Van dobbelsteen tot lopende band

Automatische Gegevensverwerking in Wiskunde A

H. Verhage
OW & OC, RU Utrecht

Summary

One tenth of the time available for the HEWET new mathematics program is to be devoted to the teaching of Data Processing. The subject will be introduced by means of teaching algorithms and using a calculator to deal with them, this being followed by some elementary programming. Apart from this introduction, Data Processing will be integrated into other maths A subjects such as matrices, linear programming and statistics. The use of standard programs for these subjects will allow the treatment of matters that otherwise it would be impossible to deal with.
The introduction of Data Processing in the pilot schools and later on in all secondary schools will give rise to some practical problems owing to the fact that a great many different types of micro-computers are being used.

De invulling van het wiskunde A programma gaat steeds duidelijker vormen aannemen. Van de meeste te ontwerpen pakketjes heeft al tenminste één versie het licht gezien en met een groot gedeelte daarvan heeft u al kennis kunnen maken via bijv. De Nieuwe Wiskrant. Welke plaats neemt Automatische Gegevensverwerking nu in te midden van de veelheid aan wiskunde A onderwerpen? Weliswaar is in het HEWET-rapport de helft van de toelichting op het A-programma aan AGV besteed, maar dit neemt niet weg dat slechts 10% van de beschikbare uren voor AGV is ingeruimd.
Om AGV toch een volwaardige plaats in het A-programma te kunnen geven, is er voor gekozen AGV sterk te integreren met de verdere onderwerpen van het nieuwe programma. Dientengevolge loopt de leerstofontwikkeling van AGV min of meer in de pas met die van de rest van wiskunde A. Geleidelijk aan begint de puzzel echter compleet te worden.

Tot nu toe

De ervaringen tot nu toe op de twee experimenteerscholen hebben het nodige vallen en opstaan met zich meegebracht. Gedurende het eerste experimentele jaar hebben de leerlingen voor AGV alléén met diverse standaardprogramma's gewerkt. Nadeel hiervan was, dat de leerlingen het gevoel hadden voornamelijk knoppen in te moeten drukken, zonder te begrijpen wat er gebeurde. Dit bezwaar is het lopende cursusjaar enigszins ondervangen door te beginnen met een inleiding "kennismaken met algoritmen en programmeren". Om niet meteen al aan een computertaal vast te zitten, begon deze inleiding met algoritmen die op een gewoon rekenmachientje zijn uit te voeren. Een probleem dat zich hierbij voor deed, was om te komen tot een goede notatie voor deze algoritmen.
Gewapend met deze ervaringen wordt thans gewerkt aan een definitieve invulling van het AGV programma. Er wordt begonnen met een inleiding die bestaat uit:

- kennismaken met algoritmen en het uitvoeren ervan op de rekenmachine;
- "ruiken" aan programmeren.

Deze inleiding is bedoeld voor maximaal 8 lessen.
De rest van het programma bestaat uit:

- het werken met standaardprogramma's behorend bij onderwerpen uit Grafische Verwerking, Matrices, Lineair Programmeren;
- zelf programmaatjes maken bij onderwerpen uit Differentiëren, Kansverdelingen.

De diverse algoritmen en (standaard)programma's worden beschreven in structuurdiagrammen. Het voordeel hiervan is, dat deze structuurdiagrammen (althans voor eenvoudige programma's) dicht bij het computerprogramma liggen.
Een en ander zal geïllustreerd worden aan de hand van een viertal voorbeelden uit AGV.

Dobbelsteen

In het vierde klas pakketje Kansrekening hebben de leerlingen kennis gemaakt met het gebruik van toevalsgetallen bij simulaties. Daartoe zijn achter in het boek door de computer geproduceerde lijsten toevalsgetallen opgenomen. Een computer die iets toevalligs doet? Een uitstekende contradictio in terminis om duidelijk te maken dat alles wat een computer doet stap voor stap in een algoritme vastgelegd moet zijn.
Een algoritme dat pseudo-toevalsgetallen maakt moet in elk geval a) alle getallen ongeveer even vaak voor laten komen en b) er voor zorgen dat de rij getallen zich niet te snel herhaalt. Een dergelijk algoritme is gemakkelijk op het rekenmachientje uit te voeren. Met een kleine handgreep wordt deze algoritme zodanig gewijzigd dat deze bruikbaar is om het gooien met een dobbelsteen te simuleren.
Het structuurdiagram (voor verwerking op het rekenmachientje) hierbij is:

