

# Einführung in das wissenschaftliche Arbeiten

## Einheit IV: LaTeX - Teil 2

Dr. Björn Eichmann ([eiche@tp4.rub.de](mailto:eiche@tp4.rub.de))

Bochum, 27. März 2019

# Mathemodus: Gleichungen & Formeln

$$\delta \left( t - \frac{m_p c}{3 a_{ad}} \ln \left( \frac{a_{cc} + a_{ad} p_0^3}{a_{cc} + a_{ad} p^3} \right) \right) = \frac{\delta(p_0 - p_0^{\text{zero}}(t))}{\left| \frac{\partial}{\partial p_0} \left( t - \frac{m_p c}{3 a_{ad}} \ln \left( \frac{a_{cc} + a_{ad} p_0^3}{a_{cc} + a_{ad} p^3} \right) \right) \right|_{p_0 = p_0^{\text{zero}}(t)}}$$

$$= \frac{(a_{cc} + a_{ad} (p_0^{\text{zero}}(p, t))^3)}{m_p c \cdot (p_0^{\text{zero}}(p, t))^2} \cdot \delta(p_0 - p_0^{\text{zero}}(p, t)). \quad (7.68)$$

# Mathemodus

In LaTeX werden Formeln, Gleichungen und Ausdrücke im **Mathemodus** eingegeben.

- Der Mathemodus kann auf **verschiedene Arten** aufgerufen werden *im Fließtext, abgesetzt, eingerückt, zentriert, nummeriert, ...* um nicht-Inline-Matheumgebungen *linksbündig einzurücken statt zu zentrieren: amsmath-Paket mit optionalem Argument **fleqn**: `\usepackage[fleqn]{amsmath}`*
- Die Bereiche, in denen Mathemodus geschrieben werden kann, heißen **Matheumgebung**
- Für Matheumgebungen gibt es *Befehle, die außerhalb nicht funktionieren: Sub- und Superskripte, Symbole, ...*

# Mathemodus

Verschiedene Matheumgebungs-Aufrufe:

- **Zeilenmodus (Inline)**  $(a^2+b^2=c^2)$  im Fließtext, nicht-nummeriert, alternativ:  
`\begin{math}`  
`a^2+b^2=c^2`  
`\end{math}`
- **Robuster Zeilenmodus:**  $\$a^2+b^2=c^2\$$  funktioniert auch in Überschriften u.Ä.

**Meist ist  $\$...\$$  die beste Wahl** für den Zeilenmodus, aber in einer `\alltt`-Umgebung (darin werden Befehle nicht interpretiert, sondern als Text ausgegeben) funktioniert dies nicht. Dort sollte `\(...\)` verwendet werden.

Der Output sieht bei all diesen drei Varianten identisch aus.

**Achtung:** Inline-Matheumgebungen haben Auswirkungen auf die Zeilenabstände!

# Mathemodus

## Verschiedene Matheumgebungs-Aufrufe:

- **Abgesetzter Modus** (`displaymath`) `\[a^2+b^2=c^2\]` einzeilig, nicht-nummeriert, abgesetzt, alternativ: `\begin{displaymath}...\end{displaymath}`  
`$$a^2+b^2=c^2$$` vermeiden!
- Ebenfalls möglich:  
`\begin{equation}`  
`a^2+b^2=c^2`  
`\end{equation}`  
liefert allerdings eine **nummerierte Gleichung**. Abhilfe:  
a) \* nach `equation` (`amsmath`-Paket einbinden) oder  
b) `\nonumber` vor `\end{equation}`

Vorsicht: **Keine leeren Zeilen vor und nach abgesetzten Formeln im Quellcode!**  
Diese können zu ungleichen vertikalen Abständen führen.

