Einführung in LaTeX

Im Rahmen dieser Einführung sollen die grundlegenden Funktionen von LaTeX erlernt werde. Deshalb empfehlen wir, dass du ebenfalls ein ein Dokument erstellst und bestenfalls die Befehle händisch abtippst und nicht nur per „Copy&Paste“ einfügst.

Sowohl Overleaf, als auch die üblichen grafischen Oberflächen sind sehr ähnlich aufgebaut. Die linke Seite ist dafür vorgesehen um „LaTeXcode“ zu schreiben und rechts wird das kompilierte Dokument angezeigt. Meist befindet sich oben in der Menüzeile ein Knopf mit „Aktualisieren“, „Kompilieren“ oder „Erstellen“, welcher die Vorschau des getippten Textes erzeugt.

Prinzipell können auch mit einer lokalen grafischen Oberfläche wie TexStudios, oder TeXmaker LaTeX-Dokumente erstellt werden, trotzdem empfehlen wir die Verwendung von Overleaf, da hier viele Stolpersteine für Anfänger aus dem Weg geräumt sind.

Diese Einführung ist so aufgebaut, dass LaTeXcode als Textblock eingefügt wird und darunter wird die kompilierte Ausgabe präsentiert.

Inhaltsverzeichnis

1 Aufbau

Ein LaTeX-Dokument ist immer folgendermaßen aufgebaut

Präampel
```
\documentclass{...}
\usepackage{...}
...
```

Inhalt
```
\begin{document}
      Text & Bilder...
\end{document}
```

Oben im Dokument befindet sich die sog. Präambel. Durch den Befehl `\documentclass{...}` wird die Dokumentenklasse festgelegt. Hier sind sämtliche Formatdetails, wie Schriftgröße, Seitenränder usw. festgelegt.

Darauf folgt in der Regel eine Reihe an `\usepackage{...}` aufrufen. Hier werden zusätzliche Pakete geladen, welche mehr Funktionalität, wie z.B. den Mathemodus ermöglichen.

Außerdem finden noch viele andere Befehle in der Päambel ihren Platz, wie z.B. Settings die von den Standardeinstellungen abweichen oder selbst festgelegt Abkürzungen.

Für den Anfang genügen uns die Standardeinstellungen der Dokumentenklasse „article“, also starten wir unser Dokument mit der folgenden Eingabe

```
\usepackage[utf8]{inputenc}

\begin{document}
\end{document}
```

2 Texte

Reine Texte in LaTeXzu erstellen ist sehr einfach, da man sich um Formatierungen keine Gedanken machen muss.

Bevor Du diesen Text in dein Beispiel einfügst, solltest du noch `\usepackage{xcolor}` und `\usepackage[ngerman]{babel}` in die Präambel einbinden. Diese ermöglichen die deutschen Umlaute, sowie Text in verschiedenen Farben anzeigen zu können.

Input:

```
Wenn wir bei \LaTeX einen Text, ist es egal ob wir
zwischen zwei Wörter ein        oder mehrere Leerzeichen
setzen. Der kompilierte Text sieht immer gleich, mit
einem Leerzeichen aus. Auch ist es egal, ob
und wo
Zeilenumbrüche gemacht werden. \LaTeX kompiliert alles zu
 einem Blocktext.

Aber selbstverständlich können auch Absätze erzeugt
werden, indem eine oder mehrere Zeilen frei gelassen
werden.

Will man Text \textbf{fett}, \textcolor{blue}{bunt} oder
\textit{kursiv} schreiben, so wird dies mit
entsprechenden Befehlen direkt im Code gemacht.
```

Output:

Wenn wir bei LaTeXeinen Text, ist es egal ob wir zwischen zwei Wörter ein oder mehrere Leerzeichen setzen. Der kompilierte Text sieht immer gleich, mit einem Leerzeichen aus. Auch ist es egal, ob und wo Zeilenumbrüche gemacht werden. LaTeXkompiliert alles zu einem Blocktext.
Aber selbstverständlich können auch Absätze erzeugt werden, indem eine oder mehrere Zeilen frei gelassen werden.
Will man Text **fett**, bunt oder *kursiv* schreiben, so wird dies mit entsprechenden Befehlen direkt im Code gemacht.

