
Mathematik II

L^AT_EX-Kurs der Unix-AG

Andreas Teuchert

20. Juni 2011



Wiederholung: Umgebungen aus Teil I

- ▶ `equation` – eine einzeilige Gleichung
- ▶ `gather` – mehrere einzeilige Gleichungen (ohne Ausrichtung)
- ▶ `multline` – eine mehrzeilige Gleichung (feste Ausrichtung)

Was noch fehlt:

- ▶ mehrere Gleichungen mit Ausrichtung (`align` und `alignat`)
- ▶ mehrzeilige Gleichungen mit Ausrichtung (`split`)

- ▶ align: Spalten werden so plaziert, dass die Seitenbreite ausgenutzt wird
- ▶ manchmal nicht erwünscht (z. B. bei der Darstellung von Gleichungssystemen)
- ▶ Lösung: alignat verwenden (Abstand zwischen Spalten wird minimiert)
- ▶ Anzahl der Spalten muss mit angegeben werden

AMS: split-Umgebung

- ▶ wie bei `multline` können Gleichungen durch `\\` umgebrochen werden
- ▶ durch `&` kann zusätzlich eine Ausrichtung erzeugt werden
- ▶ wird eingebettet in andere Umgebungen verwendet
- ▶ stellt dann eine einzelne Gleichung dar

```
\begin{equation}
\begin{split}
x &= abc + dec + ghc \\
&= c(ab + de + gh)
\end{split}
\end{equation}
```

$$\begin{aligned} x &= abc + dec + ghc \\ &= c(ab + de + gh) \end{aligned} \tag{6}$$

\mathcal{AMS} : Eingebettete Gleichungen

- ▶ wie bei `split` werden Gleichungen in Gleichungen eingebettet
- ▶ Umgebungen: `gathered`, `aligned` und `alignedat`
- ▶ verhalten sich wie die Formen ohne „ed“; aber: keine Nummerierung, nicht ganz so breit
- ▶ Anwendung: mehrere Gleichungen mit Klammern versehen
- ▶ Unterschied zu `split`: `split` ist eine Gleichung, `...ed` sind mehrere
- ▶ `split` mit Klammern zu versehen ergibt keinen Sinn (\LaTeX versucht dann `aligned` zu verwenden)!

AMS: Eingebettete Gleichungen – Beispiel

```
\begin{equation}
\left .
\begin{aligned}
&\sum_{\nu=1}^n I_{\nu} &&= 0 \\
&\sum_{\nu=1}^n U_{\nu} &&= 0
\end{aligned}
\right\} \text{Kirchhoffsche Gesetze}
\end{equation}
```

$$\left. \begin{array}{l} \sum_{\nu=1}^n I_{\nu} = 0 \\ \sum_{\nu=1}^n U_{\nu} = 0 \end{array} \right\} \text{Kirchhoffsche Gesetze} \quad (7)$$

AMS: Untergleichungen

- ▶ mit der subequations-Umgebung können logisch zusammengehörige Gleichungen zusammengefasst werden
- ▶ Nummerierung spiegelt den Zusammenhang wider

```
\begin{subequations}
\begin{align}
x &= \frac{1}{2} at^2 + v_0t + x_0 \\
v &= at + v_0
\end{align}
\end{subequations}
```

$$x = \frac{1}{2}at^2 + v_0t + x_0 \quad (8a)$$

$$v = at + v_0 \quad (8b)$$

\mathcal{AMS} : Gleichungsnummerierung anpassen

- ▶ Erinnerung: mit `\notag` kann die Nummerierung einzelner Gleichungen unterdrückt werden
- ▶ mit `\tag{Irgendwas}` kann statt der automatischen Nummerierung beliebiger Text (hier „Irgendwas“) verwendet werden
- ▶ dieser Text wird auch für Verweise (`\ref`, `\eqref`) verwendet

AMS: Gleichungsnummerierung anpassen – Beispiel

```
\begin{align}
f_1(x) &= a_1x^2 + b_1x + c_1\\
f_2(x) &= a_2x^3 \tag{Fkt.-Gl. $f_2$} \label{eqn:f2}\\
f_3(x) &= d_4 e^x
\end{align}
Siehe auch \ref{eqn:f2}.
```

$$f_1(x) = a_1x^2 + b_1x + c_1 \tag{9}$$

$$f_2(x) = a_2x^3 \tag{Fkt.-Gl. f_2}$$

$$f_3(x) = d_4e^x \tag{10}$$

Siehe auch Fkt.-Gl. f_2 .

