

# Korte programma's of inkijkprogramma's

B. van Etten

Moller Instituut, Tilburg

## Samenvatting

*Inkijkprogramma's kunnen een nieuw didactisch hulpmiddel zijn: het zijn inzichtelijke programma's waarvan het lezen en veranderen het leren van wiskunde moet ondersteunen.*

Binnen het kader van het NIVO-project bestaat een cursus over het gebruik van de computer in wiskundelessen. 'Korte programma's' is een onderwerp van deze cursus. Korte programma's kunnen voor de wiskundeleraar een nieuw didactisch hulpmiddel zijn. Een leraar schrijft voor zijn les van morgen een programma. Hijzelf of zijn leerlingen zullen de gebruikers zijn. Dat stelt minder zware eisen dan bij een professioneel programma. Er hoeft minder gelet te worden op invoerbeveiliging, scherm lay-out en dergelijke.

Programma's kunnen in de klas gebruikt worden zonder dat de leerling de programmatekst ziet. Maar in de bovenbouw kunnen programma's een meerwaarde aan het onderwijs geven wanneer de leerling wel in de programmatekst moet lezen. Dit stelt extra eisen aan de programmatekst. Deze moet heel doorzichtig zijn voor de leerling, die niet alles hoeft te begrijpen, maar er wel de weg in moet kunnen vinden. Het deel van de programmatekst waar de wiskunde aan de orde komt zal voor de leerling begrijpelijk moeten zijn. Van die leerling mag daarbij niet gevraagd worden om de gebruikte programmeertaal machtig te zijn.

Daarom is het nuttig om 'korte programma's' te onderscheiden van 'inkijkprogramma's'. Voor deze laatste is korthed niet het eerste vereiste, hoewel de meeste inkijkprogramma's wel kort zullen zijn. Een belangrijker eis voor inkijkprogramma's is de inzichtelijkheid van de programmatekst. Het programma lezen en veranderen moet het leren van wiskunde ondersteunen.

Dit stelt naar mijn mening de volgende eisen aan inkijkprogramma's:

- Het programma heeft een duidelijke structuur.

- Variabelen hebben goed gekozen namen.
- Het deel van het programma dat wiskundig van belang is moet zeer goed leesbaar zijn voor de leerling.
- Basis opdrachten die de leerling begrijpend moet kunnen lezen zijn beperkt tot:  
REM, LET, DEF FN,  
FOR ... NEXT ...,  
WHILE ... WEND,  
IF ... THEN ... ELSE ...,  
GOSUB,  
INPUT en PRINT  
Hij hoeft zelf geen taalconstructie te kunnen maken. Voor manipulaties met de programmatekst moeten leerlingen met de editor kunnen omgaan en met de commando's LIST en RUN.

De leraar die inkijkprogramma's schrijft moet zelf gedisciplineerd programmeren. De programmeertaal Basic ondersteunt deze discipline niet. Om tot inzichtelijke programma's te komen moet Basic gestructureerd worden gehanteerd. Ik wil dit duidelijk maken met een voorbeeld.

In de NIVO-wiskundecursus wordt bij het onderdeel 'korte programma's' gewerkt met het programma GRAFIEK.BAS.

```
1 CLS
5 INPUT "welke stapgrootte ":H
20 SCREEN 2:WINDOW (-10,-2) - (10,2)
30 LINE (-10,0) - (10,0)
40 LINE (0,-10) - (0,10)
100 DEF FNF(X)=SIN(X)
110 FOR X = -10 TO 10 STEP H
120 X1=X:X2=X+H
130 LINE(X1,FNF(X1)) (X2,FNF(X2))
140 NEXT X
```

Leerlingen brengen veranderingen aan in functievoorschriften, domein en bereik. De feedback die de computer levert moet hun kennis over functies vergroten. In dit programma moet hij daarvoor Basic-opdrachten zoals: SCREEN, WINDOW, LINE kennen. Dat kan het leerproces waarin wiskunde centraal staat belemmeren. Een mijns inziens betere tekst voor een inkiijkprogramma is:

```

10 REM tekenen van grafiek van een functie
20 GOSUB 200: REM lees functie's domein en bereik
30 GOSUB 300: REM maak tekenschermbereik
40 FOR X = XLINKS TO XRECHTS STEP STAP
50 GOSUB 700: REM teken volgend lijnstuk
70 NEXT X
80 GOSUB 800: REM sluit af
90 END

200 REM lees functievoorschrift, domein en bereik
210 DEF FNF(X) = SIN(X)
220 LET XLINKS = -3: LET XRECHTS = 3
230 LET YONDER = -1: LET YBOVEN = 1
290 RETURN

300 REM maak tekenschermbereik
310 CLS: KEY OFF
320 SCREEN 2: REM kiezen voor tekenschermbereik
330 VIEW (5,5) - (620,160)
340 WINDOW (XLINKS, YONDER) - (XRECHTS, YBOVEN)
350 LINE (XLINKS, 0) - (XRECHTS, 0): REM tekenen X-as
360 LINE (0, YONDER) - (0, YBOVEN): REM tekenen Y-as
370 LET STAP = 0.1
390 RETURN

700 REM teken volgend lijnstuk van de grafiek
710 LET X1 = X: LET X2 = X + STAP
720 LINE (X1, FNF(X1)) - (X2, FNF(X2))
790 RETURN

800 REM sluit af
810 LOCATE 24, 1: PRINT "druk op de spatiebalk. ";
820 LET TOETS$ = " "
830 WHILE TOETS$ <> " ": LET TOETS$ = INKEY$: WEND
840 KEY ON: SCREEN 0: REM kiezen voor tekenschermbereik
890 RETURN

```

Deze tekst is een stuk langer dan het eerste programma. Beide programma's doen op het scherm precies hetzelfde. Voor de leerling is in dit programma alleen belangrijk de subroutine 'lees functievoorschrift, domein en bereik' (regel 200-290). Dat hij daar moet zijn leest hij in het hoofdprogramma in regel 20.

