

Nybörjarguide i L^AT_EX, andra upplagan.

Att T_EXa: en praktisk guide

Simon Sigurdhsson



Licenserad under Creative Commons By-Sa 4.o.

<http://creativecommons.org/licenses/by-sa/4.0/>

Den här boken är en inkomplett guide till att skriva och typsätta L^AT_EX-dokument riktad till studenter på Chalmers Tekniska Högskola, specifikt programmen Teknisk Matematik och Teknisk Fysik. Inspiration har tagits från bland annat von Schultz (2005) och Voß (2010), men främst från Oetiker m. fl. (2011).

Boken är typsatt med X_YL^AT_EX, och designen är baserad på dokumentklassen scrbook, som finns tillgänglig på CTAN som en del av KOMA-Script. Typsnitten som används är *Linux Libertine*, *Linux Biolinum* och *Bitstream Vera Sans Mono*. Dessa typsnitt finns tillgängliga på <http://www.linuxlibertine.org> och <https://www-old.gnome.org/fonts/>, respektive.

©2014 SIMON SIGURDHSSON.

Detta verk är licensierat under *Creative Commons Erkännande-DelaLika 4.0*. För att ta del av en kopia av licensen, besök <http://creativecommons.org/licenses/by-sa/4.0/deed.sv> eller skicka ett brev till *Creative Commons*, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA. Källkoden till boken finns tillgänglig på <http://github.com/urdh/latexbok/>.

Göteborg, 2014.

Tack till

Phaddergrupp 255,
som korrekturläst och
hjälpt till att förbättra boken

och

Christian von Schultz,
Tobias Oetiker och Herbert Voß
för inspirerande texter på samma tema.

Förord

Denna bok, vars första upplaga publicerades 2011, är i grunden ett försök både översätta och förbättra *The Not So Short Introduction to L^AT_EX 2_ε* (Oetiker m. fl. 2011), och därmed få en introduktion till L^AT_EX som är både modern och skriven på svenska. I det tidigaste skedet var tanken att boken skulle kunna användas som en inofficiell eller informell kursbok eller referens i datointroduktionen som ges till förstaårsstudenter i Teknisk Fysik och Teknisk Matematik vid Chalmers Tekniska Högskola, som innehåller ett antal föreläsningar om L^AT_EX. Glädjande nog verkar boken fylla denna roll, och den har dessutom använts som referenslitteratur i motsvarande kurs på Maskinteknik vid Chalmers.

I likhet med många andra IT-system utvecklas L^AT_EX ständigt, och därför är en introduktion som denna inte aktuell hur länge som helst. Först upplagan av denna bok är förvisso fortfarande någorlunda korrekt i de flesta avseenden — speciellt de tre första kapitlen som behandlar de absoluta grunderna — men L^AT_EX (och min kunskap om detsamma, vilket illustrerar hur mycket man kan lära genom att bara använda L^AT_EX när tillfälle ges) har utvecklats mycket under de tre år som gått sedan boken först publicerades på internet. Därför började jag under 2013 fila på förbättringar, både stora och små, av en del avsnitt i boken. De största förändringarna har skett i kapitel 4, som i förra upplagan var mycket kort men som numera behandlar PGF/TikZ och graphicx mer utförligt och även diskuterar hur man allmänt skapar bra grafik, och i kapitel 5 som nu behandlar den moderna ersättaren av B_BT_EX, bibl_{at}ex.

Utöver dessa stora förändringar har ett antal små uppdateringar gjorts i andra kapitel, både för att förtydliga sådant som var otydligt i förra upplagan, för att uppdatera sådant som blivit utdaterat (detta gäller inte minst

Förord

listan över paket i kapitel 6) och för att göra boken mer tillämpbar för en bredare publik — även om första upplagan till stor del kunde användas även utanför den tänkta målgruppen bör denna upplaga vara än mer allmän.

L^AT_EX utvecklas som sagt med rasande fart, och delar av boken (framför allt kapitel 6) kommer säkerligen vara utdaterade inom ett par år — förhoppningsvis blir det då en ny upplaga av boken. Tills dess välkomnas givetvis rapporter om alla sorters errata, förbättringsmöjligheter eller liknande, som med fördel kan rapporteras som ärenden på Github.¹

Simon Sigurdhsson
Göteborg, 3 januari 2014

¹ <https://github.com/urdh/latexbok>

Innehåll

Inledning	1
Vad är L ^A T _E X?	1
Varför L ^A T _E X?	2
I denna introduktion	2
 I Grunderna	 5
1 Grundläggande begrepp	6
1.1 Snabbkurs i L ^A T _E X-kod	6
1.2 Dokumentets layout	10
1.3 Från T _E X till PDF	14
1.4 Filer du kanske stöter på	16
 2 Typsättning med L^AT_EX	 18
2.1 Text- och språkstruktur	18
2.2 Figurer, tabeller, listor	26
2.3 Typografiska betänkligheter	35
2.4 Andra viktiga delar	39

Innehåll

II	Matematiken	44
3	Matematik med \LaTeX och \mathcal{AMS}	45
3.1	Att visa ekvationer	45
3.2	En snabbkurs i \LaTeX -matematik	47
3.3	Ekvationsomgivningar	60
3.4	Mellanrum	66
3.5	Punkter	68
3.6	Enheter med siunitx	68
III	Figurerna	71
4	Grafik med \LaTeX	72
4.1	Vad är bra grafik?	72
4.2	Inkludera grafik	76
4.3	Exportera grafik eller data	79
IV	Referenserna	86
5	Referenser med biblatex	87
5.1	Referensdatabaser	87
5.2	Stilar och inställningar i biblatex	90
5.3	Referenskommandon och referenslistor	94

Innehåll

V	Fortsättningen	96
6	Vidare läsning	97
6.1	Andra resurser	97
6.2	Tips för stora projekt	100
6.3	Rekommenderade paket	102
6.4	Andra T _E X-baserade projekt	111
	Litteraturförteckning	115
	Sakregister	122
	Förteckning över L^AT_EX-paket	126
	Förteckning över kommandon och miljöer	128
VI	Appendix	133
A	En enkel mall	134

Inledning

Att publicera något, oavsett om det är långa böcker eller korta artiklar, är inte helt enkelt. Förr, innan det fanns en dator i varje hem och innan programvaror som Microsoft Word och OpenOffice blev tillgängliga för var man så var publicering något komplext, något som utfördes av många olika personer. Arbetet delades upp i olika bitar och en expert inom varje område fick sköta den biten; författande, typsättning, design och så vidare.

Moderna programvaror (ordbehandlare, främst) fungerar på ett helt annat sätt. De ger all makt till författaren, som kanske varken vill eller bör (författaren är ju inte typsättare) ha makt över typografin. Som en konsekvens blir typsättningen av dokument inte alltid bra — användarna är under illusionen att ett estetiskt tilltalande dokument även är ett bra dokument rent typografiskt. Så är givetvis inte fallet.

För att få snygga dokument måste man återskapa den gamla ordningen där varje uppgift löses av någon som är bra på det. En författare ska inte behöva berätta *hur* saker ska se ut — det ska designern göra — författaren ska bara berätta *vad* saker är.¹ L^AT_EX låter dig som författare göra just detta.

Vad är L^AT_EX?

L^AT_EX (uttalas [la'tɛʃ]) är i princip en påbyggnad till T_EX, ett typsättnings-system. Man kan se det som att L^AT_EX är designern, och T_EX är typsättaren.

¹ Detta är inte helt olikt den uppdelning som görs mellan HTML och CSS.

Varför \LaTeX ?

\LaTeX berättar i någon mening för \TeX hur den tror att du vill ha ditt dokument (utifrån dokumentklassen samt den kod du skriver) och låter sedan \TeX typsätta detta för att skapa ett ”tryckt” dokument.

Men \LaTeX är inte bara ett typsättningssystem; både språket dokumenten skrivs i och den kompilator som omvandlar dokumenten till DVI-filer kallas \LaTeX (men just kompilatorn brukar benämnas `latex`, eftersom det är så den körs från terminalen). Det finns även modernare versioner av \LaTeX -kompilatorn, till exempel $\text{pdf}\LaTeX$ som skapar PDF-dokument direkt (så att man slipper konvertera dem med `dvipdf`) och $\text{Xe}\LaTeX$, som baseras helt på `UNICODE` och har bättre stöd för diverse typsnitt.

Varför \LaTeX ?

Fördelen med \LaTeX är att författaren endast behöver lära sig några enkla kommandon — sådana som gör texten kursiv eller infogar en figur — men att dokumentet ändå håller mycket hög standard, precis som om det typsatts av en riktig typsättare.

Eftersom språket man skriver dokument i uppmanar till struktur och ordning blir det dessutom lättare att skriva strukturerade texter. Även om saker som finns i en vanlig ordbehandlare (stavningskontroll, till exempel) saknas i själva språket så kan (och bör) dessa tillhandahållas av externa program, vilket gör att \LaTeX kan koncentreras på att vara bra på *en* sak: att typsätta dokument. Saker som är komplicerade eller omöjliga att göra i en vanlig ordbehandlare, till exempel korsreferenser, figurer, tabeller, fotnoter, referenslistor och dylikt är mycket enkla att göra i \LaTeX och oftast fullt automatiserade.

Dessutom har \LaTeX traditionellt använts av akademiker eftersom det gör typsättning av just matematik och fysik både enkelt och visuellt attraktivt.

I denna introduktion

Den här korta introduktionen kommer att visa dig hur man på ett enkelt sätt typsätter dokument med \LaTeX i vanliga tillämpningar. Dessutom kommer den framåt slutet peka på specifika paket eller resurser som kan vara användbara för mer avancerade fall.

Efter att ha läst den här introduktionen bör läsaren kunna skriva dokument och rapporter utan problem. Det är dock inte tänkt att denna introduktion ska vara en fullgod referens till \LaTeX ; för detta rekommenderas istället Lampport (1994) och Mittelbach, Goossens m. fl. (2004).

Introduktionen innehåller följande kapitel:

Kapitel 1, Grundläggande begrepp beskriver den grundläggande strukturen hos \LaTeX -dokument och hur det språk dokumenten skrivs i fungerar i korta drag. Efter denna del bör du veta på ett ungefär hur \LaTeX fungerar.

Kapitel 2, Typsättning med pdf \LaTeX beskriver i detalj hur man skriver ett vanligt \LaTeX -dokument, och förklarar några av de viktigaste miljöerna och kommandona som används. Efter denna del bör du kunna skriva enkla dokument med \LaTeX .

Kapitel 3, Matematik med \LaTeX och $\mathcal{A}\mathcal{M}\mathcal{S}$ beskriver hur man på bästa sätt använder \LaTeX tillsammans med $\mathcal{A}\mathcal{M}\mathcal{S}$ -paketen för att typsätta det \LaTeX typsätter bäst, matematik, och går även kort in på hur man typsätter en del fysik med paketet siunitx.

Kapitel 4, Grafik med \LaTeX beskriver hur man inkluderar grafik i \LaTeX med paketet graphicx, och visar några korta exempel på hur man kan rita direkt i \LaTeX med PGF/TikZ. Efter den här och föregående del bör du kunna skriva fullständiga rapporter med \LaTeX .

Kapitel 5, Referenser med Bib \TeX beskriver hur du använder Bib \TeX för att hålla koll på och använda referenser i \LaTeX , och beskriver även i korthet paketet chscite som hjälper dig att typsätta referenser på det sätt Chalmers bibliotek rekommenderar. Efter denna delen bör du kunna skriva långa arbeten (till exempel kandidatrapporter) i \LaTeX .

Kapitel 6, Vidare läsning tipsar om andra resurser, paket, dokumentklasser och rekommendationer som kan vara av nytta när du skriver långa (eller korta) rapporter. Kan vara en språngbräda om du vill göra något som inte förklaras i resten av introduktionen.

Appendix A, En enkel mall innehåller en mall du med fördel kan basera dina framtida L^AT_EX-dokument på. Den är fullt kommenterad och motiverar de paket som inkluderas och kommandon som definieras.

Det är viktigt att läsa dessa kapitel i rätt ordning; varje kapitel bygger på de föregående, och de är ju trots allt inte särskilt långa. Se till att studera och förstå de exempel som presenteras, och lek gärna lite själv om du inte riktigt förstår. Det finns inget bättre sätt att lära sig än att vara nyfiken!

L^AT_EX finns till många plattformar, och finns installerat på Chalmers Linux-datorer. Vill du installera L^AT_EX på din egen dator finns det sannolikt i din pakethanterare (om du använder Linux), eller i form av T_EX Live.¹ Använder du Mac OS X finns det istället MacT_EX,² och till Windows finns MiK_TE_X.³ Den här introduktionen kan tyvärr inte ge fullständiga instruktioner för att installera dessa paket (det är inte introduktionens syfte); konsultera istället respektive pakets dokumentation.

Under introduktionens gång kommer det refereras till så kallade *paket*, som används för att utöka L^AT_EX med intressanta (och ibland nödvändiga) funktioner. Dessa paket kommer oftast att beskrivas lite kort, men vill man se fullständig dokumentation för varje paket kan man leta på *the Comprehensive T_EX Archive Network* (CTAN).⁴ Det lättaste sättet att hitta paket på CTAN är att använda dess sökfunktion.⁵

¹ <http://www.tug.org/texlive/>

² <http://www.tug.org/mactex/>

³

<http://miktex.org/> ⁴ <http://www.ctan.org/> ⁵ <http://www.ctan.org/search/>

Del I.

Grunderna

1. Grundläggande begrepp

1.1. Snabbkurs i L^AT_EX-kod

Introduktionen har förklarat lite kort vad T_EX/L^AT_EX är och varför du bör använda systemet för att typsätta dina rapporter, artiklar och böcker. Denna del kommer att inleda din T_EX-bana genom att lite kort förklara hur ett L^AT_EX-dokument är uppbyggt, några viktiga begrepp samt hur L^AT_EX-kompilatorn (i det här fallet pdf_latex) fungerar.

De dokument L^AT_EX läser in är enkla textfiler, oftast med filändelsen `.tex`. Dessa kan skapas med vilken textredigerare som helst, men det rekommenderas att man använder en textredigerare med syntaxfärgning (då det underlättar felsökande) och stavningskontroll. Oftast sparar man filen med teckenkodningen UTF-8 (detta gör de flesta moderna textredigerare), men det finns en risk att filerna sparas med teckenkodningen ISO-8859-1,¹ och då måste man ha koll på detta eftersom man måste berätta för L^AT_EX hur filen ska läsas in.

1.1.1. Tomrum

Tomrumstecken, det vill säga mellanslag, tabbar och liknande behandlas alla som "tomrum" av L^AT_EX. Flera sådana tecken i rad tolkas som ett enda tomrum, vilket gör att man kan ex. indentera textstycken i sin L^AT_EX-fil utan att detta påverkar resultatet. Tomrum i början av en rad ignoreras generellt sett, och ett enkelt nyradstecken tolkas också som tomrum.

¹ Det finns även andra teckenkodningar som kan dyka upp; Windows-1252 är en sådan.

1. Grundläggande begrepp

Två nyradstecken på rad (det vill säga en tom rad mellan två textrader) tolkas som styckesindelning, och flera tomma rader efter varandra tolkas som en enda tom rad. Detta gör att \LaTeX -filerna blir mycket enkla att både skriva och läsa, även om man inte är en särskilt bevandrad \TeX niker.

1.1.2. Specialtecken

Vissa tecken är speciella för \LaTeX . Dessa används internt av \LaTeX för att representera speciella konstruktioner, och kan inte användas hur som helst i ett \LaTeX -dokument. Gör man det ändå slutar det vanligtvis bara med att tecknet inte syns, men det kan potentiellt göra att ditt dokument inte ens kompilerar. Specialtecknen är inte särskilt många:

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

För att använda dem måste man lägga till ett (bakvänt) snedstreck, `\`, innan tecknet man vill använda (dock ej innan `\` självt, eftersom \LaTeX tolkar `\\` som att man vill tvinga fram en radbrytning — vill man ha ett `\` använder man istället `\textbackslash`). När det gäller `^` och `~` måste man dessutom lägga till måsvingar efteråt, eftersom dessa tecken normalt används för att modifiera vokaler. Tecknen måste alltså skrivas på följande sätt:

```
\# \^{} \& \_ \{ \} \~{} \textbackslash \% \$
```


1.1.3. L^AT_EX-kommandon

För att säga åt L^AT_EX vad som ska göras används kommandon. Dessa kommandon följer några enkla regler:

- De börjar med ett bakvänd snedstreck (\) och följs av sitt namn, som bara får innehålla bokstäver.^{1,2} Kommandon avslutas av ett mellanslag, en siffra eller annan godtycklig ”icke-bokstav”.
- De är skriftlägeskänsliga (\Kommando är inte samma sak som \kommando).
- Vissa kommandon har även en ”stjärnvariant”, då en stjärna (*) läggs till på slutet av kommandonamnet.

L^AT_EX ignorerar tomrum efter ett kommando. Det är därför nödvändigt, om man vill ha ett mellanslag efter ett kommando, att lägga till antingen en tom parameter {} och ett mellanslag eller ett speciellt mellanslagskommando (till exempel ~) mellan kommandot och nästa ord. Detta stoppar L^AT_EX från att äta upp allt tomrum efter kommandot.

Vissa kommandon kräver en obligatorisk och/eller frivillig parameter som på ett eller annat sätt bestämmer hur kommandot beter sig. Den generella strukturen hos ett kommando med sådana parametrar är relativt enkel:

```
\kommando[frivillig parameter]{obligatorisk parameter}
```

Finns det ingen obligatorisk parameter, eller om den frivilliga parametern inte används, kan hela biten inklusive hakparantes/måsvinge helt utelämnas. Ibland kan den frivilliga parametern innehålla många bitar information, till exempel olika flaggor till paket. Dessa brukar då separeras med ett kommatecken.

¹ Det finns även ett antal kommandon som består av \ och exakt en icke-bokstav, till exempel \&. ² Att namnet bara får innehålla bokstäver är något av en sanning med modifikation, eftersom man med så kallade kategorikoder (eng. *catcodes*) kan ändra vilka tecken L^AT_EX tolkar som bokstäver — inom ramen för den här introduktionen kan man dock anta att endast ”riktiga” bokstäver är tillåtna.

1. Grundläggande begrepp

1.1.4. Kommentarer

Ibland kan det vara bra att kommentera bort (alltså säga åt \LaTeX att ignorera) vissa bitar av dokumentet. Det kan vara för att förklara olika kommandon eller definitioner, eller för att ta bort ett stycke man inte är helt säker på. Detta görs med procenttecknet; när \LaTeX stöter på ett sådant (såvida det inte sitter ett \backslash före) så ignorerar den hela resten av den raden i \TeX -filen:

```
Lite text\ldots{} % ...och en kommentar
```

1.1.5. Grundläggande struktur

När du nu vet hur \LaTeX tolkar tomrum, vad ett kommando är och hur du kommenterar delar av dina filer är det dags att förklara lite närmre hur en typisk \LaTeX -fil ser ut. Kompilatorn förväntar sig en viss struktur; till exempel så måste varje dokument inledas med kommandot

```
\documentclass{...}
```

som berättar för \LaTeX vilken dokumentstil du vill använda (det finns ett antal olika, för till exempel rapporter, böcker eller brev).

Därefter bör man köra kommandon som påverkar hela dokumentet, till exempel genom att ladda in extra paket eller definiera nya kommandon. Paket laddar man in med kommandot \backslash usepackage:

```
\usepackage{...}
```

Både \backslash documentclass och \backslash usepackage tar en obligatorisk parameter (namnet på stilen/paketet) och en frivillig parameter som används för att skicka inställningar till stilen eller paketet som laddas in — en närmare förklaring av båda dessa kommandon kommer senare.

1. Grundläggande begrepp

När allt förarbete är gjort kan man inleda dokumentet. Först berättar man för \LaTeX att innehållet börjar med hjälp av kommandot

```
\begin{document}
```

och därefter följer dokumentets innehåll. När allt innehåll är slut avslutar man dokumentet med det snarlika kommandot

```
\end{document}
```

som säger åt \LaTeX att det inte finns något mer som ska typsättas. Vad man egentligen gör här är att skapa en omgivning med namnet `document`, som betecknar att detta är dokumentets innehåll. Omgivningar förklaras utförligt på [sida 26](#).

1.2. Dokumentets layout

\LaTeX är inte helt oflexibelt, och man kan med hjälp av olika kommandon i inledningen till dokumentet ändra både utseendet och beteendet hos detsamma. Utseendet ändrar man med så kallade dokumentklasser (varje dokument måste specificera exakt en sådan), medan extra funktionalitet läggs till i form av så kallade paket.

1.2.1. Dokumentklasser

Den första informationen \LaTeX behöver när den kompilerar ett dokument är vilken typ av dokument författaren vill skapa. Detta specificeras med hjälp av kommandot `\documentclass`:

```
\documentclass[inställningar]{klass}
```

1. Grundläggande begrepp

Här berättar *klass* vilken sort dokument som ska skapas, eller vilken stil som ska användas. Med de flesta L^AT_EX-distributioner följer ett antal standardklasser, och av dessa är det fyra man kan tänkas komma i kontakt med relativt ofta:

article är en klass för artiklar till tidsskrifter, korta rapporter, dokumentation, och annat som inte har någon annan, specifik klass. Om du inte vet vilken klass du vill ha, börja med article.

report är en klass för längre rapporter (sådana med flera delar eller kapitel), kortare böcker, kandidatrappporter, examensarbeten och doktorsavhandlingar.

book är en klass för riktiga böcker, som ska gå till tryck.

beamer är en klass för presentationer och overhead-bilder.

Utöver dessa finns det även mer specialiserade klasser, anpassade för olika publikationer,¹ tillämpningar² eller konventioner.³

De dokumentklasser (eller paket av dokumentklasser) som kan vara intressanta att titta på om man typsätter dokument efter europeiska konventioner, eller böcker i allmänhet, är ett fåtal:

KOMA-Script är en samling klasser som är tänkta att vara moderna efterträdare till standardklasserna, baserade på europeiska standarder. Samlingen innehåller bland annat klasserna scrbook, scrreprt och scrartcl som ersätter book, report och article, respektive.

memoir är en dokumentklass som skapats för att vara väldigt flexibel och möjliggöra drastiska layoutändringar utan att behöva skriva mycket T_EX-kod på en låg nivå. Den är främst tänkt att ersätta book och report.

octavo är tänkt att ersätta book, med fokus på klassiska design- och layoutprinciper.

¹ Till exempel elsarticle, ieeeTRAN eller revtex ² Exempelvis refman eller sfms ³ Bland annat eskdx, som implementerar ryska konventioner

1. Grundläggande begrepp

tufte-book är en del av samlingen Tufte- \LaTeX och möjliggör typsättning av böcker liknande de som statistikern och pionjären inom datavisualisering Edward Tufte författade. Klassen är tänkt att användas istället för `book`.

Den intresserade uppmanas att leta upp dokumentationen för dessa på CTAN.

Standardklasserna har också ett antal inställningar som kan ges som en kommaseparerad lista i den frivilliga parametern till `\documentclass`:

10pt, 11pt, 12pt sätter textstorleken (för brödtext) i dokumentet. Om ingen av dessa anges är 10pt standard.

a4paper, ... definierar pappersstorleken. Det finns ett antal olika, bland annat `a5paper` och `b5paper`, men standard är `letterpaper` — se därför till att ändra, eftersom denna pappersstorlek inte används i Sverige.

onecolumn, twocolumn sätter antalet kolumner i dokumentet. Standard är en kolumn (`onecolumn`), och vill man ha två kolumner rekommenderas det att man använder paketet `multicol` istället för alternativet `twocolumn`.

twoside, oneside specificerar om man vill ha ett dubbel- eller enkelsidigt dokument. `article` och `report` är enkelsidiga och `book` är dubbelsidig om inget annat anges. Notera att detta bara påverkar dokumentets stil; `twoside` berättar *inte* för din skrivare att du vill ha dubbelsidiga utskrifter.

Säg till exempel att du vill skriva en rapport och att du vill använda den allmänt accepterade textstorleken 12 pt på ett A4-papper. Det är ingen lång rapport, kanske till och med en inlämningsuppgift, och du bestämmer dig därför för att använda `article`-klassen:

```
\documentclass[12pt,a4paper]{article}
```

1.2.2. Paket

När du skriver ett dokument kommer du troligtvis att märka att vissa saker är svåra eller ointuitiva (kanske rent av omöjliga) att göra med grundläggande \LaTeX -kod. Lyckligvis finns det då nästan alltid ett paket på CTAN som förenklar det du vill göra. Installerade paket kan inkluderas med kommandot `\usepackage`:

```
\usepackage[inställningar]{paketnamn}
```

Notera dock att du måste installera dessa paket innan du kan inkludera dem i ditt \LaTeX -dokument. Din \LaTeX -distribution kommer förmodligen med de flesta paket förinstallerade, men om du får ett felmeddelande när du försöker använda ett paket beror det troligtvis på att det inte är installerat. I \TeX Live och Mac \TeX kan man installera nya paket med kommandot `tlmgr install paketnamn`. De flesta paketen kommer även med dokumentation, som kan nås med kommandot `texdoc`, men ibland kan det vara lättare att leta upp motsvarande paket på CTAN och kolla på dokumentationen där istället, eller använda online-tjänsten `TeXdoc.net`.¹

Några paket finns alltid och är mycket viktiga eftersom de berättar för \LaTeX hur in- och utdata (ska) formateras:

fontenc berättar för \LaTeX vilken sorts typsnitt det typsatta dokumentet ska använda; de vanligaste är T3, OT1 och T1 — oftast vill man ha T1, som är vektorbaserad:

```
\usepackage[T1]{fontenc}
```

inputenx berättar för \LaTeX vilken teckenkodning indatan har skrivits i. \LaTeX (som stammar från 80-talet) antar att teckenkodningen är ISO-8859-1 om inget annat anges, men många moderna system använder UTF-8 vilket man då måste berätta:

```
\usepackage[utf8]{inputenx}
```

¹ Man kan då helt enkelt gå till <http://texdoc.net/pkg/<paketnamn>> för att visa dokumentationen för ett paket.

1.3. Från T_EX till PDF

För att skapa ett typsett PDF-dokument utifrån en L^AT_EX-fil måste man köra den genom den så kallade kompilatorn. L^AT_EX kommer inte med något fint GUI med behändiga knappar att trycka på¹; man måste istället köra kompilatorn från terminalen.

Normalt måste man köra kompilatorn åtminstone två gånger, så att L^AT_EX får en chans att förbättra och korrigera småsaker som referenser, innehållsförteckningar med mera.^{2,3} Kompilatorn brukar varna om detta och be om ännu en körning. Använder man vissa externa verktyg som B_BL_AT_EX måste man även köra detta program; arbetsflödet blir då L^AT_EX → B_BL_AT_EX → L^AT_EX → L^AT_EX.

1.3.1. Kompilatorn: pdfL^AT_EX

Det finns ett antal olika kompilatorer att tillgå (den vanligaste heter bara latex och genererar DVI-filer), men den som är att föredra är pdfL^AT_EX. Fördelen med denna kompilator är att den har utökat stöd för vissa mikrotypografiska funktioner, till exempel hängande punktuation (Thành 2000), men framför allt så kompilerar den dina L^AT_EX-filer till PDF-dokument, istället för de DVI-filer latex producerar.

Användningen är enkel; när du vill kompilera ditt dokument (vilket du som sagt vill göra några gånger i rad) anropar du helt enkelt programmet pdf_latex (testa gärna på koden i [exempel 1.1](#)):

```
pdflatex filnamn.tex
```

Detta gör att pdfL^AT_EX kompilerar ditt program och skapar ett antal filer, däribland filnamn.pdf, som är det slutgiltiga (eller, så slutgiltigt L^AT_EX hunnit göra det) PDF-dokumentet.

¹ Även om det finns sådana verktyg, till exempel L_X och T_EXnicCenter. ² Detta är en konsekvens av hur T_EX fungerar. ³ Kompilatorn läser filen bit för bit och expanderar kommandon — det går inte att röra sig ”bakåt” i filen, så vill man ändra på något man redan gått förbi måste man spara information om detta i en extern fil som man läser in under nästa körning.

