Handleiding LaTeX

Piet van Oostrum
Vakgroep Informatica, Universiteit Utrecht

Versie 5
3 januari 1997

LaTeX [1, 2] is een programma, dat bijzonder geschikt is voor het vervaardigen van 'drukrijpe' technische documentatie; vooral dié documentatie die mathematische formules bevat. Het kan natuurlijk ook voor andere teksten gebruikt worden, variërend van brieven tot complete boeken. LaTeX is noch een **D**esk**T**op **P**ublishing-pakket noch een tekstverwerker, maar een **zet**-systeem. Deze korte handleiding zou voor de meeste toepassingen van LaTeX voldoende moeten zijn. Een volledige beschrijving kan men in het *LaTeX-Manual* [1] vinden.

LaTeX kan op een groot aantal verschillende computers worden geïnstalleerd. Het is te verkrijgen voor o.a. UNIX-, MSDOS-, Atari-, Amiga- en Macintosh-systemen, etc. Vaak zijn er zowel commerciële als 'public domain' versies te verkrijgen.

Deze beschrijving is gebaseerd op de nieuwste versie van LaTeX , die LaTeX 2_ε genoemd wordt. Appendix B bevat een overzicht van de verschillen met de originele LaTeX versie (LaTeX 2.09).

Deze handleiding is gemaakt met LaTeX 2ε. Gebruikt zijn: de 'report'-class, de pakketten `latexsym`, `amsfonts`, `graphicx`, `longtable` *en* `picins` *en de nederlandse taal optie (*`babel` *pakket). Voor de titel, inhoudsopgave, figuren, tabellen en literatuurverwijzingen is gebruik gemaakt van standaard LaTeX commando's.*

Dit document is gebaseerd op "LaTeX Kurzbeschreibung"[1] van
Hubert Partl[2]
Elisabeth Schlegl[3]
Irene Hyna[2]
vertaald door
André v.d. Vlies[4]

Hoofdstuk 5 is gebaseerd op "Plaatjes in een tekst" van Jeroen Fokker[5]

De auteur is te bereiken via het Internet:
e-mail: `<piet@cs.ruu.nl>`
WWW: `http://www.cs.ruu.nl/~piet`
Errata kunnen gevonden worden op `http://www.cs.ruu.nl/~piet/ltx-errata.html`

[1] Copyright © H.Partl, E.Schlegl, I.Hyna (1987)
[2] EDV-Zentrum der Technischen Universität Wien, Abt. Digitalrechenanlage
[3] EDV-Zentrum der Karl-Franzens-Universität Graz
[4] AHA-TMF, Algemene Hogeschool Amsterdam, Amsterdam
[5] Vakgroep Informatica, Universiteit Utrecht

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Hoofdstuk 1

Inleiding

1.1 The Name of the Game

1.1.1 TEX

TEX (uitspraak "Tech" als in techniek, kan ook als 'TeX' geschreven worden) is een computer-progamma van Donald E. Knuth [3]. Het is speciaal ontworpen voor het zetten en drukken van mathematische teksten en formules. Een belangrijk kenmerk van TEX is dat het *uitbreidbaar* is, het bevat een soort programmeertaal waarin ingewikkelde functies beschreven kunnen worden (met z.g. *macros*).

1.1.2 LATEX

LATEX (uitspraak "Lah-tech" of "Lee-tech", kan ook als 'LaTeX' geschreven worden) is een zogenaamd macro-pakket dat door Leslie Lamport [1] is geschreven en van TEX gebruik maakt. Het stelt de auteur in staat zijn publicaties op eenvoudige wijze en met gebruik van een van te voren opgegeven structuur, met boekdruk-kwaliteit te zetten en af te drukken. Het wordt veel gebruikt voor wetenschappelijke artikelen en boeken, vooral in de exacte wetenschappen.

1.2 Achtergrond

1.2.1 Auteur, ontwerper en zetter

Bij de meeste publicaties geeft de auteur de uitgever een (vaak getypt) manuscript. De ontwerper van de uitgeverij beslist dan over de layout van de publicatie (regellengte, lettertype, spatiëring etc.) Het document draagt dan het stempel van de ontwerper en niet van de auteur.

LATEX is te beschouwen als de ontwerper, TEX als de zetter. De LATEX-commando's worden in TEX-commando's vertaald.

Een menselijke ontwerper herkent de doelstellingen van de auteur meestal vanuit zijn vak-bekwaamheid en de inhoud van het manuscript. LATEX is daarentegen 'alleen maar' een programma en heeft dus informatie van de auteur nodig, waarmee de logische struktuur van de tekst aangegeven wordt. Deze informatie wordt door middel van zogenaamde *commando's* binnen de tekst aangegeven.

Tekstverwerkers en DTP (DeskTop Publishing) pakketten zijn anders van opbouw. Hiermee wordt op 'interactieve' wijze de layout van de publicatie vastgelegd. Op het scherm is te zien hoe de pagina er uit gaat zien. Zulke pakketten worden daarom ook wel WYSIWYG pakketten genoemd (What You See Is What You Get).

Met LaTeX ziet de auteur over het algemeen niet hoe de tekst die hij/zij aan het invoeren is er uit zal zien. Het is wel altijd mogelijk om een proefafdruk te maken of op het beeldscherm te laten zien.

De eigenlijke layout is bij LaTeX gescheiden van de invoer van de auteur. Op deze manier kan voor een andere layout gekozen worden zonder dat de auteur het document (ingrijpend) moet veranderen, bijvoorbeeld wanneer een artikel naar een andere uitgever gaat. In dat geval kan de uitgever bepalen hoe de uiteindelijke layout eruit komt te zien. We spreken daarom bij LaTeX van *logische* i.p.v. *visuele* opmaak. Bij sommige tekstverwerkingssystemen kan iets soortgelijks met behulp van z.g. *styles* gerealiseerd worden.

1.2.2 Layout-ontwerp

Typografisch ontwerpen is een vak, dat men moet (kan) leren. Ongeoefende auteurs maken vaak zware typografische fouten. Vele leken denken, ten onrechte, dat typografie alleen maar een kwestie van smaak is; wanneer een document er 'mooi' uitziet, is het ook goed ontworpen. Daar documenten echter gelezen dienen te worden, zijn 'leesbaarheid' en begrijpbaarheid belangrijker dan het uiterlijk.

Voorbeeld: De lettergrootte en nummering van titels van hoofdstukken en paragrafen moet zo gekozen worden, dat de de struktuur van hoofdstukken en alinea's duidelijk herkenbaar is. De regellengte dient zo gekozen te worden, dat vermoeiende oogbewegingen voor de lezer voorkomen worden en niet zo, dat het papier zo mooi mogelijk 'gevuld' wordt met letters.

Met DTP-pakketten maken auteurs soms 'esthetisch mooie', maar slecht leesbare documenten. Een veel gemaakte fout is bijvoorbeeld het gebruiken van teveel verschillende lettertypes. LaTeX voorkomt typografische fouten, omdat het de auteur dwingt de logische struktuur van een tekst aan te geven. LaTeX gebruikt dan automatisch de meest geschikte layout.

1.2.3 Voor- en nadelen

LaTeX heeft, vergeleken met andere tekstverwerkingspakketten, de volgende voordelen:

- Er zijn diverse professioneel vormgegeven layouts beschikbaar, waarmee de documenten er inderdaad als 'gedrukt' uitzien. Uitgeverijen en tijdschriften hebben vaak hun eigen voorgedefinieerde layout.
- Het zetten van mathematische formules en ingewikkelde tabellen wordt bijzonder goed ondersteund.
- De gebruiker hoeft maar een paar, gemakkelijk te begrijpen, commando's te leren. Deze commando's betreffen alleen de logische struktuur van het document, de gebruiker hoeft zich nauwelijks bezig te houden met de druktechnische details.

- Complexe strukturen zoals voetnoten, literatuuropgaven, inhoudsopgaven, tabellen etc. en zelfs eenvoudige tekeningen, kunnen zonder al te veel moeilijkheden gemaakt worden.
- Het is beschikbaar voor bijna alle computersystemen, en meestal gratis of goedkoop.

LaTeX heeft natuurlijk ook nadelen:

- Het systeemverbruik (rekentijd en opslagcapciteit) is hoger dan bij primitievere tekstverwerkers. Moderne tekstverwerkers hebben echter een even zwaar systeem nodig als LaTeX.
- De documenten kunnen alleen op laserprinters, inktjetprinters of professionele zetapparatuur afgedrukt worden. De 'goedkopere' (matrix) printers voldoen nauwelijks en worden over het algemeen alleen maar voor kladversies gebruikt. Omdat tegenwoordig betrekkelijk goedkope inktjetprinters te koop zijn die prima voldoen, is dit punt niet zo belangrijk meer.
- Binnen de door LaTeX ondersteunde layouts kunnen weliswaar enige parameters gevarieerd worden, maar ingrijpende afwijkingen van de gebruikte layout kunnen slechts met veel moeite tot stand gebracht worden (Ontwerp van een nieuwe 'documentclass').
- Het intypen van LaTeX commando's is ingewikkelder dan het aanbrengen van layout met moderne tekstverwerkers, zeker als die voorzien zijn van een menu-besturing. Ook al omdat het resultaat van de commando's niet direct zichtbaar is.
- Het is gemakkelijker fouten te maken met LaTeX dan met een tekstverwerker en de foutmeldingen van LaTeX zijn niet altijd even duidelijk.

Kort gezegd zijn voor kleine documenten (brieven, posters, memo's) tekstverwerkers meestal in het voordeel, terwijl voor grote, complexe documenten LaTeX vaak betere resultaten en uiteindelijk minder werk oplevert.

1.3 Werkwijze

Om met LaTeX een document te maken moet je (met een tekst-editor) een file aanmaken waarin je tekst staat, samen met de aanwijzingen (commando's) waaraan LaTeX kan zien hoe de tekst opgemaakt kan worden. Er bestaan editors die LaTeX documenten begrijpen en je kunnen helpen met het bijhouden van de logische struktuur, maar elke gewone tekst-editor kan ook gebruikt worden. Als de tekst klaar is wordt hij verwerkt door het LaTeX programma, waaruit een z.g. *DeVice Independent* (DVI) file komt, die daarna met een *previewer* bekeken kan worden of met een *printer-driver* afgedrukt. Zie de Local Guide (appendix A) voor details.

De rest van deze handleiding behandelt eerst de structuur van gewone alledaagse tekst (hoofdstuk 2), daarna wordt aandacht besteed aan tabellen (hoofdstuk 3), formules (hoofdstuk 4), figuren (hoofdstuk 5) en geavanceerde documentstructuren.

Tabel 1.1: Document Classes

`article` voor artikelen in wetenschappelijk tijdschriften, voordrachten, praktikumhandleidingen, seminars, korte berichten, programmabeschrijvingen, uitnodigingen, etc.

`artikel1,artikel2,artikel3` Als boven, maar dan met een nederlandse layout. De nederlandse layout is érg verschillend van de amerikaanse.

`report` voor langere stukken, die uit meer hoofdstukken bestaan, dissertaties, scripties, etc.

`rapport1,rapport3` een nederlandse rapport stijl.

`book` voor boeken.

`boek` een nederlandse boek stijl.

`letter` voor brieven.

`brief` voor nederlandse brieven (volgens de NEN-normen).

`slides` voor overhead transparanten.

1.4 De structuur van een LaTeX document

Een LaTeX invoer document ziet er altijd als volgt uit:

`\documentclass[`*options*`]{`*class*`}`

preamble

`\begin{document}`

het document

`\end{document}`

Tussen de accolades bij `\documentclass` *moet* één van de in tabel 1.1 genoemde document classes aangegeven worden. De document class vertelt wat voor soort document we hebben, en dit bepaalt o.a. de layout. De *options* geven variaties op de document class aan. Zie tabel 1.2 op pagina 12.

De *preamble* kan definities bevatten die het gehele document beïnvloeden, bijvoorbeeld eigen gedefinieerde commando's, selectie van een ander lettertype voor het hele document, of het laden van uitbreidingen van LaTeX. Er mogen in de preamble geen dingen voorkomen die tot gevolg hebben dat iets afgedrukt zou worden.

Tabel 1.2: Document Style Options

`11pt` voor een 11 punts letter. Deze handleiding is ook met een 11 punts letter gezet.

`12pt` voor een 12 punts letter.

Zonder een van deze twee opties gebruikt LaTeX een 10 punts letter, wat in boekdruk de 'normale' grootte is.

`fleqn` voor links uitgelijnde in plaats van gecentreerde vergelijkingen.

`leqno` om vergelijkings nummers links in plaats van rechts van de vergelijking te krijgen.

`titlepage` om een 'losse' titel pagina te verkrijgen. (in documentclass `article`)

`notitlepage` om niet een 'losse' titel pagina te verkrijgen. (in documentclass `report` en `book`)

`twocolumn` om een tweekoloms output te verkrijgen.

`twoside` om het document dubbelzijdig te zetten (de linker en rechter pagina zijn verschillend!). In document class `book` is dit standaard. Daar kan de `oneside` optie gebruikt worden om dit niet te doen.

`landscape` om het papier in de breedte te gebruiken, dus met de lange kant horizontaal. In het algemeen zal dan ook een speciale printopdracht gegeven moeten worden!

`a4paper` om alles netjes op een A4-tje te krijgen. (de amerikaanse papierformaten zijn anders).

De tekst van het document tenslotte staat tussen de `\begin{document}` en de `\end{document}`. De tekst zal in het algemeen commando's bevatten die de structuur van het document aangeven.

Fig. 1.1 laat een (bijna) minimale LaTeX-file zien. Een ietwat complexer voorbeeld is in fig. 1.2 geschetst.

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

Figuur 1.1: Een minimale LaTeX-file

1.5 LaTeX-commando's

De meeste LaTeX-commando's zien er als volgt uit: ze beginnen met een backslash (`\`) en hebben dan óf een naam die alleen uit letters bestaat en die door één of meer spaties of een bijzonder teken of een cijfer beëindigd worden, óf die bestaat uit precies één speciaal teken of cijfer. Hoofd- en kleine letters hebben óók in commando's een verschillende betekenis. Als men na een commando dat uit letters bestaat een spatie in de uitvoer wil hebben moet men `{}` gebruiken om het commando te beëindigen of het commando `\ ` (`\` gevolgd door een spatie) gebruiken om de spatie te verkrijgen. Dit komt doordat LaTeX na zo'n commando de spaties negeert (anders was er geen manier om verder te gaan met letters in de tekst).

Commando's worden voor verschillende doelen gebruikt, o.a.:

- om de structuur van een document aan te geven, bijvoorbeeld `\chapter` voor een nieuw hoofdstuk.
- om symbolen aan te geven die niet op het toetsenbord voorkomen bijvoorbeeld `\copyright` voor ©.
- als afkorting voor een stuk tekst, bijv. `\LaTeX` voor LaTeX.
- om de layout te beïnvloeden bijv. `\large` om een groter lettertype te selecteren.

Het is ook mogelijk eigen commando's te definiëren, zie hoofdstuk 6.

Vandaag is het 3 januari 1997.
Of: Vandaag is het 3 januari 1997.
Fout is: 3 januari 1997regent het.
Goed: 3 januari 1997 schijnt de zon.
Of: 3 januari 1997 sneeuwt het.
	`Vandaag is het \today.`
	`Of: Vandaag is het \today .`
	`Fout is:`
	` \today regent het.`
	`Goed:`
	` \today{} schijnt de zon.`
	` Of: \today\ sneeuwt het.`

```
\documentclass[11pt]{artikel1}
\usepackage[dutch]{babel}
\author{Piet van Oostrum}
\title{Over \LaTeX}
\setlength{\parindent}{0pt}
\setlength{\parskip}{5pt plus 2pt minus 1pt}
\frenchspacing
\sloppy

\begin{document}
\maketitle
\begin{abstract}
Voorbeeld van een wetenschappelijk artikel
in de nederlandse taal.
\end{abstract}
\tableofcontents

\section{Begin}

Hier komt dan dat fraaie artikel\ldots

\section{Einde}

\ldots\ en hier eindigt het.

\end{document}
```

Figuur 1.2: Opbouw van een artikel

Veel commando's hebben parameters die tussen accolades `{}` aangegeven dienen te worden. Andere commando's hebben parameters die weggelaten kunnen worden of tussen blokhaken aangegeven moeten worden. Zie als voorbeeld de *opties* hierboven in het `\documentclass` commando. Sommige commando's hebben varianten die door het tussenvoegen van een ster *aan*- of *uit*gezet kunnen worden.

1.5.1 Commentaar

Alles wat achter een procentteken (`%`) staat (tot aan het einde van de regel) wordt niet gezien door LaTeX. Dit kan door de auteur gebruikt worden om notities te maken die niet gedrukt dienen te worden. Bijvoorbeeld om uitleg te geven over een ingewikkelde constructie. Of aantekeningen over iets dat nog toegevoegd of gewijzigd moet worden. Wanneer een regel met commentaar eindigt, geldt de regelovergang niet als spatie. Daarom staat in het volgende voorbeeld een spatie voor het `%`-teken, anders zouden de woorden 'een' en 'voorbeeld' aan elkaar gedrukt zijn. Aan de andere kant kun je met een `%` aan het einde van een regel ongewenste spaties verwijderen.

Dit is een voorbeeld.

```
Dit is een % dom
% Beter: een leerzaam <---
voorbeeld.
```

1.6 Pakketten

Pakketten zijn uitbreidingen op LaTeX waarvan je per document moet aangeven dat ze gebruikt moeten worden. Dit gebeurt met het commando

`\usepackage[`*options*`]{`*packages*`}`

Er bestaan zeer veel pakketten en appendix D geeft een overzicht van de belangrijkste. De meeste pakketten hebben hun eigen documentatie (zie de Local Guide).

Pakketten worden voor verschillende doelen gebruikt:

- Om de layout van een document class te wijzigen, bijv. `a4` om een andere pagina afmeting te gebruiken. Deze pakketten introduceren geen nieuwe commando's.
- Pakketten die bestaande LaTeX commando's uitbreiden of aanpassen, bijv. `array` of `theorem` die de gelijknamige omgevingen aanpassen.
- Pakketten die nieuwe functies creeren met meestal ook nieuwe commando's, bijvoorbeeld `graphics` voor het invoegen van plaatjes.

Tabel 1.3 op pagina 16 geeft alvast een overzicht van de meest gebruikte pakketten.

Tabel 1.3: De belangrijkste pakketten

`a4` Gebruik A4 papier. Dit pakket gebruikt een groter oppervlak van het papier dan de `a4paper` optie.

`array` Uitbreiding van de omgevingen `array`, `tabular` en `tabular*`.

`babel` Uitbreiding voor meertalige of niet-engelse documenten. Als optie worden de gebruikte talen opgegeven.

`color` Definieert commando's voor kleurgebruik.

`float` Uitbreidingen voor de `figure` en `table` omgevingen.

`graphics` Het invoegen van plaatjes die uit andere programma's komen.

`identfirst` Verandert de eerste alinea van elke sectie zo dat deze ook inspringt.

`latexsym` Maakt extra LaTeX symbolen beschikbaar.

`multicol` Maakt meerkoloms uitvoer mogelijk.

`showkeys` Drukt de namen van `\label`s, `\ref`s und `\pageref`s af.

`theorem` Uitbreidingen van de `theorem`-omgeving.

Hoofdstuk 2

Platte tekst

Onder 'platte' tekst verstaan we simpele tekst zonder toeters en bellen, zoals formules, tabellen en plaatjes. Bijvoorbeeld een brief of een simpel memo. Het grootste gedeelte van zulke tekst kan gewoon ingetypt worden, maar er zijn een paar dingen waarop gelet moet worden.

2.1 Spaties

LaTeX verzorgt de layout van je document. Het is daarom niet van belang wáár in je invoer de spaties en de regelovergangen staan omdat LaTeX zich daar toch niets van aantrekt. We hebben al gezien dat spaties direct na een commando dat uit letters bestaat helemaal genegeerd worden. Verder geldt een rijtje spaties alsof er maar één staat. Een overgang op een nieuwe regel geldt ook als spatie en spaties aan het begin en eind van een regel worden compleet genegeerd. Als je een spatie wilt invoeren die niet door LaTeX genegeerd wordt dan moet je het commando `\ ` (`\` gevolgd door spatie) gebruiken. Omdat spaties aan het einde van een regel genegeerd worden moet dit dus niet aan het eind van een regel gedaan worden.

2.2 Alinea's

Een lege regel (of een aantal opvolgende lege regels) geldt echter als een alinea overgang. Dat wil nog niet zeggen dat er in de uitvoer ook een lege regel komt.

Het hangt van de document class af, hoe de alinea's gezet worden. In artikelen, rapporten en boeken worden alinea's weergegeven door het inspringen van de eerste regel. Net zoals in deze alinea. In brieven, of als men een definitie van de volgende vorm invoert,

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

worden alinea's door een verticale afstand, zonder inspringen, aangegeven. Net zoals in de rest van deze handleiding. Als je één enkele alinea zonder inspringen wilt beginnen kun je het commando `\noindent` aan het begin zetten.

2.3 Afbreken van regel en pagina's

2.3.1 Blokmode

Normale tekst wordt in blokmode gezet, d.w.z. met (linker en rechter) kantlijnen uitgelijnd. LaTeX breekt regels en pagina's automatisch af. Voor elke alinea wordt de best mogelijke verdeling van woorden en regels gezocht en worden, waar nodig, woorden automatisch afgebroken.

Het einde van een woord of regel kenmerkt zich door een spatie. Het speelt geen rol of men één of honderd spaties invoert.
Eén of meer lege regels kenmerken het eind van een alinea.

```
Het einde van een woord of
regel kenmerkt zich door
een spatie.
Het speelt geen rol
of men \'{e}\'{e}n of       honderd
spaties invoert.


E\'{e}n of meer lege regels
kenmerkt het eind van een
alinea.
```

Met behulp van de in paragraaf 2.9 beschreven 'environments' (omgevingen) is het mogelijk om speciale stukken tekst anders te zetten.

