Typsättning av text och matematiska formler med hjälp av LaTeX

En handledning för handfallna

Andreas Kähäri

v 3.0
Juni 2000

http://latexintro.sourceforge.net/

Sammanfattning

ÄRADE LÄSARE: Detta kompendium är tänkt som en handledning för personer som aldrig förut har kommit i kontakt med LaTeX men som snabbt vill komma igång med en handfull enkla exempel. Kompendiet fungerar också med fördel som referensmaterial då minnet sviker den tankspridde TeXperten.

Förutom en steg-för-steg-genomgång av hur man på ett UNIX- eller UNIX-liknande system går tillväga för att producera enkla dokument med hjälp av LaTeX, tar detta kompendie upp de mest grundläggande momenten som krävs för att skriva vanlig text och enkla matematiska formler i LaTeX.

Lite mer avancerade moment, så som radbrytning i formler, referenser inom ett dokument, tabeller, figurer och egendefinierade kommandon, tas också upp för att tillgodose den som vill använda LaTeX för att skriva längre och mer professionella texter.

För den som vill använda LaTeX till att skriva större rapporter eller avhandlingar finns dessutom ett kapitel om hur man på enklaste sätt skapar och underhåller litteraturförteckningar.

Tips: I texten finns stiltips insprängda för att påminna om vad som anses som god stil vid skrivandet av LaTeX-dokument och texter i allmänhet. Stiltips visas i rutor med rundade hörn.

OBS: På samma sätt finns varningar och andra viktiga noteringar insprängda i texten där det kan behövas en extra påminnelse om något som kanske kan vara lätt att glömma bort eller kan ses som underförstått eller tvetydigt. Varningar och viktiga noteringar visas i rutor med dubbla kanter.

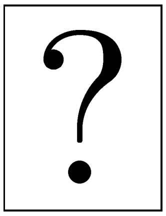
Läsarna av denna handledning uppmanas att på egen hand pröva de kommandon och konstruktioner som förekommer i kompendiet på sina egna rapporter, inlämningsuppgifter, meritlistor, brev och inköpslistor, för att på så sätt få in känslan av hur LaTeX fungerar.

Det är väl bäst att säga det på en gång: LaTeX lär man sig absolut snabbast genom praktiska övningar.

Innehåll

Kapitel 1

Vad är LaTeX?



LaTeX är vad man brukar kalla ett ”typsättningssystem” (eng. *document preparation system*), speciellt utarbetat för att typsätta böcker (skön- eller facklitteratur), rapporter, artiklar, brev, meritförteckningar och *allt annat som kan skrivas på papper*.

Vi vill alltså understryka att man inte alls behöver skriva vetenskapliga rapporter med massor av matematiska formler för att ha anledning att använda sig av LaTeX. Vardagliga texter går också utmärkt att skriva. Däremot förenklar LaTeX skrivandet av formler en hel del, vilket vi ska titta närmare på i ett senare kapitel.

Med ”typsättningssystem” menas ett system med vilket man kan typsätta texter. Ordet ”typsättning” är en term som kommer från tryckerikonsten. På en del tryckerier ”sätter” man ramar med ”typer” som bestryks eller doppas i trycksvärta för att sedan tryckas mot ett papper. En ”typ” är en mall i metall vari ett tecken, d.v.s. en bokstav, siffra eller en annan krumelur som man vill ska finnas på en tryckt sida, är gjuten eller snidad.

Numera sker typsättning inte så ofta med hjälp av ramar med typer som trycks mot papper, utan i stället med hjälp av stora skrivare som kan skriva ut flera tusen sidor per minut. En sådan skrivare behöver en bild av hur sidorna som den ska skriva ut ska se ut, och LaTeX är ett ypperligt verktyg för att göra bl.a. detta.

1.1 Lite historia

LaTeX har en historia som sträcker sig från 1977 och framåt i tiden. Här följer några viktiga hållpunkter i denna historia:

1977 Donald Knuth skriver typsättningssystemet TeX som blir en stor succé hos forskare över hela världen. TeX blir *de facto*-standard för framställning av avhandlingar, rapporter och andra dokument vid många universitet.

Knuth skriver, på första sidan i sin bok *The TeXbook* [Knu87],

> This is a handbook about TeX, a new typesetting system intended for the creation of beautiful books—and especially for books that contain a lot of mathematics.

ca. 1980 Leslie Lamport skriver ett makropaket för TeX som han kallar LaTeX [Lam94]. Detta LaTeX har en högre abstraktionsnivå än vad TeX hade, och blir på så sätt lättare att sätta sig in i och förstå.

1994 *The LaTeX3-team*, som är en samling personer som arbetar ideellt med att utveckla LaTeX, ger ut en ny och förbättrad version av LaTeX som får namnet LaTeX 2ε [GMS94]. Denna version ersätter det gamla LaTeX, som också kallas ”LaTeX 2.09”[1].

LaTeX finns nu för många specialområden. Det finns t.ex. speciella makropaket för att typsätta musik, programkod och kemiska formler, men också för att skriva hebreiska, farsi, hindi och arabiska.

Arbete pågår med att utveckla LaTeX3, som är tänkt att ersätta LaTeX i framtiden.

1.2 Att uttala och skriva ”LaTeX”

Många nybörjare har problem med hur man egentligen ska uttala TeX, LaTeX och liknande namn som förekommer i denna handledning. Donald Knuth skriver i *The TeXbook*:

> Insiders pronounce the χ in TeX as a Greek chi, not as ‘x’, so that TeX rhymes with the word blecchhh. It’s the ‘ch’ sound in Scottish words like *loch* or German words like *ach*; it’s a Spanish ‘j’ and a Russian ‘kh’. When you say it correctly to your computer, the terminal may become slightly moist.

När Lamport utvecklade LaTeX 2.09 gav han aldrig någon vink om hur det skulle uttalas, men många säger ”Lah-TeX”, eller kanske ”Ley-TeX”. Epsilonet (ε) i LaTeX 2ε är menat att symbolisera en förbättring jämtemot LaTeX 2.09, men många brukar ändå säga ”LaTeX-två-e”, eller helt enkelt bara ”LaTeX”.

Vill man, i LaTeX, typsätta logotyperna ”TeX” och ”LaTeX” så ska de fördefinierade kommandona `\TeX{}` och `\LaTeX{}` användas.

Vill du skicka e-post till dina kompisar för att säga att du kan använda LaTeX så ska du skriva ”`LaTeX`” och *inte* ”`Latex`” eller något likande. Detta är i det närmaste en religiös fråga.

[1] Vi kommer inte att referera till denna äldre version av LaTeX mer i denna text och ”LaTeX” får därför beteckna den senare varianten av systemet.

Kapitel 2

Hur gör man?

LaTeX är, liksom HTML[1], ett s.k. "markup"-språk. Därför kan det vara bra att veta vad som menas med "markup".

2.1 Markup

"Markup" brukade man ursprungligen kalla de tecken och symboler som man skrev på utkast till publikationer för att tala om att "här ska det vara kursiv stil", "gör ett nytt stycke här", "denna rad ska vara indragen" eller liknande. Men markup-symboler förekommer också på en mycket mer grundläggande nivå i alla skrivna språk.

[1] Om du inte vet vad HTML är, eller om du inte behärskar HTML, kan du med gott samvete förbise alla jämförelser mellan LaTeX och HTML i denna handledning.

I en text skriven t.ex. på svenska, talar avstånden mellan orden om att orden ska vara separerade från varandra och att de inte ska läsas ihop till ett ord. En punkt betyder att en mening har tagit slut och att en ny mening börjar. Bindestreck talar om att ett ord är sammansatt av två eller flera ord. Detta är också en form av markup.

Någonting som *inte* är en markup-symbol är radslut, eftersom ett radslut i sig inte ger texten ett innehåll. Radslut förekommer bara i en text eftersom papper i allmänhet har en begränsad bredd. Därför tar vanligtvis markup-språk inte någon hänsyn till radslut. Detta är fallet för både HTML och LaTeX och vi kommer senare att se på exempel som illustrerar detta.

Låt oss för ett ögonblick återgår till de tecken och symboler som man skrev på utkast till t.ex. en publikation. Om ett ord skulle skrivas med **fet stil** brukade man markera ordet i utkastet och i marginalen skriva "fet" (eller någonting liknande). I det gamla programmet *WordStar* för PC skrev man `^B` runt de ord som skulle skrivas ut med fet stil. Det lite modernare ordbehandlingsprogrammet *Microsoft Word* för bl.a. PC och Macintosh visar de feta orden med fet stil på skärmen och kan dessutom visa stycketecken, sidbrytningar m.m. med speciella symboler. Allt detta är exempel på olika typer av markup-symboler.

Programmet *Microsoft Word* är ett exempel på en WYSIWYG[2]-ordbehandlare, men det betyder inte på något sätt att den inte använder sig av markup-symboler. Markup-symbolerna finns osynliga i texten och lagras tillsammans med textmassan då man sparar sitt arbete till diskett eller hårddisk. När man sedan öppnar sitt arbete igen, kommer symbolerna att läsas in tillsammans med texten.

Sammanfattningsvis kan man säga att markup är *att tillhandahålla information tillsammans med textmassan som säger något om textens struktur eller utseende.*

2.1.1 Två typer av markup

I vid mening kan man säga att det finns två typer av markup. En typ talar om hur en text ska *se ut*, en annan talar om vad texten *betyder*. På engelska brukar man skilja på *appearance* och *function*, eller *presentational markup* och *logical/structural markup*.

Ett enkelt ordbehandlingsprogram använder sig för det mesta av *presentational markup* för att tala om att ord ska vara feta, kursiverade eller understrukna. Detta görs oftast genom att låta användaren markera ett stycke text och sedan trycka på en knapp eller liknande.

Ett exempel på en bit text som är skriven med någon sorts *presentational markup*-språk skulle kanske se ut någonting så här:

```
[ny sida]
[börja centrera]
[stor text "Introduktion"]
[sluta centrera]
[1cm vertikalt avstånd]
[indrag 0.5cm]Det är viktigt att [fet stil "alla"]
lär sig [fet stil "LaTeX"].
[börja kursiv stil]De som inte lär sig LaTeX riskerar att bli
lämnade efter i utvecklingen av modern
typsättning[sluta kursiv stil].
```

Som synes talar markup-symbolerna (eller kanske snarare, kommandona) om precis hur texten ska se ut då det slutliga dokumentet skrivs ut, men inte någonting om att t.ex. texten `Introduktion` ska vara en rubrik.

[2] Den lite lustiga förkortningen "WYSIWYG" står för *What you see is what you get* och innebär att det du ser på skärmen när du arbetar med ordbehandlingsprogrammet är precis vad som kommer att komma ut på papper när du skriver ut ditt arbete.

Med *logical/structural markup* talar man inte om precis hur enstaka ord ska se ut, utan i stället hur ett dokument i mer generellare termer ska vara utformat. Man kan t.ex. tala om att alla stycken ska börja med ett indrag på 5 millimeter (om de inte föregås av en rubrik) och att rubriker ska sättas med stor stil. I själva dokumentet säger man sedan om en text ska vara rubrik, stycke, numrerad lista eller dylikt. Med *logical/structural markup* talar man inte heller om att t.ex. betonad text ska vara kursiv eller understruken, bara att den ska vara betonad på något sätt.

En bit text som är skriven med någon sorts *logical/structural markup*-språk skulle kanske kunna se ut någonting i stil med

```
[börja kapitel]
[rubrik "Introduktion"]
[börja stycke]Det är viktigt att [betona "alla"]
lär sig [teknisk term "LaTeX"].
[börja betona]De som inte lär sig [teknisk term "LaTeX"] riskerar
att bli lämnade efter i utvecklingen av modern
typsättning[sluta betona].
[sluta stycke]
[sluta kapitel]
```

Här är nu texten ”Introduktion” en rubrik, men vi vet ingenting om hur rubriken skall sättas (stor stil, etc.). Det krävs alltså fortfarande information om hur t.ex. rubriker och stycken ska utformas.

LaTeX kombinerar *presentational markup* med *logical/structural markup* på ett sätt som gör att man med speciella kommandon kan säga att en viss text ska vara t.ex. kursiv och att en annan text ska vara en rubrik eller en lista. LaTeX har fördefinierade mallar som beskriver hur rubriker, stycken, innehållsförteckningar etc. ska se ut[3].

2.2 Ett första exempel

Hur skriver man då ett dokument med hjälp av LaTeX? Det som i HTML kanske skulle ha skrivits som[4]

```
<html>
<body>
Hello World!
</body>
</html>
```

skulle kunna se ut så här om man skrev det i LaTeX:

```
\documentclass[a4paper]{article}
\begin{document}
Hello World!
\end{document}
```

Detta kanske var rena grekiskan, men misströsta ej! Grekiska kommer vi till i kapitlet som handlar om matematik.

Vad exemplet ovan gör är ingenting annat än att skriva texten `Hello World!` överst och sidnumret (en etta) längst ner i mitten på en för övrigt tom sida, d.v.s. ungefär samma sak som HTML-koden innan (förutom det där med sidnumret).

En lite noggrannare analys kanske är till hjälp:

[3] I HTML kallas sådana mallar för *style sheets*.

[4] Detta är illegal HTML-kod eftersom `<head>` och `<title>` saknas, men det är aningen irrelevant för tillfället.

- `\documentclass` talar om att detta verkligen är ett LaTeX-dokument och måste *alltid* finnas på den första raden i alla LaTeX-dokument.

  Man kan säga att detta motsvarar `<html>` i ett HTML-dokument.

- `[a4paper]` är ett argument till kommandot `\documentclass` som talar om för LaTeX att vi vill typsätta vår text på A4-papper.

  De standardstorlekar på papper som finns definierade är följande:

  – `letterpaper` ($8\frac{1}{2} \times 11$ inch)

  – `legalpaper` ($8\frac{1}{2} \times 14$ inch)

  – `executivepaper` ($7\frac{1}{4} \times 10\frac{1}{2}$ inch)

  – `a4paper` (210×297 mm)

  – `a5paper` (148×210 mm)

  – `b5paper` (176×250 mm)

- `{article}` är ännu ett argument till `\documentclass` som talar om att vi vill använda ett s.k. fördefinierat stildokument som heter `article`. Man måste alltid ange *exakt ett* stildokument.

  Det finns ett antal olika stildokument som t.ex. `article`, `report`, `book` och `letter`. Stildokumentet `article` innehåller exempelvis information om hur olika sorters rubriknivåer ska se ut i en vetenskaplig artikel och är nog det stildokument som används flitigast för typsättning av inte alltför avancerade texter. Stildokumenten `report`, `book` och `letter` innehåller en del finesser som vi inte kommer att gå in på i denna handledning.

  Detta kompendium använder sig av stildokumentet `report`, så en del saker kan kanske se lite annorlunda ut än vad man skulle ha förväntat sig om man använder `article`[5] (detta gäller bl.a. numreringen av ekvationer, tabeller och figurer).

- `\begin{document}` talar om att dokumentets innehåll börjar. Denna rad måste också alltid förekomma i ett LaTeX-dokument.

- Texten `Hello World!` är den text som vi vill typsätta och i slutänden få ut på den tryckta eller utskrivna sidan.

- `\end{document}` talar om att innehållet i dokumentet är slut och att inget mer kommer. LaTeX ignorerar glatt allt som kommer efter `\end{document}`.

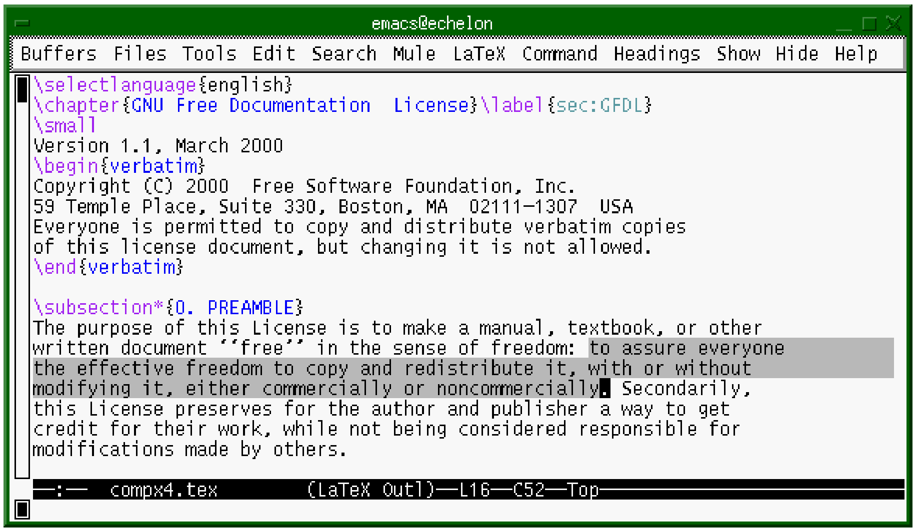
> **OBS:** Notera att man *måste* ange, med `\end{document}`, att innehållet i dokumentet är slut. Detta är en regel som inte bara gäller `\begin{document}` utan faktiskt *alla* liknande konstruktioner i LaTeX.

> **OBS:** Vi kommer i fortsättningen att prata om ett dokuments ”huvud” och ”kropp”. Med huvudet och kroppen menar vi då det som står *före* resp. *efter* raden `\begin{document}` i ett LaTeX-dokument.

[5] Pröva gärna senare med andra stildokument, för att se skillnaden mellan dem.

2.3 Hur går det till?

För att skriva ett dokument med hjälp av LaTeX börjar man med att starta en texteditor. Vilken texteditor som helst går bra att använda och det spelar ingen roll på vilket datorsystem man skriver sitt dokument. På en PC kan man till och med använda *Microsoft Word*, så länge man tänker på att spara sin dokumentfil som en *ren* textfil. En del implementationer av LaTeX tillhandahåller en egen intern editor (som t.ex. det kommersiella programmet TEXTURES för Macintosh). På ett UNIX-system (som t.ex. det kommersiella *Solaris* eller det fria *Linux*) kan man med fördel använda sig av någon av editorerna *Emacs* eller *XEmacs*, se figur 2.1.



Figur 2.1: Editorn *Emacs* lämpar sig ypperligt för redigering av LaTeX-dokument på ett UNIX-system.

Efter att ha skrivit texten i exempelvis *Emacs*, sparar man den som en textfil vars namn slutar med suffixet `.tex` vilket betyder att filen är en dokumentfil som ska typsättas med LaTeX. Vårt lilla exempel på sidan 5 skulle vi kunna spara i en fil med namnet `HelloWorld.tex`.

För att sedan få ut en färdig sida med sin text på är det ett antal steg som man måste gå igenom. Dessa steg är:

1. TeX:a dokumentet.
2. Kontrollera att det ser bra ut.
3. Skapa en Postscript-fil.
4. Skriv ut dokumentet.

Vi ska nu gå igenom dessa steg, ett efter ett, och se varför man måste göra dem.

Tips: Om du är en LaTeX-nybörjare kan det vara en god idé att följa instruktionerna som vi nu ska gå igenom samtidigt som du läser i handledningen. Börja med att skriva in exemplet på sidan 5 och spara det i en fil med namnet `HelloWorld.tex` på din dator.

2.3.1 TEX:a dokumentet

Det som händer då man ”TEX:ar” ett LATEX-dokument kan vagt jämföras med att en WWW-browser (t.ex. *Mozilla* eller *Lynx*) läser in ett HTML-dokument.

Detta steg utförs så att alla rubriker och all text hamnar där de ska. Sid- och radbrytningar görs. Eventuell numrering av rubriker, sidor, listor och figurer m.m. utförs.

För att TEX:a ett LATEX-dokument ger man, på ett UNIX-system, följande kommando i ett kommandofönster:

```
latex HelloWorld.tex
```

Här är `HelloWorld.tex` namnet på den fil som vi skapade med texteditorn. Den kan t.ex. innehålla exemplet på sidan 5.

> **OBS:** Notera att man i olika versioner av LATEX på andra operativsystem än UNIX kan behöva ge något annat sorts kommando för att TEX:a sitt dokument än vad vi sagt här. Det kan t.ex. finnas ett menyalternativ i någon meny som fungerar på samma sätt som UNIX-kommandot ovan. Så är t.ex. fallet med OzTEX för Macintosh.

När du säger åt LATEX att gå igenom din dokumentfil produceras ett antal nya filer i samma katalog som din dokumentfil fanns i. En del av dessa nya filer finns beskrivna i tabell 2.1. Det är DVI-filen vi är mest intresserade av just nu.