1. Grundläggande begrepp

```
\documentclass[a4paper,12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}

\begin{document}
Det här är ett första stycke, ett exempel
på hur text typsätts av \LaTeX{}.

Oj, ett nytt stycke! (Två stycken i samma
dokument, kan du tänka dig?)
\end{document}
```

Exempel 1.1: *Ett mycket enkelt \LaTeX -dokument med endast det allra nödvändigaste.*

1.3.2. Automatisera med latexmk

Det kan vara tröttsamt att konstant kompilera sina filer och hålla koll på hur många gånger man ska köra \LaTeX och andra verktyg. Lyckligtvis finns det sätt att automatisera processen, och ett av det enklaste är Perl-skriptet `latexmk`, som finns tillgängligt på CTAN.¹ Skriptet kör \LaTeX och tolkar loggen som skapas för att avgöra om det behövs fler körningar, och kan även konfigureras för att kontinuerligt kolla efter ändringar i \LaTeX -filen (och andra berörda filer) och kompilera om hela dokumentet när dessa ändras.

Att köra `latexmk` är lika lätt som att köra `pdflatex`:

```
latexmk -pdf filnamn.tex
```

¹ <http://www.ctan.org/tex-archive/support/latexmk/>

1.4. Filer du kanske stöter på

När man arbetar med \LaTeX finner man ganska snabbt att det dyker upp en stor mängd olika filer (därför bör man också ha varje dokument i en egen mapp) med mer eller mindre uppenbara filändelser. Vissa behövs för att \LaTeX ska fungera, vissa genereras av \LaTeX och används i senare körningar och vissa är relativt överflödiga.

Eftersom olika paket också kan skapa filer av godtycklig typ kommer listan nedan inte vara fullständig, men förhoppningsvis innehåller den de filer man oftast stöter på i sitt arbete med \LaTeX .

Filer som används av \LaTeX

- .tex** Ett \TeX - eller \LaTeX -dokument. Kompileras med `pdflatex` (eller `latex`, om man vill ha DVI-filer).
- .sty** Ett \LaTeX -paket, som kan inkluderas med `usepackage`.
- .dtx** Dokumenterad \TeX -kod. Hittar du en sån här fil så är det troligtvis ett distribuerat paket, och det borde följa med en fil med ändelsen `.ins`. Om du kör \LaTeX på en `.dtx`-fil så kommer dokumentationen för motsvarande \LaTeX -paket genereras.
- .ins** Installationsfil för en motsvarande `.dtx`-fil. Kör man \LaTeX på denna så kommer den oftast att generera en `.sty`- eller `.cls`-fil (detta beror så klart på vad paketet i fråga innehåller).
- .cls** En dokumentklass. Denna kan väljas med `documentclass`.

Filer som genereras av \LaTeX

- .pdf** PDF-dokument (eng. *Portable Document File*). Det här är sannolikt den fil \LaTeX genererat från din `.tex`-fil, förutsatt att du inte lagrar andra PDF-dokument i samma mapp (vilket du inte bör göra).

1. Grundläggande begrepp

- .log** En detaljerad logg som berättar vad som hände under den senaste körningen av `pdflatex`.
- .toc** Lagrar alla rubriker, och läses in av \LaTeX under nästa körning då den genererar en innehållsförteckning (om en sådan ska finnas med i dokumentet).
- .lof** Som `.toc`, fast med en lista över figurer.
- .lot** Som `.toc` och `.lof`, fast med en lista över tabeller.
- .aux** En fil som innehåller information om korsreferenser, etiketter och annat. Läses in vid nästa körning av `pdflatex`.

2. Typsättning med L^AT_EX

När du nu vet hur den grundläggande strukturen i ett L^AT_EX-dokument ser ut är det dags att börja typsätta riktiga dokument. Att typsätta dokument är dock något som tål att tänkas på — den här delen kommer därför inte bara gå igenom vanliga L^AT_EX-konstruktioner och kommandon, utan även att inleda med en kort diskussion gällande struktur och språk. Efter den här delen bör du med andra ord inte ha några problem med att producera riktiga dokument, varken L^AT_EX- eller strukturmässigt.

2.1. Text- och språkstruktur

Struktur är mycket viktigt för att läsaren ska kunna ta del av de idéer och den information texten förmedlar. Strukturen ska givetvis framgå direkt ur texten, men med väl genomtänkt typografi förstärks denna struktur och hjälper läsaren (och författaren) en bit på vägen.

L^AT_EX skiljer sig från andra typsättningssystem (och ordbehandlare) på det sättet att du endast behöver berätta för L^AT_EX vilken logisk och semantisk struktur din text har. Dessutom så främjar det en god struktur genom det sätt språket är uppbyggt. När man berättat för L^AT_EX vilken struktur texten har tolkar kompilatorn dessa förklaringar efter förbestämda regler (givna av dokumentklassen och de paket som används) och skapar den typografiska strukturen utifrån dessa. Du får alltså vara författaren, och L^AT_EX tar rollen som typsättare.

2.1.1. Textstycket

Den mest grundläggande byggstenen i \LaTeX (och både typografi och författande) är stycket. Vi kallar det ”byggsten” eftersom ett stycke är en typografisk enhet som ska förmedla en tanke eller en idé. Om en ny tanke påbörjas ska alltså även ett nytt stycke påbörjas, och om så inte är fallet ska endast radbrytningar användas.¹ Tvekar du på om du borde ha ett nytt stycke, tänk på din text som en förmedlare av tankar; om du har en styckesindelning men din gamla tanke fortsätter, så ska den troligtvis bort, och om du påbörjar en ny tanke mitt i ett stycke så ska det förmodligen brytas.

Många underskattar vikten av välgenomtänkta styckesindelningar. Andra vet inte ens vad syftet med en styckesindelning är, eller (speciellt i \LaTeX) introducerar nya stycken utan att vara medvetna om det. Detta misstag är mycket lätt att göra om till exempel ekvationer används i texten. Ta en snabb titt på [exempel 2.1](#) och försök lista ut varför tomma rader (det vill säga ett nytt stycke) används i vissa tillfällen men inte andra — det är en subtil men mycket viktig skillnad.

Nästa enhet i sammanhanget är en mening. I engelska texter används ett extra stort mellanrum i slutet av varje mening; detta gör vi inte i svenskan² och eftersom \LaTeX försöker lista ut var den ska lägga in större mellanrum måste man berätta för \LaTeX att den ska låta bli, vilket görs med hjälp av kommandot `\frenchspacing`. Skriver man på engelska ska man alltså utelämna detta kommando, och \LaTeX kommer att efter bästa förmåga gissa var dina meningar tar slut. Gör kompilatorn fel får man ersätta det mellanslag den förlänger med ett fast mellanslag (~).

¹ Notera dock att det oftast är helt meningslöst att bry sig om radbrytningar i \LaTeX , eftersom systemet både avstavar och radbryter bra på egen hand. ² Det finns ganska många saker \LaTeX gör som inte bör göras i svensk typografi — det mesta kommer att tas upp senare i den här delen.

2. Typsättning med L^AT_EX

```
% Exempel ett
\ldots{}med hjälp av identiteten
\begin{equation}
\int\limits_{-1}^1\!\!\sqrt{1-x^2}\;\mathrm{d}x
= \frac{\pi}{2}
\end{equation}
kan man alltså med Monte Carlo-integration ta fram
ett approximativt värde på  $\pi$ .

% Exempel två
\ldots{}Weibullfördelningen, döpt efter den svenska
matematikern Waloddi Weibull, har den kumulativa
fördelningsfunktionen
\begin{equation}
F(x) = 1 - e^{-(x/\alpha)^\beta} \; ; .
\end{equation}

Från denna kan vi med derivering\ldots{}

% Exempel tre
\ldots{}relativt ointressanta observationer.

\begin{equation}
z_{n+1} = z_n^2 + c
\end{equation}
ger å andra sidan upphov till den mycket kända
Mandelbrot-mängden, som\ldots{}

```

Exempel 2.1: Tre exempel på korrekt styckesindelning i samband med ekvationer.

2.1.2. Rubriker

Det andra strukturelementet man måste tänka på¹ är rubriker. Dessa skapar en logisk struktur som delar upp innehållet, och den typografiska effekten av att införa en rubrik är så stark att det nästan är självklart hur man ska använda dessa för att strukturera sin text.

Standardklasserna i L^AT_EX har sex rubriknivåer:

```
\part{...}           % Ingen nivå
\section{...}         % Nivå 1
\subsection{...}     % Nivå 2
\subsubsection{...}  % Nivå 3
\paragraph{...}      % Nivå 4
\subparagraph{...}   % Nivå 5
```

Dessutom finns det i report- och book-klasserna en rubriknivå till, nivå noll, som ges av `\chapter`. Eftersom article-klassen inte känner igen `\chapter` kan man enkelt inkludera artiklar som kapitel i till exempel en bok.

Rubriknivån som ges av `\part` påverkar inte de andra numreringarna, medan alla andra rubrikkommandon återställer numreringen av lägre nivåer. Detta gör att man kan dela upp sitt dokument i delar, som L^AT_EX dessutom per automatik numrerar ner till en viss nivå. Denna definieras av dokumentklassen men kan ändras genom att ställa in den inbyggda räknaren `secnumdepth`:

```
\setcounter{secnumdepth}{2} % Numrera ner till subsection
```

Dessutom kan man införa onumrerade rubriker genom att lägga till en stjärna efter rubrikkommandot:

```
\section*{Inledning} % Onumrerad rubrik
```

¹ Undantaget strukturen i meningar, som är den del av svenska språket och som förklaras bättre av Christoffersson m. fl. (1998).

2. Typsättning med \LaTeX

Notera även att man inte bör lämna någon tom rad mellan sin rubrik och det efterföljande textstycket.

Dessa rubriker kan sedan om så önskas listas i en innehållsförteckning med hjälp av kommandot `\tableofcontents`, där endast de numrerade rubrikerna inkluderas.¹ Djupet på innehållsförteckningen styrs med räkneren `tocdepth`:

```
\setcounter{tocdepth}{2}
```

2.1.3. Rad- och sidbrytningar

Med \LaTeX behöver man i princip aldrig bry sig om att rad- eller sidbryta eftersom detta görs automatiskt. Vid vissa tillfällen kan det dock vara önskvärt att göra det manuellt ändå; framför allt kan det vara intressant att lägga in manuella sidbrytningar för att undvika änkor, horungar eller störande avbrott i textflödet. **Tabell 2.1** listar några av de kommandon som kan användas för att åstadkomma sidbrytningar.

Notera dock att \LaTeX inte lyder kommandoföljder à la `"\newpage\newpage"` då sidbrytningskommandon endast inforgar en sidbrytning *om det behövs*. Vill man ha en tom sida måste man alltså fylla den med något (osynligt), till exempel en `\mbox` innan man sidbryter en andra gång.

Radbrytningar kan enkelt göras med hjälp av `\\` eller `\newline` i de fall då de behövs; oftast är styckesindelning fullt tillräckligt.

2.1.4. Avstavning

Avstavning sköts automatiskt av \LaTeX , förutsatt att det finns avstavningsdata för det språk man använder. Denna kan man ladda in med hjälp av paketet `babel`.

¹ Vill man inkludera även onumrerade rubriker i sin innehållsförteckning måste man uttryckligen lägga till dessa med kommandot `\addcontentsline`. Detta läggs lämpligen till på raden efter rubriken.

2. Typsättning med L^AT_EX

Tabell 2.1: Tre kommandon för att skapa sidbrytningar i L^AT_EX.

Kommando	Resultat
<code>\newpage</code>	Skapar en ny sida.
<code>\clearpage</code>	Skapar en ny sida och tvingar alla <i>floats</i> (mer om dessa på sida 27) att läggas ut innan den nya sidan. Bättre än <code>\newpage</code> om syftet är att bryta för ett nytt kapitel eller liknande.
<code>\cleardoublepage</code>	Gör samma sak som <code>\clearpage</code> , men ser till att den nya sidan är en udda sida, förutsatt att dokumentet är tvåsidigt. Gör exakt samma sak som <code>\clearpage</code> i enkelsidiga dokument.

Paketet `babel` följer med i princip alla L^AT_EX-installationer. Det innehåller avstavningsregler och översättningar till inbyggda kommandon (rubriker till innehållsförteckning, referenslista och så vidare) för ett stort antal språk, däribland svenska och engelska. Det är enkelt att använda `babel` (paketet förklaras närmre i [kapitel 6](#)), till och med för dokument flera olika språk, och oftast räcker det med att helt enkelt inkludera paketet med rätt inställning:

```
\usepackage[swedish]{babel} % Dokumentet är på svenska
```

Inga avstavningspaket är dock kompletta, och skriver man tekniska rapporter är risken stor att något ord inte finns med eller avstavas på fel sätt. Man kan då berätta för L^AT_EX hur man vill att ordet ska avstavas med hjälp av kommandot `\hyphenate`. Kommandots argument är en lista av ord (separerade med mellanslag) som har bindestreck placerade där L^AT_EX får bryta ordet. Sålunda kan raden

```
\hyphenate{FORTRAN span-ku-le-ra}
```

användas för att berätta för L^AT_EX att ordet "FORTRAN" inte får brytas, medan ordet "spankulera" får brytas på de angivna positionerna.

2.1.5. Betoning, med mera

För att betona text i exempelvis böcker använder man *kursiv* text. I L^AT_EX kan man betona text med hjälp av kommandot `\emph`, som i de flesta fall gör just detta; kursiverar texten. Det finns dock andra tillfällen då man använder kursiv text, och det är givetvis dumt att i sådana stycken kursivera för att betona text; i sådana fall använder L^AT_EX därför helt vanlig, upprätt text (för nästan alla dokumentklasser, i alla fall).

Tabell 2.2 visar utöver kommandot `\emph` som används för betoning ett antal andra textförändrande kommandon som kan vara bra att känna till. Exempelvis så finns `\textsc`, som typsätter texten med kapitaler ("små versaler"), och `\textsf` som typsätter med ett sans-serif-typsnitt istället för det vanliga serif-typsnittet.

Man bör dock inte använda dessa kommandon direkt, utan istället med hjälp av `\newcommand` definiera kommandon som beskriver *vad* som typsätts, inte *hur* det ska typsättas. Säg till exempel att du vill typsätta namn på L^AT_EX-paket med sans-serif-typsnitt. Vi kan då definiera ett kommando `\package` som i sin tur tillämpar `\textsf`:

```
\newcommand\package[1]{\textsf{#1}}
```

När detta sedan används i dokumentet ser man direkt att det är ett paket som beskrivs; att det sedan typsätts med sans-serif-typsnitt är inte så intressant just när man skriver dokumentet.

2. Typsättning med L^AT_EX

Tabell 2.2: Viktiga L^AT_EX-kommandon för att ändra textstil.

L ^A T _E X-kommando	Resultat	Kommentar
<code>\textnormal{...}</code>	Dokumentets standardtypsnitt	Det typsnitt som används för ”vanlig” text i dokumentet.
<code>\emph{...}</code>	<i>Betonad text</i>	Ska alltid användas för betoning då kommandot automatiskt hanterar nästlade betoningar på ett bra sätt.
<code>\textrm{...}</code>	Serif-typsnitt	Detta typsnitt har fast bredd och kan användas för att typsätta till exempel kodstycken.
<code>\textsf{...}</code>	Sans serif-typsnitt	
<code>\texttt{...}</code>	Teletype-typsnitt	
<code>\textit{...}</code>	<i>Kursiv text</i>	Använd <code>\emph</code> istället för detta kommando för att betona text.
<code>\textbf{...}</code>	Fetstilt text	Använd <code>\emph</code> istället för detta kommando för att betona text.
<code>\textsc{...}</code>	KAPITÄLER	

2.2. Figurer, tabeller, listor

Figurer, tabeller och listor¹ har i sin \LaTeX -representation en sak gemensamt; de kräver mer än enkla kommandon för att kunna uttryckas på ett enkelt sätt. Därför har man i \LaTeX valt att införa en konstruktion som grupperar sådana företeelser och gör det tydligt hur textens struktur verkligen ser ut — denna konstruktion kan på svenska kallas för en omgivning.

2.2.1. Omgivningar (eng. *environments*)

Omgivningar i \LaTeX definieras utifrån två kommandon; `\begin` och `\end`. De kan nästlas (på det sättet att en omgivning kan innehålla en annan) men kan inte överlappa — varje omgivning måste avslutas innan dess ”förälder” avslutas, vilket gör att de ger en tydlig trädliknande struktur. En omgivning (här *verbatim*) kan alltså skrivas

```
\begin{verbatim}
Det här är omgivningens innehåll
\end{verbatim}
```

där omgivningens innehåll (oftast) med fördel kan indenteras för att göra strukturen tydlig.

Det finns många omgivningar, och de fyller alla olika syften. De viktigaste bildar listor, figurer, tabeller och ekvationer, men en omgivning kan i princip göra vad som helst.

2.2.2. Tre sorters listor

De första omgivningarna vi ska bekanta oss med är de som skapar listor. I \LaTeX finns det tre användbara sorters listor, som alla går att nästla

¹ Men även ekvationer, programkod, vissa kemiska formler, citat och algoritmer, för att nämna några ytterligare exempel.

2. Typsättning med L^AT_EX

i varandra utan problem och som löser tre distinkta problem: numrerade listor (`enumerate`), onumrerade listor (`itemize`) och beskrivningslistor (`description`). Dessa används tillsammans med kommandot `\item` för att bygga enkla listor (se [exempel 2.2](#)) och är relativt självförklarande.

2.2.3. Flytande objekt (eng. *floats*)

Tabeller och figurer är viktiga verktyg för att förmedla resultat på ett effektivt sätt. De är också mycket speciella objekt rent typografiskt, eftersom de till skillnad från vanliga textstycken inte kan brytas över sidor hur som helst. För att placera figurer och tabeller på ett tillfredsställande sätt använder L^AT_EX sig av så kallade *floats*, vilket innebär att objektet får ”flyta iväg” en bit ifrån det ställe där det från början definieras. Detta undviker fula halvtomma sidor och andra obehagliga resultat.

I många böcker typsätts figurer och tabeller så att de antingen flyter till toppen av en sida, botten av en sida, eller så att de tar upp en hel sida. Man typsätter i princip aldrig figurer och tabeller så att de bryter texten. Man kan även göra samma sak med till exempel kod, men detta är inte lika vanligt. Även om det finns sätt att tvinga L^AT_EX att lägga en figur där den definieras så rekommenderas detta alltså inte.

Ett flytande objekt definieras av en omgivning, och har ett valfritt argument som berättar för L^AT_EX hur objektet får placeras. Detta kan innehålla bland annat `t` (*top*), `p` (*page*) och `b` (*bottom*) och är en lista över alla de sätt L^AT_EX tillåts placera objektet. Om inget av dessa fungerar kommer L^AT_EX att placera objektet på en egen sida. Sålunda kan man skapa en *float* med följande kod:

```
\begin{<floattyp>}[tbp] % Får placeras som top eller bottom
% Innehåll i det flytande objektet
\end{<floattyp>}
```

Flytande objekt kan även ha figurtexter, som definieras med kommandot `\caption`. En figurtext ska komplettera figuren och förklara vad som visas, inte varför det visas. Slutsatser och liknande dras istället i brödtexten som

2. Typsättning med \LaTeX

```
\begin{enumerate}  
\item Skapa en lista  
\item Lägg till ett element  
\item Upprepa steg 2 om så  
önskas  
\end{enumerate}
```

1. Skapa en lista
2. Lägg till ett element
3. Upprepa steg 2 om så önskas

(a) En numrerad lista

```
\begin{itemize}  
\item Tyskland  
\item Topologi  
\item Teknolog  
\end{itemize}
```

- Tyskland
- Topologi
- Teknolog

(b) En onumrerad lista

```
\begin{description}  
\item[Häst] ett stort,  
fyrbent däggdjur.  
\item[Öl] en mycket god  
dryck.  
\end{description}
```

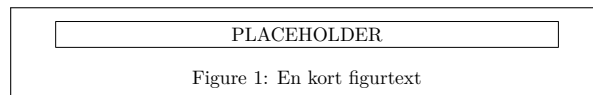
- Häst** ett stort, fyrbent däggdjur.
Öl en mycket god dryck.

(c) En beskrivningslista

Exempel 2.2: De tre enkla listomgivningar \LaTeX tillhandahåller.

2. Typsättning med \LaTeX

```
\begin{figure}[tbp]
  \centering
  \framebox[0.75\textwidth]{
    \medskip PLACEHOLDER \medskip
  }
  \caption{En kort figurtext}
\end{figure}
```



Exempel 2.3: Ett exempel på hur man skapar ett flytande objekt med `figure`.

refererar till figuren. Figurer har sin figurtext undertill medan tabeller har figurtexten ovanför tabellen.

Figurer

Figurer är den kanske vanligaste typen av flytande objekt, och dessa (grafer, illustrationer, diagram och så vidare) infogas i omgivningen `figure`, vilken visas i [exempel 2.3](#).

Att infoga själva figuren kan göras på några olika sätt; man kan skapa den med ett externt program (till exempel `gnuplot` eller `Inkscape`), exportera den direkt från `MATLAB` eller `Mathematica`, eller till och med rita den direkt i \LaTeX . Hur man gör detta diskuteras närmre i [kapitel 4](#).

För att centrera innehållet i figuren använder man kommandot `\centering`, som centrerar allt innehåll i resten av omgivningen. En del får för sig att istället använda omgivningen `center`, men eftersom den omgivningen lägger till extra tomrum både före och efter är det lämpligare att bara använda `\centering`.

Tabell 2.3: De fyra viktigaste kolumntyperna i *tabular*.

Typ	Kommentar
<code>l</code>	Vänsterjusterar innehållet. Använd till text, datum och så vidare.
<code>c</code>	Centrerar innehållet. Använd inte.
<code>r</code>	Högerjusterar innehållet. Använd till tal, mätresultat och så vidare.
<code>p{...}</code>	Avstavat textstycke med angiven bredd. Använd när texten är så lång att tabellen bli för bred för sidan, eller när du skriver långa beskrivningar och/eller kommentarer.

Tabeller

Tabeller är ett annat vanligt flytande objekt, och är ett alldeles förträffligt verktyg när det gäller presentation av data som till exempel mätserier eller liknande. \LaTeX definierar två tabellrelaterade omgivningar; `table` och `tabular`. Det förstnämnda är ett flytande objekt och används precis som `figure`:

```
\begin{table}[tpb]
\centering
\caption{En beskrivning av tabellen}
% Här ska själva tabellen in
\end{table}
```

Den andra omgivningen, `tabular`, är den som faktiskt används för att definiera tabellen. \LaTeX kan i denna omgivning hjälpa till med att rita linjer både kors och tvärs, och definiera kolumner på ett antal olika sätt, men för att hålla sig till en enkel (om än bokinspirerad) devis så bör man ta paketet `booktabs` till hjälp när man skapar tabeller i \LaTeX .¹ En en-

¹ Exakt varför detta är en bra idé förklaras mycket bra av `booktabs`-manualen (Fear 2005).

2. Typsättning med \LaTeX

kel tabell skapad med hjälp av booktabs kan se ut som tabellen i [exempel 2.4](#).

Som vi ser tar tabular-omgivningen ett obligatoriskt argument, som beskriver de kolumner tabellen innehåller. [Tabell 2.3](#) listar de vanligaste kolumntyperna och deras användningsområden. Kolumntyperna anges direkt efter varandra, men kan om man vill separeras med exempelvis mellanslag. Den speciella separatoren | skapar en vertikal linje mellan två kolumner, men detta är inte att rekommendera (Fear 2005).

Efter att omgivningen inletts med \begin följer tabellens rader. Här måste varje rad markeras med explicita nyrader (\), och kolumnerna separeras med och-tecken (&). Notera även de kommandon som skapar horisontella linjer; dessa är från booktabs och används enligt [exempel 2.4](#).

Det är även viktigt att komma ihåg några typografiska regler när det gäller tabeller. Alla dessa ges av Fear (2005, sidan 3). Fritt översatt:

1. Använd aldrig någonsin lodräta linjer.
2. Använd aldrig dubbla linjer.
3. Lägg enheter i tabellhuvudet.
4. Decimalkomma ska alltid föregås av en siffra; alltså 0,1 *inte* bara ,1.
5. Använd inte upprepningstecken eller liknande konventioner för att repetera föregående värde. I många fall räcker det med en tom cell. Gör det inte det, repetera värdet.

Punkt ett och två är enkla att följa om man använder booktabs som manualen föreskriver, och punkt tre och fem är snarare arbetsbesparande än ansträngande att hålla sig till. Även punkt fyra är enkel att följa. Följer man dessa fem punkter får man snygga tabeller varje gång.¹

¹ Läs gärna igenom Fear (2005) ordentligt de tre-fyra första gångerna du sitter med tabeller i \LaTeX , så går användandet in i ryggmärgen.

2. Typsättning med L^AT_EX

```
\begin{tabular}{l r p{6cm}}
\toprule
Konstant & Värde & Kort beskrivning \\
\midrule
\(\gamma\) & \((0,577)\) & Skillnaden mellan den
harmoniska summan från \((1)\) till \((n)\) och den
naturliga logaritmen av \((n)\) då \((n\rightarrow\infty)\).\\
\((e)\) & \((2,718)\) & Den konstant som har
egenskapen att \((e^x)' = e^x\). \\
\((\pi)\) & \((3,1415)\) & Kvoten mellan en cirkels
omkrets och dess diameter. Mycket viktig
konstant i många sammanhang; dyker upp
lite varstans. \\
\bottomrule
\end{tabular}
```

Konstant	Värde	Kort beskrivning
γ	0,577	Skillnaden mellan den harmoniska summan från 1 till n och den naturliga logaritmen av n då $n \rightarrow \infty$.
e	2,718	Den konstant som har egenskapen att $(e^x)' = e^x$.
π	3,1415	Kvoten mellan en cirkels omkrets och dess diameter. Mycket viktig konstant i många sammanhang; dyker upp lite varstans.

Exempel 2.4: En tabell skapad med hjälp av booktabs.

2.2.4. Etiketter och korsreferenser

Eftersom figurer får (och bör få) flyta en bit bort, i vissa fall till andra sidor, måste man kunna referera till dem i texten. Detta görs med så kallade korsreferenser till etiketter. En etikett kan man definiera med kommandot `\label`, inte bara för figurer och tabeller utan även för rubriker, ekvationer och annat. För flytande objekt och andra omgivningar placerar man `\label`-kommandot inuti omgivningen (men efter `\caption` om denna används), medan man för rubriker placerar etiketten direkt efter `\section` (eller motsvarande).

En etikett kan innehålla många olika tecken, men det kan vara bra att hålla sig till (amerikanska) bokstäver, siffror samt kolon (:) som separator. För att kunna hålla koll på sina etiketter bör man även namnge dem på ett logiskt sätt (varje etikett bör beskriva det den refererar till) och gärna inleda varje etikett med en liten beskrivande förkortning. Man kan till exempel märka en figur som visar β -sönderfall med följande etikett:

```
\label{fig:betasonderfall}
```

Etiketterna kan man sedan referera till med `\ref`, som skriver ut det nummer figuren (eller motsvarande) har. Dessutom finns `\pageref`, som skriver ut vilken sida figuren ligger på. Dessa kommandon är dock inte medvetna om vilken typ av objekt man refererar till, så detta måste man berätta:

```
...som man ser i figur~\ref{fig:betasonderfall}...
```

Paketet `varioref` inför det något mer intelligenta kommandot `\vref` (och även `\vpageref`) som formulerar referensen på ett något bättre sätt, beroende på var figuren ligger. Ligger figuren på samma sida är `\vref` i princip ekvivalent med `\ref`, men om figuren är på en annan sida skriver `\vref` även ut vilken sida detta är.

2. Typsättning med \LaTeX

Ett annat paket, som kan ses som en förbättring av varioref, är cleveref. Detta paket räknar automatisk ut vilken typ av referens det är du refererar till, och skriver ut detta. Exemplet ovan kan då reduceras till detta:

```
...som man ser i \cref{fig:betasonderfall}...
```

Paketet hyperref förbättrar alla dessa kommandon genom att göra referenserna till länkar i den slutgiltiga PDF-filen. Med dessa kan man alltså lättare navigera i dokumentet. Förutom detta så lägger hyperref dessutom till en innehållsförteckning som PDF-läsaren kan visa i programmet, och annan PDF-specifik funktionalitet. Paketet rekommenderas starkt om du använder pdf \LaTeX .

2.2.5. Ordlistor

När man skriver tekniska eller vetenskapliga texter finns det ofta anledning att inkludera en ordlista. Det väldigt omfattande paketet glossaries löser precis denna uppgift.

Paketet, som dokumenteras väldigt utförligt av Talbot (2013b), är mycket omfattande och kan användas för att skapa många sorters ordlistor och förteckningar. Den kortare introduktionen för nybörjare (Talbot 2013a) går igenom hur man använder paketet för att skapa en enkel ordlista eller lista över förkortningar.

