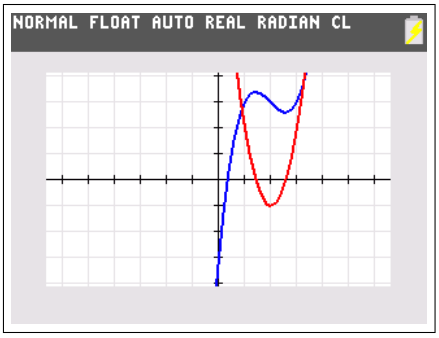


We zien dat $y = -x + 3$ de raaklijn is in $(0, 3)$ en dat de $y = -3x^2 - x + 3$ de raakparabool is in datzelfde punt. Dit is algemeen geldig en een onmiddellijk gevolg van de stelling van Taylor.

Opdracht 6. Gebruik de **TI-84 Plus Color** om de grafiek van een functie in het blauw te tekenen en haar afgeleide functie in het rood. Wat is het verband tussen nulwaarden en extrema van de afgeleide en de grafiek van de oorspronkelijke functie?

Oplossing. We beschouwen het voorbeeld $f(x) = x^3 - 6x^2 + 11x - 3$. De afgeleide functie is $f'(x) = 3x^2 - 12x + 11$. Dit komt in een andere kleur.





We zien dat de (enkelvoudige) nulwaarden van $f'(x)$ overeenkomen met de extrema $f(x)$ en dat de extrema van $f'(x)$ overeenstemmen met de buigpunten van $f(x)$.

Opdracht 7. Volgende stelling laat zien dat elke functie de som is van een even en een oneven functie.

Stelling. *Zij $f : \mathbb{R} \to \mathbb{R}$ een functie. Stel*

$$f_e(x) = \frac{f(x) + f(-x)}{2} \quad \textit{en} \quad f_o(x) = \frac{f(x) - f(-x)}{2}$$

Dan is f_e even, f_o oneven en $f(x) = f_e(x) + f_o(x)$.

Bewijs deze stelling. Werk enkele voorbeelden met veeltermen uit. Waar komt de benaming 'even' en 'oneven' vandaan? Wat gebeurt er indien f zelf al even of oneven is, bijvoorbeeld $f(x) = \sin x$ of $f(x) = \cos x$? Gebruik de **TI-84 Plus Color** om de grafieken van f, f_e en f_o te maken voor een willekeurige f. Bekijk de voorbeelden $f(x) = e^x$ en $f(x) = |x + 2|$.

Oplossing. Het bewijs van de stelling is een eenvoudige verificatie. Wanneer men echter de stelling als volgt formuleert, is het bewijs uitdagender en leidt dit vanzelfsprekend tot bovenstaande formules. Dit kan gegeven worden in de betere klassen.

Stelling. *Elke functie is de som van een even en een oneven functie.*

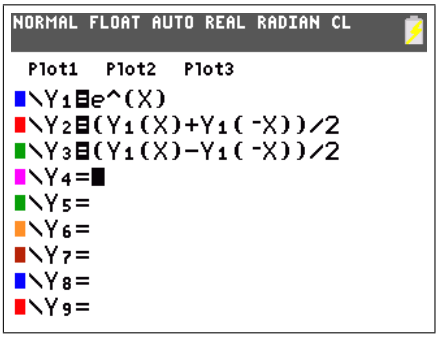
Bewijs. Stel dat $f(x) = f_e(x) + f_o(x)$ waarbij f_e even is en f_o oneven. Dan bekomen we:

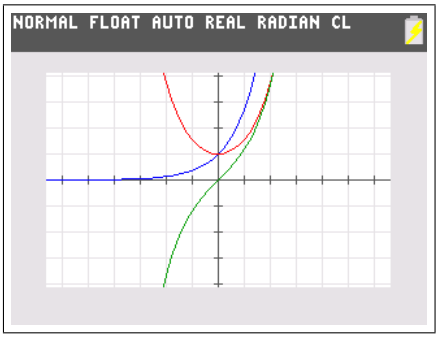
$$\begin{cases} f(x) & = & f_e(x) + f_o(x) \\ f(-x) & = & f_e(x) - f_o(x) \end{cases}$$

Oplossen van dit stelsel naar f_e en f_o levert de bovenstaande formules. □

Als $f(x)$ een veelterm is, dan bestaat $f_e(x)$ uit alle termen van even graad en $f_o(x)$ uit alle termen van oneven graad. Indien f reeds even is zal $f_e(x) = f(x)$ en $f_o(x) = 0$. Eenzelfde conclusie kan getrokken worden indien f oneven is.

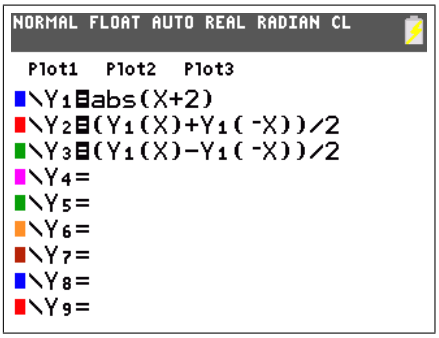
Deze stelling door de **TI-84 Plus** laten illustreren kan op eenvoudige wijze. We beschouwen de functie $f(x) = e^x$

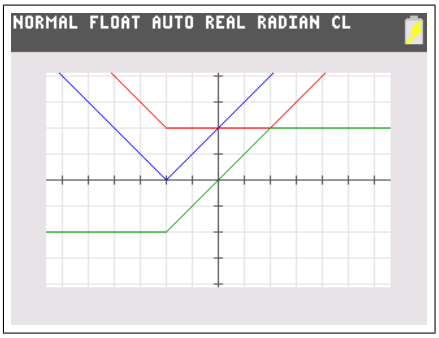




De bovenstaande aanpak laat ook toe om te spreken over sinh en cosh. Alhoewel deze functies vaak als 'vergezocht' worden betiteld, hebben ze vele toepassingen bijvoorbeeld in de fysica (hangende ketting) of in de architectuur en de kunst (gewelven van Gaudi).

Indien $f(x) = |x + 2|$ de functie is, hebben we het moeilijk om een voorschrift voor f_e en f_o te vinden. De grafiek maken met de **TI-84 Plus Color** geeft ons:

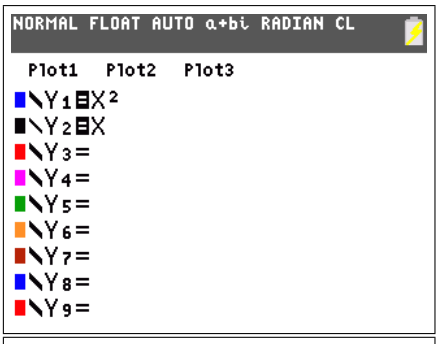


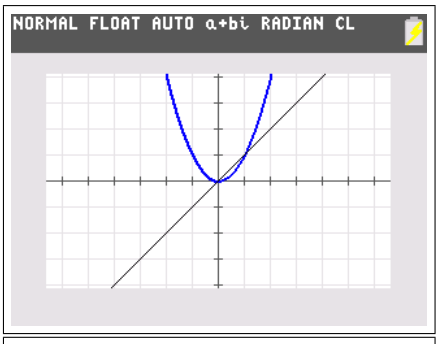


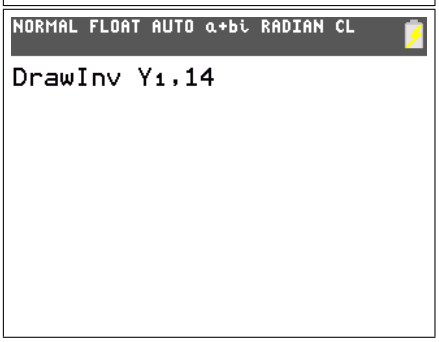
Hieruit kan je gemakkelijk een meervoudig voorschrift aflezen voor f_e en f_o.

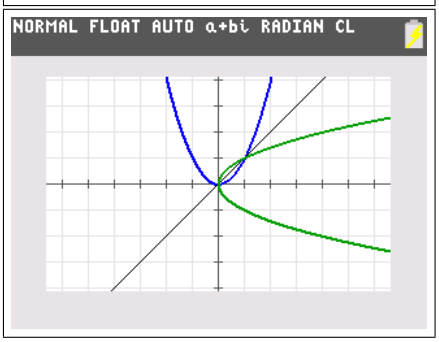
Opdracht 8. Maak de grafiek van een functie (bijvoorbeeld $y = x^2$ of $y = \cos x$) en van de eerste bissectrice. Gebruik 2nd [draw] [drawinv] om de grafiek volledig te spiegelen. Je hebt nu de inverse *relatie* getekend. Is dit de grafiek van een functie? Teken nu de inverse functie (bijvoorbeeld $y = \sqrt{x}$ of $y = \text{Bgcos}\ x$). Begrijp je nu waarom men voor de inverse *functie* het domein en het bereik moet beperken?

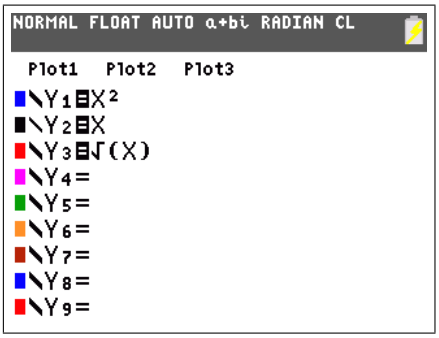
Oplossing. Maak de grafieken als volgt:

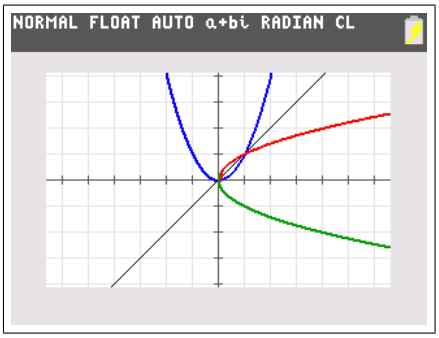








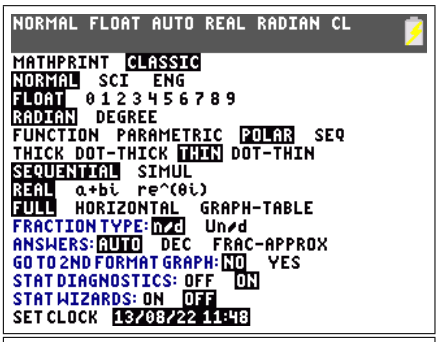


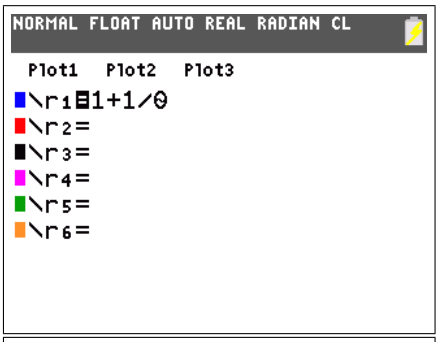


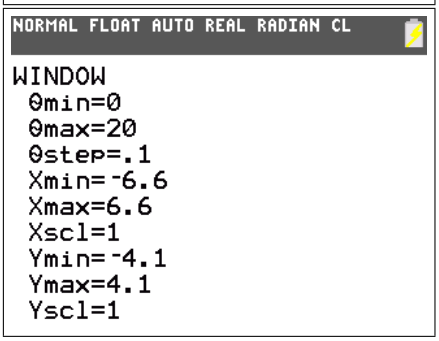
Men ziet duidelijk dat de grafiek van de inverse *relatie* de spiegeling is van de grafiek van de functie t.o.v. de eerste bissectrice. In het algemeen echter is dit niet de grafiek van een functie. Indien men echter het domein en het bereik beperkt, bekomt men wel de grafiek van de inverse *functie*.

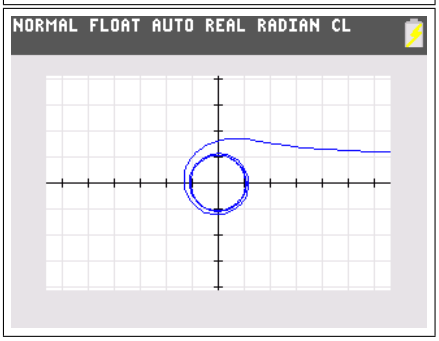
Opdracht 9. Gebruik de **TI-84 Plus Color** om de grafiek van de kromme $r(\theta) = 1 + \frac{1}{\theta}$, $\theta \in]0, +\infty[$ (in poolcoördinaten) te maken. Teken ook de asymptoten van deze kromme.

Oplossing. We maken eerst de grafiek met de **TI-84 Plus Color** . Eerst stellen we poolcoördinaten in via `mode` [pol] en kiezen `zoom` [zdecimal]. We passen met `window` de grenzen van θ aan. We kunnen nu de grafiek maken.









Leerlingen zullen waarschijnlijk wel een horizontale asymptoot $y = 1$ herkennen voor $x \to +\infty$. Verder is er ook een "asymptotische cirkel" die zichtbaar wordt tijdens het tekenen.

Bevestiging van de horizontale asymptoot kan men krijgen door volgende limieten uit te rekenen.

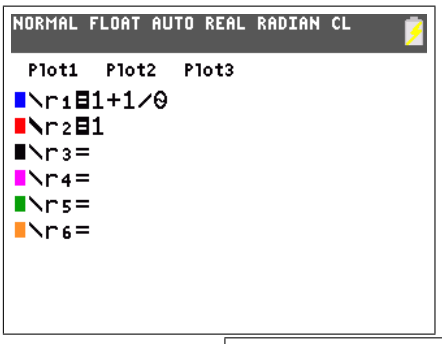
$$\lim_{\theta \overset{\to}{>} 0} x = \lim_{\theta \overset{\to}{>} 0} r \cos\theta = \lim_{\theta \overset{\to}{>} 0} (1 + \frac{1}{\theta}) \cos\theta = +\infty$$

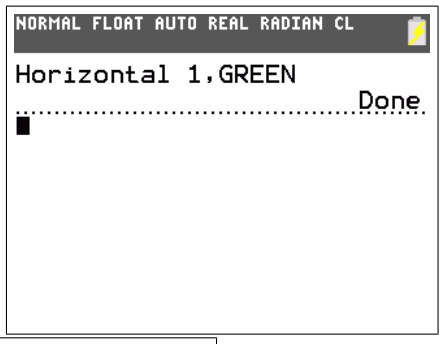
$$\lim_{\theta \overset{\to}{>} 0} y = \lim_{\theta \overset{\to}{>} 0} r \sin\theta = \lim_{\theta \overset{\to}{>} 0} (1 + \frac{1}{\theta}) \sin\theta = \lim_{\theta \overset{\to}{>} 0} \sin\theta + \lim_{\theta \overset{\to}{>} 0} \frac{\sin\theta}{\theta} = 1$$

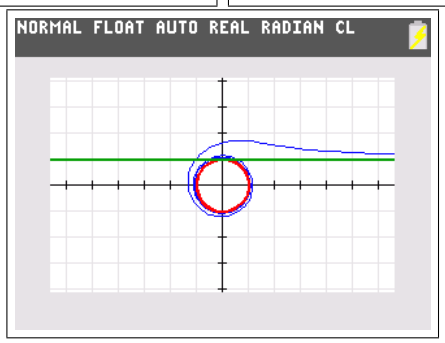
De "asymptotische cirkel" volgt uit het feit dat het punt (r, θ) in poolcoördinaten om de oorsprong blijft draaien indien $\theta \to +\infty$ en dat

$$\lim_{\theta \to +\infty} r = \lim_{\theta \to +\infty} 1 + \frac{1}{\theta} = 1$$

We kunnen deze beide asymptoten als volgt tekenen (we gebruiken hierbij `2nd` `[draw]` `[horizontal]` om de horizontale asymptoot te tekenen):







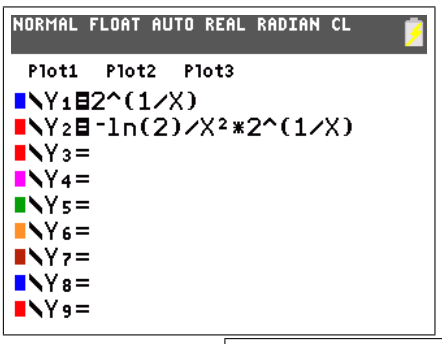
Zo zie je maar dat asymptotisch gedrag niet rechtlijnig hoeft te zijn. Een ander voorbeeld is de functie met voorschrift $f(x) = x^2 + \frac{1}{x}$ dat een asymptotisch parabolisch en hyperbolisch gedrag heeft.

