



LHR01

LaTeX-stijl HR



Handleiding

Versie 2.5

J.Z.M. Broeders

Versiehistorie

Datum	Versie	Omschrijving	Auteur
01-01-2023	2.5 ¹	Pinky assembler toegevoegd in hoofdstuk 4 .	BroJZ
31-12-2020	2.4	Belangrijkste aanpassingen: <ul style="list-style-type: none"> • Tabel commando's werken nu zonder gebruik van package <code> tabu </code>. • Nieuwe commando's <code>\tabelX</code>, <code>\tabelXm</code>, <code>\langetabelX</code> en <code>\langetabelXm</code>. • Enkele tips toegevoegd. 	BroJZ
28-12-2018	2.3	Belangrijkste aanpassingen: <ul style="list-style-type: none"> • Lege ruimte rond floats en regelafstand aangepast. • Kleur red vervangen door <code>hrred</code>. • <code>\hrfigure</code>-commando als alternatief voor <code>\figuur</code>-commando. • Commando's <code>\tikzfiguur</code> en <code>\hrtikzfiguur</code>. • Commando's <code>\tikzinlinefiguur</code> en <code>\hrtikzinlinefigure</code>. • Commando's <code>\schakeling</code> en <code>\hrschakeling</code>. • Gekleurde opmaak van \TeX-code gebaseerd op TeXstudio. 	BroJZ
23-02-2016	2.2	\TeX -code verplaatst naar public bitbucket repository en alle links hierop aangepast. <code>\hcode</code> -commando verbeterd.	BroJZ
22-07-2015	2.1	Voorbeelden toegevoegd. <code>\hcode</code> -commando verbeterd. Commando's <code>\pdfdeelref</code> en <code>\citepdfdeel</code> toegevoegd. Diverse commando's robuuster gemaakt.	BroJZ
09-07-2015	2.0	Eerste versie voor HR.	BroJZ
09-04-2015	1.0	Laatste versie voor HHS.	BroJZ

In principe is het onzinnig om versiehistorie in het document zelf bij te houden als de broncode beschikbaar is in een publiek toegankelijke VCS (version control system). Maar omdat dit niet voor alle documenten het geval hoeft te zijn, wordt hier toch, als voorbeeld, een versiehistorietabel gebruikt.



Handleiding LaTeX-stijl HR van Hogeschool Rotterdam is in licentie gegeven volgens een [Creative Commons Naamsvermelding-NietCommercieel-GelijkDelen 3.0 Nederland-licentie](#). [Figuren 2.1](#), [B.1](#) en [C.1](#) en ?? in deze handleiding zijn afkomstig van [Pixabay](#) en vallen onder het publieke domein volgens een [Publiek Domein Verklaring \(CC0 1.0\)](#).

De \LaTeX -code van deze handleiding kun je vinden op: https://bitbucket.org/HR_ELEKTRO/latex-stijl.

¹ Toelichting versiecodering *A.Bc*: *A* = grote aanpassing, *B* = kleine aanpassing, *c* = taal- of wiskundige correcties.

Inhoudsopgave

1	Inleiding	6
2	Gebruik \LaTeX-stijl Hogeschool Rotterdam	8
2.1	Structuur	9
2.2	Cursushandleiding	10
2.3	Tentamen	11
2.4	Titelpagina	11
2.5	Versiehistorie	11
2.6	Licentie	13
2.7	Karakter encoding	14
2.8	Afbreken van woorden	14
2.9	Afbreken van pagina's	14
2.10	Aanhalingstekens	15
2.11	Figuren	16
2.11.1	Figuren van internet	18
2.11.2	Figuren zelf tekenen met een extern programma	18
2.11.3	Grafieken aanmaken met MATLAB	19
2.11.4	Screenshots	19
2.11.5	Tekenen in \LaTeX met <i>TikZ</i>	20
2.11.6	Elektronische schema's tekenen in \LaTeX met <i>circuiTikZ</i>	24
2.11.7	Gafieken tekenen in \LaTeX met PGFPLOTS	26
2.12	Tabellen	29
2.13	Verwijzingen en citaten	32
2.14	Opdrachten	37
3	English Style	38

3.1	Titles of Chapters and Sections	38
3.2	License	38
3.3	References	39
3.4	Quotes	39
4	Listings en code-fragmenten	40
4.1	Listings	40
4.2	\LaTeX -code in listings	41
4.3	Code-segmenten	42
4.4	Console in- en output	43
4.5	Overige commando's die handig zijn bij documenten met programmeercode .	43
5	Algemene \LaTeX-tips en -trucs	45
5.1	Titel ²	47
5.2	$E = mc^2$	47
	Bibliografie	49
A	Voorbeelden van codefragmenten	50
B	Voorbeelden van floats	52
C	Voorbeelden van Here floats	54

² Bedenk zelf een goede titel

1

Inleiding

Deze handleiding laat zien hoe je de \LaTeX -stijl die is ontwikkeld voor documenten van de opleiding Elektrotechniek van de Hogeschool Rotterdam kunt gebruiken. Deze stijl is gebaseerd op een stijl die eerder door de auteur ontwikkeld is bij de opleiding Elektrotechniek aan De Haagse Hogeschool en op een Word-template van de Hogeschool Rotterdam die ontwikkeld is door Johan Peltenburg.

Met deze template kan vanuit één \LaTeX -bestand zowel een papieren als een ebook versie gegenereerd worden. De papieren versie is bedoeld om dubbelzijdig te printen, heeft een bindmarge, hoofdstukken beginnen op oneven pagina's en links hebben geen kleur (klikken op papier werkt namelijk niet, de links werken overigens wel als het document toch op een elektronisch device wordt gelezen). De ebook versie is enkelzijdig en heeft gekleurde links om de lezer er op te attenderen dat je er op kunt klikken. In de rest van deze handleiding wordt uitgelegd hoe je deze \LaTeX -stijl kunt gebruiken en worden voorbeelden gegeven om te laten zien hoe de layout van bepaalde dingen is.

Er is ook een versie van deze template beschikbaar voor het maken van Engelstalige documenten. Specifieke aandachtspunten voor het gebruik van die template worden in een apart hoofdstuk besproken. Tot slot volgt nog een hoofdstuk met enkele algemene tips voor het gebruik van \LaTeX .

Deze stijl is getest met de \TeX -distributie Tex Live³ en de verwachting is dat deze stijl met andere distributies net zo goed werkt, maar dat is niet getest.

³ <http://www.tug.org/texlive/>

Zelf maak ik gebruik van de \LaTeX -editor TeXstudio⁴. Er is een zogenoemd session- bestand beschikbaar waarmee in één keer alle relevante bestanden geopend worden: `main.txss`. Maar je kunt deze template ook prima met elke andere editor gebruiken.

Voorbeelden van documenten (book style) die met deze template gemaakt zijn:

- *Objectgeoriënteerd Programmeren in C++* [2] (Nederlands).⁵
- *Training Digital Signal Processing Work Lab Handbook* [6] (Engels).⁶
- Dit document zelf.⁷

Voorbeelden van documenten (article style) die met deze template gemaakt zijn:

- Modulewijzer TDS02 (Nederlands).⁸
- Report Requirements for TDS02 (Engels).⁹

Op- en aanmerkingen zijn altijd welkom. Je kunt me mailen op <mailto:J.Z.M.Broeders@hr.nl>.

⁴ <https://www.texstudio.org/>

⁵ Zie: [Dictaat_OPinCpp.pdf](#) en [Dictaat_OPinCpp_ebook.pdf](#). De sourcecode vind je hier: https://bitbucket.org/HR_ELEKTRO/cppprog/src.

⁶ Zie: [Lab_Work_Handbook_TDS02.pdf](#) en [Lab_Work_Handbook_TDS02_ebook.pdf](#). De sourcecode vind je hier: https://bitbucket.org/HR_ELEKTRO/tds02/.

⁷ Sourcecode: https://bitbucket.org/HR_ELEKTRO/latex-stijl/.

⁸ Zie: [Modulewijzer_TDS02.pdf](#) en [Modulewijzer_TDS02_ebook.pdf](#). De sourcecode vind je hier: https://bitbucket.org/HR_ELEKTRO/tds02/.

⁹ Zie: [Report_Requirements_TDS02.pdf](#) en [Report_Requirements_TDS02_ebook.pdf](#). De sourcecode vind je hier: https://bitbucket.org/HR_ELEKTRO/tds02/.

2

Gebruik \LaTeX -stijl Hogeschool Rotterdam

In dit hoofdstuk wordt het gebruik van de \LaTeX -stijl `style_nl_NL` toegelicht. Ik ga er daarbij vanuit dat je al goed bekend bent met $\LaTeX 2_{\epsilon}$. Als je een opfrisser nodig hebt dan kan ik de, online beschikbare, introductie *The Not So Short Introduction to $\LaTeX 2_{\epsilon}$* [4] van harte aanbevelen. Als je haast hebt, kun je volstaan met het lezen van paragrafen 1.3, 1.8, 2.1, 2.6, 2.7, 2.9, 2.10, 2.11.1 en 2.11.3.

Let er wel op dat sommige paragrafen in [4] niet relevant zijn als je de in deze handleiding beschreven stijl gebruikt. Bijvoorbeeld: [4, paragraaf 2.4.1] kan worden vervangen door paragraaf 2.10, [4, paragraaf 2.4.9] is overbodig vanwege de in deze stijl gebruikte karakter encoding, zie paragraaf 2.7, [4, paragraaf 2.8] wordt ‘overruled’ door paragraaf 2.13, [4, paragraaf 2.11.5] wordt vervangen door hoofdstuk 4, [4, paragraaf 2.11.6] wordt vervangen door paragraaf 2.12 en [4, paragraaf 2.12] wordt vervangen door paragrafen 2.11 en 2.12.

Als je een document gaat maken waar wiskundige formules in voorkomen dan raad ik je aan om ook [4, hoofdstuk 3] te lezen. De package `asmmath` die beschreven wordt in dit hoofdstuk wordt ook in deze handleiding beschreven stijl gebruikt. De in paragrafen 3.5.2 en 3.5.3 van [4] beschreven package `IEEEtrantools` moet je, als je deze wilt gebruiken, zelf opnemen in je document.

Tot slot van deze paragraaf, wil ik iedereen met voldoende tijd en historisch besef aanraden om het originele \LaTeX -boek *\LaTeX A Document Preparation System: User’s Guide and Reference Manual* van Lamport [3] te lezen.

2.1 Structuur

Deze handleiding bestaat uit een master document genaamd `main.tex`. De ‘stijl’ van het document wordt bepaald in de preamble en is in een apart bestand geplaatst genaamd `style_n1_NL.tex`. Er is ook een style voor Engelse documenten `style_en_US.tex`, zie [hoofdstuk 3](#). Veel documenten binnen een opleiding Elektrotechniek zullen programmeercode (C, MATLAB, enz) bevatten. De specifieke commando’s die daarvoor gebruik kunnen worden zijn in een apart bestand geplaatst genaamd `styleCode.tex`. De figuren zijn geplaatst in het directory `./figs` en de programma’s zijn geplaatst in het directory `./progs`.

Er is een makefile waarmee je het dictaat kunt compileren in twee versies: een papieren en een ebook versie. De papieren versie is bedoeld om dubbelzijdig te printen, heeft een bindmarge, hoofdstukken beginnen op oneven pagina’s. De ebook versie is enkelzijdig. De makefile maakt daarbij gebruik van het bestand `args.tex` om argumenten aan de \LaTeX -code mee te geven. Als je *géén* gebruik maakt van de makefile, kun je deze opties zelf aanpassen in `args.tex`. Als je *wél* gebruik maakt van de makefile, dan worden deze opties vanuit de makefile overschreven en moet je de opties dus in de makefile aanpassen.

De volgende opties zijn beschikbaar:

- `\ebookfalse`; Om de papieren versie te compileren.
- `\ebooktrue`; Om de ebook versie te compileren.
- `\chartertrue`; Om te kiezen voor de lettertypes Charter en Bera Sans.
- `\opensanstrue`; Om te kiezen voor de lettertypes Open Sans en Computer Modern Bright.
- `\timestrue`; Om te kiezen voor de lettertypes Times Roman en Helvetica.

Als beide opties `\chartertrue`, `\opensanstrue` en `\timestrue` niet gebruikt worden, dan worden de standaard Computer Modern lettertypes van \LaTeX gebruikt.