Den som är intresserad av att använda glossaries bör läsa Talbot (2013a), men i princip går det ut på att man deklarerar en ordlista med kommandot `\makeglossaries`, definierar termer till ordlistan med `\newglossaryentry` och refererar till dessa med `\gls`, `\glspl`, `\Gls` och `\Glspl`. Därefter skriver man ut själva ordlistan med kommandot `\printglossaries`.

Paketet använder även programmet xindy för att sortera ordlistan, och därför måste man (i likhet med biblatex som förklaras i [kapitel 5](#)) köra pdf \LaTeX , xindy och sedan pdf \LaTeX igen (eller `\makeglossaries` istället för xindy, om du har Perl installerat). Detta förklaras närmre av Talbot (2013b) och Talbot (2013a).

2.2.6. Förkortningar

I viss teknisk litteratur (avhandlingar, artiklar och liknande) kan det även vara bra att inkludera en lista över förkortningar. Det problemet går att lösa med glossaries, men det är ofta bättre att använda ett specialiserat paket, till exempel acro (Niederberger 2013). Till skillnad från glossaries använder acro inte några externa program, och är därför något enklare och mindre tidskrävande att använda.

2.2.7. Sakregister

En sista komponent i många större projekt, speciellt böcker eller liknande litteratur, är ett sakregister. Även dessa kan skapas på ett relativt automatisk sätt av \LaTeX , med hjälp av paketet imakeidx. Paketet förklaras av Beccari och Gregorio (2012), men användningen liknar på många sätt glossaries. I likhet med glossaries måste man köra xindy eller makeindex mellan \LaTeX -körningarna, men under vissa förhållanden¹ kan \LaTeX göra detta automatiskt. Givetvis görs detta även av latexmk om man använder det verktyget för att kompilera sina dokument.

2.3. Typografiska betänkligheter

När man skriver rapporter är det även viktigt att hålla sig till de typografiska regler \LaTeX inte håller reda på; sådana som har med citattecken, tankstreck och annat att göra. Även datum, decimalavskiljare och liknande är viktigt att tänka på. Dessa typografiska regler kommer i korthet att förklaras på de närmsta sidorna, till vilka viss inspiration tagits från von Schultz (2005).

¹ Om man kör pdf \LaTeX med flaggan `--shell-escape`, eller om \TeX -distributionen är konfigurerad med restriktiv `\write18`-åtkomst som tillåter dessa program.

2. Typsättning med \LaTeX

Tabell 2.4: Citattecken i \LaTeX .

\LaTeX -kod	Resultat	Kommentar
<code>``Text''</code>	“Text”	Engelska citattecken. Använd inte i svensk text!
<code>`Text'</code>	‘Text’	Nästlade engelska citattecken (för citat inuti citat). Använd inte i svensk text!
<code>''Text''</code>	”Text”	Svenska citattecken.
<code>'Text'</code>	’Text’	Nästlade svenska citattecken (för citat inuti citat).

2.3.1. Citattecken

Det mest grundläggande man bör uppmärksamma i \LaTeX , eftersom det är en grop man lätt faller i, är citattecken. Man kan inte i \LaTeX använda det vanliga citattecknet (") för att generera ett citattecken, utan man måste istället använda kombinationer av *backticks* (‘) och apostrofer (').¹

Tabell 2.4 visar hur dessa används för att citera text.

2.3.2. Streck av olika längd

Det kan tyckas vara ett rent typografinönderi man inte behöver bry sig om, men faktum är att det finns olika sorters streck (lite på samma sätt som att det finns olika skiljetecken). Det är relativt viktigt att skilja på dessa och se till att använda rätt streck vid rätt tillfälle.

Det första strecket, minustecknet, är relativt långt (—) och går i princip inte att missa eller missbruka. Det infogas av \LaTeX då man använder ett vanligt bindestreck (-) i matematikläge²:

¹ Men man kan även använda paketet `csquotes`, som gör det enkelt med nästlade citat och olika stilar i olika språk. Detta paket och även miljöer för lite längre citat behandlas närmre på sida 41 ² Mer om detta i kapitel 3.

2. Typsättning med L^AT_EX

```
...och ett flyttal kan därför ha värdet \(-0\).
```

Nästa tecken, det vanliga bindestrecket (-), behöver man i princip aldrig använda själv. Det infogas av L^AT_EX då ord avstavas, men kan även infogas manuellt genom att helt enkelt skriva ett vanligt bindestreck i L^AT_EX-koden.

Det tredje strecket kallas intervallstreck (–) och är något längre än bindestrecket. Detta används som namnet antyder för att typsätta intervall, och skrivs med hjälp av två bindestreck (--):

```
...det kan ta upp emot 3--4 timmar att...
```

Det fjärde och sista strecket kallas långt tankstreck (—) och är det längsta av strecken. Det långa tankstrecket är relativt ovanligt inom svensk typografi, men det finns ingen anledning att undvika det.¹ Ett långt tankstreck skriver man med tre bindestreck i följd (---), och man bör omge det med mellanslag:

```
...det är --- för tillfället åtminstone --- omöjligt att...
```

2.3.3. Avstånd mellan stycken

Det finns två vedertagna sätt att separera stycken; indrag och mellanrum. Man tillämpar endast *ett* av dessa åt gången, och traditionellt sett så brukar man i svensk typografi använda mellanrum, medan man i amerikansk typografi använder indrag (varför detta är vad L^AT_EX gör *out-of-the-box*).

¹ För tankar (*no pun intended*) angående användningen av tankstreck, se Christoffersson m. fl. (1998, sidorna 46–47) — observera dock att man *inte* som Christoffersson m. fl. föreslår ska använda tankstrecket i intervall eller punktlistor. Det Språkrådet (2008) och Christoffersson m. fl. betecknar som "tankstreck" är i själva verket mycket närmre det som i L^AT_EX anses vara ett intervallstreck, det vill säga ett lite kortare tecken. Personligen föredrar jag att göra skillnad på de två, och gör man det bör intervallstrecket användas till intervall, och det långa tankstrecket som alternativ till kommatecknet.

2. Typsättning med \LaTeX

För att få \LaTeX att använda mellanrum istället för indrag bör man egentligen skriva helt nya dokumentklasser, men eftersom detta är ett stort jobb kan man istället ”patcha” \LaTeX . Detta görs av paketet `parskip`:

```
\usepackage{parskip}
```

2.3.4. Andra saker att hålla reda på

- Datum skrivs på bästa sätt i ISO-format (YYYY-MM-DD), där man använder intervallstreck för att skilja delarna åt.
- Decimalavskiljare i svensk skrift är kommatecknet. I matteläge tolkat \LaTeX ett komma som en koordinatavskiljare, och lägger till ett litet mellanrum efter. För att skriva ut numeriska värden med kommatecken som avskiljare kan man använda kommandot `\num` från `siunitx`, som förklaras närmre på [sida 68](#).
- Enheter ska typsättas med ett (litet) mellanrum mellan tal och enhet. Detta görs enklast med paketet `siunitx`.

2.3.5. Skarpare typsnitt: `fontenc` och `lmodern`

Eftersom \LaTeX och dess tillhörande typsnitt (Computer Modern) är relativt gamla, och eftersom de designats för att fungera väl med det mycket gamla POSTSCRIPT-formatet, så uppstår det ibland några smärre problem. Det första problemet som brukar dyka upp i samband med typsnitt i \LaTeX är att texten blir otydlig och suddig i det typsatta PDF-dokumentet. Detta beror på att de typsnitt som används är baserade på bitmappar, som inte skalar särskilt väl. Därför måste man använda paketet `fontenc` för att säga åt \LaTeX att använda vektorbaserade typsnitt istället:

```
\usepackage[T1]{fontenc}
```

2. Typsättning med \LaTeX

Det är då man brukar stöta på nästa problem, som yttrar sig i att text (främst skandinaviska tecken och andra tecken som inte finns representerade i ANSI) inte kopieras på ett korrekt sätt. Detta är en konsekvens av att de mycket gamla Computer Modern-typsnitten inte innehåller de glyfer som motsvarar sådana tecken, och istället används då (något fulhackade) komposita tecken istället. För att lösa detta måste man byta typsnitt.

Det typsnitt som brukar användas istället för Computer Modern är dess mer moderna klon *Latin Modern*. Detta typsnitt ser i princip ut som Computer Modern, men innehåller alla de glyfer som behövs för att typsätta svensk (och annan icke-anglikansk) text. För att använda Latin Modern inkluderar man paketet `lmodern`:

```
\usepackage{lmodern}
```

2.4. Andra viktiga delar

Även om innehållet är det viktigaste så finns det andra delar av en rapport eller artikel som också måste ges lite uppmärksamhet. Titelsidor bör man till exempel alltid ha, och det är även viktigt att ha koll på hur man citerar och vilken teckenkodning man använder. I vissa fall kan man även vilja referera till vissa längder, eller använda kommandon som \LaTeX anser vara "ömtåliga" i situationer de inte är tänkta att användas. Avslutningen på detta kapitel kommer därför att gå igenom dessa småsaker.

2.4.1. Titelsidan

Det kan vara trevligt att i sitt arbete ha en fin titelsida. Även denna kan \LaTeX (åtminstone standardklasserna) typsätta åt dig. Detta är enkelt att göra, och fungerar i princip så att man definierar sin metadata (titel, datum,

2. Typsättning med L^AT_EX

författare) med hjälp av ett par kommandon, varpå man kör kommandot `\maketitle` för att skriva ut en titelsida:

```
\title{En fantastisk rapport\thanks{Tack till DD-gudarna
    för visad barmhärtighet och stabila filservrar.}}
\date{2011-12-13}
\author{E.~Johansson\thanks{erik.johansson@example.org}}
\maketitle
```

Kommandot `\thanks` kan användas för att infoga korta tack eller extra information om titel eller författare, så som mailadresser eller liknande. Den typografiska effekten av `\thanks` (i standardklasserna) är en fotnot.

Man kan dessutom infoga en sammanfattning (eng. *abstract*) med hjälp av omgivningen `abstract`. Denna omgivning inkluderas lämpligen direkt efter `\maketitle`, och skapar automatisk ett litet textstycke med rubriken ”Sammanfattning” (eller, om engelska valts som språk, ”Abstract”).

Vill man (säg, i ett kandidatarbete) ha både *abstract* och sammanfattning måste man använda `babel`s funktioner för att byta språk mitt i dokumentet. Man måste då först ladda in `babel` med stöd för båda språken (notera att det sista språket i listan är aktivt från dokumentets start):

```
\usepackage[english,swedish]{babel}
```

Därefter byter man helt enkelt språk när man vill ha sitt *abstract*:

```
\begin{otherlanguage}{english}
\begin{abstract}
    % Abstract
\end{abstract}
\end{otherlanguage}
\begin{abstract}
    % Sammanfattning
\end{abstract}
```

2.4.2. Citat

I L^AT_EX används ofta `` och '' för att typsätta citattecken,¹ och dessa teckenkombinationer resulterar då i det öppnande och stängande citattecknet så som det används i engelsk typografi. I svensk typografi används endast det senare, så skriver du på svenska ska du bara använda '. Använder man UTF-8 i sina dokument (vilket man bör) kan man även använda citattecknet " självt (notera att detta inte är det tecken som normalt produceras med Shift+2 utan ett korrekt citattecken).

Det finns även två omgivningar i L^AT_EX som kan användas för att märka upp så kallade blockcitat (eng. *blockquotes*), alltså lite längre citat på ett eller flera stycken; quote och quotation. Det första, quote, är tänkt att användas till citat på högst ett stycke, medan quotation ska användas för flera stycken.

För att det här med citattecken, längre citat och även källhänvisning ska kunna hanteras på ett enklare och mer konsekvent sätt skapades paketet csquotes. Det är ett paket som varmt rekommenderas och som inför ett antal användbara kommandon. Bland annat definieras kommandot \enquote, som automatiskt omsluter texten med rätt citattecken även för citat inuti citat, och \textquote, som är en variant som även lägger till en källhänvisning. Dessutom definieras omgivningen displayquote, som ersätter både quote och quotation. Paketets dokumentation, som finns på CTAN, är värd en genomläsning då paketet har många användbara kommandon.

2.4.3. Fotnoter

Fotnoter är små noteringar eller parantetiska inlägg² som inte är väsentliga men som författaren ändå vill inkludera utan att störa textrytmen. I L^AT_EX kan man skapa fotnoter med kommandot \footnote:

```
\ldots{}ibland vill\foonote{Eller måste, beroende på situation} man\ldots
```

¹ Detta förklarades redan på sida 36. ² Ibland används även fotnoter för källhänvisning — mer om detta i kapitel 5.

2. Typsättning med \LaTeX

Om man använder fotnoter bör man även använda paketet `fnpct`, som förutom att fixa fotnoter nära skiljetecken även definierar ett kommando `\multfootnote`, som gör det möjligt att typsätta flera fotnoter brevid varandra.

2.4.4. Teckenkodning (UTF-8)

Teckenkodning, som kort nämndes på [sida 13](#), är viktigt att hålla koll på. Det finns två större teckenkodningar på Linux-plattformar, UTF-8 och ISO-8859-1, och en till på Windows-system, CP-1252. Det är viktigt att man vet vilken teckenkodning man skriver sina dokument i, då man måste berätta detta för \LaTeX . Gör man inte det förutsätter \LaTeX att dokumentet är skrivet i ISO-8859-1, och är det då inte det så kommer hemska saker hända. Teckenkodningen specificeras med paketet `inputenx`¹:

```
\usepackage[<teckenkodning>]{inputenx}
```

Här refererar man till UTF-8 som `utf8`, ISO-8859-1 som `latin1` och CP-1252 som `cp1252`. Det finns även en del mer obskyra teckenkodningar,² men det är troligtvis de tre nämnda du kommer att stöta på.

2.4.5. Längder

Titt som tätt behöver man i \LaTeX använda längder för att till exempel kontrollera bredden på figurer och liknande. Den viktigaste fördefinierade längden är `\textwidth`, som representerar bredden av det område på sidan som får fyllas med text. Genom att lägga till ett flyttal innan längden kan man dessutom berätta för \LaTeX att man till exempel vill ha en figur som endast täcker $3/4$ av bredden:

```
0.75\textwidth
```

¹ Tidigare användes `inputenc`, men det har ersatts av det mer uppdaterade `inputenx`.

² De som stöds av `inputenx` listas i paketets manual.

2. Typsättning med L^AT_EX

Det finns en stor mängd längder i L^AT_EX, och många av dem relaterar till olika mått på sidan (Oetiker m. fl. 2011, sidan 132), medan andra (till exempel `\smallskipamount`) är mer obskyra och godtyckliga.

2.4.6. Ömtåliga kommandon

En del text, så som rubriker, figurtexter med mera, kan dyka upp mer än en gång i ett dokument (till exempel i innehållsförteckningen). Argument till sådana kommandon kallas för rörliga (eng. *moving arguments*) och kan ställa till med en del problem. Det finns nämligen kommandon (`\footnote`, till exempel) som är ”ömtåliga” (eng. *fragile*), och som därför inte kan användas i rörliga argument utan vidare. Vill man göra det måste man skydda dem, något som görs med kommandot `\protect`:

```
\section{En rubrik\protect\footnote{Med en fotnot.}}
```

Del II.

Matematiken

3. Matematik med \LaTeX och \mathcal{AMS}

\LaTeX är ett mycket bra system för typsättning, och ett av de områden \LaTeX är starkast inom är typsättning av matematik. För att utöka och förenkla de grundläggande mekanismer \LaTeX har för typsättning av matematik har *American Mathematical Society*, \mathcal{AMS} , skapat ett antal paket (främst `amsmath` och `amssymb`) som kollektivt refereras till som $\mathcal{AMS}\text{\LaTeX}$ och som gör typsättning av matematik med \LaTeX mycket bättre.

Dessa paket är dock enorma, och det finns hela böcker tillägnade endast typsättning av matematik i \LaTeX (till exempel Voß 2010, som för övrigt är en mycket bra referens att ha till hands när man typsätter matematik i \LaTeX). Den här delen kommer att ge en kort introduktion till de allra enklaste och vanligaste konstruktionerna man stöter på när man typsätter matematik i \LaTeX , och tar mycket inspiration från Voß (2010).

För att typsätta matematik så som det beskrivs i den här delen måste du använda paketen `amsmath` och `amssymb`, som bör följas med din \LaTeX -distribution:

```
\usepackage{amsmath}
\usepackage{amssymb}
```

3.1. Att visa ekvationer

Ekvationer är mångfacetterade och man kan vilja inkludera ekvationer av olika anledningar och på olika sätt. I \LaTeX kan figurer infogas *inline*

3. Matematik med \LaTeX och \AMS

(det vill säga i löpande text) eller i omgivningar (vilket frilägger ekvationen, något som är lämpligt för längre ekvationer eller ekvationer av större vikt).

3.1.1. Ekvationer i löpande text

Ekvationer i löpande text kan infogas på två olika sätt; dels genom att placera matematiken mellan dollartecken ($\sin(x)$) och dels genom att använda de \LaTeX -specifika alternativen till dollartecken, $\left($ och $\right)$. Det rekommenderas att använda det senare, eftersom detta inte ger fullt så obskyra felmeddelanden när saker är fel, men även dollartecknen fungerar bra och det är ingen typografisk skillnad.

Matematik som typsätts i löpande text kommer att anpassas i höjd för att inte störa texten omkring. Det innebär att till exempel bråk kommer att se tilltryckta ut. Oftast bör man dock inte använda matematik i löpande text vid de tillfällen då det kommer se ”fel” ut, eftersom ett block då är mer logiskt, textmässigt. Matematik i löpande text är istället oftast konstanter, enstaka värden och kanske någon olikhet.

3.1.2. Ekvationer som ekvationer

Längre och viktigare ekvationer bör inte typsättas i löpande text — man bör istället använda en av de många omgivningar ($\AMS\LaTeX$ ger tillgång till. Den allra enklaste omgivningen (som fungerar i de flesta fall) är `equation`, som finns i en vanlig variant som skapar en numrerad ekvation samt en stjärnvariant (`equation*`) som ger en onumrerad ekvation. Som en tumregel bör man endast använda den numrerade varianten om man behöver referera till ekvationen, eller om det är en viktig definition som andra kan tänkas vilja referera till.

Fristående ekvationer kan typsättas mycket bättre eftersom de inte är begränsade av omgivande text. Integraler och summor kan göras så höga som det krävs, och paranteser kan växa nästan obegränsat. Det är dock viktigt att komma ihåg att en fristående ekvation *oftast* inte ska vara ett

3. Matematik med \LaTeX och \AMSTeX

eget stycke. **Exempel 2.1** på sida 20 visade hur ett antal ekvationer och mellanrummet mellan dessa och den omgivande texten varierade beroende på ekvationens plats i stycket. Trots att ekvationen är "fristående" bör man alltså behandla den precis som en ekvation i löpande text. Skillnaden är främst typografisk.

3.1.3. Referera till ekvationer

Likt figurer och tabeller finns det även ett behov av att kunna referera till sina ekvationer. På samma sätt som för figurer markerar man den ekvation man vill referera till med hjälp av `\label`, men istället för att referera med hjälp av `\ref` eller `\vref` bör man använda `\eqref` för att referera till ekvationer.¹ Sådana referenser typsätts lite annorlunda, och bör inte heller föregås av någon text som indikerar att det är en referens.

3.2. En snabbkurs i \LaTeX -matematik

Nu när du vet hur man skapar förutsättningarna för att typsätta matematik är det dags att gå in på detaljerna. Det finns många konventioner och betänkligheter när det gäller typsättning av matematik, så det är svårt att göra något annat än att skrapa på ytan, men sidorna framöver ämnar presentera tillräckligt mycket för att du ska klara att typsätta det mesta du stöter på under en typisk kandidatutbildning i Teknisk Fysik eller Teknisk Matematik.

Till att börja med måste två elementära operationer förklaras; *superskript* (för potenser och dylikt) och *subskript* (för index och liknande). Ett superskript inför man i \LaTeX med hjälp av en cirkumflex (^); koden `\(e^x\)` producerar alltså e^x . Ett subskript å andra sidan införs med hjälp av ett understreck (_) på liknande sätt; `\(x_i\)` blir x_i . Notera att dessa tecken endast påverkar nästa *kommandoföljd*, vilket innebär att man om man vill

¹ Använder man `cleveref` kan man hålla sig till `\cref`, som sköter det snyggt ändå.

3. Matematik med \LaTeX och \AMS

upphöja/nedsänka mer än ett tecken (eller kommando), så måste man avsluta dessa med måsvingar:

```
 $\backslash(x_{i,j})$ 
```

Detta innebär även att om man vill ha både superskript och subskript efter en variabel så är det inget problem; man inkluderar bara dessa i följd direkt efter variabeln (eller operatoren):

```
 $\backslash(x_i^2)$ 
```

Dessa två exempel producerar alltså $x_{i,j}$ och x_i^2 , respektive. Det skall även nämnas att man kan sätta sub- och superskript framför en tom grupp, vilket är användbart om man vill beteckna exempelvis isotoper: $\text{\texttt{_92}\^{\texttt{235}}\text{\texttt{U}}}$ kan till exempel beteckna en uranatom och typsätts ${}^{235}_{92}\text{U}$.

Det finns även några viktiga skillnader mellan matematiktypsättning och vanlig texttypsättning, som främst har att göra med mellanrum:

- Mellanslag i koden ignoreras nästan alltid. De mellanrum som ska visas i det typsatta dokumentet placeras istället ut av \LaTeX . Det finns även explicita kommandon som gör detta. Detta gör att du kan vara liberal med mellanrum i din kod, vilket ökar läsbarheten.
- Tomma rader tillåts inte i matematiktypsättning. Endast ett ”stycke” per ekvationsomgivning får förekomma.
- Alla bokstäver anses vara variabler och typsätts därför med kursiv text. De bokstäver som inte är variabler (till exempel differentieringsoperatoren d) måste skrivas in med kommandot $\text{\texttt{\textbackslashmathrm\{...\}}}$. Längre bitar text (eller snarare text som egentligen skulle kunna vara brödtext) bör skrivas in med kommandot $\text{\texttt{\textbackslashtext}}$.

3.2.1. Enkla konstruktioner

Det finns ett antal enkla konstruktioner inom matematiken som var man (och kvinna) bör känna till och kunna typsätta. Dessa är konstruktioner som bråk, binomialkoefficienter, rötter och så vidare. Dessa visas kollektivt i [tabell 3.1](#).

Ibland kan det även vara önskvärt att markera bitar av ekvationer med små kommentarer ovan och under matematiken. Detta kan göras med kommandona `\overbrace` och `\underbrace`, vilket visas i [exempel 3.1](#).

3.2.2. Funktioner etc.

Många välanvända eller kända funktioner och operatorer (trigonometriska funktioner och liknande) har långa namn som bör skrivas i upprätt text (det vill säga med `\mathrm`), men då detta är både tidskrävande och inelegant ur en semantisk synpunkt definierar $\text{\AMS}\text{\LaTeX}$ några av dessa som kommandon istället — sinus blir `\sin`, cosinus `\cos` och så vidare.

De vanligaste trigonometriska funktionerna, logaritmer samt vanliga operationer så som `max`, `min`, `sup`, `inf` och `lim` finns som kommandon, och om man behöver typsätta en sådan funktion kan man oftast satsa på att den finns. Finns den inte så kan man skapa den med hjälp av kommandot `\DeclareMathOperator`:

```
\DeclareMathOperator{gcd} % Största gemensamma nämnare
```

3.2.3. Symboler

Matematiken innehåller ett stort antal symboler, däribland det grekiska alfabetet, talmängder, oändligheter, tomma mängden och så vidare (och då inkluderar vi inte de pilar, operatorer, hattar och olikhetstecken som diskuteras senare). Några av dessa symboler listas i [tabell 3.2](#) till [3.3](#) — notera

3. Matematik med \LaTeX och \mathcal{AMS}

Tabell 3.1: Enkla matematiska konstruktioner i \LaTeX .

Konstruktion	\LaTeX -kod	Resultat
Kvadratrot	<code>\sqrt{x}</code>	\sqrt{x}
n -rot	<code>\sqrt[n]{x}</code>	$\sqrt[n]{x}$
Binomialkoefficient	<code>\binom{n}{k}</code>	$\binom{n}{k}$
Bråk	<code>\frac{a}{b}</code>	$\frac{a}{b}$
Alternativt bråk ^a	<code>\sfrac{a}{b}</code>	a/b

^a Med paketet xfrac

```
\begin{equation*}
\underbrace{y'' + 2y' + y}_{\mathrm{D}(y)}
= \overbrace{ax^2+bx+c}^{f(x)}
\end{equation*}
```

$$\underbrace{y'' + 2y' + y}_{\mathrm{D}(y)} = \overbrace{ax^2 + bx + c}^{f(x)}$$

Exempel 3.1: Annotering med `\overbrace` och `\underbrace`.

3. Matematik med \LaTeX och \AMS

Tabell 3.2: Diverse användbara symboler i \LaTeX .

Kommando		Kommentar
<code>\hbar</code>	\hbar	Konstant inom kvantfysiken
<code>\Re</code>	\Re	Realdel av komplext tal
<code>\Im</code>	\Im	Imaginärdel av komplext tal
<code>\aleph</code>	\aleph	Beskriver kardinalitet av mängder
<code>\forall</code>	\forall	
<code>\exists</code>	\exists	
<code>\nexists</code>	\nexists	
<code>\complement</code>	\complement	
<code>\emptyset</code>	\emptyset	Tomma mängden; även <code>\varnothing</code>
<code>\infty</code>	∞	Oändlighet

att några av de grekiska bokstäverna inte har några kommandon för versaler, eftersom dessa är identiska med latinska bokstäver, samt att några har "varianter" som ser något annorlunda ut (till exempel `\varepsilon`, som är mer lik det lilla epsilon man brukar rita för hand än `\epsilon`). Notera även att de inte typsätts i kursiv stil, vilket oftast är lämpligt. I så fall bör man använda paketet `fixmath`, som ser till att även de grekiska bokstäverna typsätts som de variabler de faktiskt brukar vara.

Det är givetvis svårt (en del skulle säga omöjligt) att hålla reda på motsvarande \LaTeX -kommandon för alla dessa olika symboler själv, och det kan därför vara bra att ha en bra referens (till exempel Pakin 2009). Man kan även använda det utmärkta verktyget Detexify,¹ i vilket man kan rita sin symbol och få en lista över liknande \LaTeX -symboler och deras motsvarande kommandon.

¹ <http://detexify.kirelabs.org/classify.html>

3. Matematik med \LaTeX och \AMSTeX

Tabell 3.3: Kommandon motsvarande symbolerna i grekiska alfabetet.

Gemen		Versal		Variant	
<code>\alpha</code>	α	<code>\mathrm{A}</code>	A		
<code>\beta</code>	β	<code>\mathrm{B}</code>	B		
<code>\gamma</code>	γ	<code>\Gamma</code>	Γ		
<code>\delta</code>	δ	<code>\Delta</code>	Δ		
<code>\epsilon</code>	ϵ	<code>\mathrm{E}</code>	E	<code>\varepsilon</code>	ε
<code>\zeta</code>	ζ	<code>\mathrm{Z}</code>	Z		
<code>\eta</code>	η	<code>\mathrm{H}</code>	H		
<code>\theta</code>	θ	<code>\Theta</code>	Θ	<code>\vartheta</code>	ϑ
<code>\iota</code>	ι	<code>\mathrm{I}</code>	I		
<code>\kappa</code>	κ	<code>\mathrm{K}</code>	K		
<code>\lambda</code>	λ	<code>\Lambda</code>	Λ		
<code>\mu</code>	μ	<code>\mathrm{M}</code>	M		
<code>\nu</code>	ν	<code>\mathrm{N}</code>	N		
<code>\xi</code>	ξ	<code>\Xi</code>	Ξ		
<code>o</code>	o	<code>\mathrm{O}</code>	O		
<code>\pi</code>	π	<code>\Pi</code>	Π	<code>\varpi</code>	ϖ
<code>\rho</code>	ρ	<code>\mathrm{P}</code>	P	<code>\varrho</code>	ϱ
<code>\sigma</code>	σ	<code>\Sigma</code>	Σ	<code>\varsigma</code>	ς
<code>\tau</code>	τ	<code>\mathrm{T}</code>	T		
<code>\upsilon</code>	υ	<code>\Upsilon</code>	Υ		
<code>\phi</code>	ϕ	<code>\Phi</code>	Φ	<code>\varphi</code>	φ
<code>\chi</code>	χ	<code>\mathrm{X}</code>	X		
<code>\psi</code>	ψ	<code>\Psi</code>	Ψ		
<code>\omega</code>	ω	<code>\Omega</code>	Ω		

Operatorer

Förutom symbolerna som nyss diskuterades finns även en mycket stor mängd operatorer inom matematiken. Även dessa kan man leta fram med hjälp av Detexify, men de flesta (åtminstone de som används frekvent) har logiska namn; nabla-operatorn (∇) heter till exempel `\nabla`, och tillhörandeoperatorn (\in) heter `\in`. Listor över de vanligaste pilarna, relationssymbolerna och operatorerna i \LaTeX återfinns i [tabell 3.5](#) till [3.7](#).