Voor uitzonderingsgevallen kan men het afbreken met de volgende commando's beïnvloeden: Het commando `\\` of `\newline` breekt de regel af zonder een nieuwe alinea te beginnen. Het commando `\\*` breekt een regel af, maar er mag op die plaats niet op een nieuwe pagina begonnen worden. Deze commando's kunnen niet aan het begin van een regel gegeven worden, dus ook niet om extra lege regels in te voeren. Wanneer men extra witruimte tussen de regels wil hebben kan dat als optionele parameter aan het commando `\\` gegeven worden, bijv. met het commando `\\[5mm]`. Omdat `\\` een optioneel argument heeft en een `*` vorm kan de volgende regel niet zonder meer met een `*` of `[` beginnen. Zet in dat geval accolades om dit teken heen. Met behulp van de commando's `\linebreak[`*n*`]`, `\nolinebreak[`*n*`]`, `\pagebreak[`*n*`]` en `\nopagebreak[`*n*`]` kan men de wenselijkheid van het afbreken van regel (pagina's) instellen, n geeft de mate van wenselijkheid weer (0, 1, 2, 3 of 4 met 0 = niet wenselijk, 4 = zeer wenselijk). Het commando `\newpage` kan gegeven worden om een nieuwe pagina te beginnen, `\clearpage` doet dit ook, maar drukt eerst alle nog niet afgedrukte figuren en tabellen af; `\cleardoublepage` doet hetzelfde, maar bij de `book` documentclass wordt daarna altijd op een rechterpagina begonnen.

LaTeX doet erg veel moeite om mooie regels te maken. Als geen van de methoden een mogelijkheid biedt om een gladde rechterkantlijn te produceren, wordt een foutmelding geproduceerd ('overfull hbox'). Dit treedt vooral op als er geen geschikte plek wordt gevonden om een woord af te breken. Als men het commando `\sloppy` geeft, is LaTeX gewoonlijk veel minder streng en vermijdt zulke lange regels door de woordafstand te vergroten (dit ziet er minder mooi uit). Het effect van `\sloppy` kan weer ongedaan maken met `\fussy`. Er worden in dit geval ook foutmeldingen geproduceerd ('underfull hbox'), maar het resultaat is meestal goed bruikbaar.

2.4 Speciale Tekens

De volgende symbolen hebben een speciale betekenis voor LaTeX, zij kunnen niet zomaar gebruikt worden.

`$ & % # _ { } ~ ^ " \ | < >`

De volgende zeven tekens kunnen door er een `\` (backslash) voor te zetten gewoon afgedrukt worden.

$ & % # _ { }	`\$ \& \% \# \_ \{ \}`

Als aanhalingsteken wordt *niet*, het op typemachines gebruikelijke, (`"`) gebruikt. In boeken worden voor aanhalingsteken openen en aanhalingsteken sluiten verschillende tekens gebruikt. Voor aanhalingstekens openen worden er twee ``` `` ``` gebruikt, voor aanhalingstekens sluiten gebruikt men `''`. Tegenwoordig wordt ook vaak met enkele aanhalingstekens gewerkt ‘zoals dit’.

“Nee,” zei hij, “ik weet het niet!”	``` ``Nee,'' zei hij, ``` ``` ``ik weet het niet!'' ```

Voor de ouderwetse Nederlandse „aanhalingstekens” zie tabel 2.3.

De overige symbolen en andere bijzonder tekens kunnen, zoals we nog zullen zien, met speciale commando’s als accenten, of in mathematische formules afgedrukt worden.

2.4.1 Gedachte- en verbindingsstreepjes

In boeken worden er streepjes van diverse lengtes gebruikt om een gedachtestreepje, verbindingsstreepje of min-teken mee aan te geven. Deze verschillende lengtes worden in LaTeX door kombinaties van min-tekens aangegeven:

O-benen	`O-benen \\`
10–18 uur	`10--18~uur \\`
ja – of nee?	`ja -- of nee? \\`
yes—or no?	`yes---or no? \\`
0, 1 en -1	`0, 1 en $-1$`

2.4.2 Punten (dots)

In tegenstelling tot typemachines, waar elke punt en komma evenveel ruimte in beslag neemt als een gewone letter, worden in boeken punten en komma’s zo dicht mogelijk tegen de vorige letter aangezet (‘kerning’). Als ‘voortzettingspunten’ (drie punten met een speciale afstand) is er het commando `\ldots`.

Niet zo ... maar zo:	`Niet zo ... maar zo: \\`
Haarlem, Amsterdam, . . .	`Haarlem, Amsterdam, \ldots`

Tabel 2.1: Accenten en speciale letters

input	output	input	output
\`o	ò	\'o	ó
\^o	ô	\~o	õ
\=o	ō	\.o	ȯ
\u o	ŏ	\v o	ǒ
\H o	ő	\"o	ö
\c o	o̧	\d o	ọ
\b o	o̱	\t oo	o͡o
\oe	œ	\OE	Œ
\ae	æ	\AE	Æ
\aa	å	\AA	Å
\o	ø	\O	Ø
\l	ł	\L	Ł
\i	ı	\j	ȷ
!`	¡	?`	¿

2.4.3 Ligaturen

In boeken is het gebruikelijk om bepaalde letterkombinaties anders te zetten dan de enkele karakters.

ﬀ ﬁ ﬂ AV Te… in plaats van ff fi fl AV Te…

Wanneer er een nieuw teken gebruikt wordt i.p.v. de combinatie dan wordt dit een *ligatuur* genoemd. Wanneer alleen maar de afstand tussen de letters verandert spreken we over *kerning*. Beide verschijnselen kunnen voorkomen worden door tussen de twee letters het commando `\/` te zetten.

Niet eﬀectief	`Niet effectief \\`
maar effectief	`maar ef\/fectief`

2.4.4 Accenten en speciale letters

Met LaTeX is het mogelijk om accenten en speciale karakters uit een grote verscheidenheid van talen te gebruiken (zie tabel 2.1). In voorbeelden wordt meestal de letter o gebruikt, maar het is, in principe, mogelijk om op elke letter een accent te zetten. Als er een accent op een i of een j gezet moet worden, dan moet er een 'puntloze'-i (of -j) gebruikt worden. Een dergelijke ı (of ȷ) wordt met het commando `\i` (of `\j`) te voorschijn getoverd.

Hôtel, naïve, smørebrød.
¡Señorita!
	`H\^otel, na\"\i ve,`
	`sm\o rebr\o d. \\`
	`!‘Se\~norita!`

2.5 Afstanden

Om een mooie gladde rechterkantlijn te krijgen varieert LaTeX de ruimte tussen de woorden. Na punten, vraagtekens en andere tekens die een zin beëindigen wordt een ietwat grotere afstand gebruikt, wat de leesbaarheid van de tekst vergroot. LaTeX neemt aan, dat een hoofdletter gevolgd door een punt een afkorting is en dat alle andere punten een zin beeindigen.

Uitzonderingen op deze regel kan men LaTeX met de volgende commando's meedelen: Een 'backslash' voor een spatie betekent dat deze spatie niet breder gemaakt mag worden. Een ~ (tilde) is een spatie die niet breder mag worden gemaakt en waarop de regel niet mag worden afgebroken. De tilde wordt vaak gebruikt om dingen bij elkaar te houden als het lelijk staat om de regel ertussen af te breken. Het commando `\@` voor een punt betekent, dat deze punt een regel beëindigt, hoewel er een hoofdletter voor de punt staat.

Dr. Partl woont op de 1^{e} verdieping.
… 5 m breed.
Ik gebruik vitamine C. Jij niet?
	`Dr.~Partl woont op de 1$^{e}$~verdieping. \\`
	`\ldots\ 5~m breed. \\`
	`Ik gebruik vitamine~C\@.`
	`Jij niet?`

Behalve dat is er ook nog de mogelijkheid om met het commando

```
\frenchspacing
```

te bewerkstelligen de afstanden tussen woorden en het einde van zinnen hetzelfde te behandelen. Dit is vooral bij de niet-angelsaksische talen gebruikelijk. In dit geval hoeven de commando's `\ ` en `\@` niet meer gebruikt te worden. Dit commando wordt meestal in de *preamble* gegeven.

2.6 Hoofdstukken en paragrafen (section)

Het begin van een hoofdstuk of paragraaf wordt met het commando `\chapter{...}` e.d. aangegeven. Als parameter wordt de titel van het hoofdstuk, sectie etc. gegeven. In deze parameter mogen bepaalde commando's zoals `\ref` en `\sqrt` niet gebruikt worden, zie sectie 7.9. Er moet wel een logische structuur aangehouden worden. Er is geen apart commando voor appendices, maar als één keer het commando `\appendix` gegeven wordt, dan gelden de hoofdstukken, (secties in `article`) die daarna komen als appendices.

Bij artikelen:

```
\section  \subsection  \subsubsection
```

Bij rapporten en boeken:

```
\chapter  \section  \subsection  \subsubsection
```

Artikelen kunnen dan eenvoudig als hoofdstuk in een boek opgenomen worden.

De afstand tussen het kopje en de tekst, het nummeren en het lettertype worden door LaTeX zelf vastgesteld.

Een nog fijnere indeling geven de commando's `\paragraph` en `\subparagraph`. Deze worden niet in de inhoudsopgave opgenomen en de titel wordt in de lopende tekst gezet.

De titel (titelpagina) van het totale artikel wordt met behulp van het commando `\maketitle` gezet. Wat er in de titel (op de titelpagina) komt te staan moet dan wel eerst even met de commando's `\title`, `\author` etc. duidelijk gemaakt worden. Het `\maketitle` commando moet na de `\begin{document}` gegeven worden, de gegevens mogen ook eerder gezet worden.

```
\title{Handleiding \LaTeX}
\author{Piet van Oostrum}
date{20 april 1995} \maketitle
```

Tussen `\begin{abstract}` en `\end{abstract}` kan een samenvatting gegeven worden.

Het commando `\tableofcontents` zorgt ervoor dat er op die plaats de inhoudsopgave wordt afgedrukt. LaTeX gebruikt de kopjes en nummering van de *vorige* 'run'. Als er tekst wordt tussengevoegd of weggelaten, moet men LaTeX twee keer aanroepen om een juiste inhoudsopgave te verkrijgen.

Dan is er ook nog het commando `\chapter*{...}`, dat ervoor zorgt dat het hoofdstuk *niet* genummerd wordt en dat het niet in de inhoudsopgave verschijnt. Soms is het nuttig om in de inhoudsopgave een iets andere tekst op te nemen dan in het kopje, bijvoorbeeld wanneer de sectietitel te lang is of voetnoten bevat. In dat geval kan de versie voor de inhoudsopgave tussen vierkante haken achter het commando, maar voor de titel gegeven worden, zoals

```
\section[Kort]{Een lange sectietitel}
```

Dit is gedaan bij sectie 2.9.3.

Met het commando `\label` en `\ref` is het mogelijk om de door LaTeX gegenereerde hoofdstuknummers in de tekst te gebruiken. LaTeX vervangt `\ref{...}` door het in de tekst met `\label{...}` gedefiniëerde nummer. Het `\label{...}` commando moet direct na het sectie-commando gegeven worden. Als parameter neem je een zelf gekozen naam. Ook hier wordt het nummer van de laatste 'run' gebruikt. Voorbeeld:

```
\chapter{Algoritmes}
...
Het bewijs daarvoor is in hoofdstuk~\ref{hfst:bew} gevonden.
...
\chapter{Bewijzen} \label{hfst:bew}
...
```

2.7 Voetnoten (footnote)

Voetnoten[1] worden automatisch genummerd en onderaan de bladzijde geplaatst. Om een mooie layout te krijgen moet het `\footnote` strak tegen het voorgaande woord aangezet worden.

```
Voetnoten\footnote
{Dit is een voetnoot.}
worden automatisch ...
```

2.8 Benadrukte woorden (emphasize)

Met de typemachine worden benadrukte teksten onderstreept, in boeken worden daarvoor verschillende lettertypes gebruikt. Het commando `\emph` (emphasize) zet het 'benadrukken' aan.

In gedrukte tekst gebruikt men *geen* onderstrepingen.

```
In gedrukte tekst gebruikt men
\emph{geen} onderstrepingen.
```

LaTeX gebruikt voor de te benadrukken tekst *schuin* schrift.

Wordt er binnen benadrukte tekst nog een keer *woorden benadrukt, dan gebruikt LaTeX daarvoor weer* recht *schrift.*

2.9 Environments

Speciale stukken tekst, die anders gezet dienen te worden dan in normale blokmode, worden gekenmerkt door zg. 'environments' (omgevingen). Environments hebben de volgende vorm.

`\begin{`*name*`}` *tekst* `\end{`*name*`}`

Zij kunnen ook 'genest' worden, daarbij moet men wel de goede volgorde aanhouden.

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

2.9.1 Citaten (quote, quotation, verse)

De `quote`-environment wordt gebruikt voor korte citaten, benadrukte zinnen en voorbeelden, de tekst springt links en rechts in. Bij de `quote` omgeving geeft een lege regel een nieuw citaat aan, er wordt dan witruimte gereserveerd.

[1] Dit is een voetnoot.

Een typografische vuistregel voor de regellengte luidt:

> Geen enkele regel mag meer dan 75 letters bevatten.

Daarom worden in kranten meerdere kolommen naast elkaar gebruikt.

```
Een typografische vuistregel
voor de regellengte luidt:
\begin{quote}
Geen enkele regel mag meer
dan 75~letters bevatten.
\end{quote}
Daarom worden in kranten
meerdere kolommen naast
elkaar gebruikt.
```

De `quotation`-environment wordt gebruikt voor citaten die uit meerdere alinea's bestaan.

De `verse`-environment wordt gebruikt voor gedichten en voorbeelden waarbij de regelopbouw belangrijk is. De regels worden door `\\` gescheiden, strofen door lege regels. Regels die te lang zijn worden afgebroken en de vervolgregels worden ingesprongen zoals bij gedichten gebruikelijk is.

2.9.2 Lijsten (itemize, enumerate, description)

De environment `itemize` wordt gebruikt voor eenvoudige lijsten (zie fig. 2.1). De environment `enumerate` wordt gebruikt voor lijsten met genummerde regels (zie fig. 2.2). De environment `description` wordt gebruikt voor lijsten met een beschrijvend kopje (zie fig. 2.3).

Elk onderdeel van de lijst begint met het commando `\item`. Dit heeft een optioneel argument waarmee het standaard merkteken (bolletje, nummer etc.) gewijzigd kan worden. Bij de `description` omgeving is dit argument verplicht omdat daarvoor geen standaard gedefinieerd is. Omdat `\item` een optioneel argument heeft kan de volgende regel niet zonder meer met een `[` beginnen. Zet in dat geval accolades om dit teken heen.

2.9.3 Linksuitvullen, rechtsuitvullen, centreren (flushleft, flushright, center)

De environment `flushleft` en `flushright` vullen een zin links- dan wel rechts uit. Er onstaat dus een onregelmatige kantlijn, `center` zet de tekst in het midden van de regel. De losse zinnen worden door `\\` gescheiden. Als men geen gebruik maakt van `\\` dan breekt LaTeX de zinnen automatisch af.

links
Bakboord

```
\begin{flushleft}
links \\
Bakboord
\end{flushleft}
```

rechts
Stuurboord

```
\begin{flushright}
rechts  \\
Stuurboord
\end{flushright}
```

Lijsten:

- In de `itemize`-environment worden de elementen door punten en andere symbolen gemerkt.
- Lijsten kunnen ook 'genest' worden:
  - Er kan tot maximaal 4 nivo's diep genest worden.
  - Inspringen en het merken van de elementen gebeurt automatisch.
- enz.

```
Lijsten:
\begin{itemize}

\item In de \texttt{itemize}-environment .....

\item Lijsten kunnen ook
'genest' worden:
  \begin{itemize}
  \item Er kan tot .....
  \item Inspringen en .....
  \end{itemize}

\item enz.

\end{itemize}
```

Figuur 2.1: Een `itemize` voorbeeld

Mooi
In
Het
Midden

```
\begin{center}
Mooi \\ In \\ Het \\ Midden
\end{center}
```

Binnen een andere omgeving (zoals `quote`) kan hetzelfde effect bereikt worden met een z.g. *declaratie*.

omgeving	*declaratie*
`center`	`\centering`
`flushleft`	`\raggedright`
`flushright`	`\raggedleft`

Deze declaraties hebben geen effect buiten de omgeving waarin ze gebruikt worden.

2.10 Afbreken van woorden

Als de automatische woordafbreekroutines in enkele gevallen niet het juiste resultaat opleveren, kan men LaTeX met de volgende commando's terechtwijzen. Bij samengestelde of buitenlandse woorden kan dit zelfs noodzakelijk zijn.

Het commando `\hyphenation` zorgt ervoor, dat de daar genoemde woorden alleen maar op de met '`-`' gemarkeerde plekken afgebroken kunnen worden. Zo'n lijst moet aan het begin van de inputfile staan en mag geen cijfers of speciale tekens bevatten. Voorbeeld:

Nummeren:

1. In de `enumerate`-environment worden de elementen met een cijfer of letter genummerd.
2. Het nummeren gebeurt automatisch.
3. Lijsten kunnen ‘genest’ worden.
   (a) Er kan tot 4 nivo’s genest worden.
   (b) Het inspringen en de nummering gebeurt automatisch.
   (c) Er kan naar de nummers gerefereerd worden met de `\label` en `\ref` commando’s.
4. enz..

```
Nummeren:
\begin{enumerate}

\item In de {\tt enumerate}-environment
worden ....

\item Het nummeren gebeurt ....

\item Lijsten kunnen
'genest' worden.
  \begin{enumerate}
  \item Er kan tot ....
  \item Het inspringen ....
  \item Er kan naar de ....
\end{enumerate}

\item enz.

\end{enumerate}
```

Figuur 2.2: Een `enumerate` voorbeeld

Dierkunde:

Gnat: Een klein dier, dat leeft in het noorden en geen moeilijkheden veroorzaakt.

Gnoe: Een groot dier, dat voornamelijk in kruiswoordpuzzels voorkomt.

Gordeldier: Een middelmaat dier, dat hier alleen maar vanwege de lengte van zijn naam voorkomt.

```
Dierkunde:
\begin{description}
\item[Gnat:]
   Een klein dier ...
\item[Gnoe:]
   Een groot dier ...
\item[Gordeldier:]
   Een middelmaat dier ...
\end{description}
```

Figuur 2.3: Een `description` voorbeeld

```
\hyphenation{ge-woon-lijk bij-zon-der
   uit-zon-der-lijk}
```

Het commando `\-` zorgt ervoor, dat het woord voor deze ene keer op de met `\-` gemarkeerde plek afgebroken mag worden. Dit commando geldt voor alle tekst, dus *ook* woorden met cijfers, en/of symbolen.

gewoonlijk, LaTeX-input-file, überhaupt	`ge\-woon\-lijk,`
	`\LaTeX-input\-file,`
	`\"uber\-haupt`

Voor speciale gevallen is er ook nog het commando `\discretionary`. Dit komt voornamelijk van pas bij 'vreemde' talen, zoals duits, waar andere afbreekregels gelden. Er zijn echter ook in het nederlands woorden waar speciale afbreekregels voor gelden. De eerste twee parameters laten het tekstgedeelte zien zoals het woord afgebroken dient te worden. De derde parameter laat het tekstgedeelte zien als het woord niet wordt afgebroken. Als je zo'n woord veel gebruikt is het handiger er een eigen commando voor te maken.

Drucker oder Druk-ker. Papaatje of papa-tje. Cafeetje of café-tje	`Dru\discretionary{k-}{k}{ck}er`
	`pap\discretionary{a-}{}{aa}tje`
	`caf\discretionary{\'e-}{}{ee}tje`

Het commando `\mbox{...}` zorgt ervoor, dat het woord helemaal niet afgebroken wordt.

Het telefoonnummer is nu (0222) 5601-3694.
De parameter `filename` geeft de filenaam aan.
	`Het telefoonnummer is nu`
	`\mbox{(0222) 5601-3694}. \\`
	`De parameter \mbox{\tt filename}`
	`geeft de file\-naam aan.`

2.11 Nederlandse teksten

In deze paragraaf worden de verschillen behandeld tussen de orginele (amerikaanse) LaTeX en de nederlandse (europese) LaTeX. De originele versie ondersteunt alleen maar het zetten van engelse tekst. De 'europese' versie ondersteunt in ieder geval meer talen. Elke taal heeft zijn eigen typografische eigenaardigheden. We zullen hier echter alleen maar de nederlandse eigenaardigheden behandelen.

Bij het uitwisselen van LaTeX-files, moet men er rekening mee houden dat op de andere computer niet alle pakketten en/of commando's ter beschikking staan. Het kan dus nodig zijn een aantal speciale files mee te sturen.

2.11.1 Afbreekroutines

Bij het opstarten van LaTeX wordt vastgesteld welke afbreekroutines er gebruikt gaan worden. Als we niets doen zal dit i.h.a. het Engelse systeem zijn, wat voor Nederlandse woorden een uitgerekend slecht resultaat geeft. Door middel van het `babel` pakket kan een ander afbreeksysteem gekozen worden, bijvoorbeeld door in de *preamble* het volgende commando op te nemen:

```
\usepackage[dutch]{babel}
```

Met het commando

```
\selectlanguage{english}
```

kan tijdelijk weer op Engels overgegaan worden en met

```
\selectlanguage{dutch}
```

weer teruggegaan worden.

2.11.2 Namen

Bij gebruik van `babel` worden hoofdstukken, figuren, tabellen enz. goed benoemd (Hoofdstuk i.p.v. Chapter, etc.). In tabel 2.2 staat welke namen veranderd worden en ook welk commando gebruikt wordt door LaTeX. Als een van de namen je niet aanstaat dan kun je het veranderen met een commando als:

```
\renewcommand\contentsname{Inhoud}
```

2.11.3 Trema's

Tenslotte wordt bij gebruik van dit pakket het teken `"` speciaal gemaakt. Het heeft dan de betekenis van een trema als het gevolgd wordt door een letter `aeou`, maar dit trema zal verdwijnen als de regel ervoor wordt afgebroken. De combinatie `"i` geeft een trema op een puntloze ı. In tabel 2.3 staan nog een paar speciale gebruiksmogelijkheden van het teken `"` met de `dutch` optie.

2.12 Layout

In Europa zijn andere layouts en conventies gebruikelijk dan in Amerika. Ook de papierformaten zijn in Europa anders dan in Amerika. Daarom zijn er een aantal opties en layouts die ervoor zorgen dat LaTeX (in ieder geval) aan de nederlandse conventies voldoet. Met het commando

```
\documentclass[a4paper]{artikel1}
```

wordt LaTeX duidelijk gemaakt dat we op een A4-tje willen afdrukken en dat we van de nederlandse artikel-class gebruik willen maken. Andere nederlandse stijlen zijn: artikel2, artikel3, rapport1, rapport3, boek en brief.