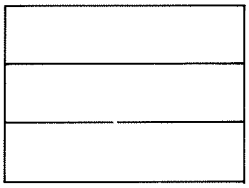
```
geheugen ← INVOER
→ 10 keer
    venster ← 23 * geheugen
    venster ← venster – INT (venster)
    geheugen ← venster
    venster → 6 * venster + 1
    UITVOER INT (venster)
STOP
```

Uitvoeren van de algoritme met als INVOER 0.439147 geeft:

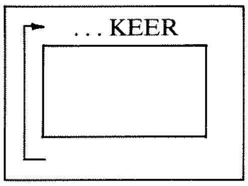
geheugen ← 0.439147
9.870381 ← 23 * 0.429147
0.870381 ← 9.870381 – INT (9.870381)
geheugen ← 0.870381
6.222286 ← 6 * 0.870381 + 1
6 opschrijven

Als bijvoorbeeld 6 leerlingen elk 10 worpen simuleren (natuurlijk met ieder een ander startgetal) geeft de simulatie al een heel aardig resultaat.
De functie van het structuurdiagram hierbij is duidelijk: het is een hulpmiddel bij het beknopt en ondubbelzinnig beschrijven van de algoritme. Bovendien maakt het voor de leerlingen duidelijk welke handelingen achtereenvolgens op de rekenmachine uitgevoerd moeten worden.
In dit eerste hoofdstuk worden de volgende "structuren" geïntroduceerd:

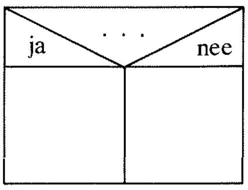
opeenvolging van stappen
in diagram aangegeven door:



herhaling
aangegeven door:



en de *keuze tussen ja en nee*
aangeduid met:



Tramlijn

De structuurdiagrammen die in het vorige hoofdstuk zijn opgesteld bij diverse algoritmen, worden in dit hoofdstuk omgezet in computerprogramma's en verwerkt op de computer. Hier zullen de wegen voor verschillende scholen uiteen gaan lopen: sommige scholen hebben (nu al of in de toekomst) de beschikking over een aantal microcomputers, andere zullen gebruik gaan maken van de mogelijkheid om schrapkaarten centraal op het Onderwijs Computercentrum te laten verwerken.
Daarnaast moet er een keus gemaakt worden voor een bepaalde computertaal. Als er met schrapkaarten gewerkt wordt, zal dit ECOL zijn. Bij micro's kan er, afhankelijk van de machine, gewerkt worden met ECOL, COMAL of BASIC. De laatste is het meest gangbaar, maar helaas tevens het minst wenselijk. Het zelf programmeren zoals dat bij AGV voorkomt is echter dusdanig summier, dat het er niet zoveel toe doet in welke taal dat gebeurt.
Dit hoofdstuk wordt afgesloten met de simulatie van een tramlijn. De leerlingen hoeven niet zelf een programma te maken, het is de bedoeling een gegeven programma uit te testen en de kwaliteit ervan te beoordelen. De opgave (in BASIC) luidt:

26 *Een tramlijn heeft 18 haltes, begin- en eindpunt meegerekend. Bij iedere halte, behalve bij het eindpunt, stappen er ten hoogste 20 mensen in. Vanaf de vierde halte stappen er hoogstens 22 mensen uit. Het programma beschrijft hiervan een simulatie.*

```
10 TAL =0
20 FOR HALTE=1 TO 18
30 IF HALTE<18 THEN IN=INT(21*RND(0))
   ELSE IN=0
40 TAL=TAL+IN
50 IF HALTE>=4 THEN UIT=INT(23*RND(0))
   ELSE UIT=0
60 TAL=TAL–UIT
70 PRINT IN, UIT, TAL
80 NEXT HALTE
90 END
```

Hier kan het maken van een structuurdiagram een hulpmiddel zijn bij het doorgronden van het programma.
Een greep uit het commentaar dat de leerlingen in Zevenaar op de resultaten van de computerverwerking hadden:

- *Er wordt een nieuw programma ingevoerd. (Dat wordt er samen ingetypt). Het programma beschrijft een simulatie van tramhaltes en het aantal mensen dat in- en uit de trams komt bij die haltes. Op dit moment is nog niet helemaal duidelijk wat bedoeld wordt met "TAL". Maar eerst wordt het programma verder ingetypt. Het is allemaal nog niet erg duidelijk. Bespreking van het structuurdiagram volgt. Er wordt nagegaan wat de verschillende termen betekenen. Na veel uitleg van de zijde van Astrid, begrijpt iedereen waar het over gaat. Er is nog een klein foutje in regel 40 gemaakt; dat wordt even weggewerkt. Nu volgt de uitvoer van het programma. We krijgen dan het volgende te zien:*