Hattar, prickar, primtecken och liknande

Ibland vill man även modifiera variablers betydelse, kanske för att beteckna en transformerad variabel, en vektor eller en derivata med avseende på tid. Detta kan göras med de \LaTeX -kommandon som listas i [tabell 3.4](#), tillsammans med en kort förklaring av notationens innebörd.

Just för vektorer existerar det dock flera olika notationer. En populär variant (som \LaTeX främjar) är att sätta dit en liten pil ovanför variabeln (\vec{a}), vilket görs med `\vec`. Andra föredrar att skriva variabeln fetstilt eller med krittavletypsnitt, vilket görs med `\boldmath` och `\mathbb`, respektive.

Min rekommendation är att använda `\vec` (eftersom det i koden indikerar att vi pratar om en vektor), och omdefiniera den om man föredrar en annan notation. Man kan till exempel använda följande kommando för att göra så att `\vec` istället visar variabeln fetstilt¹:

```
\renewcommand\vec[1]{\boldmath{#1}}
```

¹ Med paketet `isomath` — som kan rekommenderas — kan koden reduceras till `\let\vec\vectorsym`

3. Matematik med \LaTeX och \AMS

Tabell 3.4: Hattar och dylikt med \LaTeX -kommandon.

Notation		Kommentar
<code>\hat{a}</code>	\hat{a}	Transformerade variabler
<code>\vec{a}</code>	\vec{a}	Vektorer (alternativ notation existerar)
<code>\dot{a}</code>	\dot{a}	Förstaderviata med avseende på tid
<code>\ddot{a}</code>	\ddot{a}	Andraderivata med avseende på tid
<code>a'</code>	a'	Generell förstaderivata
<code>a''</code>	a''	Generell andraderivata
<code>\bar{a}</code>	\bar{a}	Statistiskt medelvärde
<code>\tilde{a}</code>	\tilde{a}	Statistisk median
<code>\widehat{abc}</code>	\widehat{abc}	Variant av <code>\hat</code> för flera tecken

Tabell 3.5: Viktiga pilar i matematisk notation och deras \LaTeX -kommandon.

Kort		Lång	
<code>\gets</code>	\leftarrow	<code>\longleftarrow</code>	\longleftarrow
<code>\to</code>	\rightarrow	<code>\longrightarrow</code>	\longrightarrow
<code>\leftrightharpoonup</code>	\leftrightarrow	<code>\longleftrightarrow</code>	\longleftrightarrow
<code>\Leftarrow</code>	\Leftarrow	<code>\Longleftarrow</code>	\Longleftarrow
<code>\Rightarrow</code>	\Rightarrow	<code>\Longrightarrow</code>	\Longrightarrow
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\iff</code>	\iff
<code>\mapsto</code>	\mapsto	<code>\longmapsto</code>	\longmapsto

3. Matematik med \LaTeX och \AMSTeX

Tabell 3.6: Viktiga relationssymboler och deras \LaTeX -kommandon.

Mindre		Större		Lika	
$<$	$<$	$>$	$>$	$=$	$=$
$\backslash nless$	\nless	$\backslash ngtr$	\ngtr	$\backslash neq$	\neq
$\backslash leq$	\leq	$\backslash geq$	\geq	$\backslash equiv$	\equiv
$\backslash nleq$	\nleq	$\backslash ngeq$	\ngeq		
$\backslash lesssim$	\lesssim	$\backslash gtrsim$	\gtrsim	$\backslash approx$	\approx
$\backslash ll$	\ll	$\backslash gg$	\gg		
$\backslash subset$	\subset	$\backslash supset$	\supset		
$\backslash subseteq$	\subseteq	$\backslash supseteq$	\supseteq		

Tabell 3.7: Viktiga operatorer och deras \LaTeX -kommandon.

Operator		Kommentar
$+$, $-$	$+$, $-$	
$\backslash times$, $\backslash div$	\times , \div	Multiplikation kan även skrivas ut tydligt med $\backslash cdot$ (\cdot); $\backslash times$ och $\backslash cdot$ har speciella betydelser inom linjär algebra
$\backslash pm$, $\backslash mp$	\pm , \mp	
$\backslash cup$, $\backslash cap$	\cup , \cap	Union/snitt i mängdlära
$\backslash lor$, $\backslash land$	\vee , \wedge	Och/eller i Boolesk algebra
$\backslash partial$	∂	Partiell deriveringsoperator
$\backslash nabla$	∇	Flerdimensionell deriveringsoperator
$\backslash Delta$	Δ	Laplaceoperator

3.2.4. Paranteser och dylikt

Paranteser är mycket viktiga inom matematiken för att visa olika typer av grupper, och besitter dessutom egenskapen att de måste skalas med sin omgivning för att de ska se bra ut. I \LaTeX skapar man sådana grupper av paranteser med kommandona `\left` och `\right`, vilket gör att dessa paranteser automatiskt sträcks ut för att bli lika höga som innehållet i gruppen (vilket visas i [exempel 3.2](#)). Man använder `\left` och `\right` genom att sätta dit parantesen direkt efter kommandot:

```
\left(\frac{1}{2}\right)
```

De paranteser som oftast används listas i [tabell 3.8](#). Värt att notera är att `\left` och `\right` måste användas i par, just eftersom de skapar en grupp. Vill man bara använda en av dem måste man ändå sätta dit den andra på en lämplig plats, men med en punkt efter istället för en parantes. Punkten motsvarar tomrum:

```
\left.\frac{1}{2}\right]
```

Paranteserna måste alltså inte heller matcha typmässigt, även om detta ofta är önskvärt:

```
\left(\frac{1}{2}\right]
```

Att ha omatchade paranteser kan vara önskvärt inom till exempel kvantfysik, där man har bra-ket-notation (se dock notisen om `\mid` på [sida 58](#) och paketet `braket` som förklaras på [sida 110](#)):

```
\left\langle\psi\right\rangle
```

3. Matematik med \LaTeX och \mathcal{AMS}

Tabell 3.8: Fyra vanliga parantestyper i \LaTeX .

Paranteser	Kommentar
<code>\{ \}</code> <code>()</code> <code>[]</code>	<code>{ }</code> <code>()</code> <code>[]</code>
<code>\langle \rangle</code> <code>\rangle</code>	<code>\langle \rangle</code> <code>\rangle</code>
	Används för bland annat mängddefinitioner Används av vissa istället för <code>()</code> , av andra för att visa insättning av en variabel Används för att visa skalärprodukten i till exempel funktionsrum

```
\begin{equation*}
\int\limits_0^1\!\!\left(\dfrac{x^3}{3} +
x^2\right)\mathrm{d}x = \left[\dfrac{x^4}{12} +
\dfrac{x^3}{3}\right]_{x=0}^1 = \dfrac{5}{12}
\end{equation*}
```

$$\int_0^1 \left(\frac{x^3}{3} + x^2 \right) dx = \left[\frac{x^4}{12} + \frac{x^3}{3} \right]_{x=0}^1 = \frac{5}{12}$$

Exempel 3.2: Skalbara paranteser med `\left` och `\right`.

Vertikala streck

Vertikala streck är mycket lika paranteser; man kan även använda dessa genom att skriva exempelvis `\left|` och `\right|` (eller `\left\vert` och `\right\vert`), om man vill. Det finns dock även kommandon `\lvert` och `\rvert` som motsvarar dessa (och som beter sig något bättre typsättningsmässigt), samt `\lVert` och `\rVert` som motsvarar dubbla vertikala streck. Man kan således definiera både absolutbelopps- och normkommandon i \LaTeX ,¹ som sedan kan användas i matematikläget:

```
\DeclarePairedDelimiter\abs{\lvert}{\rvert}
\DeclarePairedDelimiter\norm{\lVert}{\rVert}
% ...
\left( \norm{x}_1 = \sum_{i=1}^n \abs{x_i} \right)
```

Dessutom används det vertikala strecket ibland mitt i block (exempelvis vid definition av mängder, $\emptyset = \{A \mid A \in A\}$) och betecknas då `\mid`. Ett exempel med tidigare nämnda bra-ket-notation kan se ut så här:

```
\left\langle \psi \mid \Psi \right\rangle
```

Dessutom kan det användas som en operator, exempelvis för att visa insättning av en variabel som ett led i integrering ($\int_1^e x^{-1} dx = \ln x|_{x=1}^e = 1$) i vilket fall man istället skriver `\vert`:

```
\ldots = \ln x \vert_{x=1}^e = \ldots
```

3.2.5. Summor och integration

Integraler och summor (och upprepade produkter och så vidare) är mycket enkla att uttrycka med \LaTeX . Med hjälp av kommandot `\sum` (eller `\int`, se även [tabell 3.9](#)) kan man infoga en summa (eller integral) utan gränser. Vill man ha en definit integral, eller en summa med gränser, så använder

¹ Kommandot `\DeclarePairedDelimiter` definieras av paketet `mathtools`

3. Matematik med \LaTeX och \AMSTeX

Tabell 3.9: Sum-, produkt- och integraltecken i \LaTeX .

<code>\sum</code>	\sum	<code>\prod</code>	\prod	<code>\coprod</code>	\coprod
<code>\int</code>	\int	<code>\iint</code>	\iint	<code>\iiint</code>	\iiint
<code>\iiint</code>	\iiint	<code>\idotsint</code>	$\int \cdots \int$	<code>\oint</code>	\oint

man sub- och superskriptskommandona `_` och `^` som tidigare diskuterats på [sida 47](#):

```
\sum_{x \in A} \left( \ldots \right)
```

Detta resulterar oftast i ett korrekt resultat, med gränser över och under tecknet då detta får plats, och med gränser likt vanliga upphöjningar och nedsänkningar annars (till exempel i ekvationer i text, där det är ont om plats i höjdlängd). Ibland gör \LaTeX dock fel. När detta händer får man hin- ta \LaTeX om att gränserna inte typsätts korrekt med hjälp av kommandot `\limits`, som placeras mellan tecknet och gränserna:

```
\sum \limits_{x=1}^{\infty} \frac{1}{x^2}
```

Kommandot `\limits` kan även användas med andra operatorer, till exem- pel `max`, `inf` och `lim`:

```
\lim \limits_{x \rightarrow \infty} \frac{1}{x}
```

Snyggare (och mer korrekta) integraler

Integraler, så som \LaTeX typsätter dem från början, blir ofta felaktiga i det avseendet att avståndet mellan integraltecken och innehåll blir stort, sam- tidigt som avståndet till differentieringsoperatorn (som dessutom inte typ-

3. Matematik med \LaTeX och \mathcal{AMS}

sätts korrekt utan visst krångel) blir för litet. Det första problemet (som i viss mån även finns för summor) kan åtgärdas genom att lägga in ett negativt mellanrum (`\!`) mellan integraltecken och innehåll:

```
\int_0^1\! x^2 dx
```

Det andra problemet kan också lösas genom att lägga till mellanrum (i kombination med att typsätta dx med `\mathrm`), men eftersom det är så ofta man behöver differentieringsoperatorn är det bättre att definiera ett kommando för detta ^{1,2}:

```
\makeatletter
\renewcommand\mathrm[1]{\ensuremath{%
\!;\mathrm{d}\!#1\ifnextchar\mathrm{\!}\!}}
\makeatother
```

Med denna definition kan man mycket enklare typsätta snygga integraler med korrekta differentieringsoperatorer, vilket [exempel 3.3](#) visar. Notera även att exemplet inte använder `\iint` trots att det handlar om en dubbelintegral; detta eftersom varje ”underintegral” har egna gränser. Integraltecknet `\iint` används istället med fördel när man ska visa integrering över exempelvis ett område A , då detta bara är ”en” gräns:

```
\iint_A\! f(x) \mathrm{d}A
```

3.3. Ekvationsomgivningar

Redan på [sida 46](#) diskuterades två av de ekvationsomgivningar $\mathcal{AMS}\LaTeX$ erbjuder (`equation` och `equation*`), och dessa två är fullt tillräckliga för väldigt många tillämpningar. Det finns dock tillfällen då dessa inte räcker

¹ Tack till de Beaudrap (2011) för grunden till detta kommando. ² Detta kommando, och en del andra användbara matematik-kommandon, definieras även av paketet `skmath` som finns tillgängligt på CTAN.

3. Matematik med \LaTeX och \AMS

$\int_0^{\infty} \int_0^1 f(x, y) dy dx$	<pre>\begin{equation*} \int\limits_0^{\infty} \int\limits_0^1 f(x,y) dy dx \end{equation*}</pre>
$\int_0^{\infty} \int_0^1 f(x, y) dy dx$	<pre>\begin{equation*} \int\limits_0^{\infty}\! \int\limits_0^1\! f(x,y) \, \mathrm{d}y \mathrm{d}x \end{equation*}</pre>

Exempel 3.3: Integral med korrekt (underst) och inkorrekt (överst) typsättning. Notera att d inte finns i \LaTeX från början utan måste definieras med den kod som presenteras på [sida 60](#).

till, och därför definierar \AMS även en stor mängd andra omgivning-ar för att lösa de problem som kan uppstå. Det kan handla om att man har en väldigt lång ekvation som måste brytas över flera rader, att man har flera små ekvationer som hör ihop men som bör vara radbrutna, att man vill ha en matris eller något helt annat.

Den här introduktionen kommer bara att presentera några av de viktigas-te omgivningarna; en fullständig lista finns i användarguiden till \amsmath (AMS 1999) och en ordentlig genomgång ges av Voß (2010).

3.3.1. Långa ekvationer med `multline`

Ibland (oroväckande ofta, till och med) stöter man på ekvationer som är alldeles för långa för att få plats på en rad. Hamnar man i en sådan situation är det `multline` (notera avsaknandet av ett `i`) man bör använda. Omgivningen typsätter matematik på flera rader, där den första raden vänsterjusteras, sista raden högerjusteras och rader däremellan centreras — ett exempel kan ses i [ekvation \(3.1\)](#).

3. Matematik med \LaTeX och \AMS

$$\int_0^\infty R(t) dt = -3\sqrt{2\pi}e \operatorname{erfc}\left(\frac{1}{\sqrt{2}}\right) - 3e^2\sqrt{2\pi} \left(e^{5/2} \operatorname{erfc}\left(\frac{3}{\sqrt{2}}\right) - 2 \operatorname{erfc}(\sqrt{2})\right) + 2 \quad (3.1)$$

Rader separeras helt enkelt med radbrytningskommandot `\:`

```
\begin{multline}
\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} = \dots \\
= f'(a)
\end{multline}
```

Precis som `equation` finns `multline` både med och utan stjärna, och på exakt samma sätt har varianten utan stjärna ett ekvationsnummer, vilket den med stjärna inte har. I övrigt finns det inga skillnader mellan de båda. Notera att varje `multline`-omgivning endast får *ett* ekvationsnummer; vill man ha ett per rad ska man istället använda `align`.

3.3.2. Relaterade ekvationer: `align` och `gather`

Ibland kan man ha flera olika, men relaterade ekvationer man vill typsätta i en grupp eller på rad. Detta kan man göra med två olika omgivningar: `align` och `gather` (även dessa har stjärnvarianter utan nummer). **Figur 3.1** visar hur dessa omgivningar ser ut; den enda skillnaden mellan de två är att man med `align` kan placera ekvationer mer exakt; `gather` centrerar helt enkelt bara ekvationerna.

3. Matematik med \LaTeX och \AMS

$\begin{aligned} y_1 &= x^2 + 2x + 1 & (3.2) \\ y_2 &= x^2 - 2x + 1 & (3.3) \\ y_3 &= x^2 - 1 & (3.4) \end{aligned}$	$\begin{aligned} y_1 &= x^2 + 2x + 1 & (3.5a) \\ y_2 &= x^2 - 2x + 1 & (3.5b) \\ y_3 &= x^2 - 1 & (3.5c) \end{aligned}$
--	---

(a) Tre ekvationer typsatta med `align` (b) Tre ekvationer typsatta med `gather`

Figur 3.1: Relaterade ekvationer typsatta med `align`, `gather` och `subequations`.

Nya rader skapas i båda omgivningarna med nyradskommandot `\\`, och i `align` använder man (på liknande sätt som i `tabular`) ett et-tecken (`&`) för att skapa en ”kolumn”. Koden för [figur 3.1a](#) är således som följer:

```
\begin{align}
y_1 &=& x^2 + 2x + 1 \\
y_2 &=& x^2 - 2x + 1 \\
y_3 &=& x^2 - 1 \\
\end{align}
```

I varianterna utan stjärna får varje rad ett eget ekvationsnummer. Detta är helt i sin ordning; `align` och `gather` ska användas för att gruppera ekvationer, och om man istället vill bryta upp en ekvation på flera rader bör man använda `multline`. Man kan dock tvinga \LaTeX att utelämna ekvationsnumret för en specifik rad genom att använda kommandot `\nonumber` precis innan man bryter raden. Man kan även sätta en etikett på en enskild rad genom att placera `\label` på samma sätt.

Underekvationer

Förutom `align` och `gather` visar även [figur 3.1b](#) hur omgivningen `subequations` fungerar. Denna omgivning gör att man kan skapa ”underekvationer” (likt underrubriker), och får bara innehålla en (och endast en) matematikomgivning, inget annat:

```
\begin{subequations}
\begin{align}
y_1 &= x^2 + 2x + 1 \\
y_2 &= x^2 - 2x + 1 \\
y_3 &= x^2 - 1
\end{align}
\end{subequations}
```

3.3.3. Matriser

Matriser byggs upp på ett sätt som liknar tabular, och definieras av de omgivningar som visas i [figur 3.2](#). Dessa måste användas i matematikläge, det vill säga i en annan omgivning som används för att typsätta matematik. Kolumner separeras med `&` och radbrytningar görs med `\\`.

I stora eller repetitiva matriser (till exempel enhetsmatrisen) behöver man inte alltid skriva ut alla element, utan istället använda notation för upprepande element. Detta görs med hjälp av tre kommandon; `\cdots`, `\vdots`

$$\begin{array}{ccccc}
\left\| \begin{matrix} a & b \\ c & d \end{matrix} \right\| & \left\{ \begin{matrix} a & b \\ c & d \end{matrix} \right\} & \left| \begin{matrix} a & b \\ c & d \end{matrix} \right| & \left[\begin{matrix} a & b \\ c & d \end{matrix} \right] & \left(\begin{matrix} a & b \\ c & d \end{matrix} \right) \\
\text{(a) } Vmatrix & \text{(b) } Bmatrix & \text{(c) } vmatrix & \text{(d) } bmatrix & \text{(e) } pmatrix
\end{array}$$

Figur 3.2: Fem av de sex matristyper $\text{\AMS}\text{\LaTeX}$ definierar.

3. Matematik med \LaTeX och \AMS

och `\ddots`, som skapar horisontella, vertikala och diagonala punkter vilka kan användas istället för innehåll i kolumnerna i en matris:

```
\begin{equation*}
\lvert I \rvert = \begin{vmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{vmatrix} = 1
\end{equation*}
```

3.3.4. Olika fall med `cases`

Omgivningen `cases` är mycket användbar då man definierar styckvisa funktioner eller vissa talföljder, till exempel den som behandlas i Collatz problem. Omgivningen liknar både `align` och `matrix`-omgivningarna, men har endast två kolumner (som dock separeras av mer mellanrum än i andra omgivningar). **Exempel 3.4** visar hur `cases` används för att definiera en styckvis funktion (i det här fallet rampfunktionen).










$$R(x) = \begin{cases} x, & x \geq 0; \\ 0, & x < 0 \end{cases} \quad (3.6)$$

```
\begin{equation}
R(x) = \begin{cases}
x, & x \geq 0; \\
0, & x < 0
\end{cases}
\end{equation}
```

Exempel 3.4: Rampfunktionens definition typsatt med hjälp av `cases`.

3. Matematik med \LaTeX och \AMSTeX

Tabell 3.10: De mellanrum \LaTeX definierar. (1 mu = 1/18 em)

Kommando	mu	Exempel
<code>\negthickspace</code>	−5	
<code>\negmedspace</code>	−4	
<code>\"! , \negthinspace</code>	−3	
	0	
<code>\,, \thinspace</code>	3	
<code>\:, \medspace</code>	4	
<code>\;, \thickspace</code>	5	
<code>\quad</code>	18	
<code>\qquad</code>	36	

3.4. Mellanrum

Mellanrum i matematikläge ignoreras som sagt av \LaTeX , som automatiskt sätter in korrekt mellanrum där det behövs. Normalt brukar detta bli mycket bra, men ibland (som i fallet med integraler) kan man behöva göra en del manuella justeringar. Detta kan man göra med de kommandon som ges i [tabell 3.10](#).

Det finns både positiva (vanliga) och negativa mellanrum. De negativa flyttar inte helt oväntat text ”bakåt”, och är användbara när man till exempel vill ångra de mellanrum \LaTeX lägger in.

Vilket mellanrum \LaTeX lägger in beror på vilken typ tecknet är och hur omgivningen ser ut; [tabell 3.11](#) visar de teckentyper som används i matematikläge samt några exempel och mängden mellanrum som tillförs. Notera dock att tecken av typ 2 (binära operatorer) görs om till typ o om de inte har någon text (en operand) till vänster.

3. Matematik med \LaTeX och \AMS

Tabell 3.11: Avstånd före (pre) och efter (post) olika typer av tecken i matematikläge.

Typ	Beskrivning	Avst. (mu)	
		Pre	Post
0	$A0\Phi\infty$	Enkla tecken ("substantiv")	o o
1	$\sum \prod \int$	Prefixoperatorer	o 3
2	$+ \cup \wedge$	Binära operatorer	4 4
3	$= < \subset$	Jämförelse ("verb")	5 5
4	$(\{ \{ \{$	Öppnande avgränsare	o o
5	$\} \} \}$	Stängande avgränsare	o o
6	$., ; !$	Postfix, punktuation	o o

$$p(x) = \left(\frac{1}{2}x^3 + x^2 + x + 1 \right) = 0 \quad (3.7)$$

$$p(x) = \left(\frac{1}{2}x^3 + x^2 + x + 1 \right) = 0 \quad (3.8)$$

```
\begin{multline}
p(x) = \left(\frac{1}{2}x^3 +
x^2 + \right. \left. \right.
\left. x + 1 \right) = 0
\end{multline}
```

```
\begin{multline}
p(x) = \left(\frac{1}{2}x^3 +
x^2 + \right. \left. \right.
\left. x + 1 \right\phantom{\frac{1}{2}}
\left. \frac{1}{2} \right) = 0
\end{multline}
```

Figur 3.3: Problem med `multline` och lösningen med `\vphantom`.

3.4.1. Fantomer

Ibland finns det även tillfällen då man vill ha mellanrum som är *exakt* lika stora som en viss bit typsatt text. Man använder då fantomer. De viktigaste fantomerna är `\hphantom` (ingen höjd, bredd från dess innehåll) och `\vphantom` (ingen bredd, höjd från dess innehåll). Dessa kan vara användbara om man till exempel använder `\multline` för att bryta en ekvation, samtidigt som man använder `\left` och `\right` (som inte fungerar ”runt” radbrytningar).

Figur 3.3 illustrerar problemet och lösningen. Här använder vi `\vphantom`, eftersom vi vill lägga till höjd, och vi fyller den med `\frac{1}{2}` eftersom det är den högsta delen av gruppen vi vill matcha.

3.5. Punkter

Även om \LaTeX definierar kommandon som `\ldots` och `\cdots` för att typsätta punkter så bör man undvika dessa eftersom de inte berättar vilket sammanhang punkterna används i. Det finns många olika konventioner, och för att enkelt kunna byta mellan dessa är det bättre att använda *semantiska* punktkommandon. $\mathcal{AMS}\LaTeX$ definierar fem sådana (**tabell 3.12**) för olika tillfällen. Använd alltid dessa om det är möjligt!

3.6. Enheter med `siunitx`

Fysiker och matematiker (främst fysiker) behöver ofta typsätta enheter i anslutning till sin text, något som inte alltid är helt enkelt eftersom en del konventioner existerar gällande avstånd och så vidare. Paketet `siunitx` ämnar förenkla detta genom att tillhandahålla kommandon som till exempel `\SI`.

3. Matematik med \LaTeX och \AMS

Tabell 3.12: Semantiska punktkommandon definierade av $\text{\AMS}\text{\LaTeX}$.

Kommando	Exempel	Kommentar
<code>\dotsc</code>	A_1, A_2, \dots, A_n	Används i anslutning till kommatecken, det vill säga listor och dylikt (eng. dots <i>with commas</i>)
<code>\dotsb</code>	$A_1 + A_2 + \dots + A_n$	Används i anslutning till binära operatorer, alltså addition, subtraktion, booleska operatorer och så vidare (eng. dots <i>with binary operators</i>)
<code>\dotsm</code>	$A_1 A_2 \dots A_n$	Används i anslutning till multiplikation (eng. dots <i>for multiplication</i>)
<code>\dotsi</code>	$\int_{A_1} \int_{A_2} \dots \int_{A_n}$	Används med integraler, summor och liknande (eng. dots <i>with integrals</i>)
<code>\dotso</code>	$A_1 \dots A_n$	Används då inga andra punkter passar (eng. dots <i>for other situations</i>)

3. Matematik med \LaTeX och \AMS

Paketet definierar tre kommandon (`\num`, `\ang` och `\SI`) som används för att typsätta tal, vinklar och tal med enheter, respektive. Det tal man vill typsätta kan även inkludera förenklande notation så som \pm istället för `\pm`, `e` för att beteckna tiopotenser och så vidare. Även komplexa tal stöds.

Kommandona `\num` och `\ang`, som inte accepterar enheter, kräver endast ett argument medan `\SI` kräver två; talet som ska typsättas och enheten som hör till. Enheterna ges av \LaTeX -kommandon så som `\metre`, `\per`, `\milli` och så vidare (en full lista ges av Wright 2013, sidorna 9–12):

```
\num{12345,60} % Typsätter ett tal
\num{.35e100} % Typsätter ett stort tal
\num{1+-2i}    % Typsätter ett komplext tal
\ang{180}      % Typsätter en vinkel
\SI{15}{\kilogram\metre\per\second\squared} % Tal med enhet
```

Dessa kommandon typsätts då sedan som 12 345,60, $0,35 \cdot 10^{100}$, $1 \pm 2i$, 180° och 15 kg m s^{-2} , respektive.

Som en extra bonus löser `siunitx` även, om man ställt in paketet ordentligt, problemet med tusentals- och decimalseparatorn som \LaTeX ofta ger upphov till (eftersom systemet baseras på engelska konventioner). Detta gör man genom att använda `\sisetup` för att sätta `locale` till `DE`¹:

```
\sisetup{locale=DE}
```

Paketet kan göra väldigt mycket (man kan även definiera sina egna enheter), och det är omöjligt att gå igenom allt i en kort introduktion. Den intresserade hänvisas till `siunitx`-manualen (Wright 2013), som utförligt förklarar hur paketet fungerar och vilka inställningar som kan göras.

¹ Tyska konventioner väljs eftersom paketet inte innehåller inställningar för svenska, och eftersom tyska konventioner är mycket nära de svenska.

Del III.

Figurerna

4. Grafik med L^AT_EX

Grafik, i form av figurer och diagram, kan vara ett mycket bra sätt att illustrera eller beskriva samband, algoritmer och data. Nedan följer ett antal olika tips, inte bara om hur man inkluderar (eller till och med skapar) grafik i L^AT_EX, utan även en inledning som diskuterar vad *bra* grafik är, och hur man på bästa sätt använder grafik.

Den här delen av boken handla således inte uteslutande om L^AT_EX och dess ekosystem, utan även om datavisualisering. Den som främst är intresserad av de L^AT_EX-relaterade bitarna kan därför hoppa över det inledande stycket (även om det är både kort och informativt) och gå direkt på graphicx och PGF/TikZ på [sida 76](#).