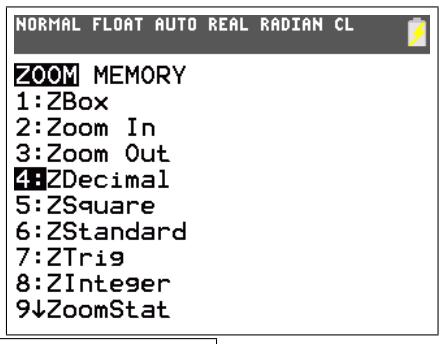
Opdracht 10. Maak de grafiek van de functie $f(x) = 2^{\frac{1}{x}}$. Gebruik de afgeleide om een idee te bekomen van het asymptotisch gedrag. Reken nadien de nodige limieten uit ter controle.

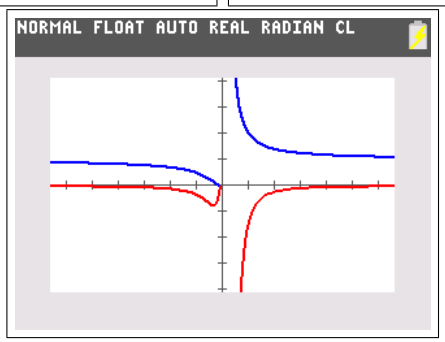
Oplossing. De afgeleide functie wordt gegeven door

$$f'(x) = -\frac{\ln 2}{x^2}\, 2^{\frac{1}{x}}$$

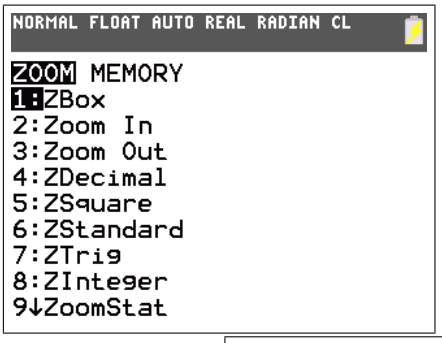
We maken nu de grafiek van f en f'.

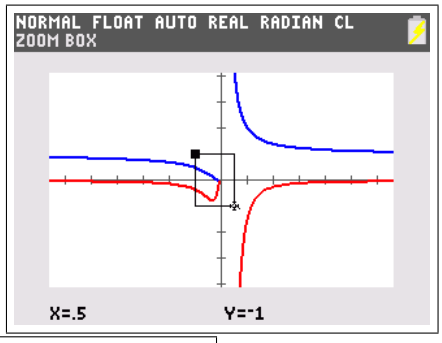


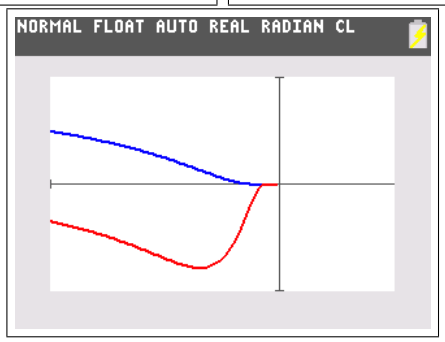




Het is duidelijk dat er een horizontale asymptoot is in $\pm\infty$, de afgeleide functie gaat immers naar 0 in $\pm\infty$. In 0 is de situatie tamelijk ongewoon. Voor positieve waarden zien we duidelijk een verticale asymptoot $y = 0$, dit zien we ook aan de afgeleide functie die naar $-\infty$ duikt. Voor negatieve waarden lijkt de grafiek van f naar 0 te dalen. Indien we naar f' kijken, zien we echter dicht bij 0 een duidelijk minimum, hetgeen eraan doet denken dat het gedrag van f misschien ingewikkelder is dan eerst gedacht. We gebruiken `zoom` `[zbox]` om de grafieken te vergroten.







De grafiek van f daalt inderdaad naar 0, maar er is een buigpunt (het minimum van f') waarna de functie op horizontale wijze de oorsprong nadert.

Deze resultaten bekomen we natuurlijk ook door de passende limieten uit te rekenen.

$$\lim_{x\to\pm\infty} 2^{\frac{1}{x}} = 2^0 = 1$$

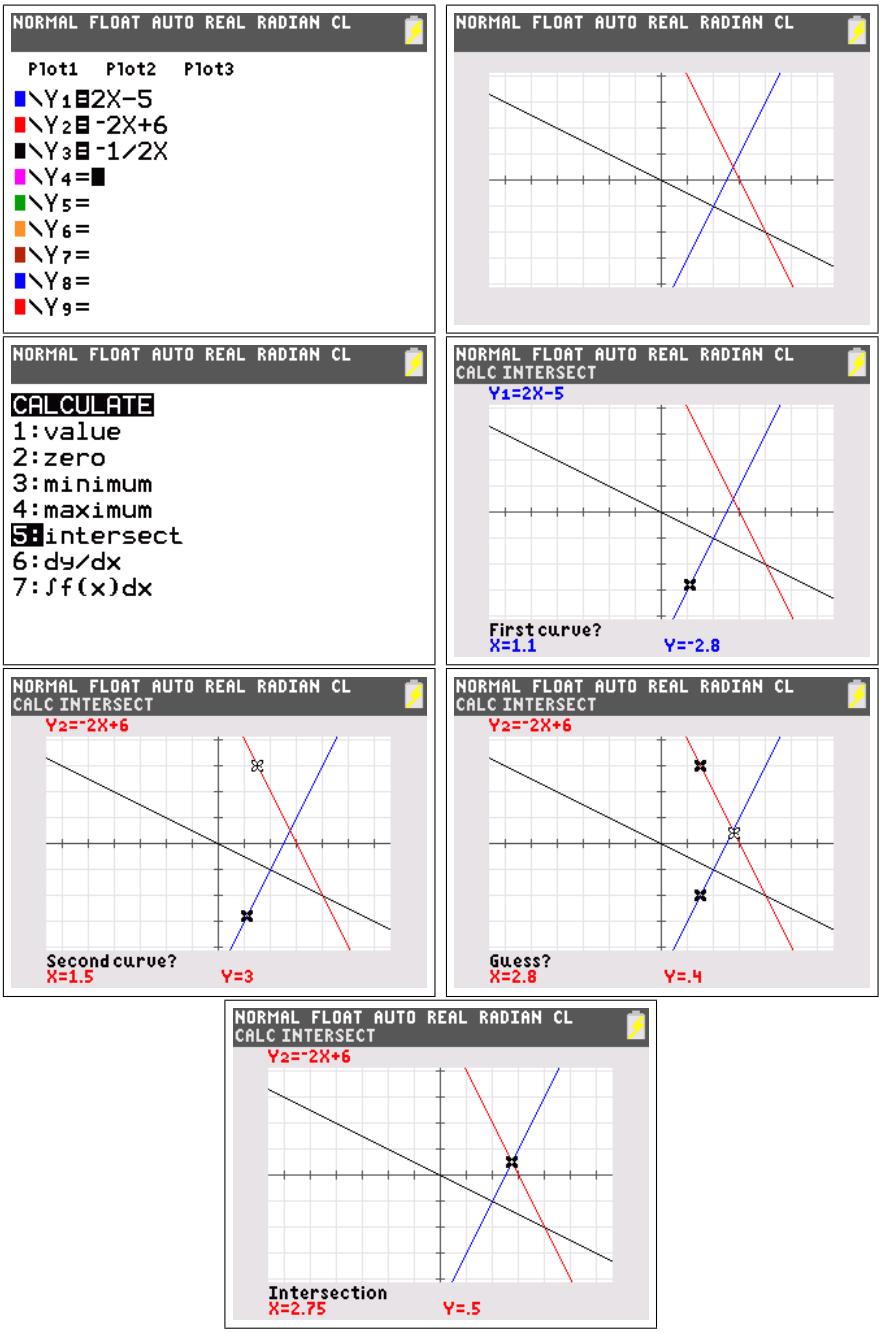
$$\lim_{x\overset{>}{\to}0} 2^{\frac{1}{x}} = 2^{+\infty} = +\infty$$

$$\lim_{x\overset{<}{\to}0} 2^{\frac{1}{x}} = 2^{-\infty} = 0$$

2.2 Meetkunde

Opdracht 11. Bepaal het zwaartepunt van de driehoek, die begrensd wordt door de rechten $y = 2x - 5$, $y = -2x + 7$ en $y = -\frac{1}{2}x$.

Oplossing. De opeenvolgende grafieken worden in een andere kleur getekend. Via `2nd` `[calc]` `[intersect]` kan je de eerste rechte en dan de tweede selecteren. Daarna geef je een eerste benadering voor het snijpunt. De **TI-84 Plus Color** berekent dan numeriek het snijpunt.

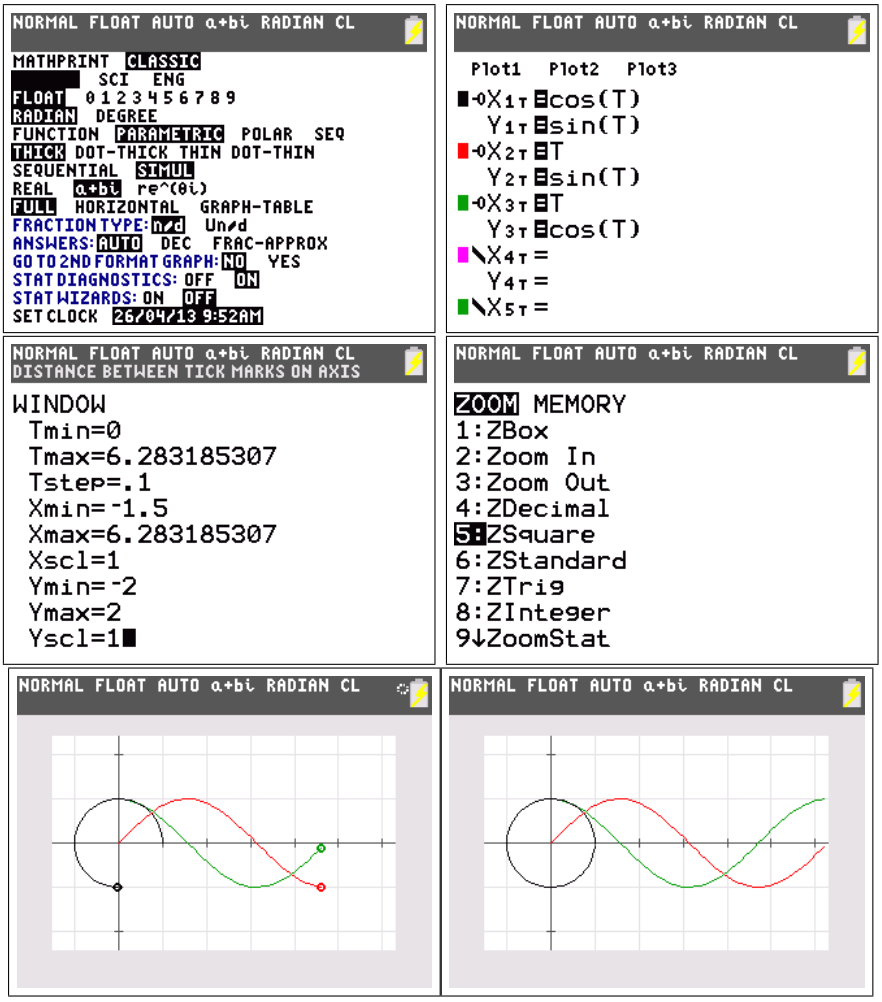


De snijpunten zijn $(\frac{11}{4}, \frac{1}{2})$, $(2, -1)$ en $(4, -2)$. Het zwaartepunt is dus $(\frac{35}{12}, -\frac{5}{6})$. Merk hierbij op dat de omzetting naar breuken nodig is, anders zit men met benaderde waarden.

2.3 Goniometrie

Opdracht 12. Een mooie oefening voor het begrijpen van de goniometrische cirkel en de goniometrische getallen is het maken van volgende grafieken. Stel met `mode` [radian], [simul], [connected] en [par] in. Voer dan de goniometrische cirkel en de goniometrische functies in (let op de tekenstijl). Gebruik `window` en stel de parameter t zo in dat die van 0 tot 2π gaat, pas vervolgens de assen aan. Maak de tekening via `zoom` [ZSquare]. Met `enter` kun je pauzeren. Gebruik tenslotte `trace` en de pijltjestoetsen om de corresponderende punten op de goniometrische cirkel en op de goniometrische krommen te tonen.

Oplossing. Het resultaat is een soort animatie.



Vergeet niet om achteraf de opties via `mode` terug te zetten.

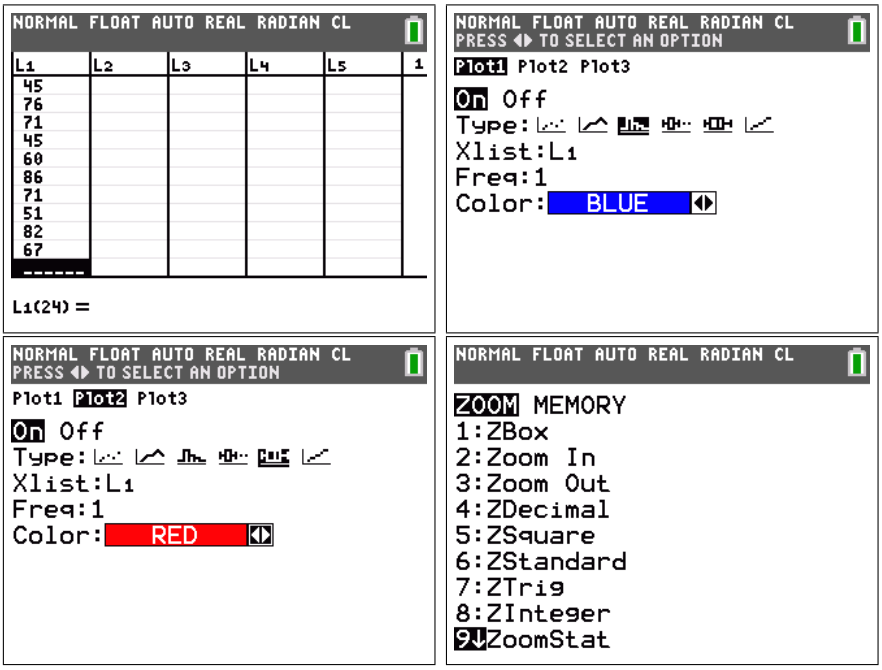
2.4 Statistiek

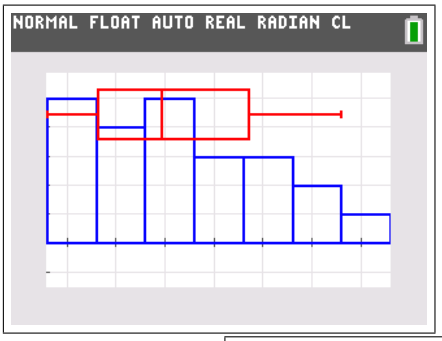
Opdracht 13. Leerlingen krijgen vaak resultaten van toetsen, rapporten of examens. Deze punten worden bekomen door de cijfers van de individuele vragen op te tellen. De centrale limietstelling zegt dat een som van dergelijke gelijkverdeelde en onafhankelijke toevalsveranderlijken normaal verdeeld is. Vandaar dat wij als leerkrachten in ons puntenboekje graag de Gauss-curve zien opduiken. Onderstaande cijfers waren de resultaten van een 5de jaar op het examen wiskunde in december.