Aantal mensen dat instapt	*Aantal mensen dat uitstapt*	*Aantal mensen dat in de bus zit*
10	*13*	*12*
13	*21*	*4*
8	*21*	*–9*
3	*12*	*–18*

Commentaar:

- *Een gedeelte van de gegevens is niet te zien. (Niet erg overzichtelijk).*
- *Ook klopt het niet helemaal wat er op het scherm te zien is. Als er b.v. 12 mensen in de bus zitten en de bus rijdt verder en er stappen onderweg nog 13 mensen in en er stappen ook 21 mensen uit, dan betekent dat dat er mensen zijn die instappen en meteen weer uitstappen.*

- *Met het gegeven programma kan het voorkomen dat er na de eindhalte nog mensen in de bus zitten. Praktisch gezien komt dit niet voor.*
  Er kunnen negatieve aantallen mensen in de bus blijven zitten en ook dit komt niet voor.

- *Bij de laatste halte stappen 16 van de 46 mensen uit de tram. Er blijven dus 30 mensen in de tram zitten. Bij de tweede verwerking: er stappen bij de laatste halte 10 van de 6(!) mensen uit. Dat klopt niet.*
  Ons aangepast programma:

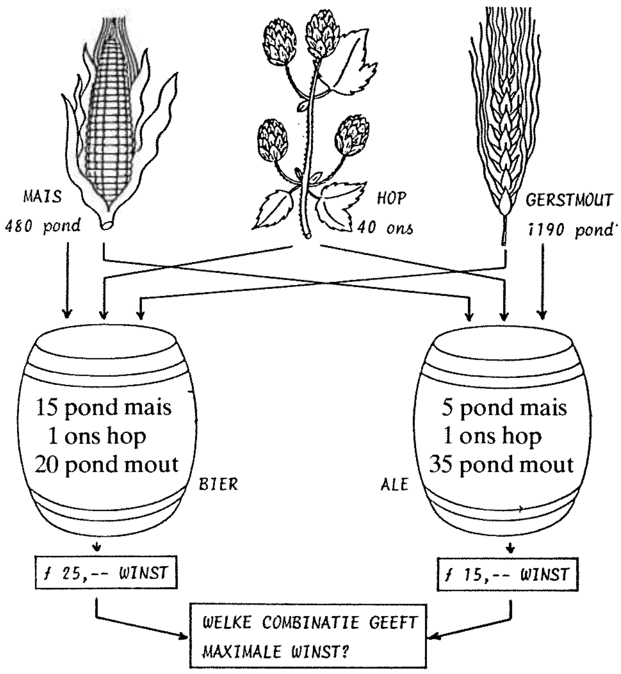
```
10 TAL =0
20 FOR HALTE=1 TO 18
30 IF HALTE<18 THEN IN=INT(21*RND(0)) ELSE
   IN=0
40 TAL=TAL+IN
50 IF HALTE>3 THEN UIT=INT(23*RND(0))
   ELSE UIT=0
60 IF HALTE=18 THEN UIT=TAL (anders blijven er
   mensen zitten)
70 IF UIT>TAL–IN THEN 50 (anders gaan er mensen
   in en meteen uit)
80 TAL=TAL–UIT
90 IF TAL>50 THEN 30 (niet meer dan 50 mensen erin)
100 PRINT "HALTE", HALTE, "IN", IN, "UIT",
    UIT, "TAL", TAL
110 NEXT HALTE
120 END
```

In Zevenaar liep deze opgave als een tram. Alhoewel de leerlingen te weinig oefenen in het programmeren om zelf echt interessante programma's te kunnen maken, kan met een opgave als deze, waarvan de leerlingen het programma wel kunnen lezen en eventueel aanpassen, toch enig inzicht verkregen worden in de werking van een programma.