4.1. Vad är bra grafik?

Att skapa bra grafik, grafik som på ett effektivt och tydligt sätt kommunicerar den information din data innehåller, är inte så självklart som man kan tro. Det finns oerhört många variabler, från vilken typ av grafik du ska välja (Henry 2012) till hur du ska sortera din data (Carnegie 2010; Tufte 2001, sidan 178) för att förmedla innehållet på absolut bästa sätt. Ibland kan man till och med ifrågasätta om grafik verkligen behövs, eller om en tabell presenterar datan bättre. Tufte, som skrivit många böcker inom ämnet,¹ sammanfattar bra grafik på ett mycket exakt och kortfattat sätt Tufte (2001, sidan 51):

Fulländad grafik består av komplicerade idéer kommunicerade med klarhet, precision och effektivitet.

¹ Tufte's trilogy The Visual Display of Quantitative Information (Tufte 2001), Envisioning Information (Tufte 1990) och Visual Explanations (Tufte 1997) är väl värd att läsa för den som är intresserad av grafik och hur man presenterar data.

4. Grafik med \LaTeX

Precisionen bör vara det enklaste att uppnå. Bland annat handlar det om att inte (medvetet eller omedvetet, till exempel genom inkorrekta relativa skalor eller 3D-effekter) ljuga om sin data (Tufte 2001, sidorna 57 ff.), vilket man undviker genom att förse sina figurer med rättvisande, relevanta axlar och att inte ta data ur sitt sammanhang (Tufte 2001, sidorna 75 f.), vilket bör vara enkelt att undvika genom att inte klippa bort för mycket data. Även korrekta och relevanta figurtexter hjälper precisionen.

Klarheten är något mer komplicerad att uppnå. Här handlar det bland annat om att undvika överflödigt information (eng. *chart junk*), men även om att använda rätt typ av diagram, bra färgscheman (Tufte 1997, sidorna 75–77) och rimliga längdskalor (Tufte 1997, sidorna 20–23). Även proportioner kan spela roll; det är till exempel lättare att se förändringar i ett ”långt”, horisontellt linjediagram än i ett ”högt”, vertikalt dito (Tufte 2001, sidorna 186 f.).

Effektiviteten är svårast att uppnå. Med hög datatäthet (Tufte 2001, sidan 162), det vill säga många datapunkter per areaenhet (till en rimligt gräns, så att klarheten inte försummas), är ett sätt att uppnå effektivitet. Att dessutom se till att så mycket av figuren som möjlig förmedlar faktisk data, genom att till exempel göra figuraxlar mer spartanska eller utelämna rutsystem, kan också öka grafikens effektivitet (Tufte 2001, sidorna 123 ff.). Effektiviteten är förmodligen den del av ”fulländad” grafik som är svårast att uppnå i matematisk programvara.

4.1.1. Allmänna tips

Även om datavisualisering är ett stort och inte helt trivialt område, och därmed kanske överkurs för de som inte skriver böcker eller långa rapporter, så bör man hålla någon form av miniminivå för att åtminstone få fram sitt budskap. Man skulle kunna sammanfatta en sådan lägstanivå i ett par enkla punkter:

- *Se till att axlar i figurerna är korrekta och informativa.* Detta innebär att axlarna bör vara tydligt numrerade, att det framgår om en an-

4. Grafik med \LaTeX

nolunda skala (till exempel en logaritmisk sådan) används, och att axlarna har beskrivande etiketter (med enheter).

- *Se till att välja lämpliga färgscheman.* Ta till exempel färgblindhet eller kontrast i svartvita utskrifter i beaktande. För enkla linjediagram och dylikt är det ganska enkelt, men i andra fall kan det vara svårt att välja ett färgschema¹ som på ett bra sätt förmedlar informationen (Borland och Taylor II 2007; Moreland 2009).
- *Undvik otydliga eller missvisande diagram.* Tredimensionella linjediagram kan vara oerhört svåra att läsa, och cirkeldiagram lider av liknande problem, till exempel. Om du illustrerar skillnader med hjälp av figurer av olika storlek, kom då ihåg att *arean* är det som läses av, inte höjden.
- *Undvik onödiga extradetaljer.* Mönster, fyllda områden, skuggor, 3D-effekter och liknande tillför sannolikt ingen information, och syftar mest till att förvirra läsaren.

4.1.2. Programspecifika tips: Bra R-figurer med ggplot2

Riktlinjerna och tipsen som introducerats ovan kan vara mer eller mindre komplicerade att implementera när man ritar figurer i programvara. Vissa program skapar redan bra och informativ grafik utan extra arbete, men annan programvara kan kräva mer arbete. Ett program av den senare typen är R.

Standardgrafiken i R är överlag ganska bra när det gäller statistisk analys, men bristande på andra områden. På många sätt är det dessutom svårt att göra lite mer involverade saker som att ändra färger, sätta ordentliga axlar, enheter och så vidare. R-paketet ggplot2 skapades för att åtgärda detta, och skapar snyggare grafer på enklare sätt.

Paketet är intuitivt och baserat på logiska kopplingar mellan data och grafiska element, vilket gör att man kan bygga grafik genom att tänka på datans innebörd och vad man vill kommunicera, istället för själva grafiken.

¹ I sådana fall är Colorbrewer, tillgängligt på <http://colorbrewer2.org>, ett mycket bra verktyg.

4. Grafik med \LaTeX

En bra sammanfattning av paketet, som gör sig bäst på originalspråket, är ”makes doing the right thing easy, while keeping harder things possible”.

Att installera och använda ggplot2 är enkelt. Efter att ha installerat paketet genom att köra kommandot `install.packages("ggplot2")` i R använder man det genom att inkludera paketet och använda funktionen `ggplot` i kombination med någon av paketets många plot-funktioner. Man kan även kombinera olika grafer:

```
require(ggplot2)
# Exempeldata
df <- data.frame(
+   trt = factor(c(1, 1, 2, 2)),
+   resp = c(1, 5, 3, 4),

+   group = factor(c(1, 2, 1, 2)),
+   se = c(0.1, 0.3, 0.3, 0.2)
+)
df2 <- df[c(1,3),]
# Toppen/botten på errorbars
limits <- aes(ymax = resp + se, ymin=resp - se)
dodge <- position_dodge(width=0.9)
# Plotta!
p <- ggplot(df, aes(fill=group, y=resp, x=trt))
p + geom_bar(position=dodge) +
  + geom_errorbar(limits, position=dodge, width=0.25)
```

En mer utförlig referens över alla kommandon i ggplot2 finns på internet,¹ och en grundlig genomgång ges av Wickham (2009).

¹ <http://had.co.nz/ggplot2/>

4.2. Inkludera grafik

Att kunna inkludera grafik av olika slag i sitt dokument kan verka så vitalt och grundläggande att det rimligtvis inte bör finnas många olika möjligheter. Så är dock inte fallet. Under den långa tid som \LaTeX har existerat (och utvecklats) har en rad olika metoder för att inkludera grafik, alla med specifika för- och nackdelar, utvecklats för systemet.

I grunden kan man dock dela upp dessa metoder i två undergrupper: de paket som inkluderar grafik som skapats av externa program (till exempel bilder i PDF-, PNG- eller JPEG-format) och de som använder funktionalitet i den underliggande \TeX -kompilatorn för att skapa grafik.

I den första kategorin finns i princip bara ett alternativ — `graphicx`. Den senare kategorin har ett antal kandidater, men för $\pdf\LaTeX$ och $X\TeX$ finns det en mycket stark kandidat: PGF/TikZ. Båda dessa, samt deras för- och nackdelar, diskuteras nedan.

4.2.1. Paketet `graphicx`

Den metod som oftast används, eftersom den kräver kortare kompileringstider och mindre direkt ansträngning, är att inkludera redan existerande bilder i PDF- eller PNG-format med hjälp av paketet `graphicx`. Paketet har ett mycket enkelt gränssnitt, och har egentligen bara ett kommando av intresse: `\includegraphics`.

Med hjälp av detta kommando kan man inkludera grafik av format som är kompatibla med den \TeX -kompilator som används. I fallet $\pdf\LaTeX$ är dessa format PNG, PDF och JPEG. Kommandot tar ett valfritt argument, en lista av nycklar, som används för att på olika sätt skala om bilden som inkluderas, och ett obligatoriskt argument som berättar var bilden finns:

```
\includegraphics[width=\textwidth]{filnamn}
```

Fler sådana nycklar listas av Carlisle (2005, sidorna 8 f.).

4. Grafik med \LaTeX

Fördelen med denna metod är att den i princip *alltid* fungerar, eftersom nästan alla verktyg kan exportera PNG- eller PDF-filer (och i de fall detta inte går kan man oftast konvertera till PNG eller i västa fall använda skärmdumpar). Metoden är såklart också bra för att inkludera fotografier i JPEG-format, till exempel bilder på experimentuppställningar.

Nackdelen är att dessa format inte är vektorbaserade. Detta innebär att om du behöver skala om bilden i efterhand, så kan grafiken bli suddig eller otydlig. För till exempel linjediagram och liknande kan dessutom texten på axlarna bli för liten eller otydlig, vilket såklart gör att informationen inte kommuniceras lika effektivt.

Sammanfattat kan man säga att `graphicx` är utmärkt för att inkludera JPEG-bilder (dvs. fotografier), medans diagram, grafer och liknande oftast blir bättre med PGF/TikZ eller pgfplots.

4.2.2. Rita med PGF/TikZ

En metod som ofta ger bättre resultat, men som är lite mer avancerad att använda (både eftersom den ger längre kompileringstider och eftersom den kräver mer arbete av användaren) är att använda PGF/TikZ för att rita den grafik som behövs.

PGF/TikZ är ett (mycket stort) paket som används för att rita vektorgrafik direkt i \LaTeX . Paketet har en omfattande manual (Tantau 2010) som innehåller många väldokumenterade exempel. Dessutom ges en bra, kortfattad introduktion av Mertz och Slough (2007). Använder man ett program som kan exportera PGF/TikZ-kod finns det oftast ingen anledning att inte göra det, och i vissa fall (så som för enkla figurer över uppställningar eller mekaniska figurer) kan det vara lättare att skriva PGF/TikZ-kod än att rita figuren i ett externt program. CTAN-katalogen innehåller ett stort antal tilläggspaket till PGF/TikZ,¹ bland annat `circuitikz` (Redaelli 2012) som används för att rita kopplingsscheman, flowchart (Robson 2013) som gör det lättare att rita flödesscheman, `pgfgantt` (Skala 2013) för Gantt-scheman och `tikz-timing` (Scharrer 2011) som ritat timing-diagram.

¹ <http://www.ctan.org/tex-archive/graphics/pgf/contrib>

4. Grafik med \LaTeX

Utöver de exempel som finns i manualen går det även att hitta mycket material på internet.¹ Ska man bara rita enkla grafer finns dessutom det lite enklare paketet `pgfplots`, som använder `PGF/TikZ` internt (och som diskuteras nedan).

Fördelen med `PGF/TikZ` ligger främst i det faktum att grafiken är vektorbaserad, och därför ser lika bra ut oavsett hur mycket den skalas om, samt att text i figuren behandlas av \LaTeX och därför kommer visas i rätt typsnitt och vettiga storlekar vilket ökar läsbarheten i figuren.

Den främsta nackdelen är att `PGF/TikZ`-figurer ofta tar en stund att kompilera, och att det inte alltid är möjligt att exportera figurer i det formatet. `PGF/TikZ`-figurer är alltså att föredra för diagram, grafer och liknande, men går inte att använda för till exempel fotografier.

Plotta med `pgfplots`

Paketet `pgfplots` kan sägas vara ett tillägg till `PGF/TikZ` som gör det lättare att rita vanliga diagram (stapel- eller linjediagram, till exempel) med hjälp av `PGF/TikZ`. Bland annat kan `pgfplots` läsa in data från externa filer och automatiskt omvandla den till figurer.

Paketet har en omfattande manual (Feuersänger 2013a), som detaljerat beskriver hur man använder paketet för att rita olika sorters diagram. Bland annat använder `matlab2tikz` — som diskuteras på [sida 80](#) — funktionalitet från `pgfplots`. Det finns även ett påbyggnadspaket, `pgfplotstable` (Feuersänger 2013b), som kan läsa CSV-liknande filer och använda `pgfplots` för att rita diagram (och tabeller med booktabs) utifrån dessa data.

¹ Bland annat finns galleriet `TikZExample.net`² som innehåller många exempel på `PGF/TikZ`-kod, och taggen `tikz-pgf` på `TikZ.SE`³ där man kan få svar på många frågor. ² <http://www.texample.net> ³ <http://tex.stackexchange.com/questions/tagged/tikz-pgf>

4.3. Exportera grafik eller data

När man väl skapat sin grafik i det program man arbetar med och är nöjd med denna måste man exportera den till ett format som kan användas med \LaTeX . Använder man $\pdf\LaTeX$ (vilket man bör) så stöds formaten JPEG, PNG och PDF. Av dessa lämpar sig JPEG bäst till fotografier, medan PNG och PDF lämpar sig väl till figurer och grafer. Generellt sett är PDF att föredra då formatet är vektorbaserat och därför kan skalas både upp och ner utan att man förlorar någon kvalitet.

Om möjligheten finns är det bästa dock att exportera sina figurer som PGF/TikZ-kod, vilket innebär att \LaTeX kommer att rendera dem och att man därför har något större frihet när det gäller finjustering av figuren i efterhand, och även att axlar och dylikt typsätts på samma sätt som dokumentets brödtext vilket ger ett enhetligt intryck. Ett alternativ till att exportera hela figurer som PGF/TikZ-kod är att exportera datan i ett format som kan läsas av pgfplots, och sedan använda det paketet för att rita figurerna.

Det finns många (matematiska) programvaror, och det är orimligt att inkludera instruktioner till alla i detta kapitel. Jag har därför valt att beskriva hur man enklast exporterar figurer som PGF/TikZ-kod eller i PDF-format (i nödfall PNG) från två vanliga programvaror: R och MATLAB®.

4.3.1. Exportera i PGF/TikZ-format

PGF/TikZ brukar inte vara ett format som normalt stöds av kommersiella (eller fria) programvaror, och därför är funktionen ofta implementerad som någon form av tillägg. Lyckligtvis brukar dessa vara öppna och fria att använda för vem som helst.

Från R med tikzDevice

R-paketet `tikzDevice` (Sharpsteen och Bracken 2012) gör det möjligt att i R exportera figurer som PGF/TikZ-kod. Det kan som många andra R-

4. Grafik med \LaTeX

paket hittas på CRAN och installeras enklast med R-kommandot `install`:

```
install.packages("tikzDevice")
```

Man använder sedan kommandot `tikz` för att berätta för R att grafiken ska skrivas som PGF/TikZ-kod, och filen som skapas kan sedan inkluderas i \LaTeX -dokument med kommandot `\input`. [Exempel 4.1](#) visar resultatet av detta.

Från MATLAB[®] med `matlab2tikz`

Då MATLAB[®], i likhet med R, inte har något inbyggt verktyg för att exportera figurer i PGF/TikZ-format får man även i det fallet förlita sig på öppen programvara från tredje part. I fallet MATLAB[®] finns det utmärkta `matlab2tikz`,¹ som implementerats som en enkel funktion som anropas direkt efter det att man skapat en figur — användningen framgår tydligt av [exempel 4.2](#).

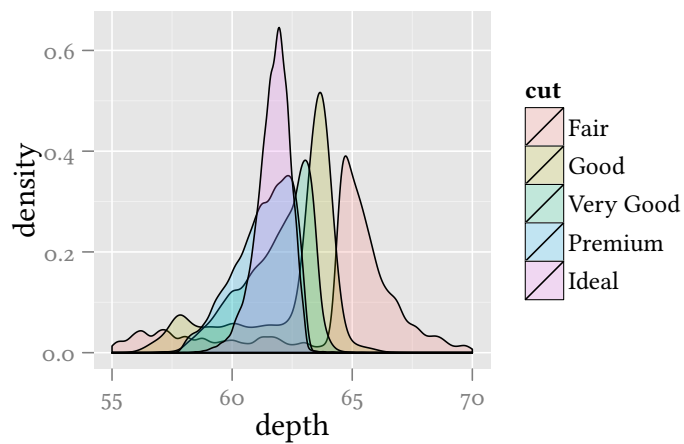
Eftersom `matlab2tikz` är skriven i MATLAB[®]-kod räcker det i teorin med att kopiera dess `.m`-filer till katalogen MATLAB[®]-skripten man skriver ligger i, men det kan vara fördelaktigt att placera filerna i en av de kataloger MATLAB[®] alltid läser in (The Mathworks, Inc. 2013a).

4.3.2. Exportera i PDF eller PNG-format

I de fall då det inte finns något enkelt sätt att exportera figurer som PGF/TikZ-kod, eller i de fall då det är olämpligt (ger orimligt långa kompileringstider, till exempel) är PDF och PNG relativt bra alternativ. Båda dessa format är oftast rasterbaserade, men med PDF finns i alla fall möjligheten (och den utnyttjas ibland) att figurerna exporteras som vektorgrafik.

¹ <https://github.com/nschloe/matlab2tikz>

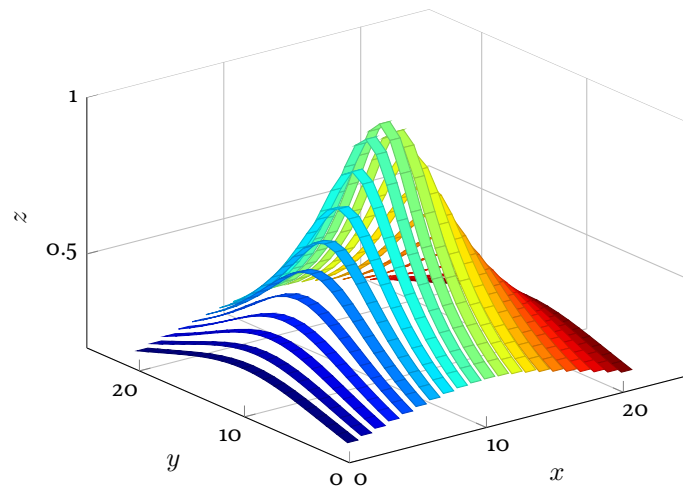
4. Grafik med \LaTeX



```
require(tikzDevice)
# width och height specificeras i tum
tikz('figur.tex', width=4, height=2.5)
# exempel med ggplot2, som diskuteras på sida 74
require(ggplot2)
ggplot(diamonds, aes(depth, fill = cut)) +
  + geom_density(alpha = 0.2) + xlim(55, 70)
# stäng TikZ-filen
dev.off()
```

Exempel 4.1: R-koden nederst genererar den PGF/TikZ-bild som syns överst.

4. Grafik med \LaTeX



```
% skapa lite exempeldata
[x, y] = meshgrid(-2:0.2:2,-2:0.2:2);
R = (1./((x+y).^2+(y-x).^2+1).^(1/2));
% plotta figuren
ribbon(R);
xlabel('$x$'); ylabel('$y$'); zlabel('$z$');
% spara figuren i tikz-format
matlab2tikz('figur.tex', 'width', '8cm', 'height', '6cm');
```

Exempel 4.2: MATLAB[®]-koden nederst genererar den PGF/TikZ-bild som syns överst.

4. Grafik med \LaTeX

Från R till PDF

R har ett antal inbyggda så kallade enheter (eng. *device*) som kan användas för att rita diagram, grafer och liknande, bland annat för EPS, PNG och SVG (The R Core Team 2013, sidorna 675–676). Enheten vi är mest intresserade av är den som producerar PDF-filer. Denna används enkelt genom att köra kommandot `pdf`, som tar samma parametrar som `tikz`-kommandot som beskrevs tidigare. PDF-filen som genereras kan sedan enkelt inkluderas i ett \LaTeX -dokument med `\includegraphics`. **Exempel 4.3** visar hur det kan se ut.

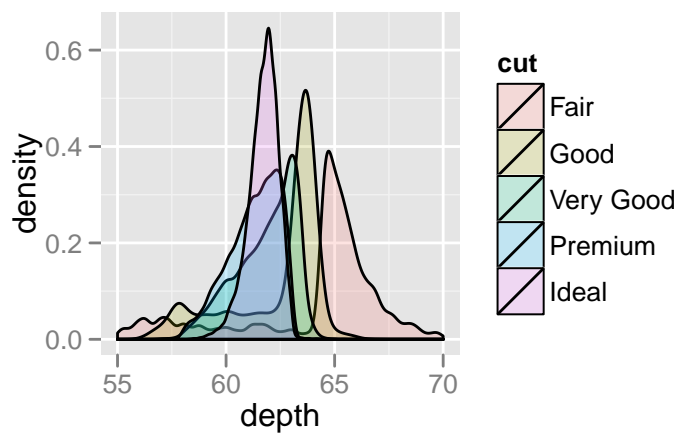
Från MATLAB[®] till PNG

I MATLAB[®] finns ett par olika sätt att exportera figurer, men det enklaste sättet att spara figurer programmatiskt är med hjälp av funktionen `print` (The Mathworks, Inc. 2013b). Funktionen kan, trots sitt namn, spara figurer till ett antal olika format (tyvärr inga vektorbaserade), där det som oftast fungerar bäst är PNG. **Exempel 4.4** visar hur detta kan se ut i praktiken.

I sitt grundutförande lägger MATLAB[®] till ganska stora marginaler till sina figurer. Det är praktiskt när de visas på skärmen, eftersom det ger lite luft mellan innehållet och fönstrets kant, men det är mest till besvär när man ska inkludera figuren i \LaTeX . Lyckligtvis är detta problemet lätt att lösa (Robertson 2010) och man kan till exempel använda funktionen `figuresize.m`,¹ som är fritt tillgänglig under BSD-licensen.

¹ <https://github.com/wspr/matlabpkg/blob/master/figuresize.m>

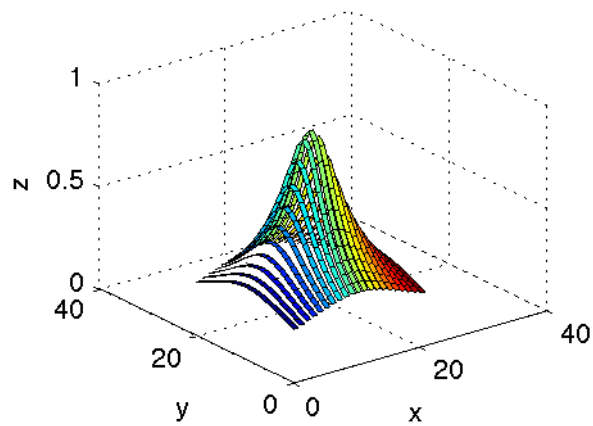
4. Grafik med \LaTeX



```
# width och height specificeras i tum
pdf('figur.tex', width=4, height=2.5)
# exempel med ggplot2, som diskuteras på sida 74
require(ggplot2)
ggplot(diamonds, aes(depth, fill = cut)) +
  + geom_density(alpha = 0.2) + xlim(55, 70)
# stäng PDF-filen
dev.off()
```

Exempel 4.3: R-koden nederst genererar den PDF-bild som syns överst.

4. Grafik med \LaTeX



```
% skapa lite exempeldata
[x, y] = meshgrid(-2:0.2:2,-2:0.2:2);
R = (1./((x+y).^2+(y-x).^2+1).^(1/2));
% plotta figuren
ribbon(R);
xlabel('x'); ylabel('y'); zlabel('z');
% skala om och spara figuren i PNG-format
figure(8, 6, 'centimeters');
print('-dpng', 'figur.png');
```

Exempel 4.4: MATLAB[®]-koden nederst genererar den PNG-bild som syns överst.

Del IV.

Referenserna

5. Referenser med biblatex

En stor fördel med \LaTeX är att man kan automatisera hanteringen av till exempel referenser. Paketet biblatex, tillsammans med programmet biber,¹ gör detta mycket enkelt genom att låta dig samla alla dina referenser i en extern databas, som man sedan läser in och hänvisar till. Denna databasen lagras i ett textbaserat format, så den går att hantera med samma program du skriver \LaTeX -kod i, eller i ett specialiserat program.²

Användningen av biblatex-biber (där biblatex är det \LaTeX -paket som hanterar typsättningen, och biber är ett externt program som behandlar och sorterar referenserna) kan delas upp i några olika delar. Till att börja med kommer vi att diskutera formatet av själva referensdatabasen. Därefter diskuteras paketet biblatex, några av dess viktigaste inställningar och stilar, och till sist sammanfattas användningen av de två tillsammans, speciellt hur man refererar och genererar referenslistan.

5.1. Referensdatabaser

Databaserna biblatex använder är, som nämndes tidigare, helt vanliga textfiler. Dessa innehåller ett antal block, ett per referens i databasen, som i sin tur innehåller fält med information om varje referens. Ett typiskt block innehåller en nyckel (som används när man sedan i \LaTeX -dokumentet refererar till referensen), en titel, en författare och ett årtal. Beroende på vilken

¹ Tekniskt sett kan även BibTeX användas tillsammans med biblatex, men det finns egentligen ingen anledning att göra det. Den som är intresserad av BibTeX, det gamla systemet för att hantera referenser, hänvisas till introduktionen i Fenn (2006) samt BibTeX-manualen (Patashnik 1988). ² Exempel på sådana är *JabRef*, *BibDesk* och *refbase*.

5. Referenser med biblatex

```
@Article{Friedman01,
  author = {Friedman, James-W. and Mezzetti, Claudio},
  title = {Learning in Games by Random Sampling},
  journaltitle = {Journal of Economic Theory},
  date = {2001-05},
  volume = 98,
  number = 1,
  doi = {10.1006/jeth.2000.2694},
  url = {http://www.idealibrary.com/links/doi/10.1006/jeth.2000.2694},
  hyphenation = {english}
}
```

Exempel 5.1: En enkel exempelreferens ur en referensdatabas.

typ av referens (bok, artikel, och så vidare) det handlar om kan även andra fält finnas med.

Exempel 5.1 visar ett block ur en referensdatabas. Blocket inleds med en blocktyp (`@article`), som indikerar vilken typ referensen har, och den nyckel som används när man refererar till källan (`Friedman01`). Därefter följer ett antal självförklarande fält som lagrar information om referensen, som i det här fallet är en artikel ur en tidsskrift. En del av dessa fält är obligatoriska (bland annat titel och författare), medan andra är frivilliga (till exempel `month` och `doi`). **Tabell 5.1** listar de några blocktyper och en del av de obligatoriska/frivilliga fält som hör till. En fullständig förteckning finns i Lehman m. fl. (2013, sidorna 7 ff.).

Även om formatet på databaserna är enkelt att redigera för hand kan det vara skönt att använda något program som gör det enklare. Några sådana program, bland annat *JabRef*¹ (open-source, finns tillgängligt för alla vanliga plattformar) och *BibDesk*² (för Mac OS X), nämndes lite kort på **sidan 87**. Det finns såklart många fler, både kommersiella (till exempel *Mendeley*) och open-source-baserade, men att lista alla är inte denna bokens syfte.

¹ <http://jabref.sourceforge.net/> ² <http://bibdesk.sourceforge.net/>

5. Referenser med biblatex

Tabell 5.1: Några av referenstyperna i biblatex och en del av deras tillhörande fält. Både typnamn och fältnamn är relativt självförklarande, men se Lehman m. fl. (2013, sidorna 7 ff.) för mer information. Nyckel: + obligatoriskt, + valfritt fält.