Es gibt ein paar Sonderzeichen, welche in LaTeX nicht direkt benutzt werden können: _, &, % sowie ^. Diese haben in LaTeX spezielle Bedeutungen und führen zu Fehler beim kompilieren.

Ein weiterer wichtiger Befehl ist `\newpage`. Dieser erzwingt einen Seitenumbruch - also Text unterhalb dieses Kommandos startet auf einer neuen Seite.

3 Umgebungen und Befehle

LaTeX ist so aufgebaut, dass alles was von der „normalem Text“ abweicht über Befehle und Umgebungen gesteuert wird. Die meisten Befehle haben den selben Aufbau

```
\name[optionale Argumente]{nötige Argumente}
```

Einen solchen Befehl, ohne optionales Argument, haben wir bereits bei `\usepackage{xcolor}` kennengelernt.

Eine Umgebeung hat stets die Form

```
\begin{name}
    ...
\end{name}
```

Umgebungen bewirken ein gewissen Verhalten, von allem was darin steht. So kann z.B. mit

```
\begin{center}
    ....
\end{center}
```

Text zentriert werden.

Umgebungen können auch ineinander verschachtelt werden, was wir eigentlich im Moment sogar schon machen, da wir unser Dokument mit `\begin{document}` gestartet haben und in dieser Umgebung die center-Umgebung aufgerufen haben.

Wichtig ist nur, dass die Umgebung die zuerst geöffnet wurde, als erstes wieder geschlossen wird.

4 Kapitel und Referenzen

Verschiedene Kapitel und Abschnitte lassen sich in LaTeX mit den folgenden Befehlen erstellen

---

Input:

```
\section{Abschnitt}
\subsection{Unterabschnitt}
\subsubsection{Unter-Unterabschnitt}
\paragraph{Paragraf}
\subparagraph{Unterparagraf}
```

Output:

1 Abschnitt

1.1 Unterabschnitt

1.1.1 Unter-Unterabschnitt

Paragraf

Unterparagraf

Bei z.B. der Dokumentklasse „report“ gibt es außerdem noch die Abstufungen `\part{}` und `\chapter{}`.

Mit dem Befehl `\setcounter{secnumdepth}{5}` können wir die Tiefe der Nummerierung ändern, das heißt, dass auch Paragrafen und Unterparagrafen eine Nummerierung erhalten. Der gleiche Input wie oben liefert dann

Output:

2 Abschnitt

2.1 Unterabschnitt

2.1.1 Unter-Unterabschnitt

2.1.1.1 Paragraf

2.1.1.1.1 Unterparagraf

Und hier sehen wir schon eine weitere Stärke von LaTeX: Sections (auch Bilder Subsections, Tabellen usw.) werden automatisch fortlaufend nummeriert. Dies hat den Vorteil, wenn wir ein Kapitel weiter oben einfügen, werden einfach alle darauf folgenden Nummern angepasst.

Um im Text auf einen anderen Abschnitt verweisen zu können verwenden wir den Befehl `\label{labelname}` direkt nach dem erstellen des Abschnitts. Dieses Vorgehen hat den Vorteil, dass ein Label direkt mit dem Abschnitt verknüpft ist, also nach dem Einfüen eines neuen Abschnitts auch hier die Nummerierung noch korrekt ist.

Input:

```
 Erstellen wir nun einen neuen Abschnitt, genauer
 \section{Ein sehr spannenden Abschnitt, der später noch
relevant wird}
 \label{seq:WichtigerAbschnitt}
```

Output:
Erstellen wir nun einen neuen Abschnitt, genauer

3 Ein sehr spannenden Abschnitt, der später noch relevant wird

Jetzt kann weiter unten im Text(oder auch oben) auf genau diesen wichtigen Abschnitt 3 verweisen werden, indem wir an der richtigen Stelle den Befehl `\ref{seq:WichtigerAbschnitt}` aufrufen. Das selbe gilt auch für Tabellen, Abbildungen und Gleichungen, doch dazu später mehr.

Durch dieses Vorgehen lässt sich ganz einfach ein Inhaltsverzeichnis mit dem Befehl `\tableofcontents` erstellen. Die „Tiefe“ des Inhaltsverzeichnisses, also ob Abschnitte, Unterabschnitte usw. auch angezeigt werden sollen, lässt sich mit dem Kommando `\setcounter{tocdepth}{...}` steuern, wobei statt ... ei-

ne Zahl zwischen -1 und 5 übergeben wird.