Bier of Ale

Een van de extra onderwerpen die in het HEWET-rapport genoemd worden en waarvan de haalbaarheid onderzocht moet worden is: "Algoritmen voor het lineair programmeren met meer dan twee variabelen, ook met behulp van standaardprogramma's en computerverwerking daarvan". Hiertoe wordt in het pakketje Lineair Programmeren ook de Simplexmethode behandeld en is een standaardprogramma beschikbaar.
De introductie op LP wordt gevormd door het volgende bierbrouwersprobleem:

▶ 1 *Een bierbrouwer maakt twee soorten bier: "Bier" en "Ale". Om een vat bier te maken heeft de brouwer nodig: 15 pond mais, 1 ons hop en 20 pond mout. Om een vat ale te produceren heeft hij nodig: 5 pond mais, 1 ons hop en 35 pond mout. De winst op een vat bier bedraagt ƒ 25,–, op een vat ale ƒ 15,–. De brouwer beschikt over 480 pond mais, 40 ons hop en 1190 pond mout. Bij welke combinatie van de twee biersoorten maakt de brouwer de meeste winst?*



Het probleem van de bierbrouwer in beeld gebracht.

Na de nodige oefening in het grafisch oplossen van dit soort problemen (ook driedimensionaal) komt het bierbrouwersprobleem verderop in het boekje terug bij de behandeling van de Simplexmethode. Deze algoritme bestaat uit de volgende stappen:

1. Stel de *doelfunctie* en *beperkende voorwaarden* op.

   Doelfunctie: $W = 15x_1 + 25x_2$
   Beperkende voorwaarden:

$$\begin{aligned} 5x_1 + 15x_2 &\leq 480 && \text{(maïs)} \\ x_1 + x_2 &\leq 40 && \text{(hop)} \\ 35x_1 + 20x_2 &\leq 1190 && \text{(gerstmout)} \end{aligned}$$

2. Voer *spelingsvariabelen* in. (Kanonieke vorm).

$$\begin{array}{lcl} 5x_1 + 15x_2 + s_1 & = & 480 \\ x_1 + x_2 \qquad + s_2 & = & 40 \\ 35x_1 + 20x_2 \qquad\quad + s_3 & = & 1190 \\ \hline 15x_1 + 25x_2 & = & W \end{array}$$

3. Maak het *simplex-tableau*; vind eerste basisoplossing.

5	15	1	0	0	480
1	1	0	1	0	40
35	20	0	0	1	1190
15	25	0	0	0	W

Oplossing: (0, 0, 480, 40, 1190)
Winst: 0

4. Ga op zoek naar tweede basisoplossing:
   a. Veeg de kolom met de *grootste coëfficiënt* in de *doelfunctie*. Maak eerst de elementen gelijk aan 1 in die kolom.

$$\begin{array}{ccccc|l} 1/3 & 1 & 1/15 & 0 & 0 & 32 \leftarrow 2 \\ 1 & 1 & 0 & 1 & 0 & 40 \\ 1\,3/4 & 1 & 0 & 0 & 1/20 & 59{,}5 \\ \hline 15 & 25 & 0 & 0 & 0 & W \\ & \uparrow & & & & \\ & 1 & & & & \end{array}$$

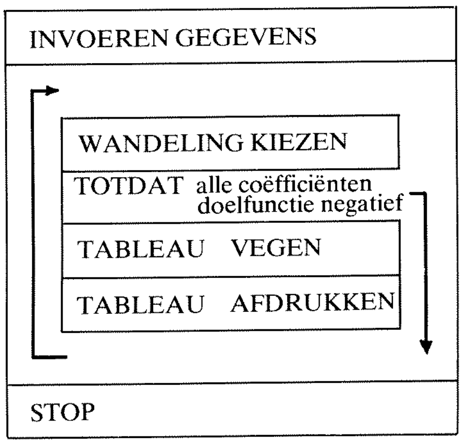
   b. Kijk daarna naar de constantes rechts van de streep. De *kleinste constante* (kleinste afstand) bepaalt welk element in de veegkolom de 1 is.
   c. Bepaal na het vegen een tweede basisoplossing.

$$\begin{array}{ccccc|l} 1/3 & 1 & 1/15 & 0 & 0 & 32 \\ 2/3 & 0 & -1/15 & 1 & 0 & 8 \\ 85/3 & 0 & -4/3 & 0 & 1 & 550 \\ \hline 6\,2/3 & 0 & -5/3 & 0 & 0 & W-800 \\ \uparrow & & & & & \\ 1 & & & & & \end{array}$$

   Oplossing: (0, 32, 0, 8, 550)
   Winst: 800

5. Herhaal stap 4 totdat alle coëfficiënten van de doelfunctie negatief zijn. Dan ben je op het hoogste punt.

Kortom, een algoritme die bij uitstek geschikt is om door een computer te laten uitvoeren. Er is dan ook een standaardprogramma voor beschikbaar. Bij de beschrijving van zo'n standaardprogramma kan een structuurdiagram weer uitstekende diensten bewijzen.