De optie `a4paper` kan ook met andere documentclasses gebruikt worden. Het heeft tot gevolg dat dezelfde pagina afmeting gebruikt wordt, maar dat deze op A4 papier gepositioneerd wordt. De standaard pagina afmeting van LaTeX gebruikt het papier niet erg efficient omdat het afgestemd is op het formaat van boeken en tijdschriften. Wil je het papier beter benutten dan kan i.p.v. de `a4paper` optie het pakket `a4` of `a4wide` gebruikt worden. Het laatste gebruikt nog meer van het papieroppervlak maar bij een klein lettertype worden de regels te lang voor een typografisch verantwoord resultaat, tenzij men met meer kolommen gaat werken.

2.13 Letterlijke weergave (verbatim, verb)

De tussen `\begin{verbatim}` en `\end{verbatim}` staande zinnen, worden precies zo gedrukt zoals zij zijn ingevoerd, maar dan in het 'typewriter' lettertype. Dat betekent, dat alle spaties, einde van zinnen en alle speciale tekens zonder enige bemoeienis van LaTeX afgedrukt worden. Bv. een (kort) computer-programma laat zich goed in deze environment zetten.

Binnen een zin kunnen korte stukken tekst ook 'letterlijk' worden afgedrukt. Een en ander gebeurt door de betreffende tekst tussen `\verb|` en `|` in te sluiten. In plaats van `|` mag ook een ander teken gebruikt worden. Met dit commando zijn alle voorgaande voorbeelden gezet.

Het `\ldots`-commando …	`Het \verb|\ldots|-commando \ldots`

Tabel 2.2: Onderdeelnamen

`\contentsname`	Contents	Inhoudsopgave
`\listfigurename`	List of Figures	Lijst van figuren
`\listtablename`	List of Tables	Lijst van tabellen
`\abstractname`	Abstract	Samenvatting
`\refname`	References	Referenties
`\bibname`	Bibliography	Bibliografie
`\indexname`	Index	Index
`\figurename`	Figure	Figuur
`\tablename`	Table	Tabel
`\partname`	Part	Deel
`\chaptername`	Chapter	Hoofdstuk
`\appendixname`	Appendix	Bijlage
`\pagename`	Page	Pagina
`\prefacename`	Preface	Voorwoord

`"‘abc"’`	„abc”
`"-`	afbreekbaar - teken
`"|`	voorkom ligatuur
`"y`	ij
`"Y`	IJ

Tabel 2.3: Speciaal gebruik van `"`

De `verbatim`-environment en het `\verb` commando mogen *niet* binnen de parameters van andere commando's en ook niet binnen het definitiegedeelte van een eigen commando gebruikt worden.

2.14 Lettergroottes en -types (Fonts)

Normaal gesproken kiest LaTeX de lettergrootte en het lettertype op grond van de commando's, die de logische structuur van de tekst aangeven. In speciale gevallen kan men ook *zelf* het lettertype of de lettergrootte kiezen, zie tabel 2.4 op pagina 31 en tabel 2.5 op pagina 32. De commando's in tabel 2.4 komen in twee smaken: In de eerste kolom staan commando's die de te veranderen tekst als argument hebben; het gebruik van deze commando's is analoog aan het gebruik van het `\emph` commando. In de tweede kolom staan declaraties; deze hebben geen argument maar zorgen dat alle volgende tekst in het betreffende lettertype gezet worden. Dit geldt ook voor de commando's in tabel 2.5. Het effect van een declaratie beperkt zich tot een *groep*: dit is een omgeving, een formule, een element in een tabel of een stuk LaTeX invoer tussen accolades `{}` (maar niet de accolades die gebruikt worden om argumenten van een commando aan te geven). Zie het volgende voorbeeld. De commando's uit tabel 2.4 kunnen gecombineerd worden, bijv. één *family* met één *shape*. De declaraties uit tabel 2.5 veranderen ook de regelafstand *als er een alineaovergang (lege regel) in hun bereik ligt.*

Het `\emph` commando heeft ook een bijbehorende declaratie, nl. `\em`. Tenslotte is het mogelijk om de naam van een declaratie zonder de `\` als omgeving te gebruiken, bijvoorbeeld `\begin{large}` ... `\end{large}`. Voor grote hoeveelheden tekst maakt dit de structuur duidelijker.

Deze kleine **vette** Romeinen regeerden het grote *Italië.*

le 2$^{\text{ème}}$ régime

```
{\small Deze kleine \textbf{vette}
Romeinen regeerden} {\Large het grote
\textit{Itali"e}.}
\\
le $2^{\textrm{\scriptsize\`eme}}$
r\'egime
```

Hoe *minder* verschillende lettertypes men gebruikt, hoe leesbaarder het document wordt.

2.15 Modes

LaTeX heeft verschillende z.g. *modes* waarin het verwerken van tekst op andere manieren plaatsvindt. De belangrijkste mode is de *paragraph mode* waarin tekst gewoon in alinea's gezet wordt. Regels worden zoveel mogelijk opgevuld (tenzij aangegeven wordt dat een nieuwe regel of alinea moet beginnen), ze worden afgebroken en regels worden onder elkaar gezet. De beweging is dus van links naar rechts en van boven naar beneden. De tweede mode is de *LR-mode* waarin alleen van links naar rechts gegaan wordt en geen regels afgebroken worden. We hebben al het `\mbox` commando gezien waarmee deze mode geselecteerd wordt. Ook de inhoud van tabellen wordt in *LR-mode* gezet (hoofdstuk 3). Het commando `\fbox` maakt ook een box in LR-mode, maar zet er bovendien een kader omheen. Omdat de inhoud in LR-mode is, kan dit dus niet gebruikt worden om een kader om een alinea te zetten. Tenslotte hebben we *math-mode* waarin formules gezet worden (zie hoofdstuk 4).

Het commando `\parbox` en de `minipage` omgeving creëren een *paragraph mode*. Op deze manier kan binnen een andere mode een stuk tekst in paragraph mode opgenomen worden. Beide hebben een optioneel argument dat aangeeft hoe deze tekst binnen de omliggende tekst opgenomen moet worden: `[t]` wanneer de *top* regel op dezelfde hoogste als de omliggende tekst moet worden gezet, `[b]` wanneer dit de benedenste regel moet zijn. Als dit argument weggelaten wordt dan wordt het midden van de box genomen. Als verplicht argument bij beiden moet de breedte van de tekst opgegeven worden. De eigenlijke tekst is bij `\parbox` het volgende argument, bij `minipage` is het de inhoud van de omgeving. Voetnoten in een `minipage` omgeving worden onder aan de minipage gezet, niet onderaan de pagina.

Een
kleine
par-
box – en – een iets
bredere
minipage

```
\parbox[b]{1cm}{Een kleine parbox}
-- en --
\begin{minipage}[t]{2cm}
een iets bredere minipage
\end{minipage}
```

Tabel 2.4: Lettertypes

`\textrm`	`\rmfamily`	normaal schrift (roman)
`\textsf`	`\sffamily`	'sans serif' schrift
`\texttt`	`\ttfamily`	`schrijfmachineschrift (typewriter)`
`\textbf`	`\bfseries`	**vet schrift (boldface)**
`\textup`	`\upshape`	rechtopstaand schrift
`\textit`	`\itshape`	*cursief schrift (italic)*
`\textsl`	`\slshape`	*schuin schrift (slanted)*
`\textsc`	`\scshape`	'Caps and Small Caps' Schrift

Tabel 2.5: Lettergroottes

`\tiny`	vreselijk klein schrift
`\scriptsize`	heel klein schrift (indices)
`\footnotesize`	klein schrift (voetnoten)
`\small`	klein schrift
`\normalsize`	normaal schrift
`\large`	groot schrift
`\Large`	groter schrift
`\LARGE`	heel groot schrift
`\huge`	reuzegroot
`\Huge`	gigantisch groot

Hoofdstuk 3

Tabellen

Tabellen zijn meestal de moeilijkste onderdelen van een document. Op een typemachine of tekstverwerker worden tabellen meestal gemaakt met z.g. *tabstops* waarnaar men kan springen met de TAB-toets. De plaatsen van de tabstops moeten dan experimenteel vastgesteld worden. LaTeX heeft iets soortgelijks met de `tabbing` omgeving. Hiermee kunnen meestal alleen eenvoudige tabellen gemaakt worden. Als er in een kolom delen moeten voorkomen die meer dan één regel beslaan dan kan dit bij de meeste tekstverwerkers niet, en bij LaTeX moeilijk. Ook tabellen binnen andere tabellen zijn op deze manier niet of moeilijk te doen. LaTeX heeft hiervoor een andere omgeving, `tabular`, waarmee dit wel kan.

3.1 Tabulatoren (tabbing)

In de `tabbing`-environment kan men tabulatorstops net als op de typemachine instellen en gebruiken. Het commando `\=` zet een tabulator-positie. Dit gebeurt meestal in de eerste regel van de tabel, maar kan ook in volgende regels gebeuren, o.a. wanneer de tab-stops onderweg veranderd moeten worden. Soms komt het voor dat de tabstops bepaald worden door tekst die niet in de eerste regel voorkomt. In dat geval kunnen de tabstops gezet worden in een voorbeeldregel die dan met het commando `\kill` weer teniet gedaan wordt (dus niet afgedrukt). Extra ruimte kan gegenereerd worden met het `\hspace` commando met de lengte als parameter, bijv. `\hspace{3cm}`. Aan het begin van de regel wordt ruimte door `\hspace` gegenereerd weggegooid, dan moet men het commando `hspace*` gebruiken. In tabel 3.1 staan de eenheden die gebruikt kunnen worden. Het commando `\quad` is een afkorting van `\hspace{1em}`.

Het commando `\>` springt naar de volgende tabulator-positie en `\\` breekt de regel af. Een verschil tussen `\>` in LaTeX en een tabstop op een typemachine, is dat LaTeX altijd naar de volgende tabstop springt m.b.t. het nummer ervan, en niet afhankelijk van de plaats waar de vorige tekst eindigt. Dit heeft het voordeel dat de tekst niet per ongeluk in de verkeerde kolom terecht komt als een kolom te smal is, maar kan tot gevolg hebben dat tekst over elkaar heen afgedrukt wordt.

Tabel 3.1: Eenheden voor lengtes

`pt`	point	(1 in = 72.27 pt)
`pc`	pica	(1 pc = 12 pt)
`in`	inch	(1 in = 25.4 mm)
`bp`	big point	(1 in = 72 bp)
`cm`	centimeter	(1 cm = 10 mm)
`mm`	millimeter	(1 mm = 0.01 m)
`dd`	didot point	(1157 dd = 1238 pt)
`cc`	cicero	(1 cc = 12 dd)
`sp`	scaled point	(65536 sp = 1 pt)
`ex`	x-hoogte	hoogte van letter x
`em`	M-breedte	breedte van letter M

links	middenstuk	rechts
Er		
was eens	maar nu	niet meer,
een,		gerepareerde
		teddybeer

```
\begin{tabbing}
was eens\quad \=
 middenstuk\quad \= \kill
links \> middenstuk \> rechts\\
Er \\
was eens \> maar nu
 \> niet meer\\
een  \>  \> gerepareerde\\
 \>  \> teddybeer
\end{tabbing}
```

Minder gebruikte commando's in de `tabbing` omgeving zijn:

`\+` zorgt ervoor dat alle volgende regels één extra tabstop inspringen (er hoeft dan dus één `\>` minder gegeven te worden).

`\-` doet het effect van één `\+` weer teniet voor de volgende regels.

`\<` annuleert het effect van één `\+` alleen voor de regel waar het in staat. Dit kan alleen aan het begin van een regel (regel betekent hier een regel van de tabel, niet van de invoer).

`\'` zorgt ervoor dat de voorafgaande tekst in de vorige kolom rechts aangeschoven wordt (met tussenruimte die bepaald wordt door de waarde van `\tabbingsep`)

`` \` `` schuift de rest van de regel naar de rechterkantlijn; de tekst moet afgesloten worden met `\\`.

Omdat de accentcommando's `\=`, `\'` en `` \` `` in de `tabbing` omgeving een bijzondere betekenis hebben kunnen ze dus niet als accenten gebruikt worden. Je moet in plaats daarvan dan `\a=`, `\a'` en `` \a` `` gebruiken.

```
Als het vriest
    en het regent
        dan hebben we ijzel
   en   we glijden uit
    of het sneeuwt
        dan wordt alles wit
anders is alles gewoon
```

```
\begin{tabbing}
Als \= het vriest \+\\
       en \= het regent \+\\
             dan hebben we ijzel \\
             en \' we glijden uit \\
       \< of \> het sneeuwt \\
             dan wordt alles wit\-\-\\
anders is alles gewoon
\end{tabbing}
```

De `tabbing` omgeving kan binnen verschillende omgevingen niet gebruikt worden, i.h.b. niet binnen een andere `tabbing`, tenzij de binnenste in een `minipage` omgeving staat.

3.2 Automatische tabellen (tabular)

De `tabular`-environment is voor het zetten van tabellen, waarbij LaTeX automatisch de benodigde kolombreedtes berekent en waarin ook uitvullen en hulplijnen toepasbaar zijn.

In de parameterlijst van het commando `\begin{tabular}{...}` word het formaat van de tabel opgegeven. Een `l` duidt een kolom aan met links uitgevulde tekst, een `r` een kolom met rechtsuitgevulde tekst een `c` een kolom met gecentreerde tekst, met `p{`*breedte*`}` wordt een kolom aangeduid van de aangegeven breedte waarin meerregelige tekst kan staan (in feite wordt dan een `\parbox` gemaakt), een `|` betekent een verticale lijn. Als een bepaalde kolomspecificatie een aantal keer achter elkaar voorkomt dan kan dat afgekort worden met `*{`*aantal*`}{`*spec*`}`.

Binnen de tabel betekent `&` spring naar de volgende kolom, `\\` breek de regel af, `\hline` (op een alleenstaande regel) trekt een horizontale lijn. De laatste regel moet geen `\\` hebben behalve als er nog een horizontale lijn achter komt.

7C0	hex	Zo kan tekst toegevoegd worden
3700	octaal	
11111000000	binair	
1984	decimaal	OK

```
\begin{tabular}[t]{|rl|p{1.5cm}|}
\hline
7C0 & hex &
    Zo kan tekst toegevoegd worden\\
3700 & octaal & \\
11111000000 & binair & \\
\hline
1984 & decimaal & OK \\
\hline
\end{tabular}
```

Het is ook mogelijk horizontale lijnen op te nemen die niet over de hele breedte gaan met het `\cline` commando: `\cline{1-1}` geeft een lijn in kolom 1, `\cline{2-4}` geeft een lijn in kolom 2 t/m 4, Voor koppen is het soms nodig de standaard layout tijdelijk overboord te zetten. Dit kan met het commando `\multicolumn` met als eerste argument hoeveel kolommen tot één cel samengenomen worden, als tweede de layout van deze cel (`l`, `r` of `c` met eventueel `|`) en als derde de inhoud van de cel.

Een `tabular` omgeving geldt als een eenheid die temidden van andere tekst geplaatst kan worden. Deze kan dus in een regel tekst opgenomen worden zoals

dit	tab
mini	je

. Er is een optionele parameter die aangeeft welke regel van de `tabular` als referentie genomen wordt: `[t]` voor de top regel, `[b]` voor de onderste regel, en als niets opgegeven wordt dan wordt het midden van de `tabular` op dezelfde hoogte als de rest van de regel gezet. Deze extra parameters zijn ook handig als twee `tabular`'s naast elkaar gezet worden. Een tabular die op zichzelf moet staan kan het beste omgeven worden door alineascheiders (lege regels), eventueel met het commando `\noindent` ervoor. Ook een goede methode is er een `quote` of `center` omgeving omheen te zetten.

Item		*Prijs*
gnat	(dozijn)	3.24
gnoe	(per stuk)	24,985.47

```
\begin{tabular}[t]{llr}
\hline
\multicolumn{2}{c|}{\em Item} &
   \multicolumn{1}{c}{\em Prijs} \\
\hline
gnat    & (dozijn)      & 3.24      \\
gnoe    & (per stuk)    & 24,985.47 \\
\hline
\end{tabular}
```

Hier volgt een overzicht van de belangrijkste verschillen tussen `tabbing` en `tabular`

- In een `tabbing` omgeving kan een pagina overgang plaatsvinden, in een `tabular` omgeving niet[1].
- Een `tabbing` omgeving levert complete regels af, een `tabular` omgeving levert een "box" die ergens anders ingezet kan worden.
- `tabular` kan binnen een iedere andere omgeving gebruikt worden, in het bijzonder kan binnen een `tabular` weer een andere `tabular` gebruikt worden. Een `tabbing` omgeving is in dit opzicht beperkt.
- Bij een `tabbing` omgeving moet je zelf de kolomposities bepalen en kunnen deze onderweg veranderd worden, bij een `tabular` omgeving bepaalt LaTeX de posities en zijn ze voor de hele tabel hetzelfde.

3.3 Drijvende tabellen (table)

Tabellen worden meestal als eenheid in een document opgenomen, d.w.z. dat het ongewenst is om de tabel te laten onderbreken door een pagina overgang. Dat kan tot gevolg hebben dat een tabel niet meer op de pagina past en in zijn geheel naar de volgende pagina moet verhuizen. Er blijft dan een lege ruimte op de vorige pagina over, wat niet zo fraai is. Daarvoor heeft LaTeX "drijvende tabellen" waarbij LaTeX zelf uitzoekt waar een geschikte plaats is voor de tabel en de pagina verder met de rest van de tekst opvult. Bovendien kan een onderschrift

[1] Als dit wel gewenst is dan kan het pakket `longtable` of `supertab` gebruikt worden.

gegeven worden, wordt de tabel genummerd, en kan deze in een lijst van tabellen opgenomen worden.

Drijvende Tabellen worden tussen `\begin{table}` en `\end{table}` gezet. Voor het samenstellen van een tabel wordt meestal de `tabbing`- of de `tabular`-environment gebruikt, maar men kan ook andere strukturen (B.v. `enumerate` of `description`) gebruiken of een willekeurige andere tekst. Fig. 3.1 laat een skelet zien van een dergelijk gebruik. De `table` omgeving heeft een optionele parameter waarmee de plaatsing gestuurd kan worden. Dit kan een rijtje letters zijn, die aangeven waar LaTeX mag proberen de tabel te plaatsen: `h` voor *hier*, `t` voor de top van een pagina, `b` voor de benedenkant van een pagina, en `p` voor een aparte pagina met alleen tabellen en figuren (zie 5.2). LaTeX heeft een verzameling parameters die het aantal tabellen op een pagina etc. beperken, zie hiervoor [1]. Als in de positieparameter een `!` opgenomen wordt dan trekt LaTeX zich hier niets van aan. Wanneer de positieparameter niet opgegeven wordt dan neemt LaTeX `[tbp]`. In tweekoloms uitvoer kan met de `table*` omgeving een tabel over de hele breedte van de pagina opgenomen worden.

De titel van een tabel wordt met het `\caption` commando gegeven, dit commando kan zowel boven de tabel als onder de tabel gegeven worden. LaTeX genereert automatisch een nummer voor de tabel. Dit nummer kan met het `\label` commando een naam gegeven worden, waarnaar met het `\ref` commando gerefereerd wordt. *Het* `\label` *commando moet* ***na of in*** *het* `\caption` *commando gegeven worden.* In de tekst van het `caption` commando mogen bepaalde commando's zoals `\ref` en `\sqrt` niet gebruikt worden, zie sectie 7.9.

Een lijst van tabellen kan verkregen worden met het `\listoftables` commando, analoog aan het `\tableofcontents` commando. Als in de lijst een andere tekst opgegeven moet worden dan in de tabel dan kan deze als (eerste) optioneel argument aan het `\caption` commando gegeven worden, net als bij de `section` commando's.

Het is mogelijk in een `table` omgeving meer tabellen op te nemen, bijv. door een aantal `tabular` omgevingen naast elkaar te zetten. Als hierbij elke tabel een eigen caption moet hebben dan kan dit door elke tabel met het bijbehorende `\caption` commando in een `minipage` omgeving op te nemen.

```
\begin{table}[!btp]
\caption{...} \label{...}
  \begin{center}
    \begin{tabular}{...}
    ...
    \end{tabular}
  \end{center}
\end{table}
```

Figuur 3.1: Opbouw van een `table`-omgeving

Hoofdstuk 4

Formules

4.1 Algemeen

Formules vormen een belangrijk onderdeel van LaTeX. Formules kunnen in lopende tekst opgenomen worden of als losstaande elementen. De invoer van een formule kan in deze gevallen hetzelfde zijn, maar de layout kan dan verschillen (omdat in een lopende tekst minder ruimte beschikbaar is).

Formules worden in de z.g. *math-mode* gezet, waarbij heel andere regels gelden dan in tekst.

1. Het zetten van formules is meestal een *tweedimensionaal* gebeuren.
2. Spaties en regel-einden hebben geen enkele betekenis, alle afstanden worden aan de hand van de logica van de mathematische formule vastgesteld en kunnen alleen met speciale commando's zoals `\,` of `\quad` beïnvloed worden.
3. Lege regels zijn niet toegestaan (mathematische formules moeten in één alinea staan). Om de invoer overzichtelijker te maken kan men regels met alleen `%` invoegen.
4. Alle 'losse' letters worden als variabelen beschouwd en als zodanig gezet (cursief met de bijbehorende afstand). Wil men ook nog normale tekst (recht schrift) gebruiken dan dient men deze tekst met speciale commando's (bijv. `\mathrm{...}`) zetten (zie tabel C.14).
5. Binnen *math* mode kunnen veel commando's en symbolen gebruikt worden die in gewone tekst niet beschikbaar zijn.

Formules binnen een alinea worden tussen `$` en `$` of tussen `\(` en `\)` of tussen `\begin{math}` en `\end{math}` gezet. De `$...$` methode heeft het nadeel dat een weggelaten `$` LaTeX makkelijker in de war stuurt, maar op sommige plaatsen (bijvoorbeeld in hoofdstuk-, sectie- en tabel titels) kunnen de andere niet gebruikt worden.

Los de vergelijking $ax^2 + bx + c$ op.

```
Los de vergelijking
$a x^2 + b x + c$ op.
```

Zijn a en b de rechthoekszijden en c de hypotenusa, dan geldt $c^2 = a^2 + b^2$ (Stelling van Pythagoras).

```
Zijn $a$ en $b$ de rechthoekszijden
en $c$ de hypotenusa,
dan geldt $c^{2}=a^{2}+b^{2}$
(Stelling van Pythagoras).
```

TeX spreekt men als $\tau\epsilon\chi$ uit.