Tiefe	-1	0	1	2
Befehl	part	chapter	section	subsection
	3	4	5	
	subsubsection	paragraph	subparagraph	

5 Aufzählungen und Nummerierungen

Um Stichworte übersichtlich aufzubereiten werden sehr oft Aufzählungen (`itemize`) oder Nummerierungen (`enumerate`) benutzt. Diese Umgebungen sind in LaTeX sehr einfach umgesetzt.

Input:

```
\begin{itemize}
    \item Leonhard Euler
    \item Carl Friedrich Gauß
    \item Berhard Riemann
\end{itemize}
oder nummeriert
\begin{enumerate}
    \item Leonhard Euler
    \item Carl Friedrich Gauß
    \item Berhard Riemann
\end{enumerate}
```

Output:

- Leonhard Euler
- Carl Friedrich Gauß
- Berhard Riemann

oder nummeriert

1. Leonhard Euler
2. Carl Friedrich Gauß
3. Berhard Riemann

Hier eine kleine Fleißaufgabe: Versuche doch mal die Symbole der Aufzählung zu ändern, oder mit römischen Zahlen zu nummerieren (hierfür kann das package „enumerate“ nützlich sein) Ein mögliches Ergebnis könnte so aussehen:

$$ Leonhard Euler

* Carl Friedrich Gauß

$\int$ Berhard Riemann

(i) Leonhard Euler

(ii) Carl Friedrich Gauß

(iii) Berhard Riemann

Wie bereits erwähnt, lassen sich Umgebeungen ineinander verschachteln. Dies gilt auch für (`itemize`) und (`enumerate`).

- Leonhard Euler
  1. Exponentialfunktion & Logarithmus
  2. Eulerkreis
- Carl Friedrich Gauß
  - Fundamentalsatz der Algebra
  - Gauß-Verteilung
- Berhard Riemann
  1. Riemann-Integral
  2. Riemannsche Geometrie

6 Der Mathemodus

Der „Mathemodus“ erlaubt uns mathematische Formeln sauber darzustellen. Es gibt verschiedene Möglichkeiten den Mathemodus zu aktivieren, doch vorher müssen wir noch `\usepackage{amsmath,amssymb,amstext}` in die Präambel einbinden, um die volle Bandbreite an mathematischen Symbolen ausnutzen zu können.

In einem Fließtext, lässt sich der Mathemodus durch „$ ... $“ aktivieren. Alles zwischen den Symbolen wird als mathematische Formel interpretiert. Griechische Buchstaben werden einfach mit \Buchstabe bzw. \buchstabe (Groß- und Kleinschreibung!) eingefügt. So liefert z.B. `$\alpha + \Gamma = 26$` in einer Textzeile den Output $\alpha + \Gamma = 26$ (Achtung: es gibt nicht für jeden großen griechischen Buchstaben `\Beta` liefert z.B. eine Fehlermeldung).

Gleichungen können mithilfe der equation-Umgebung auch außerhalb eines Fließtexts platziert werden. Zur Veranschaulichung werden hier diverse mathematischen Operatoren ohne weitere Erklärung benutzt.

Anmerkung: Die Nummerierung der Gleichungen wird mit einem * unterdrückt.

Input:

```
\begin{equation}
    \sqrt{a^2+b^2}=c^2
    \label{eq:Pythagoras}
\end{equation}
\begin{equation*}
    E=m\cdot c^2
\end{equation*}
```

Output:

$$\sqrt{a^2+b^2}=c^2 \tag{1}$$
$$E=m\cdot c^2$$

Mit einer Kombination aus \label{...} und \ref{...} auch auf den Satz von Pythagoras in Gleichung 1 verwiesen werden. Die equation-Umgebung ist aber nur für einzeilige Gleichungen zu empfehlen. Für Mehrzeiler wird in der Regel `align` benutzt. Auch hier lässt sich die Nummerierung wieder mit einem * unterdrücken. Mit \\wird ein Zeilenumbruch erzeugt und durch „&“ z.B. hinter den „=“ lassen sich diese untereinder ausrichten.

Will man Text in einer der beiden Gleichungsumgebungen schreiben, so kann dies mit \text{...} gemacht werden (hier z.B. bei div). Text in Gleichungen wird oft bei verwendet um den Operator „div“ von dem Produkt $d \cdot i \cdot v$ zu unterscheiden. Deutlich eleganter ist, sich hierfür eigens ein Operator zu definiert, mehr dazu findest Du bei Overleaf.