In de makefile maak ik (uiteraard) gebruik van \LaTeX -commando’s en ook van enkele eenvoudige Unix commando’s. Ik maak gebruik van Biber in plaats van BibTeX omdat Biber UTF8 ondersteunt. De paden naar deze executables moeten waarschijnlijk aangepast worden als je gebruik wilt maken van mijn makefile.

```
PDFLATEXOPT := -shell-escape -interaction=batchmode -file-line-error
BIBEROPT := --quiet
```

Als je in het document onderscheid wilt maken tussen de ebook en de papieren versie, dan kan dat door gebruik te maken van `\iftoggle{ebook}`. De code:

Dit is de `\iftoggle{ebook}{ebook}{papieren}` versie van dit dictaat.

geeft de volgende uitvoer: Dit is de ebook versie van dit dictaat.

De stijl file kan zowel gebruikt worden voor de documentclass `book` als `article`. Voor- dat `style_n1_NL.tex` wordt ingelezen moeten de commando's: `\myClass`, `\modulecode`, `\titel`, `\documenttype`, `\huidigeversie` en `\auteurs` gedefinieerd zijn.

Bijvoorbeeld:

```
\newcommand{\myClass}{book}
\newcommand{\modulecode}{LHR01}
\newcommand{\titel}{LaTeX-stijl HR}
\newcommand{\documenttype}{Handleiding}
\newcommand{\huidigeversie}{2.5}
% start a new line by using \\
\newcommand{\auteurs}{
  J.Z.M. Broeders
}
```

In een `book` begint elk hoofdstuk op een nieuwe pagina, in een `article` niet. De class `article` kun je gebruiken voor eenvoudige documenten zoals bijvoorbeeld een korte instructie¹⁰.

Als je de commando's `\frontmatter` en `\mainmatter` gebruikt in een `book`, dan worden de pagina's tussen deze twee commando's (in de papieren versie) Romeins genummerd. In de ebook versie worden de pagina's altijd gewoon doorgenummerd zodat het paginanummer dat door de reader wordt weergegeven overeenkomt met het werkelijke paginanummer.

2.2 Cursushandleiding

De in dit document beschreven stijl is ook te gebruiken om een cursushandleiding mee te maken. Er is daarvoor een speciale file `styleCursusHandleiding.tex` beschikbaar. Je kunt de cursushandleiding van EMS20¹¹ als voorbeeld gebruiken.

¹⁰ Zie: [Report_Requirements_TDS02.pdf](https://bitbucket.org/HR_ELEKTRO/tds02.pdf) en [Report_Requirements_TDS02_ebook.pdf](https://bitbucket.org/HR_ELEKTRO/tds02_ebook.pdf). De sourcecode vind je hier: https://bitbucket.org/HR_ELEKTRO/tds02/.

¹¹ Zie https://bitbucket.org/HR_ELEKTRO/ems20, de source is te vinden in de private repository van EMS20.

2.3 Tentamen

De in dit document beschreven commando's zijn ook te gebruiken om een tentamen mee te maken. Ivo Grondman heeft een class voor tentamens gemaakt `hrtoets.cls`. Er is een voorbeeldtoets¹² beschikbaar.

2.4 Titelpagina

Een titelpagina kun je eenvoudig maken met het commando `\titelpagina`. Bijvoorbeeld:

```
\titelpagina{academieplein}
```

Dit commando heeft één argument: de filenaam van het plaatje (zonder extensie) dat zich in het directory `./figs` moet bevinden. Neem een vectorplaatje of een rasterplaatje met een hoge resolutie (minstens 1024 pixels breed). De hoogte moet ongeveer $\frac{1}{2} \times$ de breedte zijn.

Dit commando maakt gebruik van de commando's `\modulecode`, `\titel`, `\documenttype`, `\huidigeversie` en `\auteurs`. Het logo (`./figs/logoHR.pdf`) en vervagende logo (`./figs/-logoHRfadetop.png`) zijn hard gecodeerd in de stijl maar kunnen daar natuurlijk wel aangepast worden. Deze bestanden moeten aanwezig zijn in het `./figs` directory.

2.5 Versiehistorie

Een tabel waarin de versiehistorie staat weergegeven kan eenvoudig aangemaakt worden met het commando `\versiehistorie`. Dit commando maakt gebruik van een `\longtabu` wat ervoor zorgt dat het geen probleem is als de tabel langer is dan een pagina. De tabel heeft 4 kolommen: Datum, Versie, Omschrijving en Auteur. De kolommen moeten gescheiden worden door een `&`-teken (zoals gebruikelijk bij tabellen in L^AT_EX). Gebruik `\\ \line` om de rijen te scheiden. Het is de bedoeling dat de meest recente versie bovenaan in de tabel komt.

In de kolom Auteur kun je gebruik maken van het commando `\mailafko` om de standaard HR-afkorting van de auteur aanklikbaar te maken. Door het aanklikken van de afkorting kan de auteur eenvoudig worden gemaïld.

Als je in de kolom Omschrijving gebruik wilt maken van een puntsgewijze lijst, dan moet je gebruik maken van de environment `itemizeintabu` omdat `\baselineskip` wordt aangepast in `\tabu`.

¹² In de private repository `hrtoets`.

De versiehistorie van dit document ziet er in \LaTeX -code als volgt uit:

```
\versiehistorie{
  01-01-2023 & \huidigeversie\footnotemark{} &
  Pinky assembler toegevoegd in \cref{sec:listings}.
  & \mailafko{BroJZ}
  \\ \hline
  31-12-2020 & 2.4 &
  Belangrijkste aanpassingen:
  \begin{itemizeintabel}
    %TODO: nog invullen
    \item Tabel commando's werken nu zonder gebruik van ↵
    ↵ package |tabu|.
    \item Nieuwe commando's \lcode{\tabelleX}, ↵
    ↵ \lcode{\tabelXm}, \lcode{\langetabelleX} en ↵
    ↵ \lcode{\langetabelleXm}.
    \item Enkele tips toegevoegd.
  \end{itemizeintabel}
  & \mailafko{BroJZ}
  \\ \hline
  28-12-2018 & 2.3 &
  Belangrijkste aanpassingen:
  \begin{itemizeintabel}
    \item Lege ruimte rond floats en regelafstand aangepast.
    \item Kleur \code{red} vervangen door \code{hrred}.
    \item \lcode{\hrfigure}\hyp{}commando als alternatief ↵
    ↵ voor \lcode{\figuur}\hyp{}commando.
    \item Commando's \lcode{\tikzfiguur} en ↵
    ↵ \lcode{\hrtikzfiguur}.
    \item Commando's \lcode{\tikzinlinefiguur} en ↵
    ↵ \lcode{\hrtikzinlinefiguur}.
    \item Commando's \lcode{\schakeling} en ↵
    ↵ \lcode{\hrschakeling}.
    \item Gekleurde opmaak van \LaTeX{}-code gebaseerd op ↵
    ↵ TeXstudio.
  \end{itemizeintabel}
  & \mailafko{BroJZ}
  \\ \hline
  23-02-2016 & 2.2 &
    \LaTeX{}-code verplaatst naar public bitbucket repository ↵
    ↵ en alle links hierop aangepast.\newline
    \lcode{\hcode}\hyp{}commando verbeterd.
```

```

& \mailafko{BroJZ}
\\ \hline
22-07-2015 & 2.1 &
    Voorbeelden toegevoegd.\newline
    \lcode{\hyp{}}commando verbeterd.
    Commando's \lcode{\pdfdeelref} en \lcode{\citepdfdeel} ←
↪ toegevoegd. Diverse commando's robuuster gemaakt.
& \mailafko{BroJZ}
\\ \hline
09-07-2015 & 2.0 &
    Eerste versie voor HR.
& \mailafko{BroJZ}
\\ \hline
09-04-2015 & 1.0 &
    Laatste versie voor HHS.
& \mailafko{BroJZ}
}

```

2.6 Licentie

Onder de versiehistorie kan de licentie voor het gebruik van het document vermeld worden. Ik adviseer om gebruik te maken van de Creative Commons Naamsvermelding-NietCommercieel-GelijkDelen 3.0 Nederland-licentie. Dit kan als volgt in de \LaTeX -code worden opgenomen:

```

\vspace*{\stretch{100}}
\includegraphics{figs/by-nc-sa_eu.pdf}
\par
\documenttype{} \titel{} van Hogeschool Rotterdam is in licentie ←
↪ gegeven volgens een ←
↪ \href{http://creativecommons.org/licenses/by-nc-sa/3.0/nl/} ←
↪ {Creative Commons ←
↪ Naamsvermelding\hyp{}}NietCommercieel\hyp{}}GelijkDelen 3.0 ←
↪ Nederland\hyp{}}licentie}.

```

2.7 Karakter encoding

Ik gebruik UTF8 source encoding, dit maakt het eenvoudig om karakters zoals ë, ï, é, è enz. in te voeren.¹³

2.8 Afbreken van woorden

Woorden met een koppelteken worden in L^AT_EX alleen afgebroken bij dat koppelteken. Dat geeft soms niet van die mooie resultaten.

Bijvoorbeeld de zin: heel erg erg veel data-elementen data-elementen data-elementen data-elementen en data-elementen en data-elementen data-elementen data-elementen data-elementen data-elementen data-elementen wordt ‘vreemd’ afgebroken.

De voorgaande zin geeft in de ebook versie twee overfull hboxen (de regels komen aan de rechterkant buiten de kantlijn terecht. Dit heb ik opgelost door het commando `\hyp{}` te definiëren. Als je de koppeltokens door dit commando vervangt, wordt het woord wel op andere plaatsen afgebroken.

Bijvoorbeeld de zin: heel erg erg veel data-elementen data-elementen data-elementen data-elementen data-elementen en data-elementen data-elementen data-elementen data-elementen data-elementen data-elementen data-elementen wordt correct afgebroken.

Als je Engelse woorden gebruikt in een Nederlands dictaat, dan worden die op een Nederlandse manier afgebroken. Om deze woorden correct af te laten breken moet je ze markeren. Dit kan eenvoudig gedaan worden met het commando `\USE` dat in de stijl gedefinieerd is.

2.9 Afbreken van pagina's

Om weduwen (enkele regels aan de bovenkant van een pagina) en wezen (enkele regels aan het einde van een pagina) te voorkomen gebruik ik de package `nowidow` met een optie waarbij er altijd 2 regels bijeen gehouden worden. Het commando `\noclub[n]` kan meteen na een paragraaf gebruikt worden om aan te geven dat de eerste n regels van deze paragraaf

¹³ Om er voor te zorgen dat de spatie achter enz. niet als een spatie aan het einde van de zin wordt gezien moet je de code `enz.\` gebruiken. Anders wordt de spatie na enz. te ver opgerekt. Maar dat wist je natuurlijk al lang.

bij elkaar moeten blijven en het commando `\nowidow[m]` kan meteen na een paragraaf gebruikt worden om aan te geven dat de laatste m regels van deze paragraaf bij elkaar moeten blijven. In het verleden gebruikte ik de package `needspace`, maar `nowidow` is veel handiger omdat je daarmee ‘automatisch’ weduwen en wezen kunt voorkomen. Package `needspace` kan nog steeds van pas komen als je meerdere paragrafen bij elkaar wilt houden, maar deze package is niet in `style_nl_NL.tex` opgenomen.

De papieren versie is two-sided en gebruikt standaard `\flushbottom` wat ervoor zorgt dat de inhoud van een pagina verticaal verdeeld wordt over de pagina door rubber spaces (ruimte tussen paragrafen en rondom floats) op te rekken. Als je dit niet wilt, dan moet je het commando `\raggedbottom` ergens in het begin van je document gebruiken. De ebook versie is one-sided en gebruikt standaard `\raggedbottom`.

2.10 Aanhalingstekens

Als je een woord tussen dubbele aanhalingstekens wilt plaatsen, dan kun je dat, zoals je vast wel weet, in L^AT_EX niet doen door het teken " te gebruiken omdat dit teken gebruikt wordt om karakters van een trema te voorzien. De code "extra" geeft dus `ëxtra`. Omdat ik UTF8 code gebruik kun je een `ë` gewoon intypen als `ë`. Het probleem met de aanhalingstekens kun je oplossen door de code ``trema'`: “trema”, maar in de stijl is het commando `\dquote` gedefinieerd waarmee je ook dubbele aanhalingstekens om een tekstfragment kunt plaatsten. Bijvoorbeeld: `\dquote{trema}` geeft: “trema”. Omdat in dit commando gebruik gemaakt wordt van de package `csquote` is het nu gemakkelijker om de stijl van de aanhalingstekens te veranderen (bijvoorbeeld in oldschool Nederlandse aanhalingstekens: „oudhollandse aanhalingstekens”), dit kun je dan doen door de parameter van de package `csquote` aan te passen.

