

Lösungen für breite Tabellen in PDF-Seiten einpassen

Problemstellung und Herausforderungen

Bei der Konvertierung von Markdown (z.B. aus einem GitBook-Repository) zu PDF mit Pandoc und LuaLaTeX können Tabellen mit vielen Spalten oder langen Inhalten leicht breiter als eine A4-Seite werden. Das führt dazu, dass Tabellen am rechten Rand abgeschnitten werden oder über den Rand hinauslaufen. Pandoc nutzt standardmäßig das **LaTeX Longtable**-Umfeld für Tabellen (insbesondere bei Tabellen mit Caption) und bricht Zeilen um, wenn der Inhalt eine bestimmte Breite überschreitet. Allerdings gibt es Fälle, in denen Zeilenumbrüche nicht erfolgen – etwa wenn sehr lange „Wörter“ ohne natürliche Trennstellen (z.B. Strings mit Unterstrichen) in Zellen stehen ¹ ². LaTeX bricht standardmäßig keine Wörter ohne Trennmöglichkeiten um, was zu **Überlauf** führen kann ². Zudem erlauben einfache Spaltentypen wie `l`/`r` (links-/rechtsbündig) keine automatischen Zeilenumbrüche in Zellen ³.

Herausforderung Unterstriche: Ein spezielles Problem sind Unterstriche `_` in langen Bezeichnern. Wie von LaTeX-Experten beschrieben, verhindert ein Unterstrich im normalen Text die automatische Worttrennung, was dazu führen kann, dass der Begriff über den Rand ragt ⁴ ⁵. Ohne Anpassung versucht LaTeX entweder den langen Begriff in der Zeile zu behalten (und überläuft) oder würde ihn nur ungern ganz auf die nächste Zeile schieben, da beides typografisch suboptimal ist ⁶.

Strategien zur Anpassung breiter Tabellen

Es gibt mehrere etablierte Ansätze, um zu breite Tabellen auf die Seitenbreite anzupassen, selbst wenn sie über mehrere Seiten laufen dürfen:

- **Querformat-Seiten für breite Tabellen verwenden:** Wenn eine Tabelle extrem viele Spalten hat, kann man sie auf einer querformatigen Seite platzieren. Dazu wird die Seite temporär gedreht, um mehr horizontale Fläche zu bieten. In LaTeX lässt sich dies etwa mit der **pdfscape**-Umgebung erreichen. Man kann im Header z.B. `\usepackage{pdfscape}` definieren und eigene Makros für Einstieg/Ausstieg ins Querformat (`\blandscape`/`\elandscape`) hinzufügen ⁷. Die Tabelle wird dann zwischen `\begin{landscape}` und `\end{landscape}` gesetzt, sodass für diese Seite Querformat gilt ⁸ ⁹. Dieses Vorgehen wurde auf TeX.StackExchange erfolgreich demonstriert. Wichtig ist, nach der Tabelle wieder zurück auf Hochformat zu wechseln. (Hinweis: In Pandoc kann man alternativ einen **fenced Div** mit Klasse `.landscape` nutzen, der mittels `fenced_divs` + `raw_attribute` Extension im Markdown die `\begin{landscape}`-Umgebung erzeugt ¹⁰.)
- **Schriftgröße der Tabelle verkleinern:** Eine einfache Maßnahme ist, den Text in der Tabelle in kleinerer Schrift zu setzen, damit mehr Inhalt in die Breite passt. So empfiehlt es sich etwa, vor der Tabelle `\small` oder `\footnotesize` einzuschalten und danach wieder `\normalsize` ¹¹ ¹². Auf Stack Overflow wurde z.B. vorgeschlagen, direkt im Markdown eine LaTeX-Anweisung `\footnotesize` vor die Tabelle einzufügen ¹¹. Damit wird die gesamte Tabelle in kleineren Schriftgraden gesetzt, was die benötigte Breite deutlich reduziert. Wichtig ist, nach der

Tabelle die Schriftgröße wieder auf den Normalwert zurückzusetzen (etwa durch eine leere Zeile und `\normalsize`¹² oder durch Einbettung der Tabelle in `{ \begingroup ... \endgroup }`).

