

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

35. Jahrgang Heft 3/2023 August 2023

3/2023

Impressum

»Die \TeX nische Komödie« ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Autoren wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nicht gestattet. Alle Rechte zur weiteren Verwendung außerhalb von DANTE e.V. liegen bei den jeweiligen Autoren.

Beiträge sollten in Standard- \LaTeX -Quellcode unter Verwendung der Dokumentenklasse dtk erstellt und per E-Mail oder Datenträger (z. B. CD/DVD) an unten stehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, \LaTeX -Pakete oder Schriften notwendig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden. Weitere Informationen für Autoren findet man auf der Projektseite <https://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit Lua \LaTeX , Version 1.17.0 (TeX Live 2023) erstellt. Als Standard-schriften kamen Libertinus Serif, Libertinus Sans Serif, Anonymous Pro und Libertinus Math zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2150

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung \TeX e.V.
Postfach 11 03 61
69072 Heidelberg

E-Mail: office@dante.de (DANTE e.V.)
dtkred@dante.de (Redaktion)

Druck: Schleunungsdruck GmbH
Eltertstraße 27, 97828 Markttheidenfeld

Redaktion: Luzia Dietsche (verantwortliche Redakteurin)

Mitarbeit: Adelheid Bonnetsmüller Eberhard Lisse Ralf Mispelhorn
Rolf Niepraschk Stefan Pinnow Bernd Raichle
Christine Römer Herbert Voß

Redaktionsschluss für Heft 4/2023: 15. Oktober 2023

ISSN 1434-5897

Die \TeX nische Komödie 3/2023

Editorial

Liebe Leserinnen und Leser,

wie immer steht diese Kolumne für Sie am Anfang einer Ausgabe und wie immer, wenn auch nicht nach außen sichtbar, ist diese Kolumne bei der Entstehung einer Ausgabe die letzte, die mit Inhalt gefüllt wird. Weil ich erst am Ende weiß, *womit* ich sie füllen will.

Dieses Mal hat mich der Brief einer anderen Chefredakteurin daran erinnert, dass ich viel zu selten allen danke, die zum Entstehen einer Mitgliederzeitung beitragen. Ich bin mir nicht sicher, ob ich zuerst die Mitglieder des Redaktionsteams oder die Autor*innen nennen soll – ohne Beiträge keine Mitgliederzeitung, ohne Redaktionsteam eine sehr fehlerhafte Mitgliederzeitung. Also geht mein Dank an beide Gruppen in gleichem Maß für ihre Ausdauer und Geduld mit mir.

Dann sind da aber auch noch die vielen »stillen« Helfer, als da wären Herbert Voß, der nach wie vor immer einspringt, wenn ich beim Kompilieren nicht weiter weiß. Rolf Niepraschk, der die dtk-Klasse betreut und anpasst. Bernd Raichle, der direkter Ansprechpartner für meinen Frust über meine Unfähigkeit ist. Erik Braun, der als Server-Admin auf das Geduldigste meine ahnungslosen Fragen zu unserer Arbeitsumgebung beantwortet. Thomas Demmig, der einmal zustimmte, die *L^AT_EX News* zu übersetzen und nun nicht mehr davon los kommt. Jürgen Fenn, der ohne Unterlass die Rubrik *Neue Pakete auf CTAN* betreut. Adelheid Bonnetsmüller, die die Stammtisch-Seiten aktuell hält. Uwe Ziegenhagen, der die Mitgliederzeitung online verwaltet. Und die vielen Anderen, deren Namen mir mit Sicherheit einfallen, wenn diese Ausgabe fertiggestellt und im Druck sein wird. Ohne viele, viele Beteiligte wäre diese Mitgliederzeitung nicht möglich – Dank an alle für ihre Mitwirkung!

So verbleibe ich dieses Mal ohne direkten Bezug zu Beiträgen dieser Ausgabe und wünsche Ihnen und Euch wieder viel Spaß beim Lesen

Luzia Dietsche



Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

ich gebe es zu: Ich habe kurz mit dem Gedanken gespielt, mich von ChatGPT bei der Erstellung dieses Grußworts unterstützen zu lassen. Doch ist ein persönliches Grußwort (noch) nicht so einfach nachzuahmen wie so manches »Standardschriftstück« bzw. ist die Menge an zusätzlich nötigen Informationen zwar nicht unüberwindbar, doch (für dieses Mal) noch ausreichend abschreckend gewesen. Und natürlich darf man als Mitglied zurecht ein bisschen Einsatz erwarten. Und so beginne ich diesmal mit meinen Eindrücken der TUG 23.

Eine TUG-Tagung ist immer wieder etwas besonderes. Zunächst ist der Tagungsort in der Regel ein Hotel, so dass die Wege zwischen Tagungen, Mahlzeiten, Abendprogramm und Nickerchen zwischendurch kurz sind (sofern man im selben Hotel nächtigt). Zudem sind die Teilnehmerinnen und Teilnehmer wie auch die Referentinnen und Referenten sehr international und die Vorträge decken ein äußerst breites Themenspektrum ab. Abgerundet wird das ganze immer auch von einer touristischen Komponente.

Ich freue mich sehr, dass aus vermutlich mindestens einem dieser Gründe viele von Euch die Gelegenheit genutzt haben, die TUG 23 in Bonn zu besuchen und sich aktiv am Vortragsprogramm, aber eben auch an den Diskussionen beim Frühstück, in den Pausen oder am Abend zu beteiligen. Vier Tage »die volle Dröhnung \TeX « könnte man sagen. Mehr dazu gibt es in dieser Ausgabe der Mitgliederzeitung ab S. 7.

Im Vorfeld der TUG 23 fand am 13. Juli unsere 65. Mitgliederversammlung statt. Im Mittelpunkt des Vorstandsberichts und der anschließenden Diskussionen standen die aktuelle Bürosituation sowie die Planungen für den Auszug aus den Räumen in Heidelberg.

Gerade der letzte Punkt wird uns im Vorstand, aber auch den Verein insgesamt bis April 2024 noch viel beschäftigen. Weil es viele Einzelfragen betrifft und wir uns

an der ein oder anderen Stelle Unterstützung aus der Mitgliederschaft wünschen, findet ihr auf S. 5 Informationen und Anmerkungen rund um den Auszug.

Zu den diesjährigen Vorstandswahlen trat der bisherige Vorstand geschlossen wieder an. Zusätzlich kandidierte Oliver Rath für das Amt eines Beisitzers. Alle Kandidaten wurden gewählt, so dass wir in den kommenden Jahren wieder mit insgesamt acht Personen im Vorstand agieren.

Alles wie gehabt also? – Mitnichten. Die Entlastung des alten Vorstands für das Geschäftsjahr 2022 wurde aufgrund noch offener Fragen der Rechnungsprüferinnen nicht beantragt und musste daher verschoben werden. Auch wenn sich die Fragen sicherlich klären lassen, hat mir der Vorgang verdeutlicht, wie sehr die aktuelle Situation uns fordert und wie wichtig es ist, die laufenden Aufgaben durch externes Personal und professionelle Softwarelösungen zu unterstützen.

Daher haben wir in den vergangenen Wochen zum einen eine Buchhalterin engagiert, die zunächst die Finanzen des aktuellen Jahres in den Blick nehmen wird. Zum anderen wird es mit Campai eine neue Software für die Mitgliederverwaltung und die Buchungen geben. Wir werden damit auch in der Lage sein, den Mitgliederbestand gemeinsam zu pflegen und dadurch die aufkommenden Arbeiten besser zu verteilen.

Insgesamt haben wir somit hoffentlich eine tragfähige Lösung für die fortlaufenden Alltagsaufgaben gefunden und können uns auf die anstehende Büroräumung sowie weitere organisatorische und strategische Themen konzentrieren.

In diesem Sinne wünsche ich noch einen schönen Sommer und viel Vergnügen bei der weiteren Lektüre dieser „Die T_EXnische Komödie“. Oder wie mir ChatGPT vorschlagen würde: »Happy T_EXing«.

Herzlichst Ihr/Euer
Martin Sievers

Büroräumung – Bitte um Unterstützung

DANTE e.V. wird das Büro in Heidelberg spätestens zum 24. März 2024 abgeben. Wie im Grußwort dieser Ausgabe angekündigt, folgen ein paar weiterführende Informationen dazu verbunden mit der Bitte, an der ein oder anderen Stelle mit Rat und Tat beiseite zu stehen.

Im September trifft sich der Vorstand, um vor Ort die einzelnen Schritte zu planen. Folgende Dinge stehen in jedem Fall an:

Buchbestand: Wir werden die »Präsenzbibliothek« zusammenhalten, die zahlreichen Bücher der DANTE-Edition aber nach Möglichkeit abgeben.

Falls ihr Kontakte zu Bibliotheken habt, fragt gerne nach, ob dort grundsätzliches Interesse an Büchern besteht und gebt uns Bescheid. Ihr benötigt alte Ausgaben der Mitgliederzeitung? Dann schnell melden.

Buchversand: Der Buchversand war lange Zeit eine wichtige Säule. Auch wenn die Nachfrage in den vergangenen Jahren nachgelassen hat, werden wir den Buchversand für Mitglieder voraussichtlich weiterführen. Einen Freiwilligen für die Organisation gibt es bereits.

Entsorgung: Das papierlose Büro ist bislang ein Traum geblieben. Wir haben viele Unterlagen, die nicht so einfach über den Papiermüll entsorgt werden können. Auch müssen hunderte DVDs entsorgt werden.

Kennt ihr eine Firma, die datenschutzkonform Unterlagen entsorgt? Oder eine, die große Papiermengen und/oder Datenträger entsorgt?

Wir werden im Spätherbst zu Räumungsterminen im Büro zusammenkommen und würden uns freuen, wenn uns Mitglieder aus Heidelberg und Umgebung bei der Arbeit unterstützen.

Mobiliar: Im Büro stehen einige alte Schränke, Tische und Stühle, die wir sehr wahrscheinlich über entsprechende Kleinanzeigen zur Selbstabholung ausschreiben werden. Einen Freiwilligen zur Umsetzung gibt es bereits.

Schönheitsreparaturen: Eine endgültige Klärung steht noch aus, es könnte aber sein, dass wir auch noch streichen und ein paar Dinge im Büro in Stand setzen müssen. Hierfür ist Zeit und Expertise gefragt.

Wir suchen für vermutlich Februar 2024 zuverlässige Handwerksfirmen aus Heidelberg und Umgebung bzw. Personen mit handwerklichem Geschick.

Fragen und Anmerkungen zu einem der obigen Punkte oder zu etwas anderem im Zusammenhang mit der Büroauflösung bitte direkt an office@dante.de.

Eindrücke von Bonn, Mitgliederversammlung und TUG 23¹

Luzia Dietsche

In diesem Jahr fanden sowohl die Mitgliederversammlung von DANTE e.V. als auch die Tagung der TeX Users Group in der ehemaligen Bundeshauptstadt Bonn statt. Die vielfältigen Eindrücke, die ich gewonnen habe, sollen hier in Wort und Bild dargestellt werden.

Vorbemerkung

Normalerweise versuche ich als verantwortliche Redakteurin der Mitgliederzeitung gleich zu Beginn einer Tagung jemanden für die Berichterstattung zu finden. Dieses Mal bin ich gescheitert und sehe mich nun in der Position, diesen Bericht selbst zu schreiben. Da ich zugegeben nicht allen Vorträgen der Tagung hingebungsvoll gelauscht habe und Kurzzusammenfassungen der Vorträge auf den Konferenzseiten unter <https://www.tug.org/tug2023/> zu finden sind, werde ich nur auf ganz wenige Vorträge näher eingehen. Die ausführlichen Beiträge werden im TUGBoat, Vol. 44, No. 2 veröffentlicht werden. Mein Artikel beschränkt sich hauptsächlich auf meine Eindrücke in Wort und Bild und bietet keine ausführliche Beschreibung meines Tagesablaufs.



Bonn

Am ersten Nachmittag in Bonn hatte ich die Wahl zwischen einem *PDF Developers' Workshop* und dem Stadtrundgang durch einen *Greeter*. Shame on me – ich beschloss, dass ich nicht der Entwicklerszene angehöre, sondern besser bei dem Rundgang aufgehoben bin.

Dabei erfuhr ich als Erstes von den *Greeters*², einer internationalen Vereinigung, deren ehrenamtlich fungierende Mitglieder Gäste in ihrer Stadt willkommen heißen und sie mitnehmen auf einen persönlichen, zwei- bis dreistündigen Spaziergang. Der lokale Organisator der Tagung, Gert Fischer, ist Mitglied und hat mit einem zweiten Freiwilligen zwei Gruppen durch die Stadt geführt.

Ich weiß jetzt, dass Bonn deutlich mehr als nur Bundeshaupt- und Beethovenstadt ist. Es gibt eine wunderschöne Innenstadt, ein Kaufhaus, in dem eine winzige Kapelle eingeschlossen ist, und eine Buchhandlung, die deutlich erkennbar früher ein

¹ Die verwendeten Fotografien stammen von R. Kotucha, R. Laakso, B. Raichle und L. Dietsche.

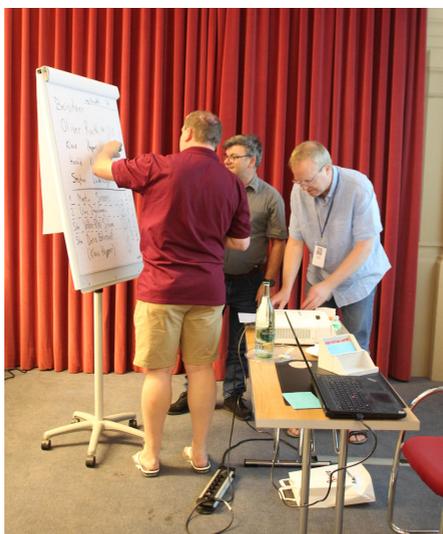
² <https://internationalgreeter.org/de>



Lichtspielhaus war. Und ich glaube, es gibt nur in Bonn ein ehemaliges Priesterseminar, das gleichzeitig Hotel und Seniorenresidenz ist, in dem auch noch Tagungen stattfinden. Diese Kombination führte zu einer sehr speziellen Stimmung.

65. Mitgliederversammlung von DANTE e.V.

Die Mitgliederversammlung fand im wunderschönen Kapitelsaal statt, der allerdings nicht mehr als die ca. 35 anwesenden Mitglieder hätte aufnehmen können. Die Ergebnisse der Versammlung werden im Protokoll des Schriftführers nachzulesen sein. Mir bleibt in Erinnerung, dass die Versammlung relativ gelassen abließ in Anbetracht dessen, dass aufgrund des Berichts der Rechnungsprüfer die Entlastung des Vorstandes verschoben wurde. Die Wahl des neuen Vorstandes ergab das zu erwartende Ergebnis – alle, die sich aufstellen ließen, wurden auch gewählt.



Die Auszählung der abgegebenen Stimmzettel.

TUG 23

Am meisten habe ich mich in diesen Tagen auf und über all die alten Bekannten gefreut, die ich endlich wieder getroffen habe. Und zwar sowohl die aus Deutschland als auch die, die ich bei internationalen Tagungen kennengelernt habe. Und ich habe natürlich neue Kontakte geknüpft – so habe ich unter anderem den Organisator der diesjährigen BayT_EX und neuen Beisitzer im Vorstand Oliver Rath, mit dem ich mich zuvor per E-Mail ausgetauscht habe, nun auch persönlich kennengelernt.



Alle, die sich zum Photoshooting einfanden.

Außerdem habe ich im persönlichen Gespräch und durch die Vorträge viele Hinweise, Ideen und neue Ansätze bekommen, wie und was mit $\text{T}_{\text{E}}\text{X}$ et al. zu bewerkstelligen ist. Besonders beeindruckt hat mich der Einsatz der Firma Overleaf, die durch etliche Vorträge auf sich aufmerksam machte. Ihre explizite Nachfrage, Anregungen zur Entwicklung ihrer Software an sie weiterzugeben, konnte ich direkt meinem Sohn übermitteln, der Overleaf mehr oder weniger regelmäßig benutzt (wie ich auf Rückfrage erfahren habe – »Kennst Du eigentlich Overleaf?« »Klar Mama, habe ich Dir doch schon vor ewigen Zeiten erzählt!«).



Teilnehmer, die immer wieder sorgfältig von Paulo Cereda arrangiert wurden.

Ein weiterer, für mich spannender Vortrag war der, in dem Frank Mittelbach über die Entstehung seines eben erschienenen Werks »The $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Companion, third edition« erzählte. Offensichtlich hatte er im Großen ähnliche Probleme und Herausforderungen zu bewältigen, wie ich sie mit jeder Ausgabe der Mitgliederzeitung im Kleinen

habe. Aber obwohl er jetzt offensichtlich geübt ist, ließ er sich nicht als Nachfolger von mir anwerben. Sehr Schade!



Frank Mittelbach bei einem seiner Vorträge.

Neben weiteren Vorträgen, interessanten Pausengesprächen und leckerem Mittagessen trafen sich am Samstagabend viele zu einem überaus edlen und genussvollen Bankett im *Konrad* in der 17. Etage des Marriott Hotels <https://www.marriott.com/de/hotels/cgnbo-bonn-marriott-hotel/dining/>. Begrüßt wurden wir mit dem besten Sekt der Gegend (überhaupt? Zumindest laut Aussage des Servicepersonals) und einem gigantischen Überblick über Bonn von der Terrasse des Restaurants.

Danach durften wir ein delizioses Mahl genießen, in dessen Verlauf ich mit Freude feststellen konnte, dass mein Sohn nicht der Einzige ist, der schon im zarten Alter von einem halben Jahr seine erste T_EX-Tagung erlebt. Der kleine Sohn von Ben Davis von der Firma Overleaf erleichterte die Kontaktaufnahme zumindest in seiner Ecke des Tisches ungemein.





