



FACULTEIT LANDBOUWKUNDIGE . . . ERR. . .
ZEUS WERKGROEP INFORMATICA

Academiejaar 2005–2006

EEN INTRODUCTIE TOT HET ZETSYSTEEM \LaTeX

Gaspard LEQUEUX

Promotor: Prof. dr. A. Armagneau

Scriptie voorgedragen tot het behalen van de graad van
GOEROE IN DE \LaTeX OLOGIE

Copyright 2003, 2004, 2005, 2006, 2007, 2008 Gaspard Lequeux. Alle rechten voorbehouden. Dit werk mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm, geluidsband, kleitabletten, elektronische of welke andere wijze ook, onder de volgende voorwaarden:

- Vermelding van de auteur.
- Niet commercieel. Dit vloeit voort uit het feit dat sommige paragrafen overgenomen werden uit ?, wiens werk niet mag gebruikt worden voor commerciële doeleinden.
- Als je dit werk wijzigt en/of verdeelt, moet dit gebeuren onder dezelfde voorwaarden.

De auteur is niet aansprakelijk voor eventuele fouten en onvolledigheden in dit werk.

Versie 0.6.0

Gecompileerd op: 18 maart 2014

Woord vooraf

Deze cursus werd in November 2003 voor het eerst uitgegeven in het kader van een Zeus (Studenten Werkgroep Informatica) initiatief om thesistudenten te helpen hun thesis in een aantrekkelijke vorm te gieten. De tekst is opgevat als een soort minithesis zodat studenten gemakkelijk alles kunnen overnemen. Om die reden zijn de bronbestanden beschikbaar gemaakt op het internet.¹

Hoewel sommige deeltjes van deze cursus expliciet gericht zijn op het maken van een thesis, kan de tekst gebruikt worden als algemene inleiding op L^AT_EX. In 2004 zag ook Prof. Ottoy dit in. Vandaar dat deze cursus nu deel uitmaakt van de lessen informatica in de tweede bachelor bio-ir van de Universiteit Gent. Bij die gelegenheid heeft hij de cursus volledig nagekeken op taalfouten, waarvoor dank.

In 2006 heeft Prof. Dawyndt de cursus ook grondig herlezen en nagekeken, waarvoor dank. Sindsdien wordt deze cursus gebruikt in de lessen Computergebruik gegeven in de eerste bachelor Informatica van de Universiteit Gent.

Voor het schrijven van deze handleiding L^AT_EX, werd gebruik gemaakt van twee werken: A guide to L^AT_EX van ? en Handleiding L^AT_EX van Piet van Oostrum (1996). Uit dit laatste werden zelfs hele paragrafen overgenomen. Daarnaast werd natuurlijk rijkelijk geput uit de documentatie die meegeleverd wordt met L^AT_EX zelf.

Minder belangrijke delen worden in een kleiner lettertype weergegeven. Verder zijn er verschillende voetnoten die verwijzen naar L^AT_EX documentatie op een Debian GNU/Linux systeem. Niemand gebruikt dat natuurlijk. Maar de bedoeling ervan is dat de lezer weet dat de documentatie bestaat, welke bestandsnaam die heeft en waar ongeveer die te vinden is in de L^AT_EX directorystructuur. Het is dan niet zo moeilijk meer om in je favoriete besturingssysteem te zoeken naar de desbetreffende bestandsnaam.

In een standaard MikTeX installatie (dé L^AT_EX distributie voor Windows), zijn de helpfiles van de verschillende L^AT_EX packages te vinden in C:\texmf\doc\latex. In C:\texmf\doc\guides zijn enkele algemene handleidingen te vinden over L^AT_EX.

¹<http://zeus.ugent.be/~gaspard/latex>

Het woord vooraf dient ook om mensen te bedanken:

- *De mensen van Zeus, voor de stimulerende Vrije–Open Source sfeer en het organiseren van lessen hier rond.*
- *Schamper, het studentenblad van de Universiteit Gent, voor het leveren van de promotor van dit werk.*
- *Rudy Gevaert, die in het academiejaar 2001-2002 als eerste een L^AT_EX les gaf.*
- *Geert Vernaeve, voor het eerste contact met L^AT_EX en het C-voorbeeld op bladzijde 36.*
- *Mensen die fouten rapporteerden en/of verbeteringen suggereerden: David De Wolf, Annelies Huyck, Yves Nevelsteen, Stijn Gors, Geert Vernaeve, Michiel Meire, Hendrik Maryns, Olivier Verhoogen, Jean-Pierre Otttoy, Hugo Coolens, Lieven Clement, Andy Peene, Reinout Debergh, Frederik De Schrijver, David van der Ha, Brecht Donckels, Dominique Lebbe, Christopher De Dobbelaere, Paul Vogels, Heidi Vanparys, Stijn Depuydt, Veerle Gevaert, Joke Van Hevele, Katrien De Dauw, Francis Santens, Nicolas Vanden Bossche en Peter Dawyndt.*
- *De (thesis)studenten van het Boerenkot, die in 2003 gevraagd hebben naar deze handleiding.*

L^AT_EX lijkt in het begin moeilijk: alles in tekstmode, geen knopjes, je moet speciale comando's kennen om iets te bereiken ... De eerste dagen zul je inderdaad enkele problemen ondervinden. Zoeken, doorbijten en hulp vragen aan meer ervaren gebruikers zullen ervoor zorgen dat je na enkele weken zelfs je wiskundige redeneringen rechtstreeks in L^AT_EX uitvoert.

Deze handleiding is waarschijnlijk niet foutloos. Het rapporteren van meer dan één fout, zorgt voor je naam in de volgende editie van dit woord vooraf. Ook inhoudelijke opmerkingen zijn steeds welkom.

*Gaspard Lequeux
Gent 9 Augustus 2006*

Inhoudsopgave

1	Inleiding	1
1.1	Hoe het begon	1
1.1.1	TEX	1
1.1.2	L ^A T _E X	1
1.2	Het verwerken van tekst	1
1.3	Voor- en nadelen	3
1.4	Het L ^A T _E X compileerproces	4
1.5	L ^A T _E X installatie en gebruik onder Windows	5
2	De documentstructuur	7
2.1	Preamble	7
2.1.1	Het documenttype	7
2.1.2	Extra functionaliteiten	8
2.1.3	Documentinstellingen, zelfgedefinieerde commando's en omgevingen	10
2.2	Body	12
2.3	Bladspiegel	13
2.3.1	Koptekst en voettekst	13
2.3.2	Bladzijdenummering	15
2.4	Onderverdeling van een document	15
2.4.1	Structuur van een boek	15
2.4.2	Titelpagina	16
2.4.3	Inhoudsopgave	16
2.4.4	Abstract	17
2.4.5	Secties	17
2.4.6	Appendix	18
2.5	Refereren naar andere delen	19
3	Platte tekst	20
3.1	Speciale Tekens	20
3.1.1	L ^A T _E X symbolen	20
3.1.2	Euro	21
3.1.3	Ellipsis (dots)	21
3.1.4	Koppelteken en verbindingstreepjes	21
3.1.5	Speciale letters en accenten	21

3.2	Spaties	22
3.3	Paragrafen	22
3.4	Woordafbreking	24
3.5	Arbitraire witte ruimte	26
3.5.1	Lengte eenheden in L ^A T _E X	26
3.5.2	Horizontale witte ruimte	27
3.5.3	Verticale witte ruimte	28
3.6	Lettertypes	28
3.6.1	Lettergrootte	28
3.6.2	Letterstijlen	30
3.7	Centreren en indentatie	30
3.8	Lijsten	31
3.8.1	Itemize	31
3.8.2	Enumerate	32
3.8.3	Description	32
3.8.4	Geneste lijsten	33
3.8.5	Veranderen van het label	33
3.8.6	Theorem — Stelling	34
3.9	Voetnoten	35
3.10	Kantlijnopmerkingen	35
3.11	Letterlijke tekst	36
3.12	Eenheden	36
4	Boxen, tabellen en figuren	38
4.1	Boxen	38
4.1.1	Links-rechts boxen	38
4.1.2	Parboxen en minipagina's	39
4.1.3	Nesten van boxen	40
4.2	Tabellen	40
4.2.1	Opbouwen van tabellen	40
4.2.2	Stijlparameters voor tabellen	43
4.2.3	Voorbeelden	43
4.3	Figuren	46
4.4	Zwevende tabellen en figuren	48
4.4.1	Zwevend maken	48
4.4.2	Bijschrift	50
4.4.3	Verwijzingen naar tabellen en figuren	51
4.4.4	Enkele voorbeelden	52
4.4.5	Nieuw commando om gemakkelijk <i>floats</i> in te voegen	53
5	Bibliografische verwijzingen	55
5.1	BIB _T E _X , het programma	55
5.2	De bibliografische databank	57
5.2.1	Opbouw van de databank	57

5.2.2	De verschillende items	59
5.2.3	De verschillende eigenschappen	60
5.2.4	Templates	62
5.3	Citeren in de tekst	63
6	Wiskundige formules	65
6.1	Wiskundige omgeving	65
6.1.1	Lange formules: <code>multline</code>	66
6.1.2	Meerdere formules tegelijkertijd: <code>gather</code> , <code>align</code> , <code>eqnarray</code> , <code>array</code> . .	66
6.1.3	Subnummering van vergelijkingen	68
6.1.4	Omgevingen binnen de <code>equation</code> omgeving	68
6.2	De wiskundetaal	70
6.2.1	Exponenten en indices — superscript en subscript	70
6.2.2	Breuken	71
6.2.3	Binomiaalcoëfficiënten	71
6.2.4	Wortels	71
6.2.5	Sommen en integralen	71
6.2.6	Ellipsis	72
6.2.7	Operatornamen	72
6.2.8	Accenten	73
6.2.9	Symbolen	74
6.2.10	Gewone tekst in <i>math mode</i>	74
6.2.11	Haakjes	75
6.2.12	Matrices	76
6.2.13	Witte ruimte	76
6.3	Lettertypes	77
6.3.1	Vette letters	77
6.3.2	Rechte letters	78
6.3.3	Andere letters	78
6.4	Commando's in wiskundige modus	78
6.5	Conventies voor matrices en vectoren	79
6.6	Chemische formules	79
7	Conclusies en perspectieven	80
A	Wiskundige symbolen	81

Hoofdstuk 1

Inleiding

1.1 Hoe het begon

1.1.1 T_EX

T_EX (uitspraak “Tech” als in technéut, kan ook als ‘TeX’ geschreven worden) is een computer-programma van Donald E. ? . Het is speciaal ontworpen voor het zetten en drukken van wiskundige teksten en formules. Een belangrijk kenmerk van T_EX is dat je er zelf nieuwe functies in kan programmeren (zogenaamde macro’s). Het pakket is dus uitbreidbaar. Vele mensen die uitbreidingen geschreven hebben, maken die publiek beschikbaar. Het is dus aangewezen om eerst te kijken wat er al bestaat, vooraleer zelf iets te implementeren.

1.1.2 L^AT_EX

L^AT_EX (uitspraak “Lah-tech”, kan ook als ‘LaTeX’ geschreven worden) is een zogenaamd macro-pakket dat door Leslie ? werd geschreven en gebruik maakt van T_EX. Het stelt de auteur in staat zijn publicaties op eenvoudige wijze en met gebruik van een op voorhand opgegeven structuur, met boekdrukwaliteit te zetten en af te drukken.

Om die speciale zes L^AT_EX letters te krijgen, moet je het commando `\LaTeX` gebruiken. T_EX verkrijg je met het commando `\TeX`.

1.2 Het verwerken van tekst

Het maken van printbare documenten kan op verschillende manieren gebeuren. Enerzijds heb je de wuziwuk¹ tekstverwerkers, waar het merendeel van de mensen mee vertrouwd is, en anderzijds heb je de zogenaamde *markup* tekstprocessing programma’s, zoals L^AT_EX.

¹Wat U Ziet Is Wat U Krijgt

De wuziwuk tekstverwerkers zijn initieel gemakkelijk in gebruik. Je typt en er verschijnt juist hetzelfde als hetgeen op papier zal komen.² Voor korte documenten is dit zeer gemakkelijk. Maar voor lange documenten werkt dit niet meer zo goed: die drie enters die je op het einde van een bladzijde had gezet om de daarop volgende titel op een nieuw blad te krijgen (omdat dat mooi was) blijken na het doorvoeren van een verbetering plots in het midden van een blad te staan. Of de figuren die zo mooi geschikt waren, beginnen rond te zweven, of nog erger: worden vervangen door een rood kruis. Zelfs als dat je bespaard wordt, blijft er nog altijd het probleem van consistentie: bij de ene figuur heeft het bijschrift lettergrootte 10, bij de andere 11.

Vandaar de andere manier: \LaTeX . De opmaak wordt gescheiden van de inhoud. De auteur geeft de structuur aan en de computer maakt aan de hand van de gestructureerde tekst een prettig leesbaar document. Dit kan een printbaar bestand zijn (met `latex`) of een webpagina (met `latex2html`). Met één commando kan je je thesis publiceren op het *www*!

De tekst wordt getypt in een gewone editor. In een editor kun je alleen tekst typen: geen knopjes om vet of cursief te zetten, geen animaties, geen figuren invoegen. Enkel tekst, platte tekst. Dit lijkt archaïsch, omdat de meeste mensen alleen *wordpad* als editor kennen, maar er bestaan veel betere: `vim` is er één van, `emacs` is een andere. Beiden zijn vrij verkrijgbaar op het net. Onder Windows bestaat er zoiets als *winedit* (te betalen) maar ook `vim` kan daar geïnstalleerd worden. Dé \LaTeX -editor onder Windows is echter TeXnicCenter.³

Platte tekst. Da's niet veel. Hoe krijg je dan titeltjes, paginanummering, opmaak? Wel, er bestaan van die regels om goede teksten te maken. Die kun je vanbuiten leren en dan toepassen in een wuziwuk programma. Maar die regeltjes toepassen kan je ook laten doen door de computer. In \LaTeX zeg je: dit is een hoofdstuk, dit een titel, dit is een belangrijk woord, dit is letterlijke tekst, in de buurt van deze paragraaf moet deze figuur komen, dit is een Latijnse naam voor een bacterie. Allemaal in de teksteditor (hoe je dat doet, zien we later). Je tekstbestand geef je dan te eten aan \LaTeX . Op basis van het tekstdocument dat je gemaakt hebt, maakt \LaTeX een mooi printbaar document. Waar je zegt: dit is de titel van een nieuw hoofdstuk, gaat \LaTeX een nieuwe bladzijde beginnen, "Hoofdstuk x" schrijven (waarbij x het juiste nummer van het hoofdstuk is), op een nieuwe lijn gaan staan en dan je hoofdstuk titel in een gepast lettertype plaatsen.

Hoe geef je je wensen te kennen aan \LaTeX ? Via speciale tekstsequenties natuurlijk. Aanduiden dat een nieuw hoofdstuk begint, gaat als volgt:

```
\chapter{Een nieuw hoofdstuk}
```

²Althans, dit is de theorie. In de praktijk durft dit wel eens lelijk tegen te vallen.

³www.texniccenter.org

Die *backslash* maakt de \LaTeX compiler duidelijk dat wat volgt geen gewone tekst is, maar een commando dat betrekking heeft op hetgeen tussen de accolades staat.

Iets over opmaakontwerp

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.

Voorbeelden: De lettergrootte en nummering van titels van hoofdstukken en paragrafen moet zo gekozen worden, dat de structuur 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 wuziwuk-pakketten maken auteurs soms mooie, maar slecht leesbare documenten. Een veel gemaakte fout is bijvoorbeeld het gebruiken van te veel verschillende lettertypes of het onderlijnen van titeltjes — een typografische ramp. \LaTeX voorkomt typografische fouten, omdat het de auteur dwingt de logische structuur van een tekst aan te geven. \LaTeX gebruikt dan automatisch de meest geschikte opmaak.

1.3 Voor- en nadelen

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

- De gebruiker hoeft maar een paar, gemakkelijk te begrijpen, commando's te leren. Deze commando's betreffen alleen de logische structuur van het document, de gebruiker hoeft zich nauwelijks bezig te houden met de technische details.
- Complexe structuren zoals voetnoten, literatuuropgaven, inhoudsopgaven, tabellen, formules etc. en zelfs eenvoudige tekeningen, kunnen zonder al te veel moeilijkheden gemaakt worden. Het zetten van complexe wiskundige formules is bijzonder gemakkelijk in \LaTeX .
- Het geproduceerde document is consistent: alle titels zien er gelijkaardig uit (zelfde lettertype, zelfde letterstijl), de bijschriften bij tabellen en figuren hebben allemaal dezelfde opmaak en het citeren gebeurt altijd op dezelfde manier (inderdaad, wie houdt er zich in deze tijd van krachtige computers nog bezig met het handmatig opstellen van een referentielijst?).
- Het is beschikbaar voor bijna alle computersystemen, en gratis.

- \LaTeX is zeer stabiel. Crashes of corrupte bestanden zijn ongekend. \LaTeX bronbestanden bevatten eigenlijk gewone tekst. Zij nemen dan ook weinig plaats in en kunnen gemakkelijk gebackupt worden op diskette.
- Vele mensen gebruiken \LaTeX . Er bestaan dus een hele hoop pakketten (extensies) voor \LaTeX die zeer gespecialiseerde dingen doen. Op www.ctan.org⁴ zijn alle mogelijke pakketten verzameld. De meesten hiervan zijn waarschijnlijk echter al op je systeem geïnstalleerd.

\LaTeX heeft natuurlijk ook nadelen:

- Binnen de door \LaTeX ondersteunde opmaakstijlen kunnen weliswaar enige parameters gevarieerd worden, maar ingrijpende afwijkingen van de gebruikte opmaakstijl kunnen slechts met veel moeite tot stand gebracht worden.
- Het intypen van \LaTeX commando's is ingewikkelder dan het aanbrengen van opmaak 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.

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.

1.4 Het \LaTeX compileerproces

\LaTeX leest dus een tekstbestand in, en aan de hand daarvan worden een aantal bestanden gegenereerd waarvan vooral één bestand ons interesseert, namelijk het printbaar document. Het verwerken van een \LaTeX bronbestand door de \LaTeX compiler gebeurt met het commando `latex thesis.tex`, waarbij *thesis.tex* het \LaTeX bestand is dat we willen compileren.

Oorspronkelijk was het document dat uit de latex-compiler kwam, een **dvi**-bestand (van *DeVice Independent*). Dit bestand is onafhankelijk van een printer. Om het naar een printer te kunnen sturen, moet het met een zogenaamde *printer driver* worden omgezet naar een bestand geschikt voor die bepaalde printer. Een veel gebruikt printerformaat is het **ps** of PostScript formaat. Het nieuwere formaat om printklare bestanden op te slaan en door te geven is **pdf** (*Portable Document Format*).

Het `latex` commando genereert een *dvi* bestand. Dit kan worden omgezet naar een *ps* bestand met het commando `dvips thesis.dvi -o thesis.ps` (of korter: `dvips thesis -o`). Voor het omzetten van een *dvi* bestand naar een *pdf*-bestand bestaan er twee conversieprogramma's: `dvipdf` en `dvipdfm`. De syntax is iets anders dan bij `dvips`, namelijk: `dvipdf thesis.dvi` en er wordt een *thesis.pdf* aangemaakt.

⁴Comprehensive T_EX Archive Network

Maar het is beter om rechtstreeks met pdf te werken. Met `pdflatex` maak je rechtstreeks een *pdf*-bestand aan. De syntax is dezelfde als voor `latex`:

```
pdflatex thesis.tex
```

Dit document werd gecompileerd met `pdflatex`. We zullen verder ook alles uitleggen voor `pdflatex`. Dus wanneer gesproken wordt over het laten compileren door de latex compiler, bedoelen we eigenlijk de `pdflatex` compiler.

Naast een dvi of pdf bestand, genereert \LaTeX nog twee (of soms meer) andere bestanden: een **aux** en een **log**-bestand. In het *log*-bestand komen alle commentaren, die je op het scherm ziet voorbij rollen. Het *aux*-bestand is een help-bestand voor \LaTeX zelf. Als je bijvoorbeeld in hoofdstuk 1 verwijst (verwijzingen worden behandeld in sectie 2.5) naar een bladzijde in hoofdstuk 2, kan \LaTeX tijdens het verwerken van hoofdstuk 1 niet weten welke bladzijde dit is. Wanneer echter hoofdstuk 2 wordt verwerkt, gaat \LaTeX in de *aux*-file de juiste bladzijde bijhouden en de volgende keer dat je de latex-compiler loslaat op je document, zal de juiste bladzijde ingevoerd worden. Zie sectie 2.5 op bladzijde 19 voor meer uitleg. Het is om die reden dat een document finaal minstens twee keer door de latex-compiler moet gejaagd worden (maar we zullen in de loop van deze cursus zien dat het tot vier keer nodig kan zijn).

Voor de mensen die met Windows werken en bang zijn van een *command line*: geen nood. TeXnicCenter, een goede editor voor \LaTeX onder Windows, beschikt over een aantal knopjes waar je kan op klikken en die dan al deze zaken voor jou uitvoeren.

1.5 \LaTeX installatie en gebruik onder Windows

Onder GNU/Linux wordt \LaTeX bij de meeste distributies standaard meegeleverd (het zogenaamde *texlive* pakket). Onder Windows is het iets moeilijker.

- Dé \LaTeX distributie voor Windows is MikTeX.⁵ Op deze website staat uitgelegd wat we moeten doen om \LaTeX te installeren op onze computer. Na de installatie hebben we ‘slechts’ een \LaTeX compiler, geen teksteditor, noch een handig knopje waarop we kunnen klikken om ons *tex*-bestand om te zetten naar een *pdf*-bestand. Slechts de DOS *command line* is beschikbaar en *notepad* voor het bewerken van onze *tex*-bestanden.
- Om het resultaat te bekijken, moeten we (althans voor `pdflatex`), een pdf-lezer installeren. De beste hiervoor is Acrobat Reader van Adobe.⁶ Meestal is die echter al geïnstalleerd.

⁵www.miktex.org

⁶www.adobe.com

- Om plezier te werken, installeren we TeXnicCenter.⁷ Dit is een geïntegreerde omgeving voor het maken van L^AT_EX documenten. Je hebt van die handige menu's waarin je kan kiezen uit verschillende commando's, zodat je ze niet allemaal moet onthouden. Het is best om eerst Acrobat Reader te installeren, zodat TeXnicCenter hiermee kan linken tijdens de installatie.

Omdat L^AT_EX gebaseerd is op redelijk oude computercode, kan de latexcompiler moeilijk of niet omgaan met spaties (en andere rare tekens) in bestandsnamen of directories. Installeer MikTeX dus niet in *C:\Program Files* maar bijvoorbeeld wel in *C:\miktex*. Gebruik ook nooit spaties in je *tex*-bestandsnamen. Dus NIET opslaan in *C:\Mijn Documenten* maar wel in *C:\latexbestanden* of iets dergelijks. Dit zal veel kopzorgen vermijden.

⁷www.texniccenter.org

Hoofdstuk 2

De documentstructuur

Elk L^AT_EX document bestaat uit twee delen: een **preamble** en een **body**. In de *preamble* komen verschillende opties die voor het hele document gelden. In de *body* komt de eigenlijke tekst van het document. De *preamble* begint bij het begin van het bestand en eindigt bij het L^AT_EX commando `\begin{document}`. Dit is ook het begin van de *body*. De *body* eindigt met `\end{document}`. Een minimaal L^AT_EX bestand ziet er dus als volgt uit:

```
\documentclass{article}
\begin{document}
In theorie zijn theorie en praktijk gelijk, maar in praktijk zijn ze verschillend.
\end{document}
```

2.1 Preamble

In de *preamble* komen de verschillende instellingen die moeten gelden voor het hele document, de verschillende extra functionaliteiten die moeten toegevoegd worden aan het basis L^AT_EX pakket en eventueel zelfgemaakte commando's.

2.1.1 Het documenttype

Veel van de instellingen voor de opmaak zijn standaard voorgeprogrammeerd voor bepaalde types van documenten. Deze zijn: *book* om een boek te schrijven, *article* om een artikel te schrijven, *report* om een rapport te schrijven en *letter* om een brief te schrijven. Dit is de zogenaamde *documentclass*. Hierbij kunnen ook opties meegegeven worden. Belangrijke opties zijn de grootte van het letters, het papierformaat en of er later dubbelzijdig of enkelzijdig zal afgeprint worden. Voor dit document werd de volgende *documentclass* declaratie gebruikt:

```
\documentclass[11pt,a4paper,oneside]{book}
```

Tussen vierkante haakjes worden de opties meegegeven en tussen accolades komt de *document-class*. Dingen die ingegeven worden tussen vierkante haakjes, zijn nooit verplicht. Indien hier geen opties worden meegegeven, dus `\documentclass{book}`, worden letters van 10 punten groot gebruikt en gaat L^AT_EX ervan uit dat het boek dubbelzijdig wordt afgeprint.