Input:

```
\begin{align*}
    \nabla \cdot E =& \text{ div } E \\
    F =&G\frac{m_1m_2}{r^2}
\end{align*}
```

Output:

$$\begin{aligned}\nabla \cdot E =& \text{ div } E \\ F =&G\frac{m_1m_2}{r^2}\end{aligned}$$

Um Gleichungen hübsch aussehen zu lassen, kann es manchmal nötig sein die Größe von Klammern händisch zu ändern, insbesondere wenn Brüche oder Wurzeln verwendet werden. Der Befehl \quad wird hier verwendet um den Abstand zwischen den Brüchen zu vergrößern, dazu später mehr.

Input:

```
\begin{align*}
    (\frac{1}{2})^2 \quad
    \bigl(\frac{1}{2}\bigr)^2 \quad
    \Bigl(\frac{1}{2}\Bigr)^2 \quad
    \biggl(\frac{1}{2}\biggr)^2 \quad
    \Biggl(\frac{1}{2}\Biggr)^2
\end{align*}
```

Output:

$$(\frac{1}{2})^2 \quad \bigl(\frac{1}{2}\bigr)^2 \quad \Bigl(\frac{1}{2}\Bigr)^2 \quad \biggl(\frac{1}{2}\biggr)^2 \quad \Biggl(\frac{1}{2}\Biggr)^2$$

Die gleiche Systematik funktioniert auch mit anderen Klammer: $[\ldots]$, $|\ldots|$, $\|\ldots\|$ (Befehl: `\|` `\|`), $\langle\ldots\rangle$ (Befehl: `\langle` `\rangle`), $\{\ldots\}$ (Befehl: `\{` `\}`), $\lceil\ldots\rceil$ (Befehl: `\lceil` `\rceil`) und $\lfloor\ldots\rfloor$ (Befehl: `\lfloor` `\rfloor`).

6.1 Indizierung

Im Mathemodus lassen sich durch „_“ bzw. „^“ Indizes unten oder oben erstellen. Soll mehr als nur ein Buchstabe oder eine Zahle höher oder tiefer gestellt werden, muss der Inhalt in geschweifte Klammern gepackt werden. Sollen Indizes oben und unten erstellt werden, spielt die Reihenfolge, ob erst _{...} oder ^{...}, keine Rolle.

Input:

```
\begin{align*}
    x^2 \cdot y_i = v_{min} & \neq v_min
    a_i^{(j)} & = a^{(j)}_i
\end{align*}
```

Output:

$$\begin{aligned} x^2 \cdot y_i = v_{min} &\neq v_min \\ a_i^{(j)} &= a^{(j)}_i \end{aligned}$$

6.2 Abstände

Oft ist es nötig den Standardabstand von Zeichen händisch zu manipulieren. Dazu gibt es diverse Operatoren. Ebenso lässt sich der Abstand zwischen Zeilen händisch verändern.

Input:

```
\begin{align*} %Abstand in Zeilen
    f(x)    &=  e^x\!           +5\\
    f(x)    &=  e^x             +5\\
    f(x)    &=  e^x\,           +5\\
    f(x)    &=  e^x\:           +5\\
    f(x)    &=  e^x\;           +5\\
    f(x)    &=  e^x\            +5\\
    f(x)    &=  e^x\quad        +5\\
    f(x)    &=  e^x\qquad       +5
\end{align*}
\begin{align*} %Abstand zwischen Zeilen
    f(x)    &=  e^x+5\\
    f(x)    &=  e^x+5\\[0.5cm]
    f(x)    &=  e^x+5\\[2ex]
    f(x)    &=  e^x+5
\end{align*}
```

Output:

$$
\begin{aligned}
f(x) &= e^x+5\\
f(x) &= e^x+5\\
f(x) &= e^x\ +5\\
f(x) &= e^x\ +5\\
f(x) &= e^x\ +5\\
f(x) &= e^x\ +5\\
f(x) &= e^x\quad +5\\
f(x) &= e^x\qquad +5
\end{aligned}
$$

$$
\begin{aligned}
f(x) &= e^x+5\\
f(x) &= e^x+5\\[0.5cm]
f(x) &= e^x+5\\[2ex]
f(x) &= e^x+5
\end{aligned}
$$

6.3 Matrizen

Oftmals werden in mathematischen Dokumenten auch Matrizen benötigt. Um eine „normale“ Matrix mit runden Klammern zu erzeugen wird die `pmatrix` Umgebung benutzt. LaTeX bietet jedoch auch andere Klammern an. Hier der Reihe nach ein paar Beispiele mit den Umgebungen `matrix`, `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` und `Vmatrix`. Mit „&“ sowie \\ lassen sich die Elemente der Matrix anordnen (von links nach rechts und von oben nach unten).