Die Vorspeise eines ausgezeichneten Menüs.

Am Ende der Tagung war mein Eindruck, dass sich erstaunlich viele junge Menschen mit $\text{T}_{\text{E}}\text{X}$ und weiteren Entwicklungen beschäftigen, so dass unser geliebtes Satzsystem mitnichten vom Aussterben bedroht ist. Das Zusammentreffen von Alten und Jungen in Bonn hat hoffentlich etliche neue Kooperationen geschaffen und zwar weltweit, waren doch 12 Nationen vertreten: Brasilien, Tschechien, Frankreich, Ungarn, Indien, Italien, Litauen, Schweden, Niederlande, Großbritannien, USA und natürlich Deutschland.

Ein großer Dank meinerseits geht an Ulrike und Gert Fischer, die diese schönen Tage in Bonn ermöglicht und mit Sicherheit sehr viel Zeit dafür investiert haben!



Gert und Ulrike Fischer (mitte) bei der Übergabe eines kleinen Dankeschön durch Robyn Laakso (rechts) und Boris A Veytsman (links) von der $\text{T}_{\text{E}}\text{X}$ Users Group.

Bretter, die die Welt bedeuten

ΛT_EX und Schulphysik 3: Messwertediagramme

Keno Wehr

Der dritte Teil der Artikelreihe zur Schulphysik erläutert das Zeichnen von Messwertediagrammen mit Hilfe des Pakets `pgfplots`. Außerdem wird auf lineare Regressionen mit Hilfe von `pgfplotstable` eingegangen.

Wie viele Aufgaben aus dem Grafikbereich kann auch das Zeichnen von Messwertediagrammen mit Hilfe unterschiedlicher Pakete bewerkstelligt werden. Dieser Artikel beschränkt sich auf das Paket `pgfplots`, das auf `pgf/TikZ` beruht. Für diese Lösung sprechen die ausgereifte Nutzerschnittstelle und die breite Nutzerbasis, die es bei Problemen einfacher macht, Hilfe zu finden.

Wer `PSTricks` bevorzugt, findet mit `pst-plot` das passende Paket (siehe auch [2]). Für die Freunde von `MetaPost` gibt es ebenfalls ein Paket¹, das von W. Entenmann ausführlich beschrieben wird [1, Kap. 13: Das Graph-Paket].

Die Paketversion festlegen

`pgfplots` hat die Besonderheit, dass standardmäßig nicht die neueste Paketversion geladen wird. Es ist sehr zu empfehlen, eine aktuelle Version ausdrücklich anzufordern, da ansonsten die Einstellungen der veralteten Version 1.2 verwendet werden. Die einfachste Möglichkeit dazu ist der folgende Befehl in der Präambel.

```
\pgfplotsset{compat=newest}
```

Die Paketanleitung rät von der Verwendung des Werts `newest` ab, da sich das Aussehen der Grafik verändern könnte, sobald eine neue Paketversion installiert wurde. Nach meinem Dafürhalten sind bei den elementaren Diagrammen, um die es hier geht, keine wesentlichen Änderungen im Paket zu erwarten, sodass die obige Einstellung keine Probleme verursachen dürfte. Wollte man dem Ratschlag der Anleitung folgen, müsste man die zur Zeit aktuelle Versionsnummer als Wert von `compat` verwenden.

¹ Die Anleitung hierzu ist am besten mit dem Befehl `texdoc -1 mgraph` aufzufinden.

```
\pgfplotsset{compat=1.18}
```

Ein Koordinatensystem anlegen

Koordinatensysteme werden in pgfplots durch die Umgebung `axis` erzeugt, die innerhalb der Umgebung `tikzpicture` stehen muss. Der Umgebung `axis` können zahlreiche Optionen übergeben werden, die das Erscheinungsbild des Koordinatensystems festlegen. Wir legen hier ein Koordinatensystem an, das von 0 bis 25 auf der Rechtsachse und von 0 bis 0,4 auf der Hochachse reicht (Abb. 1).

```
\begin{tikzpicture}
\begin{axis}[xmin=0,xmax=25,ymin=0,ymax=0.4]
\end{axis}
\end{tikzpicture}
```

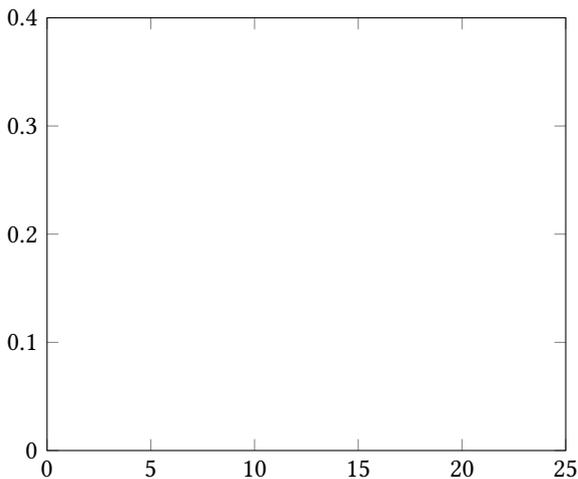


Abb. 1: Koordinatensystem mit Standardeinstellungen von pgfplots

Aus dem Koordinatensystem der Abbildung 1 ergeben sich zunächst zwei Wünsche: Wir hätten gern gewöhnliche Koordinaten mit Pfeilspitzen anstelle des Rahmens. Hierfür gibt es die Option `axis lines=middle`.

Außerdem brauchen wir ein Dezimalkomma, keinen `-punkt`. Im ersten Teil dieser Artikelreihe wurde gezeigt, wie sich für `siunitx` ein Dezimalkomma einstellen lässt [3, S. 9]. Diese Einstellung ist für unseren Zweck nutzlos, da `pgfplots` das

Paket `siunitx` nicht verwendet. Stattdessen braucht man eine recht unhandliche `pgf`-Option, wie im folgenden Codebeispiel aufgeführt. Für die Zukunft ist zu wünschen, dass eine grundlegende Einstellung wie für das Dezimaltrennzeichen vom Sprachpaket vorgenommen wird.

Die Optionen übergeben wir dem Befehl `\pgfplotsset` in der Präambel, sodass sie für alle Koordinatensysteme im Dokument gültig sind. Mit ihnen ergibt sich die Darstellung in Abbildung 2.

```
\pgfplotsset{compat=newest,
  axis lines=middle,
  ticklabel style={/pgf/number format/use comma}}
```

```
\begin{tikzpicture}
\begin{axis}[xmin=0,xmax=25,ymin=0,ymax=0.4]
\end{axis}
\end{tikzpicture}
```

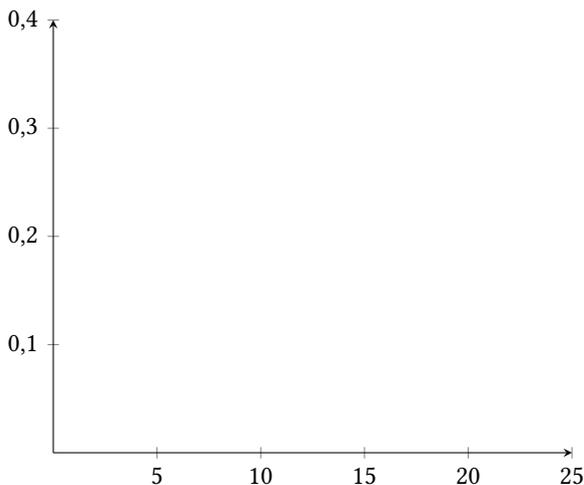


Abb. 2: Koordinatensystem mit Koordinatenachsen und Dezimalkomma

Nun zeigt sich, dass ein runder Wert für das Maximum der beiden Skalen keine gute Wahl war, denn so fällt ein Skalenstrich genau auf die Pfeilspitze am Ende der Achse. Wir vergrößern daher beide Maximalwerte etwas.

Sehr wichtig für ein Diagramm sind natürlich auch die Angabe der Messgröße und der Maßeinheit. Wir nehmen an, dass ein Kraft-Weg-Diagramm mit dem Weg in

Zentimetern auf der Rechtsachse und der Kraft in Newton auf der Hochachse zu zeichnen ist und ergänzen entsprechende Achsenbeschriftungen. Außerdem fügen wir mit `grid=major` ein Koordinatengitter hinzu (Abb. 3).

```
\begin{tikzpicture}
\begin{axis}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s in cm},
ylabel={F in N},grid=major]
\end{axis}
\end{tikzpicture}
```

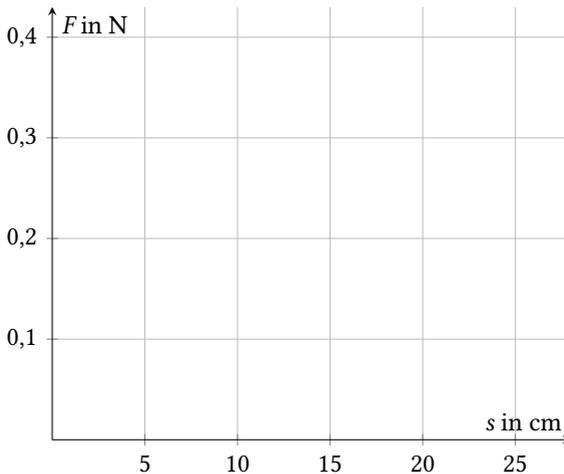


Abb. 3: Koordinatensystem mit Achsenbeschriftungen und Koordinatengitter

Durch das Gitternetz sind die Skalenstriche nun etwas unscheinbar geworden. Wir ergänzen die Präambel um die Option `tick style=thick` und machen sie auf diese Weise breiter.

Die Beschriftungen von Abbildungen erfolgen üblicherweise in etwas kleinerer Schrift als der Haupttext [1, S. 24]. Wir versehen die Achsenbeschriftung mit dem Schriftgrößenbefehl `\small` und die Skalenbeschriftung mit `\footnotesize`.

Wem der Achsenschnittpunkt als Markierung des Koordinatenursprungs nicht deutlich genug ist, der mag noch eine 0, die standardmäßig nicht angezeigt wird, zu einer der Skalen ergänzen wollen. Dies kann für die Rechtsachse mit der Option `hide obscured x ticks=false` geschehen (Abb. 4).

```
\pgfplotsset{compat=newest,
  axis lines=middle,
  tick style=thick,
  label style={font=\small},
  ticklabel style={/pgf/number format/use comma,font=\footnotesize}}
```

```
\begin{tikzpicture}
\begin{axis}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s in cm},
  ylabel={F in N},hide obscured x ticks=false,grid=major]
\end{axis}
\end{tikzpicture}
```

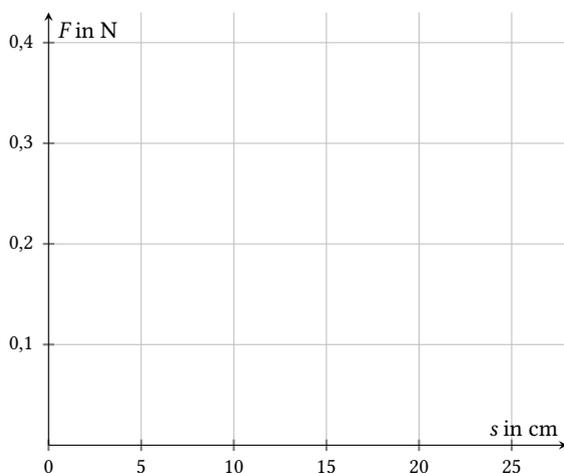


Abb. 4: Koordinatensystem mit kleinerer Beschriftung und Markierung der 0

Das Koordinatensystem ist nun vorbereitet, um mit Messpunkten bevölkert zu werden. Wie dies geschehen kann, wird im folgenden Abschnitt erläutert.

Schüler sollten Messwertediagramme vorzugsweise komplett selbst zeichnen, um einen Blick dafür zu bekommen, worauf es dabei ankommt. In Unterrichtsreihen, in denen viele Diagramme zu zeichnen sind, wird man ihnen aber durch vorgegebene Koordinatensysteme entgegenkommen. Für solche Fälle ist es vorteilhaft, das vertraute 5-Millimeter-Raster erzeugen zu können.

Um in ein solches Raster zu kommen, wählen wir auf der Rechtsachse den Maßstab $1\text{ cm} \hat{=} 5\text{ cm}$ und auf der Hochachse $1\text{ cm} \hat{=} 0,1\text{ N}$. `pgfplots` bietet zur Festlegung des Maßstabs die Optionen `x` und `y` an, denen die Länge übergeben wird, die jeweils

einer Koordinateneinheit der dargestellten Größe entspricht. Wir müssen daher in unserem Fall $x=2\text{mm}$ und $y=10\text{cm}$ wählen. Außerdem ändern wir die Option `grid=major` in `grid=both` und ergänzen `minor tick num=1`, um ein Nebengitternetz mit *einer* Nebengitternetzlinie zwischen zwei Hauptgitternetzlinien zu erhalten (Abb. 5).

```
\begin{tikzpicture}
\begin{axis}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s in cm},
ylabel={F in N},hide obscured x ticks=false,grid=both,minor tick num=1,
x=2mm,y=10cm]
\end{axis}
\end{tikzpicture}
```

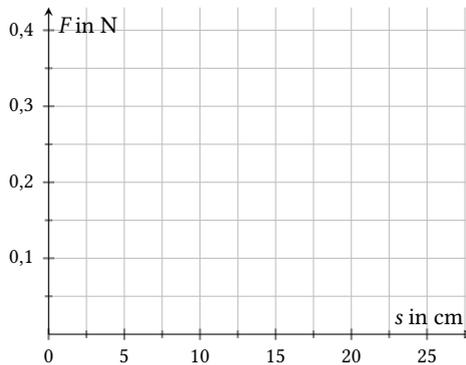


Abb. 5: Koordinatensystem mit 5-Millimeter-Raster

Das Koordinatensystem ist nun recht klein geraten, was der Größe dieser Zeitschrift geschuldet ist. Bei Ausdruck auf A4-Papier würde man eher $x=5\text{mm}$ und $y=20\text{cm}$ wählen.

Wie wir sehen, müssen sehr viele Optionen gesetzt werden, um das gewünschte Ergebnis zu erhalten. Das Paket `schulma` aus dem Schulmathematik-Bündel stellt eine Reihe häufig benötigter Optionen ein und definiert für Koordinatensysteme die auf `axis` beruhende Umgebung `Kosy`, die von selbst eine `tikzpicture`-Umgebung einfügt. Hierdurch lässt sich der Code vereinfachen; insbesondere können alle oben für die Präambel angeführten Codezeilen wegfallen (Abb. 6).

```
\usepackage{schulma}
```

```
\begin{Kosy}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s in cm},
  ylabel={F in N},hide obscured x ticks=false,x=2mm,y=10cm]
\end{Kosy}
```

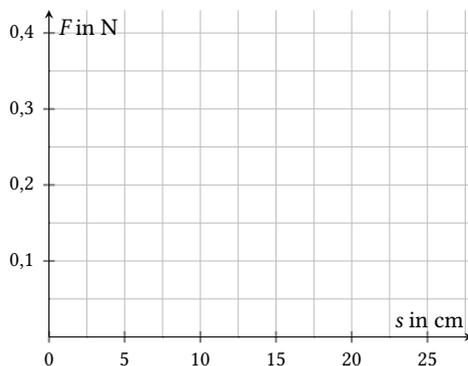


Abb. 6: Koordinatensystem mit `schulma`

Der einzige Unterschied zwischen Abb. 5 und Abb. 6 ist der Wegfall der Skalenstriche der Nebenskala. Dies bewerkstelligt `schulma` mit der Option `minor tick length=0pt`.

Wir verwenden ab hier für alle Koordinatensysteme die Umgebung `Kosy`. Wenn das Paket `schulma` nicht zum Einsatz kommt, lässt sich das gezeigte Vorgehen leicht auf die Umgebung `axis` übertragen.

Wenn man den Schwierigkeitsgrad etwas erhöhen will, kann man die Skalenbeschriftung entfernen, sodass sich die Schüler eigene Gedanken über einen passenden Maßstab machen müssen (Abb. 7).

```
\begin{Kosy}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s in cm},
  ylabel={F in N},hide obscured x ticks=false,xticklabels=\empty,
  yticklabels=\empty,x=2mm,y=10cm]
\end{Kosy}
```

Messpunkte darstellen

Wir tragen nun Messpunkte, wie sie sich aus einer typischen Messreihe mit einer Schraubenfeder ergeben könnten, in das Koordinatensystem aus Abbildung 6 ein.