# Mathemodus

## Verschiedene Matheumgebungs-Aufrufe:

- **Mehrzeiliger abgesetzter Modus:** eqnarray

```
\begin{eqnarray}
```

```
  a & b & c \\\
```

```
  ...
```

```
  x & y & z \\\
```

```
\end{eqnarray}
```

eventuell problematisch: immer 3-spaltige Matrix mit Ausrichtung rcl,

**jede Zeile nummeriert**

- **Besser: array in equation** (beliebige Spaltenzahl, **nur eine Nummer**, wie tabular)

```
\begin{equation}
```

```
  \begin{array}{rccl}
```

```
    a & b & c & d \\\
```

```
  ...
```

```
    w & x & y & z \\\
```

```
  \end{array}
```

```
\end{equation}
```

# Mathemodus

## Verschiedene Matheumgebungs-Aufrufe:

- *Noch besser:* **align** (meine Empfehlung für mehrzeilige Gleichungen)

```
\begin{align}
  a &= b + c &= d \label {abc} \\
  e &= f - g &= h \ nonumber \\
  i &= j \cdot k &= l \\
\end{align}
```

Symbole/Buchstaben/Zahlen unmittelbar nach dem & werden untereinander angeordnet, beliebig viele & pro Zeile möglich (solange in jeder Zeile gleich viele), Spalten immer im Wechsel rechts- und linksbündig

- *Nummerierung bei align:* **Standardmäßig in jeder Zeile**, kann komplett abgestellt werden (durch \* nach align) oder zeilenweise (**\nonumber**)  
(im Bsp.: erste Zeile wird nummeriert und kann referenziert werden, die zweite erhält keine Nummer, die dritte erhält eine, kann aber nicht referenziert werden)
- **Leerzeilen in der align-Umgebung führen zu einer Fehlermeldung**

# Mathemodus

Ein paar typische Matheumgebungs-Befehle:

- `\frac{a}{b}` Bruch  $a/b$
- `_a` und `^b` Subskript (Index)  $a$  und Superskript (Exponent)  $b$ , können an andere Symbole in beliebiger Reihenfolge angehängt werden
- `\int_a^b x^2 \, \mathrm{d}x`
- `\overset{!}{=}` ergibt das Symbol für „wird gleich gesetzt mit/gefordert“, man kann mit `overset` aber auch andere Symbole übereinander setzen
- `\underbrace{abc}_{\text{Teil 1}}` geschweifte Klammer unter „abc“, beschriftet mit „Teil 1“
- `\overbrace{def}^{\text{Teil 2}}` geschweifte Klammer über „def“, beschriftet mit „Teil 2“

# Mathemodus

Ein paar typische Matheumgebungs-Befehle:

- `\cdot` Malpunkt, z.B. für Skalarprodukt
- `f(x) =`  
`\begin{cases}`  
`2x^2 & \text{für } x \text{ less 4}`  
`2x^3 + 4^2 & \text{für } 4 \le x \text{ less 27}`  
`3x^2 \cdot \sin(x) & \text{für } x \ge 27`  
`\end{cases}` Fallunterscheidung einer abschnittsweise definierten Funktion
- `\mathrm{a}`, ... wie `\text{a}`, aber in Matheschriftart; `\mathbb{R}` erzeugt  $\mathbb{R}$
- `\mathfrak{G}` Frakturbuchstaben in Matheschriftart

# Mathemodus

Ein paar typische Matheumgebungs-Befehle:

- `\left( ... \right)`, `\left[ ... \right]` dynamisch angepasste Klammern, Größe passt sich dem Inhalt ... an (Klammerarten dürfen gemischt werden)  
Achtung: In einem solchen Paar Klammern sind **Zeilenumbrüche** nicht ohne Weiteres zu realisieren
- `\text{Beschreibung}` gibt Text „Beschreibung“ in normalem Schriftbild aus
- `\cdots` Auslassungspunkte, z.B. in Mengenklammern
- `\Delta`, `\nabla` Laplace- und Nabla-Operator

# Mathemodus

Ein paar typische Matheumgebungs-Befehle:

- `\big(`, `\Big(`, `\bigg(`, `\Bigg(` manuell vergrößerte Klammern (analog auch eckige Klammern sowie rechte Klammern)
- `\bigintssss`, `\bigintsss`, `\bigintss`, `\bigints`, `\bigint` größere Integralzeichen (Paket `bigints` laden)
- `\alpha`, `\Alpha`, ... erzeugt griechische Buchstaben
- `\vec{r}` erzeugt einen Vektorpfeil über  $r$
- `\partial` Symbol für partielle Ableitungen
- `\;`, `\:`, `\!` Leerstellen in sehr groß, groß, mittel, klein, negativ (z.B.:  $a \backslash b$  oder  $a \; b$ )

# Mathemodus

Ein paar typische Matheumgebungs-Befehle:

- `\rightarrow`, `\Rightarrow`, `\leftrightharrow`, `\LeftRightarrow` erzeugt unterschiedliche Pfeile (Reaktionspfeil, Folgepfeil, doppelseitiger Pfeil, Äquivalenzsymbol), funktioniert auch mit anderen Richtungen
- `\iint`, `\iiint` Doppel- bzw. Dreifachintegral
- `\oint`, `\oiint` erzeugt ein geschlossenes Kurvenintegral bzw. Flächenintegral
- `\sum_{i=1}^n` erzeugt eine Summe mit Index  $i=1..n$
- `\prod_{i=1}^n` erzeugt ein Produkt mit Index  $i=1..n$
- `\sqrt[n]{x}` erzeugt die n-te Wurzel aus x

# Mathemodus

Ein paar typische Matheumgebungs-Befehle:

- **Vektoren:**

```
\vec{r} = \left(\begin{array}{c}1\\2\\3\end{array}\right)
```

$$\vec{r} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

auf Dauer umständliche Schreibweise

Tipp: Neuen, **kürzeren Befehl definieren** (in Präambel, später mehr)

```
\newcommand{\vect}[1]{\ensuremath{\left(\begin{array}{c}#1\end{array}\right)}}
```

dank **\ensuremath** auch außerhalb von Matheumgebungen verwendbar

```
\vect{1\\2\\3}
```

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

# Mathemodus

Ein paar typische Matheumgebungs-Befehle:

- **Matrizen:** Befehl wie array (Standard: alle Spalten zentriert)

```
\left(\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}\right)
```

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

*matrix* allein erzeugt keine Klammern. **Varianten:**

- pmatrix Matrix in ( )
- bmatrix Matrix in [ ]
- Bmatrix Matrix in { }
- vmatrix Matrix in | |
- Vmatrix Matrix in || ||

Ein \* nach Xmatrix erlaubt manuelle Textausrichtung, Bsp.: `\begin{bmatrix*}[lcr]`

# Farbe

Colors make this text 20% cooler in 10 seconds flat.

# Farbe

LaTeX unterstützt auch farbige Ausgabe, zum Beispiel mit dem **Paket xcolor**.

```
\color{red} C^2
```

```
\color{yellow} A = {\color{blue}B} \mathbin{{\color{red}-}} {\color{green}C}
```

`\mathbin` sagt LaTeX, dass „-“ ein Minus sein soll, kein Bindestrich (wichtig für korrekte Abstände darum)

- `\color` ist ein **Schalter** mit dem Argument `<Farbe>`
- Mögliche Farben: white, black, red, green, blue, cyan, magenta, yellow  
andere Farben können manuell definiert werden
- `\color` funktioniert **innerhalb und außerhalb** von Matheumgebungen

# Eigene Befehle definieren

```
\newcommand{\vect}[1]{\ensuremath{\left( \begin{array}{c}#1\end{array}\right)}}
```

# Eigene Befehle definieren

LaTeX ermöglicht die Definition eigener (oder auch Neudefinition bestehender) Befehle

- **`\newcommand{\neu}[nargs]{abc}`**

- ist die Syntax für die Definition des neuen Befehls „\neu“
  - dabei gibt „nargs“ die **Anzahl der Argumente** an
  - diese können in den Anweisungen, die durch Aufrufen von „\neu“ ausgeführt werden, mit **#1, #2, ...** referenziert werden
- Dieser Befehl „\neu“ gibt den Text „abc“ aus.

- Beispiel: Definition eines Befehls zur bequemen Referenzierung von Gleichungen

```
\newcommand{\glref}[1]{Gl.~(\ref{#1})},
```

```
Aufruf: \glref{label_7}
```

dabei wird der bestehende Befehl `\ref{}` benutzt.

