

# Van dictaat tot boek

## Cees Fortuin

Technische Faculteit  
Hogeschool Gelderland  
Ruitenberglaan 26  
6826 CC Arnhem

### Abstract

In 1980 werd ik docent natuurkunde aan de HTS te Arnhem voor de afdeling elektrotechniek. Toentertijd werd een leerboek gebruikt dat heel aardig de meer traditionele componenten behandelde, maar bijna niets wist te melden over wat tegenwoordig in een schakeling zit: dioden, transistoren en dat soort componenten. Een eerste navraag bij landelijke collega's leverde wel een soort werkgroepje op, maar niet de gezochte 'nieuwe' leerstof. Dus maar zelf schrijven. Dat moet in eerste instantie letterlijk worden opgevat, maar ook met een tekstverwerker als L<sup>A</sup>T<sub>E</sub>X moet veel zelf worden gedaan. Wat begon rond 1985 als dictaat, per hoofdstuk afgedrukt, eindigde rond 1995 als boek: twee delen 'Elektron natuurkunde' in de serie *natuurkunde voor het HBO* bij de Educatieve Uitgevers.

## 1 Van tekstverwerker naar tekstverwerker

In eerste opzet werd de geschreven tekst door een behulpzame hand bij de administratie uitgetikt: formules en tekeningen werden met de hand toegevoegd. In een volgend stadium werd de tekstverwerker van de Prime minicomputer gebruikt—maar dat was een ramp: praktisch onleesbare afdrukken; de studenten klaagden steen en been. De komst van de pc ging bij ons gepaard met 'Word Perfect'. Het lettertype werd daarbij ietsje beter, maar formules, in het bijzonder die in de lopende tekst, of griekse letters: ho maar. Door een artikel in, ik dacht, de American Mathematical Review, waarin tekstverwerkers werden vergeleken, kwam ik op het spoor van T<sub>E</sub>X. Doch hoe aan een exemplaar te komen van een Amerikaans produkt: een collega die naar een 'vacantie' cursus aan het MIT ging kreeg een papier mee waarop mijn wensen stonden. Na (mijn) vakantie bleek hij een campusexemplaar op de kop te hebben getikt van μT<sub>E</sub>X. Precies herinner ik me het niet meer, maar ik meen dat de fonts nogal onvolledig waren. Dus ging ik op zoek naar andere T<sub>E</sub>X 'leveranciers', en die vond ik in eerste instantie bij de universiteit in Utrecht, waar PCT<sub>E</sub>X werd gebruikt—overschakelen dus. Maar ook daar waren weer problemen: het bijgeleverde afdrukprogramma (is dat de vertaling van 'printerdriver'?) liet na zo'n 25 pagina's correct te hebben afgedrukt op de volgende lettertypen 'verdwijnen'. Ondertussen lid geworden van de NTG vond ik via de ledenlijst iemand die een betere 'dvidriver' wist: ik werd geruisloos fervent emT<sub>E</sub>X gebruiker.

## 2 De decimale komma

Bij het schrijven kwamen verschillende problemen te voorschijn waarop niet een 'onmiddellijk' antwoord was. Sommige daarvan verdienen ruimere aandacht, zoals het pro-

bleem van de decimale komma. Van huis uit is T<sub>E</sub>X een Amerikaanse tekstverwerker, het draagt daar de sporen van. In natuurkunde tekst komen natuurlijk heel veel grootheden voor, bestaande uit getallen en eenheden. In Amerika hebben getallen een decimale punt: 1.2, in Europa behoren zulke een decimale komma te hebben: 1,2; achter de komma wordt echter teveel ruimte gezet. Naast de *komma* en de *punt* heeft ook de *ster* een bijzondere positie binnen wiskundige tekst. Dus werden die drie aangepast met de volgende code (zie T<sub>E</sub>Xbook 344):

```
\gdef\plainmathcomma {\mathcode'\,=24891\relax}
%      reset plain mathcomma
\gdef\decimalekomma {\mathcode',=315 \relax}
%      ', als normale komma
\decimalekomma
%      initialisatie
\gdef\plainmathperiod {\mathcode'\.=314\relax}
%      reset plain mathperiod
\gdef\produktpunt {\mathcode'.'.513 \relax}
%      '.' als gewone cdot
\produktpunt
%      initialisatie
\gdef\plainmathstar {\mathcode'*=8707 \relax}
%      reset plain mathstar
\gdef\produktster {\mathcode'*=8706 \relax}
%      '** als binaire maal
\produktster
%      initialisatie
```

Daarmee is 1,2 kilometer: 1,2·10<sup>3</sup> m. Alles leek in orde, maar dat was het niet. In eerste aanzet waren de codegetallen namelijk *hexadecimaal* geschreven achter het symbool ". Helaas ging dat fout bij het gebruik van babel-nederlands: daar wordt " actief gemaakt. Overgaan naar *decimale* codegetallen was de eenvoudigste oplossing.