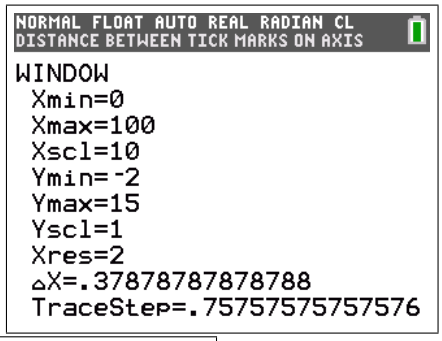
80	59	63	51	45	51
52	61	53	45	60	82
62	73	49	76	86	67
79	52	52	71	71	

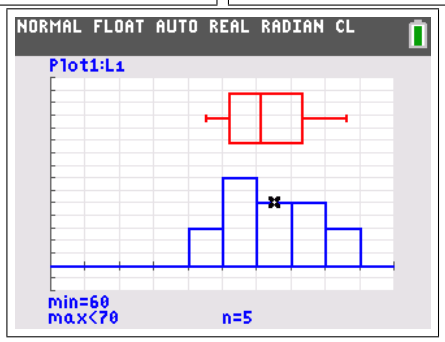
Gebruik de **TI-84 Plus Color** om deze gegevens grafisch te analyseren en vergelijk met de passende normale verdeling.

Oplossing. We geven eerst alle metingen in via `stat` [edit] en stellen daarna [stat plot1] in om een histogram te maken en [stat plot2] om een boxplot te verkrijgen. Deze worden getekend via `zoom` [zoomstat]. Uiteindelijk worden de grenzen en de klassebreedte via `window` ingesteld.

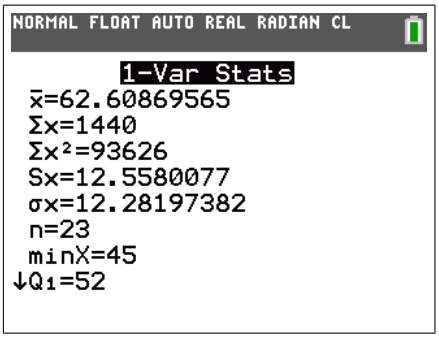


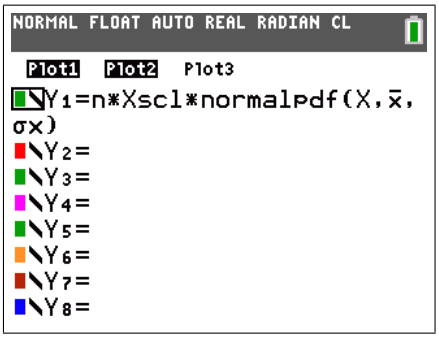


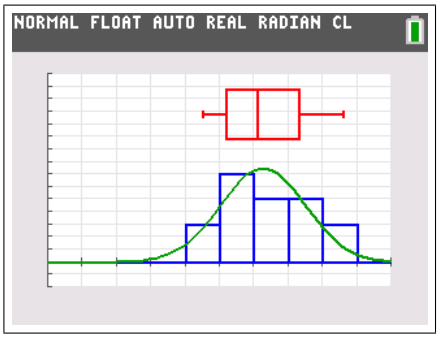




Om te vergelijken met een normale verdeling gebruiken we `[normalpdf]` om de dichtheidsfunctie van de normale verdeling te bekomen. De totale oppervlakte hieronder is altijd 1. De oppervlakte van het histogram bedraagt `n*Xscl` (aantal metingen $\times$ klassebreedte), we zullen dit dus als voorfactor gebruiken. Verder hebben we ook de parameters $\mu = \bar{x}$ en $\sigma = \sigma_x$ nodig. Deze bepalen we via `[1-var stat]`, dit moet gebeuren voordat de grafiek van y_1 gemaakt wordt. Hierin zijn immers $\bar{x}$ en σ_x nodig en deze zijn pas uitgerekend na het uitvoeren van `[1-var stat]`.







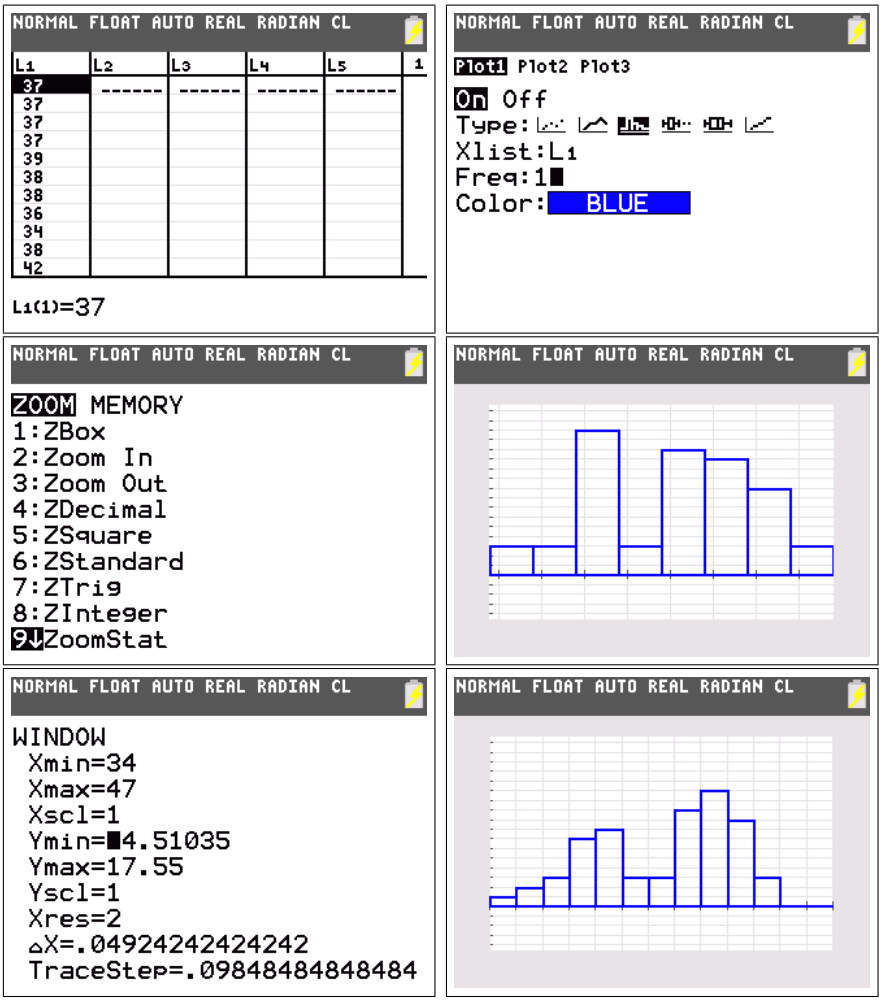
We zien dat de punten van deze klas inderdaad tamelijk goed overeenstemmen met een normale verdeling. Wie echt wil weten hoe goed een reeks metingen past bij een normale verdeling moet de χ^2-goodness-of-fit gebruiken (stat [tests] [χ^2GOF-Test]).

Opdracht 14. In een 4de jaar werd gevraagd aan alle 15 leerlingen om aan vier vriendjes een enquête te laten invullen. Hierbij werd o.a. gevraagd naar leeftijd, geslacht, gewicht, lengte en schoenmaat. Dit waren de metingen voor de schoenmaten:

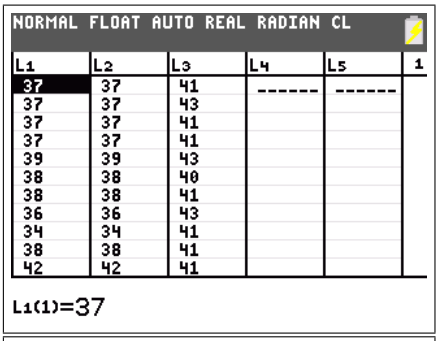
37	38	37	41	42	42
37	36	38	43	43	41
37	38	35	41	40	44
39	38	35	41	42	43
38	38	41	41	43	42
38	39	43	41	42	43
36	39	41	42	42	42
34	37	41	42	40	43
38	36	43	42	43	44
42	37	40	41	42	44

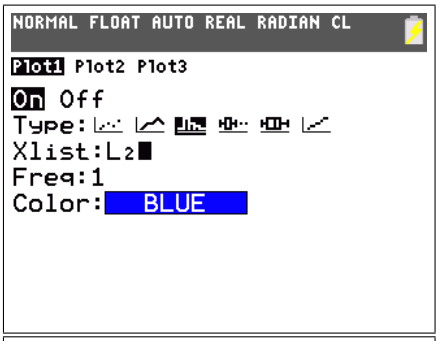
Bestudeer deze resultaten aan de hand van de nodige histogrammen.

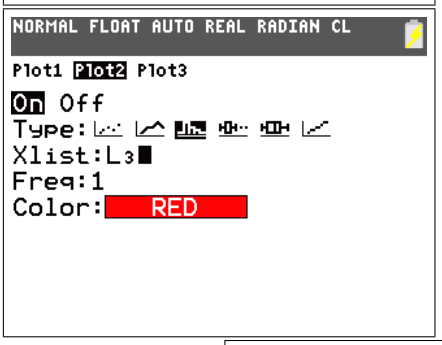
Oplossing. We geven eerst alle metingen in via stat [edit] en stellen daarna [stat plot1] in om een histogram te maken. Deze wordt getekend via zoom [zoomstat]. Uiteindelijk worden de grenzen en de klassebreedte via window ingesteld.

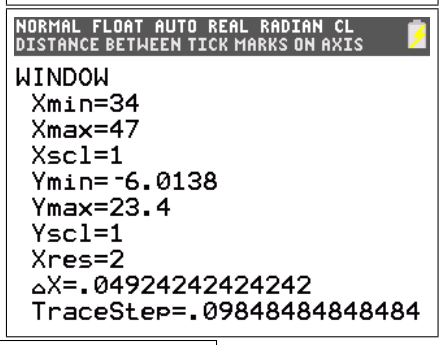


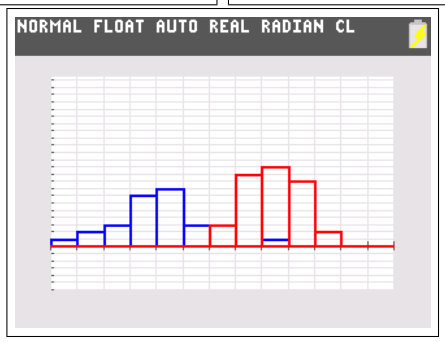
Dit histogram toont duidelijk dat deze verdeling niet *uni*modaal maar *bi*modaal is. De leerlingen zullen al snel het geslacht als reden hiervoor aanhalen. Om dit te controleren turven we opnieuw en maken twee lijsten aan, eentje voor de meisjes en eentje voor de jongens. We maken dan beide histogrammen samen. (Gemakshalve werden de bovenstaande metingen per geslacht geordend, de eerste 25 zijn afkomstig van meisjes, de overige metingen komen van jongens.)



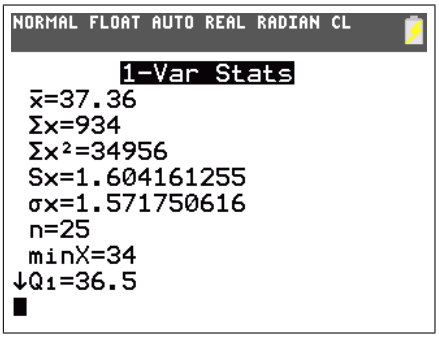


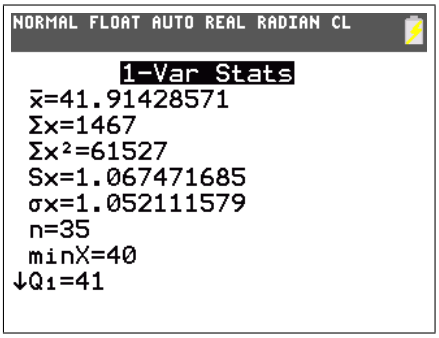






We zien nu duidelijk dat beide pieken van het oorspronkelijk histogram afkomstig zijn van twee afzonderlijke histogrammen. De oorzaak is inderdaad het geslacht. Aan de hand van de statistische kengetallen kan je zien dat de gemiddelden aanzienlijk verschillen en dat de spreiding bij de jongens kleiner is.

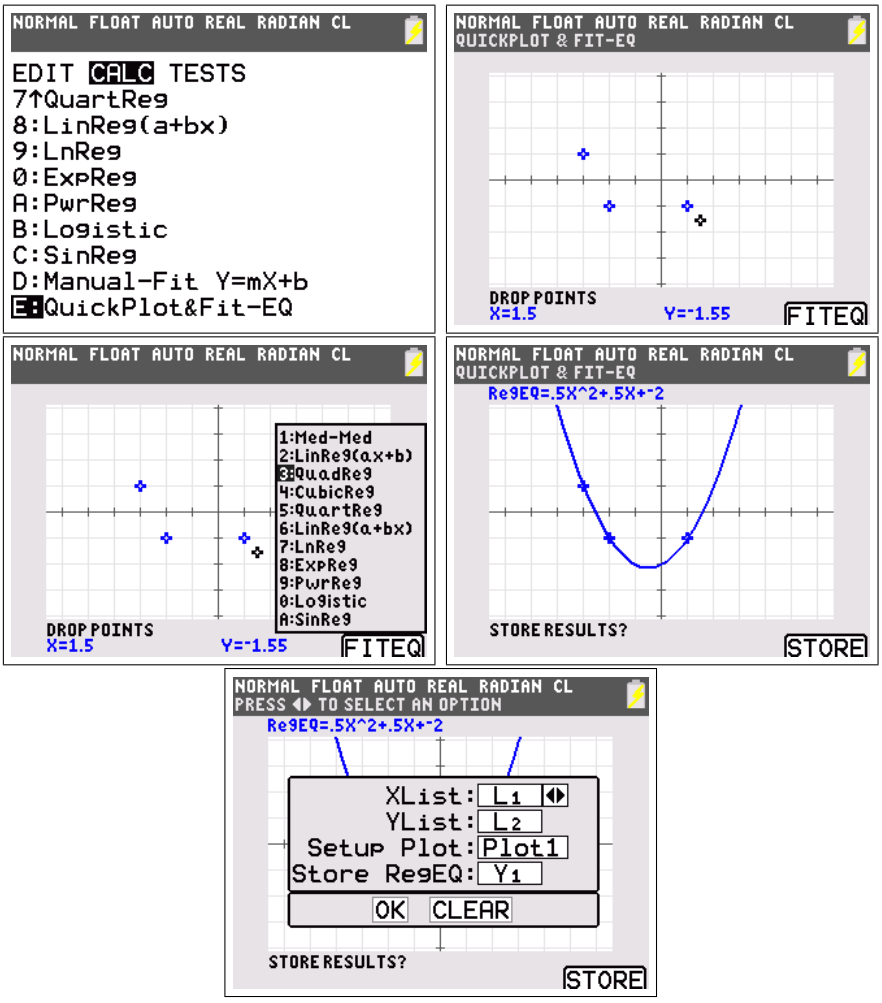




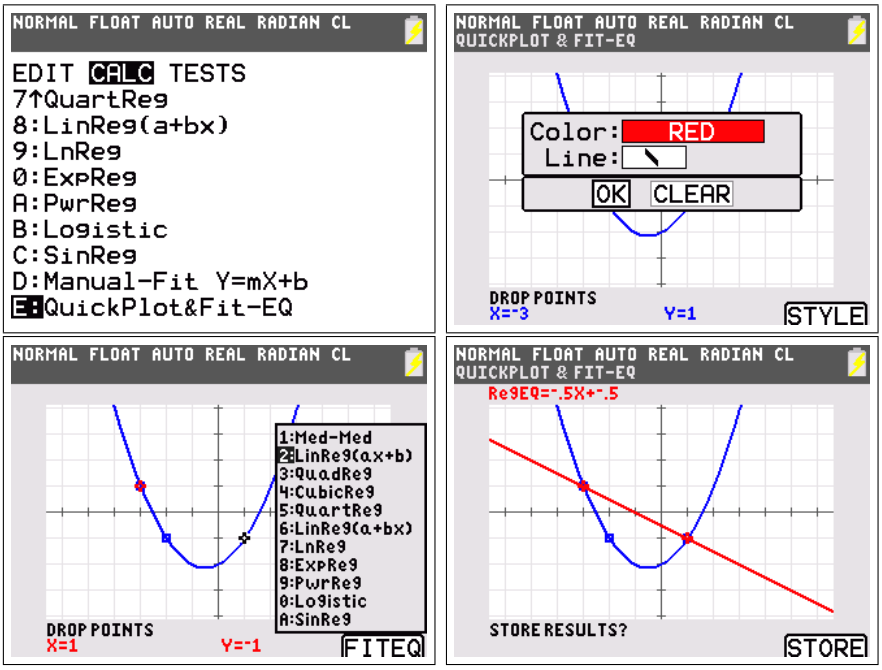
Dit fenomeen komt zeer vaak voor in de statistiek. Bij bovenstaande gegevens speelt duidelijk het geslacht een rol, maar leerlingen kunnen ook de vrijetijdsbesteding of de werkdruk bekijken. In deze gevallen zal de leeftijd soms het verschil maken. Vaak krijg je dan een *tri*modale verdeling onder invloed van kinderen - ouders - grootouders.