Output:

$$\begin{matrix} 1 & a \\ x_r & b \cdot m \end{matrix} \qquad \begin{pmatrix} 1 & a \\ x_r & b \cdot m \end{pmatrix} \qquad \begin{bmatrix} 1 & a \\ x_r & b \cdot m \end{bmatrix}$$

$$\begin{Bmatrix} 1 & a \\ x_r & b \cdot m \end{Bmatrix} \qquad \begin{vmatrix} 1 & a \\ x_r & b \cdot m \end{vmatrix} \qquad \begin{Vmatrix} 1 & a \\ x_r & b \cdot m \end{Vmatrix}$$

Zwar keine Matrix, jedoch optisch fast schon ähnlich ist eine Fallunterscheidung, welche sich mithilfe der `cases` Umgebung erzeugen lässt

Input:

```
\begin{align*}
    \text{sgn}(x)=
    \begin{cases}
        1\quad &\text{ if } x>0\\
        0\quad &\text{ if } x=0\\
        -1\quad &\text{ if } x>0\\
    \end{cases}
\end{align*}
```

Output:

$$\begin{aligned} \text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x > 0 \end{cases} \end{aligned}$$

6.4 Summen und Integrale

Im diesem Unterabschnitt widmen wir uns noch Summen, Integralen und anderen Operatoren, die in irgendeiner Form Grenzen verwenden. Dies Funktioniert im Prinzip gleich wie die Indizerung, bzw. Höher- und Tieferstellen. Mit `_{...}` oder `^{...}` können die oberen, bzw. unteren Grenzen übergeben werden.

Input:

```
\begin{align*}
    \sum_{k=0}^{\infty} q^k = 1 + q + q^2 + \ldots
    = \frac{1}{1-q}\\
    \prod_{k=0}^{\infty} k = 0 \cdot 1 \cdot 2 + \ldots
    = 0\\
    \int_{a}^b f(x) \ dx = F(b)-F(a)\\
    \oint_\gamma f(z) \ dz =0\\
    \lim_{x\rightarrow\infty}\frac{1}{x}=0
\end{align*}
```

Output:

$$\sum_{k=0}^{\infty} q^k = 1 + q + q^2 + \ldots = \frac{1}{1-q} \qquad (\texttt{\textbackslash sum})$$

$$\prod_{k=0}^{\infty} k = 0 \cdot 1 \cdot 2 + \ldots = 0 \qquad (\texttt{\textbackslash prod})$$

$$\int_a^b f(x)\ dx = F(b) - F(a) \qquad (\texttt{\textbackslash int})$$

$$\oint_\gamma f(z)\ dz = 0 \qquad (\texttt{\textbackslash oint})$$

$$\lim_{x\to\infty} \frac{1}{x} = 0 \qquad (\texttt{\textbackslash lim})$$

7 Tabellen

Tabellen werden in LaTeX mit in der `tabular` Umgebung erstellt. Meist wird diese noch von einer `table` Umgebung eingefasst. Dabei handelt es sich um eine sog. *float Umgebung*, d.h. die Position der Tabelle im kompilierten Dokument befindet sich „da wo es gerade passt“. Dies kann aber auch durch gewisse Befehle beeinflusst werden. Der Vorteil der `table` Umgebung ist, dass sich Titel und Label direkt an die Tabelle knüpfen lassen. So können wir später wieder mit den Befehl `ref{...}` auf Tabelle 1 verweisen und auch hier werden die Nummerierungen dynamisch angepasst. Hier eine kleine Übersicht mit möglichen Argumenten, um die Tabelle zu positionieren.