Ausgabe: „Gl. (7)“ (wenn das label „label\_7“ auf Gleichung 7 verweist)

# Eigene Befehle definieren

LaTeX ermöglicht die Definition eigener (oder auch Neudefinition bestehender) Befehle

- **`\newcommand{\neu}[nargs]{abc}`**
  - ist die Syntax für die Definition des neuen Befehls „`\neu`“
  - dabei gibt „`nargs`“ die Anzahl der Argumente an
  - diese können in den Anweisungen, die durch Aufrufen von „`\neu`“ ausgeführt werden, mit `#1`, `#2`, ... referenziert werdenDieser Befehl „`\neu`“ gibt den Text „`abc`“ aus.
- Beispiel: Definition eines Befehls zur einfacheren Eingabe von nicht-kursiven Differential-d's im Mathemodus:

```
\newcommand{\diff}{\ensuremath{\mathop{\!}\!\mathrm{d}}}
```

**Übung:** Definition eines Befehls zur Eingabe solcher d's mit beliebiger Potenz

# Eigene Befehle definieren

LaTeX ermöglicht die Definition eigener (oder auch Neudefinition bestehender) Befehle

- Nach **Definition eines Befehls in der Präambel** kann er im Dokument verwendet werden
- Versucht man einen bestehenden Befehl zu definieren, gibt LaTeX eine **Fehlermeldung** aus
- Will man einen *bestehenden Befehl anders definieren*, funktioniert dies mit **\renewcommand**  
die Syntax ist die selbe wie bei `\newcommand`  
bestehende Befehle **nur in Ausnahmefällen umdefinieren!**

# Typische Schwierigkeiten & Kniffe

- LaTeX trennt **Worte** nicht dort, wo es sie **trennen** soll  
**Sprache für Trennung richtig eingestellt** (babel)?  
Manuelle Sil\ben\tren\nung vorgeben (zeigt LaTeX, wo getrennt werden darf)
- LaTeX trennt Worte oder **bricht** zwischen Zahl und Einheit  
die Zeile **um**  
Trick: **unsichtbare Box** um `\mbox{Wort}` oder `\mbox{Zahl&Einheit}` legen  
Alternativ bei Zahl und Einheit: **Tilde** im Mathemodus dazwischen setzen  
(verhindert Trennung, erzeugt aber mit unter eine zu große Leerstelle, ich bevorzuge `\mbox{3\,cm}`)

# Typische Schwierigkeiten & Kniffe

- LaTeX erzeugt **zu große Leerstellen** nach Abkürzungen im Satz (1,5-fache Größe)

Im Textmodus interpretiert LaTeX eine Leerstelle nach einem **Punkt als Satzende** und lässt standardmäßig eine 1,5-fache Leerstelle zwischen Sätzen.

Um das zu unterbinden, einen `\` unmittelbar nach dem Punkt setzen: z.B. `\,` `ggf.\,` ...

- Ich habe keine Ahnung, wie ich **ein bestimmtes Symbol in LaTeX erzeugen** kann, und ich kenne auch den Namen des Symbols nicht

Trick: Detexify (<http://detexify.kirelabs.org/classify.html>) besitzt ein Feld, in das man mit der Maus das Symbol zeichnen kann, und schlägt danach mehrere LaTeX-Befehle vor, die dieses Symbol (oder ähnliche) erzeugt

# Typische Schwierigkeiten & Kniffe

- Ich habe **Abbildungen in verschiedenen Formaten**, die LaTeX nicht im selben Dokument verarbeiten kann

Sind alle Abbildungen **Vektorgrafiken**?

Falls die Abbildungen selbst erstellt sind: Kann man sie **in einem anderen Format erzeugen**?

*Unter Linux* einfach **umwandelbar**: „**convert bild.eps bild.pdf**“ erzeugt aus bild.eps eine pdf-Version

*Unter Windows* u.U. in der Kommandozeile ebenfalls möglich, alternativ: in einem **Grafikprogramm** umwandeln.