Naast het commando `\dquote` is er ook het commando `\squote` waarmee een tekstfragment van enkele aanhalingstekens kan worden voorzien.

Volgens het genootschap Onze Taal gebruik je dubbele aanhalingstekens alleen om een letterlijk citaat weer te geven¹⁴ en gebruik je enkele aanhalingstekens in alle andere gevallen zoals: als alternatief voor cursivering, voor ironisch bedoelde woorden, bij een gefingeerd citaat en bij een citaat binnen een citaat¹⁵.

Het gebruik van aanhalingstekens in het Engels wijkt hier erg vanaf (zie: [paragraaf 3.4](#)).

¹⁴ <https://onzetaal.nl/taaladvies/advies/aanhalingstekens-dubbele-aanhalingstekens>

¹⁵ <https://onzetaal.nl/taaladvies/advies/enkele-aanhalingstekens>

2.11 Figuren

Er zijn verschillende soorten figuren die je in een document kunt gebruiken: figuren van internet en zelfgemaakte figuren. Je kunt figuren zelf maken in \LaTeX zelf maar ook m.b.v. andere software (zoals: PowerPoint, Visio, Inkscape, MATLAB, enz.). Figuren moeten voorzien zijn van een titel en een label (zodat je ernaar kunt verwijzen, zie [paragraaf 2.13](#)). Als je voor figuren een vectorafbeelding gebruikt, dan schaaft dat figuur mooi mee als je inzoomt in het document.

Figuren die niet in \LaTeX zelf gemaakt zijn bevinden zich in een apart bestand (met bijvoorbeeld de extensie `.pdf`, `.png` of `.jpg`). Er wordt vanuit gegaan dat deze in de directory `./figs` worden geplaatst. Ook wordt ervan uitgegaan dat het bestand alleen het kale figuur bevat (dus zonder titel of kader) en dat er zoveel mogelijk witruimte om het figuur heen verwijderd is.

Omdat ik de standaard door \LaTeX gebruikte witruimte boven en onder een figuur niet zo mooi vind en om er voor te zorgen dat de gebruiker van deze stijl niet vergeet om een titel en label op te geven is het commando `\figuur` in de stijl opgenomen.

Dit commando heeft vier parameters waarvan de eerste optioneel¹⁶ is. De eerste parameter bevat de ‘placement’ argumenten voor de `figure`-environment. Bijvoorbeeld `tb` of `H`, de default waarde is `!htbp`. \LaTeX zal dan de figuur proberen te plaatsen op de plaats waar de figuur gedefinieerd is in de code, als dat niet lukt aan de bovenkant van een pagina zo dicht mogelijk bij die plaats, als dat niet lukt aan de onderkant van een pagina zo dicht mogelijk bij die plaats en als dat niet lukt op een pagina met verzamelde figuren. Deze parameter moet alleen in uitzonderlijke gevallen gebruikt worden. We gebruiken niet voor niets \LaTeX om ons document vorm te geven: laat \LaTeX zijn werk doen, bijna altijd levert dit een betere layout dan handmatig ‘gepiel’.

Het tweede argument is (zijn) de optie(s) voor het `\includegraphics` commando. Bijvoorbeeld `height=5cm`, `scale=1.5`, of `width=.9\textwidth`. Het opgeven van de breedte als een factor van de tekstbreedte zorgt ervoor dat het figuur altijd past (zowel in de papieren als in de ebook versie). Het opgeven van een absolute waarde zorgt ervoor dat het figuur altijd even groot is.

De derde parameter is de filenaam (zonder extensie). Dit bestand moet in het directory `./figs` geplaatst zijn. De filenaam wordt ook als suffix van het label gebruikt: het label wordt

¹⁶ Optionele argumenten worden in \LaTeX tussen rechte haken opgegeven, verplichte argumenten worden tussen accolades opgegeven. Het `\figuur` commando heeft dus de volgende vorm: `rg2rg3rg4\figuur[arg1]aaa`.

`fig:filenaam`. Dit heeft wel als consequentie dat de filenaam geen tekens mag bevatten die niet in een L^AT_EX label gebruikt mogen worden. De vierde en laatste parameter is de titel van het figuur.

De code: `\figuur{width=.5\textwidth}{zebra}{Een zebra.}` produceert [figuur 2.1](#).



Figuur 2.1: Een zebra.

In plaats van het commando `\figuur` kun je ook het commando `\hrfigure` gebruiken. Dit commando heeft slechts twee parameters waarbij de eerste optioneel is. De tweede parameter geeft de filenaam (zonder extensie). Dit bestand moet in het directory `./figs` geplaatst zijn.

De eerste parameter bestaat uit `name=value`-paren. De volgende names kunnen gebruikt worden:

- `label` kan gebruikt worden om een specifiek label op te geven. De default waarde is `fig:filename`. Als dezelfde filenaam twee keer in een document gebruikt wordt, dan is het noodzakelijk om een specifiek label op te geven om te voorkomen dat het label `fig:filename` dubbel gedefinieerd wordt.
- `caption` kan gebruikt worden om de ondertitel op te geven. De default waarde is `filename`.

- `placement` kan gebruikt worden om de ‘placement’ argumenten voor de `\figure` environment te specificeren. De default waarde is `!htbp`.
- ... alle overige `name=value`-paren worden doorgegeven aan het `\includegraphics` commando.

De code: `\hrfigure[width=2cm, label=klzebra, caption=Kleine zebra.]{zebra}` produceert [figuur 2.2](#).



Figuur 2.2: Kleine zebra.

Als de witruimte onder of boven de figuur niet naar je zin is, dan kun je de `\vspace` commando's in de environment `myFigure` wijzigen.

2.11.1 Figuren van internet

Veel public domain vectorafbeeldingen in SVG-formaat zijn te vinden op Pixabay¹⁷. Je kunt Inkscape¹⁸ gebruiken om deze plaatje op te slaan in pdf-formaat. Indien nodig kun je overtollige witruimte verwijderen met het programma `pdfcrop`¹⁹.

2.11.2 Figuren zelf tekenen met een extern programma

Als je figuren zelf tekent met een extern programma probeer dan een programma te gebruiken waarmee je een vectorafbeelding²⁰ kan maken, bijvoorbeeld Visio, PowerPoint of Inkscape. Sla de figuur op in het pdf-formaat of print het naar een pdf-bestand. Overtollige witruimte kan worden verwijderd met het al eerder genoemde programma `pdfcrop`.

¹⁷ <http://pixabay.com>

¹⁸ <https://inkscape.org/en/>

¹⁹ Dit programma is onderdeel van de Tex Live distributie (<http://www.tug.org/texlive/>, zie <http://archive.cs.uu.nl/mirror/CTAN/support/pdfcrop/README>).

²⁰ Zie: https://en.wikipedia.org/wiki/Vector_graphics.

2.11.3 Grafieken aanmaken met MATLAB

MATLAB kan goed gebruikt worden om grafieken te maken. De MATLAB-code `lpfbodelog.m` die gegeven is in [listing 2.1](#) produceert het bodediagram dat in [figuur 2.3](#) is weergegeven.

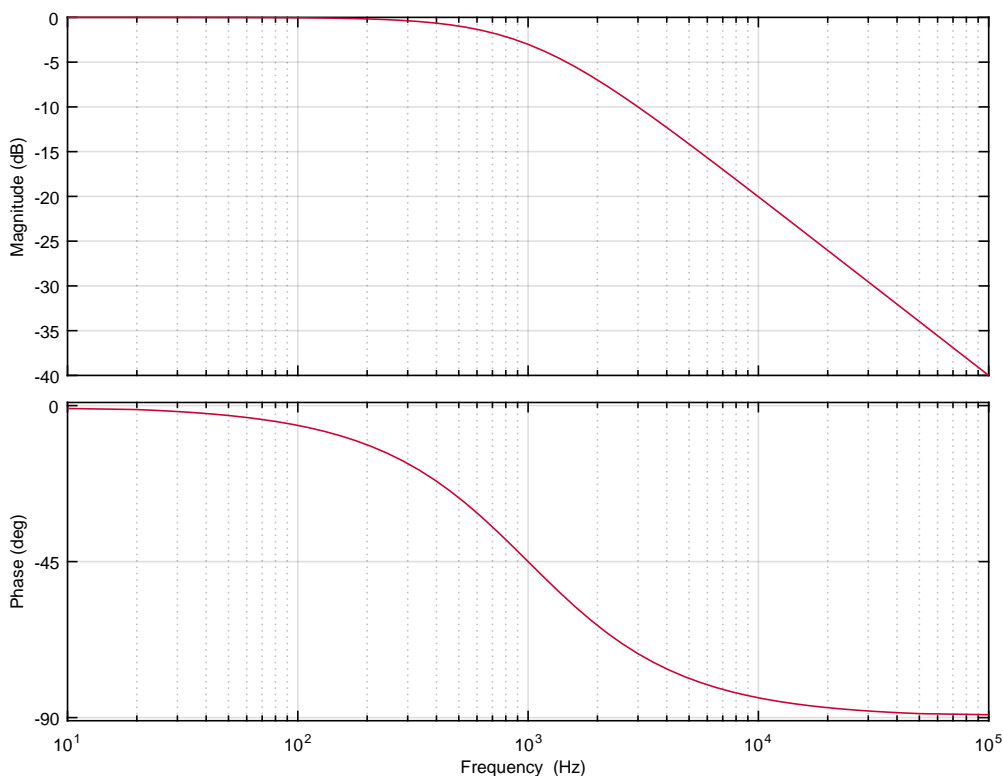
```
hs = tf(1000*2*pi, [1 1000*2*pi]);
opts = bodeoptions;
opts.Title.String = '';
opts.FreqUnits = 'Hz';
opts.FreqScale = 'log';
opts.grid = 'on';
h = bodeplot(hs, opts);
L = findobj(gcf, 'type', 'line');
set(L, 'Color', [204/255 0 51/255]);
%generate pdf file for use in LaTeX
file = '../figs/lpfbodelog';
print(gcf, '-dpdf', file);
system(sprintf('pdftocrop %s.pdf %s.pdf', file, file));
```

Listing 2.1: MATLAB programma dat een bodediagram van een laagdoorlaatfilter produceert, zie [figuur 2.3](#).

2.11.4 Screenshots

Screenshots komen vaak voor in handleidingen waarin het gebruik van een of ander programma wordt uitgelegd. Maak géén gebruik van screenshots als je ook gebruik kunt maken van een vectorafbeelding. Sla screenshots op in het `.png`-formaat (en niet in het `.jpg`-formaat) om er voor te zorgen dat het screenshot er in het document exact zo uitziet als op het scherm.

Zoals al eerder vermeld moet je de neiging om figuren op een vaste plaats te dwingen zo snel mogelijk afleren. \LaTeX weet het meestal beter! Screenshots in tutorials die stapsgewijs moeten worden doorlopen vormen hier een uitzondering op en je kunt hier als eerste optionele parameter van het commando `\figuur` het argument `H` gebruiken als het screenshot op een vaste plaats in de tekst moet komen te staan.



Figuur 2.3: Bodediagram van een laagdoorlaatfilter met een kantelfrequentie van 1 kHz.

2.11.5 Tekenen in \LaTeX met TikZ

In plaats van een extern tekenprogramma kun je tekeningen ook rechtstreeks in \LaTeX maken met behulp van de package `tikz`²¹. Dit is een zeer uitgebreide package met een vrij steile leercurve²². Er zijn wel hele mooie resultaten mee te behalen. Deze package is in de stijl opgenomen en wordt gebruikt om de titlepage te ‘tekenen’. De `tikz`libraries `fadings` en `shadows` worden daar ook in gebruikt. Daarnaast worden ook de `tikz`libraries `shapes`, `arrows`, `arrows.meta` en `positioning` in de stijl gebruikt, maar overige `tikz`libraries moet je zelf opnemen in de preamble van je document met het commando `\usetikzlibrary`.