Position	
h	„here“ - ’ungefähr’ an dieser Position.
t	„top“ - oben auf dieser Seite
b	„bottom“ - unten auf dieser Seite
P	„page“ - auf eine bestimmte Seite
h! oder H	„here“ - exakt hier

Das Aussehen der Tabelle wird als Argument der `tabular` Umgebung übergeben. Ein Aufruf könnte dann z.B. so aussehen `\begin{tabular}{| l || c r}`. Diese Umgebung erzeugt eine Tabelle mit drei Spalten, die erste ist linksbündig `l`, in der Zweiten ist der Text zentriert `c` und in der Dritten wird der Inhalt rechts ausgerichtet `r`. Die Spalten werden mit `|` oder `||` durch eine oder zwei vertikale Linien getrennt. Druch keinen dieser Striche werden die Spalten durch keine Linie getrennt.

Mit dem Befehl `\hline` werden horizontale Linien in die Tabelle gezogen. Um Zellen zu verbinden wird der Befehl `\multicol{}{}{}` benutzt. Im ersten Argument wird die Anzahl der zu verbindenden Zellen, im Zweiten das Aussehen, nach der gleichen Logik wie gerade beschrieben und im dritten Argument wird der Inhalt der Zelle übergeben. Hier ein Minimalbeispiel zum Verständnis.

Input:

```
\begin{table}[h]        % Position auf der Seite
    \centering
    \begin{tabular}{|| l | c  r ||}
                % Gestalt der Tabelle
        \hline      % Horizontale Linie
        \multicolumn{3}{||c||}{Überschrift}\\
                % Zellen verbinden
        \hline
         Versuch 1 & $30.1$ N  & $4$ cm\\
         Versuch 2 & $a^2+b^2=c^2$  & $5$ cm\\
         Zelle 1 & \multicolumn{2}{l||}{Zwei verbundene
                   Zellen}\\
         \multicolumn{2}{||c}{Und hier links} &
                   $\Gamma(\alpha,\beta)$\\
         \hline
    \end{tabular}
    \caption{Exemplarisch Tabelle ohne viel Sinn}
    \label{tab:Beispiel1}
\end{table}
```

Output:

Überschrift		
Versuch 1	30.1 N	4 cm
Versuch 2	$a^2 + b^2 = c^2$	5 cm
Zelle 1	Zwei verbundene Zellen	
Und hier links		$\Gamma(\alpha, \beta)$

Tabelle 1: Exemplarisch Tabelle ohne viel Sinn

Selbstverständlich sind der Kreativität keine Grenzen gesetzt und die Tabelle lässt sich mit verschiedenen Farben und Formen individualisieren. Hierfür wieder ein Verweis zu Overleaf mit einer ausführlichen Anleitung dazu wie viel Spaß man mit Tabellen haben kann.

8 Grafiken

In LaTeX gibt es auch unzählige Möglichkeiten Bilder und Grafiken anzuzeigen und in ein Dokument einzubinden. Üblicherweise wird dies mit der `figure` Umgebung und dem Befehl `\includegraphics[]{}`gemacht. Doch dafür ist das Package `graphicsx` nötig. Also laden wir dieses Package in der Präambel mit dem Kommando `\usepackage{graphicx}`

Die `figure` Umgebung ist wie auch die `table` Umgebung (vgl. Abschnitt 7) eine float Umgebung. LaTeX positioniert also das Bild an einer Stelle die „passt“. Dies kann mit den Positionsbefehlen aus Abschnitt 7 jedoch gesteuert werden.

Dem Befehl `\includegraphics[...]{...}` werden in der Regel zwei Argumente übergeben, auch wenn das Erste (in eckigen Klammern) optional ist. Im ersten Argument übergibt man die Größe des Bilds. Dies kann entweder absolut in Zentimetern (oder Millimetern) `[width=5cm]`, oder relativ zur Textbreite `[width=0.5\textwidth]` gemacht werden. Hierbei muss jedoch immer angegeben werden ob es sich um die Höhe (`height`) oder Breite (`width`) des Bilds handelt.

Das zweite Argument ist der Pfad zum Bild. Wenn sich dieses im selben Ordner wie das .tex Dokument befindet, dann ist nur der Name und das Format nötig, sieht also dann so, oder so ähnlich aus

`\includegraphics[width=0.42\textwidth]{huebschesBild.pdf}`.