AMS: Verschachtelte Brüche

- ▶ verschachtelte Brüche mit `\frac` werden mit zunehmender Tiefe kleiner
- ▶ `\cfrac` (continued fractions) erhöht die Lesbarkeit

```
\begin{equation}
\frac{a}{b+\frac{c}{d+\frac{e}{f+\frac{g}{h}}}} =
\cfrac{a}{b+\cfrac{c}{d+\cfrac{e}{f+\cfrac{g}{h}}}}
\end{equation}
```

$$\frac{a}{b + \frac{c}{d + \frac{e}{f + \frac{g}{h}}}} = \frac{a}{b + \frac{c}{d + \frac{e}{f + \frac{g}{h}}}} \quad (11)$$

Brüche in Fließtext

- ▶ für Brüche in Fließtext bietet sich das nicefrac-Paket an (`\usepackage{nicefrac}`)
- ▶ mit dem `\nicefrac`-Befehl werden Zähler und Nenner durch einen Schrägstrich getrennt, wodurch der Bruch kleiner wird

`\nicefrac{1}{2}` ist hübscher als `\frac{1}{2}`.

$1/2$ ist hübscher als $\frac{1}{2}$.

Einheiten richtig setzen

Vorgaben für Einheiten

- ▶ Einheiten werden anders als Variablen nicht kursiv gesetzt
- ▶ Wiederholung aus „Große Dokumente“: zwischen Zahl und Einheit gehört ein dünnes Leerzeichen

Mögliche Lösungen

- ▶ Möglichkeit 1: `\,` (Leerzeichen) und `\mathrm{}` (nicht kursiv) verwenden (schlechte Idee)
- ▶ Möglichkeit 2: `SIunits`-Paket verwenden

- ▶ Einbinden mit `\usepackage{SIunits}`
- ▶ heißt sich mit `amssymb`:
 - ▶ Erst `amssymb` einbinden und dann `SIunits` mit `\usepackage[amssymb]{SIunits}`
 - ▶ `\square` wird neu definiert
- ▶ stellt intuitive Befehle für Einheiten und Prefixe bereit
- ▶ auch außerhalb des Mathe-Modus verwendbar

```
\unit{1}{\pascal} = \unit{1}{\newton\per\square\metre}  
= \unit{1}{\kilogram\per(\metre\usk\square\second)}
```

$$1 \text{ Pa} = 1 \text{ N/m}^2 = 1 \text{ kg}/(\text{m s}^2)$$

- ▶ Wiederholung aus Teil I: pmatrix-Umgebung: Matrix mit runden Klammern
- ▶ andere Möglichkeiten sind bmatrix ($[\cdot]$), Bmatrix ($\{ \cdot \}$), vmatrix ($| \cdot |$), Vmatrix ($\| \cdot \|$) und matrix (keine Klammern)
- ▶ bei mehr als zehn Spalten muss MaxMatrixCols mit `\setcounter{MaxMatrixCols}{Wert}` erhöht werden
- ▶ für kleine Matrizen (in Fließtext) kann die smallmatrix-Umgebung verwendet werden, Begrenzer müssen dann manuell gesetzt werden

Beispiel: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

vs. $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix} \right)$

Beispiel: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ vs. $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

AMS: Matrizen – Beispiele

```
\begin{equation*}
\begin{matrix} a & b \\ c & d \end{matrix} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix} \quad \left. \begin{matrix} a & b \\ c & d \end{matrix} \right\}
\end{equation*}
```

$$\begin{matrix} a & b \\ c & d \end{matrix} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{Bmatrix} a & b \\ c & d \end{Bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix} \quad \left. \begin{matrix} a & b \\ c & d \end{matrix} \right\}$$