- **Textumbruch innerhalb von Tabellenzellen ermöglichen:** Damit lange Zellinhalte auf mehrere Zeilen umgebrochen werden können, muss der Tabellenmodus dies zulassen. In LaTeX bedeutet das, Spaltentypen wie `p{Breite}` oder flexiblen `X` (aus dem **tabularx**-Paket) anstelle von starren `l/r/c` zu verwenden³¹³. Pandoc bricht von sich aus Zellen um, *falls* eine Zeile der Markdown-Tabelle länger ist als die definierte Spaltenbreite (Standard 72 Zeichen, anpassbar via `--columns`)¹. In diesem Fall verteilt Pandoc die Tabelle über die volle Textbreite und setzt relative Spaltenbreiten gemäß der Markdown-Definition (Anzahl der `-` in der Trennlinie)¹.
Praktische Tipps: Man kann in der Markdown-Tabelle gezielt längere Zeilen (z.B. durch Einfügen eines unsichtbaren langen Kommentars) erzeugen, um Pandocs Umbruch-Mechanismus zu triggern. Alternativ kann ein **Pandoc Lua-Filter** die Spaltenbreiten nachträglich so anpassen, dass die Summe 100% Textbreite ergibt – damit werden alle Zellen umbrechbar. Ein Beispiel dafür wurde von Pandoc-Entwicklern vorgestellt: Der Filter iteriert über `tbl.colspecs` und normiert die Breiten auf relative Werte, sodass `\textwidth` komplett ausgenutzt wird¹⁴¹⁵. Ergebnis: Die Tabelle wird als Longtable mit `p{X\textwidth}`-Spalten ausgegeben, und LaTeX bricht längere Wörter an passenden Stellen um (bei normalen Wörtern durch Silbentrennung, bei Code mit Unterstrichen jedoch nicht automatisch, siehe unten).
- **Lange Wörter aufteilen oder trennbar machen:** Für Inhalt ohne Leerzeichen (z.B. `SOME_LONG_VARIABLE_NAME`) muss man LaTeX aktiv erlauben, an bestimmten Stellen zu brechen. Eine Möglichkeit ist, manuell `\linebreak` oder `\-` Soft-Hyphen einzufügen, was jedoch die Markdown-Quellen unleserlich macht. Besser ist eine generelle Lösung: **Unterstriche als Trennstelle definieren**. Auf TeX.SE wird empfohlen, den Unterstrich-Befehl umzudefinieren, damit nach einem `_` ein Zeilenumbruch möglich ist¹⁶. Zum Beispiel kann im Präambel stehen: `\renewcommand_ {\textunderscore\allowbreak}`, wodurch LaTeX bei Bedarf an Unterstrichen umbrechen darf¹⁶. In Pandoc lässt sich so etwas mittels `--include-in-header` einbinden (wie in einem Pandoc-Issue vorgeschlagen wurde) – damit werden Variablennamen mit `_` nicht mehr als untrennbarer Block behandelt⁴. Neben Unterstrichen können ähnlich auch *Bindestriche* oder Punkte als erlaubte Bruchstellen definiert werden (z.B. via `\hyphenchar\font45` für `-`). Dies stellt sicher, dass Begriffe wie URLs, Pfade oder Code-Identifikatoren innerhalb der Tabellenzellen umbrechen statt über den Rand zu ragen.
- **Spaltenabstände verringern:** Wenn nur wenig Platz fehlt, kann man den horizontalen Innenabstand in Tabellenzellen reduzieren. LaTeX verwendet den Parameter `\tabcolsep` für halbseitigen Zellenrand. Durch `\setlength{\tabcolsep}{<Wert>}` oder relativer Änderung `\addtolength{\tabcolsep}{-2pt}` kann die Tabelle etwas zusammengedrückt werden¹⁷. Dieser Trick, kombiniert mit kleinerer Schrift, kann oft ein paar Zentimeter gewinnen¹⁸. Man sollte jedoch darauf achten, die Lesbarkeit nicht zu sehr zu beeinträchtigen.
- **Notfalls Skalierung verwenden:** Als letzte Option kann die gesamte Tabelle skaliert oder verkleinert werden. Mit `\resizebox{\textwidth}{!}{...}` lässt sich eine Tabelle zwangsweise auf Seitenbreite schrumpfen¹⁹. Oder mit `\scalebox{0.9}{...}` um 90% verkleinern²⁰. Dies sollte allerdings wirklich nur im Notfall benutzt werden, da es die Schrift sehr klein machen kann. Besser ist, gezielt Schriftgrade und Umbrüche zu steuern wie oben beschrieben, anstatt die Tabelle als Grafik zu skalieren.