100 m^2 nuttige oppervlakte

Met $\heartsuit$-elijke groeten

```
\TeX\ spreekt men als
 $\tau\epsilon\chi$ uit.\\[6pt]
100~m$^{2}$ nuttige oppervlakte\\[6pt]
Met $\heartsuit$-elijke
 groeten
```

Grotere formules of vergelijkingen kan men beter op een aparte regel zetten. Ze worden dan tussen `\[` en `\]` of tussen `\begin{displaymath}` en `\end{displaymath}` gezet. Als ze ook nog genummerd moeten worden dan worden ze tussen `\begin{equation}` en `\end{equation}` gezet. De formules worden gecentreerd in de breedte, tenzij de documentclass optie `fleqn` meegegeven wordt, en het nummer wordt rechts gezet, tenzij de optie `leqno` gegeven wordt (zie tabel 1.2 op pagina 12).

De vergelijking

$$ax^2 + bx + c$$

heeft als oplossing:

$$x_{12} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
De vergelijking
\[ a x^2 + b x + c \]
heeft als oplossing:
\[
x_{12} =
\frac{-b \pm \sqrt{b^2 -4ac}}{2a}
\]
```

Zijn a en b de rechthoekszijden en c de Hypotenusa, dan geldt

$$c = \sqrt{a^2 + b^2} \tag{4.1}$$

(Stelling van Pythagoras).

```
Zijn $a$ en $b$ de rechthoekszijden
en $c$ de Hypotenusa,
dan geldt
\begin{equation}
c = \sqrt{  a^{2}+b^{2}  }
\end{equation}
(Stelling van Pythagoras).
```

Met `\label` en `\ref` kan men aan de vergelijkingsnummers refereren. Het `\label` commando moet na de `\begin{equation}` gegeven worden.

$$\epsilon > 0 \tag{4.2}$$

Uit (4.2) volgt ...

```
\begin{equation} \label{eps}
\epsilon > 0
\end{equation}

Uit (\ref{eps}) volgt \ldots
```

4.2 Symbolen in mathematische formules

In deze paragraaf worden de belangrijkste symbolen die in mathematische formules gebruikt worden kort behandeld. Een lijst van alle beschikbare symbolen vindt met in appendix C.

Kleine **griekse letters** worden ingevoerd met `\alpha`, `\beta`, `\gamma`, etc. Griekse hoofdletters worden ingevoerd via `\mathrm{A}`, `{\mathrm{B}`, `\Gamma`, `\Delta` (er is geen apart symbool voor Griekse hoofdletters die er hetzelfde uitzien als een Romeinse).

$\lambda, \xi, \pi, \Phi, \Omega$

```
$\lambda, \xi, \pi,
 \Phi, \Omega $
```

Daarnaast zijn er nog **mathematische symbolen**: van $\in$, $\Rightarrow$ tot ∞ (tabel C.7, C.4, C.9)

Exponenten en Indices kunnen met het teken `^` of `_` boven resp. onder een variabele geschreven worden. Als de exponent of index uit één teken bestaat mogen de accolades weggelaten worden.

$a_1 \qquad x^2 \qquad e^{-\alpha t} \qquad a^{3}_{ij}$

```
$a_1$ \qquad
$x^2$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$
```

Het **wortelteken** wordt met `\sqrt` aangegeven, de n-de wortel met `\sqrt[`n`]`. De grootte van het wortelteken wordt door LaTeX zelf vastgesteld.

$\sqrt{x} \qquad \sqrt{ x^{2}+\sqrt{y} } \qquad \sqrt[3]{2}$

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$ \qquad
$\sqrt[3]{2}$
```

De commando's `\overline` en `\underline` zetten een **horizontale lijn** direkt boven danwel onder de aangegeven tekst.

$\overline{m+n}$

```
$\overline{m+n}$
```

De commando's `\overbrace` en `\underbrace` zetten een **horizontale accolade** boven resp. onder de aangegeven tekst. Met `^` resp. `_` wordt de tekst bij de accolade aangegeven.

$\underbrace{ a+b+\cdots+z }_{26}$

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

Om mathematische **'Accenten'** zoals pijlen of slingertjes boven variabelen te zetten, staan de commando's uit tabel C.1 tot onze beschikking. Langere slingers en dakjes, die zich over meerdere (tot 3) karakters uitstrekken, verkrijgt men met behulp van de commando's `\widetilde` en `\widehat`. Differentiaaltekens worden met `'` (apostroph) aangegeven.

$$y = x^2 \qquad y' = 2x \qquad y'' = 2$$

```
\begin{displaymath}
y=x^{2} \qquad
y'=2x    \qquad
y''=2
\end{displaymath}
```

Een **breuk** (fraction) wordt dmv. het commando commando `\frac{...}{...}` gezet. Voor eenvoudige breuken kan men echter ook de operator `/` gebruiken.

$1\frac{1}{2}$ gulden

$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

```
$1\frac{1}{2}$~gulden
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

Mathematische **funkties** worden in de literatuur over het algemeen niet cursief (zoals variabelen) maar 'normaal' geschreven. Een en ander kan met de volgende commando's bereikt worden:

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`	`\min`	`\sinh`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`	`\sup`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`	`\tan`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`	`\tanh`

Voor de modulo-functie zijn er twee verschillende commando's: `\bmod` voor de binaire operator $a \bmod b$ en `\pmod{...}` voor deze vorm

$x \equiv a \pmod{n}$

```
$ x\equiv a \pmod{n} $
```

$$\lim_{x \to 0} \frac{\sin x}{x} = 1$$

```
\begin{displaymath}
\lim_{x \to 0} \frac{\sin x}{x}
=1
\end{displaymath}
```

Binomiaal-coefficiënten kunnen met `{...\choose...}` gezet worden. Met het commando `\atop` bereikt men hetzelfde maar dan zonder haakjes.

$$\binom{n}{k} \qquad {x \atop y+2}$$

```
\begin{displaymath}
{ n \choose k } \qquad
{ x\atop y+2 }
\end{displaymath}
```

Het **integraalteken** verkrijgt met `\int`, het **sommatieteken** met `\sum`. De boven- en ondergrenzen worden met `^` danwel `_` aangegeven.

Normaal gesproken worden de grenzen naast het integraalteken gezet (om ruimte te besparen), met het commando `\limits` wordt bereikt dat de grenzen boven en beneden het integraalteken gezet.

Bij het sommatieteken daarentegen worden de grenzen met het commando `\nolimits` naast het sommatieteken gezet en in alle andere gevallen boven en onder het sommatieteken.

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \int \limits_{-\infty}^{+\infty}$$

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\int \limits_{-\infty}^{+\infty}
\end{displaymath}
```

Voor haakjes zijn er in LaTeX verschillende symbolen (Bv. $[\ \langle\ \|\ \updownarrow$). Ronde haken en blokhaken kunnen met de vergelijkbare toets ingevoerd worden, accolades worden met `\{` en `\}` ingevoerd, voor de andere symbolen zijn er speciale commando's (Bv. `\updownarrow`). Zie tabel C.9.

Gebruikt men het commando `\left` voor haakjes openen en het commando `\right` voor haakjes sluiten, dan wordt automatisch de juiste grootte gekozen. De haakjes hoeven niet bij elkaar te passen en ook een `|` is toegestaan. Wil men *geen* linker- of rechterhaakje dan gebruikt men daar het `\left.` of het `\right.` commando. De `\left` en `\right` moeten altijd paarsgewijs voorkomen. Het tussenliggende deel van de formule bepaalt de grootte van de haakjes. Ertussen mogen geen nieuwe regels (`\\`) gegeven worden.

$$1 + \left\{ \frac{1}{1-x^{2}} \right|^3_1$$

```
\begin{displaymath}
1 + \left\{ \frac{1}{ 1-x^{2} }
    \right|^3_1
\end{displaymath}
```

In veel gevallen wil men de grootte van de haakjes liever zelf vaststellen. Daartoe kan men de commando's `\bigl`, `\Bigl`, `\biggl` en `\Biggl` inplaats van `\left` en analoog `\bigr` etc. inplaats van `\right` gebruiken.

$$\Bigl((x+1) (x-1) \Bigr)^{2}$$

```
\begin{displaymath}
\Bigl( (x+1) (x-1) \Bigr) ^{2}
\end{displaymath}
```

Wil men in een formule puntjes hebben (bv. $1, 2, \ldots, n$) dan kan men gebruik maken van de commando's `\ldots` en `\cdots`. `\ldots` zet de puntjes op de basislijn (low) `\cdots` zet de puntjes in het midden (centered). Daarnaast zijn er ook nog de commando's `\vdots` voor vertikale en `\ddots` voor diagonale puntjes.

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

4.3 Naast elkaar zetten

Als men niet tevreden is met de door LaTeX gekozen **afstanden**, dan kan men deze met expliciete commando's veranderen. De belangrijkste zijn `\,` voor een heel kleine afstand, `\:` voor een middelgrote afstand, `\;`, `\ `, `\quad` en `\qquad` voor grote afstanden en `\!` om een afstand te verkleinen.

$$F_n = F_{n-1} + F_{n-2} \qquad n \ge 2$$

```
\begin{displaymath}
F_{n} = F_{n-1} + F_{n-2}
 \qquad n \ge 2
\end{displaymath}
```

$$\int\!\!\!\int_{D} dx\,dy \quad \mbox{in plaats van} \quad \int\int_{D} dx dy$$

```
\begin{displaymath}
\int\!\!\!\int_{D} dx\,dy
\quad \mbox{in plaats van} \quad
\int\int_{D} dx dy
\end{displaymath}
```

4.4 Boven elkaar zetten

Voor **matrices** e.a is er de `array`-environment, die net zoals de `tabular`-environment functioneert. Het commando `\\` breekt de regels af. Het verschil tussen `array` en `tabular` is dat `array` alleen in *math*-mode gebruikt kan worden en dat elk element automatisch in *math*-mode staat, terwijl `tabular` helemaal in text mode is. Met behulp van de optionele parameter `[tb]` kan de verticale positie bepaald worden net als bij `tabular`.

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

Arrays kunnen ook in onverwachte situaties nuttig zijn:

$$|x| = \left\{ \begin{array}{ll} x & \mbox{if $x \geq 0$};\\ -x & \mbox{if $x < 0$}. \end{array} \right.$$

```
\[ |x| = \left\{
  \begin{array}{ll}
    x & \mbox{if $x \geq 0$};\\
    -x & \mbox{if $x < 0$}.
  \end{array} \right.  \]
```

Voor **meerregelige** formules of stelsels van vergelijkingen gebruikt men de environment `eqnarray` en `eqnarray*` inplaats van `equation`. Bij `eqnarray` krijgt elke regel zijn eigen vergelijkingsnummer bij `eqnarray*` wordt, net zoals bij `displaymath`, *geen* nummer toegevoegd. Voor stelsels van vergelijkingen die één gemeenschappelijk nummer dienen te hebben, kan men een `array`-environment binnen de `equation`-environment gebruiken.

De environments `eqnarray` en `eqnarray*` functioneren als een 3-koloms tabel van de vorm `{rcl}`, waarbij de middelste kolom voor het vergelijkingsteken wordt gebruikt. het commando `\\` breekt de regels af.

$$
\begin{aligned}
f(x) &= \cos x && (4.3)\\
f'(x) &= -\sin x && (4.4)\\
\int_0^x f(y)dy &= \sin x && (4.5)
\end{aligned}
$$

```
\begin{eqnarray}
f(x) & = & \cos x        \\
f'(x) & = & -\sin x      \\
\int_{0}^{x} f(y)dy &
 = & \sin x
\end{eqnarray}
```

Te lange vergelijkingen worden door LaTeX *niet* automatisch afgebroken. De auteur dient zelf aan te geven op welke plaats afgebroken moet worden en hoeveel er ingesprongen dient te worden. Het commando `\nonumber` zorgt ervoor dat er op deze plek geen vergelijkings nummer gezet wordt. Het commando `\lefteqn` maakt uitzonderingen op de kolomindeling binnen een `eqnarray` mogelijk. Het zet zijn argument, maar doet net of het geen ruimte inneemt.

$$
\begin{aligned}
\sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
&\quad -\frac{x^7}{7!} + \cdots && (4.6)
\end{aligned}
$$

```
\begin{eqnarray}
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!} -
                        \nonumber\\
 & & -\frac{x^{7}}{7!} + \cdots
\end{eqnarray}
```

$$
\begin{aligned}
&\cos x = 1 - \frac{x^2}{2!} + \\
&\qquad + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots && (4.7)
\end{aligned}
$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!} + }
                        \nonumber\\
 & & +\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!} + \cdots
\end{eqnarray}
```

Wanneer de `eqnarray` omgeving te beperkt is dan kan het `eqnarray` pakket uitkomst bieden. Dit levert de nummering van `eqnarray` met de flexibiliteit van `array`.

Hoofdstuk 5

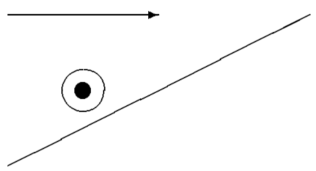
Figuren

5.1 Plaatjes in LaTeX

Voor het maken van eenvoudige plaatjes, die bestaan uit lijnen, pijlen, cirkels en tekst is in LaTeX de omgeving `picture` beschikbaar. Een picture kan midden in een zin worden opgenomen, maar dat is niet erg gebruikelijk (het is dan wel verstandig om de afmetingen klein te houden, zoals hier:). Vaak zal een picture bijvoorbeeld in een `quote` of een `center` omgeving opgenomen worden. Er is ook een drijvende figuur, net als bij tabellen (zie 5.2).

Een picture omgeving begint met de tekst `\begin{picture}`. Direct daarachter moet, tussen ronde haakjes, worden aangegeven wat de breedte en de hoogte van het plaatje is. De linkeronderhoek van het plaatje heeft meestal de coördinaten $(0, 0)$, maar als dat niet goed is kunnen ze als extra paar getallen tussen haakjes achter de afmeting opgegeven worden. Alle afmetingen worden gegeven in veelvouden van `\unitlength`. Default is dat 1 pt (zie tabel 3.1) maar met `\setlength\unitlength{...}` kun je de lengte-eenheid veranderen.

In een picture omgeving staan gewoonlijk alleen aanroepen van het commando `\put`. Direct achter `\put` staan de coördinaten waar iets neergezet moet worden, daarachter staat tussen accolades wàt er neergezet moet worden. Dat kan bijvoorbeeld een lijn of een cirkel zijn. Een voorbeeld van de opbouw van een picture volgt hieronder.



```
\setlength{\unitlength}{1mm}
\begin{picture}(40,20)
  \put(10,10){\circle{6}}
  \put(10,10){\circle*{2}}
  \put(0,0){\line(1,2){40}}
  \put(0,20){\vector(1,0){20}}
\end{picture}
```

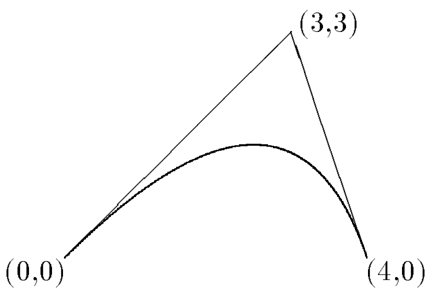
De volgende objecten kunnen optreden als parameter van `\put`:

- tekst: deze wordt in LR-mode gezet.
- `\circle{`*diam*`}` — een cirkel met gegeven diameter wordt neergezet met het middelpunt op de plaats die achter `\put` is gespecificeerd (maximum diameter is 40 punt).
- `\circle*{`*diam*`}` — de cirkel wordt zwart opgevuld (maximum diameter is 15 punt).
- `\line(`x, y`){`*breedte*`}` — een lijn met (x, y) als richtingsvector met een gegeven breedte (als de lijn vertikaal staat wordt niet de breedte maar de hoogte opgegeven).

- `\vector`(x,y){*breedte*} — een pijl in de gegeven richting met de gegeven breedte.
- `\oval`(x,y)[*deel*] — een ovaal dat past in een rechthoek met gegeven afmetingen. De parameter *deel* bepaalt (indien aanwezig) welk deel van het ovaal getekend moet worden: zet hier één (voor een half ovaal) of twee (voor een kwart ovaal) van de letters `t` (top), `b` (bottom), `l` (left) of `r` (right).
- `\makebox`(x,y)[*positie*]{*tekst*} — de gegeven tekst wordt in een onzichtbare rechthoek met gegeven afmetingen (x, y) gezet, waarbij de *positie* aangeeft waar de tekst wordt gezet: `l` (tegen de linkerkant), `r` (tegen de rechterkant), `t` (tegen de bovenkant), `b` (tegen de onderkant), `rb` (tegen de rechteronderhoek), enz.
- `\framebox`(x,y)[*positie*]{*tekst*} — als `\makebox`, alleen is het rechthoekige kader nu ook zichtbaar.
- `\dashbox`{*dl*}(x,y)[*positie*]{*tekst*} — als `\framebox`, alleen is het rechthoekige kader nu gestreept. *dl* is de lengte van de streepjes.
- Een andere `picture` omgeving. Deze heeft dan zijn eigen coördinatensysteem en kan desgewenst een eigen `\unitlength` krijgen.

Het volgende commando moet *niet* in een `\put` opdracht staan:

- `\qbezier`[n](x_1,y_1)(x_2,y_2)(x_3,y_3) — Een Bezier kromme van (x_1, y_1) naar (x_3, y_3) met (x_2, y_2) als stuurpunt. Het stuurpunt ligt niet op de kromme, maar is het snijpunt van de raaklijnen aan de eindpunten. Zie figuur 5.1. Als [n] opgegeven wordt dan is dit het aantal punten dat getekend wordt, anders bepaalt LaTeX dit zelf, met als maximum de waarde van `\qbeziermax` (gebruik `\renewcommand` om dit te wijzigen).



Figuur 5.1: `\qbezier(0,0)(3,3)(4,0)`
`\line(1,1)3`
`\line(-1,3)1`

Er zijn nogal wat beperkingen aan het picture-omgeving[1]. De schuinte van lijnen is bijvoorbeeld aan voorwaarden gebonden: de x en y waarmee de richtingsvector van een lijn wordt gespecificeerd moeten tussen -6 en 6 liggen en onderling ondeelbaar zijn. Voor pijlen (`\vector`s) moet dit zelfs tussen -4 en 4 zijn. Bovendien zijn, behalve de kwartcirkels die door `\oval` worden geproduceerd, ronde lijnen niet mogelijk. Tenslotte zijn alle tekeningen *lijn*tekeningen; het is niet mogelijk om arceringen aan te brengen.

De dikte van de lijnen kan aangegeven worden met het commando `\linethickness`{*lengte*}, waarbij een echte lengte opgegeven moet worden, bijv. `0.1mm`. Standaard diktes kunnen geselecteerd worden met de commando's `\thinlines` en `\thicklines`, waarvan de eerste standaard is.

Een belangrijk voordeel van de picture-omgeving is dat het plaatje in de tekst kan worden opgenomen, en gegarandeerd overal kan worden uitgeprint waar men LaTeX verstaat. Ook

[1] In de toekomst zal er een LaTeX 2_ε pakket pict2e beschikbaar komen waarmee deze beperkingen opgeheven kunnen worden.

zijn alle teksten in het plaatje netjes in hetzelfde lettertype als de rest van de tekst, en kunnen bijvoorbeeld formules worden opgenomen in het plaatje.

5.1.1 Pakketten voor LaTeXplaatjes

Om het maken van LaTeX-pictures wat eenvoudiger te maken zijn er pakketten beschikbaar die extra commando's bieden om plaatjes te maken. De naam van zo'n pakket moet worden opgenomen tussen de accolades in het `\usepackage` commando.

De volgende macro-pakketten kunnen handig zijn:

epic (een afkorting van 'extended picture'): hiermee kunnen eenvoudig allerlei soorten stippellijnen worden gemaakt. Bovendien kan een lijnfiguur worden gespecificeerd door de achtereenvolgende hoekpunten op te sommen, zonder voor ieder lijnstuk apart de richtingsvector te hoeven berekenen. De beperking dat de richtingscoëfficiënt van een lijn een simpele breuk moet zijn blijft; specificeer je een lijn die dat niet heeft, dan wordt deze benaderd met korte lijntjes die dat wel hebben.

eepic Dit is uitbreiding van `epic` waarbij de beperkingen opgeheven zijn. Gebruik `\usepackage{epic,eepic}`.

xypic met dit pakket kunnen relatief eenvoudig het soort diagrammen getekend worden waar wiskundigen dol op zijn: formules met pijlen ertussen waarbij bijschriften staan.

Het voert te ver om deze pakketten hier uitgebreid te behandelen; er is meer over te lezen in de bijbehorende documentatiefiles.

5.2 Drijvende Plaatjes (figure)

Onder een drijvende figuur (*figure*) wordt in LaTeX een stuk van een tekst verstaan dat niet op een vaste plaats hoeft te staan. Dat kan een plaatje zijn, maar ook bijvoorbeeld een listing van een programma, of een speciaal stuk tekst. Zo'n figuur kan een onderschrift hebben, en wordt door LaTeX automatisch genummerd. De figuur wordt door LaTeX op een plek gezet waar voldoende ruimte over is. Als een figuur meer ruimte kost dan de resterende ruimte op een pagina, dan wordt de figuur op de volgende pagina gezet.

Een figuur kan gemaakt worden met een `figure`-omgeving. Alle tekst die tussen `\begin{figure}` en `\end{figure}` staat behoort tot de figuur. In die tekst kan met een aanroep van `\caption` worden aangegeven wat het onderschrift (of bovenschrift) van de figuur is. In de tekst van het `caption` commando mogen bepaalde commando's zoals `\ref` en `\sqrt` niet gebruikt worden, zie sectie 7.9. De makkelijkste manier om een plaatje te maken is wel, om in een `figure`-omgeving simpelweg ruimte te reserveren door middel van een aanroep van `\vspace`.

Zoals gezegd wordt de nummering van de figuren door LaTeX zelf uitgevoerd. Als je vanuit de tekst daarnaar wilt verwijzen, kun je het figuur een symbolische naam geven met `\label` (na het `\caption` commando). Later kun je dan naar de figuur verwijzen met een aanroep van `\ref`. Bijvoorbeeld:

Figuur 5.2: Functioneel wit

In figuur 5.2 staat een mooi plaatje.