- ▶ horizontale Punkte auf Linienhöhe: `\hdots (...)`
- ▶ horizontale Punkte auf Höhe des Malpunktes: `\cdots (\dots)`
- ▶ vertikale Punkte: `\vdots (:)`
- ▶ diagonale Punkte: `\ddots (\dots)`
- ▶ in Matrizen: Punkte über mehrere Spalten mit `\hdotsfor{Spalten}`

AMS: Punkte – Beispiel

```
\begin{equation}
\begin{pmatrix}
a & b & \hdots & c \\
d & \ddots & \ddots & \vdots \\
\hdotsfor{4} \\
e & f & \hdots & g
\end{pmatrix}
\end{equation}
```

$$\begin{pmatrix} a & b & \dots & c \\ d & \ddots & \ddots & \vdots \\ \dots & \dots & \dots & \dots \\ e & f & \dots & g \end{pmatrix} \quad (12)$$

Geschweifte Klammern über und unter Termen

- ▶ geschweifte Klammern über Termen: `\overbrace{Term}`
- ▶ geschweifte Klammern unter Termen: `\underbrace{Term}`
- ▶ mit `^` und `_` kann Text über bzw. unter die Klammer gesetzt werden
- ▶ beliebige (auch unsinnige) Kombinationen möglich

```
\begin{equation}
\overbrace{abc} \quad \underbrace{abc} \quad
\overbrace{abc}^{\{=5\}} \quad \underbrace{abc}_{\{=5\}} \quad
\overbrace{abc}_{\{=5\}} \quad \underbrace{abc}^{\{=5\}} \quad
\overbrace{\underbrace{abc}_{\{=5\}}^{\{=5\}}}
\end{equation}
```

$$\overbrace{abc} \quad \underbrace{abc} \quad \overbrace{abc}^{\{=5\}} \quad \underbrace{abc}_{\{=5\}} \quad \overbrace{abc}_{\{=5\}} \quad \underbrace{abc}^{\{=5\}} \quad \overbrace{\underbrace{abc}_{\{=5\}}^{\{=5\}}} \quad (13)$$

AMS: Text über und unter Symbole setzen

- ▶ mit `\overset{über}{Symbol}` wird „über“ über „Symbol“ gesetzt
- ▶ mit `\underset{unter}{Symbol}` wird „unter“ unter „Symbol“ gesetzt

```
\begin{equation}
\neg (a \wedge b)
\overset{\text{De Morgan}}{\equiv}
\neg a \vee \neg b
\end{equation}
```

$$\neg(a \wedge b) \overset{\text{De Morgan}}{\equiv} \neg a \vee \neg b \quad (14)$$

Integrale für Fortgeschrittene

- ▶ aus Teil I: Integrale mit $\int_a^b f(x) dx$
- ▶ Verbesserungsmöglichkeiten:
 - ▶ d sollte nicht kursiv gesetzt werden: d
 - ▶ Abstand zwischen \int und Formel verringern: $\!$
 - ▶ kleiner Abstand zwischen Formel und d : $\,$
 - ▶ Grenzen unter und über \int setzen: \limits

```
\begin{equation}
\int_a^b f(x) dx \quad
\int \limits_a^b \! f(x) \, \mathrm{d}x
\end{equation}
```

$$\int_a^b f(x) dx \quad \int_a^b f(x) dx \quad (15)$$

AMS: Mehrfach- und Ringintegrale

- ▶ Mehrfachintegrale: `\iint`, `\iiint` und `\idotsint`
- ▶ Ringintegrale: `\oint`

```
\begin{equation*}
\iint \limits_A \! f(x,y) \, \mathrm{d}x \, \mathrm{d}y \quad
\idotsint \limits_X \! f(x_1, \hdots, x_n) \, \mathrm{d}x_1 \, \hdots \, \mathrm{d}x_n \quad
\oint \! f(s) \, \mathrm{d}s
\end{equation*}
```

$$\iint_A f(x,y) \, dx \, dy \quad \int_X \cdots \int f(x_1, \dots, x_n) \, dx_1 \dots dx_n \quad \oint f(s) \, ds$$