Filsuffix	Innehåll
`.log`	En logfil som innehåller information om den senaste TEX:ningen av dokumentfilen.
`.aux`	En fil som bl.a. innehåller information som senare kan komma att användas för att skapa en innehållsförteckning m.m.
`.dvi`	En fil som innehåller information om vilka typsnitt som används och om hur bokstäverna ska placeras på skärmen eller på pappe-ret.

Tabell 2.1: Några av de olika filer som LATEX skapar då ett dokument TEX:as.

2.3.2 Kontrollera att det ser bra ut

Förkortningen ”DVI” står för *Device Independent* och DVI-filen kan, som namnet antyder, flyttas mellan alla olika datorsystem där TEX eller LATEX finns installerat.

Filen med suffixet `.dvi` är en DVI-fil som du kan visa på skärmen med hjälp av programmet *XDvi* på följande sätt:

```
xdvi HelloWorld.dvi
```

> **OBS:** På andra system än UNIX kan det hända att LATEX automagiskt visar DVI-filen, eller att något som motsvarar *XDvi* finns i en meny någonstans.

På ett UNIX-system bör man, då man givit detta kommando i ett kommandofönster, få upp ett fönster liknande det i figur 2.2 på nästa sida.

Figur 2.2: *XDvi* ger en klar bild av hur ditt dokument kommer att se ut i utskriven form.

Med *XDvi* kan man kontrollera att all text ligger där man förväntade sig att det skulle ligga och att formler och rubriker har hamnat rätt. Man kan trycka på `<n>` eller `<space>` för att visa nästa sida, och på `<p>` eller `<del>` för att gå till föregående sida. Man kan också trycka ned en musknapp i fönstret för att se en förstoring av ett område på sidan. Tryck på `<q>` för att avsluta *XDvi*.

> **OBS:** Notera att *XDvi* kanske inte visar bilder helt korrekt. Bildernas färg, rotering och skalning kan vara missvisande och äldre versioner av *XDvi* kan inte visa bilder i LaTeX-dokument alls.

> **Tips:** Det är bra att ta som vana att alltid granska sitt dokument i *XDvi* när man har TeX:at det. Gå inte vidare till nästa steg (”Skapa en Postscript-fil”) innan du har gjort det.

2.3.3 Skapa en Postscript-fil

För att få ut den slutliga produkten som ett dokument på ett papper är det en sak till som vi måste göra. DVI-filer kan nämligen inte skrivas ut direkt på en skrivare, trots att de går att visa på skärmen.

Postscript är ett språk som de flesta skrivare förstår och som talar om för skrivaren hur sidorna som den ska skriva ut ska se ut.

För att omvandla en DVI-fil till en Postscript-fil måste vi ge följande kommando i ett kommandofönster:

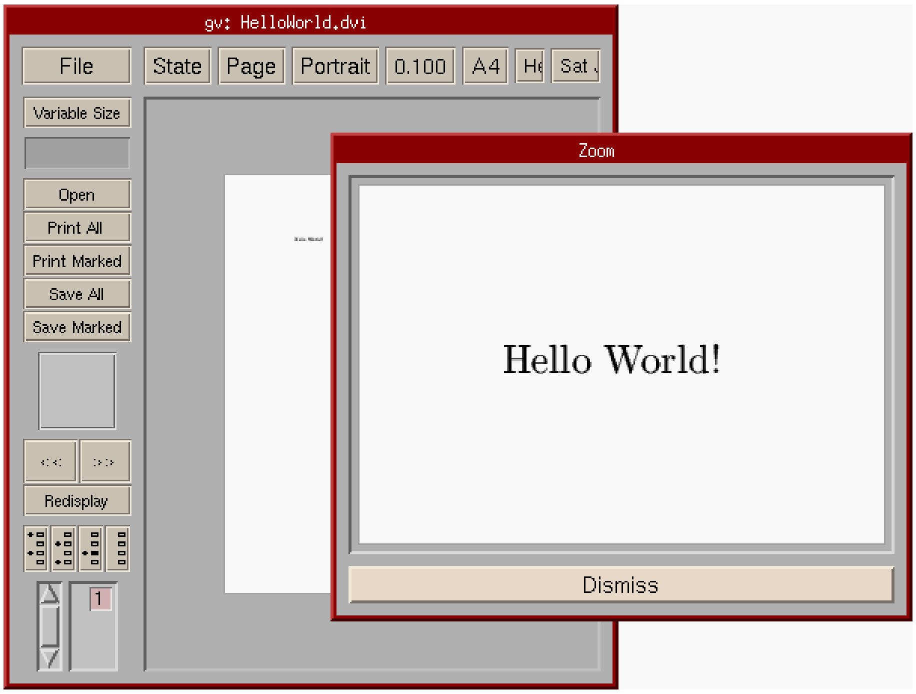
```
dvips HelloWorld.dvi -o HelloWorld.ps
```

eller bara

```
dvips HelloWorld.dvi
```

Här är `HelloWorld.dvi` namnet på den DVI-fil som skapades då vi TeX:ade dokumentet och som vi nu vill omvandla till en Postscript-fil. Det efterföljande `-o HelloWorld.ps` talar om att vi vill skapa en Postscript-fil som heter `HelloWorld.ps`. Utelämnar man `-o HelloWorld.ps` skapas antingen en Postscript-fil eller också så skrivs filen ut på skrivaren direkt, beroende på hur systemet är konfigurerat.

Postscript-filen kan man faktiskt också titta på, ungefär på samma sätt som då man tittade på DVI-filen, men nu med hjälp av ett annat program. På de flesta UNIX-system finns något av kommandona `ghostview`, `pageview` eller `gv` som man kan använda till detta. Programmet `gv` finns, förutom till UNIX-system, även att få tag på till både IBM-kompatibla PC och till Macintosh. Se figur 2.3.



Figur 2.3: Med hjälp av programmet `gv` kan man titta på Postscript-filen som `dvips` genererar.

Om allting är klart, och om dokumentet såg bra ut då det granskades med *XDvi*, är vi redo för att skriva ut dokumentet.

2.3.4 Skriv ut dokumentet

Följande kommando skickar Postscript-filen till skrivare `lw446`[6]:

```
lp -d lw446 HelloWorld.ps
```

Efter ett par sekunder bör dokumentet komma ut på skrivaren.

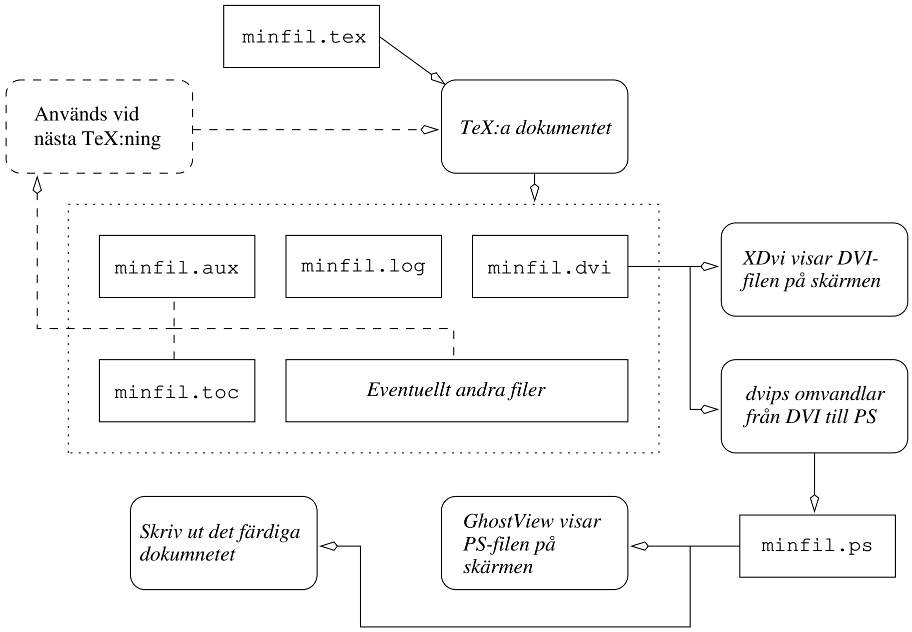
2.4 Hela proceduren som ett diagram

I figur 2.4 på nästa sida har vi försökt visa den procedur som här beskrivits då man TEX:ar en fil med namnet `minfil.tex`.

1. Filen TEX:as varvid det skapas ett antal nya filer. Några av dessa filer hittar du information om senare i de kommande avsnitten.

   AUX-filen (och kanske även andra filer) används vid nästa TEX:ning för att uppdatera en ev. innehållsförteckning och referenser.

[6]Tag kontakt med en lokal guru för att få reda på vad utskriftskommandot på ditt system heter och vilka Postscript-skrivare som finns tillgängliga.



Figur 2.4: Hela proceduren som ett diagram.

2. DVI-filen kan visas på skärmen med hjälp av *XDvi* och/eller göras om till en Postscript-fil med hjälp av `dvips`.
3. Postscript-filen kan skrivas ut på papper eller visas på skärmen (med hjälp av t.ex. `gv`).

På ett UNIX-system jobbar man alltså dels i texteditorn (*Emacs* kanske) och dels genom att ge kommandon till operativsystemet. De kommandon som vi som snabbast har gått igenom hittills har varit `latex`, `xdvi`, `dvips` och `lp`.

Kapitel 3

Vanlig text

Nu ska vi se på hur vanlig text kan skrivas i LaTeX. Vi börjar med ett enkelt exempel och tittar sedan på lite mer avancerade funktioner.

3.1 Enkel vanlig text, textstorlekar m.m.

Här kommer ett exempel på ett enkelt LaTeX-dokument:

```
\documentclass[a4paper, 12pt]{article}
\usepackage[latin1]{inputenc}

\begin{document}
                              % Ett kort Puh-citat
{\Large Puh}, ett citat:

När  man  är  en  Björn,  med   Mycket   Liten Hjärna
och  Tänker Ut Saker, upptäcker  man ibland att en Idé
som   verkade  vara  riktigt Idéaktig  inne i hjärnan,
är \emph{helt} annorlunda när den kommer ut i det fria
och  andra  människor  ser  på.
\end{document}
```

I detta exempel har vi lagt till `12pt` i argumenten till `\documentclass[...]{...}`. Det betyder att de normala bokstävernas höjd ska vara 12 punkter. En ”punkt” är 0.351 millimeter och man har att välja mellan 10, 11 och 12 punkters storlek på vanlig text. Vanligast är att man använder sig av 10 punkters storlek på texten, vilket också är det förvalda värdet som används då man utelämnar textstorleken som vi gjorde i exemplet på sidan 5. Att det inte finns t.ex. `9pt`, `7pt` eller `14pt`, beror på att de typer av texter som vanligtvis typsätts med hjälp av LaTeX brukar sättas med 10, 11 eller 12 punkters textstorlek. Det *finns* naturligtvis fortfarande möjlighet att få större och mindre text, och hur man gör det kommer vi till alldeles strax.

Andra nya saker som har använts i exemplet är

- Tecknet `%` talar om att resten av raden helt ska ignoreras. Man kan alltså skriva korta kommentarer i sitt dokument utan att det hamnar på det utskrivna papperet. Detta kan vara bra och användbart om man vill kommentera bort ett stycke i en text i stället för att ta bort stycket helt och hållet.
- `\usepackage[latin1]{inputenc}` laddar in ett paket som heter `inputenc` och gör så att man kan skriva sin text med vanliga svenska tecken (d.v.s. genom att använda å, ä och ö samt é etc.). Skulle inte denna rad finnas i dokumentets huvud skulle alla svenska specialtecken försvinna.

  Om tangentbordet som du skriver på inte innehåller de svenska specialtecknen, eller om det saknas andra tecken som du behöver, kan du ta en titt i avsnitt 3.4 på sidan 18 för att få reda på hur du ska göra.

  Texten `latin1` är ett argument till `inputenc`-paketet som talar om att vi använder oss av standarden ISO Latin-1 (standard på de flesta UNIX-datorer) när vi skriver vår text. Om vi satt vid en Macintosh skulle vi vara tvungna att skriva `applemac` i stället för `latin1` för att kunna använda specialtecknen. Se tabell 3.1 på motstående sida för andra argument till `inputenc`-paketet.

  På liknande sätt kan man m.h.a. kommandot `\usepackage[...]{...}` få tillgång till andra paket med massor av nya kommandon som inte finns i LaTeX i vanliga fall. Dessa paket är skrivna av andra LaTeX-användare och vi kommer att se på några av de mest använda paketen senare i denna handledning.
- `{\Large Puh}` innebär att texten `Puh` kommer att skrivas med stor stil.

  Kommandon för storleksreglering av text finns i tabell 3.2 på nästa sida.

> **OBS:** Kommandona för storleksreglering av text är ”aktiva” inom det aktuella `{...}`-blocket. Detta medför att vi inte kan skriva `\Large Puh` eftersom detta skulle medföra att *all* text efteråt blev stor. Lösningen är att i stället skriva `{\Large Puh}` som vi gjorde i exemplet på föregående sida, eller alternativt, `\Large Puh \normalsize`.
> Ett vanligt fel som man brukar göra är att skriva `\Large{...}` eller liknande, men detta är faktiskt samma sak som att skriva `\Large ...`, och man måste alltså ”stänga av” den stora stilen explicit med `\normalsize`.

> **OBS:** Alla kommandon eller kommandosekvenser i LaTeX är *case sensetive*, d.v.s. det spelar roll om man använder stora eller små bokstäver. Således är `\Large` ett annat kommando än vad `\LARGE` är.

- `\emph{helt}` betyder att ordet `helt` ska vara betonat (eng. *emphasized*). Ordet kommer att skrivas med kursiv stil. Om man skrev `\emph{Nu lyser solen}, solen är gul` så kommer följaktligen texten ”Nu lyser solen” att bli betonad, medan texten ”solen är gul” kommer att vara normal.

Kortnamn	Beskrivning
`ansinew`	Windows 3.1 ANSI
`applemac`	Macintosh
`ascii`	ASCII-koder i intervallet 32–127
`cp437de`	IBM 437 (tyska versionen)
`cp437`	IBM 437
`cp850`	IBM 850
`cp852`	IBM 852
`cp865`	IBM 865
`latin1`	ISO Latin-1 ("Western")
`latin2`	ISO Latin-2 ("Central European")
`next`	Next

Tabell 3.1: De olika kodscheman som finns tillgängligt för `inputenc`-paketet.

Kommando	Resultat
`{\tiny ABC abc 123}`	ABC abc 123
`{\scriptsize ABC abc 123}`	ABC abc 123
`{\footnotesize ABC abc 123}`	ABC abc 123
`{\small ABC abc 123}`	ABC abc 123
`{\normalsize ABC abc 123}`	ABC abc 123
`{\large ABC abc 123}`	ABC abc 123
`{\Large ABC abc 123}`	ABC abc 123
`{\LARGE ABC abc 123}`	ABC abc 123
`{\huge ABC abc 123}`	ABC abc 123
`{\Huge ABC abc 123}`	ABC abc 123

Tabell 3.2: Kommandon för typsättning av olika textstorlekar.

Kommando	Resultat		Alternativ
`\emph{ABC abc 123}`	*ABC abc 123*	betonad stil, eng. *emphasized*	`{\em ...}`
`\textit{ABC abc 123}`	*ABC abc 123*	kursiverad stil, eng. *italics*	`{\it ...}`
`\textbf{ABC abc 123}`	**ABC abc 123**	fet stil, eng. *boldface*	`{\bf ...}`
`\texttt{ABC abc 123}`	`ABC abc 123`	"skrivmaskinsstil", eng. *teletype*	`{\tt ...}`
`\textsl{ABC abc 123}`	*ABC abc 123*	lutande stil, eng. *slanted*	`{\sl ...}`
`\textsc{ABC abc 123}`	ABC ABC 123	kapitäler, eng. *small caps*	`{\sc ...}`
`\textsf{ABC abc 123}`	ABC abc 123	sans serif	`{\sf ...}`

Tabell 3.3: Kommandon för typsättning av olika textstilar.

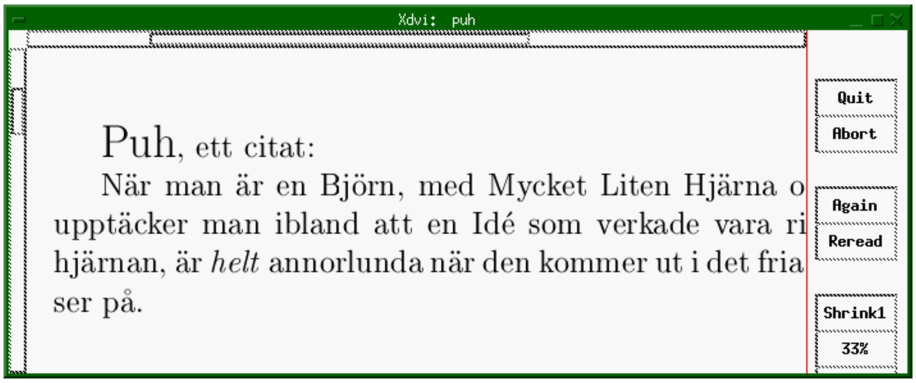
Kommandon som liknar `\emph{...}` för typsättning av olika textstilar finns i tabell 3.3 på föregående sida.

Tips: Kommandot `\textit{...}` (kursivering, eng. *italics*) gör *nästan* samma sak som kommandot `\emph{...}`. Man bör dock använda `\textit{...}` då man vill skriva kursiv text, och `\emph{...}` då man vill skriva betonad text. Detta eftersom `\emph{...}` och `\textit{...}` fungerar olika då man gör en redan betonad text betonad eller gör en redan kursiv text kursiv.
Som ett exempel på detta kan man titta på de två kommandosekvenserna `\emph{ja ja \emph{ja} ja}` och `\textit{ja ja \textit{ja} ja}` som i det färdiga dokumentet typsätts som ”*ja ja* ja *ja*” respektive ”*ja ja ja ja*”, en viss skillnad alltså.

OBS: Liksom kommandona för storleksreglering av text är de alternativa kommandosekvenserna för textstilar (längst till höger i tabell 3.3 på föregående sida) ”aktiva” inom det aktuella `{...}`-blocket.
Se varningen om detta på sidan 14 för mer info.

Tips: Även om man har själv har möjlighet att reglera storleken och stilen på sin text, avråder man ofta från detta. En text skriven med många olika storlekar och stilar kan vara förvillande och kan dessutom lätt anses som oseriös.

Om du tittar på figur 3.1, eller ännu hellre testar själv, så kan du se hur exemplet på sidan 13 ser ut då man visar det med *XDvi*.



Figur 3.1: *XDvi* visar hur LaTeX behandlar radslut m.m. i vår textfil.

Vi noterar följande:

1. Stycken i texten skiljs åt genom att man lämnar minst *en* tom rad mellan dem. Det spelar ingen roll om det finns flera tomma rader mellan två stycken.

2. Ett radslut i textfilen som innehåller LaTeX-dokumentet innebär inget speciellt. LaTeX kommer att göra ett vanligt mellanrum mellan det sista ordet på den ena raden och det första ordet på den andra raden, men ingen ny rad.

3. LaTeX bryr sig inte heller om att man lagt in flera mellanslag mellan två ord. Texten `en␣␣␣smörgås` (*tre* mellanslag mellan orden ”en” och ”smörgås”) kommer att typsättas exakt likadant som `en␣smörgås` (med *ett* mellanslag mellan orden).
4. Radbrytningar genereras automatiskt. Även sidbrytningar genereras automatiskt då textmassan blir större.

3.2 Avancerad vanlig text, rubriker

I rapporter vill man ofta ha olika typer av rubriker för att markera nya avdelningar i sin text. Om vi väljer att skriva ett LaTeX-dokument med `article` som stildokument, får vi tre nivåer av rubriker. Dessa är `section`, `subsection` och `subsubsection`.

I vårt LaTeX-dokument (i dokumentets kropp, d.v.s. *efter* `\begin{document}`) kan vi nu skriva

```
\section{Teori}
Vi härleder nu vår fantastiska formel i två simpla steg.

\subsection{Steg ett}
Gör si och så.

\subsection{Steg två}
Rulla runt och sitt fint.
```

Resultatet kommer att bli tre rubriker, numrerade som **1 Teori**, **1.1 Steg ett** och **1.2 Steg två**, på ett liknande sätt som rubrikerna i denna handledning är numrerade.

