

L^AT_EX och prov D1:*Snöflingan* D om texter och L^AT_EXi kursen D0015E

Håkan Jonsson

Institutionen för system- och rymdteknik
Luleå tekniska universitet

16 september 2025

Sammanfattning

Prov D1:Snöflingan går ut på att skapa ett dokument identiskt med ett annat med hjälp av L^AT_EX. Här ges en förklaring till innehållet i den startfil du får.

1 Allmänt

L^AT_EX erbjuder fantastiska möjligheter att producera professionellt typsatt text, men också lika stora möjligheter att göra fel. Du måste därför vara *mycket noggrann* då du skriver ditt L^AT_EX-manus. Minsta tecken har betydelse och även små fel, som att råka använda en stor bokstav där det ska vara en liten, gör att varje försök att generera ett dokument från manuset rasar samman som ett korthus.

Skriv manuset i mycket små steg och generera dokument mellan stegen, så får du eventuella fel i småportioner och kan enklare åtgärda dem. Jag gör alltid så här och jag råder även dig att göra det.

1.1 Layout

Skilj mellan *innehåll* och *utseende*. Koncentrera dig på innehållet, dvs att text och formler får korrekt innehåll, och överlåt sen åt L^AT_EX att bestämma

utseendet. Kanske detta känns ovant, för WYSIWYG¹-program som t ex Microsoft Word bygger ju på att skribenter också bestämmer layouten. Men \LaTeX är inte något WYSIWYG-program. \LaTeX är programmerat med hjälp av de bästa layoutreglerna och layoutar bättre än oss alla i 99 fall av 100.

Om du trots allt till slut tycker att layouten inte är bra så vänta med uttryckliga ändringar av den tills du fått allt innehåll korrekt(!) Alltså:

- Först se till att allt innehåll finns inknappat och går att generera pdf från utan att det blir särskilt snyggt.
- Sen jobba med hur layouten blivit.

Annars skapar du bara en massa problem för dig själv. Även små ändringar av innehållet kan orsaka mycket stora ändringar av hur \LaTeX utformar dokumentets layout.

1.2 Kommentarer

\LaTeX -systemet ignorerar allt på en rad efter ett procenttecken (%). Med % kan man således inkludera kommentar i sitt manus som inte påverkar det färdiga dokumentet men som kan vara viktiga för någon som läser manuset.

1.3 Kommandon och omgivningar

Ett \LaTeX -kommando startar med ett bakåt-snedstreck (\). Sen följer kommandots namn och eventuella argument². Vill man t ex ha **text med fet stil** använder man kommandot `\textbf{text med fet stil}`. Vill man istället att *text ska framhävas* så skriver man `\emph{text ska framhävas}`. Det finns väldigt många olika kommandon och ett råd är att snarare googla efter/slå upp detaljer då du behöver dem än att lära dig dem alla utan till.

Förutom kommandon finns det även \LaTeX -omgivningar (*environments* på engelska). Dessa inleds med `\begin`, avslutas med `\end` och påverkar det som skriv mellan `begin-end`. Exempel: Om man vill

centrera text så här

använder man omgivningen `center` och skriver

```
\begin{center}
  centrera text så här
\end{center}
```

¹What-You-See-Is-What-You-Get.

²Jämför med hur funktioner och metoder används i programmeringsspråk som `python`.

Om du läser manuset till detta dokument kommer du att se att jag använt omgivningen *verbatim* för att få in centreringsomgivningen ovan i min text.

1.4 Tomrum

En *tomrad* (eller flera efter varandra) bland text markerar *nytt stycke*.

Detta kan märkas genom att första raden dras in eller att L^AT_EX lägger in lite tomrum vertikalt mellan styckena. Var noga med eventuella tomrader du väljer att ta med i ditt manus.

Däremot spelar godtyckligt många mellanslag mellan ord i text ingen annan roll än att de byts ut mot ett lagom stort mellanrum oavsett hur många de är. Därför kan man i regel fritt *indentera rader*, dvs stoppa in mellanslag i början av rader för att visa på hur på varandra följande rader hör ihop, och därigenom dramatiskt öka manusets läsbarhet.

Jag brukar indentera sånt som är inne i en omgivning minst 2 mellanslag relativt närmast omgivande omgivning. Det gör att jag enklare ser vad som tillhör vad. Om jag inte hinner indentera när jag skriver så återvänder jag till texten och gör det senare. Jag råder dig att indentera på samma sätt som mig, och även då du programmerar datorer. I vissa språk, t ex python, har indenteringen även betydelse för slutresultatet.