Opdracht 15. Gegeven zijn de punten $A(-3,1)$, $B(-2,-1)$ en $C(1,-1,)$. Bepaal de vergelijking van de parabool door deze drie punten. Bepaal ook de vergelijking van de rechte door A en C. Wat als we een rechte proberen te vinden door alledrie de punten?

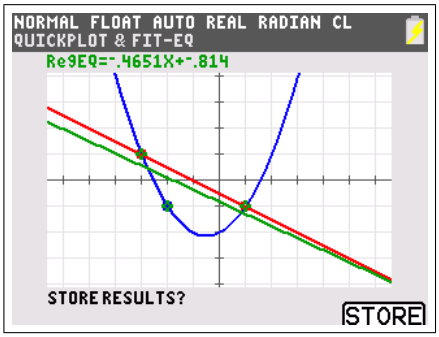
Oplossing. Via stat [calc] [Quickplot&Fit-EQ] kom je in een applicatie terecht waar je gemakkelijk de vergelijkingen kan vinden van krommen door gegeven punten. Eerst geef je de punten in, daarna kies je de gepaste regressie (kwadratisch in ons geval). We vinden de vergelijking $y = \frac{1}{2}x^2 + \frac{1}{2}x - 2$. Je kan tenslotte opteren om de punten en het resultaat op te slaan.



Om de vergelijking van de rechte AC te vinden, gebruiken we opnieuw [Quickplot&Fit-EQ]. We passen eerst de kleur aan en voeren de twee punten in. Daarna passen we een lineaire regressie toe. De vergelijking van AC is $y = -\frac{1}{2}x - \frac{1}{2}$.



Indien we een lineaire regressie voor drie punten invoeren, bekomen we de regressierechte, i.e. de beste rechte door deze puntenwolk.



Hoofdstuk 3

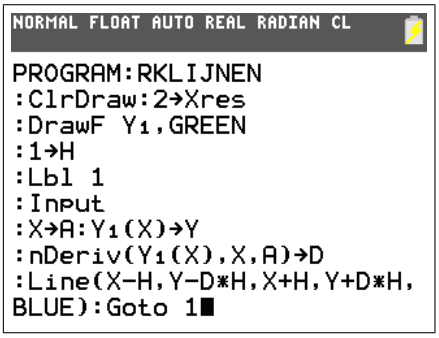
Programmeren

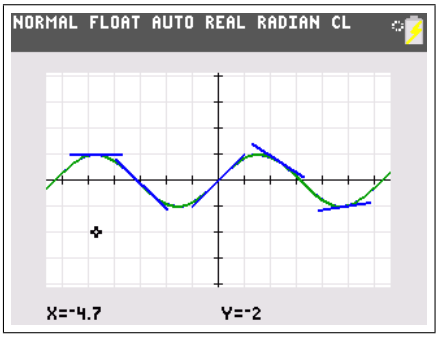
Het kleurenpalet van de **TI-84 Plus Color** is goed te gebruiken wanneer men programmeert.

3.1 Raaklijnen

Opdracht 16. Schrijf een programma dat de grafiek van een functie maakt in het groen en vervolgens de gebruiker vraagt om een punt te kiezen. In dit punt wordt dan een raaklijntje getekend in het blauw.

Oplossing. We gebruiken het commando `[input]` zodat de gebruiker een punt op het grafisch scherm kan kiezen. Het programma ziet er als volgt uit.





3.2 Families van krommen

Opdracht 17. Een familie krommen is een reeks krommen gegeven door een vergelijking met een parameter. Onderzoek volgende families krommen:

(i) de rechtenfamilie $y = 2x + A$

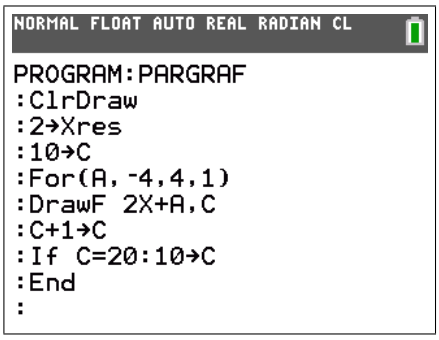
(ii) de rechtenfamilie door $(1, 2)$

(iii) de parabolenfamilie met nulwaarden 1 en 2

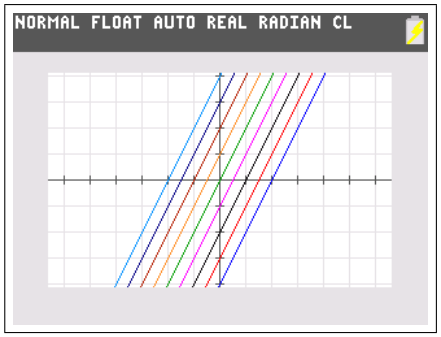
(iv) de parabolenfamilie met top in $(-2, 1)$

(v) de waterfontein $y = -x^2 + Ax$

Oplossing. We schrijven eerste een klein programmaatje dat in een `for`-lus de parameter laat variëren alsook de kleur C. We laten deze vanaf kleurwaarde 10 stijgen tot 20, daarna wordt C terug naar 10 gebracht.



(i) De rechten uit de familie $y = 2x + A$ zijn allen evenwijdig.



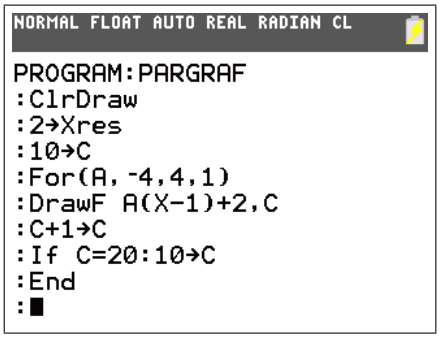
(ii) De algemene vergelijking van een rechte door een punt wordt gegeven door

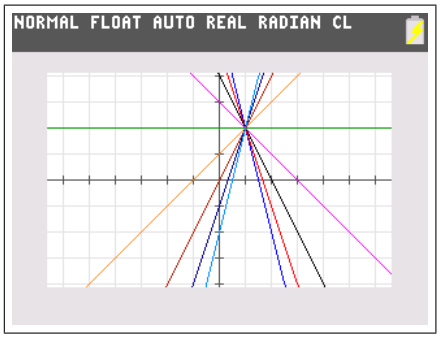
$$y = m(x - x_1) + y_1$$

De vergelijking van de rechtenfamilie door $(1, 2)$ wordt dan gegeven door

$$y = A(x - 1) + 2$$

We passen het programma aan. Het resultaat is het volgende:



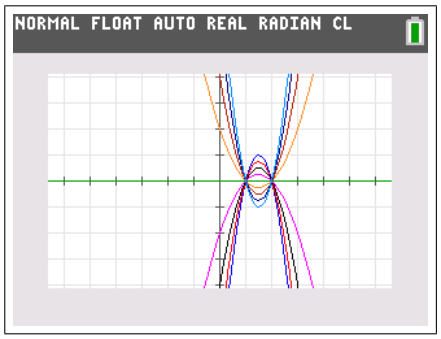


(iii) De algemene vergelijking van een parabool met nulwaarden x_1 en x_2 is

$$y = a(x - x_1)(x - x_2)$$

De parabolenfamilie met nulwaarden -1 en 2 wordt dus gegeven door

$$y = A(x - 1)(x - 2)$$

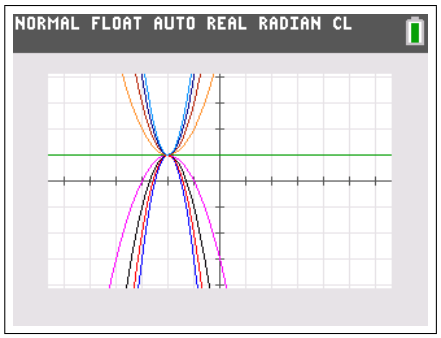


(iv) De vergelijking van een parabool met top (p, q) is

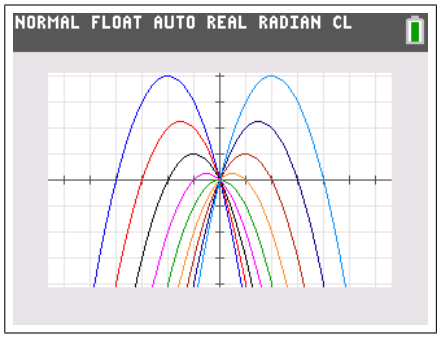
$$y = a(x - p)^2 + q$$

Zodat de parabolenfamilie beschreven wordt door

$$y = A(x + 2)^2 + 1$$



(v) Voor een parabool $y = ax^2 + bx + c$ is $y = bx + c$ de vergelijking van de raaklijn in $(0, c)$. Voor de waterfontein $y = -x^2 + Ax$ is de raaklijn in $(0, 0)$ dus de rechte $y = Ax$. De parameter A regelt dus de richting van een waterstraal.



Opdracht 18. Zoek zelf naar een familie krommen die een regenboog zal geven. Kan je de kleuren correct ordenen?

3.3 Matrices en emoticons

Eén van de belangrijkste toepassingen van matrices is beeldverwerking. Elk beeld, foto of icoon op een digitale media wordt intern voorgesteld als een matrix. Dankzij de kleuren van de **TI-84 Plus Color** kunnen we dit gemakkelijk illustreren.

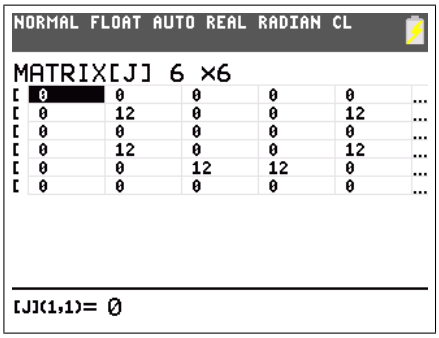
Opdracht 19. Gebruik 6×6 matrices om verschillende emoticons te maken. Zorg dat mond en ogen in twee afzonderlijke matrices zitten en dat de kleuren aangepast kunnen worden. Schrijf een programma om de icoontjes af te beelden.

Oplossing. We starten met een $0-1$ matrix waarin 0 betekent dat een pixel uit staat en 1 dat een pixel aan staat. De standaard *smiley* kan dus worden voorgesteld als de 6×6 matrix

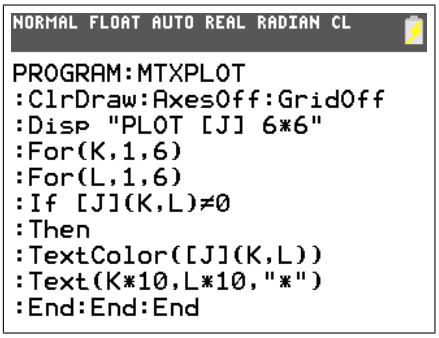
$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

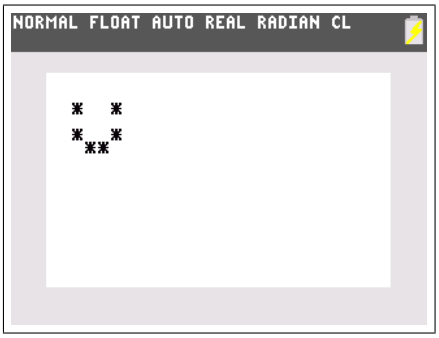
We willen deze door de **TI-84 Plus Color** laten tekenen. Pixels met waarde 0 worden niet getekend. De waarde van de andere pixels interpreteren we als

een kleur. Omdat de kleur zwart overeen komt met 12 op de **TI-84 Plus Color** (zie de tabel in de inleiding) zullen we niet de matrix A gebruiken, maar wel $J = 12A$.



Onderstaand programma tekent de 6×6 matrix `[J]`. Eerst wordt het grafisch scherm leeggemaakt en worden de assen verwijderd. Daarna wordt voor elk element in de matrix `[J]` nagegaan of die verschilt van 0. Indien dit zo is wordt de pixel getekend in de passende kleur met behulp van een `*`. 0mdat dit character 10 op 10 puntjes groot is op het scherm worden de coördinaten aangepast.



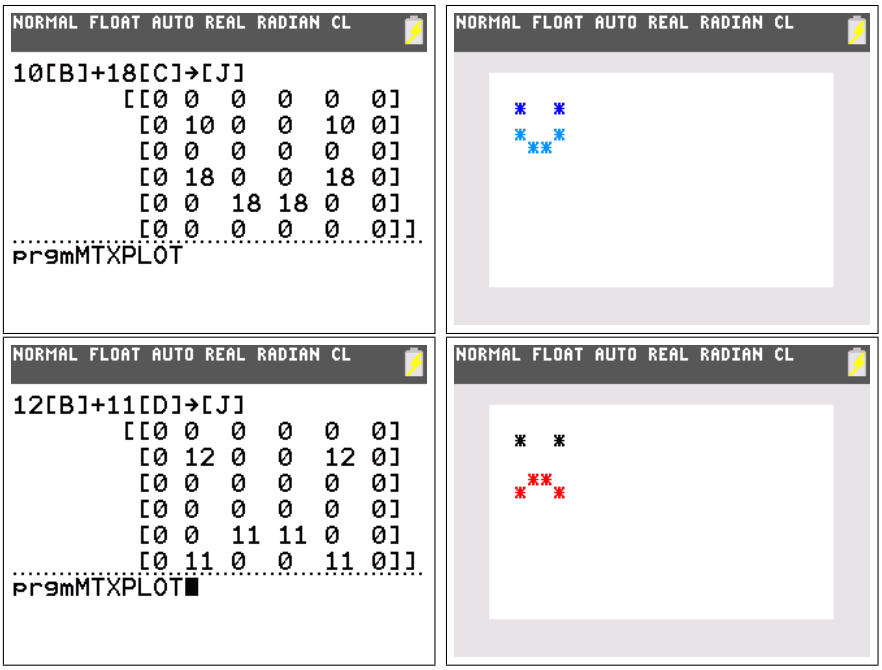


We wensen echter meerdere emoticons te maken en kleuren te kunnen gebruiken. Hiervoor zullen we de afzonderlijke delen van de emoticon in een aparte matrix steken.

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Met deze matrices kunnen we de emoticons samenstellen en verschillende kleuren gebruiken. De waarde 10 staat voor blauw en 18 voor lichtblauw.

De matrix $10B + 18C$ staat dus voor een *smiley* met blauwe ogen en een lichtblauwe glimlach. Door de matrix D te gebruiken kan ook een droevig gezicht gemaakt worden.



Opdracht 20. Zoek zelf de matrices om een kusjes-emoticon te maken en om een knipoog te geven.