## 3 Het hok om formules

Ohm wist het al:

$$R = \frac{U}{I}$$

Om het belang van de formule aan te geven zetten we er een hok om. Maar `\[ \fbox{...} \]` geeft helaas foutmeldingen vanwege missende `$`'s. Ook `\[ \fbox{$. . . $} \]` geeft niet wat werd beoogd:

$$R = \frac{U}{I}$$

Ziet u wat er aan de hand is? De `\fbox` werkt alleen voor tekst (is dat nu een 'bug' van  $\text{\LaTeX}$ ?). Na enig gepuzzel werd de gebruikersvriendelijkere `\fbox`:

```
\let\LaTeXfbox=\fbox
\global\def\newfbox#1{\ifmmode\ifinner%
\LaTeXfbox{#1$}
\else\LaTeXfbox{\$ \displaystyle #1$}\fi
\else\LaTeXfbox{#1}\fi}
\let\fbox=\newfbox
```

waarmee Ohm's grootheid recht wordt gedaan (jawel, dit is dubbelzinnig taalgebruik):

$$R = \frac{U}{I}$$

#### 4 Extensie bestanden

Een ander probleem deed zich voor bij het verwerken van antwoorden van opgaven en dergelijke. Met behulp van de  $\text{\LaTeX}$  'environment' `list` was een `\opgave` gedefinieerd. Bij meerkeuzeopgaven worden de alternatieven gezet door een macro `\keus` of `\goedekeus`, die het antwoord (optioneel voorzien van commentaar) wegschrijft naar het bestand met extensie '.MKA'. Die antwoorden moesten per hoofdstuk worden verwerkt. Dat gaf twee problemen:

1. hde antwoorden (of de inhoudsopgave) *per hoofdstuk* af te drukken;
2. hoe het aantal gelijktijdig *openstaande* bestanden (aux, toc, lof, idx, mka, vwl, ...) te beperken;

$\text{\LaTeX}$  maakt voor het gehele document één extensiebestand aan, alhoewel de .AUX bestanden per 'include'-bestand worden gemaakt. Gekozen werd voor een andere manier van omgaan met extensie bestanden.

Er werd een stijl: 'EXT.STY' ontworpen, die het verwerken van het '.AUX' bestand zodanig verandert dat ook extensie bestanden in principe per 'include'-bestand wordt verwerkt. Verder worden de verschillende extensiebestanden, aan het eind van het document, één voor één verwerkt, waardoor het aantal tegelijk openstaande bestanden drastisch verminderde. Daartoe moest `\enddocument` worden aangepast, hetwelk een forse versie afhankelijke ingreep is in de code. Verder waren aanpassingen nodig van `\include`, `\@starttoc`, `\@inputaux`, `\@writefile`, `\addtocontents`, `\addcontentsline`. Maar: nu bepaalt het bestand waarin de aanroep van bijv. `\tableofcontents` staat of alleen de inhoudsopgave van het hoofdstuk of die van alle hoofdstukken (of beide, indien gewenst) worden afgedrukt.

#### 5 Figuren en tekst naast elkaar

Als iets me veel tijd heeft  $\text{\LaTeX}$  floats staan in een gekost, dan is het wel om 'vbox', die niet geknipt kan worden. Dus een simpele oplossing is: een `\vbox` waarin twee `verb+hbox` naast elkaar staan. Helaas, verschillende problemen lagen op de loer: Hoe zorgen we ervoor dat de tekst in de 'box' precies aansluit (regelafstand) op de tekst boven en onder de 'box' (ook als die forse halen bevat)?

Een ander op te lossen probleem was hoe mee te gaan met de verschillende linker- en rechtermarges van de  $\text{\LaTeX}$  omgeving waarin de figuur wordt gezet.  $\text{\LaTeX}$  floats zijn (waren?) *ongevoelig* voor de omgeving waarin ze worden aangeroepen: een omissie naar mijn idee.