2 En genomgång av koch-start.tex

Här följer nu en genomgång av startfilen `koch-start.tex` som du ska utgå från. Uppgiften är att lägga till text och kod. Du behöver inte ta bort något, utan allt i filen ska vara kvar.

2.1 Inledningen

```
\documentclass[12pt,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage{graphicx}
\usepackage{amsmath, amsthm, amssymb}
\usepackage[a4paper,includeheadfoot,margin=2.54cm]{geometry}
\newtheorem{theorem}{Theorem}
```

Varje L^AT_EX-manus inleds med lite allmänna deklARATIONER och importer, och så är också fallet med detta manus.

- `\documentclass[12pt,a4paper]{article}` säger att detta är ett manus för en artikel som ska typsättas med 12 punkters typsnitt och passa på ett A4-ark.
- `\usepackage[utf8]{inputenc}` anger den teckenkodning som används. Detta är datorspecifikt så du kan behöva ändra `utf8` till något annat t ex `latin1`.

`\usepackage` är ett kommando som importerar andra kommandon och omgivningar än de som det grundläggande L^AT_EX-systemet består av. Kommandot gör det importerade tillgängligt när man skriver manuset.
- `\usepackage{graphicx}` gör det möjligt att inkludera grafik, t ex JPEG-bilder eller PDF-filer. Detta gör man i praktiken sen med kommandot `\includegraphics`.
- `\usepackage{amsmath, amsthm, amssymb}` utökar stödet för matematik.
- Den långa raden `\usepackage[a4paper,includeheadfoot,margin=2.54cm]{geometry}` förtydligar att sidorna ska ha marginal på 2.54mm (1 tum).
- `\newtheorem{theorem}{Theorem}` introducerar `theorem`, en ny omgivning för matematiska satser/teorem.

Inledningen kan även definiera titel, författarnamn, datum, rubrikstilar och annat. Men så görs inte här. Inledningen ovan ska vara som den är, dvs du ska inte ändra i den.

2.2 Resten av manuset

Här följer nu en genomgång, rad för rad, av resten av `koch-start.tex`. Denna genomgång gör det lättare för dig att komplettera manuset och få det rätt.

```
\begin{document}
```

Börjar beskrivningen av dokumentinnehållet. Allt som skrivs mellan denna rad och raden `\end{document}` i slutet av manuset, och i den ordning det skrivs (uppfifrån och ned, vänster till höger), anger dokumentinnehåll, dvs text blandad med kommandon som säger vad textdelarna är för något.

```
\section{A Famous Fractal: The Koch Snowflake}
```

Att använda `\section` ger en automatnumrerad huvudrubrik. Argumentet till kommandot, i detta fall `The Koch Snowflake`, är rubriktexten. \LaTeX kommer att välja en lagom storlek på rubriken som passar till det slutgiltiga dokumentets storlek och text.

```
The \emph{Koch snowflake},
```

Med `\emph` markeras *betonad* text.

```
Helge von Koch~\cite{koch}.
```

`\cite` används för att referera till en referens, som beskrivs i omgivningen `thebibliography` i slutet av manuset. En referens är typiskt en artikel eller rapport. Här refererar vi referensen `koch`, vilket är varför vi skriver `\cite{koch}`. \LaTeX kommer att byta ut `\cite{koch}` mot ett löpnummer inom hakparenteser.

Har man flera referenser ger \LaTeX varje ett eget nummer och ser till att utbytena blir rätt. Detta är mycket smidigt eftersom man inte själv behöver hålla reda på löpnumren, utan kan helt överlåta detta åt \LaTeX -systemet. Däremot måste man (förstås) själv definiera referenserna och ge dem unika nyckelord (i detta fall `koch`).

Notera det lilla vågtecknet (`~`) i `Koch~\cite`. Detta är nödvändigt för att förhindra radbrytning just där. Utan detta tecken finns risk att referensen inom hakparenteser, som \LaTeX skapar och lägger in, annars skulle kunna hamna först på ny rad. Så typsättning vore fel (och ful).

```
\begin{figure}[h] \label{koch}
```

`figure` är en omgivning för figurer som man vill ha bildtext till, t ex bilder. `[h]` anger att vi vill ha figuren just **h**är, och inte någon annanstans(!) Beroende på \LaTeX :s layoutregler *kan* figuren ändå hamna på annan plats, vilket man då kan åtgärda. Men sådan åtgärd ska man alltså vidta först när allt annat innehåll är korrekt (inte omedelbart när man till sin fasa ser att figuren hamnar på fel ställe).