Het leuke aan L^AT_EX is dat die 11pt ervoor zorgt dat niet alleen de tekst in de juiste grootte wordt geplaatst, maar dat ook de titels evenredig worden aangepast. De mogelijke keuzes zijn 10pt, 11pt en 12pt. L^AT_EX kiest standaard altijd voor 10pt. Maar voor een thesis is het aan te raden om 11pt te gebruiken. Dit is iets groter dan de 11pt van die andere populaire tekstverwerker.

De optie `a4paper` zorgt ervoor dat de grootte van de gegenereerde bladzijden A4 is. En de optie `oneside`, tenslotte, geeft aan dat het boek enkelzijdig afgeprint zal worden. Dit is verplicht voor thesissen.

Daarnaast kunnen er nog meer opties meegegeven worden. Ze worden hier kort overlopen, maar zijn normaalgezien niet nodig voor het schrijven van een thesis:

- `onecolumn`, `twocolumn`: om één of twee kolommen tekst te hebben.
- `oneside`, `twoside`: enkelzijdig of dubbelzijdig afdrukken.
- `openright`, `openany`: bij een boek begint een nieuw hoofdstuk normaalgezien op een rechter, oneven bladzijde. Indien nodig wordt een witte bladzijde ingevoegd. Met `openany` kan ervoor gezorgd worden dat hoofdstukken gewoon beginnen op de volgende bladzijde. Deze optie heeft natuurlijk alleen zin als ook de optie `twoside` wordt meegegeven.
- `leqno`: zorgt ervoor dat wiskundige formulenummers links worden geplaatst in plaats van rechts.
- `fleqn`: wiskundige formules worden links geplaatst in plaats van gecentreerd.
- `draft`: als L^AT_EX niet op tijd een nieuwe lijn begint (en dus in de rechtermarge schrijft), wordt dit sterk in de verf gezet met een dikke zwarte markering, zodat het goed wordt opgemerkt bij het herlezen en er nog iets kan aan gedaan worden. Die optie moet natuurlijk verwijderd worden bij het finaal processen van het document.

2.1.2 Extra functionaliteiten

Extra functionaliteiten kunnen aan L^AT_EX toegevoegd worden met behulp van zogenaamde pakketten (*packages*). Dit gebeurt door in de *preamble* dat pakket op te roepen met de volgende syntax:

```
\usepackage[opties]{package_naam}
```

Interessante basispakketten voor de thesis zijn:

<code>\usepackage{a4wide}</code>	% Iets meer tekst op een bladzijde
<code>\usepackage[dutch]{babel}</code>	% Voor nederlandstalige hyphenatie
<code>\usepackage{amsmath}</code>	% Uitgebreide wiskundige mogelijkheden
<code>\usepackage{url}</code>	% Om url's te verwerken
<code>\usepackage{graphicx}</code>	% Om figuren te kunnen verwerken
<code>\usepackage[latin1]{inputenc}</code>	% Om niet ascii karakters te kunnen typen
<code>\usepackage[small,bf,hang]{caption}</code>	% Om de captions wat te verbeteren

a4wide : Om meer tekst op een A4 blad te krijgen.

babel : Zorgt voor de juiste taal in het document. De optie **dutch** zorgt ervoor dat alles aangepast wordt aan de Nederlandse taal: de titels (“Hoofdstuk X”) en de hyphenatie of woordafbreking. \LaTeX bepaalt zelf wanneer een woord afgebroken moet worden om een nieuwe lijn te beginnen.

amsmath : Is een uitbreiding van \LaTeX door de *American Mathematical Society*. Heb je nodig als je een beetje formules in je thesis hebt.

url : Deze uitbreiding zorgt voor een mooie opmaak van webadressen en moet als volgt gebruikt worden: `\url{http://zeus.ugent.be/}` geeft `http://zeus.ugent.be/`.

graphicx : Is een belangrijk pakket om figuren te kunnen invoeren in je document.

inputenc : De \LaTeX compiler verstaat standaard alleen ascii tekst, dus geen accenten en zo. Accenten worden ingevoerd als volgt: `\"e\'e\'e\^e` geeft `ëëë`. Dus je moet het gewenste accent ingeven via een gelijkaardig symbool gecombineerd met een backslash vòòr dat symbool. Niet altijd zeer handig. Met het pakket **inputenc** mag een geaccentueerd karakter in het bronbestand voorkomen. Dus `ëëëë` mag. De optie **latin1** moet erbij om de lettercodering toe te kennen. Zet die optie er gewoon bij, het werkt.

caption¹ : Dient om de verklarende tekst bij tabellen en figuren in een andere vorm te gieten. De verklarende tekst wordt iets kleiner weergegeven dan de lopende tekst (**small**); de titel “Tabel X:” wordt in het vet gezet (**bf** van *Bold Face*); en de verklarende tekst springt in ten opzichte van de linkermarge (**hang**).

De uitleg over de verschillende opties hierboven is zeer summier. Elk pakket komt meestal met een eigen help-bestand dat ergens op je systeem staat. Het help-bestand voor **babel**,

¹Vroeger moest **caption2** gebruikt worden omdat dat een verbetering was tegenover **caption**. Maar tegenwoordig zijn die verbeteringen opgenomen in **caption** en is **caption2** verouderd.

bijvoorbeeld, is 50 pagina's lang. De manier om dat bestand te vinden is door eens te zoeken naar `pakket-naam.dvi`, `pakket-naam.ps` of `pakket-naam.pdf`.

De hierboven opgesomde pakketten zijn maar een kleine deelverzameling van alles wat er in een \LaTeX systeem beschikbaar is. In het vervolg van deze cursus zullen we nog enkele nieuwe pakketten zien voor meer gespecialiseerde dingen.

2.1.3 Documentinstellingen, zelfgedefinieerde commando's en omgevingen

Documentinstellingen

Documentinstellingen worden ook in de *preamble* meegegeven. Enkele voorbeelden, gebruikt voor dit document:

```
\setlength{\parindent}{0cm}           % Niet inspringen bij eerste lijn paragraaf
\renewcommand{\baselinestretch}{1.2} % De interlinie afstand wat vergroten.
```

Voor de uitleg over `\parindent` wordt verwezen naar sectie 3.3. Door `\baselinestretch` een andere waarde te geven, kan de interlinie afstand veranderd worden. Dit document werd opgemaakt met `baselinestretch` gelijk aan 1.2.

Commando's

Met bepaalde commando's kan de tekst waarop het commando betrekking heeft, gewijzigd worden. Wanneer je bijvoorbeeld `\emph{E. coli}` in je brontekst hebt staan, wordt dit in het finale document weergegeven als: *E. coli* (`\emph` komt van *emphasis*, om tekst te benadrukken; dat benadrukken gebeurt meestal door de tekst cursief te zetten als de omliggende tekst niet cursief is en omgekeerd).

De kracht van \LaTeX laat zich pas ten volle kennen, als je gebruik maakt van de mogelijkheid om zelf commando's te definiëren. Nieuwe commando's worden meestal in de *preamble* gedefinieerd, maar kunnen ook in de *body* aangemaakt worden. Het maken van nieuwe commando's gebeurt als volgt:

```
\newcommand{\kort}{laaaaang}
\newcommand{\vet}[1]{\textbf{#1}}
\newcommand{\wissel}[2]{#2 #1}
\newcommand{\bier}[2][leeg]{\textit{#1} \textbf{#2}}
\newcommand{\naam_commando}[aantal_args][standaardwaarde eerste arg]{wat het doet}
```

Dus zoals te zien, kan `\newcommand` tot vier argumenten slikken. Het eerste argument is verplicht (vandaar de accolades) en bevat de naam van het nieuwe commando (vergeet de *backslash* niet!). Het tweede argument is optioneel (vandaar de vierkante haken) en bevat het aantal argumenten dat verwacht wordt door het nieuwe commando. Wanneer dit niet wordt

meegegeven, gaat L^AT_EX ervan uit dat het nieuwe commando nul argumenten verwacht. Het derde argument van `\newcommand` is ook optioneel en bevat een standaardwaarde voor het eerste argument van het nieuwe commando. In het vierde argument van `\newcommand` staat tenslotte wat het nieuwe commando juist moet doen. Zoals te zien is aan de accolades, is dit argument verplicht. Met `#1`, `#2` ... kunnen de verschillende argumenten van het nieuwe commando afgeprint worden.

Het eerste voorbeeld schrijft `laaaaang` wanneer je `\kort` typt. Het tweede voorbeeld verwacht een extra argument en zet dat in het vet: `\vet{bol}` geeft **bol**. Het derde voorbeeld wisselt de twee argumenten om: `\wissel{sterk}{ijzer}` geeft ijzersterk. We zien dat de twee argumenten van `\wissel` tussen accolades staan. In het vierde voorbeeld wordt het eerste argument *italic* gezet en het tweede wordt in het vet geplaatst. We kunnen echter het eerste argument laten vallen. Standaard wordt dan ‘leeg’ afgeprint: `\bier{vat}` geeft *leeg* **vat** terwijl `\bier[vol]{vat}` *vol* **vat** geeft. Merk op dat `vol` tussen vierkante haakjes staat en niet meer tussen accolades: het is namelijk een optioneel argument.

Er blijft echter een probleem: als we `\kort en goed` schrijven krijgen we ‘laaaaangen goed’. De ‘en’ plakt vast aan de `laaaaang`. Om die ‘en’ los te weken moeten we een harde spatie gebruiken: `\kort\ en goed`, wat dan het gewenste ‘laaaaang en goed’ geeft. Maar elke keer die *backslash* achter een commando, dat is vervelend. Een oplossing zou kunnen zijn om die harde spatie in het commando zelf te steken: `\newcommand{\kort}{laaaaang\ }`. Maar wat dan met een leesteken: `\kort?` geeft ‘laaaaang ?’ Geen goede oplossing dus.

Wat we eigenlijk wensen te bekommen, is dat L^AT_EX zelf uitzoekt wanneer er een spatie nodig is en wanneer niet. Er bestaat hiervoor een pakket, `xspace`, dat dat doet. Om het te gebruiken moet ergens in de *preamble* de volgende lijn voorkomen:

```
\usepackage{xspace}           % Magische spaties na een commando
```

Op het einde van elke commandodefinitie geven we het commando `\xspace` waardoor onze zelfgemaakte commando’s werken zoals we intuïtief zouden verwachten. Het korte voorbeeld van hierboven aanpassen tot `\newcommand{\kort}{laaaaang\xspace}` en `\kort en goed \kort!` geeft nu ‘laaaaang en goed laaaaang!’

Wanneer we een bestaand commando willen herdefiniëren, moeten we gebruik maken van `\renewcommand`. De syntax is volledig gelijk aan die van `\newcommand`.

Wanneer we niet weten of een commando al gedefinieerd is, kunnen we niet weten of we `\renewcommand` of `\newcommand` moeten gebruiken. We kunnen dan gebruik maken van `\providecommand` die dezelfde syntax heeft als `\newcommand`. Wanneer het commando al bestaat, wordt de nieuwe definitie niet doorgevoerd en wordt de oude definitie behouden. Het omgekeerde effect (de huidige definitie overschrijven met de nieuwe) wordt bereikt door eerst `\providecommand` op te roepen en dan — we zijn

namelijk zeker dat het commando nu bestaat, maar we weten niet of het niet een oude versie is, daar `\providecommand` de oude versie nooit overschrijft — `\renewcommand`.

Omgevingen

Door ergens een `\begin{omgeving}` te geven, ga je in een bepaalde omgeving, waar een aantal dingen ‘aan’ staan. Bijvoorbeeld `\begin{equation}` gaat in *math mode*, zodat je formules kan invoeren (zie hoofdstuk 6). Met `\end{omgeving}` ga je weer weg uit je omgeving. Het is mogelijk om zelf omgevingen te definiëren in de *preamble*. Dit gebeurt met het commando:

```
\newenvironment{naam_omgeving}[aantal_arg][standaardwaarde eerste arg]{beginacties
}{eindacties}
```

Hierbij is `naam_omgeving` de naam van de nieuwe omgeving, `aantal_arg` en `standaardwaarde eerste arg` hebben juist dezelfde betekenis als bij het maken van een commando, `beginacties` zijn commando's die moeten uitgevoerd worden in het begin en `eindacties` zijn commando's die moeten uitgevoerd worden wanneer je de omgeving afsluit. Ook hier is er een `\renewenvironment` commando beschikbaar.

In dit document zijn er stukken tekst die kleiner gezet zijn dan andere. Het zou moeten opvallen dat die delen minder belangrijk zijn om de basis van L^AT_EX onder de knie te krijgen. Dit kleiner krijgen wordt verkregen door de tekst in de volgende omgeving te zetten:

```
\newenvironment{MinderBelangrijk}{\small}{}
```

Deze omgeving wordt gestart met `\begin{MinderBelangrijk}`. Bij de start wordt het commando `\small`. Hierdoor is de daarop volgende tekst kleiner. Bij het afsluiten van de omgeving met `\end{MinderBelangrijk}` moet er niets extra gebeuren.

2.2 Body

Zoals vermeld op bladzijde 7, begint de body met een `\begin{document}` en eindigt ze met een `\end{document}`. Het is hier dat de eigenlijke tekst van het document komt. Deze manier om een tekst te omsluiten (namelijk tussen een `\begin{iets}` en een `\end{iets}` zetten), noemt men de tekst in een **omgeving** zetten. Dus alle tekst wordt geplaatst in de omgeving ‘document’. Je kan dat zien als een soort ‘open de haakjes, sluit de haakjes’.

Commando's werken niet met omgevingen, zij werken met accolades: `\textit{gekrulde haakjes}` zorgt ervoor dat alleen ‘*gekrulde haakjes*’ cursief (*italic*) staat.

Wanneer je een groot document schrijft, zoals een thesis, is het gebruikelijk om niet alle tekst in hetzelfde bestand te steken. De inhoud van een ander bestand kan in een L^AT_EX document ingelezen worden met het commando:

```
\input{thesis-hfdstk01}
```

Hierbij is *thesis-hfdstk01.tex* het *tex*-bestand dat zal ingevoegd worden in het hoofddocument. Het effect van het `\input` commando is hetzelfde als wanneer de inhoud van *thesis-hfdstk01.tex* getypt zou worden op de plaats waar het commando gegeven wordt. Het commando mag op gelijk welke plaats in het document voorkomen. Ook in de *preamble*. Het mag genest worden: een bestand dat via `\input` ingelezen wordt, mag op zijn beurt een `\input` commando bevatten. Bijvoorbeeld al je hoofdstukken in aparte bestanden en in de hoofdstukken worden tabellen ingelezen die op hun beurt in aparte bestanden zitten. Op die manier hou je overzicht over je document.

Extra bestanden kunnen ook ingevoerd worden met het commando `\include{thesis-hfstk01}`. Dit commando zorgt ervoor dat ieder geïncludeerd bestand apart door de *latex* compiler wordt gecompileerd. Met het commando `\includeonly{bestand1, bestand2, bestand4}` in de *preamble* kan geselecteerd worden welke include-bestanden daadwerkelijk ingevoegd moeten worden. Includecommando's mogen niet genest worden. Best is om met `\input` te werken.

2.3 Bladspiegel

2.3.1 Koptekst en voettekst

Het basisformaat van de bladzijden wordt bepaald door de documentclass. Dit kan echter gewijzigd worden met behulp van het commando `\pagestyle{stijl}`, waarbij *stijl* één van de volgende woorden is:

plain De koptekst is leeg, en de paginanummers staan gecentreerd onderaan. Dit is standaard bij *article* en *report*.

empty Noch koptekst, noch voettekst. Er worden geen paginanummers afgedrukt.

headings De koptekst bevat de titel van het lopende hoofdstuk, samen met het paginanummer. Dit is de standaard bij *book*.

myheadings Hetzelfde als *headings* maar de titels in de koptekst moeten handmatig ingegeven worden met behulp van de commando's:

```
\markright{Koptekst}
\markboth{Koptekst van links}{Koptekst van rechts}
```

Hierbij wordt `\markright` gebruikt voor enkelzijdige documenten, of dubbelzijdige documenten waarbij we alleen een koptekst rechts wensen. Om rechts en links een verschillende koptekst te krijgen in een dubbelzijdig document, wordt `\markboth` gebruikt. Deze kopteksten blijven behouden totdat ze weer veranderd worden ergens in het document.

Het commando `\pagestyle` wordt meestal geplaatst in de *preamble* (als het al gegeven wordt). De commando's (`\markright` en `\markboth` worden natuurlijk gegeven daar waar de koptekst moet veranderen). Om ergens één pagina midden in het document te veranderen, kan gebruik gemaakt worden van `\thispagestyle{stijl}`, bijvoorbeeld als de bladzijdennummering even niet wenselijk is, kan `\thispagestyle{empty}` gegeven worden.

Voor een boek ziet de koptekst er standaard als volgt uit:

HOOFDSTUK 2. EEN VOORBEELD

14

Alles wordt dus in hoofdletters gezet. Dit is niet altijd zeer handig wanneer je iets langere titels hebt: zij kunnen niet op één lijn en overschrijden de rechterkantlijn. Hier kan echter aan verholpen worden met het pakket **fancyhdr**. Dit pakket laat toe om de kop- en voetteksten volledig te personaliseren (onder andere op elke bladzijde een iets ander beeldje in de kop, zodat wanneer je zeer snel bladert, je een soort filmpje krijgt). We gaan hier echter geen volledige beschrijving geven van **fancyhdr**, daar dit pakket een zeer goed help-bestand bevat: *fancyhdr.dvi*². We geven juist mee wat er moet gebeuren om een koptekst te verkrijgen zoals in dit document.

Vooreerst wordt het pakket geladen door in de *preamble* de volgende lijn te zetten:

```
\usepackage{fancyhdr} % Voor fancy headers en footers.
```

Om het pakket nu effectief te gebruiken, wordt het volgende in de *preamble* opgenomen:

```
\pagestyle{fancy} % De bladzijdestijl
\fancyhf{} % Resetten fancy settings.
\renewcommand{\headrulewidth}{0pt} % Geen lijn onder de header.
\fancyhf[HL]{\nouppercase{\textit{\leftmark}}} % Links in header het hoofdstuk,
\fancyhf[HR]{\thepage} % Rechts het paginanummer.
```

Voor diegenen die er het fijne van willen weten: in `\leftmark` wordt de koptekst die op linkerpagina's moet komen, bewaard. In het geval van een boek is dit het lopende hoofdstuk. In `\rightmark` wordt de koptekst die op rechterpagina's moet komen, bewaard. Voor een boek, is dit de lopende *section*. Wij drukken enkelzijdig af, en willen op elke bladzijde in de koptekst het lopende hoofdstuk. Zodus maken we geen gebruik van `\rightmark`.

Met `\fancyhead[HL]{\nouppercase{\textit{\leftmark}}}` wordt de hoofdstuktitel effectief links (vandaar die L) in de koptekst (vandaar die H) gezet waarbij `\nouppercase` ervoor zorgt dat niet alles in hoofdletters staat. Met `\textit` wordt alles cursief gezet. In `\thepage` zit het paginanummer van de huidige pagina.

²Op een Debian systeem: `/usr/share/doc/texmf/latex/fancyhdr/fancyhdr.dvi.gz`

Sommigen hebben liever het paginanummer onderaan gecentreerd. Om dit te bereiken moet in het hiervoor gegeven voorbeeld `\fancyhf[HR]{\thepage}` vervangen worden door `\fancyhf[FC]{\thepage}` (FC staat voor *Footer* en geCentreerd).

2.3.2 Bladzijdenummering

De declaratie die de paginanummeringsstijl bepaalt is:

```
\pagenumbering{num-stijl}
```

Hierbij is num-stijl één van de volgende mogelijkheden:

- arabic** Voor Arabische cijfers (standaard).
- roman** Voor Romeinse cijfers (kleine letters).
- Roman** Voor Romeinse cijfers (hoofdletters).
- alph** Voor letternummering (kleine letters).
- Alph** Voor letternummering (hoofdletters).

Wanneer dit commando wordt gegeven, wordt het tellen van pagina's opnieuw begonnen (je kunt dus niet zomaar veranderen midden in je document, de telling begint dan namelijk terug vanaf 1). Dit commando wordt daarom best gegeven in de *preamble*.

De bladzijdenummering kan begonnen worden op een andere waarde dan 1 met behulp van het commando `\setcounter{page}{n}`, waarbij **n** het gewenste begingetal is.

2.4 Onderverdeling van een document

2.4.1 Structuur van een boek

Er zijn enkele simpele commando's om een goede boekstructuur te creëren (alleen in de documentclass *book*):

```
\frontmatter
Hier komen dingen zoals het voorwoord en de inhoudsopgave.
\mainmatter
Het grootste deel van het document komt hier.
\backmatter
Bibliografie, index... komen hier.
```

Het `\frontmatter` commando zorgt voor Romeinse paginanummering en ongenummerde hoofdstukken. Met `\mainmatter` wordt de paginanummering terug op 1 gezet en Arabisch. De hoofdstuknummering wordt geactiveerd. Deze hoofdstuknummering wordt weer gedesactiveerd met `\backmatter`. Het is gebruikelijk om appendices nog in de `\mainmatter` op te nemen (zie sectie 2.4.6). Sommige faculteiten hebben ook de (rare) gewoonte om de bibliografie vòòr de appendices te steken, dus ook in de `\mainmatter`, na het laatste hoofdstuk.

2.4.2 Titelpagina

Een titelpagina kan op twee manieren aangemaakt worden. Een eerste manier is manueel:

```
\begin{titlepage}  
Titel tekst.  
\end{titlepage}
```

Deze manier werd gebruikt voor dit document. Het is de meest soepele manier, maar vraagt werk van de auteur om alles goed te zetten.

We kunnen het werk echter ook door L^AT_EX laten doen. Dat gebeurt met de volgende commando's:

```
\title{Titel van ons werk}  
\author{auteur1\Adres1 \and auteur2\adres2}  
\date{29 Februari 1998}  
\maketitle
```

Op de plaats waar we `\maketitle` geven, bouwt de L^AT_EX compiler de titelpagina op met behulp van de gegevens die te vinden zijn in `\title`, `\author` en `\date`. In `\title` steken we de titel van ons werk, in `\date` de datum. Indien het commando `\date` niet wordt gegeven, verschijnt de datum waarop het document gecompileerd werd. In `\author` komen de namen en adressen van de auteurs. De verschillende auteursgegevens worden gescheiden door een `\and` (vergeet de backslash niet). Een nieuwe lijn beginnen, gebeurt met `\\` (zie sectie 3.3).

2.4.3 Inhoudsopgave

De inhoudsopgave wordt geplaatst waar het volgende commando wordt gegeven:

```
\tableofcontents
```

De inhoudsopgave wordt normaalgezien geplaatst na de titel, het woord vooraf en de samenvatting. Als je een inhoudstafel wenst in het begin van het document, kan L^AT_EX onmogelijk weten wat daarin moet staan voordat het hele document verwerkt is. Maar dan is het te laat om die inhoudsopgave te plaatsen. Tijdens het compileren schrijft L^AT_EX echter de verschillende lijnen die in de inhoudsopgave moeten komen, weg in een bestand met de extensie *toc*. Wanneer ergens in het document naar de inhoudsopgave gevraagd wordt, leest L^AT_EX die in vanuit het *toc*-bestand. Vandaar dat het voor de finale versie nodig is om je thesis drie keer door de compiler te jagen: de eerste keer wordt een inhoudsopgave in het *toc* bestand aangemaakt. Deze eerste inhoudsopgave is niet correct omdat de bladzijdenummering niet klopt (de inhoudsopgave neemt immers ook bladzijden in en die zijn nog niet verrekend). De tweede keer wordt die foute inhoudsopgave in het pdf document ingebracht. Tegelijkertijd

wordt een juiste inhoudsopgave naar het *toc*-bestand weggeschreven. De derde keer wordt de juiste inhoudsopgave in het *pdf*-bestand ingebracht.

Om te bepalen wat er in de inhoudsopgave moet komen, kan het volgende commando in de *preamble* geplaatst worden:

```
\setcounter{tocdepth}{sectie-diepte}
```

Hierbij is ‘sectie-diepte’ 0 om alleen de *chapters* in de inhoudsopgave op te nemen, 1 om ook de *sections* op te nemen, 2 voor de *subsections*, en 3 voor de *subsubsections*. Standaard is dit voor een boek gelijk aan 2.

Het is mogelijk om eigenhandig lijnen aan de inhoudsopgave toe te voegen, bijvoorbeeld wanneer de stervorm van de *chapter*, *section* ... gebruikt wordt (deze zet namelijk niets in de inhoudsopgave):

```
\section*{Een niet genummerde sectie}
\addcontentsline{toc}{section}{Een niet genummerde sectie}
```

In de inhoudsopgave verschijnt dan de lijn ‘Een niet genummerde sectie’ opgemaakt en geïndenteerd als een *section*. Analoog kan dit met *chapter* of *subsection*

2.4.4 Abstract

In de documentklassen *report* en *article* is er een **abstract** omgeving voorzien. Hierin komt normaalgezien een samenvatting van de tekst. Voor een boek is die omgeving niet aanwezig. De samenvatting wordt als gewone tekst beschouwd, en moet dus in een gewoon hoofdstuk geplaatst worden. Bij een thesis is het gebruikelijk om de samenvatting en/of abstract in de `\frontmatter` te plaatsen.