Er zijn, met dank aan Daniël Versluis, verschillende commando’s opgenomen in de stijl om TikZ-tekeningen in een document op te nemen. Het commando `\tikzfiguur` kan

²¹ <https://www.ctan.org/pkg/pgf>

²² Ik raad je sterk aan om de tutorials uit deel I van de `pgfmanual` <http://ctan.cs.uu.nl/graphics/pgf/base/doc/pgfmanual.pdf> te bestuderen.

gebruikt worden om een `tikzpicture`-environment die in een apart document is geplaatst in een document op te nemen. Het commando is vergelijkbaar met het [paragraaf 2.11](#) besproken commando `\figuur`. Het commando `\tikzfiguur` heeft drie parameters waarvan de eerste optioneel is. De eerste parameter bevat de ‘placement’ argumenten voor de `\figure` environment. Zoals al eerder vermeld is, moet deze parameter alleen in uitzonderlijke gevallen gebruikt worden. De tweede parameter is de filenaam (zonder extensie). Dit bestand moet in het directory `./tikz` geplaatst zijn. De filenaam wordt ook als suffix van het label gebruikt: het label wordt `fig:filenaam`. Dit heeft wel als consequentie dat de filenaam geen tekens mag bevatten die niet in een \LaTeX label gebruikt mogen worden. De derde en laatste parameter is de titel van het figuur.

De code die gegeven is in [listing 2.2](#) produceert het blokdiagram dat in [figuur 2.4](#) is weergegeven. De inhoud van het bestand `dsp.tex` wordt weergegeven in [listing 2.3](#).

```
\tikzfiguur{dsp}{Digitale signaalbewerking in een analoge omgeving.}
```

Listing 2.2: \LaTeX -code waarmee [figuur 2.4](#) gemaakt is.



Figuur 2.4: Digitale signaalbewerking in een analoge omgeving.

Het in een apart (los) bestand plaatsen van de TikZ-code heeft een groot voordeel: het bestand kan dan ook bewerkt worden met andere programma’s. Ik kan de WYSIWYG editor `TikzEdt`²³ van harte aanbevelen.

In plaats van het commando `\tikzfiguur` kun je ook het commando `\hrtikzfigure` gebruiken. Dit commando heeft slechts twee parameters waarbij de eerste optioneel is. De tweede parameter geeft de filenaam (zonder extensie). Dit bestand moet in het directory `./tikz` geplaatst zijn.

De eerste parameter bestaat uit `name=value`-paren. De volgende names kunnen gebruikt worden:

- `label` kan gebruikt worden om een specifiek label op te geven. De default waarde is `fig:filename`. Als dezelfde filenaam twee keer in een document gebruikt wordt, dan is het noodzakelijk om een specifiek label op te geven om te voorkomen dat het label `fig:filename` dubbel gedefinieerd wordt.

²³ <http://www.tikzedt.org/>.

```

\tikzset{%
  block/.style = {draw, thick, rectangle, minimum height = 3em, ←
  ↪ minimum width = 3em, inner sep=1em, font=\sffamily\Large},
}
\begin{tikzpicture}[auto, thick, ->, >={Stealth[scale=1.2, ←
  ↪ inset=0.6pt]}, shorten >=1pt]
  \draw
    node (text) {$x(t)$}
    node [coordinate, node distance=.5cm, right of=text] ←
  ↪ (input) {}
    node [block, node distance=2cm, right of=input] (adc) {ADC}
    node [block, node distance=3.5cm, right of=adc] (dsp) {DSP}
    node [block, node distance=3.5cm, right of=dsp] (dac) {DAC}
    node [coordinate, node distance=2cm, right of=dac] ←
  ↪ (output) {}
    node [node distance=.5cm, right of=output] {$y(t)$};
  \draw (input) -- (adc);
  \draw (adc) -- node[pos=.4] {$x[n]$} (dsp);
  \draw (dsp) -- node[pos=.4] {$y[n]$} (dac);
  \draw (dac) -- (output);
\end{tikzpicture}

```

Listing 2.3: TikZ-code uit het bestand dsp.tex waarmee [figuur 2.4](#) gemaakt is.

- caption kan gebruikt worden om de ondertitel op te geven. De default waarde is filename.
- placement kan gebruikt worden om de ‘placement’ argumenten voor de `figure`-environment te specificeren. De default waarde is `!htbp`.

De code gegeven in [listing 2.4](#) produceert het blokdiagram dat in [figuur 2.5](#) is weergegeven.

```

\hrtikzfigure[label=dsp2, caption=Herhaling van \cref{fig:dsp}.]{dsp}

```

Listing 2.4: \LaTeX -code waarmee [figuur 2.5](#) gemaakt is.



Figuur 2.5: Herhaling van [figuur 2.4](#).

Als je de TikZ-code liever in het document zelf opneemt, dan kun je het commando `\tikzinlinefigure` gebruiken. Dit commando heeft vier parameters waarvan de eerste optioneel

is. De eerste parameter bevat de ‘placement’ argumenten voor de `figure`-environment. De tweede parameter is het suffix van het label (`fig:suffix`). De derde parameter bevat de TikZ-code die in een `tikzpicture`-environment zal worden geplaatst. De vierde en laatste parameter is de titel van het figuur.

De code gegeven in [listing 2.5](#) produceert het blokdiagram dat in [figuur 2.6](#) is weergegeven.

```
\tikzinlinefiguur{adc}{
  \tikzset{auto, thick, >={Stealth[scale=1.2, inset=0.6pt]}, ←
  ↪ shorten >=1pt, node distance=1.5cm}
  \draw
    node [coordinate] (input) {}
    node [draw, thick, rectangle, minimum height = 3em, ←
  ↪ minimum width = 3em, inner sep=1em, font=\sffamily\Large, ←
  ↪ right = of input] (adc) {ADC}
    node [coordinate, right = of adc] (output) {}
    (input) edge[->] node{\$x(t)\$} (adc)
    (adc) edge[->] node{\$x[n]\$} (output);
}{Een Analog to Digital Converter.}
```

Listing 2.5: \LaTeX -code waarmee [figuur 2.6](#) gemaakt is.



Figuur 2.6: Een Analog to Digital Converter.

In plaats van het commando `\tikzinlinefiguur` kun je ook het commando `\hrtikzinlinefigure` gebruiken. Dit commando heeft slechts twee parameters waarbij de eerste optioneel is. De tweede parameter bevat de TikZ-code die in een `tikzpicture`-environment zal worden geplaatst.

De eerste parameter bestaat uit `name=value`-paren. Dezelfde names als bij het commando `\hrtikzfiguur`, zie [pagina 21](#) kunnen gebruikt worden: `label`, `caption` en `placement`.

De code gegeven in [listing 2.6](#) produceert het blokdiagram dat in [figuur 2.7](#) is weergegeven.



Figuur 2.7: Nog een Analog to Digital Converter.

```

\hrtikzinlinefigure[label=adc2, caption=Nog een Analog to Digital ←
  ↪ Converter.]{
  \tikzset{auto, thick, >={Stealth[scale=1.2, inset=0.6pt]}, ←
  ↪ shorten >=1pt, node distance=1.5cm}
  \draw
    node [coordinate] (input) {}
    node [draw, thick, rectangle, minimum height = 3em, ←
  ↪ minimum width = 3em, inner sep=1em, font=\sffamily\Large, ←
  ↪ right = of input] (adc) {ADC}
    node [coordinate, right = of adc] (output) {}
    (input) edge[->] node{\$x(t)\$} (adc)
    (adc) edge[->] node{\$x[n]\$} (output);
}

```

Listing 2.6: \LaTeX -code waarmee [figuur 2.7](#) gemaakt is.

2.11.6 Elektronische schema's tekenen in \LaTeX met `circuitikz`

Elektronische schema's kunnen ook rechtstreeks in \LaTeX gemaakt worden met behulp van de package `circuitikz`²⁴ die in deze stijl is opgenomen. Deze package bouwt weer voort op TikZ. Door Daniël Versluis is het commando `\schakeling` gedefinieerd waarmee een elektronisch schema gedefinieerd kan worden. Dit commando heeft vier parameters waarvan de eerste optioneel is. De eerste parameter bevat de 'placement' argumenten voor de `figure`-environment. De tweede parameter bevat de `circuitikz`-code die in een `circuitikz`-environment zal worden geplaatst. De derde parameter is de titel van het figuur. De vierde en laatste parameter is de suffix van het label (`sch:suffix`). Dit zijn dezelfde parameters als het commando `\tikzinlinefiguur` alleen in een andere volgorde²⁵.

De code gegeven in [listing 2.7](#) produceert het schema dat in [schakeling 2.1](#) is weergegeven.

In plaats van het commando `\schakeling` kun je ook het commando `\hrschakeling` gebruiken. Dit commando heeft slechts twee parameters waarbij de eerste optioneel is. De tweede parameter bevat de `circuitikz`-code die in een `circuitikz`-environment zal worden geplaatst.

De eerste parameter bestaat uit `name=value`-paren. Dezelfde names als bij het commando `\hrtikzfiguur`, zie [pagina 21](#) kunnen gebruikt worden: `label`, `caption` en `placement`.

²⁴ <https://www.ctan.org/pkg/circuitikz>

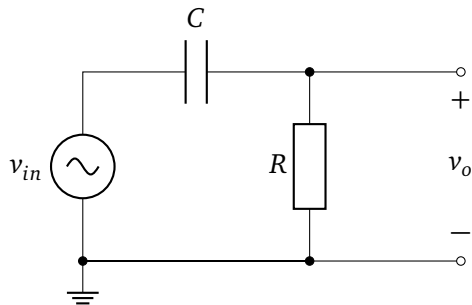
²⁵ Deze historische fout is nu niet meer te herstellen omdat er al te veel gebruik gemaakt is van deze commando's.


```

\schakeling{
  \draw
    (0,0) node[ground] (gnd){}
    to[sV, l=$v_{in}$, *-] ++(0,2.5)
    to[C, l=$C$] ++(3,0) coordinate(C)
    to[R, l_=$R$, *-] (C |- gnd)
    to[short] (gnd)
    (C) to[short] ++(2,0) coordinate(vout)
    to[open, v^=$v_o$, o-o] (vout |- gnd)
    to[short] (gnd);
}{Een CR spanningsdeler.}{RChigh}

```

Listing 2.7: \LaTeX -code waarmee [schakeling 2.1](#) gemaakt is.



Schakeling 2.1: Een CR spanningsdeler.

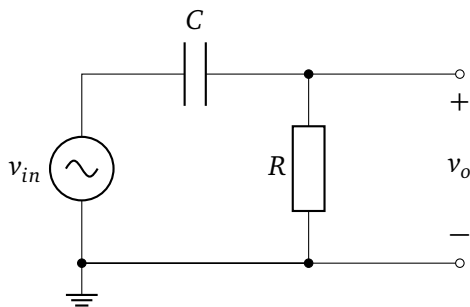
De code gegeven in [listing 2.8](#) produceert het schema dat in [schakeling 2.2](#) is weergegeven.

```

\hrschakeling[label=RChigh, caption=Nog een RC-filter.]{
  \draw
    (0,0) node[ground] (gnd){}
    to[sV, l=$v_{in}$, *-] ++(0,2.5)
    to[C, l=$C$] ++(3,0) coordinate(C)
    to[R, l_=$R$, *-] (C |- gnd)
    to[short] (gnd)
    (C) to[short] ++(2,0) coordinate(vout)
    to[open, v^=$v_o$, o-o] (vout |- gnd)
    to[short] (gnd);
}

```

Listing 2.8: \LaTeX -code waarmee [schakeling 2.2](#) gemaakt is.



Schakeling 2.2: Nog een RC-filter.

Elektrische schema's kunnen in plaats van met `circuitikz` ook met behulp van de `Circuit libraries`²⁶ van `TikZ` getekend worden. Deze `TikZ`-libraries zijn gebaseerd op `circuitikz`. Het is mij niet duidelijk welke aanpak de voorkeur heeft. Lees paragraaf 1.4 van de `circuitikz` documentatie²⁷ als je beide wilt combineren.

2.11.7 Grafieken tekenen in \LaTeX met PGFPLOTS

Grafieken kunnen ook rechtstreeks in \LaTeX gemaakt worden met behulp van de package `pgfplots`²⁸. Deze package bouwt weer voort op `TikZ`. De package `pgfplots` is niet in de stijl opgenomen. Voorbeelden van het gebruik van `pgfplots` vind je op <http://pgfplots.sourceforge.net/gallery.html>. De code die gegeven is in `listing 2.9` produceert de grafiek die in `figuur 2.8` is weergegeven.