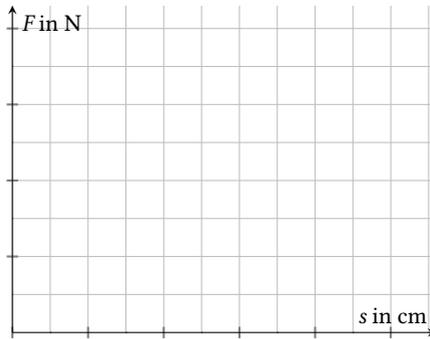


Abb. 7: Koordinatensystem mit leeren Skalen

Dazu verwenden wir den Befehl `\addplot coordinates`, dem eine Liste von Punkten übergeben wird. Bei diesem wie bei allen `\addplot`-Befehlen ist es wichtig, an das Semikolon am Ende zu denken (Abb. 8).

```
\begin{Kosy}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s$ in cm},
  ylabel={F$ in N},hide obscured x ticks=false,x=2mm,y=10cm]
\addplot coordinates {(5,0.07)(10,0.14)(15,0.23)(20,0.3)(25,0.39)};
\end{Kosy}
```

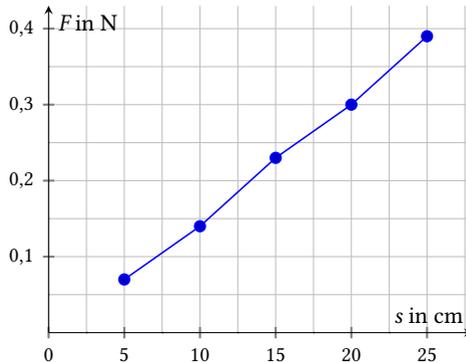


Abb. 8: Koordinatensystem mit Messpunkten

Das Ergebnis ist so noch nicht annehmbar. Eine Zickzacklinie durch die Messpunkte verbieten wir unseren Schülern streng, die wollen wir auch in unserem Musterdiagramm nicht haben. Stattdessen werden wir später eine Ausgleichs-

Regressionsgerade einzeichnen. Außerdem ist eine dezentere Markierung der Messpunkte wünschenswert. Wir ersetzen `\addplot` durch `\addplot+[Optionen]`, um geeignete Optionen für die Messpunkte zu wählen (Abb. 9).

```
\begin{Kosy}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s in cm},
  ylabel={F in N},hide obscured x ticks=false,x=2mm,y=10cm]
\addplot+[only marks,mark=x] coordinates {(5,0.07)(10,0.14)(15,0.23)(20,0.3)
  ↪(25,0.39)};
\end{Kosy}
```

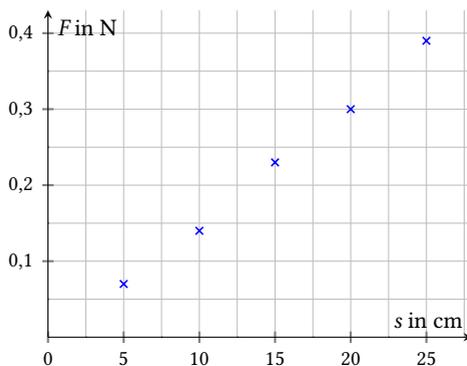


Abb. 9: Koordinatensystem mit unverbundenen Messpunkten

Schließlich wäre es sinnvoll, die Messpunkte etwas zu vergrößern, was sich mit der Option `mark size` erreichen lässt. Eine zweite Messreihe im gleichen Koordinatensystem darzustellen, fällt nicht schwer; wir müssen nur einen weiteren `\addplot`-Befehl ergänzen und die Achsenskalierung anpassen (Abb. 10).

```
\begin{Kosy}[xmin=0,xmax=28,ymin=0,ymax=0.88,xlabel={s in cm},
  ylabel={F in N},hide obscured x ticks=false,x=2mm,y=5cm,mark size=2.5pt]
\addplot+[only marks,mark=x] coordinates {(5,0.07)(10,0.14)(15,0.23)(20,0.3)
  ↪(25,0.39)};
\addplot+[only marks,mark=x] coordinates {(5,0.17)(10,0.32)(15,0.51)(20,0.66)
  ↪(25,0.81)};
\end{Kosy}
```

Eine Ausgleichsgerade zeichnen

Wenn man zusätzlich zu `pgfplots` das Ergänzungspaket `pgfplotstable` lädt, ist es möglich, sich eine Regressionsgerade zu zeichnen und ihre Steigung anzeigen zu

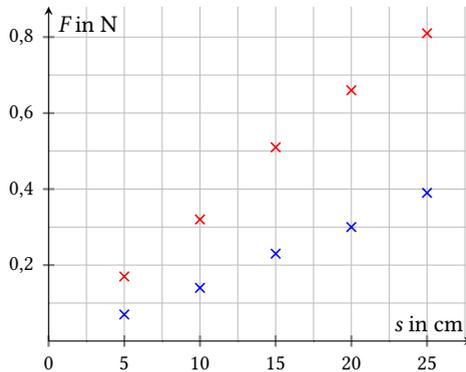


Abb. 10: Koordinatensystem mit zwei Messreihen

lassen. Hierzu ist es nötig, zusätzlich zur Eingabe der Messpunkte noch eine Tabelle mit den gleichen Messdaten einzugeben, für die dann die Regression durchgeführt wird. Die entsprechende Syntax ist dem folgenden Codebeispiel zu entnehmen. Wenn man die Messdaten in eine externe Datei auslagerte, könnte man die Doppelingabe vermeiden², was angesichts der in der Schulphysik üblicherweise kurzen Messreihen aber übertrieben wäre.

Die Geradensteigung wird nicht automatisch ausgegeben, sondern im Makro `\pgfplotstableregressiona` abgelegt. Wir geben den Wert mit Hilfe eines TikZ-Knotens an einer freien Stelle im Koordinatensystem aus (Abb. 11).

```
\usepackage{schulma}
\usepackage{pgfplotstable}
```

```
\begin{Kosy}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={s in cm},
  ylabel={F in N},hide obscured x ticks=false,x=2mm,y=10cm,mark size=2.5pt]
\addplot+[only marks,mark=x] coordinates {(5,0.07)(10,0.14)(15,0.23)(20,0.3)
  \c-(25,0.39)};
\addplot[red] table[ $y=\text{create col/linear regression}=\{y=F\}$ ]{
  s F
  5 0.07
  10 0.14
  15 0.23
  20 0.3
  25 0.39
};
```

² Ein Beispiel hierzu ist in der `pgfplots`-Anleitung im Abschnitt »Fitting Lines – Regression« zu finden.

```
\node[red] at (20,0.1) {\pgfplotstableregressiona};
\end{Kosy}
```

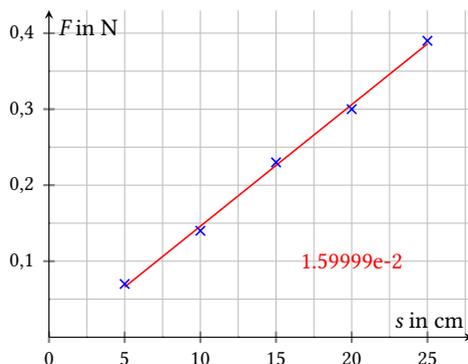


Abb. 11: von pgfplots berechnete Regressionsgerade mit Steigung

Die Regressionsgerade wird leider nicht über die gesamte Breite des Koordinatensystems durchgezogen, sondern nur zwischen dem ersten und letzten Messpunkt. Dieses Problem können wir etwas abmildern, indem wir das Wertepaar $(0|0)$ zur Tabelle hinzufügen (was die berechnete Steigung natürlich minimal verändert). Außerdem wollen wir die Steigung angemessen formatieren und eine Maßeinheit ergänzen, wofür wir den Befehl `\tqty` aus dem Paket `schulma-physik` [3, S. 14] in Verbindung mit geeigneten `siunitx`-Optionen einsetzen (Abb. 12).

```
\usepackage{schulma}
\usepackage{schulma-physik}
\usepackage{pgfplotstable}
```

```
\node[red] at (20,0.1) {
  \setup{exponent-mode=fixed,round-mode=figures,round-precision=3}%
  $\frac{\Delta F}{\Delta s}=\tqty{\pgfplotstableregressiona}{\N\per\cm}$
};
```

Die Durchführung der Regression mit `pgfplotstable` ist recht umständlich, auch überzeugt das Ergebnis nicht völlig: Die Regressionsgerade bricht beim x -Wert des letzten Messpunktes ab; aus didaktischen Gründen wäre eine Ursprungsgerade wünschenswert, die aber aufgrund des verwendeten Regressionsverfahrens nicht erzwungen werden kann. In der Abbildung 12 mag der Unterschied zur Ursprungsgeraden vernachlässigbar sein, aber bei anderen Messreihen könnte die Abweichung

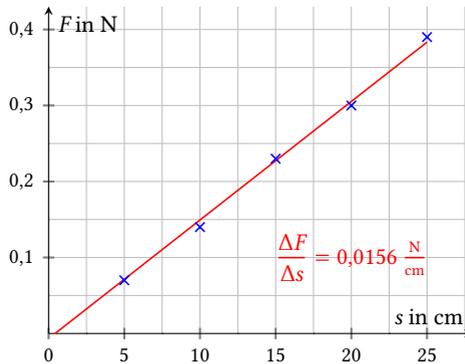


Abb. 12: Regressionsgerade mit formatierter Ausgabe der Steigung

deutlicher ausfallen. Zudem bietet pgfplots für nichtlineare Graphenverläufe gar keine Regressionsbefehle.

In vielen Fällen wird man daher eine andere Lösung wählen und die Gleichung der Ausgleichsgeraden oder -kurve mit externen Rechenwerkzeugen bestimmen, sofern sie nicht ohnehin bekannt ist, oder im Falle einer Ursprungsgeraden die Steigung einfach nach Augenmaß festlegen. Dann reduziert sich das Problem für pgfplots auf die Darstellung eines Funktionsgraphen mit explizit angegebenem Funktionsterm (Abb. 13).

```
\begin{Kosy}[xmin=0,xmax=28,ymin=0,ymax=0.43,xlabel={\$s\$ in cm},
  ylabel={\$F\$ in N},hide obscured x ticks=false,x=2mm,y=10cm,mark size=2.5pt]
\addplot+[only marks,mark=x] coordinates {(5,0.07)(10,0.14)(15,0.23)(20,0.3)
  \c-(25,0.39)};
\addplot[red,domain=0:28] {0.015*x};
\end{Kosy}
```

Mit einem TikZ-\draw-Befehl innerhalb der Umgebung Kosy bzw. axis lässt sich problemlos ein Steigungsdreieck ergänzen, das die zeichnerische Ermittlung der Geradensteigung demonstriert (Abb. 14).

```
\draw[red,semithick,dashed] (10,0.15)--(25,0.15)
  node[midway,below] {\$\Delta s=\qty{15}{\cm}\$}
  --(25,0.375) node[left,near start] {\$\Delta F=\qty{0,225}{\N}\$};
```

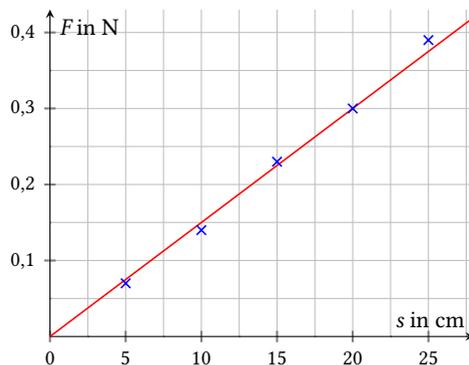


Abb. 13: Ursprungsgerade mit manuell festgelegter Gleichung

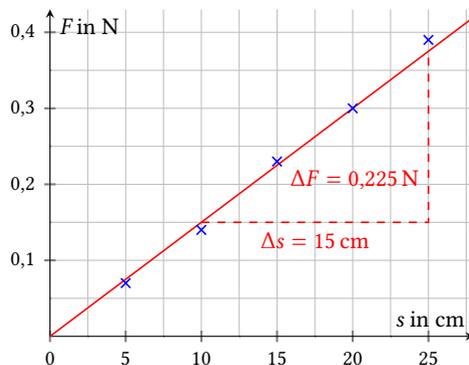


Abb. 14: Ursprungsgerade mit Steigungsdreieck

Eine Ausgleichskurve zeichnen

Bei nichtlinearen Zusammenhängen muss die Regressionsfunktion auf jeden Fall manuell eingegeben werden. Als Beispiel dient eine Messreihe zur Schwingungsdauer eines Federpendels in Abhängigkeit von der Masse, wobei der Zusammenhang durch eine Quadratwurzelfunktion beschrieben wird (Abb. 15).

```
\begin{Kosy}[xmin=0,xmax=0.28,ymin=0,ymax=1.9,xlabel={m$ in kg},
  ylabel={T$ in s},hide obscured x ticks=false,x=20cm,y=2cm,mark size=2.5pt]
\addplot+[only marks,mark=x] coordinates {(0.05,0.81)(0.1,1.13)(0.15,1.36)
  \c(0.2,1.55)(0.25,1.74)};
\addplot[red,domain=0:0.28,samples=200] {3.53*sqrt(x)};
```

```
\end{Kosy}
```

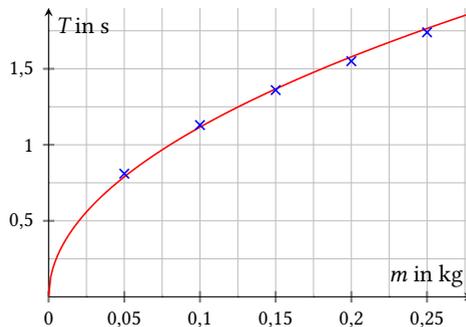


Abb. 15: Messreihe mit Ausgleichskurve

pgfplots zeichnet einen Funktionsgraphen durch Berechnen der Funktionswerte an bestimmten Stützstellen und lineares Verbinden der entsprechenden Punkte. Dies führt bei nichtlinearen Funktionen und zu geringer Anzahl der Stützstellen (voreingestellt sind 25) zu zackigen Graphenverläufen. Daher wurde beim Zeichnen des Wurzelgraphen die Option `samples=200` verwendet. Von der alternativen Option `smooth` ist abzuraten, da sie oft nicht zu befriedigenden Ergebnissen führt.

Die in diesem Artikel vorgestellten Beispiele dürften einen Großteil der Anforderungen des schulischen Physikunterrichts abdecken, schöpfen die Möglichkeiten des Pakets pgfplots aber bei weitem nicht aus, wie sich mit Blick auf den Umfang der Paketanleitung von z. Zt. 573 Seiten erahnen lässt. Nicht dargestellt wurden hier beispielsweise logarithmisch skalierte Achsen und dreidimensionale Graphen.

Literatur

- [1] Walter Entenmann: MetaPost, Grafik für T_EX und L^AT_EX, Berlin, 2017.
- [2] Laura E. Jackson, Herbert Voß: »Die Plot-Funktionen von pst-plot«, *DTK*, 14.2 (2002), 27–33.
- [3] Keno Wehr: »L^AT_EX und Schulphysik 1: Größen und Einheiten«, *DTK*, 35.1 (2023), 7–16.

LaTeX per pandoc mit Markdown füttern

Christian Böttger

Der Beitrag befasst sich mit der Anpassung eines Fundstücks von GitHub an die persönlichen Anforderungen des Autors. Dabei werden Markdown-Dokumente (mit YAML-Header) mittels pandoc [3] über eine LaTeX-Vorlage zu ansehnlichen PDF-Dateien kompiliert.

Ausgangspunkt

Vorwegstellen möchte ich, dass meine LaTeX-Erfahrungen nur eine sehr kurze Zeitspanne umfassen und ein ständiges »learning by doing« sind, weit weg von schönen Lösungen oder gar elegantem Code. Ich freue mich noch immer über jeden fehlerfreien Kompilerlauf...

Anfang der Neunziger Jahre kam ich im Rahmen des Studiums erstmals in Kontakt mit LaTeX. Wochenlanger »Enthusiasmus« (vor allem für die Literaturverwaltung für rechtswissenschaftliche Studiengänge), begrenzter Zugang zum Internet und ein noch nicht erfundenes StackOverflow [4] führten dazu, dass die Studienarbeiten schließlich nicht per LaTeX gesetzt wurden, sondern mit AmiPro unter OS/2. Rückblickend betrachtet, war das die falsche Entscheidung.

Im Zuge der starken Zunahme mobiler Arbeit und des Einsatzes verschiedener Geräteklassen (Laptop, Smartphone/Tablet) hierfür, wurde für mich immer wichtiger, portable Lösungen, erst für Notizen, dann mehr und mehr auch für fertige Arbeitsergebnisse, einzusetzen.

Markdown

In einem ersten Schritt habe ich die Erstellung von Notizen und Entwürfen sowie E-Mails auf das Markdown-Format umgestellt.

Markdown, entwickelt von John Gruber und Aaron Schwarz, ist eine vereinfachte Auszeichnungssprache [6], die sich grundsätzlich mit jedem beliebigen Texteditor schreiben lässt. Für angenehmes Arbeiten (vor allem auf Mobilgeräten mit kleineren Bildschirmen und virtuellen Tastaturen) gibt es aber auch spezialisierte Editoren, die zum Beispiel die Vervollständigung von Auszeichnungselementen unterstützen.

Für Einsteiger praktisch: Es werden oft auch die Ergebnisse der Auszeichnung von Text direkt angezeigt. Überschriften sind größer, hervorgehobener Text wird **fett** dargestellt und meist lässt sich auch eine Befehlsreferenz aufrufen.

Beispiele für solche Editoren sind Joplin und Byword (Letzteres gibt es leider nur für iOS, iPad OS und macOS).

Viele Anforderungen an die oben genannten Dokumententypen lassen sich einfach mit Markdown umsetzen. Es gibt strukturierte Überschriftenebenen, Hervorhebungen, Zitate und vieles mehr. Hier ein einfaches Beispiel:

```
# Überschrift der Ebene 1 oder Titel

Hier folgt ein Absatz mit einfachem Text ohne
Besonderheiten, bis auf eine fett hervorgehobene
oder kursiv gesetzte Wortfolge.

## Überschrift der Ebene 2

Aufzählungen gibt es entweder nummeriert:

1. Listenpunkt 1
2. Listenpunkt 2

oder auch einfach:

- ein einfacher Aufzählungspunkt
- noch ein weiterer Aufzählungspunkt.
```

Ohne die Syntax von Markdown im Einzelnen zu erläutern, hier kurz die Erklärungen zum Beispiel:

- Überschriften werden mittels Doppelkreuz # erzeugt, die Anzahl bestimmt die Ebene von 1 bis 6,
- Listen werden entweder mit Zahlen für nummerierte oder mit Zeichen für nicht nummerierte Aufzählungspunkte erzeugt,
- Hervorhebungen erfolgen mittels ** für Dickdruck und * für kursiven Druck.