Hoofdstuk 4

Uitgebreidere programeerprojecten

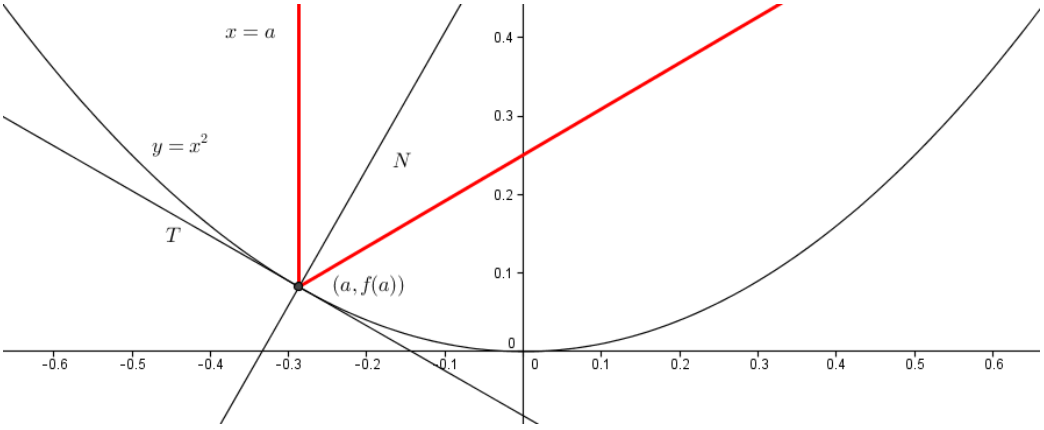
De hieronder beschreven onderwerpen (met programma's) vormen iets uitgebreidere opdrachten en kunnen gebruikt worden in het kader van de onderzoekscompetenties.

4.1 Kaustiek van een parabool

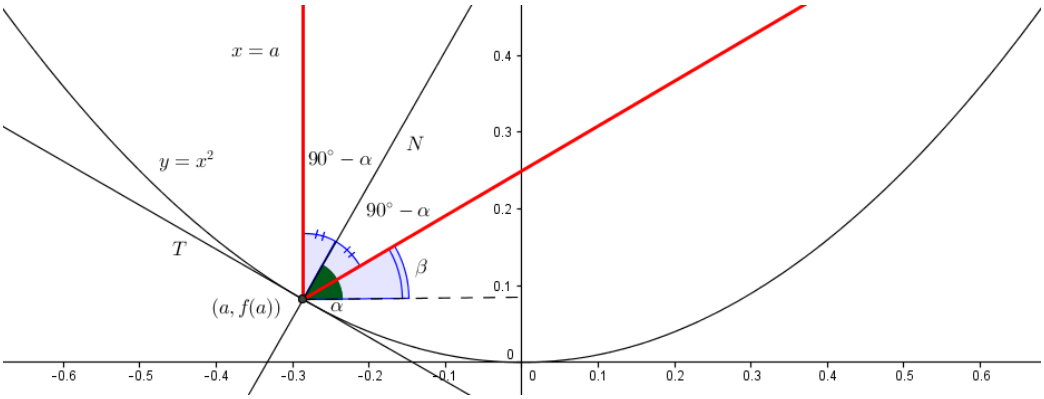
Wanneer lichtstralen worden weerkaatst, gebeurt het dat de weerkaatste stralen elkaar versterken op bepaalde plaatsen. Deze zullen dan oplichten, men noemt dit kaustieken. Je kan dit zien in een zwembad.

Een heel speciaal geval met belangrijke toepassingen is de parabolische spiegel. Wanneer een lichtstraal invalt, wordt deze weerkaatst volgens een

spiegeling t.o.v. de normaal.



We maken nu de nodige berekeningen om dit fenomeen te programmeren. Gemakshalve kiezen we de parabool gegeven door $f(x) = x^2$. We beschouwen de hellingshoeken van de verschillende rechten.



De hellingshoek van de normaal N noemen we α. Omdat de lichtstraal evenwijdig met de y-as invalt, weten we dat de hoek met N gelijk is aan $90° - \alpha$. De weerkaatste straal maakt dus dezelfde hoek met N. De hellingshoek van de uitgaande straal is dus

$$\beta = 90° - 2(90° - \alpha) = 2\alpha - 90°$$

Omdat we de vergelijking van de uitgaande straal nodig hebben, beschouwen we de richtingscoëfficiënten. De normaal N staat loodrecht op raaklijn T. We vinden dus als richtingscoëfficiënt

$$m_N = -\frac{1}{f'(a)} = -\frac{1}{2a} = \tan\alpha$$

De richtingscoëfficiënt van de uitgaande straal is dus

$$m = \tan\beta = \tan(2\alpha - 90^\circ) = -\tan(90^\circ - 2\alpha) = -\cot(2\alpha) = -\frac{1}{\tan(2\alpha)}$$

Gebruik makend van de formule $\tan(2\alpha) = \frac{2\tan\alpha}{1-\tan^2\alpha}$, vinden we

$$m = -\frac{1-\tan^2\alpha}{2\tan\alpha} = -\frac{1-\frac{1}{4a^2}}{-2\frac{1}{2a}} = a(1-\frac{1}{4a^2}) = a - \frac{1}{4a}$$

De vergelijking van de weerkaatste straal is gegeven door $y = m(x-a)+f(a)$ dus

$$y = (a-\frac{1}{4a})(x-a)+a^2 = (a-\frac{1}{4a})x - a^2 + \frac{1}{4} + a^2 = (a-\frac{1}{4a})x + \frac{1}{4}$$

In ons programma tekenen we dus

- de parabool $y = x^2$
- de invallende straal $x = a$ (via 2nd [draw] [vertical])
- de uitgaande straal $y = (a - \frac{1}{4a})x + \frac{1}{4}$

We gebruiken `[input]` om de gebruiker grafisch de invallende straal te laten bepalen en zorgen er opnieuw voor dat het kleurenpalet wordt doorlopen.

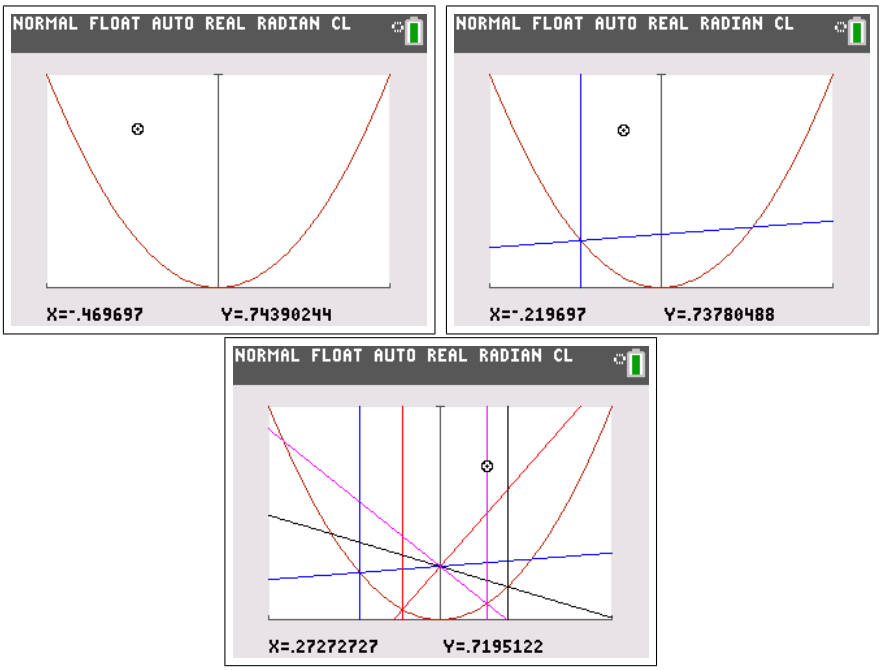
```
NORMAL FLOAT AUTO REAL RADIAN CL
PROGRAM:MIRPARA
:ClrDraw:2→Xres:10→C
:DrawF X²,BROWN
:Lbl 1
:Input
:X→A:Vertical A,C,1
:DrawF (A-1/(4A))X+1/4,C
:C+1→C
:If C=20:10→C
:Goto 1
```

We passen de grenzen van het grafisch venster aan om een goed beeld te krijgen en roepen dan het programma op.

```
NORMAL FLOAT AUTO REAL RADIAN CL
WINDOW
 Xmin=-1
 Xmax=1
 Xscl=1
 Ymin=0
 Ymax=1
 Yscl=1
 Xres=2
 ΔX=.00757575757575
 TraceStep=.01515151515151
```

```
NORMAL FLOAT AUTO REAL RADIAN CL
prgmMIRPARA
```

De gebruiker kan nu met de cursor bepalen waar de lichtstraal invalt.



We zien dat alle teruggekaatste stralen samenkomen in één enkel punt, het brandpunt van de parabool. Dit is de basis voor technologie zoals parabooltelescopen, schotelantennes of lichten van auto's.

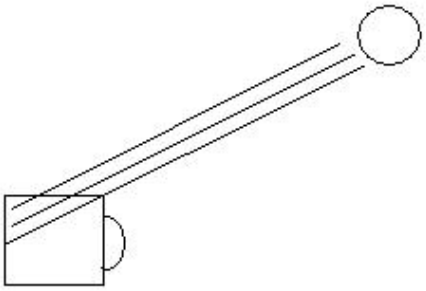
4.2 De cardioïde: over wiskunde, optica, kunst en timmeren

4.2.1 Kaustieken in een kopje

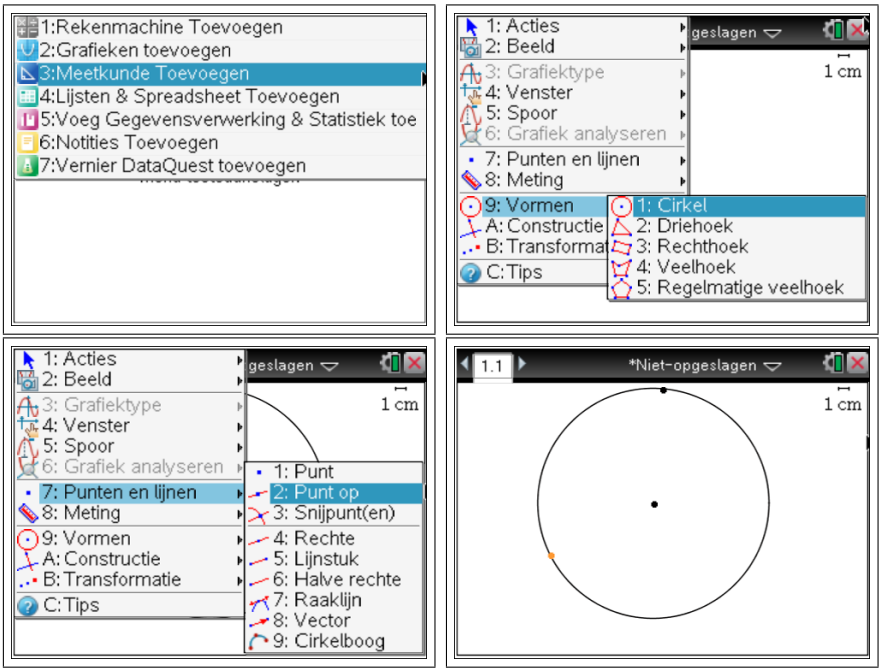
We zullen nu een ander geval van kaustieken beschouwen, namelijk het weerkaatste licht in een kopje. Dit voorbeeld biedt ons de mogelijkheid om verschillende didactische aanpakken te tonen en te vergelijken. Het is een onderwerp dat zich leent tot onderzoekscompetentie of ook gewoon als voorbeeld bij het onderwerp raaklijnen kan worden gebruikt.

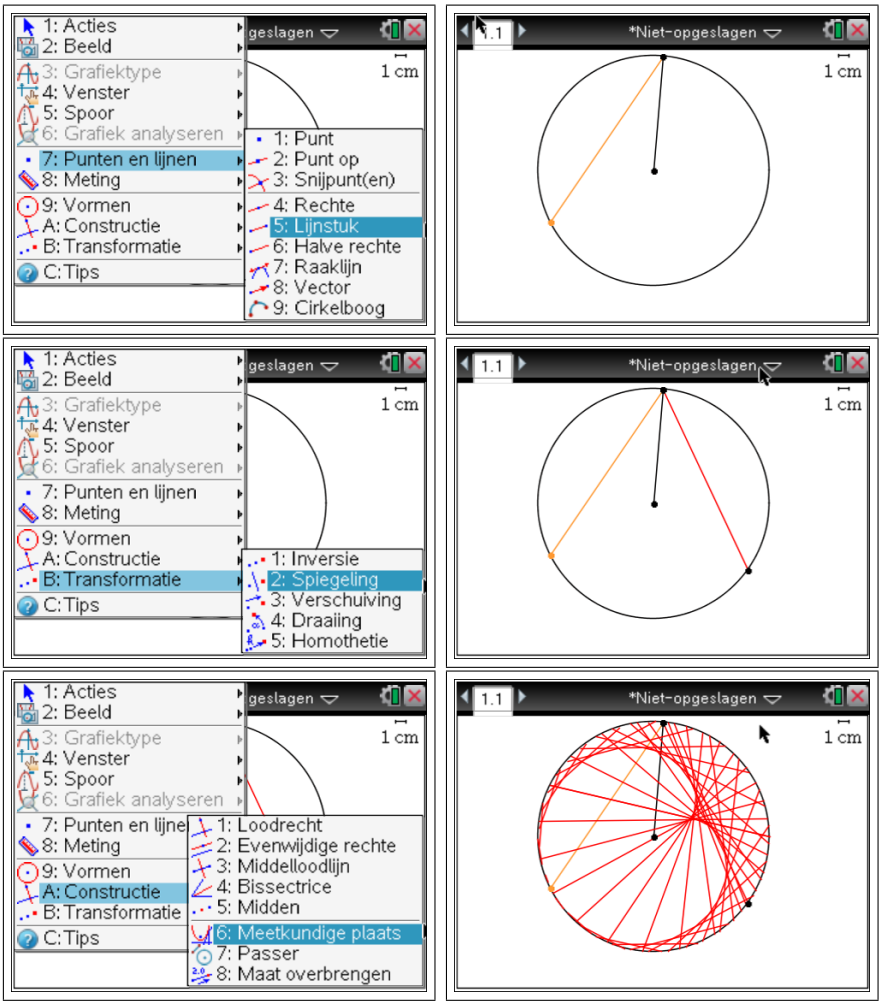
Simulatie

De probleemstelling is de volgende. Wanneer licht over de rand van een kopje invalt, wordt het weerkaatst. Wanneer men in het kopje kijkt, ziet men een kromme oplichten. Deze kromme is een cardioïde. Hoe komt die daar? Om dit na te gaan kan men met behulp van de **TI-Nspire** een simulatie doen.



We vertrekken van een cirkel. De lichtbron bevindt zich in een vast punt op de cirkel en een invallende straal gaat naar een ander punt op de cirkel. De straal wordt weerkaatst in dit punt en dus gespiegeld t.o.v. de normaal. We gebruiken nu de meetkundige plaats van de weerkaatste stralen afhangend van het invalspunt op de cirkel en bekomen zo alle teruggekaatste stralen.





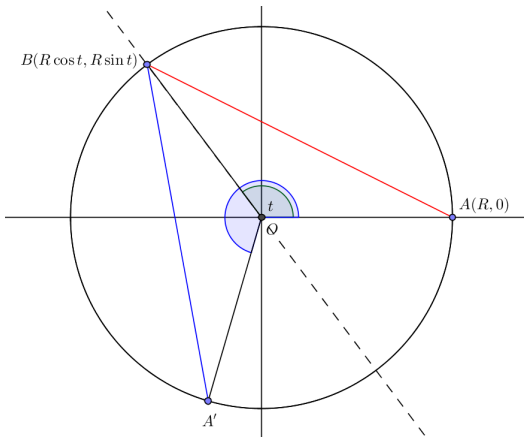
We zien dat alle teruggekaatste stralen samen de raaklijnen vormen aan een bepaalde kromme. Die kromme noemt men de **omhullende** van deze verzameling rechten. In ons geval is de omhullende een cardioïde. In werkelijkheid is er dus een ophoping van weerkaatste lichtstralen, de kromme licht dus op in het kopje.