De stappen zoals de leerlingen die eerst zelf uitgevoerd hebben, zijn hierin duidelijk terug te vinden. Wie de leerlingen niet al te zeer voor de gek wil houden, is nu natuurlijk wel gedwongen het programma ook werkelijk zo op te bouwen, goed gestructureerd dus. De standaardprogramma's zullen in meerdere talen verspreid worden, maar steeds zal de opbouw in subroutines volgens het structuurdiagram aangehouden worden. Omdat de programma's niet al te ingewikkeld zijn, lukt dit ook voor BASIC nog wel.
De verwerking van het bierbrouwersprobleem met behulp van dit programma levert:

```
programma simplexmethode
het aantal beslissingsvariabelen is (0<v<=20) ?2
het aantal restricties r is (0<r<=20) ?3
toets coefficienten beperkende voorwaarden en
doelfunctie in, een voor een
aan het eind gelegenheid tot verbeteren van fouten
restrictie  1  :
?5
?15
rechterlid: ?480
restrictie  2  :
?1
?1
rechterlid: ?40
restrictie  3  :
?35
?20
rechterlid: ?1190
doelfunctie :
?15
?25
het simplex tableau ziet er zo uit:
     5.00    15.00     1.00     0.00     0.00   480.00
     1.00     1.00     0.00     1.00     0.00    40.00
    35.00    20.00     0.00     0.00     1.00  1190.00
    15.00    25.00     0.00     0.00     0.00     0.00
de waarde van de doelfunctie is            0.00
zijn alle gegevens goed? (j/n) ?j
het simplex tableau ziet er zo uit:
     0.33     1.00     0.07     0.00     0.00    32.00
     0.67     0.00    -0.07     1.00     0.00     8.00
    28.33     0.00    -1.33     0.00     1.00   550.00
     6.67     0.00    -1.67     0.00     0.00  -800.00
de waarde van de doelfunctie is   800.00
het simplex tableau ziet er zo uit:
     0.00     1.00     0.10    -0.50     0.00    28.00
     1.00     0.00    -0.10     1.50     0.00    12.00
     0.00     0.00     1.50   -42.50     1.00   210.00
     0.00     0.00    -1.00   -10.00     0.00  -880.00
de waarde van de doelfunctie is   880.00
de optimale oplossing is nu gevonden
```

Dankzij het gebruik van de computer is het mogelijk dat de leerlingen zich vooral kunnen richten op het gieten van de gegevens in LP gedaante en het interpreteren van de resultaten, terwijl toch grotere problemen aangepakt kunnen worden.

Lopende band werk

De analyse poot van wiskunde A is zoals bekend sterk gericht op toepassingen. Voor de differentiaalrekening betekent dit, dat de vraagstukken wiskundig gezien minder complex zijn dan de huidige wiskunde 1 opgaven. Daar staat tegenover, dat het formuleren van wiskundige modellen van bijv. optimaliseringsproblemen de nodige aandacht krijgt.
Het hoofdstuk "Optimaliseren" uit "Differentiëren 2" begint met:

▶ *31. In een fabriek werken aan een lopende band 40 mensen, die onderdelen klaarmaken voor electronische apparatuur.*
De gemiddelde dagproduktie per man/vrouw is 100 onderdelen. De bedrijfsleider wil de totale produk-

tie opvoeren en extra personeel aantrekken. Het effect van meer mensen aan de lopende band is echter dat de gemiddelde produktie per man/vrouw terugloopt. De bedrijfsleider vermoedt op grond van zijn ervaringen, dat iedere man/vrouw extra, de gemiddelde dagproduktie (p.p.) met twee onderdelen doet verminderen. Als hij dus 43 mensen laat werken aan de lopende band is de gemiddelde dagproduktie 94 onderdelen.
Ga bij beantwoording van de volgende vragen ervan uit dat de bedrijfsleider "goed zit" met zijn schatting.
a. Vergelijk de totale dagproduktie van 40 mensen met de totale produktie van 43 mensen aan de lopende band.
b. Je ziet dat bij een toename van 40 naar 43 mensen de totale produktie is gestegen. Neemt de bedrijfsleider echter veel meer mensen in dienst, bijv. 60, dan loopt de totale produktie terug door de sterke daling van de individuele produkties.
Hoe groot is de totale produktie van 60 mensen?