2.4.5 Secties

De volgende commando’s zijn beschikbaar om de grove structuur van een document aan te duiden:

```
\part
\chapter
\section
\subsection
\subsubsection
\paragraph
\subparagraph
```

Het zijn deze commando’s (met uitzondering van `\part`) die de nummering voor de titels vormen. Met interne tellers houdt L^AT_EX de nummering van elke sectie bij. In de documentklassen *book* en *report*, is het hoogste onderverdeelniveau `\chapter`. De hoofdstukken worden

onderverdeeld in *sections* die op hun beurt weer onderverdeeld worden in *subsection* en zo voort. In de documentklasse *article* bestaan er geen hoofdstukken en begint de hiërarchie vanaf `\section`. In de praktijk wordt `\paragraph` en `\subparagraph` nooit gebruikt.

Wanneer een nieuwe `\chapter` wordt begonnen, worden de tellers voor alle secties eronder (`\section`, `\subsection` ...) terug op één gezet.

Het commando `\part` wordt gebruikt om het werk onder te verdelen. De hoofdstuknummering loopt echter door. Een gebruik hiervan in de thesis kan bijvoorbeeld zijn dat in het eerste deel de literatuurstudie komt, die bestaat uit drie hoofdstukken. Het tweede deel bevat materiaal en methoden (hoofdstuk vier tot vijf), het derde deel bevat de resultaten (hoofdstuk 6 tot 8) en het vierde deel de conclusies (hoofdstuk 9). Op die manier is het mogelijk om de grote delen van de thesis naar voor te laten komen, zonder een nummeringsniveau te verliezen. Het is namelijk zo dat voor de duidelijkheid van de structuur, het niet aangewezen is om meer dan drie nummeringsniveau's diep te gaan. Iets als '3.9.5.8.A.3.b Fase 3 van de extractie' is uit den boze. Geen lezer die nog kan volgen waar in de hiërarchie deze titel zich bevindt. Het is om die reden dat `\subsubsection`-titels in boeken nooit genummerd worden.

De syntax van al deze commando's is (we nemen hier als voorbeeld `\section`):

```
\section{Titel}
\section[Korte titel]{Lange titel}
\section*{Titel}
```

Twee varianten dus. De eerste wordt gebruikt wanneer de titel een nummering krijgt en in de inhoudsopgave wordt opgenomen. Soms zijn titels te lang om op te nemen in de inhoudsopgave, of bovenaan de bladzijde (in de paginakoppen of *headers*). De mogelijkheid bestaat dan om optioneel, tussen vierkante haakjes (niet verplichte parameters voor een commando worden eerst gegeven en tussen vierkante haakjes, in plaats van tussen accolades) een kortere titel te geven, die gebruikt wordt in de inhoudsopgave en in de paginakoppen.

Bij gebruik van de sterretjesvorm wordt er geen nummering getoond (noch aangepast, dus de interne teller van die sectie wordt niet verhoogd) en wordt er niets in de inhoudsopgave gezet.

2.4.6 Appendix

Het deel van het document dat appendices bevat, wordt aangezet door de declaratie `\appendix` te geven. Dit commando zorgt ervoor dat de teller van de hoofdstukken (of secties voor een *article*) opnieuw op 1 wordt gezet. Verder wordt vanaf dan de hoofdstuknummering in letters weergegeven en wordt het woord 'Hoofdstuk' vervangen door 'Appendix'.

2.5 Refereren naar andere delen

Soms wensen we in onze tekst te verwijzen naar een ander hoofdstuk, tabel, bladzijde, vergelijking of figuur. Natuurlijk typen we deze getallen niet rechtstreeks in. We hebben tenslotte toch de computer die voor ons de administratie kan bijhouden.

Op de plaats waar we naar willen refereren, zetten we het commando:

```
\label{labeltje}
```

Hierbij is `labeltje` een sleutelwoord dat uit letters, cijfers, en leestekens (het verbindingsstreepje en de *underscore* zijn ook toegelaten) bestaat. Om het even waar in het document kunnen we dan de commando's

```
\ref{labeltje}  
\pageref{labeltje}
```

geven. Het eerste print de nummer van de tabel, sectie, hoofdstuk (afhankelijk van waar het `\label` commando gegeven werd) af, terwijl het tweede altijd de juiste bladzijde geeft.

Voor diegenen die de perfectie nastreven,³ is er het pakket `varioref`. Hiermee is het mogelijk om iets intelligenter te refereren: als er gerefereerd wordt naar een label op dezelfde bladzijde of de vorige/volgende bladzijde, verschijnt er iets als “op deze/de vorige/volgende bladzijde”. We gaan hier echter niet in detail op in. Meer uitleg is te vinden in `varioref.dvi`⁴.

³Maar weet dat het betere de vijand is van het goede.

⁴Op Debian in `/usr/share/doc/texmf/latex/tools/varioref.dvi.gz`

Hoofdstuk 3

Platte tekst

Onder ‘platte’ tekst (*plain text*) verstaan we simpele tekst zonder toeters en bellen, dus geen formules, tabellen of figuren. Alle tekst die komt na een percentteken (%) wordt als commentaar geïnterpreteerd en wordt niet weergegeven in het finale document.

3.1 Speciale Tekens

3.1.1 \LaTeX symbolen

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

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

De volgende zeven tekens kunnen door er een `\` (backslash) voor te zetten gewoon afgedrukt worden `$ & % # _ { }`:

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

Een tilde (~) kan met `\~{ }` verkregen worden. Een hoedje, ook wel accent circonflexe genoemd, (^) is met `\^{ }` te verkrijgen. Groter dan (>), kleiner dan (<) en de rechte streep (|) verkrijgt men door er dollartekens rond te plaatsen: `$>$`, `$<$` en `$|`. De *backslash* (\) krijgt men met `\verb? \?` of met `\backslash` (dollartekens dienen om in *math mode* te komen, zie hoofdstuk 6).

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

Voor de ouderwetse Nederlandse „aanhalingstekens” wordt het volgende gebruikt: “‘xxx’”. Om te openen dus een dubbele *quote* gevolgd door een *back quote*. Het sluiten is weer een dubbele *quote* maar nu gevolgd door een normale enkele *quote*.

3.1.2 Euro

Het **euro**-symbool is iets dat pas later werd toegevoegd aan L^AT_EX. Om het te kunnen gebruiken moet het pakket **eurosim** geladen worden. In de *preamble* wordt hiertoe de volgende lijn opgenomen:

```
\usepackage{eurosym}           % Om het euro symbool te krijgen
```

Een euro symbool invoeren gebeurt dan met het commando `\euro`, wat € geeft (die laatste slash na `\euro` zorgt ervoor dat er een spatie is na het eurosymbool).

3.1.3 Ellipsis (dots)

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

3.1.4 Koppelteken en verbindingsstreepjes

Er bestaan drie soorten streepjes: -, –, —. Het kleinste (-) wordt gebruikt als koppelteken tussen woorden en voor woordafbreking op het einde van een regel. Het tweede (–) wordt gebruikt voor bereiken tussen getallen (1–3) en het grootste wordt als gedachtenstreepje gebruikt — zoals dit. Zij worden getypt door het koppelteken op het toetsenbord één, twee of drie keer na elkaar te gebruiken. Een vierde soort streepje is het minteken, maar dat zien we later in *math mode*.

3.1.5 Speciale letters en accenten

Om accenten op letters te plaatsen, moeten de volgende commando’s gebruikt worden (tenzij het pakket **inputenc** gebruikt wordt, zie sectie 2.1.2 op bladzijde 9):

```
ó=\’o   ò=\’o   ô=\^o   ö=\\"o   õ=\~o   õ=\=o   ô=\.o   q=\c{o}   ô=\r{o}
```

Dus om een cédille (ç) te produceren, gebruiken we het volgende commando: `\c{c}`.

Om graden celsius (°C) weer te geven moet een nieuw pakket geladen worden: **textcomp**:

```
\usepackage{textcomp}         % Voor onder andere graden celsius
```

Met het commando `\textcelsius` kan dan het graden celsius symbool gegenereerd worden. Indien het pakket `SIunits` geladen is (zie sectie 3.12), kan graden celsius ook verkregen worden met `\celsius`.

Enkele speciale letters die soms van pas kunnen komen:

`œ=\oe` `Œ=\OE` `ø=\o` `Ø=\O` `Å=\AA` `ß=\ss` `¡=!` `¿=?`

3.2 Spaties

\LaTeX verzorgt de opmaak van je document. Het is daarom niet van belang wáár je de spaties en de regelovergangen plaatst in je bronbestand, omdat \LaTeX zich daar toch niets van aantrekt. Meerdere spaties na elkaar worden door \LaTeX vervangen door één enkele spatie. Spaties in het begin van een paragraaf worden genegeerd.

Als je expliciet ergens een spatie wilt invoeren, dan doe je dat met `\` (i.e. een backslash gevolgd door een letterlijke spatie).

Na commando's worden spaties normaalgezien ingeslikt. Als we bijvoorbeeld het volgende in onze broncode typen: `\euro spatie.` dan geeft dit €spatie, terwijl we liever € spatie zien. Om dit te vermijden, moeten we een harde spatie invoegen door een backslash te plaatsen na het commando: `\euro\ spatie.`

Soms mogen twee woorden die gescheiden worden door een spatie, niet op een verschillende lijn terecht komen. Hiervoor wordt een beschermde spatie gebruikt, aangegeven door een tilde: `~`. Dus `A.~Armagneau` geeft 'A. Armagneau', waarbij nooit een nieuwe lijn zal begonnen worden na de 'A.'.

Een meer algemene manier om iets bij elkaar te houden, is het gebruik van een `mbox`: `\mbox{sa-men}` zal altijd bij elkaar blijven. Ook wanneer het \LaTeX goed uitkomt om de regel af te splitsen tussen 'sa' en 'men' (als je 'sa-men' zomaar in je tekst schrijft, zal \LaTeX er namelijk van uitgaan dat er mag gesplitst worden op de plaats van het streepje).

3.3 Paragrafen

Eén harde enter op het einde van een regel, wordt door \LaTeX aanzien als een spatie. Een lege regel (een aantal opeenvolgende lege regels worden, analoog aan opeenvolgende spaties, slechts als één lege regel aanzien) geldt echter als een paragraafovergang. Dat wil nog niet zeggen dat er in het gecompileerde document ook een lege regel komt, wel dat er een nieuwe lijn wordt begonnen. Een nieuwe paragraaf beginnen, kan ook door het commando `\par` te geven.

Het hangt van de documentklasse af, hoe de paragrafen gezet worden. In artikels, rapporten en boeken worden paragrafen weergegeven door het inspringen van de eerste regel en wordt er geen witruimte gelaten tussen de paragrafen.¹

Dit kan veranderd worden door de volgende aanwijzingen in de *preamble* te plaatsen:

```
\setlength{\parindent}{0pt} % Geen inspringen van paragrafen
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex} % Ruimte tssn paragrafen, slecht
```

Met de eerste aanwijzing wordt de paragraafinspringing op nul punten gezet. De tweede aanwijzing geeft aan dat tussen elke paragraaf een witte ruimte moet gelaten worden: namelijk één keer de hoogte van de letter x. L^AT_EX krijgt echter speling van ons: als blijkt dat er een mooiere bladschikking is als de witte ruimte wordt vergroot (maximum met de helft van de hoogte van de letter x) of verkleind (met 0.2 keer de hoogte van de letter x) dan mag dat gebeuren. Dit is een zogenaamde **rubberen lengte** (zie ook sectie 3.5.1).

Er is echter ook een nadeel verbonden aan het instellen van een niet-nul **parskip**: die afstand wordt namelijk ook gebruikt om te plaatsen rond titels. Zodus krijg je veel te veel witte ruimte rond je titels. Als oplossing hiervoor werd in dit document elke paragraaf niet gescheiden door een lege regel, maar door het zelfgebakken commando **\npar**, dat in de *preamble* als volgt werd gedefinieerd:

```
\newcommand{\npar}{\par \vspace{2.3ex plus 0.3ex minus 0.3ex}}
```

Dit commando zorgt ervoor dat wanneer het geplaatst wordt, een nieuwe paragraaf wordt begonnen en een verticale lege ruimte wordt ingebracht. Natuurlijk werd de **\parskip** niet aangepast (vandaar dat hierboven ‘slecht’ in de commentaar staat)!

Normale tekst wordt uitgevuld. L^AT_EX 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.

Voor uitzonderingsgevallen² kan men het afbreken beïnvloeden met de volgende commando’s: Het commando **** of **\newline** breekt de regel af zonder een nieuwe paragraaf te beginnen. Het commando ***** breekt een regel af, maar er mag op die plaats geen nieuwe pagina begonnen worden.

Wanneer extra witruimte tussen de regels gewenst is, kan dat als optionele parameter aan het commando **** doorgegeven worden, bijvoorbeeld **\[5mm]**.

¹In de nederlandse documentklasse **artikel3** is dit echter niet zo. Daarin worden paragrafen gescheiden door witte ruimte en niet door het inspringen van de eerste regel.

²Het is dus niet de bedoeling dat je dit constant doorheen je document gebruikt. L^AT_EX weet meestal zelf beter waar regels afgebroken moeten worden.

Een alternatief voor het gebruik van `\npar` is `\\` (twee keer een nieuwe regel). Technisch gezien wordt er dan geen nieuwe paragraaf begonnen en dus zal er ook nooit ingesprongen worden (wat de waarde van `\parindent` ook weze). Verder is de verticale witte ruimte gelijk aan de hoogte van een tekstregel, wat eigenlijk te veel is.

Met behulp van de commando's

```
\linebreak[n]
\nolinebreak[n]
\pagebreak[n]
\nopagebreak[n]
```

kan de wenselijkheid voor het afbreken van regels (pagina's) ingesteld worden, `[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. Het commando `\clearpage` doet dit ook, maar drukt eerst alle nog niet afgedrukte figuren en tabellen af. Het commando `\cleardoublepage` doet hetzelfde, maar wanneer het document wordt gecompileerd om dubbelzijdig af te drukken, wordt er op een rechterbladzijde begonnen.

L^AT_EX doet erg veel moeite om mooie regels te maken. Als echter geen enkele van de ingebouwde methoden een mogelijkheid biedt om een gladde rechterkantlijn te produceren, dan resulteert dit in een waarschuwing (*‘overfull hbox’*). Dit treedt vooral op als er geen geschikte plek wordt gevonden om een woord af te breken. Als het commando `\sloppy` wordt gegeven, is L^AT_EX gewoonlijk veel minder streng en vermijdt zulke lange regels door de woordafstand te vergroten (wat er echter minder mooi uitziet). Het effect van `\sloppy` kan weer ongedaan gemaakt worden met `\fussy`. Er worden in dit geval ook waarschuwingen geproduceerd (*‘underfull hbox’*), maar het resultaat is meestal goed bruikbaar.

3.4 Woordafbreking

Het afbreken van woorden (hyphenatie) op het einde van de regels, zodat de tekst mooi uitgevuld is, gebeurt automatisch. Die woordafbreking kan vermeden worden door een paragraaf tussen `\begin{sloppypar}` en `\end{sloppypar}` te zetten (of door het commando `\sloppy` te geven: vanaf die plaats wordt dan in *sloppypar* mode gewerkt; ongedaan maken van `\sloppy` gebeurt met `\fussy`). Dit zet echter de woordafbreking uit voor de hele paragraaf, wat voor niet zo mooi ogende tekstopmaak zorgt. Een betere manier is om L^AT_EX te laten weten hoe een woord moet gesplitst worden. Dit kan door in de *preamble* het volgende commando op te nemen:

```
\hyphenation{af-split-sen woord-af-bre-king}
```

De woorden worden dus van elkaar gescheiden door een spatie en niet door een komma. Midden in de tekst kan ook het volgende gegeven worden: `af\-\bre\-\ken`. Dit geeft gewoon ‘afbreken’ maar maakt duidelijk aan L^AT_EX waar het woord afgebroken mag worden.

L^AT_EX werd oorspronkelijk in Amerika ontwikkeld, waar Engels de voertaal is. Daardoor worden Engelse afsplitsingsregels gebruikt, die niet altijd correct zijn in het Nederlands. Vandaar dat we specifiek in ons document moeten meegeven welke taal we willen gebruiken. De naam van de L^AT_EX uitbreiding die daarvoor instaat is `babel`. Door de volgende lijn in onze *preamble* op te nemen, zorgen we voor Nederlandstalige ondersteuning in ons document:

```
\usepackage[dutch]{babel}          % Voor nederlandstalige hyphenatie
```

Dit zorgt niet alleen voor de juiste hyphenatie, maar ook voor juiste titels (Inhoudsopgave in plaats van *Table of Contents*, Hoofdstuk in plaats van *Chapter*).

Wanneer we nu zomaar ons document compileren, is de kans groot dat we de volgende waarschuwing krijgen (te vinden in het *log*-bestand dat door de latexcompiler gegenereerd werd):

```
Package babel Warning: No hyphenation patterns were loaded for
(babel)                the language ‘Dutch’
(babel)                I will use the patterns loaded for \language=0 instead.
```

De taal 0 staat voor Amerikaans Engels. Niet goed voor onze hyphenatieregels. Dit komt omdat het Nederlands niet echt belangrijk is op wereldschaal en daardoor niet standaard ingeschakeld staat in L^AT_EX. Er moet ergens in de configuratiebestanden iets veranderd worden, en daarna moet aan L^AT_EX meegedeeld worden dat er een configuratiebestand veranderd is (de latexcompiler leest namelijk niet elke keer alle configuratiebestanden; dat zou veel te lang duren). Om Nederlands te kunnen gebruiken in een document, moet het volgende gebeuren (hiervoor heb je administratieve rechten nodig op je computersysteem):

1. Open het bestand *language.dat* met een gewone teksteditor (wordpad, vim, emacs). Het zou kunnen dat er zo verschillende bestanden op je L^AT_EX systeem staan. Het juiste bestand is op Debian GNU/Linux te vinden onder de *generic* directory³.
2. In dat bestand vind je een lijn die er ongeveer als volgt uit ziet:

```
%! dutch          nehyph.tex
```

Verwijder het commentaarteken, het uitroepteken (het zou kunnen dat er op sommige systemen geen uitroepteken staat) en de spatie.

3. Bewaar het bestand.

³namelijk: `/usr/share/texmf/tex/generic/config/language.dat`

4. Geef dan ergens in een commandovenster (dos-prompt) het volgende commando

```
fmtutil --all
```

Het zou kunnen dat er dan een foutmelding verschijnt dat het commando niet te vinden is. Zoek er dan naar, ga naar de directory waar het zich bevindt⁴ en voer het vandaar uit. Verder zou het kunnen dat op een Windowssysteem het volgende moet gebruikt worden: `fmtutil /all`.

Voor de Debian gebruikers is *The Debian Way to do it* als root `dpkg-reconfigure tetex-bin` via de commando prompt in te geven. Op andere GNU/Linux of Unix systemen is er een handig menugebaseerd programmaatje dat je, als root, oproept met `texconfig`.

Onder Windows kan je in het MikTeX configuratie programma (start >> programma's >> MikTeX >> MikTeX Options) verschillende talen selecteren. De \LaTeX compiler wordt dan automatisch geupdated.

Mensen die graag meerdere talen in eenzelfde document gebruiken, kunnen dat. Selecteer hiertoe meerdere talen in de *preamble*: `\usepackage[dutch,english]{babel}`. De taal die standaard gebruikt wordt in het document is de laatste die ingegeven wordt. Wanneer we willen overschakelen op de andere taal, geven we het commando `\selectlanguage{dutch}`. Om terug te keren naar het Engels: `\selectlanguage{english}`. Meer opties kunnen gevonden worden in het helpbestand van Babel.⁵

3.5 Arbitraire witte ruimte

In dit deel wordt uit de doeken gedaan hoe je \LaTeX kan verplichten om ergens witte ruimte vrij te laten. Gebruik deze commando's echter niet lichtzinnig. Het is niet de bedoeling dat de auteur constant aangeeft waar er witte ruimte moet komen. \LaTeX weet dit meestal zelf beter. Slechts in uitzonderlijke gevallen mag de auteur dit corrigeren.

3.5.1 Lengte eenheden in \LaTeX

Lengtes hebben in \LaTeX altijd een eenheid, ook als we de lengte nul wensen. De meest gebruikelijke eenheden zijn:

<code>cm</code>	centimeter
<code>mm</code>	millimeter
<code>in</code>	inch
<code>pt</code>	punten (1 inch = 72.27 pt)
<code>em</code>	lettertype specifiek: de breedte van een hoofdletter M
<code>ex</code>	lettertype specifiek: de hoogte van de letter x

⁴Op een Debian GNU/Linux systeem: `/usr/bin/fmtutil`

⁵Op een Debian GNU/Linux systeem te vinden in `/usr/share/doc/texmf/generic/babel/user.dvi.gz`

Soms zijn zogenaamde rubberen lengtes nodig. Dit betekent dat we een gemiddelde lengte opgeven met een maximum en een minimum. Rubberen lengtes worden als volgt voorgesteld: `{1ex plus0.5ex minus0.3ex}`. In sectie 3.3 op bladzijde 23 wordt een rubberen lengte gebruikt om de ruimte tussen paragrafen te bepalen.

Met `\fill` wordt een speciale rubberen lengte gezet. Dit is een rubberen lengte met gemiddelde nul en maximale waarde gelijk aan oneindig.

3.5.2 Horizontale witte ruimte

De volgende commando's kunnen gebruikt worden om ergens in een zin horizontale ruimte in te voeren:

```
\hspace{lengte}
\hspace*{lengte}
```

Hierin is `lengte` een lengte-eenheid van de vorm uitgelegd in sectie 3.5.1. De gewone vorm (zonder asterisk) van `\hspace` negeert de vraag naar witte ruimte wanneer die op het einde van een regel komt, zodat een nieuwe regel nooit met een spatie begint. De `*`-vorm zet altijd witte ruimte, ook aan het begin van een regel.

De lengtespecificatie mag negatief zijn. Op die manier is het mogelijk om twee karakters bovenop elkaar te printen, bijvoorbeeld de euro van de arme man (arm, omdat hij het `eurosym` pakket niet heeft): €⁶.

Het commando `\hfill` is een afkorting van `\hspace{\fill}` en zorgt ervoor dat de tekst links van het commando naar links wordt geduwd en rechts van het commando naar rechts. Bijvoorbeeld

```
De auteur \hfill De begeleider \hfill De promotor
```

geeft het volgende:

```
De auteur                De begeleider                De promotor
```

Variaties op `\hfill` zijn `\dotfill` en `\hrulefill`. Het eerste vult aan met punten, het tweede met een horizontale lijn.

Verder kunnen `\quad` en `\qquad` gebruikt worden. Het eerste zet een spatie van 10, 11 of 12 punten, afhankelijk van welke optie in de documentklasse wordt meegegeven (zie sectie 2.1.1 op bladzijde 7). Het tweede zet een spatie die dubbel zo groot is. Korte spaties kunnen ingegeven worden met `\;` en `\,`, waarbij het laatste echt wel klein is.

⁶Geproduceerd met `C\hspace{-0.8em}=`

Er bestaat nogal wat verwarring over het feit of er nu een spatie moet komen tussen een getal en zijn eenheid, of dat die spatie overbodig is. De conventie is een halve spatie: `\,.` Schrijf dus `10\,km/u`, wat het juiste `10 km/u` geeft en niet het foute `10 km/u`. Het lettertype moet recht zijn, niet cursief. Nog beter is echter gebruik maken van het pakket `SIunits` (zie bladzijde 3.12).

3.5.3 Verticale witte ruimte

Naast het gebruik van `\\[lengte]` kan verticale witte ruimte ook gegenereerd worden met het commando `\vspace`, waarbij de syntax juist dezelfde is als voor `\hspace`. Ook hier kunnen negatieve waarden ingegeven worden. Praktisch nut hiervan kan zijn om de tekst eens hoger op een nieuwe bladzijde te beginnen.

Het commando `\vfill` doet hetzelfde als `\hfill` maar dan in verticale richting.

Speciale commando's om witte ruimte in te voeren, afhankelijk van het gekozen lettertype, zijn: `\bigskip`, `\medskip` en `\smallskip`.

3.6 Lettertypes

In dit deel wordt overlopen hoe het mogelijk is om lettertypes te wijzigen. Een veel voorkomende fout bij wjziwk tekstverwerkers, is het gebruik van veel te veel verschillende lettertypes, wat zorgt voor een rommelig document. Het is veel stijlvoller wanneer alle titels hetzelfde lettertype hebben en slechts variëren in grootte.