Berekening

In oudere tijden, toen men bovenstaande simulatie met ict niet kon maken was men verplicht om berekeningen uit te voeren om de weerkaatste stralen te bepalen.

We vertrekken van een cirkel met straal R. De lichtbron bevindt zich in het punt $A(R, 0)$ en een invallende straal gaat naar een punt $B(R\cos t, R\sin t)$ op de cirkel. Het is gemakkelijk in te zien dat de normaal in dit punt eigenlijk

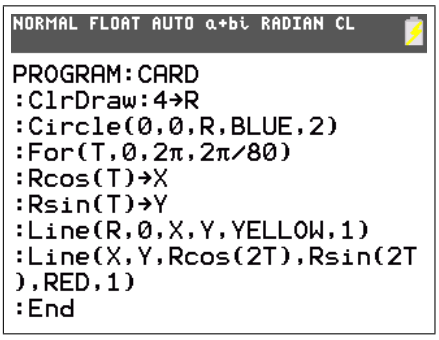
een straal is. We moeten nu het beeldpunt A' van A bepalen door spiegeling t.o.v. deze straal.

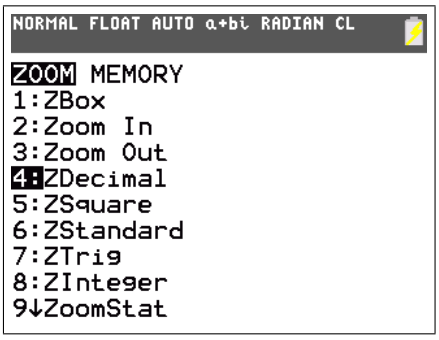


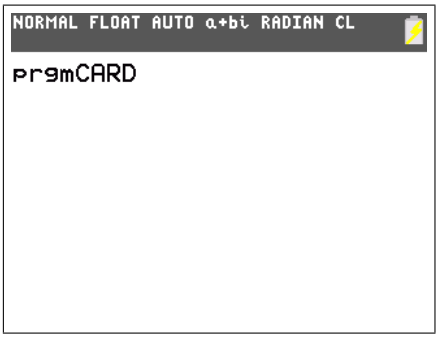
Omdat ΔOAB en $\Delta OA'B$ elkaars spiegelbeeld zijn, weten we dat in beide driehoeken de hoek O gelijk is. De aangeduide hoek $\widehat{AOA'}$ is dus $2t$. De coördinaten van A' zijn dan $(R\cos 2t, R\sin 2t)$.

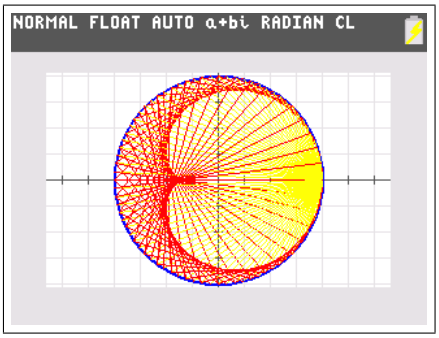
Programma

We schrijven nu een kort programma op de **TI-84 Plus Color** . Eerst wordt er een cirkel getrokken, daarna worden de invallende stralen als lijnstukken tussen $(R, 0)$ en het variabel punt $(x, y) = (R\cos t, R\sin t)$ getekend. Tegelijk worden de weerkaatste stralen in een andere kleur weergegeven, dit zijn de lijnstukken tussen (x, y) en $(R\cos 2t, R\sin 2t)$. We bekomen aldus de gezochte cardioïde als omhullende van de weerkaatste stralen.









Didactisch gezien is het maken van een simulatie met de huidige ICT-middelen uitstekend om het fenomeen te verklaren. Bovendien ziet de leerling hoe raaklijnen en normalen gebruikt kunnen worden om weerkaatste stralen in een kromme spiegel na te bootsen.

Wanneer men echter zelf het programma wil schrijven, is een grondigere kennis van wiskunde nodig. Volgende technieken werden expliciet gebruikt:

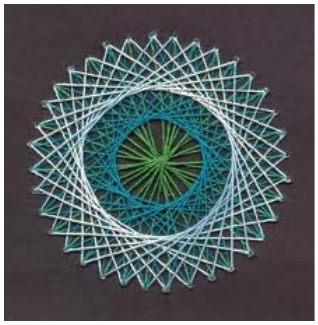
- het goed kiezen van een assenstelsel en van punten (probeer maar eens hetzelfde te doen indien de lichtbron niet in $(R, 0)$ zit)
- de goniometrische beschrijving van een cirkel
- de stelling die zegt dat een straal van een cirkel door een punt loodrecht staat op de raaklijn in dit punt
- de eigenschap over het behoud van hoekgrootten bij spiegelingen

Merk dus op dat de didactische doelen van beide methoden verschillen en dat het gebruik maken van ict er soms voor zorgt dat de achterliggende wiskunde verborgen wordt.

4.2.2 String art

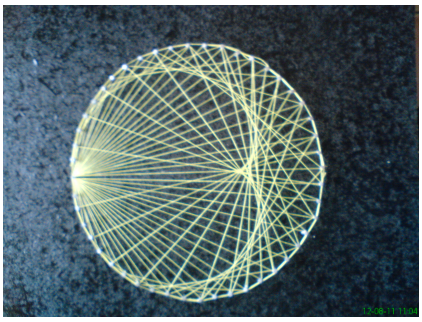
Knutselen

In de 'string art' worden vormen bekomen door touwtjes tussen spijkers te spannen. De vorm ontstaat meestal door de omhullende van de touwtjes en niet door de touwen zelf.



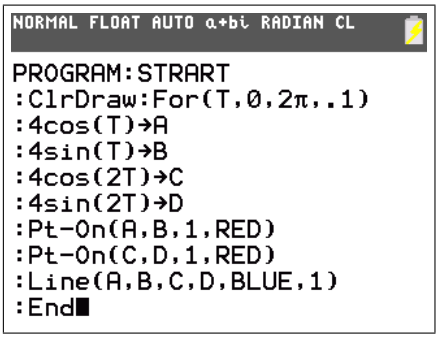
De 'string art' of 'curve stitching' is een techniek uitgevonden aan het einde van de 19de eeuw door Mary Everest Boole (1832 - 1916). Ze was één van de weinige vrouwelijke wiskundigen uit deze tijd. Haar man was de beroemde wiskundige George Boole die de boole algebra ontwikkelde. Ze ontwikkelde 'curve stitching' als een didactische methode om kinderen de wiskunde te laten ontdekken.

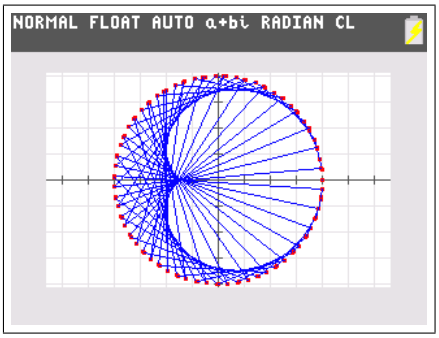
Wanneer we het bovenstaand voorbeeld van de cardioïde beschouwen, blijkt dit zeer vergelijkbaar met 'string art'. We verdelen onze spijkers gelijkmatig over een cirkel en verbinden de $t$de spijker met de $2t$de, i.e. het punt $(\cos t, \sin t)$ met $(\cos 2t, \sin 2t)$. Aldus bekomen we volgend resultaat.



Programma

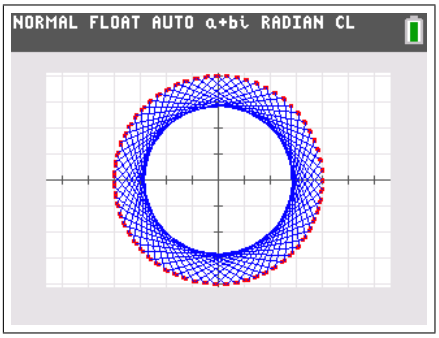
Het programma dat we reeds geschreven hebben, kan gemakkelijk aangepast worden om meerdere 'string-art' te maken. We tekenen punten voor de plaatsen van de spijkers en trekken met een andere kleur de lijnstukken. In onderstaand programma wordt telkens het lijnstuk getrokken van (A, B) naar (C, D). De posities van deze punten kunnen we willekeurig aanpassen.

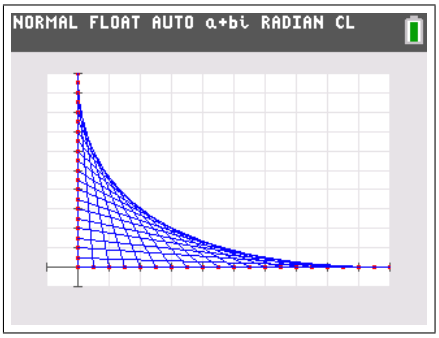




Je kan volgende voorbeelden uitwerken.

- $(A, B) = (\cos t, \sin t)$ en $(C, D) = (\cos(t + \phi), \sin(t + \phi))$
- $(A, B) = (\cos 2t, \sin 2t)$ en $(C, D) = (\cos 3t, \sin 3t)$
- $(A, B) = (t, 0)$ en $(C, D) = (0, 10 - t)$ met $0 < t < 10$





Op papier

Natuurlijk is het fantastisch om leerlingen te motiveren om met hamers en spijkers te werken in de klas ... maar dit is misschien niet zo praktisch! Je kan hetzelfde resultaat bereiken door potlood, papier en een meetlat te gebruiken!

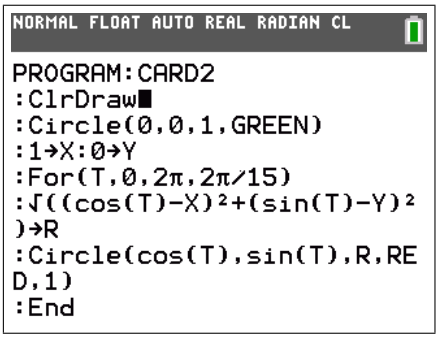
4.2.3 Een bijzondere familie cirkels

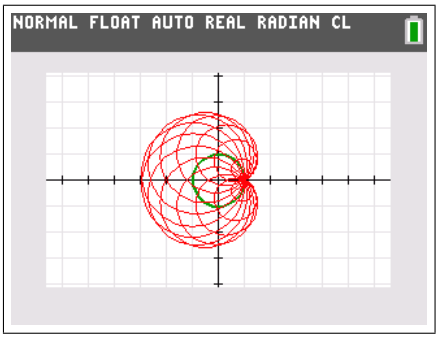
Beschouw alle cirkels waarvan de middelpunten op een gegeven cirkel liggen en die door een vast punt op deze cirkel gaan. De omhullende van deze familie cirkels zal een cardioïde zijn.

Om deze cirkels te tekenen kiezen we als vast punt $(x, y) = (1, 0)$ en zorgen we ervoor dat de middelpunten op de eenheidscirkel liggen, d.w.z. $(\cos t, \sin t)$. De straal van elk van de cirkels is dus

$$r = \sqrt{(\cos(t) - x)^2 + (\sin(t) - y)^2}$$

Het programma om deze familie cirkels te tekenen ziet er als volgt uit.





4.3 Oppervlakte tussen twee krommen

In de analyse is het in vele richtingen aangewezen om leerlingen de oppervlakte tussen twee krommen $y = f(x)$ en $y = g(x)$ te laten uitrekenen aan de hand van integralen. Hierbij is een goede tekening zeker van grote waarde. Ook hier kan de **TI-84 Plus Color** helpen.

Onderstaande stelling geeft aan hoe het bepalen van de gezochte oppervlakte gebeurt aan de hand van een integraal.

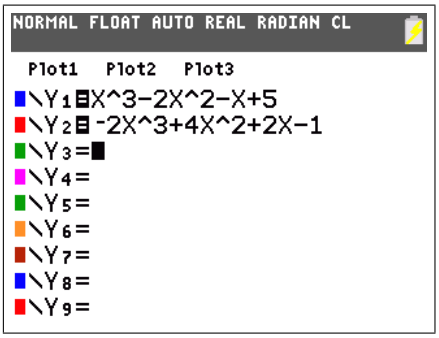
Stelling. *Zij f en g continue functies, zodat op het interval $[a, b]$: $f \geq g$. Dan is de oppervlakte tussen beide krommen op dat interval gelijk aan*

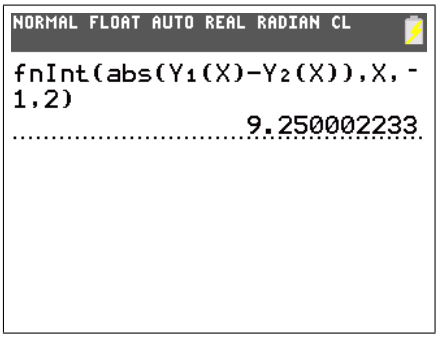
$$\int_a^b (f(x) - g(x))\mathrm{d}x$$

In het algemeen echter is het vaak zo dat f en g afwisselend de rol van grootste functie hebben, in zulke gevallen moet de oppervlakte tussen beiden berekend worden door op te splitsen in verschillende integralen. We zullen nu echter de ingebouwde numerieke integratie van de **TI-84 Plus Color** gebruiken om een nauwkeurige benadering te maken voor de gezochte oppervlakte. Dit kan door volgende integraal uit te rekenen:

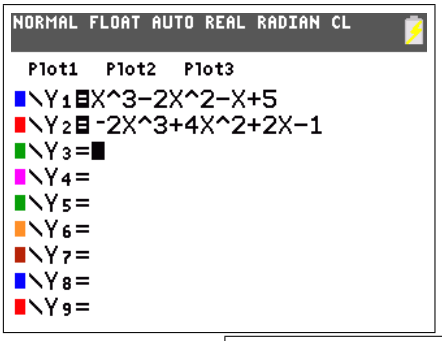
$$\int_a^b |f(x) - g(x)|\mathrm{d}x$$

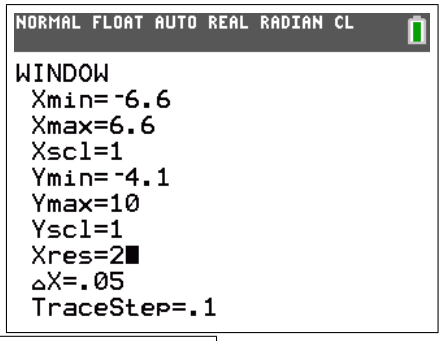
Gezien de absolute waarde herleidt een algebraïsche berekening zich tot het opsplitsen van de integraal en winnen we niets. De **TI-84 Plus Color** gebruikt echter numerieke methodes en moet deze integraal dus niet opsplitsen. We beschouwen de krommen $y_1 = x^3 - 2x^2 - x + 5$ en $y_2 = -2x^3 + 4x^2 + 2x - 1$. De ingesloten oppervlakte wordt begrensd door de snijpunten waarvoor $x = -1$ en $x = 2$. Je vindt de functie `[fnInt]` terug via `math`.

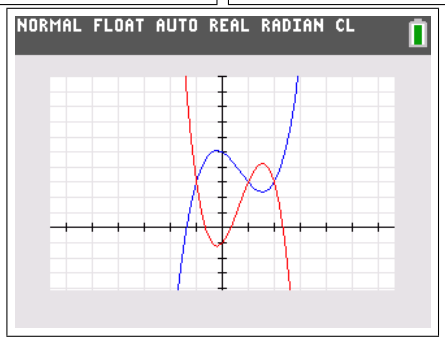




Het resultaat dat de **TI-84 Plus Color** geeft (na enige rekentijd) zit behoorlijk dicht bij de werkelijke waarde $\frac{37}{4} = 9.25$ die een leerling kan vinden door zelf de gepaste integralen uit te rekenen. Natuurlijk biedt de **TI-84 Plus Color** ook onmiddellijk de grafiek aan (na aanpassing van het grafisch venster - let op de parameter `Xres` om de grafiek sneller te laten tekenen).