Input:

```
\begin{figure}[h]
    \centering
    \includegraphics[width=0.42\textwidth]{surf.pdf}
    \caption{Oberflächenplot von $\text{sin}
                (\sqrt{x^2+y^2})$}
    \label{fig:Plot_sin_sqrt}
\end{figure}
```

Output:

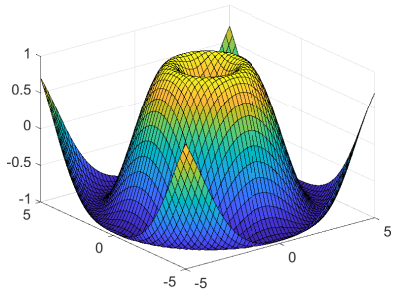


Abbildung 1: Oberflächenplot von $\sin(\sqrt{x^2+y^2})$

Auch wenn es zahlreiche andere Lösungen dafür gibt, hier eine sehr einfach Möglichkeit um Bilder in einem Gitter anzuordnen. Dafür benutzen wir das Package `subcaption`, welches wir mit dem Befehl `\usepackage{subcaption}` in der Päambel laden.

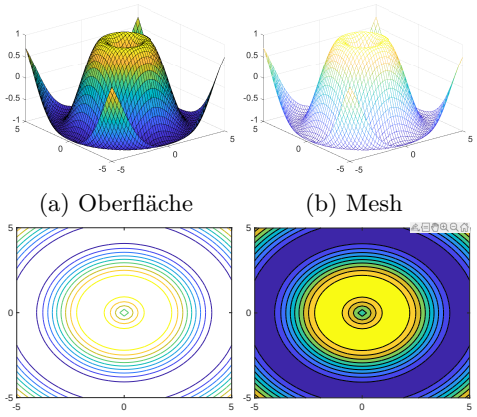
Nun wird innerhalb der `figure` Umgebeung der Befehl `\subcaptionbox{}{}` aufgerufen. Mit dem ersten Argument wird der Text, der unter dem Bild stehen soll, übergeben und mit dem Zweiten ein `\includegraphics[]{}` Befehl. Wir können wie gewohnt jedes einzelne Bild labeln, vgl. 2a oder 2b, indem wir `\label{...}` mit dem Text im ersten Argument übergeben, oder auf das ganze Gitter verweisen 2 indem wir inerhalb der `figure` Umgebung ein Label definieren.

Die Anordnung im Gitter wird über die Bildbreite der Einzelbilder gesteuert. Die Bildbreiten sollten sich maximal zur vollen Textbreite aufsummieren. Eine neue Zeile im Gitter wird wie üblich mit `\\` erstellt.

Input:

```
\begin{figure}[h]
    \centering
    \subcaptionbox{Oberfläche \label{fig:oberflaeche}}
            {\includegraphics[width=0.2\textwidth]
            {surf.pdf}}
    \subcaptionbox{Mesh \label{fig:mesh}}
            {\includegraphics[width=0.2\textwidth]
            {mesh.pdf}}\\
    \subcaptionbox{Kontur \label{fig:Konur}}
            {\includegraphics[width=0.2\textwidth]
            {contour.pdf}}
    \subcaptionbox{Kontour \& Fläche \label{fig:Konturf}}
            {\includegraphics[width=0.2\textwidth]
            {contourf.pdf}}
    \caption{Verschiedene Repräsentationen von
            $\text{sin}(\sqrt{x^2+y^2})$}
    \label{fig:gitter}
\end{figure}
```

Output:



(a) Oberfläche (b) Mesh

(c) Kontur (d) Kontour & Fläche

Abbildung 2: Verschiedene Repräsentationen von $\sin\left(\sqrt{x^2+y^2}\right)$

---

9 Referenzieren

[1] In LaTeX zu referenzieren ist sehr einfach, da eine Referenz nur einmal erstellt werden muss und dann mit einem „Label“ an beliebigen Stellen im Text darauf verwiesen werden kann. Mit nur einem Befehl kann dann ein Literaturverzeichnis erstellt werden kann, welches alle verwendeten Quellen enthält.