```
\begin{figure}
\vspace{15mm}
\caption{Functioneel wit}
\label{fig:mooi}
\end{figure}
In figuur~\ref{fig:mooi} staat
een mooi plaatje.
```

In plaats van met `\vspace` gegenereerde witruimte mag er in de figuur natuurlijk ook iets anders staan. Een figuur kan bijvoorbeeld goed gebruikt worden om een programma-listing weer te geven die niet op een specifieke plaats in de tekst hoeft te staan:

```
int fac(int n)
{   return (n==0?1:fac(n-1))
}
```

Figuur 5.3: De faculteit-functie

```
\begin{figure}
\small
\begin{verbatim}
    int fac(int n)
    {   return (n==0?1:fac(n-1))
    }
\end{verbatim}
\caption{De faculteit-functie}
\end{figure}
```

Het is over het algemeen gebruikelijk om de tekst *onder* de tekening te zetten. Verder is alles van toepassing wat bij de `table` omgeving gezegd is (sectie 3.3). Het is dus ook mogelijk twee of meer plaatjes samen te laten drijven door ze samen (ieder in een `minipage`) in een `figure` te zetten. Voor nog meer flexibiliteit, o.a. sub-nummering is het `subfigure` pakket een goede keus. Om een lijst van figuren te krijgen neemt men het `\listoffigures` commando op op de plaats waar de lijst gewenst is. Als in de lijst een andere tekst opgegeven moet worden dan in de figuur dan kan deze als (eerste) optioneel argument aan het `\caption` commando gegeven worden, net als bij de `\section` commando's en bij tabellen. Dit is gedaan bij figuur 5.1 op pagina 46. Hier is ook de caption tekst naast de figuur gezet door deze in een aparte `\parbox` op te nemen.

5.2.1 Plaatjes naast een tekst

Met het `\parpic` commando uit het `picins` pakket is het mogelijk om plaatjes op te nemen die niet de hele breedte van het papier beslaan. Er zijn wel enige beperkingen aan dit commando: zo gaat bijv. het gebruik van een `verbatim` omgeving rechts naast de figuur niet goed, en moet je ook zelf opletten dat de figuur niet op een pagina-overgang terecht komt.

Hier de figuur

Figuur 5.4: Een `picins` figuur

Het `picins` pakket gebruik je door vooraan in je document op te nemen:

```
\usepackage{picins}
```

Je kunt dan een plaatje invoegen in een alinea door bijvoorbeeld:

```
\piccaption{Een \texttt{picins} figuur\label{fig:picins}}
\parpic(5cm,2cm)[s]{Hier de figuur}
```

Het `\parpic` commando heeft 4 optionele argumenten en 1 verplicht:

(breedte, hoogte) geeft de grootte van het plaatje aan. Als dit weggelaten wordt, dan wordt de natuurlijke grootte genomen.

(x-offs, y-offs) geeft de verschuiving van het plaatje aan. Dit is vooral nuttig wanneer er teveel of te weinig wit om het plaatje heen staat.

[opties] `l` als het plaatje links van de tekst moet staan, `r` voor rechts, `f` om een lijnkader om het plaatje heen te zetten, `d` voor een stippelkader, `o` voor een kader met afgeronde hoeken, `s` voor een kader met schaduwrand zoals in figuur 5.4, en `x` voor een 3-dimensionaal kader.

[positie] bepaalt hoe het plaatje in de box gezet wordt (alleen van toepassing als de grootte apart opgegeven wordt): `l` voor links, `r` voor rechts, `t` voor top (bovenin), `b` voor bottom (onderin). Bij geen opgave wordt gecentreerd.

Verplicht: De figuur. Dit kan iedere LaTeX constructie zijn.

Als `picins` het aantal regels dat ingesprongen moet worden niet goed berekent dan kun je op een bepaald punt het commando `\picskip{`*n*`}` geven. Er worden dan vanaf dat punt n regels gereserveerd. Dit commando start een nieuwe alinea, en mag niet gevolgd worden door een lege regel. Als $n = 0$ dan stopt de tekst naast het plaatje. Als je twee plaatjes vlak achter elkaar hebt, kun je hiermee voorkomen dat ze door elkaar lopen.

De horizontale afstand tussen de tekst en het plaatje kan gedefinieerd worden met het commando `\pichskip{`*lengte*`}`.

Het commando `\picchangemode` heeft als effect dat de betekenis van de `l` en `r` opties omgewisseld wordt op linker (even) pagina's als de `twoside` optie aangezet is en `\nopicchangemode` heft dit weer op.

De volgende commando's definiëren eigenschappen van de kaders: `\dashlength{`*lengte*`}` definieert de lengte van de streepjes in een streepjeskader, `\shadowthickness{`*lengte*`}` definieert de dikte van de schaduwrand en `\boxlength{`*lengte*`}` geeft de diepte van een 3D kader aan.

De tekst bij een `picins` figuur moet aangegeven worden met het commando `\piccaption[inh]{tekst}`. Het optionele argument is voor de lijst van figuren, net als bij het gewone `\caption` commando. *Dit commando moet vóór het* `\parpic` *commando gegeven worden.* Een eventueel `\label` commando moet *in* het `\piccaption` gegeven worden. Waar het caption geplaatst wordt kan gestuurd worden met de volgende commando's:

`\piccaptionoutside` onder de box.
`\piccaptioninside` onder de figuur, in de box.
`\piccaptionside` naast de figuur, gecentreerd. In dit geval kan er geen tekst naast de figuur staan.
`\piccaptiontopside` naast de figuur, aan de bovenkant.

5.3 Postscript

Postscript is een taal waarmee de opmaak van een pagina beschreven kan worden. Postscript wordt veel gebruikt om printers te besturen, en is daarom ook een geliefd formaat voor tekenprogramma's. Behalve afbeeldingen kan in een Postscript-plaatje ook tekst worden opgenomen.

Postscript is veel flexibeler dan LaTeX voor het maken van plaatjes. Zo zijn er bijvoorbeeld de volgende dingen mogelijk:

- lijnen in alle richtingen
- gebogen lijnen
- geroteerde tekst
- gearceerde oppervlaktes
- *bitmaps* (1-op-1-afbeelding van een grafisch scherm)

Postscript heeft echter minder kennis van tekstopmaak dan LaTeX. Het is bijvoorbeeld niet mogelijk om automatisch woorden over regels te verdelen, en regels over pagina's. Voor lange stukken tekst is LaTeX dus geschikter, voor ingewikkelde figuren is Postscript geschikter.

Een Postscript-file is een gewone tekstfile, en kan dus worden aangemaakt en bekeken met de tekstverwerker. Hoewel dit in principe mogelijk is, is het meer gebruikelijk om Postscript-files te laten *genereren* door andere programma's.

Postscript is in feite een programmeertaal, waarbij het executeren van een programma tot gevolg heeft dat de layout van een pagina wordt bepaald. Het is niet nodig om te weten hoe deze taal is opgebouwd om Postscript-verwerkende programma's te kunnen gebruiken. Kennis van de taal is alleen nodig als je met de hand (of liever gezegd: met de editor) Postscript-programma's wilt schrijven. Ook is deze kennis natuurlijk nodig als je zelf programma's wilt schrijven die Postscript genereren. Een inleiding in de taal volgt in appendix E. Op bijna alle computers zijn tekenprogramma's te verkrijgen waarmee Postscript plaatjes gemaakt kunnen worden. Bekende tekenprogramma's zijn Corel Draw en Adobe Illustrator.

5.3.1 Inclusie in LaTeX

Postscript-plaatjes kunnen in een LaTeX-document worden opgenomen. Het nadeel daarvan is dat de resulterende dvi-file niet echt device-independent meer is: hij kan alleen nog maar worden gebruikt op devices die ook Postscript kennen. Dit hoeft geen groot bezwaar te zijn, omdat dvi-files vaak worden afgedrukt op een Postscript-laserprinter. De dvi-file wordt daartoe met het programma `dvips` omgezet in Postscript, en de toe te voegen Postscript-illustratie kan daarbij naadloos worden ingelast. De zogenaamde EPS (zie beneden) plaatjes kunnen met xdvi gepreviewd worden, voor andere postscript plaatjes kan dit wel eens een probleem zijn.

Een probleem is dat ten tijde van de conversie LaTeX → dvi bekend moet zijn hoeveel ruimte er opengelaten moet worden. Je moet in het algemeen een Postscript-tekst geheel interpreteren om te weten hoe groot een tekening is. Zo veel kennis van Postscript heeft LaTeX niet, dus moet hier iets anders op gevonden worden.

Net als in veel andere talen kan in een Postscript-programma commentaar worden opgenomen. Een regel commentaar begint met een procent-teken en loopt tot het einde van de regel. Een veelgebruikte conventie is om een speciale commentaar-regel in een Postscript-tekst op te nemen:

```
%%BoundingBox: 75 435 523 743
```

Hiermee wordt gespecificeerd wat de 'begrenzende rechthoek' van de tekening is. De eerste twee getallen zijn de coördinaten van de linker-onderhoek de andere twee getallen de coördinaten van de rechter-bovenhoek (in de eenheid *point*, dit is ongeveer $\frac{1}{3}$mm). Veel programma's die Postscript genereren nemen daarin een bounding-box commentaarregel in op. Zelf kan je natuurlijk ook zo'n regel toevoegen aan een Postscript-tekst.

Postscript-plaatjes waarin een bounding box wordt gespecificeerd heten *encapsulated postscript*, afgekort *EPS*. Aan een LaTeX-tekst kun je encapsulated-postscript plaatjes toevoegen. Om de afmeting te bepalen, hoeft LaTeX alleen de regel waarin `%%Bounding Box` staat op te zoeken.

Het toevoegen van een Postscript-plaatje aan een LaTeX-tekst gebeurt in modern LaTeX met `\includegraphics` en traditioneel met `\epsfbox`. Voor nieuwe documenten wordt `\includegraphics` aangeraden, maar we beginnen met de bespreking van `\epsfbox`.

In de *preamble* van het LaTeX-document moet je het pakket `epsf` of `epsfig` ('encapsulated Postscript figure') gebruiken, dus bijvoorbeeld

```
\usepackage{epsfig}
```

Het plaatje kan dan in een LaTeX-box geplaatst worden met het commando `\epsfbox{`*filenaam*`}`. Dit kan (voor hele kleine plaatjes) midden in de regel gebeuren, maar meestal zal deze aanroep in een `center` of `quote` omgeving staan. In dat geval moet er ook nog `\noindent` voor staan, om TeXnische redenen. Dus:

```
\begin{quote}
\noindent\epsfbox{plaatje.ps}
\end{quote}
```

Als in de Postscript-file onverhoopt geen Bounding box staat gedefinieerd, kun je die als optionele parameter meegeven aan `\epsfbox`, bijvoorbeeld

```
\epsfbox[0 0 480 220]{file.ps}
```

Het plaatje krijgt z'n 'natuurlijke grootte'. Je kunt de grootte beïnvloeden door de macro `\epsfsize` te herdefiniëren. Deze macro heeft twee parameters: de natuurlijke x-afmeting en de natuurlijke y-afmeting. Hij moet opleveren wat de gewenste x-afmeting in het eindresultaat is. De y-afmeting wordt vanzelf meegeschaald. Bijvoorbeeld om een plaatje te halveren:

```
\renewcommand{\epsfsize}[2]{0.5#1}
```

Of om een plaatje kleiner te maken als het te breed is:

```
\renewcommand{\epsfsize}[2]{\ifnum#1>\hsize\hsize\else#1\fi}
```

Een simpelere constructie is om via `\epsfxsize` de gewenste horizontale grootte of via `\epsfysize` de gewenste verticale grootte op te geven, bijvoorbeeld:

```
\setlength\epsfxsize{5cm}
```

In LaTeX 2$_\varepsilon$ kan je het standaard pakket `graphics` of `graphicx` nemen en dan het commando `\includegraphics{`*filenaam*`}` gebruiken. Als optionele argumenten kunnen opgegeven worden `[`x_1, y_1`]` en `[`x_2, y_2`]` voor resp. de coördinaten van de linkeronderhoek en de rechterbovenhoek (of alleen de rechterbovenhoek als de linkeronderhoek `[0,0]` is). Als je het pakket `graphicx` gebruikt heb je meer mogelijkheden: Als optionele argumenten kunnen o.a. meegegeven worden:

> `bb=`x_1 y_1 x_2 y_2 geeft de bounding box aan, voor het geval deze niet in de file aanwezig is.
> `angle=`*hoek* geeft de hoek aan waarover het plaatje geroteerd moet worden.
> `origin=`...geeft aan om welk punt geroteerd wordt (`c`=center, `tr`=top-right etc. Default=linksonder.
> `width=`*breedte*. Plaatje wordt zo nodig vergroot of verkleind.
> `height=`*hoogte*. Idem.
> `scale=`*vergroting of verkleining*, bijv. `scale=0.5`.

Bij meerdere van deze argumenten worden ze gescheiden door komma's tussen [ ] gezet. Voorbeeld:

```
\includegraphics[width=5cm,angle=90]{plaatje.ps}
```

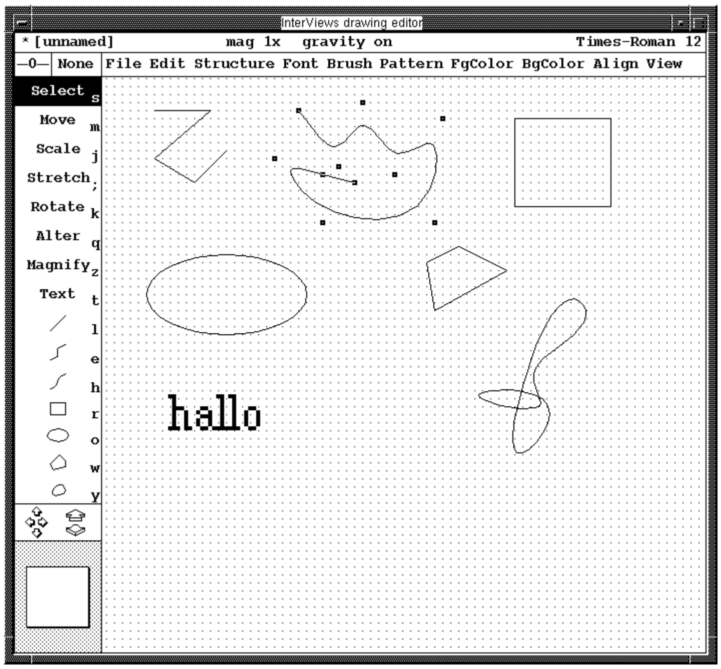
5.4 Tekenprogramma's

De programma's die in deze sectie behandeld worden zijn specifiek voor het Unix operating systeem.

5.4.1 idraw: Een tekenprogramma

De makkelijkste manier om een plaatje te maken is om hiervoor een interactief tekenprogramma te gebruiken. Een van de mogelijkheden is het programma `idraw`. Na een tijdje gebruik kan het scherm er uitzien als in figuur 5.5. Het grootste deel van het window wordt gebruikt om het plaatje weer te geven. Erboven staat en menubalk, links een 'itembox'. Linksonder staat een situatie-diagram, dat aangeeft welk deel van het plaatje 'in beeld' is. Door het witte vlak hierop te schuiven, wordt een ander deel van het plaatje zichtbaar. Met de vier pijltjes boven het situatiediagram kan ook worden geschoven. De twee grote pijltjes worden gebruikt om in- en uit te zoomen op een detail van de tekening.

Door op een item in de itembox te klikken (of door een letter op het toetsenbord te tikken) kunnen verschillende soorten objecten getekend worden: rechte lijn, stuksgewijs rechte lijnen, curve, rechthoek, ovaal, gesloten polygon, gesloten curve, of tekst. Bij de figuren die uit



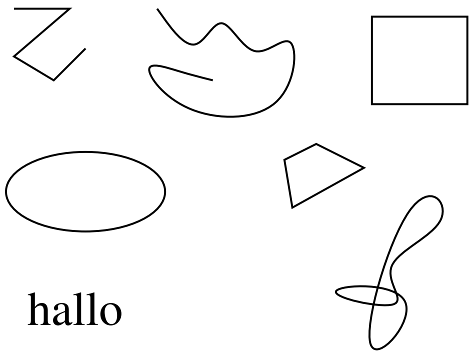
Figuur 5.5: `idraw`, een interactief tekenprogramma

meerdere delen bestaan (zoals curves) wordt het laatste punt gespecificeerd door op de rechter muisknop te klikken.

Aan het laatst getekende object zitten zwarte vierkantjes. Een object kan verschoven worden door 'move' te selecteren, en hem met de muis bij zo'n vierkantje op te pakken. Met 'scale' kan een figuur groter of kleiner gemaakt worden, met 'stretch' worden uitgerekt in één richting. 'Rotate' draait een figuur, en met 'alter' kunnen (bijvoorbeeld in een curve) nog hoekpunten worden verplaatst. Transformaties kunnen op meerdere objecten tegelijk werken (bijvoorbeeld om ze allemaal evenveel te vergroten) door ze aan te klikken terwijl de shift-toets is ingedrukt. Nog handiger is het om meerdere geselecteerde objecten te groeperen met het commando 'group' in het 'structure' menu, en ze daarna als één geheel te behandelen. Daarna kunnen ze desgewenst weer worden losgmaakt met 'ungroup'.

De tekening is een vector-tekening, dat wil zeggen dat alle objecten opgeslagen worden in de vorm van hoekpunten en dergelijke. Objecten die elkaar overlappen kunnen daarom ook weer worden gescheiden. De afbeelding op het scherm is een ruwe benadering van het uiteindelijke resultaat: dit kan veel mooier zijn (vooral wat betreft teksten en curves). Het resultaat, zoals dat op de printer wordt afgedrukt, is te zien in figuur 5.6.

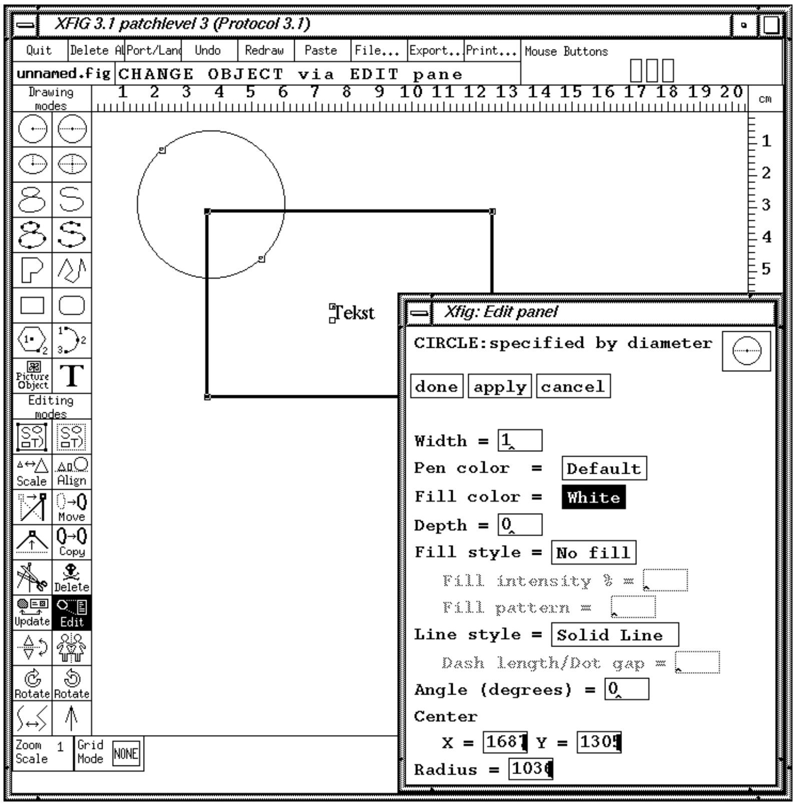
Het stippenpatroon op de achtergrond wordt niet afgedrukt. Het is bedoeld als hulpmiddel bij het tekenen. Met het 'align'-menu kan worden ingesteld dat alle hoekpunten moeten samenvallen met een rasterpunt ('align to grid'). Daarmee kun je nette tekeningen maken. Met de andere menu's kunnen dingen worden ingesteld als lijndikte, invulkleur van gesloten objecten, lettertype van teksten, en pijlkoppen aan lijnuiteinden



Figuur 5.6: uitvoer van `idraw` op de printer

5.4.2 xfig: nog een tekenprogramma

Een ander tekenprogramma is `xfig`. Zoals blijkt uit de ikoontjes in figuur 5.7 heeft dit programma ongeveer dezelfde mogelijkheden als `idraw`.



Figuur 5.7: `xfig`, een tweede interactief tekenprogramma

Een extra eigenschap van `xfig` is dat het programma op diverse manieren het plaatje kan weergeven, o.a. als een LaTeX `picture` omgeving, als `epic` of `eepic` uitbreidingen en als `EPS`. Bij gebruik van de LaTeX `picture` omgeving mogen dan natuurlijk geen dingen gebruikt worden die in zo'n omgeving onmogelijk zijn, zoals gebogen lijnen. Er zijn aparte ikoontjes beschikbaar om je te helpen om alleen maar lijnen te tekenen met door LaTeX toegestane richtingscoëfficiënt. Bovendien heeft `xfig` de mogelijkheid om de tekst in een plaatje in een LaTeX lettertype af te drukken i.p.v. een Postscript lettertype. Dit geeft een betere uniformiteit in je document en ook de mogelijkheid om formules e.d. in de tekst op te nemen. Hiervoor moet de "special" optie van de tekst gekozen worden.

De plaatjes worden door het programma opgeslagen in een privé-formaat. Deze files kunnen worden geconverteerd naar Postscript en, binnen de genoemde beperkingen, naar LaTeX met de `export` optie.

`xfig` kan ook EPS plaatjes opnemen maar deze niet meer wijzigen. Wel kunnen elementen aan een tekening toegevoegd worden, waardoor EPS plaatjes uit een andere bron geannoteerd kunnen worden.

Behalve interaktief tekenen kan `xfig` elementen van een tekening ook via de `edit` optie wijzigen, waarbij bijvoorbeeld de coördinaten exact aangegeven kunnen worden. In figuur 5.7 is zo'n "edit panel" zichtbaar.

In `xfig` wordt de muis anders gebruikt dan in `idraw`. Zo wordt een rechthoek niet getekend door met ingedrukte muisknop de rechthoek te trekken, maar door twee afzonderlijke hoekpunten aan te klikken. In de rechterbovenhoek laat `xfig` voortdurend zien wat de functies van de drie muisknoppen zijn.

5.4.3 Ipe: Integrated Picture Environment

Ipe is een recent tekenprogramma dat de opmerkelijke eigenschap heeft dat het interne formaat waarin tekeningen opgeslagen worden zowel een geldige postscript file als een geldig stuk LaTeX is. De Ipe-files kunnen zonder meer in een LaTeX document worden opgenomen. Ipe splitst het tekstgedeelte van het plaatje van het grafische deel. Voor de tekst worden gewone LaTeX-commando's gebruikt, voor het grafische deel Postscript. Verder heeft ipe de mogelijkheid om bestaande Postscript files in te lezen en te bewerken. Voor meer informatie moet u de Ipe documentatie bestuderen.

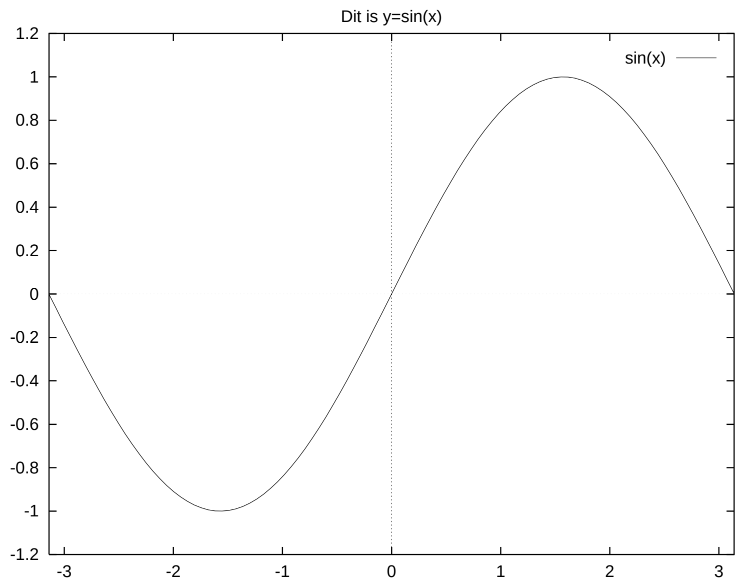
5.5 Grafieken maken

Voor het maken van grafieken, bijvoorbeeld voor de weergave van meetresultaten, zijn de interactieve tekenprogramma's niet altijd even handig. Hiervoor zijn aparte programma's beschikbaar:

5.5.1 Gnuplot

Gnuplot is een programma om grafieken te maken. Gnuplot kan de grafiek van een formule tekenen, maar ook een file van coördinaten gebruiken. Gnuplot is verkrijgbaar voor vele computersystemen en kan ook uitvoer genereren in verschillende formaten, o.a. LaTeX `picture` formaat en Postscript. Gnuplot wordt meestal niet interactief gebruikt maar via commando's die in een file ("script") gezet worden. Hier beneden ziet u eenvoudige scripts voor het tekenen van een functie, en van twee grafieken waarvan de gegevens uit een file gehaald worden. De bijbehorende grafieken staan in figuur 5.8 en 5.9.