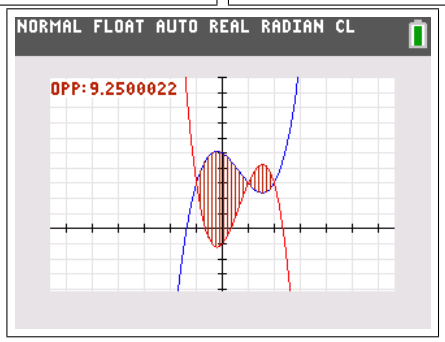


We kunnen nu bovenstaande stukken code samenzetten tot één enkel programma. We profiteren er tevens van om de oppervlakte te arceren met een twintigtal lijntjes.

```
NORMAL FLOAT AUTO REAL RADIAN CL
PROGRAM:INTFG
:ClrDraw
:2→Xres
:Disp "FTIES Y1,Y2"
:Disp "GRENZEN"
:Prompt A,B
:For(K,A,B,(B-A)/20)
:Line(K,Y1(K),K,Y2(K),BROW
N,1)
:End
```

```
NORMAL FLOAT AUTO REAL RADIAN CL
PROGRAM:INTFG
:fnInt(abs(Y1-Y2),X,A,B)→S

:TextColor(BROWN)
:Text(1,1,"OPP: ",S)
:
:
:
:
:
```



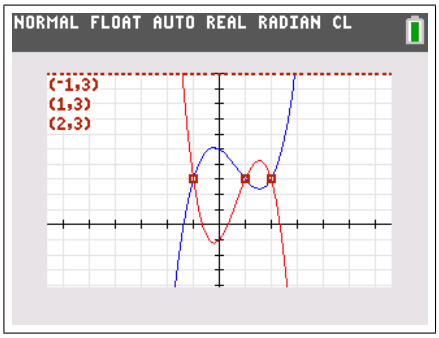
Het is een idee om eventueel dit programma te nemen als vertrekpunt van een OC wiskunde. Een leerling kan de code uitbreiden door bijvoorbeeld zelf de numerieke integratie te programmeren. Ook een programma dat zelf op zoek gaat naar de snijpunten is nuttig. Onderstaand programma doet exact dit en gebruikt hiervoor de dichotomie- of bisectiemethode.

```
NORMAL FLOAT AUTO REAL RADIAN CL
PROGRAM:INTERSEC
:ClrDraw
:2→Xres
:Y1(Xmin)-Y2(Xmin)→P
:(Xmax-Xmin)/50→S
:0→J
:For(K,Xmin,Xmax,S)
:Pt-On(K,Ymax,BROWN)
:If P*(Y1(K)-Y2(K))≤0
:Then
```

```
NORMAL FLOAT AUTO REAL RADIAN CL
PROGRAM:INTERSEC
:K-S→A
:K→B
:For(I,1,15)
:(A+B)/2→C
:If (Y1(A)-Y2(A))*(Y1(C)-Y
2(C))≤0
:Then
:C→B
:Else
```

```
NORMAL FLOAT AUTO REAL RADIAN CL
PROGRAM:INTERSEC
:C→A
:End
:End
:TextColor(BROWN)
:Text(J*15+1,2,"(",round(C
,3),",",round(Y1(C),3),")"
)
:Pt-On(C,Y1(C),BROWN,2)
:J+1→J
```

```
NORMAL FLOAT AUTO REAL RADIAN CL
PROGRAM:INTERSEC
:End
:Y1(K)-Y2(K)→P
:End
:
:
:
:
:
:
```

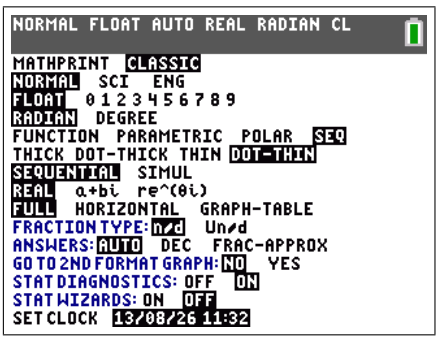


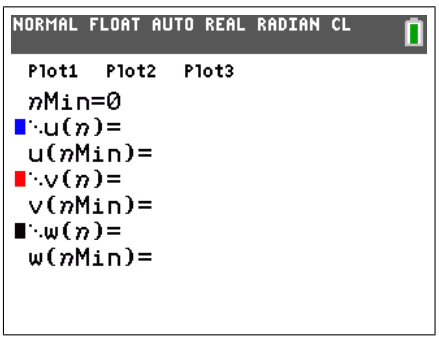
Het uitzoeken hoe het programma werkt, welke stellingen er worden gebruikt en in welke gevallen de snijpunten gevonden zullen worden, is een goede taak op zich. Voor meer informatie en programma's omtrent numerieke methoden verwijzen we naar Cahier 14: 'Voorbeelden met de TI-84+ uit de analyse' uit deze reeks.

4.4 Model van Verhulst en chaos

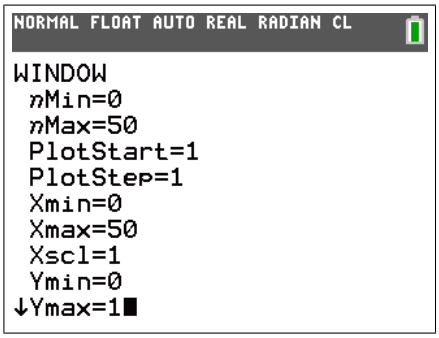
De recursieve rij $u_n = au_{n-1}(1 - u_{n-1})$ is in de biologie bekend als "het model van Verhulst". Voor verschillende waarden van $0 < a < 4$ verandert het convergentiegedrag volledig. Voor meer info refereren we naar gepaste literatuur [1].

Het grafisch voorstellen van rijen gaat met de **TI-84 Plus Color** als volgt. Via `mode` selecteer je `[seq]` en `[dot]`. Als je nu op `y=` drukt kun je rijen invoeren. De `[nmin]` optie laat je toe om rijen te indexeren vanaf 0 of een andere positieve waarde. Je kan rijen geven door de algemene term of door recursie. Merk op dat n verkregen wordt door `X,T,θ,n` en u, v, w door bijvoorbeeld `2nd` `[u]` (`2nd` `7`). Voor je de grafiek maakt, doe je er goed aan van met `window` de gewenste grenzen correct in te stellen.

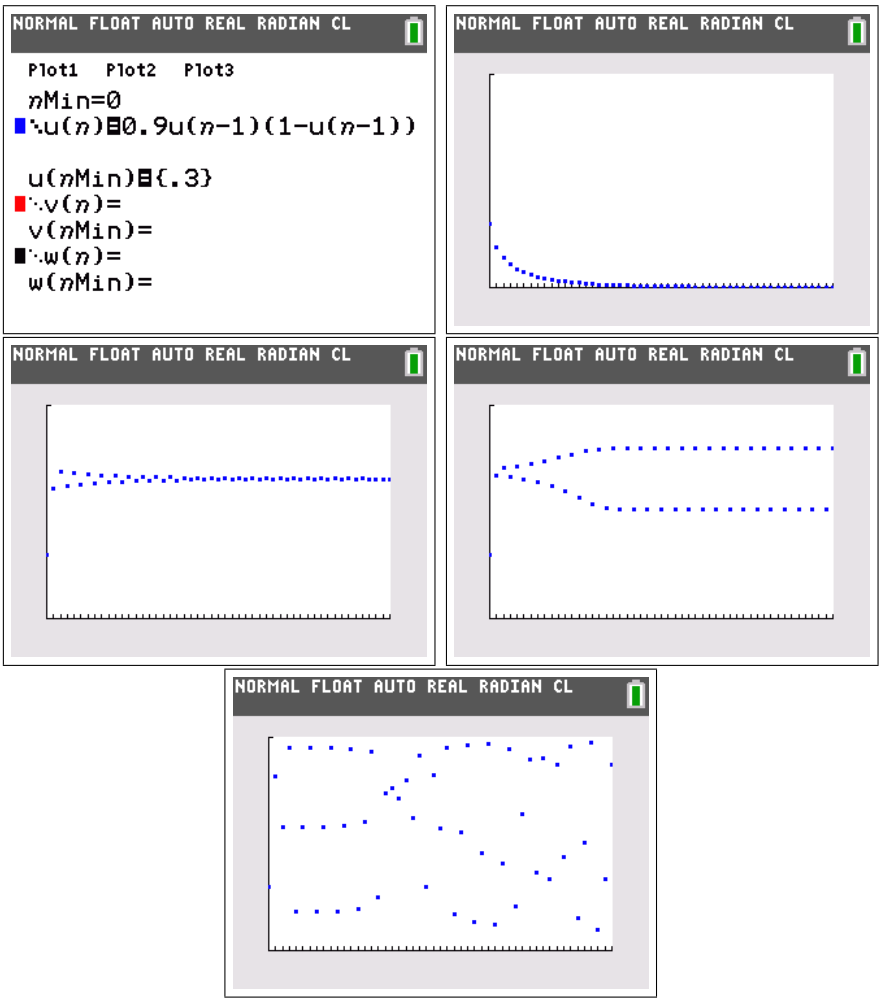




[1] H. A. Lauwerier, *Chaos met de Computer*, epsilon-uitgaven, 1996.



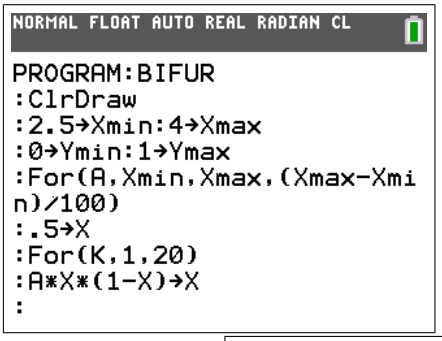
We gebruiken de **TI-84 Plus Color** om voor verschillende waarden van a de grafiek van de rij te maken.

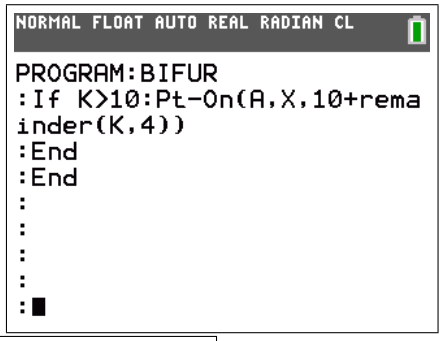


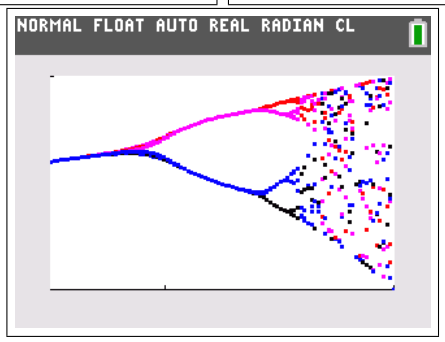
We zien duidelijk dat het convergentiegedrag heel verschillend kan zijn. Eerst convergentie naar 0, daarna naar een constante waarde. Voor grotere a bekomen we adherentie aan twee (of meerdere) waarden, dit gebeurt na een

zekere stabilisatiefase. Wanneer a de grens van 4 nadert, wordt het gedrag chaotisch.

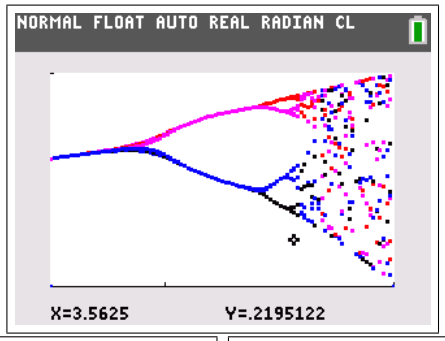
Om dit convergentiegedrag in kaart te brengen kunnen we een bifurcatiediagram maken. We plaatsen op de x-as de parameter a. Voor elke waarde van a tekenen we de punten van de rij, na een stabilisatiefase van 10 termen. Indien de rij adhereert aan twee waarden, zullen dus twee punten getekend worden boven de waarde van a. We kiezen ervoor om de opeenvolgende punten van een rij telkens een andere kleur te geven. Om de vertakkingen duidelijk te kunnen zien beperken we ons tot vier verschillende kleuren.

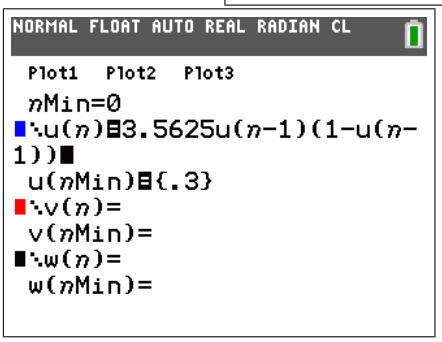


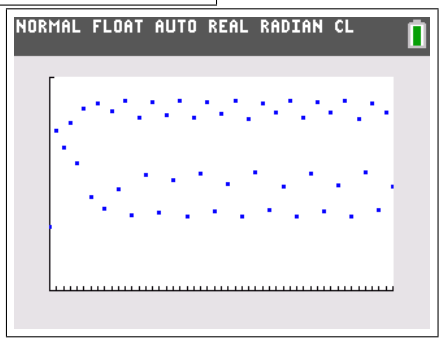




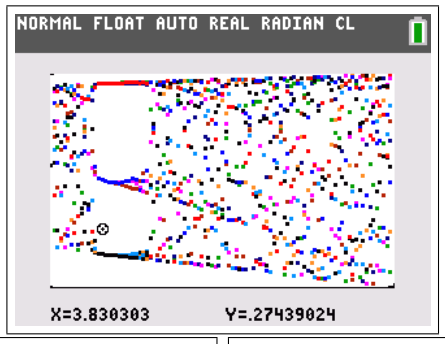
We zien duidelijk het chaotisch gedrag verschijnen. De eerste bifurcatie heeft plaats rond $a = 2,8$. De rij zal dan niet meer convergeren maar wel adhereren aan twee waarden. Voor een waarde rond $3,4$ (na de volgende bifurcatie) zal de rij adhereren aan vier waarden. Elke tak splitst opnieuw tot het systeem volledig chaotisch wordt. Deze bifurcaties gebeuren niet gelijktijdig, je kan waarden voor a vinden waarvoor er adherentie is aan 3 of 7 waarden.

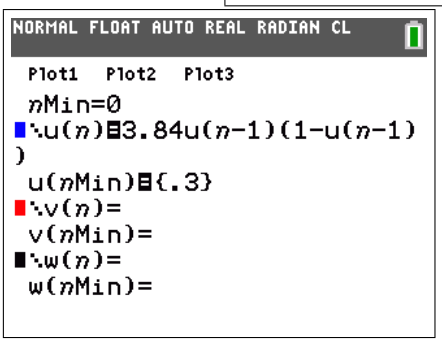


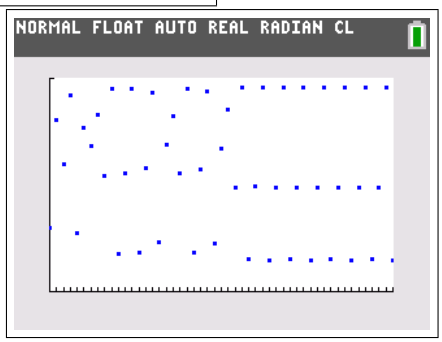




Midden in het chaotisch gedeelte is er plots weer structuur. We zien dit duidelijk als we $Xmin = 3.8$ stellen.







4.5 Kaarten tellen

4.5.1 Blackjack of 21

Het belang van (on)afhankelijkheid in de kansrekening blijkt duidelijk bij het kaartentellen in casino's. Je kan niet voor elk kaartspel kaarten tellen. Bij *Blackjack* of 21 lukt het wel.

In een casino wordt 21 als volgt gespeeld. Zes pakjes kaarten worden samengelegd (de *shoe*). De speler zet een som geld in en zowel het casino als een speler krijgt twee kaarten. Daarna mag elk om beurt een kaart bijvragen uit de stapel. Doel van het spel is van zo dicht mogelijk bij 21 te geraken, maar er nooit over. De speler kan stoppen wanneer hij wil, het casino moet een *stiff hand* hebben (meer dan 16). Dit laatste zorgt voor een tactisch voordeel die de speler kan benutten.