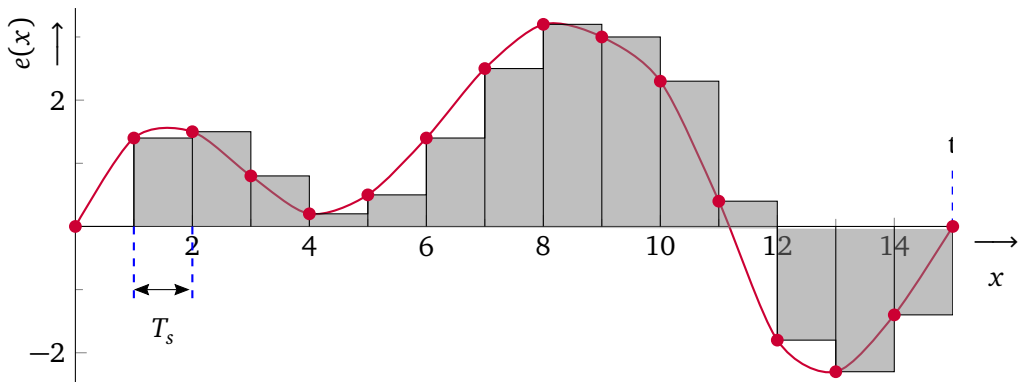
Grafieken kunnen in plaats van met `PGFPLOTS` ook met behulp van de `Data Visualizations` bibliotheek²⁹ van `TikZ` getekend worden. Het is mij niet duidelijk welke aanpak de voorkeur heeft.

²⁶ Zie hoofdstuk 47 van de `pgfmanual`: <http://ctan.cs.uu.nl/graphics/pgf/base/doc/pgfmanual.pdf#section.47>.

²⁷ <http://ctan.cs.uu.nl/graphics/pgf/contrib/circuitikz/doc/circuitikzmanual.pdf#4>.

²⁸ <https://www.ctan.org/pkg/pgfplots>

²⁹ Zie hoofdstuk 75 van de `pgfmanual`: <http://ctan.cs.uu.nl/graphics/pgf/base/doc/pgfmanual.pdf#section.75>.



Figuur 2.8: Numeriek integreren volgens de Forward Rectangular Method (FRM).

```

\begin{myFigure}
  \centering%
  \begin{tikzpicture}[>={Stealth[scale=1.2,inset=0.6pt]}, ←
↪ shorten >=1pt]
  \begin{axis}[
width=0.9\textwidth, height=0.45\textwidth,
minor tick num=1, axis lines*=middle,
xmin=0, xmax=15, ymin=-2.5, ymax=3.5,
xlabel style={at={{ticklabel* cs:1.05}}, anchor=near ←
↪ ticklabel, align=center},
xlabel={\longrightarrow \ \ $x$},
ylabel style={at={{ticklabel* cs:.9}}, anchor=near ticklabel, ←
↪ align=center},
ylabel={e(x) \ \ \longrightarrow },
]
  \addplot[thick, smooth, hrred, mark=*] table {data/data1.dat};
  \addplot[const plot, fill=gray, fill opacity=0.5] table ←
↪ {data/data1.dat};
  \addplot[ycomb, black, mark=none] table {data/data1.dat};
  \addplot[thick, dashed, blue] coordinates {(15,0)(15,1)}
node[above, black] {t};
  \addplot[thick, dashed, blue] coordinates {(1,0)(1,-1.3)};
  \addplot[thick, dashed, blue] coordinates {(2,0)(2,-1.3)};
  \addplot[thick, blue] coordinates {(2,-1)};
  \node[coordinate] (a1) at (axis cs:1,-1) {};
  \node[coordinate] (a2) at (axis cs:2,-1) {};
  \draw[<->] (a1) -- (a2)
node[anchor=north] at (axis cs:1.5,-1.3) {$T_s$};
  \end{axis}
  \end{tikzpicture}
  \caption{Numeriek integreren volgens de Forward Rectangular ←
↪ Method (FRM).}
  \label{fig:frm}
\end{myFigure}

```

Listing 2.9: \LaTeX -code waarmee [figuur 2.8](#) gemaakt is.

2.12 Tabellen

Voor tabellen werd in eerdere versies van deze style de package `tabu`³⁰ gebruikt. Deze optie geeft echter problemen sinds de 2018 versie van `array`. De auteur van `array` schrijft op 11 mei 2018: “I have mailed the tabu author pointing out this incompatibility, so hopefully he’ll update to match the latest `array`”. Op 14 juni 2018 voegt iemand daaraan toe: “Don’t hold your breath”³¹.

Inmiddels zijn de voorheen met `tabu` gedefinieerde commando’s vervangen door commando’s die gebruik maken van `\tabular`³² en `tabularx`³³.

Voor normale tabellen die gebruik maken van `tabular` kunnen de commando’s: `\tabel`, `\tabelmaxpaginabreedteschaald` en `\langetabel` gebruikt worden.

Het commando `\tabel` heeft zes parameters waarvan de eerste optioneel is. De eerste parameter bevat de ‘placement’ argumenten voor de `table`-environment. De default waarde is `!htbp`. De tweede parameter is de titel van de tabel. De derde parameter is de suffix van het label, de labelnaam wordt `tab:derde-argument`. De vierde parameter is de kolomdefinitie voor de `tabular`-environment zoals gedefinieerd in de package `array`³⁴. De vijfde parameter is de eerste rij van de tabel (de kop van de tabel) en de laatste parameter is de inhoud van de tabel. Kolommen worden gescheiden door het `&`-teken en rijen worden gescheiden door `\`, zoals gebruikelijk in L^AT_EX. Als je een regel wilt afbreken in een tabel, dan kun je het commando `\newline` gebruiken.

De code die gegeven is in [listing 2.10](#) produceert [tabel 2.1](#).

Het commando `\tabelmaxpaginabreedteschaald` heeft exact dezelfde parameters als het commando `\tabel`. Het enige verschil is dat het commando `\tabelmaxpaginabreedteschaald` de tabel zodanig schaalt dat hij de volledige tekstbreedte van de pagina benut. Dit is vooral handig als de tabel bij het gebruik van het commando `\tabel` te breed is om op de pagina te passen.

³⁰ <https://www.ctan.org/pkg/tabu>

³¹ <https://tex.stackexchange.com/questions/430973/tabu-m-column-vertical-alignment-not-working-after-package-updates>.

³² <http://joshua.smcvt.edu/latex2e/tabular.html>

³³ <https://www.ctan.org/pkg/tabularx>

³⁴ <https://www.ctan.org/pkg/array>

```
\tabel{Zomaar een tabel.}{zomaar}{lrr}{
  Getal & Decimaal & Hexadecimaal
}{
  een & 1 & 1 \\
  tien & 10 & A \\
  twintig & 20 & 14 \\
  tweehonderdvijfenvijftig & 255 & FF
}
```

Listing 2.10: \LaTeX -code waarmee tabel 2.1 gemaakt is.

Tabel 2.1: Zomaar een tabel.

Getal	Decimaal	Hexadecimaal
een	1	1
tien	10	A
twintig	20	14
tweehonderdvijfenvijftig	255	FF

Het commando `\langetabel` moet gebruikt worden als een tabel langer is dan een pagina. Dit commando zorgt ervoor dat de tabel netjes wordt verdeeld over meerdere pagina's. De parameters zijn bijna gelijk aan die van het commando `\tabel`. De eerste (optionele) parameter ontbreekt en er is een extra parameter waaraan het aantal kolommen moet worden meegegeven. Een lange tabel is geen 'floating' object en wordt dus altijd op de plaats waar hij gedefinieerd is neergezet. Dit kan er toe leiden dat een tabel die eigenlijk wel op een pagina past toch verdeeld wordt over meerdere pagina's. Gebruik het commando `\langetabel` dus alleen als een tabel die gemaakt is met het commando `\tabel` niet op een pagina past.

Als de ruimte om een tabel je niet bevalt, dan kun je de `\vspace` commando's in de environment `myTable` aanpassen.

Als je tabellen wilt maken waarbij de tekst in een kolom uitgevuld wordt, dan moet je gebruik maken van commando's die gebaseerd zijn op `tabularx`. Dit zijn de commando's: `\tabelX` en `\langetabelX`.

Deze commando's gebruiken normaal gesproken de volledige tekstbreedte. Je kunt als eerste optionele parameter ook de gewenste breedte opgeven. Bij deze commando's kun je gebruik maken van de kolomspecificatie X. Hiermee kun je kolommen 'uitvullen' zodat de tabel de gehele tekstbreedte van de pagina gebruikt. De kolomspecificatie X zorgt er bijvoorbeeld

voor dat de tekst in de kolom met rechte kantlijnen links en rechts uitgevuld en uitgelijnd wordt. De code die gegeven is in [listing 2.11](#) produceert [tabel 2.2](#).

```
\tabelX{Zomaar een uitgevulde tabel.}{gebruikX}{XX}{
  Kolom 1 & Kolom2
}{
  Zomaar een stukje tekst om te laten zien hoe deze tekst ↔
  ↔ uitgevuld en uitgelijnd wordt.&
  Zomaar een stukje tekst om te laten zien hoe deze tekst ↔
  ↔ uitgevuld en uitgelijnd wordt.
  De rechterkolom zal net zo breed zijn als de linker.
}
```

Listing 2.11: \LaTeX -code waarmee [tabel 2.2](#) gemaakt is.

Tabel 2.2: Zomaar een uitgevulde tabel.

Kolom 1	Kolom2
Zomaar een stukje tekst om te laten zien hoe deze tekst uitgevuld en uitgelijnd wordt.	Zomaar een stukje tekst om te laten zien hoe deze tekst uitgevuld en uitgelijnd wordt. De rechterkolom zal net zo breed zijn als de linker.

Het is ook mogelijk om de kolommen die uitgevuld worden niet even breed te maken. De code die gegeven is in [listing 2.12](#) produceert [tabel 2.3](#). Hierbij is de verhouding tussen de linker- en rechterkolom $0.67 : 1.33$ oftewel $1 : 2$. Het totaal van deze verhoudingsgetallen moet gelijk zijn aan het aantal X -kolommen. In dit geval $0.67 + 1.33 = 2$.

Tabel 2.3: Zomaar een uitgevulde tabel met kolommen van ongelijke breedte.

Kolom 1	Kolom2
Zomaar een stukje tekst om te laten zien hoe deze tekst uitgevuld en uitgelijnd wordt.	Zomaar een stukje tekst om te laten zien hoe deze tekst uitgevuld en uitgelijnd wordt. De rechterkolom zal twee keer zo breed zijn als de linker.

Als je de X -kolom alleen links, alleen rechts of helemaal niet uitgevuld wilt hebben, dan kan dat door een prefix toe te voegen aan de X -kolomspecificatie, zie [tabel 2.4](#).

```

\tableX{Zomaar een uitgevulde tabel met kolommen van ongelijke ←
  ← breedte.}{gebruikXongelijk}{ ←
  ← >{\hsize=0.67\hsize\linewidth=\hsize}X ←
  ← >{\hsize=1.33\hsize\linewidth=\hsize}X}{
  Kolom 1 & Kolom2
}{
  Zomaar een stukje tekst om te laten zien hoe deze tekst ←
  ← uitgevuld en uitgelijnd wordt.&
  Zomaar een stukje tekst om te laten zien hoe deze tekst ←
  ← uitgevuld en uitgelijnd wordt.
  De rechterkolom zal twee keer zo breed zijn als de linker.
}

```

Listing 2.12: \LaTeX -code waarmee [tabel 2.3](#) gemaakt is.

Tabel 2.4: Verschillende prefixen van de X-kolomspecificator.

uitlijning	X-kolomspecificatie met prefix
niet	<code>>{\centering\arraybackslash}X</code>
rechts	<code>>{\raggedleft\arraybackslash}X</code>
links	<code>>{\raggedright\arraybackslash}X</code>
links en rechts	<code>X</code>

Het commando `\arraybackslash` is alleen maar nodig in de prefix als in de tabel zelf het commando `\newline` gebruikt wordt. De code die gegeven is in [listing 2.13](#) produceert [tabel 2.5](#).

Zoals je in [tabel 2.2](#) kunt zien worden de regels verticaal aan de top van de regel uitgelijnd. Als je de regels verticaal uitgelijnd wilt hebben op het midden van de regel, dan moet je een van de commando's `\tableXm` of `\langetableXm` gebruiken.

De code die gegeven is in [listing 2.14](#) produceert [tabel 2.6](#).