`\label` är en etikett för något i dokumentet som man sen vill kunna referera till som avsnitt, ekvationer, sidor, bilder, tabeller mm. Till `\label` fogar man ett unikt namn på etiketten som man sen känner igen. En etikett för

ett inledande avsnitt skulle kunna vara `sec:intro` medan man för en ekvation kanske använder `eq:pythagoras`. De här prefixen `sec:` och `eq:` är inget \LaTeX -systemet kräver, utan bara till för att man snabbt ska se vad för något en referens leder till. För en figur brukar jag skriva t ex `\label{fig:circle}`, där `fig:` gör det lätt att skilja etiketten från etiketter för avsnitt och ekvationer. Du får döpa dina etiketter som du vill men håll dig till vanliga bokstäver i etikettnamnen.

Avsnitt, ekvationer, figurer och annat numreras separat inom respektive sort och stigande. Det går dock att ändra detta (men det behöver man sällan).

```
\centering
```

`\centering` inne i en `figure` ger centrering av figuren. Omgivningen `center` fungerar likartat men ska inte användas just här.

```
\includegraphics[width=10cm]{snowflake.jpg}
```

Kommandot `\includegraphics` säger att här ska grafik in från en fil vars namn ett argument anger. Vår fil heter `snowflake.jpg` och dessutom föreskriver vi att vi vill att bilden får en bredd på 10 cm. \LaTeX kommer då att skala om bilden så den får bredden 10 cm och behåller sina proportioner.

Skriver man istället `width=\textwidth` skalas bilden istället om så den blir lika bred som texten. Man kan skippa att ange en bredd men då får bilden sin naturliga bredd (oskalad), vilket ofta ser dåligt ut.

Bilder är något \LaTeX inte klarar av särskilt väl. Bildhantering i \LaTeX är krångligt, blir ofta fel, och därför har jag inkluderat hela det kommando du ska använda för att få in bilden på snöflingans första konstruktionssteg.

```
\caption{The Koch snowflake }
```

`\caption` anger helt enkelt en bildtext.

```
\end{figure}
```

Varje omgivning börjar med `begin` och måste avslutas med `end`, och här avslutas således `figure`-omgivningen.

```
infinite number of times:
\begin{quote}
  \textit{First, divide.}
\end{quote}
```

Omgivningen `quote` lägger in ett längre citat, och indenterar *lagom mycket* från både höger och vänster. `\textit` ger kursiv stil.

```
Figure~\ref{koch}.
```

Kommandot `\ref` används för att referera till något i dokumentet som märkts ut med en etikett. Om man t ex lagt in en (enda) figur/bild i sitt dokument, och i denna inkluderat etiketten `\label{koch}`, så kommer `\ref{koch}` i texten sen att bytas ut mot [1]. Skulle figuren vara den andra figuren i ordning, sker utbytet istället till [2] osv.

```
\begin{theorem}  
  infinite length.  
\end{theorem}
```

Omgivningen `theorem` typsätter en matematisk sats (ett teorem).

```
\begin{proof}
```

Efter `theorem`, ett matematiskt påstående, passar det med ett bevis av påståendet. Det är vad omgivningen `proof` används för. I ditt manus ska du skriva kod för två bevis (ja, även två teorem) som alltså läggs in i omgivningen `proof`. Notera att `proof` automatiskt lägger till en passande slutsymbol.

```
 $\Delta$  
 $N_i$  
 $L_i$
```

Matematisk löptext märks i L^AT_EX ut genom att omges av dollartecken (\$). Skriver man t ex `$f(x)+f(2x)/2+f(3x)/3$` typsätts det som matematisk text, dvs

$$f(x) + f(2x)/2 + f(3x)/3,$$

och inte som vanlig text, dvs $f(x) + f(2x) / 2 + f(3x) / 3$. Lagg noga märke till stilskillnaden. Skilj noga mellan vad i manuset som är vanlig text och vad som är matematik. Det är fel att typsätta matematik, t ex variabler, som vanliga text.