3.6.1 Lettergrootte

De algemene lettergrootte wordt gekozen in het begin van het document, wanneer de documentklasse wordt gekozen (zie sectie 2.1.1 op bladzijde 7). Er kan gekozen worden tussen 10pt, 11pt en 12pt lettergrootte. Het verschil tussen deze basisgroottes is groter dan men op het eerste zicht zou denken:

Ideaal voor veel tekst op weinig bladzijden: 10pt.

Ideaal voor een leesbaar document: 11pt.

Ideaal om te doen alsof er veel tekst is: 12pt.

Soms is het nodig om in de tekst een groter of een kleiner lettertype te kiezen. Dit wordt niet gedaan door een andere puntsgrootte te nemen, maar door specifiek te zeggen wat je wenst. Op die manier hoef je niet alles opnieuw in te stellen, als je zou beslissen om de algemene documentgrootte te veranderen. Er bestaan tien⁷ specificaties om de lettergrootte te bepalen. In tabel 3.1 worden ze weergegeven, samen met het commando om ze te verkrijgen.

⁷Exodus 20:1-17

De `\normalsize` is hetgeen normaalgezien doorheen het document gebruikt wordt. Let op het gebruik van hoofdletters in de commandonamen.

Tabel 3.1: De tien lettergroottes.

<code>\Huge</code>	Bovenal bemin één God,
<code>\huge</code>	Zweert niet ijdel, vloekt noch spot.
<code>\LARGE</code>	Heilig steeds de dag des Heren,
<code>\Large</code>	Vader, moeder zult gij eren.
<code>\large</code>	Dood niet, geef geen ergernis,
<code>\normalsize</code>	Doe nooit wat onkuisheid is.
<code>\small</code>	Vlucht het stelen en bedriegen,
<code>\footnotesize</code>	Ook de achterklap en 't liegen.
<code>\scriptsize</code>	Wees steeds kuis in uw gemoed,
<code>\tiny</code>	En begeer nooit iemands goed.

Wanneer een lettergrootte-declaratie wordt gegeven, bijvoorbeeld `\small`, wordt alle tekst vanaf dan in die lettergrootte weergegeven. Om enkele woorden klein te krijgen, moet je het volgende typen: `\tiny schattig klein \normalsize`, wat `schattig klein` geeft. Deze lettergrootte-declaraties zijn zogenaamde omgevings-declaraties: zij wijzigen de omgeving. Vandaar dat het bovenstaande voorbeeld kan herschreven worden als `{\tiny schattig klein}` (een manier om een omgeving te openen is ‘{’ en om te sluiten ‘}’). Op die manier hoeft het originele lettertype (de `\normalsize`) niet gekend te zijn. Men zou verleid kunnen worden om het volgende te gebruiken: `\tiny{schattig klein}`, maar dit is ten zeerste af te raden, daar het niet werkt en de documentopmaak volledig in de war stuurt.

Het veranderen van de lettergrootte wijzigt ook de interlinieruimte (gelukkig maar!). De interlinieruimte wordt bepaald door twee lengtes: `\baselineskip` en `\baselinestretch`. Om de totale interlinieruimte te kennen, moeten deze twee lengtes met elkaar vermenigvuldigd worden. De `\baselineskip` is de natuurlijke (of minimale) interliniespatiëring voor elk lettertype. De gebruiker zou dit dus eigenlijk nooit moeten wijzigen. De `\baselinestretch` is een factor die met de `\baselineskip` wordt vermenigvuldigd, wat de gebruikte interlinieafstand geeft. Dit is hetgeen mag gewijzigd worden door de gebruiker. Standaard is de `\baselinestretch` gelijk aan 1, maar voor dit document werd 1.2 gebruikt. Het wijzigen kan gebeuren in de *preamble* met het commando:

```
\renewcommand{\baselinestretch}{1.2} % De interlinie afstand wat vergroten.
```

Maar ook in de *body* van het document kan met hetzelfde commando de `\baselinestretch` gewijzigd worden. Om de wijziging actief te maken, moet echter een andere lettergrootte gekozen worden (doordat de totale interlinieafstand slechts één keer berekend wordt, namelijk bij het begin van het gebruik

van een lettergrootte), bijvoorbeeld `\small\normalsize` wanneer je toch wenst in `\normalsize` te blijven.

3.6.2 Letterstijlen

Soms is het nodig om andere letterstijlen te selecteren. Dit kan op verschillende manieren. In tabel 3.2 staan de verschillende mogelijkheden. In de tweede kolom staat hoe de omgeving kan gewijzigd worden. In de derde kolom staan de verschillende commando's om korte stukken tekst te veranderen. Omgevingen kunnen op twee manieren gebruikt worden: enerzijds door de declaratie `\scshape` te geven (om bijvoorbeeld *Small Caps* te verkrijgen), anderzijds door met `\begin{scshape}` en `\end{scshape}` te werken. De commandovorm wordt gebruikt voor korte stukken tekst en is eigenlijk equivalent met '`{\scshape xxx}`'.

Tabel 3.2: Hoe verschillende letterstijlen selecteren: met een nieuwe omgeving of met een commando.

	Omgeving	Commando	Voorbeeld
Familie	<code>\rmfamily</code>	<code>\textrm{xxx}</code>	Roman letters.
	<code>\ttfamily</code>	<code>\texttt{xxx}</code>	Typewriter letters. Typemachine.
	<code>\sffamily</code>	<code>\textsf{xxx}</code>	Sans serif letters.
Vorm	<code>\upshape</code>	<code>\textup{xxx}</code>	Upright tekst. Rechtopstaande tekst.
	<code>\itshape</code>	<code>\textit{xxx}</code>	<i>Italic tekst.</i>
	<code>\slshape</code>	<code>\textsl{xxx}</code>	<i>Slanted letters.</i>
	<code>\scshape</code>	<code>\textsc{xxx}</code>	SMALL CAPS. HOOFDLETTERS.
Serie	<code>\mdseries</code>	<code>\textmd{xxx}</code>	Medium weight letters, de standaard.
	<code>\bfseries</code>	<code>\textbf{xxx}</code>	Bold face. Vettige letters.
Default		<code>\textnormal{xxx}</code>	De standaard.
Benadrukken		<code>\emph{xxx}</code>	<i>Te benadrukken tekst, meestal italic.</i>

Het `\emph{}` is een speciaal commando. Het zorgt ervoor dat de te benadrukken tekst in een ander lettertype wordt gezet. Dit is meestal *italic* maar als je al in *italic* zit, wordt dit weer normale tekst. Dus `\emph{E.~coli} \textit{latijn \emph{E.~coli} latijn}` geeft: *E. coli latijn E. coli latijn*.

3.7 Centreren en indentatie

L^AT_EX zorgt ervoor dat je automatisch rechte kantlijnen hebt. Als dit niet lukt, krijg je *underfull* (wanneer de lijn niet volledig vol geraakt) of *overfull* (wanneer de tekst uitsteekt buiten de rechterkantlijn) waarschuwingen. Soms is iets anders gewenst dan uitgevulde tekst.

Tekst centreren gebeurt met de `center` omgeving:

```
\begin{center}
Te centreren tekst.
\end{center}
```

Om één enkele lijn te centreren kan ook het commando `\centerline{tekst}` gebruikt worden.

Om alle tekst naar links uit te lijnen, kan het volgende gebruikt worden:

```
\begin{flushleft}
Naar links uit te lijnen tekst.
\end{flushleft}
```

Analoog kan `flushright` gebruikt worden om tekst evenwijdig met de rechterkantlijn uit te lijnen.

Om tekst uit te vullen waarbij de kantlijnen inspringen, zoals deze paragraaf, kan je `\begin{quote} tekst \end{quote}` gebruiken.

In plaats van `quote` kan ook `quotation` gebruikt worden. Bij dit laatste commando wordt de eerste zin van een paragraaf ingesprongen, terwijl bij `quote` de paragrafen gescheiden worden door extra verticale ruimte.

Om poëzie te indenteren, kan gebruik worden gemaakt van de `verse` omgeving. Individuele lijnen worden dan beëindigd met `\\`, strofen worden aangegeven door blanco lijnen.

3.8 Lijsten

Er bestaan drie soorten geordende lijsten in L^AT_EX: **itemize**, **enumerate** en **description**. De lijsten beginnen met `\begin{soort_lijt}` en eindigen met `\end{soort_lijt}`. Binnen deze omgeving worden de verschillende items begonnen met `\item`, waarbij al dan niet een optioneel argument kan meegegeven worden: `\item[argument]`.

3.8.1 Itemize

- De *itemize* omgeving zorgt ervoor dat voor elk item een *bullet* verschijnt.
- Elk item wordt begonnen met `\item`.
- * Indien iets anders gewenst is dan een bolletje in het begin, kan dat meegegeven worden als extra argument: `\item[*]`.
- Waarom gemakkelijk doen als moeilijk ook kan? Met `\item[\bullet]` verkrijg je hetzelfde als met `\item`.

De bovenstaande *itemize* lijst werd als volgt gegenereerd:

```

\begin{itemize}
\item De \engels{itemize} omgeving zorgt ervoor dat voor elk item een \engels{
    bullet} verschijnt.
\item Elk item wordt begonnen met \lcommand{\item}.
\item[*] Indien iets anders gewenst is dan een bolletje in het begin, kan dat
    meegegeven worden als extra argument: \lcommand{\item[*]}.
\item[$\bullet$] Waarom gemakkelijk doen als moeilijk ook kan? Met \lcommand{\item\[$\bullet$]} verkrijg je hetzelfde als met \lcommand{\item}.
\end{itemize}

```

3.8.2 Enumerate

1. De *enumerate* omgeving zet nummers voor de verschillende items.
2. De nummering begint vanaf 1.
- 3. Ook hier kan een optioneel argument meegegeven worden: `\item[--3.]`.
3. Let er echter op dat de teller van de nummering dan niet verhoogd wordt.

De bovenstaande *enumerate* lijst werd als volgt gegenereerd:

```

\begin{enumerate}
\item De \engels{enumerate} omgeving zet nummers voor de verschillende items.
\item De nummering begint vanaf 1.
\item[--3.] Ook hier kan een optioneel argument meegegeven worden: \lcommand{\item[--3.]}.
\item Let er echter op dat de teller van de nummering dan niet verhoogd wordt.
\end{enumerate}

```

3.8.3 Description

doel Deze lijst kan gebruikt worden wanneer een aantal termen moeten verklaard worden.

werking Het optioneel argument van `item` wordt in het vet gezet.

Wanneer geen optioneel argument wordt meegegeven, krijg je iets zoals dit item.

De bovenstaande *description* lijst werd als volgt gegenereerd:

```

\begin{description}
\item[doel] Deze lijst kan gebruikt worden wanneer een aantal termen moeten
    verklaard worden.
\item[werking] Het optioneel argument van \lcommand{item} wordt in het vet gezet.
\item Wanneer geen optioneel argument wordt meegegeven, krijg je iets zoals dit
    item.
\end{description}

```

3.8.4 Geneste lijsten

Lijsten kunnen genest worden. Als lijsten van dezelfde soort genest worden, verandert de labeling (bij *itemize*) of de manier van nummering (bij *enumerate*).

- Overdrijven •
 1. Overdrijven 1.
 - Overdrijven –
 - (a) Overdrijven (a)
 2. Overdrijven 2.

Dit overdreven gebruik van ‘overdrijven’ werd als volgt gegenereerd:

```
\begin{itemize}
\item Overdrijven $\bullet$
  \begin{enumerate}
\item Overdrijven 1.
  \begin{itemize}
\item Overdrijven --
    \begin{enumerate}
\item Overdrijven (a)
    \end{enumerate}
  \end{itemize}
\item Overdrijven 2.
  \end{enumerate}
\end{itemize}
```

Het nesten in eenzelfde omgeving mag tot vier niveau’s diep gaan. Wanneer verschillende soorten lijsten worden gebruikt, kan men tot zes niveau’s diep gaan. Het is niet omdat het mogelijk is om zeer diep te nesten, dat het ook goed is om dat te doen. Denk aan de leesbaarheid: vanaf drie niveau’s zal de lezer nog moeilijk weten in welk subniveau hij zich bevindt en gaat hij de lijst lezen als gewone doorlopende tekst.

3.8.5 Veranderen van het label

Wat \LaTeX eigenlijk doet, telkens als het het `\item` commando wordt gegeven, is dit commando vervangen door (afhankelijk van de omgeving waarin je zit):

```
\labelitemi \labelitemii \labelitemiii \labelitemiv
\labelenumi \labelenumii \labelenumiii \labelenumiv
```

Hierbij wordt de *i* voor het eerste, *ii* voor het tweede, *iii* voor het derde en *iv* voor het vierde niveau gebruikt.

Deze commando’s kunnen echter gewijzigd worden. Om bijvoorbeeld plustekens te krijgen in *itemize* in plaats van *bullets*, geven we het volgende commando, juist na `\begin{itemize}`:


```
\renewcommand{\labelitemi}{+}
```

Indien we deze aanpassing algemeen willen laten gelden, geven we dit commando in de *preamble*.

Voor de *enumerate* omgeving ligt het iets moeilijker: er moet namelijk een teller afgedrukt worden. De verschillende commando's om een teller af te drukken zijn `\arabic{tel}`, `\roman{tel}`, `\Roman{tel}`, `\alph{tel}` en `\Alph{tel}`. De uitleg over hoe een teller dan afgeprint wordt, is te vinden op bladzijde 15. De verschillende tellers zijn voor de vier niveau's: `enumi`, `enumii`, `enumiii` en `enumiv`.

Om bijvoorbeeld de nummering van het tweede niveau van *enumerate* te veranderen, naar Romeinse cijfers met een haakje achter ('I'), ('II'), ..., geef je het volgende commando juist na `\begin{enumerate}` of in de *preamble*:

```
\renewcommand{\labelenumii}{\Roman{enumii}}
```

Het zou ook kunnen dat je graag het vorige niveau erbij betreft, met letters ('A.I'), ('A.II'), ..., ('B.I'), ('B.II'), ...):

```
\renewcommand{\labelenumii}{\Alph{enumi}.\Roman{enumii}}
```

Vergeet dan ook niet het eerste niveau te hernoemen, want dit staat nog altijd op *arabic*.

Andere instellingen die enkel aan het begin van een lijst (na `\begin{soort_lijst}`) kunnen gegeven worden:

- `\topsep` De verticale ruimte die ingevoegd wordt tussen de lijst en de bovenliggende en onderliggende tekst.
- `\parsep` De verticale ruimte tussen verschillende paragrafen van eenzelfde item.
- `\itemsep` De verticale ruimte die bovenop `\parsep` wordt gebruikt om verschillende items te scheiden.

Het instellen van deze lengtes gebeurt bijvoorbeeld voor `\topsep` met `\setlength{\topsep}{lengte}`, waarbij `lengte` een lengte is zoals uitgelegd in sectie 3.5.1 op bladzijde 26.

3.8.6 Theorem — Stelling

Een speciale vorm van een *enumerate* omgeving, is een lijst waarvan de items doorheen heel het document lopen. Soms kan het van pas komen om doorheen de tekst, een soort nummering te hebben van bijvoorbeeld belangrijke eigenschappen (stellingen, definities) zoals het onderstaande voorbeeld:

Levensles 3.1 (Hoogmoed) *Hoogmoed is in zichzelf denken: “Bewijzen dat ik juist ben, is toegeven dat ik fout zou kunnen zijn.”*

Je zou dit zelf kunnen doen via het gebruik van verschillende letterstijlen, maar wanneer je dan ergens een extra item wilt toevoegen, moet je heel de nummering handmatig aanpassen. Niet handig. We zullen hier uitleggen hoe dit automatisch kan gebeuren. Hiervoor gebruiken we het voorbeeld van de 'levensles'. Met het commando (liefst in de *preamble* te geven)

```
\newtheorem{levensles}{Levensles}[chapter]
```

maken we de omgeving zoals gebruikt in levensles 3.1. Hierbij is `levensles` de naam van het *theorem* en `\Levensles` is hetgeen verschijnt in het begin (let hier op de hoofdletter). Het optionele argument `[chapter]` zorgt ervoor dat de telling van de levenslessen elk hoofdstuk opnieuw begint. Dit kan ook `[section]` of `[subsection]` zijn.

Het definiëren van een nieuwe levensles gebeurt als volgt:

```
\begin{levensles}[Schrijven]
  Wanneer je een boek schrijft, weet dan dat het papier geduldig is, maar de lezer
    niet.
\end{levensles}
```

Levensles 3.2 (Schrijven) *Wanneer je een boek schrijft, weet dan dat het papier geduldig is, maar de lezer niet.*

Het optionele `[Schrijven]` zorgt ervoor dat je de levensles kan benoemen.

3.9 Voetnoten

Voetnoten zijn te vermijden. Zij leiden de aandacht van de lezer af. Meestal kan de tekst van een voetnoot opgenomen worden in de doorlopende tekst, eventueel tussen haakjes.⁸

Voetnoten worden gegenereerd met het commando `\footnote{tekst}`, daar waar het nummer van de voetnoot moet verschijnen.

Voetnoten mogen alleen voorkomen in doorlopende tekst, niet in titels, tabellen en wiskundige modus. Wanneer we daar toch een voetnoot wensen, doen we dit door er een `\footnotemark` te zetten (dit lukt echter niet in titels). De tekst van de voetnoot geven we dan later in met `\footnotetext{tekst}`. Zorg wel dat dit laatste commando gegeven wordt voordat er een andere voetnoot wordt aangemaakt.

3.10 Kantlijnopmerkingen

Een kantlijnopmerking kan geproduceerd worden met:

```
\marginpar{Niet gebruiken, is lastig\\bij het\\afdrukken}
```

Dit plaatst de kantlijnopmerking in de rechterkantlijn. Wanneer dubbelzijdig wordt afgeprint, komt de kantlijnopmerking altijd in de buitenste kantlijn, dus voor oneven pagina's in de rechterkantlijn, voor even pagina's in de linkerkantlijn.

Een kantlijnopmerking is 1.9 cm breed. Deze breedte kan hergedefiniëerd worden met:

```
\setlength{\marginparwidth}{1cm} % kan ook andere lengte zijn ...
```

Niet ge-
bruiken, is
lastig
bij het
afdrukken

⁸Ja, dit document is een slecht voorbeeld.

3.11 Letterlijke tekst

Soms is het nodig om tekst exact weer te geven zoals hij getypt werd (bijvoorbeeld voor de voorbeelden die we ter illustratie geven in deze cursus). Voor korte stukken tekst waarin geen nieuwe lijnen zitten, kan het commando `\verb!tekst!` gebruikt worden. De uitroeptekens mogen vervangen worden door gelijk welk teken dat niet voorkomt in de letterlijke tekst zelf. De tekst wordt gezet in typmachineletters. Alle tekst wordt ook op één regel gepropt. Bijvoorbeeld `\verb?Verboden dingen zijn leuk: % $ & }{?geeft Verboden dingen zijn leuk: % $ & }{.` En zoals je kan zien, wordt de regel niet op tijd afgebroken.

Voor langere stukken tekst is de `verbatim` omgeving aangewezen (links de tekst die ingegeven werd om het rechtse resultaat te produceren):

<code>\begin{verbatim}</code>	
<code># include <stdio.h></code>	<code># include <stdio.h></code>
<code>void main(void)</code>	<code>void main(void)</code>
<code>{</code>	<code>{</code>
<code>printf("Dag Grootmoeder!\n")</code>	<code>printf("Dag Grootmoeder!\n")</code>
<code>}</code>	<code>}</code>
<code>\end{verbatim}</code>	

Het `\verb` commando en de `verbatim` omgeving mogen niet gebruikt worden in zelfgemaakte L^AT_EX commando's.

Er bestaat ook een pakket `verbatim` (`\usepackage{verbatim}` in de *preamble*) waarmee hele bestanden letterlijk kunnen ingevoegd worden in een document: `\verbatiminput{letterlijke-tekst.txt}`.

Voor grotere stukken code of wanneer het typmachinelettertype niet geschikt is, kan een speciaal pakket gebruikt worden: `listings`. Dit pakket is speciaal geschreven om vooral programmacode in een mooi lettertype te zetten (het werd in dit werk voor alle voorbeelden gebruikt). Met `listings` kunnen ook nieuwe `verbatim` omgevingen gemaakt worden. Verder bevat `listings` *syntax highlighting* (het in het vet/kleur zetten van sleutelwoorden) voor verschillende programmeertalen. Voor meer uitleg verwijzen we naar de zeer goede documentatie van het pakket: `listings.dvi`.⁹

3.12 Eenheden

Getallen hebben meestal een eenheid. Er moet een spatie staan tussen een waarde en zijn eenheid, maar geen nieuwe lijn. Verder is die spatie kleiner dan de spaties tussen woorden. Nogal complex om elke keer handmatig te doen. Het gebruik van het pakket `SIunits` is dan aangewezen. De documentatie ervan is verhelderend¹⁰.

⁹Op een Debian systeem te vinden in `/usr/share/doc/texmf/latex/listings/listings.dvi.gz`

¹⁰Op een Debian systeem te vinden in `/usr/share/doc/texmf/latex/SIunits/SIunits.pdf.gz`

Het pakket wordt geladen met de volgende regel in de *preamble*:

```
\usepackage[Gray,squaren,thinqspace,thinspace]{SIunits} % Elegant eenheden zetten
```

De opties `Gray` en `squaren` zijn nodig om conflicten te vermijden met de pakketten `pstricks` en `amssymb`. Optie `thinqspace` zorgt ervoor dat er slechts een kleine spatie is tussen een getal en zijn eenheid. Indien meer ruimte gewenst is, kan `mediumqspace` of `thickqspace` gebruikt worden. De optie `thinspace` (dus zonder die q van de vorige optie) geeft aan dat vermenigvuldiging van eenheden moet voorgesteld worden door een kleine spatie (bijvoorbeeld de spatie tussen newton meter: N m). Dit is eigenlijk de definitie van `\usk`, die we verder tegenkomen. Andere mogelijkheden zijn: `cdot` (een punt als vermenigvuldigingssymbool), `mediumspace` en `thickspace`.

Een getal koppelen aan zijn eenheid gebeurt met het commando `\unit{getal}{eenheid}` waarbij `getal` het getal is (mogen ook letters zijn, bijvoorbeeld `\unit{x}{\metre}`) en `eenheid` de eenheid. Als eenheid kunnen de voorgedefinieerde eenheden gebruikt worden, maar ook eigen eenheden kunnen gecreëerd worden. We kunnen ervan uitgaan dat we in *math mode* zitten (zie hoofdstuk 6), dus superscript kan: `\unit{9.81}{m/s^2}` geeft 9.81 m/s².

`SIunits` heeft ook een hoop voorgedefinieerde eenheden. Een overzicht hiervan kan gevonden worden in de documentatie van `SIunits`.

Eenheden Meestal komt het erop neer, te schrijven wat je zegt met een backslash ervoor: `\celsius \degree \arcminute \bbar` geeft °C°/bar. Let op, meter is `\metre` en liter is `\litre`.

Prefixen Kunnen gewoon geplaatst worden voor de eenheid: `\micro\metre` voor µm. Liever numeriek? Geen probleem, plaats er een d achter: `\microd\metre` voor 10⁻⁶m.

Machten Kan je ofwel in wiskundige modus zetten: `\metre^2` (m²), of voor tweede en derde machten via `\squaren\metre` (m²) `\cubic\second` (s³) `\newton\squared` (N²) of `\candela\cubed` (cd³). Gebruik `rp` (van *reciprocal*) vóór elk macht commando om de macht negatief te krijgen: `\rpcubic\metre` (m⁻³).

Samengestelde eenheden Gebruik `\per` als het volgende getal in de noemer komt en `\usk` als het daaropvolgende getal in de teller moet: `\watt\per\metre\usk\kelvin` voor W/m K.

Power Met `\power{grondtal}{exponent}` kan je machten maken: `\power{10}{100}` (10¹⁰⁰).

Hoofdstuk 4

Boxen, tabellen en figuren

4.1 Boxen

Een box is een stuk tekst dat door L^AT_EX als één enkel karakter behandeld wordt. Boxen worden gebruikt om speciale opmaakeffecten te bekomen. De meeste schrijvers hoeven geen boxen te gebruiken (met uitzondering van een `\mbox` om dingen bij elkaar te houden). Het is pas wanneer je rare dingen wilt doen, dat boxen van pas kunnen komen.

4.1.1 Links-rechts boxen

Links-rechts boxen bevatten de tekst van links naar rechts (zij printen dus al hun tekst af op één enkele lijn). De syntax is:

```
\mbox{tekst}
\fbbox{tekst}
\makebox[breedte][tekstpositie]{tekst}
\framebox[breedte][tekstpositie]{tekst}
```

Met `\mbox` wordt de tekst in een box gezet. Om een kadertje te krijgen rond een box, kan `\fbbox` gebruikt worden. Met `\makebox` (analoog aan `\mbox`) en `\framebox` (analoog aan `\fbbox`), kan de breedte van de box meegegeven worden. De tekstpositie geeft aan hoe de tekst in de box uitgelijnd wordt: `c` voor gecentreerd (standaard), `l` voor links uitgelijnd, `r` voor rechts uitgelijnd en `s` om de tekst uit te vullen. Een voorbeeld:

```
-----\framebox[2mm][c]{centrum}-----
```

geeft `centrum`. We zien dat L^AT_EX ervan uitgaat dat de box slechts 2mm breed is, ook al is de tekst eigenlijk breder (om dit te illustreren, werden rondom de box gedachtenstreepjes aangebracht).