- Der **spellcheck** meines Editors markiert mir **LaTeX-Befehle als Fehler**

Eventuell muss in ihrem Editor erst die **Syntax-Erkennung** eingeschaltet werden. Typischerweise gilt: Kann ihr Editor die Syntax hervorheben, so kann er sie auch im spellcheck erkennen

# Typische Schwierigkeiten & Kniffe

- LaTeX erzeugt in meinem **deutschen Dokument englische Überschriften** wie Bibliography statt Literaturverzeichnis  
Ist das **babel Paket mit ngerman** korrekt eingebunden?  
→ `\usepackage[ngerman]{babel}` in der Präambel  
*Bei zwei erlaubten Sprachen ist die letzte hier gelistete die Hauptsprache, in der die Überschriften erzeugt werden!*
- LaTeX meldet einen Fehler, dessen **Ursache ich nicht lokalisieren kann**  
LaTeX-Fehlermeldungen sind zwar oft nützlich, manchmal aber nicht einmal mit einer Zeilenangabe versehen.  
Grundsätzlich: **Fehler von vorne nach hinten abarbeiten!**  
→ Empfehlung: Nicht nur 1 tex-Datei, sondern mehrere (s. nächster Abschnitt)

# Typische Schwierigkeiten & Kniffe

- Mein fertiges pdf hat eine **enorme Dateigröße**. Kann ich die reduzieren?

Ist z.B. bei Bewerbungen im pdf-Format wichtig, da oft eine Grenze für die Dateigröße vorgegeben ist, erlaubt aber auch, die Arbeit im pdf-Format schneller an Bekannte zu verschicken.

→ *unter Linux*: im Terminal in das Verzeichnis mit der pdf-Datei navigieren, dort  
`„gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dPDFSETTINGS=/screen -dNOPAUSE -dQUIET -dBATCH -sOutputFile=output.pdf input.pdf“`

Optionen für dPDFSETTINGS in absteigender Auflösung der Abbildungen:  
`/screen, /ebook, /printer, /prepress`

Achtung: Dabei geht natürlich Information verloren. Das geschieht typischerweise bei den Abbildungen, sofern sie nicht als Vektorgrafiken vorliegen. Die Datei ist danach nicht für den finalen Druck der Arbeit geeignet, da die Abbildungen teilweise deutlich schlechter aufgelöst sind!

# Typische Schwierigkeiten & Kniffe

- Kann ich **mehrere pdf-Dokumente aneinanderhängen**, sodass ein großes pdf-Dokument entsteht?  
Ist z.B. praktisch, wenn eine Erklärung über das eigenständige Verfassen der Arbeit oder ein Lebenslauf an die Arbeit angehängt werden soll.

→ *unter Linux*: im Terminal in das Verzeichnis mit den pdf-Dateien navigieren, dort „`pdftk seite1.pdf seite2.pdf seite3.pdf cat output zieldatei.pdf`“

Dabei stehen `seite1.pdf` usw. für die einzelnen Dokumente, die zusammengefügt werden sollen, `zieldatei.pdf` ist der Name der erzeugten großen Datei.

*Noch unter LaTeX*: mit dem Paket `pdfpages`.

Alle Seite einbinden mit `\includepdf[pages=-]{Name der Datei}`

Meist ist es dennoch sinnvoller, sämtliche Inhalte und Zusätze der Arbeit noch in LaTeX ins Dokument einzubinden.

# Mehrere Dokumente



[http://upload.wikimedia.org/wikipedia/commons/0/03/Istituto\\_agronomico\\_per\\_l%27oltremare,\\_int.,\\_biblioteca,\\_schedario\\_05.JPG](http://upload.wikimedia.org/wikipedia/commons/0/03/Istituto_agronomico_per_l%27oltremare,_int.,_biblioteca,_schedario_05.JPG)