2.13 Verwijzingen en citaten

De verwijzingen naar een tabel, figuur enz. maak ik met `\cref{label}`, de tekst tabel, figuur enz. wordt er dan vanzelf bijgezet en dit wordt ook onderdeel van de link. Bijvoor-


```

\tableX{Tabel met X kolommen.}{gebruikXspecs}{>\centering}X ←
  ↪ >\raggedleft}X >\raggedright}XX}{
  \texttt{>\textbackslash}centering\}X} &
  \texttt{>\textbackslash}raggedleft\}X} &
  \texttt{>\textbackslash}raggedright\}X} &
  \texttt{X}
}{
  Deze iets langere tekst komt te staan in kolom 1 en zou niet ←
  ↪ uitgelijnd moeten zijn. & Deze iets langere tekst komt te ←
  ↪ staan in kolom 2 en zou links uitgelijnd moeten zijn. & Deze ←
  ↪ iets langere tekst komt te staan in kolom 3 en zou rechts ←
  ↪ uitgelijnd moeten zijn. & Deze iets langere tekst komt te ←
  ↪ staan in kolom 4 en zou uitgevuld moeten zijn. \\
  Deze iets langere tekst komt te staan in kolom 1 en zou niet ←
  ↪ uitgelijnd moeten zijn. & Deze iets langere tekst komt te ←
  ↪ staan in kolom 2 en zou links uitgelijnd moeten zijn. & Deze ←
  ↪ iets langere tekst komt te staan in kolom 3 en zou rechts ←
  ↪ uitgelijnd moeten zijn. & Deze iets langere tekst komt te ←
  ↪ staan in kolom 4 en zou uitgevuld moeten zijn.
}

```

Listing 2.13: \LaTeX -code waarmee [tabel 2.5](#) gemaakt is.

Tabel 2.5: Tabel met X kolommen.

<code>>\centering}X</code>	<code>>\raggedleft}X</code>	<code>>\raggedright}X</code>	<code>X</code>
Deze iets langere tekst komt te staan in kolom 1 en zou niet uitgelijnd moeten zijn.	Deze iets langere tekst komt te staan in kolom 2 en zou links uitgelijnd moeten zijn.	Deze iets langere tekst komt te staan in kolom 3 en zou rechts uitgelijnd moeten zijn.	Deze iets langere tekst komt te staan in kolom 4 en zou uitgevuld moeten zijn.
Deze iets langere tekst komt te staan in kolom 1 en zou niet uitgelijnd moeten zijn.	Deze iets langere tekst komt te staan in kolom 2 en zou links uitgelijnd moeten zijn.	Deze iets langere tekst komt te staan in kolom 3 en zou rechts uitgelijnd moeten zijn.	Deze iets langere tekst komt te staan in kolom 4 en zou uitgevuld moeten zijn.

beeld de code `\cref{tab:zomaar}` geeft: [tabel 2.1](#). Andere voorbeelden van verwijzingen: [hoofdstuk 1](#) en [figuren 2.1](#) en [2.8](#). De \LaTeX -code van de vorige zin is:

```

\tableXm{Zomaar een uitgevulde tabel waarbij de regels verticaal ←
  ↪ in het midden uitgelijnd worden}{gebruikXm}{XX}{
  Kolom 1 & Kolom2
}{
  Zomaar een stukje tekst om te laten zien hoe deze tekst ←
  ↪ uitgevuld en uitgelijnd wordt.&
  Zomaar een stukje tekst om te laten zien hoe deze tekst ←
  ↪ uitgevuld en uitgelijnd wordt.
  De rechterkolom zal net zo breed zijn als de linker.\
  Tweede regel & Zomaar een stukje tekst om te laten zien hoe ←
  ↪ deze tekst uitgevuld en uitgelijnd wordt.
  De rechterkolom zal net zo breed zijn als de linker.
}

```

Listing 2.14: L^AT_EX-code waarmee tabel 2.6 gemaakt is.

Tabel 2.6: Zomaar een uitgevulde tabel waarbij de regels verticaal in het midden uitgelijnd worden

Kolom 1	Kolom2
Zomaar een stukje tekst om te laten zien hoe deze tekst uitgevuld en uitgelijnd wordt.	Zomaar een stukje tekst om te laten zien hoe deze tekst uitgevuld en uitgelijnd wordt. De rechterkolom zal net zo breed zijn als de linker.
Tweede regel	Zomaar een stukje tekst om te laten zien hoe deze tekst uitgevuld en uitgelijnd wordt. De rechterkolom zal net zo breed zijn als de linker.

Andere voorbeelden van verwijzingen: `\cref{sec:inleiding}`, `\cref{fig:zebra, fig:frm}`.

Als een zin met een verwijzing begint, dan kun je `\Cref` gebruiken om de verwijzing met een hoofdletter te laten beginnen. [Figuur 2.1](#) staat op [pagina 17](#). De voorgaande regel is als volgt gecodeerd:

```
\Cref{fig:zebra} staat op \cpageref{fig:zebra}.
```

Om te verwijzen naar een section of chapter is het commando `\csecref` in de stijl gedefinieerd dat, alleen in de printed versie, een verwijzing naar de betreffende pagina toevoegt.

Bijvoorbeeld de code `\cscoref{sec:inleiding}` geeft: [hoofdstuk 1](#) in de ebook versie en [hoofdstuk 1 \(pagina 6\)](#) in de papieren versie.

Als je naar een met L^AT_EX gemaakt document verwijst, dan kun je een link opnemen naar een specifiek gedeelte van dit document met behulp van het in de stijl gedefinieerde commando `\pdfdeelfref`.

Zo verwijst deze link: [paragraaf 6.2](#) naar paragraaf 6.2 van mijn C++-dictaat. De voorgaande link is als volgt gecodeerd:

```
\pdfdeelfref{section}{paragraaf~}{%
https://bytebucket.org/HR_ELEKTRO/cppprog/wiki/Dictaat_OPinCpp%
\iftoggle{ebook}{_ebook}{}.pdf}{6.2}
```

Als eerste argument wordt de naam van het betreffende gedeelte, b.v. chapter, section, subsection, enz. opgegeven. Als tweede argument wordt de naam voor het betreffende gedeelte, b.v. hoofdstuk, paragraaf, enz. opgegeven. De URL van het betreffende pdf-document moet als derde argument worden opgegeven. Als vierde argument wordt het nummer van het betreffende gedeelte opgegeven.

Merk op dat de link in de ebook versie van dit document verwijst naar de ebook versie van het dictaat en dat de link in de ‘papieren’ versie verwijst naar de ‘papieren’ versie van het dictaat.

De code:

```
Bla, bla, zie \pdfdeelfref{chapter}{hoofdstuk~}{%
https://bytebucket.org/HR_ELEKTRO/dis10/wiki/Dictaat/Dictaat_DIS10%
\iftoggle{ebook}{_ebook}{}.pdf}{4} van het DIS10 dictaat.
```

Levert de volgende uitvoer: Bla, bla, zie [hoofdstuk 4](#) van het DIS10 dictaat.

Dit commando kan ook handig gebruikt worden als optioneel argument van het `\cite` commando:

```
Bla, bla namespaces, zie \cite[\pdfdeelfref{section}{paragraaf~}{%
https://bytebucket.org/HR_ELEKTRO/cppprog/wiki/Dictaat_OPinCpp%
\iftoggle{ebook}{_ebook}{}.pdf}{15.4}]{Broeders2019}.
```

Bovenstaande code geeft: Bla, bla namespaces, zie [[2, paragraaf 15.4](#)].

Als de URL ook in de biblatex informatie is opgenomen dan is het eenvoudiger om het command `\citepdfdeelf` te gebruiken. Dit commando heeft dezelfde parameters als het

`\pdfdeelref` commando behalve dat als derde parameter de biblatex key in plaats van de URL wordt opgegeven.

Dus de code:

```
Bla, bla namespaces, zie ↩
↩ \citepdfdeel{section}{paragraaf~}{Broeders2019}{15.4}.
```

Levert de volgende uitvoer³⁵: Bla, bla namespaces, zie [2, paragraaf 15.4].

Het commando `\pdfpageref` is vergelijkbaar met `\pdfdeelref` maar verwijst naar een pagina.

De code:

```
Bla, bla, zie \pdfpageref[14]{%
https://ftp.snt.utwente.nl/pub/software/tex/info/lshort/ ↩
↩ english/lshort.pdf}%
{85} van \emph{The Not So Short Introduction to \LaTeXe}.
```

Levert de volgende uitvoer: Bla, bla, zie pagina 85 van *The Not So Short Introduction to L^AT_EX₂_ε*.

Als eerste (optionele) argument kan het verschil tussen het paginanummer dat wordt weergegeven in de pdf-viewer en het paginanummer weergegeven op de pagina zelf, worden opgegeven. Dit is nodig omdat bij veel documenten de paginanummering niet op de eerste pagina begint. Als tweede argument wordt de URL van het betreffende pdf-document opgegeven en als derde argument wordt het nummer dat wordt weergegeven op de pagina zelf opgegeven. In de bovenstaande code wordt verwezen de pagina met paginanummer 85 maar dat is in werkelijkheid de 99^{ste} pagina in het document vandaar dat als eerste parameter $99 - 85 = 14$ wordt meegegeven.

Het commando `\citepdfpag` is vergelijkbaar met `\citepdfdeel` maar verwijst naar een pagina. Het commando `\citepdfpag` heeft dezelfde parameters als `\pdfpageref`, behalve dat als tweede parameter de biblatex key in plaats van de URL wordt opgegeven.

De code `\citepdfpag[14]{lshort}{85}` levert als uitvoer: [4, pagina 85], een verwijzing naar pagina 85 van bron [4].

³⁵ Als het compileren van deze code fouten oplevert, dan moet je eerst het commando `bibber` uitvoeren.

2.14 Opdrachten

Veel dictaten en studiehandleidingen zullen opdrachten bevatten. Traditioneel bevinden de bij een hoofdstuk behorende opdrachten zich in een aparte paragraaf aan het einde van een hoofdstuk. De lezer zal waarschijnlijk meer geactiveerd worden als de opdrachten in de tekst van het hoofdstuk worden opgenomen. Het is dan handig om elke opdracht in een kader te plaatsen. Je kunt hiervoor de environment `opdracht` gebruiken. Deze environment heeft één parameter, namelijk de titel van de opdracht. De opdrachten worden automatisch genummerd. Als je in een `opdracht`-environment een label gebruikt, dan kun je m.b.v. een `\cref`-commando naar de betreffende opdracht verwijzen.

Opdracht 1: De opdracht-environment

Dit is een voorbeeld van een opdracht die in een kader geplaatst is door gebruik te maken van het in de stijl gedefinieerde `opdracht`-environment.

- a) Geef de \LaTeX -code die nodig om dit opdrachtkader te maken.
- b) Geef de \LaTeX -code die nodig om vanuit de tekst naar deze opdracht te verwijzen.

De code waarmee het bovenstaande opdrachtkader is gemaakt is gegeven in [listing 2.15](#). Je kunt vanuit de tekst verwijzen naar deze opdracht met de code `\cref{opdr:opdracht}`. Deze code produceert de volgende tekst: [opdracht 1](#).

```
\begin{opdracht}{De opdracht-environment}
  \label{opdr:opdracht}
  Dit is een voorbeeld van een opdracht die in een kader ↵
  ↵ geplaatst is door gebruik te maken van het in de stijl ↵
  ↵ gedefinieerde |opdracht|\hyp{}environment.
  \begin{enumerate}[label=\alph*]
    \item Geef de \LaTeX\hyp{}code die nodig om dit ↵
    ↵ opdrachtkader te maken.
    \item Geef de \LaTeX\hyp{}code die nodig om vanuit de ↵
    ↵ tekst naar deze opdracht te verwijzen.
  \end{enumerate}
\end{opdracht}
```

Listing 2.15: \LaTeX -code waarmee [opdracht 1](#) gemaakt is.

3

English Style

When you want to write a document in English using this style, you have to use the file `style_en_US.tex` instead of `style_nl_NL`.

In this chapter, only the differences between these styles will be discussed. It is assumed that the reader is already familiar with the Dutch version of this style.

The `\USE`-command which is used to specify English phrases in a Dutch document is not included in the `en_US` style for an obvious reason.

3.1 Titles of Chapters and Sections

The principal words in the titles and subtitles of a document should be capitalized.