Boxen kunnen ook naar boven verplaatst worden:

```
\raisebox{1ex}{Duc in altum!}
\raisebox{-1ex}{Errare humanum est, perseverare in errore diabolicum.}
```

Het eerste argument bevat de hoogte waarmee de box moet verplaatst worden (in \LaTeX -eenheden, zie bladzijde 26), het tweede bevat de tekst. Het resultaat van het eerste voorbeeld is: Duc in altum!. Het tweede voorbeeld toont aan dat er ook gewerkt kan worden met negatieve hoogten: Errare humanum est, perseverare in errore diabolicum.

4.1.2 Parboxen en minipagina's

Om hele paragrafen in een box te zetten, kan gebruik gemaakt worden van het `\parbox` commando en de `minipage` omgeving. Parboxen worden gebruikt voor kortere stukken tekst, minipages voor langere stukken. Maar eigenlijk doen ze allebei hetzelfde. Hun syntax is als volgt:

```
\parbox[positie][hoogte][verticale-tekstpositie]{breedte}{tekst}

\begin{minipage}[positie][hoogte][verticale-tekstpositie]{breedte}
tekst
\end{minipage}
```

Argumenten tussen vierkante haakjes zijn niet verplicht. De betekenis van de argumenten is de volgende:

positie De positie van de box ten opzichte van de omliggende tekst. Met **b** hangt de box uit de tekstlijn, met **t** ligt ze erop en met **c** zit ze er tussenin. Dit laatste is de standaard. Meer uitleg over de *positie* parameter is te vinden in de sectie over tabellen (sectie 4.2.1).

hoogte Hier kan de hoogte van de box ingegeven worden (in \LaTeX eenheden, zie bladzijde 26).

verticale-tekstpositie Deze parameter kan alleen worden meegegeven wanneer ook de vorige werd gegeven en bepaalt hoe de tekst in de box wordt geplaatst (ervan uitgaande dat de box te groot is). Er zijn vier mogelijkheden:

- t** De tekst wordt tegen de top van de box geduwd:

Plafondplakker

, gemaakt met `\parbox[c][4ex][t]{15ex}{Plafondplakker}`
- b** De tekst wordt tegen de vloer van de box geduwd:

Landpladijs

, gemaakt met `\parbox[c][4ex][b]{12ex}{Landpladijs}`
- c** De tekst wordt verticaal gecentreerd in de box:

In medio virtus

, gemaakt met `\parbox[c][4ex][c]{16ex}{In medio virtus}`

- s De tekst wordt uitgerekt zodat de hele box wordt bestreken. Er moet minimum één rubberen lengte voorkomen (zie sectie 3.5.1 op bladzijde 26) zodat de tekst daar kan uitgerekt worden.

breedte Hierin moet (verplicht argument) de breedte van de box worden ingegeven.

4.1.3 Nesten van boxen

De verschillende boxen kunnen in elkaar genest worden. Om bijvoorbeeld een kadertje te krijgen rondom een `\parbox`, zet je die in een `fbox`:

```
\fbox{\parbox{30em}{...}}
```

Het commando hierboven is wat gebruikt werd voor het genereren van deze box, met uitzondering van de drie puntjes. Maar indien we de drie puntjes willen vervangen door de letterlijke tekst die in deze box voorkomt, krijgen we deze cursus nooit af.

Om een box te maken die dezelfde breedte heeft als de lopende tekst, kan het commando `\textwidth` handig zijn. Dat bevat namelijk de breedte van de tekst:

```
\fbox{\parbox{\textwidth}{...}}
```

Deze breedte kan ook vermenigvuldigd worden met een getal:

```
\fbox{\parbox{0.8\textwidth}{...}}
```

4.2 Tabellen

4.2.1 Opbouwen van tabellen

De omgevingen `tabular`, `tabular*` en `array` vormen de basis van waaruit elke tabel wordt opgebouwd. Hun syntax is als volgt:

<code>\begin{array}[positie]{kols}</code>	rijen	<code>\end{array}</code>
<code>\begin{tabular}[positie]{kols}</code>	rijen	<code>\end{tabular}</code>
<code>\begin{tabular*}[breedte][positie]{kols}</code>	rijen	<code>\end{tabular*}</code>

De `array` omgeving kan enkel toegepast worden in *math mode* (zie hoofdstuk 6), maar wordt hier vermeld omdat de syntax dezelfde is als voor de `tabular` omgeving. De betekenis van de verschillende gebruikte argumenten is:

positie Elke tabel wordt door L^AT_EX beschouwd als één grote letter. Het is een nogal grote ‘letter’, vandaar dat die een bepaalde positie moet krijgen op de lijn: **top**, **bottom** of **centreren**.

- t** De top van de tabel is gealigneerd met de basistekstlijn. Dit betekent dat de tabel uit de tekstlijn ‘hangt’

galg
galg

- b** De onderkant van de tabel is gealigneerd met de basistekstlijn. De tabel schiet naar

boven,

pisa
pisa

, zoals dit voorbeeld.

- c** Het midden van de tabel is gealigneerd met de basistekstlijn, zoals te zien is in dit

voorbeeld:

pisa
galg

Het positie-argument is optioneel (te zien aan de vierkante haakjes). Wanneer het niet wordt meegegeven, wordt standaard het midden van de tabel gealigneerd met de basistekstlijn (de `c` optie).

breedte Dit speelt alleen maar mee in de `tabular*` omgeving. Het bepaalt de totale breedte van de tabel. In dit geval moet het `kols` argument minstens één `@{extracolsep{\fill}}` bevatten. Bij de twee andere tabellen wordt de breedte bepaald door de inhoud.

kols Voor elke kolom moet worden aangegeven of de tekst rechts of links uitgelijnd wordt, of gecentreerd wordt. Verder kan het ook nodig zijn om de randen van de tabel te versieren, of om extra spatiëring (of speciale tekens) te voorzien tussen de kolommen. Dit wordt allemaal meegegeven in het `kols` argument. De positie van de tekst in de kolommen kan bepaald worden met de volgende formatteringssymbolen:

l De tekst wordt links uitgelijnd.

r De tekst wordt rechts uitgelijnd.

c De tekst wordt gecentreerd.

p{breed} De tekst wordt uitgevuld over een breedte ‘breed’, die uitgedrukt wordt in de typische L^AT_EX lengte eenheden (zie bladzijde 26).

***{aantal}{kols}** Luiheid is de moeder van de vooruitgang. Als je de formattering van 10 dezelfde kolommen moet opnieuw schrijven, is dat nogal lastig. Gemakkelijker is als je die één keer ingeeft en dan zegt hoeveel keer je die kolom wilt hebben. Bijvoorbeeld `rlrlrlrlrlrl` kan korter geschreven worden als `*{5}{r1}`

Soms is het gewenst dat in elke rij een bepaald symbool terugkomt (zoals een verticale lijn, of een punt van de kommagetallen). Hiervoor kan het volgende in het `kols` argument gestoken worden.

| Zet een verticale lijn op die plaats.

`|` | Zet een dubbele verticale lijn op die plaats. Merk op dat verticale lijnen typografisch nooit zeer mooi zijn in een tabel, en dat ze meestal niet nodig zijn voor de leesbaarheid. Zo weinig mogelijk gebruiken dus.

`@{text}` Zet op die plaats in elke rij de tekst ‘text’. Dit is een zogenaamde @-uitdrukking. \LaTeX zet tussen elke kolom witte ruimte (zodat de tekst van de verschillende kolommen niet aaneen plakt). Met een @-uitdrukking valt deze witte ruimte weg. Als we die willen bewaren, moeten we in de @-uitdrukking een witte ruimte opnemen (met `\hspace{length}`). Langs de andere kant, als deze witte ruimte niet wenselijk is tussen bepaalde kolommen, kan ze worden teniet gedaan met `@{}`.

Met `@{\extracolsep{breed}}` wordt er in elke kolom die volgt op deze uitdrukking, een witte ruimte van breedte `breed` ingevoegd totdat een andere `@{\extracolsep{breed}}` ingegeven wordt. Deze extra spatiëring wordt niet verwijderd wanneer andere @-uitdrukkingen gebruikt worden. In een `tabular*` omgeving moet er ergens een `@{\extracolsep{\fill}}` voorkomen, zodat de breedte van de daaropvolgende kolommen kan uitgerekte worden opdat de tabel de voorgedefinieerde breedte kan aannemen.

rijen Hier worden de verschillende rijen van de tabel ingegeven. Elke rij wordt afgesloten met `\\` (twee rechten met een negatieve richtingscoëfficiënt) of met het commando `\tabularnewline`. De verschillende kolommen in een rij worden gesplitst door `&` tekens. Elke rij bevat juist evenveel kolommen als het aantal gedefinieerd bij het begin van de tabel. Sommige elementen mogen leeg zijn (aangegeven door twee opeenvolgende `&` tekens). In de rijen kunnen bepaalde speciale commando’s gegeven worden om horizontale lijnen te trekken en om cellen samen te nemen.

`\hline` Dit commando mag alleen geplaatst worden in het begin van een rij. Het zorgt voor een horizontale lijn over de volledige breedte van de tabel. Twee `\hline` commando’s na elkaar zorgen voor een dubbele horizontale lijn.

`\cline{m-n}` Dit commando plaatst een horizontale lijn vanaf het begin van kolom `m` tot het einde van kolom `n`. Het mag alleen gebruikt worden in het begin van een rij. Het mag meerdere keren gebruikt worden op dezelfde rij.

`\vline` Dit commando plaatst een verticale lijn met dezelfde hoogte als de rij. Op die manier kunnen verticale lijnen verkregen worden die niet doorlopen over heel de hoogte van de tabel.

`\multicolumn{aantal}{kols}{text}` Dit commando combineert het ‘aantal’ kolommen volgend op de plaats waar het gegeven wordt, in één kolom. Deze kolom moet juist één specificatie krijgen zoals gedefinieerd in `kols`. Het kan ook gebruikt worden om in juist één cel, de tekstuitlijning te wijzigen. Hiertoe wordt ‘aantal’ gelijk aan 1 gekozen.

4.2.2 Stijlparameters voor tabellen

De stijlparameters die L^AT_EX gebruikt voor het genereren van tabellen kunnen door de gebruiker gewijzigd worden. Deze parameters worden ofwel gewijzigd in de *preamble*, zodat ze gelden voor het hele document, ofwel lokaal rond de `tabular` omgeving (in een omgeving rondom). Echter nooit in de `tabular` omgeving zelf. Bijvoorbeeld zo:

```
{                                % Begin omsluitende omgeving
\setlength{\tabcolsep}{5pt}    % Spatie tussen kolommen
\begin{tabular}{ccc}           % Begin tabular omgeving
    een & twee & drie        % De rijen
\end{tabular}                  % Einde tabular omgeving
}
```

De verschillende lengteparameters die kunnen gewijzigd worden, zijn:

- `\tabcolsep` De helft van de ruimte die tussen de verschillende kolommen wordt aangebracht. Geldt enkel voor de `tabular` en `tabular*` omgeving.
- `\arraycolsep` Analooeg als `\tabcolsep` maar dan voor de `array` omgeving.
- `\arrayrulewidth` De dikte van de horizontale en verticale lijnen in een tabel.
- `\doublerulesep` De witte ruimte tussen twee opeenvolgende lijnen.

Het wijzigen van de afstand tussen de verschillende rijen, gebeurt met

```
\renewcommand{\arraystretch}{factor}
```

Hierbij is `factor` standaard gelijk aan 1. Dezelfde uitleg als voor `\baselinestretch` (bladzijde 29) geldt hier.

4.2.3 Voorbeelden

Tabellen produceren lijkt moeilijk, maar is zeer gemakkelijk. Eigenlijk schrijf je gewoon hoe de tabel er moet uit zien. We zullen nu enkele voorbeelden geven. Links geven we de tabel, rechts de L^AT_EX code nodig om de tabel te genereren.

Decimale getallen, het gebruik van de @-uitdrukking

Het wetenschappelijk ordenen van decimale getallen in een kolom, eist dat de decimale tekens onder elkaar vallen. Dit kan als volgt verkregen worden:

987.75025	<code>\begin{tabular}{r@{.}l}</code>
	<code>987 & 75025\\</code>
514229.13	<code>514229 & 13</code>
	<code>\end{tabular}</code>

De eerste kolom wordt rechts uitgelijnd, dan wordt een punt geplaatst (de @-expressie zorgt daarvoor) en tenslotte worden de cijfers van de derde kolom links tegen het punt geduwd.

Een @-uitdrukking kan ook gebruikt worden om telkens hetzelfde te laten verschijnen in een kolom:

	<code>\begin{tabular}{l@{ = }r}</code>
F10 = 55	<code>F10 & 55\\</code>
F11 = 89	<code>F11 & 89</code>
	<code>\end{tabular}</code>

Merk echter op dat een kolom waarin veel keer hetzelfde voorkomt, kan vervangen worden door extra uitleg in het bijschrift van de tabel.

Samenvoegen van cellen

Voor bijvoorbeeld multidimensionale tabellen kan het samenvoegen van cellen handig zijn:

		<code>\begin{tabular}{lrrr}</code>
	Metingen	<code>& \multicolumn{3}{c}{Metingen} \\</code>
F10	55 54 56	<code>F10 & 55 & 54 & 56 \\</code>
F11	89 90 88	<code>F11 & 89 & 90 & 88</code>
		<code>\end{tabular}</code>

De eerste cel van de eerste rij is leeg. De drie volgende cellen worden samen genomen en vervangen door één cel waarbinnen de tekst gecentreerd staat (die {c}). Met kommagetallen kan dit ook, dit bewijst het volgende voorbeeld waarin ook het gebruik van `*{aantal}{kols}` wordt getoond:

		<code>\begin{tabular}{l*{3}{r@{.}l}}</code>
	Metingen	<code>& \multicolumn{6}{c}{Metingen} \\</code>
F10	55.3 54.5 56.8	<code>F10 & 55 & 3 & 54 & 5 & 56 & 8 \\</code>
F11	89.13 90.21 88.34	<code>F11 & 89 & 13 & 90 & 21 & 88 & 34</code>
		<code>\end{tabular}</code>

We beginnen met een links uitgelijnde kolom. We doen verder met `*{3}{r@{.}l}`. Dit zijn in totaal 6 kolommen. Drie keer een rechtsuitgelijnde kolom, gevolgd door een punt, gevolgd door een linksuitgelijnde kolom. Daarom dat we bij `\multicolumn` aangeven dat we 6 kolommen wensen samen te voegen (het punt telt niet mee bij de kolomtelling).

Het samenvoegen van cellen in verticale richting is niet direct mogelijk. Er bestaan natuurlijk wel omwegen: het is perfect toegelaten om een tabel binnen een andere tabel te gebruiken (geneste tabellen):

Grootheid (mm)	Metingen		
F10	55.3	54.5	56.8
F11	89.13	90.21	88.34

```

\begin{tabular}{l*{3}{r@{.}l}}
\begin{tabular}{c}
Grootheid \\\ (mm) \\\
\end{tabular}
& \multicolumn{6}{c}{Metingen} \\\
F10 & 55 & 3 & 54 & 5 & 56 & 8 \\\
F11 & 89 & 13 & 90 & 21 & 88 & 34
\end{tabular}

```

Het eerste element is een tabelletje op zich. Terloops zien we ook dat elke rij niet op één tekstlijn moet staan. Voor grote tabellen is het zeer handig om de verschillende cellen op verschillende lijnen te plaatsen. Verder valt op dat in de minitabel, de verschillende kolommen in de input niet op verschillende lijnen staan. We kunnen de tabel zelfs op één lijn zetten in de broncode (dit wordt gedaan in het volgende voorbeeld). De opmaak in het bronbestand draagt niet bij tot de finale opmaak in het *pdf*-bestand.

Lijntjes trekken

In dit laatste voorbeeld tonen we hoe we gemakkelijk lijnen in onze tabel kunnen plaatsen:

Grootheid (mm)	Metingen		
F10	55.3	54.5	56.8
F11	89.13	90.21	88.34

```

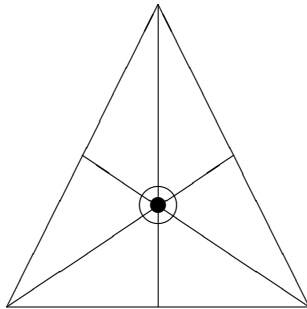
\begin{tabular}{l|*{3}{r@{.}l}}
\hline \hline % Horizontale dubbele lijn
\begin{tabular}{c}
Grootheid\\(mm)\\ \end{tabular}
& \multicolumn{6}{c}{Metingen} \\\
\hline % Horizontale lijn
F10 & 55 & 3 \vline & 54 & 5 & 56 & 8 \\\
\cline{5-6} % Lijntje op kolom 5 en 6
F11 & 89 & 13 & \vline & 90 & 21 & 88 & 34 \\\
\hline \hline % Horizontale dubbele lijn
\end{tabular}

```

De tabel wordt omsloten door dubbele horizontale lijnen, geproduceerd door telkens twee keer na elkaar het commando `\hline` te geven. Na de hoofding volgt een enkele lijn (één keer het commando `\hline`). Het valt op de merken dat er na een `\hline` commando geen `\\` moet volgen. Met `\cline{5-6}` zorgen we voor een horizontale lijn in kolom 5 en 6. Dit commando moet gegeven worden in het begin van een rij en heeft invloed op de bovenkant van die rij. Door hier en daar `\vline` te gebruiken, kunnen we midden in de tabel van die verticale lijntjes trekken. Merk op dat die verticale lijnen nogal tegen de tekst plakken, veel meer dan wanneer een verticale lijn in de kolomspecificatie wordt gedefinieerd: `l|` in `\begin{tabular}{l|*{3}{r@{.}l}}`.

4.3 Figuren

Er bestaan L^AT_EX commando's om rechtstreeks figuren te tekenen:



```
\setlength{\unitlength}{1mm}
\begin{picture}(40,40)
\put(0,0){\line(1,0){40}}
\put(0,0){\line(1,2){20}}
\put(0,0){\line(3,2){30}}
\put(20,0){\line(0,1){40}}
\put(40,0){\line(-1,2){20}}
\put(40,0){\line(-3,2){30}}
\put(20,13.5){\circle{5}}
\put(20,13.5){\circle*{2}}
\end{picture}
```

Maar deze zijn niet handig in gebruik. Vandaar dat we ze hier ook niet gaan beschrijven. We zullen ons meer toespitsen op het importeren van figuren. Wanneer we met **pdf_latex** werken, en dus rechtstreeks pdf's produceren (zie bladzijde 4), kunnen we verschillende soorten figuren importeren: **pdf** (ja, figuren kunnen ook bewaard worden in pdf's, dit is trouwens het te verkiezen formaat, daar het vectorgericht is), **png** (beste formaat voor pixelfiguren), **jpg** ... Wanneer we toch om één of andere reden met **latex** werken, kunnen we alleen **eps** bestanden invoegen.

Er bestaan twee pakketen voor het invoegen van figuren: **graphics** en **graphicx**. Het eerste is iets verouderd. Daarom dat we hier alleen het tweede bespreken. Voor meer details verwijzen we naar het bestand *grfguide.ps*,¹ te vinden in de meeste L^AT_EX distributies.

Om figuren te kunnen invoegen in het document, moet de volgende regel voorkomen in de *preamble* (let op de **x** op het einde van **graphicx**):

```
\usepackage{graphicx}           % Om figuren te kunnen verwerken
\usepackage[draft]{graphicx}    % Draft mode
```

Twee opties kunnen meegegeven worden: **draft** en **final**. In *draft* mode wordt lege ruimte voorzien daar waar de figuren moeten komen, maar worden de figuren zelf niet ingevoegd. Wanneer je bezig bent met het goedzetten van tekst en de figuren voorlopig minder belangrijk zijn, kun je die optie aanzetten: dit versnelt het compileerproces danig. De optie **final** zorgt ervoor dat alle figuren zeker in het pdf document verschijnen.

Het invoegen van een figuur in het document gebeurt met (hier een **s** op het einde van **\includegraphics**):

¹Op een Debian systeem te vinden in: `/usr/share/doc/texmf/latex/graphics/grfguide.ps.gz`

```
\includegraphics{figuurnaam.png}
```

De extensie moet niet altijd meegegeven worden. L^AT_EX zoekt zelf naar *figuurnaam.pdf*. Als dat niet gevonden wordt zoekt de compiler naar *figuurnaam.png* en zo verder. In de *preamble* kunnen we eventueel opgeven naar welke extensies moet gezocht worden:

```
\DeclareGraphicsExtensions{pdf, png, jpg, jpeg}
```

Merk op dat het niet is omdat L^AT_EX de figuur vindt, dat ze ook verwerkt kan worden. Sommige figuren kunnen nu eenmaal niet geïmporteerd worden. Zet ze met een ander programma om in bijvoorbeeld png of pdf formaat².

Bij het invoegen van figuren zijn er een aantal opties die onontbeerlijk zijn:

width Is een L^AT_EX lengte (zie bladzijde 26 voor meer uitleg) en geeft aan hoe breed onze figuur moet zijn. Indien geen **height** wordt opgegeven, behoudt de figuur zijn breedte–hoogte verhouding.

height Is een L^AT_EX lengte en bepaalt hoe hoog onze figuur moet zijn. Indien geen breedte wordt opgegeven, behoudt de figuur zijn breedte–hoogte verhouding.

angle Is een getal dat aangeeft hoeveel graden de figuur moet gedraaid worden. De hoogte en de breedte draaien mee. Indien dus een hoogte of breedte wordt opgegeven en dan wordt een hoek van 90 graden gegeven, zal de hoogte de breedte worden en vice versa.

totalheight De totale hoogte van de figuur: wanneer een figuur gedraaid wordt, neemt de totale hoogte bijna altijd toe (vraag maar aan Pythagoras). Met **height** wordt alleen de hoogte van de niet geroteerde figuur ingesteld terwijl met **totalheight** de totale hoogte wordt ingesteld.

Al deze opties worden tussen vierkante haken ingegeven via de *key=value* notatie:

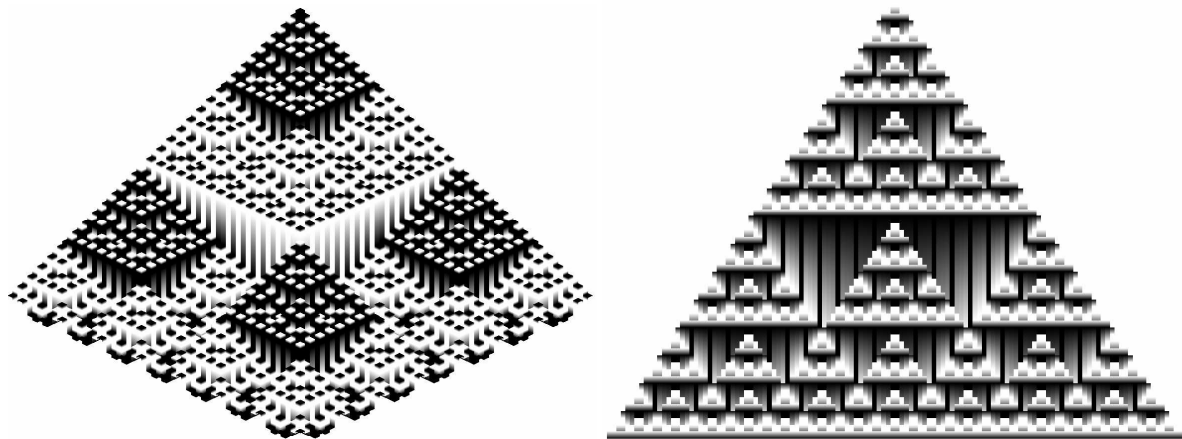
```
\includegraphics[width=7cm]{cheops}
\includegraphics[width=0.5\textwidth]{chephren}
```

We zien dat we gebruik kunnen maken van de speciale lengte `\textwidth`, die de breedte van de tekst weergeeft. De tweede figuur wordt even breed als de helft van de tekstbreedte. Het resultaat van deze commando's is te bezichtigen in figuur 4.1.

Figuren bewaren we best niet in de hoofddirectory van onze thesis. We maken bijvoorbeeld een subdirectory *figuren* aan. Zodus wordt het bovenstaande commando

```
\includegraphics{figuren/figuurnaam}
```

²Onder Unix is er het handige `convert` van de ImageMagick software. Voor het omzetten van eps naar pdf is echter `epstopdf` aangewezen ...



Figuur 4.1: Een figuur zegt meestal meer dan duizend woorden.