Wird ein solcher kurzer Text (hier im Beispiel mit dem Standard-CSS von GitHub) gerendert, ergibt sich ein Bild wie in Abbildung 1.

Eine vollständige Darstellung der Möglichkeiten, die Markdown als Auszeichnungssprache bietet, würden den Umfang dieses Artikels sprengen. Zwar gibt es mittlerweile auch eine Vielzahl an Markdown-»Dialekten«, die auch Fußnoten, Tabellen oder farbige Text hervorhebungen (highlighting) unterstützen. Ausgangspunkt ist jedoch nach wie vor die bereits seit 2004 spezifizierte Urversion, die es auf der Homepage von John Gruber (daringfireball.net [2]) zum Nachlesen gibt.

Überschrift der Ebene 1 oder Titel

Hier folgt ein Absatz mit einfachem Text ohne Besonderheiten, bis auf eine **fett hervorgehobene** oder *kursiv gesetzte* Wortfolge.

Überschrift der Ebene 2

Aufzählungen gibt es entweder nummeriert:

1. Listepunkt 1
2. Listepunkt 2

oder auch einfach:

- ein einfacher Aufzählungspunkt
- noch ein weiterer Aufzählungspunkt.

Abb. 1: Markdown gerendert mit GitHub-CSS

Vorstellungen vom Ergebnis

Die mittels Markdown erzielbaren Ergebnisse sind für die persönliche Anwendung und die rein elektronische Korrespondenz per E-Mail (wenn das E-Mail-Programm die automatische Umwandlung unterstützt) vollständig ausreichend.

Anders verhält es sich mit Korrespondenz, die in Form von Briefen oder Berichten versendet werden soll. Hier sind die Möglichkeiten aus Anwendersicht stark begrenzt. Wem die so gebotenen Ergebnisse nicht ausreichen, sucht nach Lösungen. So auch ich.

Fündig wurde ich schließlich auf GitHub. Dort gibt es zahlreiche Projekte, die sich im Wesentlichen mit dem Schreiben von wissenschaftlichen Werken, insbesondere auch Dissertationen, mittels Markdown befassen. Ausgangspunkt ist dort meist R-Markdown, ein Dialekt, der von der Sprache R (mit der große Datenmengen ausgewertet werden können) unterstützt wird.

Einen weniger spezialisierten Ansatz benutzt Mattia Tezzele [1]. Sein Projekt geht von einer Markdown-Datei aus, die mit zusätzlichen Metadaten – entweder mittels eines YAML-Kopfes innerhalb der Markdown-Datei oder über eine separate Metadaten-datei (`settings.yaml`) – angereichert wird. Diese Metadaten enthalten zum Beispiel Absender und Empfänger eines Briefes, deren Adressen sowie Be-

treffzeilen. Ebenfalls gesteuert werden können damit Schriftarten, Abstände oder Papiergrößen.

Umsetzung

Dieser Ansatz hat mich auf verschiedene Ideen zu Ergänzungen und Erweiterungen bis hin zur Automatisierung der Dokumentenerstellung gebracht.

Diese Metadaten, die (fast) beliebig erweiterbar sind, können dann über Variablen in eine LaTeX-Datei, die als Vorlage (Template) verwendet wird, eingefügt werden. So sind dann individuell gestaltete Briefköpfe oder Adressfeldeinträge etc. realisierbar. Ein solcher Metadaten-Vorspann (YAML header) in einer Markdown-Datei sieht beispielhaft wie folgt aus:

```
---
title: 'Titel des Dokuments'
author: Ich bin Autor
date: 2023-06-01
---
```

Mehr zu YAML und dem Aufbau solcher Kopfdaten gibt es auch in der pandoc-Dokumentation. [5]

Vorteil dieser Methode aus meiner Sicht ist, dass vor allem wenig geübte Anwender:innen schöne PDF-Dokumente per LaTeX erstellen können, ohne in die damit meist verbundene Komplexität des Dokumentenaufbaus eintauchen zu müssen. Auch wird dadurch eine gewisse Automation vereinfacht.

Um sinnvoll mit dieser Methode einen Brief zu schreiben, bedarf es jedoch etwas mehr. Nachfolgend ein Beispiel mit Angaben für einen einfachen Brief:

```
---
author: 'Berta Briefsender'
city: Absenderhausen
from:
- Poststraße 1
- 12345 Sendelfingen
to:
- Hr. Max Mustermann
- Wegenallee 1
- 54321 Musterhausen

# Subject / Betreff
subject:
- Ihr Schreiben von damals
```

```
# Signature
sig:
  include: true
  # sign: true
  name: 'Berta Briefsender'
  post: Die Absenderin
  # secondpost:

# Settings
mainfont: Open Sans
altfont: Hoefler Text
monofont: Monaco
language: de
fontsize: 10pt # For fonts like Open Sans use 10pt.
geometry: a4paper, left=30mm, right=20mm, top=25mm, bottom=20mm
letterhead: true
# customdate: YYYY-MM-DD
# draft: true
drafttext: ENTWURF # Custom text.
draftsize: '144pt' # Lower size when using longer draft texts.
---
```

Das Beispiel ist bereits etwas komplexer aufgebaut und entspricht – im Wesentlichen – den bei mir in der Praxis eingesetzten Metadaten. Es enthält folgende Ergänzungen:

- Wörter am Zeilenanfang, die mit einem Doppelpunkt abschließen, bezeichnen eine Variable, die später in der Vorlage zur Verfügung steht.
- Doppelkreuze (#) dienen zur Einleitung eines Kommentars oder kommentieren aktuell nicht genutzte Einstellungen (z. B. ein Entwurf-Wasserzeichen) aus.
- Spiegelstriche (praktisch: Markdown-Aufzählungen) werden als array angesehen, durch das später in der Vorlage iteriert wird. So sind mehrzeilige Adressfelder oder Betreffzeilen möglich.
- Variablen, die mit true (oder auch false) befüllt werden, können in der L^AT_EX-Vorlage später abgefragt und für Verzweigungen in if-Schleifen verwendet werden.

Darüber hinaus werden mit `mainfont`, `altfont` und `monofont` Schriftarten definiert, die später per X_YL^AT_EX als Schriftdefinitionen zum Rendern verwendet werden sollen. Sollen nicht die im System installierten, sondern z. B. L^AT_EX-Standardschriften verwendet werden, müssen die Pakete entsprechend in der Template-Datei definiert werden. Wer mehr über die Verwendung von Schriften unter X_YL^AT_EX erfahren möchte, dem sei der lesenswerte Beitrag »Schriften für X_YL^AT_EX und LuaL^AT_EX« von Herbert Voß in Ausgabe 1/2022 von „Die T_EXnische Komödie“ ans Herz gelegt.

Weitere Variablen betreffen Schriftgröße (`fontsize`), Seitengeometrie (`geometry`) und die (Haupt-)Sprache (`language`) des späteren Dokuments.

Über die Variable `letterhead: true` steuert die von mir verwendete Vorlage, ob überhaupt ein Briefkopf verwendet werden soll. Dieser wird in der Praxis – noch – mit einer PDF-Datei als Hintergrund über das Paket `wallpaper` eingebunden. Zukünftige Versionen sollen die von `koma-script` bereitgestellten Klassen `scrletter` bzw. `scr1ttr2` und deren Briefkopfoptionen nutzen. Das ist zum jetzigen Zeitpunkt aber noch eine Baustelle, die gegebenenfalls einen eigenen Beitrag füllen dürfte.

Schlussendlich wird mit `drafttext` und `draftsize` über das Paket `draftwatermark` die aus Textverarbeitungen bekannten Wasserzeichen (Begriffe wie »Entwurf« oder »vertraulich«) eingestellt. Größe und Text sind hier beliebig anpassbar. Da das Paket `draftwatermark` auch Farben unterstützt, ich aber praktisch nur Verwendung für das einfache hellgraue Wasserzeichen habe, sind die Farben in der Vorlage hartkodiert und nicht über eine Variable aus der Markdown-Datei (bzw. der `settings.yaml` heraus) anzusprechen. Möglich wäre das aber ohne Weiteres.

Inhalt und Form vereinen

Die Datei `template.tex` beinhaltet die »Magie« in Form eines L^AT_EX-Dokuments, das als Vorlage bzw. Template verwendet wird. Diese Vorlagendatei ist eigentlich nichts anderes als eine sorgfältig konstruierte L^AT_EX-Datei, die aber keine (oder nur sehr wenige) Inhalte enthält. Lediglich die Form und die Formatierung des Dokuments werden darin definiert. Der Inhalt, also der zu setzende Text, wird über die Markdown-Datei importiert.

Dazu werden entsprechende Platzhalter und Variablen im L^AT_EX-Dokument verwendet, die dann mit Inhalten befüllt werden. Im einfachsten Fall enthält die Vorlage lediglich ein `$body$`, das benötigt wird, um in diese Variable den gesamten Text »hineinfließen« zu lassen.

Ein etwas vereinfachtes Beispiel einer Vorlagendatei, das wieder dem Projekt von Mattia Tezzele[1] entlehnt ist, könnte wie folgt aussehen:

```
%!TEX TS-program = xelatex
%!TEX encoding = UTF-8 Unicode

\documentclass[$fontsize$, a4paper]{scr1ttr2}
\usepackage{fontspec}

\usepackage{geometry}
\geometry{$geometry$}
```

```

\pagenumbering{arabic}

\usepackage[document]{ragged2e}

\if(letterhead)$
\usepackage{wallpaper}
\ThisULCornerWallPaper{1}{letterhead-front.pdf}
% \ULCornerWallPaper{1}{letterhead.pdf}
\endif$

\if(draft)$
% \usepackage{wallpaper}
\usepackage[color={gray}{0.85}, text=$drafttext$, fontsize=$draftsize$]{
  ↪draftwatermark}
% \ULCornerWallPaper{1}{draft.pdf}
\endif$

\if(language)$
\usepackage{polyglossia}
\setmainlanguage{language$}
\endif$

\usepackage{xunicode}
\usepackage{xltextra}
\usepackage[protrusion=true, final]{microtype}

\defaultfontfeatures{Mapping=tex-text}
\setmainfont [Ligatures={Common}, Numbers={OldStyle}]{mainfont$}
\setsansfont[Scale=0.9]{altfont$}
\setmonofont[Scale=0.8]{monofont$}

\setlength{\parindent}{0pt}
\setlength{\parskip}{1em plus 2pt minus 1pt}
\setlength{\emergencystretch}{3em} % Prevent overfull lines

\newcommand{\ampers}{\fontspec[Scale=.95]{mainfont$}\selectfont\itshape\&}

\if(mainfont)$
  \setmainfont{mainfont$}
\endif$
\if(altfont)$
  \setsansfont{altfont$}
\endif$

```

```

\providecommand{\tightlist}{%
  \setlength{\itemsep}{0pt}\setlength{\parskip}{0pt}}

$if(customdate)$
\usepackage[$language]{datetime2}
\DTMsavedate{customdate}{$customdate$}
$endif$

% DOCUMENT
\begin{document}
\thispagestyle{empty}
\sffamily \scriptsize
\underline{
\textsc{\textbf{$author$}}
$for(from)$
\textbullet{} \textsc{$from$}
$endfor$}

\vspace{1em}

\rmfamily \normalsize
$for(to)$
$to$\
$endfor$

\vspace{13em}

\sffamily
\begin{flushright}
  $city$, $if(customdate)$\DTMusedate{customdate}$else$\today$endif$
  \
\end{flushright}

\rmfamily

$if(subject)$
  $for(subject)$
    \textbf{$subject$}\
  $endfor$
$endif$

\vspace{3em}

$body$

```

```

$if(sig.include)$
  \begin{minipage}[t]{6.5in}
    $if(sig.sign)$
      \includegraphics[height=1in]{signature.pdf}
    $else$
      \includegraphics[height=1em]{empty.pdf}
    $endif$
    \\
    \vspace{-0.2in}$sig.name$\\
    $sig.post$
    $if(sig.secondpost)$
      \\
      $sig.secondpost$
    $endif$
  \end{minipage}
$endif$

\end{document}

```

Das Beispiel ist bereinigt, vor allem um zahlreiche Kommentare und in der Praxis notwendige vertikale Abstände, die mittels `vspace`-Anweisungen gesetzt werden. Viele dieser Abstandsangaben sind durch den als PDF-Datei eingebunden Briefkopf bedingt, um die Empfängeradresse, das Datum oder die Betreffzeile zu positionieren. Auch entspricht das Beispiel weder einer DIN-Norm, noch wird die Ästhetik des L^AT_EX-Satzes von Briefen anderer Pakete gewahrt (was hier jedoch teilweise beabsichtigt ist).

Da der PDF-Briefkopf keine Absenderangabe für Fensterumschläge enthält, werden diese aus den Angaben in der Markdow-Datei mit gesetzt. Zukünftig soll der PDF-Briefkopf zudem ersetzt werden. Das Paket `koma-script` ist ohnehin im Einsatz und für die später geplante Serienbrieffunktion ebenfalls verwendbar. (Bislang habe ich keine andere Möglichkeit gefunden, Serienbriefe mit dem hier verwendeten Ansatz zu erhalten, die dem Briefkopf der Textverarbeitung der Kollegen entsprechen.)

Nach dem eigentlichen Text setzt die L^AT_EX-Vorlage noch eine Unterschriftenzeile. Im Beispiel wird je nach zur Verfügung gestellter Signaturdatei (Namensschriftzug) im PDF-Format auch eine »Unterschrift« gesetzt.

Dokumentenkonverter pandoc

Nach langer Vorrede kommen wir nun zum Dreh- und Angelpunkt, mit dem aus dem Markdown-Ausgangsdokument und einer L^AT_EX-Vorlagendatei schließlich unser schönes PDF gerendert wird: `pandoc`.

Völlig unspektakulär ersetzt pandoc – etwas vereinfacht dargestellt – einen Aufruf von `pdflatex quelle.tex` oder `xelatex quelle.tex`.

Die vollständige Befehlszeile lautet:

```
pandoc -o ziel.pdf --pdf-engine=xelatex --template=template.tex
-- quelle.md
```

(Hinweis: Der im Beispiel vorhandene Zeilenumbruch ist im Terminal nicht mit einzugeben.)

X_YL^AT_EX nutze ich, wie bereits erläutert, um einfacher die im System vorhandenen Schriften einsetzen zu können.

Genau wie bei anderen L^AT_EX-Dokumenten kann ein mehrfacher Durchlauf erforderlich sein, wenn Seitenzahlen, Gliederungspunkte oder andere Zähler erstellt oder aktualisiert werden sollen. In diesem Fall wird pandoc mit der kompletten Befehlszeile mehrfach ausgeführt.

Ausblick und Todos

Auch wenn ich das vorgestellte System im produktiven Einsatz habe, sind noch viele Verbesserungen möglich (und zum Teil auch erforderlich). Aus Zeitgründen und dem altbekannten »Läuft doch auch so« habe ich an etlichen Stellen aufgehört, weil mein gewünschtes Ergebnis erreicht wurde. Für eine allgemeine Verwendung hat das Projekt aber auf jeden Fall noch genügend Erweiterungspotenzial.

Einige davon sind:

- make-Datei zur Vereinfachung des Kompileraufrufs
- Originalgetreue Briefkopfgestaltung ohne PDF-Hintergrund
- Verwendung von pandoc-Filtern, z. B. für Querverweise
- Erstellen von Serienbriefen

Einer oder mehrere dieser Punkte werden vielleicht in einem Folgeartikel wieder aufgegriffen. Ansonsten ist dies eine gute Möglichkeit, »neue« Anwender an L^AT_EX heranzuführen. Das gleiche System wird übrigens produktiv genutzt, um sehr schnell ansehnliche Präsentationen mit dem Paket `beamer` zu erstellen.

Literatur

- [1] Mattia Tezzele (mrzool): <https://github.com/mrzool/letter-boilerplate>.
- [2] John Gruber: <https://daringfireball.net/projects/markdown/>.
- [3] <https://pandoc.org>.
- [4] <https://stackoverflow.com>.

[5] pandoc.org: https://pandoc.org/MANUAL.html#extension-yaml_metadata_block.

[6] Wikipedia: <https://de.wikipedia.org/wiki/Markdown>.

Erfahrungsbericht zum Einsatz von T_EX im Alltag eines Gemeindepfarrers

Pfarrer HG Unckell

Vor Jahren hatte ich in der T_EXnischen Komödie von einem Pfarrer einen Erfahrungsbericht über die Nutzung von T_EX wahrgenommen. Damals konnte ich den Hinweis zu Lilypond, einem Notensatzprogramm, welches gut zu T_EX passt, für mich nutzen.

Persönlich nutze ich T_EX seit gut 35 Jahren, am Anfang meist für private Zwecke bzw. das Engagement in einem Ehrenamt. In dieser Zeit war ich als Systemingenieur bei einem großen Hersteller in der Qualitätssicherung tätig. Die Erfahrung, dass der Umstieg von einem Apple II auch mit der Herausforderung verbunden war, persönliche Daten zu migrieren, führte mich damals zu T_EX als einem Programm, welches auf unterschiedlichen Plattformen verfügbar ist.

Während der Umschulung, die ein weiteres Studium und mehrfache Referendariatszeiten umfasste, entstanden weitere Dokumente, die ich noch heute, 30 Jahre später gut nutzen kann.

Diese Seite eines offenen Standards lohnt sich, finde ich, immer wieder ins Gespräch zu bringen. So kann ich auf Texte zurückgreifen, die auf einem Atari ST, einem Windows 3.1 Laptop, Mac OS 9, Mac OS X mit den unterschiedlichen Versionsnummern oder auf Linux entstanden sind.