Hoe het programma verder werkt is voor de leerling-gebruiker niet van belang. Hij kan zich op de wiskunde concentreren.

Met kleine variaties kan de leraar dergelijke programma's aanpassen. Bij de volgende veranderingen in subroutines 200 en 700 wordt van een functie ook zijn afgeleide getekend. De leerling kan gevraagd worden om het functievoorschrift van de afgeleide in te voeren in regel 240. Het programma geeft dan feedback op zijn leren door de ingevoerde afgeleide te vergelijken met de berekende.

```

200 REM lees functievoorschriften, domein en bereik
210 LET STAP = .1
220 DEF FNF(X) = SIN(2*X)
230 DEF FNH(X) = (FNF(X+STAP) - FNF(X))/STAP
240 DEF FNFACC(X) = 0
250 LET XLINKS = -3: LET XRECHTS = 3
260 LET YONDER = -2: LET YBOVEN = 2
290 RETURN

```

```

700 REM teken volgend lijnstuk van de grafiek
710 LET X1 = X: LET X2 = X + STAP
720 LINE (X1, FNF(X1)) - (X2, FNF(X2))
730 LINE (X1, FNH(X1)) - (X2, FNH(X2))
740 LINE (X1, FNFACC(X1)) - (X2, FNFACC(X2))
790 RETURN

```

Het volgende inkiijkprogramma heet 'bisectie'. Het bepaalt een nulpunt van een functie volgens de halveringsmethode; daarbij is het van belang een geschikt interval  $[A, B]$  te kiezen ( $f(A) * f(B) > 0$  en  $f$  monotoon). In een wiskundeles waar dit onderwerp aan de orde is, moet de leerling leren om geschikte intervallen te kiezen. Het programma beveiligd daarom de intervalinvoer niet.

In dit programma moet de leerling in regel 210 zijn functievoorschrift schrijven. De subroutine 'doe volgende halveringslag' (regel 400-490) is het eigenlijke rekenproces.

```

10 REM bisectie
20 GOSUB 200: REM kies functie en interval
30 GOSUB 300: REM maak scherm in orde
40 WHILE B - A > TOLERANTIE
50 GOSUB 400: REM doe volgende halveringslag
60 WEND
70 GOSUB 600: REM druk resultaat af
90 END

200 REM kies functie en interval
210 DEF FNF(X) = X^3 + 2*X^2 + 10*X - 20
220 INPUT "Linkergrens interval: ", A
230 INPUT "Rechtergrens interval: ", B
240 LET TOLERANTIE = .00001
290 RETURN

300 REM maak scherm in orde
310 CLS
320 PRINT "ondergrens", "bovengrens", "intervallengte"
330 PRINT
390 RETURN

400 REM doe volgende halveringslag
410 LET X = (B + A)/2
420 IF FNF(A)*FNF(X)<0 THEN B = X ELSE A = X
430 PRINT A, B, B - A
490 RETURN

600 REM druk resultaat af
610 PRINT: PRINT "nulpunt is: "; (A + B)/2: PRINT
690 RETURN

```

Een simpel programma is het inkiijkprogramma 'berekenen van een functietabel'. Het is vaak bruikbaar wanneer een leerling een grafiek moet tekenen in zijn schrift. Ook het 'zoeken' naar een nulpunt is met dit programma mogelijk. De leerling hoeft alleen in de subroutine 'lees functie, xmidden en stapgrootte' (regel 200-290) met de programmatekst te manipuleren.

```

10 REM berekenen van functietabel
20 GOSUB 200: REM lees functie, xmidden en stapgrootte
30 GOSUB 300: REM maak startklaar
40 FOR I = -5 TO 5
50 GOSUB 400: REM bereken tabel en druk tabel af
60 NEXT I
90 END

200 REM lees functie, xmidden en stapgrootte
210 DEF FNF(X) = X * X - 1000
220 CLS: LOCATE 10
230 INPUT "Midden van het interval: ", XMIDDEN
240 LOCATE 15: INPUT "Stapgrootte: ", STAP
290 RETURN

```

```
300 REM maak startklaar
310 CLS: LOCATE 2
320 PRINT "Stapgrootte is ";STAP
330 LOCATE 10: PRINT "Midden: "
340 LOCATE 4
390 RETURN

400 REM bereken tabel en druk tabel af
410 LET X = XMIDDEN + I * STAP
420 LOCATE I+10,15: PRINT X
430 LOCATE I+10,45: PRINT FNF(X)
490 RETURN
```

Natuurlijk zijn er betere grafiekprogramma's, en betere numerieke methoden om nulpunten te bepalen.

Inkijkprogramma's zijn als een zakmes. Je kunt er van alles mee. Voor het echte grote werk kies je andere hulpmiddelen, maar soms ben je blij een zakmes tot je beschikking te hebben.