Blocktyp	author	booktitle	chapter	date ^a	doi	edition	editor	howpublished	hyphenation	institution	isbn	issue	journaltitle	language	location	month ^a	note	organization	pages	publisher	series	subtitle	title	type	url	urldate	version	volume	volumes	year ^a
@article	+			+	+		+		+			+	+		+	+		+		+	+	+	+	+	+	+	+	+	+	
@book	+		+	+	+	+	+		+		+			+	+		+		+		+	+	+	+	+	+	+	+	+	
@mvbook	+			+	+	+	+		+		+			+	+					+	+	+	+	+	+	+		+	+	
@inbook	+	+	+	+	+	+	+		+		+			+	+		+		+		+	+	+	+	+	+	+	+	+	
@bookinbook	+	+	+	+	+	+	+		+		+			+	+			+		+	+	+	+	+	+	+	+	+	+	
@suppbook	+	+	+	+	+	+	+		+		+			+	+		+		+		+	+	+	+	+	+	+	+	+	
@booklet ^b	+		+	+	+		+	+					+	+		+		+				+	+	+	+	+		+	+	
@collection				+	+	+	+		+		+			+	+			+		+	+	+	+	+	+	+	+	+	+	
@mvcollection					+	+	+	+		+				+	+		+				+	+	+	+	+	+		+	+	
@incollection	+	+	+	+	+	+	+		+		+			+	+		+		+		+	+	+	+	+	+	+	+	+	
@suppcollection	+	+	+	+	+	+	+		+		+			+	+		+		+		+	+	+	+	+	+	+	+	+	
@manual ^b	+		+	+	+	+	+		+		+			+	+		+	+	+	+	+	+	+	+	+	+	+	+	+	
@misc ^b	+			+	+		+	+	+				+	+	+	+	+					+	+	+	+	+	+	+	+	
@online ^b	+			+			+	+	+					+	+	+	+					+	+	+	+	+	+	+	+	
@patent	+			+	+			+						+	+	+	+					+	+	+	+	+	+	+	+	
@periodical				+	+		+	+	+		+		+		+	+				+	+	+	+		+	+	+	+	+	
@suppperiodical	+			+	+		+	+	+		+	+	+	+	+		+	+	+	+	+	+	+	+	+	+	+	+	+	
@proceedings			+	+	+		+	+	+		+			+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
@mvproceedings				+	+		+	+	+					+	+	+	+			+	+	+	+	+	+	+		+	+	
@inproceedings	+	+	+	+	+		+	+	+		+			+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
@reference				+	+	+	+		+		+			+	+		+		+		+	+	+	+	+	+	+	+	+	
@mvreference				+	+	+	+		+		+			+	+		+			+	+	+	+	+	+	+		+	+	
@inreference	+	+	+	+	+	+	+		+		+			+	+		+		+	+	+	+	+	+	+	+	+	+	+	
@report	+		+	+	+			+	+					+	+	+	+	+		+		+	+	+	+	+	+	+	+	
@thesis	+		+	+	+			+	+	+				+	+	+	+	+		+		+	+	+	+	+	+	+	+	
@unpublished	+		+				+	+	+	+				+	+	+	+				+	+	+	+	+			+	+	

^a Antingen month/year (där month är valfri) eller date ska specificeras, ej båda två.

^b Antingen author eller editor ska specificeras, ej båda två.

5. Referenser med biblatex

Många webbaserade sökverktyg för akademiska publikationer, till exempel Scopus,¹ gör det möjligt att enkelt exportera information om de artiklar man refererar till i det format \LaTeX och biblatex använder, och dessa kan sedan läggas till direkt i den egna referensdatabasen. Detta är mycket praktiskt i större projekt så som kandidatuppsatser.

5.2. Stilar och inställningar i biblatex

Nästa steg efter att ha skapat en referensdatabas är att inkludera paketet biblatex. Paketet är gjort för att kunna vara relativt generellt, och har därför en uppsjö olika inställningar och bibliografistilar att välja mellan, alla dokumenterade i manualen (Lehman m. fl. 2013).

5.2.1. Vanliga inställningar

Eftersom biblatex är så omfattande är det omöjligt att gå igenom alla de inställningar som behandlas av manualen (Lehman m. fl. 2013). Det finns dock ett par inställningar som är viktiga att hålla koll på eftersom de berör saker man ofta vill ändra på.

Den första inställningen man alltid bör använda är `style`. Denna inställning berättar vilken bibliografistil biblatex ska använda i dokumentet, där `numeric` är standardvärdet. Bibliografistilarna behandlas lite mer utförligt senare i denna del.

Nästa viktiga inställning är `backend`, som kontrollerar vilket program biblatex ska använda för att processera referenserna (dvs. vilket program som måste köras efter första \LaTeX -rundan). Standardvärdet är `biber`, och det finns nästan aldrig någon anledning att ändra på detta. Det kan dock vara värt att komma ihåg att just `biber` används. Den intresserade kan även läsa `biber`-manualen (Kime och Charette 2013, sidorna 3 ff.) för lite kort historik kring `biber` och anledningen till att just `biber` bör användas.

¹ <http://www.scopus.com/home.url>

5. Referenser med biblatex

Utöver dessa finns inställningar för hur referenser sorteras, hur många namn som visas, vilket datumformat som ska användas och liknande, men dessa sätts ofta av bibliografistilen. Det finns även stilspecifika inställningar (`isbn`, `url`, `doi` och `eprint`) som kontrollerar om några av de tillhörande fälten ska döljas från referenslistan.

5.2.2. Bibliografistilar

Bibliografistilarna kontrollerar dels utseendet på själva referenslistan men även hur de kommandon som används för att referera (`\textcite`, `\parencite` och `\footcite`) beter sig. Figur 5.1 till 5.2 visar ett antal olika bibliografistilar, några som följer med biblatex och några som följer med andra paket eller samlingar.

En bibliografistil laddas i biblatex genom inställningen `style`, och man kan till exempel ladda den alfabetiska stilen som visas i figur 5.1a på följande sätt:

```
\usepackage[style=alphabetic]{biblatex}
```

Figur 5.1 på sida 92 visar som sagt de bibliografistilar som inkluderas med biblatex. Vanligt inom matematik och fysik är att man använder något liknande `numeric` (figur 5.1d), men på Chalmers använder man istället författare-årtal (figur 5.1c), eller om möjligt Chicago-stilen (som visas i figur 5.2c och implementeras av ett eget paket, `biblatex-chicago`, som diskuteras senare i detta kapitel). Många fält och publikationer har egna specifika stilar, varav de flesta stora publikationer har biblatex-stilar — ett urval av sådana stilar visas i figur 5.2 på sida 93, men den intresserade hänvisas till CTAN-katalogen,¹ där fler stilar listas.

¹ <http://www.ctan.org/tex-archive/macros/latex/contrib/biblatex-contrib>

5. Referenser med biblatex

Referera direkt till Friedman och Mezzetti [FM01] eller indirekt till en rapport [FM01].

Referenser

[FM01] James W. Friedman och Claudio Mezzetti. "Learning in Games by Random Sampling". I: *Journal of Economic Theory* 98.1 (maj 2001). DOI: 10.1006/jeth.2000.2694. URL: <http://www.idealibrary.com/links/doi/10.1006/jeth.2000.2694>.

(a) Alfabetisk stil (*alphabetic*)

Referera direkt till Friedman och Mezzetti ("Learning in Games by Random Sampling") eller indirekt till en rapport (Friedman och Mezzetti, "Learning in Games by Random Sampling"). Man kan även referera med fotnoter^a.

Referenser

Friedman, James W. och Claudio Mezzetti. "Learning in Games by Random Sampling". I: *Journal of Economic Theory* 98.1 (maj 2001). DOI: 10.1006/jeth.2000.2694. URL: <http://www.idealibrary.com/links/doi/10.1006/jeth.2000.2694>.

^aFriedman och Mezzetti, "Learning in Games by Random Sampling".

(b) Författare-titel-stil (*authortitle*)

Referera direkt till Friedman och Mezzetti (2001) eller indirekt till en rapport (Friedman och Mezzetti 2001). Man kan även referera med fotnoter^a.

Referenser

Friedman, James W. och Claudio Mezzetti (2001). "Learning in Games by Random Sampling". I: *Journal of Economic Theory* 98.1. DOI: 10.1006/jeth.2000.2694. URL: <http://www.idealibrary.com/links/doi/10.1006/jeth.2000.2694>.

^aFriedman och Mezzetti 2001.

(c) Författare-årtal-stil (*authoryear*)

Referera direkt till Friedman och Mezzetti [1] eller indirekt till en rapport [1].

Referenser

[1] James W. Friedman och Claudio Mezzetti. "Learning in Games by Random Sampling". I: *Journal of Economic Theory* 98.1 (maj 2001). DOI: 10.1006/jeth.2000.2694. URL: <http://www.idealibrary.com/links/doi/10.1006/jeth.2000.2694>.

(d) Numerisk stil (*numeric*)

Figur 5.1: Fyra av de standardstilar som följer med biblatex.

5. Referenser med biblatex

Referera direkt till Friedman och Mezzetti (2001) eller indirekt till en rapport (Friedman & Mezzetti, 2001). Man kan även referera med fotnoter^a.

Referenser

Friedman, J. & Mezzetti, C. (2001 maj). Learning in games by random sampling. *Journal of Economic Theory*, 98(1). doi:10.1006/jeth.2000.2694

^aFriedman och Mezzetti, 2001.

(a) American Psychological Association-stil (apa)

Referera direkt till Friedman och Mezzetti [1] eller indirekt till en rapport [1].

Referenser

(1) Friedman, J.; Mezzetti, C. *Journal of Economic Theory* maj 2001, 98, DOI: 10.1006/jeth.2000.2694.

(b) American Chemical Society-stil (chem-acs)

Referera direkt till Friedman och Mezzetti (2001) eller indirekt till en rapport (Friedman och Mezzetti 2001). Man kan även referera med fotnoter^a.

Referenser

Friedman, James W., och Claudio Mezzetti. 2001. "Learning in Games by Random Sampling". *Journal of Economic Theory* 98, nummer 1 (). doi:10.1006/jeth.2000.2694. <http://www.idealibrary.com/links/doi/10.1006/jeth.2000.2694>.

a. Friedman och Mezzetti 2001.

(c) Chicago-stil (biblatex-chicago)

Med IEEE-stilen refererar man alltid indirekt [1].

Referenser

[1] J. Friedman och C. Mezzetti, "Learning in games by random sampling", *Journal of Economic Theory*, vol. 98, nr 1, maj 2001. DOI: 10.1006/jeth.2000.2694. URL: <http://www.idealibrary.com/links/doi/10.1006/jeth.2000.2694>.

(d) Institute of Electrical and Electronics Engineers-stil (ieee)

Figur 5.2: Några av de bibliografistilar som definieras av olika paket och inte följer med biblatex.

5. Referenser med biblatex

Chicago-stilen med biblatex-chicago

Chicago-stilen (Chicago Editorial Staff 2010) är väldigt omfattande (även om man bortser från de delar som inte relaterar till referenslistor), och av tekniska skäl finns den därför inte implementerad som en enkel biblatex-stil. Istället implementeras stilen som ett paket, biblatex-chicago, som laddar biblatex automatiskt.

Paketets manual (Fussner 2013) förklarar i stora drag hur det är tänkt att användas, men i princip gäller samma inställningar som för biblatex bortsett från bibliografistilar. Istället för alla de stilar som finns tillgängliga för biblatex använder sig biblatex-chicago endast av två: fotnoter (notes) och författare-årtal (authordate, den stil som rekommenderas av Chalmers och som syns i [figur 5.2c](#)). För att använda paketet istället för någon annan biblatex-stil räcker det oftast med att istället ladda biblatex-chicago:

```
\usepackage[authordate,backend=biber]{biblatex-chicago}
```

5.3. Referenskommandon och referenslistor

När man så skapat sin referensdatabas (som givetvis uppdateras i takt med att rapporten blir större), valt bibliografistil och laddat biblatex är det dags att börja referera till artiklar.

Lehman m. fl. (2013, sidorna 79 ff.) listar en uppsjö med kommandon för att åstadkomma detta, men det är egentligen bara fyra eller fem som är viktiga att ha koll på. Fyra av dessa har dessutom varianter som kan vara bra att känna till, och alla fem tar i sitt huvudargument en kommaseparerad lista av referensnycklar från referensdatabasen.

Alla referenskommandon (utom \nocite) tar två frivilliga argument utöver den obligatoriska listan av referensnycklar. Det första frivilliga argumentet används för att sätta in text före referensen (vilket är praktiskt speciellt när man använder \parencite), och burkar oftast inte användas

5. Referenser med biblatex

alls. Det andra (eller det enda, om bara ett anges) används för att lägga till text *efter* referensen, vilket ofta används för att ange sidnummer. Då finns även kommandon `\pno`, `\ppno`, `\psq` och `\psqq` som formaterar sidhänvisningar snyggt, men oftast ska biblatex kunna göra detta automatiskt om argumentet bara består av siffror:

```
\parencite[12-14]{Lehman13}
```

`\nocite` används då man vill ha med en referens i referenslistan utan att behöva referera till den i text. Kommandot tar även ett specialargument, `*`, som indikerar att alla referenser i databasen ska inkluderas i listan.

`\textcite` används då man refererar i löpande text, *dvs.* där man använder referensen som ett substantiv ("...NN har visat att...").

`\parencite` används i de fall då man vill "indirekt" referera till en källa, exempelvis genom att referera till tidigare resultat ("...tidigare resultat (NN) har varit positiva...").

`\footcite` används när man vill referera genom en fotnot.

`\autocite` försöker vara intelligent och använda det kommando som passar bäst i situationen med hänsyn till den bibliografistil som används, men bör bara användas i situationer där man normalt skulle använda `\parencite` eller `\footcite`. Kommandot förklaras tydligare i manualen (Lehman m. fl. 2013, sidan 82).

De fyra sista har varianter för multipla referenser med olika sidhänvisningar, definierade på formen

```
\textcites[<sidor>]{<referens 1>}[<sidor>]{<referens 2>}...
```

med analoga ekvivalenter för `\parencite` och `\footcite`. Trion `\textcite`, `\parencite` och `\autocite` (och deras multipelvarianter) har dessutom varianter à la `\Textcite`, som är gjorda för att användas i början på en mening eller i andra situationer där texten bör inledas med en versal.

Del V.

Fortsättningen

6. Vidare läsning

Den här introduktionen har förhoppningsvis gett dig en bra \LaTeX -grund som låter dig typsätta både rapporter och artiklar utan problem. Tyvärr kommer du, eftersom \LaTeX är ett så stort system, sannolikt att behöva ytterligare hjälp, tips och resurser allt eftersom du använder \LaTeX och stöter på problem eller svårigheter.

Denna del av introduktionen ämnar ge dig några tips på sådana resurser. Delen inleds med några relevanta resurser så som böcker och journaler som behandlar \TeX och \LaTeX på en mer eller mindre avancerad nivå, samt några tips när det gäller att få hjälp med specifika problem. Därefter följer några tips gällande stora projekt, som du sannolikt kommer att ha nytta av förr eller senare, speciellt om du studerar på en högskola eller ämnar författa böcker och liknande.

Därefter följer en lista av viktiga och nyttiga \LaTeX -paket, som man bör åtminstone skumma igenom för att få en hyfsad uppfattning om vilka paket som finns och när man bör använda dem. Avslutningsvis ges även en lista över andra projekt som liknar \LaTeX , till exempel det tidigare nämnda $X\TeX$.

6.1. Andra resurser

Även om denna introduktion siktar på att ge dig allt du behöver för att lära dig \LaTeX är det inte säkert att den är tillräcklig. Man kan inte diskutera allt i en kort introduktion, och vill man lära sig mer om den inre strukturen hos \TeX eller \LaTeX så finns det redan mycket bra och utförliga

6. Vidare läsning

resurser tillgängliga. Dessutom kan man i en kort introduktion inte diskutera specifika problem i detalj, eftersom dessa ofta beror på den specifika situationen. Nedan följer därför en lista över resurser i form av böcker, journaler och forum som kan hjälpa dig släcka din kunskapstörst eller lösa dina \LaTeX -problem.

6.1.1. Böcker och artiklar

Det finns mycket material tillgängligt när det gäller \LaTeX . Många böcker inom ämnet har publicerats av Addison-Wesley, och det finns även en uppsjö artiklar, böcker och guider tillgängliga på CTAN, och en journal (*The \LaTeX Journal*¹) som finns tillgänglig på internet. Nedan följer en lista på några av de böcker och artiklar som kan vara av intresse för dig som precis börjat med \LaTeX .

***An essential guide to \LaTeX 2_ε usage* (Fenn 2007)** brukar också refereras till som *lztabu* (dess namn på CTAN), och ger en kort lista över utdaterade paket, dödssynder inom \LaTeX samt några små tips. Förhoppningsvis lär du dig inget nytt av att läsa *lztabu* (det skulle ju indikera att den här introduktionen är felaktig), men det kan vara värt att skumma igenom den ändå.

***The Not So Short Introduction to \LaTeX 2_ε* (Oetiker m. fl. 2011)** går även under namnet *lshort* och är en kort introduktion till \LaTeX , skriven på engelska. Den går *lite* djupare in på vissa bitar av \LaTeX , och fokuserar en del på den nu i princip överflödiga \LaTeX -motorn som skapar DVI-filer, men har varit en stor inspiration till den här introduktionen. Absolut värd en (snabb) genomläsning.

***Math into \LaTeX* (Grätzer 1996)** ger än ännu längre introduktion till \LaTeX än *lshort*, och är även den skriven på engelska. En bra ytterligare referens om det är något man vill veta mer om eller något man tycker är otydligt i den här introduktionen och *lshort*.

¹ <http://tug.org/pracjourn/>

6. Vidare läsning

Math Mode (Voß 2010) ger en mycket ordentlig genomgång av matematiktypsättning både i vanliga \LaTeX och med $\mathcal{A}\mathcal{M}\mathcal{S}\text{\LaTeX}$, och är en nästintill oundgänglig referens när man skriver lite mer komplicerade ekvationer. Bokmärk och titta igenom varje gång du skriver matematik i \LaTeX .

Short Math Guide for \LaTeX (Downes 2002) är en kort guide till matematik med $\mathcal{A}\mathcal{M}\mathcal{S}\text{\LaTeX}$ skriven av en av huvudpersonerna bakom paketsamlingen. Ger några små värdefulla tips, en lista över symboler och en introduktion till `\DeclareMathOperator` och några andra $\mathcal{A}\mathcal{M}\mathcal{S}$ -konstruktioner som inte diskuteras utförligt i den här introduktionen. Sjutton mycket läsvärda sidor.

The \LaTeX Companion (Mittelbach, Goossens m. fl. 2004) beskriver alla \LaTeX -kommandon och en stor mängd paket. Om du ska skriva ett eget paket eller en egen dokumentklass, eller bara är intresserad av att "hacka" \LaTeX lite, så bör du absolut ta en titt på den här boken. En mycket bra referens för den vane \TeX aren.

\LaTeX : A Document Preparation System (Lamport 1994) skrevs av författaren till \LaTeX och kan anses vara det närmsta en "manual" till \LaTeX man kan komma. Främst av historiskt intresse, och inget för nybörjaren.

\TeX by Topic (Eijkhout 1992) ger en utförlig förklaring till \TeX och är en relevant referens för alla som ska utföra lågnivåarbete i \TeX eller \LaTeX . Främst riktad till mycket vana användare av \TeX och \LaTeX , och absolut inte riktad till nybörjaren.

The \TeX book (Knuth 1986) skrevs av skaparen av \TeX och förklarar i detalj hur systemet fungerar. Främst av historiskt intresse, och egentligen bara värd att titta på om man vill veta hur \TeX *egentligen* fungerar.

Utöver dessa bör man givetvis även läsa manualerna till de paket man använder (dessa finns alltid på CTAN).

6.1.2. Hjälp med specifika problem

Det tar lång tid att bemästra \LaTeX fullt ut, och i början kommer man garanterat att stöta på problem. Några av dem kanske går att lösa med den hjälp som ges i den här introduktionen och de andra böcker och artiklar som presenteras, men vissa måste man fråga någon om.

Det finns så klart en uppsjö olika maillistor, forum och sökmotorer (och phaddrar, för dig som går på en teknisk högskola) som kan användas för att lösa problem, ställa frågor och utforska \LaTeX . En mycket bra resurs är *TeX Stack Exchange*¹ där man kan ställa frågor om \TeX , \LaTeX och andra relaterade system. Många stora namn i \TeX -världen dyker upp där lite då och då, och undrar man något om \LaTeX så är det ett utmärkt ställe att fråga. Använder man IRC kan man också med fördel besöka #latex-kanalen på Freenode.²

6.2. Tips för stora projekt

För stora \LaTeX -projekt (till exempel kandidatarbeten, examensarbeten och liknande) är det viktigt att kunna ha en ordentlig ordning på sitt dokument. Är det dessutom ett dokument många ska samarbeta med är det ännu viktigare att det är strukturerat.

Den första tumregeln, som även bör tillämpas vid mindre projekt, är att man bör lägga varje \LaTeX -dokument (eller projekt) i en egen undermapp. Ännu bättre blir det om man även lägger alla externa filer (figurer, inkluderad programkod och så vidare) i ännu en undermapp. En bra mappstruktur skulle alltså kunna se ut ungefär som i **exempel 6.1**.

Utöver detta kan man se till att försöka abstrahera bort till exempel kommandodefinitioner eller stiländringar till ett eget paket eller en egen dokumentklass, om möjligt. Att göra detta lämnas som en övning åt läsaren, men mer information om hur man gör sådant ges av bland annat Flynn (2006), The \LaTeX 3 Project (1999) och Robertson (2006).

¹ <http://tex.stackexchange.com/> ² <irc://chat.freenode.net>

6. Vidare läsning

```
.
|-- FFM233-projekt
|  |-- img
|  |  |-- degradering-1.png
|  |  |-- degradering-2.png
|  |  |-- ...
|  |  |-- triangel-1.png
|  |  |-- triangel-2.png
|-- kod
|  |-- diffekv.m
|  |-- ode.m
|  |-- plot.m
|-- projekt.tex
|-- referenser.bib
```

Exempel 6.1: En bra mappstruktur för ett enkelt \LaTeX -projekt.

6.2.1. Versionshantering

För större projekt och projekt som utförs i grupp är det mycket praktiskt att kunna spåra ändringar och ändra dokumentet från många olika platser (gärna samtidigt). Med hjälp av ett versionshanteringssystem så som Subversion¹ blir detta enkelt. Ännu bättre blir det om man använder ett distribuerat system så som Mercurial² eller Git,³ hostat helt gratis av något snällt företag, till exempel på Bitbucket⁴ (Github⁵ eller Gitorious⁶ om man föredrar Git framför Mercurial).

Att utförligt förklara hur Mercurial fungerar är utanför den här bokens område, men är man intresserad bör man läsa *Hg Init*,⁷ som trots sitt bruk av Comic Sans är en utmärkt resurs för den som vill lära sig Mercurial och versionshantering.

¹ <http://subversion.tigris.org/> ² <http://mercurial.selenic.com/>

³ <http://git-scm.com/> ⁴ <https://bitbucket.org/> ⁵ <https://github.com/>

⁶ <http://gitorious.org/> ⁷ <http://hginit.com/>

6.2.2. Uppdelning av dokumentet

Stora projekt (med flera kapitel eller stora delar) kan med fördel delas upp på flera filer. Man har då en grundfil (säg, `projekt.tex`) som refererar till flera olika underfiler (kanske en per kapitel, lämpligen placerade i en undermapp till projektet). Dessa kan man sedan inkludera i grundfilen på ett antal olika sätt, varpå man bara kompilerar

Den första metoden för att inkludera underfilerna är `\input`, som i princip lägger in koden från underfilen direkt där kommandot används och fortsätter kompilera som om koden alltid funnits där. Detta är praktiskt om man av någon anledning inte vill använda den andra metoden.

Den andra metoden, `\include`, är i princip ekvivalent till `\input` med den skillnaden att en sidbrytning läggs in före och efter den inkluderade koden. Dessutom kan man med hjälp av kommandot `\includeonly` bestämma vilka underfiler som inkluderas, för att till exempel spara tid vid kompileringen om man bara är intresserad av en specifik del.

Den tredje och sista metoden är med paketet `subfiles`. Denna är generellt sett att föredra om den är tillgänglig, och fungerar ungefär som metod ett men med undantaget att man även kan kompilera underfilerna var för sig, om man är intresserad av detta. Det kan vara lämpligt att göra om man bara redigerar ett kapitel, och inte vill kompilera hela projektet under tiden. Paketet `subfiles` finns på CTAN men inte i Chalmers datorsystem eller i \TeX Live.

6.3. Rekommenderade paket

Eftersom \TeX är Turingkomplett kan i princip allt göras med språket, även om det främst är tänkt för typsättning. Således kan det mesta i typsättningsväg lösas relativt enkelt. Kan det inte det, och problemet man vill lösa är ett relativt vanligt problem, så finns det sannolikt ett paket som löser problemet. Dessa paket finns listade på CTAN (se [sida 4](#)) och kan, om man använder \TeX Live, installeras med verktyget `tlmgr`.

6. Vidare läsning

Dokumentation för alla paket finns på CTAN och kan även visas genom att skriva `texdoc paketnamn` i terminalen, eller på <http://texdoc.net/pkg/<paketnamn>>.

6.3.1. Allmänt nyttiga paket

Dessa paket finns i princip alltid och tillhandahåller så grundläggande funktionalitet att man alltid bör använda dem. Det handlar om allt från tecken- och typsnittskodning till avstavning och interna länkar.

babel (ej med Xe_{La}T_EX) översätter interna strängar (till exempel "Referenser" och "Sammanfattning") till det språk som önskas och gör det möjligt för dessa språk att avstavas ordentligt. Paketet är ett måste då man använder L^AT_EX, men har ersatts av polyglossia i Xe_{La}T_EX.

fixltx2e korrigerar en del buggar i L^AT_EX 2_ε-kärnan och gör till exempel matematikkommandon robusta. Inkludera alltid.

fontenc (ej med Xe_{La}TeX och Lua_{La}TeX) låter användaren välja typsnittskodning. Från början använder L^AT_EX OT1, som inte innehåller icke-anglikanska tecken så som å, ä och ö. Detta medför problem när det gäller bland annat avstavning, och man bör därför istället använda T1. Detta diskuteras närmre på [sida 13](#).

fontspec (endast med Xe_{La}TeX och Lua_{La}TeX) låter användaren välja OTF-typsnitt helt fritt, antingen från typsnitt installerade på datorn eller från TTF- eller OTF-filer.

hyperref gör om innehållsförteckningar, referenser och URler till riktiga länkar i de fall detta stöds (det vill säga om slutformatet är PDF). Dessutom skapar den inbyggda innehållsförteckningar som kan användas för att navigera i dokumentet i en del PDF-läsare (till exempel Skim och Acrobat Reader). Ett oömbärligt paket som alltid bör inkluderas.

6. Vidare läsning

inputenx (ej med $\text{X}\text{\LaTeX}$ och $\text{Lua}\text{\TeX}$) är ett paket som låter användaren berätta för $\text{L}\text{\TeX}$ vilken teckenkodning indatan sparats med. De vanligaste inställningarna är `utf8` och `latin1`, men många andra teckenkodningar stöds också. Detta paket introducerades lite kort på [sida 13](#).

kpfonts inkluderar typsnittsfamiljen *Kepler* som är en gratis familj baserad på Adobe Palatino. Personligen tycker jag att *Kepler* ser bättre och tydligare ut än *Latin Modern*, och paketet innehåller dessutom en enorm mängd typsnitt som täcker alla situationer. Välj antingen detta paket *eller* *lmodern*, inte båda.

lmodern inkluderar typsnittet *Latin Modern* för att ersätta *Computer Modern*, standardtypsnittet i $\text{L}\text{\TeX}$. *Latin Modern* ser i princip likadant ut som *Computer Modern*, men är skarpare och har fler glyfer. Det här paketet motiveras lite närmre på [sida 38](#). Välj antingen detta paket *eller* *kpfonts*, inte båda.

nag försöker upptäcka saker som är *bad practice*, saker som kanske inte fungerar som man tänkt sig, och liknande. Bör inkluderas med alternativen `l2tabu` och `orthodox`:

```
\usepackage[l2tabu,orthodox]{nag}
```

polyglossia (endast med $\text{X}\text{\LaTeX}$) ersätter `babel` för $\text{X}\text{\LaTeX}$ och har bättre stöd för UTF-8.

6.3.2. Paket för snyggare typsättning

Även om $\text{L}\text{\TeX}$ är mycket bra på att typsätta så blir det ibland inte så snyggt som man kanske kan önska. Vid dessa tillfällen kan man applicera diverse olika paket för att få hjälp med detta. Det gäller till exempel tabeller, figurtexter, SI-enheter och matematik.

booktabs är ett paket som gör det möjligt att skapa mycket snygga tabeller. Paketet diskuteras redan på [sida 30](#), och är i princip ett måste om man ska inkludera tabeller i sitt $\text{L}\text{\TeX}$ -dokument.

6. Vidare läsning

caption gör det möjligt att ändra stilen på figur- och tabelltexter så att de syns tydligare och inte smälter in i texten.

csquotes gör det lättare att typsätta citat på ett snyggt och korrekt sätt. Bland annat definieras kommandot `\enquote`, som kan användas istället för de (inkorrekta) citattecknen `` `` och `' '`, och kommandot `\blockquote` för längre block av citerad text.

fnpct ser till att fotnoter nära skiljetecken typsätts på ett snyggt sätt, och gör det lättare att typsätta flera fotnoter brevid varandra. Introducerades kort på [sida 41](#).

siunitx gör det enklare att typsätta SI-enheter, decimaltal och vinklar på ett korrekt sätt för icke engelskspråkiga rapporter. Mycket användbart paket som tyvärr inte finns på Chalmers datorer. Förklaras lite kort på [sida 68](#).