T_EX ist ein Traum

T_EX ist ein Traum, so äußerte sich ein jüngerer Mann bei einem Gespräch auf dem Kirchentag in Nürnberg, wo ich für den Verein LUKi (Linux User in den Kirchen) am Stand aktiv war. Einstieg war dabei, wie in vielen anderen Gesprächen, die Frage gewesen, was denn konkret im Alltag einer Gemeinde an freier/offener Software genutzt wird.

Im Anschluss an den Kirchentag habe ich zwei Beispiele meiner Nutzung kurz in eigenen Blog-Beiträgen auf der Homepage von <https://LUKi.org> beschrieben.

Diese Beispiele wollen ermutigen, Möglichkeiten von T_EX zu nutzen. Gleichzeitig ist es nichts Spektakuläres, d. h. für alle, die mit T_EX vertraut sind, ist es als Anregung gedacht, diese Möglichkeit bei Gelegenheit einzubringen, als eine Weise, wie man gut, Bestehendes wieder benutzen kann.

Nachbau eines geschenkten Designs

Mein erster Artikel beschreibt mein Vorgehen in der Begleitung eines Familienzentrums, welches in einer Gemeinde, die ich geleitet hatte, vom Status eines Projekts in den Regelbetrieb übernommen wurde.

Für die Öffentlichkeitsarbeit dieses Zentrums hatte eine Schule für Gestaltung der Gemeinde ein Design geschenkt, welches mit einer professionellen Software erstellt worden war. Dieses Dokument musste nun halbjährlich fortgeschrieben werden. Dafür sich eine Lizenz zu erwerben, wäre für die Gemeinde unverhältnismäßig gewesen. Als Alternative stand im Raum, diese Fortschreibung an eine Firma zu vergeben, oder das Design mit Hilfe von T_EX nachzubauen. Mehr dazu ist hier beschrieben: <https://luki.org/2023/06/welche-nutzungs-moeglichkeiten-haben-freie-programme/>.



Im Gemeindealltag gab es solche Gelegenheiten öfter. Ein ähnliches Beispiel wäre der Gemeindebrief gewesen, der 3- bis 4-mal jährlich veröffentlicht wurde.

Besondere Gestaltung von Texten

Mein zweiter Artikel beschreibt mein Vorgehen bei der besonderen Gestaltung von Texten, wie z. B. Bibelversen, die als Jahreslosung bewusster wahrgenommen werden sollten. Ähnlich kann man bei Einladungen oder besonderen Grüßen vorgehen. Ich nenne diese Gestaltung oft Kalligraphie, auch wenn sie mit Hilfe von T_EX entsteht, also in einem strengen Verständnis dieses Begriffes anders bezeichnet werden müsste.

In gewisser Weise geht es auch hier wieder um das Wiederbenutzung von Gestaltungsideen, wie einem Herz, einem Kreis oder freierer Formen. Über den Blogartikel sind die Quellen zu den Kalligraphien zugänglich, d. h. man kann damit schnell auch eigene Akzente setzen.

Durch die online-Nutzung von Overleaf sogar, ohne eine Systeminstallation vorab gemacht haben zu müssen. Mehr dazu hier: <https://luki.org/2023/07/graphische-gestaltung-von-texten-kalligraphie-mit-tex/>



ConT_EXt kurz notiert

Henning Hraban Ramm

In der ConT_EXt-Welt ist vieles in Bewegung, das eine Erwähnung wert ist, aber keinen ganzen Artikel begründet. (Stand: 12. 7. 2023)

Module

Gegenüber einer Vielzahl von L^AT_EX-Paketen gibt es nur eine Handvoll ConT_EXt-Module. Das Installations-Skript für MkIV konnte auch Module installieren, seit LMTX hat diese Möglichkeit aber gefehlt, und wir mussten uns mit einer rsync-basierten Lösung behelfen. Doch Anfang Mai hat Hans Hagen geliefert:

```
mtxrun --script install-modules --list
mtxrun --script install-modules --install tikz simpleslides
mtxrun --script install-modules --install --all
```

PGF/TikZ wird wegen seines Umfangs in einen eigenen TDS-Baum `texmf-tikz` installiert und Bestandteile werden weggelassen, die für ConT_EXt nicht relevant sind.

Ich bin inzwischen für die Modulverwaltung zuständig und in Zusammenarbeit mit Taco Hoekwater dabei, die Verwaltungs-Website zu verbessern – die Skripte sind bald 20 Jahre alt... Wir arbeiten auch daran, veraltete Module zu aktualisieren und an die TDS anzupassen.

Kaktovik-Zahlensystem

Kaktovik (Iñupiaq: Qaaktuġvik) ist ein Ort mit etwa 280 Einwohner*innen auf einer Insel vor der Nordküste Alaskas. Die neun Schüler*innen der Zwergschule entwickelten 1994 Zeichen für die Zahlen ihrer Muttersprache Iñupiaq, mit denen das Vigesimalssystem¹ der Eskimo-Aläutischen Sprachen abgebildet werden kann. Sie setzten sich folgende Ziele:

1. Optisch einfach: Die Symbole sollten leicht zu merken sein.
2. Ikonisch: Es sollte eine klare Beziehung zwischen Symbol und Bedeutung geben.

¹ Zahlensystem mit Basis 20

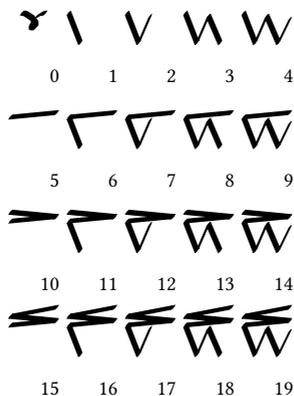


Abb. 1: Kaktovik-Ziffern

3. Effizient: Die Symbole sollten leicht zu schreiben sein, ohne den Stift abzusetzen.
4. Unverwechselbar: Sie sollten sich deutlich von den arabischen Zahlen unterscheiden, um Verwechslungen zu vermeiden.
5. Ästhetisch: Gut aussehen sollten sie auch noch.

Typisch für Vigesimalssysteme gibt es eine Subbasis 5: Mit (mehr oder weniger) senkrechten Strichen wird bis 4 gezählt, dann wird ein waagerechter Strich darüber gelegt, dann wieder senkrecht bis 9, ein zweiter waagerechter Strich für die 10, usw. Ab 20 gibt es eine weitere Stelle. Im Inñupiaq gibt es kein Wort für Null; das Zeichen soll an gekreuzte Arme erinnern, um anzudeuten, dass nichts gezählt wird.

Ein Effekt, mit dem die Schüler*innen nicht gerechnet hatten: Arithmetik wird mit dem System sehr klar und verständlich.² Die Mathe-Noten an der Schule verbesserten sich deutlich. Das neue Zahlensystem wurde von den Ureinwohner-Gremien unterstützt und verbreitete sich schnell. Im September 2022 wurden die Kaktovik-Zahlen in den Unicode Standard (15.0) aufgenommen.

Im April 2023 stieß die ConT_EXt-Community zufällig auf einen Beitrag zu dem Thema, und Hans Hagen lieferte sofort die ConT_EXt-Unterstützung. (Bis zum Setzen eines Schulbuchs auf Inñupiaq fehlen allerdings noch ein paar Schritte.)

² Mehr darüber steht z.B. in der englischsprachigen Wikipedia, https://en.wikipedia.org/wiki/Kaktovik_numerals.

Dezimal	Kaktovik
$2 + 2 = 4$	$\rightarrow \vee + \vee = \mathbb{W}$
$17 - 7 = 10$	$\rightarrow \overline{\vee} - \vee = \overline{\vee}$
$1503/3 = 501$	$\rightarrow \mathbb{W} \overline{\vee} \vee / \vee = \vee \overline{\vee}$
$364/3 = 121, \text{ Rest } 1$	$\rightarrow \overline{\mathbb{W}} \mathbb{W} / \vee = \overline{\vee} \vee \text{ Rest } \vee$

Abb. 2: Einfache Arithmetik mit Kaktovik-Zahlen

Spaß mit Einheiten

Nachdem wir bereits 2021 die Unterstützung für Donald Knuths Potrzebie (als dk) eingeführt³ und zum 1. April 2023 die Unterstützung für imperiale Einheiten offiziell abgeschafft haben⁴, hat Hans Hagen beim Bacht_EX-Treffen neue Einheiten auf der Grundlage von Körpermaßen der Teilnehmer*innen ergänzt: Edith (es) ersetzt das Zoll, Tove (ts) ist ein Zehntel davon, und die »European Unit« (eu) hat den Standardwert von 1 es, lässt sich aber durch Änderung von `\eufactor` flexibel anpassen. Und weil T_EX zwar eine Fehlermeldung über die maximale Länge von 19 Fuß enthält, aber keine Fuß (ft) als Maßeinheit kennt, wurde aufgrund der kleinsten anwesenden Schuhgröße 1 Theodore = 5 Edith definiert.⁵ Kaiser Arthur (»emperor of hyphenation and protector of the patterns«) verteilte an die Beteiligten Mützen mit der Aufschrift »Make T_EX little again«. In diesem Zusammenhang sei an die Zitate »It's a mess!« und »Wer misst, misst Mist« erinnert.

In der folgenden Diskussion auf der Mailingliste kritisierte Hans Hagen die unschöne Definition von 1 Fuß = 19 Zoll; unter Berücksichtigung des Amsterdamer Fußes seien 20 Zoll passender, was wiederum zum Kaktovik-Zahlensystem passt, das wir zur allgemeinen Verwendung propagieren.

Fließkommazahlen

Bekanntlich rechnet T_EX grundsätzlich mit ganzzahligen Dimensionen in *scaled point*. Das hat nicht nur historische Gründe, sondern bietet auch eine gleichbleibende Genauigkeit – im Gegensatz zu Fließkommazahlen, die entweder große Zahlen oder viele Nachkommastellen ermöglichen. Die winzigen Rundungsfehler der Integer-Berechnungen können sich aber aufschaukeln, und Hans Hagen wollte gerne ausprobieren, ob sich Mikrotypographie und Umbruch durch Fließkomma-Berechnungen verbessern lassen. Daher kennt LuaMetaT_EX nun native *floats* (in diesem Fall keine

³ Artikel in TUGboat Nr. 42, <https://www.tug.org/TUGboat/tb42-3/tb132hagen-dk.pdf>

⁴ siehe »ConT_EXt kurz notiert« in DTK 2/2023

⁵ Edith ist die Tochter von Mikael Sundqvist, Theodore der Sohn von Arthur Rosendahl.

Gleitobjekte), die sich ähnlich wie Dimensionen (`\newdimen`, `\newfloat`) verwenden lassen.

Mehrstufiger Umbruch

Als »Abfallprodukt« des verbesserten Formelumbruchs lässt sich der Absatzumbruch mit Strafpunkten (Penaltys, Demerits), Zeichendehnung (HZ), Satzdichte (Looseness) und hängender Interpunktion (Protrusion) jetzt mehrstufig einstellen (*multi-pass parbuilder*), so dass ConT_EXt zuerst strengere Einstellungen versucht und sie dann schrittweise lockert, bis sich »ideale« Absätze ergeben. Geeignete Parameter werden noch experimentell geprüft.

Der mehrstufige Umbruch wird derzeit mit `\setupalign[granular]` aktiviert, soll aber das Standardverfahren werden. Mit `\setupalignpass` kann eine benannte Konfiguration aktiviert werden; derzeit gibt es `decent`, `quality` sowie `test1` bis `test5` – Hans Hagen wartet auf Rückmeldungen, welche Konfigurationen sich bewährt haben. Die Hintergründe und Parameter sind in `lowlevel-lines` beschrieben.

Nachdem Hans Hagen Anfang Juli angekündigt hat, sich jetzt im *page builder mode* zu befinden, warten wir gespannt auf Verbesserungen im vertikalen Ausgleich, bei `\testpage` und Fehlerverfolgung. (Dabei hat VZ nichts mit früheren sozialen Netzwerken zu tun...)

Dokumentation

Ein Werk, dessen Anfänge der Autor 2016 unter dem Arbeitstitel »Typesetting with ConT_EXt for scientists, engineers, academics and others« vorgestellt hat, befindet sich inzwischen in der Schlusskorrektur – allerdings in einer französischen Version; die englische Übersetzung soll folgen.

Mein Buch »Drucksachen gestalten mit ConT_EXt und LuaMetaT_EX« hat mittlerweile über 410 Seiten und einen Umschlag, aber auch immer noch viele Lücken; ich mache mir wieder einmal Hoffnungen, es »bald« in der DANTE-Reihe veröffentlichen zu können.

Von fremden Bühnen

LaTeX News – Issue 37, Juni 2023¹

Frank Mittelbach

Neue Funktionalität als Teil des »LaTeX Tagged PDF«-Projekts

Wir haben nun neue automatische Tagging-Funktionalität für zusätzliche LaTeX-Elemente hinzugefügt – darunter die meisten Anzeigeumgebungen, Standard-Abschnittsbefehle, Inhalts-, Abbildungs- und Tabellenverzeichnisse, Floats und Grafiken sowie Bibliographien. Dies lässt sich durch

```
\DocumentMetadata{testphase=phase-III}
```

aktivieren. Aktuell steht Tagging nur für einen eingeschränkten Satz an Dokumenten zur Verfügung – solche mit den Standard-Dokumentenklassen (`article`, `report` und `book`) und nur beim Einsatz von Befehlen und Umgebungen, die in Lamports LaTeX-Manual beschrieben sind.

Der Einsatz anderer Dokumentenklassen oder das Hinzufügen zusätzlicher Pakete in der Präambel kann funktionieren (zumindest teilweise), aber aktuell ist das eher unwahrscheinlich – zumindest für Pakete und Klassen, die die Interna von LaTeX umfangreich verändern.

Beachten Sie auch, dass es immer noch einige im LaTeX-Manual beschriebene Umgebungen und Befehle gibt, für die noch keine Tagging-Unterstützung vorhanden ist – unter anderem `Tabellen`, `tabbing`, die diversen Mathematik-Umgebungen. Diese werden als Teil von `phase-III` Unterstützung erhalten, aber manche von ihnen erst nach dem Juni-Release.

Ein Prototyp für Mathematik-Tagging (einschließlich der Unterstützung für die `amsmath`-Umgebungen) ist schon verfügbar, aber vor allem zum Experimentieren und für Rückmeldungen gedacht. Das sich daraus ergebende Tagging entspricht definitiv noch nicht dem, was wir letztendlich anstreben (auch wenn es »korrektes« Tagging erzeugt). Wenn Sie es ausprobieren wollen, nutzen Sie folgende Zeile:

```
\DocumentMetadata{testphase={phase-III,math}}
```

¹ Der Newsletter wurde zuvor im *TUGboat* 44:2 veröffentlicht [3] und von Thomas Demmig übersetzt.

Beachten Sie, dass der Code zum Mathematik-Tagging aktuell mit Paketen kollidiert, die das $\$$ -Zeichen umdefinieren (was dann zu seltsamen Fehlern führen kann), und dass Pakete, die den Mathematik-Modus für nicht-mathematische Konstrukte verwenden, in Bezug auf Tagging überraschende Ergebnisse liefern können. Feedback dazu, welche Pakete mit dem Code auf die eine oder andere Art und Weise Probleme haben, ist sehr willkommen.

Das `latex-lab`-Bundle enthält eine Reihe von (noch nicht getaggten) Dokumentations-Dateien über den neuen Code, auf die Sie mit `texdoc -l latex-lab` zugreifen können.

Feedback ist willkommen! Für diesen Zweck gibt es neues Repository in dem Probleme (und Lösungen) von Benutzern eingestellt und Ideen diskutiert werden können. Die URL lautet <https://github.com/latex3/tagging-project>.

Neue oder verbesserte Befehle

Hooks um Argumente erweitern

Hooks waren schon immer Container für Code, dessen Ergebnis vollständig vom Inhalt des Hooks abhängt. Wenn irgendeine Form kontextbezogener Information an den Hook übergeben werden musste, hatte das durch das Setzen einer Variable vor dem Hook zu geschehen, so dass der Code im Hook diese nutzen konnte. Aber das lässt sich nur schwer überblicken, ist schwierig zu implementieren und beim Programmieren brauchte man eine Art »Hook vor dem Hook«, um dieses Setup umsetzen zu können.

Um das Ganze etwas einfacher zu machen, wurde `lthooks` so erweitert, dass Hooks mit Argumenten unterstützt werden. Hooks können nun mit Argumenten deklariert und eingesetzt werden – der Code für diese Hooks kann die Hook-Argumente über `#1`, `#2` und so weiter referenzieren, so dass sich Hooks dann ähnlich wie Makros und nicht wie *Token-Listen* verhalten (in `exp13`-Terminologie). Normale Hooks ohne Argumente funktionieren weiterhin genauso wie zuvor: Diese Erweiterung ist vollständig zu älteren Dokumenten und Paketen kompatibel.

Um einen Hook mit Argumenten zu deklarieren, nutzen Sie

```
\NewHookWithArguments {<hook>} {<num-args>}
```

Um den Code im Hook zu verwenden, schreiben Sie (bei zwei Argumenten)

```
\UseHookWithArguments {<hook>} {2} {<arg1>} {<arg2>}
```

Wollen Sie einem Hook Code hinzufügen, der Argumente erwartet, schreiben Sie

```
\AddToHookWithArguments {<hook>} [<label>] {<code>}
```

Das ist das Gleiche wie bei normalen Hooks, nur dass $\langle code \rangle$ die Argumente nutzen kann, indem er #1, #2 usw. nutzt. Wollen Sie in diesem Fall ein tatsächliches Parameter-Token (#) zu $\langle code \rangle$ hinzufügen, müssen Sie es wie üblich verdoppeln.