Bestehende professionelle Lösungen

Auf Plattformen wie GitHub und Stack Overflow wurden diese Strategien bereits diskutiert und umgesetzt. Einige Beispiele:

- **Pandoc + LaTeX Community:** Das Pandoc-Team selbst hat bekannte Issues dazu. Ein Problembericht zeigte, dass GitHub-Flavored Markdown Tabellen in Pandoc 2.x zunächst nicht umbrachen und Text über den Rand ging ²¹. Die Lösung dort bestand darin, Pandocs Markdown-Parser so zu benutzen, dass Spaltenbreiten erkannt werden (z.B. durch Nutzung von `markdown_github` Input oder entsprechende Updates im Pandoc-Code). Für konkrete Sonderfälle (Unterstriche in Wörtern) wurde – wie oben erwähnt – empfohlen, via `--include-in-header` einen LaTeX-Fix einzubinden, der Unterstriche trennbar macht ⁴. Diese Vorgehensweise kann als „professionelle“ Empfehlung gelten, da sie direkt vom Pandoc-Repository kommt.
- **Stack Overflow:** In einem Q&A wurde vorgeschlagen, einfach LaTeX-Befehle ins Markdown einzubetten, um Tabellen anzupassen. Beispielsweise riet ein Nutzer dazu, `\footnotesize` vor die Tabelle zu setzen, da Markdown selbst keine Font-Size Syntax hat ¹¹. Ein anderer Kommentar ergänzte, man solle danach ein `\normalsize` einfügen, um den Effekt auf nachfolgenden Text zu begrenzen ¹². Diese Lösung ist pragmatisch und schnell umzusetzen – sie erfordert aber, dass man den Markdown-Quelltext anpasst.
- **TeX-StackExchange:** Hier findet man komplexere Lösungen für Layout-Probleme. So wurde bei einer „table too wide“-Frage empfohlen, anstatt des normalen `tabular`-Umfelds lieber `tabularx` mit automatisch brechenden X-Spalten zu verwenden ³. In der Musterlösung definierte der Experte eine neue Spaltendefinition `X` mit `\raggedright\arraybackslash\hspace{0pt}` (oder `\Centering` für zentriert) und ersetzte sämtliche `r`-Spalten durch diese flexiblen Spalten ²². Dadurch verteilt sich die Tabelle gleichmäßig über die Textbreite und bricht an den Zelleninhalten automatisch um. Zudem wurde geraten, überflüssige vertikale Linien zu entfernen und Tabellen hübscher mit **booktabs** zu gestalten ²³ – das trägt zwar nicht direkt zur Breite bei, verbessert aber die Lesbarkeit. Diese Lösung aus der LaTeX-Community zeigt, dass professionelle Typografierung oft Kombinationen mehrerer Tricks nutzt: passende Umgebung (`longtable/tabularx`), kleinere Schrift für bestimmte Spalten (`>\small` in der Präambel) ²⁴, und sauberes Design.
- **Pandoc Lua-Filter auf GitHub/Reddit:** Fortgeschrittene Anwender haben Lua-Filter geschrieben, die Pandocs Abstract Syntax Tree (AST) manipulieren, um automatisch breite Tabellen umzubauen. Ein auf Reddit geteilter Filter (von Pandoc-Maintainer Albert Krewinkel) iteriert über jede Tabelle und skaliert die Spaltenbreiten relativ auf 1.0 Gesamtbreite ¹⁴ ¹⁵. So etwas könnte man erweitern, um z.B. ab einer bestimmten Spaltenanzahl automatisch einen `landscape`-Div um die Tabelle zu setzen, oder `\small` innerhalb der Tabelle einzufügen. Vorteil solcher Filter: Man muss den Markdown-Quelltext nicht ändern; der Filter greift während der PDF-Erstellung ein. Einige GitHub-Repositories (wie *Wandmalfarbe/pandoc-latex-template*) diskutieren ebenfalls, wie man Landscape-Seiten oder breite Tabellen unterstützt, was zeigt, dass es ein aktuelles Thema ist ²⁵ ²⁶.