> **Tips:** Det anses vara god stil att lägga `subsection`:s *under* (efter) `section`:s, och `subsubsection`:s *under* `subsection`:s för att på så sätt skapa en logisk hierarki (rangordning) av rubriknivåer. Dessutom ska man aldrig lägga en `subsubsection` under en `section` utan att däremellan ha lagt en `subsection`.

Man kan också stänga av numreringen av rubriker genom att sätta en `*` direkt efter rubrikkommandot, före rubriknamnet. Nästa exempel illustrerar detta.

```
\section{Teori}
Vi härleder nu vår fantastiska formel i två simpla steg.

\subsection*{Steg ett}
Gör si och så.

\subsection{Steg två}
Rulla runt och sitt fint.
```

Nu kommer rubriken ”Steg ett” att vara onumrerad medan rubriken ”Steg två” kommer att få nummer 1.1 i stället för 1.2 som den hade förut.

> **OBS:** Onumrerade rubriker kommer *inte* att synas i en ev. innehållsförteckning. Se avsnitt 5.1 på sidan 33 för mer information om hur du automatiskt kan skapa en innehållsförteckning med hjälp av LaTeX.

3.3 Fotnoter

LaTeX erbjuder ett enkelt sätt att skriva fotnoter, d.v.s. kortare beskrivningar eller hänvisningar som man vill slippa ha i den löpande texten. En fotnot läggs i sidans "fot" (d.v.s. längst ner på sidan) och får ett nummer associerat till sig. Numret skrivs sedan i den löpande texten så att man vet till vilket ord eller avsnitt i texten som fotnoten hör.

I LaTeX skrivs fotnoter med kommandot `\footnote{...}` på följande enkla sätt:

```
Detta är ett exempel på ett kort\footnote{Relativt sett.} stycke som
har två fotnoter i sig.  Fotnoterna får var sitt nummer och placeras
längst ner på sidan\footnote{Detta är längst ner på sidan.}.  Numret
placeras i texten och vid själva noten.
```

Resultatet blir:

> Detta är ett exempel på ett kort[1] stycke som har två fotnoter i sig. Fotnoterna får var sitt nummer och placeras längst ner på sidan[2]. Numret placeras i texten och vid själva noten.

Notera de två fotnoterna längst ner på denna sida.

> **Tips:** Det är lätt att glömma den avslutande punkten i texten som utgör fotnoten. Gör inte det.

3.4 Accenter och andra textsymboler

Exempel på tecken med accenter är á, ç och ñ. Dessa tecken finns normalt inte på ett tangentbord, men man kan, med speciella LaTeX-kommandon, lätt skriva texter med dem i alla fall.

I tabell 3.4 ges exempel på hur man i LaTeX kan skriva de vanligast förekommande accenterna. Observera att det går bra att sätta accenter på andra bokstäver än de som finns presenterade i tabellen.

Kommando	Resultat	Alternativ
`\"{o}`	ö	`{\"o}`
`\'{e}`	é	`{\'e}`
`\.{o}`	ȯ	`{\.o}`
`\={o}`	ō	`{\=o}`
`\H{o}`	ő	`{\H{o}}`
`\^{e}`	ê	`{\^e}`
`` \`{e} ``	è	`` {\`e} ``
`\b{o}`	o̱	`{\b{o}}`
`\c{c}`	ç	`{\c{c}}`
`\d{o}`	ọ	`{\d{o}}`
`\t{oo}`	o͡o	`{\t{oo}}`
`\u{o}`	ŏ	`{\u{o}}`
`\v{c}`	č	`{\v{c}}`
`\~{n}`	ñ	`{\~n}`

Tabell 3.4: Kommandon för att typsätta olika typer av accenter i LaTeX.

[1] Relativt sett.
[2] Detta är längst ner på sidan.

Andra symboler som är vanligt förekommande i vissa texter finns presenterade i tabell 3.5.

Kommando	Resultat
`!‘`	¡
`?‘`	¿
`\#`	#
`\$`	$
`\%`	%
`\&`	&
`\P`	¶
`\S`	§
`\_`	_
`\aa \AA`	å Å
`\ae \AE`	æ Æ
`\copyright`	©
`\dag`	†
`\ddag`	‡
`\i`	ı
`\j`	ȷ
`\l \L`	ł Ł
`\ldots`	...
`\o \O`	ø Ø
`\oe \OE`	œ Œ
`\pounds`	£
`\textasciitilde`	˜
`\{`	{
`\}`	}

Tabell 3.5: Kommandon för typsättning av i vissa texter vanligt förekommande textsymboler.

Tecknen ı och ȷ (i och j utan prickar) används med fördel för att typsätta t.ex. tecknen î och ǰ (`\^{\i}` resp. `\v{\j}`).

> **OBS:** Notera att de sju tecknen #, $, %, &, _, { och } har speciella innebörder för LaTeX, och att man därför behöver speciella kommandon för att typsätta dem. Om de speciella kommandona (se tabell 3.5) *inte* används kommer antagligen fel att uppstå under TeX:ningen.

3.5 Streck och ”fnuttar”

I LaTeX gör man skillnad mellan olika sorters streck. Man skiljer mellan avstavnings-, binde- och tankstreck. Dessutom finns två typer av ”fnuttar” (apostrofer). Här följer en förklaring.

3.5.1 Avstavnings-, binde- och tankstreck

Förutom accenter kan det vara svårt att hitta tecknen ”-” (avstavningsstreck, eng. *hyphen*). ”–” (bindestreck, eng. *en-dash*) och ”—” (tankstreck, eng. *em-dash*) på ett tangentbord (nja, avstavningsstrecket brukar finnas som ett minustecken).

Många har inte en aning om att det faktiskt är skillnad mellan de olika typerna av streck, så här kommer en kort förklaring av vad de betyder och var man ska använda dem:

- Avstavningsstrecket ("-") används för sammansatta ord som t.ex. "Fortran-kompilator" och det skrivs som *ett* minustecken i dokumentfilen.

  Avstavningsstrecket kommer att hjälpa LaTeX med att avstava ordet, men om ingen avstavning behövs (om ordet ligger t.ex. mitt på en rad) kommer strecket ändå att synas. Se avsnitt 6.2 på sidan 38 om du veta mer om hur du hjälper LaTeX med avstavningen *utan* att avstavningsstrecket syns.

  > **Tips:** Då man skriver en text på svenska skall de flesta sammansatta ord *inte* skrivas isär. Orden "examensarbete", "programexempel", "hösttermin", "användarmanual", "hemadress" etc. ska skrivas just så, utan isärskrivning. Det engelska språket är annorlunda.

- Bindestrecket ("–") skrivs som *två* minustecken och använd bl.a. för intervallangivelser som t.ex. i texten "på sidorna 17–25" (som typsätts som `på sidorna 17--25`).

  Bindestrecket används också då man skriver t.ex. "Newton–Raphson", "Navier–Stokes" och "Runge–Kutta" eftersom det i samtliga fall rör sig om *två* personer och inte om enskilda personer med dubbelnamn.

  > **Tips:** Tag för vana att alltid använda bindestreck för intervallangivelser samt för sammansatta namn.

- Tankstrecket ("—") används mycket sällan i svenska texter, men skrivs som *tre* på varandra följande minustecken.

3.5.2 Citationstecken och apostrofer

Tecknet " finns egentligen inte i LaTeX:s teckenuppsättning utan skrivs i stället som två apostrofer (d.v.s. som `''`). Apostrofer finns det däremot två typer av: "högersvängda" och "vänstersvängda". Därför ska man vara noga med vilken sorts apostrof man skriver.

Engelska citationstecken[3] brukar se ut som “...”. I LaTeX skrivs detta ` ``...'' `. Svenska citationstecken ska skrivas "...". I LaTeX skrivs detta `''...''`.

En text där en apostrof föregår ett citationstecken skrivs `'\,''`. Kommandot `\,` infogar ett kort avstånd i texten. Resultatet av detta blir: ’ ”. Jämför med `'''` som resulterar i ”’.

> **Tips:** Då du skriver en text på svenska i LaTeX, ska du använda de s.k. "vänstersvängda" citationstecknen (`''`) på *båda* sidor om ordet eller orden. En seriös skribent håller alltid reda på vilket språk hon eller han skriver på.

[3] En liten allmänbildande not bara: Det heter "citationstecken" eftersom det är citat det handlar om, inte "situationstecken" vilket man hör alltför ofta.

3.6 Indentering och måttenheter

Indentering betyder att skjuta in en textrad något åt höger. Om man inte säger åt LaTeX att indentera på något speciellt sätt, kommer den första raden i varje nytt stycke att indenteras något jämfört med resten av styckets text. Detta gäller alla nya stycken utom de som ligger direkt efter en rubrik. Dessutom blir det normalt inget vertikalt mellanrum mellan de olika styckena i en text. Detta kan man naturligtvis ändra på.

Det finns två variabler som talar om hur mycket den första raden på varje nytt stycke ska indenteras och hur långt mellanrummet mellan stycken ska vara. Dessa variabler heter `\parindent` resp. `\parskip` och kan ändras med `\setlength`-kommandot.

Vill man t.ex. ha 2.5 millimeters indentering av den första raden i samtliga stycken, skriver man

```
\setlength{\parindent}{2.5mm}
```

i början av dokumentets kropp.

På samma sätt kan man välja att ha 1 centimeters mellanrum mellan stycken genom att skriva

```
\setlength{\parskip}{1cm}
```

i början av dokumentets kropp.

Enheter, förutom `mm` (millimeter) och `cm` (centimeter), som kan användas för att välja längder finns i tabell 3.6.

Ett exempel på ett val av `\parindent` och `\parskip` som brukar fungera bra i de flesta typer av dokument är

```
\setlength{\parindent}{0mm}
\setlength{\parskip}{0.5ex}
```

Tips: Det är god stil att *antingen* ha indentering av den första raden i stycken *eller* att ha avstånd mellan styckena i en text.

Enhet	Namn	Längd	Ungefärligt exempel
Scaled point	sp	1 sp = $\frac{1}{65536}$ pt ≈ 5.3 × 10^{-6} mm (den minsta enheten i LaTeX)	ı
Point	pt	1 pt = $\frac{1}{72.27}$ in = 0.351 mm	ı
Big point	bp	1 bp = $\frac{1}{72}$ in (Postscript point)	ı
Didôt point	dd	72:a-del av en fransk inch = 0.376 mm	ı
Millimeter	mm	1 mm = 2.845 pt	␣
Pica	pc	1 pc = 12 pt = 4.218 mm	␣
Cicero	cc	1 cc = 12 dd = 4.531 mm	␣
Centimeter	cm	1 cm = 10 mm = 2.371 pc	␣
Inch	in	1 in = 25.4 mm = 72.27 pt	␣
	ex	Höjden av lilla x i det aktuella typsnittet	␣ (vid 10pt)
	em	Bredden av stora M i det aktuella typsnittet	␣ (vid 10pt)

Tabell 3.6: De olika måttenheterna i LaTeX.

3.6.1 Automatisk justering av indentering och styckeavstånd

Genom att lägga till raden

```
\usepackage{parskip}
```

i dokumentfilens huvud, aktiveras `parskip`-paketet. Detta paket sätter själv värden på `\parindent` och `\parskip` på ett sådant sätt att sidbrytningar sker på ”bra” ställen.

Paketet `parskip` sätter `\parindent` till noll och justerar sedan `\parskip` för varje sida.

> **OBS:** Kom ihåg att om du använder `parskip`-paketet skall du *inte* ange, m.h.a. `\setlength{...}{...}`, värdena på `\parindent` och `\parskip`.

3.7 Bredare sidor

LaTeX sätter texten med ganska korta rader och många tycker att de marginaler som används är alldeles för breda. Genom att aktivera ett av paketen `a4`, `a4dutch` eller `a4wide` kan man komma till rätta med detta.

I dokumenthuvudet kan man lägga raden

```
\usepackage{a4dutch}
```

för att få tillgång till lite längre textrader.

Paketen `a4` och `a4wide` aktiveras på samma sätt men ger lite annorlunda storlek på textytan än vad `a4dutch` ger. Pröva dig fram till vilket paket som verkar ge den bästa layouten på dina sidor.

Paketet `geometry` kan också användas för att sätta en *godtycklig* storlek på sidor. Se informationen som levereras med paketet eftersom vi inte kommer att ta upp det här.

Kapitel 4

Matematiska formler

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$$

Med LaTeX kan man typsätta mycket avancerade matematiska formler som med annan programvara skulle vara krånglig att ”få till”. Ett exempel på en matematisk formel är den komplexa Fourierserien, som kan skrivas som

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{(in\pi/L)x}, \quad c_n = \frac{1}{2L}\int_{-L}^{L} f(x)e^{-i(n\pi/L)x} dx \tag{4.1}$$

LaTeX skiljer på två olika typer av formler. Dessa är ”displayed” och ”inline”.

> **Tips:** Samtliga matematiska symboler som använts i detta avsnitt finns i bilaga A på sidan 71. Finns inte symbolerna där så kanske du hittar dem i bilaga B på sidan 75.

4.1 ”Displayed”-formler

En ”displayed”-formel är en formel, ekvation eller annan matematisk text som står ensam och centrerad på en rad, så som formel (4.1) här ovanför.

Formel (4.1) skrevs i LaTeX som

```
\begin{equation}
  f(x) = \sum_{n = -\infty}^{+\infty} c_{n}e^{(in\pi/L)x},
  \quad
  c_{n} = \frac{1}{2L} \int_{-L}^{L} f(x)e^{-i(n\pi/L)x} dx
\end{equation}
```

Detta kan verka lite krångligt vid första anblicken, men koden är logisk och lätt att skriva.

I formelkoden har man skrivit följande:

- `\begin{equation}` talar om att det kommer en "displayed"-formel. Man får dessutom, liksom för `section`, `subsection` etc., en konsekutiv numrering vid sidan av formler som startas med `\begin{equation}`.

  Vill man inte ha någon numrering av formeln då det kanske ser "kladdigt" ut om *alla* formler är numrerade, kan man byta ut kommandot `\begin{equation}` mot `\[`, förutsatt att man också byter ut `\end{equation}` mot `\]`.

  Efter `\begin{equation}`, eller `\[`, befinner sig LaTeX i vad man kan kalla en "matematikomgivning". Med detta menas att speciella kommandon för att skapa matematiska formler finns tillgängliga. Vi ska snart gå igenom skillnader mellan "matematikomgivningen" och den vanliga "textomgivningen".

> **OBS:** Gör en mental indelning av kommandona i LaTeX i två "högar", en hög för de som fungerar i "textomgivning", och en annan för de som fungerar i "matematikomgivning".
> I den första högen skall du lägga samtliga kommandon från föregående kapitel medan den andra högen skall innehålla t.ex. `\pi`, `\int`, `\frac{...}{...}`, m.fl. (som vi snart ska gå igenom) som *endast* fungerar i "matematikomgivning". Försök att hålla dessa två högar av kommandon isär.

- `\sum` skapar ett summationstecken, d.v.s. ett $\sum$.

- `_{n = -\infty}` betyder "index $n = -\infty$". När detta används på ett summationstecken (`\sum`) betyder det att summans *undre* gräns, det som ska stå under summationstecknet, ska vara $n = -\infty$. Kommandot `\infty` skapar symbolen ∞.

- `^{+\infty}` betyder "upphöjt till $+\infty$". När detta används på ett summationstecken (`\sum`) betyder det att summans *övre* gräns, det som ska stå ovanför summationstecknet, ska vara $+\infty$.

  Med vad som har sagts här om `_` och `^` kan man enkelt inse att t.ex. c_t^{k+1} skrivs som `c_{t}^{k + 1}`, eller som `c^{k + 1}_{t}`, eftersom ordningen inte spelar någon roll. Ska man skriva $2^{x^{y_j}}$ skriver man `2^{x^{y_{j}}}`. Det gäller alltså att hålla reda på var man sätter { och }.

> **OBS:** Det är tillåtet att skriva `y^2` i stället för `y^{2}` men man bör nog inte göra det eftersom detta kan bli till en ovana som kan ställa till problem. Vad typsätts t.ex. `y^2_i` som? Blir det y^{2_i} (`y^{2_{i}}`) eller y_i^2 (`y^{2}_{i}`)? Rätt svar: y_i^2

- På samma sätt som `\infty` skapar symbolen ∞, skapar `\pi` symbolen π.

- Kommandot `\quad` kan användas i formler *och* i text och skapar ett kort horisontalt avstånd på raden. Vill man skapa ett längre avstånd kan kommandot `\qquad` användas.

- `\frac{1}{2L}` skapar kvoten $\frac{1}{2L}$. Med detta kommando kan man t.ex. skriva $\frac{1+y}{x^2}$ som

`\frac{1 + y}{x^{2}}` eller

$$\frac{\frac{1}{2}}{3} = \frac{1}{6}$$

som `\frac{\frac{1}{2}}{3} = \frac{1}{6}`.

- `\int` skapar ett integraltecken på samma sätt som `\sum` skapar ett summationstecken.
- `\end{equation}` avslutar formeln som påbörjades med `\begin{equation}`. Om formeln påbörjades med `\[` så ska den avslutas med `\]`.

Det som här kallas för ”matematikomgivning” skiljer sig alltså från vanlig ”textomgivning” genom att speciella matematikkommandon kan användas som inte går att använda i ”textomgivning”. LaTeX förväntar sig att allt som står i ”matematikomgivning” är matematiska formler (symboler, variabler m.m.) och typsätter dem därför som sådana.

Radslut i en formel behandlas som vanligt, d.v.s. det ignoreras av LaTeX och de två raderna skriv på samma rad när de typsätts. Se avsnitt 4.4 på sidan 30 för information om hur du kan göra för att bryta raderna i en matematisk formel.

Mellanslag i en formel behandlas annorlunda än mellanslag i en vanlig text. Om man i en formel skriver `x␣y` så kommer mellanslaget mellan `x` och `y` att försvinna, och det hela kommer att typsättas som xy. Detta beror på att LaTeX har en bestämd åsikt om var mellanslag i formler ska ligga och hur stora de ska vara.

4.1.1 Text i formler med hjälp av `mbox`-kommandot

Som redan nämnts kan vanlig text inte skrivas i en formel. Skulle vi t.ex. försöka skriva formeln

```
\[
a = 1 om och endast om b \neq 1
\]
```

så skulle vi på vårt papper få

$$a = 1 omochendastomb \neq 1$$

i stället för

$$a = 1 \mbox{ om och endast om } b \neq 1$$

Varför blir det så? Jo, i och med att vi är i den ”matematiska omgivningen” tror LaTeX att bokstäverna `om och endast om` är variabler som ska multipliceras med varandra och typsätter dem alltså utan mellanrum.

Detta kan kringgås genom att använda kommandot `\mbox{...}`.

För att få det vi vill ha skriver vi alltså i stället

```
\[
a = 1 \mbox{ om och endast om } b \neq 1
\]
```

OBS: Notera att mellanslagen före det första och efter det sista ”om” måste innefattas i `\mbox{...}`-kommandot. Annars försvinner de.

Ett annat användningsområde för `\mbox{...}`-kommandot finner du i avsnitt 6.2.1 på sidan 39.

4.1.2 Text i formler med hjälp av `text`-kommandot

Då kommandot `\mbox{...}` används får man alltid ”normal textstorlek” på texten. Detta innebär att formeln

```
\[
k_{\mbox{minimal}}
\]
```

typsätts som

$$k_{\mbox{minimal}}$$

men inte som

$$k_{\text{minimal}}$$

Med hjälp av kommandot `\text{...}` kan man stoppa in text var som helst i formler. Texten får dessutom automatiskt rätt storlek. Kommandot `\text{...}` används precis som kommandot `\mbox{...}` och kan bara förekomma i ”matematisk omgivning”.

Innan man kan använda `\text{...}`-kommandot måste man ladda `amsmath`-paketet i dokumentets huvud. Detta görs med kommandot

```
\usepackage{amsmath}
```

4.2 ”Inline”-formler

Den andra typen av formler kallas för ”inline”-formler. Dessa typsätts inte på en egen rad utan i stället inuti ett stycke text, därav namnet ”inline”. Denna typ av formler är bra för kortare matematiska uttryck.