Wollen Sie zudem »normalen« Code zu einem Hook mit Argumenten hinzufügen, können Sie weiterhin `\AddToHook` nutzen – in diesem Fall werden #-Token wie bisher *nicht* verdoppelt. Das bedeutet, dass sich ein Paketautor dafür entscheiden kann, Argumente zu bestehenden Hooks hinzuzufügen, ohne sich um Kompatibilität Sorgen machen zu müssen: `\AddToHook` wird alles richtig machen und die neu hinzugefügten Argumente nicht fälschlicherweise referenzieren.

Die Befehle `\NewReversedHookWithArguments`, `\NewMirroredHookPairWithArguments`, `\AddToHookNextWithArguments`, `\UseOneTimeHookWithArguments` und die `expl3`-Gegenstücke der in diesem Abschnitt besprochenen Befehle wurden ebenfalls hinzugefügt. Die vollständige Dokumentation finden Sie in der Dokumentation von `lthooks` [1].

Generische `cmd`-Hooks mit Argumenten

Neben der eben vorgestellten Möglichkeit, Argumente an einen regulären Hook zu übergeben, können nun generische `cmd`-Hooks auf die Argumente des Befehls zugreifen, in den sie gepatcht sind, wobei die im vorigen Abschnitt beschriebene Schnittstelle zum Einsatz kommt.

Wollen Sie beispielsweise mit Hilfe von Hooks Code zum `\title`-Befehl hinzufügen, können Sie auf den im Argument übergebenen Titel zugreifen. Um also beispielsweise den Titel des Dokuments im Terminal auszugeben, könnten Sie schreiben:

```
\AddToHookWithArguments{cmd/title/before}
  {\typeout{Dokumenttitel: #1}}
```

Wie bei regulären Hooks kann Code, der zu einem `cmd`-Hook mit Hilfe von `\AddToHook` hinzugefügt wird, nicht auf die Argumente des Befehls zugreifen. Das bedeutet, dass diese Änderung wie bei regulären Hooks vollständig abwärtskompatibel ist, daher werden ältere Verwendungen von `cmd`-Hooks genau wie zuvor funktionieren.

Kopier- und Anzeigefunktionen für Umgebungen bereitstellen

Um eine Befehlsdefinition zu kopieren, haben wir 2022 den Befehl `\NewCommandCopy` eingeführt. Damit können Sie sogar Befehle kopieren, die aus diversen internen Komponenten bestehen, wie zum Beispiel robusten Befehlen oder solchen mit einer komplexen Signatur. Um das gleiche mit Umgebungen erzeugen zu können, zum

Beispiel zu definieren, dass die Umgebung `myitemize` äquivalent zu `itemize` sein soll, können Sie nun schreiben:

```
\NewEnvironmentCopy{myitemize}{itemize}
```

Es gibt auch `\Renew...` und `\Declare...`, die abhängig von den Umständen nützlich sein können.

Zudem bieten wir den Befehl `\ShowEnvironment` an, der den `\begin-` und `\end-`Code der als Argument übergebenen Umgebung ausgibt. So führt beispielsweise `\ShowEnvironment{center}` zu folgender Ausgabe:

```
>> \begin{center}=environment:
>> ->>\trivlist \centering \item \relax .
<<recently read>> }
1. ... \ShowEnvironment{center}
>> \end{center}:
>> ->>\endtrivlist .
<<recently read>> }
1. ... \ShowEnvironment{center}
```

(Github Issue 963)

`\IfFileAtLeastTF`

Das L^AT_EX-Release 2020-10-01 hat die CamelCase-Tests `\IfClassAtLeastTF` und `\IfPackageAtLeastTF` zum Prüfen von Klassen- und Paketdaten mitgebracht. Wir haben nun auch noch `\IfFileAtLeastTF` hinzugefügt, um das gleiche für generische Dateien zu ermöglichen, die eine `\ProvidesFile`-Deklaration enthalten.

(Github Issue 1015)

`\DeclareLowercaseMapping`, `\DeclareTitlecaseMapping` und
`\DeclareUppercaseMapping`

Der Wechsel von einem Ansatz zum Transformieren zwischen Groß- und Kleinschreibung mit `\lccode-` und `\uccode-`Daten zu einem, bei dem die Informationen durch eine vom Kernel verwaltete Struktur gespeichert werden, hat eine Lücke für Anwender hinterlassen, die Ergebnisse des Wechsels von Groß- und Kleinschreibung *anzupassen*. Das wurde nun durch das Hinzufügen der drei folgenden Befehle behoben:

- `\DeclareLowercaseMapping`
- `\DeclareTitlecaseMapping`
- `\DeclareUppercaseMapping`

Diese lassen sich nutzen, um das Ergebnis für einzelne Codepoints anzupassen. Das lässt sich allgemein oder für ein spezifisches Locale umsetzen (siehe auch den nächsten Abschnitt). Eine kleine Zahl vordefinierter Konfigurationen wurde im Kernel eingerichtet, bei denen die Ausgabe für pdf_TE_X sich von denen für Unicode-Engines unterscheiden sollen. So erlaubt beispielsweise

```
\DeclareUppercaseMapping{"01F0}{\v{J}}
```

das Erzeugen von Ĵ in 8-Bit-Engines: Ohne diese Anpassung würde es zu einem Fehler kommen, weil es in Unicode kein vordefiniertes Ĵ gibt. Mehr Details dazu finden Sie im usrguide. *(Github Issue 1033)*

\BCPdata

Verbesserungen im Unicode-Handling für das Ändern von Groß- und Kleinschreibung haben deutlich gemacht, dass der Kernel bisher Locale nicht berücksichtigt hat. Die Pakete `babel` und `polyglossia` bieten eine umfassende Locale-Unterstützung, aber es gibt keine einheitliche Schnittstelle, um diese Informationen an anderen Code weiterzugeben. Nach Gesprächen mit den Betreuern dieser beiden Bundles definiert der Kernel nun `\BCPdata` als Stub (so dass es immer definiert ist) und `babel` und `polyglossia` redefinieren es, um die Locale-Daten bereitzustellen. Ein gemeinsamer Satz von Schlüsselwörtern bedeutet, dass `\BCPdata` sowohl vom Kernel wie auch von anderen »Consumer«-Paketten strukturiert abgefragt werden kann.

(Github Issue 1035)

Verbessertes `\samepage`

Die `\samepage`-Deklaration setzt diverse Parameter auf `10000`, um unerwünschte Seitenumbrüche zu verhindern. Der Parameter `\predisplaypenalty` hat standardmäßig immer einen Wert von `10000` und um Platz zu sparen, wurde er daher in der Vergangenheit nie explizit gesetzt. Aber es gibt ein paar Klassen, die den Parameter ändern und Benutzer sehen sich dann im Geltungsbereich von `\samepage` eventuell einem Seitenumbruch vor einer abgesetzten Formel gegenüber, wenn solche Klassen zum Einsatz kommen. Das wurde nun korrigiert und `\predisplaypenalty` wurde ebenfalls auf `10000` gesetzt. *(Github Issue 1022)*

Gruppen in `\MakeUppercase`

Vor 2022 haben `\MakeUppercase` und `\MakeLowercase` eine Gruppe mit geschweiften Klammern um ihre Argumente genutzt, um einen Scope für jegliche Deklarationen im Argument zu bieten. Dieses Gruppieren wurde wiederhergestellt (auch für

`\MakeTitlecase`), obwohl die zugrundeliegenden L3-Befehle für Groß- und Kleinschreibung keine Gruppen einsetzen. (Github Issue 1021)

Erweiterungen des `\label`-Befehls

In Standard-L^AT_EX hat der `\label`-Befehl eine `\newlabel`-Deklaration in die `.aux`-Datei geschrieben und zwei Werte im zweiten Argument dieses `\newlabel`-Befehls übergeben: `\@currentlabel` enthält normalerweise den Status des aktuellen Zählers und `\thepage` für die aktuelle Seitenzahl.

Die Pakete `hyperref` und `nameref` haben dann den `\label`-Befehl gepatcht, um stattdessen fünf Werte zu übergeben. Dazu haben sie zusätzlich `\@currentlabelname` gesichert, was normalerweise den aktuellen Titeltext enthält und über `\nameref` ausgelesen werden kann, und `\@currentHref`, das den Namen des Ziels enthält, um einen aktiven Link erzeugen zu können. Das fünfte Argument wurde nur genutzt, wenn externe Referenzen mit dem Paket `xr-hyper` geladen wurden.

Ab diesem Release wurde die Zahl der Werte, die in `\newlabel` abgelegt werden, vereinheitlicht. `\label` schreibt nun einen `\newlabel`-Befehl, der immer fünf Werte im zweiten Argument enthält (jedes in einer Gruppe aus geschweiften Klammern): `\@currentlabel`, `\thepage`, `\@currentlabelname`, `\@currentHref` und `\@kernelreserved@label@data` (welches für den Kernel reserviert ist).

Zudem wurde ein Hook mit dem Namen `label` hinzugefügt. Er besitzt ein Argument: den Label-Namen, also das Argument von `\label`. Code, der zum Hook hinzugefügt wird, kann auf dieses Argument über `#1` zugreifen. Der Hook wird direkt vor dem Schreiben durch den `\label`-Befehl in die `.aux`-Datei ausgeführt, aber *nachdem* der `\@bshpack`-Befehl seine Spacing-Magie ausgeführt hat. Er befindet sich *innerhalb* einer Gruppe, so dass sein Code nur die Schreiboperation beeinflusst.

Codeverbesserungen

Performance beim Prüfen von Dateiendungen

Das Hinzufügen von Hooks und Ähnlichem zu Dateioperationen hat den Nebeneffekt, dass mehrfach geprüft wird, ob die Datei existiert. In größeren Dokumenten mit vielen Dateien haben diese Dateisystemoperationen nicht-triviale Auswirkungen auf die Performance gehabt. Wir cachen nun die Existenz von Dateien, so dass diese wiederholten Dateisystemaufrufe vermieden werden.

doc: _ korrekt im Index verarbeiten

Aufgrund einiger Probleme im Code war es nicht möglich, zu verhindern, dass _ im Index erscheint – \DoNotIndex{_}, usw. hatte keinen Effekt. Das wurde nun korrigiert. *(Github Issue 943)*

doc: Unterstützung für das upquote-Paket

Die Standardzeichen für Apostroph und Gravis bei Schreibmaschinenschriften sind typographische Zeichen, so wird beispielsweise die Eingabe

```
\verb/`prog 'my input'`/
```

je nach verwendeter Schriftart gerendert als ‘prog ’my input’ ‘ und nicht als `prog 'my input'`, wie dies bei der Programmierung oft bevorzugt wird.

Das lässt sich beispielsweise mit dem Paket upquote anpassen, was dann zur zweiten Ausgabe führt. Aber aus historischen Gründen hatte doc seine eigene Definition von \verb und verbatim, weshalb die beiden Pakete nicht zusammenarbeiten konnten. Dies wurde nun behoben und das Laden von upquote zusammen mit doc hat den gewünschten Effekt. *(Github Issue 953)*

Standarddefinition für \do

Der Befehl \do mit seinem netten öffentlichen Namen ist in Wirklichkeit ein interner Befehl, der von Plain-T_EX für die Listenverarbeitung übernommen wurde. Aber er erhielt erst dann eine Definition, wenn \begin{document} ausgeführt wurde, was dazu führte, dass eine Benutzerdefinition in der Präambel immer an dieser Stelle überschrieben wurde. Um den Benutzer ordentlich darüber zu informieren, dass dieser Befehl nicht frei verfügbar ist, haben wir nun eine Definition im Format vorgenommen, so dass \newcommand & Co. eine saubere Fehlermeldung erzeugen, statt eine Definition zuzulassen, die nicht bestehen bleibt. *(Github Issue 975)*

Neuer Schlüssel für filecontents

Die filecontents-Umgebung warnt am Terminal, wenn eine Datei überschrieben wird – auch wenn das mit Absicht geschieht, zum Beispiel beim wiederholten Schreiben einer temporären Datei. Um die Warnung in diesem Fall weniger störend zu gestalten, haben wir einen neuen Schlüssel nowarn hinzugefügt, der die Overwrite-Warnung in die Transkript-Datei umleitet. Unserer Meinung nach sollte die Aktion trotzdem irgendwo dokumentiert werden, um beim Debuggen helfen zu können, weshalb sie nicht vollständig unterdrückt wird. Die Warnung, dass nichts

geschrieben wird, weil die Datei schon existiert (und force nicht genutzt wurde), ist unverändert und wird weiterhin am Terminal ausgegeben. (*Github Issue 958*)

Ein weiterer Hook bei der Ausgabe von Seiten

Seit Oktober 2020 bietet der Seitenausgabe-Prozess eine Reihe von Hooks, um Einfluss auf das Verhalten vor, während und nach `\shipout` nehmen zu können. So ist es beispielsweise Paketen mit dem Hook `shipout/before` möglich, Code zurückzusetzen, den sie verändert haben (zum Beispiel `\catcode` bei einer Verbatim-ähnlichen Verarbeitung), und mit `shipout/background` und `shipout/foreground` kann Material zusätzlich auf den Seiten erscheinen. Details dazu finden Sie in [2].

Es fehlte aber bisher ein Hook, der es einem Paket-Schreiber erlaubt, die komplette Seite (mit Vordergrund und Hintergrund) kurz vor der eigentlichen Ausgabe zu beeinflussen. Dafür bieten wir nun den zusätzlichen Hook `shipout` an. Ein Anwendungsfall (manchmal bei der Druckproduktion erforderlich) ist das Spiegeln der gesamten Seite per `\reflectbox` einschließlich aller zusätzlichen Daten, die im Vorder- oder Hintergrund hinzugefügt wurden. (*Github Issue 920*)

Release-Informationen in der .log-Datei ausgeben

L^AT_EX gibt seine Release-Informationen ganz am Anfang des L^AT_EX-Laufs am Terminal aus und schreibt sie zusätzlich in die Transkript-Datei, wenn diese zu dem Zeitpunkt schon geöffnet ist. Das ist zwar meist der Fall, nicht aber, wenn beim L^AT_EX-Lauf zusätzlicher T_EX-Code an der Befehlszeile übergeben wurde, zum Beispiel:

```
pdflatex '\PassOptionsToClass{11pt}{article} \input{myarticle}'
```

In diesem Fall werden die Release-Informationen zu dem Zeitpunkt ausgegeben, wenn `\PassOptionsToClass` verarbeitet wird. Die Transkript-Datei wird aber erst geöffnet, wenn der Name der Ausgabedatei bekannt ist – nachdem `\input` eingelesen wurde. Daher werden die Release-Informationen hier nur am Terminal ausgegeben.

Um dieses Szenario zu berücksichtigen, wiederholen wir nun die Release-Informationen auch ganz am Ende der Transkript-Datei, wenn wir sicher sein können, dass sie offen ist und Material übernehmen kann. (*Github Issue 944*)

Fehlerkorrekturen

Inkompatibilität zwischen doc und unicode-math

Das Paket `unicode-math` verändert den Catcode von `|`, passt aber nicht dessen Wert für die Verwendung in `doc` an. Das führt dazu, dass »or«-Module wie $\langle A|B \rangle$ seltsam

angezeigt werden. Dies wurde nun durch einen Fix behoben, der später noch nach `unicode-math` umgezogen werden wird. *(Github Issue 820)*

Eine Korrektur für `\hspace`

Im Jahr 2020 wurde `\hspace` geändert, um es für `calc` nutzbar zu machen. Das hatte leider den unerwünschten Nebeneffekt, dass der Beginn eines Absatzes mit `\hspace` zum Ausführen von `\everypar` innerhalb einer Gruppe führt (jede lokale Änderung würde also sofort zurückgenommen, was beispielsweise in dieser speziellen Situation `wrapfig` nicht mehr funktionieren lässt). Das wurde mit dem 2022-11 PL1 Hotfix behoben, war also schon im vorigen Release korrigiert, aber jetzt erst im Newsletter dokumentiert. *(Github Issue 967)*

Sicherstellen, dass `\cs` in `ltxdoc` definiert ist

Die Klasse `ltxdoc` hat den Befehl `\cs` definiert, um einen Befehlsnamen mit einem Backslash davor zu setzen. Diese Definition wurde in das Paket `doc` selbst verschoben, fehlte damit aber nun plötzlich, wenn zur alten `doc`-Paketimplementierung über die Klassenoption `doc2` zurückgekehrt wurde. Das wurde nun korrigiert. *(Github Issue 981)*

Abstände am oberen Rand von `minipages` verbessern

Eine Liste und diverse andere Dokumentelemente fügen vor sich vertikalen Abstand ein. Das sollte aber nicht am Anfang einer Box (wie `minipage`) geschehen und tut es normalerweise auch nicht, weil \TeX solchen Abstand automatisch zu Beginn einer vertikalen Liste verwirft. Gibt es aber unsichtbares Material, wie zum Beispiel einen `\color`-Befehl, einen `hyperref`-Anker, ein `\write` oder ähnliches, ist die Liste nicht mehr länger leer und \TeX verwirft den vertikalen Abstand nicht.