Paket för matematiktypsättning

Som teknisk matematiker (eller fysiker) kommer stora delar av de rapporter man skriver oundvikligen innehålla matematik. Även om \LaTeX för sig självt är relativt bra på att typsätta matematik kan det ibland behövas några tillägg för att göra det enklare. Många av dessa tillhandahålls av \mathcal{AMS} i det som kallas $\mathcal{AMS}\text{\LaTeX}$, men det finns fler relevanta matematikpaket.

amsmath går igenom i [kapitel 3](#) och är i princip oundgängligt om man ska typsätta matematik med \LaTeX . Inkludera alltid detta paket om du skriver något som kan tänkas innefatta ekvationer.

amssymb definierar en hel del matematiska symboler, till exempel `\ggg` (\ggg) och `\therefore` (\therefore), samt kommandot `\mathbb` som ger krittavletecken (som används för de grundläggande talmängderna).

amsthm definierar omgivningar för att typsätta teorem, satser, lemman, bevis och dylikt. Användbart i vissa sammanhang, främst för att typsätta föreläsningssanteckningar eller uppgifter.

6. Vidare läsning

isomath korrigerar typsättningen av grekiska bokstäver, så att även dessa typsätts kursivt (som variabler). Introducerar även semantiska kommandon som `\vectorsym` och `\tensorsym` för typsättning av vektorer och tensorer.

mathtools lagar några småfel i `amsmath` och definierar kommandon som till exempel `\DeclarePairedDelimiter`, som introducerades på [sida 58](#).

skmath definierar en del kommandon som gör det lättare att typsätta matematik (till exempel `\d` som presenterades på [sida 60](#)) och omdefinierar andra för att förbättra utseendet.

unicode-math (endast med `XYTeX` och `LuaTeX`) kan användas för att ladda OTF-typsnitt för matematik och gör det även möjligt att skriva matematik med Unicode-tecken.

xfrac definierar ett kommando `\sfrac`, som låter dig typsätta bråk med ett snedstreck: $\frac{2}{3}$.

6.3.3. Paket för grafik

Grafik i form av figurer eller illustrationer är givetvis en viktig del av många rapporter, vare sig det är figurer av uppställningar, plottar eller flödesscheman. Det finns tre relevanta paket när det gäller grafik i `LATeX`; ett som används för att importera grafik från externa filer och två som används för att rita direkt med `LATeX`-kommandon.

graphicx förklaras kort i [kapitel 4](#) och gör det möjligt att inkludera figurer från externa filer i sitt `LATeX`-dokument. Mycket användbart om man har data från till exempel MATLAB eller Mathematica.

tikz nämns också i [kapitel 4](#) och gör det möjligt att rita vektorbaserade figurer direkt i `LATeX`. Mycket användbart om man vill rita exempelvis uppställningar, enklare illustrationer eller flödesscheman, men kan användas till otroligt mycket mer. Fungerar endast med `pdfLATeX` i PDF-läge eller moderna varianter som `XYTeX`.

6. Vidare läsning

pstricks är en slags motsvarighet till tikz som endast fungerar med de \LaTeX -varianter som genererar DVI- eller PS-filer. Inte lika användbart som tikz eftersom det inte fungerar med pdf \LaTeX i PDF-läge.

6.3.4. Paket för bibliografier

Eftersom biblatex (som diskuteras i [kapitel 5](#)) har bra stöd för många olika språk, bibliografistilar och såväl numeriska system som Harvard-stilen behövs det sällan extra paket. Det kan däremot vara värt att nämna några av de bibliografistilar som finns. Dessa stilar används normalt genom att ange stilen när man laddar biblatex, dvs. `\usepackage[style=...]{biblatex}`, om inget annat anges.

Utöver de stilar som listas här finns även ett stort antal stilar som riktar sig till specifika områden eller publikationer på CTAN.¹

alphabetic liknar numeric, men använder en alfabetisk etikett skapad utifrån författarn och årtal istället för en numerisk etikett.

authoryear motsvarar ungefär Harvard-stilen. Referenser i texten typsätts på formen "Författare (2001)" (eller liknande former, se [kapitel 5](#) för en längre diskussion) och litteraturlistan på motsvarande sätt.

biblatex-chicago implementerar Chicago-stilen enligt Chicago Editorial Staff (2010). Den här stilen ska inte laddas genom biblatex, utan laddas istället som ett eget paket trots att det är en biblatex-stil i grunden:

```
\usepackage[authordate,backend=biber]{biblatex-chicago}
```

ieee implementerar stilen som används av IEEE i deras officiella dokumentklasser för \LaTeX .

¹ <http://www.ctan.org/tex-archive/macros/latex/contrib/biblatex-contrib>

6. Vidare läsning

numeric är en numerisk bibliografistil som liknar standarstilen i BibTeX. Referenser i texten visas på formen "[1]", och litteraturförteckningen har motsvarande numeriska etiketter.

phys implementerar en stil som följer de riktlinjer som satts upp av *American Physical Society* och *American Institute of Physics*.

6.3.5. Paket som löser problem

Ibland vill man göra något som är väldigt svårt att göra med vanlig L^AT_EX-kod, till exempel ändra sidstorlek eller skapa underfigurer. Då många sådana problem är vanligt förekommande finns det ofta paket som löser dem.

acro låter dig skapa listor över förkortningar på ett enkelt sätt, men är lite enklare än glossaries som egentligen är gjort för hela ordlistor. Till skillnad från glossaries kräver acro inget externt program. Diskuteras kort på [sida 35](#).

cleveref definierar kommandot `\cref` (och många liknande kommandon), ett alternativ till `\ref` som automatiskt skriver ut vilken typ av referens det handlar om. Fullt kompatibelt med `varioref`, och tillsammans gör det korsreferenser mycket enklare.

enumitem gör det möjligt att definiera egna sorters listor, och modifierar `enumerate` så att man även kan specificera en egen etikett, återuppta numreringen från förra listan, och mycket mer.

float definierar ett kommando `\newfloat` som skapar nya sorters flytande objekt (vilka diskuterats på [sida 27](#)).

geometry kan användas för att ändra en del mått i L^AT_EX (till exempel sidans storlek eller marginalerna) om så önskas. Oftast behöver man inte göra detta, eftersom dokumentklassen har definierat de mått den har av en anledning. Kan vara värdefullt om man vill skapa dokument för A5- eller A6-papper (eller andra ovanliga storlekar).

6. Vidare läsning

- glossaries** diskuterades tidigare på [sida 34](#), och låter dig skapa ordlistor och listor över förkortningar på ett enkelt och relativt automatiserat sätt. Paketet är mycket användbart i tekniska dokument där en ordlista är viktig för förståelsen.
- imakeidx** nämndes på [sida 35](#) och låter dig skapa sakregister på ett automatiserat sätt. Paketet kan vara användbart i större dokument, främst böcker och liknande facklitteratur.
- multicol** löser problemet med att typsätta text i kolumner på ett mycket bättre sätt än standardklasserna, med hjälp av omgivningen `multicols`. Paketet stödjer bland annat korrekt balanserade kolumner.
- multirow** gör det möjligt att skapa tabellceller som spänner över flera rader.
- sidecap** definierar omgivningen `SCfigure` som typsätter en figur med figurtexten brevid istället för under eller över figuren. Kan vara användbart om man har smala figurer med lång figurtext.
- subfig** definierar kommandon som möjliggör skapandet av ”underfigurer”, det vill säga grupper av relaterade figurer som alla kan refereras till antingen som grupp (till exempel ”Figur 1”) eller individuellt (till exempel ”Figur 1a”).
- varioref** gör det möjligt att, med hjälp av kommandot `\vref`, skapa korsreferenser som inte bara refererar till etiketterna med nummer utan även till den sida figuren eller tabellen finns på. Detta görs på ett intelligent sätt, så att referensen blir ”figur X på nästa sida” om figuren är på nästa sida, och så vidare. Mycket relevant paket, speciellt i större rapporter.
- wrapfig** skapar ett nytt flytande objekt `wrapfig` som placerar figuren till höger eller vänster på sidan och låter övrig text ”flyta” runt figuren.
- xspace** definierar ett kommando `\xspace` som kan läggas till på slutet av kommandodefinitioner för att de ska bete sig snällare i brödtext, så att man slipper skriva `{}` efter kommandot.

6.3.6. Paket för att ändra utseende

Standardklasserna lämnar ofta en del att önska i termer av utseende. Vill man göra något åt detta kan man använda paket som utformats för att låta dig ändra stilen av vissa element, till exempel rubriker eller sidhuvuden.¹

fancyhdr låter dig förändra sidhuvud och sidfot för att införa snyggare, mer informativa eller tydligare stilar. Till exempel så kan texten i sidhuvudet ändras för att visa sidnummer och kapitel, medan sidfoten ändras för att visa till exempel kontakt- eller copyrightinformation.

titlesec gör det möjligt att ändra stilen på rubrikerna i dokumentet, till exempel genom att byta textstil eller sättet rubriken typsätts.

tocloft definierar kommandon för att styra utseendet av innehållsförteckningen.

6.3.7. Andra specialiserade paket

För en del snäva områden så som kvantfysik eller datavetenskap finns det specialiserade paket som löser specifika uppgifter. Dessa kan med fördel användas om man skriver rapporter inom området, eller rör vid ämnet i någon inlämningsuppgift.

algorithmic är ett paket för typsättning av algoritmer och pseudokod.

braket definierar kommandon för att handskas med braket-notationen som används inom kvantfysik. Kommandot `\Braket` kan således resultera i följande ekvation:

$$\left\langle \phi \left| \frac{\partial^2}{\partial t^2} \right| \psi \right\rangle$$

¹ Vill man lösa problemet på riktigt ska man givetvis använda en dokumentklass som ser bättre ut istället.

6. Vidare läsning

Paketet definierar även kommandot `\Set`, för att på liknande sätt typsätta mängddefinitioner:

$$\{ x \in \mathbf{R}^2 \mid 0 < |x| < 5 \}$$

listings kan användas för att inkludera programkod i \LaTeX -dokument. Det finns möjlighet att skriva ut radnummer, lägga till ramar, associera figurtexter och även en mycket enkel syntaxfärgning (eng. *syntax highlighting*). Fungerar inte om koden innehåller svenska tecken, även om paketet *listingsutf8* försöker lösa detta. Använder man $X\TeX$ uppstår inte detta problem.

minted kan sägas vara en förbättring av **listings** som använder Python-programmet *Pygments* för att även sätta färg på koden. Ett bra alternativ om man ska inkludera kod och vill ha fullständig syntaxfärgning. Paketet lider dock av samma problem med svenska tecken som **listings**.

todonotes låter dig infoga (synliga) ”att göra”-anteckningar i ditt dokument, tillsammans med en lista över dessa. Paketet innehåller även ett kommando `\missingfigure`, som kan användas för att markera en saknad figur.

6.4. Andra \TeX -baserade projekt

\LaTeX är inte det enda användbara \TeX -baserade projektet, även om det är det makropaket som används mest i praktiken. Det finns alternativ som baseras på \LaTeX men löser problem (till exempel $X\TeX$ och $\text{Lua}\TeX$) men även alternativa makropaket som är mycket olika \LaTeX i sin struktur (till exempel $\text{Con}\TeX\text{t}$). Dessutom finns det projekt för att utveckla $\LaTeX 2_{\epsilon}$ och göra vissa saker som till exempel utveckling av dokumentklasser mycket enklare.

6.4.1. Unicode-baserade Xe_ΛTeX

Xe_ΛTeX är, precis som pdf_ΛTeX, en TeX-kompilator. Denna kan användas på samma sätt som vanliga TeX och pdf_ΛTeX (dvs. den kan köras på i princip samma kod) med hjälp av dess program, `xelatex`.

Skillnaden mellan pdf_ΛTeX och Xe_ΛTeX som gör att man kanske föredrar det senare är relativt stor, sett till de djupa delarna av TeX: Xe_ΛTeX förutsätter, till skillnad från pdf_ΛTeX, att all indata är UTF-8 och kan därmed läsa UTF-8-dokument på ett korrekt sätt. Detta innebär bland annat att saker som inte fungerar i pdf_ΛTeX ens med `inputenx`, till exempel svenska tecken i kodlistningar eller inkluderade filer, kommer att fungera utmärkt med Xe_ΛTeX.

Dessutom kan man med hjälp av speciella kommandon välja typsnitt på ett mycket enklare sätt, eftersom Xe_ΛTeX stödjer OTF- och TTF-typsnitt, och kan hitta alla typsnitt som är installerade på datorn, inte bara de som finns i form av _ΛTeX-paket.

En relativt övergripande bild av vad som är nytt i Xe_ΛTeX ges av Robertson (2013); för det mesta kan man förutsätta att _ΛTeX-kod kommer att fungera precis likadant i Xe_ΛTeX med några få undantag:

- Paketet `polyglossia` bör användas istället för `babel`
- Varken `inputenx` eller `fontenc` bör användas
- Paketet `fontspec` bör användas för att ladda typsnitt
- Paketet `unicode-math` kan användas för att ladda typsnitt för matematik och för att skriva matematik med Unicode-tecken.

Xe_ΛTeX är idag så stabilt att det kan användas istället för pdf_ΛTeX. Den här boken är typsatt med Xe_ΛTeX, och eftersom skillnaderna främst ligger i att `inputenx` och `fontenc` inte bör användas (Xe_ΛTeXklarar nämligen även gamla pdf_ΛTeX-typsnitt) kan man i princip alltid använda Xe_ΛTeX istället för pdf_ΛTeX.

6.4.2. Skripta med LuaTeX

LuaTeX är en kombination av programmeringsspråket Lua och typsättningspråket TeX. Introduktionen till Lua^ATeX, som är LuaTeXs motsvarighet till vanliga L^ATeX, beskriver systemet som uppföljaren till pdfL^ATeX:

It is the designated successor of pdfTeX and includes all of its core features: direct generation of PDF files with support for advanced PDF features and micro-typographic enhancements to TeX typographic algorithms.

Pégourié-Gonnard (2013)

I många avseenden är LuaTeX mycket likt X_YTeX; man kan använda konventionella typsnitt med båda motorerna (med paketet fontspec), och de hanterar båda UTF-8 mycket bättre än sina företrädare. Skillnaden är att LuaTeX även gör det möjligt att bädda in Lua-kod direkt i dokumentet, vilket gör en del saker mycket enklare eftersom man har tillgång till ett fullgott skriptspråk. I likhet med X_YTeX är LuaTeX stabilt nog för att användas istället för pdfL^ATeX.

6.4.3. Ett alternativ: ConTeXt

ConTeXt är ett alternativt makropaket för TeX som utvecklats parallellt med L^ATeX men med en annan inriktning. Medan L^ATeX försöker isolera användaren från typografiska beslut, vilket gör det lämpligt för att till exempel skicka in artiklar till förlag, så försöker ConTeXt ge användaren en lite mer strukturerad tillgång till de typografiska funktionerna i TeX.

Det finns en del artiklar som jämför ConTeXt och L^ATeX (till exempel Hokenwater 1998), men kontentan av det hela är att L^ATeX är vanligare i akademiska kretsar och bättre på att typsätta matematik, och att man därför oftast bör hålla sig till L^ATeX. Är man trots detta intresserad av ConTeXt bör man referera till ConTeXt-manualen (Hagen 2001).

6.4.4. Framtiden: L^AT_EX₃

Den variant av L^AT_EX som diskuterats i boken, L^AT_EX 2_ε, härstammar från 1994 då den ersatte den tidigare versionen, L^AT_EX 2.09. Denna variant uppdateras fortfarande regelbundet, men redan 1997 startade ett initiativ för att utveckla nästa version av L^AT_EX, kallad L^AT_EX₃ (Mittelbach och Rowley 1999), som fortgår än idag. Än så länge är L^AT_EX₃ inte något man som användare behöver ha stenkoll på, eftersom det främst består av en uppsättning paket som underlättar i skrivandet av paket.

Hur L^AT_EX₃ kommer se ut är inte helt klart, men att det kommer kunna användas med samma kompilatorer som används idag är givet. Utöver detta listar Mittelbach och Rowley (1999, sidorna 4 f.) ett antal mål för projektet.

I dagsläget består projektet som sagt av ett ramverk som underlättar skrivandet av nya paket. Detta består av en uppsättning konventioner (The L^AT_EX₃ Project 2013a), som bland annat beskriver hur man ska namnge de interna kommandon man definierar i ett paket, och en samling paket (The L^AT_EX₃ Project 2013b) som enligt dessa konventioner definierar kommandon som abstraherar användbara programmeringskoncept (till exempel listor, Booleska variabler, heltal, flyttal m.fl.). Av den anledningen är L^AT_EX₃ något att hålla ögonen på om man är intresserad av att skriva egna paket eller dokumentklasser, men för den ”vanliga” användaren ligger L^AT_EX₃ fortfarande bortom horisonten.

Litteraturförteckning

Nedan följer en litteraturförteckning. I denna listas alla de böcker, artiklar och manualer som refererats till i boken, med hänvisningar till de sidor verket nämndes på. En del av de verk som listas, främst olika manualer, har en CTAN-länk istället för ett ISBN-nummer eller en URL. Dessa verk kan man hitta genom att lägga till <http://mirrors.ctan.org/> innan länken, eller genom att titta i sin \LaTeX -distribution (och då är det enklast att använda texdoc, som beskrivs på [sida 13](#)).

American Mathematical Society (1999). *User's Guide for the amsmath Package*. Version 2.0. Refererad på sida [61](#).

URL: <ftp://ftp.ams.org/ams/doc/amsmath/amslldoc.pdf>.

De Beaudrap, Niel (2011). *Good practice on spacing*. Svar till fråga ställd på *T_EX Stack Exchange*. 10 maj 2011. Senast hämtad 2013-10-20. Refererad på sida [60](#).

URL: <http://tex.stackexchange.com/a/5518/66>.

Beccari, Claudio och Enrico Gregorio (2012). *The package imakeidx*. Version 1.2e. 22 oktober 2012. Refererad på sida [35](#).

CTAN: <macros/latex/contrib/imakeidx/imakeidx.pdf>.

Borland, David och Russell M. Taylor II (2007). "Rainbow Color Map (Still Considered Harmful". I: *IEEE Computer Graphics and Applications* 27.2 (mars 2007), sidorna 14–17. Refererad på sida [74](#).

Camoes, Jorge (2010). *Alphabetical Sorting Must Die*. 26 oktober 2010. Senast hämtad 2013-11-04. Refererad på sida [72](#).

URL: <http://www.excelcharts.com/blog/alphabetical-sorting-must-die/>.

Carlisle, D. P. (2005). *Packages in the "graphics" bundle*. November 2005. Refererad på sida [76](#).

CTAN: <macros/latex/required/graphics/grfguide.pdf>.

Litteraturförteckning

- Chicago Editorial Staff (2010). *The Chicago Manual of Style*. 16. utgåvan. Chicago: University of Chicago Press. Refererad på sidorna 94, 107.
ISBN: 978-0-226-10420-1.
- Christoffersson, Karin m. fl. (1998). *Språkguiden*. Linköping: Linköpings Universitet. Refererad på sidorna 21, 37.
ISBN: 978-9-163-06527-9.
URL: <http://www.student.liu.se/service/sprakguide/1.265737/sprakguiden.pdf>.
- Downes, Michael (2002). *Short Math Guide for L^AT_EX*. Refererad på sida 99.
URL: <ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>.
- Eijkhout, Victor (1992). *T_EX by Topic*. Reading, Massachusetts: Addison-Wesley. Refererad på sida 99.
ISBN: 978-0-201-56882-0.
URL: <http://eijkhout.net/texbytopic/texbytopic.html>.
- Ensenbach, Marc och Mark Trettin (2011). *Das L^AT_EX 2_ε-Sündenregister. Veraltete Befehle, Pakete und andere Fehler*. September 2011. Refererad på sida 116.
CTAN: <info/l2tabu/german/l2tabu.pdf>.
- Fear, Simon (2005). *Publication quality tables in L^AT_EX*. Version 1.61803. April 2005. Refererad på sidorna 30, 31.
CTAN: <macros/latex/contrib/booktabs/booktabs.pdf>.
- Fenn, Jürgen (2006). "Managing Citations and Your Bibliography with B_BT_EX". I: *The PracT_EX Journal* 4. Refererad på sida 87.
URL: <http://www.tug.org/pracjourn/2006-4/fenn/fenn.pdf>.
- Fenn, Jürgen (2007). *An essential guide to L^AT_EX 2_ε usage. Obsolete commands and packages*. Juni 2007. Översättning av Ensenbach och Trettin (2011). Refererad på sida 98.
CTAN: <info/l2tabu/english/l2tabuen.pdf>.
- Feuersänger, Christian (2013a). *Manual for Package pgfplots*. Version 1.9. 3 oktober 2013. Refererad på sida 78.
CTAN: <graphics/pgf/contrib/pgfplots/doc/pgfplots.pdf>.
- Feuersänger, Christian (2013b). *Manual for Package pgfplotstable*. Version 1.9. 3 oktober 2013. Refererad på sida 78.
CTAN: <graphics/pgf/contrib/pgfplots/doc/pgfplotstable.pdf>.

Litteraturförteckning

- Flynn, Peter (2006). "Rolling your own Document Class: Using \LaTeX to keep away from the Dark Side". I: *The PracTeX Journal* 4. Refererad på sida 100.
URL: <http://tug.org/pracjourn/2006-4/flynn/flynn.pdf>.
- Fussner, David (2013). *The biblatex-chicago package. Style files for biblatex*. Version 0.9.9c (beta). Mars 2013. Refererad på sida 94.
CTAN: [macros/latex/contrib/biblatex-contrib/biblatex-chicago/doc/biblatex-chicago.pdf](http://ctan.org/macros/latex/contrib/biblatex-contrib/biblatex-chicago/doc/biblatex-chicago.pdf).
- Grätzer, George A. (1996). *Math into \LaTeX* . Boston: Birkhäuser. Refererad på sida 98.
ISBN: 978-3-764-34131-2.
CTAN: [info/mil/mil.pdf](http://ctan.org/info/mil/mil.pdf).
- Hagen, Hans (2001). *ConTeXt the manual*. PRAGMA ADE. Refererad på sida 113.
URL: <http://www.pragma-ade.com/general/manuals/cont-eni.pdf>.
- Henry, Alan (2012). *How to Choose the Best Chart for Your Data*. 5 november 2012. Senast hämtad 2013-11-04. Refererad på sida 72.
URL: <http://liferhacker.com/5909501/how-to-choose-the-best-chart-for-your-data>.
- Hoekwater, Taco (1998). "Comparing ConTeXt and \LaTeX ". I: *MAPS Journal*. Refererad på sida 113.
URL: http://maps.aanhet.net/maps/pdf/20_42.pdf.
- Kime, Philip och François Charette (2013). *biber. A backend bibliography processor for biblatex*. Version 1.7. Juli 2013. Refererad på sida 90.
CTAN: [biblio/biber/documentation/biber.pdf](http://ctan.org/biblio/biber/documentation/biber.pdf).
- Knuth, Donald (1986). *The TeXbook*. Reading, Massachusetts: Addison-Wesley. Refererad på sida 99.
ISBN: 978-0-201-13448-3.
- Lamport, Leslie (1994). *\LaTeX : A Document Preparation System*. 2. utgåvan. Reading, Massachusetts: Addison-Wesley. Refererad på sidorna 3, 99.
ISBN: 978-0-201-52983-8.
- The \LaTeX 3 Project (1999). *\LaTeX 2_ε for class and package writers*. Refererad på sida 100.
URL: <http://www.latex-project.org/guides/clsguide.pdf>.

Litteraturförteckning

- The L^AT_EX₃ Project (2013a). *The expl3 package and L^AT_EX₃ programming*. Version 4582 (juli 2013). Refererad på sida 114.
CTAN: [macros/latex/contrib/l3kernel/expl3.pdf](http://www.ctan.org/macros/latex/contrib/l3kernel/expl3.pdf).
- The L^AT_EX₃ Project (2013b). *The L^AT_EX₃ Interfaces*. Juli 2013. Refererad på sida 114.
CTAN: [macros/latex/contrib/l3kernel/interface3.pdf](http://www.ctan.org/macros/latex/contrib/l3kernel/interface3.pdf).
- Lehman, Philipp m. fl. (2013). *The biblatex package. Programmable Bibliographies and Citations*. Version 2.5. Januari 2013. Refererad på sidorna 88–90, 94, 95.
CTAN: [macros/latex/contrib/biblatex/doc/biblatex.pdf](http://www.ctan.org/macros/latex/contrib/biblatex/doc/biblatex.pdf).
- The Mathworks, Inc. (2013a). *MATLAB Documentation*. What is the MATLAB Search Path? Version R2013b. Refererad på sida 80.
URL: http://www.mathworks.se/help/matlab/matlab_env/what-is-the-matlab-search-path.html.
- The Mathworks, Inc. (2013b). *MATLAB Documentation*. Print figure or save to file. Version R2013b. Refererad på sida 83.
URL: <http://www.mathworks.se/help/matlab/ref/print.html>.
- Mertz, Andrew och William Slough (2007). "Graphics with TikZ". I: *The PracT_EX Journal* 1. Refererad på sida 77.
URL: <http://www.tug.org/pracjourn/2007-1/mertz/mertz.pdf>.
- Mittelbach, Frank, Michel Goossens m. fl. (2004). *The L^AT_EX Companion*. 2. utgåvan. Reading, Massachusetts: Addison-Wesley. Refererad på sidorna 3, 99.
ISBN: 978-0-201-36299-2.
- Mittelbach, Frank och Chris Rowley (1999). *The L^AT_EX₃ Project*. whitepaper. The L^AT_EX₃ Project. Refererad på sida 114.
URL: <http://www.latex-project.org/guides/ltx3info.pdf>.
- Moreland, Kenneth (2009). *Diverging Color Maps for Scientific Visualization*. Refererad på sida 74.
URL: <http://www.sandia.gov/~kmorel/documents/ColorMaps/ColorMapsExpanded.pdf>.
- Niederberger, Clemens (2013). *acro. Typeset Acronyms*. Version 1.4a. 2 september 2013. Refererad på sida 35.
CTAN: [macros/latex/contrib/acro/acro_en.pdf](http://www.ctan.org/macros/latex/contrib/acro/acro_en.pdf).

Litteraturförteckning

- Oetiker, Tobias m. fl. (2011). *The Not So Short Introduction to L^AT_EX 2_ε*. L^AT_EX 2_ε in 157 minutes. April 2011. Refererad på sidorna **E**, **43**, **98**, **134**.
CTAN: [info/lshort/english/lshort.pdf](http://www.ctan.org/info/lshort/english/lshort.pdf).
- Pakin, Scott (2009). *The Comprehensive L^AT_EX Symbol List*. November 2009. Refererad på sida **51**.
CTAN: [info/symbols/comprehensive/symbols-a4.pdf](http://www.ctan.org/info/symbols/comprehensive/symbols-a4.pdf).
- Patashnik, Oren (1988). *B_BT_EXing*. Februari 1988. Refererad på sida **87**.
CTAN: [biblio/bibtex/base/btxdoc.pdf](http://www.ctan.org/biblio/bibtex/base/btxdoc.pdf).
- Pégourié-Gonnard, Manuel (2013). *A guide to LuaL^AT_EX*. Refererad på sida **113**.
CTAN: [info/luatex/lualatex-doc/lualatex-doc.pdf](http://www.ctan.org/info/luatex/lualatex-doc/lualatex-doc.pdf).
- The R Core Team (2013). *R: A Language and Environment for Statistical Computing*. Version 3.0.1. Maj 2013. Refererad på sida **83**.
URL: <http://cran.r-project.org/doc/manuals/fullrefman.pdf>.
- Redaelli, Massimo A. (2012). *CircuitikZ*. Version 0.3.0. 30 december 2012. Refererad på sida **77**.
CTAN: [graphics/pgf/contrib/circuitikz/circuitikzmanual.pdf](http://www.ctan.org/graphics/pgf/contrib/circuitikz/circuitikzmanual.pdf).
- Robertson, Will (2006). "Productivity with macros and packages". I: *The PracT_EX Journal* 3. Refererad på sida **100**.
URL: <http://tug.org/pracjourn/2006-3/robertson/robertson.pdf>.
- Robertson, Will (2010). *How to avoid large margins around Matlab plot in PDF*. Svar till fråga ställd på *T_EX Stack Exchange*. 18 november 2010. Senast hämtad 2013-10-20. Refererad på sida **83**.
URL: <http://tex.stackexchange.com/a/5562/66>.
- Robertson, Will (2013). *The X_YT_EX reference guide*. Refererad på sida **112**.
CTAN: [info/xetexref/xetex-reference.pdf](http://www.ctan.org/info/xetexref/xetex-reference.pdf).
- Robson, Adrian P. (2013). *The flowchart package. Flowchart Shapes for TikZ*. Version 3.2. 28 januari 2013. Refererad på sida **77**.
CTAN: [graphics/pgf/contrib/flowchart/flowchart.pdf](http://www.ctan.org/graphics/pgf/contrib/flowchart/flowchart.pdf).
- Scharrer, Martin (2011). *The tikz-timing Package. A L^AT_EX Package for Timing Diagrams*. Version 0.7d. 9 januari 2011. Refererad på sida **77**.
CTAN: [graphics/pgf/contrib/tikz-timing/tikz-timing.pdf](http://www.ctan.org/graphics/pgf/contrib/tikz-timing/tikz-timing.pdf).