Om dit in te zien veronderstellen we dat het casino 14 in hand heeft. Het casino is dus verplicht verder te spelen.

- Grote kaarten ($10 \to A$, beeldjes tellen voor 10) zijn in het nadeel van het casino ($14 + 10 > 21$) .
- Kleine kaarten ($2 \to 6$) bieden dan weer een voordeel voor het casino ($14 + 2 < 21$).

Er worden zes pakjes kaarten gebruikt omdat er anders teveel tijd verloren gaat met het herschudden van de kaarten. Casino's willen zoveel mogelijk partijtjes kunnen spelen op korte tijd. Om dezelfde reden worden na een spelletje de gebruikte kaarten gewoon opzij gelegd. Ze spelen bij een volgend spel niet meer mee. Dit maakt dat de kansen **afhankelijk** zijn van de kaarten die gevallen zijn bij de vorige spellen. Deze afhankelijkheid maakt het tellen van de kaarten mogelijk.

4.5.2 Hi-Lo count

Behalve enkele autistische gevallen die overdreven zijn (denk aan de film *Rain man*), kan niemand alle kaarten van zes pakjes tegelijk bijhouden. Het kaarten tellen verloopt op een andere wijze. De *count* staat aanvankelijk op nul. Wanneer de kaarten vallen wordt de *count* als volgt aangepast:

- $2 \to 6 : +1$
- $7 \to 9 : +0$
- $10 \to A : -1$

Er moet dus slechts één getal bijgehouden worden!

- Wanneer *count* > 0 zijn er veel $2 \rightarrow 6$ gevallen, dit wil zeggen dat er nog veel $10 \rightarrow A$ over zijn, dit is een nadeel voor het casino en dus een voordeel voor de speler. Deze kan nu grotere sommen inzetten.

- Wanneer *count* < 0 ligt het voordeel bij het casino en kan de speler dus best minder inzetten.

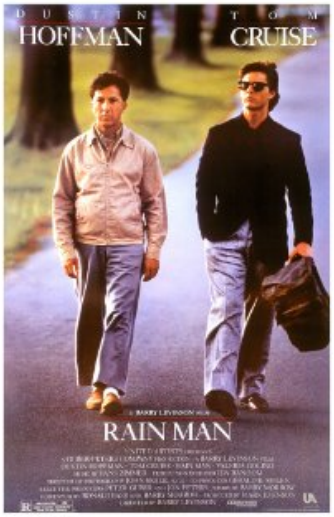
In het beste geval levert deze methode een voordeel van $2,5\%$ voor de speler. Natuurlijk laten casino's dit niet zomaar toe.

- Door gebruik te maken van elektronische jetons en zelf een *count* bij te houden kunnen ze statistisch nagaan of de speler de *count* benut om meer of minder in te zetten. In dit geval wordt de speler buitengezet.

- Regelmatig worden alle kaarten terug samengebracht en geschud. Dit vernietigt de *count*.

- Tegenwoordig gebruiken casino's ook automatische mengmachines die vliegensvlug de kaarten kunnen schudden voor elk spel, dan heeft een *count* geen nut meer.

Soms wordt de *Wong-methode* gebruikt. Aan verschillende tafels staan toeschouwers die een *count* bijhouden. Wanneer die goed genoeg is, wordt een speler gesignaleerd die dan het voordeel benut. Aldus wordt het verlies geminimaliseerd. Een aanrader is de film *21*.



4.5.3 Blackjack op de TI-84 Plus Color

Onderstaand programma simuleert het kaartspel Blackjack.

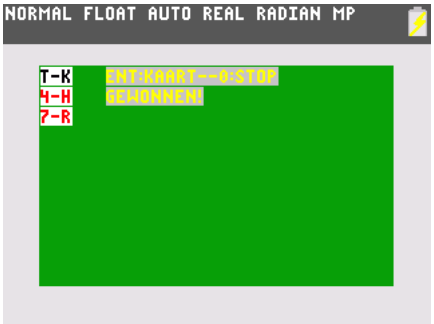
```
BackgroundOn GREEN
GridOff:AxesOff:FnOff
seq(1+I/4,I,0,51)->L6

Lbl 0
rand(52)->L5
SortA(L5,L6):52->K

Lbl 1
0->S:0->P
ClrDraw
TextColor(YELLOW)
Text(0,50,"ENT:KAART--0:STOP")

Lbl 2
getKey->G
If G=0:Goto 2
TextColor(BLACK)
If fPart(L6(K))<.5:TextColor(RED)
Text(P*15,1,sub("123456789TBDK",iPart(L6(K)),1)+"-"+sub("HRKS",4fPart(L6(K))+1,1))
S+min(iPart(L6(K)),10)->S
TextColor(YELLOW)
If G=102
Then
  If S<21:Text(15,50,"SPIJTIG")
  If S=21:Text(15,50,"SPIJTIG")
  If S>21:Text(15,50,"OP TIJD GESTOPT")
Else
  K-1->K:P+1->P
  If S<21:Goto 2
  If S=21:Text(15,50,"GEWONNEN!")
  If S>21:Text(15,50,"VERLOREN!")
End

Lbl 3
getKey->G
If G=0:Goto 3
If K>10:Goto 1
Text(30,50,"MENGEN...")
Goto 0
```



Het spelletje werd doelbewust eenvoudig gehouden, zodat de programmeertechnieken duidelijk zichtbaar zijn. We overlopen het programma.

1-2 Het grafisch scherm wordt ingesteld. We gebruiken groene achtergrond om een casino-look te bekomen.

3 Een pakje kaarten wordt in lijst L_6 geplaatst. Elke kaart heeft een waarde $(1, \ldots, 13)$ en een soort (Harten, Ruiten, Klaveren, Schoppen). Om deze beide elementen in één enkel getal te plaatsen gebruiken we volgende truc. De soorten komen overeen met de getallen na de komma $(0.0, 0.25, 0.5, 0.75)$. Zo komt het laatste getal (13.75) overeen met $K\spadesuit$.

5-7 Nu worden de kaarten gemengd. Dit gebeurt door een tweede lijst van willekeurige getallen te plaatsen in L_5. Beide lijsten worden dan geordend afhankelijk van de waarden in L_5, de kaarten in L_6 staan daarna in een willekeurige volgorde. De variabele K is de bovenste kaart, voorlopig de 52ste.

9-13 Het spel begint hier. De som S wordt op 0 gezet en het aantal kaarten P van de speler ook. Het grafisch scherm wordt leeggemaakt en de instructies verschijnen.

15-17 Deze drie regels zijn een trucje om te wachten tot de gebruiker een toets heeft ingedrukt.

18-20 Eenmaal een toets ingedrukt, wordt de kleur op zwart gezet. Indien echter het deel na de komma (`fPart`) van de bovenste kaart (K) kleiner is dan 0.5 gaat het om een rode kaart en wordt de tekstkleur aangepast. Hierna wordt met `iPart` de waarde van de kaart bepaald. Deze gebruikt men dan om het juiste symbool uit `123456789TBDK` te selecteren (we schrijven hier `T` voor 10 zodat elke kaart juist één symbool nodig heeft). Op analoge wijze wordt `fPart` gebruikt om de soort voor te stellen `HRKS` (Harten, Ruiten, Klaveren, Schoppen). De speler heeft P kaarten en

elke lijn tekst is 15 pixels hoog, dus wordt de kaart weggeschreven op hoogte $P * 15$ van het scherm.

21 De nieuwe som wordt bepaald. Door gebruik te maken van `min(iPart(L6(K)),10)` tellen alle beeldjes voor 10, de azen tellen in deze versie voor 1.

23-27 Indien de speler op de toets 0 heeft gedrukt (toetscode 102) wil hij stoppen. Naargelang de som S wordt een boodschap afgedrukt die aangeeft of dit de juiste beslissing was.

28-33 Indien de speler niet wenst te stoppen, krijgt hij de kaart. Er is nu één kaart minder op de stapel $(K - 1)$ en de speler heeft een kaart meer $(P + 1)$. Indien de som minder dan 21 bedraagt wordt er verder gespeeld. Indien niet, heeft de speler gewonnen of verloren en krijgt hij de passende boodschap.

35-37 Opnieuw een wachtlus.

38-40 Indien er nog meer dan tien kaarten over zijn, wordt een volgend spel begonnen, anders worden de kaarten opnieuw gemengd.

Deze versie van Blackjack is verschillend van de versie in casino's, toch is een *Hi-Lo count* mogelijk. Als je speelt, wil je weten of je moet stoppen of niet. Indien er nog veel kleine kaarten zijn, loont het risico om verder te gaan. De *count* kan je dus als volgt beschrijven.

- $1 \rightarrow 4 : -1$
- $5 \rightarrow 9 : +0$
- $10 \rightarrow K : +1$

Indien de *count* positief is, zijn er al vele grote kaarten gevallen en blijven er dus kleine kaarten over. Het is dan in je voordeel van niet te snel te stoppen. Natuurlijk garandeert dit geen winst! Let wel dat je de *count* terug op nul zet als de kaarten worden gemengd.

Hoofdstuk 5

Appendix: Eventjes leren programmeren

Je kan de capaciteiten van de **TI-84 Plus Color** uitbreiden door zelf een programma toe te voegen. In tegenstelling tot hetgeen vaak wordt gezegd is programmeren voor de **TI-84 Plus Color** niet echt moeilijk. Als leerlingen zelf eens een programma behandelen tijdens de lessen, krijgen ze inzicht in hoe een grafisch rekenmachine werkt. Bovendien is tegenwoordig de belangrijkste toepassing van de wiskunde de informatica, in ruime zin. Van grafische rekenmachine tot statistische computerprogramma's, via gsm's, mp3-spelers, gameboy's en play stations, computer games en internettoepassingen, in bijna elke moderne technologie zit meer dan 50% wiskunde. Zonder wiskunde geen moderne technologie!

De programmeertaal die in de **TI-84 Plus Color** zit, is TIBasic. Dit is een dialect van BASIC, een programmeertaal waarmee Bill Gates (Microsoft) zijn faam heeft verworven. BASIC (Beginners All-purpose Symbolic Instruction Code) werd ontworpen om ook de leek in staat te stellen om kleine programma's te schrijven. Het TIBasic-dialect is trouw aan deze filosofie: er is geen enkele programmeer-ervaring nodig om programma's te schrijven voor de **TI-84 Plus Color** . Enkel een beetje doorzettingsvermogen en zelfvertrouwen is nodig.

Om een programma te schrijven ga je naar `prgm`. Je kunt hier kiezen om een programma uit te voeren (`[exec]`), te veranderen (`[edit]`) of om een nieuw programma te schrijven (`[new]`). Indien je de laatste keuze maakt wordt er naar een naam gevraagd. Nadien kom je op de editor uit, waar je jouw programma kan invoeren. De commando's die met het programmeren te maken hebben, zitten nu onder `prgm`. We geven hier een kort overzicht van de nuttigste programmeerfuncties.

[ctl]	
[if]	Eerste vorm:
	`:if` *voorwaarde* `:` *commando*
[then]	Tweede vorm: uitgebreide `if` structuur:
[else]	`:if` *voorwaarde*
	`:then`
	`:` *commando's*
	`:else`
	`:` *commando's*
	`:end`
[for]	om lussen te maken
	`:for(` *var, beginwaarde, eindwaarde [, stapgrootte]* `)`
	`:` *commando's*
	`:end`
[end]	om bovenstaande blokken te eindigen

[i/o]	
[disp]	om een waarde/string op het scherm te printen
[prompt]	om de waarde van een variabele te vragen aan de gebruiker
[input]	om een text op het scherm te tonen en een waarde te vragen aan de gebruiker, deze kan ook een functie zijn.
	`:input` *"text", variabele* (bijv. y_1)

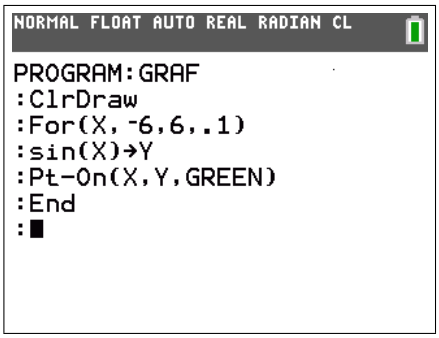
Nu we de nodige commando's kennen, kunnen we een zeer eenvoudig voorbeeld behandelen. Dit voorbeeld geeft inzicht over hoe de **TI-84 Plus** grafieken maakt.

Opdracht 21. Gebruik een `for`-lus om een programma te schrijven dat de grafiek van een functie (bijvoorbeeld de sinusfunctie) maakt d.m.v. het volgende algoritme.

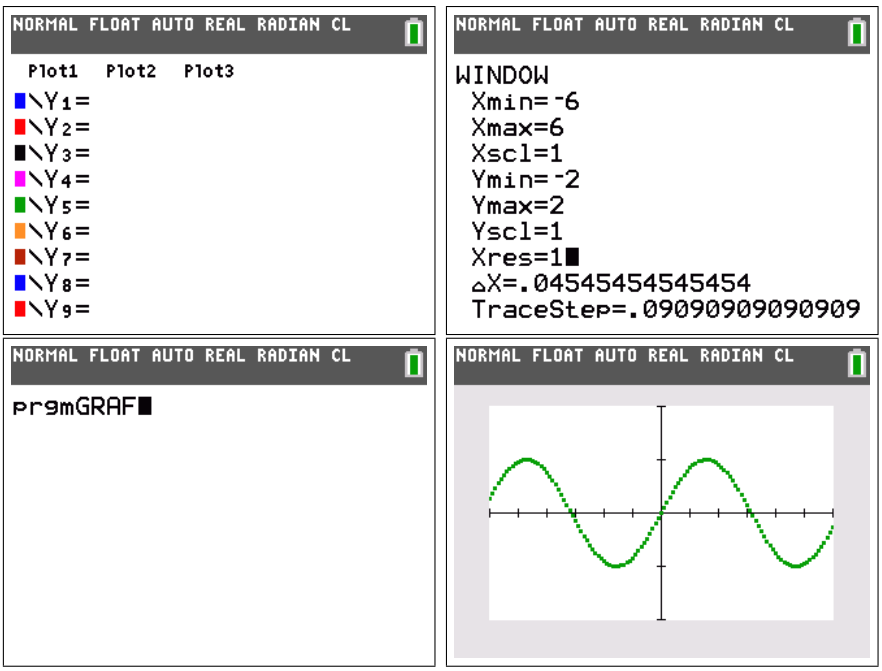
```
voor x gaande van -6 tot 6 met een stap van 0.1
bereken y=sin(x)
teken het punt (x,y)
sluit de lus
```

Gebruik de grafische commando's `2nd` `[draw]` `[clrdraw]` (om een leeg scherm te krijgen) en `2nd` `[draw]` `[point]` `[pt-on]` om een punt te tekenen. Vergeet niet van via `y=` alle functies weg te halen en om met `window` de grenzen aan te passen!

Oplossing. Het programma telt exact vijf regels:



Het resultaat is hetzelfde als hetgeen we zouden verkrijgen door de ingebouwde functies te gebruiken.



We bekijken de nieuwigheden van de TI-84 Plus Color.
Naast enkele veranderingen in de menu's en opties wordt vooral aandacht besteed aan het gebruik van het nieuwe kleurenscherm. Enkele klassieke oefeningen worden beschouwd, alsook een voorbeeld uit de optica. Hiermee maken we enkele eenvoudige programma's. Deze zullen voluit het nieuwe kleurenscherm gebruiken.
We leggen een link naar de kunstwereld, hetgeen tenslotte leidt naar een knutseloefening voor leerlingen.

DIDIER DESES is leerkracht wiskunde aan het Koninklijk Atheneum Koekelberg en geeft les aan de Wetenschappelijke (5u wisk/week) en de Latijnse richtingen (3u wisk/week). Hij is tevens wetenschappelijk medewerker aan de Vrije Universiteit Brussel.

December 2013

© 2013 Dit cahier is bedoeld als lesmateriaal, mag hiervoor vrij gekopieerd worden en kan gedownload worden via de website **www.t3vlaanderen.be**