# Mehrere Dokumente

Seine Arbeit in mehrere Dateien zu teilen, hat mehrere *Vorteile*

- **Übersichtlichkeit**  
kein stundenlanges Scrollen
- Einfachere **Fehlersuche**  
LaTeX-Fehlermeldungen gibt i.d.R. die Datei an, in der das Problem auftritt
- Eine Datei pro Kapitel erlaubt unkompliziertes **Ändern der Reihenfolge** von Kapiteln, auch später  
nur in anderer Reihenfolge einbinden

# Mehrere Dokumente

*Empfehlung:*

- 1 Header-Datei  
enthält die Dokumentklasse und Präambel (damit auch alle Pakete und Befehls-Definitionen)
- 1 Datei pro Kapitel  
das schließt auch die Titelseite/das Deckblatt ein, darin kann sofort der Inhalt getippt werden, z.B. `\chapter{Einleitung \label{einleitung}}`
- 1 Hauptdatei  
hier steht “`\begin{document}` und `\end{document}`”, dazwischen u.A. die Befehle zum Einbinden der einzelnen Dateien, z.B. `\input{Kapitel_Einleitung}`
- 1 bib-Datei  
enthält alle Literaturangaben, daraus wird das Literaturverzeichnis generiert

# Warnungen, Fehler, bad boxes

LaTeX-Quellcode wird kompiliert, um ein Dokument zu erzeugen. Dabei können Warnungen, Fehler, und bad boxes auftreten

- **Warnung:** Hinweis des Compilers, dass etwas nicht in Ordnung ist/sein könnte, aber *nicht das Kompilieren und Erzeugen des Dokuments verhindert*  
z.B. Hinweis auf nicht erkannte Label/Referenzen, doppelte Vergabe einer Seitennummer/eines Labels, ...
- **Fehler:** Hinweis des Compilers, dass etwas nicht in Ordnung ist und das *Kompilieren und Erzeugen des Dokuments oftmals verhindert*  
z.B. nicht-kompilierbarer Quellcode (oft durch falsche Klammersetzung oder Syntax), fehlende Dateien/Pakete, unbekannte Befehle (Tippfehler, Paket nicht eingebunden), ggf. bereits geöffnetes pdf welches nicht überschrieben werden darf (fatal error)

# Warnungen, Fehler, bad boxes

LaTeX-Quellcode wird kompiliert, um ein Dokument zu erzeugen. Dabei können Warnungen, Fehler, und bad boxes auftreten

- **Bad box:** Hinweis des Compilers, dass etwas nicht den Einstellungen entsprechend **platziert** werden kann bzw. **ungewollte Abstände** im Text ergibt oder gar über die Seite hinausragt  
z.B. Hinweis auf Formel, die nicht in eine Zeile passt, Hinweis auf größeren Abstand zwischen zwei Absätzen als gewünscht, ...  
wird immer **mit einer Zahl angegeben** (badness 1-10 000), die angibt, als wie schlimm LaTeX diesen Umstand findet oder die Zahl an pt, um die der Text aus dem Dokument herausragt (24.7 pt too wide)

Oft kann LaTeX auch (zumindest grob) angeben, wo im Dokument das Problem auftritt.

# Warnungen, Fehler, bad boxes

## Umgang mit Fehlern:

- Aus offensichtlichen Gründen muss man sich grundsätzlich **sofort um Fehler kümmern**  
sonst wird das Dokument nicht kompiliert und man kann den anderen Meldungen nicht nachgehen
- Bei mehreren Fehlern immer das Dokument **von vorn nach hinten** durchgehen  
vielleicht sind spätere Fehler durch vorige verursacht (z.B. bei falscher Kammersetzung oder fehlenden Paketen)
- Bei unbekanntem Befehlen: Sind die **zugehörigen Pakete in der Präambel eingebunden**? Sind die Befehle vorher definiert worden?  
Oder ist ein **Tippfehler** schuld an der Fehlermeldung?