Mit der neuen, im Jahr 2021 eingeführten Absatzverwaltung ist es nun endlich möglich, dieses Problem zu erkennen, zu vermeiden und passende Gegenmaßnahmen zu ergreifen, so dass die Abstände jetzt immer korrekt sein werden. *(Github Issue 989)*

Eine Korrektur für `\NewCommandCopy` und `\ShowCommand`

Beim Kopieren und Anzeigen von Definitionen von (nicht-expandierbaren) Dokumentbefehlen (also Befehlen, die durch `\NewDocumentCommand` und seine Partner definiert wurden), die leere oder nur `m`-Typ-Argumente enthalten, wurden diese

Befehle fälschlicherweise als expandierbar erkannt. Das ist nun im aktuellen L^AT_EX-Release behoben. (Github Issue 1009)

Korrekturen für den Wechsel der Mathematik-Version

Es gibt interne Codeverbesserungen, die den Wechsel der Mathematik-Version unterstützen, wenn sie in einem äußeren Mathematik-Ausdruck eingebettet ist. Das verbessert `\boldsymbol`, `\bm` und ähnliche Befehle. (Github Issue 1028)

par als Dateinamen erlauben

`\input{par}` oder `\includegraphics{par}` konnte zu unberechtigten Fehlermeldungen führen. Das wurde nun behoben, indem ein interner Behl `\long` gemacht wurde. (Github Issue 942)

Korrektes Setzen von `\endlinechar` in `+v`-Argumenten

Wird ein Dokumenten-Befehl mit einem Argument vom Typ `+v` innerhalb von `\ExplSyntaxOn/Off` genutzt, würden Zeilenumbrüche fälschlicherweise als Leerzeichen interpretiert, weil der Wert von `\endlinechar` zu spät gesetzt wurde. Das ist nun korrigiert und Zeilenumbrüche werden nun korrekt in `»the character ^^M«` umgewandelt. (Github Issue 876)

Korrektes Verarbeiten von Hooks, die nur `'next'`-Code enthalten

Hat man `\AddToHookNext` mit einem noch nicht deklarierten Hook verwendet, wurde dieser Hook fälschlicherweise von `\ShowHook` als leer erkannt. Wurde der Hook dann später deklariert, wurde dieser `'next'`-Code trotzdem nicht ausgeführt. Das ist nun behoben, indem die Hook-Struktur beim Verwenden von `\AddToHookNext` korrekt initialisiert wird. (Github Issue 1052)

Leerraum nach `$$` ignorieren

Leerraum wird normalerweise nach einem schließenden `$$` ignoriert, aber der interne L^AT_EX-Code zum Font-Handling konnte dazwischengrätschen, wenn `\eqno` zum Einsatz kam. `\eqno` und `\leqno` wurden neu definiert, um nach der Mathematik-Gruppe `\ignorespaces` hinzuzufügen. (Github Issue 1059)

Verbesserungen an der Dokumentation

Aktualisieren der Guides

Als L^AT_EX 2_ε veröffentlicht wurde, hat das Team in den beiden Dateien `usrguide` und `clsguide` Dokumentation für Autoren, aber auch für Paket/Klassen-Entwickler bereitgestellt. Seitdem wurden diese Dokumente gelegentlich erweitert, wenn neue Methoden zum Kernel hinzukamen. Aber die Struktur blieb gleich und sie ging davon aus, dass man mit L^AT_EX 2.09 vertraut war. Für neue Anwender und Anwenderinnen gab es damit aber Material, das nicht mehr länger relevant war, und bezüglich der mittlerweile empfohlenen Vorgehensweisen gab es auch weniger Klarheit.

Die beiden Dateien wurden nun (teilweise) neu geschrieben, wobei die vorherigen Versionen mit ihrem letzten Stand als `usrguide-historic` und `clsguide-historic` vorhanden bleiben. Es ist mehr Material in den Klassen/Paket-Guide gewandert als in den User Guide, aber beide lohnen für erfahrene L^AT_EX-Benutzer ein erneutes Lesen.

Das genaue Releasedatum für L^AT_EX anzeigen

In manchen Situationen ist es notwendig, das genaue Releasedatum für ältere Versionen des L^AT_EX-Formats herauszufinden – zum Beispiel, wenn Sie in einem Paket abhängig von der Verfügbarkeit eines bestimmten Features unterschiedlichen Code nutzen müssen und daher `\ifFormatAtLeastTF{<date>}` oder die ziemlich gruselige Konstruktion `\@ifl@t@r\fmtversion{<date>}` nutzen, wenn Sie Formate berücksichtigen wollen, die älter als 2020 sind.

Oder Sie wissen, dass Ihr Paket definitiv nicht mit einem Format vor einem bestimmten `<date>` funktionieren wird, weshalb Sie `\NeedsTeXFormat{LaTeX2e}[<date>]` nutzen, um sicherzustellen, dass man bei einem zu alten Format benachrichtigt wird.

Das große Problem ist, das genaue `<date>` herauszubekommen, das in solche Befehle einzusetzen ist – in der Vergangenheit war das nicht leicht zu ermitteln. Sie könnten in der Datei `changes.txt` nachschauen, aber die ist irgendwo in Ihrer Installation versteckt und wenn Sie versuchen, `texdoc -l changes.txt` aufzurufen, erhalten Sie mehr als 30 Ergebnisse und die richtige Datei ist mit Sicherheit nicht die erste.

Yukai Chou (@muzimuzhi) hat freundlicherweise einen Patch dafür erstellt, so dass wir nun das genaue Datum für jedes L^AT_EX-Format an einem leicht zu merkenden Ort finden – in `ltnews.pdf`. Diese Datei enthält auch praktischerweise alle größeren Features und Änderungen von L^AT_EX im Verlauf der Jahre – vermutlich gehört dazu auch der Grund, warum Sie `<date>` überhaupt brauchen.

Das Datum wird nun in Klammern im Newsletter-Titel angegeben – dieser Newsletter teilt Ihnen daher mit, dass der Befehl `\NewEnvironmentCopy`, ein neuer `shipout-Hook` und so weiter am 1. Juni 2023 verfügbar gemacht wurden. Mit einem Blick in `ltnews.pdf` können Sie nun auch leicht herausfinden, dass der L^AT_EX3 Programming Layer am 2. Februar 2020 hinzugefügt wurde (weil das Datum so schön war) und nicht am Ersten des Monats. (Github Issue 982)

Frisch aus der Druckerpresse: »The L^AT_EX Companion, third edition« jetzt verfügbar

Die dritte Auflage von *The L^AT_EX Companion* ist nun verfügbar. Sie ist das Ergebnis von fünf Jahren sorgfältiger Arbeit und wir hoffen, dass Sie beim Lesen all die Informationen erhalten, die Sie für einen erfolgreichen Aufenthalt im L^AT_EX-Ökosystem benötigen, und dadurch wunderschöne Dokumente erzeugen können.

Seit der Veröffentlichung der letzten Auflage (2004) ist in der L^AT_EX-Welt viel geschehen und es war eine vollständige Überarbeitung notwendig. Alle Kapitel wurden umfassend begutachtet und in vielen Fällen deutlich erweitert, um neue, wichtige Funktionalität und Features zu beschreiben. Mehr als 5.000 Addon-Pakete wurden im Detail analysiert, von denen ungefähr 10 % in *The L^AT_EX Companion* aufgenommen wurden. Alle wichtigen Aspekte dieser Pakete sind beschrieben, um beim Lesen für das nächste Jahrzehnt keine anderen Quellen mehr benötigen zu müssen.²

Um all das abzudecken, was unserer Meinung nach heutzutage beschreibenswert ist, hat sich der Umfang des Buches nahezu verdoppelt. Die gedruckte Ausgabe wird daher in zwei Bänden erstellt und im Paket verkauft. Beide Bände erscheinen als Hardcover mit Lesezeichen, um Seiten im Buch leicht kennzeichnen zu können. Um eine Idee davon zu bekommen, was in der dritten Auflage behandelt wird, finden Sie ein paar Auszüge unter

<https://www.latex-project.org/news/2023/03/17/TLC3>

Die Auflage ist auch als eBook (Teil I und II kombiniert) verfügbar – im PDF- und ePub-Format, ohne DRM. Der Verlag bietet zudem eine Kombination aus gedrucktem Buch und Digitalversionen zu einem sehr attraktiven Preis an, der woanders nicht erreicht werden wird.³

² Anmerkung des Lektors des Originalartikels: Ein Review findet sich in *TUGboat* 44:2.

³ Leider hat es in den Datenbanken der Großhändler Fehler gegeben, so dass die beiden Bände, bzw. das Bundle bisher nicht oder nur eingeschränkt in Europa erhältlich sind. Viele Benutzer haben deshalb direkt in den USA bestellt, wo man bei Pearson/InformIT (siehe Link oben) aufgrund des Verlagsrabattes trotz Zoll einen attraktiven Preis erhält.

Änderungen an Paketen der tools-Kategorie

`multicol`: Bessere Unterstützung für CJK-Sprachen

Die standardmäßige minimale Tiefe jeder Spalte in einer `multicols`-Umgebung hängt von der Tiefe eines »p« im aktuellen Font ab. Das hilft für eine Einheitlichkeit, wenn Linien zwischen den Spalten genutzt werden, und ist für auf lateinischen Buchstaben basierenden Sprachen sinnvoll. Bisher war das festverdrahtet, aber für CJK (Chinese/Japanese/Korean)-Sprachen ist es besser, eine Tiefe von Null zu nutzen, weil alle Zeichen die gleiche Höhe und Tiefe besitzen. Und selbst bei auf dem lateinischen Schriftsystem basierenden Sprachen möchte man vielleicht die Tiefe eines `\strut` oder die einer Klammer nutzen. Daher bieten wir jetzt die Möglichkeit der Anpassung an, wobei Abwärtskompatibilität besteht: Definieren Sie `\multicolmindepthstring` neu mit etwas, dessen Tiefe ausgemessen werden soll (die Breite ist nicht relevant). (Github Issue 698)

`multicol`: Korrektur bei verschachtelten Umgebungen

Wenn `multicols`-Umgebungen ineinander verschachtelt wurden (die innere in einer Box), konnte das zu einem Fehler führen, wenn die sich in der Box befindliche Umgebung nahe einem Seitenumbruch gesetzt wurde. Das Problem war, dass die Ausgaberroutine aufgerufen wurde, während `\hsize` weiterhin so angepasst war, dass die Spaltenbreite zur inneren `multicols` passt – was das Platzieren der Spalten auf der Seite durcheinander brachte. Dies wurde nun behoben. (Github Issue 1002)

Literatur

- [1] Frank Mittelbach, Phelype Oleinik, L^AT_EX Project Team: L^AT_EX's hook management, run `texdoc lthooks-doc` to view.
- [2] Frank Mittelbach, L^AT_EX Project Team: The `ltshipout` documentation, run `texdoc ltshipout-doc` to view.
- [3] »L^AT_EX News 37, June 2023«, *TUGboat*, 42.2 (2023), ISSN: 0896-3207, <https://tug.org/TUGboat/tb44-2/tb1371tnews37.pdf>.

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss in umgekehrter chronologischer Reihenfolge vor. Bloße Updates können auf der moderierten *CTAN-ann*-Mailingliste oder als RSS-Feed auf <https://ctan.org/> verfolgt werden.

rit-fonts von *C. V. Radhakrishnan* enthält zehn Schriftarten, die das *Rachana Institute of Typography* (RIT) für den Textsatz in Malayalam, einer Landessprache der indischen Staat Kerala, entwickelt hat, samt der dazugehörigen \LaTeX -Unterstützung. Die Schriften liegen in den Formaten TrueType und WOFF2 vor.

CTAN: fonts/rit-fonts

hep-graphic von *Jan Hajer* sorgt für eine konsistente Anmutung von Plots und Feynman-Diagrammen mit *pgf/TikZ* und dem Paket *standalone*.

CTAN: macros/latex/contrib/hep-graphic

wordle von *Cédric Pierquet* dient zum Erstellen von Wordle-Spielen mit Hilfe von *TikZ*.

CTAN: graphics/pgf/contrib/wordle

beamertheme-rainbow von *samcarter* ist ein Beamer-Theme, das dafür sorgt, dass jede Folie mit einer anderen Farbe versehen wird. Standardmäßig durchläuft man damit die Farben des Farbspektrums bzw. des Regenbogens (daher der Name des Pakets).

CTAN: macros/latex/contrib/beamer-contrib/themes/beamertheme-rainbow

luaplot von *Chetan Shirore* und *Ajit Kumar* ermöglicht das Plotten von mathematischen Standardfunktionen mit $\text{Lua}\LaTeX$. Angeschlossen sind *TikZ* und *METAPOST* (letzteres via *luampplib*).

CTAN: macros/luatex/latex/luaplot

fail-fast von *Yegor Bugayenko* sorgt dafür, dass alle Warnungen zur Laufzeit als Fehlermeldungen ausgegeben werden, damit man darauf deutlicher aufmerksam gemacht wird.

CTAN: macros/latex/contrib/fail-fast

rouequestions von *Maryline Vignal Danilet* dient zum Setzen einer Art »Glücksrad« mit Fragen und verdeckten Antworten für einen spielerischen Unterricht.

CTAN: graphics/pgf/contrib/rouequestions

mlawriter von *Qifan Wang* formatiert ein Plain- \TeX -Dokument nach den Vorgaben des *MLA style* der *Modern Language Association*. Das Ergebnis sieht aus, »als

wäre es mit MS Word geschrieben worden«.

CTAN:macros/plain/contrib/mlawriter

profsio von *Cédric Pierquet* stellt Makros bereit, um den Mathematik-Teil für Prüfungen des *Brevet de technicien supérieur – Services informatiques aux organisations*, einen berufsbildenden Abschluss, zu zeichnen und zu setzen (Karnaugh-Veitch-Diagramme, Metra-Potenzial-Diagramme, einfache Graphen, Wahrheitstabellen).

CTAN:graphics/pgf/contrib/profsio

mitthesis von *John H. Lienhard* ist eine Klasse für Abschlussarbeiten am *Massachusetts Institute of Technology* nach den neuen Vorgaben aus dem Jahr 2023. Sie ersetzt den früher verwendeten *mitthesis.sty*, den es schon seit den 1980er Jahren gab. Die Bibliografie kann sowohl mit Bib_{TeX} mit *natbib* als auch mit Bib_{LaTeX} erstellt werden.

CTAN:macros/latex/contrib/mitthesis

beautybook von *Lu Shilong* ist eine Klasse für »schöne« Bücher in chinesischer und englischer Sprache.

CTAN:macros/latex/contrib/beautybook

rorlink von *Florian Lukas* dient zur Angabe und zum Verlinken der ID-Nummer einer Forschungseinrichtung im *Research Organisation Registry* (ROR) bei den Autorenangaben, ähnlich dem Paket *orcidlink* von *Leo Stein*.

CTAN:macros/latex/contrib/rorlink

quizztex von *Cédric Pierquet* erzeugt eine Auswahl an Fragen und Antworten mit einem Layout, wie man es aus den Einblendungen in Quizsendungen im Fernsehen kennt (»Wer wird Millionär?«).

CTAN:macros/latex/contrib/quizztex

luahttp von *Johannes Casaburi* stellt fünf Befehle bereit, um mit Lua und Lua_{TeX} Inhalte aus dem Internet zur Laufzeit abzurufen und in ein Dokument zu übernehmen. Auf diese Weise kann auf Schnittstellen und auf RSS-Feeds zugegriffen oder es können Bilder unter einer bestimmten URL abgerufen werden.

CTAN:macros/luatex/latex/luahttp

javascripthttp von *Cedric V. Zwahlen* stellt Befehle bereit, um Buttons in ein interaktives PDF-Dokument einzubauen, mit dem man Ressourcen aus dem Internet abrufen kann. Das Ergebnis kann man mit JavaScript verändern. Dazu müssen die PDF-Dateien in Adobe Pro geöffnet werden.

CTAN:web/javascripthttp

examz von *Christopher McClain* baut auf dem Paket *exam* von *Philip S. Hirschhorn* auf und bietet einige zusätzliche Möglichkeiten, beispielsweise für Fragen mit zufälligen Teilen oder zum Erstellen mehrerer Versionen einer Übung.

CTAN:macros/latex/contrib/examz

antique-spanish-units von *Noel Merino Hernández* ist eine kurze Übersicht in spanischer Sprache über die alten Maßeinheiten, die in Spanien und in den spanischen Kolonien vom 16. bis ins 19. Jahrhundert verwendet wurden. Eine Lösung, ähnlich zu dem Paket *siunitx*, soll folgen.

CTAN:info/spanish/antique-spanish-units

tilings von *Andrew Stacey* ist aus dem Paket *penrose* desselben Autors hervorgegangen, das nur Penrose-Parkettierungen zeichnen kann. *tilings* geht darüber hinaus und bietet weitere Möglichkeiten.

CTAN:graphics/pgf/contrib/tilings

profmaquette von *Christophe Poulain* ist ein Paket für den Mathematikunterricht in Frankreich, das es erleichtern soll, Aufgaben in verschiedenen Dokumenten wieder zu verwenden.

CTAN:macros/latex/contrib/profmaquette

jourrr von *Rok Kukovec* erleichtert es dem Verfasser, einen sogenannten *rebuttal letter* zu schreiben, in dem er auf die Kritik der Gutachter im Peer-Review-Verfahren für einen Zeitschriftenaufsatz eingehen kann.

CTAN:macros/latex/contrib/jourrr

string-diagrams von *Paolo Brasolin* zeichnet schöne String-Diagramme mit Hilfe von TikZ.

CTAN:graphics/pgf/contrib/string-diagrams

postit von *Cédric Pierquet* erstellt schöne farbige Postit-ähnliche Notizzettel, die man inline oder separat in ein Dokument einfügen kann, um etwas hervorzuheben oder um eine Anmerkung zu machen. Die Zettel sind vielfach konfigurierbar und können beliebige Inhalte aufnehmen, also beispielsweise auch mathematische Formeln.

CTAN:macros/latex/contrib/postit

tikz2d-fr von *Cédric Pierquet* stellt französische Befehle für einfache Zeichnungen mit TikZ bereit.