Niet zeer handig om altijd diezelfde directory te moeten intypen. Vandaar dat je in de *preamble* kan opgeven in welke directories L^AT_EX naar figuren moet gaan zoeken:

```
\graphicspath{{figuren/}{nog_een_dir_met_figuren/}}
```

Wanneer de L^AT_EX compiler een figuur moet invoegen, gaat hij nu zoeken in de directory *figuren/* en in *nog_een_dir_met_figuren/*. Merk op dat ook onder Windows de slash met een positieve richtingscoëfficiënt moet gebruikt worden (en niet de *backslash*).

4.4 Zwevende tabellen en figuren

4.4.1 Zwevend maken

Met de `tabular` omgeving wordt de tabel daar geplaatst waar ze gedefinieerd werd. Met `\includegraphics` wordt de figuur daar geplaatst waar het commando gegeven werd. Als een tabel niet op de huidige bladzijde past, wordt een nieuwe bladzijde begonnen. Hetzelfde bij figuren. Dit kan zorgen voor nogal wat halflege bladzijden. Vandaar dat het beter is om de L^AT_EX compiler te laten beslissen waar de tabellen en figuren moeten geplaatst worden. Dit gebeurt via het zogenaamde **float** mechanisme. De tabellen en figuren worden in een *float* omgeving geplaatst.

Voor tabellen wordt de `table` omgeving gebruikt:

```
\begin{table}[waar]
  \begin{tabular}{ccc}
    ... % tabeltekst
  \end{tabular}
\end{table}
```

Het genereren van de tabel zelf gebeurt zoals uitgelegd in sectie 4.2. Figuren worden in een `figure` omgeving geplaatst:

```
\begin{figure}[waar]
  \includegraphics[width=5cm]{vbfig01}
\end{figure}
```

Op zich doen `table` en `figure` juist hetzelfde: ze nemen de tabel of figuur vast en zetten die op een typografisch verantwoorde plaats. Met het optionele argument (bemerkt de vierkante haken) `waar` kunnen we onze voorkeur uitdrukken over waar we de figuur graag zouden hebben. Het argument `waar` kan uit verschillende letters bestaan. Deze letters zijn de volgende:

h (Hier) De figuur of tabel mag op de plaats komen waar hij in de brontekst wordt ingegeven.

t (Top) De figuur of tabel mag aan de bovenkant van de huidige bladzijde komen, of als daar niet genoeg plaats is, aan het begin van de volgende bladzijde.

b (Beneden) De figuur of tabel mag aan de onderkant van de huidige bladzijde komen. Als daar niet genoeg plaats meer is, wordt de onderkant van de volgende bladzijde gebruikt.

p (Pagina) De figuur of tabel wordt opgespaard tot het einde van het hoofdstuk of sectie, waar een aantal bladzijden worden voorzien met alleen maar figuren en tabellen.

! Normaalgezien respecteert \LaTeX een aantal regeltjes over de hoeveelheid *floats* op een bladzijde, het aantal *floats* in het begin van een pagina, en zo meer. Dit zorgt ervoor dat figuren en tabellen soms nogal ver van hun oorspronkelijke tekst geplaatst worden. Door een uitroepteken mee te geven met de positioneringsargumenten, zeggen we tegen \LaTeX om geen rekening te houden met die typografische regels. We riskeren dan wel bladzijden te krijgen waar enkel *floats* op staan.

H (Hier en nergens anders!) Soms willen we de tabel of figuur krijgen daar waar ze gedefinieerd wordt. Zelfs als daarvoor een nieuwe bladzijde moet begonnen worden. Deze optie is niet aanwezig in de standaard \LaTeX maar moet opgeladen worden met een nieuw pakket met de originele naam **float**:

```
\usepackage{float} % Om nieuwe float omgevingen aan te maken. Ook optie H!
```

Dit pakket laat ook toe om eigen float omgevingen te maken, maar hiervoor verwijzen we naar de excellente documentatie van het pakket zelf.³

Dus met `\begin{table}[ht!] tabel \end{table}` maken we een zwevende tabel die hier geplaatst wordt en als dat niet gaat, op de top van de volgende bladzijde. \LaTeX moet zich ook niets aantrekken van typografische beperkingen in verband met *floats*.

³Op een Debian systeem: `/usr/share/doc/texmf/latex/styles/float.dvi.gz`

4.4.2 Bijschrift

Het is altijd leuk als de lezers wat extra uitleg krijgen bij een tabel of figuur. Dit versieren gebeurt met het commando `\caption{uitleg}`. Bij de `figure` omgeving genereert dit de titel ‘Figuur 1.3: uitleg’ en bij de `table` omgeving ‘Tabel 1.3: uitleg’. Traditioneel wordt een figuur versierd met een onderschrift en een tabel met een bovenschrift. Daarom dat bij een tabel het `\caption` commando wordt gegeven vóórdat de `tabular` omgeving begint en bij een figuur ná de `\includegraphics`.

Standaard is het bijschrift in L^AT_EX niet echt te onderscheiden van de rest van het document (zelfde lettertype, zelfde lettergrootte, geen inspringing). Om hieraan te verhelpen, kan het pakket `caption` gebruikt worden. Voor de uitgebreide mogelijkheden die dit pakket biedt, verwijzen we naar de uitstekende documentatie.⁴

Om `caption` te gebruiken, is de volgende lijn nodig in de *preamble*:

```
\usepackage[opties]{caption}
```

Hierbij zijn het `opties` die de bijschriften van onze tabellen en figuren bepalen. De mogelijkheden zijn:

Stijl De volgende argumenten kunnen gebruikt worden om te bepalen hoe het bijschrift wordt uitgelijnd.

normal Volle regels worden uitgemiddeld. De laatste lijn wordt links uitgelijnd. Dit is zoals gewone tekst.

flushleft Alle regels worden tegen de linkerkantlijn geduwd.

flushright Alle regels worden tegen de rechterkantlijn geduwd.

center Alle regels worden gecentreerd.

centerlast Alle regels worden uitgemiddeld. De laatste lijn wordt gecentreerd.

hang Hetzelfde als **normal**, behalve dat vanaf de tweede regel van het bijschrift de tekst inspringt met een lengte gelijk aan die van de titel van het bijschrift.

noonline Al de bovenvermelde uitlijningsopties hebben alleen betrekking op bijschriften die meerdere regels tellen. Wanneer er slechts één regel tekst is, wordt het bijschrift gecentreerd. Om dit tegen te gaan, kan de optie **noonline** meegegeven worden. Is niet echt aan te raden wanneer je je tabellen en figuren centreert ...

Lettergrootte De lettergrootte voor het bijschriftlabel (‘Figuur 1.3’) en de bijschrifttekst kan bepaald worden door één van de volgende opties mee te geven: `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`. Uitleg over de betekenis hiervan is te vinden op bladzijde 28.

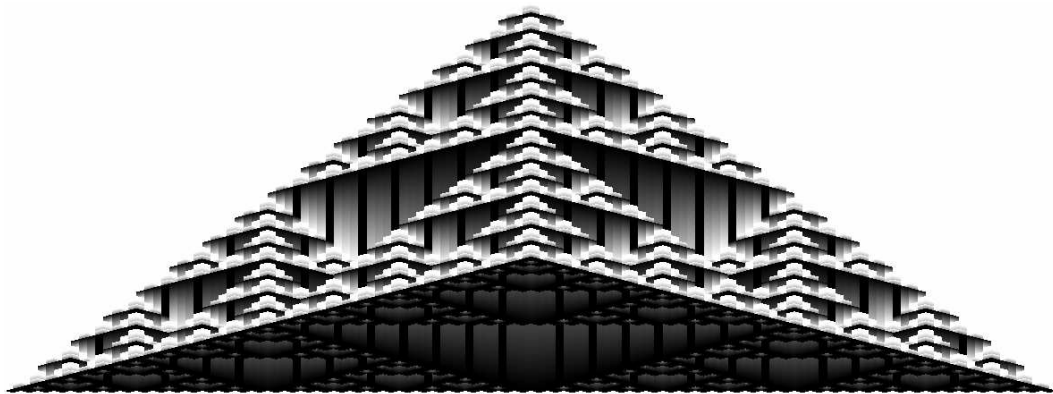
⁴Op een Debian systeem: `/usr/share/doc/texmf/latex/caption/caption.pdf.gz`.

Lettertype van het bijschriftlabel Met de volgende opties kan het lettertype van het bijschriftlabel veranderd worden: `sc` voor SMALL CAPS, `it` voor *cursief*, `sl` voor *slanted* en `up` voor rechte tekst. Met `bf` wordt het label in het vet gezet. Om Sans Serif als lettertype te gebruiken, kan `sf` als optie meegegeven worden.

Voor dit document werd `caption` opgeroepen met de volgende argumenten:

```
\usepackage[small,bf,hang]{caption} % Om de captions wat te verbeteren
```

Het bijschrift wordt iets kleiner afgeprint dan gewone tekst. Het label staat in het vet en vanaf de tweede lijn springt de tekst in. Als voorbeeld is er figuur 4.2.



Figuur 4.2: We hadden ook de demonstratie kunnen doen met een tabel, maar tabellen bevatten meestal slechts woorden en volgens figuur 4.1 zegt een figuur meer dan duizend woorden. Dus hoewel dit bijschrift lang lijkt (nodig om dat inspringen te illustreren), is de figuur nog veel langer.

4.4.3 Verwijzingen naar tabellen en figuren

Het is handig wanneer we kunnen verwijzen naar onze tabellen en figuren. Dit gebeurt met het mechanisme uitgelegd in sectie 2.5 op bladzijde 19. We plaatsen een `\label{labeltjefig}` in de `caption`. Let erop dat dit wel degelijk in de `caption` is en niet erbuiten. Dit is logisch: de `caption` zorgt voor nummering van de figuur; geen `caption`, geen nummering, geen mogelijkheid om de tabel aan te wijzen. Met `\ref{labeltjefig}` of `\pageref{labeltjefig}` kunnen we refereren naar de *float*.

In wetenschappelijke teksten is het gebruikelijk dat tabellen en figuren nooit verschijnen voordat ze in de tekst vermeld worden. \LaTeX durft dit nogal eens wel te doen. Wanneer je op het einde van een bladzijde een figuur invoegt, en je hebt er juist voor het invoeren naar verwezen, kan het zijn dat die figuur bovenaan de bladzijde geplaatst wordt, dus voordat ernaar verwezen wordt. \LaTeX gaat dus tekst die in het bronbestand vóór de figuur staat, in

het finale pdf-document ná de figuur plaatsen. Om dit te vermijden, moet de volgende lijn in de *preamble* opgenomen worden:

```
\usepackage{flafter}          % Opdat floats niet zouden voorsteken
```

Let er dan wel op dat je eerst naar een figuur of tabel verwijst, vooraleer ze in te brengen in de tekst.

4.4.4 Enkele voorbeelden

Het is mooi wanneer onze tabellen en figuren gecentreerd staan. Daarom dat we de *floats* in een `center` omgeving zetten. Dat centreren moet gebeuren binnen de *float* omgeving (zie voorbeeld hieronder).

De code nodig om figuur 4.2 te genereren is:

```
\begin{figure}[ht]
\begin{center}
\includegraphics[width=0.9\textwidth]{mycerinus}
\caption{We hadden ook de demonstratie kunnen doen met een tabel, maar
tabellen bevatten meestal slechts woorden en volgens figuur \ref{cheops-
chephren} zegt een figuur meer dan duizend woorden. Dus hoewel dit
bijschrift lang lijkt (nodig om dat inspringen te illustreren), is de
figuur nog veel langer.\label{mycerinus}}
\end{center}
\end{figure}
```

Voor tabellen gebruiken we de `table` omgeving. In plaats van een `\includegraphics`, komt er nu een `tabular` omgeving. Eventueel kunnen we de `tabular` in een apart bestand plaatsen en in onze `table` omgeving invoegen met een `\input{tabelbestandsnaam}` commando. Vergeet ook niet om de `caption` vóór de eigenlijke tabel te definiëren. Om tabel 3.2 op bladzijde 30 te genereren werd de volgende code gebruikt:

```
\begin{table}[h]
\begin{center}
\caption{Hoe verschillende letterstijlen selecteren: met een nieuwe omgeving
of met een commando.\label{letterstijlen}}
\begin{tabular}{l@{\;}l@{\quad}l@{\;}l}
... % de verschillende rijen
\end{tabular}
\end{center}
\end{table}
```

Het bijschrift even breed krijgen als de *float* kan gebeuren met een `\parbox` (de uitleg over `parboxen` is te vinden op bladzijde 39):

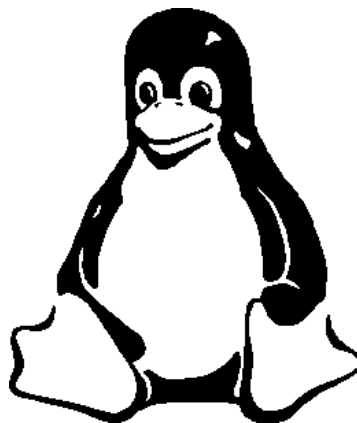
```

\begin{figure}[ht]
  \begin{center}
    \includegraphics[width=0.3\textwidth]{tux}
    % De volgende witte lijn is belangrijk: anders komt caption naast figuur!

    \parbox{0.3\textwidth}{\caption{Giza is dood. Leve de Penguin!\label{tux}}}
  \end{center}
\end{figure}

```

Het resultaat is te bewonderen in figuur 4.3.



Figuur 4.3: Giza is dood.
Leve de Pen-
guin!

4.4.5 Nieuw commando om gemakkelijk *floats* in te voegen

Altijd dat `\begin{center}`, `\caption`, `\label` intypen is nogal vermoeiend. Algauw ga je de instellingen van de vorige figuur kopiëren en wijzig je de figuurnaam, het label en het bijschrift. Lastig, je kan gemakkelijk fouten maken. Veel handiger is als je een commando maakt dat dat allemaal voor jou doet. Wat we eigenlijk wensen, is iets als:

```

\mijnfiguur[plaatsingsopties]{opties-includegraphics}{bestand}{Bijschrift}

```

Inderdaad, het is niet echt nodig om een label mee te geven. Als we ervan uitgaan dat elke bestandsnaam geen *slashes* bevat (zie bladzijde 48 over het ingeven van de directories waar \LaTeX moet zoeken naar figuren) en uniek is, kunnen we de bestandsnaam gebruiken als label. De plaatsingsopties bepalen waar de figuur moet terechtkomen (zie bladzijde 49); dit is een optioneel argument, daar we een standaardwaarde kunnen gebruiken voor het hele document. De opties voor `includegraphics` zijn verplicht. Meestal moet de figuur toch herschaald worden en is een `width=xxx` nodig. Dit kan moeilijk dezelfde zijn voor elke figuur.

Het commando `\mijnfiguur` wordt als volgt gedefinieerd (voor uitleg over het aanmaken van nieuwe commando's verwijzen we naar sectie 2.1.3 op bladzijde 10):

```
\newcommand{\mijnfiguur}[4][ht]{
  \begin{figure}[#1]
    \begin{center}
      \includegraphics[#2]{#3}
      \caption{#4\label{#3}}
    \end{center}
  \end{figure}
}
```

Het eerste argument bevat de opties die vertellen waar onze figuur moet geplaatst worden. Standaard proberen we figuren te plaatsen waar ze gedefinieerd worden. Als dat niet lukt, hebben we ze liefst bovenaan een bladzijde. Het tweede argument bevat de opties voor `includegraphics`. Het derde argument bevat de bestandsnaam, wat ook het label wordt, en het vierde argument bevat de tekst van het bijschrift.

Figuur 4.2, zou dus als volgt kunnen geplaatst worden in ons document:

```
\mijnfiguur{width=0.9\textwidth}{mycerinus}{We hadden ook de demonstratie...}
```

Wat veel eleganter is dan heel die rompslomp in sectie 4.4.4.

Iets analoogs kan gedaan worden voor tabellen. Er moeten nu geen opties meer meegegeven worden met `includegraphics` zodat we slechts drie argumenten hebben (merk ook op dat de caption vóór de tabelinput komt):

```
\newcommand{\mijntabel}[3][ht]{
  \begin{table}[#1]
    \begin{center}
      \caption{#3\label{#2}}
      \input{#2}
    \end{center}
  \end{table}
}
```

Ingeven van een tabel gebeurt dan als volgt:

```
\mijntabel[hb]{bestand-met-tabel}{Bijschrift bij de tabel.}
```

Hoofdstuk 5

Bibliografische verwijzingen

Het verwijzen naar literatuur is iets dat frequent gebeurt in wetenschappelijke documenten. En dat verwijzen moet volgens welbepaalde regels gebeuren, geen komma teveel of te weinig.¹ Maar er zijn leukere dingen om te doen dan te letten op punten en komma's. Laat het vervelende werk over aan de computer. Het systeem is in L^AT_EX zo opgebouwd dat je met het commando `\citet{auteurkey}` de naam van de auteur samen met het jaar krijgt en de volledige referentie achteraan in de bibliografie. Hiervoor moeten alle gebruikte referenties in een bibliografische databank zitten. In dit deel wordt kort overlopen hoe een bruikbare bibliografie kan verkregen worden voor een thesis. Voor meer opties en andere manieren van citeren (bijvoorbeeld met nummers in plaats van met auteur, jaar) verwijzen we naar de helpbestanden van B_IB_TE_X.²

5.1 Bib_TE_X, het programma

In feite gebeurt het genereren van de bibliografie door een extern programma, dat de bibliografische informatie uit een aantal databanken³ haalt. Daarom dat op de plaats waar we de bibliografie wensen in te voegen, het volgende commando moet gegeven worden:

```
\bibliography{bibliodatabase1, bibliodatabase2, ... }
```

De argumenten `bibliodatabase1` en `bibliodatabase2`, gescheiden door komma's, zijn de bestandsnamen van de databanken zonder de extensie `.bib`. Dus de volledige naam van de databanken luidt `bibliodatabase1.bib` en `bibliodatabase2.bib`. Meerdere bestanden mogen opgegeven worden. Dit opent perspectieven voor samenwerking ...

Het citeren in de tekst wordt uitgelegd in sectie 5.3. Om de bibliografie effectief in het pdf document op te nemen, laten we onze tekst een eerste keer verwerken door de L^AT_EX compiler.

¹Wellicht is het van literatuurverwijzingen dat de term 'kommaneuterij' komt

²Op een Debian systeem in `/usr/share/doc/texmf/bibtex/base/btxdoc.dvi.gz`

³Een duur woord voor platte tekstbestanden met speciale opmaak

Daarna geven we het commando:

```
bibtex naam-van-ons-document-zonder-extensie
```

In TeXnicCenter (de L^AT_EX editor onder windows) is er een knopje om BIB_TE_X op te roepen. Dit commando genereert dan twee bestanden: *ons-document.bbl* en *ons-document.blg*. Het *.blg*-bestand (van BIB_TE_X log) is analoog aan het L^AT_EX *.log*-bestand (zie bladzijde 5) en is niet echt belangrijk. Het *.bbl*-bestand (van BIB_TE_X BibLiografie) is te vergelijken met het L^AT_EX *.aux*-bestand en bevat informatie voor de L^AT_EX compiler om op een juiste manier de bibliografie te genereren.

Wanneer we ons document opnieuw door de L^AT_EX compiler laten verwerken (minimum twee keer, zie bladzijde 16), wordt de bibliografie in het document opgenomen.

Standaard produceert `\citet{auteurkey}` nummers in de tekst, zoals dit: [13]. Voor een thesis is het auteur-jaar mechanisme te verkiezen, zoals dit: ?. Hiervoor moeten we een aantal extra instructies meegeven in ons document. Vooreerst moet **natbib** gebruikt worden. Dit is een pakket dat de functionaliteit van BIB_TE_X vergroot.

```
\usepackage[round]{natbib} % Voor auteur-jaar citaties.
```

Deze regel zorgt ervoor dat **natbib** geladen wordt en ronde haakjes gebruikt. Om de punctuaties in te stellen kan eventueel de volgende lijn in de *preamble* worden opgenomen (maar de standaardwaarden zijn meestal wel goed):

```
\bibpunct{(}{)}{;}{y}{,}{,} % Auteur-jaar citaties, zie natbib.dvi voor uitleg
```

Er zijn zes argumenten, waarmee de volgende zaken kunnen aangepast worden:

1. Het ‘open de haakjes’ symbool. Standaard gelijk aan (.
2. Het ‘sluit de haakjes’ symbool, Standaard gelijk aan).
3. Het punctuatiesymbool tussen verschillende citaties. Standaard is dit ; Dit toepassen op een voorbeeld geeft ??.
4. De letter **n** voor citaties met nummers, **s** om de nummers in superscript te zetten en alle andere letters om auteur-jaar citatie te bekomen. Dit laatste wordt standaard gebruikt.
5. Het punctuatiesymbool dat gebruikt wordt tussen de auteur en het jaar in het geval dit laatste niet tussen haakjes wordt gezet. Standaard is dit , zoals in (?).
6. Het punctuatiesymbool dat verschijnt wanneer de auteurs van twee (of meer) artikels gelijk zijn. Door BIB_TE_X wordt dit dan gecombineerd tot 1999a,b. Standaard dus een komma.

Doordat citeren vooral belangrijk is in wetenschappelijke literatuur en dat wetenschappelijke literatuur meestal Engelstalig is, zet BIB_TE_X standaard *and* tussen de verschillende auteurs, bv Jones and Allen (2002). Voor een Nederlandse thesis is dit geen zicht. De oplossing is om

zelf een bibliografisch opmaakbestand te maken. De manier waarop `BIBTEX` de bibliografie verwerkt zit namelijk niet in het programma zelf ingebakken, maar wordt ingelezen uit een opmaakbestand.

Een nieuw `BIBTEX` opmaakbestand maken, kan gebeuren door het speciale bestand *makebst.tex* te `LATEX`en (met `latex` en niet met `pdflatex`). Dit bestand bevindt zich normaalgezien ergens in de `LATEX` installatie. Daar `LATEX` ook een programmeertaal is, kan het gebruikt worden om programma's te schrijven. Dit is wat het commando `latex makebst.tex` doet: we krijgen een vraag-en-antwoord-dialoog waarin we puntje voor puntje kunnen kiezen hoe onze bibliografie er moet uitzien. Al deze opties worden bewaard onder de naam *bibliodutch* (die bestandsnaam is het antwoord op één van de eerste vragen) met standaard extensie *.dbj*. Dit is opnieuw een `LATEX` programma (ja, ja, een programma dat een programma schrijft). Met `latex bibliodutch.dbj` wordt het bestand *bibliodutch.bst* gecreëerd. Dit is het bibliografisch opmaakbestand dat we nodig hebben.

Natuurlijk is het veel gemakkelijker om ergens zo een opmaakbestand van iemand te kopiëren. Het opmaakbestand dat voor dit document gebruikt werd, is vrij beschikbaar ...

Om dit opmaakbestand effectief te gebruiken, moet de volgende regel in de *preamble* toegevoegd worden:

```
\bibliographystyle{bibliodutch}
```

Verder moet het bestand *bibliodutch.bst* zich in de werkdirectory bevinden, zodat `BIBTEX` eraan kan.

Om ervoor te zorgen dat de bibliografie in de inhoudstafel wordt opgenomen (standaard gebeurt dit namelijk niet) moet het pakket `tocbibind` in de *preamble* opgeladen worden:

```
\usepackage[nottoc]{tocbibind} % Bibliografie in ToC; zie tocbibind.dvi
```

De optie `nottoc` zorgt ervoor dat de inhoudsopgave zelf niet in de inhoudsopgave komt.

5.2 De bibliografische databank

5.2.1 Opbouw van de databank

De bibliografische databank bestaat uit verschillende items. Elk item komt overeen met één referentie en is van de vorm:

```
@article{athos:92,
  author = {A. Athos and D. Porthos and D. Aramis},
  title = {The influence of Women on society and how to cope with},
  journal = {Journal of Random Logic},
  year = {1992},
  volume = {13},
```



```

    pages = {11-19}
}

```

Het eerste woord, vooraf gegaan door een @, bepaalt het type item dat we wensen in te voegen. In dit geval is dat een artikel. Dan worden de verschillende eigenschappen van het artikel opgegeven tussen accolades. De verschillende eigenschappen worden van elkaar gesplitst door komma's. Na de laatste eigenschap (in het voorbeeld `pages`) staat er dus geen komma. De eerste eigenschap is de **key** van die referentie. Dit is hetgeen gebruikt wordt om in de tekst naar dat artikel te refereren (e.g. `\citet{athos:92}` geeft ‘?’). De rest van de eigenschappen mag in gelijk welke volgorde voorkomen.

Samengevat hebben we dus: `@article{key, eig1, eig2, ..., eigN}` waarbij de verschillende eigenschappen de volgende vorm hebben: `eig1 = {waarde}` of `eig1 = "waarde"` (dubbele aanhalingstekens of accolades, je mag kiezen). Verder mag elke eigenschap op een nieuwe regel geplaatst worden. Dit bevordert de overzichtelijkheid.