Ett exempel på en ”inline”-formel är $\frac{\partial}{\partial t} u(x,t) = \frac{\partial}{\partial x} u(x,t)$ som i LaTeX skrevs som

```
Ett exempel på en ''inline''-formel är $\frac{\partial}{\partial t}
u(x, t) = \frac{\partial}{\partial x} u(x, t)$ som i \LaTeX{}
skrevs som
```

Formler som ska vara ”inline” startas alltså med ett `$` och avslutas sedan med ännu ett `$`. Man får naturligtvis ingen numrering av ”inline”-formler och inte heller nu spelar antalet mellanslag i formeln någon roll.

Följande är ett stycke med ”inline”-formler, hämtat ur en (något äldre) lärobok i numerisk analys:

> [Cordic-algoritmen] är alltså en iterativ metod för beräkning av koordinaterna för vektorn v, d.v.s. $\cos(\beta)$ och $\sin(\beta)$, och man kan visa att den konvergerar för lämpliga val av vinklar $\{\gamma_0, \gamma_1, \ldots\}$ (om man väljer t.ex. $\gamma_i = 2^{-i}\pi/4$ påminner algoritmen om intervallhalveringsmetoden).

Stycket typsattes genom att i LaTeX-filen skriva följande:

```
[Cordic-algoritmen] är alltså en iterativ metod för beräkning av
koordinaterna för vektorn $v$, d.v.s. $\cos(\beta)$ och
$\sin(\beta)$, och man kan visa att den konvergerar för lämpliga val
av vinklar $\{ \gamma_{0}, \gamma_{1}, \ldots \}$ (om man väljer
t.ex. $\gamma_{i} = 2^{-i}\pi/4$ påminner algoritmen om
intervallhalveringsmetoden).
```

Här har man använt andra speciella matematiska symboler som t.ex. β (`\beta`) och γ (`\gamma`). Samtliga (eller åtminstone väldigt många) symboler och tillhörande LaTeX-kommandon finner du i bilaga A på sidan 71 och B på sidan 75.

Tips: Då man skriver en text där variabler av typen x, y etc. förekommer, d.v.s. variabler som är ”vanliga bokstäver”, skall även dessa typsättas inom `$...$` eftersom de är variabler. Man skall *inte* skriva dem som kursiverad eller betonad text eftersom variabler *är* matematiska uttryck och inte kursiverade eller betonade tecken. Man skall inte heller utelämna `$...$` eftersom variablerna då ej får rätt utseende och texten lätt blir svårläslig.

4.3 Avancerade formler

4.3.1 Matriser

Inte alltför sällan blir man tvungen att skriva matriser i sina rapporter eller inlämningsuppgifter. I LaTeX kan man se matriser som ett specialfall av tabeller. Se avsnitt 8.1 på sidan 45.

En onumrerad formel med en liten matris, som t.ex.

$$P_{i}v_{i} = c_{i}\left(\begin{array}{cc} 1 & -\sigma_{i}t_{i} \\ \sigma_{i}t_{i} & 1 \end{array}\right)v_{i}$$

skrivs i LaTeX som

```
\[
P_{i}v_{i} = c_{i}\left(
  \begin{array}{cc}
    1               & -\sigma_{i}t_{i} \\
    \sigma_{i}t_{i} & 1                \\
  \end{array}
\right)v_{i}
\]
```

Indenteringen och mellanslagen i koden ovan är bara för till för att göra exemplet mer lättförståeligt, de påvekar inte resultatet på något sätt.

Vad är då detta?

- `\[` och `\]` känner vi igen sedan förut, se avsnitt 4.1 på sidan 23.
- Kommandona `\left` och `\right` kan du läsa mer om i avsnitt 4.3.3 på följande sida.
- `\begin{array}` talar om att en matris påbörjas.
- `{cc}` talar om att vi vill ha två kolumner och att dessa ska vara centrerade (`c`). Man kan med `l` begära vänsterjustering av en eller flera kolumner och med `r` högerjustering.
- Elementen i matrisen avgränsas inom raderna m.h.a. `&` och raderna skiljs åt med `\\`.
- Allt som skrivs i matrisen skrivs i ”matematisk omgivning”.
- `\end{array}` avslutar matrisen.

4.3.2 Matriser à la $\mathcal{AMS}$

Paketet `amsmath`[1] erbjuder ett enklare sätt att skapa matriser. Detta genom de fem omgivningarna `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` och `Vmatrix`.

För att använda $\mathcal{AMS}$-matriser måste du först aktivera `amsmath`-paketet genom att i dokumenthuvudet skriva

```
\usepackage{amsmath}
```

Sedan skrivs t.ex. formeln ovan som

```
\[
P_{i}v_{i} = c_{i}
\begin{pmatrix}
  1               & -\sigma_{i}t_{i} \\
  \sigma_{i}t_{i} & 1                \\
\end{pmatrix}
v_{i}
\]
```

vilket resulterar i

$$P_i v_i = c_i \begin{pmatrix} 1 & -\sigma_i t_i \\ \sigma_i t_i & 1 \end{pmatrix} v_i$$

Notera att vi slipper skriva `\left(` och `\right)` och att vi dessutom kan utelämna `{cc}`.

$\mathcal{AMS}$-matrisernas avgränsare, d.v.s. parenteserna i matrisen ovan, sitter ”tätare inpå” matriselementen.

De matristyper som finns tillgängliga är: `pmatrix` ( ), `bmatrix` [ ], `Bmatrix` { }, `vmatrix` | |, `Vmatrix` || ||.

4.3.3 Stora parenteser o.dyl.

Ofta vill man skriva ett matematisk uttryck som ska inneslutas i någon form av parentes. Ett exempel på ett sådant uttryck är

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$$

Om vi skrev detta uttryck utan att tänka oss för skulle vi kanske skriva någonting i stil med

```
\[
x_{n + 1} = \frac{1}{2}(x_{n} + \frac{a}{x_{n}})
\]
```

Denna LaTeX-kod skulle resultera i formeln

$$x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n})$$

vilket ju inte alls var vad vi vill ha!

Skriver vi i stället

[1] $\mathcal{AMS}$ står för *American Mathematical Society.*

```
\[
x_{n + 1} = \frac{1}{2} \left( x_{n} + \frac{a}{x_{n}} \right)
\]
```

så får vi precis vad vi ville ha.

Kommandona `\left` och `\right` kan bland annat användas på vanliga parenteser men också på andra typer av avgränsare (se tabell 4.1).

Kommando	Resultat
`\left( \frac{A^{x}}{B^{y}} \right)`	$\left(\frac{A^{x}}{B^{y}} \right)$
`\left[ \frac{A^{x}}{B^{y}} \right]`	$\left[\frac{A^{x}}{B^{y}} \right]$
`\left| \frac{A^{x}}{B^{y}} \right|`	$\left\lvert \frac{A^{x}}{B^{y}} \right\rvert$
`\left\\| \frac{A^{x}}{B^{y}} \right\\|`	$\left\lVert \frac{A^{x}}{B^{y}} \right\rVert$
`\left\{ \frac{A^{x}}{B^{y}} \right\}`	$\left\{ \frac{A^{x}}{B^{y}} \right\}$
`\left< \frac{A^{x}}{B^{y}} \right>`	$\left< \frac{A^{x}}{B^{y}} \right>$

Tabell 4.1: Avgränsare som går att göra godtyckligt stora.

OBS: Observera att då man vill ha t.ex. ett { blir man tvungen att skriva `\{` eftersom tecknet är ett specialtecken i LaTeX. Se tabell 3.5 på sidan 19.

Notera att varje `\left` måste ha ett motsvarande `\right`. Om man vill skriva någonting i stil med

$$(f, g) = \left\{ \begin{array}{ll} \int_{a}^{b} w(x)f(x)g(x)dx & \mbox{(kontinuerliga fallet)} \\[1em] \sum_{i=0}^{M} w_{i}f(x_{i})g(x_{i}) & \mbox{(diskreta fallet)} \end{array} \right.$$

där det verkar som om vi utelämnat `\right`, vilket vi inte gjort, kan man t.ex. skriva

```
\[
(f, g) = \left\{
  \begin{array}{ll}
    \int_{a}^{b} w(x)f(x)g(x) dx &
    \mbox{(kontinuerliga fallet)} \\[1em]
    \sum_{i = 0}^{M} w_{i}f(x_{i})g(x_{i}) &
    \mbox{(diskreta fallet)} \\
  \end{array}
\right.
\]
```

Genom att säga `\right.` (notera punkten) kan man få en ”osynlig” avgränsare.

Texten `[1em]` sätter in ett 1 em stort avstånd (se tabell 3.6 på sidan 21) efter den första raden i matrisen. Man kan på detta sätt justera matrisradernas avstånd.

4.4 Radbrytning i ”displayed”-formler med hjälp av `split` och `multline`

Ibland blir formler lite väl långa. Detta är inget problem när det gäller ”inline”-formler eftersom de bryts då textraden bryts, men då det gäller ”displayed”-formler kan det bli lite knivigare. Vad vi då behöver är ett sätt på vilket man kan tvinga in en radbryting.

Paketet `amsmath` tillhandahåller bl.a. kommandona `split` och `multline` som kan användas i formler. Båda kommandona kräver dock att följande rad finns med i LaTeX-dokumentets huvud:

```
\usepackage{amsmath}
```

4.4.1 Kommandot `split`

Kommandot `split` använder man då man vill att en matematisk formel ska brytas vid ett speciellt ställe och att fortsättningen av formeln ska justeras mot föregående rad. Ett exempel på detta är formeln

$$
\begin{split}
(a+b)^{4} &= (a+b)^{2}(a+b)^{2} \\
&= (a^{2}+2ab+b^{2})(a^{2}+2ab+b^{2}) \\
&= a^{4}+4a^{3}b+6a^{2}b^{2}+4ab^{3}+b^{4}
\end{split}
$$

som skrevs som

```
\[
\begin{split}
  (a + b)^{4} &= (a + b)^{2} (a + b)^{2} \\
  &= (a^{2} + 2ab + b^{2}) (a^{2} + 2ab + b^{2}) \\
  &= a^{4} + 4a^{3}b + 6a^{2}b^{2} + 4ab^{3} + b^{4}
\end{split}
\]
```

Vi använder alltså `split` *innanför* `\[ ... \]` (eller `\begin{equation} ... \end{equation}`) och bryter sedan raderna genom att skriva `\\` precis som i tabeller (se avsnitt 8.1 på sidan 45) och matriser (se avsnitt 4.3.1 på sidan 27). Vi använder ett `&` för att tala om för LaTeX var vi vill ha justeringen mellan raderna i formlen.

4.4.2 Kommandot `multline`

Kommandot `multline` skiljer sig ganska mycket från `split`. Man får bl.a. ingen justering av formelns rader, men det passar bättre för riktigt långa uttryck.

Ett exempel på användandet av `multline` är formeln

$$
\begin{multline}
v^{n+1} = v^{n-1} + \\
2k\left(\sum_{\nu=1}^{d}\left(A_{\nu\nu}D_{+x^{(\nu)}}D_{-x^{(\nu)}} + \sum_{\mu=\nu+1}^{d}A_{\nu\mu}D_{0x^{(\nu)}}D_{0x^{(\mu)}} + B_{\nu}D_{0x^{(\nu)}}\right) + C\right)v^{n} - \\
2k\gamma\sum_{\nu=1}^{d}\frac{\varrho(A_{\nu\nu})}{h_{\nu}^{2}}(v^{n+1} - 2v^{n} + v^{n-1}) \tag{4.2}
\end{multline}
$$

Denna formel skrevs i LaTeX som

```
\begin{multline}
  v^{n + 1} = v^{n - 1} + \\
  2k \left( \sum_{\nu = 1}^{d} \left(
      A_{\nu\nu}D_{+x^{(\nu)}}D_{-x^{(\nu)}} + \sum_{\mu = \nu +
        1}^{d} A_{\nu\mu}D_{0x^{(\nu)}}D_{0x^{(\mu)}} +
      B_{\nu}D_{0x^{(\nu)}} \right) + C \right)v^{n} - \\
  2k\gamma \sum_{\nu = 1}^{d}
  \frac{\varrho(A_{\nu\nu})}{h_{\nu}^{2}}(v^{n + 1} - 2v^{n} + v^{n
    - 1})
\end{multline}
```

och vi ser att man alltså *ersätter* `equation` (eller `\[ ... \]`) med `multline` och sedan bryter raderna med `\\` på samma sätt som när vi använder `split`. Däremot finns här ingen möjlighet att justera raderna mot varandra.

Om man i stället vill ha en onumrerad `multline`-formel byter man ut `\begin{multline}` mot `\begin{multline*}`. Detta kräver dock att man sedan avslutar formeln med ett motsvarande `\end{multline*}`.

4.5 Justering av flera formler mot varandra m.h.a. `align`

Ibland vill man kunna skriva flera numrerade formler efter varandra på ett snyggt sätt, t.ex. genom att justera samtliga formler efter ett likhetstecken eller liknande. Detta kan enkelt åstadkommas genom att använda `align`.

Kommandot `align` kräver `amsmath`-paketet, och därför måste följande rad finnas i dokumenthuvudet:

```
\usepackage{amsmath}
```

Kommandot `align` ersätter `equation`, precis som `multline` gör. Inuti ett `align`-block skriver man på samma sätt som i ett `split`-block, se avsnitt 4.4.1 på motstående sida. Således kan vi skapa formlerna

$$b_{n+2} = b_{n+1} = 0 \tag{4.3}$$

$$b_k = a_k + \frac{2k+1}{k+1} x b_{k+1} - \frac{k+1}{k+2} b_{k+1} \qquad k = n, n-1, \ldots, 0 \tag{4.4}$$

genom att skriva

```
\begin{align}
  b_{n + 2} &= b_{n + 1} = 0 \\
  b_{k} &= a_{k} + \frac{2k + 1}{k + 1}xb_{k + 1} - \frac{k + 1}{k +
    2}b_{k + 1} \qquad k = n, n - 1, \ldots, 0
\end{align}
```

Notera att varje formel-rad får ett eget ekvationsnummer.

Man kan om man vill stänga av numreringen av `align`-formler på samma sätt som man stänger av numreringen av `multline`-formler (d.v.s. infoga en `*` efter ordet `align`, se avsnitt 4.4.2 på föregående sida). Beteendet blir då samma som med `split`-kommandot.

4.5.1 Att infoga text mellan `align`-formler

För att få plats med kortare texter, typiskt en rad eller mindre, mellan formlerna i ett `align`-block, så som i

$$F_2(2h) = f'(x) + c_4(2h)^4 + \mathcal{O}(h^6) \tag{4.5}$$

d.v.s.

$$\frac{F_2(2h) - F_2(h)}{15} = c_4h^4 + \mathcal{O}(h^6) \tag{4.6}$$

(där texten "d.v.s." finns inuti `align`-blocket), måste vi använda kommandot `\intertext{...}`. Detta kommando ger oss möjligheten att lägga in en kortare text i ett `align`-block utan att för den skull förlora justeringen mellan formlerna i blocket.

Formlerna ovan skrevs som

```
\begin{align}
  F_{2}(2h) &= f'(x) + c_{4}(2h)^{4} + \mathcal{O}(h^{6}) \\
  \intertext{d.v.s.}
  \frac{F_{2}(2h) - F_{2}(h)}{15} &= c_{4}h^{4}+ \mathcal{O}(h^{6})
\end{align}
```

OBS: Kommandot `\intertext{...}` får *endast* förekomma direkt efter `\\` (radbrytningen).

Kommandot `\intertext{...}` kräver, eftersom det endast kan förekomma inom `align`-block, `amsmath`-paketet.

Kapitel 5

Innehållsförteckningar, framsidor m.m.

5.1 Automatgenererade innehållsförteckningar

Med hjälp av LaTeX kan man skapa automatgenererade innehållsförteckningar i sina dokument. Detta kan man göra genom att helt enkelt lägga till följande korta rad där man vill ha sin innehållsförteckning:

```
\tableofcontents
```

Då man TeX:ar ett dokument, skapas en s.k. AUX-fil (en fil med suffixet `.aux`). Denna fil innehåller information om var de olika `section`:s, `subsection`:s och `subsubsection`:s finns i dokumentet samt vad de heter. Dessutom lagras information om referenser i AUX-filen, se avsnitt 9 på sidan 51 för information om hur man refererar till avsnitt, formler, figurer och tabeller inom en text.

När LaTeX stöter på kommandot `\tableofcontents` under TeX:ningen skapas en innehållsförteckning utgående från informationen i AUX-filen. Hur en innehållsförteckning kan se ut kan du se om du tittar på innehållsförteckningen i början av denna handledning.

Efter slutet av innehållsförteckningen börjar dokumentets text direkt, utan sidbrytning. Vill man ha en sidbrytning efter innehållsförteckningen kan man lägga till kommandot `\clearpage` direkt efter kommandot `\tableofcontents` vilket betyder att resten av sidan kommer att vara tom och att den påföljande texten hamnar på nästa sida.

OBS: I och med att AUX-filen inte finns då man TeX:ar sitt dokument för första gången ,eller då man medvetet har raderat den, blir man tvungen att TeX:a sitt dokument två gånger för att innehållsförteckningen säkert ska bli korrekt.
Om man har flyttat på rubriker så att de hamnat på nya sidor är man också tvungen att TeX:a sitt dokument två gånger för att innehållsförteckningen ska bli korrekt eftersom AUX-filen alltid innehåller information från *den förra* TeX:ningen.

OBS: Notera att onumrerade rubriker, skapade med t.ex. `\section*{...}`, inte kommer att synas i innehållsförteckningen.

5.2 Framsidor

Man kan enkelt skapa snygga framsidor med LaTeX också.

Låt oss säga att ett LaTeX-dokument i princip ser ut så här:

```
\documentclass[a4paper, 10pt]{article}

\title{My report}
\author{Me}

\begin{document}
\maketitle

text text text...
text text text...
text text text... (antagligen i flera sidor)

\end{document}
```

Då man TeX:ar detta dokument kommer LaTeX, då den stöter på raden där man har skrivit kommandot `\maketitle`, att skapa en framsida med den titeln som definierats m.h.a. `\title{...}`. Detta görs i stor stil ovanför namnet på författaren, definierad m.h.a. `\author{...}`, som skrivs med aningen mindre stil. Dessutom får man datumet då dokumentet TeX:ades under författarens namn. Vill man inte ha något datum på framsida blir man tvungen att definiera ett tomt datum. Det gör man genom att ge följande kommando någonstans i dokumentets huvud:

```
\date{}
```

Man kan också använda sig av `\date{...}` för att sätta ett fixt datum på sitt dokument, t.ex. genom att skriva

```
\date{Torsdagen den 2:a juli, 1998}
```

Vill man ha datumet i sans serif-stil kan man skriva

```
\date{\textsf{\today}}
```

Kommandot `\today` ger dagens datum.

Är dokumentet skrivet av två eller flera författare, vilket ofta är fallet om dokumentet är en vetenskaplig artikel, kan man skilja författarna åt m.h.a. kommandot `\and`, som t.ex. då författarna "N. Puh" och "A. Anka" har skrivit något:

```
\author{N. Puh \and A. Anka}
```

Man kan också tvinga in en radbrytning i titeln eller lägga till en e-post-adress för en eller flera författare:

```
\title{A story of hate and passion \\ in a world gone mad}

\author{A. Anka \\ \texttt{\small aanka@soder.sthlm.se} \and N. Puh \\
  \texttt{\small npuh@the.wood.org}}
```

Här bryter vi titeltexten mellan orden ”passion” och ”in” samt anger författarnas e-post-adresser med liten ”skrivmaskinsstil”.

5.3 Sammanfattningar (abstracts)

I längre rapporter eller avhandlingar brukar man ha en sammanfattning (eng. *abstract*) av texten för att en läsare snabbt ska kunna skaffa sig en bild av vad dokumentet handlar om.

I LaTeX finns en speciell typ av ”avsnitt” som kallas `abstract`. Detta avsnitt fungerar nästan som en vanlig `section` etc. och brukar läggas mellan titelsidan och innehållsförteckningen.

För att i LaTeX skriva en sammanfattning skriver man följande:

```
\begin{abstract}
  Målsättningen med denna rapport är att visa på de brister som...
  ...
  ...
\end{abstract}
```

OBS: Notera `\end{abstract}` i slutet av sammanfattningen.

Ett exempel på en sammanfattning kan beskådas i början av denna handledning.

5.4 Onumrerad titelsida med hjälp av `titlepage`

Om man följer exemplet i avsnitt 5.2 på föregående sida kommer framsidan att hamna i början av den första sidan med sidnumret längst ned och med de första styckena av texten på samma sida. Om dessutom en sammanfattning hade funnits efter kommandot `\maketitle` i dokumentet så hade den hamnat på samma sida som titeln, vilket kanske inte varit så lyckat. Man hade nog hellre velat ha titelsidan onumrerad, sammanfattningen på nästa sida och början av texten på sidan därefter. Det kan man rätta till genom att lägga till ett ord (och ett komma-tecken).