Zum Beispiel können wir auf Bernhard Riemanns Arbeiten in der Primzahltheorie [2] oder vielleicht doch lieber auf Harry Potter [1] verweisen. Dafür müssen wir eine neue Datei mit der Endung .bib im selben Verzeichnis erstellen, in welchem die .tex Datei liegt (vgl. Abbildung 3), erstellen. Nennen wir diese Datei `literatur.bib`. Bevor wir in dort die gewünschten Referenzen erstellen müssen wir folgendes in die Präampel integrieren

---

```
\usepackage[sorting=none]{biblatex}
\addbibresource{literature.bib}
```

In dieser Datei erstellen wir nun die gewünschten Referenzen. Diese sind wie folgt aufgebaut:

Mit `@type{...}` wird eine Umgebung dem „Typ“ der Quelle entsprechend erstellt. Üblich sind `@article{...}` oder `@book{...}`, es gibt aber deutlich mehr Typen zur Auswahl, siehe hier. Innerhalb der geschweiften Klammern werden sämtliche Details zur Quelle genannt, dies sollte mindestens Autor, Titel und Datum sein, jedoch sind es meist deutlich mehr Items die nötig sind um

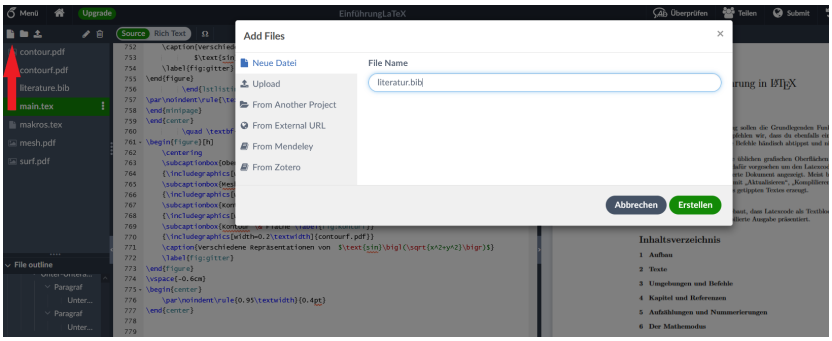


Abbildung 3: So erstellt man eine neue Datei in overleaf.

korrekt zu referenzieren. Eine Liste mit allen möglichen Items findet sich hier unter „fields“. Um die beiden oben genannten Quellen verweisen zu können sieht `literature.bib` also so, oder so ähnlich aus

```
@article{riemann1859,
  title={Über die Anzahl der Primzahlen unter einer
         gegebenen Größe},
  author={Riemann, Bernhard},
  journal={Ges. Math. Werke und Wissenschaftlicher
         Nachlaß},
  volume={2},
  number={145-155},
  pages={2},
  year={1859}
}

@book{rowling2002,
  address = {Hamburg},
  author = {Rowling, Joanne K.},
  publisher = {Carlsen},
  title = {Harry Potter und der Stein der Weisen},
  year = {2002}
}
```

Direkt nach dem Öffnen der `@type` Umgebung befindet sich noch die Id bzw. das Label der Referenz, in unserem Fall also `riemann1859` und `rowling2002`. Das Label kann grundsätzlich frei gewählt werden, doch bietet sich eine Kombination aus Autor, Jahr und/oder Titel meist an. Mit dem Kommando `\cite{riemann1859}` können wir nun an einer beliebigen Stelle in unserem Text auf das Paper von Riemann [2] verweisen. Am Ende des Dokuments können wir nun ganz einfach das Literaturverzeichnis erstellen indem wir `\printbibliography` aufrufen. Dieses Vorgehen hat den großen Vorteil, dass das Literaturverzeichnis immer geordnet ist - wenn eine Referenz weiter vorne im Text eingefügt wird,

werden alle Nummerierungen direkt aktualisiert. Auch das geordnete Literaturverzeichnis wird durch nachträglich eingefügte Quellen nicht aus der Ordnung gebracht. Durch zusätzliche Einstellungen, auf welche hier nicht eingegangen wird, lässt sich auch das Layout der Literaturverzeichnisse ändern.

Literatur

[1] Joanne K. Rowling. *Harry Potter und der Stein der Weisen*. Hamburg: Carlsen, 2002.

[2] Bernhard Riemann. „Ueber die Anzahl der Primzahlen unter einer gegebenen Grosse“. In: *Ges. Math. Werke und Wissenschaftlicher Nachlaß* 2.145-155 (1859), S. 2.

Hier erkennt man auch auch den Unterschied zwischen der `@article` und `@book`: Der Titel des Papers wird in „ “ geschrieben und das Journal wird mit aufgeführt, wobei beim Buch der Titel kursiv geschrieben ist.