Meestal komen vele artikels uit hetzelfde tijdschrift. Het is lastig om dan elke keer opnieuw de volledige naam van dat tijdschrift te moeten intypen. Vandaar dat we in het begin van onze `BIBTEX` database afkortingen mogen definiëren. Dit gebeurt als volgt:

```
@string{JRL = "Journal of Random Logic"}
```

Dit kunnen we dan als volgt gebruiken in de `BIBTEX` items:

```

...
    journal = JRL,
    year = {1992}
...

```

Merk op dat afkortingen die gedefinieerd zijn met `@string{...}` niet tussen accolades of aanhalingstekens mogen staan.

`BIBTEX` heeft de eigenschap om in titels (het `title` veld) alle hoofdletters om te zetten naar kleine letters. Om dit tegen te gaan (voor namen bijvoorbeeld), moeten de letters nog eens extra tussen accolades gezet worden. Dus

```

...
    title = {The influence of {Women} on society and how to cope with},
...

```

zorgt ervoor dat `Women` met een hoofdletter blijft staan in de bibliografie van ons finaal *pdf*-document. Verder mogen allerhande `LATEX` opmaak commando's gebruikt worden in de bibliografie:

```
...
    title = {The relation of \textit{E. coli} to humankind},
...
```

Het is aangeraden om accenten en andere specialiteiten tussen accolades te zetten, dus `{\"o}` en niet `\"o` om ergens in de bibliografie ö te produceren.

5.2.2 De verschillende items

Zoals in de vorige sectie werd beschreven, begint elk `BIBTEX` item met een `@` gevolgd door de naam van het documenttype waarnaar het item verwijst (boek, artikel, ...). Afhankelijk van wat het item is, kunnen verschillende eigenschappen gegeven worden. Niet elke eigenschap kan bij elk item gegeven worden. Het is bijvoorbeeld niet mogelijk om de *journal* mee te geven bij een boek. Een *journal* past meer bij een artikel. Verder zijn sommige velden verplicht, en andere optioneel. Bij een artikel moet bijvoorbeeld altijd de auteur gegeven worden, maar niet het tijdschriftnummer. De verschillende velden worden beschreven in de volgende sectie. Hier geven we een overzicht van de mogelijke items. Diegenen die dieper op het onderwerp willen ingaan, worden doorverwezen naar *btldoc.dvi*.

@article Om een artikel op te nemen.

Verplicht: `author, title, journal, year`.

Optioneel: `vol, number, pages, month, note`.

@book Een boek met een expliciete uitgever.

Verplicht: `author of editor, title, publisher, year`.

Optioneel: `volume of number, series, address, edition, month, note`.

@booklet Een boek zonder auteur of uitgever.

Verplicht: `title`.

Optioneel: `author, howpublished, address, month, year, note`.

@inbook Een deel van een boek (hoofdstuk, sectie).

Verplicht: `author of editor, title, chapter en/of pages, publisher, year`.

Optioneel: `volume or number, series, type, address, edition, month, note`.

@incollection Een deel van een boek met een eigen titel.

Verplicht: `author, title, booktitle, publisher, year`.

Optioneel: `editor, volume of number, series, type, chapter, pages, address, edition, month, note`.

@inproceedings Een artikel in de *proceedings* van een conferentie.

Verplicht: `author, title, booktitle, year`.

Optioneel: `editor, volume of number, series, pages, address, month, organization, publisher, note`.

- @manual** Voor technische documentatie zoals gebruiksaanwijzingen.
 Verplicht: **title**.
 Optioneel: **author, organization, address, edition, month, year, note**.
- @masterthesis** Voor een afstudeerthesis.
 Verplicht: **author, title, school, year**.
 Optioneel: **type, address, month, note**.
- @misc** De vuilbak, als de rest niet past.
 Verplicht: **Geen**.
 Optioneel: **author, title, howpublished, month, year, note**.
- @phdthesis** Voor een doctoraatsthesis. Eigenlijk hetzelfde als een afstudeerthesis ...
 Verplicht: **author, title, school, year**.
 Optioneel: **type, address, month, note**.
- @proceedings** Een publicatie voortvloeiend uit een conferentie.
 Verplicht: **title, year**.
 Optioneel: **editor, volume of number, series, address, month, organization, publisher, note**.
- @techreport** Voor een technisch of intern rapport.
 Verplicht: **author, title, institution, year**.
 Optioneel: **type, number, address, month, note**.
- @unpublished** Voor ongepubliceerd werk met titel en auteur.
 Verplicht: **author, title, note**.
 Optioneel: **month, year**.

Laat je niet afschrikken door dit overweldigend aanbod aan mogelijkheden. De meest gebruikte items zijn **article** en soms nog **book**. De rest is voor speciale literatuur. Verder moet de benaming gezien worden als hulpmiddel om te onthouden welke mogelijkheden er bestaan, maar moet dit niet strikt opgevolgd worden. Je mag zeker **masterthesis** gebruiken voor een Ph.D thesis. Beide items geven juist hetzelfde resultaat.

5.2.3 De verschillende eigenschappen

De verschillende eigenschappen die in de items kunnen ingevuld worden, zijn:

address Het adres van de uitgever.

author De auteurs.

booktitle De titel van het boek, wanneer slechts een deel van dat boek geciteerd wordt.

chapter Een hoofdstuk of sectienummer.

edition De editie van het boek. Bijvoorbeeld ‘Derde’. Het woordje ‘editie’ moet er niet achter komen, daar zorgt **BIBTEX** voor.

editor De uitgever van het boek, niet te verwarren met de auteur.

howpublished Hierin kunnen allerhande speciale mededelingen gestoken worden, bijvoorbeeld ‘Persoonlijke mededeling’.

institution In een technisch rapport komt hier de naam van de financierder.

journal De naam van het tijdschrift waaruit een artikel komt.

month De maand in dewelke het werk geschreven werd. Wordt zelden gebruikt.

note Extra informatie die toegevoegd moet worden aan de bibliografie. Moet beginnen met een hoofdletter.

number Het nummer van een tijdschrift of een ander werk dat in series wordt uitgegeven. Wetenschappelijke tijdschriften hebben meestal een volume en een nummer. De bladzijdetelling loopt meestal door in de verschillende nummers van eenzelfde volume. Zodus is het niet nodig om ook het nummer van een tijdschrift mee te geven in de bibliografie.

organization De sponsor van de conferentie.

pages De bladzijden bestreken door het werk. Voor tijdschriften is het gebruikelijk om een bereik mee te geven: 99–102.

publisher De naam van de uitgever.

school De naam van de instelling waar de thesis geschreven werd.

series De naam van de serie. De titel bevat dan de naam van het boek zelf.

title Titel van het werk.

type Het type werk, bijvoorbeeld ‘Beleidsnota’.

volume Het volumenummer van een tijdschrift. Is niet verplicht maar sterk aangeraden.

year Het jaar van publicatie.

Ook hier moet men zich niet te streng houden aan de benaming. In het auteursveld mag zeker ‘Anonymous’ staan.

Iets meer over het veld `author`

Elke auteur wordt gescheiden door het woordje `and`. Een auteur kan ingegeven worden als `voornaam achternaam` of als `achternaam, voornaam`. Dit heeft geen invloed op hoe de auteurs geciteerd worden in de bibliografie. `BIBTEX` wisselt die volgorde om, afhankelijk van de gebruikte citatiestijl. Bij de eerste mogelijkheid `voornaam achternaam` kan het gebeuren dat delen van samengestelde namen niet meegenomen worden in de achternaam. Bij de tweede manier weet `BIBTEX` met zekerheid wat de voornaam is en wat de achternaam (de twee worden namelijk gescheiden door een komma). Het voorbeeld van sectie 5.2.1 zou dus beter als volgt geschreven worden:

```
...
  author = {Athos, A. and Porthos, D. and Aramis, D.},
...
```

5.2.4 Templates

Het is niet leuk om telkens opnieuw al die types en eigenschappen in te typen. Vandaar dat we met *templates* kunnen werken. Dit zijn lege bibitems. Alle tekst in een `BIBTEX` databank die niet in een item zit, wordt beschouwd als commentaar (in een item zelf kan geen commentaar ingebracht worden⁴). Bijgevolg laten we de *templates* niet starten door een `@` maar bijvoorbeeld wel door een `A`. Wanneer we een nieuw item willen toevoegen aan onze databank, kopiëren en plakken we de lege *template*, vervangen de `A` door een `@` en vullen we de verschillende items aan. Om de optionele velden te onderscheiden van de verplichte, worden ze ingesprongen. Een optioneel argument mogen we leeg laten. Als voorbeeld voor een `article` hebben we de volgende *template*:

```
Aarticle{,
author = {},
title = {},
journal = {},
year = {},
    volume = {},
    number = {},
    pages = {},
    note = {}
}
```

⁴Hoewel. Aangezien `BIBTEX` alle eigenschappen negeert die het niet nodig acht voor een bepaalde item, kunnen wij de eigenschap `commentaar = Mijn opmerkingen` toevoegen, zonder dat deze verschijnt in de bibliografie.

Merk op dat de laatste eigenschap `note` geen komma op het einde heeft. Denk hieraan wanneer je dit weinig gebruikte veld uit je items verwijdt. De komma van het vorige veld (in casu `pages`) moet dan ook verdwijnen.

Voor mensen die echt niet graag in tekstbestanden rommelen en alles liever met knopjes besturen, bestaan er programma's om je `BIBTEX` databank te beheren. Een goed programma dat op alle besturingssystemen werkt, is JabRef: <http://jabref.sourceforge.net/>. Voor de Mac fanaten is er BibDesk: <http://bibdesk.sourceforge.net/>.

5.3 Citeren in de tekst

We hebben nu alles gezien om een bibliografie te maken, behalve het citeren in de tekst zelf. Hiervoor bestaan verschillende commando's. De twee basiscommando's zijn

- `\citet{athos:92}` (CITE in Text) om te citeren in doorlopende tekst. Voorbeeld: volgens ? moet je met gevaar leren leven.
- `\citep{athos:92}` (CITE in Parentheses) om het geciteerde werk tussen haakjes te zetten. Voorbeeld: Leve de koning (?).

Deze commando's hebben ook een stervorm om alle namen te laten zien: `\citet*{athos:92}` geeft ?. Meerdere items kunnen in één enkel commando samengebracht worden: `\citep{jones02,athos:92}` geeft (??).

Je kan ook commentaar kwijt in citaties, wat handig is wanneer de haakjes automatisch worden geplaatst:

<code>\citet[Hoofdstuk 13]{athos:92}</code>	→	?, Hoofdstuk 13
<code>\citep[sectie 3]{jones02}</code>	→	(?, sectie 3)
<code>\citep[in][sectie 3]{jones02}</code>	→	(in ?, sectie 3)
<code>\citep[in][]{jones02}</code>	→	(in ?)

Wanneer haakjes ongewenst zijn, kunnen we gebruik maken van de ALternatieve commando's (`\citealt`, met de `t` of `p` erachter, afhankelijk of we in de tekst citeren of tussen haakjes):

<code>\citealt{athos:92}</code>	→	?
<code>\citealt*{athos:92}</code>	→	?
<code>\citealp{athos:92}</code>	→	?
<code>\citealp*[in][sectie 3]{jones02}</code>	→	in ?, sectie 3

Soms wens je alleen de auteur of alleen het jaar. Dit laat toe om iets gevarieerder te citeren:

<code>\citeauthor{athos:92}</code>	→	?
<code>\citeauthor*{athos:92}</code>	→	?
<code>\citeyear{athos:92}</code>	→	?
<code>\citeyearpar{jones02}</code>	→	(?)
<code>\citeyearpar[in][sectie 3]{jones02}</code>	→	(in ?, sectie 3)

Om in het begin van een zin een auteur te citeren wiens familienaam begint met een kleine letter, kan je gebruik maken van de hoofdlettervariant van de hierboven vermelde commando's: `\citet{vanOostrum96}` geeft: ?, terwijl `\Citet{vanOostrum96}` ? geeft.

Artikels die nooit geciteerd worden in de tekst, komen normaalgezien ook niet voor in de bibliografie. Indien we toch ongciteerde artikels in de bibliografie wensen op te nemen, gebruiken we het commando `\nocite{key}` om het artikel met `key` op te nemen in de bibliografie. Met `\nocite{*}` worden alle artikels uit de bibliografische databank in de bibliografie opgenomen.

Hoofdstuk 6

Wiskundige formules

Wiskundige formules in een tekst inbrengen was voor het T_EX tijdperk tijdrovend en dus duur. Dit is één van de hoofdredenen waarom Donald Knuth T_EX ontwikkelde. Dit is ook de reden waarom wiskundige formules tussen dollartekens staan (niet altijd echter, zie verder).

De mathematische mogelijkheden van L^AT_EX worden sterk uitgebreid door gebruik te maken van het `amsmath` pakket (*American Mathematical Society*):

```
\usepackage{amsmath}      % Uitgebreide wiskundige mogelijkheden
```

In dit hoofdstuk zullen we ervan uitgaan dat dit standaard gebruikt wordt. Sommige voorbeelden zullen anders niet werken. Verder leggen we hier niet alles uit, verre van. Voor meer exotische dingen kan onder andere de help bij het `amsmath` pakket gebruikt worden.¹

6.1 Wiskundige omgeving

Wiskundige formules kunnen op twee manieren ingebracht worden. Enerzijds in de tekst, zoals $E = mc^2$ en anderzijds op een aparte lijn, zoals:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (6.1)$$

Wiskunde inbrengen in de tekst, gebeurt door de formule tussen dollartekens te zetten: `$E = mc^2$`.² Om de formules op een aparte lijn te krijgen:

```
\begin{equation}
  F(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}
  \label{gauss} % Om te kunnen refereren naar deze vergelijking
\end{equation}
```

¹Op een Debian systeem: `/usr/share/doc/texmf/latex/amsmath/amslldoc.dvi.gz`

²Ook `\begin{math} E = mc^2 \end{math}` en `\(E = mc^2\)` mogen gebruikt worden.

In plaats van `equation` mag ook `equation*`³ gebruikt worden. Deze laatste vorm vermijdt dat de vergelijkingen genummerd worden. In dit geval is het onzinnig om een label mee te geven. In een `equation` omgeving, wordt met het labelcommando het nummer van de vergelijking in een label gezet. Bijgevolg produceert `\ref{gauss}` het nummer van vergelijking 6.1.

6.1.1 Lange formules: `multline`

In de `equation` omgeving, worden lijnen nooit afgebroken. L^AT_EX kan en wil niet op eigen houtje beslissen op welke plaats een formule moet gesplitst worden. Er bestaat echter een speciale omgeving in dewelke je een formule over verschillende lijnen kan uitsmeren. Dit is de `multline` omgeving (en dus niet `multIline`!):

```
\begin{multline}
(x+y+z)^4 = z^4 + 4 y z^3 + 4 x z^3 + 6 y^2 z^2 + 12 x y z^2 + \\
6 x^2 z^2 + 4 y^3 z + 12 x y^2 z + 12 x^2 y z + 4 x^3 z + \\
y^4 + 4 x y^3 + 6 x^2 y^2 + 4 x^3 y + x^4
\end{multline}
```

$$(x + y + z)^4 = z^4 + 4yz^3 + 4xz^3 + 6y^2z^2 + 12xyz^2 + 6x^2z^2 + 4y^3z + 12xy^2z + 12x^2yz + 4x^3z + y^4 + 4xy^3 + 6x^2y^2 + 4x^3y + x^4 \quad (6.2)$$

Een nieuwe lijn beginnen gebeurt met `\\`. De eerste lijn wordt links uitgelijnd, de volgende lijnen worden gecentreerd en de laatste lijn wordt rechts uitgelijnd. Het vergelijkingnummer wordt rechts tegen de laatste lijn geplaatst. Indien geen vergelijkingnummer gewenst is, kan `multline*` gebruikt worden.

6.1.2 Meerdere formules tegelijkertijd: `gather`, `align`, `eqnarray`, `array`

Meerdere deelvergelijkingen in één vergelijking kan bekomen worden met de `gather` omgeving:

```
\begin{gather}
(a + b)^2 = a^2 + 2 a b + b^2 \qquad \qquad \qquad \backslash\label{gather1} \quad \\
(a + b)^3 = a^3 + 3 a^2 b + 3 a b^2 + b^3 \quad \backslash\text{nonumber} \quad \\
(a + b)(a - b) = a^2 - b^2 \qquad \qquad \qquad \backslash\label{gather2} \\
\end{gather}
```

$$(a + b)^2 = a^2 + 2ab + b^2 \quad (6.3)$$

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(a + b)(a - b) = a^2 - b^2 \quad (6.4)$$

³Of `\begin{displaymath}` formule `\end{displaymath}` of `$$` formule `$$` of `\[` formule `\]`.

De verschillende lijnen worden gecentreerd en elke lijn krijgt een vergelijkingsnummer. Er kan dus een label toegekend worden aan elke lijn. Wanneer het niet gewenst is om aan een bepaalde lijn een vergelijkingsnummer toe te kennen, kan het commando `\nonumber` gegeven worden. Dit is hetgeen gebeurde tussen vergelijking 6.3 en 6.4.

Het hierboven gegeven voorbeeld zou mooier zijn, moesten de verschillende gelijkheidstekens onder elkaar staan. Hiervoor is de `align` omgeving zeer geschikt. Het gebruik is analoog aan een `tabular` omgeving (bladzijde 40), alleen hoeft de uitlijningsformattering van de kolommen niet meegegeven te worden: de eerste kolom wordt rechts uitgelijnd, de tweede kolom links, de derde opnieuw rechts, de vierde weer links, enzovoort. Elke rij moet hetzelfde aantal kolommen bevatten. Het wordt gebruikt om verschillende kleine vergelijkingen naast elkaar te plaatsen.

```
\begin{align}
1+0 &= 1 & 1+1 &= 2 & 2+1 &= 3 & 3+1 &= 4 & \label{pascal2} \\
1+0 &= 1 & 1+2 &= 3 & 3+3 &= 6 & 6+4 &= 10 & \nonumber \\
1+0 &= 1 & 1+3 &= 4 & 4+6 &= 10 & 10+10 &= 20 & \label{pascal4}
\end{align}
```

$$1 + 0 = 1 \qquad 1 + 1 = 2 \qquad 2 + 1 = 3 \qquad 3 + 1 = 4 \qquad (6.5)$$

$$1 + 0 = 1 \qquad 1 + 2 = 3 \qquad 3 + 3 = 6 \qquad 6 + 4 = 10$$

$$1 + 0 = 1 \qquad 1 + 3 = 4 \qquad 4 + 6 = 10 \qquad 10 + 10 = 20 \qquad (6.6)$$

Door het commando `\nonumber` te geven op het einde van een lijn, wordt die lijn niet genummerd. Wanneer we geen enkele lijn wensen te nummeren, kunnen we `align*` gebruiken.

De `align` omgeving doet veel moeite om de verschillende kolommenparen mooi te spatieren. Met `alignat`, kunnen we die spatiëring teniet doen. Alle kolommen worden dan tegen elkaar geplakt. De syntax is `\begin{alignat}{N}` waarbij `N` het aantal kolomkoppels is. Met de `flalign` omgeving worden de kolomkoppels zo ver mogelijk uit elkaar gezet, zodat heel de breedte van de bladzijde gevuld is.

Om een bepaald symbool in een vergelijking telkens op dezelfde plaats te krijgen, kan de `eqnarray` omgeving gebruikt worden. In die omgeving bestaat elke lijn uit drie delen: het eerste deel wordt rechts uitgelijnd, het tweede deel wordt gecentreerd en het derde deel wordt links uitgelijnd. Elke lijn krijgt een vergelijkingsnummer, tenzij het commando `\nonumber` wordt gegeven.

```
\begin{eqnarray}
(a + b)^2 &= & a^2 + 2 a b + b^2 & \label{eqnarray1} \\
(a + b)^3 &= & a^3 + 3 a^2 b + 3 a b^2 + b^3 & \nonumber \\
(a + b)(a - b) &= & a^2 - b^2 & \label{eqnarray2} \\
&= & - b^2 + a^2 & \label{eqnarray3}
\end{eqnarray}
```

$$(a + b)^2 = a^2 + 2ab + b^2 \quad (6.7)$$

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(a + b)(a - b) = a^2 - b^2 \quad (6.8)$$

$$= -b^2 + a^2 \quad (6.9)$$

Merk op dat het, zoals in de `tabular` omgeving, ook hier mogelijk is om cellen leeg te laten, wat gebeurde in vergelijking 6.9.

6.1.3 Subnummering van vergelijkingen

Om vergelijkingen die bij elkaar horen hetzelfde hoofdnummer te geven, maar verschillende subnummers, kan gebruik gemaakt worden van de `subequations` omgeving. Deze omgeving is eigenlijk een superomgeving, die rond alle andere wiskundige omgevingen kan geplaatst worden. Alle vergelijkingen die binnen die superomgeving worden ingegeven, krijgen hetzelfde hoofdnummer, maar verschillen in het subnummer. Het `\label` commando juist na `\begin{subequations}` geeft een label met alleen de hoofdnummering. De subnummering in labels steken, gebeurt door het `\label` commando te geven in de verschillende vergelijkingen.

```
\begin{subequations}
  \label{subvgl}
  \begin{equation}
    (a + b)^2 = a^2 + 2 a b + b^2 \quad \label{subvgl1}
  \end{equation}
  Tussen de verschillende vergelijkingen mag tekst geplaatst worden.
  \begin{gather}
    (a + b)^3 = a^3 + 3 a^2 b + 3 a b^2 + b^3 \label{subvgl2} \quad \backslash
    (a + b)(a - b) = a^2 - b^2 \quad \label{subvgl3}
  \end{gather}
\end{subequations}
```

$$(a + b)^2 = a^2 + 2ab + b^2 \quad (6.10a)$$

Tussen de verschillende vergelijkingen mag tekst geplaatst worden.

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3 \quad (6.10b)$$

$$(a + b)(a - b) = a^2 - b^2 \quad (6.10c)$$

Het algemene nummer van de drie bovenstaande vergelijkingen is 6.10, verkregen met het commando `\ref{subvgl}`. De individuele vergelijkingen zijn 6.10a, 6.10b en 6.10c. Dit wordt verkregen met `\ref{subvgl1}`, `\ref{subvgl2}` en `\ref{subvgl3}`.

6.1.4 Omgevingen binnen de equation omgeving

Om ingewikkelde constructies te maken binnen wiskundige omgevingen, kunnen subomgevingen gebruikt worden. Er zijn er vijf: `aligned`, `gathered`, `split`, `array` en `cases`. Zij nummeren de vergelijking niet. Dat nummeren gebeurt door de buitenste wiskundige omgeving (in het voorbeeld hieronder `equation`).

De eerste twee hebben de volgende syntax:

```
\begin{aligned}[positie] vgl-lijnen \end{aligned}
\begin{gathered}[positie] vgl-lijnen \end{gathered}
```

Hierbij is `positie` een optioneel positioneringsargument dat `b` (*bottom*), `t` (*top*) of `c` (*center*, standaard) is. Het gebruik is analoog als bij `align` en `gather`:

```
\begin{equation}
\begin{aligned}[c]
a      &= b+c   \quad \quad a-b      &= c   \quad \quad a-b-c &= 0   \quad \quad \\
\end{aligned}
\quad \quad \quad \leftarrow \text{center---top} \rightarrow \quad \quad \quad \\
\begin{gathered}[t]
a      &= b+c   \quad \quad a-b      &= c   \quad \quad a-b-c &= 0   \quad \quad \\
\end{gathered}
\quad \quad \quad \leftarrow \text{top---bottom} \rightarrow \quad \quad \quad \\
\begin{aligned}[b]
a      &= b+c   \quad \quad a-b      &= c   \quad \quad a-b-c &= 0   \quad \quad \\
\end{aligned}
\end{equation}
```

$$\begin{array}{ccccccc}
 & & & & & & a = b + c \\
 & & & & & & a - b = c \\
 a = b + c & & \leftarrow \text{center---top} \rightarrow & & a = b + c & & \leftarrow \text{top---bottom} \rightarrow & & a - b - c = 0 \\
 a - b - c = 0 & & & & a - b = c & & & & \\
 & & & & a - b - c = 0 & & & &
 \end{array} \quad (6.11)$$

De `split` omgeving moet ook binnen een wiskundige omgeving voorkomen en wordt gebruikt om vergelijkingen te splitsen. Het verschil met de `multline` omgeving is dat er in elke lijn een uitlijningsteken voorkomt & zodat de schrijver meer controle heeft over hoe de vergelijkingen eruit zien.

```
\begin{equation}
\begin{split}
(x+y+z)^4 = & z^4 + 4 y z^3 + 4 x z^3 + 6 y^2 z^2 + 12 x y z^2 + \quad \quad \quad \\
& 6 x^2 z^2 + 4 y^3 z + 12 x y^2 z + 12 x^2 y z + 4 x^3 z + \quad \quad \quad \\
& y^4 + 4 x y^3 + 6 x^2 y^2 + 4 x^3 y + x^4
\end{split}
\end{equation}
```

$$\begin{aligned}
 (x + y + z)^4 = & z^4 + 4yz^3 + 4xz^3 + 6y^2z^2 + 12xyz^2 + \\
 & 6x^2z^2 + 4y^3z + 12xy^2z + 12x^2yz + 4x^3z + \\
 & y^4 + 4xy^3 + 6x^2y^2 + 4x^3y + x^4
 \end{aligned} \quad (6.12)$$

Merk op dat analoge resultaten kunnen verkregen worden met `align`, `aligned` en `array` omgeving.