Skriv om den första raden i exemplet på motstående sida till

```
\documentclass[a4paper, 10pt, titlepage]{article}
```

Ordet `titlepage` betyder att framsidan och en ev. sammanfattning ska behandlas speciellt, t.ex. kommer ingen numrering av framsidan eller sidan med sammanfattningen att ske. Sidan efter sammanfattningen får sidnummer 1. Ingen annan text än den som definierar titeln, författaren och datumet kommer att synas på titelsidan och dessutom kommer sammanfattningen att stå ensam på sidan efter titelsidan.

5.5 Bilagor

Genom att i dokumentet lägga in raden

```
\appendix
```

kommer de efterföljande `section`:s etc. att bli bilagor (eng. *appendix*).

Kapitel 6

Språk och avstavning

6.1 Avstavning på svenska/franska/engelska o.s.v.

LaTeX är i grunden engelskspråkigt. Det märks bl.a. på att innehållsförteckningen kallas för ”contents” och att alla datum (t.ex. det på framsidan) blir på engelska. Förutom detta får man automatiskt en engelsk avstavning.

För att vrida allt rätt, så att man får det språk som man vill ha, använder man sig av ett paket som kallas `babel`. Paketet `babel` innehåller information om avstavningsregler m.m. för ett flertal olika språk, bl.a. svenska, engelska, tyska och franska. Se nästa avsnitt för en mer komplett lista av språk.

För att använda `babel` lägger man i dokumenthuvudet till raden

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

Detta gör så att man senare i dokumentet kan växla mellan att skriva på svenska till att skriva på engelska (eller tvärt om naturligtvis) och alltså få en korrekt avstavning för de båda språken.

Då raden `\usepackage[swedish, english]{babel}` finns i början av LaTeX-dokumentet kan man sedan ge kommandot

```
\selectlanguage{swedish}
```

vilket får till följd att all text därefter kommer att behandlas som om vore det svenska.

Med `\selectlanguage{english}` kan man sedan växla tillbaka till engelska. På så sätt kan man t.ex. skriva en engelsk text med svenska citat och få rätt avstavning i båda språken.

Om man nu skriver ett dokument och *enbart* använder sig av svenska ska således LaTeX-kommandot `\usepackage[swedish]{babel}` finnas i dokumentets huvud. Det första kommandot efter dokumenthuvudet skall sedan vara `\selectlanguage{swedish}`.

6.1.1 Kortare avsnitt på annat språk

För kortare fraser på t.ex. esperanto (eller vilket annat språk som helt) kan man också använda konstruktionen

```
\foreignlanguage{esperant}{Vi estas la sola esperantisto
  kiun mi renkontas.}
```

för att tillfälligt byta mellan det förvalda språket och, i detta fall, esperanto.

För avsnitt som är längre än en eller två rader kan man använda konstruktionen

```
\begin{otherlanguage}{english}
  \subsection{Course Content}
  Classification of parallel computers.  Different forms of memory
  organization.  Different forms of program control.  Different
  forms of parallelism.
\end{otherlanguage}
```

Tips: Den avstavning som LaTeX använder sig av då *inget* språk har valts och då `babel`-paketet *inte* har laddats är *inte* strikt engelsk. Den är i stället en heuristisk avstavning som *bygger på* det engelska språket. Därför är det lämpligt att, för helt engelspråkiga texter, använda sig av `babel`-paketet med `english` som valt språk.

6.1.2 Vilka språk finns tillgängliga?

De språk som finns tillgängliga i `babel`-paketet finns listade i tabell 6.1, skrivna som de heter internt i `babel`-paketet.

bahasa	english	greek	portuges	spanishb
breton	esperant	irish	romanian	swedish
catalan	estonian	italian	russianb	turkish
croatian	finnish	lsorbian	scottish	usorbian
czech	frenchb	magyar	slovak	welsh
danish	galician	norsk	slovene	
dutch	germanb	polish	spanish	

Tabell 6.1: De språk som `babel`-paketet stödjer.

6.2 Dum a-vst-avn-ing?

Då `12pt` eller `11pt`, och även `10pt` i vissa fall, används för att skriva ett dokument kan det ibland hända att LaTeX får problem med avstavningen. Dessa problem kan t.ex. yttra sig som att en rad blir för lång och därför sticker ut i den högra marginalen. Man kan då hjälpa LaTeX på traven med avstavningen.

Låt oss säga att ordet ”avstavning” blivit felaktigt avstavat eller kanske inte avstavat alls i ett dokument. Då kan man sätta in `\-`, ett s.k. mjukt avstavningsstreck, på de ställen i ordet där man tycker att ordet skulle kunna avstavas.

Ordet ”avstavning” skulle alltså kunna skrivas som `av\-stav\-ning` i dokumentet. Avstavningsstrecken kommer inte att synas, om ordet inte avstavas, i den slutliga DVI- eller Postscript-filen som man sedan skriver ut.

OBS: Då ett mjukt avstavningsstreck lagts i ett ord kommer LaTeX inte att kunna avstava ordet någon annanstans än just på det ställe där avstavningsstrecket lagts.

Ett annat sätt att garantera att ett ord avstavas korrekt i hela dokumentet, utan att behöva stoppa in mjuka avstavningsstreck överallt där ordet förekommer, är att använda sig av kommandot `\hyphenation{...}`. Man ger då kommandot

```
\hyphenation{av-stav-ning}
```

i början av dokumentet för att tala om för LaTeX att ordet "avstavning" ska avstavas som "av-stav-ning".

> **OBS:** Kommandot `\hyphenation{...}` tycks inte fungera på ord som innehåller de svenska specialtecknen.

6.2.1 Att undvika avstavning

Vill man *inte* att ett visst ord ska avstavas kan man helt enkelt stoppa in det i en `\mbox{...}`, t.ex. genom att skriva `\mbox{avstavning}`.

6.2.2 Hårda mellanslag, "no-break space"

När det gäller namn, som t.ex. "T. S. Elliot", brukar man inte vilja att raden bryts mellan initialerna. Då kan man använda ett s.k. hårt mellanslag (eng. *no-break space*) som i LaTeX skrivs som `~` (tilde).

Namnet "T. S. Elliot" ska alltså skrivas som `T.~S.~Elliot` vilket medför att man är garanterad att radbrytningen hamnar någon annanstans än mellan initialerna i namnet.

Hårda mellanslag är också lämpliga i andra avseenden, bl.a. då man hänvisar, m.h.a. referenser, till något. Till exempel skulle man vilja att en referens till detta avsnitt typsattes som "se avsnitt 6.2.2 på sidan 39" utan att någon av siffrorna hamnade först på en ny rad. Referensen skulle i detta fall ha skrivits som

```
se avsnitt~\ref{sec:nbsp} på sidan~\pageref{sec:nbsp}
```

Läs mer om referenser i avsnitt 9 på sidan 51.

Kapitel 7

Olika typer av listor

Olika typer av listor kan man ofta få användning av i de dokument som man skriver. I LaTeX finns tre typer av listor. Dessa är

- ”itemized” eller onumrerad lista,
- ”enumerated” eller numrerad lista, och
- ”descriptive” eller beskrivande lista.

Objekten i en onumrerad lista föregås av en punkt (•) medan objekten i en numrerad lista föregås av siffror, från 1 och uppåt. Den tredje typen av lista är något annorlunda.

För samtliga tre typer av listor gäller

- Listor startas med `\begin{itemize}` för onumrerade listor, med `\begin{enumerate}` för numrerade listor, eller med `\begin{description}` för beskrivande listor. Dessa kommandon måste sedan avslutas med ett motsvarande `\end{...}` i slutet av listan.
- Varje listobjekt föregås av kommandot `\item` förutom i den beskrivande listan var listobjekt föregås av `\item[...]`, se avsnitt 7.1.3 på nästa sida.
- En lista kan innehålla en annan lista.

7.1 Exempel på de olika listtyperna

7.1.1 Onumrerade listor

Onumrerade listor används då listobjekten ej har någon inbördes ordning, som t.ex. i en inköpslista. Ett exempel på en onumrerad lista är:

- In the Beginning there was nothing, which exploded.
- The trouble with having an open mind, of course, is that people will insist on coming along and trying to put things in it.

som skapades genom att i dokumentfilen skriva

```
\begin{itemize}
\item In the Beginning there was nothing, which exploded.
\item The trouble with having an open mind, of course, is that
  people will insist on coming along and trying to put things in it.
\end{itemize}
```

7.1.2 Numrerade listor

En numrerad lista passar bäst för saker som *har* en inbördes ordning, som t.ex. i en prioriteringslista eller i ett recept för kladdkaka. Ett exempel på användandet av en numrerad lista:

For animals, the entire universe has been neatly divided into things to

1. mate with,
2. eat,
3. run away from, and
4. rocks.

Denna lista skrevs i LaTeX som

```
For animals, the entire universe has been neatly
divided into things to
\begin{enumerate}
\item mate with,
\item eat,
\item run away from, and
\item rocks.
\end{enumerate}
```

7.1.3 Beskrivande listor

Beskrivande listor används t.ex. i ordböcker.

AIRPORTS A place where people hurry up and wait.

CONCORDE It goes twice as fast as a bullet and you get smoked salmon.

HOTELS A place where travelling humans are parked at night. Other humans bring them food, including the famous bacon, lettuce and tomato sandwich. There are beds and towels and special things that rain on people to get them clean.

LaTeX-koden för detta:

```
\begin{description}
\item[AIRPORTS] A place where people hurry up and wait.
\item[CONCORDE] It goes twice as fast as a bullet and you get
  smoked salmon.
\item[HOTELS] A place where travelling humans are parked at night.
  Other humans bring them food, including the famous bacon,
  lettuce and tomato sandwich. There are beds and towels and
  special things that rain on people to get them clean.
\end{description}
```

7.2 Speciella onumrerade listor

Man kan få en onumrerad lista att uppföra sig *ungefär* som en beskrivande lista precis som i avsnitt 1.1 på sidan 1 där vi har bytt ut punkterna i den beskrivande listan mot årtal. Detta görs helt enkelt genom att i stället för `\item` skriva t.ex. `\item[1977]` före varje listobjekt.

På detta sätt kan man skapa listan

Woody: Can I pour you a draft, Mr. Peterson?

Norm: A little early, isn't it Woody?

Woody: For a beer?

Norm: No, for stupid questions.

genom att skriva

```
\begin{itemize}
\item[Woody:] Can I pour you a draft, Mr. Peterson?
\item[Norm:]  A little early, isn't it Woody?
\item[Woody:] For a beer?
\item[Norm:]  No, for stupid questions.
\end{itemize}
```

Vi kan jämföra detta med motsvarande beskrivande lista, där vi helt enkelt bytt ut ordet `itemize` mot `description`:

Woody: Can I pour you a draft, Mr. Peterson?

Norm: A little early, isn't it Woody?

Woody: For a beer?

Norm: No, for stupid questions.

Kapitel 8

Tabeller och figurer

Tabeller och figurer i LaTeX påminner lite om varandra (de är båda s.k. ”floats”), varför vi har valt att lägga dessa två ämnen tillsammans i detta kapitel.

8.1 Tabeller

Det enklaste sättet på vilket man skapar en tabell med LaTeX är att bygga en konstruktion som ser ut som

```
\begin{table}[htb]
  \centering
  \begin{tabular}{|l||r|c|}
    \hline
    Kolumn A & Kolumn B & Kolumn C \\
    \hline
    \hline
    A1       & B1       & C1       \\
    A2       & B2       & $mc^{2}$ \\
    \hline
  \end{tabular}
  \caption{En enkel tabell.}
  \label{tab:enkel}
\end{table}
```

Denna LaTeX-snutt skapar tabell 8.1.

Kolumn A	Kolumn B	Kolumn C
A1	B1	C1
A2	B2	mc^2

Tabell 8.1: En enkel tabell.

Vi tittar som vanligt på vad som försigår:

- Blankstegen och indenteringen i tabellkoden är bara för att det ska se snyggt ut.
- Kommandot `\begin{table}` skapar vad man skulle kunna kalla en ”tabular float”. En ”tabular float” är en ruta i dokumentet som innehåller en tabell och som kan ”flyta omkring”.

Genom att skapa en sådan flytande tabell har LaTeX friheten att flytta tabellen från den plats i texten där man skapade den, till en annan plats. Detta kan vara bra om t.ex. tabellens storlek skapar problem med sidbrytningar eller om det av andra layout-tekniska skäl vore bättre om tabellen flyttades. Se avsnitt 8.1.2 på sidan 48 för hur du skapar tabeller som inte flyter.

Kommandot `\begin{table}` skall ha ett motsvarande `\end{table}` i slutet av tabellen.

- Med hjälp av `[htb]` kan vi tala om för LaTeX ungefär var vi vill ha vår tabell. Vi ger här ett antal möjligheter:
  - `h` (eng. *here*), placera tabellen *om möjligt* här.
  - `t` (eng. *top*), placera tabellen *om möjligt* i början av en sida (denna eller en av de kommande sidorna).
  - `b` (eng. *bottom*), placera tabellen *om möjligt* i slutet av en sida (denna eller en av de kommande sidorna).

  Genom att ge samtliga tre alternativ ger vi LaTeX möjligheten att välja själv var tabellen ska ligga.

  Man kan också specificera, med `[p]`, att man vill samla tabeller på speciella ”float-sidor”.
- Med `\centering` centrerar vi tabellen på sidan.
- `\begin{tabular}` talar om att själva tabellinnehållet inleds. Detta måste avslutas med ett motsvarande `\end{tabular}` i slutet av tabellens innehåll.
- `{|l||r|c|}` talar om för LaTeX att vi kommer att vilja ha tre kolumner i vår tabell. Den första kolumnen ska vara vänsterjusterad (`l`), den andra ska vara högerjusterad (`r`) och den tredje ska vara centrerad (`c`).

  De vertikala strecken i `{|l||r|c|}` talar om att vi vill ha vertikala streck på dessa ställen i tabellen.
- Innehållet i tabellcellerna separeras inom raderna med `&` och raderna skiljs åt från varandra med `\\`.
- Kommandot `\hline` kommer att skapa ett horisontalt streck i tabellen. Två på varandra följande `\hline` kommer således att skapa två streck.
- Efter `\end{tabular}` har vi möjligheten att skapa en tabelltext (eng. *caption*). Detta gör man med kommandot `\caption{...}`.
- Med kommandot `\label{...}` *efter* `\caption{...}` kan man skapa en `label` för tabellen. Läs mer om `label`:s i avsnitt 9 på sidan 51.

OBS: För att referensen till en tabell skall bli korrekt måste `\label{...}` alltid ligga *efter* `\caption{...}`.
Läs mer om kommandot `\label{...}` i avsnitt 9 på sidan 51.

8.1.1 Radbrytningar i tabellceller

Vi har i föregående avsnitt sett att det finns tre olika sorters kolumner som man kan använda sig av i en tabell; `r` (högerjusterad), `l` (vänsterjusterad) och `c` (centrerad). Ingen av dessa tre typer av kolumner tillåter radbrytning i en tabellcell, vilket kan vara lite irriterande om man vill skriva längre texter i en cell.

Jag tycker om ...	Jag tycker *inte* om ...
att sova	sport
att lyssna på musik	insekter som flyger in i min mun när jag cyklar
mat	reklam i radio (och är det på morgonen så är det extra jobbigt)
godis	regn

Tabell 8.2: Vad jag gillar, och inte.

Låt oss t.ex. säga att vi vill skapa en tabell med samma utseende som tabell 8.2. Vad vi då kanske skulle ha skrivit är något i stil med

```
\begin{table}[htb]
  \centering
  \begin{tabular}{|l|l|}
    \hline
    Jag tycker om\ldots & Jag tycker \emph{inte} om\ldots \\
    \hline
    \hline
    att sova            & sport                     \\
    \hline
    att lyssna på musik & insekter som flyger in i
                          min mun när jag cyklar    \\
    \hline
    mat                 & reklam i radio (och är det på
                          morgonen så är det extra jobbigt) \\
    \hline
    godis               & regn                      \\
    \hline
  \end{tabular}
  \caption{Vad jag gillar, och inte.}
  \label{tab:gott_och_ont}
\end{table}
```

Vad denna LaTeX-kod skapar kan ni se i tabell 8.3.

Jag tycker om ...	Jag tycker *inte* om ...
att sova	sport
att lyssna på musik	insekter som flyger in i min mun när jag cyklar
mat	reklam i radio (och är det på morgonen så är det extra jobbigt)
godis	regn

Tabell 8.3: Vad jag gillar, och inte.

Vi måste på något sätt tala om att vi vill begränsa kolumnbredden till en viss bredd. Detta kan man göra genom att i stället för `{|l|l|}` specificera `{|l|p{4cm}|}` vilket innebär att den högra kolumnen får en bredd på, i detta fall fyra centimeter. Den vänstra kolumnen har fortfarande ”variabel” bredd.

I standard LaTeX kan man tyvärr inte utan svårigheter få en kolumn med specificerad bredd att bli något annat än vänsterjusterad.

8.1.2 Tabeller som inte flyter

Ibland vill man kunna skriva någonting i stil med " ... som vi nu kan se i följande tabell:" med en tabell direkt efteråt, men i och med att tabeller "flyter omkring" i dokumentet kan vi inte anta att tabellen hamnar just precis där vi sagt att den ska ligga.

Om man vill ha en fast, eller icke-flytande, tabell så kan man få det genom att använda sig av paketet `float`. Paketet laddas i dokumenthuvudet med hjälp av kommandot

```
\usepackage{float}
```

och ger oss sedan möjligheten att specificera `[H]` som önskemål på var vi vill ha vår figur. Med `[H]` menas att man vill ha figuren precis *här*.

> **OBS:** Risken är stor att LaTeX kan få problem med layouten om tabellen inte flyter.

8.2 Figurer

Vi ska här presentera en mall för hur man enklast gör för att infoga en figur i ett LaTeX-dokument. I dokumentets huvud behöver man lägga till raden

```
\usepackage{graphicx}
```

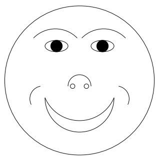
för att få tillgång till kommandot `\includegraphics[...]{...}`.

Själva figuren skall vara i EPS-format[1], vilket är ett format som liknar Postscript och som kan skapas av t.ex. UNIX-programmet *Xfig*. Programmet startas med kommandot `xfig` på de flesta UNIX-system[2].

Om vår EPS-fil heter `bild.eps` och ligger i katalogen `images`, skriver vi i vårt LaTeX-dokument följande kodsnutt:

```
\begin{figure}[htb]
  \centering
  \includegraphics[height=2cm]{images/bild}
  \caption{En kul gubbe.}
  \label{fig:gubbe}
\end{figure}
```

Resultatet blir figur 8.1 (denna fina bild ritades med *Xfig*).



Figur 8.1: En kul gubbe.

Vad vi har gjort här är följande:

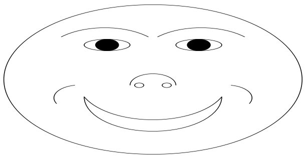
[1] EPS står för *Encapsulated Postscript* och en EPS-fil är en Postscript-fil med lite extra information om bildens eller figurens storlek.

[2] Många ritprogram på andra operativsystem kan exportera bilder till EPS-format. *Xfig* distribueras fritt tillsammans med t.ex. *Linux* (en UNIX-variant som är gratis).

- De två kommandona `\begin{figure}` och `\end{figure}` svarar mot motsvarande tabellkommandon, se avsnitt 8.1 på sidan 45.
- Kommandona `\centering`, `\caption{...}` och `\label{...}` fungerar precis som för tabeller. Detsamma gäller `[htb]`.

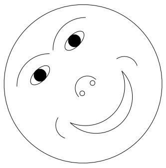
> **OBS:** Observera att `\label{...}`, om en sådan ska användas måste stå *efter* `\caption{...}` för att hänvisningar till figuren skall bli helt korrekta.

- Kommandot `\includegraphics[...]{...}` är det kommando som plockar in figuren i dokumentet. Vi har angett med `[height=2cm]` att figuren ska skalas om så att höjden blir exakt två centimeter. Man kan också ange bredden, m.h.a. `[width=...]`, på figuren. Ger man både höjd och bredd kan figurens perspektiv förvridas. Se t.ex. figur 8.2 där vi angett `[height=2cm, width=4cm]`.