`\Delta` ger den grekiska bokstaven stora D (dvs Δ). `Ni` och `Li` ger i matematisk text N_i respektive L_i . Rent generellt ger understrykningstecken () ett *index* i matematisk text.

```
Then,  
\begin{displaymath}
```

`displaymath` ger en formel som typsätts fritt i ett större vertikalt mellanrum. Man behöver inga dollartecken i omgivningen, utan \LaTeX förutsätter att det som skrivs där måste vara matematik.

Samma matematiska text kan i regel antingen typsättas i löptext, med dollartecken, eller fritt med bl a `displaymath`; se även `equation` nedan.

```
=  
\begin{cases}  
3, & \text{if } n=0 \\  
\end{cases}
```

`cases` används för ekvationer med många fall. Koden `\text{en mening}` i matematiska formler gör att `en mening` typsätts som vanlig löptext, inte som matematik. Detta är användbart när man vill lägga till förklaringar eller villkor (som här där värdet i första fallet är 3 men endast om $n = 0$).

Sekvensen `\\` betyder att här ska en radbrytning alltid in. Så måste man skriva i en del matematikomgivningar (t ex `cases`) men annars ska man endast använda det sparsamt och försiktigt. Det förstör nämligen \LaTeX :s egen typsättning och ger (oftast) ett fult resultat.

```
\end{displaymath}
```

Detta avslutar mera luftig typsättning av matematik.

```
This  
\begin{equation}  
 \label{eq:1}  
 \cdot  
\end{equation}
```

Omgivningen `equation` används som `displaymath` men ger automatiskt ett ekvationsnummer som skrivs ut i marginalen. Koden

```
\begin{equation}  
 \label{eq:first}  
 f(x) = x^2 + 2x + 9  
\end{equation}
```


ger t ex

$$f(x) = x^2 + 2x + 9 \quad (1)$$

och refererar man sen till etiketten `eq:first` med `Eq.\ref{eq:first}` gör L^AT_EX om detta till

Eq. ??

Som tidigare påpekats är den ”våg” (`~`) som syns i koden (och som även kallas *tilde*) ett fast mellanslag, ett slags klister som beordrar L^AT_EX att behandla det ihopklistrade som ett ord. Detta används för att hindra radbrytning mellan Eq. och siffran `??`. I just detta sammanhang, precis som för `\cite`, får radbrytning inte ske, och L^AT_EX måste då upplysas om det.

`\cdot` ger en centrerad punkt i matematik. En sådan brukar användas för att visa multiplikation mellan två tal eller där man annars vill trycka på att det verkligen är multiplikation. Annars ska man aldrig använda en punkt för multiplikation. Alltså, man skriver $f(x)g(x)$ och inte $f(x) \cdot g(x)$. Däremot måste man skriva $3 \cdot 4$ för annars blir det 34, dvs ett tal och inte en multiplikation.

```
while
\begin{equation}
\label{eq:5}
L_n = \frac{L_{n-1}}{3} =.
\end{equation}
```

Kommandot `\frac` ger ett bråk. Ovan är L_{n-1} täljare och 3 nämnare.

`L_{n-1}` gör hela uttrycket $n - 1$ till index. Klammerparenteser kan alltid användas för att gruppera ihop uttryck som (bland annat) index. Skriver man t ex

$$B_{N_{i+1}+M_{j+2i}}$$

får man

$$B_{N_{i+1}+M_{j+2i}}$$

Observera att `\frac` ofta passar mindre bra i löptext, då allt kan se högst ihoptryckt ut: Koden `\frac{L_{n-1}}{3}` ger $\frac{L_{n-1}}{3}$. Då kan man ofta få finare text genom att använda ett helt vanligt snedstreck: $L_{n-1}/3$.

From Eqs.\ref{eq:1}

Återigen nödvändig användning av tecknet `~`.

```

\begin{displaymath}
N_{nL_n} =
\left( \right)
.
\end{displaymath}
it follows $\to \infty$, which.