Wanneer slechts één nummer gewenst is voor heel de set vergelijkingen, kan gebruik gemaakt worden van de meer algemene `array` omgeving. Het gebruik van die omgeving is zoals bij de `tabular` omgeving. De `array` omgeving mag enkel geplaatst worden in wiskundige modus. Het is eigenlijk een tabelomgeving voor wiskunde:

```

\begin{equation}
\label{array}
\begin{array}{rcl}
(a + b)^2 & = & a^2 + 2 a b + b^2 \\
(a + b)^3 & = & a^3 + 3 a^2 b + 3 a b^2 + b^3
\end{array}
\end{equation}

```

$$\begin{array}{rcl}
 (a + b)^2 & = & a^2 + 2ab + b^2 \\
 (a + b)^3 & = & a^3 + 3a^2b + 3ab^2 + b^3
 \end{array}
 \tag{6.13}$$

Een structuur zoals

$$P = \begin{cases} A : & x \geq 12 \\ B : & 10 \leq x < 12 \\ C : & x < 10 \end{cases}
 \tag{6.14}$$

komt dikwijls voor in de wiskunde. Dit kan gemakkelijk aangemaakt worden met de `cases` omgeving. Het voorbeeld hierboven werd als volgt verkregen:

```

\begin{equation}
P = \begin{cases}
A : & x \geq 12 \\
B : & 10 \leq x < 12 \\
C : & x < 10
\end{cases}
\end{equation}

```

Er mag slechts één ampersand voorkomen in elke rij.

6.2 De wiskundetaal

6.2.1 Exponenten en indices — superscript en subscript

Om iets te verheffen (superscript), wordt het `^` symbool gebruikt. Iets beneden krijgen (subscript) gebeurt met `_` symbool. Deze commando's hebben slechts betrekking op de daaropvolgende letter. Wil je meer in de exponent of index, gebruik dan accolades. Beiden mogen gecombineerd worden om tegelijkertijd een exponent en een index te hebben. Ze kunnen ook genest worden.

$$A^2b + A_{2b} + A_2^b + A^{B^C} \longrightarrow A^2b + A_{\{2b\}} + A_2^b + A^{\{B^C\}}$$

6.2.2 Breuken

Breuken worden gemaakt met het commando

`\frac{teller}{noemer}`

Breuken kunnen genest worden. Hierbij wordt de geneste breuk kleiner afgeprint:

$$\frac{\frac{x+y}{x-y} + 1}{x+y}$$

```
\begin{equation*}
\frac{\frac{\frac{x+y}{x-y} + 1}{x+y}}{x+y}
\end{equation*}
```

Om dit te vermijden, kan `\dfrac` gebruikt worden:

$$\frac{x+y}{x-y} + 1$$

```
\begin{equation*}
\frac{\dfrac{x+y}{x-y} + 1}{x+y}
\end{equation*}
```

Ook in andere situaties waarin met `\frac` de breuk kleiner wordt afgebeeld dan normaal, kan `\dfrac` gebruikt worden.

6.2.3 Binomiaalcoëfficiënten

Binomiaalcoëfficiënten kunnen eigenlijk gezien worden als breuken zonder breukstrepen en met haken rond. Zij worden verkregen met het commando `\binom{boven}{onder}`:

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} \quad \longrightarrow \quad \texttt{\backslash binom{n+1}{k}} = \texttt{\backslash binom{n}{k}} + \texttt{\backslash binom{n}{k-1}}$$

6.2.4 Wortels

Wortels worden met het commando `\sqrt[n]{onder-wortel}` gegenereerd. Hierbij is `n` gelijk aan de gewenste machtswortel. Voor de tweede machtswortel kunnen we dit optionele argument weglaten.

$$\sqrt{A+B} + \sqrt[3]{\frac{A-B}{A+B}} \quad \longrightarrow \quad \texttt{\backslash sqrt{A+B}} + \texttt{\backslash sqrt[3]{\frac{A-B}{A+B}}}$$

6.2.5 Sommen en integralen

Sommen worden geproduceerd met het commando `\sum`, integralen met `\int`

$$\sum a_i \sum_{i=1}^n b_i \quad \longrightarrow \quad \texttt{\backslash sum a_i \, ; \, \backslash sum_{i=1}^n b_i}$$

$$\int a \, dx \int_m^n x \, dx \quad \longrightarrow \quad \texttt{\backslash int a \, , \, \mathrm{d}x \, ; \, \backslash int_m^n x \, , \, \mathrm{d}x}$$

In *math mode* wordt geen rekening gehouden met de spaties in het *latex*-bestand. Alles wordt aan elkaar geplakt. Wanneer we ergens witte ruimte willen zetten, moeten we dit expliciet aangeven. Het commando `\,` zorgt voor een kleine spatie; `\;` geeft wat meer ruimte. Verder zien we het gebruik van `\mathrm` (van *math roman*) om een recht lettertype te selecteren. Het symbool voor een afgeleide is namelijk geen variabele en mag dus niet cursief staan.

Sommigen hebben liever de limieten van de integraal boven en onder de integraal, in plaats van ernaast. Met `\limits` kan dit verkregen worden:

$$\int_m^n dx \quad \longrightarrow \quad \int\limits_m^n \mathrm{d}x$$

Meerdere integralen na elkaar produceer je met `\iint`, `\iiint`, `\iiiiint` en `\idotsint`: \iint , \iiint , \iiiiint en $\int \cdots \int$. Met `\oint` wordt een kringintegraal \oint .

6.2.6 Ellipsis

Doordat \LaTeX de spaties tussen punten negeert, kunnen we niet gewoon `...` (resultaat: ...) gebruiken. In *math mode* bestaan er verschillende soorten ellipsis:

<code>\ldots</code>	...	<i>Low dots</i>	<code>\cdots</code>	...	<i>Center dots</i>
<code>\vdots</code>	⋮	<i>Vertical dots</i>	<code>\ddots</code>	⋱	<i>Diagonal dots</i>

Het verschil tussen `\ldots` (*lower*) en `\cdots` (*center*) is dat het ene gebruikt wordt bij het opsommen van lijsten, zoals in x_1, \dots, x_n (gemaakt met `x_1, \ldots, x_n`) en het andere bij het sommeren van lijsten, zoals in $x_1 + \cdots + x_n$ (gemaakt met `x_1 + \cdots + x_n`).

6.2.7 Operatornamen

In de wiskunde is het gebruikelijk om de namen van variabelen cursief te zetten. Constanten en namen van operatoren moeten echter recht staan. In *math mode* wordt alles cursief gezet (behalve getallen). Ook wordt er geen plaats gelaten tussen de letters onderling. Bijgevolg geeft `\sin x` *sinx* terwijl we graag $\sin x$ zouden zien. Om dit te bereiken moet de operatornaam vooraf gegaan worden door een *backslash*: `\sin x`. Dit lukt alleen maar met de meest gebruikelijke operatoren (zoals `\cos`, `\tan`, `\lim` ...); `\mynoperator x` zal een foutmelding geven. Om van `\mynoperator` een operator te maken die door \LaTeX herkend wordt, moet het volgende commando gegeven worden in de *preamble*:

```
\DeclareMathOperator{\mynoperator}{mynoperator}
```

De naam (`\mynoperator`) van de operator moet niet dezelfde zijn als hetgeen verschijnt in het finale document:

`\DeclareMathOperator{\integ}{Integraal}`

zorgt ervoor dat telkens we `\integ x` schrijven (in *math mode* natuurlijk!) er ‘Integraal x ’ verschijnt.

Sommige operatoren plaatsen het subscript of superscript onderaan of bovenaan. Om dit tegen te gaan, kan juist na de operator het commando `\nolimits` gegeven worden:

$\lim_{x \rightarrow \infty} \lim_{x \rightarrow \infty} \longrightarrow \code{\lim_{x \rightarrow \infty}} \code{\lim \nolimits_{x \rightarrow \infty}}$

6.2.8 Accenten

De volgende accenten zijn beschikbaar in *math mode*:

\hat{x} <code>\hat{x}</code>	\breve{x} <code>\breve{x}</code>	\grave{x} <code>\grave{x}</code>	\bar{x} <code>\bar{x}</code>
\check{x} <code>\check{x}</code>	\acute{x} <code>\acute{x}</code>	\tilde{x} <code>\tilde{x}</code>	\vec{x} <code>\vec{x}</code>
\dot{x} <code>\dot{x}</code>	\ddot{x} <code>\ddot{x}</code>	\mathring{x} <code>\mathring{x}</code>	

De letters i en j moeten gebruikt worden zonder punt wanneer er een accent opgezet wordt. Gebruik hiertoe `\imath` en `jmath`. Dit geeft \hat{i} voor `\hat{\imath}` en \bar{j} voor `\bar{jmath}`.

Er bestaan versies van `\hat` en `\tilde` die over meer dan één karakter kunnen gespreid worden: `\widehat` en `\widetilde`:

$\widehat{A_3^{\frac{\widetilde{b+c}}{m}}}$ \longrightarrow `\widehat{A_3^{\frac{\widetilde{b+c}}{m}}}`

Met `\overline{uitdrukking}` overstrepen we *uitdrukking*; `\underline{uitdrukking}` kan gebruikt worden om te onderstrepen. Analoot bestaat er ook `\overbrace` en `\underbrace`. Deze laatste twee zorgen voor accolades. Door er een superscript, respectievelijk een subscript aan toe te voegen, kunnen we iets schrijven aan die accolades.

```
\begin{equation*}
\underbrace{\ldots, -3, -2, -1, \overbrace{0, 1, 2, 3, \ldots}^{\mathbb{N}}}_{\mathbb{Z}}
\end{equation*}
```

$$\underbrace{\ldots, -3, -2, -1, \overbrace{0, 1, 2, 3, \ldots}^{\mathbb{N}}}_{\mathbb{Z}}$$

Het op elkaar stapelen van karakters kan gebeuren met `\stackrel{boven}{onder}`. Dit commando centreert het bovenste symbool op het onderste: `\stackrel{<}{=}` geeft \leq .

6.2.9 Symbolen

Er bestaan een heleboel symbolen om wiskundige begrippen voor te stellen. In bijlage A zijn enkele tabellen terug te vinden met de meest gangbare symbolen.

De Griekse letters worden verkregen door de naam van de letter te laten voorafgaan door een *backslash*: `\alpha`, `\beta` geeft α , β . De hoofdletters bekomt men door met een hoofdletter te beginnen: `\Gamma` geeft Γ . Als de hoofdletter dezelfde is in het Latijns alfabet, bestaat er geen Griekse variant.

Om een schuine streep te trekken doorheen een symbool, laat je het voorafgaan door `\not`. Bijgevolg geeft `\not\subset` $\not\subset$, en resulteert `\notin` in \notin .

Een symbool dat verschrikkelijk lastig kan zijn om terug te vinden, is het partieel afgeleide symbool. Het is nochtans niet moeilijk: `\partial` geeft ∂ , maar je moet het weten.

6.2.10 Gewone tekst in *math mode*

Met het commando `\text{tekst}` kunnen we gewone tekst invoegen tussen de vergelijkingen:

$$A + A + A \quad \text{is ook} \quad 3A \quad \longrightarrow \quad A+A+A \quad \text{\texttt{\text{is ook}}}$$

In de omgevingen waarbij vergelijkingen over verschillende lijnen kunnen uitgesmeerd worden, is het soms interessant om een lijn tekst tussen te voegen zonder dat de aliniëring verandert. Dit kan bereikt worden met het commando `\intertext{tekst}` dat alleen mag voorkomen juist na een `\`.

```
\begin{align}
E &= m \times a^2 \quad \text{\texttt{Nee, slecht}} && \text{\texttt{\label{einstein1}}} \\
E &= m \times b^2 \quad \text{\texttt{Bah, ziet er nog niet goed uit}} && \text{\texttt{\label{einstein2}}} \\
&\text{\texttt{\intertext{Eureka!}}} \\
E &= m \times c^2 && \text{\texttt{\label{einstein3}}}
\end{align}
```

$$E = m \times a^2 \quad \text{Nee, slecht} \tag{6.15}$$

$$E = m \times b^2 \quad \text{Bah, ziet er nog niet goed uit} \tag{6.16}$$

Eureka!

$$E = m \times c^2 \tag{6.17}$$

Merk op hoe vergelijking 6.17 niet gecentreerd staat, opdat het gelijkheidsteken juist onder dat van vergelijking 6.16 en 6.15 zou komen.

In tabel A.8 staat beschreven hoe je pijlen kunt gebruiken als haakjes. Die kun je met de hierboven beschreven commando's echter niet vergroten. Om die te vergroten, moet je ze laten voorafgaan door `\big`, `\Big`, `\bigg` of `\Bigg`, waarbij de betekenis analoog is aan de hiervoor beschreven haakjes.

$$\lceil \Downarrow = \big\Downarrow \Big\rangle = \Big\rangle = \bigg\} \Uparrow = \Big\Uparrow \quad (6.20)$$

6.2.12 Matrices

Matrices zouden eventueel kunnen ingevoerd worden door een `split` omgeving tussen haakjes te zetten, maar dit is nogal omslachtig. Er bestaan specifieke omgevingen om matrices in te voeren: `matrix` (zonder haakjes, kan gebruikt worden als vervanging van de `array` omgeving), `pmatrix` (geeft ronde haken: (m)), `bmatrix` (vierkante haken: $[m]$), `Bmatrix` (geeft accolades: $\{m\}$), `vmatrix` (geeft verticale strepen: $|m|$) en `Vmatrix` (geeft dubbele verticale strepen: $||m||$).

Deze matrixomgevingen kunnen enkel binnen een andere wiskundige omgeving gebruikt worden.

Er moet niet worden ingegeven hoeveel kolommen de matrix telt. Het aantal kolommen moet wel in elke rij gelijk zijn. Elementen in een rij worden gesplitst door een ampersand (&); verschillende rijen door een dubbele *backslash*.

```
\begin{equation}
  A =
  \begin{bmatrix}
    a & b \\
    c & d
  \end{bmatrix}
\end{equation}
```

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (6.21)$$

Het aantal kolommen is beperkt tot tien. Indien we matrices willen invoeren die meer dan tien kolommen bevatten, moeten we het volgende commando in de *preamble* plaatsen:

```
\setcounter{MaxMatrixCols}{20} % Max 20 kolommen in een matrix
```

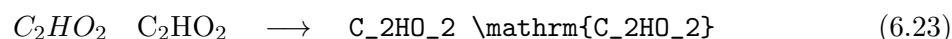
Dit zorgt ervoor dat we tot twintig kolommen in een matrix kunnen steken.

6.2.13 Witte ruimte

Doordat in *math mode* alle spaties worden genegeerd, is het daar zeker belangrijk om harde spaties in te voeren. (Ook regelovergangen worden genegeerd. Je mag dus een formule over

6.3.2 Rechte letters

Om het cursief zijn van de letters in *math mode* tegen te gaan, kunnen ze in het `\mathrm{recht}` commando gezet worden (van *Math RoMan*).



Een ander soort rechte letter in *math mode* is *sans serif*, te verkrijgen met `\mathsf{formule}`. Dit lettertype wordt gebruikt voor matrices (zie sectie 6.5).

6.3.3 Andere letters

Door gebruik te maken van het pakket `amssymb`, kunnen de zogenaamde *blackboard*, *Gothic* of *Fraktur* en *script* letters gebruikt worden.

```
\usepackage{amssymb}
```

Hierna kunnen de volgende commando's gebruikt worden. Met `\mathbb{A B C ...}` wordt het volgende verkregen:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Met `\mathcal{A B C ...}` wordt het volgende verkregen:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Met `\mathfrak{A B C ... \ a b c ...}` wordt het volgende verkregen:

$\mathfrak{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$
 $\mathfrak{a b c d e f g h i j k l m n o p q r s t u v w x y z}$

6.4 Commando's in wiskundige modus

Wanneer je commando's definieert in de *preamble*, weet je niet zeker of ze in *math mode* zullen gebruikt worden. Je zou dollars kunnen plaatsen, maar dit geeft dan weer problemen wanneer het commando wel in *math mode* wordt gebruikt. Vandaar het commando `\ensuremath{wiskunde}` dat ervoor zorgt dat het argument altijd in *math mode* staat.

```
\newcommand{\diff}{\ensuremath{\mathrm{d}}}
```

In dit voorbeeld is de `\ensuremath` een beetje overbodig. De differentiaal operator ga je altijd gebruiken in *math mode*. Het volgende voorbeeld maakt een eigen commando aan voor subscript en superscript. In gewone tekst is dit namelijk moeilijk te bereiken.

```

\newcommand{\supsc}[1]{\ensuremath{\text{\textsuperscript{#1}}}} % Superscript in tekst
\newcommand{\subsc}[1]{\ensuremath{\text{\textsubscript{#1}}}} % Subscript in tekst

```

Met `\supsc{tekst}` krijgen we dit en met `\subsc{tekst}` krijgen we dat.

6.5 Conventies voor matrices en vectoren

Het is algemeen aangenomen in de wiskunde om vectoren vet en cursief te zetten (ja, het `\vec` commando zet een pijl bovenop de vector, wat dus fout is) en matrices in *sans serif* letters. Het is handig om hiervoor enkele nieuwe commando's te definiëren.

```
\newcommand{\vt}[1]{\ensuremath{\boldsymbol{#1}}} % vector in juiste lettertype
\newcommand{\mx}[1]{\ensuremath{\mathsf{#1}}}      % matrix in juiste lettertype
```

Met `\vt{a}` krijgen we ***a*** en met `\mx{A}` krijgen we **A**. Deze commando's kunnen wegens `\ensuremath` ook in gewone tekst gebruikt worden.

6.6 Chemische formules

Chemische formules worden conventioneel in een recht lettertype gezet. Met `\mathrm` zouden we al die subscripten kunnen zetten. Er bestaat echter een pakket dat ons het leven veel gemakkelijker maakt, namelijk `mhchem`, in te laden met⁴:

```
\usepackage[version=3]{mhchem} % Voor elegante scheikundige formules
```

Nu kunnen we zeer eenvoudig in onze tekst scheikundige formules zetten door ze als argument van het commando `\ce{ChemischeFormule}` mee te geven:

<code>\ce{H2SO4}</code>	\longrightarrow	H_2SO_4
<code>\ce{1/2H2O}</code>	\longrightarrow	$\frac{1}{2}\text{H}_2\text{O}$
<code>\ce{^{227}_{90}Th+}</code>	\longrightarrow	$^{227}_{90}\text{Th}^+$
<code>\ce{H2O <=> H+ + OH-}</code>	\longrightarrow	$\text{H}_2\text{O} \rightleftharpoons \text{H}^+ + \text{OH}^-$
<code>\ce{H2O <<=> H+ + OH-}</code>	\longrightarrow	$\text{H}_2\text{O} \rightleftharpoons \text{H}^+ + \text{OH}^-$

Dit mag ook in *math mode*, zoals te zien in vergelijking 6.24. Op die manier kunnen we chemische formules nummeren en kunnen we ernaar verwijzen in de tekst:

```
\begin{equation}
\ce{CO2 + 6H2O -> C6H12O6 + 6O2}
\label{mathchem}
\end{equation}
```

$$\text{CO}_2 + 6\text{H}_2\text{O} \longrightarrow \text{C}_6\text{H}_{12}\text{O}_6 + 6\text{O}_2 \quad (6.24)$$

In de handleiding van `mhchem` staat ter voorbeeld hoe een nieuwe scheikundige *equation* omgeving kan gemaakt worden.

⁴Die `[version=3]` moet erbij omdat dan de nieuwste snuffjes van het pakket gebruikt worden.

Hoofdstuk 7

Conclusies en perspectieven

Elk goed werk eindigt met algemene slotconclusies (pleonasme) en een blik op wat beter kan (maar niet elk werk dat daarmee eindigt is een goed werk). Nu, bij deze cursus zijn er niet veel conclusies. Perspectieven zijn er des te meer. We hebben het vooral gehad over de syntax van \LaTeX . Niet over de programma's er rond. Mensen die met \LaTeX blijven werken, worden door de jaren heen zeer kritisch over de kwaliteit van hun documenten. Maar niet alleen hun documenten, heel hun (computer) tot op het pathologische af.¹ Bijgevolg zijn de programma's er rond zeer belangrijk.

- **make** Bij de finalisatie van een document, moeten verschillende commando's opgeroepen worden: 1 keer `latex`, 1 keer `bibtex`, 3 keer `latex`, 1 keer `makeindex`, 2 keer `latex`. Met een *Makefile* kan je dat allemaal inkorten tot één enkel commando: `make full`.
- **ispell** Spellingscontrole.
- **xfig** Voor maximale homogeniteit tussen de figuren en de tekst. De letters op je figuur zijn dezelfde als die in je document.
- **vim** De teksteditor. In het begin moeilijk te gebruiken, maar één keer je er mee weg bent, lijken alle andere tekstverwerkers zo kleurloos.
- **gimp** *GNU Image Manipulation Program*. Om figuren op te poetsen.
- **gnuplot**, **gnuplot** Voor het omzetten van data in grafieken.

¹Waarom denk je, werd deze cursus geschreven op een Debian GNU/Linux systeem? Omdat dat gratis is? Nee! Omdat het beter is? Speelt vast en zeker mee. Maar de belangrijkste reden is de VRIJHEID van dat systeem!

Bijlage A

Wiskundige symbolen

Hieronder geven we tabellen met verschillende wiskundige symbolen. Aangepast overgenomen uit de documentatie van `amsmath`.¹

Tabel A.1: Kleine Griekse letters

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

Tabel A.2: Griekse hoofdletters, als ze verschillend zijn van de Latijnse. Om ze cursief te krijgen, moet er een `var` voorgeplaatst worden. Bijvoorbeeld ‘`\varPi`’ geeft Π in plaats van Π .

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

¹Namelijk: `/usr/share/doc/texmf/latex/general/symbols.dvi.gz`

Tabel A.3: Andere symbolen

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

Tabel A.4: Verzamelingsoperatoren

\sum	<code>\sum</code>	\bigcap	<code>\bigcap</code>	\bigodot	<code>\bigodot</code>
\prod	<code>\prod</code>	\bigcup	<code>\bigcup</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>	\bigoplus	<code>\bigoplus</code>
\int	<code>\int</code>	\bigvee	<code>\bigvee</code>	\biguplus	<code>\biguplus</code>
\oint	<code>\oint</code>	\bigwedge	<code>\bigwedge</code>		

Tabel A.5: Binaire operatoren

$+$	<code>+</code>	$-$	<code>-</code>		
\pm	<code>\pm</code>	\cap	<code>\cap</code>	\vee	<code>\vee</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\wedge	<code>\wedge</code>
\setminus	<code>\setminus</code>	\uplus	<code>\uplus</code>	\oplus	<code>\oplus</code>
\cdot	<code>\cdot</code>	\sqcap	<code>\sqcap</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\sqcup	<code>\sqcup</code>	\otimes	<code>\otimes</code>
$*$	<code>\ast</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
\star	<code>\star</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\diamond	<code>\diamond</code>	\wr	<code>\wr</code>	\dagger	<code>\dagger</code>
\circ	<code>\circ</code>	\bigcirc	<code>\bigcirc</code>	\ddagger	<code>\ddagger</code>
\bullet	<code>\bullet</code>	\triangleup	<code>\triangleup</code>	\amalg	<code>\amalg</code>
\div	<code>\div</code>	\triangledown	<code>\triangledown</code>		

Tabel A.6: Relationele operatoren. Merk op dat de negatie van een symbool altijd kan verkregen worden door er `\not` voor te zetten: `\not\in` geeft \notin .

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\bowtie	<code>\bowtie</code>				
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\smile	<code>\smile</code>	\mid	<code>\mid</code>	\doteq	<code>\doteq</code>
\frown	<code>\frown</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
$\not<$	<code>\not<</code>	$\not>$	<code>\not></code>	\neq	<code>\neq</code>
...		

Tabel A.7: Pijlen

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>				

Tabel A.8: Haakjes. Rechterhaakjes kunnen bekomen worden door de eerste 1 te vervangen door een r.

$($	<code>(</code>	$[$	<code>[</code>	$\{$	<code>\{</code>	\lceil	<code>\lceil</code>	\lvert	<code>\lvert</code>
\langle	<code>\langle</code>	\lbrack	<code>\lbrack</code>	$\{$	<code>\{</code>	\lfloor	<code>\lfloor</code>	\lvert	<code>\lvert</code>