Figur 8.2: Gubben igen.

Med hjälp av `[angle=...]` kan man dessutom vrida på figuren. Se figur 8.3 där vi angett `[angle=45]`.



Figur 8.3: Gubben *igen*. Nu är han sne!

> **Tips:** Det kan vara värt att notera att `[...]` i `\includegraphics[...]{...}`-kommandot kan tas bort helt. Figuren får då den storlek som anges av EPS-filen.

> **Tips:** Det kan ofta vara trevligt att relatera storleken på bilden till t.ex. bredden på sidan. Detta kan man göra genom att specificera bredden som t.ex. `0.8\textwidth` (d.v.s. 80% av textbredden).

8.2.1 Två eller flera figurer brevid varandra

Ibland vill man kunna lägga flera figurer brevid varandra på en sida. Det kan vara praktiskt om figurerna är små. Med vad vi hittills gått igenom kan vi inte åstadkomma detta.

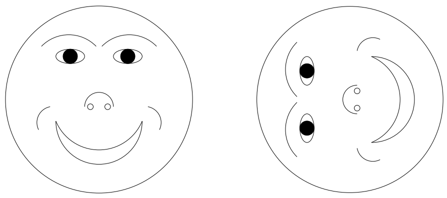
Det finns ett paket, `subfigure`, som tillåter att man kan lägga figurer brevid varandra som "subfigurer". För att kunna använda `subfigure`-paketet måste det laddas i dokumenthuvudet:

```
\usepackage{subfigure}
```

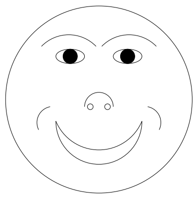
För att sedan skapa "subfigurer" kan man t.ex. skriva

```
\begin{figure}[htb]
  \centering
  \subfigure[Gubbe 1.]{\includegraphics[height=2.5cm]{images/bild}}
  \qquad
  \subfigure[Gubbe 2.]{\includegraphics[angle=90,height=2.5cm]{images/bild}
                       \label{fig:subgubbe}} \\
  \subfigure[Gubbe 3.]{\includegraphics[height=2cm]{images/bild}}
  \caption{Tre gubbar, en snurrig.}
  \label{fig:gubbar}
\end{figure}
```

Resultatet blir figur 8.4. Notera att man kan hänvisa till figur 8.4(b) med hjälp av kommandot `\ref{fig:subgubbe}`. Se kapitlet om referenser på nästa sida.



(a) Gubbe 1. (b) Gubbe 2.



(c) Gubbe 3.

Figur 8.4: Tre gubbar, en snurrig.

Kommandot `\subfigure[...]{...}` innesluts i en `figure`-omgivning (ta en titt på sidan 48) och innehåller en bildtext inom `[...]` och själva figuren (`\includegraphics[...]{...}`-kommandot) inom `{...}`. Vill man dessutom ha en `label` på en eller flera av figurerna sätts `\label{...}`-kommandot som i exemplet ovan.

Man kan skapa en ny rad av subfigurer genom att sätta in `\\` mellan dem. Dessutom kan man skapa mellanrum mellan subfigurerna på samma "rad" genom att infoga `\quad` eller `\qquad` som i exemplet.

8.2.2 Figurer som inte flyter

Man kan, liksom för tabeller, skapa figurer som inte "flyter omkring" i dokumentet. Man gör då på precis samma sätt som med tabeller, se avsnitt 8.1.2 på sidan 48.

Kapitel 9

Referenser

För mer information om referenser, se avsnitt 9 på sidan 51.

Med hjälp av referenser kan man inom ett dokument referera till t.ex. avsnitt, figurer, formler eller tabeller. Man kan t.ex. skriva ” … som vi kommer att se i avsnitt 12.3 på sidan 45 … ”.

Problemet med detta är att vi inte vet på vilken sida som, i detta fall, avsnitt 12.3 finns, eller om avsnittet ens heter 12.3 efter det att vi lagt till avsnitt före och efter det som *förut* var avsnitt 12.3.

9.1 Att namnge objekt m.h.a. `label`

På många ställen i denna handledning har vi sett kommandot `\label{...}` användas, t.ex. i tabell 8.1 på sidan 45 och i figur 8.1 på sidan 48. Med hjälp av `\label{...}` kan man ge ett avsnitt, en bild, en tabell eller en formel ett ”namn” som man sedan kan referera till.

Kommandot `\label{...}` ska stå inuti det som det avser. Således är följande två korrekt namngivna formler:

```
\begin{equation}
  \label{eq:basic1}
  1 + 1 = 2
\end{equation}

\begin{equation}\label{eq:basic2}
  2 + 2 = 4
\end{equation}
```

Den första formeln ovan får ett formelnummer (eftersom det är en ”displayed” formel som startades med kommandot `\begin{equation}`, se avsnitt 4.1 på sidan 23) och man kan sedan referera till den med namnet `eq:basic1`. På motsvarande sätt kan den andra formeln refereras till med namnet `eq:basic2`.

Namnet som man ger ett objekt är godtyckligt, se dock avsnitt 9.3 på sidan 53.

Här följer ännu ett exempel på en `subsection` med namnet `sec:teori`:

```
\subsection{The Euler equation, theory}\label{sec:teori}
text, text, text,
text, text, text.
```

OBS: För figurer och tabeller krävs att namnet definieras efter `\caption{...}`, se avsnitt 8.1 på sidan 45 och 8.2 på sidan 48.

9.2 Att referera till namn med hjälp av `ref` och `pageref`

Då man namngivit något m.h.a. kommandot `\label{...}` kan man sedan referera till det. Det görs i LaTeX med hjälp av de två kommandona `\ref{...}` och `\pageref{...}`.

`ref` Kommandot `\ref{...}` i ett dokument ersätts med formelnumret, figurnumret, tabellnumret eller avsnittsnumret på det man refererar till. Till exempel kan man referera till namnet `sec:namn`, det namn jag gett nästa avsnitt, ”Namnkonvention för referensnamn”, genom att i dokumentet skriva `\ref{sec:namn}`. Resultatet blir texten ”9.3”.

`pageref` Kommandot `\pageref{...}` fungerar som `\ref{...}` men i stället för att ersättas med formelnummer etc. ersätts `\pageref{...}` med sidnumret för den sida där det aktuella namnet finns definierat m.h.a. `\label{...}`. Jag kan t.ex. skriva `\pageref{sec:namn}` för att referera till sidan med avsnittet ”Namnkonvention för referensnamn” (som har namnet `sec:namn`) och få talet ”53” som resultat i det TeX:ade dokumentet.

Tips: När hänvisning till ekvationer görs skall ekvationens nummer stå inom parenteser. Om ekvationen har namnet `eq:basic2` skall man alltså i LaTeX referera m.h.a. kommandosekvensen `(\ref{eq:basic2})`. Se avsnitt 11.2 på sidan 60 för hur du enkelt kan göra för att slippa skriva parenteserna varje gång du refererar till en formel.

9.2.1 Paketet `varioref`

Låt oss säga att vi i LaTeX skriver

```
... ekvation~(\ref{eq:NS}) på sidan~\pageref{eq:NS}
lineariseras genom att...
```

Då får läsaren en fullständig beskrivning av vilken ekvation som ska lineariseras och på vilken sida denna ekvation står att finna. Texten ”på sidan” blir dock överflödig om ekvationen finns på samma sida som hänvisningen till den finns, dessutom skulle det kanske vara trevligare om det stod t.ex. ”på nästa sida” eller ”på föregående sida” då det man hänvisar till finns på nästa resp. föregående sida.

Paketet `varioref` tar hand om detta genom att erbjuda två kommandon:

- `\vref{...}` och
- `\vpageref{...}`.

För att aktivera paketet måste du, i dokumenthuvudet, skriva

```
\usepackage[swedish]{varioref}
```

där `[swedish]` han bytas ut mot t.ex. `[english]` om så önskas.

Byt sedan ut dina `\ref{...}` mot `\vref{...}` och dina `\pageref{...}` mot `\vpagref{...}` och se vad som händer.

Detta paket har använts till så gott som samtliga hänvisningar inom detta kompendium.

9.3 Namnkonvention för referensnamn

Som redan sagts är namnet som man tilldelar ett objekt godtyckligt, men man bör för sin egen skull försöka följa någon form av namnkonvention.

Vanligast är att man använder sig av namnkonventionerna i tabell 9.1. På detta sätt minskar man risken att av misstag använda sig av ett redan definierat namn.

Typ	Exempelnamn	”Försvenskade” exempelnamn
Formel	`eq:exempelformel`	`ekv:exempelformel`
Avsnitt	`sec:exempelavsnitt`	`avs:exempelavsnitt`
Tabell	`tab:exempeltabell`	
Figur	`fig:exempelfigur`	

Tabell 9.1: Namnkonventioner för referensnamn.

Kapitel 10

Att typsätta verbatim text

Ordet ”verbatim” är egentligen engelska och betyder ungefär ”som det står” eller ”ord för ord”. Att typsätta verbatim text innebär att man tvingar LaTeX att ta hänsyn till sådana saker som radslut och mellanslag i dokumentfilen.

Ett exempel på en verbatim text är följande:

```
Denna text typsätts som den står.

Man
   kan
      skriva                   (Till och med här ute)
      som
   man
vill.
```

Hade denna text skrivits i vanlig ”text-omgivning” hade den typsatts såhär:

> Denna text typsätts som den står.
>
> Man kan skriva (Till och med här ute) som man vill.

LaTeX tar ingen hänsyn till kommandon om de är skrivna i en ”verbatim-omgivning”. Detta kan vara bra om man t.ex. vill få LaTeX att typsätta texten

```
\begin{equation}
  1 + 1 = 2
\end{equation}
```

i ett kompendium som detta.

Riktigt nyttig blir verbatim typsättning då man skriver programkod, där indenteringar och radslut spelar stor roll för programmets semantik, syntax och läsbarhet.

Vi ska nu se på olika tekniker för att typsätta verbatim text. Men först ett, eller flera, varningens ord . . .

10.1 Ett varningens ord om verbatim text

Då verbatim text typsätts av LaTeX kommer inga radbrytningar eller avstavningar att göras. Detta kan medföra att LaTeX får problem med den fysiska sidans bredd vilket kan synas på att text sticker ut i marginalen. Därför ska man alltid vara noga med att den text man skriver får plats på sidan eller, då man skriver verbatim text i löpande vanlig text, att LaTeX klarar av att bryta raderna på ett bra sätt.

En annan sak som kan vara värd att nämna är att verbatim text aldrig ska användas till att skriva s.k. ”skrivmaskinstext” (eng. *teletype*). För detta ändamål finns kommandot `\texttt{...}`, se tabell 3.3 på sidan 15, som fungerar helt ok.

10.2 Enkel verbatim text

Här behandlas två av de i LaTeX inbyggda kommandona för att skapa verbatim text.

10.2.1 `verbatim`-omgivningen

I LaTeX finns det en enkel ”verbatim-omgivning”. Den används då en längre text skall typsättas som den står. Med hjälp av kommandona `\begin{verbatim}` och `\end{verbatim}` kan man innesluta texten som skall typsättas som den står. Ett exempel på detta är följande LaTeX-kod:

```
\begin{verbatim}
Denna text typsätts som den står.

      1 2 3      4 5 6
A A
  B B
\end{verbatim}
```

Resultatet av ovanstående kod blir:

```
Denna text typsätts som den står.

      1 2 3      4 5 6
A A
  B B
```

Man kan också få LaTeX att visa alla mellanslag som tecknet ␣ genom att byta ut kommandot `\begin{verbatim}` och `\end{verbatim}` mot `\begin{verbatim*}` resp. `\end{verbatim*}`. Texten ovan får då man gör detta följande utseende:

```
Denna␣text␣typsätts␣som␣den␣står.

␣␣␣␣␣␣1␣2␣3␣␣␣␣␣␣4␣5␣6
A␣A
␣␣B␣B
```

10.2.2 `verb`-kommandot

Skall kortare texter typsättas som de står kan kommandot `\verb` användas. För att typsätta t.ex. texten `\[  a^{2} \cdot  b \]` skriver man

```
\verb!\[  a^{2} \cdot b  \]!
```

Det som ska typsättas som det står innesluts alltså mellan utropstecknen, som också kan bytas ut mot plus-tecken.

Även här har man en möjlighet att visa mellanslag i texten genom att infoga en `*` just innan det första utropstecknet. Gör man detta i exemplet ovan blir resultatet `\[␣␣a^{2}␣\cdot␣b␣␣\]`.

10.3 moreverb-paketet

Det finns ett paket för LaTeX vid namn `moreverb` med vilket man kan åstadkomma diverse nyttiga saker, bl.a. infoga programkod direkt från sina C/C++, Fortran- eller MATLAB-filer.

Vi ska här inte gå in i detalj på samtliga kommandon i `moreverb`-paketet utan i stället ge exempel på de tre mest användbara kommandona och dess möjligheter. Samtliga tre kommandon kräver att `moreverb`-paketet aktiverats i dokumentets huvud genom kommandot

```
\usepackage{moreverb}
```

10.3.1 Skriv programkod m.h.a. listing-omgivningen

Programkod kan skrivas direkt i LaTeX-dokumentet med hjälp av `verbatim`-omgivningen, se avsnitt 10.2.1 på föregående sida. Genom att använda sig av `listing`-omgivningen får man dessutom en numrering av programmets rader. En radnumrering kan vara bra att ha för att kunna hänvisa till en speciell del av ett program eller rutin.

Ett exempel på `listing`-omgivningen är följande programkod, ett Perl-program:

```
1       #!/usr/bin/perl -w
2
3       use strict;
4
5       my $message = "Just Another Perl Hacker!";
6       print((split '', $message)[18,10,16,16,7,4,13..16,-1], "\n");
```

Koden skrevs i LaTeX som

```
\begin{listing}{1}
  #!/usr/bin/perl -w

  use strict;

  my $message = "Just Another Perl Hacker!";
  print((split '', $message)[18,10,16,16,7,4,13..16,-1], "\n");
\end{listing}
```

- Omgivningen `listing` startas med `\begin{listing}{1}` där `{1}` talar om att första raden ska ha radnummer 1.

  Man kan också skriva `\begin{listing}[n]{1}` där värdet på `n` talar om intervallet mellan de numrerade raderna. Anger man `n` = 2 får man numrering av varannan rad, `n` = 3 ger numrering av var tredje rad, o.s.v.
- All text inom omgivningen typsätts som den står.
- Avsluta med `\end{listing}`.

10.3.2 Infoga programkod m.h.a. listinginput-kommandot

Vill man inte skriva koden, eller vad det nu kan vara, direkt i LaTeX-dokumentet kan man använda sig av `listinginput`-kommandot.

Låt oss t.ex. säga att vi har en fil `main.c`, som i vårt fall är ett enkelt C-program, och att vi vill infoga filen i vårt LaTeX-dokument, som kanske är en rapport på en inlämningsuppgift eller laboration i en programmeringskurs. Då kan man skriva följande korta kommando för att stoppa in filens innehåll:

```
\listinginput{1}{main.c}
```

Resultatet av kommandot blir, beroende på hur filens innehåll ser ut just då man TEX:ar sitt dokument:

```
#include <stdio.h>

#ifndef lint
static char vcid[] = "$Id: main.c,v 1.1 2000/06/11 12:48:34 andkaha Exp $";
#endif /* lint */

int main() {
  printf("Hello World!\n");
  return 0;
}
```

Liksom med `listing`-omgivningen har man här möjligheten att få numrering på var `n`:te rad genom att skriva, till exempel, `\listinginput[n]{1}{main.c}`, där `n` byts ut mot önskat värde.

10.3.3 Inramad verbatim text med `boxedverbatim`

Omgivningen `verbatim`, se avsnitt 10.2.1 på sidan 56, kommer att typsätta texten som den står. Om man vill kan man få en ram runt texten genom att byta ut ordet `verbatim` mot `boxedverbatim`.

Omgivningen `boxedverbatim` används precis som `verbatim`-omgivningen men ger alltså dessutom en ram runt texten.

Kapitel 11

Egendefinierade kommandon

Om man skriver en lång text, där ett krångligt uttryck, som t.ex. $\sqrt[n]{\phi(x)}$, uppträder ofta, kan det vara en tröst att veta att man faktiskt kan slippa skriva om texten `$\sqrt[n]{\phi(x)}$` om och om igen i dokumentet.

Genom att i dokumenthuvudet skriva följande kommando, kan man använda det *nya* kommandot `\sqrtphi` i stället för `$\sqrt[n]{\phi(x)}$`:

```
\newcommand{\sqrtphi}{$\sqrt[n]{\phi(x)}$}
```

Med hjälp av `\newcommand{\newcom}{\oldcom}` kan man alltså definera ett sorts makro, en genväg, med kommandonamnet `\newcom`, som kan användas i stället för det längre och krångligare kommandot `\oldcom`.

Nya kommandon behöver inte bestå av matematiska formler utan kan vara i stort sett vad som helst.

11.1 Egna kommandon med argument

Det vi nyss tittade på var egendefinierade kommandon *utan* argument. Vi får t.ex. alltid $\sqrt[n]{\phi(x)}$ då vi använder det egna kommandot `\sqrtphi`. Vi måste göra ett nytt kommando för att få $\sqrt[m]{\phi(y)}$. Detta problem kan man lösa genom att ersätta kommandodefinitionen av det egna kommandot `\sqrtphi` med följande:

```
\newcommand{\sqrtphi}[2]{$\sqrt[#1]{\phi(#2)}$}
```

Nu kan vi skriva `\sqrtphi{m}{y}`, vilket resulterar i $\sqrt[m]{\phi(y)}$. I och med att argumenten till kommandot kan vara i stort sätt vilket matematiskt uttryck som helst, kan vi till och med skriva `\sqrtphi{n}{\omega^{n_{0}}}` för att få resultatet $\sqrt[n]{\phi(\omega^{n_0})}$.

- `[2]` på raden `\newcommand{\sqrtphi}[2]{$\sqrt[#1]{\phi(#2)}$}` betyder att kommandot har två argument.
- De två argumenten placeras på de platser som markeras med `#1` och `#2`, o.s.v. för upp till nio argument.

Då man skriver egna mycket komplicerade kommandon kan det vara till hjälp att ha någon av böckerna [Knu87, Lam94, GMS94] till hands.

> **OBS:** Om man vill *ändra* på ett kommando, d.v.s. först använda en definition av kommandot för att sedan ändra definitionen, kan `\newcommand` ej användas. I stället används kommandot `\renewcommand`.

11.2 Exempel på nyttiga kommandon

Här är exempel på ett par nyttiga kommandon som lätt kan användas i vilket LaTeX-dokument som helst.

Låt oss skapa kommandot `\npsection{...}` ("new-page-section") som kan användas då man vill att en `section` ska börja i början av en ny sida:

```
\newcommand{\npsection}[1]{\clearpage\section{#1}}
```

Detta kommando används precis som `\section{...}` men gör alltså en ny sida först.

Ett kommando, `\insertfig{...}{...}`, som stoppar in en figur i en text:

```
\newcommand{\insertfig}[2]{
  \begin{figure}[htb]
    \centering
    \includegraphics[width=#2]{#1}
  \end{figure}}
```

Använd kommandot genom att t.ex. skriva `\insertfig{bild}{3cm}`. Man får då bilden `bild.eps` med bredd 3 centimeter i sitt dokument. Notera dock att ingen figurtext läggs till och att det inte går att hänvisa till figuren eftersom ingen `label` finns. Mer om figurer finns i avsnitt 8.2 på sidan 48.

När vi ändå pratar om att referera `label`:s, varför inte skapa kommandot `\reffull{...}` som skriver ut texten "se 12.3 på sida 52" då man hänvisar till något som har nummer "12.3" och ligger på sida 52.

```
\newcommand{\reffull}[1]{se~\ref{#1} på sida~\pageref{#1}}
```

Tecknet ~ (tilde) är ett s.k. hårt mellanslag, se avsnitt 6.2.2 på sidan 39 för mer info om sådana.

Ett kommando för att referera till ekvationer kanske? Skriv följande kommando:

```
\newcommand{\refeq}[1]{(\ref{#1})}
```

Detta ger dig kommandot `\refeq{...}` som skriver ut texten "(3)" om du refererar till något med nummer "3" (t.ex. en ekvation).