3.2 License

The English version of the license is generated by the following \LaTeX -code:

```
\vspace*{\stretch{20}}
\includegraphics{figs/by-nc-sa_eu.pdf}
\par
\documenttype{} \titel{} from Rotterdam University of Applied
↔ Sciences is licensed by a ↔
↔ \href{https://creativecommons.org/licenses/by-nc-sa/ ↔
↔ 3.0/nl/deed.en}{Creative Commons ↔
```

↔ Attribution\hyp{}NonCommercial\hyp{}ShareAlike 3.0 Netherlands ↔
↔ license}.

3.3 References

When a reference to, for example, a figure is made by using `\cref{label}`, the word “Figure” is capitalized. For example, the code `\cref{tab:zomaar}` gives: [Table 2.1](#). If you do not want this, remove the option `capitalise` for the `cleverref` package. You also have to change the `\crefname` commands and you have to provide proper `\Crefname` commands.

3.4 Quotes

The use of quotation marks in US English is very different than it is in Dutch. As stated on https://en.wikipedia.org/wiki/Quotation_mark: “In American English, double quotes are used normally. If quote marks are used inside another pair of quote marks, then single quotes are used.” Therefore, the commands `\squote` and `\dquote` are not needed because you can use the commands `\enquote` and `\enquote*` which are defined in the package `csquotes`.

An example of a document which is using the `style_en_US` is *Training Digital Signal Processing Work Lab Handbook* [6].

4

Listings en code-fragmenten

De commando's die gebruikt kunnen worden voor listings en codefragmenten zijn in een apart bestand geplaatst: `styleCode.tex`. Er wordt gebruikt gemaakt van de package `listings`³⁶.

Er worden een aantal zogenoemde 'styles' gedefinieerd die in de onderstaande commando's gebruikt kunnen worden: `linenumbers` voegt regelnummers toe, `cstyle` maakt de code op als C-code, `cplusplusstyle` als C++-code, `lstyle` als L^AT_EX-code, `pstyle` als Python-code, `mstyle` als MATLAB-code, `astyle` AVR-assembler-code, `amstyle` als MSP430-assembler-code, `legstyle` als LEG-assembler-code³⁷ en `pinkystyle` als Pinky-assembler-code³⁸.

4.1 Listings

De commando's `\floatlstinput` en `\lstinput` lezen code uit (een deel van) een bestand. Deze commando's hebben vier parameters waarvan de eerste optioneel is. Als eerste argument kunnen extra `key=value` paren voor het commando `\lstinputlisting` worden meegegeven, bijvoorbeeld: `style = cstyle`, `style = linenumbers`, `linerange = {1-6, 13-27}`. Als tweede argument wordt de filenaam opgegeven. Deze file moet in het directory `./progs` staan. De derde parameter is een deel van het label, de labelnaam wordt `lst:derde-argument`. De laatste parameter is de titel van de listing. Het commando `\floatlstinput` heeft

³⁶ <https://www.ctan.org/pkg/listings>

³⁷ LEG-assembler-code wordt gebruikt in het boek *Computer Organisation and Design, The Hardware/Software Interface ARM Edition* [5].

³⁸ Pinky-assembler-code is een subset van de ARMv7 Thumb instructieset en is gedefinieerd in [1].

de voorkeur omdat \LaTeX dan zelf kan bepalen waar de listing wordt geplaatst en de listing dan niet halverwege afgebroken wordt.

De environments `floatlst` en `lst` kunnen gebruikt worden als de code niet uit een bestand moet worden gelezen, maar in de environment zelf wordt opgenomen. Deze environments hebben drie parameters waarvan de eerste optioneel is. Als eerste argument kunnen extra `key=value` paren voor het commando `\lstset` worden meegegeven, bijvoorbeeld: `style = cstyle`, `style = linenumbers`. De tweede parameter is een deel van het label, de labelnaam wordt `lst:tweedeargument`. De laatste parameter is de titel van de listing. De environment `floatlst` heeft de voorkeur omdat \LaTeX dan zelf kan bepalen waar de listing wordt geplaatst en de listing dan niet halverwege afgebroken wordt.

Het commando `\lstinputfragment` kan gebruikt worden als je een codefragment uit een bestand wilt inlezen (zonder titel en zonder label). Het fragment wordt op de plaats van het commando ingevuld. Dit commando heeft twee parameters die gelijk zijn aan de eerste twee parameters van het commando `floatlstinput`.

De environment `lstfragment` kan gebruikt worden als een codefragment niet uit een bestand moet worden gelezen, maar in de environment zelf wordt opgenomen (zonder titel en zonder label). Het fragment wordt op de plaats van het commando ingevuld. Dit commando heeft één parameters die gelijk is aan de eerste parameter van de environment `floatlst`.

4.2 \LaTeX -code in listings

In een listing kun je ook \LaTeX -commando's gebruiken. Deze moeten dan wel omringd zijn door zogenoemde escape-karakters. Je moet zelf definiëren welke escape-karakters gebruikt worden m.b.v. de optie:

```
escapeinside={BEGIN-ESCAPE-KARAKTERS}{END-ESCAPE-KARAKTERS}.
```

Je kunt op deze manier tekst in een bepaalde stijl afdrukken of een voetnoot in je listing gebruiken, zie [listing 4.1](#).

```
\begin{lstfragment}[style=cstyle,escapeinside={(*}{*)}] {
  // Dit is een (*\textcolor{hrred}{belangrijk}*) voorbeeld.
  int main() {
    return 0(*\footnotemark*);
  }
}\end{lstfragment}
```

`\footnotetext`{Sinds C99 mag dit `\ccode{return}`-statement ook ↔
 ↔ weggelaten worden. De functie `\ccode{main}` geeft dan toch ↔
 ↔ `\ccode{0}` terug.}

Listing 4.1: Een voorbeeld van het gebruik van \LaTeX -code in een listing.

Dit levert de volgende output:

```
// Dit is een belangrijk voorbeeld.
int main() {
    return 039;
}
```

4.3 Code-segmenten

Inline code kan gedefinieerd worden met de commando's: `\code` voor taalafhankelijke code, `\ccode` voor C-code, `\cppcode` voor C++-code, `\lcode` voor \LaTeX -code, `\pycode` voor Python-code, `\mcode` voor MATLAB-code, `\acode` voor AVR-assembler-code, `\amcode` voor MSP430-assembler-code, `\legcode` voor LEG-assembler-code of `\pinkycode` voor Pinky-assembler-code. Deze commando's hebben twee parameters waarvan de eerste optioneel is. Als eerste argument kunnen extra `key=value` paren voor het commando `\lstinline` worden meegegeven. De tweede parameter is de code die inline moet worden geplaatst. Het karakter `@` in het verplichte argument van het commando `\code` moet voorafgegaan worden door een `\`.

Het commando `\hcode` is een uitbreiding van het commando `\code`. In de inline code kunnen mogelijke afbreeklocaties worden aangegeven met de tekencombinatie `\\-`. Dit is nodig als een stuk inline code erg lange identifiers bevat. Er zijn geen programmeertaal specifieke versies van gemaakt. Dus als je de identifier `ditIsEenLangeFunctienaam` wilt laten afbreken in inline C-code dan kan dit als volgt in \LaTeX -code worden geschreven:

```
\hcode[style=cstyle]{return ↔
↔ dit\\-Is\\-Een\\-Lange\\-Functie\\-naam();}
```

Deze code levert het volgende, netjes afgebroken, resultaat: `return ditIsEenLangeFunctienaam();`.

Code kan nog eenvoudiger in lopende tekst worden opgenomen in één van de environments: `clstShortInline`, `cpplstShortInline`, `llstShortInline`, `pylstShortInline`, `mlstShort-`

³⁹ Sinds C99 mag dit `return`-statement ook weggelaten worden. De functie `main` geeft dan toch `0` terug.

`Inline`, `alstShortInline`, `amlstShortInline`, `leglstShortInline` of `pinkylstShortInline`. Als in één van deze environments de syntax `|code|` wordt gebruikt, dan wordt deze code inline geplaatst. Maar pas op: dit werkt niet in voetnoten, titels enz.

4.4 Console in- en output

De environment `coutput` kan gebruikt worden om console output weer te geven. Binnen deze environment kan het commando `\cinput` gebruikt worden om console input weer te geven. Console input wordt donkergroen en onderstreept weergegeven. De console in- en output van een programma dat de som van twee integers bepaald, kun je bijvoorbeeld, als volgt in \LaTeX coderen:

```
\begin{coutput}
Geef een geheel getal: \cinput{13}
Geef nog een geheel getal: \cinput{7}
13 + 7 = 20
\end{coutput}
```

Bovenstaande \LaTeX -code geeft de volgende uitvoer:

```
Geef een geheel getal: 13
Geef nog een geheel getal: 7
13 + 7 = 20
```

4.5 Overige commando's die handig zijn bij documenten met programmeercode

Het commando `\Cpp` kan gebruikt worden om de naam van de programmeertaal C++ netjes af te drukken. Vergelijk C++ met C++. De \LaTeX -code van de vorige zin is: Vergelijk C++ met `\Cpp{}`.

Het commando `\proglink` kan gebruikt worden om een link te creëren naar de sourcecode van een programma op internet. De URL van het directory waar de bij het betreffende document behorende programma's zich bevinden moet zijn gespecificeerd door het commando `\progurl` te definiëren. In deze handleiding is dat als volgt gedaan:

```
\newcommand{\progsurl}{
  https://bitbucket.org/HR_ELEKTRO/latex-stijl/src/%
  master/Handleiding/progs}
```

De code `\proglink{lpfbodelog.m}` levert nu de volgende link op: [lpfbodelog.m](#).

5

Algemene \LaTeX -tips en -trucs

Nog enkele algemene \LaTeX -tips en -trucs:

- Latex kent minstens vier verschillende liggende streepjes, zie [tabel 5.1](#). De laatste moet je als rekenkundig minteken gebruiken.

Tabel 5.1: Liggende streepjes in \LaTeX

Engelse naam ⁴⁰	code	resultaat
hyphen	-	-
en dash	--	-
em dash	---	—
minus sign	$\$-\$$	—

- `~` geeft een “harde” spatie. De regel wordt niet afgebroken bij deze harde spatie. Voorbeelden:
 - `lengte~l`. Het is een raar gezicht als de regel na “lengte” wordt afgebroken. Het gebruik van de harde spatie voorkomt dit. De lengte l wordt opgegeven in m.
 - `1, 2 of~3`. De barman vroeg nogmaals: “Hoeveel biertjes wil je? 1, 2 of 3?”.
 - Verhoog de `$teller$` met~1 modulo~2. Verhoog de `teller` met 1 modulo 2.

⁴⁰ Zie: <https://en.wikipedia.org/wiki/Dash>.

- `\`, geeft een halve spatie. Bij gebruik van en-dash in ranges is bijvoorbeeld een beetje ruimte links en rechts nodig. Vergelijk pagina 1--7 met pagina 1\, --\, 7. Vergelijk pagina 1—7 met pagina 1 -- 7.
- Er is een verschil tussen `\textbf{tekst1}` en `{\bfseries tekst2}`. In `tekst2` mag een paragraafeinde voorkomen, in `tekst1` niet.
- `package menukeys`⁴¹ is handig als je menu-opties en toetsaanslagen wilt benadrukken in tutorials.
- Kolomkoppen behoren gecentreerd te zijn. De tabelcommando's in de stijl doen dat automatisch.
- Als je in math-mode een regelmatige rij met getallen wilt opgeven, dan kan het commando `\ldots` van pas komen: 2, 4, 6, ..., 100. Als je een regelmatige som van getallen wilt opgeven, dan geeft het commando `\cdots` een mooier resultaat: 2 + 4 + 6 + ... + 100.
- Gebruik `\ensuremath` om eigen math-commando's in text- en math-mode te kunnen gebruiken
- Bibtex:
 - Gebruik `and` tussen namen van verschillende auteurs.
 - Gebruik accolades rond dubbele achternamen.
 - Neem `Langid = {...}` op in elke entry zodat er correct wordt afgebroken.
 - Gebruik `--` bij het aangeven van een aangesloten aantal pagina's: `Pages = {293--326}`.
 - Geef ISBN-nummers op met streepjes ertussen: `ISBN = {978-0-201-89684-8}` om een overfull hbox te voorkomen. Gebruik: <http://tools.wmflabs.org/isbn/IsbnCheckAndFormat> om ISBN-nummers zonder streepjes om te zetten in de officiële notatie (met streepjes).
- `\sloppy` zorgt ervoor dat overfull hbox vermeden wordt door spaties ongelimiteerd lang te maken. Gebruik daarna `\fussy` om weer netjes uit te vullen. In de ebook versie en ook in de papieren versie geeft de url bij het bovenstaande puntje een overfull hbox. Kijk wat er met `\sloppy` gebeurt:
 - Gebruik: <http://tools.wmflabs.org/isbn/IsbnCheckAndFormat> om ISBN nummers zonder streepjes om te zetten in de officiële notatie (met steepjes).