Net toen ik dacht alles netjes te hebben opgelost kwam ik voor nieuwe verrassingen te staan: de kop van een paragraaf stond opeens in de figuur (?). Zo leerde ik het speciale mechanisme van  $\text{\LaTeX}$  om kopjes te zetten kennen: via `\everypar`, die slechts bij de overgang van 'vmode' naar 'hmode' wordt aangeroepen. Ook dat werd opgelost; uiteindelijk kreeg de omgeving `\floatfix` gestalte. Ik ben nu tevreden over de manier waarmee figuren (tabellen) links of rechts (of pagina afhankelijk) van de tekst kunnen worden gezet, in aanpasbare breedte. De figuur zelf kan bijvoorbeeld als een `\picture`-bestand worden ingelezen. Onderschriften zijn 'bijna'  $\text{\LaTeX}$ -onderschriften. (Nou ja, een beetje aangepast wat de ruimtelijke instelling betreft, maar ik geloof dat de nieuwe  $\text{\LaTeX}2_{\epsilon}$  daar nu ook in voorziet.)

#### 6 Plaatjes: $\text{\Pictex}$ en PS

Ter afsluiting de plaatjes. Binnen  $\text{\LaTeX}$  is er een zeer sterke ondersteuning van het zetten van wiskundige tekst zoals formules, maar wanneer het gaat om de visuele ondersteuning met grafieken en figuren is het heel wat minder. Dat lijnen slechts in een beperkt aantal richtingen worden ondersteund is een voorbeeld. Gelukkig ondersteunt  $\text{\emTeX}$  de `\emline` lijnen in willekeurige richtingen met *zijn* 'drivers'. Maar om andere lijnen te kunnen tekenen: *gestippeld* of *gestreept*, of om andere grafieken te kunnen tekenen als *ellipsen*, *hyperbolen*, *parabolen*, *wortelfuncties*... zijn andere hulpmiddelen nodig. Ik vond  $\text{\Pictex}$ , ook geschikt voor het *roteren* of *arceren* van figuren. Wel moesten de verschillende lengte-eenheden, lijndikten, puntgrootte, streep lengte en dergelijke van  $\text{\LaTeX}$  en  $\text{\Pictex}$  netjes worden afgestemd op elkaar. Hopelijk zullen bovenstaande 'wensen' allemaal in  $\text{\LaTeX}3$  zijn ingebouwd, want nu was ik daar ook aardig wat tijd zoet mee. Misschien dat in  $\text{\LaTeX}3$  alle 'rekenmachine functies' standaard beschikbaar kunnen zijn, zoals: `\plot{sin}`, `\plot{ln}`, `\plot{squ}`, `\plot{xtoy}`, `\plot{circular}`, ...?

Om tabellen dwars af te kunnen afdrukken (dat lijkt zo eenvoudig, maar is het niet) moest ik de hulp inroepen van PostScript. Ik gebruikte het ‘package’ ‘rotating.sty’ en het conversieprogramma DVIPS; ‘psboxit.sty’ was goed voor het zetten van rasters in figuren. Dan nog een laatste wens geformuleerd. Ik heb me afgevraagd waarom niet reeds in  $\text{\TeX}$  een optie is ingebouwd om fonts te kunnen spiegelen of dwars te draaien. Dat lijkt op het eerste gezicht niet echt moeilijk, en zou aan veel wensen tegemoet komen. Maar misschien heb ik problemen over het hoofd gezien (?).

## 7 Besluit

Voor mij was  $\text{\LaTeX}$  een onmisbaar hulpmiddel—natuurlijk, aangevuld met de vele ‘packages’ die door allerlei mensen zijn gemaakt en allerlei problemen oplossen—waarmee een dictaat kon worden omgezet in een ‘echt’ boek. Het ideaal: ‘de opmaak overlaten aan de tekstverwerker’ bleef echter bij een aantal onderwerpen buiten beeld, waardoor extra inspanning nodig was. Maar de opzet van  $\text{\LaTeX}$  kan dat ideaal wel dichterbij brengen. Bovendien: stoeien met problemen, ook bij tekstverwerking, vind ik persoonlijk nog leuk ook. Vandaar ....