Operatoren

- ▶ \LaTeX kennt verschiedene Typen von Operatoren
 - ▶ binäre Operatoren: $+$ (+), \vee (\vee)
 - ▶ große Operatoren: \sum (Σ), \prod (Π)
 - ▶ benannte Operatoren: \sin (sin), \lim (lim)
- ▶ bei manchen Operatoren werden Indizes/Exponenten unter/über den Operator gesetzt (große Operatoren, manche benannte Operatoren)
- ▶ im Inline-Modus $\$ \dots \$$ werden große Operatoren kleiner dargestellt und Indizes/Exponenten hinter den Operator gesetzt
- ▶ abweichende Darstellung kann erzwungen werden:
 $\backslash limits$, $\backslash nolimits$, $\{\backslash displaystyle \dots\}$,
 $\{\backslash textstyle \dots\}$ (Positionen der geschweiften Klammern beachten!)

Operatoren – Beispiel I

Eine Summe im Inline-Modus: \sum_a^b .

Indizes/Exponenten unter/über das Summenzeichen:

$\sum\limits_a^b$.

Display-Anzeige erzwingen:

$\{\displaystyle\sum_a^b\}$.

Eine Summe im Inline-Modus: \sum_a^b . Indizes/Exponenten

unter/über das Summenzeichen: \sum_a^b . Display-Anzeige

erzwingen: \sum_a^b .

Operatoren – Beispiel II – Dualer Fall

```
\begin{equation}
\sum_a^b \quad \quad
\sum\nolimits_a^b \quad \quad
{\textstyle\sum_a^b}
\end{equation}
```

$$\sum_a^b \quad \sum_a^b \quad \sum_a^b \quad (16)$$

AMS: Eigene Operatoren definieren

- ▶ `\DeclareMathOperator{\Befehl}{Text}` definiert einen neuen Operator
- ▶ im Header (vor `\begin{document}`)
- ▶ aber: erst `amsmath` einbinden
- ▶ gesternzte Form (`\DeclareMathOperator*{...}{...}`) definiert einen Operator, bei dem Indizes/Operatoren unter/über den Operator gesetzt werden

AMS: Eigene Operatoren definieren – Beispiel

```
\usepackage{amsmath}
...
\DeclareMathOperator{\ITE}{ITE}
\DeclareMathOperator*\{staring}\{bigstar}
...
\begin{document}
...
\begin{equation}
\ITE_{a,b,c}(a, b, c) = a \wedge b \vee \neg a \wedge c
\quad
\staring_{i=1}^n x_i
\end{equation}
```

$$\text{ITE}_{a,b,c}(a, b, c) = a \wedge b \vee \neg a \wedge c \quad \bigstar_{i=1}^n x_i \quad (17)$$