Bij een opgave als deze ligt het tamelijk voor de hand om voor het bepalen van de optimale situatie eerst eens een tabelletje te maken. Maar dat kan de computer ook wel voor ons doen. Als steuntje in de rug is het structuurdiagram om de produktie te berekenen voor 40, 41, ... 59 werknemers, gegeven:

20 KEER (n van 0 tot 19)
$a \leftarrow 40+n$
$p \leftarrow 100-n*2$
$t \rightarrow a*p$
UITVOER a, t
STOP

▶32. *a. Wat stellen de variabelen n, a, p, t in het structuurdiagram voor?*
b. Maak een programma bij dit structuurdiagram, laat dit verwerken door de computer en noteer de resultaten.

Aangezien het structuurdiagram van dit probleem erg dicht ligt bij het computerprogramma (in welke taal dan ook) zal het vertalen niet al te moeilijk zijn.
De computer is hier vooral ingezet als didactisch hulpmiddel. Je zou met een rekenmachientje ook wel een tabel kunnen maken, maar als je het op de computer doet, ben je gedwongen om op een heel natuurlijke manier de nodige variabelen in te voeren. En deze variabelen heb je nu net nodig om het wiskundig model bij het probleem op te stellen. Verderop in het hoofdstuk wordt hier gebruik van gemaakt:

De eindvariabele t in het structuurdiagram is een functie van de beginvariabele n. Door substitutie van de tussenvariabelen a en p, kun je t als functie van n vinden.

Kijk maar:

$t = a \cdot p$

$40+n$ $\quad$ $100-2n$

dus: $t = (40+n)(100-2n)$
ofwel: $t = -2n^2 + 20n + 4000$

▶38. *Het maximum van deze functie kan nu worden gevonden met behulp van de afgeleide functie. Hoe?*

En zo had de lezer het natuurlijk meteen al willen doen!

Tenslotte

Van de ruim 50 scholen die in de experimentele fase wiskunde A (gaan) onderwijzen, heeft op dit moment bijna eenderde deel in het geheel geen computer apparatuur, de rest bezit iets tussen de 1 en 11 micro's (of een eigen schrapkaartenlezer). Geen twee scholen hebben dezelfde faciliteiten. Zoveel zielen, zoveel gedachten. Dit brengt uiteraard nogal wat praktische problemen met zich mee (over de meer fundamentele bezwaren die hier aan kleven is elders in deze Nieuwe Wiskrant meer te lezen): zowel voor wat betreft de leerstofontwikkeling, als ook voor de organisatie van AGV op school-niveau. Ter overdenking een kleine opsomming van de problemen en probleempjes waar elke leraar wel eens tegen aan zal lopen:

- Bij het werken met schrapkaarten moet er rekening mee gehouden worden dat het twee tot drie dagen duurt voor de output terug ontvangen is.
- Om een goede output te krijgen, moeten de leerlingen heel nauwkeurig schrappen. Dit klinkt makkelijker dan het blijkt te zijn!
- De scholen die minder dan zeg zes micro's hebben (van de vijftig scholen is dat thans ongeveer eenderde deel) zullen het computergebruik extra goed moeten organiseren om alle leerlingen aan hun trekken te laten komen.
- Is de nodige apparatuur op elk gewenst moment inderdaad voor wiskunde A beschikbaar, of moet het computerlokaal zolang van te voren gereserveerd worden dat net die ene toepassing maar overgeslagen wordt?
- Is er een printer beschikbaar of moeten de leerlingen alle resultaten van het scherm overschrijven?
- Laten de leerlingen zich niet te veel afleiden door de apparatuur waardoor ze aan nadenken en interpreteren niet meer toekomen?
- Voor scholen die alleen BASIC tot hun beschikking hebben:
  de standaardprogramma's moeten voordat ze gebruikt kunnen worden eerst omgezet worden in het BASIC dialect van de eigen machine.
- Meer onderwijskundig: Wat doe je met het verschil in voorkennis van leerlingen? (wel of niet computerkunde in de onderbouw)
- Last but not least: Hoe toets je AGV?

Aan de hand van de ervaringen van de 12 scholen die het komend cursusjaar (al dan niet voor het eerst) met wiskunde A en dus ook AGV van start gaan, hopen we in de toekomst meer over al dit soort zaken te kunnen rapporteren.