Zusammenfassend stehen also **mehrere kombinierbare Lösungen** bereit, die auch in der Praxis von anderen Entwicklern benutzt werden. Im nächsten Abschnitt folgt eine mögliche Umsetzungsschritt-Liste (Tasks), wie man diese Lösungen automatisieren könnte.

Umsetzungsschritte (Codex Tasks)

Um die obigen Strategien in einem Konvertierungs-Tool oder Skript (z.B. mittels eines Codex-Systems oder Pandoc-Filtern) umzusetzen, bieten sich folgende Schritte an:

- 1. Weite Tabellen erkennen:** Analysiere den Markdown-Inhalt nach Tabellen. Kriterien können die Spaltenanzahl (z.B. > 5 Spalten) oder die Zeichenlänge von Tabellenzeilen sein. Pandocs `--columns` Wert (standardmäßig 72) kann als Schwelle dienen. Falls eine Tabelle in reiner Textbreite voraussichtlich breiter als die Seite wird (z.B. Summierte Spaltenbreite > ~15cm bei A4-Hochformat), markiere sie als „zu breit“. (Diese Erkennung kann im Pandoc Lua-Filter via `tbl.width` oder heuristisch via Spaltenanzahl erfolgen.)
- 2. Tabellenlayout anpassen:** Für als zu breit erkannte Tabellen wende eine oder mehrere der Strategien an:
- 3. Landscape-Umgebung einsetzen:** Füge vor und nach der Tabelle eine Markierung ein, die LaTeX veranlasst, diese Seite im Querformat auszugeben. In Pandoc kann man um die Table-Node einen Div mit Attribut `{.landscape}` legen, vorausgesetzt im Header ist `\usepackage{pdflscape}` geladen ⁷. Alternativ im LaTeX-Template automatisch `\begin{landscape}` vor und `\end{landscape}` nach der entsprechenden Tabelle einfügen.
- 4. Schrift verkleinern:** Ergänze unmittelbar vor der Tabelle einen LaTeX-Befehl wie `\small` oder `\footnotesize` und nach der Tabelle `\normalsize`. Dies kann der Filter tun, indem er z.B. die Tabelle von `{\small ... }` umgibt oder entsprechende Raw-Blocks einfügt ¹¹. So wird nur die Tabelle in kleiner Schrift gesetzt.
- 5. Spaltenbreiten setzen für Umbruch:** Modifiziere die Pandoc-Tabelle im AST so, dass jede Spalte eine relative Breite erhält. Beispielsweise kann für jede `colspec` ohne feste Breite ein Default-Wert vergeben werden. Anschließend skaliere alle Spaltenbreiten so, dass ihre Summe = 1.0 ist (100%) ¹⁴ ¹⁵. Pandoc wird dann in LaTeX `p{<width>\textwidth}`-Spalten erzeugen, die automatische Zeilenumbrüche ermöglichen. Falls Pandoc stattdessen `tabular` mit `l/r` Spalten nutzen würde, zwinge den Gebrauch von `longtable` / `tabularx` entweder durch Hinzufügen einer Caption (damit Pandoc `longtable` wählt) oder durch einen gezielten Raw-Ersatz der Umgebung.
- 6. Unterstrich-Trennbarkeit sicherstellen:** Falls im Dokument Code-Bezeichner mit `_` vorkommen, füge einmalig in den Präambel-Header der LaTeX-Ausgabe eine Definition ein: `\renewcommand_ {\textunderscore\allowbreak}` ¹⁶. Dies kann via Pandocs Option `--include-in-header` geschehen oder durch den Filter, der einen Header-includes Block generiert. Dadurch werden Unterstriche in der Tabelle nicht mehr als untrennbare Stelle behandelt.
- 7. Zusätzliche Feinjustierung:** Je nach Bedarf kann der Filter weitere Optimierungen vornehmen, z.B. vertikale Linien aus Tabellen entfernen (für Platz und Ästhetik), `\tabcolsep` verkleinern (durch Einfügen von `\setlength{\tabcolsep}{...}` vor der Tabelle) oder bestimmte breite Spalten gezielt in kleinerer Schrift rendern (`\scriptsize` für nur diese Spalte, mit Hilfe der LaTeX `>\scriptsize` Syntax in der Tabellenpräambel ²⁴). Diese Schritte sind optional, können aber bei extrem vielen Spalten hilfreich sein.
- 8. Konvertierung mit Pandoc ausführen:** Wende den Pandoc-Aufruf mit obigem Filter bzw. den eingebauten Modifikationen an. Zum Beispiel: `pandoc -f gfm+emoji+fenced_divs+raw_attribute -t latex --pdf-engine=lualatex --`