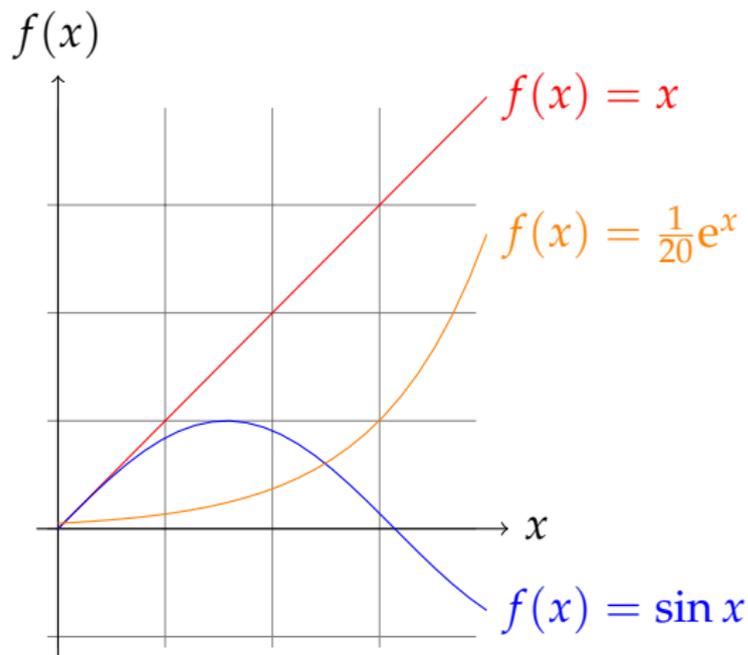
Funktionsgraphen mit TikZ

- ▶ aus Grafik: mit TikZ können Grafiken in \LaTeX „programmiert“ werden
- ▶ TikZ hat viele Anwendungen in der Mathematik (Venn-Diagramme, Geometrie, ...)
- ▶ hier nur betrachtet: Funktionen plotten

Vorteile:

- ▶ nahtlose Integration in \LaTeX -Dokumente, keine externen Dateien
- ▶ sehr gute Anpassungsmöglichkeiten (Farben, Gitter, Beschriftung)
- ▶ Plotten von externen Daten auch möglich

Funktionsgraphen mit TikZ – Beispiel



(Beispiel aus pgfmanual.pdf, Version 2.10, Abschnitt 19.5)

Funktionsgraphen mit TikZ – Beispiel – Quellcode

```
\begin{tikzpicture}[domain=0:4]
  \draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
  \draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
  \draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
  \draw[color=red] plot (\x,\x)
    node[right] {$f(x) = x$};
  \draw[color=blue] plot (\x,{sin(\x r)})
    node[right] {$f(x) = \sin x$};
  \draw[color=orange] plot (\x,{0.05*exp(\x)})
    node[right] {$f(x) = \frac{1}{20} \mathrm{e}^x$};
\end{tikzpicture}
```

(Beispiel aus pgfmanual.pdf, Version 2.10, Abschnitt 19.5)

Funktionsgraphen mit gnuplot

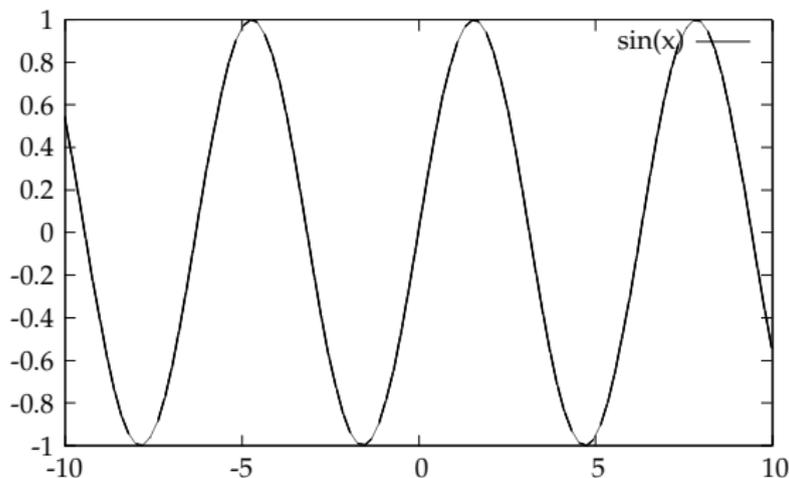
- ▶ gnuplot ist ein eigenständiges Plotprogramm
- ▶ <http://gnuplot.sourceforge.net/>
- ▶ L^AT_EX-Integration durch das gnuplottex-Paket
- ▶ TikZ kann zum Plotten auch auf gnuplot zurückgreifen
- ▶ latex muss mit `-shell-escape` aufgerufen werden (Sicherheitsmaßnahmen werden umgangen)

Vorteile

- ▶ kann auch 3D-Plots
- ▶ fortgeschrittene Funktionen wie Fitting
- ▶ Vorteile bei mathematisch anspruchsvollen Aufgaben (L^AT_EX ist kein Rechenprogramm)

Funktionsgraphen mit gnuplot – Beispiel

```
\usepackage{gnuplottex}  
...  
\begin{gnuplot}[scale=0.6]  
plot sin(x)  
\end{gnuplot}
```



Theorem-Umgebungen – zuerst ein Beispiel

Axiom 1 (Peano-Axiome).