```
set terminal postscript eps
set output "plot1.eps"
set title "Dit is y=sin(x)"
plot [-3.14:3.14] sin(x)
```

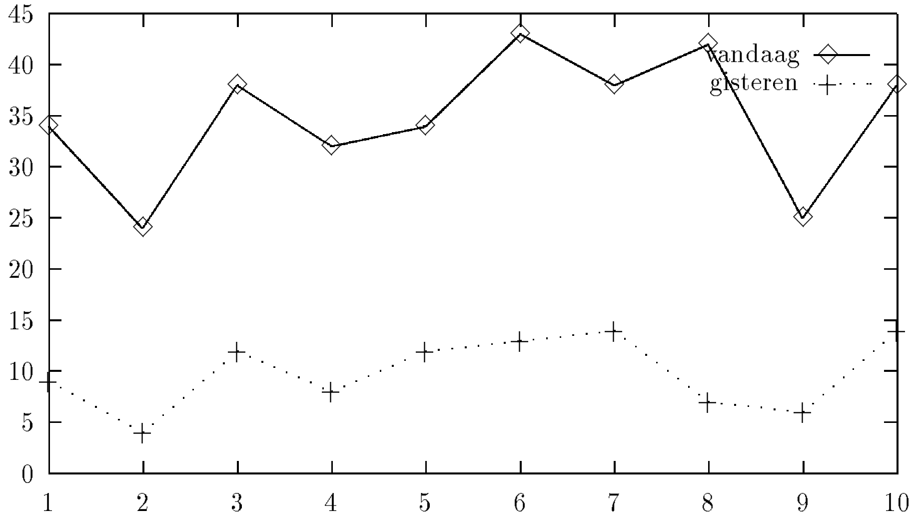


Figuur 5.8: $y = sin(x)$

```
set terminal latex
set output "plot2.tex"
set data style linespoint
plot "plot.data" using 1:2 title "vandaag",\
     "plot.data" using 1:3 title "gisteren"
```

Met het commando `set terminal` wordt aangegeven wat voor soort uitvoer gewenst is. Enkele mogelijkheden zijn: `latex` voor een LaTeX `picture` omgeving, `eepic` voor een `eepic picture` omgeving, `postscript` voor Postscript uitvoer, en `postscript eps` voor EPS. Wanneer EPS uitvoer gebruikt wordt dan deze met het bovengenoemde `\includegraphics` commando ingevoegd worden. Een LaTeX uitvoer kan met het commando `\input{`*filenaam*`}` ingevoegd worden.

Ingewikkelde grafieken kunnen bij uitvoer als `latex` of `eepic` te ingewikkeld worden waardoor LaTeX geheugen tekort komt. Voor zulke plaatjes is een vorm van Postscript meestal beter.



Figuur 5.9: Gnuplot grafiek

Maar dit heeft als nadeel dat de tekst in de grafiek een ander lettertype krijgt dan de rest van het document, en bovendien dat er geen LaTeX constructies (bijvoorbeeld formules) in de tekst gezet kunnen worden. Met `set terminal pslatex` wordt een combinatie van Postscript en LaTeX uitgevoerd. De uitvoer kan met het `\input` commando in het document gezet worden.

Het `set output` commando vertelt naar welke file gnuplot de uitvoer moet sturen. Met `set title` kan een titel voor de grafiek gedefinieerd worden. Deze wordt boven de grafiek geplaatst. Bij het tekenen van grafieken van een datafile kan aangegeven worden op welke manier de punten getekent moeten worden met het commando `set data style`. Een paar mogelijkheden zijn: `points` om alleen de punten te tekenen, `lines` om lijnen te tekenen en `linespoints` voor beide.

Het belangrijkste gnuplot commando is `plot` waarmee de echte grafiek getekend wordt. Voor het tekenen van een functie is de vorm: `plot [beg:eind] functie`, waarbij *beg* en *eind* de begin- en eindwaarde van de x-coördinaat zijn. Voor het tekenen van een grafiek van een datafile is de opdracht: `plot "filenaam" using colx:coly`, waarbij *colx* aangeeft in welke kolom van de invoer de x-coördinaat staat, en *coly* in welke de y-coördinaat. Als dit weggelaten wordt, dan wordt 1 resp. 2 genomen. Met de `title` aanduiding kan een titel voor de grafiek aangegeven worden, wat handig is als er meer dan één tegelijk getekend wordt (in dat geval moet je één plot commando geven met meer functies en/of `using`'s).

Voor meer bijzonderheden over gnuplot zie de documentatie.

5.5.2 Xmgr – een interactief plot-programma

Xmgr is een Unix programma, waarmee op een meer interactieve manier grafieken gemaakt kunnen worden. Bovendien kunnen diverse bewerkingen uitgevoerd worden op de data en op de layout van de grafiek(en), bijv. het gebruik van logaritmische schalen. Omdat dit programma zeer veel mogelijkheden heeft is de documentatie nogal uitgebreid en voert het te ver om dit hier te bespreken.

5.6 Bitmaps

5.6.1 Bitmap-formaten

De tot nu toe besproken tekenprogramma's zijn bedoeld om vector-tekeningen te maken. Het voordeel van dit soort tekeningen is dat ze later eenvoudig kunnen worden aangepast, en dat ze onafhankelijk zijn van de resolutie van het gebruikte apparaat. Zo blijken tekeningen op de printer vaak mooier te zijn dan ze op het scherm leken.

Als je een afbeelding wilt maken van het beeldscherm, is de meer aangewezen weg een *bitmap*. Elk puntje op het scherm verschijnt dan als één puntje in de afbeelding. In zo'n bitmap kunnen geen losse objecten meer worden onderscheiden: dingen die over elkaar heen getekend worden, zijn nooit meer te scheiden.

Er is een groot aantal formaten in gebruik om een bitmap te beschrijven. Sommige zijn alleen voor zwart-wit plaatjes, andere kunnen ook voor kleurenplaatjes gebruikt worden. Sommige slaan elke acht pixels op in één byte, andere passen slimme compressietechnieken toe om grote egale vlakken efficiënter op te slaan.

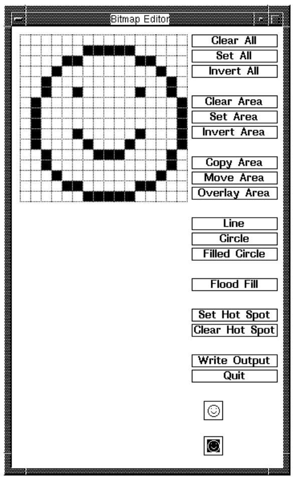
Het standaardformaat voor het X-window systeem (het grafische systeem voor Unix) is het *XBM*-formaat ('X-bitmap'). Voordeel hiervan is dat het met een editor te bekijken is, nadeel is dat het erg veel geheugen kost: bijna 1 byte per pixel. Voor kleine plaatjes is dat echter niet zo erg.

Een compacter formaat om bitmaps op te slaan is *GIF* ('Graphics Interchange Format'). Op pc's is worden de formaten *PCX*, *BMP* en *IMG* veel gebruikt. Dit zijn alledrie gecomprimeerde formaten.

Er zijn een aantal programma's onder de naam `pbmplus` en `Imagemagick` beschikbaar voor het omzetten van de diverse formaten in andere.

5.6.2 Bitmaps maken en gebruiken

Met het programma `bitmap` kunnen kleine bitmaps in xbm-formaat worden gemaakt en veranderd. Het gebruik van het programma wijst zichzelf; een impressie is te zien in figuur 5.10.Als je probeert een grote bitmap met dit programma te editten, worden de vierkantjes zo klein dat je ze niet goed meer kunt aanklikken. Het programma is dan ook in eerste instantie bedoeld voor het ontwerpen van cursors en kleine patroontjes.

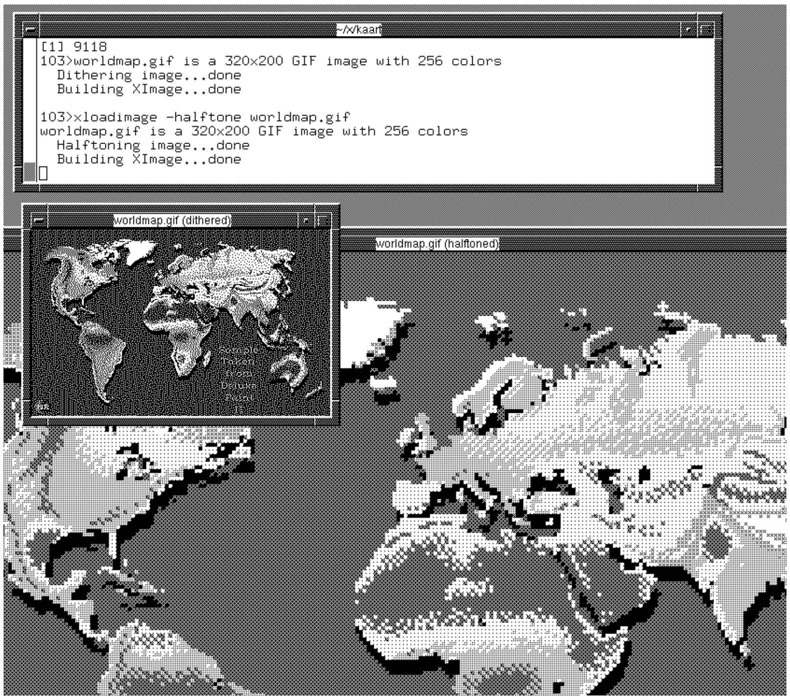


Figuur 5.10: `bitmap`, ontwerpen van kleine bitmaps

Het programma `xpaint` is een tekenprogramma dat geschikt om grotere bitmaps en ook kleurenplaatjes, z.g. pixelmaps te maken. Het kan een aantal verschillende formaten aan, o.a. GIF en XBM.

Er is een conversieprogramma om van een xbm-file een tekstfile te maken waarin elk pixel met een sterretje of een spatie wordt weergegeven. Dit programma heet `bmtoa`. De omgekeerde transformatie wordt uitgevoerd door `atobm`.

Bitmaps in alle genoemde formaten kunnen worden bekeken met de programma's `xloadimage` of `xv`. Het programma probeert (meestal met succes) zelf te raden wat het type is van de bitmap. Zwart-wit plaatjes worden direct afgebeeld; kleurenplaatjes worden zonodig naar zwart-wit geconverteerd. Daarbij zijn er twee mogelijkheden: 'dithering', waarbij het plaatje even groot blijft, en 'halftoning', waarbij het plaatje 4×4 keer zo groot wordt. Beide worden getoond in figuur 5.11. Beide programma's



Figuur 5.11: `xloadimage`, met twee manieren om kleurenplaatjes ZW weer te geven

kunnen ook bitmaps in een ander formaat wegschrijven. Xv kan ook interactief gebruikt worden om een aantal wijzigingen in een plaatje aan te brengen, bijvoorbeeld kleurwijziging of verandering van afmetingen.

5.6.3 Scherm-snapshots

Vaak wil je in een handleiding van een programma een 'snapshot' van het scherm neerzetten. Met het programma `xgrabsc` kun je daartoe een bitmap maken van een deel van het scherm. Bijna alle afbeeldingen in dit hoofdstuk zijn met `xgrabsc` gemaakt. Dit commando kun je in een ander window geven. Met de muis kun je daarna een deel van het scherm uitknippen of een heel window aangeven dat opgeslagen moet worden. Het programma kent opties voor de vorm van de output: `-eps` voor encapsulated Postscript, `-bm` voor xbm-formaat. Dat laaste wordt al snel erg groot.

Normaliter wordt het gewenste deel van het scherm aangegeven door met de muis een rechthoek te trekken. Het is ook mogelijk om een compleet window te tonen. Hiervoor is de optie `-click`. Met `-borders` of `-noborders` kun je aangeven of de rand er ook bij moet. Vaak wil je voordat het snapshot genomen wordt nog iets prepareren aan het window, bijvoorbeeld het window activeren. Je kunt dan de `-s` optie geven om aan te geven hoe lang het programma moet wachten:

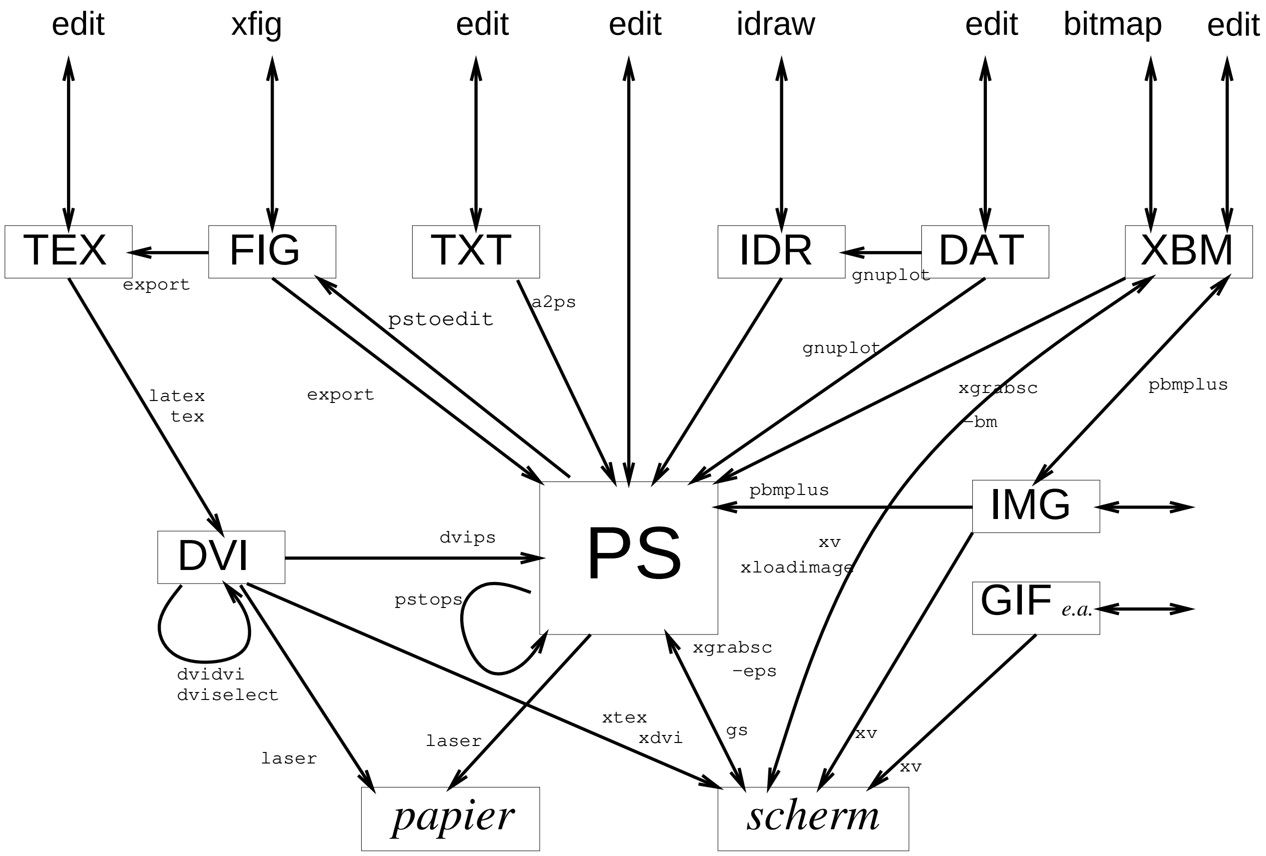
```
xgrabsc -eps -click -s 5> scherm.ps
```

Voor afbeeldingen van kleurenschermen zijn er ook opties om de kleuren naar zwart-wit te converteren (`-bw` om een puur zwart-wit plaatje te krijgen, `-halftone`, `-dither` of `-mdither` om grijstinten te krijgen.

In combinatie met `xloadimage` kan `xgrabsc` gebruikt worden om files uit allerlei grafische formaten om te zetten naar Postscript, om ze te kunnen opnemen in een LaTeX-tekst. Het programma `xv` kan zelf ook een "grab" doen.

5.7 Samenvatting

In figuur 5.12 wordt van (bijna) alle besproken conversieprogramma's aangegeven tussen welke filetypen ze werken.



Figuur 5.12: Overzicht van conversieprogramma's

Hoofdstuk 6

Commando's

6.1 Macro's

Het is soms nuttig om eigen commando's te kunnen definieren, bijvoorbeeld wanneer een bepaalde constructie vaker voorkomt in een document. Het simpelste gebruik van een commando (ook wel macro genoemd) is als afkorting van een stuk tekst of een LaTeX constructie. Stel dat in een artikel nogal vaak de volgende constructie voorkomt:

Γ_i	`$\Gamma_{i}$`

Het is dan handiger om te zeggen:

```
\newcommand{\Gi}{$\Gamma_{i}$}
```

waarna we verder in de tekst `\Gi` kunnen gebruiken. `\newcommand` heeft als eerste argument het commando dat we willen definiëren en als tweede de betekenis ervan. Het commando mag nog niet gedefinieerd zijn.

Er zijn in dit voorbeeld een aantal problemen: Ten eerste moeten we erop letten dat spaties na `\Gi` verdwijnen, zoals bij alle commando's die uit letters bestaan, ten tweede geeft dit commando een fout als het in *math-mode* gebruikt wordt, om dat de eerste `$` dan de *math-mode* afsluit i.p.v. begint. Er zijn twee manieren om dit laatste te voorkomen: ten eerste kunnen we de hele constructie in een `\mbox` zetten:

```
\newcommand{\Gi}{\mbox{$\Gamma_{i}$}}
```

of we kunnen de `$` tekens weghalen en LaTeX verzoeken om in *math-mode* over te gaan als dat nog niet zo is:

```
\newcommand{\Gi}{\ensuremath{\Gamma_{i}}}
```

De eerste oplossing heeft als nadeel dat LaTeX de grootte van de tekens niet goed bepaalt wanneer deze constructie in een exponent of index gebruikt wordt, de tweede oplossing verdient dus de voorkeur.

De namen van commando's mogen alleen uit letters bestaan, dus geen cijfers zoals in andere programmeertalen. Commando's mogen ook argumenten hebben (maximaal 9). Wanneer we een commando met parameters definiëren dan geven we als optioneel argument het aantal argumenten op. In het betekenis-deel kan aan de argumenten gerefereerd worden met de constructies `#1` ... `#9`, bijvoorbeeld:

$a_1, \ldots, a_n$
$x_i, \ldots, x_m$

```
\newcommand{\vect}[3]
  {\ensuremath{#1_{#2}, \ldots, #1_{#3}}}
\vect{a}{1}{n}\\
$\vect{x}{i}{m}$
```

Het is ook mogelijk een eigen commando te definiëren waarvan het eerste argument optioneel is, door in de definitie na het aantal argumenten een extra optioneel argument mee te geven. Dit argument bepaalt dan de waarde van het optionele argument van het commando wanneer dit niet meegegeven wordt:

```
\newcommand{\mijncmd}[2][eerste]{de argumenten zijn #1 en #2}
\mijncmd[een]{twee}
\mijncmd{twee} is hetzelfde als \mijncmd[eerste]{twee}
```

Zoals boven te zien is worden verplichte argumenten van commando's tussen accolades `{}` meegegeven, en optionele tussen blokhaken `[ ]`. Bij het invullen in de definiërende tekst worden deze verwijderd. Deze accolades vormen dan ook géén groep voor declaraties. De accolades die om te betekenis staan worden ook weggehaald bij het substitueren van de commando-aanroep.

Als je een bestaand commando een nieuwe betekenis wilt geven dan kan dat met `\renewcommand` i.p.v `\newcommand`. Hierbij moet uiterste zorgvuldigheid betracht worden dat niet een standaard LaTeX commando veranderd wordt!

6.2 Omgevingen

Op bijna dezelfde manier als commando's kunnen eigen omgevingen gemaakt worden. Hiervoor is het commando `\newenvironment`. Het enige verschil met `\newcommand` is dat zowel moet worden opgegeven wat er aan het begin als aan het einde van de omgeving moet gebeuren. Als de omgeving argumenten heeft mogen deze alleen in het *begin*stuk gebruikt worden. Heel vaak zal een omgeving in termen van een andere omgeving gedefinieerd worden, maar nodig is dit niet. Bijv.

> *Voorbeeld*: Dit is de binnenkant van de omgeving, niet een argument ervan!