CTAN:graphics/pgf/contrib/tikz2d-fr

movement-arrows von *Alan Munn* dient zum Zeichnen von Pfeilen unter oder über Texten, die das Umstellen von Satzgliedern oder sonstigen Textbestandteilen anzeigen sollen. Es arbeitet mit den wichtigsten linguistischen Paketen *gb4e*, *linguex*, *covington* und *ExPex* zusammen.

CTAN:macros/latex/contrib/movement-arrows

counterz von *Christopher McClain* enthält Erweiterungen, mit denen man Zähler für unterschiedliche Input-Dateien unterscheiden und auch zufällige Zähler erzeugen kann.

CTAN:macros/latex/contrib/counterz

glossaries-norsk von *Jakob Stendahl* ist die Unterstützung für die norwegische Standardvarietät Bokmål zu dem Paket *glossaries* von *Nicola Talbot*.

CTAN:macros/latex/contrib/glossaries-norsk

adobeornaments von *Elijah Z Granet* kann Ornamente, die in einigen Schriften von Adobe »verborgen« enthalten sind, mit Hilfe von Lua_{TEX} und Xe_{TEX} darstellen.

CTAN:macros/unicodetex/latex/adobeornaments

ptlatexcommands von *Geraldo Pereira de Moraes Junior* stellt grundlegende \LaTeX -Befehle in einer portugiesischen Übersetzung bereit. So soll vor allem Geisteswissenschaftlern die Benutzung von \LaTeX erleichtert werden.

CTAN:macros/latex/contrib/ptlatexcommands

starray von *Alceu Frigeri* richtet sich an Paketautoren und führt vektor-ähnliche »Strukturen« zum Speichern von Daten ein, wie man sie aus der Programmiersprache C kennt. Der Name leitet sich ab von *structured arrays*.

CTAN:macros/latex/contrib/starray

unicode-math-input von »user202729« ermöglicht den Einsatz von Unicode-Symbolen im Mathematiksatz. Es unterscheidet sich von dem Paket *unicode-math*, indem es die Ausgabekodierung unberührt lässt.

CTAN:macros/latex/contrib/unicode-math-input

codedescribe von *Alceu Frigeri* ist ganz in \LaTeX 3 implementiert und stellt ein minimales Set von Befehlen zur Dokumentation von \LaTeX -Quelltext und Paketen zur Verfügung.

CTAN:macros/latex/contrib/codedescribe

huffman von *Maxime Chupin* zeichnet Huffman-Bäume mit Hilfe von METAPOST.

CTAN:graphics/metapost/contrib/macros/huffman

unifront von *Weifan Wei* erzeugt einen Fortschrittsbalken für den Bearbeitungsstand jedes Kapitels und des ganzen Dokuments, der zusammen mit einem Inhaltsverzeichnis abschnittsweise ausgegeben wird.

CTAN:macros/latex/contrib/unifront

sympycalc von *Cédric Pierquet* ermöglicht es, SymPy-Befehle beim Bearbeiten von \LaTeX -Dokumenten auszuführen.

CTAN:macros/luatex/latex/sympycalc

tikz3d-fr von *Cédric Pierquet* stellt Befehle bereit, um einfache dreidimensionale Zeichnungen mit Hilfe von *pgf/TikZ* zu erstellen.

CTAN:graphics/pgf/contrib/tikz3d-fr

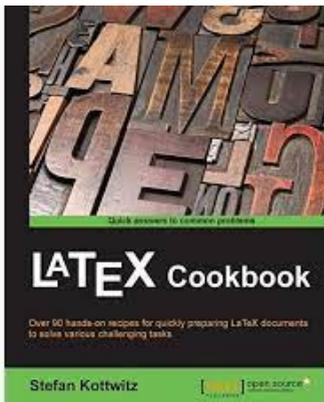
playcards von *Clément Pagès* zeichnet Spielkarten im Format 59 × 89 mm mit Hilfe von *pgf/TikZ*.

CTAN:graphics/pgf/contrib/playcards

Bücher

LaTeX Cookbook, 1. Auflage

Uwe Ziegenhagen



Stefan Kottwitz:

LaTeX Cookbook: Over 90 recipes to quickly prepare LaTeX documents of various kinds to solve challenging tasks

ISBN-10: 1784395145

ISBN-13 : 978-1784395148

Packt Publishing (28. Oktober 2015)

Nachdem ich mich bei Stefan Kottwitz »beklagt« hatte, dass sein LaTeX Beginner's Guide ja nichts über TikZ und biblatex erzählen würde, verwies Stefan auf sein 2015 veröffentlichtes LaTeX Cookbook, das er mir auch freundlicherweise für eine Rezension bereitstellte. Wer anders als ich nicht das Glück hat, ein kostenloses Exemplar zu erhalten, findet das Buch für etwas mehr als 30 Euro im online- oder offline-Buchhandel.

Aufbau

Das in englisch verfasste Buch beinhaltet auf 378 Seiten insgesamt 100 Kochrezepte zum Thema LaTeX, die sich in die folgenden Kategorien unterteilen:

- Variety of Text (Vorstellung von Dokumentenklassen)
- Tuning the Text (Akzente, Boxen, Absolute Positionierung)

- Adjusting Fonts (Schriftarten, Ligaturen, Schriftfamilien)
- Working with Images (Bilder, Bilder, Bilder...)
- Beautiful Designs (Hintergrundbilder, Ornamente)
- Designing Tables (Tabellensatz)
- Contents, Indexes and Bibliographies (Verzeichnisse und Literatur)
- Getting the Most out of PDF (PDF Metadaten)
- Creating Graphics (TikZ)
- Advanced Mathematics (Plotten von Funktionen)
- Science and Technology (Einheitensatz, Chemische Symbole)
- Getting Support on the Internet (Hilfe-Foren im Netz)

Innerhalb der einzelnen Kategorien gibt es die einzelnen Themen, wie beispielsweise »Creating a large poster« (»Große Poster erstellen«). Jedes Thema unterteilt sich in die folgenden Abschnitte:

- Einleitung, also worum es geht
- »How to do it...« eine narrensichere Schritt-für-Schritt-Anleitung mit den einzelnen Befehlen.
- »How it works...« mit einer Erläuterung, was man genau getan hat und
- »There is more...« mit weiterführenden Hinweisen auf alternative Pakete und Herangehensweisen

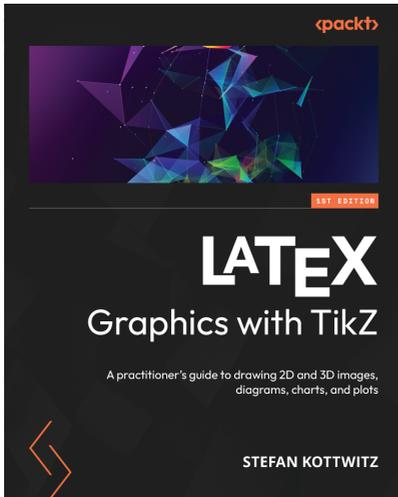
Fazit

Dieser Ansatz gefällt mir persönlich sehr gut. Auch thematisch habe ich noch einiges gelernt, das ich in meinen L^AT_EX-Kursen weitergeben kann. Insofern kann ich das Buch nur empfehlen.

Ein wenig schade ist, dass es das Cookbook nicht auch in einer deutschen Fassung gibt, es gäbe sicherlich einen – wenn auch kleinen – Markt dafür.

L^AT_EX Graphics with TikZ

Uwe Ziegenhagen



Stefan Kottwitz:

**L^AT_EX Graphics with TikZ: A
practitioner's guide to drawing
2D and 3D images, diagrams,
charts, and plots**

ISBN-10 : 1804618233

ISBN-13 : 978-1804618233

Packt Publishing (9. Juni 2023)

Just für den Tag, an dem unser Flug in den Süden starten sollte, hatte UPS die Lieferung von »L^AT_EX Graphics with TikZ« angekündigt. Glücklicherweise hatte der packt-Verlag mir nicht nur die gedruckte, sondern auch eine PDF-Version des neuesten Buchs von Stefan Kottwitz zur Verfügung gestellt, sodass dem ersten Reinschnuppern während des Fluges nichts im Wege stand.

Auf insgesamt 285 Seiten gibt Stefan eine ausführliche – englischsprachige – Einführung in TikZ. Da es eventuell noch Leserinnen und Leser gibt, die mit dem Begriff »TikZ« nichts anfangen können: TikZ und PGF sind zwei aufeinander aufbauende Pakete oder Paketsammlungen, mit denen sich sehr komplexe Grafiken erstellen lassen. In T_EX-Analogien übersetzt könnte man sagen: TikZ ist für PGF, was L^AT_EX für T_EX ist. Ursprünglich von Till Tantau, der uns auch Beamer beschert hat, werden TikZ und PGF seit einiger Zeit von einer kleinen Entwicklergemeinde vorangetrieben.

Aufbau

Das Buch gliedert sich in insgesamt fünfzehn Kapitel. Im ersten Kapitel werden kurz notwendige Begriffe sowie die Installation erklärt, im zweiten Kapitel geht es dann schon los mit der Erstellung der ersten Grafiken.

Die Kapitel 3 und 4 widmen sich dann den TikZ-Nodes und ihren Verbindungen untereinander. Kapitel 5 enthält Informationen, wie man in TikZ – ähnlich zu CSS – Style-Informationen erstellen kann, mit denen man einzelne Nodes grafisch formatieren kann.

Kapitel 6 enthält ausführliche Informationen, wie sich mit TikZ Bäume und Graphen zeichnen lassen.

Ab Kapitel 7 geht es dann ans »Eingemachte«, hier beginnt Stefan mit den komplexeren Themen:

Kapitel 7 Füllungen, Schattierungen und das Beschneiden von Grafiken

Kapitel 8 Pfad-Dekorationen

Kapitel 9 Ebenen, Transparenzen und mehr

Kapitel 10 Rechnen mit Koordinaten

Kapitel 11 Transformationen von Koordinaten und Zeichenfläche

Kapitel 12 Zeichnen von geglätteten Kurven

In Kapitel 13 werden die TikZ-Befehle zum Plotten von Funktionen in 2D und 3D vorgestellt; Kapitel 14 widmet sich dem Erstellen von »Business«-Diagrammen mit TikZ.

Das letzte Kapitel, Kapitel 15, ist dem Spaß mit TikZ gewidmet. Hier findet man Informationen zu den TikZ-lingen und lernt, wie man mit TikZ Pinguine und Schneemänner malen, ähhh, setzen kann.

Fazit

Dem Anspruch, eine Einführung in TikZ zu geben, wird das Buch von Stefan absolut gerecht. TikZ ist ein komplexes, aus meiner Sicht auch oft kompliziertes Thema, wobei es diesem Buch gelingt, die Lernkurve ein wenig angenehmer zu gestalten. Für meine L^AT_EX-Anfänger- und Fortgeschrittenenkurse werde ich es definitiv einsetzen.

Spielplan

10. 9. – 16. 9. 2023 **ConTExT Meeting 2023**
Prag-Sibřina, Tschechien
<https://meeting.contextgarden.net/2023/>



April 2024 (geplant) **DANTE 2024**
Wielandgut Oßmannstedt
Wielandstraße 16
99510 Ilmtal-Weinstraße, OT Oßmannstedt

Stammtische



In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von TeX-Anwendern statt, die für jeden offen sind. Im Web gibt es aktuelle Informationen unter <https://projekte.dante.de/Stammtische/WebHome>.

Aachen

Torsten Bronger

bronger@physik.rwth-aachen.de

Mailingliste: <https://lists.rwth-aachen.de/postorius/lists/tex-stammtisch>.

lists.rwth-aachen.de

»Anvers«, Kockerellstr. 20, 52062 Aachen

Erster Donnerstag im Monat, 20:00 Uhr

Bad Doberan

Carsten Vogel

texnicer@web.de

zur Zeit inaktiv, Interessenten bitte per Mail melden

Berlin

Michael-E. Voges, Tel.: 0 33 62/ 50 18 35,

mevoges@t-online.de

»La Esperanza Restaurant Tapas Bar«, Chausseestr. 131 B, 10115 Berlin

Zweiter Donnerstag im Monat, 19:00 Uhr

Darmstadt

Karlheinz Geyer

geyerk@posteo.de

ab 2023 Neustart geplant, Interessenten können sich bei Karlheinz melden

Erlangen

Peter Seitz

p.seitz@KplusS-Ing.de

<https://www.ks-ingenieurconsult.de/TeX/Stammtisch.html>

Gaststätte »Deutsches Haus«, Luitpoldstr. 25, 91052 Erlangen

Dritter Dienstag im Monat, 19:00 Uhr

Frankfurt a. Main

Harald Vajkonny

vajkonny@t-online.de

zur Zeit inaktiv, Interessenten bitte per Mail melden

Göttingen

Holger Nobach

holger.nobach@nambis.de





<http://goetex.nambis.de/>

*Restaurant »Mazzoni Cucina Italiana«, Hermann-Rein-Straße 2, 37075 Göttingen
Dritter Donnerstag im Monat, 18:00 Uhr*

Hamburg

Günther Zander

guenther.zander@lug-balista.de

zur Zeit inaktiv. Bei Fragen steht Günther gern per Mail zur Verfügung.

Hannover

Reiko Kaps

kaps@luis.uni-hannover.de

<http://tex-hannover.de/>

*RRZN/LUIS, 3D-Raum, Schloßwender Str. 6 (Gebäude 1210), 30159 Hannover
Zweiter Donnerstag im Monat, 18:30 Uhr, Terminabsprache über Mailingliste*



Heidelberg

Martin Wilhelm Leidig, Tel.: 01 70 41 83 32 9,

moss@moss.in-berlin.de

Anmeldeseite zur Mailingliste: <https://tinyurl.com/stammtisch-HD>

*letzter Freitag in Biergartenmonaten (ca. April bis Oktober), möglichst im Freien
Details und Abweichungen werden über die Mailingliste bekannt gegeben.*



Köln

Uwe Ziegenhagen

uwe@dante.de

zur Zeit inaktiv, Interessenten bitte per Mail melden

Leipzig

Erhard Pross

Erhard.Pross@gmx.de

nächstes Treffen am 11.Mai 2023 18:00 Uhr

le-tex publishing services GmbH, Weissenfeller Str. 84, 04229 Leipzig

München

Leah Neukirchen

leah@vuxu.org

Erste Woche in geraden Monaten, wechselnde Wochentage und Orte, 19:00 Uhr

Info folgt über Mailingliste dante-ev@dante.de

Stuttgart

Bernd Raichle

bernd.raichle@gmx.de

Zweiter Dienstag im Monat, 19:30 Uhr

»Wichtel«, Bahnhofstr. 30, 70372 Stuttgart - Bad Canstatt

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 11 03 61
69072 Heidelberg

Tel.: (0 62 21) 2 97 66
Fax: (0 62 21) 16 79 06
E-Mail: info@dante.de

Konto: VR Bank Rhein-Neckar eG
IBAN DE67 6709 0000 0002 3100 07 SWIFT-BIC GENODE61MA2

Vorstand

Vorsitzender:	Martin Sievers	president@dante.de
stv. Vorsitzender:	Uwe Ziegenhagen	vice-president@dante.de
Schatzmeisterin:	Doris Behrendt	treasurer@dante.de
Schriftführer:	Volker RW Schaa	secretary@dante.de
Beisitzer:	Klaus Höppner Harald König Stephan Lukasczyk Oliver Rath	

Ehrenmitglieder

Peter Sandner	22.03.1990	Klaus Thull († 2012)	22.03.1990
Yannis Haralambous	05.09.1991	Barbara Beeton	27.02.1997
Luzia Dietsche	27.02.1997	Donald E. Knuth	27.02.1997
Eberhard Mattes	27.02.1997	Hermann Zapf († 2015)	19.02.1999
Joachim Lammarsch	12.04.2014	Rainer Schöpf	12.04.2014

Webserver und Mailingliste

DANTE: <https://www.dante.de/> (Erik Braun)
CTAN: <https://mirror.ctan.org/> (Gerd Neugebauer)
DANTE-EV: <https://lists.dante.de/mailman/listinfo/dante-ev>

FAQ

DTK: <https://projekte.dante.de/DTK/WebHome>
T_EX: <https://projekte.dante.de/DanteFAQ/WebHome>

T_EXnische Fragen

beraterkreis@dante.de
ak-schule@dante.de

Autoren/Organisatoren

Christian Böttger boettger_ch@gmx.de	[26]	Henning Hraban Ramm hraban@fieee.net	[38]
Luzia Dietsche 71394 Kernen dtkred@dante.de	[3,7]	Martin Sievers siehe Seite 66	[4,5]
Jürgen Fenn Neu-Isenburg juergen.fenn@gmx.de	[55]	Pfarrer HG Unckell hgu@mailbox.org	[36]
Frank Mittelbach L ^A T _E X Project Team Mainz frank.mittelbach@latex-project.org	[42]	Keno Wehr wehr@abgol.de	[12]
		Uwe Ziegenhagen Köln	[59,61]

Die T_EXnische Komödie

35. Jahrgang Heft 3/2023 August 2023

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 5 Büroräumung – Bitte um Unterstützung
- 7 Eindrücke von Bonn, Mitgliederversammlung und TUG 23

Bretter, die die Welt bedeuten

- 12 L^AT_EX und Schulphysik 3: Messwertdiagramme
- 26 L^AT_EX per pandoc mit Markdown füttern
- 36 Erfahrungsbericht zum Einsatz von T_EX im Alltag eines Gemeindepfarrers
- 38 ConT_EXt kurz notiert

Von fremden Bühnen

- 42 L^AT_EX News – Issue 37, Juni 2023
- 55 Neue Pakete auf CTAN

Bücher

- 59 L^AT_EX Cookbook, 1. Auflage
- 61 L^AT_EX Graphics with TikZ

Spielplan

- 63 Termine
- 64 Stammtische

Adressen

- 67 Autoren/Organisatoren