1. 0 ist eine natürliche Zahl
2. zu jeder natürlichen Zahl n gibt es einen Nachfolger $n' = n + 1$, der eine natürliche Zahl ist
3. zu jedem n' gibt es maximal ein n
4. \mathbb{N} ist die kleinste Menge, die 0 und mit jedem n auch n' enthält

Satz 1. Es gibt unendlich viele natürliche Zahlen.

Beweis. Folgt aus Axiom 1. □

Beispiel 0.1. 1, 2, 3 und 3780312 sind natürliche Zahlen.

Theorem-Umgebungen

- ▶ Sätze, Beispiele, Beweise, Bemerkungen, etc. werden in \LaTeX allgemein als Theoreme bezeichnet
- ▶ viele Pakete mit vordefinierten Theoremumgebungen und/oder der Möglichkeit, selbst welche zu definieren existieren
- ▶ Beispiele: `theorem`, `amsthm`, `ntheorem`
- ▶ \LaTeX unterstützt auch von Haus aus Theoreme
- ▶ grundsätzlich sind alle Pakete ähnlich:
 - ▶ Theoremumgebungen werden bereitgestellt
 - ▶ Theoreme sind vom restlichen Text abgesetzt
 - ▶ Nummerierung von Theoremen
- ▶ Unterschiede: Anpassbarkeit, vordefinierte Umgebungen, Endsymbole (bei Beweisen)

- ▶ Einbinden mit `\usepackage{amsthm}`
- ▶ bei gleichzeitiger Verwendung mit `amsmath` erst `amsmath` laden
- ▶ Definition von Theorem-Umgebungen mit `\newtheorem{Name}[nName]{Markierung}[Nummerierung]`
 - ▶ Name: Name der neuen Umgebung (z. B. lemma)
 - ▶ nName: Nummerierung wie nName (z. B. satz)
 - ▶ Markierung: Markierung der Umgebung im Dokument (z. B. Lemma)
 - ▶ Nummerierung: Nummerierung nach welchem übergeordneten Element (z. B. section)
 - ▶ nName und Nummerierung sind optional
- ▶ `proof` ist schon vordefiniert, am Ende wird ein \square gesetzt
- ▶ funktioniert manchmal nicht – dann kann `\qedhere` verwendet werden

- ▶ Theorem-Stile legen das Aussehen der Umgebung fest
- ▶ vordefinierte Stile: `plain` (Standard, Kursivschrift), `definition`, `remark` (Nicht-Kursivschrift)
- ▶ Abweichungen je nach Dokumentklasse möglich
- ▶ werden vor dem `\newtheorem`-Aufruf mit `\theoremstyle{...}` ausgewählt
- ▶ gelten für alle danach definierten Umgebungen

amsthm – Beispiel

```
\usepackage{amsmath} % ggf. zuerst laden!  
\usepackage{amsthm}  
  
\theoremstyle{plain} % nicht nötig, da Standard  
\newtheorem{axiom}{Axiom}  
\newtheorem{theorem}{Satz}  
\newtheorem{lemma}[theorem]{Lemma}  
\theoremstyle{definition}  
\newtheorem{example}{Beispiel}[section]
```

- ▶ es werden die Umgebungen axiom, lemma und example definiert
- ▶ Lemmata und Sätze werden mit einem Zähler nummeriert
- ▶ Beispiele werden nach Abschnitten nummeriert

amsthm – Verwendung

```
\begin{axiom}[Peano-Axiome]\label{axm:peano} ~\\
\begin{enumerate}
\item 0 ist eine natürliche Zahl
...
\end{enumerate}
\end{axiom}
\begin{theorem}
Es gibt unendlich viele natürliche Zahlen.
\end{theorem}
\begin{proof}
Folgt aus Axiom \ref{axm:peano}.
\end{proof}
\begin{example}
1, 2, 3 und 3780312 sind natürliche Zahlen.
\end{example}
```