`include-in-header=table_fixes.tex input.md -o output.pdf`. Stelle sicher, dass `table_fixes.tex` die definierten Anpassungen (wie die `_Neu-Definition` oder `Landscape-Paket`) enthält. Falls der Prozess im Code erfolgt (z.B. via einem Python-Skript mit `py pandoc` oder dem Pandoc CLI-Aufruf), achte darauf, dass `args.wrap_wide_tables==True` diese Logik aktiviert.

9. **Überprüfung und Iteration:** Prüfe das Ergebnis-PDF hinsichtlich Tabellen. Kontrolliere, ob alle breiten Tabellen nun vollständig sichtbar sind (ggf. auf Querformat-Seiten) und optisch ansprechend dargestellt werden. Achte darauf, dass nach einer Querformat-Tabelle wieder Hochformat verwendet wird und dass Schriftgrößen zurückgesetzt wurden. Falls eine Tabelle immer noch Probleme macht (z.B. Text zu klein oder unschöne Umbrüche), justiere die Strategie: eventuell doch Landscape verwenden oder Spalten aufteilen. Dieser Schritt stellt sicher, dass die implementierte Lösung robust ist.

Durch diese Tasks wird eine **automatische Anpassung** breiter Tabellen erreicht. Die Tabellen dürfen über mehrere Seiten laufen (LaTeX `longtable` sorgt für Seitenumbruch), werden aber in der Breite so skaliert bzw. formatiert, dass sie innerhalb der Seitenränder bleiben. Die Kombination aus Pandocs Fähigkeiten (z.B. fenced Divs für Landscape) und LaTeX-Anpassungen (Schriftgröße, flexible Spalten) gewährleistet ein professionelles Layout der Tabellen im PDF-Dokument.

Quellen: Die oben genannten Lösungen und Schritte stützen sich auf Empfehlungen aus der Pandoc-Dokumentation ¹, bewährten LaTeX-Tricks aus Foren ²⁷ ¹⁶ sowie Code-Beispielen von Stack Exchange und GitHub (siehe Referenzen). Diese Vorgehensweise fasst aktuelle Best Practices zusammen, um das Ziel *"Tabellen sollen von der Breite auf eine PDF-Seite passen"* erfolgreich zu erreichen.

¹ Tables

<https://www.pandoc.org/demo/example33/8.9-tables.html>

² ⁵ ⁶ ¹⁶ Underscore makes text go past end of line into margins - TeX - LaTeX Stack Exchange

<https://tex.stackexchange.com/questions/9937/underscore-makes-text-go-past-end-of-line-into-margins>

³ ¹³ ²² ²³ latex table too wide - TeX - LaTeX Stack Exchange

<https://tex.stackexchange.com/questions/435860/latex-table-too-wide>

⁴ ²¹ long text in pipe table does not wrap in pdf · Issue #4239 · jgm/pandoc

<https://github.com/jgm/pandoc/issues/4239>

⁷ ⁸ ⁹ ¹⁰ xetex - How to rotate a Markdown table to landscape? Solutions I've tried print the markup instead of the rendered table - TeX - LaTeX Stack Exchange

<https://tex.stackexchange.com/questions/580386/how-to-rotate-a-markdown-table-to-landscape-solutions-ive-tried-print-the-mark>

¹¹ ¹² r - Reduce cell width and font size of table using pandoc.table() - Stack Overflow

<https://stackoverflow.com/questions/16424545/reduce-cell-width-and-font-size-of-table-using-pandoc-table>

¹⁴ ¹⁵ Fixed width tables in PDFs : r/pandoc

https://www.reddit.com/r/pandoc/comments/x4ssrb/fixed_width_tables_in_pdfs/

¹⁷ ¹⁸ ¹⁹ ²⁰ ²⁴ ²⁷ Table Size and Figure Spacing in LaTeX - LaTeX.org

<https://latex.org/forum/viewtopic.php?t=803>

²⁵ ²⁶ GitHub · Where software is built

<https://github.com/Wandmalfarbe/pandoc-latex-template/issues/45>