```
\newenvironment{beschrijving}[1]%
   {\begin{quote}\emph{#1}:}% begin-stuk
   {\end{quote}}%             end-stuk

\begin{beschrijving}{Voorbeeld}
Dit is de binnenkant van de
omgeving, niet een argument ervan!
\end{beschrijving}
```

Net als `\renewcommand` bestaat er ook een `\renewenvironment`.

6.3 Stellingen

In veel wiskundige artikelen komen definities, stellingen, lemma's e.d. voor die allemaal dezelfde vorm hebben:

Stelling 6.1 *Voor elk geheel getal n bestaat er een volgend getal $n + 1$.*

```
begin{theorem}
Voor elk geheel getal $n$
bestaat er een volgend getal $n+1$.
\end{theorem}
```

Omgevingen als `theorem` zijn niet standaard in LaTeX gedefinieerd, maar kunnen eenvoudig zelf gemaakt worden met het commando `\newtheorem`, bijvoorbeeld:

```
\newtheorem{theorem}{Stelling}
```

Het eerste argument is de naam van de omgeving, het tweede is de tekst die voor het nummer afgedrukt wordt. Op dezelfde manier kunnen lemma's definities e.d. gedefinieerd worden. Ze worden gebruikt als een omgeving. Elk van de definities krijgt een eigen teller, maar het is mogelijk om de vorm van de teller te beïnvloeden of meerdere van deze omgevingen dezelfde teller te laten gebruiken.

Om een teller van een hoofdstuk of sectie te laten afhangen gebruik je de volgende vorm:

```
\newtheorem{theorem}{Stelling}[chapter]
```

Zoals in dit hoofdstuk worden de stellingen nu binnen de hoofdstukken genummerd. Het is ook mogelijk om hiervoor *section*, `subsection` e.d. te nemen. Let op dat geen `\` gegeven wordt.

Wanneer een stelling-achtige omgeving doorlopend genummerd moet worden met een andere dan kan dat door deze laatste als optioneel argument tussen de beide argumenten van `\newtheorem` op te geven:

```
\newtheorem{vermoeden}[theorem]{Vermoeden}
```

Vermoedens nummeren nu mee met stellingen.

De stelling-achtige omgevingen hebben één optioneel argument, dat tussen haakjes achter het nummer afgedrukt wordt:

Vermoeden 6.2 (Fermat) *Het bewijs dat Fermat claimde voor zijn beroemde vermoeden, was waarschijnlijk niet correct.*

```
\begin{vermoeden}[Fermat]
  Het bewijs dat Fermat claimde
  voor zijn beroemde vermoeden, was
  waarschijnlijk niet correct.
\end{vermoeden}
```

6.4 Condities

Het pakket `ifthen` maakt het gemakkelijk om if-then-else constructies te gebruiken. In een document kunnen we deze gebruiken door

```
\usepackage{ifthen}
```

Er zijn dan de volgende extra commando's mogelijk:

```
\ifthenelse{TEST}{THENCLAUSE}{ELSECLAUSE}
```

Een if-then-else commando waarbij

```
TEST == \equal{STRING}{STRING} |
        number OP number | \isodd{number}
        \(TEST\) | \not TEST | TEST \or TEST | TEST \and TEST |
        \lengthtest{length OP length}
OP   == < | > | =
```

met `\or`, `\and` and `\not` de bekende logische operators, `\(` and `\)` als haakjes, en `\equal` voor het testen van gelijkheid van strings.

```
\equal{STRING}{STRING}
```

Test of de twee argumenten gelijk zijn, nadat alle macro's geëxpandeerd zijn. Bijv. na `\newcommand\foo{bar}` ... geldt dat `\equal{\foo}{bar}` 'true' oplevert.

```
\whiledo{TEST}{DO-CLAUSE}
```

Een while-do constructie, met TEST hetzelfde als bij `\ifthenelse`.

Het is mogelijk logische variabelen te definiëren met `\newboolean{`*naam*`}`, een waarde te geven met `\setboolean{`*naam*`}` en te gebruiken met `\boolean{`*naam*`}`.

Hoofdstuk 7

Specialiteiten

Het complete menu van specialiteiten dat door LaTeX geserveerd wordt, is in het *LaTeX-Manual* [1] te vinden. Hier worden slechts enige bijzondere 'schotels' genoemd.

7.1 Afstanden

Eigen lengtevariabelen kunnen gemaakt worden met het commando `\newlength{`*naam*`}`. *Naam* moet een nog niet bestaande commandonaam zijn (beginnend met `\`). De variabele kan een waarde krijgen, net als voorgedefinieerde lengtevariabelen, met het commando `\setlength{`*naam*`}{`*lengte*`}`. *Lengte* mag een absolute lengte zijn, zoals `2cm`, of een lengtevariabele, of een factor maal een lengtevariabele, bijvoorbeeld `3.5\unitlength`. Het is ook mogelijk een lengtevariabele te verhogen (of te verlagen d.m.v. een negatieve waarde) met het commando `\addtolength{`*naam*`}{`*lengte*`}`.

7.1.1 Speciale horizontale afstanden

Het commando `\hfill` kan men gebruiken om een plek uit te vullen. `\hfill` is hetzelfde als `\hspace{\fill}` Voorbeeld:

links   rechts   `links\hfill rechts`

Tabel 7.1: Vertikale afstanden

`\smallskip`	ca. $\frac{1}{4}$ regel
`\medskip`	ca. $\frac{1}{2}$ regel
`\bigskip`	ca. 1 regel
`\vfill`	een afstand, die zich van 0 tot ∞ kan uitstrekken

7.1.2 Speciale vertikale afstanden

De afstanden tussen alinea's, hoofdstukken enz. wordt door LaTeX automatisch vastgesteld. In speciale gevallen kan men extra ruimte *tussen twee alinea's* creëren met het commando

`\vspace{`*lengte*`}`.

Dit commando dient altijd tussen twee lege regels te staan. Als de ruimte ook aan het begin of het einde van een bladzijde behouden moet blijven, dient men het commando `\vspace*` in plaats van `\vspace` gebruiken. De commando's in tabel 7.1 zijn afkortingen voor speciale vertikale afstanden. Het commando `\vfill` is hetzelfde als `\vspace{\fill}` en kan samen met `\pagebreak[4]` gebruikt worden om een tekst naar de bovenste of de onderste regel te 'drukken' of een tekst vertikaal te centreren.

7.2 Pagina referenties (pageref)

Met `\label` en `\ref` kunnen we refereren naar diverse nummers. Het is soms ook nuttig om een pagina nummer te kunnen gebruiken voor de referentie. Hiervoor is het commando `\pageref`, dat als argument dezelfde naam heeft als `\ref`, maar dan het pagina nummer oplevert. Het is mogelijk om voor dezelfde label zowel de `\ref` als de `\pageref` te gebruiken. Als je "intelligente" paginaverwijzingen wilt, waarbij, de tekst "op pagina x" weggelaten wordt op die pagina zelf, dan kan hiervoor het pakket `varioref` gebruikt worden.

7.3 Page Style

Het paginaformaat kan met het commando

`\pagestyle{`*style*`}`

vastgelegd worden:

Met `plain` (of als men `\pagestyle` niet opgeeft) staat het paginanummer aan de onderkant. Met `headings` staan de kopjes van de hoofdstukken en paginanummers bovenaan. Met `empty` staat er *niets* onder- of bovenaan.

In het *LaTeX-Manual* [1] is aangegeven, hoe men de pagina lay-out en bladzijdenummering met de commando's `\thispagestyle`, `\pagenumbering`, `\twocolumn` en `\onecolumn` kan beinvloeden.

Om ingewikkelder pagina-layouts te krijgen gebruikt men het pakket `fancyheadings` .

7.4 Papierformaten

Het in Europa gebruikte DIN-A4-papier is hoger en iets smaller als het in Amerika gebruikte papier (waar LaTeX van uit gaat). Met definities als

```
\addtolength{\topmargin}{-...}
\addtolength{\textheight}{...}
```

kan men de bovenste en onderste (witte) rand verkleinen en zo het aantal regels per pagina vergroten. De verandering van `\textheight` dient altijd in veelvouden van `\baselineskip` te geschieden, die weer van de letterhoogte afhangt. Aanbevolen waarden zijn:

letterhoogte	10pt	11pt	12pt
`baselineskip`	12pt	13.6pt	15pt
Δ`topmargin`	−30pt	−34pt	−30pt
Δ`textheight`	60pt	68pt	60pt

Al naar gelang de kop- of voetregel leeg is, kan men `\topmargin` ook wat groter of kleiner maken.

```
\addtolength{\topmargin}{-27pt}
\addtolength{\textheight}{68pt}
```

De regellengte en linker en rechter rand dient men niet te veranderen, omdat veel LaTeX-commando's die daarvan afhankelijk zijn dan vanzelfsprekend ook veranderd moeten worden. In veel gevallen kan men echter ook de papiergrootte ook in de `documentclass` opgeven. B.v.

```
\documentclass[a4paper, ...]{.....}
```

Wanneer een enkele pagina langer (of korter) gemaakt moet worden, kan dit met het commando

`\enlargethispage{`*lengte*`}`
`\enlargethispage*{`*lengte*`}`

De tweede vorm maakt de pagina niet langer dan strikt nodig is.

```
\documentclass[12pt]{brief}
\usepackage[dutch]{babel}
\address{Universiteit Utrecht\\
         Vakgroep Informatica\\
         Postbus 80.089\\
         3508 TB Utrecht}
\signature{P. van Oostrum}
\begin{document}
\begin{letter}{Jan Student\\
      IBB 1023/12345\\
      3501 XY Utrecht}
\opening{Beste Jan,}
Bedankt voor ....

..... We zullen dat dan verder bespreken.
\closing{Hartelijke groeten,}
\end{letter}
\end{document}
```

Figuur 7.1: Brief van PvO. aan JS.

7.5 Brieven (letter)

Als men als documentclass `letter` gebruikt, kan men tussen `\begin{document}` en `\end{document}` één of meer brieven schrijven.

Met `\signature` en `\address` definieert men de naam en het adres van de afzender. `\begin{letter}{...}` begint een brief met als parameters de naam en het adres van de ontvanger. `\opening{...}` verzorgt de aanhef. `\closing{...}` verzorgt samen met `\signature` de afsluitende groet. `\end{letter}` beeïndigt een brief.

Fig. 7.1 laat een voorbeeld van een brief zien. Omdat het papierformaat hierbij voorgedefinieerd is, moet hier geen `a4paper` optie, of `a4` package gebruikt worden.

Er is ook een `brief` documentclass die briefpapier volgens de officiële nederlandse normen produceert; je kunt hierbij je eigen briefhoofd ontwerpen. Zie de betreffende documentatie voor meer informatie.

7.6 Slides

Overhead transparanten kunnen gemaakt worden met de documentclass `slides`. Elke transparant wordt gemaakt met een `slide` omgeving. LaTeX gebruikt hiervoor een groter lettertype. Bij deze documentclass kunnen geen `figure` en `table` omgeving gebruikt worden.

7.7 Literatuurverwijzingen

Literatuurverwijzingen kunnen in de tekst opgenomen worden met het `\cite{`*key*`}` commando, waarbij *key* een naam is vergelijkbaar met die van `\label` en `\ref`. Er mag ook een rijtje namen met komma's ertussen opgegeven worden.

De literatuurlijst wordt afgedrukt met de `thebibliography` omgeving die één argument heeft dat iets groter dient te zijn dan de grootste label die voor een literatuurverwijzing voorkomt. Dit argument wordt alleen gebruikt om de layout te bepalen, het wordt niet afgedrukt. De `thebibliography` omgeving lijkt op een `enumerate`, maar i.p.v. `\item` wordt `\bibitem{`*key*`}` gebruikt.

Het is ook mogelijk om de `thebibliography` omgeving automatisch te laten genereren door het BibTeX programma. Op de plaats waar de bibliografie moet komen, geef je dan:

```
\bibliography{naam}
\bibliographystyle{stijl}
```

Zie de documentatie van BibTeX voor meer informatie.

7.8 Indexen

Een index kan gemaakt worden met het `makeindex` programma. Op de plaats waar het woord of begrip staat dat in de index moet worden opgenomen geeft je het commando `\index{`*woord*`}`. Het woord zelf moet ook gewoon in de tekst opgenomen worden.

Om de index te krijgen moet in de preamble opgenomen worden:

```
\usepackage{makeidx}
\makeindex
```

en op de plaats waar de index moet komen:

```
\printindex
```

Nadat het document door LaTeX verwerkt is wordt de index gesorteerd door het commando `makeindex` *`file`*, waarna de gesorteerde index opgenomen kan worden door weer LaTeX aan te roepen.

7.9 Verboden combinaties

Sommige commando's hebben argumenten die *verschoven* worden, d.w.z. dat het argument op een andere plaats gebruikt wordt dan waar het in de tekst voorkomt. Een duidelijk voorbeeld is de titel van een hoofdstuk of sectie, die in de inhoudsopgave terecht komt. Sommige commando's kunnen in deze verschuivende (*moving*) argumenten niet voorkomen, omdat ze breekbaar (*fragile*) zijn. Ze gaan dan kapot wanneer ze verschoven worden. LaTeX controleert niet hierop, maar geeft in zo'n situatie i.h.a. onbegrijpelijke foutmeldingen. Hieronder volgt een lijst van verschuivende argumenten en breekbare commando's. Om een breekbaar commando toch op te kunnen nemen is er een simpele oplossing: zet er het commando `\protect` voor (zonder accolades). Voorbeeld:

```
\section{Berekening van $\protect\sqrt{x}$}
```

Een `\protect` commando op een andere plaats doet i.h.a. geen kwaad.

7.9.1 Verschuivende argumenten

- De optionele argumenten van `\chapter`, `\section`, ... en `\caption`. Als geen optioneel argument meegegeven wordt, dan is het verplichte argument verschuivend.
- Het argument van de `letter` omgeving.

7.9.2 Breekbare commando's

- `\begin` `\end` `\(` `\)` `\[` `\]` `\footnote` `\sqrt`[1] `\ref` `\pageref` `\underline` `\cleardoublepage` `\cite`[1] `\vspace`
- Alle commando's met optionele argumenten en de meeste met een `*` vorm, bijv. `\item`, `\\`.

[1] `\sqrt` en `\cite` zijn in de nieuwste LaTeX 2_ε versie niet meer breekbaar.

Appendix A

Local Guide

Deze sectie beschrijft de bijzonderheden van de LaTeX-installatie zoals die is te vinden op de werkstations bij de Vakgroep Informatica van de Universiteit Utrecht. Andere installaties, en bijv. LaTeX versies op PC's zullen i.h.a. in details verschillen.

Het Unix commando `latex` wordt gebruikt voor het verwerken van LaTeX 2_ε documenten en LaTeX2.09 documenten in compatibility mode (zie hoofdstuk B). LaTeX2.09 documenten die niet met het `latex` commando verwerkt kunnen worden, kunnen met het Unix commando `latex209` verwerkt worden. Het wordt echter aangeraden deze documenten om te zetten naar LaTeX 2_ε, omdat `latex209` niet meer onderhouden wordt.

Andere programma's die aanwezig zijn:

`dvips` dit programma converteert een dvi-file naar Postscript. Het wordt automatisch gebruikt bij het laseren van een dvi-file.

`xdvi` previewen.

`laser` met dit programma wordt een Postscript-plaatje op de laserprinter afgedrukt. In feite kan de printer alleen maar Postscript-plaatjes afdrukken. De manier waarop de printer te werk gaat om het type van een file te bepalen is als volgt. Als het eerste character van een file het teken `%` is, dan neemt de printer aan dat de file een Postscript-file is. Veel Postscript-programma's beginnen om deze reden met een regel commentaar. Als het eerste character van de file `0xF7` is, dan neemt de printer aan dat het een dvi-file is, en wordt automatisch de transformatie `dvips` toegepast. In alle andere gevallen neemt de printer aan dat het een gewone tekstfile betreft, en wordt automatisch de transformatie `a2ps` toegepast. De environment variabele `LASER` of de optie `-P` kan gebruikt worden om aan te geven op welke printer afgedrukt moet worden.

`makeindex` sorteren van indexen.

`bibtex` maken van bibliografieën.

`bibview` een grafisch programma voor het maken van `bibtex` files.

`idraw, xfig, ipe` tekenprogramma's.

`dvidvi, dviselect, dvibook` manipuleren van dvifiles.

`dviconcat` plakt dvifiles aan elkaar.

`pstops` manipuleren van Postscript files.

`ispell` spellingscontrole.

`lacheck` is een programma dat uitgebreide controles doet op een LaTeX file. O.a. de correcte nesting van `\begin` en `\end` aanroepen, accolades, maar ook gebruik van `~`, `\@` e.d. Het programma is soms overijverig, maar kan nuttig zijn om hardnekkige fouten op te sporen. Vanuit emacs met de aux-tex mode is `lacheck` op te roepen met `C-c C-c Check`, waarna met het commando `` C-x ` `` telkens naar de volgende probleemregel gesprongen kan worden.

`mathpad` een interactieve grafische editor voor LaTeX documenten.

`gnuplot, xmgr` grafieken maken.

`ghostview` (ook wel `gv`) dit is een programma waarmee Postscript-programma's op het scherm kunnen worden bekeken. Als het een document met meerdere pagina's betreft, kun je daardoorheen bladeren door in het window waar je het programma aanroept op return te drukken. Dit programma is ideaal om postscript-programma's te debuggen zonder al te veel papier te gebruiken. Het programma kan ook gebruikt worden om te controleren of Postscript-plaatjes op de goede plaats in LaTeX-files terecht zijn gekomen als de previewer niet in staat is dit te laten zien. Daartoe moet je de dvi-file eerst met `dvips` converteren naar een ps-file, die je vervolgens met `ghostview` kunt bekijken.

Documentatie voor deze programma's kan gevonden worden via het WWW, afdeling "Software Documentation" subgroep "tex", of direct als

`http://www.cs.ruu.nl:82/Text.html#tex`.

Hier is ook documentatie te vinden over:

- De standaard LaTeX commando's
- De aanwezige pakketten
- De oplossing van verschillende problemen

Voor gebruikers van `emacs` staat de `auc-tex` mode ter beschikking waarmee gemakkelijker LaTeX documenten samen te stellen zijn. Deze mode kent de namen en de argumenten van de LaTeX omgevingen, heeft speciale commando's voor het genereren van fontwijzigingen en voor het uitvoeren van verschillende programma's (`latex, bibtex, makeindex, ispell`). Om deze te gebruiken moet je in je `.emacs` file opnemen:

```
(load "tex-site")
(require 'ltx-help)
```

De laatste regel heeft tot gevolg dat vanuit `emacs` de documentatie van de LaTeX commando's simpel is op te vragen met de emacs-opdracht `C-H C-L` (control/H control/L). Deze opdracht vraagt om de naam van een commando of omgeving (default die waar de cursor op staat) en geeft dan de informatie via de `emacs info` mode. Je kunt ook rechtstreeks met het info commando naar de menu-entry "LaTeX" springen. Dit is dezelfde informatie als die via het WWW `http://www.cs.ruu.nl/net/doc/tex/html/latex2e/latex2e_toc.html` te vinden is.

Alle pakketten die genoemd worden in appendix D zijn geïnstalleerd, en up-to-date informatie is te vinden in `http://www.cs.ruu.nl/net/doc/tex/list.html`. Speciale locale pakketten zijn `uusol` voor het genereren van het beeldmerk van de universiteit, en de documentclasse `uuletter` voor het maken van brieven op officieel briefpapier.

Appendix B

Compatibility mode

LaTeX2.09 is de vorige versie van LaTeX, waar sommige commando's anders zijn dan in de huidige versie van LaTeX (LaTeX 2ε). LaTeX 2ε kan echter de meeste LaTeX2.09 documenten gewoon verwerken. Deze sectie beschrijft de belangrijkste verschillen tussen beide LaTeX versies.

Een LaTeX2.09 document wordt gekenmerkt doordat het begint met `\documentstyle` i.p.v. `\documentclass`. Bovendien wordt het commando `\usepackage` niet gebruikt, maar de namen van pakketten worden als opties aan het `\documentstyle` opgegeven. Dit heeft tot gevolg dat de pakketten geen opties kunnen hebben. I.h.b. bij het gebruik van `babel` heeft dit tot gevolg dat `\documentstyle[dutch]{...}` gebruikt moet worden, i.p.v. de combinatie `\documentclass` en `\usepackage`. Een aantal opties zijn in LaTeX2.09 niet gedefinieerd, en sommige LaTeX 2ε pakketten werken niet samen met LaTeX2.09 (bijv. `graphics`).

Een aantal LaTeX 2ε commando's is niet gedefinieerd in LaTeX2.09, bijv. het commando `\enlargethispage`, `\ensuremath`, `\qbezier` en de lettertype-commando's zoals `\textrm`, `\mathrm`, etc. en de bijbehorende declaraties. In plaats daarvan heeft LaTeX2.09 declaraties `\rm`, `\bf`, `\it`, `\tt`, `\sf`, `\sl`, `\sc`. Het verschil met de LaTeX 2ε commando's is dat de LaTeX2.09 varianten niet te combineren zijn, en ook niet goed combineren met de commando's die de lettergrootte veranderen, zoals `\large`. Deze commando's zetten het lettertype terug naar `\rm`, dus `\bf\large` geeft het verkeerde effect, terwijl `\large\bf` het gewenste grote vette lettertype geeft. Ook geeft `\bf\it` niet een vet cursief, maar een gewoon cursief lettertype.

LaTeX 2ε kent deze commando's nog steeds om het gemakkelijker te maken bestaande documenten om te zetten, maar het gebruik ervan wordt ontraden i.v.m. deze ongewenste verschijnselen.

In de `figure` en `table` environments kan in LaTeX2.09 niet de `!` optie gebruikt worden.

B.1 Omzetting van LaTeX2.09 naar LaTeX 2ε documenten

Om een LaTeX2.09 document om te zetten naar LaTeX 2ε moeten de volgende wijzigingen uitgevoerd worden:

- `\documentstyle[options, style, style, ...]{class}` moet worden omgezet in