```

`\left` och `\right` ger storleksanpassade parenteser. Skriver man

(`\frac{4}{3}`)

får man

$\left(\frac{4}{3}\right)$

medan `\left(\frac{4}{3} \right)` ger det betydligt snyggare

$\left(\frac{4}{3}\right)$

Istället för parenteser kan man använda annat t ex klammerparenteser eller hakparenteser. `\left[\frac{4}{3} \right]` skulle t ex ge

$\left[\frac{4}{3}\right]$

Observera att `\left` och `\right` alltid förekommer i par. Punkt som parentestecken gör att inget parentestecken skrivs ut. Detta kan användas både efter `\left` och `\right`. `\left\{ \frac{4}{3} \right.` skulle t ex ge

$\left\{\frac{4}{3}\right.$

(Av tekniska skäl måste man skriva `\{` om man menar en klammerparentes, tecknet alltså, för `{` och `}` används ju för att skriva kommandon och omgivningar.)

`\to` ger en högerpil och `\infty` ger tecknet ∞ , oändligheten.

```

\end{proof}

```

Beviset är klart och med `\end{proof}` avslutas det med en symbol till höger.

The Koch snowflake has finite area.

Ett nytt teorem inleder den andra delen av dokumentet. Till detta ska fogas ett bevis.

```
In an iteration,, the number of new triangles $T_n$,
Eq.~\ref{eq:1},
can be simplified to

\label{eq:2}
```

Början på beviset.

```
$a_n$
\begin{displaymath}
a_0=
\end{displaymath}
```

Notera att i matematikomgivningar behövs inga dollartecken. `\sqrt` ger kvadratroten ur något. `\sqrt{x}` blir \sqrt{x} .

```
 $\Delta$, the initial equilateral triangle,, or
\begin{equation}
\label{eq:3}
a_n = \frac{a_{n-1}}{9} = \ldots .
\end{equation}
```

`\ldots` ger tre punkter på textens baslinje och används för att markera att något utelämnats.

```
Eqs.~\ref{eq:2} and \ref{eq:3}
\begin{equation*}
```

Omgivningen `equation*` fungerar som `displaymath`.

```
b_n = = \left( \cdot 4^n \right) \left( a_0 \right) =.
\end{equation*}
```

Tecknet `^` ger ”upphöjt i”, dvs en exponent i matematisk text. Exempel: Att skriva `$x^{f(x)}$` ger $x^{f(x)}$. Notera att du alltså kan använda klammerparenteser för att gruppera ihop hela uttryck som exponenter på samma sätt som du ovan tidigare kunde använda dem för att ha uttryck som index (se sidan ??).

Har man en enkel exponent behövs inga klammerparenteser. För att få x^2 räcker det att skriva `x^2`.

```
total area
\begin{align*}
A &= a + \sum_{k=1}^n b_k \\
&= a_0 \left( 1 + \left( \frac{r}{R} \right)^k \right) \\
&= .
\end{align*}
```

Omgivningen `align*` är till för ekvationer som går över många led och rader. Notera hur `\\` används i `align*` för att avsluta rader/fall. Raderna justeras i förhållande till varandra så att de tecken som följer efter `&`-tecknen (i detta fall likamedtecken) hamnar under varandra.

`\sum` ger ett summatecken i matematisk text. Om man ger L^AT_EX-koden `\sum_{k=1}^n b_k` får man t ex $\sum_{k=1}^n b_k$ i löptext och

$$\sum_{k=1}^n b_k$$

i mera luftig text.

```
Now, since
\begin{displaymath}
\lim_{n \rightarrow \infty} \left( \frac{r}{R} \right)^n = 0,
\end{displaymath}
```

Kommandot `\lim` används för gränsvärden. Skriver man t ex

$$\lim_{x \rightarrow \infty} \frac{1}{x} = 0$$

blir det $\lim_{x \rightarrow \infty} \frac{1}{x} = 0$ i löptext och

$$\lim_{x \rightarrow \infty} \frac{1}{x} = 0$$

i en `displaymath`.

```
\lim_{n \rightarrow \infty} A_n
```

Samma här.

```
\begin{thebibliography}{99}
```

`thebibliography` är en omgivning som bygger upp en referenslista där varje referens anges med hjälp av kommandot `bibitem`. Det är dessa som sen refereras till med kommandot `\cite`.

```
\bibitem{koch} Helge. \emph{Sur une courbe continue sans  
tangente, obtenue par une construction géométrique  
élémentaire.}, Arkiv,  
Kungliga Vetenskapsakademien. \textbf{1}, 681-702,.
```

En referens (den enda i dokumentet). Kommandot `\textbf` ger fet stil.

```
\end{thebibliography}
```

Avslutar referenslistan.

```
\end{document}
```

Avslutar hela manuset.