Litteraturförteckning

- Von Schultz, Christian (2005). *Rekommendationer för L^AT_EX-dokument* (november 2005). Refererad på sida 35.
URL: <http://web.student.chalmers.se/~von/latex/rekommendationer.pdf>.
- Sharpsteen, Charlie och Cameron Bracken (2012). *tikzDevice: R Graphics Output in LaTeX Format*. Version 0.6.3/r48. Refererad på sida 79.
URL: <http://R-Forge.R-project.org/projects/tikzdevice/>.
- Skala, Wolfgang (2013). *Drawing Gantt Charts in L^AT_EX. The pgfgantt Package*. Version 4.0. 1 juni 2013. Refererad på sida 77.
CTAN: [graphics/pgf/contrib/pgfgantt/pgfgantt.pdf](#).
- Språkrådet (2008). *Svenska skrivregler*. 3. utgåvan. Solna: Liber AB. Refererad på sida 37.
ISBN: 978-91-47-08460-9.
- Talbot, Nicola L. C. (2013a). *The glossaries package. A guide for beginners*. (5 juli 2013). Refererad på sida 34.
CTAN: [macros/latex/latex/contrib/glossaries/glossariesbegin.pdf](#).
- Talbot, Nicola L. C. (2013b). *User Manual for glossaries.sty*. Version 3.07. 5 juli 2013. Refererad på sida 34.
CTAN: [macros/latex/contrib/glossaries/glossaries-user.pdf](#).
- Tantau, Till (2010). *The TikZ and PGF Packages*. Version 2.10. Oktober 2010. Refererad på sida 77.
CTAN: [graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf](#).
- Thành, Hàn Thế (2000). "Micro-typographic extensions to the T_EX typesetting system". Doktorsavhandling. Brno: Faculty of Informatics, Masaryk University, oktober 2000. Refererad på sida 14.
URL: <http://www.pragma-ade.com/pdftex/thesis.pdf>.
- Tufte, Edward R. (1990). *Envisioning Information*. Cheshire, Connecticut: Graphics Press LLC. Refererad på sida 72.
ISBN: 0-9613921-1-8.
- Tufte, Edward R. (1997). *Visual Explanations. Images and Quantities, Evidence and Narrative*. Cheshire, Connecticut: Graphics Press LLC. Refererad på sidorna 72, 73.
ISBN: 0-9613921-2-6.

Litteraturförteckning

- Tufte, Edward R. (2001). *The Visual Display of Quantitative Information*. 2. utgåvan. Cheshire, Connecticut: Graphics Press LLC. Refererad på sidorna 72, 73.
ISBN: 0-9613921-4-2.
- Wickham, Hadley (2009). *ggplot2: elegant graphics for data analysis*. New York: Springer. Refererad på sida 75.
ISBN: 978-0-387-98140-6.
- Voß, Herbert (2010). *Math mode*. December 2010. Refererad på sidorna 45, 61, 99.
CTAN: info/math/voss/mathmode/Mathmode.pdf.
- Wright, Joseph (2013). *siunitx — A comprehensive (SI) units package*. Version 2.5r. Juli 2013. Refererad på sida 70.
CTAN: macros/latex2e/exptl/siunitx/siunitx.pdf.

Sakregister

Detta register innehåller en förteckning över olika teman, termer och liknande som diskuteras i boken. Den innehåller inte en förteckning över de \LaTeX -paket som diskuteras (en sådan återfinns på [sida 126](#)) eller de kommandon och miljöer som nämns (en förteckning över dessa finns på [sida 128](#)).

- \$
 - i text, *se* specialtecken
 - matematikläge, *se* matematikläge
- %
 - i \LaTeX -kod, *se* kommentar
 - i text, *se* specialtecken
- &
 - i tabell, *se* tabell
 - i text, *se* specialtecken
- avstavning, [22](#), [103](#), [104](#)
- betoning, [24](#)
- bibliografi, [87](#), [94](#)
 - stil, [91–94](#)
- bindestreck, [36](#)
- bra-ket, [56](#), [110](#)
- Chicago-stil, *se* bibliografistil
- citattecken, [36](#), [41](#), [105](#)
- ConTeXt, [113](#)
- CTAN, [4](#), [13](#), [115](#)
- dokumentklass, [9](#), [10](#)
 - standardklass, [10](#)
- dokumentmiljö, [9](#)
- dollartecken, *se* specialtecken
- ekvation, *se* matematikläge
- enhet, *se* SI-enhet
- etikett, [33](#)
- exponent, *se* superskript
- fetstil, *se* textstil
- figur, [29](#), [72](#), [73](#), [104](#)
 - exportera, [79–83](#)
 - infoga, *se* grafik
- filtyp
 - .aux, [17](#)
 - .cls, *se* dokumentklass
 - .log, [16](#)
 - .pdf, *se* PDF
 - .sty, *se* paket

Sakregister

- `.tex`, 6, 16
- `.toc`, 17
- övriga, 16–17
- flytande objekt, 27, 108
- figur, *se* figur
- tabell, *se* tabell
- fotnot, 41, 105
- framsida, *se* titelsida
- färgschema, 74
- förkortning, 35, 108
- förteckning, *se* register
 - innehålls-, *se*
 - innehållsförteckning
- grafik, 72
 - generera, 77, 106
 - inkludera, 76, 106
- Harvard-stil, *se* bibliografistil
- hyperlänk, 34, 103
- index
 - register, *se* register
 - subskript, *se* subskript
- installera
 - L^AT_EX, 4
 - paket, *se* paket, installera
- inställningar
 - biblatex, 90
 - standardklass, 10, 12
- intervallstreck, 36
- kapitel, *se* rubrik
- kapitälär, *se* textstil
- klass, *se* dokumentklass
- kommando, 8
- kommentar, 9
- kompilator, 2, 14
 - LuaT_EX, 113
 - pdfL^AT_EX, 2, 14
 - X_YL^AT_EX, 112
- kompilera
 - automatiskt, *se* latexmk
 - till PDF, *se* PDF
- korsreferens, 33, 108
 - till ekvation, 47
- kursiv, *se* textstil
- L^AT_EX₃, 114
- latexmk, 15
- lista, 26
- LuaT_EX, *se* kompilator, LuaT_EX
- längd, 42
- matematik
 - absolutbelopp, 58
 - binomialkoefficient, 49
 - bråk, 49
 - derivata, 53
 - gränsvärde, 49
 - hatt, 53
 - icke-kursiv text, 48
 - integral, 58, 59
 - kvadratroten, 49
 - matris, 64
 - operator, 49, 53
 - parantes, 56
 - pi, 53
 - punkter, 68
 - relationssymbol, 53
 - summa, 58
 - symbol, 49

Sakregister

- trigonometri, 49
- variabel, 48
- vektor, 53
- matematikläge, *se även*
 - matematik
 - fristående ekvation, 46
 - i löpande text, 46
 - radbruten ekvation, 61
 - samlade ekvationer, 62
 - styckvis funktion, 65
 - tomrum, *se* tomrum
- mellanrum, *se* tomrum
- miljö, 26
- mått
 - i L^AT_EX, *se* längd
 - i text, *se* SI-enhet
- nyradstecken, *se* tomrum
- och-tecken, *se* specialtecken
- ordlista, 34, 108
- paket, 9, 13
 - bibliografi, 107
 - dokumentation, 13
 - förteckning, 102–111
 - grafik, 106
 - installera, 13
 - matematik, 105
 - viktiga, 13
- pappersstorlek, *se* inställningar,
 - standardklass
- parameter, 8, 9
- PDF, 2, 14, 16, 34, 38, 76, 77, 79, 80, 83, 84, 103, 106, 107, 113
 - kompilera till, 14
- pdfL^AT_EX, *se* kompilator,
 - pdfL^AT_EX
- PGF/TikZ, *se* grafik, generera
- procenttecken, *se* specialtecken
- radbrytning
 - automatisk, *se* avstavning
 - manuell, *se*
 - styckesindelning
- referens
 - databas, 88
 - lista, *se* bibliografi
 - kors-, *se* korsreferens
- register, 35, 109
- resurser, 97–100
 - böcker, 98–99
 - CTAN, *se* CTAN
 - på internet, 100
 - T_EX.SE, *se* T_EX Stack Exchange
- rubrik, 21
- sakregister, *se* register
- sans-serif-typsnitt, *se* typsnitt
- serif-typsnitt, *se* typsnitt
- SI-enhet, 68, 105
- sidbrytning, 22
- specialtecken, 7
- stavningskontroll, 2
- struktur
 - dokument-, 9
 - filsystems-, 100
 - text-, 18–24
- styckesindelning, 19, 37
- subskript, 47

Sakregister

- superskript, 47
- syntax
 - $\text{\texttt{B}}\text{\texttt{E}}\text{\texttt{T}}\text{\texttt{E}}\text{\texttt{X}}$, 87–88
 - $(\text{\texttt{L}}^{\text{\texttt{A}}})\text{\texttt{T}}\text{\texttt{E}}\text{\texttt{X}}$, 6–9
- tabell, 30, 104
- tankstreck, 36
- teckenkodning, 6, 13, 42, 103
 - UTF-8, 6, 13, 42, 104, 106, 112, 113
- $\text{\texttt{T}}\text{\texttt{E}}\text{\texttt{X}}$ Stack Exchange, 100
- textstil
 - fetstil, 24
 - kapitälär, 24
 - kursiv, 24
- titelsida, 39
- tomrum
 - i $\text{\texttt{L}}^{\text{\texttt{A}}}\text{\texttt{T}}\text{\texttt{E}}\text{\texttt{X}}$ -kod, 6
 - i matematikläge, 48, 66
 - matematikläge, 68
 - nyradstecken, 6
- typsnitt, 38
 - icke-proportionellt, 24
 - OTF- och TTF-, 103, 112
 - paket, 104
 - sans-serif, 24
 - serif, 24
 - skärpa, 38
- uppdelning
 - av $\text{\texttt{L}}^{\text{\texttt{A}}}\text{\texttt{T}}\text{\texttt{E}}\text{\texttt{X}}$ -kod, 102
- versionshantering, 101
- $\text{\texttt{X}}_{\text{\texttt{T}}}\text{\texttt{T}}\text{\texttt{E}}\text{\texttt{X}}$, *se* kompilator, $\text{\texttt{X}}_{\text{\texttt{T}}}\text{\texttt{T}}\text{\texttt{E}}\text{\texttt{X}}$

Förteckning över L^AT_EX-paket

Denna förteckning listar alla de L^AT_EX-paket som nämns i boken.

acro, 35, 108, 118	eskdex, 11
algorithmic, 110	expl3, 114, 118
alphabetic, 107	
amsmath, 45, 61, 105, 106, 115	fancyhdr, 110
amssymb, 45, 105	fixltx2e, 103
amsthm, 105	fixmath, 51
article, 11, 12, 21	float, 108
authoryear, 107	flowchart, 77, 119
	fnpct, 42, 105
babel, 22, 23, 40, 103, 104, 112	fontenc, 38, 103, 112
biber, 90	fontspec, 103, 112, 113
biblatex, 34, 87, 89–95, 107, 117, 118	
biblatex-chicago, 91, 93, 94, 107, 117	geometry, 108
book, 11, 12, 21	glossaries, 34, 35, 108, 109, 120
booktabs, 30–32, 78, 104	graphicx, 72, 76, 77, 106
braket, 56, 110	
	hyperref, 34, 103
caption, 105	
circuitikz, 77	ieee, 107
cleveref, 34, 47, 108	ieeetran, 11
csquotes, 36, 41, 105	imakeidx, 35, 109, 115
	inputenc, 42
elsarticle, 11	inputenx, 42, 104, 112
enumitem, 108	isomath, 53, 106
	kpfonts, 104

Förteckning över L^AT_EX-paket

l2tabu, 98
listings, 111
listingsutf8, 111
lmodern, 38, 39, 104
lshort, 98

mathtools, 58, 106
memoir, 11
minted, 111
multicol, 12, 109
multirow, 109

nag, 104
numeric, 107, 108

octavo, 11

parskip, 38
pgfgantt, 77, 120
pgfplots, 77–79, 116
pgfplotstable, 78, 116
phys, 108
polyglossia, 103, 104, 112
pstricks, 107

refman, 11
report, 11, 12, 21

revtex, 11

scrartcl, 11
scrbook, 11
scrreprt, 11
sffms, 11
sidecap, 109
siunitx, 38, 68, 70, 105, 121
skmath, 60, 106
subfig, 109
subfiles, 102

tikz, 106, 107
tikz-timing, 77, 119
titlesec, 110
tocloft, 110
todonotes, 111
tufte-book, 12

unicode-math, 106, 112

varioref, 33, 34, 108, 109

wrapfig, 109

xfrac, 50, 106
xspace, 109

Förteckning över kommandon och miljöer

Denna förteckning listar alla de kommandon och miljöer som diskuteras i boken.

<code>\!</code> , 60, 66	<code>\boldmath</code> , 53
<code>\(</code> , 46	<code>\Braket</code> , 110
<code>\)</code> , 46	
<code>\,</code> , 66	<code>\cap</code> , 55
<code>\:</code> , 66	<code>\caption</code> , 27, 33
<code>\;</code> , 66	<code>cases</code> (miljö), 65
<code>\&</code> , 8	<code>\cdot</code> , 55
<code>\\</code> , 7	<code>\cdots</code> , 64, 68
	<code>\centering</code> , 29
<code>\addcontentsline</code> , 22	<code>\chapter</code> , 21
<code>\aleph</code> , 51	<code>\chi</code> , 52
<code>align</code> (miljö), 62–65	<code>\cleardoublepage</code> , 23
<code>\alpha</code> , 52	<code>\clearpage</code> , 23
<code>\ang</code> , 70	<code>\complement</code> , 51
<code>\approx</code> , 55	<code>\coprod</code> , 59
<code>\autocite</code> , 95	<code>\cos</code> , 49
	<code>\cref</code> , 47, 108
<code>\bar</code> , 54	<code>\cup</code> , 55
<code>\begin</code> , 26, 31	
<code>\beta</code> , 52	
<code>\blockquote</code> , 105	<code>\ddot</code> , 54
<code>Bmatrix</code> (miljö), 64	<code>\ddots</code> , 65
<code>bmatrix</code> (miljö), 64	<code>\DeclareMathOperator</code> , 49, 99

Förteckning över kommandon och miljöer

<code>\DeclarePairedDelimiter</code> , 58, 106	<code>\gets</code> , 54
<code>\Delta</code> , 52, 55	<code>\gg</code> , 55
<code>\delta</code> , 52	<code>\ggg</code> , 105
<code>displayquote</code> (miljö), 41	<code>\Gls</code> , 34
<code>\div</code> , 55	<code>\gls</code> , 34
<code>document</code> (miljö), 10	<code>\Glspl</code> , 34
<code>\documentclass</code> , 9, 10, 12	<code>\glspl</code> , 34
<code>\dot</code> , 54	<code>\gt;sim</code> , 55
<code>\dotsb</code> , 69	<code>\hat</code> , 54
<code>\dotsc</code> , 69	<code>\hbar</code> , 51
<code>\dotsi</code> , 69	<code>\hphantom</code> , 68
<code>\dotsm</code> , 69	<code>\hyphenate</code> , 23
<code>\dotso</code> , 69	<code>\idotsint</code> , 59
<code>\emph</code> , 24, 25	<code>\iff</code> , 54
<code>\emptyset</code> , 51	<code>\iiiint</code> , 59
<code>\end</code> , 26	<code>\iiint</code> , 59
<code>\enquote</code> , 41, 105	<code>\iint</code> , 59, 60
<code>enumerate</code> (miljö), 108	<code>\Im</code> , 51
<code>\epsilon</code> , 51, 52	<code>\in</code> , 53
<code>\eqref</code> , 47	<code>\include</code> , 102
<code>equation</code> (miljö), 60, 62	<code>\includegraphics</code> , 76, 83
<code>\equiv</code> , 55	<code>\includeonly</code> , 102
<code>\eta</code> , 52	<code>\infty</code> , 51
<code>\exists</code> , 51	<code>\input</code> , 80, 102
<code>\footcite</code> , 91, 95	<code>\int</code> , 58, 59
<code>\footnote</code> , 41, 43	<code>\iota</code> , 52
<code>\forall</code> , 51	<code>\item</code> , 27
<code>\frenchspacing</code> , 19	<code>\kappa</code> , 52
<code>\Gamma</code> , 52	<code>\label</code> , 33, 47, 63
<code>\gamma</code> , 52	<code>\Lambda</code> , 52
<code>gather</code> (miljö), 62–64	<code>\lambda</code> , 52
<code>\geq</code> , 55	<code>\land</code> , 55

Förteckning över kommandon och miljöer

<code>\ldots</code> , 68	<code>multline</code> (miljö), 61–63, 67, 68
<code>\left</code> , 56, 57, 68	<code>\nabla</code> , 53, 55
<code>\Leftarrow</code> , 54	<code>\negmedspace</code> , 66
<code>\Leftrightarrow</code> , 54	<code>\negthickspace</code> , 66
<code>\leftrightharpoonarrow</code> , 54	<code>\negthinspace</code> , 66
<code>\leq</code> , 55	<code>\neq</code> , 55
<code>\lessssim</code> , 55	<code>\newcommand</code> , 24
<code>\limits</code> , 59	<code>\newfloat</code> , 108
<code>\ll</code> , 55	<code>\newglossaryentry</code> , 34
<code>\Longleftarrow</code> , 54	<code>\newline</code> , 22
<code>\longleftarrow</code> , 54	<code>\newpage</code> , 22, 23
<code>\longleftrightharpoonarrow</code> , 54	<code>\nexists</code> , 51
<code>\longmapsto</code> , 54	<code>\ngeq</code> , 55
<code>\Longrightarrow</code> , 54	<code>\ngtr</code> , 55
<code>\longrightarrow</code> , 54	<code>\nleq</code> , 55
<code>\lor</code> , 55	<code>\nless</code> , 55
<code>\lVert</code> , 58	<code>\nocite</code> , 94, 95
<code>\lvert</code> , 58	<code>\nonumber</code> , 63
<code>\makeglossaries</code> , 34	<code>\nu</code> , 52
<code>\maketitle</code> , 40	<code>\num</code> , 38, 70
<code>\mapsto</code> , 54	<code>\oint</code> , 59
<code>\mathbb</code> , 53, 105	<code>\Omega</code> , 52
<code>\mathcal</code> , 51	<code>\omega</code> , 52
<code>\mathrm</code> , 48, 49, 52, 60	<code>\overbrace</code> , 49, 50
<code>matrix</code> (miljö), 65	<code>\pageref</code> , 33
<code>\mbox</code> , 22	<code>\parencite</code> , 91, 94, 95
<code>\medspace</code> , 66	<code>\part</code> , 21
<code>\metre</code> , 70	<code>\partial</code> , 55
<code>\mid</code> , 56, 58	<code>\per</code> , 70
<code>\milli</code> , 70	<code>\Phi</code> , 52
<code>\missingfigure</code> , 111	<code>\phi</code> , 52
<code>\mp</code> , 55	<code>\Pi</code> , 52
<code>\mu</code> , 52	<code>\pi</code> , 52
<code>\multfootnote</code> , 42	
<code>multicols</code> (miljö), 109	

Förteckning över kommandon och miljöer

<code>\pm</code> , 55, 70	<code>\subset</code> , 55
<code>\pmatrix</code> (miljö), 64	<code>\subseteq</code> , 55
<code>\pno</code> , 95	<code>\sum</code> , 58, 59
<code>\ppno</code> , 95	<code>\supset</code> , 55
<code>\printglossaries</code> , 34	<code>\supseteq</code> , 55
<code>\prod</code> , 59	
<code>\protect</code> , 43	<code>\tableofcontents</code> , 22
<code>\Psi</code> , 52	<code>\tabular</code> (miljö), 63, 64
<code>\psi</code> , 52	<code>\tau</code> , 52
<code>\psq</code> , 95	<code>\tensorsym</code> , 106
<code>\psqq</code> , 95	<code>\text</code> , 48
	<code>\textbackslash</code> , 7
<code>\qquad</code> , 66	<code>\textbf</code> , 25
<code>\quad</code> , 66	<code>\Textcite</code> , 95
<code>\quotation</code> (miljö), 41	<code>\textcite</code> , 91, 95
<code>\quote</code> (miljö), 41	<code>\textit</code> , 25
	<code>\textnormal</code> , 25
<code>\Re</code> , 51	<code>\textquote</code> , 41
<code>\ref</code> , 33, 47, 108	<code>\textrm</code> , 25
<code>\rho</code> , 52	<code>\textsc</code> , 24, 25
<code>\right</code> , 56, 57, 68	<code>\textsf</code> , 24, 25
<code>\Rrightarrow</code> , 54	<code>\texttt</code> , 25
<code>\rVert</code> , 58	<code>\textwidth</code> , 42
<code>\rvert</code> , 58	<code>\thanks</code> , 40
	<code>\therefore</code> , 105
<code>\SCfigure</code> (miljö), 109	<code>\Theta</code> , 52
<code>\section</code> , 33	<code>\theta</code> , 52
<code>\Set</code> , 111	<code>\thickspace</code> , 66
<code>\sfrac</code> , 106	<code>\thinspace</code> , 66
<code>\SI</code> , 68, 70	<code>\tilde</code> , 54
<code>\Sigma</code> , 52	<code>\times</code> , 55
<code>\sigma</code> , 52	<code>\to</code> , 54
<code>\sin</code> , 49	
<code>\sisetup</code> , 70	<code>\underbrace</code> , 49, 50
<code>\smallskipamount</code> , 43	<code>\Upsilon</code> , 52
<code>\subequations</code> (miljö), 63, 64	<code>\upsilon</code> , 52

Förteckning över kommandon och miljöer

<code>\usepackage</code> , 9, 13	<code>Vmatrix</code> (miljö), 64
	<code>vmatrix</code> (miljö), 64
<code>\varepsilon</code> , 51, 52	<code>\pageref</code> , 33
<code>\varnothing</code> , 51	<code>\vphantom</code> , 67, 68
<code>\varphi</code> , 52	<code>\vref</code> , 33, 47, 109
<code>\varpi</code> , 52	
<code>\varrho</code> , 52	<code>\widehat</code> , 54
<code>\varsigma</code> , 52	<code>\write18</code> , 35
<code>\vartheta</code> , 52	
<code>\vdots</code> , 64	<code>\Xi</code> , 52
<code>\vec</code> , 53, 54	<code>\xi</code> , 52
<code>\vectorsym</code> , 106	<code>\xspace</code> , 109
<code>\vert</code> , 58	<code>\zeta</code> , 52

Del VI.

Appendix

A. En enkel mall

Följande mall är tänkt att användas för att skriva enklare rapporter på Chalmers Tekniska Högskola. Eftersom \LaTeX -installationen på deras datorer saknar en del paket (siunitx och chscite) är dessa bortkommenterade. Vill man använda dem kan man installera dessa i sin hemmapp, under $\sim/\text{texmf}/$.¹

Kommentarerna och texten i mallen bör förklara och motivera vad som görs vilka paket som inkluderas någorlunda. Om något är oklart, referera till motsvarande del av den här introduktionen. Hela mallen finns dessutom tillgänglig som en `.tex`-fil för nedladdning.²

```
\RequirePackage[l2tabu]{nag}
\documentclass[a4paper,11pt]{scrartcl}
% Allmängiltiga paket
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[swedish]{babel}
\usepackage{microtype}
\usepackage{graphicx}
\usepackage{fixltx2e}
% Matematikpaket
\usepackage[intlimits]{amsmath}
\usepackage{amssymb}
\usepackage{mathtools}
```

¹ Detta förklaras närmre av Oetiker m. fl. (2011, ss. 89–90)

² <https://raw.githubusercontent.com/urdh/latexbok/master/examples/A/mall.tex>

A. En enkel mall

```
%\usepackage{siunitx} % finns ej på Chalmers-datorer
\usepackage{textcomp}
% Snyggare tabeller
\usepackage{booktabs}
% Snyggare figurtexter
\usepackage[font=small,%
             format=plain,%
             labelfont=bf,%
             textfont=it]{caption}
% Länkade och intelligenta referenser
\usepackage{hyperref}
\usepackage{cleveref}
% Källförteckning
\usepackage[style=authoryear]{biblatex}

% Inställningar för siunitx
%\sisetup{locale=DE}

% Kommandodefinitioner (finns även i paketet skmath)
% \abs{}, \norm{} – absolutbelopp och norm
% \frac omdefinierad för att se bättre ut
% \d{} – integraldifferentialen
\makeatletter
\DeclarePairedDelimiter\abs{\lvert}{\rvert}
\DeclarePairedDelimiter\norm{\lVert}{\rVert}
\renewcommand{\frac}[2]{\genfrac{}{}{}{}{}%
                      {\displaystyle #1}{\displaystyle #2}}
\renewcommand\d[1]{\ensuremath{\;\mathrm{d}\,#1}%
                  \@ifnextchar\d{\!}{}}
\makeatother

% Dokumentets metadata
\title{Ett exempelldokument}
\author{Simon Sigurdhsson}
\date{6 augusti 2013}
```


A. En enkel mall

```
% Här börjar dokumentet
\begin{document}
  \maketitle
  \begin{abstract}
    Man bör alltid ha en kort sammanfattning (eng.
    \emph{abstract}) till sina rapporter och
    artiklar. En sådan skapas med omgivningen
    \texttt{abstract}.
  \end{abstract}

  % En innehållsförteckning vill man ha ibland:
  \tableofcontents
  \newpage

  \section*{Introduktion}
  Det här avsnittet är tänkt att innehålla en kort
  introduktion till rapporten som förklarar dess syfte
  och mål. Man behöver så klart inte alltid en
  introduktion, men om man har en så bör den vara
  onummerad, och ska således skapas med
  \texttt{\textbackslash section*}.

  \section{Första avsnittet}
  Som du ser kan man radbryta sin text i \LaTeX{} utan
  att det påverkar resultatet. Gör det --- det blir mer
  lättläst. Vissa textredigerare kan göra detta
  automatiskt.

  Det fungerar i ekvationer också:
  \begin{equation}
    f(x) = \sin x
    \implies
    F(x) = \int\! f(x)\mathrm{d}x = \cos x + c.
  \end{equation}
```

A. En enkel mall

```
\subsection{Ett underavsnitt}
Man kan även infoga tabeller i \LaTeX{} (se tabell
\ref{tab:priser}, till exempel) och figurer (figur
\ref{fig:rulltarta}). Dessa kan man referera till på olika
sätt, till exempel med kommandot
\texttt{\textbackslash cref}: ``\cref{fig:rulltarta}``.

\begin{table}[tp]
  \centering
  \caption{En tabell över fiktiva priser}
  \label{tab:priser}
  \begin{tabular}{lp{0.4\textwidth}r}
    \toprule
    Namn & Beskrivning & Pris (\$) \\
    \midrule
    Cykel & Enkelt tvåhjuligt fordon utan motor
    för transport av enstaka personer över korta
    sträckor & 200 \\
    Polkagris & Röd-vit-randig sötsak med mycket
    hård konsistens, tillverkad i Gränna & 2,50 \\
    \bottomrule
  \end{tabular}
\end{table}

\begin{figure}[bp]
  \centering
  % \includegraphics[width=0.8\textwidth%
  % ]{bilder/rulltarta.png}
  \caption{En mycket god rulltårta}
  \label{fig:rulltarta}
\end{figure}
\end{document}
```