> **OBS:** Att vi inte kallat ovanstående kommando för `\eqref{...}` beror på att `amsmath`-paketet redan innehåller en sådan funktion. Funktionaliteten hos `\eqref{...}` är exakt densamma som för `\refeq{...}` ovan.

Kapitel 12

Litteraturförteckningar

En litteraturförteckning är en lista av böcker eller andra typer av informationskällor som man har använt sig av för att sammanställa t.ex. en rapport.

En litteraturförteckning är till för att läsare ska kunna verifiera och/eller fördjupa sig i ämnet som den aktuella artikeln/rapporten handlar om. Därför bör litteraturförteckningar vara exakta och så fullständiga som möjligt.

I en vetenskaplig artikel hittar man ofta texter av typen

> By Lemma 3.8 $f = F^*_{n+1}\tilde{f}$ is real and from [5] we get ...

där "[5]" är en hänvisning till en informationskälla i litteraturförteckningen som antagligen ligger i slutet av texten.

I litteraturförteckningen kan denna informationskälla se ut som

> [5] C. van Loan, *Computational Frameworks for the fast Fourier transform*,
> SIAM, Philadelphia 1992.

Problemet med att skapa en litteraturförteckning för hand är att det är svårt att vara konsekvent när det gäller litteraturförteckningens utseende. Detta är svårt eftersom olika institutioner eller förläggare helst vill att man följer en utlagd standard vad gäller t.ex. förkortningar eller de förekommande informationskällornas inbördes ordning.

Hur man skapar hänvisningar till informationskällor i litteraturförteckningen och hur man skapar litteraturförteckningen en gång för alla, utan att bry sig så mycket om vem som ställer kraven på standard, ska vi nu ta en titt på.

12.1 BIB-filen

En av idéerna med litteraturförteckningar i LaTeX är att man ska kunna lägga upp en stor databas, en s.k. bibliografi-fil eller BIB-fil, från engelskans *bibliography*, med massor av olika sorters informationskällor, för att sedan kunna använda den för att göra hänvisningar till texter man hämtat information från. BIB-filen ska kunna användas till *alla* dokument som man skriver. Om en källa inte finns med i BIB-filen ska man lätt kunna lägga till den och på så sätt samtidigt utöka sin databas av litteraturkällor.

När ett LaTeX-dokument sedan hänvisar till t.ex. en bok ur en BIB-fil, eller kanske till och med "ur BIB-fil*en*", läggs boken i LaTeX-dokumentets litteraturförteckning och får en individuell hänvisningssymbol, som t.ex. "[5]" i exemplet ovan, och symbolen läggs in på den plats i dokumentet där hänvisningen gjordes.

För att detta ska fungera krävs först och främst att man skapar BIB-filen.

12.1.1 BIB-filens utseende

I databasen som BIB-filen utgör skriver man alla informationskällor som man kan komma på. BIB-filen skrivs som en vanlig textfil vars namn slutar med suffixet `.bib`.

En informationskälla i en sådan fil kan t.ex. se ut som

```
@techreport{HoBrSt97,
  author =       {Sverker Holmgren and Henrik Brand{\'e}n
                  and Erik Sterner},
  title =        {Convergence acceleration for the Navier--Stokes
                  equations using optimal semicirculant approximations},
  institution =  {Dept. of Scientific Computing, Upp\-sala Univ.,
                  Upp\-sala, Sweden},
  year =         1997,
  type =         {Report},
  number =       192
}
```

Ordet `techreport`, som föregås av ett `@`, betyder att just denna informationkälla är en teknisk rapport. Författaren, titeln etc. kallas för källans "fält" och olika typer av källor har olika fält. Se avsnitt 12.1.2 på nästa sida för en lista på de olika typer av källor som får förekomma i en BIB-fil.

En viktig sak som måste förekomma i alla typer av källor är det som ovan utgörs av texten `HoBrSt97`. Detta är en kod som vi gett till just denna informationskälla och som vi sedan kan använda vid själva hänvisningen till källan. Koden kan i viss mån jämföras med de namn som vi skapar m.h.a. kommandot `\label{...}` för att kunna referera till figurer, formler och avsnitt i en text, se avsnitt 9 på sidan 51. Alla sådana koder måste vara unika men kan i övrigt vara vad som helst. I exemplet ovan utgörs koden av en kombination av författarnas namn och årtalet för publiceringen.

Andra saker som vi noterar:

- En källa påbörjas alltid med `@` direkt följt av källans typ, se nästa avsnitt, och ett {.
- Källans privata kod finns antingen direkt efter { eller på nästa rad, före fältnamnen.
- De olika fälten inom källan avdelas från varandra med ett kommatecken.
- Siffror behöver inte stå inom `{...}`, men andra fält kräver detta. Det är aldrig fel att skriva alla fält inom `{...}`.
- Bokstäver som é, î etc. måste kodas med de koder som finns i avsnitt 3.4 på sidan 18 för att synas. Observera att kontrollsekvenser *måste* stå inom `{...}` (gäller ej `\textbf{...}` etc.).
- Man skriver "and" mellan samtliga författare, editorer, förläggare etc. (BibTeX, som vi kommer till snart, tar för givet att förteckningen är på engelska).
- Om man vill vara säker på att t.ex. texten "Navier–Stokes" i en titel typsätts rätt bör man skriva `{Navier--Stokes}` eller `{N}avier--{S}stokes`, annars finns risken att N:et och S:et blir omgjorda till små bokstäver (gemener).

Indenteringen i exemplet ovan är till för läsbarhetens skull och är alltså inget krav, detsamma gäller radbrytningarna. Ordningen i vilken man skriver de olika källorna i BIB-filen spelar för det mesta ingen som helst roll och därför är det enkelt att lägga till nya källor i t.ex. slutet av filen.

12.1.2 Källtyper i BIB-filen

Följande är en lista av *samtliga* typer av källor med de fält som *krävs*, samt fält som är *valfria*. Notera att "eller" i listan innebär "antingen eller", d.v.s. om en typ av källa kräver `author` eller `editor` så *måste* en av dessa finnas, men ej båda:

`article` En artikel från en journal eller tidskrift.
kräver `author`, `title`, `journal`, `year`
valfritt `volume`, `number`, `pages`, `month`, `note`

`book` En bok med en "explicit" förläggare.
kräver `author` (eller `editor`), `title`, `publisher`, `year`
valfritt `volume` (eller `number`), `series`, `address`, `edition`, `month`, `note`

`booklet` En skrift som är inbunden men som saknar förläggare eller sponsrande institution.
kräver `title`
valfritt `author`, `howpublished`, `address`, `month`, `year`, `note`

`inbook` En del av en bok, d.v.s. en del, ett kapitel eller likande och/eller ett sidintervall.
kräver `author` (eller `editor`), `title`, `chapter` (och/eller `pages`), `publisher`, `year`
valfritt `volume` (eller `number`), `series`, `type`, `address`, `edition`, `month`, `note`

`incollection` En del av en bok som har sin egen titel.
kräver `author`, `title`, `booktitle`, `publisher`, `year`
valfritt `editor`, `volume` (eller `number`), `series`, `type`, `chapter`, `pages`, `address`, `edition`, `month`, `note`

`inproceedings` En artikel i en s.k. "conference proceedings".
kräver `author`, `title`, `booktitle`, `year`
valfritt `editor`, `volume` (eller `number`), `series`, `pages`, `address`, `month`, `organization`, `publisher`, `note`

`manual` Teknisk dokumentation.
kräver `title`
valfritt `author`, `organization`, `address`, `edition`, `month`, `year`, `note`

`mastersthesis` Ett examensarbete för fil.mag. (motsvarar *masters*-titeln i bl.a. USA).
kräver `author`, `title`, `school`, `year`
valfritt `type`, `address`, `month`, `note`

`misc` När inget annat passar används `misc`.
kräver inget
valfritt `author`, `title`, `howpublished`, `month`, `year`, `note`

`phdthesis` En Ph.D.-avhandling, d.v.s. en doktorsavhandling.
kräver `author`, `title`, `school`, `year`
valfritt `type`, `address`, `month`, `note`

`proceedings` En s.k. "conference proceedings".
kräver `title`, `year`
valfritt `editor`, `volume` (eller `number`), `series`, `address`, `publisher`, `organization`, `month`, `note`

`techreport` En rapport publicerad av en skola eller annan institution, ofta med ett nummer i en serie.
kräver `author`, `title`, `institution`, `year`
valfritt `type`, `number`, `address`, `month`, `note`

`unpublished` En skrift som har en titel och författare men som aldrig formellt blivit publicerad.
kräver `author`, `title`, `note`
valfritt `month`, `year`

12.1.3 Fälttyper i BIB-filen

Här kommer en kort beskrivning av de fälttyper som förekom i föregående avsnitt. För fler fälttyper än de som finns här eller för mer utförliga förklaringar, se [GMS94].

`address` Vanligtvis adressen till `publisher` eller till en institution. För större förläggare brukar man bara skriva staden, för mindre förläggare kan man specificera fullständig adress för att hjälpa läsaren.

`author` Namnet eller namnen på författaren eller författarna. En författare med namnet "Donald E. Knuth" skrivs som `{Donald E. Knuth}` men kan också skrivas som `{D[onald] E. Knuth}` såvida inte "Donald E. Knuth" och "D. E. Knuth" är två olika personer.

Namnen `{John Chris Smith}` och `{Smith, John Chris}` är ekvivalenta. `{Thomas von Neumann}` och `{von Neumann, Thomas}` är också ekvivalenta.

För personer med *von* i sitt namn gäller följande:

- `{Förnamn von Efternamn}` *eller*
- `{von Efternamn, Förnamn}` *eller*
- `{von Efternamn, Jr, Förnamn}` (om personen är *junior*).

Se [GMS94] för fler namnregler.

`booktitle` Titeln på boken som innehåller den del som man hänvisar till. Används *inte* i källtypen `book`.

`chapter` Ett kapitel, avsnitt eller liknande.

`edition` Till exempel `{Second}`, utgåvan av textmassan. Ska skrivas med bokstäver och den första bokstaven ska vara en versal. Beroende på vilken stil som används kommer `edition`-fältet att typsättas med stor eller liten bokstav.

`editor` Namnet eller namnen på editorn eller editorerna till källan. Ska vara på samma form som `author`.

`howpulished` Hur något konstigt har blivit publicerat.

`institution` Institutionen som sponsrat en teknisk rapport.

`journal` Journalnamn.

`month` Månaden i vilken källan blev publicerat eller, för opublicerade texter, i vilken månad som det skrevs. Använd de engelska förkortningarna "Jan.", "Feb." etc.

`note` Tilläggsinformation som kan hjälpa läsaren.

`number` Numret på en journal, tidskrift, rapport eller en artikel i en serie. En utgåva av en journal eller tidskrift identifieras ofta med ett nummer. Rapporter har ofta nummer och ibland kan böcker i en serie ha individuella nummer.

`organization` Organisationen som sponsrar en konferens eller som ger ut en manual.

`pages` En eller flera sidor eller ett sidintervall, t.ex. `{32--110}`, `{12, 20, 30--40}` eller `{79+}` där `+` talar om att sidorna inte bildar ett enkelt intervall.

`publisher` Förläggarens namn.

`school` Namnet på skolan/universitetet där avhandlingen skrevs.

`title` Källans titel.

Titeln bör alltid skrivas i det format som den står på den fysiska källan, således ska titeln "Linear and Nonlinear Programming" alltid skrivas just så.

Beroende på vilken BibTeX-stil som används (se 12.2.1 på sidan 66) kan titelns namn typsättas olika vad gäller stora och små bokstäver. För att *tvinga* in versalerna i titelexemplet skriver man dem inom `{...}`.

`type` Typen hos en teknisk rapport, t.ex. `{Research note}`. För `phdthesis` kan man t.ex. sätta `type` till `{{Ph.D.} dissertation}`. För `inbook` och `incollection` kan man ersätta "chapter 1.2" med "section 1.2" genom att sätta `chapter` till `{1.2}` och `type` till `{section}`.

`volume` Volymsnumret hos en journal eller en bok med många individuella delböcker.

`year` Året som ett publicerat arbete blev publicerat eller året då ett icke publicerat arbete skrevs.

De sista två tecknen i `year` ska vara siffror, således kan man sätta `year` till `{about 1972}`.

12.1.4 Kommandot `string` i BIB-filen

Genom att i BIB-filen skriva t.ex.

```
@string{TDB = {Dept. of Scientific Computing, Upp\-sala Univ.,
    Upp\-sala, Sweden}}
```

får vi en variabel med namnet `TDB` som vi sedan kan använda som fältvärde. Vi kan skriva, t.ex. i en `techreport`, att

```
institution = TDB,
```

i stället för

```
institution = {Dept. of Scientific Computing, Upp\-sala Univ.,
  Upp\-sala, Sweden},
```

Strängar kan också konkateneras (sättas ihop) m.h.a. tecknet `#`. Således blir `TDB # {, the world}` en konkatenering av strängvariabeln `TDB` och texten ", the world".

12.2 Skapa litteraturförteckningen och hänvisningar

För att litteraturförteckningen ska hamna i dokumentet skriver man, antagligen i slutet av dokumentet, eller var man nu vill ha sin förteckning, följande:

```
\bibliographystyle{plain}
\bibliography{minbibfil}
```

Här är `minbibfil` samma namn som BIB-filen har, minus suffixet. Har man flera BIB-filer kan man rada upp dem efter varandra inom `{...}` i `\bibliography{...}`-kommandot. Till exempel kan man skriva

```
\bibliographystyle{plain}
\bibliography{minbibfil, petersbibfil, lajbans}
```

om man har tre BIB-filer med namnen `minbibfil.bib`, `petersbibfil.bib` och `lajbans.bib`.

Vad `plain` är kommer vi till i avsnitt 12.2.1 på nästa sida.

Nu kan man börja att lägga in hänvisningar i sin text. Detta görs med kommandot `\cite{...}` och om man t.ex. ska hänvisa till exempelkällan som vi visade förut får man skriva, där man vill ha hänvisningen, följande kommando:

```
\cite{HoBrSt97}
```

där `HoBrSt97` är koden som vi gett till den källa som vi vill hänvisa till.

Då man TeX:ar ett dokument kommer samtliga hänvisade koder att läggas i AUX-filen. Ett program som kallas BibTeX ska sedan köras på LaTeX-filen genom att ge kommandot[1]

```
bibtex mittlatexdokument
```

Här är namnet `mittlatexdokument` det namn som man gett LaTeX-filen *utan* suffixet `.tex`. Detta skapar bl.a. en fil med suffixet `.bbl` som sedan används av LaTeX för att bygga förteckningen på den plats som man specificerat med `\bibliography{...}`. Detta görs genom ännu en TeX:ning.

[1] Åter igen, om kommandot `bibtex` inte finns på ditt system, ta kontakt med en systemansvarig person eller konsultera en manual.

12.2.1 Olika stilar på litteraturförteckningar

I avsnitt 12.2 på föregående sida såg vi kommandot `\bibliographystyle{plain}`. Ordet `plain` är här en "litteraturförteckningsstil".

Det finns ett otal olika stilar på litteraturförteckningar varav `plain` nog är den enklaste[2]. I tabell 12.1 finns en lista på de stilar som oftast medföljer en LaTeX-distribution, se C.1 på sidan 79 för information om var du kan finna flera.

Stil	Beskrivning
`plain`	Standard BibTeX-stil. Litteraturförteckningen sorteras i alfabetisk ordning och hänvisningarna är numeriska av typen "[*n*]".
`unsrt`	Standard BibTeX-stil. Samma som `plain` men förteckningen är sorterad i den ordning som anges av BIB-filen.
`alpha`	Standard BibTeX-stil. Samma som `plain` men hänvisningarna byggs upp automatiskt utgående från författarens/författarnas namn och årtalet.
`abbrv`	Standard BibTeX-stil. Samma som `plain` men förteckningen är mer kompakt tack vara automatgenererade förkortningar.
`acm`	Alternativ BibTeX-stil. Används i journaler från *Association for Computing Machinery*. Förteckningen består av författarnas för- och efternamn i KAPITÄLER och hänvisningarna blir som med `plain`.
`siam`	Alternativ BibTeX-stil. Används i journaler från *Society of Industrial and Applied Mathematics*. Hänvisningarna är som med `plain` och förteckningen utgörs av författarnas efternamn och förkortade förnamn i KAPITÄLER.

Tabell 12.1: De vanligaste litteraturförtecknings-stilarna i LaTeX.

Denna handledning innehåller få hänvisningar till källor, men litteraturförteckningen sist i handledningen finns ändå. Den är typsatt med `alpha`-stilen.

12.2.2 Jag vill ha *hela* BIB-filens innehåll!

Under TeX:ningen plockar LaTeX ut endast de källor i BIB-filen som det hänvisas till i LaTeX-dokumentet. Vill man, av en eller annan anledning, lista *hela* BIB-filens innehåll i litteraturförteckningen kan man lätt göra det genom att före kommandona `\bibliograpystyle{...}` och `\bibliography{...}` lägga till kommandot

```
\nocite{*}
```

Kommandot `\nocite{...}` kan också användas till att plocka in enstaka litteraturkällor i litteraturförteckningen utan att för den skull explicit hänvisa till dem.

[2] En systemansvarig person borde vet vilka stilar som finns inlagda på din dator. Annars kan du söka efter filer med suffixet `.bst`, dessa filer är BibTeX-stilar.

Kapitel 13

Felsökning

Hur länge man än har hållit på med LaTeX blir det *alltid* fel någon gång då och då. Är man en alldeles "grön" LaTeX-användare är det så gott som oundvikligt att man skriver fel.

Vad vi här ska titta på är olika feltexter som LaTeX då och då ger ifrån sig. Men först ...

13.1 Hur lämnar man LaTeX då ett fel har uppstått?

En del fel är allvarliga, andra inte. Då ett allvarligt fel inträffar avbryts vanligtvis TeX:ningen av dokumentet. En feltext skrivs ut och ett frågetecken talar om att LaTeX väntar på att man ska göra något.

Ett exempel:

```
! Undefined control sequence.
l.1793 \sectionHur
                    l^^e4mnar man \LaTeX?}
?
```

Vad felet här är lämnar vi så länge.

Vid frågetecknet kan man skriva något av tecknen i tabell 13.1 på nästa sida.

Tangent	Innebörd
`<q>`	Avsluta, men skumma igenom texten så att ev. `label`:s m.m. hamnar i AUX-filen så att den uppdateras korrekt.
`<x>`	Avsluta nu på en gång!
`<r>`	Samma som `<q>` men skriv ut feltexter om det finns fler fel.
`<h>`	Hjälp! Ge ett tips om vad som kan vara fel här.
`<ret>`	OK, fortsätt till nästa fel.

Tabell 13.1: Kommandon som man kan ge till LaTeX då ett fel inträffat vid TeX:ningen.

13.2 Missing $ inserted

Detta felmeddelande betyder att man har försökt skriva en matematisk symbol i ”text-omgivning”. En komplett feltext från LaTeX kan se ut såhär:

```
! Missing $ inserted.
<inserted text>
                $
l.229 ^^e4r \nabla
                      stor!
```

Här har man försökt använda kommandot `\nabla`, som är en matematiksymbol, som om det vore en text-symbol.

Felet ligger på rad 229 i dokumentfilen. Det ser vi på felmeddelandet där det står `l.229`.

De konstiga tecknen `^^e4r` är ett ”ä” i LaTeX:s interna 7-bitars representation och kan ignoreras.

Lösning

Stoppa in `$` på lämpliga ställen, eller se om du verkligen stavat kommandonamnet rätt.

13.3 Overfull hbox

Om en textrad i dokumentet inte avstavas ordentligt eller om avstavning inte är möjlig kan följande feltext uppenbara sig:

```
Overfull \hbox (111.63849pt too wide) in paragraph at lines 1819--1820
\T1/cmtt/m/n/10 Denna l^^e5nga textrad kommer att g^^e5 ^^f6ver
marginalen utan att brytas vilket inte ^^e4r s^^e5 bra
```

Här får vi dessutom en vink om var felet uppstår. I detta fall är det i stycket på raderna 1819–1820 i dokumentfilen.

Lösning

Titta med *Xdvi* på den felande raden, se avsnitt 2.3.2 på sidan 8 om hur du kan använda *Xdvi*. Notera ordet som avstavats felaktigt. Gå tillbaka till dokumentfilen och hjälp LaTeX på traven med avstavningen. Se avsnitt 6.2 på sidan 38 för information om hur du gör det.