⁴¹ <https://www.ctan.org/pkg/menukeys>

- Gebruik `\` tussen een kleine letter en een punt die niet het einde van een zin is. B.v. bij enz. en andere afko's. De \LaTeX -code van de vorige zin is: B.v.\ `bij enz.\ en andere afko's.`
- Gebruik `@` tussen een hoofdletter en een punt, vraagteken of uitroepteken aan het einde van een zin (om einde zin spatie te krijgen). Ik gebruik geen vitamine C. Gebruik jij wel vitamine C? Of ook niet. De \LaTeX -code van de vorige drie zinnen is: Ik gebruik geen vitamine~C\`@`. Gebruik jij wel vitamine~C\`@`? Of ook niet.
- Een moving argument is een argument dat ook op een andere plaats dan waar het gedefinieerd wordt, wordt gebruikt. B.v. het argument van `\section` wordt ook in de inhoudsopgave gebruikt. Fragile commands mogen niet in moving argumenten gebruikt worden. Als je toch een fragile command als moving argument wil gebruiken, dan moet je er `\protect` voorzetten. Bijvoorbeeld als je een `\footnote` in een `\section` wilt opnemen:

```
\hcode{\section{\texorpdfstring{Titel\protect \footnote{Bedenk ←
↪ zelf een goede titel}}}{Titel}}
```

5.1 Titel⁴²

- `\texorpdfstring` is handig als je \LaTeX -code in een titel wilt gebruiken. De titel komt namelijk ook in de inhoudsopgave van het pdf document maar daar mag geen \LaTeX -code in gebruikt worden. Bijvoorbeeld:

```
section{\texorpdfstring{$E=mc^2$}{E=mc2}}
```

5.2 $E = mc^2$

- `\centering` start *geen* nieuwe paragraaf `\begin{center}... \end{center}` wel.
- Gebruik liever geen absolute lengtes zoals cm, mm en pt maar relatieve zoals em (breedte van een M) of ex (hoogte van een x).
- Het eerste woord in een zin wordt nooit afgebroken, dat kan wel eens problemen geven in een smalle kolom in een tabel. Zet er `\hspace{0pt}` voor om het toch af te laten breken.
- Gebruik een `\mbox` om ongewenst afbreken te voorkomen.

⁴² Bedenk zelf een goede titel

- Bij het maken van een toets met behulp van de class exam kun je een `\solution` op een nieuwe regel beginnen met `\ifprintanswers\\\fi`.
- Soms wordt een extra lege regel toegevoegd aan het einde van een paragraaf als de laatste regel van de paragraaf exact past binnen de tekstbreedte. Dit kun je oplossen door `\vskip 0pt` achter de laatste regel van de paragraaf te plaatsen.
- `#`, `$`, `%`, `~`, `_`, `^`, `\`, `{` en `}` zijn speciale karakters en moeten als volgt ingevoerd worden: `\#`, `\$`, `\%`, `\textasciitilde`, `_`, `\textasciicircum`, `\textbackslash`, `\{` en `\}`.
- In lopende tekst kun je het commando `\textcite` gebruiken om de naam van de auteur met de verwijzing te genereren. De code `\textcite{latex}` beweert `\ldots{}` levert de volgende tekst op: Lammport [3] beweert
- Een leesteken tussen **vet**, normaal, *cursief* moet in de lichtste font worden opgemaakt.
- Gebruik smallcaps voor afkortingen zoals `\textsc{html}`: HTML. Zo ziet IP-adres er mooier uit dan IP-adres.
- Het woord finish wordt door \LaTeX opgemaakt met een zogenaamde ligatuur⁴³. In woorden waarbij de f en de i niet bij dezelfde lettergreep horen, zoals bijvoorbeeld selfish is het beter om geen ligatuur te gebruiken. Gebruik daarom `self\makebox{}ish` selfish. Vergelijk selfish met selfish⁴⁴. Hetzelfde geldt voor woorden zoals offline enzovoorts.
- Gebruik de package upgreek voor rechtopstaande Griekse letters.

Nog enkele tips voor het gebruik van \LaTeX in combinatie met git:

- Als de \LaTeX -code in een VCS wordt opgenomen worden alle wijzigingen en contributies netjes bijgehouden. Binnen de opleiding Elektrotechniek van de Hogeschool Rotterdam maken we gebruik van BitBucket (zie: https://bitbucket.org/HR_ELEKTRO/). We maken gebruik van BitBucket en niet van GitHub omdat de eerste gratis private repositories biedt.
- Het is handig om na elke zin een enter in te typen. Git vergelijkt bestanden namelijk regel voor regel en wijzigingen worden dus veel duidelijker weergegeven als de regels niet te lang zijn.

⁴³ Zie: [https://nl.wikipedia.org/wiki/Ligatuur_\(typografie\)](https://nl.wikipedia.org/wiki/Ligatuur_(typografie)).

⁴⁴ Je moet wel inzoomen om het verschil te kunnen zien. In selfish zit de f vast aan de i en bij het selecteren van tekst is de fi één letter. In selfish zit een beetje ruimte tussen de f en de i en bij het selecteren van tekst zijn het twee aparte letters.

Bibliografie

- [1] Harry Broeders. *De LEGv7 architectuur en de Pinky instructieset*. 1.1. 2022. URL: https://bytebucket.org/HR_ELEKTRO/rts10/wiki/LEGv7/LEGv7-Pinky_ebook.pdf (geciteerd op p. 40).
- [2] Harry Broeders. *Objectgeoriënteerd Programmeren in C++*. 1.2b. 2020. URL: https://bytebucket.org/HR_ELEKTRO/cppprog/wiki/Dictaat_OPinCpp_ebook.pdf (geciteerd op pp. 7, 35, 36).
- [3] Leslie Lamport. *L^AT_EX A Document Preparation System: User's Guide and Reference Manual*. 2de ed. Addison Wesley, 1994. ISBN: 978-81-7758-414-1 (geciteerd op pp. 8, 48).
- [4] Tobias Oetiker e.a. *The Not So Short Introduction to L^AT_EX 2_ε*. 2015. URL: <http://ftp.snt.utwente.nl/pub/software/tex/info/lshort/english/lshort.pdf> (geciteerd op pp. 8, 36).
- [5] David A. Patterson en John L. Hennessy. *Computer Organisation and Design, The Hardware/Software Interface ARM Edition*. 1ste ed. Morgan Kaufmann, 2016. ISBN: 978-0-12-801733-3 (geciteerd op p. 40).
- [6] Johan Peltenburg en Harry Broeders. *Training Digital Signal Processing Work Lab Handbook*. 2015. URL: <https://harrybroeders.bitbucket.io/TDS02/index.htm#handbook> (geciteerd op pp. 7, 39).

A

Voorbeelden van code-fragmenten

Hier is een stukje C code:

```
int main() {  
    printf("Hello LaTeX");  
    return 0;  
}
```

De functie `main`⁴⁵ returned de waarde `0` met een `return`-statement. Het woord `class` is geen keyword in C.

Hier is een stukje C++ code:

```
// voorbeeld van C++ code  
int main() {  
    std::cout << "Hello LaTeX" << std::endl;  
    return 0;  
}
```

De functie `main`⁴⁶ returned de waarde `0` met een `return`-statement. Het woord `class` is een keyword in C++.

Hier is een stukje MATLAB-code:

⁴⁵ In een C-programma wordt de functie `main` als eerste aangeroepen.

⁴⁶ Ook in een C++-programma wordt de functie `main` als eerste aangeroepen.

```
% voorbeeld van MATLAB code
b = 0.25 * sinc(0.25 * [-10:10]);
[H,f] = freqz(b, 1, 512, 8000);
HdB = 20 * log10(abs(H));
plot(f, HdB, 'Color', [204/255 0 51/255]);
grid;
ylabel('Magnitude (dB)');
ylim([-45 5]);
xlabel('Frequency (Hz)');
```

B

Voorbeelden van floats

Listing B.1 is een door MATLAB gegenereerde headerfile. De listing is floating en kan dus ergens anders geplaatst worden dan meteen onder deze tekst.

```

1 const Int16 BL = 21;
2 const Int16 B[21] = {
3     1043, 819, 0, -1054, -1738, -1475, 0, 2458,
4     5215, 7375, 8192, 7375, 5215, 2458, 0, -1475,
5     -1738, -1054, 0, 819, 1043
6 }; // -----

```

Listing B.1: 16-bit signed integer coefficients generated by MATLAB.

Figuur B.1 laat een mooi plaatje zien als een floating figuur en kan dus ergens anders geplaatst worden dan meteen onder deze tekst.

Tabel B.1 is een voorbeeld van een simpele floating tabel en kan dus ergens anders geplaatst worden dan meteen onder deze tekst.

Tabel B.1: Een eenvoudige vertaaltabel.

Engels	Nederlands
biscuit	kaakje
cake	cake
holiday	vakantie



Figuur B.1: Nulleen en enen — het mooiste wat er is.

Deze tekst komt (misschien) onder de tabel zodat de afstand tussen de tabel en de tekst zichtbaar wordt.

Listing B.2 is een eenvoudig AVR-assembler programma. De listing is floating en kan dus ergens anders geplaatst worden dan meteen onder deze tekst.

```

1 ; ledcopy.asm - Simply copies switches to LEDs
2 ;
3 .include "m32def.inc"
4
5     LDI     R16,0x00
6     OUT     DDRA,R16           ; configure Port A as 8 inputs
7     LDI     R16,0xFF
8     OUT     DDRB,R16           ; configure Port B as 8 outputs
9
10 LOOP: IN     R0,PINA           ; read Port A (switches)
11     OUT     PORTB,R0           ; output to Port B (leds)
12     RJMP    LOOP              ; and again

```

Listing B.2: Kopiëren van inputs naar outputs in AVR-assembler.

C

Voorbeelden van Here floats

Here floats zijn helemaal geen floats want ze worden op de plaats gezet waar ze zijn gedefinieerd. Dit kan ongewenste witruimte aan het einde van een pagina tot gevolg hebben. Het verdient dan ook de voorkeur om hier alleen in uiterste nood gebruik van te maken. \LaTeX weet vaak beter waar figuren, tabellen en listings geplaatst moeten worden om een mooie bladspiegel te creëren. [Listing C.1](#) is een door MATLAB gegenereerde headerfile. Deze listing komt gegarandeerd meteen na deze tekst.

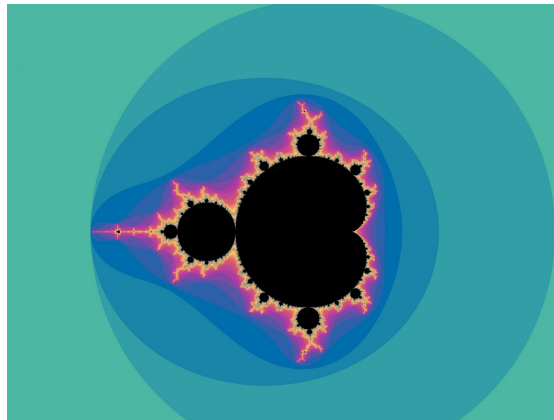
```

1 const Int16 BL = 21;
2 const Int16 B[21] = {
3     1043, 819, 0, -1054, -1738, -1475, 0, 2458,
4     5215, 7375, 8192, 7375, 5215, 2458, 0, -1475,
5     -1738, -1054, 0, 819, 1043
6 }; // -----

```

Listing C.1: 16-bit signed integer coefficients generated by MATLAB.

[Figuur C.1](#) laat een mooi figuur zien en komt gegarandeerd meteen na deze tekst.



Figuur C.1: Fractals — ook mooi.

Tabel C.1 is een voorbeeld van een simpele tabel en komt gegarandeerd meteen na deze tekst.

Tabel C.1: Een eenvoudige vertaaltabel.

Engels	Nederlands
biscuit	kaakje
cake	cake
holiday	vakantie

Deze tekst komt onder de tabel zodat de afstand tussen de tabel en de tekst zichtbaar wordt.