```
\documentclass[options]{class}
\usepackage{style}
...
```

- De taal-style opties `dutch`, `german` etc. moeten gebruikt worden met het `babel` pakket:

  ```
  \usepackage[dutch]{babel}
  ```

- Het gebruik van AMS symbols (bijv. `\input{mssymb}` of de `amssym` optie of vergelijkbare constructies) moeten worden vervangen door

  ```
  \usepackage{amssymb}
  ```

  De declaratie `\Bbb` of `\msy` moet vervangen worden door het commando `\mathbb{...}`.

- Er is geen apart commando `slitex` meer. LaTeX2.09 slide files hebben een `\blackandwhite` of `\colorslides` commando. Dit moet verwijderd worden en de gerefereerde file wordt gewoon in de hoofdfile opgenomen.

- Sommige LaTeX2.09 files gebruiken interne LaTeX commando's om lettertypes/groottes te selecteren. Soms kan dit opgelost worden door het pakket `rawfonts` op te nemen.

- De declaraties als `\bf` e.d. werken niet altijd goed in *math-mode*. Deze moeten dan omgezet worden naar de bijbehorende `\math..{...}` commando's. Dit geldt in ieder geval altijd voor `\sc` en `\sl` in math-mode.

Appendix C

Lijst van mathematische symbolen

De volgende tabellen laten alle symbolen zien, die standaard in de mathematische mode beschikbaar zijn.

Tabel C.1: Mathematische accenten

`\hat a`	$\hat{a}$	`\check a`	$\check{a}$
`\tilde a`	$\tilde{a}$	`\acute a`	$\acute{a}$
`\grave a`	$\grave{a}$	`\dot a`	$\dot{a}$
`\ddot a`	$\ddot{a}$	`\breve a`	$\breve{a}$
`\bar a`	$\bar{a}$	`\vec a`	$\vec{a}$

Tabel C.2: Kleine griekse letters

α	`\alpha`	ι	`\iota`	ϱ	`\varrho`
β	`\beta`	κ	`\kappa`	σ	`\sigma`
γ	`\gamma`	λ	`\lambda`	ς	`\varsigma`
δ	`\delta`	μ	`\mu`	τ	`\tau`
ϵ	`\epsilon`	ν	`\nu`	υ	`\upsilon`
ε	`\varepsilon`	ξ	`\xi`	ϕ	`\phi`
ζ	`\zeta`	o	`o`	φ	`\varphi`
η	`\eta`	π	`\pi`	χ	`\chi`
θ	`\theta`	ϖ	`\varpi`	ψ	`\psi`
ϑ	`\vartheta`	ρ	`\rho`	ω	`\omega`

Tabel C.3: Griekse hoofdletters

Γ	\Gamma	Ξ	\Xi	Φ	\Phi
Δ	\Delta	Π	\Pi	Ψ	\Psi
Θ	\Theta	Σ	\Sigma	Ω	\Omega
Λ	\Lambda	Υ	\Upsilon		

Tabel C.4: Andere symbolen

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists
$\imath$	\imath	∇	\nabla	$\neg$	\neg
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat
ℓ	\ell	$\top$	\top	$\natural$	\natural
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp
$\Re$	\Re	$\Vert$	\\|	$\clubsuit$	\clubsuit
$\Im$	\Im	$\angle$	\angle	$\diamondsuit$	\diamondsuit
∂	\partial	$\triangle$	\triangle	$\heartsuit$	\heartsuit
∞	\infty	$\backslash$	\backslash	$\spadesuit$	\spadesuit

Tabel C.5: Verzamelingsoperatoren

$\sum$	\sum	$\bigcap$	\bigcap	$\bigodot$	\bigodot
$\prod$	\prod	$\bigcup$	\bigcup	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup	$\bigoplus$	\bigoplus
$\int$	\int	$\bigvee$	\bigvee	$\biguplus$	\biguplus
$\oint$	\oint	$\bigwedge$	\bigwedge		

Tabel C.6: Binaire operatoren

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\cap$	`\cap`	$\vee$	`\vee`
$\mp$	`\mp`	$\cup$	`\cup`	$\wedge$	`\wedge`
$\setminus$	`\setminus`	$\uplus$	`\uplus`	$\oplus$	`\oplus`
$\cdot$	`\cdot`	$\sqcap$	`\sqcap`	$\ominus$	`\ominus`
$\times$	`\times`	$\sqcup$	`\sqcup`	$\otimes$	`\otimes`
$\ast$	`\ast`	$\triangleleft$	`\triangleleft`	$\oslash$	`\oslash`
$\star$	`\star`	$\triangleright$	`\triangleright`	$\odot$	`\odot`
$\diamond$	`\diamond`	$\wr$	`\wr`	$\dagger$	`\dagger`
$\circ$	`\circ`	$\bigcirc$	`\bigcirc`	$\ddagger$	`\ddagger`
$\bullet$	`\bullet`	$\bigtriangleup$	`\bigtriangleup`	$\amalg$	`\amalg`
$\div$	`\div`	$\bigtriangledown$	`\bigtriangledown`	$\lhd$	`\lhd`*
$\rhd$	`\rhd`*	$\unlhd$	`\unlhd`*	$\unrhd$	`\unrhd`*

* alleen bij gebruik van een van het pakket latexsym.

Tabel C.7: Relationele operatoren

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq`	$\geq$	`\geq`	$\equiv$	`\equiv`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\ll$	`\ll`	$\gg$	`\gg`	$\asymp$	`\asymp`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubseteq$	`\sqsubseteq`*	$\sqsupseteq$	`\sqsupseteq`*	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\smile$	`\smile`	$\mid$	`\mid`	$\doteq$	`\doteq`
$\frown$	`\frown`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\Join$	`\Join`*				

* alleen bij gebruik van het pakket latexsym.

Tabel C.8: Negaties

$\not<$	`\not<`	$\not>$	`\not>`	$\neq$	`\not=`
$\not\leq$	`\not\leq`	$\not\geq$	`\not\geq`	$\not\equiv$	`\not\equiv`
$\not\prec$	`\not\prec`	$\not\succ$	`\not\succ`	$\not\sim$	`\not\sim`
$\not\preceq$	`\not\preceq`	$\not\succeq$	`\not\succeq`	$\not\simeq$	`\not\simeq`
$\not\subset$	`\not\subset`	$\not\supset$	`\not\supset`	$\not\approx$	`\not\approx`
$\not\subseteq$	`\not\subseteq`	$\not\supseteq$	`\not\supseteq`	$\not\cong$	`\not\cong`
$\not\sqsubseteq$	`\not\sqsubseteq`	$\not\sqsupseteq$	`\not\sqsupseteq`	$\not\asymp$	`\not\asymp`

Tabel C.9: Pijlen

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleft...`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleft...`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\leadsto$	`\leadsto`*		

* alleen bij gebruik van een van het pakket latexsym.

Tabel C.10: Linker haakjes

$($	`(`	$[$	`[`	$\{$	`\{`
$[$	`\lbrack`	$\lfloor$	`\lfloor`	$\lceil$	`\lceil`
$\{$	`\lbrace`	$\langle$	`\langle`		

Tabel C.11: Rechter haakjes

$)$	`)`	$]$	`]`	$\}$	`\}`
$]$	`\rbrack`	$\rfloor$	`\rfloor`	$\rceil$	`\rceil`
$\}$	`\rbrace`	$\rangle$	`\rangle`		

Tabel C.12: Bijzondere constructies

$\widetilde{abc}$	`\widetilde{abc}`	$\widehat{abc}$	`\widehat{abc}`
$\overleftarrow{abc}$	`\overleftarrow{abc}`	$\overrightarrow{abc}$	`\overrightarrow{abc}`
$\overline{abc}$	`\overline{abc}`	$\underline{abc}$	`\underline{abc}`
$\overbrace{abc}$	`\overbrace{abc}`	$\underbrace{abc}$	`\underbrace{abc}`
$\sqrt{abc}$	`\sqrt{abc}`	$\sqrt[n]{abc}$	`\sqrt[n]{abc}`
f'	`f'`	$\frac{abc}{xyz}$	`\frac{abc}{xyz}`

Tabel C.13: Synoniemen

Voor vele symbolen zijn er meerdere commando's beschikbaar.

$\neq$	`\ne` or `\neq`	`\not=`
$\leq$	`\le`	`\leq`
$\geq$	`\ge`	`\geq`
$\{$	`\{`	`\lbrace`
$\}$	`\}`	`\rbrace`
$\to$	`\to`	`\rightarrow`
$\gets$	`\gets`	`\leftarrow`
$\owns$	`\owns`	`\ni`
$\land$	`\land`	`\wedge`
$\lor$	`\lor`	`\vee`
$\lnot$	`\lnot`	`\neg`
$\vert$	`\vert`	`|`
$\Vert$	`\Vert`	`\\|`

Tabel C.14: Lettertypes in formules

Stukken tekst binnen math-mode kunnen in een ander lettertype gezet worden met de volgende commando's:

ABCdef	`\mathrm{ABCdef}`	ABCdef	`\mathit{ABCdef}`
$\mathnormal{ABCdef}$	`\mathnormal{ABCdef}`	$\mathbf{ABCdef}$	`\mathbf{ABCdef}`
ABCdef	`\mathsf{ABCdef}`	$\mathtt{ABCdef}$	`\mathtt{ABCdef}`
$\mathcal{ABC}$	`\mathcal{ABC}`	*alleen hoofdletters*	
$\mathbb{ABC}$	`\mathbb{ABC}`	*met pakket amsfonts of amssymb*	

Het commando `\boldmath` gegeven *buiten math-mode* zet alle volgende formules **vet**, het commando `\unboldmath` doet dit weer teniet.

Tabel C.15: Formule stijlen

Met de volgende commando's kan de grootte van formules beïnvloed worden. Standaard worden formules in een tekst in `\textstyle` en losstaande formules in `\displaystyle` gezet. Deze commando's zijn declaraties en gelden alleen binnen het deel van de formule waarin ze staan.

`\displaystyle`	$\displaystyle\sum_{i=1}^{n} x^i$
`\textstyle`	$\textstyle\sum_{i=1}^{n} x^i$
`\scriptstyle`	$\scriptstyle\sum_{i=1}^{n} x^i$
`\scriptscriptstyle`	$\scriptscriptstyle\sum_{i=1}^{n} x^i$

Tabel C.16: Niet-mathematische symbolen

De volgende symbolen zijn ook in tekstmode beschikbaar.

§	`\S`	¶	`\P`	©	`\copyright`*
†	`\dag`	‡	`\ddag`	£	`\pounds`

* alleen in text mode.

Appendix D

Pakketten

Hier volgt een overzicht van de belangrijkste pakketten. Omdat er telkens nieuwe pakketten verschijnen is het onmogelijk hiermee up to date te zijn. Voor de documentatie van de pakketten zie de Local Guide (appendix A).

naam	functie
afterpage	Voer een commando uit aan het begin van de volgende pagina
algorithms	Omgevingen voor algoritmes
amsmath	AmSLaTeX: Uitbreidingen van de American Mathematical Society vooral voor wiskundige formules
amssymb	Extra symbolen van de AMS
array	Uitbreiding van de `array` en `tabular` omgevingen
babel	Meertalige en niet-engelse documenten
beton	Fonts die gebruikt zijn in Knuth's "Concrete Mathematics"
bibunits	Bibliografie per sectie
calc	Maakt het mogelijk om te rekenen
caption	Aanpassing van figure en table captions
caption2	Verbeterde versie van `caption`
changebar	Gebruik van "changebars" in LaTeX documenten (lijnen die aangeven wat veranderd is)
chapterbib	Bibliografie per hoofdstuk
chess	Afdrukken van schaakdiagrammen
cite	Sorteren en samenvoegen van numerieke cite keys
color	Het gebruiken van kleur in tekst
colortab	Gebruik kleur in tabellen
dcolumn	Decimale getallen uitlijnen op de punt in `tabular`
delarray	Delimiters (haken e.d.) om `array`s
deleq	Flexibele nummering van vergelijkingen
eepic	Uitbreiding van `picture`
enumerate	Aanpassing van de nummering in `enumerate` omgevingen
epsfig	Het invoegen van Postscript plaatjes (oud)

... vervolgd op volgende pagina

naam	functie
eqnarray	Generalizatie van de `eqnarray` omgeving
euler	Gebruik van de "euler" fonts in math mode
fancybox	Fancy boxes, o.a. boxes om alinea's
fancyhdr	Aanpassing van pagina headers en footers
float	Aanpassing van *figure* an `table` en zelf maken van nieuwe "floats"
floatflt	Een pakket vergelijkbaar met `picins`
formlett	Het prodiceren van seriebrieven e.d.
ftnright	Zet voetnoten in de rechterkolom in tweekoloms documenten
geometry	Selecteer niet-standaard pagina formaat
graphics	Het invoegen van Postscript plaatjes (nieuw)
graphicx	Het invoegen van Postscript plaatjes (uitgebreid)
harvard	De Harvard bibliography style familie
hhline	Flexibeler horizontale en verticale lijnen in `tabular` en `array`
ifthen	Conditionale expressies in LaTeX documenten
indentfirst	Spring de eerste alinea na een sectiekop ook in
index	Gebruik van meer dan één index in een document
lastpage	Refereer het aantal pagina's in een document.
longtable	Tabellen die langer zijn dan een pagina (meer mogelijkheden dan `supertab`)
lscape	Afwisselen van portrait/landscape
ltxtable	Combineer longtable en tabularx
minitoc	Inhoudsopgave per hoofdstuk
moreverb	Uitbreidingen van verbatim
multicol	Meerkoloms documenten, evt. afwisselend met enkelkoloms
musictex	Muziek afdrukken
newalg	Afdrukken van algoritmen zoals Cormen, Leiserson and Rivest.
overcite	Citeer m.b.v. superscript
picins	Plaatjes naast een tekst
psfrag	Kan teksten in EPS plaatjes vervangen, o.a. om de juiste fonts en LaTeX constructies te gebruiken.
pstricks	Gebruik Postscript constructies in LaTeX
rawfonts	Gebruik ongedocumenteeerde LaTeX2.09 font commando's (helpt soms om oude documenten te verwerken)
rcs	Gebruik RCS keywords in een document
rotating	Roteren van (o.a.) plaatjes
seminar	Seminar: een uitgebreider pakket voor overhead transparanten
showkeys	Laat label, ref, cite en bib keys zien
showlabels	Laat label commando's in de kantlijn zien
stmaryrd	Allerlei nieuwe symbolen voor o.a. functioneel programmeren
subeqnarray	Equation array met sub-nummering
subfigure	Verdeel een figuur in sub-figuren
supertab	Nog een pakket voor tabellen over meer pagina's (simpeler dan `longtable`)
tabularx	`Tabular` die automatisch de pagina breedte opvult

... vervolgd op volgende pagina

naam	functie
theorem	Uitbreidingen voor de `theorem` omgevingen
varioref	Intelligente pagina referenties
verbatim	Uitbreiding van de `verbatim` omgeving
wasysym	Diverse nieuwe symbolen
wrapfig	Nog een pakket voor text naast een plaatje
xr	Referenties naar andere LaTeX documenten
xspace	Definieer commando's die geen spaties opslikken
xypic	Commutatieve diagrammen

Appendix E

Postscript

E.1 Postscript als taal

Een Postscript-programma is opgebouwd uit operatoren die worden toegepast op parameters. Er is een groot aantal operatoren standaard aanwezig, en het is ook mogelijk om zelf nieuwe operatoren te definiëren. Omdat ook keuze en herhaling/recursie mogelijk zijn, is Postscript een echte programmeertaal.

Het voeden van parameters aan operatoren gebeurt via een stack. Bij het executeren van een Postscript-programma wordt elk ding dat geen operator is op de stack gezet. Een operator plukt de parameters die hij nodig heeft van de stack. Daardoor moeten alle expressies in *postfix*-notatie worden geschreven: eerst de parameters (die op de stack worden gezet), en dan de operator (die ze er weer af haalt).

Naast de stack wordt een belangrijke rol gespeeld door de *current page*, waarop de tekening wordt opgebouwd, het *current path*, bestaande uit een aantal lijnstukken (die niet noodzakelijkerwijs aan elkaar vast hoeven te zitten), en het *current point*. Het gebruik van al deze zaken wordt gedemonstreerd in het volgende eenvoudige, doch complete Postscript-programma:

```
100 200 moveto
150 230 lineto
150 250 lineto
stroke
showpage
```

De operator `moveto` pakt twee parameters van de stack, en zorgt ervoor dat het *current point* hier komt te staan. De operator `lineto` voegt een lijnstuk van het huidige punt naar het nieuwe gespecificeerde punt toe aan het *path*. Bovendien wordt dit nieuwe punt het huidige punt. De operator `stroke` zorgt ervoor dat het *path* als lijn-tekening op de huidige pagina wordt gezet. De operator `showpage` tenslotte laat de huidige pagina zien (op papier of scherm), en begint met een schone lei.

Behalve de genoemde operatoren die een grafische aktie uitvoeren, zijn er ook operatoren voor het meer conventionele rekenwerk. Ook deze operatoren worden in postfix-notatie opgeschreven. De expressie $1 + 2 * 3 + 4$ ziet er daarom als volgt uit:

```
1 2 3 mul add 4 add
```

Het leren van Postscript bestaat voor een deel uit het handig worden in het gebruik van deze stack-gebaseerde notatie. Verder is het natuurlijk belangrijk om te weten welke operatoren er beschikbaar zijn.

Hieronder volgt een overzicht van een aantal operatoren. De lijst is niet compleet, maar geeft wel een idee van wat er zoal mogelijk is. In de tabellen wordt voor de operator aangegeven wat de parameters zijn, en erachter wat de operator op de stack achterlaat (of – als de parameter 'verbruikt' wordt).

- Operatoren om de stack te manipuleren

x	`pop`	–	verwijder bovenste element
$x\ y$	`exch`	$y\ x$	verwissel bovenste twee elementen
x	`dup`	$x\ x$	verdubbel bovenste element
$x_1 \ldots x_n\ n$	`copy`	$x_1 \ldots x_n\ x_1 \ldots x_n$	kopieer elementen

- Rekenkundige operatoren

$x\ y$	`add`	z	optellen
$x\ y$	`sub`	z	aftrekken
$x\ y$	`mul`	z	vermenigvuldigen
$x\ y$	`div`	z	delen
$x\ y$	`idiv`	z	integer delen
$x\ y$	`mod`	z	rest bij deling
x	`abs`	z	absolute waarde
x	`neg`	z	omgekeerde
x	`sqrt`	z	wortel
x	`sin`	z	sinus (x in graden)
	...		

- Operatoren om een pad op te bouwen

–	`newpath`	–	begin een nieuw pad
–	`currentpoint`	$x\ y$	zet coördinaten huidige punt op stack
$x\ y$	`moveto`	–	verplaats 'huidige punt'
$dx\ dy$	`rmoveto`	–	verplaats 'huidige punt' relatief
$x\ y$	`lineto`	–	trek lijn
$dx\ dy$	`rlineto`	–	trek lijn relatief
–	`closepath`	–	maak pad gesloten
$x\ y\ r\ a\ b$	`arc`	–	cirkelboog naar (x, y), straal r tussen hoek a en b
$x_1\ y_1\ x_2\ y_2\ x_3\ y_3$	`curveto`	–	Bézier curve naar p_3 met p_1 en p_2 als controlepunt

- Operatoren om een pad te gebruiken

–	`stroke`	–	trek lijnen langs het pad
–	`fill`	–	kleur het pad in
x	`setlinewidth`	–	verander de dikte van de lijnen
x	`setgray`	–	verander de grijstint van lijnen en vlakken

E.2 Postscript-verwerkende programma's

De volgende programma's hebben Postscript als invoer en/of als uitvoer:

`a2ps` dit programma maakt van een gewone tekstfile een Postscript-file, waarbij de tekst in twee kolommen is ingedeeld, de pagina's van een header en een nummering worden voorzien, enzovoort. Je hebt dit programma meestal niet nodig omdat het standaard door `laser` wordt gebruikt, maar het kan expliciet worden aangeroepen om extra opties mee te geven (bijvoorbeeld om de header-regel weg te laten, de regels te nummeren, of om hele lange regels weer te geven).

`pstoedit` Dit programma kan postscript vertalen naar diverse andere formaten, o.a. naar het xfig formaat, zodat aanpassingen gemaakt kunnen worden.

`pstops` Met dit programma kan een Postscript-programma worden geconverteerd in een ander Postscript-programma. Er kunnen bijvoorbeeld pagina's mee worden geselecteerd of worden samengevoegd. Een veel gebruikte optie is om twee postscript-pagina's naast elkaar op één vel te zetten. De wat cryptische commandoregel hiervoor is

```
pstops '2:0L@.7(21cm,0)+1L@.7(21cm,14.85cm)' input.ps output.ps
```

Om de output van `dvips` te verwerken moeten de `0` en de `14.85` worden omgewisseld.

`psbook` Rangschikt de pagina's van een Postscript file zodat ze als een boekje afgedrukt kunnen worden.

`psselect` selecteert pagina's uit een Postscript file

`psnup` Drukt meerdere pagina's verkleind op één vel af. De afdrukkwaliteit verslechtert hierdoor. Voor DVI-files is het beter om `dvidvi` te gebruiken.

`epsffit` Maakt een een EPSF file passend voor een gegeven rechthoek

De volgende programma's verbeteren soms Postscript output van andere programma's zodat deze beter te gebruiken is met LaTeX of met de hierbovengenoemde programma's.

`fixfmps` Framemaker documenten

`fixmacps` Macintosh documenten

`fixpsditps` Transcript psdit documenten

`fixpspps` PSPrint PostScript

`fixtpps` Troff Tpscript documenten

`fixwfwps` Word for Windows documenten

`fixwpps` WordPerfect documenten

`fixwwps` Windows Write documenten

Literatuur

[1] L. Lamport: *LaTeX, A Document Preparation System, User's Guide and Reference Manual – 2nd ed.*, Addison-Wesley Publishing Company (1994), ISBN 0-201-52983-1.

[2] M. Goossens, F. Mittelbach, A. Samarin: *The LaTeX Companion*, Addison-Wesley Publishing Company (1994), ISBN 0-201-54199-8.

[3] D. E. Knuth: *The TeXbook*, Deel A van de serie *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9, in de boekhandel verkrijgbaar.

Index