Fungerar inte detta, utveckla texten en aning genom att t.ex. lägga till en eller ett par extra ord som kanske till och med förtydligar stycket en aning. Prova igen!

Denna varningstext uppstår gärna då verbatim text skrivs med alltför långa rader. Se avsnitt 10.2.1 på sidan 56 för mer information om verbatim text.

13.4 Undefined control sequence

Detta är LaTeX sätt att säga ”jag förstår inte det här” och feltexten kan se ut som

```
! Undefined control sequence.
l.1845 \secion
              {Undefined control sequence}
```

På rad 1845 har man av misstag skrivit `\secion` i stället för `\section`.

Lösning

Gå tillbaka till dokumentfilen och rätta till felet genom att skriva rätt.

13.5 Label(s) may have changed

Då man ändrat på texten så att `label`:s har hamnat på nya sidor, t.ex. genom att ändra textstorleken från `10pt` till `12pt` eller genom att lägga till eller ta bort delar av text, får man följande varningstext från LaTeX:

```
LaTeX Warning: Label(s) may have changed. Rerun to get
cross-references right.
```

Detta är inget allvarligt och det enda sätt på vilket detta syns i dokumentet är att någonting kanske hänvisar till fel sida, figur eller ekvation. Lösningen är helt enkelt att TEX:a dokumentet ännu en gång.

TEX:ningen av dokumentet avbryts inte av denna varning.

13.6 There were undefined references

Får man varningstexten

```
LaTeX Warning: There were undefined references.
```

i slutet av TEX:ningen så har man antagligen dessutom fått ett meddelande av typen

```
LaTeX Warning: Reference `sec:latexprylar'
on page 42 undefined on input line 1953.
```

lite längre upp i den text som LaTeX alltid skriver ut under TEX:ningens gång. Detta innebär att man försökt referera till en `label` som inte finns.

Lösning

Öppna dokumentfilen och se om du verkligen har skrivit rätt. Antagligen finns felet någonstans i en `\label{...}`, `\ref{...}` eller i en `\pageref{...}`. Radnumret där referensen skedde står i feltexten.

> **OBS:** Om man just lagt till en `label` har denna inte ännu hunnit hamna i AUX-filen, eftersom detta var första TEX:ningen av filen efter tillägget. En ny TEX:ning hjälper då.

Bilaga A

Matematiska symboler

Här följer ett urval av matematiska symboler som kan användas i matematiska formler.

En symbol som föregås av `\not` får ett streck genom sig. På så sätt kan t.ex. $\not\in$ eller $\not\exists$ skapas genom att skriva `\not \in` resp. `\not \exists`.

En variabel kan göras **fet** genom att använda kommandot `\mathbf{...}`. Texten `d\mathbf{r}` typsätts som $d\mathbf{r}$ och man får $\mathbf{N}_{0^{+}}$ genom att skriva `\mathbf{N}_{0^{+}}`. Detta kommando fungerar inte på grekiska tecken.

Variabler kan också göras feta genom att använda `\boldsymbol{...}`-kommandot. Med detta kommando blir variablerna dessutom fortfarande lutande, som t.ex. $\boldsymbol{\gamma}$ och $\boldsymbol{Q}$. Detta kommando fungerar på alla symboler.

Vill man skapa tecken av typen $\mathbb{C}$ eller $\mathbb{R}$ skriver man `\mathbb{C}` resp. `\mathbb{R}`. Detta kommando fungerar endast på stora bokstäver och kräver att man har lagt följande rad före `\begin{document}`:

```
\usepackage{amssymb}
```

Det går också att skapa s.k. ”kalligrafiska” tecken som t.ex. $\mathcal{N}$ eller $\mathcal{A}$. Dessa skapas m.h.a. kommandot `\mathcal{...}` men kommandot kan bara användas på stora bokstäver (versaler).

Alla symboler som har en liten stjärna ($\star$) på sig kräver att man lägger till följande rad före `\begin{document}`:

```
\usepackage{latexsym}
```

Detta för att dessa symboler ej tillhör den egentliga LaTeX standarden.

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

Tabell A.1: Accenter.

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

Tabell A.2: Grekiska bokstäver (stora).

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

Tabell A.3: Grekiska bokstäver (små).

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

Tabell A.4: Binära operatorer.

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

Tabell A.5: Max, min, logaritmer och liknande.

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

Tabell A.6: Stora symboler.

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

Tabell A.7: Relationssymboler.

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

Tabell A.8: Pilsymboler.

<table>
<tr><td>$\uparrow$</td><td><code>\uparrow</code></td><td>$\Uparrow$</td><td><code>\Uparrow</code></td><td>$\downarrow$</td><td><code>\downarrow</code></td><td>$\Downarrow$</td><td><code>\Downarrow</code></td></tr>
<tr><td>$\{$</td><td><code>\{</code></td><td>$\}$</td><td><code>\}</code></td><td>$\updownarrow$</td><td><code>\updownarrow</code></td><td>$\Updownarrow$</td><td><code>\Updownarrow</code></td></tr>
<tr><td>$\lfloor$</td><td><code>\lfloor</code></td><td>$\rfloor$</td><td><code>\rfloor</code></td><td>$\lceil$</td><td><code>\lceil</code></td><td>$\rceil$</td><td><code>\rceil</code></td></tr>
<tr><td>$\langle$</td><td><code>\langle</code></td><td>$\rangle$</td><td><code>\rangle</code></td><td>$\vert$</td><td><code>|</code></td><td>$\Vert$</td><td><code>\|</code></td></tr>
<tr><td>$/$</td><td><code>/</code></td><td>$\backslash$</td><td><code>\backslash</code></td><td></td><td></td><td></td><td></td></tr>
</table>

Tabell A.9: Avgränsare.

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

Tabell A.10: Matematiska konstruktioner.

$\ldots$	`\ldots`	$\cdots$	`\cdots`	$\vdots$	`\vdots`	$\ddots$	`\ddots`
$\aleph$	`\aleph`	$\prime$	`\prime`	$\forall$	`\forall`	∞	`\infty`
$\hbar$	`\hbar`	$\emptyset$	`\emptyset`	$\exists$	`\exists`	∇	`\nabla`
$\surd$	`\surd`	$\Box$	`\Box*`	$\triangle$	`\triangle`	$\Diamond$	`\Diamond*`
$\imath$	`\imath`	$\jmath$	`\jmath`	ℓ	`\ell`	$\neg$	`\neg`
$\top$	`\top`	$\flat$	`\flat`	$\natural$	`\natural`	$\sharp$	`\sharp`
$\bot$	`\bot`	$\wp$	`\wp`	$\clubsuit$	`\clubsuit`	$\diamondsuit$	`\diamondsuit`
$\mho$	`\mho*`	$\angle$	`\angle`	$\heartsuit$	`\heartsuit`	$\spadesuit$	`\spadesuit`
$\Re$	`\Re`	$\Im$	`\Im`	∂	`\partial`		

Tabell A.11: Övriga symboler.

Bilaga B

$\mathcal{AMS}$-symboler

De matematiska symboler som finns i denna bilaga har skapats av $\mathcal{AMS}$ (*American Mathematical Society*) och kräver `amssymb`-paketet.

För att använda dem behöver man lägga till följande rad i dokumenthuvudet:

```
\usepackage{amssymb}
```

$\digamma$	`\digamma`	$\varkappa$	`\varkappa`
$\beth$	`\beth`	$\daleth$	`\daleth`
$\gimel$	`\gimel`		

Tabell B.1: Grekiska och hebreiska $\mathcal{AMS}$-symboler.

$\ulcorner$	`\ulcorner`	$\urcorner$	`\urcorner`
$\llcorner$	`\llcorner`	$\lrcorner$	`\lrcorner`

Tabell B.2: $\mathcal{AMS}$-avgränsare.

$\dashrightarrow$	`\dashrightarrow`	$\dashleftarrow$	`\dashleftarrow`
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`
$\leftrightarrows$	`\leftrightarrows`	$\rightleftarrows$	`\rightleftarrows`
$\Lleftarrow$	`\Lleftarrow`	$\Rrightarrow$	`\Rrightarrow`
$\twoheadleftarrow$	`\twoheadleftarrow`	$\twoheadrightarrow$	`\twoheadrightarrow`
$\leftarrowtail$	`\leftarrowtail`	$\rightarrowtail$	`\rightarrowtail`
$\looparrowleft$	`\looparrowleft`	$\looparrowright$	`\looparrowright`
$\leftrightharpoons$	`\leftrightharpoons`	$\rightleftharpoons$	`\rightleftharpoons`
$\curvearrowleft$	`\curvearrowleft`	$\curvearrowright$	`\curvearrowright`
$\circlearrowleft$	`\circlearrowleft`	$\circlearrowright$	`\circlearrowright`
$\Lsh$	`\Lsh`	$\Rsh$	`\Rsh`
$\upuparrows$	`\upuparrows`	$\downdownarrows$	`\downdownarrows`
$\upharpoonleft$	`\upharpoonleft`	$\downharpoonleft$	`\downharpoonleft`
$\upharpoonright$	`\upharpoonright`	$\downharpoonright$	`\downharpoonright`
$\rightsquigarrow$	`\rightsquigarrow`	$\leftrightsquigarrow$	`\leftrightsquigarrow`
$\multimap$	`\multimap`		

Tabell B.3: $\mathcal{AMS}$-pilar.

$\leqq$	\leqq	$\geqq$	\geqq
$\leqslant$	\leqslant	$\geqslant$	\geqslant
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox
$\lessdot$	\lessdot	$\gtrdot$	\gtrdot
$\lll$	\lll	$\ggg$	\ggg
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless
$\bumpeq$	\bumpeq	$\Bumpeq$	\Bumpeq
$\fallingdotseq$	\fallingdotseq	$\risingdotseq$	\risingdotseq
$\doteqdot$	\doteqdot	$\backsim$	\backsim
$\backsimeq$	\backsimeq	$\eqcirc$	\eqcirc
$\circeq$	\circeq	$\triangleq$	\triangleq
$\thicksim$	\thicksim	$\thickapprox$	\thickapprox
$\backepsilon$	\backepsilon	$\approxeq$	\approxeq
$\smallsmile$	\smallsmile	$\smallfrown$	\smallfrown
$\Vdash$	\Vdash	$\Vvdash$	\Vvdash
$\vDash$	\vDash	$\varpropto$	\varpropto
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq
$\Subset$	\Subset	$\Supset$	\Supset
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc
$\precsim$	\precsim	$\succsim$	\succsim
$\precapprox$	\precapprox	$\succapprox$	\succapprox
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq
$\blacktriangleleft$	\blacktriangleleft	$\blacktriangleright$	\blacktriangleright
$\therefore$	\therefore	$\because$	\because
$\shortmid$	\shortmid	$\shortparallel$	\shortparallel
$\between$	\between	$\pitchfork$	\pitchfork

Tabell B.4: Binära $\mathcal{AMS}$-relationer.

$\dotplus$	\dotplus	$\smallsetminus$	\smallsetminus
$\Cap$	\Cap	$\Cup$	\Cup
$\barwedge$	\barwedge	$\veebar$	\veebar
$\doublebarwedge$	\doublebarwedge	$\intercal$	\intercal
$\boxminus$	\boxminus	$\boxplus$	\boxplus
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot
$\divideontimes$	\divideontimes	$\centerdot$	\centerdot
$\ltimes$	\ltimes	$\rtimes$	\rtimes
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes
$\curlywedge$	\curlywedge	$\curlyvee$	\curlyvee
$\circleddash$	\circleddash	$\circledast$	\circledast
$\circledcirc$	\circledcirc		

Tabell B.5: Binära $\mathcal{AMS}$-operationer.

$\hbar$	\hbar	$\hslash$	\hslash
$\vartriangle$	\vartriangle	$\triangledown$	\triangledown
$\blacktriangle$	\blacktriangle	$\blacktriangledown$	\blacktriangledown
$\square$	\square	$\lozenge$	\lozenge
$\blacksquare$	\blacksquare	$\blacklozenge$	\blacklozenge
$\circledS$	\circledS	$\Game$	\Game
$\angle$	\angle	$\measuredangle$	\measuredangle
$\sphericalangle$	\sphericalangle	$\eth$	\eth
$\nexists$	\nexists	$\mho$	\mho
$\Finv$	\Finv	$\Bbbk$	\Bbbk
$\backprime$	\backprime	$\varnothing$	\varnothing
$\bigstar$	\bigstar	$\complement$	\complement
$\diagup$	\diagup	$\diagdown$	\diagdown

Tabell B.6: Övriga $\mathcal{AMS}$-symboler.

$\iint\limits_{\partial \Gamma}$	\iint \limits_{\partial \Gamma}
$\iiint\limits_{\partial \Gamma}$	\iiint \limits_{\partial \Gamma}
$\iiiint\limits_{\partial \Gamma}$	\iiiint \limits_{\partial \Gamma}
$\idotsint\limits_{\partial \Gamma}$	\idotsint \limits_{\partial \Gamma}

Tabell B.7: $\mathcal{AMS}$ multipelintegraler och gränser.

$\overrightarrow{\lambda(A^{T}A)}$	\overrightarrow{\lambda(A^{T}A)}
$\underrightarrow{\lambda(A^{T}A)}$	\underrightarrow{\lambda(A^{T}A)}
$\overleftarrow{\lambda(A^{T}A)}$	\overleftarrow{\lambda(A^{T}A)}
$\underleftarrow{\lambda(A^{T}A)}$	\underleftarrow{\lambda(A^{T}A)}
$\overleftrightarrow{\lambda(A^{T}A)}$	\overleftrightarrow{\lambda(A^{T}A)}
$\underleftrightarrow{\lambda(A^{T}A)}$	\underleftrightarrow{\lambda(A^{T}A)}

Tabell B.8: $\mathcal{AMS}$ över- och under-pilar.

Bilaga C

Övrig information

C.1 Hur får jag tag på LaTeX?

Ett bra ställe att börja leta på, med en WWW-browser, är

```
ftp://ftp.sunsite.kth.se/pub/tex/
```

Där finns *allt* som har med LaTeX att göra. Om du t.ex. vill ha en implementation av LaTeX till din dator så kan du söka i katalogen `systems`. Där finner du LaTeX för de flesta vanliga persondatorsystem. I filen `CTAN.sites` finner du andra ställen som har samma material som `ftp.sunsite.kth.se`.

Ett annat ställe på WWW som kan vara värt ett besök, inte minst p.g.a. att det är lättare att verkligen *hitta* olika paket som man vill använda, är

```
ftp://ftp.sunsite.kth.se/pub/tex/help/Catalogue/catalogue.html
```

Den ”officiella hemsidan” för LaTeX på WWW finns på sidan

```
http://www.latex-project.org/
```

C.2 En not om teTeX

LaTeX-distributionen teTeX för UNIX-system kommer med en utomordentlig mängd makropaket och dokumentation. Normalt ligger dokumentationen i katalogen

```
SOMEPATH/texmf/doc/
```

Här kan `SOMEPATH` vara t.ex. `/opt/teTeX`, `/usr/lib` eller `/usr/local` beroende på hur katalogstrukturen ser ut på ditt UNIX-system (detta kan variera kraftigt).

I `doc`-katalogen finner du ett antal underkataloger. Är du intresserad av information om t.ex. paketet `wasy` kan du ge UNIX-kommandot

```
find SOMEPATH/texmf/doc/ -name "wasy*.dvi" -print
```

i ett kommandofönster för att se om någon dokumentation i DVI-format finns tillgänglig. Byt ut `SOMEPATH` mot rätt sökväg först. Om du använder *Linux* kan du ge kommandosekvensen

```
locate wasy | grep .dvi
```

för att göra ungefär samma sak.

C.3 Om denna handledning

C.3.1 Författaren

Jag heter Andreas Kähäri och nås säkrast via e-post. Min e-postadress är i skrivande stund bl.a.

`andkaha@users.sourceforge.net`

Jag svarar (mer eller mindre) gärna på frågor om LaTeX (efter förmåga) eller skickar vidare frågan till rätt person alternativt ger en WWW-adress till en sida som innehåller svaret på frågan.

Om jag inte svarar inom rimlig tid så kan jag rekomendera det engelskspråkiga diskussionsforumet `comp.text.tex`[1] som kan nås via t.ex. Deja.com på

`http://deja.com/`

C.3.2 SourceForge

Detta kompendium finns sedan i juni 2000 tillgängligt via WWW på adressen

`http://latexintro.sourceforge.net/`

Där finns, förutom PDF- och Postscript-versioner av kompendiet, även ett publikt diskussionsforum för diskussioner kring innehållet i kompendiet (inga LaTeX-frågor där tack). Där kan man t.ex. komma med förslag på förbättringar eller påpeka felaktigheter.

SourceForge erbjuder mig som författare plats på en hårddisk för själva kompendiet och för en websida men också för själva LaTeX-koden som jag skapar PDF- och Postscript-filerna från. Detta gör det enkelt för dig som läsare att ta del av koden och själv införa förändringar och korrigera fel.

C.3.3 Återkoppling

Skicka gärna ett e-brev till mig om

- Om du hittar fel.
- Om du gör ändringar i kompendiet.
- Om du gillar kompendiet.
- Om du använder kompendiet i undervisningssyfte.

C.3.4 Kompendiets historik

Alternativa titlar till detta kompendie är ”Typsättning för satta”, ”LaTeX för lata”, ”ε and friends”, ”Makron för maktlösa”, ”Knep och knåp med Emacs”, ”LaTeX allergy survival kit” m.fl.[2]

hösten 1996 Detta kompendiums första inkarnation var i form av ca. 20 overhead-sidor (ungefär två föreläsningstillfällen) som användes i ett kursmoment i kursen ”Informationsteknologisk baskurs” vid IT/TDB[3] under hösten 1996. Materialet bestod då i huvudsak av en ”så-gör-du”-text tillsammans med de grundläggande momenten i LaTeX. Kursen är sedan dess kliniskt ren från typsättning, men i och med att kopior av overhead-sidorna kom i omlopp bland studenter var kompendiets rot satt i bördig jord.

[1] Här gäller, liksom på alla andra publika diskussionsforum, netikett-regler. Du bör läsa forumets FAQ innan du postar frågor.

[2] Samtliga titlar är naturligtvis uppfunna under sena kvällar då syret och kaffet i rummet börjat ta slut.

[3] Institutionen för informationsteknologi (teknisk databehandling), Uppsala universitet.

vintern 1996–1997 Under vintern 1996–1997 utvidgades overhead-materialet till att omfatta lite mer avancerade moment, bl.a. kapitlen om figurer och referenser.

våren 1997 Under våren 1997 övergavs idén med att ha materialet i overhead-variant till fördel för kompendievarianten på allmän begäran.

sommaren 1997 Kapitlen om litteraturförteckningar och egendefinierade kommandon lades till och så gott som all text skrevs om, omorganiserades och utvidgades, också detta på allmän begäran.

augusti 1997 Första kompendieutgåvan. Från och med nu måste man betala *pengar* för det hela.

sommaren 1998 Avsnittet om tabeller skrevs om och ett avsnitt om verbatim text lades till. Tabellerna i dokumentet gjordes flytande. Allmän uppsnyggning i texten. Bilagorna med matematiska symboler snyggades upp. Idén med varningar och stiltips föddes och implementerades.

augusti 1998 Andra kompendieutgåvan.

juni 2000 Kompendiet når version 3.0. Efter en alldeles för lång väntan fixade jag äntligen fram en publik och *gratis* version av kompendiet! Aningen småändringar har införts men inget drastiskt. Kompendiet ligger numera på SourceForge och är licensierad under *The GNU Free Documentation License*.

Att göra: Vill ha en `amsmath`-avdelning någonstans i handledningen eftersom avsnittet om matematiska formler är lite ”stökigt”, men det blir ett framtida projekt är jag rädd. Experimenterar dessutom med ett sätt att få in fler exempelformler, t.ex. i en egen bilaga. Ett index behövs snart.

Det dokument du just nu håller i din hand producerades med LaTeX den 11:e juni 2000.

Appendix D

GNU Free Documentation License

Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[GMS94] Michael Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion.* Addison–Wesley Publishing Company, 1994.

[Knu87] Donald Erving Knuth. *The TeXbook.* Addison–Wesley Publishing Company, 1987.

[Lam94] Leslie Lamport. *LaTeX, A Document Preparation System.* Addison–Wesley Publishing Company, 2nd edition, 1994.