# Warnungen, Fehler, bad boxes

## Umgang mit Warnungen:

- Zwar muss man sich nicht grundsätzlich sofort um Warnungen kümmern, aber *meist ist es doch sinnvoll*  
aus den meisten Warnungen kann man auch vergleichsweise schnell schlau werden
- Bei mehreren Warnungen immer das Dokument von vorn nach hinten durchgehen  
vielleicht sind spätere Warnungen durch vorige verursacht (z.B. bei falscher Kammersetzung oder fehlerhaften Paket-Parametern)
- **Bei unbekanntem Labeln/Referenzen:** Aufgrund der Reihenfolge beim Kompilieren eines LaTeX-Dokuments funktionieren Referenzen auf Label oder Literatur **erst beim zweiten Kompilieren**  
→ **mehrfach Kompilieren, mindestens zweimal**

# Warnungen, Fehler, bad boxes

## Umgang mit bad boxes:

- *Manche kann und sollte man sofort angehen* (z.B. zu lange Formeln), andere erst beim finalen Formatieren des Dokuments  
sonst wird man ggf. einen zu großen Freiraum los, um sich später dafür einen anderen einzuhandeln
- Nach Möglichkeit die betroffene Stelle im Dokument anschauen, um **selbst zu entscheiden**, ob es schlimm aussieht  
vielleicht ist man selbst toleranter als LaTeX
- Typisch: overfull hbox oder underfull hbox  
**overfull:** LaTeX konnte den Text an einer Stelle nicht auf die Breite des Dokuments reduzieren, etwas ragt (vermutlich rechts) über den Rand  
→ Lösung: z.B. **manueller Zeilenumbruch** in Umgebung oder `Wort\tren\nung`  
**underfull:** LaTeX war gezwungen, einen Freiraum so groß zu wählen, dass er die Dokumentvorgabe übersteigt (oft: Zeilenumbruch `\` in leerer Zeile das Problem)  
→ Lösung: in solchen Fällen `\vspace{xy pt}` statt `\` verwenden (oder gar nichts)

# Warnungen, Fehler, bad boxes

## Umgang mit bad boxes:

- LaTeX **meldet immer hbox**, ganz egal, ob es sich um eine hbox oder vbox oder sonst eine box handelt  
Ursache: Meldung wird durch TeX erzeugt, nicht durch LaTeX

- Zusätzliche Angaben:

detected at line <>: manuell erzeugte Box, endet in Zeile <> reduzieren, etwas ragt (vermutlich rechts) über den Rand

has occurred while \output is active: Problem trat auf, als eine Überschrift o.Ä. gesetzt wurde, daher keine Zeilenangabe; da hilft nur Suchen

in alignment at lines <>: Problem in ausgerichteter Formel- oder tabular-Umgebung, Zeilennummern geben Position der Umgebung an

in paragraph at lines <>: unschöne Zeile in einem Absatz, recht leicht zu finden, oft durch Verwendung von manuellem Zeilenumbruch `\` erzeugt

# Warnungen, Fehler, bad boxes

*Grundsätzlich:*

Man ist praktisch **nie der Erste**, der vor diesem Problem steht  
es gibt viele Plattformen, auf denen man Hinweise oder Hilfe findet

*Nützliche Anlaufstellen:*

- [www.google.de](http://www.google.de) bedarf wohl keiner Erklärung
- [www.ctan.org](http://www.ctan.org) und [www.dante.de](http://www.dante.de) : Englisch- und Deutschsprachige Uservereinigungen
- <http://de.wikibooks.org/wiki/LaTeX-Kompendium>: Praktische Sammlung an Tipps, Tricks, auch als Tutorial zu empfehlen (deutsch und englisch nicht zwingend deckungsgleich)
- [www.golatex.de](http://www.golatex.de) großes, deutsches LaTeX-Forum
- LaTeX-Einführungen von Manuela Jürgens, FernUni Hagen, am Beispiel des Buches “Der kleine Hobbit”: [http://www.wiwiss.fu-berlin.de/fachbereich/vwl/iso/links/latex\\_einfuehrung\\_manuela\\_juergens.pdf?1353072163](http://www.wiwiss.fu-berlin.de/fachbereich/vwl/iso/links/latex_einfuehrung_manuela_juergens.pdf?1353072163) , <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf>

# Ausblick

Nächste Vorlesung:

Zitieren mit LaTeX: bibtex und biblatex